Une Thèse de doctorat par

Alain Droniou

présentée à

Université Pierre et Marie Curie

École doctorale Informatique, Télécommunications et Électronique (EDITE)

Institut des Systèmes Intelligents et de Robotique

pour obtenir le grade de

Docteur en Informatique

Apprentissage de représentations et robotique développementale

Quelques apports de l'apprentissage profond pour la robotique autonome

Jury

Rapporteurs	Alain Dutech	Université de Lorraine, LORIA
	Pierre-Yves Oudeyer	INRIA Bordeaux Sud-Ouest
Examinateurs	Ludovic Denoyer	Université Pierre et Marie Curie, LIP6
	David Filliat	ENSTA ParisTech
	Michèle Sebag	Université Paris-Sud Orsay, LRI
Directeur de thèse	Olivier Sigaud	Université Pierre et Marie Curie, ISIR

Résumé

La question des représentations est un enjeu majeur pour la robotique : afin de pouvoir évoluer de manière autonome et sûre dans leur environnement, les robots doivent être capables d'en construire un modèle fiable et pertinent. Lorsque les tâches sont définies à l'avance ou les environnements restreints, des algorithmes dédiés peuvent doter les robots de mécanismes *ad-hoc* leur permettant de mener à bien leur mission. Cependant, pour des tâches variées dans des environnements complexes, il devient difficile de prévoir de manière exhaustive les capacités nécessaires au robot.

Il est alors intéressant de doter les robots de mécanismes d'apprentissage leur donnant la possibilité de construire eux-mêmes des représentations adaptées à leur environnement. Se posent alors deux questions : quelle doit être la nature des représentations utilisées et par quels mécanismes peuvent-elles être apprises ?

Cette thèse aborde le problème d'un point de vue développemental, arguant qu'afin de rendre possible une représentation de l'environnement permettant de faire face à des situations nouvelles, il est nécessaire que les représentations soient apprises de manière autonome et non-supervisée. Nous proposons pour cela l'utilisation de l'hypothèse des sous-variétés, définissant les *symboles* perceptifs et moteurs comme des sous-variétés dans des espaces sensoriels et fonctionnels, afin de développer des architectures permettant de faire émerger une représentation symbolique de flux sensorimoteurs bruts. Nous montrons que le paradigme de l'apprentissage profond fournit des mécanismes appropriés à l'apprentissage autonome de telles représentations. Dans un premier travail, nous démontrons que l'exploitation de la nature multimodale des flux sensorimoteurs permet d'en obtenir une représentation symbolique pertinente. Dans un second temps, nous étudions le problème de l'évolution temporelle des stimuli du point de vue de l'hypothèse des sous-variétés et de la robotique développementale. Nous discutons les défauts de la plupart des approches aujourd'hui utilisées et nous esquissons une approche à partir de laquelle nous approfondissons deux sous-problèmes. Nous étudions d'abord l'émergence de représentations des transformations entre stimuli successifs, puis nous testons la viabilité de l'hypothèse des sous-variétés dans des espaces fonctionnels pour expliquer notamment l'existence de répertoires d'actions. Dans une troisième partie, nous proposons des pistes de recherche pour permettre le passage des expériences de laboratoire à des environnements naturels. Nous explorons plus particulièrement la problématique de la curiosité artificielle dans des réseaux de neurones non supervisés.

Mots clefs

Apprentissage de représentations non-supervisé, Robotique développementale, Apprentissage profond

Table des matières

Table des figures

Notations

x	Valeur scalaire
$\mathbf{x}$	Vecteur
x_i	i$^{\text{ème}}$ composante du vecteur $\mathbf{x}$
X	Matrice
$\hat{x}$	Reconstruction ou estimation de la variable x
$.^\top$	Matrice adjointe (transposée conjuguée)
$*$	Produit terme à terme
σ	Fonction d'activation non-linéaire / sigmoïde (selon le contexte)
σ_+	Fonction *softplus*, $\sigma_+(x) = \log(1 + \exp(x))$
σ_{max}	Fonction *softmax*

Chapitre 1

Introduction

Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's ? If this were then subjected to an appropriate course of education one would obtain the adult brain.

Alan Turing - 1950

Sommaire

Pourquoi s'obstiner à écrire des programmes dédiés pour réaliser des tâches complexes, alors que l'on pourrait apprendre ce que l'on veut à un programme simulant l'esprit d'un enfant ? C'est en substance la question que pose Alan Turing en 1950 dans son article *Computing Machinery and Intelligence* (Turing 1950). Quelque peu oubliée aux débuts de l'intelligence artificielle – systèmes experts et raisonnement symbolique occupent alors le devant de la scène – cette remarque trouve aujourd'hui un formidable essor dans le domaine de la robotique développementale. Prolongeant l'idée de Turing, le but de ce champ de recherche est non seulement de développer des algorithmes capables de simuler le cerveau d'un enfant, mais également de les doter d'un corps leur permettant d'interagir directement avec leur environnement.

Ce programme peut sembler ambitieux : ne disposant pas encore de programmes capables d'apprendre ne serait-ce qu'à parler, il peut paraître vain de s'attaquer à un problème de prime abord plus complexe. Pourtant, le programme de la robotique développementale est susceptible de fournir la solution à un vieux problème de l'intelligence artificielle : le problème de l'ancrage des symboles. Initialement posé par Stevan Harnad (Harnad 1990) comme critique des approches purement symboliques de la cogni-

tion (Newell 1980), le problème de l'ancrage des symboles formule la nécessité de lier les symboles manipulés à des référents dans l'environnement. Harnad a proposé les réseaux connexionnistes comme candidats pour réaliser cette liaison.

En 1958, Frank Rosenblatt a proposé l'un des premiers algorithmes *apprenant* : le perceptron (Rosenblatt 1958). S'inspirant très librement du fonctionnement neuronal, le perceptron est un classifieur linéaire, dont la sortie dépend de la projection des entrées par multiplication avec une matrice de poids interne au perceptron, avant que ce résultat ne soit seuillé pour retourner une variable binaire. L'originalité de cet algorithme est de faire évoluer les valeurs de la matrice de poids au cours du temps, selon la différence observée entre la projection des entrées et les sorties désirées. La linéarité de ce problème en fait à la fois sa force, mais également sa faiblesse. Sa force, tout d'abord, car la règle d'apprentissage se traduit par une formule très simple de mise à jour parfaitement adaptée à la puissance de calcul de l'époque. Elle permet de résoudre des problèmes, certes simples, mais qui suscitent néanmoins un terrible engouement incitant Frank Rosenblatt à prédire qu'un perceptron sera capable de prendre des décisions et de traduire des langues [1]. Sa faiblesse également, car cette linéarité contraint grandement les problèmes solubles par cette méthode. Aussi, le livre *Perceptrons* de Minsky et Papert publié en 1969 (Minsky et Papert 1969) sonnera-t-il le glas [2] du perceptron en exhibant ses faiblesses, dont la plus connue est son incapacité à apprendre la fonction OU-Exclusif.

Il faudra attendre les années 1980 pour que l'apprentissage sur des réseaux de neurones retrouve ses lettres de noblesse en intelligence artificielle, grâce notamment à la publication de la technique de propagation du gradient dans des réseaux de neurones (Werbos 1974 ; Parker 1985 ; Le Cun 1986 ; Rumelhart et al. 1986). Cet algorithme étend la technique du perceptron en permettant de propager l'erreur entre la sortie produite et la sortie désirée dans plusieurs couches de neurones consécutives, et à travers des fonctions d'activation non-linéaires, s'affranchissant ainsi du principal défaut du perceptron.

Pour autant restait-il toujours un problème de fond : pour apprendre, ces réseaux ont besoin qu'on leur fournisse en permanence la *bonne* réponse. Le problème soulevé est illustré par John Searle à travers le dilemme connu sous le nom de *chambre chinoise* (Searle 1980). Considérons une pièce close, dans laquelle se trouve enfermé un homme qui peut envoyer et recevoir des symboles chinois. Celui-ci dispose également d'un livre, donnant des règles d'associations entre différents signes chinois. En appliquant bêtement les règles de ce livre, l'homme est capable de fournir une réponse cohérente aux interrogations qu'il reçoit de l'extérieur, sans pour autant comprendre un seul mot de chinois, mais donnant, du point de vue d'un observateur extérieur, l'impression d'être intelligent. Le réseau de

1. À la suite d'une conférence de presse à Washington le 7 juillet 1958 donnée par Frank Rosenblatt et le US Office of Naval Research, le *New York Times* écrit dans un article intitulé *New Navy Device Learns by Doing* :
The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence. Later perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech and writing in another language, it was predicted. (source (Olazaran 1996)).

2. De nombreuses controverses existent cependant autour du rôle joué par le livre *Perceptrons* dans le désintérêt qu'ont connu les réseaux de neurones à cette époque et dans la sur-interprétation qui en a été faite. Voir par exemple (Olazaran 1996) pour une discussion.

neurones que nous avons décrit auparavant se trouve dans la même situation : on lui a fourni le livre à travers un ensemble d'exemples entrée/réponse qu'il ne fait qu'appliquer par la suite. Dans ces deux cas, soutient Searle, l'intelligence se trouve au niveau de celui qui a été capable de fournir les règles d'associations correctes et de valider le fait que les réponses fournies sont bien les bonnes, et non dans celui qui se contente de les appliquer :

> *The information in the Chinese case is solely in the eyes of the programmers and the interpreters, and there is nothing to prevent them from treating the input and output of my digestive organs as information if they so desire.*
> *[. . .]*
> *The fact that the programmer and the interpreter of the computer output use the symbols to stand for objects in the world is totally beyond the scope of the computer. The computer, to repeat, has a syntax but no semantics.*
> – John Searle, *Minds, brains, and programs*, 1980 (Searle 1980)

Searle permet de mettre en lumière un aspect important du problème de l'ancrage des symboles : il ne suffit pas de lier bêtement stimulus et symbole correspondant pour obtenir un système intelligent.

À la fin des années 1980 se développe le courant de pensée de l'*encorporation*[3]. Celui-ci postule l'importance des relations entre le cerveau et le corps, et donc son environnement, pour le développement de l'intelligence. Cette dernière ne repose pas seulement dans un esprit manipulant des symboles et des concepts de haut niveau mais également dans le corps lui-même. En 1984, Braitenberg montre par exemple qu'il n'est nul besoin de représentations ou de calculs complexes pour construire un véhicule qui s'approche, ou au contraire qui fuit, des sources lumineuses (Braitenberg 1984). Une simple connexion directe entre des photodiodes et des moteurs reliés aux roues suffit. Pour développer des comportements plus complexes, il faut alors apprendre à exploiter efficacement les possibilités de son corps, ses actions, par rapport à des stimuli, ses sensations, ce qui suppose d'apprendre des contingences sensori-motrices (O'Regan et Noë 2001). C'est ainsi que les symboles, au lieu d'être donnés *a priori* par des experts, pourraient émerger d'une structuration du flux sensorimoteur : plutôt que de fournir une définition précise de "chaise", il vaut mieux la voir comme un ensemble de stimuli qui ont la propriété, quand ils sont présents, d'associer une certaine sensation ("être assis"), à une certaine commande motrice ("s'asseoir"). On retrouve ici la théorie des *affordances* développée par J.J. Gibson en 1977. D'après cette théorie (Gibson 1977), les agents perçoivent leur environnement sous forme de *possibilités d'action*, qui sont suggérées par les objets eux-mêmes. Les objets se définissent alors à travers les relations qu'il existe entre les actions que l'on peut faire sur eux, ou avec eux, et les effets produits.

Dès lors que l'on adopte ce point de vue, impossible pour un ordinateur d'être intelligent comme peut l'être un enfant de 10 ans, et encore moins de le devenir, puisque, coupé du monde physique, il ne pourra jamais saisir l'essence des objets et des concepts. Pour cela, il faut le doter d'un corps. Poussée dans la communauté robotique par Rodney Brooks et

3. Nous choisissons la traduction proposée et défendue par Jean-Paul Laumond lors de son cours de robotique au Collège de France en 2011-2012 afin de traduire le terme anglais *embodiment*.

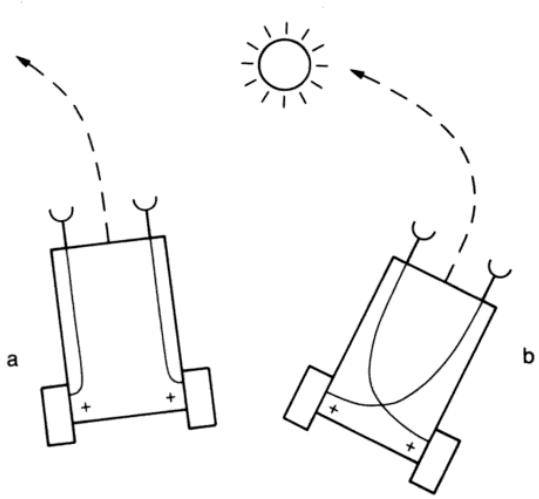


Figure 1.1 – *Les véhicules de Braitenberg exhibent des comportements cohérents à partir de leur seule structure physique. Ceux-ci sont dotés de photodiodes qui produisent un signal électrique lorsqu'elles sont éclairées. En reliant les photodiodes aux roues du même côté, le véhicule va exhiber un comportement de fuite devant une source lumineuse. Au contraire, en croisant les liaisons entre roues et photodiodes, le véhicule exhibe un comportement d'approche. (Image d'après (*Braitenberg *1984))*

Rolf Pfeifer notamment (Brooks 1991 ; Pfeifer et Scheier 1999 ; Pfeifer et Bongard 2007), qui ont montré que des comportements complexes et cohérents pouvaient être obtenus par une conception mécanique appropriée et réduisant le besoin de recourir à des algorithmes complexes, la théorie de l'*encorporation* et l'idée d'Alan Turing de simuler le cerveau d'un enfant ont donné naissance à la robotique développementale (Asada et al. 2001 ; Lungarella et al. 2003 ; Weng 2004) : dotons l'ordinateur d'un corps et des mécanismes d'apprentissage de l'enfant, et l'on peut espérer concevoir des robots sachant donner du sens à leur environnement et donc s'adapter efficacement à des situations nouvelles et complexes.

Cette thèse s'inscrit dans le cadre de la robotique développementale et aborde plus précisément le problème de l'apprentissage autonome de représentations. Avant de détailler plus avant les techniques et problématiques en robotique et intelligence artificielle manipulées lors de cette thèse, il paraît nécessaire de développer quelques éléments sur les sources d'inspirations exposées ci-dessus. Les capacités d'apprentissage de l'enfant restent en effet un idéal pour tout chercheur en robotique développementale. Nous nous pencherons sur quelques aspects de cette psychologie du développement. Après avoir décrit les approches robotiques correspondantes, nous exposerons notre problématique.

Figure 1.2 – *À gauche : robot industriel Kuka opérant dans des conditions maîtrisées pour une tâche bien définie (image : Kuka Systems GmbH). À droite : voiture sans conducteur développée par Google, qui doit s'adapter à des environnements ouverts en partie non prédictibles (image : Google).*

1.1 Robotique autonome et robotique développementale

Que l'enfant parle d'abord selon la structure de son corps, cela ne peut étonner personne. Qu'il parle ainsi son propre langage, par mouvements, cris variés ou gazouillements, sans savoir le moins du monde ce qu'il dit, cela n'est pas moins évident. Comment le saurait-il, tant qu'il n'est pas compris ? [...] Ainsi l'enfant apprend sa propre langue, il apprend ce qu'il demande d'après la chose qui lui est donnée.

Alain - Les Idées et les Âges, 1927

Le premier essor des robots est attribuable au développement de la robotique industrielle. Leur rôle était alors de suppléer l'homme pour des tâches répétitives et potentiellement difficiles, dangereuses ou fatigantes.

Ces robots agissaient donc dans des environnements clos, dans des usines parfaitement contrôlées, et exécutaient des tâches entièrement prédictibles et définies à l'avance (figure 1.2, gauche).

De nos jours, l'utilisation de robots est envisagée pour des tâches beaucoup plus complexes comme l'assistance à la personne ou la conduite autonome. Ces tâches nécessitent des robots capables d'évoluer dans des environnements ouverts, de s'adapter à des situations nouvelles dont il est impossible de prévoir la diversité à l'avance (figure 1.2, droite). Une première approche de ces tâches a été proposée, adaptation des systèmes experts à la robotique, qui attribue à l'ingénieur une place prépondérante dans les capacités des robots. La prochaine section décrit cette approche et en discute les limites. Dans une seconde section, nous présenterons conjointement quelques aspects de la psychologie du développement intellectuel chez l'enfant et les travaux de robotique développementale qui s'en inspirent.

1.1.1 Une approche "systèmes experts" de la robotique autonome

Le premier pas vers la robotique autonome a consisté à doter les robots d'une autonomie de décision. Dans ce cadre, le robot s'appuie sur une vaste base de connaissances, appelée ontologie et programmée à l'avance par des ingénieurs, pour planifier ses actions en fonction de l'environnement perçu et de la tâche courante (Lemaignan et al. 2010 ; Ingrand et Ghallab 2014). Dans un premier temps, ces bases de connaissances servent au robot à extraire une représentation de l'environnement à partir de son flux sensoriel, en appariant des concepts avec des ensembles de caractéristiques perçues. Par exemple, la perception d'une surface plane et horizontale, d'environ $1m^2$, à une hauteur d'environ 1m par rapport au sol sera identifiée comme une table. Dans un second temps, le robot utilise la base de connaissances pour inférer l'action qui produira l'effet désiré dans l'environnement perçu : pour pouvoir nettoyer la table, il faut d'abord aller chercher une éponge (*petit objet de forme parallélépipédique, d'environ 10 cm de long pour 2 cm de haut et 5 cm de large, plutôt mou, avec une face verte et une face jaune*), l'humidifier, donc aller au robinet (nous n'essaierons pas d'en donner une description...), l'ouvrir, mettre l'éponge sous le jet d'eau, fermer le robinet, presser l'éponge, retourner à la table, la nettoyer, revenir à l'évier, rincer l'éponge puis la reposer. L'ontologie utilisée se doit donc de fournir un ensemble de règles permettant au robot de choisir l'action à accomplir à chaque instant.

Une telle approche requiert une très lourde tâche d'ingénierie, afin de fournir des ontologies assez complètes pour contenir les connaissances nécessaires à chaque situation. En particulier, les concepts les plus simples pour un être humain peuvent être très difficiles à traduire sous la forme d'ontologies manipulables par un robot (figure 1.3). De plus, une grosse partie de ce travail est spécifique à chaque robot, selon le type de capteurs et d'actionneurs dont il dispose, et limite quoi qu'il arrive les capacités du robot à ce qui a été envisagé lors de son développement. On pourrait arguer que certaines approches utilisant le contenu d'Internet (Tenorth et al. 2011 ; Waibel et al. 2011 ; Tenorth et Beetz 2013) pourraient fournir assez de connaissances pour rendre un robot autonome. Elles se heurtent cependant à la barrière de l'ancrage des symboles (Harnad 1990) : soit elles utilisent le web sémantique, c'est-à-dire s'appuient sur des "experts" humains qui ont indiqué par avance les relations entre différents concepts, ce qui n'est aucunement différent des approches classiques, soit elles doivent "comprendre" la connaissance contenue sur Internet ce qui, comme nous l'avons déjà évoqué, est impossible sans apprentissage autonome.

En effet, ces approches posant le symbole *a priori* puis cherchant à l'identifier dans l'environnement inversent les liens entre perception et représentation défendus notamment par les théories des contingences sensorimotrices ou des affordances que nous avons mentionnées auparavant. Cette inversion des relations n'est pas sans conséquence d'un point de vue théorique, puisqu'elle est à l'origine du problème soulevé par Searle : donner les symboles et des règles de manipulations ne suffit pas à rendre un système intelligent, puisqu'il peut alors n'avoir aucune notion de ce qu'il est en train de manipuler, ce qui ne lui permettra donc pas de faire face à une situation ambigüe ou sortant légèrement du

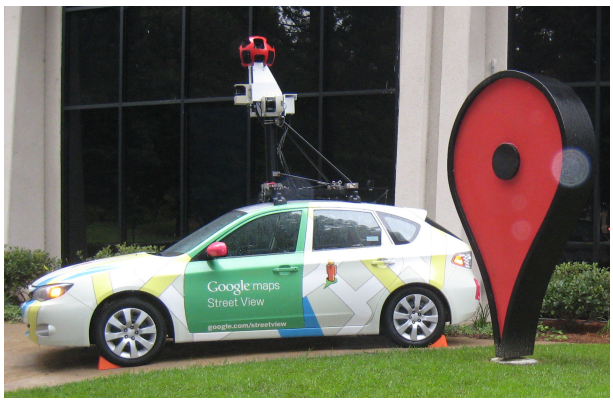


Figure 1.3 – *"Devant la voiture." Voici une expression qui paraît extrêmement simple et que l'on aimerait pouvoir utiliser pour communiquer avec un robot autonome.* "Ne va pas devant la voiture, c'est dangereux !" "Qu'est-ce que c'est cette statue devant la voiture ?" *Bien que ces deux phrases contiennent exactement la même expression, l'endroit désigné n'est pas le même. L'interprétation de l'expression dépend fortement du contexte. Elle nécessite de construire un cadre de référence, qui peut être égocentré ou exocentré. On peut éventuellement imaginer des solutions pour le cas égocentré qui ne soient pas trop complexes (par exemple : "si distance(moi,x) + distance(x,y) = distance(moi,y), alors x est devant y"), ce qui n'est pas le cas du cadre exocentré. Il faudrait pour cela attacher un repère à chaque objet ou entité susceptible d'être utilisé dans l'expression : devant la chaise, devant l'ordinateur, devant la boulangerie, devant la porte, etc. Mais ce repère, pour un même objet, dépend lui-même de la situation ("Le facteur attend devant la porte" (à l'extérieur de la maison), "Où ai-je posé mon parapluie ? Devant la porte !" (à l'intérieur de la maison)). Il faudrait de plus une règle permettant de décider entre l'utilisation du cadre égocentré ou exocentré... Intégrer tout cela dans une ontologie semble épineux.*

cadre de ces connaissances. Il faut pour cela *ancrer* les symboles dans la réalité (Harnad 1990), c'est-à-dire relier les symboles à des entités perceptuelles, afin de leur donner un sens. Plusieurs problèmes se posent alors. Quels sont les symboles *élémentaires* qui doivent nécessairement être ancrés, par rapport à ceux qui peuvent se définir uniquement à partir de règles et de ces symboles élémentaires ? Comment les ancrer ? Comment être sûr que tout concept peut bien être décrit avec les symboles élémentaires choisis, en évitant les définitions circulaires ?

Une autre ligne de défense des approches ontologiques consiste à suggérer qu'il n'y a qu'à lier les symboles, toujours donnés *a priori*, à des entités physiques par apprentissage : on apprend que tel ensemble de pixels correspond à une éponge, que tel autre ensemble correspond à une table, etc. Devant un objet inconnu, il suffirait alors qu'un humain indique de quoi il s'agit pour que le robot puisse l'apprendre et l'intégrer à ses connaissances. C'est possible, mais toujours comme l'a montré Searle, le robot n'aura pas plus *compris* l'objet, on lui aura juste donné une règle d'association supplémentaire. Il est d'ailleurs frappant de voir comment les réseaux de neurones profonds (dont nous reparlerons au chapitre 3) entraînés de manière supervisée à classifier des images et qui obtiennent pour cette tâche les performances de l'état de l'art [4], peuvent être trompés d'une manière qui nous paraît totalement incongrue (figure 1.4).

Afin de doter les robots d'une plus grande autonomie et versatilité, il est nécessaire qu'ils soient capables d'apprendre par eux-mêmes toutes ces connaissances : il faut leur donner une autonomie de perception et de représentation en plus de l'autonomie de décision.

1.1.2 Vers une robotique autonome réellement autonome : la robotique développementale

Au début du vingtième siècle, Jean Piaget a consacré de nombreuses années à l'étude et à l'observation de ses propres enfants, travail qui a donné lieu à la publication de deux ouvrages fondateurs qui font toujours référence actuellement (Piaget 1936, 1937). Depuis, de nombreuses études ont tenté d'identifier et de décrire les mécanismes clés de l'apprentissage. Ces travaux sont une source d'inspiration évidente pour de nombreux algorithmes en robotique développementale.

Smith et Gasser ont discerné six principes importants dans le développement intellectuel des enfants (Smith et Gasser 2005) : la multimodalité, l'apprentissage incrémental, les interactions physiques avec l'environnement, l'exploration et la curiosité, les interactions sociales et l'apprentissage du langage. Trois d'entre eux sont plus particulièrement importants dans nos travaux : l'interaction physique avec l'environnement, la multimodalité et la curiosité artificielle.

4. Lors de la dernière édition du challenge ILSVRC qui consiste à classifier les images de la base ImageNet, les vainqueurs (Szegedy et al. 2014) ont proposé un réseau profond à 22 couches qui atteint un taux d'erreur de classification de 6.67%, lorsque deux humains entraînés à classifier la même base de données atteignent respectivement 5.1% et 12.0% d'erreur (Russakovsky et al. 2014).

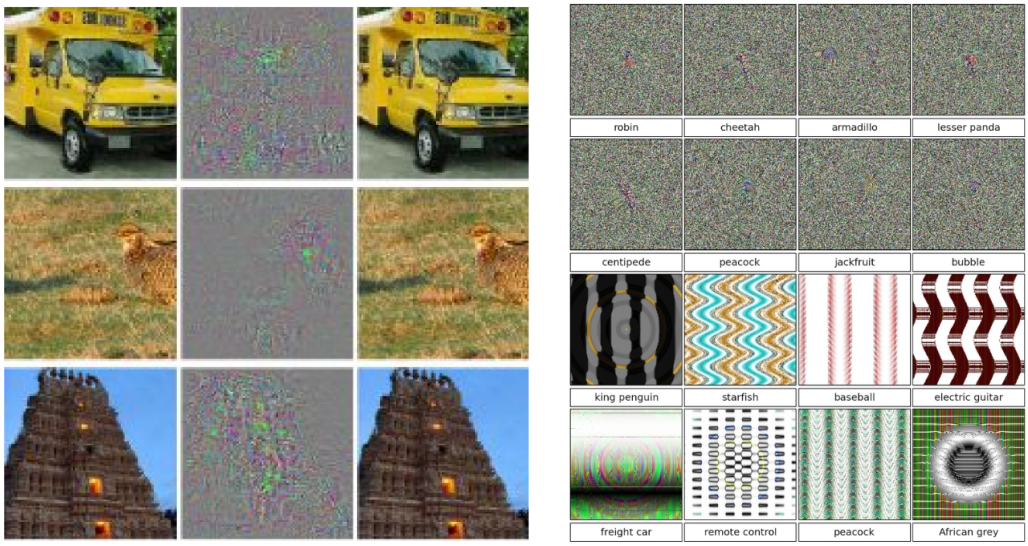


Figure 1.4 – *Des réseaux de neurones profonds entraînés à classifier des images de manière supervisée peuvent être trompés de manière étonnante. À gauche : image extraite de (*Szegedy *et al. 2013), qui montre sur la colonne de gauche des images correctement classifiées par un réseau, mais qui, légèrement modifiées en ajoutant un masque (colonne centrale), donnent des images mal classifiées (colonne de droite), alors que les changements sont imperceptibles pour un humain. À droite : image extraite de (*Nguyen *et al. 2014), qui montre des images classifiées avec une très grande confiance (supérieure à 99.6%) par un réseau entraîné sur la base ImageNet (les catégories prédites sont indiquées en dessous de chaque image), alors qu'elles n'ont aucune signification pour un humain.*

Interactions physiques avec l'environnement

Dans (Gibson 1977), l'auteur consacre le terme *affordance* pour désigner le fait que l'environnement est perçu de manière propre à chaque agent, en fonction de ses capacités : une chaise n'est perçue en tant que chaise que par un agent capable de s'asseoir. Les objets se définissent alors comme des entités de l'environnement possédant un certain nombre d'affordances. C'est la combinaison de ces affordances et non la possession d'une affordance précise qui définit l'objet : tout objet possédant l'affordance "*asseyable*" n'est pas une chaise (on peut par exemple s'assoir sur le bord d'un bureau ou sur un rocher).

Le cadre théoriques des *complexes action-objet-effet* (Montesano et al. 2008) (aussi appelés *effet-entité-comportement* dans (Şahin et al. 2007)) constitue l'implémentation robotique la plus populaire de la théorie des affordances. Malheureusement, ces approches utilisent souvent des représentations symboliques données *a priori*, sous la forme notamment de répertoires d'actions (toucher, saisir, lancer, pousser, etc.) (Cos-Aguilera et al. 2004 ; Moldovan et al. 2012) et d'objets prédécoupés dans le flux visuel (on identifie d'abord les zones de l'image contenant un objet, de laquelle on extrait ensuite des descripteurs divers et variés) (Cos-Aguilera et al. 2004 ; Montesano et Lopes 2009 ; Akgun et al. 2009 ; Ugur et al. 2011). De plus, les représentations utilisées sont souvent construites de manière *ad-hoc* pour l'expérience : si l'on veut par exemple apprendre un modèle de "roulabilité", on va décrire les objets à l'aide de descripteurs de forme (par exemple des histogrammes de normales à la surface de l'objet), et les effets en terme de distances parcourues entre le point initial (avant l'action) et final (en laissant un délai suffisant pour la stabilisation de l'objet) (Akgun et al. 2009).

Nous illustrerons aux chapitres 4 et 5 nos travaux sur des tâches robotiques, à l'aide du robot humanoïde iCub. Nous nous attacherons en particulier au chapitre 4 à ne pas fournir de connaissances *a priori* sous la forme de descripteurs pertinents extraits manuellement pour les tâches considérées.

Multimodalité

Nous interagissons avec l'environnement par l'intermédiaire de multiples sens : vue, ouïe, toucher, odorat, goût, proprioception. Chacune de ces modalités reflète une facette d'une même réalité physique, ce qui introduit de nombreuses redondances dans ce que nous percevons. Nous pouvons ainsi faire face à des situations où certains sens nous font défaut, par exemple si nous devons évoluer dans une pièce plongée dans le noir. Cela permet également à chaque modalité de guider l'apprentissage pour les autres modalités, à travers leurs corrélations temporelles et spatiales. Ainsi, lorsque je touche une pomme, je peux lier les sensations tactiles à l'apparence de l'objet touché et à ses mouvements provoqués par mes actions, puis, portant la pomme à ma bouche, je peux y associer son goût et son odeur.

Dans le cadre d'un apprentissage non supervisé, chaque modalité fournit donc aux autres modalités des informations qui peuvent être dans une certaine mesure vues comme des *labels* qui permettent de se rapprocher d'un cadre supervisé. Nous reviendrons plus en détail sur la multimodalité, qui joue un rôle central dans notre travail, et les travaux

qui s'y rattachent dans les chapitres 2 et 4. Nous verrons notamment en quoi les cartes auto-organisatrices utilisées dans de nombreux travaux peuvent être un problème dans des environnements complexes.

Il peut être tentant d'utiliser la multimodalité comme argument pour réintroduire de manière *ad-hoc* des représentations symboliques dans les approches développementales (Cangelosi et Harnad 2001 ; Cangelosi et Riga 2006 ; Nakamura et al. 2009 ; Madden et al. 2010 ; Nakamura et al. 2011). Par exemple, lorsque leur enfant regarde un objet, les parents ont tendance à prononcer le nom de l'objet regardé, permettant ainsi à l'enfant d'associer une perception visuelle au mot correspondant (Gogate et al. 2001). La même chose se produit lorsque les parents commentent les actions de leur enfant, qui peut ainsi associer certains verbes à ses actions. Il peut alors être tentant de considérer que ces interactions placent l'enfant (ou le robot) dans un cadre *supervisé* : lorsque l'on montre un bol au robot et qu'on lui dit que c'est un *bol*, ne lui fournit-on pas l'équivalent d'un *label* ? Cette vision est beaucoup trop simpliste. Il est facile d'oublier que les mots ne sont des mots qu'à partir du moment où l'on est capable de les segmenter, puis de les identifier, dans un flux auditif continu. De même, il est tentant d'utiliser un prédécoupage *a priori* des objets dans la scène transformant alors l'espace visuel en un ensemble de vignettes qu'il ne reste qu'à relier à une représentation symbolique, contraignant alors les capacités du robot par l'algorithme de prédécoupage utilisé.

Le rôle du langage dans l'apprentissage est cependant important. Le langage est une forme très particulière de communication, puisqu'elle consiste en l'échange d'une représentation symbolique du monde. Chaque mot se réfère à toute une catégorie de stimuli, dont les propriétés sont généralement complètement découplées de la forme même du mot (la similarité visuelle et sonore des mots *chat* et *chaud* ne traduit aucune similarité de sens). Une telle représentation a plusieurs avantages outre le fait que, partagée par tous les membres d'une communauté, elle permet une communication efficace. Tout d'abord, elle permet de créer ou au contraire de rompre des corrélations entre plusieurs stimuli : désigner deux stimuli différents par le même mot crée une corrélation susceptible d'unifier ces stimuli au sein d'une même catégorie, tandis que désigner deux stimuli similaires par deux mots différents permet de rompre cette similitude pour créer deux catégories différentes (Smith et al. 2002 ; Samuelson 2002). Nous reviendrons sur cet aspect au chapitre 2.

Exploration et curiosité

N'ayant quasiment aucune connaissance *a priori*, la découverte du monde offre au nouveau-né une infinité de possibilités qui rendent tout apprentissage particulièrement complexe. L'enfant doit à la fois apprendre mais aussi découvrir ce qu'il y a à apprendre.

Découvrir ce qu'il y a à apprendre nécessite d'explorer son environnement. Les comportements des jeunes enfants ne sont pas toujours dirigés vers des buts, ce qui leur permet de découvrir de nouveaux effets dans l'environnement. Ces effets peuvent ensuite devenir des buts à reproduire pour les enfants qui viennent de les découvrir : pour cela, ces derniers commencent par répéter l'action effectuée au moment de l'apparition de ce nouvel effet (Rovee et Rovee 1969), puis apprennent progressivement à l'affiner et à reproduire

l'effet recherché avec de plus en plus de précision, ce que Piaget désigne sous le nom de *réaction circulaire* (Piaget 1936).

Ce passage d'une exploration aléatoire à une exploration dirigée vers un effet particulier introduit la notion de curiosité : l'enfant porte alors ses efforts sur un aspect précis de son environnement, en développant ainsi de nouvelles capacités. Plusieurs études (voir notamment (Berlyne 1960 ; Csikszentmihalyi 1991)) montrent que ces efforts ne sont pas portés au hasard. Au contraire, l'enfant semble en permanence se focaliser sur des tâches qui se situent à la frontière entre les capacités qu'il maîtrise déjà et les capacités qui sont encore trop complexes par rapport à ses compétences actuelles.

Le problème formel de la curiosité et de la motivation intrinsèque est illustré dans (Lopes et Oudeyer 2012) à partir de l'exemple d'un étudiant devant passer des examens dans K matières et devant maximiser sa moyenne générale. Son problème est alors d'allouer son temps de révision aux différentes matières de manière optimale, étant donné qu'il a plus de facilité dans certaines matières (il apprend donc plus vite) et que réviser certaines matières a un impact positif sur d'autres matières (en révisant ses mathématiques, l'étudiant sera aussi plus à l'aise en physique). Si l'étudiant est capable d'estimer l'effet qu'aura une heure de révisions en plus dans une matière sur sa note, il a alors intérêt à choisir à chaque fois la matière pour laquelle ce gain sera le plus fort. On peut voir à partir de cet exemple que certaines approches qui pourraient sembler intuitives sont en réalité très mauvaises. Par exemple, choisir d'étudier la matière dans laquelle on a la plus mauvaise note est largement sous-optimal dès lors qu'il existe une matière trop difficile que l'étudiant a beaucoup de mal à maîtriser : il va alors y "gâcher" son temps. Au contraire, choisir la matière dans laquelle l'étudiant est le meilleur est également un mauvais choix, puisqu'il y a peu d'améliorations à attendre d'une heure de révisions supplémentaire.

La curiosité consiste donc à s'intéresser à chaque instant à des tâches pour lesquelles le taux d'apprentissage, défini par rapport à une certaines fonction à optimiser, est maximal (Oudeyer et al. 2007 ; Schmidhuber 2010). Ce cadre théorique a été appliqué avec succès dans de nombreuses expériences. Cela soulève toutefois le problème de l'estimation de ce gradient pour chaque tâche, étant donné que les mesures peuvent être bruitées, que la fonction à optimiser peut être définie de manière implicite et difficilement mesurable, etc. Le dilemme de l'exploration/exploitation apparaît également : pour estimer ce gradient pour différentes tâches, il est nécessaire de les réaliser périodiquement afin de ne pas rester focalisé sur un tâche devenue sous-optimale en termes de progrès d'apprentissage. De plus, le fait d'appliquer les algorithmes de curiosité à des espaces de buts et non à des espaces moteurs permet d'améliorer la performance des agents (Baranes et Oudeyer 2013).

Ces approches reposent donc généralement sur la définition de représentations adéquates, notamment pour un espace de buts, qui sont données *a priori* par l'ingénieur (variables pertinentes pour décrire un but, caractéristiques perçues de l'environnement, fonction définissant le succès de l'apprentissage, etc.).

Nous reviendrons au chapitre 6 sur la problématique de la curiosité artificielle dans des réseaux de neurones non-supervisés.

1.2 L'enjeu des représentations

We pay too much attention to the details of algorithms. [...] We must begin to subordinate the engineering to the philosophy.

John Hartigan - 1996

Nous avons dressé dans la section précédente un bref aperçu de quelques problématiques posées par la robotique développementale. Néanmoins, dans la plupart des travaux, la question de la nature des représentations apprises et manipulées est centrale mais souvent négligée. En effet, les travaux de psychologie développementale s'intéressent principalement aux principes généraux permettant un développement de l'intelligence, mais peu à la nature même des représentations apprises et manipulées par le cerveau. Les travaux robotiques quant à eux simplifient souvent à l'extrême les problématiques d'apprentissage de représentations : utilisation directe de représentations symboliques (notamment pour les actions), extractions de caractéristiques *ad-hoc* en fonction des problèmes traités, etc.

Or, comme nous l'avons expliqué dans notre critique des approches à base d'ontologies, le passage de stimuli bruts perçus dans l'environnement à une structuration du flux perceptif haut niveau, notamment sous forme de symboles, est un problème fondamental. Dans la section suivante, nous introduisons les théories de codage prédictif qui forment une famille populaire de modèles cognitifs. Nous présentons ensuite notre problématique et les contributions de cette thèse.

1.2.1 Le rôle de la prédiction pour la cognition

Nous venons de décrire quelques-uns des principes directeurs du développement intellectuel chez l'enfant. Cependant, ceux-ci n'avancent pas d'explication sur ce qui est réellement appris par le cerveau. Une des théories les plus influentes actuellement avance la prééminence de la prédiction comme mécanisme fondamental : le cerveau structure le flux sensorimoteur d'une manière qui lui permet de le prédire aussi précisément que possible (Clark 2013). Dans ce cadre, le système nerveux est considéré comme un système dynamique auquel l'environnement impose des "conditions aux limites", par l'intermédiaire de la stimulation de nos capteurs sensoriels, qui sont intégrées par notre cerveau de manière à être correctement prédites. La plupart de ces théories supposent de plus un traitement hiérarchique de l'information : plusieurs couches de traitements sont empilées, chaque couche apprenant à prédire au mieux l'activité de la couche précédente.

Ces théories font donc jouer un rôle primordial à l'erreur de prédiction, certaines allant jusqu'à supposer que c'est en fait la seule information montante dans le cerveau. Nous reviendrons sur l'une de ces théories, la théorie de la minimisation de l'énergie libre, au chapitre 2.

Ces théories reposent le plus souvent sur des approches bayésiennes modélisant le flux sensorimoteur sous forme de distributions de probabilités. D'un point de vue computationnel, la difficulté est alors de proposer des algorithmes permettant d'approcher la solution optimale avec une complexité calculatoire raisonnable (Friston et al. 2007). Une approche consiste à définir un flot de calculs qui permet de minimiser de manière

approchée les erreurs de prédiction (ou à maximiser la probabilité des observations). On trouve dans cette famille des réseaux de neurones stochastiques, comme les machines de Helmholtz (Dayan et al. 1995), les machines de Boltzmann (Ackley et al. 1985) et les machines de Boltzmann restreintes [5] (Smolensky 1986). Ces dernières ont récemment connu un regain d'intérêt dans la communauté de l'apprentissage automatique, en étant utilisées au sein de réseaux profonds, obtenant des résultats de l'état de l'art dans de nombreux domaines. Nous y reviendrons au chapitre 3.

Si les théories modélisant le cerveau comme machine prédictive permettent de proposer un cadre unificateur pour de nombreux aspects de son fonctionnement, elles nécessitent toutefois elles-mêmes l'introduction de variables *ad-hoc*, laissant ouverte la question de la structure même des représentations apprises (Clark 2013) sur laquelle nous reviendrons au chapitre 2.

1.2.2 Problématique et contributions

Cette thèse vise à proposer des mécanismes permettant le passage de la perception brute aux représentations symboliques par apprentissage autonome. Les théories à base de codage prédictif et génératif font l'hypothèse implicite qu'une représentation purement symbolique n'est pas suffisante, puisqu'en particulier elle ne permettrait pas de prédire avec précision la diversité des stimuli représentés par un même symbole. C'est pourquoi, après avoir étudié quelques caractéristiques de la perception et de l'action chez l'homme, le chapitre 2 présente notre hypothèse de travail sur la nature des représentations apprises.

Le chapitre 3 présente quant à lui les algorithmes de réseaux de neurones et d'apprentissage profond que nous utilisons tout au long des travaux présentés dans cette thèse. En effet, en plus d'être une méthode de référence en apprentissage automatique, l'apprentissage profond propose comme nous le verrons un cadre intéressant pour l'apprentissage non supervisé, en adéquation avec les théories de codage prédictif.

Nous présentons ensuite dans le chapitre 4 une architecture de traitement de stimuli multimodaux permettant l'émergence d'une représentation symbolique. Cette architecture ne prenant pas correctement en compte l'aspect temporel des flux sensoriels d'un point de vue développemental, nous développons au chapitre 5 quelques éléments d'architecture pour le traitement de flux temporels.

Comme nous l'avons vu au cours de cette introduction, l'apprentissage de représentations n'est qu'une facette de la robotique développementale. Nous approfondissons les problématiques de l'apprentissage par renforcement et de la curiosité artificielle à l'aune de l'apprentissage profond au chapitre 6. Nous discutons ensuite au chapitre 7 quelques critiques souvent adressées à l'encontre de l'apprentissage profond dans un cadre développemental ainsi que les limites de nos travaux pour leur application concrète dans un environnement ouvert et complexe.

5. *Restricted Boltzmann Machine*, RBM.

1.2.3 Publications

Plusieurs articles ont été publiés au cours de cette thèse. Ce manuscrit n'en exposera cependant qu'un sous-ensemble.

Travaux présentés dans ce manuscrit

Chapitre 4

Alain Droniou, Serena Ivaldi et Olivier Sigaud (2014, in press). « Deep unsupervised network for multimodal perception, representation and classification ». Dans : *Robotics and Autonomous Systems*

Chapitre 5

Alain Droniou et Olivier Sigaud (2013). « Gated Autoencoders with Tied Input Weights ». Dans : *Proceedings of International Conference on Machine Learning*, p. 154–162

Alain Droniou, Serena Ivaldi et Olivier Sigaud (2014). « Learning a Repertoire of Actions with Deep Neural Networks ». Dans : *Proceedings of ICDL-EpiRob*. Italie

Travaux annexes et collaborations

Apprentissage de modèles cinématiques

Alain Droniou, Serena Ivaldi, Vincent Padois et Olivier Sigaud (oct. 2012a). « Autonomous Online Learning of Velocity Kinematics on the iCub : a Comparative Study ». Dans : *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS*. Vilamoura, Portugal, p. 3577–3582

Alain Droniou, Serena Ivaldi, Patrick Stalph, Martin Butz et Olivier Sigaud (2012c). « Learning Velocity Kinematics : Experimental Comparison of On-line Regression Algorithms ». Dans : *Proceedings Robotica*, p. 15–20

Alain Droniou, Serena Ivaldi et Olivier Sigaud (2012b). « Comparaison expérimentale d'algorithmes de régression pour l'apprentissage de modèles cinématiques du robot humanoïde iCub ». Dans : *Conférence Francophone sur l'Apprentissage Automatique (Cap)*, p. 95–110

Architectures cognitives pour la robotique développementale

Serena Ivaldi, Natalia Lyubova, Damien Gerardeaux-Viret, Alain Droniou, Salvatore Anzalone, Mohamed Chetouani, David Filliat et Olivier Sigaud (sept. 2012a). « A cognitive architecture for developmental learning of objects and affordances : perception and human interaction aspects ». Dans : *IEEE Ro-man Workshop on Developmental and bio-inspired approaches for social cognitive robotics*. Paris, France

Serena Ivaldi, Natalia Lyubova, Damien Gerardeaux-Viret, Alain Droniou, Salvatore Anzalone, Mohamed Chetouani, David Filliat et Olivier Sigaud (2012b). « Perception and human interaction for developmental learning of objects and affordances ». Dans : *Proc. of the 12th IEEE-RAS International Conference on Humanoid Robots - HUMANOIDS*, p. 1–8

Sao Mai Nguyen, Serena Ivaldi, Natalia Lyubova, Alain Droniou, Damien Gerardeaux-Viret, David Filliat, Vincent Padois, Olivier Sigaud et Pierre-Yves Oudeyer (2013). « Learning to recognize objects through curiosity-driven manipulation with the iCub humanoid robot ». Dans : *Proc. IEEE Int. Conf. Development and Learning and on Epigenetic Robotics - ICDL-EPIROB*, p. 1–8

Serena Ivaldi, Sao Mai Nguyen, Natalia Lyubova, Alain Droniou, Vincent Padois, David Filliat, Pierre-Yves Oudeyer et Olivier Sigaud (2014). « Object learning through active exploration ». Dans : *IEEE Transactions on Autonomous Mental Development* 6.1, p. 56 –72

Chapitre 2

Perception, action et sous-variétés

A perceptual process does not start with the stimulus ; rather, the stimulus is an END of the process, like the last piece of a jig-saw puzzle, which fits in its place only because all the other pieces have been placed in a particular way. [...] A stimulus is present only if there is an organisation into which it can be fitted.

T. Jarvilehto - 1998

Sommaire

"*Nous proposons le terme d'énaction dans le but de souligner la conviction croissante selon laquelle la cognition, loin d'être la représentation d'un monde prédonné, est l'avènement conjoint d'un monde et d'un esprit à partir de l'histoire des diverses actions qu'accomplit un être dans le monde*" écrivent Varela et ses collègues en 1993 (Varela et al. 1993). Les théories de psychologie développementale placent les boucles sensorimotrices au cœur de l'intelligence humaine, alors que nombre d'expériences de robotique développementale relèguent perception et action à de simples tâches d'extraction de caractéristiques et d'exécution de primitives motrices définies manuellement. Dans ce chapitre, nous approfondirons les problématiques de la perception et de l'action et présenterons

succinctement quelques théories qui les unifient dans un cadre commun. Dans une dernière partie, nous présenterons l'hypothèse des sous-variétés comme cadre mathématique sur lequel s'appuiera la suite de notre travail.

2.1 La perception, un problème difficile

Si toutes mes perceptions étaient supprimées par la mort et que je ne puisse ni penser ni sentir, ni voir, ni aimer, ni haïr après la dissolution de mon corps, je serais entièrement annihilé et je ne conçois pas ce qu'il faudrait de plus pour faire de moi un parfait néant.

Hume - Traité de la nature humaine, 1740

Dans la théorie de l'énaction qui nous a servi d'ouverture à ce chapitre, le monde n'acquiert un sens qu'à travers l'existence d'agents définis comme des systèmes *auto-poïétiques*, c'est-à-dire dont le but est de s'auto-reproduire. Dès lors, la manière dont est perçu le monde n'est pas une image du monde physique, mais une interprétation utile à l'agent pour survivre. Le sucre, par exemple, n'acquiert la signification de nutriment qu'à partir du moment où des bactéries le métabolisent dans le but de survivre et se reproduire. En cela, le sens du monde est propre à chaque agent et se construit petit à petit selon l'historique de ses interactions – le sucre ne devient sucre que s'il a déjà été consommé au moins une fois par l'agent. Il n'y a donc pas de perception sans développement, pas de perception sans apprentissage. La perception est un phénomène guidé à la fois par l'existence physique d'entités dans l'environnement et par l'état interne de l'agent à un instant t, lui même influencé par les perceptions passées. Face à cette complexité, il n'est alors pas très étonnant que de multiples mécanismes entrent en jeu et qu'il soit aujourd'hui encore si difficile de construire des robots capables de percevoir le monde comme nous le faisons.

Il est important de distinguer la perception de la sensation. Le corps humain est doté d'une multitude de capteurs : les yeux, les oreilles, la peau ne sont que quelques exemples des plus utilisés. En permanence, et de manière non consciente, ces capteurs traduisent en influx nerveux les stimuli présents dans notre environnement. Mais ces informations de nos *sens*, nos *sensations*, ne sont pas toujours *perçues* correctement. Comme nous le verrons dans la suite de ce chapitre, la perception peut prendre beaucoup de liberté par rapport à nos sens. Nous pouvons percevoir une absence de stimulus, tout comme nous pouvons ne pas percevoir la présence de stimuli. Nous pouvons également percevoir sans stimuli, comme lorsque nous rêvons, et certaines perceptions nécessitent en réalité plusieurs sens, c'est le cas par exemple du goût qui nécessite non seulement – cela n'étonnera personne – le sens du goût mais également l'odorat, comme un bon rhume aime à nous rappeler à quel point les aliments peuvent être insipides lorsque l'on a le nez bouché. La perception n'est donc pas sensation, mais plutôt *notre* interprétation de la réalité.

Traiter de la perception dans son ensemble serait une tâche de bien trop grande ampleur dans le cadre de cette thèse, ce pourquoi nous nous limiterons à quelques-uns de ses aspects. Nous en donnerons tout d'abord quelques éléments biologiques, tirant quelques enseignements des activités neuronales impliquées dans la perception. Du point de vue

fonctionnel ensuite, nous étudierons quelques phénomènes qui doivent nous guider dans l'implémentation de mécanismes perceptifs.

2.1.1 Biologie de la perception

De nombreuses études s'intéressent au substrat biologique de la perception. Nous n'en ferons pas une étude exhaustive mais nous nous contenterons de décrire quelques résultats.

Des représentations distribuées...

En 1959, Hubel et Wiesel étudient l'activité des neurones dans l'aire visuelle primaire des chats (Hubel et Wiesel 1959). Cette aire primaire, aussi appelée aire V1, constitue la porte d'entrée dans le cortex visuel des informations en provenance de la rétine.

Grâce à leur expérience, Hubel et Wiesel ont montré que cette aire est organisée en colonnes corticales, dans lesquelles tous les neurones répondent aux mêmes stimuli. Ces stimuli correspondent majoritairement à des barres de contraste, chaque colonne corticale répondant préférentiellement à une orientation donnée dans une zone du champ visuel précise. Les colonnes corticales sont en effet organisées selon une carte rétinotopique, c'est-à-dire de manière homéomorphe au champ visuel. Ainsi, la zone de la fovéa se projette-t-elle sur une surface plus grande que les régions périphériques de la rétine. Cette projection rétinotopique et le fait que toutes les orientations sont détectées dans l'aire V1 explique la structure en forme de "moulinet" (*pinwheel* en anglais) de cette aire cérébrale (voir figure 2.1) (Petitot 2003).

Une telle disposition implique une représentation distribuée des stimuli : la vue d'un visage provoquera ainsi l'activation de neurones couvrant une large zone de l'aire V1, le décomposant en une superposition de bordures et de traits saillants.

Cette représentation distribuée se retrouve également dans d'autres aires sensorielles. Au niveau du cortex auditif, par exemple, des neurones répondent préférentiellement à la présence d'une fréquence précise dans les sons perçus (Schreiner 1992).

L'émergence de telles représentations semble découler d'un mécanisme commun. En 2000, Sur et ses collègues opèrent des furets à leur naissance pour isoler dans l'hémisphère gauche le cortex auditif des signaux nerveux en provenance des oreilles et pour y connecter à la place le nerf optique, l'hémisphère droit étant laissé intact pour contrôle. Le champ visuel droit des furets est donc à partir de ce moment là traité par les neurones du cortex auditif gauche. Les furets sont alors élevés normalement. Lorsqu'ils ont grandi, on observe l'organisation de leur cortex auditif : des structures en moulinet similaires à celles observées dans un cortex visuel normal se sont développées (Sharma et al. 2000). De plus, en faisant faire au furet un exercice qui demande de répondre différemment à des sons ou à des flashs lumineux, placés tantôt à droite, tantôt à gauche, il est possible de montrer que des signaux dans le champ visuel droit (traités donc par le cortex auditif gauche), sont bien interprétés par le furet comme des signaux de nature visuelle, et non de nature auditive (Von Melchner et al. 2000). Cette expérience illustre le fait que notre perception de l'environnement peut émerger à partir de mécanismes très généraux, sans nécessiter d'invoquer des traitements de nature différente entre les différents sens.

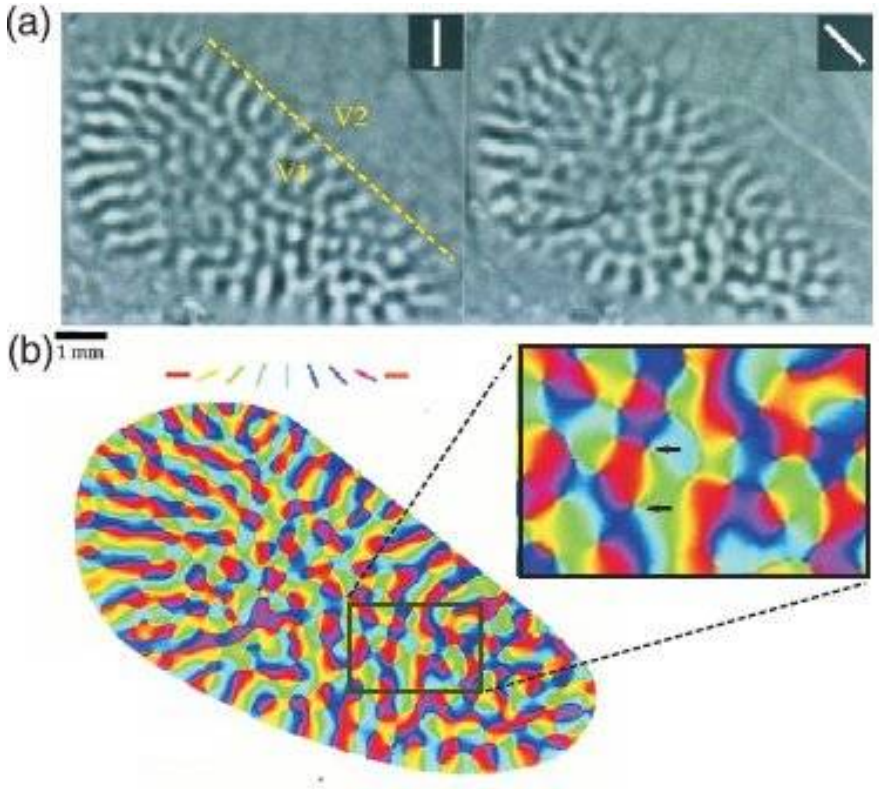


Figure 2.1 – *Carte de la réponse des neurones V1 en fonction de l'orientation des stimuli. En haut : imagerie optique des motifs d'activité des neurones de l'aire V1 en fonction de l'orientation des stimuli (vertical puis oblique). En bas : Carte des préférence d'orientation calculées à partir de l'imagerie optique. La carte fait apparaître des singularités appelées "moulinets". (Image : (*Kaschube *et al. 2008))*

Cette représentation distribuée se prolonge dans les aires supérieures. Une disposition rétinotopique a également été observée dans les aires visuelles suivantes (V2, V3, V4) (Silver et Kastner 2009 ; Henriksson et al. 2012), mais de manière de plus en plus floue : chaque neurone de ces aires reçoit en entrée des informations en provenance d'un champ visuel rétinien de plus en plus large. Certains neurones de ces aires supérieures ont donc un champ récepteur qui couvre une très grande partie du champ visuel (Gross et al. 1969). De tels neurones peuvent dès lors être le support d'une représentation de beaucoup plus haut niveau de l'information perçue que l'on peut parfois assimiler, comme nous allons le voir par la suite, à une représentation symbolique.

. . . aux neurones *grand-mère*

Le terme de neurone *grand-mère* a été introduit pour la première fois par Jerry Lettvin (Gross 2002), pour désigner un neurone qui répondrait de manière spécifique à un concept ou objet complexe.

Des neurones répondant spécifiquement aux visages ont été mis en évidence chez le macaque (Desimone et al. 1984), au niveau du cortex temporal inférieur. D'autres neurones de cette zones répondent quant à eux à la présence d'une main dans le champ visuel. Chez l'homme, des indices suggèrent que des neurones de l'hippocampe ont une réponse spécifique à certaines catégories de stimuli (Kreiman et al. 2000).

Bien qu'étant un argument en faveur d'une représentation symbolique des stimuli dans certaines aires cérébrales, ces résultats n'en sont toutefois pas une preuve. En effet, il est d'une part impossible de prouver que ces neurones ne répondent à aucun autre stimulus, et il est d'autre part fortement probable que d'autres neurones répondent également aux mêmes stimuli. L'hypothèse la plus communément admise est l'hypothèse d'une représentation parcimonieuse (Quiroga et al. 2008) : chaque stimulus est représenté non pas par un seul neurone, mais par l'activation coordonnée d'un très petit nombre de neurones dont la répartition permet de représenter chaque stimulus de manière discriminante.

Cependant, de tels neurones peuvent représenter une information très haut niveau. Chez l'homme, il a été mis en évidence que certains neurones, en plus de présenter une réponse invariante par certaines transformations physiques (face/profil par exemple), répondent également à des stimulations d'autres modalités. Ainsi, certains neurones répondant à la présentation d'une image de l'actrice Halle Berry, répondent également à la présentation du nom écrit "Halle Berry", alors qu'ils ne répondent pas à la présentation d'autres actrices américaines (Quiroga et al. 2005). D'autres neurones vont répondre spécifiquement à des images de Luke Skywalker, à la présentation de son nom écrit, ou à l'écoute de son nom, prononcé par un homme ou par une femme (Quiroga 2012).

La coexistence de représentations distribuées et quasi-symboliques a également été mise en évidence pour d'autres sens, telle la localisation dans l'espace (Moser et al. 2008). Ainsi, des *grid cells* tirent leur nom de leur réponse qui dessine une grille dans l'espace euclidien : en enregistrant l'activité isolée d'un seul neurone chez un rat qui se déplace librement, il peut-être mis en évidence une réponse localisée au niveau des sommets de triangles pavant l'espace euclidien. Différents neurones répondent selon des pavages décalés et selon plusieurs tailles de pavage. Ainsi, un point de l'espace est codé par l'activation

Figure 2.2 – *Peinture de Rodolphe II par Gisueppe Arcimboldo. Dès le premier coup d'œil, on y voit un visage, et non un simple de tas de fruits et légumes. Ceci illustre le principe de la Gestalt : le tout est différent de la somme de ses parties.*

de plusieurs neurones, ce qui le définit au croisement de pavages à plusieurs échelles. À l'inverse, des *place cells* répondent spécifiquement lors de la présence en un point donné de l'espace. Ces cellules répondent donc selon un découpage de l'espace sous forme de tuiles, chaque tuile pouvant être considérée comme un symbole.

2.1.2 Des mécanismes intriqués

De nombreuses œuvres et expériences montrent qu'il est assez facile de piéger nos sens et tromper notre perception. Au delà d'une simple distraction, les dysfonctionnements ainsi mis en évidence sont une source utile d'information pour tenter de comprendre ce qu'est et ce que fait réellement la perception.

Selon la théorie de la Gestalt (Kohler 1929), *le tout est différent de la somme de ses parties*. Ainsi sur l'image 2.2 perçoit-on un visage humain et non un simple amas de légumes.

De même, la figure 2.3 illustre le phénomène de multistabilité perceptive : il est possible de percevoir une même image de plusieurs manières différentes, et il est possible de sauter entre les interprétations de façon volontaire ou involontaire.

Nous énumérons de nombreux mécanismes qui ont été mis en évidence pour tenter de rendre compte de ce type de phénomène dans les sections suivantes.

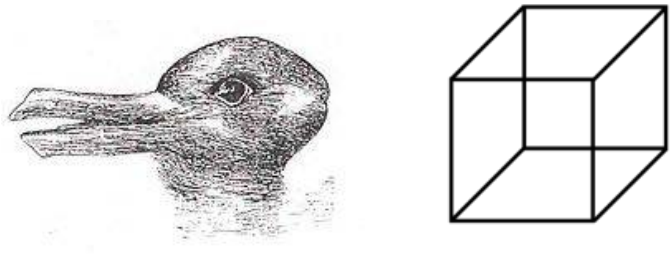


Figure 2.3 – *Ces deux images illustrent le phénomène de multistabilité perceptive. L'image de gauche peut-être perçue tantôt comme un canard, tantôt comme un lapin, le cube de droite (appelé cube de Necker) peut-être perçu soit orienté en haut à droite, soit en bas à gauche (selon la face perçue comme étant la face avant).*

Fusion multimodale

L'existence de neurones recevant en entrée des informations en provenance de différentes modalités a été mise en évidence expérimentalement, à l'aide à la fois de données anatomiques (Fort et al. 2002 ; Falchier et al. 2002) et d'études de sujets atteints de lésions cérébrales (Damasio 1989a,b, 1990). D'autre part, plusieurs illusions peuvent s'expliquer grâce à l'existence de ces neurones multimodaux.

Ainsi, l'effet McGurck (Mcgurck et Macdonald 1976) se manifeste lorsque les stimuli visuels et auditifs perçus sont en contradiction. En observant le mouvement des lèvres correspondant par exemple à la présentation du son *ga* et en écoutant le son *ba*, la majorité des sujets perçoit en réalité le son *da*.

L'illusion de la main en caoutchouc (Botvinick et Cohen 1998) met quant à elle en évidence l'existence d'un couplage entre perception visuelle et perception tactile. Pour cette illusion, un sujet regarde sa main tout en étant équipé de lunettes qui lui montrent en réalité un bras en caoutchouc. Une stimulation tactile sous la forme d'une caresse au dos de sa main est alors couplée à la visualisation d'une même caresse sur la main en caoutchouc. Après plusieurs répétitions, un coup de marteau est porté sur le bras factice : le sujet ne peut s'empêcher de retirer vivement sa main et rapporte percevoir la douleur due au coup alors même qu'il n'a pas été touché.

La multimodalité est centrale dans la théorie des zones de Convergence-Divergence (Damasio 1989b ; Meyer et Damasio 2009), que nous présenterons par la suite, et joue un rôle important dans nos travaux présentés au chapitre 4.

Hiérarchie, interactions montantes et descendantes

Nous avons déjà parlé de représentations hiérarchiques à partir de l'activité neuronale des aires sensorielles, par exemple avec le passage de neurones sélectifs à l'orientation de l'aire V1 au neurone spécifique à "Halle Berry". Seul un mécanisme hiérarchique peut également expliquer pourquoi nous percevons des visages dans les peintures d'Arcimboldo (figure 2.2) et non pas seulement des tas de légumes. Les mécanismes donnant lieu à ces représentations hiérarchiques sont cependant complexes et encore largement mystérieux.

Figure 2.4 – *L'impression de pouvoir percevoir deux images en même temps (ici une clef et une pipe) est une reconstruction a posteriori faite par notre cerveau. En réalité, à chaque instant, nous ne percevons que l'objet sur lequel notre attention est portée. (Image :* (Lachaux *2011))*

Un aspect important est l'existence de nombreuses connexions descendantes, c'est-à-dire reliant les aires supérieures aux aires inférieures (par opposition aux connexions montantes). Ces connexions permettent aux aires supérieures de moduler l'activité des aires inférieures, et ainsi de transformer une poire en nez dans les portraits d'Arcimboldo. Ces connexions descendantes sont cruciales pour la perception : il a en effet été montré que l'activation des aires corticales inférieures par les connexions descendantes est nécessaire pour avoir une perception visuelle consciente (Bullier 2001).

Ces constructions hiérarchiques jouent un rôle crucial dans notre motivation à utiliser l'apprentissage profond, présenté au chapitre 3, pour la suite de nos travaux.

Rôle de l'attention

L'attention joue également un rôle déterminant dans la perception. En effet, bien que notre intuition puisse nous faire croire le contraire, nous ne percevons à chaque instant que ce à quoi nous portons attention. L'impression de percevoir plusieurs objets, d'embrasser la totalité d'une scène, n'est qu'une reconstruction *a posteriori*.

La figure 2.4 en donne une illustration. En regardant cette image, la plupart des personnes ont le sentiment de voir deux objets à la fois : une pipe et une clef. Cependant, les patients atteints de simultagnosie sont incapables de percevoir les deux objets simultanément, à cause d'une lésion à la jonction du lobe pariétal et du lobe occipital dans les deux hémisphères (Rafal 2003 ; Lachaux 2011, p. 89-91).

D'autres expériences illustrent le rôle de l'attention dans ce que nous percevons. La cécité au changement (Rensink et al. 1997, 2000), par exemple, illustre à quel point notre sentiment de percevoir une scène en totalité n'est qu'une illusion : dès lors qu'un stimulus perturbant l'attention est inséré entre la présentation de deux images similaires (comme un écran gris, un flash lumineux, ou le fait de devoir tourner la page par exemple, voir figure 2.5), il est extrêmement difficile de percevoir une modification, potentiellement importante, d'un des éléments de la scène. Ce phénomène est bien connu des cinéastes et de leurs faux raccords, qui passent inaperçus la plupart du temps.

La cécité inattentionnelle (Simons 2000) ressemble à la cécité au changement, à ceci

Figure 2.5 – *Notre perception des scènes en tant qu'entités globales n'est qu'une reconstruction faite par notre cerveau a posteriori. Il est en effet très difficile de percevoir des différences entre deux images similaires, dès lors qu'un élément extérieur vient perturber notre attention, par exemple le fait de devoir tourner la page (voir figure 2.6). (Image tirée de* `http://nivea.psycho.univ-paris5.fr/`*)*

près qu'elle illustre le fait qu'il est possible de rendre non perceptible un évènement saillant en demandant explicitement à un sujet de porter son attention à un autre endroit, ou sur une autre tâche. Ce phénomène est largement utilisé par les prestidigitateurs dont le métier consiste à détourner l'attention des spectateurs vers un endroit éloigné de là où le "truc" se passe...

Nous reviendrons sur ces phénomènes attentionnels dans le chapitre 7.

2.2 L'action

"To be is to do"—Socrates.
"To do is to be"—Jean-Paul Sartre.
"Do be do be do"—Frank Sinatra.

Kurt Vonnegut - Deadeye Dick, 1982

Lorsque l'on parle de l'action, deux sous-problèmes peuvent être considérés. Le premier est celui de savoir et pouvoir agir dans un but donné : comment s'utilise une tasse, une chaise, une souris d'ordinateur ? Le second est celui d'optimiser une action déjà maîtrisée par ailleurs du point de vue précédent : il est "facile" de savoir utiliser un clavier d'ordinateur, moins de l'utiliser à une vitesse de frappe élevée utilisant les dix doigts.

Le premier problème consiste à découvrir et maîtriser les différentes possibilités d'action offertes par l'environnement dans lequel on évolue, alors que le second optimise une fonction de coût explicite que se fixe l'agent de manière consciente (vitesse de frappe au clavier, distance parcourue par un javelot, précision de tir sur une cible, etc.) et présuppose en particulier que ces actions soient déjà en partie connues, notamment car la définition de la fonction de coût à optimiser nécessite de connaître les effets de l'action.

Figure 2.6 – *Quelle est la différence avec la figure 2.5 ? (Image tirée de* `http://nivea.psycho.univ-paris5.fr/`*)*

Dans cette thèse en général et dans ce chapitre en particulier, nous nous focaliserons sur ce premier problème. L'optimisation d'une fonction de coût sera brièvement abordée au chapitre 6.

Savoir comment agir dans un environnement nécessite en particulier un couplage fin entre action et perception, comme nous allons le voir dans les prochains paragraphes.

2.2.1 Agir pour percevoir

Une partie de l'action est directement dirigée vers la perception : je suis des yeux un objet en mouvement en tournant éventuellement la tête, je me tourne vers une source de bruit, je tends la main pour toucher un objet et en déterminer la matière, etc.

Ce type d'action ne vise pas à interagir directement avec le monde, mais plutôt à obtenir des informations qui permettent de mettre à jour et d'affiner un modèle interne du monde : quelle est la trajectoire suivie par un objet, quelle est la source de bruit, comment un objet va-t-il se comporter si je tape dedans ? Il s'agit d'une des facettes des boucles action-perception la plus communément admise (Brooks 1986) : l'action modifie la perception, qui permet de mettre à jour un modèle interne de l'environnement (éventuellement réduit à la perception présente dans le cas des robots réactifs), qui permet à son tour de décider de la prochaine action.

Mais le rôle de l'action pour la perception est bien plus fondamental que cela. En 1963, Held et Hein mènent une expérience sur des chatons (Held et Hein 1963). Le principe de l'expérience est le suivant. Deux chatons sont élevés dans des pièces entièrement plongées dans l'obscurité, de sorte qu'ils ne puissent utiliser aucune information visuelle. Une heure par jour cependant, ils sont placés dans un manège constitué de deux paniers tournant autour d'un axe central. Le premier chaton est placé dans un panier duquel ses pattes sortent pour être en contact avec le sol (il peut donc se déplacer librement), le deuxième chaton est placé dans un panier sans aucune liberté de mouvement (ses pattes ne sortent pas du panier). Ce panier est toutefois relié au premier de façon à en reproduire les mouvements

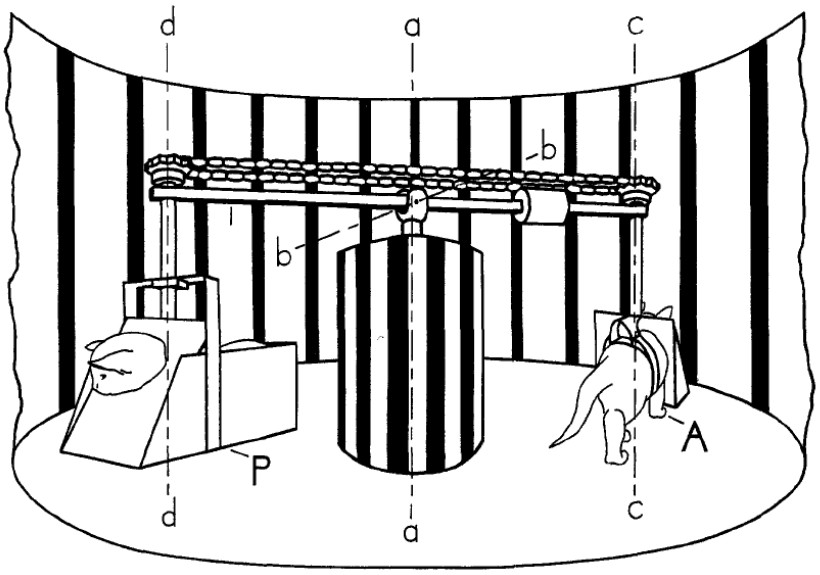


FIG. 1. Apparatus for equating motion and consequent visual feedback for an actively moving (A) and a passively moved (P) *S*.

Figure 2.7 – *Dans leur expérience, Held et Hein (*HELD *et* HEIN *1963) élèvent des chatons dans des conditions différentes : l'un peut bouger librement, et donc percevoir les effets de ses actions, tandis qu'un autre est privé de sa liberté de mouvement, et subit passivement les déplacements du premier chaton, reproduits à l'aide d'un "manège". Après huit semaines, le premier chaton est capable de se déplacer normalement dans son environnement, alors que le second montre de graves déficits dans la perception des distances et de la profondeur. (Image tirée de (*HELD *et* HEIN *1963))*

(voir figure 2.7). La pièce dans laquelle ils se trouvent est alors éclairée. Le premier chaton peut se déplacer librement dans son environnement, en utilisant les informations visuelles qu'il perçoit, tandis que le deuxième chaton reçoit les mêmes informations visuelles que le premier, sans en être à l'origine et sans pouvoir les utiliser pour se déplacer. Les chatons sont ainsi élevés dès leur naissance et pendant huit semaines. Au bout de ces huit semaines, les chatons sont placés dans une pièce éclairée et laissés totalement libres de leurs mouvements. Le premier chaton a alors un comportement indiscernable de celui de chatons élevés normalement, alors que le second chaton présente une incapacité à percevoir les distances et la profondeur. Cette expérience illustre de manière frappante l'intrication entre action et perception dans le développement des capacités perceptives d'un agent : la perception s'appuierait sur l'apprentissage de *contingences sensorimotrices* (O'REGAN et NOË 2001).

Cependant, l'importance de l'action dans la perception ne s'arrête pas après une phase de développement, mais les deux restent profondément intriquées tout au long de la vie : la perception est toujours liée à l'action, même lorsqu'aucun apprentissage n'est en jeu. L'illusion d'Akiyoshi Kitaoka offre une illustration de cette influence. Il s'agit d'une simple image de damier, dont les cases noires centrales ont été dotées de petits carrés blancs disposés le long de leurs bords (figure 2.8). Lorsque l'on regarde cette image normalement, en laissant ses yeux la parcourir, on perçoit un effet de relief, soit de creux soit de bosse

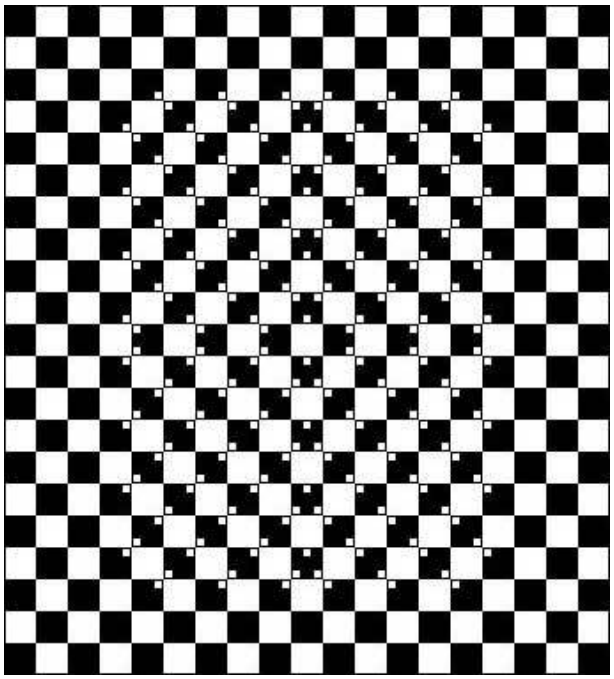


Figure 2.8 – *Illusion d'Akiyoshi Kitaoka. À première vue, les lignes du damier ne sont pas parallèles. Cependant, en fixant le regard sur une seule case, les lignes sont perçues telles qu'elles sont en réalité, parallèles. Cette illusion illustre le couplage intime entre action et perception : le simple fait de bouger les yeux modifie la perception que l'on a du stimulus.*

selon les personnes, provoqué par une perception des lignes du damier non parallèles. Mais si l'on fixe maintenant une case précise en figeant son regard, les lignes redeviennent parallèles. Cette illusion illustre l'importance du rôle des mouvements des yeux dans la perception d'un même stimuli visuel.

2.2.2 Percevoir pour agir

Nous avons discuté dans la partie précédente du rôle de l'action dans les capacités perceptives d'un agent. Le lien réciproque existe lui aussi : on agit en fonction de ce que l'on perçoit.

Certains mouvements sont des réflexes déclenchés par des stimuli précis (Schott et Rossor 2003) : le réflexe de préhension chez le nourrisson lors d'une stimulation tactile de la paume de la main, le mouvement de recul et la fermeture des paupières lorsque l'on perçoit un flash lumineux ou un flux optique divergent (qui correspond à un obstacle qui se rapproche), etc. Ces mouvements sont réactifs : la perception d'un stimulus donné déclenche de façon quasi-automatique le mouvement associé. Mais il s'agit là d'un nombre très restreint de mouvements, dont beaucoup disparaissent au cours du développement (Schott et Rossor 2003). Pour la plupart, nos mouvements au quotidien résultent d'une planification pour atteindre un but : je veux boire de l'eau, je vais donc saisir la bouteille fermement de la main gauche, de manière à pouvoir ouvrir le bouchon

de la main droite, avant de porter le goulot à ma bouche. Au contraire, si j'avais voulu donner la bouteille d'eau à mon voisin de droite, je l'aurais plus certainement saisie de la main droite. Mais le but poursuivi n'est pas le seul à influer sur l'action effectuée. Ainsi, si je devais saisir un verre, je ne le saisirais probablement pas avec la même force selon qu'il s'agit d'un verre doté d'un épais fond en verre ou qu'il s'agit d'un gobelet en plastique. Dans ce cas, bien que le but soit le même, l'action effectuée dépend des caractéristiques perçues.

Ce principe est mis en exergue dans l'illusion taille-poids (Van Biervliet et al. 1895). Dans cette illusion, plusieurs objets de tailles et de formes différentes, mais pesant exactement le même poids, sont présentés à des sujets. Ceux-ci peuvent les manipuler et doivent indiquer entre deux objets lequel est le plus lourd. La majorité des sujets (49 sur 50 rapportés dans (Van Biervliet et al. 1895)) indique que les objets les plus petits sont les plus lourds. Cependant, lorsque l'expérience est recommencée yeux bandés, les sujets sont capables de correctement identifier que les poids sont égaux. La même illusion peut être mise en évidence en jouant non sur la taille des objets, mais sur leur matière (par exemple métal/bois (Seashore 1899)). Lorsque cette illusion est mise en œuvre, on observe en outre que l'action de saisie des objets diffère selon la taille : l'accélération et la hauteur à laquelle l'objet est soulevé sont notamment plus élevées pour le plus grand des objets de même poids (Davis et Roberts 1976). Ceci montre que la perception a bien fourni des "paramètres" pour l'action en s'appuyant sur des connaissances *a priori* (plus un objet est grand, plus il est lourd). Il a été montré que ces connaissances *a priori* résultent bien d'un apprentissage : en entraînant les sujets sur un ensemble d'objets ayant la propriété inverse (les plus petits pèsent plus lourd que les plus grands), l'effet est inversé (Flanagan et al. 2008). De plus, deux sous-systèmes semblent jouer un rôle lors de la saisie d'objet. Le premier s'appuie principalement sur la perception et sur des connaissances apprises *a priori* pour estimer la force nécessaire pour soulever un nouvel objet. Ce système semble prépondérant dans l'illusion taille-poids. En effet, un second système adapte correctement la force exercée sur un objet pour le saisir et le soulever au bout de quelques essais seulement, alors même que l'illusion persiste (Flanagan et Beltzner 2000). Ces deux systèmes ne sont toutefois pas indépendants, puisque les sujets ne sont pas capables d'adapter la force de saisie (ou l'adaptent beaucoup plus lentement) lorsqu'ils portent des lunettes qui masquent l'objet au moment de la saisie, ce qui permet à l'expérimentateur d'échanger l'objet vu avec l'objet saisi, de sorte que les sujets saisissent toujours le même objet (et devraient donc rapidement adapter la force de saisie), même s'ils en voient des différents (Buckingham et Goodale 2010).

Au cours de notre vie, nous apprenons un répertoire de plus en plus riche et complexe d'actions qui nous permettent d'interagir avec notre environnement. Pour chaque action, nous apprenons les situations dans lesquelles elles sont pertinentes, leurs effets et les buts qu'elles permettent d'atteindre, ainsi qu'une paramétrisation fine selon le contexte (par exemple la force à appliquer pour saisir un verre). Petit à petit, le monde est perçu en termes d'actions possibles. Il est parfois difficile de s'empêcher de boire le verre placé devant soi lorsque l'on est assis à la table d'un restaurant. Des études ont montré que ce phénomène pouvait s'expliquer par une défaillance momentanée du cortex moteur supplémentaire (CMS) situé en avant du cortex moteur dans le lobe frontal du cerveau. La

vue d'un objet déclenche en effet de la part du lobe pariétal la proposition d'actions stéréotypées associées à cet objet, qui sont envoyées vers le cortex moteur (Lachaux 2011, p.180-182). Pour l'atteindre, il faut cependant passer la porte du CMS, qui filtre parmi toutes les actions proposées celles qui doivent réellement être exécutées (Nachev et al. 2008). Certains patients atteints d'une lésion du CMS souffrent de syndromes étranges, comme le syndrome du membre étranger (Feinberg et al. 1992). Ces patients peuvent par exemple avoir la sensation de ne plus maîtriser les actions de leur bras et en viennent à le considérer comme un être autonome. Ils voient, impuissants, leur bras saisir un stylo ou un verre posé devant eux, même s'ils n'ont rien à écrire ou s'ils n'ont pas soif. Pire encore, ils peuvent assister impuissants à leur bras déboutonnant leur chemise, qu'ils sont par ailleurs en train de boutonner de leur autre bras. Ces patients illustrent la manière dont le cerveau perçoit en permanence l'environnement en terme d'affordances : dès que je vois un verre, je le perçois en terme d'objet servant à boire, et je prépare donc l'action de saisie. Seul un contrôle *a posteriori* par le CMS me permet de garder le contrôle sur mon corps en filtrant les actions que je vais réellement effectuer (Sumner et al. 2007) (des sous-ensembles distincts de neurones du CMS semblent au contraire indispensables pour initier une action (Nachev et al. 2008)). D'autres neurones du CMS sont quant à eux impliqués dans l'apprentissage de séquences d'actions, plus particulièrement au niveau des associations stimulus-réponse qui servent de base à l'enchaînement de plusieurs sous-actions déclenchées par des stimuli spécifiques (Sakai et al. 1999).

Le CMS contient aussi chez l'homme des *neurones miroirs* (Keysers et Gazzola 2010) (ce qui n'exclut pas leur présence dans d'autres aires cérébrales, comme chez le singe). Ces neurones ont la propriété de répondre aussi bien lors de l'exécution d'une action spécifique (par exemple saisir un objet), que lors de la perception de la même action effectuée par un autre agent (Rizzolatti et Craighero 2004). En particulier, ces neurones semblent mieux répondre pour des actions dirigées vers un but (même si celles-ci sont mimées) (Kilner et al. 2009). De nombreuses controverses existent sur le rôle et la fonction de ces neurones. Nous retiendrons uniquement que la fonction de ces neurones semble être compatible avec une représentation symbolique des actions, de la même manière que les neurones *grand-mère* fournissent une représentation de très haut niveau de stimuli statiques.

Chez le singe, d'autres neurones situés dans le cortex préfontal latéral ont également une activité compatible avec une représentation symbolique de séquences d'actions (Shima et al. 2006). Dans cette expériences, des singes doivent apprendre à reproduire des séquences motrices de différentes catégories. Une première catégorie contient par exemple les séquences "Tirer-Tirer-Tourner-Tourner" et "Pousser-Pousser-Tirer-Tirer" lorsqu'une autre catégorie contient les séquences "Tirer-Tourner-Tirer-Tourner" et "Pousser-Tirer-Pousser-Tirer". En enregistrant l'activité de neurones du cortex préfontal latéral, les auteurs de (Shima et al. 2006) ont montré que certains neurones répondaient de manière spécifique à une catégorie, quelle que soit la séquence réalisée au sein de cette catégorie, lorsque d'autres neurones codent la séquence spécifiquement réalisée.

L'action dépend également de la manière dont on perçoit son corps. En effet, pour pouvoir saisir correctement un objet, il faut tout d'abord savoir le situer par rapport à soi. Chez le singe, des neurones du cortex pariétal postérieur codent la position des objets

dans différents repères (Avillac et al. 2005) : rétinotopique (aire intrapariétale latérale, LIP), crâniotopique et oculocentrique (aire intrapariétale ventrale, VIP). Les neurones de LIP répondent en particulier aux stimuli saillants, que leur saillance soit intrinsèque (par exemple apparition abrupte) ou comportementale (pertinence pour le comportement de l'agent) (Goldberg et al. 2006). Les neurones de l'aire intrapariétale antiétieure (AIP) répondent quant à eux à la forme, à la taille et à l'orientation des objets perçus afin de préparer le geste correspondant (Murata et al. 2000). Lors d'un geste de saisie, le lobe pariétal a donc deux missions : les neurones AIP préparent le geste de saisie adéquat (on ne saisit pas une tasse et un tournevis de la même manière), tandis que les neurones LIP et VIP déterminent sa position et calculent donc le mouvement nécessaire pour l'atteindre. Ces deux informations convergent au niveau du cortex moteur qui réalise l'action après que celle-ci a passé le filtre du CMS (Lachaux 2011, p.179). Il a été montré que les neurones AIP codent les actions stéréotypées pour différentes familles d'objets. Ainsi, la stimulation électrique de certains neurones de cette zone déclenche chez l'animal des mouvements de saisie d'objets imaginaires qui varient en fonction des neurones stimulés (Lachaux 2011). Chez l'homme, des lésions au niveau de cette région ont des conséquences dramatiques pour les individus concernés, qui perdent la capacité à utiliser correctement les objets (apraxie idéatoire) (Grafton 2003). Ainsi, devant une bougie et une boîte d'allumette, un individu pourra prendre la bougie et la frotter sur la boîte pour tenter de l'allumer.

2.3 Unification de l'action et de la perception : modèles cognitifs

The best material model of a cat is another, or preferably the same, cat.

Norbert Wiener - Philosophy of Science, 1945

Les parties précédentes illustrent les liens très forts entre perception et action et leur interdépendance, bien plus forte que les simples boucles perception/action proposées aux débuts de la robotique autonome, dans lesquelles l'action était décidée à partir d'un modèle interne de l'environnement, mis à jour à chaque pas de temps par la perception.

Différents modèles cognitifs ont depuis été développés pour rendre compte de ces influences réciproques dans le développement des capacités cognitives d'un agent. Nous en présenterons trois : le modèles des contigences sensorimotrices de O'Regan et Noë (O'Regan et Noë 2001), le principe de minimisation de l'énergie libre par Friston (Friston 2010) et les zones de Convergence-Divergence de Damasio (Meyer et Damasio 2009).

2.3.1 Les contingences sensorimotrices

Le modèle des contingences sensorimotrices postule que la perception résulte de la possibilité de prédire les conséquences de ses actions en terme de stimuli reçus. Ainsi, on perçoit une ligne droite à travers la prédiction qu'un mouvement des yeux dans la direction tangente à celle du segment actuellement perçu au niveau de la fovéa ne modifiera pas le

stimulus perçu. De même, la couleur rouge est perçue à travers la modification d'activité des cônes de la rétine qui sont induits (et prédits) par un mouvement du regard :

> *Visual experience does not arise because an internal representation of the world is activated in some brain area. On the contrary, visual experience is a mode of activity involving practical knowledge about currently possible behaviors and associated sensory consequences.*
>
> O'Regan et Noë, 2001 (O'Regan et Noë 2001)

Cette théorie s'oppose de manière frontale à la vision cartésienne de la perception, qui ne serait qu'une *re-présentation* du monde extérieur. Au contraire, d'après (O'Regan et Noë 2001), la perception découle de la connaissance de la possibilité de produire certaines sensations prédictibles grâce aux contingences sensorimotrices.

Selon cette théorie, toute perception est subordonnée à la possibilité d'agir dans son environnement. Il faut cependant noter que cette possibilité d'agir n'est strictement requise que lors de l'apprentissage (en ligne avec l'expérience du *manège à chatons* de Held et Hein). Une fois apprise, la perception passe avant tout par les capacités de prédiction (on associe un stimulus donné à un ensemble de contingences sensorimotrices apprises lors de précédentes rencontres avec des stimuli similaires, ce qui permet donc de percevoir ce nouveau stimulus), et ne requiert donc pas de mouvement. Ces mouvements ne sont alors strictement requis que pour lever l'ambiguïté entre plusieurs perceptions possibles. Il est toutefois possible que certaines combinaisons de stimuli et de mouvements activent faussement certaines contingences, comme dans le cas de l'illusion d'Akiyoshi Kitaoka.

Une prédiction de cette théorie est la cécité au changement que nous avons présentée dans la première partie de ce chapitre : puisque la perception ne s'appuie pas sur une représentation interne du monde mais simplement sur la capacité de prédire les sensations provoquées par certaines actions, il est très difficile de percevoir des modifications entre deux images similaires tant que l'on ne porte pas son attention sur l'élément modifié.

Cette théorie des contingences sensorimotrices peut être mise en parallèle avec celle des affordances, qui postule dans sa version forte que l'on perçoit un objet à travers les effets prédits des actions que l'on peut faire avec. Dans ce cas, les contingences sensorimotrices étendent la théorie des affordances au plus bas niveau de la perception, en définissant chaque stimulus à travers les contingences sensorimotrices qui lui sont associées.

2.3.2 Zones de Convergence-Divergence

La théorie des contingences sensorimotrices implique l'existence de zones du cerveau recevant en entrée à la fois des informations perceptives et des informations motrices afin de pouvoir extraire leurs corrélations.

Comme nous l'avons vu, l'existence de telles zones multimodales dans le cerveau a été démontrée et a servi de base à la théorie des zones de convergence-divergence (Meyer et Damasio 2009). Ces zones consistent en des groupes neuronaux recevant en entrée des informations de différentes natures : information perceptive / information motrice, ou encore information visuelle / information auditive. De telles structures multimodales sont donc de bonnes candidates non seulement comme support biologique de la théorie des

contingences sensorimotrices, mais semblent également nécessaires pour rendre compte de certaines illusions perceptives comme l'effet McGurk.

Un des éléments importants de cette théorie est que les représentations cognitives haut niveau ne sont pas de simples copies des stimuli perçus, mais seulement une représentation minimale des données nécessaires pour être capable de reproduire les motifs d'activité neuronale liés à la perception d'un stimulus dans les aires sensorielles primaires.

De telles zones constituent également de bons candidats pour expliquer le rôle et le fonctionnement des neurones miroirs, qui s'activent aussi bien lors de l'exécution d'une action que lors de la visualisation de cette action exécutée par un autre agent, ou des neurones de l'aire AIP par exemple qui associent des informations visuelles (taille et forme d'un objet) avec les mouvements associés à sa préhension.

Selon cette théorie, le cerveau contient un grand nombre de zones de convergence-divergence qui fusionnent des informations de manière hiérarchique. Cette hiérarchie peut s'étendre du plus bas niveau des contingences sensorimotrices à l'émergence de neurones ayant une très grande invariance, comme dans l'exemple des neurones spécifiques à "Halle Berry".

En cela, elles constituent un mécanisme générique intéressant dont s'inspirent naturellement de nombreux algorithmes d'intelligence artificielle et de robotique développementale (Wermter et al. 2004 ; Papliński et Gustafsson 2005 ; Johnsson et al. 2009 ; Ridge et al. 2010 ; Vavrečka et Farkaš 2013 ; Lallee et Dominey 2013) que nous discuterons plus en détail au chapitre 4.

La possibilité d'utiliser les zones de Convergence-Divergence comme des "pivots" susceptibles de reproduire une activité neuronale "imaginée" dans une aire sensorielle à partir de l'activité dans une autre aire permet de faire des liens avec la théorie des contingences sensorimotrices : la perception d'un stimulus visuel est ainsi susceptible d'activer une commande motrice associée pouvant à son tour activer une représentation des transformations qu'elle impliquerait au niveau sensoriel. Ce mécanisme génère alors la prédiction au cœur de la théorie des contingences sensorimotrices.

2.3.3 Principe du minimum d'énergie libre

La dernière théorie que nous décrirons, le principe de minimisation de l'énergie libre (Friston 2010), est un modèle théorique s'inspirant des outils de la physique statistique (Dayan et al. 1995).

Pour cette théorie, chaque agent est un système auto-organisé en équilibre avec son environnement. Cet état d'équilibre implique qu'à chaque instant l'agent s'adapte pour contrer la tendance naturelle au désordre (découlant du principe thermodynamique de maximisation de l'entropie). En particulier, pour un organisme vivant, un très petit nombre d'états parmi tous ceux possibles sont viables. L'agent cherche donc à maximiser la probabilité de se trouver dans ces états viables.

Cette théorie montre que cela revient à minimiser une fonction, appelée énergie libre, dépendant de l'état interne de l'agent. Cet état interne de l'agent lui fournit en particulier un modèle interne de l'environnement qui lui permet de prédire ses sensations (définies comme un échantillonnage des capteurs de l'agent au niveau de l'interface entre

lui et l'environnement). La perception correspond alors au changement de l'état interne de l'agent afin de modifier son modèle interne de manière à mieux prédire ses sensations, alors que l'action vise à modifier la dynamique de l'environnement afin de placer l'agent dans les zones où son modèle interne est le plus précis.

La minimisation de l'énergie libre permet donc de dériver le cadre théorique du codage prédictif dont nous avons parlé en introduction, en liant de plus étroitement action et perception. D'une part, la perception implique que l'agent infère de manière implicite les causes de ses sensations, c'est-à-dire des variables cachées de l'environnement, ce qui fournit un modèle suffisamment précis pour que l'action modifie l'état de l'environnement et la configuration des capteurs de l'agent de façon à ce que les sensations suivantes de l'agent soient elles aussi prédictibles. L'exemple donné dans (Friston 2010) de cette boucle considère le cas où nous cherchons notre chemin dans le noir. Notre perception (majoritairement tactile) adapte notre modèle interne de l'environnement en terme de meubles présents et de leur position relative ce qui nous permet d'anticiper ce que nous devrions toucher à tel endroit, ce que nous tentons de confirmer en tendant effectivement le bras dans cette direction.

2.4 L'hypothèse des sous-variétés

Les mathématiques sont la seule science où on ne sait pas de quoi on parle
ni si ce qu'on dit est vrai.

Bertrand Russell

Les modèles précédents développés dans un cadre théorique fournissent des explications intéressantes à certains phénomènes et ont permis d'en prédire d'autres effectivement mis en évidence. Cependant, lorsqu'il s'est agi de s'en inspirer pour développer des algorithmes utilisables notamment pour la robotique développementale, ils ont donné lieu à des algorithmes limités à des problèmes simples, généralement de faible dimensionalité.

En particulier, il est difficile à partir de ces modèles seuls de franchir le fossé du symbolique, c'est-à-dire le passage d'un flux d'informations brutes en provenance des capteurs à une représentation sémantique de l'information, par exemple sous forme de symboles manipulables par des mécanismes de raisonnement et de planification.

C'est pourquoi nous introduisons dans cette partie une hypothèse centrale de notre travail, l'hypothèse des sous-variétés. Cette dernière donne une définition mathématique des concepts que l'on peut implémenter de manière algorithmique et que nous développerons dans la suite de notre travail.

2.4.1 Des données dans des espaces de grande dimensionalité

Avant de décrire plus avant l'hypothèse des sous-variétés, il est important de discuter quelques propriétés des espaces de grande dimensionalité. En effet, certaines intuitions tirées de raisonnements en deux ou trois dimensions se révèlent complètement fausses lorsque le nombre de dimensions augmente. De nouveaux problèmes se posent alors pour

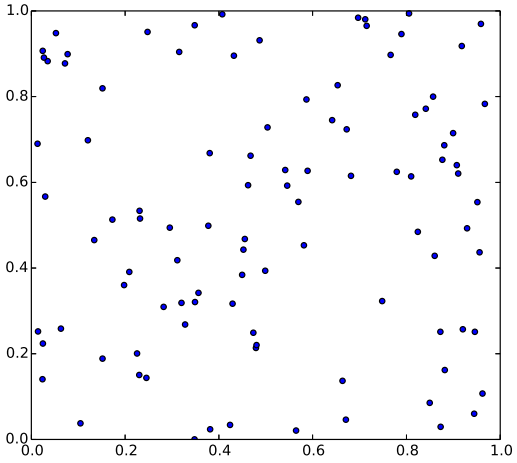


Figure 2.9 – *100 points ont été tirés aléatoirement dans le carré de côté 1. La surface du carré semble correctement couverte et de manière évidente les distances entre deux points varient grandement selon les couples de points considérés. Cette propriété intuitive n'est plus vraie lorsque le nombre de dimensions augmente.*

les algorithmes. Il s'agit de la malédiction de la dimensionalité, terme introduit par Richard Bellman (Bellman 1957).

Des espaces vides...

Prenons comme exemple un hypercube de côté 1 dans un espace en d dimensions. Un tel hypercube possède 2^d sommets. Le nombre de points nécessaires ne serait-ce que pour paver cet hypercube de manière à séparer chaque dimension en deux parties croît donc exponentiellement avec d. Ainsi, pour $d = 100$, il faudrait déjà plus de 10^{30} points. Manipuler de telles quantités de données est aujourd'hui inimaginable. Il s'ensuit que dans de tels espaces, la probabilité de trouver un point dans un quadrant choisi au hasard est quasi nulle.

Considérons maintenant n points $(p_1, \cdots, p_n)$ tirés aléatoirement selon une loi uniforme dans ce cube. La figure 2.9 montre ce qu'il se passe pour $d = 2$, avec 100 points. Il paraît évident que la surface du carré est correctement couverte et que les points sont plus proches de certains autres points que d'autres. Cette intuition se révèle complètement fausse quand d augmente. En effet, le rapport des distances entre les deux points les plus proches et les deux points les plus éloignés tend vers 1 quand d tend vers l'infini, ou plus rigoureusement (Beyer et al. 1999) :

$$\forall \epsilon > 0, \lim_{d \to +\infty} \mathbb{P}\left[max_{i \neq j} ||p_i - p_j|| \leq (1+\epsilon) min_{i \neq j} ||p_i - p_j||\right] = 1. \tag{2.1}$$

En particulier pour des distributions uniformes, avec $d = 100$ et un million de points, (Beyer et al. 1999) ont montré en simulation que le rapport entre les distances

maximale et minimale était inférieur à 2 en moyenne. Dans des espaces de grande dimensionalité, la notion de plus proche voisin peut donc perdre de son sens puisque les distances entre deux points ont tendance à devenir toutes "égales". La prise en compte de contraintes liées à l'environnement permettra à l'hypothèse des sous-variétés d'échapper à ce problème.

. . . où tout se joue sur les bords

Considérons toujours n points à l'intérieur d'un hypercube. Nous avons vu qu'il faut un très grand nombre de points pour remplir l'espace. Toutefois, de manière assez contre-intuitive, même quand ce volume est rempli la plupart des points se situent sur les bords de l'hypercube. Pour le voir, considérons le rapport r entre le volume d'un hypercube de côté $1-\epsilon$ et celui d'un hypercube de côté 1 :

$$r = \frac{(1-\epsilon)^d}{1^d} = (1-\epsilon)^d. \tag{2.2}$$

Ainsi,

$$\forall \epsilon \in]0,1], \lim_{d \to +\infty} r = 0. \tag{2.3}$$

Tout le volume d'un hypercube est donc concentré sur ses bords. Par exemple avec $d = 300$ et $\epsilon = 0.02$, on obtient $r \approx 0.002$, c'est-à-dire que 99.8% du volume de l'hypercube est concentré sur une bordure qui dont l'épaisseur représente 1% de la longueur d'un côté.

On peut tirer de cette observation une remarque importante : dans de tels espaces, la majorité des données peut être approchée de manière satisfaisante dans un système de coordonnées de plus faible dimensionalité que l'espace d'origine. Dans l'exemple précédent, il suffit de décrire les points par leur projection sur la frontière de l'hypercube (c'est-à-dire dans un système de $d-1$ coordonnées) pour décrire 99.8% des points (répétons-le, tirés aléatoirement et uniformément répartis dans l'hypercube) avec une erreur de moins de 1%. On voit donc ici que dans des espaces de grande dimensionalité, les points peuvent être décrits grâce à des variétés de plus faible dimensionalité.

2.4.2 Des régularités de l'environnement

Les exemples précédents partent d'un tirage aléatoire uniforme des points dans l'espace. Cependant, comme nous allons le voir, les régularités de l'environnement imposent des contraintes supplémentaires sur la répartition de ces points qui permettent d'envisager des propriétés particulières.

Des variétés naturelles

Dans la partie précédente, nous avons vu que des points dans des espaces de grandes dimensionalité peuvent être efficacement décrits le long de variétés de plus faible dimensionalité, alors même que la notion de plus proche voisin n'a que peu de sens dans ces espaces. Ce qui est vrai pour ces points aléatoires l'est encore plus pour des points représentant des stimuli naturels.

En effet, les degrés de liberté de l'environnement et d'un agent contraignent les variations possibles de la perception de l'agent. Pour illustrer ce propos, prenons l'exemple d'une personne avec qui l'on parle. On voit d'une part son visage bouger, on entend d'autre part les mots qu'elle prononce. Ces stimuli créent une activité riche au niveau des récepteurs rétiniens et de l'oreille interne, qui se propage aux cortex visuels et auditifs, qui contiennent des millions de neurones. Pour autant, l'apparence du visage à un instant t est décrite par la contraction des différents muscles du visage (une cinquantaine) et à sa pose dans l'espace (6 paramètres). De même, les sons entendus peuvent être décrits par les contractions des muscles du système articulatoire et de quelques paramètres décrivant la morphologie des cavités impliquées. En supposant que l'activité des neurones des cortex sensoriels décrit fidèlement les stimuli présentés en entrée (ceci étant toutefois peu probable, notamment à cause de la présence d'interactions descendantes que nous avons précédemment évoquées), l'activité de ces quelques millions de neurones peut alors être décrite quasi entièrement par les valeurs de quelques dizaines, ou quelques centaines, de variables.

De même, la morphologie d'un visage répond à certains critères, par exemple la présence d'un nez entre les deux yeux et la bouche, une taille (et le plus souvent une couleur) similaire pour les deux yeux, etc. Ainsi, les contraintes liées aux degrés de liberté de l'environnement font qu'un stimulus perçu est solution d'un système sous contraintes, ce qui permet de le décrire par un faible nombre de variables, même si sa perception modifie l'activité de plusieurs millions de neurones. On se trouve donc dans le cas de variétés naturelles de beaucoup plus faible dimensionalité que l'espace de départ (figure 2.10).

Si la partie précédente a illustré le fait qu'il est pertinent de vouloir représenter les données dans des espaces de grande dimensionalité à l'aide de sous-variétés dans ces espaces, cette partie permet d'avancer l'hypothèse que les contraintes naturelles imposées par l'environnement impliquent une grande densité de points au voisinage de variétés de très faible dimensionalité par rapport à l'espace de départ. En effet, là où des données aléatoires rendent la notion de voisin peu pertinente, la description d'un stimulus par quelques dizaines de variables permet quant à elle d'envisager une relation de plus proche voisin bien définie pour des stimuli naturels. Cela suppose toutefois que ces stimuli soient traités par des mécanismes suffisamment réguliers pour en conserver la topologie.

Dans la section suivante, nous présentons une hypothèse qui développe plus avant la notion de voisinage le long de ces variétés, en faisant un critère de base pour la définition de concepts.

2.4.3 Une hypothèse forte : le rôle des sous-variétés

Nous l'avons vu, il paraît naturel de décrire les stimuli par des variétés dans un espace perceptif de grande dimensionalité. Deux stimuli peuvent alors être différenciés selon leur distance le long de cette variété. Il est dès lors tentant de définir deux stimuli proches comme appartenant à une même famille, ou une même catégorie, qui sont les briques de base de la notion de concept : des visages, des chaises, des chiens, etc. L'hypothèse des sous-variétés postule l'existence de sous-variétés distinctes sur lesquelles se projettent un grand nombre de stimuli et séparées les unes des autres par des zones vides de l'espace.

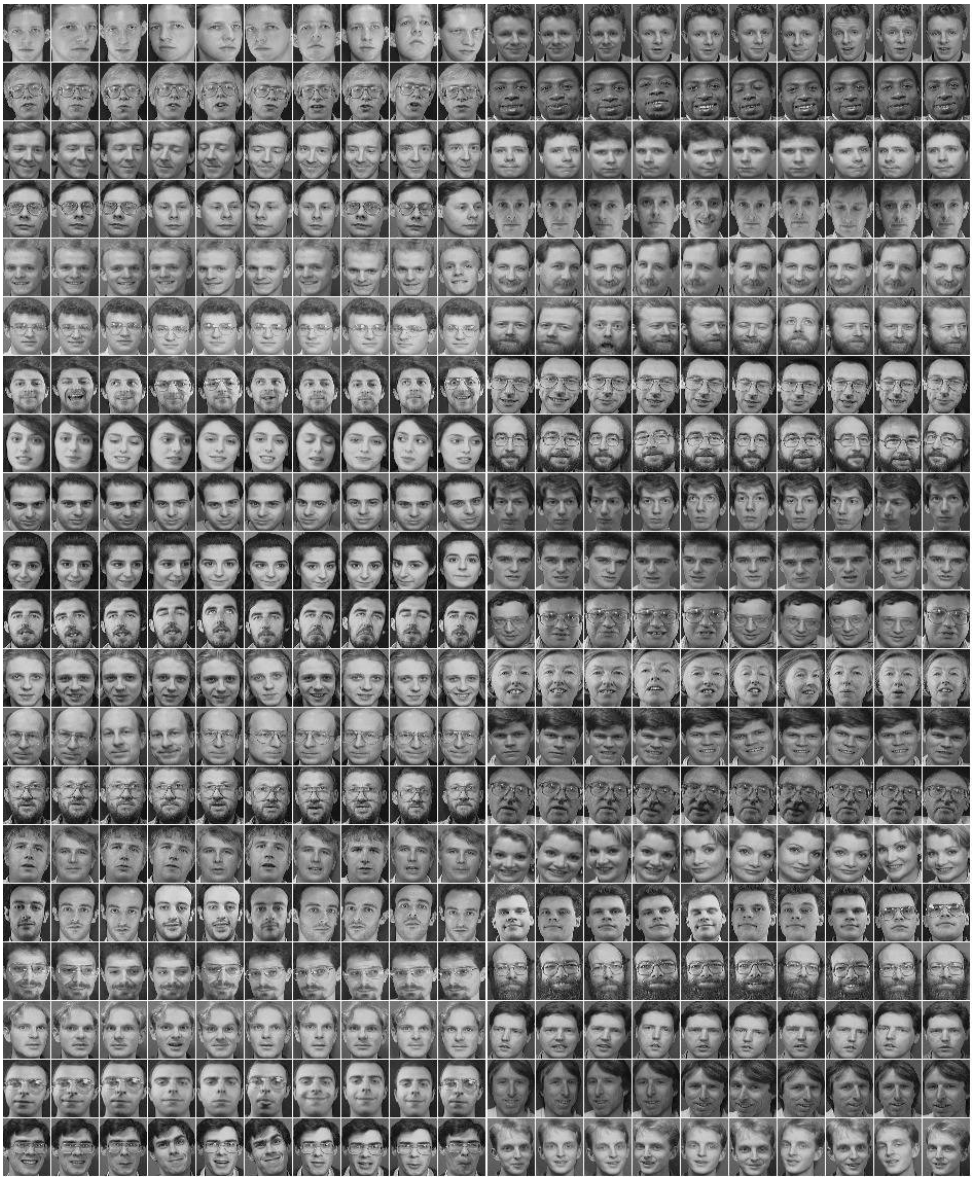


Figure 2.10 – *La vue de ces visages se traduit par l'activité de millions de neurones. Cependant, l'apparence de chaque visage peut être décrite par un faible nombre de variables, ce qui permet de le placer, ainsi que l'activité neuronale associée, le long d'une sous-variété perceptive. Différents niveaux de granularité peuvent être envisagé : sous-variété pour un visage précis (pour les proches par exemple), et sous-variété des visages en général. (Images de la base de données Olivetti research)*

Ces sous-variétés peuvent alors être identifiées aux différentes catégories de stimuli. Cette formulation de l'hypothèse des sous-variétés est celle présentée dans (Cayton 2005 ; Narayanan et Mitter 2010 ; Rifai et al. 2011a).

Nous voudrions en proposer une légère variante, qui s'appuie toujours sur l'existence de sous-variétés distinctes dans l'espace, mais en relâchant le critère de séparation de ces sous-variétés dans l'espace. Nous proposons au contraire que ces sous-variétés puissent s'intersecter. Pour deux stimuli appartenant à deux sous-variétés distinctes en dehors de toute intersection, le cas est similaire à l'hypothèse usuelle des sous-variétés (chacun des deux stimuli appartient à une catégorie différente). Dans le cas d'un stimulus à l'intersection de deux sous-variétés, son interprétation est instable et dépend de la sous-variété choisie pour le décrire. Ce choix de la sous-variété peut être arbitraire (tirage aléatoire) ou conditionné (la présentation passée d'autres stimuli appartenant uniquement à l'une des deux sous-variétés peut par exemple influencer la perception d'un stimulus à l'intersection en biaisant son interprétation en faveur de la même sous-variété). Ceci permet d'introduire un phénomène attentionnel pouvant modifier la perception d'un même stimulus. Un phénomène attentionnel similaire peut être invoqué pour traiter le cas de stimuli qui ne seraient pas situés exactement sur une sous-variété apprise : la manière de les projeter sur une sous-variété proche, et donc leur interprétation, pourrait dépendre d'un mécanisme externe, par exemple d'une interaction descendante. Notons aussi qu'ajouter la possibilité d'intersection de ces sous-variétés permet d'apporter un élément de réflexion pour la hiérarchie des concepts, en transformant la hiérarchie en relation d'inclusion (figure 2.11). En effet, s'il est possible d'avoir des sous-variétés incluses dans d'autres sous-variétés plus grandes, l'interprétation d'un même stimulus peut alors se faire selon plusieurs sous-variétés (par exemple *animal*, *mammifère*, *chien*, *Médor*, ou encore dans le cas des visages en général ou du visage d'une personne en particulier), là encore sur la base d'un autre mécanisme sélectif.

Hypothèse des sous-variétés et perception

L'hypothèse des sous-variétés permet donc de définir les concepts comme des sous-variétés de l'espace perceptif. Les percepts peuvent alors être définis par la variété à laquelle ils appartiennent et par un jeu de coordonnées le long de cette sous-variété. Ce jeu de coordonnées permet de définir les degrés de variations pertinents pour chaque percept. Ceci rappelle le modèle des espaces conceptuels introduit par Gardenförs (Gardenfors 2004). Dans ce modèle, les concepts sont définis comme étant des régions convexes d'un espace dont les dimensions sont les traits caractéristiques des concepts. Cependant, dans le cas des espaces conceptuels ces traits sont définis *a priori* (par exemple la largeur, le poids, etc.). Dans le cas de l'hypothèse des sous-variétés, la paramétrisation est laissée libre. Le système de coordonnées choisi représente donc des variations pertinentes, mais pas obligatoirement les traits caractéristiques d'un concept tels que définis par exemple dans une ontologie.

Dans le cadre de l'hypothèse des sous-variétés, la multimodalité joue un rôle particulier dans la définition des concepts. Prenons comme illustration les concepts de chouette et de hibou (figure 2.12). Du point de vue de la modalité auditive, il ne fait pas trop de doute

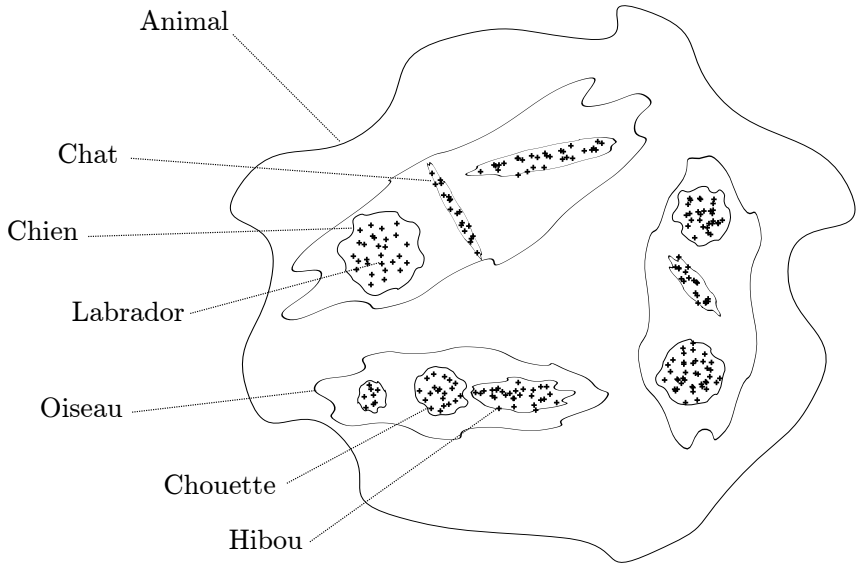


Figure 2.11 – *Selon la granularité observée, différentes sous-variétés se dessinent, permettant de définir une hiérarchie de concepts. Cette image en deux dimensions ne rend pas compte du fait que dans des espaces de grande dimensionalité, des dimensions négligées pour certaines sous-variétés peuvent devenir importantes pour des sous-variétés incluses, tandis que d'autres dimensions pertinentes pour la sous-variété générale peuvent au contraire être inutiles pour une sous-variété incluse : chaque sous-variété possède son propre système de coordonnées.*

que les mots *chouette* et *hibou* puissent assez facilement être distingués l'un de l'autre. D'un point de vue visuel en revanche, la distinction semblent beaucoup plus floue. Aussi les anglais ne font-ils pas la différence, désignant les deux ensembles d'espèces par le même nom *owl*. L'hypothèse des sous-variétés permet d'avancer l'idée que la vue conjointe d'un hibou (ou d'une chouette) avec la perception auditive du mot correspondant induit une séparation claire de deux sous-variétés différentes dans un espace perceptif multimodal. Chacune des deux sous-variétés peut alors développer son propre système de coordonnées. Tous les hibous ayant des aigrettes, aucune chouette n'en ayant, les deux sous-variétés peuvent alors se différencier sur ce critère. Par la suite, la vue seule d'un oiseau aura pour conséquence la perception d'un stimulus qui sera plus proche de l'une de ces deux sous-variétés. Le petit anglais, à l'inverse, n'aura pu apprendre à distinguer les deux espèces à partir de la modalité auditive. Au contraire, il peut même se représenter le concept *owl* comme une sous-variété dont une dimension (i.e. une caractéristique pertinente de variation de ce concept) est la taille des aigrettes. Par la suite, la vue d'une chouette ou d'un hibou sera donc associée au même concept dont une caractéristique sera la taille des aigrettes.

On peut tenter de réfuter cet exemple par le fait qu'un enfant français apprendra la différence entre une chouette et un hibou grâce à un raisonnement symbolique (on lui apprendra directement qu'une chouette n'a pas d'aigrette à l'inverse du hibou). C'est en effet peut-être le cas pour les concepts aux différences subtiles, comme la chouette et le hibou. En revanche, le jeune enfant apprend à distinguer un grand nombre de concepts

Figure 2.12 – *Chouette ou hibou ? Les anglais n'utilisent qu'un seul et même mot pour les désigner : owl. Comment expliquer l'émergence de deux catégories différentes ? (Images wikimedia commons)*

sans avoir besoin qu'on lui en fournisse une description précise : tout le monde sait faire la différence entre un chat et un chien, et pourtant lorsque l'on y réfléchit, il est assez peu aisé de formuler un critère permettant de les distinguer visuellement. Dans le cas des chiens et des chats, la différence visuelle peut être apprise elle aussi à partir de la modalité auditive, soit à partir de l'écoute du mot correspondant (“*Oh, tu as vu le chat ?*”) mais également à partir de l'entente des miaulements et aboiements.

Hypothèse des sous-variétés pour l'action

Les sous-variétés correspondant à la perception sont imposées par la structure de l'environnement. Dans le cas de l'action, chaque action est le résultat d'un certain motif d'activité des neurones qui peut en théorie varier de manière continue pour couvrir tout l'espace, ce qui ferait tomber la notion de sous-variété. La vie quotidienne nous montre cependant que nous utilisons un répertoire d'action stéréotypées (jeter, saisir, tourner, etc.) et que nous maitrisons très mal un très grand nombre d'actions que nous sommes toutefois capables d'exécuter (peu de gens savent se servir un verre d'eau dans leur dos, bien que cela soit physiquement possible).

L'existence de ces répertoires d'actions permet de représenter les actions sous la forme de sous-variétés. Dans ce cas, chaque type d'action (saisir, jeter, etc.) correspond à une sous-variété et les coordonnées le long des sous-variétés correspondent à l'exécution précise de chaque action (par exemple force appliquée, vitesse, paramétrisation de la trajectoire, etc.). Ces sous-variétés d'actions seraient donc définies dans des espaces fonctionnels dont la nature reste à définir. L'hypothèse des sous-variétés implique uniquement l'existence d'une famille de fonctions paramétrables générant des trajectoires motrices et dont les différents jeux de paramètres correspondants aux différentes actions se répartissent le long de sous-variétés.

Se pose alors la question de savoir pourquoi, à partir de la capacité à réaliser n'importe quelle action, nous apprenons un répertoire d'actions sous la forme de sous-variétés. Nous n'apporterons pas de réponse définitive à cette question, mais nous pouvons avancer deux hypothèses. La première, imparfaite, s'appuie sur l'existence de réflexes innés (préhension, mouvement du bras, succion, etc.) qui définissent des points distincts dans l'espace des fonctions motrices. Les mécanismes d'assimilation et d'accommodation à la Piaget explorent alors l'espace moteur à partir de ces points. Certaines variantes de ces actions n'apportant pas de résultats intéressants, elles sont délaissées (ceci suppose un mécanisme d'apprentissage par renforcement), et l'exploration se fait alors suivant certaines dimensions, donnant lieu à des sous-variétés. Cette hypothèse n'est toutefois pas très convaincante, car elle reporte le problème sur l'existence innée de différents types de mouvements, et il est assez compliqué d'expliquer le passage de quelques réflexes innés à un répertoire d'actions complexes maîtrisé par n'importe quel adulte. L'autre hypothèse s'appuie sur les liens entre action et perception. Dès lors que la perception prend la forme de sous-variétés dans un espace perceptif et que la perception peut déclencher l'action, comme on l'a vu notamment avec l'exemple des lésions du cortex moteur supplémentaire (CMS), il est possible de faire l'hypothèse que la projection de la perception vers le cortex moteur soit suffisamment régulière pour obtenir une répartition des actions effectuées le long de sous-variétés. De même, la perception des effets de ces actions définit également des sous-variétés dans un espace perceptif. Dans ce cas, l'exploration peut être uniforme et aléatoire, et la perception aide à structurer l'espace moteur. Ainsi, deux actions différentes peuvent produire un résultat perceptif similaire (par exemple les différentes manières de lancer une balle). Cette similarité perceptive peut alors être utilisé pour biaiser l'apprentissage des représentations de ces deux actions pour qu'elles soient similaires elles aussi. De même, deux actions similaires peuvent produire des effets perceptifs radicalement différents (par exemple lancer une balle et frapper un mur). Les représentations apprises pour ces deux actions tendront donc elles aussi à être différentes.

Nous avons jusqu'à présent considéré la représentation des actions à haut niveau, sans en considérer l'aspect temporel (la trajectoire suivie étant un paramètre de la sous-variété correspondante), comme le suggère notamment l'existence de neurones miroirs. Le problème de la temporalité sera abordé de manière plus approfondie dans le chapitre 5. Nous y développerons également la notion de sous-variété dans un espace fonctionnel.

Chapitre 3

Réseaux de neurones et apprentissage profond

- Radford Neal : I don't necessarily think that
the Bayesian method is the best thing to do in all cases...
- Geoffrey Hinton : Sorry Radford, my prior probability
for you saying this is zero, so I couldn't hear what you said.

Échange à un séminaire CIFAR2004 - rapporté par Yann LeCun [1]

Sommaire

Nous quittons temporairement le domaine de la robotique développementale et les problématiques liées à la perception et à l'action pour présenter les réseaux de neurones et l'apprentissage profond. Il s'agit en effet des briques de base que nous utiliserons dans la suite de notre travail.

Nous commençons par décrire les réseaux de neurones avant d'étudier les problématiques spécifiques à l'apprentissage profond.

1. Les citations de ce chapitre proviennent de la page "Fun Stuff" du site `yann.lecun.com`.

3.1 Les réseaux de neurones

Geoff Hinton doesn't need to make hidden units. They hide by themselves when he approaches.

Yann LeCun

3.1.1 Neurones et perceptron

Avant de décrire le perceptron, premier algorithme d'apprentissage proposé s'inspirant du fonctionnement neuronal, nous allons très brièvement décrire le neurone biologique et la règle de Hebb.

Neurone biologique

Le neurone est une cellule fondamentale du système nerveux des êtres vivants. Le cerveau humain en contient plusieurs dizaines de milliards. Chaque neurone est composé (figure 3.1) :

- d'un corps cellulaire contenant le noyau,
- de dendrites, nombreuses et ramifiées, qui conduisent l'influx nerveux de leur périphérie jusqu'au corps cellulaire,
- d'un axone, qui conduit le potentiel d'action émis au niveau du corps cellulaire jusqu'aux dendrites d'autres neurones. L'influx nerveux y est alors transmis par voie chimique au niveau des synapses. Les axones peuvent mesurer plusieurs dizaines de centimètres de longueur et sont entourés de gaines de myéline permettant d'optimiser la transmission de l'influx nerveux.

Chaque neurone reçoit donc en entrée des signaux en provenance d'autres neurones, transmis par les dendrites jusqu'au corps cellulaire où ils s'additionnent. L'importance de chaque signal reçu est modulée à la fois par la longueur de la dendrite lui permettant d'atteindre le corps cellulaire et par l'efficacité de la liaison synaptique entre l'axone présynaptique et la dendrite post-synaptique. S'il dépasse un certain seuil, le signal résultant au niveau du corps cellulaire peut alors donner lieu à un potentiel d'action, c'est-à-dire un pic de potentiel électrique qui se propage à travers l'axone jusqu'aux autres neurones dont les dendrites sont connectées à cet axone : on dit que ce neurone *décharge*.

En 1949, Donald Hebb postule que les capacités d'apprentissage du cerveau résultent d'une règle très simple, souvent résumée par "des neurones qui déchargent en même temps sont des neurones qui se lient ensemble" (Hebb 1949). Ainsi, si un neurone A décharge régulièrement juste avant un neurone B, alors un mécanisme biochimique accroît l'efficacité de la cellule A à induire un potentiel d'action dans la cellule B. Ce mécanisme est désigné sous le nom de *plasticité synaptique*. Dans la règle de Hebb originale, cette précédence temporelle de l'activité de A sur celle de B est essentielle et a été mise en évidence expérimentalement (Zhang et al. 1998). Elle s'accompagne également de l'effet inverse : si le neurone A décharge régulièrement juste après le neurone B, alors la liaison entre A et B est affaiblie (apprentissage anti-hebbien) (voir figure 3.2). Dans les modèles computationnels les plus abstraits, cette précédence temporelle est souvent écartée et la règle se

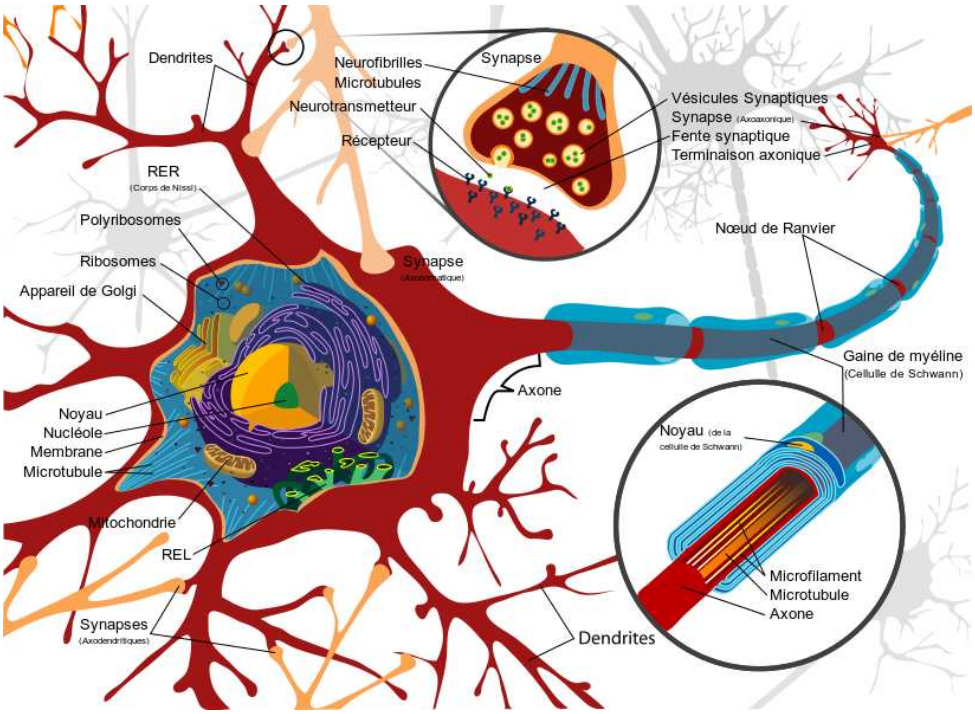


Figure 3.1 – *Schéma d'un neurone biologique. (Image wikimedia commons)*

transforme en une association simple entre deux neurones qui sont actifs régulièrement aux mêmes pas de temps de la simulation.

Neurone formel

En 1943, Warren McCulloch et Walter Pitts proposent un modèle mathématique très simplifié du neurone biologique (McCulloch et Pitts 1943). Il s'agit du premier modèle de *neurone formel*.

Étant donnés un ensemble d'entrées $\mathbf{x} \in \mathbb{R}^n$ et une sortie $y \in \mathbb{R}$, le neurone formel de McCulloch et Pitts [2] associe un poids w_i à chaque entrée x_i et calcule la somme pondérée des entrées par leurs poids respectifs à laquelle s'ajoute un biais b. Le résultat est alors transformé par une fonction d'activation non linéaire σ :

$$y = \sigma\left(\sum_i w_i x_i + b\right) = \sigma\left(\mathbf{w}^\top \mathbf{x} + b\right). \tag{3.1}$$

2. Dans l'article initial (McCulloch et Pitts 1943), les entrées et les sorties sont des variables booléennes, les poids synaptiques ne peuvent prendre que des valeurs entières (correspondant au nombre de synapses connectant deux neurones) et les connexions peuvent être de deux types (OU logique et NON logique). Des propositions logiques définissent alors la valeur de sortie du neurone. Nous présentons ici une extension de ce modèle, généralement considérée par abus de langage comme étant *le* neurone formel de McCulloch et Pitts.

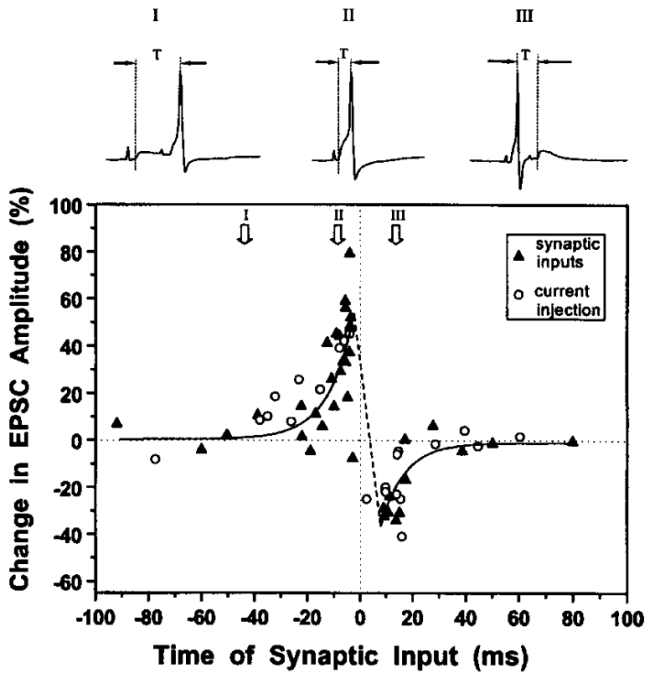


Figure 3.2 – *Plasticité synaptique d'un neurone biologique. Deux neurones sont stimulés électriquement de manière répétée selon différents intervalles de temps. Après 100 secondes de stimulations, on mesure le courant post-synaptique induit par une stimulation du neurone pré-synaptique. Ce graphique représente en ordonnée la variation du courant post-synaptique par rapport à sa valeur initiale (avant stimulation répétée) en fonction de l'intervalle de temps séparant la stimulation des deux neurones. Dans le cas où le neurone pré-synaptique est stimulé avant le neurone post-synaptique, on observe une augmentation du courant induit, c'est-à-dire un renforcement de la liaison synaptique. Au contraire, si le neurone pré-synaptique a été stimulé après le neurone post-synaptique, la liaison synaptique a été affaiblie. (Image tirée de (*Zhang *et al. 1998))*

Dans la version originelle, la fonction d'activation est la fonction de Heaviside :

$$H(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases} \tag{3.2}$$

mais d'autres fonctions ont été largement utilisées, comme la fonction sigmoïde

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.3}$$

De nombreux modèles de neurones formels ont depuis été développés, du plus simple au plus complexe et du plus abstrait au plus réaliste.

D'un côté ont été développés des modèles qui visent à étudier précisément les phénomènes neuronaux, comme le modèle de (Hodgkin et Huxley 1952). D'autres modèles visent à rendre compte du comportement d'assemblées de neurones, comme (Eggert et Hemmen 2001). Ces modèles revendiquent une plausibilité biologique aussi grande que possible et non le développement d'algorithmes d'intelligence artificielle concis et économes. Nous ne nous y attarderons donc pas.

De l'autre côté, de nombreux modèles de neurones formels ont été développés pour servir de brique de base à la conception d'algorithmes d'intelligence artificielle. Les plus simples sont des prolongements du modèle de McCulloch et Pitts en remplaçant la fonction de Heaviside par une autre fonction non-linéaire, comme une sigmoïde ou une tangente hyperbolique. Plus récemment, des fonctions comme le "softplus", la fonction linéaire rectifiée, ou le maximum ont été l'objet d'un intérêt soutenu (Nair et Hinton 2010 ; Glorot et al. 2011 ; Goodfellow et al. 2013). La figure 3.3 illustre les fonctions les plus populaires. La philosophie de ces modèles consiste à identifier la valeur de sortie de ces neurones au taux de décharge d'un neurone biologique (ou d'un groupe de neurones). Des modèles reflétant plus finement la dimension temporelle de la réponse des neurones ont également été proposés, comme le modèle "intègre et décharge" (*integrate and fire*) (Burkitt 2006), qui reste confiné aux modèles bio-mimétiques ainsi qu'à certains réseaux spécifiques (par exemple (Maass et al. 2002)) du fait du coût computationnel élevé.

Perceptron

En 1958, Frank Rosenblatt propose de doter les réseaux de neurones d'une règle d'apprentissage supervisé inspirée de l'apprentissage Hebbien, à ceci près que l'activité post-synaptique est remplacée par l'erreur entre l'activité post-synaptique souhaitée y et celle $\hat{y}$ obtenue en sortie du réseau :

$$\Delta w_i \propto (y - \hat{y}) x_i. \tag{3.4}$$

Cette règle permet d'apprendre un classifieur linéaire qui sépare l'espace d'entrée par un hyperplan, comme illustré figure 3.4.

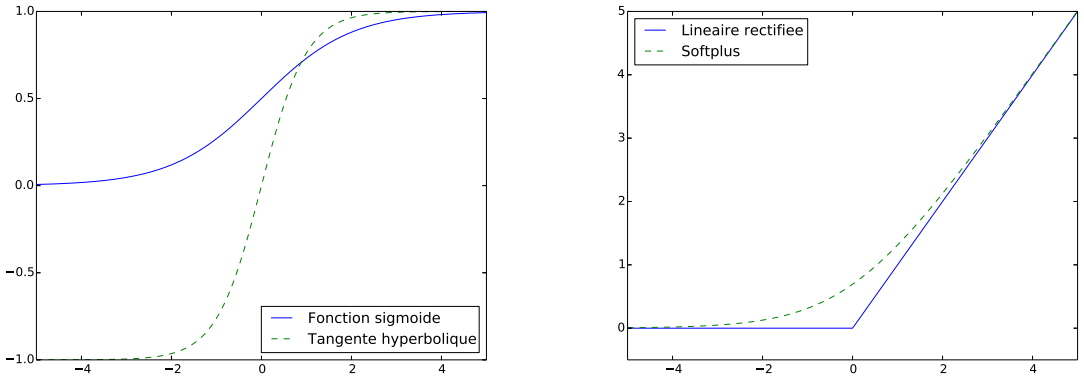


Figure 3.3 – *Fonctions d'activation couramment utilisées dans les réseaux de neurones.*

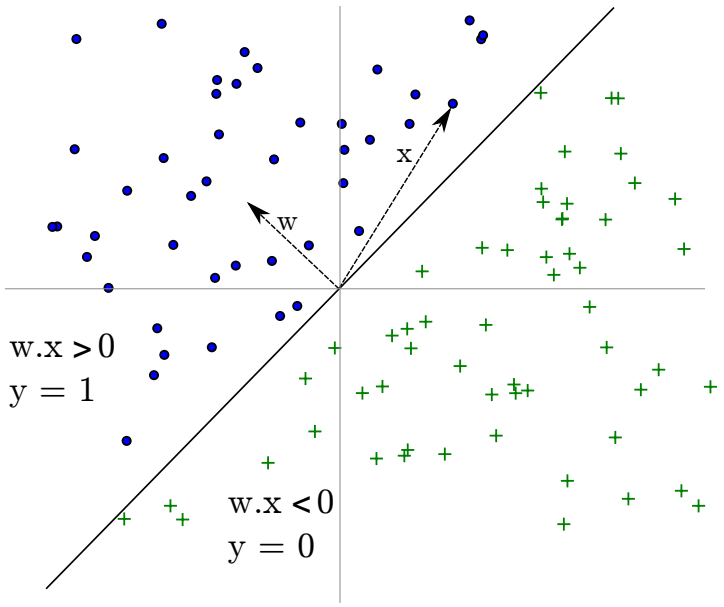


Figure 3.4 – *Un perceptron utilisant la fonction de Heaviside apprend un classifieur linéaire qui partage l'espace selon un hyperplan orthogonal au vecteur w.*

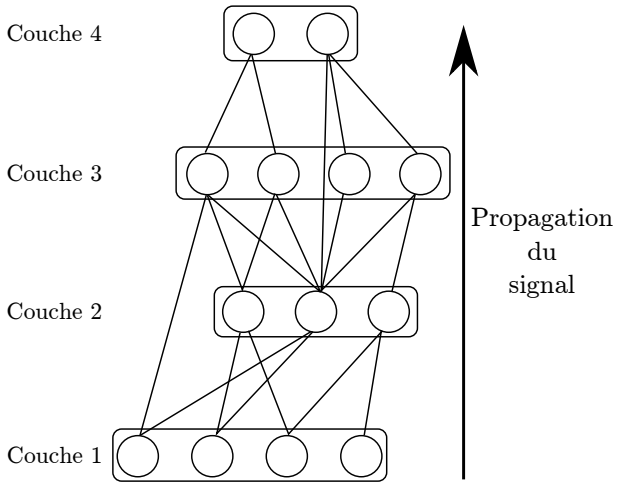


Figure 3.5 – *Exemple de réseau feedforward. Il est possible de définir un sens de propagation du signal dans le réseau depuis la première couche jusqu'à la dernière, sans repasser par un neurone d'une couche inférieure.*

3.1.2 Réseaux feedforward

Le perceptron est le plus simple des réseaux de neurones, se comportant comme un simple classifieur linéaire. Afin d'apprendre des tâches plus complexes, il est possible d'augmenter le nombre de neurones et de les connecter entre eux.

Plusieurs types de réseaux peuvent ainsi être obtenus. Les plus simples sont les réseaux dits *feedforward*.

Dans ceux-ci, il est possible d'affecter chaque neurone à une couche et d'ordonner les différentes couches entre elles de sorte que les sorties des neurones d'une couche i ne soient reliées qu'à des entrées de neurones appartenant à une couche $j > i$ (figure 3.5).

Perceptron multicouches

En empilant plusieurs perceptrons, on obtient un perceptron multicouches. Chaque neurone de chaque couche se comporte toujours comme un classifieur linéaire, mais l'utilisation de couches intermédiaires permet de créer des partitions complexes de l'espace. Ceci permet de projeter les données fournies en entrée dans de nouveaux espaces, dans lesquels une tâche initialement non linéaire peut devenir linéaire.

Cependant, l'utilisation de couches intermédiaires rend impossible d'entraîner ces réseaux en utilisant la règle d'apprentissage du perceptron. C'est pourquoi il a fallu attendre la publication des techniques de rétropropagation du gradient ((Werbos 1974 ; Parker 1985 ; Le Cun 1986) et plus particulièrement (Rumelhart et al. 1986)) pour que ces réseaux soient plus largement utilisés. Comme nous le verrons par la suite, ces techniques nécessitent en particulier que les fonctions d'activation utilisées soit dérivables. À partir de ce moment là, la fonction de Heaviside a donc été généralement remplacée par la fonction sigmoïde.

Sigmoïdes versus Gaussiennes et malédiction de la dimensionalité

D'autres fonctions que la sigmoïde peuvent être utilisées, par exemple des fonctions gaussiennes. Comme nous allons le voir, la fonction sigmoïde a cependant un avantage important dans les espaces de grande dimensionalité. Une fonction sigmoïde peut être vue comme une approximation de la fonction de Heaviside, c'est-à-dire séparant l'espace en deux : une partie où la sigmoïde vaut 0, l'autre où elle vaut 1. Cette séparation étant un hyperplan (voir figure 3.4), l'espace est partagé en deux parties à peu près égales (dépendant de la valeur du biais) quelle que soit la dimensionalité de cet espace. Au contraire, en utilisant une fonction d'activation gaussienne (comme le font souvent les *Radial Basis Function Networks*), on retrouve une manifestation de la malédiction de la dimensionalité exposée au paragraphe 2.4.1. En considérant la largeur à mi-hauteur l de la gaussienne pour distinguer deux zones de l'espace – celle où la gaussienne prend une valeur supérieure à 0.5 et celle où elle prend une valeur inférieure – c'est-à-dire en considérant que la gaussienne partage l'espace en deux selon une hypersphère, l'exemple de l'hypercube du paragraphe 2.4.1 montre qu'il est nécessaire d'avoir un nombre exponentiel de gaussiennes pour couvrir l'espace. En effet, le rapport entre le volume de l'hyperboule de rayon $l/2$ et le volume de l'hypercube de côté l vaut :

$$\frac{\pi^{d/2}}{2^d \Gamma\left(\frac{d}{2}+1\right)} \tag{3.5}$$

où Γ est la fonction Gamma ($\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} dt$). Dans le cas $d = 100$, on obtient un ratio d'environ 10^{-70}. Ceci veut dire qu'une gaussienne dont la largeur à mi-hauteur est égale au côté d'un hypercube prend une valeur supérieure à 0.5 sur environ $10^{-68}\%$ du volume de l'hypercube. L'utilisation de gaussiennes dans des espaces de grande dimensionalité doit donc être envisagée avec la plus grande prudence.

Rétropropagation du gradient

Dans la suite, on note $\mathbf{h_i}$ la valeur de sortie de la couche i du réseau, $W_{i,j}$ la matrice de poids synaptiques entre la couche i et la couche j, et l'on implémente les biais sous la forme d'un neurone dont l'activation est toujours 1 (la valeur des biais étant dès lors codée dans les matrices W par les valeurs des connexions synaptiques avec ce neurone). En conséquence et par souci de simplification des équations, nous ne faisons dorénavant plus apparaître les biais de manière explicite. On note de plus $\mathbf{g_i}$ les activités des neurones de la couche i avant application de la fonction d'activation

$$\mathbf{g_i} = W_{i-1,i}\mathbf{h_{i-1}} \tag{3.6}$$

$$\mathbf{h_i} = \sigma(\mathbf{g_i}). \tag{3.7}$$

Tout comme la règle du perceptron, la technique de rétropropagation du gradient consiste à modifier de manière incrémentale l'ensemble des paramètres Θ du réseau (poids synaptiques et biais), de manière à rapprocher la sortie obtenue pour une entrée $\mathbf{x}$ de la

sortie désirée $\mathbf{y}$. Cette différence permet en effet de définir une erreur sur un ensemble de données $D = \{(\mathbf{x}, \mathbf{y})\}$:

$$E(D, \Theta) = \frac{1}{2} \sum_{(\mathbf{x},\mathbf{y}) \in D} ||\mathbf{y} - \hat{\mathbf{y}}||^2 = \frac{1}{2} \sum_{(\mathbf{x},\mathbf{y}) \in D} ||\mathbf{y} - f(\mathbf{x}, \Theta)||^2 \tag{3.8}$$

où la fonction $f(\mathbf{x}, \Theta) = h_n \circ h_{n-1} \circ \cdots \circ h_1(\mathbf{x})$ correspond à l'ensemble des transformations subies par une donnée $\mathbf{x}$ à travers le réseau de paramètres Θ, composé de n couches.

En utilisant une fonction dérivable comme fonction d'activation des neurones, il est possible de calculer le gradient de cette erreur par rapport aux poids du réseau :

$$\frac{\partial E}{\partial W_{i,j}^{k,l}} = \frac{\partial E}{\partial h_j^l} \frac{\partial h_j^l}{\partial g_j^l} \frac{\partial g_j^l}{\partial W_{i,j}^{k,l}} \tag{3.9}$$

soit

$$\frac{\partial E}{\partial W_{i,j}^{k,l}} = \frac{\partial E}{\partial h_j^l} \times \sigma'(g_j^l) \times h_i^k \tag{3.10}$$

où $W_{i,j}^{k,l}$ représente le poids synaptique entre le neurone k de la couche i et le neurone l de la couche j.

La valeur $\frac{\partial E}{\partial h_j^l}$ dépend de la couche considérée :

– si j est la couche de sortie ($j = n$), alors $\mathbf{h_j} = \hat{\mathbf{y}}$ et

$$\frac{\partial E}{\partial h_j^l} = \hat{y}^l - y^l, \tag{3.11}$$

– si j est une couche intermédiaire, il faut considérer les neurones de la couche $j + 1$ qui reçoivent la couche j en entrée

$$\frac{\partial E}{\partial h_j^l} = \sum_m \frac{\partial E}{\partial h_{j+1}^m} \frac{\partial h_{j+1}^m}{\partial g_{j+1}^m} \frac{\partial g_{j+1}^m}{\partial h_j^l} \tag{3.12}$$

soit

$$\frac{\partial E}{\partial h_j^l} = \sum_m \frac{\partial E}{\partial h_{j+1}^m} \sigma'(g_{j+1}^m) W_{j,j+1}^{l,m}. \tag{3.13}$$

L'équation 3.13 donne donc une formule récursive pour calculer le gradient de l'erreur à la couche j à partir du gradient à la couche $j + 1$, d'où le nom de rétropropagation du gradient. L'erreur entre la valeur de sortie désirée et la valeur obtenue peut alors être minimisée par une descente de gradient avec un taux d'apprentissage α :

$$\Delta W_{i,j} = -\alpha \frac{\partial E}{\partial W_{i,j}}. \tag{3.14}$$

Le gradient peut lui-même être calculé pour l'erreur sur tout l'ensemble d'apprentissage D (*batch learning*), ce qui permet d'avoir un taux d'apprentissage assez élevé, pour chaque donnée $\mathbf{x}, \mathbf{y}$ séparément (apprentissage en ligne), avec un taux d'apprentissage très faible,

ou, de façon intermédiaire, pour des sous-ensembles de l'ensemble d'apprentissage (*mini-batches*).

Il existe également d'autres méthodes de descente de gradient qui peuvent se révéler plus efficaces, comme le gradient conjugué (Nocedal et Wright 2006) ou les algorithmes dérivés de la méthode de Newton qui font appel à des informations d'ordre deux sur la fonction (matrice Hessienne) (Martens 2010 ; Dauphin et al. 2014) et sont donc généralement assez coûteux en calculs.

Dans nos expériences, nous utiliserons à plusieurs reprises la méthode du *momentum* qui consiste à intégrer (avec pertes) les gradients successifs calculés à chaque itération. Cette méthode a l'avantage de demander très peu de calculs supplémentaires par rapport à la version de base de la descente de gradient. Elle introduit un moment η en plus du taux d'apprentissage α et l'équation 3.14 devient :

$$\Delta W_{i,j}^t = -\alpha \frac{\partial E}{\partial W_{i,j}^t} + \eta \Delta W_{i,j}^{t-1}. \tag{3.15}$$

Le moment η agit donc comme un terme de fuite sur l'intégration des mises à jour $\Delta W_{i,j}$ successives. Il est généralement choisi proche de 1.

Cartes auto-organisatrices

Le principe des cartes auto-organisatrices, ou cartes de Kohonen (Kohonen 1990), diffère très largement des perceptrons multicouches. Le but de ces réseaux est d'apprendre une représentation topologique des données fournies en entrée. Pour cela, elles ne possèdent qu'une seule couche cachée fonctionnant sur le principe du *gagnant prend tout*. Chaque neurone de cette couche cachée est associé à une valeur d'entrée (l'équivalent d'un prototype pour ce neurone) et connecté à ses neurones voisins selon une topologie donnée (par exemple 2D ou 3D). Chaque nouvelle donnée est alors comparée aux prototypes de chaque neurone et seul le neurone le plus proche, ainsi que ses voisins topologiques, sont modifiés selon leur distance par rapport à la nouvelle donnée. Différentes méthodes ont été proposées, la plus simple consistant à rapprocher fortement le prototype du neurone vainqueur de la nouvelle donnée et de rapprocher plus faiblement les prototypes des neurones voisins (voir figure 3.6).

Réseaux *gated*

Les réseaux *gated* constituent une famille particulière. Contrairement aux autres réseaux dont les connexions entre couches sont linéaires, les réseaux *gated* introduisent des interactions d'ordre supérieur. Au sein de ceux-ci, le poids de connexion w_{ij} entre deux neurones x_i et y_j est en effet modulé par l'activité d'un troisième neurone z_k :

$$y_j = \sigma((z_k \times w_{ij}) \times x_i). \tag{3.16}$$

Un tel mécanisme peut-être utilisé dans plusieurs buts : contrôler les flux de données dans le réseau ou modéliser des interactions multiplicatives entre plusieurs entrées (figure 3.7). Dans le premier cas, l'activation du neurone z_k varie entre 0 et 1 et celui-ci se

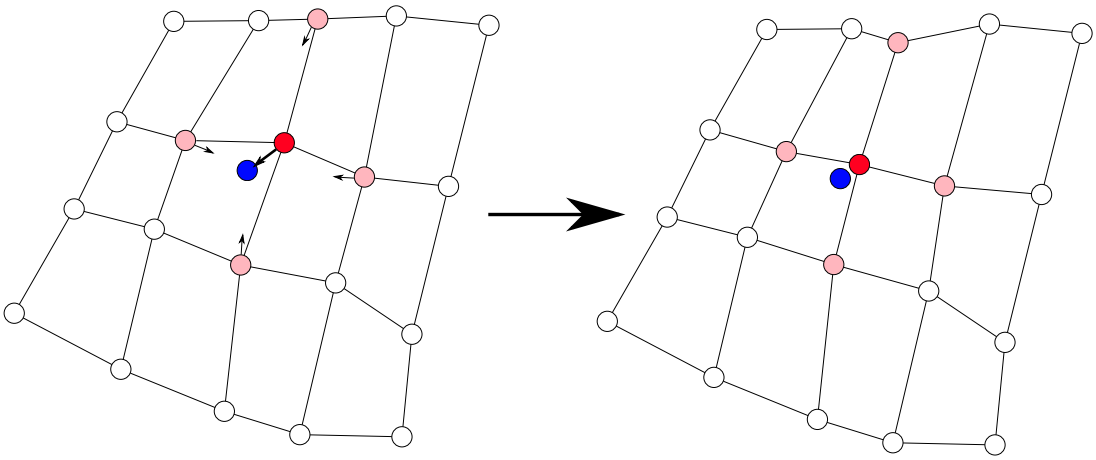


Figure 3.6 – *Dans une carte de Kohonen, les neurones sont reliés à leurs voisins selon une grille topologique. Lorsqu'une nouvelle donnée est présentée en entrée (ici en bleu), le neurone le plus proche (en rouge foncé) est désigné "vainqueur". Il est alors rapproché de la donnée correspondante, ainsi que ses plus proches voisins (en rouge clair). Différente règles de mise à jour existent, pour définir le voisinage concerné ainsi que l'amplitude de la mise à jour selon la distance au neurone vainqueur.*

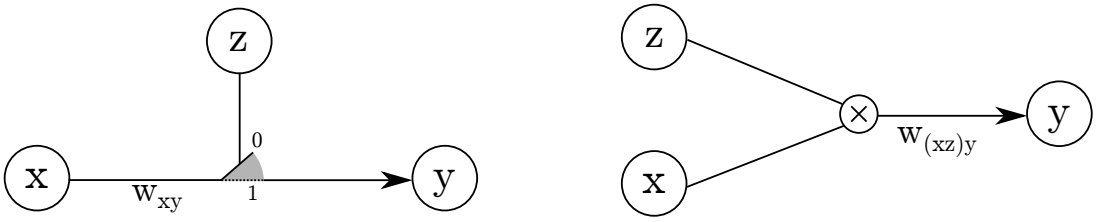


Figure 3.7 – *Deux visions d'une connexion gated pour deux utilisations différentes. À gauche, la connexion gated sert à contrôler le flux d'information dans le réseau. Le neurone z agit alors comme une barrière qui bloque, ou non, la transmission entre x et y. À droite, la connexion sert à implémenter une relation multiplicative entre deux entrées x et z.*

comporte comme une barrière qui laisse (valeur 1) ou non (valeur 0) passer l'information entre les deux neurones connectés. Ce mécanisme est utilisé notamment dans les réseaux "*Long-short term memories*" (Hochreiter et Schmidhuber 1997a). Dans le second cas, **x** et **z** sont considérés comme des entrées et ont un rôle symétrique. Leurs valeurs sont alors multipliées avant d'être projetées vers y par la connexion synaptique. De telles relations multiplicatives sont impossibles à modéliser avec des perceptrons standards [3] (Memisevic 2012a).

Dans la suite de notre travail, nous utiliserons à plusieurs reprises de tels réseaux sous la forme d'une couche insérée dans un réseau classique. Cette couche encapsule l'interaction multiplicative de façon à la rendre entièrement symétrique entre les trois couches **x**, **y** et **z**

3. Apprendre une relation multiplicative entre deux couches avec un simple perceptron, en utilisant la concaténation des deux couches en entrée, demanderait, comme nous le verrons par la suite, un nombre de neurones qui croît exponentiellement avec la précision désirée.

(voir figure 3.8) (Memisevic 2011). Afin de pouvoir faire interagir n'importe quel neurone d'une couche avec n'importe quel neurone d'une autre couche, chaque couche est d'abord projetée linéairement dans un nouvel espace. Étant donné que cet espace est utilisé pour réaliser les interactions multiplicatives, ces projections sont appelées "facteurs", et notées $\mathbf{f_x}$, $\mathbf{f_y}$ et $\mathbf{f_z}$:

$$\mathbf{f_x} = W_x \mathbf{x} \tag{3.17}$$
$$\mathbf{f_y} = W_y \mathbf{y} \tag{3.18}$$
$$\mathbf{f_z} = W_z \mathbf{z} \tag{3.19}$$

Chacune des paires de facteurs peut alors être multipliée terme à terme (opérateur $*$). Ce produit terme à terme est alors assimilé aux facteurs correspondant à la troisième couche :

$$\hat{\mathbf{f_x}} = \mathbf{f_y} * \mathbf{f_z} \tag{3.20}$$
$$\hat{\mathbf{f_y}} = \mathbf{f_x} * \mathbf{f_z} \tag{3.21}$$
$$\hat{\mathbf{f_z}} = \mathbf{f_x} * \mathbf{f_y}. \tag{3.22}$$

Ces facteurs peuvent alors être de nouveau projetés vers leur propre couche en utilisant les mêmes poids synaptiques que ceux permettant de les calculer :

$$\hat{\mathbf{x}} = \sigma(W_x^\top \hat{\mathbf{f_x}}) \tag{3.23}$$
$$\hat{\mathbf{y}} = \sigma(W_y^\top \hat{\mathbf{f_y}}) \tag{3.24}$$
$$\hat{\mathbf{z}} = \sigma(W_z^\top \hat{\mathbf{f_z}}). \tag{3.25}$$

Ces équations permettent de mettre en évidence la symétrie entre les trois couches $\mathbf{x}$, $\mathbf{y}$ et $\mathbf{z}$: étant données deux de ces couches, il est possible de les projeter vers la troisième couche. Cette symétrie est indispensable pour son utilisation dans les réseaux profonds non supervisés que nous exposerons par la suite.

Il est de plus important de remarquer que toutes les opérations associées à cette couche sont continues et dérivables et qu'il est donc possible, suivant le même principe que pour les perceptrons multicouches, de modifier les matrices de poids de manière incrémentale en suivant le gradient de l'erreur entre les valeurs obtenues et les valeurs désirées pour une couche.

L'utilisation de couches intermédiaires de "facteurs" permet de réduire le nombre de paramètres nécessaires pour faire interagir un neurone d'une couche avec n'importe quel neurone d'une autre couche. En effet, sans ces facteurs, il faudrait rajouter une matrice de poids reliant chaque neurone d'une couche à tous les poids connectant les neurones des deux autres couches. Le nombre de paramètres nécessaires serait donc de l'ordre de $n_x \times n_y \times n_z$ (avec n_x le nombre de neurones de la couche x). En projetant d'abord ces couches sur les couches de facteurs, le nombre de paramètres est de l'ordre de $n_f \times (n_x + n_y + n_z)$, ce qui permet donc de passer d'une complexité cubique à une complexité quadratique (Memisevic et Hinton 2010).

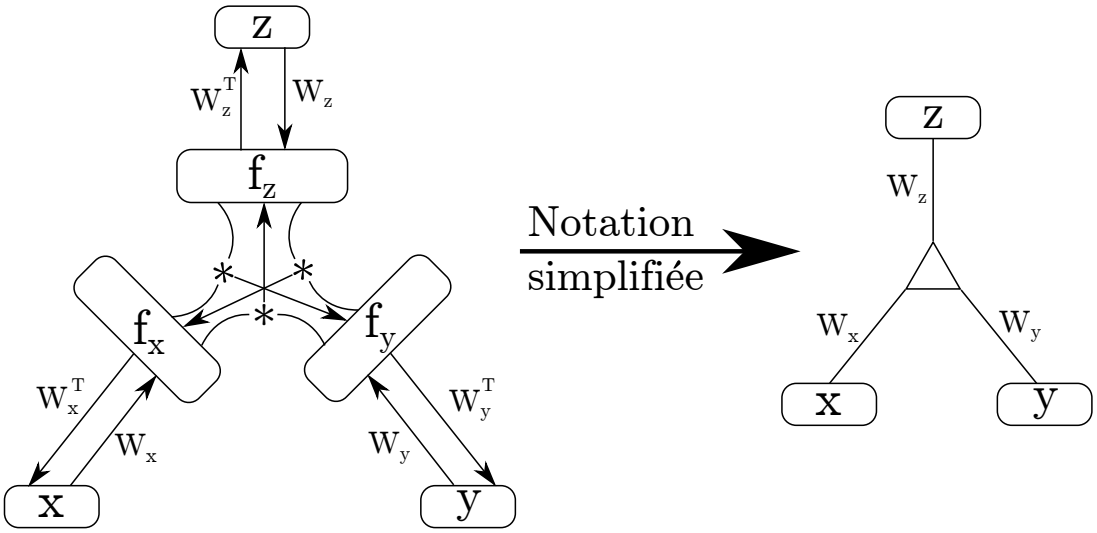


Figure 3.8 – *Schéma de principe d'une couche "gated". Le symbole $*$ désigne la multiplication terme à terme entre deux couches de neurones (par exemple : $\mathbf{f_z} = \mathbf{f_x} * \mathbf{f_y}$). Les trois couches* **x**, **y** *et* **z** *jouent un rôle parfaitement symétrique. À droite, nous donnons la notation simplifiée du réseau que nous utilisons dans la suite de notre travail.*

3.1.3 Réseaux récurrents

Nous avons décrit dans la partie précédente les familles les plus utilisées de réseaux feedforward, c'est-à-dire des réseaux dans lesquels l'information se propage de manière séquentielle entre les entrées et les sorties, sans bouclage au sein du réseau. Lorsqu'au contraire il est impossible d'ordonner les neurones selon des couches parcourues séquentiellement, on parle de réseaux récurrents. Là encore, différentes familles ont été proposées.

Réseaux de Elman et variantes

Une des familles les plus connues sont les réseaux de type Elman (Elman 1990). Ces réseaux ont une architecture proche des perceptrons multicouches, à ceci près qu'il contiennent des couches cachées dans lesquelles sont copiées les activités des neurones des couches intermédiaires à chaque pas de temps. Ces valeurs sont alors réinjectées dans le réseau au pas de temps suivant. Différents noms peuvent être attribués à ces réseaux selon les couches d'où proviennent les données copiées et dans lesquelles sont réinjectées ces valeurs (voir figure 3.9 pour deux exemples).

Réseaux de Hopfield et machines de Boltzmann

Les réseaux de Hopfield et les machines de Boltzmann sont une famille assez distincte des réseaux précédents. Ces réseaux s'appuient sur une approche énergétique, inspirée de la physique statistique, pour décrire l'évolution de l'activité des neurones.

Pour illustrer cette approche, considérons le cas d'un cristal de particules ferromagnétiques dans lequel chaque particule peut prendre deux valeurs de moment magnétique σ

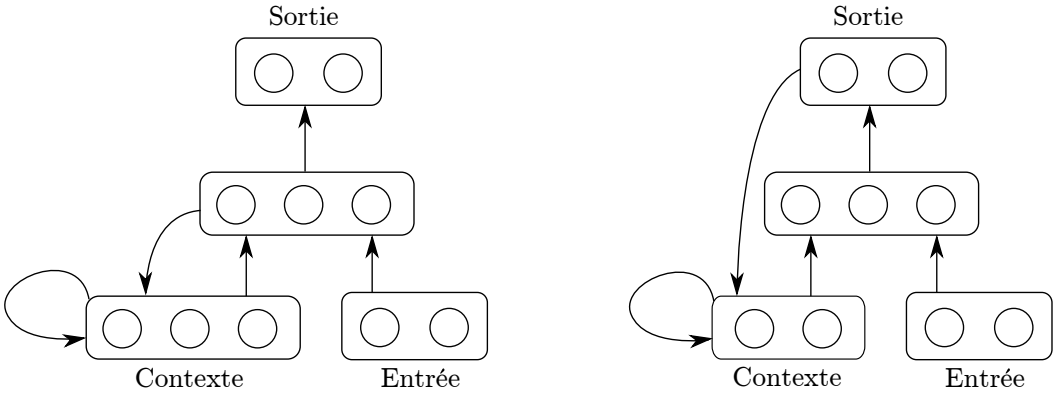


Figure 3.9 – *Exemples de réseaux récurrents populaires, le réseau de Elman (à gauche) et le réseau de Jordan (à droite). À chaque pas de temps, les données de la couche intermédiaire ou de la couche de sortie sont copiées dans une couche de "contexte" qui est réinjectée en entrée au pas de temps suivant.*

différentes, notées +M et -M. Chaque particule i interagit avec ses plus proches voisins j, avec une constante de couplage (appelée interaction d'échange) W_{ij}. Ce couplage traduit le fait que l'orientation magnétique d'une particule a tendance à influer sur l'orientation magnétique de ses voisines. De même, la présence d'un champ magnétique b externe peut lui aussi influencer l'orientation des particules. En considérant l'ensemble des couplages, on peut alors définir une énergie du cristal comme

$$E = -\sum_{i,j} \sigma_i W_{ij} \sigma_j - b \sum_i \sigma_i. \tag{3.26}$$

Ainsi, si la constante de couplage est positive, l'énergie sera minimale si toutes les particules ont un moment magnétique orienté dans la même direction. Le retournement d'une particule ($\sigma_i \to \sigma_i'$) aura alors un effet énergétique de $\Delta\sigma_i(\sum_j W_{ij}\sigma_j + b)$, où $\Delta\sigma_i = \sigma_i' - \sigma_i$. La physique statistique nous dit alors que ce retournement a une probabilité

$$p(\sigma_i \to \sigma_i') = \frac{1}{1 + \exp(\beta(\sum_j W_{ij} + b)\Delta\sigma_i)} \tag{3.27}$$

où β est proportionnel à l'inverse de la température du système ($\beta = \frac{1}{k_b T}$) : plus un retournement augmente l'énergie totale du système, moins celui-ci est probable. À une température donnée, il est donc possible de simuler l'évolution de l'état du cristal en choisissant aléatoirement une particule, en calculant sa probabilité de retournement, et en la retournant aléatoirement selon cette probabilité.

Les machines de Boltzmann suivent exactement ce principe : les particules sont remplacées par des neurones qui prennent généralement les valeurs 0 et +1 et chaque neurone est couplé aux autres par ses connexions synaptiques W_{ij}. À la différence d'un cristal où ce couplage dépend directement de relations de voisinage spatial en étant généralement invariant au sein du réseau (tous les atomes interagissent avec leurs voisins selon les mêmes lois physiques), les machines de Boltzmann n'ont pas cette contrainte et les

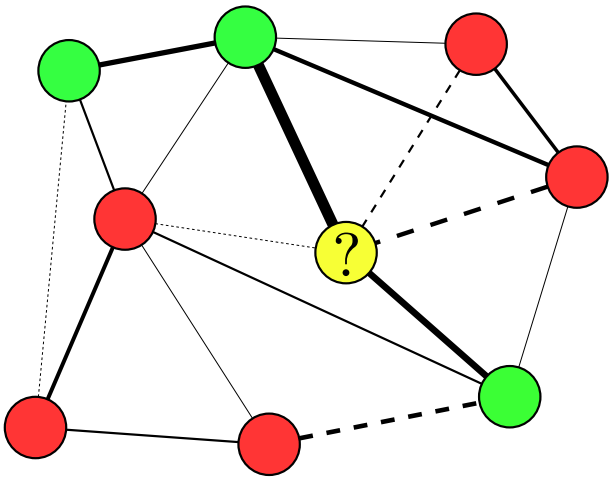


Figure 3.10 – *Illustration du principe d'une machine de Boltzmann. On représente les poids synaptique positifs par des traits pleins, dont l'épaisseur est proportionnelle à la valeur de la connexion. Les traits pointillés correspondent à des valeurs négatives. Les couleurs rouges et vertes correspondent aux deux valeurs possibles pour l'activité des neurones. On peut faire évoluer l'état du réseau en choisissant aléatoirement un neurone, et en tirant l'activité de ce neurone selon l'activité de ses voisins. Dans le cas du neurone " ?", il est lié fortement à deux neurones "verts", et de manière négative à trois neurones "rouges". La probabilité qu'il passe dans l'état "vert" est donc beaucoup plus importante que celle qu'il passe dans l'état "rouge".*

couplages sont totalement libres. De même, l'équivalent du champ magnétique externe n'est pas contraint à être le même pour l'ensemble du réseau, ce qui permet d'avoir un biais propre à chaque neurone.

Tout comme pour un cristal de particules ferromagnétiques, il est possible de faire évoluer l'état d'un réseau de Boltzmann en tirant de manière séquentielle l'état de neurones choisis aléatoirement (figure 3.10). En répétant indéfiniment ce mécanisme, le réseau produit différents états selon une distribution de probabilité, qui dépend des valeurs W_{ij} et b_i. D'après les lois de la physique statistique, un état $\mathbf{s}$ (représenté par un vecteur $(s_0, \cdots, s_n)$ des activités des n neurones du réseau) a une probabilité :

$$p(\mathbf{s}) = \frac{1}{Z} \exp(\beta(\sum_{i,j} s_i s_j W_{ij} + \sum_i b_i s_i)) \tag{3.28}$$

où Z est une constante de normalisation telle que la somme des probabilités de tous les états est égale à 1.

Les machines de Boltzmann peuvent alors être utilisées pour représenter une distribution de probabilité donnée. Par exemple, en considérant une base de données *a priori*, il est possible d'identifier certains neurones de la machine de Boltzmann avec les valeurs des entrées de la base de données, puis apprendre à générer une activité de ces neurones qui reproduise la distribution de probabilité des entrées de la base de données. Les neurones identifiés avec les valeurs des entrées de la base sont appelés neurones *visibles*, les autres neurones étant des neurones *cachés*.

En notant p^d la distribution définie par la base de données et p^g la distribution générée par la machine de Boltzmann pour les neurones visibles, l'apprentissage consiste à minimiser la divergence de Kullback-Leibler $KL(p^d, p^g)$. Cette minimisation peut s'effectuer grâce à une règle d'apprentissage concise (Ackley et al. 1985) :

$$\frac{\partial KL}{\partial W_{ij}} = -\alpha(p^d_{ij} - p^g_{ij}) \tag{3.29}$$

où $p_{ij} = p_i \times p_j$ est égale à la probabilité que les neurones i et j soient actifs en même temps, avec α le taux d'apprentissage. De la même manière

$$\frac{\partial KL}{\partial b_i} = -\alpha(p^d_i - p^g_i). \tag{3.30}$$

La règle d'apprentissage est donc très simple, mais cependant coûteuse en temps de calcul, car elle nécessite d'estimer la distribution de probabilité générée par la machine de Boltzmann, ce qui implique d'échantillonner chaque neurone un très grand nombre de fois de manière à atteindre d'une part l'équilibre thermique, puis à échantillonner les valeurs un nombre suffisant de fois pour pouvoir approcher correctement la distribution de probabilité. En effet, la complexité exponentielle de la constante de normalisation Z (il faut sommer sur tous les états possibles) rend tout calcul analytique prohibitif. Une heuristique a depuis été proposée pour approcher cette règle d'apprentissage. Nous la détaillerons dans la section 3.2.4 dans le cas des machines de Boltzmann restreintes.

Les réseaux de Hopfield (Hopfield 1982) sont un cas particulier de machines de Boltzmann, dont la tâche est de stocker n motifs d'activation distincts. Ces motifs correspondent donc à des minima locaux de l'énergie. Ces réseaux ont alors la propriété de débruiter des motifs ne correspondant pas à des motifs appris, en modifiant séquentiellement l'activation des neurones jusqu'à atteindre un minimum d'énergie.

Les réseaux "réservoir"

Les réseaux "*echo-state*" (Jaeger 2001) et "*liquid state*" (Maass et al. 2002) forment une troisième famille de réseaux récurrents, dont la principale caractéristique est que les valeurs des connexions récurrentes initialisées aléatoirement ne sont pas modifiées par apprentissage. Seules les connexions entre les neurones de ce "réservoir" récurrent et des neurones identifiés comme neurones de sortie sont apprises, à la manière d'un perceptron.

Ce type de réseaux initialisés aléatoirement et avec un apprentissage limité à certains poids seulement est précurseur d'un courant récent appelé *Extreme learning machine* (Huang et al. 2006) qui, contrairement à ce que son nom pourrait laisser croire, choisit de minimiser l'étendue de l'apprentissage en le reléguant à la couche de sortie des réseaux de neurones. Pour un perceptron à une seule couche cachée, cette méthode consiste à doter la couche cachée d'un très grand nombre de neurones, qui sont reliés aux entrées par des poids tirés aléatoirement. Seuls les poids entre la couche cachée et la couche de sortie sont appris. La philosophie de cette approche consiste donc à projeter aléatoirement les données dans un espace de très grande dimension, en espérant que n'importe quel problème y devienne linéaire et puisse être par la suite appris par une seule couche de neurones.

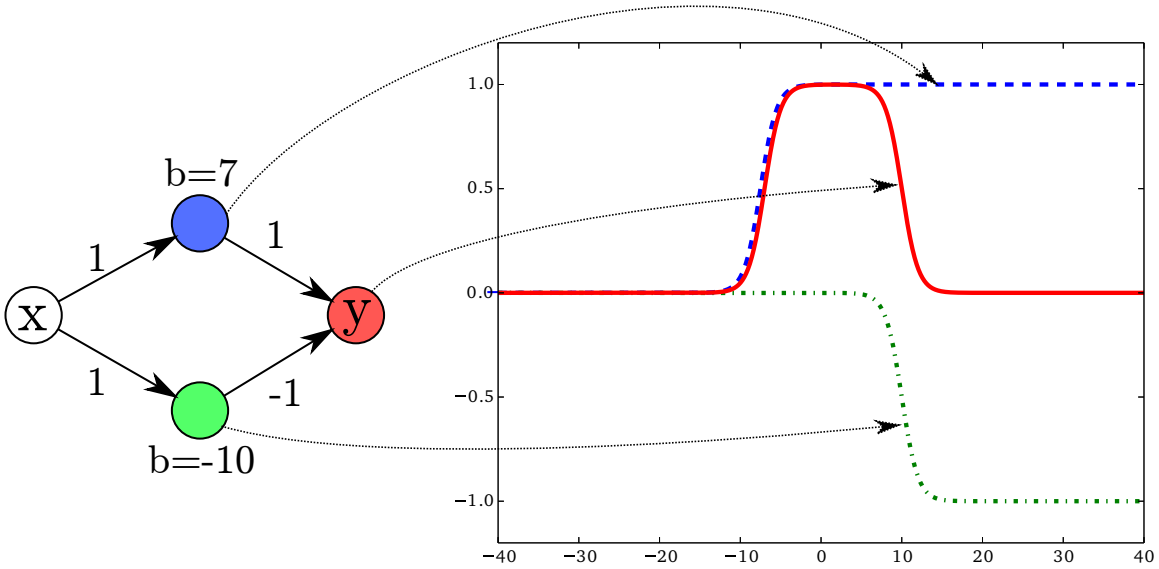


Figure 3.11 – *Illustration du principe des perceptrons à une couche cachée comme approximateurs universels. En faisant la différence entre deux sigmoïdes qui ont des biais différents, il est possible d'obtenir l'approximation d'un créneau. En multipliant le nombre de créneaux (chacun nécessitant 2 neurones cachés), il est possible de construire une approximation par palier de quasiment n'importe quelle fonction. (*Hornik *et al. 1989) généralise ce résultat à d'autres fonctions d'activation. Cette figure illustre cependant le problème pratique soulevé par ce résultat théorique : s'il est possible d'approcher n'importe quelle fonction, le nombre de neurones cachés augmente exponentiellement avec la résolution choisie, sauf dans des cas très particuliers de couples fonction d'activation/fonction à approcher (par exemple si la fonction à approcher est elle-même une sigmoïde. . .).*

3.1.4 Universalité des réseaux de neurones

Nous avons déjà vu que la linéarité des perceptrons restreint grandement les possibilités de ce réseau, ce qui leur a valu de tomber en désuétude à la fin des années 1960. L'introduction des perceptrons multicouches a permis de lever cette limitation. Les perceptrons avec une seule couche intermédiaire (ou couche cachée) sont en effet des approximateurs universels, c'est-à-dire qu'ils peuvent approcher quasiment n'importe quelle fonction avec une précision arbitrairement élevée. Ce résultat a été prouvé notamment par (Hornik et al. 1989) pour n'importe quelle fonction Borel-mesurable définie sur des espaces de dimension finie. La figure 3.11 donne l'intuition de ce résultat. Cependant, ce résultat théorique a des limitations pratiques importantes. En effet, pour une grande famille de fonctions, qui dépend notamment de la fonction d'activation choisie, le nombre de neurones nécessaires au niveau de la couche cachée croît exponentiellement avec la précision désirée, en particulier lorsque la fonction à approcher n'est pas continue.

Les réseaux profonds permettent en partie de contourner cette restriction. Ils font l'objet de la section suivante.

3.2 Apprentissage profond

Deep Belief Nets actually believe deeply in Geoff Hinton.
Yann LeCun

Dans la partie précédente, nous avons exposé le principe du perceptron multicouches ainsi que la règle d'apprentissage associée, la rétropropagation du gradient. Nous avons également rappelé que les réseaux de neurones à une seule couche cachée sont des approximateurs universels, c'est-à-dire que sous certaines hypothèses peu restrictives ils permettent d'approcher n'importe quelle fonction avec une précision arbitraire. Dans cette partie, nous allons nous intéresser aux cas des réseaux possédant un nombre de couches plus important. Nous allons tout d'abord en discuter l'intérêt, avant de voir les difficultés que pose ce type de réseaux pour l'apprentissage. Nous exposerons ensuite les techniques récentes qui ont permis à de tels réseaux de se développer et de fournir les résultats de l'état de l'art sur de nombreux domaines.

3.2.1 Intérêt des réseaux profonds

Grâce à leur couche intermédiaire, les perceptrons à une couche cachée peuvent être vus comme opérant une description des données en combinaisons de caractéristiques : chaque neurone de la couche cachée s'active pour un certain motif de valeurs des entrées et l'activation globale de la couche cachée traduit la présence d'un certain nombre de ces motifs dans la donnée en entrée. L'apprentissage dans le perceptron consiste alors à déterminer les caractéristiques qui sont pertinentes pour prédire la valeur de sortie correspondant à une entrée. Cette vision permet d'avoir une intuition de la performance médiocre des perceptrons à une seule couche cachée pour des tâches usuelles, par exemple de classification. En effet, chaque neurone de la couche cachée est indépendant des autres, ce qui oblige donc à devoir apprendre des facteurs très spécifiques (par exemple, pour classifier des visages, un neurone de la couche cachée peut se spécialiser sur la détection d'un œil à un endroit précis) afin d'avoir une réponse spécifique permettant un critère de classification précis. En effet, la décision étant prise par rapport à un seuil sur la somme des réponses de tous les neurones de la couche cachée, il faut limiter au maximum des activations résiduelles en dehors des zones de l'espace à classifier. Dès lors, pour pouvoir décrire les variabilités intra-classe, il faut d'autant plus de neurones que ceux-ci sont spécifiques, ce qui conduit à des tailles de couche cachée prohibitives et surtout à des difficultés de généralisation.

En empilant plusieurs couches cachées successives, les réseaux profonds permettent d'apprendre des caractéristiques au niveau des couches supérieures qui dépendent elles-mêmes d'autres caractéristiques au niveau des couches inférieures. Ceci permet à chaque neurone des couches inférieures d'être moins spécifique, puisque sa réponse sera croisée avec celle des autres neurones de sa couche. Ainsi, si un neurone se spécialise sur la détection d'un œil mais a toutefois une réponse peu spécifique (il répond aussi par exemple à un simple cercle), ceci introduit moins de variance dans la classification finale qu'avec une seule couche cachée, puisque les couches supérieures vont pouvoir apprendre à valider

sa réponse en la croisant avec celle d'autres neurones qui auront appris par exemple à détecter un nez et une bouche. Chaque neurone pouvant donc être moins spécifique, ceci permet d'en réduire le nombre requis pour une même performance de classification.

Nous venons de développer un argument intuitif pour expliquer l'avantage des réseaux profonds. Il existe bien évidemment des approches mathématiques pour le démontrer dans certains cadres théoriques. Par exemple (Hastad 1986) montre que certaines fonctions calculables en k couches avec un nombre polynomial de neurones nécessitent un nombre exponentiel de neurones avec $k-1$ couches. De même, dans (Pascanu et al. 2013), les auteurs étudient la finesse de la partition de l'espace rendue possible pour deux réseaux ayant le même nombre de neurones, l'un possédant une seule couche cachée et l'autre plusieurs. Le résultat principal, obtenu pour un réseau utilisant la fonction linéaire rectifiée en guise de fonction d'activation, est qu'avec kn neurones cachés et n_0 neurones en entrée, un réseau à une seule couche est capable de construire $O(k^{n_0}n^{n_0})$ régions différentes [4], alors que si les kn neurones sont répartis sur k couches de n neurones, le nombre de régions représentables évolue en $\Omega(\lfloor n/n_0 \rfloor^{k-1} n^{n_0})$. Ce dernier résultat indique en particulier qu'il peut être intéressant d'utiliser un plus grand nombre de neurones sur la première couche cachée qu'il n'y a de neurones en entrée, ce qui apporte des éléments de justification à des architectures expérimentales utilisées auparavant (par exemple (Hinton et Salakhutdinov 2006)), même si des précautions doivent être prises dans la généralisation de ce résultat, notamment car les fonctions d'activation utilisées étaient différentes.

D'un point de vue expérimental, les travaux de (Lee et al. 2009) ont effectivement montré qu'un réseau profond est capable d'apprendre des caractéristiques hiérarchiques. Entraîné sur des images de visages, de voitures et d'avions notamment, la première couche apprend des caractéristiques proches des filtres de Gabor qui sont progressivement combinées par les couches suivantes en descripteurs plus haut niveau (des yeux, des roues, des ailes par exemple) puis quasiment en prototypes de chaque classe au niveau de la couche supérieure (figure 3.12).

Une telle représentation hiérarchique permet également de factoriser efficacement les connaissances apprises par le réseau. En effet, il est par exemple possible d'utiliser la réponse d'un neurone spécifique à un œil à la fois comme information pour classifier un visage humain, mais également pour des classifieurs de têtes d'animaux variés.

3.2.2 Difficulté de l'apprentissage dans les réseaux profonds

Les problèmes posés par les réseaux profonds sont multiples. Premièrement, la rétropropagation du gradient a tendance à dégénérer lorsque le nombre de couches augmente (Bengio et al. 1994 ; Glorot et Bengio 2010). En effet, l'équation (3.13) permet de distinguer deux cas :

– lorsque $\sigma'(g_{j+1})W_{j,j+1} > 1$, l'erreur propagée croît de manière exponentielle, ce qui induit des modifications de poids elles aussi exponentielles. Ceci crée une instabilité de l'apprentissage. Cette situation peut notamment arriver si la matrice $W_{i,j}$ vient à

4. La notation $f(n) = O(n)$ signifiant $\exists c, \exists N, \forall n > N, f(n) \leq cn$ et $f(n) = \Omega(n)$ signifiant $\exists c, \exists N, \forall n > N, f(n) \geq cn$.

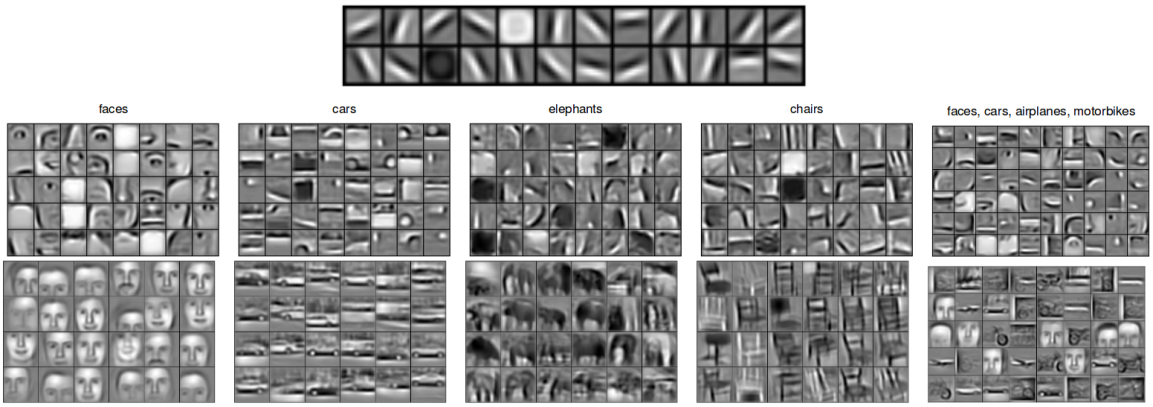


Figure 3.12 – *Représentations hiérarchiques apprises par un réseau profond entraîné sur des images de différentes catégories. La première couche apprend des caractéristiques très génériques ressemblant à des filtres de Gabor (en haut), qui sont combinées au niveau de la deuxième puis de la troisième couche en caractéristiques de plus en plus haut niveau. (Images tirées de (*Lee *et al. 2009))*

prendre des valeurs trop élevées en valeur absolue. On parle de gradient "explosif".

– lorsque $\sigma'(g_{j+1})W_{j,j+1} < 1$, l'erreur propagée décroît de manière exponentielle vers 0, ce qui ralentit très fortement l'apprentissage. Cette situation, plus courante que la précédente, arrive notamment avec l'utilisation de fonctions sigmoïdes lorsque ces fonctions sont utilisées en zone de saturation, pour lesquelles la dérivée est presque nulle. On parle de gradient "évanescent".

L'apprentissage modifiant à la fois la valeur des poids synaptiques et l'activité reçue par chaque neurone, il est difficile d'assurer un bon conditionnement du gradient au cours de l'apprentissage.

Un deuxième problème est posé par le nombre de paramètres appris. Il est en effet courant d'utiliser des réseaux contenant plusieurs milliers de connexions synaptiques : le réseau utilisé par (Le et al. 2012) possède par exemple 1 milliard de connexions. Dans le cas d'un perceptron à une seule couche, le problème d'optimisation est linéaire, et garantit donc de trouver une solution optimale si elle existe. Ce n'est plus le cas pour les réseaux multi-couches. Pendant longtemps, les piètres performances des réseaux profonds étaient justifiées par un argument avançant l'existence de très nombreux minima locaux bloquant l'apprentissage (voir par exemple (Erhan et al. 2010)). En effet, le nombre de paramètres optimisés étant très important, il est très probable, pensait-on, de tomber rapidement dans un minimum local qui bloque l'apprentissage. Cependant, il a récemment été montré qu'il était possible d'utiliser une simple rétropropagation du gradient dans un réseau profond pour obtenir des résultats de l'état de l'art (Ciresan et al. 2010). Cette réalisation a été rendue possible d'une part par un soin particulier porté à l'initialisation du réseau afin de conditionner au mieux le gradient, mais également par l'explosion récente des moyens de calculs qui permettent de compenser l'évanescence du gradient par un nombre d'itérations très élevé. Elle porte néanmoins une attaque directe contre l'argument des minima locaux, qui auraient dû empêcher d'atteindre ces performances.

Récemment, s'appuyant sur des résultats de physique statistique, il a été postulé que le problème principal n'était pas l'existence de minima locaux mais plutôt l'existence de plateaux où le gradient d'apprentissage est intrinsèquement faible (sans influence du problème de l'évanescence du gradient). En effet, les minima locaux sont en réalité très peu nombreux dans les grands espaces (Dauphin et al. 2014).

Nous ne donnerons ici qu'une intuition de ce résultat et renvoyons le lecteur vers l'article correspondant pour la démonstration mathématique (Dauphin et al. 2014). L'argument intuitif repose sur une observation assez simple : pour qu'il y ait existence d'un minimum local, il faut, condition nécessaire mais non suffisante, un point de l'espace où toutes les dérivées partielles premières s'annulent et où la matrice hessienne est positive. Plus la dimensionalité de l'espace augmente, plus le nombre de dérivées premières à considérer augmente lui aussi et plus la probabilité qu'un tel point existe est faible. De plus, lorsqu'un tel point existe tout de même, la probabilité que la matrice hessienne soit positive (c'est-à-dire que toutes ses valeurs propres soient positives) décroît elle aussi. En particulier, il suffit qu'il existe au moins une valeur propre strictement négative et une autre strictement positive pour que le point considéré soit un point-col (ou point-selle). De manière intuitive, si l'on considère une fonction quelconque, cela revient à considérer d valeurs propres aléatoires pour la hessienne en un point. La probabilité d'en avoir au moins une de chaque signe est exponentiellement plus importante lorsque d augmente que la probabilité qu'elles soient toutes positives. De ce fait, dans des espaces de grande dimensionalité, il y a exponentiellement plus de points-selles (i.e. de plateaux de gradient) que de minima locaux.

Il est à noter que, même si la densité de ces points-selles décroît avec la dimensionalité (il est toujours nécessaire que toutes les dérivées premières s'annulent), ils aimantent l'apprentissage et doivent donc réellement être pris en compte. En effet, lors d'un apprentissage par descente de gradient simple, les poids sont modifiés proportionnellement à ce gradient. Ainsi, lorsque l'un des paramètres atteint une valeur pour laquelle le gradient selon ce paramètre est presque nul, il ne sera quasiment pas modifié par les itérations suivantes. Au contraire, les autres paramètres (pour lesquels le gradient est non nul) seront plus fortement modifiés, augmentant la probabilité de tomber sur une zone dans laquelle le gradient sera faible pour eux également. Ainsi, les points-selles ont tendance à attirer les paramètres de manière séquentielle, ce qui transforme en première approximation un problème de complexité exponentielle (tomber dessus en échantillonnant l'espace aléatoirement) en un problème de complexité linéaire (trouver séquentiellement une valeur de chaque paramètre pour laquelle le gradient correspondant est nul). C'est pourquoi les méthodes récentes d'apprentissage dans les réseaux profonds s'appuyant sur une descente de gradient globale dans l'ensemble du réseau s'appuient généralement sur des méthodes inspirées de la méthode de Newton, c'est-à-dire prenant en compte la matrice Hessienne (Martens 2010 ; Dauphin et al. 2014), tout en utilisant des astuces pour éviter son calcul exact trop coûteux. Nous ne nous attarderons cependant pas sur ces méthodes que nous n'utiliserons pas par la suite, et revenons au développement historique de l'apprentissage profond.

3.2.3 L'émergence de l'apprentissage profond

En raison des difficultés d'apprentissage dans les réseaux profonds, ceux-ci ont longtemps été laissés de côté. Ils reviennent cependant sur le devant de la scène au milieu des années 2000, lorsque Geoffrey Hinton propose de les entraîner de manière incrémentale (Hinton et Salakhutdinov 2006).

Puisqu'entraîner toutes les couches en même temps pose des problèmes d'optimisation, Hinton propose de pré-entraîner chaque couche séparément afin d'amener les différents poids dans des zones pertinentes de l'espace d'apprentissage. Ainsi peut-on espérer que le réseau, une fois pré-entraîné, peut être optimisé globalement en évitant la majorité des écueils posés par un apprentissage global direct. Cependant, dès lors que chaque couche doit être pré-entraînée séparément, il faut spécifier la fonction de coût à optimiser. En effet, il est alors impossible d'utiliser le coût défini par la tâche à apprendre puisque le réseau n'est pas complet et que l'on ne peut donc pas utiliser en même temps les valeurs des entrées et des sorties attendues.

Dans (Hinton et Salakhutdinov 2006), les couches sont pré-entraînées à l'aide de l'algorithme des autoencodeurs, mais d'autres techniques comme les machines de Boltzmann restreintes peuvent aussi être utilisées (Hinton et al. 2006). La description de ces algorithmes fait l'objet de la partie suivante. La caractéristique primordiale de ces algorithmes est de fournir un critère d'entraînement non supervisé, qui vise à apprendre une bonne représentation des données présentées en entrée. L'intérêt du pré-entraînement repose alors sur l'hypothèse que les différentes couches apprendront des représentations des données en entrée sous forme de caractéristiques qui se révèleront utiles par la suite.

3.2.4 Entraîner des réseaux profonds

Nous présentons dans cette partie deux des algorithmes les plus utilisés pour le pré-entraînement des réseaux profonds : les autoencodeurs et les machines de Boltzmann restreintes. D'autres techniques existent et de nouvelles variations sont régulièrement proposées. Nous renvoyons le lecteur vers (Bengio et al. 2013) pour une revue plus complète.

Autoencodeurs

Les autoencodeurs constituent une famille assez ancienne de réseaux de neurones, aussi connus sous le nom "d'auto-associateurs" (Rumelhart et al. 1986). Comme leur nom le suggère, ces réseaux consistent à *encoder* les données en entrée dans une couche cachée, puis à les *décoder* de manière à reconstruire ces entrées (figure 3.13). La fonction de coût est alors naturellement définie par la différence entre les données fournies en entrée et leur reconstruction. Selon le type de données, les deux coûts les plus populaires sont la norme euclidienne de la différence et l'entropie croisée. En notant $\mathbf{x}$ la donnée initiale et $\widehat{\mathbf{x}}$ sa reconstruction, ces coûts sont donnés respectivement par $||\mathbf{x} - \widehat{\mathbf{x}}||^2$ et $-\sum_i (x_i \log(\hat{x}_i) + (1 - x_i) \log(1 - \hat{x}_i))$.

Ces algorithmes doivent cependant être utilisés avec précaution, car leur optimisation peut donner lieu à une solution triviale : la fonction identité. En effet, dès lors que le

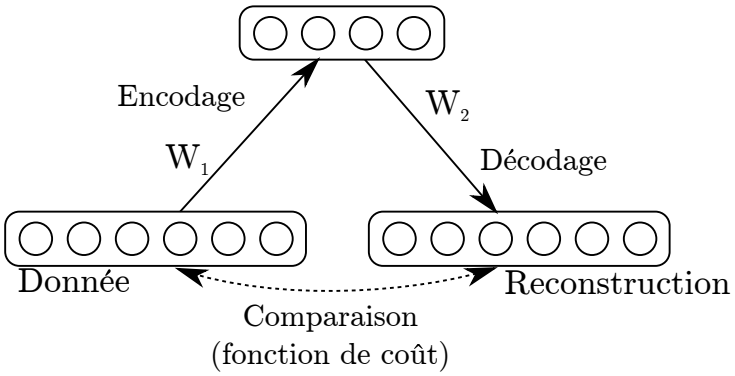


Figure 3.13 – *Principe général des autoencodeurs. Les données en entrée sont encodées dans une couche cachée, puis décodées pour en obtenir une reconstruction. Une fonction de coût peut alors être définie entre la donnée originale et sa reconstruction.*

nombre de neurones de la couche cachée est au moins égal au nombre de neurones de la couche visible, le réseau peut apprendre à recopier les données brutes, et donc obtenir une erreur de reconstruction nulle. Différentes méthodes de régularisation existent pour éviter ce phénomène (figure 3.14). Nous en présentons quelques unes dans les paragraphes suivants.

Utiliser un nombre de neurones cachés plus petit que le nombre de neurones visibles. Il s'agit de la technique la plus évidente pour empêcher le réseau d'apprendre la fonction identité. Cependant, les résultats de l'état de l'art sont généralement obtenus en utilisant un nombre de neurones cachés plus grands que le nombre de neurones visibles au niveau des premières couches. D'autres techniques de régularisation sont donc nécessaires.

Bruiter les entrées avant de les encoder (Vincent et al. 2008). Il s'agit d'une approche très populaire. Elle consiste à bruiter les données fournies en entrée avant de les encoder puis décoder, généralement avec un bruit de masquage. Le plus souvent, de l'ordre de 30% des neurones de la couche visible choisis aléatoirement sont mis à 0. L'erreur de reconstruction est alors calculée par rapport à la donnée initiale non bruitée. Une telle approche force donc le réseau à apprendre des caractéristiques globales sous la forme de relations entre l'activité de plusieurs neurones de la couche visible afin de pouvoir contrer le bruit lors de la reconstruction.

Bruiter la couche cachée avant le décodage (Hinton et al. 2012). Il s'agit d'une technique connue sous le nom de *dropout* qui consiste, comme pour l'approche précédente, à mettre à 0 une proportion importante de neurones de la couche cachée (généralement de l'ordre de 50%). Ceci a deux conséquences. Premièrement, chaque donnée est encodée par moitié moins de neurones. Deuxièmement, le réseau est obligé d'apprendre des caractéristiques indépendantes les unes des autres. En effet, celui-ci ne peut plus utiliser une activation conjointe de plusieurs neurones cachés pour représenter une caractéristique pertinente, puisque un ou plusieurs de ces neurones risquent d'être mis à 0 pour chaque

nouvelle donnée. Cette méthode revient à entraîner un sous-réseau tiré aléatoirement pour chaque donnée, et peut être mise en perspective avec la technique du *bagging* (voir par exemple (Baldi et Sadowski 2014)), où un ensemble de modèles est entraîné sur des sous-ensembles différents de données. Dans le cas du *dropout* cependant, les différents modèles partagent des sous-ensembles de paramètres. Après entraînement, il est possible d'approcher la réponse moyenne de tous les sous-réseaux en arrêtant le masquage des neurones cachés et en redimensionnant les poids de la matrice de reconstruction : si 50% des neurones cachés sont mis à 0 pendant l'entraînement, il faut diviser les poids de la matrice de reconstruction par deux lorsqu'il n'y a plus de masquage (Hinton et al. 2012).

Pénaliser les variations de la couche cachée (Rifai et al. 2011a,b). Connue sous le nom de *Contractive Autoencoders*, cette technique pénalise les variations de la couche cachée $\mathbf{h}$ pour une petite variation de la couche visible $\mathbf{v}$, sous la forme de la norme de la jacobienne $J = \frac{\partial \mathbf{h}}{\partial \mathbf{v}}$ qui est ajoutée à la fonction de coût à optimiser. Dès lors, le réseau se focalise sur les degrés de variation pertinents pour représenter les données (directions $\partial\mathbf{v}$ pour lesquelles la norme de la jacobienne sera élevée, mais nécessaire afin de pouvoir reconstruire correctement les données concernées) et non sur des variations non présentes dans l'ensemble d'apprentissage, comme par exemple des variations correspondant à l'ajout d'un bruit gaussien indépendant à chaque pixel d'une image (selon ces directions, la jacobienne pourra alors être nulle).

Imposer des contraintes de parcimonie sur l'activité de la couche cachée (Lee et al. 2006 ; Rebecchi et al. 2014). Une première méthode consiste à fixer une fréquence d'activité désirée pour chaque neurone et à pénaliser la déviation entre l'activité moyenne observée et l'activité désirée. On peut pénaliser cette déviation en ajoutant un terme à la fonction de coût à minimiser (par exemple en considérant les fréquences désirée et observée comme les moyennes de lois de Bernoulli et en pénalisant la divergence de Kullback-Leibler entre les deux lois obtenues). Cette approche permet donc d'obtenir une parcimonie au niveau de l'activité de chaque neurone. Une autre approche de la parcimonie consiste à coder chaque donnée par l'activité d'une faible proportion des neurones. Une technique consiste alors à alterner deux phases lors de l'apprentissage (Rebecchi et al. 2014). La première phase consiste à entraîner l'autoencodeur à reconstruire ses entrées de façon standard. La deuxième phase consiste à calculer l'activité de la couche cachée pour une entrée donnée, à la rendre parcimonieuse (par exemple en ne conservant que les n neurones les plus activés), puis à entraîner le réseau à produire cette représentation rendue parcimonieuse à partir de l'entrée (par rétropropagation classique).

Utiliser la même matrice de connexion pour l'encodage et le décodage (poids partagés). Cette technique n'empêche pas un réseau trop grand d'apprendre la fonction identité. Cependant, c'est une technique très utilisée de régularisation, notamment car elle divise par deux le nombre de paramètres appris dans le réseau.

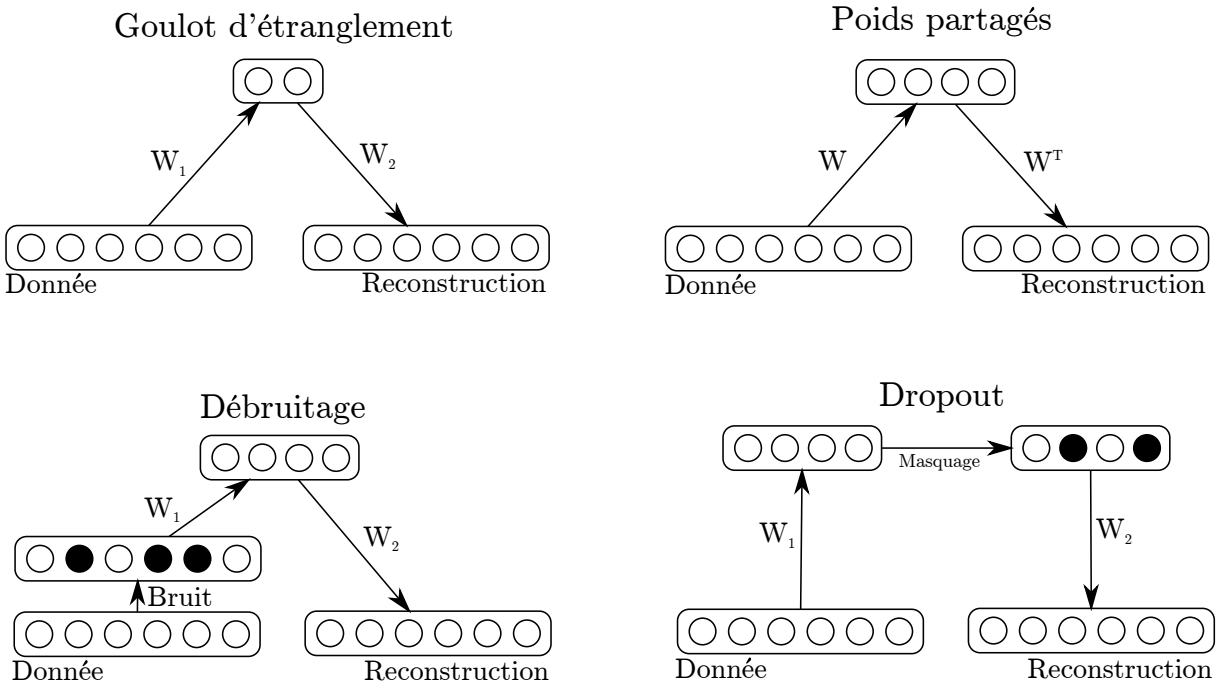


Figure 3.14 – *Illustration des techniques de régularisation les plus populaires pour les autoencodeurs. La technique du goulot d'étranglement consiste à encoder les données sur un faible nombre de neurones. La technique des poids partagés consiste à utiliser la même matrice de poids pour l'encodage et le décodage. Pour la technique du débruitage, les données sont d'abord corrompues (par exemple en mettant à 0 un certain pourcentage des neurones d'entrée), avant d'être encodées puis décodées. Le dropout enfin, consiste à mettre à 0 une certaine proportion des neurones cachés avant le décodage.*

Machines de Boltzmann restreintes

Les machines de Boltzmann restreintes (ou *Restricted Boltzmann Machines*, RBM) sont une sous-famille des machines de Boltzmann (Smolensky 1986) et reposent donc sur une approche énergétique. À la différence des machines de Boltzmann au sein desquelles tous les neurones sont connectés entre eux, les RBMs sont divisées en deux couches distinctes, l'une dite *visible* ($\mathbf{v}$) et l'autre dite *cachée* ($\mathbf{h}$). Chaque neurone de la couche visible est connecté à tous les neurones de la couche cachée, mais aucune connexion n'existe entre deux neurones d'une même couche. Cette simplification par rapport aux machines de Boltzmann permet de simplifier le calcul de l'activité des neurones. En effet, les neurones d'une couche étant alors conditionnellement indépendants étant donnée l'autre couche, l'activité de toute une couche peut être calculée en parallèle, réduisant la complexité algorithmique pour atteindre la convergence.

Bien qu'il soit possible d'utiliser différentes fonctions pour représenter la distribution de probabilité des valeurs des neurones d'une RBM (Hinton 2012), en particulier pour les neurones visibles pour lesquels une distribution gaussienne est bien adaptée pour représenter les valeurs continues observées pour les pixels d'une image par exemple, nous nous restreignons au cas le plus répandu de neurones à activité binaire.

L'énergie d'une configuration $(\mathbf{v}, \mathbf{h})$ dans une RBM s'écrit :

$$E(\mathbf{h}, \mathbf{v}) = -\mathbf{v}^\top \mathbf{b}_\mathbf{v} - \mathbf{h}^\top \mathbf{b}_\mathbf{h} - \mathbf{h}^\top W \mathbf{v}. \tag{3.31}$$

Il est possible d'échantillonner la couche cachée étant donnée la couche visible :

$$\mathbf{h} \sim \frac{1}{1 + \exp(-W\mathbf{v} - \mathbf{b}_\mathbf{h})} \tag{3.32}$$

où $x \sim p$ veut dire que la variable x est tirée selon une loi de Bernoulli de probabilité p (x prend la valeur 1 avec probabilité p et la valeur 0 avec la probabilité $1 - p$). Réciproquement, il est possible d'échantillonner la couche visible étant donnée la couche cachée par :

$$\mathbf{v} \sim \frac{1}{1 + \exp(-W^\top \mathbf{h} - \mathbf{b}_\mathbf{v})}. \tag{3.33}$$

Comme pour les machines de Boltzmann, la probabilité d'un couple $(\mathbf{v}, \mathbf{h})$ est alors donnée par :

$$p(\mathbf{h}, \mathbf{v}) = \frac{\exp(-E(\mathbf{h}, \mathbf{v}))}{Z} \tag{3.34}$$

où Z est la fonction de partition permettant de normaliser la somme des probabilités à 1 :

$$Z = \sum_{\mathbf{h},\mathbf{v}} \exp(-E(\mathbf{h}, \mathbf{v})). \tag{3.35}$$

La probabilité d'un vecteur visible $\mathbf{v}$ peut être obtenue en marginalisant sur toutes les valeurs possibles de $\mathbf{h}$

$$p(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} p(\mathbf{h}, \mathbf{v}) \tag{3.36}$$

ce qui se réécrit en introduisant l'énergie libre $\mathcal{F}(\mathbf{v})$:

$$p(\mathbf{v}) = \frac{\exp(-\mathcal{F}(\mathbf{v}))}{Z} \tag{3.37}$$

avec

$$\mathcal{F}(\mathbf{v}) = -\log \sum_{\mathbf{h}} e^{-E(\mathbf{h},\mathbf{v})} \tag{3.38}$$

et

$$Z = \sum_{\mathbf{v}} \exp(-\mathcal{F}(\mathbf{v})). \tag{3.39}$$

Grâce à l'indépendance conditionnelle obtenue par la séparation en deux couches, l'expression de l'énergie libre dans les RBMs se simplifie en :

$$\mathcal{F}(\mathbf{v}) = -{\mathbf{b}_\mathbf{v}}^\top \mathbf{v} - \sum_i \log(1 + e^{b_{h_i} + W_i \mathbf{v}}) \tag{3.40}$$

où W_i représente la $i^{\text{ème}}$ ligne de la matrice W.

Les RBM sont entraînées à maximiser la log-probabilité des données fournies en entrée (ensemble noté V) :

$$\texttt{argmax}_W \sum_{\mathbf{v} \in V} log(p(\mathbf{v})). \tag{3.41}$$

On peut montrer que (Hinton 2012)

$$-\frac{\partial \sum_V log(p(\mathbf{v}))}{\partial \theta} = \mathbb{E}\left[\frac{\partial E(\mathbf{h}, \mathbf{v})}{\partial \theta} \middle| \mathbf{v} \in V\right] - \mathbb{E}\left[\frac{\partial E(\mathbf{h}, \mathbf{v})}{\theta}\right]. \tag{3.42}$$

Le premier terme, appelé *phase positive*, fait intervenir l'espérance de l'activité de la couche cachée étant donnée la couche visible. Son calcul est donc direct en utilisant l'équation 3.32. Le deuxième terme quant à lui, appelé *phase négative*, nécessite de calculer l'espérance par rapport à la distribution représentée par la RBM. Le calcul de cette distribution n'étant pas faisable de manière exacte (notamment à cause de la fonction de partition qui nécessite de sommer sur toutes les valeurs possibles), on est obligé d'approcher ce deuxième terme (Hinton 2002). Pour ce faire, on utilise une chaîne de Markov pour tirer des échantillons selon $p(\mathbf{v})$ en effectuant un échantillonnage de Gibbs pour calculer itérativement

$$\mathbf{h}^{(t+1)} \sim p(\mathbf{h}|\mathbf{v}^{(t)}) \tag{3.43}$$

et

$$\mathbf{v}^{(t+1)} \sim p(\mathbf{v}|\mathbf{h}^{(t+1)}). \tag{3.44}$$

Après convergence, cette chaîne de Markov reproduit en effet la distribution de probabilité représentée par la RBM. Cette technique porte le nom de *Contrastive Divergence* (CD) (Hinton 2002). En pratique, on attend rarement la convergence et on effectue entre 1 et 10 itérations. On parle de CD-k où k représente le nombre d'itérations calculées.

Dans le cas CD-1, la règle d'apprentissage est alors donnée par :

$$\Delta W = \alpha \left(\mathbb{E}\left[\mathbf{v}^{(0)} \mathbf{h}^{(0)\top} \right] - \mathbb{E}\left[\mathbf{v}^{(1)} \mathbf{h}^{(1)\top} \right] \right). \tag{3.45}$$

De même que pour les autoencodeurs, différentes techniques de régularisation peuvent être utilisées pour les machines de Boltzmann restreintes (parcimonie, *dropout*, etc.). Cependant, le besoin de régularisation est moins fort que pour les autoencodeurs. En effet, le tirage aléatoire de l'activité des neurones dans une machine de Boltzmann agit comme une contrainte de régularisation forte.

Optimisation globale

Les réseaux profonds sont donc composés de plusieurs couches pré-entraînées de manière séquentielle selon la technique des autoencodeurs ou des RBMs. Ce pré-entraînement commence par la couche d'entrée puis se poursuit jusqu'à la dernière couche, chacune des couches étant entraînée en utilisant en entrée les sorties produites par la couche inférieure (il est donc possible pour chaque couche de précalculer son ensemble d'entraînement en calculant l'activité de la dernière couche entraînée pour l'ensemble de la base de données).

Différentes méthodes peuvent ensuite être utilisées pour réunir toutes les couches en un seul réseau.

Dans le cas le plus simple, toutes les couches sont mises bout à bout et le réseau résultant est considéré comme un perceptron multicouches standard. On peut alors l'utiliser pour apprendre une tâche de classification ou de régression grâce à l'algorithme de rétropropagation du gradient classique (par exemple (Hinton et Salakhutdinov 2006 ; Salakhutdinov et Hinton 2008 ; Salman et Chen 2011)).

Une autre méthode consiste à fabriquer un *Deep Belief Net* (Hinton et al. 2006) qui conserve en partie les propriétés stochastiques des RBMs. Pour un réseau de n couches, cela consiste à considérer les $n-1$ couches inférieures comme des perceptrons classiques, alors que la couche supérieure reste une RBM. Les $n-1$ couches inférieures ont alors pour rôle d'encoder une représentation des entrées (qui peut être inversée selon le principe des autoencodeurs pour reconstruire les entrées correspondantes), et la dernière couche peut échantillonner ces représentations selon la distribution de probabilité de l'ensemble d'apprentissage.

Une dernière méthode consiste à réaliser une machine de Boltzmann profonde (Salakhutdinov et Hinton 2009). Toutes les couches restent alors des RBMs. Dans ce cas, les neurones d'une couche intermédiaire $\mathbf{h_l}$ reçoivent en entrée les activations des couches $\mathbf{h_{l-1}}$ et $\mathbf{h_{l+1}}$. Les auteurs de (Salakhutdinov et Hinton 2009) ont montré que chaque couche recevant ainsi deux fois plus de signaux en entrée que lors de son pré-entraînement, il faut diviser la valeurs des poids synaptiques appris par deux.

Pour la suite, nous laissons de côté les machines de Boltzmann pour nous concentrer sur les autoencodeurs.

Comme nous l'avons vu, les autoencodeurs profonds peuvent être pré-entraînés couche par couche avant de les concaténer en un seul réseau. Le réseau ainsi obtenu peut alors être à son tour optimisé comme un autoencodeur classique (on calcule l'activité de la

couche supérieure induite par une entrée donnée, avant de la décoder en la faisant passer l'information en sens inverse dans le réseau, ce qui permet de calculer une erreur de reconstruction qu'il est possible de minimiser par descente de gradient) ou comme un perceptron multicouches classique. Dans ce cas, la couche de sortie est généralement utilisée comme entrée d'un algorithme classique (par exemple perceptron ou SVM, (Stuhlsatz et al. 2010)) et on peut rétropropager la différence entre la sortie finale obtenue et la sortie désirée dans l'ensemble du réseau.

À la différence des RBMs stochastiques, les autoencodeurs sont généralement déterministes (à l'exception des techniques de régularisation à base de bruitage). Ils sont donc assez mal appropriés pour re-générer des entrées selon la distribution de probabilité de leur ensemble d'apprentissage. Néanmoins, de récentes techniques se placent à mi-chemin entre autoencodeurs et RBMs, dotant les premiers de propriétés génératives plus importantes (Bengio et Thibodeau-Laufer 2013).

Chapitre 4

Des capteurs aux concepts

Misérable raison c'est de nous [les sens] que tu tires les éléments de ta croyance, et tu prétends nous réfuter ! Tu te terrasses toi-même en prétendant nous réfuter.

Democrite

Sommaire

Des capteurs aux concepts est un titre à la fois ambitieux et polémique. En effet, la définition même de la notion de *concept* est sujette à de nombreuses controverses (voir par exemple l'introduction de (Barsalou 1999) et la deuxième partie de (Ghadakpour 2002)). Notre but n'est pas ici d'y prendre part, et notre acception de concept s'inspirera fortement des travaux de Gärdenfors qui a introduit la théorie des *espaces conceptuels* (Gardenfors 2004). Ses travaux partent de la remarque qu'il existe un très grand fossé entre d'une part les modèles de représentations symboliques, qui présentent

d'importantes limitations comme le problème de l'ancrage des symboles dont nous avons parlé en introduction de cette thèse, et les modèles associatifs tels le connexionisme, qui manquent selon lui cruellement de mécanismes pour traiter l'information à un haut niveau conceptuel. Nous nous inspirons également de (Barsalou 1999) qui donne un rôle prépondérant à la perception dans la formation de symboles.

En nous appuyant sur l'hypothèse des sous-variétés présentée dans le chapitre 2, nous proposons une définition des concepts comme des sous-variétés d'un espace perceptif, lorsque Gärdenfors propose quant à lui une définition de concept en tant que régions convexes dans un espace particulier dont les dimensions sont appelées *qualités*. L'appartenance ou non à une région donnée de l'espace permet donc de définir un aspect symbolique pour ces représentations, mais dans les deux cas les concepts sont plus que des symboles, puisqu'ils sont représentés par un ensemble de points qui peuvent être paramétrés soit le long de sous-variétés, soit dans l'espace des qualités. La comparaison s'arrête toutefois là, puisqu'alors que Gärdenfors s'appuie sur des qualités définies *a priori*, telle la taille et le poids, notre but est de proposer un apprentissage autonome de ces sous-variétés sans en spécifier à l'avance les dimensions pertinentes.

Lorsque nous utilisons le mot *concept*, nous n'entendons donc par là qu'une représentation semi-symbolique de la perception ou de l'action qui vise à combler le fossé entre d'une part des représentations associatives entièrement distribuées et des représentations purement symboliques abstraites des stimuli physiques correspondants. Nous ne nous attachons pas à leur dimensions cognitives telle la faculté de raisonnement.

Dans une première partie, nous dressons l'état de l'art en relation avec notre approche sur l'apprentissage de variétés. Nous présentons l'architecture proposée et les expériences associées dans un second temps. Les travaux décrits dans ce chapitre correspondent à l'article

> Alain Droniou, Serena Ivaldi et Olivier Sigaud (2014, in press). « Deep unsupervised network for multimodal perception, representation and classification ». Dans : *Robotics and Autonomous Systems*.

4.1 Structuration du flux sensoriel

Dans le chapitre 2, nous avons développé l'hypothèse des sous-variétés pour proposer un cadre théorique à l'apprentissage de concepts à partir d'informations brutes dans des espaces de grande dimension. En nous appuyant sur cette hypothèse, nous proposons dans ce chapitre un algorithme permettant de représenter un flux sensoriel sous forme de sous-variétés distinctes. Avant de présenter cette architecture et les résultats obtenus, nous allons tout d'abord rappeler les différentes méthodes qui ont été proposées d'une part pour l'apprentissage de variétés et la réduction de la dimensionalité, d'autre part pour l'apprentissage de représentations symboliques et enfin pour le traitement d'informations multimodales.

4.1.1 Apprentissage de variétés et réduction de la dimensionalité

L'apprentissage de variétés a été l'objet de nombreuses études. En effet, les variétés constituent un outil intéressant pour réduire efficacement la dimensionalité des données, afin d'une part de lutter contre la malédiction de la dimensionalité à laquelle sont sujets de nombreux algorithmes, afin d'autre part de fournir une méthode efficace pour produire une visualisation pratique des données.

Représenter des données de grande dimension dans des espaces de plus faible dimensionalité revient à projeter ces données dans une base de caractéristiques extraites de l'espace initial et construites à partir des données, c'est-à-dire à trouver un nouveau codage de ces données. Différentes méthodes ont été proposées, imposant différentes contraintes sur ce codage. Ainsi, l'analyse en composantes principales correspond à un codage linéaire des entrées en cherchant les directions de l'espace pour lesquelles la variance des données est maximale (Hotelling 1933). Les algorithmes de clustering, quant à eux, codent les données selon leur proximité par rapport à certains points de l'espace initial, ce qui revient à projeter des sous-espaces convexes sur des variétés de dimension 0. Les techniques de codage parcimonieux (Olshausen et Field 1997 ; Lee et al. 2006) apprennent une base de caractéristiques dans laquelle chaque point peut-être exprimé à l'aide d'un nombre réduit de ces caractéristiques. Si chaque point est donc caractérisé par un faible nombre de coordonnées non nulles, l'espace global formé à partir de toutes les caractéristiques apprises n'est cependant pas contraint à être de faible dimensionalité. D'autres contraintes de codage, comme la positivité des coordonnées, peuvent également être utilisées (par exemple pour la factorisation en matrice non négative (Lee et Seung 2001)). L'intérêt de ces techniques est de pouvoir calculer assez facilement la projection d'un nouveau point dans l'espace des caractéristiques.

De nombreuses autres techniques ont été développées (Cayton 2005), mais comme souligné dans (Bengio et al. 2013), beaucoup d'entre elles s'appuient sur la notion de voisinage de chaque point afin de définir un critère à optimiser dans l'espace d'arrivée (par exemple la conservation des distances entre plus proches voisins entre l'espace initial et l'espace final). L'utilisation intensive de ce voisinage aboutit souvent à des algorithmes ne permettant pas de calculer facilement les coordonnées d'un nouveau point dans le nouvel espace (Bengio et al. 2013) et nécessitent des algorithmes complexes (Kouropteva et al. 2005 ; Law et Jain 2006 ; Zhao et al. 2006). C'est pourquoi ce genre de techniques est généralement réservé à la visualisation de données, problème pour lequel toutes les données sont connues dès le début. Par ailleurs, l'utilisation de la notion de voisinage peut être problématique dans les espaces de grande dimensionalité, comme nous l'avons vu au chapitre 2 et requiert de prendre en compte un ensemble de couples de points, dont le nombre croît de manière quadratique avec le nombre de points.

Réduction de la dimensionalité et réseaux profonds

Les réseaux profonds ont été largement utilisés comme technique de réduction de la dimensionalité. Un de leurs avantages principaux est de fournir une projection explicite des données dans le nouvel espace (à la différence par exemple de la factorisation en

matrice non-négative qui définit la projection de manière implicite à travers un problème de minimisation de distance) : il suffit de projeter une donnée de la couche d'entrée jusqu'à la couche de sortie.

De plus, comme nous l'avons vu au chapitre 3, les réseaux profonds ont la particularité intéressante de pouvoir apprendre une décomposition hiérarchique des données (Lee et al. 2009). Leur capacité à extraire des dimensions pertinentes a été démontrée notamment dans (Hinton et Salakhutdinov 2006), où les auteurs obtiennent de manière non supervisée une représentation des chiffres manuscrits (de 0 à 9) séparant assez nettement chacune des dix classes, cela dans un espace réduit à deux dimensions.

Parmi les algorithmes de base de l'apprentissage profond exposés chapitre 3, certains favorisent explicitement l'apprentissage le long des dimensions les plus pertinentes. C'est le cas par exemple des *contractive autoencoders*. Ceux-ci sont notamment utilisés par (Rifai et al. 2011c) pour extraire un ensemble de plans tangents aux données fournies en entrée en s'appuyant sur la Jacobienne $J = \frac{\partial h}{\partial v}$ pour rechercher les directions selon lesquelles une légère variation de l'entrée n'induit pas de modification de la couche cachée.

Dans (Reed et Lee 2013), les auteurs proposent également un réseau capable de démêler des facteurs de variation des données, ce qui permet ensuite de parcourir la variété définie par les données de l'ensemble d'apprentissage en fixant certains facteurs de variation tout en faisant varier les autres. Leur réseau utilise toutefois des neurones à activation binaire, ce qui définit la variété à travers les sommets d'un hypercube. De plus, le réseau apprend une variété globale représentant tout l'ensemble d'apprentissage, sans distinguer différentes classes (ou sous-variétés).

4.1.2 Représentations symboliques

Si les réseaux profonds ont été appliqués avec succès pour apprendre une représentation des données sous la forme d'une variété, peu de travaux à notre connaissance les ont utilisés pour distinguer différentes sous-variétés de manière non supervisée. Les représentations parcimonieuses peuvent être considérées comme un premier pas dans cette direction : chaque donnée étant représentée par un faible nombre de caractéristiques, il peut éventuellement exister des regroupements de caractéristiques en sous-ensembles distincts utilisés par différentes familles de données. Une telle propriété n'est cependant pas garantie.

Une telle distinction en plusieurs sous-variétés amène toutefois à considérer les algorithmes de clustering. La plupart des algorithmes proposés (Jain et al. 1999) sont des variantes de l'algorithme des k-moyennes (Macqueen 1967) ou du clustering hiérarchique (Ward 1963). Les cartes auto-organisatrices par exemple fonctionnent selon un principe très proche de l'algorithme des k-moyennes (Bação et al. 2005). Ces algorithmes s'appuient sur les distances entre deux points pour représenter leur similarité et distinguer différents sous-ensembles. La difficulté principale de ces algorithmes consiste alors à définir une métrique adéquate selon la tâche considérée (Xing et al. 2002). En effet, pour des données redondantes dans des espaces de grande dimensionalité (par exemple des images), des métriques simples comme la distance euclidienne reflètent de manière très médiocre la similarité entre deux points. C'est pourquoi la plupart des approches utilisant

un algorithme de clustering utilisent auparavant un autre algorithme d'extraction de caractéristiques sur lesquelles des distances simples peuvent être plus pertinentes que sur les données brutes. Une approche populaire consiste par exemple à utiliser de telles métriques sur des "sacs de caractéristiques" (*bags of features*) (O'Hara et Draper 2011).

Nous nous appuierons dans notre cas sur les autoencodeurs pour l'extraction de caractéristiques. L'intérêt des réseaux profonds pour la classification a en effet été démontré à de multiples reprises, en utilisant néanmoins des algorithmes dédiés et supervisés pour l'étape finale de classification (voir par exemple (Stuhlsatz et al. 2010 ; Salakhutdinov et al. 2011)). L'utilisation de réseaux *gated* a également été proposée par (Memisevic et al. 2010), en utilisant une variante énergétique (de type RBM) des réseaux *gated* présentés chapitre 3. L'utilisation de deux couches pour représenter les données permet de distinguer une couche chargée de représenter la classe tandis que l'autre permet de paramétriser chacune des classes ainsi définies. Le travail présenté dans (Memisevic et al. 2010) est toutefois limité à un cadre supervisé (l'entraînement du réseau nécessite qu'on lui fournisse les labels de chaque exemple) et chaque classe est paramétrée par des variables booléennes, qui peuvent être vue comme codant la présence ou l'absence de caractéristiques apprises. Nous nous inspirerons de cette architecture pour notre travail (section 4.2), en gardant la distinction entre une couche de classification, qui utilisera pour cela une fonction d'activation *softmax* et une couche de paramétrisation, pour laquelle nous utiliserons une fonction *softplus* qui nous permettra d'apprendre une paramétrisation continue de chaque classe.

4.1.3 Multimodalité

Nous avons également évoqué en introduction de cette thèse ainsi qu'au chapitre 2 l'importance de la multimodalité. Plusieurs travaux l'ont abordée dans le cadre de l'apprentissage de représentations.

Dans (Mangin et Oudeyer 2013), une représentation conjointe de gestes et de phrases est apprise par factorisation en matrices non-négatives. Les auteurs montrent que la représentation apprise peut-être utilisée pour retrouver une modalité étant donnée l'autre (par exemple retrouver le geste correspondant à une phrase). En testant l'information mutuelle entre les caractéristiques apprises et des labels sémantiques fournis de manière supervisée, ils montrent également que les représentations apprises ont intégré un contenu sémantique. Ceci permet notamment de catégoriser efficacement les données, mais nécessite toutefois l'utilisation de labels supervisés.

Dans (Morse et al. 2010a,b ; Lefort et al. 2010), une règle d'apprentissage hebbienne est utilisée pour associer les activations de neurones dans plusieurs cartes auto-organisatrices, s'inspirant notamment de la théorie des zones de convergence-divergence (décrite section 2.3.2). Ces associations peuvent être utilisées pour influer sur l'apprentissage de chacune des cartes (Lefort et al. 2010, 2014). L'utilisation de cartes auto-organisatrices pour l'apprentissage d'associations multimodales a été étudiée dans de nombreux travaux (par exemple (Wermter et al. 2004 ; Papliński et Gustafsson 2005 ; Johnsson et al. 2009 ; Ridge et al. 2010 ; Vavrečka et Farkaš 2013 ; Lallee et Dominey 2013)). Cependant, si ces architectures peuvent facilement apprendre des

associations multimodales, elles souffrent de la malédiction de la dimensionalité : il est difficile de projeter des données de grande dimension dans des espaces à deux ou trois dimensions tout en préservant la topologie locale, alors que le respect de cette topologie est au cœur même de la règle d'association multimodale. Ainsi, une hiérarchie de cartes auto-organisatrices est utilisée dans (Lallee et Dominey 2013) pour réduire la dimensionalité des entrées, puis les coordonnées des neurones les plus actifs dans chaque carte monomodale sont utilisées comme entrées de cartes multimodales. Dans ce cas, deux stimuli similaires dans une modalité doivent être représentés par deux neurones assez proches pour que l'association soit renforcée, ce qui peut se révéler problématique lorsque le phénomène physique sous-jacent est hautement dimensionnel. Il faut d'ailleurs noter qu'en utilisant des cartes auto-organisatrices en trois dimensions pour une expérience de *pierre - feuille - ciseaux* pour laquelle l'apparence de la main du robot peut elle-même être définie dans un espace en trois dimensions (malgré ses 9 degrés de liberté), les auteurs se placent dans le cas idéal pour leur architecture.

De même, utilisant un codage de type carte auto-organisatrices, (De Sa et Ballard 1997, 1998) dérivent une règle d'apprentissage hebbienne à partir d'une théorie de "minimisation du désaccord" (*disagreement minimization framework*). Dans ce cadre, chaque modalité est codée par un dictionnaire, puis catégorisée par un classifieur linéaire (initialisé par un clustering non supervisé du dictionnaire). La catégorie prédite pour chaque exemple dans une modalité est ensuite utilisée pour entraîner le classifieur de l'autre modalité. Ceci conduit donc à un apprentissage conjoint de catégories multimodales, les classifieurs apprenant petit à petit à prédire les mêmes catégories. Appliqué à une tâche de catégorisation de stimuli audio-visuels (mouvement des lèvres et sons produits pour différentes syllabes), les auteurs montrent en particulier que l'utilisation de deux modalités améliore la classification obtenue pour chaque modalité considérée isolément. Cependant, l'utilisation de cartes auto-organisatrices implique les mêmes faiblesses que précédemment. Les auteurs expliquent d'ailleurs que plusieurs réinitialisations des classifieurs après modification des dictionnaires sont nécessaires pour atteindre de bonnes performances.

L'apprentissage d'une distribution de probabilité conjointe à plusieurs modalités a aussi été exploré par (Nakamura et al. 2009) en utilisant des allocations de Dirichlet latentes comme modèles probabilistes. Comme pour les travaux précédents, l'ajout de modalités permet d'obtenir des catégorisations plus proches de celles effectuées par l'humain. Ces travaux sont toutefois limités à des espaces de faible dimensionalité qui impliquent souvent de travailler sur des caractéristiques extraites des flux sensoriels définies manuellement. Ils sont étendus dans (Nakamura et al. 2011) pour faire face à des structures de catégorisation plus complexes, par exemple l'appartenance d'un objet à plusieurs catégories (telles *jouet* et *mou*).

Enfin, l'utilisation de réseaux profonds dans des cadres multimodaux a également été étudiée, notamment par (Ngiam et al. 2011) pour apprendre une représentation conjointe des mouvements des lèvres et des sons perçus pour différentes syllabes. En plus de pouvoir reconstruire le signal d'une modalité étant donné le signal d'une autre modalité, les auteurs montrent qu'un tel réseau peut reproduire l'effet McGurk (Mcgurck et Macdonald 1976). Après avoir entraîné le réseau de manière non supervisée, un classifieur est entraîné de manière supervisée à distinguer les syllabes `ba`, `ga` et `da` à partir des représentations

apprises dans le domaine audio-visuel. En testant par la suite le classifieur sur un stimulus hybride correspondant à un `ga` visuel couplé à un `ba` auditif, la catégorie prédite est un `da` la plupart du temps, comme observé chez l'humain.

4.1.4 Synthèse

De nombreux algorithmes ont été proposés pour traiter des données de grande dimensionalité. D'une part, les techniques de réduction de la dimensionalité permettent d'obtenir une représentation compressée des données de manière non supervisée, mais sans fournir de représentation symbolique potentiellement utilisable par des algorithmes de raisonnement et de planification. De l'autre côté, les techniques de clustering sont mal adaptées à des espaces de dimensionalité importante lorsqu'elles utilisent des métriques simples, mais peuvent être utilisées de manière efficace sur les sorties produites par des algorithmes de réduction de la dimensionalité. L'utilisation successive de plusieurs algorithmes soulève toutefois le problème des synergies, notamment à travers l'influence que le clustering peut avoir sur les dimensions pertinentes pour la réduction de la dimensionalité, et *vice versa*, et contraint grandement les interactions descendantes qui requièrent alors l'introduction de techniques dédiées.

L'utilisation de réseaux de neurones est par ailleurs intéressante. D'un côté, les réseaux profonds fournissent une technique efficace de réduction de la dimensionalité, tandis que le clustering peut être obtenu en introduisant des processus de compétition entre neurones (comme dans le cas des cartes auto-organisatrices). De plus, les réseaux de neurones étant agnostiques vis-à-vis de la nature des signaux traités, différentes modalités peuvent être traitées par un même réseau, ce qui les dote naturellement de propriétés multimodales.

4.2 Architecture

Cette section est consacrée à la description de l'architecture proposée. Dans un premier temps, nous allons décrire une version monomodale de l'architecture, avant de présenter son extension au cas de plusieurs modalités.

4.2.1 Réseau monomodal

Le réseau proposé est illustré figure 4.1. La représentation des données se sépare en deux sous-parties distinctes : une couche de neurones *concepts* sous la forme d'une couche dont la fonction d'activation est la fonction *softmax* et dont le but est d'obtenir une représentation symbolique des données, et une couche de paramétrisation dont la fonction d'activation est la fonction *softplus*. Ces deux couches interagissent par l'intermédiaire d'une connexion *gated*. Celle-ci permet de paramétrer une sous-variété différente pour chaque neurone de la couche *concept* tout en ayant un partage efficace des ressources parmi toutes les sous-variétés. Nous nous réfèrerons par la suite à ces deux couches sous les noms "couche *softmax*" et "couche *softplus*". Grâce à l'utilisation de réseaux de neurones, l'espace d'entrée est arbitraire : il peut s'agir de n'importe quelle modalité, ayant éventuellement

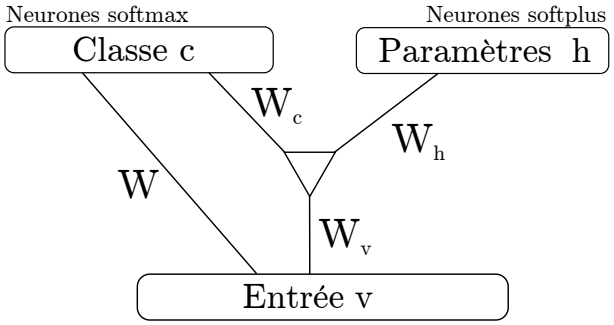


Figure 4.1 – *Architecture du réseau monomodal pour l'apprentissage de sous-variétés. Le réseau classifie les données en entrée en les projetant à l'aide de la matrice W sur une couche softmax. Cette classification est utilisée par un réseau gated qui apprend à représenter la sous-variété correspondant à chaque catégorie.*

subie plusieurs traitements auparavant. En particulier, il est possible d'utiliser en entrée la représentation haut niveau apprise par la couche supérieure d'un réseau profond, par exemple dans le cas où les entrées sont des images.

Ce réseau est appris selon le principe de l'entraînement non-supervisé de l'apprentissage profond. Dans notre travail, nous nous appuyons plus spécifiquement sur le principe des autoencodeurs dont le but est de minimiser l'erreur de reconstruction des stimuli présentés en entrée, comme expliqué dans le chapitre précédent.

Un stimulus $\mathbf{v}$ présenté en entrée subit tout d'abord une catégorisation à travers sa projection sur la couche *softmax* $\mathbf{c}$:

$$\mathbf{c} = \sigma_{max}(W\mathbf{v}) \tag{4.1}$$

où

$$\sigma^i_{max}(\mathbf{x} = (x_1, .., x_n)) = \frac{e^{x_i}}{\sum_j e^{x_j}}. \tag{4.2}$$

Ensuite, cette catégorisation est utilisée pour projeter le stimulus sur la couche de paramétrisation $\mathbf{h}$:

$$\mathbf{h} = \sigma_+(W_h(W_c\mathbf{c}) * (W_v\mathbf{v}))). \tag{4.3}$$

Cette dernière projection implique une interaction multiplicative entre l'entrée et la couche de neurones *softmax* dont le rôle est de déformer l'espace des facteurs calculés à partir des entrées en fonction du "concept" retenu : certains facteurs calculés à partir de l'entrée (qui peuvent être vus comme des caractéristiques extraites) peuvent être plus ou moins pertinents selon le "concept" considéré. L'interaction multiplicative permet alors de pondérer l'importance de ces facteurs (éventuellement par un poids nul) afin que la couche *softplus* puisse se focaliser sur les seuls facteurs pertinents. Ceci permet donc à la couche de paramétrisation d'apprendre à représenter les variations pertinentes de ce stimuli par rapport au "concept" de base.

Suivant le principe des autoencodeurs et des connexions *gated*, cette représentation *concept* $\times$ *paramétrisation* peut être projetée en sens inverse afin d'obtenir une reconstruction de l'entrée :

$$\widehat{\mathbf{v}} = \sigma(W_v^T(W_c\mathbf{c}) * (W_h^T\mathbf{h}))). \tag{4.4}$$

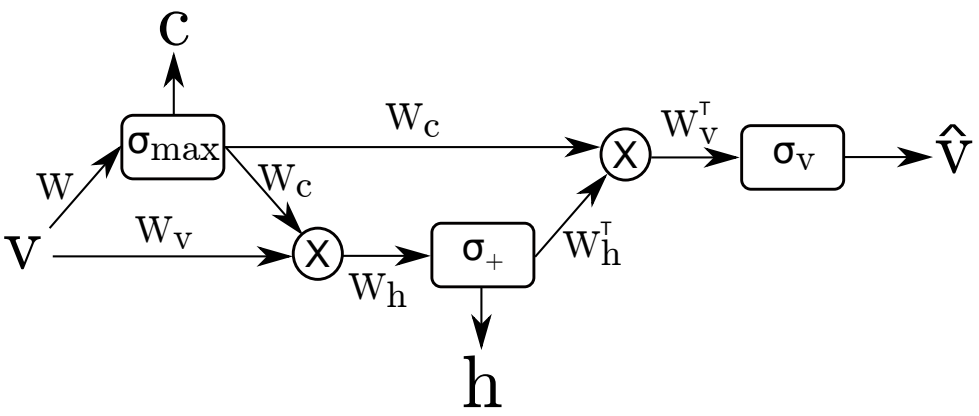


Figure 4.2 – *Flot de calculs dans le réseau monomodal pour calculer la reconstruction d'une entrée.*

L'entraînement du réseau consiste alors à minimiser l'erreur $||\widehat{\mathbf{v}} - \mathbf{v}||^2$ par une descente de gradient classique. Il est important de noter que dans le cas où l'entrée du réseau correspond à la couche supérieure d'un autre réseau, cette descente de gradient peut se propager dans le reste de ce réseau.

Étant donnée une entrée $\mathbf{v}$, sa reconstruction globale est obtenue en combinant les équations 4.1, 4.3 et 4.4, soit (figure 4.2) :

$$\widehat{\mathbf{v}} = \sigma_v(W_v^T(W_c \underbrace{\sigma_{max}(W\mathbf{v})}_{\mathbf{c}}) * (W_h^T \underbrace{\sigma_+(W_h(W_c \underbrace{\sigma_{max}(W\mathbf{v})}_{\mathbf{c}}) * (W_v\mathbf{v}))}_{\mathbf{h}}))). \tag{4.5}$$

L'apprentissage vise donc à la fois à sélectionner les caractéristiques de l'entrée spécifiques à chaque concept (à travers la matrice W), mais également à apprendre les degrés de liberté pertinents pour chaque concept (à travers les matrices W_c, W_h et W_v).

Régularisation

Étant donnée l'utilisation d'une fonction *softmax* pour réaliser la catégorisation de l'entrée, le réseau peut apprendre une représentation distribuée de chaque entrée en répartissant l'activation totale parmi tous les neurones. Pour empêcher ce phénomène, nous ajoutons un bruit gaussien η à l'activité de cette couche (bruit gaussien indépendant pour chaque neurone de la couche) avant d'appliquer la fonction *softmax* :

$$\mathbf{c} = \sigma_{max}(W\mathbf{v} + \eta). \tag{4.6}$$

Ainsi, le réseau ne peut pas s'appuyer sur une combinaison précise d'activations de tous les neurones pour représenter différentes entrées, mais doit au contraire sélectionner vivement un seul neurone afin de contrer le bruit ajouté. L'influence de ce bruit de régularisation sera détaillée dans la section 4.3.2.

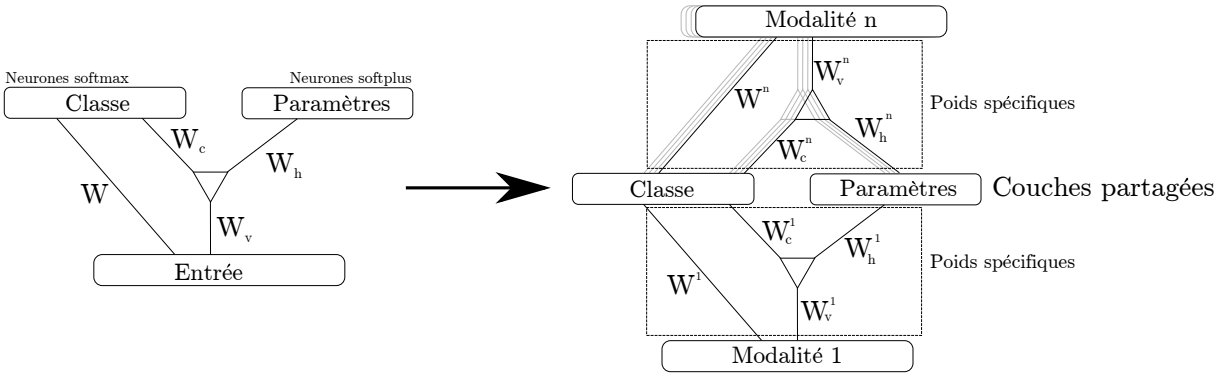


Figure 4.3 – *Architecture du réseau multimodal pour l'apprentissage de sous-variétés. L'architecture se généralise à un nombre arbitraire de modalités, en dupliquant les couches d'entrée tout en partageant les couches softmax et softplus. Sur la figure, la modalité 1 est séparée des autres modalités dans un souci de clarté et ne traduit aucunement une distinction entre les différentes modalités.*

4.2.2 Réseau multimodal

Le réseau monomodal peut se généraliser directement au cas de plusieurs modalités : il suffit pour cela de partager les deux couches *softplus* et *softmax* entre toutes les modalités, comme illustré figure 4.3. Ceci force le réseau à apprendre une représentation conjointe des différentes modalités. Puisque la couche *softmax* force l'activation d'un seul neurone à la fois, le réseau est amené à associer les entrées de plusieurs modalités à un même concept. Cependant, les neurones de la couche de paramétrisation peuvent se spécialiser de manière plus ou moins indépendante sur la représentation des détails de l'une ou l'autre des modalités, afin de donner plus de flexibilité au réseau. Ainsi, une image de chat et l'écoute du mot "chat" sont associées à la même catégorie, mais il n'y a aucune raison de déduire la prononciation du mot à partir de l'apparence de l'image. En revanche, une bonne représentation de la trajectoire articulaire permettant d'écrire un chiffre peut quant à elle être déduite d'une image de ce chiffre, et *vice versa*.

Régularisation multimodale

Lorsque l'une des modalités est plus bruitée ou irrégulière qu'une autre, le réseau peut apprendre à classifier les entrées en s'appuyant sur une seule des modalités. Après apprentissage, le réseau est alors incapable d'exploiter une entrée partielle pour laquelle une seule modalité est disponible. Pour éviter un tel comportement, nous pondérons chaque modalité au niveau de la couche *softmax* de manière indépendante et aléatoire pour chaque classe et pour chaque exemple d'entraînement. Dans le cas de deux modalités, l'activité de la couche *softmax* devient :

$$\mathbf{c} = \sigma_{max}\left(\Omega * (W_1\mathbf{v_1}) + (1-\Omega) * (W_2\mathbf{v_2}) + \eta\right) \tag{4.7}$$

où Ω correspond à un vecteur de nombres aléatoires indépendants et uniformes entre 0 et 1, $\mathbf{v_1}$ et $\mathbf{v_2}$ aux entrées de chacune des modalités, W_1 et W_2 aux matrices de classification

correspondantes et η au bruit de régularisation introduit dans la section précédente.

Cette pondération aléatoire empêche le réseau de s'appuyer sur une seule modalité, dont le poids peut être nul pour quelques exemples. De plus, puisque le réseau est entraîné à reconstruire correctement les deux modalités, il est forcé de se comporter de la même manière quel que soit le poids de chaque modalité, en particulier lorsqu'une modalité est éliminée par un poids nul, ou lorsque les deux modalités ont le même poids (de 0.5). Le réseau doit donc apprendre des matrices de classification W_1 et W_2 qui produisent autant que faire se peut une projection semblable des deux modalités au niveau de la couche *softmax*.

Ce mécanisme s'étend directement au cas de n modalités, en considérant n vecteurs aléatoires dont la somme est normalisée de manière à ce que chaque terme soit égal à 1.

4.3 Expériences

Nous allons illustrer les capacités de l'architecture proposée en deux temps. Premièrement, nous menons une analyse détaillée du réseau monomodal afin d'étudier l'influence des différents paramètres et de visualiser les sous-variétés apprises. Ensuite, nous étudions les performances du réseau multimodal en utilisant deux, puis trois modalités. Nous étudions l'influence du nombre de modalités et illustrons le comportement du réseau lorsqu'une information partielle est fournie en entrée.

4.3.1 Entraînement du réseau

Dans toutes les expériences, nous encodons chaque modalité à l'aide d'un autoencodeur monocouche, au-dessus duquel nous ajoutons l'architecture décrite précédemment (voir figure 4.4). L'ajout de ces autoencodeurs vise à illustrer la capacité du réseau proposé à gérer des entrées provenant d'autres réseaux, la simplicité des stimuli traités par la suite ne justifiant pas d'utiliser des réseaux plus complexes que de simples autoencodeurs. Le réseau global est entraîné de manière incrémentale :

- les autoencodeurs de chaque modalité sont entraînés pour 3000 pas de temps ;
- le réseau d'apprentissage de sous-variétés est entraîné pour 3000 nouveaux pas de temps ;
- le réseau global est entraîné à reconstruire ses entrées brutes pour 4000 pas de temps.

Le réseau est entraîné par descente de gradient, avec un taux d'apprentissage de 0.001 pour les matrices de poids et 0.0001 pour les biais et un momentum de 0.9. Ces paramètres, ainsi que la durée de chacune des phases d'apprentissage, ont été choisis expérimentalement de telle sorte que l'apprentissage ne présente pas d'instabilité et que l'erreur de reconstruction atteigne un plateau à la fin de chaque phase. Ils n'ont donc pas été l'objet d'une optimisation poussée.

Pour encoder chaque modalité, nous utilisons des autoencodeurs régularisés par débruitage, pour lesquels chaque modalité est corrompue par un bruit de masquage de 30% (30% des neurones d'entrée choisis aléatoirement sont mis à 0).

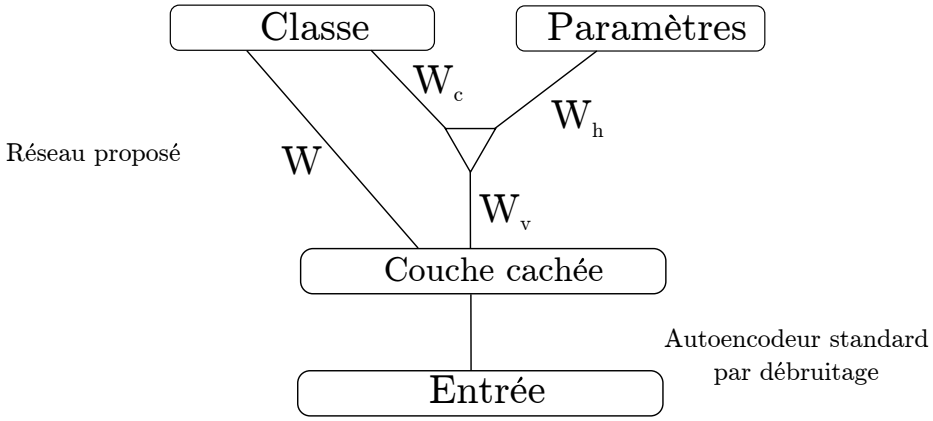


Figure 4.4 – *Architecture du réseau utilisé pour les expériences (une seule modalité est représentée). Les données sont tout d'abord encodées par un autoencodeur classique régularisé par débruitage, dont la sortie est ensuite utilisée comme entrée du réseau présenté dans les sections précédentes.*

Sauf mention contraire explicite, nous utilisons 10 neurones *softmax*, 2 neurones *softplus* et pour chaque modalité la taille de la couche de facteurs est égale à la taille de la couche de sortie de l'autoencodeur.

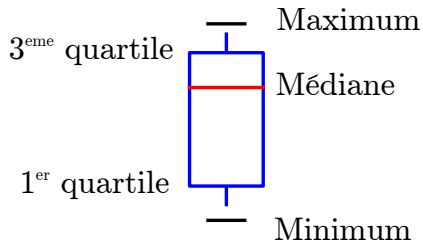
4.3.2 Classification de MNIST

Dans un premier temps, nous étudions l'influence de différents paramètres sur le réseau. Pour cela, nous utilisons la base de données MNIST [1], composée d'images 20x20 pixels de chiffres manuscrits. L'utilisation de ces données permet une analyse simple de ce qui a été appris : 10 classes naturelles sont définies et la structure des images est assez simple pour pouvoir interpréter les caractéristiques apprises. Nous rapportons les résultats obtenus pour 10 répétitions de chaque expérience comme illustré figure 4.5. Nous utilisons des autoencodeurs avec 100 neurones en couche cachée.

Nous commençons par entraîner le réseau sur un ensemble de données comprenant 1000 instances de chaque chiffre, et nous le testons sur un autre jeu de données contenant 1000 nouvelles instances pour chaque chiffre. Nous étudions l'influence du bruit ajouté dans la couche *softmax*. Nous utilisons un bruit gaussien centré et nous faisons varier son écart-type (auquel nous nous référons en temps que "niveau de bruit"). La figure 4.6 illustre l'activité moyenne du neurone *softmax* le plus actif pour chaque instance de l'ensemble de test. Nous rapportons la performance de classification par rapport aux 10 classes naturelles mesurée à l'aide de l'"indice de Rand ajusté" (ARI) (Hubert et Arabie 1985) sur la figure 4.7. Il prend la valeur maximale de 1 pour une correspondance parfaite entre les deux classifications et un score de 0 correspond à une classification aléatoire.

Les figures 4.6 et 4.7 illustrent l'existence d'un compromis optimal entre la précision de la classification (en terme de performance ARI) et sa netteté (en terme d'activité de

1. `http://yann.lecun.com/exdb/mnist/`



+ Valeur aberrante

Figure 4.5 – *Pour chaque expérience, nous rapportons les résultats de 10 répétitions indépendantes sous la forme de boîtes à moustaches. Une valeur est considérée comme aberrante si elle est éloignée de plus de 1.5 fois l'écart entre le premier quartile (Q1) et troisième quartile (Q3) du quartile le plus proche, c'est-à-dire si elle est supérieure à Q3+1.5×(Q3-Q1) ou inférieure à Q1-1.5×(Q3-Q1).*

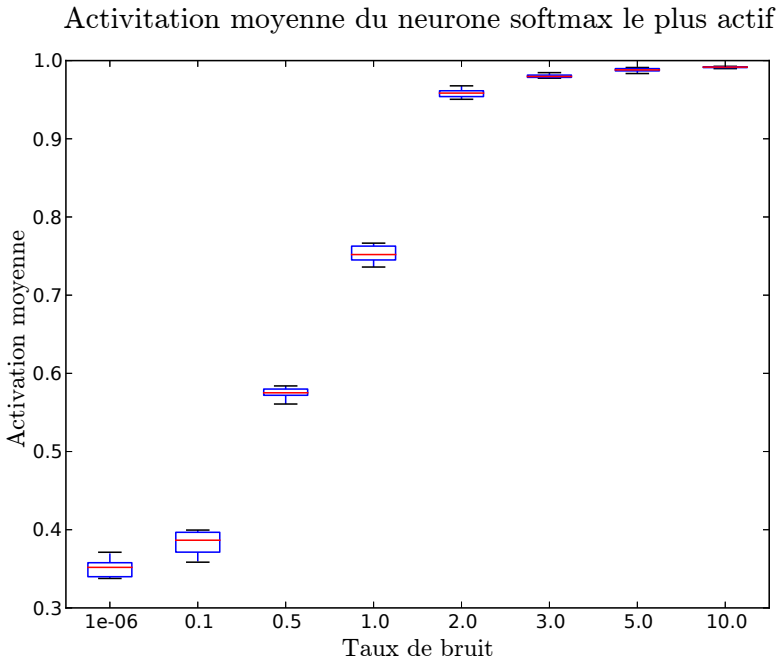


Figure 4.6 – *Activité moyenne du neurone le plus actif de la couche softmax. Chaque boîte correspond à 10 répétitions de l'expérience. Pour un niveau de bruit supérieur à 2, le réseau effectue une classification franche : en moyenne pour chaque donnée en entrée, un des neurones softmax est actif à plus de 95%.*

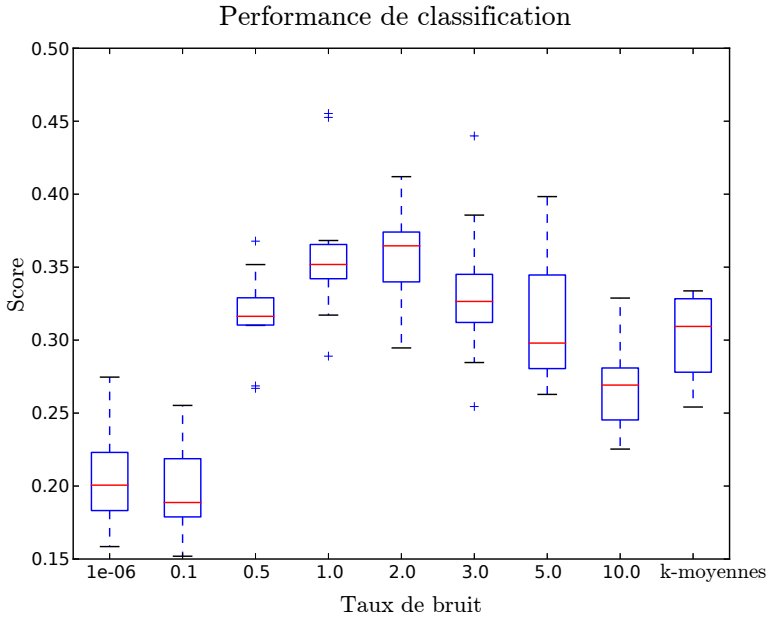


Figure 4.7 – *Performance de la classification mesurée à l'aide de l'indice Rand ajusté. Pour chaque donnée, nous considérons le neurone le plus actif comme étant le label prédit. L'algorithme est comparé aux k-moyennes initialisées avec 10 données tirées aléatoirement dans l'ensemble d'apprentissage.*

la couche *softmax*) pour un niveau de bruit de 2. Nous utilisons donc cette valeur dans la suite des expériences.

Le nombre de neurones de la couche *softmax* peut être considéré comme une connaissance *a priori* importante fournie au réseau. Nous étudions donc maintenant le comportement du réseau pour des nombres différents de neurones. La figure 4.8 montre qu'il ne s'agit pas d'un paramètre critique : la performance de la classification avec 10 neurones (qui est le nombre de classes dans la base MNIST) est similaire à la performance obtenue avec 100 neurones. Ceci différencie ce réseau de l'algorithme des k-moyennes, qui obtient une performance similaire pour 10 neurones mais dont la performance décroît de manière significative quand ce nombre augmente.

La figure 4.9 illustre la capacité du réseau à n'utiliser qu'un sous-ensemble de neurones parmi ceux disponibles : dans notre expérience sur la base MNIST, le nombre de classes apprises semble converger vers 25.

Nous étudions maintenant les représentations apprises par le réseau. La figure 4.10 montre les images qui provoquent l'activation la plus importante pour chacun des 10 neurones *softmax*, c'est-à-dire les caractéristiques discriminantes qui ont été apprises par le réseau pour classifier les données. Chacune de ces images est accompagnée de la variété correspondante apprise par le réseau. Ces variétés sont obtenues en fixant l'activité d'un neurone *softmax* à 1 et en faisant varier l'activité des neurones de la couche *softplus*. Étant donné que la multiplication par un même scalaire de toutes les activités des neurones *softplus* induit une reconstruction plus ou moins contrastée (cf. équation 4.4), une

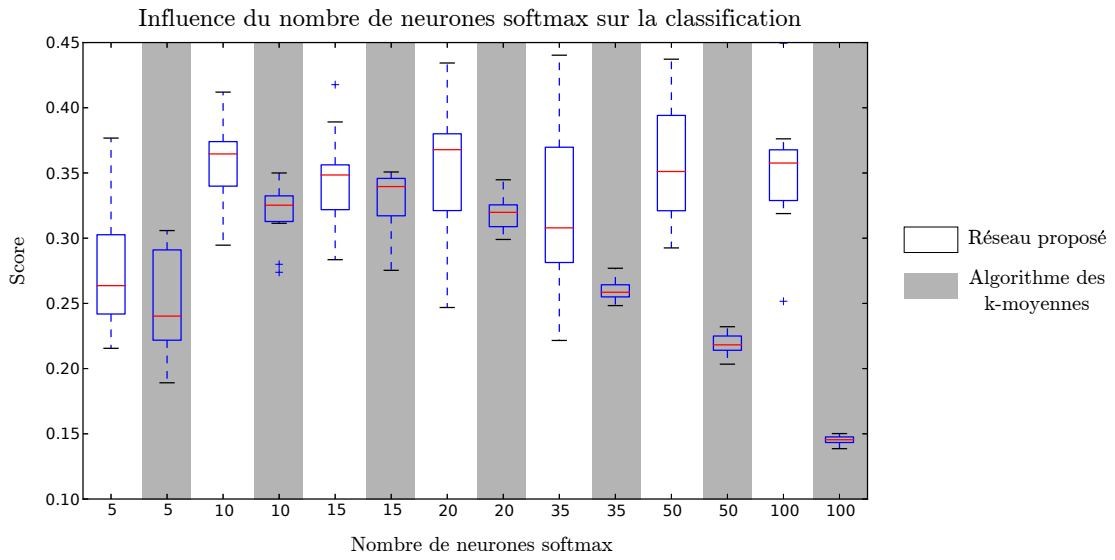


Figure 4.8 – *Influence du nombre de neurones softmax sur la performance de classification. Pour un nombre de neurones différent du nombre de classes effectivement présentes dans l'ensemble d'apprentissage, la performance du réseau proposé est plus régulière que celle obtenue avec l'algorithme des k-moyennes.*

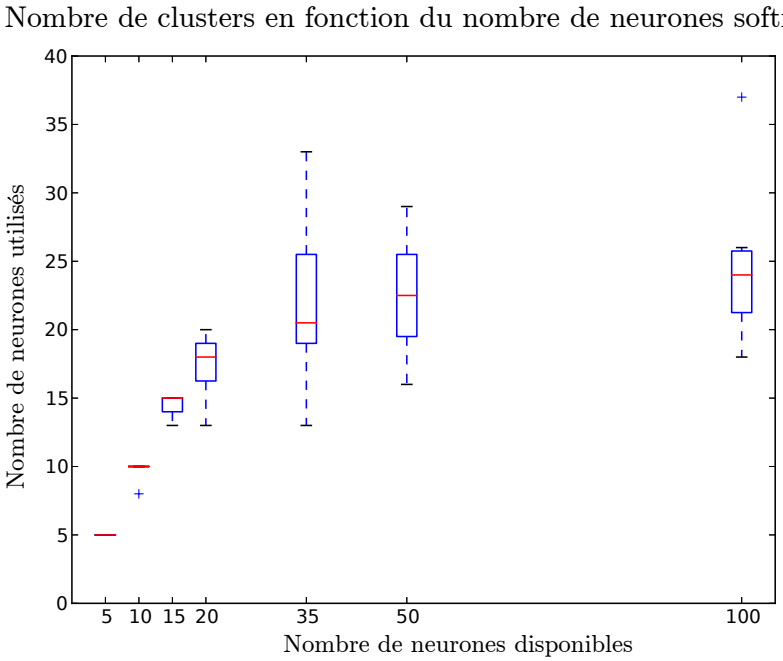


Figure 4.9 – *Nombre de neurones softmax utilisés par le réseau en fonction du nombre de neurones disponibles. Un neurone est considéré comme utilisé si son activité est supérieure à celle de tous les autres neurones pour au moins une donnée de l'ensemble d'apprentissage.*

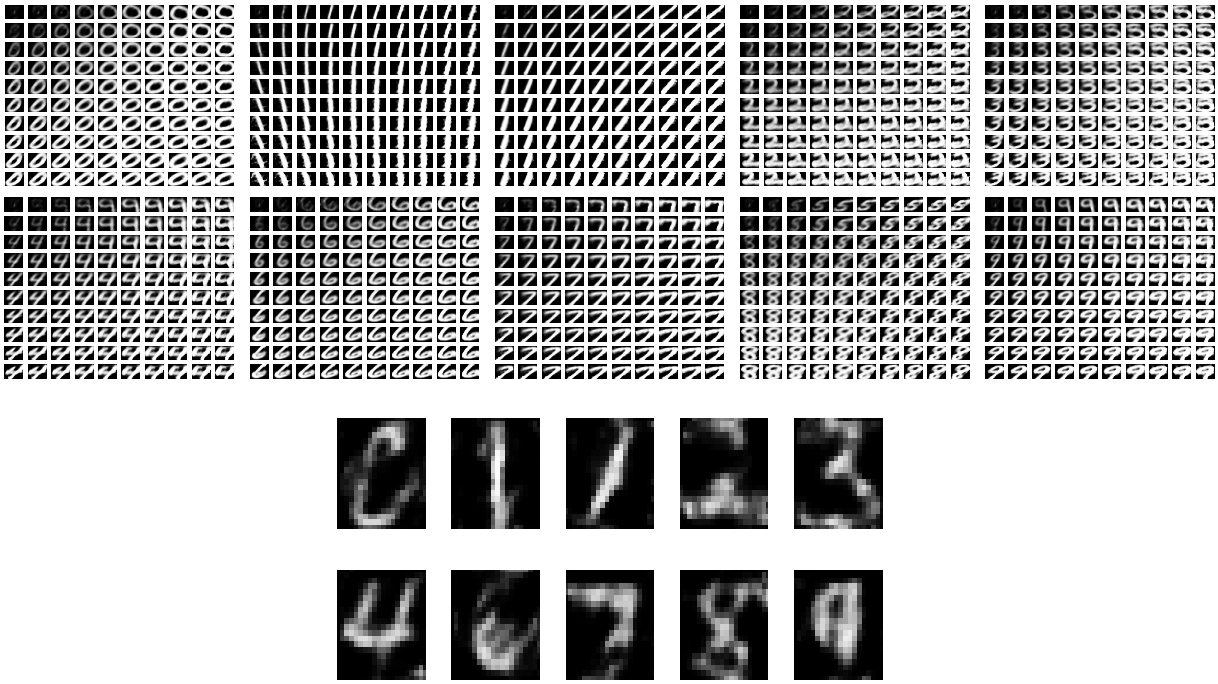


Figure 4.10 – *Représentations apprises par un réseau avec 10 neurones softmax et 2 neurones softplus, avec les prototypes correspondants. Chaque sous-variété (en haut) est obtenue en fixant l'activation d'un neurone softmax à 1 et en faisant varier l'activation des neurones softplus. Les prototypes (en bas) correspondent aux images qui provoquent la plus forte activation pour chaque neurone softmax.*

dimension des variétés correspond naturellement à la luminosité. La figure 4.12 illustre le cas d'un réseau avec 10 neurones *softmax* et 3 neurones *softplus* pour lequel les deux autres dimensions sont représentées.

La figure 4.11 correspond au cas d'un réseau sans couche *softplus*. Comparé à la figure 4.10, cela illustre les synergies entre la classification et l'apprentissage des sous-variétés : la présence d'une couche *softplus* permet aux neurones *softmax* de se concentrer sur les caractéristiques discriminantes de chaque classe, tandis que les variations internes de chaque classe sont représentées par la couche *softplus*.

4.3.3 Mélanger vision et proprioception

Comme nous l'avons argué dans le chapitre 2, l'utilisation d'entrées multimodales permet de guider l'apprentissage de concepts. Pour tester le comportement de notre réseau dans un cadre multimodal, nous menons une expérience similaire à l'expérience précédente sur MNIST, en couplant cette fois-ci l'image à de la proprioception. Pour ce faire, nous utilisons le robot humanoïde iCub (Natale et al. 2013) pour collecter un nouvel ensemble de données. Nous enregistrons les vitesses articulaires des six degrés de liberté du bras droit du robot pendant qu'il est manipulé par un opérateur humain qui lui fait écrire des chiffres sur un tableau blanc. Pour chaque trajectoire, nous enregistrons l'image

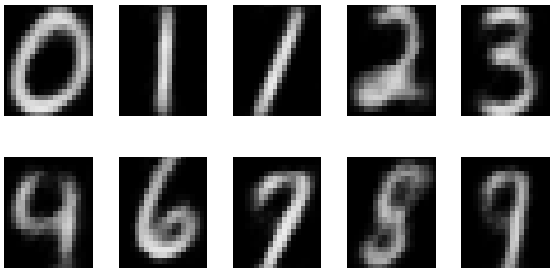


Figure 4.11 – *Prototypes appris par un réseau sans neurone softplus. Dans ce cas, le réseau ne comprend que 10 neurones softmax entraînés à reconstruire les données fournies en entrée, comme dans le cas classique des autoencodeurs par débruitage. Les prototypes appris sont beaucoup plus proches de l'instance moyenne de chaque classe que dans le cas de la figure 4.10, où les neurones se spécialisent sur les caractéristiques discriminantes.*

du chiffre dessiné telle que filmée par les caméras du robot. La photo 4.13 illustre le dispositif expérimental. Nous avons enregistré un total de 760 exemples (76 exécutions de chacun des 10 chiffres).

Les trajectoires articulaires sont normalisées de manière à avoir la même durée, fixée à 100 pas de temps. Ceci donne donc des entrées à 600 dimensions (6 degrés de liberté). Les images sont post-traitées pour augmenter le contraste et réduire les images à une boîte englobante des chiffres de taille de 20x20 pixels, de manière à les rendre similaires à la base MNIST. Ce post-traitement est entièrement automatisé et se décompose en plusieurs étapes :

- nous isolons le canal rouge de l'image,
- nous inversons le contraste,
- nous découpons la moitié inférieure de l'image (zone correspondant au tableau blanc) ; à cette étape, les chiffres écrits en vert sur fond blanc sont donc devenus blancs sur fond noir,
- nous calculons le centre de masse des pixels blancs et découpons une fenêtre de 50x50 pixels centrée sur ce point,
- nous redimensionnons cette fenêtre à une taille de 20x20 pixels.

La figure 4.14 montre l'exemple d'une image brute enregistrée par la caméra et l'image obtenue après traitement. La figure 4.15 représente quelques images de la base ainsi obtenue.

Comme précédemment, nous utilisons 100 neurones pour la couche cachée de l'autoencodeur appliqué aux images. Celui appliqué aux trajectoires est quant à lui doté de 150 neurones.

Deux modalités valent mieux qu'une

Nous comparons tout d'abord la performance de classification obtenue par le réseau lorsqu'une seule modalité est utilisée ou quand les deux modalités sont disponibles. La figure 4.16 compare les performances dans chaque cas.

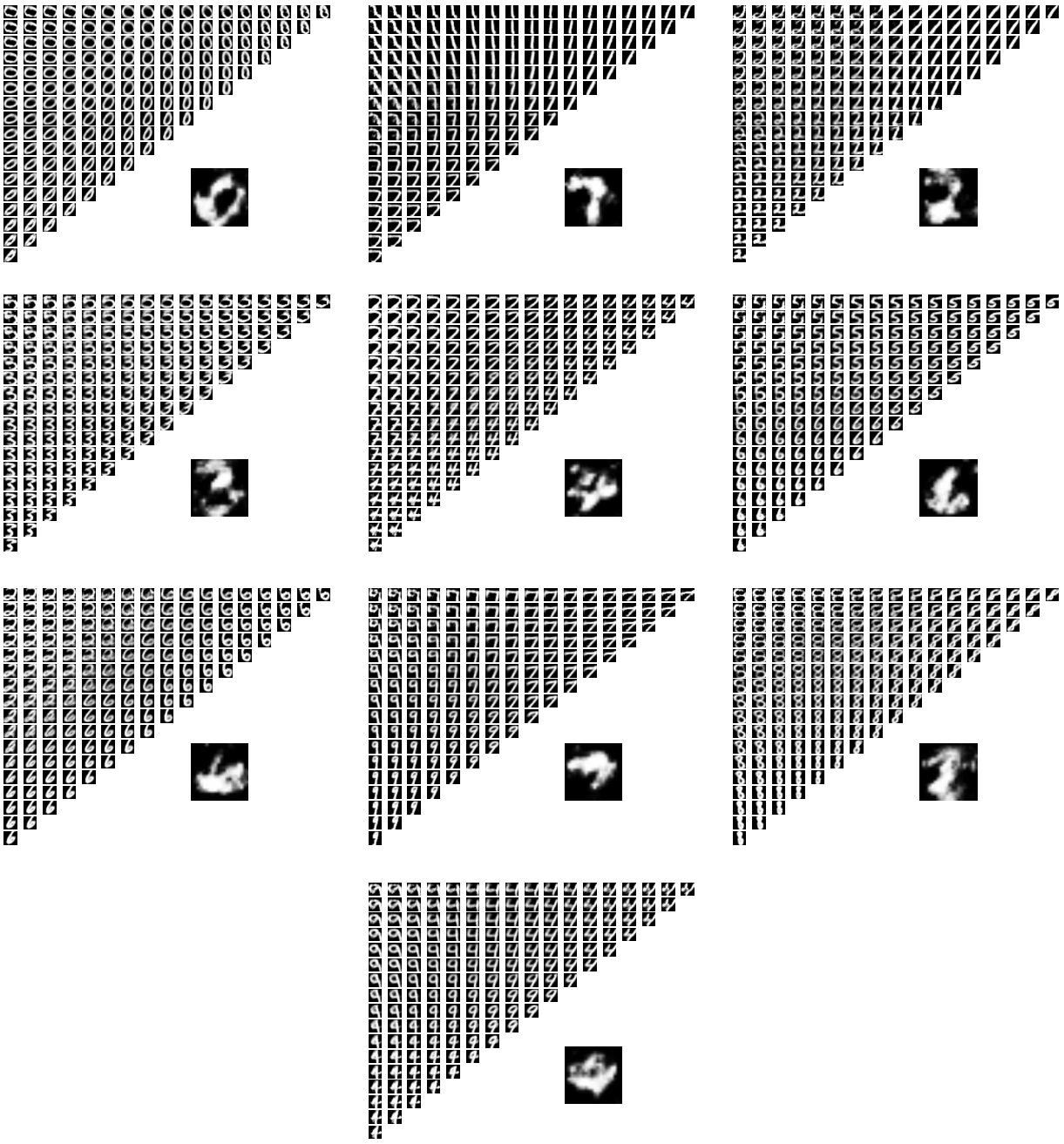


Figure 4.12 – *Représentations apprises par un réseau avec 10 neurones softmax et 3 neurones softplus. Chaque sous-variété est obtenue en fixant l'activation d'un neurone softmax à 1 et en faisant varier l'activité de la couche softplus de manière à avoir une activité totale constante (d'après l'équation 4.4, multiplier toutes les activités par un même scalaire produit une reconstruction plus ou moins contrastée). Chaque variété est représentée aux côtés de son prototype (image qui provoque la plus forte activation du neurone softmax correspondant).*

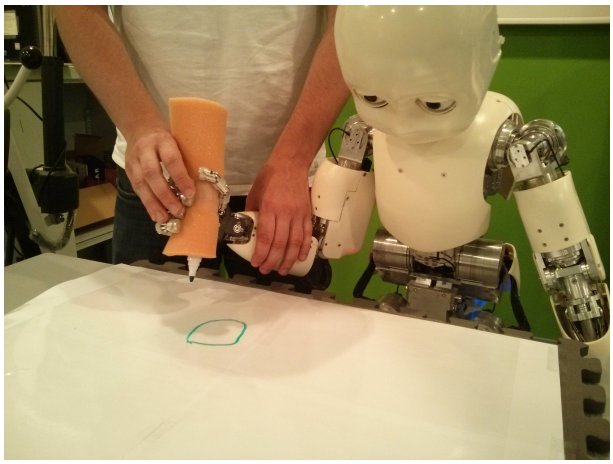


Figure 4.13 – *Dispositif expérimental. Le robot iCub est contrôlé en couple nul (*Ivaldi *et al. 2011) pendant qu'un opérateur humain manipule son bras pour lui faire écrire des chiffres. Pour chaque chiffre, nous enregistrons la trajectoire articulaire et l'image du chiffre tel qu'il est filmé par les caméras du robot.

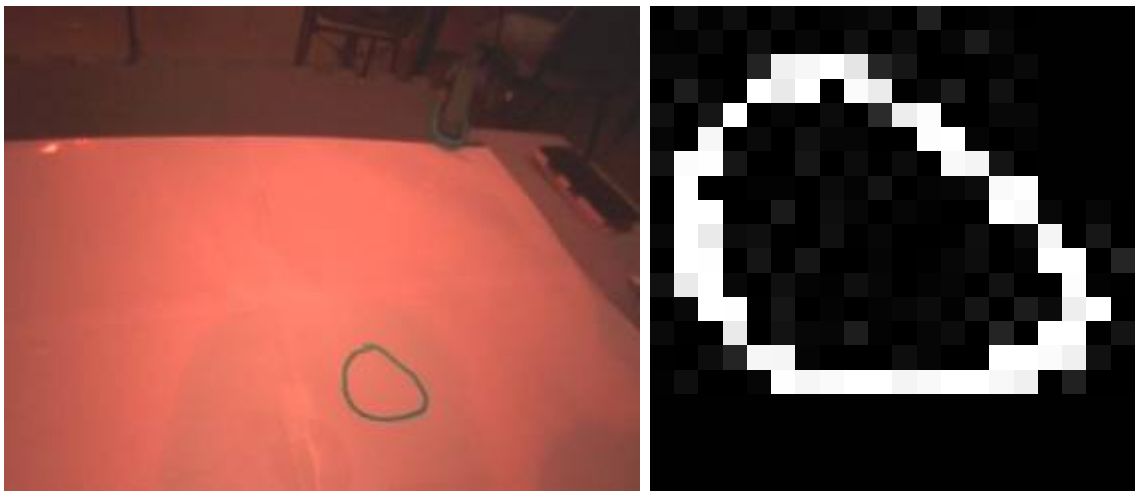


Figure 4.14 – *À gauche : image enregistrée depuis la camera d'iCub. À droite : même image après post-traitement.*

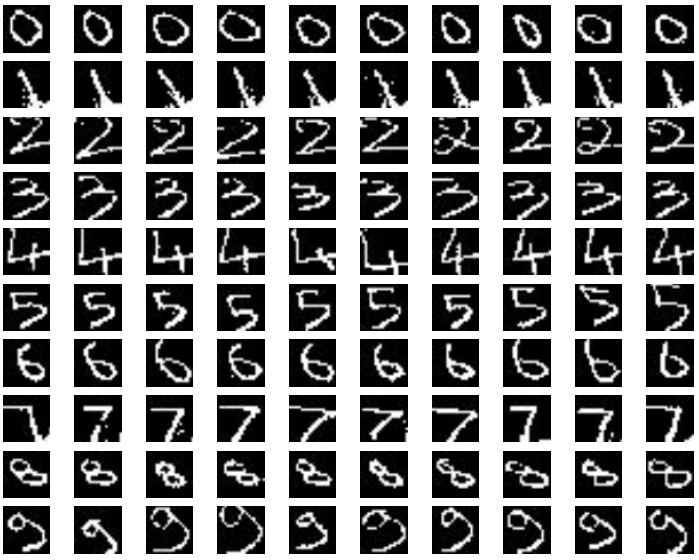


Figure 4.15 – *Quelques exemples d'images enregistrées sur le robot iCub après post-traitement. Un total de 760 images a été enregistré.*

Prédire une modalité à partir de l'autre

Le réseau a été conçu de telle sorte qu'il peut utiliser une entrée partielle où seule une modalité est fournie pour inférer les valeurs des autres modalités. Pour tester cette capacité, nous entraînons le réseau sur 700 des 760 exemples enregistrés, puis nous le testons sur les 60 exemples restants. Pour chaque modalité, nous calculons l'erreur de reconstruction d'une modalité étant donnée la seconde (figure 4.17). La figure 4.18 illustre les reconstructions des images obtenues à partir des trajectoires. Il apparaît assez nettement que seul un des chiffres "2" a été mal classifié et confondu avec un "3". De la même manière, la figure 4.19 illustre les trajectoires inférées à partir des images (nous avons utilisé un modèle cinématique du robot pour les représenter dans l'espace cartésien). Dans ce cas, on observe davantage de mauvaises classifications : deux 0 sont considérés comme 6 et 2, un 2 est confondu avec un 4 et un 3 avec un 9 (4 erreurs pour 60 exemples).

4.3.4 Avec trois modalités

L'architecture proposée peut s'adapter à un nombre quelconque de modalités. Dans cette partie, nous l'illustrons en introduisant une troisième modalité, le son. Pour cela, nous enregistrons un locuteur qui prononce le nom des dix chiffres, 76 fois pour chaque chiffre. L'entrée fournie au réseau consiste alors en un spectrogramme brut du son enregistré, calculé avec une fenêtre temporelle de 42 millisecondes se chevauchant à 50%. L'autoencodeur utilisé contient, comme pour la proprioception, 150 neurones cachés. Afin

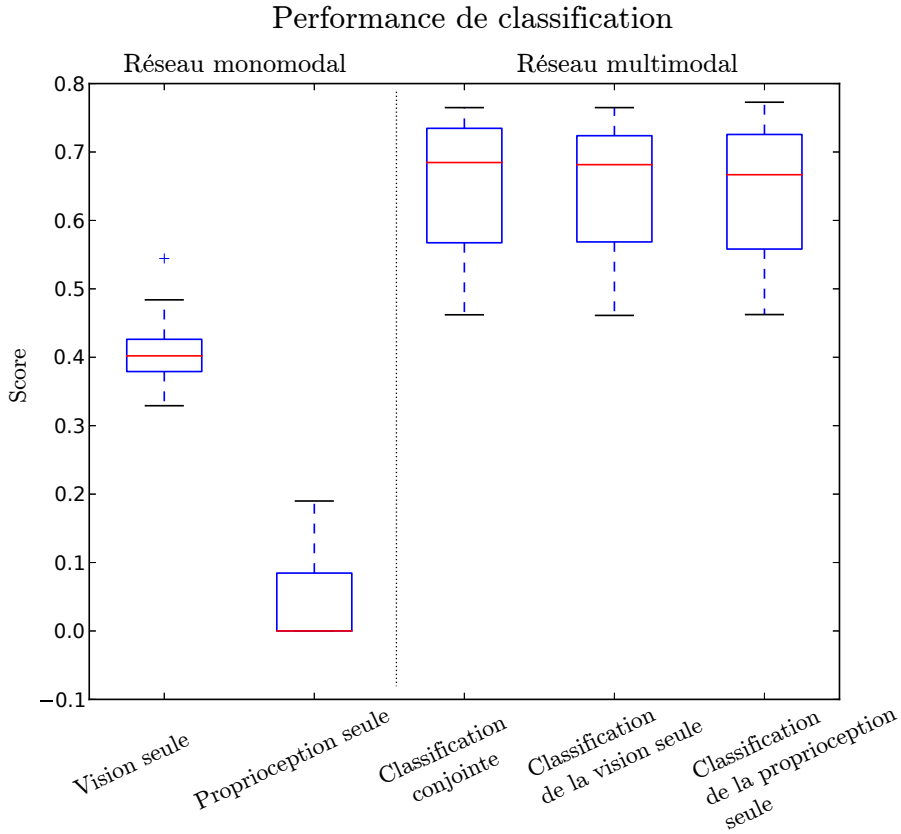


Figure 4.16 – *Performance de classification avec des entrées multimodales. L'utilisation de plusieurs modalités augmente les performances de l'algorithme alors même que chaque modalité prise séparément donne de piètres résultats. En particulier, les trajectoires sont difficiles à classifier étant donné leur fort taux de bruit (dû notamment à l'échantillonnage des capteurs) mais permettent néanmoins d'apporter des informations utiles en combinaison avec les images. De plus, la performance de classification à partir des trajectoires seules est nettement améliorée lorsque le réseau a été entraîné en utilisant les deux modalités visuelle et proprioceptive.*

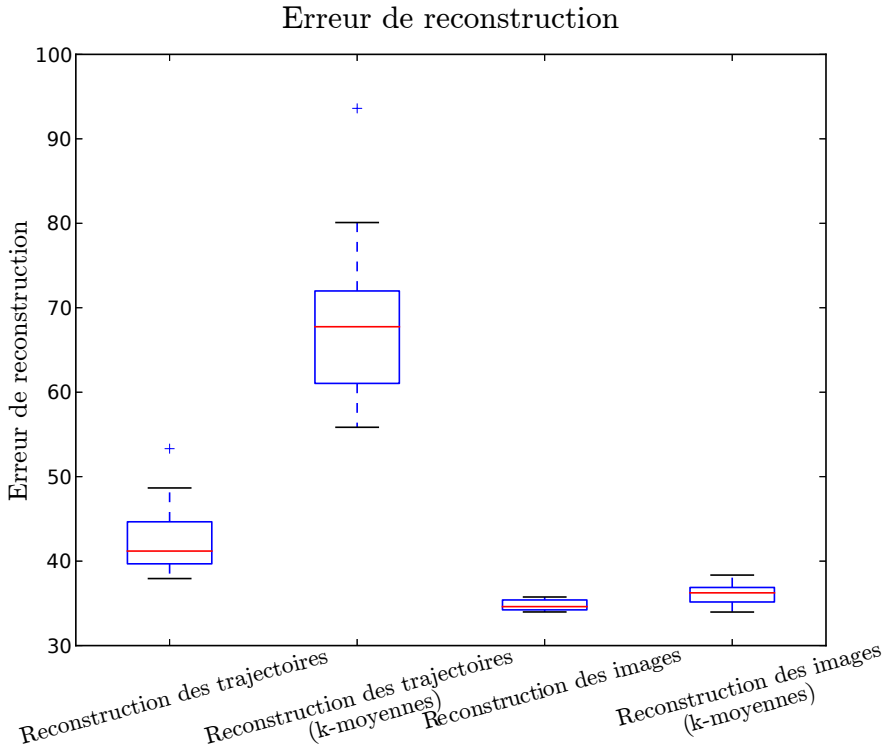


Figure 4.17 – *Erreur de reconstruction d'une modalité étant donnée la seconde. Le réseau est entraîné à partir des deux modalités, puis utilisé pour reconstruire une modalité étant donnée la seconde. Nous le comparons avec l'algorithme des k-moyennes entraîné sur la concaténation des deux modalités. Lorsque ce dernier est utilisé pour reconstruire une modalité, nous calculons le centroïde le plus proche à partir d'une distance euclidienne calculée uniquement sur la modalité fournie en entrée. Nous considérons alors l'autre partie du centroïde comme la reconstruction de l'autre modalité. En tant que référence, si nous calculons l'erreur en considérant la moyenne de toutes les entrées en tant que reconstruction, nous obtenons une erreur de 42.1 pour les images et 106.3 pour les trajectoires.*

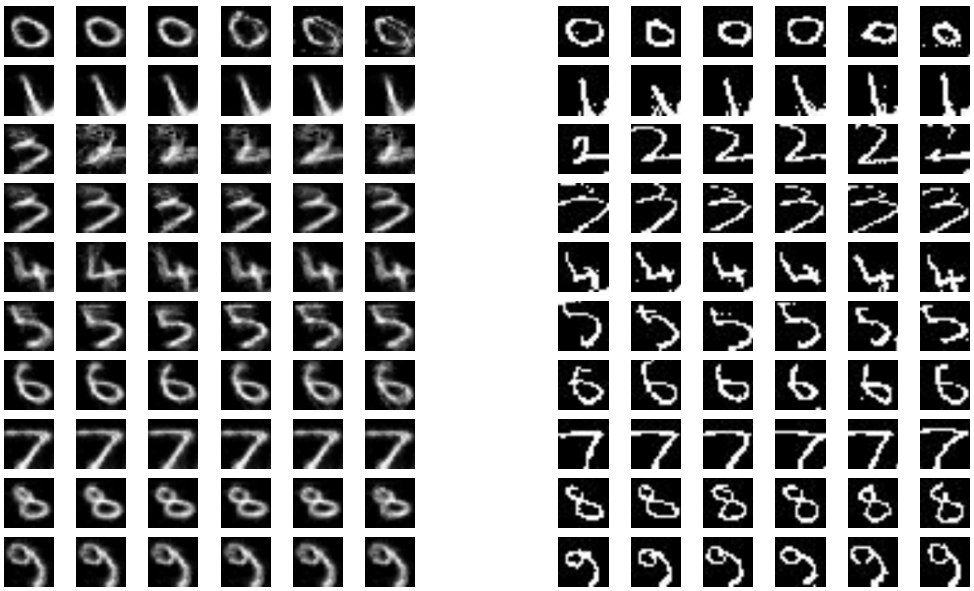


Figure 4.18 – *À gauche : images inférées par le réseau à partir des trajectoires. À droite : vérité terrain.*

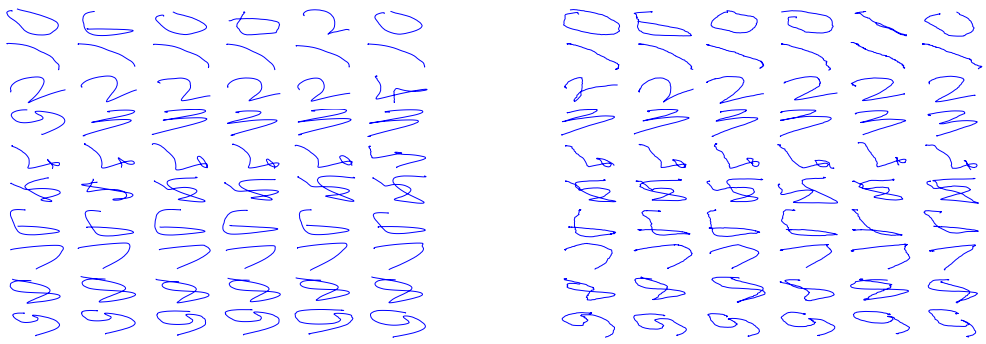


Figure 4.19 – *À gauche : trajectoires inférées par le réseau à partir des images. À droite : vérité terrain.*

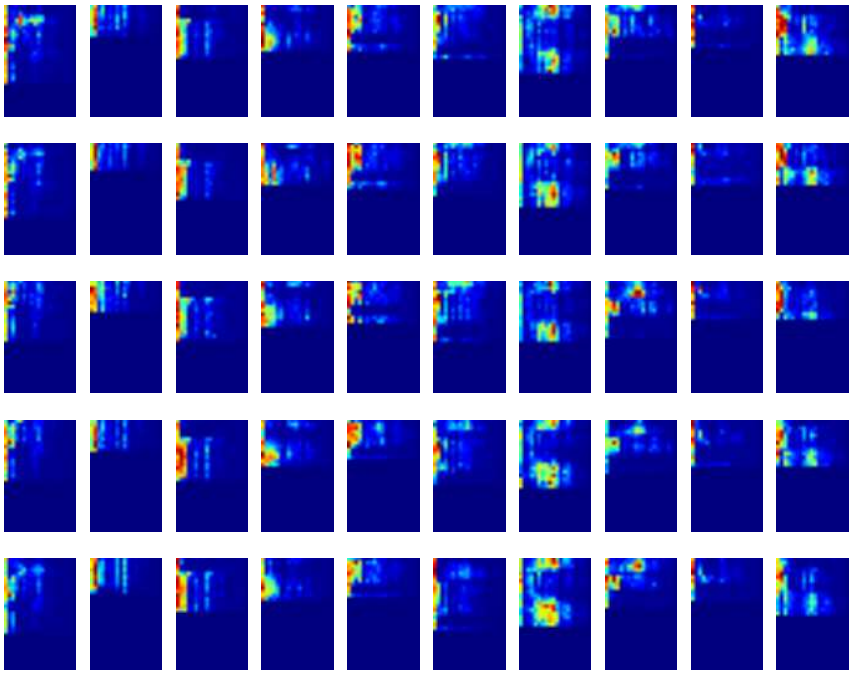


Figure 4.20 – *Exemples de spectrogrammes enregistrés et utilisés en tant que troisième modalité. Chaque colonne correspond à un nombre de 0 (à gauche) à 9 (à droite).*

que tous les spectrogrammes aient la même taille, ils sont complétés avec des 0 jusqu'à atteindre la taille du plus long (correspondant à environ 1.3 secondes). La figure 4.20 montre quelques-uns des spectrogrammes obtenus. Chaque spectrogramme correspond à un vecteur de 620 valeurs.

La performance en classification obtenue par le réseau entraîné sur les trois modalités est représentée figure 4.21. On peut y observer que l'utilisation des spectrogrammes seuls donne une performance comprise entre celle obtenue avec les images et celle obtenue avec les trajectoires (score médian : 0.4 pour les images, 0.0 pour les trajectoires et 0.18 pour l'audio). De plus, comme dans le cas de la figure 4.16 avec deux modalités, entraîner le réseau avec les trois modalités améliore la classification obtenue pour n'importe quelle combinaison de modalités.

4.4 Discussion

Dans un premier temps nous discutons les résultats présentés dans la partie précédente, avant d'exposer quelques perspectives offertes par ce travail.

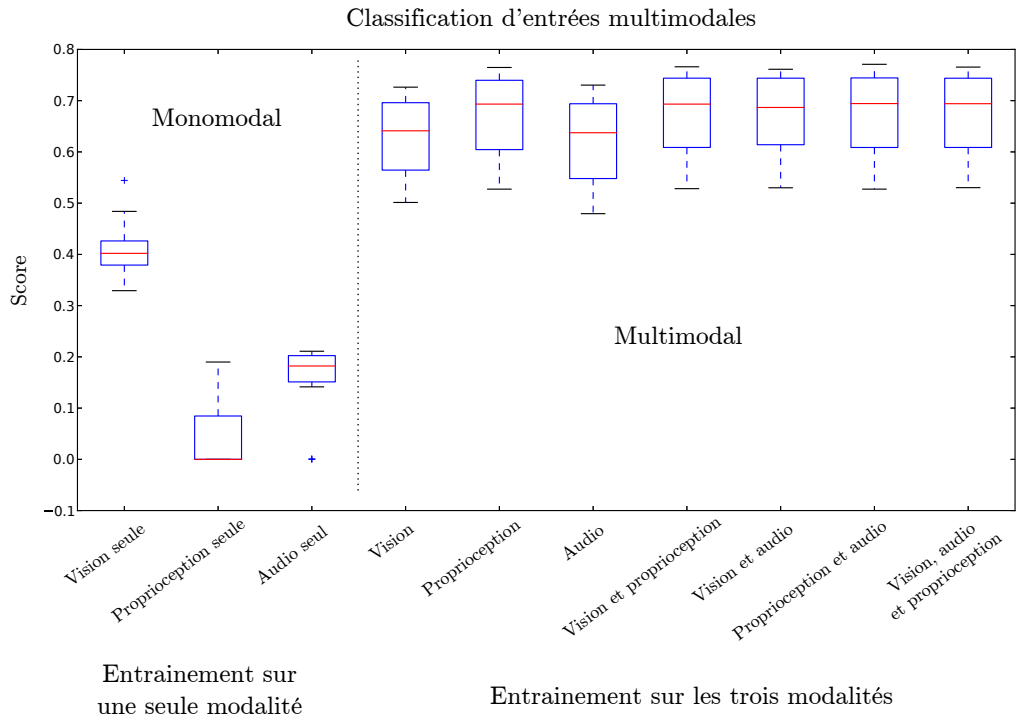


Figure 4.21 – *Score de classification obtenu par le réseau entraîné avec trois modalités et testé sur différents sous-ensembles des trois modalités.*

4.4.1 Classification

La première expérience vise à étudier l'influence de deux paramètres importants du modèle : le niveau de bruit de régularisation et le nombre de neurones *softmax*. Comme expliqué dans la section 4.2.1, la figure 4.6 montre qu'en l'absence de bruit le réseau a tendance à apprendre une représentation distribuée sur tous les neurones *softmax* : l'activité moyenne du neurone *softmax* le plus actif pour chaque exemple est d'environ 0.35. En augmentant le niveau de bruit, on obtient une représentation plus discriminante. Par exemple pour un écart-type de 10, l'activité moyenne du neurone le plus actif est supérieure à 0.99. La figure 4.7 montre cependant que, si le score de classification augmente pour des niveaux de bruit relativement faibles, il décroît lorsque l'écart-type augmente au delà de 2. Ceci s'explique par le fait qu'un niveau de bruit important se traduit par une sélection aléatoire du neurone *softmax* qui empêche le réseau de pouvoir apprendre les prototypes de chaque classe. Le niveau de bruit optimal résulte donc d'un compromis entre l'obtention d'une représentation symbolique (un seul neurone *softmax* actif) et la performance de la classification. Différents facteurs peuvent influencer la valeur optimale, en particulier la base de données utilisée et les poids initiaux du réseau (c'est-à-dire la valeur moyenne reçue en entrée par chaque neurone *softmax* par rapport à la valeur moyenne du bruit) et le taux d'apprentissage (c'est-à-dire la capacité du réseau à modifier suffisamment rapidement les poids synaptiques afin de casser la symétrie induite par une sélection aléatoire des neurones).

La figure 4.9 souligne quant à elle la capacité du réseau à utiliser un nombre de catégories plus faible que le nombre de neurones disponibles. Cette caractéristique résulte de plusieurs mécanismes. Premièrement, le bruit de régularisation force le réseau à apprendre des prototypes discriminants pour chaque catégorie permettant une ségrégation marquée entre catégories afin de pouvoir contrer l'effet du bruit, ce qui rend le réseau moins sensible à de petites variations des entrées. Deuxièmement, l'utilisation d'une paramétrisation des données à l'aide d'une couche *softplus* distincte permet au réseau d'absorber progressivement ces petites variations dans cette paramétrisation, permettant alors aux neurones *softmax* de se focaliser sur les caractéristiques réellement discriminantes partagées par tous les exemples d'une même catégorie. Par conséquent, un nouveau neurone *softmax* ne peut être activé que par une nouvelle entrée suffisamment éloignée des exemples déjà appris. Cette propriété du réseau est intéressante dans une perspective d'apprentissage permanent, puisque qu'un grand nombre de neurones peut être alloué au réseau lors de son initialisation sans grand impact sur la performance tout en permettant au réseau de pouvoir apprendre de nouvelles catégories lorsque de nouveaux stimuli sont rencontrés.

Il est important de remarquer que cette distinction entre la classification et la propriété générative du réseau, qui utilisent des matrices de poids différentes, permet au réseau de se focaliser sur des sous-parties des stimuli, par opposition aux approches par dictionnaire ou par cartes auto-organisatrices (par exemple (De Sa et Ballard 1998 ; Morse et al. 2010b)). De plus, cette capacité est en accord avec le système des symboles perceptuels de (Barsalou 1999), selon lequel les concepts consistent en un sous-ensemble sélectionné de la perception (par exemple le concept de *chaise* ne dépend pas des couleurs perçues).

Ce mécanisme de *décision puis représentation* partage des points communs avec les réseaux profonds séquentiels proposés par (Denoyer et Gallinari 2014). En effet, ces derniers s'appuient sur une politique apprise pour sélectionner une manière d'encoder une entrée. Dans notre cas, la politique est codée à travers la matrice W reliant la couche d'entrée à la couche *softmax*, dont l'activité est utilisée pour moduler la projection de l'entrée vers la couche *softmax*. Cependant, à la différence de (Denoyer et Gallinari 2014), l'utilisation d'une couche *gated* permet un partage des ressources et des représentations entre toutes les politiques. Notre approche a donc une puissance représentationnelle plus faible, celle de (Denoyer et Gallinari 2014) permet par exemple d'utiliser différentes fonctions d'activation selon les politiques, mais est susceptible de posséder des propriétés de généralisation et de transférabilité plus élevées (l'ajout d'une nouvelle option dans la politique peut par exemple s'appuyer directement sur des éléments déjà utilisés par les autres options de la politique). De plus amples études sont nécessaires pour quantifier précisément les avantages et inconvénients de chacune de ces approches.

4.4.2 Apprentissage de variétés

La dernière expérience menée sur la base MNIST étudie les représentations apprises par le réseau. La figure 4.10 correspond à un réseau utilisant deux neurones *softplus*. Dans ce cas, à la fois les prototypes et les variétés apprises sont facilement interprétables au travers des catégories naturelles des chiffres. La plupart de ces sous-variétés sont spécifiques (0, 1, 2, 6, 7 et 9), alors que d'autres mélangent plusieurs chiffres (3 et 5, 4 et 9 ainsi que 8

et 5). Le chiffre 5 en particulier n'est pas représenté dans une catégorie dédiée, mais se retrouve tantôt représenté comme un 3, tantôt comme un 8.

La capacité à représenter chaque donnée à l'aide d'un système de coordonnées sur une sous-variété pourrait être utilisée par un système de raisonnement symbolique, en considérant les classes comme des symboles et les coordonnées comme des traits. On obtient alors une plus grande richesse qu'en utilisant des cartes auto-organisatrices, comme dans (Morse et al. 2010b ; Lallee et Dominey 2013). En effet, ces dernières fournissent une représentation symbolique sous la forme du neurone le plus actif, mais la discrimination entre deux stimuli similaires correspondant au même neurone passe par leur distance dans l'espace de départ. En cela, l'approche proposée s'apparente plus à la théorie des zones de convergence-divergence (Meyer et Damasio 2009), qui stipule que les représentations de haut niveau ne sont pas de simples copies de la perception, mais sont plutôt les informations minimales nécessaires pour pouvoir reconstruire une approximation des perceptions originelles dans les cortex sensoriels primaires.

L'ajout d'un troisième neurone *softplus* (figure 4.12) augmente les capacités représentationnelles du réseau, ce qui implique que plus de variations des données peuvent être représentées autour d'un même prototype. Ceci résulte en une classification moins claire en terme de chiffres, plusieurs d'entre eux pouvant être représentés par des variations autour d'un même prototype, mais la continuité entre des images adjacentes sur chaque sous-variété est cependant préservée, la représentation étant alors toujours cohérente avec l'hypothèse des sous-variétés.

4.4.3 Fusion multimodale

Dans les expériences subséquentes, nous avons étudié l'influence de la multimodalité sur la performance en classification. Nous avons tout d'abord ajouté la proprioception, à travers les vitesses articulaires enregistrées au cours d'une tâche d'écriture. La figure 4.16 montre que, prises isolément, les images et les trajectoires sont assez mal classifiées. En particulier, à cause d'un bruit important (notamment d'échantillonnage), le score de classification pour les trajectoires est très mauvais. Cependant, lorsque le réseau est entraîné sur ces deux modalités, la classification apprise est bien meilleure et s'approche des dix classes naturelles des chiffres. Ce résultat est similaire à celui obtenu par (Nakamura et al. 2009) en utilisant une allocation de Dirichlet latente. Notre approche est toutefois plus générale dans le sens où elle ne repose pas sur un dictionnaire prédéfini pour encoder chaque entrée.

Surtout, l'apprentissage du réseau sur les deux modalités améliore la classification pour chaque modalité considérée séparément, comme observé par (De Sa et Ballard 1997). En effet, après apprentissage, le score de classification ne dépend plus de la modalité choisie en entrée : il est à peu de choses près le même en utilisant les trajectoires seules, les images seules, ou les deux modalités en même temps. Le même effet est obtenu lorsqu'une troisième modalité est ajoutée (figure 4.21). Ces résultats sont en accord avec l'observation chez l'humain que la multimodalité aide à la compréhension de chaque modalité isolée (Giard et Peronnet 1999).

Une autre propriété du réseau est sa capacité à inférer une modalité étant donné une

autre, comme dans (Lallee et Dominey 2013), propriété au cœur de la théorie des zones de convergence-divergence (Meyer et Damasio 2009). En effet, étant donnée une modalité, le réseau peut inférer une classification et une paramétrisation qui peuvent à leur tour être utilisées pour reconstruire la modalité absente. La figure 4.17 montre que la reconstruction est plus spécifique que celle obtenue en utilisant les centroïdes calculés par l'algorithme des k-moyennes. De plus, la reconstruction des chiffres mal classifiés illustrée figures 4.18 et 4.19 montre que de petites variations de l'entrée peuvent produire un changement complet de la perception, grâce à l'utilisation de deux couches distinctes pour la représentation. Ceci rappelle l'effet de la perception catégorielle (Goldstone et Hendrickson 2010), pour lequel la distance perçue entre deux stimuli n'est pas liée à la distance physique réelle entre les stimuli mais dépend d'une certaine représentation interne. Ceci peut sembler contradictoire avec la discussion du paragraphe 4.4.1 sur la robustesse du réseau à de petites variations des entrées (qui lui permet de ne pas utiliser tous les neurones *softmax* disponibles). Cependant, dans le cas exposé ici, les variations doivent être ciblées sur des aspects bien précis des stimuli dont l'association avec d'autres classes a été renforcée au cours de l'apprentissage. Ainsi, ce phénomène de "perception catégorielle" n'intervient qu'après apprentissage de différentes classes et non pour de petites variations aléatoires autour d'une classe si celle-ci n'est pas en compétition directe avec d'autres.

4.4.4 Perspectives

Comme évoqué dans la section 4.4.2, le nombre de neurones *softplus* a une influence importante sur la classification apprise par le réseau. Il est donc nécessaire d'étudier des mécanismes permettant au réseau de choisir dynamiquement le nombre de variables pertinentes.

Un autre facteur limitant de notre approche est lié au fait que la classification s'appuie sur un seul motif perceptuel. Aussi, le réseau ne peut pas gérer le cas où une même catégorie devrait être représentée par une disjonction de plusieurs prototypes. Ce cas requiert de remplacer le mécanisme de classification (lié dans le réseau proposé à la matrice W) par un mécanisme plus complexe. Les réseaux somme-produit (Poon et Domingos 2011) peuvent constituer pour cela une source d'inspiration intéressante.

Une autre question ouverte est le nombre de modalités qui peuvent être utilisées en entrée de l'algorithme proposé. Comme nous l'avons expliqué, ce nombre n'est pas limité d'un point de vue théorique. Cependant, l'ajout de nouvelles modalités augmente la probabilité que certaines modalités ne soient pas corrélées avec les autres. Plusieurs solutions peuvent être envisagées. Premièrement, un meilleur critère de fusion multimodale pourrait être développé. Dans le réseau proposé, chaque modalité est pondérée aléatoirement pour chaque stimulus, faisant l'hypothèse implicite que toutes les modalités reflètent un même évènement et que la décision de classification peut donc être prise à partir de n'importe laquelle. Cette approche peut-être considérée comme une version forte de la théorie de "minimisation du désaccord" (De Sa et Ballard 1997) qui pourrait être affaiblie, par exemple en s'appuyant sur la notion de prédictabilité (Lefort et al. 2014), qui consiste à pondérer l'apprentissage pour une modalité selon sa prédictabilité par les autres. Une

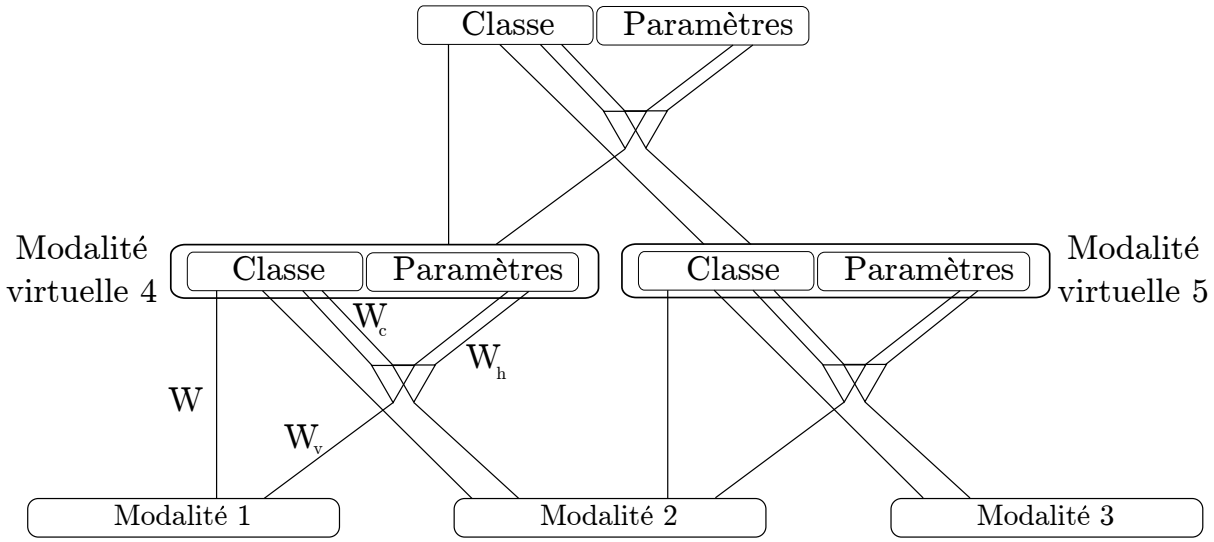


Figure 4.22 – *Le réseau proposé peut utiliser n'importe quel nombre de modalités en entrée. Cependant, d'un point de vue statistique, il peut être plus intéressant de chaîner plusieurs réseaux, chaque réseau étant limité à deux modalités en entrée (voir le texte pour les détails). Plusieurs architectures peuvent être envisagées : soit chaque combinaison de deux modalités possède son propre réseau (ce qui nécessiterait dans cette figure un réseau supplémentaire entre les modalités 1 et 3), ou certaines modalités peuvent être utilisées comme "pivots" (modalité 2 sur cette figure). Dans ce cas, les modalités "pivots" permettent de propager l'activité d'une modalité vers le reste du réseau. Sur l'exemple de cette figure, une activité dans la modalité 1 peut générer une reconstruction correspondante dans l'activité 2, qui peut à son tour produire une activité dans la modalité 3. Cette figure illustre également la manière dont le réseau peut être combiné de manière hiérarchique, en considérant la sortie d'un réseau comme une modalité "virtuelle" utilisée en entrée d'un réseau supérieur.*

autre idée, inspirée par la théorie des zones de convergence-divergence (Meyer et Damasio 2009), consiste à utiliser plusieurs instantiations du réseau en limitant à deux ou trois le nombre de modalités utilisées par chaque instance. Les différentes instances du réseau peuvent alors être chaînées en utilisant à chaque fois une modalité comme "pivot" (voir figure 4.22). De cette manière, la stimulation d'une modalité peut toujours se propager à travers l'ensemble du réseau et réactiver les stimuli correspondants dans les autres modalités, tout en limitant durant l'apprentissage l'effet des non-corrélations à des zones d'influence limitées. L'utilisation des représentations apprises par chaque instance comme modalités virtuelles pour construire des représentations hiérarchiques peut également être envisagée.

Une lacune importante du réseau proposé est la nécessité de pré-traiter les données temporelle pour les transformer en entrées statiques. Dans le cas des expériences présentées, les trajectoires étaient ainsi segmentées, normalisées à la même durée et linéarisées en un unique vecteur. De même, la modalité auditive était représentée par le spectrogramme complet du mot prononcé. Les problématiques liées à l'aspect temporel sont développées dans le prochain chapitre.

Chapitre 5

La temporalité

Ces deux temps-là donc, le passé et le futur, comment "sont"-ils, puisque s'il s'agit du passé il n'est plus, s'il s'agit du futur il n'est pas encore ? Quant au présent, s'il était toujours présent, et ne s'en allait pas dans le passé, il ne serait plus le temps mais l'éternité... Nous ne pouvons dire en toute vérité que le temps est, sinon parce qu'il tend à ne pas être.

Saint Augustin - Confessions

Sommaire

Dans le chapitre précédent, nous avons décrit une architecture permettant de représenter une entrée multimodale sous la forme d'un symbole accompagné de ses traits de variations les plus pertinents. Cependant, nous n'avons considéré que des entrées statiques, mettant de côté tout aspect temporel.

Cet aspect joue pourtant un rôle crucial dans la perception : perception du mouvement, perception de la parole, perception des causes et de leurs effets, mais aussi perception des objets à travers leur cohérence temporelle,... Certains effets perceptifs observés chez l'homme sont une source d'information intéressante sur la manière dont peuvent être traités les stimuli temporels. Nous en décrirons quelques-uns dans une première partie.

De nombreux travaux de recherche en intelligence artificielle se sont penchés sur des problèmes temporels : modélisation de séquences, transcription de la parole, classification de vidéos,... Cependant, la plupart de ces travaux reposent sur des mécanismes qui

font des hypothèses incompatibles avec les conditions dans lesquelles l'homme est placé tout au long de son apprentissage et se révèlent en contradiction avec certains principes développementaux. Nous les présenterons dans une seconde partie.

Nous ne proposerons pas dans ce chapitre de solution globale et convenable au problème de la temporalité pour la robotique développementale. Nous présenterons cependant dans une troisième partie deux travaux liminaires ouvrant quelques pistes et montrant certains avantages de l'apprentissage profond dans le cadre de cette problématique.

5.1 La perception des phénomènes temporels

Nous ne pouvons pas nous mettre en retrait par rapport au temps, comme nous ferions pour un objet ordinaire. Nous pouvons le mesurer, mais pas l'observer en le mettant à distance, car il nous affecte sans cesse. Nous sommes inexorablement dans *le temps.*

Étienne Klein - Dictionnaire de l'ignorance, 1998

Ce chapitre est dédié au rôle du temps dans la perception et non à la perception du temps lui-même. Nous entendons par là nous intéresser aux mécanismes permettant par exemple de percevoir l'écoute d'un mot comme une entité unique, ou encore de générer une séquence motrice correspondant à une action précise, et non pas aux mécanismes permettant de dire qu'un phénomène a duré trois secondes ou qu'il a été plus long qu'un autre. Bien qu'il soit probable que ces deux capacités soient liées, le problème du temps est en effet trop complexe pour que nous soyons capable de l'aborder dans sa globalité.

Nous avons abordé dans l'introduction les théories qui considèrent que les capacités de prédiction sont un mécanisme essentiel pour expliquer le fonctionnement et les capacités du cerveau. De ce point de vue, la temporalité joue un rôle crucial : les représentations apprises doivent amener à construire un modèle de l'environnement qui permet de prédire l'état sensoriel à l'instant t à partir des états passés.

Chez l'homme, le traitement de la parole, stimuli intrinsèquement temporel, met en exergue l'importance de ces prédictions pour la perception. L'effet "*cocktail party*" (Arons 1992) par exemple montre que dans un environnement bruyant, il est possible de séparer très nettement les paroles d'une personne du flux sonore ambiant même lorsque d'autres voix, c'est-à-dire des signaux de même nature aux mêmes propriétés spectrales, sont superposées. De même, lorsque l'on écoute de la musique, il est facile de porter son attention sur l'un ou l'autre des instruments, même si le son est rendu à travers un haut-parleur et qu'il n'est donc pas possible de discriminer les sons à partir de leur localisation spatiale. Même si la difficulté est plus grande et demande une concentration plus importante, il est également possible de distinguer deux discours superposés à la fois en propriétés spectrales et spatiales, comme il est possible de s'en convaincre en écoutant deux discours sur un même ordinateur [1]. De tels effets et capacités sont en accord avec les modèles prédictifs. Le rôle du modèle interne ainsi construit peut être de filtrer les stimuli importants dans le flux sensoriel brut : lorsque certains motifs ont été correctement prédits, ils peuvent servir à renforcer la partie du modèle qui les a prédits tout en filtrant la partie non prédite du

1. L'expérience montre que l'attention a tendance à sauter d'un discours à l'autre à chaque silence.

flux sensoriel (par exemple filtrage du bruit de fond dans l'effet "*cocktail party*") ou ils peuvent au contraire être éliminés au profit de la partie non prédite (on ne peut pas se chatouiller soi-même).

De ce point de vue, il est donc possible de penser que le cerveau est naturellement porté à segmenter une séquence temporelle en périodes pendant lesquelles les variations du signal peuvent être expliquées par une seule et même cause (ou variable latente). On peut faire l'hypothèse que c'est le cas par exemple lors de l'apprentissage des mots d'une langue à partir d'un flux auditif continu : un découpage en mots permet de prédire l'évolution des stimuli auditifs pendant toute la prononciation du mot et constitue donc une modélisation intéressante de l'environnement perçu. De plus, la faculté de découper un flux auditif en segments sur la base de régularités rythmiques et séquentielles semble apparaître très tôt, entre l'âge de six et neuf mois (Morgan et Saffran 1995). De même l'émergence de la notion d'objet peut elle aussi s'expliquer par l'identification d'un ensemble de stimuli visuels partageant des propriétés temporelles identiques (Lee et Blake 1999).

La tendance du cerveau à segmenter un flux temporel en séquence indépendantes a été étudiée dans de nombreux travaux. Ainsi, les auteurs de (Sakai et al. 2003) ont mené une expérience où des sujets doivent apprendre à reproduire une séquence d'appuis sur des touches en suivant des stimuli lumineux. La reproduction de ces séquences montrent que les sujets ont tendance à regrouper les appuis sur les touches en sous-séquences au sein desquelles les délais entre deux appuis sont très brefs, chaque sous-séquence étant séparée de ses voisines par des délais plus longs. Un réordonnancement des stimuli visuels (et donc de la séquence d'appuis devant être exécutée) respectant les sous-séquences apprises, propres à chaque sujet, donne lieu à un apprentissage beaucoup plus rapide que lorsque des sous-séquences se trouvent séparées en plusieurs morceaux lors du réordonnancement. De plus, dans (Wiestler et Diedrichsen 2013) les auteurs ont montré que l'apprentissage de séquences motrices se traduisait par le développement d'une activité neuronale spécifique dans certaines régions du cerveau, notamment les aires motrices primaires et secondaires, sans toutefois induire d'augmentation de l'activité globale, ce qui peut être expliqué selon les auteurs par l'apprentissage de représentations appropriées et dédiées à chaque séquence. De même, l'existence de représentations des séquences apprises sous forme globale et non comme concaténation de sous-séquences plus élémentaires est corroborée par exemple par le fait que demander à des sujets de porter leur attention sur une sous-partie d'une séquence apprise diminue leur performance. Cet effet a été montré notamment par (Ford et al. 2005) en demandant à des joueurs de football de dribbler en slalomant entre des plots (tâche principale), tout en faisant attention soit aux mouvements de leur pied (sous-tâche liée à la tâche principale), soit de leur bras (sous-tâche non liée à la tâche principale mais faisant appel à la proprioception de manière similaire à la sous-tâche précédente), ou alors à des mots prononcés par une autre personne (sous-tâche complètement indépendante de la tâche principale). Les auteurs ont alors comparé les performances des joueurs par rapport à une situation de référence où aucune tâche secondaire ne leur était demandée. Dans les trois cas, les joueurs professionnels voient leur performance diminuer, avec cependant un impact plus faible lorsqu'ils doivent porter attention aux mots prononcés. Cette différence d'impact disparaît en revanche lorsqu'ils doivent dribbler en utilisant leur pied non dominant (pour lequel ils sont moins entraînés).

Au contraire, des joueurs amateurs sont perturbés par les deux tâches non liées à la tâche principale, mais par par la sous-tâche consistant à porter attention à leur pied.

De même, dans le cas de phénomènes purement perceptifs, un effet connu est celui de la restauration de la parole (Warren 1970) : si l'on remplace certains phonèmes d'un discours par un bruit blanc, les sujets perçoivent en réalité le phonème correct et les sujets ayant perçu la présence d'un bruit blanc dans la phrase ne sont qu'au mieux capables de dire qu'un bruit blanc a été ajouté, sans pouvoir le localiser précisément. En revanche si le phonème est remplacé par un silence, ce dernier est alors consciemment perçu et correctement localisé (Warren 1970). De même, des retournements temporels du signal sur de courtes fenêtres temporelles (jusqu'à une centaine de millisecondes) n'ont qu'un faible impact sur la compréhension de la parole (Saberi et Perrott 1999) (les sujets identifiant toutefois une distorsion anormale du son).

Dans le domaine visuel, (Johansson 1973) a montré que la visualisation d'une dizaine de points lumineux fixés aux articulations principales d'un sujet en train de marcher suffisait à identifier un mouvement de marche. Le même type de visualisation sous la forme de points en mouvement permet également, entre autres, de reconnaître des personnes familières (Cutting et Kozlowski 1977) ou certaines actions (Dittrich 1993). La visualisation de points lumineux judicieusement placés sur un visage permet aussi d'améliorer la compréhension de la parole dans des environnement bruités (Rosenblum et al. 1996). L'identification de mouvements humains à l'aide de ce type de visualisation est également robuste à l'ajout de bruit sous la forme de points aléatoires, mais cette robustesse est cependant acquise au cours du développement de l'enfant (entre 6 ans et l'âge adulte), bien plus tard que la capacité à comprendre les versions non bruitées de ces stimuli (Freire et al. 2006). Ces expériences montrent que le traitement de flux temporels repose en grande partie sur la présence de caractéristiques bien précises, avec une grande robustesse aux détails du contenu même des séquences.

5.2 Des approches insatisfaisantes de la temporalité

On n'a pas la même perception du temps selon les species*, c'est ce qui fait que je peux passer la main entre toi et moi comme ça, parce que pour l'oxygène, une seconde, c'est peut-être dix secondes, et pour le béton, une seconde, c'est peut-être un millième de seconde.*

J.-C. Van Damme

De nombreux travaux en apprentissage automatique se sont penchés sur des problèmes intrinsèquement temporels : identification de séquences, transcription et compréhension du langage parlé, catégorisation de vidéo, génération de mouvements, etc. Cependant, comme nous allons le voir, beaucoup de ces travaux reposent sur des mécanismes lourds incompatibles avec les principes de base de la robotique développementale.

5.2.1 Mémoire du passé : le problème de l'œuf et de la poule

À partir du moment où l'on suppose que la perception résulte de l'interprétation de chaque stimulus, invoquer la possibilité de stocker une séquence de stimuli bruts et de la

"rejouer" par la suite (notamment pour pouvoir en apprendre une représentation) oblige à introduire une deuxième voie de traitement des stimuli, distincte de la voie perceptive, dont le seul but est de stocker des copies parfaites des stimuli à chaque instant. À notre connaissance aucune aire de ce type n'a été mise en évidence ni chez l'homme ni chez l'animal. Si l'on refuse cette possibilité, comme nous le faisons dans la suite de ce chapitre, il s'ensuit que les seuls stimuli directement accessibles pour l'apprentissage sont les stimuli de l'instant présent [2]. La mémoire est alors une conséquence de l'apprentissage de représentations temporelles et non un outil : on ne peut "rejouer" une séquence mémorisée que si l'on a auparavant appris une représentation condensée de cette séquence et non apprendre une représentation de cette séquence en "rejouant" une mémorisation de la séquence brute. Même si l'expérience suivante n'élimine pas la possibilité d'une zone de "copie" inconsciente de séquences de stimuli, on peut se convaincre très facilement du fait que la mémoire (consciente) que l'on a des stimuli est bien apprise et n'est pas une simple copie : il suffit d'écouter quelques instants la radio dans sa langue maternelle, puis dans une langue étrangère non maîtrisée. Au bout de quelques secondes, n'importe qui arrivera à restituer les stimuli auditifs correspondant à sa langue maternelle, mais sera bien incapable de restituer ceux en langue étrangère.

Cette simple remarque élimine d'emblée un très grand nombre d'algorithmes qui nécessitent d'accéder aux stimuli bruts sur une large fenêtre temporelle. C'est le cas par exemple des approches utilisant de la rétropropagation à travers le temps [3] (Werbos 1990) qui nécessite de se rappeler tous les éléments d'une séquence afin de calculer l'erreur et les modifications synaptiques. Les "réseaux à échelles spatio-temporelles multiples" proposés par Jun Tani par exemple fonctionnent selon ce principe (voir par exemple (Yamashita et Tani 2008 ; Jung et al. 2014)). La rétropropagation à travers le temps étant de plus sujette à l'évanescence du gradient et étant coûteuse en temps de calcul lorsqu'elle est appliquée sur de larges fenêtres temporelles, certaines approches contournent ce problème en remplaçant la récurrence par l'utilisation de la concaténation de plusieurs pas de temps en entrée (Taylor et al. 2006 ; Mohamed et al. 2012). Ces approches sont alors contraintes à ne pouvoir apprendre que des relations restreintes à la durée de la fenêtre temporelle utilisée.

D'autres approches ne nécessitent pas de stocker les stimuli bruts, mais une représentation de ces stimuli, avant de pouvoir les catégoriser. C'est le cas par exemple de (Griffith et al. 2012), où les auteurs commencent par encoder les stimuli à chaque pas de temps

2. S'il est facile en informatique de définir la durée d'un "instant" comme étant le pas de temps de l'expérience ou de la boucle de contrôle, il est bien plus dur d'en donner une définition dans un cadre biologique. Dans le cas de la vision, (Fraisse 1966) a par exemple montré que la perception de la simultanéité de deux apparitions visuelles (points lumineux ou lettres) est largement influencée par la durée entre l'apparition du premier stimulus et la disparition du second stimulus, alors que la durée propre de chaque stimulus et la présence ou non d'un intervalle entre la disparition du premier stimulus et l'apparition du second n'a que peu d'influence : les deux stimuli sont perçus comme simultanés si la durée totale est inférieure à environ 100 millisecondes. Une telle mesure fait cependant appel à la *perception* et implique donc une étape de représentation, ce qui ne permet par conséquent pas d'en faire une définition d'*instant* selon le point de vue que nous défendons. Cela permet néanmoins de donner un ordre de grandeur maximal.

3. *Backpropagation through time*, BPTT.

par une carte auto-organisatrice, avant de se servir de la concaténation des coordonnées du neurone le plus actif pour chaque pas de temps comme représentation de la séquence dans sa globalité. Une fois qu'un nombre suffisant de séquences a été enregistré, celles-ci peuvent être catégorisées à l'aide d'un algorithme de clustering spectral utilisant une distance d'édition comme mesure de similarité entre séquences. Une telle approche ne nécessite donc pas de stocker les stimuli bruts, mais n'est pas pour autant plus convaincante. En effet, d'une part le stockage des séquences sous une forme aussi dispendieuse en ressources est quasiment équivalente au stockage des stimuli bruts. D'autre part, cette approche suppose également que chaque séquence a été segmentée avant l'apprentissage, ce qui n'est pas le cas en situation réelle où le flux sensoriel est continuel. Enfin, l'utilisation d'une distance d'édition et d'un clustering spectral semble excessivement gourmande en calculs (cela nécessite de comparer toutes les séquences stockées deux à deux, à l'aide d'une distance elle-même non triviale).

5.2.2 Le don de voyance

D'autres travaux comme (Liwicki et al. 2007 ; Graves et Jaitly 2014) par exemple supposent que l'on a accès à chaque instant à *l'ensemble* d'une séquence dans laquelle on peut propager des signaux à la fois depuis le passé et depuis le futur, pendant l'apprentissage mais aussi pendant l'utilisation de ce réseau après apprentissage (les stimuli futurs influencent alors la manière dont sont perçus les stimuli présents). De telles approches sont évidemment inapplicables à des utilisations temps réel requises par la robotique autonome, puisque la nécessité de connaître le futur sur un nombre important de pas de temps entraîne obligatoirement un délai incompressible entre le stimulus et son analyse (et donc sa perception).

Dans (Contardo et al. 2014), les auteurs apprennent des états latents à partir d'observations, sous les contraintes qu'un état latent z_t doit permettre de reconstruire l'observation o_t de l'instant t, mais également de prédire l'état latent suivant z_{t+1}. Avec cette approche, chaque état latent est défini par rapport au futur et non par rapport au passé. Ceci pose un problème évident pour une utilisation temps réel, dans un contexte robotique par exemple. Du fait qu'à partir d'un état z_t plusieurs états z_{t+1} soient atteignables (ce nombre croissant exponentiellement lorsque le nombre de pas de temps augmente) impose par exemple de recalculer régulièrement une inférence globale sur tous les états passés afin de les mettre à jour pour pouvoir avoir une meilleure estimation de l'état courant (prédit par les états passés, eux-mêmes définis par les états qui leur sont postérieurs) et de mieux prédire ainsi l'état suivant, comme le montrent les résultats des auteurs. On remarque alors que cette inférence globale, outre le fait d'être coûteuse, requiert d'avoir stocké toutes les observations passées, ajoutant les défauts critiqués dans la section précédente à cette approche.

Notre critique ne remet pas en cause le fait que chez l'homme la perception d'un stimulus à un instant t puisse modifier la représentation (et donc la perception) des stimuli aux instants précédents. De tels mécanismes peuvent notamment être envisagés pour expliquer le phénomène de restauration auditive décrit précédemment. Nous pensons cependant que les modifications des représentations sont induites par des mécanismes hiérarchiques (la

perception d'un nouveau stimulus modifie la représentation haut niveau de la séquence et peut alors induire une reconstruction différente des stimuli passés) et non d'une mise en attente du stimulus jusqu'à réception des stimuli suivants.

5.2.3 Avec un peu de chance...

Ni don de voyance, ni problème de rétropropagation du gradient : c'est ce que promettent les approches par "réservoir" comme les réseaux *echo state* (Jaeger 2001) ou les *liquid-state machines* (Maass et al. 2002) que nous avons brièvement présentés au chapitre 3. En effet, ces approches reposent sur l'utilisation d'un ensemble de neurones connectés entre eux aléatoirement lors de l'initialisation, sans modification de ces connexions au cours de l'apprentissage. L'idée centrale étant alors que les propriétés temporelles des séquences interagissent avec la dynamique propre du réservoir pour créer des motifs d'activité spatiale qui peuvent à leur tour être "décodés" par une couche de sortie non récurrente. Pour être capable de capturer des propriétés temporelles sur d'assez longues durées, il faut que la matrice de poids du réservoir ait un rayon spectral très légèrement inférieur à 1 : trop faible, l'information entrante s'évanouit très rapidement dans le réseau, supérieur à 1, l'activité globale du réseau croît de manière exponentielle, jusqu'à saturation (Jaeger 2001).

Les capacités de ce type de réseaux sont donc entièrement déterminées dès l'initialisation. En particulier, il n'y a aucune garantie que la dynamique interne du réseau puisse permettre de séparer clairement deux séquences temporelles très différentes, ni qu'au contraire, deux séquences temporelles très proches ne se traduisent par deux dynamiques très différentes, empêchant une bonne généralisation. Par exemple dans le cas du phénomène de la restauration acoustique, le remplacement d'une partie de la séquence par un bruit blanc est tout à fait susceptible de provoquer une perturbation importante de la dynamique du réseau, modifiant en profondeur le motif d'activité obtenu à la fin de la séquence.

Il faut cependant noter que certaines approches tentent d'introduire de la plasticité au sein des réservoirs, notamment à l'aide de mécanismes hebbiens (Babinec et Pospíchal 2005). Leur impact est toutefois encore peu clair.

5.2.4 Quelques pistes intéressantes

Nous avons vu que la rétropropagation à travers le temps ne convient pas pour une approche développementale. Un autre algorithme a toutefois été proposé pour entraîner les réseaux récurrents : le *real-time recurrent learning* (Williams et Zipser 1989). Celui-ci consiste à accumuler pour chaque neurone et pour chaque poids synaptique du réseau le changement d'activité qui aurait été induit par une légère modification du poids synaptique dès le début de la séquence, ce qui permet à la fin de la séquence de modifier les poids synaptiques selon l'erreur produite au cours de la séquence. Cet algorithme a par conséquent le défaut d'être coûteux en terme de mémoire requise pour stocker ces informations. Les réseaux LSTM (*Long Short Term Memory*) (Hochreiter et Schmidhuber 1997a) en proposent une variante (mélangée à une rétropropagation à travers le

temps tronquée à un seul pas de temps) qui réduit ce coût. En utilisant cette technique[4], les réseaux LSTM sont capables d'apprendre des corrélations temporelles sur de longues durées tout en évitant les écueils de la rétropropagation du gradient (Hochreiter et Schmidhuber 1997b ; Graves et al. 2004). Si les réseaux LSTM sont performants dans des tâches supervisées, peu de travaux se sont intéressés à notre connaissance au cas non supervisé. Les auteurs de (Klapper-Rybicka et al. 2001) par exemple travaillent sur la représentation de séquences par des réseaux LSTM, mais supposent au préalable que les séquences ont déjà été correctement segmentées.

La "*slow features analysis*" (Wiskott et Sejnowski 2002 ; Kompella et al. 2011b) vise à apprendre une représentation du signal d'entrée à l'aide de neurones cachés dont les activités varient lentement dans le temps et sont indépendantes les unes des autres. Une telle approche donne des résultats intéressants : le travail de (Kompella et al. 2011a) montre par exemple qu'il est possible de faire émerger de manière autonome une notion de position dans l'espace en utilisant une vidéo brute en entrée (encodée par un autoencodeur). En effet, les variations rapides et chaotiques des pixels au bas niveau de l'image lors d'un déplacement dans l'environnement peuvent s'expliquer simplement par la position d'un objet par rapport à la caméra (qui varie de manière beaucoup plus douce que les pixels de l'image), que l'algorithme est donc conduit à extraire.

Contrairement à (Contardo et al. 2014) que nous avons critiqué précédemment, l'approche de (Gisslén et al. 2011) impose aux états latents d'être capables de reconstruire à la fois l'observation courante et l'état latent précédent (au lieu de suivant dans (Contardo et al. 2014)). Dans cette approche, les états latents ne dépendent pas du futur (et ne nécessitent donc pas d'inférence coûteuse) et permettent de plus de reconstruire l'ensemble des observations passées sans avoir besoin de les stocker (il suffit de remonter la chaîne des états latents de manière itérative). Les auteurs ont notamment montré que cette approche permettait de résoudre des tâches de type labyrinthe en utilisant les états latents dans un apprentissage par renforcement classique, en fournissant uniquement l'image obtenue par une caméra embarquée sur le robot en tant qu'observation.

Un des sujets peu abordé est le passage d'un flux continu d'informations sensorielles à une représentation haut niveau de ce flux qui requiert à la fois la segmentation de ce flux et la représentation de chacun des segments ainsi extraits. Ce problème est discuté dans la prochaine section.

5.3 Vers une approche intégrée de la temporalité

Comme nous l'avons annoncé en introduction de ce chapitre, nous n'avons pas de solution adéquate à proposer au problème de la temporalité. Nous allons toutefois présenter dans cette partie deux travaux liminaires qui s'attaquent à deux sous-problèmes distincts.

Nous allons donc commencer par présenter l'approche théorique du codage de séquences temporelles que nous avons adoptée et identifier les deux sous-problèmes sur lesquels nous

4. (Graves et Schmidhuber 2005) ont toutefois montré que les performances de ce type de réseau pouvaient être améliorées en utilisant une rétropropagation à travers le temps complète, qui est de fait la méthode actuellement utilisée dans la majorité des travaux avec ces réseaux.

nous sommes penchés.

Nous présenterons les deux travaux correspondants dans un second temps. Le premier consiste à apprendre des représentations de transformations orthogonales, c'est-à-dire en faisant l'hypothèse que l'information contenue dans deux stimuli consécutifs reste constante. C'est le cas par exemple en première approximation des stimuli visuels, pour lesquels un mouvement de translation ou de rotation ne modifie pas les objets perçus et donc laisse constante la quantité d'information nécessaire pour décrire la scène. Le second quant à lui s'attache à étudier l'apprentissage de séquences distinctes par un même réseau, chaque séquence pouvant être contextualisée par un jeu de paramètres afin d'expliquer les variations intrinsèques d'une catégorie donnée de séquences (par exemple les différentes manières de prononcer un mot).

5.3.1 Quelques éléments d'architecture

Considérons l'exemple de la vision et imaginons deux scènes distinctes : l'une dans laquelle on observe une balle rouge rouler sur une table, l'autre dans laquelle on observe une balle verte rouler sur la même table. Du point de vue de la perception au plus bas niveau, ces stimuli sont très différents : ce ne sont pas les mêmes cellules qui sont excitées dans les deux cas au niveau de la rétine. Une approche mathématique naïve consiste à apprendre directement chaque séquence par une fonction de la forme :

$$x_t = f(t, \Theta_x) \tag{5.1}$$

où x correspond à la séquence à encoder, t au pas de temps considéré, et Θ_x désigne un jeu de paramètres de la fonction f permettant de décrire la séquence considérée. Cette formule fait ressortir deux problèmes. D'une part, en utilisant explicitement une variable temporelle t, elle nécessite d'introduire une référence temporelle sous la forme d'une horloge interne indépendante. D'autre part, Θ_x doit être suffisamment complexe pour être capable de coder toute l'information contenue dans la séquence.

L'existence d'horloges internes indépendantes a longtemps été défendue, sous différentes formes : véritable horloge émettant des pics réguliers pouvant être comptés (voir par exemple (Allan 1979) pour une revue de nombreux modèles de ce type) ou encore codage spectral à l'aide de neurones ayant des réponses temporelles de différentes durées (Grossberg et Schmajuk 1989). Comme expliqué dans (Mauk et Buonomano 2004), ces méthodes souffrent néanmoins de défauts importants. Premièrement, la possibilité de coder des informations hiérarchiques est mal prise en compte (pour différencier par exemple deux séquences de trois stimuli identiques, mais séparés dans un cas de deux périodes de 50 et 150ms, dans l'autre cas de 150 et 50ms : les mêmes neurones "compteurs" seraient activés dans les deux cas, et il faudrait un réseau plus haut niveau pour mesurer l'écart temporel entre l'activation des deux, ce qui crée une régression infinie lorsque l'on poursuit le raisonnement). De même, se pose le problème de la "mise à zéro", c'est-à-dire la question de déterminer l'instant 0 à partir duquel les mesures temporelles commencent. Au contraire, les codages implicites s'appuyant sur des dynamiques neuronales d'ensembles semblent plus adaptés pour des motifs temporels complexes (Mauk et Buonomano 2004).

S'inspirant de tels modèles, une autre possibilité consiste à coder les séquences sous la forme :

$$x_t = f(x_{t-1}, \Theta_{t-1}) \tag{5.2}$$

$$\Theta_t = g(\Theta_{t-1}, x_t, x_{t-1}). \tag{5.3}$$

De cette manière, la dépendance au temps est implicite, et les paramètres Θ doivent encoder moins d'information sur la séquence (en particulier, ils n'ont plus à être capables d'encoder le contenu commun à x_t et x_{t-1}). De plus, l'équation (5.3) est compatible avec le cadre théorique du codage prédictif sur la base des erreurs de prédiction en considérant des fonctions de la forme

$$\Theta_t = g(\Theta_{t-1}, \underbrace{x_t - f(x_{t-1}, \Theta_{t-1})}_{\text{erreur de prédiction}}). \tag{5.4}$$

Nous laissons temporairement de côté l'équation (5.3), nous y reviendrons à la fin de cette section, pour nous pencher sur l'équation (5.2). Sa forme est très générale. Si nous ajoutons l'hypothèse que le signal x est discrétisé selon un pas de temps suffisamment petit par rapport à la période caractéristique des variations et que la fonction f est dérivable, il est possible de linéariser l'équation (5.2) en

$$x_t = T_{\Theta_{t-1}} x_{t-1} + \epsilon_t \tag{5.5}$$

où $T_{\Theta_{t-1}}$ désigne une matrice de transformation permettant de passer de x_{t-1} à x_t et ϵ_t désigne un reste non représentable par la transformation T. Le but de l'apprentissage est alors d'atteindre $\epsilon = 0$, c'est-à-dire de minimiser

$$||x_t - T_{\Theta_{t-1}} x_{t-1}||^2. \tag{5.6}$$

Dans le cas général, une infinité de transformations linéaires T permettent de passer de x_{t-1} à x_t. Une telle approche nécessite donc une contrainte supplémentaire pour rendre l'apprentissage viable. Un critère peut par exemple être de minimiser la norme de la matrice T, ou encore de minimiser le nombre de paramètres Θ permettant de coder l'ensemble des transformations rencontrées au cours de l'apprentissage.

Nous allons pour notre part faire une hypothèse sur le type de transformations à utiliser : nous allons contraindre les matrices T à être orthogonales, c'est-à-dire telles que

$$T_{\Theta_t}^\top T_{\Theta_t} = T_{\Theta_t} T_{\Theta_t}^\top = I \tag{5.7}$$

où I est la matrice identité.

La validité et la pertinence de ce choix sont discutables. En effet, l'utilisation de transformations orthogonales gomme la notion de causalité (toute transformation étant inversible) et n'est pas réaliste dans de nombreux cas (par exemple lorsqu'un nouvel objet apparaît dans le champ visuel, il y a rupture d'orthogonalité dès que cette apparition est imprévisible). Cependant, nous pensons que ces ruptures d'orthogonalité peuvent jouer un rôle crucial dans la segmentation de séquences temporelles puisqu'elles correspondent

aux instants où une information nouvelle est ajoutée au flux perceptif. Utilisées conjointement à une architecture implémentant l'équation (5.4), les ruptures d'orthogonalité permettraient de créer un signal d'erreur important pouvant être à l'origine d'un changement conséquent des paramètres Θ. Cette idée rejoint certains travaux de représentation de séquences temporelles à partir des instants non prédictibles (Schmidhuber 1992a). Il faut de plus souligner que la restriction à des transformations orthogonales est compatible avec une compression de séquences temporelles utilisant le même type de transformations, puisque toute composition de transformations orthogonales est elle-même une transformation orthogonale (l'ensemble des matrices orthogonales doté de la loi de multiplication matricielle forme un groupe algébrique) :

$$x_T = T_{\Theta_T} \times T_{\Theta_{T-1}} \times \cdots \times T_{\Theta_0} x_0 = T x_0 \tag{5.8}$$

où T est également une matrice orthogonale. Une telle représentation est intéressante notamment pour être capable de simuler efficacement l'évolution de son environnement, sans avoir besoin de calculer chaque pas de temps, et permet également de pouvoir s'adapter à des variations de vitesse (la matrice globale T ne dépend pas du nombre de pas de temps) et à des séquences perçues de manière incomplète (la perception de x_0 et x_T peut être suffisante pour déduire T, avec néanmoins le risque de ne pouvoir distinguer entre plusieurs transformations produisant le même résultat à partir de x_0). La possibilité d'obtenir une telle représentation motive notamment notre choix de coder les transformations sous la forme de l'équation (5.5) en supprimant toute non-linéarité. Dans (Michalski et al. 2014), les auteurs argumentent de plus que l'utilisation de matrices orthogonales permet d'éviter en partie le problème du gradient évanescent/explosif puisque les transformations orthogonales préservent les normes.

L'apprentissage doit alors se faire sur deux aspects en parallèle : d'une part, trouver les matrices orthogonales permettant de transformer x_{t-1} en x_t, d'autre part trouver un codage efficace de ces matrices par des paramètres Θ. Ce sera l'objet du travail présenté à la section 5.3.2. Un troisième aspect pourrait également être ajouté : au lieu de travailler sur les stimuli bruts x_t, il serait possible de travailler sur un premier encodage de ces stimuli. L'apprentissage en parallèle d'un encodage des stimuli et des transformations peut permettre de rendre valide la linéarisation de l'équation (5.2) sur de plus grandes fenêtres temporelles. De plus, l'apprentissage parallèle des représentations et des transformations a déjà été utilisé avec succès par plusieurs auteurs (voir par exemple (Schmidhuber 1992b ; Gisslén et al. 2011 ; Wahlström et al. 2014)), généralement sans faire l'hypothèse de linéarisation de l'équation (5.5), parmi lesquels les auteurs de (Wahlström et al. 2014) ont montré qu'une telle approche permettait d'apprendre une structuration intéressante du flux perceptif comparé à un apprentissage séparé. Dans notre cas, grâce à l'utilisation de transformations orthogonales, nous pensons qu'une telle approche permettrait d'apprendre à mieux structurer la représentation des stimuli bruts en mettant l'accent sur l'apprentissage de représentations robustes dans le temps : si les transformations orthogonales gomment la notion de causalité comme expliqué ci-dessus, elles forcent cependant à apprendre des représentations corrélées dans le temps.

Nous nous sommes jusqu'à présent intéressés à l'équation (5.2), laissant de côté l'équation (5.3). Suivant l'hypothèse des sous-variétés, nous voulons que les séquences

soient codées par des sous-variétés dans un espace fonctionnel, c'est-à-dire introduire une paramétrisation de la fonction g permettant de modifier sa dynamique :

$$(\Theta_t, \Omega_t) = g(\Theta_{t-1}, \Omega_{t-1}, x_t, x_{t-1}). \tag{5.9}$$

Notons que cette dernière formule revient à distinguer deux types de paramètres, mais la forme de l'équation (5.9) est équivalente à celle de l'équation (5.3). L'hypothèse des sous-variétés implique qu'il existe un codage pour lequel Ω_t est constant au cours du temps pour une même séquence et qu'il peut être mis sous une forme similaire à la représentation utilisée dans l'architecture présentée au chapitre 4, c'est-à-dire sous la forme d'une représentation symbolique couplée à une paramétrisation de chaque symbole. Nos travaux présentés à la section 5.3.3 ne présentent pas une solution globale à ce problème, en particulier sur l'émergence d'un tel codage à partir de séquences temporelles. S'ils sont consacrés à un problème légèrement différent, à savoir la génération de trajectoires pour un bras robotique, ils permettront cependant de valider la possibilité d'utiliser un tel codage dans des réseaux de neurones pour générer différentes familles de séquences.

5.3.2 Représentation des transformations

La représentation des transformations est depuis longtemps populaire dans le domaine du traitement de vidéos. Récemment, les réseaux de neurones *gated* ont été appliqués avec succès pour la reconnaissance d'actions (Taylor et al. 2010) et au codage de relations entre différentes images (Memisevic et Hinton 2007 ; Memisevic 2012b).

Dans ses travaux, Roland Memisevic a fait le lien entre certains algorithmes impliquant une sommation (*pooling*) d'interactions multiplicatives et l'apprentissage de transformations orthogonales (Memisevic 2012a). Dans la suite de cette section, nous exposons une extension de ces travaux intégrant la notion de transformations orthogonales au sein de la structure du réseau, publiée dans l'article :

> Alain Droniou et Olivier Sigaud (2013). « Gated Autoencoders with Tied Input Weights ». Dans : *Proceedings of International Conference on Machine Learning*, p. 154–162.

Travaux de (Memisevic 2012a) et transformations orthogonales

Les travaux de Memisevic dont nous sommes parti considèrent l'apprentissage d'une transformation linéaire T entre deux stimuli $\mathbf{x}$ et $\mathbf{y}$

$$\mathbf{y} = T\mathbf{x} \tag{5.10}$$

Dans le cas où T est une matrice orthogonale, elle peut être diagonalisée par :

$$T = UDU^{\top} \tag{5.11}$$

où D est une matrice diagonale contenant les valeurs propres de T, qui ont la propriété d'être toutes des nombres complexes de module 1. La matrice U est quant à elle composée des vecteurs propres correspondants et "$.^{\top}$" représente la matrice adjointe (matrice

transposée conjuguée). Ces vecteurs propres ont la propriété d'être identiques pour deux transformations qui commutent, ce qui permet un partage des ressources efficace pour la représentation de familles de transformations (par exemple dans le cas visuel, pour représenter toute la famille des translations à l'aide d'un seul jeu de vecteurs propres). Comme T est une matrice à coefficients réels, on a par ailleurs $U^{-1} = U^\top$ (car U est alors également la matrice de passage de T à D), ce qui permet de réécrire l'équation 5.10 sous la forme

$$U^\top \mathbf{y} = DU^\top \mathbf{x}. \tag{5.12}$$

On voit ici apparaître le fait qu'il est possible de retrouver les valeurs propres en mesurant l'angle de la rotation entre les projections de $\mathbf{x}$ et $\mathbf{y}$ par $U^\top$ (une multiplication par un complexe de module 1 étant en effet équivalente à une rotation dans le plan complexe). Memisevic a montré que le produit scalaire des projections normalisées fournit directement le cosinus de cet angle, mais que la normalisation est toutefois problématique pour des valeurs proches de zéro et peut mener à de fausses détections.

Le problème est reformulé dans (Memisevic 2012b,a) comme une tâche de détection consistant à apprendre des filtres d'entrée et de sortie U et V tels que V fusionne D et U. L'équation 5.12 devient alors

$$U^\top \mathbf{y} = V^\top \mathbf{x} \tag{5.13}$$

et les corrélations entre les projections par U et V permettent de détecter la présence d'une transformation. Cependant, les valeurs de ces projections dépendent des valeurs de $\mathbf{x}$ et $\mathbf{y}$, ce qui oblige à sommer la réponse de plusieurs filtres représentant la même transformation mais couvrant l'espace des valeurs possibles pour $\mathbf{x}$ et $\mathbf{y}$ (on parle de filtres en quadrature). De plus, le passage à l'équation 5.13 a supprimé toute référence aux transformations orthogonales, puisque les valeurs prises par U et V ne sont plus contraintes (en particulier si $U^\top$ n'est pas inversible, alors la transformation entre $\mathbf{x}$ et $\mathbf{y}$ n'est plus représentable selon le formalisme de l'équation 5.10).

Apprentissage de transformations orthogonales

Notre travail a donc consisté à repartir de l'équation 5.12 pour apprendre à extraire une représentation efficace de la matrice D.

Puisque la matrice D ne contient que des valeurs complexes de module 1, il suffit de calculer le cosinus et le sinus de l'angle entre les projections de $\mathbf{x}$ et de $\mathbf{y}$ pour la caractériser. Par la suite, nous noterons $\mathbf{u_x}$ et $\mathbf{u_y}$ les projections de $\mathbf{x}$ et $\mathbf{y}$ par un vecteur propre de T (colonnes de U). En considérant ces valeurs complexes comme des vecteurs dans le plan, le produit scalaire $\mathbf{u_x}.\mathbf{u_y}$ et la norme du produit vectoriel $\mathbf{u_x} \times \mathbf{u_y}$ donnent respectivement les valeurs du cosinus et du sinus à une constante multiplicative $||\mathbf{u_x}||\,||\mathbf{u_y}||$ près :

$$\begin{aligned} \mathbf{u_x}.\mathbf{u_y} &= ||\mathbf{u_x}||.||\mathbf{u_y}||.cos(\mathbf{u_x}, \mathbf{u_y}) \\ \mathbf{u_x} \times \mathbf{u_y} &= ||\mathbf{u_x}||.||\mathbf{u_y}||.sin(\mathbf{u_x}, \mathbf{u_y}). \end{aligned} \tag{5.14}$$

Du point de vue des nombres complexes, on a de plus

$$\begin{aligned} \mathbf{u_x}.\mathbf{u_y} &= Reel(\mathbf{u_y}\overline{\mathbf{u_x}}) \\ \mathbf{u_x} \times \mathbf{u_y} &= Imag(\mathbf{u_y}\overline{\mathbf{u_x}}). \end{aligned} \tag{5.15}$$

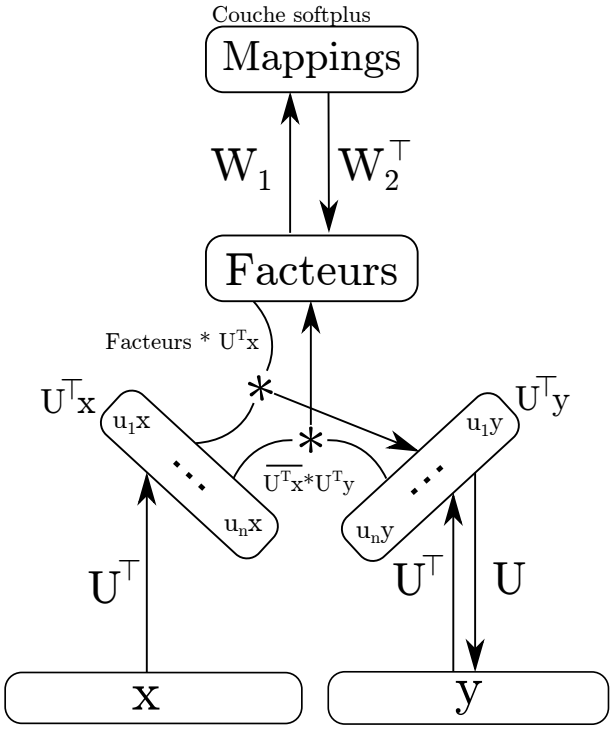


Figure 5.1 – *Architecture pour l'apprentissage de transformations orthogonales.*

où $\overline{\cdot}$ désigne le conjugué et *Reel* et *Imag* désignent respectivement les parties réelles et imaginaires. Ceci nous amène à introduire une couche de *facteurs* sous la forme

$$\mathbf{f} = U^\top \mathbf{y} * \overline{U^\top \mathbf{x}} \tag{5.16}$$

où $*$ représente la multiplication terme à terme. Cette couche de facteurs peut à son tour être représentée par une autre couche, permettant notamment de compresser cette information. Le parallèle avec une couche *gated* classique est immédiat, comme illustré figure 5.1.

La représentation haut niveau (*mappings*) est donnée par

$$\mathbf{m} = \sigma_+ \left(W_1 \mathbf{f} + \mathbf{b}_{\text{mappings}} \right) \tag{5.17}$$

où σ_+ est la fonction *softplus*. Nous faisons le choix de la fonction *softplus* au lieu de la fonction sigmoïde utilisée par les travaux de (Memisevic 2012b,a) décrits précédemment car nous cherchons à représenter le spectre des matrices de transformations et non des détecteurs de corrélations pour lesquels l'interprétation probabiliste de l'activité permise par la fonction sigmoïde paraît plus adaptée.

La reconstruction de $\mathbf{y}$ peut alors être calculée avec la formule

$$\mathbf{r} = U(W_2^\top \mathbf{m} * U^\top \mathbf{x}) + \mathbf{b}_y. \tag{5.18}$$

Nous utilisons deux matrices différentes W_1 et W_2 entre les facteurs et la couche haut niveau pour deux raisons. Premièrement, la reconstruction calculée à l'aide de

l'équation (5.18) n'est exacte que si $W_2^\top \mathbf{m}$ est un vecteur égal à la diagonale de D (équations 5.11 et 5.12). En particulier $W_2^\top \mathbf{m}$ ne peut être confondu avec $\mathbf{f}$ calculé d'après l'équation (5.16), puisque d'après l'équation 5.12 nous avons $\mathbf{u_y}\overline{\mathbf{u_x}} = d||\mathbf{u_x}||^2$. L'utilisation de deux matrices W_1 et W_2 permet donc de moduler la reconstruction de la couche de facteurs de telle manière qu'elle soit différente de sa valeur calculée à partir des entrées $\mathbf{x}$ et $\mathbf{y}$. Deuxièmement, lorsque plusieurs transformations qui ne commutent pas sont représentées par le même réseau, la matrice U contient plusieurs sous-ensembles de vecteurs propres. Étant donnée une transformation pour laquelle un seul sous-ensemble est requis, d'autres vecteurs propres peuvent produire une activité non nulle au niveau de la couche de facteurs. La projection dans une couche supérieure $\mathbf{m}$ permet notamment de filtrer ces activités, de telle manière qu'un neurone de la couche $\mathbf{m}$ peut être activé et inhibé par différents sous-ensembles de vecteurs propres et l'utilisation d'une deuxième matrice permet de n'induire une activité lors de la reconstruction que sur un sous-ensemble bien défini de facteurs.

Ce réseau peut être entraîné selon la méthode standard des autoencodeurs, c'est-à-dire avec le but de minimiser l'erreur de reconstruction de $\mathbf{y}$: $||\mathbf{y} - \mathbf{r}||^2$.

Passage des complexes aux réels

Toute la description de l'architecture effectuée dans la section précédente implique de se placer dans le corps des nombres complexes. Les réseaux de neurones ont cependant été développés majoritairement sur le corps des nombres réels, et peu d'études ont été menées sur des théories plus générales (voir par exemple (Baldi 2012 ; Baldi et al. 2012)). Bien que l'extension aux nombres complexes puisse sembler triviale, certains problèmes apparaissent comme par exemple la non définition de la fonction sigmoïde en $(2k+1)i\pi$ (et l'apparition d'un gradient explosif au voisinage de ces points).

C'est pourquoi nous exposons dans cette section une implémentation de l'architecture précédente sur le corps des réels.

La solution immédiate consiste évidemment à séparer parties réelles et imaginaires. Cependant, la multiplication terme à terme opérée par les couches *gated* ne permet alors pas de simuler la multiplication complexe. Pour ce faire, nous introduisons donc une duplication des facteurs de telle sorte que la première moitié puisse correspondre à une multiplication terme à terme classique tandis que la deuxième moitié soit "croisée" de manière à permettre une multiplication entre les parties réelles et imaginaires de chaque projection (voir figure 5.2). Enfin, une sommation bien choisie permet de calculer directement les valeurs des produits scalaires et produits vectoriels introduits dans l'équation (5.14), tout en permettant de retrouver une taille de la couche de facteurs égale à celle utilisée avant duplication. L'ensemble du processus est illustré figure 5.2.

Son implémentation peut se faire à l'aide uniquement de multiplications matricielles, ce qui permet d'utiliser n'importe quelle bibliothèque standard utilisée pour la programmation de réseaux de neurones [5].

5. Nous utilisons dans notre cas la bibliothèque python theano (`http://deeplearning.net/software/theano/`) (Bergstra et al. 2010).

Soient l la taille de la couche de facteurs avant duplication et I_l la matrice identité de taille l. Le "croisement" de la seconde moitié des facteurs consiste à simplement dupliquer la projection de $\mathbf{x}$ en la multipliant par

$$E_1 = \begin{pmatrix} I_l \\ I_l \end{pmatrix} \tag{5.19}$$

tandis que la projection $\mathbf{y}$ est inversée par multiplication avec

$$E_2 = \begin{pmatrix} I_l \\ B_l \end{pmatrix} \tag{5.20}$$

où B_l est une matrice diagonale par blocs $B = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$. L'étape de sommation est alors directement effectuée par multiplication avec la matrice P_l de taille $l \times 2l$

$$P_l = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 1 & 0 & 0 & \dots \\ & & & \ddots & & & \end{pmatrix}. \tag{5.21}$$

Durant la phase de reconstruction, le "croisement" des facteurs nécessite quant à lui un réordonnancement des facteurs par multiplication avec

$$E_3 = \begin{pmatrix} R_l \\ B_l R_l \end{pmatrix} \tag{5.22}$$

où

$$R_l = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots & -1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots & 0 & -1 & 0 & \dots \\ & & & & \vdots & & & \end{pmatrix}. \tag{5.23}$$

Les équations (5.17) et (5.18) peuvent donc finalement se réécrire

$$\begin{aligned} \mathbf{m} &= \sigma_+ \left(W_1 P((E_1 U^\top \mathbf{x}) * (E_2 U^\top \mathbf{y})) + \mathbf{b}_{\text{mappings}} \right) \\ \mathbf{r} &= U P((E_3 W_2^\top \mathbf{m}) * (E_1 U^\top \mathbf{x})) + \mathbf{b}_y \end{aligned} \tag{5.24}$$

Expériences

Les expériences suivantes visent à illustrer le fonctionnement de l'architecture proposée. Nous utilisons pour cela des images de 13x13 pixels représentant des points tirés aléatoirement selon une distribution normale et de manière indépendante. Nous appliquons alors à ces images différentes transformations :

– des combinaisons de translations horizontales et verticales d'amplitude tirée uniformément entre +/-3 pixels ;

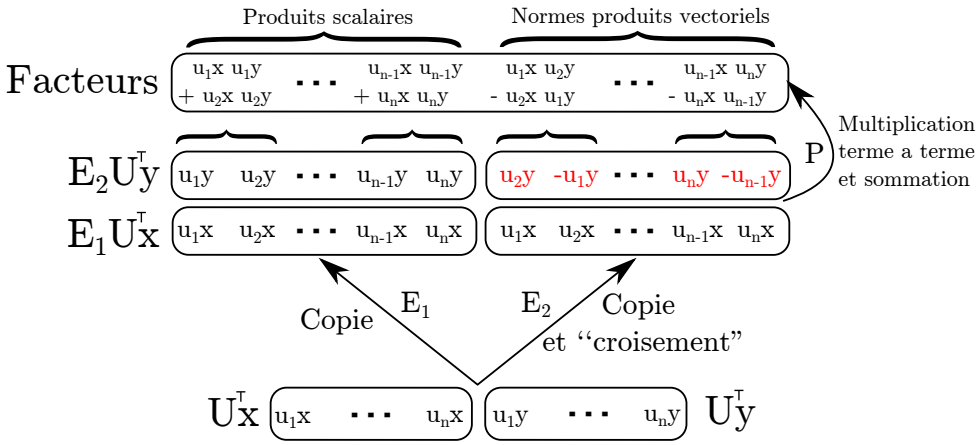


Figure 5.2 – *L'implémentation concrète du réseau utilise des connexions spécifiques codées par des matrices constantes pour simuler les opérations sur les nombres complexes.*

Table 5.1 – *Paramètres communs à toutes les expériences.*

Paramètre	Valeur
Corruption	Masquage à 0
Taux de corruption	30%
Taille de minibatch	100
Taille d'entrée	13×13 pixels
Momentum	0.9
Taux d'apprentissage	$\frac{0.005}{max(1, floor(\text{epoch} * 0.1))}$

– des rotations d'angles tirés uniformément entre -50 et +50 degrés.

Les pixels de l'image ainsi produite qui ne sont pas couverts par l'image initiale sont à leur tour tirés aléatoirement selon la même loi normale. Les images sont alors transformées en vecteurs en concaténant leurs lignes. Pour chaque expérience décrite par la suite, nous construisons un ensemble d'entraînement constitué de 100 000 paires d'images uniformément réparties selon les transformations considérées. L'erreur de reconstruction que nous rapportons dans les résultats est donnée par la moyenne arithmétique de $||\mathbf{y} - \mathbf{r}||^2$.

Nous utilisons pour l'entraînement du réseau la même structure que celle utilisée par Memisevic et disponible en ligne [6], dont nous tirons également les résultats de comparaison avec l'architecture proposée dans (Memisevic 2012b) que nous désignons par l'acronyme "CGA" (*Classical Gated Autoencoder*). Nous désignons par "CGA-softplus" une variante de cette architecture utilisant la fonction d'activation *softplus* au lieu de la fonction sigmoïde. Les principaux paramètres de l'apprentissage sont donnés dans le tableau 5.1. Nous avons choisi le taux d'apprentissage de telle sorte que l'erreur de reconstruction ne présente pas d'instabilité visible durant l'apprentissage. L'apprentissage est également régularisé avec la technique du débruitage en utilisant un taux de masquage de 30%.

6. `http://learning.cs.toronto.edu/~rfm/code/rae/index.html`

De plus, nous rappelons que notre approche mathématique suppose des transformations orthogonales, ce qui n'est pas le cas des données que nous utilisons pour l'apprentissage (les pixels en bordure sont générés aléatoirement après transformation). Durant l'apprentissage, nous pondérons donc l'erreur de reconstruction par un masque gaussien centré sur l'image conçu de telle sorte que les pixels situés sur les bords aient un poids moins important (environ moitié moindre) lors du calcul de l'erreur.

Première expérience La première expérience a uniquement pour but de tester la validité de notre approche et vérifier que le réseau se comporte bien comme attendu du point de vue mathématique.

Nous considérons pour cela la propriété de base des transformations orthogonales, à savoir $TT^\top = I$, ce qui permet de déduire

$$T^\top = U\overline{D}U^\top \tag{5.25}$$

à partir de l'équation 5.11. Il est alors possible de tester les transformations inverses, comme des rotations d'angles $+\theta$ et $-\theta$ et de vérifier que les valeurs des facteurs correspondants sont bien conjuguées, c'est-à-dire que les produits scalaires sont identiques tandis que les produits vectoriels sont de valeurs opposées.

Nous entraînons donc un réseau utilisant 400 neurones facteurs et 40 neurones *softplus* sur un ensemble de rotations, avant de comparer les valeurs des facteurs pour des rotations opposées. Nous moyennons les résultats sur 100 rotations de même angle appliquées à des images différentes, pour tous les angles entiers entre -50 et +50 degrés.

Comme prévu par notre modèle, la figure 5.3 montre que les produits scalaires sont corrélés tandis que les produits vectoriels sont anti-corrélés. Le réseau apprend donc bien une représentation cohérente avec notre description mathématique, même si les conditions théoriques d'orthogonalité ne sont pas totalement remplies (à cause notamment du bruit de masquage et des pixels en bordure).

Deuxième expérience Cette deuxième expérience a pour but de valider le fait que les filtres appris dans la matrice U sont partagés par toutes les transformations qui commutent entre elles, et donc que l'architecture que nous proposons permet une meilleure généralisation que celle des CGA décrite dans (Memisevic 2012a) pour laquelle chaque filtre est spécifique à une transformation donnée.

Pour ce faire, nous entraînons le même réseau que précédemment sur un sous-ensemble de rotations, d'angles compris entre -50 et 50 degrés par pas de 10 degrés. Nous testons ensuite le réseau sur toutes les rotations d'angle entier compris entre -50 et 50 degrés. Comme précédemment, nous moyennons les résultats sur 100 rotations d'images différentes pour chaque angle.

La figure 5.4 représente les distances entre les activités de la couche *softplus* obtenues pour différents angles. L'algorithme CGA tend à créer des clusters autour des rotations présentes dans l'ensemble d'entraînement là où l'architecture proposée apprend une représentation plus régulière, mieux interpolée. Le comportement de l'algorithme CGA le rapproche des techniques du "plus proche voisin", même si cet effet est moins visible pour

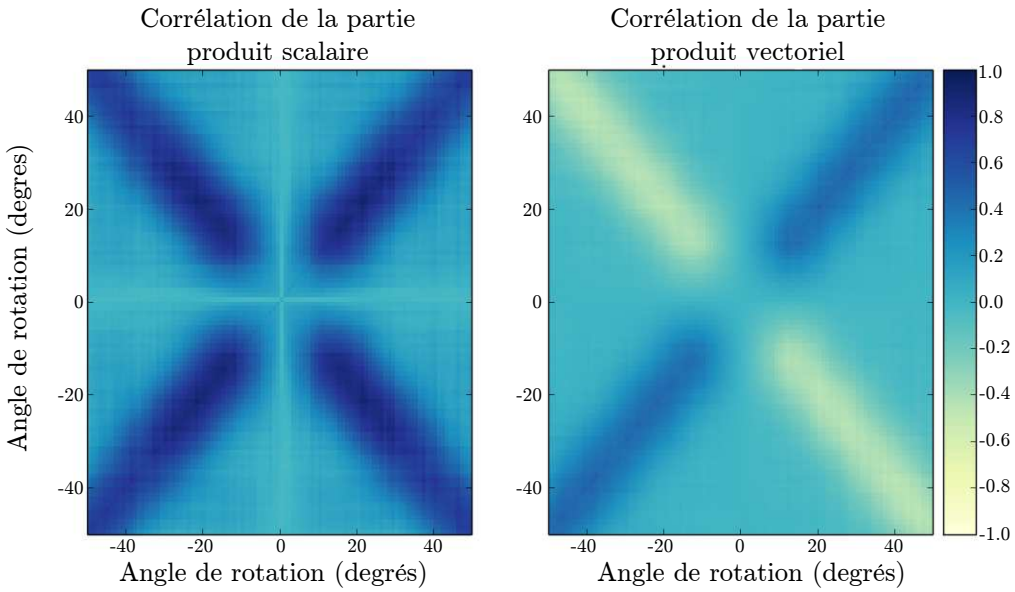


Figure 5.3 – *Corrélations de l'activité des neurones de la couche de facteurs pour des rotations comprises entre -50 et +50 degrés (représentées par la matrice de covariance). Comme attendu, le réseau apprend une partie correspondant au produit scalaire (corrélé pour des rotations opposées) et une autre correspondant au produit vectoriel (anti-corrélé).*

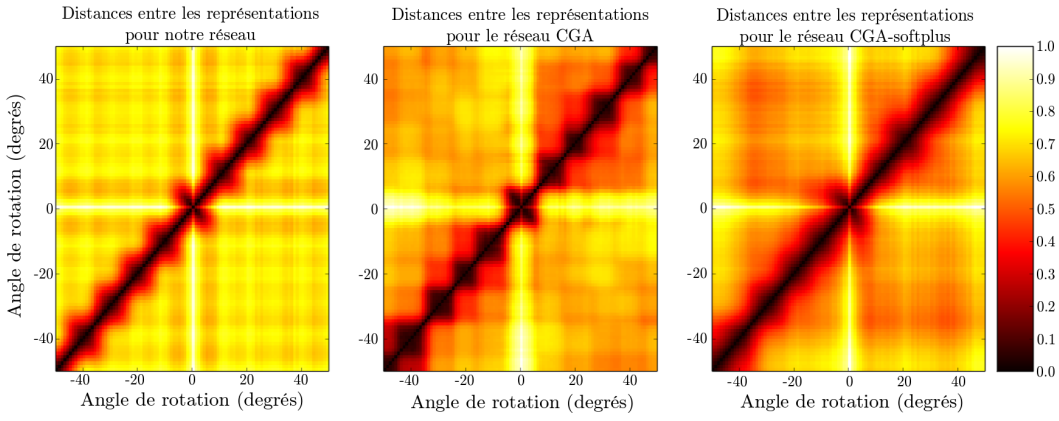


Figure 5.4 – *Distances calculées à partir de l'activité moyenne de la couche softplus pour des rotations comprises entre -50 et 50 degrés, à partir d'un réseau entraîné sur un sous-ensemble de seulement 11 angles équirépartis. Les distances ont été normalisées entre 0 et 1. La diagonale est mieux définie avec notre approche, ce qui montre que les représentations apprises sont plus discriminatives tout en permettant une meilleure généralisation à des angles non présents dans l'ensemble d'entraînement.*

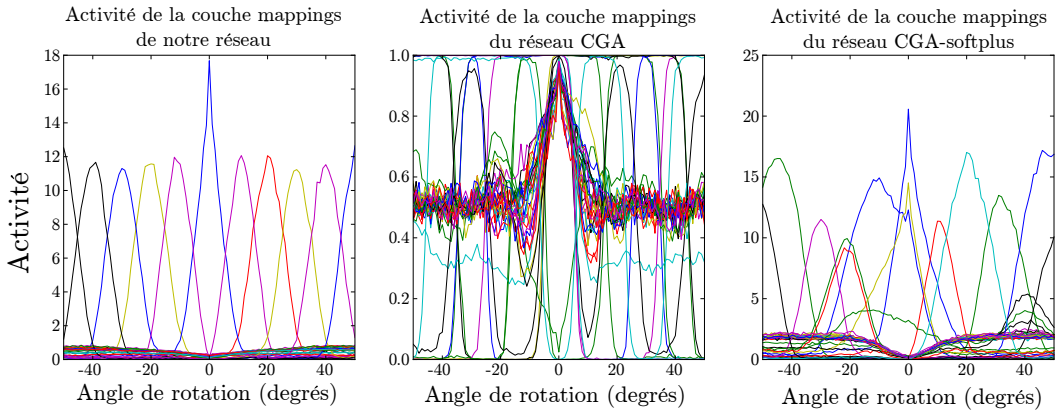


Figure 5.5 – *Activité des neurones de la couche supérieure pour des rotations comprises entre -50 et 50 degrés, pour un réseau entraîné sur un sous-ensemble de seulement 11 angles équirépartis. Cette figure montre la plus grande régularité de la représentation apprise par notre approche par rapport aux approches CGA et CGA-softplus.*

CGA-softplus. Ceci peut s'expliquer par le fait que la fonction sigmoïde a tendance à forcer l'activité à 0 ou 1 alors que la fonction *softplus* permet des variations plus régulières. Le pouvoir de discrimination de notre réseau reste cependant meilleur que les deux variantes de CGA : les distances sont plus grandes en dehors de la diagonale, et cette dernière est mieux définie.

Cette différence peut être analysée en regardant de manière plus détaillée l'activité des neurones de la couche supérieure du réseau, représentée figure 5.5. La représentation apprise par notre réseau est plus régulière que pour les deux variantes de CGA. En effet, un neurone de la couche se spécialise pour chaque rotation présente dans l'ensemble d'entraînement, là où la répartition semble beaucoup plus aléatoire dans les deux autres cas. Étant donné que seuls 11 angles différents sont présentés durant l'apprentissage pour 40 neurones *softplus*, il apparaît que 29 neurones ne sont pas utilisés et que leur activité reste proche de zéro, alors même qu'aucune contrainte de parcimonie n'a été introduite.

La spécialisation d'un neurone pour chaque angle soulève toutefois la question de sa pertinence pour la propriété recherchée de généralisation. En supposant que les poids appris pour chaque neurone correspondent au spectre de la rotation pour laquelle la réponse du neurone est maximale (ce qui est cohérent avec l'étude mathématique confirmée par l'expérience précédente et le fait qu'un seul neurone se spécialise sur chaque rotation), la figure 5.5 montre que le réseau approche les rotations intermédiaires par une combinaison quasi-linéaire des spectres des rotations voisines. Si les rotations présentes dans l'ensemble d'apprentissage ne sont pas trop distantes les unes des autres, il s'agit d'une bonne approximation. En effet, considérons deux transformations A et B qui commutent et notons a et b leurs valeurs propres respectives correspondant au vecteur propre v. Notons de plus T la transformation intermédiaire entre A et B, c'est-à-dire $T = (AB)^{\frac{1}{2}}$ où la racine carrée matricielle est définie telle que $M^{\frac{1}{2}}M^{\frac{1}{2}} = M$. Lorsque M est diagonalisable, $M = VDV^{-1}$ et $M^{\frac{1}{2}} = VD^{\frac{1}{2}}V^{-1}$. Les valeurs propres de $M^{\frac{1}{2}}$ sont donc les

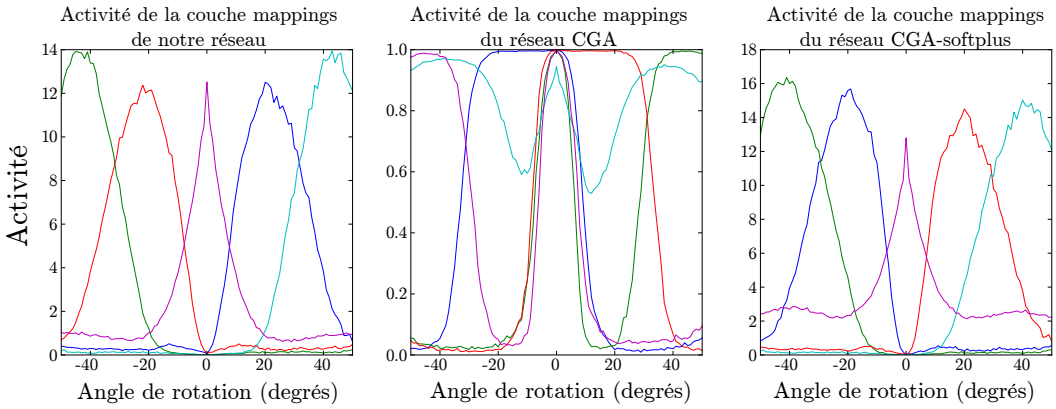


Figure 5.6 – *Activité des neurones de la couche supérieure pour des rotations comprises entre -50 et 50 degrés, pour une couche softplus de 5 neurones. Les représentations apprises par les différentes approches sont assez similaires. Comparé à la figure 5.5 ceci montre que l'apprentissage de transformations orthogonales agit comme une contrainte de régularisation forte lorsque la quantité de ressources disponibles augmente.*

racines (éventuellement complexes) de celles de M. Dans notre cas, puisque par propriété des matrices orthogonales, A, B et T partagent les mêmes vecteurs propres, les valeurs propres t de T sont des les moyennes géométriques des valeurs propres a de A et b de B : $t = \sqrt{ab}$. Finalement, approcher la moyenne géométrique par la moyenne arithmétique est une bonne approximation quand les deux nombres ne sont pas trop éloignés (ce qui est le cas des valeurs propres de deux matrices orthogonales proches, par continuité) : si $a = c + \delta$ et $b = c - \delta$, l'erreur au premier ordre est de $\frac{\delta^2}{2c}$.

La figure 5.6 illustre le comportement du réseau quand il y a moins de neurones dans la couche supérieure que de transformations dans l'ensemble d'apprentissage. Pour cela, nous avons entraîné un réseau sur le même ensemble que précédemment, mais avec seulement 5 neurones dans la couche supérieure au lieu de 40. Dans ce cas, il apparaît que les trois types de réseaux apprennent une représentation régulièrement distribuée des transformations. Comparé à la figure 5.5, ceci montre la capacité de l'architecture proposée à n'utiliser qu'un nombre réduit de neurones même quand un grand nombre sont disponibles. Les ressources non utilisées sont ainsi disponibles pour apprendre à représenter de nouvelles transformations (notamment qui ne commutent pas avec celles déjà apprises) dans le cas où celles-ci seraient rencontrées par la suite.

Troisième expérience Nous menons une troisième et dernière expérience en entraînant le réseau sur deux familles de transformations qui ne commutent pas entre elles, à savoir les translations et les rotations. Nous entraînons l'architecture proposée ainsi que les deux variantes de CGA pour 500 époques (une époque correspondant à une passe sur l'ensemble d'apprentissage) puis nous mesurons l'erreur de reconstruction sur 10 000 paires d'images couvrant l'ensemble des transformations utilisées pendant l'apprentissage. Étant donnée la proximité des deux algorithmes, nous utilisons les mêmes paramètres d'apprentissage

(en particulier le taux d'apprentissage) pour tous les algorithmes.

La figure 5.7 permet de comparer la rapidité d'apprentissage des différentes approches ainsi que l'influence du nombre de neurones de la couche de facteurs. Nous représentons en abscisse l'erreur de reconstruction obtenue par l'architecture proposée et en ordonnée l'erreur obtenue par l'algorithme CGA (traits pleins) ou CGA-softplus (traits pointillés). Chaque courbe correspond à l'évolution de l'erreur au cours de l'apprentissage, la première époque étant située en haut à droite. Cette figure montre que l'apprentissage est plus rapide avec notre algorithme qu'avec les variantes CGA (les courbes sont proches de l'horizontale au début). De plus, pour des tailles de la couche de facteurs suffisantes, l'erreur finale obtenue après 500 époques est plus faible avec l'architecture proposée. En revanche, le réseau CGA est meilleur lorsque le nombre de facteurs est réduit. Ce résultat peut sembler surprenant étant donnée notre étude qui stipule que les vecteurs propres sont partagés par les transformations qui commutent. Il faut cependant remarquer que lorsqu'il n'y a pas assez de facteurs pour représenter l'ensemble des vecteurs propres de la transformation, ceci revient à assigner une valeur propre nulle aux vecteurs propres manquants. Comme les valeurs propres des transformations orthogonales sont de module unitaire l'erreur engendrée peut donc être importante. De plus, il faut également rappeler que, pour une taille de la couche de facteurs donnée, les réseaux CGA ont deux fois plus de poids entre les entrées et les facteurs, augmentant ainsi leur flexibilité. Par ailleurs, la figure 5.7 permet de confirmer la limite théorique prédite par l'étude mathématique sur le nombre de facteurs utiles. En effet, avec deux familles de transformations, nous atteignons une limite pour $2 \times 2 \times 13^2 = 676$ facteurs (chaque vecteur propre nécessitant deux facteurs pour représenter partie réelle et partie imaginaire).

La figure 5.8 présente l'erreur de reconstruction finale pour les trois algorithmes pour différentes tailles de la couche de facteurs. L'algorithme CGA est sujet à une forte dégradation des performances lorsque le nombre de facteurs augmente au delà d'une certaine limite (passage de 700 à 800 facteurs), ce qui n'est pas le cas de l'algorithme proposé (qui ne présente qu'une légère dégradation des performances entre 700 et 800 facteurs, avec 100 neurones *softplus*), ce qui est cohérent avec une meilleure capacité de généralisation. La figure 5.8 permet également de comparer les performances de l'algorithme proposé avec les variantes CGA pour un même nombre de poids entre les entrées et les facteurs. Les variantes CGA sont alors dépassées pour tous les cas testés.

La figure 5.9 montre quant à elle la matrice U apprise par notre réseau. Comme attendu, la plupart des colonnes se présentent en paires partie réelle/partie imaginaire de vecteurs propres complexes. Ils sont similaires aux filtres appris par les CGA (voir par exemple (Memisevic 2012b)).

5.3.3 Apprentissage de séquences contextuelles

Cette section est consacrée à la présentation d'une architecture permettant d'apprendre à générer des séquences à partir d'une représentation haut niveau. Dans son développement actuel, cette architecture ne permet cependant pas d'apprendre cette représentation haut niveau de manière autonome et non supervisée. Nous montrons en revanche qu'elle permet de générer des séquences à partir d'une représentation à la fois symbolique de

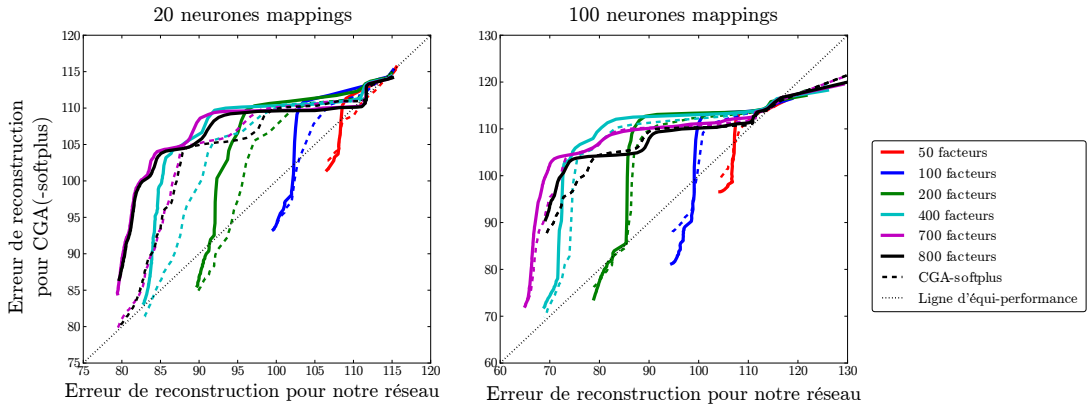


Figure 5.7 – *Courbes d'apprentissage pour notre algorithme et les variante de CGA, entraînés sur 500 époques. Nous représentons l'évolution de l'erreur de reconstruction entre notre algorithme (en abscisse) et CGA (en ordonnée, traits pleins) et CGA-softplus (en ordonnée, pointillés) pour différentes tailles de la couche de facteurs et deux tailles de la couche softplus. La première époque correspond aux points situés dans l'angle supérieur droit et l'apprentissage se traduit par un déplacement vers le coin inférieur gauche. Lors des premières époques, les courbes sont proches de l'horizontale : ceci montre que l'apprentissage est beaucoup plus rapide avec notre approche. Pour un nombre de facteurs inférieur à 400, les courbes se terminent cependant en-dessous de la diagonale, ce qui traduit une meilleure performance finale des approches CGA et CGA-softplus. Cependant, pour un nombre de facteurs plus important, notre approche devient meilleure. En particulier avec 100 neurones mappings, la comparaison des courbes pour 700 et 800 facteurs montre un très grande dégradation de la performance pour les approches CGA et CGA-softplus.*

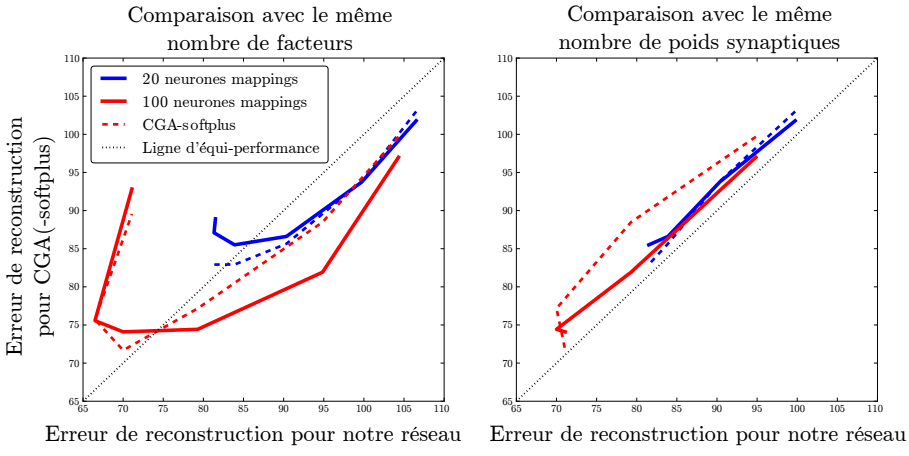


Figure 5.8 – *Comparaison de l'erreur de reconstruction obtenue avec notre algorithme et les variantes CGA pour 50, 100, 200, 400, 700 et 800 facteurs. À gauche : comparaison avec le même nombre de facteurs ; à droite : comparaison avec le même nombre de paramètres. La meilleure performance des approches CGA et CGA-softplus observée sur la figure 5.7 pour un nombre égal de facteurs disparaît lorsque l'on compare à nombre égal de paramètres.*

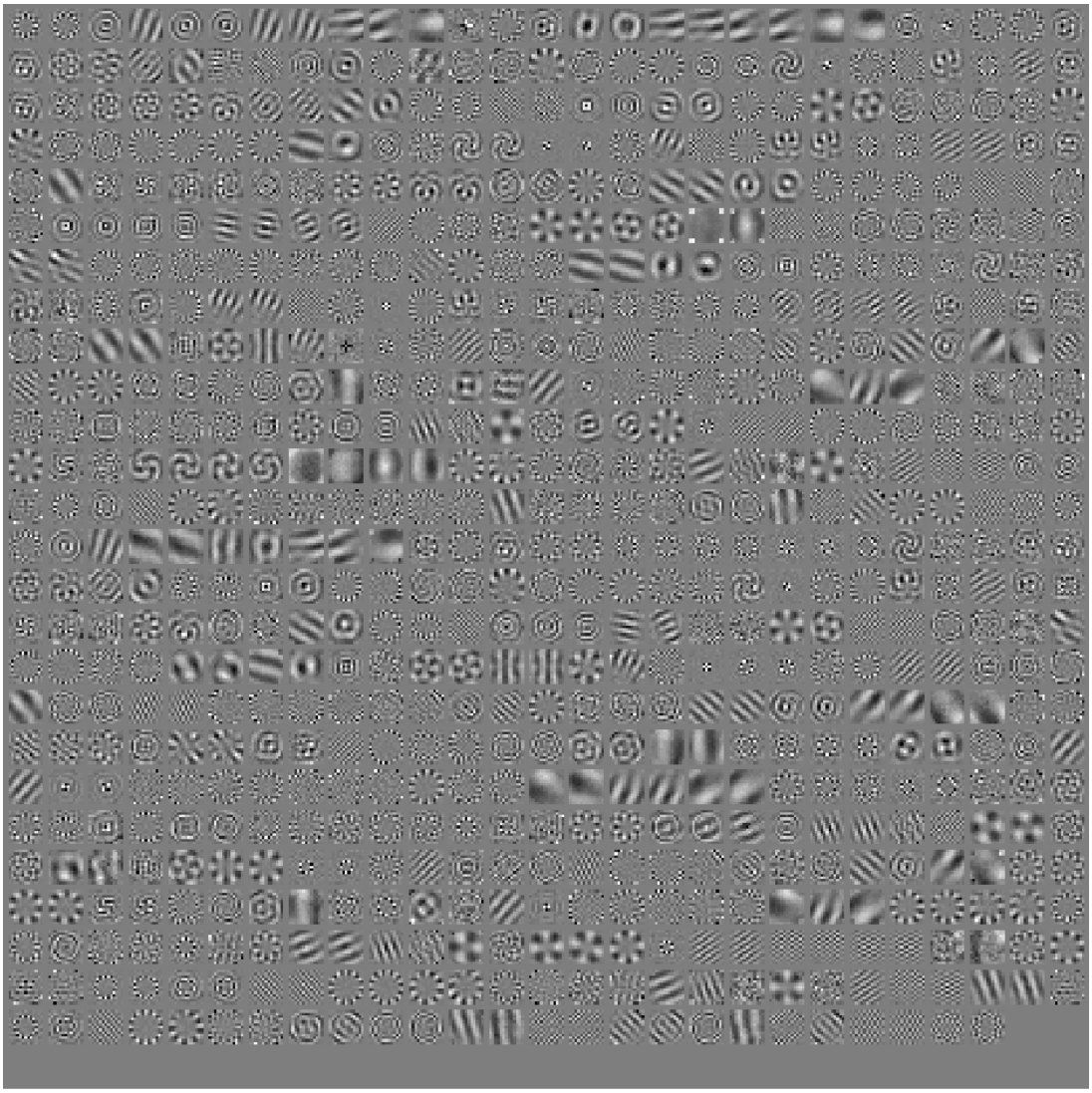


Figure 5.9 – *Matrice U apprise par le réseau sur un ensemble de rotations et translations, avec 700 facteurs et 100 neurones softplus. Chaque imagette correspond à une colonne de la matrice U redimensionnée en carré de 13×13 pixels. Comme attendu d'après notre approche mathématique, les colonnes de la matrice U se présentent à quelques exceptions près en paires partie réelle / partie imaginaire de vecteurs propres complexes distincts pour les rotations et pour les translations.*

différentes catégories de séquences, tout en permettant de paramétrer chaque séquence au sein de sa catégorie.

Dans toute la suite, nous appliquons cette architecture au problème de la génération de trajectoires cartésiennes, ce qui permet d'illustrer aisément les capacités de l'architecture.

Les travaux présentés dans cette section ont été publiés dans

> Alain Droniou, Serena Ivaldi et Olivier Sigaud (2014). « Learning a Repertoire of Actions with Deep Neural Networks ». Dans : *Proceedings of ICDL-EpiRob*. Italie.

Apprentissage et représentation de séquences

La génération d'actions est un domaine d'application évident de la génération de séquences. Une des techniques les plus populaires est l'utilisation de *Dynamic Motion Primitives* (DMP) (Ijspeert et al. 2013) (voir par exemple (Pastor et al. 2009 ; Neumann et al. 2009 ; Muelling et al. 2010 ; Daniel et al. 2012b,a)) qui combinent un oscillateur amorti qui assure la convergence et la stabilité du système avec un terme appris qui permet de déformer la trajectoire canonique pour générer théoriquement n'importe quelle trajectoire. Ce terme appris est généralement paramétré par une variable temporelle (ou une variable de phase), mais d'autres variables peuvent être ajoutées afin d'avoir une plus grande flexibilité (Pastor et al. 2013 ; Stulp et al. 2013). Si la trajectoire canonique peut théoriquement être déformée pour obtenir n'importe quelle trajectoire, cela nécessite en pratique une quantité importante de connaissances *a priori* afin de définir des primitives adéquates avec un nombre réduit de paramètres à apprendre afin d'éviter la malédiction de la dimensionalité (les DMPs peuvent être vus comme des implémentations directes de l'équation (5.1)).

Comme nous l'avons vu à travers les exemples de la section 5.2, les réseaux de neurones récurrents ont également été utilisés pour reproduire des séquences temporelles. Ces approches utilisent généralement des représentations implicites du temps (équation (5.2)) reposant sur les dynamiques propres des réseaux récurrents.

Nous travaillons dans cette section sur des modèles généraux de la forme

$$\mathbf{q_t} = f(\mathbf{s_t}, \mathbf{m_t}, \mathbf{c_t}) \tag{5.26}$$

où $\mathbf{q_t}$ correspond à la séquence générée (des commandes motrices dans le cas des actions), $\mathbf{s_t}$ à l'état du système à l'instant t (par exemple la position d'un bras robotique), $\mathbf{m_t}$ à une mémoire des états passés (qui sera l'objet de discussions à la section 5.4) et $\mathbf{c_t}$ à une description haut niveau de la séquence (par exemple le type d'action en train d'être exécutée). Notons que l'état $\mathbf{s_t}$ est défini de manière très générale et peut par exemple intégrer une représentation de $\mathbf{q_{t-1}}$.

Nous choisissons de factoriser l'équation (5.26) en deux termes, de manière à introduire une représentation (apprise) de buts intermédiaires :

$$\mathbf{q_t} = f(\mathbf{s_t}, g(\mathbf{m_t}, \mathbf{c_t})). \tag{5.27}$$

Le premier terme de cette factorisation, $g(\mathbf{m_t}, \mathbf{c_t})$ permet en effet de générer une représentation du prochain état désiré, à partir de la dynamique de la séquence passée $\mathbf{m_t}$ et de

la représentation haut niveau de la séquence $\mathbf{c_t}$. Cette représentation peut alors être utilisée par la deuxième fonction f en combinaison avec l'état courant pour générer l'élément de la séquence $\mathbf{q_t}$. Cette factorisation ne diminue pas l'expressivité de l'équation (5.26) (il suffit que g corresponde à la concaténation de $\mathbf{m_t}$ et $\mathbf{c_t}$), mais peut permettre d'apprendre des synergies entre différentes actions : pour deux couples $(\mathbf{m_t}, \mathbf{c_t})$ différents, il est possible de générer la même valeur de g, qui peut être réutilisée par f de manière transparente.

Comme nous l'avons déjà indiqué, nous allons nous focaliser sur l'exemple des actions. Nous allons prendre l'exemple d'une tâche d'écriture pour laquelle un robot doit apprendre à écrire les chiffres de 0 à 9. Dans ce cas, la commande $\mathbf{q_t}$ peut être la vitesse cartésienne du bras du robot, $\mathbf{s_t}$ la position cartésienne de l'extrémité du bras, $\mathbf{m_t}$ la trace des dernière positions cartésiennes et $\mathbf{c_t}$ une représentation haut niveau de l'action. Dans la suite, pour la clarté de l'exposé, on notera $\dot{\mathbf{x}}_\mathbf{t}$ la vitesse cartésienne du bras, $\mathbf{x_t}$ sa position et $\mathbf{a}$ l'action en train d'être effectuée (qui sera dans notre cas du type "écrire un 1", "écrire un 2", ou bien encore dans un second temps "écrire un 3 penché de 45° vers la droite"). Nous conservons la notation $\mathbf{m_t}$ pour la trace des anciennes positions. L'équation (5.27) devient donc :

$$\dot{\mathbf{x}}_\mathbf{t} = f(\mathbf{x_t}, g(\mathbf{m_t}, \mathbf{a})). \tag{5.28}$$

Architecture

L'implémentation de l'équation (5.28) par un réseau de neurones est illustrée figure 5.10. Ce réseau se décompose en trois sous-parties.

Deux autoencodeurs classiques dont le but est de "normaliser" les données en entrée de façon à pouvoir travailler avec n'importe quelles données sur n'importe quelles plages de valeurs. Pour cela, ils utilisent deux matrices distinctes pour l'encodage et le décodage. Grâce à l'utilisation d'une fonction sigmoïde pour leur couche cachée et linéaire pour leur couche visible, ils permettent au reste du réseau de travailler sur des entrées toutes comprises entre 0 et 1, sans pour autant limiter les capacités génératives à de telles valeurs (ils remplacent notamment les étapes de centrage et normalisation de la variance de l'entrée souvent utilisées dans d'autres travaux). On note leurs sorties respectives

$$\xi_\mathbf{t} = \sigma(W^{in}\mathbf{x_t}) \tag{5.29}$$

$$\widetilde{\xi}_\mathbf{t} = \sigma(W^{out}\dot{\mathbf{x}}_\mathbf{t}). \tag{5.30}$$

Une couche supérieure dont le but est de calculer la représentation intermédiaire g qui prend $\mathbf{m_t}$ et $\mathbf{a}$ en entrée :

$$\mathbf{g}(\mathbf{m_t}, \mathbf{a}) = \sigma\left(W_{g_1}\left((W_m\mathbf{m_t}) * (W_a\mathbf{a})\right)\right). \tag{5.31}$$

Une couche intermédiaire qui calcule f étant donnés $\mathbf{x_t}$ et la sortie $g(\mathbf{m_t}, \mathbf{a})$ de la couche précédente.

$$\widetilde{\xi}_\mathbf{t} = \sigma\left(W_{\widetilde{\xi}}\left((W_\xi\xi_\mathbf{t}) * (W_{g_2}\mathbf{g}(\mathbf{m_t}, \mathbf{a}))\right)\right). \tag{5.32}$$

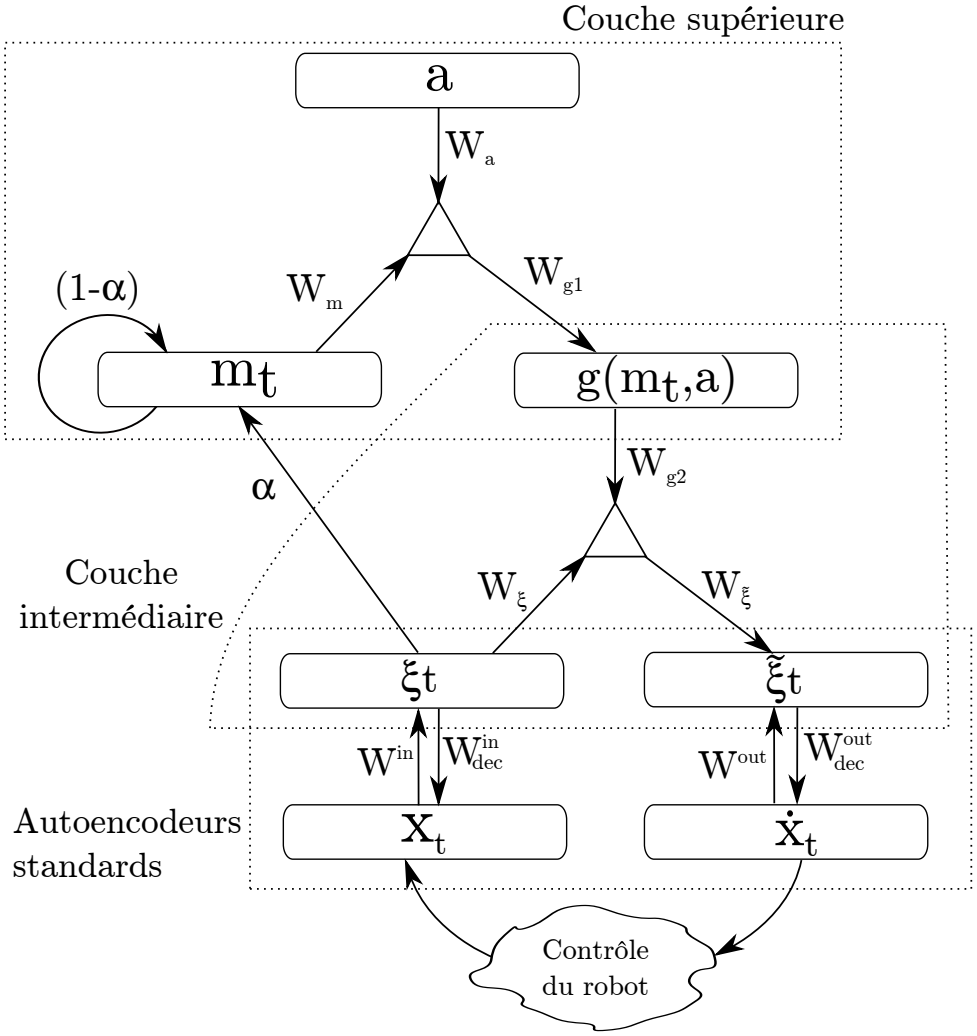


Figure 5.10 – *Architecture du réseau utilisé pour la génération de séquences motrices. Les positions cartésiennes* $\mathbf{x_t}$ *sont tout d'abord encodées par un autoencodeur classique qui apprend une représentation* ξ*. Cette représentation est utilisée pour calculer une trace* $\mathbf{m_t}$ *des dernières positions ainsi que la vitesse désirée* $\dot{\mathbf{x}}_\mathbf{t}$ *(par l'intermédiaire de sa représentation* $\tilde{\xi}$*). La couche supérieure du réseau utilise la trace* $\mathbf{m_t}$ *et la représentation haut niveau de l'action* $\mathbf{a}$ *pour produire une représentation intermédiaire* $\mathbf{g}(\mathbf{m_t}, \mathbf{a})$*. Sur le schéma, les flèches indiquent la direction des projections encodées par les matrices correspondantes (les matrices transposées sont utilisées pour les projections dans le sens inverse, sauf au niveau des deux autoencodeurs d'entrée/sortie). Le choix des orientations correspond au sens du flux de calculs dans le réseau lorsqu'il est utilisé pour générer des vitesses désirées* $\dot{\mathbf{x}}_\mathbf{t}$ *étant donnés une position* $\mathbf{x_t}$*, une action* $\mathbf{a}$ *et son progrès courant* $\mathbf{m_t}$*.*

Dans ce travail, nous utilisons une formule très simple pour le calcul de $\mathbf{m_t}$ qui consiste en une moyenne sur une fenêtre exponentielle

$$\mathbf{m_t} = (1 - \alpha)\mathbf{m_{t-1}} + \alpha\xi_\mathbf{t}. \tag{5.33}$$

Il s'agit d'une des limitations les plus importantes sur laquelle nous reviendrons.

Le réseau est entraîné par apprentissage de toutes les matrices de connexions W_* dans les équations (5.29), (5.30), (5.31) et (5.32). Pour cela, on minimise l'erreur de reconstruction des vitesses cartésiennes $\widehat{\dot{\mathbf{x}}}$ prédites par le réseau étant donnés les positions $\mathbf{x_t}$ et un vecteur $\mathbf{a}$, en utilisant une descente de gradient classique sur l'erreur $\sum_t ||\widehat{\dot{\mathbf{x}}_\mathbf{t}} - \dot{\mathbf{x}}_\mathbf{t}||^2$.

En suivant le paradigme de l'apprentissage profond, chaque couche est pré-entraînée de manière indépendante. Premièrement, les deux autoencodeurs apprennent une représentation ξ de $\mathbf{x}$ et une représentation $\widetilde{\xi}$ de $\dot{\mathbf{x}}$ en minimisant l'erreur de reconstruction de $\mathbf{x}$ et $\dot{\mathbf{x}}$ respectivement. Ensuite, la couche intermédiaire est entraînée pour apprendre une représentation $\mathbf{g_t}$, étant donnés $\xi_\mathbf{t}$ et $\widetilde{\xi}_\mathbf{t}$, en minimisant la distance avec leurs reconstructions $\xi_\mathbf{t}^\mathbf{recons}$ et $\overset{\sim}{\xi}_\mathbf{t}^\mathbf{recons}$:

$$\mathbf{g_t} = \sigma(W_{g_2}^\top(W_\xi\xi_\mathbf{t} * W_{\widetilde{\xi}}^\top\widetilde{\xi}_\mathbf{t})) \tag{5.34}$$

$$\xi_\mathbf{t}^\mathbf{recons} = \sigma(W_\xi^\top(W_{g_2}\mathbf{g_t} * W_{\widetilde{\xi}}^\top\widetilde{\xi}_\mathbf{t})) \tag{5.35}$$

$$\overset{\sim}{\xi}_\mathbf{t}^\mathbf{recons} = \sigma(W_{\widetilde{\xi}}(W_\xi\xi_\mathbf{t} * W_{g_2}\mathbf{g_t})). \tag{5.36}$$

La couche supérieure est alors entraînée à inférer la représentation intermédiaire $\mathbf{g_t}$ calculée par la couche précédente, étant donnés $\mathbf{m_t}$ et $\mathbf{a}$, en minimisant la différence entre $\mathbf{g_t}$ et $\mathbf{g}(\mathbf{m_t}, \mathbf{a})$ (équation (5.31)). Enfin, une descente de gradient globale est effectuée sur l'ensemble du réseau afin de minimiser l'erreur de prédiction de $\dot{\mathbf{x}}_\mathbf{t}$.

Expériences

Nous avons testé cette architecture sur une tâche avec le robot humanoïde iCub consistant à écrire les dix chiffres de 0 à 9. Nous avons pour cela utilisé les trajectoires cartésiennes enregistrées lors de l'expérience sur l'apprentissage de représentations multimodales présentée au chapitre 4 (76 trajectoires ont été enregistrées pour chaque chiffre). L'origine du repère cartésien est définie par le point de départ de chaque trajectoire, et les trajectoires ont été échantillonnées à une fréquence de 100Hz, ce qui donne entre 100 points pour les chiffres "courts" comme le 1 et jusqu'à 500 points pour les chiffres plus longs, comme le 8. La figure 5.11 illustre quelques-unes des trajectoires enregistrées.

Le nombre de neurones utilisés pour chaque couche est donné dans le tableau 5.2. La constante de temps α de la fenêtre exponentielle est choisie à 0.02, nous utilisons un taux d'apprentissage de 0.001 et un momentum de 0.95. La couche $\mathbf{m_t}$ est remise à zéro au début de chaque trajectoire.

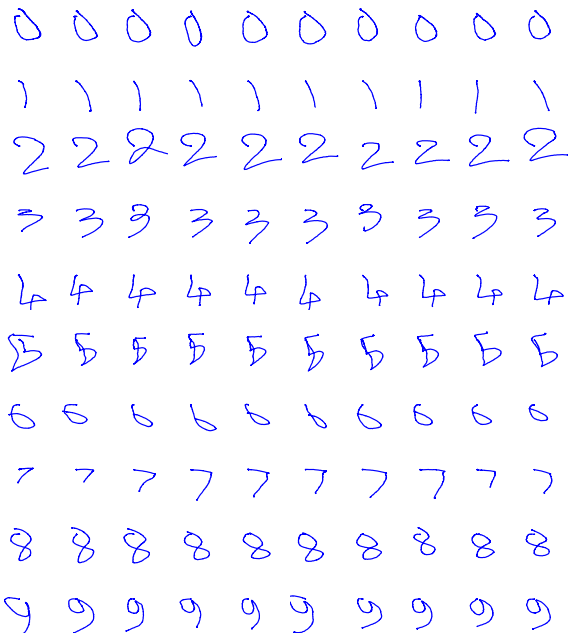


Figure 5.11 – *Quelques exemples des trajectoires enregistrées sur le robot iCub et utilisées pour l'entraînement du réseau présenté. Chaque trajectoire est dessinée dans un rectangle de 12x8 cm.*

Table 5.2 – *Nombre de neurones utilisés pour les expériences*

Couche	Nombre de neurones
$\mathbf{a}$	10
$\mathbf{m_t}$	100
facteurs ($\mathbf{m_t}$,$\mathbf{a}$)	1000
$\mathbf{g}(\mathbf{m_t}, \mathbf{a})$	100
$\xi_\mathbf{t}$	100
facteurs ($\mathbf{g}$,$\xi_\mathbf{t}$)	100
$\tilde{\xi}_\mathbf{t}$	100

Nous allons présenter trois expériences. La première est dédiée à la validation de l'approche, en montrant que le réseau proposé peut apprendre à générer différentes séquences à partir d'une représentation symbolique des actions. Dans un second temps, nous étendons cette représentation en ajoutant des variables continues permettant de paramétrer une variété pour chacune des actions possibles. Cette représentation étant très proche de celle obtenue au chapitre 4 à partir d'entrées multimodales, nous montrons dans une dernière expérience qu'il est effectivement possible d'utiliser ces représentations pour apprendre à générer les séquences correspondantes.

Première expérience Dans cette première expérience, nous définissons dix actions distinctes correspondant au tracé de chacun des chiffres entre 0 et 9. Le vecteur **a** est donc constitué de 10 neurones parmi lesquels un seul est actif à la fois (sur la base d'un étiquetage supervisé de chacune des trajectoires). Le réseau est entraîné sur des sous-ensembles (*mini-batches*) de 1000 couples position/vitesse obtenus en concaténant des trajectoires choisies aléatoirement jusqu'à obtenir 1000 points. Après apprentissage, le réseau est utilisé pour générer des trajectoires. Pour cela, nous simulons la boucle de contrôle du robot grâce à l'équation

$$\mathbf{x_{t+1}} = \mathbf{x_t} + 0.001 \times \eta \times (\widehat{\dot{\mathbf{x}}_\mathbf{t}} + \nu) \tag{5.37}$$

où η est un bruit de Poisson (de paramètre $\lambda = 10$) qui simule un délai variable de la boucle de contrôle et ν est un bruit gaussien centré (écart-type de $0.0025m/s$) simulant une commande imprécise du robot. Ces paramètres simulent donc une boucle de contrôle avec un délai moyen de $0.01s$ et d'écart-type d'environ $0.003s$ (sur une plage de valeur allant de 0 à environ 0.025s) et une erreur de commande d'environ 5% en moyenne. Le processus de génération de chaque trajectoire démarre au point de coordonnées $(0, 0, 0)$. Il est itéré pour un nombre de pas de temps égal à la longueur moyenne des trajectoires enregistrées pour chacun des dix chiffres. La figure 5.12 illustre les trajectoires générées.

Deuxième expérience Nous avons testé dans l'expérience précédente le comportement du réseau appris à partir d'une représentation symbolique des actions. Dans cette nouvelle expérience, nous montrons que le réseau est capable d'apprendre à générer des séquences à partir d'une représentation de sous-variétés similaire à celle apprise par l'architecture présentée au chapitre 4.

Pour cela, nous générons un nouvel ensemble d'apprentissage à partir de dix trajectoires extraites de l'ensemble utilisé pour l'expérience précédente, auxquelles nous appliquons un ensemble de rotations d'angles $\Theta \in \{-\pi/2, -\pi/4, 0, \pi/4, \pi/2\}$ (en multipliant les trajectoires par les matrices de rotation correspondantes). Nous obtenons donc un ensemble de 50 trajectoires (une trajectoire par chiffre et par rotation). Nous ajoutons ce paramètre de rotation à la représentation haut niveau de l'action en concaténant deux nouveaux neurones prenant les valeurs $\frac{1}{2}(1 + cos(\Theta))$ et $\frac{1}{2}(1 + sin(\Theta))$. Le vecteur **a** est donc composé de 10 neurones binaires représentant chacun des dix chiffres et de deux neurones à valeurs réelles représentant l'angle de la rotation. Tous les autres paramètres sont identiques à

Figure 5.12 – *Trajectoires générées par le réseau avec une boucle de contrôle simulée bruitée (voir le texte pour les détails). Chaque trajectoire est représentée dans un rectangle de 12x8 cm.*

ceux de l'expérience précédente. Cette expérience correspond donc à l'apprentissage de séquences réparties le long de sous-variétés unidimensionnelles.

Après apprentissage, le réseau est utilisé pour générer des trajectoires avec des angles de rotation $\Theta = k\frac{\pi}{8}$ pour $k \in \{-4, \cdots, 4\}$. La boucle de contrôle est cette fois simulée sans ajout de bruit. La figure 5.13 montre les trajectoires générées.

Troisième expérience Comme nous l'avons déjà souligné, l'expérience précédente utilise une représentation très proche de celle générée par l'architecture présentée au chapitre 4. Il est donc naturel de tester le réseau en utilisant les représentations apprises par cet autre réseau. Pour cela, nous entraînons d'abord le réseau du chapitre précédent sur des entrées bimodales images et spectrogrammes, en utilisant les données décrites au chapitre précédent. Nous enregistrons alors les représentations obtenues (10 neurones *softmax* et 2 neurones *softplus*), que nous utilisons pour entraîner le réseau présenté dans cette section, en les associant aux trajectoires correspondantes. Nous utilisons les mêmes paramètres que dans l'expérience précédente.

Nous entraînons le réseau sur 70 des 76 exemples enregistrés pour chaque chiffre. Nous utilisons alors les 6 exemples restants pour tester le réseau. Grâce à l'architecture décrite au chapitre précédent, nous pouvons générer la représentation au choix à partir de l'image seule, du spectrogramme seul ou des deux en même temps. La figure 5.14 illustre les trajectoires obtenues dans chacun des cas. Certains chiffres sont très mal reproduits, notamment les chiffres 4, 5 et 9. Il faut cependant rappeler que, dans cette expérience,

Figure 5.13 – *Trajectoires générées par le réseau pour différents angles de rotation. Les trajectoires sur fond gris correspondent à des angles présents dans l'ensemble d'apprentissage, tandis que les trajectoires sur fond blanc correspondent à des généralisations apprises par le réseau.*

la difficulté d'apprentissage est augmentée par le fait que s'accumulent les erreurs de catégorisation du réseau bimodal image/spectrogramme et les erreurs du réseau apprenant les trajectoires. En particulier, toute erreur de classification d'un chiffre vient grandement perturber l'apprentissage des trajectoires, puisque le réseau doit alors apprendre à générer deux (ou plus) types de trajectoires complètement différentes en se basant uniquement sur les valeurs des neurones *softplus*. Ceci amène naturellement un sur-apprentissage et une mauvaise généralisation pour les chiffres les plus concernés. En outre, on observe sur la figure que, selon les entrées fournies au réseau, la qualité des trajectoires varie grandement. Les meilleures trajectoires semblent obtenues à partir des spectrogrammes seuls, alors que les plus mauvaises semblent obtenues avec les images seules. Ce dernier résultat peut être surprenant, puisqu'il s'agit *a priori* de l'entrée la mieux corrélée aux trajectoires. Mais il s'agit aussi par la même de l'entrée dont les corrélations ont été le plus brouillées par les chiffres mal catégorisés.

Cette expérience montre les résultats prometteurs que l'on peut attendre de la combinaisons d'architectures profondes pour l'apprentissage non supervisé, mais montre également qu'il reste une très grande marge de progrès à atteindre avant de pouvoir être réellement utilisable.

5.4 Enjeux et perspectives

Comme nous l'avions annoncé en tête de ce chapitre, nous n'avons pas de solution satisfaisante au problème de la temporalité. Nous avons présenté deux travaux distincts, l'un permettant d'apprendre à représenter des transformations orthogonales, l'autre permettant d'apprendre à reproduire des séquences à partir d'une représentation haut ni-

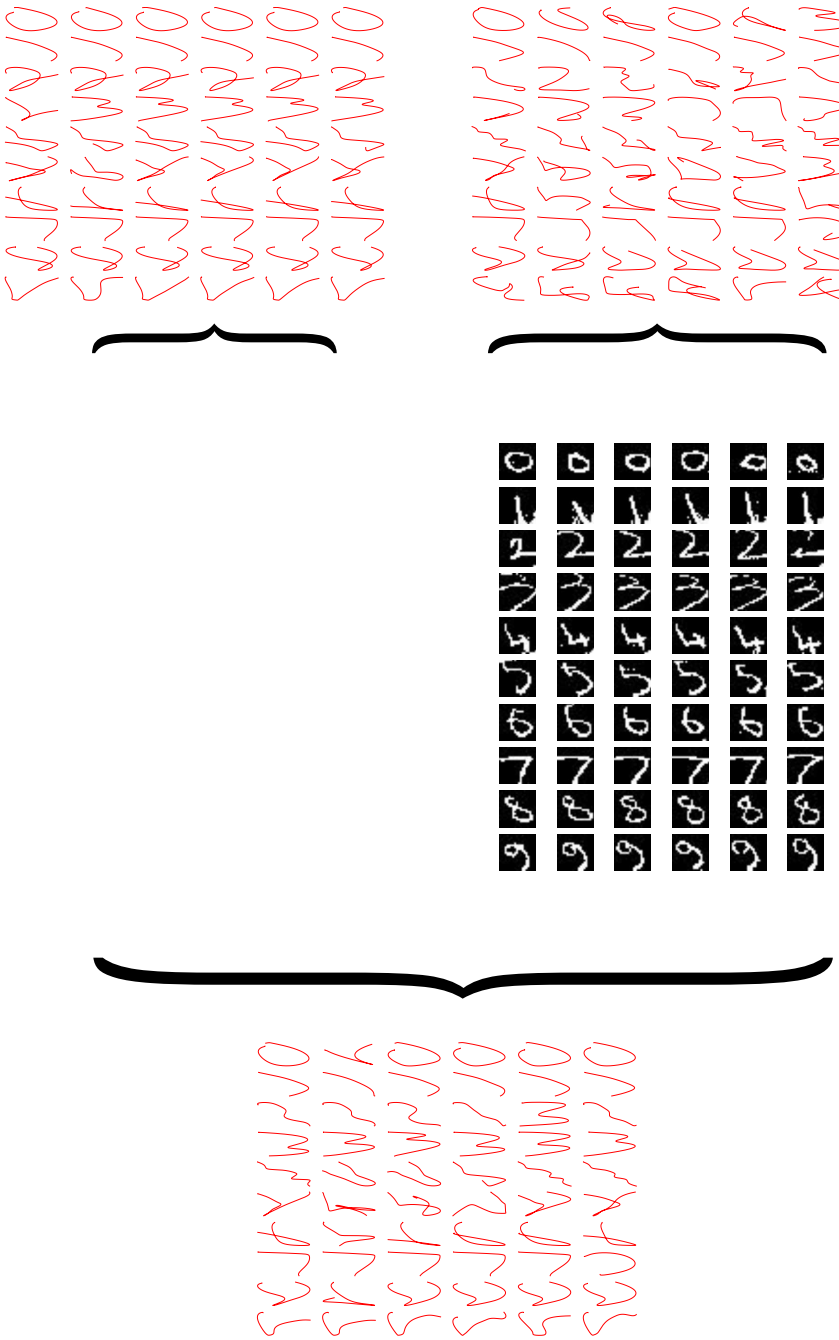


Figure 5.14 – *Trajectoires générées par le réseau à partir des représentations apprises par le réseau décrit au chapitre 4. Le réseau a été entraîné sur 70 des 76 exemples enregistrés pour chaque chiffre. Nous donnons ici les trajectoires générées à partir des 6 exemples restants. Nous avons entraîné le réseau du chapitre 4 sur les images des chiffres et les spectrogrammes correspondants à leur prononciation (voir le chapitre 4 pour les détails sur l'acquisition des données). Les représentations retenues pour l'apprentissage des trajectoires correspondent aux représentations conjointes image/spectrogramme. Pour la phase de test, nous avons utilisé les trois variations possibles : à partir des images seules, des spectrogrammes seuls, ou de la combinaison des deux.*

veau, pouvant notamment avoir la même forme que les représentations produites par l'architecture présentée au chapitre 3.

Il manque encore un mécanisme crucial qui apprend une représentation haut niveau de séquences temporelles respectant l'hypothèse des sous-variétés à partir d'un flux non segmenté. Comme nous l'avons déjà expliqué, nous pensons que l'utilisation de transformations orthogonales peut être un élément intéressant permettant d'identifier les instants auxquels de nouvelles informations ont été ajoutées dans le flux, indiquant le passage d'une séquence à une autre. L'émergence d'une représentation haut niveau à partir de ces informations reste cependant une question ouverte. L'implémentation de la couche mémoire $\mathbf{m_t}$ de la seconde architecture que nous avons utilisée est excessivement simple. Dans un cadre plus général, il serait souhaitable que le réseau puisse apprendre par lui-même quels éléments mémoriser. Ceux-ci dépendent notamment de la séquence considérée (par exemple, dans le cas du tracé du chiffre 0, le point de départ est une information importante à mémoriser afin de générer une fin de trajectoire convenable pour fermer la boucle). Des travaux plus approfondis sont nécessaires pour étudier cette question. Une approche possible est l'utilisation d'une couche à base de LSTM entraînée en même temps que le reste du réseau. De plus, il est fortement probable que ce mécanisme de mémorisation soit un élément essentiel d'une architecture capable d'apprendre une représentation haut niveau des séquences : cette représentation haut niveau doit influencer la façon de mémoriser les éléments importants de la séquence en cours, tandis que les éléments mémorisés influencent directement les prédictions des instants suivants et sont donc à l'origine de l'erreur de prédiction pouvant servir à segmenter les différentes séquences.

Il faut souligner que l'introduction d'*a priori* sur les transformations orthogonales dans notre architecture nous a permis d'utiliser une seule et même matrice (notée U dans l'exposé de l'architecture) pour toutes les entrées du réseau, à la différence du réseau *gated* classique, ce qui est compatible avec le principe d'avoir un seul chemin de traitement pour toutes les entrées et non plusieurs chemins différents distinguant différents pas de temps, ce qui réduit d'autant les ressources calculatoires nécessaires.

Chapitre 6

Pistes de recherche

Mieux vaut regarder là où on ne va pas, parce que, là où on va, on saura ce qu'il y a quand on y sera ; et, de toute façon, ce sera jamais que de l'eau.

Jacques Rouxel - Les Shadoks

Sommaire

Les travaux que nous avons présentés dans cette thèse se sont principalement intéressés à la construction non supervisée et à l'utilisation de représentations haut niveau des flux de capteurs d'un robot, à l'aide de réseaux de neurones profonds. De telles capacités ne sont cependant qu'une brique élémentaire d'une architecture plus globale si l'on veut réellement doter les robots d'une certaine autonomie. Nous discutons dans ce chapitre l'intérêt de l'apprentissage profond pour deux problématiques liées : l'apprentissage par renforcement et la curiosité artificielle. Si la première partie est principalement dédiée à la présentation de différentes approches de la littérature pour l'apprentissage par renforcement à l'aide de réseaux profonds et de techniques associées, nous présentons dans la seconde partie des travaux originaux dédiés à la curiosité artificielle dans les réseaux de neurones non-supervisés.

6.1 Apprentissage profond et renforcement

L'apprentissage par renforcement est un domaine de l'intelligence artificielle dont le but est d'apprendre à suivre une politique optimale à partir d'un signal de supervision très faible, appelé récompense, qui n'indique que pour certains états ou actions à quel point ils sont *bons* ou *mauvais* en leur associant une valeur réelle plus ou moins importante.

Pour ce faire, une des approches les plus populaires consiste à apprendre une fonction de valeur (*Q-fonction*), qui associe une valeur $Q(s, a)$ (*Q-valeur*) à chaque paire état-action. Cette valeur traduit la récompense cumulée future attendue si l'action a est réalisée dans l'état s. De nombreuses variantes se différencient par la manière de calculer cette récompense cumulée attendue (Sutton et Barto 1998).

Un des obstacles majeur à l'application de cette approche à la robotique est de développer une méthode qui fonctionne dans des espaces état-action de grande dimensionalité, avec des entrées éventuellement continues et capable de généraliser correctement à de nouvelles situations.

Martin Riedmiller a proposé une approche consistant à apprendre une représentation des états à l'aide d'un réseau profond, de manière à en réduire la dimensionalité. La représentation apprise peut alors être utilisée par un algorithme standard d'apprentissage par renforcement (avec états continus) (Lange et Riedmiller 2010 ; Lange et al. 2012). Ces approches supposent néanmoins que les états peuvent être encodés par un très faible nombre de neurones, de façon à pouvoir y appliquer les algorithmes standards d'apprentissage par renforcement. Une variante consiste à entraîner un réseau à prédire directement la Q-valeur de différentes actions étant donné un état en entrée (Dutech 2012 ; Mnih et al. 2013). Cette approche ne nécessite plus de devoir encoder chaque état sur un faible nombre de neurones. De plus, pour (Mnih et al. 2013), en entraînant directement le réseau global à prédire les Q-valeurs, ce dernier peut être amené à mieux encoder les caractéristiques des états qui sont pertinentes pour la tâche (permettant donc de mieux prédire les Q-valeurs). Cependant, les actions possibles doivent être discrètes (en sortie, chaque neurone code la Q-valeur d'une seule action).

Une autre approche nous semble plus prometteuse. Dans (Sallans et Hinton 2000), les auteurs proposent une méthode utilisant des machines de Boltzmann restreintes pour approcher la Q-fonction. L'idée générale consiste à identifier l'énergie libre $\mathcal{F}$ d'un vecteur visible (cf. section 3.2.4) avec sa Q-valeur. Dans cette optique, le vecteur d'état $\mathbf{s}$ et le vecteur d'action $\mathbf{a}$ sont donc concaténés en un vecteur unique $\mathbf{v}_{(\mathbf{s},\mathbf{a})}$ utilisé en tant que vecteur d'entrée de la RBM, et on pose :

$$Q(\mathbf{s}, \mathbf{a}) = -\mathcal{F}(\mathbf{v}_{(\mathbf{s},\mathbf{a})}). \tag{6.1}$$

Les RBMs possèdent plusieurs propriétés intéressantes. Premièrement, il est possible de fixer la valeur d'un sous-ensemble du vecteur visible et de tirer aléatoirement la valeur du sous-ensemble complémentaire selon l'énergie libre du vecteur entier, en utilisant un échantillonnage de Gibbs (voir chapitre 3). Ainsi, on peut choisir les actions aléatoirement selon leurs Q-valeurs étant donné un état $\mathbf{s}$, sans avoir ni le besoin de stocker un grand tableau de valeurs (ce qui serait impossible en grande dimension ou avec des entrées continues), ni de s'appuyer sur des algorithmes d'optimisation complexes pour trouver le minimum d'une fonction continue. De plus, il est également possible de fixer une partie des valeurs du vecteur action et de tirer aléatoirement les valeurs restantes. En utilisant une représentation des actions sous la forme *symbole* × *paramètres* que nous avons utilisée dans les chapitres précédents, on peut imaginer fixer le type d'action désirée (par exemple "saisir" ou "pousser"), et générer les paramètres optimaux correspondants en fonction de

l'état. Deuxièmement, une solution au dilemme exploitation/exploration est donnée en introduisant un paramètre de température β au sein de la fonction d'échantillonnage [1] : avec une température élevée, les actions sont tirées aléatoirement de manière quasi-uniforme, alors qu'une faible température force le réseau à générer uniquement les actions dont les Q-valeurs sont les plus grandes. Enfin, les codages utilisés pour les états et les actions sont laissés libres.

L'entraînement de la RBM consiste donc à apprendre une énergie libre égale à la Q-valeur des couples état/action rencontrés. Cette Q-valeur peut être calculée par n'importe quel algorithme standard de la littérature de l'apprentissage par renforcement. Les auteurs de (Sallans et Hinton 2000) utilisent l'algorithme Sarsa, qui met à jour de manière incrémentale les Q-valeurs en fonction des récompenses obtenues. La règle de mise à jour est donnée par :

$$Q(\mathbf{s_t}, \mathbf{a_t}) \leftarrow Q(\mathbf{s_t}, \mathbf{a_t}) + \alpha(r_t + \gamma Q(\mathbf{s_{t+1}}, \mathbf{a_{t+1}}) - Q(\mathbf{s_t}, \mathbf{a_t})) \tag{6.2}$$

ce qui se traduit par la règle suivante d'apprentissage de la RBM :

$$\Delta W = \lambda(r_t + \gamma Q(\mathbf{s_{t+1}}, \mathbf{a_{t+1}}) - Q(\mathbf{s_t}, \mathbf{a_t}))\frac{\partial Q(\mathbf{s_t}, \mathbf{a_t})}{\partial W} \tag{6.3}$$

où nous avons fusionné le taux d'apprentissage α de l'algorithme Sarsa avec le taux d'apprentissage de la RBM en un unique paramètre λ, avec r_t la récompense à l'instant t et γ le coefficient d'actualisation. Dans le cas des RBMs binaires,

$$\frac{\partial Q(\mathbf{s_t}, \mathbf{a_t})}{\partial W} = \widehat{\mathbf{h}}\mathbf{v}_{(\mathbf{s_t}, \mathbf{a_t})}{}^{\top} \tag{6.4}$$

où $\widehat{\mathbf{h}}$ est le vecteur dont la i$^{\text{ème}}$ composante contient la probabilité $P(h_i = 1|\mathbf{v}_{(\mathbf{s_t}, \mathbf{a_t})})$. Il faut noter que, dans l'équation (6.3), les valeurs $Q(\mathbf{s_t}, \mathbf{a_t})$ et $Q(\mathbf{s_{t+1}}, \mathbf{a_{t+1}})$ sont obtenues en calculant l'énergie libre des vecteurs correspondants et ne nécessitent donc pas de mécanisme supplémentaire. Comparée aux autres approches présentées au début de cette section, cet algorithme a l'avantage de ne pas contraindre les actions à être discrètes, ni les états à être correctement représentés par un faible nombre de neurones.

6.1.1 Pistes de recherche

Nous proposons deux pistes de recherches à partir de ce modèle.

Optimiser une fonction de coût

L'approche présentée ci-dessus peut être généralisée de manière assez simple à l'optimisation d'une fonction de coût quelconque. En effet, en identifiant l'énergie libre d'une configuration avec le coût correspondant, il est possible d'échantillonner les états selon l'estimation de la fonction de coût. De plus, en variant la température du système,

1. Par exemple, la fonction sigmoïde s'écrit $\frac{1}{1+\exp(-\beta x)}$.

l'algorithme obtenu ressemble très fortement à l'algorithme du recuit simulé. Une telle approche possède cependant quelques défauts, comme le grand nombre d'évaluations nécessaires pour optimiser une fonction, qui peut être plus élevé que celui requis par les techniques d'optimisation de l'état de l'art. En revanche, on peut attendre d'une telle approche de bonnes propriétés de généralisation entre différentes tâches et contextes grâce à l'apprentissage de représentations factorisées au niveau de la couche cachée (Sallans et Hinton 2004).

La généralisation de Q-valeurs à de nouveaux contextes peut être envisagée de deux manières différentes. La plus immédiate consiste à concaténer un vecteur de contexte au vecteur état/action pour entraîner la RBM. La deuxième consiste à utiliser des RBMs à plusieurs entrées (Krizhevsky, Hinton et al. 2010), de manière similaire aux réseaux *gated*, pour coder d'un côté le contexte, et de l'autre côté le vecteur état/action.

Ceci constitue une piste de recherche qui n'est à notre connaissance pas explorée.

L'adaptation de l'algorithme à un cadre théorique reposant sur les autoencodeurs est aussi une question ouverte. Dans (Kamyshanska et Memisevic 2013) les auteurs proposent une mesure de *score* reflétant l'adéquation entre un vecteur d'entrée et les capacités de représentation apprises par un autoencodeur. Cette mesure repose sur une interprétation d'un autoencodeur comme un système dynamique, en considérant des reconstructions successives d'un vecteur d'entrée initial jusqu'à atteindre un point fixe pour la reconstruction. Ces reconstructions successives forment un champ de vecteurs dans l'espace d'entrée, dont on peut montrer sous certaines conditions qu'il forme en réalité un champ de gradient, définissant ainsi une énergie potentielle. Fait remarquable, l'énergie potentielle ainsi calculée pour un autoencodeur à neurones sigmoïdes est égale à l'énergie libre d'une RBM à neurones binaires (Kamyshanska et Memisevic 2013). Ce résultat peut donner une piste pour adapter l'apprentissage d'un autoencodeur afin de refléter des Q-valeurs.

Model-based et *model-free*

Deux grandes familles de modèles ont été développées par la communauté de l'apprentissage par renforcement : les approches *model-based* et *model-free* (Sutton et Barto 1998 ; Dayan et Niv 2008). La première s'appuie sur un modèle du monde pour prévoir les effets des actions et les récompenses atteignables afin de planifier la meilleure politique, tandis que la seconde apprend à associer à chaque état l'action qui donne la meilleure récompense espérée. De par leur différences, différentes familles d'algorithmes ont été développées de manière indépendante pour chacune des deux approches.

La combinaison de ces deux modèles est un champ de recherches actif (Huys et al. 2012 ; Simon et Daw 2012 ; Lesaint et al. 2014 ; Renaudo et al. 2014 ; Daw et Dayan 2014). Souvent, le *model-based* est utilisé dans un premier temps pour trouver une politique optimale, qui sert par la suite à entraîner petit à petit un *model-free* : l'agent développe des habitudes. La plupart des approches *model-based* sont toutefois confrontées à un problème d'envergure : dans un environnement complexe et réaliste, les possibilités d'action sont beaucoup trop nombreuses pour pouvoir être toutes explorées et il faut s'appuyer sur des heuristiques afin de contraindre l'exploration. On peut également interroger l'approche consistant à développer deux modèles distincts fonctionnant en parallèle,

apprenant chacun leur propre représentation de l'environnement et les valeurs associées et dont les sorties sont fusionnées tardivement : ceci semble impliquer un gaspillage important de ressources.

Nous esquissons une approche gommant cette frontière entre *model-based* et *model-free*. Tout d'abord, il convient de remarquer que le principe d'entraînement des réseaux profonds, dans lequel la prédiction, par exemple de l'état futur, est centrale, convient parfaitement à un cadre *model-based* : ce dernier peut s'appuyer sur un modèle implicite du monde auquel il peut accéder par simulation. Contrairement à des approches s'appuyant sur des processus markoviens qui requièrent habituellement le stockage explicite du modèle du monde (on connaît alors tous les états possibles et toutes les transitions entre états), les réseaux profonds peuvent permettre de simuler le résultat d'une politique à tester mais rendent plus difficile l'énumération des transitions possibles. Ceci introduit la nécessité d'avoir un mécanisme pour suggérer les politiques à tester qui, quant à lui, s'accommode assez bien d'un algorithme de type *model-free* : étant donné l'état actuel, l'expérience passée suggère que certaines actions paraissent être meilleures que d'autres. Cet algorithme *model-free* pourrait alors s'appuyer sur l'algorithme de (Sallans et Hinton 2000) décrit au début de cette partie.

Détaillons plus avant le principe de l'architecture. Étant donné un état $\mathbf{s_t}$, le réseau *model-free* de la section précédente échantillonne n actions (possiblement plusieurs fois la même) qui semblent pertinentes (Q-valeurs assez élevées). Ces actions sont alors fournies à un réseau profond, entraîné à prédire leurs conséquences en terme d'état résultant $\widehat{\mathbf{s}}_{\mathbf{t+1}}$. Cet état peut alors être utilisé pour itérer le mécanisme jusqu'à tomber soit sur l'état désiré, soit sur un état récompensé, soit jusqu'à ce qu'un nombre d'itérations maximal ait été atteint. Il faut alors postuler l'existence supplémentaire d'une mémoire de travail capable de stocker la séquence d'actions explorées [2] pour pouvoir l'exécuter réellement dans l'environnement. La figure 6.1 illustre ce fonctionnement.

Ce mécanisme global n'est pas optimal, dès lors que les actions sont proposées par un algorithme *model-free* qui lui-même n'est pas optimal. Il se démarque en cela des recherches qui s'attachent à l'optimalité de la méthode. Il possède cependant l'avantage de s'adapter à des environnements complexes et continus difficiles à gérer avec des approches classiques. De plus, certains de ses défauts ne semblent pas si absurdes que cela par comparaison au comportement humain :

- Tant que la bonne action n'a pas été proposée par le *model-free* (ce qui peut prendre un temps indéfini), l'algorithme ne peut pas l'explorer et ne peut donc pas la trouver, même si elle est par ailleurs connue : il a la solution *sur le bout de la langue*.
- Une fois la solution découverte (ou montrée par un autre agent), la mise à jour du *model-free* peut permettre de la proposer immédiatement face à une même situation : la solution peut donc devenir "triviale" dès qu'elle a été trouvée.
- Des actions qui ont été récompensées dans le passé de l'individu vont avoir tendance à être proposées plus souvent par le *model-free* et donc à être explorées plus souvent par le *model-based*. Elles seront donc utilisées dès qu'elles permettent ef-

2. Il est également possible de mémoriser uniquement les premières actions de la séquence explorée, auquel cas le mécanisme d'exploration *model-based* est recommencé à chaque fin de séquence mémorisée.

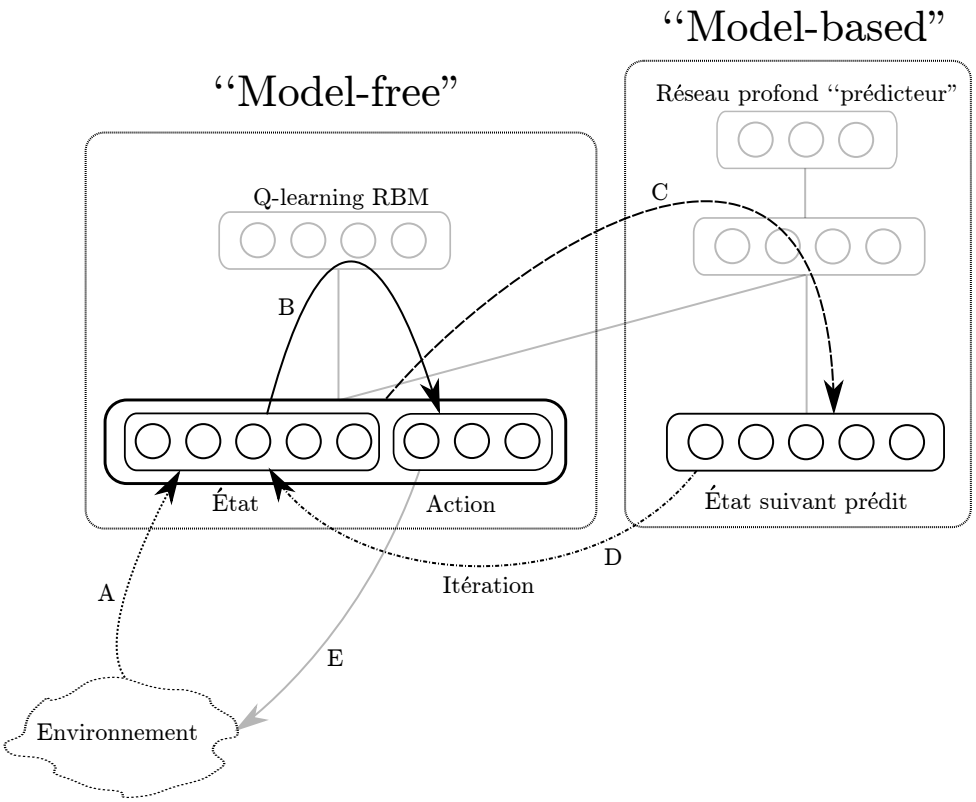


Figure 6.1 – *Illustration de l'architecture proposée pour fusionner model-based et model-free.* **A :** *un état est observé depuis l'environnement.* **B :** *une RBM échantillonne une action suivant les Q-valeurs apprises.* **C :** *Un réseau profond prédit le résultat de l'action en terme d'état suivant.* **D :** *ce nouvel état remplace l'état observé (ou l'état peut rester inchangé si l'on veut explorer plusieurs actions pour cet état, auquel cas on retourne à l'étape B pour tirer une nouvelle action à partir du même état). La boucle B-C-D peut être itérée plusieurs fois.* **E :** *les actions échantillonnées par la RBM peuvent être exécutées dans l'environnement (ou mises en mémoire de travail en attente d'exécution), selon un critère restant à définir (par exemple l'état prédit est en adéquation avec le but recherché).*
Dans cette architecture, aucun modèle du monde explicite n'est stocké. En particulier, il est impossible de lister de manière exhaustive les transitions entre états. Le modèle de l'environnement n'est accessible que de manière implicite, en simulant les résultats de certaines actions dans certains états.

fectivement d'atteindre le but recherché, même si elles ne sont pas optimales dans l'absolu. L'algorithme est donc susceptible de développer des "manies" et des biais comportementaux.

Il est important de remarquer que l'architecture proposée n'est pas spécifique aux réseaux profonds : n'importe quels algorithmes *model-based* et *model-free* peuvent être utilisés, pour peu que l'on puisse influer sur le mécanisme de planification de l'algorithme *model-based*. En revanche, l'utilisation de réseaux profonds fournit un cadre naturel pour sa mise en œuvre : les réseaux profonds sont naturellement conçus pour être capables de prédire les conséquences d'une action, et l'algorithme de renforcement à base de RBMs possède des propriétés intéressantes de généralisation à de nouveaux états ou contextes et permet un échantillonnage aléatoire des actions selon leur Q-valeur. De plus, à la différence d'autres approches populaires, ils ne nécessitent pas de stockage explicite du modèle du monde ni de table de Q-valeurs, à la source de grandes difficultés de passage à l'échelle.

L'approche proposée permet d'unifier les deux types d'algorithmes, dans le sens où le *model-based* ne peut exister sans le *model-free* puisqu'aucun stockage explicite des états et actions possibles n'existe. En particulier, l'algorithme *model-based* ne repose sur aucune valeur ou récompense propre, lesquelles sont gérées par le *model-free*. On peut de plus remarquer que le *model-free* est strictement équivalent à un *model-based* non itéré.

Nous n'avons qu'esquissé une idée d'architecture. De nombreuses questions restent en suspens, notamment sur les mécanismes permettant d'arrêter la planification pour réellement exécuter une action, en particulier dans le cas où la planification n'a pas permis de trouver une politique permettant d'atteindre le but fixé. La question du stockage en mémoire de travail des actions explorées est elle aussi ouverte. Par ailleurs, une idée proche concernant les liens entre *model-free* et *model-based* a récemment été suggérée dans la discussion de (Daw et Dayan 2014) :

> *One possibility invited by these refined interactions between [model-based] and [model-free] systems is to consider [model-based] evaluation at a much finer grain. We can envisage a modest set of operations : such things as creating in working memory a node in a search tree ; populating it with an initial [model-free] estimate ; sampling one or more steps in the tree using a model ; backing value information up in the current tree, possibly improving [a model-free] estimate using the resulting model-influenced (though not necessarily purely [model-based]) prediction error and finally picking an external action.*

6.2 Apprentissage profond et curiosité artificielle

Durant leur développement, les enfants sont exposés à des environnements complexes en perpétuelle évolution, ce qui fait de l'apprentissage un problème très complexe. L'apprentissage demande alors un engagement actif vis-à-vis de l'environnement pour en explorer sélectivement certaines facettes, ce que font les enfants en jouant par exemple. On observe de plus que les activités choisies spontanément correspondent à des situations ni totalement familières, ni totalement nouvelles : elles se situent à la frontière des situations correctement maîtrisées (Berlyne 1960 ; Csikszentmihalyi 1991).

Plusieurs modèles computationnels ont été proposés pour imiter ce développement (Oudeyer et al. 2007 ; Schmidhuber 2010). La plupart définissent un progrès d'apprentissage comme la dérivée d'une mesure de compétence. Par exemple, pour un agent qui apprend un modèle de son environnement, la mesure de compétence peut être définie comme l'erreur de prédiction de ses entrées sensorielles et le progrès d'apprentissage comme sa dérivée. Le module de curiosité artificielle partitionne alors l'espace sensorimoteur afin de se focaliser sur les sous-espaces pour lesquels le progrès moyen est maximal. Comme expliqué dans (Oudeyer et al. 2007), utiliser la dérivée de la compétence et non la compétence elle-même permet d'éviter certains effets indésirables, comme répéter indéfiniment les tâches les plus faciles (si l'agent est attiré par les zones de compétence maximale) ou au contraire rester bloqué à essayer d'apprendre des relations complètement aléatoires (si l'agent se focalise sur les zones de compétence minimale).

La curiosité artificielle peut donc être vue comme un cas particulier d'apprentissage par renforcement, pour lequel la récompense associée à un couple état/action est le progrès d'apprentissage résultant. Il est donc possible d'utiliser l'architecture exposée dans la partie précédente pour l'adapter au problème de la curiosité artificielle.

L'utilisation de cet algorithme permet de profiter des capacités de généralisation des RBMs pour se passer d'un mécanisme de partitionnement de l'espace, en approchant directement la valeur du progrès d'apprentissage sur tout l'espace, et non région par région.

Il reste toutefois le problème du calcul du progrès d'apprentissage. Dans le cadre des architectures profondes et des algorithmes que nous avons présentés dans cette thèse, une mesure naturelle de l'apprentissage s'impose : l'erreur de reconstruction (ou de prédiction) atteinte par l'architecture. L'utilisation de cette mesure, que nous noterons $\mathcal{L}$ par la suite, a une conséquence importante car elle permet de calculer très simplement le progrès d'apprentissage de manière instantanée, sans surcoût computationnel et sans nécessiter plusieurs mesures.

En effet, la règle de mise à jour des poids au pas de temps t après présentation d'une entrée $\mathbf{x_t}$ s'écrit :

$$W_{t+1} = W_t - \lambda \frac{\partial \mathcal{L}}{\partial W}(\mathbf{x_t}). \tag{6.5}$$

Étant donné que

$$\frac{\partial \mathcal{L}}{\partial t}(\mathbf{x_t}) = \frac{\partial \mathcal{L}}{\partial W}(\mathbf{x_t}) \frac{\partial W}{\partial t}(t) \tag{6.6}$$

et que

$$W_{t+1} - W_t \approx \frac{\partial W}{\partial t}(t), \tag{6.7}$$

la dérivée de la compétence peut se réécrire

$$\frac{\partial \mathcal{L}}{\partial t}(\mathbf{x_t}) \approx -\lambda \left\| \frac{\partial \mathcal{L}}{\partial W}(\mathbf{x_t}) \right\|^2 \approx -\frac{1}{\lambda} \left\| \frac{\partial W}{\partial t}(t) \right\|^2. \tag{6.8}$$

L'entrée qui génère le progrès d'apprentissage le plus important, c'est-à-dire l'entrée pour laquelle l'erreur de reconstruction décroît le plus rapidement, est donc l'entrée qui génère la modification des poids du réseau la plus importante.

Il faut souligner que cette approche diffère des approches usuelles pour lesquelles le progrès est mesuré de manière externe et différée (le progrès à l'instant t ne peut être calculé que lorsqu'une nouvelle mesure de performance est obtenue pour le même sous-espace d'entrée). L'approche proposée s'appuie quant à elle sur une estimation interne et instantanée de ce progrès, ce qui peut mener à de fausses croyances de progrès. Prenons l'exemple d'une action qui provoque un retour sensoriel correspondant à un bruit pur. Chaque nouveau stimulus est donc mal prédit et entraîne une modification des poids synaptiques, donc une mesure de progrès non nulle, alors que les conséquences de cette action sont en réalité non prédictibles et que cette action devrait donc être abandonnée. Le progrès mesuré dépend cependant fortement des représentations apprises par le réseau. Il suffit par exemple que des neurones sigmoïdes saturent pour que le gradient soit annulé et que le progrès soit donc nul.

En outre, l'introduction d'une approche de curiosité artificielle pour l'entraînement des réseaux profonds peut être un élément de solution pour contourner le problème des plateaux de gradient dont nous avons parlé au chapitre 3, en permettant de sélectionner les entrées pour lesquelles un gradient d'apprentissage maximal est obtenu. Elle rejoint en cela l'idée du *curriculum learning* (Bengio et al. 2009) qui consiste à complexifier l'ensemble d'apprentissage au fur et à mesure de l'entraînement. Utilisée pour l'entraînement supervisé de réseaux de neurones, cette méthode permet d'avoir une convergence plus rapide et de meilleurs résultats finaux. Néanmoins, à la différence d'une approche reposant sur une motivation intrinsèque, le *curriculum learning* fait appel à un expérimentateur pour définir la manière dont la complexité de l'ensemble d'apprentissage évolue au cours de l'entraînement. De ce point de vue, le *curriculum learning* peut être considéré plus proche des algorithmes à base de contraintes maturationelles comme (Baranes et Oudeyer 2011) que des algorithmes de curiosité artificielle pure.

6.2.1 Preuve de concept

En tant que preuve de concept pour l'approche proposée, nous considérons un protocole expérimental dans lequel un agent est confronté à trois variantes de la base MNIST, de différentes complexités (voir figure 6.2) :

- MNIST-basic : il s'agit de la base MNIST standard
- MNIST-bg-im : une image aléatoire est ajoutée en fond
- MNIST-bg-rand : le fond de chaque image est composé de pixels tirés aléatoirement selon un bruit gaussien

Pour l'expérience, nous introduisons une association très simple entre action et perception : chacun des 10 chiffres de chacune des variantes MNIST est associé à une action différente, ce qui fait un total de 30 actions, qui peuvent donc être décrites comme "montrez-moi un exemple du chiffre i de la base d". Il faut insister sur le fait que, même si les labels sont utilisés pour définir les différentes actions, l'algorithme n'y a aucun accès direct et l'apprentissage reste totalement non supervisé. Nous utilisons cette expérience de principe pour simuler le fait que, dans des environnements plus complexes, différentes actions produisent des résultats différents en terme de stimuli perçus, de plus ou moins grande complexité.

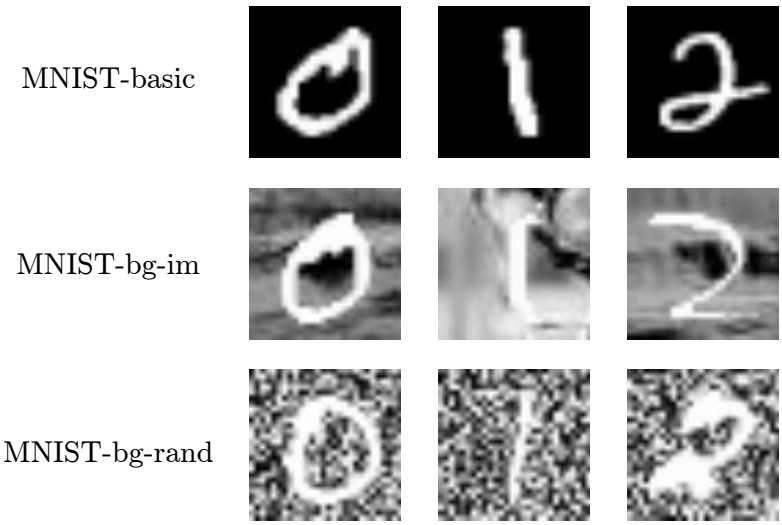


Figure 6.2 – *Variantes de la base MNIST utilisées pour l'expérience. MNIST-bg-im est obtenue par ajout d'une portion d'image naturelle en fond, tandis que MNIST-bg-rand est obtenue par ajout de bruit gaussien.*

Le but de l'expérience est d'entraîner un autoencodeur à représenter ces chiffres. Nous utilisons un simple autoencodeur avec 784 neurones visibles (images de 28x28 pixels) et 500 neurones cachés. L'apprentissage est régularisé par débruitage : 30% des neurones de la couche visible sont mis à 0 pour chaque donnée. La RBM utilisée pour approximer la Q-fonction est composée de 30 neurones visibles (un par action), et de 15 neurones cachés. Afin de rendre compte du codage de l'action sous la forme d'un vecteur "un contre tous", nous utilisons une fonction d'activation *softmax* au niveau de la couche visible de la RBM. L'action est ensuite choisie aléatoirement selon la distribution de probabilité obtenue. Étant donné qu'aucun aspect temporel n'intervient dans cette expérience, le coefficient d'actualisation γ de l'algorithme Sarsa (équation 6.3) est fixé à 0. Le taux d'apprentissage est quant à lui de 0.1 pour la RBM et 0.005 pour l'autoencodeur.

Apprentissage actif

Nous commençons par étudier comment évolue l'action choisie au cours de l'apprentissage. Nous calculons donc le nombre de sélections de chaque variante MNIST par le réseau (nous regroupons pour chaque variante les 10 actions correspondant à chacun des chiffres). Les résultats sont lissés à l'aide d'une fenêtre glissante sur 100 itérations. La figure 6.3 illustre les résultats obtenus.

La figure 6.4 représente quant à elle le choix des actions au tout début de l'apprentissage (pour les 100 premières itérations), pour une des répétitions (représentative des autres).

On voit qu'après une courte phase d'initialisation où chaque variante est choisie uni-

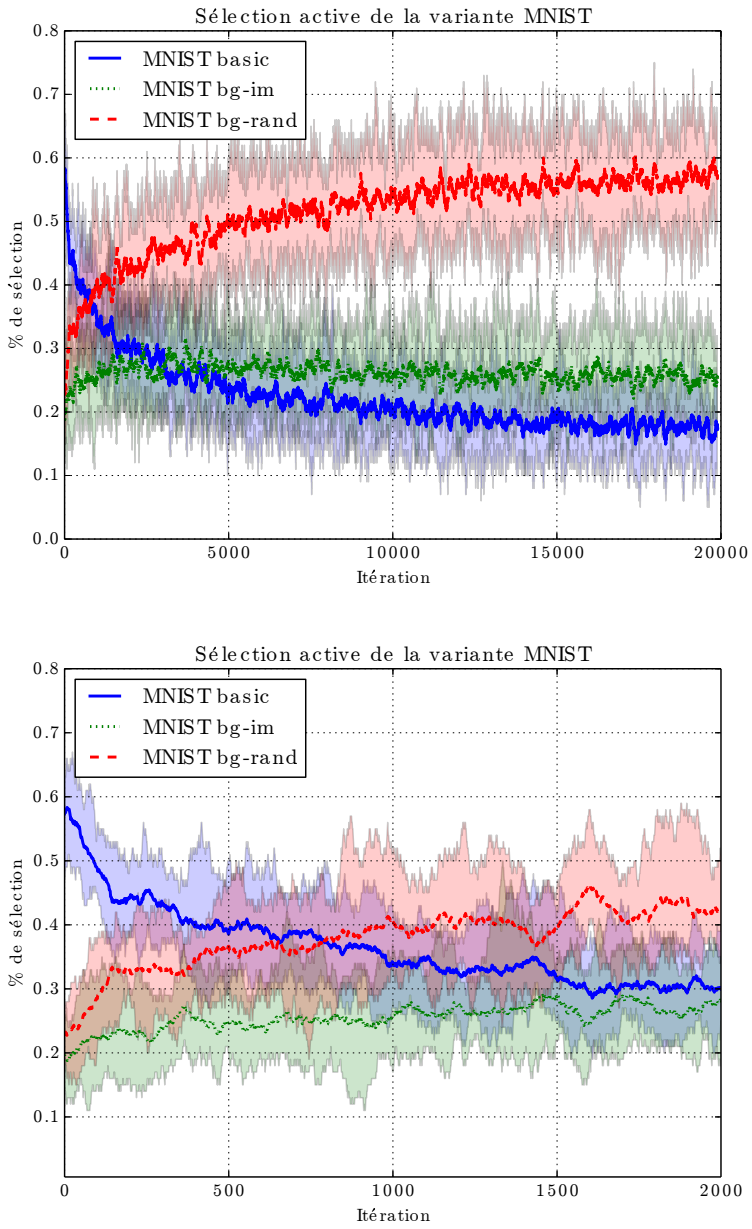


Figure 6.3 – *Évolution de la variante MNIST choisie au cours de l'apprentissage (en bas : zoom sur les 2000 premières itérations). Le graphique représente les résultats obtenus sur 15 répétitions de l'expérience (les courbes pleines représentent la moyenne, tandis que les enveloppes correspondent aux valeurs minimales et maximales). Après une très courte phase d'initialisation où chaque variante est choisie uniformément (voir figure 6.4), le réseau commence par choisir majoritairement la variante MNIST de base. Après environ 2000 itérations, la probabilité de choisir cette variante passe sous la barre uniforme de 33% tandis que la probabilité de choisir la variante avec fond aléatoire augmente.*

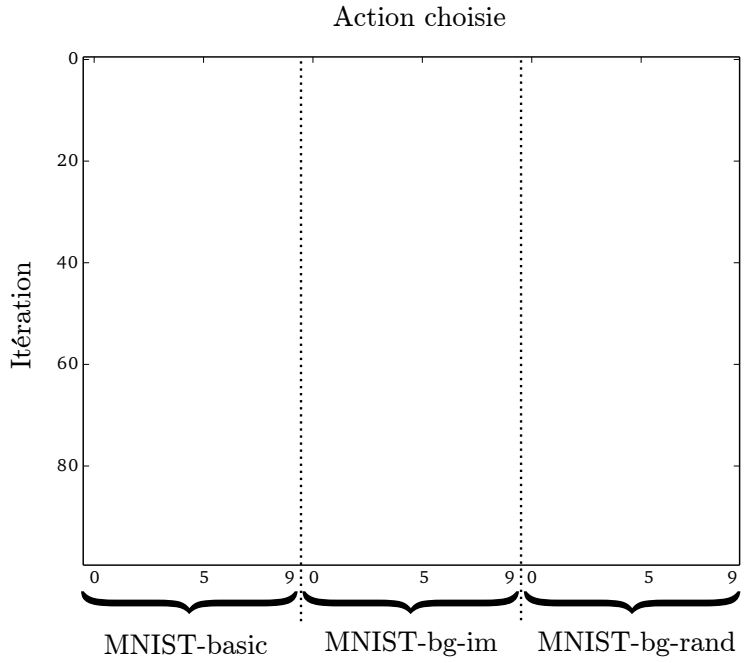


Figure 6.4 – *Actions choisies par l'algorithme au cours des 100 premières itérations. L'algorithme se focalise très rapidement sur les 10 premières actions, qui correspondent aux chiffres de la variante MNIST de base.*

formément, le réseau commence par choisir majoritairement la variante MNIST de base. Après environ 2000 itérations, la probabilité de choisir cette variante passe sous la barre uniforme de 33%. Le réseau ne présente pas de phase pendant laquelle il choisirait prioritairement la variante avec des images en fond, mais passe directement à la variante au fond aléatoire. C'est une conséquence probable des "fausses croyances de progrès" induite par le bruit gaussien de la variante bg-rand.

Accélération de l'apprentissage

Nous étudions maintenant comment l'algorithme proposé influence l'erreur de reconstruction et la vitesse d'apprentissage. Nous comparons les performance de l'autoencodeur entraîné avec sélection active de l'entrée avec un autoencodeur pour lequel chaque entrée est choisie avec une probabilité uniforme entre les 30 classes disponibles. Les résultats sont reproduits sur la figure 6.5. La figure 6.6 illustre quant à elle les reconstructions obtenues par les deux approches.

Avec l'approche proposée, l'erreur de reconstruction de la variante MNIST de base décroît de manière beaucoup plus rapide au début de l'apprentissage. Cependant, la différence s'annule puis s'inverse au bout d'environ 2500 itérations. Au final, après 20000 itérations, l'erreur de reconstruction des variantes MNIST-basic et MNIST-bg-im est plus élevée avec notre approche. Au contraire, une erreur légèrement plus faible pour MNIST-bg-rand peut laisser supposer un léger sur-apprentissage de cette variante.

La figure 6.7 montre quant à elle l'évolution de la différence des normes des gradients

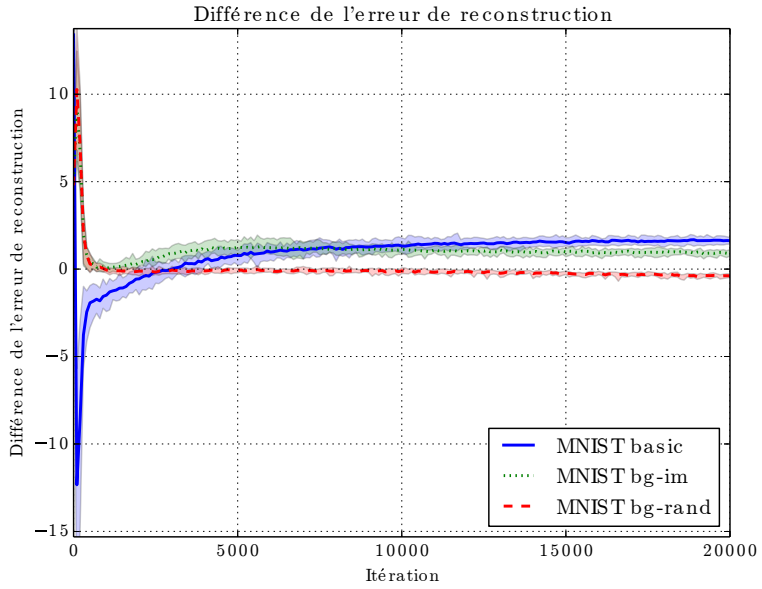


Figure 6.5 – *Évolution de la différence entre l'erreur de reconstruction obtenue par un autoencodeur entraîné avec sélection active de ses entrées et l'erreur de reconstruction d'un autoencodeur entraîné sur des entrées choisies aléatoirement. Comme pour le graphique 6.3, nous représentons les valeurs extrêmes ainsi que les valeurs moyennes obtenues sur 15 répétitions de l'expérience. Avec l'approche proposée, l'erreur de reconstruction de la variante MNIST de base décroît de manière beaucoup plus rapide au début de l'apprentissage. Cependant, la différence s'annule et s'inverse au bout d'environ 2500 itérations. Après 20000 itérations, l'erreur de reconstruction des variantes MNIST-basic et MNIST-bg-im est plus élevée avec l'approche proposée qu'avec un entraînement standard. Au contraire, l'erreur est très légèrement plus faible pour MNIST-bg-rand, ce qui peut laisser supposer un léger sur-apprentissage pour cette variante.*

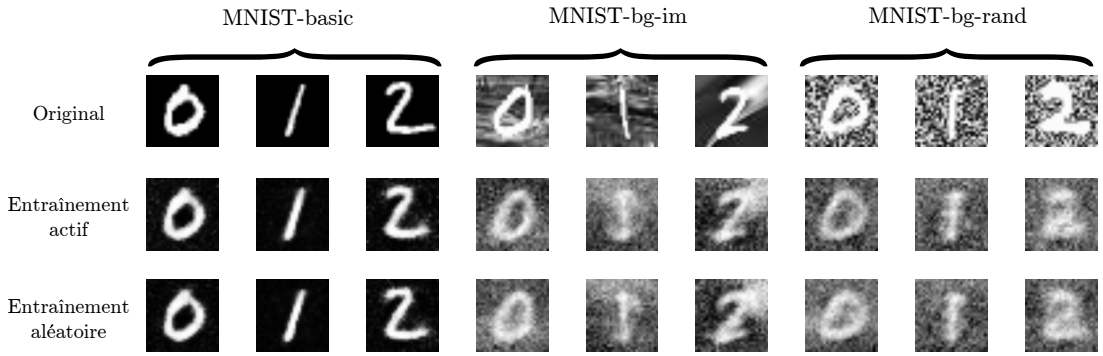


Figure 6.6 – *Exemple de reconstructions de chaque variante MNIST au bout de 20000 itérations. En haut : images originales ; milieu : reconstructions obtenues avec un autoencodeur entraîné avec sélection active des entrées ; bas : reconstructions obtenues avec un autoencodeur entraîné avec sélection aléatoire des entrées. Malgré la différence de reconstruction mesurée (figure 6.5), celle-ci n'est pas toujours visuellement très marquée. On remarquera cependant la reconstruction du chiffre "1" de la base MNIST-bg-im qui peut ressembler plus à un 0 ou un 8 avec notre approche. Ceci laisse supposer qu'avoir appris prioritairement la base MNIST-basic a pu biaiser les représentations vers les chiffres naturels.*

obtenus au cours de l'entraînement de l'autoencodeur. De manière étonnante, les modifications sont plus faibles pendant les 4000 premières itérations approximativement. Elles deviennent cependant plus importantes par la suite et sont environ 15% plus importantes après 20000 itérations. Ceci peut sembler contradictoire avec le fait que l'algorithme proposé est conçu pour maximiser cette norme. Cependant, comme on peut le voir sur la figure 6.8, les premières actions obtiennent un gradient plus important, ce qui peut avoir tendance à amener rapidement le réseau dans des zones de faible gradient. Au contraire, avec une sélection aléatoire des entrées, les directions prises à chaque itération peuvent être plus aléatoires et donc mener le réseau à rester plus longtemps dans des zones de gradient plus élevé (voir figure 6.9). De même, il est peu clair si le gradient plus important obtenu après 4000 itérations est dû à une réelle amélioration de l'apprentissage grâce à notre approche, ou s'il résulte d'une marche plutôt aléatoire provoquée par la plus grande probabilité de sélection de la variante MNIST-bg-rand. C'est pourquoi dans la section suivante, nous tentons de caractériser la pertinence des représentations apprises.

Performance

Nous avons montré dans les sections précédentes que notre approche induit une distribution non-stationnaire des entrées. Étant donné que les autoencodeurs et les RBMs apprennent à représenter la distribution de probabilité de leurs entrées, cela soulève la question de savoir si notre approche détériore les représentations apprises. Afin de tester ces dernières, nous utilisons les sorties produites par l'autoencodeur pour entraîner un classifieur linéaire à classifier les 10 chiffres. Nous comparons les représentations apprises en utilisant notre approche avec les représentations apprises par un autoencodeur entraîné de manière standard, tous les autres paramètres étant égaux.

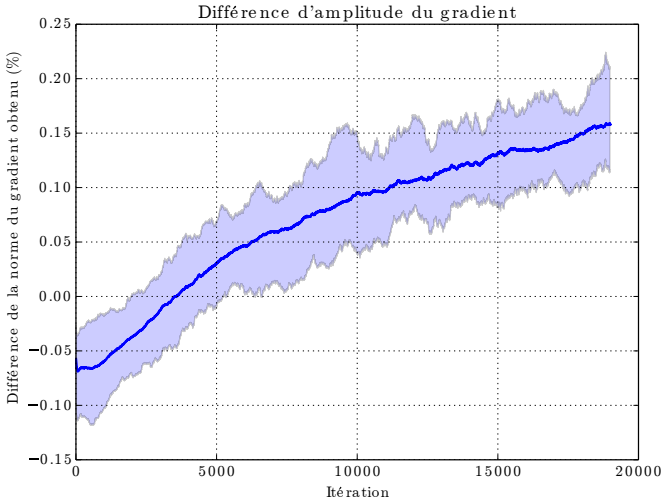


Figure 6.7 – *Évolution du gradient obtenu au cours de l'apprentissage, lissé par moyenne sur une fenêtre glissante de 1000 itérations. Jusqu'à la 4000ème itération, le gradient obtenu avec notre approche est plus faible, tandis qu'il devient supérieur par la suite (il est 15% supérieur au bout de 20000 itérations).*

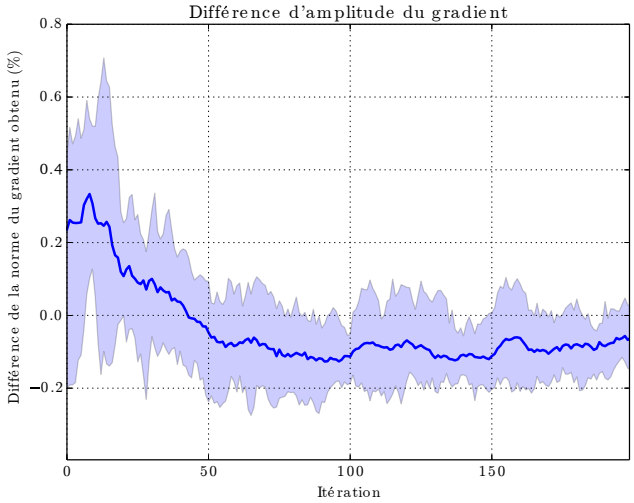


Figure 6.8 – *Évolution du gradient obtenu pour les 200 premières itérations, lissé par moyenne sur une fenêtre glissante de 10 itérations. Au tout début de l'apprentissage, le gradient obtenu avec notre approche est plus important. Le réseau va donc avoir tendance à être mené dans des zones au gradient plus faible, ce qui explique par la suite qu'il obtienne un gradient plus faible (voir figure 6.7) le temps que l'entraînement standard parvienne dans une zone similaire de gradient.*

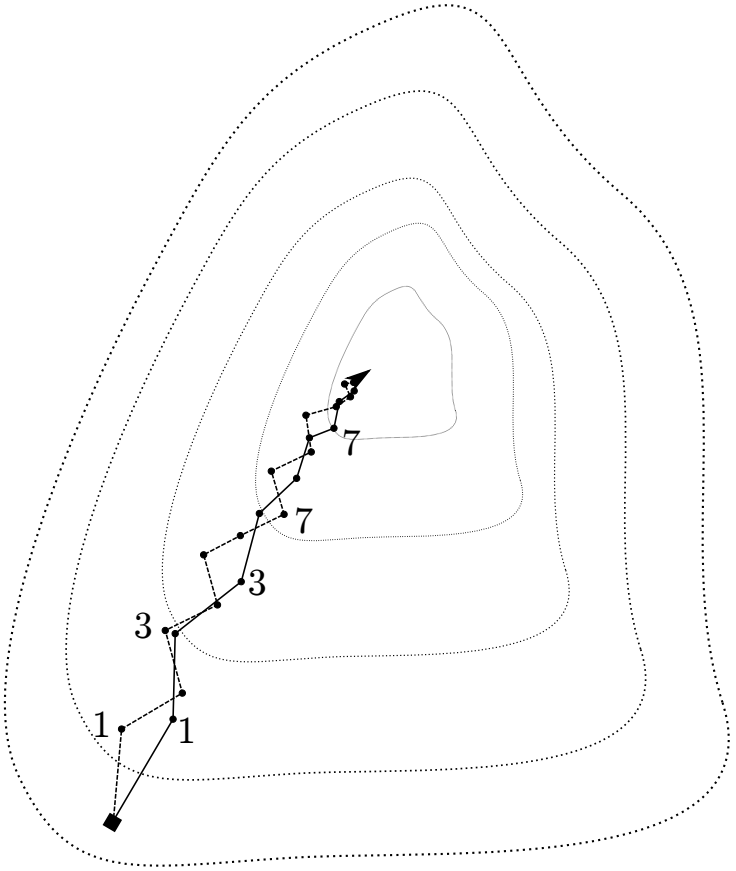


Figure 6.9 – *Ce schéma illustre le fait qu'une sélection active des entrées impliquant les gradients d'apprentissage les plus élevés (courbe pleine) peut se traduire par un gradient plus faible à une même itération qu'avec une sélection aléatoire des entrées (courbe pointillée). En effet, les premières itérations vont amener plus rapidement l'algorithme dans des zones de plus faible gradient. Ainsi, si pendant les premières itérations le gradient obtenu est plus élevé (par exemple pour les itérations 1 et 3), il peut devenir plus faible par la suite (itération 7 par exemple).*

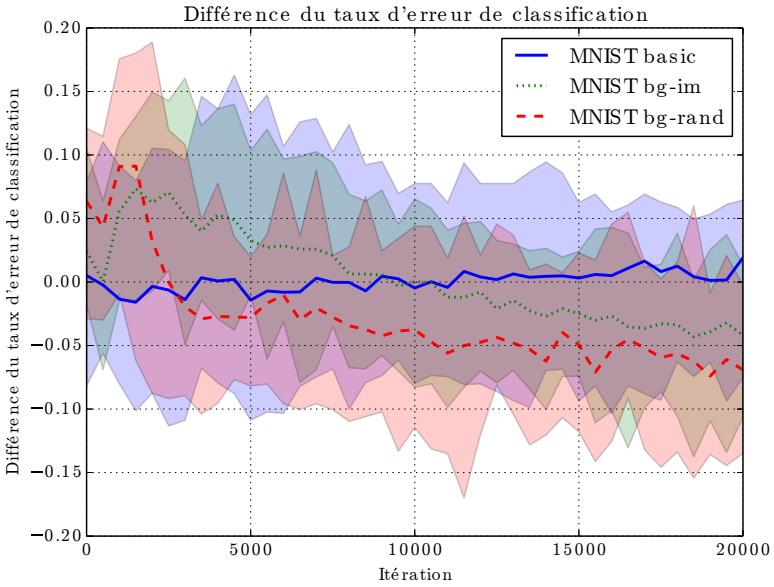


Figure 6.10 – *Évolution de la différence entre l'erreur de classification obtenue par un autoencodeur entraîné avec sélection active de ses entrées et l'erreur de classification obtenue par un autoencodeur entraîné sur des entrées choisies aléatoirement. Comme pour le graphique précédent, nous représentons les valeurs extrêmes ainsi que les valeurs moyennes obtenues sur 15 répétitions de l'expérience. Celle-ci ne présente pas de différence significative en ce qui concerne la variante MNIST standard. En revanche, pour les deux autres variantes, il y a une amélioration absolue de l'ordre de 2%, ce qui représente une amélioration relative notable de 6 à 10% environ (l'autoencodeur entraîné de manière standard atteint une performance de classification comprise entre 20 et 30%).*

Le classifieur linéaire est entraîné à l'aide d'un ensemble d'entraînement et d'un ensemble de validation, avec un taux d'apprentissage de 0.13 pour 500 itérations. Nous rapportons la performance sur un ensemble de test distinct correspondant au meilleur score de validation obtenu au cours de l'apprentissage. Chaque ensemble (apprentissage, validation et test) contient 200 images de chaque chiffre. Nous soulignons que nous n'étudions pas la performance de classification en tant que telle, mais plutôt la façon dont elle est influencée par la sélection active des entrées.

La différence entre l'erreur de classification obtenue avec notre approche et avec l'approche standard est rapportée figure 6.10. Celle-ci ne présente pas de différence significative en ce qui concerne la base MNIST standard. En revanche, l'amélioration est significative pour les deux autres variantes.

Bien que l'erreur de reconstruction obtenue avec notre approche sur les variantes bg-im et bg-rand de MNIST soit plus élevée que dans le cas d'une approche standard, notre approche permet d'atteindre un meilleur taux de reconnaissance lorsque les représentations apprises sont utilisées par un classifieur linéaire. On peut faire l'hypothèse que le fait de se focaliser d'abord sur la base MNIST standard permet d'apprendre une représentation

des chiffres plus pertinente (puisque non perturbée par la présence d'un fond). Une fois ces représentations apprises sur la base MNIST, celles-ci sont en effet utiles pour décrire les deux autres variantes, ce qui leur confère *a priori* une certaine stabilité, même si la variante MNIST-basic est par la suite beaucoup moins souvent sélectionnée.

6.2.2 Pistes de recherche

Notre preuve de concept s'appuie sur une expérience simpliste, sans notion d'état ni d'aspect temporel. Des analyses plus détaillées doivent être menées dans des cas plus généraux sur des réseaux et des données plus complexes.

Par ailleurs, nos expériences semblent fortement impactées par les fausses croyances, qui amènent l'algorithme à se focaliser sur les données aléatoires. Ceci peut être induit par le protocole même de l'expérience, menée sur un ensemble de données très peu variées. Confronter l'algorithme à un environnement ouvert et de plus grande complexité semble donc nécessaire afin d'étudier son comportement dans des conditions plus réalistes.

Nous avons également suggéré d'introduire un paramètre de température pour réguler l'exploration et l'exploitation. Comment l'utiliser à bon escient ? Une méthode évidente peut consister à le faire décroître au cours du temps, faisant ainsi diminuer l'exploration au profit de l'exploitation. Cependant, d'autres critères peuvent être plus pertinents. Par exemple, le faire dépendre du progrès d'apprentissage : quand celui-ci devient trop faible, il peut être intéressant d'augmenter la température afin de favoriser l'exploration, pour permettre éventuellement de découvrir une nouvelle zone où le progrès d'apprentissage pourra être plus important.

Nous avons illustré notre approche dans le cadre d'une curiosité artificielle appliquée aux actions, or il a été montré que des performances supérieures pouvaient être obtenues en utilisant la curiosité pour se fixer des objectifs dans un espace de buts (Baranes et Oudeyer 2013). Si rien ne s'oppose à utiliser l'architecture présentée pour générer des vecteurs représentant des buts à atteindre (en lieu et place des actions de notre exemple), la définition d'une mesure de progrès interne et instantanée est toutefois plus complexe.

Enfin, dans un cadre entièrement développemental, la représentation des états et des actions évolue dans le temps. La signification des vecteurs utilisés en entrée/sortie de la RBM change donc au cours de l'apprentissage. Les effets d'une telle interaction nécessitent de plus amples recherches.

Chapitre 7

Discussion

Dans une discussion, le difficile, ce n'est pas de défendre son opinion, c'est de la connaître.

André Maurois - De la conversation

Sommaire

Nous avons présenté dans les chapitres précédents différentes architectures à base de réseaux de neurones non-supervisés. Plusieurs critiques sont régulièrement adressées à l'apprentissage profond, dont certaines sont particulièrement pertinentes dans un cadre développemental : comment traiter le cas d'environnements non stationnaires et réduire le temps de calcul nécessaire à l'apprentissage ? Nous discutons ces questions dans une première partie à partir de quelques travaux récents susceptibles d'apporter des éléments de réponse. Dans la seconde partie, nous revenons sur nos résultats et en discutons les limites pour une application dans des environnements naturels.

7.1 Apprentissage profond et apprentissage permanent, temps réel

Plusieurs critiques peuvent être adressées à l'encontre de l'utilisation des réseaux profonds pour la robotique développementale, qui implique un apprentissage permanent dans des environnements changeants, ainsi qu'un apprentissage en temps réel.

7.1.1 Apprentissage permanent et lutte contre l'oubli

Les algorithmes d'apprentissage profond généralement utilisés supposent que la distribution des données d'entrée est constante au cours de l'apprentissage. Or ceci n'est pas le cas dans un cadre autonome et développemental : les stimuli rencontrés évoluent au cours de l'apprentissage et du développement des capacités du robot, et dépendent de plus de la tâche effectuée. Les algorithmes d'apprentissage doivent donc faire face au dilemme stabilité/plasticité.

Nous avons toutefois vu dans la section précédente qu'une distribution changeante, à cause d'une sélection active des entrées, n'impactait pas forcément négativement les performances. Le problème de l'oubli n'en est pas moins une vraie question : comment ne pas oublier des stimuli qui sont rencontrés assez rarement, mais néanmoins de façon récurrente et ayant donc intérêt à être mémorisés ?

Une piste intéressante a été proposée par (Calandra et al. 2012) pour des RBM. Elle part de la remarque que ces architectures sont entraînées à reproduire leurs entrées selon la distribution de probabilité rencontrée au cours de l'apprentissage. Dès lors, elles peuvent être utilisées pour générer de cette manière un nouvel ensemble d'apprentissage auquel sont mélangées les nouvelles données en provenance de l'environnement réel. De cette manière, une donnée rencontrée par le passé mais disparaissant momentanément de l'environnement sera réintroduite artificiellement dans l'ensemble d'apprentissage, évitant son oubli.

L'idée est intéressante, mais n'en est qu'à un stade très précoce de développement. Tout d'abord, tant que le réseau n'a pas été correctement entraîné, il génère du bruit. Si celui-ci est utilisé pour l'entraînement, il va au mieux ralentir l'apprentissage, au pire le conduire à renforcer des "fantaisies". L'adaptation de ce mécanisme aux algorithmes de type autoencodeur est également un problème. En effet, ceux-ci sont entraînés à reconstruire leurs entrées, mais non à les générer selon une distribution quelconque. L'utilisation de variantes de ces algorithmes, intégrant un aspect stochastique (Bengio et Thibodeau-Laufer 2013 ; Raiko et al. 2014), peut constituer une piste de recherche.

Une autre proposition a été développée dans (Pape et al. 2011). Elle consiste à utiliser différents modules indépendants et à n'entraîner pour chaque entrée que le module capable de la reconstruire au mieux. Combinée à un taux d'apprentissage adaptatif en fonction du nombre d'entrées que chaque module est capable de reconstruire mieux que les autres, les auteurs ont montré que cette approche permettait au réseau global de mémoriser une classe de données, même si celle-ci disparaît subitement de l'ensemble d'apprentissage.

7.1.2 Apprentissage temps réel

Une autre critique couramment adressée aux réseaux profonds est leur lenteur d'apprentissage. À titre d'exemple, la plupart des expériences que nous avons présentées dans cette thèse correspondent à des temps d'apprentissage de l'ordre d'une dizaine d'heure sur un ordinateur de bureau standard.

Cette critique appelle plusieurs commentaires. Premièrement, le temps d'apprentissage reste généralement inférieur ou du même ordre de grandeur que le temps nécessaire pour

collecter des données sur un robot réel. Deuxièmement, le développement d'architectures efficaces pour l'apprentissage et l'utilisation de réseaux profonds est un domaine très actif : parallélisation massive des algorithmes (notamment pour GPU) (Raina et al. 2009 ; Strigl et al. 2010 ; Dean et al. 2012 ; Coates et al. 2013), implémentation des algorithmes sur FPGA (Farabet et al. 2011) ou développement de puces dédiées (Gokhale et al. 2014). Dans les expériences que nous avons menées au cours de cette thèse, nous n'avons fait aucun effort particulier pour optimiser le temps d'exécution de nos algorithmes. Cependant, des réseaux bien plus importants peuvent être appris sur des durées très raisonnables avec les approches que nous venons de citer.

Une deuxième critique peut être portée sur le pré-entraînement de chaque couche de manière séquentielle, qui augmente le temps nécessaire à l'apprentissage puisque cela implique d'être confronté aux mêmes données (ou à des données similaires) à de multiples reprises tout au long de l'apprentissage. Nous avons vu au chapitre 3 qu'une des justification à ce pré-entrainement est la tendance à rester bloqué sur des plateaux de gradient lorsque l'on essaie d'entraîner un réseau en entier dès le début. L'utilisation de techniques à base de curiosité artificielle comme celle présentée dans la section précédente pourrait être un élément de solution. L'utilisation d'autres techniques de descente de gradient plus efficaces (Dauphin et al. 2014 ; Martens 2010) peuvent aussi être envisagées.

7.2 Des expériences de laboratoire aux environnements naturels

Nous avons mené toutes nos expériences dans des conditions de laboratoire, c'est-à-dire dans des environnements contraints et dans des situations "idéales". Le passage à des environnements naturels implique la prise en compte de problématiques complexes.

7.2.1 Rôle de l'attention

Dans les expériences que nous avons menées au cours de cette thèse, nous avons utilisé des entrées avec une dimensionalité importante mais raisonnable (de l'ordre de 600 dimensions au maximum dans le cas du réseau présenté au chapitre 4). En particulier, quand nous utilisions des images, celles-ci étaient redimensionnées afin d'en limiter la taille. Le passage à des cas d'utilisation concrets nécessite de pouvoir travailler avec des champs visuels beaucoup plus larges.

Les réseaux à convolution sont depuis longtemps utilisés dans le but de traiter de grandes images. Ils consistent à utiliser un réseau de taille modeste prenant une faible portion de l'image en entrée (de l'ordre de quelques centaines de pixels) et de l'appliquer à plusieurs reprises de manière à couvrir toute l'image. À chaque position du réseau sur l'image, on obtient donc des valeurs pour les neurones de sortie, qui peuvent à leur tour être considérées comme une nouvelle "image" (chaque neurone de sortie du réseau correspond à un canal de l'image, à la manière des canaux RGB sur une image réelle). Cette représentation peut donc à son tour être traitée par un réseau à convolution. Afin de réduire petit à petit la dimensionalité des représentations produites, les valeurs d'un

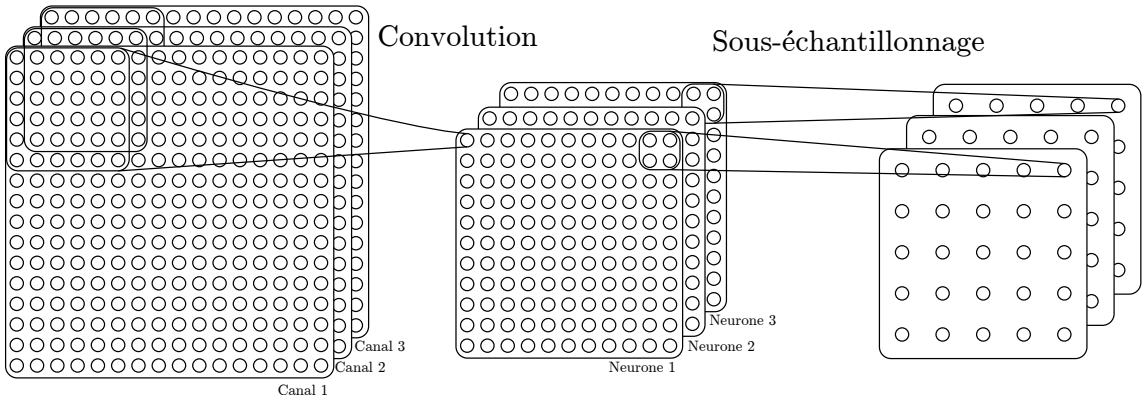


Figure 7.1 – *Schéma d'une couche d'un réseau à convolution. L'entrée (ici composée de trois canaux) est "filtrée" par un réseau (ici de* $6 \times 6 \times 3$ *neurones en entrée), dont chaque sortie produit à son tour une "carte" de caractéristiques constituée de la valeur d'un neurone de sortie pour chaque position du réseau au sein de l'entrée. Les cartes produites sont alors sous-échantillonnées afin de réduire la dimensionalité. On utilise pour cela souvent la fonction maximum qui consiste simplement à retenir la valeur maximale obtenue sur la zone couverte (ici de taille* 2×2*). Les cartes de sortie obtenues peuvent à leur tour être traitées par une nouvelle couche de convolution, ou, si leur taille est raisonnable, par une couche classique densément connectée (toutes les valeurs de sorties sont alors regroupées sous la forme d'un seul vecteur).*

même neurone de sortie sont fusionnées localement en une seule valeur. Une technique usuelle consiste par exemple à retenir la valeur maximale sur un carré de 2x2 sorties, de manière à diviser par 4 la taille globale de la représentation. Après application de plusieurs réseaux à convolution successifs, la sortie produite a une taille suffisamment faible pour pouvoir être utilisée par un réseau standard densément connecté (voir figure 7.1).

Ce type de réseau permet d'obtenir des résultats à l'état de l'art dans les applications de reconnaissance ou d'annotation d'images par exemple (Kiros et al. 2014 ; Donahue et al. 2014 ; Vinyals et al. 2014 ; Krizhevsky et al. 2012). Ils sont cependant coûteux en calculs et reflètent de manière très imparfaite la manière dont est traité le flux visuel chez l'homme. De plus, en augmentant le champ récepteur de chaque neurone au fur et à mesure de la hiérarchie, les stimuli reçus sont susceptibles de mélanger des informations diverses : il n'y a par exemple rarement qu'un seul objet dans le champ de vision. Faire la part entre ces différents stimuli de manière non-supervisée nous semble être un défi difficile. De même, appliquer directement un réseau du type de celui que nous avons présenté au chapitre 4 nous semble voué à l'échec, dès lors que plusieurs catégories de stimuli sont présentes simultanément en entrée. En effet, il devient alors impossible de représenter l'entrée sensorielle à l'aide d'un seul symbole comme nous l'avons fait dans nos travaux.

Chez l'homme, seule une très faible partie du champ visuel, au niveau de la fovéa, est perçue avec précision. Face à une scène, nos yeux la parcourent donc grâce aux mouvements saccadiques. Ceux-ci sont loin de ressembler au parcours d'une image tel qu'il est fait lors d'une convolution, c'est-à-dire de manière linéaire et exhaustive. En particulier, nous ne parcourons pas de la même manière un visage ou un paysage. Les capacités pré-

dictives du cerveau semblent ici jouer un rôle important : chaque saccade peut être vue comme un sondage de l'environnement permettant ou non de confirmer une prédiction, rendue possible grâce au modèle de l'environnement déjà construit grâce aux saccades précédentes (Volpi et al. 2014). Il y a donc une boucle entre perception de l'environnement et saccades oculaires : une première fixation permet d'envisager plusieurs causes possibles au stimulus perçu, notamment en terme d'objet présent. Ces causes possibles permettent de prédire pour chacune d'elles les conséquences d'une nouvelle saccade, ce qui permet de confirmer ou d'infirmer le modèle de l'environnement perçu en effectuant cette saccade. Dans ce cadre, on peut proposer un modèle d'attention comme phénomène qui consiste à se focaliser sur un élément particulier du modèle pour l'affiner : les saccades sont alors dirigées vers les parties de l'environnement liées à cet élément du modèle. De plus, les nouvelles perceptions sont alors prioritairement interprétées à l'aune du modèle sur lequel est portée notre attention. On trouve ici un parallèle possible avec le postulat de la Gestalt "le tout est différent de la somme des parties" : chaque partie influence l'ordre et la manière de percevoir les autres, aboutissant à une perception globale différente. Un lien, qu'il faudrait approfondir, se dessine également avec les travaux portant sur l'apprentissage budgété (Dulac-Arnold et al. 2014). En outre, dans ce cadre, les points saillants correspondent aux éléments non (ou mal) prédits, qui attirent donc notre attention afin d'affiner notre modèle de l'environnement. Ils sont alors de deux types : des changements brusques de l'environnement (comme un mouvement ou un flash lumineux), ou alors des points d'intérêt permettant d'affiner notre modèle (c'est-à-dire de différencier deux stimuli proches). Dans le cas des visages par exemple, les yeux, le nez et la bouche sont des éléments importants pour pouvoir discriminer deux personnes différentes, et il n'est donc pas étonnant qu'ils attirent notre regard.

Cette opposition convolution versus attention peut être étendue au cas des stimuli temporels. En effet, le traitement usuel du temps par un réseau récurrent peut être vu comme une convolution : on applique le même réseau à chaque pas de temps (ou fenêtre temporelle). Par analogie avec le modèle précédent, on peut penser que l'attention permet de faire des sauts d'instant clé en instant clé, ignorant les instants intermédiaires tant qu'ils n'entrent pas en conflit direct avec le modèle courant du monde. Prenons l'exemple de la perception de phrases que nous avons abordé au chapitre 5. En remplaçant certaines syllabes par des bruits blancs, la perception n'est pas altérée, alors qu'un silence est tout de suite détecté. On peut proposer le fait que les bruits blancs contenant les fréquences attendues, ceux-ci ne sont pas détectés par l'attention et passent plutôt inaperçus. En revanche, un silence correspond à un conflit fort avec les attentes de la perception : ceux-ci focalisent l'attention.

La combinaison de réseaux profonds et de modèle attentionnels commence à être explorée par plusieurs auteurs. Dans (Ranzato 2014), l'auteur entraîne un réseau à générer des saccades pour prédire la catégorie d'une image. La première saccade est prédite à partir d'une version de faible résolution de l'image globale, tandis que les suivantes sont calculées en utilisant l'information déjà obtenue grâce aux saccades précédentes. Un modèle similaire est développé dans (Mnih et al. 2014). À la différence du précédent, ce modèle intègre également la position de chaque saccade avec l'image perçue afin de construire une représentation du stimulus à classifier. La représentation sert alors à la fois à obtenir une

classification intermédiaire et à calculer la position de la prochaine saccade. Ce modèle est proche de celui proposé auparavant par (Larochelle et Hinton 2010). Cependant, ces approches fixent généralement *a priori* le nombre de saccades, et ne s'adaptent pas à chaque nouvelle situation en choisissant le nombre de saccades optimal. De plus, tous ces modèles ont été développés dans des cadres supervisés, où l'apprentissage de la politique optimale nécessite l'existence d'une tâche externe (typiquement de la classification) afin de définir un coût à minimiser. Le développement de telles approches dans un cadre non supervisé reste un problème ouvert.

Il serait intéressant d'utiliser l'architecture que nous avons présentée au chapitre 4 dans ce cadre, en étudiant la possibilité de pré-activer certains neurones de la couche *softmax* en fonction de ce qui a déjà été perçu et de la saccade qui va être effectuée. Ainsi, à titre d'exemple, reconnaître un nez pourrait pré-activer le fait de percevoir un œil en faisant une saccade vers le haut (à droite ou à gauche), biaisant par la suite l'interprétation de ce qui sera perçu. On peut imaginer que cette pré-activation découle d'une couche supérieure, qui, recevant comme information qu'un œil a été perçu, tend à reconnaître un visage et active donc une politique de saccades et de pré-activations pour les autres zones d'intérêt d'un visage (pouvant ainsi confirmer ou infirmer la présence d'un visage). Afin de pouvoir être apprise, cette couche supérieure devrait donc nécessairement recevoir en entrée la sortie de la couche inférieure, ainsi que les actions effectuées. Sa tâche pourrait alors consister à apprendre les conséquences des différentes actions en terme de modifications des sorties de la couche inférieure (ces prédictions servant par la suite à pré-activer les neurones correspondants). On peut remarquer que ces modifications pourraient être naturellement codées par des transformations orthogonales en utilisant l'architecture présentée au chapitre précédent : un œil se "transforme" en nez, en bouche ou en autre œil selon la saccade effectuée, produisant un transfert de l'activité de la couche *softmax* d'un neurone vers un autre. Ces trois couches (élément perçu, action effectuée, modification engendrée) utilisées comme entrées/sorties de cette couche supérieure permettent d'utiliser une architecture du même type que celle présentée au chapitre 4 (le concept de *visage* étant alors une représentation haut niveau permettant de prédire une des couches étant données les deux autres) et ne sont pas sans rappeler le triptyque objet/action/effet utilisé comme modèle computationnel des affordances (Montesano et al. 2008). La description que nous venons de faire de cette architecture est toutefois simpliste et ignore plusieurs obstacles à surmonter. Premièrement, l'architecture du chapitre 4 doit être étendue afin de pouvoir intégrer les informations de plusieurs actions successives en une unique représentation. Deuxièmement, elle passe sous silence l'émergence des "bonnes" actions, permettant d'apprendre quelque chose : une représentation haut niveau ne peut émerger que si les régularités sont suffisamment importantes et donc si les actions effectuées pendant l'apprentissage ne sont pas trop aléatoires. Un mécanisme de motivation intrinsèque, faisant le lien entre élément perçu et action à effectuer reposant sur l'erreur de prédiction semble ici nécessaire (Volpi et al. 2014). La possibilité de générer des saccades permettant de centrer la vue sur un élément recherché (possiblement imaginé) a également été étudiée par (Tang et al. 2014). Enfin, l'aspect de segmentation temporelle, problème difficile comme nous l'avons vu au chapitre précédent et nécessaire pour définir les représentations des actions et des effets, est également négligé.

7.2.2 Gestion de l'incertitude : autoencodeur versus RBM

Poussons encore le raisonnement du paragraphe précédent et imaginons qu'une couche supérieure représente la perception à un très haut niveau conceptuel, par exemple "magasin". Visitant un magasin, il est illusoire de penser que l'on pourrait obtenir, comme au chapitre 4, une paramétrisation du lieu en question permettant de reconstruire précisément, après propagation dans les couches inférieures, le flux visuel correspondant. Les expériences sur la cécité au changement décrites au chapitre 2 montrent d'ailleurs que ce n'est pas ce que semble faire le cerveau. Au contraire, on peut imaginer coder plutôt une distribution de probabilité représentant la probabilité de trouver tel ou tel objet à tel ou tel endroit (les stimuli avec une probabilité très faible pourraient alors retenir l'attention).

De ce point de vue, il peut sembler opportun d'utiliser des algorithmes intrinsèquement probabilistes, comme les machines de Boltzmann restreintes, par comparaison avec les autoencodeurs dont le cadre général est déterministe.

Nous avons développé les architectures présentées dans les chapitres 4 et 5 dans le cadre théorique des autoencodeurs. Ceci est un choix arbitraire : nous aurions tout aussi bien pu nous reposer sur des machines de Boltzmann restreintes, que nous avons d'ailleurs utilisées au chapitre 6. Nous y avons certainement gagné une plus grande facilité de paramétrage des algorithmes et de spécifications des règles d'apprentissage, les algorithmes de type autoencodeur ayant notamment moins de méta-paramètres que les algorithme de type RBM. L'utilisation de RBMs plutôt que d'autoencodeurs pour traiter le problème de l'apprentissage par renforcement et de la curiosité au chapitre 6 découle par ailleurs du cadre probabiliste de ces problématiques lié à la nécessité de l'exploration.

Est-ce à dire qu'il faudrait transposer les architectures que nous avons proposées dans le cadre théorique des RBMs, et cela remettrait-il en question les résultats présentés dans cette thèse ? Pas nécessairement. Même s'il est difficile d'apporter une réponse définitive à ces questions, comme nous l'avons déjà évoqué, plusieurs travaux tendent à suggérer qu'autoencodeurs et RBMs sont en réalité deux facettes d'un même mécanisme sous-jacent et apprennent en réalité la même chose (Vincent 2011 ; Kamyshanska et Memisevic 2013). Aussi peut-on imaginer conserver les architectures et les techniques d'apprentissage que nous avons présentées tout au long de cette thèse, et faire des modifications à la marge notamment lors de l'exploitation des propriétés génératives de ces réseaux, en y introduisant un aspect stochastique plus proche du cadre des RBMs. La nécessité de transposition des architectures présentées dans un cadre théorique plus général, qui reste encore à définir, n'est toutefois pas exclue.

Chapitre 8

Conclusion

That the automobile has practically reached the limit of its development is suggested by the fact that during the past year no improvements of a radical nature have been introduced.

Scientific American, 2 janvier 1909

Nous avons présenté au cours de ce manuscrit plusieurs travaux montrant les atouts des réseaux de neurones pour la robotique développementale, plus particulièrement en ce qui concerne la problématique de l'apprentissage autonome de représentations à partir de flux sensorimoteurs bruts. À partir d'une définition des symboles en tant que sous-variétés dans des espaces sensoriels et fonctionnels, nous développons des architectures utilisant des réseaux de neurones non-supervisés afin de structurer l'espace sensorimoteur d'un robot. Nous montrons pour cela l'intérêt d'exploiter la nature multimodale des signaux et suggérons l'utilisation de capacités prédictives afin de permettre l'émergence de représentations symboliques de séquences temporelles.

Nous avons présenté l'hypothèse des sous-variétés au chapitre 2. Celle-ci est précédée d'une vue d'ensemble de quelques phénomènes liés à la perception et à l'action chez l'humain qui ont été une source d'inspiration importante pour nos travaux. Nous pensons qu'ils constituent en outre des éléments de réflexion intéressants pour étudier la manière dont la nature a résolu le problème de l'intelligence tel qu'il se pose à la robotique développementale.

Nous avons voulu au chapitre 3 offrir une vision didactique de l'apprentissage profond. Nous nous sommes cependant souvent restreints à l'aspect non-supervisé de ces réseaux, sans rentrer dans les détails de toutes les variantes régulièrement proposées pour résoudre des problèmes d'apprentissage automatique toujours plus variés. Ce champ de recherche est actuellement très dynamique, ce qui rend très difficile de rendre compte d'un état de l'art exhaustif et actualisé.

Nous avons montré au chapitre 4 comment la définition de la perception et de l'action sous la forme de sous-variétés dans un flux sensorimoteur pouvait être implémentée dans un réseau non-supervisé afin de faire effectivement émerger une représentation symbolique à partir de stimuli bruts. Nous montrons notamment comment la multimodalité peut être exploitée par l'apprentissage de ces représentations. Les *a priori* sous-jacents y sont ce-

pendant encore importants. Notre architecture fait par exemple l'hypothèse que toutes les modalités utilisées en entrée sont corrélées, avec en plus en découpage pertinent des stimuli temporels. En ce sens, on peut adresser à l'architecture présentée les mêmes critiques que celles que nous avons portées à l'encontre de beaucoup de travaux sur les affordances utilisant notamment un prédécoupage des objets dans les images et l'identification des instants initiaux et finaux pour calculer les effets des actions.

Conscient de ce problème, nous avons développé le problème de la temporalité au chapitre 5. Si nous n'avons pas su proposer d'architecture permettant un découpage non supervisé d'un flux temporel, préalable entre autres à l'émergence du langage, nous avons toutefois montré l'intérêt des représentations sous forme de sous-variétés pour la génération d'actions. En utilisant ce type de représentations dans des réseaux neuronaux entraînés sur le principe des autoencodeurs, nous avons montré que notre approche restait un candidat valable pour résoudre le problème de la temporalité d'un point de vue développemental. En effet, la capacité de générer de telles séquences est un caractère préalable à la capacité de prédiction au cœur des théories de codage prédictif (Rao et Ballard 1999 ; Friston 2010 ; Clark 2013). Le second travail présenté dans ce chapitre, portant sur l'apprentissage de représentations de transformations orthogonales, a un caractère plus spéculatif pour le traitement de la temporalité. Par rapport à d'autres approches, il a l'avantage de mettre en avant la notion de prédictabilité des transitions entre états. En effet, en faisant l'hypothèse que la quantité d'information entre deux états successifs ne change pas, ceux-ci deviennent entièrement prédictibles. Un tel mécanisme pourrait donc se révéler utile pour biaiser l'apprentissage de représentations minimisant la surprise (Friston 2010).

Dans le chapitre 6, nous avons voulu avancer quelques hypothèses sur la manière dont l'apprentissage profond pourrait être utilisé pour plusieurs problématiques liées à la robotique développementale. Comme nous l'avons présenté dans l'introduction, beaucoup de problématiques supplémentaires existent et ont été largement abordées par d'autres travaux. Il serait donc possible d'étudier le comportement des différents algorithmes de la littérature à partir des représentations apprises par les approches que nous avons proposées. Nous pensons cependant que la faculté d'utiliser un unique cadre théorique se traduisant par une même règle d'apprentissage à tous les niveaux de l'architecture facilite grandement les possibilités d'interaction entre les différents modules, sans nécessiter l'introduction de mécanismes plus ou moins *ad-hoc*, notamment pour modéliser les interactions descendantes.

Enfin, dans le dernier chapitre, nous avons discuté quelques limitations de nos travaux et esquissé quelques idées pour parvenir à appliquer les techniques proposées dans des conditions naturelles. Il faudra en particulier réussir à supporter l'abondance de stimuli divers, complexes et superposés, ainsi qu'à gérer l'incertitude intrinsèque des environnements ouverts.

Bibliographie

Ackley, David H, Geoffrey E Hinton et Terrence J Sejnowski (1985). « A learning algorithm for boltzmann machines ». Dans : *Cognitive science* 9.1, p. 147–169 (cf. p. 26, 70).

Akgun, Baris, Nilgun Dag, Tahir Bilal, Ilkay Atil et Erol Sahin (2009). « Unsupervised learning of affordance relations on a humanoid robot ». Dans : *Computer and Information Sciences, 2009. ISCIS 2009. 24th International Symposium on*. IEEE, p. 254–259 (cf. p. 22).

Allan, Lorraine G (1979). « The perception of time ». Dans : *Perception & Psychophysics* 26.5, p. 340–354 (cf. p. 123).

Arons, Barry (1992). « A review of the cocktail party effect ». Dans : *Journal of the American Voice I/O Society* 12.7, p. 35–50 (cf. p. 116).

Asada, Minoru, Karl F MacDorman, Hiroshi Ishiguro et Yasuo Kuniyoshi (2001). « Cognitive developmental robotics as a new paradigm for the design of humanoid robots ». Dans : *Robotics and Autonomous Systems* 37.2, p. 185–193 (cf. p. 16).

Avillac, Marie, Sophie Deneve, Etienne Olivier, Alexandre Pouget et Jean-René Duhamel (2005). « Reference frames for representing visual and tactile locations in parietal cortex ». Dans : *Nature neuroscience* 8.7, p. 941–949 (cf. p. 43).

Babinec, S et Jiří Pospíchal (2005). « Two approaches to optimize echo state neural networks ». Dans : *Proc. of the 11th Int. Conf. on Soft Computing, Mendel*, p. 39–44 (cf. p. 121).

Bação, Fernando, Victor Lobo et Marco Painho (2005). « Self-organizing maps as substitutes for k-means clustering ». Dans : *Computational Science–ICCS 2005*. Springer, p. 476–483 (cf. p. 88).

Baldi, P. (2012). « Autoencoders, Unsupervised Learning, and Deep Architectures. » Dans : *Proceedings of ICML Workshop on Unsupervised and Transfer Learning* (cf. p. 129).

Baldi, P., S. Forouzan et Z. Lu (2012). « Complex-Valued Autoencoders ». Dans : *Neural Networks* 33, p. 136–147 (cf. p. 129).

Baldi, Pierre et Peter Sadowski (2014). « The dropout learning algorithm ». Dans : *Artificial intelligence* 210, p. 78–122 (cf. p. 78).

Baranes, Adrien et P-Y Oudeyer (2011). « The interaction of maturational constraints and intrinsic motivations in active motor development ». Dans : *Development and Learning (ICDL), 2011 IEEE International Conference on*. T. 2. IEEE, p. 1–8 (cf. p. 157).

Baranes, Adrien et Pierre-Yves Oudeyer (2013). « Active learning of inverse models with intrinsically motivated goal exploration in robots ». Dans : *Robotics and Autonomous Systems* 61.1, p. 49–73 (cf. p. 24, 166).

Barsalou, Lawrence W. (août 1999). « Perceptual symbol systems ». Dans : *Behavioral and Brain Sciences* 22 (04), p. 577–660. issn : 1469-1825 (cf. p. 85, 86, 110).

Bellman, Richard (1957). *Dynamic Programming*. 1re éd. Princeton, NJ, USA : Princeton University Press (cf. p. 47).

Bengio, Y, P Simard et P Frasconi (1994). « Learning long-term dependencies with gradient descent is difficult ». Dans : *Neural Networks* 5.2, p. 157–166 (cf. p. 73).

Bengio, Yoshua et Eric Thibodeau-Laufer (2013). « Deep generative stochastic networks trainable by backprop ». Dans : *arXiv preprint arXiv :1306.1091* (cf. p. 83, 168).

Bengio, Yoshua, Jérôme Louradour, Ronan Collobert et Jason Weston (2009). « Curriculum learning ». Dans : *Proceedings of the 26th annual international conference on machine learning*. ACM, p. 41–48 (cf. p. 157).

Bengio, Yoshua, Aaron Courville et Pascal Vincent (2013). « Representation learning : A review and new perspectives ». Dans : *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.8, p. 1798–1828 (cf. p. 76, 87).

Bergstra, James, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley et Yoshua Bengio (juin 2010). « Theano : a CPU and GPU Math Expression Compiler ». Dans : *Proceedings of the Python for Scientific Computing Conference (SciPy)*. Austin, TX (cf. p. 129).

Berlyne, Daniel E (1960). *Conflict, arousal, and curiosity*. Macgraw-Hill (cf. p. 24, 155).

Beyer, Kevin, Jonathan Goldstein, Raghu Ramakrishnan et Uri Shaft (1999). « When is "nearest neighbor" meaningful ? » Dans : *Database Theory—ICDT'99*. Springer, p. 217–235 (cf. p. 47).

Botvinick, M et J Cohen (fév. 1998). « Rubber hands 'feel' touch that eyes see. » Dans : *Nature* 391.6669, p. 756. issn : 0028-0836 (cf. p. 35).

Braitenberg, Valentino (1984). *Vehicles : Experiments in synthetic psychology*. MIT press (cf. p. 15, 16).

Brooks, Rodney A (1986). « A robust layered control system for a mobile robot ». Dans : *Robotics and Automation, IEEE Journal of* 2.1, p. 14–23 (cf. p. 38).

— (1991). « Intelligence without representation ». Dans : *Artificial intelligence* 47.1, p. 139–159 (cf. p. 16).

Buckingham, Gavin et Melvyn A Goodale (2010). « Lifting without seeing : the role of vision in perceiving and acting upon the size weight illusion ». Dans : *PLoS One* 5.3, e9709 (cf. p. 41).

Bullier, J (sept. 2001). « Feedback connections and conscious vision ». Dans : *Trends Cogn Sci* 5.9, p. 369–370 (cf. p. 36).

Burkitt, Anthony N (2006). « A review of the integrate-and-fire neuron model : I. Homogeneous synaptic input ». Dans : *Biological cybernetics* 95.1, p. 1–19 (cf. p. 59).

Calandra, Roberto, Tapani Raiko, Marc Peter Deisenroth et Federico Montesino Pouzols (2012). « Learning deep belief networks from non-stationary streams ».

Dans : *Artificial Neural Networks and Machine Learning–ICANN 2012*. Springer, p. 379–386 (cf. p. 168).

Cangelosi, Angelo et Stevan Harnad (2001). « The adaptive advantage of symbolic theft over sensorimotor toil : Grounding language in perceptual categories ». Dans : *Evolution of communication* 4.1, p. 117–142 (cf. p. 23).

Cangelosi, Angelo et Thomas Riga (2006). « An embodied model for sensorimotor grounding and grounding transfer : Experiments with epigenetic robots ». Dans : *Cognitive science* 30.4, p. 673–689 (cf. p. 23).

Cayton, Lawrence (2005). *Algorithms for manifold learning*. Rap. tech. University of California, San Diego (cf. p. 51, 87).

Ciresan, Dan Claudiu, Ueli Meier, Luca Maria Gambardella et Jürgen Schmidhuber (2010). « Deep, big, simple neural nets for handwritten digit recognition ». Dans : *Neural computation* 22.12, p. 3207–3220 (cf. p. 74).

Clark, Andy (2013). « Whatever next ? Predictive brains, situated agents, and the future of cognitive science ». Dans : *Behavioral and Brain Sciences* 36.03, p. 181–204 (cf. p. 25, 26, 176).

Coates, Adam, Brody Huval, Tao Wang, David Wu, Bryan Catanzaro et Ng Andrew (2013). « Deep learning with cots hpc systems ». Dans : *Proceedings of The 30th International Conference on Machine Learning*, p. 1337–1345 (cf. p. 169).

Contardo, Gabriella, Ludovic Denoyer, Thierry Artieres et Patrick Gallinari (2014). « Learning States Representations in POMDP ». Dans : *Internation Conference on Learning Representations (poster) ICLR 2014* (cf. p. 120, 122).

Cos-Aguilera, Ignasi, G Hayes et Lola Cañamero (2004). « Using a SOFM to learn object affordances ». Dans : *Procs 5th Workshop of Physical Agents (WAF'04)*. University of Edinburgh (cf. p. 22).

Csikszentmihalyi, Mihaly (1991). *Flow : The psychology of optimal experience*. T. 41. HarperPerennial New York (cf. p. 24, 155).

Cutting, James E et Lynn T Kozlowski (1977). « Recognizing friends by their walk : Gait perception without familiarity cues ». Dans : *Bulletin of the psychonomic society* 9.5, p. 353–356 (cf. p. 118).

Damasio, Antonio R (1989a). « The brain binds entities and events by multiregional activation from convergence zones ». Dans : *Neural Computation* 1.1, p. 123–132 (cf. p. 35).

— (1989b). « Time-locked multiregional retroactivation : A systems-level proposal for the neural substrates of recall and recognition ». Dans : *Cognition* 33.1, p. 25–62 (cf. p. 35).

— (1990). « Category-related recognition defects as a clue to the neural substrates of knowledge ». Dans : *Trends in neurosciences* 13.3, p. 95–98 (cf. p. 35).

Daniel, Christian, Gerhard Neumann et Jan Peters (2012a). « Hierarchical relative entropy policy search ». Dans : *Int. Conf. on Artificial Intelligence and Statistics* (cf. p. 139).

Daniel, Christian, Gerhard Neumann, Oliver Kroemer et Jan Peters (2012b). « Learning Sequential Motor Tasks ». Dans : *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)* (cf. p. 139).

Dauphin, Yann, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli et Yoshua Bengio (2014). « Identifying and attacking the saddle point problem in high-dimensional non-convex optimization ». Dans : *arXiv preprint arXiv :1406.2572* (cf. p. 64, 75, 169).

Davis, Christopher M et William Roberts (1976). « Lifting movements in the size-weight illusion ». Dans : *Perception & Psychophysics* 20.1, p. 33–36 (cf. p. 41).

Daw, Nathaniel D et Peter Dayan (2014). « The algorithmic anatomy of model-based evaluation ». Dans : *Philosophical Transactions of the Royal Society B : Biological Sciences* 369.1655, p. 20130478 (cf. p. 152, 155).

Dayan, Peter et Yael Niv (2008). « Reinforcement learning : the good, the bad and the ugly ». Dans : *Current opinion in neurobiology* 18.2, p. 185–196 (cf. p. 152).

Dayan, Peter, Geoffrey E Hinton, Radford M Neal et Richard S Zemel (1995). « The helmholtz machine ». Dans : *Neural computation* 7.5, p. 889–904 (cf. p. 26, 45).

De Sa, Virginia R et Dana H Ballard (1997). « Perceptual learning from cross-modal feedback ». Dans : *Psychology of learning and motivation* 36, p. 309–351 (cf. p. 90, 111, 112).

— (1998). « Category learning through multimodality sensing ». Dans : *Neural Computation* 10.5, p. 1097–1117 (cf. p. 90, 110).

Dean, Jeffrey, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le et al. (2012). « Large scale distributed deep networks ». Dans : *Advances in Neural Information Processing Systems*, p. 1223–1231 (cf. p. 169).

Denoyer, L. et P. Gallinari (oct. 2014). « Deep Sequential Neural Network ». Dans : *ArXiv e-prints* (cf. p. 110).

Desimone, Robert, THOMAS D Albright, Charles G Gross et Charles Bruce (1984). « Stimulus-selective properties of inferior temporal neurons in the macaque ». Dans : *The Journal of Neuroscience* 4.8, p. 2051–2062 (cf. p. 33).

Dittrich, Winand H (1993). « Action categories and the perception of biological motion ». Dans : *PERCEPTION-LONDON-* 22, p. 15–15 (cf. p. 118).

Donahue, Jeff, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko et Trevor Darrell (2014). « Long-term Recurrent Convolutional Networks for Visual Recognition and Description ». Dans : *arXiv preprint arXiv :1411.4389* (cf. p. 170).

Droniou, Alain et Olivier Sigaud (2013). « Gated Autoencoders with Tied Input Weights ». Dans : *Proceedings of International Conference on Machine Learning*, p. 154–162 (cf. p. 27, 126).

Droniou, Alain, Serena Ivaldi, Vincent Padois et Olivier Sigaud (oct. 2012a). « Autonomous Online Learning of Velocity Kinematics on the iCub : a Comparative Study ». Dans : *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS*. Vilamoura, Portugal, p. 3577–3582 (cf. p. 27).

Droniou, Alain, Serena Ivaldi et Olivier Sigaud (2012b). « Comparaison expérimentale d'algorithmes de régression pour l'apprentissage de modèles cinématiques du robot humanoïde iCub ». Dans : *Conférence Francophone sur l'Apprentissage Automatique (Cap)*, p. 95–110 (cf. p. 27).

Droniou, Alain, Serena Ivaldi, Patrick Stalph, Martin Butz et Olivier Sigaud (2012c). « Learning Velocity Kinematics : Experimental Comparison of On-line Regression Algorithms ». Dans : *Proceedings Robotica*, p. 15–20 (cf. p. 27).

Droniou, Alain, Serena Ivaldi et Olivier Sigaud (2014). « Learning a Repertoire of Actions with Deep Neural Networks ». Dans : *Proceedings of ICDL-EpiRob*. Italie (cf. p. 27, 139).

— (2014, in press). « Deep unsupervised network for multimodal perception, representation and classification ». Dans : *Robotics and Autonomous Systems* (cf. p. 27, 86).

Dulac-Arnold, Gabriel, Ludovic Denoyer, Nicolas Thome, Matthieu Cord et Patrick Gallinari (2014). « Sequentially Generated Instance-Dependent Image Representations for Classification ». Dans : *Internation Conference on Learning Representations - ICLR 2014* (cf. p. 171).

Dutech, Alain (2012). « Self-organizing developmental reinforcement learning ». Dans : *From Animals to Animats 12*. Springer, p. 310–319 (cf. p. 150).

Eggert, Julian et J Leo van Hemmen (2001). « Modeling neuronal assemblies : theory and implementation ». Dans : *Neural Computation* 13.9, p. 1923–1974 (cf. p. 59).

Elman, Jeffrey L (1990). « Finding structure in time ». Dans : *Cognitive science* 14.2, p. 179–211 (cf. p. 67).

Erhan, Dumitru, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent et Samy Bengio (2010). « Why does unsupervised pre-training help deep learning ? » Dans : *The Journal of Machine Learning Research* 11, p. 625–660 (cf. p. 74).

Falchier, Arnaud, Simon Clavagnier, Pascal Barone et Henry Kennedy (juil. 2002). « Anatomical evidence of multimodal integration in primate striate cortex. » Dans : *The Journal of neuroscience : the official journal of the Society for Neuroscience* 22.13, p. 5749–59. issn : 1529-2401 (cf. p. 35).

Farabet, Clément, Yann LeCun, Koray Kavukcuoglu, Eugenio Culurciello, Berin Martini, Polina Akselrod et Selcuk Talay (2011). « Large-scale FPGA-based convolutional networks ». Dans : *Machine Learning on Very Large Data Sets* (cf. p. 169).

Feinberg, Todd E, Rachel J Schindler, Natalie Gilson Flanagan et Laurence D Haber (1992). « Two alien hand syndromes ». Dans : *Neurology* 42.1, p. 19–19 (cf. p. 42).

Flanagan, J Randall et Michael A Beltzner (2000). « Independence of perceptual and sensorimotor predictions in the size–weight illusion ». Dans : *Nature neuroscience* 3.7, p. 737–741 (cf. p. 41).

Flanagan, J Randall, Jennifer P Bittner et Roland S Johansson (2008). « Experience can change distinct size-weight priors engaged in lifting objects and judging their weights ». Dans : *Current Biology* 18.22, p. 1742–1747 (cf. p. 41).

Ford, Paul, Nicola J Hodges et A Mark Williams (2005). « Online attentional-focus manipulations in a soccer-dribbling task : Implications for the proceduralization of motor skills ». Dans : *Journal of Motor Behavior* 37.5, p. 386–394 (cf. p. 117).

Fort, Alexandra, Claude Delpuech, Jacques Pernier et Marie Hélène Giard (juin 2002). « Early auditory-visual interactions in human cortex during nonredundant tar-

get identification. » Dans : *Brain research. Cognitive brain research* 14.1, p. 20–30. ISSN : 0926-6410 (cf. p. 35).

FRAISSE, Paul (1966). « Visual perceptive simultaneity and masking of letters successively presented ». Dans : *Perception & Psychophysics* 1.9, p. 285–287 (cf. p. 119).

FREIRE, Alejo, Terri L LEWIS, Daphne MAURER et Randolph BLAKE (2006). « The development of sensitivity to biological motion in noise ». Dans : *Perception* 35.5, p. 647 (cf. p. 118).

FRISTON, Karl (2010). « The free-energy principle : a unified brain theory ? » Dans : *Nature Reviews Neuroscience* 11.2, p. 127–138 (cf. p. 43, 45, 46, 176).

FRISTON, Karl, Jérémie MATTOUT, Nelson TRUJILLO-BARRETO, John ASHBURNER et Will PENNY (2007). « Variational free energy and the Laplace approximation ». Dans : *NeuroImage* 34.1, p. 220–234 (cf. p. 25).

GARDENFORS, Peter (2004). « Conceptual spaces as a framework for knowledge representation ». Dans : *Mind and Matter* 2.2, p. 9–27 (cf. p. 51, 85).

GHADAKPOUR, Laleh (2002). « Le système conceptuel, à l'interface entre le langage, le raisonnement et l'espace qualitatif : vers un modèle de représentations éphémères ». Thèse de doct. Ecole Polytechnique (cf. p. 85).

GIARD, M. H. et F. PERONNET (sept. 1999). « Auditory-Visual Integration During Multimodal Object Recognition in Humans : A Behavioral and Electrophysiological Study ». Dans : *J. Cognitive Neuroscience* 11.5, p. 473–490. ISSN : 0898-929X (cf. p. 111).

GIBSON, J J (1977). « The Theory of Affordances ». Dans : *Perceiving, Acting, and Knowing*. Robert Shaw et John Bransford (cf. p. 15, 22).

GISSLÉN, Linus, Matt LUCIW, Vincent GRAZIANO et Jürgen SCHMIDHUBER (2011). « Sequential constant size compressors for reinforcement learning ». Dans : *Artificial General Intelligence*. Springer, p. 31–40 (cf. p. 122, 125).

GLOROT, Xavier et Yoshua BENGIO (2010). « Understanding the difficulty of training deep feedforward neural networks ». Dans : *International Conference on Artificial Intelligence and Statistics (AISTATS'10)*. T. 9, p. 249–256 (cf. p. 73).

GLOROT, Xavier, Antoine BORDES et Yoshua BENGIO (2011). « Deep sparse rectifier networks ». Dans : *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. JMLR W&CP Volume*. T. 15, p. 315–323 (cf. p. 59).

GOGATE, Lakshmi J, Arlene S WALKER-ANDREWS et Loraine E BAHRICK (2001). « The intersensory origins of word-comprehension : an ecological–dynamic systems view ». Dans : *Developmental Science* 4.1, p. 1–18 (cf. p. 23).

GOKHALE, Vinayak, Jonghoon JIN, Aysegul DUNDAR, Berin MARTINI et Eugenio CULURCIELLO (2014). « A 240 g-ops/s mobile coprocessor for deep neural networks ». Dans : *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*. IEEE, p. 696–701 (cf. p. 169).

GOLDBERG, Michael E, James W BISLEY, Keith D POWELL et Jacqueline GOTTLIEB (2006). « Saccades, salience and attention : the role of the lateral intraparietal area in visual behavior ». Dans : *Progress in brain research* 155, p. 157–175 (cf. p. 43).

GOLDSTONE, Robert L et Andrew T HENDRICKSON (2010). « Categorical perception ». Dans : *Wiley Interdisciplinary Reviews : Cognitive Science* 1.1, p. 69–78 (cf. p. 112).

Goodfellow, Ian, David Warde-Farley, Mehdi Mirza, Aaron Courville et Yoshua Bengio (2013). « Maxout Networks ». Dans : *Proceedings of The 30th International Conference on Machine Learning*, p. 1319–1327 (cf. p. 59).

Grafton, ST (2003). « Apraxia : a disorder of motor control ». Dans : *neurological foundations of cognitive neuroscience (D'Esposito M, ed.)* P. 239–258 (cf. p. 43).

Graves, Alex et Navdeep Jaitly (2014). « Towards end-to-end speech recognition with recurrent neural networks ». Dans : *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, p. 1764–1772 (cf. p. 120).

Graves, Alex et Jürgen Schmidhuber (2005). « Framewise phoneme classification with bidirectional LSTM and other neural network architectures ». Dans : *Neural Networks* 18.5, p. 602–610 (cf. p. 122).

Graves, Alex, Douglas Eck, Nicole Beringer et Juergen Schmidhuber (2004). « Biologically plausible speech recognition with LSTM neural nets ». Dans : *Biologically Inspired Approaches to Advanced Information Technology*. Springer, p. 127–136 (cf. p. 122).

Griffith, Shane, Jivko Sinapov, Vladimir Sukhoy et Alexander Stoytchev (2012). « A behavior-grounded approach to forming object categories : Separating containers from noncontainers ». Dans : *Autonomous Mental Development, IEEE Transactions on* 4.1, p. 54–69 (cf. p. 119).

Gross, CG, DB Bender et CE Rocha-Miranda (1969). « Visual receptive fields of neurons in inferotemporal cortex of the monkey ». Dans : *Science* 166.3910, p. 1303–1306 (cf. p. 33).

Gross, Charles G (2002). « Genealogy of the "grandmother cell" ». Dans : *The Neuroscientist* 8.5, p. 512–518 (cf. p. 33).

Grossberg, Stephen et Nestor A Schmajuk (1989). « Neural dynamics of adaptive timing and temporal discrimination during associative learning ». Dans : *Neural Networks* 2.2, p. 79–102 (cf. p. 123).

Harnad, Stevan (1990). « The symbol grounding problem ». Dans : *Physica D : Nonlinear Phenomena* 42.1, p. 335–346 (cf. p. 13, 18, 20).

Hastad, Johan (1986). « Almost optimal lower bounds for small depth circuits ». Dans : *Proceedings of the eighteenth annual ACM symposium on Theory of computing*. ACM, p. 6–20 (cf. p. 73).

Hebb, Donald Olding (1949). *The organization of behavior : A neuropsychological approach*. John Wiley & Sons (cf. p. 56).

Held, Richard et Alan Hein (1963). « Movement-produced stimulation in the development of visually guided behavior. » Dans : *Journal of comparative and physiological psychology* 56.5, p. 872 (cf. p. 38, 39).

Henriksson, Linda, Juha Karvonen, Niina Salminen-Vaparanta, Henry Railo et Simo Vanni (2012). « Retinotopic maps, spatial tuning, and locations of human visual areas in surface coordinates characterized with multifocal and blocked fMRI designs ». Dans : *PloS one* 7.5, e36859 (cf. p. 33).

Hinton, G E et R R Salakhutdinov (2006). « Reducing the Dimensionality of Data with Neural Networks ». Dans : *Science* 313.5786, p. 504–507. issn : 1095-9203 (cf. p. 73, 76, 82, 88).

Hinton, Geoffrey, Simon Osindero et Yee-Whye Teh (2006). « A fast learning algorithm for deep belief nets ». Dans : *Neural computation* 18.7, p. 1527–1554 (cf. p. 76, 82).

Hinton, Geoffrey E (2002). « Training products of experts by minimizing contrastive divergence ». Dans : *Neural Comput.* 14.8, p. 1771–1800. issn : 0899-7667 (cf. p. 81).

— (2012). « A practical guide to training restricted boltzmann machines ». Dans : *Neural Networks : Tricks of the Trade*. Springer, p. 599–619 (cf. p. 80, 81).

Hinton, Geoffrey E, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever et Ruslan R Salakhutdinov (2012). « Improving neural networks by preventing co-adaptation of feature detectors ». Dans : *arXiv preprint arXiv :1207.0580* (cf. p. 77, 78).

Hochreiter, Sepp et Jürgen Schmidhuber (1997a). « Long short-term memory ». Dans : *Neural computation* 9.8, p. 1735–1780 (cf. p. 65, 121).

— (1997b). « LSTM can Solve Hard Long Time Lag Problems ». Dans : *Advances in Neural Information Processing Systems*, p. 473–479 (cf. p. 122).

Hodgkin, Alan L et Andrew F Huxley (1952). « A quantitative description of membrane current and its application to conduction and excitation in nerve ». Dans : *The Journal of physiology* 117.4, p. 500 (cf. p. 59).

Hopfield, John J (1982). « Neural networks and physical systems with emergent collective computational abilities ». Dans : *Proceedings of the national academy of sciences* 79.8, p. 2554–2558 (cf. p. 70).

Hornik, K, M Stinchcombe et H White (1989). « Multilayer feedforward networks are universal approximators ». Dans : *Neural networks* 2.5, p. 359–366 (cf. p. 71).

Hotelling, Harold (1933). « Analysis of a complex of statistical variables into principal components. » Dans : *Journal of educational psychology* 24.6, p. 417 (cf. p. 87).

Huang, Guang-Bin, Qin-Yu Zhu et Chee-Kheong Siew (2006). « Extreme learning machine : theory and applications ». Dans : *Neurocomputing* 70.1, p. 489–501 (cf. p. 70).

Hubel, David H et Torsten N Wiesel (1959). « Receptive fields of single neurones in the cat's striate cortex ». Dans : *The Journal of physiology* 148.3, p. 574 (cf. p. 31).

Hubert, Lawrence et Phipps Arabie (1985). « Comparing partitions ». Dans : *Journal of classification* 2.1, p. 193–218 (cf. p. 96).

Huys, Quentin JM, Neir Eshel, Elizabeth O'Nions, Luke Sheridan, Peter Dayan et Jonathan P Roiser (2012). « Bonsai trees in your head : how the Pavlovian system sculpts goal-directed choices by pruning decision trees ». Dans : *PLoS computational biology* 8.3, e1002410 (cf. p. 152).

Ijspeert, Auke Jan, Jun Nakanishi, Heiko Hoffmann, Peter Pastor et Stefan Schaal (2013). « Dynamical movement primitives : learning attractor models for motor behaviors ». Dans : *Neural computation* 25.2, p. 328–373 (cf. p. 139).

Ingrand, Félix et Malik Ghallab (2014). « Deliberation for autonomous robots : A survey ». Dans : *Artificial Intelligence* (cf. p. 18).

Ivaldi, S., M. Fumagalli, M. Randazzo, F. Nori, G. Metta et G. Sandini (2011). « Computing robot internal/external wrenches by means of inertial, tactile and F/T sensors : theory and implementation on the iCub ». Dans : *Proc. of the 11th IEEE-RAS International Conference on Humanoid Robots - HUMANOIDS*. Bled, Slovenia, p. 521–528 (cf. p. 103).

Ivaldi, Serena, Natalia Lyubova, Damien Gerardeaux-Viret, Alain Droniou, Salvatore Anzalone, Mohamed Chetouani, David Filliat et Olivier Sigaud (sept. 2012a). « A cognitive architecture for developmental learning of objects and affordances : perception and human interaction aspects ». Dans : *IEEE Ro-man Workshop on Developmental and bio-inspired approaches for social cognitive robotics*. Paris, France (cf. p. 27).

— (2012b). « Perception and human interaction for developmental learning of objects and affordances ». Dans : *Proc. of the 12th IEEE-RAS International Conference on Humanoid Robots - HUMANOIDS*, p. 1–8 (cf. p. 28).

Ivaldi, Serena, Sao Mai Nguyen, Natalia Lyubova, Alain Droniou, Vincent Padois, David Filliat, Pierre-Yves Oudeyer et Olivier Sigaud (2014). « Object learning through active exploration ». Dans : *IEEE Transactions on Autonomous Mental Development* 6.1, p. 56 –72 (cf. p. 28).

Jaeger, Herbert (2001). « The "echo state" approach to analysing and training recurrent neural networks-with an erratum note ». Dans : *Bonn, Germany : German National Research Center for Information Technology GMD Technical Report* 148, p. 34 (cf. p. 70, 121).

Jain, AK, MN Murty et PJ Flynn (1999). « Data clustering : a review ». Dans : *ACM computing surveys (CSUR)* 31.3, p. 264–323 (cf. p. 88).

Johansson, Gunnar (1973). « Visual perception of biological motion and a model for its analysis ». Dans : *Perception & psychophysics* 14.2, p. 201–211 (cf. p. 118).

Johnsson, M, C Balkenius et G Hesslow (2009). « Associative Self-organizing Map. » Dans : *IJCCI*, p. 363–370 (cf. p. 45, 89).

Jung, M, J Hwang et J Tani (2014). « Multiple Spatio-Temporal Scales Neural Network for Contextual Visual Recognition of Human Actions ». Dans : *Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics*, p. 227–233 (cf. p. 119).

Kamyshanska, Hanna et Roland Memisevic (2013). « On autoencoder scoring ». Dans : *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, p. 720–728 (cf. p. 152, 173).

Kaschube, Matthias, Michael Schnabel et Fred Wolf (2008). « Self-organization and the selection of pinwheel density in visual cortical development ». Dans : *New Journal of Physics* 10.1, p. 015009 (cf. p. 32).

Keysers, Christian et Valeria Gazzola (2010). « Social neuroscience : mirror neurons recorded in humans ». Dans : *Current Biology* 20.8, R353–R354 (cf. p. 42).

Kilner, James M, Alice Neal, Nikolaus Weiskopf, Karl J Friston et Chris D Frith (2009). « Evidence of mirror neurons in human inferior frontal gyrus ». Dans : *The Journal of Neuroscience* 29.32, p. 10153–10159 (cf. p. 42).

Kiros, Ryan, Ruslan Salakhutdinov et Richard S Zemel (2014). « Unifying visual-semantic embeddings with multimodal neural language models ». Dans : *arXiv preprint arXiv :1411.2539* (cf. p. 170).

Klapper-Rybicka, Magdalena, Nicol N Schraudolph et Jürgen Schmidhuber (2001). « Unsupervised learning in LSTM recurrent neural networks ». Dans : *Artificial Neural Networks—ICANN 2001*. Springer, p. 684–691 (cf. p. 122).

Kohler, W (1929). « Gestalt Psychology (1929) ». Dans : *Liveright, New York* (cf. p. 34).
Kohonen, Teuvo (1990). « The self-organizing map ». Dans : *Proceedings of the IEEE* 78.9, p. 1464–1480 (cf. p. 64).
Kompella, Varun Raj, Leo Pape, Jonathan Masci, Mikhail Frank et Jürgen Schmidhuber (2011a). « Autoincsfa and vision-based developmental learning for humanoid robots ». Dans : *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*. IEEE, p. 622–629 (cf. p. 122).
Kompella, Varun Raj, Matthew D Luciw et Jürgen Schmidhuber (2011b). « Incremental Slow Feature Analysis. » Dans : *IJCAI*. T. 11, p. 1354–1359 (cf. p. 122).
Kouropteva, Olga, Oleg Okun et Matti Pietikäinen (2005). « Incremental locally linear embedding ». Dans : *Pattern recognition* 38.10, p. 1764–1767 (cf. p. 87).
Kreiman, Gabriel, Christof Koch et Itzhak Fried (2000). « Category-specific visual responses of single neurons in the human medial temporal lobe ». Dans : *Nature neuroscience* 3.9, p. 946–953 (cf. p. 33).
Krizhevsky, Alex, Geoffrey E Hinton et al. (2010). « Factored 3-way restricted boltzmann machines for modeling natural images ». Dans : *International Conference on Artificial Intelligence and Statistics*, p. 621–628 (cf. p. 152).
Krizhevsky, Alex, Ilya Sutskever et Geoffrey E Hinton (2012). « Imagenet classification with deep convolutional neural networks ». Dans : *Advances in neural information processing systems*, p. 1097–1105 (cf. p. 170).
Lachaux, Jean-Philippe (2011). *Cerveau attentif (Le) : Contrôle, maîtrise, lâcher-prise*. Odile Jacob (cf. p. 36, 42, 43).
Lallee, S. et P.F. Dominey (2013). « Multi-Modal Convergence Maps : From Body Schema and Self-Representation to Mental Imagery ». Dans : *Adaptive Behavior* 21, p. 274–285 (cf. p. 45, 89, 90, 111, 112).
Lange, Sascha et Martin Riedmiller (2010). « Deep auto-encoder neural networks in reinforcement learning ». Dans : *Neural Networks (IJCNN), The 2010 International Joint Conference on*. IEEE, p. 1–8 (cf. p. 150).
Lange, Sascha, Martin Riedmiller et A Voigtlander (2012). « Autonomous reinforcement learning on raw visual input data in a real world application ». Dans : *Neural Networks (IJCNN), The 2012 International Joint Conference on*. IEEE, p. 1–8 (cf. p. 150).
Larochelle, Hugo et Geoffrey E Hinton (2010). « Learning to combine foveal glimpses with a third-order Boltzmann machine ». Dans : *Advances in neural information processing systems*, p. 1243–1251 (cf. p. 172).
Law, Martin H C et Anil K Jain (mar. 2006). « Incremental nonlinear dimensionality reduction by manifold learning. » Dans : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.3, p. 377–91. issn : 0162-8828 (cf. p. 87).
Le, Quoc V, Rajat Monga, Matthieu Devin, Kai Chen, Greg S Corrado, Jeff Dean et Andrew Y Ng (2012). « Building High-level Features Using Large Scale Unsupervised Learning ». Dans : arXiv :arXiv:1112.6209v5 (cf. p. 74).
Le Cun, Yann (1986). « Learning process in an asymmetric threshold network ». Dans : *Disordered systems and biological organization*. Springer, p. 233–240 (cf. p. 14, 61).

Lee, Daniel et Sebastian Seung (2001). « Algorithms for non-negative matrix factorization ». Dans : *Advances in Neural Information Processing Systems*. T. 13, p. 556–562 (cf. p. 87).

Lee, Honglak, Alexis Battle, Rajat Raina et Andrew Y. Ng (2006). « Efficient sparse coding algorithms ». Dans : *Advances in Neural Information Processing Systems* (cf. p. 78, 87).

Lee, Honglak, Roger Grosse, Rajesh Ranganath et Andrew Y Ng (2009). « Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations ». Dans : *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. New York, NY, USA : ACM, p. 609–616. isbn : 978-1-60558-516-1 (cf. p. 73, 74, 88).

Lee, Sang-Hun et Randolph Blake (1999). « Visual form created solely from temporal structure ». Dans : *Science* 284.5417, p. 1165–1168 (cf. p. 117).

Lefort, Mathieu, Yann Boniface et Bernard Girau (2010). « Self-organization of neural maps using a modulated BCM rule within a multimodal architecture ». Dans : *BICS* (cf. p. 89).

Lefort, Mathieu, Thomas Kopinski, Alexander Gepperth et al. (2014). « Multimodal space representation driven by self-evaluation of predictability ». Dans : *ICDL-EPIROB-The fourth joint IEEE International Conference on Development and Learning and on Epigenetic Robotics* (cf. p. 89, 112).

Lemaignan, S., R. Ros, L. Mösenlechner, R. Alami et M. Beetz (2010). « ORO, a knowledge management module for cognitive architectures in robotics ». Dans : *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (cf. p. 18).

Lesaint, Florian, Olivier Sigaud, Shelly B Flagel, Terry E Robinson et Mehdi Khamassi (2014). « Modelling Individual Differences in the Form of Pavlovian Conditioned Approach Responses : A Dual Learning Systems Approach with Factored Representations ». Dans : *PLoS computational biology* 10.2, e1003466 (cf. p. 152).

Liwicki, Marcus, Alex Graves, Horst Bunke et Jürgen Schmidhuber (2007). « A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks ». Dans : *Proc. 9th Int. Conf. on Document Analysis and Recognition*. T. 1, p. 367–371 (cf. p. 120).

Lopes, Manuel et Pierre-Yves Oudeyer (nov. 2012). « The Strategic Student Approach for Life-Long Exploration and Learning ». Dans : *IEEE Conference on Development and Learning / EpiRob 2012*. San Diego, États-Unis (cf. p. 24).

Lungarella, Max, Giorgio Metta, Rolf Pfeifer et Giulio Sandini (2003). « Developmental robotics : a survey ». Dans : *Connection Science* 15.4, p. 151–190 (cf. p. 16).

Maass, Wolfgang, Thomas Natschläger et Henry Markram (2002). « Real-time computing without stable states : A new framework for neural computation based on perturbations ». Dans : *Neural computation* 14.11, p. 2531–2560 (cf. p. 59, 70, 121).

Macqueen, J.B. (1967). « Some methods for classification and analysis ». Dans : *Berkeley Symposium on Mathematical Statistics and Probability*. T. 233, p. 281–297 (cf. p. 88).

Madden, Carol, Michel Hoen et Peter Ford Dominey (2010). « A cognitive neuroscience perspective on embodied language for human–robot cooperation ». Dans : *Brain and language* 112.3, p. 180–188 (cf. p. 23).

Mangin, Olivier et Pierre-Yves Oudeyer (2013). « Learning semantic components from subsymbolic multimodal perception ». Dans : *Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*. IEEE, p. 1–7. isbn : 978-1-4799-1036-6 (cf. p. 89).

Martens, James (2010). « Deep learning via Hessian-free optimization ». Dans : *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, p. 735–742 (cf. p. 64, 75, 169).

Mauk, Michael D et Dean V Buonomano (2004). « The neural basis of temporal processing ». Dans : *Annu. Rev. Neurosci.* 27, p. 307–340 (cf. p. 123).

McCulloch, Warren S et Walter Pitts (1943). « A logical calculus of the ideas immanent in nervous activity ». Dans : *The bulletin of mathematical biophysics* 5.4, p. 115–133 (cf. p. 57).

Mcgurck, H et J W Macdonald (1976). « Hearing lips and seeing voices ». Dans : *Nature* 264.246-248 (cf. p. 35, 90).

Memisevic, Roland (2011). « Gradient-based learning of higher-order image features. » Dans : *Proceedings of the International Conference on Computer Vision ({ICCV})* (cf. p. 66).

— (2012a). « Learning to relate images : Mapping units, complex cells and simultaneous eigenspaces ». Dans : *ArXiv e-prints* (cf. p. 65, 126–128, 132).

— (2012b). « On multi-view feature learning ». Dans : *ICML* (cf. p. 126–128, 131, 136).

Memisevic, Roland et Geoffrey Hinton (2007). « Unsupervised Learning of Image Transformations ». Dans : *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE (cf. p. 126).

Memisevic, Roland et Geoffrey E Hinton (2010). « Learning to Represent Spatial Transformations with Factored Higher-Order Boltzmann Machines ». Dans : *Neural Computation* 22.6, p. 1473–1492 (cf. p. 66).

Memisevic, Roland, Christopher Zach, Geoffrey Hinton et Marc Pollefeys (2010). « Gated Softmax Classification ». Dans : *Advances in Neural Information Processing Systems*. Sous la dir. de J Lafferty, C K I Williams, J Shawe-Taylor, R S Zemel et A Culotta. T. 23, p. 1603–1611 (cf. p. 89).

Meyer, Kaspar et Antonio Damasio (2009). « Convergence and divergence in a neural architecture for recognition and memory ». Dans : *Trends in neurosciences* 32.7, p. 376–382 (cf. p. 35, 43, 44, 111–113).

Michalski, V., R. Memisevic et Kishore Konda (2014). « Modeling Deep Temporal Dependencies with Recurrent "Grammar Cells" ». Dans : *Advances in Neural Information Processing Systems 27* (cf. p. 125).

Minsky, M et S Papert (1969). *Perceptrons* (cf. p. 14).

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra et Martin Riedmiller (2013). « Playing Atari with deep reinforcement learning ». Dans : *arXiv preprint arXiv :1312.5602* (cf. p. 150).

Mnih, Volodymyr, Nicolas Heess, Alex Graves et Koray Kavukcuoglu (2014). « Recurrent Models of Visual Attention ». Dans : *arXiv preprint arXiv :1406.6247* (cf. p. 171).

Mohamed, Abdel-rahman, George E Dahl et Geoffrey Hinton (2012). « Acoustic modeling using deep belief networks ». Dans : *Audio, Speech, and Language Processing, IEEE Transactions on* 20.1, p. 14–22 (cf. p. 119).

Moldovan, Bogdan, Plinio Moreno, Martijn van Otterlo, José Santos-Victor et Luc De Raedt (2012). « Learning relational affordance models for robots in multi-object manipulation tasks ». Dans : *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, p. 4373–4378 (cf. p. 22).

Montesano, Luis et Manuel Lopes (2009). « Learning grasping affordances from local visual descriptors ». Dans : *Development and Learning, 2009. ICDL 2009. IEEE 8th International Conference on*. IEEE, p. 1–6 (cf. p. 22).

Montesano, Luis, Manuel Lopes, Alexandre Bernardino et José Santos-Victor (2008). « Learning Object Affordances : From Sensory–Motor Coordination to Imitation ». Dans : *Robotics, IEEE Transactions on* 24.1, p. 15–26 (cf. p. 22, 172).

Morgan, James L et Jenny R Saffran (1995). « Emerging integration of sequential and suprasegmental information in preverbal speech segmentation ». Dans : *Child development* 66.4, p. 911–936 (cf. p. 117).

Morse, Anthony F, Joachim De Greeff, Tony Belpeame et Angelo Cangelosi (2010a). « Epigenetic robotics architecture (ERA) ». Dans : *Autonomous Mental Development, IEEE Transactions on* 2.4, p. 325–339 (cf. p. 89).

Morse, Anthony F, Tony Belpaeme, Angelo Cangelosi et Linda B Smith (2010b). « Thinking with your body : Modelling spatial biases in categorization using a real humanoid robot ». Dans : *Proc. of 2010 annual meeting of the Cognitive Science Society. Portland, USA*, p. 1362–1368 (cf. p. 89, 110, 111).

Moser, Edvard I, Emilio Kropff et May-Britt Moser (2008). « Place cells, grid cells, and the brain's spatial representation system ». Dans : *Neuroscience* 31.1, p. 69 (cf. p. 33).

Muelling, Katharina, Jens Kober et Jan Peters (2010). « Learning table tennis with a mixture of motor primitives ». Dans : *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS Int. Conf. on*. IEEE, p. 411–416 (cf. p. 139).

Murata, Akira, Vittorio Gallese, Giuseppe Luppino, Masakazu Kaseda et Hideo Sakata (2000). « Selectivity for the shape, size, and orientation of objects for grasping in neurons of monkey parietal area AIP ». Dans : *Journal of neurophysiology* 83.5, p. 2580–2601 (cf. p. 43).

Nachev, Parashkev, Christopher Kennard et Masud Husain (2008). « Functional role of the supplementary and pre-supplementary motor areas ». Dans : *Nature Reviews Neuroscience* 9.11, p. 856–869 (cf. p. 42).

Nair, Vinod et Geoffrey E Hinton (2010). « Rectified linear units improve restricted boltzmann machines ». Dans : *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, p. 807–814 (cf. p. 59).

Nakamura, T., Takayuki Nagai et N. Iwahashi (2009). « Grounding of word meanings in multimodal concepts using LDA ». Dans : *International Conference on Intelligent Robots and Systems*. IEEE, p. 3943–3948 (cf. p. 23, 90, 111).
— (2011). « Bag of multimodal LDA models for concept formation ». Dans : *International Conference on Robotics and Automation*. IEEE, p. 6233–6238 (cf. p. 23, 90).
Narayanan, Hariharan et Sanjoy Mitter (2010). « Sample complexity of testing the manifold hypothesis ». Dans : *Advances in Neural Information Processing* (cf. p. 51).
Natale, L., F. Nori, G. Metta, M. Fumagalli, S. Ivaldi, U. Pattacini, M. Randazzo, A. Schmitz et G. G. Sandini (2013). « The iCub platform : a tool for studying intrinsically motivated learning ». Dans : *Intrinsically motivated learning in natural and artificial systems - Ed. Baldassarre, G.and Mirolli, M.* Springer-Verlag, p. 433–458 (cf. p. 100).
Neumann, Gerhard, Wolfgang Maass et Jan Peters (2009). « Learning complex motions by sequencing simpler motion templates ». Dans : *Proc. of the 26th Annual Int. Conf. on Machine Learning*, p. 753–760 (cf. p. 139).
Newell, Allen (1980). « Physical Symbol Systems ». Dans : *Cognitive science* 4.2, p. 135–183 (cf. p. 14).
Ngiam, Jiquan, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee et Andrew Y Ng (2011). « Multimodal Deep Learning ». Dans : *International Conference on Machine Learning*. Bellevue, USA, p. 689–696 (cf. p. 90).
Nguyen, Anh, Jason Yosinski et Jeff Clune (2014). « Deep Neural Networks are Easily Fooled : High Confidence Predictions for Unrecognizable Images ». Dans : *arXiv preprint arXiv :1412.1897* (cf. p. 21).
Nguyen, Sao Mai, Serena Ivaldi, Natalia Lyubova, Alain Droniou, Damien Gerardeaux-Viret, David Filliat, Vincent Padois, Olivier Sigaud et Pierre-Yves Oudeyer (2013). « Learning to recognize objects through curiosity-driven manipulation with the iCub humanoid robot ». Dans : *Proc. IEEE Int. Conf. Development and Learning and on Epigenetic Robotics - ICDL-EPIROB*, p. 1–8 (cf. p. 28).
Nocedal, Jorge et Stephen J Wright (2006). *Numerical Optimization*. Springer, p. 101–134 (cf. p. 64).
O'Hara, Stephen et Bruce A. Draper (2011). « Introduction to the bag of features paradigm for image classification and retrieval ». Dans : *arXiv preprint arXiv :1101.3354* (cf. p. 89).
Olazaran, Mikel (1996). « A sociological study of the official history of the perceptrons controversy ». Dans : *Social Studies of Science* 26.3, p. 611–659 (cf. p. 14).
Olshausen, Bruno A et David J Field (1997). « Sparse coding with an overcomplete basis set : A strategy employed by V1 ? » Dans : *Vision research* 37.23, p. 3311–3325 (cf. p. 87).
O'Regan, J Kevin et Alva Noë (2001). « A sensorimotor account of vision and visual consciousness ». Dans : *Behavioral and brain sciences* 24.05, p. 939–973 (cf. p. 15, 39, 43, 44).
Oudeyer, P-Y, Frédéric Kaplan et Verena Vanessa Hafner (2007). « Intrinsic motivation systems for autonomous mental development ». Dans : *Evolutionary Computation, IEEE Transactions on* 11.2, p. 265–286 (cf. p. 24, 156).

Pape, Leo, Faustino Gomez, Mark Ring et Jürgen Schmidhuber (2011). « Modular deep belief networks that do not forget ». Dans : *Neural Networks (IJCNN), The 2011 International Joint Conference on*. IEEE, p. 1191–1198 (cf. p. 168).

Papliński, AP et Lennart Gustafsson (2005). « Multimodal feedforward self-organizing maps ». Dans : *Computational Intelligence and Security*, p. 81–88 (cf. p. 45, 89).

Parker, David B (1985). « Learning logic ». Dans : (cf. p. 14, 61).

Pascanu, Razvan, Guido Montufar et Yoshua Bengio (2013). « On the number of inference regions of deep feed forward networks with piece-wise linear activations ». Dans : *arXiv preprint arXiv :1312.6098* (cf. p. 73).

Pastor, P., H. Hoffmann, T. Asfour et S. Schaal (2009). « Learning and generalization of motor skills by learning from demonstration ». Dans : *Int. Conf. on Robotics and Automation (ICRA2009)* (cf. p. 139).

Pastor, Peter, Mrinal Kalakrishnan, Franziska Meier, Freek Stulp, Jonas Buchli, Evangelos Theodorou et Stefan Schaal (avr. 2013). « From Dynamic Movement Primitives to Associative Skill Memories ». Dans : *Robot. Auton. Syst.* 61.4, p. 351–361. issn : 0921-8890 (cf. p. 139).

Petitot, Jean (2003). « The neurogeometry of pinwheels as a sub-Riemannian contact structure ». Dans : *Journal of Physiology-Paris* 97.2, p. 265–309 (cf. p. 31).

Pfeifer, Rolf et Josh Bongard (2007). *How the body shapes the way we think : a new view of intelligence*. MIT press (cf. p. 16).

Pfeifer, Rolf et Christian Scheier (1999). *Understanding intelligence*. MIT press (cf. p. 16).

Piaget, Jean (1936). « La naissance de l'intelligence chez l'enfant ». Dans : (cf. p. 20, 24).

— (1937). « La construction du réel chez l'enfant. » Dans : (cf. p. 20).

Poon, Hoifung et Pedro Domingos (2011). « Sum-Product Networks : A New Deep Architecture ». Dans : *UAI*, p. 337–346 (cf. p. 112).

Quiroga, R Quian, Leila Reddy, Gabriel Kreiman, Christof Koch et Itzhak Fried (2005). « Invariant visual representation by single neurons in the human brain ». Dans : *Nature* 435.7045, p. 1102–1107 (cf. p. 33).

Quiroga, R Quian, Gabriel Kreiman, Christof Koch et Itzhak Fried (2008). « Sparse but not 'grandmother-cell'coding in the medial temporal lobe ». Dans : *Trends in cognitive sciences* 12.3, p. 87–91 (cf. p. 33).

Quiroga, Rodrigo Quian (2012). « Concept cells : the building blocks of declarative memory functions ». Dans : *Nature Reviews Neuroscience* 13.8, p. 587–597 (cf. p. 33).

Rafal, Robert (2003). « Balint's syndrome : A disorder of visual cognition ». Dans : *Neurological foundations of cognitive neuroscience*, p. 27 (cf. p. 36).

Raiko, Tapani, Li Yao, Kyunghyun Cho et Yoshua Bengio (2014). « Iterative Neural Autoregressive Distribution Estimator (NADE-k) ». Dans : *arXiv preprint arXiv :1406.1485* (cf. p. 168).

Raina, Rajat, Anand Madhavan et Andrew Y Ng (2009). « Large-scale deep unsupervised learning using graphics processors. » Dans : *ICML*. T. 9, p. 873–880 (cf. p. 169).

Ranzato, Marc'Aurelio (2014). « On Learning Where To Look ». Dans : *arXiv preprint arXiv :1405.5488* (cf. p. 171).

Rao, Rajesh PN et Dana H Ballard (1999). « Predictive coding in the visual cortex : a functional interpretation of some extra-classical receptive-field effects ». Dans : *Nature neuroscience* 2.1, p. 79–87 (cf. p. 176).

Rebecchi, Sébastien, Hélène Paugam-Moisy et Michèle Sebag (2014). « Learning sparse features with an auto-associator ». Dans : *Growing Adaptive Machines*. Springer, p. 139–158 (cf. p. 78).

Reed, Scott et Honglak Lee (2013). « Learning Deep Representations via Multiplicative Interactions between Factors of Variation ». Dans : *NIPS Workshop* (cf. p. 88).

Renaudo, Erwan, Benoît Girard, Raja Chatila et Mehdi Khamassi (2014). « Design of a Control Architecture for Habit Learning in Robots ». Dans : *Biomimetic and Biohybrid Systems*. Springer, p. 249–260 (cf. p. 152).

Rensink, Ronald A, J Kevin O'Regan et James J Clark (1997). « To see or not to see : The need for attention to perceive changes in scenes ». Dans : *Psychological science* 8.5, p. 368–373 (cf. p. 36).

Rensink, Ronald A, J Kevin O'Regan et James J Clark (2000). « On the failure to detect changes in scenes across brief interruptions ». Dans : *Visual cognition* 7.1-3, p. 127–145 (cf. p. 36).

Ridge, Barry, D Skocaj et A Leonardis (2010). « Self-supervised cross-modal online learning of basic object affordances for developmental robotic systems ». Dans : *International Conference on Robotics and Automation*. IEEE, p. 5047–5054. isbn : 9781424450404 (cf. p. 45, 89).

Rifai, Salah, Pascal Vincent, Xavier Muller, Xavier Glorot et Yoshua Bengio (2011a). « Contractive Auto-Encoders : Explicit Invariance During Feature Extraction ». Dans : *Proceedings of the 28th International Conference on Machine Learning*, p. 833–840 (cf. p. 51, 78).

Rifai, Salah, Grégoire Mesnil, Pascal Vincent, Xavier Muller, Yoshua Bengio, Yann Dauphin et Xavier Glorot (2011b). « Higher Order Contractive Auto-Encoder ». Dans : *ECML/PKDD (2)*, p. 645–660 (cf. p. 78).

Rifai, Salah, Yann N Dauphin, Pascal Vincent, Yoshua Bengio et Xavier Muller (2011c). « The Manifold Tangent Classifier ». Dans : *Advances in Neural Information Processing Systems*. Sous la dir. de J Shawe-Taylor, R S Zemel, P Bartlett, F C N Pereira et K Q Weinberger, p. 2294–2302 (cf. p. 88).

Rizzolatti, Giacomo et Laila Craighero (2004). « The mirror-neuron system ». Dans : *Annu. Rev. Neurosci.* 27, p. 169–192 (cf. p. 42).

Rosenblatt, Frank (1958). « The perceptron : a probabilistic model for information storage and organization in the brain. » Dans : *Psychological review* 65.6, p. 386 (cf. p. 14).

Rosenblum, Lawrence D, Jennifer A Johnson et Helena M Saldana (1996). « Point-light facial displays enhance comprehension of speech in noise ». Dans : *Journal of Speech, Language, and Hearing Research* 39.6, p. 1159–1170 (cf. p. 118).

Rovee, Carolyn Kent et David T Rovee (1969). « Conjugate reinforcement of infant exploratory behavior ». Dans : *Journal of experimental child psychology* 8.1, p. 33–39 (cf. p. 23).

Rumelhart, D.E., G.E. Hintont et R.J. Williams (1986). « Learning representations by back-propagating errors ». Dans : *Nature* 323.6088, p. 533–536 (cf. p. 14, 61, 76).

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein et al. (2014). « Imagenet large scale visual recognition challenge ». Dans : *arXiv preprint arXiv :1409.0575* (cf. p. 20).

Saberi, Kourosh et David R Perrott (1999). « Cognitive restoration of reversed speech ». Dans : *Nature* 398.6730, p. 760–760 (cf. p. 118).

Şahin, Erol, Maya Çakmak, Mehmet R Doğar, Emre Uğur et Göktürk Üçoluk (2007). « To afford or not to afford : A new formalization of affordances toward affordance-based robot control ». Dans : *Adaptive Behavior* 15.4, p. 447–472 (cf. p. 22).

Sakai, Katsuyuki, Okihide Hikosaka, Satoru Miyauchi, Yuka Sasaki, Norio Fujimaki et Benno Pütz (1999). « Presupplementary motor area activation during sequence learning reflects visuo-motor association ». Dans : *Journal of Neuroscience* 19, RC1–1 (cf. p. 42).

Sakai, Katsuyuki, Katsuya Kitaguchi et Okihide Hikosaka (2003). « Chunking during human visuomotor sequence learning ». Dans : *Experimental brain research* 152.2, p. 229–242 (cf. p. 117).

Salakhutdinov, Ruslan et Geoffrey E Hinton (2008). « Using deep belief nets to learn covariance kernels for gaussian processes ». Dans : *Advances in neural information processing systems*, p. 1249–1256 (cf. p. 82).

— (2009). « Deep boltzmann machines ». Dans : *International Conference on Artificial Intelligence and Statistics*, p. 448–455 (cf. p. 82).

Salakhutdinov, Ruslan R, Josh Tenenbaum et Antonio Torralba (2011). « Learning to Learn with Compound HD Models ». Dans : *Advances in Neural Information Processing Systems*. Sous la dir. de J Shawe-Taylor, R S Zemel, P Bartlett, F C N Pereira et K Q Weinberger. T. 24, p. 2061–2069 (cf. p. 89).

Sallans, Brian et Geoffrey E. Hinton (2000). « Using Free Energies to Represent Q-values in a Multiagent Reinforcement Learning Task. » Dans : *NIPS*. MIT Press, p. 1075–1081 (cf. p. 150, 151, 153).

Sallans, Brian et Geoffrey E Hinton (2004). « Reinforcement learning with factored states and actions ». Dans : *The Journal of Machine Learning Research* 5, p. 1063–1088 (cf. p. 152).

Salman, Ahmad et Ke Chen (juil. 2011). « Exploring speaker-specific characteristics with deep learning ». Dans : *The 2011 International Joint Conference on Neural Networks*. IEEE, p. 103–110. isbn : 978-1-4244-9635-8 (cf. p. 82).

Samuelson, Larissa K (2002). « Statistical regularities in vocabulary guide language acquisition in connectionist models and 15-20-month-olds. » Dans : *Developmental psychology* 38.6, p. 1016 (cf. p. 23).

Schmidhuber, Jürgen (1992a). « Learning complex, extended sequences using the principle of history compression ». Dans : *Neural Computation* 4.2, p. 234–242 (cf. p. 125).

Schmidhuber, Jürgen (1992b). « Learning factorial codes by predictability minimization ». Dans : *Neural Computation* 4.6, p. 863–879 (cf. p. 125).
— (2010). « Formal theory of creativity, fun, and intrinsic motivation (1990–2010) ». Dans : *Autonomous Mental Development, IEEE Transactions on* 2.3, p. 230–247 (cf. p. 24, 156).
Schott, JM et MN Rossor (2003). « The grasp and other primitive reflexes ». Dans : *Journal of Neurology, Neurosurgery & Psychiatry* 74.5, p. 558–560 (cf. p. 40).
Schreiner, Christoph E (1992). « Functional organization of the auditory cortex : maps and mechanisms ». Dans : *Current opinion in neurobiology* 2.4, p. 516–521 (cf. p. 31).
Searle, John R (1980). *Minds, brains, and programs*. T. 3. 03. Cambridge Univ Press, p. 417–424 (cf. p. 14, 15).
Seashore, CE (1899). « Some psychological statistics. 2. The material-weight illusion ». Dans : *University of Iowa Studies in Psychology* 2, p. 36–46 (cf. p. 41).
Sharma, Jitendra, Alessandra Angelucci et Mriganka Sur (2000). « Induction of visual orientation modules in auditory cortex ». Dans : *Nature* 404.6780, p. 841–847 (cf. p. 31).
Shima, Keisetsu, Masaki Isoda, Hajime Mushiake et Jun Tanji (2006). « Categorization of behavioural sequences in the prefrontal cortex ». Dans : *Nature* 445.7125, p. 315–318 (cf. p. 42).
Silver, Michael A et Sabine Kastner (2009). « Topographic maps in human frontal and parietal cortex ». Dans : *Trends in cognitive sciences* 13.11, p. 488–495 (cf. p. 33).
Simon, Dylan A et Nathaniel D Daw (2012). « Dual-system learning models and drugs of abuse ». Dans : *Computational Neuroscience of Drug Addiction*. Springer, p. 145–161 (cf. p. 152).
Simons, Daniel J (2000). « Attentional capture and inattentional blindness ». Dans : *Trends in cognitive sciences* 4.4, p. 147–155 (cf. p. 36).
Smith, Linda et Michael Gasser (2005). « The development of embodied cognition : Six lessons from babies ». Dans : *Artificial life* 11.1-2, p. 13–29 (cf. p. 20).
Smith, Linda B, Susan S Jones, Barbara Landau, Lisa Gershkoff-Stowe et Larissa Samuelson (2002). « Object name learning provides on-the-job training for attention ». Dans : *Psychological Science* 13.1, p. 13–19 (cf. p. 23).
Smolensky, Paul (1986). « Information processing in dynamical systems : Foundations of harmony theory ». Dans : (cf. p. 26, 80).
Strigl, Daniel, Klaus Kofler et Stefan Podlipnig (2010). « Performance and scalability of GPU-based convolutional neural networks ». Dans : *Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on*. IEEE, p. 317–324 (cf. p. 169).
Stuhlsatz, André, Jens Lippel et Thomas Zielke (2010). « Discriminative feature extraction with Deep Neural Networks ». Dans : *IJCNN*. IEEE, p. 1–8 (cf. p. 83, 89).
Stulp, Freek, Gennaro Raiola, Antoine Hoarau, Serena Ivaldi et Olivier Sigaud (2013). « Learning Compact Parameterized Skills with Expanded Function Approximators ». Dans : *Proc. of the IEEE Int. Conf. on Humanoids Robotics*, p. 1–7 (cf. p. 139).

Sumner, Petroc, Parashkev Nachev, Peter Morris, Andrew M Peters, Stephen R Jackson, Christopher Kennard et Masud Husain (2007). « Human medial frontal cortex mediates unconscious inhibition of voluntary action ». Dans : *Neuron* 54.5, p. 697–711 (cf. p. 42).

Sutton, R. et Andrew G Barto (1998). *Reinforcement learning : An introduction*. MIT press (cf. p. 150, 152).

Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow et Rob Fergus (2013). « Intriguing properties of neural networks ». Dans : *arXiv preprint arXiv :1312.6199* (cf. p. 21).

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke et Andrew Rabinovich (2014). « Going deeper with convolutions ». Dans : *arXiv preprint arXiv :1409.4842* (cf. p. 20).

Tang, Yichuan, Nitish Srivastava et Ruslan R Salakhutdinov (2014). « Learning Generative Models with Visual Attention ». Dans : *Advances in Neural Information Processing Systems 27*. Sous la dir. de Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence et K.Q. Weinberger. Curran Associates, Inc., p. 1808–1816 (cf. p. 172).

Taylor, Graham W, Geoffrey E Hinton et Sam T Roweis (2006). « Modeling human motion using binary latent variables ». Dans : *Advances in neural information processing systems*, p. 1345–1352 (cf. p. 119).

Taylor, Graham W., Rob Fergus, Yann LeCun et Christoph Bregler (sept. 2010). « Convolutional learning of spatio-temporal features ». Dans : *ECCV'10*, p. 140–153. isbn : 3-642-15566-9, 978-3-642-15566-6 (cf. p. 126).

Tenorth, Moritz et Michael Beetz (2013). « KnowRob : A knowledge processing infrastructure for cognition-enabled robots ». Dans : *The International Journal of Robotics Research* 32.5, p. 566–590 (cf. p. 18).

Tenorth, Moritz, Ulrich Klank, Dejan Pangercic et Michael Beetz (2011). « Web-enabled robots ». Dans : *Robotics & Automation Magazine, IEEE* 18.2, p. 58–68 (cf. p. 18).

Turing, Alan M (1950). « Computing machinery and intelligence ». Dans : *Mind*, p. 433–460 (cf. p. 13).

Ugur, Emre, Erhan Oztop et Erol Sahin (2011). « Goal emulation and planning in perceptual space using learned affordances ». Dans : *Robotics and Autonomous Systems* 59.7, p. 580–595 (cf. p. 22).

Van Biervliet, J-J et al. (1895). « La mesure des illusions de poids ». Dans : *L'année psychologique* 2.1, p. 79–86 (cf. p. 41).

Varela, F.J., E. Thompson et E. Rosch (1993). *L'inscription corporelle de l'esprit : sciences cognitives et expérience humaine*. La couleur des idées. Seuil. isbn : 9782020134927 (cf. p. 29).

Vavrečka, M et I Farkaš (2013). « A Multimodal Connectionist Architecture for Unsupervised Grounding of Spatial Language ». Dans : *Cognitive Computation*, p. 1–12 (cf. p. 45, 89).

Vincent, Pascal (2011). « A connection between score matching and denoising autoencoders ». Dans : *Neural computation* 23.7, p. 1661–1674 (cf. p. 173).

Vincent, Pascal, Hugo Larochelle, Yoshua Bengio et Pierre-Antoine Manzagol (2008). « Extracting and composing robust features with denoising autoencoders ». Dans : *Proceedings of the 25th international conference on Machine learning - ICML '08*. New York, New York, USA : ACM Press, p. 1096–1103. isbn : 9781605582054 (cf. p. 77).

Vinyals, Oriol, Alexander Toshev, Samy Bengio et Dumitru Erhan (2014). « Show and Tell : A Neural Image Caption Generator ». Dans : *arXiv preprint arXiv :1411.4555* (cf. p. 170).

Volpi, Nicola Catenacci, Jean Charles Quinton et Giovanni Pezzulo (2014). « How active perception and attractor dynamics shape perceptual categorization : A computational model ». Dans : *Neural Networks* 60, p. 1–16 (cf. p. 171, 172).

Von Melchner, Laurie, Sarah L Pallas et Mriganka Sur (2000). « Visual behaviour mediated by retinal projections directed to the auditory pathway ». Dans : *Nature* 404.6780, p. 871–876 (cf. p. 31).

Wahlström, Niklas, Thomas B Schön et Marc Peter Deisenroth (2014). « Learning deep dynamical models from image pixels ». Dans : *arXiv preprint arXiv :1410.7550* (cf. p. 125).

Waibel, Markus, Michael Beetz, Javier Civera, Raffaello d'Andrea, Jos Elfring, Dorian Galvez-Lopez, Kai Haussermann, Rob Janssen, JMM Montiel, Alexander Perzylo et al. (2011). « A World Wide Web for Robots ». Dans : *IEEE Robotics & Automation Magazine* (cf. p. 18).

Ward, JH (1963). « Hierarchical grouping to optimize an objective function ». Dans : *Journal of the American statistical association* 58.301, p. 236–244 (cf. p. 88).

Warren, Richard M (1970). « Perceptual restoration of missing speech sounds ». Dans : *Science* 167.3917, p. 392–393 (cf. p. 118).

Weng, Juyang (2004). « Developmental robotics : Theory and experiments ». Dans : *International Journal of Humanoid Robotics* 1.02, p. 199–236 (cf. p. 16).

Werbos, Paul (1974). « Beyond regression : New tools for prediction and analysis in the behavioral sciences ». Thèse de doct. (cf. p. 14, 61).

Werbos, Paul J (1990). « Backpropagation through time : what it does and how to do it ». Dans : *Proceedings of the IEEE* 78.10, p. 1550–1560 (cf. p. 119).

Wermter, S., C. Weber, M. Elshaw, C. Panchev, H. Erwin et F. Pulvermüller (juin 2004). « Towards multimodal neural robot learning ». Dans : *Robotics and Autonomous Systems* 47.2-3, p. 171–175. issn : 09218890 (cf. p. 45, 89).

Wiestler, Tobias et Jörn Diedrichsen (2013). « Skill learning strengthens cortical representations of motor sequences ». Dans : *Elife* 2 (cf. p. 117).

Williams, Ronald J et David Zipser (1989). « Experimental analysis of the real-time recurrent learning algorithm ». Dans : *Connection Science* 1.1, p. 87–111 (cf. p. 121).

Wiskott, Laurenz et Terrence Sejnowski (2002). « Slow feature analysis : Unsupervised learning of invariances ». Dans : *Neural computation* 14.4, p. 715–770 (cf. p. 122).

Xing, Eric P, Michael I Jordan, Stuart Russell et Andrew Y Ng (2002). « Distance metric learning with application to clustering with side-information ». Dans : *Advances in neural information processing systems*, p. 505–512 (cf. p. 88).

Yamashita, Yuichi et Jun Tani (2008). « Emergence of functional hierarchy in a multiple timescale neural network model : a humanoid robot experiment ». Dans : *PLoS computational biology* 4.11, e1000220 (cf. p. 119).

Zhang, Li I, Huizhong W Tao, Christine E Holt, William A Harris et Mu-ming Poo (1998). « A critical window for cooperation and competition among developing retinotectal synapses ». Dans : *Nature* 395.6697, p. 37–44 (cf. p. 56, 58).

Zhao, Haitao, Pong Chi Yuen et James T Kwok (août 2006). « A novel incremental principal component analysis and its application for face recognition. » Dans : *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society* 36.4, p. 873–86. issn : 1083-4419 (cf. p. 87).