Références bibliographiques

[1] A. Tickle, R. Andrews, M. Golea et J. Diederich. *The truth will come to light : directions and challenges in extracting the knowledge embedded within trained artificial neural networks.* IEEE transactions on neural networks 9:6, pages 1057-1068, 1998.

[2] I. Aleksander. *An automata-theoretic assessment of the cognitive debate.* Publication dans artificial neural networks 2, Elsevier science publishers, 1992.

[3] A. P. Azcarraga. *Modèles neuronaux pour la classification incrémentale de formes visuelles.* Thèse de doctorat en informatique, INPG, France, 1993.

[4] E. Alpaydin. *Multiple neural experts for improved decision making.* Travaux de recherche TR-80815, Bogazici University, Turkey, 1999.

[5] E. Alpaydin. *GAL : networks that grow when they learn and shrink when they forget.* Technical Report TR-91-032, ICSI-Berkley, 1991.

[6] Y. Belala. *Systèmes de production et unification connexionnistes.* Thèse de doctorat en informatique, université Paris-Sud, Centre d'Orsay, France, 1992.

[7] N. Beauboucher. *ANAIS : Raisonnement à partir de cas en résolution de problèmes.* Thèse de doctorat en informatique, université de Paris VI, 1997.

[8] H. Bersini. *Locality and oddity in the connectivity of biological networks.* Journées de Rochebrune, ENST'95-S-001, 1995.

[9] O. Brousse. *Generativity and systematicity in neural network combinatorial learning.* Technical report CU-CS-676-93, department of computer science, university of Colorado, 1993.

[10] G. A. Carpenter et G. Grossberg. *A massively parallel architecture for a self-organizing neural pattern recognition machine.* Publication dans Computer vision graphics image processing, volume 37, 1987.

[11] NASA's Johnson Space Center. *CLIPS : Reference manual (basic programming guide).* Software Technology Branch, USA, 1998.

[12] M. Chaillot. *Une architecture de contrôle réactif pour la résolution coopérative de problèmes.* Thèse de doctorat en informatique, INPG, Grenoble, 1993.

[13] A. Colmerauer. *Prolog in 10 figures.* Publication dans Communications of the ACM 28:12, pages 1296-1310, 1985.

[14] A. Cornuejols. *Connexionnisme et représentation de haut niveau.* Actes du Workshop sur les modèles symboli-connexionnistes, ENST-Paris, avril 1990.

[15] M. Cottrell, J. C. Fort, et G. Pagès. *Two or three things that we know about the Kohonen algotithm.* Publication dans Verlaysen, pages 235-244, 1994.

[16] M. Crucianu. *Représentations structurées dans les réseaux connexionnistes.* Thèse de doctorat en informatique, université Paris XI Orsay, France, 1994.

[17] J. M. David, J. P. Krivine, et R. Simmons. *Second generation expert systems : a step forward in knowledge engineering.* Publication dans Second generation expert systems, Springer Verlag, 1993.

[18] G. Deco. *Pruning algorithms for the GALATEA N-Library.* Publication dans Advanced in neural information systems 2, Morgan Kofmann, 1992.

[19] Demongeot, Benaouda, et Jezequel. *Random simulations and confiners : their application to neural networks.* Dans Acta Biotheorica, volume 4, N°2/3, pages 203-213, 1994.

[20] B. Deuker, M. Perrier, et B. Amy. *Fault-diagnosis using neuro-symbolic hybrid systems.* In 9th international workshop on principles of diagnosis, USA, 1996.

[21] G. Dorffner et E. Prem. *Connectionism, symbol grounding and autonomous agents.* Technical report TR-2510, Austrian research institue for artificial intelligence, university of Vienna, 1993.

[22] J. L. Elman. *Finding structure in time.* Rapport technique N° 8801, center of research in language, university of California, 1988.

[23] J. Euzenat. *Représentations de connaissances : de l'approximation à la confrontation.* Mémoire d'habilitation à diriger des recherches en informatique, université de Joseph Fourier, Grenoble, 1999.

[24] S. E. Fahlman et C. Lebiere. *The cascade correlation learning architecture.* Rapport de recheche CMU-CS-90-100, computer science department, Carnegie Mellon university, Febrary 1990.

[25] E. A. Feigenbaum. *Themes and case studies of knowledge engineering.* Dans expert systems in the micro-electronic age, Edimburg university press, pages 3-25, 1979.

[26] B. Fritzke. *Growing cell structure, a self organizing network for unsupersised and supervised learning.* Publication dans Neural networks, 7(9), 1994.

[27] E. Gauthier. *Utilisation des réseaux de neurones artificiels pour la commande d'un véhicule autonome*. Thèse de doctorat en informatique, laboratoire de Leibniz-IMAG, Grenoble, 1999.

[28] A. Giacometti. *Modèles hybrides de l'expertise*. Thèse de doctorat en informatique, Ecole Nationale Supérieure des Télécommunications, Paris, 1992.

[29] M. Gordon et D. Berchier. *Minimerror : a perceptron learning rule that finds the optimal weights*. Dans ESANN 93, Brussels, April 7-9, proceedings Michel Verleysen edition, 1993.

[30] M. Gutknecht et R. Pfeifer. *Experiments with a hybrid architecture : integrating expert systems with connetionist networks*. Proceedings of 10th international conference on artificial intelligence, expert systems and natural language, Avignon, 1990.

[31] R. F. Hadley. *Systematicity in connectionist language learning*. Publication dans mind & language, volume 4, 1994.

[32] R. F. Hadley. *Cognition, systematicity and nomic necessity*. Publication dans mind & language, volume 12, 1996.

[33] J. Hertz, A. S. Krogh, et R. G. Palmer. *Introduction to the theory of neural computation*. Dans Book Review in Artificial Intelligence, 1993.

[34] M. Hilario, Y. Lallement, et F. Alexandre. *Neurosymbolic integration : unified versus hybrid approaches*. In the european symposium on artificial neural networks, Brussels, Belgium, 1995.

[35] R. Hecht-Nielsen. *Neurocomputing*. Addison-Wesley, 1989.

[36] T. Kohonen. *Self-organised formation of topologically correct feature maps*. Dans biological cybernetics (43), pages 59-69, 1982.

[37] A. Labbi. *Sur l'approximation et les systèmes dynamiques dans les réseaux neuronaux : applications en intelligence artificielle*. Thèse de doctorat en informatique de l'INPG, Grenoble, 1993.

[38] Y. Lallement, M. Hilario, et F. Alexandre. *Neurosymbolic integration : cognitive grounds and computational strategies*. World conference on the fundamentals of artificial intelligence, Paris, 1995.

[39] A. Lallouet. *Modularité, validation et parallélisme de données en programmation logique*. Thèse de doctorat en informatique, université d'Orléans, France, 1996.

[40] K. McGarry, S. Wermter, et J. MacIntyre. *Hybrid neural systems : from simple coupling to fully integrated neural networks.* In neural computing surveys (2), pages 62-93, 1999.

[41] W. S. McCulloch et W. Pitts. *A logical calculus of the ideas immanent in nervous activity.* Publication dans Bull Mathematics Biophysics (5), pages 115-113, 1943.

[42] M. Malek. *Un modèle hybride de mémoire pour le raisonnement à partir de cas.* Thèse de doctorat en informatique, université Joseph Fourier, Grenoble, 1996.

[43] A. Ram et J. C. Santamaria. *A Multistrategy case-based and reinforcement learning approach to self-improving reactive control systems for autonomous robotic navigation.* Proceedings of the second international workshop on multistrategy learning, May 1993.

[44] L. R. Medsker et D. L. Baily. *Models and guidlines for integrating expert systems and neural networks.* Dans Kandel et Langholz [1992], Chapitre 8, pages 154-171, 1992.

[45] B. Neveu. *Systèmes experts et conception d'outils pour l'intelligence artificielle.* Rapport technique - Projet SECOIA, INRIA, France, 1991.

[46] N. J. Nilsson. *Introduction to machine learning (draft of a proposed new text book).* Department computer science, Stanford university, USA, 1996.

[47] B. Orsier, B. Amy, V. Rialle, et A. Giacometti. *A study of the hybrid system SYNTHESYS.* In ECAI94 Workshop : "combining symbolic and connectionist processing", Amsterdam, 1994.

[48] B. Orsier et A. LABBI. *NESSY3L : a NEuroSymbolic SYstem with 3 Levels.* Publication dans knowledge-based systems, 1996.

[49] F. S. Osório. *INSS : Système hybride neuro-symbolique pour l'apprentissage automatique constructif.* Thèse de doctorat en informatique, Laboratoire de Leibniz-IMAG, Grenoble, 1998.

[50] P. Partakelidis, F. Fessant, et B. Amy. *Application des réseaux de neurones à l'intelligence artificielle - application de l'apprentissage par renforcement à l'évitement d'obstacles et à la recherche d'une source lumineuse (simulateur de robot KHEPERA).* Rapport technique, Laboratoire de Leibniz-IMAG, 1996.

[51] P. Peretto et J. J. Niez. *Stochastic dynamics of neural networks.* Publication dans IEEE transactions on systems, man and cybernetics, volume SMC 16, n°1, 1986.

[52] D. Reilly, L. Cooper, et C. Elbaum. *A neural model for category learning.* Dans biological cybernetics (45), pages 35-41, 1982.

[53] G. Reyes. *Etude des connaissances dans les réseaux de neurones artificiels : représentation et explicitation de règles de haut niveau.* Rapport de DEA en sciences cognitives, INPG, Grenoble, 1997.

[54] F. Rechenmann. *Shirka : un système de gestion de bases de connaissances centreés-objet.* Manuel d'utilisation, unité de recherche INRIA Rhone-Alpes / IMAG, 1993.

[55] J. Sagaut. *Les réseaux de neurones.* Rapport de recherche N° 14/SGDN/VST/5, SGDN/VST et AI-ACCESS, 1995.

[56] L. Shastri et J. A. Feldman. *Semantic networks and neural nets.* Technical Report TR-131, Rocherster university, computer science department, 1984.

[57] H. A. Simon. *Why should Machines Learn ?.* Dans machine learninig : an artificial intelligence approach, volume 1, San Mateo, CA-USA, 1983.

[58] J. Stalin. *Vectorized backpropagation and automatic pruning for PML networks optimization.* Dans IEEE-ICNN international conference, Volume 3, 1993.

[59] R. Sun. *Integrating rules and connectionism for robust reasoning.* Technical Report TR-CS-90-154, Brandeis university, computer science department, 1991.

[60] G. Towell et J. Shavlik. *Extracting refined rules from knowledge-based neural networks.* Publication dans Machine learning (13), 1993.

[61] G. Towell. *Symbolic knowledge and neural networks : insertion, refinement and extraction.* PHD Thesis, university of Wisconsin-Madison, USA, 1991.

[62] J. M. Torres Monero et M. Gordon. *An evolutive architecture coupled with optimal perceptron learning for classification.* Proceedings of ESANN'95, Michel Verleysen Editions, Brussels, April 19-21, 1995.

[63] V. N. Vapnik. *Principles of risk minimization for learning theory.* Publication dans neural information processing systems (4), 1992.

[64] F. J. Varela. *Essay 1 : autopoiesis and a biology of intentionality.* Technical Report TR 231-CREA, CNRS-Ecole Polytechnique de Paris, France, 1991.

[65] G. Weisbuch. *Dynamique des systèmes complexes, une introduction aux réseaux d'automates.* InterEditions/Editions du CNRS, 1989.

[66] J. J. Hopfield. *Neural networks and physical systems with emergent collective computational abilities.* In proceedings national academy of sciences 79, USA, 1982.

[67] N. Szilas. *Apprentissage dans les réseaux récurrents pour la modélisation mécanique et étude de leurs interactions avec l'environnement.* Thèse de doctorat en sciences cognitives de l'INPG, Grenoble, 1996.

[68] J. Gould et R. Levinson. *Method Integration for experience-based learning.* Rapport de recherche UCSC-CRL-91-27, Baskin center for computer engineering and information sciences, university of California, Santa Cruz, 1991.

[69] V. Ciesielski, S. Hayes, et B. Kelly. *Comparison of an expert system and an hybrid neural network/expert system for a respiratory monitoring problem.* Dans AAAI-92 workshop on integrating neural and symbolic processes, California, 1992.

[70] W. R. Becraft, P. L. Lee, et R. B. Newell. *Integration of neural networks and expert systems.* In international joint conference on artificial intelligence, pages 832-837. Morgan-Kaufmann, 1991.

[71] D. A. Handelman, S.H. Lane, et J. J. Gelfand. *Robotic skill acquisition based on biological principles.* Dans proceedings of the the 10th international conference on artificial intelligence, expert systems and natural language. Avignon, 1992.

[72] R. Sun. *A connectionist module for commonsense reasoning incorporating rules and similarities.* Publication dans knowledge acquisition (4), pages 293-321, 1992.

[73] D. S. Touretzky. *Boltzcons : Dynaminc symbol structures in a connectionist network.* Publication dans artificial intelligence, 46(1-2), 1990.

[74] L. Fu. *Learning capacity and sample on expert networks.* Publication dans IEEE transactions on neural networks 7(6), pages 1517-1520, 1996.

[75] V. Nenov et M. Dyer. *Perceptually grounded language learning : Part 1- a neural network architecture for robust sequence association.* Dans connection science 5(2), 1993.

[76] C. McMillan, M. C. Mozer, et P. Smolensky. *The connectionist scientist game : rule extraction and refinement in a neural network.* Dans proceedings of the 13th annual conference of the cognitive science society, pages 424-430, Hillsdale, 1991.

[77] R. Andrews, J. Diederich, et A. B. Tickle. *A survey and critique of techniques for extracting rules from trained artificial neural networks.* Publication dans knowledge-based systems 8(6), pages 373-389, 1995.

[78] J. A. Feldman, G. Lakoff, D. Bailey, S. Narayanan, T. Regier et A. Stolcke. *The first five years of an automated language acquisition project.* Publication dans AI review special on integration of vision and language, volume 8, 1996.

[79] P. Smolensky, G. Legengre, et Y. Miyata. *Principles for integrated connectionist/ symbolic theory of higher cognition*. Rapport de recherche CU-CS-600-92, département d'informatique, université de Colorado Boulder, 1992.

[80] F. Alexandre, Y. Burnod, F. Guyot, et J-P. Haton. *The cortical column : a new processing unit for multilayered networks*. Publication dans neural networks 4(1), 1991.

[81] M. Hilario, C. Pellegrini, et F. Alexandre. *Modular integration of connectionist and symbolic processing in knowledge-based systems*. Publication dans knowledge-based systems - proceedings of the international symposium on integrating knowledge and neural heuristics pages 123-132, Pensacola, Florida, 1994.

[82] G. Towell and J. Shavlik. *Knowledge-based artificial neural networks*. Publication dans artificial intelligence 70, pages 119-165, 1994.

[83] J. Shavlik. *An overview of research at wisconsin on knowledge-based neural networks*. In proceedings of the international conference on neural networks, pages 65-69, Washington, 1996.

[84] V. Ajjanagadde et L. Shastri. *Rules and variables in neural nets*. Publication dans neural computation (3), pages 121-134, 1991.

Chapitre 1

Réseaux de neurones artificiels

Le principe des réseaux de neurones artificiels est né dans les années 40 à partir d'une analogie avec le système nerveux humain. Le terme désigne aujourd'hui un très grand nombre de modèles, dont beaucoup n'ont plus grand chose à voir avec le fonctionnement des neurones biologiques, et doit donc être pris comme une métaphore.

Un réseau de neurones artificiels est un réseau fortement connecté de processeurs élémentaires (neurones formels) fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit en entrée.

On associe généralement aux réseaux de neurones artificiels des algorithmes d'apprentissage permettant de modifier de manière plus au moins automatique le traitement effectué afin de réaliser une tâche donnée.

Aujourd'hui de nombreux termes sont utilisés dans la littérature pour désigner le domaine des RNA, comme connexionnisme ou neuromimétique. Cependant, il est nécessaire d'associer à chacun de ces noms une sémantique précise. Ainsi, les réseaux de neurones artificiels désignent les modèles manipulés. Connexionnisme et neuromimétique sont tous deux des domaines de recherche qui manipulent ces modèles, mais avec des objectifs différents. L'objectif des ingénieurs et chercheurs connexionnistes est d'améliorer les capacités de l'informatique en utilisant des modèles aux composants fortement connectés. Pour leur part, les neuromiméticiens manipulent des modèles de réseaux de neurones artificiels dans l'unique but de vérifier leurs théories biologiques sur le fonctionnement du système nerveux central.

Nous présenterons dans ce chapitre la structure d'un neurone formel, les principales classes de réseaux de neurones artificiels, leurs méthodes d'apprentissages, leurs classes d'applications, ainsi qu'une analyse de leurs points forts et points faibles.

1.1 Neurone formel

Le neurone formel qui représente la brique de base des réseaux de neurones artificiels est un automate dont le modèle s'inspire de celui d'un neurone biologique (figure 1.1). Un neurone formel d'indice *i* est décrit par les éléments suivants :

- son état (ou activation) a_i, qui peut être une valeur réelle ou booléenne. Cet état est généralement choisi comme valeur de sortie du neurone ;
- ses connexions d'entrée auxquelles sont associés des poids w_{ij} (j est l'indice du neurone partageant la connexion) ;
- sa fonction d'entrée réalisant un prétraitement (généralement une somme pondérée e_i des entrées) ;
- sa fonction d'activation (ou de transfert) g, qui calcule à partir du résultat de la fonction d'entrée l'activation du neurone.

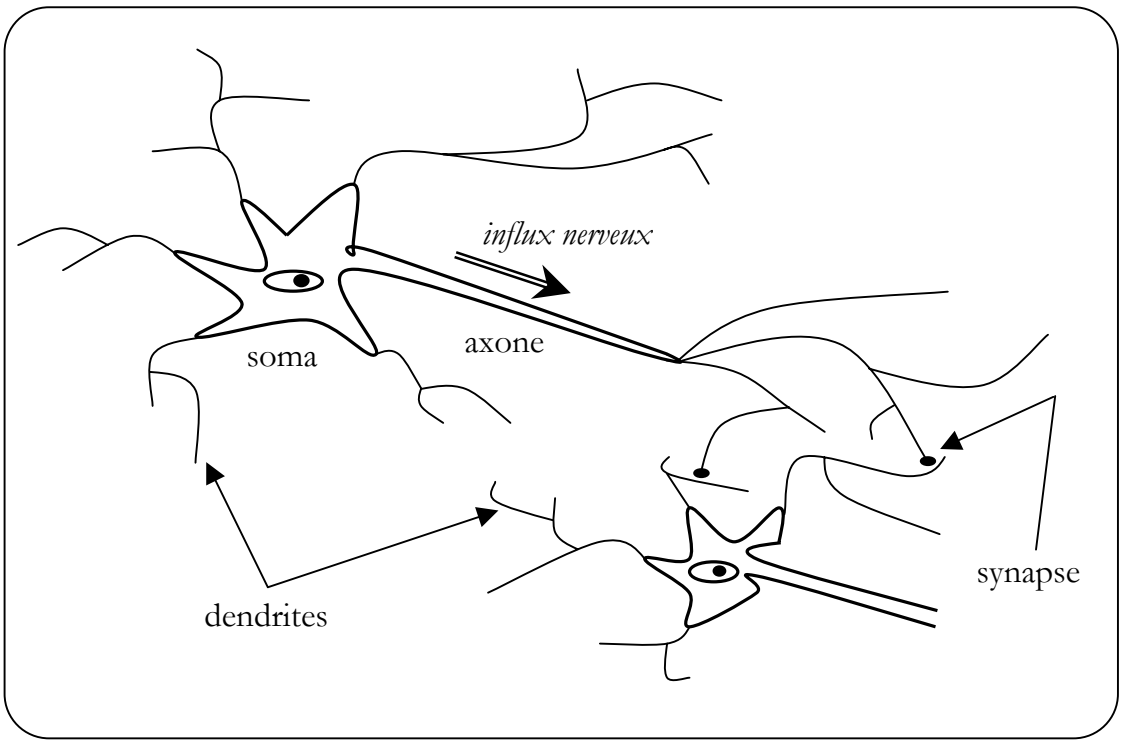

Figure 1.1 : Illustration d'un neurone pyramidal du cortex cérébral.

La figure 1.2 représente la structure d'un neurone formel appliquant une fonction non-linéaire sur la somme pondérée de ses trois entrées. La première modélisation du neurone, présentée par McCulloch et Pitts [41], date de 1943.

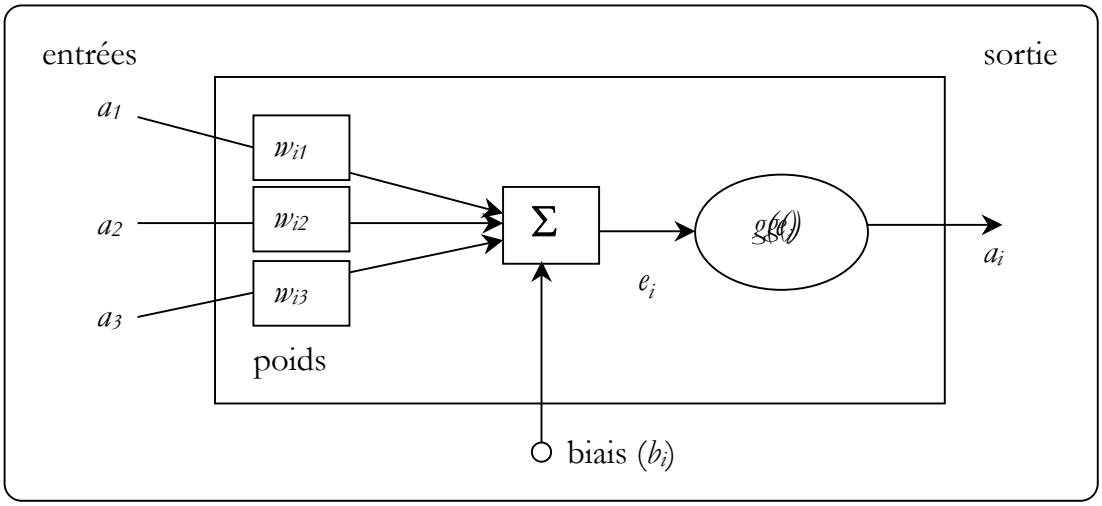

Figure 1.2 : Structure d'un neurone formel.

S'inspirant de leurs travaux sur les neurones biologiques, la sortie du neurone formel s'exprime simplement par une fonction non-linéaire de la somme pondérée des entrées,

$$e_i = \sum_{j=1}^{N} w_{ij}\, a_j + b_i \tag{1.1}$$

$$a_i = g(e_i) \tag{1.2}$$

où a_j est l'activation du neurone j et représente les entrées du neurone i, w_{ij} désigne les poids synaptiques, et b_i est appelé le *biais* du neurone qui peut être représenté par le poids d'une connexion provenant d'un neurone (neurone biais) dont l'activité reste fixée à 1. La non-linéarité de g se nomme la *fonction d'activation*. Cette fonction était grossièrement représentée par une fonction à seuil du type :

$$g(e_i) = \begin{cases} +1 & si \quad e_i \succ \beta \\ 0 & sinon \end{cases}$$

Ainsi, si la somme pondérée (1.1) dépasse un certain seuil β, sa sortie (1.2) est +1 et le neurone est activé ; dans le cas contraire, la sortie est 0 et le neurone est inactivé. Seules des modifications mineures ont été opérées sur se modèle de base. De nos jours, il est plus commun d'employer une fonction non-linéaire bornée et dérivable sur]-∞, +∞[plutôt qu'une fonction linéaire par morceaux. La fonction sigmoïde est couramment employée :

$$g(e_i) = \frac{1}{1 + e^{-\beta e_i}} \quad \text{ou} \quad g(e_i) = tanh(e_i)\,,$$

1.2 Principales classes de RNA

Il s'agit de décrire les grandes classes de modèles de réseaux de neurones artificiels à partir des principes qui président à leur fonctionnement, et de montrer leur grande variété, et ceci même s'il n'existe pas de classification parfaite. L'habitude a été prise de diviser les réseaux de neurones artificiels en deux grandes classes : les RNA supervisés et les RNA non-supervisés. Sagaut [55] a remarqué que cette habitude n'est pas satisfaisante. D'une part, ces deux appellations sont incorrectes, puisqu'elles s'appliquent en fait non aux réseaux de neurones mais aux algorithmes d'apprentissage qui permettent de les construire. D'autre part, certains réseaux de neurones entrent dans les deux catégories.

De manière générale, deux réseaux de neurones ont été à l'origine du renouveau des travaux sur le connexionnisme au début des années 80 : le perceptron multi-couches et le réseau de Hopfield. Ces deux modèles caractérisent respectivement deux grandes classes de réseaux : les RNA unidirectionnels et les RNA récurrents. Comme il existe aussi d'autres classes particulières souvent déterminées par l'utilisation que l'on veut faire des réseaux ou par l'intention qui préside à leur étude et leur conception. Elles contiennent des modèles nouveaux construits spécialement pour certaines applications. Se sont : les RNA cellulaires, localistes, probabilistes, modulaires, neuro-flous, et neuro-biologiquement plausibles (voir section 1.2.3).

1.2.1 Les RNA unidirectionnels

Les RNA unidirectionnels (*feedforward networks*) sont caractérisés par le fait que la propagation d'activité à travers le réseau se fait directement depuis des automates d'entrée jusqu'aux automates de sortie sans qu'il y ait de boucles de rétroaction. Le perceptron multi-couches (figure 1.3) a été à l'origine de ces modèles. Les RNA unidirectionnels sont répartis en deux sous-classes : les RNA multi-couches classiques et les RNA à base de prototypes.

Les RNA multi-couches classiques

Le modèle standard, le perceptron multi-couches (PMC), est le plus connu, avec ses trois couches de cellules dont une couche dite cachée. L'interprétation de cette couche en terme de connaissances sur le domaine d'entrée pose de nombreux problèmes. C'est cette difficulté à interpréter la configuration du réseau après apprentissage qui a conduit à voir les réseaux de neurones artificiels comme des "boites noires" au comportement certes souvent efficace mais inexplicable.

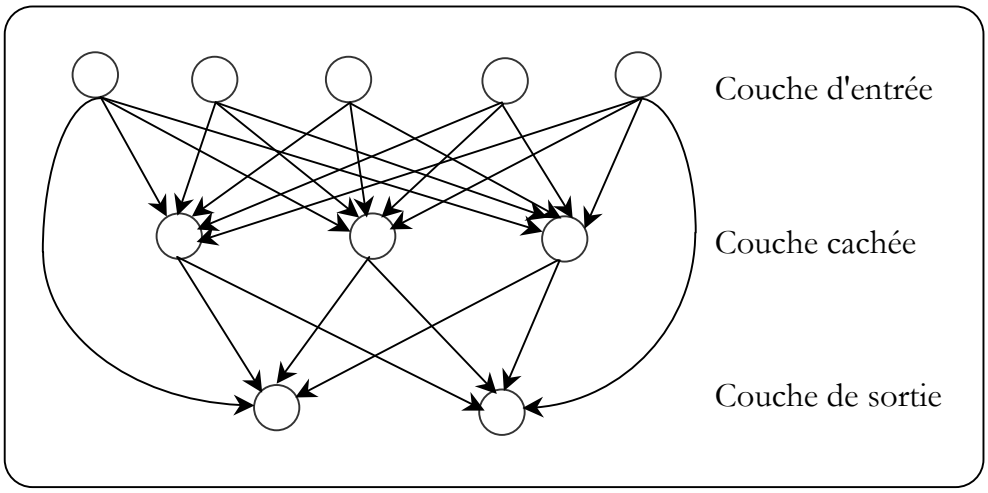

Figure 1.3 : Un RNA de type PMC.

Les problèmes de construction des PMC tels que l'absence de méthodes systématiques, l'aspect heuristique de la conception des réseaux de neurones, la rigidité des architectures que l'on ne peut modifier en cours de conception puis d'utilisation, ont poussé les chercheurs à mettre au point des RNA multi-couches à architectures évolutives (RNA incrémentaux). Parmi ces réseaux on cite :

- l'algorithme d'apprentissage incrémental monoplan de Gordon et Torres-Moreno [62], qui construit un réseau de neurones artificiels à une couche cachée en ajoutant des perceptrons au fur et à mesure des besoins, et qui montre des performances excellentes tant en apprentissage qu'en généralisation ;
- le modèle INSS (*Incremental Neuro-Symbolic System*) qui part d'une compilation de règles de connaissances symboliques pour construire un RNA incrémental du type *Cascade Correlation* [49].

Les RNA à base de prototypes

Le principe qui préside au fonctionnement de ces modèles est de mémoriser un nombre limité d'exemples ou de prototypes représentatifs de chaque classe dans le cas de la classification, de chaque partie de la fonction dans le cas de l'approximation [42].

Un RNA à base de prototypes comprend ainsi une mémoire d'exemples, un mécanisme d'apprentissage permettant de mémoriser de nouveaux exemples ou de modifier les prototypes déjà mémorisés, et un mécanisme d'utilisation permettant de trouver les exemples les plus proches du vecteur présenté en entrée du réseau.

Ces réseaux ont en général trois couches, la couche centrale (ou couche cachée) comprenant des automates dont chacun représente un prototype. Les valeurs des attributs d'un prototype, c'est à dire ses composantes si chaque point de l'espace d'entrée est représenté par un vecteur, sont enregistrées dans les poids des connexions qui lient l'automate prototype aux unités de la couche cachée. Les paramètres w_{ij} représentent donc les composantes d'un prototype ou d'un exemple. Comme exemples de réseaux à base de prototypes, on cite :

- Les différents travaux de Kohonen [36] sont à la base de ce type de modèles connexionnistes.
- Les RNA du type RBF (*Radial Basis Function*) ont été construits à l'origine pour résoudre des problèmes d'approximation. Un RNA de ce type contient trois couches dont une couche cachée sur laquelle les automates représentent des fonctions "noyaux" formant une nouvelle base pour les exemples d'entrée. La transformation de la couche d'entrée vers la couche cachée est non-linéaire alors que la transformation de la couche cachée vers la couche de sortie est linéaire. Ces RNA sont très utilisés en classification, en approximation fonctionnelle ou pour l'approximation de densités de probabilité dans les processus de décision bayésienne.
- Les différents modèles du type ART (*Adaptive Resonance Theory*) développés par Carpenter et Grossberg [10] sont également des RNA à base de prototypes. Un mécanisme de va et vient entre la couche d'entrée et la couche des prototypes permet de sélectionner le prototype le mieux adapté au pattern d'entrée. Ces modèles sont complexes et difficiles à mettre en œuvre.
- Le réseau GAL (*Grow And Learn*) [5] est un réseau incrémental à bases d'exemples. Il en existe une version dans laquelle les exemples sont remplacés par des prototypes.

1.2.2 Les RNA récurrents

Dans les RNA récurrents au contraire, on autorise les boucles, chaque automate pouvant être connecté à des voisins déjà activés ou à lui-même. On a ainsi un processus de relaxation au cours duquel le réseau passe par une série d'états d'activation. Le réseau de Hopfield (figure 1.4) a été à l'origine de ces modèles. Les RNA récurrents sont répartis en deux sous-classes : les RNA fortement connectés et les RNA unidirectionnels bouclés.

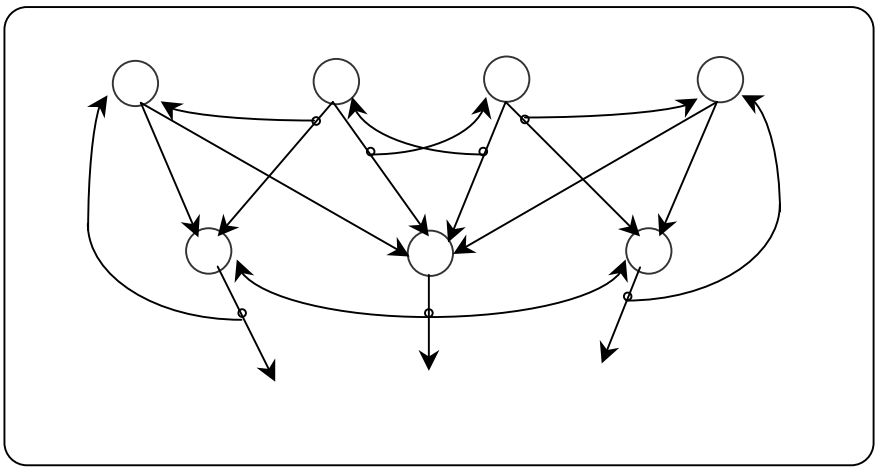

Figure 1.4 : Un RNA de type Hopfield.

Les RNA fortement connectés

Le modèle de Hopfield [66] est le plus connu. Ses publications font partie des articles fondateurs du domaine. Hopfield a relancé l'étude des réseaux connexionnistes en 1982. D'autres modèles dynamiques fortement connectés sont étudiés par des équipes comme celle de l'ONERA (Office National d'Etudes et de Recherche Aérospatiales) de Toulouse ou celle de l'IMAG (Institut des Mathématiques Appliquées de Grenoble). On peut citer en particulier les RNA récurrents à très grand nombre d'automates, utilisés comme mémoires associatives [19] ou les RNA dynamiques à variables externes dont la dynamique est "*pilotée*" par les données de l'environnement (dans les tâches d'aide à la prise de décision ou de pilotage, le système peut *"changer d'avis"* pendant la phase de prise de décision si l'environnement change) [37].

Hors le domaine de la modélisation neurobiologique, le modèle de Hopfield et ses dérivés sont surtout utilisés comme mémoires associatives ou comme processeurs dans des calculs d'optimisation [55].

Les RNA unidirectionnels bouclés

Les RNA unidirectionnels bouclés, si on les compare aux modèles précédents, sont faiblement connectés. On parle aussi de "RNA faiblement récurrents".
Les premiers réseaux de neurones artificiels construits à partir de ce principe ont été des réseaux pour le traitement des séries temporelles. Si l'on observe qu'une série simple $A \to B \to C \to D \to ect.$ peut être décomposée en une suite d'associations $A \to B$, $B \to C$, $C \to D$, *etc.*, on voit alors qu'il suffit d'apprendre à un RNA du type PMC

cette suite d'associations puis de connecter les cellules de sortie sur les cellules d'entrée pour obtenir un RNA récurrent ayant la propriété suivante : si on lui présente le vecteur d'entrée *A*, il va donner en sortie successivement *B, C, D, etc.*

De manière générale, dans ce type de réseau, on réinjecte donc en entrée d'un RNA unidirectionnel les sorties de certains neurones des couches cachées ou de la couche de sortie, ce qui a pour effet de boucler le réseau sur lui-même. Les deux principaux modèles sont ceux imaginés par Jordan [33] dont le principe de fonctionnement est celui décrit ci-dessus, et ceux de Elman [22] dans lesquels on réinjecte en entrée les niveaux d'activité des cellules de la couche cachée d'un PMC.

L'idée de refermer sur elle-même une architecture unidirectionnelle associative a été reprise dans de nombreux modèles des processus cognitifs. C'est le cas des travaux de Aleksander [2] et de ce que certains appellent l'Ecole Britannique sur les NSMM (*Neural State Machine Model*).

1.2.3 Classes particulières de RNA

Il existe d'autres classes de réseaux de neurones artificiels que celles qui viennent d'être citées. Elles sont souvent déterminées par l'utilisation que l'on veut faire des réseaux ou par l'intention qui préside à leur étude et leur conception. Elles contiennent des modèles déjà cités ci-dessus, mais aussi des modèles nouveaux construits spécialement pour certaines applications. Se sont : les RNA cellulaires, localistes, probabilistes, modulaires, neuro-flous et neurobiologiquement plausibles.

Les RNA cellulaires

Au contraire des modèles neuronaux classiques, les connexions entre automates d'un RNA cellulaire constituent des voisinages locaux réguliers, chaque automate ne communiquant qu'avec certains voisins plus ou moins proches [65]. Les automates les plus utilisés sont soit des automates à seuil, soit des automates booléens opérant sur des variables binaires et dont les fonctions de transition d'état sont des fonctions booléennes.

Les RNA cellulaires ont été parmi les premiers étudiés, en particulier par les physiciens et les mathématiciens des mathématiques discrètes. Ils sont principalement utilisés pour la modélisation de processus physiques tels que les écoulements fluides, les phénomènes de combustion, les réactions chimiques ou physico-chimiques, les épidémies, les systèmes immunitaires, et les phénomènes vibratoires [67].

Les RNA localistes

Un des concepts clés des réseaux de neurones est celui de la distribution de l'information. Cette caractéristique a pour conséquence l'impossibilité d'interpréter au niveau local le fonctionnement du réseau : l'examen de l'évolution et des états successifs de chaque automate ne renseigne pas sur la fonction globale du système et ne donne aucune indication sur son domaine de fonctionnement.

Dans un RNA localiste on garde le caractère distribué du traitement, mais on en revient à une représentation localisée de l'information sur le domaine d'application : alors que le fonctionnement du réseau repose toujours sur un mécanisme de relaxation et d'évolution vers des états attracteurs, chaque automate du réseau représente à lui seul une information sur le domaine. C'est le cas des RNA localistes étudiés pour l'Intelligence Artificielle (IA) à l'université de Rochester aux Etats-Unis au début des années 80 par Shastri [56] et les chercheurs du département informatique. Ces réseaux sont la "traduction" connexionniste des réseaux sémantiques. Chaque cellule représente un concept ou une connaissance sur des concepts, l'inférence étant représentée par la propagation d'activité à travers les mailles du réseau.

Les RNA probabilistes

Dans les RNA probabilistes, les automates ne sont plus déterministes. Leurs lois de transition d'état sont des lois donnant les probabilités de transition des sorties et des états internes. Appartiennent à cette catégorie les RNA probabilistes de Hopfield [51].

Les RNA modulaires

La difficulté qu'ont les réseaux de neurones les plus courants à traiter des données dont le volume et la complexité deviennent très grands a conduit les chercheurs à imaginer des architectures modulaires dans lesquelles plusieurs réseaux coopèrent.

Une première approche consiste à décomposer le problème traité en plusieurs sous-problèmes, chacun étant pris en charge par un réseau. C'est ce que l'on fait dans les apprentissages multitâches, en particulier en robotique modulaire.

La méthode la plus classique consiste à ne considérer qu'un seul niveau de décision et à spécialiser chaque module sur une classe [58]. On peut aussi utiliser ce que l'on appelle un "réseau d'experts" dans lequel la partition des données est faite automatiquement au cours de l'apprentissage. Les différents réseaux sont placés sous la supervision d'un réseau qui pondère leurs sorties et prend la décision finale.

Une autre méthode consiste à faire fonctionner plusieurs réseaux de neurones sur la totalité du domaine et, à chaque fois, de chercher le réseau de neurones qui donne le meilleur résultat possible [4].

Les RNA neuro-flous

La possibilité d'utiliser en entrée d'un réseau des cellules dont les niveaux d'activité soient des grandeurs continues (par exemple variant entre 0 et 1, au lieu de prendre les valeurs 0 ou 1), permet de coder la plus ou moins forte appartenance d'un objet à un ensemble. D'où l'idée d'utiliser des réseaux neuronaux pour effectuer des traitements dans le cadre de la logique floue.

Il se trouve aussi que les ensembles de règles manipulées par les systèmes de raisonnement à base de logique floue sont facilement représentables par des réseaux dont l'architecture est celle des RNA multi-couches. Ces liens entre le connexionnisme et le flou a donné lieu à de nombreux travaux qui ont eux-mêmes conduit à la réalisation de réseaux neuronaux très performants et utilisés dans des applications industrielles [55].

Les RNA neuro-biologiquement plausibles

Le paradoxe du connexionnisme tient dans le fait suivant : les données de la neurobiologie ont servi à la fois de point de départ pour le domaine (les travaux de MacCulloch et Pitts [41] visaient à modéliser les neurones vivants) et de stimulants pour le redémarrage des recherches en 1982 (le réseau présenté par Hopfield [66] se voulait neuromimétique), alors que le modèle qui est devenu le plus connu, le PMC, est justement le moins neuromimétique de tous, surtout au niveau de ses algorithmes d'apprentissage.

Ces dernières années de nombreux chercheurs sont revenus à l'étude de RNA plus biologiquement plausibles. Pour l'instant, ces travaux sont orientés surtout vers la modélisation neurobiologique ou l'étude de ce que l'on appelle la Vie Artificielle. Les architectures étudiées dans le cadre de ces recherches sont essentiellement des architectures récurrentes. Nous citons les RNA du type Hopfield à grand nombre de cellules, les RNA à poids variables, les RNA écologiques et les RNA biologiques asynchrones [8].

1.3 Méthodes d'apprentissages des RNA

Le savoir faire des réseaux connexionnistes dépend des poids des connexions et des grandeurs formant les paramètres des fonctions de transition des automates. En ajustant ces paramètres, on peut donc modifier cette fonction, la rapprocher d'une fonction voulue et faire en sorte qu'un réseau ait un comportement prévu à l'avance. Tout se passe comme si le réseau apprenait son comportement. On parle ainsi d'apprentissage, même si la procédure suivie est loin d'être un apprentissage au sens de la psychologie du comportement et de la didactique.

Cette propriété des architectures connexionnistes est désormais largement connue et a donné lieu à d'innombrables travaux. On distingue les types d'apprentissage suivants :

- les apprentissages supervisés au cours desquels on impose au réseau de neurones de donner des réponses connues,
- les apprentissages dits non-supervisés au cours desquels le RNA se contente de traiter ses données d'entrées en fonction de ses seuls mécanismes internes,
- les apprentissages particuliers qui sont situés entre les apprentissages supervisés et les apprentissages non-supervisés tels que les apprentissages par renforcement et les apprentissages par compilation des connaissances.

1.3.1 Apprentissages supervisés

Dans les cas des apprentissages supervisés, le problème est en fait un problème d'optimisation puisque l'on cherche des valeurs des paramètres qui minimisent l'erreur faite par le RNA. La méthode la plus utilisée a longtemps été une méthode de descente du gradient à pas constant, la célèbre *"Rétro-Propagation du Gradient (RPG)"* [27].

Algorithme de rétro-propagation du gradient

Fondé sur la modification des poids, l'algorithme de rétropropagation du gradient est le plus connu pour réaliser l'adaptation des RNA multi-couches. Nous allons présenter brièvement la méthode d'obtention de ce gradient, qui se base sur le calcul des dérivées partielles successives de fonctions composées. La mesure de performance utilisée est l'erreur quadratique suivant :

$$Q = \frac{1}{2}\sum_i [a_i - s_i]^2,$$

où i parcourt les indices des neurones de sortie, et a_i et s_i représentent respectivement l'activation mesurée et l'activation désirée pour ces neurones.

Les poids du réseau de neurones sont modifiés en suivant la règle :

$$\Delta w_{ij} = -\eta \frac{\partial Q}{\partial w_{ij}} , \tag{1.3}$$

où η est une constante positive appelée pas du gradient. Le calcul de la quantité $\partial Q / \partial w_{ij}$ se fait en partant de la couche de sortie et en se déplaçant vers la couche d'entrée. Cette propagation, suivant le sens inverse de celui de l'activation des neurones du réseau, justifie le nom de l'algorithme. Le calcul est décomposé de la manière suivante :

$$\frac{\partial Q}{\partial w_{ij}} = \frac{\partial Q}{\partial a_i} \frac{\partial a_i}{\partial e_i} \frac{\partial e_i}{\partial w_{ij}} .$$

En posant, $\delta_i = \frac{\partial Q}{\partial a_i} \frac{\partial a_i}{\partial e_i}$ on obtient $\frac{\partial Q}{\partial w_{ij}} = \delta_i \frac{\partial e_i}{\partial w_{ij}}$,

Et puisque $\frac{\partial e_i}{\partial w_{ij}} = a_j$ alors : $\Delta w_{ij} = -\eta \, \delta_i a_j$

La quantité δ_i est appelée contribution à l'erreur du neurone *i*. Dans le cas où *i* est l'indice d'un neurone de sortie, on obtient :

$$\frac{\partial Q}{\partial a_i} = (a_i - s_i) , \quad \frac{\partial a_i}{\partial e_i} = g'(e_i) ,$$

et donc : $\delta_i = g'(e_i)\,(a_i - s_i)$

Dans le cas où *i* est l'indice d'un neurone caché, on pose :

$$\frac{\partial Q}{\partial a_i} = \sum_k \frac{\partial Q}{\partial a_k} \frac{\partial a_k}{\partial a_i} ,$$

où *k* parcourt les indices de tous les neurones vers lesquels le neurone *i* envoie une connexion. Le calcul nous donne :

$$\frac{\partial Q}{\partial a_k} \frac{\partial a_k}{\partial a_i} = \frac{\partial Q}{\partial a_k} \frac{\partial a_k}{\partial e_k} \frac{\partial e_k}{\partial a_i} = \delta_k \frac{\partial e_k}{\partial a_i} = \delta_k w_{ki}$$

Nous obtenons donc : $\frac{\partial Q}{\partial a_i} = \sum_k \delta_k \, w_{ki}$, et $\delta_i = g'(e_i) \sum_k \delta_k \, w_{ki}$

Cet algorithme, présenté ici dans sa version la plus simple, possède de nombreuses variantes. Elles correspondent généralement à l'utilisation de valeurs variables pour la constante η, ou l'utilisation de méthodes du deuxième ordre pour le calcul du gradient. On utilise souvent une version légèrement différente de l'équation (1.3) pour calculer la quantité dont doivent être modifiés les poids :

$$\Delta w_{ij}(t) = -\eta \frac{\partial Q}{\partial w_{ij}} + \mu \Delta w_{ij}(t-1),$$

où μ représente une constante appelée momentum, et t représente le temps. Cette version introduit un deuxième terme proportionnel à la dernière adaptation de w_{ij}.

Les modifications des poids peuvent intervenir après chaque présentation d'un patron, ou après la présentation de l'ensemble de la base d'exemples. L'apprentissage nécessite dans tous les cas un grand nombre de présentations de la totalité de ces exemples pour obtenir un résultat satisfaisant.

La méthode de la rétropropagation du gradient a été conçue pour les RNA multi-couches du type PMC. Dès que l'on passe aux RNA récurrents, le problème devient plus complexe. Dans le cas des RNA unidirectionnels bouclés la généralisation de la RPG aboutit à des algorithmes tels que la "*Back-propagation through time*" ou le "*Real time recurrent learning*" que leur complexité les rend pour l'instant difficilement utilisables [55].

1.3.2 Apprentissages non-supervisés

Les apprentissages non-supervisés s'appliquent aux RNA à bases de prototypes, bien que ceux-ci puissent être utilisés aussi en mode supervisé. Dans ce cas, le réseau fonctionne comme un système de catégorisation (*clustering*) dans lequel les différents prototypes sont adaptés au fur et à mesure que sont présentées les données. Cette adaptation des prototypes peut être réalisée de trois façons différentes :

- **par assimilation :** si l'exemple présenté en entrée n'est proche d'aucun prototype connu, la procédure d'apprentissage crée un nouveau prototype en recopiant l'exemple nouveau.

- **par accommodation :** si l'exemple présenté est reconnu comme proche d'un prototype, il est ajouté à l'ensemble des exemples représenté par ce prototype. Il faut alors modifier légèrement ce dernier de façon à ce qu'il reste le centre de gravité du nouvel ensemble.

- **par différentiation :** si le prototype activé par le réseau de neurones lorsqu'on présente un exemple correspond à une fausse classe, il faut réduire la région d'influence du prototype de façon à exclure l'exemple.

Dans tous les cas le but est de couvrir au mieux les classes définies par le réseau de neurones et donc d'effectuer le meilleur pavage possible du domaine d'entrée à l'aide des prototypes.

On parle d'apprentissage non-supervisé parce que l'on ne donne pas au réseau de neurones la réponse attendue. Mais le réseau se modifie pourtant, et il le fait d'une façon qui est prédéterminée par les lois d'évolution fixées par le concepteur. Ce dernier intervient donc, même si c'est de manière moins directe que dans le cas de la RPG. Il le fait en particulier à travers le choix d'une loi importante, celle qui précise comment le réseau calcule la proximité d'un exemple avec les différents prototypes. Le concepteur doit pour cela choisir une mesure de distance qui va influencer implicitement les types de classes que pourra déterminer le réseau. Il n'y a pas d'apprentissage totalement non-supervisé.

1.3.3 Autres apprentissages particuliers

Il existe deux autres types d'apprentissages particuliers, à savoir : les apprentissages par renforcement et les apprentissages par compilation de connaissances.

Apprentissages par renforcement

Il existe une classe très particulière de méthodes d'apprentissage située à mi-chemin entre les apprentissages supervisé et non-supervisé. Il s'agit des apprentissages par renforcement dans lesquels on se contente de donner au réseau de neurones artificiels une évaluation de la qualité de sa réponse (bonne ou mauvaise). Ces méthodes sont très utiles dans les domaines où l'on a peu d'information sur la réponse désirée. C'est en particulier le cas des problèmes de conduite de trajectoires en robotique où l'on se trouve souvent dans le cas où pour une situation donnée on a plusieurs comportements possibles du robot sans que l'on puisse dire lequel est le meilleur. Ces méthodes d'apprentissage sont aujourd'hui très étudiées [27].

Apprentissages par compilation de connaissances

Les méthodes d'apprentissage étant des méthodes d'optimisation, elles se heurtent à un problème classique des techniques de recherche, celui du choix du point de départ. Les réseaux de neurones ont rencontré ce problème dès le début du connexionnisme. Et les

chercheurs ont très vite essayé de trouver des méthodes leur permettant d'initialiser l'architecture à optimiser aussi près que possible de l'optimum recherché. Pour cela on utilise toutes les connaissances a priori que l'on peut avoir sur le domaine d'utilisation du réseau et sur sa fonction.

Des travaux théoriques ont été menés pour tenter de préciser au mieux la structure du réseau de neurones (nombre de couches du réseau, nombre de cellules sur chaque couche) et les données qui seront utilisées pour l'apprentissage (nombre d'exemples, ordre de présentation des exemples), et ceci avant même l'apprentissage. Les derniers en date sont les travaux de Vapnik [63]. Ils ont suscité de nouvelles recherches et il ne fait nul doute que cet axe de travail sera largement développé dans les années qui viennent.

L'autre approche consiste à intégrer dans l'architecture même du réseau de neurones les connaissances sur le domaine, de manière à commencer l'apprentissage sur une structure qui n'ait pas été initialisée au hasard. C'est ce que fait dans le domaine de l'IA un chercheur comme Towell [60] quand il traduit sous forme de réseaux multi-couches des règles expertes avant de faire de l'apprentissage sur le réseau obtenu.

D'autres méthodes tentent en classification de construire des réseaux de neurones à partir d'arbres de classification obtenus par les méthodes classiques, ou d'utiliser pour une tâche donnée un réseau déjà entraîné sur une autre tâche mais sur le même domaine. Le but est donc toujours le même : essayer de partir d'un réseau de neurones qui aura déjà intégré dans sa structure des données sur le domaine d'application.

1.4 Classes d'applications des RNA

L'identification est une procédure qui précise les tâches que peuvent accomplir les RNA. Elle est toutefois rendue difficile par la variété des modèles et du vocabulaire employé. On distingue cinq classes d'applications : l'approximation de fonctions, la compression de données, le regroupement et la quantification, l'auto-organisation et l'optimisation.

1.4.1 Approximation de fonctions

Certains RNA, en particulier les RNA multi-couches, montrent des capacités d'approximation de fonctions très intéressantes. D'autre part, l'approximation de fonctions est un cadre théorique pour des applications concrètes, comme :

- **la classification :** dans le cas des RNA, en s'intéresse essentiellement à la classification d'objets pouvant être décrits par des vecteurs de caractéristiques numériques et booléennes (la classification d'objets structurés, au sens des représentations de connaissances par objets [23]). Ce type de classification est aussi fréquemment appelé *reconnaissance des formes*, tandis qu'une sous-classe très importante de la classification est le *diagnostic*, médical ou industriel par exemple.
- **le contrôle :** il s'agit ici de construire un contrôleur pour un appareil donné. Le contrôleur reçoit des informations de l'appareil, et doit fournir des commandes qui modifient le fonctionnement de l'appareil de façon à obtenir une performance maximum par rapport à certains objectifs.

1.4.2 Compression de données

Il s'agit ici d'obtenir une représentation compacte d'un ensemble de vecteurs. Une utilisation particulière des RNA multi-couches permet de compresser des images (c'est la technique de Cottrel/Munro/Zipser [35], dans laquelle le réseau de neurones apprend à approximer la fonction identité, et les valeurs d'activités de la couche cachée constituant la représentation compacte cherchée).

1.4.3 Regroupement et quantification

Le regroupement (*clustering*) consiste à faire une partition d'un ensemble d'objets (bien sûr décrits par des vecteurs) en un certain nombre de groupes (*clusters*). Chaque groupe peut être étiqueté par un symbole ou un nombre. Le fait que des vecteurs de

caractéristiques continues peuvent ainsi être étiquetés par une quantité discrète justifie l'autre dénomination de cette tâche : la quantification (*quantization*).

1.4.4 Auto-organisation des cartes de Kohonen

Il s'agit d'associer à un nombre fini de vecteurs d'entrée un nombre fini de vecteurs représentants, appartenant à un espace de dimension inférieure à celle de l'espace d'entrée, de telle sorte que des vecteurs voisins dans l'espace d'entrée se trouvent associés à des vecteurs représentants voisins. Il y a donc en même temps, regroupement, réduction de dimensionnalité, préservation de propriétés topologiques, etc. Ces méthodes sont énormément utilisées par des chercheurs qui ont développés un grand savoir-faire [15].

1.4.5 Optimisation

Certains RNA (comme ceux de type Hopfield [66]) permettent de résoudre une certaine classe de problèmes d'optimisation (par exemple le problème classique du voyageur de commerce). D'autre part, comme l'approximation, l'optimisation est un cadre théorique très général, et on peut définir des sous-classes plus précises, la plus importante étant celles des *mémoires associatives*. Une mémoire associative est une mémoire qui permet de retrouver les éléments qu'elle stocke à partir d'une description partielle ou bruitée de ces éléments. Ce type de mémoire est adressable par son contenu, contrairement aux mémoires traditionnellement utilisées en informatique. Il s'agit d'un des thèmes de recherche les plus importants dans le domaine des réseaux de neurones, car ces mémoires peuvent servir à modéliser l'une des caractéristiques importantes de la mémoire humaine [37].

Cette hiérarchie montre que les possibilités d'application des réseaux de neurones artificiels sont nombreuses, il s'agit souvent d'applications auxquelles s'est intéressée l'intelligence artificielle, comme par exemple le traitement automatique des langues ou la reconnaissance des chiffres manuscrits [3].

1.5 Points forts et points faibles des RNA

Les types de modèles neuronaux que nous venons de décrire dans la section 1.2 présentent des avantages et des inconvénients qui vont permettre d'illustrer l'analyse des points forts et faibles des réseaux de neurones artificiels.

1.5.1 Points forts

Les réseaux de neurones artificiels possèdent des avantages qui les rendent plus attractifs que d'autres algorithmes. Leurs points forts sont : le parallélisme massif, la robustesse, l'apprentissage et la dégradation progressive.

Le parallélisme massif

Une des caractéristiques importantes des RNA est leur parallélisme massif, qui permet leur réalisation sur machine parallèle généraliste ou dédiée (par exemple un circuit intégré). Dans ce cas, la réalisation est facilitée par le fait que les réseaux de neurones sont également composés d'un grand nombre d'unités simples souvent identiques. Par exemple le modèle COLD, proposé par Azcarraga [3] pour la reconnaissance des chiffres manuscrits, est composé de 3338 unités toutes identiques. Ce parallélisme intrinsèque permet des exécutions très rapides, et cela est très intéressant dans toutes les applications temps réel, notamment la robotique, d'autant plus que la possibilité de réaliser des RNA en circuits intégrés permet de les embarquer facilement.

La robustesse

Une deuxième caractéristique intéressante des réseaux de neurones artificiels est leur robustesse. Par robustesse, on entend généralement :

- *la résistance aux pannes des neurones :* dans les RNA distribués, la redondance des informations peut conduire le réseau à bien fonctionner même quand des unités sont en pannes. Par exemple, Belala [6] distribue des règles sous forme de réseau de neurones, selon différentes méthodes, et montre que dans certains cas on peut obtenir de bonnes réponses avec 40 % des unités en panne. Il s'agit là aussi d'un avantage intéressant pour les systèmes embarqués.
- *la résistance au bruit :* de nombreux modèles neuronaux donnent de bons résultats quand leurs entrées sont bruitées. Par exemple, Azcarraga [3] montre que son classifieur neuronal COLD bien au bruit : ce classifieur classe des images de chiffres manuscrits bruitées aléatoirement par inversion de pixels de 0 à 1 ou de 1 à 0. Le niveau de bruit $B \in [0, 1]$ est la probabilité d'inversion de chaque pixel.

L'apprentissage

Un troisième point fort intéressant comme alternative à l'acquisition des connaissances est l'apprentissage. Par rapport aux problèmes posés par les applications, les réseaux de neurones artificiels présentent l'intérêt de pouvoir prendre en compte la non-linéarité (les formes d'activation sont non-linéaires en général), et dans certains cas de pouvoir prendre en compte le temps (les RNA récurrents).

La dégradation progressive

Enfin les RNA, de par leur nature continue, ne fonctionnent pas en tout ou rien, et leurs performances ont plutôt tendance à diminuer progressivement en cas de problème (bruit, panne, entrée inconnue), si bien que l'on parle souvent de dégradation progressive (*graceful degradation*). Cette propriété est très recherchée car les systèmes cognitifs vivants montrent une telle faculté [30].

1.5.2 Points faibles

Malgré ces propriétés intéressantes, que l'on peut d'ailleurs mettre en parallèle avec les difficultés des systèmes symboliques, les RNA présentent, en l'état actuel des recherches, plusieurs points faibles : manque de transparence pour l'utilisateur, sensibilité aux symboles et structures de symboles, nombre d'unités d'entrée et de sortie fixé, difficultés dans leur construction et leur entraînement à partir d'exemples.

Manque de transparence pour l'utilisateur

Les poids des connexions d'un RNA n'ont en général pas de signification évidente, si bien qu'un réseau de neurones apparaît souvent comme une boite noir [44]. Il n'est donc pas aisé d'expliquer (au sens des systèmes symboliques) le résultat produit par un réseau de neurones. De plus, si l'on peut montrer que certains RNA sont capables de construire des représentations plus au moins structurées de leur environnement, ces représentations restent opaques, et doivent être analysées avec des outils statistiques.

Sensibilité aux symboles et structures de symboles

Les réseaux de neurones artificiels ne sont pas faits pour travailler naturellement avec des symboles et des structures de symboles, ce qui les rend difficilement utilisables pour traiter des problèmes de haut niveau qui exigent souvent de telles structures (listes, frames, objets, etc.). Des solutions limitées existent toutefois : par exemple Belala [6] présente une méthode d'unification connexionniste, mais celle-ci est de peu d'intérêt pratique.

Nombre fixé d'unités d'entrée et de sortie

Un réseau de neurones artificiels donné fonctionne avec un nombre fixé d'unités d'entrée et de sortie. Cela pose d'importants problèmes de représentation et de codage quand le nombre d'unités d'entrée et de sortie est variable, ce qui arrive fréquemment dans les problèmes réels. Par exemple, dans le système SHADE [47], les médecins effectuent des mesures sur des segments de nerfs, mais le nombre de segments peut être différent selon le patient.

Difficultés dans leur construction

Les réseaux de neurones artificiels restent difficiles à composer et à combiner pour résoudre des problèmes complexes : il n'existe pas encore de bonnes méthodes pour construire et entraîner de grands réseaux, ainsi que des réseaux de réseaux.

Difficultés dans leur apprentissage à partir d'exemples

L'apprentissage à partir d'exemples n'est pas une tâche facile : en effet, le problème de l'acquisition des connaissances n'est pas complètement résolu mais plutôt déplacé, car en pratique obtenir un jeu d'exemples de bonne qualité demande beaucoup de travail.

1.6 Conclusion

Dans ce chapitre nous avons présenté les principaux modèles de réseaux de neurones artificiels, qui sont répartis en deux grandes classes : Les RNA unidirectionnels et les RNA récurrents. D'autres classes particulières de réseaux de neurones artificiels sont aussi présentées dans ce chapitre telles que les réseaux cellulaires, localistes, probabilistes, modulaires, neuro-flous, et neurobiologiquement plausibles. Ces classes de réseaux sont liées à des domaines d'applications bien particuliers.

La seconde contribution de ce chapitre est une synthèse des méthodes d'apprentissage des RNA, qui sont : les apprentissages supervisés au cours desquels on impose au RNA de donner des réponses connues, les apprentissages dits non-supervisés au cours desquels le réseau se contente de traiter ses données d'entrées en fonction de ses seuls mécanismes internes, les apprentissages par renforcement qui est une classe très particulière située à mi-chemin entre l'apprentissage supervisé et l'apprentissage non-supervisé, et les apprentissages par compilation de connaissances qui consiste à initialiser le réseau construit aussi près que possible de l'optimum recherché, avant même d'appliquer un algorithme d'apprentissage.

Une autre contribution de ce chapitre est l'identification des tâches que peuvent accomplir les RNA. Pour cela, cinq classes d'applications sont à distinguer : l'approximation de fonctions, la compression de données, le regroupement et la quantification, l'auto-organisation et l'optimisation.

Enfin, une analyse des points forts et faibles des réseaux de neurones apparaît explicitement dans ce chapitre. Nous en donnons ci-dessous une synthèse globale.

D'inspiration neurobiologique, les réseaux de neurones artificiels se caractérisent par une bonne adéquation algorithme et architecture matérielle. Ils sont utilisés dans l'approximation de fonctions pour la classification et le contrôle, la compression de données, le regroupement et la quantification, l'analyse de données, l'optimisation, etc. Les points forts d'un réseau de neurones artificiels idéal s'ajoutent à sa puissance potentielle de "super-modèle" : la redondance et la résistance aux pannes, la dégradation progressive et la résistance au bruit, l'apprentissage comme alternative au problème d'acquisition des connaissances, la prise en compte des non-linéarités et l'adéquation aux problèmes réels, le parallélisme massif et la rapidité des calculs.

La plupart des modèles neuronaux n'ont pas toutes ces qualités, par contre sont presque tous concernés par les points faibles suivants : manque de transparence et d'explication, mauvaise adaptation au traitement et à la représentation de structures de symboles, inadaptation à la résolution de problèmes de haut-niveau, nombre d'unités d'entrée et de sortie fixé, problèmes de représentation et de codage des données, faible modularité et difficulté de conception de réseaux de neurones artificiels de grande taille.

Chapitre 2

Systèmes experts symboliques

Le terme Intelligence Artificielle (IA) a été introduit par McCarthy en 1956 pendant une conférence de presse au Dartmouth collège, et depuis, ce terme a été retenu pour représenter le domaine. Parmi les domaines d'étude de l'IA, on cite :

- Vision par ordinateur : reconnaissance de formes, classification d'images, conduite autonome de véhicules et reconnaissance de textes.
- Robotique intelligente et évolution artificielle : systèmes réactifs, robots intelligents, robotique autonome, contrôle et planification de trajectoires.
- Traitement du langage naturel : traduction automatique, analyse automatique de textes, reconnaissance de la parole et recherche automatique d'informations.
- Raisonnement automatique et acquisition de connaissances : systèmes intelligents pour l'exploitation de données, systèmes d'aide au diagnostic, système pour la classification et/ou la prévision, systèmes d'aide à la décision, systèmes à base de connaissances et systèmes de raisonnement fondé sur des cas.
- Modélisation cognitive : analyse et validation de modèles du comportement cognitif humain (modèles fondés sur des études psychologiques).
- IA distribuée et systèmes multi-agents : modélisation et conception de systèmes fondés sur des interactions avec multiples modules, théorie des jeux et agents coopératifs et collaboratifs (e.g. modèles sociaux, comportements collectifs en robotique).
- Logique formelle : logiques pour l'intelligence artificielle (e.g. logique temporelle), langages d'intelligence artificielle (e.g. Prolog), preuve automatique de théorèmes.

D'une façon générale, on peut dire que, "le but de l'IA est de construire des systèmes qui puissent exhiber un comportement intelligent et réaliser des tâches complexes avec un niveau de compétence qui est équivalent, sinon supérieur, à celui des experts humains" [49]. Quand les recherches en IA ont démarré, un grand effort a été déployé dans la recherche d'un outil capable de résoudre n'importe quel type de problème

(*General Problem Solvers - GPS*). Ce type d'approche s'est avéré trop ambitieux, car on n'est pas capable de créer un seul et unique système qui puisse traiter en même temps toute une large gamme de problèmes complexes. En conséquences de cet échec, les chercheurs ont conclu que des systèmes spécialisés utilisés dans la résolution de certains problèmes pourraient être développés plus facilement, et quand même être appliqués dans différents domaines (les connaissances du domaine changent, mais les mécanismes de base restent à peu près les mêmes). Au lieu de chercher une seule et unique solution générale omnipotente de résolution de problèmes, il vaut mieux alors disposer d'outils spécialisés dans la résolution de tâches plus spécifiques.

Des systèmes spécialisés dans la résolution de problèmes plus spécifiques sont donc apparus. Ces systèmes ont été nommés des *systèmes experts symboliques (SES)*. Ce type de système appartient à la classe des *systèmes à base de connaissances*.

2.1 Architecture d'un SES

Feigenbaum [25] a défini un système expert symbolique comme étant "un logiciel intelligent qui utilise des connaissances et un procédé d'inférence pour résoudre des problèmes. Ces problèmes sont assez difficiles à résoudre et requièrent une expertise humaine significative pour arriver à une solution. Les connaissances d'un système expert symbolique sont composées de faits et des heuristiques".

Cette définition est d'une façon générale encore valable, comme on pourra le constater dans la suite de ce chapitre. Pourtant, elle est un peu dépassée quand on fait référence aux systèmes experts de deuxième génération. Ce deuxième type de systèmes est plus développé et possède des propriétés telles que l'intégration de différentes méthodes de raisonnement et l'automatisation de l'acquisition de connaissances [49].

Donc, les systèmes experts symboliques sont destinés à résoudre des problèmes dans un domaine spécifique d'expertise et ne sont pas des outils généraux de résolution de problèmes. Actuellement, les systèmes experts symboliques couvrent un large domaine d'application, dans lequel on peut distinguer deux principales applications : les systèmes d'aide à la décision et les systèmes d'aide au diagnostic. Ces systèmes doivent être conçus pour aider l'homme dans la réalisation de tâches difficiles et ne doivent jamais être utilisés pour le remplacer complètement. Ils ont des caractéristiques particulières qui les distinguent des autres systèmes informatiques, telles que :

- l'architecture particulière de leurs modules,
- la codification des connaissances dans une base de connaissances,
- la séparation des connaissances de la partie contrôle,

- la capacité à raisonner sur des données incomplètes, inexactes et floues (sur certains systèmes experts symboliques),
- les outils d'acquisition de connaissances,
- la possibilité d'explication des résultats obtenus.

Ces capacités sont principalement dues au fait que ces systèmes sont capables de dériver des solutions à partir d'une heuristique, plutôt que par les approches algorithmiques employées dans les systèmes informatiques conventionnels.

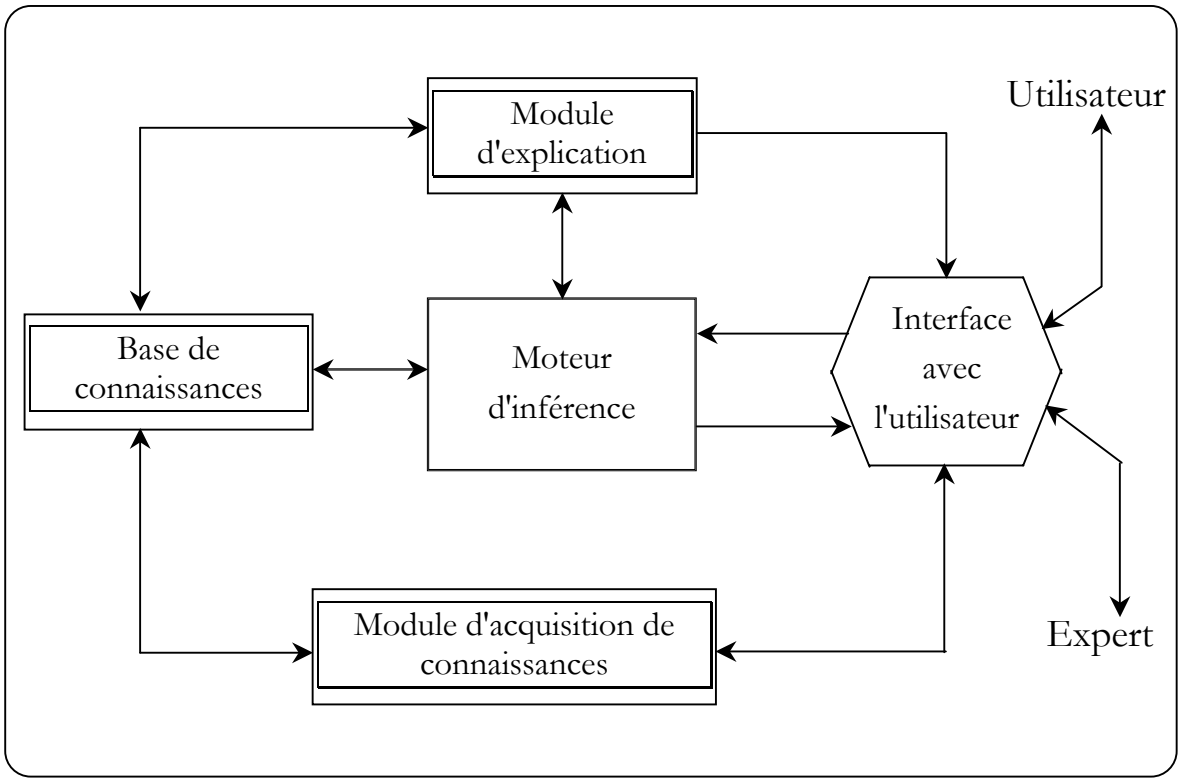

Figure 2.1 : Architecture d'un système expert symbolique.

Un système expert symbolique est en général composé d'une architecture à cinq modules (figure 2.1) : une base de connaissances, un moteur d'inférence, un module d'acquisition de connaissances, un module d'explication et une interface avec l'utilisateur.

Base de connaissances

La base de connaissances contient les données spécifiques du domaine qui sont utilisées pour résoudre un certain problème. Elle est composée usuellement de règles et de faits.

Moteur d'inférence

Le moteur d'inférence permet de manipuler les données stockées dans la base de connaissances de façon à pouvoir résoudre les problèmes posés au système. Le moteur d'inférence reste séparé des connaissances du domaine et généralement il n'est pas dépendant de son application à un problème particulier. Cela ne veut pas dire que l'on a un seul type de moteur d'inférence, car chaque formalisme de représentation possède ses propres techniques d'inférence et donc, le mécanisme d'inférence reste directement lié au type de représentation des connaissances employé. Un moteur d'inférence est typiquement fondé sur une règle d'inférence (permet de réaliser un raisonnement et de déduire de nouvelles connaissances à partir de la base) et une stratégie de recherche.

Module d'acquisition des connaissances

Ce module permet de créer, ajouter et maintenir les connaissances nécessaires pour la résolution d'un problème dans un domaine d'application. L'acquisition de connaissances peut être faite par explicitation de connaissances d'un domaine avec l'aide d'un ingénieur de connaissances et de l'expert, ou à travers des processus semi-automatiques ou totalement automatiques d'acquisition de connaissances. Ces méthodes automatiques sont connues comme *méthodes d'apprentissage automatique (machine learning methods)* [49]. La figure 2.2 présente un schéma simplifié du processus d'acquisition de connaissances.

Module d'explication

Ce module permet à l'utilisateur de connaître les processus et le raisonnement qui ont amenés le système à donner une certaine réponse. De cette façon, il est possible à l'utilisateur de poser des questions et interagir avec le système afin de comprendre les réponses fournies. Le système est capable de justifier ses réponses.

Interface avec l'utilisateur

L'interface avec l'utilisateur permet l'échange d'informations entre l'utilisateur, l'expert, l'ingénieur de connaissances et les différents modules du système expert symbolique.

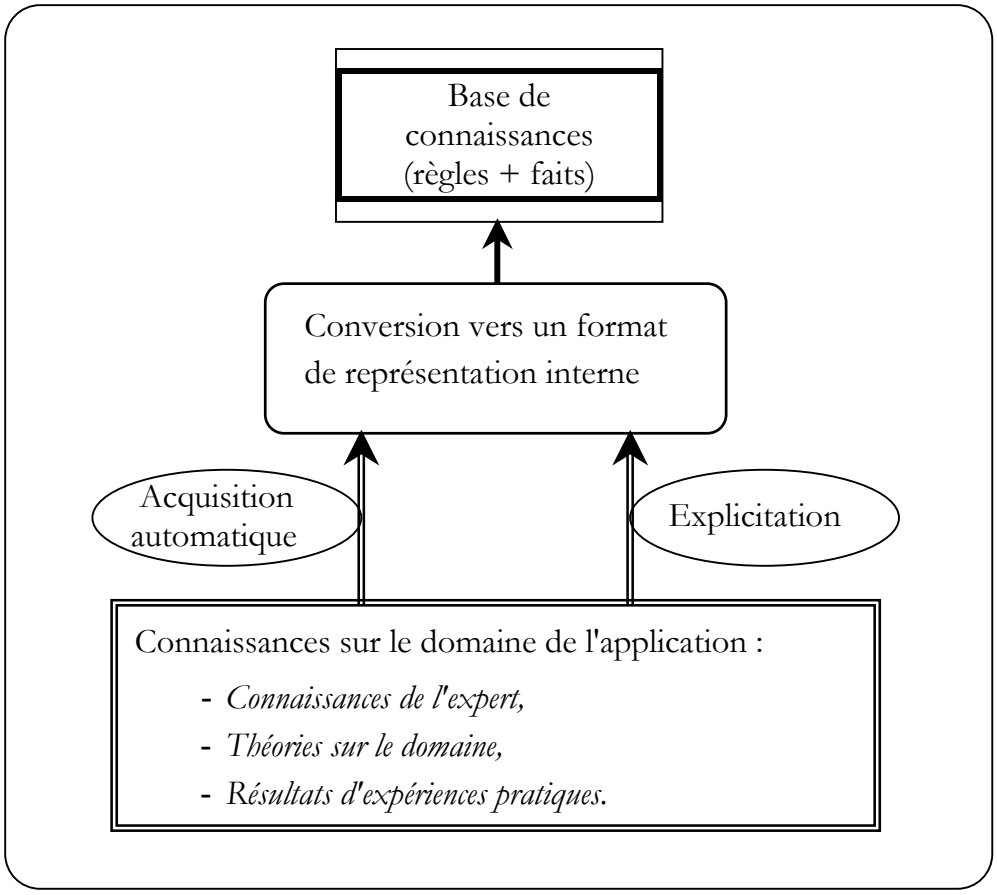

Figure 2.2 : Acquisition de connaissances dans un SES.

Les systèmes experts symboliques permettent d'automatiser certains aspects du raisonnement humain, en utilisant des méthodes d'inférence telles que l'induction, la déduction et le raisonnement par analogie. Certains types de moteurs d'inférence sont très connus, comme ceux basés sur la méthode des chaînages avant et/ou arrière.

Dans les méthodes de chaînage avant, c'est à partir des prémisses (faits) qu'on arrive aux conclusions. Un exemple de système exploitant ce type de méthode d'inférence est le langage CLIPS (*C Language Integrated Production System*) [11], et il sert à la construction des systèmes experts symboliques. Par contre, la méthode de chaînage arrière part des conclusions et cherche à trouver les prémisses nécessaires à leur déclenchement. Un exemple d'outil informatique utilisé pour la construction de systèmes experts symboliques est le langage Prolog (Programmation en Logique) [13]. Ce langage utilise une méthode de chaînage arrière dans l'implémentation du moteur d'inférence par une méthode de résolution. Ce type de méthode automatique de raisonnement, employé par les langages comme Prolog, repose sur l'idée que "tout ce qui n'est pas vrai est faux et tout ce qui n'est pas faux est vrai". C'est pour cela que de tels systèmes ont besoin d'une théorie (base de règles et faits) qui soit à la fois correcte et complète sur un domaine.

2.2 Représentation des connaissances

De nombreuses activités "intelligentes" de l'être humain, aussi bien dans sa vie quotidienne que dans son activité professionnelle, reposent sur l'exploitation d'une masse importante d'informations, de faits, d'expériences et de connaissances plus ou moins spécifiques d'un domaine particulier. Nous avons montré dans la figure 2.2, que les connaissances utilisées dans un système expert symbolique appliqué à un domaine spécifique peuvent être obtenues à partir de différentes sources, à savoir :

- l'explicitation directe par l'expert de ces connaissances ;
- l'interrogation de l'expert de façon à *extraire* ses connaissances ;
- l'analyse et la représentation des connaissances à partir de livres, manuels, etc. ;
- l'observation des résultats obtenus dans des expériences pratiques ;
- l'emploi d'outils d'analyse, de traitement et de manipulation automatique de connaissances qui permettent d'induire ou de déduire de nouvelles connaissances.

On peut classer selon leur source les connaissances d'un système expert symbolique en deux grands groupes principaux : les connaissances empiriques et les connaissances théoriques.

Les connaissances empiriques

Les connaissances empiriques (connaissances expérimentales), sont représentées par l'ensemble des cas pratiques observés sur un sujet (ensemble d'exemples). Ce sont des connaissances "pures" qui n'ont pas été traitées, analysées ou modifiées. Ces connaissances représentent les résultats d'expériences ou les exemples de cas pratiques ; elles n'ont pas encore subi de transformations en vue d'obtenir une théorie plus générale sur le domaine. On peut dire que ce sont des connaissances de bas niveau. Ce type de connaissances peut être utilisé dans les systèmes experts symboliques à travers l'application de processus de raisonnement par induction (obtention de connaissances de plus haut niveau à partir des connaissances empiriques de base - la généralisation), ou à travers des processus de raisonnement par analogie (rapprochement des cas nouveaux aux cas anciens connus les plus proches - la similarité).

Les connaissances théoriques

Les connaissances théoriques modélisent les connaissances sur un sujet à l'aide d'une théorie correspondant au problème posé. Elles sont des connaissances "traitées" qui ont été obtenues à partir de l'analyse des connaissances de base. Ce type de connaissances

représente une généralisation du savoir. Ce sont des connaissances dites de haut niveau. Elles sont habituellement représentées par des structures symboliques, des objets structurés et des propositions.

Les connaissances empiriques et les connaissances théoriques doivent être converties dans une représentation compatible avec les outils informatiques de raisonnement automatique afin de pouvoir être exploitées. Le "savoir-que" d'un expert dans un domaine donné sera formalisé par un ensemble de propositions de même forme syntaxique, pouvant être ensuite manipulées selon des règles de la logique classique (règles d'inférence).

Il existe différents modes de représentation des connaissances proposés dans le cadre des approches de l'IA [49], à savoir : description de cas pratiques, règles de production et formules logiques, réseaux sémantiques, objets structurés et frames, et méta-connaissances.

2.2.1 Description de cas pratiques

C'est la forme de représentation de connaissances la plus simple. Les connaissances sont décrites par une liste d'attributs et ses respectives valeurs associées, tel qu'on les a observé dans une expérience pratique. Le simple fait de sélectionner quelques attributs spécifiques et d'associer une classification au cas peut donner l'origine à une information plus structurée comme les règles de production.

Exemple : Age = 38 ans.

2.2.2 Règles de production et formules logiques

Les propositions, originellement en langage naturel, doivent être converties dans une syntaxe bien précise, pouvant donner ainsi l'origine à des règles de production et des formules logiques.
Une règle de production est un quantum de connaissance, déclaratif et autonome, de la forme générale :

Si *[conditions]* ***Alors*** *[conclusion]*

La partie *conditions* (antécédent) est constituée d'une formule logique qui doit être vérifiée pour que la règle s'applique. La partie *conclusion* (conséquent) correspond habituellement au déclenchement d'une action ou d'une autre règle. Les formules peuvent être des types : atomique, moléculaire ou généralisé.

Les formules atomiques

Les formules atomiques utilisent des symboles atomiques et des relations unaires, binaires, ternaires, etc.
Exemples : Froide(neige) ; Couleur(neige, blanche) ; Lancer(philippe, neige, denise).

Les formules moléculaires

Les formules moléculaires peuvent être définies à partir des formules atomiques combinées à l'aide de connecteurs logiques tels que '$\wedge$' (ET logique), '$\vee$' (OU logique), '$\neg$' (NON logique).
Exemple : Froide(neige) $\wedge$ Couleur(neige, blanche).

Les formules généralisées

Les formules généralisées peuvent être construites à partir des formules moléculaires. Il s'agit tout d'abord de substituer à des constantes (symboles atomiques) d'une formule moléculaire par des variables. Nous obtenons ainsi des fonctions sur les formules. Ces fonctions sont représentées par des quantificateurs tels que '$\exists$x' (il existe au moins une valeur possible pour la variable x), '$\forall$x' (pour toutes les valeurs possibles que la variable x peut assumer).
Exemple : ($\exists$x) (Couleur(x, blanche)).

Les règles de production sont classées selon leur pouvoir d'expressivité en règles d'ordre 0, d'ordre 0^+, ou de premier ordre (ordre 1). Les règles d'ordre 0 contiennent uniquement des formules atomiques ou moléculaires telles que définies précédemment. Les règles d'ordre 0+ contiennent en plus des formules du type 'x$\in$I' (où 'x' est une variable décrivant une situation ou un événement, et 'I' est un ensemble de valeurs possibles, c'est-à-dire un intervalle auquel x peut s'appliquer). Les règles de premier ordre sont des règles qui peuvent contenir des formules généralisées et par conséquence des quantificateurs. Ce dernier type de règles est aussi appelé le *prédicat*.

Des règles de production et des méthodes de raisonnement automatique plus particulières ont été proposées par des chercheurs [53] afin d'augmenter le pouvoir de représentation et d'inférence des systèmes experts symboliques. On peut citer comme exemples :

- Les règles floues,
  Exemple : ***Si*** x $\in$ Fuzzy_Set_I
  Alors y = a.x + b. ('a' et 'b' sont des valeurs constantes).

- Les règles obtenues à partir des réseaux connexionnistes,
  Exemple : ***Si*** 2-parmi-4 ($F_1(a)$, $F_2(b)$, $F_3(c)$, $F_4(d)$)
  Alors conclusion_est_vraie ***Sinon*** conclusion_est_fausse.

- Les règles probabilistes et les coefficients de vraisemblance (facteurs de certitude),
  Exemple : ***Si*** possède_propriété_X ($exemple_i$)
  Alors probabilité_d'appartenir_classe_A ($exemple_i$) = Valeur1
  et probabilité_non_appartenir_classe_A ($exemple_i$) = Valeur2.

- Les règles d'inférence graduelle,
  Exemple : ***Si*** plus X est un oiseau typique
  Alors plus on est certain que X vole.

- Les règles de "haut niveau" (règles de modulation ou de contexte),
  Exemple : ***Si*** X_i est plus grand par rapport à X_j
  Alors moins important sera X_j

Ces règles de production prennent en considération le facteur d'incertitude, les types de données utilisées (données discrètes ou continues), les relations entre données et valeurs constantes (appartenance à un intervalle, supérieur, inférieur), les relations entre une donnée et une autre, et même la force de ces relations.

2.2.3 Réseaux sémantiques

Un réseau sémantique est un graphe étiqueté dans lequel les nœuds représentent des objets ou des concepts, et les arcs des relations entre eux. Issus des travaux de psychologie cognitive sur l'organisation de la mémoire (mécanismes d'association), les réseaux sémantiques ont donné lieu à de nombreux systèmes et langages de représentation de connaissances [49]. Les réseaux sémantiques constituent donc un moyen de structuration des bases de connaissances.

Grâce au mécanisme d'héritage de propriétés, un nœud d'un réseau sémantique peut hériter des propriétés des nœuds qui sont plus hauts que lui dans la hiérarchie. Ce mécanisme constitue un mode économique de raisonnement, permettant notamment à un concept particulier de posséder automatiquement les propriétés des concepts plus généraux dont il est issu. On retrouve ce mécanisme dans les représentations par objets [23], actuellement beaucoup utilisées que les réseaux sémantiques.

2.2.4 Objets structurés et frames

Les représentations à base d'objets structurés, telles que les *schémas*, les *frames*, les *scripts* ou les *scénarios*, permettent de regrouper ensemble des connaissances relatives à un objet, un concept, une situation ou un événement. A partir des travaux de psychologie cognitive sur l'organisation de la mémoire chez l'homme, la notion de schéma a été proposée comme modèle de représentation prototypique d'expériences passées mises à profit pour résoudre un problème nouveau. Ce concept a été repris en IA sous la forme de *frames*, et de *scripts* ou *scénarios*. Un exemple de langage basé sur les *frames* est le langage Shirka [54] pour la gestion de bases de connaissances centreés-objet.

Frames et scénarios sont des entités de granularité importante regroupant de façon structurée l'ensemble de connaissances relatives à un objet, un concept ou une situation typique. Un *frame* est composé d'un ensemble d'attributs (*slots*) correspondant aux diverses notions relatives au concept représenté. A titre d'exemple, la figure 2.3 donne une description possible du concept de vin [49].

```
< frame vin :
   sorte de : boisson
   appellation : (domaine : AOC/DQ/vin de pays)
                 (défaut : vin de pays)
                 (si besoin : demande appellation)
   nature : (domaine : sec/demi-sec/doux/liquoreux/corsé)
   degré alcool : (intervalle : 9 à 15)
                  (défaut : 12)
   producteur : (si besoin : trouver nom sur étiquette)
   … >
```

Figure 2.3 : Un frame décrivant le concept de vin.

2.2.5 Méta-connaissances

Une forme importante de connaissances est la méta-connaissance (connaissance sur la connaissance). Elle correspond au recul que l'on prend par rapport à un certain domaine d'activité et représente de ce fait souvent une part notable de l'expertise humaine. Elle intervient souvent, car elle est liée à la façon d'utiliser un ensemble de connaissances, aux stratégies de raisonnement et aussi à l'acquisition de nouvelles connaissances. Ce type de connaissances peut être représenté par des structures comme celles décrites dans les items ci-dessus ; cependant il mérite d'être classé dans un groupe à part, du fait de ses caractéristiques très particulières.

2.3 Acquisition des connaissances

Le processus de développement d'un système expert symbolique se compose d'une étape d'acquisition des connaissances appelée *ingénierie de connaissances*. Elle est réalisée par l'ingénieur de connaissances, le responsable de l'obtention, de la vérification, de la validation et de la structuration (codification) des connaissances.

L'acquisition de connaissances est un processus de rassemblement des connaissances nécessaires à la résolution d'un problème lié à une application spécifique. Ce processus inclut la codification des connaissances dans un formalisme adapté aux outils informatiques utilisés. Il y a plusieurs façons de coder les connaissances afin de créer la base de connaissances, comme on l'a vu dans la section 2.2. En ce qui concerne l'obtention des connaissances qui vont être placées dans cette base, on peut diviser les processus d'acquisition de connaissances en deux grands groupes : obtention par explicitation de connaissances et obtention par apprentissage automatique ou semi-automatique [49].

L'explicitation de connaissances est réalisée par l'ingénieur de connaissances à partir d'interviews avec des experts et d'informations disponibles dans des livres, manuels, rapports, etc. Cette approche est très intéressante, puisqu'on profite des connaissances déjà acquises et bien élaborées. Malheureusement, l'explicitation de connaissances présente un certain nombre d'inconvénients :

- Le processus d'extraction de connaissances d'un expert est extrêmement ennuyeux, il prend beaucoup de temps et sa mise en place présente de sérieuses difficultés.
- Les experts humains utilisent souvent des activités mentales d'un niveau "subcognitif" pour résoudre les problèmes : ils ne savent pas verbaliser ce type de connaissances.
- Parfois les experts ne sont même pas conscients d'avoir utilisé certaines connaissances dans la résolution d'un problème et par conséquent ne les explicitent pas.

Par conséquent, les chercheurs du domaine de l'IA ont beaucoup investi dans l'étude et l'implémentation de systèmes d'acquisition automatique de connaissances. Les recherches sur l'acquisition automatique de connaissances sont à l'origine des méthodes d'apprentissage automatique, l'un des domaines de l'IA en plus grande expansion actuellement [49].

L'apprentissage automatique est une partie très importante de l'IA et doit être une des principales caractéristiques des systèmes intelligents. Une des meilleures définitions de l'apprentissage automatique est celle donnée par Simon [57] : "l'apprentissage dans un système est indiqué par les changements qu'il subit. Ces changements sont adaptatifs dans la mesure où ils rendent possible au système de réaliser une même tâche, ou des tâches tirées d'une même population, d'une façon plus efficace et plus efficiente la prochaine fois qu'elle sera réalisée". Donc, l'apprentissage automatique se fait par des outils qui permettent d'acquérir, élargir et améliorer les connaissances disponibles au système. En général, l'apprentissage implique des processus d'adaptation et de modification des structures de contrôle et/ou de représentation de connaissances du système en question.

Les méthodes d'apprentissage automatique sont classées en deux groupes : les méthodes d'apprentissage empirique (*empirical learning*) et les méthodes d'apprentissage fondées sur l'explication (*explanation based learning*) [61].

2.3.1 Méthodes d'apprentissage empirique

Les méthodes d'apprentissage empirique sont fondées sur l'acquisition de connaissances à partir d'exemples. Elles incluent aussi des méthodes connues sous les noms : *instance based learning* ou *exampled based learning*. On peut citer comme exemples de méthodes d'apprentissage empirique les techniques suivantes [49] : le raisonnement fondé sur des cas, les méthodes de construction de concepts et prototypes, les réseaux de neurones artificiels, les arbres de décision, les algorithmes génétiques d'induction de règles. Ces méthodes se divisent entre les méthodes d'apprentissage par analogie et les méthodes d'apprentissage par induction.

Apprentissage par analogie

Les approches fondées sur l'analogie essayent de faire le transfert des connaissances d'une tâche bien connue vers une autre moins connue. Ainsi, il est possible d'apprendre des nouveaux concepts ou de dériver des nouvelles solutions à partir de concepts et solutions similaires connues. Ainsi, deux notions deviennent très importantes dans la définition de l'apprentissage par analogie : Le *transfert* et la *similarité*. L'apprentissage par analogie est aussi appelé *apprentissage fondé sur la similarité*.

Les approches empiriques fondées sur l'analogie, telles que les systèmes de raisonnement à base de cas (CBR), peuvent être considérées plutôt comme des méthodes de raisonnement que comme des méthodes d'apprentissage. Certains de ces

systèmes implémentent uniquement des fonctions de calcul de similarité, sans utiliser de mécanismes d'adaptation.

Apprentissage par induction

L'apprentissage par induction reste toujours une des principales méthodes étudiées dans le domaine de l'apprentissage automatique. Dans cette approche, on cherche à acquérir des règles générales qui représentent les connaissances obtenues à partir d'exemples. Les règles ainsi obtenues peuvent être représentées d'une façon explicite (facilement interprétables) ou d'une façon implicite avec un codage qui n'est toujours pas facile à interpréter.

L'algorithme d'apprentissage par induction doit produire des règles de classification permettant de classer les nouveaux exemples. Le processus d'apprentissage cherche à créer une représentation plus générale des exemples selon une méthode de généralisation de connaissances. Ce type de méthodes est aussi appelé apprentissage de concepts ou bien acquisition de concepts. Parmi les approches d'apprentissage empirique par induction les plus connues, on trouve les réseaux de neurones et les arbres de décision. L'algorithme d'apprentissage par induction peut fonctionner de façon supervisée ou non-supervisée.

- Apprentissage supervisé (*supervised learning*) : les exemples d'apprentissage sont étiquetés afin d'identifier la classe à laquelle ils appartiennent. Le but de l'algorithme de classification est de classifier correctement les nouveaux exemples dans les classes définies dans la phase d'apprentissage.
- Apprentissage non-supervisé (*unsupervised learning*) : l'algorithme d'apprentissage cherche à trouver des régularités dans une collection d'exemples, puisque dans ce type d'apprentissage on ne connaît pas la classe à laquelle les exemples d'apprentissage appartiennent. Une technique employée consiste à implémenter des algorithmes pour approcher les exemples les plus similaires et éloigner ceux qui ont le moins de caractéristiques communes. Ces groupes d'exemples similaires sont parfois appelés des *prototypes*.

2.3.2 Méthodes d'apprentissage fondées sur l'explication

Les méthodes inductives, telles que les réseaux de neurones ou les arbres de décision, ont besoin d'un nombre significatif d'exemples pour pouvoir bien généraliser les connaissances (induire des règles ou des concepts). Ceci restreint les possibilités d'application de ces méthodes, puisqu'on n'a pas toujours une base d'exemples assez grande et complète sur le domaine traité. Les méthodes d'apprentissage fondées sur l'explication (*Explanation Based Learning - EBL*) utilisent des connaissances préexistantes et un raisonnement déductif pour augmenter l'information fourni par des ensembles d'exemples. Ces méthodes son connues sous le nom d'apprentissage par analyse (a*nalytical learning*) [46].

Dans les méthodes EBL, les connaissances sont dérivées à partir d'un simple cas par explication des raisons pour lesquelles il représente un exemple du concept appris. La méthode EBL utilise les connaissances préexistantes pour analyser, ou expliquer, comment chaque exemple observé lors de l'apprentissage satisfait les concepts existants. Ensuite, cette explication est utilisée pour différencier les attributs pertinents de l'exemple d'apprentissage de ceux qui ne le sont pas. De cette façon, l'exemple pourra être généralisé par un raisonnement logique, à la place des raisonnements statistiques souvent utilisés par les autres méthodes. Ces méthodes servent donc à améliorer les performances du système, grâce à des traitements qui rendent l'utilisation des connaissances plus efficaces.

2.4 Points forts et points faibles des SES

Les systèmes experts symboliques ont plusieurs points forts du point de vue informatique et cognitif qui expliquent leur succès, mais aussi des points faibles qui limitent leurs capacités à prendre en charge des problèmes complexes.

2.4.1 Les points forts

Les systèmes experts symboliques possèdent plusieurs points forts tels que : capacité d'explication, séparation des connaissances du moteur d'inférence, nature déclarative des connaissances, connaissances fortement structurées, la productivité, la systématicité, et la compositionnalité.

Capacité d'explication

Les systèmes experts symboliques ont la capacité d'explication au moins potentielle des conclusions avancées, même si dans la pratique il reste difficile de dépasser le stade de la trace du raisonnement.

Séparation des connaissances du moteur d'inférence

La séparation des connaissances du système qui les exploite (moteur d'inférence) dans un système symbolique permet l'évolution séparée et la réutilisation de ces deux composants de base.

Nature déclarative des connaissances

Dans un système expert symbolique les connaissances sont de nature déclarative. Ceci facilite l'expression de ces connaissances.

Connaissances fortement structurées

Les systèmes experts symboliques ont un caractère fortement structuré des connaissances qu'ils permettent de représenter. Ce caractère reste pour le moment indispensable dans la plupart des applications en IA (traitement des langues).

La productivité

La productivité (*productivity*) est la capacité non bornée de représenter des propositions par un système expert symbolique, d'après Crucianu [16] : "la propriété hypothétique de la cognition d'avoir une capacité de représentation et de traitement (donc une compétence) potentiellement illimitée".

La systématicité

La systématicité (*systematicity*) est la propriété de la cognition de présenter certaines symétries [16, 32]. Il faut distinguer systématicité d'inférence (ou même de traitement) et systématicité de représentation. La systématicité d'inférence est la capacité de faire toutes les inférences d'un certain type logique, pas seulement quelques unes d'entre elles. Par exemple, un système pouvant inférer P à partir de $P \wedge Q \wedge R$ mais ne pouvant inférer P à partir de $P \wedge Q$ ne possède pas la propriété de systématicité. Par contre, la systématicité de représentation suppose qu'un système soit toujours capable de représenter la proposition P quand (1) il est capable de représenter une proposition Q, et (2) P et Q sont suffisamment similaires. Par exemple, un système symbolique possédant la systématicité de représentation et pouvant représenter *Mary aime John* devrait pouvoir aussi représenter *John aime mary*.

La compositionnalité

La compositionnalité (*compositionality*) n'est pas une propriété observable de la cognition, mais plutôt un élément d'explication des deux propriétés ci-dessus [16]. Pour avoir les propriétés en question, un système cognitif doit posséder la compositionnalité: les représentations utilisées par le système cognitif doivent avoir une structure reposant sur la composition, ce qui veut dire que les constituants d'une représentation doivent être explicitement concaténés selon des règles de combinaison strictes, et que les processus cognitifs doivent être systématiquement sensibles à la structure. Par exemple, la transformation de la représentation d'une phrase active en représentation d'une phrase passive doit être sensible à la structure de la phrase active.

2.4.2 Les points faibles

Nous avons regroupé dans cette section les principaux points faibles soulevés par les concepteurs de systèmes hybrides. Toutefois, ces diverses limites des systèmes symboliques sont bien connues, et des solutions sont en cours d'étude (par exemple, logique floue, raisonnement à base de cas, apprentissage automatique).

Outre le problème classique de l'acquisition des connaissances [7], les systèmes experts symboliques ont les points faibles suivants : non-amélioration des performances avec l'expérience, incapacité de s'adapter, faible dégradation progressive des performances, sensibilité aux données provenant de capteurs, difficulté à généraliser une solution, fragilité dans le traitement de quelques aspects du raisonnement, et le problème de l'ancrage des symboles.

Non-amélioration des performances

Pour résoudre un problème déjà rencontré, le système expert symbolique refait exactement les mêmes étapes de raisonnement. Alors qu'un expert humain mémorise et utilise la connaissance précédemment acquise [30].

Incapacité de s'adapter

En cas de changement dans l'environnement (par exemple l'amélioration d'une partie d'un appareil sujette à des pannes fréquentes) le comportement d'un système expert symbolique ne change pas (à moins de modifier sa base de connaissances), alors qu'un expert changerait immédiatement sa stratégie de diagnostic [30].

Faible dégradation progressive des performances

Lorsqu'une situation n'a pas été prédéfinie, ou lorsque les données sont incomplètes ou bruitées, un système expert symbolique échoue complètement, alors qu'un expert humain donnera quand même une décision, éventuellement moins bonne [30].

Sensibilité aux données provenant de capteurs

Les systèmes experts symboliques ont la difficulté de prendre en compte des données provenant de capteurs. De telles données se caractérisent par leur appartenance à un espace de dimension élevée et par le fait qu'elles peuvent être bruitées ou incomplètes.

Difficulté à généraliser une solution

Lorsque la solution d'un problème *A*, exprimé sous la forme de structures de symboles, est connue, il n'est pas simple pour uns système expert symbolique de donner une solution pour un problème *B* proche du problème *A*. En effet, même la notion de proximité est délicate à définir dans un système symbolique.

Fragilité des systèmes symboliques

La fragilité des systèmes symboliques est l'une de leurs plus importantes limitations. Sun [59] considère que la fragilité c'est l'incapacité à traiter, d'une manière systématique et dans un cadre unifié, les aspects suivants du raisonnement : information partielle, information incertaine, absence de règles applicables, les interactions entre règles (c'est-à-dire le manque de consistance et de complétude dans une base de règles partielle), l'héritage de propriétés, l'apprentissage de nouvelles règles et la modification des règles existantes.

Pour d'autres auteurs [17], les systèmes à base de règles tendent à être fragiles car ils ne fonctionnent pas bien en dehors de leur domaine d'expertise. En raison d'absence de connaissances, les systèmes experts symboliques tendent à donner des réponses absurdes au lieu de répondre simplement "je ne sais pas". Une autre cause de fragilité, pour ces auteurs, provient des interactions entre règles : il y a souvent des interactions négatives (conflits) entre les règles, ce qui réduit les performances. En particulier, l'ajout de nouvelles règles peut rendre le système incapable de résoudre certains problèmes qu'il pouvait résoudre avant.

Problème de l'ancrage des symboles

Le problème de l'ancrage des symboles est une autre limitation fondamentale des systèmes experts symboliques, à l'origine mise en lumière par des chercheurs en sciences cognitives mais aujourd'hui largement reprise par des informaticiens qui cherchent des solutions concrètes [38].

La définition d'un système symbolique correspond à celle d'un système formel en logique, par exemple le calcul des propositions. Dans un système de ce type, les symboles on une forme arbitraire (comme *a, b,* $\Rightarrow$*,* $\wedge$*,* …). Ils sont combinables en expressions plus complexes par des règles de construction ($a \Rightarrow b$), et sont manipulés à l'aide de règles de production ou de composition, qui ne dépendent que de la syntaxe des expressions. La sémantique des symboles et des expressions symboliques est externe, c'est-à-dire qu'elle est attribuée par un observateur extérieur au système.
Par exemple, l'expression symbolique $a \wedge b \Rightarrow c$ peut prendre comme sémantique tant *"s'il y a de la neige et que je suis en vacances, je vais faire du ski"*, que *"le jaune mélangé à du bleu donne du vert"*, mais n'a aucune signification *par elle-même* : en d'autres termes, la syntaxe ne suffit pas pour définir la sémantique.

Les symboles manipulés par un être humain ont quant à eux une sémantique *interne*. Chacun connaît la signification des mots qu'il emploie, sans qu'un système extérieur soit nécessaire pour l'attribution d'une sémantique *a posteriori*. Pour Lallement [38], une solution possible au problème de l'ancrage consiste à réaliser un système hybride neuro-symbolique, dans lequel les réseaux de neurones artificiels auraient un rôle important à jouer.

Beaucoup de recherches visent à relier les symboles d'un système symbolique au monde réel, en dotant l'ordinateur de capteurs et d'effecteurs ; pour cette raison on parle beaucoup d'*ancrage perceptif des symboles* [21]. Mais cela ne suffit pas, il faudrait passer à des systèmes d'intelligence artificielle autonomes capables de forger eux-mêmes leurs concepts et symboles.

2.5 Conclusion

Dans ce chapitre nous avons présenté les différents composants d'un système expert symbolique et nous avons décrit brièvement : la base de connaissances, le moteur d'inférence, le module d'acquisition des connaissances, le module d'explication et l'interface utilisateur.

La seconde contribution de ce chapitre est la description des méthodes de représentation des connaissances dans un système expert symbolique. Celles-ci peuvent être représentées par : des descriptions de cas pratiques, des règles de productions et formules logiques, des réseaux sémantiques, des objets structurés et frames, ou des méta-connaissances.

Une autre contribution de ce chapitre est l'étude du processus d'acquisition des connaissances qui est un processus de rassemblement des connaissances empiriques et théoriques nécessaires à la résolution d'un problème lié à une application spécifique. Ces connaissances sont obtenues de deux manières :

- par explicitation de connaissances réalisé par l'ingénieur des connaissances à partir des interviews avec des experts et d'informations disponibles dans des livres, manuels, rapports, etc. ;
- par apprentissage automatique permettant d'acquérir, élargir et améliorer les connaissances disponibles au système symbolique. Pour cela, deux groupes de méthodes d'apprentissage automatique existent : les méthodes empiriques (apprentissage par analogie, apprentissage par induction) et les méthodes fondées sur l'explication.

Enfin, une analyse des points forts et faibles des systèmes experts symboliques apparaît explicitement dans ce chapitre. Nous en donnons ci-dessous une synthèse globale.

Les systèmes experts symboliques se caractérisent par une séparation des connaissances de leur contrôle. Ils permettent la prise en compte de connaissances expertes pour résoudre des problèmes tels que le diagnostic ou la planification. Leur nature déclarative facilitant la formulation et l'explication s'adapte bien à la représentation de connaissances structurées. Du point de vue cognitif, leurs propriétés sont la productivité, la systématicité et la compositionnalité. Notons que ces points forts sont valables quel que soit le système symbolique. Leurs points faibles, outre leur fragilité sont : la difficile acquisition des connaissances ; l'absence d'amélioration des performances avec l'expérience ; la mauvaise adéquation à l'environnement réel, aux données provenant de capteurs, aux données incomplètes ou bruitées, à l'absence de règles et aux interactions entre règles ; et à l'absence d'ancrage des symboles et des concepts.

Chapitre 3

Intégration neuro-symbolique

Les systèmes hybrides tentent de tirer parti des points forts de différents paradigmes pour résoudre des problèmes jusqu'alors insolubles, pour couvrir un champ d'application plus large, ou pour obtenir des performances plus élevées.

Dans ce chapitre, nous nous consacrons à l'intégration des composants symboliques et des composants neuronaux dans des Systèmes Hybrides Neuro-Symboliques (SHNS) puissants. C'est un domaine de recherche très récent qui est en cours d'évolution et de maturation, appelé *Intégration Neuro-Symbolique (INS)*. L'idée de réaliser des systèmes hybrides dépasse le cadre des systèmes neuro-symboliques et même celui de l'informatique. Nous commençons ce chapitre par situer la place de l'hybridation dans la résolution de problèmes, puis nous donnons les classifications antérieures des SHNS et nous proposons une nouvelle classification plus cohérente et plus complète.

3.1 Résolution de problèmes et hybridation

Les systèmes hybrides neuro-symboliques font partie du domaine de *la résolution de problèmes*. Au moins trois grandes approches peuvent être distinguées dans ce domaine :

- La seule recherche d'une solution : la qualité, l'efficacité de la méthode employée sont prépondérantes ; des méthodes mathématiques classiques non-neuronales (contrôle, approximation, optimisation, statistique, etc.) sont souvent utilisées.
- La recherche d'une solution explicable : il est indispensable d'expliquer à l'utilisateur comment et pourquoi tel résultat a été obtenu, et pour cela il est souvent utile de disposer de connaissances explicites sur le processus de résolution de problèmes; ces connaissances explicites sont généralement fournies par un expert, sont modélisées à l'aide de systèmes à base de connaissances [12].
- La recherche de la compréhension : comprendre les mécanismes cognitifs qui permettent à un expert, ou plus généralement à un être vivant, de résoudre des problèmes ; cette approche relève des sciences cognitives [64].

Des liens existent entre ces approches, notamment entre la deuxième et la troisième. En effet, la compréhension de certains mécanismes cognitifs peut nous aider dans la représentation des connaissances, tandis que certains modèles de représentation de connaissances peuvent être étudiés comme modèles cognitifs.

Nous présentons ci-dessous les points complémentaires entre les systèmes symbolique et connexionniste, les motivations qui conduisent à réaliser des systèmes hybrides neuro-symboliques, puis nous donnons quelques exemples montrons que la volonté de réaliser des systèmes hybrides existe dans d'autres domaines que celui des SHNS.

3.1.1 Systèmes symboliques, connexionnistes et l'hybridation

Dans le cas des SHNS, nous constatons que les points forts des systèmes symboliques (systèmes experts symboliques) compensent à peu près les points faibles des systèmes connexionnistes et vice versa. Cette compensation est résumée comme suit :

- L'insertion de connaissances théoriques sur le problème peut être faite d'une façon simple et directe dans un système symbolique. Il suffit de les convertir dans le formalisme de représentation de connaissances utilisé. Par contre, on ne peut pas profiter de ces connaissances dans un système connexionniste. Il faut toujours des exemples pour pouvoir acquérir des connaissances.
- Le traitement est séquentiel, les temps de réponse sont longs lors de la consultation d'un système symbolique. Par contre, les RNA sont composés d'une série d'unités de traitement qui peuvent opérer en parallèle, avec un temps de réponse très rapide.
- L'insertion de connaissances (e.g. règles) dans un système symbolique peut être faite rapidement une fois qu'elles ont été déjà traitées par un expert et/ou ingénieur de connaissances. Par contre, le processus d'apprentissage peut être assez long dans un système connexionniste puisque les poids des connexions sont adaptés petit à petit.
- L'apprentissage n'est pas un processus à la base des systèmes symboliques. L'acquisition de connaissances se fait plutôt par explicitation, d'où le *problème du goulot d'étranglement*. Par contre, l'apprentissage et la généralisation de connaissances à partir d'exemples sont les points forts des approches connexionnistes.
- Les approches symboliques permettent d'obtenir des explications sur les réponses données par le système. Les connaissances sont facilement interprétables. Par contre, les réseaux connexionnistes sont des "boites noires", où les connaissances sont codées dans les poids et les interconnexions. Nous n'avons pas accès à une forme compréhensible de ces connaissances, qui puisse être interprétée directement par un être humain afin d'expliquer les réponses obtenues.

- Usuellement, pour qu'un système symbolique puisse bien fonctionner, les connaissances théoriques doivent être à la fois correctes et complètes. L'approche symbolique n'est pas adaptée aux traitements d'informations approximatives ou incomplètes. Par contre, les réponses d'un système connexionniste se dégradent progressivement en présence d'une entrée bruitée. Les réseaux connexionnistes sont très adaptés au traitement d'informations approximatives et incomplètes.
- Dans un système symbolique, les connaissances sont représentées par des règles et par des structures de données. Ce sont des connaissances de haut niveau. Par contre, les connaissances codées dans les réseaux connexionnistes représentent bien les relations entre leurs variables d'entrée.

3.1.2 Recherche d'une synergie neuro-symbolique

Nous retrouvons dans les SHNS l'ambivalence des approches possibles en résolution de problèmes : certains veulent construire de meilleurs outils informatiques pour résoudre des problèmes concrets [20], d'autres veulent construire de meilleurs modèles cognitifs [31, 9]. Beaucoup de chercheurs ont des démarches intermédiaires [28, 50] : utiliser des idées provenant des sciences cognitives pour obtenir de meilleurs outils informatiques. Toutefois, que ces recherches soient plutôt informatiques ou plutôt cognitives, elles ont un point commun important : elles sont motivées par l'incapacité actuelle d'un paradigme donné (symbolique, connexionniste ou autre) à résoudre à lui seul des problèmes difficiles.

Etant donné qu'il est difficile d'inventer de toutes pièces un nouveau paradigme plus satisfaisant, la "démarche du moindre effort" consiste à tirer parti des points forts de plusieurs paradigmes et à réaliser des systèmes ou modèles hybrides. Dans le cas des SHNS, nous constatons que les points forts des systèmes symboliques (systèmes experts symboliques) compensent à peu près les points faibles des systèmes connexionnistes et vice versa (voir section 3.1.1). De plus, l'intégration de ces deux systèmes permet de diminuer, sinon de résoudre complètement, certaines limitations de l'un ou de l'autre système afin d'avoir des systèmes plus robustes [49].

Les chercheurs vont plus loin que la "démarche du moindre effort" en donnant des justifications cognitives à la réalisation de modèles hybrides. Dans un système expert symbolique, il est indispensable de disposer de plusieurs modes complémentaires d'expression des connaissances, de manière à pouvoir représenter le "savoir-que" et le "savoir-faire" d'un problème : un SHNS est un des moyens d'offrir différents modes d'expression [28].

3.1.3 Exemples de systèmes hybrides

Il existe de nombreux exemples de systèmes hybrides, non nécessairement neuro-symboliques, qui donnent beaucoup de résultats intéressants dans la résolution de problèmes [49].

Par exemple, un système de contrôle réactif d'un robot autonome [43], qui doit trouver un chemin dans un terrain encombré d'obstacles. Les systèmes réactifs sont utiles dans les environnements inconnus et variables, mais ne peuvent pas en général modifier ou améliorer leur comportement avec leur expérience. Pour cette raison, les auteurs ont ajouté un module d'apprentissage à un module de navigation réactive. Ce module d'apprentissage règle en permanence certains paramètres du module de navigation et combine lui-même deux méthodes, le raisonnement à base de cas (RBC) et l'apprentissage par renforcement. Le composant RBC perçoit et caractérise l'environnement, recherche un cas approprié, et utilise les recommandations de ce cas pour régler les paramètres ; le composant d'apprentissage par renforcement, de son coté, affine le contenu des cas en fonction de l'expérience courante du système.

Il existe aussi un système appelé MORPH [68], qui combine plusieurs méthodes d'apprentissage et parvient à jouer à un niveau moyen aux échecs, mais avec peu de connaissances sur le domaine, une faible profondeur de recherche (un coup de profondeur) et aucun professeur indiquant la qualité d'un coup. Les méthodes étudiées sont les algorithmes génétiques, la généralisation à base d'explication, l'induction de concepts structurés, le tout combiné avec deux fonctions d'évaluation heuristique. On change donc d'échelle par rapport au système précédent, du point de vue de la complexité du système. Les auteurs soulignent que cette intégration n'aurait pas été possible sans une représentation commune à toutes ces méthodes. Nous voyons donc là une première contrainte pouvant apparaître dans la conception d'un système hybride. Enfin, le besoin de structuration des connaissances dans les SES a conduit à réaliser des systèmes experts hybrides, comportant par exemple des règles de production et des objets [45]. Les problèmes auxquels sont toujours confrontés les concepteurs de systèmes hybrides, informatiques ou non, sont :

- Définir la répartition du travail entre les différents composants.
- Définir les coopérations possibles entre ces composants, et trouver les mécanismes permettant d'obtenir cette coopération.
- Voir si on garde bien les avantages de chaque composant sans prendre ses faiblesses.
- Evaluer les apports du système hybride par rapport aux systèmes non hybrides.
- Etudier la rentabilité du système hybride par rapport aux systèmes non hybrides.

3.2 Classification des SHNS

Les SHNS sont nombreux et très variés, de par leurs composants et par les techniques de couplage utilisées, mais aussi en raison de différents sens donnés au mot “hybride” dans le cadre général de l'intégration des points forts des systèmes symboliques et connexionnistes. Clarifier ce domaine en pleine évolution, comparer les systèmes, distinguer ce qui est nouveau sont des préalables indispensables à de nouvelles recherches.

Plusieurs schémas de classification des SHNS existent actuellement, et la plupart de ces schémas sont basés sur des définitions étroites et ne décrivent pas complètement toutes les caractéristiques qu'un système peut développer.

3.2.1 Approches antérieures

Un grand nombre de travaux [84, 40, 44, 48] se situent dans le courant de l'intégration des meilleures caractéristiques de chacun des mondes connexionniste et symbolique. Ces travaux ont donné naissance à des schémas de classifications de SHNS plus au moins cohérents et complémentaires.

Classification établie par Medsker et Baily [44]

Une première classification de SHNS a été proposée par Medsker et Baily [44] qui classent les divers systèmes intégrant les réseaux de neurones et les systèmes experts symboliques selon leur degré de couplage. Ils distinguent tout d'abord le cas où les modules sont entièrement séparés, qui ne sont pas en réalité hybrides car il n’y a pas d'interaction entre les modules, et le cas des systèmes qui transforment un module en un autre par une procédure plus au moins automatique.

Ces auteurs examinent ensuite les classes de systèmes faiblement, étroitement et totalement intégrés pour décrire le degré de couplage entre les modules. Dans les systèmes faiblement couplés la communication est assurée par des fichiers. Par contre, dans les systèmes étroitement et totalement couplés par la mémoire vive.

La classification des SHNS présentée par Medsker et Baily [44] ne s'intéresse qu'au degré de couplage des modules et ne donne aucune description sur la hiérarchie des modules. De plus, les modèles unifiés comme ceux définis par Lallement et al. [38] (voir section suivante) ne figurent pas dans la classification des auteurs.

Classification établie par Lallement et al. [38]

Deux types d'approches ont été envisagées par Lallement et al. [38] pour intégrer des fonctionnalités numériques et symboliques dans des systèmes cognitifs puissants : une approche hybride et une approche unifiée.

- ✓ ***Approche hybride :*** on distingue clairement un module symbolique et un module connexionniste. Ces modules peuvent être faiblement ou fortement couplés qui interagissent selon quatre modes : le traitement chaîné, le sous-traitement, le méta-traitement, ou le co-traitement.
- ✓ ***Approche unifiée :*** comprend les modèles descendants dont le but est de construire des architectures connexionnistes effectuant des traitements symboliques, et les modèles ascendants dont la finalité est de former des architectures biologiquement plausibles. Les modèles descendants peuvent être localistes (un atome de connaissance correspond à une unité) ou distribués (un atome de connaissances est représenté par un ensemble d'unités du réseau).

Dans leur classification des SHNS, Lallement et al. [38] ont ignoré la classe des systèmes effectuant des transformations de modules. Aussi, le terme "hybride" est sous-utilisé par les auteurs du moment que tous les systèmes qu'ils décrivent sont hybrides, y compris les modèles classifiés sous l'approche unifiée.

Classification établie par McGarry et al. [40]

McGarry et al. [40] décrivent trois types de SHNS : les systèmes hybrides unifiés, les systèmes hybrides transformationnels et les systèmes hybrides modulaires.

- ✓ ***Systèmes hybrides unifiés :*** des systèmes dont toutes les activités de traitement sont implémentées par des éléments neuronaux. Dans ce groupe, les auteurs distinguent deux grandes tendances : les modèles localistes (mode de représentation localisé) et les modèles distribués (mode de représentation distribué).
- ✓ ***Systèmes hybrides transformationnels :*** ce groupe représente les systèmes permettant la conversion d'un module symbolique en un module connexionniste ou vice-versa. Ces systèmes consiste en la compilation de base de règles en réseau de neurones, l'extraction de règles à partir d'un réseau de neurones et le raffinement de règles à l'aide d'un réseau de neurones.
- ✓ ***Systèmes hybrides modulaires :*** se sont des systèmes composés de modules neuronaux et symboliques distincts, qui peuvent avoir différents degrés de couplage et d'intégration. La complexité des systèmes hybrides modulaires peut être mesurée par le flux d'information entre les modules, qui peut être

unidirectionnel ou bidirectionnel. Une autre caractéristique prise en compte par les auteurs est le degré de couplage entre les modules, il peut être faible, étroit ou fort.

Dans leur classification de SHNS, les auteurs ont ignorés les modèles ascendants sous l'approche unifiée comme ceux définis par Lallement et al. [38] (section précédente), ainsi que le mode combiné de représentation d'un concept ou d'un atome de connaissance dans un réseau sous les approches purement connexionnistes comme celui proposé par Orsier et Labbi [48] (voir section suivante).

Classification établie par Orsier et Labbi[48]

Orsier et Labbi [48] distinguent trois classes de systèmes hybrides neuro-symboliques d'importance comparable du point de vue applications selon trois approches : approche hybride, approches purement connexionnistes, et approche semi-hybride.

- ✓ ***Approche hybride :*** sous cette approche sont classifiés les systèmes qui sont effectivement composés d'au moins deux modules de type différent. Ces systèmes sont répartis en deux sous-classes : systèmes avec composants bien distincts, qui peuvent être faiblement ou étroitement couplés, et systèmes avec composants intégrés qui sont fortement couplés. Les interactions entre les composants faiblement couplés sont de type coopération ou pré/post-traitement. Par contre, les interactions entre les composants étroitement ou fortement couplés sont de type coopération, méta-traitement ou sous-traitance.
- ✓ ***Approches purement connexionnistes :*** le but de ces approches est d'émuler avec un système connexionniste un système symbolique, afin de prouver la capacité des systèmes connexionnistes à effectuer des traitements de haut niveau. Nous parlerons d'émulation car il s'agit de reproduire le fonctionnement et l'architecture d'un système symbolique. Ces approches sont réparties en trois sous-classes : approche localiste, approche distribuée et approche combinée.
- ✓ ***Approche semi-hybride :*** les systèmes semi-hybrides sont destinés à réaliser des traductions. Cela regroupe essentiellement la compilation de base de règles en réseau de neurones, l'extraction de règles à partir d'un réseau de neurones, et le raffinement de règles à l'aide d'un réseau de neurones.

Le terme "hybride" est sous-utilisé par Orsier et Labbi [48] du moment que tous les systèmes qu'il décrit sont hybrides, y compris les systèmes classifiés sous les approches purement connexionniste et semi-hybride. De plus, les modèles ascendants unifiés comme ceux définis par Lallement et al. [38] ne figurent pas dans la classification des auteurs.

3.2.2 Notre classification des SHNS

Plusieurs schémas de classification de systèmes hybrides existent, avec différents objectifs et terminologies mais avec un certain degré de similarité dans la description. Par conséquent, nous proposons une nouvelle classification, qui reprend certaines classes mentionnées ci-dessus, pour les raisons suivantes :

- Notre schéma attache ensemble plusieurs fils des autres schémas de classification dans une approche cohérente et que les développements récents en technologie hybride exigent une nouvelle perspective non couverte par les schémas existants.

- En ayant une approche commune à classifier les différents systèmes hybride neuro-symboliques, les développeurs de tels systèmes seront en meilleure position pour décrire l'opération et les objectifs de leurs systèmes, tandis que d'autres pourront évaluer de tels systèmes avec une compréhension plus claire des issues du traitement de l'information.

Le développement de systèmes hybrides neuro-symboliques est dans un état continu d'avancement et notre schéma de classification peut représenter une autre étape dans le développement des systèmes hybrides neuro-symboliques.

L'analyse de l'état de l'art des SHNS montre que tous les systèmes neuro-symboliques étudiés et développés jusqu'à présent dans le domaine sont hybrides du fait qu'ils combinent les deux techniques neuronale et symbolique de résolution de prob-lèmes, et qu'un schéma complet de classification peut être établi en suivant trois types d'approches : approche unifiée, approche modulaire et approche transformationnelle.

Approche unifiée

Nous distinguons deux types de modèles hybrides unifiés : les modèles descendants et les modèles ascendants. Dans le premier cas, le but est de construire des architectures connexionnistes effectuant des traitements symboliques. Les modèles sont guidés par les fonctionnalités visées : faire ancrer des symboles dans un substrat connexionniste. Dans le second cas, le but est de construire des architectures biologiquement plausibles, et les modèles sont guidés par la neurobiologie : faire émerger des symboles d'un substrat connexionniste.

Les modèles descendants sont à leur tour répartis en trois types : les modèles localistes, distribués et combinés. Ce qui fait la différence entre les trois est le mode de représentation des connaissances, localisé dans le premier cas, distribué dans le second et combiné (localisé/distribué) dans le dernier.

Approche modulaire

Dans l'approche modulaire, nous distinguons clairement un module symbolique et un module connexionniste. Cette classification est de type structurel : nous distinguerons les différents types de systèmes hybrides par leur structure plutôt que par leurs fonctions ou celles de leurs modules. Elle prend en compte deux dimensions : le degré de couplage et le mode de couplage entre les modules connexionnistes et symboliques.

Le degré de couplage définit la force d'interaction entre les deux modules. Le couplage sera dit *faible* si les modules interagissent faiblement, c'est à dire s'ils sont par exemple simplement juxtaposés et sous l'autorité d'un même superviseur, ou s'ils ne font que s'échanger des messages. Le couplage sera dit *fort* si les deux modules agissent d'une manière très étroite, par exemple en partageant des structures de données.

Le mode de couplage définit l'architecture logique du système. Nous distinguons quatre modes de couplage : le pré/post-traitement, le sous-traitement, le méta-traitement et le co-traitement.

Approche transformationnelle

Sous l'approche transformationnelle sont classifiés les systèmes permettant la traduction (transformation) d'un module symbolique en un module connexionniste ou vice-versa. Le processus de transformation peut être une compilation complète ou partielle de l'information d'une forme à l'autre. Ces systèmes consiste en la compilation de base de règles en réseau, l'extraction de règles à partir d'un réseau et le raffinement de règles à l'aide d'un réseau.

Ces trois approches pour l'intégration neuro-symbolique sont résumées sur la figure 3.1. Les modèles hybrides issus de notre classification seront étudiés en détail dans les chapitres 4, 5 et 6 du présent mémoire.

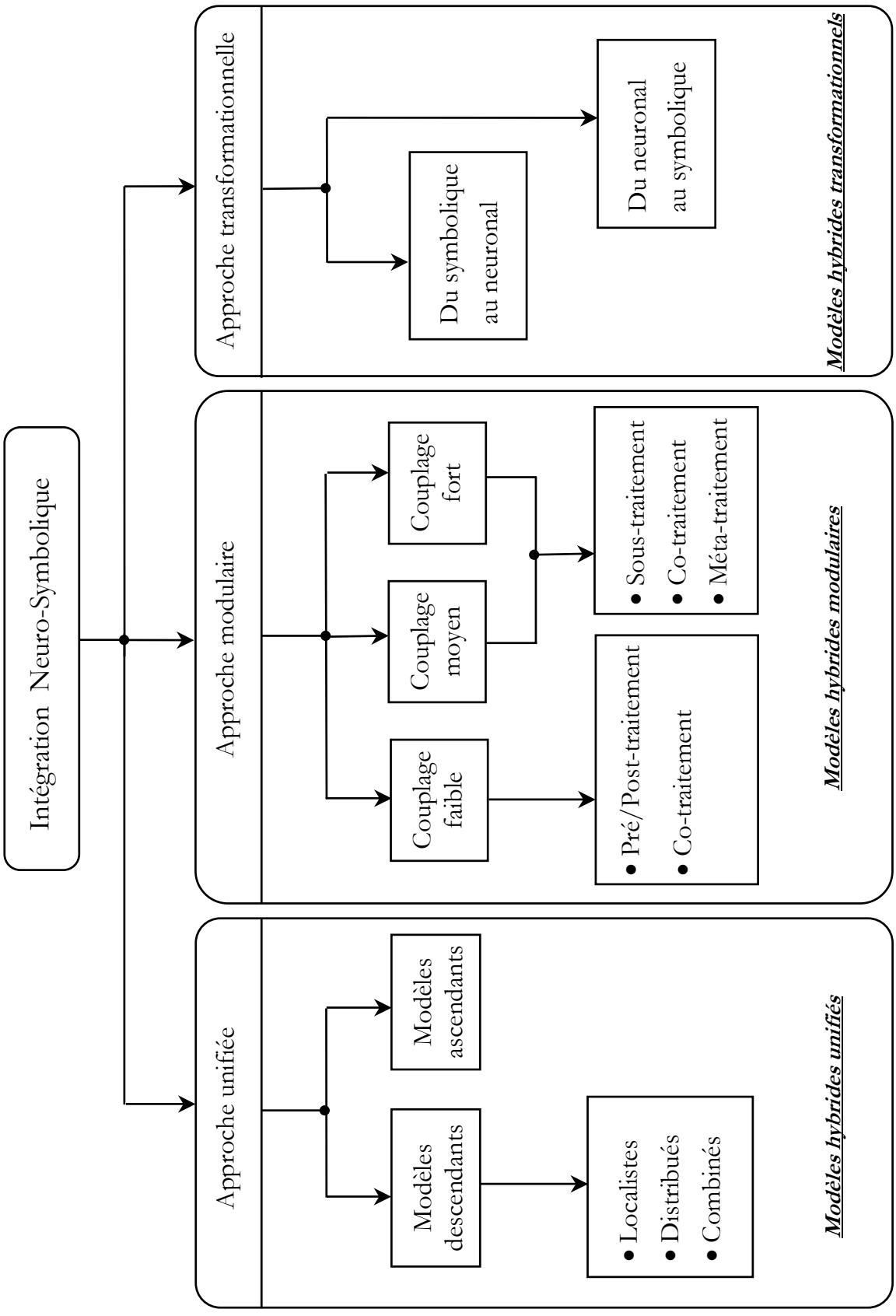

Figure 3.1 : Notre classification des SHNS

3.3 Conclusion

Dans ce chapitre nous avons donné plusieurs arguments forts en faveur de l'intégration des points forts des systèmes symboliques et connexionnistes, leur complémentarité pour la résolution de problèmes complexes au sein des systèmes hybrides neuro-symboliques.

Nous avons présenté aussi un état de l'art sur la classification des systèmes hybrides neuro-symboliques, et nous avons proposé une nouvelle classification cohérente qui rassemble les points intéressants des schémas de classification existants et qui prend en compte les développements récents dans le domaine de l'intégration neuro-symbolique. Notre schéma de classification est basée sur trois approches : unifiée, hybride et transformationnelle.

- ✓ Dans le cas de l'approche unifiée, un seul paradigme est utilisé pour l'intégration neuro-symbolique : toutes les opérations cognitives seront prises en charge par un seul module (modèles unifiés ascendants). Cependant une autre sous-classe de l'approche unifiée, ni purement connexionniste ni purement symbolique existe : le traitement connexionniste symbolique (*connectionist symbol processing*), dont le but est de manipuler des symboles à l'aide de réseaux connexionnistes (modèles unifiés descendants).
- ✓ Dans le cas de l'approche modulaire, plusieurs paradigmes sont utilisés pour prendre en charge les divers types d'opérations cognitives. Nous distinguons clairement un module symbolique et un module connexionniste. L'hypothèse de base de l'approche modulaire est que les modèles symboliques et connexionnistes ne sont pas suffisants par eux-mêmes, mais que leur combinaison peut permettre d'exécuter tous les types d'opérations cognitives. Les modules peuvent être faiblement, étroitement ou fortement couplés.
- ✓ Dans le cas de l'approche transformationnelle, les modèles sont capables de convertir une structure symbolique en une architecture neuronale ou vice-versa. Cette approche permet donc la construction de systèmes qui peuvent fonctionner entre les deux niveaux neuronal/symbolique de représentation des connaissances.

Notre schéma de classification permettra aux développeurs de systèmes hybrides neuro-symboliques de décrire facilement leur démarche de conception et de bien fixer les objectifs de leurs systèmes.

Chapitre 4

Les modèles hybrides unifiés

Dans le cas de l'approche unifiée, les opérations cognitives de type symbolique ou connexionniste sont exécutées par un même module. Comme on peut avoir des réseaux connexionnistes qui manipulent des symboles. Plusieurs travaux de recherche [72, 73, 75, 78, 79, 80] ont montré que les réseaux de neurones artificiels peuvent manipuler avec succès des connaissances de haut niveau habituellement réservées au processus symboliques. L'intégration de composants symboliques avec la puissance inductive des réseaux de neurones artificiels a donné naissance à des modèles hybrides unifiés pour la résolution de problèmes complexes.

Dans ce chapitre, nous donnons les motivations des chercheurs pour ce type de modèles, nous classifions les modèles hybrides unifiés et nous étudions quelques exemples de systèmes représentatifs.

4.1 Motivations

Les modèles hybrides unifiés ont une architecture proche de celle du cerveau humain, qui est elle-même unifiée en ce sens qu'elle est fondée sur un élément de base : le neurone, à partir duquel toutes les structures plus complexes sont construites. Grâce à ce rapprochement d'architectures les modèles hybrides unifiés peuvent s'attaquer aux tâches de haut niveau que requière l'étude de la cognition humaine. L'idée fondamentale de l'approche hybride unifiée est donc de créer des briques de base, comme les neurones, sur lesquelles la cognition pourra se fonder, avec en particulier le côté analytique et le côté synthétique. Cette approche est fondée principalement sur deux notions, à savoir l'ancrage et l'émergence des symboles.

4.1.1 Ancrage des symboles

Dans un système symbolique, la sémantique des symboles et des expressions symboliques est externe, c'est-à-dire qu'elle est attribuée par un observateur extérieur au système. Par contre, les symboles manipulés par un être humain ont une sémantique

interne : chacun de nous connaît la signification des mots qu'il emploie, sans qu'un système extérieur soit nécessaire pour l'attribution d'une sémantique a posteriori.

Lorsqu'un symbole a une sémantique interne, on dira qu'il est ancré dans les perceptions, comme dans le cas des symboles manipulés par un être humain. L'internalité de la sémantique repose sur un rapport étroit avec le monde extérieur, via des canaux de perception ; la sémantique est construite par interaction avec le monde extérieur. La forme des symboles n'est alors plus arbitraire, mais contrainte par les perceptions. Pour que les symboles ancrés apparaissent, il faut à la fois un appareil perceptif et un appareil compositionnel. Les symboles ne doivent pas être des entités monolithiques, mais plutôt être représentés dans plusieurs niveaux, compris entre le niveau perceptif et le niveau raisonnement. Les réseaux connexionnistes, par leurs capacités à traiter les perceptions, sont un bon candidat pour participer à un système d'IA manipulant des symboles.

4.1.2 Emergence des symboles

Une notion à laquelle il est souvent fait référence dans la littérature est liée à la notion de l'ancrage : il s'agit de l'émergence. Cette notion est subtile et difficile à saisir, mais les chercheurs lui adopte une définition très générale : ce qui se produit quand un ensemble de comportements simples produisent un comportement global complexe.

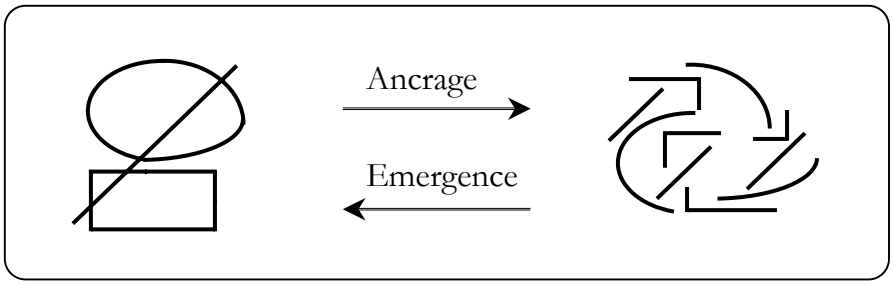

Figure 4.1 : Ancrage / Emergence des symboles

Il est aisé de trouver des exemples de phénomènes émergents : le comportement social d'une colonie de fourmis émerge du comportement simple des fourmis individuelles; le comportement d'un réseau de neurones émerge du comportement très simple des neurones. Dans le cas particulier des symboles, la notion d'émergence est à rapprocher de la notion d'ancrage. L'émergence de symboles est un moyen pour ancrer ces symboles. Un symbole peut en effet émerger à partir des perceptions des objets qu'il symbolise, et il est alors ancré dans ces perceptions (voir figure 4.1). Chez un enfant le symbole *avion* va émerger à partir des images d'avions qui lui auront été présentées.

4.2 Classification des modèles hybrides unifiés

Dans cette section, nous présentons une classification des modèles hybrides unifiés (sous l'approche hybride unifiée) que nous avons proposé au chapitre précédent. On distingue deux types de modèles : les modèles ascendants et les modèles descendants (voir figure 4.2).

4.2.1 Modèles ascendants

Le but des modèles ascendants est de construire des architectures biologiquement plausibles. Ces modèles sont guidés par la neurobiologie : l'approche est de type ascendante, il faut faire émerger des symboles d'un substrat connexionniste.

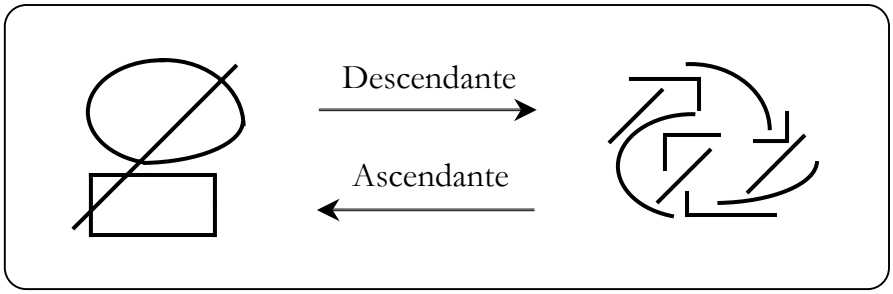

Figure 4.2 : Approche ascendante / Approche descendante

Les modèles ascendants tentent de couvrir l'intégralité des opérations cognitives en allant du côté connexionniste au côté symbolique.

4.2.1 Modèles descendants

Le but des modèles descendants est de construire des architectures connexionnistes de traitement symbolique (*connectionist symbol processing*). Ces modèles sont guidés par les fonctionnalités visées : l'approche est de type descendante, il faut ancrer les symboles dans un substrat connexionniste (figure 4.2). On distinguera trois types de modèles descendants, selon que le mode de représentation des connaissances dans le réseau est localisé, distribué, ou combiné.

- ✓ **Modèles localistes :** les connaissances sont représentées de façon localisée (un atome de connaissance correspond à un neurone du réseau).
- ✓ **Modèles distribués :** les connaissances sont représentées de façon distribuée (un atome de connaissance est représenté par un ensemble de neurones du réseau) ;
- ✓ **Modèles combinés :** dans un même modèle on retrouve les deux modes localisé et distribué de représentation des connaissances.

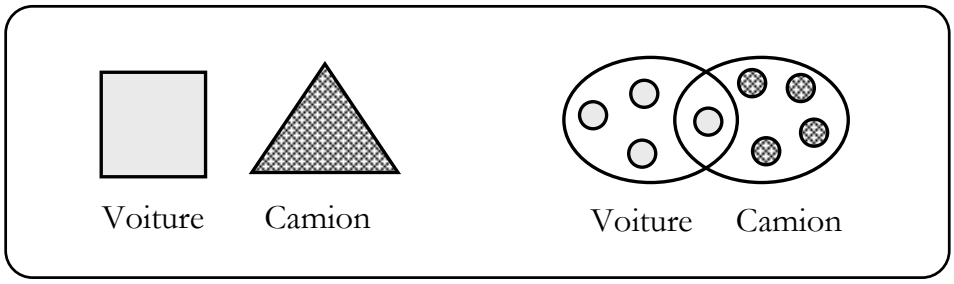

Figure 4.3 : Représentation localisée / Représentation distribuée

On remarque que les modèles localistes héritent plus facilement des caractéristiques des systèmes symboliques, et les modèles distribués des caractéristiques des systèmes connexionnistes (figure 4.3). Les modèles descendants tentent donc de couvrir l'intégralité des opérations cognitives en allant du côté symbolique au côté connexionniste.

4.3 Exemples de modèles hybrides unifiés

Cette section est consacrée à l'étude de quelques exemples de modèles hybrides unifiés (de types descendant ou ascendant) tels que le réseau connexionniste à traitement symbolique de Ajjanagadde et Shastri [84], le système de raisonnement de sens commun de Sun [72], le système Boltzcons de Touretzky [73], et les mémoires katamiques de Nenov et Dyer [75]. Ces modèles sont développés dans le domaine de l'intégration des composants neuronaux et symboliques dans des systèmes hybrides puissants.

4.3.1 Réseau connexionniste à traitement symbolique

Une architecture connexionniste pour le raisonnement à partir de règles et la représentation de variables est proposée par Ajjanagadde et Shastri [84].

Les connaissances du système sont représentées de façon locale en utilisant plusieurs types de nœuds : les nœuds *prédicats* (pour les noms de prédicats), les nœuds *variables* (pour les arguments des prédicats et les variables) et les nœuds *valeurs* (pour les valeurs possibles des variables). Les règles du système sont codées par des connexions entre les nœuds *prédicats* et les nœuds *variables*.

Le réseau est constamment parcouru par une activation cyclique ; une variable et une valeur sont liées si leurs nœuds hôtes oscillent en phase. Ce système ne présente pas de capacités d'apprentissage et trouve ses limites lorsque la taille du réseau augmente.

Ce modèle unifié est de type descendant, puisqu'on cherche à définir une architecture permettant d'obtenir les capacités symboliques visées, et localiste car chaque unité possède sa propre sémantique.

4.3.2 Système de raisonnement de sens commun

Le système CONSYDERR (*CONnectionist SYstem with Dual representation for Evidential Robust Raisoning*) proposé par Sun [72] a pour but d'effectuer des raisonnements de sens commun (raisonnements que font couramment les gens) et qui sont difficilement pris en charge par l'IA classique. Ces formes de raisonnement, identifiées et analysées par des psychologues, ont été décrites par onze petits protocoles sous forme de question/réponse. On donne ici deux exemples :

Q : Are the roses in England ?
R : There are a lot of flowrs in England. So I guess there are roses.

Q : Is the Chaco the cattle country ?
R : It is like Western Texas, so in some sens I guess it's cattle country.

Certains systèmes, connexionnistes ou non, peuvent rendre compte de certaines de ces formes, mais aucun n'est capable de toutes les traiter dans un cadre unifié [49]. Réaliser ce cadre est le but recherché par le système CONSYDERR qui comprend deux niveaux, chacun d'entre eux est un réseau connexionniste : un niveau Connexionniste Local (CL) et un niveau Connexionniste Distribué (CD).

Au niveau CL sont représentées des règles de production d'ordre 0 : une règle est un ensemble de liens entre ses prémisses et sa conclusion. Chaque lien est pondéré, et l'inférence se fait en utilisant de la logique floue. Les nœuds représentent des entités, des concepts précis (par exemple : fleur, rose). Dans CL l'opération de base est le calcul de sommes pondérées. Par exemple, soit la règle $A_1, A_2, A_3 \rightarrow B$ où les A_i et B sont des propositions ; elle est codée dans CL en connectant les nœuds A_i à B et en faisant calculer par le nœud B l'activation $ACT_B = \sum_{i=1}^{3} w_i * ACT_i$, les poids w_i étant donnés.

Au niveau CD, un concept A de CL est représenté par un ensemble de nœuds dans CD, qui correspondent à ses caractéristiques F_A (par exemple : a-une-couleur, a-une-odeur...). De même une règle de CL est représentée par un ensemble de liens dans CD. D'autre part, les caractéristiques représentées par les nœuds de CD peuvent être communes à plusieurs concepts de CL : par exemple *désertique* est une caractéristique du concept *Chaco* mais aussi du concept *Western-Texas* ; en fait le nombre de nœuds de CD communs à deux concepts de CL est proportionnel au degré de similarité entre ces deux concepts.

Le raisonnement dans le système s'effectue en plusieurs phases (voir figure 4.4) :

1. **Phase réception des entrées :** l'utilisateur active certains nœuds du CL indiquant les concepts existants ;
2. **Phase CL → CD :** les concepts correspondants et leurs propriétés sont à leur tour activés dans le CD (propagation d'activité) ;
3. **Phase d'inférence :** dans chacun des deux niveaux, les activations sont propagées en suivant les liens, ce qui correspond à l'activation des règles. De nouveaux nœuds sont donc activés dans CL et CD ;
4. **Phase CD → CL :** les activations obtenues dans CD sont propagées vers CL. Les nœuds de CL prennent pour activation le maximum de leur activation à la fin de la phase d'inférence et d'une valeur calculée à partir de ce qui remonte de CD. C'est l'utilisateur qui doit alors exploiter le résultat (*cattle-country*) pour répondre à la question posée (*Is the Chaco the cattle country ?*).

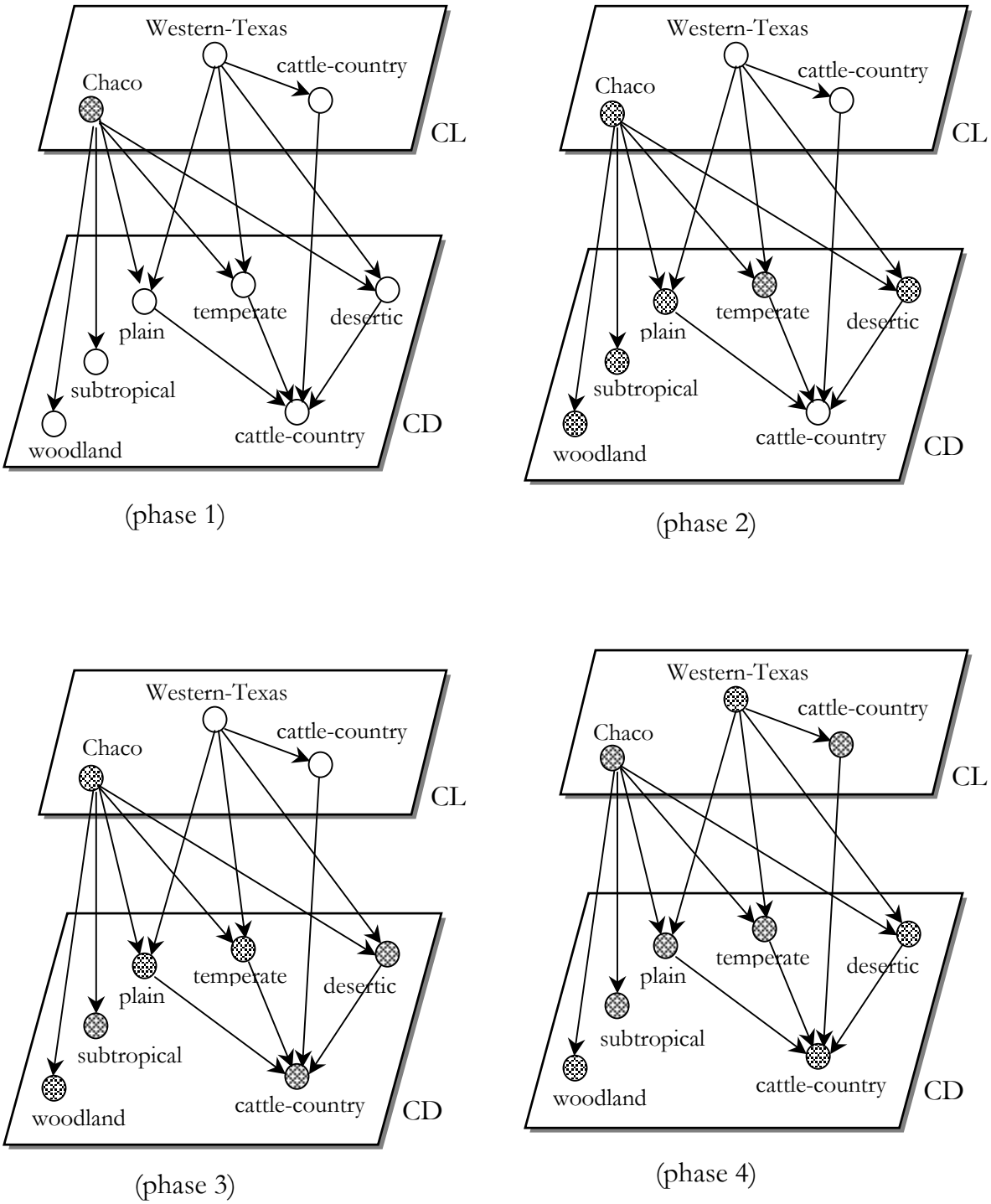

Figure 4.4 : D'après Sun [72].
Résolution avec CONSYDERR du protocole « Chaco ».

La phase d'inférence présente le fonctionnement suivant :

- Dans CL, si le concept A a un coefficient de certitude C_A, et la règle $A \rightarrow B$ un coefficient de confiance ($A \rightarrow B$), alors B aura un coefficient de certitude égale à $C_A * (A \rightarrow B)$.
- Dans CD, on mesure la similarité entre A et B, notée $A \sim B$, par le nombre de caractéristiques communes à A et à B. Lorsque A reçoit une activation C_A, alors B recevra une activation $C_A * (A \sim B)$.

Par exemple, si l'on a $A \sim B$, et $B \rightarrow C$, si A a un coefficient de certitude C_A, alors C aura, à la fin des 4 phases, un coefficient de certitude $C_A * (A \sim B) * (B \rightarrow C)$.

CONSYDERR est un modèle hybride unifié de type combiné (un niveau localiste et un niveau distribué). Son problème est l'absence de mécanisme d'apprentissage : une fois réalisée, une application est complètement figée. Introduire une capacité d'apprentissage serait très difficile puisque tous les poids sont pré-calculés de manière formelle pour que les interactions entre CL et CD implémentent bien le calcul de similarité et l'application de règles.

4.3.3 Système Boltzcons

Boltzcons [73] est un système hybride unifié qui implémente des primitives à la Lisp pour stocker et manipuler des représentations structurées (listes par exemple) de façon connexionniste.
Les listes sont représentées par des séries de triplets (*étiquette, élément, étiquette_suiv*) où *étiquette_suiv* indique l'étiquette du triplet suivant. Les différents triplets présents à un instant donné sont stockés dans une mémoire distribuée de 2000 cellules ; chaque cellule est un tableau de 6 triplets choisis au hasard, et sera active quand un de ses triplets sera présent en mémoire. Pour un triplet donné, plusieurs cellules seront actives simultanément. Un système de maintenance connexionniste assure l'ajout, le retrait ou la modification des triplets présents en mémoire.

Boltzcons est un système descendant distribué d'où sa robustesse en cas de défaillance de cellules. De plus, au contraire des systèmes descendants localistes, deux structures différentes comme (*a, b, c*) et (*a, c, b*) peuvent être représentées simultanément et sans collision en mémoire.

4.3.4 Mémoires Katamiques

Les mémoires Katamiques proposées par Nenov et Dyer [75] correspondent à des réseaux d'inspiration nettement biologique. Ces modèles fondés sur la perception permettent d'acquérir la sémantique d'un langage naturel à partir de ses interactions avec le monde réel. Le principe est de fournir au réseau des stimuli sensori-moteurs en même temps que des informations verbales décrivant ces stimuli, afin de permettre au système de construire une relation entre les deux représentations.

Les mémoires katamiques sont des réseaux inspirés de la colonne corticale (unité fonctionnelle de base du cortex cérébral) pour l'apprentissage de séquences. Les unités de base ont un comportement plus complexe que le neurone formel classique.

Ces modèles unifiés ascendants ont seulement été testés sur des exemples d'école [75]. Pourtant, ils sont très prometteurs au sens ou ils sont directement inspirés par des tâches humaines, des tâches perceptives ou motrices de bas niveau jusqu'à un comportement cognitif de haut niveau.

4.4 Conclusion

Le but des systèmes hybrides unifiés est d'implémenter des traitements symboliques avec des composants neuronaux. La motivation pour cette axe de recherches est que certaines formes de ces traitements peuvent facilement être effectuées par une structure neuronale. L'utilisation des représentations localisées, distribuées et combinées permet d'augmenter la capacité de généralisation des réseaux de neurones. Actuellement, la plupart des systèmes unifiés développés n'apparaissent pas en mesure de supporter des problèmes réels complexes, bien que la recherche dans cette direction continue.

Chapitre 5

Les modèles hybrides modulaires

Dans le cas de l'approche modulaire, on distingue clairement un module symbolique et un module connexionniste. L'hypothèse de base de l'approche modulaire est que les modèles symboliques et connexionnistes ne sont pas suffisants par eux-mêmes, mais que leur combinaison peut permettre d'exécuter tous les types d'opérations cognitives. Plusieurs travaux de recherche [69, 70, 71, 81] décrivent les avantages de l'approche modulaire pour l'intégration de composants neuronaux et symboliques dans des systèmes hybrides.

Dans ce chapitre, nous examinons les motivations des chercheurs pour ce type de modèles, nous donnons les différentes classes des modèles hybrides et nous étudions quelques exemples de systèmes de ce type existants.

5.1 Motivations

Plusieurs arguments plaident en faveur des modèles hybrides modulaires, à savoir l'aspect traitement, les types d'informations manipulées, la réutilisation des modules, et la croissance de la complexité de ces modèles afin de se rapprocher du cerveau humain.

5.1.1 Aspect traitement

Nous avons vu au troisième chapitre (section 3.1.1) que les deux paradigmes, connexionniste et symbolique, sont complémentaires sur bien des points, en particulier la résistance au bruit, l'apprentissage, l'explication du raisonnement, la représentation des connaissances, etc. Il est tout à fait clair qu'un modèle intégrant les points forts de chaque paradigme serait particulièrement souhaitable.

De plus, un modèle hybride modulaire constitué de composants symboliques et de composants numériques offrira de meilleures performances, en termes de capacités de simulation de diverses opérations cognitives, que les modèles connexionnistes ou symboliques par eux-mêmes.

5.1.2 Types d'informations

Les systèmes hybrides modulaires permettent de bien prendre en charge des applications ayant à la fois des connaissances et des données sur le domaine. Ces connaissances peuvent être, par exemple, sous forme de règles et les données sous forme d'exemples.

Si pour une application on ne dispose pas suffisamment de connaissances sur le domaine de l'application, mais on a des données, on pourra faire l'apprentissage sur ces données. Ces deux types d'informations sont pris en compte de manière naturelle par des systèmes hybrides modulaires, où la partie symbolique serait chargée du traitement des connaissances, et la partie connexionniste du traitement des données.

5.1.3 Modules réutilisables

Les différents modules constituant un système hybride modulaire peuvent être réutilisés, seuls le contrôle et la communication entre modules doivent être entièrement développés. Ceci n'est certainement pas trivial, mais probablement plus simple pour une application que de recommencer depuis le début. De plus, l'IA distribuée peut fournir des méthodes d'interfaçage et de communication entre les divers composants d'un système hybride modulaire.

5.1.4 Complexité

Les modèles d'IA actuels sont certainement d'une complexité très faible en regard du cerveau humain et des opérations cognitives qu'il permet d'accomplir. Le processus visant l'augmentation de cette complexité est en cours depuis les débuts de l'IA.

Le nombre de modèles différents, tant symbolique que numériques, est en constante augmentation. De plus, la taille de ces modèles est beaucoup plus grande que celle des modèles anciens.

Chaque paradigme, symbolique ou connexionniste, a jusqu'ici principalement suivi son propre chemin, et a vu la complexité de ses modèles croître. Ceci est certainement dû à l'augmentation du nombre de chercheurs, ainsi qu'à celle de la puissance des machines utilisées. Sans aucun doute, cette complexification va continuer. Les modèles hybrides modulaires, en combinant la complexité de ses paradigmes, offrent la possibilité de franchir une nouvelle étape dans ce processus.

5.2 Classification des modèles hybrides modulaires

Dans cette section, nous présentons une classification des systèmes hybrides modulaires (sous l'approche modulaire) que nous avons proposé au chapitre 3 (figure 3.1). Cette classification est de type structurel : on distinguera les différents types de modèles par leur structure plutôt que par leurs fonctions ou celles de leurs modules. Notre classification est basée sur deux facteurs : le degré de couplage et le mode d'interaction entre les modules symbolique et connexionniste.

5.2.1 Degré de couplage

Le degré de couplage représente la force d'interaction entre les deux modules symbolique et connexionniste. La connaissance du degré de couplage permet de plus d'avoir une idée sur la manière dont les modules communiquent entre eux. Ce degré de couplage peut être faible, moyen ou fort (figure 5.1).

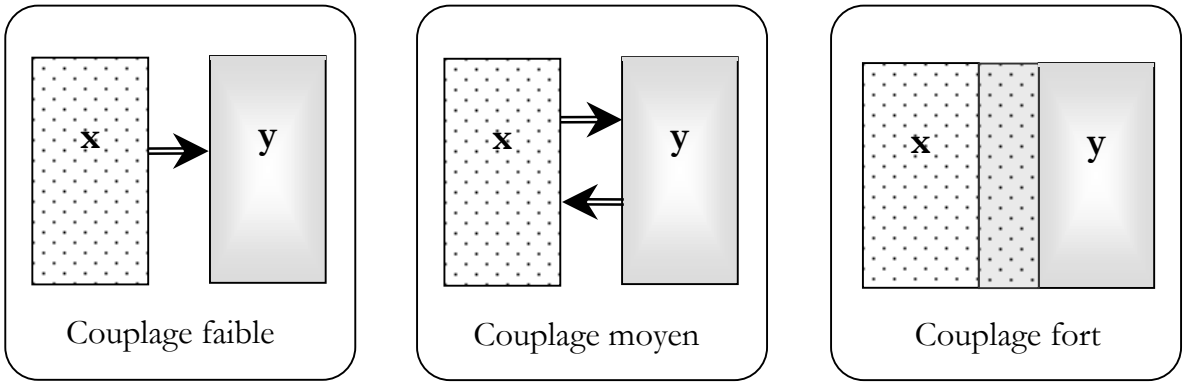

Figure 5.1 : Degrés de couplage entre deux modules x et y,
avec x, y ∈ { symbolique, connexionniste } mais x ≠ y.

- **Couplage faible :** les modules sont reliés par une simple relation d'entrée/sortie, et les communications sont donc uni-directionnelles.
- **Couplage moyen :** les interactions entre les modules sont plus souples, car elles sont bi-directionnelles, et chacun des modules peut influencer dans une certaine mesure le fonctionnement de l'autre (couplage étroit).
- **Couplage fort :** les modules partagent des structures de données communes, ou au moins l'un des modules a un accès direct à certaines structures de données de l'autre. Les connaissances et les données ne sont pas uniquement transférées, elles sont aussi partagées entre modules qui utilisent des structures de données internes communes.

5.2.2 Mode d'interaction

Le mode d'interaction permet d'obtenir immédiatement des renseignements sur l'architecture générale du système. On distingue quatre types de modes d'interaction : le pré/post-traitement, le sous-traitement, le co-traitement, et le méta-traitement (voir figure 5.2).

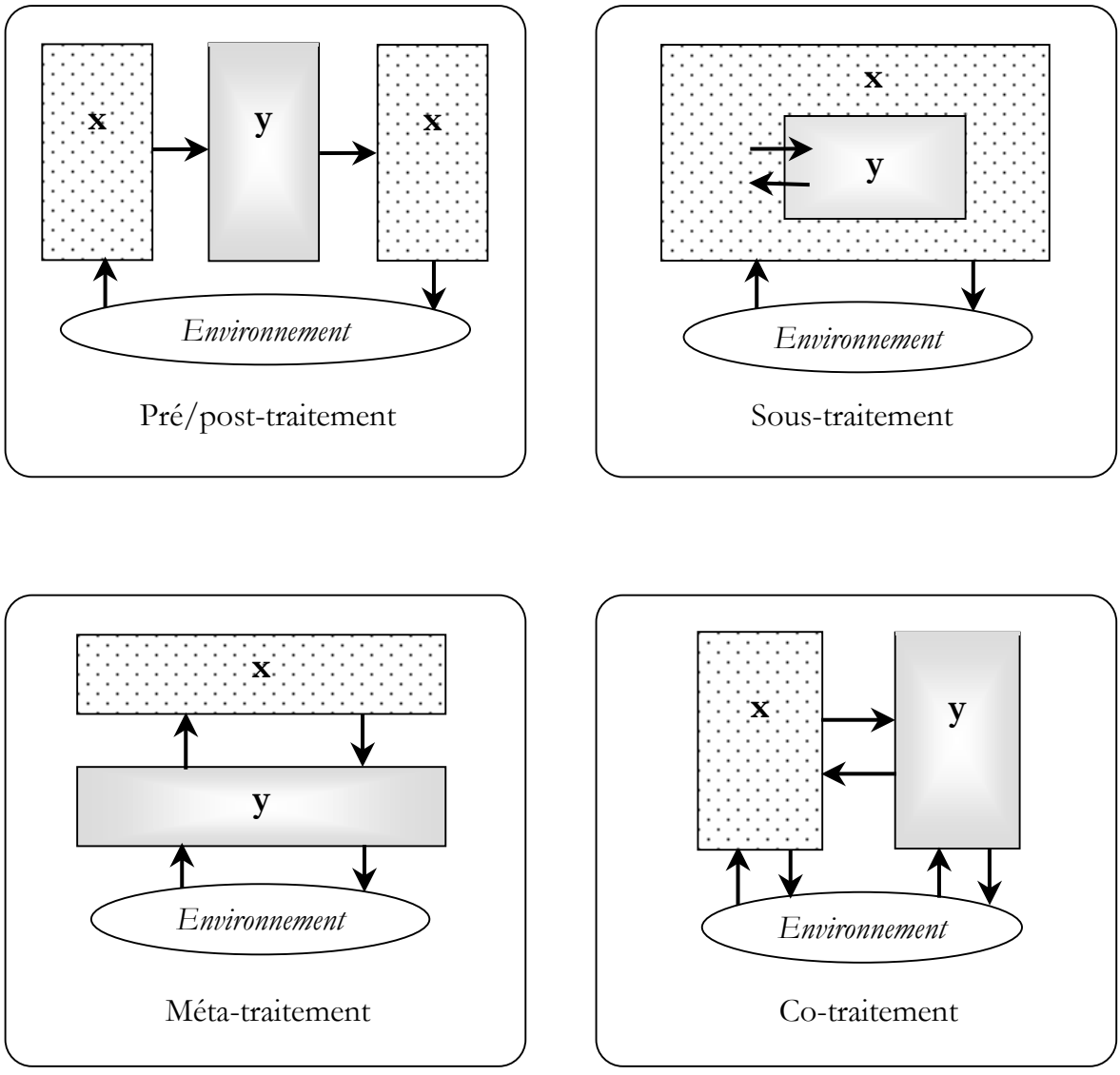

Figure 5.2 : Modes d'interaction entre deux modules x et y,
avec x, y ∈ { symbolique, connexionniste } mais x ≠ y.

- **Mode pré/post-traitement** *(chainprocessing)* : le traitement se fait séquentiellement par un module, puis par l'autre. Donc, les deux modules opèrent en séquence.

- **Mode sous-traitement** (*subprocessing*) : le traitement principal est assuré par un des modules, qui utilise l'autre comme un partenaire de service pour résoudre des points particuliers. Donc, l'un des deux modules est subordonné à l'autre, et de plus n'a pas de contact direct avec l'environnement.

- **Mode méta-traitement** (*metaprocessing*) : un des deux modules joue le rôle de résolveur de problèmes, et l'autre intervient à un méta-niveau (surveillance, contrôle, amélioration des performances).

- **Mode co-traitement** (*coprocessing*) : les modules sont sur un même pied d'égalité, coopèrent pour résoudre un problème, et chacun peut interagir directement avec l'environnement. Les deux modules peuvent proposer des solutions qui seront ensuite évaluées par un module tiers.

5.3 Exemples de modèles hybrides modulaires

Nous consacrons cette section à l'étude de quelques exemples de modèles hybrides modulaires développés dans le cadre l'intégration neuro-symbolique, tels que le système de surveillance respiratoire de Ciesielski et al. [69], le système de diagnostic de pannes de Becraft et al. [70], et le système de pilotage d'un avion de Handelman et al. [71].

5.3.1 Système de surveillance respiratoire

Les capacités de classification des réseaux de neurones peuvent être par exemple utilisés pour fournir des faits symboliques traités ensuite par un système expert. C'est le cas notamment du perceptron dans le système hybride de Ciesielski et al. [69], utilisé dans le domaine de la surveillance temps réel de patients ayant des difficultés respiratoires, et dont le principe est résumé dans la figure 5.3.

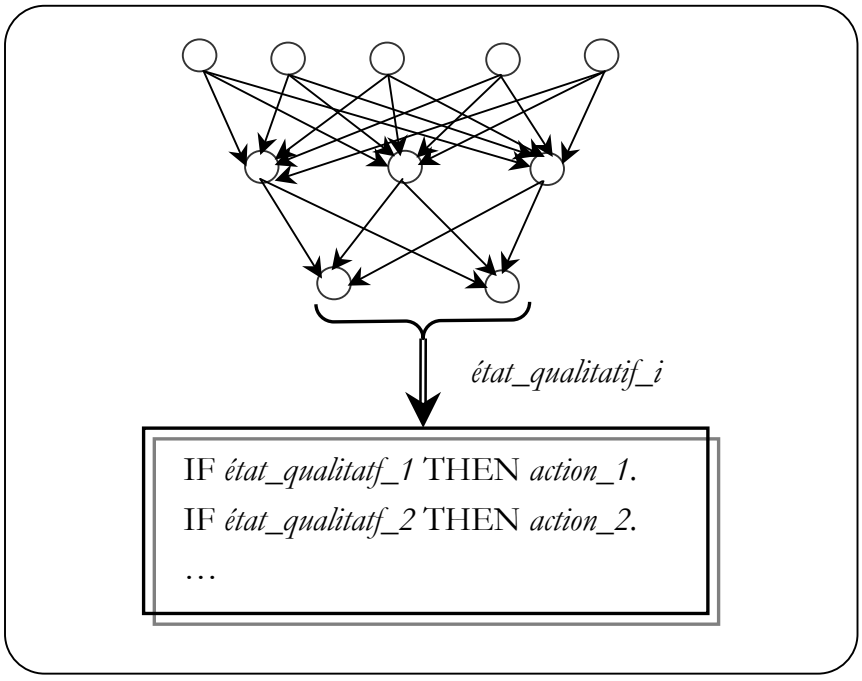

Figure 5.3 : Principe simplifié d'un système modulaire à couplage faible du type de celui de Ciesielski et al. [69].

Connaissant l'évolution de la pression dans les poumons et les voies respiratoires du patient, le système hybride détermine les tâches à effectuer pour éviter des complications respiratoires. Ces pressions sont mesurées par des sondes toutes les 15

secondes. Durant la réalisation du premier système expert symbolique, les auteurs ont constaté que la plus grande difficulté consistait à obtenir des experts une formulation précise de certains états qualitatifs, par exemple : *la pression est constante*, *la pression augmente lentement* ou *la pression augmente rapidement*, tandis que l'obtention de règles de plus haut niveau du type « Si *état qualitatif* Alors *action* », recommandant des actions à prendre à partir des états qualitatifs, posait moins de problèmes. La détermination de ces états qualitatifs à partir des données de pression est donc très difficile. D'où l'idée d'entraîner un RNA à reconnaître ces situations, et d'utiliser les sorties du réseau comme faits dans le système expert. Pour cela, un réseau multicouches, composé de 20 unités d'entrée, de deux couches cachées de 10 et 8 unités respectivement, de 6 unités de sortie, est entraîné séparément avec la méthode de la rétro-propagation du gradient sur une cinquantaine d'exemples.

Les entrées du réseau sont les données fournies par les sondes, et ses sorties sont associées chacun à un état qualitatif. Puis en cours d'utilisation, à chaque intervalle de temps, les sorties du réseau sont passées en entrée au système expert, dans le sens où l'interprétation symbolique associée à l'unité de sortie qui est la plus activée est stockée dans la mémoire de travail du système expert.

D'après les auteurs, le système hybride donne des résultats meilleurs que leur premier système expert symbolique :

- un taux de succès supérieur, 97,5 % contre 74,5 % ;
- détection plus rapide des problèmes respiratoires ;
- effort de développement inférieur, 2 mois contre 3 mois.

Ce système est donc du type pré/post-traitement, puisque les deux modules connexionniste et symbolique fonctionnent consécutivement. Il est à couplage faible, car la communication entre modules est simple et unidirectionnelle.

5.3.2 Système de diagnostic de pannes

Un système de diagnostic de pannes constitué d'un module symbolique et d'un module connexionniste, appelé INNATE/QUALMS [70], est appliqué à une usine de distillation. Le module connexionniste, INNATE, est une cascade à deux niveaux de réseaux de type perceptron multi-couches (voir figure 5.4).

Les entrées du module INNATE sont les données des différents capteurs, et ses sorties correspondent aux pannes candidates. A chaque panne est associé un coefficient de pondération selon l'activation du neurone qui lui correspond. Plus précisément, le réseau du premier niveau est chargé de localiser la panne dans une des différentes unités

de l'usine (à chaque neurone de sortie est associée une certaine unité). Les réseaux du second niveau qui prennent en entrée des données complémentaires sont entraînés à déterminer l'ensemble des pannes candidates : à chaque unité de l'usine est associé un réseau. Avec cette architecture en deux niveau, le module connexionniste est capable de proposer au moins une localisation pour la panne, même s'il n'arrive pas complètement à la déterminer.

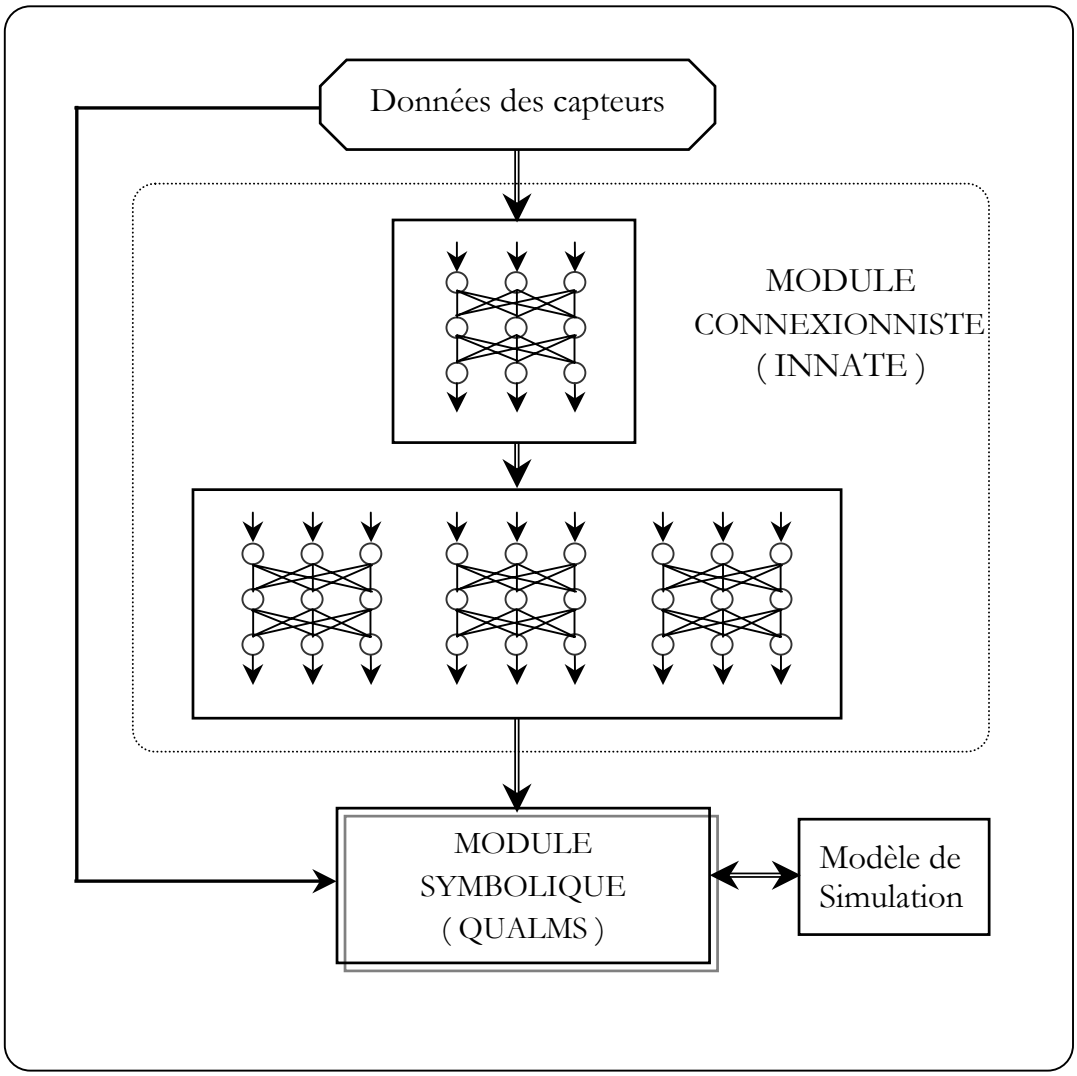

Figure 5.4 : Principe simplifié d'un système hybride modulaire à couplage moyen du type de celui de Becraft et al. [70].

Le module symbolique, QUALMS, est un système expert classique utilisant un modèle pour simuler l'usine de distillation à surveiller. Il exploite des données symboliques qui

ne sont pas fournies par les capteurs. Les deux modules sont couplés de la manière suivante :

- le module symbolique (système expert) reçoit des données concernant la panne, puis active le module connexionniste ;
- le module connexionniste génère un ensemble de pannes candidates pondérées qui seront communiquées au module symbolique ;
- le module symbolique a la possibilité de confirmer le diagnostic neuronal, ou de proposer une autre solution.

Les auteurs ont testé le système sur plusieurs scénarios, avec du bruit dans les capteurs, et des fautes nouvelles. Dans les deux cas, le système donnait des réponses satisfaisantes. Le gain apporté par ce système est la robustesse (résistance aux données bruitées ou nouvelles).

Ce système est du type sous-traitement, puisque le module symbolique utilise le module connexionniste qui génère l'ensemble des pannes candidates. Il est à couplage moyen, car le module symbolique active le module connexionniste puis récupère les résultats de ce dernier, mais sans partage de structures (communications bi-directionnelles).

5.3.3 Système de pilotage d'un avion

L'objectif du système hybride proposé par Handelman et al. [71], appelé RSA2 (*Robotic Skill Acquisition Architecture*), est d'intégrer des systèmes à bases de connaissances et des réseaux de neurones pour faire du contrôle en robotique (voir figure 5.5). Cette intégration est fondée sur les mécanismes qui permettent à un individu d'apprendre peu à peu des réponses complexes et pourtant efficaces lorsqu'on lui donne, pour accomplir une certaine tâche, des explications verbales, des exemples de mouvements typiques qu'elle nécessite, et du temps pour s'entraîner.

Les auteurs utilisent des analogies avec des modèles biologiques pour définir quatre grandes phases de fonctionnement de leur système.

- **Phase déclarative :** les composants à base de connaissances effectuent la tâche de pilotage, d'une manière approximative.

- **Phase hybride d'apprentissage :** les composants neuronaux commencent tout d'abord à apprendre grâce aux exemples fournis par les systèmes à base de connaissances, à accomplir des parties de la tâche de pilotage, mais sans contribuer à sa réalisation.

- **Phase hybride de pilotage :** après apprentissage, les composants neuronaux se mettent à partager la responsabilité du contrôle effectif, leur performances s'améliorant au fil du temps ; ce partage est facilement réalisé car la commande est la somme algébrique des commandes des divers composants.

- **Phase réflexive :** durant cette dernière phase, le contrôle effectué par les réseaux de neurones est optimisé par apprentissage renforcé (amélioration des performances par expérience).

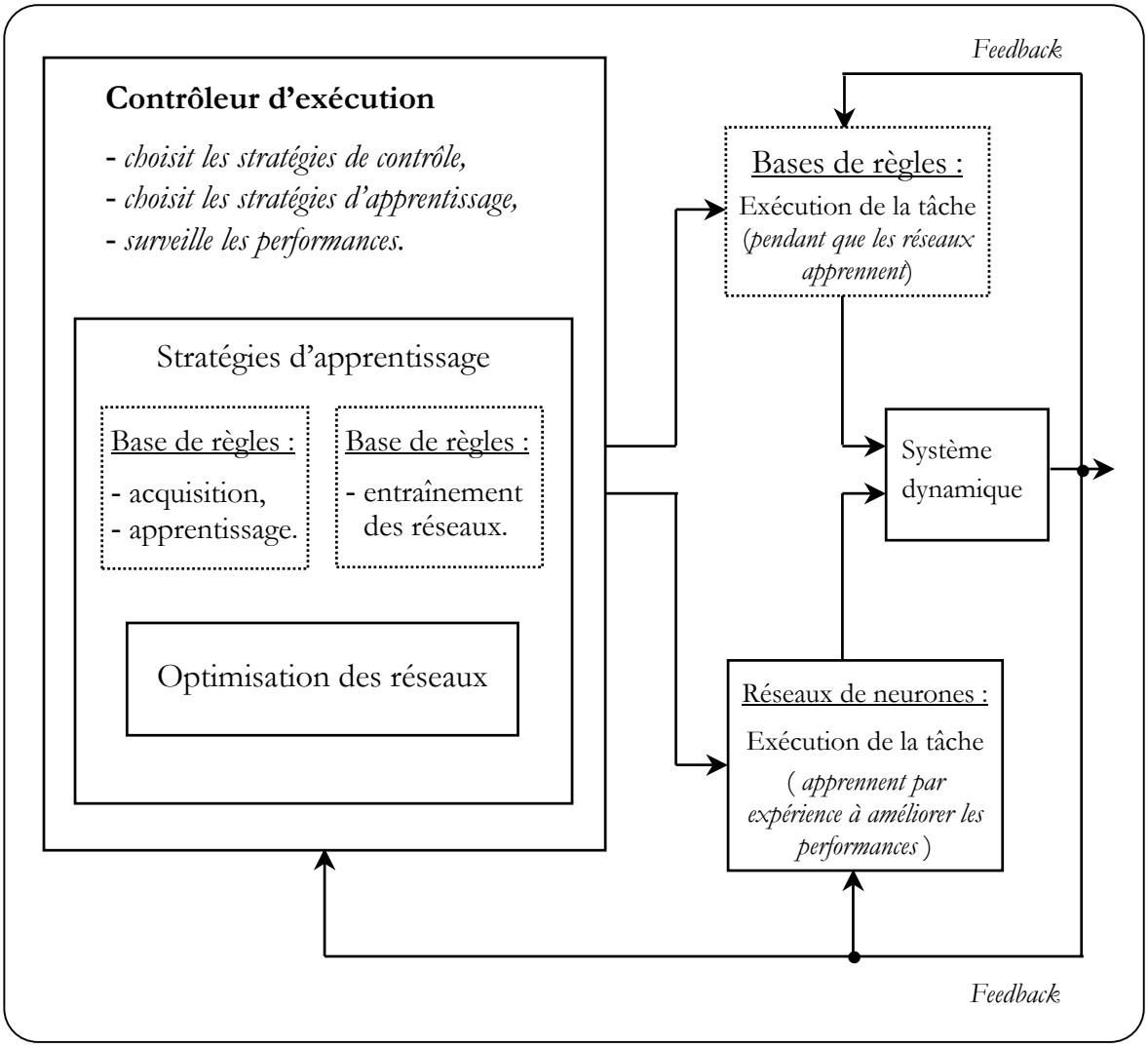

Figure 5.5 : Architecture du système hybride modulaire RSA2.

Ce système est appliqué dans une simulation de contrôle d'un avion durant les phases d'approches et d'atterrissage. Diverses expériences montrent que les performances sont meilleures en fin de phase hybride que dans la phase déclarative (l'avion atterrit mieux en termes de vitesse, distance, etc.), et satisfaisantes en phase réflexive, dans laquelle les réseaux sont utilisés seuls.

Pour les auteurs, un tel système est conçu de manière à pouvoir intégrer des techniques de contrôle conventionnelles, quand elles peuvent s'appliquer, tout en utilisant des techniques d'apprentissage neuronal pour réduire certains des coûts de la modélisation de systèmes complexes. D'autre part, ils supposent qu'une telle architecture, d'inspiration biologique, fournira une bonne interface homme-machine, permettant à la fois non seulement de dire quoi faire à une machine, mais aussi de lui montrer.

Ce système hybride est à couplage moyen assez complexe et qui présente deux types d'interactions, à savoir méta-traitement et co-traitement. La composante méta-traitement a pour fonction principale de surveiller les erreurs d'apprentissage des réseaux et de modifier des paramètres de la fonction d'apprentissage. Cela peut conduire des méta-règles à décider de remplacer des groupes de règles de la partie exécution par des réseaux lorsque ceux-ci sont jugés suffisamment entraînés. Par contre, la composante co-traitement est représentée par le partage de la responsabilité du contrôle effectif entre les deux modules symbolique et connexionniste.

5.4 Conclusion

Les systèmes hybrides modulaires sont les plus puissants et les plus utilisés parmi les systèmes hybrides. Cependant, un continuum de complexité de systèmes modulaires existe et dépend du degré de couplage et de communication inter-modules exigés par le domaine d'application. Beaucoup d'anciens systèmes exige seulement un canal uni-directionnel limité de communication entre les modules. Quelques systèmes récents plus complexes ont été développés pour exécuter des tâches avec un degré élevé d'intégration. De tels systèmes se composent de plusieurs modules coopératifs avec plusieurs canaux de communication. Bien que les communications entre les modules deviennent plus complexes, les différents modules restent conceptuellement séparés en tant que réseau de neurones et composants symboliques.

Chapitre 6

Les modèles hybrides transformationnels

Les modèles hybrides transformationnels transforment des représentations symboliques en représentations connexionnistes ou vice versa. Le processus de transformation peut être une compilation complète ou partielle de l'information d'une forme à l'autre. Des travaux de recherche très récents [49, 60, 61, 77] montrent les capacités importantes des modèles hybrides transformationnels à résoudre des problèmes complexes.

Dans ce chapitre, nous donnons les motivations des chercheurs pour ce type de modèles, les architectures possibles, et nous étudions quelques exemples de systèmes transformationnels représentatifs du domaine.

6.1 Motivations

Le processus transformationnel permet la construction de systèmes hybrides qui peuvent opérer entre les deux niveaux neuronal/symbolique de représentation des connaissances. De tels flux bi-directionnels d'informations permettent un processus interactif de compilation, d'extraction et de raffinement de connaissances symboliques. La connaissance symbolique manipulée par ces modèles est souvent représentée par des règles de production.

La puissance des modèles hybrides transformationnels réside dans la possibilité de construire des architectures combinant les avantages du traitement symbolique et du traitement neuronal dans un même système. De tels systèmes hybrides sont essentiellement des réseaux de neurones dont les neurones correspondent aux concepts de haut niveau. Ces réseaux peuvent apprendre dans un processus qui est semblable à celui des réseaux supervisés au moyen d'exemples d'entraînement et souvent en utilisant des algorithmes de la descente du gradient. En outre, ces systèmes peuvent également modifier l'architecture initiale et les connaissances du réseau de neurones, par conséquent ils sont souvent décrits en tant que réseaux de neurones à base de connaissances (*KBNN : Knowledge-Based Neural Networks*).

Ces architectures transformationnelles possèdent un certain nombre de caractéristiques intéressantes, de niveau élevé qui permettent aux réseaux de neurones d'exécuter les fonctions suivantes :

- La possibilité d'un apprentissage incrémental, ce qui signifie que des réseaux de neurones n'ont pas besoin d'être ré-entraînés avec toutes les données d'entraînement initiales en plus des nouvelles données acquises.
- L'inclusion de la connaissance antérieure aura pour effet d'accélérer le processus d'apprentissage et sera utile dans des situations où les exemples d'entraînement sont rares. Ceci s'appelle compilation (insertion), extraction et raffinement de la connaissance dans plusieurs systèmes [61, 76].
- Une architecture plus déterministe est possible plutôt qu'un processus empirique pour déterminer une bonne architecture (nombre de couches, nombre d'unités cachées, etc.).

Les travaux expérimentaux menés par de nombreux chercheurs sur différentes architectures neuronales à base de connaissances ont donné des résultats très intéressants [74]. Ils montrent une bonne performance en terme d'exactitude de classification, de la vitesse d'entraînement, du raisonnement avec du bruit et absence de données, et une bonne capacité de généralisation avec de petits ensembles d'exemples d'entraînement.

6.2 Classification des modèles hybrides transformationnels

Plusieurs modèles hybrides transformationnels incorporent dans leur architecture les phases de compilation, d'extraction et de raffinement des connaissances (figure 6.1). L'utilisation des connaissances du domaine sous forme de règles peut être employée pour définir l'architecture du réseau de neurones initial. Ce réseau peut être raffiné par un apprentissage inductif à l'aide des exemples d'entraînement fournis. La performance améliorée du réseau de neurones à base de connaissances ainsi obtenue peut être employée pour raffiner la base de règles initiale par un processus d'extraction des connaissances à partir du réseau de neurones. Les nœuds et les connexions du réseau de neurones à base de connaissances correspondent à la signification symbolique de la connaissance initiale du domaine et sont facilement convertis de nouveau dans un format symbolique. Le processus entier peut être répété plusieurs fois jusqu'à ce que le système montre une performance globale améliorée.

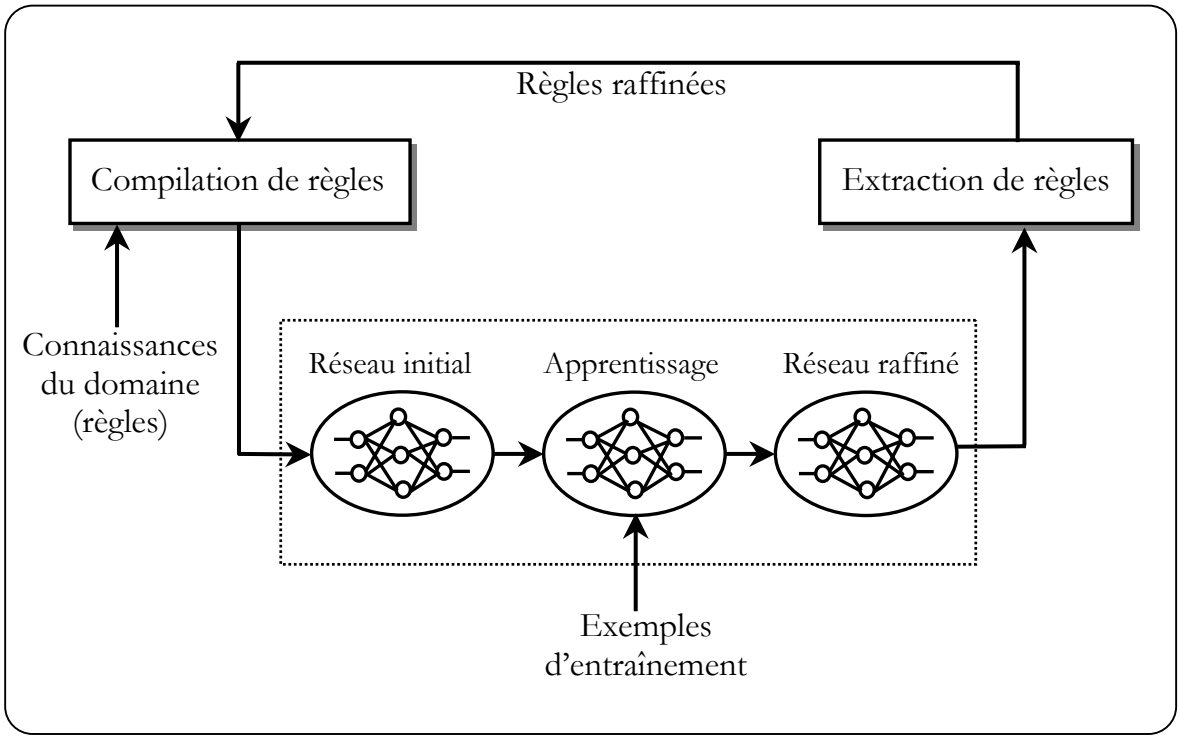

Figure 6.1 : Cycle de compilation, d'extraction et de raffinement de règles.

La combinaison des connaissances du domaine et de l'apprentissage inductif s'est avérée être un facteur important dans le succès des systèmes transformationnels. Les deux prochaines sous-sections décrivent les techniques employées pour mettre en application les processus qui se produisent dans les modèles hybrides transformationnels.

6.2.1 Compilation de règles

Une technique qui peut convertir une représentation symbolique en une représentation neuronale est la compilation de règles. Le processus de compilation de règles nous permet d'obtenir un réseau de neurones qui donne exactement les mêmes réponses que nous pouvons obtenir à partir des règles qui ont servi à le créer. Le processus de compilation de règles passe en général par les étapes suivantes :

- **Définir la structure du réseau :** cette étape permet d'établir les correspondances entre un ensemble de règles et un réseau de neurones. La figure 6.2 montre les correspondances entre ces deux types de représentations. Ainsi, la structure du réseau peut être créé à partir des règles symboliques disponibles.
- **Définir les paramètres du réseau :** une fois que toutes les unités sont bien disposées dans les différentes couches du réseau selon ces correspondances, on passe à la définition des poids des connexions ainsi que du seuil (biais) de chacune des unités. Cette définition de paramètres varie d'un système à l'autre.
- **Compléter le réseau :** une fois que la structure du réseau est construite, des unités et des connexions peuvent y être ajoutées manuellement sur indication de l'expert.

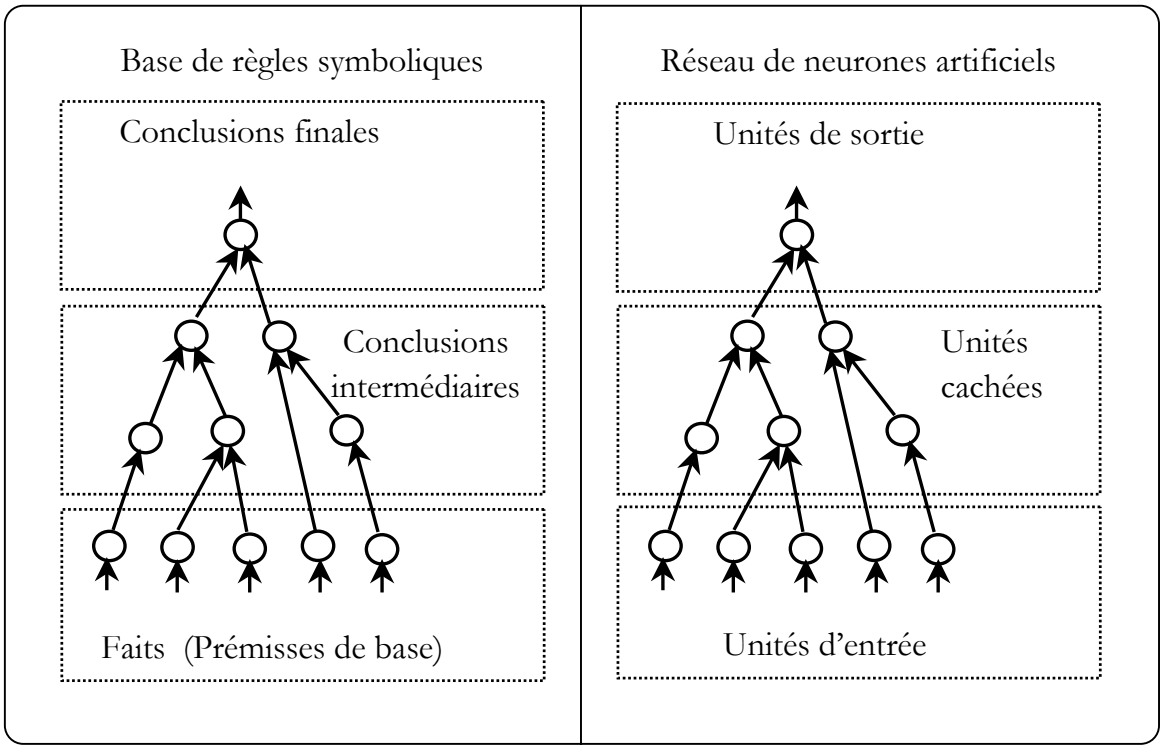

Figure 6.2 : Correspondance entre la structure d'une base de règles et d'un RNA.

Ce processus de compilation de règles a les avantages suivants :

- Permet d'obtenir un réseau de neurones qui donne exactement les mêmes réponses que nous pouvons obtenir à partir des règles qui ont servi à le créer.
- Cette façon de créer les réseaux permet de simplifier beaucoup le processus de spécification et d'apprentissage d'un réseau de neurones.
- Une architecture initiale du réseau est obtenue de façon automatique et nous n'avons plus de soucis par rapport à la détermination du nombre de couches, d'unités et d'interconnexions du réseau.
- Les règles une fois insérées donnent au réseau de neurones des connaissances de départ. Ainsi, le problème du choix aléatoire des poids initiaux d'un réseau et de ses conséquences (souvent négatives) sur l'apprentissage peut être contrôlé et minimisé.

Towell a montré dans ses recherches que ce type d'approche (compilation de règles) permet de réduire le temps d'apprentissage et le nombre d'exemples nécessaires à un réseau pour bien apprendre à résoudre un problème [60, 61].

6.2.2 Extraction de règles

Une manière directe de convertir une connaissance neuronale en une connaissance symbolique est par l'extraction de règles. L'extraction de règles est un processus de classification qui découvre les différents schémas de liens (les dépendances) entre les unités d'entrée, les unités cachées et les unités de sortie du réseau de neurones. Ces schémas sont alors représentés sous forme de règles « *SI .. ALORS .. Sinon* » en prenant en compte les unités d'entrée les plus importantes agissant comme antécédents (prémisses) de la règle (voir figure 6.3). La détermination de ces schémas peut être basée sur un ensemble de techniques qui analysent les poids et les seuils du réseau de neurones [77].
Une partie importante dans le processus d'extraction de règles est consacrée au traitement des règles dérivées pour s'assurer qu'un ensemble compact de bonne qualité décrit en juste proportion le réseau de neurones initial.

- **Règle de subsumption :** les algorithmes d'extraction de règles peuvent produire un ensemble de règles supérieur à celui normalement nécessaire pour décrire le réseau de neurones. Subsumption est une manière de réduire le nombre de telles règles superflues en optant pour des règles plus générales. On dit qu'une règle subsume une autre si elle est plus générale ou plus spécifique. Dans ce cas la règle spécifique peut être supprimée.

- **Règle de résolution :** si deux ou plusieurs règles ont une même conclusion et un même nombre d'antécédents mais diffèrent par un antécédent étant une négation alors elles peuvent être réduites en une seule règle.

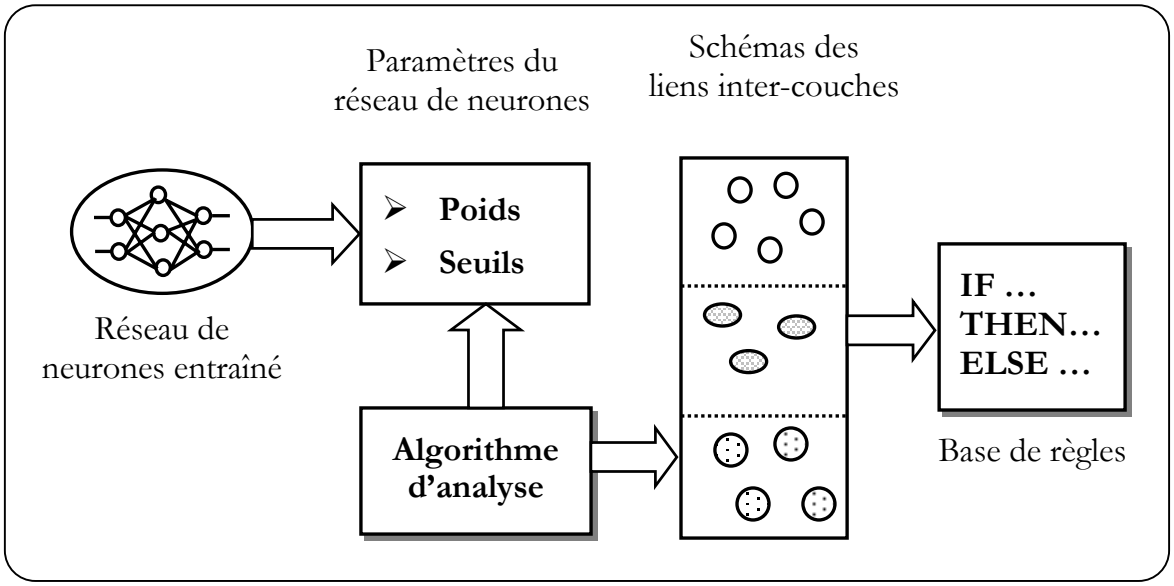

Figure 6.3 : Processus d'extraction de règles.

Le processus d'extraction de règles a les avantages suivants :

- Avoir un moyen d'explication du raisonnement en examinant les règles extraites pour les différentes configurations d'entrée.
- Des insuffisances dans l'ensemble initial de formation peuvent être identifiées, ainsi la généralisation du réseau peut être améliorée par addition de nouvelles classes.
- On ayant des règles extraites d'un réseau de neurones nous avons une base de règles qui peut être insérée à nouveau dans un autre réseau résolvant un problème similaire.

6.3 Exemples de modèles hybrides transformationnels

Cette section est consacrée à l'étude de deux exemples de modèles hybrides de type transformationnel les plus représentatifs du domaine : les réseaux KBANN (*Knowledge-Based Artificial Neural Networks*) de Towell et Shavlik [60, 61, 82, 83], et le système INSS (*Inremental Neuro-Symbolic System*) de Osorio [49].

6.3.1 Réseaux KBANN

Les réseaux KBANN ont été développés par Towell et Shavlik [60, 61, 82, 83]. Les réseaux KBANN classés sous l'approche hybride transformationnelle permettent la complilation, l'extraction et le raffinement de règles symboliques. Cette approche permet une intégration des connaissances théoriques et empiriques d'une façon simple et avec des résultats pratiques très bons.

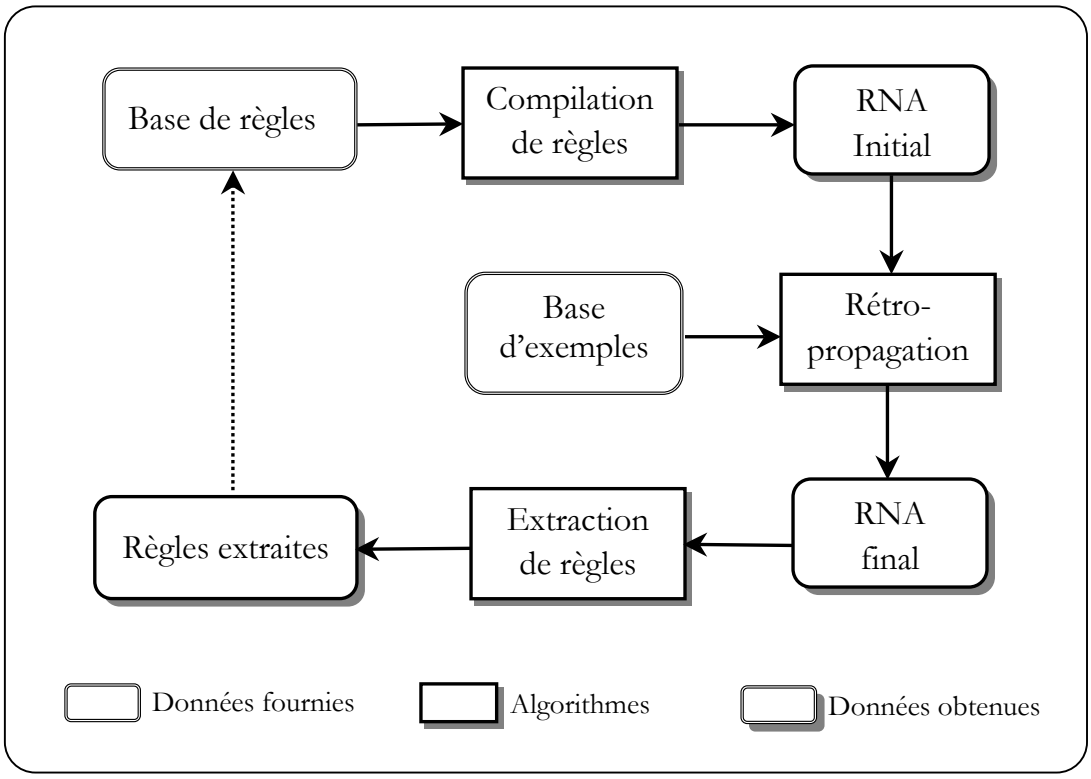

Figure 6.4 : Schéma d'intégration utilisé dans les réseaux KBANN.

Les réseaux KBANN sont le résultat de l'implémentation de trois algorithmes : le premier algorithme assure la traduction de règles symboliques vers un RNA de type PMC ; le deuxième algorithme fait le raffinement des connaissances du réseau par apprentissage d'une base d'exemples à travers l'utilisation de la rétro-propagation ; le troisième algorithme réalisera l'extraction des règles raffinés à partir du réseau qui a été entraîné. De cette façon, ces trois modules permettent de faire des transferts de connaissances entre un module symbolique et le module connexionniste. On peut éventuellement réintroduire les règles raffinées dans le réseau. La figure 6.4 présente un schéma des trois modules principaux qui constituent ce type d'approche.

Selon Towell [61], les réseaux KBANN peuvent soit éliminer complètement soit réduire significativement une grande partie des problèmes inhérents des réseaux de neurones classiques et des systèmes symboliques. Cette approche a été testée sur différentes applications, tel que des problèmes réels du domaine de la biologie moléculaire et dans la modélisation du développement du raisonnement géométrique chez l'enfant. Les résultats obtenus avec ce système sont très encourageants.

Notre étude des réseaux KBANN sera concentrée sur les algorithmes de compilation et d'extraction de règles. Ces algorithmes transforment un module symbolique en un module connexionniste et vice versa.

6.3.1.1 Compilation de règles dans KBANN

Le transfert de connaissances théoriques (règles symboliques) vers le module neuronal est réalisé dans les réseaux KBANN par un algorithme de compilation de règles approximativement correctes. Les règles utilisées par KBANN sont exprimées par des clauses de Horn avec une notation décrite dans le tableau 6.1. Deux restrictions sont imposées sur l'ensemble des règles :

- les règles doivent exprimer des propositions et ne doivent pas contenir des variables,
- les règles doivent être acyclique ; c'est-à-dire qu'une règle ne doit pas avoir un conséquent qui intervient dans un de ses antécédents (directement ou indirectement).

Donc, les règles symboliques utilisées par KBANN sont d'ordre 0 codées de façon hiérarchique.

<Conséquent> :- <Antécédent>, <Antécédent>, …, <Antécédent>

<Conséquent> : c'est la partie qui représente la conclusion de la règle.

<Antécédent> : c'est la partie qui représente les prémisses. Un antécédent peut être composé par <Opérateur> suivi d'une formule atomique.

<Opérateur> : il s'applique sur une formule atomique. L'opérateur le plus utilisé est le 'Not', mais les réseaux KBANN acceptent aussi les opérateurs 'Greater-Than' et 'Less-Than'.

Exemple: La règle symbolique, Si (A et B) ou (A et Non(D)) Alors C est décrite par deux règles, C :- A, B.
C :- A, Not D.

Tableau 6.1 : Syntaxe et exemple d'une règle symbolique dans KBANN.

Le processus de compilation de règles dans KBANN est réalisé par un algorithme divisé en quatre étapes principales, à savoir :

1. **Réécriture des disjonctions :** les disjonctions sont réécrites de façon qu'une règle n'ait qu'une seule prémisse (formule atomique, sans négation) comme antécédent. La réécriture des règles ne change pas les connaissances ni le résultat de son application, elle va changer seulement la représentation écrite des règles.

   Voici un exemple de réécriture des disjonctions :

Avant réécriture	Après réécriture
A :- B, C, D.	A :- A1. A1 :- B, C, D.
A :- D, E, F, G.	A :- A2. A2 :- D, E, F, G.

2. **Construction du réseau :** Cette étape permet de trouver les correspondances entre un groupe de règles et un réseau de neurones. La structure du réseau est ainsi créé à partir des règles et les unités sont bien disposées dans les différentes

couches du réseau selon ces correspondances. Puis, les poids des connexions ainsi que le seuil de chacune des unités sont définis comme suit :

- Les poids des connexions originaires d'une prémisse positive reçoivent des valeurs égales à : $+W$.
- Les poids des connexions originaires d'une prémisse niée reçoivent des valeurs égales à : $-W$.
- Les seuils des unités qui représentent les conclusions obtenues à partir d'une conjonction reçoivent des valeurs égales à : $(-P + 0.5) * W$.
- Les seuils des unités qui représentent les conclusions obtenues à partir d'une disjonction reçoivent des valeurs égales à : $-0.5 * W$.
- La valeur W est une valeur constante spécifiée par l'utilisateur. Plus grand est W, plus grand sera l'écart entre une conclusion positive et une conclusion négative. KBANN utilise une valeur $W=4$, qui a été déterminée empiriquement et semble donner des bons résultats avec ce type de réseau.
- La valeur P indique le nombre de prémisses positives (sans négation) présentes dans la liste d'antécédents d'une unité qui représente une conclusion.

Une preuve mathématique formelle et détaillée de l'exactitude des méthodes de transformation des règles vers les réseaux KBANN est présentée par Towell [61].

3. **Ajout d'unités et de connexions :** une fois que la structure du réseau est construite, des unités peuvent y être ajoutées. Cette étape est optionnelle dans la construction des réseaux KBANN. On peut ajouter dans la couche d'entrée des unités qui représentent des attributs non référencés dans la base de règles. On peut aussi ajouter des unités dans la couche cachée en cas d'indication de l'expert (procédure manuelle d'insertion). Après l'ajout d'unités, on passe à l'ajout des connexions. Les connexions peuvent être ajoutées de façon à : lier toutes les unités appartenant à deux couches adjacentes ; lier toutes les unités d'une couche avec les unités des couches supérieures ; lier chacune des entrées avec chacune des unités cachées ou de sortie. Les connexions ajoutées reçoivent des poids et seuils égaux à zéro, et par conséquence, ne changent pas le comportement du réseau par rapport aux règles introduites.
4. **Perturbation des poids :** l'étape finale de la transformation de règles en réseau est réalisée par une perturbation de tous les poids du réseau. Cette perturbation consiste à additionner une valeur aléatoire très petite à chacun des poids des connexions. La perturbation introduite dans les poids est si faible qu'elle n'a aucun

effet sur le comportement des réseaux KBANN avant l'apprentissage. Cependant, cette méthode permet d'éviter les problèmes d'apprentissage résultant d'une symétrie du réseau. La figure 6.5 présente un exemple de compilation de règles.

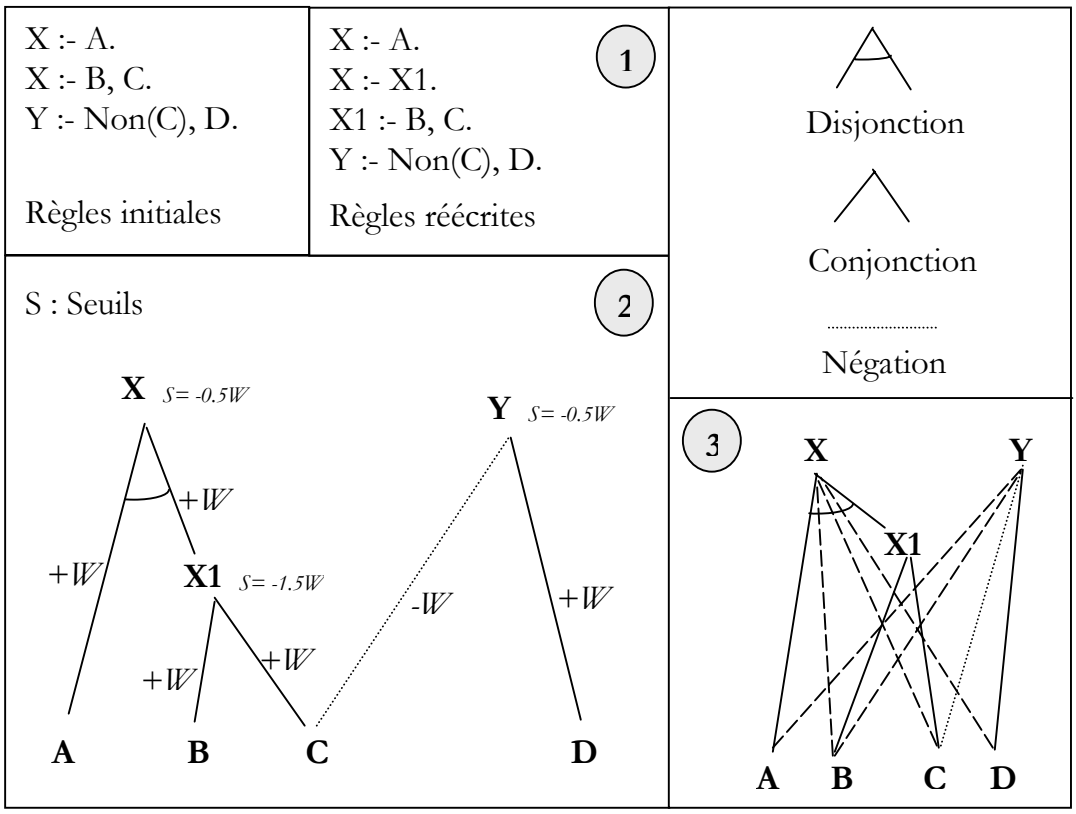

Figure 6.5 : Principales phases de transformation de règles dans KBANN.

Les réseaux KBANN ainsi obtenus sont des réseaux de type PMC, où les valeurs d'entrée de chaque unité sont usuellement comprises dans l'intervalle [0, 1] et les sorties de toutes les unités sont obligatoirement comprises dans l'intervalle [0, 1]. Les réseaux KBANN utilisent une fonction de transfert du type sigmoïde asymétrique. Dans ces réseaux, une valeur d'entrée ou de sortie égale à 0.0 symbolise une prémisse/conclusion fausse et une valeur égale à 1.0 symbolise une prémisse/conclusion vraie. Les variables avec des attributs manquants sont codées par des valeurs égales à 0.5. Cette valeur indique ainsi une 'indécision' dans la sortie d'une unité.

6.3.1.2 Extraction de règles des réseaux KBANN

Deux méthodes d'extraction de règles à partir des RNA ont été utilisées avec les réseaux KBANN. La première méthode s'appelle *SUBSET* et la deuxième *NofM*. Avant de détailler chacune de ces méthodes, nous allons présenter les propriétés et les

caractéristiques qui sont communes aux deux types de méthodes. Deux hypothèses sont faites par rapport aux réseaux obtenus par compilation et raffinés par apprentissage :

- Les unités des réseaux KBANN peuvent être assimilées à des unités à seuil : lorsque leurs valeurs d'entrée (somme pondérée) sont en dessous du seuil qui leur correspond, leur activation est proche de 0.0, sinon elle est proche de 1.0. La fonction d'activation utilisée est toujours la sigmoïde asymétrique.
- L'apprentissage ne modifie pas d'une façon significative le rôle qu'une unité avait avant l'apprentissage. Les règles extraites à partir d'un réseau conservent les noms attribués aux prémisses et conclusions des règles symboliques.

Les méthodes d'extraction *SUBSET* et *NofM* étudient les relations établies entre l'ensemble des entrées et la valeur du seuil d'activation d'une unité. Une valeur résultante de la somme pondérée des entrées d'une unité supérieure au seuil implique que cette unité va être activée (valeur de l'activation proche de 1.0), et par conséquence on va exprimer cette relation entre les entrées et l'activation de la sortie par une règle. Une valeur résultante de la somme pondérée des entrées d'une unité inférieure au seuil implique que cette unité ne va pas être activée (valeur de l'activation proche de 0.0), et par conséquence on ne va pas exprimer par une règle la relation entre les entrées et la sortie de cette unité.

Algorithme SUBSET d'extraction de règles

La méthode a été nommée *SUBSET* car on cherche les sous-ensembles des poids d'entrée d'une unité qui cumulés permettent de dépasser son seuil d'activation. Donc, le principe de l'algorithme *SUBSET* va être de trouver des combinaisons d'unités entrantes (connectées à l'entrée de l'unité en question) qui doivent être activées ou inactivées pour que l'unité correspondante soit activée. Pour chaque combinaison obéissant à cette définition, on crée et on ajoute à la base de règles la règle ayant en prémisses les noms des unités devrant être activées et les noms des unités devrant être inactivées, et en conclusion, le nom de l'unité étudiée.

On essaie, pour cela, de trouver des sous-ensembles d'unités reliées par des poids positifs dont la somme dépasse le seuil de l'unité étudiée. Pour chaque groupe de poids ainsi exhibés, on regarde parmi les poids négatifs, s'il en existe dont la somme des poids annule les poids précédents plus le seuil. S'il existe de tels poids, alors, les entrées respectives doivent être niées, afin que le groupe de poids positifs extrait active l'unité, sinon, l'activation de ce groupe de poids suffit pour produire l'activation de la sortie de l'unité en question. A cet effet, l'algorithme présenté dans le tableau 6.2 a été développé.

Pour chaque sous-ensemble B_p de poids positifs reliés à l'unité U_i
    Si Somme(B_p) + S_i > 0.0 Alors ajouter B_p à G_p

Pour chaque ensemble $P \in G_p$

    Pour chaque sous-ensemble B_n de poids négatifs reliés à U_i
        Soit Z un nouveau prédicat non utilisé jusqu'à présent
        Si Somme(B_p) + Somme(B_n) + S_i < 0.0 Alors CréerRègle('Z :- B_n')
    Si Z a des antécédents Alors CréerRègle('U_i :- P, non(Z)')
                        Sinon CréerRègle('U_i :- P').

Elimination des règles dupliquées

U_i : Unité i de la couche cachée ou de la couche de sortie
S_i : Seuil d'activation de l'unité i

Tableau 6.2 : Description de l'algorithme SUBSET.

La figure 6.6 présente un exemple de l'application de l'algorithme *SUBSET* à une unité d'un réseau KBANN.

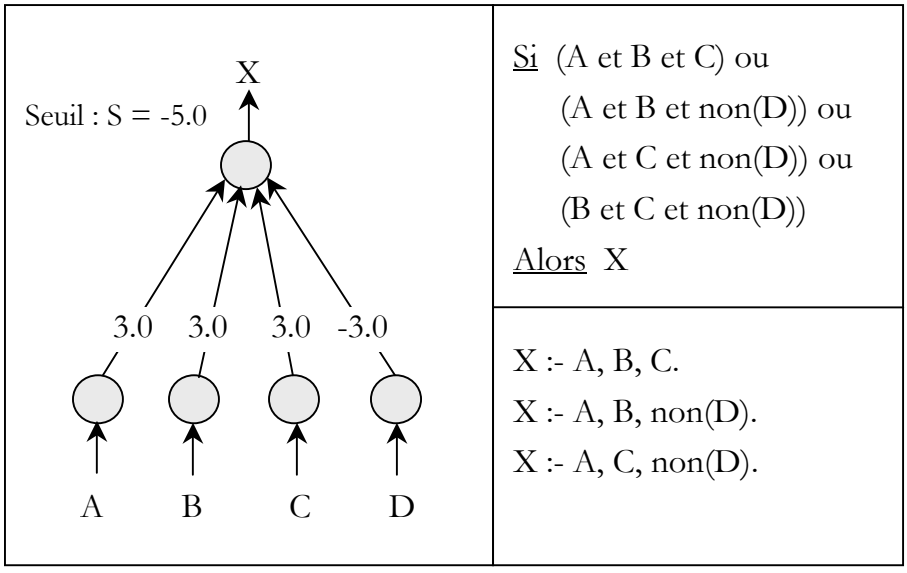

Figure 6.6 : Exemple d'application de l'algorithme SUBSET.

L'avantage de la méthode SUBSET est que les règles qui en sont extraites sont compréhensibles et faciles à exploiter. Par contre, la complexité de cet algorithme impose un temps d'exécution qui peut augmenter significativement en fonction du nombre d'entrées de l'unité analysé, car il exécute une recherche de toutes les possibilités de combinaisons des entrées/poids qui mènent à l'activation de la sortie.

Algorithme NofM d'extraction de règles

Les règles extraites par cette méthode sont exprimées en fonction d'une primitive de base : *NofM*. Cette primitive prend en entrée un entier et une base de faits et rend en sortie un booléen. La formule $NofM(i, P_1, P_2, \ldots, P_n)$ rend vraie (1.0) sa sotie si et seulement si au moins i propositions parmi $P_1, P_2, \ldots, P_n$ sont vraies, sinon elle rend la valeur fausse (0.0). Cette forme a l'avantage d'être plus réduite et nécessite moins de règles pour exprimer les connaissances.

L'algorithme de la méthode *NofM* fonctionne selon le principe suivant, en six étapes :

1. *Regroupement :* pour chaque unité, on partitionne l'ensemble des liaisons d'entrée en classes d'équivalence. Les liaisons d'une même classe d'équivalence portent des poids dont les valeurs sont proches les unes des autres. KBANN utilise une valeur constante de distance inférieur à 0.25 pour regrouper les valeurs des poids des connexions.

2. *Moyenne :* on fixe la valeur des poids des liaisons d'une même classe à la moyenne des poids de cette classe.

3. *Elagage :* on détermine, de façon heuristique, les classes non significatives pour l'unité étudiée, c'est à dire les classes dont la valeur d'activation est indifférente pour l'activation de l'unité dont on extrait les règles. Pour cela, on procède sur une base d'exemples : pour chaque exemple, on calcule l'activation du réseau, et on regarde si, en annulant toutes les activations des poids d'une classe particulière, l'activation de l'unité étudiée est modifiée. Si aucun changement notable n'est observé, alors la classe n'est pas significative pour l'exemple courant. Une fois que toute la base a été examinée, on supprime les classes qui ne sont significatives pour aucun exemple de la base.

4. *Optimisation :* des modifications sur le seuil de l'unité étudiée peuvent être nécessaires, du fait de l'erreur que peut apporter la partition en classes d'équivalence de ses poids d'entrées, ainsi que la suppression des classes non significatives. Pour cela, on recommence un apprentissage, en gelant tous les poids

portés par les liaisons, de manière à ce que seuls les seuils des unités du réseau soient modifiés.

5. *Extraction :* on exprime ensuite, pour chaque unité de réseau, les relations entre l'unité étudiée et ses antécédents sous forme d'une inéquation. Soit *S* le seuil de l'unité étudiée et $C_1, C_2, \ldots, C_n$ les identificateurs des liaisons en entrée, portant les poids $W_1, W_2, \ldots, W_n$, l'inéquation associée à chaque unité est la suivante :

   $$W_1 * \text{number-true}(C_1) + W_2 * \text{number-true}(C_2) + \ldots + W_n * \text{number-true}(C_n) + S > 0$$

   Où *number-true* est une primitive prenant en entrée une liste de faits, et en rendant en sortie un entier qui représente le nombre de faits justes (vrais) parmi ceux de la liste d'entrée.

6. *Simplification :* on réécrit enfin les inéquations sous forme de règles de type *NofM*, en recherchant, dans chaque groupe, le nombre de règles permettant à l'unité responsable de la conclusion d'être activée.
   Par exemple : soit X une unité qui possède un seuil S = -10.0 et des liaisons d'entrée à partir des antécédents identifiés par A, B, C, D, reliés par des connexions de poids égaux à 5.1, et des antécédents identifiés par F, G, H, reliés par des connexions de poids égaux à 3.5. Cette unité est représentée par l'inéquation suivante :

   $$5.1 * \text{number-true}(A, B, C, D) + 3.5 * \text{number-true}(F, G, H) > 10.0$$

   Après simplification, cette inéquation va être représentée par les règles suivantes :

   X :- NofM(2, A, B, C, D).
   X :- NofM(1, A, B, C, D), NofM(2, F, G, H).
   X :- F, G, H.

La figure 6.7 montre un exemple du processus complet d'extraction de règles en appliquant la méthode *NofM*. Cet algorithme fourni des ensembles de règles assez compacts qui sont usuellement beaucoup plus petits que les ensembles de règles obtenus par l'algorithme *SUBSET*. Ceci peut nous permettre d'en réduire le nombre, et ainsi d'augmenter la compréhension des règles. Par contre, les règles de type *NofM* sont moins naturelles pour un utilisateur humain.

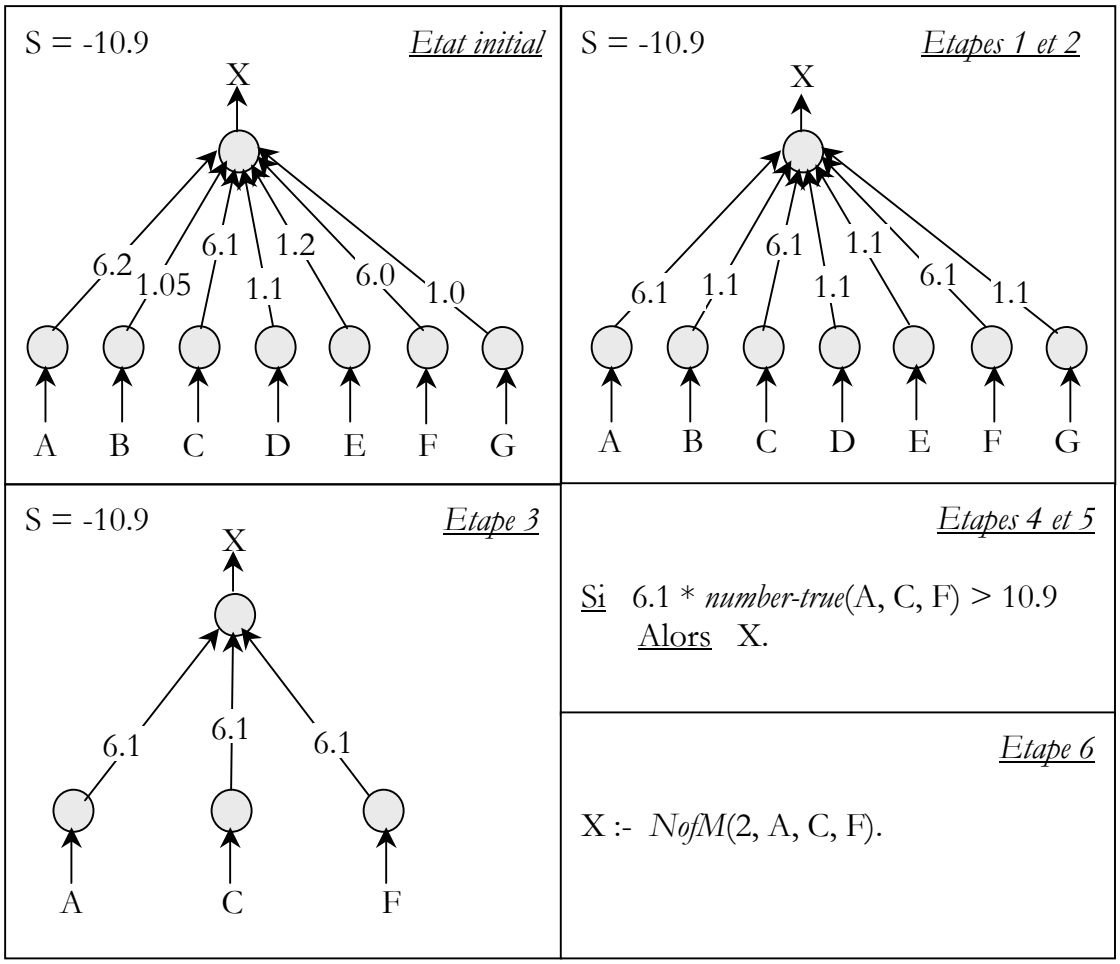

Figure 6.7 : Exemple d'application de l'algorithme NofM.

Les réseaux KBANN est un modèle hybride transformationnel très intéressant, car il permet de transformer un module symbolique en un module connexionniste et vice versa (transfert de connaissances). Par contre, son inconvénient majeur est que, lors de l'apprentissage, il ne modifie pas la structure du réseau, mais seulement les valeurs des poids entre les unités.

6.3.2 Système INSS

Le système INSS (*Incremental Neuro-Symbolic System*) [49], est un système hybride neuro-symbolique qui possède des modules capables de réaliser des transformations partielles du module symbolique au module connexionniste et vice versa (transfert de connaissances). Le système INSS est un outil pour la construction de systèmes experts. Ses principales applications sont dans le domaine des problèmes de classification, de l'aide au diagnostic, ou de l'aide à la décision. Cependant, il a été aussi employé dans des

tâches diverses, telles que le contrôle de robots autonomes et l'étude de modèles cognitifs du comportement humain [49].

Le système INSS a été inspiré par le modèle développé par Towell pour les réseaux KBANN [61]. Il essaie d'améliorer leurs points faibles en ajoutant de nouvelles propriétés. Au contraire du système KBANN qui utilise l'algorithme de la Rétro-Propagation, fondé sur des réseaux statiques, INSS utilise la méthode d'apprentissage Cascade-Correlation [24]. Cet autre algorithme permet l'ajout de nouvelles unités au réseau pendant l'apprentissage – c'est une méthode constructive. Les principales améliorations du système INSS par rapport aux réseaux KBANN, sont : l'utilisation de l'algorithme Cascade-Correlation, l'extraction incrémentale de règles, et la validation des nouvelles connaissances acquises par le système.

Le système INSS est un système hybride qui permet le transfert de connaissances théoriques, représentées par un ensemble de règles symboliques, d'un module symbolique (MS) à un module connexionniste (MC). Pour réaliser ce transfert le système INSS utilise un convertisseur capable de transformer des règles symboliques en réseau de neurones. Le réseau ainsi obtenu pourra subir un apprentissage supervisé à partir d'un ensemble d'exemples (connaissances pratiques). Après l'amélioration des connaissances du réseau par apprentissage de la base de connaissances pratiques, on peut appliquer à ce réseau un algorithme d'extraction de règles. Cet algorithme va extraire les règles qui représentent les nouvelles connaissances acquises par le réseau. Ces connaissances peuvent être réinsérées dans le module symbolique.

Le système INSS est composé de deux modules principaux, le module symbolique et le module connexionniste, et en plus des modules de transfert et de validation de connaissances. La figure 6.8 présente l'architecture de ce système. Le module symbolique est le seul module d'INSS qui n'a pas été développé par Osório [49]. Toutes les autres composantes du système ont été réalisées par l'auteur, à savoir :

- *NeuComp :* il réalise la compilation de règles dans un réseau connexionniste.
- *NeuSim :* c'est le module connexionniste proprement dit du système INSS. Il est responsable de la simulation et de l'apprentissage des réseaux.
- *Extract :* il permet d'extraire des règles à partir des réseaux connexionnistes.
- *Valid :* il fait la validation des nouvelles connaissances acquises par rapport aux anciennes. Il utilise tout l'ensemble de connaissances disponible sur le problème (règles et exemples).

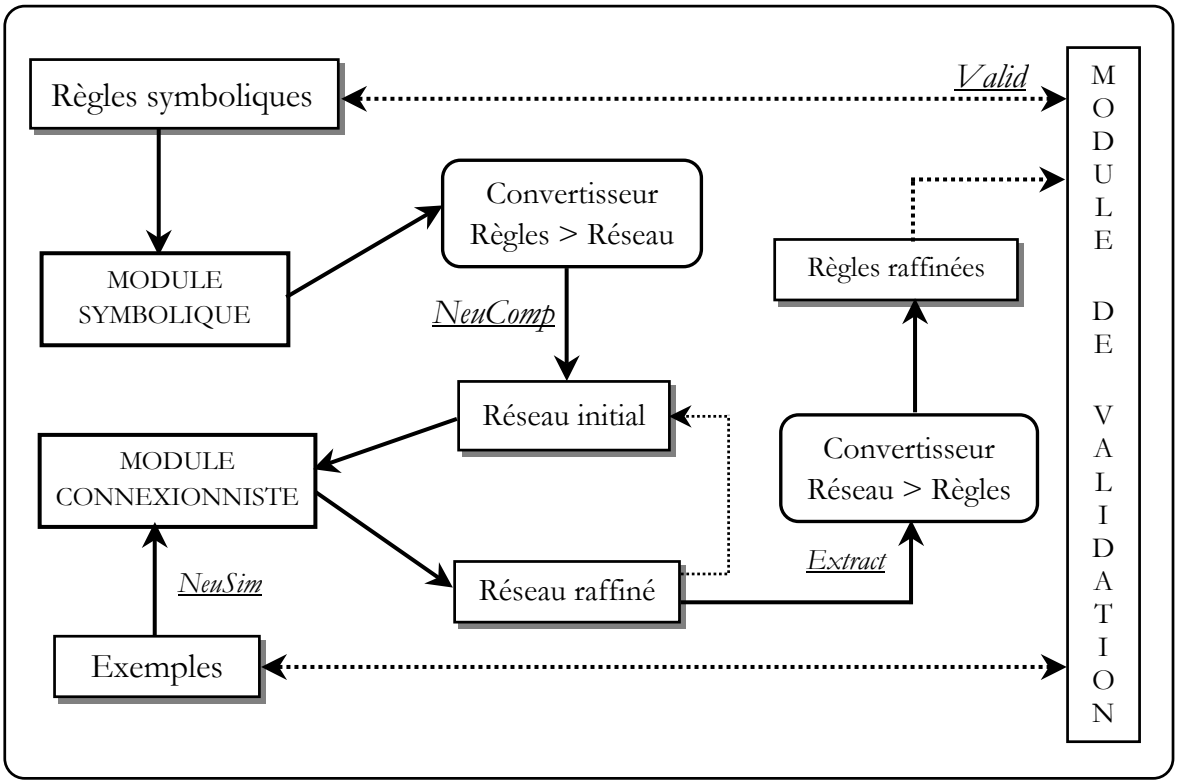

Figure 6.8 : Architecture du système hybride INSS.

Notre étude du système INSS sera concentrée sur les algorithmes de compilation et d'extraction de règles. Ces algorithmes transforment un module symbolique en un module connexionniste et vice versa.

Compilation de règles dans INSS

Le module *NeuComp (Neural Compilar)* permet d'obtenir un réseau de neurones à partir d'un ensemble de règles symboliques. La transformation de règles en réseau suit les mêmes principes que dans le système KBANN (voir section 6.3.1.1).

Les règles symboliques acceptées par *NeuComp* sont d'ordre 0^+ (inclusion de la notion d'intervalle et utilisation de prémisses formées par des valeurs continues). Ainsi, *NeuComp* accepte aussi des règles de production avec des composants du type suivant :

<Attribut> <Opérateur> <Valeur> ou *<Attribut> <Opérateur> <Attribut>*
(avec des opérateurs de comparaison tels que : Plus-Grand-Que, Plus-Petit-Que, Egal)

Le tableau 6.3 montre un résumé de la syntaxe et des différentes règles symboliques utilisées dans *NeuComp*.

Conséquent <- Antécédent [, Antécédent, …] ;
Conséquent <= Antécédent [, Antécédent, …] ;
Conséquent <- Fonction(Antécédent), Fonction(Antécédent, Liste-Paramètres) ;
Conséquent <- Antécédent(True), Antécédent(False) ;

Où,

'<-' : représente une règle adaptable (peut être modifiée au cours de l'apprentissage)
'<=' : représente une règle statique (n'est pas modifiable durant l'apprentissage)
Fonction : choisit un opérateur parmi les suivants :
Equal, In, In_Range, Ntrue, NofM, Not,
LT, Less_Than, LTOE, Less_Or_Equal,
GT, Greater_Than, GTOE, Greater_Or_Equal.

Tableau 6.3 : Résumé de la syntaxe utilisée dans NeuComp.

Une des principales contributions des recherches de Osorio [49] par rapport au modèle proposé par Towell [61], est l'extension de la capacité de traitement des bases de règles utilisées par KBANN. *NeuComp* permet d'insérer des règles d'ordre 0^+ dans ses réseaux, le tout d'une façon compatible avec le mode de fonctionnement des réseaux du type KBANN.

Extraction de règles dans INSS

Le module *Extract* permet d'obtenir des règles symboliques à partir d'un réseau créé par INSS. Les méthodes d'extraction de règles utilisées par Osorio [49] sont basées sur les algorithmes employés avec les réseaux KBANN (*SUBSET* et *NofM* – voir section 6.3.1) auxquels sont apportées des améliorations.
Le module *extract* est un outil capable d'obtenir des règles symboliques à partir d'un réseau préalablement entraîné. Pour ce faire, Osorio [49] réalise tout d'abord une simplification du réseau composée d'une étape de sélection d'unités et d'une étape de sélection de connexions. Une fois simplifié le réseau, on applique les algorithmes d'extraction *SUBSET* et *NofM* pour obtenir les règles symboliques.

Cette manière d'extraction de règles avec l'élagage des réseaux à été testée sur différents problèmes (e.g. les problèmes du Moine) et comparée avec l'extraction de règles sans simplification des unités et des connexions. Les résultats obtenus montrent bien une

réduction très sensible de la complexité du processus d'extraction après la simplification du réseau, ainsi qu'une amélioration de la qualité des règles obtenues (augmentation de la compréhensibilité par la réduction du nombre de règles obtenues) [49].

6.4 Conclusion

Les modèles hybrides transformationnels forment une technique réussie pour intégrer les avantages des réseaux de neurones et des systèmes symboliques à base de règles. La fonction principale d'un tel système est de passer d'un mode de représentation des connaissances à l'autre. Les modèles transformationnels utilisent toutes les connaissances disponibles sur un problème : théoriques et empiriques. Le processus transformationnel permet donc la construction de systèmes hybrides qui peuvent opérer entre les deux niveaux neuronal/symbolique de représentation des connaissances.

Conclusions et perspectives

Dans ce mémoire, nous avons présenté nos recherches sur l'apport des réseaux connexionnistes dans les systèmes experts symboliques. Pour cela, nous avons étudié les deux approches afin de dégager leurs points forts et leurs limitations. Ces limitations font qu'un paradigme donné (symbolique ou connexionniste) est désormais incapable de résoudre à lui seul des problèmes complexes. Etant donné qu'il est difficile d'inventer de toutes pièces un nouveau paradigme plus satisfaisant, la "démarche du moindre effort" consiste à tirer parti des points forts des deux paradigmes et à réaliser des systèmes hybrides neuro-symboliques capables de prendre en charge des problèmes complexes. Dans de tels systèmes, on constate que :

- les points forts des systèmes symboliques compensent à peu près les points faibles des systèmes connexionnistes et vice versa,
- l'intégration des deux systèmes connexionniste et symbolique permet de diminuer, sinon de résoudre complètement, certaines limitations de l'un ou de l'autre système afin d'avoir des systèmes plus robustes.

Nous avons présenté et critiqué plusieurs architectures de systèmes hybrides neuro-symboliques proposées par des chercheurs du domaine. Ces architectures sont très variées, de par leurs composants et par les techniques de couplage utilisées, mais aussi en raison de différents sens donnés au mot 'hybride' dans le cadre général de l'intégration des points forts des systèmes symboliques et connexionnistes.

Nous avons proposé une nouvelle classification des systèmes hybrides neuro-symboliques, car nous croyons que notre schéma de classification a les avantages suivants :

- attache ensemble plusieurs fils (structure du système, modes de couplage et modes d'interaction entre modules) des autres schémas de classification existants dans une approche cohérente ;
- en ayant une approche commune à classifier ces systèmes hybrides, les développeurs de tels systèmes seront capables de décrire l'opération et les objectifs de leurs systèmes, tandis que d'autres pourront les évaluer avec une compréhension plus claire des issues des traitements ;
- prend en compte les développements récents dans le domaine de l'intégration neuro-symbolique.

L'analyse de l'état de l'art des systèmes hybrides neuro-symboliques nous a permit de déduire que tous les systèmes étudiés et développés jusqu'à présent sont hybrides et qu'un schéma complet de classification peut être établi en suivant trois types d'approches :

- l'approche unifiée, dont le but est de construire deux types d'architecture : des réseaux neuronaux manipulant des symboles ou des architectures biologiquement plausibles (se rapprocher de l'architecture du cerveau humain) ;
- l'approche modulaire, qui distingue clairement un module symbolique d'un module connexionniste. Ces modules sont couplés et interagissent pour résoudre un problème ;
- l'approche transformationnelle, regroupant les approches de type transformation (traduction) d'une structure symbolique en une architecture connexionniste et vice versa.

Nous avons ensuite étudié les trois types de modèles hybrides issus de notre schéma de classification, en donnant les motivations des chercheurs ainsi que les différentes classes pour chaque type de modèle hybride. Ces modèles sont :

- ✓ les modèles hybrides unifiés, qui sont constitués de deux sous classes : les modèles ascendants et les modèles descendants. Les modèles descendants sont de trois types, localiste, distribué ou combiné ;
- ✓ les modèles hybrides modulaires, qui peuvent être à couplages faible, moyen ou fort. Les modules formant de tels systèmes interagissent selon quatre modes, pré/post-traitement, sous-traitement, co-traitement ou méta-traitement ;
- ✓ les modèles hybrides transformationnels, qui sont répartis en deux sous classes : les modèles de compilation de règles dans un réseau de neurones et les modèles d'extraction de règles à partir d'un réseau de neurones. Ces modèles peuvent fonctionner entre les deux niveaux neuronal/symbolique de représentation des connaissances.

Nous avons ensuite appuyé nos recherches par l'étude de quelques exemples de modèles hybrides neuro-symboliques (unifiés, modulaires, et transformationnels), développés dans le cadre de l'intégration des points forts des réseaux connexionnistes et des systèmes symboliques. Ces exemples nous ont permit de voir de près les différentes

structures de systèmes hybrides neuro-symboliques, les types de couplage ainsi que les modes d'interaction entre leurs modules.

Comme perspectives nous souhaitons que ce travail de recherche soit suivi d'une application réelle qui permettra de comparer les différents modèles hybrides neuro-symboliques et peut être de répondre à un certain nombre de questions :

- le travail supplémentaire que requiert la conception et la réalisation d'un système hybride est-il réellement rentable ?

- la maintenance d'un système hybride est-elle plus difficile ? En général, l'augmentation de la complexité engendre l'augmentation du risque de pannes ;

- les systèmes hybrides permettent-ils de résoudre complètement certains des problèmes rencontrés par leurs composants, tels que la fragilité des systèmes symboliques ?

Introduction générale

De nos jours, l'informatique à pris une place importante dans notre société. Les systèmes informatiques deviennent de plus en plus complexes, comme les tâches qui leur sont confiées. Les systèmes d'Intelligence Artificielle (I.A), dont les systèmes experts est l'une des principales applications, ont été crées avec l'espoir de pouvoir aider l'homme dans les tâches les plus complexes, celles qui requièrent l'utilisation de l'intelligence ou la réalisation de procédés dits intelligents. Les résultats obtenus jusqu'à présent sont encore assez limités par rapport à notre conception d'un vrai système intelligent. A quelques rares exceptions près, l'homme reste toujours supérieur à la machine pour la réalisation de tâches complexes.

En résolution de problèmes, comme dans d'autres secteurs de l'informatique, et des sciences en général, réaliser des systèmes hybrides est une démarche très courante. Lorsque deux méthodes ou deux approches entrent quelque peu en compétition, il n'est pas rare que des chercheurs proposent de combiner les points forts, si possible, de chacune d'entre elles. Cela permet d'obtenir, à peu de frais, des systèmes hybrides de performances plus élevées pour un champ d'application plus large. Un autre aspect très important du développement des systèmes hybrides intelligents est leur capacité d'acquérir de nouvelles connaissances (parfois à partir de plusieurs sources différentes) et de les faire évoluer.

Dans plusieurs domaines (sciences cognitives, informatique), deux grandes approches sont plus ou moins en compétition :

- une approche symbolique, qui cherche à modéliser explicitement des processus de résolution, et emploie pour cela divers types de systèmes à base de connaissances, souvent appelés "systèmes symboliques" ;
- une approche connexionniste, qui met plutôt l'accent sur l'utilisation d'exemples de résolution, et qui utilise des réseaux d'automates simples (RNA : Réseaux de Neurones Artificiels), d'inspiration neurobiologique, souvent appelés "systèmes connexionnistes".

Chaque approche comporte des points forts, mais aussi des limitations. Puisque les deux approches sont en compétition du point de vue informatique, d'une part parce que dans certains cas elles visent à accomplir le même type de tâches, et d'autre part, parce que leurs domaines d'application sont souvent les mêmes : vision, robotique, traitement des langues, diagnostic, etc. des chercheurs ont proposé de les combiner pour réaliser

des Systèmes Hybrides Neuro-Symboliques (SHNS) puissants, capables d'accomplir des tâches complexes et de mieux modéliser des processus cognitifs. 2

Dans cette thèse, nous avons développé des recherches sur l'apport des réseaux de neurones artificiels dans les systèmes experts symboliques. Nous avons présenté une vision structurée d'un domaine de recherche récent, appelé intégration neuro-symbolique (INS). Nous avons aussi proposé une nouvelle classification des systèmes hybrides neuro-symboliques appuyée par des études de quelques systèmes hybrides représentatifs du domaine, développés dans le cadre de l'intégration des points forts des réseaux connexionnistes et des systèmes symboliques.

Le mémoire est organisé de la manière suivante :

- Le chapitre (1) est consacré à une étude des réseaux de neurones artificiels. Nous présentons les principaux modèles, qui sont répartis en deux grandes classes : les réseaux unidirectionnels et les réseaux récurrents. Nous donnons une synthèse des méthodes d'apprentissage de ces réseaux. Nous présentons les différentes tâches que peuvent accomplir les réseaux de neurones artificiels. Enfin, une analyse des points forts et points faibles des réseaux de neurones artificiels apparaît explicitement dans ce chapitre.

- Le chapitre (2) est réservé à une étude des systèmes experts symboliques. Nous donnons la structure générale d'un système expert symbolique. Nous décrivons les méthodes de représentation des connaissances dans un système expert symbolique ainsi que le processus d'acquisition des connaissances, qui est un processus de rassemblement des connaissances empiriques et théoriques nécessaires à la résolution d'un problème lié à une application spécifique. Enfin, nous présentons une analyse des points forts et points faibles des systèmes experts symboliques.

- Le chapitre (3) présente les motivations des chercheurs pour une intégration des points forts de chacune des deux approches symbolique et connexionniste dans des systèmes hybrides. Nous présentons et nous critiquons plusieurs classifications de tels systèmes proposées par des chercheurs du domaine. Puis, nous présentons une nouvelle classification plus complète et cohérente qui servira de base pour les concepteurs de systèmes hybrides neuro-symboliques. L'analyse de l'état de l'art nous a mené à croire qu'un nouveau schéma complet de classification peut être établi en suivant trois types d'approches : approche unifiée, approche modulaire et approche transformationnelle. Ces dernières sont développées dans les chapitres 4, 5 et 6 du présent mémoire.

- Le chapitre (4) est réservé à l'étude des modèles hybrides unifiés que nous avons classé sous l'approche unifiée. Nous présentons les motivations des chercheurs3 pour ce type de modèles. Nous donnons une classification de ces modèles, nous distinguons deux types de modèles : les modèles ascendants et les modèles descendants. Puis, nous étudions quelques exemples de modèles hybrides du domaine développés sous l'approche unifiée.

- Le chapitre (5) traite les modèles hybrides modulaires que nous avons classé sous l'approche modulaire. Nous présentons les motivations des chercheurs pour ce type de modèles. Nous présentons une classification de ces modèles, nous distinguons trois sous classes : les modèles faiblement couplés, les modèles étroitement couplés et les modèles fortement couplés. Enfin, nous étudions quelques exemples de modèles hybrides représentatifs du domaine, développés sous l'approche modulaire.

- Le chapitre (6) concerne les modèles hybrides transformationnels que nous avons classé sous l'approche transformationnelle. Nous présentons les motivations des chercheurs pour ce type de modèles. Nous donnons une classification de ces modèles, nous distinguons deux sous-classes : la compilation de règles dans un réseau de neurones et l'extraction de règles à partir d'un réseau de neurones. Puis, nous étudions deux exemples de modèles hybrides transformationnels développés dans le cadre du transfert de connaissances.

N° d'ordre : 03/2002-M/IN

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENNE (USTHB) – ALGER

Faculté de Génie Electronique et Informatique

THESE

Présentée à l'USTHB pour obtenir le diplôme de

MAGISTER

Spécialité : Informatique

Par

Mohamed AISSANI

Apport des réseaux connexionnistes dans les systèmes experts. Une nouvelle classification des systèmes hybrides neuro-symboliques.

Soutenue le lundi 08 avril 2002 devant le jury :

N. BADACHE	**Maître de conférences, USTHB**	**Président**
H. AZZOUNE	**Maître de conférences, USTHB**	**Directeur de thèse**
M. AHMED NACER	**Maître de conférences, USTHB**	**Examinateur**
M. BOUKALA	**Maître de conférences, USTHB**	**Examinatrice**
H. KHELALFA	**PhD, CERIST**	**Examinateur**

Avant-propos

Ce travail de thèse a été réalisé au niveau du Département Informatique de l'Université des Sciences et de la Technologie Houari Boumedienne (USTHB) d'Alger et au Laboratoire d'Intelligence Artificielle, Unité d'Enseignement et de Recherche en Informatique (UERI) de l'Ecole Militaire Polytechnique (EMP), Bordj el Bahri.

Je tiens à exprimer mes vifs remerciements à Monsieur H. AZZOUNE, Maître de conférences à l'USTHB, qui a dirigé cette thèse, pour sa disponibilité et les conseils qu'il n'a cessé de me prodiguer tout au long de ce travail. Je lui serais toujours reconnaissant pour toute sa confiance, le regard critique qu'il a apporté à ce travail ; me permettant ainsi d'aborder un domaine de recherche d'actualité.

J'exprime mes vifs remerciements à Monsieur N. BADACHE, Maître de conférences à l'USTHB pour l'honneur qu'il me fait en présidant ce Jury. Mes sincères remerciements vont également à Madame M. BOUKALA et Monsieur M. AHMED NACER, Maîtres de conférences à l'USTHB, et à Monsieur H. KHELALFA, PhD au CERIST, pour l'honneur qu'ils me font en acceptant d'examiner ce travail.

Je voudrais remercier Monsieur O. SERIR, Chef de l'UERI/EMP et ses Collaborateurs, pour leur disponibilité.

Mes sincères remerciements vont également à Monsieur H. DEMMOU, Maître de conférences et au Professeur A. SAHRAOUI, pour leur accueil chaleureux et leur aide durant mon séjour au Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS, France).

Toute mon amitié va à l'ensemble du Personnel du Département Informatique de l'USTHB, pour leur amabilité.

Je tiens à remercier ma famille et mes amis pour leurs encouragements.

Table des matières