

## LES RÉSEAUX DE NEURONES

G. DREYFUS

École Supérieure de Physique et de Chimie Industrielles de la Ville de Paris (ESPCI),

Laboratoire d'Électronique

10, rue Vauquelin

75005 PARIS

### Résumé

Les réseaux de neurones formels sont devenus en quelques années des outils précieux dans des domaines très divers de l'industrie et des services. Néanmoins, ils n'ont pas encore atteint leur plein développement, pour des raisons plus psychologiques que techniques, liées aux connotations biologiques du terme, et au fait qu'ils sont considérés, à tort, comme des outils d'Intelligence Artificielle. Or l'intérêt des réseaux de neurones, dans le domaine des Sciences de l'Ingénieur, ne doit rien à la métaphore biologique : il est uniquement dû aux propriétés mathématiques spécifiques de ces réseaux. Nous expliquons ici, à partir de principes simples, ce que sont réellement les réseaux de neurones, et nous délimitons leurs domaines d'excellence. Ces principes sont illustrés par diverses applications industrielles.

### 1. INTRODUCTION

Lorsqu'apparaît une nouvelle technique, l'ingénieur se demande naturellement en quoi cette nouveauté peut lui être utile. Si elle est dotée d'un nom plus métaphorique que technique - ce qui est évidemment le cas pour les réseaux de neurones - la réponse à cette question doit être particulièrement précise et motivée. De plus, la mise en œuvre des réseaux de neurones est très simple ; la tentation peut être grande, d'appliquer cette technique de manière irréfléchie ou inadaptée, ce qui ne peut conduire qu'à des déceptions. C'est pourquoi nous expliquerons ici les principes fondamentaux qui justifient l'intérêt pratique des réseaux de neurones, et nous situerons ces derniers dans la perspective des méthodes classiques de traitement statistique de données ; nous montrerons que la technique des réseaux de neurones formels doit être considérée comme une extension puissante de techniques bien connues des ingénieurs, telles que la régression, et nous illustrerons notre propos par quelques exemples industriels choisis dans les domaines de la modélisation, de la commande, et de la classification automatique.

### 2. LES NEURONES FORMELS

*Un "neurone formel" (ou simplement "neurone") est une fonction algébrique non linéaire et bornée, dont la valeur dépend de paramètres appelés coefficients ou poids. Les variables de*

cette fonction sont habituellement appelées "entrées" du neurone, et la valeur de la fonction est appelée sa "sortie".

Un neurone est donc avant tout un opérateur mathématique, dont on peut calculer la valeur numérique par quelques lignes de logiciel. On a pris l'habitude de représenter graphiquement un neurone comme indiqué sur la Figure 1.

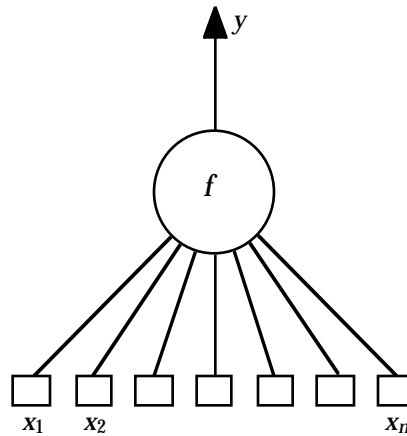


Figure 1

Un neurone réalise une fonction non linéaire bornée

$$y = f(x_1, x_2, \dots, x_n; c_1, c_2, \dots, c_p)$$

où les  $\{x_i\}$  sont les variables et les  $\{c_j\}$  sont des paramètres.

Pour des raisons que nous expliquerons plus loin, les neurones les plus fréquemment utilisés sont ceux pour lesquels la fonction  $f$  est une fonction non linéaire (généralement une tangente hyperbolique) d'une combinaison linéaire des entrées :

$$y = \tanh \left[ \sum_{i=1}^n c_i x_i \right] .$$

Les  $\{x_i\}$  sont les variables (ou entrées) du neurone, les  $\{c_i\}$  sont des paramètres ajustables.

Un neurone formel ne réalise donc rien d'autre qu'une somme pondérée suivie d'une non-linéarité. C'est l'association de tels éléments simples sous la forme de *réseaux* qui permet de réaliser des fonctions utiles pour des applications industrielles.

### 3 . LES RESEAUX DE NEURONES FORMELS

On distingue deux grands types d'architectures de réseaux de neurones : les réseaux de neurones *non bouclés* et les réseaux de neurones *bouclés*.

#### 3 . 1 . Les réseaux de neurones non bouclés [1]

*Un réseau de neurones non bouclé réalise une (ou plusieurs) fonctions algébriques de ses entrées, par composition des fonctions réalisées par chacun de ses neurones.*

Un réseau de neurones non bouclé est représenté graphiquement par un ensemble de neurones "connectés" entre eux, l'information circulant des entrées vers les sorties sans "retour en arrière" ; si l'on représente le réseau comme un graphe dont les nœuds sont les neurones et les arêtes les "connexions" entre ceux-ci, le graphe d'un réseau non bouclé est *acyclique*. Le terme de "connexions" est une métaphore : dans la très grande majorité des applications, les réseaux de neurones sont des formules algébriques dont les valeurs numériques sont calculées par des programmes d'ordinateurs, non des objets physiques (circuits électroniques spécialisés) ; néanmoins, le terme de connexion, issu des origines biologiques des réseaux de neurones, est passé dans l'usage, car il est commode quoique trompeur. Il a même donné naissance au terme de *connexionnisme*.

La Figure 2 représente un réseau de neurones non bouclé qui a une structure particulière, très fréquemment utilisée : il comprend des entrées, une couche de neurones "cachés" et des neurones de sortie. Les neurones de la couche cachée ne sont pas connectés entre eux. Cette structure est appelée *Perceptron multicouche*.

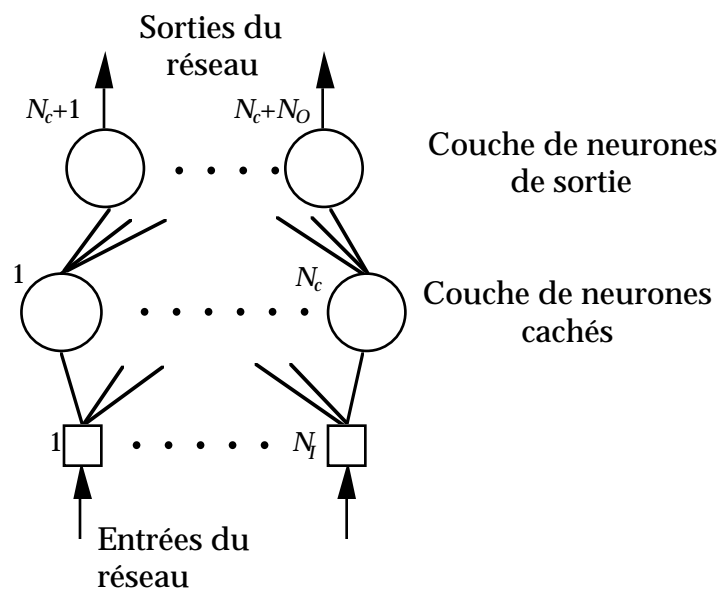


Figure 2

Un Perceptron multicouche.

Les réseaux de neurones non bouclés sont des objets *statiques* : si les entrées sont indépendantes du temps, les sorties le sont également. Ils sont utilisés principalement pour effectuer des tâches d'*approximation de fonction non linéaire*, de *classification* ou de *modélisation de processus statiques non linéaires*.

### 3.2. Les réseaux de neurones bouclés

Contrairement aux réseaux de neurones non bouclés dont le graphe de connexions est acyclique, les réseaux de neurones bouclés peuvent avoir une topologie de connexions quelconque, comprenant notamment des boucles qui ramènent aux entrées la valeur d'une ou plusieurs sorties. Pour qu'un tel système soit causal, il faut évidemment qu'à toute boucle soit associé un *retard* : un réseau de neurones bouclé est donc un système *dynamique*, régi par des équations différentielles ; comme l'immense majorité des applications sont réalisées par des programmes d'ordinateurs, on se place dans le cadre des systèmes à temps discret, où les équations différentielles sont remplacées par des équations aux différences.

*Un réseau de neurones bouclé à temps discret est donc régi par une (ou plusieurs) équations aux différences non linéaires, résultant de la composition des fonctions réalisées par chacun des neurones et des retards associés à chacune des connexions.*

La forme la plus générale des équations régissant un réseau de neurones bouclé est appelée *forme canonique* ;

$$x(k+1) = \varphi [x(k), u(k)]$$

$$y(k) = \psi [x(k), u(k)]$$

où  $\varphi$  et  $\psi$  sont des fonctions non linéaires réalisées par un réseau de neurones non bouclé (par exemple, mais pas obligatoirement, un Perceptron multicouche), et où  $k$  désigne le temps (discret). La forme canonique est représentée sur la Figure 3. Tout réseau de neurones, aussi compliqué soit-il, peut être mis sous cette forme canonique, de manière complètement automatique [2]. Ainsi, le réseau représenté sur la Figure 4 peut être mis sous la forme canonique, strictement équivalente mais beaucoup plus facilement manipulable, représentée sur la Figure 5.

Les réseaux de neurones bouclés sont utilisés pour effectuer des tâches de *modélisation de systèmes dynamiques*, de *commande de processus*, ou de *filtrage*.

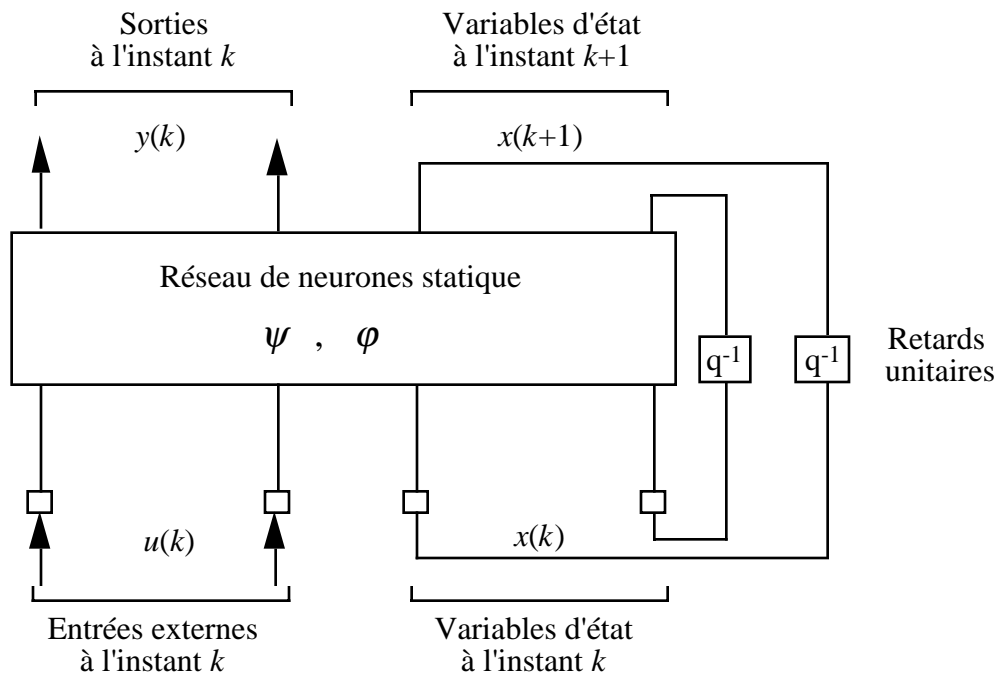


Figure 3

Forme canonique d'un réseau de neurones bouclé

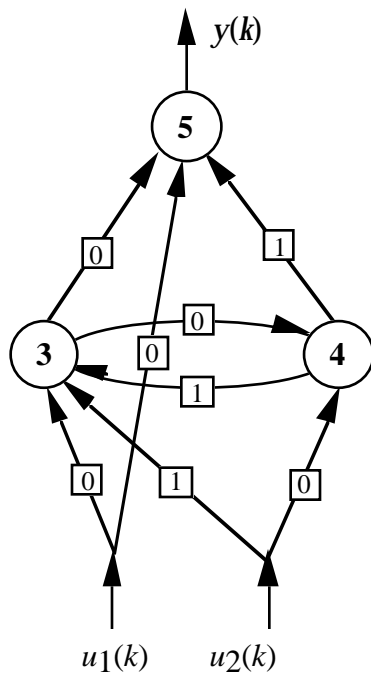


Figure 4

Un réseau de neurones bouclé. Les nombres dans les carrés sont les retards (exprimés en nombre de périodes d'échantillonnage) associés à chaque connexion.

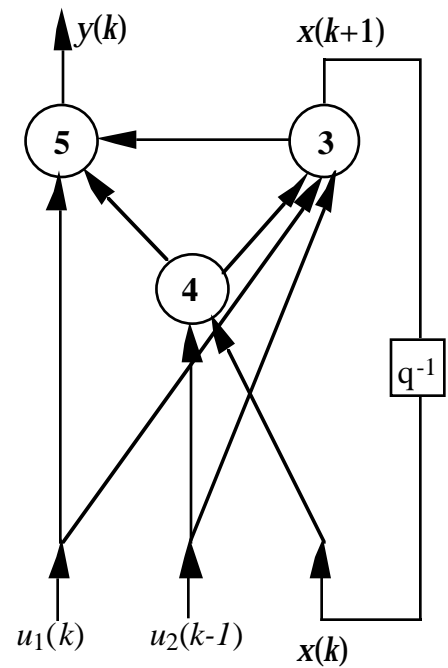


Figure 5

La forme canonique du réseau de la Figure 4.

#### 4. PROPRIÉTÉ FONDAMENTALE DES RÉSEAUX DE NEURONES FORMELS : L'APPROXIMATION PARCIMONIEUSE

Les réseaux de neurones formels, tels que nous les avons définis dans le paragraphe précédent, possèdent une propriété remarquable qui est à l'origine de leur intérêt pratique dans des domaines très divers : ce sont *des approximateurs universels parcimonieux*.

Sans entrer dans les détails mathématiques, la propriété d'approximation peut être énoncée de la manière suivante : *toute fonction bornée suffisamment régulière peut être approchée avec une précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation, et un neurone de sortie linéaire* [3]. Cette propriété n'est pas spécifique aux réseaux de neurones : il existe bien d'autres familles de fonctions paramétrées possédant cette propriété ; c'est le cas notamment des ondelettes, des fonctions radiales, des fonctions splines, par exemple.

La spécificité des réseaux de neurones réside dans le caractère *parcimonieux* de l'approximation : à précision égale, les réseaux de neurones nécessitent moins de paramètres ajustables (les poids des connexions) que les approximateurs universels couramment utilisés ; plus précisément, le nombre de poids varie *linéairement* avec le nombre de variables de la fonction à approcher, alors qu'il varie *exponentiellement* pour la plupart des autres approximateurs [4]. Nous verrons au paragraphe 5 que c'est cette remarquable parcimonie qui justifie l'intérêt industriel des réseaux de neurones. En pratique, dès qu'un problème fait intervenir plus de deux variables, les réseaux de neurones sont, en général, préférables aux autres méthodes.

Qualitativement, la propriété de parcimonie peut se comprendre de la manière suivante : lorsque l'approximation est une combinaison *linéaire* de fonctions élémentaires fixées (des monômes par exemple, ou des gaussiennes à centres et écarts-types fixes), on ne peut ajuster que les coefficients de la combinaison ; en revanche, lorsque l'approximation est une combinaison linéaire de fonctions non linéaires à *paramètres ajustables* (un Perceptron multicouche par exemple), on ajuste à la fois les coefficients de la combinaison *et la forme des fonctions que l'on combine*. Ainsi, dans un Perceptron multicouche, les poids de la première couche déterminent la forme de chacune des sigmoïdes réalisées par les neurones cachés, et les poids de la seconde couche déterminent une combinaison linéaire de ces fonctions. On conçoit facilement que cette souplesse supplémentaire, conférée par le fait que l'on ajuste la forme des fonctions que l'on superpose, permet d'utiliser un plus petit nombre de fonctions élémentaires, donc un plus petit nombre de paramètres ajustables. Nous allons voir dans le paragraphe suivant pourquoi cette propriété de *parcimonie* est précieuse dans les applications industrielles.

## 5 . RESEAUX DE NEURONES ET REGRESSION NON LINEAIRE

Dans la pratique, on n'utilise pas les réseaux de neurones pour réaliser des approximations de fonctions *connues*. Le plus souvent, le problème qui se pose à l'ingénieur est le suivant : il dispose d'un ensemble de mesures de variables d'un processus de nature quelconque (physique, chimique, économique, financier, ...), et du résultat de ce processus ; il suppose qu'il existe une relation déterministe entre ces variables et ce résultat, et il cherche une forme mathématique de cette relation, valable dans le domaine où les mesures ont été effectuées, sachant que (1) les mesures sont en nombre fini, que (2) elles sont certainement entachées de bruit, et que (3) toutes les variables qui déterminent le résultat du processus ne sont pas forcément mesurées. En d'autres termes, l'ingénieur cherche un *modèle* du processus qu'il étudie, à partir des mesures dont il dispose, et d'elles seules : on dit qu'il effectue une modélisation "boîte noire". Dans le jargon des réseaux de neurones, les données à partir desquelles on cherche à construire le modèle s'appellent des *exemples*.

En quoi la propriété d'approximation parcimonieuse peut-elle être utile pour résoudre ce genre de problèmes ? Ce que l'ingénieur cherche à obtenir à l'aide de son modèle, c'est la "vraie" fonction qui relie la grandeur  $y_p$  que l'on veut modéliser aux variables  $\{x\}$  qui la déterminent, c'est-à-dire la fonction que l'on obtiendrait en faisant une infinité de mesures de  $y_p$  pour chaque valeur possible de  $\{x\}$  : en termes de statistiques, l'ingénieur cherche la *fonction de régression* de la grandeur à modéliser. Cette fonction est inconnue, mais on peut en chercher une approximation à partir des mesures disponibles : les réseaux de neurones sont donc de bons candidats pour cela, si la fonction de régression cherchée est non linéaire. Cette approximation est obtenue en estimant les paramètres d'un réseau de neurones au cours d'une phase dite *d'apprentissage*. C'est ici que la propriété d'approximation parcimonieuse des réseaux de neurones est précieuse : en effet, le nombre de mesures nécessaires pour estimer les paramètres de manière significative est d'autant plus grand que le nombre de paramètres est grand. Ainsi, pour modéliser une grandeur avec une précision donnée à l'aide d'un réseau de neurones, il faut *moins de données* que pour la modéliser, avec une précision comparable, à l'aide d'une régression linéaire multiple ; de manière équivalente, un réseau de neurones permet, avec les mêmes données disponibles, de réaliser une approximation plus précise qu'une régression linéaire multiple.

De manière générale, un réseau de neurones *permet donc de faire un meilleur usage des mesures disponibles que les méthodes de régression non linéaires conventionnelles*. Ce gain peut être considérable lorsque le processus à modéliser dépend de plusieurs variables : rappelons en effet que le nombre de paramètres (donc de mesures) varie *exponentiellement* pour les méthodes conventionnelles de régression non linéaire, alors qu'elle varie *linéairement* pour les réseaux de neurones.

Ainsi, à la lumière de cette propriété fondamentale, la technique des réseaux de neurones apparaît comme une puissante *méthode de régression non linéaire* : ce n'est donc rien d'autre qu'une extension des méthodes de régression linéaire ou multilinéaires proposées par tous les logiciels qui permettent de faire de la modélisation de données. Contrairement à une croyance répandue, elle ne relève donc pas de l'Intelligence Artificielle au sens classique du terme, mais elle constitue une branche des statistiques appliquées. Il ne faut donc pas être victime du vocabulaire anthropomorphique utilisé (neurones, apprentissage, etc.) ; le tableau ci-dessous résume les équivalences entre le vocabulaire des statistiques et celui des réseaux de neurones.

RÉSEAUX DE NEURONES	STATISTIQUES
Choix de l'architecture	Choix de la famille de fonctions destinées à approcher la fonction de régression
Ensemble d'apprentissage	Observations
Apprentissage	Estimation des paramètres de l'approximation de la fonction de régression
Généralisation	Interpolation, extrapolation

Tableau 1  
Réseaux de neurones et statistiques

## 6. MISE EN ŒUVRE DES RESEAUX DE NEURONES ET DOMAINE D'APPLI-CATIONS

Pour réaliser l'approximation de la fonction de régression cherchée, à partir d'échantillons généralement bruités, à l'aide d'un réseau de neurones, trois étapes successives sont nécessaires :

- il faut tout d'abord choisir l'architecture du réseau, c'est-à-dire les entrées externes, le nombre de neurones cachés, et l'agencement des neurones entre eux, de telle manière que le réseau soit en mesure de reproduire ce qui est déterministe dans les données ; le nombre de poids ajustables est un des facteurs fondamentaux de la réussite d'une application : si le réseau possède un trop grand nombre de poids, c'est-à-dire si le réseau est trop "souple", il risque de s'ajuster au bruit qui est présent dans les données de l'ensemble d'apprentissage, et, même en l'absence de bruit, il risque de présenter des oscillations non significatives entre les points d'apprentissage, donc de posséder de mauvaises propriétés d'interpolation (ou, dans le jargon des réseaux de neurones, de "généralisation") ; si ce nombre est trop petit, le réseau est trop "rigide" et ne peut reproduire la partie déterministe de la fonction. Le problème de la détermination de l'architecture optimale est resté pendant longtemps un problème ouvert, mais il existe actuellement diverses méthodes, mettant notamment en jeu des tests



statistiques, qui permettent de déterminer cette architecture pour une vaste classe de réseaux [5] ; la Figure 6 illustre l'influence du nombre de paramètres sur la qualité de l'approximation ;

- il faut calculer les poids du réseau - ou, en d'autres termes, estimer les paramètres de la régression non linéaire - à partir des exemples, en minimisant l'erreur d'approximation sur les points de l'ensemble d'apprentissage, de telle manière que le réseau réalise la tâche désirée. Ce calcul des coefficients synaptiques constitue *l'apprentissage supervisé* pour le réseau de neurones ;
- il faut enfin estimer la qualité du réseau obtenu en lui présentant des exemples qui ne font pas partie de l'ensemble d'apprentissage.

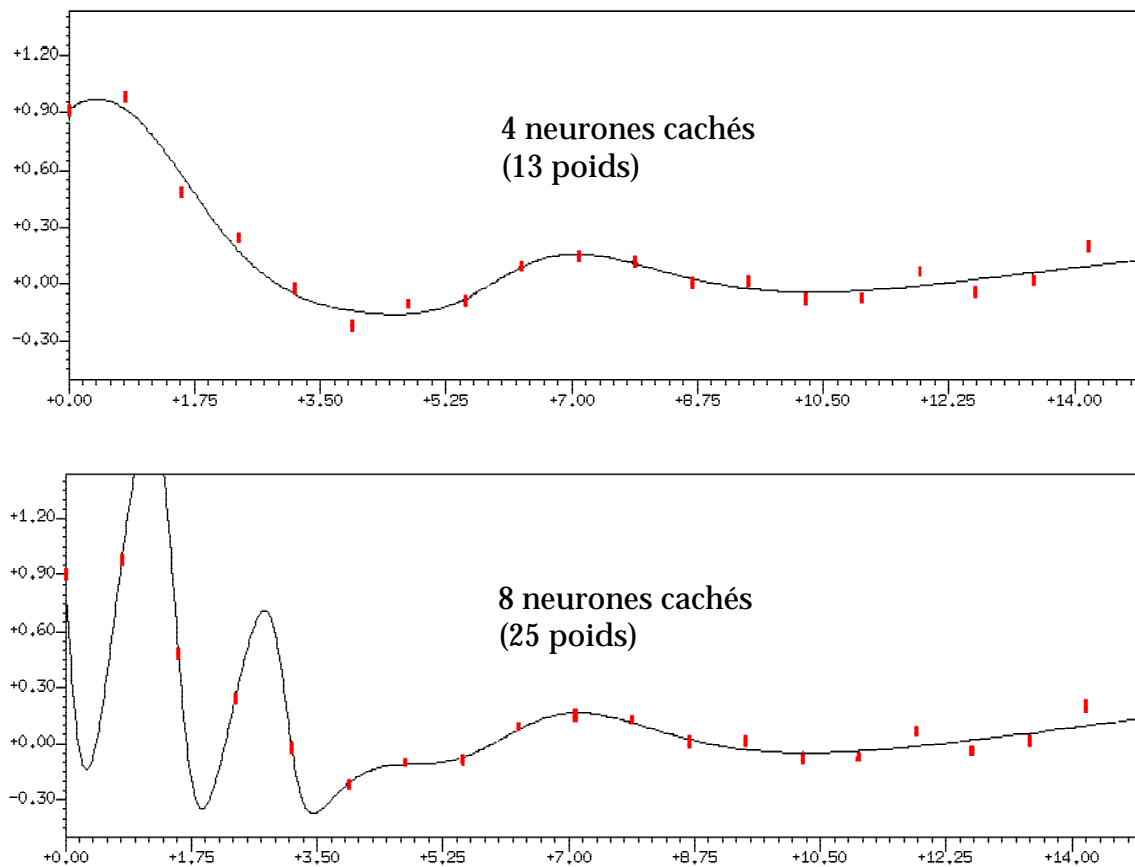


Figure 6

Le "surajustement" : le réseau le plus parcimonieux (4 neurones cachés, soit 13 coefficients) produit un bien meilleur ajustement qu'un réseau trop riche en coefficients (8 neurones cachés, soit 25 coefficients).

Les grands domaines d'application des réseaux de neurones découlent naturellement de leur propriété fondamentale :

- **la régression non linéaire, ou modélisation de données *statiques*** : il existe une immense variété de phénomènes statiques qui peuvent être caractérisés par une relation

déterministe entre des causes et des effets ; les réseaux de neurones sont de bons candidats pour modéliser de telles relations à partir d'observations expérimentales, sous réserve que celles-ci soient suffisamment nombreuses et représentatives ;

- **la modélisation de processus *dynamiques non linéaires*** : modéliser un processus, c'est trouver un ensemble d'équations mathématiques qui décrivent le comportement dynamique du processus, c'est-à-dire l'évolution de ses sorties en fonction de celle de ses entrées ; c'est donc typiquement un problème qui peut être avantageusement résolu par un réseau de neurones, si le phénomène que l'on désire modéliser est non-linéaire. La prédiction de séries chronologiques (prédictions financières, prédiction de consommation, etc.) entre dans ce cadre.
- **la commande de processus** : commander un processus, c'est imposer à celui-ci un comportement défini à l'avance en fonction des signaux de commande ; l'ensemble commande + processus peut donc être considéré comme un système qui réalise une fonction (non-linéaire) qu'un réseau de neurones peut approcher.
- **la classification** : supposons que l'on désire classer des formes en deux catégories, A ou B, en fonction de certaines caractéristiques de ces formes ; on peut définir une fonction  $\varphi$  qui vaut +1 pour toutes les formes de la classe A et -1 pour toutes les formes de la classe B. Les réseaux de neurones sont de bons candidats pour réaliser une approximation de cette fonction  $\varphi$ , et l'on peut démontrer que cette approximation constitue une estimation de la probabilité d'appartenance de la forme inconnue à la classe A. Les réseaux de neurones fournissent donc une information très riche, qui est loin d'être une simple réponse binaire. Cette propriété remarquable (que les réseaux de neurones partagent avec d'autres classificateurs) n'est malheureusement pas mise à profit dans la plupart des applications.

## 7. L'APPRENTISSAGE DES RESEAUX DE NEURONES FORMELS

Comme nous l'avons indiqué plus haut, l'apprentissage "supervisé", pour les réseaux de neurones formels, consiste à calculer les coefficients de telle manière que les sorties du réseau de neurones soient, pour les exemples utilisés lors de l'apprentissage, aussi proches que possibles des sorties "désirées", qui peuvent être la classe d'appartenance de la forme que l'on veut classer, la valeur de la fonction que l'on veut approcher ou de la sortie du processus que l'on veut modéliser, ou encore la sortie souhaitée du processus à commander. La plupart des algorithmes d'apprentissage des réseaux de neurones formels sont des algorithmes d'optimisation : ils cherchent à minimiser, par des méthodes d'optimisation non linéaire, une fonction de coût qui constitue une mesure de l'écart entre les réponses réelles du réseau et ses réponses désirées. Cette optimisation se fait de manière itérative, en modifiant les poids en fonction du gradient de la fonction de coût : le gradient est estimé par une méthode spécifique aux réseaux de neurones, dite méthode de rétropropagation, puis il est utilisé par l'algorithme d'optimisation pro-

prement dit. Les poids sont initialisés aléatoirement avant l'apprentissage, puis modifiés itérativement jusqu'à obtention d'un compromis satisfaisant entre la précision de l'approximation sur l'ensemble d'apprentissage et la précision de l'approximation sur un ensemble de validation disjoint du précédent. Contrairement à des affirmations maintes fois répétées, l'apprentissage des réseaux de neurones n'est pas spécialement lent : il existe des algorithmes d'optimisation non linéaire extrêmement rapides [6] qui permettent de faire des développements industriels sur de simples PC. L'apprentissage des réseaux de neurones bouclés (pour réaliser des modèles dynamiques) est très semblable à celui des réseaux non bouclés [7].

## 8. MODELISATION DE PHENOMENES NON LINEAIRES STATIQUES

On peut trouver de nombreux exemples d'utilisation, potentielle ou opérationnelle, des réseaux de neurones pour la modélisation de phénomènes statiques. Ainsi, dans le domaine des *relations structure-activité* (ou *QSAR*), les réseaux de neurones permettent de prévoir l'activité d'une molécule (caractérisée par un nombre réel) à partir de descripteurs chimiques de cette molécule (qui sont eux-mêmes des nombres réels) ; par exemple, il est possible de prédire la solubilité dans l'eau, le point d'ébullition, le coefficient de partage eau-octanol, ou toute autre propriété caractérisée par un nombre, en fonction de descripteurs tels que la masse moléculaire, le moment dipolaire, la charge portée par les divers atomes, le "volume" de la molécule, etc. (Figure 7). Certains de ces descripteurs sont mesurables, d'autres sont calculés par des méthodes semi-empiriques ou *ab initio*. On peut imaginer de nombreuses extensions de cette approche : prédiction de propriétés pharmacologiques de molécules, formulation de mélanges, prédiction de propriétés mécaniques ou optiques de matériaux, etc. Par exemple, un système de prédiction des propriétés mécaniques des caoutchoucs pour pneumatiques a été développé pour la société Michelin. Dans ces domaines, la combinaison des connaissances physiques ou chimiques de l'expert et la maîtrise des techniques statistiques "neuronales" permet d'obtenir des résultats très supérieurs à ceux qui étaient obtenus auparavant en utilisant les mêmes bases de données [8].

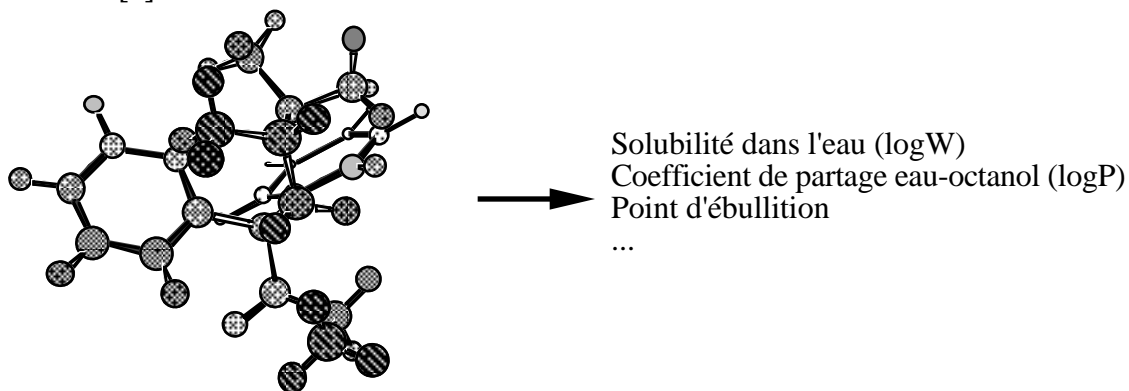


Figure 7

## 9. MODELISATION DE PROCESSUS NON LINEAIRES DYNAMIQUES

Comme nous l'avons vu plus haut, la propriété d'approximation universelle parcimonieuse des réseaux de neurones peut avantageusement être mise à profit pour la modélisation dynamique non linéaire de processus très divers.

On distingue habituellement

- les *modèles de connaissance*, dont l'expression mathématique, comprenant un petit nombre de paramètres ajustables, résulte d'une analyse (physique, chimique, économique, etc.) du processus,
- les *modèles "boîte noire"*, qui sont établis uniquement à partir des mesures effectuées sur le processus, sans intervention d'autre connaissance.

Les réseaux de neurones sont souvent utilisés comme des modèles "boîtes noires". Néanmoins, nous verrons plus loin qu'ils peuvent aussi mettre en œuvre des connaissances, constituant ainsi un excellent compromis entre les modèles "boîtes noires" et les modèles de connaissances.

### 9.1. Les modèles dynamiques "boîtes noires"

Pour réaliser un modèle dynamique, il convient alors d'utiliser des réseaux bouclés, qui, comme indiqué plus haut, sont eux-mêmes des systèmes dynamiques. Comme précédemment, l'apprentissage est l'estimation des paramètres du modèle neuronal utilisé, selon le schéma représenté sur la Figure 8.

L'objectif de l'apprentissage n'est *pas* ici d'annuler l'erreur de prédiction, puisque, si tel était le cas, le réseau serait capable de reproduire l'effet des perturbations non mesurables ! Il s'agit plutôt d'obtenir une erreur de prédiction dont la variance est minimale, c'est-à-dire égale à celle du bruit. Si l'on peut obtenir un tel résultat, le réseau reproduit complètement le comportement déterministe du processus, bien que l'apprentissage ait été effectué en présence de perturbations. Des résultats théoriques prouvent que cet objectif est accessible, et de nombreux exemples montrent qu'il est effectivement atteint [9].

### 9.2. Les modèles neuronaux de connaissances

Une des caractéristiques particulièrement avantageuses - et généralement ignorée - des réseaux de neurones est la possibilité d'introduire, dans la conception même du réseau, des connaissances mathématiques résultant de l'analyse de la physico-chimie du processus. De telles connais-

sances sont très souvent disponibles, et il serait déraisonnable de ne pas en tirer profit. Ainsi, on bénéficie à la fois de l'intelligibilité des modèles de connaissance et des capacités d'adaptation des réseaux de neurones : loin de constituer des "boîtes noires", ces derniers sont alors de véritables *modèles neuronaux de connaissances* [9], [10]. Citons, à titre d'exemple, un modèle de colonne à distiller réalisé par NETRAL S.A. pour Elf-Atochem, qui permet de détecter précocement des anomalies de fonctionnement, par examen de la différence entre les prédictions du modèle (qui fonctionne en temps réel sur PC) et les grandeurs effectivement mesurées sur la colonne [10], [11].

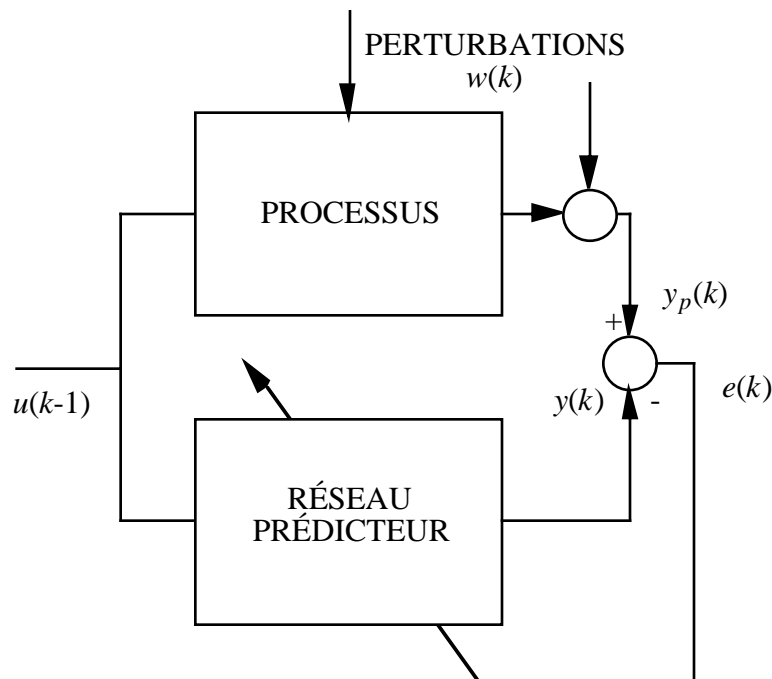


Figure 8

Identification des paramètres d'un modèle : le réseau de neurones doit, à partir des commandes à l'instant  $k-1$ , prédire la sortie qu'aurait le processus à l'instant  $k$  si les perturbations  $w(k)$ , non mesurables, n'existaient pas.

## 10 . COMMANDE DE PROCESSUS

L'utilisation des réseaux de neurones pour la commande (adaptative ou non) de processus non linéaires découle naturellement des aptitudes de ces derniers à la modélisation. Il s'agit essentiellement d'une extension non linéaire de la commande optimale avec coût quadratique sur un horizon infini [12]. Considérons en effet une structure de commande typique (la commande avec modèle interne) représentée sur la Figure 9.

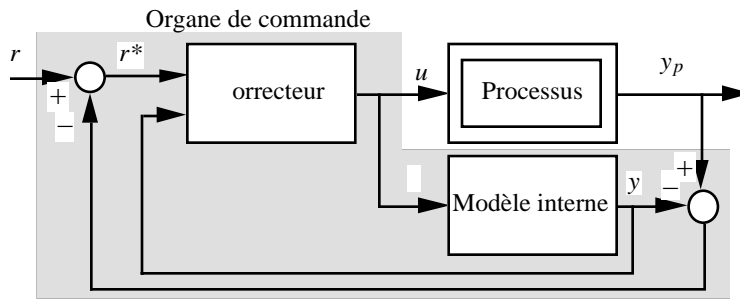


Figure 9

Exemple d'architecture pour la commande neuronale robuste de processus non linéaires : la commande avec modèle interne.

Elle comprend :

- un modèle neuronal, obtenu comme indiqué au paragraphe précédent
- un correcteur neuronal dont les coefficients sont mis à jour périodiquement si la commande est adaptative ; dans le cas contraire, ses coefficients sont fixes une fois l'apprentissage terminé.

Pour l'apprentissage de systèmes de poursuite, il est nécessaire, de surcroît, d'utiliser un modèle de référence qui traduit le cahier des charges en termes de dynamique de poursuite désirée.

La commande de processus non linéaires semble être l'un des domaines les plus prometteurs pour les réseaux de neurones à l'heure actuelle. Ainsi, la société SAGEM, en collaboration avec l'ESPCI, a réalisé la commande d'un véhicule 4x4 Mercedes : le volant, l'accélérateur et le frein sont commandés par des réseaux de neurones, de telle manière que le véhicule reste sur une trajectoire déterminée, avec un profil de vitesse déterminé, *en tout-terrain*, indépendamment des perturbations possibles (glissement, terrain en dévers, vent, ...) ; le véhicule est équipé pour cela de capteurs proprioceptifs (permettant de connaître l'angle du volant, la pression dans le circuit de freinage, l'angle du papillon d'admission, la vitesse de rotation des roues) et extéroceptifs (permettant de connaître la position du véhicule dans son environnement) [13]. Les comparaisons entre commandes "neuronales" (faisant intervenir des connaissances a priori sur le processus) et commandes non linéaires traditionnelles ont montré que les réseaux de neurones permettent d'obtenir des résultats au moins aussi bons, et souvent meilleurs, mais surtout qu'ils sont de mise en œuvre beaucoup plus simple en raison du caractère *générique* des algorithmes mis en œuvre : quelle que soit l'architecture du réseau bouclé utilisé, c'est toujours le même algorithme qui est mis en œuvre.

## 11. CLASSIFICATION

Lorsque l'on cherche à résoudre un problème de classification, on a affaire à des formes (par exemples des chiffres manuscrits, des phonèmes, des pièces mécaniques, etc.) susceptibles d'appartenir à des catégories, ou classes, différentes. Un classifieur est capable d'attribuer une

classe à une forme inconnue qui lui est présentée, ou - caractéristique très importante - de refuser de lui attribuer une classe si la forme inconnue est trop éloignée des formes utilisées pour l'apprentissage. Les performances d'un classifieur dépendent essentiellement de la représentation des formes utilisée, c'est-à-dire du prétraitement effectué sur les données brutes. Si la représentation des formes est bien choisie, et si le réseau est bien conçu, les réseaux de neurones offrent des performances équivalentes à celles des meilleurs classifieurs en termes de taux de reconnaissance, mais ils se distinguent de ceux-ci par une plus grande facilité d'implantation sous forme de circuits spécialisés très rapides. Néanmoins, il faut souligner que l'on doit bien se garder d'utiliser systématiquement les réseaux de neurones pour tout problème de classification : il faut d'abord évaluer la difficulté du problème à traiter, et n'utiliser les réseaux de neurones que lorsque c'est réellement nécessaire.

Comme nous l'avons indiqué plus haut, les réseaux de neurones, contrairement à une idée reçue, ne se contentent pas de prendre la décision d'attribuer un objet à une classe. Il permet d'obtenir une information beaucoup plus riche : *la probabilité d'appartenance de l'objet à chacune des classes*. À partir de cette information, il est possible de prendre une décision "douce" ; ceci permet notamment de combiner cette probabilité avec des informations provenant, par exemple, d'autres systèmes de classification, utilisant d'autres algorithmes ou d'autres représentations des formes. La vue caricaturale du réseau de neurones prenant une décision d'appartenance ou de rejet doit donc être oubliée au profit d'une vision plus complète de ce que savent réellement apporter les réseaux dans le domaine de la classification automatique [1]. A titre d'exemple de systèmes industriels complexes utilisant les réseaux de neurones en collaboration avec d'autres techniques, citons l'utilisation par la poste française de machines de tri automatique faisant intervenir des réseaux de neurones pour la reconnaissance du code postal. Plus spectaculaire encore, les réseaux de neurones sont utilisés industriellement pour la lecture automatique du montant écrit en toutes lettres (en écriture cursive) sur les chèques bancaires et postaux (Figure 10) [14], [15].

Les problèmes de classification ne se posent pas uniquement pour la reconnaissance des formes : ainsi, la Caisse des Dépôts et Consignations utilise un réseau de neurones pour l'évaluation de la qualité d'actions françaises comme supports d'investissements. Un échantillon de 350 entreprises, notées (de A à C) par des experts à partir de 45 ratios financiers a été utilisé pour concevoir un réseau de neurones, qui, chaque mois, évalue un certain nombre d'entreprises, et fournit notamment une liste des entreprises qui ont changé de classe au cours des derniers mois. Une caractéristique intéressante de ce développement a été l'utilisation de méthodes statistiques de sélection d'entrées, afin de déterminer, parmi les nombreux ratios que les experts envisageaient de mettre en œuvre, ceux qui étaient réellement pertinents. Ce type d'*outil d'aide à la décision* semble constituer un débouché particulièrement prometteur pour les réseaux de neurones.

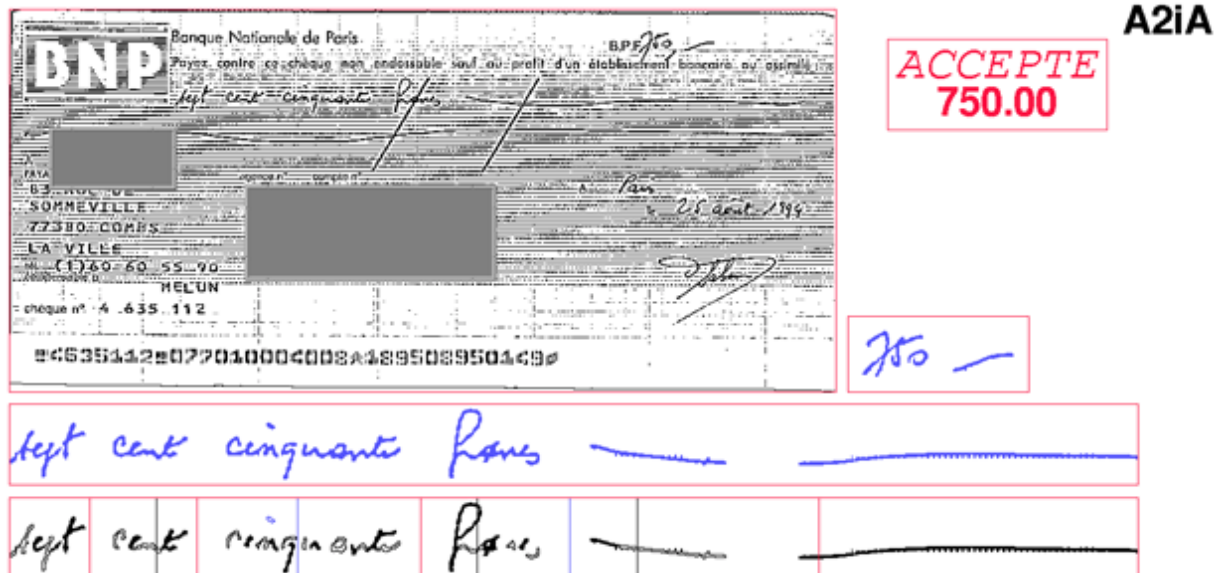


Figure 10

La société A2iA a mis au point un système opérationnel de lecture automatique des montants littéraux de chèques

## 12. RESEAUX DE NEURONES ET FLOU

Pour des raisons commerciales, beaucoup plus que techniques ou scientifiques, les réseaux de neurones sont souvent, dans l'esprit des ingénieurs et des décideurs, associés au flou, notamment pour la modélisation et la commande de processus. Ces deux techniques n'ont pourtant à peu près rien en commun, que ce soit dans leur esprit, dans leurs modalités de mise en œuvre, ou dans leurs domaines d'application.

Rappelons que les réseaux de neurones sont des approximateurs universels parcimonieux ; ils peuvent donc être utilisés pour modéliser ou commander tout processus, statique ou dynamique, non linéaire : en raison de leur parcimonie, ils sont avantageux par rapport aux autres approximateurs - et notamment au flou - dès que le processus à modéliser ou à commander possède plus de deux ou trois entrées. Néanmoins, comme toute autre technique, les réseaux de neurones sont soumis à des contraintes : étant des outils statistiques, ils traitent uniquement de données *numériques*, dont le nombre et la représentativité doivent être convenables - même si, comme nous l'avons indiqué plus haut, leur parcimonie leur permet d'utiliser moins de données que d'autres méthodes statistiques. S'il est possible de tirer profit, pour la conception du réseau, des connaissances, même imprécises, que l'on peut avoir sur le processus, il faut qu'elles soient sous forme *mathématique* : les réseaux de neurones ne permettent pas de traiter aisément des données linguistiques.

En revanche, les systèmes flous sont par essence capables de modéliser ou de commander des processus sur lesquels existe une *expertise linguistique imprécise*. Ils sont en revanche d'utili-



sation malaisée, voire impossible, ou sans intérêt car les règles deviennent trop compliquées pour être interprétables, pour des processus comprenant plus de deux ou trois entrées. Les deux techniques sont donc complètement différentes. On trouvera dans [17] une mise en perspective des méthodes classiques, neuronales et floues pour la modélisation et la commande de processus.

### 13. CONCLUSION

Nous avons exposé les éléments essentiels qui permettent de comprendre pourquoi, et dans quels cas, il est avantageux de mettre en œuvre des réseaux de neurones. En présentant quelques applications typiques, nous avons tenté de montrer, de manière concrète, ce que l'ingénieur peut attendre de cette technique. S'ils sont bien compris et convenablement utilisés, les réseaux de neurones peuvent rendre des services considérables dans des domaines très variés.

Résumons donc les points fondamentaux qu'il convient de toujours garder à l'esprit lorsque l'on cherche à mettre en œuvre des réseaux de neurones :

- les réseaux de neurones sont des outils *statistiques*, qui permettent d'ajuster des fonctions non linéaires très générales à des ensembles de points ; comme toute méthode statistique, l'utilisation de réseaux de neurones nécessite que l'on dispose de données suffisamment nombreuses et représentatives ;
- les réseaux de neurones sont des approximateurs *parcimonieux* ;
- les réseaux de neurones permettent de modéliser des phénomènes statiques (réseaux non bouclés) et dynamiques (réseaux bouclés) ;
- il est toujours souhaitable, et souvent possible, d'utiliser, pour la conception du réseau, les connaissances mathématiques dont on dispose sur le phénomène à modéliser : les réseaux de neurones ne sont pas nécessairement des "boîtes noires".

### RÉFÉRENCES

- [1] On ne saurait trop recommander la lecture d'un excellent ouvrage, portant sur les réseaux de neurones non bouclés et leur mise en œuvre pour la classification automatique : C. BISHOP, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
- [2] G. DREYFUS, Y. IDAN, "The Canonical Form of Nonlinear Discrete-Time Models", Neural Computation Vol. 10, n°1, 1998.
- [3] K. HORNIK, M. STINCHCOMBE, H. WHITE, "Multilayer Feedforward Networks are Universal Approximators", Neural Networks Vol. 2, pp. 359-366, 1989

- [4] K. HORNIK, M. STINCHCOMBE, H. WHITE, P. AUER, "Degree of approximation results for feedforward networks approximating unknown mappings and their derivatives", *Neural Computation* Vol. 6, pp. 1262-1275, 1994.
- [5] D. URBANI, P. ROUSSEL-RAGOT, L. PERSONNAZ, G. DREYFUS, "The selection of neural models of nonlinear dynamical systems by statistical tests", *Neural Networks for Signal Processing I*, pp. 229-237, IEEE Press, 1994.
- [6] W.H. PRESS, S.A. TEUKOLSKY, W.T. VETTERLING, B.P. FLANNERY, *Numerical Recipes in C : the Art of Computing*, Cambridge University Press, 1992.
- [7] O. NERRAND, P. ROUSSEL-RAGOT, L. PERSONNAZ, G. DREYFUS, "Neural networks and nonlinear adaptive filtering : unifying concepts and new algorithms", *Neural Computation* Vol.5, pp. 165-199, 1993.
- [8] A. F. DUPRAT, T. HUYNH, G. DREYFUS, "Parsimonious, Accurate Neural Network Prediction of LogP", *J. Chem. Inf. Comput. Sci.*, à paraître.
- [9] I. RIVALS, L. PERSONNAZ, G. DREYFUS, J.L. PLOIX, "Modélisation, classification et commande par réseaux de neurones : principes fondamentaux, méthodologie de conception et applications industrielles", *Les réseaux de neurones pour la modélisation et la commande de processus*, J.P. Corriou éd., Lavoisier Tec et Doc, 1995.
- [10] G. DREYFUS, J.L. PLOIX, Early Fault Detection in a Distillation Column: an Industrial Application of Knowledge-based Neural Modelling, *in Best Neural Network Practice in Europe*, World Scientific, 1998.
- [11] J.L. PLOIX, G. DREYFUS, Knowledge-based Neural Modeling: Principles and Industrial Applications, *in Industrial Applications of Neural Networks*, F. Fogelman, P. Gallinari, eds., World Scientific, 1997.
- [12] I. RIVALS, Modélisation et commande par réseaux de neurones : application au pilotage d'un véhicule autonome, Thèse de Doctorat de l'Université Pierre et Marie Curie, 1995.
- [13] I. RIVALS, D. CANAS, L. PERSONNAZ, G. DREYFUS, "Modeling and Control of Mobile Robots and Intelligent Vehicles by Neural Networks", *IEEE Conference on Intelligent Vehicles*, pp. 137-142, 1994.
- [14] D. PRICE, S. KNERR, L. PERSONNAZ, G. DREYFUS, "Pairwise Neural Network Classifiers with Probabilistic Outputs", *Neural Information Processing Systems*, vol. 7, Morgan Kaufman, 1994.
- [15] S. KNERR, V. ANISIMOV, O. BARET, N. GORSKI, D. PRICE, J.C. SIMON, "The A2iA Recognition System for Handwritten Checks", *International Journal of Pattern Recognition and Artificial Intelligence*, à paraître.
- [16] H. STOPPIGLIA, Y. IDAN, G. DREYFUS, "Neural-Network-Aided Portfolio Management", *in Industrial Applications of Neural Networks*, F. Fogelman, P. Gallinari, eds., World Scientific, 1997.

- [17] G. DREYFUS, P.Y. GLORENNEC, M. GRABISCH, P. de LARMINAT, "Complémentarités et spécificités de la logique floue, des réseaux de neurones et de l'automatique linéaire dans le domaine de la modélisation et de la commande de processus", document de synthèse des Clubs CRIN "Réseaux de neurones" et "Logique floue", Association ECRIN, 1997.