

A Learning Scheme for Asymmetric Threshold Network

Une procédure d'apprentissage pour réseau à seuil asymétrique

YANN LE CUN

Ecole Supérieure d'Ingénieurs en Electrotechnique et Electronique

91 rue Falguière 75015 Paris

and

Laboratoire de Dynamique des Réseaux

1 rue Descartes 75005 Paris

COGNITIVA 85

Paris, 4-7 June 1985

Abstract | Résumé

A new parametric method for supervised learning is presented which is based on a threshold network structure. The model is composed of three types of units: input units, output units, and hidden units, the last group having no interaction with the outside world. The learning process is a local iterative scheme which minimizes a particular cost function by modifying the interactions between units. The non-symmetric nature of the weight matrix as well as the modification of the hidden units weights by the learning process constitute the main particularities of this model. This system can learn high order binary predictions which are not linearly separable in the input space. Simulations have been performed using hierarchical networks containing several hundred cells. The network exhibits generalization qualities (i.e. production of a correct output for a previously unobserved input pattern) on a low-resolution noisy pattern recognition task. Other simulations have been done in self-supervised learning (i.e. with desired outputs inputs derived from inputs) that model Pavlovian learning and object-symbol associations.

Une nouvelle méthode paramétrique d'apprentissage supervisé utilisant un réseau parallèle d'automates à seuil est proposée. Le modèle est composé de trois types d'éléments: les cellules d'entrée, les cellules de sortie, et les cellules internes, ces dernières n'ayant aucune interaction directe avec l'extérieur. L'apprentissage est un processus itératif local qui minimise une fonction de coût en modifiant les interactions entre cellules. La nature non-symétrique de la matrice de poids ainsi que la modification par l'apprentissage des poids des cellules internes constituent les principales particularités de ce modèle. Ce système peut apprendre des prédicts d'ordre élevé et des discriminations dans le cas non linéairement séparable. Des simulations ont été effectuées sur des réseaux hiérarchiques mettant en jeu plusieurs centaines de cellules. Le réseau fait preuve de capacités de généralisation (i.e. production d'une sortie correcte pour une forme non apprise) dans le cas de la reconnaissance d'images bruitées de basse résolution. D'autres simulations ont été réalisées en conditions d'auto-apprentissage (i.e. avec une sortie désirée auto-générée) pour modéliser l'apprentissage Pavlovien et les associations objet-symbole.

1 Introduction

Highly parallel cellular logic systems have recently sparked numerous works concerning diverse domains such as form recognition [2,13], memories associated

Les systèmes logiques cellulaires hautement parallèles ont récemment suscité de nombreux travaux concernant des domaines aussi divers que la Recon-

with neural systems [3-9,12]. A particularly studied class of cellular network in these works is that of threshold automata networks. The threshold automaton is an operator possessing several inputs and one output representing the truth value of a predicate by taking values +1 or -1. The state of the automaton, which also constitutes its output, is calculated on a discrete time scale and is defined by:

$$X_i(t) = f \left(\sum_j W_{ij} X_j(t-1) \right) \quad (1)$$

with

$$f(Z) = \begin{cases} +1 & \text{if } Z \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

where $X_i(t)$ is the state of automaton i at instant t and W_{ij} is the weight of the connection linking automaton j to automaton i . Assuming that the update of states of elements occurs synchronously (parallel iteration), it is possible to write matricially the evolution equation of the network:

$$X(t) = f(W \cdot X(t-1)) \quad (2)$$

where $X(t)$ is the state vector of the network at instant t , W is the weight matrix, and f is the vector-valued function applying f to each component of its argument. The theoretical tools for characterizing the dynamic behavior of threshold networks are diverse and relate to statistical physics [4,13], or to discrete mathematics [1,3].

It is appropriate to describe the type of learning algorithm to be implemented. The goal is to teach the network to perform associations between a set of objects and a set of output. Each object (or action) is represented by a vector of bits associated with the ground truth values of a set of predicates that describe it.

The level of abstraction of these output predicates can be arbitrary and may correspond just as well to the pixels of a binary image as to abstract concepts or features. A training session consists of supplying the network with a series input-output pairs, each composed of an input pattern and the output that is to be associated with that pattern. Three types of neurons are involved: input neurons whose state is coupled to the input values, output neurons corresponding to the output / actions / terminal predicates, and internal neurons associated with intermediate representations which have no direct interaction with the outside.

naissance des formes [2,13], les mémoires associatives ou la modélisation des systèmes nerveux animaux [3-9,12]. Une classe particulière de réseau cellulaire largement étudiée dans ces travaux est celle des réseaux d'automates à seuil. L'automate à seuil est un opérateur possédant plusieurs entrées et une sortie représentant la valeur de vérité d'un prédicat en prenant les valeurs +1 ou -1. L'état de l'automate, qui constitue également sa sortie, est calculé sur une échelle de temps discrète et est défini par:

où $X_i(t)$ est l'état de l'automate i à l'instant t et W_{ij} est le poids de la liaison reliant l'automate j à l'automate i . En supposant que la mise à jour des états des éléments s'effectue de manière synchrone (itération parallèle), il est possible d'écrire matriciellement l'équation d'évolution du réseau:

où $X(t)$ est le vecteur d'état du réseau à l'instant t , W est la matrice des pondérations, et f est la fonction vectorielle appliquant f à chaque composante de son argument. Les outils théoriques permettant de caractériser le comportement dynamique des réseaux à seuil sont très divers et relèvent de la physique statistique [4,13], ou des mathématiques discrètes [1,3].

Il convient de décrire le type d'apprentissage devant être mis en œuvre. Le but est d'apprendre au réseau à effectuer des associations entre un ensemble d'objets et un ensemble d'actions. Chaque objet ou action est représenté par une chaîne de bits associée aux valeurs de vérité d'un ensemble de prédicats qui le décrit.

Le niveau d'abstraction de ces prédicats peut être quelconque et correspondre aussi bien aux pixels d'une image binaire qu'à des caractéristiques abstraites. Une session d'apprentissage consiste à fournir au réseau une série de paires, chacune constituée d'une forme et de l'action devant être associée à cette forme. Trois types de cellules sont donc mis en jeu: les cellules d'entrée dont l'état est couplé aux prédicats d'entrée, les cellules de sortie correspondant aux actions ou aux prédicats terminaux, et les cellules internes associées aux prédicats intermédiaires et n'ayant aucune interaction directe avec l'extérieur.

An important characteristic, without which one merely has simple memorization, is *generalization*, that is, the extension of correct predictive behavior to patterns that have not been observed. It seems extremely important to bring this type of behavior to light through simulations. In order to be able to solve a wide class of problems, a threshold network must contain elements whose state is not directly determined from the outside; otherwise the accessible domain is limited to functions that are linearly separable in the inputs, which is an increasingly severe restriction as the complexity of the patterns grows.

The presence of such internal neurons poses a problem for learning, since when a network produces an incorrect response it is virtually impossible to know which parameter is responsible and how it should be modified. This problem, which we shall call CAP (the “*Credit Assignment Problem*” [5]), is at the root of the limitations of most of the models described in the literature, which generally do not possess internal neurons [11][4]. A solution to CAP was recently proposed [5] for a network of probabilistic threshold automata described by a *symmetric* connection matrix. We propose another solution that can be applied in the case of a non-symmetric matrix, in particular to hierarchical networks organized in layers, whose usefulness for visual perception tasks has often been emphasized [14][2].

The presence of internal neurons enabling the network to learn a very large class of Boolean functions of high dimensionality is an important feature of the proposed model, which removes the limitation to linearly separable functions and makes it possible to generalize learning to higher-order predicates.

Une caractéristique importante, sans laquelle il ne s’agit que d’une simple mémorisation, est la *généralisation*, c’est-à-dire l’extension du comportement correct à des formes non apprises. Il semble extrêmement important de mettre ce type de comportement en évidence par simulations. Pour être capable de résoudre une large classe de problèmes, un réseau à seuil doit comporter des éléments dont l’état n’est pas directement déterminé de l’extérieur ; dans le cas contraire le domaine accessible est limité aux fonctions linéairement séparables des entrées, ce qui est une restriction d’autant plus forte que la dimensionnalité des formes d’entrée est grande.

La présence de telles cellules internes pose un problème en conditions d’apprentissage, en effet lorsqu’un réseau produit une mauvaise réponse il est virtuellement impossible de savoir quel paramètre est en cause et comment il doit être modifié. Ce problème, que nous appellerons CAP (pour “*Credit Assignment Problem*” [5]), est à l’origine des limitations de la plupart des modèles décrits dans la littérature, ne possédant généralement pas de cellule interne [11][4]. Une solution au CAP a récemment été proposée [5] pour un réseau d’automates à seuil probabilistes décrit par une matrice de connexions *symétrique*. Nous proposons une autre solution pouvant s’appliquer au cas d’une matrice non symétrique, en particulier aux réseaux hiérarchiques organisés en couches dont l’intérêt pour les tâches de perception visuelle a souvent été mis en avant [14][2].

La présence de cellules internes permettant au réseau d’apprendre une très large classe de fonctions booléennes de grande dimensionnalité est une caractéristique importante du modèle proposé, qui permet de lever la limitation aux fonctions linéairement séparables et de généraliser l’apprentissage à des prédicts d’ordre élevé.

2 Model Description | Description Du Modèle

At each instant on the discrete time scale, an input pattern $E(t)$ is presented to the network. This means that the output of each input cell is clamped to the corresponding component of $E(t)$. The states of the other cells then evolve for several successive time steps according to the threshold equation introduced earlier. The output equation of cell i can therefore be written as:

$$X_i(t) = \begin{cases} E_{n(i)}(t) \\ f\left(\sum_j W_{ij} X_j(t-1)\right) \end{cases}$$

À chaque instant de l’échelle de temps discrète, une forme $E(t)$ est présentée au réseau. Cela signifie que la sortie de chaque cellule d’entrée est forcée à la composante correspondante de $E(t)$. L’état des autres cellules évolue alors pendant plusieurs instants successifs selon l’équation seuil introduite précédemment. L’équation de sortie de la cellule i peut donc s’écrire:

$$\begin{aligned} &\text{if } i \text{ is the index of an input cell,} \\ &\text{otherwise.} \end{aligned} \tag{3}$$

where $n(i)$ is an index mapping that specifies which component of the input vector is associated with input cell i . The evolution of the global network state can then be summarized as

$$X(t) = G_x(W(t-1), X(t-1), E(t)), \quad (4)$$

where the W_{ij} are now regarded as dynamic variables and G_x is a function that is directly defined from (3).

2.1 Minimizing a Cost Function

Existing supervised learning procedures are generally based on the minimization of a cost function, whether this is explicit or implicit. The method presented here is no exception. Suppose that at any instant we have a vector $Y(t)$ defining the ideal state of each cell for the task the network has to perform. We can then define the cost function as the mean squared error between the actual state X and this ideal state Y :

$$\langle B \rangle = \left\langle \sum_i (Y_i - X_i)^2 \right\rangle \quad (5)$$

where the symbol $\langle \cdot \rangle$ denotes an average over time steps. In matrix form:

$$\langle B \rangle = \langle (Y - X)^T (Y - X) \rangle \quad (6)$$

We seek a set of weights W_{ij} that minimizes this quantity. Doing so optimally is difficult, but it can be done suboptimally using a multidimensional version of the Widrow–Hoff procedure [15]. The minimized criterion is then

où $n(i)$ est un changement d'indices décrivant quelle composante du vecteur d'entrée est associée à la cellule d'entrée i . L'évolution de l'état global du réseau peut alors être résumée par

où les W_{ij} sont maintenant considérées comme des variables dynamiques et G_x est une fonction directement définie à partir de (3).

2.1 Minimiser une fonction de coût

Les procédures d'apprentissage supervisé existantes sont généralement basées sur la minimisation d'une fonction de coût, que celle-ci soit explicite ou implicite. La méthode présentée ici n'échappe pas à cette règle. Supposons qu'à tout instant nous disposons d'un vecteur $Y(t)$ définissant l'état idéal de chaque cellule pour la tâche que le réseau doit réaliser. Il est alors possible de définir la fonction de coût comme l'écart quadratique moyen entre l'état réel X et cet état idéal Y :

où le symbole $\langle \cdot \rangle$ représente la moyenne temporelle. Ceci peut s'écrire matriciellement:

Il s'agit de trouver une configuration de pondérations W_{ij} qui minimise cette quantité. Ceci est difficilement réalisable de manière optimale, mais peut être fait de manière sous-optimale en utilisant une version multidimensionnelle de la procédure de Widrow–Hoff [15]. Le critère minimisé devient alors

$$\langle C \rangle = \langle (Y - A)^T (Y - A) \rangle \quad (7)$$

$$A(t) = W(t-1) X(t-1) \quad (8)$$

The procedure uses a stochastic gradient:

La procédure utilise un gradient stochastique:

$$W(t) = W(t-1) - K \cdot \text{Grad}_W(C) \quad (9)$$

where K is a positive matrix, chosen to be diagonal, which defines each iteration step. After expansion, equation (9) becomes:

où K est une matrice positive que nous choisirons diagonale et qui définit le pas d'itération. Après développement, l'équation (9) devient:

$$W_{ij}(t) = W_{ij}(t-1) + K_i (Y_i(t) - A_i(t)) X_j(t-1) \quad (10)$$

This procedure and its applications to linear associative memories have been extensively studied in the literature.

Cette procédure et ses applications aux mémoires associatives linéaires ont été longuement étudiées dans la littérature.

2.2 Calculation of Ideal States

The problem is therefore to find a method for computing, at each instant, the ideal state Y of the network from the desired state of the output neurons. In other words, we want to identify a function G_y such that:

$$Y(t) = G_y(W(t-1), Y(t-1), E(t), D(t)) \quad (11)$$

where $D(t)$ is the configuration of the output cells that represents the desired output to be associated with the input $E(t)$.

The model uses the following definition for G_y :

$$Y_i(t) = \begin{cases} E_{n(i)}(t) & \text{if } i \text{ is an input cell index,} \\ D_{n(i)}(t) & \text{if } i \text{ is an output cell index,} \\ f\left(\sum_j W_{ij} X_j(t-1)\right) & \text{otherwise.} \end{cases}$$

where $n(i)$ is an index mapping derived from the association between the components of the input or desired output vectors and the corresponding cells.

The overall behaviour of the system is thus governed by the following set of equations:

$$\begin{aligned} X(t) &= G_x(W(t-1), X(t-1), E(t)), \\ Y(t) &= G_y(W(t-1), Y(t-1), E(t), D(t)), \\ W_{ij}(t) &= W_{ij}(t-1) + K_i(Y_i(t) - A_i(t)) X_j(t-1). \end{aligned} \quad (13)$$

Note that the learning criterion is completely local in time and space: the modification of a weight only involves quantities that are available at the two cells it connects. When there are no internal cells, the criterion is convex and the Widrow–Hoff procedure converges to a global minimum. In the general case, with internal cells, the criterion is no longer convex and is defined only in a statistical sense through the temporal average.

2.2 Calcul des états idéaux

Le problème est donc de trouver une méthode permettant de calculer, à chaque instant, l'état idéal Y du réseau à partir de l'état désiré des cellules de sortie. Autrement dit, il s'agit d'identifier une fonction G_y telle que:

$$Y(t) = G_y(W(t-1), Y(t-1), E(t), D(t)) \quad (11)$$

où $D(t)$ est la configuration des cellules de sortie représentant la sortie désirée devant être associée à l'entrée $E(t)$.

Le modèle utilise pour G_y la définition suivante:

$$\begin{aligned} &\text{if } i \text{ is an input cell index,} \\ &\text{if } i \text{ is an output cell index,} \\ &\text{otherwise.} \end{aligned} \quad (12)$$

où $n(i)$ est un changement d'indices dérivé de l'association entre les composantes des vecteurs d'entrée ou de sortie désirée et les cellules correspondantes.

Le comportement global du système est ainsi régi par le système d'équations suivant:

On notera que le critère d'apprentissage est totalement local dans le temps et dans l'espace: la modification d'une pondération ne fait intervenir que des grandeurs disponibles localement au niveau des deux cellules qu'elle relie. Lorsque le réseau ne comporte pas de cellules internes, le critère est convexe et la procédure de Widrow–Hoff converge vers un minimum global. Dans le cas général, en présence de cellules internes, ce critère n'est plus convexe et n'est défini que de manière statistique, par la moyenne temporelle.

3 Simulations

For the simulations, the choice was made for a particular network structure, oriented towards image recognition applications.

3.1 Description of network architecture

Three essential properties characterize the network's structure:

Pour les simulations le choix a été porté sur une structure particulière de réseau, orientée vers une application à la reconnaissance d'image.

3.1 Description de l'architecture du réseau

Trois faits essentiels caractérisent cette structure: Premièrement, les connexions entre cellules sont

Firstly, the connections between cells are local. This is done, on one hand, to allow taking into account the short-distance correlations specific to visual information, and on the other hand, to maintain compatibility with an eventual hardware implementation, since long-distance interactions are technologically costly.

Secondly, the network is hierarchical, organized in successive processing layers. This architecture allows the elaboration of information moving towards increasing levels of abstraction.

Thirdly, the network contains no loops, for several reasons. The first is that the absence of circuits in the connection graph allows trivial characterization of the fixed points of the dynamics of free states and ideal states. This simplification allows easier study of the phenomenon that particularly concerns us, that is, the evolution of weights. The second reason is the computational economy provided by this architecture. It is however useful to insist on the fact that this does not constitute a restriction in principle.

The network used comprises a series of layers of 64 cells arranged in 8×8 matrices, with the number of layers variable between 1 and 10. The receptive field of each cell in a layer consists of a neighborhood in the immediately lower layer containing 9, 25 or 49 cells (3×3 , 5×5 or 7×7 neighborhoods). There is no interaction between cells of the same layer. The output cells are variable in number depending on the task to be performed and take their inputs from the 64 cells of the last layer. The value of the convergence coefficient K is identical for all cells except the output cells.

Each learning iteration is divided into 3 steps:

- Positioning of the free state of input cells corresponding to a shape, and calculation of the stable free state of the network.
- Positioning of the ideal state of output cells corresponding to a desired response, and calculation of the stable ideal state of the network.
- Application of one iteration of the synaptic modification rule.

3.2 Simulation examples and generalization

The example chosen to test the procedure is character recognition. Naturally, this is only a test but clearly demonstrates the complexity of Boolean func-

locales. D'une part pour permettre la prise en compte des corrélations à courte distance spécifiques des informations de type visuel, et d'autre part pour garder la compatibilité avec une éventuelle implémentation matérielle, les interactions à longue distance étant technologiquement coûteuses.

Deuxièmement, le réseau est hiérarchique, organisé en couches de traitement successives. Cette architecture permet l'élaboration des informations allant vers des niveaux d'abstraction croissants.

Troisièmement, le réseau ne comporte pas de boucle et ceci pour plusieurs raisons. La première est que l'absence de circuit dans le graphe de connexion permet de caractériser trivialement les points fixes de la dynamique des états libres et des états idéaux. Cette simplification permet d'étudier plus facilement le phénomène qui nous occupe particulièrement c'est-à-dire l'évolution des pondérations. La seconde raison est l'économie de calculs que procure cette architecture. Il est toutefois utile d'insister sur le fait que ceci ne constitue pas une restriction de principe.

Le réseau utilisé comporte une série de couches de 64 cellules rangées en matrices 8×8 , le nombre de couches est variable entre 1 et 10. Le champ récepteur de chaque cellule d'une couche est constitué d'un voisinage de la couche immédiatement inférieure comportant 9, 25 ou 49 cellules (voisinage 3×3 , 5×5 ou 7×7). Il n'y a pas d'interaction entre cellules d'une même couche. Les cellules de sortie sont en nombre variable dépendant de la tâche à réaliser et prennent leurs entrées sur les 64 cellules de la dernière couche. La valeur du coefficient de convergence K est identique pour toutes les cellules exceptées les cellules de sortie.

Chaque itération d'apprentissage est divisée en 3 étapes:

- Positionnement de l'état libre des cellules d'entrées correspondant à une forme, et calcul de l'état libre stable du réseau.
- Positionnement de l'état idéal des cellules de sortie correspondant à une réponse désirée, et calcul de l'état idéal stable du réseau.
- Application d'une itération de la règle de modification synaptique.

3.2 Exemples de simulations, généralisation

L'exemple choisi pour le test de la procédure est la reconnaissance de caractères. Naturellement ceci n'a qu'une valeur de test mais démontre bien la

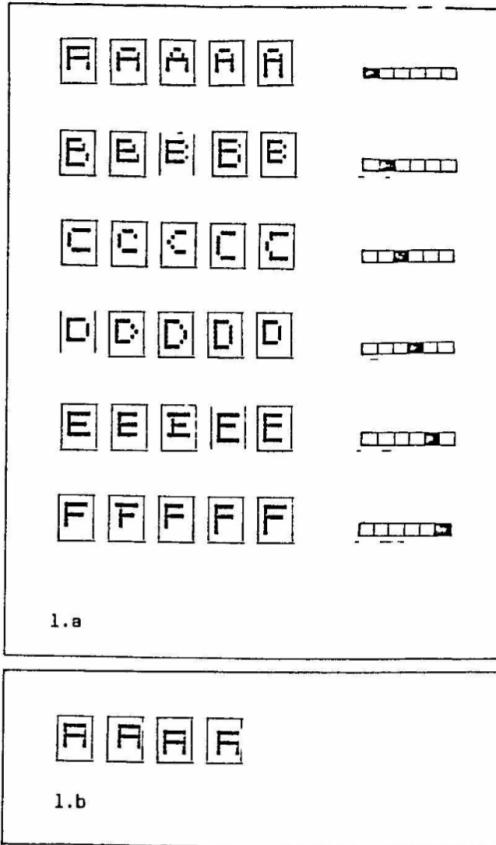


Figure 1: (a) Training set with associated responses / *Ensemble d'apprentissage avec les réponses associées*
 (b) Each shape is presented in four different positions / *Chaque forme est présentée dans 4 positions différentes*

tions that the network can synthesize. Furthermore, with this type of data, the observer has a semantic criterion (the only valid one, though subjective) for judging the quality of generalization.

Figure 1 shows the shapes in the training set. Five examples of the first letters of the alphabet are presented (Fig. 1.a), each in four different positions (Fig. 1.b), giving a total of 120 different shapes grouped into 6 classes of 20 shapes. Each class is associated with the activation of only one of the 6 output cells. Correct classification of the 120 shapes is obtained with a network comprising five layers (including input cells but not counting output cells) and a connectivity of 25.

This task is relatively complex since shapes that are very different (in terms of Hamming distance) must be grouped into the same class, while shapes that are extremely close must be differentiated. For example, the distance between the third B and the second E in Figure 1.a is 2 pixels, whereas the distance between that same B and its version translated downward and to the right is 30.

complexité des fonctions booléennes que le réseau peut synthétiser. En outre, sur ce type de données l'observateur dispose d'un critère sémantique (le seul valable, bien que subjectif) pour juger de la qualité de la généralisation.

Sur la figure 1 sont représentées les formes de l'ensemble d'apprentissage. Cinq exemples des premières lettres de l'alphabet sont présentées (Fig. 1.a), chacune dans quatre positions différentes (Fig. 1.b), soit au total 120 formes différentes groupées en 6 classes de 20 formes. À chacune des classes est associée l'activation d'une seule des 6 cellules de sortie. La classification correcte des 120 formes est obtenue avec un réseau comprenant cinq couches (y compris les cellules d'entrée mais sans compter les cellules de sortie) et une connectivité de 25.

Cette tâche est relativement complexe puisque des formes très différentes (au sens de la distance de Hamming) doivent être regroupées en une même classe, alors que des formes extrêmement proches doivent être différencierées. Par exemple, la distance entre le troisième B et le deuxième E de la figure 1.a

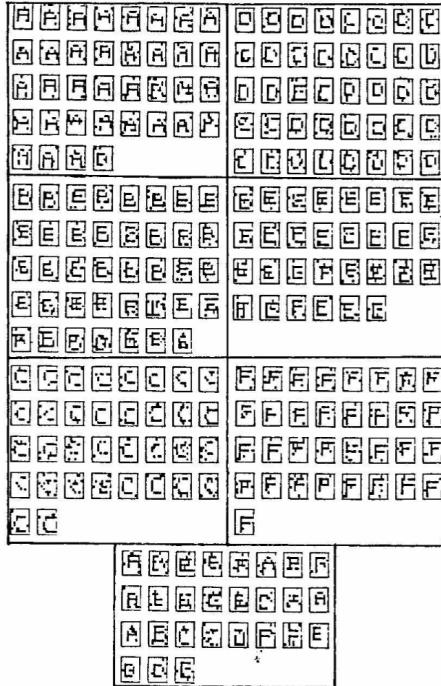


Figure 2: Example of classification performed by the network on noisy images. The set shown at the bottom of the figure groups the unclassified forms. / *Exemple de classement effectué par le réseau sur des images bruitées. L'ensemble représenté en bas de la figure regroupe les formes non classées.*

Figure 2 shows the classification performed by a five-layer network with connectivity 49 when training took place in the presence of noise (5 to 7 percent of pixels are randomly inverted). One can observe fairly frequent confusions between B and E, the differences between these letters being drowned in the noise. The group of shapes at the bottom of the figure is the one for which none of the six desired responses was produced.

Figure 3 shows the classification performed by the previous network on distorted (and translated) shapes different from those used for training, demonstrating the generalization capabilities of the learning procedure as well as the invariance properties of the generated Boolean function.

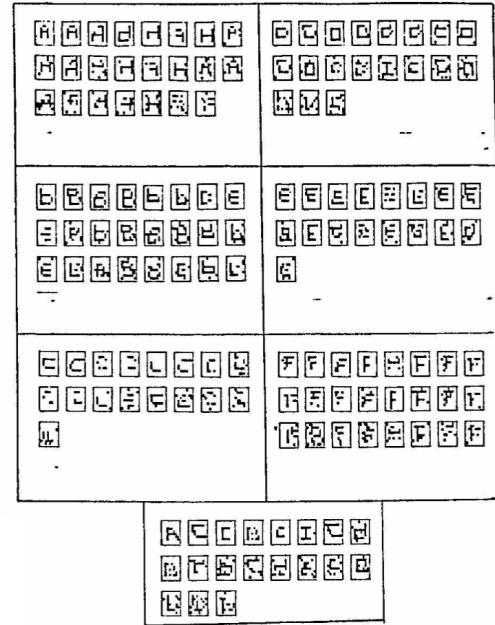


Figure 3: Example of generalization, classification performed on non-learned forms / *Exemple de généralisation, classement opéré sur des formes non apprises*

est de 2 pixels, par contre la distance entre ce même B et sa version translatée vers le bas et la droite est de 30.

La figure 2 montre la classification opérée par un réseau de cinq couches avec connectivité 49 lorsque l'apprentissage s'est déroulé en présence de bruit (5 à 7 pour cent des pixels sont aléatoirement inversés). On peut remarquer des confusions assez fréquentes entre B et E, les différences entre ces lettres étant noyées dans le bruit. Le groupe de formes en bas de la figure est celui pour lequel aucune des six réponses désirées n'a été produite.

La figure 3 montre la classification opérée par le réseau précédent sur des formes distordues (et translatées) différentes de celles qui ont servi à l'apprentissage, ceci pour montrer les capacités de généralisation que possède la procédure d'apprentissage ainsi que les propriétés d'invariance de la fonction booléenne générée.

3.3 Self-supervised learning and Pavlovian association

The kind of learning just described, which is closer to animal conditioning, was tested on a simple example. A training session is split into three phases, the last two being intended to model classical Pavlovian conditioning.

During all three phases, the input layer (or, the “retina”) is divided into two interlaced parts on which patterns of different nature are projected. One part, called retina I, contains the even-numbered rows and receives the *unconditional stimuli* (US). The other part, retina C, contains the odd-numbered rows and receives *neutral or conditional stimuli* (CS). Figure 4 summarizes the three phases.

3.3 Auto-apprentissage, associations Pavloviennes

Le type d’apprentissage décrit ci-dessus, plus proche des comportements cognitifs animaux, a été testé sur un exemple simple. Une session est découpée en trois phases dont les deux dernières ont pour but de modéliser le conditionnement pavlovien classique.

Durant ces trois phases, l’ensemble des cellules d’entrée (la «rétine») est partagé en deux parties recevant des formes de nature différente. L’une, appelée rétine I, regroupe les lignes d’ordre pair et reçoit les *stimuli inconditionnels* (SI). L’autre, la rétine C, regroupe les lignes d’ordre impair et reçoit les *stimuli neutres ou conditionnels* (SC). La figure 4 résume les trois phases.

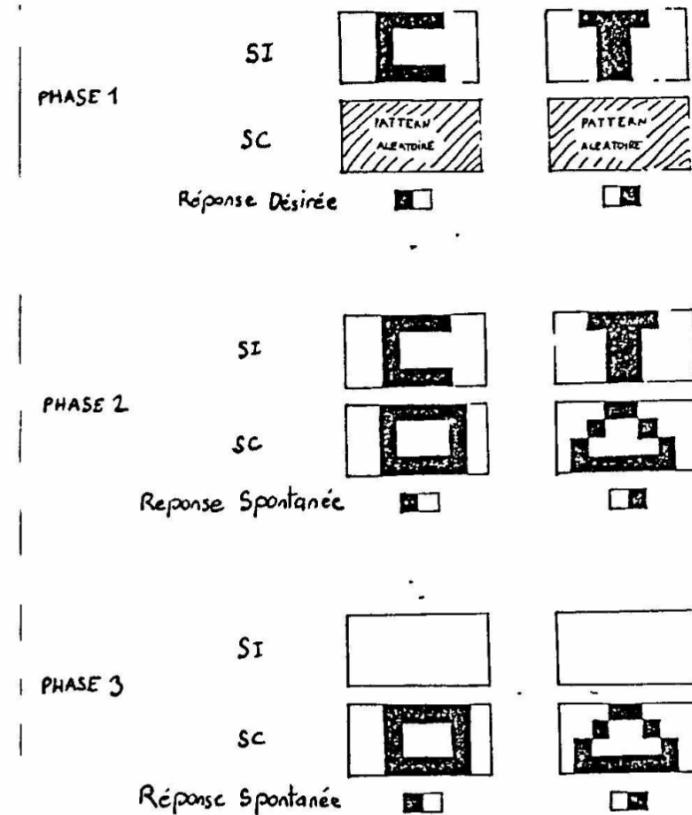


Figure 4: Learning phases of Pavlovian conditioning / *Phases d’apprentissage dans le conditionnement Pavlovien*

Phase 1: This phase has no direct counterpart in biological Pavlovian experiments. Its purpose is to create an *unconditioned response* in the network, i.e. the natural association between a food stimulus and an action.

Two letter patterns, C and T, are used as un-

Phase 1: Cette phase n’a pas d’équivalent direct dans les expériences pavloviennes biologiques. Elle sert à créer dans le réseau une *réaction inconditionnelle*, c’est-à-dire l’association « naturelle » entre un stimulus de nourriture et une action.

Deux formes en lettres, C et T, sont utilisées

conditional stimuli on retina I. They correspond to the presence of food on the left or right arm of a maze and are associated with two actions (turn left or turn right). Learning is supervised as in the previous experiments: at each iteration the network is shown one of the pairs (SI, neutral pattern) illustrated in Fig. 4 on the two retinas, and the desired output pattern is imposed on the output cells. Random black/white pixel noise is added, in particular on retina C when no conditional stimulus is present.

Phase 2: This phase performs conditioning in self-supervised learning mode. The desired response is no longer supplied from the outside but taken to be the network’s own spontaneous response.

At each iteration one of the pairs (square on retina C, noisy C on retina I) or (triangle on retina C, noisy T on retina I) is presented. After a sufficient number of presentations the weights are frozen. During this phase the presence of both the unconditional stimulus on retina I and the conditional stimulus on retina C is required in order to obtain the appropriate reaction.

Phase 3: In the final phase the unconditional stimulus is removed. Retina I now receives only background noise, while the noisy square or triangle alone is projected on retina C. As indicated schematically in Fig. 4, the network still produces the correct spontaneous response (turn left or right) in the absence of the food stimulus. This behaviour can be interpreted as a Pavlovian association between the geometric pattern on retina C and the action represented by the output cells.

It is worth noting that fairly strong noise on the inputs does not prevent this association from being formed.

comme stimuli inconditionnels sur la rétine I. Elles représentent la présence de nourriture sur la branche gauche ou droite d’un labyrinthe en T et sont associées à deux actions (tourner à gauche ou à droite). L’apprentissage est supervisé comme précédemment : à chaque itération on présente au réseau l’un des couples (SI, forme neutre) de la fig. 4 sur les deux rétines, et la réponse désirée est imposée sur les cellules de sortie. Un bruit (pixels aléatoirement noirs ou blancs) est ajouté, en particulier sur la rétine C lorsqu’aucun stimulus conditionnel n’est présent.

Phase 2 : Cette phase correspond au conditionnement en auto-apprentissage. La réponse désirée n’est plus imposée de l’extérieur mais prise égale à la réponse spontanée du réseau.

À chaque itération, on présente au réseau un des couples (sur la rétine C un carré, sur la rétine I la lettre C bruitée) ou (sur la rétine C un triangle, sur la rétine I la lettre T bruitée). Après un nombre suffisant de présentations, la configuration des pondérations est figée. Au cours de cette phase, la présence conjointe du stimulus inconditionnel sur la rétine I et du stimulus conditionnel sur la rétine C est nécessaire pour provoquer la réaction appropriée.

Phase 3: Dans la phase finale, le stimulus inconditionnel est supprimé. La rétine I ne reçoit plus que du bruit de fond, tandis que seul le carré ou le triangle bruité est projeté sur la rétine C. Comme l’indique la fig. 4, le réseau produit néanmoins la réponse spontanée correcte (tourner à gauche ou à droite) en l’absence du stimulus de nourriture. Ce comportement s’interprète comme l’apparition d’une association pavloviennne entre la forme géométrique présentée sur la rétine C et l’action représentée par les cellules de sortie.

On peut remarquer que la présence d’un bruit assez important sur les entrées n’empêche pas la mise en place de cette association.

4 Conclusion

Several additional tests under different conditions should be conducted. In particular it would be good to evaluate the effect of various parameters on quality and learning time (value of K, number of layers, etc...) as well as testing the case where the network connectivity graph is recurrent.

Nevertheless, it would be interesting to test this type of structure on data that are more abstract than image pixels. This would make it possible to compare it with other inductive inference methods, which generally operate on lower-dimensional descriptions. It seems that the neural network ap-

Plusieurs tests additionnels dans des conditions différentes doivent être effectués. En particulier pour évaluer l’influence des divers paramètres sur la qualité et le temps d’apprentissage (valeur de K, nombre de couches...) ainsi que pour tester le cas où le graphe de connexion comporte des circuits.

Néanmoins, il serait intéressant de tester ce type de structure sur des données plus abstraites que les pixels d’une image. Ceci permettrait la comparaison avec d’autres méthodes d’inférence inductive traitant généralement des descriptions de dimensionnalité plus faible. Il semble que l’approche des réseaux

proach provides a partial answer to the problems of massive parallelism required for the real-time execution of artificial perception tasks, and moreover is a good fit for highly regular hardware architectures, which is technologically valuable [10].

cellulaires apportent une réponse partielle aux problèmes du parallélisme massif nécessaire à l'exécution en temps réel des tâches de perception artificielle, elle offre en outre des architectures matérielles d'une grande régularité ce qui est technologiquement précieux [10]

5 Acknowledgments | Remerciements

The author wishes to thank Mr. Philippe Metsu with whom this work was initiated.

L'auteur tient à remercier Mr Philippe Metsu avec qui ce travail a été débuté.

6 References | Références

- [1] Amari S.I.: Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Trans. Computers*, Vol C-21, No 11, Nov 72.
- [2] Duff M.J.B.: Computing structures for image processing. Academic Press, 1983.
- [3] Fogelman F., Goles E., Weisbuch G.: Transient length in sequential iteration of threshold functions. *Discrete Applied Math.* 6, 1983.
- [4] Hopfield J.J.: Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. USA*, Nov 84.
- [5] Hinton G.E., Sejnowski T., Ackley D.H.: Boltzman machines, constraint satisfaction networks that learn. Carnegie-Mellon U. tech. report CMU-CS-84-119, May 84.
- [6] Kohonen T.: Correlation matrix memories. *IEEE Trans. Comp.*, Vol C-21, No 4, Apr 72.
- [7] Kohonen T., Ruohonen M.: Representation of associated data by matrix operators. *IEEE Trans. Comp.*, July 73.
- [8] Kohonen T.: An adaptive associative memory principle. *IEEE Trans. Comp.*, Apr 74.
- [9] Kohonen T.: Self-organization and associative memories. Springer, 1984.
- [10] Mead C.A., Conway L.: Introduction to VLSI systems. Addison Wesley, 1980.
- [11] Minsky M., Papert S.: Perceptron. MIT Press, 1968.
- [12] Nakano K.: Associatron, a model of associative memory. *IEEE Trans. Syst. Man Cyb.*, Vol SMC-2, No 3, July 72.
- [13] Peretto P.: Properties of neural networks, a statistical physics approach. *Biological Cybernetics*, 50, 1984.
- [14] Tanimoto S.L., Crettez J.P., Simon J.C.: Alternative hierarchies for cellular logic. *Proc. ICPR Montreal*, 1984.
- [15] Widrow B., Hoff M.E.: Adaptive switching circuits. 1960 IRE WESCON Conv. Record, Part 4, 96-104, Aug 1960.