COGNITIVA 85

Paris, 4-7 Juin 1985

A LEARNING SCHEME FOR ASSYMETRIC THRESHOLD NETWORK.

UNE PROCEDURE D'APPRENTISSAGE POUR RESEAU A SEUIL ASSYMETRIQUE.

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.

RESUME

Une nouvelle méthode paramétrique d'apprentissage supervisé utilisant un réseau parallèle d'automates à seuil est proposée. Le modèle est constitué 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. L'utilisation d'une matrice de connexions assymetrique ainsi que la modification par l'apprentissage des paramètres des cellules internes constituent les principales particularités de ce modèle. Ceci permet l'apprentissage de discriminations dans le cas non linéairement séparable ainsi que la synthèse de prédicats d'ordre élevé. Des simulations effectuées sur un réseau hiérarchique de quelques centaines d'éléments mettent en évidence les capacités de *généralisation* du réseau (production d'une réponse correcte pour une forme non apprise) dans le cas de la reconnaissance d'images bruitées de basse résolution avec réponse invariante par faible translation et distortion. Des simulations en conditions d'*auto-apprentissage* (avec une sortie désirée auto-générée) ont également été effectuées pour modéliser l'apprentissage Pavlovien et les associations objet-symbole.

SUMMARY

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-symetric 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 predicates and discrminations in the non-linearly separable case. Simulations have been performed using hierarchical networks containing several hundred cells. The network exhibits generalization abilities (i.e. production of a correct output for a non-learned input pattern) on a low-resolution noisy picture recognition task. Other simulations have been done in self-learning conditions (i.e. with self-generated desired output) that modelize Pavlovian learning and object-symbol associations.

Y. LE CUN - A LEARNING SCHEME FOR THRESHOLD NETWORK.

1 - INTRODUCTION

Les systèmes logiques cellulaires hautement parallèles ont récemment suscité de nombreux travaux concernant des domaines aussi divers que la Reconnaissance 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 définie par :

$$X_i(t)=F\{\sum_j W_{ij}X_j(t-1)\} \qquad (1)$$

avec $$F(Z) = \begin{cases} +1 \text{ si } Z > 0 \\ -1 \text{ sinon} \end{cases}$$

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 :

$$X(t)=F(W.X(t-1)) \qquad (2)$$

Où X(t) est le vecteur d'état du réseau à l'instant t, W la matrice des pondérations, et F 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 font appel à la physique statistique [4,13], ou aux mathématiques discrètes [1,3]. Il est à remarquer que pour un graphe de connexion particulier, l'ensemble des fonctions réalisables par un réseau à seuil peut être généré en jouant sur des paramètres *continus* que sont les pondérations. La fonction d'apprentissage est réalisée par une modification itérative de ces paramètres souvent appelés poids synaptiques par analogie avec les systèmes nerveux animaux.

Il convient de décrire le type d'apprentissage devant être mis en oeuvre. 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 chaine 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. Une caractéristique importante de tout processus d'apprentissage sans laquelle il n'est qu'une simple mémorisation est la *généralisation*, c'est à dire l'extension d'un 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 possèdant 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'intéret 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 générer par apprentissage des prédicats d'ordre élevé.

2 - DESCRIPTION DU MODELE

A chaque instant t de l'échelle de temps discrète une forme est présentée au réseau, ce qui veut dire que la sortie de chacune des cellules d'entrée est forcée à la valeur (±1) du prédicat associé. La même forme peut éventuellement être présentée à plusieurs instants successifs. L'équation de la sortie d'une cellule est légèrement modifiée par rapport à (1):

soit E(t) le vecteur d'entrée à l'instant t,

$$X_i(t) = \begin{cases} E_{n(i)}(t) \text{ si i est l'indice d'une cellule d'entrée} \\ F\{\sum_j W_{ij}X_j(t-1)\} \text{ sinon} \end{cases} \qquad (3)$$

où n(i) est un changement d'indice décrivant l'association entre les composantes du vecteur d'entrée et les cellules d'entrée. L'équation d'évolution des sorties peut s'écrire sous la forme :

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

où les W_{ij} sont maintenant considéres comme des variables dynamiques et G_x est une fonction aisément définie à l'aide de (3).

2.1 - Minimiser une fonction de cout

Les procédures d'apprentissage supervisées 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 la règle.

Supposons qu'à tout instant nous disposions d'un vecteur Y(t) définissant *l'état idéal* de chaque cellule pour que le réseau réalise sa tâche, il nous est alors possible de définir la fonction de cout comme étant l'écart (quadratique) moyen entre l'état réel X et cet état idéal Y. C'est à dire :

$$<B>=<\sum_i (Y_i-X_i)^2> \qquad (5)$$

où le symbole <.> représente une moyenne temporelle. Ceci peut s'écrire matriciellement :

$$<B>=<(Y-X)^T(Y-X)> \qquad (6)$$

où ^{T} note la transposition.
Il s'agit de trouver une configuration de pondérations W_{ij} minimisant cette quantité. Ceci est difficilement réalisable de manière optimale, on peut cependant le faire de manière suboptimale en utilisant une version multidimensionnelle de la procédure de Widrow-Hoff [15]. Le critère minimisé est alors :

$$<C>=<(Y-A)^T(Y-A)> \qquad (7)$$

avec

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

La procédure utilise un gradient stochastique :

$$W(t)=W(t-1)-K.Grad_W(C) \qquad (9)$$

où K est une matrice positive que nous choisirons diagonale et qui défini le pas d'itération.
Après développement, (9) s'écrit :

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

Cette procédure et ses applications aux mémoires associatives linéaires ont été longuement étudiées [7][8].

2.2 - Calcul des états idéaux

Le problème est donc de trouver une méthode pour calculer *l'état idéal* Y du réseau à chaque instant à partir de l'état désiré des cellules de sortie. C'est à dire identifier une fonction G_y telle que :

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

Où D est la configuration des cellules de sortie représentant la sortie désirée devant être associée à E.
Le modèle utilise pour G_y la définition suivante :

$$Y_i(t) = \begin{cases} E_{n(i)}(t) \text{ si i est l'indice d'une cel. d'entrée} \\ D_{m(i)}(t) \text{ si i est l'indice d'une cel. de sortie} \\ F\{\sum_j W_{ji}X_j(t-1)\} \text{ sinon} \end{cases} \qquad (12)$$

où m(i) est un changement d'indices décrivant l'association entre les composantes du vecteur de sortie désirée et les cellules de sortie.
La dynamique des états idéaux Y est donc du même type que celle des états libres X, et en diffère d'une part par la prise en compte des sorties désirées, et d'autre part par l'utilisation *de la transposée de la matrice de pondérations* .
Le comportement global est donc régi par un système d'équations couplées :

$$\begin{aligned} &X(t)=F(W(t-1)X(t-1)) \\ &Y(t)=F(W^T(t-1)Y(t-1)) \\ \forall(i,j)\quad &W_{ij}(t)=W_{ij}(t-1)+K_i.(Y_i(t)-A_i(t)).X_j(t-1) \end{aligned} \qquad (13)$$

Etant entendu que les X_i et Y_i correspondant aux cellules d'entrée sont forcés par la forme d'entrée et que les Y_i correspondant aux cellules de sortie sont forcés par la sortie désirée. Il est utile d'insister sur le fait que *les états idéaux des cellules internes ne sont pas directement déterminés de l'exterieur* mais font l'objet d'une dynamique propre. La règle d'apprentissage est totalement locale dans le temps et l'espace, la modification d'une pondération ne faisant intervenir que des données locales aux automates qu'elle relie. Si le nombre de cellules internes est non nul, le critère <C> n'est en général pas convexe, il est de plus défini de manière statistique.

3 - SIMULATIONS

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 *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 couteuses. 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 arrangées en matrices 8x8, 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 3x3, 5x5 ou 7x7). 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_1 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 complexité des fonctions booléennes que le réseau peut synthétiser. En outre, sur ce type de donnée 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 six premières lettres de l'alphabet sont présentées (Fig.1.a), chacune dans quatre positions différentes (Fig.1.b), soit au total 120 formes différentes a grouper en 6 classes de 20 formes. A 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 extrèmement proches doivent être différenciées. Par exemple, la distance entre le troisième B et le deuxième E de la figure 1.a 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).

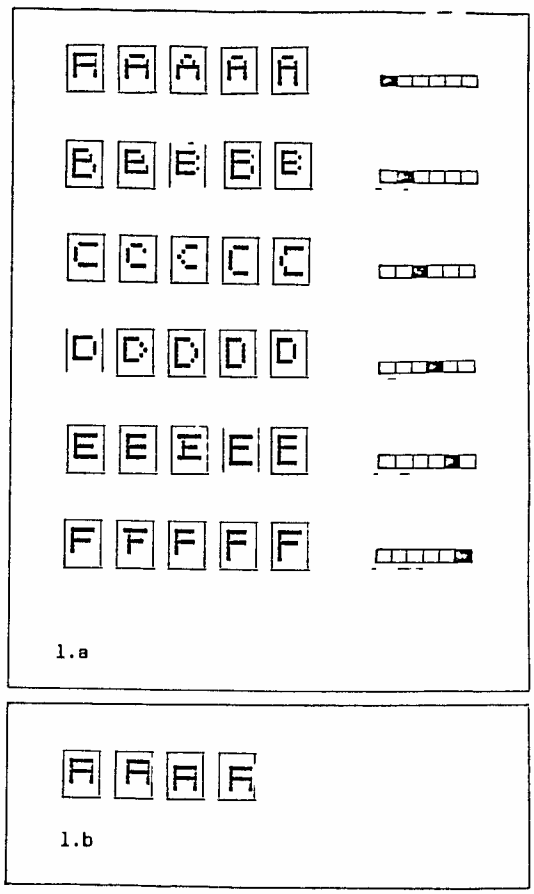

Fig 1 : (a)- Ensemble d'apprentissage avec les réponses associées (b)- Chaque forme est présentee dans 4 position différentes.

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 - Auto-apprentissage, associations Pavloviennes

Le type d'apprentissage décrit ci-après, plus proche des comportements cognitifs des animaux, a été testé sur un exemple simple. La session se divise en trois phases dont les deux dernières tentent de simuler le conditionnement Pavlovien classique. Au cours des trois phases, l'ensemble des cellules d'entrée du réseau est divisé en deux parties sur lesquelles seront projetées des formes de natures différentes. Sur l'une

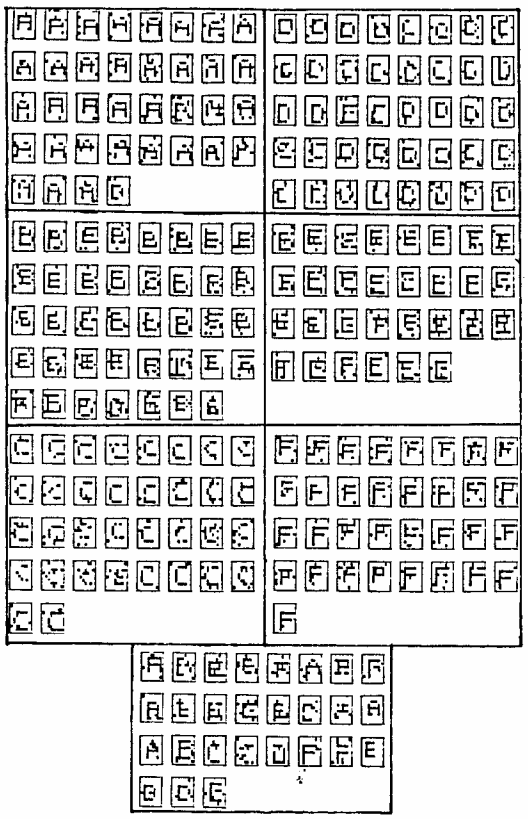

Fig 2 : Exemple de classement effectué par le réseau sur des images bruitées. L'ensemble représenté au bas de la figure regroupe les formes non classées.

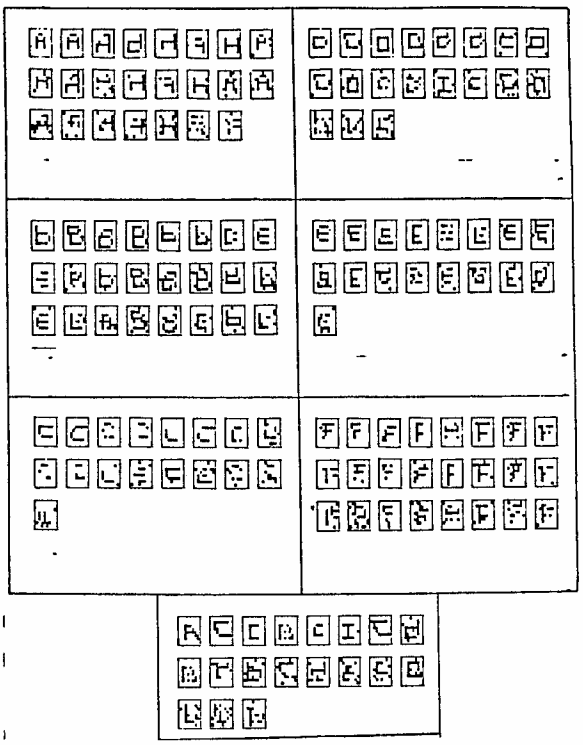

Fig 3 : Exemple de generalisation, classement opéré sur des formes non apprises.

d'elles, correspondant aux lignes d'ordre pair de la "rétine", seront projetés les *stimuli inconditionnels* (SI), et sur l'autre, correspondant aux lignes d'ordre impair, les *stimuli neutres et conditionnels* (SC) (dans la suite nous nommerons ces deux moitiés respectivement rétine I et rétine C).

Phase 1 : Cette phase n'a pas d'équivalent dans l'apprentissage pavlovien, elle consiste à créer dans le réseau l'apparition d'une *réaction inconditionnelle*, c'est à dire l'association naturelle entre un stimulus (ex. la perception de la nourriture), et une action (se diriger vers la nourriture). Dans cet exemple, nous employons deux stimuli (les caractères C et T) correspondant à la présence de nourriture à gauche ou à droite d'un labyrinthe en T, devant être associés à deux réponses (tourner à gauche ou à droite). Cet apprentissage se déroule comme les précédents, les réponses étant dictées de l'extérieur. A chaque itération on donne au réseau :

- Le stimulus inconditionnel (T ou C) sur la rétine I. Ces figures sont apprises en présence de bruit : 10 à 30 pour cent de leurs pixels choisis au hasard sont inversés à chaque présentation.
- La réaction inconditionnelle correspondante sur les cellules de sortie
- Un bruit (pixels aléatoirement noirs ou blancs) sur la rétine C correspondant à l'absence de stimulus conditionnel.

Une fois cet apprentissage achevé, l'association Pavlovienne proprement dite peut débuter.

Phase 2 : Cette phase constitue le conditionnement en *auto-apprentissage*. La réponse désirée n'est plus imposée de l'extérieur mais par la *réponse du réseau lui-même*, il est alors son propre superviseur. Le stimulus conditionnel est un carré ou un triangle. A chaque itération on donne au réseau un des couples (C bruité, carré) et (T bruité, triangle) sur les rétines I et C. Au terme d'une vingtaine de présentations la configuration de pondérations est gelée.

Phase 3 : On montre au réseau le triangle ou le carré sur la rétine C, et rien sur la rétine I (que des -1). Et l'on constate un transfert spontané de la réponse du SI vers le SC

Il est intéressant de remarquer que la présence de bruit sur les stimuli inconditionnels pendant les deux premières phases est absolument nécessaire à la persistance de la réaction après suppression des SI.

4 - CONCLUSION

Plusieurs tests additionnels dans des conditions

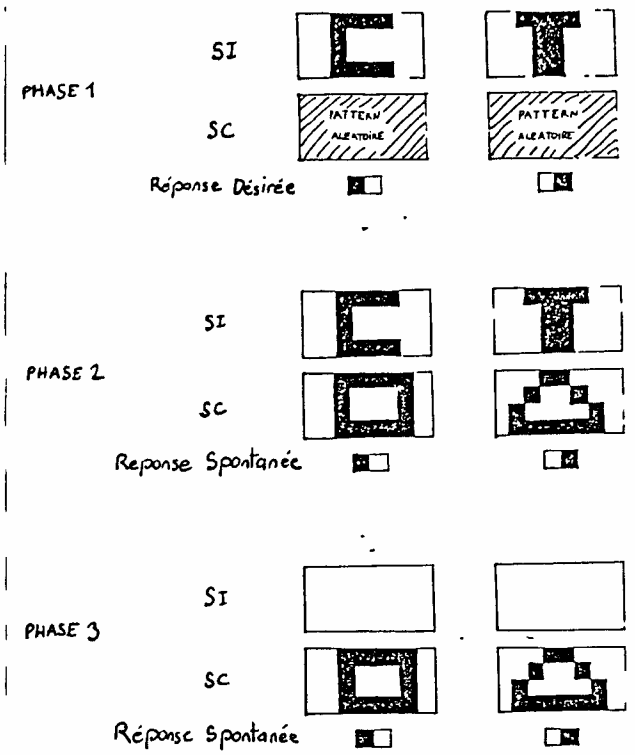

Fig 4 : Sequence utilisee pour l'auto-apprentissage.

d'apprentissage différentes doivent être menées. En particulier pour évaluer l'influence des divers paramètres sur la qualité et le temps de l'apprentissage (valeur de K, nombre de couches...) ainsi que pour tester le cas où le graphe de connexion comporte des circuits. Néammoins il semblerait 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 en général des description de dimensionnalité plus faible. Il semble que l'approche des réseaux cellulaires apportent une réponse partielle aux problèmes du parallélisme massif nécessaire à l'execution en temps réel des tâches de perception artificielle, elle offre en outre des architectures materielle d'une grande régularité ce qui est technologiquement précieux [10].

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

BIBLIOGRAPHIE

[1] AMARI S.I. : Learning patterns and patterns sequences by self-organizing net 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., RUHONEN 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