

IMPLEMENTATION PARALLELE DE RESEAU DE NEURONES ARTIFICIELS - APPLICATION A LA RECONNAISSANCE DE CARACTERES ARABES

MM. BELMEZITI T. MEHBALI F. GUETTACHE

Laboratoire architecture des systèmes
Institut d'électronique USTHB El Alia BP 32
Bab ezouar ALGER ALGERIE

RESUME

Le présent travail consiste à résoudre le problème de la reconnaissance de caractères manuscrits par les réseaux de neurones artificiels. L'objectif visé dans cette solution est la parallélisation de l'algorithme sur un réseau de processeurs dans le but d'accélérer en temps d'exécution la phase d'apprentissage.

ABSTRACT

In this paper we study how to resolve the handwritten character recognition problem by artificial neural network. The aim is to parallel algorithm over processors network so that the learning step will be accelerated in processing time.

1 - INTRODUCTION

L'algorithme de rétropropagation du gradient [2] est actuellement reconnu comme étant un des plus performants algorithmes d'apprentissage dans les réseaux à plusieurs couches. Dans cet article nous nous sommes appliqués à utiliser cet algorithme pour résoudre le problème de la reconnaissance de caractères manuscrits soumis à certaines contraintes. Cet algorithme converge assez lentement ce qui est coûteux en temps de calcul. Ce coût devient rapidement prohibitif lorsque

- l'alphabet est riche en caractères ; c'est le cas de l'alphabet arabe
- la base d'exemples (dictionnaire des prototypes) est très importante (en prévoyant plusieurs prototypes pour une même lettre de l'alphabet)
- la résolution de l'image caractère est grande
- la technique de segmentation est assez élaborée
- les caractères à reconnaître ne sont soumis à aucune contrainte
- lorsque l'on fixe un taux d'erreur très réduit

L'ensemble de ces facteurs améliorent de façon considérable le taux de

reconnaissance mais parallèlement ils ralentissent lourdement la phase d'apprentissage. En contrepartie, une fois la détermination des coefficients représentant les poids des connexions terminée, la phase de reconnaissance ou d'exploitation est très rapide. Une solution envisagée pour accélérer la phase d'apprentissage peut être l'implantation sur une architecture parallèle. Plusieurs approches existent parmi lesquelles on peut citer

- l'approche par la mise au point d'un circuit VLSI réalisant la fonction d'une cellule ou d'une ou plusieurs couches. La aussi l'accroissement du nombre de cellules ou de couches à intégrer peut poser des problèmes au niveau de l'interconnexion

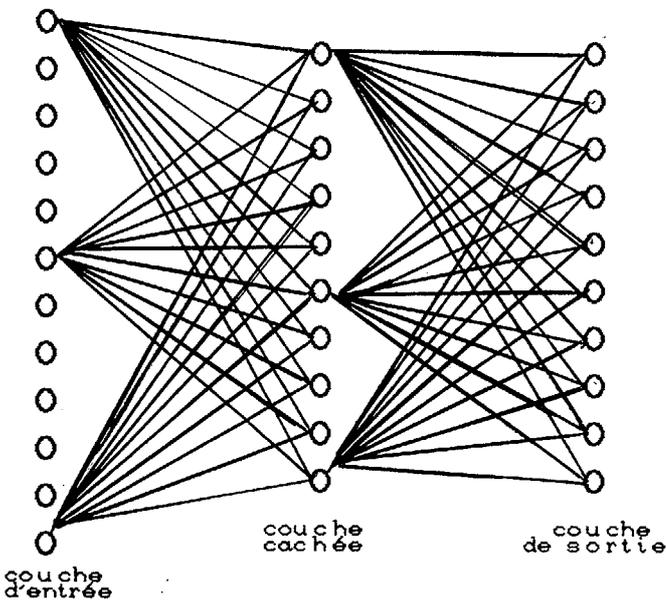
- la parallélisation des opérations sur les matrices

- l'approche par les architectures du type MIMD où les solutions sont multiples. On peut citer à titre d'exemple la distribution de la base d'exemples et du réseau intégral sur plusieurs processeurs, la distribution par ligne et la distribution par colonne. Notre choix s'est fixé sur une structure du type MIMD : c'est un réseau de transputers où l'implantation présente quelques particularités et contraintes



2 - ALGORITHME D'APPRENTISSAGE ET RESEAU MULTICOUCHES

Le réseau utilisé est un réseau multicouches comprenant une couche d'entrée, une couche cachée et une couche de sortie. Les différentes couches sont totalement connectées. La couche d'entrée comprend un nombre de cellules égal à la dimension du vecteur de codification du caractère squelettisé, chaque cellule de la couche cachée et de la couche de sortie est spécialisée dans la reconnaissance d'une forme particulière, dans notre cas un caractère alphabétique. Cette spécialisation se traduit après l'apprentissage par une quasi-égalité entre, d'une part le vecteur des poids pré-synaptiques d'une cellule et l'entrée qu'elle mémorise et, d'autre part, l'association désirée. A la présentation d'une entrée, chaque cellule cachée calcule alors le produit scalaire entre le vecteur à reconnaître et la forme qu'elle mémorise, si le résultat est supérieur au seuil de la cellule, celle-ci est activée et délivre la sortie souhaitée.



L'algorithme d'apprentissage comprend trois phases : une phase avant permettant la propagation en avant de l'état des cellules la seconde phase consiste en la propagation arrière de l'erreur et la troisième et dernière consistera en l'ajustement des poids des connexions.

Les équations correspondants aux différentes phases de l'algorithmes sont :

Les équations correspondant à la

phase de propagation des états sont exprimées par les relations ci-dessous :

$$h_j = 1 / 1 + e^{-(\sum_{i=0}^A w_{1i,j} * x_i)}$$

$$j = 1, \dots, b$$

$$O_j = 1 / 1 + e^{-(\sum_{i=0}^B w_{2i,j} * h_i)}$$

$$j = 1, \dots, c$$

Concernant la seconde phase, les gradients sont calculés à partir des expressions :

$$\epsilon_{1j} = h_j (1 - h_j) \sum_{i=1}^c \alpha^{(i)} * \epsilon_{2i} * w_{2ji}$$

$$j = 1, \dots, b$$

$$\epsilon_{2j} = O_j * (1 - O_j) * (Y_j - O_j)$$

$$j = 1, \dots, c$$

La troisième phase est régie par les équations ci-dessous qui permettent la modification des poids des connexions

$$w_{1i,j} = n * \epsilon_{1j} * x_i + w_{1ij}$$

$$i = 0, \dots, a, j = 1, \dots, b, n = 0,35$$

$$w_{2i,j} = n * \epsilon_{2j} * h_i + w_{2ij}$$

$$i = 0, \dots, b, j = 1, \dots, c, n = 0,35$$

3 - DISTRIBUTION DU RNA SUR LE RESEAU DE TRANSPUTERS

Il s'agit donc de projeter le réseau de neurones artificiels sur un réseau de transputers. Les cellules seront distribuées couches par couches et chaque processeur contiendra un certain nombre de lignes des cellules des différentes couches du réseau. Soit P le nombre de transputers que contiendra l'architecture du réseau sur laquelle on désire projeter le réseau de neurones. Le nombre de cellules que possède chaque transputer est calculé par la division du nombre total de cellule N_L de la couche L par le nombre P. Ainsi chaque processeur aura soit m_L = N_L / P soit m_L = (N_L / P) + 1

$$V_{P_i, P_{i+1}} = (e_{i1}, \dots, e_{in})$$

Au niveau de la passe arrière l'ensemble des composantes partielles du gradient calculées dans chaque transputer sont également regroupées en un vecteur destiné à la diffusion

$$G_{T_p} = (Y_{i1}, \dots, Y_{im})$$

L'algorithme de communication peut être résumé de la manière suivante :

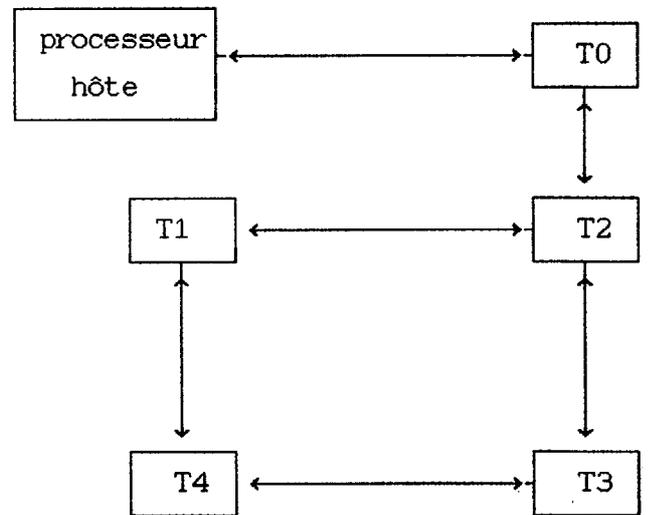
```

debut
  pour k=1 jusqu'à p - 1
    faire
      recevoir  $V_{P_{i-1}, P_i}$  de  $T_{i-1}$ 
       $V_{P_i, P_{i+1}} \leftarrow V_{P_i, P_{i+1}} + V_{P_{i-1}, P_i}$ 
      envoyer  $V_{P_i, P_{i+1}}$  de  $T_{i+1}$ 
    fait
  fin
  
```

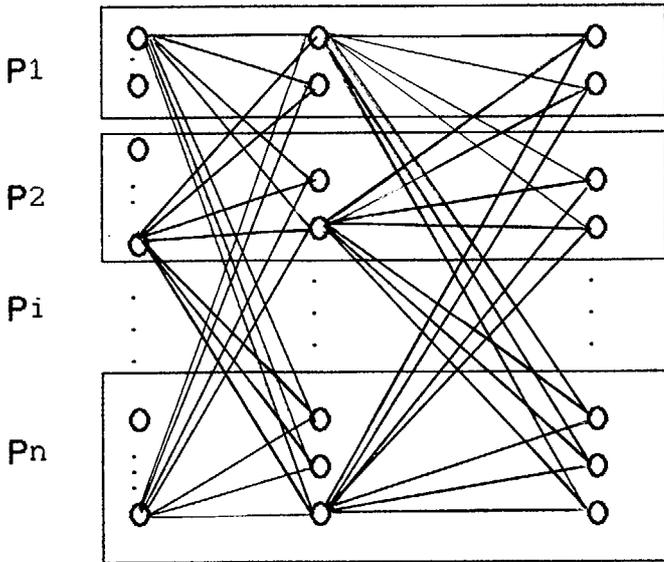
Pour la diffusion il suffit de remplacer le vecteur $V_{P_i, P_{i+1}}$ par $G_{T_{P_i, P_i}}$ le problème de l'interblocage est résolu en donnant la priorité à un transputer qui va reveiller les autres

5 - IMPLANTATION

Le système réel se compose d'une machine hôte (PC) d'un transputer racine et du réseau en anneau



Le caractère acquis subit un prétraitement préalable qui consiste en un lissage permettant d'éliminer les bruits parasites , suivi d'une traitement permettant d'aboutir à la squelettisation du caractère , on transforme ensuite le squelette sous forme d'un vecteur binaire. Cette technique est basée sur la division du tracé du caractère en régions auxquelles on associe des matrices qui



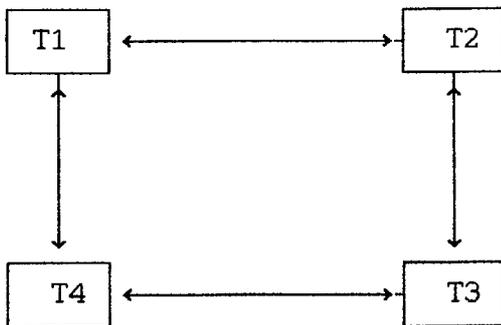
4 - COMMUNICATION ET INTERCONNEXION

Les communications sont celles qui sont liées à :

- la multiaccumulation dans la passe avant
- la diffusion du gradient dans la passe arrière

Notre choix s'est fixé pour une topologie en anneau qui est en soit une topologie simple Ce choix est liée à deux facteurs majeurs :

- le nombre de liens est limité sur le transputer utilisé
- l'handicap des communications bloquantes du langage utilisé le C parallèle qui sont du type rendez-vous



Au niveau de la passe avant chaque transputer (P) doit d'abord calculer toutes les entrées partielles issues des cellules distribuées dans ce même transputer , et ensuite les regrouper en un vecteur (paquets de données) qui sera propagé



seront par la suite parcourues pour déterminer l'existence ou non d'une portion du squelette dans chacune de ces matrices

Au niveau du processeur hôte est implanté un environnement permettant de dérouler un menu .Les algorithmes de lissage, de squelettisation de normalisation et de codification sont implantés au niveau du transputer racine .les transputers de l'anneau contiennent les algorithmes d'apprentissage et de reconnaissance

6 - RESULTATS

Le réseau pour un vecteur d'entrée de dimension 49 et appliquée pour la reconnaissance de 10 caractères avec une base d'exemples composée de 50 prototypes donné a les résultats suivants sur les neurones de sorties

	س	ص	م	د	أ
00	0,88	0,18	0,22	0,18	0,22
01	0,08	0,89	0,10	0,10	0,10
02	0,12	0,08	0,89	0,09	0,08
03	0,15	0,14	0,12	0,79	0,16
04	0,16	0,20	0,27	0,13	0,81
05	0,13	0,10	0,05	0,27	0,00
06	0,08	0,10	0,00	0,10	0,10
07	0,11	0,10	0,10	0,08	0,05
08	0,00	0,00	0,00	0,00	0,00
09	0,06	0,05	0,10	0,00	0,01
	ق	ع	ح	ر	و
00	0,22	0,16	0,24	0,18	0,15
01	0,08	0,10	0,00	0,07	0,07
02	0,06	0,00	0,08	0,05	0,05
03	0,13	0,08	0,07	0,00	0,00
04	0,00	0,06	0,06	0,00	0,16
05	0,77	0,11	0,11	0,24	0,00
06	0,13	0,89	0,10	0,15	0,15
07	0,10	0,07	0,88	0,13	0,12
08	0,09	0,00	0,01	0,78	0,14
09	0,00	0,01	0,01	0,14	0,89

le volume de communication est largement inférieur au volume de calcul .Ceci se matérialise par une accélération comprise entre 3 et 3,5

CONCLUSION

Les réseaux de neurones artificiels peuvent être considérés comme une solution très satisfaisante pour la classification de caractères et une alternance sérieuse aux méthodes classiques quelles soient syntaxiques ou statistiques . Celles-ci

présentent des inconvénients majeurs tels que réaliser des grammaires reconnaissables (l'approche syntaxique) et la lourdeur en temps d'exécution (les approches statistiques).La parallélisation de la phase apprentissage accélère la convergence de l'algorithme bien que le speed up n'est pas linéaire pour notre cas .La phase de reconnaissance utilisant les résultats de la phase d'apprentissage est en général très rapide ,la parallélisation de cette phase peut être envisagée ; elle permet une amélioration certaine et supplémentaire des performances

BIBLIOGRAPHIE

- [1] MATOUGHI "reconnaissance de caractères arabes manuscrits" thèse de magister HCR Alger 1990
- [2] D.RUMELHART,G.HINTON, et R.WILLIAM "learning internal representation by error propagation ,MIT press 82
- [3] R.P. LIPPMAN "an introduction to computing with Neural Nets " IEEE ASSP magazine 87
- [4] T.BEYNON et N.NODD "the implémentation of Multi-Layer Perceptrons on transputer Networks " Proceedings of the 7th Intern.workshop on Parallel Processing of transputer Based Machines Grenoble 87
- [5] S.WANG et F.ROBERT "implantation de l'algorithme de retro-propagation du gradient sur une machine hypercube" TSI 91
- [6] A.GORAINÉ ,M.USHER et S.AL_ALAMI "off line Arabic character recognition " COMPUTER 92
- [7] H.PAUGAM-MOISY "parallélisation de réseaux de neurones artificiels sur réseau de transputers " La lettre du transputer recherche et applications juin 1992
- [8] G.BOUVIER et A.CHEHIKIAN "Segmentation des caractères après filtrage spatial par réseau de neurones artificiels " AFCET 84 tome 1
- [9] K.KNIGHT "Connectionist ideas and algorithms" Communications of the ACM Novembre 1990 - Vol 33 N°11
- [10] R.D.GELLER et D.WHAMMERSTROM "A VLSI architecture for a Neurocomputer using Higher-Order Predicates" conférence IEEE 1987
- [11] A.GUERIN et J.HERAULT "CRASY une architecture de calcul reconfigurable pour la simulation de réseaux neuromimétiques" revue TRAITEMENT DU SIGNAL VOL 5 -N°3 1988
- [12] P.MARTIN "Reconnaissance de partitions musicales et réseaux de neurones" AFCET 89