

Un réseau de neurones pour des associations multimodales

A neural network for multimodal association

A. Reghis¹ F. Alexandre¹ Y. Boniface¹

¹CORTEX LORIA INRIA-Lorraine, Nancy, France

ReghisAbdelmalek@yahoo.com, Frederic.Alexandre@loria.fr, Yann.Boniface@loria.fr.

Résumé

Les recherches en neuromimétisme (réseaux de neurones) s'inspirent des neurosciences pour proposer des modèles cérébraux de comportements cognitifs typiquement humains, visant à s'approcher de certains phénomènes corticaux, pour développer de nouveaux paradigmes de calculs. L'intérêt du présent travail est de s'approcher du phénomène de l'intégration multisensorielle. Nous avons mené une étude sur un modèle de BAM¹ [7] adapté proposé par E. Reynaud [9] pour modéliser l'intégration multisensorielle. Nous améliorons les performances et la robustesse du modèle de la BAM en modifiant les algorithmes d'apprentissage et de rappel. Nous avons proposé un nouvel algorithme d'apprentissage en ligne pour la BAM, ainsi qu'une nouvelle approche pour la modélisation d'une mémoire multiassociative, en se basant sur la synchronisation de la convergence et de la propagation des flux d'informations. Cet article résume ces travaux et présente l'expérimentation et l'évaluation des performances de la BAM en terme d'association multimodale.

Mots Clef

Réseau de neurones, mémoire associative multimodale.

Abstract

Research in connectionism (neural networks) takes neurosciences as a starting point in order to propose cerebral models of typically human cognitive behaviors, aiming at approaching certain cortical phenomena and to develop new paradigms of calculus. The aim of this paper is to model multi-sensory integration phenomena. We undertook a study on a model of adapted BAM¹ [7,9] by E. Reynaud to model multi-sensory integration. We have improved the performances and the robustness of the BAM model modifying the learning and recall stages in the algorithm. We proposed a new algorithm of on-line learning for the BAM, and a new approach for modeling a multiassociative memory based on the synchronization of the convergence and the propagation of flows of information. This article summarizes this work and presents experimentation and performance

evaluations of the BAM in a multimodal association perspective.

Keywords

Neural networks, Multimodal associative memory.

1 Introduction

Le neuromimétisme, ou connexionnisme, est l'étude des réseaux de neurones artificiels et prend ses racines dans les travaux fondateurs de McCulloch et Pitts, Hebb et Rosenblatt [3]. Le but de cette approche était à l'origine de s'inspirer des mécanismes cérébraux afin de concevoir de nouveaux paradigmes de calculs. Les travaux décrits dans ce papier concernent une approche du phénomène de l'intégration multisensorielle défini dans les travaux neurophysiologiques en imagerie fonctionnelle du cerveau. Les résultats des études électrophysiologiques montrent que l'interaction multisensorielle est rendue possible par la convergence des aires sensorielles dans un site commun multimodal [1]. Ces études ont proposé un schéma général (Fig.1) dans lequel l'interaction entre les zones unimodales et la zone de convergence multimodale serait à la base de l'intégration multisensorielle [9].

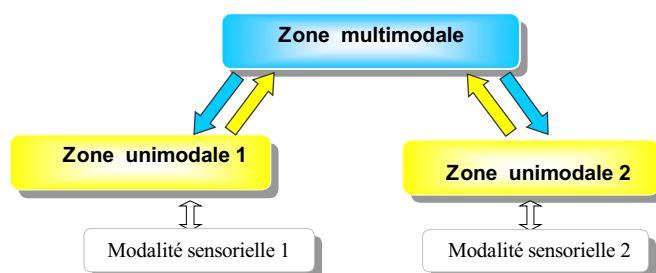


Fig.1 – Schéma proposé par les neurophysiologistes pour l'intégration multisensorielle.

2 La mémoire associative

La mémoire est un processus de stockage et d'exploitation d'une connaissance antérieurement acquise, ce processus s'opérant sur la base d'une modification des propriétés d'un support physique. On peut avoir deux techniques :

- l'approche informatique : un accès par adresse, séquentiel et localiste avec une représentation statique.

¹ BAM : Bidirectional Associative Memory

- l'approche connexionniste : un accès par le contenu, parallèle et distribuée avec une représentation dynamique.

La mémoire associative est un système qui stocke la façon de faire correspondre une sortie spécifique et une entrée spécifique, un système qui associe deux motifs de façon que lorsque l'entrée est présentée, la sortie soit rappelée. Une mémoire associative qui produit des sorties semblables aux entrées est appelée mémoire auto-associative (i.e. association des motifs à eux-mêmes) alors qu'une mémoire qui produit des sorties différentes des entrées est appelée mémoire hétéro-associative (i.e. association de motifs à d'autres motifs).

Les premiers travaux cherchant à modéliser la mémoire associative se sont essentiellement intéressés à son aspect autoassociatif (les modèles de Hopfield et de Kohonen) [3]. Par contre, il n'existe que très peu de modèles dits hétéro-associatifs. Nos travaux sont issus du modèle original présenté par B. Kosko [6][7].

2.1 La BAM de Kosko

La BAM de Kosko [7] est une mémoire hétéroassociative. Ce modèle généralise le modèle de Hopfield en associant des motifs à d'autres motifs différents.

La BAM est composée de deux couches de neurones de tailles différentes, totalement interconnectées, de façon bidirectionnelle (Fig.2), dans lesquelles l'information est réverbérée entre les deux couches de neurones, jusqu'à ce qu'un état d'équilibre soit atteint.

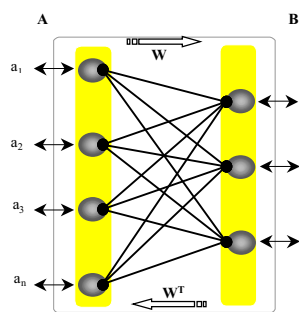


Fig.2 BAM de Kosko

Les poids de connexions de ce modèle sont déterminés selon la règle de Hebb [3]. Une des principales limites de la BAM est sa faible capacité de stockage, due à la règle d'apprentissage et à la symétrie des connexions. Wasseman [13] montre que la capacité de stockage de la BAM, lorsque N est la taille de la plus petite des couches, est de $\frac{N}{2 \log_2 N}$. Un autre problème de la BAM est l'interférence catastrophique, une conséquence de la mémorisation des exemples à peu près orthogonaux. Pour faire face à ces limites, plusieurs méthodes sont apparues qui cherchent à augmenter cette capacité de stockage. Ces méthodes explorent deux piste principales : des prolongements de l'architecture [4][5][10][14], et une adaptation des poids avec d'autres algorithmes d'apprentissage [8][11][10][13][15]. Parmi les solutions de type algorithmique, nous nous intéressons plus particulièrement à la méthode de pseudo-relaxation PRLAB (Pseudo Relaxation Learning Algorithme for BAM) de Oh et Kothari [8].

2.2 L'algorithme d'apprentissage PRLAB

PRLAB, de Oh et Kothari, est un algorithme itératif qui converge en un nombre fini d'étapes. Cette méthode est

une variation de la méthode de relaxation, basée sur une technique mathématique de résolution de systèmes d'inéquations linéaires. Selon Oh et Kothari [8], PRLAB propose plusieurs avantages, une exploitation de la capacité maximum de stockage de BAM, une garantie de rappel parfait jusqu'à N paires apprises, pour une BAM contenant N neurones dans chaque couche, et une mémorisation des exemples « à peu près » orthogonaux qui ne pose plus de problème. De plus PRLAB est stable et converge rapidement.

2.3 La BAM triple d'E. Reynaud

Pour modéliser l'intégration multisensorielle E. Reynaud [9] propose une adaptation de la BAM de Kosko. Dans ce modèle, une couche associative (Z) relie les deux couches d'entrées perceptives (X et Y). Les connexions entre la couche associative et les couches perceptives sont bidirectionnelles et asymétriques (Fig.3).

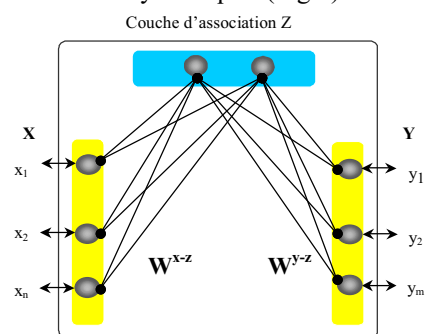


Fig.3 La BAM triple d'E. Reynaud, X et Y sont les paires à associer.

L'apprentissage est basé sur PRLAB. Le rappel des paires associées se fait par l'activation des couches perceptives. Cette triple BAM sert à stocker des associations de patterns, mais d'une manière différente de celle du modèle classique. Les paires qui vont nous intéresser sont les deux patterns encodés dans les couches perceptives, le pattern de la couche d'association pouvant être considéré comme un « code de liage ».

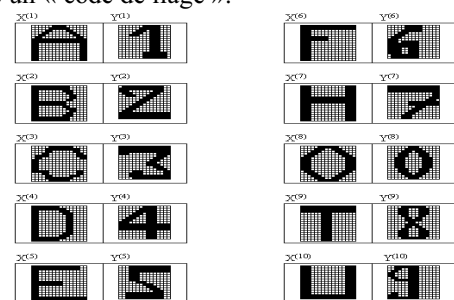


Fig.4 La base de 10 paires à apprendre dans la BAM triple.

Pour évaluer les propriétés et les capacités d'apprentissage et de généralisation de la BAM d'E. Reynaud, nous avons choisi de faire nos expériences avec la base de données utilisée par E. Reynaud [9]. La base est constituée des images de lettres et des images de chiffres (Fig.4) : Les lettres sont encodées par la couche X sur 256 neurones, les chiffres par la couche Y sur 225 neurones. Le pattern présenté à la couche Z est un code aléatoire discriminant, encodé sur 150 neurones.

2.4 Problématique

Après expérimentation, nous avons isolé des problèmes qui influent sur les performances et de robustesse de la BAM triple d'E. Reynaud:

1. Une influence de la méthode d'initialisation de la couche d'association sur la robustesse de la BAM triple au bruit.
 2. La dépendance de l'algorithme d'apprentissage à l'ordre de tirage des patterns.
- De plus, la BAM d'E. Reynaud à deux limites principales :
3. L'apprentissage est par paquet (Off-line), il n'offre pas la possibilité d'associer dynamiquement des nouveaux patterns.
 4. La faible capacité de simuler et de rappeler les associations apprises en cas d'absence de modalités.

L'objectif de notre travail est de trouver des solutions à ces problèmes afin de s'approcher du phénomène de l'intégration multisensorielle en tenant compte du caractère distribué des calculs, ainsi que des performances et de la robustesse de la mémoire.

Nous proposons dans la section 3 des solutions aux problèmes 1 et 2. Dans les sections 4 et 5, nous présentons nos contributions originales pour résoudre les limites principales de la BAM d'E. Reynaud (3 et 4).

3 Le modèle de la BAM multiassociative

3.1 Initialisation de la couche d'association par zones d'affectation

Pour converger, l'algorithme d'apprentissage de la BAM triple nécessite une initialisation discriminante du code de liage. De plus, nos expériences ont montré une sensibilité de ce modèle à l'initialisation de ce code de liage :

- Une initialisation aléatoire et discriminante offre une sensibilité au bruit,
- Une initialisation par compression des entrées perceptive donne de meilleurs résultats mais ne garantie pas la discrimination, et donc pas la convergence.

La solution que nous préconisons pour garantir la convergence de l'apprentissage et la robustesse au bruit est d'initialiser la couche d'association par zones d'affectations (Fig.5).

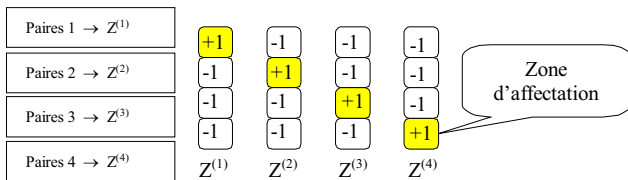


Fig.5 Pour une BAM triple avec une couche d'association de 4 neurones, il est possible d'associer 4 paires via un code de liage, en affectant une zone de 1 neurone pour chaque paire, ici le code de liage de la première paire est [+1,-1,-1,-1] et pour la paire suivante est [-1,+1,-1,-1].

Pour une couche d'association de taille N neurones et pour une zone de taille 1 neurone, il est possible d'affecter N codes de liages pour les N paires à associer. L'intérêt de cette technique est d'assurer une distance de Hamming entre les N codes de liage supérieure ou égale

à deux fois la taille de la zone d'affectation. Donc elle garantit la convergence de l'algorithme d'apprentissage. Nous verrons dans la suite que cette méthode a aussi un intérêt concernant l'apprentissage en ligne. En revanche, cette méthode ne donne plus l'accès aux propriétés du codage distribué.

3.2 Architecture et algorithmes

Le modèle de la BAM multiassociative est une adaptation de la BAM triple proposée par E. Reynaud :

- Initialisation des codes de liage par zones d'affectations,
- La formule d'adaptation de la couche d'association sera généralisée pour plusieurs couches perceptives d'entrées,
- Tirage aléatoire des patterns à apprendre à chaque époque d'apprentissage.

La BAM multiassociative est composée de deux types de couches (Fig.6) :

- Des couches perceptives unimodales ($U_1, \dots, U_N$).
- Une couche d'association multimodale (M).

L'algorithme d'apprentissage est basé sur PRLAB, adapté pour plusieurs couches perceptives et une seule couche d'association.

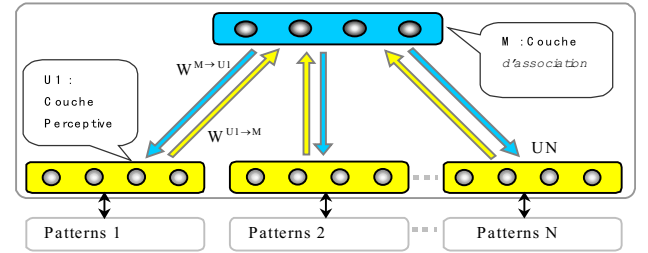


Fig.6 - BAM multiassociative

Nous présentons les algorithmes d'apprentissage et de rappel de la BAM multiassociative :

Algo.1 - Algorithme d'apprentissage de la BAM

1. Initialiser arbitrairement entre -1 et $+1$
 - les poids $W^{U_i \rightarrow M}$, $W^{M \rightarrow U_i}$ et les seuils $\theta_{U_1}, \dots, \theta_{U_N}$, θ_{M_k} .
2. Pour chaque n-uplets ($U_1^{(p)}, \dots, U_N^{(p)}$) à apprendre faire
 - Générer un code de liage $M^{(p)}$ en affectant à une zone,
3. Pour tout ($U_1^{(p)}, \dots, U_N^{(p)}, M^{(p)}$) faire
 - /* l'ordre des n-uplets sera changé aléatoirement dans chaque époque */

- Pour tous les neurones des couches perceptives faire
- Si $S_{U_n}^{(p)} \cdot U_{n_k}^{(p)} \leq 0$ Alors mettre à jour poids et seuils selon

$$\Delta w_{ki}^{M \rightarrow U_n} = - \frac{\lambda}{1 + \dim M} \cdot (S_{U_n}^{(p)} - \xi \cdot U_{n_k}^{(p)}) M_i^{(p)}$$

$$\Delta \theta_{M_k} = \frac{\lambda}{1 + \dim M} \cdot (S_{U_n}^{(p)} - \xi \cdot U_{n_k}^{(p)})$$

- Pour tous les neurones de la couche d'association M faire
- Si $S_M^{(p)} \cdot M_k^{(p)} \leq 0$ Alors mettre à jour poids et seuils selon :

$$\Delta w_{ij}^{M \rightarrow U_n} = - \frac{\lambda}{1 + \dim U_1 + \dots + \dim U_n + \dots + \dim U_N} \cdot (S_M^{(p)} - \xi \cdot M_k^{(p)}) U_{n_j}^{(p)}$$

$$\Delta \theta_{M_k} = \frac{\lambda}{1 + \dim U_1 + \dots + \dim U_n + \dots + \dim U_N} \cdot (S_M^{(p)} - \xi \cdot M_k^{(p)})$$

4. Répéter la deuxième étape jusqu'à stabilisation du réseau (pas de mise à jour).

Avec :

$$S_{M_k}^{(P)} = \sum_{i=1}^{\dim U1} w_{ik}^{U1 \rightarrow M} . U1_i^{(P)} + \dots + \sum_{j=1}^{\dim UN} w_{jk}^{UN \rightarrow M} . UN_j^{(P)} - \theta_{M_k}$$

$$S_{Un_i}^{(P)} = \sum_{k=1}^{\dim M} w_{ki}^{M \rightarrow Un} M_k^{(P)} - \theta_{Un_i}$$

Algo.2 - Algorithme de Rappel de la BAM

1. Présenter un n-uplet ($U1^{(P)}, U2^{(P)}, \dots, UN^{(P)}$)
2. faire
 - Calculer la sortie de la couche d'association $M^{(P')}$
 - Calculer la sortie des couches perceptives ($U1^{(P')}, \dots, UN^{(P')}$)
 - Jusqu'à stabilisation du réseau ($U1^{(P)} = U1^{(P')}, \dots, UN^{(P)} = UN^{(P')}$ et $M^{(P)} = M^{(P')}$)
4. La sortie des couches perceptives ($U1^{(P')}, \dots, UN^{(P')}$) correspond à la réponse de la BAM qui sera soit une donnée apprise soit un état poubelle.

3.3 Evaluation de l'initialisation par zones d'affectations

Pour mesurer l'influence de l'initialisation de la couche d'association par zones d'affectations sur le rappel des paires apprises depuis une version bruitée sur une seule modalité, nous avons fait nos expériences sur une BAM triple avec deux couches perceptives : X de 256 neurones, Y de 225 neurones et une couche d'association Z de 150 neurones. D'après les résultats présentés la figure 7, le codage par zones d'affectations, garantit à la fois la convergence de l'apprentissage (ce qui ne garantit pas le codage par compression) et une robustesse au bruit satisfaisante (moins satisfaisante que le codage par compression).

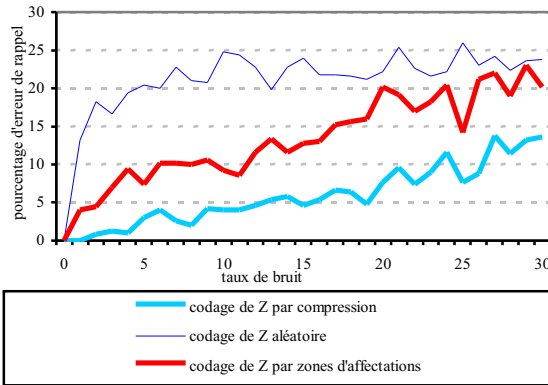


Fig.7 - Taux d'erreur de rappel des paires en fonction du bruit sur X et de la technique de codage de Z.

3.4 Evaluation du tirage aléatoire des exemples

Pour que notre étude ne soit pas liée à une base de données particulière, nous avons utilisé une base de données constituée de 50 paires composées de -1 et de +1 répartis aléatoirement. Chaque paire est codée sur 50 neurones, donc la BAM multiassociative est composée de deux couches perceptives X et Y de 50 neurones et d'une couche d'association Z de 50 neurones. Pour évaluer les capacités d'apprentissage et de rappel de la BAM triple,

nous avons fait apprendre un nombre croissant de patterns aléatoires à notre modèle (de 10 à 50 patterns), en mesurant le taux d'erreur de rappel depuis des versions complètes suivant le mode de tirage.

Nous avons utilisé, dans la suite, le codage par zones d'affectations de la couche d'association.

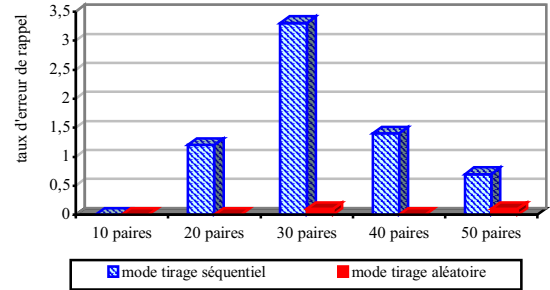


Fig.8 - Pourcentage d'erreur commis pour le rappel des paires apprises en fonction de la méthode de tirage des exemples dans l'apprentissage, pour différentes capacités de stockage.

Sur les 100 expériences, les résultats présentés dans la figure 8 montrent que la méthode de tirage des exemples aléatoirement dans chaque époque d'apprentissage améliore les performances de la BAM triple (le taux d'erreur de rappel reste inférieur à 0.5%) par rapport au mode de tirage séquentiel.

3.5 Evaluation de la résistance au bruit

Une des propriétés recherchées par les mémoires associatives est la capacité de rappel depuis des données bruitées. Nous mesurons le taux d'erreur de rappel lorsque l'un ou les deux membres de la paire présentée aux couches perceptives contiennent du bruit et nous mesurons l'effet du bruit sur la couche X seulement. Les résultats des 100 expériences réalisées sont présentés dans la figure 9.

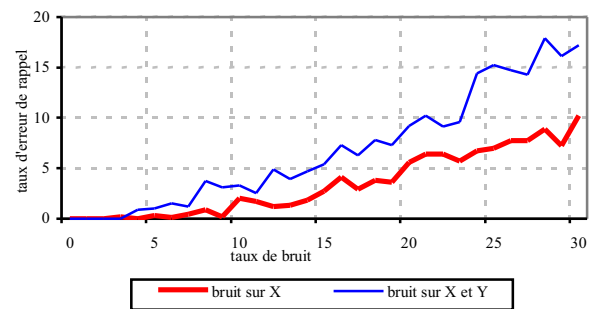


Fig.9 - Pourcentage d'erreur commis pour le rappel des paires (version bruitée) en fonction du bruit sur X.

La figure 9 montre que notre modèle résiste significativement au bruit. Le taux d'erreur de rappel de la paire est inférieur à 10 % jusqu'à un niveau de bruit de 30% sur les patterns X.

3.6 Rappel avec des données manquantes

Pour évaluer les performances d'une mémoire associative multimodale, il faut tester sa capacité à rappeler, à partir d'un seul des membres d'une paire, le reste de l'association. Si on représentons le pattern manquant par

un pattern blanc (Fig.10), le taux d'erreur de rappel est supérieur à 11.00%. Néanmoins, et nous y reviendrons en section 5, cette représentation pose un problème, la couche d'association prenant en compte tout ce qui lui arrive des couches perceptives, l'absence de stimuli est considéré comme une perception.

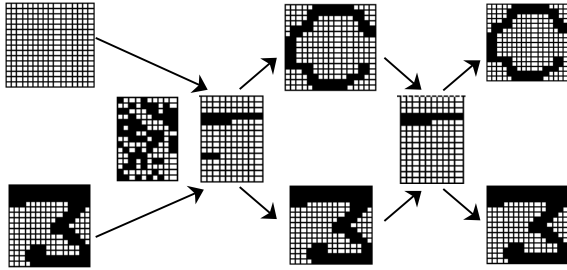


Fig.10 - Processus de rappel de la BAM de la paire (C, 3) depuis une version manquante, le pattern X est remplacé par un pattern blanc.

3.7 Synthèse sur la BAM multiassociative

Les modifications que nous avons faites au niveau de l'algorithme d'apprentissage (l'initialisation de la couche d'association, la formule d'adaptation de la couche d'association et la méthode du tirage des paires) ont apporté une réelle amélioration des performances de la BAM multiassociative en terme de robustesse au bruit et de capacité de rappel. Cependant, le phénomène de l'association multimodale dans le monde réel n'est pas par paquet (off-line), mais continu (en ligne). Conceptuellement la BAM multiassociative ne permet pas d'apprendre des nouveaux exemples dynamiquement. Ce problème est dû à PRLAB qui effectue son apprentissage de manière statique, l'algorithme a besoin de connaître l'intégralité de l'ensemble des associations à apprendre pour débiter son apprentissage.

4 La BAM multiassociative continue

Pour approcher la réalité et les contraintes techniques (la robotique par exemple), nos algorithmes d'apprentissage doivent prendre en compte les nouvelles informations quand elles sont perçues (apprentissage en ligne). Un des principaux problèmes de l'apprentissage continu est la difficulté de trouver un bon compromis entre adaptation et stabilité, ce qui peut amener le réseau à oublier les données apprises (trop d'adaptation) ou bien être incapable de s'adapter aux nouvelles données qui arrivent (trop de stabilité).

Conceptuellement PRLAB n'offre pas la possibilité d'un apprentissage continu. Pour déterminer les poids et les seuils du réseau, PRLAB exige la présentation simultanée de tous les patterns d'entrées et de sorties du réseau (apprentissage par paquets). Pour parvenir à un apprentissage continu, il faut résoudre le problème suivant : comment apprendre à la BAM triple les exemples les uns après les autres sans présenter tous les exemples par paquets simultanément, et sans perdre les données apprises.

Nous proposons, à chaque présentation d'un nouveau pattern, de chercher à estimer les états stables du réseau,

puis de présenter le nouveau pattern à apprendre avec l'ensemble des états stables trouvés avec PRLAB. La recherche des états stables se fera par la propagation inverse, c'est à dire après activation des zones dans la couche d'association.

Après l'initialisation des poids et les seuils du réseau, l'apprentissage stocke les patterns dans un espace de solutions d'un système d'inéquations linéaires. La méthode proposée consiste à inverser ce système d'inéquations linéaires afin de trouver les n états stables du réseau.

A chaque présentation d'un exemple ($X^{(A)}$, $Y^{(A)}$), si la BAM relaxe sur le même exemple alors l'exemple est appris. Dans le cas contraire on active les zones d'affectation les unes après les autres, l'information va être propagée vers les couches perceptives et retournée à la couche d'association. Si la BAM relaxe sur le même code de liage, on a estimé un pattern appris. A la fin de ce processus nous obtenons un ensemble d'états stables qui sont des estimations des patterns appris, une sorte de révision des informations mémorisées. On présente alors l'ensemble de ces états stables, complété par le nouveau pattern à apprendre, à l'algorithme d'apprentissage (Algo.1).

Estimer l'ensemble des états stables permet de remplacer les patterns appris, dans le paquet à présenter à l'apprentissage, par les états stables mémorisés, pour que le réseau n'oublie pas les patterns précédemment appris. L'algorithme d'apprentissage continu proposé pour la BAM multiassociative peut se manipuler avec une couche d'association incrémentale. Nous présentons dans ce qui suit l'algorithme d'apprentissage PRLAB en ligne pour la BAM multiassociative :

Algo.3 Algorithme d'apprentissage en ligne pour la BAM

```

- Initialisation aléatoire des poids et des seuils entre -1 et +1, (une seule fois à la création de la BAM triple)
- Présenter un exemple à apprendre ( $X^{(A)}$ ,  $Y^{(A)}$ ),
- Si la BAM multiassociative relaxe sur le même exemple
  Alors l'exemple est déjà connu
- Sinon
  début
  /* Estimer les  $n$  états stables ( $X^{(P)}$ ,  $Y^{(P)}$ ,  $Z^{(P)}$ ) du réseau
  avec  $P=1..n$  et  $n < N$  avec  $N$  est la capacité de stockage
  de la BAM =  $\dim Z$  / Taille de la zone, et Générer un code
  de liage  $Z^{(A)}$  pour l'exemple à apprendre ( $X^{(A)}$ ,  $Y^{(A)}$ ), */
  - Pour  $n$  de 1 à  $N$  faire
  début
  Générer un code de liage  $Z^{(E)}$  à la couche d'association en
  activant la Zone ( $n$ ),
  Propagation de l'information aux couches perceptives. On
  obtient ( $X^{(E)}$ ,  $Y^{(E)}$ ),
  Retour de l'information à la couche d'association  $Z^{(E)}$ ,
  S'il n'y a pas de changement ( $Z^{(E)} = Z^{(E')}$ )
  Alors ( $X^{(E)}$ ,  $Y^{(E)}$ ,  $Z^{(E)}$ ) appartient à l'espace des  $N$ 
  solutions (état stable),
  Sinon sortez de la boucle,
  Fin pour

```

- Si $n > N-1$
Alors la capacité de la BAM multiassociative est atteinte.
- Sinon
Début
 - Générer un code de liage $Z^{(A)}$ en activant la zone (n), pour l'exemple à apprendre $(X^{(A)}, Y^{(A)})$,
 - Le paquet à présenter dans l'apprentissage contient l'exemple $(X^{(A)}, Y^{(A)}, Z^{(A)})$ et les n états stables estimés $(X^{(E)}, Y^{(E)}, Z^{(E)})$ avec $E=1..n$,
 Fin sinon
 - faire
Pour tous les n états stables estimés $(X^{(P)}, Y^{(P)}, Z^{(P)})$ et l'exemple $(X^{(A)}, Y^{(A)}, Z^{(A)})$
mettre à jour les poids et les seuils du réseau selon l'algorithme d'apprentissage présenté dans la section précédente (Algo.1).
 - jusqu'à stabilisation du réseau.
- Fin sinon
- Fin

4.1 Expériences

Pour évaluer les capacités d'apprentissage et de rappel de la BAM triple continue, nous avons fait nos expériences sur la base de données aléatoires comportant 50 paires. La taille des trois couches X, Y et Z est de 50 neurones chacune. Nous avons fait apprendre un nombre croissant de patterns aléatoires à la BAM triple en ligne. En fixant les paramètres d'apprentissage $\lambda = 1.9$ et $\xi = 50$, sur 100 expériences, nous avons trouvé les résultats présentés dans le tableau (Tab.1).

	10 paires (taille zone=5)		20 paires (taille zone=2)	
	Moyen	Ecart type	Moyen	Ecart type
Nombre d'époques	3,00	0,00	5.40	1.01
Nombre de réverbérations	1,11	0,17	1.31	0.22
Taux d'erreur de rappel	0,30 %	0,64	1.30 %	1.45

Tab. 1 Apprentissage en ligne des patterns du corpus aléatoire.

Les résultats présentés dans le tableau (Tab.1) montrent que la BAM triple en ligne peut apprendre 20 paires avec un taux d'erreur de rappel de 1.30 %. Mais nous avons remarqué, durant nos expériences, que si le nombre de paires dépasse 25 paires (taille de la zone égale à un neurone) l'apprentissage devient problématique. Ceci implique que la capacité de stockage de la BAM triple en ligne est inférieure à $0.5 \cdot N$ paires (N étant la taille de la couche d'association) pour apprendre les patterns de cette base de données.

4.2 Synthèse sur la BAM associative continue

La BAM multiassociative, entraînée par l'algorithme d'apprentissage continu, s'adapte aux nouvelles données qui arrivent sans oublier les données apprises. Cet algorithme propose une solution à l'une des principales limites du modèle de la BAM multiassociative. La technique d'initialisation de la couche d'association par

un code partagé par zones et l'estimation des états stables sont les clés de cet algorithme. Une difficulté liée à l'utilisation de cet algorithme est qu'il requiert la connaissance préalable du nombre maximum des patterns à apprendre et la taille de la zone, paramètres qui influent sur les performances du réseau. Un des principaux problèmes de cette technique de zone est la difficulté de trouver un bon compromis entre la capacité et la stabilité du réseau. Ainsi cette technique n'exploite pas la taille de la couche d'association. L'apprentissage incrémental (par l'ajout de zones dans la couche associative) peut être une solution.

5 La BAM multiassociative synchrone

Les modèles neuromimétiques proposés pour une mémoire hétéro-associative ou multiassociative, qui sont des adaptations des modèles de Hopfield, BAM ou ART, posent le problème de l'influence de l'absence d'un membre de l'association sur les performances de rappel. Dans le cas du rappel des paires apprises depuis la présentation d'un seul membre de l'association, la simulation de l'absence d'une ou de plusieurs modalités en remplaçant le pattern manquant soit par un pattern blanc, soit un pattern aléatoire ou par un pattern nul, entraîne un biais conceptuel assorti d'un taux d'erreur de rappel important.

Le problème est que les modèles proposés sont basés sur l'idée de diviser l'espace d'entrées par le nombre de modalités sensorielles (Fig.11), alors que les équations de ces modèles considèrent l'espace d'entrées comme un seul pattern. Ceci peut avoir des conséquences sur les performances du réseau en cas de rappel des patterns appris depuis des versions incomplètes, pour lesquelles le réseau considère comme un ajout de bruit sur l'espace d'entrées le remplacement du pattern manquant par un pattern blanc ou un pattern aléatoire.

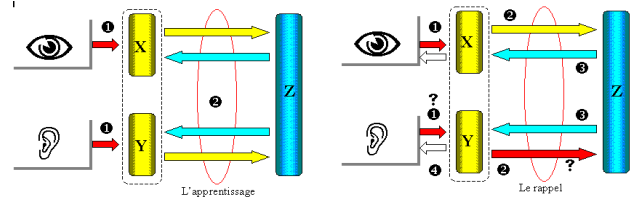


Fig.11 - L'approche de séparation de l'espace d'entrée où la convergence et la propagation sont asynchrone dans l'apprentissage et le rappel. Ce qui pose un problème de simulation de l'absence des modalités et d'ajout de bruit dans la phase de rappel.

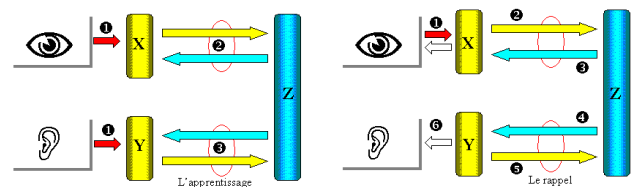


Fig.12- L'approche de séparation de l'espace d'entrée en synchronisant la convergence et la propagation dans l'apprentissage et le rappel. Ce qui résout le problème de simulation de l'absence des modalités et d'ajout de bruit dans la phase de rappel.

Le modèle que nous proposons distingue les différentes modalités au niveau des équations du réseau en

synchronisant la convergence et la propagation de l'information (Fig.12).

La synchronisation de la propagation de l'information entre les couches perceptives et la couche d'association, dans les phases d'apprentissage et de rappel, signifie que, à un moment donné, le flux d'information ne sera effectif qu'entre les couches ayant réellement été activées et la couche associative. Pour cela, dans notre modèle, une seule couche perceptive est activée, à l'apprentissage comme au rappel.

Nous présentons dans la section suivante l'architecture et la description du fonctionnement du modèle proposé.

5.1 Architecture de la mémoire multiassociative bidirectionnelle (BMM)

La BMM est composée de plusieurs couches perceptives et d'une seule couche d'association. Toutes les couches perceptives (unimodales) sont connectées à la couche d'association (multimodale) avec des connexions bidirectionnelles asymétriques (Fig.13).

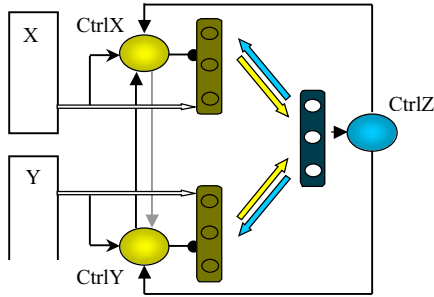


Fig.13 – Architecture de la BMM à trois couches.

Chaque couche perceptive possède un neurone de contrôle qui a pour fonction d'exciter la couche de perception quand il y a une information pertinente dans le vecteur d'entrée ou une excitation par un neurone de contrôle de la couche d'association. Sans cette excitation, la couche d'association reste inactive.

Quand la couche d'association est stable, le neurone de contrôle de la couche d'association a pour fonction de désactiver la couche de perception active et d'activer la couche de perception suivante.

5.2 L'apprentissage

La synchronisation du flux de l'information au niveau de l'apprentissage de la BMM triple sera représentée par la synchronisation de l'adaptation des poids de connexions entre les couches perceptives et la couche d'association. Le processus d'apprentissage s'effectue en deux étapes :

1. Adaptation des poids de connexions entre la couche X et la couche Z ($W^{X \rightarrow Z}$, $W^{Z \rightarrow X}$), ils sont modifiés en fonction des patterns $X^{(p)}$ et des codes de liages $Z^{(p)}$. Les connexions entre la couche Y et la couche d'association Z sont momentanément coupées ($CtrlY = 0$), à la fin de l'adaptation, la couche X sera désactivée ($CtrlX = 0$) et la couche Y sera activée ($CtrlY = 1$),

2. Adaptation des poids de connexions entre la couche Y et la couche Z ($W^{Y \rightarrow Z}$, $W^{Z \rightarrow Y}$), ils sont modifiés en fonction des patterns $Y^{(p)}$ et des codes de liages $Z^{(p)}$, les

connexions entre la couche X et la couche d'association Z sont momentanément coupées ($CtrlX = 0$).

Pour les formules d'adaptation des poids et des seuils, un problème se pose : l'adaptation des seuils de la couche d'association, qui sont modifiés dans les deux étapes.

La solution que nous préconisons est de fixer les seuils de la couche d'association à des valeurs constantes, par exemple à des valeurs nulles. Il est aussi possible de mettre deux seuils pour chaque neurone multimodal. Si on applique la première solution les formules d'adaptation des poids $W^{X \rightarrow Z}$ et $W^{Y \rightarrow Z}$ seront :

$$\Delta w_{ki}^{X \rightarrow Z} = -\frac{\lambda}{\dim X} \cdot (S_{Z_k}^p - \xi \cdot X_i^{(p)}) \cdot Z_k^{(p)}$$

$$\Delta w_{ki}^{Y \rightarrow Z} = -\frac{\lambda}{\dim Y} \cdot (S_{Z_k}^p - \xi \cdot Y_i^{(p)}) \cdot Z_k^{(p)}$$

5.3 Le rappel

Dans le modèle de la BMM triple, après la présentation d'un pattern x à la couche perceptive X, et l'initialisation aléatoire de la couche d'association Z et de la couche perceptive Y :

- le pattern x est mis en entrée de la couche X (x^0 est un état initial de la couche X),
- initialisation aléatoire de la couche Y (y^0 est un état initial de la couche Y),
- initialisation aléatoire de la couche Z (z^0 est un état initial de la couche Z),

Le processus de rappel consiste en deux phases :

1. Activation de la couche X pour aller mettre à jour les neurones de la couche d'association Z, puis mise à jour de la couche X, et ainsi de suite jusqu'à ce qu'un état d'équilibre (stable) soit atteint. Le processus de réverbération est :

- 1^{ère} réverbération $X \leftrightarrow Z$:

$$\begin{aligned} x^0 &\rightarrow W^{X \rightarrow Z} \rightarrow z^1 \\ x^1 &\leftarrow W^{Z \rightarrow X} \leftarrow z^1 \\ \text{si } ((x^0 = x^1) \text{ et } (z^0 = z^1)) \end{aligned}$$

alors arrêter la réverbération entre X et Z.

- i^{ème} réverbération $X \leftrightarrow Z$:

$$\begin{aligned} x^{i-1} &\rightarrow W^{X \rightarrow Z} \rightarrow z^i \\ x^i &\leftarrow W^{Z \rightarrow X} \leftarrow z^i \\ \text{si } ((x^{i-1} = x^i) \text{ et } (z^{i-1} = z^i)) \end{aligned}$$

alors arrêter la réverbération entre X et Z.

2. Après stabilisation des couches X et Z sur un état (x^i , z^i), les connexions entre la couche X et la couche Z seront coupées, et la couche Y sera activée. L'activation part de la couche Z pour aller mettre à jour les neurones de la couche Y, puis revient pour une mise à jour dans la couche Z, et ainsi de suite jusqu'à ce qu'un état d'équilibre (stable) soit atteint. Le processus de réverbération est :

- 1^{ère} réverbération $Z \leftrightarrow Y$:

$$\begin{aligned} z^i &\rightarrow W^{Z \rightarrow Y} \rightarrow y^1 \\ z^{i+1} &\leftarrow W^{Y \rightarrow Z} \leftarrow y^1 \\ \text{si } ((z^i = z^{i+1}) \text{ et } (y^0 = y^1)) \end{aligned}$$

alors arrêter la réverbération entre Z et Y.

- j^{ème} réverbération $Z \leftrightarrow Y$:

$$\begin{aligned} z^{j-1} &\rightarrow (W^{Z \rightarrow Y}) \rightarrow y^j \\ z^j &\leftarrow (W^{Y \rightarrow Z}) \leftarrow y^j \end{aligned}$$

si $((Z^{i-1} = z^i) \text{ et } (Y^{j-1} = y^j))$

alors arrêter la réverbération entre Z et Y.

A la fin du processus de rappel, la sortie des couches perceptives $(X^{(P)}, Y^{(P)})$ correspond à la réponse de la BMM.

Les algorithmes d'apprentissage et de rappel de la BMM sont donc les suivants :

Algo.4 - Algorithme d'apprentissage de la BMM

1. Les poids $W^{X \rightarrow Z}, W^{Z \rightarrow X}, W^{Y \rightarrow Z}, W^{Z \rightarrow Y}$ et les seuils $\theta_{X_i}, \theta_{Y_j}$, sont choisis arbitrairement entre -1 et $+1$, θ_{Z_k} sont initialisés à des valeurs nulles,

2. Pour chaque paire $(X^{(P)}, Y^{(P)})$ à apprendre faire
- Générer un code de liage $Z^{(P)}$;

3. Activer la couche X et désactiver la couche Y,
CtrlX=1 et CtrlY=0

4. Pour toutes les paires $(X^{(P)}, Z^{(P)})$ faire

Pour tous les neurones des couches perceptives X faire
si $S_{X_i}^p X_i^{(P)} \leq 0$ alors mettre à jour poids $W^{Z \rightarrow X}$ et seuils selon :

$$\Delta w_{ik} = -\frac{\lambda}{\dim Z + 1} \cdot (S_{X_i}^p - \xi \cdot Z_k^{(P)}) X_i^{(P)}$$

$$\Delta \theta_{X_i} = -\frac{\lambda}{\dim Z + 1} (S_{X_i}^p - \xi \cdot X_i^{(P)})$$

Pour tous les neurones de la couche d'association Z faire
si $S_{Z_k}^p Z_k^{(P)} \leq 0$ alors mettre à jour poids $W^{X \rightarrow Z}$ selon :

$$\Delta w_{ki} = -\frac{\lambda}{\dim X} \cdot (S_{Z_k}^p - \xi \cdot X_i^{(P)}) Z_k^{(P)}$$

5. Répéter la quatrième étape jusqu'à stabilisation du réseau (pas de mise à jour).

6. Activer la couche Y et désactiver la couche X,
CtrlX=0 et CtrlY=1

7. Pour toutes les paires $(Y^{(P)}, Z^{(P)})$ faire

Pour tous les neurones des couches perceptives Y faire
si $S_{Y_j}^p Y_j^{(P)} \leq 0$ alors mettre à jour poids $W^{Z \rightarrow Y}$ et seuils selon :

$$\Delta w_{jk} = -\frac{\lambda}{\dim Z + 1} \cdot (S_{Y_j}^p - \xi \cdot Z_k^{(P)}) Y_j^{(P)}$$

$$\Delta \theta_{Y_j} = -\frac{\lambda}{\dim Z + 1} (S_{Y_j}^p - \xi \cdot Y_j^{(P)})$$

Pour tous les neurones de la couche d'association Z faire
si $S_{Z_k}^p Z_k^{(P)} \leq 0$ alors mettre à jour poids $W^{Y \rightarrow Z}$ selon :

$$\Delta w_{kj} = -\frac{\lambda}{\dim Y} \cdot (S_{Z_k}^p - \xi \cdot Y_j^{(P)}) Z_k^{(P)}$$

8. Répéter la quatrième étape jusqu'à stabilisation du réseau (pas de mise à jours).

9. Désactiver les couches X et Y, CtrlX=0 et CtrlY=0
Avec

$$S_{X_i}^p = \sum_{k=1}^{\dim Z} w_{ki}^{Z \rightarrow X} Z_k^{(P)} - \theta_{X_i} \quad S_{Y_j}^p = \sum_{k=1}^{\dim Z} w_{kj}^{Z \rightarrow Y} Z_k^{(P)} - \theta_{Y_j}$$

$$S_{Z_k}^p = CtrlX \cdot \left(\sum_{i=1}^{\dim X} w_{ik}^{X \rightarrow Z} X_i^{(P)} \right) + CtrlY \cdot \left(\sum_{j=1}^{\dim Y} w_{jk}^{Y \rightarrow Z} Y_j^{(P)} \right) - \theta_{Z_k}$$

Algo.5 - Algorithme de Rappel de la BMM

1. Présenter un pattern $X^{(P)}$;

2. CtrlX=1, CtrlY=0, CtrlZ=0;

/* phase de relaxation entre X et Z afin de trouver le code d'association : $Z^{(P)}$ */

3. Calculer la sortie de la couche d'association $Z^{(P)}$ par la propagation de $X^{(P)}$

4. faire

- Calculer la sortie de la couche perceptive X : $X^{(P)}$

- Calculer la sortie de la couche d'association $Z^{(P)}$

Jusqu'à stabilisation du réseau : $Z^{(P)} = Z^{(P)}$ et $X^{(P)} = X^{(P)}$,

/* phase de relaxation entre Z et Y afin de trouver le pattern manquant : $Y^{(P)}$ */

5. CtrlZ=1 $\Rightarrow$ CtrlX=0 et CtrlY=1 ;

6. Calculer la sortie de la couche perceptive Y : $Y^{(P)}$ par la propagation de $Z^{(P)}$

7. faire

- Calculer la sortie de la couche d'association Z : $Z^{(P)}$

- Calculer la sortie de la couche perceptive Y : $Y^{(P)}$

Jusqu'à stabilisation du réseau et $Z^{(P)} = Z^{(P)}$ et $Y^{(P)} = Y^{(P)}$,

8. La sortie des couches perceptives $(X^{(P)}, Y^{(P)})$

correspond à la réponse de la BMM

5.4 Expériences sur la BMM

Pour évaluer les propriétés et les capacités d'apprentissage et de généralisation de la BMM, nous avons choisi de faire nos expériences sur une base de paires aléatoires : les patterns $X^{(P)}$ seront encodés par la couche perceptive X sur 50 neurones, les patterns $Y^{(P)}$ seront encodés par la couche perceptive Y sur 50 neurones, les codes d'association $Z^{(P)}$ seront encodés par la couche Z sur 50 neurones. Le codage de la couche d'association est effectué par zones d'affectations.

5.4.1 Rappel depuis un seul pattern non bruité

Dans la phase de rappel, nous présentons un seul membre des paires apprises. Le processus de rappel consiste en deux phases :

- La propagation de l'activation depuis la couche perceptive X jusqu'à la couche d'association Z, puis le retour de l'activation et ainsi de suite jusqu'à ce qu'il n'y ait plus de changement dans l'activation des deux couches,

- La propagation de l'activation depuis la couche d'association Z jusqu'à la couche perceptive Y, puis le retour de l'activation, et ainsi de suite, jusqu'à ce qu'il n'y ait plus de changement dans l'activation des deux couches.

Sur 100 expériences, les paramètres d'apprentissage de la BMM ont été fixés à : $\lambda = 1.9$ et $\xi = 25$. Les résultats de nos expériences sont présentés dans le tableau (Tab.2).

D'après les résultats présentés dans le tableau (Tab.2), le modèle de BMM peut mémoriser N associations pour un réseau de N neurones multimodaux, et il peut rappeler les N associations depuis la présentation d'un seul membre des associations apprises. Notre modèle a une caractéristique intéressante : l'absence d'une modalité n'influe pas sur les performances de rappel des paires apprises, avec cette caractéristique notre modèle est adapté au problème de l'association, puisqu'il suffit de

présenter une unique modalité pour que la BMM rappelle les modalités associées.

	10 paires		30 paires		50 paires	
	Moyen	σ	Moyen	σ	Moyen	σ
Nombre d'époques	6,57	0,74	12,61	1,35	12	1,33
Nombre de réverbérations	2,11	0,17	2,82	1,89	4,16	4,93
% d'erreur de rappel	0,36 %	0,71	1,83 %	1,87	2,25 %	2,27

Tab.2 Résultats de rappel des paires apprises depuis un seul pattern.

5.4.2 Rappel depuis un seul pattern bruité

Nous testons ici la résistance de la BMM lorsque le seul pattern présenté comme perception au réseau contient du bruit. Nous mesurons l'effet du bruit sur la couche X seulement.

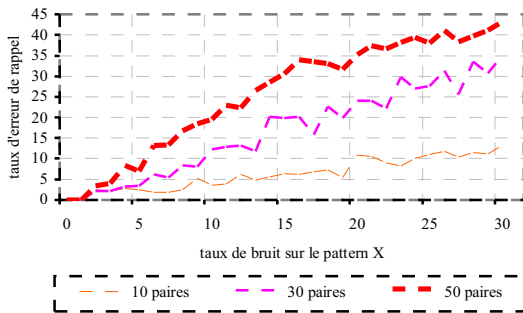


Fig.14 Pourcentage d'erreur commis pour le rappel de paires en fonction de niveau de bruit sur le pattern X et le nombre de paires à apprendre.

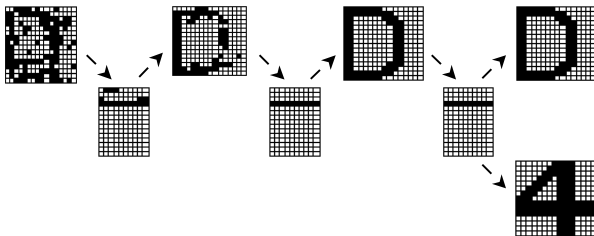


Fig.15 Processus de rappel de la BMM de la paire (D, 4) depuis la présentation d'un pattern X avec un bruit de 20% .

Nos expériences montrent que la BMM est robuste au bruit sur le seul pattern présenté en fonction du nombre de paires qu'il stocke :

- 50 paires, pour un bruit de 7.00 % sur le pattern X, le taux de rappel parfait est supérieur à 90.00 %,

- 10 paires, pour un bruit de 20.00 % sur le pattern X, le taux de rappel parfait est supérieur à 90.00 %,

Afin de comparer la robustesse de la BMM au bruit par rapport à la BAM triple, nous avons effectué diverses expériences sur le modèle de la BMM. Le rappel des associations apprises depuis la présentation d'un seul pattern X est bon, avec un taux d'erreur de rappel 0.65%. En comparaison de la BAM triple, où le taux d'erreur de rappel des paires apprises depuis une version incomplète

est supérieur à 30%, nous obtenons une réelle amélioration.

Nous avons fait d'autres expériences sur les deux modèles, en mesurant l'effet du bruit sur le pattern X seulement. Pour la BMM le pattern Y n'est pas présenté (Fig.15), tandis que dans la BAM triple le pattern Y est présenté sans bruit.

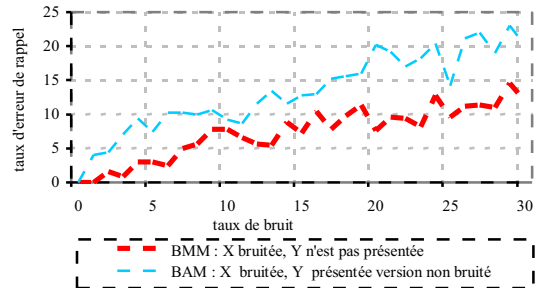


Fig.16 La résistance du modèle de BMM au bruit sur X sans présentation de Y comparée à la résistance de la BAM au bruit sur X avec la présentation d'une version non bruitée de Y.

Les résultats présentés dans la figure 16 montrent bien que la BMM est plus robuste au bruit sur le pattern X (Y n'est pas présenté) que la BAM triple avec la présentation de Y (version non bruitée). Donc notre modèle BMM répond aux propriétés du phénomène de l'association, en mémorisant les associations entre plusieurs modalités sensorielles. Une fois qu'une modalité sensorielle est présentée (apprise ou bruitée) la BMM rappelle les modalités associées.

5.4.3 Synthèse sur le modèle de BMM

La nouvelle approche que nous avons proposé pour la modélisation des mémoires multi-associatives, en se basant sur la synchronisation du flux d'informations entre les couches perceptives et la couche d'association à l'apprentissage et au rappel, a montré une réelle amélioration des performances et de la robustesse de la BMM triple. L'absence d'un membre des associations de perception apprises n'influe pas sur les performances de rappel des patterns associés. Donc le modèle de la BMM semble adapté au problème de l'association, puisqu'il suffit de présenter une unique modalité (version complète ou bruitée) pour que la BMM rappelle les modalités associées.

Pour que la BMM puisse apprendre en ligne, l'algorithme d'apprentissage proposé dans la section 4 reste utile pour ce modèle.

6 Conclusion

Le but de ce travail était de proposer un modèle de réseau de neurones pour une mémoire d'association multimodale en tenant compte des propriétés du phénomène de l'intégration multisensorielle, des performances et de la robustesse de la mémoire.

Nous avons étudié les propriétés et les capacités de la BAM triple proposée par E.Reynaud [9] dans sa thèse. Après expérimentation, nous avons isolé des problèmes qui influent sur les performances et la robustesse de

rappel de la BAM triple d'E. Reynaud : l'initialisation de la couche d'association, dans la phase d'apprentissage, influe sur la robustesse au bruit et la dépendance de l'algorithme d'apprentissage à l'ordre de tirage des patterns. De plus, la BAM d'E. Reynaud a des limites principales : elle n'apprend pas continuellement les nouvelles associations et a une faible capacité de rappel des associations apprises en cas d'absence d'une ou plusieurs modalités.

Nous avons proposé une méthode d'initialisation de la couche d'association par zones d'affectations qui garantit la convergence de l'apprentissage et la robustesse au bruit. Cette méthode offre, de plus, un intérêt pour l'apprentissage en ligne et l'aspect temporel de ce codage et améliore les performances et la robustesse de la BAM d'E. Reynaud.

L'autre objectif de notre travail est de garder les propriétés du réseau tout en permettant un apprentissage continu. Nous avons, pour cela, proposé un algorithme original, basé sur l'estimation des états stables par la propagation inverse de l'activation des zones d'affectations. Ceci a permis à la BAM multiassociative d'associer de nouvelles modalités sensorielles et d'évoquer les associations déjà acquises au fur et à mesure de l'acquisition de nouvelles données.

Nous avons également proposé un modèle de mémoire multiassociative (BMM) en se basant sur une idée originale, la synchronisation de convergence et de propagation des flux d'informations.

Après expérimentation, le modèle de la BMM multiassociative synchrone montre ses performances et sa robustesse pour rappeler les modalités associées depuis la présentation d'une unique modalité sensorielle même si elle est bruitée.

Si notre modèle de base, la BAM triple développée par E. Reynaud, cherchait à simuler les propriétés d'intégration multisensorielle, le réseau obtenu semble plus proche des capacités cognitive de l'hippocampe. En effet, ce modèle ne permet pas de phénomènes de généralisation mais possède de grandes capacités de mémoire « flash », il est capable d'apprendre par cœur des associations incomplètes.

Notre modèle, et particulièrement son algorithme d'apprentissage PRLAB, permet, dans sa version originelle, de distribuer une information (un ensemble d'associations) dans les poids de notre réseau. Cet algorithme souffre néanmoins d'un problème conceptuel, il nécessite de connaître la totalité des informations à apprendre pour débiter son apprentissage.

La solution que nous proposons, codage par zones d'affectations, s'effectue à l'aide d'un codage localiste (chaque zone code une association précise), et ne permet plus la distribution de l'information.

Nous souhaitons pour la suite approfondir les formalisations que nous avons engagées ici avec d'autres algorithmes d'apprentissages, en se basant sur la synchronisation de la convergence et de la propagation des flux d'informations, l'apprentissage en ligne et la proximité spatio-temporelle.

7 Remerciements

Ce travail a été effectué dans le cadre du projet européen IST-FET “MirrorBot” et de l'ACI Neurosciences Intégratives : “Modèles distribués de réseaux de neurones impulsions pour l'intégration multisensorielle”.

8 Bibliographie

- [1] Calvert, G., Hansen, P., Iversen, S., and Brammer, M. *Detection of audio-visuel intégration sites in humans by application of electrophysiological criteria to the BOLD effect*. Neuroimage, 14(2) :427-7438, 2001.
- [2] M. Hagiwara, *Multidirectional Associative Memory*, Proc. IEEE Joint Conf. On Neural Networks, Washington, D.C, January 15-19, 1990. New York : IEEE, Vol I, pp. 6-6, 1990.
- [3] J. Hertz, A. Krogh et R. G. Palmer. *An Introduction to the Theory of Neural Computation*. Lecture notes Volume I, Addison Wesley, 1991.
- [4] B. Humpert. *Bidirectional associative memory for three patterns*. In International neural network conference, Kluwer academic publishers. volume 2, pages 843-851, Paris, July 1990.
- [5] H. Kang, *Multilayer associative neural network (MANNs): Storage capacity versus perfect recall*, IEEE Transactions on Neural Networks, vol. 5, no. 5, pp. 812-822, September 1994.
- [6] B. Kosko, *Adaptive bidirectional associative memories*, Applied Optics, vol. 26, no. 23, pp. 4947-4960, 1987.
- [7] B. Kosko, *Bidirectional associative memories*, IEEE Transactions on Systems, Man, and Cybernetics, vol. 18, no. 1, pp. 49-60, 1988.
- [8] H. Oh and S. C. Kothari, *Adaptation of the relaxation method for learning in bidirectional associative memory*, IEEE Transactions on Neural Networks, vol. 5, no. 4, pp. 573-583, July 1994.
- [9] E. Reynaud, *Modélisation connexionniste d'une mémoire associative multimodale*. PhD. Thesis, Institut National Polytechnique de grenoble, 2002.
- [10] C.-C. Wang and H.-S. Don, *An analysis of high-capacity discrete exponential BAM*, IEEE Transactions on Neural Networks, vol. 6, no. 2, pp. 492-496, 1995.
- [11] Z. Wang, *A bidirectional associative memory based on optimal linear associative memory*, IEEE Transactions on Computers, vol. 45, no. 10, pp. 1171-1179, 1996.
- [12] Y. F. Wang, J. B. Cruz, and S. H. Mulligan, *Two coding strategies for bidirectional associative memory*, IEEE Transactions on Neural Networks, vol. 1, no. 1, pp. 81-92, 1990.
- [13] Wasserman, P. D., *Neural computing : Theory and practice*. New York. 1987.
- [14] Z.-B. Xu, Y. Leung, and X.-W. He, *Asymmetric bidirectional associative memories*, IEEE Transactions on Systems, Man, and Cybernetics, vol. 24, no. 10, pp. 1558-1564, 1994.
- [15] X. Zhuang, Y. Huang, and S.-S. Chen, *Better learning for bidirectional associative memory*, Neural Networks, vol. 6, no. 8, pp. 1131-1146, 1993.