STOCKAGE D'INFORMATION SUR UNE STRUCTURE EN BOUCLE

Jean-Pierre Bachy, MD
Séminaire de Modélisation du 7 mai 1993
TIMC IMAG
UJF Grenoble

Nous proposons un modèle de stockage d'information issu d'une réflexion sur les circuits réverbérants, supports éventuels de la mémoire à court terme. Il se prête à une recherche par le contenu de formes incomplètes ou bruitées qui l'apparente aux mémoires associatives.

Les vecteurs mémorisés sur cette structure en boucle sont quasi orthogonaux. Sachant que l'orthogonalité des images stockées est nécessaire pour obtenir une bonne mémorisation sur les réseaux de neurones on peut envisager ce mode de stockage transitoire et d' orthogonalisation comme un prétraitement d'images destinées au stockage sur mémoire associative.

Il est à noter que dans ce cas, les images étant stockées sous forme chaotique, la reconnaissance d'une image stockée sur le réseau passe obligatoirement par une phase de prétraitement.

1. Théorie de la transformation vectorielle

Dans une première partie nous décrivons la transformation vectorielle qui transforme le vecteur d'origine en un vecteur de **même dimension** dont les **composantes** sont **délocalisées** de façon **chaotique**. Lorsque certaines conditions sont remplies, l'étirement-recouvrement introduit par la fonction modulo ne fait que délocaliser les composantes sans les superposer. La transformation du vecteur est biunivoque.

1.1 Transformation d'un vecteur en un vecteur chaotique

Si on décrit un vecteur comme une suite de n composantes C_i de valeur V_i. Chaque composante a une position A_i.

Le vecteur d'origine peut s'écrire:

$C_1, C_i \ldots, C_n$ de valeurs $V_1, V_i \ldots\ldots, V_n$

avec i {1.. n}

La transformation va porter sur la position des composantes successives du vecteur: la suite des positions va subir une permutation qui va délocaliser les composantes d'origine tout en conservant leur valeur.

Pour chaque composante C_i d'adresse A_i, on va calculer une nouvelle adresse A'_i qui correspondra à la position de cette composante dans le vecteur transformé.

(1) $$\mathbf{A'_i = i * b \bmod (n+1)}$$

avec $i \{1..n\}$

b premier par rapport à (n+1)

Si on calcule les valeurs successives de la suite A'_1 à A'_n

(2) $$\mathbf{A'_i = (A'_{i-1} + b) \bmod (n+1)}$$

avec $A'_0 = 0$

$i \{1.. n\}$

b premier par rapport à (n+1)

On remarque que la suite est de la forme

(3) $$\mathbf{A_i = (a * A_{i-1} + b) \bmod (n+1)}$$

avec $a = 1$

$i \{1.. n\}$

b premier par rapport à (n+1)

Cette formule a pour particularité de générer la suite des nombres de 1 à n dans un ordre chaotique [DEW87]. Cette suite est différente pour chaque valeur des paramètres à ceci près qu'apparaissent des **récurrences** lorsque le nombre de permutations possibles est atteint.

Les composantes du vecteur d'origine sont délocalisées et de manière différente pour chaque valeur du paramètre b.

Lorsqu'on fait varier b pour un même vecteur d'origine, A'_1 (qui a pour valeur b mod (n+1)), joue le rôle de germe pour la série des nouvelles adresses. Sa valeur varie de 1 à n et peut prendre les n valeurs de 1 à n. A un vecteur donné on peut donc faire correspondre n vecteurs transformés si b est premier et non multiple ou sous-multiple de (n+1).

Cette transformation vectorielle est une permutation qui peut être aisément réalisée par un produit matriciel entre le vecteur d'origine d'une part et une matrice de permutation n x n (P) construite à partir de la série chaotique correspondant à un paramètre b donné.

exemple avec b = 3:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} e \\ c \\ a \\ f \\ d \\ b \end{bmatrix}$$

1.2 Transformation inverse

La transformation est biunivoque et la transformation inverse s'obtient en multipliant le vecteur transformé par la matrice P^{-1}.

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} e \\ c \\ a \\ f \\ d \\ b \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}$$

2- Mémorisation par superposition de vecteurs transformés

2.1- Généralités

Le principe de base d'une mémoire associative est que tout élément nouveau est interprété comme un bruit par rapport à l'ensemble des éléments déjà mémorisés par la structure. La réciproque est que les éléments mémorisés forment un bruit pour un vecteur donné.

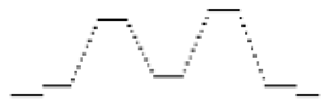
Dans une mémoire associative de type Hopfield, le vecteur à mémoriser modifie une matrice de valeurs constituée par les éléments déjà mémorisés.

Dans notre modèle le vecteur, après transformation, est superposé aux vecteurs déjà mémorisés et il est considéré comme un bruit par le réseau.

La zone de stockage est représentée par un certain nombre de vecteurs transformés de longueur différente.

Les vecteurs utilisés ont 120 composantes. Chaque composante a pour valeur un niveau de gris {0..100} d'une image 1D.

Exemple:



2.2 Phase de stockage

On transforme le vecteur d'origine O de composantes C_i , i {1..n}, en un vecteur transformé T de composantes C'_i , i {1..d} de dimension différente (d) très supérieure à celle du vecteur origine (n).

La formule (1) de calcul de $\mathbf{A'_i}$ devient alors:

$$\mathbf{A'_i = i * b \bmod (d+1)}$$

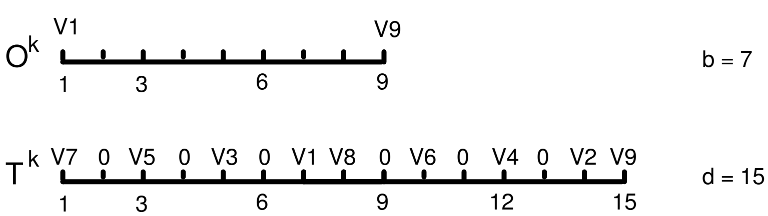
où i{1..d} b premier par rapport à (d+1)

Les composantes C'_i d'adresse $\mathbf{A'_i}$, i {1..n}, ont pour valeurs V'_i celles, permutées, du vecteur d'origine.

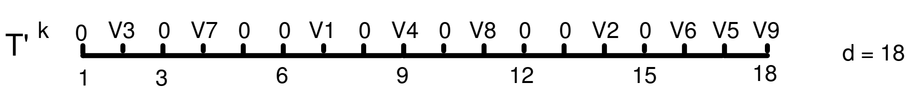
Les composantes C'_i d'adresse $\mathbf{A'_i}$, i {n+1..d}, ont pour valeur $V'_i = 0$.

Un paramètre b, choisi dans la suite des nombres premiers, est affecté à chaque vecteur O à mémoriser.

Exemple



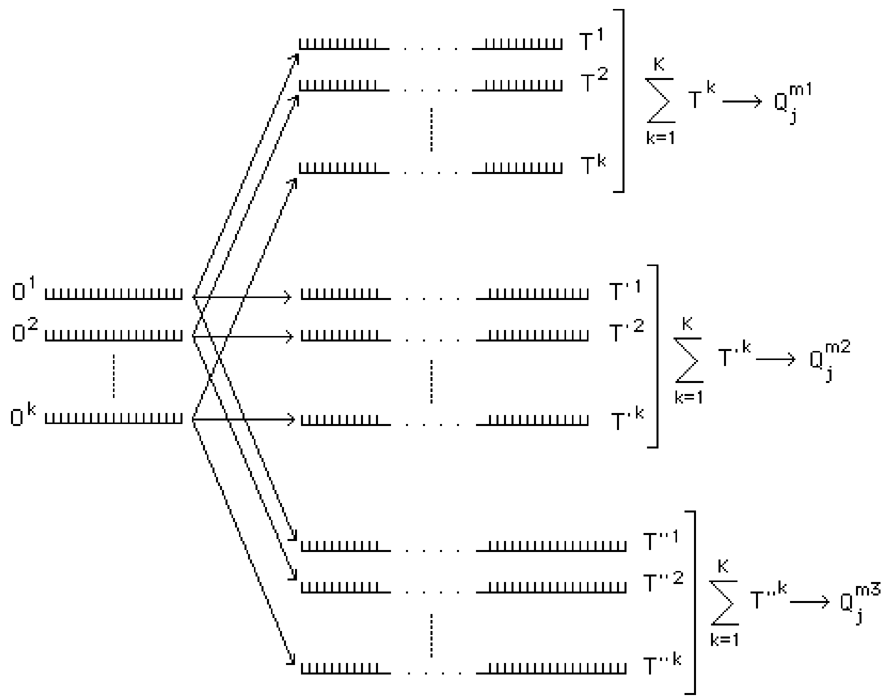
On mémorise T, on modifie d , b ne change pas, c'est la clé affectée au vecteur



On obtient ainsi pour chaque vecteur O^k, k {1, ..., K}, un ensemble de transformés de O^k, {T^k, T'^k, T''^k,...} de dimensions d (m1, m2, … , mM) différentes.

On fait de même pour chaque vecteur O^k, à mémoriser.

On additionne ensuite les vecteurs de même dimension d.

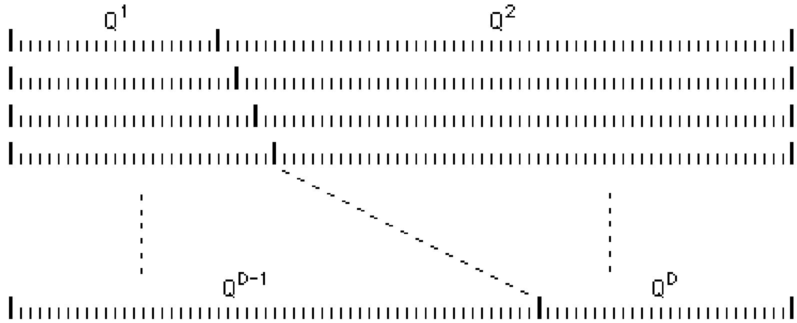


La j-ième composante d'un vecteur résultat, Q^d, d{1=m1, ..., D=mM} est, pour une dimension d donnée

$$Q_j^d = \sum_{k=1}^{K} T^k_{(j*b_k)\bmod(d+1)}$$

La variation de d va générer, pour chaque zone de stockage, un vecteur transformé différent. Un vecteur d'origine de clé “b” va donc être inscrit, sous forme de plusieurs vecteurs transformés, dans des "bruits" différents.

Les vecteurs sont associés par deux, la dimension de l'un diminue lorsque la dimension de l'autre augmente. La longueur résultante et le nombre (D) de vecteurs de stockage sont des paramètres du modèle.



2.3 Phase de restitution

La restitution du vecteur à partir de ses transformés revient à extraire un signal d'un bruit.

Nous avons choisi une méthode dérivée de celle utilisée pour les potentiels évoqués: l'annulation du bruit est obtenue par moyennage de copies multiples du signal, chacune étant bruitée de façon différente.

Pour une clé donnée le vecteur de sortie est obtenu en faisant la moyenne des vecteurs transformés inscrits sur le réseau:

$$S_j = \frac{1}{D} \sum_{d=m1}^{mD} \sum_{k=1}^{K} T^k_{(j*b_k) \bmod (d+1)}$$

3- Modèle avec une modulation associée à la transformation chaotique

Le principe est le même que le précédent mais on introduit un traitement du signal avant ou au moment du stockage. Il s’agit d’une modulation d’amplitude par un signal sinusoïdal.

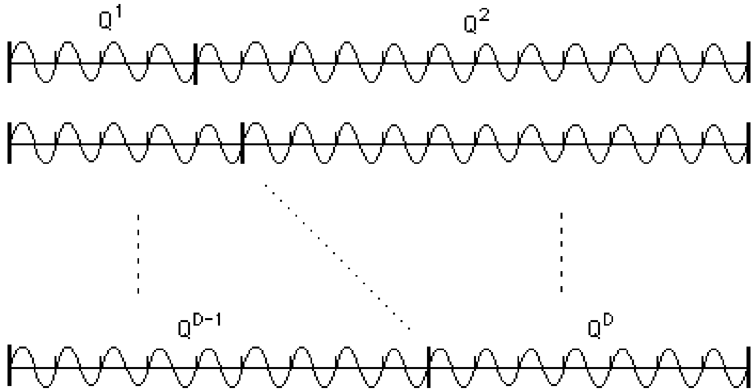
L’utilisation d’une modulation d’amplitude en radiophonie permet de décaler, dans le spectre des fréquences, la fréquence propre du signal à transmettre. Elle permet d’éviter la superposition des stations émettrices

$$Y(t) = A(t).\sin 2pi\, F.t$$

Y(t) est le signal modulé, A(t) le signal modulant, F la fréquence modulée ou porteuse.

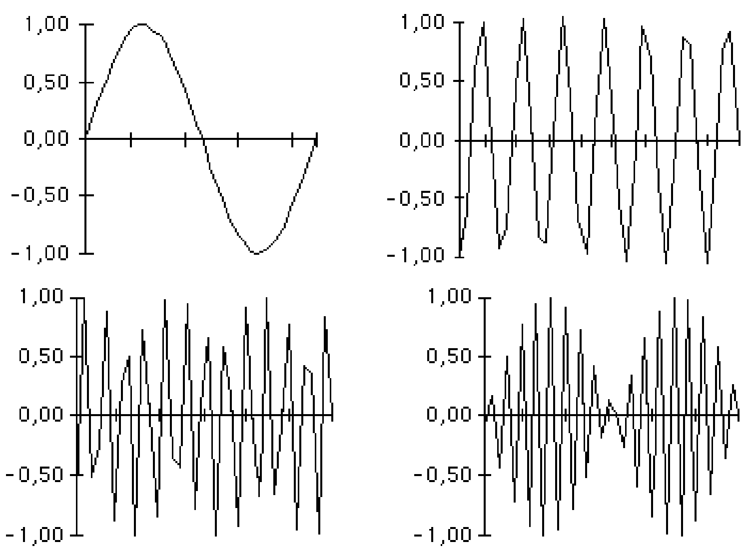
Ici la modulation va avoir pour effet d’orthogonaliser plus encore les vecteurs à mémoriser, sans perte d’information, puisque la démodulation restitue le signal d’origine.

La porteuse est générée par échantillonnage d'une sinusoïde dont la période est un sous-multiple de la dimension des vecteurs transformés. En d'autres termes, on trouve un nombre entier de périodes dans chaque vecteur transformé.

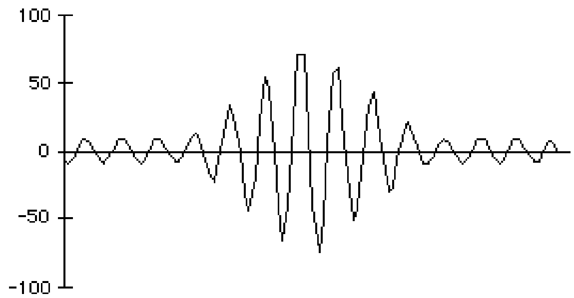


Le pas d'échantillonnage est égal au paramètre b qui est la clé affectée à chaque vecteur à mémoriser.

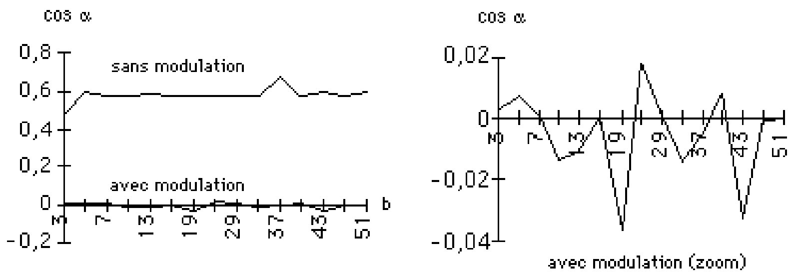
Quelques porteuses obtenues par échantillonnage de la sinusoïde de base:



exemple de signal modulé obtenu avec b = 7

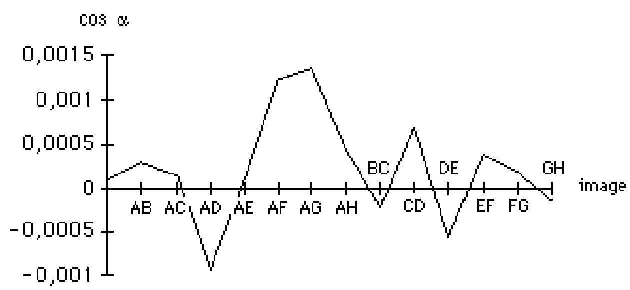


La figure suivante montre le gain obtenu en orthogonalisation pour un même vecteur, transformé avec des b premiers différents {5..51}



Le gain en orthogonalisation reste faible pour la seule transformation chaotique mais celle-ci génère une modulation propre à chaque image qui se révèle très puissante.

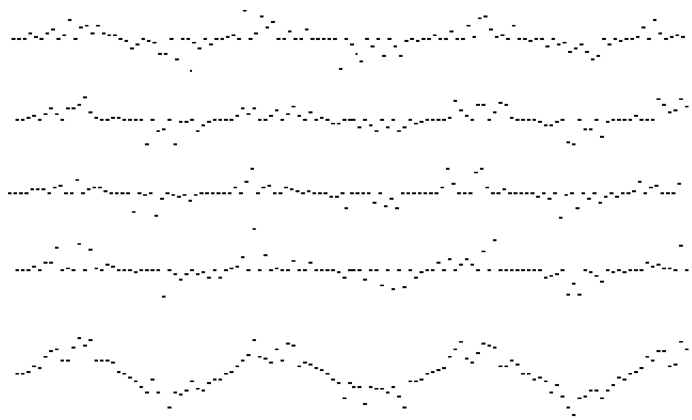
Si nous prenons deux vecteurs différents à l'origine les vecteurs transformés deviennent quasi orthogonaux:



4- Etudes:

4.1 Visualisation de l'activité du réseau

La figure ci-dessous montre les 100 premières valeurs des 4 premières zones de stockage Q^{m1}, Q^{m2}, Q^{m3}, Q^{m4}, après l'inscription de 6 vecteurs transformés T^k k{1.. 6}sur le réseau et la partie du vecteur somme correspondant. On remarque que les valeurs particulières à chaque image s'annulent et que le signal sinusoïdal échantillonné au départ est reconstitué par addition des porteuses affectées aux images.



4.2 Capacité maximale théorique de stockage:

L'opération de modulation est une méthode d'orthogonalisation. Nous l'avons comparée à la méthode de référence de GRAM SCHMIDT [KEE].

Un premier vecteur à mémoriser est transformé avec une "clé" b en un ensemble de vecteurs chaotiques de dimensions d différentes (Q^1 à Q^D). Cet ensemble constitue la base sur laquelle viendront s'additionner les ensembles suivants.

Un deuxième vecteur est transformé avec une clé différente et génère un deuxième ensemble de vecteurs chaotiques.

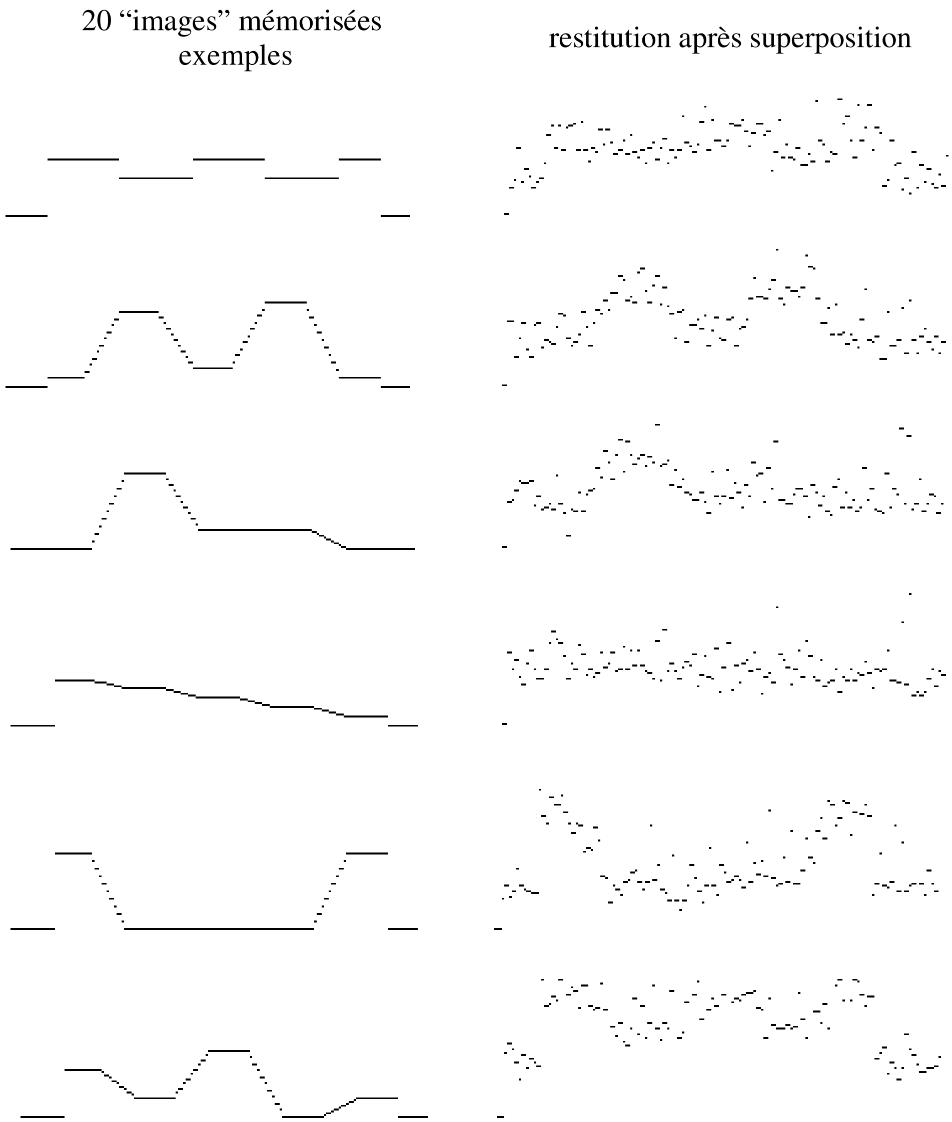
Si on veut additionner les deux ensembles de façon à ce que l'un apparaisse comme un bruit par rapport à l'autre, il faut orthogonaliser le deuxième ensemble par rapport au premier. C'était le rôle de la modulation dans notre méthode.

Dans la méthode de GRAM SCHMIDT l'ensemble est recalculé. Le coefficient à appliquer à chaque composante d'un vecteur transformé de dimension d donnée est mémorisé. Il permettra la reconstruction du vecteur d'origine en phase de restitution.

Les vecteurs suivants sont mémorisés de la même façon: chaque nouvel ensemble est recalculé de façon à être orthogonal aux éléments déjà mémorisés.

Dans notre expérimentation la méthode de GRAM SCHMIDT permet de doubler la capacité de stockage (20 vecteurs-tests à niveaux de gris au lieu de 10).

La suite logique de ce travail est l'articulation de ce modèle avec une mémoire associative classique et l'étude quantitative des capacités de mémorisation en fonction de la dimension du réseau.



Ci-dessus quelques échantillons d'images 1D mémorisées sur le réseau et leur restitution après stockage, sans traitement

5- Références

[DAV89] Eric Davalo, Patrick Naïm, *Le modèle de HOPFIED* , in des Réseaux de Neurones, Eyrolles 89, p 104-112

[DEW87] Dewdney A., *Explorez le monde étrange du chaos,* Récréations informatiques, in Pour la Science, N° 119, Sept 87, p 13-16

[KEE] James P.Keener, *Transformation and approximation,* in Principles of Applied Mathematics , ADDISON.WESLEY PUBLISHING COMPANY