

*Il y a, disions-nous, deux mémoires profondément distinctes : l'une, fixée dans l'organisme, n'est point autre chose que l'ensemble des mécanismes intelligemment montés qui assurent une réplique convenable aux diverses interpellations possibles (...) Habitude plutôt que mémoire, elle joue notre expérience passée, mais n'en évoque pas l'image. L'autre est la mémoire vraie. Coextensive à la conscience, elle retient et aligne à la suite les uns des autres tous nos états au fur et à mesure qu'ils se produisent, laissant à chaque fait sa place et par conséquent lui marquant sa date (...)*

HENRI BERGSON\*

**L**E modèle de Hopfield est peut-être celui qui, parmi les différents modèles de réseaux de neurones, a été le plus approfondi. Il fait partie de la grande classe des réseaux récurrents, ceux dont les sorties sont rebouclées en rétroaction sur les entrées.

La plupart des réseaux de neurones ne sont pas récurrents, les signaux se propageant des entrées vers des sorties si bien qu'en un nombre fini d'étapes (dépendant de la profondeur du réseau) les états de tous les neurones sont connus. Fréquemment, grâce à une règle, dite d'apprentissage, les poids des connexions évoluent et convergent vers des valeurs désirées.

En revanche dans ce chapitre les poids des connexions seront fixés au départ et ce seront les états des neurones qui varieront au cours du temps puisqu'il y a des reblouclages entre les (certaines) sorties et les (certaines) entrées. L'étude de ces réseaux portera donc sur le comportement temporel de leur *état* (défini par un vecteur dont chaque composante représente l'état, actif ou inactif, +1 ou -1, d'un neurone du réseau), et on se propose de déterminer quels sont *les états stables de la dynamique du réseau*.

Ces états stables peuvent être considérés comme des vecteurs codant pour des motifs particuliers (une image par exemple, lorsqu'on place à la queue leu-leu tous ses pixels). Il est alors possible de réaliser une *mémoire* où chaque motif mémorisé correspond à un état stable de la dynamique du réseau. Mais il nous faudra trouver une méthode pour calculer les poids des connexions afin stocker dans la mémoire un ou plusieurs motifs spécifiés.

---

\* *Matière et Mémoire*, p. 167-168.

## 2.1 Historique

Si la possibilité de stocker de l'information dans des configurations stables de systèmes dynamiques a été proposée par des auteurs comme Grossberg [Gro67], Amari [Ama72], Kohonen [Koh74], Little [Lit74] ou Cowan [Cow68], près d'une décennie avant la publication de John Hopfield [Hop82], c'est néanmoins bien ce dernier qui a formulé de façon précise une solution à ce problème et fait apparaître la notion primordiale de *bassin d'attraction* ainsi que celle, fondamentale, d'*énergie*. C'est la raison pour laquelle ce modèle, ainsi que la plupart de ses variantes ou extensions, porte son nom ; les modèles y ressemblant peu ou prou sont familièrement dénommés "à la Hopfield".

## 2.2 Introduction

Comment mémoriser des formes binaires et les retrouver même lorsqu'elles sont bruitées ? L'idée qui va être développée ici est d'utiliser des mémoires adressables par le contenu, appelées en anglais *content adressable memory* ou encore *CAM* en abrégé. Ce sont des mémoires où, pour retrouver un motif mémorisé, on présente à l'entrée une partie de ce motif. On recherche de plus un fonctionnement qui soit tel que, même si le motif présenté est bruité (c'est-à-dire possède un certain nombre de bits faux) le motif mémorisé est retrouvé intact.

Quelques exemples permettront de clarifier les idées. Supposons une base de données d'adresses du genre :

"Gaston, rue des gaffes en gros ..."

"Talon, avenue du Messie,..."

"..."

Ici, un motif est constitué, par exemple, de la concaténation des bits représentant l'adresse complète. Une mémoire associative adressable par le contenu donne l'adresse complète même si on lui présente seulement *Gaston* ou *Ga* ou encore *gros* et même, et si possible, *Gatson*.

La manière la plus évidente de procéder pour retrouver un motif mémorisé est de mesurer la distance de Hamming entre le motif présenté  $s$  et chacun des motifs mémorisés  $m^\mu$

$$H_\mu = \sum_i (s_i - m_i^\mu)^2 \quad (2.1)$$

et de choisir le motif  $\mu$  pour lequel cette distance est la plus faible. On pourrait aussi choisir le motif pour lequel le produit scalaire  $s \cdot m^\mu$  est le plus grand. Ces deux procédures sont séquentielles.

Mais si on veut faire cette opération avec un réseau de neurones, comment s'y prendra-t-on ?

On utilisera une mémoire dont la structure sera analogue à celle dessinée sur la figure 2.1. Chaque rond symbolise un neurone qui calcule à chaque pas de temps son état en fonction de l'état de tous les neurones à l'instant précédent. Il effectue tout d'abord une somme de ses entrées pondérées par les *poinds* des connexions Cette somme pondérée s'appelle *entrée totale* et est notée  $h$  pour suivre la notation des physiciens (pour des raisons que l'on verra plus tard, la physique intervient très fortement dans ce modèle) :

$$h_i(t) = \sum_j w_{ij} s_j(t) + \theta_i \quad (2.2)$$

où  $\theta_i$  est un seuil et la sortie  $s_j$  du neurone  $j$  est égale à une fonction non linéaire de l'entrée totale :

$$s_j(t+1) = \begin{cases} +1 & \text{si } h_j(t) > 0 \\ -1 & \text{si } h_j(t) < 0 \end{cases} \quad (2.3)$$

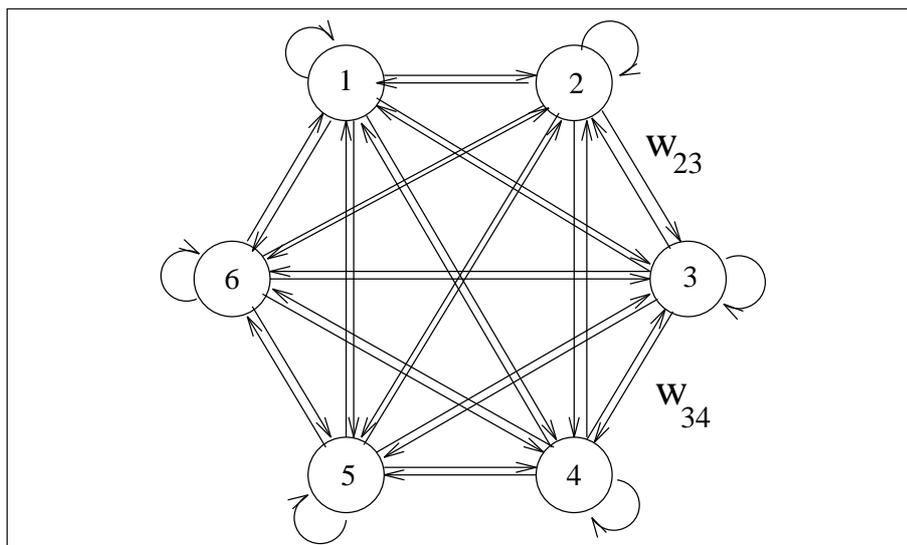


FIG. 2.1 – Un réseau tout connecté

Cette fonction non linéaire peut s'écrire sous une forme plus compacte car c'est la fonction signe( $\cdot$ ). On prendrait la fonction de Heaviside si les états actif et inactif des neurones étaient codés par les valeurs +1 et 0. Ce qui se passe en 0 est tout à fait arbitraire et pratiquement indifférent. Par souci d'esthétique on posera que le neurone reste dans l'état où il se trouvait à l'instant précédent (car cela rend la dynamique plus symétrique).

Cette mémoire associative est l'application la plus simple qui démontre le caractère « collectif » des calculs effectués par un réseau de neurones. C'est une mémoire de ce type qui a été réalisée pour le problème de base de données d'adresses cité plus haut. Avec succès.

## 2.3 Première approche : une mémoire hétéro-associative

### 2.3.1 Préliminaires : le neurone formel

C'est en 1943 qu'un mathématicien, Warren McCulloch, et un biologiste, Walter Pitts [MCP43], proposèrent une théorie générale de traitement de l'information basée sur des réseaux d'automates binaires, appelés de manière métaphorique « neurones » (il suffit de se reporter au chapitre sur la biologie pour voir la complexité des neurones biologiques ainsi que des réseaux qu'ils forment). Chacun de ces automates, encore appelé *cellule*, peut prendre des sorties désignées comme active (codé 1 par exemple) et comme inactive (codées 0 ou -1 par exemple). Afin de simuler le fonctionnement de neurones réels et en particulier l'existence de la période réfractaire, les auteurs supposent que le temps s'écoule de manière discrète  $t = 0, 1, 2, \dots$  et que l'état d'un neurone à l'instant  $t$  dépend de la somme pondérée des signaux provenant des autres neurones qu'il y a à l'entrée :

$$h_i(t) = \sum_j w_{ij} s_j(t) \quad (2.4)$$

Ici chaque élément  $w_{ij}$  représente la force d'interaction existant entre le neurone  $i$  et le neurone  $j$  ; on l'appelle aussi *efficacité synaptique* ou *poids* de la connexion correspondante. La quantité  $h_i$  est alors l'analogie du potentiel d'excitation postsynaptique des neurones biologiques. Si cette

valeur est supérieure à un certain seuil, la sortie du neurone passe à l'état actif, si non, à l'état inactif, au repos.

McCulloch et Pitts montrèrent<sup>1</sup> que des réseaux constitués de tels neurones pouvaient en principe imiter n'importe quelle machine discrète, comme peut le faire un ordinateur numérique convenablement programmé ou son abstraction mathématique, une machine de Turing.

### 2.3.2 Préliminaires : l'apprentissage de Hebb

Cette forme d'apprentissage, due à Hebb<sup>2</sup>, est en fait un postulat tiré de la biologie et est la plus connue et la plus vieille des règles d'apprentissage (1949) [Heb49].

Formulé dans un **contexte neurobiologique** il dit précisément :

When an axon of cell  $A$  is near enough to excite a cell  $B$  repeatedly or persistently takes part in firing it, some growth process or metabolic changes take place in one or both cells such that  $A$ 's efficiency as one of the cells firing  $B$ , is increased.

Que l'on peut traduire par :

Quand une cellule  $A$  excite par son axone une cellule  $B$  et que, de manière répétée et persistante elle participe à la genèse d'une impulsion dans  $B$ , un processus de croissance ou un changement métabolique à lieu dans l'une ou dans les deux cellules de sorte que l'efficacité de  $A$  à déclencher une impulsion dans  $B$  est, parmi les autres cellules qui ont cet effet, accrue.

Dans un **contexte RN** on transforme cette proposition en

1. si les 2 neurones situés de part et d'autre de la synapse sont actifs simultanément (synchrones) alors on augmente le poids,
2. si les 2 neurones ne sont pas synchrones, la synapse décroît ou est éliminée.

Une telle synapse est dite Hebbienne, elle dépend du temps, les interactions sont fortement locales. Les variations du poids se font grâce à un mécanisme de conjonction si on regarde les signaux, de corrélation si on parle en termes statistiques. (On dira aussi qu'une synapse est anti-hebbienne lorsque sa force (le poids) baisse pour des activités post – et pré – synaptiques corrélées et augmente sa force lorsque ces activités sont non corrélées.

Dans un **contexte mathématique** on pourrait poser (voir figure 2.2) :

$$\Delta w_{ij}^t = \eta s_i^t s_j^t \quad (2.5)$$

mais il y aurait variation exponentielle de  $w_{ij}$ . On ajoute alors un terme d'oubli :

$$\Delta w_{ij}^t = \eta s_i^t s_j^t - \alpha s_i^t w_{ij}^t \quad (2.6)$$

$$= \alpha s_i^t (c s_j^t - w_{ij}^t) \quad (2.7)$$

ce qui permet d'éviter les problèmes de convergence.

Dans un **contexte statistique** on pourrait poser :

$$\Delta w_{ij} = \eta \text{cov}(s_i^t, s_j^t) \quad (2.8)$$

$$= \eta E[(s_i^t - \bar{s}_i)(s_j^t - \bar{s}_j)] \quad (2.9)$$

$$= \eta \{E[s_i^t s_j^t] - \bar{s}_i \bar{s}_j\} \quad (2.10)$$

<sup>1</sup>Une volée de théorèmes récents étayent la démonstration : voir le chapitre concernant l'apprentissage statistique

<sup>2</sup>Donald Hebb était un psychologue canadien.

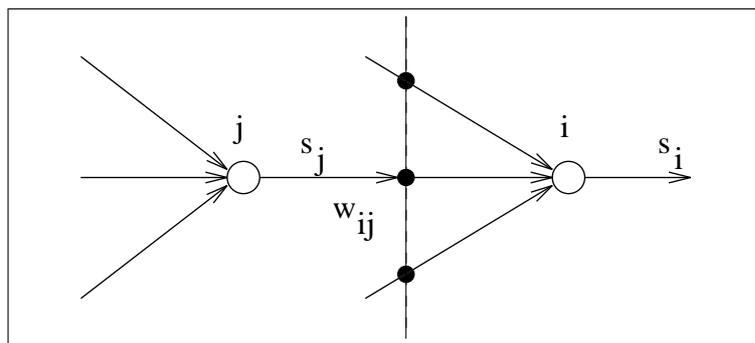


FIG. 2.2 – Interconnexion de deux neurones  $i$  et  $j$  par la synapse  $w_{ij}$ .

Le premier terme ressemble à la première proposition, le second terme  $\bar{s}_i \bar{s}_j$  est un seuil égal au produit des activités moyennes post et pré synaptiques.

Retour à un **contexte biologique** : il se pourrait qu'il y ait une plausibilité physiologique (modèle de l'hippocampe).

### 2.3.3 Objectifs et méthode

Il s'agit de mémoriser des formes binaires  $\in \{0, 1\}$  (ou encore  $\{-1, +1\}$ ) à  $N$  composantes et de les retrouver même si elles sont présentées bruitées.

Les sorties des neurones seront notées  $s_i = \pm 1$ , les poids seront notés  $w_{ij}$  et l'entrée totale,  $h_i$  sera la somme pondérée des entrées  $h_i = \sum_j w_{ij} s_j$ . L'état du réseau sera caractérisé par la donnée des  $N$  valeurs  $s_i$ .

On appellera prototypes les  $M$  états *stables* et *attracteurs* mémorisés, chaque motif prototype sera indicé par  $m$ .

On utilisera la règle de Hebb, encore appelée *prescription de Cooper* :  $w_{ij} = \sum_m s_i^m s_j^m$ .

### 2.3.4 Exercice

On se propose de réaliser l'hétéro association qui, au motif  $A = \{1, -1, -1, 1\}$  fait correspondre le motif  $A' = \{-1, -1, 1, 1\}$ .

Pour calculer les poids qui permettent cette association, on pose tout d'abord que chaque poids  $w_{ij}$  est égal à zéro. On place à l'entrée du réseau le motif  $A$  et à la sortie le motif  $A'$  (voir la figure 2.3). On calcule  $w_{ij}$  par la règle de Hebb :  $w_{ij} = A_i A'_j$ .

On obtient ainsi les poids  $+1/4$  ou  $-1/4$  qui sont représentés sur la figure 2.3. On peut ainsi vérifier, si on est pas convaincu de la procédure que si on met  $A$  en entrée, on obtient bien  $B$  en sortie.

Qu'en est-il du bruit ? Si on bruite un peu  $A$  en entrant  $\{1, -1, 0, 1\}$ , on vérifie que la sortie est bien  $A'$ , et cela se produit grâce à la fonction à seuil ; bien sûr la valeur « 0 » est un bruit particulier qui n'appartient pas à l'ensemble possible des entrées, mais même si on prend  $A = \{1, -1, 1, 1\}$  on obtient bien la sortie désirée (il faut cependant définir ce que vaut la sortie lorsque l'entrée totale est nulle).

Si l'hétéro association recherchée avait été  $B = \{1, -1, 1, -1\} \rightarrow B' = \{-1, 1, 1, -1\}$ , les poids auraient été ceux représentés sur la figure 2.4.

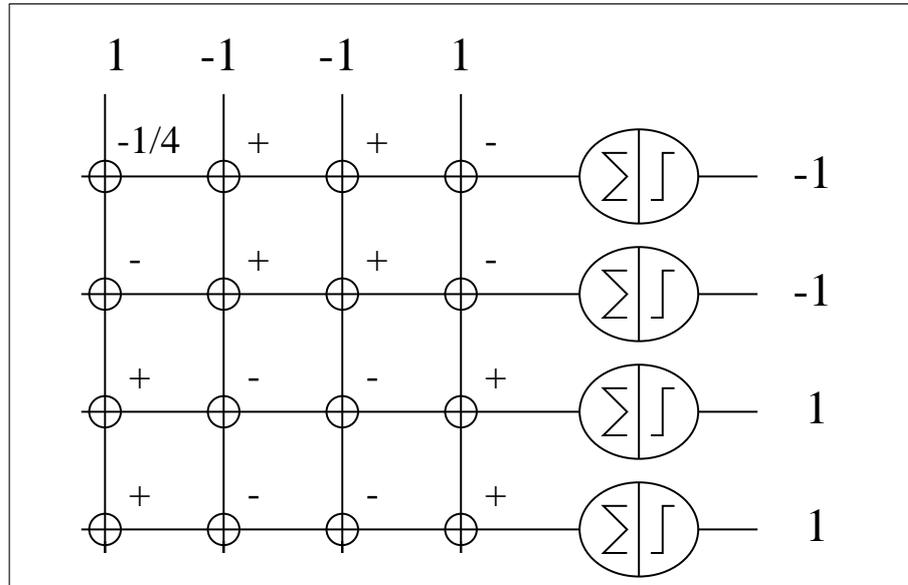


FIG. 2.3 – Le calcul des poids s’effectue en multipliant ce qu’il y a en entrée par ce qu’il y a en sortie. On normalise par le nombre de neurones pour que l’entrée totale soit égale, soit à +1, soit à -1.

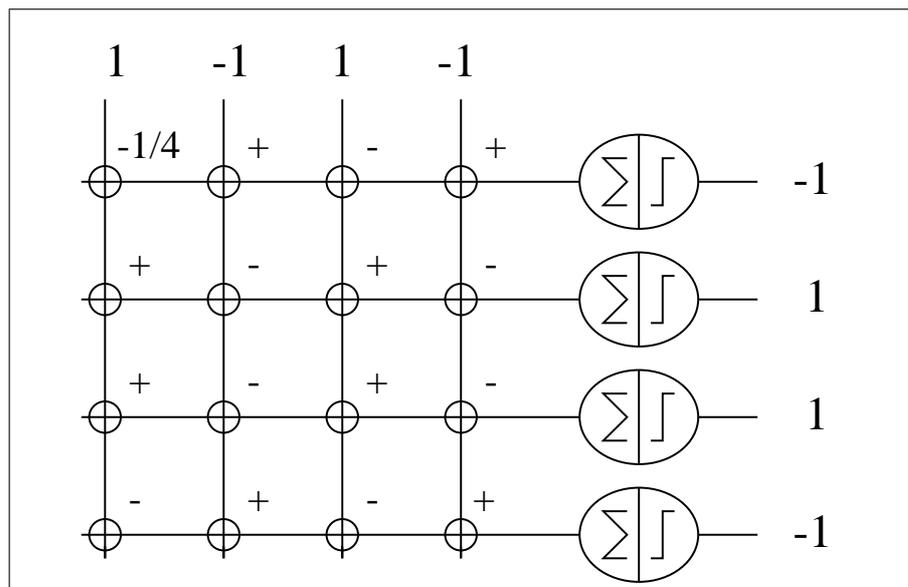


FIG. 2.4 – Le calcul des poids s’effectue en multipliant ce qu’il y a en entrée par ce qu’il y a en sortie. On normalise pour que l’entrée totale soit égale, soit à +1, soit à -1.

La règle de Hebb prescrit que pour mémoriser plusieurs motifs, il faut sommer, pour chaque connexion, les valeurs obtenues pour chacun des motifs. C’est ce qui a été réalisé sur la figure 2.5. On vérifiera que ce réseau reste insensible (dans une certaine mesure bien sûr) au bruit.

Jusqu’où peut-on aller dans la taille des motifs et dans la quantité de motifs ? Kohonen [Koh84] a réalisé une expérience impressionnante reportée sur la figure 2.6, il s’agit là d’homo association : le motif d’entrée est égal à celui de sortie mais la même procédure que précédemment (qui utilise la prescription de Cooper) a été utilisée.

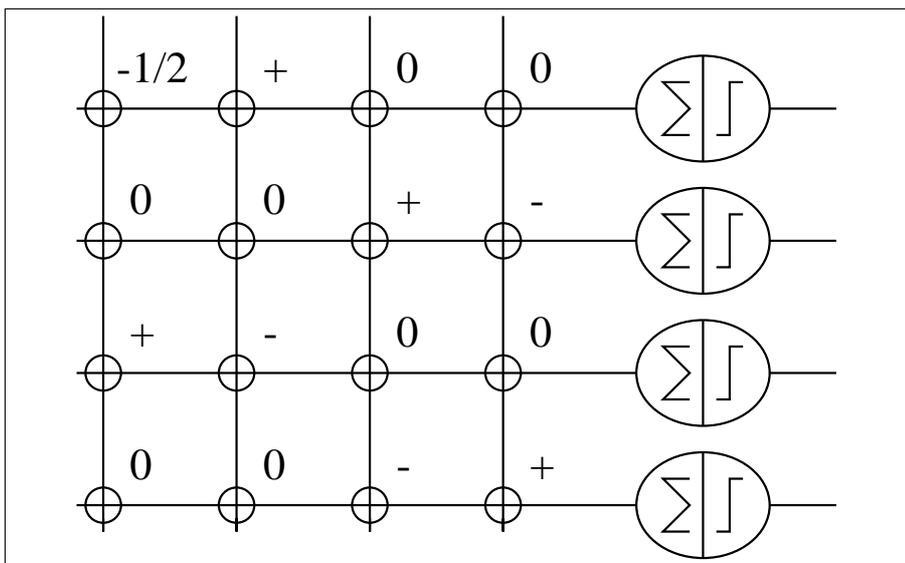


FIG. 2.5 – Le calcul des poids s’effectue en additionnant chacun des poids obtenus précédemment.

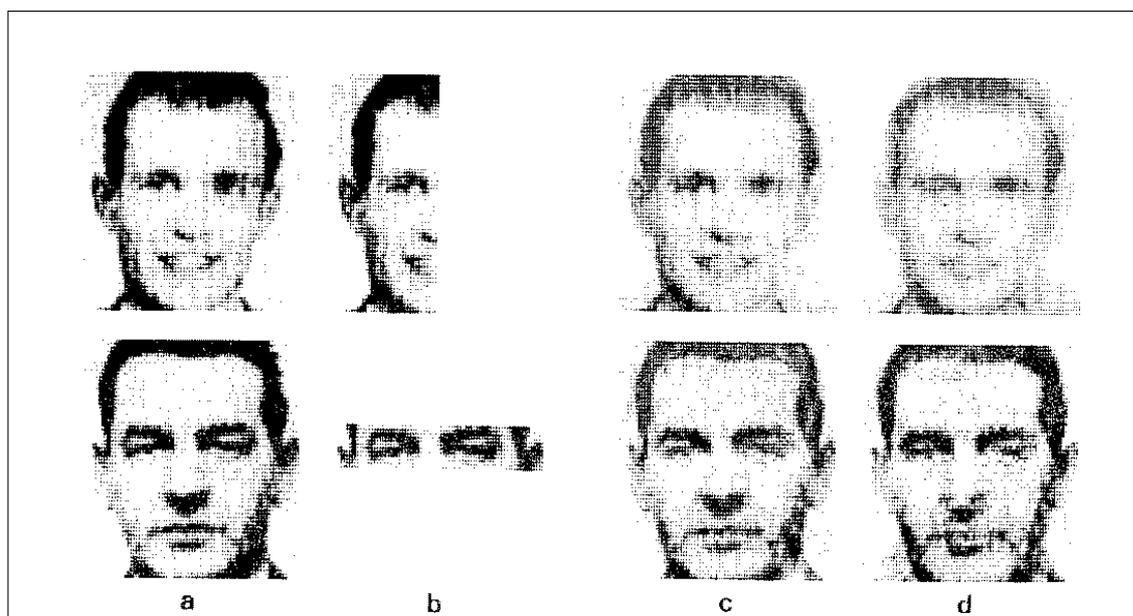


FIG. 2.6 – a : Deux des motifs stockés. b : motifs bruités correspondants. c : ce que l’on retrouve lorsque 160 images sont stockées dans la mémoire. d : ce que l’on retrouve lorsque la mémoire contient 500 images. Chaque image fait  $128 \times 128$  pixels.

Ne trouvez-vous pas le résultat remarquable ?

Il y a cependant un problème. C’est qu’une telle mémoire ne fonctionne plus du tout dès que l’image est translaturée ne serait-ce que d’un pixel. En effet le nouveau vecteur d’entrée ne ressemble plus du tout au vecteur de départ et la mémoire associative ne peut trouver un proche voisin.

Oublions cependant cet inconvénient et posons nous la question de savoir ce qui se passerait si on bouclait les sorties sur les entrées. À la première passe avec un motif bruité, la sortie devrait être plus proche de la « bonne » sortie, c’est ce que l’on voit dans la figure de Kohonen. À la seconde

passé, puisque le motif d'entrée s'est rapproché d'un motif mémorisé, la sortie devrait être encore meilleure ; et ainsi de suite jusqu'à, on l'espère, le motif prototype parfait.

C'est ce que nous allons voir dans la section suivante.

## 2.4 Seconde approche : une mémoire auto-associative récurrente

### 2.4.1 Préliminaires

Les neurones utilisés seront ceux de McCulloch et Pitts décrits plus haut.

On notera  $N$  le nombre de neurones et  $M$  le nombre de motifs que l'on veut mémoriser. Un motif sera représenté par un vecteur à  $N$  composantes, chacune étant l'état du neurone correspondant du réseau. Un motif mémorisé ou encore *prototype* sera noté  $\mathbf{m}^\mu$ ,  $\mu$  étant le numéro du motif. Un motif non prototype sera noté  $\mathbf{s}$ . Un état quelconque du réseau sera donc noté :

$$\mathbf{s} = [s_1, s_2, \dots, s_N]^T \quad (2.11)$$

où l'exposant  $T$  représente l'opération matricielle usuelle de transposition.

D'autre part, il s'avère plus judicieux de dessiner le réseau de la manière représentée sur la figure 2.7 plutôt que sous la forme d'un polygone régulier comme sur la figure 2.1.

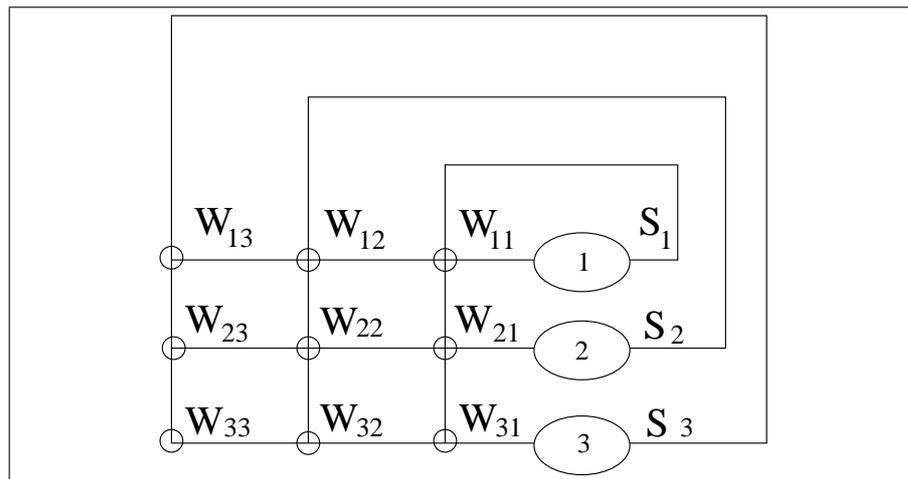


FIG. 2.7 – Une manière plus astucieuse de représenter le réseau de Hopfield.

Le fonctionnement de la mémoire associative doit être tel que si l'on part à l'instant initial d'un état du réseau représenté par  $\mathbf{s}$ , alors, après un certain nombre d'itérations le réseau doit se retrouver dans un état, mettons  $\mathbf{m}$ , celui qui est le plus proche (au sens de Hamming par exemple) de  $\mathbf{s}$ . C'est dans ce sens qu'on parle de mémoire adressable par le contenu et insensible (tout est relatif bien sûr) au bruit.

La *trajectoire* que suit le réseau est la liste, indexée par le temps, des états que prend le réseau. Sans perdre de vue que les états accessibles au réseau sont les sommets d'un cube de dimension  $N$ , il est instructif de dessiner la trajectoire de la façon qui est représentée sur la figure 2.8. Il s'agit d'une surface possédant un certain nombre de minimums (au moins les prototypes) et telle que, si l'on part d'un point quelconque on arrive dans un creux comme une bille qu'on laisserait rouler sous l'influence de la gravité (mais où il y aurait suffisamment de frottements pour ne pas dépasser le minimum une fois celui-ci atteint). Étudier la dynamique du réseau c'est étudier la

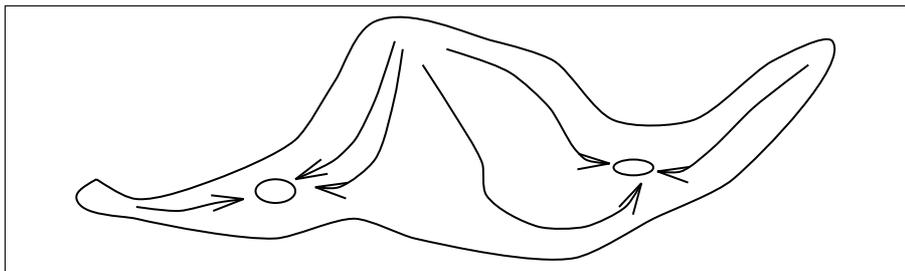


FIG. 2.8 – Un paysage énergétique avec deux motifs mémorisés attracteurs

trajectoire, les minimums sont les *points fixes* (que l'on voudrait prototypes). Avec cette image, la notion de *bassin d'attraction* est évidente.

### 2.4.2 Formalisation

Dans cette étude, les motifs prototypes ne seront pas des motifs spécifiques comme des images par exemple, mais des motifs tirés au hasard ou chaque composante  $m_i$  prendra de façon équiprobable les valeurs  $+1$  ou  $-1$ .

On ne perd pas de généralité en ne prenant pas en compte les seuils  $\theta_i$  dans l'équation (2.2) (car on s'occupera surtout de motifs aléatoires), si bien que l'on utilisera l'équation suivante pour l'actualisation des états :

$$h_i^t = \sum_j w_{ij} s_j^t \quad (2.12)$$

Il existe deux façons fondamentales d'appliquer l'équation (2.12) : soit de manière *synchrone* en calculant à chaque itération les états de tous les neurones en même temps, il s'agira alors du modèle de Little, soit de manière *asynchrone* en actualisant à chaque itération qu'un seul neurone tiré au hasard ou séquentiellement, c'est le modèle spécifique de Hopfield, celui que nous étudierons ici<sup>3</sup>.

### 2.4.3 Mémoriser un motif

Il s'agit donc de calculer quels sont les poids qui font que l'état qui correspond à ce motif  $\mathbf{m}$  est un point fixe de la dynamique du réseau (il est stable). Pour trouver ces poids écrivons la stabilité sous la forme :

$$\text{signe} \left( \sum_j w_{ij} m_j \right) = m_i \quad (2.13)$$

Une solution évidente est de prendre :

$$w_{ij} = \frac{1}{N} m_i m_j \quad (2.14)$$

car en effet on a  $\sum_i m_i^2 = \sum_i (\pm 1)^2 = N$ .

De plus nous voyons que si moins de la moitié des bits de l'état de départ sont faux ( $s_i \neq m_i$ ) alors leur effet dans le calcul de l'entrée totale

$$h_i = \sum_j w_{ij} s_j \quad (2.15)$$

<sup>3</sup>Il existe d'autres façons d'actualiser les états comme *asynchrone par blocs*, où, à chaque itération, un bloc de neurones est actualisé, tous les neurones du bloc en même temps.

sera plus faible que l'effet des bits justes, et donc que, après la fonction signe, on retrouvera exactement le motif  $\mathbf{m}$ .

Ce motif est un attracteur de la dynamique du réseau. Notons que le motif inverse  $-\mathbf{m}$  est lui aussi un attracteur, vers lequel le réseau converge lorsque plus de la moitié des bits sont faux.

#### 2.4.4 Mémoriser plusieurs motifs

En s'inspirant de la méthode précédente permettant de mémoriser un motif, on additionnera composante par composante, les quantités calculées à l'aide de l'équation (2.14) pour chacun des motifs et on obtiendra ainsi :

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^M m_i^\mu m_j^\mu \quad (2.16)$$

Cette équation ressemble à la prescription de Hebb, si ce n'est qu'ici, même si les unités ne sont pas actives ( $m_i = m_j = -1$ ), la contribution correspondante rentre dans le calcul du poids. Remarquons aussi que, par construction, la matrice de connexion  $\mathbf{W}$  est symétrique.

Examinons maintenant la condition de stabilité d'un motif particulier,  $\mathbf{m}^\mu$ , qui s'exprime par :

$$\text{signe}(h_i^\mu) = m_i^\mu \quad (\text{pour tout } i) \quad (2.17)$$

où l'entrée totale  $h_i^\mu$  vaut :

$$h_i^\mu = \sum_j w_{ij} m_j^\mu = \frac{1}{N} \sum_j \sum_\nu m_i^\nu m_j^\nu m_j^\mu \quad (2.18)$$

que l'on peut encore écrire de façon à sortir le terme correspondant à  $\mu$  :

$$\begin{aligned} h_i^\mu &= \frac{1}{N} \sum_\nu m_i^\nu \sum_j m_j^\nu m_j^\mu \\ &= \frac{1}{N} \left[ m_i^\mu \sum_j m_j^\mu m_j^\mu + \sum_{\nu \neq \mu} m_i^\nu \sum_j m_j^\nu m_j^\mu \right] \\ &= m_i^\mu + \frac{1}{N} \sum_{\nu \neq \mu} m_i^\nu \sum_j m_j^\nu m_j^\mu \end{aligned} \quad (2.19)$$

Si le deuxième terme, le terme croisé qui peut être assimilé à un terme de bruit, vaut 0 (cas des motifs orthogonaux par exemple), alors le motif  $\mu$  est stable bien évidemment, mais si ce terme est suffisamment petit pour ne pas changer le signe de  $h_i^\mu$  (si le nombre de motifs mémorisés  $M$  est faible par exemple),  $\mu$  sera quand même stable. Une configuration de départ proche (au sens de Hamming) d'un motif mémorisé convergera donc vers ce motif et le réseau se comportera bien comme une mémoire associative.

#### 2.4.5 Capacité de mémorisation

Examinons plus attentivement ce qui se passe lorsque les motifs ne sont pas orthogonaux mais sont *non corrélés*.

Nous supposons que les motifs prototypes sont aléatoires et issus de  $NM$  épreuves de Bernoulli. Le terme croisé de l'équation (2.19) contient exactement  $N(M-1)$  variables aléatoires indépendantes chacune pouvant prendre les valeurs  $\pm 1$ . Chaque variable aléatoire a une moyenne nulle

( $+1/N$  ou  $-1/N$  avec des probabilités identiques) et une variance de  $\sigma^2 = E(X^2) - (E(X))^2 = E(X^2) = 1/N^2$ .

Or, d'après le théorème limite central la somme d'un grand nombre,  $K$ , de variables aléatoires issues d'un processus aléatoire quelconque mais commun, est une distribution gaussienne centrée sur  $K$  fois la moyenne et dont la variance est  $K$  fois la variance de la distribution de probabilité d'origine.

Donc, le terme croisé est de moyenne nulle et de variance  $1/N^2 \times N(M-1) = (M-1)/N$ , autrement dit son ordre de grandeur est de  $\sqrt{(M-1)/N}$ .

D'autre part, la composante de signal dans l'équation (2.19) est  $m_i^\mu$ . Elle peut donc prendre les valeurs  $+1$  ou  $-1$  avec des probabilités égales, elle a donc une moyenne de 0 et une variance de 1. Le rapport signal à bruit, vaut alors :

$$\begin{aligned} \rho &= \frac{\text{variance du signal}}{\text{variance du bruit}} \\ &= \frac{N}{M-1} \\ &\approx \frac{N}{M} \quad \text{pour } M \text{ grand} \end{aligned} \quad (2.20)$$

Les motifs prototypes seront stables si et seulement si le rapport signal à bruit  $\rho$  est grand. Il ne faut donc pas que le nombre de motifs mémorisés soit trop grand pour que la mémoire associative fonctionne correctement (dans un sens probabiliste).

### 2.4.6 Capacité de stockage du réseau de Hopfield

C'est un euphémisme que de dire que l'étude de la capacité de stockage est quelque peu technique. Je ne donnerais ici que le résultat qui dit que, dans la limite thermodynamique ( $N$  grand) alors la mémoire associative ne fonctionne plus du tout lorsque  $M/N \rightarrow 0.138$ .

Cependant on peut très simplement se faire une idée des possibilités de mémorisation en calculant la probabilité pour que l'entrée totale  $h_i^\mu$  de l'équation (2.19) soit positive, sachant qu'un bit particulier vaut  $+1$ .

Or, la probabilité conditionnelle qu'un bit particulier provoque une erreur est donnée par la surface hachurée de la figure 2.9. Le reste de la surface sous la courbe représente la probabilité conditionnelle que le bit  $j$  du motif soit retrouvé correctement. Cette quantité est donnée par l'expression suivante, en utilisant toujours le théorème limite central :

$$\Pr[h_i^\mu > 0 \mid m_i^\mu = +1] = \frac{1}{\sqrt{2\pi\sigma^2}} \int_0^\infty \exp\left(-\frac{(h_i^\mu - moy)^2}{2\sigma^2}\right) dh_i^\mu \quad (2.21)$$

La moyenne de la variable aléatoire  $h_i^\mu$  est de 0, sa variance est  $\sigma^2 = (M-1)/N$

Dès qu'on a affaire à des distributions gaussiennes, on est amené, fréquemment, à utiliser la *fonction d'erreur*, erf(), définie par :

$$\text{erf}(y) = \frac{2}{\sqrt{\pi}} \int_0^y e^{-z^2} dz \quad (2.22)$$

où  $y$  est la variable qui définit la borne supérieure de l'intégration de la gaussienne.

On peut maintenant simplifier l'expression de la probabilité conditionnelle de retrouver le  $i^{\text{e}}$  bit du prototype en réécrivant l'équation (2.21) sous la forme :

$$\Pr[h_i^\mu > 0 \mid m_i^\mu = +1] = \frac{1}{2} \left[ 1 + \text{erf}\left(\sqrt{\frac{\rho}{2}}\right) \right] \quad (2.23)$$

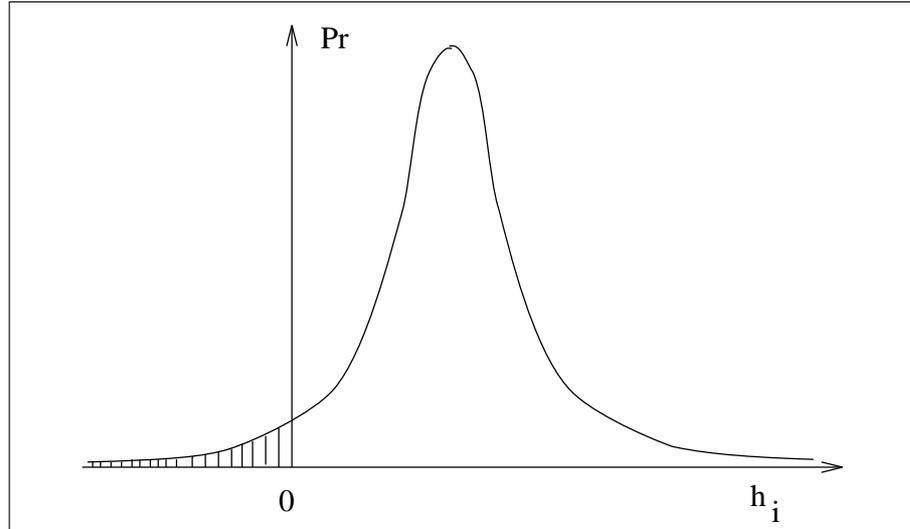


FIG. 2.9 – Distribution conditionnelle des valeurs du terme croisé de bruit, sachant qu'un bit est faux

$\rho$  étant le rapport signal à bruit défini par l'équation (2.20).

Chaque motif prototype comporte  $N$  bits et ils sont tous équiprobables et indépendants. La probabilité pour que le réseau soit stable pour tous les motifs est donc donnée par :

$$P_{stable} = (\Pr[h_i^\mu > 0 \mid m_i^\mu = +1])^N \quad (2.24)$$

Nous allons maintenant rechercher le nombre maximum de motifs que l'on peut mémoriser pour que la probabilité  $P_{stable}$  que le réseau soit stable soit égale à une valeur imposée, comme 0.99 par exemple.

Sachant que :

$$\lim_{y \rightarrow \infty} \text{erf}(y) = 1 - \frac{1}{\sqrt{\pi y^2}} e^{-y^2} \quad (2.25)$$

on peut récrire (2.23) :

$$\Pr[h_i^\mu > 0 \mid m_i^\mu = +1] = 1 - \frac{1}{\sqrt{2\pi\rho}} e^{-\frac{\rho}{2}} \quad (2.26)$$

ce qui permet d'écrire pour  $P_{stable}$  :

$$P_{stable} \approx \left(1 - \frac{1}{\sqrt{2\pi\rho}} e^{-\frac{\rho}{2}}\right)^N \quad (2.27)$$

$$\approx 1 - \frac{N}{\sqrt{2\pi\rho}} e^{-\frac{\rho}{2}} \quad (2.28)$$

Le second terme de cette équation doit être petit devant 1, lorsqu'on fait tendre  $N$  vers l'infini, pour que la probabilité moyenne que  $P_{stable}$  reste proche de 1. Ce qui veut dire qu'il faut :

$$\frac{1}{\sqrt{2\pi\rho}} e^{-\frac{\rho}{2}} < \frac{1}{N}$$

$$\text{qu'on peut récrire} \quad \rho > 2 \ln N + \frac{1}{2} \ln(2\pi\rho)$$

Afin d'assurer la stabilité de la plupart des motifs, Daniel Amit ([Ami89]) propose de prendre :

$$\rho_{min} = 2 \ln N \quad (2.29)$$

ce qui permet d'écrire l'équation de stabilité (2.27) simplement :

$$P_{stable} \approx 1 - \frac{1}{2\sqrt{\pi \ln N}} \quad (2.30)$$

Cette équation démontre que, dans la limite où le nombre de neurones tend vers l'infini, la probabilité qu'il existe un bit instable tend vers zéro. Il faut cependant que le rapport signal à bruit  $\rho$  ne soit pas inférieur au  $\rho_{min}$  défini par l'équation (2.29), ou, de façon équivalente, que le nombre de prototypes  $M$  ne dépasse pas une valeur critique  $M_{max}$ . En utilisant la définition de  $\rho$  (équation (2.20)) on trouve :

$$\begin{aligned} p_{max} &\approx \frac{N}{\rho_{min}} \\ &= \frac{N}{2 \ln N} \end{aligned} \quad (2.31)$$

Le résultat (2.31) définit la capacité de stockage d'un réseau de Hopfield lorsque l'on veut que la plupart des motifs prototypes soient parfaitement retrouvés.

On pourrait être plus exigeant et vouloir que tous les motifs soient parfaitement retrouvés. Le même Amit a montré que le nombre maximum de motifs mémorisés s'écrit alors :

$$p_{max} \approx \frac{N}{4 \ln N} \quad (2.32)$$

En conclusion, que se soit en utilisant l'équation (2.31) ou (2.32) la capacité asymptotique du réseau doit être maintenue faible pour que les motifs mémorisés soient retrouvés. Ce sera une limitation à l'utilisation du modèle de Hopfield.

### 2.4.7 Fonction énergie

Le plus grand mérite de Hopfield est probablement d'avoir étudié la dynamique de ses réseaux à l'aide des fonctions de Lyapunov<sup>4</sup>. Si vous voulez vous rafraîchir la mémoire et obtenir une vision formelle des systèmes dynamiques, vous trouverez dans les Annexes quelques vitamines.

La propriété essentielle des fonctions de Lyapunov est qu'elle décroît au cours de l'évolution du système dynamique et qu'elle est bornée. En physique elle est similaire à l'énergie.

#### Cas d'une dynamique asynchrone

Posons :

$$E = -\frac{1}{2} \sum_{i,j} w_{ij} s_i s_j \quad (2.33)$$

et montrons que c'est une fonction de Lyapunov du réseau.

- Tout d'abord, elle représente bien le réseau puisqu'elle contient tous ses paramètres : tous les poids et tous les états.

<sup>4</sup>Il y a quantité de manières de transcrire l'alphabet cyrillique, vous trouverez certainement d'autres orthographes.

• Ensuite, elle est bornée car :

$$\sum_{i,j} w_{ij} s_i s_j \leq \sum_{i,j} |w_{ij}| \quad (2.34)$$

• Montrons maintenant qu'elle est décroissante et converge vers un minimum. Pour cela il faut étudier le signe de  $\Delta = -2(E^{t+1} - E^t)$ . Si  $\Delta$  est positif c'est que  $E$  décroît au cours de l'évolution temporelle du réseau. Souvenons nous aussi que nous nous sommes placés en dynamique asynchrone et que donc à chaque pas de temps  $t$  un seul neurone, mettons le numéro  $i_0$ , voit sa sortie actualisée et qu'on a donc  $s_i^{t+1} = s_i^t \quad \forall i \neq i_0$ .

$$\Delta = \sum_{i,j} w_{ij} s_i^{t+1} s_j^{t+1} - \sum_{i,j} w_{ij} s_i^t s_j^t \quad (2.35)$$

Nous allons décomposer cette somme selon quatre termes, un premier terme correspondant à  $i = j \neq i_0$ , un terme pour  $i = i_0$  et  $j$  quelconque, un troisième terme pour  $i$  quelconque et  $j = i_0$ , un quatrième terme pour  $i = j = i_0$  :

$$\Delta = \sum_{i \neq i_0, j \neq i_0} (s_i^{t+1} s_j^{t+1} - w_{ij} s_i^t s_j^t) \quad (2.36)$$

$$+ \sum_{j \neq i_0} w_{i_0 j} (s_{i_0}^{t+1} s_j^{t+1} - s_{i_0}^t s_j^t) \quad (2.37)$$

$$+ \sum_{i \neq i_0} w_{i i_0} (s_i^{t+1} s_{i_0}^{t+1} - s_i^t s_{i_0}^t) \quad (2.38)$$

$$+ w_{i_0 i_0} [(s_{i_0}^{t+1})^2 - (s_{i_0}^t)^2] \quad (2.39)$$

Du fait de la dynamique asynchrone le premier terme est nul. Les deuxième et troisième lignes sont presque identiques, les indices  $i$  et  $j$  étant muets, à ceci près que dans un cas on a  $w_{i_0 j}$  et dans l'autre  $w_{i i_0}$ . En fait pour aller plus loin nous sommes obligés de poser que ces deux termes sont égaux, ce qui revient à dire que la matrice de connexion  $\mathbf{W}$  est symétrique. Dans ce cas l'équation précédente se ramène à :

$$\Delta = 2(s_{i_0}^{t+1} - s_{i_0}^t) \sum_{j \neq i_0} w_{i_0 j} s_j^t \quad (2.40)$$

$$+ w_{i_0 i_0} [(s_{i_0}^{t+1})^2 - (s_{i_0}^t)^2] \quad (2.41)$$

Examinons maintenant la première ligne :

$$(s_{i_0}^{t+1} - s_{i_0}^t) \sum_{j \neq i_0} w_{i_0 j} s_j^t = (s_{i_0}^{t+1} - s_{i_0}^t) \sum_j w_{i_0 j} s_j^t - (s_{i_0}^{t+1} - s_{i_0}^t) w_{i_0 i_0} s_{i_0}^t \quad (2.42)$$

Après quelques arrangements  $\Delta$  devient :

$$\Delta = 2(s_{i_0}^{t+1} - s_{i_0}^t) h_{i_0}^t + (s_{i_0}^{t+1} - s_{i_0}^t)^2 w_{i_0 i_0} \quad (2.43)$$

Le deuxième terme est toujours du signe de  $w_{i_0 i_0}$ . Quant au premier, examinons dans le tableau suivant les quatre cas possibles :

$h_{i_0}^t$	$> 0$	$> 0$	$< 0$	$< 0$
$s_{i_0}^t$	$+1$	$-1$	$+1$	$-1$
$s_{i_0}^{t+1}$	$+1$	$+1$	$-1$	$-1$
$s_{i_0}^{t+1} - s_{i_0}^t$	$0$	$2$	$-2$	$0$
$(s_{i_0}^{t+1} - s_{i_0}^t) h_{i_0}^t$	$0$	$> 0$	$> 0$	$0$

Le premier terme de 2.43 est donc toujours positif ou nul. Il est nul quand  $s_{i_0}^{t+1} = s_{i_0}^t$ , c'est-à-dire quand nous obtenons un *point fixe*.

Nous obtenons donc le théorème suivant :

**Théorème 1** *En dynamique asynchrone, si la matrice des connexions est symétrique et si les termes diagonaux sont non négatifs, alors le réseau de Hopfield converge vers un point fixe.*

On peut considérer que c'est un résultat surprenant : prenez une matrice de connexion quelconque mais respectant cependant les conditions énoncées, partez de n'importe quel état, le réseau arrivera dans un état stable en un temps fini !

### Cas d'une dynamique synchrone

Dans le cas où la dynamique est synchrone la fonction énergie suivante est une fonction de Lyapunov :

$$E = -\frac{1}{2} \sum_i \sum_j w_{ij} s_i^{t+1} s_j^t \quad (2.44)$$

Comme dans le cas asynchrone, posons  $\Delta = -2(E^{t+1} - E^t)$  et montrons que nous avons toujours  $\Delta \geq 0$  :

$$\Delta = \sum_{i,j} w_{ij} s_i^{t+2} s_j^{t+1} - \sum_{i,j} w_{ij} s_i^{t+1} s_j^t \quad (2.45)$$

$$= \sum_i s_i^{t+2} \sum_j w_{ij} s_j^{t+1} - \sum_j s_j^t \sum_i w_{ij} s_i^{t+1} \quad (2.46)$$

$$= \sum_i s_i^{t+2} \sum_j w_{ij} s_j^{t+1} - \sum_i s_i^t \sum_j w_{ji} s_j^{t+1} \quad (2.47)$$

$$\text{qui se simplifie seulement si } w_{ij} = w_{ji} : \quad (2.48)$$

$$= \sum_i (s_i^{t+2} - s_i^t) h_i^{t+1} \quad (2.49)$$

Le signe de  $\Delta$  s'étudie comme dans le cas précédent en analysant les différentes possibilités :

$h_i^{t+1}$	> 0	> 0	< 0	< 0
$s_i^t$	+1	-1	+1	-1
$s_i^{t+2}$	+1	+1	-1	-1
$s_i^{t+2} - s_i^t$	0	2	-2	0
$(s_i^{t+2} - s_i^t) h_i^{t+1}$	0	> 0	> 0	0

$\Delta$  est donc toujours  $\geq 0$ , l'énergie  $E$  décroît donc toujours au cours de l'évolution temporelle du réseau.  $E$  est bien une fonction de Lyapunov. Cette énergie est minimale pour  $s_i^{t+2} = s_i^t$ , ce qui ne peut se produire que si :

- $s_i^t = \text{Constante}$  et on a affaire à un point fixe.
- $s_i^{t+2} = s_i^t$  avec par exemple :  $s_i^t = 1 \rightarrow -1 \rightarrow 1 \rightarrow -1 \dots$   
et on a donc affaire à un cycle d'ordre 2.

D'où le théorème :

**Théorème 2** *En dynamique synchrone, si la matrice des connexions est symétrique, alors le réseau de Hopfield converge vers un point fixe ou un cycle limite d'ordre deux.*

### Profondeur des bassins d'attraction

En dynamique asynchrone, si les états des neurones  $s_i$  correspondent aux composantes  $m_i$  des motifs, alors l'énergie du réseau lorsqu'il est dans cet état vaut  $E$  telle que :

$$E = -\frac{1}{2} \sum_{i,j} w_{ij} m_i m_j \quad (2.50)$$

$$= -\frac{1}{2} \sum_i m_i \sum_j w_{ij} m_j \quad (2.51)$$

$$= -\frac{1}{2} \sum_i m_i m_i \quad (2.52)$$

$$= -\frac{1}{2} \sum_i m_i^2 \quad (2.53)$$

$$= -\frac{N}{2} \quad (2.54)$$

$$(2.55)$$

Chaque motif correspond donc à un puits d'énergie dont la profondeur est la même quel que soit le motif.

### 2.4.8 États spurieux

Spurieux est un terme tiré de l'anglais *spurious* qui signifie « faux », « bâtard ». Le réseau de Hopfield possède en effet des minimums stables pour des états qui ne correspondent pourtant pas à des motifs prototypes. Il en existe de différents types :

- Le premier type se trouve immédiatement en remarquant que le minimum de la fonction énergie est atteint pour des motifs opposés tels que  $\mathbf{m}' = -\mathbf{m}$  et puisque la matrice des connexions est symétrique. En soit ce n'est pas rédhibitoire puisqu'on peut toujours rajouter à chaque motif un bit de signe qu'il suffit d'examiner pour savoir si on a affaire à un motif ou à son négatif.
- Il existe aussi des *mélanges stables d'états* qui correspondent à une combinaison linéaire d'un nombre impair de motifs [AGS85]. La combinaison la plus simple est celle de trois motifs :

$$m_i^{mix} = \text{signe}(\pm m_i^{\mu_1} \pm m_i^{\mu_2} \pm m_i^{\mu_3}) \quad (2.56)$$

Un simple dénombrement des différents cas possible montre que, 3 fois sur 4,  $m_i^{mix}$  a le même signe que  $m_i^{\mu_1}$  ou encore, ce qui revient au même, que  $m_i^{mix}$  est en moyenne à la distance de  $N/4$  de  $m_i^{\mu_1}$  (ou, bien sûr, de  $m_i^{\mu_2}$  ou de  $m_i^{\mu_3}$ ). Ceci implique qu'en moyenne on a  $\sum_i m_i^{mix} m_i^{\mu_1} = N/4 \times (-1) + (N - N/4) \times (+1) = N/2$ .

Si tous les signes de 2.56 sont positifs alors on peut écrire :

$$h_i^{mix} = \sum_j w_{ij} m_j^{mix} = \frac{1}{N} \sum_{\mu} \sum_j m_j^{mix} m_i^{\mu} m_j^{\mu} = \frac{1}{2} m_i^{\mu_1} + \frac{1}{2} m_i^{\mu_2} + \frac{1}{2} m_i^{\mu_3} + \text{termes croisés} \quad (2.57)$$

Ainsi la condition de stabilité est aussi satisfaite pour un état de mélange comprenant un nombre impair de motifs. Si ce nombre était pair, en certains sites des combinaisons pourraient s'annuler.

- Il existe aussi d'autres états spurieux qui ne correspondent à aucune combinaison d'un nombre fini d'états. Ces états sont appelés *états de verre de spin* car ils ont une correspondance étroite avec les modèles de verre de spins que l'on rencontre en physique ; nous verrons (de très loin) ce que cela signifie dans la section suivante.

Il existe un certain nombre de trucs et d'astuces qui permettent de ne pas tomber dans ces minimums mais je ne les évoquerai que succinctement plus loin dans les paragraphes traitant des dynamiques à température finie et du champ moyen.

### 2.4.9 Réseau de Hopfield et physique des verres de spins

Un réseau de Hopfield comporte donc un grand nombre de neurones identiques qui peuvent prendre deux états différents. Cela est proche de certains modèles que l'on rencontre en physique statistique des matériaux magnétiques.

Considérons  $N$  atomes identiques organisés en réseau maillé. Chaque atome possède un spin et un moment magnétique. Dans le *modèle des spins d'Ising* chaque spin ne peut prendre que une des deux valeurs  $\pm 1$  et chaque atome interagit avec ses voisins. Si l'interaction entre spins possède une longue portée et prend des valeurs aléatoires on parle de *modèle de verre de spins*.

La dynamique de ces matériaux se décrit dans l'espace des phases engendré par les différents degrés de liberté. L'état interne du système est représenté par les variables  $s_i$   $i = 1, 2, \dots, N$ . Afin que le système puisse évoluer vers un état d'équilibre, un mécanisme, non spécifié, permet aux spins de basculer suivant les règles suivantes :

1. À chaque atome  $i$  on associe un champ local  $h_i$ .
2. Le champ local est une combinaison linéaire des états internes des autres spins :

$$h_i = \sum_j J_{ij} s_j \quad (2.58)$$

où les  $J_{ij}$  sont des *constantes de couplage*.

3. Un état interne  $s_i$  peut basculer (on dit souvent *flipper* de l'anglais *flip*) s'il satisfait à la *condition d'alignement* :

$$h_i s_i > 0 \quad (2.59)$$

Il existe aussi un Hamiltonien qui représente l'énergie d'interaction entre les différents spins du système. Il est défini par :

$$H = -\frac{1}{2} \sum_i h_i s_i = -\frac{1}{2} \sum_i \sum_j J_{ij} s_i s_j \quad (2.60)$$

où le facteur  $1/2$  est introduit parce que dans la double sommation l'énergie d'interaction entre le spin  $i$  et le spin  $j$  est comptée en même temps que l'énergie d'interaction entre le spin  $j$  et le spin  $i$ .

Les termes de couplage sont des forces et dans la nature les forces sont symétriques si bien que l'on a :

$$J_{ij} = J_{ji} \quad (2.61)$$

Ces termes de couplages sont aléatoires.

Si nous comparons ces équations à celles développées dans le réseau de Hopfield, nous trouvons un *isomorphisme* flagrant. C'est parce qu'une armée de physiciens avait travaillé sur les verres de spins pendant de nombreuses années que le modèle de Hopfield (ainsi que d'autres qui lui sont apparentés) a été aussi intensément étudié. Et les travaux ont continué même après qu'il est apparu que les performances de telles machines étaient relativement limitées.

### Dynamique à température finie

Dans les milieux magnétiques, lorsque la température n'est pas très basse <sup>5</sup>, des fluctuations thermiques tendent à faire basculer les spins dans le sens opposé à celui dans lequel ils se trouvent. Il y a alors compétition entre le champ qui tend à aligner le spin et la température qui tend à les désaligner.

La dynamique des verres de spins d'Ising à température finie est décrite par la dynamique de Glauber [Gla63]. On remplace la loi déterministe précédente par la loi probabiliste :

$$s_i(t+1) = \begin{cases} +1 & \text{avec la probabilité } g(h_i) \\ -1 & \text{avec la probabilité } 1 - g(h_i) \end{cases} \quad (2.62)$$

Il y a plusieurs possibilité quant au choix de  $g(h)$ . Nous utiliserons ici la fonction de Glauber simple, de forme sigmoïdale :

$$g(h) = f_\beta(h) = \frac{1}{1 + \exp(-2\beta h)} \quad (2.63)$$

$\beta$  est la notation classique de thermodynamique statistique pour la température :  $\beta = 1/(kT)$  où  $k$  est la constante de Boltzmann.

En remarquant que :

$$1 - f_\beta(h) = f_\beta(-h) \quad (2.64)$$

on peut récrire 2.62 de la façon suivante, symétrique :

$$\Pr[s - i = \pm 1] = f_\beta(\pm h_i) = \frac{1}{1 + \exp(\mp 2\beta h_i)} \quad (2.65)$$

La température joue sur la pente de la fonction de transfert. Plus elle est faible, plus la pente est grande. À la limite d'une température nulle, la pente est infinie et on retrouve la fonction de transfert de l'équation 2.3. D'autre part, lorsque la température tend vers l'infini, la prochaine valeur de  $s_i$  est complètement indéterminée.

### Champ moyen

Supposons pour commencer, que nous ayons un seul spin  $s$  dans un champ local  $h$ . Alors la valeur moyenne de  $s$  (ou magnétisation moyenne) s'écrit :

$$\langle s \rangle = \Pr[s = +1] \times (+1) + \Pr[s = -1] \times (-1) \quad (2.66)$$

$$= \frac{1}{1 + \exp(-2\beta h)} - \frac{1}{1 + \exp(2\beta h)} \quad (2.67)$$

$$= \frac{\exp(\beta h)}{\exp(\beta h) + \exp(-\beta h)} - \frac{\exp(-\beta h)}{\exp(-\beta h) + \exp(\beta h)} \quad (2.68)$$

$$= \tanh(\beta h) \quad (2.69)$$

Lorsque le champ varie de  $-\infty$  à  $+\infty$ , la magnétisation moyenne passe de -1 à +1, l'essentiel de la transition s'effectuant dans un intervalle de  $kT$  autour de zéro. Rappelons que, à chaque instant le spin est soit à +1 soit à -1. Du fait de la température, il reste plus ou moins *longtemps* aligné avec le champ.

<sup>5</sup>C'est dans ce sens qu'on entend « température finie », c'est-à-dire non égale à zéro.

Dans les systèmes *paramagnétiques* les spins n'ont aucune influence les uns sur les autres. Supposons qu'ils soient soumis au même champ externe. Alors la magnétisation totale de  $N$  spins étant  $M = N \langle s \rangle$ , on en déduit la loi de Curie :  $\partial M / \partial h = 1/T$  à  $h = 0$ .

Lorsqu'il y a plusieurs spins en interaction et dans un champ externe, la valeur du champ local fluctue du fait de la température et vaut :  $h_i = \sum_j w_{ij} s_j + h^{ext}$ . Le problème du calcul de la magnétisation n'est plus aussi facile. On s'en tire néanmoins en faisant une approximation qui porte un nom célèbre, celle du *champ moyen*, qui consiste à remplacer la vraie valeur de  $h$  par sa moyenne :

$$\langle h_i \rangle = \sum_j w_{ij} \langle s_j \rangle + h^{ext} \quad (2.70)$$

et la valeur moyenne de la magnétisation vaut :

$$\langle S_i \rangle = \tanh(\beta \langle h_i \rangle) = \tanh \left( \beta \sum_j w_{ij} \langle s_j \rangle + \beta h^{ext} \right) \quad (2.71)$$

Il y a toujours  $N$  équations non linéaires à  $N$  inconnues, mais au moins il n'y a plus de variables stochastiques !

Cette théorie du champ moyen, qui est beaucoup utilisée en théorie des réseaux de neurones « à la Hopfield », devient exacte dans la limite où la portée des interactions devient infinie, c'est-à-dire où les neurones agissent tous, les uns sur les autres. En gros cela se simplifie parce que  $h$  devient la somme d'un nombre infini de termes et qu'alors on peut appliquer le théorème central limite. Néanmoins, même pour des interactions à courte portée où  $w_{ij} \approx 0$  dès que les spins sont éloignés de quelques pas, la théorie du champ moyen peut donner une bonne description des phénomènes.

### 2.4.10 Prototypes corrélés : la pseudo-inverse

Le terme croisé de l'équation 2.19 provenait du fait que si on tirait un grand nombre de motifs aléatoires, alors ils n'étaient pas tous orthogonaux. Posons nous la question de savoir comment calculer la matrice de connexion lorsque les motifs sont linéairement indépendants, ce qui est une condition moins restrictive que celle d'orthogonalité.

Si nous posons le problème sous forme matricielle, nous aurons à résoudre, dans le cas de l'hétéro association :

$$\mathbf{Y} = \mathbf{W}\mathbf{X} \quad (2.72)$$

où les vecteurs de sorties sont les colonnes de  $\mathbf{Y}$ , les vecteurs d'entrée, les colonnes de  $\mathbf{X}$ .

$\mathbf{Y}$  et  $\mathbf{X}$  sont toutes deux des matrices à  $N$  lignes (le nombre de neurones) et à  $M$  colonnes (le nombre de motifs). Si elle étaient carrées le calcul de  $\mathbf{W}$  (matrice  $N \times N$ ) serait facile (formellement  $\mathbf{W} = \mathbf{Y}\mathbf{X}^{-1}$ ). Mais ce n'est pas le cas,  $\mathbf{X}$  n'étant pas carrée.

Il faut avoir recours à  $\mathbf{X}^+$ , la pseudo-inverse de Moore-Penrose [Pen55] qui permet d'écrire :

$$\mathbf{W} = \mathbf{Y}\mathbf{X}^+ + \mathbf{Z}(\mathbf{1} - \mathbf{X}\mathbf{X}^+) \quad (2.73)$$

Fort heureusement Penrose prétend que n'importe quelle matrice  $\mathbf{Z}$  peut faire l'affaire, prenons donc  $\mathbf{Z} = \mathbf{0}$  et revenons au cas de l'auto association. Nous trouvons :

$$\mathbf{W} = \mathbf{X}\mathbf{X}^+ \quad \text{avec} \quad \mathbf{X}^+ = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \quad (2.74)$$

$\mathbf{W}$  est la matrice de projection orthogonale dans le sous-espace engendré par les vecteurs prototypes. Elle peut être calculée grâce à l'algorithme itératif de Gréville [Gre60] où il y a autant

d'itérations qu'il y a de motifs. Numériquement ce n'est pas une méthode très efficace dès que la dimension augmente. Il est préférable d'utiliser une méthode de gradient stochastique qui minimise la norme de  $\| \mathbf{X} - \mathbf{W}\mathbf{X} \|$ .

Beaucoup de techniques liées aux matrices et aux pseudo-inverses se trouvent dans le premier livre de Kohonen[Koh78] qui traite des mémoires associatives linéaires.

## 2.5 Résultats de simulation

Une simulation tirée du livre de Weisbuch[Wei89] est représentée sur la figure 2.10 où chaque

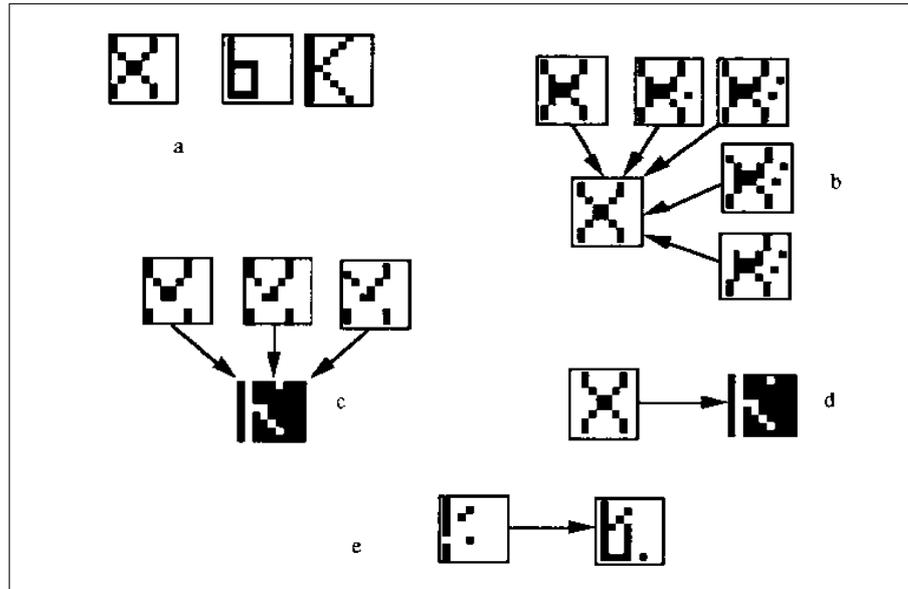


FIG. 2.10 – Exemples de convergence de configurations vers les attracteurs d'un système de Hopfield.

carré comprend  $8 \times 8$  cases (ou pixels) représentant chacune l'état d'un automate (les cases noires figurent les automates à l'état 1, les cases blanches ceux à l'état  $-1$ ). Les flèches partent des motifs pris comme configurations initiales et pointent vers les attracteurs correspondants. Les trois cases constituant la figure (a) représentent les lettres X, b, K qui ont été choisies comme prototypes. Elles ont donc déterminé le choix des connexions  $W_{ij}$  du réseau et sont invariantes dans la dynamique. On voit que de petites erreurs dans le motif X sont corrigées par la dynamique (b). D'autres, plus importantes ne le sont pas (c). La translation d'un motif conduit vers un autre attracteur (d). On note l'existence d'un attracteur mélange des deux motifs b et K (e).

## 2.6 Implémentations optiques

Les implémentations optiques de mémoires adressables par le contenu ou de dispositifs similaires ont été étudiées depuis 1960. La majorité des réseaux de neurones optiques réalisés tombent dans une des deux catégories : multiplication vecteur-matrice ou corrélateurs holographiques. La figure 2.11 en montre une version [GT85].

Psaltis et Farhat, du Caltech sont deux auteurs ayant extensivement utilisés des dispositifs optiques à fin de calcul optique ; la figure 2.12 tirée de [PF85] montre deux de leurs nombreuses réalisations.

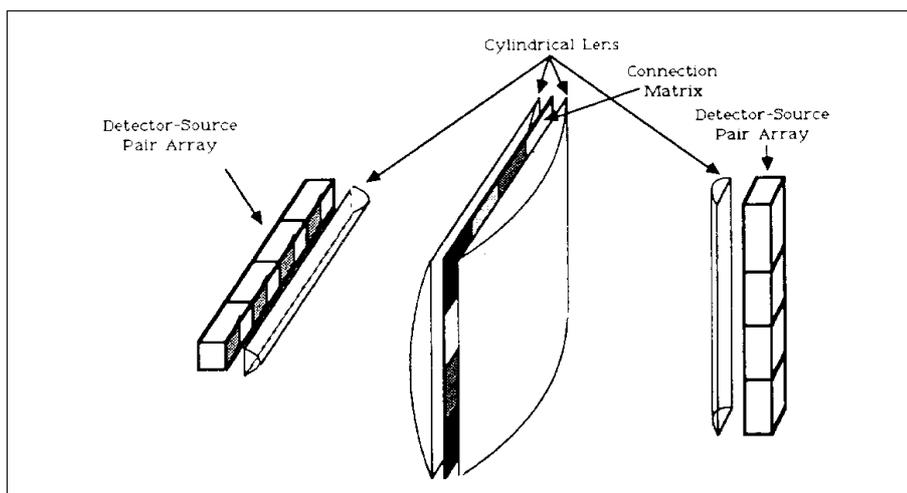


FIG. 2.11 – Réalisation d’une mémoire associative basée sur une multiplication optique vecteur-matrice. La lumière peut passer au travers de la matrice des connexions dans les deux sens.

## 2.7 Troisième approche : réseaux de Hopfield et Optimisation

L’essentiel de ce qui se dira par la suite est guidé par le fait que les applications en optimisation sont réalisées avec des amplificateurs opérationnels (AO), des dispositifs électroniques qui possèdent des gains, des courants de polarisation ou des courants de fuite . . . À moins que ce ne soit l’intuition que les modèles de Hopfield pourraient être utilisés en optimisation qui a conduit à rajouter au modèle original des grandeurs qui se trouvent avoir une interprétation simple en électronique et ainsi de potentiellement permettre une réalisation matérielle efficace. Vous connaissez l’histoire de la poule et de l’œuf. Quoi qu’il en soit Hopfield et Tank [TH86] ont parlé en premier d’AO.

On voit bien qu’un neurone formel est proche d’un AO, mais on voit aussi que le gain d’un AO ne peut pas être infini et ne pourra donc pas être ramené à la fonction de transfert d’un neurone tel qu’on l’a vu dans ce chapitre. Comment va-t-on s’en sortir ?

### 2.7.1 Modifications du modèle original

Nous modifierons le champ local (l’entrée totale) donné par l’équation 2.58 en lui ajoutant un courant externe de la façon suivante :

$$H_i = \sum_{j \neq i} T_{ij} V_j + I_i \quad (2.75)$$

Cet ajout est représenté sur la figure 2.13.

L’expression de l’énergie en découle assez naturellement :

$$E = -\frac{1}{2} \sum_i \sum_{j \neq i} T_{ij} V_i V_j - \sum_i I_i V_i \quad (2.76)$$

Il est possible de démontrer qu’il s’agit d’une fonction de Lyapunov et que l’on a :

$$\Delta E = \left[ \sum_j T_{ij} V_j + I_i \right] \Delta V_i \leq 0 \quad (2.77)$$

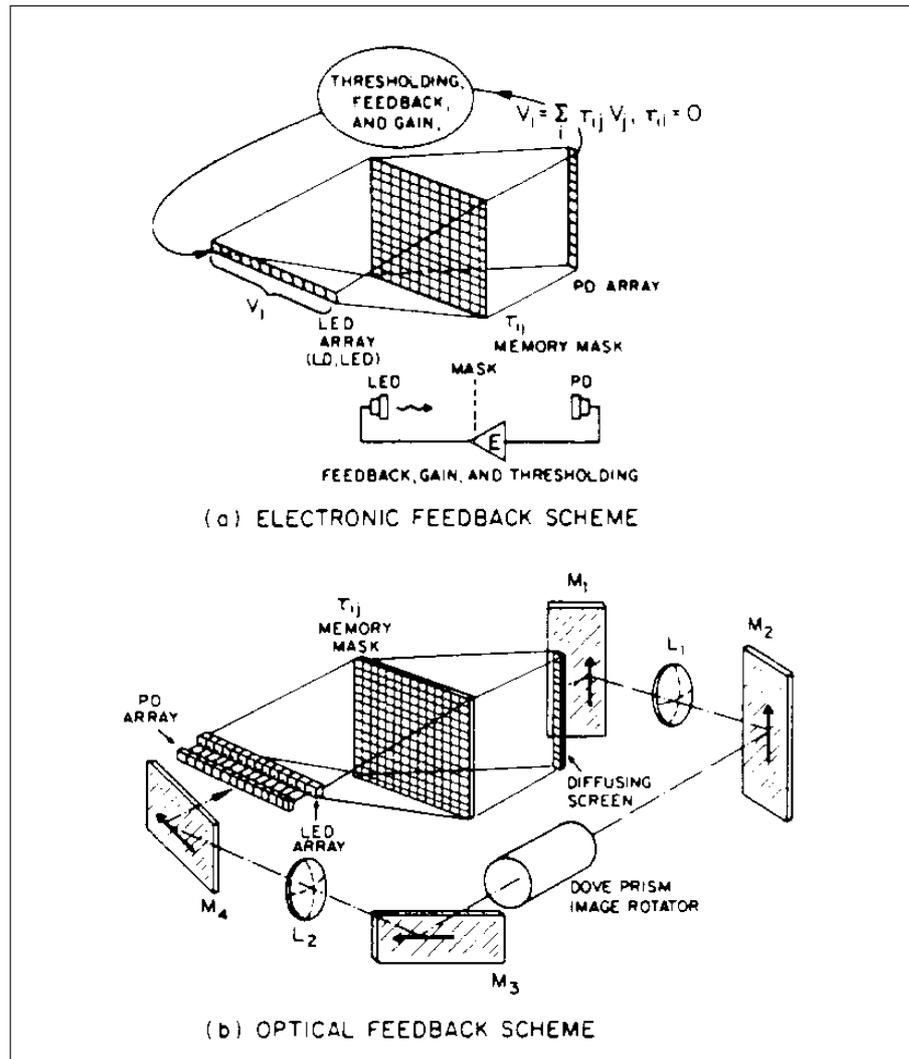


FIG. 2.12 – Deux façons de réaliser la rétroaction non-linéaire : en haut, électronique, en bas, tout optique.

### 2.7.2 Modèle continu

Nous allons voir ce qui se passe si nous utilisons de véritables AO pour réaliser les neurones du réseau de Hopfield représenté sur la figure 2.13 où un courant externe est ajouté à chaque entrée de neurone. Posons  $V_i$  la tension existant à la sortie d'un neurone  $i$  et  $U_i$  sa tension d'entrée (cf. figure 2.14).  $V$  et  $U$  sont reliés par ce qu'on appellera  $g$ , le gain, et l'on a :  $V_i = g(U_i)$ . L'équation de mouvement du réseau se déduit de l'équation fondamentale de l'électricité :

$$I = C \frac{dU}{dt} \quad (2.78)$$

$$\Rightarrow C \frac{dU_i}{dt} = I_i - \frac{U_i}{R_i} + \sum_j T_{ij} V_j \quad (2.79)$$

$$\text{où } \frac{1}{R_i} = \frac{1}{\rho_i} + \sum_j \frac{1}{R_j} \quad (2.80)$$

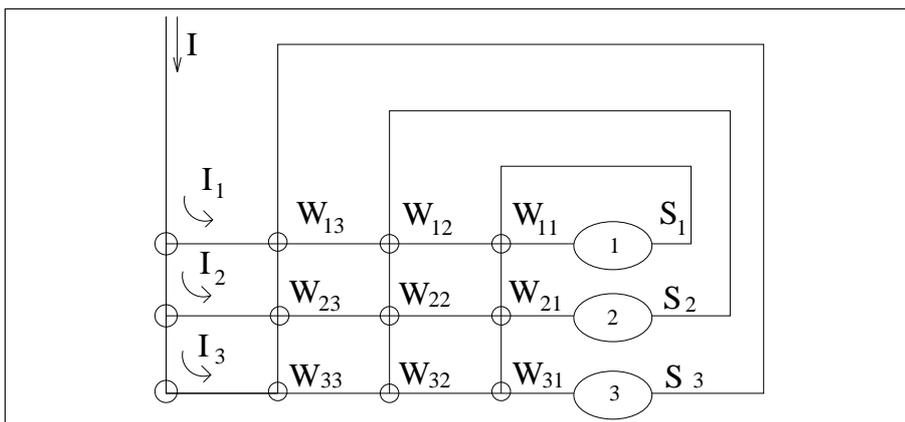


FIG. 2.13 – Un réseau de Hopfield auquel on a ajouté des entrées extérieures.

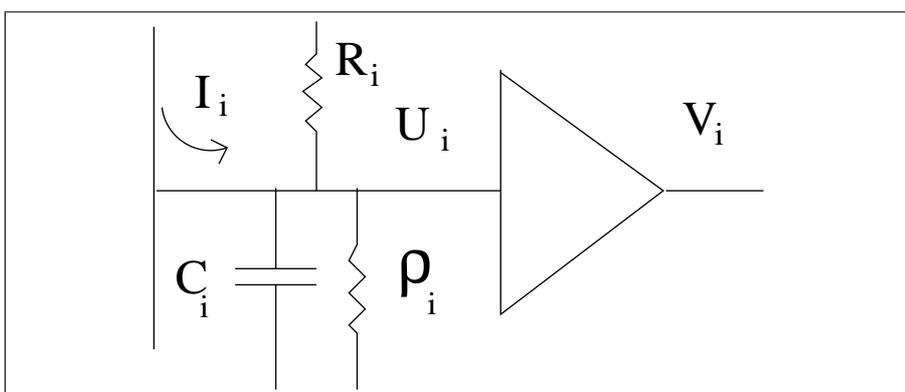


FIG. 2.14 – Un neurone réalisé avec un amplificateur opérationnel.

et où  $\rho$  représente l'impédance d'entrée de l'AO et  $R_j$  les résistances codant pour les poids des connexions.

### 2.7.3 Énergie

Considérons maintenant la quantité que nous appellerons énergie :

$$E = -\frac{1}{2} \sum_i \sum_{j \neq i} T_{ij} V_i V_j + \sum_i \frac{1}{R_i} \int_0^{V_i} g^{-1}(x) dx - \sum_i I_i V_i \tag{2.81}$$

sa variation au cours du temps lorsque  $V_i$  varie peut s'écrire, si  $T_{ij}$  est symétrique :

$$\frac{dE}{dt} = - \sum_i \left( \sum_{j \neq i} T_{ij} V_j - \frac{U_i}{R_i} + I_i \right) \frac{dV_i}{dt} \tag{2.82}$$

$$\text{car } \frac{d}{dt} \int_0^V g^{-1}(x) dx = \frac{d}{dt} f(V) = \frac{df}{dV} \frac{dV}{dt} = g^{-1}(V) \frac{dV}{dt} = U \frac{dV}{dt} \tag{2.83}$$

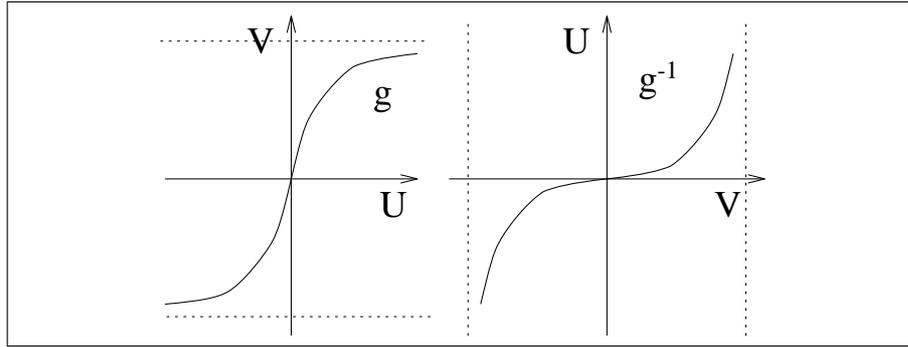


FIG. 2.15 – Sens de variation de la fonction représentant la tension d'entrée  $U$  ou la tension de sortie  $V$ .

On a donc (voir figure 2.15) :

$$\frac{dE}{dt} = - \sum_i C_i \frac{dU_i}{dt} \frac{dV_i}{dt} \quad (2.84)$$

$$= - \sum_i C_i \frac{dU_i}{dV_i} \left( \frac{dV_i}{dt} \right)^2 \quad (2.85)$$

$$= - \sum_i C_i g^{-1'} \left( \frac{dV_i}{dt} \right)^2 \quad (2.86)$$

$$\leq 0 \quad \text{car} \quad C_i, g^{-1'} > 0 \quad (2.87)$$

$E$  est donc une fonction de Lyapunov et il existe des points fixes.

Mais où sont les minima de  $E$  ?

Reprenons l'expression de  $E$  mais avec les courants externes  $I_i$  nuls :

$$E = - \frac{1}{2} \sum_i \sum_{j \neq i} T_{ij} V_i V_j + \sum_i \frac{1}{R_i} \int_0^{V_i} g^{-1}(x) dx \quad (2.88)$$

- dans le cas discret, seul le premier terme existe et on sait qu'alors les minima sont situés aux sommets du cube de dimension  $N$  (donc à  $2^N$  sommets),
- dans le cas continu ce comportement est forcément modifié, et nous allons voir de quelle façon.

#### 2.7.4 Renormalisation

Des théorèmes sur les fonctions et leurs fonctions réciproques nous déduisons que :

$$\text{si } V = g(U) \implies U = g^{-1}(V), \quad \text{alors } V = g(\lambda U) \implies U = \frac{1}{\lambda} g^{-1}(V) \quad (2.89)$$

Il y a changement de gain mais pas d'asymptotes (voir figure 2.16).

Le deuxième terme s'écrit alors :

$$\frac{1}{\lambda} \sum_i \frac{1}{R_i} \int_0^{V_i} g^{-1}(x) dx \quad (2.90)$$

si  $\lambda \rightarrow \infty$  (si le gain devient très grand), ce terme devient négligeable. De plus, s'il le fallait, ainsi qu'on peut le voir sur la figure 2.16, la surface comprise entre  $g^{-1}(x)$  et l'abscisse tend vers 0 quand le gain tend vers l'infini.

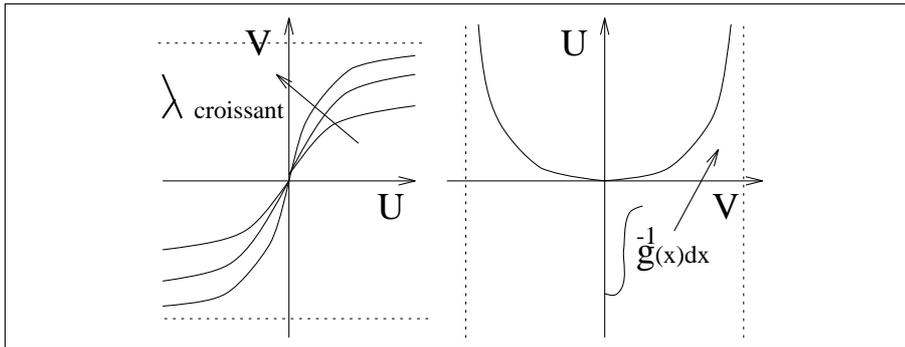


FIG. 2.16 – Effet de la renormalisation.

- **Conclusion 1** : pour un grand gain, dans le cas continu, les points stables sont ceux du cas stochastique.
- **Conclusion 2** : on va pouvoir résoudre des problèmes d’optimisation grâce à des AO à grand gain.

**2.7.5 Application au convertisseur analogique–numérique**

Supposons que l’on se fixe alors de grands gains, que les connexions sont symétriques et qu’il n’y a pas de connexion d’un neurone sur lui même ( $T_{ij} = T_{ji}, T_{ii} = 0$ ) : moyennant ces hypothèses nous pouvons réaliser un convertisseur analogique–numérique de la façon représentée sur la figure 2.17 (ne vous préoccupez pas des poids des connexions, nous verrons plus loin comment on les calcule).

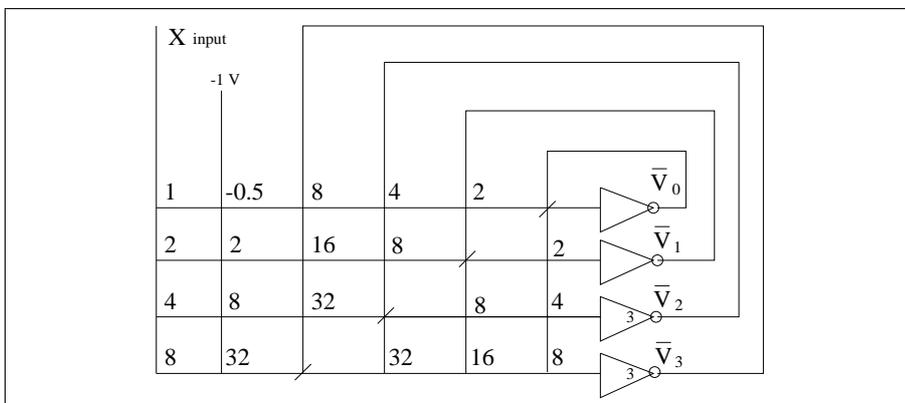


FIG. 2.17 – Un convertisseur Analogique/Numérique qui fonctionne.

La stratégie consistera à considérer que ce problème relève du domaine de l’optimisation avec contraintes et que ces contraintes sont doubles : tout d’abord la concaténation des sorties doit donner la représentation binaire de la tension d’entrée et ensuite chaque sortie doit être exactement à 0 ou à 1 (nous poserons que les tensions peuvent varier entre 0 et 1). Ce qui donne :

$$E1 = \frac{1}{2}(x - \sum 2^i V_i)^2 \tag{2.91}$$

$$E2 = \frac{1}{2} \sum \lambda_i V_i (V_i - 1) \tag{2.92}$$

Le facteur  $\lambda_i$  est là pour relier les deux contraintes.

Remarquons que  $E1$  est bien toujours positive et est minimale quand les sorties représentent les solutions ;  $E2$  est bien minimale si  $V_i = 0$  ou  $1$ . L'énergie totale sera la somme des deux termes.

Développons :

$$E = \frac{1}{2} \left( x - \sum_i 2^i V_i \right)^2 + \frac{1}{2} \sum_i \lambda_i V_i (V_i - 1) \quad (2.93)$$

$$= \frac{1}{2} \left( x^2 - 2x \sum_i 2^i V_i + \left( \sum_i 2^i V_i \right) \left( \sum_j 2^j V_j \right) \right) + \frac{1}{2} \sum_i \lambda_i V_i^2 - \frac{1}{2} \sum_i \lambda_i V_i \quad (2.94)$$

$$= \frac{1}{2} x^2 - x \sum_i 2^i V_i + \frac{1}{2} \left( \sum_i 2^{2i} V_i^2 + \sum_i \sum_{j \neq i} 2^{i+j} V_i V_j \right) + \frac{1}{2} \sum_i \lambda_i V_i^2 - \frac{1}{2} \sum_i \lambda_i V_i \quad (2.95)$$

Cette expression se simplifie si nous posons que  $\lambda_i = -2^{2i}$  car alors les termes diagonaux (ceux en  $V_i^2$ ) s'annulent. On obtient alors :

$$E = -x \sum_i 2^i V_i + \frac{1}{2} \sum_i \sum_{j \neq i} 2^{i+j} V_i V_j + \frac{1}{2} \sum_i 2^{2i} V_i \quad (2.96)$$

$$= \frac{1}{2} \sum_i \sum_{j \neq i} 2^{i+j} V_i V_j - \sum_i (x 2^i - 2^{2i-1}) V_i \quad (2.97)$$

que nous comparerons alors terme à terme à l'expression "classique" de l'énergie :

$$E = -\frac{1}{2} \sum_i \sum_{j \neq i} T_{ij} V_i V_j - \sum_i I_i V_i \quad (2.98)$$

Il vient immédiatement :

$$T_{ij} = -2^{i+j} \quad I_i = x 2^i - 2^{2i-1} \quad (2.99)$$

Les "conductances"  $T_{ij}$  sont donc toutes négatives, ce qui n'existe pas dans la réalité. Heureusement nous pouvons inverser la sortie des AO, ce qui résoud le problème.

Nous obtenons par exemple pour la connexion du premier AO sur le deuxième, puisque  $i = 1$  et  $j = 0$ ,  $T_{ij} = 2$  ; pour la connexion supplémentaire (si elle est reliée à -1 Volt),  $2^{2i-1} = 0.5$  et pour la liaison à l'entrée à convertir :  $2^i = 1$ .

Le schéma représenté sur la figure 2.17 donne toutes les valeurs des connexions.

### Remarques

- Le nombre le plus petit que l'on peut coder est 0000. Ce qui veut dire que, sur la première ligne on a la tension  $1x - 0.5 \times 1V < 0$ , (puisque tous les neurones sont à l'état inactif)  $\Rightarrow x < 0.5$  ; sur la deuxième ligne on a :  $2x - 2 < 0 \Rightarrow x < 1$  ; ... La plus petite tension que l'on peut mettre à l'entrée de ce convertisseur est 0.5 V.

Le nombre le plus grand que l'on peut coder est 1111. Ce qui veut dire que, sur la première ligne on a la tension  $x - 2 - 4 - 8 - 0.5 > 0$ , (puisque tous les neurones sont à l'état actif)  $\Rightarrow x > 14.5$  ; sur la deuxième ligne on a :  $x - 2 - 16 - 8 - 2 > 0 \Rightarrow x > 28$  ; ... La plus grande tension que l'on peut mettre à l'entrée de ce convertisseur est 14.5 V.

Sur 4 bits, on écrit 16 chiffres 0, 1, ..., 15 ; on a donc

$$-0.5 < x < 15.5 \quad (2.100)$$

- Si on change les AO, et qu’au lieu de fonctionner entre 0 et 1 V ils fonctionnent entre 0 et  $V_{max}$  il faut changer les  $T_{ij}$  en :

$$T_{ij} = -\frac{2^{i+j}}{V_{max}} \quad (2.101)$$

- Si on veut convertir sur  $[0, V_M]$  et non sur 16 V ( $16 = 2^4$ ), il faut modifier les connexions d’entrée en :

$$\frac{2^{4+i}}{V_M} \quad (2.102)$$

- Si on veut une référence  $-V_R \neq -1V$ , il faut que le courant constant soit fourni au travers d’une conductance :

$$\frac{2^{2i-1}}{V_R} \quad (2.103)$$

### Résultats

Si, pour faire une conversion, on fait partir les AO de 0V, alors la conversion est parfaite. Mais si on laisse les tensions libres, alors on observe un hystérésis (voir figure 2.18)

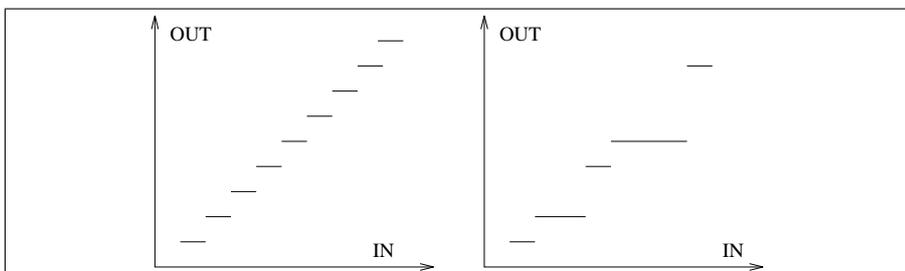


FIG. 2.18 – A gauche les conversions obtenues en partant d’un état initial où tous les AO sont positionnés à 0V. A droite ce qui se passe lorsque le système est laissé libre

### Explications

Supposons que l’on vienne de convertir 7 V. Le paysage énergétique autour de 7 est représenté sur la figure 2.19. Il y a 2 unités énergétiques entre la configuration “9” et la configuration “7” car  $\frac{1}{2} \sum T_{ij} V_i V_j = \frac{1}{2} (9 - 7)^2 = 2$ . De la même façon il y a 0.5 unités énergétiques entre la configuration “8” et la configuration “7” car  $\frac{1}{2} \sum T_{ij} V_i V_j = \frac{1}{2} (8 - 7)^2 = 0.5$ .

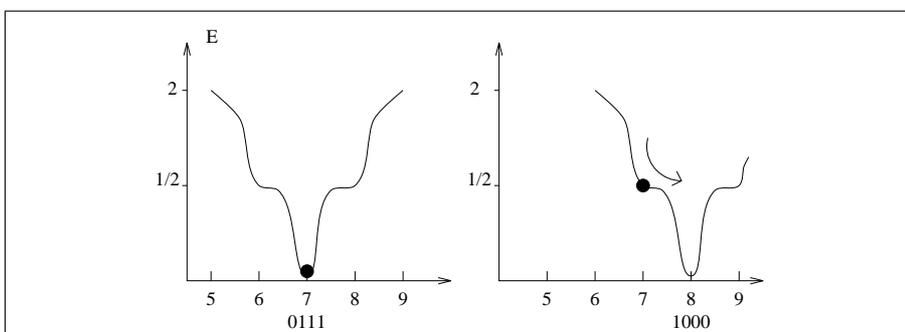


FIG. 2.19 – Le paysage énergétique autour des configuration “7” (0111) et “8” (1000)

Lorsque l'on veut convertir "8" à partir de la position "7", le système peut rester bloqué dans la configuration précédente car la courbe représentant l'énergie peut avoir un petit creux, un minimum local qui va le piéger dans cet état d'où il ne pourra pas sortir spontanément.

### 2.7.6 Application au problème du voyageur de commerce

#### Le problème

Soit un voyageur de commerce devant passer une fois et une seule dans  $N$  villes. Dans quel ordre doit-il parcourir ces villes pour minimiser la distance parcourue ?

Ce problème est l'archétype des problèmes NP-complets, ces problèmes pour lesquels la durée de résolution augmente plus vite que n'importe quelle puissance de leur taille (cf le chapitre sur la complexité). Je ne sais plus jusqu'à quelle taille on sait résoudre le problème du voyageur de commerce, actuellement, mais admettons que ce soit 314 villes ; alors on ne sait pas encore résoudre le problème pour 315 villes et il se pourrait qu'on doive attendre longtemps avant que ce soit réalisé. Et ce sera encore beaucoup, beaucoup plus dur pour 316 villes !

Bref, ce sont les problèmes les plus "durs" que l'on connaisse.

#### Mise en forme

Soit  $N$  villes  $\{A, B, \dots, N\}$  et soit  $1, 2, \dots, n$  le numéro d'une ville donnée dans un tour. Par exemple la ville  $A$  a le n° 2, la ville  $B$  le n° 3,  $C$  le n° 1 et  $D$  le n° 4. Ce qui veut dire que le tour pour 4 villes commence par  $C$ , se poursuit par  $A$  puis  $B$  et se termine par  $D$ .

Calculons le nombre de tours distincts. Il y a  $n$  façons de choisir la première ville, puis  $n-1$  façons de choisir la deuxième, . . . , 1 façon de choisir la dernière. Ce qui fait  $n!$  possibilités. Parmi celles-ci, certaines sont équivalentes pour notre problème : on peut commencer par n'importe laquelle de ces  $n$  villes et le tour peut se faire dans les deux sens. En définitive on a

$$\text{nombre de possibilités distinctes} = \frac{N!}{2N} \quad (2.104)$$

On peut représenter ce problème sous la forme d'un tableau de la façon suivante :

	1	2	3	4
A	0	1	0	0
B	0	0	1	0
C	1	0	0	0
D	0	0	0	1

On considère alors qu'il ne doit y avoir qu'un seul "1" sur chaque ligne ainsi qu'un seul "1" par colonne. Vous voyez apparaître des contraintes. Il existe aussi une contrainte supplémentaire, celle sur la distance parcourue : la somme des distances entre villes doit être minimale.

On va procéder en utilisant  $n^2$  neurones, un neurone pour chacune des cases du tableau précédent. Un neurone à "1" codera donc à la fois la ville et sa position dans le circuit.

### Fonction énergie

L'état du neurone codant pour la ville  $X$  et la position  $j$  sera noté  $V_{Xj}$ .

Écrivons alors la contribution de chacune des contraintes à la fonction énergie (Les grandeurs  $A, B, \dots$  sont ici des constantes reliant les contraintes) :

- Pas plus de un "1" par ligne :

$$\sum_X \sum_i \sum_{j \neq i} V_{X_i} V_{X_j} \times \frac{A}{2} \quad \text{doit être mini} \quad (2.105)$$

- Pas plus de un "1" par colonne :

$$\sum_i \sum_X \sum_{Y \neq X} V_{X_i} V_{Y_i} \times \frac{B}{2} \quad \text{doit être mini} \quad (2.106)$$

- Le nombre de "1" doit être égal à  $n$  :

$$\left( \sum_X \sum_i V_{X_i} - n \right)^2 \times \frac{C}{2} \quad \text{doit être mini} \quad (2.107)$$

- La distance doit être la plus petite possible :

$$\sum_X \sum_{Y \neq X} \sum_i d_{XY} V_{X_i} (V_{Y_{i+1}} + V_{Y_{i-1}}) \times \frac{D}{2} \quad \text{doit être mini} \quad (2.108)$$

- Une fois cette somme effectuée on compare terme à terme avec l'expression classique de l'énergie :

$$E = -\frac{1}{2} \sum_{X_i} \sum_{Y_j} T_{X_i Y_j} V_{X_i} V_{Y_j} - \sum_{X_i} I_i V_{X_i} \quad (2.109)$$

- Tous calculs faits, on trouve :

$$\begin{aligned} T_{X_i Y_j} &= -A \delta_{XY} (1 - \delta_{ij}) && \text{inhibition sur chaque ligne} \\ &\quad -B \delta_{ij} (1 - \delta_{XY}) && \text{inhibition sur chaque colonne} \\ &\quad -C && \text{inhibition générale} \\ &\quad -D d_{XY} (\delta_{j,i+1} + \delta_{j,i-1}) && \text{terme de données} \\ I_i &= \dots \end{aligned} \quad (2.110)$$

### Résultats

Sur la figure 2.20 qui est tirée de [HT85]

sont représentés les résultats obtenus sur un problème comportant 10 villes ainsi que pour 30 villes. Nous voyons que le réseau ne trouve pas la bonne solution, celle obtenue par Lin et Kernighan (une méthode de recherche opérationnelle), mais une solution proche.

Pour quelques dizaines de villes cette méthode peut être considérée comme meilleure si on ne recherche pas une solution exacte mais une solution approchée et rapide. Au-delà de quelques dizaines de villes, ce qui fait entre 100 et 1000 neurones ! elle est totalement inefficace *per se*, mais des méthodes dérivées existent, plus complexes évidemment.

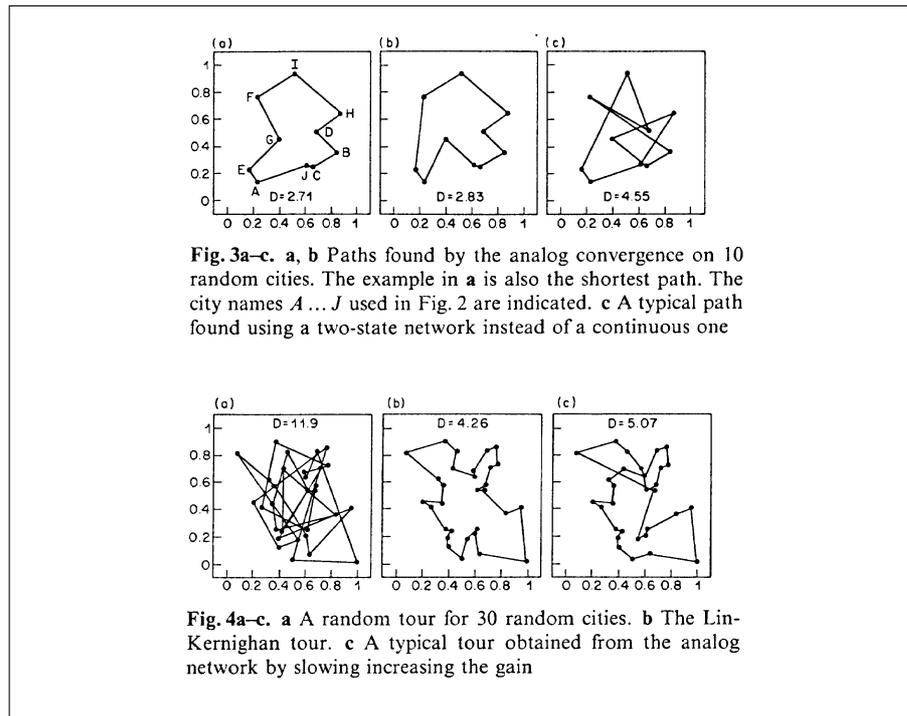


FIG. 2.20 – Résultats pour 10 et 30 villes.

## 2.7.7 Application au problème des bibliothécaires

### Mise en forme

C'est un problème d'affectation. Soit un certain nombre de bibliothécaires pour lesquels on connaît leur vitesses de classement de chaque type de volume, art, physique, etc. Quelle catégorie va-t-on donner à chacun pour que la bibliothèque soit rangée le plus rapidement possible ?

On va former un tableau où chaque ligne correspond aux différentes vitesses de rangement d'un bibliothécaire donné et chaque colonne à une catégorie particulière de livres (voir figure 2.21).

Répartir les tâches revient alors à choisir  $n$  éléments du tableau sachant que l'on ne peut choisir qu'un seul élément par ligne et par colonne, puisque, par définition, chaque bibliothécaire ne peut s'occuper que d'un seul secteur à la fois. La meilleure solution est celle qui maximalise la somme des vitesses de rangement de tous les bibliothécaires.

Pour résoudre ce problème on construit un réseau comportant  $n \times n$  AO, où chaque ligne correspond à un bibliothécaire et chaque AO de la ligne à une discipline. Les AO d'une même ligne ou d'une même colonne sont reliés par des connexions inhibitrices, car la solution consiste à n'attribuer qu'un seul domaine à un bibliothécaire.

Quel est le nombre d'états stables ? Ce sont les configurations possibles que peuvent prendre les 36 AO, six d'entre eux devant être à "1", avec un seul actif par ligne et par colonne. Comme il y a 6 façons de choisir l'AO à "1" sur la première ligne, 5 façons alors sur la deuxième, ..., il y a au total  $6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$  solutions possibles.

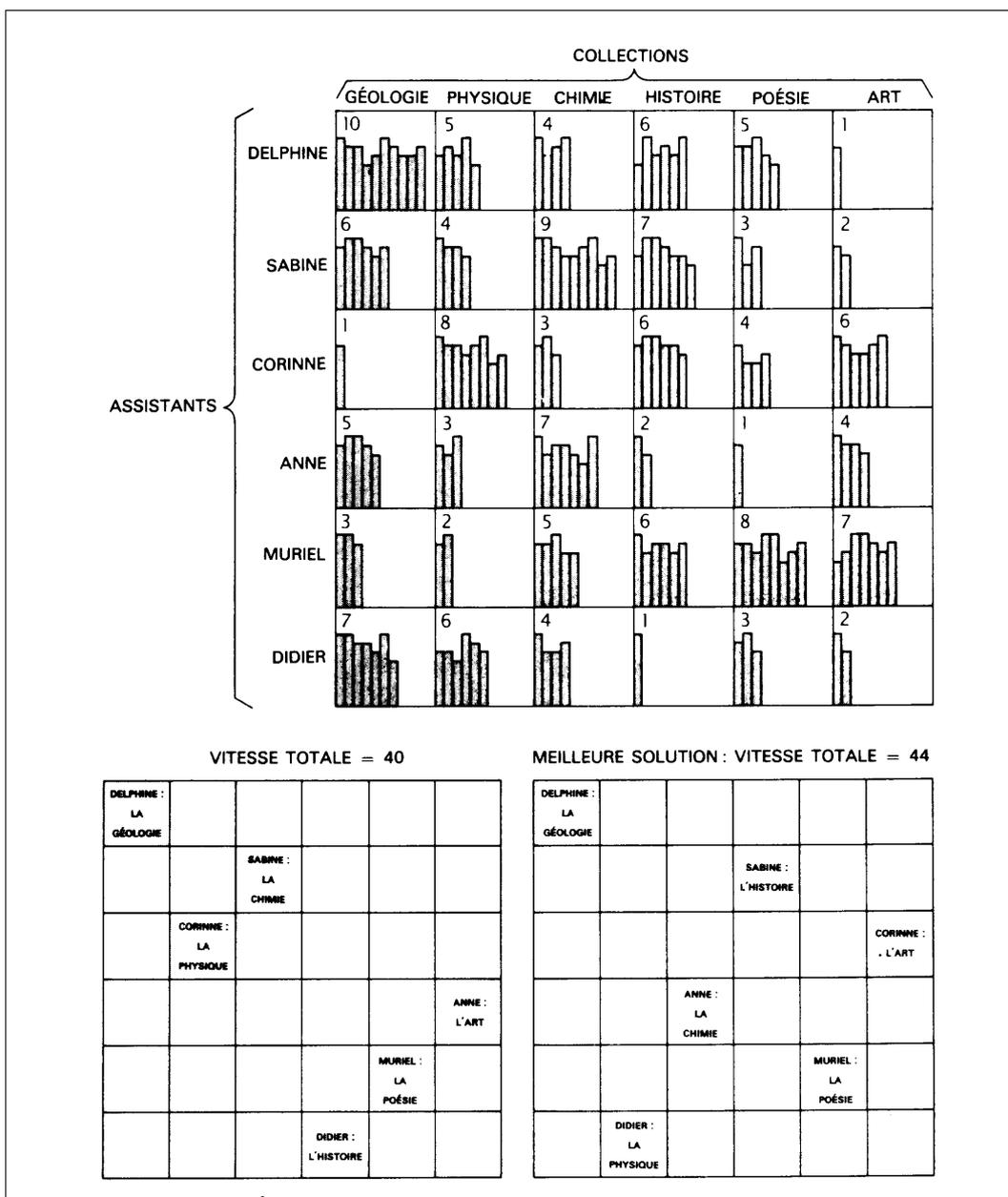


FIG. 2.21 – Ici on cherche à faire classer de la façon la plus rapide possible un certain nombre de livres, chaque bibliothécaire ayant certaines performances de classement selon les types d'ouvrages à classer.

### Résultats

Les résultats sont donnés au bas de la figure 2.21. On remarque que si le réseau n'a pas trouvé la meilleure solution, sa réponse est néanmoins très bonne.

### 2.7.8 Application à l'imagerie médicale

On posera le problème de la segmentation d'une image à  $n$  niveaux de gris contenant  $c$  objets intéressants.

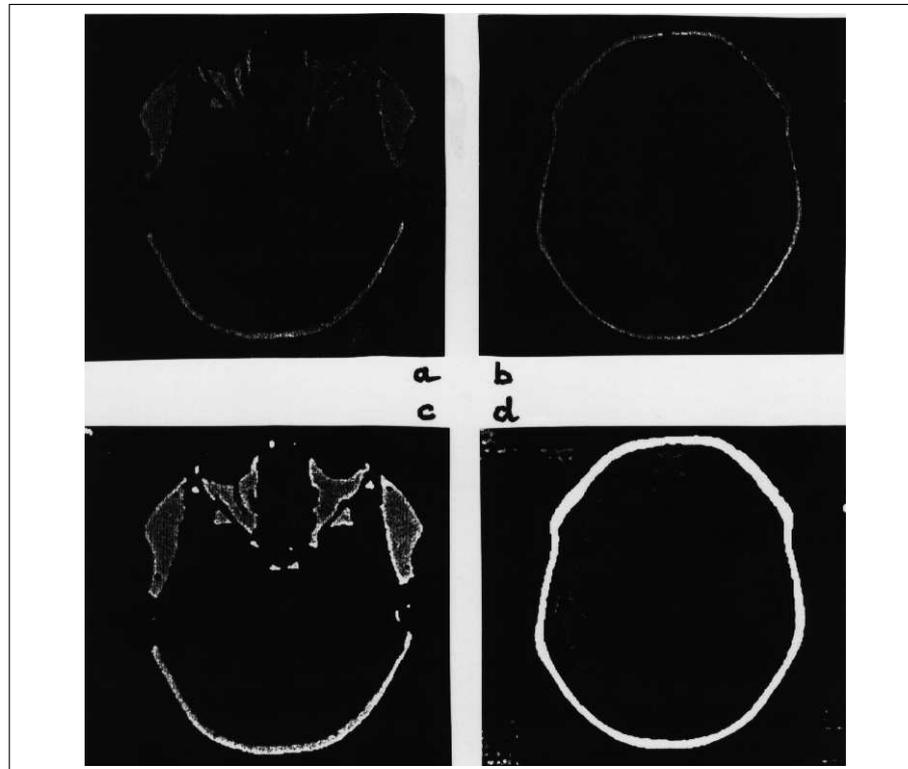


FIG. 2.22 – Les images originales en résonance magnétique (a et b). Les images respectives segmentées pour 3 classes (c et d).

Les images données en exemples sont des images du cerveau obtenues par résonance magnétique. On construira comme d'habitude un tableau où les colonnes représenteront les objets à classer et les lignes représenteront les niveaux de gris. Supposons que l'on veuille  $M$  classes. Ici les objets sont tous les pixels d'une image  $L \times L$ . Il y aura donc en fait un tableau à trois dimensions de  $L \times L \times M$  neurones.

La formalisation est un peu longue mais est directement dérivée de ce que nous avons vu précédemment. Le lecteur passionné peut se reporter à l'article de Cheng, Lin et Mao [CLM96]

## 2.8 Conclusions

Utiliser des réseaux de neurones "à la Hopfield" pour résoudre des problèmes d'optimisation nécessite donc de poser *a priori* une fonction énergie et on peut y voir un arbitraire certain car ce n'est donc pas une méthode générale ; il faut bien comprendre le problème avant d'essayer de le résoudre !

Pour peu que l'on ne recherche pas une méthode exacte ce peut être une bonne méthode car elle est rapide. J'ai discuté une fois avec un militaire qui disait préférer un résultat à 10 % immédiatement plutôt qu'une solution au millième une milliseconde trop tard !

## 2.9 Exercices

.

## 2.10 Compléments : systèmes dynamiques

.



# Bibliographie

- [AGS85] Daniel J. Amit, H. Gutfreund, and H Sompolinsky. Spin-glass models of neural networks. *Physical Review*, A32 :1007–1018, 1985.
- [Ama72] S. Amari. Characteristics of random nets of analog neuron-like elements. *IEEE transactions on Systems man and Cybernetics*, SMC-2 :643–657, 1972.
- [Ami89] Daniel J. Amit. *Modelling Brain Function : The World of Attractor Neural Networks*. Cambridge University Press, 1989.
- [CLM96] K.S. Cheng, J.S. Lin, and C.W. Mao. The application of competitive Hopfield neural network to medical image segmentation. *IEEE Trans. on Medical Imaging*, 15(4) :560, 1996.
- [Cow68] J. D. Cowan. *Statistical mechanics of nervous nets*, in *Neural Networks*, Caianiello ed. Springer-Verlag, 1968.
- [Gla63] R.J. Glauber. Time-dependant statistics of the Ising model. *Journal of Mathematical Physics*, 4 :294–307, 1963.
- [Gre60] T.N.E. Greville. Xxx. *SIAM Rev*, 11 :15, 1960.
- [Gro67] S. Grossberg. Non linear difference–differential equation in prediction and learning theory. *Proceedings of the National Academy of Sciences of the USA*, 58 :1329–1334, 1967.
- [GT85] Clarck C. Guest and Robert TeKolste. Designs and devices for optical bidirectionnal associative memory. *Applied Optics*, 26(23) :5055–5060, 1985.
- [Heb49] D.O. Hebb. *The Organization of Behavior : A Neurophysiocal Model*. Wiley, 1949.
- [Hop82] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. USA*, 79 :2554–2558, 1982.
- [HT85] J.J. Hopfield and D.W. Tank. “neural” computation of decisions in optimization problems. *Biological Cybernetics*, 52 :141–152, 1985.
- [Koh74] Tuevo Kohonen. An adaptative associative memory principle. *IEEE transactions on Computers*, C-23 :444–445, 1974.
- [Koh78] Teuvo Kohonen. *Associative Memory, A system-theoretical approach*. Springer-Verlag, 1978.
- [Koh84] Teuvo Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, 1984.
- [Lit74] W. A. Little. The existence of persistent states in the brain. *Mathematical Biosciences*, 19 :101–120, 1974.
- [MCP43] Warren S. Mc Cullogh and Walter Pitts. Logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5 :115–133, 1943.

- [Pen55] R. Penrose. Xxx. *Proc. Cambridge Philos. Soc.*, 51 :406, 1955.
- [PF85] Demetri Psaltis and Nabil Farhat. Optical information processing based on an associative-memory model of neural nets with thresholding and feedback. *Optics Letters*, 10(2) :98–100, 1985.
- [TH86] D.W. Tank and J.J. Hopfield. Simple “neural” optimization networks : An A/D converter, signal decision circuit, and a linear programming circuit. *IEEE Trans. on Circuits and Systems*, 33 :533–541, 1986.
- [Wei89] Gérard Weisbuch. *Dynamique des systèmes complexes*. Interéditions-Éditions du CNRS, 1989.