

Chapitre 6

Mémoire associative

6.1 La tâche

Le but de la mémoire associative est de retrouver un motif mémorisé auparavant. Contrairement à la mémoire dans l'architecture de von Neumann de l'ordinateur classique, les motifs mémorisés ne sont pas retrouvés grâce à une adresse, mais directement par le contenu (content-addressable memory). Regardons un exemple. Comme sur la figure 6.1, on reçoit en entrée un dessin qui peut être partiel ou bruité. La tâche est de retrouver, parmi les motifs en mémoire, le motif le plus proche de l'image d'entrée.

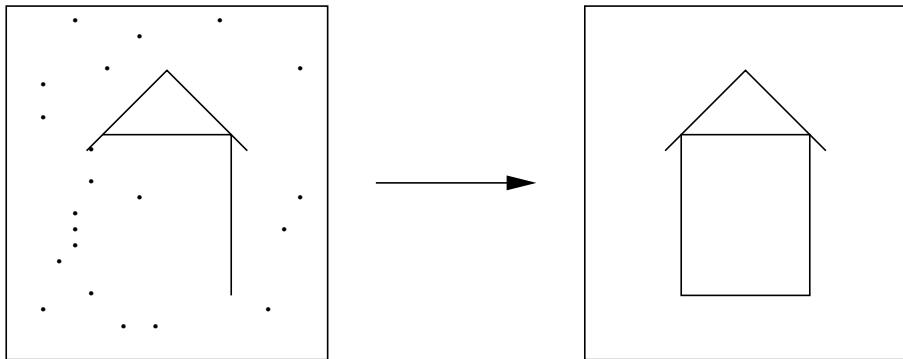


FIG. 6.1 – A gauche est montré l'état actuel $\vec{x}(t_0)$. Le but est de retrouver parmi tous les motifs à disposition, le motif $\vec{\xi}^\mu$ (à droite) qui est le plus proche de $\vec{x}(t_0)$.

Pour analyser cette tâche, on divise l'image en pixels. Les pixels ont des valeurs

x_i avec $-1 \leq x_i \leq 1$, et un indice $1 \leq i \leq N$ où N est le nombre de pixels. L'ensemble de pixels représente l'image actuelle et est noté comme \vec{x} , comme illustré sur la figure 6.2. Les motifs $\vec{\xi}^\mu$ en mémoire ont des valeurs $\xi_i^\mu = \pm 1$. Ici $1 \leq i \leq N$ est l'indice du pixel et μ avec $1 \leq \mu \leq P$ est l'indice du motif. Pour la construction d'une mémoire associative on va identifier chaque pixel avec un neurone.

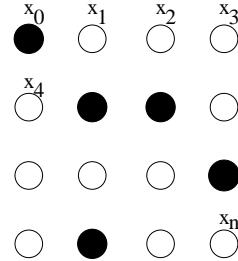


FIG. 6.2 – L'ensemble de pixels x_i représente l'image actuelle \vec{x} . Dans le modèle de Hopfield, chaque pixel correspond à un neurone.

On peut définir une mesure de similitude (“overlap”) entre l'image actuelle \vec{x} et un motif μ :

$$m^\mu(t) = \frac{1}{N} \sum_{i=1}^N \underbrace{\xi_i^\mu \cdot x_i(t)}_{=1 \text{ si } \xi_i^\mu = x_i(t)} \quad (6.1)$$

Par la normalisation, on a

$$|m^\mu(t)| \leq 1$$

Si $m^\mu = 1$, tous les pixels sont corrects: $\xi_i^\mu = x_i(t) \quad \forall i$.

Avant de discuter le modèle de Hopfield de la mémoire associative, rappelons ce que serait un algorithme classique sur un ordinateur traditionnel pour résoudre la tâche de retrouver le motif le plus proche de l'image \vec{x} .

1. Mémoriser tous les motifs $\vec{\xi}^\mu$ pour $1 \leq \mu \leq P$.

$$\xi_i^\mu = \pm 1, \quad 1 \leq \mu \leq P, \quad 1 \leq i \leq N$$

Cela fait $N \times P$ bits à stoquer.

2. Initialiser avec l'état actuel $\vec{x}(t_0)$.
3. Pour $1 \leq \mu \leq P$, calculer la similitude:

$$m^\mu(t) = \frac{1}{N} \sum_{i=1}^N \xi_i^\mu \cdot x_i$$

4. Chercher μ^* avec $m^{\mu^*} \geq m^\mu \quad \forall \mu$.

5. Sortir $\overrightarrow{\xi}^{\mu^*}$

Par contre, avec un réseau de neurones, on peut résoudre la même tâche avec:

- Une mémoire distribuée (les connexions)
- De multiples processeurs simples (les neurones)
- Un algorithme qui est implicite dans l’interaction entre les neurones.

Il nous faut donc définir les connexions ou interactions entre les neurones ainsi que la dynamique de chaque neurone et du réseau. Ces définitions nous amènent au modèle de Hopfield de la mémoire associative.

6.2 Le modèle de Hopfield

Dans le modèle de Hopfield, la dynamique des neurones, les interactions entre les neurones, et la mise-à-jour du réseau sont définis d’une manière très simple.

La dynamique d’un neurone est définie dans le temps discrèt. La nouvelle valeur $x_i(t)$ d’un neurone (pixel) i à temps $t + \Delta t$ est donnée par une fonction sigmoïde de son potentiel à temps t :

$$x_i(t + \Delta t) = g[h_i(t)] = \tanh[\beta \cdot h_i(t)] \quad (6.2)$$

ou le potentiel h_i est défini ainsi:

$$h_i(t) = \sum_{k=1}^N w_{ik} x_k(t). \quad (6.3)$$

C’est-à-dire que chaque paire de neurone (i, k) est connectée, comme illustré sur la figure 6.3.

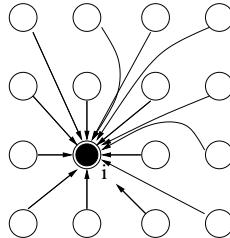


FIG. 6.3 – Chaque paire (i, k) est connectée.

Pour la mise-à-jour du réseau, on a le choix entre deux possibilités:

- i En parallèle:
 - Au temps t , calculer $h_i(t)$ pour tout i .

- Au temps $t + \Delta t$, mettre *tous* les neurones i à jour.

ii Asynchrone:

- Au temps t , calculer $h_i(t)$ pour *un* neurone i .
- Au temps $t + \Delta t$, mettre *ce* neurone i à jour.

Pour bien comprendre **le choix des connexions**, nous discutons d'abord le stockage d'un seul motif μ . On choisit une valeur de $+1$ pour une connexion entre deux neurones, si les deux neurones concernés du motif à stocker en mémoire ont la même valeur; et on prend une valeur de -1 sinon. Ainsi,

$$w_{ik} = \xi_i^\mu \cdot \xi_k^\mu$$

La figure 6.4 montre la valeur des poids que peuvent prendre les connexions entre deux neurones

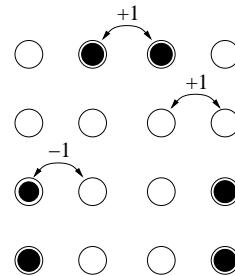


FIG. 6.4 – Connexions entre les neurones.

Le choix expliqué jusqu'ici est une excellente règle pour stocker un seul motif si le motif a à peu près le même nombre de pixels noir que de pixels blanc. Parce que nous aimerais en stocker plusieurs, nous utilisons plutôt une règle qui somme les résultats sur tous les motifs,

$$w_{ik} = \frac{1}{N} \cdot \sum_{\mu=1}^P \xi_i^\mu \cdot \xi_k^\mu \quad (6.4)$$

Le facteur $1/N$ effectue une normalisation qui s'avérera utile plus tard.

Remarque: On peut considérer w_{ik} dans l'équation 6.4 comme étant le résultat de l'apprentissage. Avant l'apprentissage on initialise tous les poids avec $w_{ik}(t=0) = 0$. Ensuite, quand un motif est présenté au réseau, un changement $\Delta w_{ik} = \frac{1}{N} \xi_i^\mu \cdot \xi_k^\mu$ est effectué, ce qui ressemble à la règle de Hebb.

On peut se demander combien de motifs on peut stocker avec la règle 6.4. On trouve que le nombre maximal p_{\max} de motifs qu'on peut ainsi stocker dépend

du nombre de neurones dans le réseau,

$$p_{\max} = \alpha_c \cdot N$$

Où α_c représente la capacité du réseau. Avec des motifs aléatoires, $\alpha_c \approx 0.14$. C'est-à-dire qu'avec des motifs aléatoires, dans un réseau de 1000 neurones, on peut stocker environ 140 images.

Remarque: Rappelons que l'information est stockée dans la valeur des connexions (et non dans les neurones). Le fait que le nombre p_{\max} est proportionnel à N est facile à comprendre car

$$\alpha_c = \frac{p_{\max}}{N} = \frac{N \cdot p_{\max}}{N^2} = \frac{\text{Nombre de pixels à stocker}}{\text{Nombre de connexions à disposition}}$$

6.3 Analyse de la dynamique

But. Dans cette section, nous cherchons à démontrer que le réseau décrit en 6.2 fonctionne correctement. C'est à dire que si nous donnons un motif à reconnaître à notre réseau, alors il se "dirigera" vers le modèle le plus proche du motif en question, parmi les modèles stockés en mémoire.

L'entrée de chaque neurone vaut:

$$h_i(t) = \sum_k w_{ik} x_k(t)$$

Par 6.4, nous pouvons réécrire:

$$h_i(t) = \sum_k \left(\frac{1}{N} \sum_{\mu} \xi_i^{\mu} \cdot \xi_k^{\mu} \right) \cdot x_k(t)$$

Ce qui est équivalent à:

$$h_i(t) = \sum_{\mu} \xi_i^{\mu} \underbrace{\left[\frac{1}{N} \sum_k \xi_k^{\mu} \cdot x_k(t) \right]}_{m^{\mu}(t) \text{ (par 6.1)}}$$

Donc,

$$h_i(t) = \sum_{\mu} \xi_i^{\mu} m^{\mu}(t) \quad (6.5)$$

Admettons que l'entrée aie la meilleure ressemblance avec le modèle ν . Plus spécifiquement, nous faisons l'hypothèse que la similitude avec le motif ν a une

valeur m^ν significative (par exemple $m^\nu = 0.3$), alors que la similitude avec d'autres motifs $\mu \neq \nu$ est $m^\mu = 0$. Prouvons que le réseau se dirige vers ν .

$$m^\nu(t + \Delta t) = \frac{1}{N} \sum_i \xi_i^\nu \cdot x_i(t + \Delta t)$$

Par 6.2,

$$m^\nu(t + \Delta t) = \frac{1}{N} \sum_i \xi_i^\nu \tanh[\beta \cdot h_i(t)]$$

Par 6.5, cela nous donne:

$$m^\nu(t + \Delta t) = \frac{1}{N} \sum_i \xi_i^\nu \tanh[\beta \cdot \sum_\mu \xi_i^\mu m^\mu(t)] \quad (6.6)$$

Nous voulons maintenant montrer que si c'est avec le modèle ν que la ressemblance est la plus grande au départ, alors m^ν augmentera avec le temps.

Comme nous avons fait l'hypothèse que $m^\mu = 0$ si $\mu \neq \nu$, nous pouvons écrire Eq. (6.6) dans la forme

$$m^\nu(t + \Delta t) = \frac{1}{N} \sum_i \xi_i^\nu \tanh[\beta \xi_i^\nu m^\nu(t)]$$

Comme la fonction \tanh est anti-symétrique par rapport à l'origine (c.a.d., $\tanh(-x) = -\tanh(x)$), et que $\xi_i^\nu = \pm 1$, nous pouvons mettre le facteur ξ_i^ν devant la fonction \tanh :

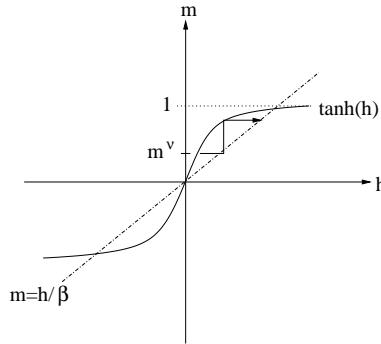
$$m^\nu(t + \Delta t) = \frac{1}{N} \sum_{i=1}^N \underbrace{\xi_i^\nu \xi_i^\nu}_{=1} \tanh[\beta m^\nu(t)].$$

En simplifiant, nous obtenons donc que:

$$m^\nu(t + \Delta t) = \tanh[\beta m^\nu(t)] \quad (6.7)$$

Comme le montre la figure 6.5, comme \tanh est strictement croissante avec une borne supérieure de 1, m^ν va donc tendre vers une valeur proche de 1. Par un raisonnement similaire on peut ensuite montrer que les autres motifs $\mu \neq \nu$, qui avait une valeur $m^\mu = 0$ au départ, restent dans l'état $m^\mu = 0$. Autrement dit, le motif ν qui était partiellement présent au départ est complété suite à la dynamique du réseau et les autres motifs restent invisibles.

Remarque. Si au départ 3 motifs ou plus sont simultanément présent, le réseau se dirige souvent vers un état mixte ('mixture state'). Dans ce cas-là, aucun motif pur est retrouvé.

FIG. 6.5 – Evolution de m^ν .

6.4 Estimation de la Capacité

On a vu précédemment que le nombre maximal p_{\max} de motifs que l'on peut stocker dépend du nombre de neurones dans le réseau,

$$p_{\max} = \alpha_c \cdot N,$$

où α_c représente la capacité du réseau. Dans ce paragraphe on aimerait estimer la capacité du réseau pour des motifs aléatoires.

Nous travaillons avec un modèle de Hopfield (section 6.2) $x_i(t+\Delta t) = \tanh(\beta h(t))$ mais nous choisissons $\beta \gg 1$. Dans ce cas-là, la fonction tanh s'approche de la fonction signe, et on peut considérer que:

$$x_i(t + \Delta t) = \text{sgn}(h(t)) \quad (6.8)$$

Les poids sont pris avec la règle 6.4,

$$w_{ik} = \frac{1}{N} \cdot \sum_{\mu=1}^p \xi_i^\mu \cdot \xi_k^\mu. \quad (6.9)$$

Les valeurs ξ_i^μ des motifs sont générées d'une manière aléatoire. Plus spécifiquement, chaque pixel est indépendant des autres. La probabilité qu'un pixel j soit noir dans le motif μ est de $p = 0.5$, indépendamment de j et μ .

En principe, une mémoire associative devrait être capable de trouver le bon motif et compléter l'image, si un motif partiel est présenté. Pour estimer la capacité du réseau nous posons une condition minimale: si un motif *complet* ν est présenté, le réseau ne doit en tout cas pas détériorer l'image. Un motif complet implique que $x_i(t) = \xi_i^\nu$ pour chaque i . Avec $h_i(t) = \sum_j w_{ij} x_j(t)$ on trouve avec les équations 6.8 et 6.9

$$x_i(t + \Delta t) = \text{sgn} \left(\sum_k \frac{1}{N} \sum_{\mu=1}^p \xi_i^\mu \xi_k^\mu \xi_k^\nu \right) \quad (6.10)$$

Pour remplir notre condition minimale, on doit montrer que $x_i(t + \Delta t) = x_i(t) = \xi_i^\nu$. Dans un premier temps nous échangeons l'ordre de sommation et séparons la somme sur μ en deux parties:

$$x_i(t + \Delta t) = \text{sgn} \left(\xi_i^\nu \left\{ \frac{1}{N} \sum_k \xi_k^\mu \xi_k^\nu \right\} + \frac{1}{N} \sum_{\mu \neq \nu} \xi_i^\mu \sum_k \xi_k^\mu \xi_k^\nu \right) \quad (6.11)$$

Notons que le terme entre crochets est égal à un. Ensuite on peut utiliser le fait que $\xi_i^\nu = \pm 1$. Puisque $\text{sgn}(y)$ est une fonction impaire (c.a.d. $\text{sgn}[y] = (-1) \text{sgn}[(-1)y]$) alors $\text{sgn}[y] = \xi_i^\nu \text{sgn}[\xi_i^\nu y]$. Si on multiplie à l'extérieur et l'intérieur de la fonction sgn avec ξ_i^ν on trouve

$$x_i(t + \Delta t) = \xi_i^\nu \text{sgn} \left(1 + \frac{1}{N} \sum_{\mu \neq \nu} \sum_k \xi_i^\mu \xi_i^\nu \xi_k^\mu \xi_k^\nu \right) \quad (6.12)$$

où on a utilisé que $(\xi_i^\nu)^2 = 1$. Si on définit $a_{i\nu} = -\frac{1}{N} \sum_{\mu \neq \nu} \sum_k \xi_i^\mu \xi_i^\nu \xi_k^\mu \xi_k^\nu$ l'équation précédente se simplifie

$$x_i(t + \Delta t) = \xi_i^\nu \text{sgn}(1 - a_{i\nu}) \quad (6.13)$$

Notre condition minimale, $x_i(t + \Delta t) = x_i(t) = \xi_i^\nu$ est remplie si $a_{i\nu} \leq 1$. Or, pour des motifs aléatoires, $a_{i\nu}$ est une variable stochastique et on peut assimiler $N a_{i\nu}$ à la distance parcourue par une marche aléatoire avec $(p-1) \cdot N$ pas. Pour un nombre de pas suffisamment grand, la distribution des distances parcourues dans une marche aléatoire est approximativement donnée par une fonction Gaussienne avec un écart type de $\sqrt{(p-1) \cdot N}$. On divise par N pour trouver l'écart type σ de la distribution des $a_{i\nu}$:

$$\sigma = \sqrt{(p-1)/N}$$

La probabilité que notre condition minimale ne soit pas remplie est donc donnée par

$$\begin{aligned} P_{\text{Error}} &= \text{Prob} \{a_{i\nu} > 1\} \\ &= \frac{1}{\sqrt{2\pi}\sigma} \int_1^\infty \exp \left[-\frac{(a_{i\nu})^2}{2\sigma^2} \right] da_{i\nu} \\ &= \frac{1}{2} \left[1 - \text{erf} \left(\sqrt{\frac{N}{2(p-1)}} \right) \right] \end{aligned} \quad (6.14)$$

Formellement, la condition minimale n'est jamais remplie. Par contre, si on assouplit la condition et si on admet un taux d'erreur de 0.001 (soit un faux pixel sur une image de 1000 pixels) on peut utiliser la formule pour trouver que le nombre maximal de motifs qu'on peut stocker est de $p_{\text{max}} = 0.105 N$. Autrement dit, la capacité du réseau pour un taux d'erreur de 0.001 est $\alpha = 0.105$.

Remarque: Le calcul montre que pour le stockage, c'est le rapport p/N qui compte, comme on l'avait indiqué auparavant. La capacité α dépend du taux d'erreurs qu'on est prêt à accepter. De

l'autre côté, le calcul indiqué en haut ne considère que le premier pas d'une boucle d'itérations. En principe, on peut craindre un effet d'avalanche. On présente un motif complet, dans le premier pas de temps 2 ou 3 erreurs apparaissent, ce qui déclenche 4 autres erreurs dans le deuxième pas de temps et ainsi de suite. Un calcul beaucoup plus détaillé montre qu'un tel effet d'avalanche existe si $p > 0.14 N$. Autrement dit, la capacité maximale (même si j'accepte quelques erreurs) est limitée à $\alpha_c = 0.14$.

6.5 Motifs corrélés

Jusqu'à maintenant notre analyse se restreint à des motifs aléatoires. Le but de cette section est d'étendre quelques-uns de nos résultats aux images réelles (par exemple des lettres) qui auront forcément des corrélations. Définissons la matrice de corrélations C , dont les éléments $C^{\mu\nu}$ représentent la mesure de similitude entre les motifs μ et ν :

$$C^{\mu\nu} = \frac{1}{N} \sum_i \xi_i^\mu \cdot \xi_i^\nu \quad (6.15)$$

On parle aussi de *corrélation* entre deux motifs. Les motifs ne sont pas corrélés si et seulement si:

$$C^{\mu\nu} = \delta_{\mu\nu} = \begin{cases} 0 & \text{pour } \mu \neq \nu \\ 1 & \text{pour } \mu = \nu \end{cases} \quad (6.16)$$

En général, on considère que les motifs aléatoires ne sont pas corrélés si N tend vers l'infini. Pour les images réelles des corrélations sont toujours présentes. La figure 6.6 nous montre que, par exemple, les images des lettres E et F sont fortement corrélées.

Détour. Avant de s'attaquer aux motifs corrélés, nous voulons d'abord montrer que le réseau décrit en 6.2, selon le modèle de Hopfield, fonctionne correctement. Pour cela, une condition *minimale* est que si on initialise le réseau avec un motif existant en mémoire:

$$x_i(t_0) = \xi_i \quad \forall i, \quad (6.17)$$

alors, le réseau doit "rester" dans le motif. Le calcul dans cette section sera similaire à celui effectué dans les sections précédentes, mais on mettra en évidence le rôle des corrélations.

Remarque. Si on pose, comme dans le modèle de Hopfield (section 6.2) que $x_i(t + \Delta t) = \tanh(\beta h(t))$ et que l'on choisit $\beta \gg 1$, alors la

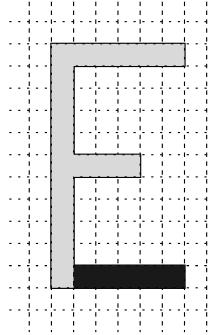


FIG. 6.6 – Les lettres E et F sont fortement corrélées. Elles ne diffèrent que par la partie noir de la lettre E.

fonction \tanh s'approche de la fonction signe, et on peut considérer que:

$$x_i(t + \Delta t) = \text{sgn}(h(t))$$

On pose donc:

$$x_i(t + \Delta t) = \text{sgn} \left(\sum_k w_{ik} x_k(t_0) \right)$$

D'après 6.17, cela nous donne:

$$x_i(t + \Delta t) = \text{sgn} \left(\sum_k w_{ik} \xi_k^\nu \right) \quad (6.18)$$

Dans le modèle de Hopfield, les connexions sont données par l'équation (6.4), En utilisant cette définition, nous obtenons

$$x_i(t + \Delta t) = \text{sgn} \left(\sum_k \frac{1}{N} \cdot \sum_{\mu=1}^P \xi_i^\mu \cdot \xi_k^\mu \xi_k^\nu \right),$$

ce qui peut se réécrire:

$$x_i(t + \Delta t) = \text{sgn} \left[\sum_{\mu} \xi_i^\mu \underbrace{\left(\frac{1}{N} \sum_k \xi_k^\mu \xi_k^\nu \right)}_{C^{\mu\nu}} \right],$$

ou encore avec la matrice de corrélations

$$x_i(t + \Delta t) = \text{sgn} \left[\sum_{\mu} \xi_i^\mu C^{\mu\nu} \right]. \quad (6.19)$$

Puisque, dans le modèle de Hopfield, on travaille avec des motifs aléatoires et avec l'hypothèse d'un grand nombre N de neurones, les motifs ne sont pas corrélés. Si on utilise Eq. (6.16), nous avons:

$$x_i(t + \Delta t) = \operatorname{sgn} \left[\sum_{\mu} \xi_i^{\mu} \delta_{\mu\nu} \right] = \xi_i^{\mu}. \quad (6.20)$$

En résumé, si les motifs ne sont pas corrélés, on reste donc bien dans le motif ν .

Motifs corrélés. Si il y a des corrélations entre les modèles, alors la règle de Hopfield

$$w_{ik} = \frac{1}{N} \cdot \sum_{\mu=1}^P \xi_i^{\mu} \cdot \xi_k^{\mu}$$

ne suffit pas.

La solution consiste en la généralisation de cette règle: On définit les connexions ainsi:

$$w_{ik} = \frac{1}{N} \cdot \sum_{\mu} \sum_{\nu} \xi_i^{\mu} \cdot [C^{-1}]^{\mu\nu} \xi_k^{\nu} \quad (6.21)$$

Ainsi en effectuant le même test que précédemment pour les conditions minimales et en initialisant avec le motif ϱ , on obtient grâce à l'équation (6.18)

$$x_i(t + \Delta t) = \operatorname{sgn} \left(\sum_k w_{ik} \xi_k^{\varrho} \right)$$

Par 6.21, cela nous donne:

$$\begin{aligned} x_i(t + \Delta t) &= \operatorname{sgn} \left(\sum_k \frac{1}{N} \cdot \sum_{\mu} \sum_{\nu} \xi_i^{\mu} \cdot [C^{-1}]^{\mu\nu} \cdot \xi_k^{\nu} \xi_k^{\varrho} \right) \\ x_i(t + \Delta t) &= \operatorname{sgn} \left(\sum_{\mu} \sum_{\nu} \xi_i^{\mu} [C^{-1}]^{\mu\nu} C^{\nu\varrho} \right) \end{aligned}$$

Avec la définition des matrices inverses on trouve

$$x_i(t + \Delta t) = \operatorname{sgn} \left(\sum_{\mu} \xi_i^{\mu} \delta_{\mu\varrho} \right) = \operatorname{sgn}(\xi_i^{\varrho}) = \xi_i^{\varrho}$$

Donc, on est bien resté dans le motif initial.

q.e.d.