

Simulation d'un réseau de neurones  
biologiques en présence de substances  
psychoactives  
(Partie théorique)

ARE DYNAMIC 2018 - L1 UPMC

# Sommaire

<b>I</b>	<b>Etude biologique d'un neurone</b>	<b>3</b>
0.1	Caractéristiques d'un neurone biologique . . . . .	4
0.2	Fonctionnement d'un neurone . . . . .	5
0.3	Action des substances psychoactives . . . . .	6
<b>II</b>	<b>Comparaison de la modélisation mathématique d'un réseau neuronal avec un réseau biologique</b>	<b>8</b>
0.4	Modélisation mathématique actuelle des réseaux de neurones . . . . .	9
0.4.1	Modélisation d'un unique neurone . . . . .	9
0.4.2	Modélisation de plusieurs neurones . . . . .	11
0.5	Le temps dans les simulations . . . . .	11
0.6	Différences avec le modèle biologique . . . . .	12
<b>III</b>	<b>Modélisations d'un réseau neuronal biologique</b>	<b>13</b>
0.7	Modélisation simplifiée . . . . .	15
0.7.1	Présentation : . . . . .	15
0.7.2	Fonctionnement algorithmique . . . . .	16
0.7.3	Exemple . . . . .	16
0.8	Modélisation tenant compte du temps de décroissance du potentiel . . . . .	19
0.8.1	Défaut de la modélisation 0.7 . . . . .	19
0.8.2	Présentation . . . . .	21
0.8.3	Fonctionnement algorithmique . . . . .	22
0.8.4	Exemple . . . . .	23
0.8.5	Remarque sur la simulation synchrone . . . . .	25
0.9	Modélisation tenant compte du poids des connexions . . . . .	25
0.9.1	Défaut des modélisations 0.7 et 0.8 . . . . .	25
0.9.2	Présentation . . . . .	26
0.10	Modélisation tenant compte des potentiels excitateurs et inhibiteurs . . . . .	27
0.10.1	Autre défaut de la modélisation 0.7 . . . . .	27
0.10.2	Présentation . . . . .	28
0.11	Modélisation tenant compte de la plasticité synaptique . . . . .	28
0.11.1	Défaut des modélisations précédentes . . . . .	28
0.11.2	Plasticité à court terme . . . . .	28
0.11.3	Présentation . . . . .	30
0.11.4	Fonctionnement algorithmique . . . . .	32
0.11.5	Exemple de fonctionnement de la simulation . . . . .	35
0.12	Modélisation des effets d'une substance psychoactive . . . . .	37
0.12.1	Exemple : . . . . .	38
0.12.2	Fonctionnement algorithmique . . . . .	40
<b>IV</b>	<b>Résultats exploitables</b>	<b>41</b>
0.13	Tracé de l'activité électrique neurone par neurone . . . . .	42
0.14	Tracé de l'activité du réseau . . . . .	42

<b>V</b>	<b>Utilisation de la matrice C, matrice des relations du réseau neuronal</b>	<b>43</b>
0.15	Initialisation de la matrice C . . . . .	44
0.16	Représentation graphique des modélisations . . . . .	44

## Partie I

# Etude biologique d'un neurone

## 0.1 Caractéristiques d'un neurone biologique

Les neurones ([5] et [7]) ont pour rôle de faire circuler les informations entre l'environnement et l'organisme, ou au sein de l'organisme. Ils permettent la communication et la transmission de l'information entre les différentes parties de l'organisme.

Un neurone est une cellule constituée :

- d'un corps cellulaire
- d'un noyau, contenu dans le corps cellulaire
- de dendrites
- d'un axone

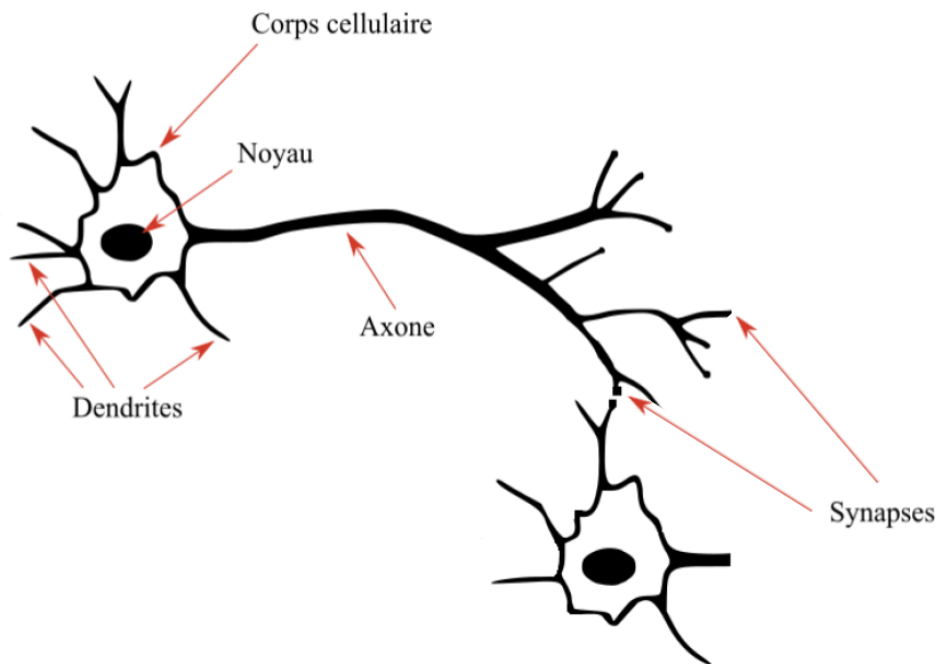


Figure 1: Structure d'un neurone biologique

**Les dendrites** sont des ramifications permettant à la cellule nerveuse de recevoir les informations, sous forme chimique, de la part des autres neurones ; elles peuvent être assimilées à des antennes. Les dendrites sont uniquement affectées à la réception de messages chimiques (d'informations) et non à leurs émissions.

**L'axone** est la partie du neurone permettant l'émission d'un message sous forme électrique. Le courant électrique y circule, en outre, grâce à la présence d'ion sodium  $Na^+$ . L'axone se scinde en nombreuses ramifications en direction de différents neurones permettant alors le passage d'une information d'un neurone à plusieurs autres neurones. Cette information conduite électriquement par l'axone est alors reçue chimiquement par les dendrites des autres neurones. Cette conversion d'un message électrique en un message chimique est assurée par les synapses.

Ainsi, les relations entre neurones sont asymétriques en général (un neurone peut envoyer une information à un autre neurone, sans que celui-ci puisse en faire de même).

**Les synapses** permettent donc la conversion d'un signal électrique en un signal chimique. Dans les faits, des canaux calciques présents sur la membrane de l'axone permettent la libération d'ion calcium  $Ca^{2+}$  lorsqu'un courant électrique les active. Dès lors, lorsque le neurone s'active, c'est-à-dire qu'il émet un courant électrique aussi appelé potentiel d'action, il active ces canaux libérant brutalement une dose massive d'ion  $Ca^{2+}$  dans le synapse.

A l'intérieur de la synapse se trouvent les vésicules présynaptiques contenant les neurorécepteurs (les molécules chimiques permettant la transmission du signal aux autres neurones). Ces vésicules ne peuvent pas traverser la membrane plasmique de la synapse; ils restent donc à l'intérieur de la synapse. Toutefois, la présence d'ions  $Ca^{2+}$  permet la fusion des membranes des vésicules avec la membrane plasmique entraînant alors le déversement du contenu des membranes (les neurorécepteurs) en dehors du synapse.

Remarquons que la différence de potentiel entre la forte présence d'ion  $Ca^{2+}$  à l'extérieur de la membrane de l'axone et sa faible présence à l'intérieur entraîne une grande force électromotrice poussant les ions  $Ca^{2+}$  à se déplacer vers la synapse.

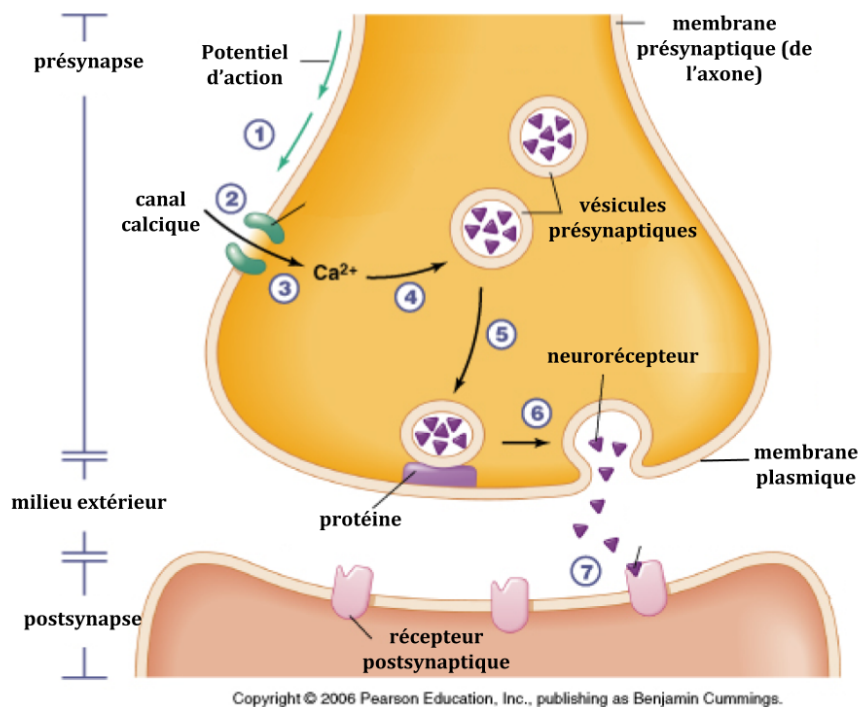


Figure 2: Fonctionnement d'une synapse

## 0.2 Fonctionnement d'un neurone

- Lorsqu'une dendrite perçoit un ou plusieurs neurorécepteurs, elle va procéder à une conversion, cette fois ci, chimique-électrique. Dès lors, chaque dendrite d'un neurone va avoir un potentiel différents (selon qu'elle a été fortement sollicitée ou non ; qu'elle a perçu un plus ou moins grand nombre de neurorécepteurs). Le neurone en lui-même va procéder à une somme, en sommant à chaque instant le potentiel associé aux dendrites.
- Si le potentiel total est supérieur ou égal à un certain seuil, appelé seuil critique de dépolarisation, alors le neurone va émettre un signal électrique, un potentiel d'action, transmis aux autres neurones par l'intermédiaire de l'axone. En d'autres termes, par cette action, le neurone va libérer son potentiel, se décharger, se dépolariser.

En fait, le neurone procède à une sommation spatiale et temporelle. Spatiale car la somme prend en compte, par l'intermédiaire de ses dendrites, le potentiel des autres neurones dans l'espace et temporelle car une fois le neurone dépolarisé, son potentiel ne chute pas immédiatement à sa valeur au repos. Après une dépolarisation, il s'en suit une lente décroissance (de l'ordre de quelques

millisecondes) du potentiel. Alors, si un potentiel d'une dendrite n'aurait pas suffi à enclencher seul la dépolarisation du neurone, il se peut qu'après une dépolarisation le potentiel total soit encore assez élevé pour permettre de franchir le seuil critique, produisant alors une seconde dépolarisation.

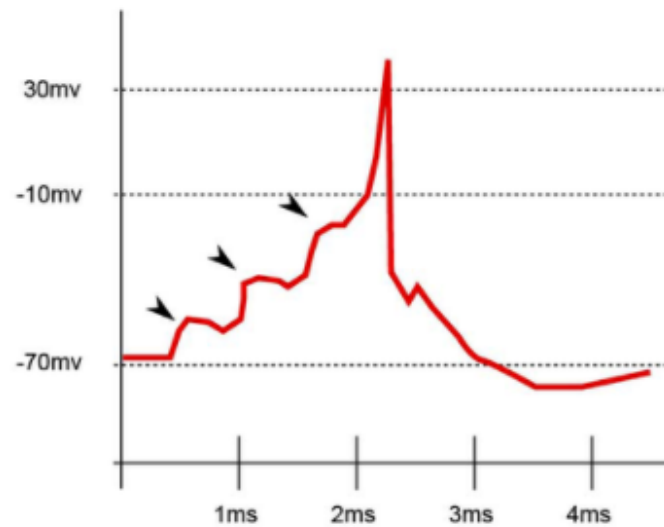


Figure 3: Courbe de l'activité électrique d'un neurone au cours du temps

La figure 3 présente l'évolution du potentiel total d'un neurone au cours du temps. Chaque flèche noire représente la réception d'un nouveau signal de la part d'une dendrite. On constate donc qu'un seul signal ne peut pas franchir seul le seuil critique, mais qu'il faut la réception très peu espacé dans le temps de 3 signaux pour le franchir.

Notons également qu'après une dépolarisation le neurone devient inefficace pour la réception de messages pendant une période de 1 milliseconde. En effet, après une dépolarisation, le neurone ne prendra pas en compte le potentiel de ses entrées (le potentiel des dendrites) pendant 1 ms. Il s'en suit que nous utiliserons ce temps comme pas de temps de simulation.

A présent, nous devons comprendre qu'un même neurorécepteur peut avoir deux effets très distincts. Cela est dû à l'existence de deux types de récepteurs postsynaptiques.

- Certains vont, après avoir été activé par des neurorécepteurs, abaisser le potentiel du neurone. En d'autres termes, il va y avoir une diminution de l'excitabilité de la cellule (on parle de potentiel postsynaptique inhibiteur).
- D'autres vont induire une augmentation du potentiel (on parle de potentiel postsynaptique exciteur).

Ainsi, les potentiels pris en compte dans la sommation effectué par le neurone de la figure 3 étaient des potentiels postsynaptiques exciteurs.

### 0.3 Action des substances psychoactives

Les substances psychoactives telles la drogue ou l'alcool ont deux effets notables. Ils vont se lier aux récepteurs postsynaptiques

- soit sans entraîner d'effet (les récepteurs touchés ne sont alors plus sensibles à la réception de neurorécepteurs)
- soit ils vont entrainer une hyperactivité des récepteurs en simulant la réception quasi-continue de neurorécepteurs

Notons qu'une fois ingéré, l'alcool se transmet dans le sang répartissant alors de manière quasi-uniforme sa présence dans l'organisme. Ainsi, nous n'aurons pas besoin de prendre en compte le

phénomène de diffusion de l'alcool dans l'organisme. Nous considérerons qu'une fois introduit dans l'organisme, chaque neurone est affecté de la même manière. Seule la concentration de l'alcool sera donc prise en compte.



## Partie II

# Comparaison de la modélisation mathématique d'un réseau neuronal avec un réseau biologique

## 0.4 Modélisation mathématique actuelle des réseaux de neurones

Actuellement [3], le modèle mathématique des réseaux de neurones, principalement utilisés pour le Deep Learning, s'inspirent du fonctionnement de neurones biologiques. Mais ces modèles présentent une grande disparité avec le fonctionnement biologique. Pour autant, l'étude de ces réseaux artificiels permet de s'approprier une base mathématique utile pour la suite.

### 0.4.1 Modélisation d'un unique neurone

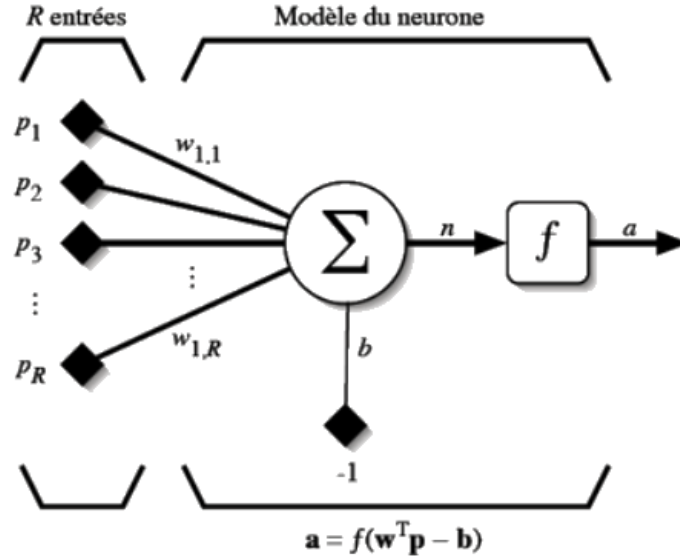


Figure 4: Fonctionnement d'un réseau de neurones artificiel

On modélise [8] alors un neurone par :

- R entrées chacune ayant une valeur  $p_i, i \in \llbracket 1; R \rrbracket$ . Ces entrées correspondent donc aux dendrites d'un neurone biologique. Ces valeurs sont initialisées aléatoirement.
- R "poids" associés aux différentes connexions. Ces poids représentent biologiquement la qualité de la connexion entre deux neurones, c'est-à-dire la relation plus ou moins forte qu'ont deux neurones. Biologiquement la qualité d'une connexion entre deux neurones est d'autant meilleur que plusieurs dendrites sont affectés au même neurone ou que plusieurs ramifications de l'axone sont en direction du même neurone.
- Une fois les valeurs et les poids attribués on somme à un instant  $t$  la valeur de toutes les entrées affectées de leurs poids. On réalise donc l'opération :

$$\sum_{i=1}^R (p_i * w_{1,i})$$

- on rajoute également un coefficient  $b$ , appelé biais du neurone, et correspondant à un facteur correctif (fixé à tâtons).
- Une fois la somme effectuée, on applique au résultat une fonction activatrice permettant de reproduire le processus de dépolarisation d'un neurone biologique vu sur la figure 3

Ce modèle mathématique se prête bien au calcul matriciel. Ainsi, on peut résumer le fonctionnement d'un neurone artificiel de la manière suivante :

On pose :

$$W = \begin{bmatrix} w_{1,1} \\ \vdots \\ w_{1,R} \end{bmatrix}, P = \begin{bmatrix} p_1 \\ \vdots \\ p_R \end{bmatrix}$$

Dès lors, si on appelle  $a$  la sortie du neurone, on obtient :

$$a = f(W^T * P - b)$$

où  $f$  est la fonction activatrice pouvant être une de celles de la figure 5

Nom de la fonction	Relation d'entrée/sortie	Icône
seuil	$a = 0 \quad \text{si } n < 0$ $a = 1 \quad \text{si } n \geq 0$	
seuil symétrique	$a = -1 \quad \text{si } n < 0$ $a = 1 \quad \text{si } n \geq 0$	
linéaire	$a = n$	
linéaire saturée	$a = 0 \quad \text{si } n < 0$ $a = n \quad \text{si } 0 \leq n \leq 1$ $a = 1 \quad \text{si } n > 1$	
linéaire saturée symétrique	$a = -1 \quad \text{si } n < -1$ $a = n \quad \text{si } -1 \leq n \leq 1$ $a = 1 \quad \text{si } n > 1$	
linéaire positive	$a = 0 \quad \text{si } n < 0$ $a = n \quad \text{si } n \geq 0$	
sigmoïde	$a = \frac{1}{1 + \exp^{-n}}$	
tangente hyperbolique	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	
compétitive	$a = 1 \quad \text{si } n \text{ maximum}$ $a = 0 \quad \text{autrement}$	

Figure 5: Tableau des fonctions activatrices (ou fonctions de transferts) possibles

### 0.4.2 Modélisation de plusieurs neurones

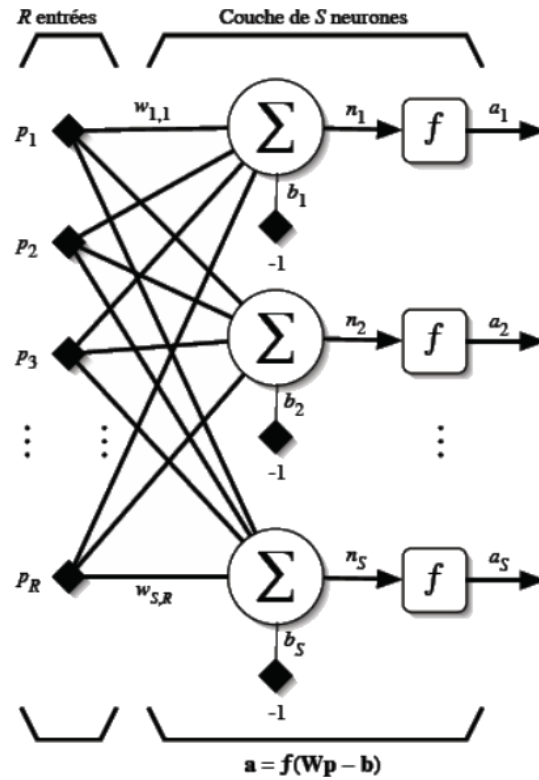


Figure 6: Association de plusieurs neurones artificiel

Comme le montre la figure 6, l'association de plusieurs neurones se fait en connectant les  $R$  entrées à tous les neurones (à toutes les fonctions somme). Ces  $R$  neurones forment alors une couche de neurone.

Dès lors, on modifie légèrement nos matrices. On pose désormais  $W$  telle que :

$$W = \begin{bmatrix} w_{1,1} & w_{2,1} & \dots & w_{n,1} \\ \vdots & \vdots & & \vdots \\ w_{1,R} & w_{2,R} & \dots & w_{n,R} \end{bmatrix}$$

où  $n$  est le nombre de couches constituées d'au plus  $R$  neurones.

Ainsi, chaque colonne de la matrice correspondant à une couche de neurone.

## 0.5 Le temps dans les simulations

Il existe dans grand types de simulations :

- Les simulations synchrones où l'état de chaque neurone d'une même couche est mis à jour en même temps
- Les simulations asynchrones où seul les états des neurones ayant été sollicité par d'autres sont mis à jour.

Une première idée est de simuler à chaque étape uniquement les neurones connectés à d'autres ayant déchargé à l'étape ultérieure. Ce type de simulation est valable pour un réseau où chaque neurone peut recevoir des stimulations que d'un seul autre neurone. Or pour les réseaux plus complexes, chaque neurone peut recevoir des stimuli de plusieurs autres neurones. Ce faisant, lorsque la sortie d'un neurone est calculée, car un neurone relie à lui en transmission à déchargé, elle est calculée avec une seule de ses entrées actualisée (les autres ayant été calculé plusieurs

étapes avant). En effet, si on calcule la sortie du neurone  $i$  à l'instant  $t$  cela signifie qu'une de ses entrées vient d'être (re)calculée au temps  $t-1$  et que le résultat a permis une stimulation. Or au temps  $t-1$  ce neurone n'est pas le seul connecté au neurone  $i$ , mais leurs sorties (correspondant aux entrées du neurone  $i$ ) n'est pas (re)calculées. Ainsi, il faudrait (re)calculer la sortie de chaque neurone connecté au neurone  $i$ , mais la problématique est la même pour les entrées des neurones reliés en transmission au neurone  $i$ .

Ainsi, à chaque étape de la simulation la sortie de tous les neurones doit être calculée. Nous optons donc pour une modélisation synchrone.

## 0.6 Différences avec le modèle biologique

- Premièrement, les valeurs des entrées  $p_i, i \in \llbracket 1, R \rrbracket$  peuvent prendre n'importe quelle valeur à tout instant (soit toutes positives, soit toutes négatives, soit un mixte des deux). Or nous avons vu que le modèle biologique possède deux types d'entrées (l'une excitatrice, l'autre inhibitrice). Nous devons donc, dans notre modèle mathématique prendre en compte ces deux types d'entrées.
- Ensuite, ce modèle mathématique ne prend pas en compte la "mémoire du neurone", c'est-à-dire la sommation temporelle que fait un neurone (il ne prend pas en compte le potentiel du neurone à l'étape ultérieure lors de la sommation).
- Nous nous passerons, de plus, du facteur d'incertitude (le coefficient  $b$ )

## Partie III

# Modélisations d'un réseau neuronal biologique

Rappelons ici les points importants de notre simulation :

- Le pas de temps de la simulation est de 1 milliseconde (le temps durant lequel le neurone est inefficace à la réception de messages après une décharge)
- La simulation sera synchrone
- Le potentiel de chaque neurone à l'étape ultérieure est pris en compte à l'étape suivante
- La fonction activatrice utilisée sera une fonction seuil dont la valeur supérieure au seuil sera de +60mV (ce qui correspond à la valeur des potentiels d'actions ramener à des valeurs positives) car tous les potentiel d'action ont le même potentiel maximal noté  $V_{max}$  (le neurone fonctionne comme un capteur tout ou rien, les informations ne sont donc pas distribuées au travers de l'intensité mais grâce à la modulation de fréquence, le nombre d'impulsions envoyées dans le temps).

On pose alors la fonction activatrice  $f$  telle que :

$$f : \begin{cases} \mathbb{R} \rightarrow [0, V_{max}] \\ x \Rightarrow \begin{cases} V_{max} \text{ si } x \geq \text{seuil} \\ 0 \text{ si } x < 0 \\ x \text{ sinon} \end{cases} \end{cases}$$

Pour la suite, on considère une **réseau récurrent** (contenant une boucle : plusieurs neurones d'une même couche sont connectés ensemble) **à une couche de R neurones**. Autrement dit, on ne considère pas les neurones organisés en différentes couches (ce indispensable pour une modélisation asynchrone comme la nôtre).

on pose :

- $V_i(t)$  le potentiel du neurone  $i$  à l'instant  $t$  de la simulation

•

$$V_t = \begin{bmatrix} V_1(t) & 0 & \dots & \dots & 0 \\ 0 & V_2(t) & 0 & \dots & 0 \\ \vdots & 0 & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \vdots & \vdots & & & 0 \\ 0 & 0 & \dots & 0 & V_R(t) \end{bmatrix} \in \mathcal{M}_R(\mathbb{R})$$

•

$$C = \begin{bmatrix} 1 & c_{2,1} & c_{3,1} & \dots & c_{R,1} \\ c_{1,2} & 1 & & & \vdots \\ \vdots & c_{2,3} & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \vdots & \vdots & & & c_{R,R-1} \\ c_{1,R} & c_{2,R} & \dots & c_{R-1,R} & 1 \end{bmatrix} \in \mathcal{M}_R(\mathbb{R})$$

$$\text{où } \begin{cases} \forall i \in \llbracket 1, R \rrbracket, c_{i,i} = 1 \\ \forall (i, j) \in \llbracket 1, R \rrbracket^2, i \neq j, c_{i,j} = \begin{cases} 1 \text{ si le neurone } i \text{ peut transmettre un signal au neurone } j \\ 0 \text{ sinon} \end{cases} \end{cases}$$

La  $i$ -ème colonne de la matrice  $C$  permet de savoir avec quels neurones le neurone  $i$  peut envoyer des informations et la  $i$ -ème ligne permet de savoir de quels neurones le neurone  $i$  peut recevoir des informations.

D'après la construction de la matrice on constate que le neurone  $i$  est connecté avec lui-même, ce qui est aberrant. En fait fixer  $c_{i,i} = 1$  pour tout  $i \in \llbracket 1, R \rrbracket$  permet de prendre en compte le potentiel du neurone du temps  $t$  au temps  $t+1$ .

- $L_{C,i}$  la i-ème ligne de la matrice C
- 

$$T_t = \begin{bmatrix} t_{1,1} \\ t_{2,1} \\ \vdots \\ t_{R,1} \end{bmatrix} \in \mathcal{M}_{R,1}(\mathbb{R})$$

Cette matrice permet de prendre en compte la décroissance du potentiel. Nous verrons dans la suite comment est initialisée cette matrice

- 

$$N_i = \begin{bmatrix} \delta i, 1 \\ \delta i, 2 \\ \vdots \\ \delta i, R \end{bmatrix} \in \mathcal{M}_{R,1}(\mathbb{R})$$

$$\text{où : } \forall (i, j) \in \llbracket 1, R \rrbracket^2, \delta i, j = \begin{cases} 1 & \text{si } i=j \\ 0 & \text{sinon} \end{cases}$$

## 0.7 Modélisation simplifiée

### 0.7.1 Présentation :

Une première modélisation [4] est celle où le fonctionnement des neurones est binaire, c'est-à-dire lorsque le neurone a été dépolarisé à l'étape ultérieure (au temps t) alors son potentiel  $V_i(t)$  vaut 0 à l'étape suivante (au temps t+1).

On modélise également de manière très simple la décroissance du potentiel après une dépolarisation par un coefficient k (biologiquement  $k \in [0.9, 0.95]$ ). Ainsi :

$$V_i(t+1) = f(k(1 - d_i(t))V_i(t) + \sum_{k \in P_i} (d_k(t) * V_k(t))) \quad (1)$$

où :

- $d_i = \begin{cases} 1 & \text{si le neurone } i \text{ a déchargé à l'instant } t \\ 0 & \text{sinon} \end{cases}$
- f la fonction activatrice choisie
- $P_i = \{i_1, i_2, i_3, \dots\}$  l'ensemble des indices des neurones connectés au neurone i

On adapte ce modèle sous forme matricielle. On obtient alors :

$$T_t = \begin{bmatrix} d_1(t) \\ d_2(t) \\ \vdots \\ d_R(t) \end{bmatrix}$$

Si le neurone i a déchargé au temps t alors  $ti, 1 = k$ , sinon  $ti, 1 = 0$ . Ce qui signifie qu'au temps t+1 on prendra en compte ( si  $ti, 1 = k$  ) ou non ( si  $ti, 1 = 0$  ) le potentiel du neurone du temps t. Autrement dit, la matrice  $T_t$  permet de connaître l'activité des neurones à l'étape ultérieure (lesquels sont actifs, ont déchargés, et lesquels ne le sont pas).

Finalement on obtient la formule suivante :

$$\boxed{\forall i \in \llbracket 1, R \rrbracket, V_i(t+1) = f(L_{C,i} * (V_t * (T_t + (-1)^{t_{i,1}} * N_i)))} \quad (2)$$



**Explications :** Dans l'équation 2 la partie  $(-1)^{t_{i,1}} * N_i$  traduit la partie  $(1 - d_i(t))$  de la formule 1. Tandis que la partie  $T_t$  traduit la partie  $d_k$  pour  $k \in P$ . En fait  $V_t * T_t$  (équation 2) est équivalent à  $\sum_{k \in P} (d_k * V_k(t))$  (équation 1). Mais lorsque que le neurone considéré i n'a pas déchargé alors  $t_{i,1} = 0$ ,

on obtiendrait donc la somme suivante :  $\sum_{k=1, k \neq i}^R (d_k * V_k(t)) + 0 * V_i(t)$ .

Ce qui ne correspond pas à l'équation 1 où on prend en compte le potentiel du neurone i des étapes précédentes s'il n'a pas déjà déchargé. Dans ce cas, sommer  $T_t$  et  $(-1)^{t_{i,1}} * N_i$  avant d'effectuer la multiplication par  $V_i(t)$  permet de s'assurer que :

- si le neurone i a déchargé, alors  $(-1)^{t_{i,1}} = -1$  puis  $T_t + (-1)^{t_{i,1}} * N_i$  permet d'avoir  $[T_t + (-1)^{t_{i,1}} * N_i]_{i,1} = 0$ , c'est-à-dire avoir  $0 * V_i(t)$  dans la somme des potentiels.
- si le neurone i n'a pas déchargé, alors  $(-1)^{t_{i,1}} = 0$  puis  $T_t + (-1)^{t_{i,1}} * N_i$  permet d'avoir  $[T_t + (-1)^{t_{i,1}} * N_i]_{i,1} = 1$ , c'est-à-dire avoir  $1 * V_i(t)$  dans la somme des potentiels (la multiplication par k est faite ultérieurement).

Pour des explications plus concrètes, voir l'exemple réalisé ci-dessous.

### 0.7.2 Fonctionnement algorithmique

- (1) : Initialiser la matrice C de manière aléatoire telle que les éléments de la diagonale soient égaux à k
- (2) : Initialiser la matrice  $V_0$
- (3) : Initialiser à 1 toute les valeurs de  $T_0$
- (4) : Calculer la sortie des R neurones du réseau.  
On pose  $\{j_1, j_2, \dots, j_n\}$  les neurones dont la sortie est supérieure au seuil
- (5) : Remplacer pour tout  $i \in \llbracket 1, R \rrbracket$   $t_{i,1}$  par  $\begin{cases} 1 & \text{si } i \in \{j_1, j_2, \dots, j_n\} \\ 0 & \text{sinon} \end{cases}$
- (6) : Mettre à jour la valeur du potentiel des R neurones du réseau
- (7) : Recommencer à l'étape (4)

### 0.7.3 Exemple

On considère le réseau de neurones représenté sur la figure 7

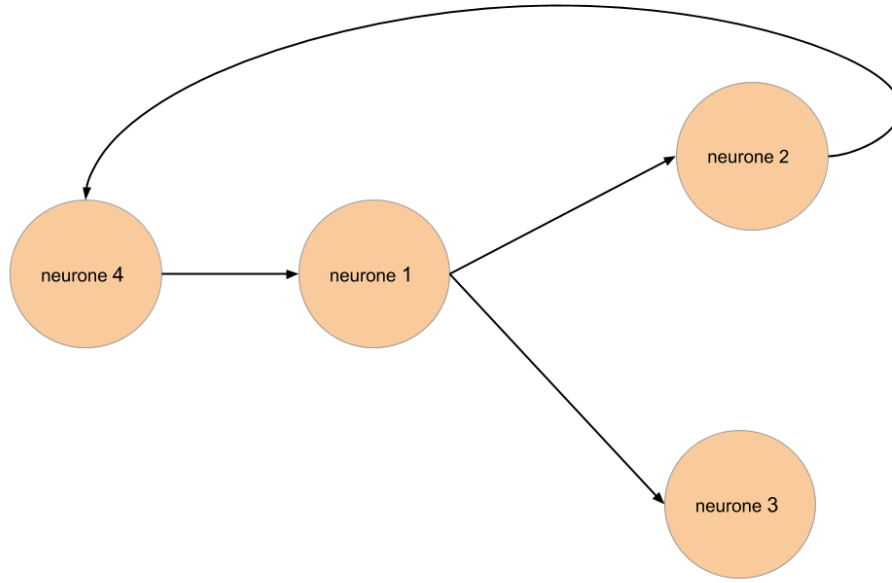


Figure 7: exemple de fonctionnement de la modélisation

Supposons que les neurones 3 et 4 ont déchargés à l'étape  $t=0$ .  
On doit maintenant calculer la sortie de tous les neurones du réseau. Commençons par le neurone 1. Selon l'équation 1, on doit obtenir (on considérant, pour simplifier, que la fonction activatrice est  $f : x \mapsto x$ ) :

- $d_1(0) = 0$  car le neurone 1 n'a pas été dépolarisé à au temps  $t=0$
- $d_4(0) = 1$  car le neurone a été dépolarisé, par hypothèse, au temps  $t=0$

D'où :

$$V_1(t=1) = k(1-0)V_1(0) + 1 * V_4(0) \quad (3)$$

Retrouvons ce résultat à l'aide de l'équation matricielle 2.  
Tout d'abord on définit nos matrices :

$$V_0 = \begin{bmatrix} V_1(0) & 0 & 0 & 0 \\ 0 & V_2(0) & 0 & 0 \\ 0 & 0 & V_3(0) & 0 \\ 0 & 0 & 0 & V_4(0) \end{bmatrix}, C = \begin{bmatrix} k & 0 & 0 & 1 \\ 1 & k & 0 & 0 \\ 1 & 0 & k & 0 \\ 0 & 1 & 0 & k \end{bmatrix}, T_0 = \begin{bmatrix} d_1(0) \\ d_2(0) \\ d_3(0) \\ d_4(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Selon la formule 3, le potentiel du neurone 1 au temps  $t=0$  est pris en compte dans la calcul au temps  $t=1$ ; car le neurone 1 n'a pas encore déchargé. Cette opération est assurée dans le calcul matricielle (équation 2) à l'aide de la matrice  $N_i$  intervenant dans  $T_t + (-1)^{t_{i,1}} * N_i$ . Illustrons ci-dessous son utilité.

Supposons que nous n'ayons pas introduit la matrice  $N_i$  dans l'équation 2. On obtiendrait alors la formule suivante :

$$V_1(1) = L_{C,1} * (V_0 * T_0)$$

Autrement dit,

$$V_1(t=1) = \begin{bmatrix} k & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} V_1(0) & 0 & 0 & 0 \\ 0 & V_2(0) & 0 & 0 \\ 0 & 0 & V_3(0) & 0 \\ 0 & 0 & 0 & V_4(0) \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$V_1(t=1) = \begin{bmatrix} k & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 0 * V_1(0) \\ 0 * V_2(0) \\ 1 * V_3(0) \\ 1 * V_4(0) \end{bmatrix}$$

$$V_1(t=1) = k * 0 * V_1(0) + 0 * 0 * V_2(0) + 0 * 1 * V_3(0) + 1 * 1 * V_4(0)$$

$$V_1(t=1) = V_4(0)$$

Ce qui n'est pas correcte puisque le potentiel de  $V_1(t=0)$ , n'est pas pris en considération dans  $V_1(t=1)$

Désormais, observons le résultat en introduisant la matrice  $N_i$ . Calculons, dans un premier temps  $T_0 + (-1)^{t_{1,1}} * N_1$  :

$$T_0 + (-1)^{t_{1,1}} * N_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} + (-1)^0 * \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

On constate ici que l'intervention de  $N_1$  dans le calcul permet d'avoir  $t_{1,1} = 1$  et non  $t_{1,1} = 0$  comme vu précédemment. Ce changement permet, comme nous allons le voir, de prendre en compte le potentiel  $V_1(0)$  dans le calcul de  $V_1(t=1)$ . En fait, nous sommes obligés d'introduire  $N_1$  car nous ne pouvons changer directement la valeur de  $T_0$ , sans quoi les futurs résultats des sorties de tous les autres neurones seraient affectés (car  $t_{1,1} = 1$  signifie que le neurone 1 à déjà déchargé, ce qui n'est pas le cas ici). Dès lors,  $N_1$  permet d'altérer temporairement et localement le résultat de  $V_1(t=1)$ , sans affecter l'ensemble du réseau.

On se retrouve donc avec :

$$V_1(t=1) = \begin{bmatrix} k & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} V_1(0) & 0 & 0 & 0 \\ 0 & V_2(0) & 0 & 0 \\ 0 & 0 & V_3(0) & 0 \\ 0 & 0 & 0 & V_4(0) \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$V_1(t=1) = \begin{bmatrix} k & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} V_1(0) * 1 \\ V_2(0) * 0 \\ V_3(0) * 1 \\ V_4(0) * 1 \end{bmatrix}$$

$$V_1(t=1) = 1 * k * V_1(0) + 0 * 0 * V_2(0) + 0 * 1 * V_3(0) + 1 * 1 * V_4(0)$$

$$V_1(t=1) = k * V_1(0) + V_4(0)$$

Ce qui est bien le résultat trouvé à l'aide de la formule 1.

Avant de poursuivre les calculs, il nous faut remplacer  $V_1(t=0)$  par  $V_1(t=1)$  dans la matrice  $V_1$

Une fois la sortie du neurone 1 calculée, il y a deux possibilités :

- Si  $V_1(t=1)$  est supérieur au seuil alors  $t_{1,1} = 1$
- Sinon  $t_{1,1} = 0$

On refait ensuite les mêmes calculs pour tous les autres neurones :

$$V_2(t=1) = 0 * V_2(0) + 0 * V_1(0) + 0 * V_3(0) + 0 * V_4(0)$$

$$V_3(t=1) = k * V_3(0) + 0 * V_1(0) + 0 * V_2(0) + 0 * V_4(0)$$

$$V_4(t=1) = k * V_4(0) + 0 * V_1(0) + 0 * V_2(0) + 0 * V_3(0)$$

On met alors à jour la matrice  $V_t$  :

$$V_1 = \begin{bmatrix} V_1(1) & 0 & 0 & 0 \\ 0 & V_2(1) & 0 & 0 \\ 0 & 0 & V_3(1) & 0 \\ 0 & 0 & 0 & V_4(1) \end{bmatrix}$$

Supposons dans la suite que le seuil de tout les neurones soit franchis. Alors :

$$T_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

On calcul de nouveau la sortie de tous les neurones. Pour le neurone 1 on obtient :

$$T_1 + (-1)^{t_{1,1}} * N_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + (-1)^1 * \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

On observe ici que la matrice  $N_1$  a permis de passer le coefficient  $t_{1,1}$  de 1 à 0 (le temps du calcul de  $V_1(t=2)$ ). Cela est bien le résultat souhaité, puisque le neurone 1 ayant déchargé au temps  $t=1$ , on ne doit pas prendre en compte ce potentiel dans le calcul de  $V_1(t=2)$ . D'où :

$$V_1(t=2) = [k \quad 0 \quad 0 \quad 1] * \begin{bmatrix} V_1(1) & 0 & 0 & 0 \\ 0 & V_2(1) & 0 & 0 \\ 0 & 0 & V_3(1) & 0 \\ 0 & 0 & 0 & V_4(1) \end{bmatrix} * \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$V_1(t=2) = k * 0 * V_1(1) + 0 * 1 * V_2(1) + 0 * 1 * V_3(1) + 1 * 1 * V_4(1)$$

De même :

$$V_2(t=2) = k * 0 * V_2(1) + 1 * 1 * V_1(1) + 0 * 1 * V_3(1) + 0 * 1 * V_4(1)$$

$$V_3(t=2) = k * 0 * V_3(1) + 1 * 1 * V_1(1) + 0 * 1 * V_2(1) + 0 * 1 * V_4(1)$$

$$V_4(t=2) = k * 0 * V_4(1) + 0 * 1 * V_1(1) + 1 * 1 * V_2(1) + 0 * 1 * V_3(1)$$

Alors :

$$V_2 = \begin{bmatrix} V_1(2) & 0 & 0 & 0 \\ 0 & V_2(2) & 0 & 0 \\ 0 & 0 & V_3(2) & 0 \\ 0 & 0 & 0 & V_4(2) \end{bmatrix}$$

Et en supposant que seul le potentiel du neurone 4 est dépassé le seuil, on obtient :

$$T_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

## 0.8 Modélisation tenant compte du temps de décroissance du potentiel

### 0.8.1 Défaut de la modélisation 0.7

La modélisation précédente (modélisation 0.7) procédait à une "décroissance binaire" du potentiel d'un neurone après une dépolarisation. En effet, tant que le neurone n'avait pas déchargé, son potentiel était pris en compte et ce même après que le temps de décroissance biologique (vu sur la figure

3) ait été passé. Cette "décroissance binaire" est simplement dû au pas de temps de 1 milliseconde, temps pendant lequel le neurone ayant dépolarisé ne prend pas en compte les sollicitations de ses entrées.

En fait, comme le montre la figure 8, au bout de 1 ms le potentiel à prendre en compte est nul (si on ramène le minimum de la courbe biologique à 0).

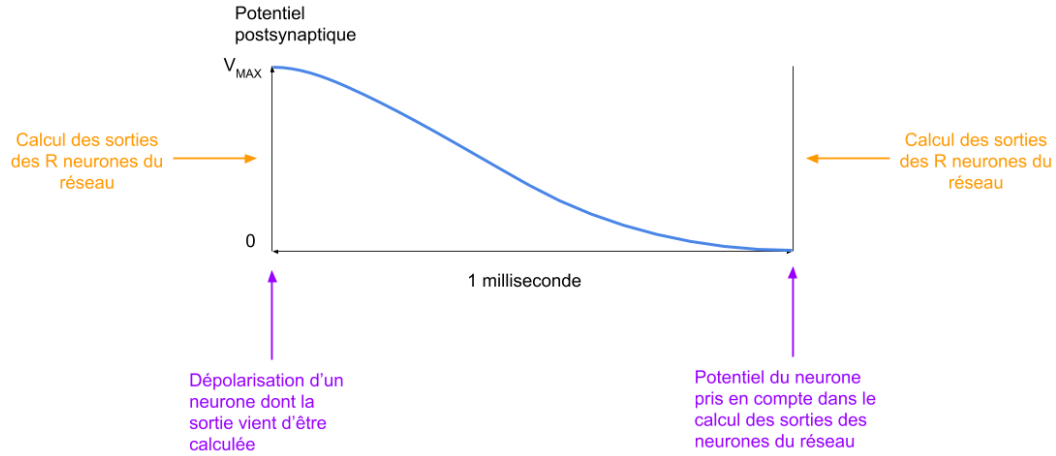


Figure 8: Illustration du problème du pas de temps de la modélisation 0.7

L'autre problème sous-jacent à ce pas de temps est qu'en calculant la sortie des neurones toutes les millisecondes, on restreint la prise en compte de la sommation temporelle que fait le neurone biologique. Dans le cas réelle, les valeurs des entrées des neurones peuvent évoluer fortement en 1 milliseconde. De ce fait, le potentiel du neurone aura fortement diminué après 1 milliseconde; ce qui signifie le cumul du potentiel précédent et de la somme des entrées sera moins susceptible de déclencher une dépolarisation.

Ainsi, prendre en compte la décroissance du potentiel avec un pas de temps de 1 milliseconde n'est pas possible. A partir de maintenant, il faut considérer un nouveau pas de temps  $\delta_t$  comme défini sur la figure 9

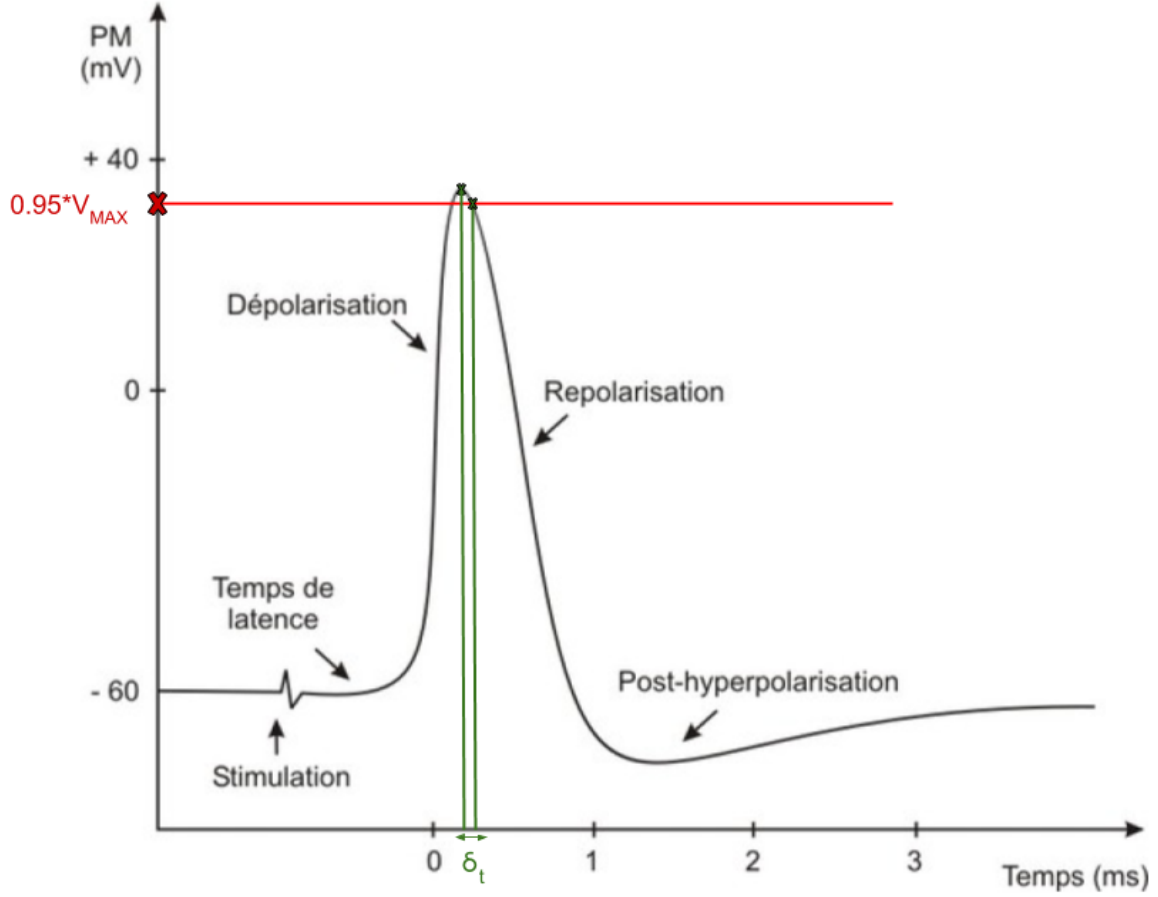


Figure 9: Définition du nouveau pas de temps de la simulation

On trouve alors :

$$\delta_t = 0.1 \text{ milliseconde} \quad (4)$$

Ainsi, si un neurone  $i$  se dépolarise au temps  $t$ , nous supposons qu'au temps  $t + \delta_t$  le neurone  $j$  connecté en réception au neurone  $i$  perçoit  $\alpha * V_{max}$  (où  $\alpha$  est un coefficient signifiant que le neurone en réception ne reçoit jamais le potentiel émis par le neurone en transmission du fait de la conversion du signal électrique en signal chimique et des différents types de récepteurs synaptique). Puis la sortie du neurone  $i$ , en transmission, ne sera pas calculé pendant 1 milliseconde, soit jusqu'au temps  $t + 10 * \delta_t$ . Ne pas calculer la sortie du neurone, revient à considérer que le neurone ne prend pas en compte les valeurs de ses entrées (car le neurone s'est dépolarisé). Pendant cette milliseconde, le potentiel du neurone  $i$  va décroître pour atteindre 0, moment après lequel le neurone va de nouveau prendre en compte les valeurs de ses entrées.

Quant au neurone  $j$ , son potentiel va décroître d'une petite quantité, à chaque nouvelle étape (c'est-à-dire tous les  $\delta_t$  milliseconde), jusqu'à atteindre 0 mV.

Notons que nous ne prendrons pas en compte la post-hyperpolarisation (défini sur la figure 9) qui correspond à une descente du potentiel en dessous de sa valeur au repos (ce qui correspond pour nous à 0 mV). Ainsi, une fois que le potentiel des neurones sera en tout temps positif ou nul.

## 0.8.2 Présentation

On introduit alors la matrice  $\Delta_V$  telle que :

$$\Delta_V = \begin{bmatrix} -\delta_V & 0 & 0 & \dots & 0 \\ 0 & -\delta_V & & & \vdots \\ \vdots & 0 & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \vdots & \vdots & & & 0 \\ 0 & 0 & \dots & 0 & -\delta_V \end{bmatrix} \in \mathcal{M}_R(\mathbb{R})$$

Dans ce cas :

$$\boxed{\forall i \in \llbracket 1, R \rrbracket, V_i(t + \delta_t) = f(L_{C,i} * [(V_t + \Delta_V) * (T_t + N_i)])} \quad (5)$$

**Remarquons** que l'introduction de la matrice  $\Delta_V$  permet de prendre en compte une modélisation où le potentiel de sortie des neurones après une dépolarisation dépend du neurone considéré; bien que biologiquement le potentiel de sortie des neurones après une dépolarisation est le même.

### 0.8.3 Fonctionnement algorithmique

Pour éviter certaines confusions, nous noterons parfois  $[A]_{i,i}$  pour le coefficient d'indice  $(i, i)$  de la matrice A

- (1) : Initialiser la matrice C de manière aléatoire telle que les éléments de la diagonale soient égaux à 1
- (2) : Initialiser la matrice  $V_0$
- (3) : Initialiser à 1 toute les valeurs de  $T_0$
- (4) : Si  $V_i(t) - \delta_V < 0$  alors  $V_i(t) \leftarrow +\delta_V$
- (5) : Pour tout  $i \in \llbracket 1, R \rrbracket$ 
  - Calculer  $V_i(t + \delta_t)$  si  $V_i(t) \neq V_{max}$   
Dans ce cas, on pose  $\{j_1, j_2, \dots, j_n\}$  les neurones tel que  $V_i(t + \delta_t) = V_{max}$
  - $V_i(t + \delta_t) = V_{max}$  sinon  
Dans ce cas,  $c_{i,i} \leftarrow c_{i,i} + 1$
- (6) : Si  $c_{i,i} = 10$  alors  $\begin{cases} c_{i,i} \leftarrow 1 \\ V_i(t + \delta_t) \leftarrow 0 \\ [\Delta_V]_{i,i} \leftarrow -\delta_V \end{cases}$
- (7) : Mettre à jour les valeurs des potentiels dans  $V_t$
- (8) : Remplacer pour tout  $i \in \llbracket 1, R \rrbracket$   $t_{i,1}$  par  $\begin{cases} 1 \text{ si } i \in \{j_1, j_2, \dots, j_n\} \\ 0 \text{ sinon} \end{cases}$
- (9) : Pour tout  $i \in \{j_1, j_2, \dots, j_n\}$ ,  $[\Delta_V]_{i,i} \leftarrow 0$
- (10) : Recommencer à l'étape (4)

**Remarquons que :**

- Incrémenter  $c_{i,i}$  une fois que le neurone i à dépolarisé, permet de savoir combien il reste d'étape avant que le neurone soit de nouveau opérationnel. En fait, il faut  $\frac{1}{\delta_t} = 10$  étapes pour qu'il se soit écoulé 1 milliseconde depuis l'instant où le neurone i a dépolarisé.  
De plus, changer la valeur de  $c_{i,i}$  n'affecte pas le calcul des autres potentiels car :

- En temps normale cette valeur n'intervient que dans le calcul de  $V_i(t)$

- Et dans cette modélisation, on ne calcul plus le potentiel du neurone  $i$  pendant 1 milliseconde, donc cette valeur n'intervient tout simplement plus.
- Fixer les valeurs des neurones s'étant dépolarisé à  $V_{max}$  durant toutes leurs périodes d'inactivité de 1 ms est un choix fait pour que l'algorithme fonctionne. La réalité est tout autre, puisque biologiquement le potentiel du neurone décroît pour atteindre 0. Mais cela importe peu car pendant cette milliseconde, le neurone ne peut recevoir, ni envoyer. Ainsi, son potentiel interne n'importe pas dans une simulation qui tente de rendre compte des interactions globales entre neurones.
- Du fait de l'algorithme, l'équation 5 est  $f(L_{C,i} * [(V_t + \Delta_V) * (T_t + N_i)])$  au lieu de  $f(L_{C,i} * [(V_t + \Delta_V) * (T_t + (-1)^{t_{i,1}} * N_i)])$  car le  $(-1)^{t_{i,1}}$  intervenait seulement lorsque le pas de temps était d'une milliseconde. Dans notre simulation, le potentiel n'est tout simplement plus calculé lorsque le neurone a dépolarisé et que le temps depuis cet instant est plus petit qu'une milliseconde.

### 0.8.4 Exemple

On considère le même réseau de neurone que celui représenté sur la figure 7. Supposons qu'au temps  $t=1$  les neurones 3 et 4 ont déchargés. Alors :

$$\begin{cases} V_3(1) = V_{max} \\ V_4(1) = V_{max} \end{cases}$$

Dans ce cas, on obtient :

$$V_1 = \begin{bmatrix} V_1(1) & 0 & 0 & 0 \\ 0 & V_2(1) & 0 & 0 \\ 0 & 0 & V_{max} & 0 \\ 0 & 0 & 0 & V_{max} \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \Delta_V = \begin{bmatrix} -\delta_V & 0 & 0 & 0 \\ 0 & -\delta_V & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, T_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

A l'instant  $t=2$ , on doit calculer la sortie de tout les neurones. Commençons par le neurone 1 :

$$T_1 + N_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Pour la suite, nous supposons que pour tout  $i \in \llbracket 1, 4 \rrbracket$ ,  $V_i(0) - \delta_V > 0$ . On obtient alors :

$$V_1(t=2) = [1 \ 0 \ 0 \ 1] * \left( \begin{bmatrix} V_1(1) & 0 & 0 & 0 \\ 0 & V_2(1) & 0 & 0 \\ 0 & 0 & V_{max} & 0 \\ 0 & 0 & 0 & V_{max} \end{bmatrix} + \begin{bmatrix} -\delta_V & 0 & 0 & 0 \\ 0 & -\delta_V & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \right) * \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$V_1(t=2) = 1 * (V_1(1) - \delta_V) + V_4(1)$$

$$V_1(t=2) = 1 * (V_1(1) - \delta_V) + V_{max}$$

De même :

$$V_2(t=2) = 1 * (V_2(1) - \delta_V) + 0 * V_1(1) + 0 * V_3(1) + 0 * V_4(1)$$

Puisque les neurones 3 et 4 se sont dépolarisé à l'étape précédente, il entame leurs période d'inactivité de  $i$  ms. Dans ce cas :

$$V_3(t=2) = V_{max}$$

$$V_4(t=2) = V_{max}$$



Supposons que seul  $V_1(2)$  soit supérieur au seuil. Alors :

$$V_2 = \begin{bmatrix} V_{max} & 0 & 0 & 0 \\ 0 & V_2(2) & 0 & 0 \\ 0 & 0 & V_{max} & 0 \\ 0 & 0 & 0 & V_{max} \end{bmatrix}$$

$$T_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\Delta_V = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\delta_V & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

On recalcule alors la sortie de tous les neurones. On met à jour la matrice C :

$$C = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix}$$

Puis, comme le neurone 1 s'est dépolarisé à l'étape précédente on a :

$$V_1(t = 3) = V_{max}$$

Et,

$$V_3(t = 3) = V_{max}$$

$$V_4(t = 3) = V_{max}$$

Le seul potentiel qui soit réellement à calculer est celui du neurone 2 :

$$V_2(t = 3) = 1 * (V_2(2) - \delta_V) + 1 * V_1(2) + 0 * V_3(2) + 0 * V_4(2)$$

$$V_2(t = 3) = 1 * (V_2(2) - \delta_V) + 1 * V_{max} + 0 * V_{max} + 0 * V_{max}$$

Alors :

$$V_3 = \begin{bmatrix} V_{max} & 0 & 0 & 0 \\ 0 & V_2(3) & 0 & 0 \\ 0 & 0 & V_{max} & 0 \\ 0 & 0 & 0 & V_{max} \end{bmatrix}$$

Puis on recommence. On obtient alors :

$$C = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 1 & 0 & 3 \end{bmatrix}$$

Les neurones 1, 2 et 3 resteront inactifs jusqu'à ce que leurs millisecondes d'inactivités soient passées (lorsque  $c_{i,i} = 10$ ). Durant cette période on aura toujours  $t_{i,1} = 0$ ; de ce fait il ne sera pris en compte dans aucun calcul de potentiel.

### 0.8.5 Remarque sur la simulation synchrone

Une simulation asynchrone permettrait théoriquement d'obtenir pour un neurone en particulier la courbe de décroissance du potentiel de la figure 6 car ce type de simulation fait qu'un neurone peut demeurer sans être stimulé par ses entrées pendant plusieurs étapes; ce qui dans la réalité n'est pas possible. En effet, chaque neurone possède des millions de connexions et est donc stimulé à chaque instant. Ce faisant la courbe de la figure 6 n'est que théorique. En fait, expérimentalement cette courbe est obtenue en isolant un neurone et en le soumettant à des stimulations contrôlée à l'aide d'électrodes [1]. Dans la réalité la probabilité que le potentiel d'un neurone est le temps de diminuer jusqu'à sa valeur au repos est très faible. On obtiendrait alors une courbe comme celle présentée sur la figure 10

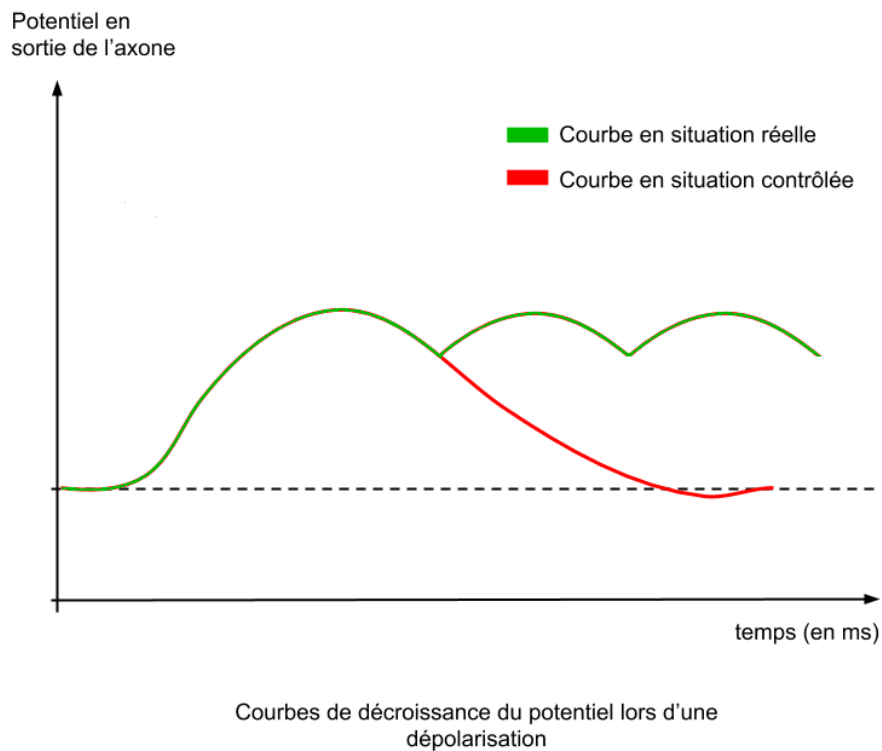


Figure 10:

Ce phénomène, difficilement remarquable dans une simulation asynchrone, est rendu visible dans une simulation synchrone qui de plus possède un pas de temps petit.

## 0.9 Modélisation tenant compte du poids des connexions

### 0.9.1 Défaut des modélisations 0.7 et 0.8

Dans les modélisations précédentes nous considérons qu'une fois le potentiel d'action libéré par le neurone, l'activation des récepteurs synaptique se faisait de manière uniforme. Autrement dit, on a considéré que  $P_B = P_C = P_D = P_E = P_F$  où  $(P_i)_i$  sont définis sur la figure 11

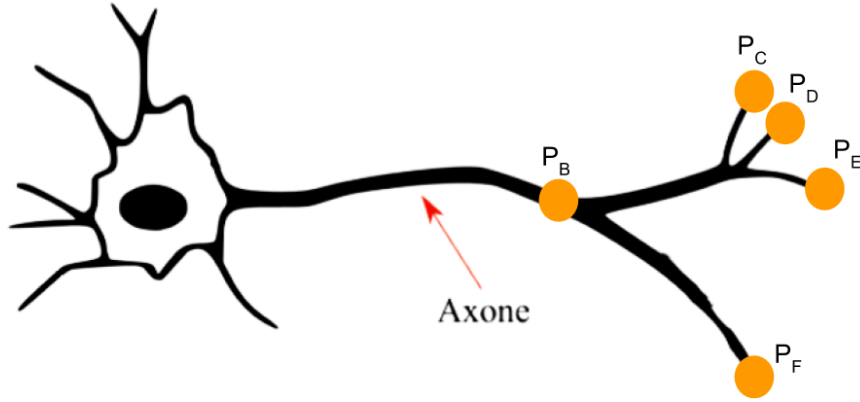


Figure 11:

Mais cette hypothèse est fausse. Rappelons-nous que la circulation du courant électrique à la sortie de l'axone est assurée, en outre, par des ions. Or ces ions ne se répartissent pas de la même manière dans les embranchements selon leurs tailles. De plus, un neurone est fortement connecté à un autre s'il possède de nombreuses synapses en direction de ce neurone. Cela revient donc à considérer un unique gros embranchement en direction de ce neurone, captant donc plus d'ions.

### 0.9.2 Présentation

Ainsi, une fois le nombre  $N_i$  de neurones connectés au neurone  $i$  considéré, on associe à chaque neurone un poids de connexion  $w_{i,j}$ .  $w_{i,j}$  représentent la qualité de la connexion du neurone  $i$  vers le neurone  $j$ . Ainsi, si  $w_{i,j}$  existe,  $w_{j,i}$  n'existe pas nécessairement et s'il existe, il n'est pas égale à  $w_{i,j}$  dans le cas général.

- $\forall (i, j) \in \llbracket 1, R \rrbracket * \llbracket 1, N_i \rrbracket, w_{i,j} \in [0, 1]$
- $\forall i \in \llbracket 1, R \rrbracket, \sum_{j=1}^{N_i} (w_{i,j}) = 1$

Autrement dit, comme le montre la figure 12, le potentiel en sortie de l'axone est  $V_i(t)$  et celui en direction du neurone  $j_1$  est  $w_{i,j_1} * V_i(t)$

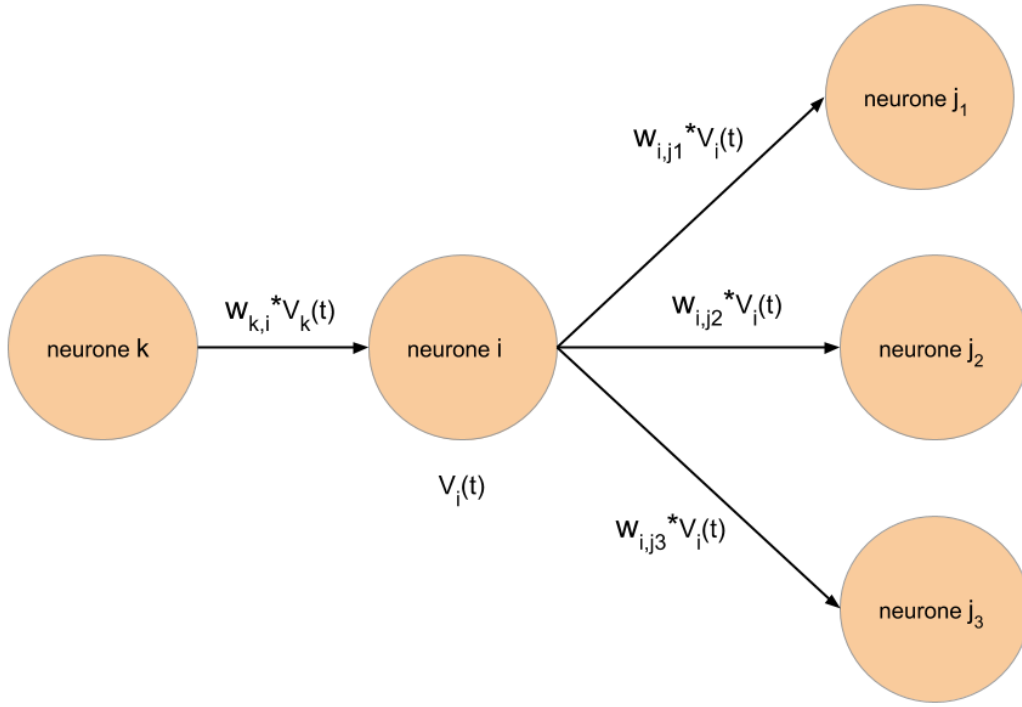


Figure 12:

La modélisation mathématique est la même que la modélisation précédente à la différence de  $C$  :

$$C = \begin{bmatrix} 1 & w_{2,1} & w_{3,1} & \dots & w_{R,1} \\ w_{1,2} & 1 & & & \vdots \\ \vdots & w_{2,3} & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \vdots & \vdots & & & w_{R,R-1} \\ w_{1,R} & w_{2,R} & \dots & w_{R-1,R} & 1 \end{bmatrix} \in \mathcal{M}_R(\mathbb{R})$$

où pour  $(i, j) \in \llbracket 1, R \rrbracket^2, i \neq j, w_{i,j}$  est éventuellement nul (absence de connexion).

Les valeurs de la diagonale sont, selon la modélisation utilisée, soit égales à 1, soit à  $f_d^i(t)$

Ces valeurs  $((w_{i,j}))$  sont ici invariantes dans le temps.

## 0.10 Modélisation tenant compte des potentiels excitateurs et inhibiteurs

### 0.10.1 Autre défaut de la modélisation 0.7

Jusqu'à présent, le potentiel des entrées de nos neurones (nos dendrites) pouvaient soit prendre des valeurs positives ou négatives de manière arbitraire. En fait, cela dépend de la fonction activatrice choisie (par exemple pour une fonction seuil, toutes les valeurs de sortie auraient été positives ou nulles). Cela ne tient donc pas compte du fait que certains récepteurs des neurones peuvent soit augmenter le potentiel soit le diminuer.

### 0.10.2 Présentation

Dans cette modélisation, on s'assure tout d'abord que la fonction activatrice donne un résultat positif si le seuil est franchi, et nul sinon. Dans ce cas, il suffit de remplacer de manière arbitraire certain  $w_{i,j}$  par  $-w_{i,j}$ . Dès lors, les poids positifs jouent le rôle de récepteurs excitateurs et les poids négatifs de récepteurs inhibiteurs.

## 0.11 Modélisation tenant compte de la plasticité synaptique

### 0.11.1 Défaut des modélisations précédentes

Pour l'instant, aucune de nos modélisation ne tenaient compte de la plasticité synaptique, c'est-à-dire la capacité que possède un neurone à augmenter l'efficacité d'une connexion avec un autre neurone. Il existe deux types de plasticité [2] :

- une à court terme (de quelques secondes à plusieurs minutes)
- une à long terme (de quelques jours à plusieurs mois, ou même indéfiniment)

Nous n'étudierons pas ici la plasticité à long terme car elle prend en compte un trop grand nombre de paramètres. Donc cette modélisation sera difficile à exploiter.

### 0.11.2 Plasticité à court terme

La plasticité à court terme se traduit par deux phénomènes notables :

- Lorsqu'un neurone est sollicité à plusieurs reprises (lorsqu'il y a des dépolarisations rapprochées dans le temps), alors le nombre de neurorécepteurs émis, pour un même potentiel d'action, par chaque synapse augmente (le calcium  $Ca^{2+}$  s'accumulent dans les synapses sans avoir eu le temps d'être évacué).
- Mais lorsque ces sollicitations deviennent trop récurrentes, le nombre de neurorécepteurs émis, pour un même potentiel d'action, diminue (les réserves en neurorécepteurs s'épuisent).

La figure 13 représente ces deux phénomènes (qui ne sont que deux parties d'une même courbe). L'axe des temps sera expliqué ultérieurement.

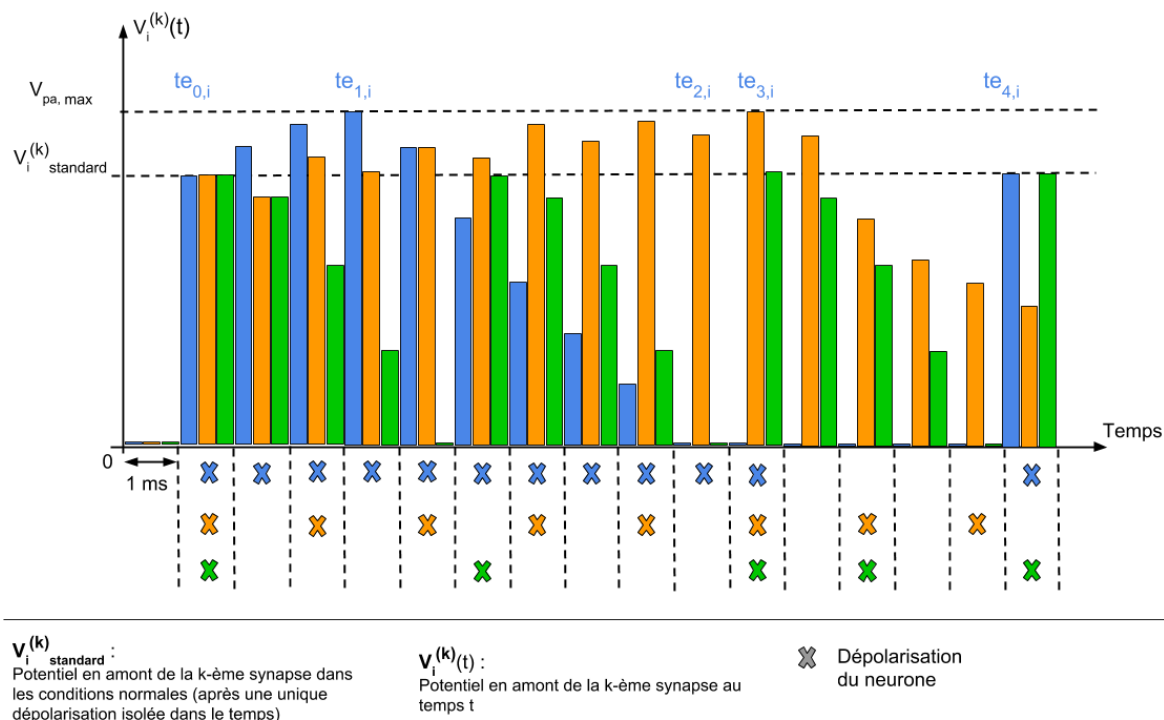


Figure 13: Représentation qualitative des effets de la plasticité à court terme sur le potentiel de sortie d'un neurone

Plus le temps entre deux dépolarisations est grand, plus les ions  $Ca^{2+}$  libérés lors de la première dépolarisation ont eu le temps d'être évacués. Ce faisant, il faut un nombre de dépolarisations successives plus important pour atteindre le maximum permis par le neurone.

Une fois ce maximum atteint, le nombre de vésicules présynaptiques (contenant les neurorécepteurs) commence à diminuer (car elles ont déjà libérées les neurorécepteurs précédemment). Cette décroissance est plus lente lorsque le temps entre deux dépolarisations est grand, car si une dépolarisation induit une perte temporaire de vésicules présynaptiques, l'espacement temporel entre deux dépolarisations permet aussi d'en recycler d'autres, qui seront alors utilisées lors de la prochaine dépolarisation. On peut voir l'influence du temps entre deux dépolarisations grâce aux courbes bleue et orange sur la figure 13.

La remontée vers un potentiel normal est également plus rapide lorsque le temps entre deux dépolarisations est grand pour la même raison.

Toutefois, lorsque cet espacement temporel devient trop grand, lors de la seconde dépolarisation, tous les ions  $Ca^{2+}$  auront été évacués et toutes les vésicules présynaptiques auront été recyclées. Ce faisant, le potentiel de sortie est le même que lors de la première dépolarisation. Ce phénomène est illustré sur la figure 13 par la courbe verte.

Lorsque le neurone subit une dépolarisation au bout d'un certain temps, le potentiel de sortie demeure nul car il n'y a plus assez de vésicules présynaptiques recyclées (l'espacement temporel entre deux dépolarisations étant trop court).

Il y a donc deux phénomènes de décroissance à distinguer :

- la décroissance du potentiel de sortie après un grand nombre de dépolarisation (une fois  $V_{pa, max}$  atteint).
- la décroissance de la concentration en  $Ca^{2+}$  qui s'évacue du synapse au cours du temps, ce qui a pour effet de ramener le potentiel de sortie à la normale (état où la concentration en  $Ca^{2+}$  et le nombre de vésicules présynaptiques est le même que dans une synapse d'un neurone ayant comme potentiel d'entrées celui du repos; c'est-à-dire un neurone qui n'a pas déchargé depuis un certain temps). Cette décroissance se traduit sur la courbe violette de la figure 13 par une diminution du potentiel de sortie lorsqu'il n'a pas encore atteint  $V_{pa, max}$  (ce qui explique que la courbe violette met plus de temps à atteindre  $V_{pa, max}$  que la courbe orange) puis par une

augmentation (une courbe dont les dépolarisations serait espacée de 3ms aurait une croissance bien plus forte que celle de la courbe violette où les dépolarisation sont espacée de 2 ms, par exemple).

**Remarquons** que les courbes de la figure 13 sont purement qualitatives, tout comme l'espacement temporel.

### 0.11.3 Présentation

Nous tentons ici de formaliser les phénomènes précédemment décrits.

- A chaque dépolarisation, le potentiel de sortie d'un neurone augmente, temporairement, d'une petite quantité, que nous noterons  $V_p$ , proportionnelle au potentiel d'action délivré lors de la dernière dépolarisation. En effet, plus le potentiel libéré lors d'une dépolarisation est grand plus le nombre de canaux calciques ouverts est grand, plus d'ions calcium  $Ca^{2+}$  s'accumulent dans la synapses, donc plus le nombre de vésicules libérées lors de la prochaine dépolarisation est grand (car toutes les vésicules présynaptiques ne sont pas évacuées lors d'une seule dépolarisation, donc la concentration en  $Ca^{2+}$  diminuera au cours de la dépolarisation mais ne sera pas nul lors de la prochaine dépolarisation si le temps entre les deux est assez court; somme toute le nombre d'ions  $Ca^{2+}$  restant après une dépolarisation dépend de l'intensité du potentiel d'action de cette dépolarisation). Ainsi, le potentiel de sortie du neurone  $i$  après une seconde dépolarisation consécutive est  $V_i(t+1) = V_i(t) + V_p$ . Or comme  $V_p$  est proportionnel à  $V_i(t)$ , il existe un coefficient  $\delta_i(t)$  tel que  $V_p = \delta_i(t) * V_i(t)$ . On obtient donc après une seconde dépolarisation consécutive  $V_i(t+1) = (1 + \delta_i(t)) * V_i(t)$ . En notant  $te_{1,i}$  le temps tel que le potentiel du neurone  $i$  atteint  $V_{pa,max}$  et en remarquant que le potentiel de sortie du neurone  $i$  ne peut pas être négatif, on a :

$$\forall t, \delta_i(t) \in [-1, \frac{V_{pa,max}}{V_i(te_{1,i}-1)} - 1]$$

En fait, la quantité à prendre en compte  $V_p$  est la même pour tous les neurones car tous les potentiels d'actions ont la même valeur en sortie de l'axone. Ainsi, il aurait été plus naturel de simplement écrire  $V_i(t+1) = V_i(t) + V_p$ . Mais avoir  $(1 + \delta_i(t)) * V_i(t)$  simplifie les calculs et la compréhension. De plus, cela ne modifie pas le résultat car entre les deux formules il n'y a qu'un facteur  $V_i(t)$  à prendre en compte puisque tous les potentiels  $V_i$  ont la même valeur après une dépolarisation.

- Cette augmentation du potentiel de sortie, débutant au temps  $te_{0,i}$ , reste valable jusqu'au temps  $te_{1,i}$  tel que  $V_i(te_{1,i}) = V_{max}$ . Après ce temps, le potentiel de sortie du neurone  $i$  décroît lentement après chaque nouvelle dépolarisation jusqu'à être nul au temps  $te_{2,i}$  et le restera jusqu'au temps  $te_{3,i}$ . Il faudra ensuite attendre jusqu'au temps  $te_{4,i}$  pour que lors d'une dépolarisation le potentiel de sortie soit le même que dans les conditions normales. Autrement dit, la quantité  $(1 + \delta_i(t))$  varie au cours du temps. Cela signifie que  $\delta_i(t)$  varie au cours du temps. Ainsi, pour la suite nous devons introduire deux quantités :

- $pas_p > 0$  la quantité que l'on ajoute à  $\delta_i(t)$  lorsque  $te_{0,i} \leq t < te_{1,i}$  ou  $te_{3,i} \leq t < te_{4,i}$
- $pas_m < 0$  (en principe  $pas_m = -pas_p$ ) la quantité que l'on ajoute à  $\delta_i(t)$  lorsque  $te_{1,i} \leq t < te_{2,i}$
- $pas = 0$  la quantité que l'on ajoute à  $\delta_i(t)$  lorsque  $te_{2,i} \leq t < te_{3,i}$

Ainsi, après une dépolarisation au temps  $t$  :

$$\forall t, \delta_i(t+1) = \begin{cases} \delta_i(t) + pas_p & \text{si } te_{0,i} \leq t+1 < te_{1,i} \text{ ou } te_{3,i} \leq t+1 < te_{4,i} \\ \delta_i(t) + pas_m & \text{si } te_{1,i} \leq t+1 < te_{2,i} \\ -1 & \text{si } te_{2,i} \leq t+1 < te_{3,i} \end{cases}$$

- Il ne nous reste plus qu'à prendre en compte la décroissance du nombre d'ion  $Ca^{2+}$  dans la synapse entre deux dépolarisations espacées dans le temps. Ainsi, on pose  $pas_{Ca} \in ]0, \min(pas_p, pas_m)[$

le nombre d'ion calcium  $Ca^{2+}$  évacués en 1 milliseconde qui est sensiblement le même pour toutes les synapses. Ce nombre ne dépend pas de l'intensité du potentiel d'action : la synapse ne peut évacuer qu'un nombre fixe d'ions à chaque instant, peu importe le nombre total de ces ions dans la synapse.

Ainsi, en tout temps :

$$\forall t, \delta_i(t+1) = \begin{cases} \delta_i(t) - pas_{Ca} & \text{si } te_{0,i} \leq t+1 < te_{1,i} \text{ ou } te_{3,i} \leq t+1 < te_{4,i} \\ 0 & \text{si } t \geq te_{4,i} \\ \delta_i(t) + pas_{Ca} & \text{sinon} \end{cases}$$

Jusqu'à présent, nous nous sommes référés au temps pour conditionner l'ajout ou le retrait d'une certaine quantité. Pour la suite, nous ne pouvons pas faire de même. Cela est plus simple de se référer à différent potentiel de référence pour connaître l'état antérieure du neurone, afin d'en déduire son prochain état.

On peut alors visualiser l'évolution de  $\delta_i(t)$  sous forme graphique comme le montre la figure 14

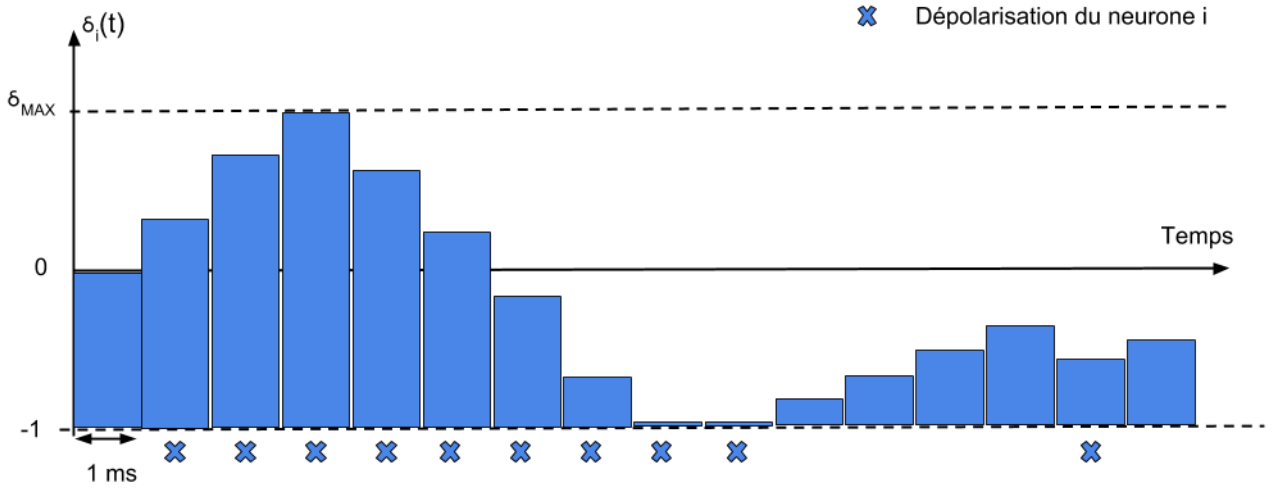


Figure 14: Évolution de  $\delta_i$  en fonction du temps et selon les dépolarisations du neurone

On traduit donc ces équations sous une forme matricielle. Pour ce faire, on pose :

- $$P = \begin{bmatrix} t_{1,1} * pas_1^{(k)} & 0 & 0 & \dots & 0 \\ 0 & t_{2,1} * pas_2^{(k)} & & & \vdots \\ \vdots & 0 & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \vdots & \vdots & & & 0 \\ 0 & 0 & \dots & 0 & t_{R,1} * pas_R^{(k)} \end{bmatrix} \in \mathcal{M}_R(\mathbb{R})$$

où  $\begin{cases} t_{i,1} \text{ est le coefficient de la matrice } T_t \text{ comme définie précédemment} \\ pas_R^{(k)} = \begin{cases} pas_p \\ pas_m \end{cases} \text{ selon les cas précédemment mis en avant} \end{cases}$



•

$$P_{Ca} = \begin{bmatrix} p_{Ca1,1} & 0 & 0 & \dots & 0 \\ 0 & p_{Ca2,2} & & & \vdots \\ \vdots & 0 & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \vdots & \vdots & & & 0 \\ 0 & 0 & \dots & 0 & p_{CaR,R} \end{bmatrix} \in \mathcal{M}_R(\mathbb{R})$$

$$\text{où } p_{Ca_{i,i}} = \begin{cases} -pas_{Ca} & \text{si } te_{0,i} \leq t+1 < te_{1,i} \text{ ou } te_{3,i} \leq t+1 < te_{4,i} \\ 0 & \text{si } t \geq te_{4,i} \\ pas_{Ca} & \text{sinon} \end{cases}$$

•

$$\Delta_t = \begin{bmatrix} \delta_1(t) & 0 & 0 & \dots & 0 \\ 0 & \delta_2(t) & & & \vdots \\ \vdots & 0 & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \vdots & \vdots & & & 0 \\ 0 & 0 & \dots & 0 & \delta_3(t) \end{bmatrix} \in \mathcal{M}_R(\mathbb{R})$$

- $\alpha_i$  le coefficient d'indice (i,i) de la matrice  $(\Delta + P + pas_{Ca} * S_{Ca})$
- $D_i \in \mathcal{M}_R(\mathbb{R})$  la matrice dont le seul coefficient non nul est d'indice (i,i)  $d_{i,i} = 1$

On obtient alors l'équation suivante :

$$\boxed{V_i(t+1) = f(L_{C,i} * [V_t * [\mathbb{I}_R + (\Delta_t + P + P_{Ca} - \alpha_i * D_i)]) * (T_t + (1 - t_{i,1}) * N_i))} \quad (6)$$

#### 0.11.4 Fonctionnement algorithmique

- (1) : Initialiser la matrice C de manière aléatoire telle que les éléments de la diagonale soient égaux à 0
- (2) : Initialiser la matrice  $V_0$
- (3) : Initialiser à 0 la matrice  $T_0$
- (4) : Initialiser à 0 la matrice  $P_{Ca}$
- (5) : Initialiser à 0 la matrice  $P$
- (6) : Initialiser à 0 la matrice  $\Delta_0$
- (7) : Calculer la sortie des R neurones du réseau.  
On pose  $\{j_1, j_2, \dots, j_n\}$  les neurones dont la sortie est supérieure au seuil
- (8) : Mettre à jour la valeur du potentiel des R neurones du réseau
- (9) : Remplacer pour tout  $i \in \llbracket 1, R \rrbracket$   $t_{i,1}$  par  $\begin{cases} 1 & \text{si } i \in \{j_1, j_2, \dots, j_n\} \\ 0 & \text{sinon} \end{cases}$
- (10) : Remplacer pour tout  $i \in \llbracket 1, R \rrbracket$   $c_{i,i}$  par  $\begin{cases} \text{la valeur de } f_d^i & \text{selon la formule 6 si } i \in \llbracket 1, R \rrbracket \setminus \{j_1, j_2, \dots, j_n\} \\ f_d^i(0) & \text{sinon} \end{cases}$
- (11) : Pour tout  $i \in \llbracket 1, R \rrbracket$ , remplacer  $\delta_i(t)$  par  $\delta_i(t) + p_{i,i} + p_{Ca_{i,i}}$

- (12) : Pour tout  $i \in \{j_1, j_2, \dots, j_n\}$  :
  - si  $(1 + \delta_i(t)) * V_i(t) \geq V_{pa, max}$ , remplacer  $\begin{cases} p_{i,i} \text{ par } pas_m \\ p_{Ca_{i,i}} \text{ par } + pas_{Ca} \end{cases}$
  - sinon si  $\delta_i(t) = 0$ , remplacer  $\begin{cases} p_{i,i} \text{ par } pas_p \\ p_{Ca_{i,i}} \text{ par } - pas_{Ca} \end{cases}$
- (13) : Pour tout  $i \in \llbracket 1, R \rrbracket \setminus \{j_1, j_2, \dots, j_n\}$  :
  - Si  $|\delta_i(t) - 0| < pas_{Ca}$ , remplacer  $\begin{cases} p_{Ca_{i,i}} \text{ par } 0 \\ \delta_i(t) \text{ par } 0 \end{cases}$
  - sinon si  $\delta_i(t) > 0$ , remplacer  $pas_{Ca_{i,i}}$  par  $-pas_{Ca}$
  - sinon remplacer  $pas_{Ca_{i,i}}$  par  $+pas_{Ca}$
  - Remplacer  $\delta_i(t)$  par  $\delta_i(t) - p_{i,i}$
- (14) : Si  $\delta_i(t) + p_{i,i} + p_{Ca_{i,i}} \leq -1$  remplacer  $\delta_i(t)$  par  $-1 - p_{i,i} - p_{Ca_{i,i}}$
- (15) : Recommencer à l'étape (7)

**Remarques sur le programme :**

- Si on choisit un  $pas_{Ca}$  de manière adaptée, on arrivera à un instant donnée à  $\delta_i(t) = 0$ . Mais dans un cas général, on vérifie plutôt que  $\delta_i(t)$  est dans un voisinage de 0. Ce qui justifie la condition  $|\delta_i(t) - 0| < pas_{Ca}$  au lieu de  $\delta_i(t) = 0$ .
- L'ordre des conditions est primordial

En s'appuyant sur la figure 15, nous expliquons le principe de fonctionnement de l'algorithme.

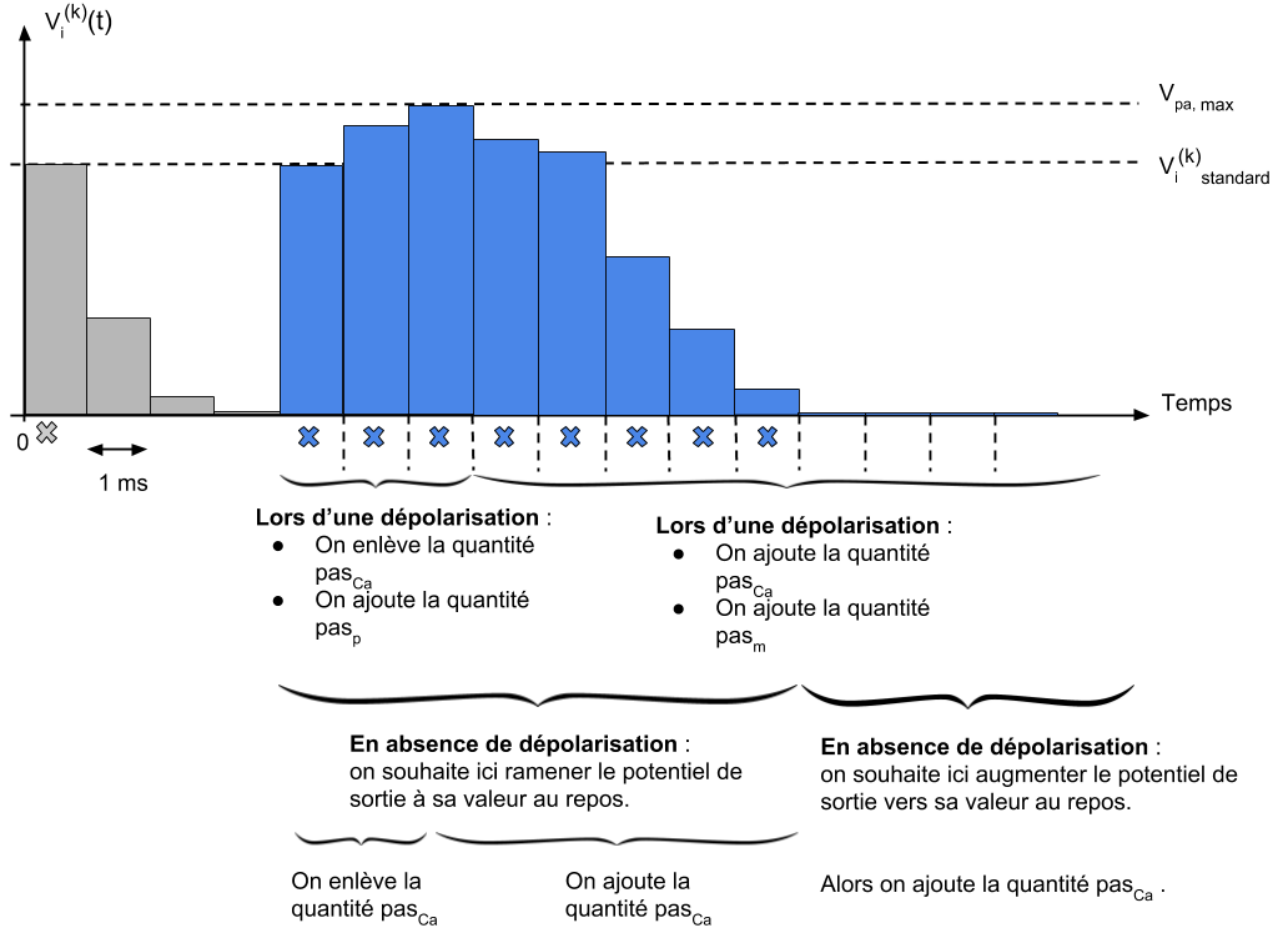


Figure 15: Justification des conditions retenues pour l'algorithme

On rentre dans l'algorithme avec  $\delta_i(0) = 0$ .

- si le neurone  $i$  considéré se dépolarise alors on vérifie la condition  $\delta_i(t) = 0$ . Dans ce cas, la prochaine fois  $\delta_i(t+1) = \delta_i(t) + pas_p - pas_{Ca}$ . Ce qui est correct car le potentiel augmentera donc lors de la prochaine dépolarisation.
- si le neurone ne se dépolarise pas, alors,  $\delta_i(t+1) = (0 - p_{i,i}) + p_{i,i} + 0$ .

Si les dépolarisations successives se continuent, on ne vérifie plus aucune condition jusqu'au moment  $t_1$  où  $(1 + \delta_i(t)) * V_i(t) \geq V_{pa, max}$ . Mais avant ce moment,  $\delta_i$  ne fait qu'augmenter car la valeur de  $p_{i,i}$  est  $pas_p$ .

Après  $t_1$ ,  $\delta_i(t)$  ne fait que décroître car la valeur de  $p_{i,i}$  est  $pas_m$ .

Si les dépolarisations s'arrêtent, comme  $pas_{Ca,i} = +pas_{Ca}$ , le potentiel remonte lentement vers sa valeur au repos (le moment où  $\delta_i(t) = 0$  alors  $(1 + \delta_i(t)) * V_i(t) = V_i(t)$ ).

Puis lorsque  $\delta_i(t)$  est dans un voisinage de 0, on réinitialise la valeur à 0 et on stoppe l'ajout de la quantité  $pas_{Ca}$ , car l'activité du neurone est redevenue normale.

Si après un faible nombre de dépolarisations (tel que  $(1 + \delta_i(t)) * V_i(t) > V_i(t)$ , c'est-à-dire tel que  $\delta_i(t) > 0$ ), le neurone ne se dépolarise plus, on fait revenir le potentiel à sa valeur normale en enlevant  $pas_{Ca}$  à  $\delta_i(t)$ .

Puis lorsque  $\delta_i(t)$  se rapproche de 0, on revient au cas précédent.

Si le neurone subit un grand nombre de dépolarisations au bout d'un temps, on obtient  $(1 - \delta_i(t))V_i(t) < V_i(t)$ , c'est-à-dire  $\delta_i(t) < 0$ . Dans ce cas, on ajoute la quantité  $pas_{Ca}$  à  $\delta_i(t)$  pour revenir à un potentiel normal.

De plus, lorsque le neurone ne dépolarise pas, on doit retirer  $p_{i,i}$  de  $\delta_i(t)$  car lors de la prochaine étape on a  $\delta_i(t+1) = \delta_i(t) + p_{i,i}$ ; or comme le neurone ne s'est dépolarisé, il ne perd aucun neurorécepteur.

Le but de la modélisation est donc de revenir à  $\delta_i(t) = 0$  et ce afin d'avoir  $(1 + \delta_i(t)) * V_i(t) = V_i(t)$ , ce qui correspond au potentiel de sortie normal.

### 0.11.5 Exemple de fonctionnement de la simulation

Reprenons le réseau de la figure 7 que nous redonnons ci-dessous :

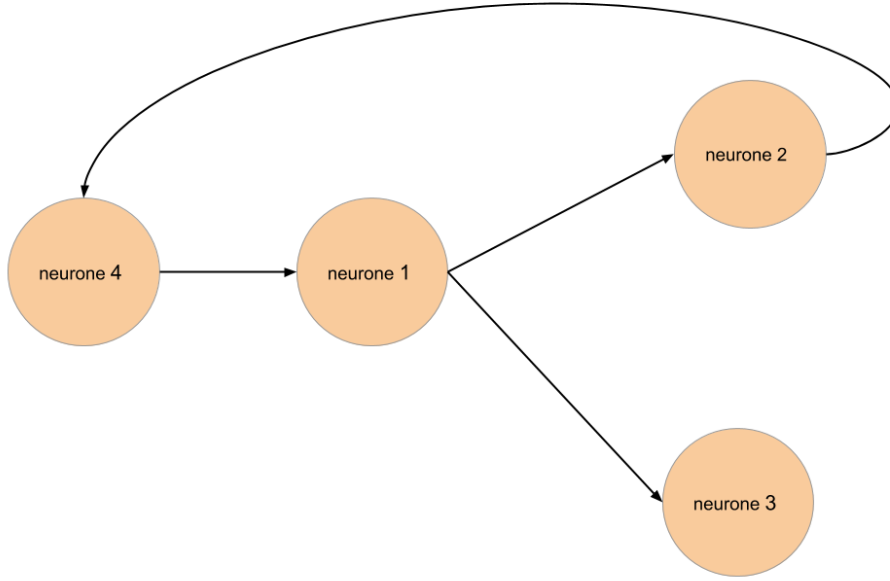


Figure 16:

Supposons qu'au temps  $t=0$  ms les neurone 1 ait déchargé et que  $V_1(0) < V_{max}$ . Alors :

$$V_0 = \begin{bmatrix} V_1(0) & 0 & 0 & 0 \\ 0 & V_2(0) & 0 & 0 \\ 0 & 0 & V_3(0) & 0 \\ 0 & 0 & 0 & V_4(0) \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, T_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$P = \begin{bmatrix} pas_p & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, P_{Ca} = \begin{bmatrix} -pas_{Ca} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \Delta_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Commençons par calculer la sortie du neurone 1. Pour ce faire, nous calculons la valeur de  $\delta_1(t=1)$ .

$$\Delta_0 + P + P_{Ca} = \begin{bmatrix} pas_p - pas_{Ca} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Ainsi,  $\delta_1(t=1) = pas_p - pas_{Ca}$ .

Rappelons que nous voulons que le potentiel de sortie du neurone 1 augmente peu à peu jusqu'à  $V_{max}$ .

Comme le neurone 1 vient de se dépolariser pour la première fois et que  $V_1(0) < V_{max}$ , alors son potentiel de sortie augmente. Il y a bien que son potentiel de sortie qui augmente (celui perçu par les autres neurones). Le potentiel donné par ses entrées (le potentiel du neurone de l'étape précédente) n'augmente pas quant à lui; il continue à suivre une règle de décroissance (celle donnée par la fonction  $f_d^i$ ). C'est pourquoi nous avons besoin de la matrice  $D_i$ .

En fait, sans la matrice  $D_i$  on obtiendrait :

$$T_0 + (1 - t_{1,1}) * N_i = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + (1 - 1) * \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Puis,

$$V_1(1) = [1 \ 0 \ 0 \ 1] * \left( \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} \delta_1(t=1) & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right) * \begin{bmatrix} V_1(0) & 0 & 0 & 0 \\ 0 & V_2(0) & 0 & 0 \\ 0 & 0 & V_3(0) & 0 \\ 0 & 0 & 0 & V_4(0) \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Autrement dit :

$$V_1(1) = [1 \ 0 \ 0 \ 1] * \begin{bmatrix} 1 + \delta_1(t=1) & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} V_1(0) & 0 & 0 & 0 \\ 0 & V_2(0) & 0 & 0 \\ 0 & 0 & V_3(0) & 0 \\ 0 & 0 & 0 & V_4(0) \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$V_1(1) = [1 \ 0 \ 0 \ 1] * \begin{bmatrix} (1 + \delta_1(t=1)) * V_1(0) & 0 & 0 & 0 \\ 0 & V_2(0) & 0 & 0 \\ 0 & 0 & V_3(0) & 0 \\ 0 & 0 & 0 & V_4(0) \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

C'est-à-dire

$$V_1(1) = (1 + \delta_1(t=1)) * V_1(0)$$

Ce qui n'est pas correcte car on devrait avoir  $V_1(1) = 1 * V_1(0)$ . Par contre, pour d'autres neurones on doit bien avoir  $V_{j \neq i}(1) = x * V_j(0) + \dots + (1 + \delta_1(t=1)) * V_1(0)$  (où x est le coefficient de décroissance actuel :  $x = c_{j,j}$ ).

C'est pour ce que nous devons introduire la matrice  $-\alpha_i * Di$ . Cette matrice, comme le fait la matrice  $N_i$ , d'obtenir temporairement 1 au lieu de  $1 + \delta_1(t=1)$  devant  $V_1(0)$  dans le calcul de  $V_1(1)$ . En effet :

$$\mathbb{I}_4 + (\Delta_0 + P + P_{Ca} - \alpha_1 * Di) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \left( \begin{bmatrix} pas_p - pas_{Ca} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} - (pas_p - pas_{Ca}) * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \right)$$

$$\mathbb{I}_4 + (\Delta_0 + P + P_{Ca} - \alpha_1 * Di) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Puis,

$$V_1(1) = [1 \ 0 \ 0 \ 1] * \begin{bmatrix} 1 * V_1(0) & 0 & 0 & 0 \\ 0 & V_2(0) & 0 & 0 \\ 0 & 0 & V_3(0) & 0 \\ 0 & 0 & 0 & V_4(0) \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 1 * V_1(0)$$

On fait alors de même pour les autres neurones du réseau :

$$V_2(1) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} V_1(0) & 0 & 0 & 0 \\ 0 & V_2(0) & 0 & 0 \\ 0 & 0 & V_3(0) & 0 \\ 0 & 0 & 0 & V_4(0) \end{bmatrix} * \begin{bmatrix} 1 + pas_p - pas_{Ca} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$V_2(1) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} (1 + pas_p - pas_{Ca}) * V_1(0) & 0 & 0 & 0 \\ 0 & V_2(0) & 0 & 0 \\ 0 & 0 & V_3(0) & 0 \\ 0 & 0 & 0 & V_4(0) \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$V_2(1) = 0 * V_2(0) + (1 + pas_p - pas_{Ca}) * V_1(0)$$

De même :

$$V_3(1) = 0 * V_2(0) + (1 + pas_p - pas_{Ca}) * V_1(0)$$

$$V_4(1) = 0 * V_4(0) + 0 * (1 + pas_p - pas_{Ca}) * V_1(0)$$

## 0.12 Modélisation des effets d'une substance psychoactive

Rappelons que les effets d'une substance psychoactive diffèrent selon son type que l'on peut regrouper au nombre de trois :

- Les stimulants (comme la caféine, la cocaïne, la nicotine) qui ont pour effets de favoriser une hyperactivité du cerveau. En d'autres termes, ces molécules se fixent aux récepteurs postsynaptiques et simulent la présence de neurorécepteurs.
- Les dépresseurs (comme l'alcool, la morphine, l'opium) abaissant l'activité globale du cerveau. Ces molécules se fixent également aux récepteurs postsynaptiques mais cette fois pour stopper la réception de neurorécepteurs.
- Les perturbateurs qui vont à la fois simuler l'arrivée de neurorécepteurs pour certains récepteurs et stopper la réception pour d'autres.

Pour modéliser ces effets, nous devons séparer la matrice C en somme de matrices, chacune associée à un type d'effet. On pose alors :

•

$$C_s = \begin{bmatrix} 0 & ws_{2,1} & ws_{3,1} & \dots & ws_{R,1} \\ ws_{1,2} & 0 & & & \vdots \\ \vdots & ws_{2,3} & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \vdots & \vdots & & & ws_{R,R-1} \\ ws_{1,R} & ws_{2,R} & \dots & ws_{R-1,R} & 0 \end{bmatrix} \in \mathcal{M}_R(\mathbb{R})$$

$$\text{où } \forall (i, j) \in \llbracket 1, R \rrbracket^2, ws_{i,j} = \begin{cases} 0 & \text{si } i = j \\ c_{i,j} & \text{si l'entrée correspondante du neurone } j \text{ est affectée par une stimulant} \\ 0 & \text{sinon} \end{cases}$$

•

$$C_d = \begin{bmatrix} 0 & wd_{2,1} & wd_{3,1} & \dots & wd_{R,1} \\ wd_{1,2} & 0 & & & \vdots \\ \vdots & wd_{2,3} & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \vdots & \vdots & & & wd_{R,R-1} \\ wd_{1,R} & wd_{2,R} & \dots & wd_{R-1,R} & 0 \end{bmatrix} \in \mathcal{M}_R(\mathbb{R})$$

où  $\forall (i, j) \in \llbracket 1, R \rrbracket^2, wd_{i,j} = \begin{cases} 0 & \text{si } i = j \\ c_{i,j} & \text{si l'entrée correspondante du neurone } j \text{ est affectée par un dépresseur} \\ 0 & \text{sinon} \end{cases}$

•

$$C_n = \begin{bmatrix} c_{1,1} & wn_{2,1} & wn_{3,1} & \dots & wn_{R,1} \\ wn_{1,2} & c_{2,2} & & & \vdots \\ \vdots & wn_{2,3} & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \vdots & \vdots & & & wn_{R,R-1} \\ wn_{1,R} & wn_{2,R} & \dots & wn_{R-1,R} & c_{R,R} \end{bmatrix} \in \mathcal{M}_R(\mathbb{R})$$

où  $\forall (i, j) \in \llbracket 1, R \rrbracket^2, wn_{i,j} = \begin{cases} c_{i,i} & \text{si } i = j \\ 0 & \text{si } wd_{i,j} = 0 \text{ ou } ws_{i,j} = 0 \\ c_{i,j} & \text{sinon} \end{cases}$

On obtient ensuite :

$$\boxed{C = C_n + C_s + C_d} \quad (7)$$

### 0.12.1 Exemple :

Supposons que l'on dispose de 4 neurones numéroté de 1 à 4 comme représenté sur la figure 17.

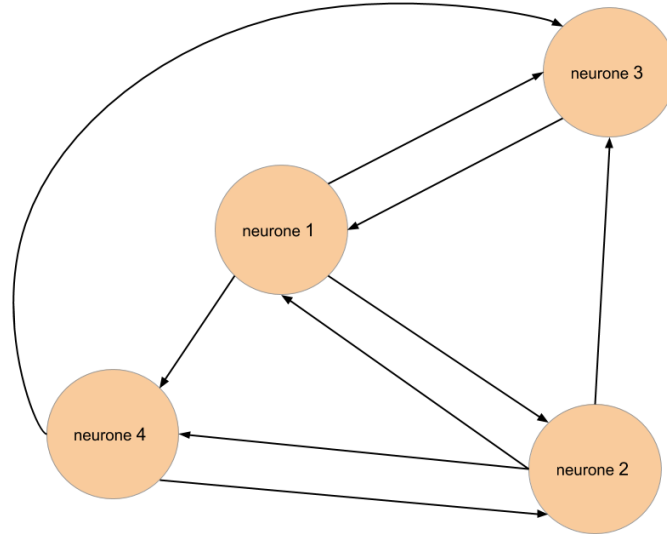


Figure 17: Exemple de réseau neuronal

Et on a par exemple :

$$C = \begin{bmatrix} 1 & 0.5 & 1 & 0 \\ 0.1 & 1 & 0 & 0.3 \\ 0.3 & 0.12 & 1 & 0.7 \\ 0.6 & 0.38 & 0 & 1 \end{bmatrix}$$

Pour la suite, on numérote les entrées de chaque neurone et on leurs associe un effet dû aux substances psychoactives.

Neurone	Entrée du neurone	Effet des substances psychoactives
1	(1.1) : Neurone 2	stimulation
	(1.2) : Neurone 3	aucun
2	(2.1) : Neurone 1	inhibition
	(2.2) : Neurone 4	stimulation
3	(3.1) : Neurone 1	aucun
	(3.2) : Neurone 4	aucun
	(3.3) : Neurone 2	inhibition
4	(4.1) : Neurone 1	stimulation
	(4.2) : Neurone 2	aucun

Commençons par remplir les matrices  $C_s$ ,  $C_n$  et  $C_r$  en suivant les entrées du neurone 1. Comme l'entrée (1.1) du neurone 1 est censé être affecté par un stimulant, on a uniquement  $c_{s2,1} = c_{2,1}$ , le reste des entrées du neurone 1 n'étant pas affectées. D'où :

$$C_s = \begin{bmatrix} 0 & 0.5 & 0 & 0 \\ ? & 0 & ? & ? \\ ? & ? & 0 & ? \\ ? & ? & ? & 0 \end{bmatrix}$$

Puis comme la sortie (1.2) du neurone 1 n'est pas affectée :

$$C_d = \begin{bmatrix} 0 & 0 & 0 & 0 \\ ? & 0 & ? & ? \\ ? & ? & 0 & ? \\ ? & ? & ? & 0 \end{bmatrix}$$

Et,

$$C_n = \begin{bmatrix} c_{1,1} & 0 & c_{3,1} & c_{4,1} \\ ? & c_{2,2} & ? & ? \\ ? & ? & c_{3,3} & ? \\ ? & ? & ? & c_{4,4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ ? & 1 & ? & ? \\ ? & ? & 1 & ? \\ ? & ? & ? & 1 \end{bmatrix}$$

On fait de même pour tous les autres neurones. On obtient alors le résultat suivant :

$$C_s = \begin{bmatrix} 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0 \\ 0.6 & 0 & 0 & 0 \end{bmatrix}$$

$$C_d = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0 \\ 0 & 0.12 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C_n = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0.3 & 0 & 1 & 0.7 \\ 0 & 0.38 & 0 & 1 \end{bmatrix}$$

Alors, on obtient bien :

$$C = \begin{bmatrix} 1 & 0.5 & 1 & 0 \\ 0.1 & 1 & 0 & 0.3 \\ 0.3 & 0.12 & 1 & 0.7 \\ 0.6 & 0.38 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0.3 & 0 & 1 & 0.7 \\ 0 & 0.38 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0 \\ 0.6 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0 \\ 0 & 0.12 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Décomposer la matrice C de cette manière permet de rendre compte de l'influence de substances psychoactives dans le réseau neuronal sans les valeurs  $w_{i,j}$  de la matrice C originelle. Dès lors, il ne



reste plus qu'à remplacer  $C$  par  $C_n$  dans les calculs des potentiels précédent ( $C \leftarrow C_n$ ). En notant  $\mathcal{E}_t(C)$  une des calculs matriciel du potentiel de  $V_i(t)$  dans lequel la matrice  $C$  intervient on obtient :

$$V_i(t+1) = \mathcal{E}_t(C_n) + L_{C_s,i} * \begin{bmatrix} V_{max} \\ V_{max} \\ \vdots \\ V_{max} \end{bmatrix} + L_{C_d,i} * \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (8)$$

**Remarque** la prise en compte de  $C_d$  peut sembler inutile car le résultat obtenu avec  $L_{C_d,i}$  pour tout  $i \in \llbracket 1, R \rrbracket$  est nul (puisque multiplié par la matrice nulle après). Mais cette matrice permet de conserver une trace des coefficients qu'il y avait au départ dans  $C$ . En fait, une fois que la concentration des substances psychoactives diminue, on doit être capable de faire revenir à la normale les entrées des neurones. A la fin de la simulation, on doit avoir  $C_n + \mathcal{V}_R + \mathcal{V}_R = C$ .

### 0.12.2 Fonctionnement algorithmique

- Si une entrée du neurone  $j$  (correspondant à la sortie du neurone  $i$ ) est affectée par un stimulant :

$$\begin{cases} c_{s_{i,j}} \leftarrow c_{i,j} \\ c_{d_{i,j}} \leftarrow 0 \\ c_{i,j} \leftarrow 0 \end{cases}$$

- Si une entrée du neurone  $j$  (correspondant à la sortie du neurone  $i$ ) est affectée par un dépresseur :

$$\begin{cases} c_{d_{i,j}} \leftarrow c_{i,j} \\ c_{s_{i,j}} \leftarrow 0 \\ c_{i,j} \leftarrow 0 \end{cases}$$

- Si une entrée du neurone  $j$  (correspondant à la sortie du neurone  $i$ ) n'est plus affectée par un stimulant :

$$\begin{cases} c_{i,j} \leftarrow c_{s_{i,j}} \\ c_{s_{i,j}} \leftarrow 0 \end{cases}$$

- Si une entrée du neurone  $j$  (correspondant à la sortie du neurone  $i$ ) n'est plus affectée par un dépresseur :

$$\begin{cases} c_{i,j} \leftarrow c_{d_{i,j}} \\ c_{d_{i,j}} \leftarrow 0 \end{cases}$$

# Partie IV

## Résultats exploitables

## 0.13 Tracé de l'activité électrique neurone par neurone

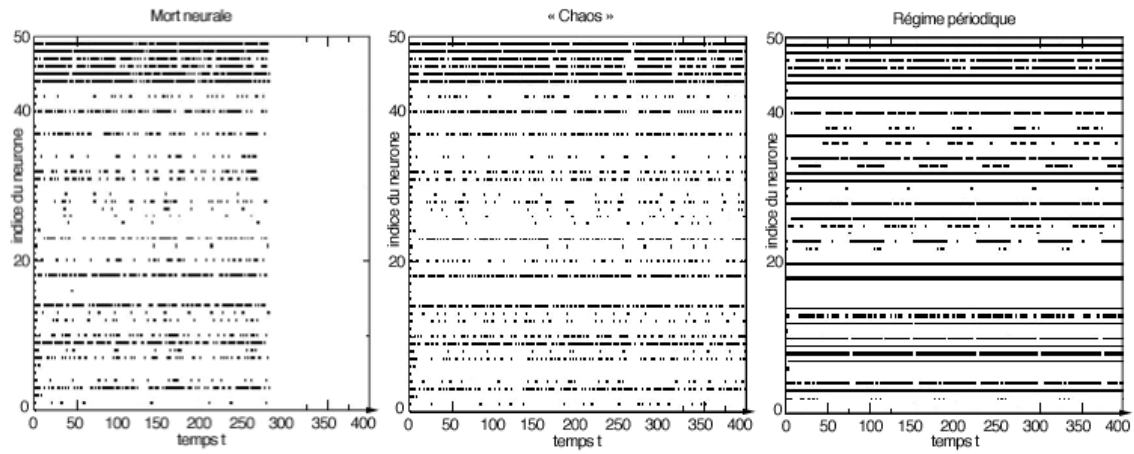


Figure 18: voir référence [4] pour ces figures

Comme le montre la figure 18, chaque ligne horizontale représente l'activité d'un neurone en fonction du temps (les points noirs correspondant au moment de la dépolarisation des neurones). Ces figures sont très semblables à celles d'encéphalogrammes, desquelles on peut observer des modifications de l'activité cérébrale lors de l'apparition de perturbations (substances psychoactives).

## 0.14 Tracé de l'activité du réseau

On se propose ici pour tout instant  $t$  de calculer le potentiel de tous les neurones en activités. On obtient alors une courbe de l'évolution du potentiel du réseau en fonction du temps; courbes similaires à celles obtenues lors d'une IRM permettant d'observer l'activité cérébrale, bien que l'IRM ne représente pas directement l'activité électrique des neurones.

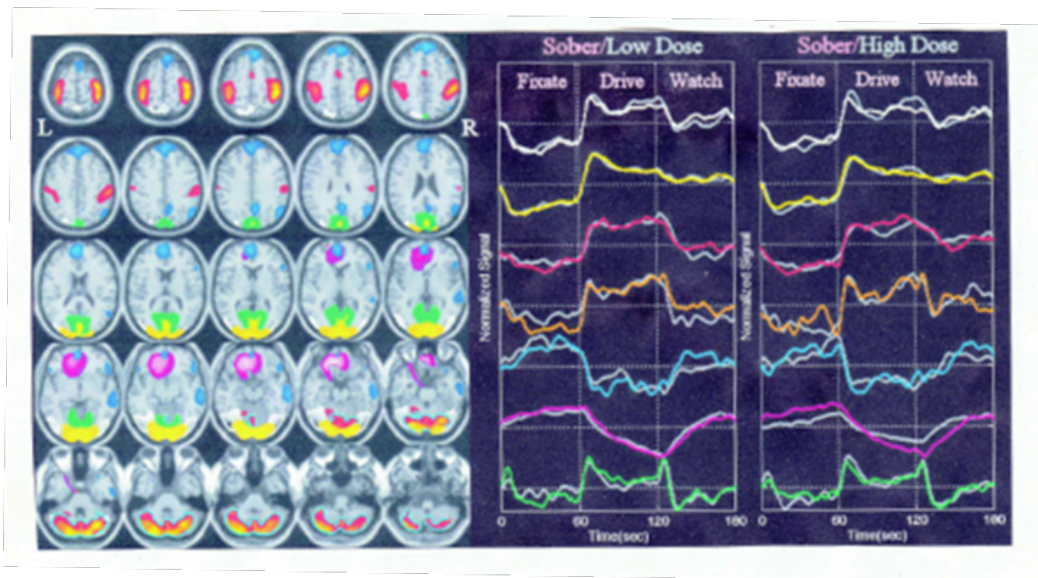


Figure 19: IRM représentant l'activité cérébrale

## Partie V

### Utilisation de la matrice $C$ , matrice des relations du réseau neuronal

## 0.15 Initialisation de la matrice C

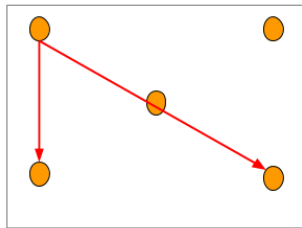
Nous étudierons nos modélisations pour 3 types d'initialisations :

- une initialisation aléatoire des liens entre neurones
- une initialisation tenant compte de la proximité spatiale des neurones. Dans ce cas, la matrice C joue le rôle de carte neuronale sur laquelle un neurone ne peut se lier qu'avec ses voisins (bien que dans la réalité deux neurones très espacés peuvent être en contact)
- une initialisation tenant compte de la spatialisation d'un réseau neuronal réelle (en l'occurrence celui du *Caenorhabditis elegans* [6])

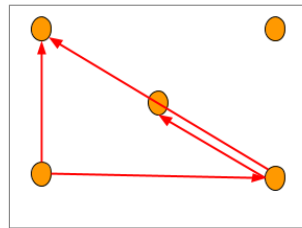
## 0.16 Représentation graphique des modélisations

La matrice C comme définit précédemment permet de rendre compte d'une organisation spatiale si on la considère comme une carte du réseau neuronal. Dès lors, en partant de la stimulation d'un neurone, on peut observer graphiquement l'évolution du réseau :

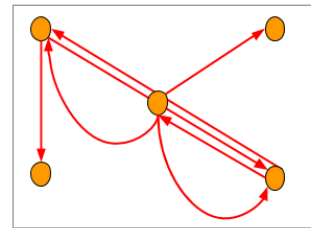
$$C = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$



Dépolarisation du neurone 1 au temps  $t_0$



temps  $t_0+1$



temps  $t_0+2$

# Références

- [1] 3ème partie : La communication intercellulaire le système nerveux. [http://mas.stephanie.free.fr/cours\\_systeme\\_nerveux.htm](http://mas.stephanie.free.fr/cours_systeme_nerveux.htm). [En ligne, accès 02-2018].
- [2] Neurosciences/la plasticité synaptique. [https://fr.wikibooks.org/wiki/Neurosciences/La\\_plasticite%CC%89\\_synaptique](https://fr.wikibooks.org/wiki/Neurosciences/La_plasticite%CC%89_synaptique), 12 2017. [En ligne, accès 02-2018].
- [3] B. M. Abdelhak. Les réseaux de neurones à réservoir en traitement d'images. <http://slideplayer.fr/slide/3678912/>, 12 2004. [En ligne, accès 02-2018].
- [4] B. Cessac. Le cerveau est-il un bon modèle de réseau de neurones ? [https://interstices.info/jcms/c\\_31668/le-cerveau-est-il-un-bon-modele-de-reseau-de-neurones](https://interstices.info/jcms/c_31668/le-cerveau-est-il-un-bon-modele-de-reseau-de-neurones), 11 2007. [En ligne, accès 02-2018].
- [5] M. P. A. C. d'Azur. Les neurones et synapses. [http://www.museum-marseille.org/marseille\\_cerveau\\_synapse.htm](http://www.museum-marseille.org/marseille_cerveau_synapse.htm), 12 2004. [En ligne, accès 02-2018].
- [6] J.Leskovec. C. elegans frontal neural network. <https://snap.stanford.edu/data/C-elegans-frontal.html>. [En ligne, accès 02-2018].
- [7] P. Sauleau. Fonctionnement cellulaire du système nerveux. <https://sites.google.com/site/aphysionado/home/cellsnv>. [En ligne, accès 02-2018].
- [8] science.ch. Réseaux de neurones formels. <http://informatique.coursgratuits.net/methodes-numeriques/reseaux-de-neurones-formels.php>. [En ligne, accès 02-2018].