



RAPPORT DE STAGE
L3 MATHÉMATIQUES

Introduction aux mathématiques des réseaux de neurones impulsionnels

05 JANVIER 2026 — 30 JANVIER 2026

Sujet traité par :

M. LOU JANS

Sous l'encadrement du :

DR. JAMES LARROUY

UNIVERSITÉ DES ANTILLES - PÔLE GUADELOUPE
U.F.R. SCIENCES EXACTES ET NATURELLES
LABORATOIRE DE MATHÉMATIQUES, INFORMATIQUE ET APPLICATION (LAMIA)

Table des matières

1	Introduction	2
2	Outils mathématiques pour réseaux neuronaux impulsionnels	4
2.1	Nomenclature liée aux signaux neuronaux	4
2.2	Fréquence de spike à un instant t	4
2.3	L'exemple du processus de Poisson homogène pour l'étude des train de spike	6
3	La Déformation Temporelle Dynamique	7
3.1	Comparaison de séries temporelles	7
3.2	Programmation dynamique et complexité	9
4	Conclusion	11

1 Introduction

On appelle **réseau neuronal** un type d'architecture informatique qui reproduit le fonctionnement du modèle du cerveau humain. En d'autres termes, il s'agit d'un système informatique qui apprend à effectuer des tâches en analysant des exemples. Les réseaux neuronaux sont constitués d'un ensemble d'unités de traitement interconnectées appelés **nœuds**, organisés en couches, qui se transmettent des données au moyen d'impulsions électriques pour traiter l'information et effectuer des prédictions. Chaque nœud exécute une fonction mathématique sur l'entrée qu'il reçoit et transmet la sortie au nœud suivant s'il dépasse un seuil. Les nœuds peuvent être organisés en plusieurs couches cachées pour plus de complexité.

Le concept de réseaux de neurones a vu le jour en 1943 avec les travaux de McCulloch et Pitts [12] dans lequel les auteurs ont proposé une modélisation mathématique du fonctionnement du cerveau à l'aide de circuits électriques. Entre les années 1950, avec les progrès réalisés dans le domaine de l'informatique, la simulation de réseaux neuronaux était possible. Ainsi, des modèles de réseaux neuronaux simples comme *ADALINE* (développée pour la reconnaissance de motifs binaires de sorte que si le modèle lisait des flux de données provenant d'une ligne téléphonique, il pourrait prédire le bit suivant) ou encore *MADALINE* (le premier réseau neuronal appliqué à des situations du monde réel qui éliminait les échos sur les lignes téléphoniques par le biais de filtres adaptatifs,) furent créés. En 1982, les travaux de Hopfield [6] change d'approche. Son approche consistait à créer des machines plus performantes grâce à l'utilisation de lignes bidirectionnelles tandis qu'auparavant, les connexions entre les neurones étaient unidirectionnelles. Les travaux de Hopfield [6] ont conduit à une utilisation généralisée des réseaux neuronaux dans la recherche sur *l'intelligence artificielle* (IA).

Les réseaux neuronaux se déclinent en différents types en fonction de leur structure et de leur fonction. Chaque type diffère dans la manière dont les données circulent et dont les couches interagissent, des modèles à propagation simple aux architectures complexes qui mettent en boucle, ignorent ou combinent les résultats. Durant ce stage, nous nous sommes focalisés sur le cas des **réseaux neuronaux impulsifs** (ou *Spiking Neural Networks*, en anglais). Ainsi, nous nous restreindrons aux réseaux neuronaux impulsifs dans la suite.

Les réseaux de neurones impulsifs (SNN) constituent une classe de modèles inspirés du fonctionnement des neurones biologiques : l'information n'y est pas portée par des valeurs continues mais par des *événements* (ou impulsions) datés. Cette représentation temporelle ouvre la voie à des méthodes de calcul sobres en énergie et à des traitements naturellement adaptés aux signaux discrets dans le temps (capteurs neuromorphiques, séquences d'événements, etc.). La distinction fondamentale entre les SNN et les autres modèles de réseaux neuronaux réside essentiellement dans le mécanisme de transmission de l'information des SNN. Dans un réseau de neurones formel classique (par exemple, un perceptron multicouche), la sortie d'un neurone est une valeur réelle continue (par exemple dans $[0, 1]$ ou dans \mathbb{R}), souvent interprétée comme une fréquence de décharge moyenne. À l'inverse, les SNN imitent plus fidèlement la neurophysiologie biologique en utilisant des impulsions (ou *spikes*, en anglais), c'est-à-dire des événements discrets dans le temps. Un neurone impulsif reste silencieux tant que son potentiel de membrane v n'atteint pas un seuil spécifique ϑ . Une fois ce seuil franchi, il émet une impulsion unique et instantanée, puis son potentiel est réinitialisé (voir, e.g., Gerstner et Kistler [4]). Ce fonctionnement induit deux propriétés majeures :

- **la parcimonie temporelle (ou *sparsity*, en anglais)** : les neurones ne sont actifs que lorsqu'un événement pertinent survient, ce qui réduit l'activité globale du réseau ;
- **le codage temporel** : l'information n'est plus seulement portée par le nombre d'impulsions (codage par taux), mais également par le moment précis de leur émission (latence) ou par le délai relatif entre plusieurs impulsions [11].

Remarque. L'architecture événementielle des SNN offre une réponse prometteuse aux limites énergétiques de l'IA traditionnelle. En n'effectuant des calculs que lors de l'arrivée d'une impulsion (plutôt qu'à chaque cycle d'horloge pour l'ensemble des neurones), les SNN permettent une efficacité énergétique théorique de plusieurs ordres de grandeur supérieure, particulièrement lorsqu'ils sont implémentés sur du matériel neuromorphique dédié (tels que les puces *Loihi* d'Intel ou *TrueNorth* d'IBM).

Un exemple de SNN : le modèle d'Izhikevich

Pour comprendre le fonctionnement du cerveau, il est nécessaire de combiner des études expérimentales sur les systèmes nerveux animaux et humains avec des simulations numériques de modèles cérébraux à grande échelle. Lors du développement de tels modèles composés de neurones à impulsions, il faut trouver des compromis entre deux exigences apparemment contradictoires : le modèle d'un neurone unique doit être

- (i) simple à calculer ;

(ii) capable de reproduire les schémas d'activation complexes observés chez les neurones biologiques réels. En 2003, Izhikevich [7] propose un modèle mathématique qui reproduit les impulsions et les salves de neurones corticaux de types connus. En se basant sur des méthodes de bifurcations (voir Izhikevich [8]), il parvient à réduire de nombreux modèles neuronaux biophysiquement précis de type Hodgkin-Huxley à un système bidimensionnel (en 2D) d'équations différentielles ordinaires de la forme

$$\begin{cases} \frac{d}{dt}v(t) = 0.04v^2(t) + 5v(t) + 140 - u(t) + I, \\ \frac{d}{dt}u(t) = a(bv(t) - u(t)), \end{cases}$$

avec la règle auxiliaire de remise à zéro après émission d'un spike (ou *auxiliary after-spike resetting*, en anglais) :

$$\text{si } v(t_0) \geq 30 \text{ mV} \quad \text{alors} \quad \begin{cases} v(t_0) \leftarrow c, \\ u(t_0) \leftarrow u(t_0) + d. \end{cases}$$

Grâce à ce modèle, il est possible de simuler en temps réel (résolution de 1 ms) des dizaines de milliers de neurones corticaux à impulsion, sur un ordinateur de bureau.

À noter qu'ici, $t > 0$ représente le temps, v et u sont des variables sans dimensions et a, b, c, d sont des paramètres sans dimensions. Par ailleurs, la variable v représente le potentiel de membrane, tandis que u est une variable de récupération (activation de courants potassiques et inactivation de courants sodiques) fournissant une rétroaction négative sur v . Le terme quadratique en v capture une dynamique d'initiation du spike de type « *quadratic integrate-and-fire* », tandis que le mécanisme de *reset* modélise l'évènement impulsionnel sans résoudre explicitement la phase rapide du potentiel d'action. Enfin, les paramètres (a, b, c, d) contrôlent finement la dynamique : a fixe l'échelle de temps de u , b règle le couplage sous-seuil entre v et u , c impose la valeur de remise à zéro de v après spike, et d détermine l'incrément de u après spike (adaptation). En ajustant ces quatre paramètres, le modèle reproduit des classes standard de neurones (RS, IB, CH, FS, LTS, neurones thalamo-corticaux, résonateurs), tout en restant adapté aux simulations de réseaux à grande échelle (résolution milliseconde) grâce à sa simplicité numérique.

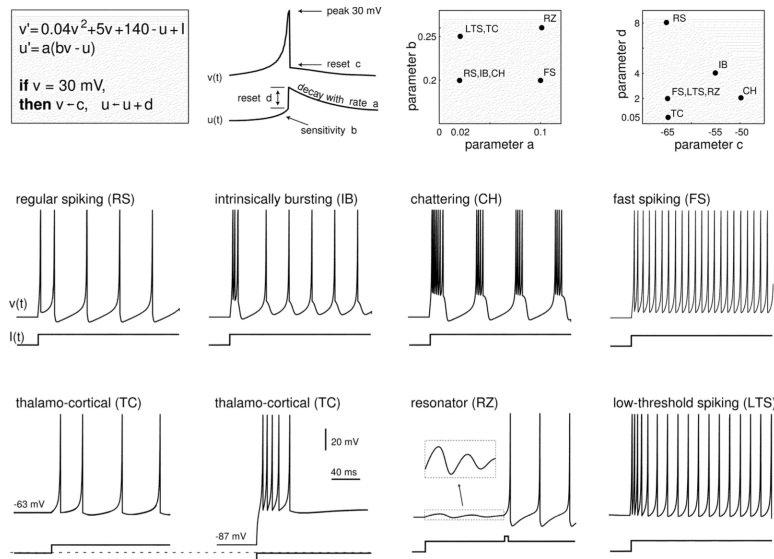


FIGURE 1 – Des types de neurones connus correspondent à différentes valeurs des paramètres a, b, c, d du modèle décrit par (1) et (2). RS, IB et CH sont des neurones excitateurs corticaux. FS et LTS sont des interneurons inhibiteurs corticaux. Chaque encart montre une réponse en tension du neurone modèle à un échelon de courant continu $I = 10$ (en bas). La résolution temporelle est de 0,1 ms. (source : [7])

L'objectif principal de ce stage était d'explorer la théorie des réseaux neuronaux impulsionnels du point de vue des mathématiques. Nous nous intéresserons également à un outil, la déformation temporelle dynamique (ou *Dynamic Time Warping*, en anglais), permettant de comparer deux séries temporelles en autorisant des déformations temporelles locales. Cet outils à récemment été utilisé par Doncescu *et al.* [10] pour étudier le potentiel d'un réseau neuronal à impulsions minimaliste dans les tâches de reconnaissance de chiffres, en utilisant l'ensemble de données MNIST comme référence.

2 Outils mathématiques pour réseaux neuronaux impulsionsnels

Dans cette partie du rapport, nous présentons un certain nombre d'outils mathématiques utilisés pour étudier et décrire les réseaux neuronaux impulsionsnels. Nous commençons avec un bref rappel sur la notions de signal pour les réseaux de neurones.

2.1 Nomenclature liée aux signaux neuronaux

Depuis les travaux de Santiago Ramon y Cajal en 1899, il est connu que la transmission du signal neuronal s'effectue des dendrites et du corps cellulaire à l'axone. Par conséquent, chaque neurone possède :

- (a) **une composante réceptrice**, le corps cellulaire et les dendrites ;
- (b) **une composante de transmission**, l'axone ;
- (c) **une composante effectrice**, les terminaisons axonales.

Ainsi, l'information est portée par des événements discrets dans le temps que l'on appelle **spike** (ou potentiel d'action), c'est-à-dire une brève impulsion électrique produite par un neurone. L'émission de spike dépend d'un certain nombre de facteurs dont les stimuli, les propriétés de la cellule, etc.

Remarque. La durée de chaque spike est d'environ 1 milli-secondes (ms). Elle est suivie d'une période durant laquelle le neurone n'est pas en mesure de produire un nouveau spike (environ 1 ms, également).

Considérons une fenêtre d'observation $]0; T]$, donné où $T > 0$ est la durée maximale d'observation. De la remarque précédente, on définit un processus ponctuel appelé *train de spike*. Typiquement, on appelle **train de spike** toute suite d'instantanés auxquels un neurone émet un spike. En d'autres termes, pour $]0; T]$, un train de spike est une suite

$$\{t_n\}_{n \in \mathcal{I}} \subset]0; T] \quad \text{telle que} \quad \mathcal{I} \subset \mathbb{N} \text{ fini et } 0 < t_0 < t_1 < t_2 < \dots < t_n \leq T.$$

Cette notion permet de représenter l'activité neuronale par un *raster plot* comme suit :

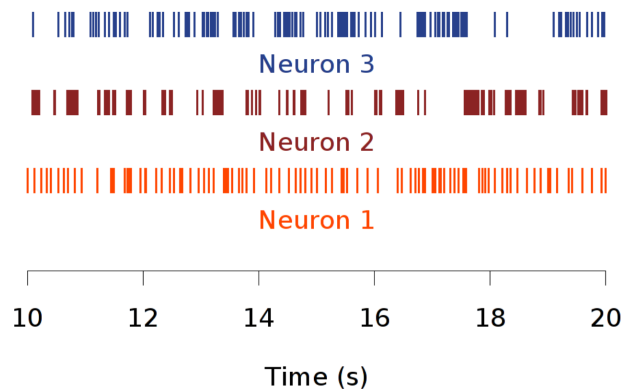


FIGURE 2 – Exemple de train de spike individuels pour 3 neurones différents.

En bref, le postulat clé des neurosciences est que la succession temporelle des émissions de spike (et pas leur formes) contient la majorité de l'information échangée entre plusieurs régions du cerveau. C'est ce postulat qui motive l'étude des trains de spike et la proposition de modèle pouvant prédire les émissions de spikes. En particulier, les modèles doivent permettre de passer de l'échelle microscopique (avec le comportement des synapses, notamment) à l'échelle macroscopique (comportement collectif liée à l'activité électrique d'une zone du cerveau, notamment). C'est en cela que les mathématiques sont importantes. À titre d'illustration, Galves et Löcherbach [3] ont proposé un modèle stochastique (c'est-à-dire qui contient une source d'aléa) qui permet de passer de l'échelle microscopique à macroscopique et inversement.

2.2 Fréquence de spike à un instant t

Comme nous l'avons dit à la sous-section précédente, le cerveau manipule l'information en combinant et en générant des spike. Il semble naturel de se demander comment l'information (concernant les variables sensorielles, les inférences sur le monde, les plans d'action, les états cognitifs, etc.) est représentée dans les trains de spike. Pour cela, il faut coupler les données expérimentales aux mathématiques.

Les données expérimentales proviennent principalement de contextes sensoriels :

- (a) on souhaite pouvoir répéter le même stimulus (*bien que cela ne garantisse pas que toute l'information représentée soit identique, certaines informations sont susceptibles d'être partagées entre les essais*) ;
- (b) on souhaite produire des méthodes computationnelles pour caractériser et quantifier ces résultats.

D'un autre côté, les mathématiques peuvent nous indiquer à quoi devraient ressembler les représentations des trains de spike. Elle permettent notamment de savoir quelles variables internes pourraient devoir être représentées en tenant compte :

- (a) de la catégorie des variables ;
- (b) de notions d'incertitude (ou d'aléas) ;
- (c) de notions d'erreur.

Au sein de cette sous-section, nous chercherons à comprendre comment compter le nombre de spike émis dans un train de spike donné.

On rappelle qu'une **variable aléatoire** est une valeur numérique associée à chaque résultat d'une expérience. Dans notre contexte, on peut penser au nombre de spike généré par un neurone sur une fenêtre de temps donnée. Ainsi, la probabilité d'obtenir chaque résultat possible est caractérisée par une **fonction de densité de probabilité**. L'intégrale d'une fonction de densité de probabilité est appelée **fonction de répartition**. Ces derniers caractérisent la probabilité d'obtenir un résultat inférieur ou égale à une valeur donnée. Enfin, un **processus aléatoire** est une règle permettant d'associer une fonction $x_t := x(t)$ à chaque résultat d'une expérience.

Exemple. Le tracé de spike enregistré à partir d'une stimulation par électrode intracellulaire sur une fenêtre de temps d'une seconde peut être considéré comme un processus aléatoire.

Considérons une fenêtre d'observation $]0; T]$ et un train de spike $\mathcal{S} = \{t_n\}_{n \in \mathcal{I}}$ où $\mathcal{I} \subset \mathbb{N}$ fini. On rappelle que la **fonction delta de Dirac** (ou *Dirac delta function*, en anglais) est la fonction $\delta : \mathbb{R} \rightarrow \{0, 1\}$ définie par

$$\delta(x) = \begin{cases} 1 & \text{si } x = 0, \\ 0 & \text{sinon,} \end{cases} \quad \text{telle que} \quad \int_{-\infty}^{+\infty} \delta(t) dt = 1.$$

Grâce à elle, on définit la **fonction de réponse neuronale** $\rho : \mathbb{R}_+ \rightarrow \mathbb{N}$ par

$$\rho(t) := \sum_{k=1}^{\#\mathcal{S}} \delta(t - t_k), \quad (1)$$

où $\#\mathcal{S}$ représente le cardinal du train de spike \mathcal{S} et les t_k sont les temps observés auxquels il y a eu un spike. Dans la littérature, la fonction de réponse neuronale ρ est considérée comme un processus aléatoire. Cette fonction est parfaitement équivalente à la liste des instants d'apparition de spike dans le train de spike sur la fenêtre d'observation. Elle s'avère également utile pour exprimer les sommes sur les potentiels d'action sous forme d'intégrales temporelles.

Exemple. Considérons deux instants $t_1, t_2 \in]0; T]$ tels que $t_1 < t_2$. Puisque $\rho(\cdot)$ est discrète, on peut écrire le nombre entier n de spike émis entre t_1 et t_2 via l'intégrale

$$\int_{t_1}^{t_2} \rho(t) dt = n.$$

De cette façon, la fréquence de spike à un instant $t \in]0; T]$ fixé peut-être définie comme l'espérance de $\rho(\cdot)$, moyenné sur un nombre infini de répétition. Toutefois, il est impossible de réaliser un nombre infini d'essais en pratique. On utilise donc un nombre fini d'essais M et on obtient une estimation de la fréquence de spike $e_M(\cdot)$ définie par :

$$e_M(t) := \frac{1}{M} \sum_{i=1}^M \rho_i(t), \quad (2)$$

où $\rho_i(t)$ est la fonction de réponse neuronale associée à chaque essais $i \in [[1; M]]$. Ainsi, le nombre moyen $\langle n \rangle$ de spike émis entre instants $t_1, t_2 \in]0; T]$ tels que $t_1 < t_2$ est approché suit :

$$\langle n \rangle := \int_{t_1}^{t_2} e_M(t) dt. \quad (3)$$

Supposons M très grand. On note sans perte de généralités $e_M(\cdot) := e(\cdot)$. Pour deux instants t_1, t_2 très proches, par exemples quand $t_1 = t - \frac{\delta t}{2}$ et $t_2 = t + \frac{\delta t}{2}$, nombre moyen $\langle n \rangle$ de spike émis entre instants $t_1, t_2 \in]0; T[$ peut-être approché par

$$\langle n \rangle \approx e(t)\delta t.$$

De cette manière, on peut réduire δt jusqu'à ce que la probabilité que plusieurs spike apparaissent dans cet intervalle soit suffisamment faible pour être négligée. Dans ce cas, le nombre moyen de spike est égale à la probabilité d'émission d'un seul spike. Autrement dit, on a

$$\mathbb{P}(\#S_{[t_1; t_2]} = 1) = e(t)\delta t, \quad (4)$$

où \mathbb{P} est une mesure de probabilité et $\#S_{[t_1; t_2]}$ désigne le nombre de spike dans la fenêtre $[t_1, t_2]$.

Remarque. Il existe d'autres méthodes pour moyenniser la fonction de réponse neuronales ρ définie en (1) sur un intervalle de temps. À titre d'exemple on peut citer la fonction de corrélation que nous ne décrirons pas dans ce rapport. Le principal avantage de cet action réside dans le fait que contrairement $\rho(\cdot)$, la fonction de fréquence moyenne $e(\cdot)$ définie en (2) (avec M très grand) donne une description simplifiée de la réponse neuronale.

Une question ouverte dans la littérature est de savoir si il est pertinent de se focaliser sur autre chose que la fréquence moyenne d'émission de spike pour étudier les trains de spike.

2.3 L'exemple du processus de Poisson homogène pour l'étude des train de spike

Dans cette sous-section on suppose que la fréquence $e(\cdot)$ est constante pour un train de spike \mathcal{S} donné. C'est ce que l'on appelle un **processus de Poisson homogène** (cf. [1, 2, 5, 9]). Nous considérons une grande fenêtre d'observation $]0; T[$ dans lequel on place aléatoirement un spike. On choisit ensuite un intervalle $]t_1; t_2[\subset]0; T[$ (avec $0 \leq t_1 < t_2 < T$). Notons Δt la taille de l'intervalle $]t_1; t_2[$. On a alors

$$\Delta t = t_2 - t_1.$$

Ainsi, la probabilité qu'un spike s soit émis dans l'intervalle de temps $]t_1; t_2[$ est :

$$\mathbb{P}(s \in]t_1; t_2[) = \frac{\Delta t}{T}. \quad (5)$$

Si on place $k \in [[1; \#S]]$ spike aléatoirement dans $]0; T[$, la probabilité que n d'entre eux se situent dans un intervalle de taille $\Delta t := t_1 - t_2$ est donnée par la formule du binôme de Newton :

$$\mathbb{P}(\#S_{\Delta t} = n) = \frac{k!}{(k-n)! \cdot n!} \left(\frac{\Delta t}{T} \right)^n \left(1 - \frac{\Delta t}{T} \right)^{k-n}, \quad (6)$$

où $\#S_{\Delta t}$ désigne le nombre d'élément d'un intervalle de longueur Δt . En fait, la formule du binôme de Newton est un outil que l'on utilise pour calculer la probabilité que n évènements se réalisent parmi k essais.

Prenons k et T très grand tout en gardant la quantité $r = \frac{k}{T} > 0$ constante. Notons que puisque k désigne le nombre de spike émis et T le temps d'observation maximal, r est la fréquence moyenne d'émission de spike, c'est-à-dire le nombre moyen de spike émis par seconde. On a :

$$T = \frac{k}{r}. \quad (7)$$

En combinant (6) et (7), on obtient :

$$\mathbb{P}(\#S_{\Delta t} = n) = \frac{k!}{(k-n)! \cdot n!} \left(\frac{r\Delta t}{k} \right)^n \left(1 - r \frac{\Delta t}{k} \right)^{k-n},$$

d'où :

$$\mathbb{P}(\#S_{\Delta t} = n) = \frac{(r\Delta t)^n}{n!} \cdot \frac{k(k-1)\dots(k-n+1)}{k^n} \left(1 - \frac{r\Delta t}{k} \right)^k \left(1 - \frac{r\Delta t}{k} \right)^{-n}. \quad (8)$$

En passant à la limite quand k tends vers $+\infty$ dans (8), on déduit que

$$\mathbb{P}(\#S_{\Delta t} = n) = \frac{(r\Delta t)^n}{n!} \exp(-r\Delta t). \quad (9)$$

En d'autres termes, on obtient la fonction de masse d'une loi de Poisson. Ainsi, pour une fréquence moyenne d'émission de spike constante $e(\cdot) = r$ donnée, la formule nous fournis la probabilité d'avoir n spike sur une intervalle de temps de longueur Δt .

Remarque. La formule donnée par (9) n'est vraie que lorsque les spikes sont complètement indépendants les uns des autres, c'est-à-dire qu'ils sont placés aléatoirement dans l'ensemble de l'intervalle de temps $]0, T[$.

Dans le cas du processus de Poisson homogène, la mesure de comptage de spike sur un intervalle de temps de longueur $\Delta t := t_2 - t_1$ est obtenue en ne considérant plus la dépendance en temps dans (3) (puisque $e(\cdot) := e_M(\cdot) = r$ est supposé constant). Plus précisément, le comptage est donné par

$$\langle n \rangle = \int_{t_1}^{t_2} r dt = r \Delta t,$$

pour tout intervalle de longueur $\Delta t = t_2 - t_1$. En faisant quelques calculs, on obtient que la variance de la mesure de comptage est donnée par :

$$\sigma_n^2 = r \Delta t.$$

Ainsi, le rapport suivant nous donne le **facteur de Fano**, noté F :

$$F = \frac{\sigma_n^2}{\langle n \rangle} = 1.$$

Le facteur de Fano caractérise la variabilité du comptage de spike. Le fait que ce quotient soit égal à 1 est une des caractéristiques d'un processus de Poisson.

Maintenant que l'on a étudié la façon dont les spikes apparaissent dans un train de spike, on peut se demander comment comparer deux trains de spike. Dans leurs travaux récents, Doncescu *et al.* [10] ont utilisé la **Déformation Temporelle Dynamique** (DTW) pour faire cela. Nous présenterons donc cet outil dans la prochaine section.

3 La Déformation Temporelle Dynamique

On appelle **série temporelle** toute suite d'observation d'une même quantité mesurée à différents instants.

Exemple. Les trains de spike sont des séries temporelles.

3.1 Comparaison de séries temporelles

Dans le cas où l'on est en présence de deux séries temporelles, une tâche courante consiste à comparer ces suites. Dans certains cas, cette comparaison peut être faite via une simple distance telle que la distance Euclidienne. Cependant, il arrive souvent que deux séquences aient des formes globales de composantes approximativement identiques, mais que ces formes ne soient pas alignées sur l'axe des abscisses (discrétisant le temps). La figure ci-dessous illustre ce phénomène par un exemple simple.

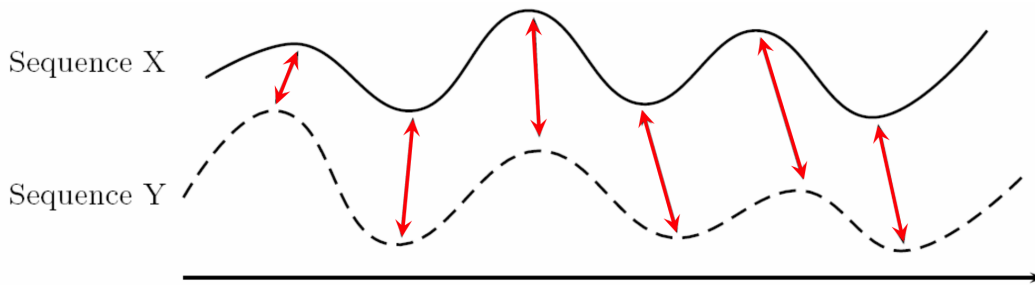


FIGURE 3 – Alignement temporel de deux séquences dépendantes du temps. Les points alignés sont indiqués par les flèches.

Afin de déterminer la similarité entre de telles séquences, ou en tant qu'étape de prétraitement avant de les moyenner, il est nécessaire de « déformer » l'axe temporel d'une (ou des deux) séquences pour obtenir un meilleur alignement. La **déformation temporelle dynamique** (ou *Dynamic Time Warping*, en anglais) est une technique permettant de réaliser efficacement cette déformation.

Soient $m, n \in \mathbb{N}^*$. Considérons deux séries temporelles X et Y de taille n et m respectives où :

$$X = \{x_k\}_{k \in [0;n]} \subset \mathbb{R}^n \quad \text{and} \quad Y = \{y_k\}_{k \in [0;m]} \subset \mathbb{R}^m. \quad (10)$$

Pour aligner les séries temporelles X et Y , à l'aide de la *DTW*, nous construisons une matrice $C := (c_{i,j})$ à n ligne et p colonnes appelée **matrice coût** telle que

$$c_{i,j} = \|x_i - y_j\|, \quad \text{pour tout } (i,j) \in [[1,n]] \times [[1,p]].$$

Ainsi, chaque élément de la matrice coût C correspond à l'alignement entre les points x_i et y_j .

Dans ce cas, il est possible de définir un (n,m) -**chemin d'alignement** (ou (n,m) -*Warping path*, en anglais) $\mathbf{W} := \{w_1, w_2, w_3, \dots, w_K\}$ où $K \in [[\max(n,m); m+n-1]]$. On a alors :

$$w_k = (u_k, v_k) \in [[1,n]] \times [[1,m]], \quad \text{pour tout } k \in [[1;K]].$$

Un chemin d'alignement est un ensemble d'éléments contigu, c'est-à-dire un ensemble d'éléments voisins, de la matrice C qui définit une fonction entre les séries temporelles X et Y et qui est soumis aux contraintes suivantes :

- (i) **Condition aux bords** : $w_1 = (1, 1)$ et $w_K = (n, m)$. Autrement dit, \mathbf{W} commence et se termine dans des cellules diagonalement opposées aux coins de la matrice.
- (ii) **Continuité** : Pour $w_k = (a, b)$ donné, on a $w_{k-1} = (u, v)$ où $a - u \leq 1$ et $b - v \leq 1$. Cela limite les étapes autorisées dans le chemin de déformation aux cellules adjacentes (y compris les cellules adjacentes en diagonale).
- (iii) **Monotonie** : Pour $w_k = (a, b)$ donné, on a $w_{k-1} = (u, v)$ où $0 \leq a - u$ et $0 \leq b - v$. Cette dernière condition oblige les éléments du chemin d'alignement \mathbf{W} à être espacés de manière monotone dans le temps.

Désormais, nous noterons $\mathcal{P}_{n,m}$ l'ensemble des (n,m) -chemins d'alignement.

Il existe une infinité de chemins de déformation qui satisfont aux conditions ci-dessus, mais dans ce rapport, nous nous focalisons uniquement sur le chemin qui minimise le **coût de déformation** $c_p(X, Y)$ entre X et Y associé à un (n,m) -chemin d'alignement $p := \{p_1, p_2, \dots, p_K\} \in \mathcal{P}_{n,m}$ donné par :

$$c_p(X, Y) = \frac{\sqrt{\sum_{k=1}^K \|x_{i_k} - y_{j_k}\|^2}}{K}$$

En d'autres termes, nous nous intéressons à la **déformation temporelle dynamique** $\text{DTW}(X, Y)$ entre les séries temporelles X et Y définie comme suit :

$$\text{DTW}(X, Y) := \min_{p \in \mathcal{P}_{n,m}} (c_p(X, Y)). \quad (11)$$

Remarque. On parle souvent de « distance » DTW par abus de langage : en général, il est possible d'avoir deux séries $X \neq Y$ telles que $\text{DTW}(X, Y) = 0$ (par exemple si le coût local c s'annule sur certains couples de points, ou si les séries contiennent des répétitions alignables à coût nul). Ainsi, DTW n'est pas une distance au sens strict (métrique) ; c'est au mieux une *pseudo-distance* selon les hypothèses faites sur C .

Nous montrons que le problème d'optimisation (11) possède au moins une solution.

Proposition 1 (Existence d'un chemin optimal). Soient $n, m \in \mathbb{N}^*$. Considérons deux séries temporelles X et Y telles que

$$X = \{x_k\}_{k \in [[0;n]]} \subset \mathbb{R}^n \quad \text{and} \quad Y = \{y_k\}_{k \in [[0;m]]} \subset \mathbb{R}^m.$$

Alors il existe $p^* \in \mathcal{P}_{n,m}$ tel que $\text{DTW}(X, Y) = c_{p^*}(X, Y)$.

Preuve. Posons $S := \{(1, 0), (0, 1), (1, 1)\}$. Montrons que d'abord qu'il existe un (n,m) -chemin d'alignement entre X et Y , i.e. que $\mathcal{P}_{n,m} \neq \emptyset$. Considérons le chemin

$$p = ((1, 1), (2, 1), \dots, (n, 1), (n, 2), (n, 3), \dots, (n, m)).$$

Alors, p vérifie $p_1 = (1, 1)$, $p_K = (n, m)$, et chaque pas appartient à S . Donc $p \in \mathcal{P}_{n,m}$.

Nous montrons maintenant qu'il est possible de majorer la longueur des éléments de $\mathcal{P}_{n,m}$. Prenons arbitrairement un élément $p = (p_1, \dots, p_K) \in \mathcal{P}_{n,m}$ avec $p_\ell = (n_\ell, m_\ell)$ pour tout $k \in [[1;K]]$. Pour tout $\ell \in [[1;K-1]]$, on a $(n_{\ell+1} - n_\ell, m_{\ell+1} - m_\ell) \in S$, donc

$$(n_{\ell+1} - n_\ell) + (m_{\ell+1} - m_\ell) \geq 1.$$

En sommant,

$$\sum_{\ell=1}^{K-1} [(n_{\ell+1} - n_{\ell}) + (m_{\ell+1} - m_{\ell})] \geq K - 1.$$

La somme télescope, et comme $p_1 = (1, 1)$ et $p_K = (n, m)$,

$$(n_K - n_1) + (m_K - m_1) = (n - 1) + (m - 1) = n + m - 2.$$

Ainsi $K - 1 \leq n + m - 2$ d'où $K \leq n + m - 1$.

On montre ensuite que $\mathcal{P}_{n,m}$ est fini. En fait, un chemin $p \in \mathcal{P}_{n,m}$ est entièrement déterminé par sa suite d'incrémentations

$$\{\Delta_1, \dots, \Delta_{K-1}\}, \quad \text{avec} \quad \Delta_k := p_{k+1} - p_k \in S = \{(1, 0), (0, 1), (1, 1)\}.$$

Puisque $p_1 = (1, 1)$ et que l'on reconstruit ensuite $p_{k+1} = p_k + \Delta_k$. Ainsi, pour tout $\ell \in [[1, K - 1]]$, il y a au plus $|S|^\ell = 3^\ell$ suites possibles de longueur ℓ . De plus, comme on a montré que $K - 1 \leq n + m - 2$, on déduit que :

$$|\mathcal{P}_{n,m}| \leq \sum_{\ell=0}^{n+m-2} 3^\ell < \infty.$$

Enfin, nous prouvons que le minimum est atteint. En fait, nous avons montré que l'ensemble $\{c_p(X, Y) \mid p \in \mathcal{P}_{n,m}\} \subset \mathbb{R}_+$ est fini et non vide, donc admet un minimum. Il existe donc $p^* \in \mathcal{P}_{n,m}$ tel que

$$c_{p^*}(X, Y) = \min_{p \in \mathcal{P}_{n,m}} c_p(X, Y),$$

ce qui achève la preuve. \square

Nous nous intéressons à désormais à la programmation de la DTW. Dans notre rapport, nous proposerons un programme en Julia qui permet de déterminer la DTW entre deux séries temporelles données.

3.2 Programmation dynamique et complexité

Pour deux séries temporelles C et Q telles que

$$C = \{x_k\}_{k \in [[0;n]]} \subset \mathbb{R}^n \quad \text{and} \quad Q = \{y_k\}_{k \in [[0;m]]} \subset \mathbb{R}^m,$$

on veut un algorithme qui permet de calculer $\text{DTW}(C, Q)$ qui existe (voir Proposition 1). Ici, on suppose que la $\text{DTW}(C, Q)$ est définie comme le minimum du coût du (n, m) -chemin d'alignement $c_p(C, Q)$, où :

$$c_p(C, Q) := \sum_{k=1}^K \|x_{i_k} - y_{j_k}\|.$$

L'idée pour réaliser un tel algorithme est de remplacer l'énumération de tous les (n, m) -chemins par une programmation dynamique. En d'autres termes, pour $1 \leq p \leq n$ et $1 \leq q \leq m$, on note les préfixes

$$C(1:p) = (x_1, \dots, x_p), \quad Q(1:q) = (y_1, \dots, y_q),$$

et on définit le coût minimal d'alignement entre les deux préfixes $X(1:p)$ et $Y(1:q)$ par :

$$D(p, q) := \text{DTW}(C(1:p), Q(1:q)).$$

La récurrence provient de la **condition de pas** suivante : tout chemin admissible arrivant en (p, q) provient nécessairement de l'un des trois prédécesseurs $(p - 1, q - 1)$, $(p - 1, q)$ ou $(p, q - 1)$, ce qui permet de calculer $D(\cdot, \cdot)$ case par case jusqu'à obtenir $\text{DTW}(C, Q) = D(n, m)$.

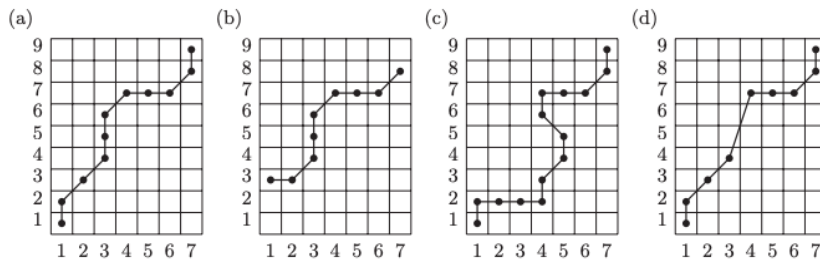


FIGURE 4 – Illustration de chemins d'indices pour une séquence C de longueur $n = 9$ et une séquence Q de longueur $m = 7$. (a) Chemin d'alignement admissible satisfaisant les conditions de bord, de monotonie et de taille de pas. (b) La condition de bord est violée. (c) La condition de monotonie est violée. (d) La condition sur la taille des pas est violée.

La DTW classique autorise des déformations temporelles très flexibles. Cette flexibilité peut toutefois provoquer un *sur-alignement* : un même point d'une série peut être aligné avec un grand nombre de points de l'autre série (multiples pas horizontaux/verticaux), ce qui peut produire des alignements peu réalistes. On introduit donc souvent des *contraintes* réduisant l'espace des chemins admissibles, améliorant à la fois la robustesse et la complexité, souvent à l'aide de la **Bande de Sakoe-Chiba** (voir Sakoe et Chiba [16]).

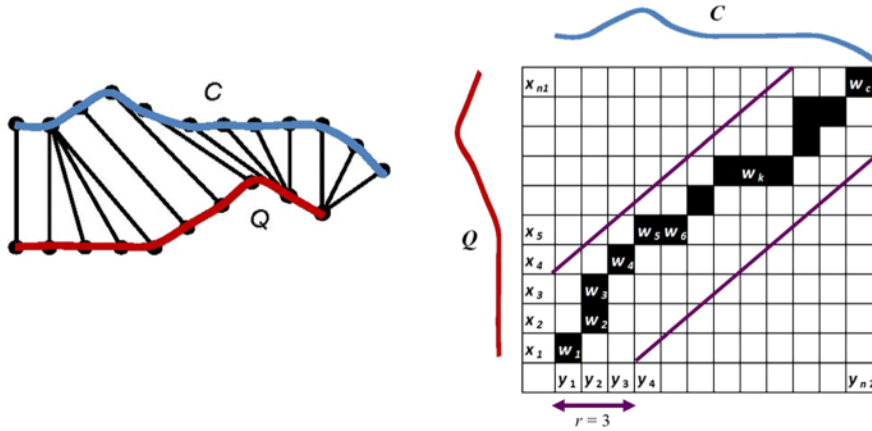


FIGURE 5 – Bande de Sakoe-Chiba : contrainte d'alignement $|n - m| \leq r$ autour de la diagonale.

Remarque (Effet algorithmique et complexité). Soit r une contrainte d'alignement pour concevoir une bande de Sakoe-Chiba. Dans la programmation dynamique, il suffit de ne calculer $\mathbf{D}(p, q)$ que pour

$$q \in [\max(1, p - r), \min(m, p + r)].$$

Ainsi, si $n \simeq m$, le nombre de cases évaluées est de l'ordre de $(2r + 1)n$ au lieu de nm , donc une complexité en temps typique $\mathcal{O}(nr)$ (et mémoire analogue si l'on stocke toute la bande).

On propose ci-dessous une implémentation simple de la DTW en JULIA, suivant directement le pseudo-code précédent. Pour ce faire, on définit une distance cumulative $\gamma(i, j)$ définie par

$$\gamma(i, j) = \|x_i - y_j\| + \min(\gamma(i - 1, j - 1), \gamma(i - 1, j), \gamma(i, j - 1))$$

et on procède comme dans le programme suivant.

Julia

```
# Dynamic Time Warping (DTW) -- version classique O(nm)
# s, t : vecteurs (ou tableaux 1D) de nombres
# dist : fonction de distance locale, par défaut |a-b|

function dtw(s, t; dist = (a,b)->abs(a-b))
    p = length(s)
    q = length(t)

    # D est de taille (p+1) x (q+1), indice de 1 a p+1 / q+1 en Julia
    D = fill(Inf, p+1, q+1)
    D[1, 1] = 0.0

    for i in 1:p
        for j in 1:q
            c = dist(s[i], t[j])
            D[i+1, j+1] = c + min(D[i, j+1], D[i+1, j], D[i, j])
        end
    end

    return D[p+1, q+1]
end
```

La complexité est en temps $\mathcal{O}(NM)$, si l'on souhaite effectuer le *backtracking*. En effet, le calcul de la matrice des coûts cumulés $\mathbf{D} \in \mathbb{R}^{n \times m}$ se fait case par case : pour chaque (p, q) , on effectue un nombre constant d'opérations (un minimum entre trois prédécesseurs puis une addition), donc le remplissage coûte $\mathcal{O}(nm)$. Le *backtracking* part de (n, m) et remonte vers l'un des trois prédécesseurs $(p - 1, q)$, $(p, q - 1)$ ou $(p - 1, q - 1)$; à chaque pas, la quantité $p + q$ diminue d'au moins 1, donc le nombre d'étapes est majoré par $n + m - 2$, ce qui donne un coût $\mathcal{O}(n + m)$. Au total, on a $\mathcal{O}(nm) + \mathcal{O}(n + m) = \mathcal{O}(nm)$.

4 Conclusion

Ce stage d'un mois au sein du Laboratoire de Mathématiques Informatique et Applications (LAMIA) a constitué une opportunité unique de confronter des connaissances théoriques académiques à une problématique de recherche active aux frontières des mathématiques et des neurosciences. Notre démarche nous a conduits à étudier les mécanismes qui permettent modélisation mathématique des neurones notamment à travers le modèle de neurone à impulsion d'Izhikevich [7].

Nous nous sommes ensuite focalisé sur la notion de train de spike et nous avons à savoir comment les mathématiques pouvaient aider à estimer le nombre de spike émis sur un train de spike donné puis sur une fenêtre de temps cible. Nous avons compris que le processus de Poisson homogène jouent un rôle clé dans l'étude du comptage des spikes. Toutefois, par manque de place dans le rapport, nous n'avons pas pu discuter du temps entre deux émissions de spike, vu en stage. Enfin, puisque les trains de spike sont des séries temporelles, nous avons cherché à comprendre comment on pouvait comparer deux trains de spike donnés. Puisque que Doncescu *et al.* [10] ont récemment utilisé la déformation temporelle dynamique pour comparer des trains de spike, nous nous sommes intéressé à cet outil.

Ainsi, un autre point crucial de notre travail a consisté à étudier la Déformation Temporelle Dynamique (DTW). Au-delà de l'implémentation algorithmique, nous avons attaché une importance particulière à la validation théorique de cet outil en montrant rigoureusement que le problème d'optimisation (11) qui définit la DTW entre deux séries temporelles admet bien une solution.

En somme, cette réflexion sur le lien entre modèle, données et mesure de similarité a été un fil conducteur tout au long du stage. J'ai pu me poser un certains nombre d'autres questions telle que l'étude de variantes de la DTW (contraintes de bande, pénalisation du warping, versions normalisées), la comparaison avec d'autres distances adaptées aux trains de spikes, et l'extension à des enregistrements multi-neurones de grande dimension. Ces pistes pourraient, par exemple, permettre de mieux caractériser la variabilité temporelle des réponses et d'améliorer l'analyse de données neuromorphiques issues d'expériences ou de simulations.

Ce stage s'est déroulé au sein du LAMIA, unité de recherche de l'Université des Antilles sous la direction du Docteur James Larrouy, que je remercie. Implanté sur les campus de Fouillole (en Guadeloupe) et de Schoelcher (en Martinique), le LAMIA est dirigé par le Professeur Gisèle Mophou et rassemble une quarantaine d'enseignant-chercheurs permanents et une quinzaine de chercheurs non permanents (ATER, doctorants, etc.). Ses activités couvrent un spectre large, allant des mathématiques appliquées (EDP, optimisation, contrôle, analyse numérique) à l'informatique (science des données, traitement d'images, apprentissage automatique, réseaux de neurones). Celles-ci sont organisées en plusieurs groupes thématiques interdisciplinaires. Le laboratoire entretient des partenariats avec des organismes tels que le CIRAD, l'INRAE, l'Institut Pasteur et le CHU, et bénéficie d'une animation scientifique régulière à travers des séminaires, des soutenances de thèse et des collaborations internationales. Cette immersion m'a permis de découvrir le fonctionnement quotidien d'un laboratoire de recherche et de bénéficier d'un environnement scientifique stimulant, fondé sur l'échange et la discussion. Cette première expérience m'a également familiarisé avec plusieurs exigences du métier de chercheur : gagner en autonomie, s'appropriier la littérature scientifique, formaliser rigoureusement une intuition, et faire dialoguer théorie et expérimentation. J'ai notamment constaté que l'avancement passe souvent par une alternance entre phases de lecture, d'essais (numériques ou conceptuels) et de reformulation des hypothèses, avant d'aboutir à un énoncé réellement démontrable.

En conclusion, ce stage a conforté mon goût pour les mathématiques appliquées. La rigueur nécessaire à la formalisation, couplée à la créativité requise pour la résolution de problèmes ouverts, confirme mon souhait de poursuivre mes études dans cette voie, avec un intérêt marqué pour la modélisation des systèmes complexes.

Références

- [1] D. J. DALEY AND D. VERE-JONES, *An Introduction to the Theory of Point Processes, Volume I : Elementary Theory and Methods*, Springer, New York, NY, 2 ed., 2003.
- [2] U. T. EDEN, *Point process notes*, n.d. Notes de cours.
- [3] A. GALVES AND E. LÖCHERBACH, *Infinite systems of interacting chains with memory of variable length—a stochastic model for biological neural nets*, *Journal of Statistical Physics*, 151 (2013), pp. 896–921.
- [4] W. GERSTNER AND W. M. KISTLER, *Spiking Neuron Models : Single Neurons, Populations, Plasticity*, Cambridge University Press, 2002.
- [5] D. HEEGER ET AL., *Poisson model of spike generation*, Handout, University of Standford, 5 (2000), p. 76.
- [6] J. J. HOPFIELD, *Neural networks and physical systems with emergent collective computational abilities.*, *Proceedings of the national academy of sciences*, 79 (1982), pp. 2554–2558.
- [7] E. M. IZHIKEVICH, *Simple model of spiking neurons*, *IEEE Transactions on Neural Networks*, 14 (2003), pp. 1569–1572.
- [8] E. M. IZHIKEVICH, *Dynamical systems in neuroscience*, MIT press, 2007.
- [9] R. E. KASS, U. T. EDEN, AND E. N. BROWN, *Analysis of Neural Data*, Springer, New York, NY, 2014.
- [10] L. LASSERRE, A. DONCESCU, AND F. FAUX, *One spiking neuron classification based on kolmogorov complexity*, in *2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, 2025, pp. 4012–4017.
- [11] W. MAASS, *Networks of spiking neurons : The third generation of neural network models*, *Neural Networks*, 10 (1997), pp. 1659–1671.
- [12] W. S. MCCULLOCH AND W. PITTS, *A logical calculus of the ideas immanent in nervous activity*, *The bulletin of mathematical biophysics*, 5 (1943), pp. 115–133.
- [13] M. MÜLLER, *Dynamic time warping*, in *Information Retrieval for Music and Motion*, Springer, Berlin, Heidelberg, 2007, ch. 4, pp. 69–84.
- [14] ———, *Dynamic time warping*. Lecture slides, *Information Retrieval for Music and Motion (Summer Term 2008)*, 2008. Lecture slides.
- [15] F. PONULAK AND A. KASIŃSKI, *Introduction to spiking neural networks : Information processing, learning and applications*, *Acta Neurobiologiae Experimentalis*, 71 (2011), pp. 409–433.
- [16] H. SAKOE AND S. CHIBA, *Dynamic programming algorithm optimization for spoken word recognition*, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26 (1978), pp. 43–49.