



# THÈSE

## En vue de l'obtention du Diplôme de Doctorat en Sciences

*Présenté par : Mme OURDIGHI Asmaa*

### *Intitulé*

**Contribution à l'étude de la robustesse des réseaux de neurones  
impulsionnels dans la reconnaissance de la parole**

*Faculté* : *Mathématiques et Informatique*  
*Département* : *Informatique*  
*Spécialité* : *Informatique*  
*Option* : *Reconnaissance des formes et intelligence Artificielle*

*Devant le Jury Composé de :*

<i>Membres de Jury</i>	<i>Grade</i>	<i>Qualité</i>	<i>Domiciliation</i>
<i>Mr. BENYETTOU Mohamed</i>	<i>Prof.</i>	<i>Président</i>	<i>USTO-MB</i>
<i>Mr. BENYETTOU Abdelkader</i>	<i>Prof.</i>	<i>Encadrant</i>	<i>USTO-MB</i>
<i>Mme. FIZAZI Hadria</i>	<i>Prof.</i>		<i>USTO-MB</i>
<i>Mr. CHIKH Mohammed Amine</i>	<i>Prof.</i>	<i>Examineurs</i>	<i>Univ. Tlemcen</i>
<i>Mr. RAHMOUN Abdellatif</i>	<i>Prof.</i>		<i>Univ. Sidi Bel-Abès</i>
<i>Mr. KHELFI Mohamed Fayçal</i>	<i>Prof.</i>		<i>Univ. Es Senia</i>

*Année Universitaire : 2016-2017*

*À la mémoire de ma très chère mère et de mon bébé Yacine.*

« إِيَّاكَ اللَّهُ وَإِنَّا إِلَيْهِ رَاجِعُونَ »

## *Remerciements*

En premier lieu, je remercie notre créateur tout puissant pour toutes les bonnes et ce que l'on croit mauvaises choses qui traversent notre courte existence. Dans ces circonstances, on ne peut que se rappeler et croire en sa grandeur, sa justice et sa glorieuse miséricorde.

Je tiens d'abord à remercier Mr Abdelkader BENYETTOU, mon directeur de thèse, pour sa contribution, son aide et qui par sa gentillesse et sa modestie nous a toujours offert un environnement de travail presque familiale au sein du laboratoire SIMPA. Aussi, Pendant toute ces années, il n'a cessé de m'encourager à accomplir cette thèse.

Tant, mes remerciements et toute ma gratitude s'adressent aux honorables membres de jury pour m'avoir honoré de leurs présences et d'avoir accepté de juger cette thèse. Je commence par le professeur Mohamed BENYETTOU, directeur du laboratoire LAMOSI, pour m'avoir fait l'honneur de présider le jury. Autant, pour avoir accepté d'examiner cette thèse, je remercie vivement Madame Hadria FIZAZI, professeur à l'université des sciences et technologie d'Oran Mohamed Boudiaf, Monsieur Mohammed Amine CHIKH, professeur à l'université de Tlemcen Abou Bekr Belkaid, Monsieur Abdellatif RAHMOUN, professeur à l'Ecole Supérieur en Informatique de Sidi Bel Abbés et Monsieur Mohamed Fayçal KHELFI, professeur à l'Université d'Oran Es-Sénia Ahmed Ben Bella.

Plus personnellement, je ne pourrai écrire ces quelques lignes sans remercier ceux avec qui j'ai partagé ma vie et qui après DIEU je leurs dois toute ma gratitude ; mes très chers parents et ma sœur. Je tiens à leurs témoigner toute la chance que j'ai de les avoir dans ma vie sur le plan personnel et professionnel. En conséquence, j'espère avoir assez honoré en ces modestes phrases mon adorable père et ma très chère et unique sœur qui ont beaucoup contribué à la finalisation de cette thèse.

Mes remerciements vont également vers mon mari pour avoir consacré beaucoup de son temps à m'aider. Je tiens absolument à le remercier pour tout son soutien, sa patience et sa participation à l'accomplissement de ce travail.

Finalement, je tiens à remercier toutes les personnes que j'ai oublié et qui ont contribué de près ou de loin à la réalisation et à la concrétisation de cette thèse. Je tiens à témoigner ma gratitude envers le Pr. IMINE Bachir et les membres du conseil scientifique de l'université de l'USTO et à leur tête Madame la Rectrice Pr. BENHARRAT Nacéra.

# Résumé

*A son apparition et en essayant de simuler le neurone biologique, le neurone artificiel a été modélisé sur un principe binaire où les sorties étaient digitales et asynchrones. Cependant, les tâches réalisées par cette première génération de neurones étaient très limitées. En imitant encore moins le modèle biologique, une nouvelle génération a été capable de réaliser des tâches complexes de régression et de classification avec toujours des limitations au niveau du traitement des données dite: temporelles.*

*Actuellement, le neurone impulsionnel ou le spiking neuron est considéré comme le modèle le plus ressemblant au modèle cérébral biologique avec un fonctionnement asynchrone accompagné d'une activité digitale où le temps devient un élément explicite au modèle. L'aspect dynamique de ces modèles très coûteux et plus délicat à simuler. Cependant, ils offrent des caractéristiques inédites et très puissantes en intégrant des propriétés nouvelles du fonctionnement neuronale et en particulier des caractéristiques du calcul temporel asynchrone qui permettent de nouvelles perspectives pour le traitement et l'analyse des données.*

*L'objectif de notre travail consiste à exploiter ce type de neurones dans des architectures neuronales classiques. Le cadre principal de notre application tournera autour de la reconnaissance de la parole.*

**Mots-clés :** *neurone impulsionnel, réseau de neurones, reconnaissance de la parole.*

# Abstract

*When it's appeared and with the intention to simulate the biological neuron, the artificial neuron was modeled on a binary process where the outputs were digital and asynchronous. However, the performances of the first generation neurons were very limited. When we tried to imitate even less the biological model, a new generation was able to perform complex regression and classification problems with some limitations to process a temporal data.*

*Now, the spiking neuron is considered as the most resembling the biological model with an asynchronous process and digital activity where time becomes an explicit element to the model. The dynamic aspect of these models is very complex and more difficult to simulate. However, they offer unprecedented and very powerful features by integrating new properties of neural behavior specifically the asynchronous temporal computation which allow new perspectives to process and analyze data.*

*The aim of our work is to use the spiking neurons in classical neuronal architectures. The main framework of our application will revolve around speech recognition.*

**Keywords:** *spiking neuron, neural network, speech recognition.*

# TABLE DES MATIERES

<b>INTRODUCTION GENERALE .....</b>	<b>1</b>
<b>CHAPITRE 1 VERS UN MODELE IMPULSIONNEL : APPORT DE LA BIOLOGIE</b>	
1 INTRODUCTION.....	6
2 ANATOMIE DU NEURONE BIOLOGIQUE.....	7
3 COMMUNICATION ENTRE NEURONES: DES ECHANGES IONIQUES A LA PRODUCTION DE L'INFLUX NERVEUX .....	8
3.1 <i>Potentiel de repos</i> .....	10
3.2 <i>Potentiel d'action</i> .....	11
3.3 <i>Au niveau de la Synapse</i> .....	11
4 CODAGE DE L'INFLUX NERVEUX .....	12
5 CARACTERISTIQUES DU CODAGE DE L'INFLUX NERVEUX .....	13
6 CONCLUSION.....	16
<b>CHAPITRE 2 MODELISATION DU NEURONE IMPULSIONNEL : HISTOIRE D'UNE EVOLUTION</b>	
1 INTRODUCTION.....	17
2 EVOLUTION DU NEURONE ARTIFICIEL .....	17
2.1 <i>Première Génération</i> .....	18
2.2 <i>Deuxième Génération</i> .....	18
2.3 <i>Troisième Génération</i> .....	19
3 MODÈLES DE NEURONES IMPULSIONNELS .....	19
3.1 <i>Modèles à base de conductance</i> .....	20
3.2 <i>Modèle comportemental</i> .....	23
3.3 <i>Modèles à base de seuillage et tire</i> .....	24
4 CODAGE DE L'INFORMATION .....	30
4.1 <i>Codages par cadence</i> .....	31
4.2 <i>Codages par impulsions</i> .....	33
4.3 <i>Corrélation et synchronie</i> .....	35
4.4 <i>Codage par corrélation renversée et reconstruction de stimulus</i> .....	36
5 CONCLUSION.....	36
<b>CHAPITRE 3 INTEGRATION DU MODELE A REPOSE IMPULSIONNELLE (SRM) DANS UNE ARCHITECTURE FEEDFORWARD</b>	
1 MOTIVATION .....	38
2 APPRENTISSAGE DES RESEAUX DE NEURONES IMPULSIONNELS.....	39
3 ARCHITECTURE DES RESEAUX DE NEURONES IMPULSIONNELS.....	40
4 DESCRIPTION DE L'APPROCHE UTILISEE .....	40
4.1 <i>Description de la formulation du SRM à délais temporels</i> .....	40
4.2 <i>Description de la propagation de l'information dans un SNN feed-forward</i> .....	44
4.3 <i>Description de l'algorithme SpikeProp à multi-spikes</i> .....	45
5 CONCLUSION.....	49
<b>CHAPITRE 4 EXPERIMENTATION DU SNN DANS DES CAS D'ECOLE</b>	
1 INTRODUCTION.....	50
2 PROBLEME DU XOR .....	51
2.1 <i>Expérimentation du SNN sur XOR</i> .....	51

2.2	<i>Paramètres du SNN</i> .....	52
2.3	<i>Première expérience</i> .....	52
2.4	<i>Deuxième expérience</i> .....	54
2.5	<i>Troisième expérience</i> .....	55
2.6	<i>Evaluation et Discussions</i> .....	56
3	IDENTIFICATION DES EMPREINTES DIGITALES SUR FVC2004 .....	58
3.1	<i>Expérimentation du SNN sur identification des empreintes digitales</i> .....	58
3.2	<i>Prétraitement</i> .....	59
3.3	<i>Paramètres du SNN</i> .....	61
3.4	<i>Evaluation et discussions</i> .....	62
4	RECONNAISSANCE LABIALE DES MOTS ISOLÉS SUR TULIPS 1.0.....	67
4.1	<i>Expérimentation du SNN sur Reconnaissance labiale des mots isolés</i> .....	69
4.2	<i>Paramètres du SNN</i> .....	70
4.3	<i>Evaluation et discussions</i> .....	70
5	DIAGNOSTIC DU CANCER DE SEIN SUR LA BASE DE DONNEES WBCD .....	72
5.1	<i>Expérimentation du SNN sur le diagnostic du cancer du sein</i> .....	74
5.2	<i>Evaluation du SNN dans le diagnostic du cancer du sein</i> .....	75
5.3	<i>Première expérience: Apprentissage, test et évaluations</i> .....	76
5.4	<i>Deuxième expérience: Apprentissage, test et évaluations</i> .....	80
5.5	<i>Comparaison avec d'autres approches</i> .....	82
6	CONCLUSION.....	84
<b>CHAPITRE 5 INTEGRATION DU SNN DANS LA DETECTION DE L'ACTIVITE DE LA VOIX (VAD)</b>		
1	INTRODUCTION.....	85
2	DETECTION DE L'ACTIVITE DE LA VOIX .....	85
1.1	<i>Principe de la VAD</i> .....	86
1.2	<i>Approches et paramètres du VAD</i> .....	87
3	CORPUS UTILISE .....	88
4	PRETRAITEMENTS ET PARAMETRES.....	90
5	STRUCTURE ET PARAMETRES DU SNN PROPOSE POUR LE VAD.....	93
6	METHODES D'EVALUATIONS DU SNN PROPOSE POUR LE VAD .....	94
7	TESTS ET EVALUATIONS DU SNN PROPOSE POUR LE VAD.....	95
8	CONCLUSION.....	98
<b>CONCLUSION GENERALE</b> .....		<b>100</b>
<b>BIBLIOGRAPHIE</b> .....		<b>102</b>
<b>PUBLICATIONS ET CITATIONS DE L'AUTEUR</b> .....		<b>116</b>

# LISTE DES FIGURES

1.1	<i>Différent type de neurones</i>	7
1.2	<i>Anatomie d'un neurone biologique</i>	7
1.3	<i>Conductances des canaux ioniques au sodium et au potassium en relation avec les étapes d'un potentiel d'action</i>	9
1.4	<i>Canaux de la membrane plasmique</i>	10
1.5	<i>Arrivée d'un potentiel d'action au niveau de la synapse</i>	12
1.6	<i>Impulsions (Spikes) émises durant 4s sur 30 neurones du cortex visuel d'un singe. Chaque '   ' représente un Spike</i>	13
1.7	<i>Caractéristiques neurocomputationnelles recensées des travaux d'Izhikevich [Izhikevich, 2003]. Les graphes représentent la réponse envoyée par un simple neurone impulsif lors d'une injection d'un courant sur 20 ms. On note qu'aucun modèle ne peut renvoyer ces vingt états à la fois.</i>	14
2.1	<i>Trois générations de neurones</i>	18
2.2	<i>Schéma électrique représentant les flux ioniques au travers de la membrane d'un neurone selon le modèle de Hodgkin-Huxley. C est la capacité membranaire, R la résistance, V le potentiel de membrane, E le potentiel d'inversion et I le courant injecté.</i>	20
2.3	<i>Schéma d'un circuit générique équivalent à un modèle impulsif comportementale</i>	24
2.4	<i>Schéma électrique du neurone LIF( Intègre et Tire à fuite). Un condensateur de capacité <math>C_m</math> est branché en parallèle avec une résistance.</i>	26
2.5	<i>Schéma électrique du modèle g.I.F.</i>	28
2.6	<i>Comportement du modèle SRM. Le neurone <math>n_i</math> émet un potentiel d'action (PA) à <math>t_0 = \hat{t}_i</math>. <math>\eta(t - \hat{t}_i)</math> décrit l'évolution du potentiel de membrane après un PA. <math>\kappa</math> décrit l'impact d'un potentiel post-synaptique (PPS) sur le potentiel de membrane. Cet impact dépend du temps écoulé depuis le dernier PA (<math>t - \hat{t}_i</math>). <math>\vartheta</math> décrit l'évolution du seuil après l'émission d'un PA</i>	30
2.7	<i><b>A</b> : Codage par cadence par utilisant la moyenne temporelle. <b>B</b> : Fonction de gain, schématique. La cadence de sortie <math>v</math> est donnée en fonction de la puissance d'entrée totale <math>I_0</math>.</i>	31
2.8	<i>Code par cadence utilisant la densité des impulsions.</i>	32
2.9	<i>Code par cadence utilisant l'activité d'une population : A. Un neurone post-synaptique reçoit les impulsions en entrée depuis la population m dont l'activité est <math>A_m</math>. L'activité de la population est définie par la fraction des neurones qui sont actifs dans un intervalle réduit <math>[t, t+\Delta t]</math> divisé par <math>\Delta t</math>.</i>	32
2.10	<i>Code par décharge utilisant le temps de la première impulsion : Le second neurone répond plus rapidement à un changement dans le stimulus que la première. Apparition de la stimulation marquée par la flèche.</i>	33
2.11	<i>Code par décharge utilisant les phases d'impulsions.</i>	35
2.12	<i>Code par le temps de la première impulsion (en haut), code par corrélation et synchronie (en bas).</i>	35
2.13	<i>Code par décharge utilisant la corrélation inversée.</i>	36
3.1	<i>Caractéristiques du modèle SRM</i>	41
3.2	<i>Modèle SRM à délais temporels</i>	42
3.3	<i>Changement du potentiel post-synaptique donné par <math>w_{ji}\varepsilon(t - t_i^{(g)} - d_k)</math></i>	43

3.4	Changement du potentiel du neurone donné par $\eta (t - t_j^{(f)})$	43
3.5	Un exemple graphique qui explique le travail d'un neurone impulsionnel.	45
3.6	Algorithme d'apprentissage pour un SNN a une couche	47
3.7	Un exemple simple d'un SNN à une couche cachée pour montrer quelles sont les variables utilisées.	48
4.1	A) ensemble de point linéairement séparables par un droite en deux classes. B) ensemble du XOR plus simple mais impossible à séparer par une droite, c'est un problème non-linéaire	51
4.2	Neurone SRM pour XOR :A) La fonction produisant la réponse impulsionnelle (en Bleu). B) La fonction produisant l'effet réfractaire (en rouge)	52
4.3	SNN monocouche [3 1] pour la résolution du problème du XOR: Un neurone pour chaque entrée plus un neurone pour le biais.	53
4.4	Représentation du PA pour les deux sorties possibles du SNN monocouche [3 1] pour XOR :( A) Réponse positive, le premier Spike de neurone de sortie est enregistré à t=8s. (B) Réponse négative, le premier Spike de neurone de sortie est enregistré à t=15s	54
4.5	SNN monocouche [2 1] pour la résolution du problème du XOR: Un neurone pour chaque entrée sans biais.	54
4.6	Représentation du PA pour de la sortie positive du SNN monocouche [2 1] pour XOR (la sortie négative le PA n'atteint pas le seuil)	55
4.7	SNN monocouche [1 1] pour la résolution du problème du XOR: Un neurone les deux entrées.	55
4.8	Représentation du PA pour de la sortie négative du SNN monocouche [1 1] pour XOR enregistré à t=12s (la sortie positive arrive généralement avant t=10 s )	56
4.9	Exemple Courbe de l'erreur $(t_j^{(1)} - \hat{t}_j^{(1)})$ du SNN pour XOR extrêmement instable.	56
4.10	Courbe de l'erreur quadratique, obtenu par l'algorithme du PMC qu'on adapté sur un SNN.	57
4.11	Exemples empreintes de la base DB1 de la FVC2004 des trois individus sélectionnés pour l'apprentissage	59
4.12	Base d'apprentissage des empreintes digitales utilisées pour l'entraînement du SNN.	59
4.13	Trois d'empreintes digitales après avoir subis conversion, rognage et binarisée	60
4.14	Algorithme de squelettisation de Hilditch	61
4.15	Binarisation (image au milieu) et squelettisation (image à droite) d'une empreinte digitale à partir de l'image de l'empreinte (image à gauche)	61
4.16	Structure et codage utilisés dans le SNN pour l'identification des empreintes digitales	62
4.17	Graphes de variation de l'erreur quadratique moyenne durant la première expérience	63
4.18	Empreintes digitales binarisées et bruitées à 10% et à 30% par une distribution Gaussienne	64
4.19	Graphes de variation de l'erreur quadratique moyenne durant la deuxième expérience	66
4.20	Empreintes digitales squelettisées et bruitées à 10% et à 30% par une distribution Gaussienne	66
4.21	Exemples de séquences d'images de la base Tulips: (A): Séquences de 7 images montrant une femme prononçant le « One », (B): Séquences de 8 images montrant un homme prononçant le «Two »	68
4.22	Six caractéristiques de l'ouverture de la bouche	69
4.23	Exemples de séquences d'images de la base Tulips ayant subit un seuillage: (A): Séquences de 7 images montrant une femme prononçant le « One », (B): Séquences de 8 images montrant un homme prononçant le «Two »	69
4.24	SNN monocouche [6 4] pour la reconnaissance de labiale des mots isolés pour la reconnaissance labiale des mots isolés sur Tulips 1.0	70

4.25	<i>Graphes de variation de l'erreur quadratique du SNN pour la reconnaissance labiale des mots isolés sur Tulips 1.0</i>	71
4.26	<i>Exemples extraits de la Wisconsin breast cancer database (WBCD)</i>	74
4.27	<i>Topologie du SNN monocouche [9 1] pour le diagnostic du cancer du sein sur WBCD</i>	77
4.28	<i>Evolution du taux de reconnaissance en variant les synapses <math>l=1..10</math> (<math>\tau_m=5</math> and <math>\tau_s=3</math>)</i>	78
4.29	<i>Différents graphes de la courbe AUC en variant les synapses <math>l=1..10</math> (<math>\tau_m=5</math> and <math>\tau_s=3</math>)</i>	78
4.30	<i>Graphes AUC du ROC curves des Test A et Test B ayant donnés un taux de reconnaissance de 99,26%</i>	79
4.31	<i>Exemple de champ réceptive gaussien (CRG): la ligne vertical en rouge représente la valeur à coder et les intersections avec les gaussiennes retournerons les valeurs des délais, si l'intersection se trouve dans la zone « No firing » le délai ne sera pas pris en considération (Dans la figure, l'intervalle supposé être de [0, 1] dans l'axe des Y a été transformé en délais de [0, 10] ms pour une meilleur représentation</i>	80
4.32	<i>Architecture du SNN à population de neurones (CRG) pour le diagnostic médicale sur WBCD</i>	81
4.33	<i>Graphes AUC du ROC du SNN à population de neurones (CRG) ayant donnés un taux de reconnaissance de 99,26% et un AUC de 0,989</i>	82
5.1	<i>Exemple illustrant le principe de la détection d'activité vocale</i>	86
5.2	<i>Processus général d'un algorithme de détection d'activité vocale</i>	87
5.3	<i>Représentation du découpage manuel du signal « sp01.wav » sous l'environnement « Aéroport »: Clean (bleu), 15 dB (rouge), 10 dB (jaune), 5 dB (vert) et 0 dB (noir)</i>	90
5.4	<i>Représentation du paramètre short-time average power parameter (E) à partir du signal clean « sp01.wav ».</i>	91
5.5	<i>Représentation du paramètre zero-order most likelihood parameter (Z) à partir du signal clean « sp01.wav ».</i>	92
5.6	<i>Représentation du paramètre pitchperiod difference parameter (P) à partir du signal clean « sp01.wav »</i>	92
5.7	<i>Représentation des variations des caractéristiques calculées du signal (a) sous l'environnement Aéroport: Clean (b), 15 dB (c), 10 dB (d), 5 dB (e) et 0 dB (f)</i>	93
5.8	<i>Structure du SNN monocouche proposé pour réaliser le VAD</i>	94
5.9	<i>Graphes AUC du VAD proposé à base de SNN du signale Clean et en variant le SNR dans l'environnement « Aéroport</i>	96
5.10	<i>Graphes AUC du VAD proposé à base de SNN du signale Clean et en variant le SNR dans l'environnement « Rue »</i>	96
5.11	<i>Graphes AUC du VAD proposé à base de SNN du signale Clean et en variant le SNR dans l'environnement « Voiture »</i>	97
5.12	<i>Graphes AUC du VAD proposé à base de SNN du signale Clean et en variant le SNR dans l'environnement « Restaurant »</i>	97

# LISTE DES TABLES

4.1	<i>Codage de type Time-to-first-Spike des Entrées et de la sortie désirée du SNN.</i>	53
4.2	<i>Récapitulatif des informations sur les bases d'empreintes.</i>	58
4.3	<i>Paramètres du SNN pour l'identification des empreintes digitales</i>	62
4.4	<i>Paramètres du SNN retenus pour la première expérience.</i>	63
4.5	<i>Evaluation des résultats du SNN sur l'identification des empreintes digitales sur des images binarisées</i>	64
4.6	<i>Evaluation des résultats du SNN sur l'identification des empreintes digitales sur des images binarisées et bruitées à 10% et à 30%</i>	65
4.7	<i>Paramètres du SNN retenus pour la deuxième expérience</i>	65
4.8	<i>Evaluation des résultats du SNN sur l'identification des empreintes digitales sur des images squelettisées</i>	66
4.9	<i>Evaluation des résultats du SNN sur l'identification des empreintes digitales sur des images squelettisées et bruitées à 10% et à 30%</i>	67
4.10	<i>Description de la base Tulips 1.0</i>	68
4.11	<i>Evaluation des résultats de l'apprentissage du SNN pour la reconnaissance labiale des mots isolés sur Tulips 1.0.</i>	71
4.12	<i>Evaluation des résultats de reconnaissance du SNN pour la reconnaissance labiale des mots isolés sur Tulips 1.0.</i>	71
4.13	<i>Description des attributs de la Wisconsin breast cancer database (WBCD)</i>	73
4.14	<i>Répartition des échantillons de la Wisconsin breast cancer database (WBCD) entre base d'apprentissage et base de test</i>	75
4.15	<i>Evaluations des Test A et Test B structures optimales du SNN dans le diagnostic médicale sur WBCD</i>	79
4.16	<i>Paramètres du SNN à population de neurones (CRG) pour le diagnostic médicale sur WBCD</i>	82
4.17	<i>Différents taux de reconnaissances obtenus dans la base WBCD</i>	83
5.1	<i>Représentation de la base de données NOIZEUS</i>	89
5.2	<i>Résultats du TR, AUC, Pd et Nd du VAD proposé à base de SNN sur les environnements aéroport, rue, voiture et restaurant avec un signal clean et un SNR variant de 5 à 15 dB.</i>	98

# LISTE D'ABREVIATIONS

<b>RNA</b>	Réseaux De Neurones Artificiels
<b>PA</b>	Potentiel d'Action
<b>SNN</b>	Réseaux de neurones impulsionnels
<b>HH</b>	Modèle de Hodgkin et Huxley
<b>PSP</b>	Neurones Présynaptiques
<b>IF</b>	Modèle Intègre & Tire
<b>LIF</b>	Modèle de Leaky-Integrate-and-Fire
<b>QIF</b>	Modèle Quadratic Integrate and Fire
<b>EIT</b>	Modèle Exponential Integrate and Fire
<b>GIF</b>	Modèle à conductance Intègre & Tire
<b>SRM</b>	Modèle à Réponse Impulsionnelles
<b>PMC</b>	Réseaux De Neurones Multicouches
<b>ROC</b>	Receiver Operating Characteristics
<b>FMR</b>	False Match Rate
<b>FNMR</b>	False Non Match Rate
<b>CRG</b>	Champs Réceptifs Gaussien
<b>FAR</b>	False Accept Rate
<b>FRR</b>	False Reject Rate
<b>AUC</b>	Area Under ROC Curve
<b>VAD</b>	Détection de l'Activité de la Voix
<b>LPC</b>	Linear Prediction Coefficients
<b>Pd</b>	Probabilité de bonne décision dans les zones actives
<b>Nd</b>	Probabilité de bonne décision dans les zones inactives
<b>RBF</b>	Radial Basis Functions
<b>PNN</b>	Probabilistic Neural Networks

<b>GRNN</b>	Generalized Regression Neural Networks
<b>Fuzzy ARTMAP</b>	Ant Colony-based Data Miner
<b>SVM</b>	Support Vector Machine
<b>SVM-CFS</b>	Support vector machines combined with feature selection
<b>CFW</b>	Classification with Filter & wrapper approach
<b>AMMLP</b>	Artificial metaplasticity Multilayer Perceptron

# INTRODUCTION GENERALE

**E**n 1963, Hodgkin et Huxley ont ficelé les premiers liens solide entre la neuroscience et l'ingénierie. Leurs travaux rapportent une description mathématique assez exacte du potentiel d'action. Mais encore très avancé pour être exploité à l'époque, on était loin de se douter que non seulement ils venaient de tracer avec précision les premiers concepts d'une toute nouvelle génération de neurone artificiel, mais aussi de révolutionner les recherches dirigée vers l'intelligence neuro-computationnel. Effectivement, Leur modèle, baptisé longtemps après « *neurone impulsionnel* », ne sera relancé d'une manière opérationnelle qu'en 1997 à travers les travaux de Maas [Maass, 1997].

Naturellement, plusieurs sujets de recherches ont alors commencé à décrire en détail la dynamique des variations du courant ionique. Ainsi, le neurone impulsionnel a élevé de nouveau le niveau du réalisme biologique manquant aux générations précédentes. Cependant, la reproduction précise de l'activité d'un neurone fait appel à un nombre excessif de variables, qui empêchent souvent la compréhension du modèle et le rend difficile à mettre en œuvre.

Aussi, la construction d'une architecture à base de neurones impulsionnels se voit limité par de nouvelle contrainte encore ignorées dans les architectures classiques. Typiquement, dans ces modèles, l'information spatiotemporelle se communique de la même manière que dans les neurones biologiques. Ainsi, au lieu d'employer le codage par cadence, ces neurones utilisent le codage par impulsion ; mécanismes où les neurones reçoivent et envoient un train d'impulsions. Par conséquent, le codage des données analogiques est devenu un facteur crucial à déterminer.

Malencontreusement, toutes ces nouvelles contributions apportées à la définition et au mode de propagation de l'information à travers un réseau de neurones impulsionnels ne font que compliquer le processus d'apprentissage de ce type de réseau. Il est claire qu'on connaissait déjà ce genre de problème dans les modèles classique à travers le choix de l'architecture, des types et des règles d'apprentissage, les problèmes d'initialisations aléatoires des paramètres du réseau ou encore le problème des minimums locaux. Ceci n'est rien comparé à la difficulté de construire un algorithme d'apprentissage pour des réseaux de neurones impulsionnels. Surtout si on choisit le mode supervisé contrairement aux modèles classiques.

## DEFINITIONS DU PROBLEME

La parole constitue une grande partie de l'information communiquée. Ainsi, la reconnaissance automatique de la parole est une technique informatique très intéressante qui permet d'analyser la parole captée au moyen d'un microphone pour la transcrire sous la forme d'information exploitable par une machine. La synthèse de la parole, l'identification du locuteur ou encore la vérification du locuteur font partie des techniques du traitement et de la reconnaissance de la parole. Cependant, il existe un inconvénient important qui affecte la plupart des systèmes de traitement de la parole. Cet inconvénient est le bruit dans les environnements d'acquisition du signal vocale et audio et son effet est néfaste sur les performances des systèmes de reconnaissance de la parole.

Dans notre thèse, on a cherché à construire un système à base d'un réseau de neurones impulsionnels (SNN) fondé autour d'un apprentissage supervisé pour réaliser une reconnaissance de la parole. Aussi, on a cherché à tester sa robustesse au niveau de la phase de généralisation.

Dans ce sens, la validation de notre architecture proposée à nécessité une phase préliminaire d'expérimentation sur des cas d'école afin de tester la performance et la robustesse de notre modèle dans divers problèmes de classification.

## MOTIVATIONS

Les réseaux de neurones impulsionnels peuvent transmettre et recevoir des quantités importantes d'informations grâce à la synchronisation relative des impulsions. En tenant compte du moment du tir de l'impulsion, tout l'intérêt porte à déterminer avec précision la modélisation des interactions synaptiques entre les neurones. Ainsi, les SNNs ont réussi à surmonter la puissance de calcul des réseaux de neurones à base d'unités à seuil ou sigmoïdes.

Sur des bases dynamiques événementielles de traitement, les SNNs ouvrent de nouveaux horizons pour le développement des systèmes de reconnaissances en augmentant leurs capacités de mémorisation et leurs facultés d'adaptation rapides sur n'importe quel domaine d'application [Meftah, 2011].

En expansion, l'utilisation de ces architectures dynamiques apparaît aujourd'hui un peu partout dans les travaux de recherche. Leurs performances commencent à s'imposer petit à petit dans des domaines assez variés. Dans la reconnaissance de la parole, par exemple, les travaux de [Verstraeten et al., 2005] utilisent la machine d'état liquide (LSM) à base du modèle impulsionnel « intègre et tire » (LIF) en intégrant une simulation d'oreille pour effectuer l'encodage des informations. Dans [Schrauwen, 2002 ; Booi, 2004] on trouve différentes applications sur la classification, auto-association et de clustering. Aussi, On trouve des architectures à base de neurones impulsionnels capable de réaliser la

segmentation des images, tels que les travaux de [Meftah, 2010] sur des images cellules microscopiques.

Actuellement, le principal défi consiste à découvrir les règles efficaces d'apprentissage qui pourraient profiter des fonctionnalités spécifiques du SNN tout en conservant les bonnes propriétés de modèles connexionnistes classiques.

## OBJECTIVES DE LA THESE

Au départ, l'objectif de notre thèse portait sur l'expérimentation d'une architecture de neurones impulsionnels sur la reconnaissance de la parole sur des signaux nets et bruités. Ensuite, on devait observer la performance et tester la robustesse de notre système. Finalement, à travers l'expérimentation, on a décidé d'orienter notre approche vers une problématique de la reconnaissance de la parole beaucoup plus ciblée où l'environnement bruité constitue une impasse beaucoup plus réelle dans le traitement de la parole.

Actuellement, la nécessité d'extraire les informations utiles du signal de parole est devenue une préoccupation majeure, d'où l'utilisation de la détection de l'activité vocale (VAD) qui permet de détecter les régions voisées et non-voisées d'un signal donnée. Elle représente une tâche fondamentale dans presque n'importe quel système de traitement et de reconnaissance de la parole.

Ainsi, on a cherché à prouver que l'on pouvait construire une architecture SNN simple et performante capable de distinguer les trames voisées et non-voisées dans un signal de parole en dépit de l'environnement où le signal a été extrait.

## STRATEGIE ET METHODOLOGIE

Pour construire notre phase expérimentale, on a choisi de travailler sur le modèle à réponse impulsionnelle (SRM). Ce modèle permet de modéliser la production des réponses impulsionnelles générées à partir d'événements pré-synaptiques avec un réalisme biologique. Ces unités peuvent être intégrées sur différents types d'architectures: monocouche, multicouche, feedforward ou récurrente. Aussi, on peut adapter les paramètres de ces architectures par un apprentissage non-supervisé ou supervisé. Généralement, il est très rare de rencontrer des applications complexes des SNNs à base d'un apprentissage supervisé même si il existe un nombre modeste d'algorithmes fondés sur ce principe. La difficulté de coder le flux d'informations propager dans les SNNs rend difficile la modélisation d'un comportement désiré.

Dans notre approche, en dépit de tous les complications observées dans l'apprentissage supervisé, on a choisi d'expérimenter une variante de l'algorithme de la descente du gradient « SpikProp » en simplifiant la structure de notre réseau pour permettre de réduire au maximum les paramètres du SNN.

Dans le choix du codage des données, on a choisi de tester différents type de codage : codage du temps d'impulsion, le codage par ordre de rang, le codage par population de neurones à base de champs réceptives gaussiens, le codage par phases et le codage par temps de la première impulsion.

Dans les différentes implémentations, on a choisi de tester notre système sur différentes problématiques dans le cadre de la classification, la vérification et les systèmes d'aide à la décision.

## ORGANISATION DE LA THESE

Le premier chapitre de cette thèse rappelle que comme tous modèle bio-inspirée, la recherche en neurosciences tend à produire des théories à partir du modèle biologique, en l'occurrence des théories du fonctionnement du système nerveux. Une théorie s'exprime concrètement par un modèle, qui est une idéalisation de la réalité guidée par la connaissance approfondie de l'aspect biophysique et computationnel de l'unité biologique. Dans ce sens, on recense les différents propriétés et caractéristiques comportemental de la cellule nerveuse afin de mieux comprendre l'origine des caractéristiques de la formalisation du neurone impulsionnel.

Dans le second chapitre, on revient sur les différentes modélisations du neurone artificiel. Son évolution sous forme de trois générations de neurone est décrite de façon à appuyer l'origine des motivations qui ont induit vers une reconduction du neurone impulsionnel. On montre aussi qu'à partir de son relancement, ce formalisme a confirmé encore la nécessité de proposé une description de nombreux courant ioniques. Naturellement, ceci a permis d'affiner cette théorie et de stimuler de nouvelles expériences dans cette direction en générant différents nouveaux modèles. Dans le même contexte, on expose aussi les types de codage d'information qui s'impose comme une étape importante dans la modélisation des caractéristiques neurocomputationnelles observées sur le plan biologique.

Dans le troisième chapitre, on introduit notre stratégie de travail en présentant le modèle impulsionnel choisi, c-à-d le modèle à réponse impulsionnelle. On l'illustre à travers ces différentes propriétés et caractéristiques. Aussi, on présente des algorithmes d'apprentissages sur des structures monocouches et multicouches à base de SRM.

Dans le quatrième chapitre, on présente les différentes étapes et applications pour tester la validité et la rentabilité de notre approche. Naturellement, on est passé du test de la performance du réseau SNN à travers des cas d'école simple tel que le problème du Xor. Ensuite, on a cherché d'étendre notre phase expérimentale à travers des applications sur la reconnaissance labiale des mots isolés, l'identification des empreintes digitales et le diagnostic du cancer de sein à travers des systèmes d'aide à la décision à base de SNN monocouche. Ceci nous a permis de peaufiner notre approche de façon à devenir plus performante et plus robuste.

Finalemment, après l'avoir testé sur différentes applications sur des cas d'école, on a dédié le dernier chapitre à l'expérimentation de notre approche sur la reconnaissance de la parole. La détection de l'activité de la voix (VAD) est l'axe qu'on a choisi pour tester notre approche avec une application où l'on pouvait expérimenter sur un plan plus concret la robustesse du SNN monocouche à base de modèle SRM.

# CHAPITRE 1

## VERS UN MODELE IMPULSIONNEL: APPORT DE LA BIOLOGIE

### 1 INTRODUCTION

Avec l'évolution des ressources technologiques, la neuroscience est devenue très rapidement un axe interdisciplinaire favorisant la contribution de chercheurs de différentes spécialités. Cet intérêt croissant à combiner plusieurs disciplines est relié à la complexité de l'étude du système nerveux qui reste jusqu'à maintenant assez énigmatique.

L'intégration de la neuroscience dans le domaine de l'intelligence artificielle a commencé à se confirmer, petit à petit depuis 1943, après que W. Mcculloch et W. Pitts proposèrent le « *neurone formel* »: première modélisation grossière du neurone biologique.

Cependant, son importance prend un élan particulier en 1963 après la publication des travaux de J. C. Eccles, A. L. Hodgkin et A.F. Huxley. En effet, avec la découverte des mécanismes ioniques impliqués dans l'excitation et l'inhibition des portions périphériques et centrales de la membrane cellulaire des nerfs, la communication entre neurones n'est plus devenue un mystère. Cette révélation encore méconnu à l'époque leurs a valu un prix Nobel en médecine.

En revanche, dans cette même période, l'exploitation des réseaux de neurones artificiels (RNA) ne faisaient que des pas timides vers la confirmation de leurs performances en tant que techniques cognitives. On était loin de se douter à ce stade de l'impact et de l'étendu de ces travaux sur la modélisation du neurone artificiel telle qu'on l'a réinterprété actuellement: « *modèle impulsionnel* ».

Dans ce chapitre, on expose brièvement ce que l'on doit savoir sur un neurone biologique en décortiquant l'anatomie et l'aspect comportemental de la cellule. Ceci permettra de mieux comprendre le neurone impulsionnel considéré comme la plus

récente réinterprétation du neurone artificielle. En effet, on ne peut se permettre de modéliser une technique bio-inspirée que si l'on est suffisamment informé des différents caractéristiques et propriétés du modèle de base.

## 2 ANATOMIE DU NEURONE BIOLOGIQUE

Le terme « neurone » a été proposé en 1881 par l'anatomiste allemand Heinrich Wilhelm Waldeyer. Depuis, le neurone s'est vu attribuer plusieurs définitions selon son anatomie, sa position et son fonctionnement [Arbib, 1998].

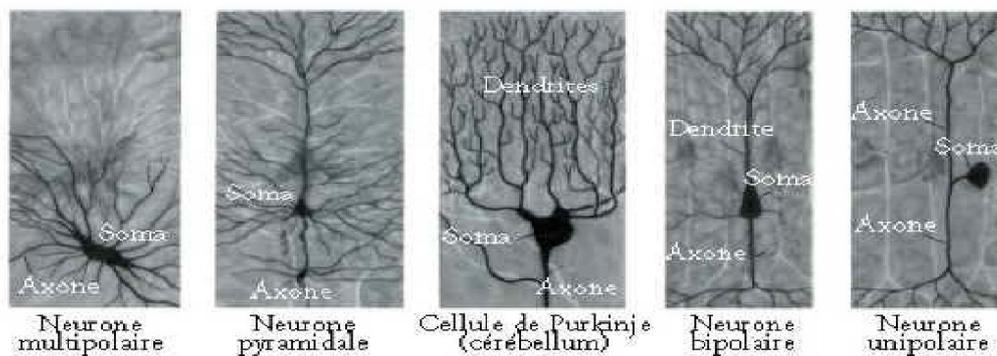


Figure 1.1 - Différent type de neurones [[http://www.corpshumain.ca/ Neurone.php](http://www.corpshumain.ca/Neurone.php)].

On recense que le système nerveux des humains comprend environ 86 à 100 milliards de neurones sous différentes formes (voir Figure 1.1) : Neurone multipolaire, neurone pyramidale, cellule de purkinje (Cérébellum), neurone bipolaire, neurone unipolaire, etc.

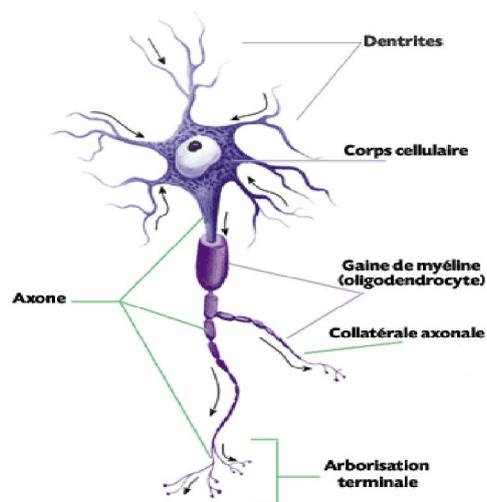


Figure 1.2 - Anatomie d'un neurone biologique [[http://www.tpe.walrey.fr /cerveau /foncgen/neurone.html](http://www.tpe.walrey.fr/cerveau/foncgen/neurone.html)].

La Figure 1.2 est la représentation typique et vulgarisée d'un neurone biologique. Sa structure de base est presque identique aux autres formes existantes. Dans ce contexte, on définira généralement un neurone biologique comme une unité se composant d'un « corps cellulaire » (le soma ; corps du neurone) et d'un « noyau ». L'ensemble des ramifications du corps cellulaire forment tout autour de ce dernier ce que l'on nomme « les dendrites ». Celles-ci sont parfois si nombreuses que l'on parle alors de chevelure dendritique ou d'arborisation dendritique. Elles rentrent en contact avec d'autres neurones, spécifiquement de l'ordre de 10000, et dont le rôle est de recevoir des informations, électriques ou chimiques de l'extérieur, et les acheminer vers le soma.

Dans les cellules du cortex, on peut distinguer un prolongement particulier et unique qu'on appelle « l'axone ». Sa longueur peut varier suivant les neurones de quelques microns à 1,50 mètre. Si l'excitation des entrées est suffisante, un signal de sortie (potentiel électrique) chemine le long de l'axone pour être transmis jusqu'à son arborisation terminale (soit aux autres neurones, soit à un muscle, soit à une glande). Ce phénomène électrique qui se propage sans atténuation le long de l'axone, s'appelle le « potentiel d'action » (PA) [Groupe Réseaux De Neurones, 1991 ; Brette, 2003].

Néanmoins, se concentrer uniquement sur l'anatomie et le comportement physique de la cellule en ignorant les caractéristiques de son fonctionnement peut induire à une modélisation fatidique où on favorisera uniquement l'aspect automatique du formalisme. Donc, il est nécessaire de comprendre le comportement d'une cellule biologique au moment de son interaction avec son environnement d'une manière interne et externe.

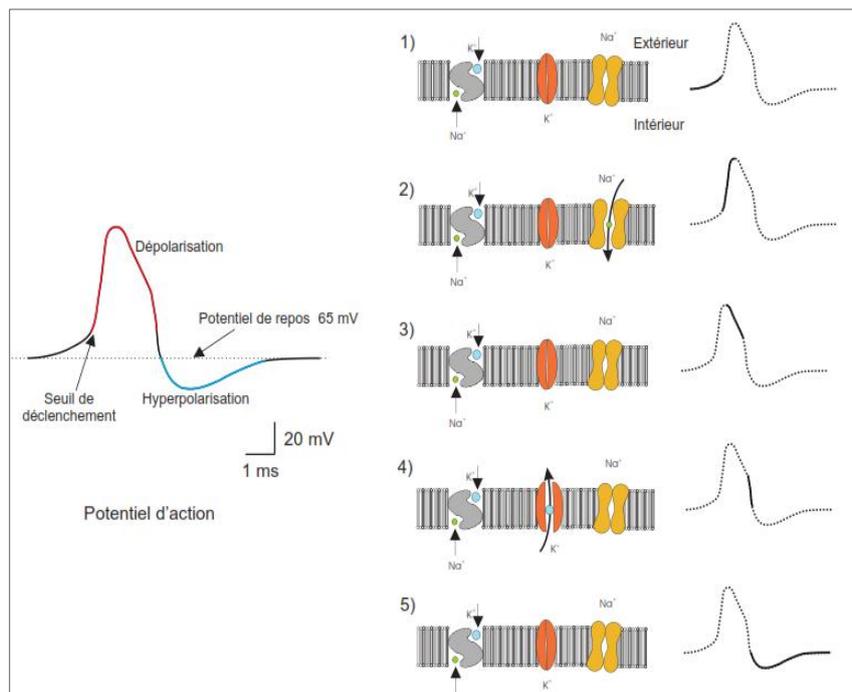
### **3 COMMUNICATION ENTRE NEURONES: DES ECHANGES IONIQUES A LA PRODUCTION DE L'INFLUX NERVEUX**

A l'origine, on s'est longtemps demandé par quels mécanismes les neurones reçoivent-ils le flux nerveux jusqu'à ce que des expériences réalisées sur les cellules nerveuses du calamar permettent d'éclaircir la nature de la communication entre neurones [Hodgkin et al., 1952].

Les premières observations ont révélé l'implication des concentrations ioniques et des changements de perméabilité membranaire dans la production des signaux électriques. Les canaux ioniques qui percent la membrane plasmique du neurone sont des protéines membranaires intrinsèques, qui peuvent être plus ou moins complexes. On peut cependant distinguer les canaux passifs des pompes à ions, ces dernières permettent de rééquilibrer les concentrations ioniques de part et d'autre de la membrane [Catterall, 1988]. Les canaux passifs peuvent être séparés en trois grandes

catégories, en fonction de ce qui cause l'ouverture de ces canaux : les canaux ouverts par des ligands intracellulaires, ceux ouverts par des ligands extracellulaires et ceux commandés par le voltage [Chevallier, 2009].

La Figure ci-dessous illustre les différentes phases de la variation de l'état électrique du neurone au cours de l'injection d'un signal.



**Figure 1.3** - Conductances des canaux ioniques au sodium et au potassium en relation avec les étapes d'un potentiel d'action [Buhry , 2010]

Initialement, dans la phase (1) dite « potentiel de repos », la membrane est dans son état d'inactivité. Une ou plusieurs stimulations du neurone entraînent une augmentation du potentiel où le neurone passe en mode de « dépolarisation », illustré par phase ascendante (2). Inévitablement, la somme de ces stimuli atteint un seuil critique qui induit le neurone à provoquer un « potentiel d'action » marquant la fin de la phase (2). Le potentiel redescend ensuite provoquant la « phase repolarisation », répartie sur deux étapes (3, 4). Arriver à la phase (5), la membrane atteint un niveau d'hyperpolarisation qui l'empêche temporairement de reproduire un potentiel d'action, c'est la « période réfractaire ». Après cette dernière, la membrane revient à son état de repos [Touzet, 1992].

Ces différentes phases sont toutes les résultats d'un processus d'échange d'ions. Dans ce qui suit, on expliquera brièvement les transactions génératrices de l'activité et du repos de la membrane du neurone.

### 3.1 POTENTIEL DE REPOS

La physiologie du neurone est liée aux propriétés de la membrane nerveuse et au métabolisme de la cellule. Le neurone est polarisé lorsqu'il est au repos, et on remarque qu'il y a une différence de potentiel entre le milieu extracellulaire et le milieu intracellulaire, mesuré à  $-70$  mV (elle peut varier entre  $-60$  mV et  $-100$  mV). Les analyses chimiques de ces deux milieux ont démontré qu'il y existe une différence de concentration ionique ( $\text{Na}^+$  et  $\text{K}^+$ ), cette différence est à l'origine du potentiel de repos. [Brette, 2003 ; Touzet, 1992].

Pour maintenir la stabilité du potentiel de repos, la cellule dispose de plusieurs pompes ioniques ( $\text{Na}^+$ ,  $\text{K}^+$ , ...). La pompe sodium/potassium ou  $\text{Na}^+/\text{K}^+$ -ATP-ase est une enzyme transmembranaire (protéine) qui assure le transport des ions potassium et sodium (voir Figure 1.4). Elle transporte 3 ions  $\text{Na}^+$  au milieu extérieur de la cellule, et 2 ions  $\text{K}^+$  en son milieu intérieur à l'encontre de leur gradient de concentration. On appelle ce mécanisme perméabilité sélective de la membrane aux ions  $\text{Na}^+$  et  $\text{K}^+$ . C'est ce qui garde la stabilité électrique de la cellule.

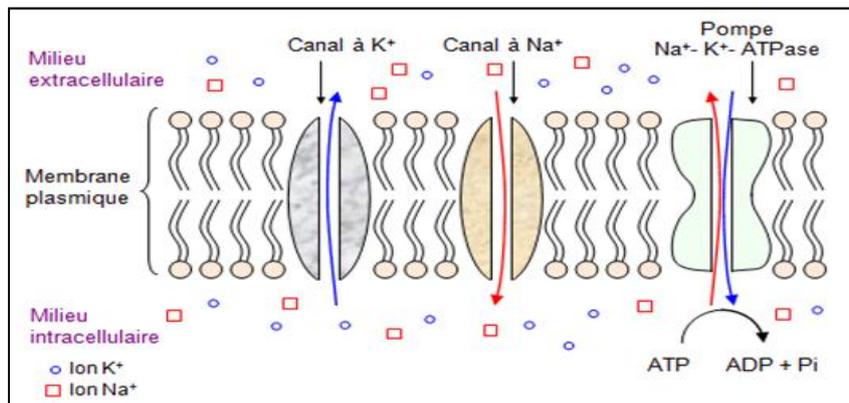


Figure 1.4 - Canaux de la membrane plasmique.

Il existe d'autres protéines appelées « canaux ioniques ». Elles laissent passer spécifiquement certains ions (soit  $\text{Na}^+$ , soit  $\text{K}^+$ ). Lorsque la membrane est perméable à un seul type d'ion, le potentiel se stabilise alors presque instantanément à une valeur spécifiée par l'équation (1.1) de Nernst :

$$E = \frac{kT}{q} \ln\left(\frac{C_e}{C_i}\right) \quad (1.1)$$

Où  $k$  est la constante de Boltzmann,  $T$  est la température,  $q$  est la charge de l'ion considéré,  $C_e$  est la concentration extérieure, et  $C_i$  est la concentration intérieure. On obtient par exemple pour le potassium, plus concentré à l'intérieur, un potentiel d'équilibre de  $-80$  mV, pour le sodium, plus concentré à l'extérieur, un potentiel d'équilibre de  $50$  mV (avec des valeurs typiques pour les concentrations).

Ainsi, au repos, la membrane du neurone est perméable uniquement au potassium, de sorte que son potentiel avoisine le potentiel d'équilibre de cet ion. Le potentiel est amené à varier avec la perméabilité aux différents ions.

### 3.2 POTENTIEL D'ACTION

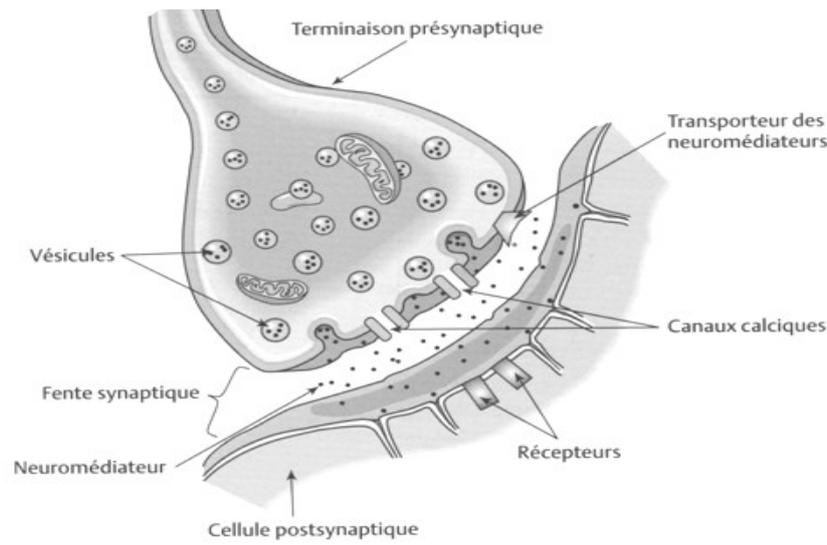
Lorsqu'un stimulus assez faible provenant d'autres neurones arrive au neurone concerné il peut entraîner une augmentation du potentiel dans la cellule. Son potentiel se modifie en proportion du stimulus, cependant, quand le potentiel atteint un seuil, généralement au tour des  $-55$  mV, le neurone envoie une impulsion électrique le long de son axone, on appelle ce phénomène « *une décharge* », elle se manifeste par une montée brutale du potentiel (dépolarisation) jusqu'à une valeur typique de  $40$  mV, suivie d'une redescende rapide (hyperpolarisation), sur le plan physiologique, ceci se traduit par l'ouverture en premier lieu des canaux  $\text{Na}^+$  qui permet au Sodium de pénétrer à l'intérieur de l'axone, on parle alors de dépolarisation de la membrane, sous l'influence de la dépolarisation de la membrane, plusieurs canaux ioniques au  $\text{K}^+$  s'ouvrent et libèrent le Potassium à l'extérieur de la membrane. Toute cette suite de phénomènes est nommée « *un potentiel d'action* ».

### 3.3 AU NIVEAU DE LA SYNAPSE

La terminaison de l'axone est séparée de la cible par un espace intercellulaire de quelques dizaines d'Angstroms ( $10^{-9}$  m). Ce lieu est appelé la « *Synapse* » qui signifie un point de jonction. C'est à ce niveau où s'effectue la communication entre les neurones. Le nombre de synapses par neurone varie aussi considérablement de plusieurs centaines à une dizaine de milliers [Purves et al., 2004].

Quand une décharge atteint une synapse elle cause un changement de potentiel dans les dendrites du neurone cible. On remarque que ce processus est relativement lent, et que l'effet est retardé avec un certain délai propre à cette synapse. Le neurone qui envoie l'impulsion s'appelle le neurone « *pré-synaptique* » et celui qui la reçoit le neurone « *post-synaptique* ». Dans la terminaison synaptique (du neurone pré-synaptique) se trouvent des neurotransmetteurs sous forme de vésicules. Lorsqu'un

potentiel d'action arrive dans la terminaison, il provoque l'ouverture de canaux au calcium  $Ca^{++}$ . Les ions calcium rentrent alors dans le milieu intracellulaire où les neurotransmetteurs sont déversés. Ils vont ensuite se lier à des récepteurs de la membrane du neurone post-synaptique, et ainsi provoquer l'ouverture des canaux ioniques ( $Na^+$ ,  $K^+$ ). Des courants ioniques sont alors produits dans le neurone post-synaptique (voir Figure 1.5).

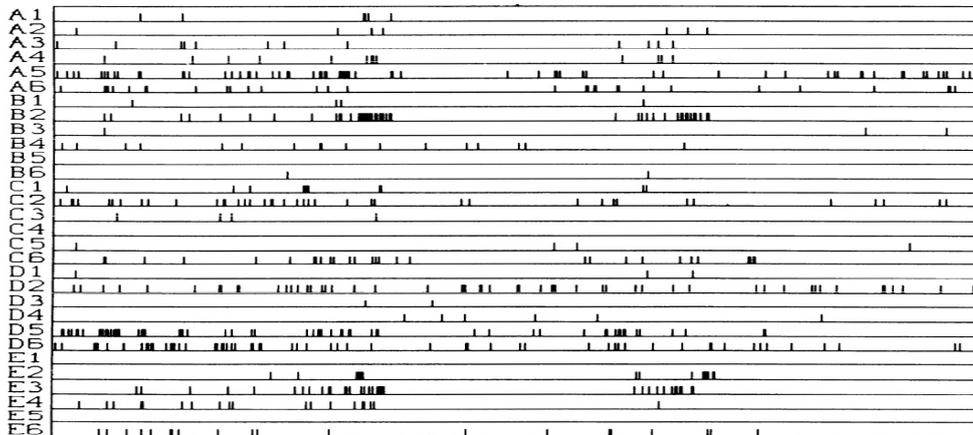


**Figure 1.5** - Arrivée d'un potentiel d'action au niveau de la synapse  
[[http://fichespsycho.acjq.qc.ca/notions\\_pharmaco.html](http://fichespsycho.acjq.qc.ca/notions_pharmaco.html)]

Le changement peut être positif, en augmentant le potentiel post-synaptique des neurones, ou négatif, en l'abaissant. Lorsque le potentiel augmente, la synapse est appelée « *excitatrice* », par contre, si elle abaisse le potentiel post-synaptique des neurones, la synapse est appelée « *inhibitrice* ». L'effet du changement du potentiel est temporaire; après un moment il se dissipera car le neurone essaye toujours de rester à son potentiel de repos. Après qu'un neurone ait émis une impulsion, il a besoin d'un certain intervalle de temps pour récupérer, avant qu'il ne puisse réémettre un signal de nouveau. On appelle ce court instant la « *période réfractaire* ».

## 4 CODAGE DE L'INFLUX NERVEUX

Le fondement des propriétés caractéristiques du neurone impulsionnel est intimement lié à des observations retenues dans le milieu biologiques. Les nouvelles technologies d'analyse et d'imagerie du cerveau ont levé, en partie, le voile sur l'interaction neuronale. Les études effectuées ont surtout mis en spot la nécessité à déchiffrer le langage du cerveau.



**Figure 1.6** - Impulsions (Spikes) émises durant 4s sur 30 neurones du cortex visuel d'un singe. Chaque ' | ' représente un Spike [Kruger et al., 1988]

A titre d'exemple, la Figure 1.6 illustre des enregistrements prélevés pendant 4 secondes sur 30 neurones du cortex visuel d'un singe à qui on a présenté des images qu'il devait reconnaître. De telles expériences démontrent l'existence et la complexité du message de l'influx nerveux. Par conséquent, l'observation de l'influx permettra de prélever tous phénomènes similaires et redondants.

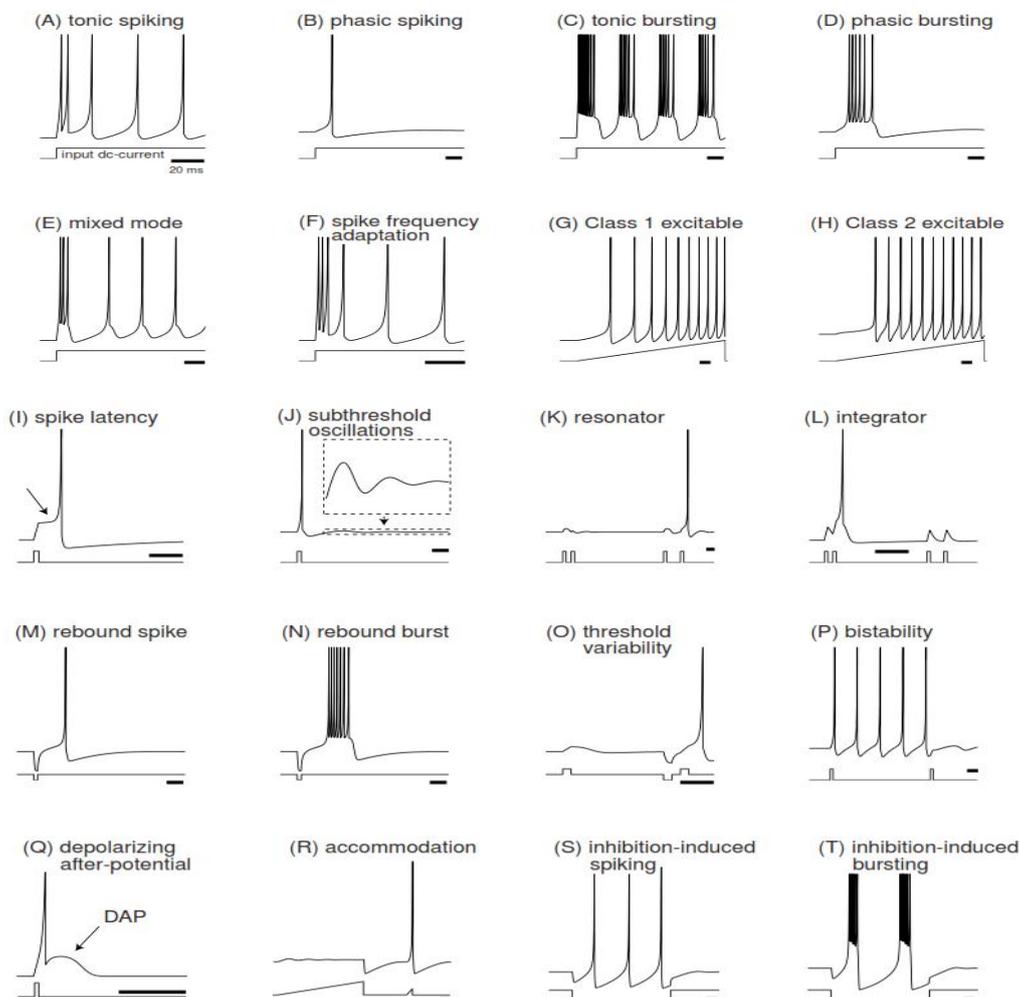
## 5 CARACTERISTIQUES DU CODAGE DE L'INFLUX NERVEUX

Dans la littérature, on distingue d'importantes caractéristiques neurocomputationnelles sur le plan biologique induisant vers une meilleure représentation du codage des stimuli que véhicule un modèle impulsionnel. En effet, les expériences ont montré qu'en exposant une cellule à un courant continu (DC : Direct Current), un neurone est capable de fournir une réponse impulsionnel qui peut se formuler d'au moins vingt différentes façons [Izhikevich, 2004].

La Figure 1.7 montre les différentes réponses impulsionnelles des neurones du cortex, on note les cas suivants [Izhikevich, 2004 ; Berrêdo, 2005 ; Gerstner et al., 2002] :

- a) **Impulsion tonique (Tonic Spiking)** : Dans ce cas-là, le neurone persiste à enchaîner une série d'impulsion tant que le courant en entrée est actif. Ce type de comportement peut être observé dans trois types de neurones du cortex.
- b) **Impulsion périodique (Phasic Spiking)** : le neurone émet une seule impulsion lors de l'activation de l'entrée.
- c) **Rafale tonique (Tonic Bursting)** : le neurone émet périodiquement une série de rafales d'impulsion lors de la stimulation.
- d) **Rafale périodique (Phasic Bursting)** : le neurone émet une seule rafale d'impulsion.

- e) **Mode mixte (Mixed Mode)**: le neurone change de mode pendant la stimulation en passant de l'impulsion tonique aux Rafales périodiques.
- f) **Adaptation de la fréquence de l'impulsion (Spike Frequency Adaptation)**: le neurone émet des impulsions tonique en diminuant la fréquence.
- g) **Excitabilité de classe 1 (Class 1 Excitability)**: Dans ce cas, la fréquence de l'impulsion tonique émise dépend de la puissance du stimulus en entrée, i.e : Si le signal DC est faible, la fréquence de l'impulsion diminuera.
- h) **Excitabilité de classe 2 (Class 2 Excitability)**: Le neurone est incapable d'émettre des trains d'impulsions à basse fréquence et ne peut les émettre qu'à haute fréquence sinon il ne marquera aucune activité.



**Figure 1.7** - Caractéristiques neurocomputationnelles recensées des travaux d'Izhikevich [Izhikevich, 2003]. Les graphes représentent la réponse envoyée par un simple neurone impulsionnel lors d'une injection d'un courant sur 20 ms. On note qu'aucun modèle ne peut renvoyer ces vingt états à la fois.

- i) **Latence d'impulsion (Spike Latency)** : le neurone émet en respectant un certain délai de la puissance du stimulus.
- j) **Oscillations sous-seuil (Subthreshold Oscillations)** : Le neurone exhibe des potentiels oscillatoires.
- k) **Orientation fréquence et résonance (Frequency Preference and Resonance)**: Cet état est dû au phénomène de la résonance, où un neurone à potentiel oscillatoire répond d'une manière sélective à un stimulus ayant une teneur en fréquence similaire à la fréquence d'oscillations sous-seuil.
- l) **Intégration et détection de coïncidence (Integration and Coincidence Detection)** : le neurone agit comme en intégrateur sans la présence d'un potentiel oscillatoire.
- m) **Impulsion de rebondissement (Rebound Spike)** : Le neurone déclenche une impulsion post-inhibitrice (rebond) quand il reçoit un stimulus.
- n) **Rafale de rebondissement (Rebound Burst)** : Le neurone émet une rafale d'impulsions post-inhibiteur.
- o) **Variabilité du seuil (Threshold Variability)** : L'activité antérieure d'un neurone peut provoquer une variabilité de son seuil. En effet, le neurone peut ne pas répondre à une courte impulsion excitatrice précédée par une entrée inhibitrice récente.
- p) **Bi-stabilité du repos et les états d'impulsion (Bi-stability of Resting and Spiking States)** : le neurone définit deux modes stables de fonctionnement : repos et décharge (soit par un tonique ou bien une rafale d'impulsions). Le mode s'échange par une pulsation inhibitrice ou excitatrice.
- q) **Dépolarisation après-potentiels (Depolarizing After-Potentials)** : Le potentiel électrochimique de la membrane d'un neurone peut montrer une prolongation de la phase après-hyperpolarisation (AHP : after-hyperpolarization) ou de la phase de dépolarisation après-potentiel (DAP : depolarized after-potential). Dans ce cas là, le neurone devient super-excitabile puisque la phase réfractaire est réduite.
- r) **Hébergement (Accommodation)** : Face à un puissant stimulus à lente diminution, ce qu'on appelle un « *flat ramp* », le neurone perd sa sensibilité et ne retourne aucune impulsion. Dans ce cas, seul un faible courant mais à une diminution rapide, « *step ramp* », permet au neurone de répondre par une impulsion.

- s) **Impulsion induite par inhibition (Inhibition-Induced Spiking)** : En absence d'une entrée, le neurone reste dans un état de repos, mais émet une impulsion quand il est hyperpolarisé par une entrée inhibitoire par injection de courant.
- t) **Rafale induite par inhibition (Inhibition-Induced Bursting)**: Similaire au cas précédent, sauf que le neurone émet une rafale d'impulsions au lieu de répondre avec un tonique d'impulsion.

En conséquence, le codage de l'information est un autre passage obligatoire imposé à la définition du neurone à spike. L'exposition de ces caractéristiques a beaucoup contribué au traitement de l'information liée au l'instant de l'impulsion (spike-timing) et au codage et encodage de l'information.

## 6 CONCLUSION

Le fonctionnement du cerveau a longtemps inspiré les chercheurs dans l'intelligence artificielle et des technique bio-inspirés. Il est profondément inscrit dans la mémoire commune que la reproduction artificielle du fonctionnement du cerveau humain est fondamentale pour plusieurs raisons. L'hypothèse principale, à base des réseaux de neurones artificiels, est que le comportement intelligent est sous-tendu par un ensemble de mécanismes qui sont restés longtemps méconnus. C'est autours de ces tentatives de modélisation que le neurone impulsionnel a été conçu et continu d'évoluer. Dans le chapitre suivant, on va plus détailler l'évolution du modèle impulsionnel à traves les années.

# CHAPITRE 2

## MODELISATION DU NEURONE IMPULSIONNEL : HISTOIRE D'UNE EVOLUTION

### 1 INTRODUCTION

Le neurone impulsionnel est souvent désigné comme la troisième génération des neurones artificiels. Fortement inspiré de la cellule naturelle, il représente une renaissance des anciens travaux de Hodgkin et Huxley lancés en 1963 et longtemps abandonnés en neurosciences.

Avec son évolution, le « *Spiking neuron* » donne lieu à la définition de plusieurs formes de modèles de calculs originaux permettant tout aussi bien l'étude de propriétés importantes du fonctionnement cérébral par des descriptions mathématiques du potentiel d'action que la réalisation de traitements de données performants [Alexandre, 2005].

Dans le même contexte que le chapitre précédent, on met en lumière l'origine de la modélisation du neurone impulsionnel d'un autre angle. En effet, le neurone biologique est bien la source d'inspiration du « *spiking neuron* » mais sa reconduction dans la neurocomputation est aussi l'œuvre de l'évolution et des ambitions de la communauté des neuroscientifiques à travers le temps. Dans ce chapitre, on commence par démontrer cet inévitable développement à travers un historique spécifiquement ciblé. Ensuite, on expose quelques modèles importants à la description des différentes transitions amenant à l'apparition des différentes générations du neurone artificiel. Finalement, on décrit les différents types de codages d'informations qu'utilisent les réseaux de neurones impulsionnels.

### 2 EVOLUTION DU NEURONE ARTIFICIEL

La conception du neurone artificiel dépend formellement sur la modélisation de son fonctionnement sur le plan conceptuelle et mathématique. Les premières modèles neuronaux ont vu le jour au début des années 40 ; Une époque où l'évolution des

mathématiques était timide et de la neuroscience pratiquement inexistante [Alexandre, 2005].

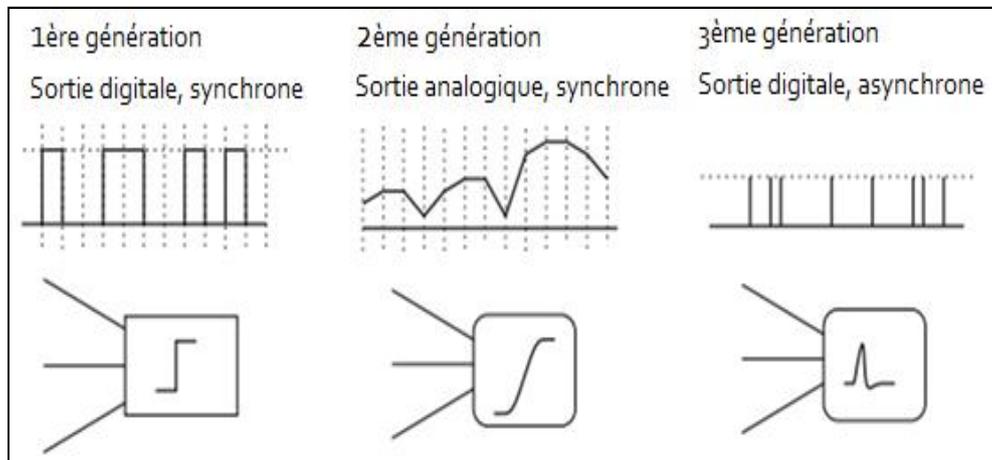


Figure 2.1 - Trois générations de neurones [Rochel, 2004].

Naturellement, l'approche connexionniste a presque failli disparaître, au bout d'une trentaine d'années de son lancement, pour leurs très modestes performances. Une lacune certainement dû à plusieurs facteurs mais qui parallèlement a engendré l'apparition de plusieurs modèles. À travers leurs évolutions, ces modèles ont été regroupés sur trois générations de neurones en se basant sur leurs modes de fonctionnement (voir Figure 2.1).

## 2.1 PREMIERE GENERATION

La première génération de neurones artificiels débute par l'automate à seuil de McCulloch et Pitts en 1943, appelé « neurone formel ». Cette modélisation, conceptuellement très simple, a été assurée par une fonction avec un rendement booléen où l'envoi d'un signal binaire ne dépend que d'une comparaison entre la somme pondérée de ses signaux entrants et d'un seuil fixé aléatoirement à l'avance [Vreeken, 2003]. L'implantation des neurones de la première génération dans une architecture connexionniste fut introduite dans les débuts des années 60, le perceptron reflète parfaitement l'image de cette modélisation. Même si ces neurones ne peuvent donner qu'un rendement binaire, ils ont été appliqués avec succès dans de puissants réseaux de neurones artificiels tel que les perceptrons multicouches, la machine de Boltzmann et le réseau Hopfield.

## 2.2 DEUXIEME GENERATION

Dans la deuxième génération, les neurones n'emploient pas une fonction à seuil pour retourner leurs signaux de sortie, mais une fonction d'activation continue, les rendant

adaptés aux entrées et sorties analogues. Les fonctions d'activation généralement utilisées dans cette catégorie de neurones sont la sigmoïde et la tangente hyperbolique [Maass et al., 1991] et on les retrouve dans des architectures de types feedforward ou encore dans les réseaux récurrents. Ils sont plus puissants que leurs prédécesseurs de première génération [DasGupta et al., 1992]. En outre, ils peuvent bien approximer n'importe quelle fonction analogue de façon arbitraire.

Les neurones des deux premières générations n'emploient pas d'impulsion individuelle, mais des signaux de sorties typiquement entre 0 et 1, ces signaux peuvent être vus comme une normalisation de la fréquence pour un neurone durant une certaine période de temps. C'est ce qu'on appelle « *codage par cadence* », où une cadence élevée correspond à un signal plus élevé.

### 2.3 TROISIEME GENERATION

Dans les deux précédentes générations, les modèles étaient des automates dont l'état était une métaphore du taux de décharge d'un neurone biologique. Un neurone recevant la stimulation d'autres neurones recalcule son état, c'est-à-dire une valeur numérique représentant sa fréquence de décharge, en fonction de la force des stimulations reçues [Vreeken, 2003].

La dernière génération élève de nouveau le niveau du réalisme biologique en se basant sur les impulsions individuelles [Maass, 1997]. Ceci permet d'incorporer l'information spatiotemporelle dans la communication, comme le font les neurones biologiques. Ainsi au lieu d'employer le codage par cadence, ces neurones utilisent le codage par impulsion ; mécanismes où les neurones reçoivent et envoient différentes impulsions. Des découvertes récentes dans le domaine de la neurologie ont prouvé que les neurones dans le cortex exécutent des calculs analogues à une vitesse incroyable [Gerstner et al., 1999].

Dans la prochaine section, on va décrire quelques modèles de cette troisième génération de neurone.

## 3 MODÈLES DE NEURONES IMPULSIONNELS

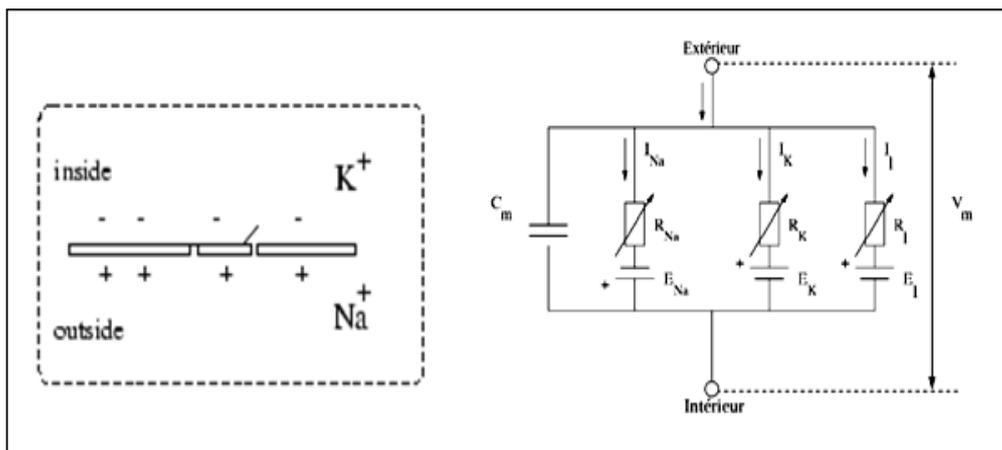
Le fonctionnement du système nerveux a toujours été une source d'inspiration pour concevoir des machines et des modèles de traitement de l'information. Ainsi, La construction du neurone artificiel dépend formellement de la modélisation de son fonctionnement sur le plan conceptuelle et mathématique. Dans cette section, on présentera quelques modèles éminents de neurones impulsionnels suivant leurs trois différentes catégories [Gerstner et al., 2002].

### 3.1 MODELES A BASE DE CONDUCTANCE

Dans cette catégorie, on retrouve le modèle de « Hodgkin & Huxley » le plus ancien prototype des neurones impulsionnels. Autour de ce modèle, d'autres variantes ont vu le jour.

#### 3.1.1 MODÈLE HODGKIN-HUXLEY

En se basant sur l'observation de l'activité électrique de l'axone géant du calmar, Hodgkin et Huxley (HH) ont proposé le premier modèle neuronal détaillé en 1952 [Hodgkin et al., 1952]. Ce modèle s'est construit en simulant la dynamique neuronal à base d'une analyse des courants ioniques, tout en montrant que ces courants représentent des caractéristiques importantes des neurones, de la génération des potentiels d'action et la période réfractaire absolue (voir Figure 2.2) [Giulioni, 2008].



**Figure 2.2** - Schéma électrique représentant les flux ioniques au travers de la membrane d'un neurone selon le modèle de Hodgkin-Huxley.  $C$  est la capacité membranaire,  $R$  la résistance,  $V$  le potentiel de membrane,  $E$  le potentiel d'inversion et  $I$  le courant injecté.

Le modèle HH retourne une description quantitative de l'électrophysiologie du neurone biologique. Ainsi, il est très difficile à modéliser sur le plan mathématique en raison de sa nature non-linéaire très élevée. La formulation de ce modèle est répartie sur quatre équations différentielles couplées reproduisant la dynamique du potentiel de membrane  $V_M$  du neurone. En revenant sur l'aspect biologique (voir Figure 1.3), on peut mieux cerner comment se construit cette modélisation. Le  $V_M$  est la fonction du courant d'entrée  $I$  appliqué au neurone lorsque celui-ci est stimulé, des courants  $I_k$  et  $I_{Na}$  générés par la circulation d'ions  $K^+$  et  $Na^+$  au travers de la membrane et d'un courant de fuite  $I_l$  représentant notamment les mouvements d'ions  $Cl^-$  [Mouraud, 2009].

Chacun de ces courants est en fonction de l'écart entre le potentiel de membrane  $V_M$  et les potentiels d'inversion  $E_{Na}$ ,  $E_K$  et  $E_l$  (voir Figure 2.2). Ainsi, le potentiel de membrane du neurone est reproduit par l'équation (2.1).

$$C_M \frac{dV_M}{dt} = -g_{Na}m^3h(V_M - E_{Na}) - g_Kn^4(V_M - E_K) - g_l(V_M - E_l) + I \quad (2.1)$$

Sachant que le  $i$  varie entre les valeurs  $K$ ,  $Na$  et  $l$ , les  $g_i$  sont des constantes et les paramètres  $h$ ,  $m$  et  $n$  décrivent les probabilités d'ouverture/fermeture des canaux ioniques : sodium pour  $h$  et  $m$ , potassium pour  $n$ .

Dans le même contexte, pour chacun de ces paramètres on dispose de l'équation différentielle (2.2) qui décrit les paramètres  $h$ ,  $m$  et  $n$  où il suffit de remplacer le  $x$  par  $h$ ,  $m$  et  $n$ .

$$\frac{dx}{dt} = \alpha_x(1 - x) - \beta_x x \quad (2.2)$$

Par ailleurs, les paramètres  $\alpha_x$  et  $\beta_x$  correspondent respectivement aux probabilités d'entrée de particules à l'intérieur de la membrane et de sortie de particules vers le milieu extérieur et ne dépendent pas directement du temps, mais dépendent du potentiel de la membrane selon les équations suivantes :

$$\alpha_n = 0.01(V_M + 10)/(e^{(V_M+10)/10} - 1) \quad (2.3)$$

$$\beta_n = 0.125e^{V_M/80} \quad (2.4)$$

$$\alpha_h = 0.01(V_M + 25)/(e^{(V_M+25)/10} - 1) \quad (2.5)$$

$$\beta_m = 4e^{V_M/18} \quad (2.6)$$

$$\alpha_h = 0.07e^{V_M/20} \quad (2.7)$$

$$\beta_h = 1/(e^{(V_M+30)/10} + 1) \quad (2.8)$$

Ainsi, à partir de ce modèle, de nombreux comportements neuronaux peuvent être reproduits. Le gros inconvénient de ce modèle est sa complexité. En effet, le couplage des quatre équations différentielles (2.1) et (2.2) avec  $x \in \{h, m, n\}$  rend extrêmement lourde la construction de réseaux à partir de tels modèles de neurones.

### 3.1.2 MODELE FITZHUGH-NAGUMO

Le modèle FitzHugh-Nagumo est une simplification du modèle de Hodgkin-Huxley où l'ensemble des quatre équations différentielles est réduit à un système de deux équations (2.9) et (2.10).

$$\frac{dV_M}{dt} = V_M - V_M^3 - \omega + I \quad (2.9)$$

$$\frac{d\omega}{dt} = V_M - \alpha - \beta\omega \quad (2.10)$$

Où  $\omega$ , variable homogène à une tension,  $I$  un courant de stimulation extérieur,  $\alpha$  et  $\beta$  des constantes à déterminer [Buhry, 2010].

L'avantage de réduire le nombre des équations à deux équations une cubique et l'autre linéaire est de lui permettre une phase systématique.

### 3.1.3 MODÈLE MORRIS-LECAR

Ce modèle a été proposé par Morris and Lecar [Morris et al., 1981]. Sa particularité est d'avoir réussi à combiner le formalisme original d'Hodgkin et Huxley et la réduction des paramètres qu'offre le modèle de FitzHugh-Nagumo.

Deux expressions représentent ce modèle où l'équation (2.11) illustre  $V_M$  le potentiel de la membrane et (2.12) pour la variable  $n$  de la décharge du potassium [Berrêdo, 2005].

$$C \frac{dV_M}{dt} = I - g_{Ca} m_\infty(V_M)(V_M - E_{Ca}) - g_K n(V_M - E_K) - g_L(V_M - E_L) \quad (2.11)$$

$$\frac{dn}{dt} = \phi \frac{[n_\infty(V_M) - n]}{\tau_n(V_M)} \quad (2.12)$$

Où les états stables pour le  $Ca^2$  et  $K$  et la constante du temps  $\tau_n$  sont :

$$m_\infty = \frac{1}{1 + \exp[-(u - u_1)/u_2]} \quad (2.13)$$

$$n_\infty = \frac{1}{1 + \exp[-(u - u_3)/u_4]} \quad (2.14)$$

$$\tau_n(u) = \frac{1}{\cosh\left(\frac{u - u_2}{2u_4}\right)} \quad (2.15)$$

Les paramétrés  $m_\infty$  et  $n_\infty$  décrites dans les équations (2.13) et (2.14) sont formellement identiques.

### 3.1.4 MODÈLE HINDMARSH-ROSE

Le modèle Hindmarsh-Rose [Hindmarsh et al., 1982 ; Hindmarsh et al., 1984] s'inspire fortement du modèle FitzHugh-Nagumo. Il est contrôlé par les trois équations différentielles suivantes :

$$\frac{du}{dt} = v - F(u) + I - \omega \quad (2.16)$$

$$\frac{dv}{dt} = G(u) - v \quad (2.17)$$

$$\frac{d\omega}{dt} = \frac{[H(n) - \omega]}{\tau} \quad (2.18)$$

Où  $F, G$  et  $H$  sont des fonctions spécifiques qui doivent être choisis correctement pour produire les résultats désirés.

Ainsi, ce modèle simple est capable de reproduire des comportements dynamiques variés sans nécessiter beaucoup de réglages.

### 3.1.5 MODELE IZHIVIT

Les travaux d'Izhikevich [Izhikevich, 2003] ont réussi à proposer l'un des modèles impulsifs les plus engagés à reproduire le comportement biologique sur le plan expérimental. Ce modèle a réussi à regrouper simplicité et capacité de générer les motifs de tir généralement observés dans les neurones biologiques et décrit dans la Section 1.5.

Inspiré du modèle de Fitzhugh-Nagumo proposé entre 1961 et 1962, ce modèle repose uniquement sur deux équations différentielles couplées décrites par les équations :

$$\frac{dV_M}{dt} = 0.04V_M^2 + 5V_M + 140 - u + I \quad (2.19)$$

$$\frac{du}{dt} = a(bV_M - u) \quad (2.20)$$

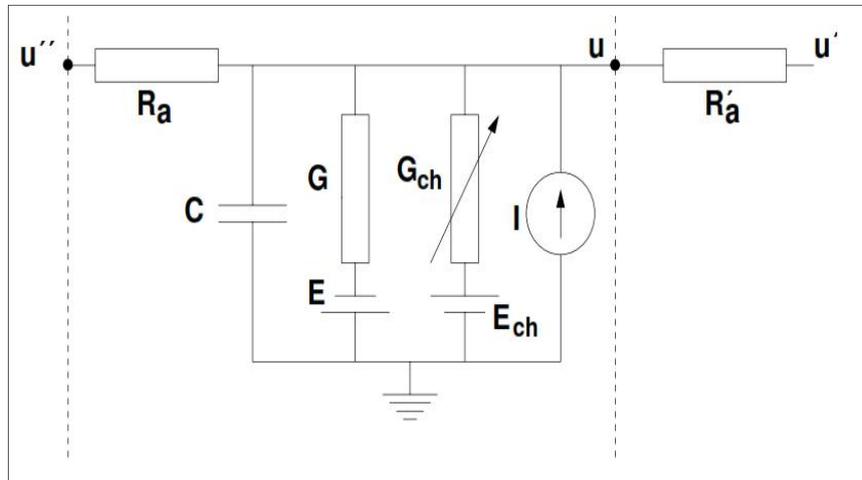
$$\text{Si } V_M \geq 30\text{mV, alors } \begin{cases} V_M \leftarrow c \\ u \leftarrow u + d \end{cases} \quad (2.21)$$

Où  $V_M$  représente le potentiel de membrane du neurone et  $u$  représente une membrane variable de récupération, qui représente l'activation de courant ionique  $K^+$  et l'inactivation de courant ionique  $Na^+$ . Après que le pic atteint sa limite supérieure (+30mV), la membrane et la variable de tension de rétablissement sont remis à zéro en fonction de l'équation (2.21). Le courant synaptique ou le courant continu injecté sont fournis via la variable  $I$ . En faisant varier les quatre paramètres  $a, b, c$  et  $d$  on peut obtenir la plupart des réponses neuronales connues, ce qui rend ce modèle particulièrement intéressant. De plus, malgré le couplage de ses deux équations différentielles, ce modèle reste relativement simple en le comparant à des modèles de type HH et il est particulièrement adapté aux simulations de réseaux de neurones impulsifs [Cauwenberghs et al., 2009].

## 3.2 MODELE COMPORTEMENTAL

En cherchant à projeter la différence entre les modèles comportementaux et les modèles à conductance, il est important de savoir que seuls les courants engendrés par le  $Na$  et  $K$  sont modélisés dans le modèle HH. Cependant, beaucoup d'ouvrages parlent de l'intervention d'autres ions, tels que  $Ca^{+2}$  et  $Ca$ -médiateur  $K$ , dans l'adaptation neuronale. Par ailleurs, la complexité des modèles biologiques est mieux exprimée dans les unités comportementales où la structure spatiale des arbres dendritiques est

prise en considération, ainsi que la modélisation détaillée de la transmission synaptique.



**Figure 2.3** - Schéma d'un circuit générique équivalent à un modèle impulsionnel comportementale [Bower et al., 2003]

La particularité des unités comportementales est d'utiliser deux petits composants uniformes. Chacun des composants est modélisé par une équation différentielle décrivant un circuit électrique équivalent capable non seulement de simuler le comportement biologique mais aussi les interactions entre les différents composants.

La Figure 2.3 illustre un circuit équivalent où le  $V_M$  est le potentiel de la membrane qui apparaît à travers  $C$  le condensateur de la membrane. Le courant  $I$  est représenté par une entrée extérieure. L'équation différentielle résultante est la suivante:

$$C \frac{dV_M}{dt} = (E - V_M)G + \sum_{ch}(E_{ch} - V_M)G_{ch} + \frac{(V_M V_M)}{R_a} + \frac{(V_M'' V_M)}{R_a} + I \quad (2.22)$$

Cependant, L'équation (2.22) ne décrit qu'un simple modèle comportemental de type linéaire. Il existe d'autres modèles non-linéaires beaucoup plus complexes avec des condensateurs pour des courants ioniques variant.

### 3.3 MODELES A BASE DE SEUILLAGE ET TIRE

Les modèles à base de seuillage et tire ont la particularité de réaliser au niveau du potentiel de la membrane  $u(t)$  une sommation temporelle de toutes les contributions reçues des neurones présynaptiques (PSP). Ces contributions induisent à produire un tire si leurs accumulations dépassent un seuil  $\vartheta$ .

Cette dernière catégorie de neurones impulsionnels nous intéresse tout particulièrement, puisqu'elle est l'objet d'étude de notre thèse.

### 3.3.1 MODELE INTEGRATE & FIRE (IF)

Dans ces travaux de 1907, Lapique avait proposé le modèle « *Lapicque* » ne tenant compte que du potentiel de membrane  $V_M$  pour représenter l'activité du neurone. Ces travaux furent repris plus tard par [Knight, 1972 ; Abbott, 1999] pour lui donner le nom de neurone IF « *Intègre et Tire* » en anglais: Integrate and Fire [Mouraud, 2009].

Comme la plupart des modèles intègre-et-tire, il se base sur les hypothèses suivantes [Dubois, 2006]: le neurone est ponctuel et le PA est un événement stéréotypé donc seul le temps d'émission des P.A. compte.

Le modèle Integrate & Fire (IF) est un modèle simple utilisant le principe du seuil d'émission. Dans son expression la plus simple, l'évolution du potentiel (sous le seuil) est donnée par l'équation différentielle suivante [Rochel, 2004 ; Buhry, 2010]:

$$C \frac{dV_M}{dt} = I(t) \quad (2.23)$$

Où C est la capacité membranaire et le courant I(t) modélise les entrées du neurone.

Ce modèle se généralisera plus tard sous une formalisation plus communément généralisé et connu sous le nom de leaky integrate and fire (LIF) qu'on détaillera dans la section suivante où on y ajoute un terme de fuite :

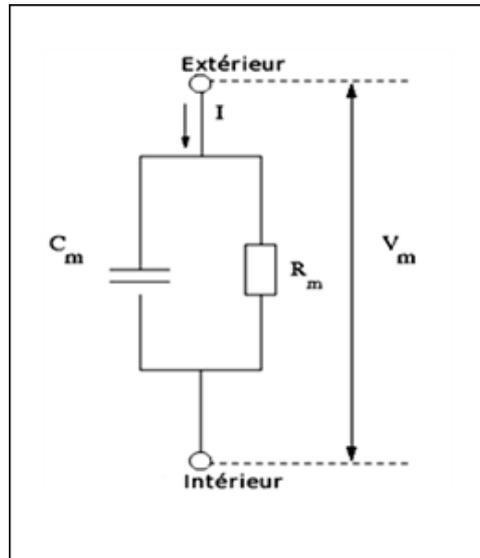
$$C \frac{dV}{dt} = -\frac{V}{R} + I(t) \quad (2.24)$$

Où R est la résistance associée à la fuite.

L'intérêt du neurone intègre-et-tire, outre sa simplicité, est de capturer deux éléments essentiels du comportement commun à de nombreux neurones biologique.

### 3.3.2 MODÈLE DE LEAKY-INTEGRATE-AND-FIRE (LIF)

Dans ce modèle, la membrane est modélisée par un condensateur avec un courant de fuite [Gerstner et al., 2002]. Cela revient à modéliser la membrane comme un circuit RC c'est-à-dire un condensateur monté en parallèle avec une résistance (exprimée par sa conductance G) et un générateur (voir Figure 2.4). Le potentiel du générateur est pris égal au potentiel de repos  $V_{rest}$ .



**Figure 2.4** - Schéma électrique du neurone LIF (Intègre et Tire à fuite). Un condensateur de capacité  $C_m$  est branché en parallèle avec une résistance.

L'idée générale de ce modèle vient du fait que le potentiel d'action du neurone peut être caractérisé par la présence de deux phases: (1) une dépolarisation lente du potentiel d'action de la membrane jusqu'à ce qu'il atteigne un "seuil", et (2) une dépolarisation rapide suivie par une repolarisation revenant au repos [Schwemmer, 2010]. Cette deuxième phase offre souvent une forme invariante qui est déterminée par l'interaction non-linéaire des ions  $Na^+$  et  $K^+$ . Typiquement, cette phase n'est pas affectée par les variations de courant. Ainsi, pour simplifier la phase (2), on suppose que les courants non-linéaires peuvent être remplacés par un seuil non-linéaire accompagné d'une condition de réinitialisation. Finalement, on part de l'hypothèse que la dynamique sous seuillage (phase (i)) est contrôlée par le courant de fuite.

Ainsi, le potentiel est décrit par l'équation différentielle [Dubois, 2006]:

$$C_m \frac{dV}{dt} = -G_L(V - V_{rest}) + I(t) \quad (2.25)$$

$$V(t) = V(t_0)e^{G_L/C_m(t-t_0)} + \left[ V_{rest} + \frac{I(t)}{G_L} \right] (1 - e^{G_L/C_m(t-t_0)}) \quad (2.26)$$

Où le paramètre  $V$  est le potentiel de membrane,  $C_m$  est la capacité de la membrane,  $G_L$  est la conductance de fuite (ou leak en anglais),  $V_{rest}$  est le potentiel de repos et  $I(t)$  est le courant injecté dans la membrane à l'instant  $t$  et  $t_0$  est l'instant initial.

Souvent, on associe souvent  $\left(\frac{dV}{dt} = 0\right)$  à l'équilibre, on a  $V = V_{rest}$ . Un tel circuit est caractérisé par une constante de temps.

$$\tau = R_L C = \frac{C}{G_L} \quad (2.27)$$

Avec ceci, l'équation s'écrit :

$$\tau \frac{dV}{dt} = -(V - V_{rest}) \quad (2.28)$$

La raison majeure de la popularité du modèle Leaky integrate-and-fire est qu'il contient seulement un seuil non-linéaire. Ainsi, on peut souvent modéliser une solution analytique. Le modèle LIF peut également présenter un comportement oscillatoire observé dans les modèles de type HH. Cependant, il est important de noter que le modèle LIF ne prend pas toutes les caractéristiques du modèle HH. Par exemple, dans sa modélisation, on a beaucoup exagéré dans la simplification de la variance du seuil et les propriétés réfractaires irréalistes [Azouz et al., 1999 ; Chacron et al., 2003].

### 3.3.3 MODÈLE QUADRATIC-INTEGRATE-AND-FIRE (QIF)

En 2003, l'équipe de Fourcaud-Trocmé a proposé le modèle Quadratic Integrate and Fire [Fourcaud-Trocmé et al., 2003]. La particularité de ce modèle est d'avoir induit une non-linéarité à travers un paramètre ajouté au potentiel de la membrane introduit grâce à un terme du deuxième degré [Mouraud, 2009].

$$C_M \frac{dV_M}{dt} = -g_l(V_M - E_l) + \Psi(V_M) + I(t) \quad (2.29)$$

$$\Psi(V_M) = \frac{g_l}{2\Delta_T}(V_M - V_T)^2 + g_l(V_M - E_l) - I_T \quad (2.30)$$

Où  $\Psi(V_M)$  est le terme quadratique ajouté à l'équation différentielle linéaire de la dynamique sous le seuil du neurone LIF.  $V_T$  correspond au potentiel seuil de déclenchement d'un potentiel d'action et  $I_T$  est le courant seuil correspondant. Le paramètre  $\Delta_T$  représente la pente du potentiel d'action et  $g$  le coefficient de conductance.

Une variante de ce modèle nommée Exponential Integrate and Fire (EIF) consiste à utiliser une exponentielle plutôt qu'une équation de second degré dans la fonction  $\Psi(V_M)$ , l'équation (2.30) est remplacée par :

$$\Psi(V_M) = g_l \Delta_T e^{(V_M - V_T)/\Delta_T} \quad (2.31)$$

Les travaux de [Wang et al., 1996] démontre que les modèles de neurones LIF, EIF, QIF arrivent à générer un potentiel d'action (PA) équivalent à celui généré par le modèle HH. Le modèle QIF ainsi défini permet de réintroduire la forme du PA dans la dynamique du neurone.

Cependant, la comparaison avec la dynamique du modèle HH montre que l'introduction du terme quadratique ne permet qu'une approximation de la forme du PA. En moyenne la forme du PA est stéréotypée et le PA a une durée de l'ordre de la

milliseconde, ce qui justifie l'approximation faite à ce niveau par le modèle LIF qui n'a qu'un faible impact sur la dynamique, à des échelles de temps supérieures à la milliseconde. Bien que l'utilité des QIF ne soit plus à prouver, notamment en tant qu'oscillateurs (e.g [Ermentrout, 2005]), de nombreux comportements neuronaux ne peuvent être reproduits avec ce type de modélisations [Mouraud, 2009].

### 3.3.4 MODÈLE À CONDUCTANCE INTEGRATE & FIRE (GIF)

Proposé par [Rudolph et al., 2006], il s'agit d'une classe de modèles qui apporte une amélioration au modèle LIF en basant les calculs de la dynamique interne du neurone sur sa conductance membranaire. Ce modèle est ainsi sujet à l'expérimentation de mécanismes d'adaptations, supervisées ou non supervisées [Martinez, 2006].

Le modèle intègre-et-tire basé sur les conductances prend cela en compte. Ainsi, les entrées synaptiques sont modélisées par des conductances variables dans le temps. On utilise deux types d'entrées synaptiques: les entrées excitatrices et les entrées inhibitrices. Par analogie avec le potentiel de d'équilibre d'un ion, on introduit le potentiel de réversion des synapses qui est le potentiel vers lequel cette synapse fait tendre le neurone. La *Figure 2.5* donne le schéma électrique équivalent [Dubois, 2006].

Le modèle comporte trois conductances :

- La conductance de fuite  $\frac{1}{\tau_m^L}$  à laquelle est associée le potentiel de repos  $E_L = V_{rest}$  qui correspond la constante de temps du modèle L.I.F.
- La conductance excitatrice  $\frac{1}{\tau_m^E(t)}$  à laquelle est associée le potentiel  $E_E$ .
- La conductance inhibitrice  $\frac{1}{\tau_m^I(t)}$  à laquelle est associée le potentiel  $E_I$ .

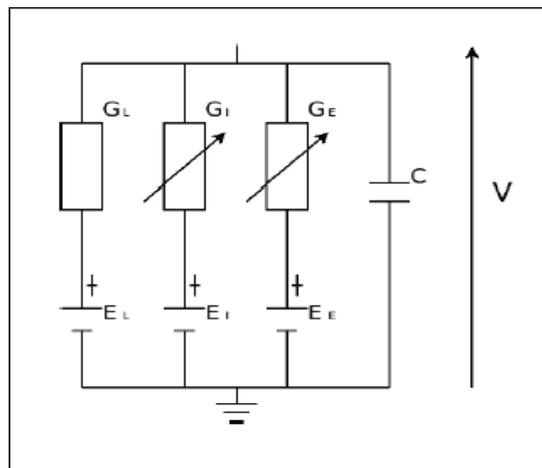


Figure 2.5 - Schéma électrique du modèle g.I.F.

Le potentiel  $V$  est régi par l'équation différentielle suivante :

$$\frac{dV}{dt} = -\frac{1}{\tau_m^L}(V - V_{rest}) - \frac{1}{\tau_m^E(t)}(V - E_E) - \frac{1}{\tau_m^I(t)}(V - V_I) \quad (2.32)$$

L'intérêt de ce modèle est qu'il est relativement simple mais capture bien la réalité. Le potentiel prévu par le modèle est proche de celui prévu par un modèle de type Hodgkin-Huxley [Dubois, 2006].

### 3.3.5 MODELE A REPOSE IMPULSIONNELLE (SRM : SPIKE RESPONSE MODEL)

Deux différences majeures distinguent les modèles précédents du modèle SRM (pour *Spike Response Model*) décrit par [Gerstner et al., 2002] qui trouve ses origines dans [Gerstner et al., 1992]. Tout d'abord, les paramètres influençant le potentiel de membrane ne sont plus des fonctions de ce potentiel, ils dépendent de l'instant  $\hat{t}_i$  du dernier potentiel d'action émis par ce neurone. De plus, le potentiel de membrane n'est plus obtenu à l'aide d'équations différentielles ; il s'agit désormais d'une intégrale tenant compte du passé du neurone pour décrire l'évolution de son potentiel de membrane. Le potentiel de membrane  $u_i$  d'un neurone  $n_i$  est alors donné par l'équation :

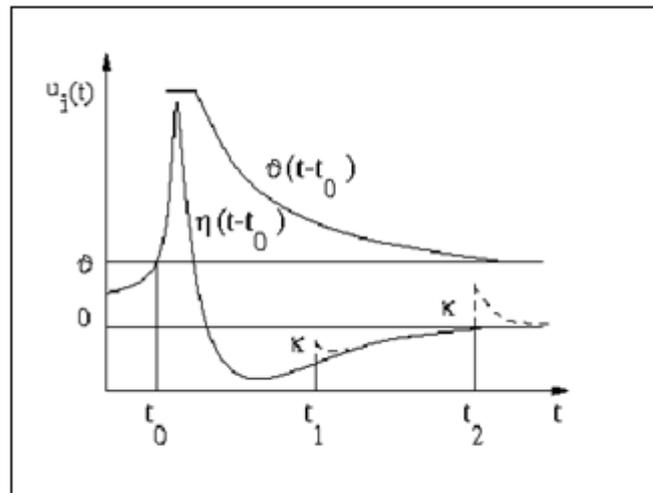
$$u_i = \eta(t - \hat{t}_i) + \sum_j w_{ij} \sum_{t_j^{(f)}} \epsilon_{ij}(t - \hat{t}_i, t - t_j^{(f)}) + \int_0^\infty K(t - \hat{t}_i, s) I^{ext}(t - s) ds \quad (2.33)$$

Un potentiel d'action est émis par le neurone  $n_i$  à la date  $t$  lorsque :

$$u_i(t) = \vartheta(t - \hat{t}_i) \quad \text{avec} \quad \frac{du_i(t)}{dt} > 0 \quad (2.34)$$

Le terme  $\eta(t - \hat{t}_i)$  décrit le comportement du neurone lors de l'émission d'un potentiel d'action et durant la période réfractaire qui suit.  $\hat{t}_i$  est l'instant du dernier potentiel d'action émis par le neurone  $n_i$ .  $w_{ij}$  est le poids de la synapse entre un neurone pré-synaptique  $n_j$  et le neurone post-synaptique  $n_i$ . L'impact de l'arrivée d'un potentiel d'action d'un neurone  $n_j$  sur le neurone  $n_i$  est décrit par le terme  $\epsilon_{ij}(t - \hat{t}_i, t - t_j^{(f)})$  et dépend de la date du dernier potentiel d'action post-synaptique  $\hat{t}_i$  et de celle d'un ou plusieurs potentiels d'action pré-synaptiques  $t_j^{(f)}$ . Dans l'équation (2.33) la double somme est effectuée sur tous les neurones pré-synaptiques et sur tous leurs potentiels d'action passés. Le dernier terme correspond à l'impact d'un éventuel courant extérieur appliqué au neurone,  $\kappa$  décrit la forme d'une impulsion appliquée au neurone. Le seuil  $\vartheta$  au-dessus duquel un potentiel d'action est émis peut également être variable et dépendre de  $\hat{t}_i$  ; il s'exprime alors selon le terme  $\vartheta(t - \hat{t}_i)$ .

La dépendance temporelle du seuil peut être évitée en fixant un seuil constant et en effectuant un changement de variable dans le terme  $\eta$  [Chevallier, 2009].



**Figure 2.6** - Comportement du modèle SRM. Le neurone  $n_i$  émet un potentiel d'action (PA) à  $t = t_0 = \hat{t}_i$ .  $\eta(t - \hat{t}_i)$  décrit l'évolution du potentiel de membrane après un PA.  $\kappa$  décrit l'impact d'un potentiel post-synaptique (PPS) sur le potentiel de membrane. Cet impact dépend du temps écoulé depuis le dernier PA ( $t - \hat{t}_i$ ).  $\vartheta$  décrit l'évolution du seuil après l'émission d'un PA.

La Figure 2.6 fournit un exemple d'évolution du potentiel de membrane d'un neurone  $n_i$  que l'on peut obtenir grâce au modèle SRM. Le modèle ainsi défini permet d'obtenir de nombreux comportements neuronaux. Le modèle LIF est alors un cas particulier du modèle SRM. Le modèle de Hodgkin et Huxley peut être reproduit avec une précision remarquable. Cependant, le modèle SRM reste complexe dans sa forme complète du fait qu'il est nécessaire de prendre en compte toutes les impulsions précédemment reçues par le neurone (voir équation 2.33).

Les auteurs proposent alors une variante beaucoup plus légère : le SRM<sub>0</sub>. Du fait de sa simplicité pour le calcul du potentiel de membrane ( $u_i$ ), le modèle SRM<sub>0</sub> est un très bon candidat pour les simulations de grands réseaux. Il permet, comme le modèle SRM, d'approcher le modèle LIF. Le modèle SRM définit en fait le cadre d'un modèle de neurone. Ce sont surtout les fonctions choisies pour la forme du potentiel d'action, du courant d'entrée et des réponses synaptiques qui détermineront la complexité réelle du modèle [Chevallier, 2009].

## 4 CODAGE DE L'INFORMATION

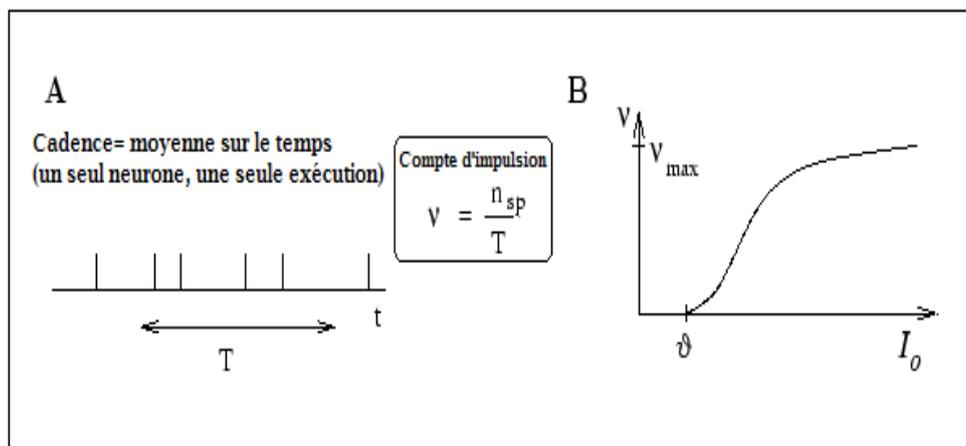
Pour la diffusion des informations entre les neurones, ces derniers doivent employer l'information spatiale et temporelle des impulsions entrantes pour coder les messages transmis. Il existe deux types de codages : codages par cadence et le codage par impulsion.

## 4.1 CODAGES PAR CADENCE

Dans la littérature, on recense trois différents concepts de codages par cadence, qui sont généralement confondues et utilisés simultanément. Ces définitions se rapportent à trois différentes procédures basées sur la moyenne : une moyenne sur le temps, une autre sur plusieurs itérations de l'expérience et une troisième sur une population de neurones sélectionnés [Bendaoua et al., 2013].

### 4.1.1 CADENCE PAR LE COMPTE DES IMPULSIONS

Ce type de codage a mené à l'idée qu'un neurone transforme des informations d'une variable à entrée unique (la force du stimulus) en une variable à sortie unique continue (la cadence de tir). Son concept repose sur un principe très simple. Il suffit de calculer le nombre d'impulsions dans un intervalle de temps  $T$  ensuite de le diviser par la grandeur de cet intervalle en question (*voir Figure 2.7*) [Loiselle, 2004].



**Figure 2.7 - A :** Codage par cadence par utilisant la moyenne temporelle. **B :** Fonction de gain, schématique. La cadence de sortie  $v$  est donnée en fonction de la puissance d'entrée totale  $I_0$ .

La longueur  $T$  de la fenêtre de temps est initialisée par l'expérimentateur et dépend du type du neurone enregistré et du stimulus. Dans la pratique, pour obtenir les moyennes sensibles, plusieurs pics doivent avoir lieu dans la fenêtre de temps. Les valeurs typiques sont:  $T = 100$  ms ou  $T = 500$  ms, mais la durée peut être plus longue ou plus courte [Gerstner et Kistler, 2002].

### 4.1.2 CADENCE PAR LA DENSITE DES IMPULSIONS

Ce type de codage ne concerne que des stimuli stationnaires ou dépendants du temps. Le principe repose sur l'enregistrement des impulsions générées par les neurones en

simulant ceux-ci à l'aide d'une séquence d'entrée. Ensuite, on archive sur un histogramme les résultats observés par une introduction répétitive de cette séquence.

Pour chaque intervalle de temps court avant, pendant et après la simulation, on doit compter le nombre de fois qu'une impulsion est produite et les additionnent sur toutes les répétitions de l'expérience. Ensuite, le nombre d'occurrences des impulsions est divisé par le nombre de répétitions et par l'intervalle de l'histogramme pour obtenir la densité pour cet intervalle (voir Figure 2.8) [Loiselle, 2004].

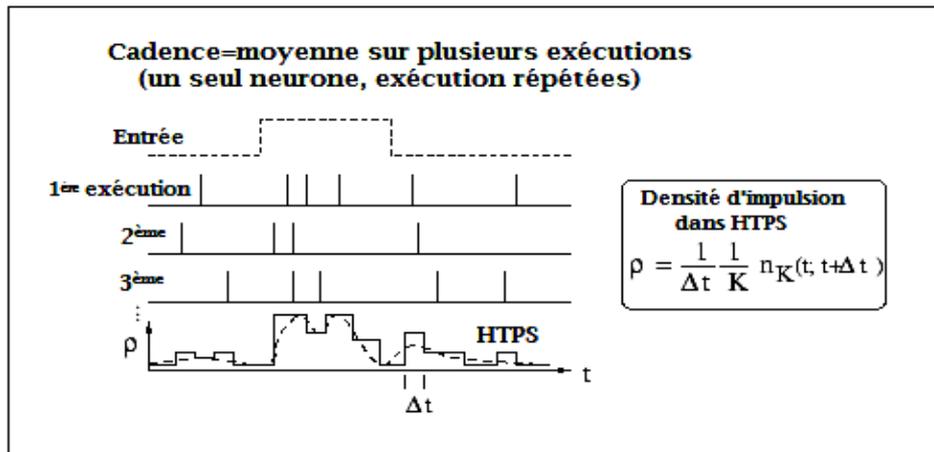


Figure 2.8 - Code par cadence utilisant la densité des impulsions.

### 4.1.3 CADENCE PAR L'ACTIVITE D'UNE POPULATION

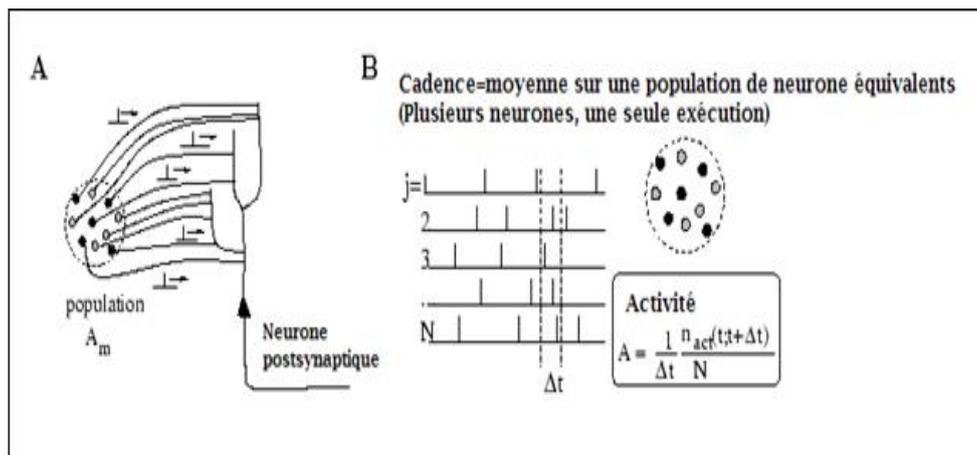


Figure 2.9 - Code par cadence utilisant l'activité d'une population : A. Un neurone postsynaptique reçoit les impulsions en entrée depuis la population  $m$  dont l'activité est  $A_m$ . L'activité de la population est définie par la fraction des neurones qui sont actifs dans un intervalle réduit  $[t, t+\Delta t]$  divisé part  $\Delta t$ .

En observant les neurones, on peut remarquer que des groupes de neurone peuvent posséder les mêmes propriétés et des réponses aux stimuli similaires. Il est alors possible de définir la cadence des impulsions à l'aide d'un ensemble (une population) de neurones [Loiselle, 2004].

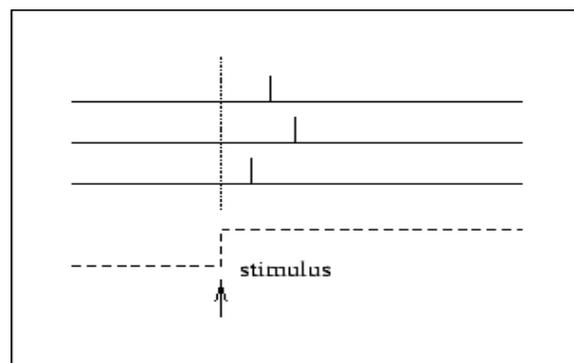
La majeure partie du problème qui se pose dans ce type de codage est la nécessité d'avoir une population homogène de neurones avec des raccordements identiques, ce qui est très peu réaliste. Les populations ainsi sélectionnées auront toutefois un certain degré d'hétérogénéité dans leurs paramètres internes et dans leurs modèles de connectivité (voir Figure 2.9) [Gerstner et al., 2002].

## 4.2 CODAGES PAR IMPULSIONS

Dans la mesure où les réseaux de neurones impulsionnels utilisent des informations temporelles, il faut une règle de transcription permettant de convertir sous forme d'informations temporelles des entrées habituellement codées en intensité sous forme de vecteurs numériques. On représentera dans cette partie les différents types de codes par impulsions.

### 4.2.1 CODAGE PAR LE TEMPS DE LA PREMIERE IMPULSION

Dans ce type de codage, seule la première impulsion de chaque neurone est jugée pertinente. Par conséquent, toutes les autres impulsions subséquentes sont de ce fait ignorées [Loiselle, 2004] ou plutôt inhibées. Ainsi, seule la chronologie détermine l'information du flux et non pas le nombre d'impulsions (voir Figure 2.10).



**Figure 2.10** - Code par décharge utilisant le temps de la première impulsion :  
Le second neurone répond plus rapidement à un changement dans le stimulus que la première. Apparition de la stimulation marquée par la flèche.

### 4.2.2 CODAGE PAR ORDRE DE RANG (RANK ORDER CODING)

La stratégie est inspirée du système visuel au niveau duquel de nombreux types de codes auraient de la difficulté à opérer à la vitesse impressionnante que l'on retrouve

pour traiter certaines informations. Le principe est simple il suffit d'utiliser l'ordre dans lequel les neurones émettent des impulsions comme un code. Dans ce cas, la latence exacte à laquelle l'activité d'un neurone est déclenchée n'est pas critique - seul l'ordre de rang de chaque neurone est important [Thorpe et al., 1997 ; Thorpe et al., 1998]. Un tel schéma présente un certain nombre d'avantages.

Tout d'abord, un codage basé sur l'ordre sera plus robuste à la gigue temporelle bruitée de chaque impulsion comparé au codage temporel brute qui doit s'appuyer sur la précision temporelle, en particulier lorsqu'il est décodé par des détecteurs de coïncidence. Notons que les variations d'essai à l'essai dans la latence des neurones individuels ne sont pas nécessairement problématiques, puisque la caractéristique critique est l'ordre d'impulsions- toute fluctuation qui affecte toute la population n'a aucun effet sur l'ordre.

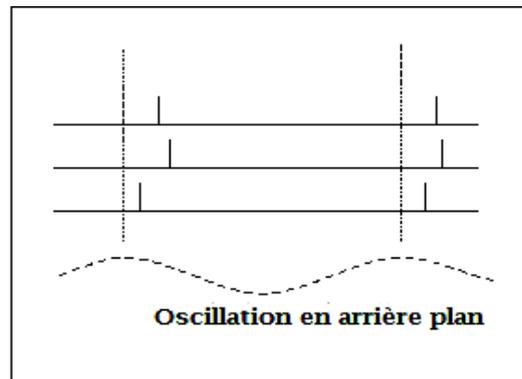
Deuxièmement, l'ordre dans lequel les neurones émettent des impulsions est totalement invariant [Gautrais et al., 1998].

Pour ce type de code, l'information est distribuée à travers une grande population de neurones et elle est représentée par le temps de décharge relatif des impulsions dans une seule vague de potentiels d'action.

Pour ce code, les neurones doivent donc être sensibles à la structure temporelle des impulsions qu'ils reçoivent en entrées. Ils doivent répondre de façon sélective à une séquence particulière d'activation de leurs entrées. Les patrons des impulsions d'entrées sont alors représentés dans le domaine temporel et un neurone donne une importance maximale à la première impulsion qu'il reçoit. L'influence des impulsions suivantes sur le neurone diminue progressivement. Le niveau d'activité résultant pour le neurone est le produit de ses poids synaptiques distribués à ses entrées et de la fonction de désensibilisation croissante. À l'aide d'un seuil approprié, le neurone peut alors être sensible à une séquence particulière de ses entrées [Loiselle, 2004].

### 4.2.3 CODAGE PAR PHASE

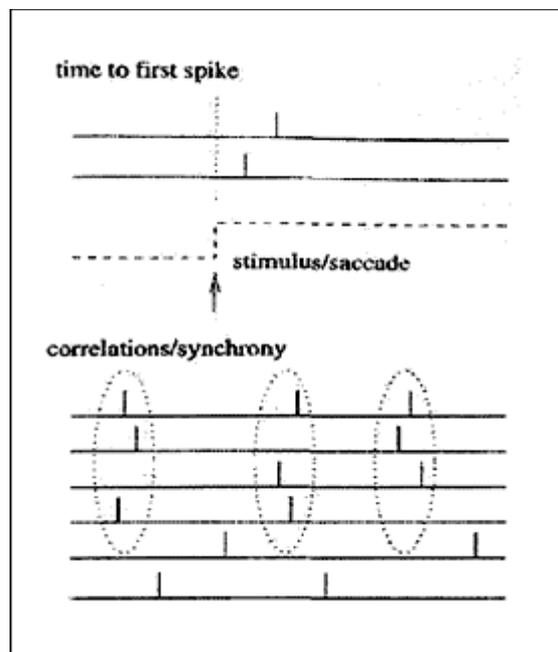
Ce codage est caractérisé par la possibilité d'utilisation du codage « *time-to-first-spike* ». Dans le cas où le signal de référence n'est pas un événement ponctuel, mais un signal périodique. Chez l'hippocampe, dans le système olfactif et aussi dans d'autres régions du cerveau, des oscillations d'une variable globale (par exemple l'activité de la population) sont très fréquentes.



**Figure 2.11** - Code par décharge utilisant les phases d'impulsions.

Ces oscillations peuvent servir de signal de référence interne. Les trains de potentiels neuronaux peuvent alors coder les informations dans la phase d'une impulsion par rapport à l'arrière-plan d'oscillation. Si l'entrée ne change pas entre un cycle et le suivant, puis le même schéma se répète périodiquement par des phases (voir Figure 2.11) [Gerstner et Kistler, 2002].

### 4.3 CORRELATION ET SYNCHRONIE



**Figure 2.12** - Code par le temps de la première impulsion (en haut), code par corrélation et synchronie (en bas).

La corrélation et la synchronisation est une technique que l'on peut facilement exploiter dans la phase de codage. Ces deux concepts peuvent s'avérer utiles si on

utilise éventuellement des neurones supplémentaires pour générer un code impulsif. La synchronisation entre deux ou plusieurs neurones aurait une signification particulière (voir Figure 2.12). D'un autre côté, on pourrait utiliser non seulement la synchronisation, mais aussi des patrons spatio-temporels précis des impulsions. Il est à noter que la corrélation des neurones auditifs est dépendante du stimulus et pourrait fournir de l'information au-delà de la cadence des décharges [Loiselle, 2004].

#### 4.4 CODAGE PAR CORRELATION RENVERSEE ET RECONSTRUCTION DE STIMULUS

Par exemple, dans ce type de codage, on prend un neurone qui est entraîné par un stimulus  $s$  en fonction du temps  $t$ . Chaque fois qu'une impulsion se produit, nous notons l'évolution temporelle du stimulus dans une fenêtre temporelle d'environ 100 millisecondes immédiatement avant la transitoire.

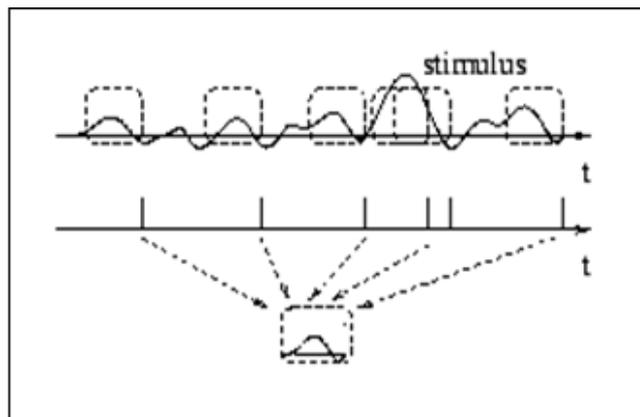


Figure 2.13 - Code par décharge utilisant la corrélation inversée.

La moyenne des résultats de plusieurs transitoires rapporte le cours typique de temps du stimulus juste avant la décharge. Un tel procédé s'appelle l'approche par «*corrélation inversée*» (voir Figure 2.13) [Gerstner et al., 2002].

## 5 CONCLUSION

Les neurones impulsifs «*spiking neurons*» a automatiquement inspiré à la définition de modèles de calculs originaux. Ainsi, la conception des SNN à l'image des neurones biologiques, tout comme les réseaux de neurones artificiels classiques, a fait intervenir la description des liens, ou connexions, entre les neurones. Les modèles impulsifs se différencient essentiellement par l'utilisation d'un mécanisme discret ou continu pour la génération des impulsions et par une approche intégrée (orientée potentiel) ou différentielle (orienté courant/conductances).

En résumé, ce chapitre décrit l'ensemble des caractéristiques de ces modèles qui sont très coûteux, mais aussi plus délicat à simuler. Forcément, on se doit de relever l'importance du concept de codage et d'illustrer ces différents types car ceci permet de mieux comprendre comment circule le flux d'informations dans cette catégorie de réseaux.

# CHAPITRE 3

## INTEGRATION DU MODELE A REPONSE IMPULSIONNELLE (SRM) DANS UNE ARCHITECTURE FEEDFORWARD

### 1 MOTIVATION

Proposé et explicitement détaillé par Gerstner [Gerstner,1995 ; Gerstner,1998], le modèle à réponse impulsionnelle (SRM) est connu pour avoir réduit le niveau de complexité des modèles biophysiques en un modèle phénoménologique [Chevallier, 2009]; C'est-à-dire une modélisation où l'on considère particulièrement le comportement du neurone mais on ignore tous mécanismes sous-jacents.

Le SRM possède des caractéristiques qui le différencient de tous autres modèles impulsionnels. Il s'agit de sa convenance à tous types d'apprentissage, sa capacité d'intégrer toutes architectures et de sa puissance de généralisation [Mouraud, 2009].

Au cours de notre travail, le choix du SRM semblait comme une évidence pour tout le potentiel que ce modèle propose sur le plan théorique et expérimentale. Ce chapitre est consacré à exposer notre approche sur plan théorique où deux optiques s'offrent à nous :

- Concernant l'unité de base du SNN, le SRM donne une description simple de la génération du potentiel d'action (PA). Ce modèle est directement inspiré du modèle integrate-and-fire (LIF) [Gerstner et al., 1996]. Ainsi, comme dans ce prédécesseur, il génère son potentiel d'action lorsque la tension dépasse un certain seuil. Cependant, contrairement au modèle LIF, le SRM comprend une période réfractaire lui permettant d'empêcher une réponse après une stimulation récente. Aussi, une différence de notation distingue ces deux modèles où le modèle LIF est formulé exclusivement d'équations différentielles, alors que le modèle SRM est formulée par l'utilisation des filtres.
- La construction d'un modèle connexionniste nécessite explicitement une bonne perception de son unité de base. En choisissant d'opter pour le modèle SRM, on débouche sur toute une série d'appréhension concernant l'organisation des neurones, les différents paramètres à initialiser et le mécanisme ainsi que

l'algorithme d'apprentissage à opter. Ainsi, divers règles d'apprentissage existent dans la littérature maniant à la fois l'apprentissage supervisé et non-supervisé offrant beaucoup de perspectives.

Dans ce chapitre, premièrement, on présente une synthèse des différents modes d'apprentissages et architectures pratiqués généralement sur les SNN. Ensuite, on va donner une description complète de notre approche à base du modèle à réponse impulsionnelle. Ainsi, on décrira les formulations du modèle SRM à block de retard et de l'algorithme d'apprentissage appliqué sur une architecture feed-forward : monocouche et multicouche.

## 2 APPRENTISSAGE DES RESEAUX DE NEURONES IMPULSIONNELS

Comme dans les modèles neuronales classiques, les SNN possèdent une large bibliothèque d'algorithmes d'apprentissages supervisés et non-supervisés. L'implémentation de ces algorithmes est contrôlée par l'architecture du SNN et le choix du modèle impulsionnel. L'apprentissage dépend exclusivement de l'adaptation des poids synaptiques cependant certains algorithmes cherche à exploiter d'autre optique tel que l'adaptation des délais de connexions [Maass et al., 1991].

Dans l'apprentissage supervisé, les règles d'apprentissages dépendent de l'instant de l'émission de l'impulsion pour passer à l'estimation du taux d'erreur. L'apprentissage non-supervisée se base sur des algorithmes stochastiques en observant l'espace des réponses impulsionnelles du réseau et en poussant le réseau à s'adapté suivant l'activité similaire détecté par le SNN face aux échantillons d'apprentissage. Cette activité déterminera les différentes catégories de réponses impulsionnelles

Concernant l'apprentissage non-supervisé, le Spike-Timing Dependent Plasticity (STDP) figure actuellement les algorithmes d'apprentissage neuromorphique les plus étudiés sur les dispositifs memristifs [Yu and Wong, 2010; Kuzum et al., 2012; Jo et al., 2010 ; Bichler et al., 2012 ; Bichler, 2012]. On retrouve aussi d'autres méthodes telles que Long Term Depression (LTD) et Long Term Potentiation (LTP)[ Kepecs et al., 2002 ; Kistler, 2002].

L'apprentissage supervisé est une pratique peu courante dans l'entraînement des SNN. Les règles les plus connus sont à base de l'algorithme de la descente du gradient tel que l'algorithme du SpikeProp [Bohte et al., 2000 ; Bohte et al., 2002 ; Moore, 2002 ; Schrauwen et al., 2004; Booiij et al.,2005] . D'autre travaux originaux tels que [Belatrache et al.,2003 ; Kasiński et al.,2006 ] ont étudié la possibilité des stratégies évolutionnaires pour l'apprentissage supervisé de réseaux de neurones impulsionnels , aussi très

répondus, les méthodes supervisées à base d'apprentissage Hebbian [Ruf et al., 1997 ; Ruf et al., 1998 ; Legenstein et al., 2003] ou encore les travaux [Ponulak et al., 2010] qui a présenté l'algorithme de Resume.

### **3 ARCHITECTURE DES RESEAUX DE NEURONES IMPULSIONNELS**

Plusieurs architectures neuronales peuvent être construites avec des neurones impulsionnels et cela selon l'objectif expérimentale à atteindre. On recense trois formats : l'architecture feedforward [Gerstner et al., 1996 ; Natschläger et al., 1998 ; Bohte et al., 2002 ; Booi,2004 ; Meftah et al. 2011], l'architecture feedback ou les réseaux récurrents [Gerstner et Kistler, 2002 ; Markram, 2002 ; Legenstein et al., 2003 ; Maass et al., 2004] et l'architecture hybride : Chaîne de décharge synchrone « *Synfire chain* » [Abeles, 1982] et Machine à états liquides [Maass et al., 2004 ; Maass et al., 2002 ; Natschläger et al., 2002].

Notre approche est exclusivement dédiée à l'intégration du SRM dans une architecture feedforward sous un apprentissage supervisé. Notre objectif est de démontrer la performance de ce type de réseaux si peu exploité dans les recherches dédiées aux réseaux de neurones impulsionnels. Mais aussi, de relever le défi quant à la complexité de réaliser un apprentissage supervisé sur ce type de réseaux.

### **4 DESCRIPTION DE L'APPROCHE UTILISEE**

Dans les sections précédentes, on a tracé les motivations qui nous ont poussés à choisir le modèle SRM pour la construction de notre SNN. Ensuite, pour justifier les prochaines étapes, on a exposé brièvement une synthèse des architectures et des modes d'apprentissages appliqués généralement sur les SNN.

Dans notre approche, il est clair qu'il faut définir explicitement les caractéristiques du SRM, l'organisation de notre SNN et l'algorithme d'apprentissage utilisé. Le type de codage sera défini ultérieurement dans la phase expérimentale.

#### **4.1 DESCRIPTION DE LA FORMULATION DU SRM A DELAIS TEMPORELS**

Typiquement dans un SRM, un processus de seuillage très primitive produit les trains d'impulsions sous le contrôle de deux fonctions à noyaux opposés qui contrôlent l'état du neurone [Maass, 1997]. Ainsi, la formalisation de ce modèle se compose d'une fonction kernel donnant la dynamique du potentiel de membrane du neurone (*voir*

Figure 3.1 (A)) et d'une autre fonction kernel calculant la sommation des potentiels post-synaptiques (PPS) (voir Figure 3.1 (B)).

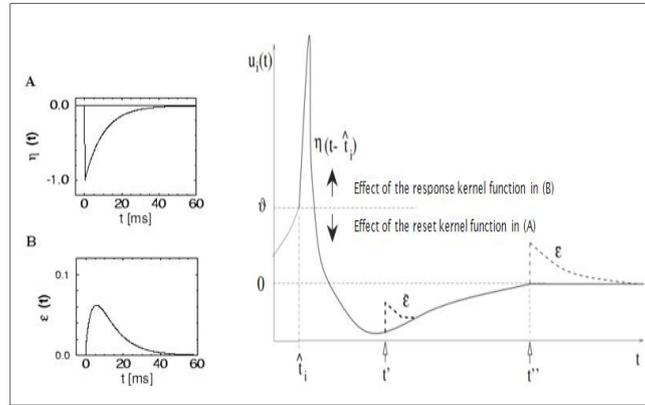


Figure 3.1 - Caractéristiques du modèle SRM

Sa formulation mathématique provient d'une intégration des équations différentielles sous-jacentes, ce qui la rend plus proche de celle des neurones classiques, même si elle reste basée sur le temps. Généralement, l'équation donnant l'évaluation du potentiel de membrane d'un neurone dans le formalisme SRM est décrit par :

$$u_i(t) = \eta(t - \hat{t}_i) + \sum_j w_{ij} \sum_{t_j^{(f)}} \varepsilon_{ij} (t - \hat{t}_i, t - t_j^{(f)}) + \int_0^\infty \kappa(t - \hat{t}_i, s) I^{ext}(t - s) ds \quad (3.1)$$

Un potentiel d'action (PA) est émis par le neurone  $n_i$  à l'instant  $t$  lorsque:

$$t = t_j^{(f)} \Leftrightarrow u_j(t) = \vartheta \wedge \frac{du_j(t)}{dt} > 0 \quad (3.2)$$

Il existe plusieurs formulations mathématiques et descriptives du modèle SRM. Notre approche s'intéresse particulièrement au modèle à délais temporels.

Dans notre approche, on a choisi un SRM à délais temporels ou à connexions retardées (voir Figure 3.2). Ce modèle fut introduit par [Gerstner et al., 1996] et a été utilisé par [Natschläger et al., 1998 ; Bohte et al., 2002 ; Booij et al., 2005 ; Meftah , 2011] . Un neurone  $j$  est connecté avec un ensemble de neurones pré-synaptique  $\Gamma_j$  limité et fixé par un intervalle de  $l$  délais.

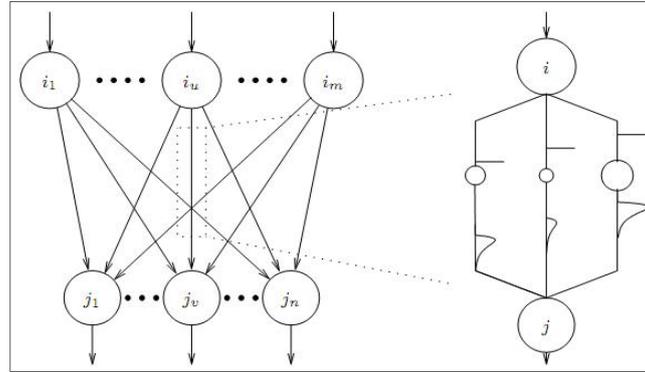


Figure 3.2 - Modèle SRM à délais temporels

La fonction réfractaire  $\eta$  contrôlé par le neurone post-synaptiques et la fonction d'accumulation du potentiel des neurones pré-synaptiques  $\varepsilon$  est illustré par l'équation (3.3) qui définit comment se calcule le PA du neurone  $j$  donné par  $u_j(t)$ .

$$u_j(t) = \sum_{t_j^{(f)} \in F_j} \eta(t - t_j^{(f)}) - \sum_{i \in \Gamma_j} \sum_{t_i^{(g)} \in F_i} \sum_{k=1}^l w_{ji}^k \varepsilon(t - t_i^{(g)} - d^k) \quad (3.3)$$

Où  $w_{ji}^k$  est le poids synaptique associé à la connexion retardé  $k$  et aux délais  $d^k$ .

Le spike d'un neurone  $j$  dépend de l'accumulation de son potentiel comparé à la valeur du seuil  $\mathcal{G}$ . Ainsi, une impulsion est émise à chaque fois que le  $u_j(t)$  atteint le seuil  $\mathcal{G}$  cette instant qu'on désigne par  $t^{(f)}$  est le temps de l'impulsion « spike-time ».

$$u_j = \mathcal{G} \wedge \left. \frac{du_j(t)}{dt} \right|_{t=t^{(f)}} > 0 \quad (3.4)$$

Par conséquent, toute l'activité impulsionnelle du neurone  $j$  est sauvegardée dans un vecteur où on marque tous les instants  $t_j^{(f)}$  où le neurone émet une impulsion par ordre chronologique.

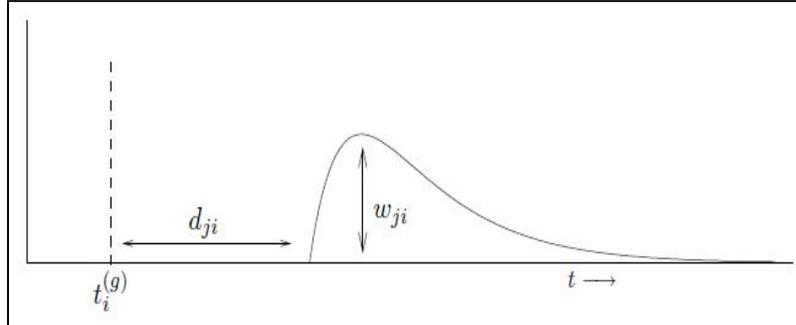
$$F_j = \{t_j^{(f)} : 1 \leq f \leq n\} : 1 \leq f < g \leq n \Rightarrow t_j^{(f)} < t_j^{(g)} \quad (3.5)$$

Aussi, l'activité d'un neurone dépend du choix de la formule qui contrôle son statut. Le PSP est modélisé généralement par la différence entre deux exponentielles décalées et ceci est souvent donné par la formule (3.6) :

$$\varepsilon(s) = \left[ \exp\left(-\frac{s}{\tau_m}\right) - \exp\left(-\frac{s}{\tau_s}\right) \right] H(s) \quad (3.6)$$

Où  $\mathcal{H}(s)$  représente la fonction à seuil :  $\mathcal{H}(s) = 0$  pour  $s \leq 0$  et  $\mathcal{H}(s) = 1$  pour  $s > 0$ . Les deux constantes de temps  $\tau_m$  et  $\tau_s$  (avec  $0 < \tau_s < \tau_m$ ) contrôlent les étapes

d'augmentation et de décalage de la fonction, et plus précisément la borne supérieure de la fonction.



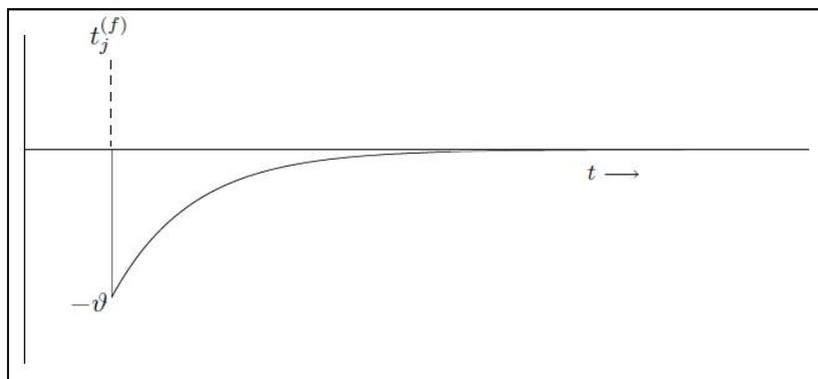
**Figure 3.3** - *Changement du potentiel post-synaptique donné par  $w_{ji}\epsilon(t - t_i^{(g)} - d_k)$  [Booij, 2004].*

Le changement du potentiel du neurone post-synaptique  $j$ , (voir Figure 3.3) causé par une impulsion du neurone pré-synaptique  $i$  est donnée par  $w_{ji}\epsilon(t - t_i^{(g)} - d_k)$ . Au temps  $t_i^{(g)}$  le neurone pré-synaptique  $i$  émet une impulsion (voir Figure 3.3 ; représenter par la ligne verticale discontinue). Après le décalage de la connexion,  $d_k$ , l'impulsion a un effet sur le neurone  $j$  à échelle du poids  $w_{ji}$ . Le potentiel du neurone post-synaptique  $j$  augmente brusquement, suivi d'une longue décroissance.

Une autre formulation de la fonction de la réponse impulsionnelle est possible

$$\epsilon(s) = \frac{s}{\tau} \exp\left(1 - \frac{s}{\tau}\right) \mathcal{H}(s) \tag{3.7}$$

Où  $\tau$  est une constante de temps qui contrôle l'augmentation et le décalage. Bien que cette fonction apparaît comme très différente, elle se comporte presque de la même façon.



**Figure 3.4** - *Changement du potentiel du neurone donné par  $\eta(t - t_j^{(f)})$  [Booij, 2004]*

Un autre processus qui change le potentiel du neurone, est la période réfractaire. Elle est modélisée par la fonction réfractaire  $\eta$ . Si le neurone  $j$  émet une impulsion à l'instant  $t_j^{(f)}$ , la fonction  $\eta(t - t_j^{(f)})$  le force à diminuer son potentiel au temps  $t$ . Encore une fois, le SRM ne fixe pas cette fonction, mais pour assurer la faisabilité il est nécessaire que  $\eta(s) = 0$  pour  $s \leq 0$  et elle est généralement non positive (voir Figure 3.4). On utilise une simple exponentielle :

$$\eta(s) = -\vartheta \exp\left(-\frac{s}{\tau_r}\right) \mathcal{H}(s) \quad (3.8)$$

Où  $\vartheta$  est le seuil du neurone,  $\mathcal{H}(s)$  représente la fonction à seuil, et  $\tau_r$  une autre constante de temps.

En résumé, le modèle SRM a des avantages très importants tels que sa capacité de simulation des réseaux de neurones avec un grand nombre de neurones à un coût raisonnable numérique et la Transparence dont il est doté dans le traitement du codage neuronal. Sa simplicité lui permet aussi d'être implémenté sur des données expérimentales. Le modèle SRM est le modèle le plus général qui combine le filtrage linéaire avec un seuil strict.

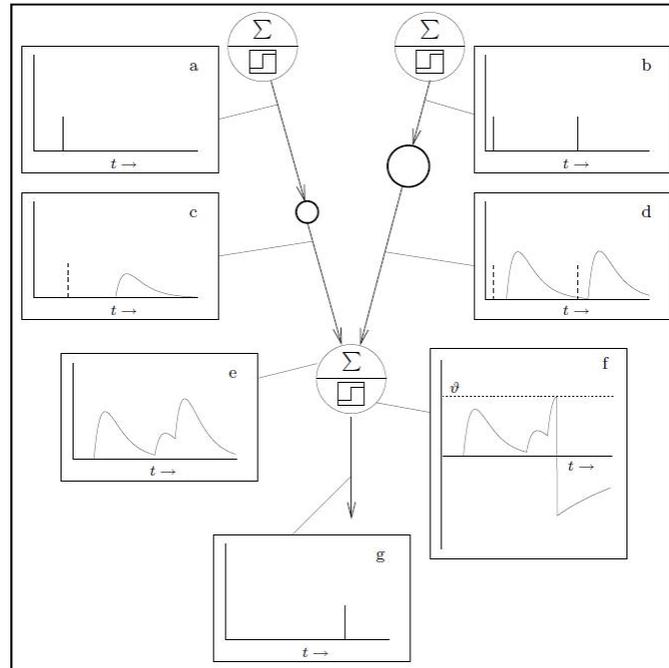
Aussi, Il existe des effets qui ne sont pas saisis par le modèle SRM car il n'est pas adapté à les décrire le blocage pharmacologique de canaux ioniques, le retard d'initiation de pointe et dépendance du seuil à l'entrée.

## 4.2 DESCRIPTION DE LA PROPAGATION DE L'INFORMATION DANS UN SNN FEED-FORWARD

Dans un réseau feedforward, chaque neurone de chaque couche est connecté avec les neurones de la couche suivante. La première couche, appelée couche d'entrée, est considérée comme l'entrée du réseau. Cette couche n'est pas considérée comme très importante car elle n'exécute aucune action sur les données en entrée (aucun traitement de calcul), mais ne fait que l'acheminement vers les couches suivantes. La dernière couche est la couche de sortie. Les trains d'impulsions qui traversent les neurones du réseau forment la sortie du réseau. Entre la couche d'entrée et la couche de sortie, il peut y avoir un nombre quelconque de couches cachées.

Comme notre approche reposera sur une architecture feed-forward, un simple exemple illustré dans la Figure 3.5 décrit le procédé des équations détaillé la section précédente. On remarque que deux neurones sont à l'origine d'une impulsion émise par un troisième neurone. Dans les schémas (a) et (b), deux neurones émettent des impulsions, celui de gauche une seule, et celui de droite, deux. Ainsi, les schémas (c), (d) illustrent

l'effet des décalages et des poids synaptiques sur le PSP qui sont transformés en utilisant la fonction  $\varepsilon$  pour avoir une réponse impulsionnelle .



**Figure 3.5** - Un exemple graphique qui explique le travail d'un neurone impulsionnel.

Sur la figure, on remarque que la synapse gauche a un plus grand délai de décalage, mais un moindre poids synaptique par rapport à la synapse de droite. Le schéma (e) regroupe les impulsions entrantes qui sont additionnées afin d'avoir le potentiel du neurone. En parallèle, le schéma (f) illustre le potentiel qui est comparé à un seuil et dès qu'il atteint ce seuil il décroît à  $-\vartheta$  pour avoir une période réfractaire. Finalement, la sortie du neurone est une impulsion émise à l'instant dans le schéma (g) où le potentiel a atteint le seuil  $\vartheta$  [Booij, 2004].

### 4.3 DESCRIPTION DE L'ALGORITHME SPIKEPROP A MULTI-SPIKES

Cette algorithme a été proposé par [Booij, 2004] qui s'est inspiré directement de la faille de l'algorithme SpikeProp proposé par [Bohte et al., 2000] quant aux traitement de l'information codé sur plusieurs impulsions. Effectivement, dans SpikeProp, la formulation de l'adaptation des poids synaptiques repose essentiellement sur la propagation d'informations codées uniquement sur un seul spike. Ce type de codage se présente rarement dans le traitement des données réelles. Ainsi, le SpikeProp à multi-spikes est plus générale.

Dans cet algorithme, deux formulations se présentent suivant le type d'architecture : monocouche, multicouche. Les sections suivantes décriront ces formulations.

### 4.3.1 FORMULATION DE L'ALGORITHME POUR UN SNN A MONOCOUCHE

Dans cette section, on va décrire formellement une règle d'apprentissage qui permet de déterminer les poids d'un réseau de neurone impulsionnel sans couche cachée. Pour trouver les bons poids, on les modifie à chaque itération afin de minimiser une certaine erreur. Cette minimisation est accomplie par la méthode de descente du gradient ; ceci, en diminuant l'étendue de l'erreur proportionnellement à sa dérivée. Ce type d'optimisation de paramètre est un outil très important pour les algorithmes d'apprentissage.

La mesure de l'erreur est déterminée par la différence entre la sortie désirée et la sortie obtenue. De ce fait, il est important de choisir le bon type de codage neuronal que doit avoir le neurone de sortie. On utilise le codage qui prend en considération la première impulsion  $t_j^{(1)}$  et ignore toutes les impulsions qui viennent après. L'erreur du réseau peut être déterminée par la somme des différences aux carrés entre le temps d'impulsion désirée et le temps d'impulsion obtenu :

$$E_{net} = \frac{1}{2} \sum_{j \in J} (t_j^{(1)} - \hat{t}_j^{(1)})^2 \quad (3.9)$$

où  $\hat{t}_j^{(1)}$  est le temps désiré de la première impulsion du neurone  $j$ , et  $J$  représente la couche de sortie.

Afin de minimiser l'erreur du réseau, on doit changer chaque poids proportionnellement à la dérivée de l'erreur on respectant ce poids. Le changement du poids d'une synapse qui va du neurone  $i$  au neurone  $j$  est alors noté par :

$$\Delta w_{ji}^k = -\eta \frac{\partial E_{net}}{\partial w_{ji}^k} \quad (3.10)$$

Où  $\eta$  est une petite valeur constante qui détermine le taux d'apprentissage et  $w_{ji}^k$  est le poids synaptique qui va du neurone d'entrée  $i$  au neurone de sortie  $j$  avec un délai  $d^k$ . Comme le poids synaptique influence seulement les temps d'impulsions du neurone de sortie  $j$ , on peut étendre le deuxième facteur de l'équation (3.11) à :

$$\frac{\partial E_{net}}{\partial w_{ji}^k} = \frac{\partial E_{net}}{\partial t_j^{(1)}} \frac{\partial t_j^{(1)}}{\partial w_{ji}^k} \quad (3.11)$$

Le premier facteur, qui exprime les changements de l'erreur du réseau selon la première impulsion du neurone de sortie  $j$ , est donnée par :

$$\frac{\partial E_{net}}{\partial t_j^{(1)}} = t_j^{(1)} - \hat{t}_j^{(1)} \quad (3.12)$$

Le calcul du second facteur de la formule (3.11), qui exprime le changement du temps d'impulsion par rapport au changement du poids, est plus difficile, car il n'y a pas de formule pour exprimer cela. Plusieurs formules successives par remplacement des facteurs permettent ce type de calcul. On obtient la formule suivante :

$$\frac{\partial t_j^{(1)}}{\partial w_{ji}^k} = \frac{-\sum_{t_i^{(g)} \in \mathcal{F}_i} \varepsilon(t_j^{(1)} - t_i^{(g)} - d^k)}{\sum_{i,k} \sum_{t_i^{(g)} \in \mathcal{F}_i} w_{ji}^k \varepsilon'(t_j^{(1)} - t_i^{(g)} - d^k)} \quad (3.13)$$

On a enfin la formule qui exprime le changement du poids concrètement comme suit :

$$\Delta w_{ji}^k = -\eta \frac{-\sum_{t_i^{(g)} \in \mathcal{F}_i} \varepsilon(t_j^{(1)} - t_i^{(g)} - d^k)}{\sum_{i,k} \sum_{t_i^{(g)} \in \mathcal{F}_i} w_{ji}^k \varepsilon'(t_j^{(1)} - t_i^{(g)} - d^k)} (t_j^{(1)} - \hat{t}_j^{(1)}) \quad (3.14)$$

**Pour tous** les neurones de sortie  $j \in J$  **faire**

    Calculer la dérivée partielle du réseau par rapport  $a_j$  suivant (3.12)

    Calculer le gradient du potentiel (le dénominateur de (3.13))

**Si** gradient < 0.1 **alors**

    gradient ← 0.1

**Fin si**

**Pour tous** les neurones d'entrée  $i \in I$  **faire**

**Pour tous** les poids  $w_{ji}^k$  des connexions de  $i$  vers  $j$  **faire**

            Calculer la dérivée partielle du réseau par rapport à  $w_{ji}^k$  en utilisant le numérateur de la formule (3.13) et le gradient

            Calculer le changement du poids  $\Delta w_{ji}^k$  en utilisant (3.10) et (3.11)

**Fin pour**

**Fin pour**

**Fin pour**

**Pour tous** les poids  $w_{ji}^k$  **faire**

$$w_{ji}^k \leftarrow w_{ji}^k + \Delta w_{ji}^k$$

**Fin pour**

**Figure 3.6** - Algorithme d'apprentissage pour un SNN à une couche [booi, 2004].

L'algorithme illustré dans la Figure.3.6 proposé par Olaf Booi (2004) résume cette règle qui n'est autre qu'une variante de l'algorithme de rétro-propagation dérivée de l'algorithme « SpikeProp ».

### 4.3.2 FORMULATION DE L'ALGORITHME POUR UN SNN MULTICOUCHE

Dans la section précédente, on a exposé un algorithme d'apprentissage pour un SNN monocouche. Cet algorithme ne peut adapter qu'un seul niveau de poids synaptiques soit les poids entre la couche d'entrée et la couche de sortie.

Dans le même contexte, on développera un algorithme qui est une extension du précédent algorithme et qui permet de réaliser un apprentissage aux réseaux de neurones impulsionnels à plus d'une couche.

On s'intéresse au cas où le réseau a une couche cachée, mais il est facile de voir comment ceci peut être généralisé aux réseaux avec un nombre de couches aléatoires. La Figure 3.7 montre un réseau très basique qui indique les différentes variables qui sont utilisées dans un réseau à deux couches. Les poids des connections de la couche d'entrée H vers la couche cachée I et ceux de la couche cachée I allant à la couche de sortie J doivent être modifiés pour minimiser l'erreur du réseau. Le changement des poids  $w_{ji}^k$  allant aux neurones de sortie est équivalent au changement du même poids d'un réseau à une couche, équation (3.10). La règle de modification des poids  $w_{ji}^k$  allant aux neurones cachés est calculée en utilisant la méthode du gradient descendant :

$$\Delta w_{ih}^k = -\eta \frac{\partial E_{net}}{\partial w_{ih}^k} \tag{3.15}$$

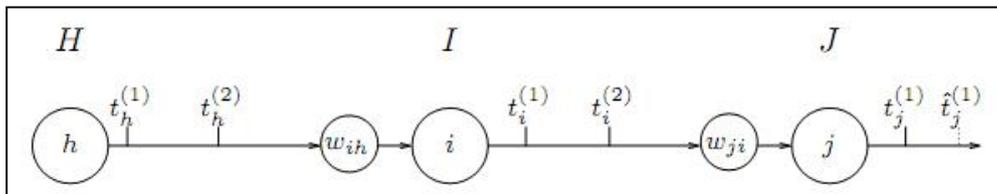


Figure 3.7 - Un exemple simple d'un SNN à une couche cachée pour montrer quelles sont les variables utilisées.

Encore une fois, la dérivée partielle de l'erreur du réseau par rapport au poids doit être calculée, de la même façon qu'il a été fait pour les poids allant vers le neurone de sortie, équation (3.10). On doit aussi prendre en considération qu'un neurone caché peut émettre plus qu'une fois, contrairement aux impulsions de sortie, où seulement la première impulsion influence l'erreur. Donc l'erreur dépend de toutes les impulsions  $t_i^{(g)}$  du neurone caché, on peut la voir comme une fonction du poids  $w_{ih}^k$  comme suit:

$$\frac{\partial E_{net}}{\partial w_{ih}^k} = \sum_{t_i^{(g)} \in \mathcal{F}_i} \frac{\partial E_{net}}{\partial t_i^{(g)}} \frac{\partial t_i^{(g)}}{\partial w_{ih}^k} \tag{3.16}$$

Le premier facteur de la formule est calculé de cette façon :

$$\frac{\partial E_{net}}{\partial t_i^{(g)}} = \sum_{j \in \Gamma^i} \frac{\sum_k w_{ji}^k \varepsilon' (t_j^{(1)} - t_i^{(g)} - d^k)}{\sum_{i,k} \sum_{t_i^{(f)} \in \mathcal{F}_i} w_{ji}^k \varepsilon' (t_j^{(1)} - t_i^{(f)} - d^k)} (t_j^{(1)} - \hat{t}_j^{(1)}) \quad (3.17)$$

On obtient en fin de calculs la formule récursive du calcul de la dérivée du temps d'impulsion d'un neurone par rapport à celle du poids synaptique correspondant :

$$\frac{\partial t_i^{(g)}}{\partial w_{ih}^k} = \frac{-\sum_{t_h^{(p)} \in \mathcal{F}_h} \varepsilon (t_i^{(g)} - t_h^{(p)} - d^k) + \sum_{t_i^{(f)} \in \mathcal{F}_i} \eta' (t_i^{(g)} - t_i^{(f)}) \frac{dt_i^{(f)}}{dw_{ih}^k}}{\sum_{t_i^{(f)} \in \mathcal{F}_i} \eta' (t_i^{(g)} - t_i^{(f)}) + \sum_{h,k} \sum_{t_h^{(p)} \in \mathcal{F}_h} w_{ih}^k \varepsilon' (t_i^{(g)} - t_h^{(p)} - d^k)} \quad (3.18)$$

## 5 CONCLUSION

La description de l'architecture et de l'apprentissage des réseaux de neurones impulsionnels est une des phases les plus importantes avant de commencer l'expérimentation du modèle. Cette phase représente une difficulté très particulière par rapport aux modèles neuronales classiques. Cependant, la bonne définition de l'approche ne suffit pas à garantir la performance du modèle ou la faisabilité du problème.

Il y a principalement deux facteurs qui influent sur l'apprentissage: la qualité de l'échantillonnage d'apprentissage (les exemples qui constituent la base d'apprentissage) et la diversité des valeurs. En effet, le réseau de neurones généralisera mieux (aura plus de chances de répondre correctement en lui donnant en entrée des informations non présentes dans les exemples d'apprentissage) si la qualité de l'échantillonnage est meilleure et si les données des exemples d'apprentissage sont plus variées. Intuitivement, on est conscient que s'il sait répondre correctement pour un nombre prédéfini de situations, il sera alors plus apte à répondre correctement face à de nouvelle situation.



# CHAPITRE 4

## EXPERIMENTATION DU SNN DANS DES CAS D'ÉCOLE

### 1 INTRODUCTION

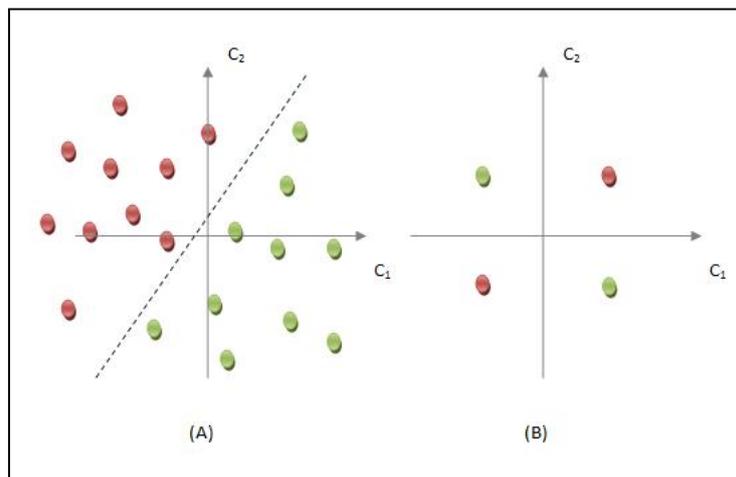
Bien que nos objectifs initiaux visent à élaborer un système robuste à base de réseaux de neurones impulsionnels dans la reconnaissance de la parole, l'importance des enjeux et des contraintes dans les quels on s'est rapidement retrouvé dès le lancement de notre thèse, nous a d'abord obligé à essayer de trouver un algorithme d'apprentissage raisonnablement et une stratégie de codage efficace pour tracer notre approche.

Par conséquent, on a commencé à construire et à tester notre modèle SNN à base d'unités SRM sur des cas d'école « *benchmarks* » afin de valider sa puissance, son efficacité et sa robustesse sur différentes structures : monocouche et multicouche ; tout en appliquant, de manière peu traditionnelle, un apprentissage supervisé. Dans la littérature, la majorité des travaux sur les SNNs ont suivi ce même parcours pour mieux défendre leurs approches [Bohte, 2002 ; Seung, 2003 ; Booiij, 2004 ; Paugam-Moisy et al., 2010 ; Marcano-Cedeño et al., 2011].

Dans ce chapitre, on présente les différentes étapes et applications qui ont permis à tester la validité et la rentabilité de notre approche. Naturellement, on a procédé au test de la performance du réseau SNN à travers des cas d'école. Ainsi, comme stratégie, on a revisité le problème classique du XOR, essentiellement, pour redémontrer l'originalité du SNN à base de SRM face à une minimisation maximale de sa structure en jouant uniquement sur la stratégie du codage. Ensuite, on a cherché à explorer divers problèmes de classification à travers la construction de différents systèmes d'aide à la décision: la reconnaissance labiale des mots isolés, l'identification des empreintes digitales et le diagnostic du cancer de sein. Ceci nous a permis de peaufiner notre approche de façon à devenir plus performante et plus robuste.

## 2 PROBLEME DU XOR

Les fonctions booléennes à deux variables sont parmi les fonctions les plus simples à réaliser pour un classificateur. Logiquement, les réseaux de neurones artificiels devraient être capables d'apprendre à calculer de telles fonctions. Cependant, le XOR avait pourtant ralenti l'évolution de l'approche connexionniste en tenant en échec le modèle monocouche « *perceptron* » après la publication de Minsky et Papert [Minsky et al., 1969]. Dans leurs travaux, ils avaient démontré l'incapacité du perceptron à résoudre tout les problèmes de classification non-linéaires (voir figure 4.1). L'exploitation des RNAs va fortement diminuer jusqu'en 1982, où T. Kohonen présente ses travaux [Kohonen, 1982] sur les mémoires associatives et propose des applications à la reconnaissance des formes. Finalement, il a fallu attendre l'introduction des réseaux de neurones multicouches (PMC) et l'algorithme de rétropropagation [Rumelhart et al., 1986] pour dépasser l'impasse engendrée par cette problématique.



**Figure 4.1** - A) ensemble de point linéairement séparables par un droite en deux classes. B) ensemble du XOR plus simple mais impossible à séparer par une droite, c'est un problème non-linéaire

Dans ce qui suit, on va redémontrer qu'à l'opposé des modèles traditionnelles des RNAs, ce cas de classification non-linéaire peut être facilement résolu avec un SNN et à l'aide d'une seule couche ou encore mieux d'un seule neurone SRM.

### 2.1 EXPERIMENTATION DU SNN SUR XOR

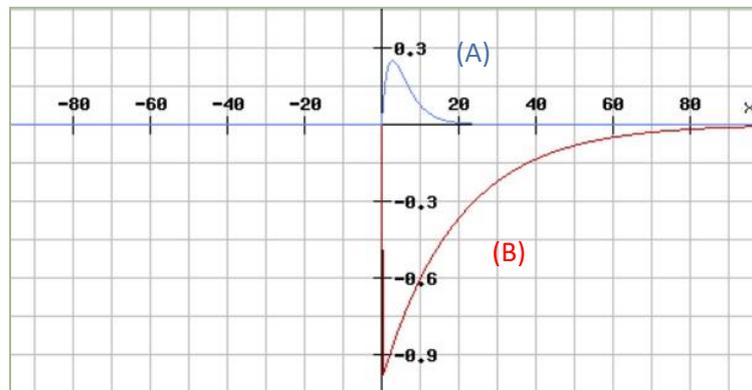
Dans cette partie, on a voulu généraliser notre approche sur le problème du XOR en passant par trois expériences [Ourdighi et al., 2009]. Dans ces trois expériences, notre objectif était de rendre l'architecture du SNN la plus compacte possible pour réduire le temps de calcul au niveau de l'apprentissage et de la généralisation. Par conséquent, on a décidé de ne pas suivre la même démarche des travaux de [Booji, 2004 ; Stromatias, 2011] qui propose d'abord un SNN multicouches mais plutôt commencer directement

par tester des solutions sans couche cachée. Chaque expérience révèle un niveau de complexité qui s'avèrera très instructif pour l'élaboration de notre approche.

## 2.2 PARAMETRES DU SNN

Dans le chapitre précédent, on avait déjà décrit la nature du SRM utilisé dans notre approche et détaillé tous ces formulations (voir Section 3.4.1). Dans cette section, on apporte plus de détails quant aux choix des fonctions et de l'initialisation des constantes.

Pour la fonction produisant la réponse impulsionnelle  $\varepsilon(s)$  (avec  $S = t - t_i - d_k$ ), on a choisi la formule (3.6) qui est modélisée par la différence entre deux exponentielles décalées. Ces paramètres  $\tau_m$  et  $\tau_s$  sont des constantes de la fonction qui définie la largeur de la courbe de la réponse impulsionnelle. Dans cette partie de l'expérimentations leurs valeurs sont:  $\tau_m = 4$  et  $\tau_s = 2$ . Pour assurer la causalité, on utilisera bien évidemment la fonction  $H(s)$  ( $H(s) = 0$  Si  $S < 0$ , sinon 1.), celle ci permettra d'ignorer toutes impulsions ultérieure à l'instant  $t$ . La fonction représentant l'effet réfractaire  $\eta(s)$  est donnée par l'équation (3.8) avec  $\tau_r = 20$ . Les deux fonctions sont illustrées par la Figure 4.2.



**Figure 4.2** - Neurone SRM pour XOR :A) La fonction produisant la réponse impulsionnelle (en Bleu). B) La fonction produisant l'effet réfractaire (en rouge)

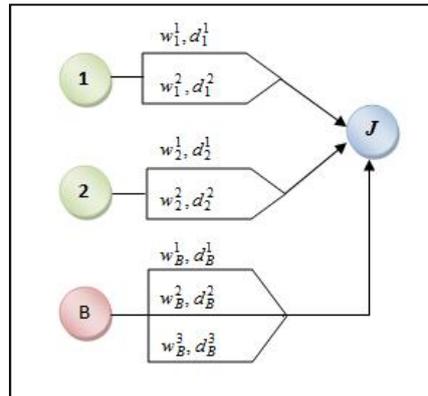
Pour l'architecture du SNN, on précisera dans la section suivante la configuration de chaque réseau SNN utilisé dans la phase expérimentale.

## 2.3 PREMIERE EXPERIENCE

Pour notre première implémentation, on utilise deux neurones qui émettent deux spikes chacun, plus un neurone biais dont le rôle est de faire en sorte que les impulsions en sortie soient émises au moment désiré (voir Figure 4.3). A titre

d'expérimentation, on y ajoute les mêmes contraintes utilisées par [Booji, 2004] décrit ci-dessous :

$$\begin{aligned}
 w_1^1 &= -w_2^1, w_1^2 = -w_2^2 \\
 d_1^1 &= d_2^1, d_1^2 = d_2^2
 \end{aligned}
 \tag{4.1}$$



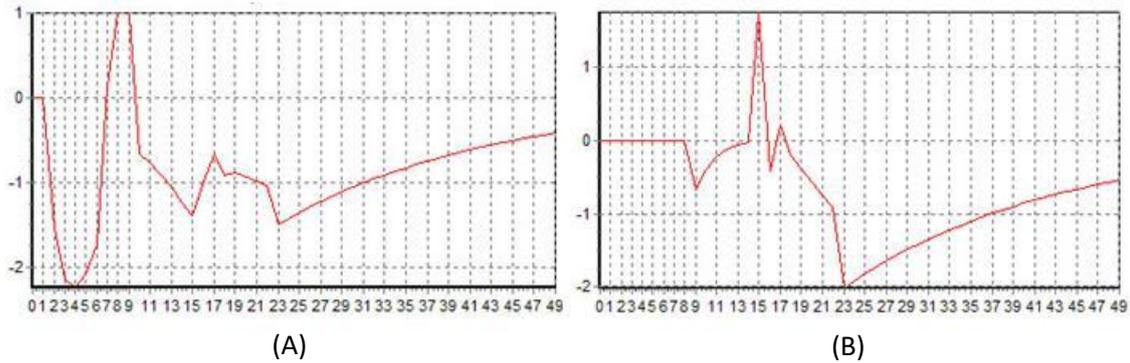
**Figure 4.3** - SNN monocouche [3 1] pour la résolution du problème du XOR: Un neurone pour chaque entrée plus un neurone pour le biais.

Ces contraintes font que les deux neurones d'entrée sont opposés (à connexion excitatrice et l'autre inhibitrice). Ceci permet de les forcer à s'annuler si ils émettent en même temps : le cas d'entrée de même valeur (sortie XOR = 0). Dans le cas où les valeurs d'entrées sont différentes, il y aura un décalage des délais et les deux neurones ne se neutraliseront pas (sortie XOR = 1). Le neurone Biais fait en sorte d'amener le réseau à n'émettre qu'en deux temps prédéfinies ( $t = 9s$  et  $t = 16s$ ).

Entrée ( $e_1$ )	Entrée ( $e_2$ )	( $t_1$ )	( $t_2$ )	Sortie désirée( $j$ )	( $t_j$ )
1	1	9 s	16 s	0	16 s
1	0	9 s	9 s	1	9 s
0	1	16 s	16 s	1	9 s
0	0	16 s	9 s	0	16 s

**Table 4.1** - Codage de type Time-to-first-Spike des Entrées et de la sortie désirée du SNN.

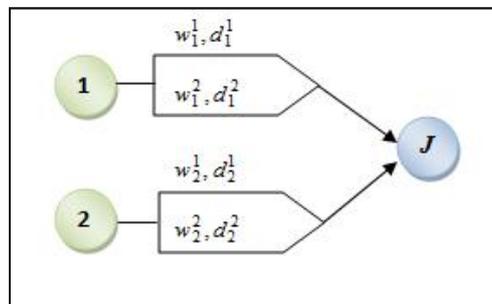
La stratégie de codage est de type Time-to-first-Spike. Concernant l'apprentissage, on a opté pour le mode supervisé en se basant sur l'algorithme de la rétropropagation du gradient de l'erreur décrit dans la section 3.4.3.1. La Table 4.1 illustre le codage utilisé dans cette expérience.



**Figure 4.4** - Représentation du PA pour les deux sorties possibles du SNN monocouche [3 1] pour XOR : (A) Réponse positive, le premier Spike de neurone de sortie est enregistré à  $t=8s$ . (B) Réponse négative, le premier Spike de neurone de sortie est enregistré à  $t=15s$

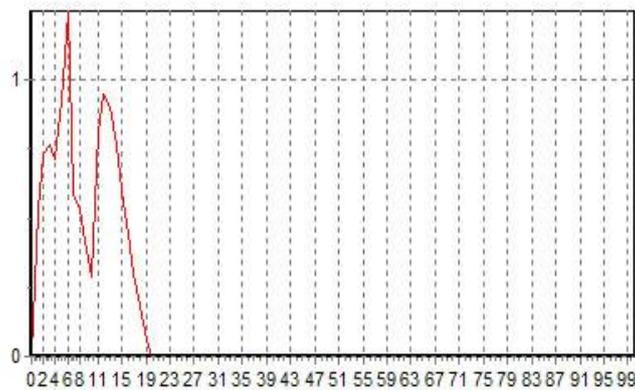
Les résultats obtenus sont décrits dans la Figure 4.4. Le graphe (A) correspond à la sortie calculée du SNN de deux entrées différentes et le graphe (B) représente la sortie calculée du SNN de deux entrées semblables, c'est-à-dire  $t=8s$  pour sortie « positive » et  $t=15s$  pour sortie « négative ». On remarque que les instants des impulsions générées par le SNN approchent clairement la sortie désirée ( $t=9s$  et  $t=16s$ ).

## 2.4 DEUXIEME EXPERIENCE



**Figure 4.5** - SNN monocouche [2 1] pour la résolution du problème du XOR: Un neurone pour chaque entrée sans biais.

Ensuite, on a voulu rendre le SNN beaucoup plus indépendant en supprimant de la structure le neurone biais. L'apprentissage se basera uniquement sur deux neurones d'entrée et un seul de sortie (voir Figure 4.5). Ainsi, on a décidé de changer la stratégie de codage de la couche de sortie. Dans le cas d'une réponse positive, une impulsion est attendue à la sortie, sinon rien.

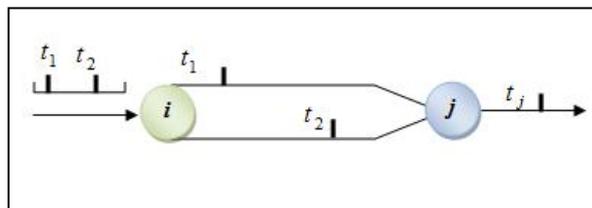


**Figure 4.6** - Représentation du PA pour de la sortie positive du SNN monocouche [2 1] pour XOR (la sortie négative le PA n'atteint pas le seuil).

La Figure 4.6 démontre l'inutilité du neurone biais pour garantir la reproduction du XOR par un SNN. Néanmoins, il a fallu changer la stratégie de codage pour réaliser cette minimisation dans la structure du SNN.

## 2.5 TROISIEME EXPERIENCE

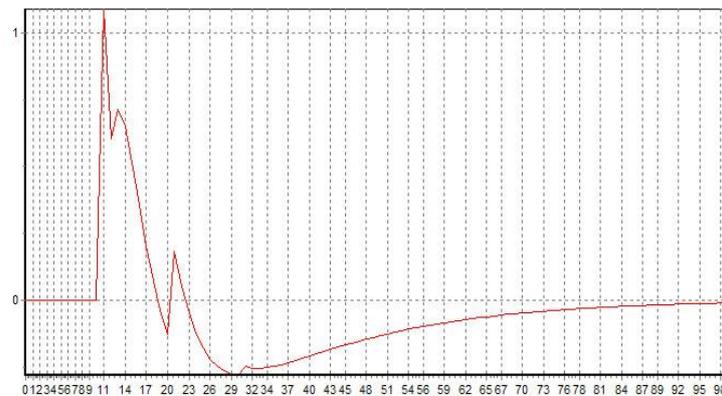
Pour pousser plus loin notre expérimentation, on a décidé de tester à réduire les paramètres de notre SNN le plus possible pour mettre en évidence sa puissance. Dans cette optique, pour modéliser le problème du XOR, on a imaginé un réseau avec un seul neurone en entrée et un seule neurone de sortie (voir Figure 4.6).



**Figure 4.7** - SNN monocouche [1 1] pour la résolution du problème du XOR: Un neurone les deux entrées.

Ainsi, le neurone  $i$  émet deux impulsions correspondantes aux deux entrées du XOR ( $e_1, e_2$ ). On fixe les délais de retard pour chaque impulsion comme suit :

- Le nombre le plus petit entre  $e_1$  et  $e_2$  est celui qui émet le plutôt.
- Plus les deux entrées  $e_1$  et  $e_2$  sont différents, plus leurs temps d'émission sont rapprochés (sinon plus ils sont proches plus leur temps de retard s'écartent).

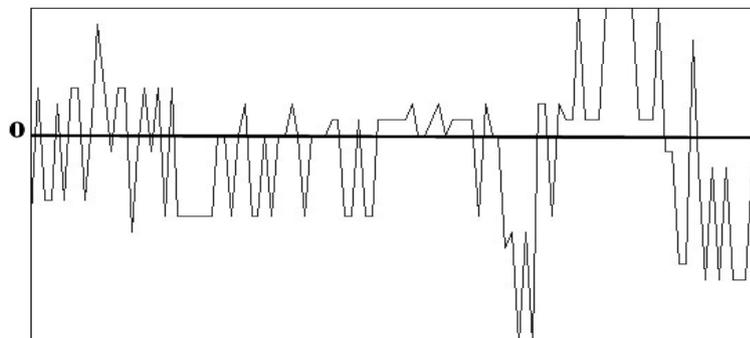


**Figure 4.8** - Représentation du PA pour de la sortie négative du SNN monocouche [1 1] pour XOR enregistré à  $t=12s$  (la sortie positive arrive généralement avant  $t=10s$ )

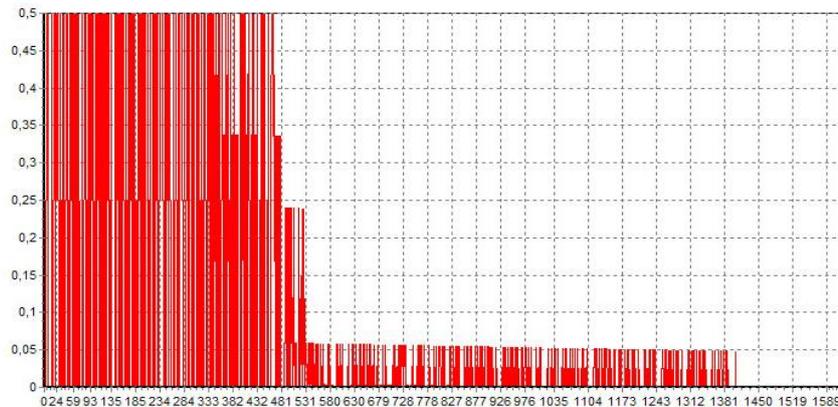
Avec cet algorithme de codage très simple, pour des poids synaptiques générés dans un intervalle de  $[1,2]$ , le SNN donne une solution semblable à celle donnée par l'expérience précédente (voir *Figure 4.8*). Ainsi, cette réduction des paramètres résout elle aussi parfaitement le problème du XOR.

## 2.6 EVALUATION ET DISCUSSIONS

Dans un apprentissage supervisé, on a remarqué que le SNN était extrêmement instable et lorsqu'il converge c'est souvent vers un minimum local (voir *figure 4.9* et *figure 4.10*). Cette instabilité est due d'une part, au fait que son algorithme d'apprentissage est contraint d'adapter les poids du réseau afin de reproduire des impulsions aux moments choisis. Sachant que dans le cas des SNNs, la sortie désirée n'est pas une valeur à initialiser comme dans les modèles classiques mais tout un vecteur à construire représentant le train d'impulsions désirées.



**Figure 4.9** - Exemple Courbe de l'erreur  $(t_j^{(1)} - \hat{t}_j^{(1)})$  du SNN pour XOR extrêmement instable.



**Figure 4. 10** - Courbe de l'erreur quadratique, obtenu par l'algorithme du PMC qu'on adapté sur un SNN.

La courbe de l'erreur quadratique (voir Figure 4.10) montre un changement notable par rapport à la Figure 4.10. Même si la courbe n'est pas lisse, elle est beaucoup plus stable, et diminue toujours.

Par ailleurs, nous avons noté deux phénomènes notables :

- il est plus facile pour le réseau de respecter la production d'une impulsion tardive (par exemple à  $t=16s$ ) que de la produire dans les premiers instants. L'explication la plus plausible à ce phénomène c'est que l'algorithme adapte les poids des connexions responsables des premières impulsions (par exemple inférieur  $t=16s$ ) contrôlant assez finement leur influence sur l'émission d'un spike à  $t=16s$ . seulement le réseau se retrouve dans un dilemme: d'un coté il doit maintenir des poids assez élevés pour générer des impulsions à  $t=16s$ , mais d'un autre coté les poids doivent aussi être diminués pour permettre un spike à  $t=9s$  par exemple et pas après.
- Ce paradoxe est à l'origine d'un « effet de bord » ou l'algorithme augmente à l'infini le poids de la connexion la plus précoce (la plus éloignée), dans le but de réduire le déficit des poids des connexions suivantes. Pour réduire ce paradoxe, une solution consiste à alterner connexions inhibitrices et excitatrices dans la structure du réseau.

De nombreuses recherches actuelles en SNN ont pour but de trouver l'algorithme d'apprentissage parfait à la fois stable et universel pour tous les domaines d'applications. Mais au vu du paradoxe cité, il semble qu'un algorithme aussi parfait soit-il ne vaut rien s'il la structure du réseau et la stratégie de codage employé ne vont pas dans le même sens.

### 3 IDENTIFICATION DES EMPREINTES DIGITALES SUR FVC2004

L'identification des empreintes digitales nécessite un outil à la fois puissant et robuste afin de réaliser une reconnaissance online. Dans cette deuxième partie de l'expérimentation, on a mis en œuvre un SNN opérationnel à base de SRM sur une problématique beaucoup plus complexe tout en limitant le nombre d'empreintes digitales à traiter de la base de données FVC2004, notre objective dans cette phase se limite au cadre expérimentale. Ainsi, L'architecture et le type du codage ne seront plus les seuls facteurs à favoriser. Mais, la rapidité de la réponse du système induira forcément à optimiser les paramètres et la structure du SNN le plus possible pour éviter une réponse lente su système pendant de la généralisation.

Pour notre implémentation, on a exploité la FVC2004 (disponible : <http://bias.csr.unibo.it/fvc2004/>) où quatre différentes bases de données (DB1, DB2, DB3 et DB4) ont été conçues en utilisant différentes technologies de capteurs comme suit:

- DB1 : capteur optique « V300 » par CrossMatch.
- DB2 : capteur optique « U.are.U 4000 » par Digital Persona.
- DB3 : capteur à balayage thermique « FingerChip FCD4B14CB » par Atmel.
- DB4 : génération d'empreintes synthétiques avec le logiciel SFinGe v3.0.

Il est à noter que les bases d'empreintes de la FVC2004 sont des patrons difficiles à traiter que celles des précédentes éditions (la FVC2002 et la FVC2000), et cela à cause des différentes perturbations qu'a subit la capture des empreintes. La table 4.2 montre un récapitulatif des informations sur chacune des quatre bases d'empreintes.

	<i>Type de Capteur</i>	<i>Taille de l'image</i>	<i>Résolution</i>
<b>DB1</b>	<i>Capteur Optique</i>	<i>640x480 (307 Kpixels)</i>	<i>500 dpi</i>
<b>DB2</b>	<i>Capteur Optique</i>	<i>328x364 (119 Kpixels)</i>	<i>500 dpi</i>
<b>DB3</b>	<i>Capteur à Balayage Thermique</i>	<i>300x480 (144 Kpixels)</i>	<i>512 dpi</i>
<b>DB4</b>	<i>SFinGe v3.0</i>	<i>288x384 (108 Kpixels)</i>	<i>environ 500 dpi</i>

**Table 4.2-** Récapitulatif des informations sur les bases d'empreintes.

#### 3.1 EXPERIMENTATION DU SNN SUR IDENTIFICATION DES EMPREINTES DIGITALES

Dans cette partie de l'expérimentation, on a choisi les empreintes de la première base (DB1). La DB1 contient huit patrons de dix personnes en format TIF. Naturellement, un apprentissage supervisé nécessite une répartition de la base de connaissance sur

deux bases: apprentissage et test. Pour la base d'apprentissage, on a extrait quatre exemplaires de trois individus, pour obtenir douze patrons d'apprentissages dont 4 à identifier et huit à écarter. Pour la phase de généralisation, en plus des patrons incluant les huit empreintes des sept individus écartés de l'apprentissage, la base contient aussi le reste des quatre exemplaires de trois individus sélectionnés pour l'apprentissage.

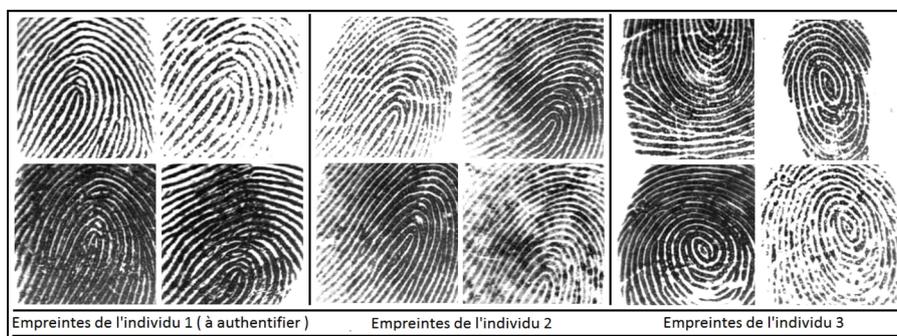


**Figure 4.11** - Exemples empreintes de la base DB1 de la FVC2004 des trois individus sélectionnés pour l'apprentissage

En résumé, on a sélectionné aléatoirement trois individus (voir figure.4.11) de la DB1 pour construire une base d'apprentissage de 12 empreintes l'entraînement du SNN soit 15 % de la base de connaissance et une base de test de 68 empreintes pour l'identification soit les 85% restant.

### 3.2 PRETRAITEMENT

Afin d'optimiser le codage d'information, on a planifié deux expériences sur la même base d'empreinte mais avec deux procédés de prétraitement d'image. Dans la première expérience, on applique une simple binarisation de l'image de l'empreinte, tandis que dans l'autre expérience, on effectue un prétraitement beaucoup plus avancé sur les empreintes. Chaque image passera par une phase de squelettisation avant de passer par la phase d'apprentissage afin de réduire non pas le nombre de paramètres d'entrées du SNN mais le flux d'information inutile à propager dans réseau.



**Figure 4.12** - Base d'apprentissage des empreintes digitales utilisées pour l'entraînement du SNN.

Dans ce sens, chaque image de la DB1 passe par une série de modification pour les préparer à la première et la deuxième expérience. Ainsi, on a commencé par convertir les images sous le format BMP. Ensuite, on a procédé à un rognage des images afin de minimiser leurs tailles et supprimer toutes informations inutiles; c'est-à-dire passer d'une image de 640x480 pixels à 200x200 pixels. Les images obtenues après rognage sont représentées dans la Figure 4.12.

Dans la DB1, initialement, les images sont en niveaux de gris, ce qui permet de passer directement à l'étape de binarisation ; cette étape est nécessaire pour les deux expériences car elle détermine le codage des entrées du SNN. On obtient des résultats comme ceux illustrés dans la figure 4.13.



**Figure 4.13** - Trois d'empreintes digitales après avoir subis conversion, rognage et binarisée.

Dans la première expérience, les images binarisées sont utilisées directement dans la phase d'apprentissage, tandis que pour la deuxième il a fallu passer par une autre étape de prétraitement qui est la « *squelettisation* ».

La squelettisation ou l'amincissement « *thinning algorithms* » est un processus de réduction des composants de l'image afin d'obtenir les informations les plus élémentaires. Sa particularité est sa capacité de reproduire l'image sans détruire le cadre de l'information qu'elle renferme à partir de sa forme originale [Shimizu et al., 2000].

Dans la littérature, on trouve différents d'algorithmes de squelettisation. Les algorithmes les plus utilisés sont l'algorithme de Rosenfeld, l'algorithme de Zhang Suen, l'algorithme Hilditch et l'algorithme Nagendraprasad-Wang-Gupta(NWG). L'algorithme Zhang Suen [Zhang et al., 1984] est connu pour être plus rapide que l'algorithme Hilditch [Lam et al., 1995]. Cependant, l'algorithme Hilditch est plus facile à mettre en œuvre [Adeline, 2005], donne de meilleurs résultats [Widiarti, 2011] et peut être utilisé pour atténuer divers types de motifs numériques.

La Figure 4.14 illustre les différentes étapes de l'algorithme de Hilditch. Il existe deux versions de cet algorithme, une version utilisant une fenêtre 4x4 pixels, et une autre avec une fenêtre 3x3 pixels ; on utilisera cette dernière version pour procéder à la squelettisation de l'ensemble des images de la base de connaissance.

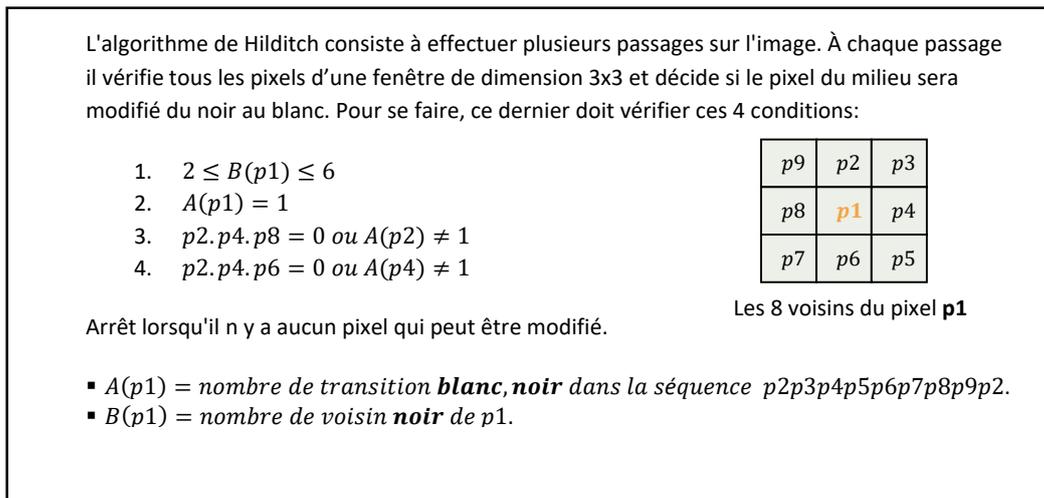


Figure 4.14 - Algorithme de squelettisation de Hilditch.

Après l'implémentation, on retrouve dans la Figure 4.15 les images binaires ainsi que le squelette de chaque empreinte combinant les deux opérations ; la binarisation et la squelettisation avec l'algorithme de Hilditch.



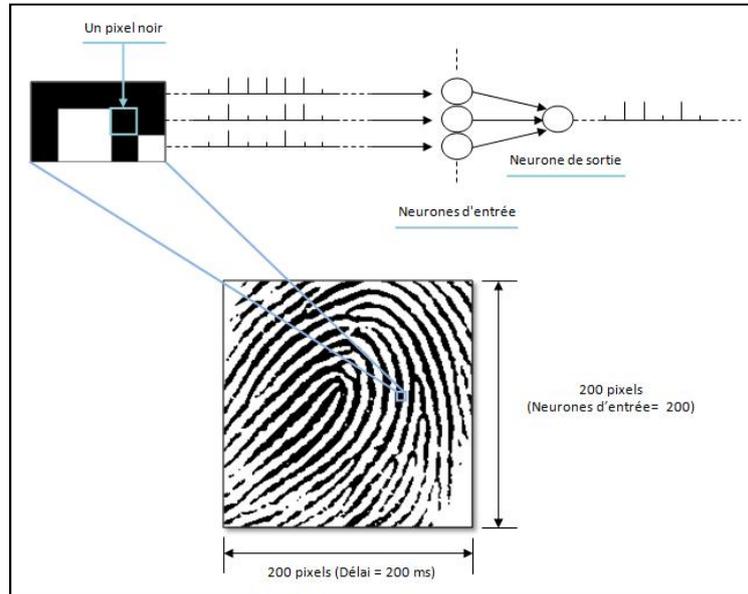
Figure 4.15 - Binarisation (image au milieu) et squelettisation (image à droite) d'une empreinte digitale à partir de l'image de l'empreinte (image à gauche).

### 3.3 PARAMETRES DU SNN

Pour garantir la rapidité de traitement de notre SNN, son architecture se doit d'être simple et d'avoir une structure sur mesure au problème surtout que les échantillons des empreintes digitales utilisées sont tous normalisés. Ainsi, il est facile d'imaginer un réseau monocouche pour l'identification des empreintes digitales.

D'une autre part, après le prétraitement des images, les données en entrées des deux expériences sont sous forme binaire. Les pixels noirs sont représentés par 1 et les blancs par 0. Ainsi, pour coder les données en entrée, on a choisi de prendre chaque ligne de l'image comme un neurone de la couche d'entrée du SNN (soit autant de neurones en entrée du réseau que de lignes dans l'image : 200 lignes = 200 neurones d'entrée). Chaque neurone est codé sur un train d'impulsions de 200 ms représentant le nombre de colonne pour chaque ligne où chaque pixel noir représente une impulsion (voir

Figure 4.16). Le délai est alors choisi comme étant le nombre de colonnes (200 colonnes = 200 ms).



**Figure 4.16** – Structure et codage utilisés dans le SNN pour l'identification des empreintes digitales.

En résumé, la Table 4.3 montre la structure finale du SNN monocouche [200 1] pour l'identification des empreintes digitales. Comme dans la section précédente, on a maintenus la même formulation des deux fonctions kernel pour produire le PA du neurone SRM et le même algorithme d'apprentissage. Pour les paramètres, les constantes  $\tau_m$ ,  $\tau_s$  et  $\tau_r$  ont été initialisées, respectivement, à 3, 4 et 10. Concernant le codage de la couche de sortie, on choisi un codage par phase en tenant compte uniquement de la première impulsion.

Réseau de neurones impulsif	Nombre de neurones en entrée	Nombre de sous-synapses par connexion	Nombre de neurones en sortie	Délais (ms)
SRM	200	3	1	200

**Table 4.3** – Paramètres du SNN pour l'identification des empreintes digitales.

### 3.4 EVALUATION ET DISCUSSIONS

Dans cette section, on va premièrement exposer les paramètres retenus et l'évolution de l'apprentissage du SNN des deux expériences abordées précédemment. Dans le même contexte, la première partie présente les résultats des tests sur des images binarisées et la seconde ceux d'images ayant subies une squelettisation.

Aussi, on a réalisé uniquement pour une base de généralisation de 12 empreintes les deux expériences mais cette fois sur des images bruitées à 10% puis à 30% par un bruit gaussien. Notre objective est de commencer à tester la robustesse du SNN dans ce cas d'école.

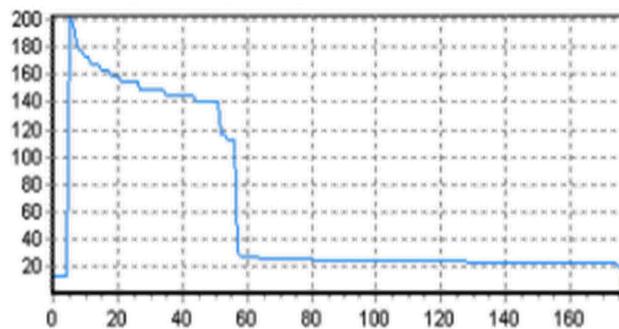
### 3.4.1 PREMIERE EXPERIENCE

Après plusieurs essais, la Table 4.4 résume les paramètres retenus du SNN pour la première expérience. Ces paramètres concernent le seuil du SRM et le pas d'apprentissage. Pour le codage de l'information, on a affecté la première phase à l'identification positive de l'empreinte digitale et l'autre phase à son rejet.

	Pas d'apprentissage ( $\eta$ )	Seuil	Phase 1	Phase 2	Temps désiré ( $\hat{t}_j^{(1)}$ )	
					Phase 1	Phase 2
SNN	0.001	90	[0, 17[	[17, 200]	10	20

**Table 4.4** - Paramètres du SNN retenus pour la première expérience.

Durant l'apprentissage, on a remarqué une amélioration dans le temps calculé de la première impulsion à partir de 80 itérations. La Figure 4.17 montre l'évolution de l'erreur quadratique moyenne en fonction de ces 180 itérations où on obtient un taux d'apprentissage de 100%.



**Figure 4.17** - Graphe de variation de l'erreur quadratique moyenne durant la première expérience.

Pour tester la performance du réseau de neurones impulsionnel et les résultats de l'apprentissage retenu, on est passé à la phase de généralisation.

Pour une meilleure représentation des résultats, on a calculé deux taux d'erreurs. Le premier est le taux des exemples négatifs mal classé ou le False Match Rate (FMR) aussi connu par False Acceptance Rate (FAR). Ce taux détermine la probabilité pour un système de reconnaître une personne qui normalement n'aurait pas dû être reconnue (voir l'équation 4.1). Le second est le taux des exemples positifs mal classés ou le False Non Match Rate (FNMR) connu aussi par False Rejection Rate (FRR). Ce taux

détermine la probabilité pour un système donné de ne pas reconnaître une personne qui normalement aurait dû être reconnue (voir l'équation 4.2).

$$FMR = \frac{N_h}{N_{Hors}} \times 100 \quad (4.1)$$

Où  $N_h$  est le nombre des empreintes étrangères classées a tort comme des empreintes identifiées et  $N_{Hors}$  est le nombre total des empreintes étrangères.

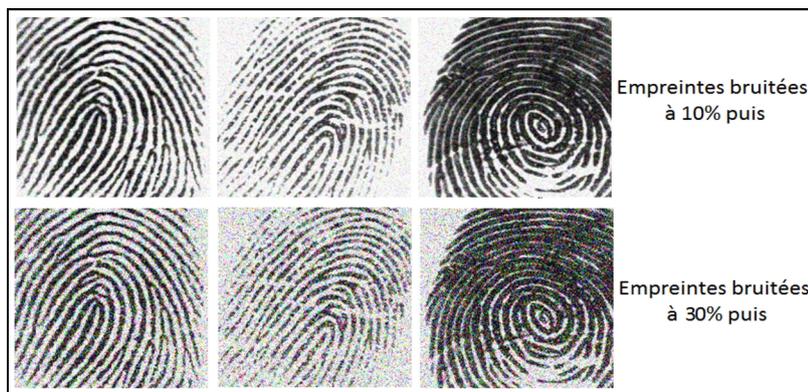
$$FNMR = \frac{N_i}{N_{Identif}} \times 100 \quad (4.2)$$

Où  $N_i$  est le nombre des empreintes à identifier classées a tort comme des empreintes étrangères et  $N_{Identif}$  est le nombre total des empreintes à identifier.

	Taux de reconnaissance (%)	FMR (%)	FNMR (%)	EER(%)
SNN	86,76	14,06	0	13,23

**Table 4.5** - Evaluation des résultats du SNN sur l'identification des empreintes digitales sur des images binarisées.

La Table 4.5 illustre les résultats de la phase de généralisation. Le taux de reconnaissance et le taux d'erreur (EER : Equal Error Rate) en plus du FMR et FNMR. On a remarqué que le SNN avait plus tendance à identifier l'empreinte avec un FNMR à 0%. Malheureusement, cette faculté lui fait défaut car 9 empreintes ont été reconnues malgré qu'elles représentent des empreintes étrangères. Ainsi, le FMR arrive à la valeur moyenne de 14,06%.



**Figure 4.18** - Empreintes digitales binarisées et bruitées à 10% et à 30% par une distribution Gaussienne.

Le deuxième test est sur les mêmes empreintes que celle apprises par le réseau mais bruitées à 10% puis à 30% en utilisant un filtre Gaussien (voir figure 4.18). La Table 4.6 montre les résultats obtenus comparés avec les résultats des images non-bruitées. Il faut préciser que la taille de la base de reconnaissance pour cette partie est réduite à 12 image seulement.

SNN	Taux de reconnaissance (%)	FMR (%)	FNMR (%)	EER(%)
Images binarisées	100	0	0	0
Images binarisées bruitées à 10%	91,66	0	25	8,33
Images binarisées bruitées à 30%	83,33	12,5	25	16,66

**Table 4.6** - Evaluation des résultats du SNN sur l'identification des empreintes digitales sur des images binarisées et bruitées à 10% et à 30%.

On peut observer à travers les taux de reconnaissances et d'erreur que le SNN est robuste dans l'identification des empreintes digitales binarisées et que sa capacité d'écarter une empreinte étrangère est assez importante à travers le FMR qui reste à 0% pour un bruit de 10% et passe à 12,5% pour un bruit de 30%. Sa faculté à exclure une empreinte correcte elle reste stable avec un FNMR à 25%, ceci représente une seule empreinte correcte jugée comme étrangère.

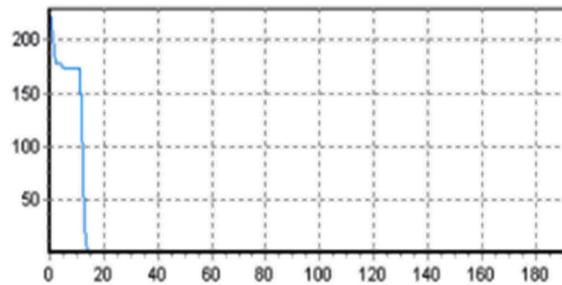
### 3.4.2 DEUXIEME EXPERIENCE

Comme pour la première expérience, on a procédé à plusieurs lancements pour valider une configuration finale pour notre SNN. Les paramètres retenus sont illustrés dans la Table 4.7. En résumé, pour la deuxième expérience, les paramètres concernent le seuil du neurone SRM initialisé à 30 et le pas d'apprentissage fixé à 0,001. Pour le codage de l'information, comme dans la première expérience, la première phase de [0,17[ms est affectée à l'individu à identifier et la deuxième phase de [17,200] ms à tous individus étrangers.

	Pas d'apprentissage ( $\eta$ )	Seuil	Phase 1	Phase 2	Temps désiré ( $\hat{t}_j^{(1)}$ )	
					Phase 1	Phase 2
SNN	0.001	30	[0, 17[	[17, 200]	10	20

**Table 4.7** - Paramètres du SNN retenus pour la deuxième expérience.

Durant l'apprentissage, on a remarqué une amélioration dans les résultats dès les 20 premières itérations mais l'évolution du taux d'apprentissage passée de 91,67% dans la 20<sup>ème</sup> itération à 83,33% dans la 100<sup>ème</sup> itération. La Figure 4.19 montre l'évolution de l'erreur quadratique moyenne en fonction de ces 180 itérations où on obtient un taux d'apprentissage de 100%.



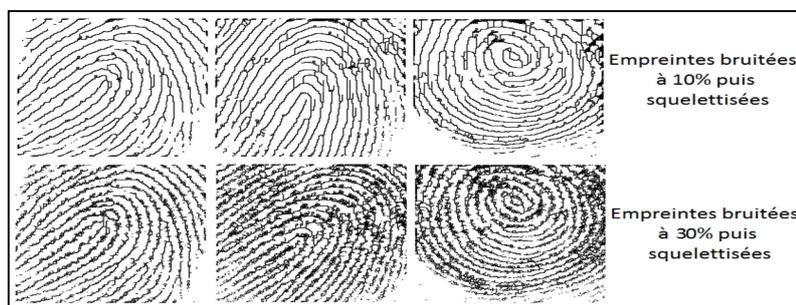
**Figure 4.19** - Graphe de variation de l'erreur quadratique moyenne durant la deuxième expérience.

Pour tester la performance du réseau de neurones impulsionnels, les résultats sont toujours représentés par False Match Rate (FMR) et le False Non Match Rate (FNMR)

	Taux de reconnaissance (%)	FMR (%)	FNMR (%)	EER(%)
SNN	73,52	28,12	0	26,47

**Table 4.8** - Evaluation des résultats du SNN sur l'identification des empreintes digitales sur des images squelettisées.

La Table 4.8 illustre les résultats de la phase de généralisation. Le taux de reconnaissance et le taux d'erreur (EER : Equal Error Rate) sont inférieur à ceux obtenus par la binarisation des images. Naturellement, c'est le mêmes cas pour le FMR et le FNMR. On a remarqué que le SNN avait aussi dans cette partie de l'expérimentation tendance à plus identifier l'empreinte avec un FNMR à 0%. Malheureusement, cette faculté lui fait défaut car 18 empreintes ont été reconnues malgré qu'elles représentent des empreintes étrangères. Ainsi, le FMR arrive à la valeur moyenne de 28,12% cette fois.



**Figure 4.20** - Empreintes digitales squelettisées et bruitées à 10% et à 30% par une distribution Gaussienne.

Pour suivre la même procédure, la Figure 4.20 montre les mêmes empreintes squelettisées utilisées pour l'apprentissage du SNN mais bruitées à 10% puis à 30% en

utilisant un filtre Gaussien (voir figure 4.20). La Table 4.9 montre les résultats obtenus comparé avec les résultats des images squelettisées non-bruitées. Il faut préciser que la taille de la base de reconnaissance pour cette partie est réduite à 12 images seulement.

SNN	Taux de reconnaissance (%)	FMR (%)	FNMR (%)	EER(%)
Images squelettisées	100	0	0	0
Images squelettisées bruitées à 10%	91,66	12,5	0	8,33
Images squelettisées bruitées à 30%	75	37,5	0	25

**Table 4.9** - Evaluation des résultats du SNN sur l'identification des empreintes digitales sur des images squelettisées et bruitées à 10% et à 30%.

On remarque, qu'avec un FNMR toujours égale à 0, le SNN a plus tendance à reconnaître l'empreinte à identifier avec des images squelettisées même en augmentant le bruit. Cependant, il perd toute sa faculté de distinguer des empreintes étrangères dès qu'on augmente le bruit.

Finalement, des deux expériences, on a remarqué, que bien qu'il converge moins rapidement vers une solution pendant l'apprentissage, le SNN identifie mieux des images binarisées que squelettisées. Les taux de reconnaissances et d'erreurs confirment cette conclusion ainsi que les scores du FMR et FNMR observées durant les deux expériences. Le SNN montre aussi dans ce cas là une meilleure robustesse au bruit. Cette conclusion ne peut que servir l'évaluation de la performance du SNN à base de SRM dans l'identification des empreintes digitales car moins il y aura de traitement plus le temps de réponse du SNN sera réduit.

## 4 RECONNAISSANCE LABIALE DES MOTS ISOLÉS SUR TULIPS

### 1.0

Dans cette section, on a décidé d'élargir notre expérimentation en allant cette fois vers une des nombreuses techniques de la reconnaissance de la parole: La reconnaissance labiale des mots isolés. Eventuellement, la reconnaissance labiale ou audio-visuelle n'est pas nécessairement la technique la plus convoitée pour réaliser un système de reconnaissance automatique de la parole. Néanmoins, on trouve dans la littérature un cas d'école très utilisé et qui a déjà été traité par les réseaux de neurones impulsifs [booi, 2004].

Chiffres prononcés	Nombre d'individus	Nombre de répétitions	Énoncés	Nombre total d'images	Taille de l'image
« one », « two », « three » et « four »	12	2	96	934	100 × 75

**Table 4.10** - Description de la base Tulips 1.0

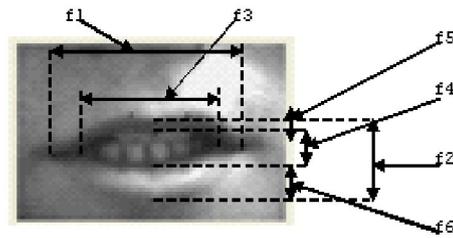
La base Tulips 1.0 est élaborée en 1995 sous la direction du Dr. Javier R. Movellan, maître-assistant au sein du département des Sciences Cognitives et chercheur à la tête du laboratoire MPLab (Machine Perception Laboratory) de l'Institut de calcul neuronal à l'Université de San Diego de Californie aux États Unis (voir Table 4.10). Tulips 1.0 est une petite base de données de l'audiovisuel de 12 personnes prononçant les quatre premiers chiffres en anglais : « one », « two », « three » et « four ». L'acquisition des énoncés a été répétée deux fois par chacun des 12 locuteurs dont 9 hommes et 3 femmes, au total: 96 énoncés. Chaque énoncé se compose de 6 à 16 séquences d'images. La base de données a été numérisée en niveau de gris à une résolution de 100×75 pixels à 30 frames/sec à 8 bits (disponible sur le site: <http://cogsci.ucsd.edu/~movellan/>). La Figure 4.21 illustre des les séquences d'images qu'on peut trouver sur la base de données.



**Figure 4.21** - Exemples de séquences d'images de la base Tulips: (A): Séquences de 7 images montrant une femme prononçant le « One », (B): Séquences de 8 images montrant un homme prononçant le « Two »

Dans notre implémentation, on essaie d'optimiser la topologie du SNN en offrant une nouvelle stratégie de codage basée sur un codage par ordre de rang. Pour une meilleure comparaison des résultats, on a utilisé des paramètres disponibles sur la base de données Tulips 1.0. Ces paramètres arrivent avec chaque séquence d'image et décrivent des caractéristiques notées par  $f_i$  où  $i = 1..6$  qui représentent les distances suivantes (voir Figure 4.22):

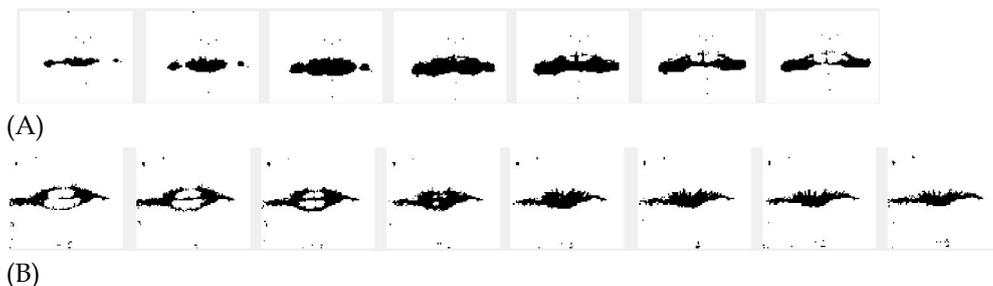
- 1) La largeur de lèvres.
- 2) L'hauteur de lèvres.
- 3) La largeur d'ouverture de la bouche.
- 4) L'hauteur d'ouverture de la bouche.
- 5) L'hauteur de lèvre inférieure.
- 6) L'hauteur de lèvre supérieure.



**Figure 4.22** - Six caractéristiques de l'ouverture de la bouche

## 4.1 EXPERIMENTATION DU SNN SUR RECONNAISSANCE LABIALE DES MOTS ISOLES

En utilisant la base de données Tulips 1.0, les travaux de Booi se sont basés sur un seuillage des séquences des images de taille 100x75 pixels pour transformer l'entrée analogique en binaires (voir Figure 4.23). Cette spatialisation temporelle des séquences d'images obtenue grâce à la variation du niveau de gris de chaque pixel de l'image durant le mouvement des lèvres permet effectivement d'obtenir un codage en train d'impulsions logiques. Par ailleurs, ce codage affecte directement la structure du réseau où le nombre de neurones d'entrées est important car il est déterminé par la taille de l'image, c'est-à-dire le nombre de neurones d'entrées pour ce cas est de 7500 neurones.



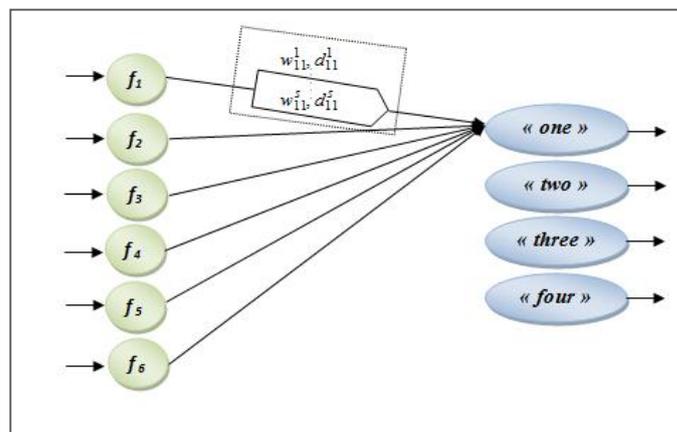
**Figure 4.23** - Exemples de séquences d'images de la base Tulips ayant subi un seuillage: (A): Séquences de 7 images montrant une femme prononçant le « One », (B): Séquences de 8 images montrant un homme prononçant le « Two »

Dans les deux sections précédentes, en se basant sur notre étude expérimentale, on avait remarqué qu'un prétraitement des entrées d'un SNN n'optimisait en rien le rendement des ces résultats. Cependant, la réduction des nombres de neurones en entrées du SNN semble renforcer sa performance et alléger sa structure. Ainsi, on a choisi d'utiliser uniquement l'évolution des six caractéristiques décrit dans la Figure 4.22 au lieu des séquences d'images.

Finalement, pour lancer notre expérimentation, la base de connaissance a été divisée en deux parties égales: 48 séquences pour la base d'apprentissage et 48 séquences pour la base test.

## 4.2 PARAMETRES DU SNN

Dans cette partie de l'expérimentation, on a construit un SNN monocouche avec six neurones dans la couche d'entrées et quatre neurones dans la couche de sortie (voir Figure 4.24).

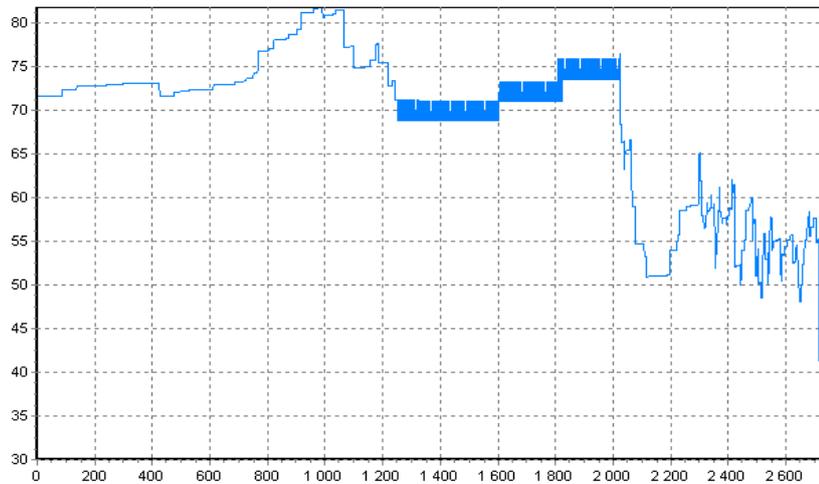


**Figure 4.24** - SNN monocouche [6 4] pour la reconnaissance labiale des mots isolés pour la reconnaissance labiale des mots isolés sur Tulips 1.0.

Les six neurones en entrées représenteront ainsi les six caractéristiques des mouvements des lèvres et les neurones de la couche de sortie retourneront la réponse du système, c'est-à-dire le chiffre prononcé: « one », « two », « three » et « four ». Pour les paramètres du SRM, les constantes  $\tau_m$ ,  $\tau_s$  et  $\tau_r$  ont été initialisées, respectivement, à 4, 2 et 20. Le seuil de chaque neurone est initialisé à 2. Concernant l'apprentissage, le codage des entrées est par ordre de rang et la sortie calculée se détermine en comparant le temps de la première impulsion des quatre neurones de la couche de sortie. Ainsi, le premier neurone à avoir émis une impulsion est la réponse désignée par le SNN.

## 4.3 EVALUATION ET DISCUSSIONS

Après avoir construit la topologie du SNN, on a procédé à plusieurs lancements de l'apprentissage pour choisir le reste des paramètres. L'avantage de ce réseau est la dimension de sa structure qui permet une propagation rapide des exemples de la base d'apprentissage et par conséquent une optimisation de la durée d'apprentissage. Ainsi, on a initialisé le pas d'apprentissage  $\eta$  à  $10^{-5}$ .



**Figure 4.25** - Graphe de variation de l'erreur quadratique du SNN pour la reconnaissance labiale des mots isolés sur Tulips 1.0.

La Figure 4.25 montre la convergence de l'erreur pendant l'apprentissage. Après plusieurs lancements, on a remarqué que dès les 50 premières itérations, on arrive à des taux d'apprentissage voisinant les 68,9% jusqu'à arriver au finale à 87,5%. La Table 4.11 montre les meilleurs résultats donnés par l'apprentissage du SNN.

Apprentissage	« one »	« two »	« three »	« four »	Total
Nombre total des séquences	12	12	12	12	48
Nombre reconnu des séquences	12	11	10	9	42
Taux d'apprentissage (%)	100	91,66	83,33	75	87,5

**Table 4.11** - Evaluation des résultats de l'apprentissage du SNN pour la reconnaissance labiale des mots isolés sur Tulips 1.0.

La phase de généralisation montre un taux de reconnaissance entre 64,58% et 75%. Dans la Table 4.12, on présente le meilleur score du taux de généralisation donnés par SNN.

Apprentissage	« one »	« two »	« three »	« four »	Total
Nombre total des séquences	12	12	12	12	48
Nombre reconnu des séquences	12	8	9	7	36
Taux d'apprentissage (%)	100	66,66	75	58,33	75

**Table 4.12** - Evaluation des résultats de reconnaissance du SNN pour la reconnaissance labiale des mots isolés sur Tulips 1.0.

La phase expérimentale a reconfirmé nos remarques précédentes concernant l'instabilité de l'algorithme de la descente du gradient. Néanmoins, Les résultats obtenus dans la phase de généralisation sont très encourageant, arrivant à 75% de taux de reconnaissance.

En résumé, on peut dire qu'il a suffi de changer la stratégie de codage sur de nouvelles caractéristiques en entrées pour améliorer le taux de généralisation de plus de 20%. Sachant que dans les travaux de [Booij, 2004] utilisant le SNN pour la même base de données, n'avaient pas dépassé les 57% dans la phase de généralisation.

Comparé à d'autres travaux, bien qu'ils n'ont pas été fait sur la même base de données, Les résultats obtenus avec le SNN dépassent de loin leurs résultats. Par exemple, le modèle de Markov caché (HMM) testé sur la base AVICAR a donné un taux de reconnaissance de 50% utilisant lui aussi les formes extrait de l'image [Fu et al., 2007]. Plus récemment, une variante du HMM, appelé Ergodic HMM [Sujatha et al., 2010], a donné un taux de reconnaissance de 66.83%. Cette application a été testée sur la base Mugshot et combine à l'image avec la forme. Enfin, les SVMs testé sur les images de la base AV letters donne 62.80 % [Zhao et al., 2009].

## 5 DIAGNOSTIC DU CANCER DE SEIN SUR LA BASE DE DONNEES WBCD

Dans cette section, on a décidé de nous intéresser de près à un benchmark très utilisé qui est le diagnostic du cancer du sein sur la Wisconsin Breast Cancer Database (WBCD). Actuellement, cette maladie fatale, qui représente le pourcentage de mortalité prématurée le plus élevé chez les femmes, n'a pas laissé la communauté scientifique indifférente quant à l'automatisation du diagnostic de cette pathologie pour réussir son dépistage à l'avance. Surtout que le diagnostic peut permettre de détecter la nature de la tumeur: « *maligne* » ou « *bénigne* ». Dans la littérature, on retrouve beaucoup de travaux à base de réseaux de neurones artificiels traitant ce problème avec cette même base de données [Setiono R., 2000 ; Kiyani et al., 2004 ; Karabatak et al., 2009; McKennoch et al., 2009 ; Mohamed et al., 2011 ; Marcano-Cedeño et al., 2011].

La WBCD a été obtenue par l'Université du Wisconsin. Elle contient les informations médicales de 699 cas cliniques relatifs au cancer du sein classé comme bénin ou malin (voir Table 4.13) : 458 patientes (soit 65.5%) sont des cas bénins et 241 patientes (soit 34.5%) sont des cas malins.

	Attribut	Domaine
1	Sample code number	Numéro de l'identificateur
2	Clump Thickness	1 - 10
3	Uniformity of Cell Size	1 - 10
4	Uniformity of Cell Shape	1 - 10
5	Marginal Adhesion	1 - 10
6	Single Epithelial Cell Size	1 - 10
7	Bare Nuclei	1 - 10
8	Bland Chromatin	1 - 10
9	Normal Nucleoli	1 - 10
10	Mitoses	1 - 10
11	Class	(2 pour bénin, 4 pour malin)

**Table 4.13** - Description des attributs de la Wisconsin breast cancer database (WBCD).

La base de données contient 16 données manquantes; les patientes sont caractérisées par 11 attributs : le premier fait référence à l'identificateur de la patiente et le deuxième représente la classe : le diagnostic est de 2 si le cas est bénin, 4 si le cas est malin quant aux 9 autres, ils représentent des cas cliniques suivants:

- 1) **Clump Thickness**: l'épaisseur de la membrane plasmique d'une cellule cancéreuse est plus importante que celle d'une cellule normale.
- 2) **Uniformity of Cell Size**: les cellules cancéreuses sont caractérisées par une anisocytose, à savoir une inégalité au niveau de la taille par comparaison avec les cellules saines.
- 3) **Uniformity of Cell Shape**: les cellules cancéreuses sont marquées par des contours irréguliers ainsi que des incisures.
- 4) **Shape Marginal Adhésion**: une surexpression de la protéine intégrine beta3 au niveau de la surface de la cellule cancéreuse.
- 5) **Single Epithelial Cell Size**: étant donné que les cellules épithéliales sont absentes à l'état naturel au niveau de la moelle osseuse et qu'elles ne sont pas détectées chez les individus sains, la moelle osseuse peut, de ce fait, être considérée comme un indicateur de maladie métastatique chez les patients atteints du cancer du sein au stade primaire.
- 6) **Bare Nuclei**: à l'état normal, les nucléoles se trouvent à l'intérieur du noyau. Dans le cas où ses derniers se trouvent confondus avec le cytoplasme cela indique que la cellule présente une anomalie et qu'elle est susceptible de devenir cancéreux.

- 7) **Bland Chromatine:** H2az est une protéine qui induit l'expression du gène du récepteur d'œstrogènes. La surproduction de cette protéine est un marqueur de présence de cellules cancéreuses au niveau du sein étant donné qu'elles sont hormono-dépendantes.
- 8) **Normal Nucléole:** L'ADN est naturellement protégé par une membrane nucléaire. Une défaillance observée au niveau de cette membrane peut refléter une croissance tumorale.
- 9) **Mitoses :** La mitose est un processus de division cellulaire régulé permettant de reproduire des cellules filles génétiquement identiques à la cellule parentale.

	id	diagnosis	thickness	size.unif	shape.unif	adhesion	epi	bare	bland	normal	mitoses
1	1000025	2	5	1	1	1	2	1	3	1	1
2	1002945	2	5	4	4	5	7	10	3	2	1
3	1015425	2	3	1	1	1	2	2	3	1	1
4	1016277	2	6	8	8	1	3	4	3	7	1
5	1017023	2	4	1	1	3	2	1	3	1	1
6	1017122	4	8	10	10	8	7	10	9	7	1

**Figure 4.26-** Exemples extraits de la Wisconsin breast cancer database (WBCD).

En résumé, la Figure 4.26 montre quelques exemples extraits de la WBCD. Pour construire notre SNN, on remarque que le premier champ est inutile pour notre partie expérimentale puisque il ne représente qu'une immatriculation du patient. Le second champ indique le diagnostic confirmé du patient donc la réponse désirée du système. Ainsi, ce champ est élémentaire pour construire une classification supervisée. Le reste des données représentent les caractéristiques de l'échantillon en entrée et sont tous des entiers ayant des valeurs situées entre l'intervalle [1,10].

Nous accorderons une importance particulière à ce cas d'école car la nature de ce problème ressemble beaucoup à l'objective principale de cette thèse qui est la détection de l'activité de la voix. En effet, dans les deux systèmes ont comme entrées un nombre très limité de caractéristiques et leurs uniques neurones de sortie retourneront soit une réponse positive soit une réponse négative.

## 5.1 EXPERIMENTATION DU SNN SUR LE DIAGNOSTIC DU CANCER DU SEIN

Dans ce qui suit, on décrit les propriétés du SNN pour l'aide au diagnostic médical sur la base WBCD. Ainsi, on présente le type de codage appliqué sur les données de la base et l'architecture du SNN utilisé ainsi que les différentes étapes par lesquelles on est passé pour le réaliser. Dans cette expérimentation, on a décidé de varier le type de codage. Par conséquent, on a tracé cette partie de l'expérimentation sur deux parties: la première expérience utilise un codage par le temps d'impulsion [Ourdighi et al., 2016] et la deuxième expérience est basé sur un codage à population de neurone utilisant des

champs réceptifs Gaussien (CRG). Ainsi, suivant l'expérience, chaque SNN nécessitera sa propre structure et ses propres paramètres.

Pour la base de connaissance, on l'a divisé en deux parties où 60% des échantillons seront destinés à l'apprentissage et les 40% restant seront utilisés pour la phase de généralisation. La Table 4.14 illustre clairement cette répartition.

Echantillon	Apprentissage (60%)	Test (40%)	Total
Malin	144	95	239
Bénin	266	178	444
Total	410	273	683

**Table 4.14** - Répartition des échantillons de la Wisconsin breast cancer database (WBCD) entre base d'apprentissage et base de test.

Dans la base de données, il existe 16 échantillons incomplets dont 2 classés malin et 14 classés bénin. Sachant que dans ces échantillons manque la septième caractéristique « *bare nuclei* », on peut les considérer comme des données bruitées.

## 5.2 EVALUATION DU SNN DANS LE DIAGNOSTIC DU CANCER DU SEIN

Dans cette partie de l'expérimentation, le SNN se charge de fournir un système d'aide à la décision pour le diagnostic médicale sur WBCD. Pour une meilleure représentation des résultats, on a calculé deux taux d'erreurs qui sont : le taux des exemples négatifs mal classé ou le False Accept Rate (FAR) et le taux des exemples positifs mal classés ou le False Reject Rate (FRR).

$$FAR = \frac{N_M}{N_{Malin}} \times 100 \quad (4.3)$$

Où  $N_M$  est le nombre de trames de tumeurs malignes classées à tort comme de tumeurs bénignes et  $N_{Malin}$  est le nombre total des trames de tumeurs malignes.

$$FRR = \frac{N_B}{N_{Bénin}} \times 100 \quad (4.4)$$

Où  $N_B$  est le nombre de trames de tumeurs bénignes classées a tort comme du silence et  $N_{Bénin}$  est le nombre total des trames de tumeurs bénignes.

Pour une meilleure d'évaluation, on a aussi utilisé le Receiver Operating Characteristics (ROC). Les graphes et les paramètres obtenus par ROC sont une technique utile pour organiser et visualiser la performance d'un système d'aide à la décision [Fawcett , 2003]. Plus précisément, la courbe AUC (Area Under ROC Curve) est le paramètre le

plus intéressant pour mesurer la précision des tests. Cette mesure est censée être entre 0 et 1 et indique la fiabilité du diagnostic dans la phase de généralisation: Plus l'AUC est proche du 1 mieux c'est pour la performance du SNN.

Ensuite, on utilisera les paramètres Sensitivité et Spécificité qui servent à évaluer la performance du réseau. Les équations (4.5) et (4.6) illustrent la formulation de ces deux paramètres.

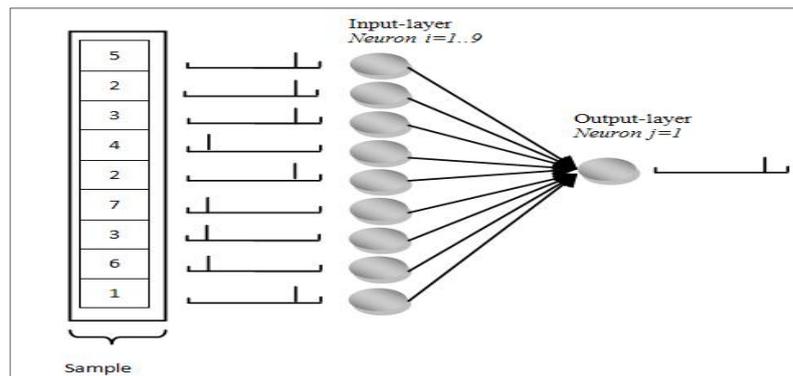
$$\text{Sensitivité} = \frac{TP}{TP+FN} \quad (4.5)$$

$$\text{Spécificité} = \frac{TN}{FP+TN} \quad (4.6)$$

- TP (True Positive): Une tumeur a été détectée maligne, ce qui coïncide avec l'avis du médecin.
- TN (True Négative): Une tumeur a été détectée bénigne, ce qui coïncide avec l'avis du médecin.
- FP (False Positive) : Une tumeur a été détectée maligne contrairement à l'avis du médecin.
- FN (False Négative) : Une tumeur a été détectée bénigne contrairement à l'avis du médecin.

### 5.3 PREMIERE EXPERIENCE: APPRENTISSAGE, TEST ET EVALUATIONS

Pour un codage des entrées par le temps d'impulsion, l'architecture du SNN est basée sur un réseau monocouche [9 1] ayant neuf neurones dans la couche d'entrée représentant chaque caractéristique du patron de l'apprentissage de la base de données WBCD. La couche de sortie est représentée par un neurone qui détermine le diagnostic : malin ou bénin. La Figure.4.27 montre la topologie du réseau. Le codage de la sortie est déterminé par le codage par phase, c'est-à-dire si le neurone émet sa première impulsion dans l'intervalle [0, 10] ms on considère le cas comme malin et s'il l'émet pendant l'intervalle ] 10, 20] ms. La durée des trains d'impulsion est fixée à 20 ms.



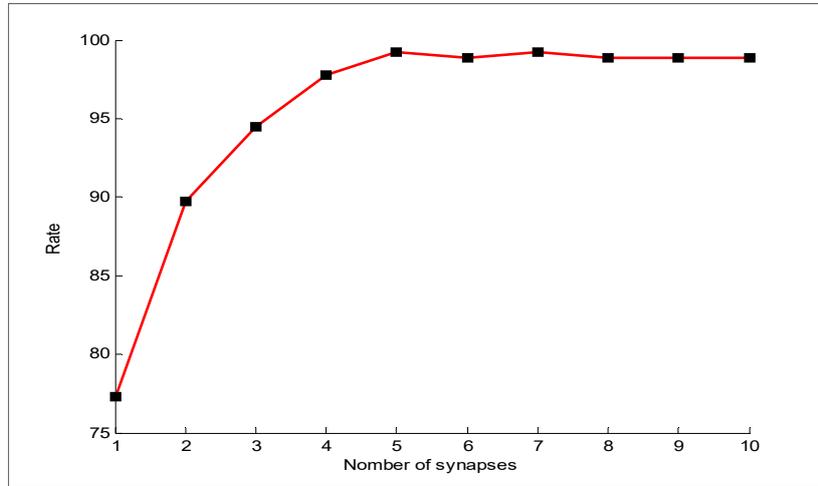
**Figure 4.27** - Topologie du SNN monocouche [9 1] pour le diagnostic du cancer du sein sur WBCD.

Pendant l'apprentissage, on a forcé le neurone de sortie à émettre une impulsion à l'instant  $t = 2 \text{ ms}$  si le patron d'apprentissage est malin et à  $t = 20 \text{ ms}$  s'il est bénin. Aussi, on a déjà expliqué dans la section 3.4.1, nous utilisant un SRM à connexions retardées. Par conséquent, chaque neurone de la couche d'entrée est connecté au neurone de sortie avec  $l$  sous-synapses. Dans cette partie de l'expérimentation, ce paramètre a montré un grand effet sur l'amélioration de la précision de classification. Les valeurs testées de  $l$  ont été choisies entre 1 et 10 retards.

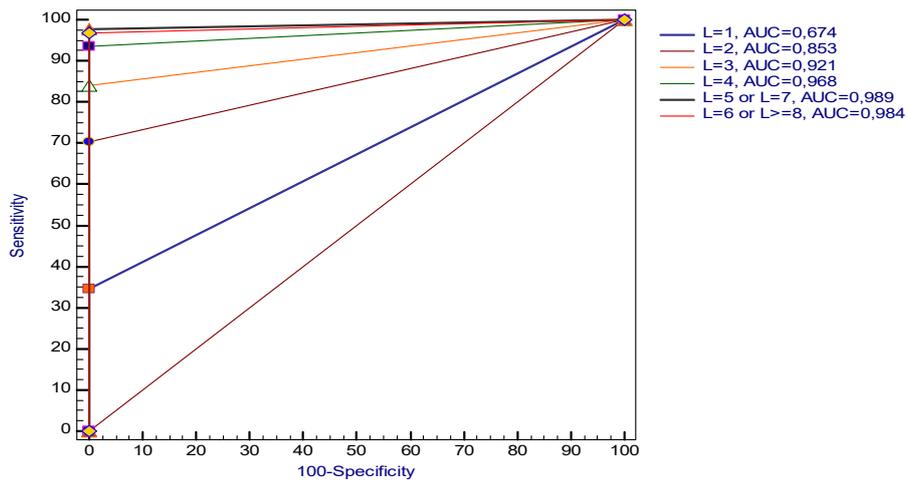
Pour la fonction à réponse impulsionnelle  $\varepsilon$ , on a choisi de varier la valeur des paramètres  $\tau_m$  et  $\tau_s$  entre 2 et 10 en gardant toujours  $\tau_m$  supérieur à  $\tau_s$ . On a remarqué que les différentes valeurs testées de ces deux paramètres avaient affectés la performance du SNN plus que le paramètre  $\tau_r$  de la fonction de réfractaire  $\eta$ . Par conséquent, on a choisi de fixer sa valeur à 10. Les poids synaptiques des connexions ont été initialisés aléatoires dans l'intervalle  $[0, 0.5]$ . Le pas d'apprentissage est initialement fixé à  $10^{-3}$ .

Dans la phase d'apprentissage, on a lancé plusieurs configurations basées sur: le nombre de sous-synapses entre deux neurones et les paramètres  $\tau_m$  et  $\tau_s$ . L'expérimentation a démontré que ces paramètres étaient les plus influents sur la performance du SNN. La variation de  $l$  a contrôlé directement l'évolution du taux de reconnaissance. Les paramètres  $\tau_m$  et  $\tau_s$  ont affecté la sensibilité et la spécificité du SNN.

Pour les résultats, on a remarqué qu'en variant le nombre des sous-synapses  $l$  entre 1 et 10 délais pour chaque configuration, le taux de reconnaissance atteint facilement 98,90%. On observe que le taux de reconnaissance optimal se démarque lorsque  $l$  se situe entre 5 et 7 délais où on arrive à 99,26%. La Figure 4.28 et Figure 4.29 montrent, respectivement, la variation du taux de reconnaissance et de la courbe AUC du ROC curves.



**Figure 4.28** - Evolution du taux de reconnaissance en variant les synapses  $l=1..10$  ( $\tau_m = 5$  and  $\tau_s = 3$  )

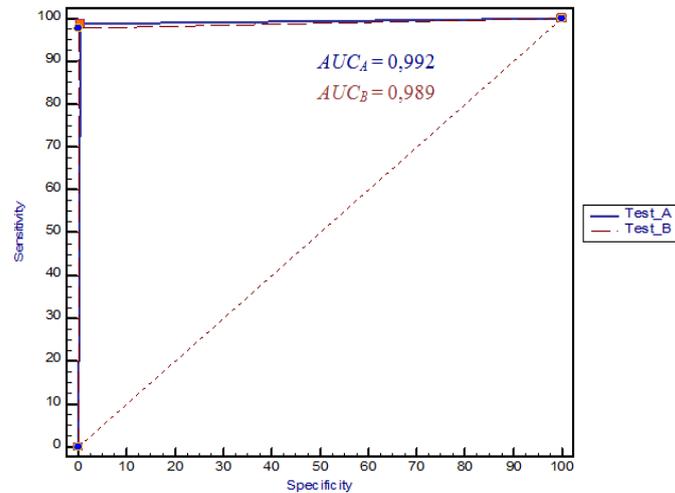


**Figure 4.29** - Différents graphes de la courbe AUC en variant les synapses  $l=1..10$  ( $\tau_m = 5$  and  $\tau_s = 3$  )

Par ailleurs, en variant les paramètres  $\tau_m$  et  $\tau_s$ , on doit toujours garder  $\tau_m$  supérieur à  $\tau_s$ . On a remarqué que si la différence entre ces deux paramètres est supérieure à 2 le SNN montre plus de sensibilité et moins de spécificité, c'est-à-dire que la sensibilité arrive facilement à 100% et la spécificité ne dépasse pas 97,89%.

La Figure 4.30 montre la courbe du ROC curves de deux configurations de SNN ayant atteint un taux de reconnaissance égale à 99.26 %. Les deux tests sont :

- Test A: où  $\tau_m = 10$  et  $\tau_s = 5$ , le  $AUC_A = 0,992$ .
- Test B: où  $\tau_m = 5$  et  $\tau_s = 3$  le  $AUC_B = 0,989$ .



**Figure 3.30** – Graphes AUC du ROC curves des Test A et Test B ayant donnés un taux de reconnaissance de 99,26%.

Configuration	Test A	Test B
$\tau_m$	10	5
$\tau_s$	5	3
$l$	7	7
Echantillon malin	94	93
Echantillon bénin	177	178
Sensitivité	98,94%	97,89%
Spécificité	99,43%	100%
AUC	0,992	0,989
Taux de Classification	99.26%	99.26%

**Table 4.15** – Evaluations des Test A et Test B structures optimales du SNN dans le diagnostic médicale sur WBCD.

La table 4.15 montre l'influence de la variation des paramètres  $\tau_m$  et  $\tau_s$  sur la sensibilité et la spécificité.

Concernant les 16 échantillons avec les valeurs manquantes, on les a testé le SNN. Par conséquent, on a obtenu un taux de reconnaissance de 87,5%, une spécificité égale à 85,71% et une sensibilité optimale égale à 100%. Le SNN a atteint le même taux de reconnaissance (99,26%) si tous les échantillons avaient la même caractéristique manquante. En outre, le SNN a montré une robustesse, en obtenant le même taux de classification, si chaque caractéristique manqué une par une dans la phase de généralisation. Aussi, le SNN renvoie la même performance avec uniquement 77,77% des caractéristiques.

## 5.4 DEUXIEME EXPERIENCE: APPRENTISSAGE, TEST ET EVALUATIONS

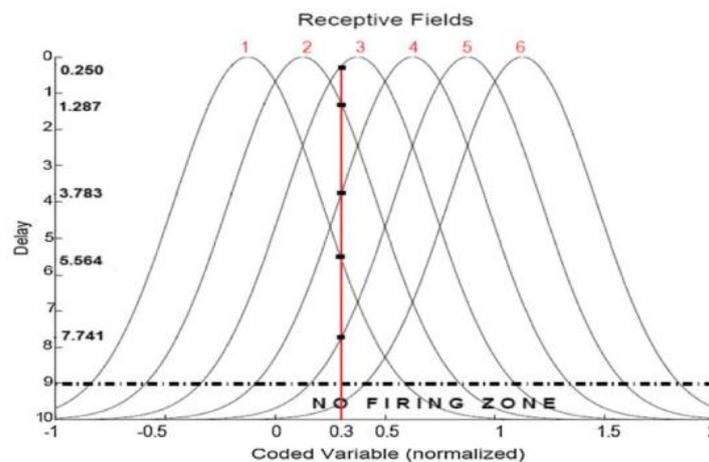
Dans la deuxième partie de l'expérimentation, l'architecture du SNN est basée sur un réseau monocouche [ $9 \times (\text{Nbr\_CRG} - 1)$ ] (Nbr\_CRG est le nombre des champs réceptifs gaussien). Afin de convertir les entrées numériques en train d'impulsions, on a utilisé un codage à base de population de neurones qui consiste à modéliser chaque neurone d'entrée par un champ réceptif gaussien (CRG). Ainsi, la structure du SNN dépend de l'initialisation du nombre du CRG.

Pour réaliser ce codage, la conversion d'une variable (ou entrée) suit une procédure assez simple. Chaque variable  $I$ , représentant une caractéristique pour le SNN de la première expérience, est bornée par un intervalle de  $[I_{max} \dots I_{min}]$ . Ainsi, un ensemble de  $m$  neurones récepteurs gaussiens seront utilisés pour représenter cette variable. Chaque neurone récepteur  $i$ , avec  $i=1..m$ , est centré sur une gaussienne à centre  $C_i$  et largeur  $\sigma_i$ . Ces paramètres sont déterminés par les équations suivantes:

$$C_i = I_{min} + \left(\frac{2i - 3}{2}\right) \left(\frac{I_{max} - I_{min}}{m - 2}\right) \quad (4.7)$$

$$\sigma_i = \frac{1}{\gamma} \left(\frac{I_{max} - I_{min}}{m - 2}\right) \quad (4.8)$$

Où la valeur de  $\gamma$  est généralement initialisée à 1,5.

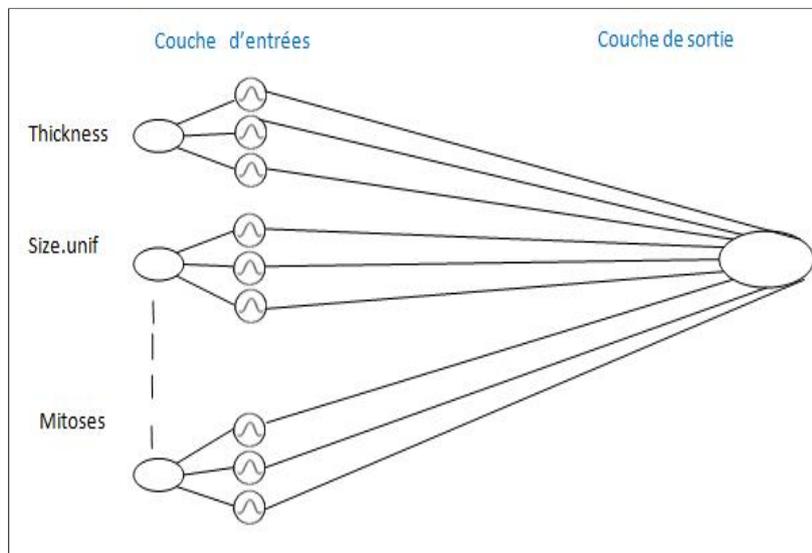


**Figure 4.31** - Exemple de champ réceptive gaussien (CRG): la ligne verticale en rouge représente la valeur à coder et les intersections avec les gaussiennes retourneront les valeurs des délais, si l'intersection se trouve dans la zone « No firing » le délai ne sera pas pris en considération (Dans la figure, l'intervalle supposé être de  $[0, 1]$  dans l'axe des Y a été transformé en délais de  $[0, 10]$  ms pour une meilleure représentation [Stromatias, 2011]).

Pour chaque échantillon, d'apprentissage ou de test, de dimension  $n$ , le résultat du codage est une matrice  $n \times m$  de valeurs comprises entre 0 et 1. Ces valeurs sont ensuite converties en délais temporels pour ensuite créer les trains d'impulsions à introduire à l'entrée du SNN (voir Figure 4.31).

Pour notre cas, les entrées sont représentées par 9 caractéristiques, chaque caractéristique est définie par une population de neurones générés par des champs réceptifs gaussiens (CRG). Pour le neurone en sortie, on a utilisé le codage par phase déterminée par le temps de la première impulsion pour calculer la réponse du système.

La Figure 4.32 montre que chaque caractéristique en entrée est définie par champs réceptifs gaussiens de trois neurones.



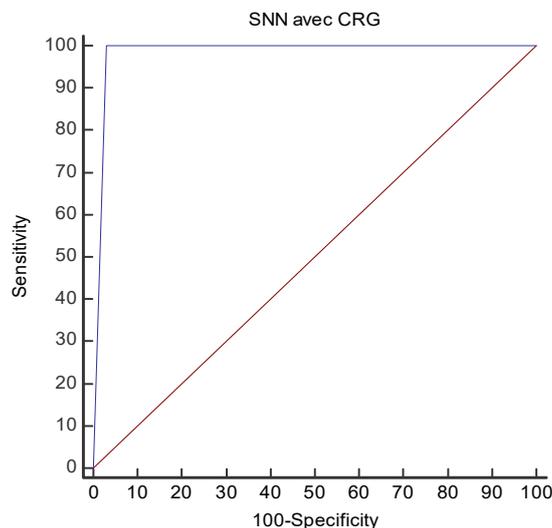
**Figure 4.32** - Architecture du SNN à population de neurones (CRG) pour le diagnostic médical sur WBCD.

La table 4.16 donne la configuration du SNN utilisé à base de population de neurones pour réaliser le diagnostic du cancer du sein sur la base WBCD. Pour la fonction à réponse impulsionnel  $\varepsilon$ , on a remarqué que pour cette expérience varier la valeur des paramètres  $\tau_m$  et  $\tau_s$  n'apporté pas grand chose à la performance du SNN. On a initialisé leurs valeurs, respectivement, à 4 et 3. Pour la fonction de réfractaire  $\eta$ , le paramètre  $\tau_r$  est de 10. Chaque neurone SRM a un seuil de 0,2 avec un nombre de sous-synapses pour chaque connexion égale à 3. Pour le pas d'apprentissage, après plusieurs lancements, on l'a fixé à  $10^{-3}$ .

Neurone du SNN	Nombre de neurones dans la couche d'entrées	Nombre de champs réceptifs gaussiens (CRG)	Nombre de neurones dans la couche de sortie	Délais (ms)
SRM	27	3	1	20

**Table 4.16** - Paramètres du SNN à population de neurones (CRG) pour le diagnostic médicale sur WBCD.

Pour l'évaluation des la phase de généralisation, on a remarqué que le réseau atteint assez facilement les mêmes performances de la première expérience. Le Figure 4.33 montre le graphe AUC. Sachant que le SNN a donné un taux de reconnaissance de 99,26%.



**Figure 3.33** - Graphes AUC du ROC du SNN à population de neurones (CRG) ayant donnés un taux de reconnaissance de 99,26% et un AUC de 0,989.

La spécificité enregistrée est de 100% et la sensibilité est égale à 97,89%. Cette deuxième expérience offre un résultat optimal comme dans la précédente mais moins de diversité dans les résultats.

## 5.5 COMPARAISON AVEC D'AUTRES APPROCHES

La Table 4.17 illustre les résultats obtenus par quelques travaux dans la littérature. Principalement, la majorité des applications utilisent les approches connexionnistes hybridées avec d'autres méthodes : Réseaux Bayésiens, colonies de fourmis, machine à vecteurs de support (SVM), etc. Les machines à vecteurs de support (SVM) et les réseaux de neurones à métaplasticité ont montré les meilleures performances avec un taux de reconnaissance de 99.54% et 99.26%. Par conséquent, avec les résultats obtenus

dans les deux expériences, on a démontré la performance du SNN dans le diagnostic médicale sur la base WBCD.

Approche		Auteurs & années [ref]	Modèle	Taux de test (%)
Neuronale	2 <sup>ème</sup> Génération	Setiono,2000	NeuroRule: Rule 2a, Rule 2b	98.10
			NeuroRule: Rule 3	98.24
		kiyan & al,2004	RBF	96.18
			PNN	97.0
			GRNN	98.8
			MLP	95.74
		Karabatak & al , 2009	Neural network : MLP(9,11,1)	95.2
			Neural network & association rules: MLP (8,11,1)	97.4
			Neural network & association rules: MLP (4,11,1)	95.6
		Mohamed & al, 2011	Fuzzy ARTMAP with genetic algorithms	97.2
	3 <sup>ème</sup> Génération	Bohte & all, 2002	Multi-layered SNN (64, 15, 2) with SpikeProg rule	97.0
		McKenoach & al,2006	Theta neuron (9, 8, 1) BP rule	99
		Bakó,2010	Implemented hardware SNN	89.5
		Marcano & al,2011	AMMLP (9, 8,1)	99.26
			BPNN (9, 8, 1)	94.51
Notre approche , 2012	One-layered SNN (9,1) based on SRM	99.26		
Autres	Parpinelli & al ,2002	Ant-Miner	96.04	
	Übeyli, 2007	SVM	99.54	
	Akay, 2009	SVM-CFS	99.51	
	Peng & al, 2009	CFW	99.50	
	Fallahi & al, 2011	Bayesian Net	97.42	
		Bayesian Net: data balancing and remove missing data	97.83	
		Bayesian Net : ReliefF,remove missing data, data balancing and sorting data upon feature F	98.15	

**Table 4.17** - Différents taux de reconnaissances obtenus dans la base WBCD

## 6 CONCLUSION

Finalement, dans le cas des réseaux de neurones impulsionnels, les contraintes d'initialisation de la topologie du réseau, les problèmes des minimums locaux et du choix des caractéristiques en entrées ne sont pas l'unique barrière qui entravent la performance de ce modèle connexionniste. Eventuellement, on peut remarquer l'importance du codage des données. Clairement, on peut avoir la meilleure topologie avec un algorithme d'apprentissage approprié mais le codage restera toujours cruciale au bon rendement du SNN.

Le problème qu'il n'y a aucun test de performance pour juger qu'un codage est bon ou non. Même si des fois il semble être le plus logique, seule la phase de généralisation déterminera l'efficacité de son usage. Ceci rajoute encore au réseau de neurones un nouveau problème d'initialisation. Cela dit, on ne peut ignorer leurs performances et leurs robustesses une fois utilisés avec le bon codage.

# CHAPITRE 5

## INTEGRATION DU SNN DANS LA DETECTION DE L'ACTIVITE DE LA VOIX (VAD)

### 1 INTRODUCTION

Pendant ces dernières années, il y a eu beaucoup de progrès dans le domaine de la reconnaissance de la parole, spécialement, dans l'amélioration des performances ces systèmes dans des conditions d'acquisition contraignantes [Luu, 2010]. Par exemple, moins de 1% de taux d'erreur a été atteint dans la reconnaissance des mots isolés du locuteur avec une base de données de plus de 20000 mots [Das et al., 1993]. Cependant, ces systèmes opèrent mieux sur des milieux calmes. Leurs performances se dégradent rapidement lorsque l'environnement devient bruyé. Dans le domaine des télécoms, la majorité des systèmes de reconnaissances opérationnels peuvent montrer de faible performance dans des environnements bruyés à partir d'un SNR à 30% [Moreno et al., 1994].

Dans ce chapitre, on a présenté une synthèse du principe et des différentes techniques et paramètres utilisés dans la littérature dans la détection des l'activité de la voix. Ensuite, on a exposé notre approche en décrivant les différents paramètres du SNN proposé pour réaliser le VAD. Aussi, on expose les méthodes d'évaluations retenues de la VAD. Finalement, on illustre les résultats du VAD proposé par différents graphes et mesures pour estimer au mieux la performance et la robustesse du SNN dans le VAD.

### 2 DETECTION DE L'ACTIVITE DE LA VOIX

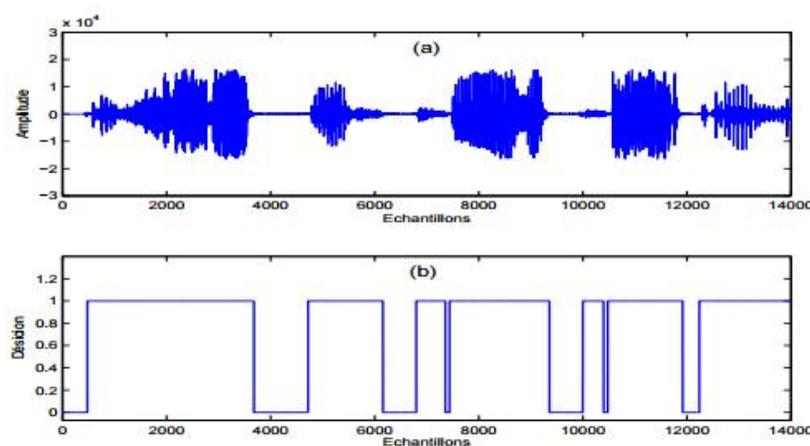
La détection d'activité vocale (VAD: Voice Activity Detection) est une phase très critique dans de nombreuses applications vocales et audio, notamment dans le codage de la parole, la reconnaissance vocale, le rehaussement du signal de parole et l'indexation automatique du signal audio. Ainsi, le VAD est une phase préliminaire essentielle dans de nombreux systèmes de traitement de la parole, y compris la reconnaissance vocale robuste [Karray et al., 2003; Ramirez et al., 2003 ; Mistra, 2012]

et les systèmes de compression pour la transmission à faible bande passante [ITU-T, 1996 ; ETSI, 1999].

Principalement, un algorithme de détection d'activité vocale se doit d'être capable de discriminer entre les régions où la parole est absente dans le signal vocal à analyser. Le bruit intense et non-stationnaire pose de sérieux problèmes aux systèmes VAD. Naturellement, la recherche ces dernières années a mis l'accent sur le développement de systèmes robustes [Ramirez et al., 2007]. Bien que le problème du VAD ait été étudié depuis plusieurs décennies, une solution optimale ne s'est pas encore imposée.

## 1.1 PRINCIPE DE LA VAD

Le VAD effectue principalement une discrimination entre deux régions dans le signal de parole: région active (région qui contient de la voix) et région inactive (région sans voix). Par convention, la sortie d'un VAD est « un » si on a activité de voix, ou « zéro » si on n'a pas d'activité de voix. La Figure 5.1 illustre un résultat de VAD pour un extrait de parole.

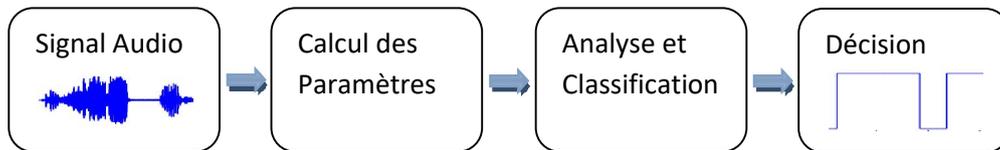


**Figure 5.1** - Exemple illustrant le principe de la détection d'activité vocale [Hamzé, 2007].

Dans la littérature, la grande majorité des VAD découpent le signal audio d'entrée en trames. Ces trames peuvent être découpées avec chevauchement (overlapping) ou sans chevauchement. Chaque trame se définit comme une portion de signal de très courte durée et de longueur fixe (habituellement de l'ordre de 10 à 50 ms, idéalement 10 ms).

À partir de cette portion de signal audio ou « trame », on extrait un ensemble de paramètres. Ces paramètres peuvent être calculés dans le domaine temporel (nombre de passages par zéro, niveau d'énergie, coefficients d'autocorrélation, etc.) ou à partir

du domaine spectral (DFT: Discrete Fourier Transform, analyse de forme spectrale, DOA: Direction Of Arrivals, ondelettes, etc.) [Chiodi, 2010]. À partir de ces paramètres, le VAD retourne soit une sortie de « 0 » ou « 1 » pour marquer, respectivement, inactivité ou activité vocale.



**Figure 5.2** - Processus général d'un algorithme de détection d'activité vocale.

La Figure 5.2 illustre le processus générale qu'un algorithme de VAD est sensé reproduire pour fournir une décision à partir d'un signal audio.

## 1.2 APPROCHES ET PARAMETRES DU VAD

De nombreux algorithmes qui utilisent une large gamme de paramètres ont été proposés pour répondre aux problèmes pratiques rencontrés [Graf et al., 2015]. Au départ, dans les premières systèmes de reconnaissances des mots, on a utilisé des caractéristiques simples, telles que l'énergie ou le taux de passage par zéro ou « *zero-crossing rate* » [Rabiner et al., 1975]. Cependant, en raison de la simplicité des caractéristiques utilisées, la performance de ces systèmes ne devenait plus crédible à partir de taux de SNR élevé. Au cours des décennies suivantes, la complexité des caractéristiques a été accrue pour atteindre des résultats de détections raisonnables dans des conditions de bruit plus difficiles. En particulier, les caractéristiques de la parole, telle que l'analyse spectrale [Rabiner et al., 1977] et la structure harmonique de la parole [Tucker, 1992], ont été soigneusement examinées.

Sohn et al. [Sohn et al., 1999] ont introduit des approches basées sur des modèles statistiques pour le VAD. Ils ont modélisé les distributions de la parole et du bruit par différentes fonctions de densité de probabilité et ont utilisé le rapport de vraisemblance « *likelihood ratio* » entre les deux modèles pour la décision. Plus tard, ce concept a été affiné dans plusieurs algorithmes avancés [Cho et al., 2001], [Chang et al., 2006], [Chang et al., 2004], [Chang et al., 2006], [Ramirez et al., 2007] et [Gòrriz et al., 2010].

Une des limitations de la plupart des premiers algorithmes de VAD étaient qu'ils ne tenaient compte que des données de la trame courante. Ramirez et al. [Ramirez et al., 2004] ont montré que la détection bénéficie d'informations à long terme sur le signal de

parole. En étendant la plage temporelle des données employées dans la décision, on peut également saisir les caractéristiques à long terme de la parole, comme le degré de stationnarité.

La tendance à considérer des informations plus contextuelles se poursuit, motivée par l'augmentation de la robustesse de la détection même dans des environnements défavorables. Les propriétés de modulation ont été identifiées comme des aspects importants de la perception humaine de la parole. Différentes caractéristiques, telle que la modulation spectro-temporelle (STM : Spectro-Temporal Modulation) [Mesgarani et al., 2006] ou le spectrogramme de modulation d'amplitude (AMS: Amplitude Modulation Spectrogram) [Anemüller et al., 2008], héritent de ce fait et reflètent la présence de la parole d'une manière similaire à la perception humaine. En raison de la diversité des caractéristiques du signal de la parole, une combinaison de caractéristiques complémentaires est souhaitable dans la pratique [Van Segbroeck et al., 2013].

Plusieurs méthodes ont été développées pour essayer d'améliorer la qualité du VAD dans des environnements bruyants. Parmi ces méthodes traditionnelles ou modernes, on cite l'algorithme de compression audio G729 [ITU-T, 1996] qui est un algorithme standardisé par l'Union Internationale des Télécommunications avec un annexe B qui comprend des algorithmes de VAD, L'ETSI (European Telecommunications Standards Institute) qui a fourni un VAD conçu pour la téléphonie mobile GSM pour les encodeurs de type AMR (Adaptive Multi-Rate) [ETSI, 1999], les réseaux de neurones RBF (Radial Basis Function) [Kim et al., 05], les méthodes basées sur les ondelettes [Chen et al., 2002 ; Chen et al., 2005 ; Wu et al., 2006], les outils statistiques [Soh et al., 1999], ou encore des méthodes basées sur l'utilisation de plusieurs microphones [Rubio et al., 2007 ; GSC, 2001], très efficaces pour cibler la source de la voix et éliminer le bruit. Il existe aussi des méthodes combinant plusieurs techniques déjà existantes mais inefficaces dans des bruits non-stationnaires [Tanyer et al., 2000].

### **3 CORPUS UTILISE**

Dans cette partie expérimentale, on utilise la base de données NOIZEUS (disponible : <http://ecs.utdallas.edu/loizou/speech/noizeus/>). Ce corpus a été conçu pour faciliter la comparaison des algorithmes de rehaussement de la parole entre les groupes de recherche. La base contient trente phrases de la base de données IEEE (Institute of Electrical and Electronics Engineers) enregistrées dans une cabine insonorisée avec le matériel d'enregistrement de Tucker Davis Technologies (TDT) réalisées par trois hommes et trois femmes. Les enregistrements ont été originalement pris avec une fréquence d'échantillonnage de 25 kHz et ont été ultérieurement échantillonnés à 8

kHz. La longueur totale de tous les enregistrements était de 80,04 s, ce qui offrait, avec un chevauchement et une longueur de trame de  $L = 256$ , au total 5000 trames pour la détection de l'activité de la voix. Le pourcentage des segments de parole est de 61,28%, soit deux fois plus élevé que celui de [Sohn et al., 1999 ; Chang et al., 2003], mais moins important de 5% que celui de [Chang et al., 2004 ; Gorriz et al., 2010] (voir Table 5.1).

Nom du Fichier	Locuteur	Genre	Phrase prononcée
sp01.wav	CH	M	<i>The birch canoe slid on the smooth planks.</i>
sp02.wav	CH	M	<i>He knew the skill of the great young actress.</i>
sp03.wav	CH	M	<i>Her purse was full of useless trash.</i>
sp04.wav	CH	M	<i>Read verse out loud for pleasure.</i>
sp05.wav	CH	M	<i>Wipe the grease off his dirty face.</i>
sp06.wav	DE	M	<i>Men strive but seldom get rich.</i>
sp07.wav	DE	M	<i>We find joy in the simplest things.</i>
sp08.wav	DE	M	<i>Hedge apples may stain your hands green.</i>
sp09.wav	DE	M	<i>Hurdle the pit with the aid of a long pole.</i>
sp10.wav	DE	M	<i>The sky that morning was clear and bright blue.</i>
sp11.wav	JE	F	<i>He wrote down a long list of items.</i>
sp12.wav	JE	F	<i>The drip of the rain made a pleasant sound.</i>
sp13.wav	JE	F	<i>Smoke poured out of every crack.</i>
sp14.wav	JE	F	<i>Hats are worn to tea and not to dinner.</i>
sp15.wav	JE	F	<i>The clothes dried on a thin wooden rack.</i>
p16.wav	KI	F	<i>The stray cat gave birth to kittens.</i>
sp17.wav	KI	F	<i>The lazy cow lay in the cool grass.</i>
sp18.wav	KI	F	<i>The friendly gang left the drug store.</i>
sp19.wav	KI	F	<i>We talked of the sideshow in the circus.</i>
sp20.wav	KI	F	<i>The set of china hit the floor with a crash.</i>
sp21.wav	SI	M	<i>Clams are small, round, soft and tasty.</i>
sp22.wav	SI	M	<i>The line where the edges join was clean.</i>
sp23.wav	SI	M	<i>Stop whistling and watch the boys march.</i>
sp24.wav	SI	M	<i>A cruise in warm waters in a sleek yacht is fun.</i>
sp25.wav	SI	M	<i>A good book informs of what we ought to know.</i>
sp26.wav	TI	F	<i>She has a smart way of wearing clothes.</i>
sp27.wav	TI	F	<i>Bring your best compass to the third class.</i>
sp28.wav	TI	F	<i>The club rented the rink for the fifth night.</i>
sp29.wav	TI	F	<i>The flint sputtered and lit a pine torch.</i>
sp30.wav	TI	F	<i>Let's all join as we sing the last chorus.</i>

**Table 5.1** – Représentation de la base de données NOIZEUS.

Ensuite, ces phrases ont été corrompus par huit différents bruits du monde réel à différents taux d'SNR (en tout 720 phrases, soit 30 phrases pour chaque environnement bruité). Le bruit a été pris à partir de la base de données AURORA incluant le bruit des environnements suivants : Foule de personnes, voiture, salle d'exposition, restaurant, rue, aéroport, train et gare de trains [Hu et al., 2007].

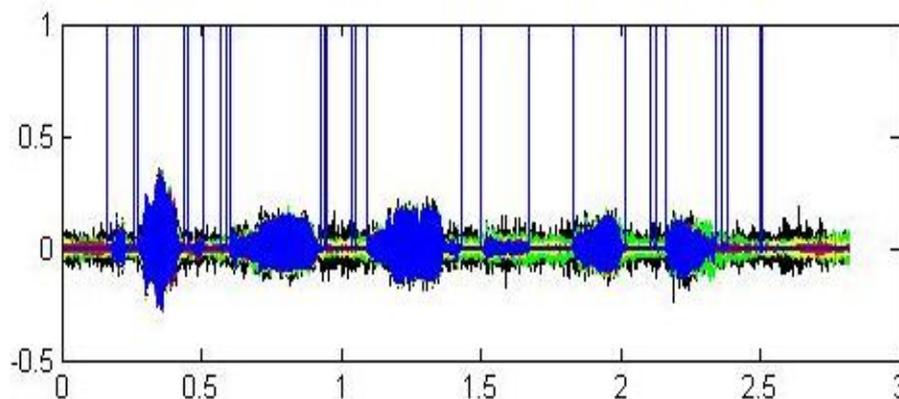
Pour notre approche, on a choisis de prendre quatre environnements relatifs à la base de données : Aéroport, rue, voiture et restaurant ; à des taux d'SNR de 5dB, 10dB et 15dB.

## 4 PRETRAITEMENTS ET PARAMETRES

Pour réaliser notre expérimentation, il est tout à fait naturel de passer par une phase de prétraitement afin d'extraire les paramètres à présenter au SNN pour réaliser correctement la détection de l'activité vocale. Ainsi, cette phase se chargera de préparer les 30 signaux nets et sous milieux bruités sous un format de caractéristiques beaucoup plus simple à intégrer à l'apprentissage et à la généralisation du SNN.

La sortie d'un VAD est de nature binaire. Ainsi, on a procédé un VAD à étiquetage manuel de chaque signal sous sa forme « *clean* » pour ensuite attribuer à chaque trame du signal une assignation théorique concernant la présence ou l'absence d'une activité vocale dans cette trame.

Formellement, le procédé consiste à diviser chaque signal en trame de 10ms ce qui est équivalent à  $L=80$  échantillons, pour chaque trame on a procédées a un codage LPC avec un ordre de  $p=10$ .



*Figure 5.3* - Représentation du découpage manuel du signal « *sp01.wav* » sous l'environnement « Aéroport »: Clean (bleu), 15 dB (rouge), 10 dB (jaune), 5 dB (vert) et 0 dB (noir).

La *Figure 5.3* résume le procédé de l'étiquetage d'un signal en entrée dans un environnement « *clean* ». Ensuite, il suffit de superposer le résultat de la VAD manuel

du signal sur celui-ci mais sous les milieux bruités sélectionnés à différents SNR pour obtenir l'étiquetage des signaux restants.

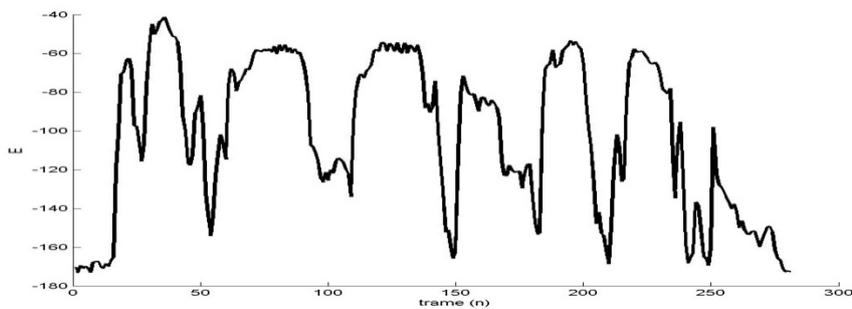
Ensuite, on a calculé trois caractéristiques pour chaque trame du signal en entrée [Kim et al., 2005]. Ces paramètres sont les suivants:

### 1) Short-time average power parameter (E)

Le Short-time average power parameter est défini comme la valeur logarithmique de la fonction d'autocorrélation d'ordre zéro. Il est donné par l'équation (5.1) où le paramètre  $N$  est la longueur de la trame en termes de nombre d'échantillons,  $m$  la position de la trame avec  $m = 1..N$  et  $s(n)$  le signal en entrée du VAD.

$$E = 10 \log \left[ \frac{1}{N} \sum_{n=m-N+1}^m s(n)^2 \right] \quad (5.1)$$

La valeur du paramètre  $E$  est généralement faible pour les trames inactives et élevé durant une période active.

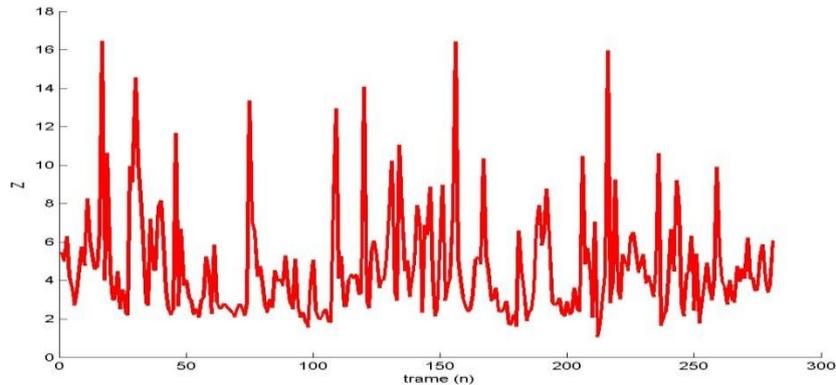


**Figure 5.4** - Représentation du paramètre short-time average power parameter ( $E$ ) à partir du signal clean « sp01.wav ».

### 2) Zero-order most likelihood parameter (Z)

Le zero-order most likelihood parameter ( $Z$ ) est connu pour être une distance de l'enveloppe spectrale plate entre les zones actives et inactives de la voix. Le paramètre  $Z$  est généralement élevé dans les zones actives de la voix. Ce paramètre est généré par le logarithme de la somme des carrés des coefficients LPC (linear prediction coefficients) comme présenté dans l'équation ci-dessous.

$$Z = \log \sum_{i=0}^p a_i^2 \quad (5.2)$$



**Figure 5.5-** Représentation du paramètre zero-order most likelihood parameter ( $Z$ ) à partir du signal clean « sp01.wav ».

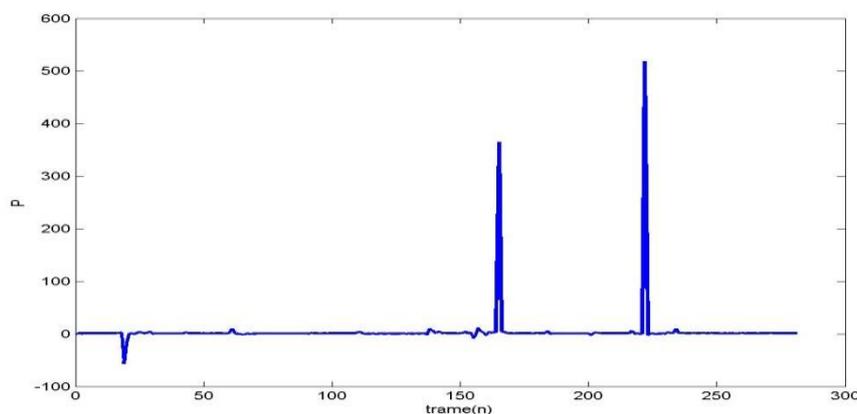
Où  $a_i$  sont les coefficients LPC et  $p$  l'ordre de prédiction.

### 3) Pitch period difference parameter (P)

Le Pitch period difference parameter est généré comme une autocorrélation maximale normalisée. Ceci est lié avec la stabilité du Pitch dans un signal. Le paramètre  $P$  est donné par l'équation (5.3).

$$P = \max_{\tau} \left[ \frac{\log \sum_{n=0}^{N-1} e(n)e(n-\tau)}{\log \sum_{n=0}^{N-1} e(n)e(n)} \right] \quad (5.3)$$

Où  $20 \leq \tau \leq 160$  et  $e(n)$  est le signal d'erreur prédit linéairement (linear prediction error signal) au temps  $n$ .



**Figure 5.6 -** Représentation du paramètre pitchperiod difference parameter ( $P$ ) à partir du signal clean « sp01.wav ».

A travers la Figure 5.4, Figure 5.5 et Figure 5.6, on remarque, en comparant avec Figure 5.3 (Le graphe du signal « Clean » en bleu), qu'effectivement lors des périodes actives

les paramètres  $E$  et le  $Z$  sont élevés et le paramètre  $P$  est plutôt stable. Dans les cas d'inactivité de voix, ces caractéristiques sont complètement opposées.

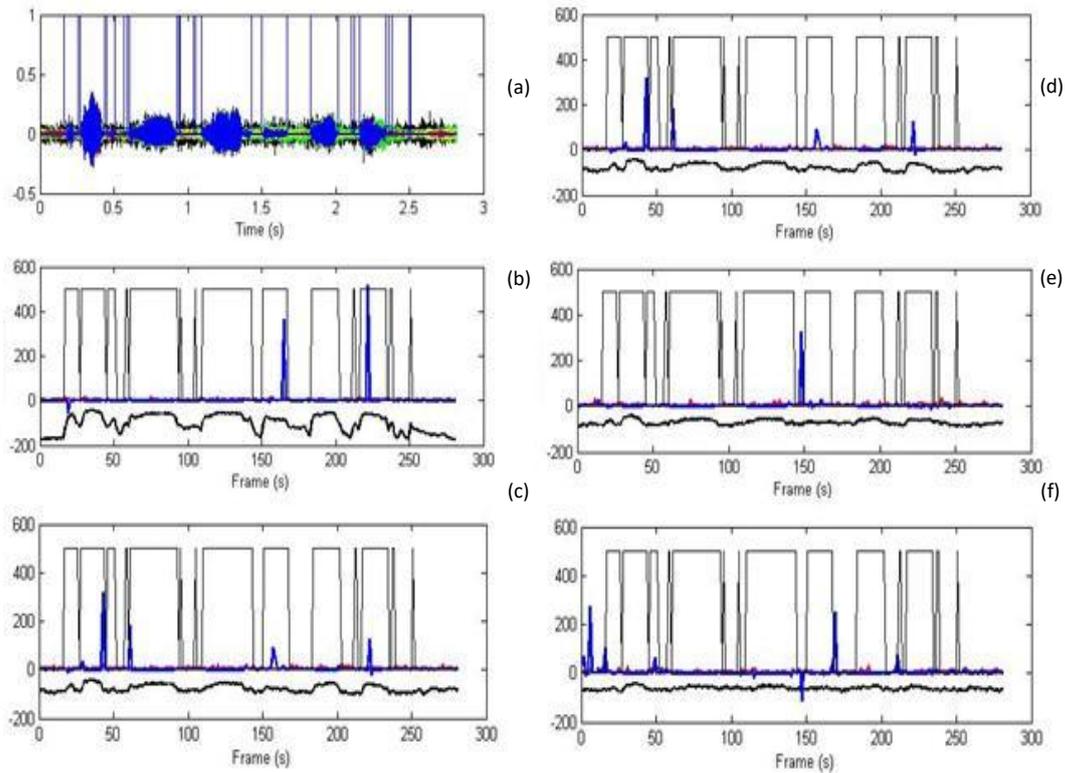


Figure 5.7 - Représentation des variations des caractéristiques calculées du signal (a) sous l'environnement Aéroport: Clean (b), 15 dB (c), 10 dB (d), 5 dB (e) et 0 dB (f).

Ces caractéristiques fonctionnent généralement bien sous des bruits de fond élevés. La Figure 5.7 montre les variations de ces paramètres à travers un SNR de 0, 5, 10 et 15dB.

## 5 STRUCTURE ET PARAMETRES DU SNN PROPOSE POUR LE VAD

Lors de l'expérimentation du SNN dans divers cas d'école dans le chapitre précédent, on a recensé certains éléments sur le choix du codage et de la structure qui pouvait avoir un impact direct sur les performances du SNN à réaliser un problème classique de classification. Par conséquent, le choix de la structure et des paramètres de notre approche SNN pour le VAD a été beaucoup influencé par ces conclusions.

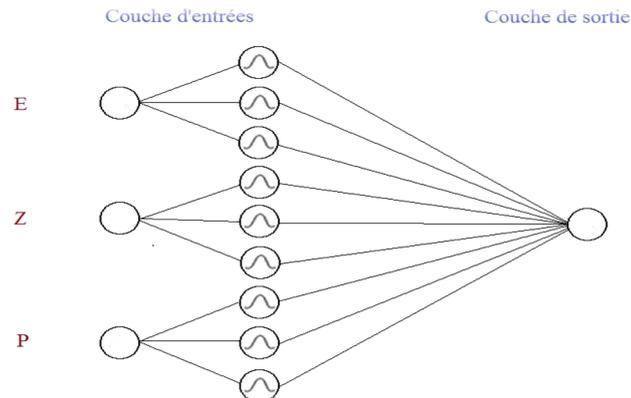


Figure 5.8 – Structure du SNN monocouche proposé pour réaliser le VAD.

Le SNN proposé pour réaliser le VAD est illustré dans la Figure 5.8. On a utilisé une structure monocouche avec des unités SRMs déjà décrit dans la Section 3.4.1. Pour la couche d'entrées, on a repris le codage à base de population de neurones qui consiste à modéliser chaque neurone d'entrée par un champ récepteur gaussien (CRG) (voir section 4.6.3). Ainsi, les caractéristiques en entrées sont définies par des champs réceptifs gaussiens de six neurones étendus sur un intervalle de délais de 25 ms. La couche de sortie est représentée par un seul neurone SRM chargé de retourner la décision du VAD. La réponse du système est déterminée par un codage par phase. Par conséquent, une trame est jugée active si le neurone de sortie tire durant la phase [0,15] ms sinon la trame sera classée inactive.

Pour les paramètres du SRM, les constantes  $\tau_m$ ,  $\tau_s$  et  $\tau_r$  ont été initialisées, respectivement, à 8, 4 et 8. Le seuil de chaque neurone est initialisé à 0,1. Le nombre de sous-synapses pour chaque connexion varié entre 6 et 10. Pour le pas d'apprentissage, après plusieurs lancements, on l'a fixé à  $10^{-3}$ .

## 6 METHODES D'EVALUATIONS DU SNN PROPOSE POUR LE VAD

Pour l'évaluation des résultats du VAD, en plus du taux de reconnaissance (TR), on a utilisé le Receiver Operating Characteristics (ROC). Dans la section 4.6.1, on a déjà décrit les paramètres AUC, Sensitivité et Spécificité (voir les équations (4.5) et (4.6)) qui servent à évaluer et à décrire la performance d'un système d'aide à la décision. Dans notre expérimentation, les courbes AUC (Area Under ROC Curve) se chargeront à mesurer et à comparer la précision des différents tests du SNN sur divers environnements en variant les taux du SNR de 5 à 15 dB. Techniquement, le AUC est une mesure située entre 0 et 1 et indique la fiabilité de la décision dans la phase de généralisation du VAD: Plus l'AUC est proche du 1 mieux c'est.

Dans le même contexte, on a aussi utilisé deux nouveaux paramètres d'évaluation : La probabilité de bonne décision dans les zones actives ( $P_d$ ) et probabilité de bonne décision dans les zones inactives ( $N_d$ ).

$$P_d = \frac{\text{Nombre de bonne décision des zones actives (ZA)}}{\text{Nombre total de zone actives}} \times 100 \quad (5.4)$$

$$N_d = \frac{\text{Nombre de bonne décision des zones inactives (ZNA)}}{\text{Nombre total de zone inactives}} \times 100 \quad (5.5)$$

Clairement, Plus la mesure du  $P_d$  est grande et plus le VAD va laisser passer de la voix. Parallèlement, Plus la mesure du  $N_d$  est grande et moins le VAD va laisser passer le bruit et autres signaux non vocaux. Donc plus  $P_d$  et  $N_d$  s'approchent de 100%, meilleure est la performance du VAD [Chiodi, 2010].

Cependant, la mesure du  $P_d$  détermine plus la qualité du VAD car plus le  $P_d$  est élevé plus la perception auditive du signal sera meilleure car si le  $P_d$  est supérieur à 90% il y aura une excellente qualité du message vocale et il sera très compréhensible. A partir d'un  $P_d$  entre 90% et 80%, le message sera toujours compréhensible excepté quelques segments aux débuts ou fins de mots. Un  $P_d$  entre 80% et 65% nuira forcément à la qualité de la VAD et rendra le message difficile à saisir mais c'est à partir d'un  $P_d$  inférieur à 65% que le signal deviendra complètement incompréhensible.

## 7 TESTS ET EVALUATIONS DU SNN PROPOSE POUR LE VAD

Après avoir réalisé la phase de prétraitement, on a divisé la base de connaissance en deux parties égales, c'est-à-dire 50% pour l'apprentissage et 50% pour le test, pour préparer le SNN à réaliser le VAD. Ainsi, La phase de généralisation se base sur 15 signaux sur quatre environnements (Aéroport, rue, voiture et restaurant) à des taux de SNR de 5 à 15dB.

Les résultats obtenus des taux de reconnaissances du SNN varient entre 89% et 97,63%. La Figure 5.9, Figure 5.10, Figure 5.11 et Figure 5.12 illustre le graphe AUC des quatre environnements : Aéroport, rue, voiture et restaurant.

Intégration du SNN dans la détection de l'activité de la voix (VAD)

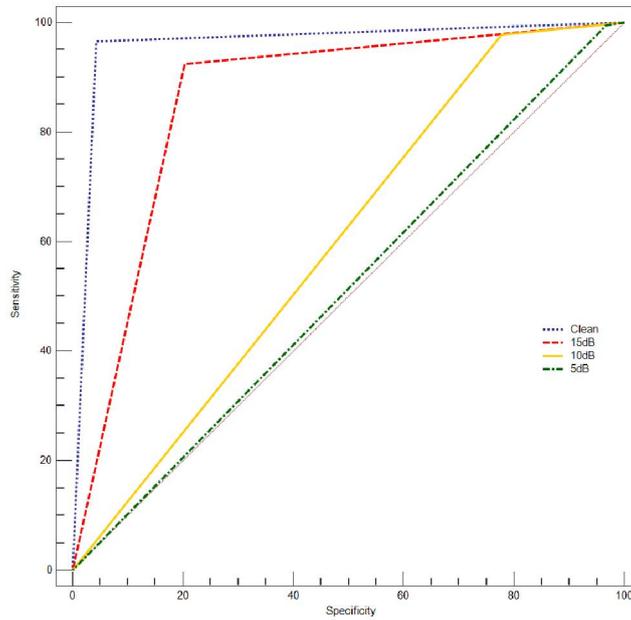


Figure 5.9 – Graphes AUC du VAD proposé à base de SNN du signale Clean et en variant le SNR dans l'environnement « Aéroport ».

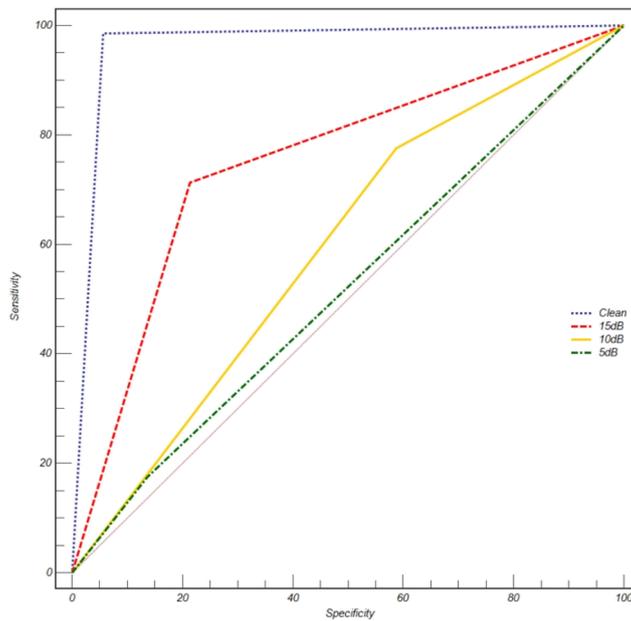
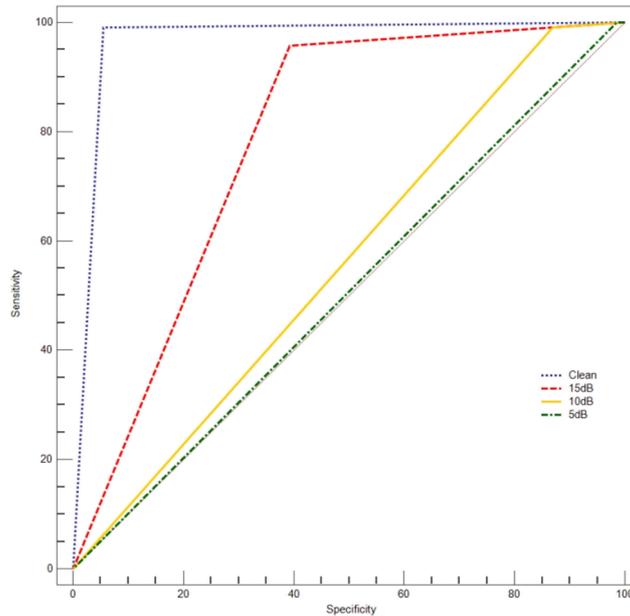
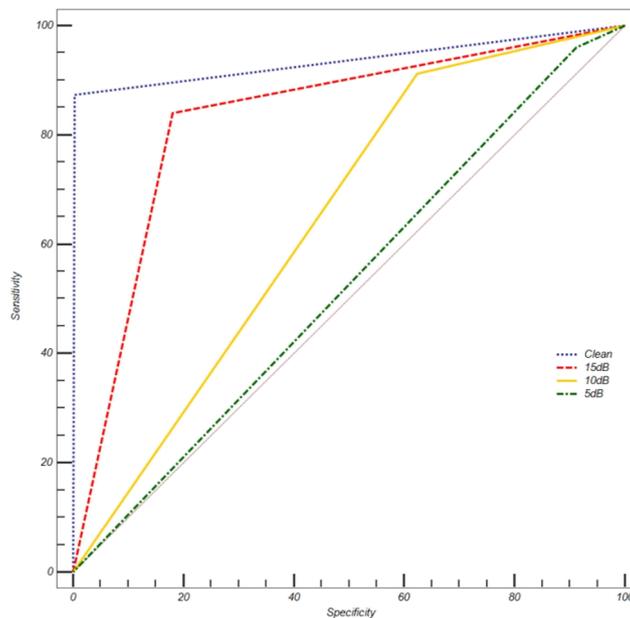


Figure 5.10 – Graphes AUC du VAD proposé à base de SNN du signale Clean et en variant le SNR dans l'environnement « Rue ».



**Figure 5.11** – Graphes AUC du VAD proposé à base de SNN du signale Clean et en variant le SNR dans l'environnement « Voiture ».



**Figure 5.12** – Graphes AUC du VAD proposé à base de SNN du signale Clean et en variant le SNR dans l'environnement « Restaurant ».

La Table 5.2 illustre tous les résultats obtenus du VAD proposé à base de SNN. On a remarqué à travers plusieurs expériences que l'optimisation de la performance du VAD proposé dans un signal clean a un impacte direct sur la robustesse de celui-ci.

Formellement, plus le SNN atteint des performances optimales dans des signaux cleans et plus sa robustesse devient médiocre surtout face à un taux de SNR inférieur ou égale à 10 dB. On remarque surtout ça dans le taux du paramètre Nd qui reflète l'incapacité du SNN à classer correctement les zones inactives à cause de l'importance du bruit qui compromet leurs statuts et forces le VAD à les considérer en zones actives. Naturellement, ceci n'affecte pas la compréhension auditive du signal après lui avoir appliqué le VAD. Néanmoins, l'objectif du VAD ne sera pas complètement atteint.

		TR (%)	AUC (%)	Pd (%)	Nd (%)
Aéroport	Clean	96,27	0,961	96,59	95,66
	15dB	88,10	0,861	92,50	79,64
	10dB	71,95	0,600	97,82	22,26
	5dB	66,12	0,514	99,00	3,23
Rue	Clean	97,15	0,965	98,62	94,34
	15dB	73,81	0,750	71,31	78,62
	10dB	65,16	0,594	77,61	41,21
	5dB	58,92	0,519	82,48	13,66
Voiture	Clean	97,55	0,968	99,19	94,48
	15dB	83,75	0,782	95,75	60,69
	10dB	69,64	0,561	99,12	13,00
	5dB	66,24	0,507	100	1,39
Restaurant	Clean	91,62	0,936	87,41	99,70
	15dB	83,35	0,830	84,09	81,92
	10dB	72,83	0,644	91,20	37,54
	5dB	66,17	0,524	96,06	8,74

**Table 5.2** - Résultats du TR, AUC, Pd et Nd du VAD proposé à base de SNN sur les environnements aéroport, rue, voiture et restaurant avec un signal clean et un SNR variant de 5 à 15 dB.

## 8 CONCLUSION

Dans ce chapitre, nous avons présenté une application de détection d'activité de la voix avec un réseau de neurones impulsionnels monocouche à apprentissage supervisé. En partant d'un SNN à structure minimale, le gain en temps d'exécution au niveau de l'apprentissage et du test se ressent rapidement malgré le nombre important des paramètres à initialiser.

Néanmoins, nous jugeons que les résultats obtenus par le VAD proposé sont plutôt modeste à voir moyen par rapport aux méthodes actuellement proposés dans la littérature. Le SNN proposé a effectivement montré une très bonne performance à réaliser le VAD dans notre expérimentation. Cependant, on ne peut affirmer sa robustesse à partir des caractéristiques sélectionnées pour réaliser cette application. Les paramètres d'évaluation le montre clairement

dans les différents niveaux de SNR testés et sous tous les milieux bruités choisis. Ainsi, cette application laisse porte ouverte à d'autres perspectives afin d'optimiser le rendement du SNN dans le VAD.

# CONCLUSION GENERALE

Avec l'évolution des recherches en neurobiologie, il devient de plus en plus aisé de simuler, avec plus ou moins de réalisme, l'évolution de l'activité dans des réseaux de neurones artificiels de plus en plus grands et de plus en plus complexes. Bien que beaucoup d'architectures connexionnistes classiques ont déjà montré leurs performances, leurs puissances de calcul et leurs capacités de généralisation dans un nombre importants de travaux de recherche portant sur la reconnaissance des formes et l'intelligence artificielle ceci n'a pas ralenti l'intérêt de reformuler le neurone artificiel en une version beaucoup plus adéquate au modèle biologique. L'objectif principale de la neuro-computation et des chercheurs est de modéliser un neurone artificiel sur un fondement asynchrone.

La formalisation du modèle impulsionnel « *spiking neuron* » est à l'image même de cette expansion de connaissance dans l'axe neuro-computationnel avec l'apparition des modèles biophysiques et des modèles computationnels. Ce modèle reconnu comme le plus récent des neurones artificiels, fait la troisième et la toute dernière de ces générations ; même si l'on reconnaît que l'origine de son fondement théorique apparaît dans les travaux très avancés de Hodgkin et Huxley en 1963.

Actuellement, l'utilisation de ce modèle s'impose largement dans l'intelligence artificielle même si il hérite des inconvénients classiques des architectures neuronales traditionnelles. Ceci n'est pas une coïncidence, il faut savoir que les neurones impulsionnels sont capables de reproduire avec exactitude le fonctionnement des neurones classiques. De plus, les unités impulsionnelles offrent la possibilité d'introduire explicitement une dimension temporelle dans des réseaux de neurones grâce à la forme dont l'information est propagée au réseau ; sous forme de train d'impulsion. Par conséquent, ce modèle se voit aussi devant de nouvelles problématiques liées aux codages de l'information qui affecte aussi la construction des règles d'apprentissage surtout si l'on choisit un mode supervisé.

Dans cette thèse, on reprend ces impasses que l'on reconnaît clairement dans les réseaux de neurones impulsionnels (SNN) en expérimentant une architecture monocouche à base de neurone à réponse impulsionnelle (SRM) utilisant un mode d'apprentissage supervisé.

L'approche porte à tester la performance et la robustesse d'un SNN avec une architecture très simple dans divers problèmes de classification mais exclusivement la reconnaissance de la parole ; ceci en utilisant une variante de l'algorithme de la descente du gradient « *SpikProp* ».

L'approche a confirmé certaines difficultés quant à choisir le type de codage où l'on retrouve expérimentation de plusieurs formes de codages tout en dévoilant l'impacte du codage sur la rentabilité des résultats à travers des comparaisons sur différentes applications. Aussi, on a pu affirmer à travers cette approche qu'un SNN ne nécessite pas forcément une architecture complexe et un temps d'apprentissage important pour réaliser une bonne performance comme on a pu le constater à travers les résultats des phases expérimentales des cas d'école traités et du VAD. Cependant, quant il s'agit d'un problème d'aide à la décision, le réseau n'est pas toujours capable de fournir de bon résultats tout en montrant une robustesse. Ceci a été clairement ressenti dans notre SNN proposé par le VAD où performance et robustesse n'allait pas dans le même sens. Mais il est clair que l'approche reste sujette de beaucoup de perspectives et d'optimisations même si elle apporte les résultats du VAD ont été plutôt modestes.

# BIBLIOGRAPHIE

## A

- 
- [Abbott, 1999] Abbott L. F., *Lapicque's introduction of the integrate-and-fire model neuron (1907)*, Brain Research Bulletin, Vol. 50, No. 5-6. pp. 303-304, 1999. doi:10.1016/s0361-9230(99)00161-6
- [Abeles, 1982] Abeles M., *Local cortical circuits: an electrophysiological study*, Springer, Berlin, 1982.
- [Adeline, 2005] Adeline E., *Enhancement of Parallel Thinning Algorithm for Handwritten Characters Using Neural Network*, Master Thesis, Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Teknologi Malaysia, 2005.
- [Alexandre, 2005] Alexandre F., *Les spiking neurons: une leçon de la biologie pour le codage et le traitement des données*, Traitement et Analyse de l'Information: Méthodes et Applications - TAIMA'2005, Hammamet, Tunisie, Faouzi Ghorbel et Stephane Derrode et Jean-Paul Haton et Lamia Ben Youssef Editors, 2005
- [Anemüller et al., 2008] Anemüller J., Schmidt D., Bach J-H., *Detection of speech embedded in real acoustic background based on amplitude modulation spectrogram features*, ISCA, in Proc. of INTERSPEECH., Brisbane, Australia, 2008.
- [Arbib, 1998] Arbib M.A., *The Handbook of Brain Theory and Neural Networks*, MIT Press, Juillet 1998.
- [Azouz et al., 1999] Azouz R., Gray C. M., *Cellular mechanisms contributing to response variability of cortical neurons in vivo*, J. Neurosci., Vol.19, pp.2209-2223, 1999
-

## B

- 
- [Bakó, 2010] Bakó L., *Real-time classification of datasets with hardware embedded neuromorphic neural networks*, Briefings in Bioinformatics. Special Issue: Parallel and Ubiquitous Methods and Tools in Systems Biology, Vol. 11, No.3, pp.348-363, 2010.
- [Belatrache et al.,2003] Belatreche A., Maguire L. P., McGinnity M., Xiang Wu Q., *A method for supervised training of spiking neural networks*, In Proceedings of the IEEE Cybernetics Intelligence - Challenges and Advances (CICA'2003), Reading, UK, pp.39-44, September 17, 2003.
- [Bendaoua et al., 2013] Bendaoua S., Bouchetara ., *La détection de l'activité du signal de la parole par les réseaux de neurones impulsionnels*, Master RFIA, USTO MB, 2013
- [Berrêdo, 2005] Berrêdo R C., *A Review of Spiking Neuron Models and Applications*, Master of Science, Universidade Federal de Minas Gerais, November 2005.
- [Bichler, 2012] Bichler O., *Contribution à la conception d'architecture de calcul auto-adaptative intégrant des nano-composants neuromorphiques et applications potentielles*, thèse de doctorat, Université Paris-Sud, 2012.
- [Bichler et al., 2012] Bichler O., Querlioz D., Thorpe S., Bourgoïn J., Gamrat C., *Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity*, Neural Networks, Vol.32, pp.339-348, 2012. doi:10.1016/j.neunet.2012.02.022
- [Bohte et al., 2000] Bohte S., Kok J. , La Poutre H., *Spike-prop: error-backpropagation in multi-layer networks of spiking neurons*, In Proceedings of the European Symposium on Artificial Neural Networks (ESANN'2000), M. Verleysen, editor, Bruges, Belgium. pp.419-425, April 26-28, 2000.
- [Bohte et al., 2002] Bohte S.M., Kok J.N., La Poutre H., *Error-backpropagation in temporally encoded networks of spiking neurons*, Neurocomputing, Vol. 48, pp.17-37, 2002.
- [Booij, 2004] Booij O., *Temporal pattern classification using spiking neural networks*, Unpublished master's thesis. University of Amsterdam, 2004. Available from <http://www.xs4all.nl/~obooij/study>
- [Booij et al., 2005] Booij O., Tat Nguyen H., *A gradient descent rule for spiking neurons emitting multiple spikes*, Inf. Process. Lett, Vol.95, No.6, pp.552-558, 2005.
-

- 
- [Bower et al., 2003] James M. Bower and David Beeman. *The Book of Genesis – Exploring Realistic Neural Models with the GEneral NEural Simulation System*. Internet Version, 2nd edition, 2003. <http://www.genesis-sim.org/GENESIS>
- [Brette, 2003] Brette R., *Modèles Impulsionnels De Réseaux De Neurones Biologiques*, Thèse dirigée par Yves BURNOD, Ecole Doctorale Cerveau-Cognition-Comportement. Soutenue le 9 décembre 2003 à Paris
- [Brette et al., 2003] Brette, R., Guigon, E., *Reliability of spike timing is a general property of spiking model neurons*, Neural Computation, Vol.15, No.2, pp.279–308, 2003
- [Buhry , 2010] Buhry Laure, *Estimation de paramètres de modèles de neurones biologiques sur une plate-forme de SNN (Spiking Neural Network) implantés "in silico"*, Thèse de doct. Université Bordeaux 1, Bordeaux, 21 septembre 2010
- 

## C

- 
- [Cauwenberghs et al., 2009] Gert Cauwenberghs, Moonjung Kyung and Eric Weiss. *A VLSI Implementation: Izhikevich's Neuron Model*. 2009.
- [Catterall,1988] Catterall W.A., *Structure and function of voltage-sensitive ion channels*. Science, Vol.242, pp.50–61, 1988.
- [Chacron et al., 2003] Chacron M.J., Pakdaman K., Longtin A., *Interspike interval correlations, memory, adaptation, and refractoriness in a leaky integrate-and-fire model with threshold fatigue*, Neural Comp., Vol.15, pp.253–278, 2003.
- [Chang et al., 2003] Chang, J.H., Kim, N.S., *Voice Activity detection Based on Complex Laplacian Model*, Electronics Letters, Vol.39, No.7, pp.632, 2003.
- [Chang et al., 2004] Chang, J.H., Shin, J.W., Kim, N.S., *Voice Activity Detector Employing Generalised Gaussian Distribution*, Electronics Letters, Vol.40, No.24, pp.25–26, 2004.
- [Chang et al., 2006] Chang J-H., Kim N.S., Mitra S.K., *Voice activity detection based on multiple statistical models*, IEEE Trans. Signal Process, Vol.54, No.6, pp.1965–1976, 2006.
- [Chevallier, 2009] Chevallier S., *Implémentation d'un système préattentionnel avec des neurones impulsionnels*, Thèse de doct. Université Paris Sud 11, Orsay, 25 juin 2009
-

- 
- [Chen et al., 2002] S.-H. Chen, I-F. Wang, *A Wavelet-Based Voice Activity Detection Algorithm in Noisy Environments*, IEEE Int. Conf. on Electronics, Circuits and Syst. , Dubrovnik, Croatie, Vol.3, pp.995-998, 2002.
- [Chen et al., 2005] Chen S.-H., Wu H.-T., Chen C.-H., Ruan J.C, Truong T.K., *Robust Voice Activity Detection Algorithm Based on The Perceptual Wavelet Packet Transform*, IEEE Int. Symp. on Intelligent Signal Processing and Communication Systems, Hong Kong, Chine, pp.45-48, 2005.
- [Chiodi, 2010] Chiodi R.. *Détection D'activité Vocale Basée sur la transformée enondelettes*. Mémoire de maîtrise en Génie Électrique, Université du Québec à Trois-Rivières, Septembre 2010.
- [Cho et al., 2001] Cho Y.D., Al-Naimi K., Kondoz, A., *Improved Voice Activity Detection Based on a Smoothed Statistical Likelihood Ratio*. In Proceeding of the International Conference on Acoustics, Speech and Signal Processing, pp.737-740, 2001.

---

**D**

- [DasGupta et al., 1992] DasGupta B., Schnitger G. *The power of approximating: a comparison of activation functions*, Advances in Neural Information Processing Systems, vol. 5 p.363-374, 1992
- [Das et al., 1993] Das S., Bakis R., Nadas A., Nahamoo D., Picheny M., *Inuence of Background Noise and Microphone on the Performance of the IBM Tangora Speech Recognition System*. In Proc. ICASSP, 1993
- [Dubois, 2006] Dubois M., *Simulation évènementielle de réseaux de neurones impulsionnels*. Rapport technique, UNIC, CNRS, GifsurYvette., Chicago, Illinois, USA, June 27-29, 2006. Available from <https://perso.limsi.fr/Individu/dubois/manuscrit/Rapport.pdf>

---

**E**

- [Ermentrout et al., 2005] Ermentrout B., Gutkin B., Pinto D., *Mathematical neuroscience: from neurons to circuits to systems*, J Physiol Paris, Vol. 97, pp.190-209, 2005
- [ETSI, 1999] ETSI EN 301 708 Recommendation. Voice Activity Detector (VAD) for Adaptive MultiRate (AMR) Speech Traffic Channels.

---

**F**

- 
- [Fallahi et al., 2011] Fallahi A., Jafari S., *An Expert System for Detection of Breast Cancer Using Data Preprocessing and Bayesian Network*, International Journal of Advanced Science and Technology, vol. 34, pp.65-70,2011
- [Fawcett , 2003] Fawcett T., *ROC Graphs: Notes and Practical Considerations for Researchers*, HP Labs Tech Report HPL-2003-4,2003
- [Fourcaud-Trocme et al., 2003] Fourcaud-Trocme, N., Hansel, D., Van Vreeswijk, C., and Brunel, N. *How spike generation mechanisms determine the neuronal response to fluctuating inputs. J. Neurosci.* 23, 11628–11640, 2003
- [Fu et al. , 2007] Fu Y., Zhou X., Liu M., Hasegawa-Johnson M., S.Huang T., *Lipreading by Locality Discriminant Graph*, in Proceedings of IEEE International Conference on Image Processing (ICIP), 2007
- 

## G

- 
- [Gautrais et al., 1998] Gautrais J., Thorpe S., *Rate coding versus temporal order coding: a theoretical approach*, BioSystems, Vol.48, pp.57–65, 1998
- [Gerstner et al., 1992] Gerstner W, van Hemmen JL., *Associative memory in a network of spiking neurons*. Network, vol. 3, pp.139-164, 1992
- [Gerstner, 1995] Gerstner W., “Time structure of the activity in neural network models”, *Physical Review E*, vol.51, no.2, pp.738-758, 1995
- [Gerstner et al., 1996] Gerstner W., Kempter R., Van Hemmen L.J., Wagner H., “A neuronal learning rule for sub-millisecond temporal coding”, *Nature*, vol.384, pp.76–78, 1996.
- [Gerstner, 1998] Gerstner W ., “Spiking neurons”, W. Maass and C.M. Bishop (Editors), *Pulsed Neural Networks*, MIT press, pp. 3-54, 1998.
- [Gerstner et al., 1999] Gerstner, W. , Kempter, R., Leo van Hemmen, J. & Wagner, H. Hebbian Learning of Pulse Timing in the Barn Owl Auditory System in Maass, W. & Bishop, C. M. (eds.) *Pulsed Neural Networks*, MIT-press, 1999
- [Gerstner et al., 2002] Wulfram Gerstner and Werner M. Kistler. *Spiking Neuron Models*. The Cambridge, University Press, Cambridge, 1st edition, 2002
- [Giulioni, 2006] Giulioni M., *Networks of spiking neurons and plastic synapses: implementation and control*. Université de ZURICH. Rome 28 Mars 2008.
- [Gorriz et al., 2010] Gorriz, J.M., Ramirez, J., Lang, E.W., Puntinet, C.G., Turiyas, I., *Improved Likelihood Ratio Test Based Voice Activity Detector*
-

- 
- Applied to Speech Recognition, Speech Communication, Vol.52, pp.664–677, 2010.*
- [Graf et al., 2015] Graf S., Herbig T., Buck M., Schmidt G., *Features for voice activity detection: a comparative analysis*, EURASIP Journal on Advances in Signal Processing, Vol.91, 2015
- [Groupe Réseaux De Neurones, 1991] Groupe Reseaux De Neurones, *Les Réseaux De Neurones*, rapport de synthèse de l'OFTA (Observatoire Français Des Techniques Avancées). Ed. MASSON Paris, Milan, Barcelone, Bonn, 1991
- [GSC, 2001] Hoffman M.W., Li Z., Khataniar D., *GSC-Based Spatial Voice Activity Detection for enhanced speech coding in the presence of competing Speech*, IEEE Transactions on speech and audio processing, Vol. 9, No.2, pp.175-178, 2001.
- 

## H

- 
- [Hamzé , 2007] Hamzé A., *Application de la transformée en nombres entiers à la conception d'algorithmes de faible complexité pour l'annulation d'échos acoustiques*, thèse de doctorat, université de Bretagne occidentale, France, 2007
- [Hindmarsh et al., 1982] J. L. Hindmarsh et R. M. Rose., *A model of the nerve impulse using two first-order differential equations*, Network, vol. 296, pp.162–164, 1982
- [Hindmarsh et al., 1984] J. L. Hindmarsh et R. M. Rose., *A model of neuronal bursting using three coupled first order differential equations*. In *Proceedings of the Royal Society, London*, vol. 221, pp.87–102, 1984
- [Hodgkin et Al., 1952] Hodgkin, A. L. and Huxley, A. F., *A quantitative description of ion currents and its applications to conduction and excitation in nerve membranes*, J. Physiol. London, Vol.117, pp.500-44, 1952
- [Hu et al., 2007] Hu Y., Loizou P.C., *Subjective Comparison Of Speech Enhancement Algorithms*, Speech Commun., Vol.49, No.7, pp.588–601, 2007.
- 

## I

- 
- [Izhikevich, 2003] Izhikevich E.M., *Simple model of spiking neurons*. *IEEE Transactions on Neural Networks*, Vol.14, No.6, pp.1569–1572, Nov 2003
- [Izhikevich, 2004] Izhikevich E.M., *Which model to use for cortical spiking neurons*, *IEEE Transactions on Neural Networks*, Vol.15, No.5, pp.1063–1070, Sep 2004.
-

## J

- 
- [Jo et al., 2010] Jo S.H., Chang T., Ebong I., Bhadviya B. B., Mazumder P., Lu W., *Nanoscale memristor device as synapse in neuromorphic systems*. Nano Letters, Vol.10, No.4, pp.1297–1301, 2010. doi:10.1021/nl904092h.
- 

## K

- 
- [Karabatak et al., 2009] Karabatak M., Cevdet M., *An expert system for detection of breast cancer based on association rules and neural network*, Expert Systems with Applications, Vol.36, no.2, pp.3465–3469, 2009.
- [Karray et al., 2003] Karray L., Martin A., *Towards improving speech detection robustness for speech recognition in adverse conditions*, Speech Communication, Vol.40, No.3, pp.261-276, 2003.
- [Kasiński et al., 2006] Kasiński A., Ponulak F., *Comparison of supervised learning methods for spike time coding in spiking neural networks*, International Journal of Applied Mathematics and Computer Science, Vol. 16, pp. 101-113, 2006
- [Kepecs et al., 2002] Kepecs A., Van Rossum M., Song S., Tegner J., *Spike-timing dependent plasticity: common themes and divergent vistas*, Biological Cybernetics, Vol. No.87, pp.446–458, 2002.
- [Kharbat et al., 2014] Kharbat F., Odeh M., Bull L., *A New Hybrid Architecture for the Discovery and Compaction of Knowledge: Breast Cancer Datasets Case*, The International Arab Journal of Information Technology, Vol. 11, no. 2, pp. 116-123, 2014
- [Kim et al., 2005] Kim H.I., Park K.S., *Voice Activity Detection Algorithm Based on Radial Basis Function Network*, IEICE Trans. Communication, Vol. E88- B, No. 4, pp. 1656-1657, 2005.
- [Kistler, 2002] Kistler W., *Spike-timing dependent synaptic plasticity: a phenomenological framework*, Biological Cybernetics, Vol.87, pp.416–427, 2002.
- [Kiyani et al., 2004] Kiyani T., Yildirim T., *Breast Cancer Diagnosis Using Statistical Neural Networks*, Journal of Electrical and Electronics Engineering, .vol. 4, no.2, pp.1149-1153, 2004.
- [Knight, 1972] B. Knight, *Dynamics of encoding in a population of neurons*, J Gen Physiol, Vol.59, pp. 734–66, 1972
- [Kohonen, 1982] Kohonen T., *Self-organized formation of topologically correct feature maps*, Biological Cybernetics, Vol.43, pp.59–69, 1982.
-

- 
- [Kruger et al, 1988] Kruger, J., Aiple, F., *Multimicroelectrode investigation of monkey striate cortex: spike train correlations in the infra-granular layers*, Journal of Neurophysiology. vol. 60, p.798-828.1988
- [Kuzum et al., 2012] Kuzum D., eyasinghR. G.D.J, Lee B., Wong H.-S.P., *Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing*, Nano Letters, Vol.12, No.5, pp.2179–2186, 2012. doi:10.1021/nl201040y.
- 

## L

- 
- [Lam et al., 1995] Lam L. , Suen Ching Y., *An Evaluation Of Parallel Thinning Algorithms For Character Recognition*, IEEE Transaction On Pattern Analysis And Machine Intelligence, Vol.17, No.9 , pp.914-919, 1995,.
- [Loiselle, 2004] Loiselle S., *Exploration de réseaux de neurones à décharges dans un contexte de reconnaissance de parole*, Mémoire de maîtrise, université du Québec. 2004.
- [Legenstein et al., 2003] Legenstein R.A., Markram H., Maass W., *Input prediction and autonomous movement analysis in recurrent circuits of spiking neurons*, Reviews in the Neurosciences, Vol.14, No.1-2, pp.5–19, 2003.
- [Luu, 2010] Luu K. Y., *Real-Time Noise-Robust Speech Detection*, Master of Engineering in Electrical Engineering and Computer Science, MIT, 2010.
- 

## M

- 
- [Maass, 1997] Maass W., *Networks of spiking neurons: The third generation of neural network models*, Neural Networks. vol. 10, no.9, pp.1659-1671, 1997.
- [Maass et al., 1991] Maass, W., Schnitger, G. & Sontag, E., *On the computational power of sigmoid versus boolean threshold circuits*, Proc. of the 32<sup>nd</sup> Annual IEEE Symposium on Foundations of Computer Science, p. 767-776 1991.
- [Maass et al., 1999] Maass W., Schmitt M., *On the complexity of learning for spiking neurons with temporal coding*, Information and Computation, Vol.153, pp.26–46, 1999
- [Maass et al., 2004] Maass W., Natschlaeger T., Markram H., *Computational models for generic cortical microcircuits*. In J. Feng, editor, Computational
-

- 
- Neuroscience: A Comprehensive Approach, Chapman and Hall/CRC, Boca Raton, 2004.
- [Markram , 2002] Markram H., *Structural and functional principles of neocortical microcircuits*, Special issue on Foundations of Information Processing of TELEMATIK, Vol.8, No.1, 2002.
- [Martinez, 2006] Martinez, R. *Apprentissage Dynamique dans les Réseaux de Neurones Temporels*. Master's thesis, Université Claude Bernard Lyon 1, 2006.
- [Marcano-Cedeño et al., 2011] Marcano-Cedeño A., Quintanilla-Domínguez J., Andina D., *WBCD breast cancer database classification applying artificial metaplasticity neural network*, Expert Systems with Applications, vol.38, no.8, pp.9573-9579, 2011.
- [McKennoch et al., 2009] McKennoch S., Voegtlin T., Bushnell L.G., *Spike-Timing Error Backpropagation in Theta Neuron Networks*, Neural Computation, Vol. 21, No.1, pp.9-45, 2009.
- [Meftah, 2011] Meftah B., *Contribution des Réseaux de Neurones Temporels dans le Traitement des Images Animées*, thèse de doctorat, Université des Sciences et de la Technologie d'Oran- Mohamed Boudiaf, 2011
- [Meftah et al., 2010] Meftah B., Lezoray O., Lecluse M., Benyettou A., *Cell Microscopic Segmentation with Spiking Neuron Networks*, 20th International Conference on Artificial Neural Networks (ICANN'2010), Thessaloniki, Greece, K. Diamantaras, W. Duch, L.S. Iliadis (Eds.), Part I, Lecture Notes in Computer Science LNCS 6352, pp. 117-126, September 15-18, 2010.
- [Mesgarani et al., 2006] Mesgarani N., Slaney M., Shamma S.A., *Discrimination of speech from nonspeech based on multiscale spectro-temporal modulations*, IEEE Trans. Audio, Speech Lang. Process, Vol.14, No.3, pp.920-930, 2006.
- [Minsky et al., 1969] Minsky M.L., Papert S., *Perceptrons: an introduction to computational geometry*, Cambridge, MA: MIT Press, 1969
- [Minsra, 2012] Misra A., *Speech/Nonspeech Segmentation in Web Videos*, In Proceedings of Interspeech, 2012.
- [Moore, 2002] Moore S. C., *Back-propagation in spiking neural networks*. MSc thesis, University of Bath, UK, 2002.
- [Mouraud, 2009] Mouraud A., *Approche distribuée pour la simulation événementielle de réseau de neurone impulsif*, Thèse de Doctorat, Université des Antilles et de la Guyane, 2009
- [Moreno et al., 1994] Moreno P.J., Stern R.M., *Sources of Degradation of Speech Recognition in the Telephone Network*. In Proc. ICASSP, 1994.
-

---

[Morris et al., 1981] Morris C., Lekar H., *Voltage oscillations in the barnacle giant muscle fiber*. Biophysical Journal, Vol.35, pp.193–213, 1981.

---

## N

[Natschläger et al., 1998] Natschläger T., Ruf B., *Spatial and temporal pattern analysis via spiking neurons*, Network: Computation in Neural Systems, Vol.9, No.3, pp.319–332, 1998.

---

## O

[O'Halloran, 2011] O'Halloran M., McGinley B., Conceicao R.C., Morgan F., Jones E., Glavin M., *Spiking Neural Networks for Breast Cancer Classification in a Dielectrically Heterogeneous Breast*, Progress In Electromagnetics Research C, Vol. 113, pp.413-428, 2011.

[Ourdighi et al., 2009] Ourdighi A., Lacheheb S.E., Benyettou A., *Phonetic Classification with Spiking Neural Network Using a Gradient Descent Rule*, International Conference on Computer and Electrical Engineering, Vol. 2, pp.36-40, 2009

[Ourdighi et al., 2012] Ourdighi A., Barbari B., Benyettou A., *An Automatic Lipreading System For Spoken Isolated Word Recognition Using Spiking Neural Network*», ICIST'2013 : Conférence Internationale sur les Systèmes d'Information et les Technologies, Tanger, Maroc, 22- 24 Mars 2013.

[Ourdighi et al., 2013] Ourdighi A., Benyettou A., *Effect of the Neuron Coding by Gaussian Receptive Fields on Enhancing the Performance of Spiking Neural Network for An Automatic Lipreading System*, Oriental Journal of Computer Science and Technology, Vol.06, No.03, pp. 287-294, 2013

[Ourdighi et al., 2016] Ourdighi A., Benyettou A., *An Efficient Spiking Neural Network Approach based on Spike Response Model for Breast Cancer Diagnostic System*, Arab J. Inf. Technol., Vol.13, No. 3, pp.284-290, 2016.

---

## P

[Ponulak et al., 2010] Ponulak F., kasinski A. *Supervised Learning in Spiking Neural Networks with ReSuMe: Sequence Learning, Classification, and Spike Shifting*, Neural Computation, Vol.22, No.2, pp.467-510, 2010.

---

- 
- [Purves et al., 2004] Purves D., Augustine G.J., Fitzpatrick D., Hall W.C., LaMantia A.S.I, McNamara J.O., and Williams S. M., editors. *Neuroscience*. Sinauer Associates, Inc., 3rd edition, 2004
- 

## R

---

- [Rabiner et al., 1975] Rabiner L.R., Sambur M.R., *An algorithm for determining the endpoints of isolated utterances*, Bell Syst. Tech. J, Vol.54, No.2, pp.297–315, 1975.
- [Rabiner et al., 1977] Rabiner L.R., Sambur M.R., *Application of an LPC distance measure to the voiced-unvoiced-silence detection problem*, IEEE Trans. Acoust. Speech Signal Process, Vol.25, No.4, pp.338–343, 1977.
- [Ramirez et al., 2003] Ramirez J., Segura J. C., Bentez M. C., Rubio A.J., *A new adaptive long-term spectral estimation voice activity detector*, In Proceedings of Eurospeech, Geneva, Switzerland, 2003
- [Ramirez et al., 2004] Ramírez J., Segura J.C., Benítez C., de la Torre Á., Rubio A., *Efficient voice activity detection algorithms using long-term speech information*, Speech Commun, Vol.42, pp.271–287, 2004.
- [Ramirez et al.a, 2007] Ramirez J., Gorriz J. M., and Segura J. C., *Voice Activity Detection. Fundamentals and Speech Recognition System Robustness*. Robust Speech Recognition and Understanding. pp. 1-22, Ed.: M. Grimm and K. Kroschel, I-TECH Education and Publishing, ISBN: 978-3-902613-08-0, 2007.
- [Ramirez et al.b, 2007] Ramirez, J., Segura, J.C., Gòrriz, J.M., Garcia, L., *Improved Voice Activity Detection Using Contextual Multiple Hypothesis Testing for Robust Speech Recognition*, IEEE Transactions on Audio Speech and Language Processing, Vol.15, No.8, pp.2177–2189, 2007.
- [Rochel, 2004] Rochel O., *Une approche évènementielle pour la modélisation et la simulation de réseaux de neurones impulsionnels*. Thèse de doctorat de l'université Henri Poincaré - Nancy1. 15 mars 2004
- [Rosado-Munoz, 2011] Rosado-Munoz A., Fijalkowski AB., Bataller-Mompean M., Guerrero-Martinez J., *FPGA implementation of Spiking Neural Networks supported by a Software Design Environment*, The 18 th IFAC World Congress, Milano, Italy, Vol.18 ,pp. 1934-1939, 2011
- [Rubio et al.,2007] Rubio J.E., Ishizuka K., Sawada H., Araki S., Nakatani T., Fujimoto M., *Two Microphones Voice Activity Detection Based on The Homogeneity of The Direction of Arrival Estimates*, IEEE Int. Conf. on Acoustic, Speech and Signal Processing, Honolulu, USA, pp.385-
-

- 
- 388, 2007.
- [Rudolph et al., 2006] Rudolph M., Destexhe A., *The discharge variability of neocortical neurons during high-conductance states*, Neuroscience, Vol.119 pp. 855–73, 2003
- [Ruf et al., 1997] Ruf B., Schmitt M., *Learning temporally encoded patterns in networks of spiking neurons*. Neural Processing Letters, Vol.5, No.1, pp.9-18, 1997.
- [Ruf et al., 1998] Ruf B., Schmitt M., *Self-organisation of spiking neurons using action potential timing*, IEEE Transactions on Neural Networks, Vol.9, No.3, pp.575–578, 1998.
- [Rumelhart et al., 1986] Rumelhart D., Hinton G., Williams R., *Learning internal representations by error propagation*, Parallel Distributed Processing, D. Rumelhart and J. McClelland Eds. Cambridge: MIT Press, Vol. 1, pp. 318-362, 1986.
- 

## S

- 
- [Schrauwen, 2002] Schrauwen B., *Pulstreincodering en training met meerdere datasets op de cbm*, Master's thesis, Ghent University. 2002.
- [Schrauwen et al., 2002] Schrauwen B., Campenhout J.V., *Improving spikeprop: enhancements to an error-backpropagation rule for spiking neural networks*. In Proceedings of the 15 ProRISC Workshop, 2004.
- [Setiono, 2000] Setiono R., *Generating concise and accurate classification rules for breast cancer diagnosis*, Artificial Intelligence in Medicine, Vol.18, no.3, pp.205–219, 2000
- [Shimizu et al., 2000] Shimizu M., Fukuda H., Nakamura G., *Thinning Algorithm for Digital Figures of Characters*, Proceeding 4th IEEE Southwest Symposium on Image Analysis and Interpretation, pp. 83-87, 2000.
- [Sohn et al., 1999] Sohn J., Kim N.S., Sung W., *A statistical model-based voice activity detection*, IEEE Signal Process. Lett., Vol.6, No.1, pp.1–3, 1999.
- [Stromatias, 2011] Stromatias E., *Developing a supervised training algorithm for limited precision feed-forward spiking neural networks*, Master thesis, Liverpool university, 2011
- [Schwemmer, 2010] Schwemmer M. A., *The Influence of Dendritic Properties on the Dynamics of Oscillatory Neurons*, PHD Dissertation, University Of California, 2010.
- [Sujatha et al., 2010] Sujatha B., Santhanam T., *A Novel Approach Integrating Geometric and Gabor Wavelet Approaches to Improvise Visual Lipreading*,
-

---

International Journal of Soft Computing (IJSC), Vol.42, No.4,  
pp.622-628, 2010

---

## T

---

- [Tanyer et al.,2000] Tanyer S.G., Ozer H., *Voice Activity Detection in NonStationary Noise*, IEEE Transactions on speech and audio processing, Vol.8, No.4, pp.478-482, juillet 2000.
- [Touzet, 1992]. Touzet C., *Les réseaux de neurones artificiels, Introduction au connexionnisme*, Cours, exercices et travaux pratiques, Juillet 1992
- [Thorpe et al., 1997] Thorpe S.J., Gautrais J., *Rapid visual processing using spike asynchrony*, In: M.C. Mozer, M.I. Jordan, T. Petsche (Eds.), *Advances in Neural Information Processing Systems 9*, MIT Press, Cambridge, MA, pp. 901–907, 1997.
- [Thorpe et al., 1998] Thorpe, S.J., Gautrais, J., *Rank Order Coding: A new coding scheme for rapid processing in neural networks*, In: J. Bower (Ed.), *Computational Neuroscience: Trends in Research 1998*, Plenum Press, New York, pp. 113–118, 1998.
- [Tucker, 1992] Tucker R., *Voice activity detection using a periodicity measure*, IEEE Proc. Commun. Speech Vis, Vol.139, No.4, pp.377–380, 1992.

## U

---

- [ITU-T, 1996] ITU-T Recommendation G.729-Annex B. A silence compression scheme for G.729 optimized for terminals conforming to recommendation V.70.

## V

---

- [Van Segbroeck et al., 2013] Van Segbroeck M., Tsiartas A., Narayanan S.S., *A robust frontend for VAD: exploiting contextual, discriminative and spectral cues of human voice*, in *Proc. of INTERSPEECH.*, ISCA, Lyon, France, 2013.
- [Verstraeten et al., 2005] Verstraeten D., Schrauwen B., Stroobandt B., *Isolated word recognition using a Liquid StateMachine*, Ghent University – ELIS. 2005
- [Vreeken, 2002] Vreeken J., *Spiking neural networks, an introduction*, Technical Report UU-CS-2003-008 Institute for Information and Computing Sciences, Utrecht University., 2002
-

## Y

- 
- [Yu et al., 2010] Yu S., Wong H-S. , *Modeling the switching dynamics of programmable metallization-cell (PMC) memory and its application as synapse device for a neuromorphic computation system*, In *Electron Devices Meeting (IEDM)*, 2010 IEEE International, pp. 22.1.1–22.1.4, 2010. doi:10.1109/IEDM.2010.5703410.
- 

## W

- 
- [Wang et al., 1996] Wang X., Buzsaki G., *Gamma oscillation by synaptic inhibition in a hippocampal interneuronal network model*, *Journal of Neuroscience*, Vol.16, No.20, pp.6402–6413, 1996.
- [WBCD., 2011] Wisconsin Breast Cancer Database (WBCD) from UCI Machine Learning Repository, available at: <http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29>, last visited: 2011
- [Widiarti, 2011] Widiarti A.R, *Comparing Hilditch, Rosenfeld, Zhang-Suen, and Nagendraprasad - Wang-Gupta Thinning*, *World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering* Vol.5, No.6, pp.563-567, 2011
- [Wu et al., 2006] Wu B.-F., Wang K.-C., *Voice Activity Detection based on Auto-Correlation Function Using Wavelet Transform and Teager Energy Operator*, *Computational Linguistics and Chinese Language Processing*, Vol.2 , No. 1, pp.87-100, 2006.
- 

## Z

- 
- [Zhang et al., 1984] Zhang T. Y. , Suen Ching Y., *A Fast Parallel Algorithms For Thinning Digital Patterns*, *Communication of the ACM*, Vol .27, No. 3, pp.236-239, 1984.
- [Zhao et al., 2009] Zhao G., Barnard M., Pietikäinen M., 2009, *Lipreading with Local Spatiotemporal Descriptors*, *IEEE Transactions on Multimedia*, pp. 1-11, 2009.
-

# PUBLICATIONS ET CITATIONS DE L'AUTEUR

## PUBLICATIONS

- A.Ourdighi; A. Benyettou, "An Adaptive Time-Delay Neural Network Training Using Parallel Genetic Algorithms in Time-Series Prediction and classification", Journal of Applied Sciences Volume 10, Number 18, 2115-2120, ISSN: 1812-5654, 2010  
<http://scialert.net/abstract/?doi=jas.2010.2115.2120>
- Asmaa Ourdighi; Abdelkader Benyettou, "Effect of the Neuron Coding by Gaussian Receptive Fields on Enhancing the Performance of Spiking Neural Network for An Automatic Lipreading System", Oriental Journal of Computer Science and Technology, Volume 06, Number 03: 287-294, ISSN : 0974-6471, 2013
- Asmaa Ourdighi, Abdelkader Benyettou," An Efficient Spiking Neural Network Approach based on Spike Response Model for Breast Cancer Diagnostic System", *The International Arab Journal of Information Technology*, publié mai 2016

## CONFERENCES ET COMMUNICATIONS

### ORGANISATIONS

- The 10th The Maghrebian Conference on Software Engineering and Artificial Intelligence, Oran , Algeria, 2008

### PRESENTATIONS

- A. Ourdighi, A. Benyettou, «Optimisation par les Algorithmes Génétiques de l'Apprentissage des Réseaux de Neurones à Délais Temporels Adaptatifs» ; MCSEAI'2006 – The 9th The Maghrebian Conference on Software Engineering and Artificial Intelligence, Maroc, 2006(Présentation Poster).

- A. Ourdighi, A. Benyettou, «*L'Intégration des Algorithmes Génétiques dans l'Apprentissage des Réseaux de Neurones à Délais Temporels Adaptatifs* » ; SETIT'2007 – Sciences Electroniques, Technologie de l'Information et des Télécommunications, 25-27 mars 2007 en Tunisie (Présentation orale).
- A. Ourdighi, A. Benyettou, «*Intégration des Algorithmes Génétiques dans les Réseaux de neurones à Délais Temporels Adaptatif pour la reconnaissance de la Parole* » ; MOAD'2007 – Colloque International sur les Méthodes et Outils d'Aide à la Décision, 18-20 Novembre 2007 béjaia ,Algérie (Présentation orale).
- Participante à l'école de printemps sur « informatique et bio-inspiré », Oran, 2008
- A. Ourdighi; A. Benyettou, « An Adaptive Time-Delay Neural Network Training Using Parallel Genetic Algorithms in Time-Series Prediction and classification », International Conference on Information & Communication Systems, ICICS'09, 20 Décembre 2009, Jordanie
- A. Ourdighi; A. Benyettou, « A Parallel Genetic Adaptive Time Delay Neural Network Learning Algorithm for Time-Series Prediction and classification », First International Conference on Image and Signal Processing and their Applications, ISPA'09, 19-21 Octobre 2009, Algérie
- A. Ourdighi; S.A Lacheheb; A. Benyettou, « Using Spiking Neurons Network for Phonetic Classification », International Arab Conference on Information Technology, ACIT'2009, 15-17 Décembre , Yemen
- A. Ourdighi; S.A Lacheheb; A. Benyettou, « Phonetic Classification with Spiking Neural Network Using a Gradient Descent Rule », the 2nd International Conference on Computer and Electrical Engineering, ICCEE 2009, 28 - 30, Décembre 2009, Dubai.
- Asmaa Ourdighi, Bessa Barbari, Abdelkader Benyettou, « *Reconnaissance labiale par les réseaux de neurones impulsionsnels à base de codage par rang et du modèle SRM* », CITIM'12: Conférence internationale sur le traitement de l'information multimédia, 09-11 Avril 2012, Mascara, Algérie
- Asmaa Ourdighi, Bessa Barbari, Abdelkader Benyettou, «*An Automatic Lipreading System For Spoken Isolated Word Recognition Using Spiking Neural Network* », ICIST'2013 : Conférence Internationale sur les Systèmes d'Information et les Technologies , 22- 24 Mars 2013, Tanger, Maroc

## CITATIONS

Ourdighi A Lacheheb S E, Benyettou A. Phonetic classification with spiking neural network using a gradient descent rule. 2009 International Conference on Computer and Electrical Engineering (ICCEE 2009), Dubai, UAE, ICCEE 2009, Vol.2, 2009:36-40.

1. Liu Wen-peng, Chen Xu, Lu Hua-xiang, *A New Hardware-oriented Spiking Neuron Model Based on SET and Its Properties*, Physics Procedia Volume 22, pp. 170-176, 2011.

2. Chaturvedi, S. ; Deptt. of E&C Eng., PIET, Nagpur, India ; Khurshid, A.A., Review of Spiking Neural Network Architecture for Feature Extraction and Dimensionality Reduction Emerging Trends in Engineering and Technology (ICETET), 2011

Ourdighi, A. and A. Benyettou, 2010. An adaptive time-delay neural network training using parallel genetic algorithms in time-series prediction and classification. J. Applied Sci., 10: 2115-2120. DOI: 10.3923/jas.2010.2115.2120

1. Salman Yussof, Rina Azlin Razali and Ong Hang See J., An Investigation of Using Parallel Genetic Algorithm for Solving the Shortest Path Routing Problem, Computer Sci., 7 (2): 206-215, 2011