



Approches neuromorphiques pour la vision par ordinateur

Stagiaire : Christophe PITON

Encadrants : Marius BILASCO, Pierre TIRILLY

Tuteur : Christophe VROLAND

Période : avril 2014 à juillet 2014

Licence mention Informatique, parcours INFO

Remerciements

Je remercie en premier lieu mon encadrant de stage, Marius BILASCO qui m'a accepté en stage au sein de l'équipe FOX et a supervisé mon travail.

Je tiens également à exprimer ma gratitude à mon tuteur de stage universitaire, Christophe VROLAND qui m'a donné de nombreux conseils tout au long du stage lors de nos points hebdomadaires.

Je remercie aussi Pierre TIRILLY pour avoir co-encadré mon stage au sein de l'équipe.

Enfin, merci à toute l'équipe FOX et particulièrement à Amel AÏSSAOUI, Rémi AUGUSTE et José MENNESSON pour m'avoir accueilli parmi eux, avoir rendu mon stage très agréable et répondu à mes questions.

Faire mon stage de licence dans cette équipe a été très agréable et riche d'expériences. J'ai découvert grâce à eux un nouvel aspect de l'informatique et ainsi pu adapter mon projet professionnel.

Sommaire

Introduction	3
1 Présentations	4
1.1 Laboratoire	4
1.2 Institut	5
1.3 Équipe	5
2 Activités	7
2.1 Objet	7
2.2 Travaux liés	8
2.2.1 Bases de neurobiologie	8
2.2.2 Réseaux de neurones artificiels	12
2.2.3 Applications à la vision par ordinateur	13
2.3 Matériel et méthodes	14
2.3.1 Brian	14
2.3.2 Maximum de trois nombres	15
2.3.3 Reconnaissance d'expressions faciales	21
2.4 Résultats	23
2.4.1 Maximum de trois nombres	23
2.4.2 Reconnaissance d'expressions faciales	29
2.5 Conclusion	29
2.6 Perspectives	30
3 Conclusion	31
Bibliographie	33

Introduction

Je suis actuellement une formation en licence informatique à l'Université Lille 1. J'ai précédemment occupé le poste d'analyste-développeur pendant plus d'un an dans une SSII¹. Initialement, mon but en suivant cette formation (et un master par la suite) était de me perfectionner et d'obtenir un poste d'expert technique dans les nouvelles technologies. Mais suite à plusieurs cours donnés par des enseignants-chercheurs, je suis maintenant de plus en plus attiré par le monde de la recherche.

L'un de mes enseignants en licence, monsieur BILASCO, était à la recherche de stagiaires pour son équipe. Après nous avoir exposé brièvement l'objet de ses recherches, la vision par ordinateur, j'ai rapidement postulé car le sujet a attisé ma curiosité. Ce stage se déroule donc au sein de l'équipe FOX du Laboratoire d'Informatique Fondamentale de Lille. Celle-ci est hébergée par l'Institut de Recherche sur les Composants matériels et logiciels pour l'Information et la Communication Avancée.

Ce stage a pour thème les approches neuromorphiques pour la vision par ordinateur. Je dois mener une recherche sur de nouveaux systèmes neuromorphiques et leurs applications dans la vision par ordinateur. Mon travail doit faciliter et accélérer de futures recherches de l'équipe. Ce stage est par nature un stage de recherche exploratoire.

Ce stage me permet principalement de découvrir l'univers des neurosciences qui m'était complètement inconnu jusque là. Il me donne également le moyen d'améliorer mes compétences de recherche d'informations. De plus, j'apprends à prendre du recul par rapport aux résultats des expériences pour les analyser, en tirer des conclusions et formuler des hypothèses. D'un point de vue technique, je compte perfectionner mes connaissances des langages Python et C++ et découvrir la bibliothèque de vision par ordinateur OpenCV.

1. Société de Services en Ingénierie Informatique.

Chapitre 1

Présentations

1.1 Laboratoire



Le Laboratoire d'Informatique Fondamentale de Lille (LIFL) est une Unité Mixte de Recherche (UMR 8022) avec comme tutelles le Centre National de la Recherche Scientifique (CNRS) et l'Université Lille 1. Le laboratoire travaille aussi avec l'Université Lille 3 et INRIA. Il a été créé en 1983 et compte environ 300 personnes dont une centaine de chercheurs et enseignants-chercheurs, trente ingénieurs, techniciens et administratifs et une centaine de doctorants. Les équipes de recherche travaillent autour de trois grandes thématiques :

- interaction, coopération, images,
- infrastructures logicielles et systèmes embarqués,
- modèles, algorithmes, calcul.

C'est l'un des plus importants laboratoires français de recherche en informatique au nord de Paris. Il a participé à la création d'un grand institut de recherche : l'Institut de Recherche sur les Composants matériels et logiciels pour l'Information et la Communication Avancée (IRCICA). Il participe activement à ses travaux.

1.2 Institut



L’Institut de Recherche sur les Composants matériels et logiciels pour l’Information et la Communication Avancée est une Unité de Service et de Recherche (USR 3380) du CNRS et de l’Université Lille 1. Il a été créé en 2003 et a tout d’abord été un groupement d’intérêts scientifiques, puis est devenu par la suite fédération de recherche. Il fonctionne comme un «hôtel à projets interdisciplinaires» en fédérant des laboratoires de recherche. Ainsi, il regroupe des enseignants-chercheurs, chercheurs, étudiants, ingénieurs et techniciens de quatre laboratoires partenaires :

- le Laboratoire d’Informatique Fondamentale de Lille (LIFL),
- l’Institut d’Électronique, de Microélectronique et de Nanotechnologie (IEMN),
- le Laboratoire de Physique des Lasers, Atomes et Molécules (PhLAM),
- le Laboratoire d’Électrotechnique et d’Électronique de Puissance de Lille (L2EP).

Son objectif est d’initier des projets de recherche interdisciplinaires associant les différentes communautés présentes, en particulier celles du matériel et du logiciel.

1.3 Équipe



L’équipe Fouille et indexation de dOcuments compleXes et multimédia (FOX) travaille sur l’extraction du comportement humain à partir des flux vidéo. L’équipe considère quatre grands domaines de validation : le regard, l’évènement, l’émotion et la reconnaissance de personnes. Leurs études sont réalisées à deux niveaux d’échelle : l’individu et la foule. Actuellement, l’équipe travaille principalement sur quatre projets :

- **EMPATHIC** : l’objectif de ce projet est de créer des applications capables de comprendre et de répondre aux intentions et aux émotions des utilisateurs afin de s’adapter en temps réel pour optimiser l’expérience des utilisateurs,

- **TWIRL** : le but est ici de résoudre des problématiques liées à la quantité de données disponibles sur Internet, l'hétérogénéité des formats, la multitude de sources,
- **PERCOL** : l'objectif de ce projet est de faciliter la reconnaissance de personnes dans des contenus audiovisuels,
- **PIRVI** : le but de ce projet est de mettre à disposition des équipements, des compétences et un cadre pour définir, piloter et réaliser des projets de développement.

Mon stage s'inscrit dans le cadre d'un projet interne sur les réseaux neuromorphiques.

Chapitre 2

Activités

2.1 Objet

L'objectif de la vision par ordinateur est de permettre à une machine de comprendre ce qu'elle «voit». C'est une vaste discipline dont les théories et les modèles s'appliquent à de nombreux systèmes comme des procédés de contrôle (robotique industrielle), la navigation (véhicule autonome, robot mobile), la détection d'évènements (surveillance, comptage automatique de personnes), l'organisation d'informations (indexation de bases de données d'images et de vidéos), la modélisation d'objets ou d'environnements (analyse d'images médicales), l'interaction (dispositif d'entrée pour une interface homme-machine) et l'inspection automatique (domaines de production).

La reconnaissance d'expressions faciales fait aussi partie de ses domaines d'application. Ces expressions peuvent être analysées pour en déduire une émotion ressentie par exemple. Actuellement, cette analyse demande beaucoup de ressources CPU (temps de calcul) à nos ordinateurs. L'équipe FOX a d'ailleurs déjà développé une application qui permet de déterminer si une personne est gaie ou surprise avec ce type d'analyse. Elle utilise un réseau de neurones, un modèle de normalisation de visages et un modèle géométrique de visages.

L'objectif de cette recherche est d'étudier une nouvelle piste dans la reconnaissance d'expressions faciales. L'utilisation de réseaux neuromorphiques de type STDP semble être une piste sérieuse. En effet, ces réseaux sont capables d'apprendre à reconnaître et à classer certaines formes comme des chiffres manuscrits [1]. De plus, les implémentations matérielles de ces réseaux nécessitent moins d'énergie et un temps de calcul moins important que les systèmes classiques. En effet, le cerveau consomme

une vingtaine de watts [15] alors qu’il traite simultanément des informations venant de nombreux capteurs.

2.2 Travaux liés

Les systèmes neuromorphiques sont des solutions matérielles ou logicielles s’inspirant fortement du cerveau. Le principal intérêt de se rapprocher du fonctionnement du cerveau est que ce dernier ne demande qu’une très faible quantité d’énergie pour fonctionner [15] et traiter une très grande quantité d’informations contrairement aux ordinateurs et supercalculateurs actuels. De plus, leur capacité de classification et de généralisation est un atout pour résoudre des problèmes de nature statistique comme la classification automatique de codes postaux ou la prise de décision concernant un achat boursier en fonction de l’évolution des cours.

2.2.1 Bases de neurobiologie

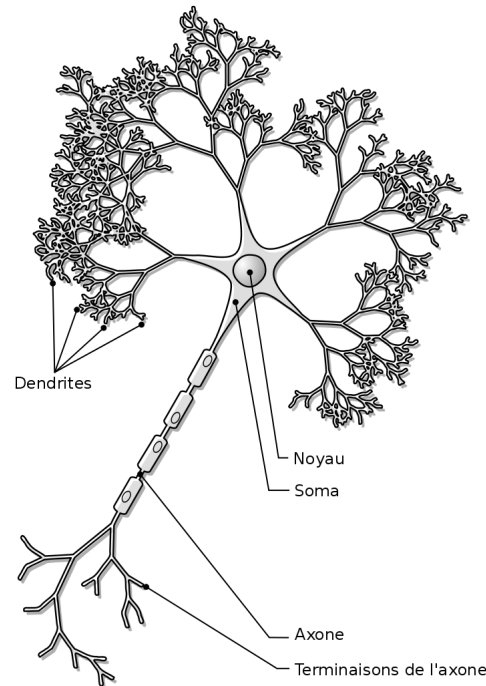
Afin de bien comprendre le fonctionnement des systèmes neuromorphiques, il est nécessaire de poser au préalable quelques notions de neurobiologie.

Neurones et synapses

Le terme de neurone a été introduit par Wilhelm von WALDEYER, l’un des fondateurs de la «théorie du neurone», en 1891. Il désigne par là les unités structurales et fonctionnelles de base du système nerveux. Il s’est basé sur les travaux de Santiago Ramón y CAJÀL à la fin du XIX^e siècle [4]. Ce dernier a eu l’idée d’utiliser une méthode de coloration inventée à la même époque par Camillo GOLGI pour étudier en détails l’anatomie des neurones. C’est au début du XX^e siècle que Charles Scott SHERRINGTON introduit le terme de synapse pour désigner la zone qui sépare les neurones devinée par CAJÀL.

Les neurones constituent la base du système nerveux. Ils permettent la transmission de signaux bioélectriques, les influx nerveux ou potentiels d’action. Ils sont principalement constitués de trois parties, le corps cellulaire (ou soma), l’axone et les dendrites (voir figure 2.1).

Une synapse est une zone de contact fonctionnelle qui s’établit entre deux neurones ou entre un neurone et une cellule. Elle assure la conversion d’un influx nerveux déclenché dans le neurone présynaptique en un signal dans la cellule postsynaptique.



(Source : Wikipedia, Nicolas.Rougier)

FIGURE 2.1 – Le soma abrite entre autres le noyau de la cellule. L'axone est un prolongement unique du soma et il permet de transmettre l'information électrique et chimique aux neurones qui l'entoure. Les dendrites sont des ramifications courtes et arborescentes et servent à recevoir les informations véhiculées par les axones d'autres neurones.

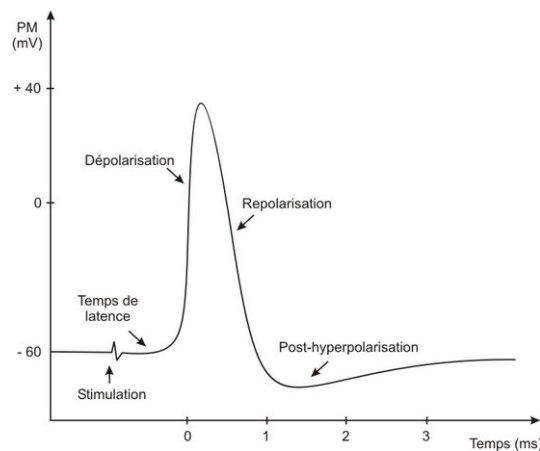
On appelle fente synaptique l'espace entre le neurone présynaptique et le neurone postsynaptique. On distingue deux types de synapses par la taille de la fente synaptique, les synapses chimiques (entre 10 et 40 nanomètres) et les synapses électriques (environ 2 nanomètres). Les synapses chimiques sont très majoritaires.

Les synapses peuvent être excitatrices, dans le cas où elles favorisent le déclenchement d'un influx nerveux par le neurone post-synaptique. Elles peuvent être aussi inhibitrices, lorsqu'elles freinent le déclenchement d'un influx nerveux par le neurone post-synaptique.

Influx nerveux

Au repos, il existe une différence de potentiel négative (entre -60 mV et -90 mV environ) entre les deux faces de la membrane du neurone. C'est le potentiel de repos. L'influx nerveux se caractérise par une modification instantanée et localisée de la perméabilité de la membrane du neurone (voir figure 2.2). Le potentiel de membrane prend alors une valeur positive (environ +35 mV) : on parle de dépolarisation. Il décroît ensuite pour aboutir à une valeur plus basse que la valeur du potentiel de

repos : on parle de repolarisation et d'hyperpolarisation. Puis, il y a une phase de retour à la normale.



(Source : Passeport Université Lille 1 [6])

FIGURE 2.2 — Différence de potentiel à la membrane d'un neurone. Une stimulation fait déclencher un influx nerveux à ce neurone (phase de dépolarisation et de repolarisation).

Plasticité synaptique

Les synapses ne sont pas fixes. Elles peuvent évoluer, se reconfigurer au fil du temps (création de nouvelles ou destruction d'existantes). De plus, l'efficacité d'une synapse à transmettre un influx nerveux (on parle du poids synaptique) peut aussi varier. Ces phénomènes sont appelés plasticité synaptique.

C'est en 1949 que Donald HEBB [16] émet la première hypothèse d'une plasticité qui modifierait l'efficacité synaptique. Nous pouvons énoncer son postulat comme suit : lorsque l'axone d'une cellule A est suffisamment proche d'une cellule B et qu'elle prend part de manière répétée et persistante dans son activation, des processus de croissance ou des changements métaboliques ont lieu, dans une seule ou les deux, de telle sorte que l'efficacité de A, en tant qu'activateur de B, est augmentée.

Plusieurs formes de plasticité ont depuis été découvertes :

- **habituatation** : atténuation de la réponse post-synaptique au fil des stimuli constants et répétés provenant des afférences.
- **sensibilisation (ou désensibilisation)** : induction de réactions plus vives (ou moins vives) face à divers stimuli sensoriels neutres suite, classiquement, à une première sommation sensorielle brusque et inopinée (stimulus inconditionné, douloureux).

- **potentialisation** : augmentation d'amplitude de la réponse post-synaptique à la suite d'une intense activation concertée de la synapse.
- **dépression** : diminution de l'amplitude du signal transmis entre deux neurones.

Cette plasticité est le phénomène qui nous permet d'apprendre. En effet, elle permet d'expliquer de nombreuses formes de mémoire simples présentes chez tous les êtres vivants présentant un système nerveux, même peu développé.

Spike-Timing-Dependent Plasticity (STDP)

La STDP est une forme de plasticité particulière. La modification de l'efficacité synaptique dépend du moment de déclenchement de l'influx nerveux dans les neurones pré-synaptique et post-synaptique (voir figure 2.3). Dans le cas où le neurone pré-synaptique déclenche un influx nerveux avant que le neurone post-synaptique n'en déclenche un autre, l'efficacité synaptique est augmentée. Dans le cas contraire, l'efficacité synaptique est diminuée. De plus, l'amplitude de la modification dépend de l'écart de temps entre les deux déclenchements d'influx nerveux. Plus l'écart est petit, plus la modification sera importante. À l'inverse, un écart important entraînera une modification minime voir aucune.

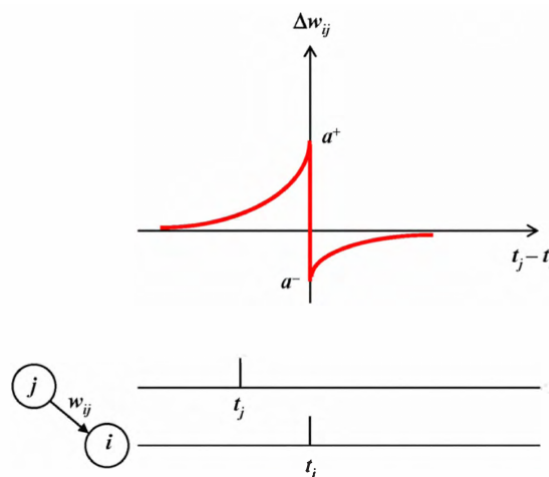
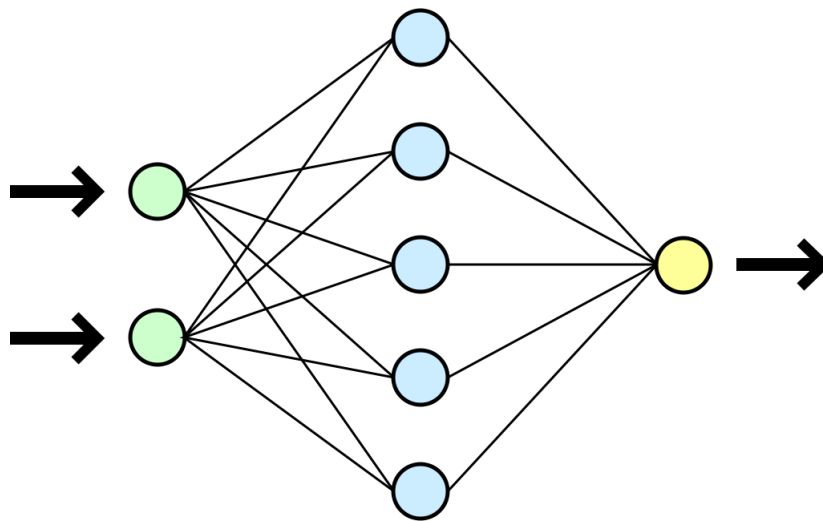


FIGURE 2.3 — Courbe de modification du poids synaptique STDP. L'abscisse représente l'écart de temps entre le déclenchement d'un influx nerveux par les neurones pré-synaptique (j) et post-synaptique (i). L'ordonnée représente l'amplitude de la modification du poids synaptique.

2.2.2 Réseaux de neurones artificiels

Les réseaux de neurones artificiels sont des modèles de calcul basés sur le fonctionnement des neurones biologiques. Ils sont constitués de différentes couches de neurones (exemple en figure 2.4). Leurs principaux atouts sont leurs capacités de classification et de généralisation. En effet, ils sont capables, à partir d'exemples, d'apprendre à classer des données. Classifier des données consiste à attribuer une classe ou catégorie à chaque élément de ces données. De plus, ils sont aussi en mesure de généraliser leur apprentissage, c'est-à-dire qu'ils peuvent également classer des cas alors que ces derniers ne faisaient pas partie de leur jeu d'apprentissage.



(Source : Wikipédia, Dake, Mysid)

FIGURE 2.4 – Exemple de réseau de neurones artificiels avec trois couches différentes. La première couche de deux neurones constitue l'entrée. Il y a une couche intermédiaire de cinq neurones. La couche de sortie ne contient qu'un seul neurone.

Apprentissage

L'apprentissage est un processus qui consiste à modifier les poids synaptiques en fonction d'un jeu de données présentées en entrée du réseau. Le but est de permettre au réseau de neurones d'apprendre à partir d'exemples. Il existe plusieurs types d'apprentissage dont deux principaux, l'apprentissage supervisé et l'apprentissage non-supervisé.

Dans un apprentissage supervisé, on force le réseau à converger vers un état final précis en même temps qu'on lui présente un motif. Concrètement, on va faire déclencher un ou plusieurs influx nerveux aux neurones de sortie souhaités en même temps qu'on présente au réseau un motif.

Lors d'un apprentissage non-supervisé, le réseau est laissé libre de converger vers n'importe quel état final lorsqu'on lui présente un motif.

Les travaux de MASQUELIER *et al.* [12, 13, 14] ont montré que la plasticité synaptique de type STDP permet de mémoriser et de détecter des motifs particuliers parmi des trains d'influx nerveux.

Différents modèles

Afin de représenter le fonctionnement d'un neurone biologique et de s'en rapprocher au plus, des chercheurs ont mis au point différents modèles mathématiques comme des modèles formels tel le Perceptron [18]. Les modèles impulsionnels sont maintenant privilégiés.

On peut décrire l'un des plus simples : le modèle *Intègre et tire avec fuite*. Dans ce modèle, l'influx nerveux est considéré comme un événement défini seulement par son instant d'apparition. Les phases de dépolarisation et de repolarisation ne sont pas représentées. Ainsi, ce modèle peut s'écrire :

$$\tau \frac{dV}{dt} = -(V(t) - V_S) + I(t)$$

$$\text{si } V \geq V_t, \text{ alors } V = V_r$$

où V_S est le potentiel de repos en l'absence de courant I , V_t est le potentiel de décharge, V_r est le potentiel de retour et τ la constante de temps du neurone. $V(t)$ est le potentiel de membrane du neurone. Le neurone déclenche un influx nerveux lorsque son potentiel de membrane dépasse le seuil V_t .

Notons que d'autres modèles existent afin de mieux représenter certains phénomènes neurobiologiques ou biophysiques. Nous pouvons citer par exemple le modèle de HODGKIN-HUXLEY [10] ou le modèle d'IZHIKEVICH [11].

2.2.3 Applications à la vision par ordinateur

Grâce à leurs capacités de classification et de généralisation, les systèmes neuro-morphiques présentent de gros atouts pour la vision par ordinateur. Ils sont d'ailleurs déjà utilisés dans ce domaine pour effectuer certains traitements.

Extraction et apprentissage de trajectoires

Les travaux de BICHLER *et al.* [2] consistent à faire apprendre à un réseau neuromorphique de type STDP à extraire les trajectoires de voitures sur une autoroute. Leur apprentissage est fait de manière non-supervisée. Leur système est capable de compter le nombre de véhicules qu'il détecte après seulement 10 minutes d'apprentissage.

Simulation de traitements visuels

Les travaux de RICHERT *et al.* [17] ont permis de développer un simulateur de réseaux neuromorphiques en C/C++. Pour montrer l'efficacité et la facilité d'utilisation de leur simulateur, l'équipe a créé un réseau sensible aux contours et aux couleurs.

Reconnaissance de chiffres manuscrits

Les travaux de BEYELER *et al.* [1] ont permis d'effectuer de la reconnaissance de chiffres manuscrits à l'aide d'un système neuromorphique de type STDP. L'apprentissage est ici effectué de façon supervisée.

2.3 Matériel et méthodes

2.3.1 Brian



Dans le but d'aider les scientifiques, GOODMAN et BRETTE [9] ont développé un nouveau simulateur de réseaux neuromorphiques en Python. Ce simulateur est un outil intuitif et flexible pour développer de nouveaux modèles. Il est d'ailleurs utilisé dans le cadre de cours de neurosciences computationnelles à l'École Normale Supérieure (ENS) de Paris.

Cette bibliothèque Python permet de créer ses propres réseaux neuromorphiques. Nous pouvons par exemple choisir le nombre de couches ainsi que le nombre de neurones par couche. L'utilisateur doit définir le modèle de fonctionnement de ses

neurones. Il peut d'ailleurs en créer des différents pour différents groupes de neurones. De plus, nous pouvons aussi créer des sous-groupes de neurones à partir des groupes principaux. L'utilisateur doit également définir le modèle que suivra les synapses.

La bibliothèque met aussi à disposition des outils de surveillance des réseaux pour donner une alternative à l'utilisation de Matlab par exemple.

2.3.2 Maximum de trois nombres

Dans le but de prendre en main les notions définies précédemment, nous avons commencé par développer un système neuromorphique simple. Le réseau devait être capable de reconnaître et de nous indiquer le maximum parmi trois nombres entiers. Ce simple exemple permet déjà d'envisager et de manipuler différentes structures de réseaux ainsi que différentes manières d'encoder les données pour les fournir au système. Plusieurs versions ont vu le jour au fur et mesure que nos connaissances du sujet se sont développées.

Version sans STDP

Notre première version possède deux couches de neurones. La première couche compte six neurones et la couche de sortie trois (voir figure 2.5). Les six neurones en entrée servent à alimenter le réseau avec les relations entre nos chiffres a , b et c : le premier neurone déclenche un influx si a est strictement supérieur à b , le second si a est strictement supérieur à c , le troisième si b est strictement supérieur à a , etc. Les neurones ont ici un comportement basé sur le modèle *intègre et tire*. Le potentiel de repos des neurones de la seconde couche est défini à 0 mV et leur seuil de déclenchement à 10 mV.

Des synapses sont créées entre les deux premiers neurones de la première couche et le premier neurone de la seconde couche qui représente «*a est le maximum*». Deux autres synapses sont créées entre les deux neurones suivants de la première couche et le second de la seconde couche qui représente «*b est le maximum*». Puis deux dernières synapses sont créées entre les deux derniers neurones de la première couche et le troisième et dernier de la seconde couche qui représente «*c est le maximum*». L'efficacité synaptique est définie à 5,5 mV et celle-ci est fixe (aucune plasticité synaptique). De cette façon, seul le bon neurone de sortie est activé pour un motif donné.

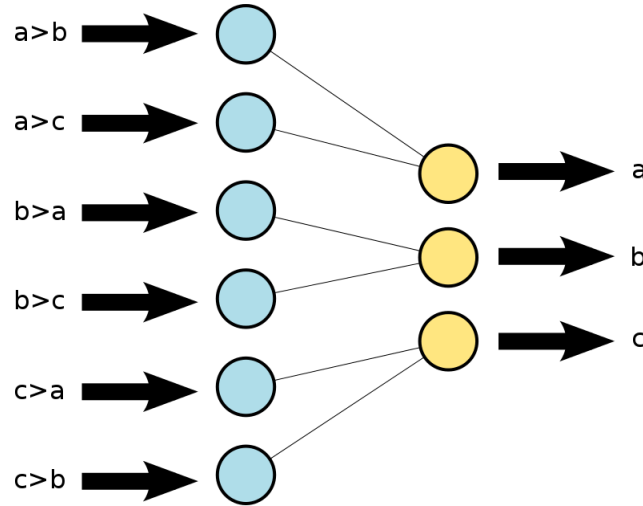


FIGURE 2.5 – Structure du réseau de la première version de reconnaissance du maximum de trois entiers. La couche d’entrée compte six neurones et la couche de sortie en compte trois. Les liens entre les neurones représentent les synapses créées.

Version avec des trains d’influx nerveux

La plupart des travaux que nous avons étudié utilisent des trains d’influx pour alimenter leurs réseaux [1, 17]. Nous avons donc mis en place dans un second temps une version du réseau pour alimenter les entrées avec des trains d’influx plutôt qu’avec un seul. Nous souhaitons également observer l’influence de synapses inhibitrices au sein du réseau ainsi que d’une inhibition latérale sur la couche de sortie. Ce réseau compte trois neurones pour la première couche et trois pour la seconde couche (voir figure 2.6).

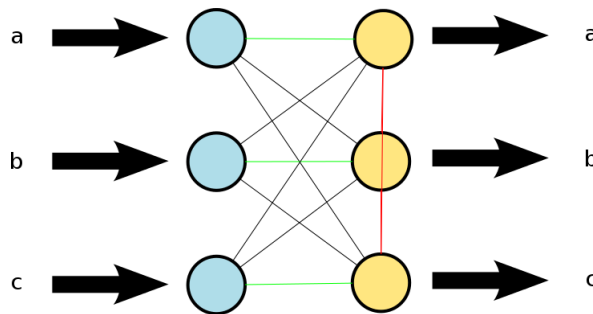


FIGURE 2.6 – Structure du réseau neuromorphique utilisé dans notre deuxième version de la reconnaissance du maximum de trois nombres. Les synapses représentées en noir sont des synapses inhibitrices alors que les vertes représentent des synapses excitatrices. Les synapses rouges constituent l’inhibition latérale sur la couche de sortie.

Les deux couches sont complètement reliées par des synapses. Chaque neurone de la première couche est connecté à tous les neurones de la seconde. Par défaut,

toutes les synapses sont définies comme inhibitrices en leur attribuant un poids synaptique négatif. Puis, nous redéfinissons comme excitatrices les synapses entre les deux premiers neurones de chaque couche, puis entre les deux seconds et entre les deux derniers. De plus, les synapses de la seconde couche s'inhibent latéralement.

Les neurones de la première couche déclenchent des influx selon une loi de Poisson avec une fréquence fonction de leur valeur :

$$100 \times \text{valeur Hz}$$

Le réseau est ainsi alimenté pendant cinq secondes pour un motif. Le maximum des trois nombres en entrée correspond alors au neurone de sortie ayant émis le plus d'influx nerveux pendant ce laps de temps.

Version avec STDP

Nous avons voulu ensuite prendre en main un réseau neuromorphique de type STDP. L'objectif est aussi de mettre en place un apprentissage supervisé. Pour cela, nous nous sommes basés sur la version sans STDP pour la structure du réseau. Ici, le réseau fonctionne en deux phases, l'une d'apprentissage et l'autre de test.

Nous avons adapté la version sans STDP en modifiant les synapses et en ajoutant une couche de neurones qui permet de superviser l'apprentissage (voir figure 2.7). Tous les neurones de la première couche sont reliés par des synapses avec STDP à chaque neurone de la seconde. Initialement, tous les poids synaptiques sont définis à 8 mV et le seuil de déclenchement d'un influx nerveux est fixé à 15 mV. De plus, une inhibition latérale est mise en place sur la couche de sortie.

Pour l'apprentissage, nous avons décidé de mettre en place un apprentissage supervisé par inhibition des neurones ne devant pas déclencher d'influx nerveux pour un motif donné. Pour cela, la couche permettant la supervision de l'apprentissage compte trois neurones, un pour chaque neurone de sortie du réseau. Pendant la phase d'apprentissage, si a est le maximum alors on déclenche des influx nerveux à partir des neurones d'apprentissage liés à « b est le maximum» et « c est le maximum» de façon à les empêcher de déclencher à leur tour un influx nerveux. De cette façon et dans ce cas, seul le neurone représentant « a est le maximum» déclenche un influx nerveux et voit le poids synaptique de ses synapses potentiellement augmenté.

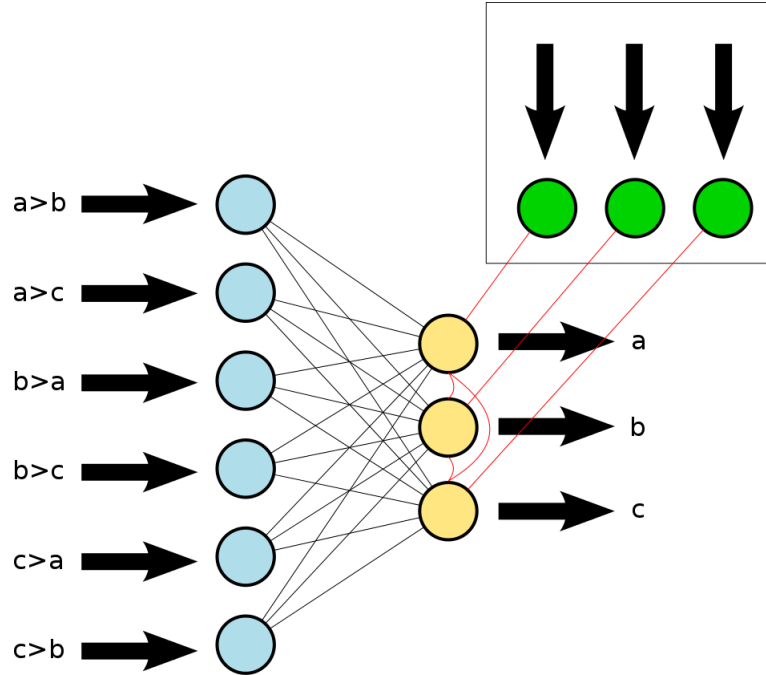


FIGURE 2.7 – Structure du réseau neuromorphique utilisé dans la version avec STDP de reconnaissance du maximum de trois nombres. Neurones bleus : couche d’entrée. Neurones jaunes : couche de sortie. Neurones verts : couche de supervision de l’apprentissage. Liens noirs : synapses de type STDP. Liens rouges : synapses inhibitrices dont le poids est fixé.

Pendant la phase d’apprentissage, nous présentons une suite de motifs au réseau. Cette suite est elle-même divisée en trois phases : apprendre «*a est le maximum*» puis apprendre «*b est le maximum*» et enfin apprendre «*c est le maximum*». Dans chaque cas, la relation entre les deux nombres non maximum est choisie aléatoirement. De cette manière, le réseau va adapter à chaque présentation d’un motif le poids synaptique de chaque synapse pour répondre au problème. Le poids est borné de façon à voir apparaître des synapses inhibitrices. Quand tous les motifs d’apprentissage ont été présentés, nous passons à la phase de test.

Durant la phase de test, nous désactivons la règle STDP des synapses de façon à figer leur poids synaptique. Puis nous présentons au réseau des motifs dans le même ordre que précédemment pour vérifier si il répond correctement ainsi configuré.

Version finale

L’objectif ici est d’optimiser la version précédente pour étudier le comportement du système avec des poids synaptiques initiaux aléatoires et indépendants pour chaque synapse. Nous avons également ajouté plusieurs options à cette version :

- mélange des cas d'apprentissage (pour trois cas de chaque maximum par exemple, permet d'utiliser un jeu d'apprentissage de la forme «b,c,a,c,a,a,b,c,b» plutôt que «a,a,a,b,b,b,c,c,c»),
 - mélange des cas de test (comme le mélange des cas d'apprentissage mais avec les cas de tests),
 - choix du mode d'apprentissage : supervision (excitation ou inhibition) ou non supervision,
 - sauvegarde des poids synaptiques à chaque étape de la phase d'apprentissage.
- La sauvegarde des poids synaptiques nous permet à l'aide d'un script de visualiser leur évolution pendant l'apprentissage sous forme de vidéos (voir figure 2.8).

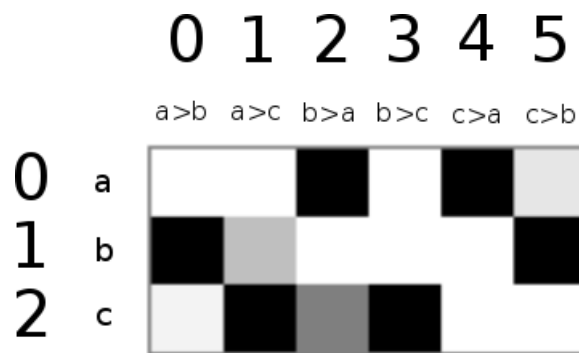


FIGURE 2.8 – Représentation des poids synaptiques des synapses reliant la couche d'entrée du réseau à la couche de sortie. Chaque case représente le poids synaptique d'une synapse. L'ordonnée représente les neurones de sortie et l'abscisse les neurones d'entrée. Le niveau de gris d'une case varie au cours du temps en fonction du poids synaptique. Plus une case est noire et plus le poids synaptique est faible. À l'inverse, plus une case est blanche, plus le poids synaptique est important.

Pour modéliser le comportement des neurones et des synapses, nous avons utilisé différents paramétrages dans notre version finale du réseau (voir tableau 2.1). Nous nous sommes ainsi rapproché d'un modèle plus proche de la réalité biologique.

TABLE 2.1 – Différents paramétrages du réseau final

V1	<p>Temps de présentation : 20ms</p> <p>Tension à la membrane des neurones : $0 \rightarrow 15V$</p> <p>Modèle des neurones : implémentation de Brian avec $\tau = \frac{2}{3} \times 5ms$ et $El = Vr = 0V$</p> <p>Modèle de STDP : basé sur le modèle d'HappyBrian :</p> <ul style="list-style-type: none"> - w le poids synaptique - $tPre$ le moment de déclenchement d'un influx par le neurone présynaptique - $tPost$ le moment de déclenchement d'un influx par le neurone postsynaptique - Lors du déclenchement d'un influx par le neurone présynaptique : <ul style="list-style-type: none"> - $tPre = t$ - $vm += w$ - Lors du déclenchement d'un influx par le neurone postsynaptique : <ul style="list-style-type: none"> - $tPost = t$ - $w = w + (2V + 1V) \times (tPre \leq tPost) \times (tPre + 10ms \geq tPost) - 1V$
V2	<p>Temps de présentation : 50ms</p> <p>Tension à la membrane des neurones : $-74 \rightarrow -54mV$</p> <p>Modèle des neurones : implémentation de Brian avec $\tau = \frac{2}{3} \times 5ms$ et $El = -74mV$</p> <p>Modèle de STDP : basé sur le modèle d'HappyBrian :</p> <ul style="list-style-type: none"> - w le poids synaptique - $tPre$ le moment de déclenchement d'un influx par le neurone présynaptique - $tPost$ le moment de déclenchement d'un influx par le neurone postsynaptique - Lors du déclenchement d'un influx par le neurone présynaptique : <ul style="list-style-type: none"> - $tPre = t$ - $vm += w$ - Lors du déclenchement d'un influx par le neurone postsynaptique : <ul style="list-style-type: none"> - $tPost = t$ - $w = w + (3mV + 1,5mV) \times (tPre \leq tPost) \times (tPre + 10ms \geq tPost) - 1,5mV$
V3	<p>Temps de présentation : 50ms</p> <p>Tension à la membrane des neurones : $-74 \rightarrow -54mV$</p> <p>Modèle des neurones : implémentation de Brian avec $\tau = \frac{2}{3} \times 5ms$ et $El = -74mV$</p> <p>Modèle de STDP : basé sur un exemple de Brian :</p> <ul style="list-style-type: none"> - w le poids synaptique - $\frac{dApr}{dt} = \frac{-Apr}{\tau_{pre}}$ - $\frac{dApost}{dt} = \frac{-Apost}{\tau_{post}}$ - Lors du déclenchement d'un influx par le neurone présynaptique : <ul style="list-style-type: none"> - $vm += w$ - $Apr += dApr$ - $w += Apost$ - Lors du déclenchement d'un influx par le neurone postsynaptique : <ul style="list-style-type: none"> - $Apost += dApost$ - $w += Apr$

2.3.3 Reconnaissance d'expressions faciales

Dès que notre première version de reconnaissance du maximum de trois nombres avec un système de type STDP donna des résultats, nous avons commencé à mettre en place un système neuromorphique de test pour effectuer de la reconnaissance d'expressions faciales (ici des visages souriants). Cet aspect de notre recherche a aussi connu plusieurs versions. Celles-ci ont vu le jour alors que la version finale du maximum de trois entiers n'était pas encore développée ce qui explique que certains comportements ici sont identiques à ceux employés dans la version avec STDP du maximum de trois entiers. Le but est de commencer à manipuler des réseaux neuromorphiques dans le but de faire de la reconnaissance faciale.

Première version

Nous travaillons ici à partir de photos de visages de 20×20 pixels en niveaux de gris. Ces images sont extraites de la base GENKI [19] à l'aide de la bibliothèque OpenCV. Pour structurer notre réseau, nous décidons d'utiliser un neurone par pixel de l'image en entrée, donc notre première couche compte quatre cents neurones. Pour les couches suivantes, nous nous basons en partie sur le réseau de neurones utilisé dans les travaux de l'équipe FOX [7]. Notre réseau comprend donc quatre couches de neurones. Les couches possèdent respectivement 400, 30, 15 et 2 neurones.

Les différentes couches sont complètement connectées les unes après les autres par des synapses. Ces synapses suivent une règle de type STDP. Leur poids est initialisé à une valeur fixée de 8 mV et identique pour toutes les synapses comme dans notre réseau de reconnaissance de trois nombres. L'apprentissage se fait de manière supervisée et par inhibition du neurone de sortie ne devant pas déclencher d'influx pour un visage donné.

Pour alimenter le réseau, nous appliquons un codage à seuil pour déterminer si un neurone donné de la couche d'entrée doit déclencher un influx ou non. Si la valeur du niveau de gris du pixel envisagé est supérieure à un seuil donné, le neurone associé déclenche un influx. On sélectionne ainsi les pixels les plus sombres pour être source d'un influx nerveux dans notre réseau. Nous fournissons au réseau une image toutes les 500 ms pendant la phase d'apprentissage et la phase de test. Les tests ont été effectués avec différents nombres d'images allant d'une centaine pour chaque phase jusqu'à un millier environ.

Nous mettons également en place ici une inhibition latérale sur chaque couche hormis la couche d'entrée. Ainsi chaque neurone qui déclenche un influx nerveux inhibe ses voisins sur la même couche. Nous ajoutons aussi un mécanisme de rétro-propagation de l'erreur. Il doit permettre d'ajuster les poids synaptiques lors d'une erreur en fonction de l'importance de celle-ci. Les poids synaptiques qui contribuent à engendrer une erreur importante se voient modifiés de manière plus significative que les poids qui ont engendré une erreur moins importante.

Seconde version

Dans un second temps, nous avons adapté la version précédente pour qu'elle ait moins d'informations inutiles à traiter. Pour cela, nous avons ajouté une phase de prétraitement des images pour en extraire d'abord les contours à l'aide d'un filtre de Canny. Ce filtre est utilisé en traitement d'image pour la détection des contours. Il a été conçu pour être optimal suivant trois critères :

- **bonne détection** : faible taux d'erreur dans la signalisation des contours,
- **bonne localisation** : minimisation des distances entre les contours détectés et les contours réels,
- **clarté de la réponse** : une seule réponse par contour et pas de faux positif.

Le gain ainsi obtenu nous permet de travailler avec des images de cinquante pixels de côté ce qui nous donne plus de précision pour les contours.

Ainsi les neurones de la couche d'entrée déclenchant des influx correspondent aux pixels contenant un contour. De cette façon, nous limitons le nombre de neurones activés de la couche d'entrée tout en conservant les informations importantes pour le réseau. Mais cette méthode nous oblige à revoir la structure de notre réseau. Notre couche d'entrée a maintenant besoin de deux mille cinq cents neurones pour conserver la propriété d'un neurone pour un pixel.

Afin d'observer le comportement du réseau dans une configuration différente d'aujourd'hui, nous supprimons une couche intermédiaire. De plus la seule couche intermédiaire restante se voit agrandie pour atteindre quatre cents neurones. La couche de sortie est, elle, limitée à un neurone pour fonctionner comme un booléen : souriant ou non-souriant.

Version avec balayage

Nous avons étudié par la suite la possibilité de réduire le nombre de neurones de la couche d'entrée. Pour cela, plutôt que de considérer une photo dans sa globalité directement, nous alimentons le réseau avec chaque colonne composant l'image,

comme si le regard «balayait» l'image. Une image est présentée pendant un laps de temps de 500 ms. Pendant la présentation d'une image, nous fournissons au réseau chaque colonne de la gauche vers la droite de l'image, les unes après les autres. Nous comptons sur la plasticité de type STDP pour extraire et détecter le motif «visage souriant» parmi le train d'influx qui représente une photo de visage entière.

Le nombre de neurones de la couche d'entrée est limité au nombre de pixels d'une colonne de l'image pré-traitée (ici quarante pixels). Nous nous rapprochons de la première version pour le reste de la structure du réseau en choisissant deux couches intermédiaires de, respectivement, quarante et vingt neurones. La couche de sortie compte, elle, deux neurones.

Une image est présentée pendant un laps de temps de 500 ms. Donc une colonne est présentée toutes les $500/40 = 12,5$ ms. Nous fournissons ici d'une centaine à un millier d'images souriantes au réseau pour la phase d'apprentissage et le même nombre d'images différentes pour la phase de test. Les images sont, comme dans la version précédente, pré-traitées pour ne fournir au réseau que les contours des visages.

Pour pouvoir mener des tests à plus grande échelle, nous avons adapté le programme pour s'interfacer avec un script. Cela nous a permis de tester de nombreuses valeurs pour les différents paramètres du réseau comme le temps de présentation des données, le poids des synapses inhibitrices, la valeur du seuil de décharge des neurones ou le temps réfractaire des neurones par exemple.

2.4 Résultats

2.4.1 Maximum de trois nombres

Version sans STDP

La première version de reconnaissance du maximum de trois nombres fonctionne correctement mais elle se résume à un ET logique.

Version avec des trains d'influx nerveux

La seconde version est, elle, laissée de côté. En effet, pour déterminer quel entier est le maximum, il faut déterminer le maximum entre les trois nombres d'influx nerveux déclenchés par la couche de sortie ce qui nous ramène donc au même problème.

On voit cependant grâce aux traces (voir figure 2.9) du programme de test que les synapses inhibitrices ont une influence sur le comportement final du réseau. Dans notre exemple, le rapport entre le nombre d'influx nerveux des neurones 3 et 1 de la première couche vaut environ 3 alors qu'en sortie ce rapport vaut presque 9. Les neurones 1 et 2 de la couche de sortie ont donc bien été inhibé.

```
# spikes from 1st G1: 109
# spikes from 2nd G1: 9
# spikes from 3rd G1: 270
```

```
# spikes from 1st G2: 10
# spikes from 2nd G2: 0
# spikes from 3rd G2: 87
```

```
a = 0.216497122595
b = 0.0203027984244
c = 0.505403707947
```

FIGURE 2.9 – Exemple de sortie produite par la seconde version du programme de reconnaissance de trois nombres. Le programme affiche le nombre d'influx nerveux déclenchés par chaque neurone de chaque couche (G1 couche d'entrée et G2 couche de sortie). Nous voyons également les valeurs prises aléatoirement au début du test par a , b et c .

Version avec STDP

La troisième version donne quant à elle de meilleurs résultats. Nous constatons sur la figure 2.10 qu'avec trop peu de cas d'apprentissage (ici cinq pour chaque maximum), l'amplitude des potentiels des membranes indique bien le maximum. Cependant, les potentiels de tous les neurones dépassent le seuil de déclenchement (10 mV). Et donc tous les neurones de sortie déclenchent un influx nerveux pour un motif donné.

Par contre, lorsque le nombre de cas d'apprentissage augmente, nous voyons sur la figure 2.11 que le potentiel des neurones de sortie ne devant pas déclencher d'influx nerveux reste bien en dessous du seuil de déclenchement tout en conservant le déclenchement d'un influx par le bon neurone.

Nous avons également testé avec cette version si le système était capable de reconnaître c comme maximum en ayant seulement appris à reconnaître a et b . Mais cette hypothèse n'a pas pu être vérifiée, le système n'arrive pas à déduire « c est le maximum» en ayant appris à reconnaître seulement a et b .

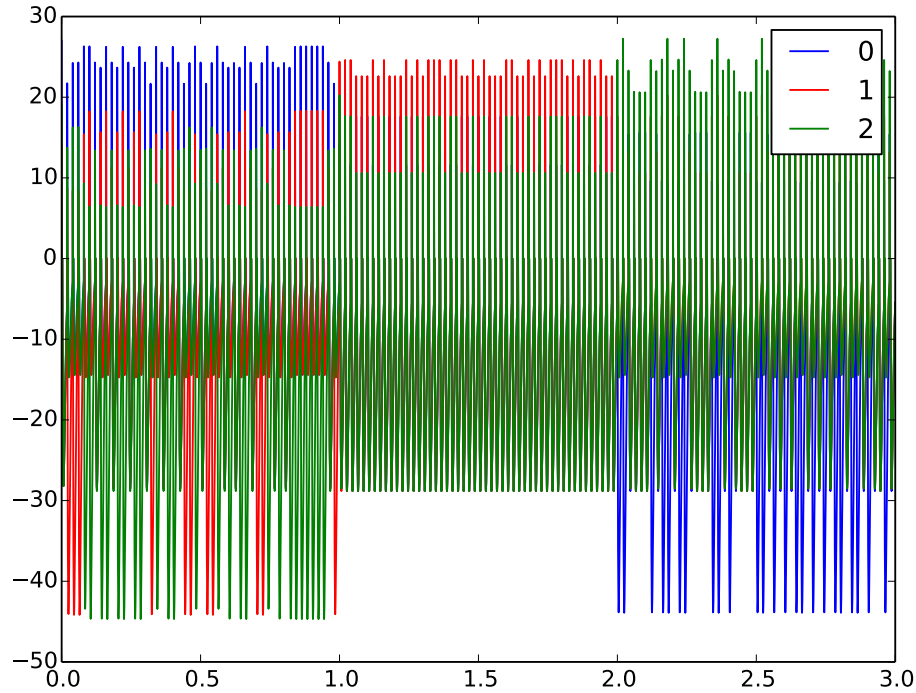


FIGURE 2.10 – Potentiel pendant les tests des membranes des neurones de la couche de sortie avec un apprentissage constitué de cinq cas pour chaque maximum. Le potentiel (V) est en ordonnée et le temps (s) en abscisse. Le neurone 0 (bleu) représente «*a est le maximum*», le neurone 1 (rouge) représente «*b est le maximum*» et le neurone 2 (vert) représente «*c est le maximum*».

Version finale

Avec cette version, nous constatons que le système réagit de la même façon avec un supervision par excitation et par inhibition. Par contre, les quelques tests effectués avec un apprentissage non-supervisé n'aboutissent pas.

Par contre, avec une initialisation aléatoire des poids synaptiques comprise entre -10 V et 10 V (pour la V1), le système n'arrive plus à apprendre malgré un grand nombre de cas d'apprentissage. Pour retrouver des résultats corrects avec une initialisation aléatoire, nous devons rester dans un intervalle positif de poids synaptique. Mais même dans ce cas, il reste des cas où le réseau n'arrive pas à apprendre correctement à reconnaître le maximum de trois nombres (20 à 25% des cas).

De plus, en générant des vidéos de l'évolution des poids synaptiques pendant l'apprentissage, nous constatons que lorsque cet apprentissage se déroule correctement, le schéma de répartition des poids synaptiques est très semblable d'un test à l'autre

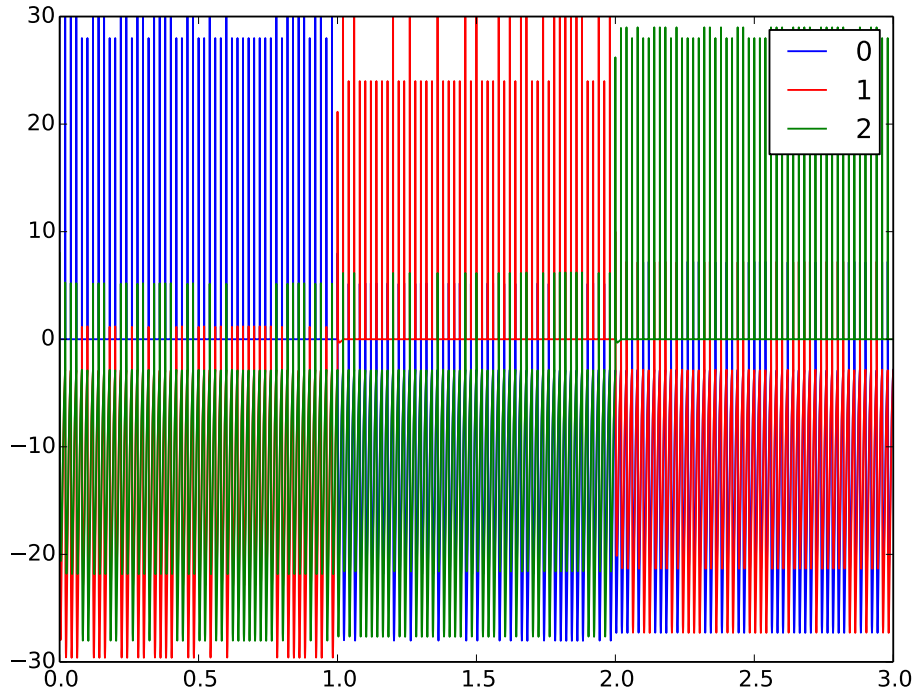


FIGURE 2.11 – Potentiel pendant les tests des membranes des neurones de la couche de sortie avec un apprentissage constitué de cinquante cas pour chaque maximum. Le potentiel (V) est en ordonnée et le temps (s) en abscisse. Le neurone 0 (bleu) représente «*a est le maximum*», le neurone 1 (rouge) représente «*b est le maximum*» et le neurone 2 (vert) représente «*c est le maximum*».

(voir figure 2.12). En effet, dans ces cas là, ce sont toujours les même synapses qui ont un poids important.

Pour améliorer le pourcentage de réussite de l'apprentissage, nous décidons d'alimenter le réseau lors de l'apprentissage avec des cas structurés de façon à avoir une meilleure répartition des cas de maximum. Ici, nous voulons entraîner le réseau avec des triplets a , b et c . L'ordre des cas au sein des triplets est laissé aléatoire. Mais les résultats ne sont pas probants (voir tableau 2.2). En effet, avec les cas complètement mélangés, nous avons un taux d'apprentissage correct de l'ordre de 75 à 80% dès qu'on dépasse les deux cents cas d'apprentissage. Par contre, avec nos cas d'apprentissage structurés, nous ne dépassons pas 37% de taux de réussite même avec deux mille cas d'apprentissage.

En utilisant des paramètres plus proches de la réalité biologique, nous gardons le même comportement. Cependant, en augmentant le temps qui sépare deux pré-



FIGURE 2.12 – Représentation des poids synaptiques des synapses reliant la couche d’entrée du réseau à la couche de sortie. Chaque case représente le poids synaptique d’une synapse. L’ordonnée représente les neurones de sortie et l’abscisse les neurones d’entrée. Le niveau de gris d’une case varie au cours du temps en fonction du poids synaptique. Plus une case est noire et plus le poids synaptique est faible. À l’inverse, plus une case est blanche et plus le poids synaptique est important.

Structure des cas d’apprentissage	Taux de réussite
Par blocs (<i>aaa bbb ccc</i>)	50%
Aléatoire (<i>bacbccaba</i>)	75 à 80%
Par triplets (<i>bac abc acb</i>)	25 à 35%

TABLE 2.2 – Tableau des pourcentages d’apprentissage réussi pour chaque méthode d’alimentation du réseau (exemple pour trois cas de chaque maximum entre parenthèses) pendant la phase d’apprentissage avec entre 300 et 2000 cas.

L’apprentissage est considéré comme réussi lorsque le réseau ne commet aucune erreur pendant la phase de test.

sentations, nous obtenons des résultats différents. L’initialisation aléatoire des poids synaptiques pose quelques problèmes (voir figure 2.13). En effet, lorsque le réseau est alimenté avec des triplets *abc* (triés ou non), le réseau n’est jamais capable de reconnaissance. Avec une initialisation non aléatoire des poids synaptiques (ici à 0V), nous obtenons un taux de reconnaissance de 100% (ou très proche) dans de nombreux cas (voir figure 2.14).

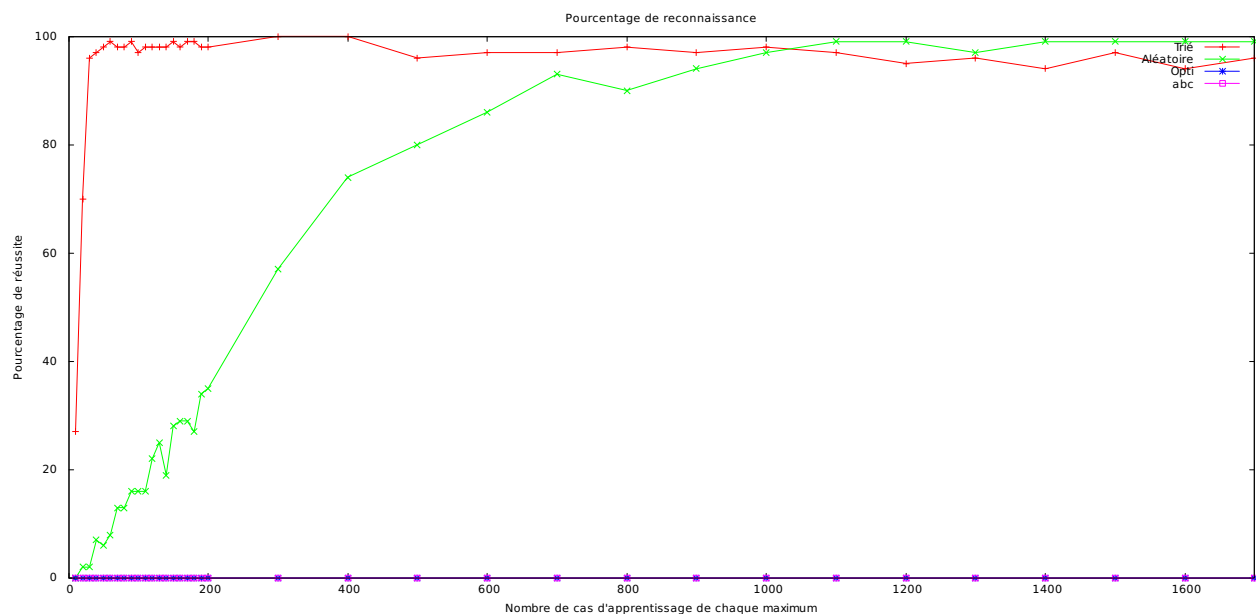


FIGURE 2.13 – Taux de reconnaissance avec une initialisation aléatoire des poids synaptiques en fonction du mode d'alimentation du réseau.

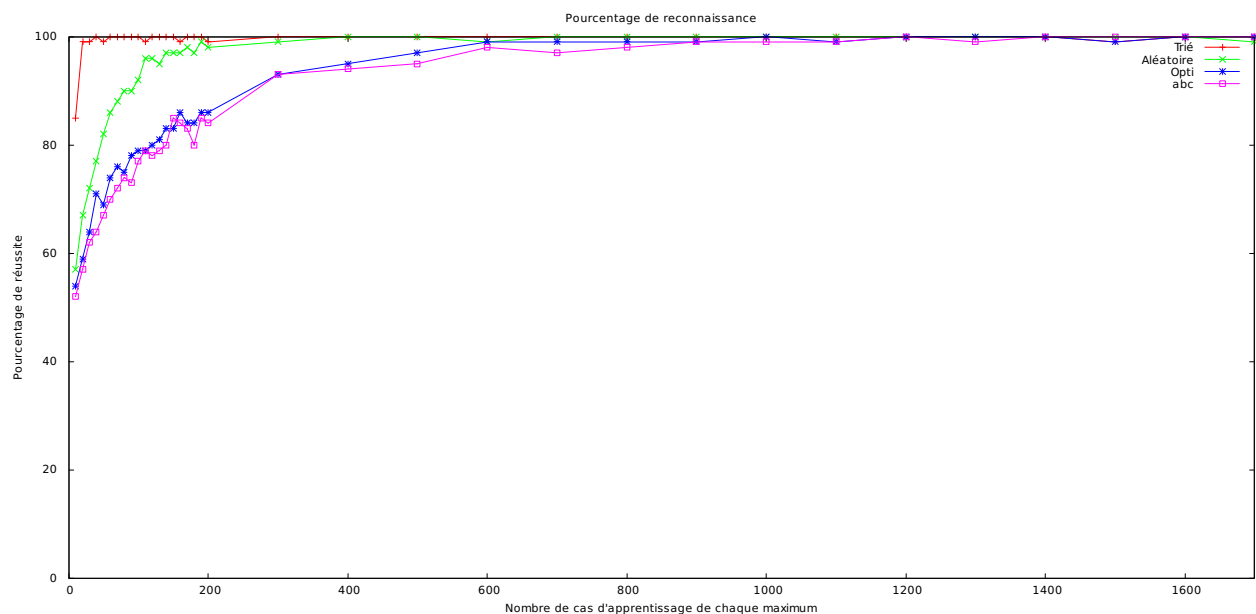


FIGURE 2.14 – Taux de reconnaissance avec une initialisation à 0 des poids synaptiques en fonction du mode d'alimentation du réseau.

2.4.2 Reconnaissance d'expressions faciales

Quel que soit la structure du réseau déployée ou la méthode d'alimentation du réseau, nous n'avons obtenu aucun résultat probant. En effet, aucune différenciation entre les visages souriants et non-souriants n'a été observée pendant les phases de test. De plus, nous avons rencontré des difficultés à observer l'état interne du réseau au vu de sa taille.

2.5 Conclusion

Nos travaux sur le maximum de trois nombres sont à analyser plus en détails. En effet, lorsque l'apprentissage ne se passe pas correctement, on peut distinguer principalement deux cas de figures. Le réseau peut «confondre» deux nombres, par exemple il désigne a comme maximum à la place de b et inversement. Dans le reste des cas, le réseau semble perturbé par la relation des deux nombres non maximum.

Pour la reconnaissance d'expressions faciales, au vu des expériences déjà menées par d'autres équipes et de nos résultats, on peut supposer que la taille des réseaux que nous avons mis en place est largement insuffisante pour mémoriser et généraliser une information aussi complexe que *«ce visage est souriant»*.

Si nous n'avons pas réussi à résoudre le problème «Un visage est-il souriant ou non?», notre étude nous permet d'émettre quelques remarques et hypothèses. L'étude du cas du maximum de trois nombres nous permet de bien voir évoluer un réseau neuromorphique de type STDP. Nous pouvons constater grâce à ce cas de test qu'un réseau de type STDP est en effet capable de mémoriser et de classifier des motifs. De plus, au vu des travaux déjà effectués dans la vision par ordinateur comme la reconnaissance de chiffres manuscrits, nous pouvons voir une piste très intéressante dans l'utilisation de réseaux neuromorphiques de type STDP pour classifier des expressions faciales. On pourrait, grâce à ce type de réseau, classifier des visages selon différentes émotions directement depuis des flux d'images.

2.6 Perspectives

Plusieurs pistes sont encore à explorer.

L'un des premiers axes à améliorer est le modèle synaptique utilisé. Dans nos réseaux, les meilleurs résultats obtenus le sont avec un modèle synaptique adapté et personnalisé où les valeurs d'incrémentations et de décrémentation des poids synaptiques sont fixées. Dans un réseau neuromorphique de type STDP, ces valeurs doivent dépendre de l'écart de temps entre deux influx nerveux.

Une autre piste intéressante à explorer est l'utilisation de trains d'influx nerveux pour coder une information. Actuellement, pour un motif donné à classer, un neurone en entrée du réseau ne déclenchera qu'un seul influx si il doit être activé. Un train d'influx nous permettrait de coder des informations plus complexes en entrée du réseau sachant que la plasticité de type STDP permet justement de reconnaître des motifs dans de tels trains.

Nous pourrions enfin essayer de résoudre des problèmes plus complexes comme des problèmes à une dimension. Par exemple, nous pourrions classer des lignes de pixels noirs ou blancs selon la répartition des pixels.

Avec plus de temps, nous pourrions créer des réseaux bien plus grands afin de résoudre des problèmes bien plus complexes comme la reconnaissance d'émotions. Pour ce faire, nous pourrions par exemple utiliser des couches d'entrée beaucoup plus grandes et divisées en sous-groupes spécialisés.

À plus grande échelle, il pourrait être utile de développer une interface graphique au simulateur Brian. Ainsi nous serions capable de visualiser les réseaux plus facilement et rapidement.

Chapitre 3

Conclusion

L'objectif du stage était d'explorer la possibilité d'utiliser des réseaux neuromorphiques de type STDP dans la vision par ordinateur et plus particulièrement dans la reconnaissance d'expressions faciales. Même si une solution concrète n'a pas été trouvée, il a été montré que ce type de réseau peut s'avérer très efficace pour résoudre ce type de problème. Notamment, la recherche de travaux existants a soulevé des études intéressantes sur le sujet dans la vision par ordinateur. De plus, les expériences menées confirment que les réseaux neuromorphiques de type STDP sont capables de mémoriser et d'extraire des motifs parmi des influx nerveux. Ceci permet dans notre cas de classer nos nombres selon le critère «lequel est le maximum».

Personnellement, ce stage a été très instructif. D'un point de vue technique, je me suis adapté aux outils existants notamment en apprenant un nouveau langage, le Python, pour utiliser le simulateur de réseaux de neurones. J'ai également perfectionné mon aptitude à développer des scripts pour automatiser mes tests. D'un point de vue plus général, j'ai pu acquérir des notions de neurosciences et améliorer ma compétence à rechercher des informations. Enfin, j'ai pu développer ma capacité à prendre du recul sur mon travail.

Bibliographie

- [1] Michael BEYELER, Nikil D. DUTT et Jeffrey L. KRICHMAR : Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule. *Neural Networks*, 48:109–124, décembre 2013.
- [2] Olivier BICHLER, Damien QUERLIOZ, Simon J. THORPE, Jean-Philippe BOURGOIN et Christian GAMRAT : Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity. *Neural Networks*, 32:339–348, août 2012.
- [3] Joseph M. BRADER, Walter SENN et Stefano FUSI : Learning real-world stimuli in a neural network with spike-driven synaptic dynamics. *Neural Computation*, 19(11):2881–2912, novembre 2007.
- [4] S. R. CAJÀL : *Textura del sistema nervioso del hombre y de los vertebrados*. Nicolás Moya, Madrid, 1904.
- [5] Bernard CALVINO : *Qu’est-ce qu’un neurone ?* Le Pommier, Paris, 2006.
- [6] Jacques COGET : Physiologie du neurone. http://passeport.univ-lille1.fr/site/biologie/scbio/Neurone/Neurone_web.publi/web/co/Neurone_web.html.
- [7] Taner DANISMAN, Ioan Marius BILASCO, Jean MARTINET et Chabane DJERABA : Intelligent pixels of interest selection with application to facial expression recognition using multilayer perceptron. *Signal Processing*, 93(6):1547–1556, juin 2013.
- [8] FOX : Site web de l’équipe fox. <http://www.lifl.fr/FOX/>.
- [9] Dan GOODMAN : Brian : a simulator for spiking neural networks in python. *Frontiers in Neuroinformatics*, 2, 2008.
- [10] A. L. HODGKIN et A. F. HUXLEY : A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol. (Lond.)*, 117(4):500–544, août 1952.
- [11] E.M. IZHIKEVICH : Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572, novembre 2003.

- [12] Timothee MASQUELIER et Simon J. THORPE : Learning to recognize objects using waves of spikes and spike timing-dependent plasticity. *In Neural Networks (IJCNN)*, pages 1–8. IEEE, juillet 2010.
- [13] Timothée MASQUELIER, Rudy GUYONNEAU et Simon J. THORPE : Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. *PLoS ONE*, 3(1):e1377, janvier 2008.
- [14] Timothée MASQUELIER, Rudy GUYONNEAU et Simon J. THORPE : Competitive STDP-Based spike pattern learning. *Neural Computation*, 21(5):1259–1276, mai 2009.
- [15] Carver MEAD : *Analog VLSI and neural systems*. Addison-Wesley, Reading, Mass., 1989.
- [16] R G MORRIS : D.O. Hebb : The organization of behavior, Wiley : New york ; 1949. *Brain Research Bulletin*, 50(5-6):437, décembre 1999.
- [17] Micah RICHERT, Jayram Moorkanikara NAGESWARAN, Nikil DUTT et Jeffrey L. KRICHMAR : An efficient simulation environment for modeling large-scale cortical processing. *Frontiers in Neuroinformatics*, 5, 2011.
- [18] F. ROSENBLATT : The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
- [19] [HTTP://MPLAB.UCSD.EDU](http://mplab.ucsd.edu) : The MPLab GENKI Database.