Simulation neuronale de la vision précoce corticale avec un modèle de Heeger

Sylvie Crahay, Thierry Viéville

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Simulation neuronale de la vision précoce corticale avec un modèle de Heeger.

S. Crahay & T. Viéville

N° 4534

Septembre 2002

THÈME 3

ISSN 0249-6399 ISRN INRIA/RR--4534--FR

INRIA
SOPHIA ANTIPOLIS

Simulation neuronale de la vision précoce corticale avec un modèle de Heeger.

S. Crahay & T. Viéville*

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Odyssee

Rapport de recherche n° 4534 — Septembre 2002 — 66 pages

Résumé : Une simulation neuronale des cartes corticales des aires V1 impliquées dans les mécanismes de vision précoce (tel que la détection de contour) est reportée ici. Le modèle biologiquement plausible sous-jacent est un modèle de Heeger [2, 8]. Il a été récemment proposé pour rendre compte de manière plus générale du fonctionnement de ces opérateurs spatio-temporels, que ce que les travaux historiques de Hubel et Wiesel proposaient [4, 3].

Notre développement logiciel est basé sur un logiciel libre qui propose une boite à outil pour la simulation de réseaux de neurones permettant de concevoir à la fois des architectures et des modèles neuronaux variés. Notre ajout à été d'implémenter les neurones "à-la" Heeger et d'expérimenter une architecture représentant les voies visuelles pré-corticales et l'aire corticale V1, sous la restriction d'une vision monoculaire et monochromatique.

L'interface logiciel réalisée permet aussi de générer des stimulus correspondant à ce qui est usuellement utilisé en neurophysiologie pour de futures comparaisons entre cette simulation et des données expérimentales effectives.

Mots-clés : Simulation neuronale, Vision précoce, Modèle de Heeger, Détection de contour.

* http://www.inria.fr/Thierry.Vieville

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Neuronal simulation of the cortical early vision using a Heeger model.

Abstract: A neuronal simulation of the cortical V1 maps involved in the early vision mechanisms (such as edge detection) is reported here. The biologically plausible underlying model is the Heeger model [2, 8]. It has been recently proposed as a more general representation of these spatio-temporal models, than the original propositions of the Hubel and Wiesel pioneer work [4, 3].

The present software development is based on a freeware toolbox used for neuronal networks simulation allowing the design of both versatile neuronal architectures and units. Our add-on has been to implement these neurons "à-la" Heeger and to experiment an architecture representing pre-cortical visual pathways and the V1 cortical area under the restrictive hypotheses of monocular and monochromatic vision.

The software interface also includes a visual stimulus generator corresponding to what is usually used in neuro-physiological experiments with the goal of performing future comparisons with effective experimental data.

Key-words: Neuronal simulation, Early vision, Heeger model, Edge detection.

Table des matières

1 Introduction

Le but de ce rapport est d´expliquer toutes les notions nécessaires à la compréhension du présent travail sur la simulation de l´activité corticale des aires V1 à l´aide d´un réseau de neurones.

Les travaux de Hubel et Wiesel [4, 3] ont permis, entre autres, d´établir le fonctionnement des cellules simples de l´aire visuelle primaire V1. Nous exposerons donc dans un premier temps leurs travaux. Nous continuerons avec les colonnes corticales dont parle F. Alexandre [1] qui regroupent des neurones ayant des fonctionnalités voisines.

Ainsi, une fois le fonctionnement biologique compris nous nous pencherons sur différents réseaux de neurones [6, 7] et mode d´apprentissage afin de déterminer celui que nous utiliserons.

Nous continuerons avec les modèles de Heeger [5] permettant de simuler l´activité des cellules simples des aires V1. Nous verrons d´abord le modèle linéaire avec ses qualités et ses défauts. Puis nous verrons une amélioration du modèle précédent : le modèle normalisé. Nous évaluerons ses points forts et ses limites.

Pour finir, nous décrirons les différents logiciels de simulation de réseaux de neurones que nous avons pu trouver, ainsi que les choix techniques effectués.

2 La vision

2.1 Le système nerveux

Chaque cellule nerveuse est constituée:

- d'un noyau, assurant les fonction métaboliques,
- d'un axone, fibre nerveuse cylindrique, qui guide les signaux émis par la cellule vers d'autres cellules,
- dendrites, fibres ramifiées recevant les signaux entrants d'autres cellules, dites afférentes,
- de synapses, les zones de contact entre axone et dendrite.

Chaque cellule intègre, c'est à dire fait la somme, des informations afférentes qui lui parviennent, et communique ensuite cette information aux autres cellules.

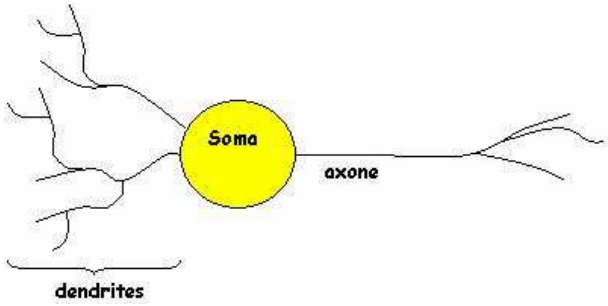

FIG. 1 – *Schéma d'un neurone.*

Le «codage» se fait sous forme d'impulsions nerveuses : la fréquence de décharge varie d'une impulsion toutes les deux à trois secondes au repos, jusqu'à plusieurs centaines d'impulsions par secondes quand la cellule est excitée.

Le mécanisme sous-jacent est le suivant : des potentiels d'action sont produits à la jonction corps cellulaire-axone, propagés jusqu'aux terminaisons axonales, provoquant la libération d'une molécule (neuromédiateur) dans la fente synaptique, qui va agir sur le neurone postsynaptique en augmentant ou en diminuant sa probabilité de décharge (selon que la synapse est excitatrice ou inhibitrice).

Ce mécanisme respecte la loi du tout ou rien : soit la somme des informations afférentes dépasse un certain seuil, et un potentiel d'action est émis, soit elle ne le dépasse pas, et rien ne se passe.

2.2 Structure du système visuel

2.2.1 Organisation globale

Le schéma illustre les différentes étapes du traitement visuel effectué par le cerveau :

– L'image est focalisée par le cristallin sur la rétine.

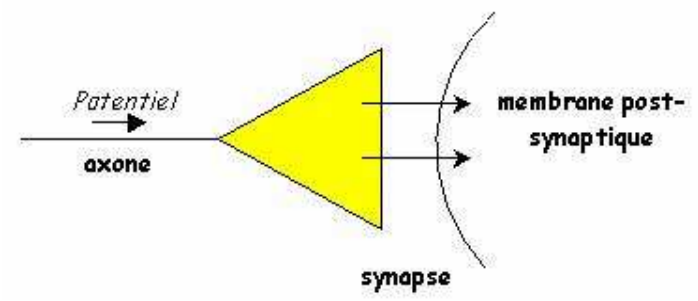

FIG. 2 – *Schéma d'une synapse.*

- Les photorécepteurs convertissent ensuite le stimulus visuel en impulsions électriques, et les transmettent, via le tractus optique aux corps genouillé latéraux.
- Ceux-ci se projettent alors vers le cortex visuel primaire, grâce aux radiations optiques.
- Les informations sont alors diffusées vers les aires corticales supérieures.

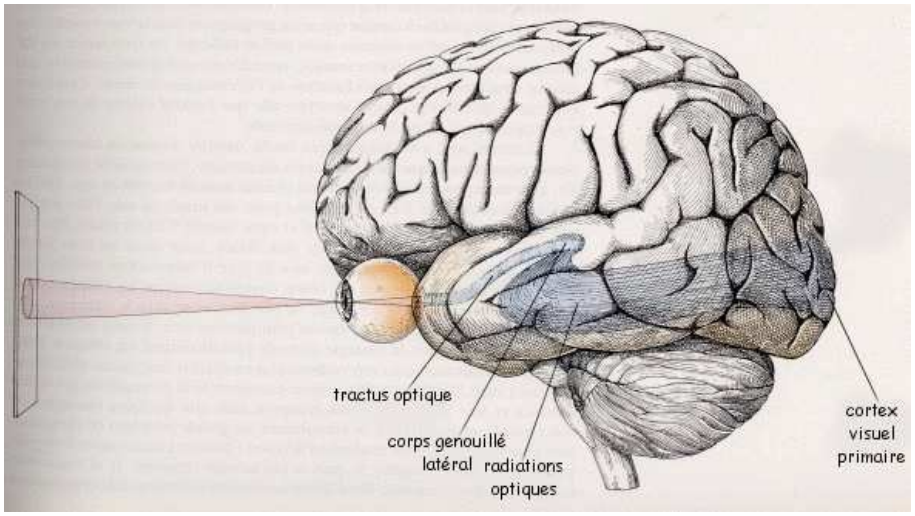

FIG. 3 – *Le système visuel (reproduit à partir de [3]).*

Il est intéressant de remarquer que ce schéma représente en fait une méthode expérimentale permettant d'étudier le système visuel. L'animal, en général un macaque, regarde un écran sur lequel est projeté une image. On enregistre alors les réponses des cellules à cette simulation visuelle en introduisant une micro-électrode dans une structure du système visuel, comme ici, par exemple, dans le cortex visuel primaire.

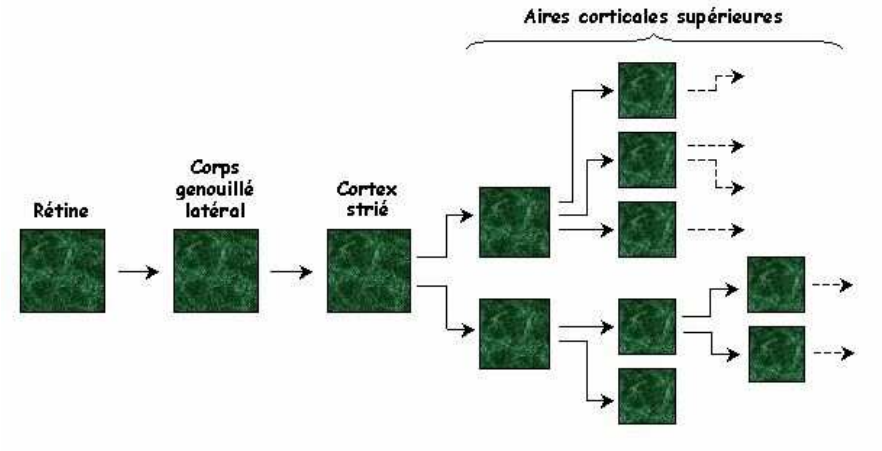

FIG. 4 – *Organisation simplifiée des structures du système visuel.*

2.2.2 La rétine

La rétine est constituée de trois couches de cellules, les voici dans l'ordre du traitement de l'information visuelle :

- La première est constituée de photo-récepteurs, qui sont au nombre de 125 millions par œil et se divisent en cônes (responsables de la perception des détails et des couleurs), et en bâtonnets (plus nombreux et permettant la vision crépusculaire).
- La seconde comprend les cellules bipolaires, qui relient les photo-récepteurs aux cellules ganglionnaires, les cellules horizontales, qui relient les photo-récepteurs aux cellules bipolaires et les cellules amacrines qui relient les cellules bipolaires aux cellules ganglionnaires.

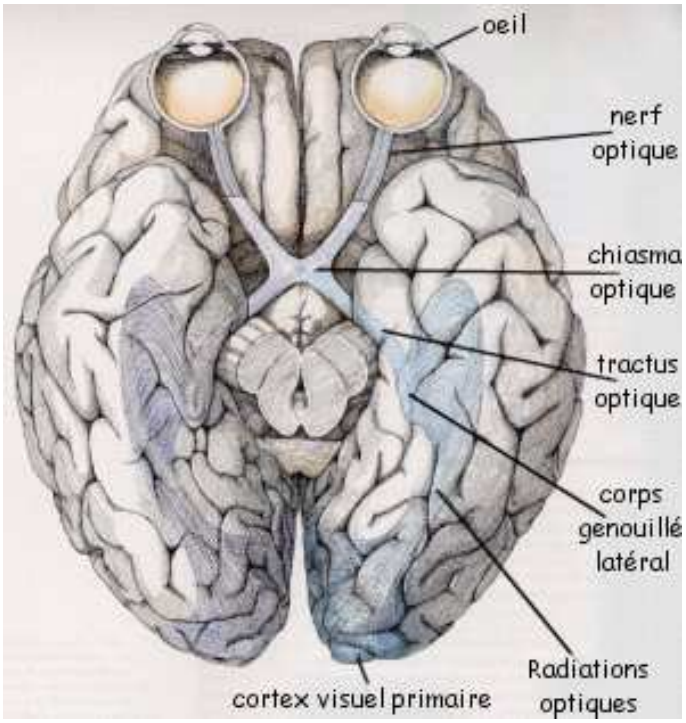

FIG. 5 – *La voie visuelle de l'œil au cortex visuel primaire (reproduit à partir de [3]).*

– La troisième est constituée des cellules ganglionnaires, qui sont relativement peu nombreuses (environ un million par œil).

Entre les 125 millions de photo-récepteurs et le million de cellules ganglionnaires, il y a donc une "compression" de l'information visuelle. Comment est elle effectuée ? On distingue en fait deux voies dans la rétine :

– La voie directe, qui consiste en un photo-récepteur, relié à une cellule bipolaire, reliée à une cellule ganglionnaire. Cette voie est compacte, en ce sens qu'elle associe une cellule ganglionnaire à un photo-récepteur. Elle correspond à une vision précise et se rencontre surtout près de la fovéa, qui est la région centrale de la rétine, et n'est constituée que de cônes.

– La voie indirecte, qui consiste en plusieurs photo-récepteurs (typiquement $1mm^2$), utilisant les cellules horizontales et amacrines en plus des cellules bipolaires et ganglionnaires. Cette voie est diffuse, en ce sens qu'elle associe une cellules ganglionnaire à de nombreux photo-récepteurs. Elle correspond à une vision plus floue, la vision périphérique.

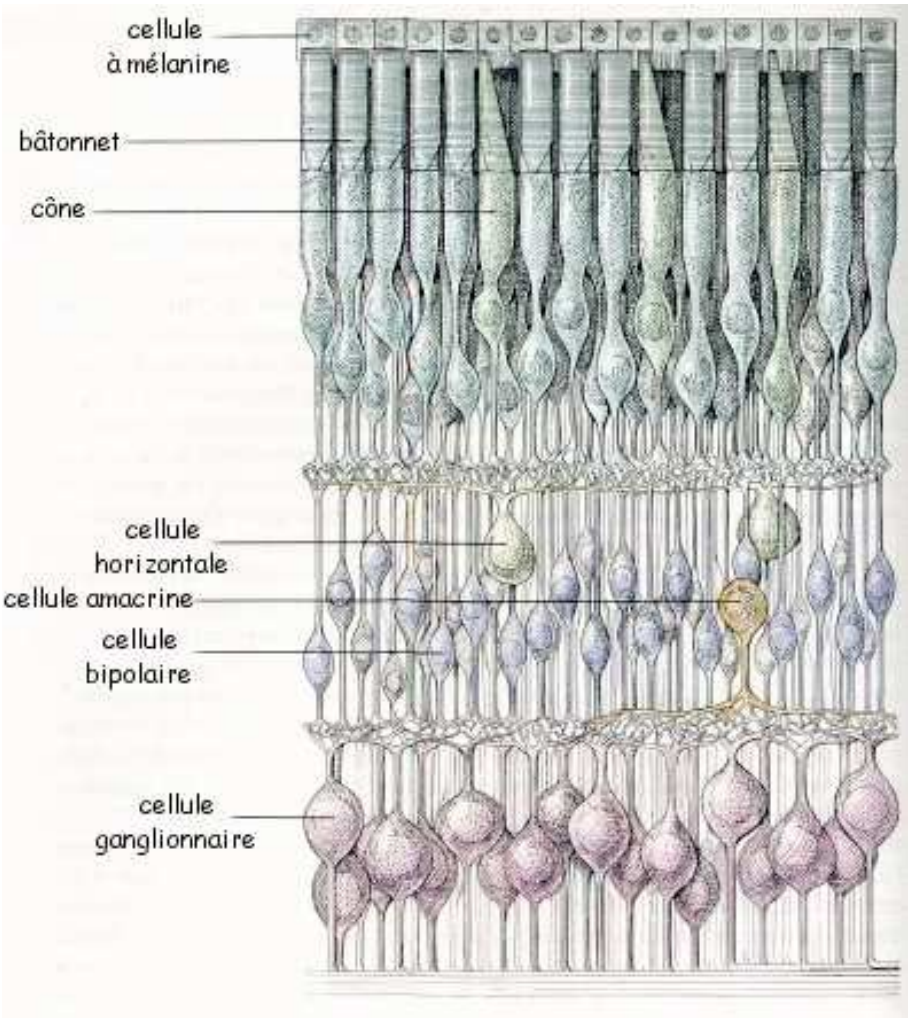

FIG. 6 – *Les cellules de la rétine (reproduit à partir de [3]).*

2.2.3 Les corps genouillés latéraux (L.G.N.)

Description Les corps genouillés latéraux sont constitués d'environ 1,5 million de cellules chacun. Outre le relais synaptique provenant de la rétine et se projetant vers le cortex cérébral, d'autres connexions existent dans les corps genouillés latéraux :

- celles participant à une rétroaction du cortex cérébral,
- celles issues de la formation réticulée du tronc cérébral (responsable de l'attention et de l'éveil),
- celles reliant latéralement des cellules des corps genouillés latéraux.

Nous pouvons remarquer sur la figure 5 que l'image des deux demi-rétines droites qui provient de la moitié gauche du champ visuel (puisque les images sont inversées sur la rétine) aboutit dans l'hémisphère cérébral droit et réciproquement.

Les cellules du corps genouillé latéral se distinguent particulièrement par leur champ récepteur. Il existe deux catégories de cellules : les cellules centre-ON et les cellules centre-OFF.

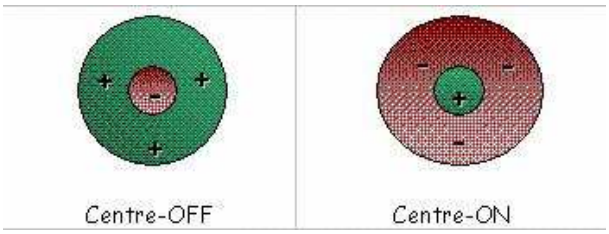

FIG. 7 – *Champs récepteurs de cellules des corps genouillés latéraux.*

Mise en évidence

Cellules à centre-ON : Ces cellules sont complètement excitées seulement si le stimulus lumineux correspond uniquement à la zone centrale du champ récepteur. Si le stimulus excite tout le champ récepteur alors il y a une excitation moyenne. Et finalement l'excitation du pourtour du champ ne donne aucune excitation.

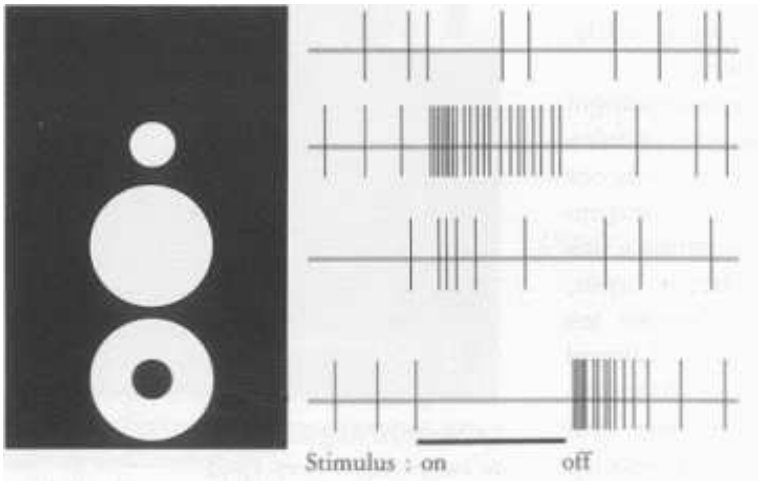

FIG. 8 – *Mise en évidence du champs récepteur d'une cellule à centre-ON* (à gauche se trouve le stimulus lumineux et à droite l'activité de la cellule soumise à ce stimulus) *(reproduit à partir de [3]).*

Cellules à centre-OFF : C'est exactement le contraire des cellules à centre-ON.

2.3 L'aire visuelle primaire

2.3.1 Les réponses des cellules corticales

Celles-ci ont principalement été étudiées par Hubel et Wiesel, en 1958. Le cortex strié a une épaisseur de 2 mm, une surface de quelques cm^2, et comporte environ 200 millions de cellules, que l'on peut différencier ainsi :

- les cellules à champ récepteur à symétrie circulaire centre/périphérie, similaires à celles que l'on peut rencontrer dans les corps genouillés latéraux,
- les cellules simples, qui ne répondent qu'à des lignes stimulatrices convenablement orientées, comme sur la figure 11, où l'on peut distinguer une cellule qui est excitée lorsqu'une ligne convenablement orientée recouvre son centre, une cellule qui est inhibée lorsque ce même événement se

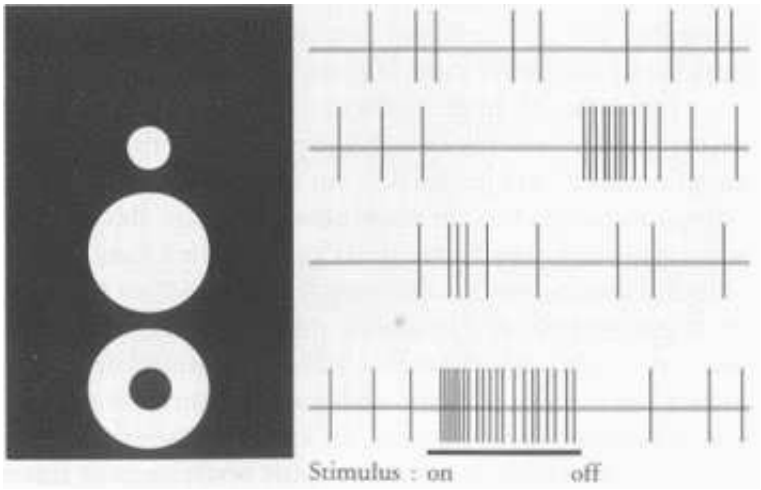

FIG. 9 – *Mise en évidence du champs récepteur d'une cellule à centre-OFF* (à gauche se trouve le stimulus lumineux et à droite l'activité de la cellule soumise à ce stimulus) *(reproduit à partir de [3]).*

produit, et une cellule qui distingue la frontière entre une zone sombre et une zone claire, si elle a la bonne orientation,

- les cellules complexes, qui représentent les trois quarts des cellules du cortex strié. Celles-ci réagissent, comme dans les cellules simples, préférentiellement aux lignes d'une orientation donnée. Par contre, contrairement aux cellules simples, ces lignes n'ont pas besoin d'être précisément positionnées dans le champs récepteur, mais elles doivent se déplacer latéralement (de manière à conserver leur orientation), même faiblement, pour être détectées.
- les cellules spécifiques d'un sens du mouvement, qui sont un cas particulier de cellules complexes, mais qui ne réagissent que si la ligne stimulatrice se déplace dans un sens, pas dans l'autre,
- les cellules à inhibition terminale, qui sont aussi un cas particulier de cellules complexes, où la ligne stimulatrice ne doit pas excéder une certaine longueur pour être convenablement détectée.

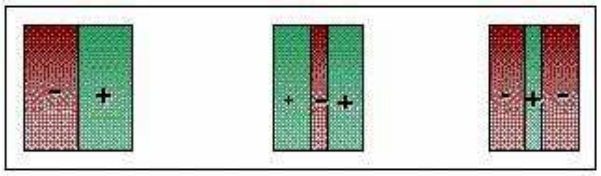

FIG. 10 – *Trois champs récepteurs typiques des cellules simples.*

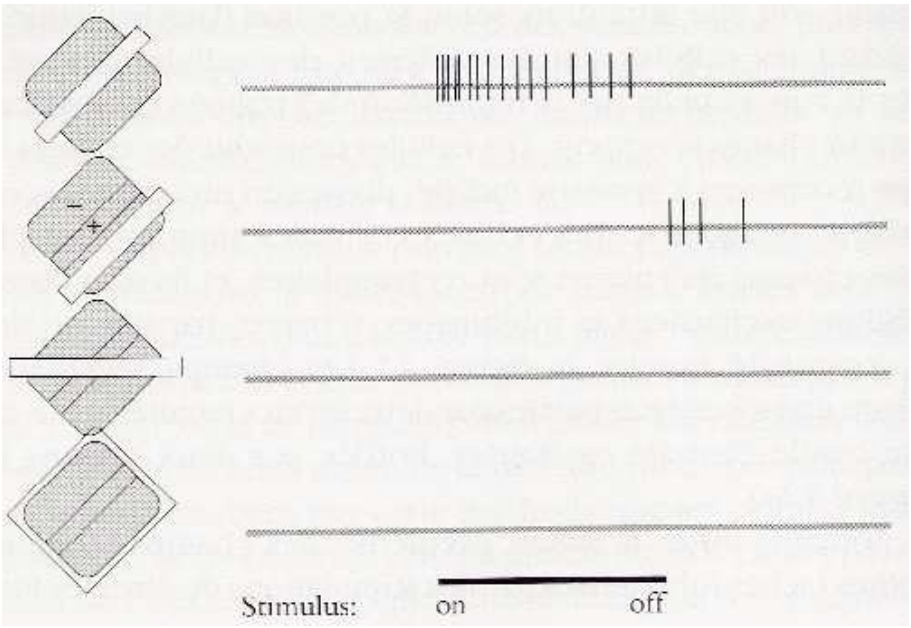

FIG. 11 – *Mise en évidence du champ récepteur d'une cellule simple (reproduit à partir de [3]).*

David Hubel a pu prouver l'origine des champs récepteurs des cellules simples : elles sont reliées à des cellules du corps genouillé latéral ayant des champs récepteurs assez proches mais un peu décalés.

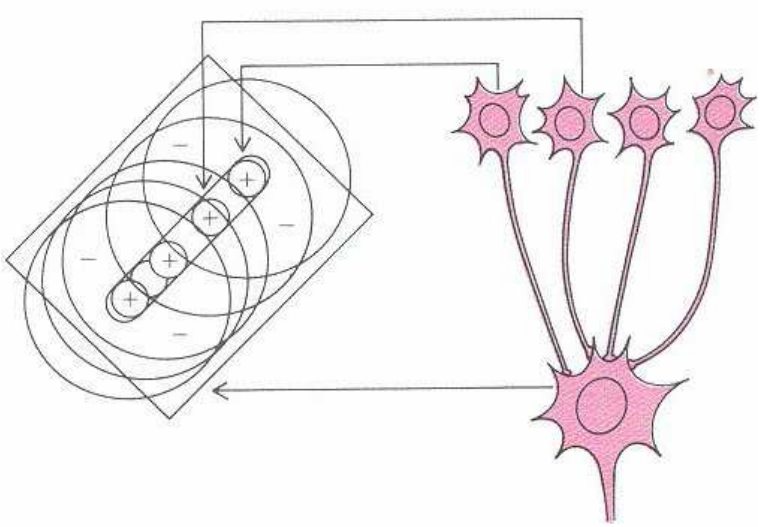

FIG. 12 – *Lien entre les champs récepteurs des cellules du LGN et les cellules simples (reproduit à partir de [3]).*

2.3.2 Architecture du cortex visuel primaire

La structure de couches du cortex strié a les propriétés suivantes : si l'on se déplace perpendiculairement à la surface, toutes les cellules ont le même champ récepteur, mais des propriétés différentes selon la couche à laquelle elles appartiennent (ce qui est du au fait qu'elles aboutissent à des zones différentes du cerveau).

Si l'on se déplace parallèlement à la surface, le champ récepteur varie de manière continue d'une cellule à l'autre. Il en va de même pour l'orientation préférée des cellules simples et complexes : en avançant de 0,05 mm le long de la surface du cortex, l'orientation préférée change d'environ 10°. On remarque également deux autres phénomènes, en plus de la variation linéaire de l'orientation préférée, se produisant à des intervalles aléatoires de quelques millimètres : des inversions du sens de rotation de l'orientation préférée, et des fractures, qui correspondent à un saut brutal de 45 à 90° combiné à une inversion du sens de rotation de l'orientation préférée.

2.4 Les colonnes corticales

L'organisation des neurones du cerveau est telle qu'en général, deux neurones ayant des activités assez proches sont situés à côté l'un de l'autre. C'est pourquoi, l'étude de certain phénomène peut souvent se faire sur un groupe de neurone ayant des fonctions très voisines. C'est ce regroupement qui est appelé colonne corticale.

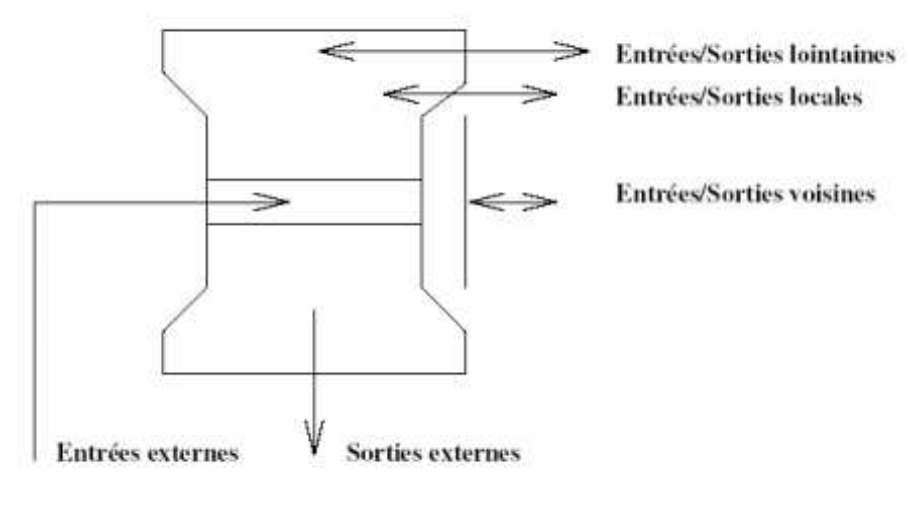

FIG. 13 – *Principe d'une colonne corticale.*

Dans le cadre du projet, avant de pouvoir faire directement l'étude de la colonne, plusieurs étapes sont nécessaires et auront été faites au préalable. Nous ne nous intéresserons qu'à l'analyse de l'activité de la colonne.

3 Les réseaux de neurones

3.1 Activité d'un neurone

En entrée, le neurone reçoit N signaux :

$$\{V_j, j = 1,..,N\}$$

où V_j représente l'activité d'un autre neurone.

Le lien entre les unités d'entrée et le neurone sont caractérisés par les $\{w_j, j = 1,..,N\}$ où w_j représente l'efficacité synaptique (aussi appelé couplage ou poids). Ce poids correspond au contact synaptique entre l'axone issu du neurone j et l'arbre dendritique du neurone concerné. S'il est positif, la synapse est excitatrice, s'il est négatif, la synapse est inhibitrice.

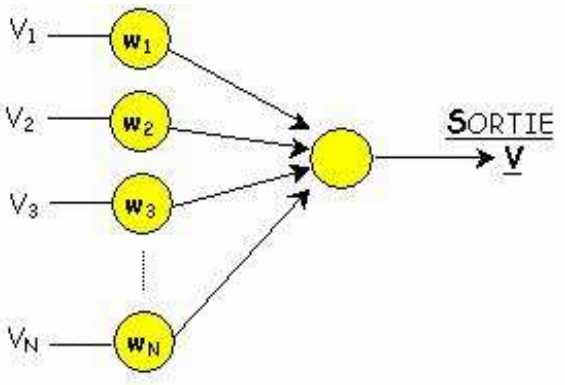

FIG. 14 – *Neurone artificiel (schéma reproduit à partir de* `http://asi.insa-rouen.fr/enseignement/siteUV/rna`*).*

La réponse du neurone se calcule ainsi :

$$V = \begin{cases} 1 & \text{si } h > \theta \\ 0 & \text{sinon} \end{cases}$$

où $h = \sum_{j=1}^{N} w_j.V_j$ est le potentiel post-synaptique du neurone et θ est le seuil d'activation du neurone.

Il s'agit ici d'un neurone à activité binaire.

3.2 L'apprentissage

3.2.1 Apprentissage par cœur

Lorsque le but de l´apprentissage est de mettre en mémoire des formes et de pouvoir les évoquer, ou alors de réaliser des associations particulières (comme associer le nom d´une personne à son image, par exemple), il correspond simplement à un apprentissage par cœur. L'architecture et la dynamique du réseau étant choisies, il reste à déterminer les valeurs des couplages. Ce sont eux qui contiendrons l´information sur les formes à mémoriser, et celle-ci sera distribuée entre tous les couplages : chaque couplage contiendra un peu d´information sur chacune des formes.

Pour réaliser une mémoire associative, on effectue un apprentissage *supervisé*. Pour chaque forme, le réseau est prié d´apprendre l´association entre la forme et la classe á laquelle elle appartient : les efficacités synaptiques sont modifiées suivant une règle d´apprentissage (ou algorithme). Après apprentissage, on teste la qualité associative du réseau : on souhaite qu´une forme apprise soit reconnue, même si on en présente une version bruitée.

3.2.2 Apprentissage par l´exemple

Apprentissage supervisé Ici, il s´agit de deviner une règle qu´illustrent des exemples d´une base d´apprentissage (ils sont appelés formes ou prototypes). Prenons un problème de classification, par exemple la reconnaissance de chiffres manuscrits. On constitue une base d´exemples en digitalisant une série d´images de chiffres. Si chaque image contient N pixels binaires, on va construire un réseau ayant N unités d´entrée et, par exemple 10 unités de sorties. On souhaite alors que, si une image représentant le chiffre k est présenté au réseau, alors la k^{ime} unité de sortie est activée et les autres restent inactives. Pour une architecture donnée, l´apprentissage va consister à modifier les

efficacités synaptiques de façon à ce que le réseau ait un bon comportement pour tous les exemples de la base d´apprentissage. On demande au réseau d´apprendre par cœur la base d´apprentissage, à ceci près qu´on ne va pas le tester sur cette base, mais sur un autre ensemble d´exemples, constituant la base de test. Il s´agit en effet de voir si, en apprenant des exemples, le réseau a extrait des régularités lui permettant d´identifier n´importe quel chiffre manuscrit.

Apprentissage non supervisé Dans certains cas, la base d´apprentissage contient un ensemble de formes à classer, mais les classes ne sont pas définies à l´avance. On va alors demander au réseau de regrouper les exemples en classes en fonction d´un critère de similarité choisi *a priori*. En général ce résultat est obtenu sans qu´il soit nécessaire de comparer directement les exemples entre eux. En pratique, on présente successivement les exemples sur la couche d´entrée, et on modifie les couplages par une règle choisie à l´avance. On parle alors d´apprentissage *non supervisé*. On emploie aussi le terme d´*auto-organisation* : partant d´un état initial quelconque, le réseau se structure «de lui-même».

3.3 Différents modèles

3.3.1 Le perceptron simple

Description Le perceptron est la brique de base d'un réseau. Un perceptron simple est composé de 2 couches de neurones : la couche d'entrée, aussi appelée rétine et la couche de sortie.
Pour calculer l'activité du neurone, on considère souvent que le seuil θ est nul. Cependant on peut aisément se ramener à ce cas en introduisant un neurone d'entrée (fictif) supplémentaire, V_0 dont l'efficacité synaptique est donnée par $w_0 = -\theta$. Comme ce neurone est toujours actif, sa valeur de sortie est toujours égale à 1 et son efficacité synaptique toujours prise en compte. On pose alors :

$$h' = \sum_{j=1}^{N} w_j.V_j + w_0 = h + w_0 = h - \theta$$

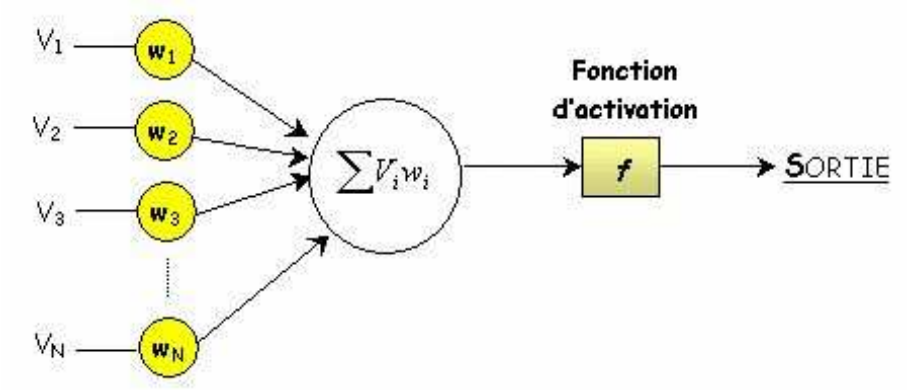

FIG. 15 – *Neurone artificiel ou perceptron simple (schéma reproduit à partir de* `http://asi.insa-rouen.fr/enseignement/siteUV/rna`*).*

Ainsi l'activité du neurone est donnée par :

$$V = \begin{cases} 1 & \text{si } h' > 0 \\ 0 & \text{sinon} \end{cases}$$

Jusqu'à présent, nous avons pris pour fonction d'activation, la fonction échelon qui est discontinue :

$$f(t) = \begin{cases} 1 & \text{si } t \geq 0 \\ 0 & \text{si t } < 0 \end{cases}$$

Pour rendre le problème continue on utilise la fonction sigmoïde ou par exemple la fonction suivante :

$$f(t) = \frac{1}{1 + e^{-x}}$$

Règle d'apprentissage du perceptron simple (algorithme de Widrow-Hoff)

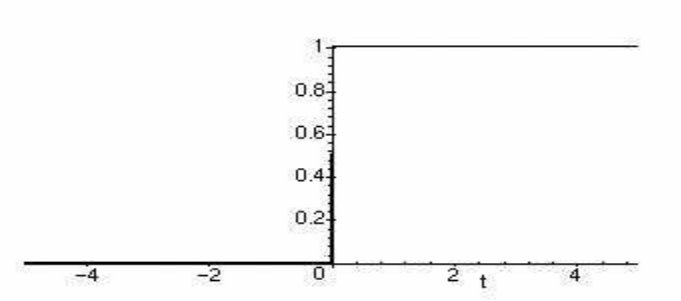

FIG. 16 – *Graphe de la fonction échelon (heaviside).*

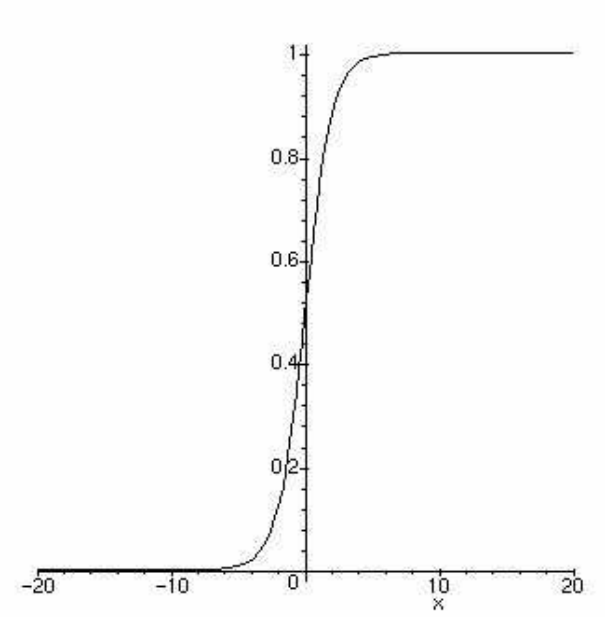

FIG. 17 – *Graphe d'une fonction d'activation continue.*

Principe Cette règle est locale, c'est à dire que l'activité d'un neurone ne dépend que de l'activité des neurones de la rétine (les neurones d'entrée). Un neurone ne modifie ses poids que s'il se trompe :

– Si le neurone de sortie est actif au lieu d'être inactif, alors les poids des neurones d'entrée actif sont diminué,

– Si le neurone de sortie est inactif au lieu d'être actif, alors les poids des neurones d'entrée actifs sont augmentés.

Ainsi, on présente au perceptron l'ensemble des stimuli à apprendre, dans un ordre arbitraire. Si, après que tous les stimulis aient été présentés, le perceptron commet toujours des erreurs, on lui présente à nouveau l'ensemble des stimuli, dans un ordre différent. Et ce jusqu'à ce qu'il ne commette plus d'erreur.

Algorithme

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \eta(V_i - y_i)x_j = w_{ij}^{(t)} + \Delta w_{ij}$$

- Δw_{ij} :la modification à effectuer sur le poids w_{ij}.
- x_j : la valeur de sortie du $j^{\text{ème}}$ neurone d'entrée.
- V_i : réponse désirée du $i^{\text{ème}}$ neurone de sortie.
- y_i : réponse du $i^{\text{ème}}$ neurone de sortie.
- $w_{ij}^{(t)}$: efficacité synaptique entre le $j^{\text{ème}}$ neurone d'entrée et le $i^{\text{ème}}$ neurone de sortie. À $t = 0$, on choisit généralement au hasard les valeurs de $w_{ij}^{(0)}$.
- η : constante positive, généralement comprise entre 0 et 1. Sa valeur influence la vitesse d'apprentissage du perceptron. Typiquement on peut prendre 0,75. Cependant, cette valeur peut varier avec t : par exemple, on peut partir de 0,9 puis décroître à chaque itération.

3.3.2 Le perceptron multicouche(Multi-Layer Perceptron : MLP)

Limites du perceptron classique Le perceptron classique ne peut pas résoudre les problèmes de classification non linéaire. Afin de palier à ce défaut, il existe plusieurs solutions envisageables, nous nous intéresserons à la plus populaire d'entre elles : le perceptron multi-couche.

Description et principe Un perceptron multi-couche peut être considéré comme une généralisation du perceptron classique.

Chaque neurone de la couche cachée calcule son activité en fonction des activités des neurones de la couche d'entrée. Puis les neurones de la couche cachée

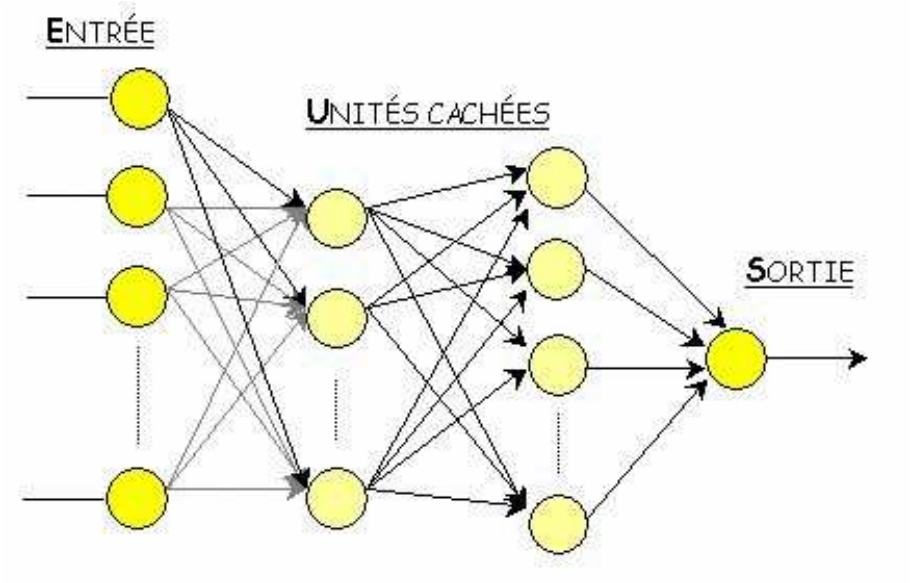

FIG. 18 – *Schéma d'un réseau "feed forward" en couche (schéma reproduit à partir de* `http://asi.insa-rouen.fr/enseignement/siteUV/rna`*).*

suivante calculent leur activité en fonctions de la couche précédente, et ainsi de suite jusqu'à la couche de sortie.

Algorithme d´apprentissage : Rétropropagation du gradient Cet algorithme est utilisé dans le cadre des réseaux multi-couche. On choisit donc une architecture avec un nombre de couches et de neurones adaptés au problème à traiter.

Position du problème

- $x = \{x_i, i = 1,..,n_0\}$: est un exemple à apprendre donné.
- w_{ij} : est le poids de la synapse entre le neurone j et le neurone i.
- $y_i^{(L)}$: est la réponse du $i^{\text{ème}}$ neurone de la couche L :

$$y_i^{(L)} = f(h_i^{(L)})$$

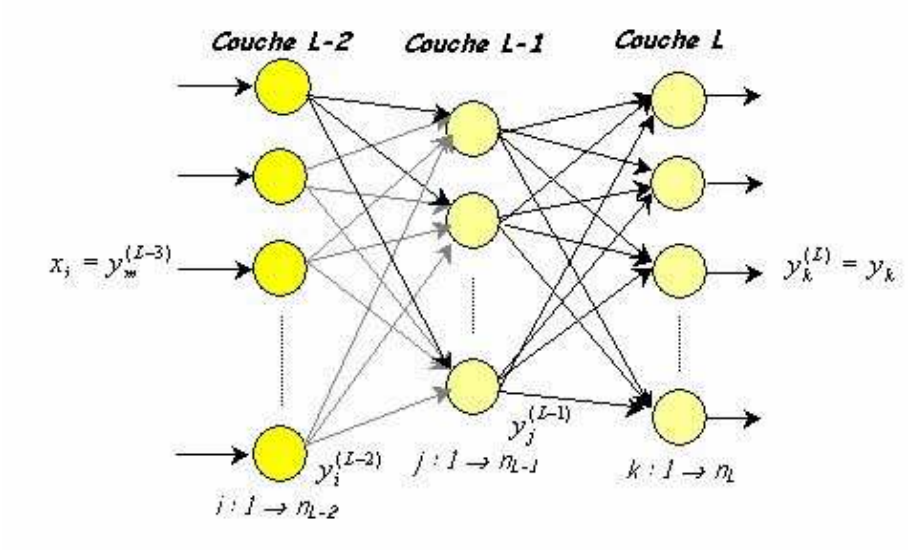

FIG. 19 – *Description des variables de l'algorithme de retropropagation (schéma reproduit à partir de* `http://asi.insa-rouen.fr/enseignement/siteUV/rna`*).*

où $f(x) = \frac{1}{1+e^{-x}}$, f est continue.

– $h_i^{(L)}$: est le potentiel post-synaptique du $j^{\text{ème}}$ neurone de la couche L :

$$h_k^{(L)} = \sum_{j=0}^{n_{L-1}} w_{kj}.y_j^{(L-1)}$$

où n_{L-1} est le nombre de neurones de la couche $L-1$. En effet, comme pour le perceptron simple, on désire prendre un seuil nul pour des commodités de calcul. Par conséquent on «rajoute» un neurone 0 à la couche précédente $(L-1)$ et son poids sera égal à l'opposé du seuil d'activité de chaque neurone de la couche L.

– On définit des fonctions coûts associées à chaque exemple :
soit $E_\mu = \frac{1}{2}.\sum_{i=1}^{N}(V_i^\mu - y_i^\mu)^2$ l'erreur associée à la forme μ , où :

- V_i^μ : est la réponse désirée du $i^{\text{ème}}$ neurone de sortie à la forme μ,

- y_i^μ : est la réponse du $i^{\text{ème}}$ neurone de sortie à la forme μ,
- N : est le nombre de neurones de sortie du MLP.

Puis on définit la fonction coût total : $E = \sum_{\mu=1}^{p} E_\mu$ où p est le nombre total de formes à apprendre.
Le but est donc de minimiser ce coût par rapport aux poids, en utilisant la méthode de la descente du gradient :
Chaque poids w_{ij} du réseau est itérativement modifié ainsi :

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$$

où η est le pas d'apprentissage.

Algorithme Pour un exemple x fixé *(pour des questions de lisibilité, on omet les indices μ)* :

– Considérons un neurone k de la couche de sortie (couche L) et un neurone j de la couche cachée L-1.
On recherche $\frac{\partial E}{\partial w_{kj}}$ pour connaître la variation du poids w_{kj} :

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial y_k^{(L)}} . \frac{\partial y_k^{(L)}}{\partial h_k^{(L)}} . \frac{\partial h_k^{(L)}}{\partial w_{kj}}$$

or

$\frac{\partial E}{\partial y_k^{(L)}} = -(V_k - y_k^{(L)})$ car $\begin{cases} y_k^{(L)} = y_k & \text{puisque L est la couche de sortie.} \\ E = \frac{1}{2}\sum_{i=1}^{N}(V_i - y_i)^2 \end{cases}$
$\frac{\partial y_k^{(L)}}{\partial h_k^{(L)}} = f'(h_k^{(L)})$ car $y_i^{(L)} = f(h_i^{(L)})$
$\frac{\partial h_k^{(L)}}{\partial w_{kj}} = y_j^{(L-1)}$ car $h_k^{(L)} = \sum_{j=0}^{n_{L-1}} w_{kj}.y_j^{(L-1)}$

on pose $Err_k^{(L)} = \frac{\partial E}{\partial h^{(L)}k} = -(V_k - y_k^{(L)}).f'(h_k^{(L)})$
dans notre cas :

$$Err_k^{(L)} = -(V_k - y_k^{(L)}).y_k^{(L)}.(1 - y_k^{(L)})$$

donc

$$\frac{\partial E}{\partial w_{kj}} = Err_k^{(L)}.y_j^{(L-1)}$$

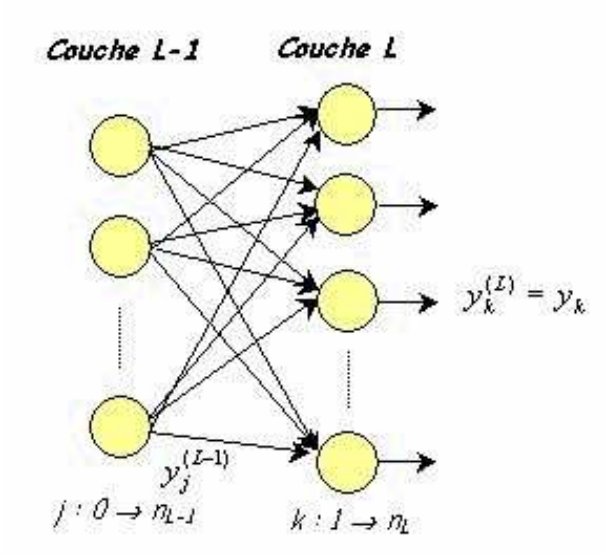

FIG. 20 – *Description des variables de l'algorithme de rétro-propagation (la couche L-1 a un neurone de plus pour les questions de seuil) (schéma reproduit à partir de* `http://asi.insa-rouen.fr/enseignement/siteUV/rna`*).*

– A présent considérons un neurone j de la couche L-1 et un neurone i de la couche L-2.
On recherche $\frac{\partial E}{\partial w_{ji}}$ pour connaître la variation du poids w_{ji} :

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial y_j^{(L-1)}} . \frac{\partial y_j^{(L-1)}}{\partial h_j^{(L-1)}} . \frac{\partial h_j^{(L-1)}}{\partial w_{ji}}$$

or

$$\frac{\partial E}{\partial y_j^{(L-1)}} = \sum_{k=1}^{n_L} \frac{\partial E}{\partial h_k^{(L)}} . \frac{\partial h_k^{(L)}}{\partial y_j^{(L-1)}} = \sum_{k=1}^{n_L} Err_k^{(L)} . w_{kj} \left\{ \begin{array}{l} \text{car} \quad h_k^{(L)} = \sum_{k=1}^{n_{L-1}} w_{kj} y_j^{(L-1)} \\ \text{et par définition de} \quad Err_k^{(L)}. \end{array} \right.$$

$$\frac{\partial y_j^{(L-1)}}{\partial h_j^{(L-1)}} = f'(h_j^{(L-1}) \quad \text{car} \quad y_j^{(L)} = f(h_i^{(L)}$$

$$\frac{\partial h_j^{(L-1)}}{\partial w_{ji}} = y^{(L-2)}i \quad \text{car} \quad h_j^{(L-1)} = \sum_{i=0}^{N_{L-2}} w_{ji} . y_i^{(L-2)}$$

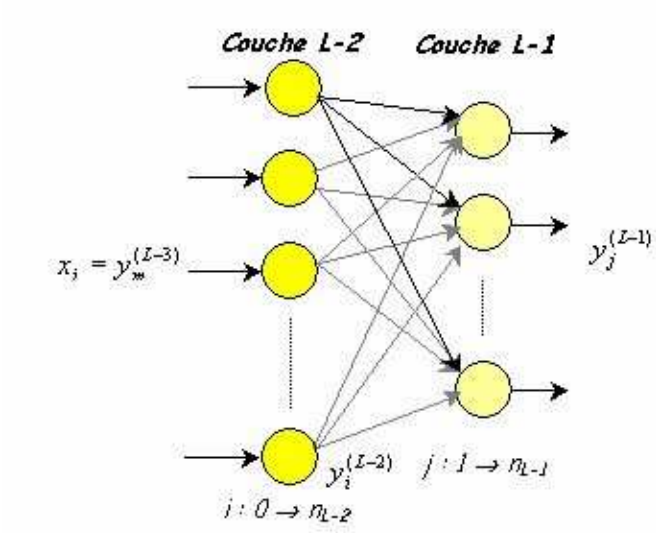

FIG. 21 – *Description des variables de l'algorithme de rétro-propagation (schéma reproduit à partir de* `http://asi.insa-rouen.fr/enseignement/siteUV/rna`*).*

on pose $Err_j^{(L-1)} = \frac{\partial E}{\partial h_j^{(L-1)}} = \left(\sum_{k=1}^{n_L} Err_k^{(L)}.w_{kj} \right).f'(h_j^{(L-1)})$,

dans notre cas :

$$Err_j^{(L-1)} = \left(\sum_{k=1}^{n_L} Err_k^{(L)}.w_{kj} \right).y_j^{(L-1)}.(1 - y_j^{(L-1)})$$

donc

$$\frac{\partial E}{\partial w_{ji}} = Err_j^{(L-1)}.y_i^{(L-2)}$$

- On réitère cette procédure de couche en couche jusqu'à la couche d'entrée.
- La modification se fait en deux passes :
  - dans un premier temps on calcule les modifications de poids sans les toucher.
  - les poids sont modifiés comme calculé dans la précédente passe.

Pas d'apprentissage Le choix du pas d'apprentissage est très important : s'il est trop petit, la convergence de l'algorithme est trop lente, s'il est trop grand il risque d'y avoir des oscillations. Pour l'adapter, il existe plusieurs solutions couramment employées :

- diminuer le pas d'apprentissage au fur et à mesure ("à la main" ou en fonction de la forme de la surface d'erreur).
- approximations du premier ordre :
  - *rétro-propagation avec un moment d'inertie* (Rumelhart et al. 1986) :

$$\Delta w_{ij}(t) = -\eta.Err_j.x_i + \alpha.\Delta w_{ij}(t-1) \quad \text{avec} \quad |\alpha| < 1$$

  - *Delta-Bar-Delta* (Jacobs 1988) :
    Tout d'abord il faut calculer un "gradient moyen" :

$$\delta_{ij}(t) = (1-\Phi)Err_j.x_i + \Phi\delta_{ji}(t-1)$$

    Puis le pas d'apprentissage est modifié selon la direction du gradient par rapport au gradient moyen :

$$\eta_{ji}(t+1) = \begin{cases} \eta_{ji}(t) + u & \text{si} \quad Err_j.x_i(t).\delta_{ji}(t-1) > 0 \\ \eta_{ji}(t).d & \text{si} \quad Err_j.x_i(t).\delta_{ji}(t-1) < 0 \\ \eta_{ji}(t) & \text{sinon} \end{cases}$$

    et

$$\Delta w_{ij}(t) = -\eta_{ji}(t).Err_j.x_i$$

  - *Rprop* (Riedmiller and Braun 1993) :
    Le pas d'apprendtissage est borné et modifié selon la direction du gradient par rapport au gradient précédent.

$$eta_{ji}(t{+}1) = \begin{cases} \min(\eta_{ji}(t).u, \eta_{max}) & \text{si} \quad (Err_j.x_i)(t).(Err_j.x_i)(t-1) > 0 \\ \max(\eta_{ji}(t).d, \eta_{min}) & \text{si} \quad (Err_j.x_i)(t).(Err_j.x_i)(t-1) < 0 \\ \eta_{ji}(t) & \text{sinon} \end{cases}$$

    Le poids n'est alors modifié que s'il va "dans le bon sens" :

$$\Delta w_{ij}(t{+}1) = \begin{cases} -\eta_{ji}(t).sgn(Err_j.x_i) & \text{si} \quad (Err_j.x_i)(t).(Err_j.x_i)(t-1) \geq 0 \\ 0 & \text{sinon} \end{cases}$$

– approximations du second ordre : Considérons le développement de Taylor de la fonction de coût :

$$E(w+h) \approx E(w) + \Big(\frac{\partial E}{\partial w}\Big)' h + \frac{1}{2} h' H(w) h$$

où H est une matrice Hessienne, le «Hessien» du coût.

$$H = \begin{pmatrix} \frac{\partial^2 E}{\partial w_1^2} & \frac{\partial^2 E}{\partial w_1 \partial w_2} & \cdots & \frac{\partial^2 E}{\partial w_1 \partial w_n} \\ \frac{\partial^2 E}{\partial w_2 \partial w_1} & \frac{\partial^2 E}{\partial w_2^2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial w_n \partial w_1} & \cdots & \cdots & \frac{\partial^2 E}{\partial w_n^2} \end{pmatrix}$$

Le gradient s´écrit donc :

$$\frac{\partial E(w+h)}{\partial w} \approx \frac{\partial E(w)}{\partial w} + h' H(w)$$

Le but est de trouver h tel que le gradient soit nul :

$$\Delta w = h = -H^{-1}(w) \frac{\partial E(w)}{\partial w}$$

Le problème se situe au niveau du calcul de H^{-1}. Voici donc des solutions qui proposent des approximations du Hessien :

- *Assimiler le hessien à une matrice diagonale* :

$$\Delta w(t) = -\frac{\frac{\partial E}{\partial w}}{\frac{\partial^2 E}{\partial w^2}}$$

- *QuickProp(Fahlman 1988)* : On évite de calculer H,

$$\frac{\partial^2 E}{\partial w^2} \approx \frac{\frac{\partial E(w(t+1))}{\partial w} - \frac{\partial E(w(t))}{\partial w}}{\Delta w(t)}$$

d´où

$$\Delta w(t) = -\Delta w(t-1) \frac{\frac{\partial E(w(t))}{\partial w}}{\frac{\partial E(w(t))}{\partial w} - \frac{\partial E(w(t-1))}{\partial w}}$$

- *méthodes de gradient conjugué, Levenberg Marquard, ...*

3.3.3 Modèle de Hopfield

Hopfield networks. Formellement, un réseau de neurones «à la Hopfield» est un système dynamique discret définit à partir d´un ensemble de N signaux binaires $\mathbf{s} = (\cdots s_i, \cdots)$ avec $s_i \in \{-1,1\}$. Les signaux sont actualisé par une équation récurrente de la forme :

$$s_i^{t+1} = Sgn\left(\sum_{j=1}^{N} \omega_{ij}\, s_j^t - \theta_i\right)$$

Remarque : Dans certaines architectures cette fonction binaire (Sgn) est remplacée par une fonction du type «sigmoïde».

Afin d´actualiser s_i^{t+1} à partir de s_i^t, une des deux règles suivante est, en général utilisée :

- Mise à jour *parallèle*, où tous les signaux s_i^t sont actualisés simultanément,
- Mise à jour *aléatoire*, oú *un seul* signal 'la fois, (sélectionné au hasard) est mis séquentiellement à jour.

Dans les réseaux de *Hopfield*, $w_{ij} = w_{ji}$ et $w_{ii} \geq 0$, lorsque la mise à jour aléatoire est utilisée. C´est pourquoi nous pouvons assimiler ce système dynamique à une «énergie» bornée :

$$E = -\frac{1}{2}\sum_i (\sum_j w_{ij}\, s_j^t - 2\,\theta_i)\, s_i^t$$

ainsi, quand un signal s_i^t est actualisé, nous pouvons facilement vérifier que $\delta E < 0$. Puisque cette énergie est bornée et décroît avec le temps, elle doit converger vers une valeur stationnaire, correspondant à un point d´équilibre stable.

Un tel système dynamique n´a ainsi que des points d´équilibre comme attracteurs. Évidemment, ils sont caractérisés par le fait que $\mathbf{s}$ n´est actualisé que si et seulement si $\forall i, s_i\,(\sum_j \omega_{ij}\, s_j - \theta_i) \geq 0$.

Règle d´apprentissage de Hebb Selon le paradigme de *Hebb*, considérons R «prototypes» $\mathbf{s}^r, r = 1..R$ composés de signaux binaires.

Ici, les données sont reliées au signal par des fonctions «binaires» qui peuvent être écrites sous la forme $\mathbf{s} = \Psi(\mathbf{x})$.

Chaque prototype doit être évalué par une méthode quelconque permettant de choisir une valeur typique, comme par exemple moyenner les données de calibration, i.e. $\mathbf{s}^r = \Psi\left(\frac{\sum_{i,r_i=r} \mathbf{x}_i}{\sum_{i,r_i=r} 1}\right)$.

Le réseau de Hopfield auxiliaire est défini avec : $\begin{cases} \omega_{ii} = 0 \\ \theta_i = 0 \\ \omega_{ij} = \sum_{r=1}^{R} s_i^r \, s_j^r \end{cases}$.

Pour cette règle d´apprentissage, la condition évoquée dans le paragraphe précédent pour qu´un prototype $\mathbf{s}^{r_0}$ soit un attracteur de l´énergie du réseau :

$$s_i^{r_0} \sum_j w_{ij} \, s_j^{r_0} = N - 1 + \sum_{r \neq r_0} s_i^r \, s_i^{r_0} \sum_j s_j^r \, s_j^{r_0} \geq 0$$

apparaît lorsque les prototypes sont décorrélés (ou linéairement indépendant), i.e. $\sum_j s_j^r \, s_j^{r_0} \simeq 0$.

Plus précisément, en considérant la probabilité pour qu´une somme de nombres binaires $s_i \in \{-1,1\}$ soit inférieure à un seuil donné τ,

$$p(\sum_{i=1}^{q} s_i < \tau) = Q(\frac{\tau}{\sqrt{q}})$$

avec

$$Q(u) = \frac{1}{\sqrt{2\,\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}u^2}$$

il a été prouvé que nous pouvons mémoriser de l´ordre de $R = o(\frac{N}{2\log(N)})$ prototypes dans un réseau de taille N avec une probabilité raisonnable (typiquement $P > 0.99$) de réussite. Autrement, le réseau a un comportement catastrophique et converge vers n´importe quoi.

La règle d´apprentissage peut aussi être écrite sous forme incrémentale :

$$\omega_{ij} = \alpha \, \omega_{ij} + \beta \, s_i^r \, s_j^r$$

Si $\alpha = \beta = 1$, cette forme est équivalente à la précédente.

Si $0 < \alpha < 1$ et $\alpha + \beta = 1$ la mémorisation des prototypes est réalisée avec quelques faiblesses : les plus anciens sont moins pris en compte. Ce mécanisme alors appelé *palimpestre* permet de borner le contenu du réseau et aussi d´éviter la surcharge.

Une autre règle d´apprentissage, dite «anti-»Hebb de la forme

$$\omega_{ij} = \omega_{ij} - \beta\, s_i^r\, s_j^r$$

pour β petit et positif peut être utilisée. Cette dernière n´est pas utilisée directement pour la mémorisation d´un prototype donné, mais plutôt pour *dé-correler* ce prototype du réseau. Elle est aussi utilisée par des mécanisme de correction d´erreurs.

Ces différentes stratégies conduisent à la classification décrite dans la figure 22.

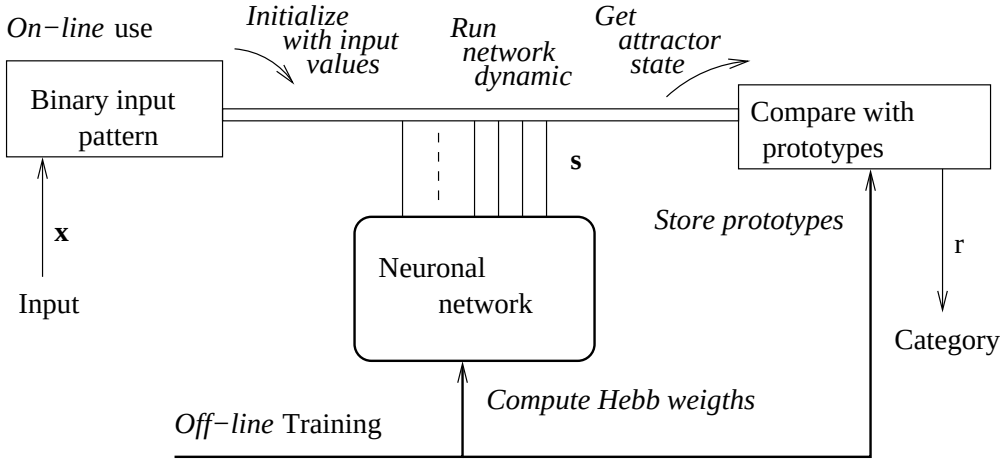

FIG. 22 – *Architecture d´un classificateur de Hebb : (A) il est* entraîné *hors ligne en calculant des poids de Hebb pour chaque prototype et en mémorisant ces prototype par comparaison; (B) il est* utilisé *en trois étapes : pour une entrée donnée (i) le prototype correspondant est calculé et utilisé comme une initialisation du réseau, (ii) la dynamique du réseau tourne jusqu'à atteindre des points d´équilibre, (iii) le prototype obtenu est comparé à chaque prototype et la catégorie correspondant au prototype le plus proche et la sortie.*

Outre toute plausibilité biologique, ce classificateur est un simple classificateur linéaire basé sur des prototypes binaires pour définir sa calibration. Un dispositif intéressant est que nous pouvons avoir *transformé* l´entrée pour ob-

tenir une forme plus appropriée. La dynamique neuronale calcule une formule générale (mais pas facile à analyser) du calcul de distance au prototype.

4 Le modèle de Heeger

4.1 Quelques définitions utiles

contraste : Valeur maximale du contraste local d'un stimulus. La valeur maximale possible du contraste est 1. Cette valeur est atteinte lorsque l'intensité la plus petite est nulle et la plus grande égale à deux fois l'intensité moyenne du signal.

énergie locale : Variance du stimulus, locale (dans l'espace et le temps) et dans une bande de fréquence spatio-temporelle.

fréquence spatio-temporelle : On applique la transformée de Fourier sur des fenêtres de temps du signal. Ainsi, on peut voir les fréquences utilisées apparaître et disparaître au cours du temps, selon la taille de la fenêtre utilisée. Cette transformée permet donc d'avoir à la fois le contraste (transformée de Fourier spatiale) et le mouvement (transformée de Fourier temporelle).

intensité : On considère ici que ce terme désigne l'intensité classique combinée avec le contraste local. Cette définition est impropre mais permet d'éviter des confusions.

damier : Ce terme désigne un stimulus appelé «Plaid» par Heeger dans [5]. Il est composé de plusieurs grilles ayant des fréquences spatiales et une orientation différentes, mais la même fréquence temporelle. Ce stimulus ressemble donc en quelque sorte un tissu écossais.

4.2 Modèle linéaire d'une cellule simple

4.2.1 Le modèle

Heeger a exposé, dans un premier temps, un modèle linéaire des cellules simples.

Ce modèle est attrayant car s'il était correct, il permettrait de prédire la réponse d'une cellule simple à n'importe quel stimulus visuel, basé sur un nombre limité de mesures.

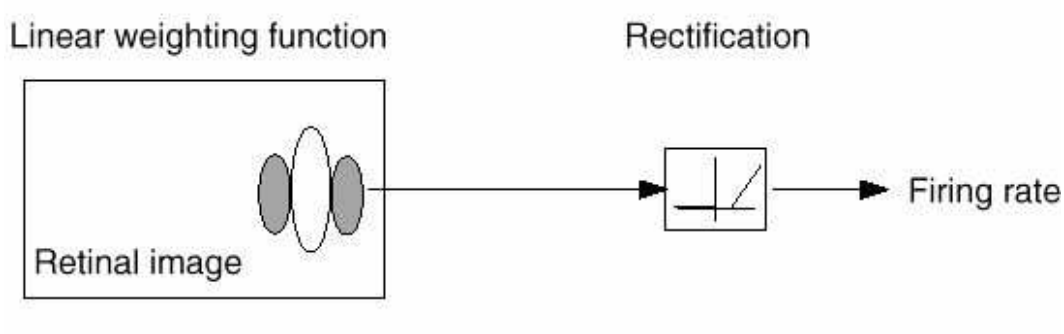

FIG. 23 – *Modèle linéaire d'une cellule simple (reproduit à partir de [5]).*

Définition d'un stimulus On décrit l'intensité d'un stimulus visuel par sa distribution d'intensité : I(x,y,t) où x et y sont les coordonnées spatiales et t la coordonnée du temps.

Remarque : l'espace n'est qu'en 2 dimensions car c'est le projeté sur la rétine qui est considéré.

Cette représentation ne tient pas compte de la couleur du stimulus et suppose une vision monoculaire. Mise à part ces restrictions, elle est assez complète.

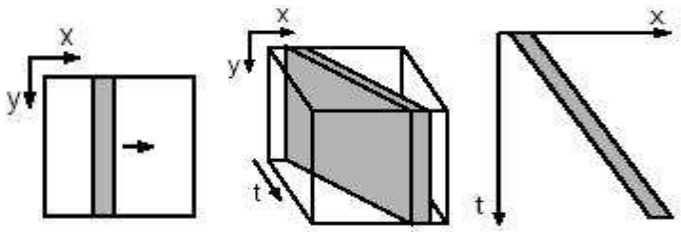

FIG. 24 – *Exemple de stimulus spatio-temporel : une barre mobile* (Tout d'abord dans le plan, puis avec le temps en plus et en troisième le stimulus en fonction dúne coordonné de l'espace et le temps) *(reproduit à partir de [5]).*

Les fonctions poids Heeger propose le modèle suivant de réponse d'une cellule simple linéaire à un stimulus I(x,y,t) :

$$L(t) = \iiint W(x,y,T).I(x,y,t-T)\,dx\,dy\,dT$$

W(x,y,t) est la fonction poids :

- Elle est nulle dans le futur.
- Si cette fonction est oblique dans le plan (x,t) (cf. Figure 24), alors la cellule a une direction préférée de mouvement.
- Sinon, elle n'a pas de direction préférée.

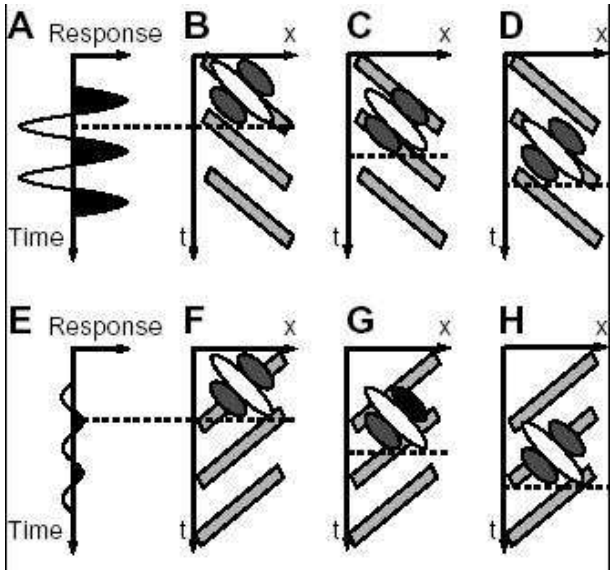

FIG. 25 – *Réponse d'une cellule linéaire avec une fonction poids orientée espace-temps à une grille mobile (reproduit à partir de [5]).*

Remarque : si la fonction poids est séparable en "espace-temps", c'est à dire qu'il existe des fonctions h et g telle que $W(x,y,t) = h(x,y).g(t)$, alors la cellule considérée n'a pas de direction préférée de mouvement.

Non-linéarités

L'adaptation à la lumière Notre propre expérience nous donne un exemple de non-linéarités négligé lors de la définition du modèle : l'adaptation de la rétine à la lumière. Cependant, nous pouvons aisément ignorer ce problème en ne considérant que des stimuli ayant une distribution lumineuse modulée autour d'une luminance moyenne m fixée.
Ainsi, la rétine peut être considérée comme étant dans un état fixé d'adaptation et sa sortie est proportionnelle au «contraste local» du stimulus :

$$I(x,y,t) = \frac{l(x,y,t) - m}{m}$$

La rectification Une cellule simple ne libérant un train de potentiels d'action que si le potentiel membranaire dépasse un certain taux (cf. Figure 26) un post-traitement a été rajouté au modèle : une rectification finale, un seuillage (cf. Figure 23).

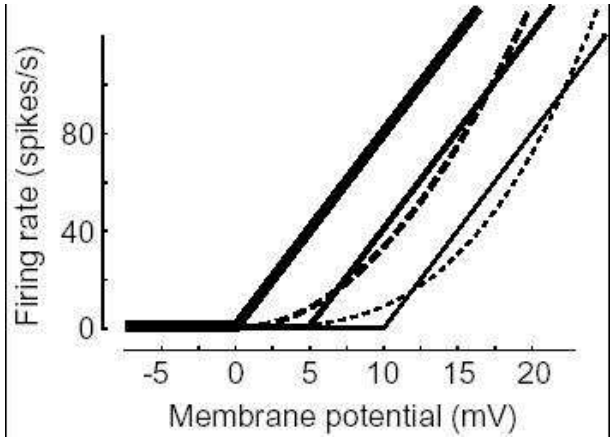

FIG. 26 – *Fréquence de d'échange des potentiels d'action en fonction du potentiel membranaire (reproduit à partir de [5]).*

4.2.2 Quelques propriétés linéaires des cellules simples

Réponses aux impulsions

les cellules ON et OFF : Hubel et Wiesel (1962) mirent en évidence pour la première fois les champs récepteurs des cellules V1 en utilisant des barres lumineuses et brèves *(«bright flashing bars»)*. Ce phénomène peut être prédit à l'aide du modèle linéaire, comme le montre la figure 27.

la corrélation inverse ou corrélation entre la fréquence de décharge et l'intensité lumineuse. Des séquences de barres lumineuses ayant différentes positions ont permis d´établir la corrélation entre les trains de potentiels d'action et l'intensité lumineuse. Cette dernière dépend du temps et de l'espace et sa valeur donne la force de la fonction poids à cet instant et cette position. Cette méthode appelée *corrélation inverse* permet de mesurer dans l'espace et le temps des fonctions poids.
Cette méthode peut être aussi utilisé pour définir des fonctions poids de cellules non-linéaires. Toutefois, c´est seulement dans le cas de cellules linéaires que ces fonctions poids peuvent être utilisés pour prédire la réponse de la cellule à n'importe quel stimulus visuel.

Réponses aux grilles mobiles : Un stimulus de choix pour l'analyse du système visuel par des systèmes linéaires est la grille sinusoïdale *(«sine grating»)* (c'est à dire un stimulus dont la luminance varie sinusoïdalement en temps et en espace), puisque les systèmes linéaires y répondent par une sinusoïde.
Une grille peut être considérée comme une somme de barres lumineuses. Par conséquent, la réponse d'une cellule linéaire à ce type de stimulus peut être prédite à partir de sa réponse aux barres (via une transformée de Fourier).
De même une barre lumineuse peut être vue comme une somme de grilles lumineuses. Donc la réponse d'une cellule linéaire à une barre lumineuse peut être prédite à partir de sa réponse aux grilles (via une transformée de Fourier inverse).
Par expérimentation on trouve un bon rapport entre les fonctions poids prédites par la transformée de Fourier inverse de la sensibilité des grilles et les

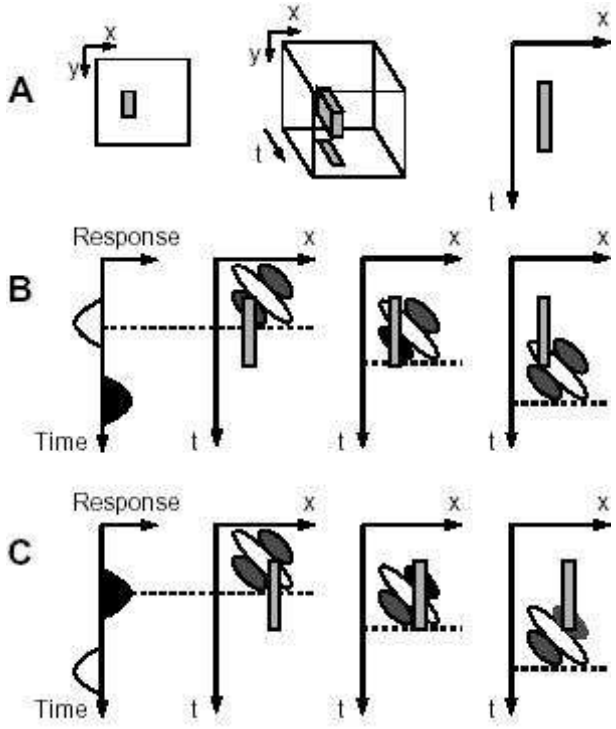

FIG. 27 – *Réponse d'une cellule linéaire à une impulsion lumineuse.*Cette impulsion ne varie pas dans le temps c´est pourquoi elle est représentée par un rectangle dans l'espace (x-t). Le haut et le bas du rectangle correspondent au début et à l'arrêt de la stimulation. Lorsque le stimulus excite la partie «OFF» du champs récepteur (partie noire), la réponse de la cellule est négative, lorsque le stimulus excite la partie «ON» du champs récepteur (partie blanche), la réponse est positive. Lorsque les deux parties sont excitées, rien ne se passe, c'est là qu'intervient la rectification. *(reproduit à partir de [5])*

fonctions poids obtenues à partir des barres lumineuses. Ceci soutient le modèle linéaire des réponses des cellules simples.
Le décalage entre la réponse aux grilles et la réponse aux impulsions peut être expliqué par la sur-rectification. Ce qui est encore compatible avec le modèle

linéaire.

Réponses aux grilles au contraste modulé : Une grille à contraste modulé est une grille sinusoïdale *(«sine grating»)* dont l'intensité est modulée sinusoïdalement dans le temps.
La réponse des cellules simples à ces grilles mobiles est assez équivalente à une sinusoïde dont la rectification a éliminé tout ce qui se trouve en dessous d'un certain seuil. Ceci coïncide toujours avec le modèle linéaire (complété d'une rectification).
D'après le principe de superposition la réponse d'une cellule linéaire à une grille au contraste modulé devrait être aisément prédite à partir de ses réponses à un grille mobile et vice versa. Cependant les cellules simples répondent non-linéairement dans deux cas :

- La prédiction linéaire à partir d'une grille au contraste modulé sous-estime le degré de la *«direction préférée»* observé avec les grilles mobiles. La rectification permet d'adapter le modèle linéaire à ce problème.
- La prédiction linéaire surestime la réponse aux grilles glissantes pour les cellules n'ayant pas de *«direction préférée»*. Dans la plupart des cas ce phénomène peut être expliqué par un mécanisme de contrôle du gain comme dans le modèle normalisé exposé plus loin. En effet, la normalisation permet de prédire que les cellules sont moins réceptives en présence de grilles mobiles qu´en présence de grilles au contraste modulé d'égal contraste.

Réponses aux stimuli composés D´après le modèle linéaire, connaître les réponses de cellules simples à des grilles lumineuses devrait permettre de prédire n´importe quel stimulus visuel, puisque ces derniers peuvent être exprimés à partir de la somme de différentes grilles lumineuses.
En 1979, De Valois réalisa un test de prédiction de la réponse des cellules simples pour deux modèles : le modèle linéaire et un modèle basé sur la détection des contours. Il utilisa deux types de stimuli lumineux, des grilles simples et des damiers. C'est pour ces derniers qu´une différence apparut dans les prédictions des deux modèles. En effet, d´après le modèle basé sur la détection de contours, les contours forts d´un damiers sont orientés le longs des lignes et des

colonnes. Par conséquent, les cellules les plus activées devrait être celles ayant des directions préférées horizontales et verticales. Du point de vue du modèle linéaire, la composante la plus forte du damier est une grille sinusoïdale *(«sine grating»)* orientée selon les diagonales. Ainsi les cellules ayant une direction préférée orientée selon la diagonale devrait répondre plus activement.
Les résultats de l´expérience de De Valois validèrent le modèle linéaire et repoussèrent le modèle basé sur la détection de contours.
D´autres expériences ont par la suite été effectuées en composant des grilles sinusoïdales *(«sine grating»)* de différentes fréquences et orientations. Les résultats obtenus ont tous validé qualitativement le modèle linéaire. Cependant, au point de vue quantitatif, les résultats étaient loin de la réalité, ce qui, à nouveau, a laissé apparaître la notion de contrôle du gain du modèle normalisé.

4.3 Le modèle normalisé

4.3.1 Quelques propriétés non linéaires des cellules simples

Réponse au contraste D´après le modèle linéaire, une cellule simple devrait répondre proportionnellement à une augmentation du contraste, or ce n´est pas du tout le cas dans la réalité: lorsque le contraste double, par exemple, la réponse de la cellule va à peine augmenter. Ce phénomène est appelé *«saturation de la réponse»*. Et même, dans le cas de «supersaturation», ces cellules peuvent répondre à une augmentation du contraste par une baisse de de l´amplitude de leur réponse.
Le phénomène de saturation dépend du contraste du stimulus et non de l´amplitude de la réponse qu´il provoque chez la cellule.
Si l´on considère les réponses d´une cellule à des grilles lumineuses ayant des fréquences spatiales différentes, leur rapport sera constant, indépendamment du contraste du contraste du stimulus. La propriété est aussi valable pour différentes orientations des grilles. Elle pourrait être expliquée aisément si les cellules étaient linéaires, mais elles ne le sont pas dans la réalité puisqu'elle saturent souvent aux forts contrastes.

Suppression non spécifique Une cellule peut ne pas répondre à son «stimulus préféré» si on lui ajoute un stimulus supplémentaire qui n´aurait pro-

voqué aucune réponse s´il avait été seul. Ce phénomène contredit le principe de superposition donc la linéarité de la réponse de la cellule supposée pour le modèle linéaire. Ce phénomène est appelé *«supression non-spécifique»* et est indépendant de la direction préférée de mouvement, de l´orientation et des fréquences spatiales et temporelles du stimulus.

Non-linéarités temporelles Les cellules simples ont d´importantes non-linéarités temporelles, par exemple, plus le contraste d´un stimulus sera fort, plus la réponse de la cellule apparaîtra rapidement. Ce phénomène est appelé *avance de phase* et n´est pas cohérent avec le modèle linéaire : multiplier par deux l´entrée d´un neurone linéaire conduit à la multiplication par deux de la sortie, pas à la modification de son temps d´exécution. L´avance de phase n´a pas une origine entièrement corticale, mais sa composante corticale est très importante.
La composition de huit stimuli différents donne une réponse qui apparaît plus tôt que celle qu´on aurait obtenue par combinaison linéaire à partir des réponses aux stimuli exercés indépendamment les uns des autres.
Pour finir, la réponse d´une cellule simple à une fréquence temporelle dépend du contraste du stimulus. En particulier, augmenter ce dernier augmente la sensibilité de la cellule aux hautes fréquences temporelles.

4.3.2 Le modèle

Les quelques non linéarités non corrigés par la rectification peuvent être expliqués par un stade d'inhibition fractionnaire (cf. Figure 28). Ce stade permet de contrôler le gain (la réponse) de la cellule en moyennant le retour intra-cortical d'un grand nombre d'autres cellules corticales. Ce modèle amélioré est appelé *modèle normalisé* («normalization model»).

La spécificité du modèle réside dans cette normalisation. De plus, cette dernière est d'autant plus plausible qu'elle peut s'expliquer biologiquement, par adaptation de l'impédance de la couche neuronale. En effet, plus la cellule décharge, plus l'impédance baisse, donc plus il y a court-circuit et donc plus la décharge sera faible.

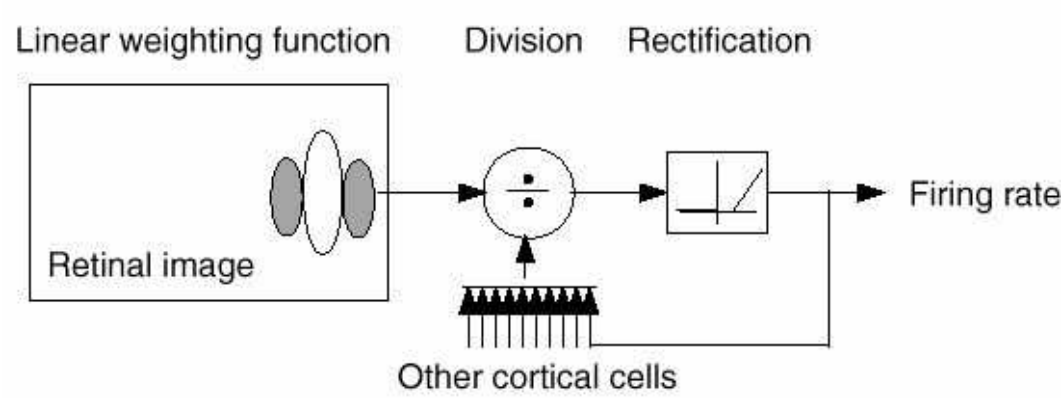

FIG. 28 – *Modèle normalisé linéaire d'une cellule simple (reproduit à partir de [5]).*

4.3.3 Tests du modèle normalisé

Les variables du modèle normalisé dépendent les unes des autres de manière circulaire : la fréquence de décharge de chaque cellule dépend de son potentiel membranaire ; le potentiel membranaire de chaque cellule dépend du courant qui la traverse et de sa conductance ; la conductance de chaque cellule dépend de la fréquence de décharge des cellules du «pool de normalisation».

Réponse aux grilles L´amplitude de la réponse est donnée par l´équation suivante :

$$amplitude(R) \propto \left[amplitude(L)\frac{c}{\sqrt{\sigma(f)^2 + c^2}}\right]^n$$

où :

c est le contraste de la grille,
f est la fréquence temporelle de la grille,
$L(t)$ est la fonction poids d´une cellule linéaire lorsque la grille a un contraste unitaire (cette fonction est sinusoïdale),
$\sigma(f)$ est relié aux propriétés «passe-bas» de la membrane cellulaire,
n est relié à l´étape de rectification qui transforme le potentiel membranaire en fréquence de décharge.

D´après le modèle linéaire, l´amplitude de la réponse est donnée par :

$$amplitude(L).c$$

Le modèle normalisé introduit une étape de normalisation par une quantité qui dépend de l´activité d´un grand nombre de neurones : $\sqrt{\sigma(f)^2 + c^2}$.

Cette équation est donc assez similaire à ce qui avait était trouvé empiriquement pour prédire les réponses au contraste de l´aire V1. Cependant le modèle ne s´arrête pas là, il prédit aussi leur dépendance vis à vis des fréquences spatiales et temporelles et de l´orientation. Afin de tester ces prédictions, les résultats du modèle ont été comparés à un vaste ensemble de données provenant de stimulations de cellules simples par des grilles mobiles de contraste, d´orientations, de fréquences spatiales et temporelles variés. Le modèle a fourni de bons ajustements aux données et a pris en compte, quantitativement, tous les comportements linéaires et non linéaires décrits précédemment.

Réponse aux damiers L´expression de la réponse à un stimulus en forme de damier est donnée par l´équation :

$$amplitude(R) \propto \Big[\frac{amplitude(c_1 L_1(t) + c_2 L_2(t))}{\sqrt{\sigma(f)^2 + c_1^2 + c_2^2}}\Big]^n$$

où :

c_1 et c_2 sont les contrastes des deux grilles,

$L_1(t)$ et $L_2(t)$ sont les réponses sinusoïdales de la fonction poids linéaire aux grilles prises individuellement et avec un contraste unitaire,

les autres symboles ont les mêmes significations que pour les grilles simples.

Puisque le champ récepteur spatio-temporel est linéaire, la réponse au damier est juste la combinaison des réponses aux grilles prises individuellement : $c_1 L_1(t) + c_2 L_2(t)$. Comme pour les grilles simples, l´étape de normalisation divise la réponse linéaire par une quantité qui dépend de l´activité d´un grand nombre d´autres cellules. Cette quantité est : $\sqrt{\sigma(f)^2 + c_1^2 + c_2^2}$. L´étape de rectification est toujours responsable de l´exposant n.

Les prédiction du modèle ont été testées en enregistrant les réponses de cellules simples à des stimuli composés de deux grilles avec des valeurs pour les contrastes c_1 et c_2 très variées. Les deux grilles avaient la même fréquence temporelle mais des fréquence spatiales et des orientations différentes.

Dans le cas où les deux grilles peuvent exciter la cellule (même un minimum) lorsqu´elles sont présentées seules, chaque composante du damier agit à la fois comme un «test» et comme un «masque». Si le masque provoque une réponse, même minime, lorsqu´il est présenté seul, il pourrait très bien engendrer une diminution des réponses au contraste. L´importance de l´interférence entre les différentes composantes d´un stimulus a déjà été prouvée et le modèle normalisé pourrait fournir une base quantitative pour comprendre ces interactions.

4.3.4 Limitations du modèle normalisé

Le modèle normalisé est limité parce qu´il est local en espace. Il n´est pas fait pour rendre compte de la forte inhibition latérale exercée par de nombreuses cellules corticales. Alors que la suppression latérale pourrait, en principe, être le résultat d´un mécanisme identique à celui à l´origine du masquage, il n´est pas sûr que sa nature soit divisible (comme la normalisation du modèle le laisserait entendre). En effet, il a été prouvé que pour le chat, le contrôle du gain par division est fortement sélectif en espace. De plus, certains neurones de l´aire V1 présentent un comportement bien plus compliqué qu´une simple normalisation divisible : pour quelques stimuli ayant une configuration très spécifique, l´introduction d´un stimulus autour du champ peut faciliter la réponse du neurone.

Une autre limitation est que le modèle est aussi local en temps : il ne tient pas compte du phénomène d´adaptation. Toutefois, et dans une certaine mesure, ce phénomène peut être inclut dans le modèle normalisé : il peut être traité comme un masquage en supposant que le contrôle du gain possède une longue mémoire. Cependant, il est peu probable que cette adaptation agisse selon le même mécanisme que celui à l´origine du masquage. En effet, il a été montré que l´adaptation était le résultat d´une hyper-polarisation tonique, ce qui

n´est pas observé durant un masquage. De plus, dans certains cas, l´adaptation ne peut pas être expliqué simplement en changeant le gain d´une cellule. En particulier, après une longue exposition à une grille fortement contrastée, la réponse à cette grille est souvent réduite par rapport aux réponses aux autres grilles, à la fois chez les chats et les singes. Cela pourrait bien être du à des circuits neuronaux supplémentaires provoquant ces phénomènes.

5 Implémentation et expérimentation

Notre développement logiciel est basé sur un logiciel libre qui propose une boite à outil pour la simulation de réseaux de neurones permettant de concevoir à la fois des architectures et des modèles neuronaux variés. Notre ajout à été d'implémenter les neurones «à-la» Heeger et d'expérimenter une architecture représentant les voies visuelles pré-corticales et l'aire corticale V1, sous la restriction d'une vision monoculaire et monochromatique.

L'interface logiciel réalisée permet aussi de générer des stimulus correspondant à ce qui est usuellement utilisé en neurophysiologie pour de futures comparaisons entre cette simulation et des données expérimentales effectives.

L´implémentation s´est faite en Java.

5.1 Choix du réseau de neurones

5.1.1 Choix de l´apprentissage

Le problème de l'apprentissage supervisé ou non supervisé est directement lié au sujet lui-même. Vu le temps imparti, nous avons du nous cantonner dans un apprentissage supervisé. Nous avons donc appris au réseau à détecter des contours. L'intérêt est directement pédagogique : il permet de voir de manière simple, le fonctionnement de l'aire visuelle, sans être perdu par des principes trop compliqués.

5.1.2 Choix de l´architecture

Tout d'abord il apparaît clairement que le perceptron simple est totalement insuffisant. De plus le réseau de Hopfield ne s'adapte pas au problème puisqu'il utilise des données discrètes tandis que notre problème gère des données continues. Donc un réseau en couche du type " perceptron multicouche " semble convenir. Cependant, il ne nous permet pas, dans sa forme classique, d'utiliser des liaisons latérales, présentes dans l'organisation neuronale du cerveau. Par conséquent nous avons abandonné l'idée d'utiliser les différentes couches du perceptron pour rendre compte des activités des couches intermédiaires entre la rétine et le cortex strié au niveau du cerveau.

5.1.3 Choix de l´algorithme d´apprentissage

Il faut donc utiliser un algorithme d'apprentissage adapté. Nous avons choisi le "célèbre" algorithme de rétro-propagation. Ce choix a clairement l'avantage, au niveau pratique, d'avoir déjà été très souvent implémenté. Ainsi avons- nous pu en trouver différentes versions sur Internet.

5.2 Sélection du logiciel de simulation de réseaux de neurones

Voici les différents simulateurs de réseaux de neurones que j'avais à disposition :

Linux Center : Simulateurs de Réseaux de Neurones J'ai beaucoup regardé ce simulateur qui semblait bien au point, cependant, vu qu'il était en C++, je lui ai préféré une version déjà en Java.

Linux Center

Matlab Neural Network Toolbox : Ce site était très bien mais ne fournissait qu'un simulateur en Matlab, et je préférais en récupérer un en java.

Neuron: a simulator of neurons and networks : Il s'en est fallu de peu de choses pour que je choisisse ce simulateur ci. En fait, c'est simplement parce que je l'ai trouvé un peu plus compliqué d'utilisation que le suivant.

OpenAi : C'est ce dernier que j'ai choisi pour son architecture et sa facilité de compréhension. De plus j'ai pu rentrer en contact avec les programmeurs de ce simulateur qui faisaient évoluer de manière continue leur projet. Si bien que j'ai du plusieurs fois changer de version. Ces changements de version m'ont dans un premier temps ennuyée puis il s'est avéré qu'ils étaient bénéfiques car les nouvelles versions contenaient des fonctionnalités dont j'avais besoin et que j'aurais eu à programmer.
Ce groupe est en fait un projet d'intelligence artificielle «open source» qui a pour but de créer des spécifications pour des composants (comme les réseaux de neurones, les algorithmes génétiques, les systèmes experts, etc...) ainsi que de créer des sources libres d'implémentations par défaut de ces composants.

5.3 Solution technique utilisée

5.3.1 Architecture autour du logiciel de simulation de réseaux de neurones sélectionné

Le projet global est divisé en quatre «packages» :

- *Neuralnet :* logiciel de simulation de réseaux de neurones,
- *Image :* la gestion des images à mettre en entrée du réseau,
- *Cells :* la préparation des données d'entraînement d'après le modèle de Heeger,
- *Viewer :* l'interface de visualisation de tous les réglages nécessaires.

Voici la description des classes rajoutées à l'architecture déjà existante du simulateur de réseaux de neurones :

Package Cells : Ce package contient les classes nécessaires à la modélisation de l'activité de cellules simples. Il comprend aussi la répartition des champs récepteurs des cellules sur l'image de la projection rétinienne.

Package Image : Ce «package» gère la donnée d'images et les transforme de sorte qu'elles soient exploitables par le simulateur. Il permet aussi la génération d'images adaptées au problème.

Package Viewer : Ce package s'occupe de la visualisation des interfaces destinées à régler les données d'entraînement et le modèle mathématique de réseau de neurone utilisé par le simulateur.

Les diagrammes de classes sont disponibles en annexe.

5.3.2 Implémentation

Exemple de données d´entrainement

Les stimuli Pour l'entraînement, j'ai utilisé trois types de stimuli : des barres, des grilles et des *«sine gratting»* de différentes orientations.

Pour modéliser les sorties, l'image est découpée en champs récepteurs correspondant aux différentes cellules modélisées. Puis un second découpage décalé d'un demi-champ récepteur est réalisé afin que toutes les parties de l'image soient exploitées.

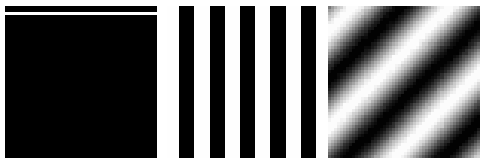

FIG. 29 – *Exemples de stimuli (taille réelle : 32x32) : barre, grille,*«sine gratting».

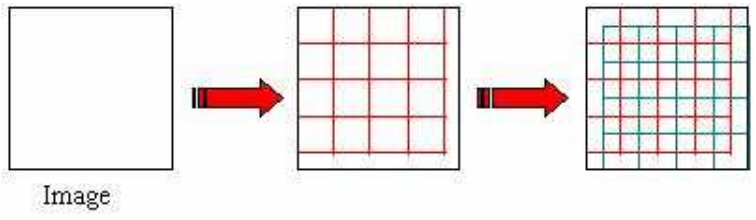

FIG. 30 – *Positionnement des champs récepteurs.*

Si les cellules simples simulées ont toutes des champs récepteurs de taille 7x7 ne détectant que des barres horizontales centrées, alors nous pouvons nous attendre, pour le premier exemple de stimuli par exemple, à ce que seules les cellules de la première couche et sur la première ligne aient une activité. Quand à la grille, il est fort probable qu'aucune cellule n'ait d'activité. En effet, les cellules simples choisies ne détectent que des contours horizontaux.

Simulation des réponses selon le modèle de Heeger Voici les résultats des simulations pour des images de taille 32x32, pour les trois stimuli cités dans les exemples de données d'entraînement, on pourra remarquer qu'ils sont tous trois cohérents avec ce qui est attendu :

Astuce d´entrainement : le «multi-échelle» La taille des données nécessitant beaucoup de calculs, une astuce pour avoir les résultats plus rapidement est le «multi-échelle» à tous les niveaux. Par exemple, nous désirons connaître les résultats pour plusieurs orientations des champs récepteurs (0° et 90°, puis 45° et 135° et enfin 30°, 60°, 120° et 150°). Nous calculons donc les poids pour

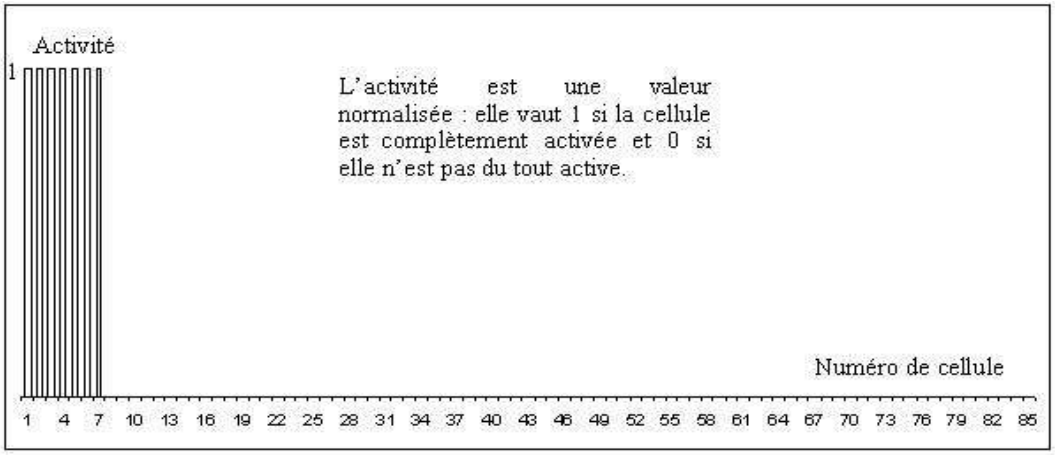

FIG. 31 – *Réponse au stimulus "barre".*

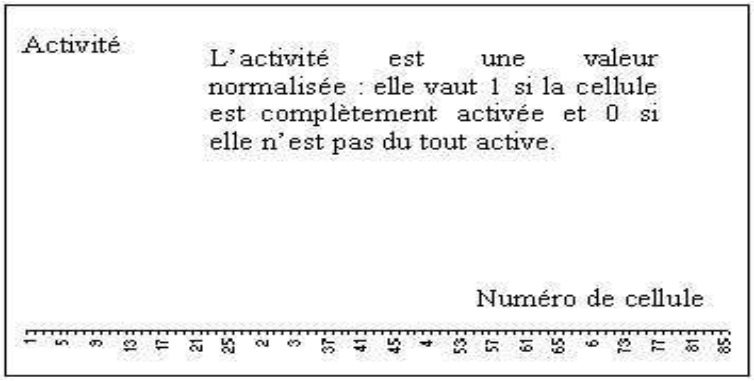

FIG. 32 – *Réponse à une grille.*

des orientations de 0° et 90°, puis nous initialisons les poids pour les calculs des orientations de 45° et 135° en fonction des poids obtenus précédemment. Et de même pour les autres orientations. Nous faisons de même pour les champs récepteurs. Nous faisons les calculs pour des champs de taille 7x7 et nous nous servons des poids obtenus pour initialiser l'apprentissage pour des champs plus petits.

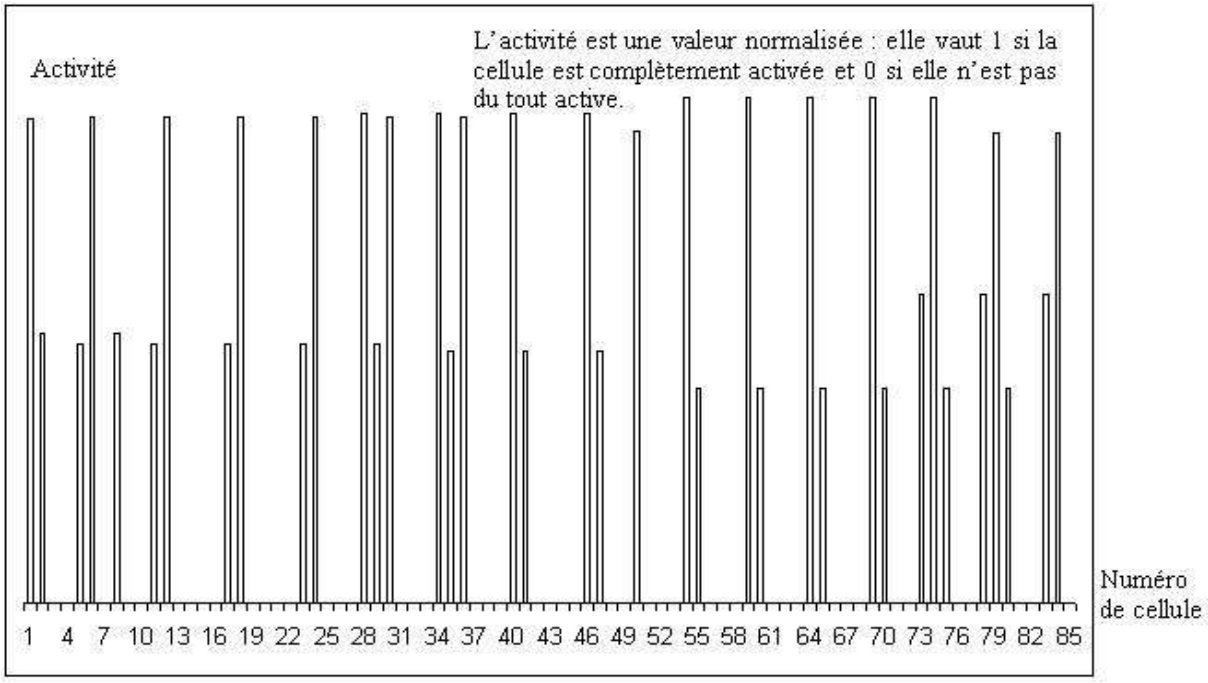

FIG. 33 – *Réponse à un stimulus du type* «sine gratting».

Interface Elle est constituée de plusieurs parties :

- l'accueil : Cette interface permet particulièrement de charger les images pour les tests, d'ouvrir des données d'entraînement et d'accéder aux différents réglages nécessaires au simulateur :
  - Configuration du réseau,
  - Tailles de couches du réseau.
- le réglage des champs récepteurs : avant d'entraîner le réseau il est nécessaire au préalable de régler la taille de l'image d'entrée et des champs récepteurs de toutes cellules dont l'activité est simulée. Il faut préciser aussi le type de champs récepteurs, ainsi que leur orientation et leur répartition sur l'image. Pour l'instant n'est implémentée que la répartition uniforme.

Des captures d´écran de ces interfaces sont exposées en annexe.

Références

[1] F. Alexandre. *Une modélisation fonctionnelle du cortex : la colonne corticale, aspects visuels et moteurs.* PhD thesis, Univ. Henri Poincaré, Nancy I, Apr. 1990.

[2] G. Boyton, S. Engel, G. Glover, and D. Heeger. Linear system analysis of functional magnetic resonance imaging in human v1. *J. Neuroscience*, 16(13):4207–4221, 1996.

[3] D. Hubel. *L'oeil, le cerveau et la vision : les étapes cérébrales du traitement visuel.* L'univers des sciences. Pour la science, 1994.

[4] D. Hubel and T. Wiesel. Functional architecture of macaque monkey. *Lecture Proc. Roy. Soc. London*, pages 1–59, 1977.

[5] M.Carandini, D.J.Heeger, and J.A.Movshon. *Linearity and gain control in V1 simple cells*, chapter 13. New York:Plenum Press, Aug. 1998.

[6] J.-P. Nadal. *Reseaux de neurones, de la physique a la psychologie.* A.Colin, 1993.

[7] Ph.Leray. Quelaues types de reseaux de neurones. Technical report, insa-rouen, 1998.

[8] E. P. Simoncelli and D. Heeger. A model of neuronal responses in visual area mt. *Vision Research*, 38:743–761, 1998.

Remerciements

Nous remercions **F. Alexandre** et **O.D. Faugeras** de l'INRIA pour leurs précieuses idées à l'origine de ce travail.

A Réseaux de neurones : algorithmes constructifs pour un apprentissage supervisé

A.1 «Tilling» (couvreur)

Il s'agit d'un problème de classification à 2 classes.

Au départ :

une base de p exemples : $x^\mu = \{x_1^\mu, x_2^\mu, \ldots, x_N^\mu\}$, $\mu = 1, \ldots, p$,
la classe d'appartenance y_μ pour chacun d'eux.
les x_i^μ sont des nombres quelconques,
tous les neurones du réseau ont une activité binaire : ± 1.

Objectifs :

réaliser un réseau à N unités d'entrées,
une unité de sortie (état 1 pour une classe, -1 pour l'autre),
un certain nombre d'unités cachées entre les deux si nécessaire.

Condition de fidélité : Supposons le réseau achevé avec un certain nombre de couches cachées.
Considérons la couche k, ayant N_k neurones binaires d'activité :

$$y^\mu = \left\{ y_j^\mu, j = 1, \ldots, N_k \right\}$$

lorsqu'en entrée il y a l'exemple x_μ. Ceci et la représentation interne de cette configuration dans la couche k.
Or l'activité d'une couche ne dépend que de la couche précédente. Par conséquent si le réseau donne une bonne classification pour tous les exemples alors deux exemples quelconques de classe différente doivent avoir (dans chaque couche) des représentations internes différentes.

Ainsi, si une couche a cette propriété, alors la représentation interne du problème (pour toutes les couches) est fidèle.

Algorithme :

- On construit un neurone en faisant tourner l'algorithme du perceptron par exemple.

  SI une solution existe alors le réseau est fini.

  SINON :

  - l'algorithme a tourné un certain temps fixé à l'avance. On a donc des couplages définissant un neurone formel : c'est l'*unité maîtresse*.
  - on rajoute d'autres unités (*unités ancillaires*) : ce sera la première couche cachée. Cette couche doit vérifier la condition de fidélité.

- On construit de même la couche suivante avec pour entrées la première couche cachée :
  - on construit l'unité maîtresse de la couche 2. Elle produit E_2 erreurs.
  - on construit la seconde couche comme pour la première couche.
- et ainsi de suite, ...

Preuve de convergence : La condition de fidélité implique l'existence d'une unité maîtresse pour la couche $k+1$ ayant E_{k+1} erreurs telles que $E_{k+1} < E_k$. Ainsi le réseau va croître jusqu'à $E_L = 0$.

Inconvénient : Cet algorithme impose une redondance.

A.2 Arbre de neurones

Algorithme : Comme tilling, dans un premier temps, on cherche à résoudre le problème avec un simple perceptron. Si l'algorithme ne converge pas après un temps déterminé à l'avance :

- on se place dans le demi-espace pour lequel la sortie de ce premier neurone M est positive.

  SI cette partie est in-homogène (i.e. il y a un mélange des deux classes). alors on crée un nouveau neurone ([1]). Ceci implique la création d'un nouvel hyperplan séparant les deux classes dans cette partie.

- on fait de même dans l'autre espace en introduisant si nécessaire un neurone [0],
- ... à la fin, on obtient un arbre dont chaque nœud est un neurone et chaque neurone est connecté aux entrées uniquement.

Pour déterminer la classe d'un exemple :

- on se place dans l'arbre au niveau d'une feuille
- la sortie de chaque neurone indique le suivant à regarder.
- le neurone terminal donne la classe.

Avantages : Cet algorithme se généralise facilement (classification de plus de deux classes, approximation de fonctions, ...).

Inconvénients : L'apprentissage se limite à la croissance du réseau.

B Plausibilité biophysique des modèles de Heeger

Cette partie décrit comment les modèles pourraient être implémentés physiologiquement.

B.1 Modèle linéaire

Linéarité et non-linéarité du corps genouillé latéral Les cellules du corps genouillé latéral peuvent se regroupé en deux classes :

- Les cellules dont la réponse est une fonction linéaire de la distribution d'intensité du stimulus.
- Les cellules dont la réponse n'est pas une fonction linéaire de la distribution d'intensité du stimulus.

Après expérimentations sur le chat, on se rend compte que l'approximation linéaire de cellules du corps genouillé latéral est très imparfaite en ce qui concerne notre sujet. En effet, la majorité des cellules qui interviennent dans la transformation de l'information pour les cellules de V1 sont non linéaires.

De plus la réponse de ces cellules aux forts contrastes met en évidence une rectification (qui est un phénomène non linéaire).

Construction des champs récepteurs des cellules simples Les cellules ayant une sous-région ON et les cellules ayant une sous-région OFF résultent de la somme d´une série de cellules du corps genouillé latéral respectivement centre-ON et centre-OFF alignées.
Si certaines entrées provenant du corps genouillé latéral arrivent avant d´autres, le champ récepteur résultant aurait alors en plus une direction préférée de mouvement (cf. Figure 12).

Arrangements d'entrées «push-pull» La combinaison linéaire d´entrées provenant du corps genouillé latéral implique à la fois des sommes et des soustractions. En effet, les champs récepteurs des cellules simples provoquent non seulement des réponses excitatrices mais aussi des réponses inhibitrices. Hubel et Wiesel (1962) ont montré que l´inhibition pouvait avoir deux causes : un retrait d'excitation ou une véritable inhibition. De même, l'excitation est le résultat d'un retrait d'inhibition ou d'une véritable excitation. Ainsi, une sous-région ON pourrait être aussi bien le résultat de l'excitation d'entrées centre ON que le résultat d'une inhibition d'entrées centre-OFF.
Il serait donc plausible que les entrées d´une cellule simple soient organisées en *«push-pull»* (pousser-tirer), c´est à dire qu´elles iraient par paires de cellules ayant des champs récepteurs de signe opposé, l´une provoquant l´excitation et l´autre l´inhibition. Par exemple, la sous-région ON d'une cellule simple pourrait résulter aussi bien de l´excitation de cellules centre-ON que de l´inhibition de cellules centre-OFF.
Pour des raisons de simplicité, nous considérons que l'excitation comme l'inhibition agissent par des connexions directes («feed-forward»). Bien qu´il existe des preuves récentes en faveur du «feed-forward», le modèle linéaire n´est pas nécessairement définit avec un arrangement d´entrées du type «feed-forward». Un champ récepteur linéaire pourrait, en principe, être construit à partir de connexions feed-forward uniquement, de connexions feed-back uniquement ou d'une combinaison des deux.

Linéarité de l'inhibition et de l'excitation Pour faire le lien, Heeger énonce dans [5] une supposition : «les conductances synaptiques dépendent linéairement du temps». Cette vision est un peu simpliste pour deux raisons :

- La transmission synaptique peut être correctement approximée par une transformation linéaire, mais seulement dans le cas où il n´y a pas de plasticité synaptique. Or celle-ci est quasiment omniprésente et a des effets non-linéaires.
- il existe des preuves comme quoi l´excitation géniculo-corticale serait directe, cependant ceci est contradictoire avec d´autres constatations anatomiques et physiologiques. La plupart des entrées inhibitrices des cellules simples provenant du corps genouillé latéral n´étant pas synaptiques, la linéarité de l´inhibition requiert plutôt une inhibition interneurones qui effectuerait l´intégration linéaire des entrées du corps genouillé latéral puis les encoderait en un potentiel d´action.

Intégration linéaire des entrées synaptiques Dans une cellule simple, l´arrangement «push-pull» des entrées provenant du corps genouillé latéral peut mener à l´intégration parfaitement linéaire des conductances synaptiques par la membrane de la cellule. Il garantit que chaque hausse de l´excitation corresponde à une diminution de l´inhibition et vice-versa.
En 1974, Blomfield a montré que si l´inhibition est localisée dans le soma et que l´excitation en est éloignée électroniquement, il existe une plage d´activations synaptiques dans laquelle le potentiel membranaire pourrait être une combinaison linéaire d´inhibitions et d´excitations. Cette approche ne requiert pas un équilibre strict entre l´inhibition et l´excitation mais nécessite des suppositions supplémentaires au sujet de la structure des dendrites, des lieux d´entrées et de la plage des conductances synaptiques.

Encodage de la fréquence de décharge Il s´agit ici de discuter de la dernière étape —non-linéaire— du modèle qui est responsable de l´encodage du potentiel membranaire en un train d´impulsions.
Beaucoup de caractéristiques de l´encodage de la fréquence de décharge sont cohérentes avec l´idée que la fréquence de décharge est une copie rectifiée du potentiel membranaire. En effet, le train de potentiel d´action l'imite parfaitement à un seuillage près, en dessous duquel aucune décharge n´est générée.

Cependant, des résultats d´expériences dénoncent l´assimilation de l´encodage du potentiel membranaire à une rectification : l´encodeur aurait des propriétés dynamiques. Par exemple, les neurones corticaux montrent souvent une adaptation de la fréquence de décharge, la plupart du temps, il s´agit d´une diminution de celle-ci avec le temps, en réponse à une dépolarisation de longue durée.
En fait, la rectification ne tient pas compte de ce genre de comportement parce qu´elle modélise une non-linéarité *statique*. C´est à dire qu´elle ne tient pas compte de l´historique de la stimulation. C´est pourquoi l´encodage est plus qu´une rectification. En particulier il pourrait être modélisé comme un filtre passe-bande suivi d´une rectification.
Finalement, le modèle comprend la fonction poids linéaire suivit de l´encodeur (passe-bande et rectification). Comme le fitre passe-bande est aussi un filtre linéaire, il est possible de considérer l´ensemble «passe-bande et fonction poids» comme un seul et même système linéaire. Et au bout du compte, la fonction poids du système linéaire global est en partie due aux entrées synaptiques et en partie due aux propriétés passe-bande de l´encodeur de fréquence de décharge.

B.2 Modèle normalisé

Les deux modèles ici présentés proposent deux sortes d´entrées inhibitrices pour les cellules simples :

- une inhibition hyper-polarisée avec les mêmes propriétés de sélections que l´excitation, afin d´assurer la linéarité des réponses des cellules simples,
- une inhibition dérivée *(«shunting inhibition»)* ayant pour origine un grand nombre d´unités corticales est ajoutée à la précédente par le modèle normalisé. Cette inhibition n'est pas sélective et permet de contrôler le gain des cellules.

Les paragraphes suivants décrivent les effets du blocage de l´inhibition dans les cellules des aires V1, la plausibilité d'une inhibition non-sélective et les preuves en faveur d'une hausse de la conductance membranaire dans le cortex visuel.

Effets du blocage de l'inhibition Bloquer l'inhibition d'une seule cellule simple ne modifie pas vraiment son orientation préférée, ce qui est cohérent

avec le modèle normalisé. Par contre ce dernier prédit qu'un tel blocage détruirait sa linéarité sous-jacente et son contrôle du gain, sans toucher à sa sélectivité. Ce qui est normal vu que ni l'inhibition dérivée («shunting inhibition»), ni l'inhibition hyper-polarisée ne sont critiques pour la sélectivité de la cellule modélisée :

- Supprimer l'inhibition dérivée («shunting inhibition») revient à ne garder que la partie linéaire du modèle normalisé, c´est à dire supprime le contrôle du gain. Ce qui est sans effet sur la sélectivité puisqu'elle provient de la fonction poids de la partie linéaire.
- Supprimer l'inhibition hyper-polarisée interfère avec la linéarité de la cellule ce qui devrait influencer la sélectivité de l'orientation. Cependant l'excitation seule permet de distinguer les régions ON et OFF d'un champ récepteur de cellule simple. La déformation de la sélectivité de la cellule n'est remarquable que lorsque l'excitation arrive à la limite de la saturation. Et pourtant, même à ce niveau elle reste très subtile.

Justification de la dérivation de l'inhibition Il y a des preuves comme quoi il existe d´importants circuits d´inhibition dans le cortex et qu'ils opèrent à l'aide du neuro-médiateur GABA par inhibition dérivée («shunting inhibition»).

Le modèle normalisé prédit une augmentation de la conductance de la cellule avec le contraste, mais cette hausse ne semble pas réaliste. En effet, ces estimations des variations de la conductance membranaire sont basées sur des données extra-cellulaires et sûrement exagérée par la supposée linéarité des cellules des corps genouillés latéraux. Or à la fois chez les singes et chez les chats, ces cellules possèdent un contrôle du gain comme les cellules simples. Ajouté à cela, des preuves montrent une transformation active et non linéaire à l'intérieur des dendrites. Tenir compte de ces phénomènes permettrait au modèle de prédire des augmentations du potentiel membranaire plus réalistes.

Des expériences intracellulaires ont montré un phénomène un peu en désaccord avec le modèle normalisé : elles ont montré que la conductance membranaire dépendait autant de l'orientation du stimulus que du contraste, or le modèle prédit qu'elle ne dépend pas du contraste.

C L´implémentation du projet

C.1 Diagramme des classes

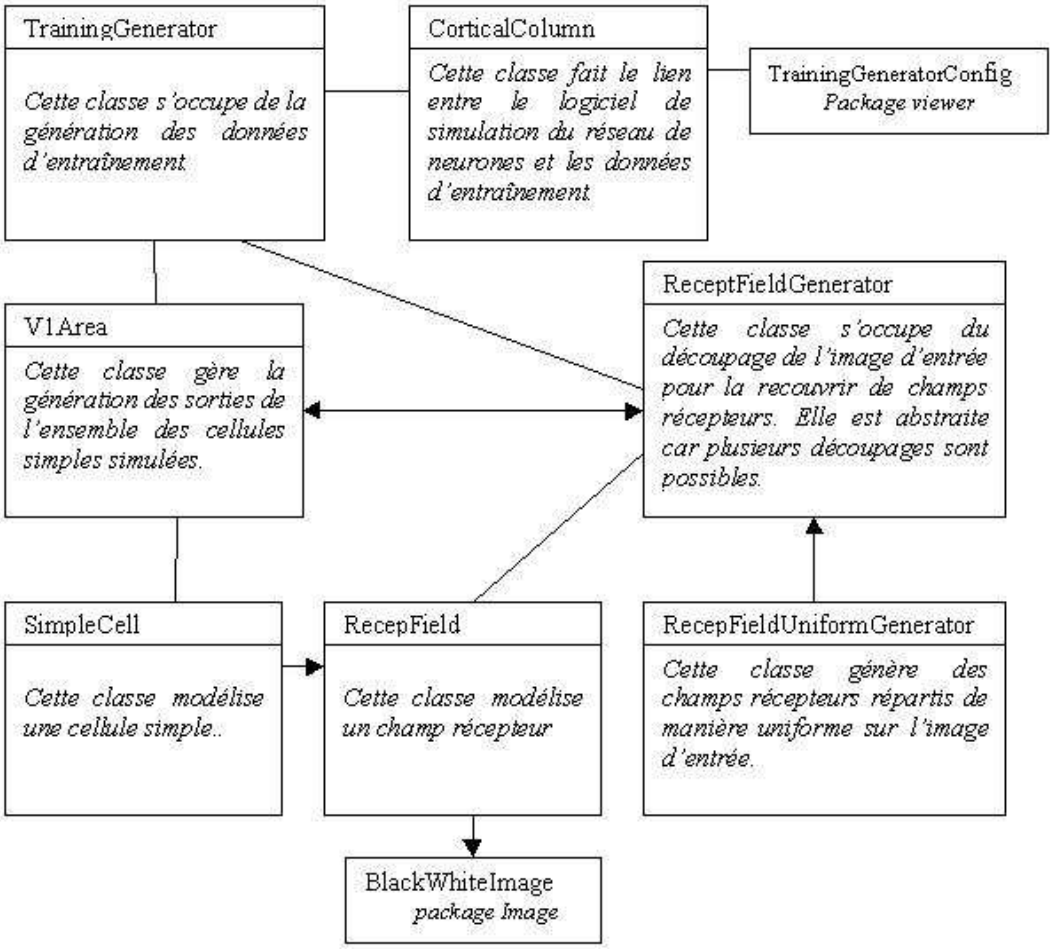

FIG. 34 – *Diagramme des classes du «package» Cells.*

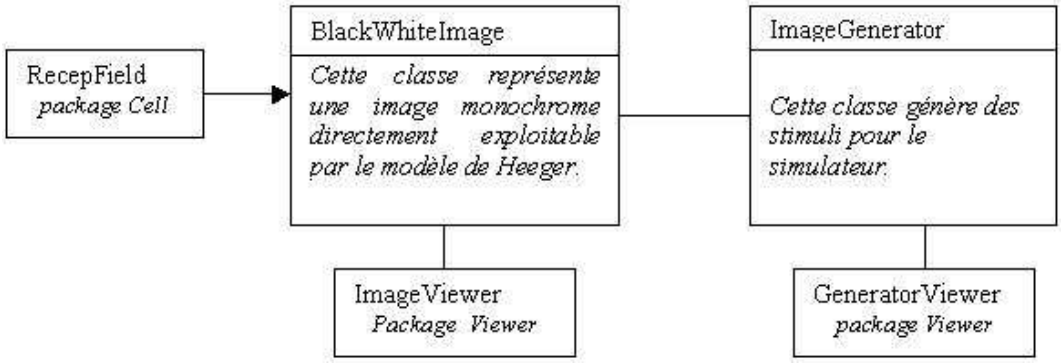

FIG. 35 – *Diagramme des classes du «package» Image.*

C.2 Interfaces du logiciel de simulation

Ces interfaces ont été créées à partir de la libraire *Imp* (Interface Manipulation Package) du projet Odyssée.

C.3 Evaluation du du logiciel de simulation d´un réseau de neurones

Voici un exemple d'apprentissage réalise sur le principe du Xor avec le simulateur de réseaux de neurones récupéré.

Le Xor est un " Ou " exclusif.

Le principe du Xor est un problème qui typiquement n'est pas réalisable par un perceptron simple. Par contre il est réalisable par un perceptron multicouche.

Nous avons testé le logiciel de simulation de réseaux de neurones avec un Xor. Nous avons choisi de mettre une seule couche intermédiaire mais comportant trois neurones dans cette couche. La couche d'entrée possède deux neurones puisqu'il y a deux entrées et la couche de sortie en contient un seul.

Voici la base d'apprentissage utilisée :

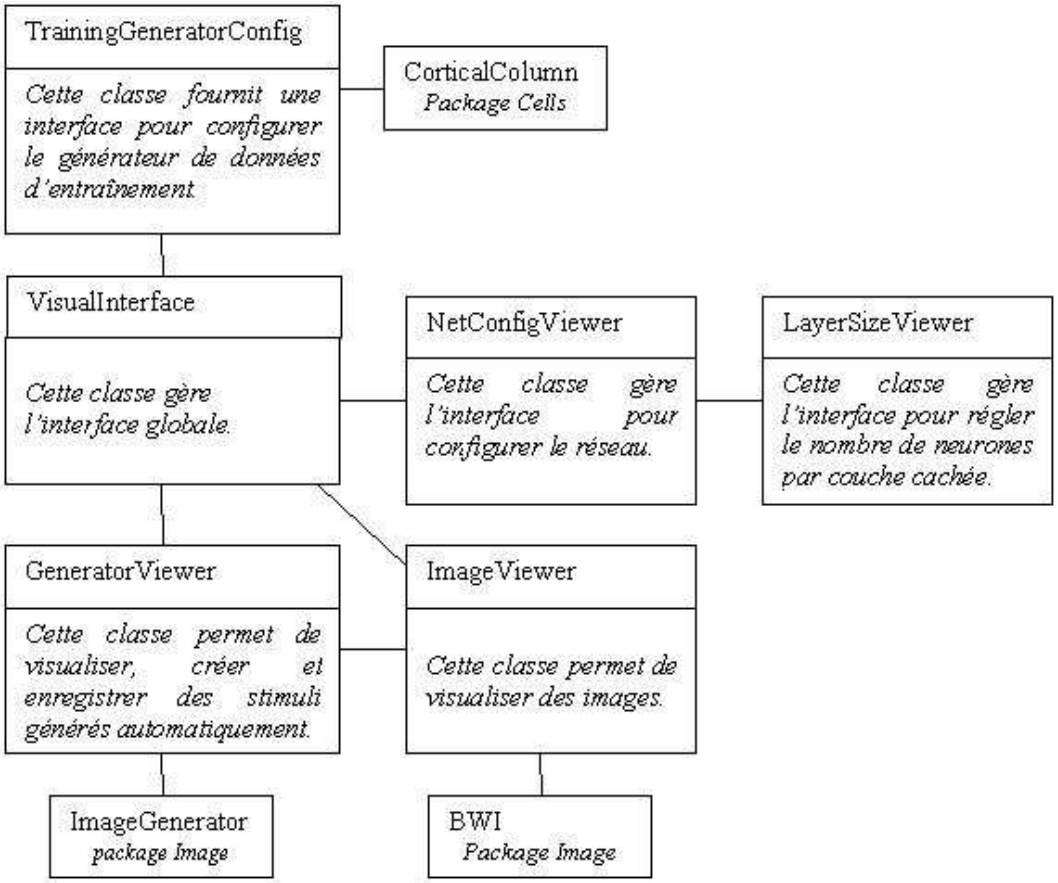

FIG. 36 – *Diagramme des classes du «package» Viewer.*

Entrée 1	Entrée2	Sortie
0,9	0,1	0,9
0,1	0,9	0,9
0,1	0,1	0,9
0,9	0,9	0,1

Après 2000 itérations, l'erreur obtenue est de l'ordre de 0,0099. Ce qui est tout à fait appréciable.

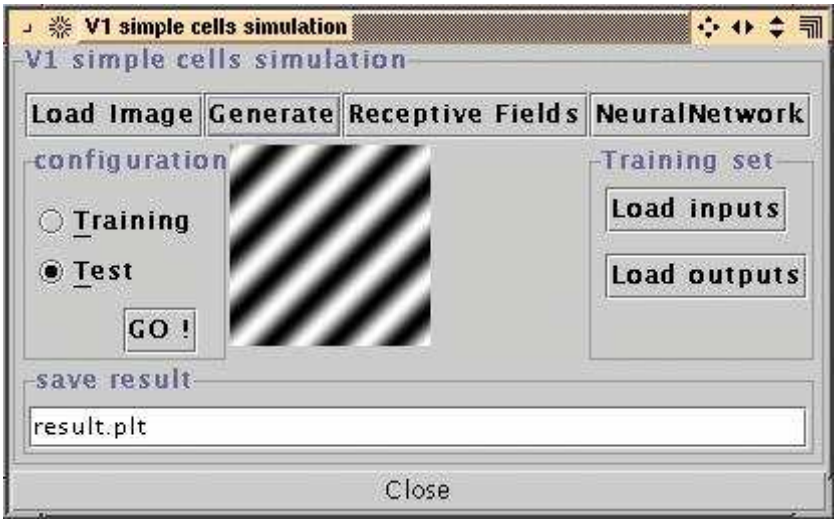

FIG. 37 – *Fenêtre principale.* Cette fenêtre propose un menu (en haut) permettant d´accéder aux différentes fenêtres de réglage des paramètres du réseau. Du plus elle permet de choisir le mode de fonctionnement : entraînement ou test, ainsi que de charger une base d´entraînement. Pour finir elle permet de visualiser les stimuli proposés et de donner un nom au fichier de sauvegarde des poids du réseau (cette partie n´a pas encore été complètement implémentée lors du projet).

Nous avons alors testé un couple de d'entrées avec ce réseau entraîne. Ainsi une entrée (1,0) a donne comme résultat : 0,8635.

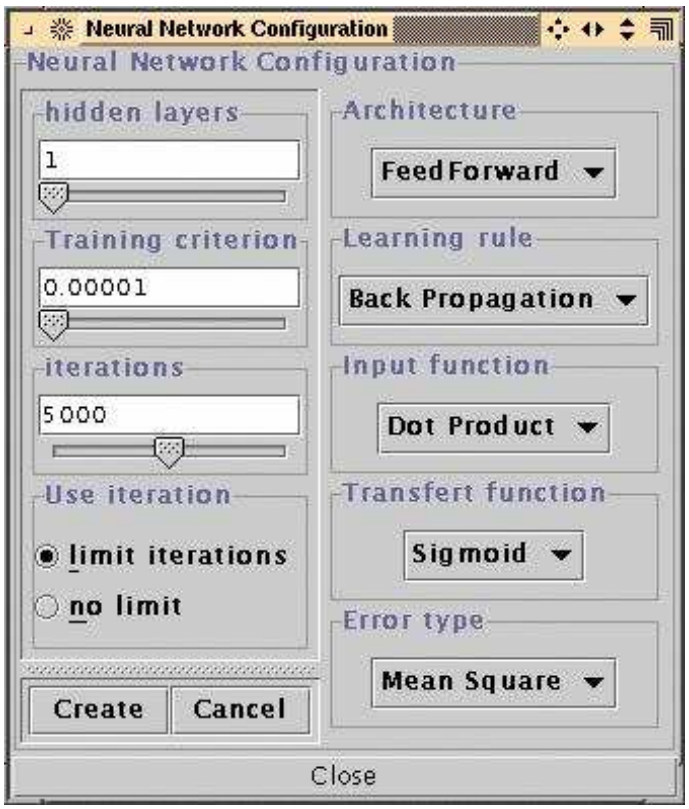

FIG. 38 – *Réglage de la configuration du réseau.* Cette fenêtre permet de configurer le réseau de neurone utilisé, afin de choisir plus facilement la configuration la plus adaptée au problème. Tout d´abord on peut y saisir le nombre de couches cachées, on peut choisir la condition d´arrêt des itérations : un nombre fixé d´itérations ou une erreur minimale. L´interface permet aussi de choisir le type d´architecture, ainsi que l´algorithme d´apprentissage, la fonction de transfert utilisée pour modéliser la réponse des neurones, le mode de calcul des erreurs, etc ...

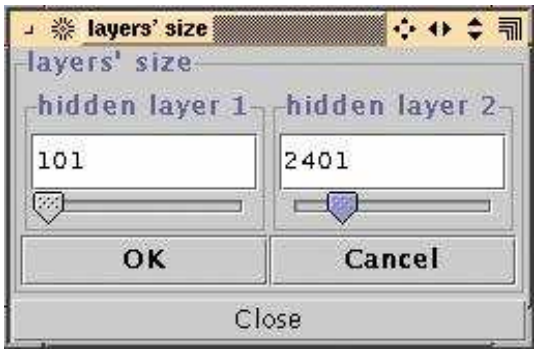

FIG. 39 – *Réglage des différentes couches de neurones.* Cette interface permet de donner le nombre de neurones contenus dans chaque couche du réseau.

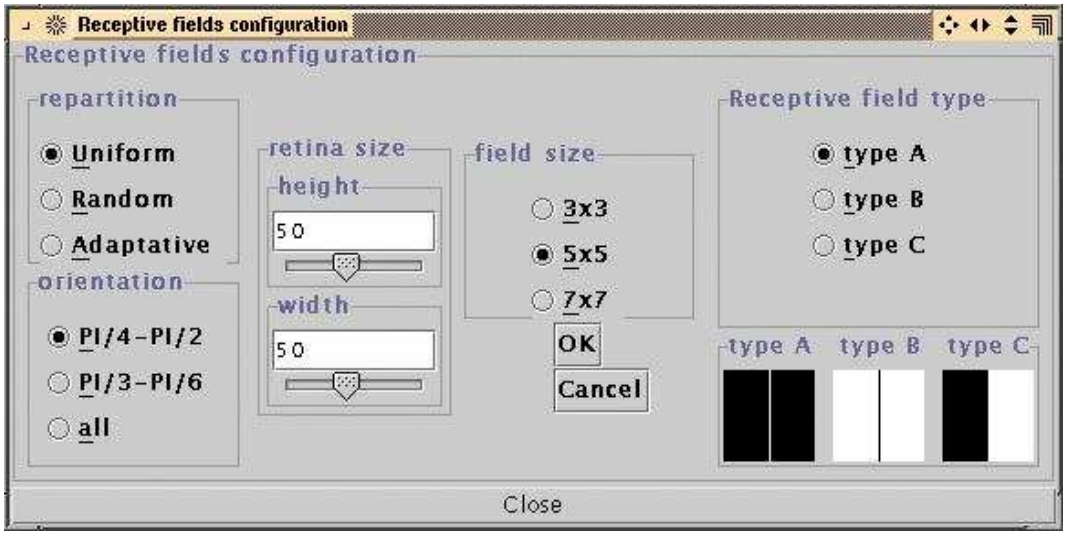

FIG. 40 – *Fenêtre de réglage des champs récepteurs.* Cette interface permet de configurer les champs récepteurs utilisés par rapport au modèle de Heeger : la taille, le type et l´orientation. Elle permet aussi de configurer la taille de la rétine (donc de la couche d´entrée du réseau) et la répartition des champs sur la rétine (ce qui donne le nombre de neurones sur la couche de sortie du réseau).

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399