

side

# Rapport projet long uipe HAL9000

Hamelain Christian, Hasan Pierre-Yves, Gaspart Quentin, Trestour Fabien

27 mai 2016

# Table des matières

<b>I</b>	<b>Introduction</b>	<b>3</b>
0.1	Contexte . . . . .	4
0.2	Prntation du sujet . . . . .	4
0.3	Dulement . . . . .	4
<b>II</b>	<b>Le Perceptron</b>	<b>5</b>
<b>1</b>	<b>Architecture d'un perceptron</b>	<b>6</b>
1.1	Introduction . . . . .	6
1.2	La brique de base : le neurone . . . . .	6
1.3	Organisation des couches . . . . .	7
<b>2</b>	<b>Algorithmes d'apprentissage</b>	<b>10</b>
2.1	salut . . . . .	10
2.2	c'est cool . . . . .	10
<b>3</b>	<b>La base MNIST</b>	<b>11</b>
<b>4</b>	<b>Influence des parames sur l'apprentissage</b>	<b>12</b>
<b>III</b>	<b>Les raux de Boltzman</b>	<b>13</b>
4.1	Les machines de Boltzmann restreintes . . . . .	14
<b>IV</b>	<b>Restricted Boltzmann Machine</b>	<b>15</b>

Première partie

Introduction

## 0.1 Contexte du projet

CentraleSupc propose es diants en deuxi anne participer es projets longs qui se risent en ipe, tout au long de l'annur un th mnt plusieurs compnces que l'ingeur se doit de possr.

Dans notre cas, c'est l'boration de raux neuronaux et les modes de Deep Learning que 12 ves ont dirtis en 3 groupes, dont HAL9000.

Ce rapport prnte les avancement de ce dernier groupe sur le sujet tout au long de l'ann

## 0.2 Prntation du sujet

Ce projet aborde le sujet du Deep Learning. Cette mode spfique de machine learning est dvu concept de rau de neurones.

Les raux de neurone sont une modsation simple du fonctionnement cbral 'elle cellulaire. Cette approche a vu le jour avec les des de McCulloch et Pitts da fin des ann 50. Malgrrtains eils dans les ann 70, l'approche connexioniste des raux de neurones a su se dlopper et devenir un sujet de recherche populaire dans les dernis ann, notamment gre aux capacit'adaptabilit de gralisation de ces raux qui en font d'excellents candidats pour des applications telles que la reconnaissance d'image ou la classification.

## 0.3 Dulement du projet

Tout d'abord, il s'agissait pour nous de comprendre le fonctionnement de ces structures et commencer oder des structures mentaires afin de comprendre les enjeux du machine learning. Nous avons pour cela travaillec la base de donn MNIST de Yves LeCun et en JAVA. Afin de travailler en groupe de mani efficace, nous avons aussi utilisoutil GIT.

Par la suite nous nous sommes interessux architectures profondes, chaque groupe se penchant sur une architecture spfique. Notre ipe s'est en particulier intssux machines de Boltzmann, en commennt par les machines de Boltzmann restreintes (RBM), puis la structure de Deep Learning associ

Cette de a encadrar Joanna Tomasik et Arpad Rimmel, enseignants entra-leSupc, campus de Gif-sur-Yvette.

Deuxième partie

Première approche du problème : le  
Perceptron

# Chapitre 1

## Architecture d'un perceptron

### 1.1 Introduction

Ici, l'objectif est la reconnaissance des caractères de la base de données MNIST grâce au perceptron.

Le perceptron fait partie des architectures de réseaux neuronaux les plus simples. Son développement nous a donc permis de nous introduire à la problématique du machine learning avant d'approfondir en étudiant des structures plus complexes.

### 1.2 La brique de base : le neurone

Le neurone est le composant élémentaire des réseaux neuronaux. Il est une modélisation du fonctionnement des neurones du système nerveux humains.

Chaque neurone reçoit un signal via une entrée qui correspond aux dendrites des systèmes biologiques. Le neurone prend en compte la valeur de toutes ses entrées et en déduit la valeur de sortie. Cette sortie est ensuite propagée le long d'un axone vers un autre neurone.

La sortie du  $j^{\text{ième}}$  neurone est donnée par la formule :

$$s_j(x) = f\left(\sum_{k=1}^N w_{j,k} * x_k + w_0\right) \quad (1.1)$$

où

- $s$  est la valeur de la sortie
- $f$  est la fonction d'activation.
- $N$  est la dimension du vecteur d'entrée
- $w_{j,k}$  est la  $k^{\text{ième}}$  composante du vecteur de poids  $w_j$  du  $j^{\text{ième}}$  neurone.  $w_0$  est le biais du neurone. Cette valeur correspond au poids d'une entrée constante valant toujours 1.
- $x_k$  est la  $k^{\text{ième}}$  composante du vecteur d'entrée.

L'entrée est un vecteur défini par un ensemble de caractéristiques que l'on choisit pour représenter les données d'entraînement.

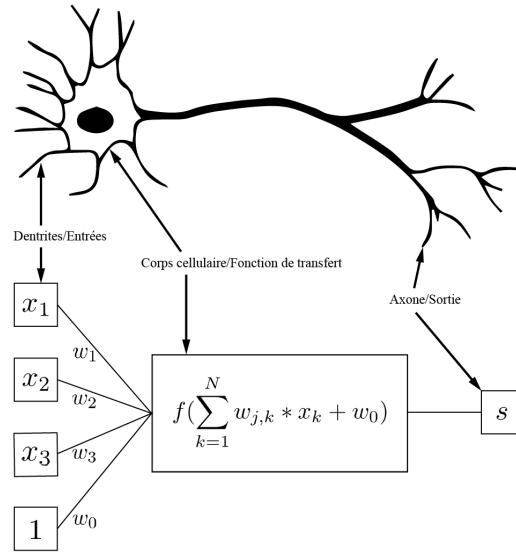


FIGURE 1.1 – Analogie entre neurone biologique et neurone formel.

La fonction d'activation permet de définir le comportement du neurone. Selon la définition de cette fonction on aura une sortie à valeurs discrètes ou continues, centrée en 0 ou en 0,5. Le choix de la fonction est donc étroitement lié au problème à traiter. Il existe différents types de fonctions d'activation. Parmi les plus utilisés figurent :

— La fonction élon

$$f(x) = \mathbf{1}_{\mathbb{R}_+^*} \quad (1.2)$$

— Les fonctions linéaires

$$f(x) = \alpha * x + \beta \quad (1.3)$$

— La fonction sigmoïde

$$f(x) = \frac{1}{1 + e^{-\lambda * x}} \quad (1.4)$$

— La fonction tangente hyperbolique

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (1.5)$$

Le vecteur de poids du  $j^{\text{ième}}$  neurone représente la pondération de chacune des entrées de ce neurone. C'est ce paramètre qui permet de modifier le comportement du neurone sans avoir à modifier son architecture. Toutes les propriétés d'un réseau neuronal sont donc issues de ce vecteur de pondération et de ses variations.

### 1.3 Organisation des couches

Il existe de nombreuses architectures de réseaux de neurones. Elles peuvent être récurrentes ou non, entièrement connectées ou seulement partiellement, organisées en couches, ... De nombreuses propriétés permettent de caractériser une architecture de réseau de neurones.



Le Perceptron est un mod assez mentaire de rau. Il est constitu couches totalement connect.

On peut distinguer trois types de couches : la couche d'entr les couches cach et la couche de sortie.

La couche d'entrst assez mentaire. C'est tout simplement une couche dont la valeur sera le vecteur d'entrx. *Cette couche doit donc constituer autant de neurones que le vecteur d'entr dedin*

Les couches cach sont celles qui effectuent les calculs. Les principales propri du rau sont ht de cet empilement de couches. On comprend alors mieux l'intt actuel pour le Deep Learning, c'est-dire l'apprentissage des raux avec un grand nombre de couches cach. Le nombre de neurones dans chaque couche et les fonctions d'activation utilis par les neurones sont choisies par le concepteur du rau selon le probl traitr le rau.

Afin d'expliciter l'importance de l'architecture du rau, abordons l'exemple usuel du ou exclusif.

Essayons de riser avec un unique neurone la fonction logique XOR. Ce neurone aura deux entr  $x_1$  et  $x_2$  aleurs dans  $\{0;1\}$  et une sortie  $s$ , elle aussi aleurs dans  $\{0;1\}$ . On prendra comme fonction d'activation la fonction de Heaviside, c'est ire  $\mathbf{1}_{\mathbb{R}_+^*}$ . Ce neurone aura par consent un fonctionnement totalement binaire. Il s'agit donc ici de drminer les pondtions  $w_1$  et  $w_2$ , respectivement associ aux entr  $x_1$  et  $x_2$ , qui conviennent pour obtenir en sortie la valeur  $x_1 \oplus x_2$ .

Pour un tel probl, le neurone agit comme un srateur linre de l'espace des entr. L'ation de la droite srant le demi-espace dnit par  $s = 0$  de celui dnit par  $s = 1$ , dule alors directement de l'ation 1.1 :

$$w_2 * x_2 + w_1 * x_1 + w_0 = 0 \quad (1.6)$$

On constate ici l'intt du biais, qui permet de riser des srations affines de l'espace des entr et pas seulement des fonctions linres.

Les limites d'un neurone seul sont alors dentes : toutes les srations de l'espace des entr ne sont pas affine, et le cas du XOR en est d une illustration.

Il est alors nssaire d'introduire la structure de rau. En prennant une couche d'entr une couche cacht une couche de sortie, respectivement compos de deux, deux et un neurone, on peut contourner le probl sus-citette structure permet en fait de grer deux srations de l'espace des entr.

La sration que l'on cherche ffectuer est en effet de la forme :

$$f(x_1, x_2) = 0 \Leftrightarrow \begin{cases} x_1 + x_2 - 0,5 < 0 \\ x_1 + x_2 - 1,5 > 0 \end{cases} \quad (1.7)$$

On obtient ainsi directement les valeurs de poids dr gre es ations :

$$blablabla \quad (1.8)$$

Ce premier exemple simpliste manifeste donc la nssitadapter la structure du rau au probl trait

De mani plus grale, le nombres de neurones du rau fait varier le nombre de poids du rau, et donc la capacit rau rer des ensembles multiples et com-

plexes. Ainsi, un *rau* sous-dimensionné est rltats pas assez *prs* et un *rau* sur-dimensionné u sur-apprentissage. On a donc environ la loi suivante :

$$N < \frac{1}{10} * T * \dim(s) \quad (1.9)$$

- $N$  est le nombre de neurones dans le *rau*.
- $T$  est le nombre de vecteurs de la base d'apprentissage.
- $s$  est le vecteur de sortie.

De plus, le nombre de neurones dans une couche doit e de moins de trois fois celui de la couche *prdente*.

Ces deux approximations donnent donc une premi ide la structure dopter pour un *rau* de neurones.

## Chapitre 2

# Algorithmes d'apprentissage

### 2.1 salut

123

### 2.2 c'est cool

## Chapitre 3

### La base MNIST

## Chapitre 4

# Influence des parames sur l'apprentissage

Troisième partie

Les taux de Boltzman

Les raux de Boltzmann sont des raux qui ont un fonctionnement particulier que nous allons prnter dans ce chapitre. Les machines de Boltzmann constituent des raux neuronnaux. La prinipale diffnce entre les raux neuronnaux comme les raux neuronnaux de convolution (CNN) et les raux de Boltzmann va e le fait qu'un raux de Boltzmann n'est pas *feed-forward* ; comme l'it le perceptron, prntns ce rapport.

Ce rapport traitera d'abord des machines de Boltzmann restreintes en les prntant et en expliquant les lois qui rssent leur fonctionnement pour ensuite s'intsser aux Raux de Boltzmann Profond.

## 4.1 Les machines de Boltzmann restreintes

Les machines de Boltzmann restreintes, ou RBM, ont d'abord t pensees par P. Smolensky (1986) mais rlement mise en oeuvres et di par G. Hinton(2006). Nous allons prnter dans cette partie leur constitution et leurs fonctionnement.

### 4.1.1 Principe et thie

Les RBM sont des raux neuronnaux et sont donc constitu de d'entite l'on nommera neurone qui poss une fonction d'activation et de synapse qui relie ces neurones.

Une RBM poss deux couches de neurones, une que l'on appellera couche visible et une autre que l'on appellera couche cach La couche visible correspond entre notre rau, ce sont les neurones de la couches d'entruxquels ont va assigner les exemples de notre jeu de donn pour entrainer le rau. Au sein d'une m couche, aucun neurone n'est relin autre, c'est ce caract restreint qui donne sa particularit son nom type de rau de Boltzmann.

Quatrième partie

Approfondir le sujet :  
Restricted Boltzmann  
Machine