



UNIVERSITE D'ANTANANARIVO

ECOLE SUPERIEURE POLYTECHNIQUE

Mention : Electronique

Parcours : Sciences Cognitives



Mémoire en vue de l'obtention du Diplôme de Master II à Visée de Recherche

INTELLIGENCE ARTIFICIELLE



# Ordonnancement automatique des phrases Malagasy et leurs traductions vocales Françaises

Présenté ce Mardi 26 Janvier 2016 par

BOURGEON ANDY MARLON

Jury

**Président :** Professeur ANDRIAMANOHISOA Hezy Zo, Enseignant à l'ESPA

**Juges :** Professeur RANDIMBINDRAINIBE Falimanana, Enseignant à l'ESPA

Docteur ROBINSON Matio, Enseignant à l'ESPA

Docteur RATSIHOARANA Constant, Enseignant à l'ESPA

**Directeur de mémoire :** Professeur ANDRIANAHARISON Yvon, Enseignant à l'ESPA

## **REMERCIEMENTS**

Ma première pensée s'adresse à notre **Dieu TOUT PUISSANT** sans quoi ce mémoire n'aurait pas été effectué. Je remercie ensuite plus particulièrement :

Professeur RATSIMBA Mamy notre Chef de Mention.

Professeur ANDRIAMANOHISOA Hery Zo, de bien vouloir présider ce mémoire.

Professeur ANDRIANAHAHARISON Yvon, mon Directeur de mémoire, de consacrer de son temps pour me guider tout au long de la réalisation de ce mémoire.

Professeur RANDIMBINDRAINIBE Falimanana, de bien vouloir examiner ce mémoire.

Docteur ROBINSON Matio, de me céder aussi de son temps précieux pour examiner ce travail.

Docteur RATSIHOARANA Constant, de m'accorder de son temps précieux pour examiner ce mémoire.

Ma famille et mes amis pour leur soutien et encouragement durant cette année de travail pour l'obtention du Diplôme de Master 2 à Visée de Recherche en Sciences Cognitives ainsi que la suite des Recherches que poursuivrai dans cette lignée.

## TABLE DES MATIERES

REMERCIEMENTS

TABLE DES MATIERES

LISTE DES FIGURES

LISTE DES ABREVIATIONS

INTRODUCTION.....1

Chapitre 1 – GENERALITES.....2

    1.1-SCIENCES COGNITIVES.....2

    1.2-LE LANGAGE HUMAIN.....3

    1.3- LA DIFFERENCE ENTRE UN SIGNE ET UN  
    SIGNAL.....5

    1.4- LA DOUBLE ARTICULATION.....5

    1.5-LE CERVEAU HUMAIN : UN PROCESSEUR BIOLOGIQUE.....6

    1.6-METHODE DE RAISONNEMENT DU CERVEAU.....8

    1.7-QUELQUES REGLES DE LANGAGE.....10

    1.8-LA LANGUE MALAGASY.....11

    1.9-SITUATION DES CHOSES.....13

    1.10-LES RESEAUX DE NEURONES ARTIFICIELS.....13

Chapitre 2 - ETAT DE L'ART.....23

INTRODUCTION.....23

    2.1-LE PERCEPTRON.....23

    2.2-LES RESEAUX MULTICOUCHES.....30

    2.3-PRESENTATION DE NOTRE CAS.....35

    2.4-TRADUCTION VOCALE AVEC Google TTS (*Text To Speech*).....39

Chapitre 3 – IMPLEMENTATION.....41

    3.1-PRESENTATION GENERALE DE LA TECHNOLOGIE OBILE.....41

    3.2-ANDROID ET SES CAPACITES.....42

AMELIORATIONS.....51

CONCLUSION.....52

BIBLIOGRAPHIE.....i

WEBOGRAPHIE.....ii

ANNEXES.....iii, iv,v,vi,vii,viii

RESUME.....ix

ABSTRACT.....ix

## **LISTE DES FIGURES**

Figure 1.01 : Présentation du langage humain.....	4
Figure 1.02 : Les niveaux d'articulations.....	5
Figure 1.03 : Anatomie générale du cerveau humain.....	7
Figure 1.04 : Structure générale d'un neurone.....	8
Figure 1.05 : Schéma d'un neurone.....	9
Figure 1.06 : Forme générale d'un réseau de neurone artificiel.....	13
Figure 1.07 : Schéma synoptique d'un Perceptron.....	15
Figure 1.08 : Perceptron simple.....	16
Figure 1.09 : Comparaison entre un neurone biologique et artificiel.....	16
Figure 1.10 : Structure d'un réseau avec une entrée, un poids et une sortie.....	17
Figure 1.11: Perceptron simple.....	19
Figure 1.12 : Présentation d'un réseau de Hopfield.....	20
Figure 1.13 : Réseau de Kohonen.....	22
Figure 2.01 : Perceptron élémentaire.....	24
Figure 2.02 : Neurone de sortie du Perceptron.....	24
Figure 2.03 : Fonctionnement d'un neurone de sortie du Perceptron.....	25
Figure 2.04 : Apprentissage du Perceptron.....	26
Figure 2.05 : Apprentissage du Perceptron.....	28
Figure 2.06 : BugDroid, le robot logo d'Android.....	40
Figure 3.01 : Logo d'Android.....	43
Figure 3.02 : Schéma général de transfert de l'application vers Android.....	44
Figure 3.03 : Présentation de l'interface d'Eclipse ADT.....	45
Figure 3.04: Présentation de l'interface de l'émulateur.....	46
Figure 3.05 : Présentation de l'interface de l'application.....	50

## **LISTE DES ABREVIATIONS**

*IA : Intelligence Artificielle*

*AVD : Android Virtual Device*

*ADT : Android Developper Tools*

*OS : Operating System*

## INTRODUCTION

Les sciences cognitives sont un ensemble de disciplines scientifiques visant à l'étude et la compréhension des mécanismes de la pensée humaine, animale ou artificielle, et plus généralement de tout système cognitif, c'est-à-dire tout système complexe de traitement de l'information capable d'acquérir, conserver, et transmettre des connaissances. De nature pluridisciplinaire, elles étudient un grand nombre de domaines allant de l'informatique, à la psychologie, en passant par la biologie et les sciences du langage.

Les sciences du langage sont l'étude du langage et des langues, qu'elles soient naturelles ou artificielles. Elles regroupent tous les aspects de la linguistique (morphologie, phonologie, prosodie, syntaxe, sémantique, pragmatique) et ses dérivés comme la sociolinguistique, l'ethnolinguistique, l'économie linguistique, etc. ainsi que le T.A.L. (Traitement Automatique du Langage) qui est la partie des sciences du langage qui est en interaction avec l'informatique et l'I.A. (Intelligence Artificielle).

Les domaines de philosophie les plus étudiés en sciences cognitives sont la philosophie du langage, la philosophie de l'esprit, la philosophie de l'action et ses dérivés, et bien entendu la philosophie cognitive. Ces domaines relèvent de manière générale de la philosophie analytique et utilisent régulièrement la logique. Un domaine essentiel est également la partie de la philosophie en rapport avec les sciences, à savoir l'épistémologie/philosophie des sciences, ainsi que l'éthique qui est indispensable à la science.

L'intelligence artificielle est la science qui permet aux machines de faire des tâches qui auraient demandé de l'intelligence si elles avaient été faites par des hommes. Cette sous-discipline de l'informatique étudie entre autres la cognition et l'apprentissage automatique et fournit des algorithmes qui seront implémentés sur des ordinateurs. Le but de ce département est de favoriser l'apprentissage de cette discipline.

Nous allons donc à partir de toutes ces définitions réaliser une application basée sur le langage « Malagasy » qui va ajuster correctement les phrases entrées en désordre puis énoncer vocalement sa traduction française.

Ceci est une avant première à Madagascar car c'est la première application pour perfectionner les langues Malagasy et Française sous Android combinée avec l'intelligence artificielle.

# **CHAPITRE 1**

## **GENERALITES**



# Chapitre 1 - GENERALITES

## 1.2-SCIENCES COGNITIVES

Les sciences cognitives s'intéressent aux *grandes fonctions mentales* comme la perception, la mémoire, le raisonnement, le langage, les émotions, la motricité, la communication, etc.

Elles ont pour but l'étude conjointe des propriétés formelles et algorithmiques de ces fonctions mentales, des mécanismes psychologiques qui les sous-tendent, et des mécanismes biologiques qui les rendent possibles (des gènes jusqu'aux circuits et aires dans le système nerveux).

Elles s'intéressent aussi à leurs équivalents ou précurseurs chez l'animal, à l'influence des variations culturelles sur leur fonctionnement chez l'homme, à leur développement chez le nourrisson ou le jeune enfant, à leur altération dans certaines pathologies neurologiques, psychiatriques ou développementales. Elles cherchent plus généralement à saisir les rapports complexes qui se nouent entre les compétences de l'espèce et la culture au sens le plus large.[Référence n°23]

La communication est l'une des clés de la réussite. Certes, communiquer c'est exprimer ses idées, son avis, définir quelques choses et surtout échanger des informations dans le but d'améliorer son existence, notre environnement physique et social à travers les langages, gestes et expressions en tout genre qui peuvent être imaginées, écrites, orales.

Le langage humain fait partie de cette communication. Ce langage humain est le plus utilisé sur la planète car dès notre enfance, nous apprenons petit à petit à parler, écrire et maîtriser ce langage afin de pouvoir communiquer avec aisance dans la société.

Remarquons que généralement chaque pays a sa propre langue pour se communiquer entre eux donc des traductions sont nécessaires pour que deux langues différentes se comprennent.

Le dictionnaire est un bon outil pour perfectionner le niveau d'autrui car tous les vocabulaires, mots, noms, verbes et bien d'autres choses y figurent.

L'expression orale est acquise presque par tout le monde et est compréhensible par tout un chacun respectant naturellement des règles de prononciations, d'élaboration de mots divers bien structuré afin d'exprimer quelque chose.

Pour ce qui est de l'écrit, la grammaire s'impose. En effet, des normes sont à respecter suivies d'orthographies correctes exigées. La structure d'une phrase doit être établie suivant des règles. Or peu de personnes maîtrisent cet art qui exige travail et persévérance.

La numérisation n'est plus chose nouvelle pour nous. Toute chose tend à être développée par la technologie. Chaque domaine est maintenant informatisé afin d'améliorer leurs performances, précisions, facilité d'utilisation ainsi que les sauvegardes des données au format numérique. D'autres alternatives existent aussi comme l'internet qui est un puissant outil incontournable de nos jours.

De ce fait, l'apparition de la technologie mobile vient renforcer la chaîne. Le numérique est presque partout et maintenant cette mobilité tient de plus en plus de place dans notre vie.

Nous devons donc nous habituer dans cette voie qui n'est pas prête de s'arrêter demain.

Côté mobile citons Android, iPhone, Windows Mobile, BlackBerry qui sont les plus répandus dans le marché actuel. Mais il existe aussi d'autres systèmes comme Firefox Os, Symbian.

Ce mémoire va alors essayer de combiner le langage humain (langue Malagasy) et ses règles grammaticales pour une structure correcte d'une phrase combinée avec la nouvelle technologie mobile pour offrir à chacun un outil efficace, rapide et convivial pour vite maîtriser cette langue qui est notre langue maternelle.

Cette application sera une avant première car elle propulsera notre chère langue bien aimée loin devant les autres langues car l'Intelligence Artificielle sera au rendez-vous pour mettre au point sa réalisation avec des traductions en diverses langues offertes par Google TTS (Text To Speech).

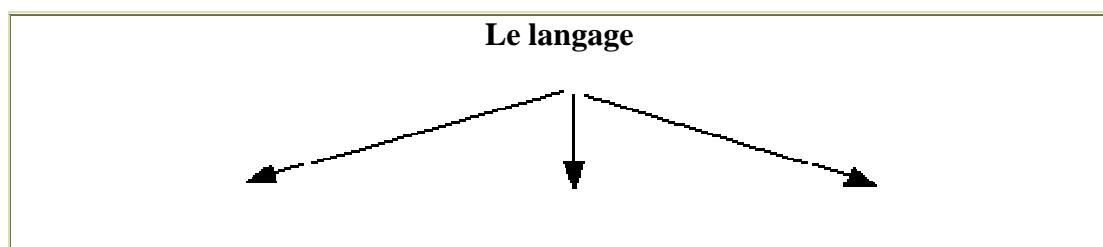
### **1.2-LE LANGAGE HUMAIN**

Le langage humain se distingue du langage animal par le pouvoir de substituer un organe à un autre, c'est-à-dire la faculté d'articulation ou de supplémentarité.

Le langage humain est le moyen le plus utilisé pour communiquer car il peut être exprimé par la voix ou écrit. Dans la majorité du temps, il est oral et respecte des règles strictes pour que l'on soit compris. Ces règles grammaticales sont pratiquées quotidiennement mais seulement 60% de ces normes sont respectées à l'oral. Tandis qu'à l'écrit, on doit suivre ces obligatoirement suivre ces règles.

Pour cela avant d'exprimer quelque chose, on devra tout d'abord réfléchir en tête ce qu'on voudrait dire, ensuite organiser et arranger dans nos pensées la façon dont on voudrait exprimer les mots pour être claire devant notre interlocuteur. C'est le cerveau qui est responsable de ce mécanisme que nous verrons par la suite. La bouche est ensuite l'outil pour exprimer le langage oral, souvent accompagnée par des gestes des deux mains.

Ainsi des tons peuvent varier en fonction de la personne qui parle, résultant de l'humeur de cette dernière qui pourrait être joyeux, en colère, fatiguée.



(1) <b>Un moyen de communication</b>	(2) <b>Une langue</b>	(3) <b>Une faculté</b>
Tout code servant à communiquer	Un système de signes particulier, doublement articulé qui sert à communiquer.	La faculté de produire et d'utiliser une langue

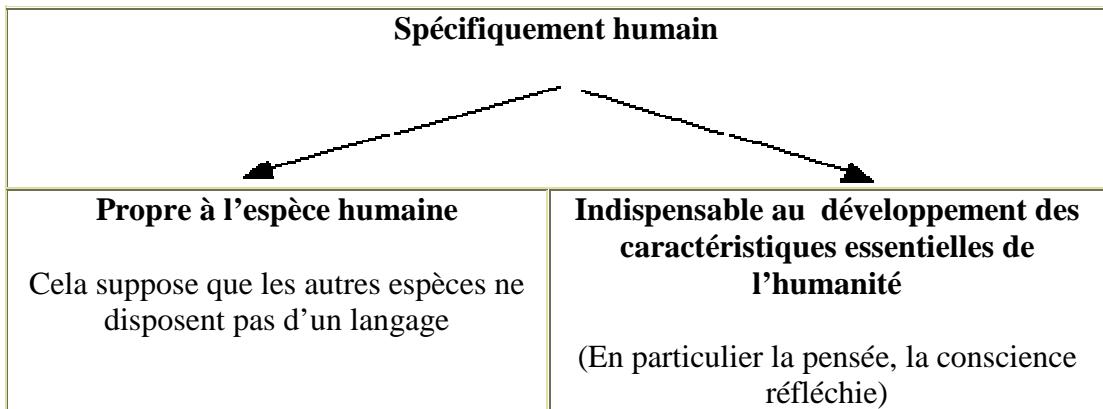


Figure 1.01 : Présentation du langage humain

Source : Auteur

La capacité de communiquer grâce à un code n'est pas propre à l'espèce humaine (Langage au sens 1)

Tout être vivant (même la plante) communique avec son milieu et tout animal social doit pouvoir communiquer avec ses congénères. Donc, SI « Langage » = Tout moyen de communication, ALORS le langage n'est pas spécifiquement humain.

**Exemples :**

- La danse des abeilles, les codes d'expression corporelle chez les mammifères.
- Les chants des oiseaux. Les ultrasons des mammifères marins.

**Mais**

- Tous les moyens de communication sont-ils équivalents ?
- Le langage humain n'est-il qu'un moyen de communication comme un autre ?

### 1.3-LA DIFFERENCE ENTRE UN SIGNE ET UN SIGNAL

- Le langage humain est un système de **signes**. (Union **conventionnelle** d'un signifiant et d'un signifié.) Cf. La définition du signe linguistique par Ferdinand de Saussure.
- Un signe renvoie à une **signification**; un signal provoque une **réaction**. La fonction d'un signal est purement **conative**.
- Dans sa fonction conative, le langage humain fonctionne comme un signal, mais il a aussi d'autres fonctions.
- Les signaux de la communication animale sont **génétiquement déterminés**. Ils ont une fonction fixe (en général étroitement reliée à la survie) plutôt qu'une véritable signification. Les signes linguistiques sont au contraire **conventionnels** et multifonctionnels.

### 1.4- LA DOUBLE ARTICULATION

Le langage humain est un **système** de signes. Les éléments significatifs de base (les monèmes ou morphèmes), non décomposables en éléments significatifs plus simples, se combinent pour former des éléments significatifs complexes. C'est le **premier niveau d'articulation**. Les éléments non significatifs de base (phonèmes) se combinent pour former des éléments significatifs de base. C'est le **deuxième niveau d'articulation**.

<b>Premier niveau d'articulation:</b> monème + monème = ensemble significatif complexe  Ex.: je + pens + e = je pense auto + mobile = automobile	<b>Deuxième niveau d'articulation :</b> phonème + phonème = ensemble significatif simple (monème)  Ex.: j + e = je
--	---

Figure 1.02 : Les niveaux d'articulations

Source : Auteur

C'est une notion vaste, qui a priori englobe tout ce qui communique ou exprime quelque chose. On parle de langage articulé, des gestes (les sourds et muets), de langage des fleurs, de l'art, etc...

1) le langage : faculté de s'exprimer et de communiquer (sans laquelle on ne peut parler)

2) un langage/la langue : système de signes différenciés permettant l'expression et la communication d'informations/de messages

3) la parole : mise en œuvre individuelle du langage dans une langue déterminée, afin de dire quelque chose (en général à quelqu'un); ne suppose pas essentiellement la voix : en effet, un sourd-muet ne peut certes pas émettre de sons mais il dit bien quelque chose à quelqu'un par l'intermédiaire de gestes (signes).

## 1.5-LE CERVEAU HUMAIN : UN PROCESSEUR BIOLOGIQUE

Le cerveau humain est la machine la plus perfectionnée de tout les temps.

En effet, des choses, inventions, peuvent être réalisées avec l'utilisation du cerveau. Des zones sont encore inconnues à la science qui tente de percer son mystère.

Comprendre l'organisation du tissu cérébral et sa logique est l'une des questions centrales de la neurobiologie moderne. Or le cerveau humain est la structure vivante la plus complexe que nous connaissons. Cet organe n'est pas homogène et sa complexité s'exprime par la juxtaposition de différents territoires dont les fonctions sont plus ou moins bien spécifiées.

Les cellules nerveuses sont nommées neurones et, bien que leur fonctionnement dépende d'un autre type cellulaire (les cellules gliales), tout indique que le neurone est l'unité fonctionnelle à l'origine des différentes fonctions que réalise le système nerveux. Rappelons que le cerveau humain comprend des milliards de neurones (de 1011 à 1012), répartis localement en circuits. Ces derniers correspondent soit aux régions corticales, s'ils sont arrangés en strates parallèles ou en colonnes profondes, soit aux noyaux, s'ils sont regroupés en amas moins structurés. Cependant, régions corticales et noyaux profonds ne constituent pas des entités fonctionnelles indépendantes et communiquent grâce à des connexions multiples établies par des projections d'axones pour former des systèmes organisés sous formes de réseaux. La seconde composante à laquelle ces voies sont connectées est dite centrale : elle est formée par les centres nerveux : chez l'homme, la moelle épinière et à l'encéphale. Situé dans la boîte crânienne, ce dernier comprend le télencéphale avec ses deux hémisphères cérébraux, droit et gauche, se rattachant l'un à l'autre par le corps calleux, en dessous duquel est le diencéphale, puis le tronc cérébral et, dorsalement sur celui-ci, le cervelet.

**Philosophique.** Le cerveau est l'organe qui perçoit, qui pense, et qui agit. C'est donc lui qui permet de donner un sens à l'existence. Du reste, la suppression de son fonctionnement signifie la mort véritable.

**Sociologique.** Le cerveau est le chef d'orchestre de l'organisme qu'il gère tout en se gérant lui-même. C'est donc lui qui est responsable de nos comportements, et donc de nos interactions avec les individus qui composent la société.

**Scientifique.** Le cerveau commence à être compris dans son fonctionnement. Nous vivons donc une époque enthousiasmante pour comprendre la genèse de nos facultés intellectuelles et de nos émotions, et donc les comportements moteurs qui en sont l'expression.

**Médical.** Le cerveau malade commence aussi à être compris, qu'il s'agisse des maladies neurologiques (Alzheimer, Parkinson, sclérose latérale amyotrophique, sclérose en plaques, épilepsie, séquelles de traumatisme de la moelle épinière, etc.) ou psychiatriques (dépression, attaques de panique, schizophrénie, autisme, troubles obsessionnels compulsifs, etc.).

Le cerveau est la partie principale de l'encéphale, qui comprend aussi le cervelet et le tronc cérébral. Protégé par la boîte crânienne, il est composé de deux hémisphères et d'une zone appelée diencéphale. Il ne représente que 2 % du poids du corps, mais les centaines de milliards de cellules qui le composent sont le siège des fonctions intellectuelles et sensitives.

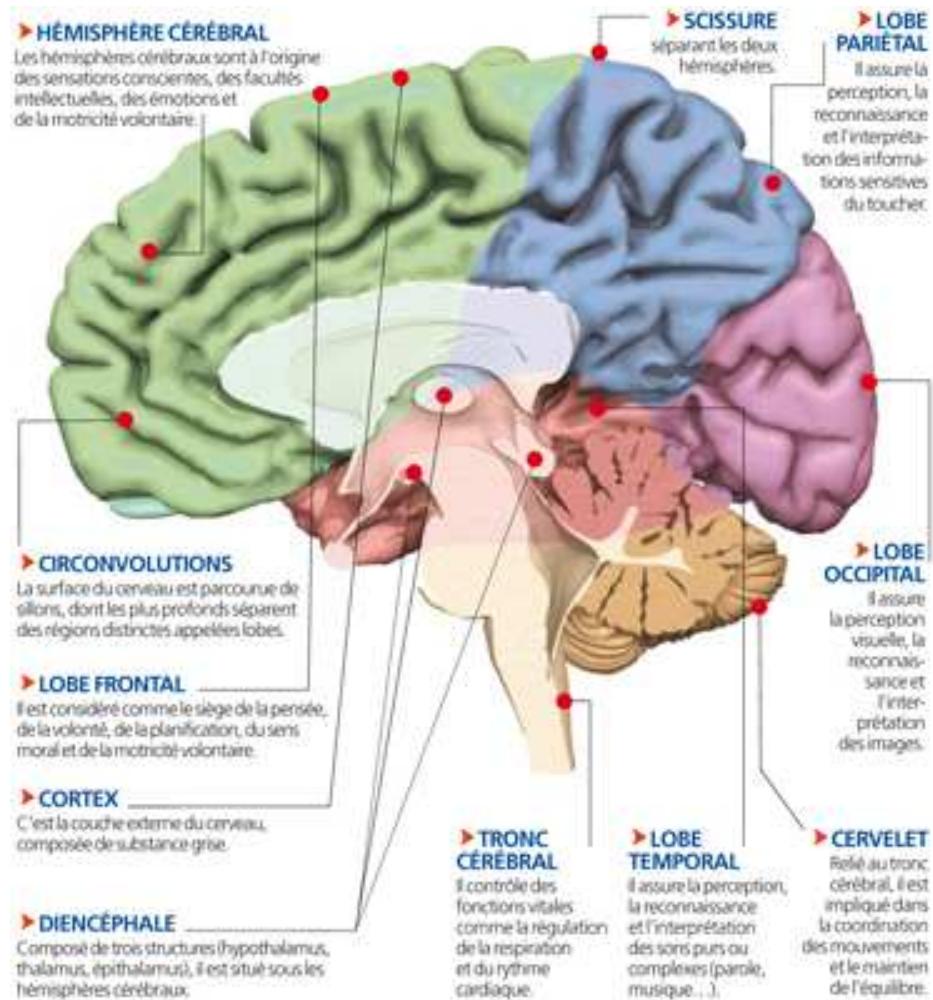
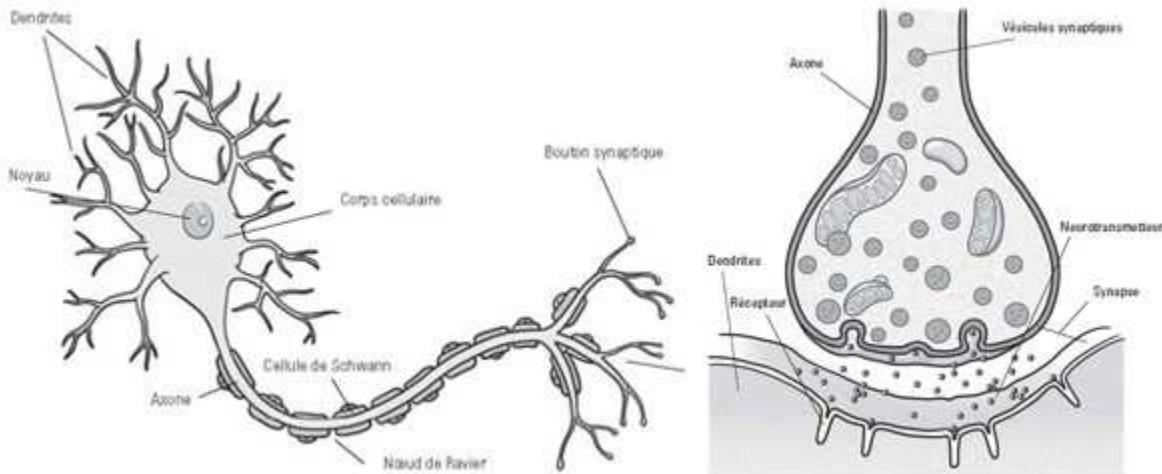


Figure 1.03 : Anatomie générale du cerveau humain

Source : [7]

L'intelligence humaine est due aux neurones.

Les neurones communiquent entre eux par signaux électriques, appelés influx nerveux (ou potentiel d'action). Chaque neurone est constitué d'un corps cellulaire, de prolongements appelés dendrites et axones. Ces derniers émettent des connexions avec d'autres neurones par l'intermédiaire de terminaisons, ayant l'aspect de petites vésicules. Ces terminaisons constituent les synapses.



Le neurone (à gauche) et la synapse (à droite)

Figure 1.04 : Structure générale d'un neurone  
Source : [6]

L'influx nerveux se propage le long de l'axone pour terminer son chemin au niveau de la terminaison synaptique. Plus la fréquence de celui-ci est importante, plus le neurone produit des substances chimiques : les neurotransmetteurs (ou neuromédiateurs).

Seul l'Homme possède des aires corticales adaptées au langage parlé et à la lecture. Le langage résulte d'un ensemble de tâches effectuées dans des régions différentes du cerveau : l'aire de Wernicke (compréhension des mots) et l'aire de Broca (production des mots) reliées entre elles par un faisceau de fibres. Les troubles du langage sont appelés « aphasie ». Le Pr Laurent Cohen étudie les processus cérébraux qui sous tendent la lecture, en particulier la reconnaissance des lettres et des mots.

### 1.6-METHODE DE RAISONNEMENT DU CERVEAU

Le cerveau trouve sa capacité d'intelligence grâce aux réseaux de neurones biologiques.

Comment l'homme fait-il pour raisonner, parler, calculer, apprendre, ...? Comment s'y prendre pour créer une ou de l'intelligence artificielle ? Deux types d'approches ont été essentiellement explorées :

- procéder d'abord à l'analyse logique des tâches relevant de la cognition humaine et tenter de les reconstituer par programme. C'est cette approche qui a été privilégiée par l'Intelligence Artificielle et la psychologie cognitive classiques. Cette démarche est étiquetée sous le nom de *cognitivisme*.
- puisque la pensée est produite par le cerveau ou en est une propriété, commencer par étudier comment celui-ci fonctionne. C'est cette approche qui a conduit à l'étude de réseaux de neurones formels. On désigne par *connexionnisme* la démarche consistant à vouloir rendre compte de la cognition humaine par des réseaux de neurones.

La seconde approche a donc menée à la définition et l'étude de réseaux de neurones formels qui sont des réseaux complexes d'unités de calcul élémentaire interconnectées. Il existe deux courants de recherche sur les réseaux de neurones : un premier motivé par l'étude et la modélisation des phénomènes naturels d'apprentissage à l'aide de réseaux de neurones, la pertinence biologique est importante ; un second motivé par l'obtention d'algorithmes efficaces ne se préoccupant pas de la pertinence biologique. Nous nous plaçons du point de vue du second groupe. En effet, bien que les réseaux de neurones formels aient été définis à partir de considérations biologiques, pour la plupart d'entre eux, et en particulier ceux étudiés dans ce cours, de nombreuses caractéristiques biologiques (le temps, la mémoire, ...) ne sont pas prises en compte. Toutefois, nous donnons, dans la suite de cette introduction, un bref aperçu de quelques propriétés élémentaires de neurophysiologie qui permettent au lecteur de relier neurones réels et neurones formels. Nous donnons ensuite un rapide historique des réseaux de neurones. Enfin, nous donnons une classification des différents types de réseau et les principales applications.

Les *neurones* reçoivent les signaux (impulsions électriques) par des extensions très ramifiées de leur corps cellulaire (les *dendrites*) et envoient l'information par de longs prolongements (les *axones*). Les impulsions électriques sont régénérées pendant le parcours le long de l'axone. La durée de chaque impulsion est de l'ordre d'1 ms et son amplitude d'environ 100 mvolts.

Les contacts entre deux neurones, de l'axone à une dendrite, se font par l'intermédiaire des *synapses*. Lorsqu'un potentiel d'action atteint la terminaison d'un axone, des neuromédiateurs sont libérés et se lient à des récepteurs post-synaptiques présents sur les dendrites. L'effet peut être excitateur ou inhibiteur.

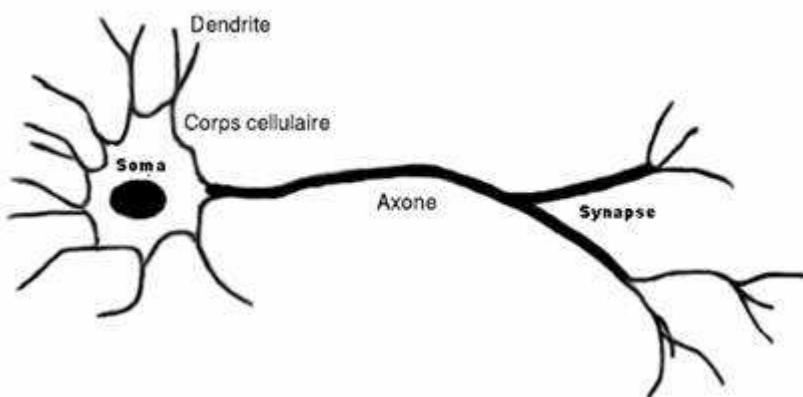


Figure 1.05 : Schéma d'un neurone

Source : [9]

## 1.7-QUELQUES REGLES DE LANGAGE

Présentons maintenant quelques règles du langage. Ces règles sont généralement appelées « grammaire » ou « fitsipi-pitenenana » en Malagasy.

Voici tout d'abord quelques grammaires françaises

### 1.7.1-H MUET

Comme son nom l'indique, le *h* muet estâ€! muet. Il ne correspond à aucun son, ni à aucun mouvement d'air. Celui qui est souvent considéré comme un vestige étymologique a la particularité de n'empêcher ni l'**élision** de la voyelle qui le précède ni la **liaison** entre deux mots.

En d'autres termes, il faut considérer les mots commençant par un *h* muet comme des mots commençant par une voyelle.

Ex. : l'**homophone**, l'**humour**, l'**hiver**, l'**honnêteté**, l'**hélicoptère**.

### 1.7.2-H ASPIRÉ

Le *h* aspiré correspond à un léger bruit de souffle qui **empêche l'élision et, par le fait même, la liaison** entre les mots.

Moins courant que le *h* muet, le *h* aspiré est une trace des origines germaniques ou anglo-saxonnes de certains mots.

Ex. : la **hache**, le **hall**, la **hauteur**, le **héron**, le **hoquet**, le **huitième**.

[Référence n°10]

Le verbe s'accorde **en nombre et en personne** avec son sujet, que le sujet soit exprimé ou sous-entendu.

*Les voitures avancent, foncent, roulent vers le front.*

S'il existe plusieurs sujets, le verbe se met au pluriel.

*Le cheval et la jument galopent vers le bois.*

Le verbe ne marque pas le genre du sujet (*il ou elle mange*); seul le participe passé l'indique. Voir sur cette question l'accord du participe passé.

[Référence n°11]

Dans la langue française une phrase est globalement de la forme :

**SUJET + VERBE + COMPLEMENT.**

Exemple 1 : Le chat mange son repas.

Exemple 2 : Maman arrose les fleurs.

Dans la langue Malagasy, une phrase est normalement de la forme :

**VERBE + COMPLEMENT + SUJET.**

Exemple 1 : Mihinan-kanina ny ankizy.

Exemple 2 : Milalao baolina i Hery.

### **1.8-LA LANGUE MALAGASY**

La langue Malagasy est une langue issue de plusieurs cultures et de plusieurs pays mais la majorité de sa fondation vient de l'Indonésie.

La langue malgache est **originaire d'Indonésie**. Plus précisément, elle appartient au groupe dit "barito" de **la branche malayo-polynésienne** des langues **austronésiennes**. Les autres langues de ce groupe sont parlées dans la province de **Kalimantan** Sud dans l'île de **Bornéo**.

La langue malgache à bien aussi de l'influence Africaine , Arabe, Indienne et Européenne.

[12]

Le malgache fait partie des langues indonésiennes, groupe rattaché à une famille linguistique très vaste, qui s'étend du Pacifique à l'océan Indien (la langue la plus proche du malgache est le maanjan, parlé dans le sud de Bornéo).

A la souche indonésienne (Ve – XIIe siècle) s'ajoutent, à partir du VIIIe siècle, des mots d'origine bantoue, rapportant surtout à l'élevage, et d'autre, arabes, touchant au domaine de la science et de l'astrologie. Les mouvements de population fixent peu à peu les limites de deux grandes familles de dialectes : l'une dans le Centre et l'Est, l'autre dans le Sud et l'Ouest.

C'est dans cette seconde aire, en pays Antaimoro, que le malgache connaît, vers le XVe siècle, Radama Ier, roi unificateur et tourné vers l'Occident, décide d'adopter les caractères latins pour transcrire le Merina, parler du Centre qui devint la langue officielle.

[13]

Notons cependant qu'une vingtaine de dialectes existent à Madagascar et qui sont pratiqués dans les régions et campagnes de l'île. Ici nous avons dix-huit tribus réparties et qui ont chacune leurs cultures, traditions ancestrales, et us.

### **1.8.1-Le terme malagasy**

Au départ, l'adjectif dérivé du nom de pays Madagascar était madécasse. Ce terme, utilisé pour la première fois par les tribus du Sud de la Grande Ile s'est transformé en malagasy, parce que dans cette partie du pays, souvent le son 'de' se transforme en 'le' ou 'la'. Les Français arrivés à la fin du XIXe siècle comme colonisateurs ont francisé le mot malagasy, devenu ainsi malgache ! Entre l'adjectif madécasse-malagasy et la francisation «malgache», le rapport (linguistique) n'est pas évident. D'aucuns pensent que «malgache» a été forgé à dessein, peut-être pour donner une connotation péjorative au mot ; «malagasse» conviendrait mieux et serait moins suspect.

### **1.8.2-Langue et dialectes**

Il y a 18 dialectes à Madagascar, un chiffre correspondant au nombre des tribus malagasy. Evidemment, je fais une simplification. Il y a peut-être plus de dialectes que de tribus car dans certaines régions, plusieurs clans font partie d'une même tribu et parlent leurs dialectes. Et toutes les tribus forment une seule ethnie, au sens premier du terme, l'ethnie malagasy. Ces dialectes ont, par définition, un substrat commun mais aussi des traits caractéristiques les distinguant les uns des autres. En termes plus simples, les dialectes malagasy sont des variantes d'une même langue. Elles varient dans l'espace et aussi dans le temps.

[14]

En Malagasy il existe de mots très similaires aux mots étrangérs. Par exemple, le mot « latabatra » vient du mot français « table », et « sezà » vient du mot français « chaise ».

« Salama » viendrait du mot Arabe « Salam ». Notre langue est un mélange complet de cultures venant de différents pays.

La langue Malagasy devient alors un véritable casse-tête avec ses dialectes.

Exemple 1 : « ianao » se prononce « iha » dans le Sud-Ouest.

Exemple 2 : « Salama » devient « mbola tsara » dans la partie Nord.

### 1.9-SITUATION DES CHOSES

L'apprentissage d'une langue n'est pas facile. Des semaines voire des mois de pratiques et beaucoup d'exercices sont nécessaires. Il s'agit ici de l'expression orale.

Par contre, l'expression écrite est bien plus difficile que l'orale. Des règles grammaticales sont à respecter, les orthographies doivent être correctes en respectant les ponctuations.

N'oublions pas les conjugaisons diverses en passant par les dissertations et les poèmes qui allient à la fois art et règles.

Nombre de personnes ne maîtrisent plus comme avant ces règles. Il est de notre devoir donc de remettre les points sur les « i » en offrant cet outil.

### 1.10-LES RESEAUX DE NEURONES ARTIFICIELS

Beaucoup de chercheurs ont mis au point des théories sur l'intelligence artificielle s'inspirant des réseaux neuronaux biologiques qui nous offrent des résultats spectaculaires.

Un réseau neuronal s'inspire du fonctionnement des neurones biologiques et prend corps dans un ordinateur sous forme d'un algorithme. Le réseau neuronal peut se modifier lui-même en fonction des résultats de ses actions, ce qui permet l'apprentissage et la résolution de problèmes sans algorithme, donc sans programmation classique.

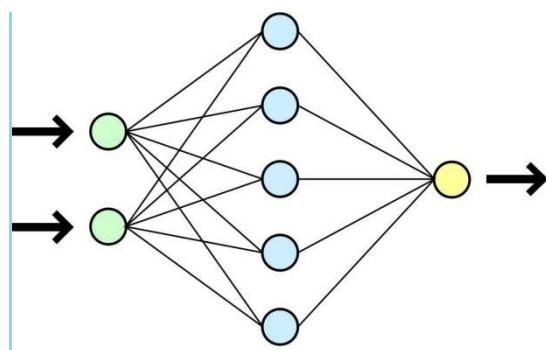


Figure 1.06 : Forme générale d'un réseau de neurone artificiel

Source :[15]

Comment l'homme fait-il pour raisonner, parler, calculer, apprendre... ? Comment s'y prendre pour créer une intelligence artificielle ? Deux types d'approches ont été essentiellement explorées.

### 1.10.1-Approches adoptées en recherche en Intelligence Artificielle

- procéder d'abord à l'analyse logique des tâches relevant de la cognition humaine et tenter de les reconstituer par programme. C'est cette approche qui a été privilégiée par l'Intelligence Artificielle symbolique et la psychologie cognitive classiques. Cette démarche est étiquetée sous le nom de cognitivisme.
- puisque la pensée est produite par le cerveau ou en est une propriété, commencer par étudier comment celui-ci fonctionne. C'est cette approche qui a conduit à l'étude des réseaux de neurones formels. On désigne par connexionnisme la démarche consistant à vouloir rendre compte de la cognition humaine par des réseaux de neurones.

La seconde approche a donc mené à la définition et à l'étude de réseaux de neurones formels qui sont des réseaux complexes d'unités de calcul élémentaires interconnectées. Il existe deux courants de recherche sur les réseaux de neurones : un premier motivé par l'étude et la modélisation des phénomènes naturels d'apprentissage pour lequel la pertinence biologique est importante ; un second motivé par l'obtention d'algorithmes efficaces ne se préoccupant pas de la pertinence biologique. Nous nous plaçons du point de vue du second groupe. En effet, bien que les réseaux de neurones formels aient été définis à partir de considérations biologiques, pour la plupart d'entre eux, et en particulier ceux étudiés dans ce cours, de nombreuses caractéristiques biologiques (le temps, la mémoire...) ne sont pas prises en compte. Toutefois, nous donnons, dans la suite de cette introduction, un bref aperçu de quelques propriétés élémentaires de neurophysiologie qui permettent au lecteur de relier neurones réels et neurones formels. Nous donnons ensuite un rapide historique des réseaux de neurones.

Un neurone est, comme vous l'avez vu, l'unité élémentaire de traitement d'un réseau de neurones. Il est connecté à des sources d'information en entrée (d'autres neurones par exemple) et renvoie une information en sortie. Voyons comment tout cela s'organise.

On note  $(x_i)_{1 \leq i < k}$  les  $k$  informations parvenant au neurone. De plus, chacune sera plus ou moins valorisée vis à vis du neurone par le biais d'un poids. Un poids est simplement un coefficient  $w_i$  lié à l'information  $x_i$ . La  $i$ -ème information qui parviendra au neurone sera donc en fait  $w_i * x_i$ . Il y a toutefois un "poids" supplémentaire, qui va représenter ce que l'on appelle le *coefficent de biais*. Nous le noterons  $w_0$  et le supposerons lié à une information  $x_0 = -1$ . Nous verrons plus tard son utilité, dans la section Fonction d'activation.

Le neurone artificiel (qui est une modélisation des neurones du cerveau) va effectuer une somme pondérée de ses entrées plutôt que de considérer séparément chacune des informations. On définit une nouvelle donnée,  $in$ , par :

$$in = \sum_{i=0}^l w_i \times x_i = ( \sum_{i=1}^l w_i \times x_i ) - w_0$$

C'est en fait cette donnée-là que va traiter le neurone. Cette donnée est passée à la fonction d'activation, qui fait l'objet de la prochaine section. C'est d'ailleurs pour ça que l'on peut parfois appeler un neurone une **unité de traitement**.[Référence n°16]

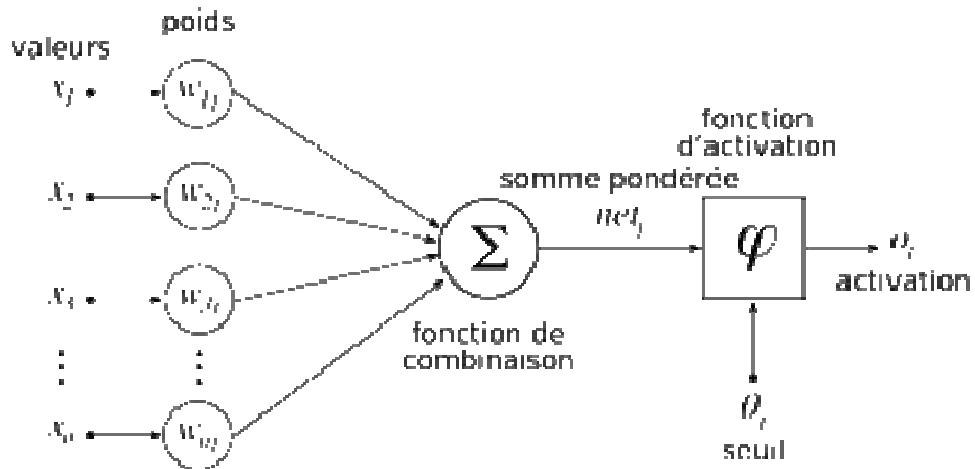


Figure 1.07 : Schéma synoptique d'un Perceptron

Source : [17]

Expression de  $o_j$

Soit :  $p = x \cdot w = x_1 \cdot w_1 + \dots + x_i \cdot w_i + \dots + x_n \cdot w_n$

Alors :  $o = 1$  si  $p > \theta$

$o = 0$  si  $p \leq \theta$

#### a-Modélisation :

Le neurone reçoit les entrées  $x_1, \dots, x_i, \dots, x_n$ . Le potentiel d'activation du neurone  $p$  est défini comme la somme pondérée (les poids sont les coefficients synaptiques  $w_i$ ) des entrées.

La sortie  $o$  est alors calculée en fonction du seuil  $\theta$ .

#### b-Les apprentissages :

- apprentissage supervisé : les coefficients synaptiques sont évalués en minimisant l'erreur (entre sortie souhaitée et sortie obtenue) sur une base d'apprentissage.(...)
- apprentissage non-supervisé : on ne dispose pas de base d'apprentissage. Les coefficients synaptiques sont déterminés par rapport à des critères de conformité : spécifications générales.
- sur-apprentissage : on minimise l'erreur sur la base d'apprentissage à chaque itération mais on augmente l'erreur sur la base d'essai. Le modèle perd sa capacité de généralisation : c'est l

'apprentissage par cœur.

Soit :  $p = x \cdot w = x_1 \cdot w_1 + \dots + x_i \cdot w_i + \dots + x_n \cdot w_n$

Alors :  $o = 1$  si  $p > \theta$

$o = 0$  si  $p \leq \theta$

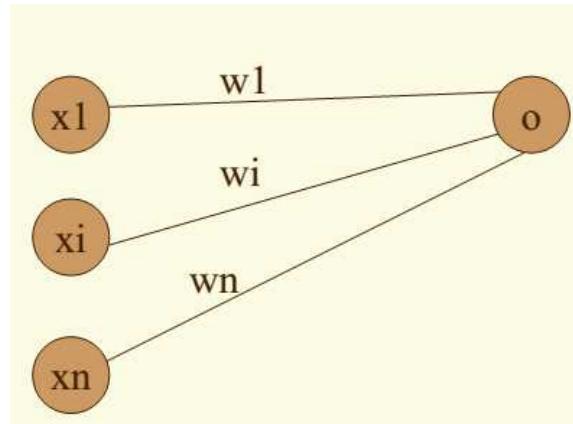


Figure 1.08 : Perceptron simple

Source : [33]

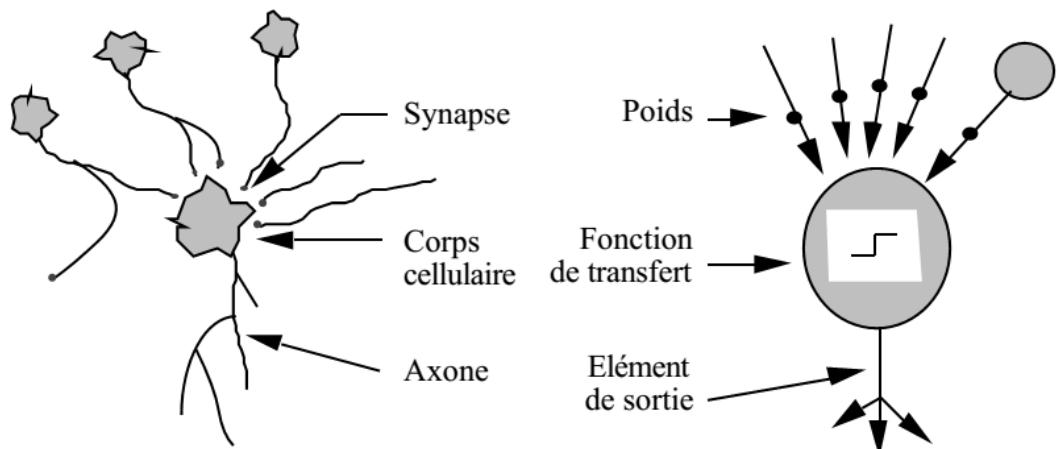


Figure 1.09 : Comparaison entre un neurone biologique et artificiel

Source : [34]

### 1.11-Quelques différents types de réseaux de neurones artificiels

Avant de présenter les célèbres réseaux de neurones artificiels, expliquons d'abord quelques lois fondamentales de ces réseaux.

#### 1.11.1-Loi de Hebb

Réseau de neurones :

- n entrées  $e_1, \dots, e_n$
- m neurones  $N_1, \dots, N_m$ .
- $w_{ij}$  le coefficient synaptique de la liaison entre les neurones  $N_i$  et  $N_j$
- une sortie  $o$
- un seuil  $S$
- Fonction de transfert : fonction Signe

si  $x > 0$  :  $\text{Signe}(x) = +1$

si  $x \leq 0$  :  $\text{Signe}(x) = -1$

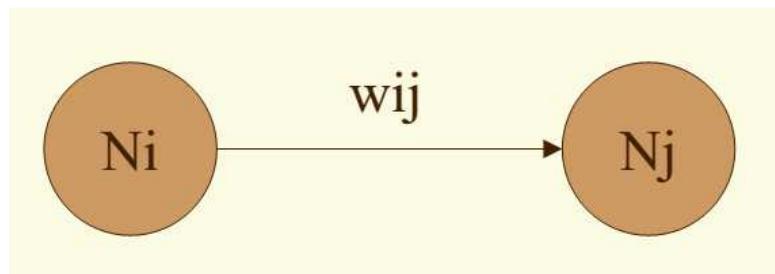


Figure 1.10: Structure d'un réseau avec une entrée, un poids et une sortie

Source : Auteur

#### a-Principe :

Si deux neurones sont activés en même temps, alors la force de déconnection augmente.

#### b-Base d'apprentissage :

On note  $S$  la base d'apprentissage.  $S$  est composée de couples  $(e, c)$  où  $e$  est le vecteur associé à l'entrée  $(e_1, \dots, e_n)$  et  $c$  la sortie correspondante souhaitée.

### 1.11.2-Le neurone de McCulloch et Pitts

Comme on pouvait l'imaginer; les premiers travaux sur les réseaux de neurones ont été menés grâce à des cybernéticiens. La première définition d'un neurone de McCulloch et Pitts date de 1943.

Le neurone *artificiel* modélisé ici ne peut fournir que des réponses du type *tout ou rien*. On parle alors de neurone booléen. La fonction d'entrée est alors très simple à calculer:

$$U_i = E(x_1, \dots, x_j, \dots, x_n) = \sum_{j=1}^n W_{ij} x_j$$

W<sub>ij</sub> représente le poids de la connexion  
x<sub>j</sub> l'activation du neurone j (0 ou 1 par exemple)  
U<sub>i</sub> représente l'entrée du neurone

L'activation du neurone est alors simplement donné par x<sub>i</sub> = f(U<sub>i</sub>)

Il existe d'autres types de neurones moins utilisés comme par exemple les unités Sigma-PI. Dans ce cas, l'entrée ne porte pas directement sur le cumul pondéré des stimulations reçues, mais au contraire sur des produits élaborés à partir d'un certain nombre d'entrées.

Bien qu'un tel modèle reproduise assez fidèlement le neurone biologique, sa mise en oeuvre implique de connaître les poids associés aux connexions. On doit donc appliquer une phase d'apprentissage au réseau pour fixer les poids des connexions.

#### 1.11.1.1-La phase d'apprentissage

La phase d'apprentissage dépend beaucoup de la structure du réseau. Son but est de fixer les poids des connexions. Dans la majorité des réseaux que nous allons étudier ici, l'apprentissage sera dit supervisé, car on impose une entrée fixe et l'on cherche à récupérer une sortie connue. On effectue alors la modification des poids pour retrouver cette sortie imposée. Malgré tout il existe des réseaux à apprentissage non-supervisé, comme par exemple les cartes topologiques de Kohonen. Enfin, certains réseaux associent les deux types d'apprentissage, c'est le cas par exemple du réseau de Boltzmann. [18]

### 1.11.2-Le Perceptron

Le *perceptron* est un modèle de réseau de neurones avec algorithme d'apprentissage créé par Frank Rosenblatt en 1958.

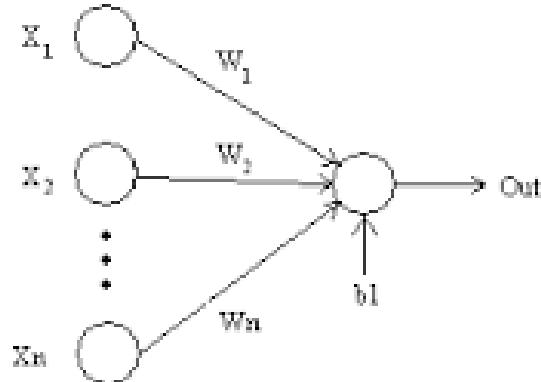


Figure 1.11: Perceptron simple

Source :[19]

Soit :  $p = x \cdot w = x_1 \cdot w_1 + \dots + x_i \cdot w_i + \dots + x_n \cdot w_n$

Alors :  $Out = 1$  si  $p > \theta$

$Out = 0$  si  $p \leq \theta$

On va développer ce modèle plus loin car il sera l'objet de notre cas.

### 1.11.3-Le réseau de Hopfield

Un réseau de Hopfield est une mémoire associative implémentée comme un réseau de neurones entièrement connectés. Chaque neurone correspond à un pixel sur le motif et peut être soit actif ( $S_i = +1$ ), soit inactif ( $S_i = -1$ ). Chaque neurone tente de se fixer dans un état stable par rapport aux entrées qu'il reçoit des neurones auxquels il est connecté. Idéalement, les états stables du réseau (c'est-à-dire lorsque tous les neurones sont dans un état stable) sont identiques aux motifs qui ont initialement été appliqués aux neurones.

La connaissance ou l'information sur ces états préférés est distribuée de manière *inhérente* dans les neurones, ainsi aucune connexion ne peut être vue comme "stockant" un motif. Cela induit d'assez intéressantes propriétés de dégradation harmonieuse et de catégorisation dans un espace surchargé.

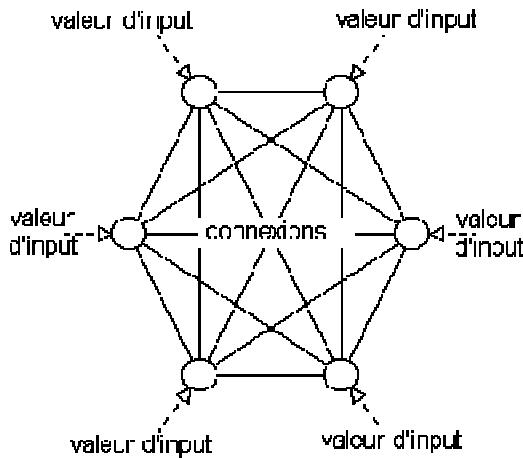


Figure 1.12: Présentation d'un réseau de Hopfield

Source :[20]

Le modèle de Hopfield fut présenté en 1982. Ce modèle très simple est basé d'une part sur le modèle *des verres de spin de Ising* mais aussi sur le principe des mémoires associatives. C'est d'ailleurs la raison pour laquelle ce type de réseau est dit associatif. Outre un intérêt pratique, ce réseau admet une analyse théorique précise et complète. Il a par ailleurs contribué à relancer les recherches sur les réseaux de neurones.

### ❶ Les mémoires auto-associatives

Dans un mémoire informatique classique, une information est retrouvée à partir d'une clé arbitraire. Par opposition, une donnée entreposée dans une mémoire associative est accessible à partir d'informations qui lui sont associées.

La fonction d'une mémoire associative ressemble beaucoup à celle d'un filtre dont le but est de restituer une information en tenant compte de sa perturbation ou de son bruit. L'information doit alors se rapprocher d'une information apprise ou connue. Si les mémoires associatives restituent des informations qu'elles ont apprises à partir d'entrées incomplètes ou bruitées, il existe aussi des mémoires hétéro-associatives qui en plus peuvent associer plusieurs informations entre elles.

### ❷ Architecture du Réseau

Les réseaux de Hopfield sont des réseaux à rétroaction et à connexions symétriques. Les sorties sont fonctions des entrées et du dernier état pris par le réseau. Le principe général se rapproche beaucoup du modèle des *verres de Spin de Ising*.

### ❸ Le principe

Le réseau de Hopfield est l'un des réseaux les plus simples en particulier grâce à un apprentissage facile à mettre en oeuvre. Ce n'est qu'au cours de la phase de test que les problèmes les plus importants apparaissent.

## ● **L'apprentissage**

La règle d'apprentissage proposée par Hopfield est basée sur la règle de Hebb. La règle de Hebb consiste à forcer les poids des liaisons entre les neurones actifs au même moment. Par contre, les poids seront diminués si les neurones sont dans des états contraires. Dans le cas de Hopfield, cette règle est légèrement étendue si l'on considère que deux neurones dans l'état -1 sont actifs. La règle de modification des poids devient donc:

$$W_{ij} = \sum_{p \in P} s_i^p s_j^p, W_{ii} = 0, W_{ij} = W_{ji}$$

P représente le nombre de patrons ou d'exemples

On remarque que la phase d'apprentissage est immédiate en calculant directement les poids à l'aide de cette fonction. C'est d'ailleurs l'un des seuls réseaux qui permet un calcul aussi simple et analytique des poids.[21]

Les réseaux de Hopfield sont des réseaux à minimisation d'énergie : ils sont composés de neurones entièrement interconnectés ; ils n'ont pas une "entrée" et une "sortie", mais ils évoluent à partir d'un état initial.

D'une façon générale, un réseau de Hopfield est donc défini par n neurones, et par une matrice de transfert, W, symétrique

On peut utiliser le réseau de Hopfield dans la reconnaissance de formes à partir d'image bruitée ou partielle, et aussi dans la reconnaissance de caractères.

### **1.11.4-Réseau de Kohonen**

#### **Présentation des cartes d'auto-organisation**

Les cartes d'auto-organisation, qui utilisent l'algorithme non supervisé de Kohonen (Kohonen (1995), Cottrell *et al.* (2003)), ont souvent été utilisées en analyse des données (Blayo et Demartines (1991), Oja et Kaski (1999)). L'exemple qui va nous servir à illustrer la présentation de la méthode est une typologie de la peau du visage établie par le CERIES à partir des données de 212 femmes selon la présence ou l'absence de 17 critères visuels comme *l'aspect gras de la peau* ou de critères tactiles comme *la rugosité au touché*. La distance utilisée dans cette étude est la distance euclidienne (on aurait pu choisir celle du  $\chi^2$  mais nous désirions pouvoir généraliser la méthode à un relevé d'intensité d'apparition du critère au lieu de sa seule présence). Cette base de données a un grand nombre de variables et relativement peu d'individus, ce qui donne beaucoup de souplesse à la structure des données et a rendu cet exemple très intéressant pour mettre en évidence les propriétés d'ajustement des cartes d'auto-organisation. [22]

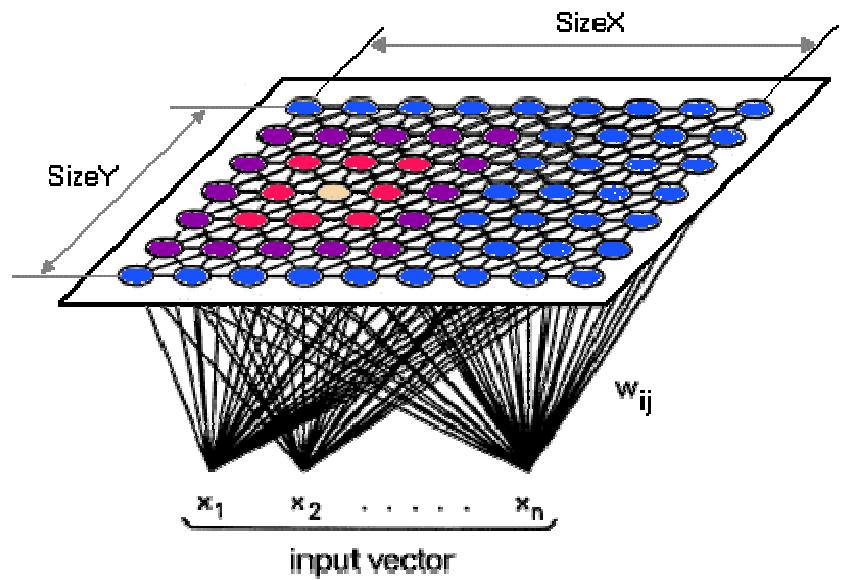


Figure 1.13 : Réseau de Kohonen

Source :[23]

On a vu que quelques réseaux de neurones artificiels, maintenant, entrons dans le vif du sujet qui est l'état de l'art présenté à travers le Perceptron multicouche.

# **CHAPITRE 2**

## **ETAT DE L'ART**

# Chapitre 2 - ETAT DE L'ART

## INTRODUCTION

Etant données que nous avons vu les grands réseaux de neurones, il est temps maintenant de rentrer à la conception de notre application. Pour cela, nous utiliserons le Perceptron multicouche qui est un réseau de neurones artificiels puissant et facile à manipuler grâce à son apprentissage supervisé.

### 2.1-LE PERCEPTRON

On doit à F. Rosenblatt le premier modèle opérationnel de réseau apprenant : le Perceptron. S'inscrivant dans un cadre physiologique et psychologique, l'auteur cherchait àcomprendre le modèle biologique :

*« La théorie présentée ici traite d'une classe de modèles cérébraux appelés « Perceptron ». Par modèle cérébral, nous entendons tout système théorique que tente d'expliquer le fonctionnement psychologique du cerveau en termes de lois physiques et mathématiques connues et de faits connus en neuroanatomie et physiologie. Les Perceptrons sont intéressants en ce que leur étude éclaire la biophysique des systèmes cognitifs. »* [Référence n°29]

Dans le modèle de base, le Perceptron est formé des éléments suivants :

- Des unités sensitives (S-Units) sensibles à des stimulations physiques diverses(son, lumière, chaleur...). Les « unités sensitives simples » renvoient +1 si le signal d'entrée dépasse un seuil  $\theta$ , sinon elles renvoient 0 ;
- Des unités d'association (A-units) qui disposent de connexions entrantes et sortantes et génèrent un signal de sortie en fonction des entrées. Les « unités d'association simples » renvoient +1 si la somme algébrique des signaux d'entrée dépasse un certain seuil  $\theta$ , sinon elles renvoient 0 ;
- Des unités de réponse (R-units) qui disposent de connexions entrantes et génèrent un signal en dehors du réseau. Les « unités de réponse simples » renvoient +1 si la somme des signaux d'entrée est strictement positive ; elles renvoient -1 si elle est négative. Si elle est égale à 0, la sortie est indéterminée.
- Une matrice d'interactions qui définit des coefficients de couplage entre différentes unités(poids synaptiques).

Dans le Perceptron simple, les différents types d'unités sont organisés en couches. Les seules connexions possibles vont de S vers A, puis de A vers R. Les coefficients de couplage de S vers A sont fixes ; ceux de A vers R peuvent varier dans le temps. Il n'ya qu'une seule unité de réponse et la vitesse de transmission est identique pour toutes les unités. Rosenblatt propose ainsi un modèle de perception visuelle où les S-units forment une « rétine ».

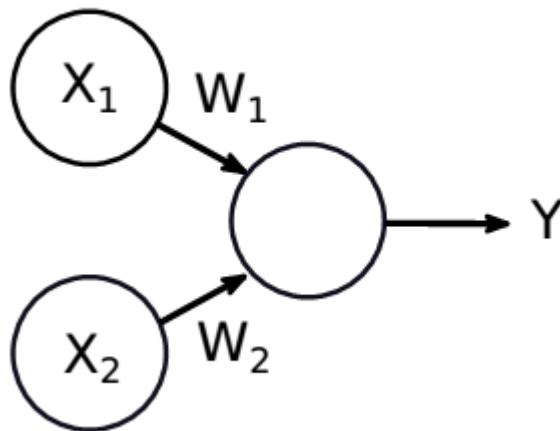


Figure 2.01 : Perceptron élémentaire  
Source : [29]

Le modèle proposé est donc très proche de celui de McCulloch-Pitts avec des neurones utilisant une fonction de transfert de type Heaviside ou signe.

La différence structurelle essentielle réside dans l'existence de « coefficients de couplage », c'est à dire de poids synaptiques ou de poids de connexion permettant de pondérer les signaux d'entrée.

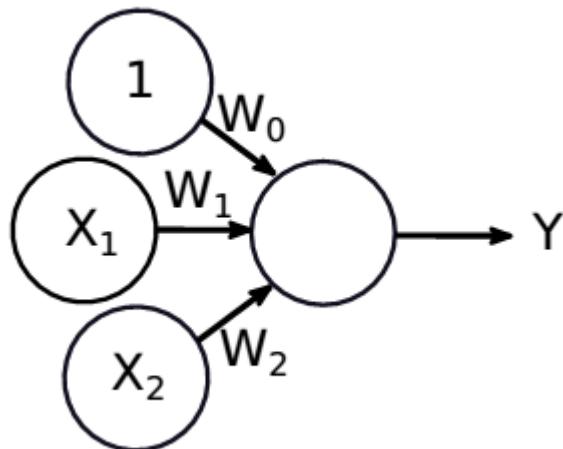


Figure 2.02 : Neurone de sortie du Perceptron  
Source : [32]

Dans la pratique, on utilise fréquemment un lien supplémentaire appelé le « biais ». Ce lien, auquel on affecte généralement l'indice 0, a un poids égal à  $\theta$  (biais et seuil sont parfois utilisés comme synonymes) et est constamment activé par un signal à -1. Ceci permet de traiter les neurones comme si le seuil d'activation était égal à 0.

Nous avons principalement trois fonctions de sortie ou fonction de transfert : Heaviside, Linéaire, Sigmoïde.

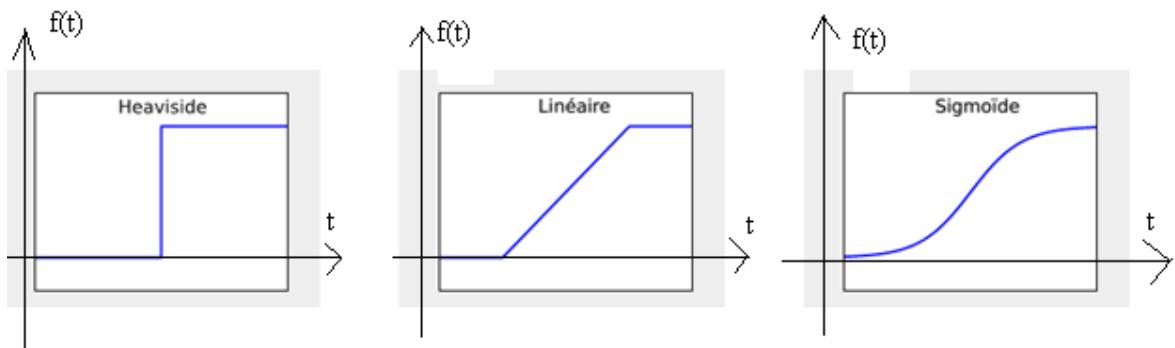


Figure 2.03 : Fonctionnement d'un neurone de sortie du Perceptron

Source :[32]

**Voici les caractéristiques de chaque fonction :**

**Heaviside (seuil  $\theta$ )**

- Si  $x < \theta$  alors  $f(x) = 0$
- Si  $x \geq \theta$  alors  $f(x) = 1$

**Linéaire (seuil  $\theta_1, \theta_2$ )**

- Si  $x < \theta_1$  alors  $f(x) = 0$
- Si  $x > \theta_2$  alors  $f(x) = 1$
- Si  $\theta_1 \leq x \leq \theta_2$  alors  $f(x) = x$

**Sigmoïde / Tangente hyperbolique**

- $f(x) = 1 + \exp 1(-x)$
- $f(x) = \exp(x) - \exp(-x)$
- $\exp(x) + \exp(-x)$

### 2.1.2-Apprentissage

Le perceptron est susceptible d'apprentissage, c'est-à-dire qu'on peut le faire passer par une phase au cours de laquelle les poids synaptiques sont progressivement modifiés pour que la sortie du réseau corresponde à la sortie désirée. Plus généralement, l'apprentissage est défini comme :

« le mécanisme par lequel les paramètres libres d'un réseau de neurones sont adaptés à travers un processus de stimulation par l'environnement dans lequel le réseau est intégré. Le type d'apprentissage est déterminé par la façon dont les changements de paramètres sont mis en œuvre ». [30]

On a donc un système à travers lequel les stimulations externes induisent via des mécanismes divers, des transformations internes modifiant les réponses à l'environnement. On distingue deux grandes classes d'apprentissage : l'apprentissage non supervisé et l'apprentissage supervisé.

Dans le cas de l'apprentissage non supervisé(dit aussi « apprentissage sans professeur »), on ne connaît pas ce que doit être la sortie du réseau. Un exemple classique est

la technique d'agrégation utilisée pour regrouper des données en fonction de régularités statistiques (de caractéristiques communes). Caractérisées par un nombre de paramètre élevé, on cherche ainsi par exemple à réduire le nombre de dimensions caractéristiques) 2 ou 3 afin d'obtenir une structure aisément représentable, tout en conservant les « distances »(variances) séparant ces données.

L'apprentissage supervisé au contraire utilise un « professeur » pour guider le réseau vers la solution recherchée. On distingue ici deux cas :

Le professeur est capable de mesurer l'écart entre les réponses souhaitées et obtenues. Les modifications du réseau sont alors fonction de l'ampleur de l'erreur. On parle d'apprentissage par correction d'erreurs ;

La mesure de l'erreur est simplement booléenne, on n'en connaît pas l'ampleur. On parle d'apprentissage par renforcement.

### ***La règle d'apprentissage du Perceptron***

Le Perceptron fonctionne grâce à un apprentissage supervisé par renforcement.

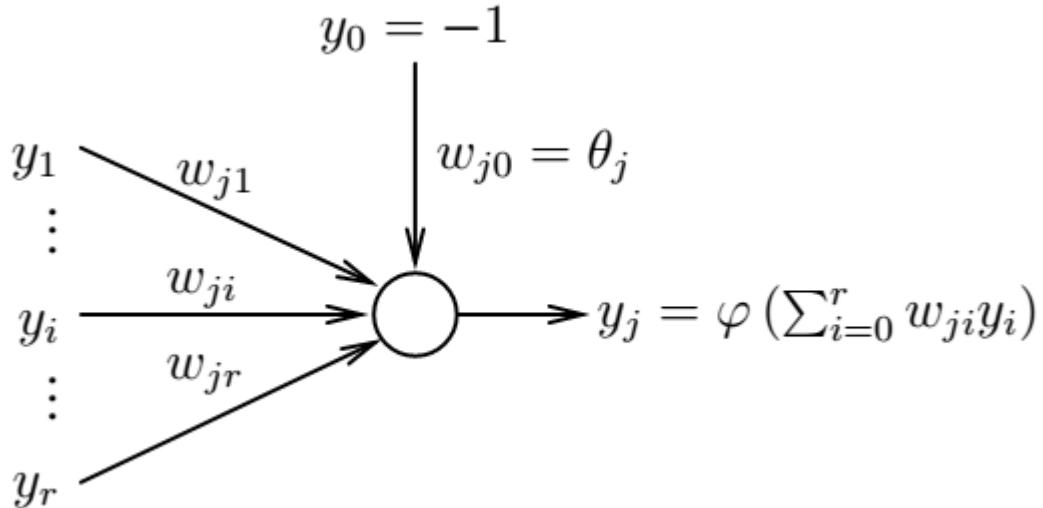


Figure 2.04 : Apprentissage du Perceptron

Source : [29]

La règle du Perceptron est une variante de la règle d'apprentissage de Hebb. Pour Hebb, le poids de la connexion entre deux neurones est renforcé quand ils ont tous deux actifs, soit :

$$\omega_{ij}(t+1) = \omega_{ij}(t) + n(s_i \cdot s_j)$$

Avec :

$\omega_{ij}(t)$ : poids de la connexion entre les neurones i et j à l'instant t.

$n$  : pas d'apprentissage,  $0 < n \leq 1$ .

$s_i$  : signal transmis par le neurone i.

$j_i$  : signal transmis par le neurone j.

Ce qui, si l'on initialise les poids à 0 et le pas d'apprentissage à 1, donne pour k exemples d'apprentissage :

$$\omega_{ij} = \sum_k s_{ik} \cdot s_{jk}$$

La règle de Hebb ne peut que renforcer les poids synaptiques et ne tient pas compte de l'écart entre les sorties réelles et désirée. La règle du Perceptron en est une version évoluée. On peut l'exprimer sous la forme suivante :

$$\omega_{ij}(t + 1) = \omega_{ij}(t) + n(d_j - s_j)s_i$$

Avec :

$\omega_{ij}(t)$  : poids de la connexion entre les neurones i et j à l'instant t.

$n$  : pas d'apprentissage. Situé entre 0 et 1, on fait souvent varier ce coefficient au cours de l'apprentissage. Partant d'une valeur élevée (typiquement environ 0.8), on le réduit progressivement quand on approche de la solution.

$d_j$  : sortie désirée pour l'entrée courante.

$s_j$  : sortie obtenue pour l'entrée courante.

$s_i$  : signal transmis par le neurone i.

**Appuyons-nous sur un exemple pour matérialiser ce mécanisme.** Considérons un Perceptron devant reconnaître une forme s'inscrivant dans une grille de 2 x 2 pixels. Nous utiliserons un neurone d'associations par pixel et un neurone de sortie unique. On cherchera à reconnaître la première diagonale.

On dispose de quatre neurones d'association A1 à A4, recevant respectivement les signaux en provenance des pixels. Le signal reçu est de 1 quand la cellule est noire et de 0 quand elle est blanche. Les neurones d'association émettent des signaux s1 à s4 égaux au x signaux reçus. Ceux qui sont reliés à des cellules noires émettent donc un signal à 1, les autres un signal à 0.

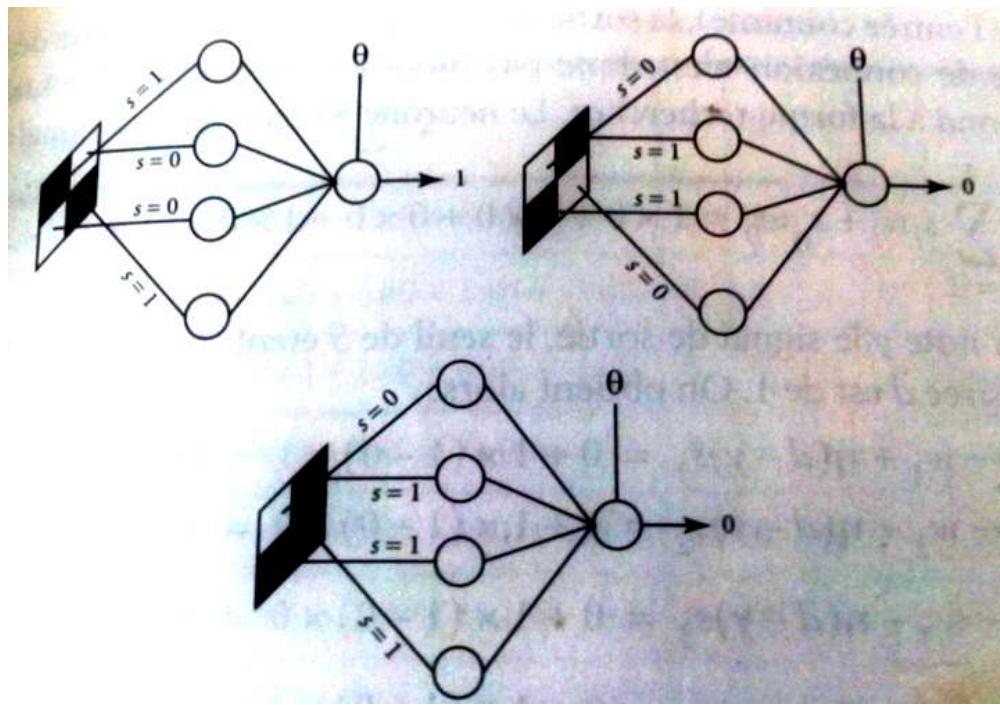


Figure 2.05 : Apprentissage du Perceptron

Source : [29]

Chacun des neurones d'association est connecté au neurone de sortie  $S$  qui émet le signal de sortie  $y$  (on utilisera ici des sorties de la forme 0/1 plutôt que -1/1 pour faciliter les calculs).

On ajoute également un biais de telle manière que le seuil du neurone de sortie soit fixé à 0. Dans cette configuration, il existe quatre liaisons entre les neurones d'association et le neurone de sortie par un lien de poids  $\omega_0$ . Il émet systématiquement le signal  $-1 \times \omega_0$ .

On obtient au final un vecteur de cinq poids qui doit être ajusté au cours du processus d'apprentissage de telle manière que le neurone de sortie s'active en fonction de la forme rechercher

Examinons le processus d'apprentissage de la première diagonale, c'est-à-dire « 1001 ». On initialise les poids à 0 (on aurait pu choisir une initialisation aléatoire) et on fixe le seuil  $\omega_0$  à 2. On utilisera ici un pas d'apprentissage  $n = 1$ .

Au cours de l'apprentissage on présente successivement l'ensemble des entrées possibles, ou des exemples disponibles, et on applique la règle d'apprentissage, c'est ce qu'on appelle une « itération » ou une époque ; ceci jusqu'à ce que les résultats soient considérés comme acceptables. Il existe dans ce cas  $2^4 = 16$  configurations d'entrée possibles que l'on va présenter une première fois au Perceptron. Pour la première itération, la sortie obtenue(0) est égale à la sortie désirée, la matrice de connexion n'est donc pas modifiée.

Dans l'exemple d'entrée (1,0,0,1,-1) on a le calcul suivant

L'on constate que sa sortie correspond à la forme recherchée. Le neurone  $S$  reçoit alors le signal :

$$\sum_{i=1}^4 s_i \omega_i + s_0 \omega_0 = 1 \times 0 + 0 \times 0 + 0 \times 0 + 1 \times 0 + -1 \times 2 = -2$$

Si l'on note  $y$  le signal de sortie, le seuil de  $S$  étant fixé à 0, on a  $y = 0$ . La sortie désirée  $d$  est de 1. On obtient alors :

$$\begin{aligned}\omega_1 &\leftarrow \omega_1 + n(d - y)s_1 = 0 + 1 \times (1 - 0) \times 1 = 1 \\ \omega_2 &\leftarrow \omega_2 + n(d - y)s_2 = 0 + 1 \times (1 - 0) \times 0 = 0 \\ \omega_3 &\leftarrow \omega_3 + n(d - y)s_3 = 0 + 1 \times (1 - 0) \times 0 = 0 \\ \omega_4 &\leftarrow \omega_4 + n(d - y)s_4 = 0 + 1 \times (1 - 0) \times 1 = 1 \\ \omega_0 &\leftarrow \omega_0 + n(d - y)s_0 = 2 + 1 \times (1 - 0) \times -1 = 1\end{aligned}$$

L'entrée  $(1,0,0,0,-1)$  envoie alors au neurone  $S$  le signal  $1 \times 1 + 0 \times 0 + 0 \times 0 + 0 \times 1 \pm 1 \times 1 = 0$ . Le seuil étant fixé à 0, la sortie  $y$  passe à 1 ce qui est contraire à la sortie désirée puisque l'entrée 1000 ne correspond pas à la forme recherchée. L'apprentissage devient :

$$\begin{aligned}\omega_1 &\leftarrow \omega_1 + n(d - y)s_1 = 1 + 1 \times (0 - 1) \times 1 = 0 \\ \omega_2 &\leftarrow \omega_2 + n(d - y)s_2 = 0 + 1 \times (0 - 1) \times 0 = 0 \\ \omega_3 &\leftarrow \omega_3 + n(d - y)s_3 = 0 + 1 \times (0 - 1) \times 0 = 0 \\ \omega_4 &\leftarrow \omega_4 + n(d - y)s_4 = 1 + 1 \times (0 - 1) \times 0 = 1 \\ \omega_0 &\leftarrow \omega_0 + n(d - y)s_0 = 1 + 1 \times (0 - 1) \times -1 = 2\end{aligned}$$

Avec ces paramètres, presque toutes les entrées du premier apprentissage induisent un signal de sortie égal à 0, c'est-à-dire à la sortie désirée. Les poids n'évoluent donc plus au cours de cette itération.

Après 5 itérations, le réseau a tout appris correctement toutes les entrées en sortant les sorties désirées.

Le Perceptron est capable d'apprendre. Il sait en particulier reconnaître différentes formes et on aurait pu de même montrer leur aptitude à apprendre certaines fonctions booléennes ou à traiter des problèmes plus complexes encore.

Mais il a atteint ses limites pour de problèmes plus difficiles. A partir des années quatre-vingt, on a appris à utiliser des réseaux multicouches, aptes à dépasser les limites du Perceptron simple.

## 2.2-LES RESEAUX MULTICOUCHE

L'utilisation de couches qu'on appelle cachées améliore grandement les capacités du Perceptron simple, car on dira que de nouveaux potentiels entrent en œuvre pour la résolution d'un problème complexe que le Perceptron simple n'arrive pas à résoudre.

Mais le problème se pose maintenant d'un algorithme d'apprentissage. Dans le cas d'un Perceptron simple, on a vu comment un apprentissage supervisé permet de corriger les connexions en fonction de l'écart entre les sorties obtenues et souhaitées. Avec un Perceptron multicouche le problème est plus difficile, il n'existe pas de lien direct entre les couches d'entrée et de sortie, on ne peut donc pas assigner directement l'erreur, c'est le fameux « credit assignment problem ».

*« (...)notre jugement intuitif est que l'extension à un Perceptron multicouche est stérile. On découvrira peut-être un théorème de convergence puissant ou alors quelques raisons profondes de l'échec à produire un « théorème d'apprentissage» intéressant pour une machine multicouche sera découverte ». [31]*

***L'objectif du réseaux de neurones multicouches est de minimiser l'erreur de la fonction coût, c'est-à-dire, arrêter l'apprentissage du réseau quand l'erreur est proche de la valeur de 0.***

Le pessimisme de Minsky et Papert à la fin des années soixante n'a pas empêché les recherches de se poursuivre et, au cours des années quatre-vingt, un algorithme d'apprentissage des Perceptrons multicouches a été découvert : l'algorithme dit de « rétropropagation du gradient » qui est en fait une extension de la règle delta utilisée par l'Adaline(autre forme de réseau de neurones artificiels).

### 2.2.1-Fonction de transfert

La rétropropagation est basée sur la méthode de la plus profonde descente (methode du gradient), il est donc nécessaire d'utiliser des fonctions dont on puisse que la composition rest continue et dérivable. Ce n'est évidemment pas le cas de la fonction de Heaviside ou de la fonction signe utilisée dans le Perceptron simple. La fonction la plus couramment utilisée est la fonction sigmoïde :

$$S_c(x) = \frac{1}{1 + e^{-cx}}$$

Calcul de sa dérivée :

$$S'_c(x) = c \cdot \frac{e^{-cx}}{(1 + e^{-cx})^2} = c \cdot S_c(x) \cdot (1 - S_c(x))$$

Soit pour c égale 1 :

$$S'_c(x) = S_c(x) \cdot (1 - S_c(x))$$

La fonction sigmoïde donne une réponse sur l'espace  $[0, -1]$ , la tangente hyperbolique permet quant à elle d'obtenir une réponse sur l'espace  $[-1, +1]$ . L'utilisation d'une réponse bipolaire facilite l'apprentissage, dans la mesure où, par rapport à une réponse binaire, elle tend à mieux « homogénéiser » l'espace de recherche. On a :

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \tanh(x)' = \frac{4}{(e^x + e^{-x})^2}$$

La fonction sigmoïde et la tangente hyperbolique sont parfaitement adaptées à la méthode de la plus profonde descente. Dans un cas comme dans l'autre, la fonction est strictement monotone, le calcul du gradient permet donc toujours d'obtenir une direction fiable. En revanche, à la différence du Perceptron simple, les valeurs de sortie ne sont pas entières ; on ne peut donc attendre du réseau une réponse binaire et on devra considérer des seuils d'acceptation (0,9 pour 1 par exemple).

Mise en œuvre de l'algorithme de rétropropagation

Afin d'illustrer l'algorithme de rétropropagation du gradient, nous pouvons en mettre en œuvre les étapes successives sur *un exemple simple*.

*On utilisera un réseau 3-2-3. La première couche comporte trois neurones d'entrée, la seconde deux neurones cachés et la troisième trois neurones de sortie.*  
Pour les besoins de notre exemple, on posera qu'à l'exemple  $X_1 = 0.9, X_2 = 0.1, X_3 = 0.9$ , doivent correspondre les sorties :  $Y_1 = 0.1, Y_2 = 0.9, Y_3 = 0.9$ .

#### Etape 1 : Initialisation du réseau

On initialise les poids de manière aléatoire. Il est recommandé de les distribuer uniformément sur un intervalle  $[-m, +m]$  de manière à ce que la moyenne soit nulle. Dans ce cas, l'espérance des entrées de chaque neurone est nulle, valeur où pour une sigmoïde, la dérivée est la plus élevée.

On augmente ainsi les corrections apportées lors de la rétropropagation.

#### Etape 2 : Propagation avant

On présente le premier exemple aux neurones d'entrée. Chacun des deux neurones de la couche cachée calcule un potentiel de la forme :

$$P_i = \sum w_{ij} \cdot s_j$$

Soit :

$$P_{h1} = 0.9 \times 0.100 + 0.1 \times 0.150 + 0.9 \times 0.050 = 0.150$$

$$P_{h2} = 0.9 \times 0.120 + 0.1 \times 0.180 + 0.9 \times 0.080 = 0.198$$

Chaque neurone de la couche cachée calcule ensuite son signal. On utilisera ici une sigmoïde de paramètre 1 (fonction logistique). Soit :

$$S_{h1} = \frac{1}{1 + e^{-0.150}} = 0.537$$

$$S_{h2} = \frac{1}{1 + e^{-0.198}} = 0.549$$

On propage ensuite les résultats de la couche cachée vers la couche de sortie. On calcule d'abord les potentiels des neurones de sortie, soit :

$$P_{y1} = 0.537 \times 0.100 + 0.549 \times 0.140 = 0.131$$

$$P_{y2} = 0.537 \times 0.125 + 0.549 \times 0.210 = 0.182$$

$$P_{y3} = 0.537 \times 0.130 + 0.549 \times 0.070 = 0.108$$

On calcule enfin le signal de chacun des neurones de sortie, soit :

$$S_{y1} = \frac{1}{1 + e^{-0.131}} = 0.533$$

$$S_{y2} = \frac{1}{1 + e^{-0.182}} = 0.545$$

$$S_{y3} = \frac{1}{1 + e^{-0.108}} = 0.527$$

### Etape 3 : Calcul de l'erreur

On calcule maintenant l'erreur totale afin de voir si le réseau a convergé, soit pour O neurones de sortie :

$$E = \frac{1}{2} \sum_{o=1}^O (d_o - s_o)^2$$

On a :

-pour  $Y_1$ :  $0.1$  (sortie désirée) -  $0.533$  (sortie actuelle) =  $-0.433$

-pour  $Y_2$ :  $0.9 - 0.549 = 0.355$

-pour  $Y_3: 0.9 - 0.527 = 0.373$

Ce qui donne :

$$- E = 0.5 \times (-0.433^2 + 0.355^2 + 0.373^2) = 0.226$$

Si l'erreur est jugée suffisamment faible, on considère que l'apprentissage est terminé.

#### Etape 4 : Calcul du signal d'erreur sur la couche de sortie

Si le réseau n'a pas convergé, on doit maintenant calculer le signal d'erreur sur chacun des neurones de la couche de sortie. On rappelle que :

$$\delta_0 = (d_0 - s_0) \cdot \varphi'(1 - s_0)$$

Soit :

-pour  $Y_1: 0.533 \times (1 - 0.533) = 0.249$

-pour  $Y_2: 0.545 \times (1 - 0.545) = 0.248$

-pour  $Y_3: 0.527 \times (1 - 0.527) = 0.249$

Ce qui donne en reprenant les écarts  $(d_0 - s_0)$  calculés à l'étape 3, les signaux d'erreur suivants :

-pour  $\delta_{y1} = -0.433 \times 0.249 = -0.108$

-pour  $\delta_{y2} = 0.355 \times 0.248 = 0.088$

-pour  $\delta_{y3} = 0.373 \times 0.249 = 0.093$

On ne peut pas calculer immédiatement les corrections des poids synaptiques entre la couche cachée et la couche de sortie. On a en effet besoin des poids initiaux pour retropropager l'erreur de la sortie sur les neurones de la couche cachée.

#### Etape 5 : Calcul du signal d'erreur sur la couche cachée

On calcule maintenant les signaux d'erreur sur chacun des neurones de la couche cachée, soit :

$$\delta_h = \varphi'(P_h) \cdot \sum_{o=1}^O \delta_o \cdot \omega_{ho}$$

Le calcul de la dérivée est identique à la couche de sortie, soit :

-pour  $H_1: 0.537 \times (1 - 0.537) = 0.249$

-pour  $H_2: 0.550 \times (1 - 0.550) = 0.247$

Pour chaque neurone de la couche cachée, on doit maintenant cumuler les signaux d'erreur des neurones de sortie qui leur sont liés, pondérés par les coefficients synaptiques. Soit :

- Pour

$$H_1: -0.108(\text{signal d'erreur de } Y_1) \times 0.100(\text{poids de la connexion de } H_1 \text{ à } Y_1) + 0.088(\text{signal d'erreur de } Y_2) \times 0.125(\text{poids de la connexion de } H_1 \text{ à } Y_2) + 0.093(\text{signal d'erreur de } Y_3) \times 0.130(\text{poids de la connexion de } H_1 \text{ à } Y_3) = 0.0123$$

$$- \text{ Pour } H_2: -0.108 \times 0.14 + 0.088 \times 0.210 + 0.093 \times 0.070 = 0.00987$$

Ce qui nous donne les signaux d'erreur suivants :

$$\delta_{h1} = 0.249 \times 0.0123 = 0.0031$$

$$\delta_{h2} = 0.247 \times 0.00987 = 0.0024$$

#### Etape 6 : Correction des poids synaptiques de la couche de sortie.

La correction utilise les fonctions suivantes :

$$\begin{aligned} \omega_{ho}(t + 1) &= \omega_{ho}(t) + \Delta\omega_{ho} \\ \Delta\omega_{ho} &= \eta \cdot \delta \cdot s_h \end{aligned}$$

Soit avec  $\eta = 1$  :

$$\begin{aligned} \text{pour } \omega_{h1,y1} &: 0.100(\text{poids de la connexion de } H_1 \text{ vers } Y_1) + \\ &[-0.108(\text{signal d'erreur pour } Y_1) \times 0.537(\text{signal de sortie de } H_1)] = 0.042 \end{aligned}$$

$$\text{-pour } \omega_{h1,y2} : 0.125 + (0.088 \times 0.537) = 0.172$$

$$\text{-pour } \omega_{h1,y3} : 0.130 + (0.093 \times 0.537) = 0.180$$

$$\text{-pour } \omega_{h2,y1} : 0.140 + (-0.108 \times 0.550) = 0.081$$

$$\text{-pour } \omega_{h2,y2} : 0.210 + (0.088 \times 0.550) = 0.258$$

$$\text{-pour } \omega_{h2,y3} : 0.070 + (0.093 \times 0.550) = 0.121$$

### Etape 7 : Correction des poids synaptiques de la couche cachée

La correction utilise les fonctions suivantes :

$$\omega_{ih}(t + 1) = \omega_{ih}(t) + \Delta\omega_{ih}$$

$$\Delta\omega_{ih} = \eta \cdot \delta_h \cdot s_i$$

Soit avec  $\eta = 1$  :

-pour  $\omega_{x1,h1} : 0.100$  (poids de la connexion de  $I_1$  vers  $H_1$ ) +  
[ $0.0031$  (signal d'erreur pour  $H_1$ )  $\times 0.9$  (signal de  $I_1$ )] =  $0.103$

-pour  $\omega_{x2,h1} : 0.150 + (0.0031 \times 0.1) = 0.150$

-pour  $\omega_{x3,h1} : 0.050 + (0.031 \times 0.9) = 0.053$

-pour  $\omega_{x1,h2} : 0.120 + (0.0024 \times 0.9) = 0.122$

-pour  $\omega_{x2,h2} : 0.180 + (0.0024 \times 0.1) = 0.180$

-pour  $\omega_{x3,h2} : 0.080 + (0.0024 \times 0.9) = 0.082$

Avec ces nouveaux paramètres, l'erreur totale qui était de  $0.226$  passerait à  $0.210$ . On est donc descendu le long de la surface d'erreurs.

Dans le cas d'un apprentissage réel, on présenterait un nouvel exemple et on retournerait à l'étape 2. Dans le code de programmation, on met une condition qui arrête l'apprentissage quand l'erreur est proche de 0, exemple :  $0.0023$ .

### **2.3-PRESENTATION DE NOTRE CAS**

**Les calculs sont les mêmes que cités précédemment mais avec une plus longue entrée et sortie, donc nous ne détaillerons plus ces détails en chiffre.**

Nous allons présenter la modélisation de notre projet avec les Perceptrons multicouches.

Comme nous l'avons dit, l'application fera l'ordonnancement automatique des phrases écrites dans n'importe quel ordre en respectant les règles grammaticales.

De plus nous pourrions aussi avoir une traduction des mots Malagasy en Français et en plusieurs langues (espagnol, anglais, allemand, portugais,...) vocales utilisant la technologie Google TTS (Text To Speech).

Les Malagasy auront donc à leur portée une sorte de correcteur, traducteur en plusieurs langues de leur langue maternelle propulsée par la technologie mobile.

Nous aurons donc deux Perceptrons multicouches pour réaliser ce projet. L'un, pour apprendre les mots Malagasy avec leur genre (nom, verbe, article,...) et le deuxième, pour faire la traduction française de chaque mot.

Voici les étapes pour réussir ce projet :

#### *Etape 1 :Identification des mots*

Avant toute chose, il est impératif d'identifier le genre (verbe, article,...) de chaque mot pour pouvoir les coordonner correctement pour avoir une phrase suivant les normes.

Exemple : Verbe – article – nom commun = Milalao ny ankizy.

Les mots Malagasy vont être introduits dans le système et vont être étiquetés avant l'apprentissage.

- Mihinana=verbe
- Andy=nom propre
- Ny=article
- Baolina=nom commun de chose
- Izaho=pronom personnel

Tous les mots Malagasy devront alors être enregistrés dans le système avec leurs genres respectifs.

Pour notre cas, on va que quelques mots courants pour démontrer la modélisation avec les Perceptrons multicouches.

Exemple : Izaho sy i Hery dia milalao baolina. Ny ankizy dia matory.

#### *Etape 2 : Présentation de chaque mot au Perceptron multicouches.*

Nous aurons 72 couches d'entrées, 7 couches cachées et 72 couches de sorties.

Nous aurons donc :

*Exemple 1 :*

Entrée : baolina

Et en sortie : nom commun

*Exemple 2 :*

Entrée : milalao

Sortie : verbe

*Exemple 3 :*

Entrée : ny

Sortie : article

Le mot d'entrée et de sortie vont donc être convertis en binaire :

pour ce faire, chaque lettre alphabétique a sa correspondance en binaire, ce qui fait par exemple : baolina = 11000101100001110111110110011010011101100001

Et on fait de même pour tous les mots :

Nomcommun = 10011101101111101101100011101111101101110110111101011101110

Donc à titre d'exemple voici un exemple de réseau pour « baolina » en entrée et « nom commun » en sortie :

Entrée : 110001011000011101111101100110100111011101100001

Sortie : 10011101101111101101100011101111101101110110111101011101110

Et tous les mots vont donc être présentés de la sorte dans le Perceptron multicouche.

### *Etape 3 : Tests*

Nous avons utilisés qu'une vingtaine de mots Malagasy ici, donc le Perceptron multicouche à appris ces mots avec leurs genres (verbe,..) en 750 itérations.

Pour une centaine de mots voire milliers, il faut réétudier les différentes couches de notre réseau.

Maintenant que le Perceptron multicouche a appris ces mots, on peut lui passer des tests à partir de ces mots.

Par exemple, si nous lui donnons en Entrée : « mihinana », le réseau sortira tout de suite la réponse : « verbe », car ce mot existe déjà dans sa connaissance.

De ce résultat nous pouvons dire que tous les mots vont être attribués à un genre (article, nom propre,...).

#### Etape 4 : Tests de phrase complète

Après cela, un nous attendrons les tests venant de l'utilisateur, ici par exemple :

« *Baolina ankizy ny milalao.* »

La phrase est en total désordre. Nous allons mettre en ordre cette phrase. Pour cela, un deuxième apprentissage de la part du Perceptron multicouche est nécessaire pour maîtriser les règles grammaticales d'une phrase correcte afin de la mettre en ordre.

#### Etape 5 : Présentation en entrée de la phrase en désordre

On doit présenter au réseau les entrées constituées de la phrase en désordre.

Exemple : « *Baolina ankizy ny milalao.* »

Chaque mot va être converti en binaire d'abord, et la sortie correcte aussi.

Prenons comme exemple : « baolina »=00011, « ankizy »=00111 et ainsi de suite. On aura une suite de binaire alors pour une phrase.

Ici, la sortie devra être : « Milalao baolina ny ankizy ». Cette phrase correcte va aussi être convertie en binaire.

#### ***Finalement :***

Entrée : « *Baolina ankizy ny milalao.* »

Sortie : « Milalao baolina ny ankizy.»

Puis on fait de même pour toutes les possibilités de phrases avec en sorties, les réponses adéquates.

#### ***Exemple 1 :***

Entrée : « *Matory ankizy ny.* »

Sortie : « Matory ny ankizy.»

#### ***Exemple 2 :***

Entrée : « *Olona i nahita dia ilay Mary.* »

Sortie : « I Mary dia nahita ilay olona.»

Le Perceptron multicouche va apprendre toutes ces propositions ainsi que chaque genre de chaque mot. Au total, il fera cela en 1700 itérations.

### *Etape 6 : Sortie correcte, résultat d'une phrase en désordre*

Maintenant si on fait des tests en entrant des mots en désordre, notre réseau de neurones artificiels va sortir d'abord le genre de chaque mot puis mettre en ordre ces mots pour avoir une phrase correcte.

Remarque : Cette application ne connaît pas encore les mots de liaisons comme : « zanak'i, entam-barotra, lalaovin'ny ankizy. »

### *Etape 7 : Interface graphique*

Une interface graphique présentera un champ d'entrée, puis après appui d'un bouton, le résultat de la phrase correcte.

## **2.4-TRADUCTION VOCALE AVEC Google TTS (Text To Speech)**

Une fois que le réseau ait appris chaque mot ainsi que chaque structure de phrase correcte, l'utilisateur peut bénéficier d'une traduction française vocale du mot ou de la phrase qu'il a entrée. D'autres langues sont aussi disponibles.

### **2.4.1-Présentation de TTS (Text To Speech)**

TTS est une bibliothèque Android qui est spécialisée à la traduction vocale d'un texte quelconque. Une trentaine de langues est supportée mais le Malagasy n'est pas encore disponible dans TTS.

Pour pouvoir l'utiliser, il faut principalement télécharger les langues disponibles de TTS sur internet et l'installer sur Android.

Le grand avantage de TTS est qu'à partir d'un mot présenté, par exemple : « bonjour », il peut nous donner la traduction de « bonjour » en plusieurs langues (espagnol-anglais-portugais-....).

Il faut et il suffit que le mot soit en français pour que TTS le traduise en plusieurs langues.

L'apprentissage des langues étrangères devient alors accessible à tous.

### *Etape 1 : Conversion des mots en binaire puis apprentissage*

Encore une fois, l'intelligence artificielle nous sera d'un grand secours car, un deuxième Perceptron multicouche se chargera d'apprendre chaque mot Malagasy et sa traduction en français.

Exemple : « baolina »=ballon, « mihinana »=manger, « nahita »=a vu.

Entrée : « baolina »

(avec en binaire = 110001011000011101111101100110100111011101100001)

Sortie : « ballon »

(sortie en binaire = 11000101100001110110011011001101111101110)

Ainsi, après 700 itérations le réseau aura appris chaque mot Malagasy avec sa propre traduction en français.

### *Etape 2 : Tests*

Après un test, prenons par exemple le mot « sakafo », le réseau de neurones artificiel va sortir la réponse « nourriture ».

Cette réponse spontanée du Perceptron multicouche va être entrée dans la classe du TTS (Text To Speech) pour avoir une traduction vocale en français. Précisons qu'on doit avoir le mot en français au minimum pour avoir la traduction d'autres langues.

### *Etape 3 :*

Démonstration Android : On va voir par la suite comment on a fait pour implémenter ce système sous Android.



Figure 2.06: BugDroid, le robot logo d'Android

Source :[24]

# **CHAPITRE 3**

# **IMPLEMENTATION**

# Chapitre 3 - IMPLEMENTATIONS

## **3.1-PRESENTATION GENERALE DE LA TECHNOLOGIE MOBILE**

La technologie mobile est actuellement très poussée et tout le monde, presque tout le monde l'utilise, en passant par les enfants, grandes personnes, âgés.

Le téléphone portable est devenu un objet du quotidien. D'ailleurs, certains n'imaginent plus leur vie sans smartphone. L'émergence de la téléphonie dite « mobile » a été une véritable révolution technologique et une évolution majeure dans la vie des consommateurs qui sont, pour la plupart, connectés en permanence. Retour sur l'histoire du téléphone portable, des années 80 à nos jours.

Entre les premiers modèles de téléphones mobiles sortis dans les années 80 et les smartphones d'aujourd'hui, l'évolution du téléphone portable semble fulgurante. Elle s'est accompagnée d'une amélioration du réseau de téléphonie mobile, passant de la 1G à la 4G et peut-être bientôt la 5G.

On peut faire beaucoup de chose avec un téléphone mobile parce qu'il renferme une multitude de performances comme les applications personnalisées, les capteurs intégrés, la gestion des caméras, sons et bien d'autres fonctionnalités.

Pour notre cas, on utilisera un smartphone Android qui contiendra notre application spécialisées.

### **L'invention du téléphone portable**

L'invention du téléphone mobile est attribuée à un ingénieur de Motorola : Martin Cooper, en avril 1973. Mais c'est seulement 19 ans plus tard, en 1992 et après moult recherches et mises au point, qu'on assiste à la naissance du premier téléphone portable. Pourtant, les ondes électromagnétiques et le réseau (radio, TV...) employés par la téléphonie mobile, sont des technologies connues depuis les années 1940.

### **Le premier téléphone portable**

Le premier téléphone mobile a été développé par Motorola (le Motorola DynaTAC 8000X) en 1983 mais il était bien trop encombrant pour être réellement portable ! Il a donc fallu travailler sur la miniaturisation des composants. Ce n'est qu'en 1992 que le véritable premier téléphone portable est arrivé sur le marché américain, mais à un prix exorbitant ! Du coup, il n'intéressait que très peu le consommateur de classe moyenne et restait l'apanage des plus aisés.

## **La 2ème génération de téléphones mobiles**

En 1995, la 2ème génération de téléphones portables couramment appelée la 2G fait son entrée sur le marché et révolutionne le monde de la téléphonie mobile. Petits, performants et dotés d'une batterie efficace, ces mobiles connaissent immédiatement un succès planétaire. Ces innovations sont à l'origine du boom de la téléphonie mobile.

## **La 3ème génération de téléphones portables**

Actuellement, c'est la 3G qui domine le marché, même si les forfaits mobiles 4G commencent à se développer. Depuis les années 2000, la téléphonie mobile est présente partout dans le monde. Dans l'hexagone, les Français de 15 à 70 ans sont équipés de téléphones portables !

Ils intègrent de nombreuses innovations dans la technologie et les services et permettent d'envoyer des SMS, mais aussi des images, des photographies, des sons et des vidéos.

Les nouveaux équipements embarqués associés au Téléphone portable permettent désormais de lire et rédiger des e-mails, naviguer sur Internet, photographier et enregistrer des vidéos, écouter de la musique ou encore regarder la Télévision

### **3.2-ANDROID ET SES CAPACITES**

#### **La naissance d'Android**

Si je vous dis « Android », une grande majorité d'entre vous me diront « Google ». Vous avez raison, à l'heure actuelle, mais l'histoire commence bien avant l'arrivée de la firme de Mountain View sur le système d'exploitation. En effet, tout a débuté avec une société américaine du nom d'Android justement, fondée en 2003.

Celle-ci a été ensuite rachetée par Google deux ans plus tard (en 2005). L'objectif premier était de développer un système d'exploitation qui permettrait à l'utilisateur d'interagir avec ce dernier. Vous le savez peut-être déjà, mais par le passé, chaque constructeur développait son **propre système embarqué**. Dès lors, il était impossible de concevoir une application compatible sur tous les appareils, sans même parler des bibliothèques de développement fournies qui s'en trouvaient bridées afin que les secrets de fabrication des marques ne soient pas divulgués.

Au premier mois de l'année 2007, la marque à la pomme a présenté une véritable révolution : l'iPhone. C'est là que tout a basculé. Le système iOS se voulait moderne, bien en avance sur la technologie actuelle et l'annonce faite par Apple a été une grande claque pour les concurrents.

Comment aurait-il été possible de faire mieux ? Les différentes marques de téléphones y ont réfléchi.



Figure 3.01 : Logo d'Android

Source : [24]

Pour la première fois, elles se sont mises d'accord, et de cet accord est né l'**Open Handset Alliance**, au mois de novembre de la même année. Concrètement, elle regroupait pas moins de 35 entreprises dont Google qui avaient suggéré le développement d'un système d'exploitation open-source, pour révolutionner le marché du mobile en proposant quelque chose de nouveau, et balayer la concurrence (Windows Mobile à l'époque et iOS surtout). À l'heure actuelle, le projet Android est un **grand succès** avec près de 85% de parts de marché sur le secteur des nouveaux smartphones vendus (deuxième trimestre 2014).

### **Les atouts qui font le succès du système**

Le petit robot vert surnommé « BugDroid » a aujourd'hui conquis le monde entier. Mais sans doute voudriez-vous savoir ce qui a fait la différence avec le système Android, pour connaître une croissance aussi exponentielle en quelques années seulement.

#### ***1. Le projet est open-source et gratuit***

Le gros point fort d'Android vient du fait qu'il utilise le noyau Linux et est un système libre. Contrairement à Windows ou Mac OSX, vous pouvez à tout moment consulter le code source (qui, une fois compilé, donne l'O.S.), le télécharger, l'adapter... bref, vous avez un véritable droit de regard et de modification (à vos risques et périls) sur la manière dont est fait Android et comment il fonctionne.

De plus, il est associé à un large ensemble de **bibliothèques open-source** elles aussi, mais qui ont fait leur preuve, à savoir OpenGL pour la gestion des images 2D et les environnements 3D, WebKit qui est employé pour le navigateur internet fourni avec Android (et qui a servi à développer Google Chrome, la nouvelle version d'Opera ou Safari), mais aussi SQL Lite pour la gestion des bases de données. Nous ne citerons que ces trois là, mais il en existe bien d'autres (FreeType, SSL, Surface Manager, Media Framework, System C library, etc.).

## **2. Le système est évolutif**

Le second avantage du système Android est qu'il est évolutif. Comme c'est un système ouvert, il est donc facilement portable d'un appareil à un autre : sur nos smartphones, nos tablettes mais aussi sur d'autres appareils électroniques du quotidien et même nos véhicules.

Par ailleurs, son fonctionnement même assure la possibilité de **combiner des fonctionnalités**, la preuve en est avec les services Google, ou la combinaison de l'appareil photo avec la géolocalisation par exemple, pour définir des lieux associés à vos clichés.

## **3. Le développement est accessible**

Enfin, un dernier atout à souligner est une autre conséquence du système open-source : la facilité de développement. Plusieurs APIs sont fournies en vue d'accélérer le développement. Il devient alors plus abordable d'apprendre à programmer sur ce système que sur un OS propriétaire.

## **4. Mise en œuvre de l'application sous Android**

Avec la technologie, tout est maintenant presque possible. Nous allons détailler les étapes pour faire une application android.

D'une manière générale, voici le schéma récapitulant ce que l'on va faire :

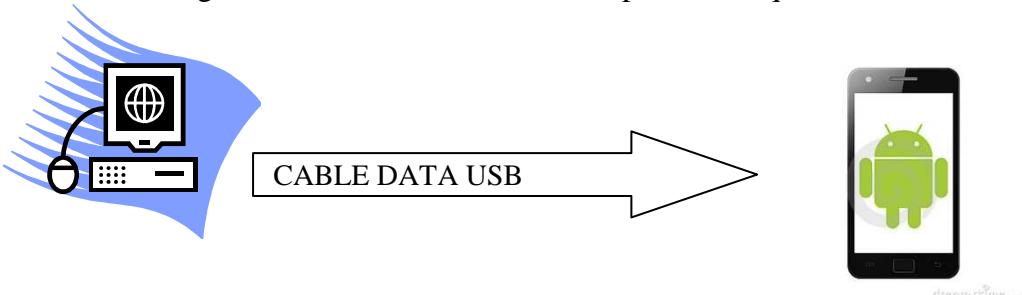


Figure 3.02 : Schéma général de transfert de l'application vers Android

Source : Auteur

D'abord, on programme l'application dans l'ordinateur à l'aide d'un IDE comme Eclipse ADT ou Android Studio. Après compilation sans erreur, le fichier « .apk » est généré par la machine virtuelle, il suffit alors de transférer ce fichier « .apk » vers l'appareil Android via le cable USB. Voici de plus amples détails :

### Etape 1 : Installation des outils

Avoir installé la JAVA DEVELOPMENT KITS puis avoir Eclipse ADT (Android Developper Tools) ou Android Studio.

Cet IDE(Integrated Development Environment) ou environnement de développement intégré est un outil simple, efficace et intègre diverses fonctionnalités pour accélérer le développement d'application Android.

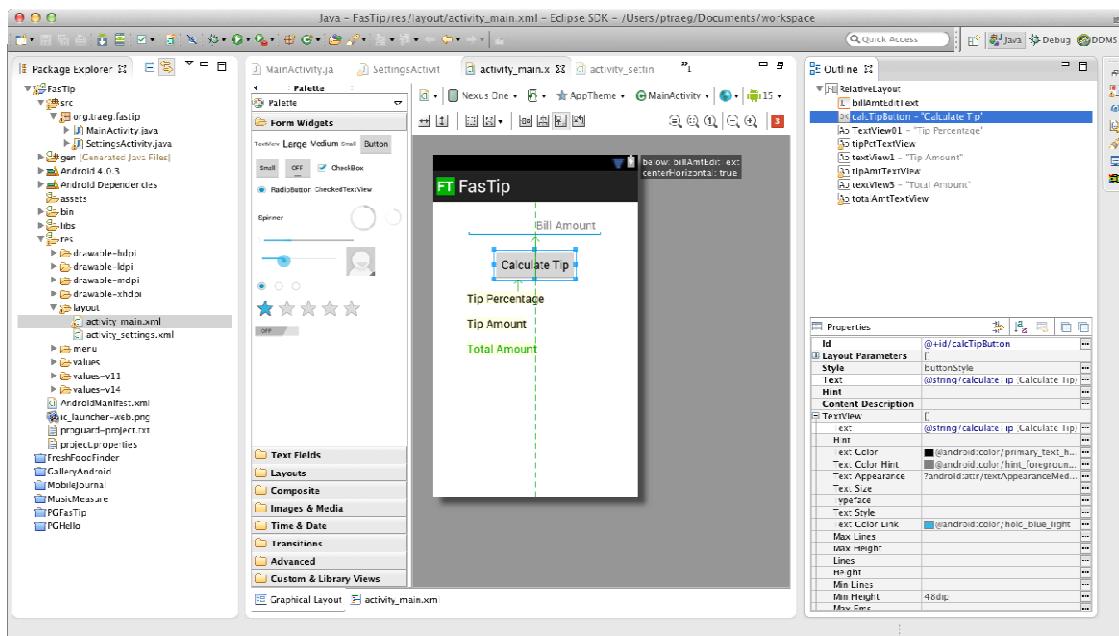


Figure 3.03 : Présentation de l'interface d'Eclipse ADT

[27]

Egalement, cet IDE intègre un émulateur Android géré par la AVD (Android Virtual Device) qui permet de visualiser l'application, les résultats qui pourraient s'afficher sur un véritable appareil Android. Cet émulateur peut être réglé suivant sa dimension, sa marque, sa capacité et bien d'autres options comme la caméra.



Figure 3.04: Présentation de l'interface de l'émulateur  
[28]

*Etape 2 : Création projet*

- Création du projet : dans Eclipse ADT, choisir « file », « new », « Android project application ».
- Nous entrerons le nom du projet avec des options qui nous intéresseraient comme les diverses versions d'Android qu'on voudraient embarquer avec notre application.
- Une fois créé, nous aurons automatiquement deux fichiers qui sont : `MainActivity.java` et « `activity_main.xml` », qui sera l'interface graphique de l'application.

*Etape 3 : Déclaration de la couche cachée sous Android, des entrées et sorties et enfin les genres.*

On va écrire le code dans le fichier « `MainActivity.java` »

Ici on va écrire que 10 nombres binaires pour limiter une trop longue écriture.

Et on ne mentionnera pas ici les genres attribués aux mots, mais juste leurs valeurs binaires.

```

public class PerceptronMulticouche {
    private static final int INPUT_NEURONS = 72;
    private static final int HIDDEN_NEURONS = 7;
    private static final int OUTPUT_NEURONS = 72;
    private static final double LEARN_RATE = 0.2;
    private static final double NOISE_FACTOR = 0.45;
    private static final int TRAINING_REPS = 10000;

    // TABLEAU DE POIDS AUX ENTRÉES AUX COUCHES CACHÉES
    private static double[][] wih[][] = new double[INPUT_NEURONS + 1][HIDDEN_NEURONS];

    // TABLEAU POIDS DES COUCHES CACHÉES A LA SORTIE
    private static double[][] who[][] = new double[HIDDEN_NEURONS + 1][OUTPUT_NEURONS];

    // TABLEAU DES ACTIVATIONS
    private static double[] inputs[] = new double[INPUT_NEURONS];
    private static double[] hidden[] = new double[HIDDEN_NEURONS];
    private static double[] target[] = new double[OUTPUT_NEURONS];
    private static double[] actual[] = new double[OUTPUT_NEURONS];

    // TABLEAU DES ERREURS
    private static double[] erro[] = new double[OUTPUT_NEURONS];
    private static double[] errh[] = new double[HIDDEN_NEURONS];

    private static final int MAX_SAMPLES = 5;

    // DECLARATION DES ENTRÉE A APPRENDRE
    private static int[][] trainInputs[][] =
        new int[][][] {{1, 1, 0, 1, 0, 1, 0, 0, 0, 0}, {1, 1, 0, 1, 1, 1, 0, 0, 0, 0}, {0, 1, 1, 1, 1, 0, 0, 1, 0, 0}, {0, 1, 1, 1, 0, 0, 0, 1, 0, 0}, {1, 0, 0, 1, 1, 1, 1, 0, 0, 1}};

    // DECLARATION DES SORTIES DESIREES
    private static int[][] trainOutput[][] = new int[][][] {{1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0}};

    // DECLARATION DES GENRES (VERBE, NOM, ARTICLE)
    List<String> anaranOlona = new ArrayList<String>();
    List<String> nn = new ArrayList<String>();
    List<String> verbz = new ArrayList<String>();
    List<String> verb = new ArrayList<String>();
    List<String> dti = new ArrayList<String>();
    List<String> dt = new ArrayList<String>();
    List<String> prp = new ArrayList<String>();
    List<String> cc = new ArrayList<String>();
    List<String> nnp = new ArrayList<String>();

```

```

        List<String>adj= new ArrayList<String>() ;
    }

Etape 4 : Calcul de la fonction sigmoïde
privatetestaticdouble sigmoid(finaldouble val)
{
return (1.0 / (1.0 + Math.exp(-val)));
}

privatetestaticdouble sigmoidDerivative(finaldouble val)
{
return (val * (1.0 - val));
}

```

*Etape 5 : Maintenant donnons le code pour déclarer la « rétropropagation »*

```

privatetestaticvoid retroPropagation()
{
// CALCUL DE L'ERREUR DE LA COUCHE DE SORTIE
for(int out = 0; out < OUTPUT_NEURONS; out++)
{
    erro[out] = (target[out] - actual[out]) *
    sigmoidDerivative(actual[out]);
}

// CALCUL L'ERREUR DE LA COUCHE CACHEE
for(int hid = 0; hid < HIDDEN_NEURONS; hid++)
{
    errh[hid] = 0.0;
    for(int out = 0; out < OUTPUT_NEURONS; out++)
    {
        errh[hid] += erro[out] * who[hid][out];
    }
    errh[hid] *= sigmoidDerivative(hidden[hid]);
}

// MISE A JOUR DES POIDS
for(int out = 0; out < OUTPUT_NEURONS; out++)
{
    for(int hid = 0; hid < HIDDEN_NEURONS; hid++)
    {
        who[hid][out] += (LEARN_RATE * erro[out] * hidden[hid]);
    }
    who[HIDDEN_NEURONS][out] += (LEARN_RATE * erro[out]);
}

//MISE A JOUR DES POIDS SUR LES COUCHES CACHEES
for(int hid = 0; hid < HIDDEN_NEURONS; hid++)
{
    for(int inp = 0; inp < INPUT_NEURONS; inp++)
    {
        wih[inp][hid] += (LEARN_RATE * errh[hid] * inputs[inp]);
    }
    wih[INPUT_NEURONS][hid] += (LEARN_RATE * errh[hid]);
}
return;
}

```

*Etape 6 : Apprentissage du Perceptron multicouche*

```

privatetestaticvoid ReseauDeNeurone( )

// APPRENTISSAGE
for(int epoch = 0; epoch <TRAINING_REPS; epoch++)
{
    sample += 1;
if(sample == MAX_SAMPLES){
        sample = 0;
    }

for(int i = 0; i <INPUT_NEURONS; i++)
{
    inputs[i] = trainInputs[sample][i];
}

for(int i = 0; i <OUTPUT_NEURONS; i++)
{
    target[i] = trainOutput[sample][i];
}

retroPropagation();
}

}

```

*Etape 7 : Erreur proche de 0*

Quand l'erreur est proche de 0, on arrête tout de suite l'apprentissage.

*Etape 8 : Interface graphique*

L'interface graphique Android est un fichier XML qu'on peut retrouver dans le dossier « res », « layout » puis « activity\_main.xml ».

La technique du « Glisser-déposer » est disponible pour faciliter la mise en place des composants graphiques (bouton, champ de texte,...).

Le code XML est généré automatiquement et doivent être relié au code du MainActivity.java.

Etape 9 : Voici le code Android pour traduire le mot en français

```
//DECLARATION D'UNE NOUVELLE INSTANCE DE TTS
```

```
t1=new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {
@Override
```

```

//INITIALISATION DE LA BIBLIOTHEQUE VOCALE

public void onInit(int status) {

    if(status != TextToSpeech.ERROR) {

        t1.setLanguage(Locale.FR);

    }

}

//DES QU'ON CLIQUE SUR LE BOUTON

b1.setOnClickListener(new View.OnClickListener() {

    @Override

    //TRADUCTION EN FRANCAIS

    public void onClick(View v) {

        String toSpeak = ed1.getText().toString();

        Toast.makeText(getApplicationContext(),
        toSpeak,Toast.LENGTH_SHORT).show();

        t1.speak(toSpeak, TextToSpeech.QUEUE_FLUSH, null);

    }

});

```

#### *Etape 10 : transfert de l'application sur un véritable appareil Android*

Après compilation du programme, et qu'il n'y a plus d'erreurs, nous aurons un fichier « .apk » qu'on retrouve dans le dossier « bin » du projet en cours.

Il suffira alors de connecter son appareil en tant que stockage de masse puis transférer le fichier .apk en question et de l'installer comme toute application standard.

Cependant, une autre option existe.

Si votre appareil Android possède un pilote et qu'il a été bien installé, vous pouvez compiler directement votre application sur l'appareil sans passer par l'émulateur intégré.

De ce fait, l'application se lancera tout de suite dans votre appareil via le cable USB.

#### **Présentation de l'interface graphique utilisateur**

L'utilisateur aura cette interface simple :

Un champ de texte pour insérer les phrases de test. Un bouton juste en bas pour donner la phrase correcte et sa traduction en français version vocale.

#### **Imageréultante des tests**

L'application que nous avons développé est bien loin d'être parfaite mais quand même permet d'exécuter quelques exemples de mots et de phrases « Malagasy » suivis de sa traduction vocale en français ou en d'autres langues.

Espérons que les « Malagasy » pourront bénéficier de cet outil qui sera disponible pour tous.

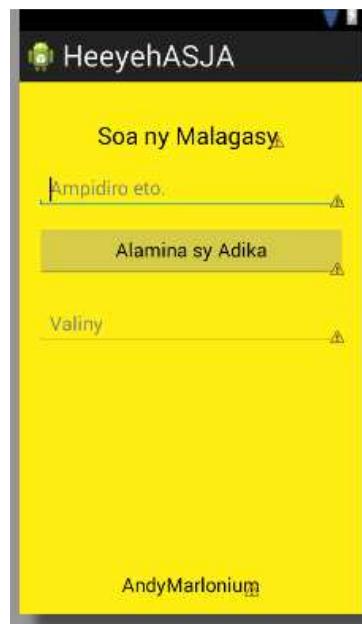


Figure 3.05 : Présentation de l'interface de l'application

Source : Auteur

## AMELIORATIONS

Ce système peut être amélioré grâce à d'autres réseaux de neurones artificiels comme le modèle de Hopfield.

Avec ce dernier, on peut corriger l'orthographe d'un mot mal écrit à partir des connaissances que le réseau a déjà appris.

Exemple : « Misaottra », va être corrigé en « Misaotra ».

Outre, le modèle de Kohonen est également très utile puisqu'on pourra classifier les mots, les phrases après quelques apprentissages.

Exemple : Baolina – kilalao – fety = “fialam-boly”.

N'oublions pas aussi qu'il n'existe pas encore de nos jours une application d'intelligence artificielle tournant sous Android traitant la langue Malagasy.

Beaucoup de perspectives sont donc envisageables comme la reconnaissance vocale, l'utilisation ou la commande d'un système complexe (automatique, électronique) en langage Malagasy.

## **CONCLUSION**

D'après tout ce qu'on a vu, beaucoup de choses sont possibles actuellement grâce à la nouvelle technologie mobile combinée aux modèles mathématiques qui s'illustrent à travers l'intelligence artificielle couvrant les réseaux de neurones artificiels.

Les sciences cognitives sont en cours d'exploitation à Madagascar et bon nombre de projets sont encore faisables à l'aide de ces outils d'analyses et décisionnels efficaces.

De plus, Android est un système d'exploitation libre, donc on peut tout faire sans limitation.

A part cela, de nouveaux appareils Android voient le jour pour faire des calculs encore plus rapide pour pouvoir faire des traitements lourds.

On pourrait dire qu'Android est l'un des outils performants du futur surtout dans le domaine des sciences cognitives car il est mobile, à la portée de tous, simple d'utilisation.

## **BIBLIOGRAPHIE**

[Référence n°29]=[Rosenblat, Edition Vuibert, F(1962) –[121], p.3

[Référence n°30]=[Haykin, Edition Dunod, S.(1999) – [60], p.50]

[Référence n°31]=[Minsky, M., & Papert, Edition Dunod, S(1988), p.232]

[Référence n°32]= Nicolas P. Rougier, 2011, *Perceptron simple Perceptron multi-couches*

[Référence n°32]= Marc Parizeau, 2011, *Le perceptron multicouche et son algorithme de retropropagation des erreurs.*

[Référence n°33]= GeraldPetitjean, 2011 *Cours réseaux neurones.*

[Référence n°34]= Claude Touzet 1992, *Les réseaux de neurones artificiels.*

## **WEBOGRAPHIE**

[Référence n°1]=[<http://www.idixa.net/Pixa/pagixa-0512121349.html>]

Diff : 05/04/2014, Consult : 12/01/16

[Référence n°2]=<http://www.ac-grenoble.fr/PhiloSophie>

Diff : 12/10/2012, Consult : 20/01/16

[Référence n°3]=[<http://www.philocours.com/cours/cours-langage.html>]

Diff : 07/05/2011, Consult : 07/01/16

[Référence n°4]=[<http://www.universalis.fr/encyclopedie/cerveau-humain/>]

Diff : 07/02/2011, Consult : 10/01/16

[Référence n°5]=[<http://www.lecorpshumain.fr/corpshumain/1-cerveau.html>]

Diff : 03/08/2014, Consult : 17/12/15

[Référence n°6]=<http://icm-institute.org/fr/actualite/comprendre-le-cerveau-et-son-fonctionnement>]

Diff : 08/03/2011, Consult : 17/12/15

[Référence n°7]=<http://www.lecorpshumain.fr/corpshumain/1-cerveau.html>

Diff : 07/08/2010, Consult : 17/12/15

[Référence n°8]=<http://www.grappa.univ-lille3.fr/polys/apprentissage/sortie005.html>

Diff : 04/02/2012, Consult : 19/12/15

[Référence n°9]=[www.psychomedia.qc.ca](http://www.psychomedia.qc.ca)

Diff : 08/07/2011, Consult : 19/12/15

[Référence n°10]=<http://ameliorersonfrancais.com/grammaire/varia/h-muet-ou-aspire/>

Diff : 06/05/2014, Consult : 19/12/15

[Référence n°11]=[[http://grammaire.cordial-nligne.fr/manuels/ACCO\\_VERB.htm](http://grammaire.cordial-nligne.fr/manuels/ACCO_VERB.htm)]

Diff : 07/02/2012, Consult : 19/12/15

[Référence n°12]=<http://madanydago.skyrock.com/1073675322-L-origine-de-la-langue-malgache.html>

Diff : 25/01/2011, Consult : 19/12/15

[Référence n°13]=<http://www.madagascar-island.com/langue-malgache/>

Diff : 05/04/2010, Consult : 22/12/15

[Référence n°14]=<http://terangaweb.com/histoire-de-la-langue-malagasy/>

Diff : 17/07/2011, Consult : 22/12/15

[Référence n°15]=<http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/informatique-reseau-neuronal-601/>

Diff : 06/05/2008, Consult : 22/12/15

[Référence n°16]=<http://alp.developpez.com/tutoriels/intelligence-artificielle/reseaux-de-neurones/>

Diff : 05/06/2010, Consult : 22/12/15

[Référence n°17]=[fr.wikipedia.org](http://fr.wikipedia.org)

Diff : 04/07/2012, Consult : 22/12/15

[Référence n°18]=<http://william.arrouy.free.fr/neural/neu2.html>

Diff : 18/08/2011, Consult : 22/12/15

[Référence n°19]=<http://www.grappa.univ-lille3.fr/~gilleron/PolyApp/node19.html>

Diff : 01/02/2007, Consult : 22/12/15

[Référence n°20]=<http://lcn.epfl.ch/tutorial/french/hopfield/html/index.html>

*Diff : 24/01/2009, Consult : 24/12/15*

*[Référence n°21]=<http://william.arrouy.free.fr/neural/neu4.html>*

*Diff : 24/02/2010, Consult : 24/12/15*

*[Référence n°22]=<http://bms.revues.org/829#tocto1n2>*

*Diff : 17/08/2007, Consult : 24/12/15*

*[Référence n°23]=[http://sapience.dec.ens.fr/cogmaster/www/f\\_01\\_presentation.php](http://sapience.dec.ens.fr/cogmaster/www/f_01_presentation.php)*

*Diff : 04/02/2010, Consult : 30/12/15*

*[Référence n°25]=<http://www.futura-sciences.com/magazines/high-tech/infos/dossiers/d/telecoms-histoire-telephone-portable-annees-80-nos-jours-1944/>*

*Diff : 05/07/2014, Consult : 30/12/15*

*[Référence n°24]=[blog.xebia.fr](http://blog.xebia.fr)*

*Diff : 02/04/2013, Consult : 30/07/15*

*[Référence n°26]=[beetelbite.com](http://beetelbite.com)*

*Diff : 08/05/2012, Consult : 30/09/15*

*[Référence n°27]=[www.smashingmagazine.com](http://www.smashingmagazine.com)*

*Diff : 11/07/2010, Consult : 30/12/15*

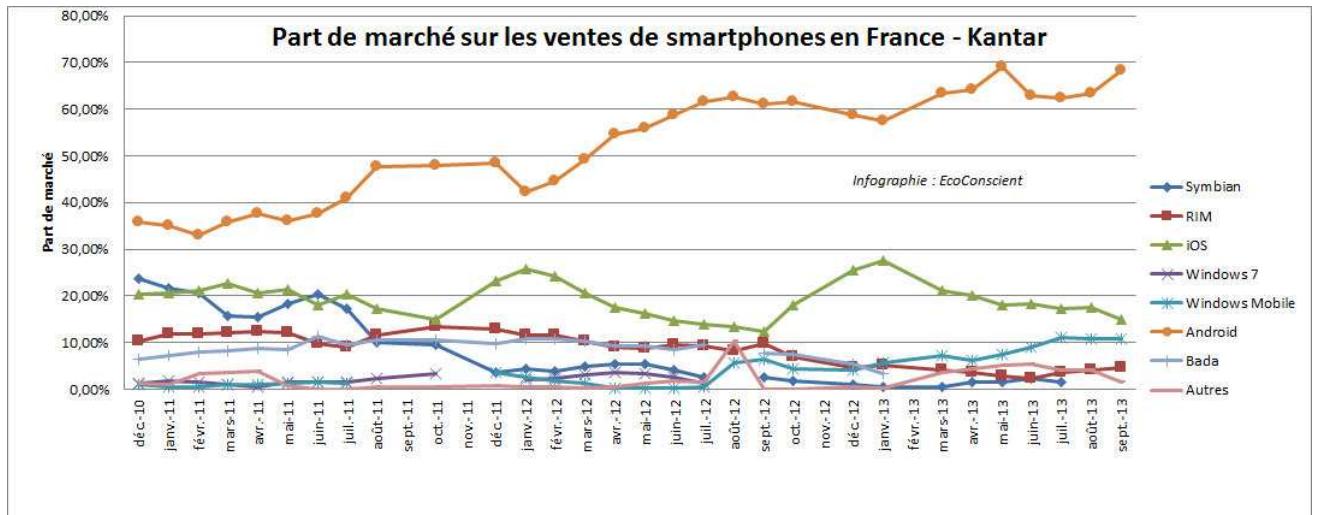
*[Référence n°28]=[www.williambrownstreet.net](http://www.williambrownstreet.net)*

*Diff : 04/09/2011, Consult : 30/10/15*

# **ANNEXES**

## ANNEXES

### *Les parts de marché en fin 2010 à 2013*



### **L'historique des versions d'Android, une véritable évolution**

Le système de Google n'aurait pas connu un tel succès s'il était resté le même en six ans. C'est là que l'on voit la puissance d'un tel OS qui a su s'adapter aux besoins des utilisateurs à chaque version majeure et qui s'enrichit de nouveautés.

Que l'on adhère ou non au concept, on ne peut que saluer les efforts menés pour **améliorer l'expérience des utilisateurs**, et conserver qui plus est sa gratuité. Nous vous proposons de (re)découvrir les débuts du système, et ce qu'il est désormais.

#### **Android 1.0 et 1.1**

La grande innovation pour cette première version d'Android fut la barre de notifications, au travers d'un menu déroulant qui regroupe plusieurs affichages pratiques et rapidement accessibles. Ce fut aussi l'apparition des widgets et de l'ancêtre du [Google Play Store](#), l'Android Market.

#### **Android 1.5**

La mise à jour 1.5 « **CupCake** » a été publiée le 30 avril 2009, améliorant l'interface graphique de Android. À partir de celle-ci, il a été décidé de nommer les différentes versions du système avec des noms de gâteaux (peut-être que les entreprises étaient gourmandes), en suivant l'ordre alphabétique.

## Android 1.6

Si la version 1.5 était une mise à jour majeure, celle qui a suivi l'était moins. Android 1.6 « **Donut** » offrait tout de même quelques nouvelles fonctionnalités comme la prise en charge du réseau CDMA, le support d'autres définitions d'écran.

Auparavant, il ne fallait compter que sur une définition maximum de 480 x 320 pixels (en mode portrait) et la mise à jour 1.6 aura permis de gérer toutes les **définitions allant jusqu'à la HD**, avec une définition de 1280 x 720 pixels, un grand pas en avant pour le système.

## Android 2.0 et 2.1

Celle-ci aura été majeure et aura apporté beaucoup de fonctionnalités intéressantes, comme les suivantes :

- Amélioration du clavier tactile
- Fonds d'écran animés
- Prise en charge du Bluetooth 2.1
- Arrivée de **Google Maps**
- Nouvelles fonctions pour l'appareil photo (flash, zoom digital, balance des blancs, effets de couleurs et focus macro)

La **fonction « speech-to-text »** fait également son entrée et introduit la synthèse vocale au sein du système Android.

## Android 2.2 et 2.3

Le 20 mai 2010 arrivait la version Android 2.2 « **Froyo** », qui a profité au Nexus One tout d'abord (Google oblige) et ce fut à nouveau de belles améliorations pour le système open-source.

De nouveaux raccourcis ont été ajoutés aux côtés du **lanceur d'applications** : le téléphone et le navigateur natif. La mise à jour proposait également du nouveau pour la galerie avec une meilleure interface pour regarder vos photos, des effets de transition, ce qui rendait l'O.S. encore plus convivial.

C'est une version particulière qui a apporté le support de Flash Player 10.1, la prise en charge des écrans ayant une densité de pixels de 320 ppi et l'ajout du fameux écran de verrouillage à mot de passe ou code PIN.

Et arriva ce qui arriva, le 6 décembre de la même année, Google publie la version 2.3 « **Gingerbread** ». Elle n'aura pas apporté grand chose, mais surtout des améliorations sur la prise en charge et le support de la VoIP, le NFC, le capteur frontal sur les appareils concernés et un passage au système de fichiers ext4 que les amateurs de GNU/Linux connaissent fort bien.

## Android 3.0

Dans toute l'histoire de l'OS, il existe une mise à jour, la 3.0 « **Honeycomb** », qui fut réservée aux [tablettes Android](#). À cette époque, la société Motorola détenait l'exclusivité avec son smartphone Droid et la version 2.0 du système. La [tablette Xoom](#) est née avec cette mise à jour Honeycomb et nous avons pu découvrir le nouveau virement de bord de la part de Google sur l'interface.

Celle-ci arborait des couleurs similaires aux versions actuelles : une dominance de bleu. Fini le vert, et terminés les boutons physiques « **Retour** » et « **Menu** », devenus entièrement tactiles pour l'occasion.

Le multitâche fut également à l'honneur dès l'arrivée de cette mise à jour grâce à Matias Duarte devenu designer chez Google pour repousser les limites de l'expérience utilisateur.

Du côté des mises à jour suivantes, la 3.1 et la 3.2, on aura apprécié de pouvoir enfin **redimensionner les widgets** sur ses pages, mais également le support du format FLAC (pour les amoureux de la musique) et la prise en charge des [processeurs Qualcomm](#) et des tablettes tactiles de 7 pouces.

## Android 4.0

C'est le 19 octobre 2011 que l'histoire d'Android va atteindre son apogée (ou presque). L'arrivée de la version 4.0 « **Ice Cream Sandwich** », compatible avec les smartphones et les tablettes, nous laisse rêveur avec beaucoup de nouveautés tant attendus par les utilisateurs, et quelques unes que je vous cite ci-dessous :

- Ajout de dictionnaires dans le clavier virtuel
- Application photo améliorée, rapidité de capture quasiment en mode rafale
- Capture d'écran native
- Consultation de son trafic data, pour ne plus avoir de hors forfait à votre prochaine facture
- Création plus facile de dossiers d'applications
- Écran de verrouillage repensé, sous forme d'un cadenas qui permet de lancer des applications en raccourcis
- Mise à jour du navigateur (ouvertures de plusieurs pages, navigation privée)
- Prise en charge de l'enregistrement vidéo Full HD 1080p

## Android 4.1, 4.2, 4.3 et 4.4

« **Jelly Bean** » est le dernier nom d’Android, annoncé le 27 juin 2012 lors du Google I/O. Le premier appareil équipé de la mise à jour 4.1 fut la Nexus 7, la tablette de Google commercialisée à partir du 13 juillet 2012 aux Etats-Unis.

C’est une mise à jour qui fut très attendue, et pour preuve, à l’heure actuelle, nos appareils sont encore équipés d’une version Jelly Bean de l’O.S. D’abord, le « **Project Butter** » a triplé la vitesse de rafraîchissement du système pour atteindre 16 millisecondes, un record. Le système d’Apple avait donc bien du soucis à se faire, n’étant plus le seul à être aussi réactif. La firme de Mountain View a également refait parler d’elle avec la venue de [Google Now](#), un véritable service qu’on pourrait rapprocher de Siri sur iOS.

## Android 4.2

On aura remarqué la nouveauté du clavier tactile, avec la [fonction Swipe](#) permettant de glisser son doigt sur ce dernier pour écrire, un moyen très rapide pour envoyer vos SMS. En ce qui concerne les tablettes, elles pouvaient enfin profiter d’un [système multi-utilisateur](#) pour diversifier l’usage de votre appareil au sein de la famille par exemple.

Sinon, la version 4.2 a également apporté la fonction [Photosphere](#) au sein de l’application photo dans le mode panorama, permettant de réaliser des clichés sphériques.

Finalement, le 24 juillet 2013 sera la date de la dernière mise à jour Android 4.3 qui est disponible sur de très nombreux appareils grâce à la mise à jour en OTA.

En plus de ça, OpenGL ES 3.0 a été ajouté au système et permet la conception d’applications générant des images 3D mais ciblant les terminaux mobiles, ainsi que l’auto-complétion lors du remplissage de formulaires.

De même, si vous avez suivi l’actualité, le **dongle HDMI de Google** ne vous sera pas inconnu. [Chromecast](#) permet en effet de streamer du contenu multimédia, directement sur son téléviseur.

Avec Android KitKat, la fameuse version 4.4, Google se renforce encore sur la branche 4.X. Le système est annoncé le 31 octobre 2013 d’abord sur le tout nouveau Nexus 5 puis sera déployé sur les autres terminaux. Il s’agit encore aujourd’hui de l’**une des versions de Android les plus couramment utilisées**.

Autre différence d’ordre graphique que vous pouvez apprécier sur les captures. Le style Holo, très bleu, à la Tron laisse sa place pour un **style bien plus épuré** au niveau des indications système. Ces dernières préfigurent la ligne qu’adoptera Android 5 Lollipop. Les icônes

concernées deviennent blanches et un léger dégradé en fond permet de les garder visibles en toutes circonstances.

L'autre cheval de bataille, Google Now, subit une **intégration plus poussée** au sein du système. L'idée est de le rendre de plus en plus indispensable, de plus en plus omnipotent. En se rapprochant de l'assistant personnel « virtuel », Google esquisse ici les lignes des futures versions de Android.

### **Android 5.0, 5.0.1, 5.0.2 et 5.1**

Android 5.0 Lollipop a été officiellement nommé ainsi le **15 octobre 2014** par Google. De nombreuses mises à jour sont depuis en cours vers cette nouvelle version majeure qui fut disponible publiquement le 3 novembre de la même année pour les appareils Nexus dans un premier temps. Les [Nexus 6 et 9 en furent d'ailleurs équipés](#) dès leur commercialisation.

Premier changement qui saute aux yeux, le **Material Design**. Google a décidé de revoir tout l'habillage graphique de son système d'exploitation. Fini l'habillage Holo introduit avec la version 4.0, Material design fait la part belle aux couleurs et au flat design, soit un rendu au visuel plus « plat ». La sécurité est aussi au centre des occupations du système et le noyau a été largement revu en ce sens.

Autre innovation à suivre, le **Projet Volta**. Avec lui, Google poursuit sur le chantier de la consommation énergétique. Ainsi c'est un tout nouveau panel d'API lancées à destination des développeurs afin de gagner en autonomie globale sur l'appareil. L'idée est ainsi de réveiller plus intelligemment le smartphone ou la tablette pour les tâches en arrière-plan.

Enfin le dernier gros changement concerne le runtime du système. Ainsi Lollipop marque l'abandon de la machine virtuelle Dalvik [au profit de ART \(Android Runtime\)](#). Avec ce dernier, on nous promet de **meilleures performances** au niveau général du système. Que de bonnes choses donc pour les utilisateurs qui ont déjà adopté Android 5.0 (ou supérieur) !

### **Android 6.0**

Elle n'est pas encore officiellement disponible mais certains ont déjà fait sa connaissance par le biais de la Developers Preview. Cette nouvelle version d'Android symbolisera tout d'abord le passage à une nouvelle unité mais aussi à une nouvelle friandise. Un changement assez radical pour une version qui était supposée originellement être la 5.2.

Le surnom de [Marshmallow a été officialisé le 18 août 2015](#). Côté nouveautés apportées, Google a pour l'instant souligné 6 fonctions principales qui porteront Android 6.0 mais en réalité, rien que la version développeur en comptabilise déjà une cinquantaine, que ce soit par

des ajouts de fonctionnalités inédites ou d'améliorations de certaines déjà existantes.



## **RESUME**

L'intelligence artificielle participe de mieux en mieux dans la vie courante et devient de plus en plus un domaine familier à tous.

Effectivement, issue de l'intelligence humaine qui est caractérisée par des réseaux de neurones interconnectés, l'intelligence artificielle tente de reproduire ces capacités surprenantes de l'Homme à l'aide de modèles mathématiques divers.

Les sciences cognitives regroupent tous ces domaines reproduisant, imitant toutes les fonctionnalités cérébrales avec l'informatique qui est incontournable pour ces rapides calculs.

Ici, nous allons nous cerner sur le « Langage ». La langue Malagasy sera le thème.

Le perceptron multicouche va arranger correctement l'ordonnancement d'une phrase quelconque entrée en désordre, ensuite, un deuxième perceptron multicouche assurera la traduction de ce mot en français et finalement la bibliothèque Google Text To Speech prononcera ce mot en mode vocale.

Des apprentissages supervisés se feront à travers des itérations, en manipulant des entrées et sorties binaires en passant par des couches cachées.

La technologie Android via un smartphone soutiendra toutes ces manipulations avec ses facultés réputées avancées.

Mots clés : Sciences cognitives, intelligence artificielle, réseaux de neurones, Android, perceptron multicouche, poids, Malagasy, entrée, sortie

## **ABSTRACT**

Artificial intelligence is involved in getting better everyday and becoming more familiar for all.

Indeed, after the human mind which is characterized by interconnected neural networks, artificial intelligence attempts to replicate these amazing capabilities of the human using various mathematical models.

Cognitive science together all these areas reproducing, imitating all brain functions with computers which is a must for those quick calculations.

Here we will identify us on the "Language". The Malagasy language is the theme.

Multilayer Perceptron will properly arrange the scheduling of any sentence entered, then a second multilayer perceptron will translate this word in French and finally the Google Text To Speech library pronounce that word in voice mode.

Learning will be supervised through iterations, by manipulating binary inputs and outputs through the hidden layers.

The Android technology via a smartphone will support all these manipulations with its famous advanced faculties.

## **RENSEIGNEMENTS SUR L'AUTEUR**

Nom : BOURGEON

Prénoms : ANDY MARLON

Date de naissance : 30 Avril 1978 à Tsarahononana Itaosy

Téléphones : 0343964417 – 0330243717

Mail : [andymarlonium@gmail.com](mailto:andymarlonium@gmail.com)

Adresse : LOT ITR 63 BIS ITAOSY

### **Cursus**

2016 : DOCTORANT EN SCIENCES COGNITIVES ET APPLICATIONS A L'ESPA

2016 : MASTER II A VISEE DE RECHERCHE EN SCIENCES COGNITIVES ET APPLICATIONS A L'ESPA

2006 : MASTER II EN INFORMATIQUE – GENIE INDUSTRIEL ASJA

2004 : LICENCE PROFESSIONNELLE EN INFORMATIQUE ASJA

Titre : ORDONNANCEMENT DES PHRASES MALAGASY AVEC LEURS TRADUCTIONS FRANCAISES VOCALES.

Nombre de pages : 64

Nombre de figures : 24