Université Paris 8

BCI : Detection et classification des activités cérébrales

Rapport I projet tuteuré

Auteurs :
Nizar ABAK-KALI
11290569
Alexis
ZURAWSKA
12310497

Tuteur :
M. Larbi BOUBCHIR

19 janvier 2016

Table des matières

Introduction

Durant notre première année de master en informatique à l'université Paris 8 Vincennes - Saint-Denis, nous devions développer des applications au cours d'un module intitulé "Projets tuteurés". Ce travail serait effectué sur toute l'année et doit faire l'objet d'une présentation en plus de ce rapport-ci. Nous devions choisir une thématique parmi celles proposées. Ensuite, notre tuteur attitré nous attribuait un projet en nous expliquant ce qu'il attendait de nous. En ce qui nous concerne, mon binôme et moi, nous devons développer un réseaux de neurones capables de reconnaître différents états de stimulation d'une personne atteinte de la maladie d'Alzheimer. C'est après vous avoir présenté l'ensemble de nos lectures sur l'utilité et l'implémentation des réseaux de neurones et des interfaces cerveaux-machines que nous vous présenterons l'architecture de notre application, ce après quoi nous évoquerons les résultats que nous attendons de ce projets et les perspectives futures de développement.

Première partie

Introdution : généralité et contexte

1 BCI

BCI : Brain Computer Interface (ou Interface Cerveau-Machine en français). Aussi appelée Interface Neuronale Directe (abrégée IND), il s'agit d'une interface de communication directe entre un cerveau et un composant externe (généralement un ordinateur). Ce procédé consiste en fait à restaurer, au moins partiellement des facultés perdues, voire non connues (dans le cas d'un handicap de naissance). C'est le cas notamment d'une personne ayant perdu totalement la vue à qui on a implanté un tel procédé au niveau de son cortex visuel, ce qui lui a permis de percevoir de nouveau la lumière. Si ces procédés ne sont pas miraculeux pour le moment, ils peuvent également permettre d'effectuer des actions par l'intermédiaire de la pensée. Ainsi, des personnes ont pu écrire sur un écran d'ordinateur ou déplacer le curseur d'une souris en imaginant simplement l'action de le faire. D'autres ont pu déplacer un bras robotisé pour qu'il leur ramène quelque chose par exemple. Ce procédé peut être unidirectionnel (la machine envoie des données au cerveau ou le contraire mais pas les deux en même temps) ou bidirectionnel. Ces interfaces neuronales directes présentent toutefois plusieurs limites :

- chacune d'elles ne fait qu'une tâche précise et non plusieurs ;
- elles étaient initialement développées dans un but médical rendant l'accès difficile au grand public ;
- à cela s'ajoute le fait que les populations défavorisées n'auront pas les moyens de se procurer de tels équipements, le prix étant très élevé.

2 EEG

EEG ou électroencéphalographie est technique de mesure de l'activité électrique du cerveau mesurée à l'aide d'électrodes placées sur le cuir chevelu. L'EEG est un examen indolore est non-invasif comparativement à l'iEEG(électroencéphalographie intracrânienne) qui elle place les électrodes sous la surface du crâne. le signal EEG obtenue est la résultante de la sommation d'un potentiel d'action post-synaptique synchrone issus d'un grand

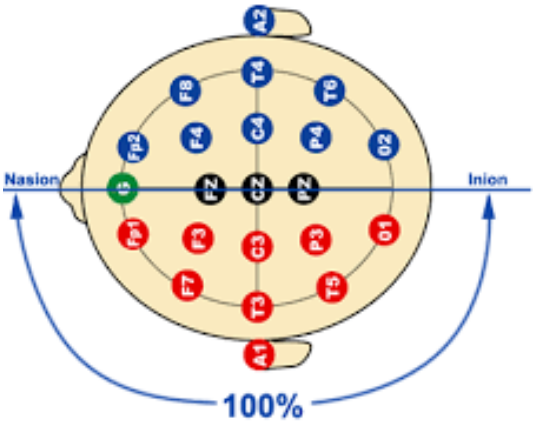

FIGURE 1 – positon des capteurs

nombre de neurones.

3 Généralité

Ici, certains concepts clés doivent être définis pour comprendre en quoi consiste notre projet :

3.1 Les réseaux de neurones

premièrement les réseaux de neurones. D'abord, il est bon de voir un réseau de neurones artificiels comme un modèle de calcul représentant schématiquement les réseaux de neurones biologiques. C'est à celui-ci que l'on va imposer l'apprentissage d'un échantillon de données. Pour cela, nous allons utiliser l'algorithme du gradient, le gradient étant, en mathématiques, un vecteur qui représente la variation d'une fonction en fonction de la variation de ses paramètres. Tout comme nous, le réseau de neurones n'est pas parfait et fera des erreurs lors de son apprentissage. Il faudra donc utiliser la rétropropagation du gradient qui est un algorithme servant à corriger les erreurs du réseau de neurones pour qu'il ne lest reproduise pas.

3.2 Notions en liaison avec BCI

Certaines notions sont importantes à connaître lorsque l'on s'intéresse au BCI, entre autre le SSVEP et le P300. ce sont tout les deux des réponses naturels à une stimulation visuelle. Pour le SSVEP, la réponse est détectée lors d'une stimulation visuelle lorsque la rétine est excitée par une stimulus visuel dont la fréquence d'affichage est entre 3.5Hz et 75Hz. Le P300 par contre est une réponse qui apparaît 300ms .

4 Contexte

Dans le cadre de ce projet, nous devions développer un réseau de neurones afin de contrôler automatiquement les activités cérébrales d'un sujet. Ainsi, le système pourrait nous dire grâce au signal EEG du patient quelle activité il est entrain de faire ou bien quelle son état. Notre système pourrait donc par exemple connaître l'activité d'un malade d'Alzheimer tel que :

- la lecture ;
- l'attention sur un film ;
- l'écriture ;
- la discussion ;
- le repos.

Deuxième partie

Etat de l'art

5 SSVEP

6 P300

7 BCI

8 Application Médical

Troisième partie

Classification automatique des activités cérébrales

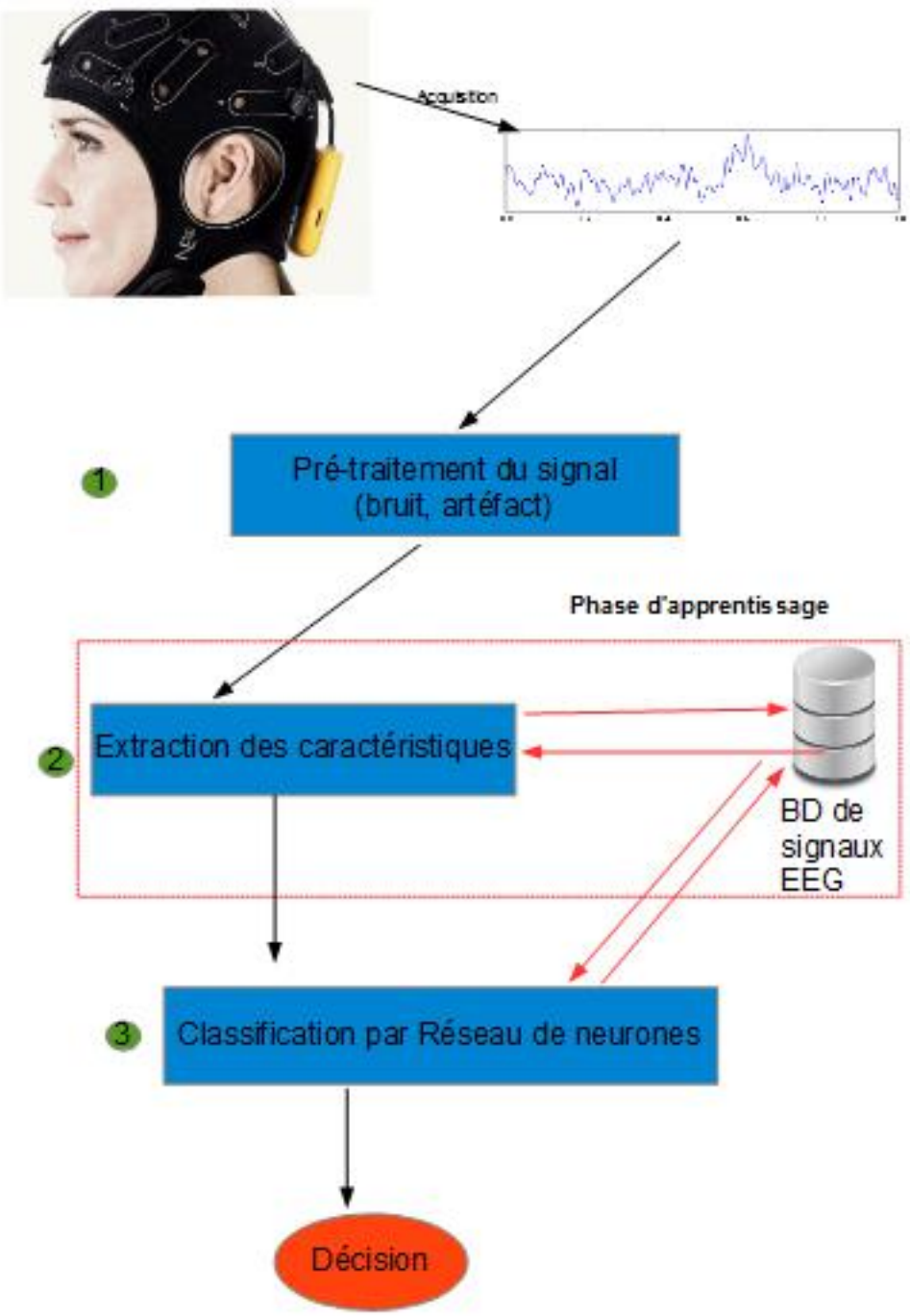

FIGURE 2 – Organigramme du fonctionnement générale de l'application

Quatrième partie

Architecture de l'application

13 Organigramme

cf Annexes .

14 Composant de l'application

cf. Annexes pour les diagrammes de classes.

L'application est composée de cinq packages : eeg, math, neuron, charts et donnees. Nous allons donc décrire le contenu de chacun de ces packages ainsi que le rôle des classes qui les composent :

- eeg : contient la classe EEG_time_signal qui contiendra les fonctions concernant le traitement du signal en fonction des données reçues et traitées ;
- math : contient toutes les fonctions mathématiques avec la classe ComplexNumber permettant de créer et gérer les nombres complexes, la classe ComplexArray qui permet de stocker les valeurs réelles et imaginaires des nombres complexes, la classe Hilbert qui contient les fonctions concernant les espaces d'Hilbert (extension des espaces euclidiens à des dimensions finies quelconques ou infinies) et enfin la classe Fft qui contient les fonctions en rapport avec la Transformée Rapide de Fourrier(ou Fast Fourrier Transform) ;
- neuron : contient la classe CaracteristiqueTemporel qui contient toutes les fonctions ayant attrait aux caractéristiques temporelles d'un neuronne pour le réseau qui sera par la suite créé. Ces caractéristiques temporelles regroupent notamment la moyenne des données, la variance, etc... ;
- charts : contient la classe Graphique afin de générer les graphiques une fois les résultats obtenus à partir des données grâce à la librairie JFreeCharts ;
- donnees : contient la classe RecupFichier permettant de récupérer la base de données pour la stocker en mémoire afin de permettre au programme de travailler avec.

15 Explication Technique

16 Idéés

Nous avons eu l'idée, pour ce projet, de créer un "classifieur" représenté par un réseau de neurones afin d'obtenir des résultats à partir d'un flux de données que nous stockerons en mémoire dans un tableau. Chaque donnée sera représentée par un vecteur. Pour chaque vecteur, chacune de ses dimensions représentera une caractéristique du signal EEG de la personne atteinte que nous chercherons à reconnaître (le tout afin de reconnaître les états tel : lecture, repos, regarder un film, etc...). Nous sommes partis de ces idées là car, lors de nos lectures sur les interfaces cerveau-machine, il s'est avéré que ce mode de traitement du signal est très utilisé .

17 Outils

Nous avons développé notre application au moyen du langage Java et nous avons utilisé l'IDE Intellij qui est très pratique car très ergonomique et très intuitif. Deplus nous avons utilisé Git comme gestionnaire de version, le code étant hébergé sur les serveurs de Github. De surcroît nous avons utilisé JavaDoc pour générer un documentation du code. Par ailleurs, nous avons inclus la librairie JFreeCharts pour réaliser des graphiques sans les contraintes de la bibliothèque Swing pour créer des interfaces ainsi que l'extension LaTex afin de rédiger plus facilement nos rapports directement au sein de l'IDE. Par ailleurs, nous avons utilisé le logiciel argoUML pour réaliser le diagramme de classes de l'application et OpenOffice Draw pour réaliser l'organigramme.

18 Algorithmes

Nous allons utiliser l'algorithme des k-means ou k-moyennes afin de réaliser notre application d'analyse des états d'une personne atteinte de la maladie d'Alzheimer. Celui-ci consiste, à partir d'un ensemble de données, un ensemble de paramètres statistiques tels que la moyenne, la variance, le mode, la médiane, etc... Par ailleurs, il sera important de calculer la distance euclidienne entre ces données qui seront représentées par des vecteurs. Ainsi, plus la distance entre ces vecteurs sera petite, plus ces données seront rapprochées afin de répartir en "petits tas" les données qui sont proches. Cela permettra

ensuite au réseau de neurones créé d'apprendre de manière supervisée (on donne la réponse en cas d'erreur jusqu'à ce que le réseau de neurones trouve tout seul) l'ensemble des données et de pouvoir les classifier ensuite afin de donner des résultats cohérents sur ces données.

Cinquième partie

Conclusion

19 Objectif atteint

20 Objectif restant

Pour conclure, nous pouvons dire que ce projet est très intéressant car il nous a permis de découvrir des domaines concrets insoupçonnés où l'informatique pouvait intervenir afin de faciliter la vie, notamment des personnes lourdement handicapées. Nous n'avons pas encore fini mais pensons vite arriver à nos fins si nous travaillons régulièrement afin de vous fournir un projet fonctionnel lors de la deuxième soutenance.

Référence

Dans cette partie nous exposons les diffenrentes sources que nous avons trouvé utile lors de la première partie du projet tutoré .

—

Sixième partie

Annexes

21 Diagrammes de classes

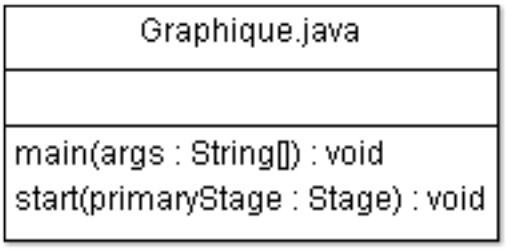

FIGURE 3 – Classe Graphique

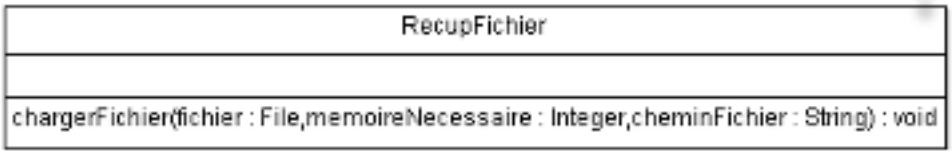

FIGURE 4 – Classe RecupFichier

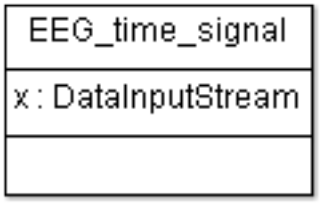

FIGURE 5 – Classe EEG_time_signal

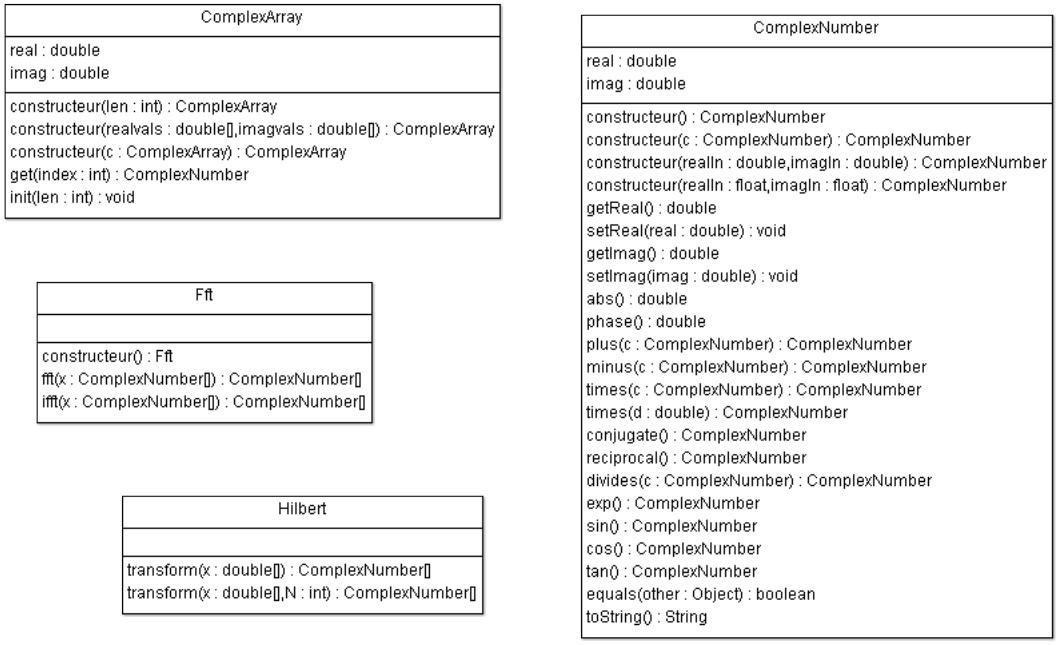

FIGURE 6 – Classes pour les calculs mathématiques

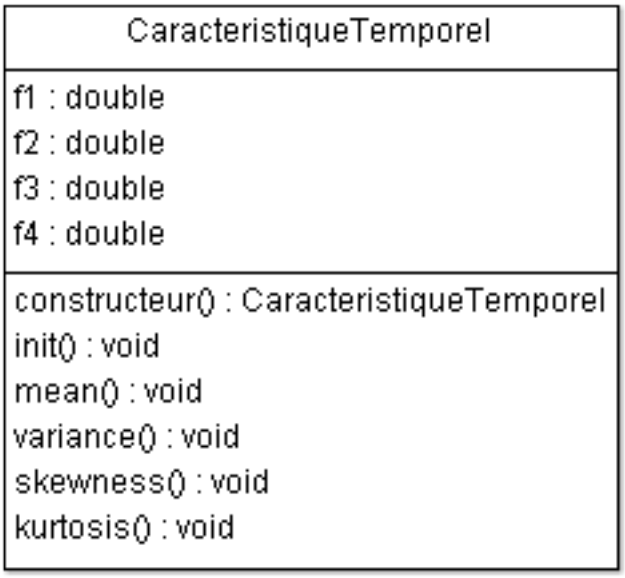

FIGURE 7 – Classe Caractéristique

FIGURE 8 – Packages composant l'application