

---

# **Rapport du travail de bachelor**

***Version 1.0***

**Sommer Nicolas**

16 July 2017

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Contexte du travail . . . . .	1
1.2	Problème à résoudre et but du projet . . . . .	1
1.3	Rappel des objectifs du projet . . . . .	1
<b>2</b>	<b>Analyses préliminaires</b>	<b>2</b>
2.1	Le format NIFTI . . . . .	2
2.2	Le calcul distribué . . . . .	3
2.3	Le deeplearning et choix d'une bibliotheque . . . . .	3
2.4	Docker . . . . .	3
<b>3</b>	<b>Conception</b>	<b>4</b>
3.1	Schémas conceptuels . . . . .	4
3.2	Description des classes . . . . .	4
3.3	Choix de la topologie du/des reseaux de neurones . . . . .	4
3.4	Description du workflow . . . . .	4
<b>4</b>	<b>Implémentation</b>	<b>5</b>
4.1	Configuration d'une expérience . . . . .	5
4.2	Lecture des données . . . . .	5
4.3	Configuration du/des réseaux . . . . .	5
4.4	Entraînement et évaluation sans Spark . . . . .	5
4.5	Entraînement et évaluation avec Spark local . . . . .	5
4.6	Entraînement et évaluation avec Spark sur un cluster . . . . .	5
<b>5</b>	<b>Expérience réalisée avec le CHUV</b>	<b>6</b>
5.1	Donnée de l'expérience . . . . .	6
5.2	Préparation et exécution de l'expérience . . . . .	6
5.3	Résultats . . . . .	6
<b>6</b>	<b>Analyses des résultats du projet</b>	<b>7</b>
<b>7</b>	<b>Gestion de projet</b>	<b>8</b>
7.1	Diagramme de Gantt . . . . .	8
7.2	Journal de travail . . . . .	8
7.3	Analyse de la gestion de projet . . . . .	8
<b>8</b>	<b>Conclusion</b>	<b>9</b>
8.1	Améliorations futures . . . . .	9
8.2	Ressenti personnel . . . . .	9
<b>9</b>	<b>Sources</b>	<b>10</b>

<b>10 Annexes</b>	<b>11</b>
10.1 Cahier des charges . . . . .	11
10.2 Journal de travail . . . . .	11
10.3 Plannification . . . . .	11
10.4 Manuel utilisateur . . . . .	11
10.5 Bibliographie . . . . .	11

---

**Introduction**

---

**1.1 Contexte du travail**

**1.2 Problème à résoudre et but du projet**

**1.3 Rappel des objectifs du projet**

---

## Analyses préliminaires

---

### 2.1 Le format NIFTI

Ce travail est un projet de neuro-imagerie, il est donc naturel de devoir travailler avec des images IRM du cerveau. Le format utilisé par le CHUV pour les images est le format NIFTI (Neuroimaging Informatics Technology Initiative), un format d'image très spécialisé mais également très répandu dans ce domaine.

Ce chapitre va donc présenter ce format afin de mieux le comprendre. Pour faire cela, nous allons voir l'origine du format, une vue d'ensemble des principales caractéristiques du format et quelques outils qui ont été utiles à la réalisation de ce travail.

#### 2.1.1 Origine du format NIFTI

(maybe add info sur les fichiers .mat qui accompagnait les fichiers analyze, parler de l'orientation radiologique ou neurologique).

NIFTI est un format de fichier pour sauvegarder des données d'IRM volumétrique. Il fonctionne sur le principe des voxels et est multidimensionnel.

Ce format a été imaginé il y a une dizaine d'années pour remplacer le format ANALYZE 7.5. Ce format était très utilisé mais était également très problématique. Le soucis principal de ce format étant le manque d'information sur l'orientation dans l'espace de l'élément scanné. Les données enregistrées ne pouvaient donc pas être lu et interprété sans ambiguïté. A cause de ce manque d'information il existait principalement une confusion entre le côté droit et le côté gauche du cerveau.

Deux conférences furent alors mises en place par quelques-uns des concepteurs des plus grands logiciels de neuroimagerie. Ces deux conférences, le Data Format Working Group (DFWG), se sont réunies au "National Institute of Health" (NIH) pour trouver un format de remplacement. De ces réunions naquit le format NIFTI. Celui-ci veut intégrer de nouvelles informations et devenir un nouveau standard de neuroimagerie.

#### 2.1.2 Vue d'ensemble du format NIFTI

Le format ANALYZE 7.5 avait besoin de deux fichiers pour fonctionner. Un fichier \*.hdr contenant le header pour stocker les méta-données et un fichier \*.img contenant les données de l'image. Le format NIFTI a conservé cette manière de faire afin de préserver la compatibilité avec les systèmes déjà en place. Toutefois, des améliorations ont été apportées et pour éviter de faire l'erreur d'oublier l'un des deux fichiers du format, il a été décidé de permettre le stockage dans un seul fichier avec l'extension \*.nii. Ces images contenant de grandes zones d'image noires, elles sont donc parfaites pour être compressées avec gzip. Il n'est donc absolument pas rare de trouver des fichiers NIFTI au format \*.nii.gz. Pour ce travail nous avons utilisé les formats \*.nii et \*.nii.gz.

Le format NIFTI est un format de fichier sur plusieurs dimensions. Au total, il peut compter jusqu'à 7 dimensions. Dans tous les cas, les 3 premières dimensions sont des dimensions spatiales (x, y, z) et la quatrième est une dimension temporelle. Les dimensions suivantes (5-7) sont des dimensions réservées à d'autre usage et sont plus

ou moins libre. Dans le cadre de ce projet, les images utilisées ne possèdent que 3 dimensions (les 3 dimensions spatiales). On peut donc voir les images comme étant un instantané du cerveau en 3 dimensions.

Les dimensions et d'autres informations importantes sur le fichier sont stockées dans un fichier header. Ce dernier est d'une taille de 348 octets. (Il y a un tableau de toutes les valeurs sur [brainder.org](http://brainder.org) il doit venir être collé ici.)

Le champs principalement utilisé lors de ce projet est le champs `short dim[8]`. Ce champs est un tableau contenant les données sur les dimensions du fichier. Ce tableau contient pour : - `Dim[0]` : Le nombre de dimensions - `Dim[1-7]` : Est un nombre positif contenant la longueur de la dimension en question.

### 2.1.3 Outils pratique

## 2.2 Le calcul distribué

(Ce chapitre présente le calcul distribué et spark, il est placé là afin de mettre en avant dans un premier temps les contraintes du projet. Il devient alors plus simple au choix suivant d'expliquer le choix de `dl4j` comme bibliothèque de deeplearning)

### 2.2.1 Qu'est ce que le calcul distribué ?

### 2.2.2 Spark

## 2.3 Le deeplearning et choix d'une bibliothèque

(Ce chapitre va résumer les avancées sur le deeplearning (avantage et inconvénient), puis il va expliquer le fonctionnement des réseaux de convolution (réseau employé durant le projet), puis on va faire un état de l'art des bibliothèques et défendre le choix de `dl4j`)

### 2.3.1 Considération générale

### 2.3.2 Réseaux de convolution

### 2.3.3 Bibliothèque disponible et choix

## 2.4 Docker

---

## Conception

---

### 3.1 Schémas conceptuels

### 3.2 Description des classes

#### 3.2.1 Package “Core”

La classe “Main”

La classe “DataReader”

#### 3.2.2 Package “Config”

La classe “Configuration”

#### 3.2.3 Package “Generator”

La classe “DataTestGenerator”

#### 3.2.4 Package “Wrapper”

La classe “WrapperDI4j”

La classe “LocalWrapperDI4j”

La classe “SparkWrapperDI4j”

### 3.3 Choix de la topologie du/des reseaux de neurones

### 3.4 Description du workflow

---

## Implémentation

---

### 4.1 Configuration d'une expérience

### 4.2 Lecture des données

### 4.3 Configuration du/des réseaux

### 4.4 Entraînement et évaluation sans Spark

### 4.5 Entraînement et évaluation avec Spark local

### 4.6 Entraînement et évaluation avec Spark sur un cluster



---

## Expérience réalisée avec le CHUV

---

### 5.1 Donnée de l'expérience

### 5.2 Préparation et exécution de l'expérience

### 5.3 Résultats

---

## Analyses des résultats du projet

---

---

## Gestion de projet

---

### 7.1 Diagramme de Gantt

### 7.2 Journal de travail

### 7.3 Analyse de la gestion de projet

---

## Conclusion

---

### 8.1 Améliorations futures

### 8.2 Ressenti personnel

---

**Sources**

---

---

**Annexes**

---

**10.1 Cahier des charges**

**10.2 Journal de travail**

**10.3 Plannification**

**10.4 Manuel utilisateur**

**10.5 Bibliographie**