Rapport du travail de bachelor

Version 1.0

Sommer Nicolas

20 July 2017

Table des matières

CHAPITRE 1

Introduction

Ce rapport présente un projet développé dans le cadre du Travail de Bachelor, au sein de la HES de Neuchâtel et en collaboration avec le Human Brain Project et le Laboratoire de Recherche en Neuro-imagerie (LREN) du CHUV de Lausanne. Il a été développé par Monsieur Nicolas Sommer sous la supervision de Monsieur Fabrizio Albertetti. Ce travail étant un mandat de la Medical Informatics Platform du Human Brain Project, la personne de contact et mandant était Arnaud Jutzeler.

Ce travail était d'une durée de 450h et c'est déroulé sur 10 semaines durant le semestre de printemps 2017 à raison de 4h par semaine puis sur une durée de 8 semaines a raison de 8h par jour.

Les premières semaines furent donc consacré à l'analyse préliminaire du projet, aux choix et à l'apprentissage des technologies qui seront employés. Le seconde partie fut consacré à l'implémentation et aux expériences.

Le reste de ce rapport présente les contraintes et les analyses préliminaires effectuées pour préparer la suite du projet, la conception et l'implémentation du programme lié au projet ainsi qu'une expérience qui a été réalisé à l'aide de ce dernier.

La suite de ce chapitre expose le contexte du travail, les buts de ce projet et les objectifs qui fixés pour celui-ci et une note sur la confidentialité liée à ce projet.

1.1 Contexte du travail

Le Human Brain Project (HBP) a pour but d'enrichir les connaissances humaines en matière de neuro-science en cherchant à mieux comprendre les mécanismes du cerveau humain. Ce projet s'inscrit dans le cadre du programme européen pour la recherche et l'innonvation Horizon 2020 et vise a accélerer les domaines des neuro-sciences, de l'informatique et de la médecine liée au cerveau. La première étape du Human Brain Project veut mettre à disposition des chercheurs un portail web. Ce portail web sera constitué d'un total de 6 plateformes de recherche. Celles-ci porteront sur la neuro-informatique, la simulation du cerveau, le calcul à haute performance, l'informatique médicale, l'informatique neuromorphique et la neuro-robotique.

Le département des Neuro-sciences Clinique du CHUV est chargé de la plateforme d'Informatique Médicale. Celle-ci est une plateforme open-source permettant aux hôpitaux et aux centre de recherche de partager des données médicales. Elle permettra aux utilisateurs d'avoir accès à des informations précises et pertinentes sur les maladies liées au cerveau en préservant la confidentialité des patients. Cette plateforme servira donc de pont entre la recherche en neuro-sciences, la recherche clinique et les soins aux patients. Elle pourrait également permettre la découverte de mécanisme à différentes échelles qui expliquerait l'apparition et le développement de maladies cérébrales.

C'est dans ce cadre que l'équipe du Laboratoire de Recherche En Neuro-imagerie cherche à développer un ensemble d' outils pour l'acquisition, le traitement et l'analyse des données. Elle cherche, entre autre chose, à pouvoir automatiser aussi efficacement que possible les diagnostiques de maladie pouvant atteindre le cerveau, tel que les maladies d'Alzheimer ou de Parkinson. Dans l'état actuel cette analyse peut se faire avec des méthodes de machine learning. Toutefois, il n'existe pas encore de méthode d'apprentissage profond disponible sur la plateforme et le LREN aimerait pouvoir proposé cette option aux utilisateurs de la plateforme.

C'est dans ce but que la Haute-Ecole Arc de Neuchâtel a été contacté et ce projet proposé comme travail de diplôme à un étudiant de troisième année.

1.2 Problème à résoudre et but du projet

Actuellement, la plateforme rassemble un certain nombre d'images d'IRM. Celles-ci sont stockés sous la forme de DICOM ou de fichier au format NIFTI. Le DICOM est une norme standard pour la gestion informatique des données issues de l'imagerie médicale. N'étant pas employé dans le reste du projet, il ne sera pas plus détaillé ici. Le format NIFTI est un format d' image IRM mis en place par quelques uns des acteurs les plus influents de la neuro-imagerie. Etant le format principalement employé dans ce projet, il fera l'objet d'une description détaillé dans la partie consacré aux analyses préliminaires.

Afin d'être employé par les outils d'automatisation de diagnostique mis en place par la plateforme d'informatique médicale, les images ont besoin d'être pré-traité. En effet, les outils de machine learning utilisé pour le diagnostique fonctionne en se basant sur un certain nombre de caractéristiques du cerveau. Ces caractéristiques peuvent être le volume de matière grise ou de matière blanche d'une zone spécifique du cerveau, la quantité de liquide cérébro-spinal, le volume du cerveau, etc. Il faut donc extraire ces informations des images à disposition. Ceci se fait à l'aide du framework SPM. Ce framework permet grâce à la segmentation de récupérer des images d'une de ces caractéristique. La segmentation permet, par exemple, de récupérer une image IRM de la matière grise du cerveau au format NIFTI. SPM utilise la segmentation afin de créer un atlas des caractéristiques. Ainsi l'atlas fait correspondre, sous la forme de tableau, un certain nombre de quantité caractéristique à chaque région du cerveau.

(Insérer image du dataflow)

Une fois ce pré-traitement effectué, les données sont prêtes pour être utilisé par l'"'algorithm factory". Cette dernière correspond à l'ensemble des outils de diagnostique de la plateforme.

Le problème principal de cette manière de faire est qu'il existe une quantité non-négligeable d'information qui sont perdu au cours du pré-traitement. L'idée du LREN est donc de trouver une solution pour traiter directement les images entières avec des outils de diagnostique automatique.

Pour se faire, ils proposent de mettre en place une extension de l'"'algorithm factory". Cette extension permettra d'appliquer des algorithmes pour l'apprentissage de modèles et de faire valider ces derniers directement sur les images d'IRM.

Ce projet vise donc à explorer la possibilité de mettre en place cette extension. Il mettra en place un workflow alternatif à celui existant dans l'"'algorithm factory". Cette alternative a pour contrainte de permettre de lancer de nouveaux algorithmes travaillant directement sur les images en utilisant le framework de calcul distribué Apache-Spark. Cette nouvelle fonctionnalitée sera illustrée par l'intégration d'une bibliothèque de deep-learning et fera l'objet d'une expérience avec de véritable image d'IRM.

1.3 Rappel des objectifs du projet

Les premières semaines du projet ont été utilisé afin de fixer les objectifs principaux et secondaires de ce projet. Ainsi, les objectifs principaux de ce travail sont : 1. L'installation et la prise en main d'Apache-Spark 2. L'intégration d'Apache-Spark à l'"'algorithm factory" 3. L'interfaçage des bases de données d'image de la plateforme à Apache-Spark. Ces bases de données sont a créer et a améliorer si besoin durant le projet. 4. Faire un état de l'art technique sur les différentes bibliothèques de deeplearning compatible avec Apache-Spark pour obtenir suffisamment d'information pour permettre le choix de l'une d'entre elles à intégrer au-dessus de Spark. 5. Traduire la partie prédictive de l'algorithme au format PFA (Portable Format for Analytics). 6. Tester les nouvelles fonctionnalités avec une expérience concrète fournit par le LREN. Cette expérience utilisera des images d'IRM utilisé par le laboratoire. Elle consistera en une classification de ces images.

En plus de ces objectifs principaux, s'ajoute un objectif optionnel. Celui-ci consiste a étendre le portail web de la plateforme pour permettre l'utilisation des nouvelles fonctionnalités de l'"'algorithm factory".

1.4 Note sur la confidentialité au cours du projet

Comme déjà rappelé dans le cahier des charges de ce travail, l'aspect de l'utilisation d'image extraite d'IRM est un aspect sensible du point de vue de la confidentialité.

Pour pallier tous soucis de confidentialité, les images employées durant la phase de développement seront des images totalement ouvertes même si ces dernières ne sont pas des images issues d'IRM.

Si des images autres que des données de recherche devaient être utilisées, elles seront anonymisée et ne quitteront jamais le réseau sécurisé des hôpitaux dont elles sont originaires.

Une attention particulière devra également être portée sur la réutilisation de l'existant afin de respecter les directives de plagiat et le droit d'auteur (cf. directives générales en matière de plagiat de la HE-ARC).

CHAPITRE 2

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 présente donc ce format afin de mieux le comprendre. Pour faire celà, 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

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

Ce format a été imaginé il y a une dizaine d'année 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 interprêté sans ambiguité. 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 réunit au "National Institue of Health" (NIH) pour trouver un format de remplacement. Dec 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 l'idée d'avoir un header et des données afin de préserver la compatibilité avec les systèmes déjà en place. Toutefois, des améliorations ont été apportés et pour évité 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 que l'ont peu représenter par une matrice multidimensionnel. 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 reservées à d'autre usage et sont plus ou moins libre. Dans le cadre de ce projet, les images utilisées ne possède que 3 dimensions (les 3 dimensions spatiales). On peut donc voir les images comme étant un instantané du cerveau en 3 dimensions et chaque case de la matrice de données représente un voxel de cette image.

Les dimensions et d'autres informations importantes sur le fichier sont stocké 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 il doit venir etre 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.

Pour ce travail deux types de NIFTI ont été employé. Le premier type de NIFTI a avoir été utilisé sont des images générés et très simple. Ces images correspondent à des sphères et des cubes. La dimension de ces images générées peut être choisi. Au début du projet, de manière a facilité les tests, la taille de ces images étaient de 100x100x100. Puis lorsque le projet eut une forme plus concrète la taille fut changer pour correspondre à la taille standard utilisé par le CHUV (190x190x160). Le second type de données correspond aux images fournient par le LREN. A savoir des images de la matière grise du cerveau avec une taille standard de 190x190x160.

2.1.3 Outils pratique

(A finir avec inspiration XD) Le format NIFTI est un format très spécifique au domaine de la neuro-imagerie. Il fallait donc, au début du projet, pouvoir visualiser et manipuler ce genre de fichier. Pour faire celà, il existe de nombreux outils. Ce chapitre va donc présenter de manière suscinte les outils qui ont été employé pour la réalisation du projet.

2.2 Le calcul distribué

(A revoir) Le nombre d'image et la taille de ces dernières font qu'il y a un nombre très important de données et de calcul a effectué. Pour le confort de l'utilisateur, le temps de traitement de ces données doit être le plus court possible. La plateforme actuellement en place au CHUV tourne donc sur un cluster de machine afin de permettre à l'utilisateur d'obtenir le plus rapidement possible les résultats des analyses qu'il demande.

Ce projet doit donc pouvoir se porter sur l'infrastructure en place. De plus, le Laboratoire de Recherche En Neuro-imagerie désire intégrer la technologie Spark pour effectuer leur calcul. Ces deux contraintes ont donc fait l'objet d'une analyse et sont exposé dans ce chapitre.

2.2.1 Qu'est ce que le calcul distribué ?

Ces dernières années la quantité de données disponibles a explosé. Rapidement, les technologies ont du s'adapter à cette quantité d'information toujours plus importante à traiter. L'une des solutions trouvé pour résoudre se problème consiste à répartir les tâches de traitement (de calcul) sur plusieurs unité de travail. Ainsi, on répartit le besoin en puissance de calcul, pour un projet, en petite entités sur autant d'ordinateurs disponible qu'il y en a dans notre réseau distribué.

Celà permet d'exploiter les ressources de chaques machines au profit d'un projet commun. Ce projet dispose alors d'une puissance de calcul de la somme de tous les ordinateurs individuels.

Le calcul distribué s'effectue donc au sein d'un cluster de machine. C'est à dire, au sein d'un groupe de machines indépendantes fonctionnant comme une seule et même entité. Chacune de ces entités correspond à un noeud. Si une machine est ajouté au cluster, la puissance de calcul est directement augmenté contrairement à une machine seule, où si l'on veut augmenter la puissance de calcul, il faut augmenter la puissance des processeurs.

Pour le calcul distribué, les noeuds sur lesquels les calculs sont exécuté sont donc distant, autonome et ne partage pas de ressources. Il faut donc que chaques noeuds communiquent avec les autres au travers de message qu'il s'envoie au travers du cluster.

Pour pouvoir distribuer son projet, il faut donc diviser le problème initial en sous-problème et assigner à chaque noeud l'un de ces sous-problèmes. Chaque noeud effectue la tâche qui lui est assigné. On récupère alors le résultat de chacun des sous-problèmes et on les combine pour obtenir le résultat finale du projet initial.

Afin de gérer tout celà il est possible d'utiliser des framework de calcul distribué. Ces framework fournissent un ensemble d'outils pour faciliter la création d'application distribuées. Le CHUV à choisi pour ce projet d'utiliser le framework Apache-Spark. La suite de ce chapitre présentera donc ce framework et son fonctionnement.

2.2.2 Spark

2.3 Le deeplearning et choix d'une bibliotheque

(Ce chapitre va résumer les avancés sur le deeplearning (avantage et inconvénient), puis il va expliquer le fonctionnement des réseaux de convolution (reseau employe durant le projet), puis on va faire un état de l'art des bibliothèques et defendre le choix de dl4j)

2.3.1 Considération générale

2.3.2 Réseaux de convolution

2.3.3 Deeplearning et calcul distribué

2.3.4 Bibliothèque disponible et choix

2.4 Docker

CHAPITRE 3

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 “WrapperDl4j”

La classe “LocalWrapperDl4j”

La classe “SparkWrapperDl4j”

3.3 Choix de la topologie du/des reseaux de neurones

3.4 Description du workflow

CHAPITRE 4

Implémentation

CHAPITRE 5

Expérience réalisée avec le CHUV

5.1 Donnée de l'expérience

5.2 Préparation et exécution de l'expérience

5.3 Résultats

CHAPITRE 6

Analyses des résultats du projet

CHAPITRE 7

Gestion de projet

7.1 Diagramme de Gantt

7.2 Journal de travail

7.3 Analyse de la gestion de projet

CHAPITRE 8

Conclusion

8.1 Améliorations futures

8.2 Ressenti personnel

CHAPITRE 9

Sources

CHAPITRE 10

Annexes

10.1 Cahier des charges

10.2 Journal de travail

10.3 Plannification

10.4 Manuel utilisateur

10.5 Bibliographie