



RAPPORT DE STAGE

Norvège

2^{ème} année F4

Projet réalisé par

Julien Feuillas

le 16 mai 2019

Établissement d'un algorithme de modification du "taux d'apprentissage" en vue d'optimiser le modèle d'apprentissage d'un réseau de neurones

Tuteur de Stage : **Arvid Lundervold**
Co-encadrant de Stage : **Alexander Lundervold**

Jury

Arvid Lundervold,	Professeur UiB	Maître de Stage
Murielle Mouzat,	Professeure ISIMA	Communication
Vincent Barra,	Professeur ISIMA	Tuteur ISIMA

durée : 5 mois

Remerciements

Je tiens à remercier Monsieur Arvid Lundervold ainsi que son fils Monsieur Alexander Lundervold qui m'ont encadré et aidé au cours de ce stage. Je souhaite également remercier Monsieur Vincent Barra qui m'a permis d'accéder à ce stage.

Résumé

Le but de ce stage est de déterminer un algorithme permettant de moduler le “taux d’apprentissage” au cours de l’entraînement d’un réseau de neurones. Comme l’étude se plaçait dans un cadre de recherche dans le domaine médical, il fut décidé que je m’intéresserais principalement à des réseaux de neurones générés par NiftyNet[4].

Le langage de programmation que j’utilise est le Python. Il s’agit actuellement d’un des langages de programmation les plus utilisés dans le cadre du Machine Learning. Pour utiliser NiftyNet, qui est une application programmée en Python, j’ai utilisé l’environnement Anaconda ainsi que la bibliothèque fondamentale en ce qui concerne les réseaux de neurones : TensorFlow.

Pour ce qui est de l’algorithme, j’utilise le langage de programmation Python ainsi que la bibliothèque TensorFlow. Comme mon travail est de modifier la manière de réagir de NiftyNet suivant son avancement dans l’apprentissage du modèle, j’utilise également des modules écrits par les programmeurs de NiftyNet[5].

Mots-Clés : Optimisation, Deep Learning, Machine Learning, réseau de neurones, Python

Abstract

The purpose of this internship is to determine an algorithm which permit us to modify the learning rate during the training of some neural networks. Because the main domain of research for this internship is related with the medicine, it was decided that I will work with neural networks which are generated by NiftyNet.

The programming language I use is Python. It is currently one of the programming language the most used when it concerns the Machine Learning. To use NiftyNet, which is a Python application, I used the anaconda environment and one of the most famous library when it comes to the neural networks : TensorFlow.

For the algorithm, I use the Python language and the TensorFlow library. Because my work is to modify the reaction of NiftyNet during the learning period, I use some part of the code written by the developers of the NiftyNet application.

Keywords : Optimisation, Deep Learning, Machine Learning, neural networks, Python

Table des matières

Remerciements	3
Résumé	4
Abstract	4
Table des matières	5
Liste des tableaux	6
Table des figures	6
Introduction	7
1 Contexte	8
2 Principe	9
2.1 Deep Learning ou Apprentissage Profond	9
2.2 NiftyNet	10
3 Méthodes	11
3.1 Le fichier de configuration	11
3.2 Segmentation	11
3.3 Différents réseaux	11
4 Difficultés rencontrées	12
5 Résultats	13
A Fichier de Configuration	14
Index	16
Bibliographie	17

Liste des tableaux

Table des figures

2.1	Exemple de méthodes de Machine Learning	9
2.2	exemple de segmentation	10
2.3	Sur-apprentissage et Sous-apprentissage en régression	10

Introduction

Dans le cadre de mon stage, je me suis plus intéressé aux méthodes de Deep Learning qui s'avèrent être un cas particulier de Machine Learning. Il s'agit de méthodes utilisant des réseaux de neurones. Un réseau de neurones est composé de plusieurs couches et chaque couche est composée de plusieurs "neurones". Dans le cas des réseaux de neurones un "neurone" représente une opération élémentaire comme une addition ou une multiplication. Cette succession d'opération élémentaires permet de réaliser des modèle mathématiques plus développé et ainsi d'apprendre "plus en profondeur" comment s'agence les données. C'est de ce principe que vient le terme de Deep Learning.

Actuellement, les techniques de Machine Learning et de Deep Learning sont de plus en plus utilisées, notamment dans le domaine médical permettant ainsi de faciliter la tâche des médecins lors de l'analyse d'IRM par exemple. Il est donc important que les méthodes utilisées soient fiables, mais cela pose aussi problème si l'ordinateur met plus de temps que l'humain pour déterminer la présence de tumeur. Mon travail a donc été de déterminer une méthode pour diminuer le temps d'apprentissage du modèle pour une méthode de segmentation.

Les méthodes de prédiction de Deep Learning ont connu un certain essor dans tous les domaines. Ainsi, un certain nombre de librairies (Python) open-source permettant de réaliser des réseaux de neurones ont vu le jour comme TensorFlow ou encore FastAI.

Partie 1

Contexte

Les méthodes de Machine Learning et notamment de Deep Learning intéressent de plus en plus. Elles ont ainsi pu se faire une place dans la plupart des domaines et notamment dans le domaine médical.

Partie 2

Principe

Le principe du Machine Learning est de donner à l'ordinateur un grand nombre de données pour que celui-ci détermine un modèle mathématiques permettant de différencier les données :

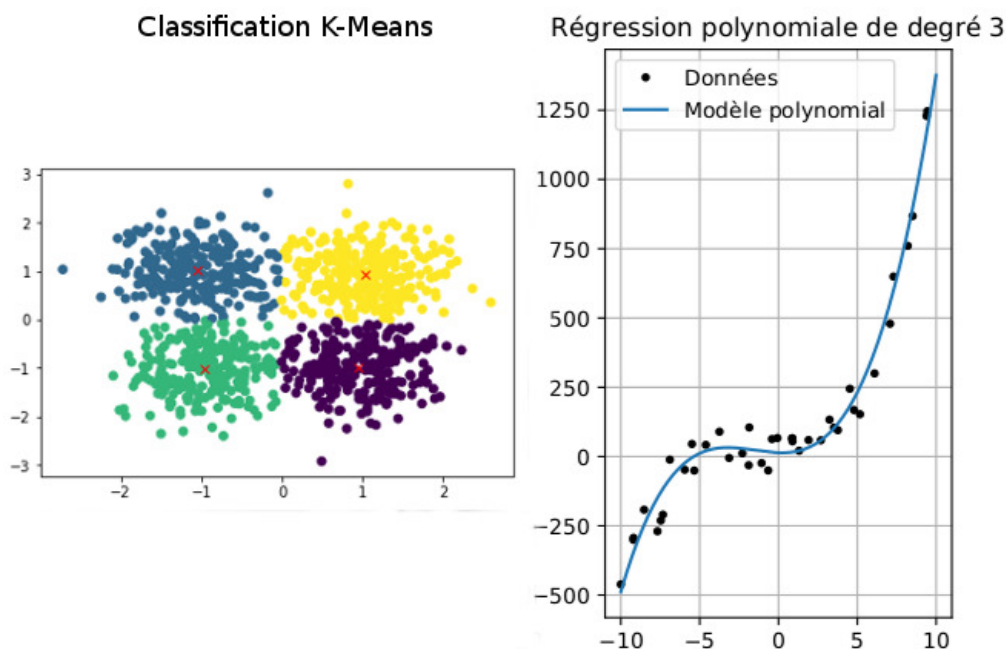


FIGURE 2.1 – Exemple de méthodes de Machine Learning

2.1 Deep Learning ou Apprentissage Profond

Les méthodes de Deep Learning peuvent aussi se décomposer en plusieurs groupes. Parmi ceux-ci se trouve la segmentation, qui est la méthode que j'ai étudiée au cours de ce stage. La segmentation est le principe de déterminer au sein d'un élément d'un jeu de données, une image par exemple, différentes classes. Ces méthodes sont différentes des méthodes de classification qui consistent à classer les éléments du jeu de données.

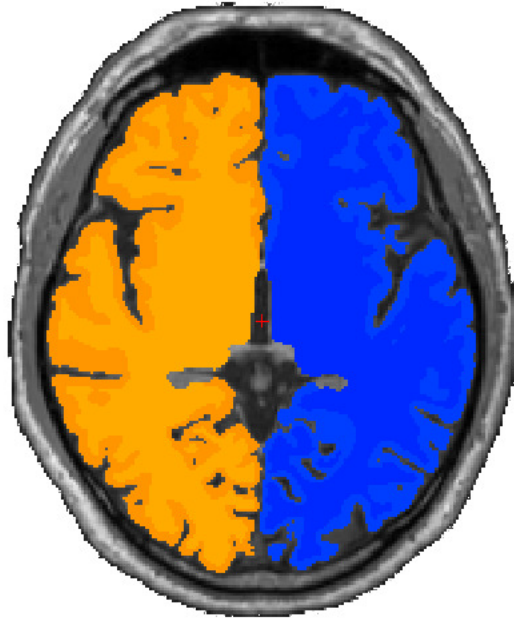


FIGURE 2.2 – exemple de segmentation

2.2 NiftyNet

Pour faciliter un usage plus spécifique de la librairie TensorFlow, la plate-forme NiftyNet a vu le jour. Cette plate-forme open-source s'est spécialisée dans le domaine médicale et dans la lecture des images issues des IRM. Cette plate-forme permet une fois installée de réaliser des réseaux de neurones sans avoir une très grande connaissance du langage de programmation, seule la complétion du fichier de configuration est nécessaire au bon fonctionnement du réseau. Pour cela, il est suffisant de lire la documentation de la plate-forme pour bien réussir son réseau de neurones.

Mais ce qui peut poser problème est l'entraînement du réseau. Ce dernier est assez long à réaliser et des problèmes tels que le sur-apprentissage ou le sous-apprentissage peuvent survenir.

FIGURE 2.3 – Sur-apprentissage et Sous-apprentissage en régression

Partie 3

Méthodes

3.1 Le fichier de configuration

A

3.2 Segmentation

3.3 Différents réseaux

Partie 4

Difficultés rencontrées

Partie 5

Résultats

Annexe A

Fichier de Configuration

```
1 [image]
2 path_to_search=data/images
3 filename_contains=IXI, orig
4 interp_order=3
5 axcodes=L,P,S
6 spatial_window_size=80, 80, 80
7
8 [label]
9 path_to_search=data/labels_bin
10 filename_contains=IXI, brain
11 interp_order=0
12 axcodes=L,P,S
13 spatial_window_size=80, 80, 80
14
15 [SYSTEM]
16 cuda_devices=0
17 num_threads=10
18 num_gpus=1
19 dataset_split_file = ./split.csv
20 model_dir = /home/julien/traineeship/mmiv/model
21
22 [NETWORK]
23 name=highres3dnet
24 activation_function=prelu
25 batch_size=1
26 reg_type=L2
27 decay=1e-5
28 queue_length=20
29 window_sampling=uniform
30
31 [TRAINING]
32 optimiser=adam
33 sample_per_volume=80
34 lr=1e-3
35 loss_type=Dice
36 starting_iter=0
37 save_every_n=2500
38 max_iter=200000
39 max_checkpoints=1000
40 exclude_fraction_for_inference = 0.01
41 tensorboard_every_n = 10
42
```

```
43 validation_every_n=10
44 validation_max_iter=1
45 exclude_fraction_for_validation= 0.02
46
47 [EVALUATION]
48 evaluations=dice
49 save_csv_dir=output/eval_1
50
51 [INFERENCE]
52 border=0, 0, 0
53 inference_iter=-1
54 save_seg_dir=output/out_1
55 output_postfix=_output
56
57 [SEGMENTATION]
58 image=image
59 label=label
60 output_prob=False
61 num_classes=2
62 label_normalisation=True
```

Index

algorithme, 4

IRM, 7, 10

Machine Learning, 4, 7, 9

- classification, 9

- Deep Learning, 7, 9

 - neural networks, 4, 7, 10

 - segmentation, 7, 9

- learning rate, 4

NiftyNet, 4, 10

open-source, 7, 10

Python, 4, 7

- Anaconda, 4

- TensorFlow, 4, 7, 10

Bibliographie

- [1] Tensorflow. <https://www.tensorflow.org/>,
date of consultation : April 2019.
- [2] Vitaly Bushaev. Adam – latest trends in deep learning optimization, October 2018.
<https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c>,
date of consultation : 25th April 2019.
- [3] Vitaly Bushaev. Improving the way we work with learning rate, November 2018.
<https://techburst.io/improving-the-way-we-work-with-learning-rate-5e99554f163b>,
date of consultation : 30th April 2019.
- [4] Eli Gibson, Wenqi Li, Carole Sudre, Lucas Fidon, Dzhoshkun I. Shakir, Guotai Wang, Zach Eaton-Rosen, Robert Gray, Tom Doel, Yipeng Hu, Tom Whyntie, Parashkev Nachev, Marc Modat, Dean C. Barratt, Sébastien Ourselin, M. Jorge Cardoso, and Tom Vercauteren. Niftynet : a deep-learning platform for medical imaging. Computer Methods and Programs in Biomedicine, 2018.
date of consultation : April 2019.
- [5] Eli Gibson, Wenqi Li, Carole Sudre, Lucas Fidon, Dzhoshkun I. Shakir, Guotai Wang, Zach Eaton-Rosen, Robert Gray, Tom Doel, Yipeng Hu, Tom Whyntie, Parashkev Nachev, Marc Modat, Dean C. Barratt, Sébastien Ourselin, M. Jorge Cardoso, and Tom Vercauteren. Niftynet source code, April 2019.
<https://github.com/NifTK/NiftyNet>,
date of last consultation : 25th April 2019.
- [6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016.
<http://www.deeplearningbook.org>.
- [7] Alexander Selvikvåg Lundervold and Arvid Lundervold. An overview of deep learning in medical imaging focusing on mri. <https://www.sciencedirect.com/science/article/pii/S0939388918301181>,
page 26, December 2018.
- [8] Marc Modat, Miklos Espak, Eli Gibson, Imanol Luengo, Dzhoshkun Shakir, Zach Eaton-Rosen, Carole Sudre, Tom Vercauteren, Matteo Mancini, Guotai Wang, Lucas Fidon, Wenq Li, Jorge Cardoso, Matt Clarkson, Mian Asbat Ahmad, and Tom Doel. Niftynet, October 2018.
<https://cmiclab.cs.ucl.ac.uk/CMIC/NiftyNet>,
date of consultation : 4th April 2019.
- [9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn : Machine learning in python, March 2019. <https://scikit-learn.org/stable/>.
- [10] Wikipedia. Content-based image retrieval, March 2019.
https://en.wikipedia.org/wiki/Content-based_image_retrieval,
date of consultation : 3rd April 2019.