



Soutenance du Stage de deuxième année F4 Détermination d'un algorithme améliorant l'apprentissage d'un réseau de neurones

Julien Feuillas

ISIMA

23 Mai 2019

Objectifs du groupe MMIV

- MMIV (Mohn Medical Imaging and Visualization Centre)
 - Centre de recherche en termes d'imagerie médicale
 - Basé en Norvège
 - Dépendant de l'Université de Bergen
 - Représentante : Mme Renate Grüner
- Objectifs
 - Mettre en place de nouvelles techniques d'apprentissage automatique

Travaux réalisés dans le cadre du projet

- Recherche effectuée par le laboratoire
- Récupération de données
 - Jeu de données d'IRM de cerveaux
- Mise en place d'une première solution
 - Acquisition de données
 - Algorithme modifiant le paramètre "Learning Rate" au cours de l'entraînement

Cadre du Stage

- Nos Objectifs
 - Déterminer l'impact du paramètre Learning Rate sur l'apprentissage d'un réseau de neurones
 - Améliorer si possible cet apprentissage
 - Étude de différentes solutions
- Cadre d'étude
 - Optimisation de fonction
 - Deep Learning
 - Segmentation
 - Learning rate

Problématique

Est-il possible d'améliorer l'apprentissage d'une méthode de Deep Learning en modifiant le "taux d'apprentissage" au cours eu cours de l'entraînement ?

Plan

1 Présentation de Niftynet

2 Travail à réaliser

Plan

1 Présentation de Niftynet

2 Travail à réaliser

Installation

- Choix du matériel
 - CPU
 - GPU
- Installation Anaconda
- Installation Tensorflow
- Installation NiftyNet

Fichier de Configuration

```
[image]
path_to_search=data/images
filename_contains=IXI, orig
interp_order=3
axcodes=L,P,S
spatial_window_size=80, 80, 80
```

```
[label]
path_to_search=data/labels
filename_contains=IXI, brain
interp_order=0
axcodes=L,P,S
spatial_window_size=80, 80, 80
```

```
[SYSTEM]
cuda_devices=0
num_threads=10
num_gpus=1
```

```
[NETWORK]
name=highres3dnet
activation_function=prelu
batch_size=1
reg_type=L2
decay=1e-5
queue_length=20
```

```
[TRAINING]
optimiser=adam
sample_per_volume=80
lr=1e-3
loss_type=Dice
starting_iter=0
save_every_n=2500
max_iter=20000
max_checkpoints=1000
```

-> validation

```
[INFERENCE]
```

```
[EVALUATION]
```

```
[SEGMENTATION]
image=image
label=label
output_prob=False
num_classes=2
label_normalisation=True
```

Entraînement du modèle

```
INFO:niftnet: training iter 2725, loss=0.5583992600440979 (0.8680345)
INFO:niftnet: training iter 2726, loss=0.6007713079452515 (0.8765895)
INFO:niftnet: training iter 2727, loss=0.6028639078140259 (0.8638375)
INFO:niftnet: training iter 2728, loss=0.7271522879600525 (0.8684865)
INFO:niftnet: training iter 2729, loss=0.5072353482246399 (0.8820975)
INFO:niftnet: training iter 2730, loss=0.5244722962379456 (0.8503695)
INFO:niftnet: validation iter 2730, loss=0.5917767882347107 (0.3542495)
INFO:niftnet: training iter 2731, loss=0.5584901571273804 (0.8557275)
INFO:niftnet: training iter 2732, loss=0.5593559145927429 (0.8369515)
INFO:niftnet: training iter 2733, loss=0.2506962716579437 (0.8295555)
INFO:niftnet: training iter 2734, loss=0.5547387599945068 (0.8322225)
INFO:niftnet: training iter 2735, loss=0.7395937412643433 (0.8774605)
INFO:niftnet: training iter 2736, loss=0.5985323190689007 (0.8776905)
INFO:niftnet: training iter 2737, loss=0.5332014560699463 (0.8878045)
INFO:niftnet: training iter 2738, loss=0.5504813194274902 (0.8793805)
INFO:niftnet: training iter 2739, loss=0.646152138710022 (0.8650545)
INFO:niftnet: training iter 2740, loss=0.6777690052986145 (0.9185095)
INFO:niftnet: validation iter 2740, loss=0.5845479965209961 (0.3716615)
INFO:niftnet: training iter 2741, loss=0.5483676791191101 (0.8667025)
INFO:niftnet: training iter 2742, loss=0.597277045249939 (0.8673515)
INFO:niftnet: training iter 2743, loss=0.5173805952072144 (0.8343795)
INFO:niftnet: training iter 2744, loss=0.6961113214492798 (0.8735465)
INFO:niftnet: training iter 2745, loss=0.5568277835845947 (0.8374325)
INFO:niftnet: training iter 2746, loss=0.5968761444091797 (0.8592965)
INFO:niftnet: training iter 2747, loss=0.5393800748740878 (0.8739805)
INFO:niftnet: training iter 2748, loss=0.5923281980835278 (0.8449805)
INFO:niftnet: training iter 2749, loss=0.5360341668128967 (0.8690455)
INFO:niftnet: training iter 2750, loss=0.0 (0.8191595)
INFO:niftnet: validation iter 2750, loss=0.5554222464561462 (0.3618795)
INFO:niftnet: training iter 2751, loss=0.5011717081069940 (0.8784815)
INFO:niftnet: training iter 2752, loss=0.6217421293258667 (0.8794405)
INFO:niftnet: training iter 2753, loss=0.5196523666381836 (0.8822105)
INFO:niftnet: training iter 2754, loss=0.2939032316207086 (0.8423635)
INFO:niftnet: training iter 2755, loss=0.6358374953269958 (0.8766345)
INFO:niftnet: training iter 2756, loss=0.578665018081065 (0.8728015)
INFO:niftnet: training iter 2757, loss=0.5292662978172302 (0.8703335)
INFO:niftnet: training iter 2758, loss=0.537699818611145 (0.8938075)
INFO:niftnet: training iter 2759, loss=0.577454149723053 (0.8938815)
INFO:niftnet: training iter 2760, loss=0.5138360857963562 (0.8732335)
INFO:niftnet: validation iter 2760, loss=0.5427689552307129 (0.3063955)
INFO:niftnet: training iter 2761, loss=0.7498124837875366 (0.9073135)
INFO:niftnet: training iter 2762, loss=0.5954206585884094 (0.8841635)
INFO:niftnet: training iter 2763, loss=0.5001587867730810 (0.8473115)
INFO:niftnet: training iter 2764, loss=0.573363268032975 (0.8646055)
```

Figure – Entraînement du système

Loss Function

– training

– validation

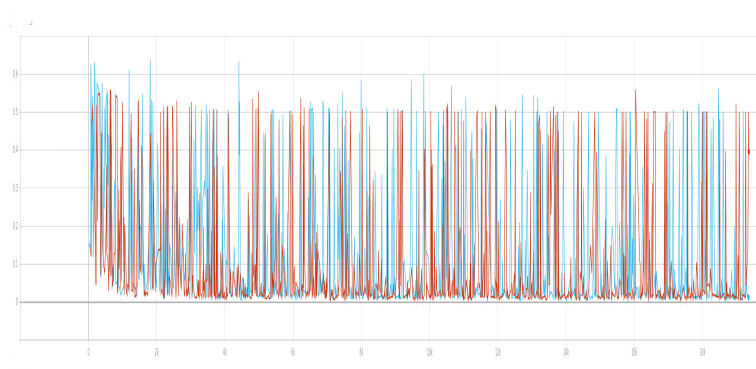


Figure – Courbe réelle

Loss Function

– training

– validation

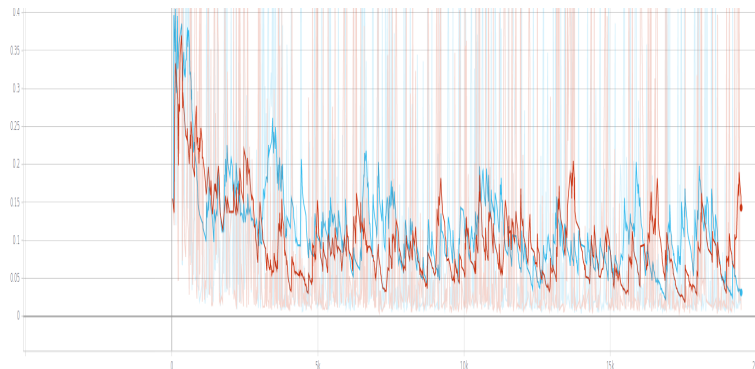


Figure – Courbe amortie avec un coefficient 0.9

Loss Function

– training

– validation

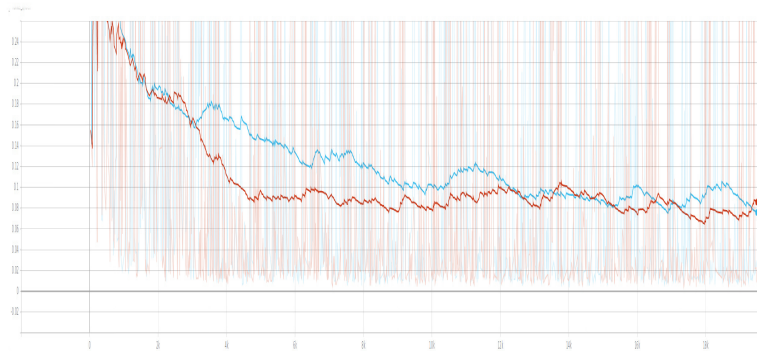


Figure – Courbe amortie avec un coefficient 0.99

Loss Function

– training

– validation

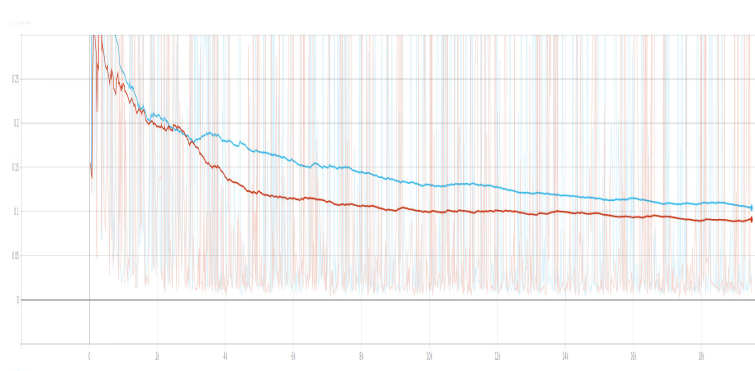


Figure – Courbe amortie avec un coefficient 0.999

Résultats

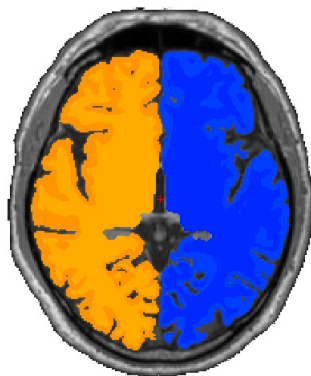


Figure – Segmentation

Plan

1 Présentation de Niftynet

2 Travail à réaliser

Étude de l'impact du paramètre lr

Étude réalisée avec 2 “learning rate” différents :

- 1 $lr = 10^{-3}$
- 2 $lr = 10^{-6}$

Étude de l'impact du paramètre lr

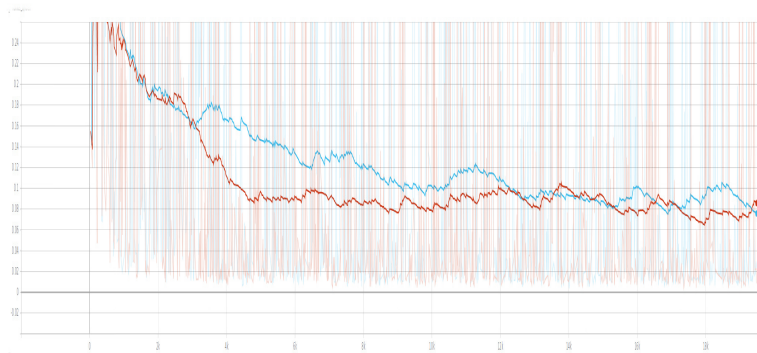


Figure – loss function with a learning rate of 10^{-3}

Étude de l'impact du paramètre lr

Figure – loss function with a learning rate of 10^{-6}

Première modification du lr

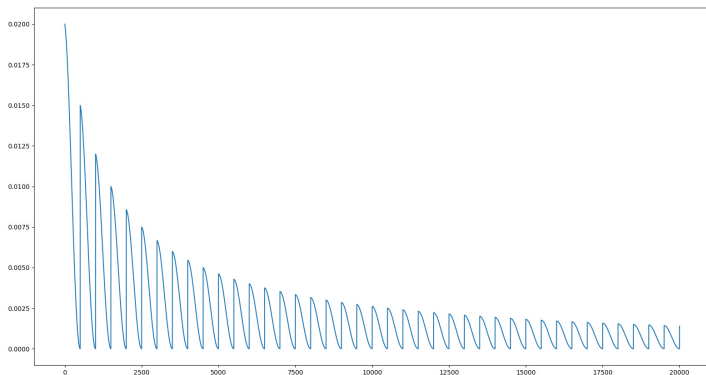


Figure – Modification du “taux d’apprentissage” à chaque itération

Première modification du lr

Amélioration et Poursuite

- Modifier le réseau pour de meilleurs résultats
- Déterminer le rôle du “taux d'apprentissage”
- Déterminer un algorithme utilisant le “taux d'apprentissage” pour optimiser l'apprentissage

Remerciements

Je vous remercie pour votre attention