Internship report 2^{nd} year

Robin Camarasa

January 27, 2019

Contents

Introduction

For our second year internship, I realised a 12 weeks mission between June the 4^{th} and August the 24^{th} at BIGR[1] a department of Erasmus MC[2]. In collaboration with Corentin Doué, a student at the EMSE. Florian Dubost, PHD student at Erasmus University and my direct supervisor and Marleen de Bruijne Mr. Dubost's supervisor.

Our main mission was the segmentation of White Matter Hyperintensities and submit the methods to an international Challenge the WMH Segmentation challenge[3]. Even if this Challenge was ended before the beginning of my internship it was interesting to send our codes to the Challenge organisation to compare our results with other teams of researchers that have worked on the same problem with the same data-set.

At the beginning of August, we applied to another International Challenge MRBrainS18[3]. The final deadline was August 15^{th}. That Challenge had the same organisation committee and the same structure than WMH challenge. It explains our submission in that short amount of time. Our results will be presented at MICCAI conference in Granada this September.

[1] Erasmus Medical Center
[2] Biomedical Imaging Group Rotterdam
[3] `http://wmh.isi.uu.nl/`
[3] `http://mrbrains18.isi.uu.nl/`

Work environment

2.1 University description

During my internship we worked at Erasmus MC based in Rotterdam and affiliated with Erasmus University. It is one of the most scientifically influent Medical Center of Europe. That Medical Center shares their buildings with an hospital and many Research Departments (Figure 2.1 on page 5). I worked in one of them the BIGR department which is specialised in Image Analysis and Deep Learning Methods.

BIGR was born from a collaboration of the Department of Radiology and Medical Computation in order to apply advanced techniques to biological imaging. In that structure evolve and collaborate researchers, PHD students and post graduated medical students from various fields such as biology or computer sciences.

2.2 Organisation aspect

The organisation in my Department was really free and up to you. They were no official schedules for coming to and leaving work. But you had to attend to several meetings. For instance, I had a weekly appointment with my direct supervisor (Florian Dubost) every Wednesday at 3pm that usually lasted an hour. Furthermore, I had to go to the team meeting every Thursday at 10am, during those meetings every member of the team used to give its last updates

Figure 2.1: Erasmus MC main entrance

and one of the member of the team used to explain us their research. Finally, once each two weeks I had to show my progress to Marleen de Bruijne, Florian Dubost's supervisor.

2.3 Human aspect

I don't know if it's because I worked in a university or in the Netherlands, but where I worked, the vision of the hierarchy was really different from the one I experimented in France. The supervisor was not seen as chief but more as a mentor that guided you to get the best of yourself.

A lot of team building activities was done to make the group work together such as Research Lunch or Research Drink. Once each 2 weeks all the service gather together around a meal to debate about a presentation done by a researcher and once a month an association of the Research Center organised a drink to make researchers know each other.

Technical part

3.1 Objectives

My internship mission was to submit a method to the former WMH Challenge that is an improvement of Cian team work[1]. This mission shifted at the beginning of August when our supervisor find that we could apply our method to MRBrainS18 challenge that had its deadline in August 15^{th}. The two methods used for those challenges are described in MRBrainS18 and WMH section. Those two parts were born from the work of Florian, Corentin and I because each challenge required a two page description that is why to have the best accuracy possible the texts will be identical in Corentin's report and my report. Those two challenges deals with MRI's brain segmentation.

3.2 Tools

Several computer tools that are summed up in Figure 3.1 page 5 were required to be able to create this project : Ubuntu 18.04, Keras 2.2 [2], Tensorflow 1.9 [3], Python 2.7 and Docker [4]

Thanks to our experience in computer sciences learnt at the EMSE[4] and in preparatory years we knew how to use Python and Ubuntu and we learnt to use Keras 2.2 [2], Tensorflow 1.9 [3] and Docker [4] easily and quickly.

Figure 3.1: Internship useful tools

[4]Ecole des Mines de Saint-Etienne

3.3 Theory

3.3.1 Deep learning

Deep learning is a part of Machine Learning field. Deep learning methods are based on neural network. [7]. A neural is a cell that is linked to other cells and it holds a value between 0 and 1. A layer is an ensemble of neural that are not linked together but can be linked if it exists to the previous layer and if it exists to the next layer. As you can see in Figure 3.2 at page 6, this system transforms input layer into an output layer. In the studies of my internship, input and output layers were tiled 3D images.

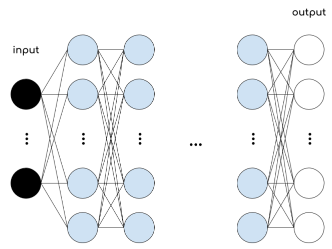


Figure 3.2: Neural network

3.3.2 Convolutional Gated Recurrent Unit.

A Convolutional Gated Recurrent Unit[1] (C-GRU) is a recurrent neural network defined by the following equations. You will find an explication diagram in Figure 3.3 at page 6. Γ_u, Γ_r, W_u, W_r, W_c, b_u, b_r, b_c, σ, $tanh$, $\circledast$, $h[x-1]$, $h[x]$, $h'[x]$, $i[x]$ and $o[x]$ respectively stand for update gate, reset gate, update weights, reset weights, computation weights, update bias, reset bias, computation bias, sigmoid function, hyperbolic tangent function, convolution operator, previous saved output, saved output, candidate, input and output.

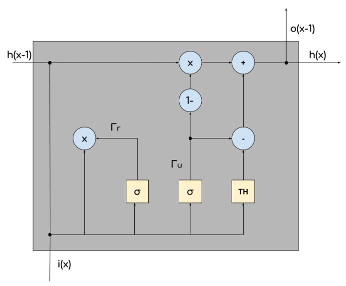


Figure 3.3: Convolutional gated recurrent unit

$$\Gamma_u = \sigma(W_u \circledast [h[x-1], i[x]] + b_u)$$

$$\Gamma_r = \sigma(W_r \circledast [h[x-1], i[x]] + b_r)$$

$$h‘[x] = tanh(W_c \circledast [\Gamma_r * h[x-1], i[x]] + b_c)$$
$$o[x] = h[x] = \Gamma_u * h‘[x] + (1 - \Gamma_u) * h[x-1]$$

The gates have the classic formula of convolutional neural networks. They are convolutional because our data are 3D images.The reset gate quantifies the importance of the previous saved output in the computation of the candidate. The update gate quantifies the importance of the candidate in the computation of the output. Because of the activation functions (σ and $tanh$) Γ_u and Γ_r have their values close to 0 or close to 1.

Like every recurrent neural network data must be processed throw a sequential dimension. Usely this dimension is Time. In our model, this dimension is spatial. In a way, our data can be seen as a video where we browse the brain the brain following a spatial dimension as described in Figure 3.4 page 7

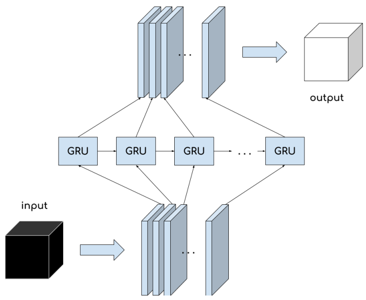


Figure 3.4: Application of a C-GRU

3.3.3 Multi Dimensional Gated Recurrent Unit.

A Multi Dimensional Gated Recurrent Unit (MD-GRU) is an ensemble of C-GRU that uses several spatial dimension forward or backward as illustrated in Figure 3.5 at page 7. The implementation of the MD-GRU was done with a modification of the source of a Convolutional Long Short Term Memory Layer. And in this implementation the sequential dimension is fixed, that's why rotations and reversed rotations are needed as described in Figure 3.5 at page 7.

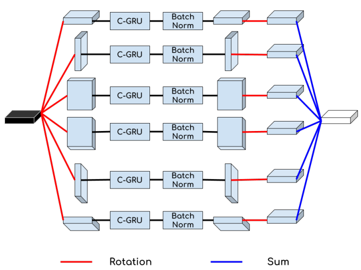


Figure 3.5: MD-GRU

3.3.4 Model

Our full model is based on several layers of MD-GRU linked by channel-wise fully connected layers. Three kind of MD-GRU were used during my internship. The first one have 6 C-GRU (each spatial dimension backward and forward), the second one have 3 C-GRU (each spatial dimension backward) and the last one have 4 C-GRU (y and z dimension backward and forward). Figure 3.6 at page 8 describes the most used model during my internship.

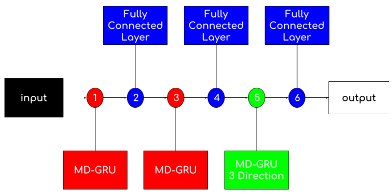


Figure 3.6: Most used model

3.4 MRBrainS18

3.4.1 Method overview

Method. Our method is an ensemble of Multi-Dimensional Convolutional Gated Recurrent Units (MD-GRU) [1]. MD-GRUs have significantly less parameters than commonly used U-net methods, which can avoid over-fitting on the training data-set. Furthermore, using MD-GRU mimics the slice by slice manual annotation process by considering a spatial dimension as a sequential dimension.

Material. The experiments were launched on four GPUs : one Nvidia GeForce GTX 1070, two Nvidia GeForce GTX 1070-Ti, and one Nvidia GeForce GTX 1080-Ti. The method uses Keras 2.2 [2] with Tensorflow 1.9 [3] as back-end.

Dataset. The data-set of the challenge contains the T2-FLAIR, T1-weighted (T1-w) and T1-weighted inversion recovery (IR) 3D scans of 7 subjects. We did not use any other data to train our algorithms.

3.4.2 Preprocessing

Brain extraction. The skull is removed using a 3D U-Net with 16 convolutionnal layers trained on the FLAIR and the T1-w scans. The ground truth is a pixelwise binarization (0 background, 1 brain structure). During training, we apply on-the-fly random translations and rotations to the images. The loss function is computed as the number of false positives plus ten times the number of false negatives to favor masks larger than the brain over smaller ones. Adadelta [5] is used as optimiser. The computed mask is then applied to the 3 modalities (FLAIR, T1 and IR). This method is described in Figure 3.7 at page 9.

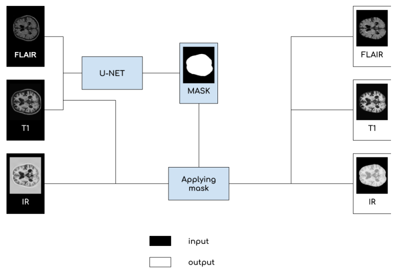


Figure 3.7: Preprocess MRBrainS18 challenge

Tiling. The images are split in tiles of 70 * 70 * 22 voxels both at train and at test time, with a 50% overlap, to fit in the GPU memory and give a more robust (averaged) prediction.

Normalization. To be robust to outliers a 1%-99% percentile normalisation is applied to brain-extracted data of each MRI sequence. Intensity values between 1% and 99% are consequently rescaled between 0 and 1.

3.4.3 Recurrent Neural Network

Model. The network takes a 3 channels array of tiles as an input : FLAIR, T1-w and IR ; and outputs a 9 channels array of tiles : cortical gray matter, basal ganglia, white matter, white matter lesions, cerebrospinal fluid in the extracerebral space, ventricles, cerebellum, brain stem and background, infarction and other. The architecture is composed of 3 MD-GRU layers linked with channel-wise fully connected layers, as in [1]. Our MD-GRU applies a non padded Convolutional Gated Recurrent Unit [1], with a 2D convolution followed by batch-normalisation in several directions before summing the outputs (6 parallel branches). The two first MD-GRU layers use the three spatial directions, both forward and backward, while the last MD-GRU layer only goes forward (3 parallel branches). The parameters are optimised with Adadelta [5] with Keras' default learning rate (1.0) and the activation function of the last layer is a softmax. This method is described in Figure 3.8 at page 10.

Training. Four models were trained with an averaged one-versus-all Dice loss on randomly selected train and validation splits. During training, on-the-fly random translations, rotations and flipping were used for the 500 first epochs, then random elastic deformations were added for the last 300 epochs. Training one model lasts 2 days on a single GPU.

3.4.4 Post-processing

Reconstruction. A reconstruction algorithm transforms the tiled output of a network into a full size output. Overlapping predictions are averaged.

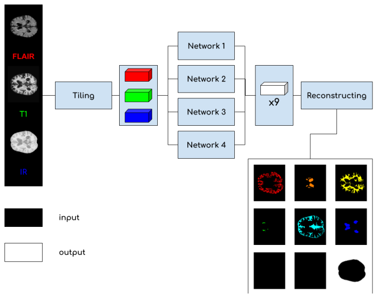


Figure 3.8: Process MRBrainS18 challenge

Table 3.1: Results of MRBrainS18 Challenge

Experiment	Dice	H95	Volume similarity
Network 1	0.789	6.68	0.896
Network 2	0.781	6.95	0.894
Network 3	82.9	4.49	0.938
Network 4	0.712	3.36	0.815
Network 5	0.739	4.41	0.821
Ensemble	0.836	4.49	0.935

Ensemble of 4 models. After reconstruction, the probabilistic outputs of the four models are averaged class-wise. For each voxel, the prediction is then the class with highest probability.

3.4.5 Results

Our results will be analysed with Hausdorff distance[8], Sørensen–Dice coefficient[9], recall[11], F_1 score[10] and avd score.

Now we only have the results on the valid set in Table 3.1 at page 10. We could not split our data-set into train, validation and test because it was too little (7 images). As a consequence, those results give us an idea of the rates we could obtain in the Challenge but they might be a bit over-estimated.

3.5 WMH

3.5.1 Method overview

Method. Our method is an ensemble of Multi-Dimensional Convolutional Gated Recurrent Units (MD-GRU) [1]. MD-GRUs have significantly less parameters than commonly used U-net methods, which can avoid over-fitting on the training data-set. Furthermore, using MD-GRU mimics the slice by slice manual annotation process by considering a spatial dimension as a sequential dimension, and is very adapted for anisotropy.

Table 3.2: Model architectures of the ensemble

	MD-GRU 1	MD-GRU 2	MD-GRU 3
1	y, z both ways	y, z both ways	y, z both ways
2	x, y, z both ways	x, y, z both ways	x, y, z forward
3	x, y, z both ways	x, y, z both ways	x, y, z forward
4	x, y, z both ways	x, y, z both ways	/
5	x, y, z both ways	x, y, z both ways	/

Material. The experiments were launched on four GPUs : one Nvidia GeForce GTX 1070, two Nvidia GeForce GTX 1070-Ti, and one Nvidia GeForce GTX 1080-Ti. The method uses Keras 2.2 [2] with Tensorflow 1.9 [3] as backend.

Dataset. To train our algorithms we used the bias filed corrected (BCR) FLAIR-w MRI and BCR T1-w registered with FLAIR of 60 subjects. We only used images available in the challenge's training set.

3.5.2 Preprocessing

For each modality, the skull is first removed with Brain Extraction Tool [6] with the factional intensity set to 0.4 and the vertical gradient set to -0.4. The images are then split in tiles of 70 * 70 * 22 or 72 * 72 * 24 voxels (depending on the depth of the network) both at train and at test time, with a 50% overlap, to fit in the GPU memory and give a more robust (averaged) prediction. Finally, to be robust to outliers a 1%-99% percentile normalisation is applied. Intensity values between 1% and 99% are consequently rescaled between 0 and 1.

3.5.3 Recurrent Neural Network

Model. A MD-GRU network takes a multi-channels array of tiles as an input (FLAIR or FLAIR and T1) ; and outputs a single channel array of tile. The architecture is composed of MD-GRU layers linked with channel-wise fully connected layers, as in [1]. Our MD-GRU applies a non padded Convolutional Gated Recurrent Unit [1], with a 2D convolution followed by batch-normalisation in several spatial dimensions before summing the outputs. The parameters are optimized with Adadelta [5] with Keras' default learning rate (1.0) and the activation function of the last layer is a sigmoid. This method is described in Figure 3.9 at page 12.

Training. Five models were trained with a Dice loss on randomly selected train and validation splits with different architectures referenced in Table 3.2 page 11. The first three models use FLAIR and T1 as input ; the last two use only FLAIR. During training, on-the-fly random translations, rotations and flipping were used. Training one model lasts 1 day on a single GPU.

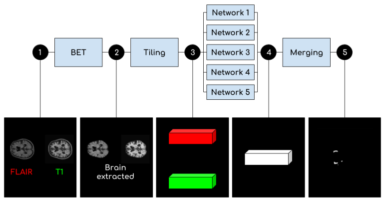


Figure 3.9: Process WMH challenge

3.5.4 Post-processing

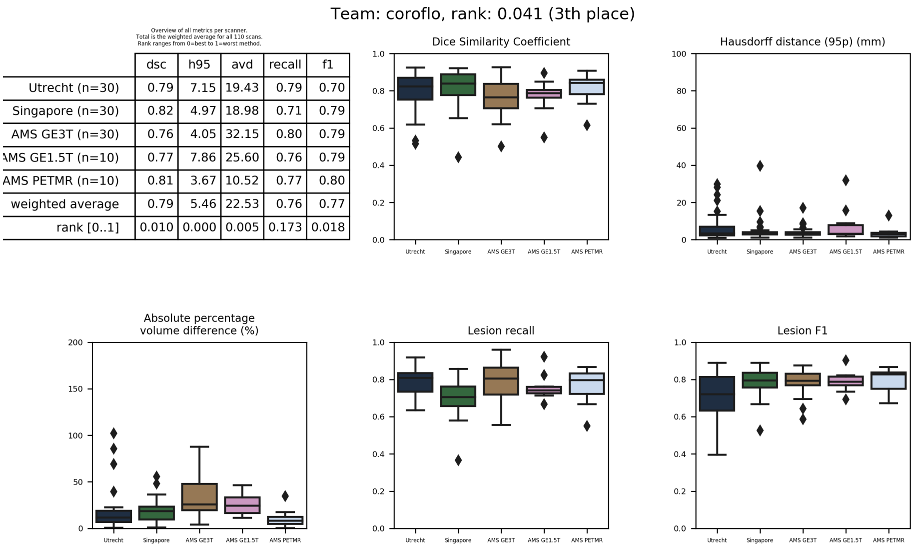
Reconstruction. A reconstruction algorithm transforms the tiled output of a network into a full size output. Overlapping predictions are averaged.

Ensemble of 5 models. After reconstruction, the probabilistic outputs of the five models are averaged. A 0.5 threshold is applied.

3.5.5 Results

Unlike the previous challenge, we had the results on a test set that was not used during the training of our models and you can find this result in Figure 3.10 at page 13. Our results like in the other challenge, were analysed with Hausdorff distance[8], Sørensen–Dice coefficient[9], recall[11], F_1 score[10] and avd score.

Our team finished 3$^{\text{rd}}$ at this challenge with the best result for Hausdorff distance[8]. This is promising for MRBrainS18 Challenge because the main advantage of our method is the few amount of parameters that prevents our network from over-fitting on the data-set. So our network should be more efficient on a challenge that a very few data like the MRBrainS18 that have 7 images.



	dsc	h95	avd	recall	f1
Utrecht (n=30)	0.79	7.15	19.43	0.79	0.70
Singapore (n=30)	0.82	4.97	18.98	0.71	0.79
AMS GE3T (n=30)	0.76	4.05	32.15	0.80	0.79
AMS GE1.5T (n=10)	0.77	7.86	25.60	0.76	0.79
AMS PETMR (n=10)	0.81	3.67	10.52	0.77	0.80
weighted average	0.79	5.46	22.53	0.76	0.77
rank [0..1]	0.010	0.000	0.005	0.173	0.018

Figure 3.10: Results of the WMH challenge

Acknowledgement

This research was funded by The Netherlands Organisation for Health Research and Development (ZonMw), Project 104003005.

Thank you to Gauthier Picard for accepting to monitor our internship.

We are gratefull to Marleen de Bruijne for her analysis, her kindness and her helfull indications.

A special mention to Florian Dubost for its implication and its wised leadership.

I would like to thank Arno, John, Marloes, Maria, Willem, Kim, Shuai, Jose, Thomas and Florian : my favorite BIGR Sloths for their warm welcome. It was really great to meet you all !

Conclusion

This internship gave me a unique opportunity to live a 3 month experience in the Research World and in the Artificial Intelligence field. Moreover the lab were I worked gave me one of the best formation in state of the art Deep Learning techniques applied to image analysis. We submit our codes to 2 International Challenges and finished in 3^{rd} position in one of them.

Furthermore, this internship is just the beginning of other researches. Indeed our supervisor Florian Dubost will apply our method to various data-set such as Erasmus MC and Boston data-set to study the influence of White Matter Hyperintensities on strokes. And we are still waiting for the results on MRBrainS18 Challenge.

To conclude this internship helps me to develop my autonomy, my adaptability and my knowledge in Artificial Intelligence. To go abroad made me learn how to work in a foreign culture with an international team and how to confront different points of view to improve the team results.

Bibliography

[1] Andermatt, S., Pezold, S. and Cattin, P., 2016. Multi-dimensional gated recurrent units for the segmentation of biomedical 3D-data. In Deep Learning and Data Labeling for Medical Applications (pp. 142-151). Springer, Cham.

[2] Chollet, F., 2015. Keras.

[3] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., 2016, November. Tensorflow: a system for large-scale machine learning. In OSDI (Vol. 16, pp. 265-283).

[4] Merkel, D., 2014. Docker: lightweight linux containers for consistent development and deployment. Linux Journal, 2014(239), p.2.

[5] Zeiler, M.D., 2012. ADADELTA: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.

[6] M. Jenkinson, M. Pechaud, and S. Smith. BET2: MR-based estimation of brain, skull and scalp surfaces. In Eleventh Annual Meeting of the Organization for Human Brain Mapping, 2005.

[7] Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y., 2016. Deep learning (Vol. 1). Cambridge: MIT press.

[8] Huttenlocher, D.P., Klanderman, G.A. and Rucklidge, W.J., 1993. Comparing images using the Hausdorff distance. IEEE Transactions on pattern analysis and machine intelligence, 15(9), pp.850-863.

[9] Sørensen, T., 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. Biol. Skr., 5, pp.1-34.

[10] Sasaki, Y., 2007. The truth of the F-measure. Teach Tutor mater, 1(5), pp.1-5.

[11] Powers, D.M., 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.