

Computer Science Tripos – Part II – Project Proposal

Brain tumour segmentation using Convolutional Neural Networks

S. Borgeaud dit Avocat, Fitzwilliam College

Originator: D. Wang

11 October 2016

Project Supervisor: Dr M. Jamnik & D. Wang

Director of Studies: Dr R. K. Harle

Project Overseers: Prof J. Bacon & Prof R. Anderson

Introduction

Over the last years deep learning, more specifically convolutional neural networks (CNN), have outperformed other machine learning techniques in many tasks such as image classification [?]. The field of bioinformatics is no exception to this, in particular, convolutional neural networks have been shown to perform as well as previous state-of-the-art algorithms on the problem of brain tumour segmentation [?].

The aim of this project is to use convolutional neural networks to replicate these recent results on the brain tumour segmentation problem. This project will concentrate exclusively on the dataset provided by the BraTS2013¹ grand challenge on which many different algorithms have already been tested and will provide a good framework to test and compare my work.

As a starting point I will follow the paper written by Pereira et al [?]. The approach used is to cut the magnetic resonance images into multiple patches and regard the problem as a classification problem of the pixel located at the center of the patch. The aim is to classify each pixel into one of these four classes:

1. Non-tumor
2. Surrounding edema
3. Non-enhancing tumour
4. Enhancing tumour

¹<http://martinos.org/ctim/miccai2013/>

Starting point

The starting point for this project is the part 1B course 'Artificial Intelligence 1' which provided a short introduction to machine learning. In particular multi-layer perceptrons, the sigmoid activation function, backpropagation and stochastic gradient descent training were introduced. These concepts are all reused in convolutional neural networks which add convolutional and pooling layers to conventional multi-layer perceptrons network.

I will be using the Keras library with Tensor Flow to create and train convolutional neural networks.

Keras is a library written for Python. Fortunately, I have used Python before in small side projects meaning that I won't have to spend time learning a new language.

As the problem is self contained and formulated purely as a machine learning task, none or very little biological/medical background knowledge is required.

Resources required

For this project I will mainly use my own quad-core computer which runs Mac OS X El Capitan. I accept full responsibility for this machine and I have made contingency plans to protect myself against hardware and/or software failure. In case of failure, I will be able to terminate my project using the MCS machines. Backups will be to my external hard disk. Once a week, I will also copy all files to my Google Drive to add an extra level of redundancy in case of hardware failure. All written code will be under version control using git and will be backed up on a private GitHub repository at least daily while working on it.

I will also need a computer with an external GPU to train the neural network in a reasonable amount of time. For this, I will be using the Cambridge High Performance cluster. Alternatively, if this is not possible, I will use a GPU that would be provided by the AI group.

Work to be done

The project breaks down into the following phases:

1. The first phase of the project will be mainly focused on research during which I will learn how convolutional networks work and read up on how they have been used on the brain tumour segmentation problem in different papers. I plan to complete the Stanford CS321n² course on convolutional neural networks that I have already started. I will also need to learn how to use the Keras and TensorFlow libraries and review some of the more advanced Python features that I haven't used recently.
2. The second phase will mainly be devoted to preparing the images obtained from the BRATS dataset. The images will need to be cut into patches each of which

²<http://cs231n.github.io/>

will have to be normalised. I will need to perform bias field correction as magnetic resonance images can exhibit non-uniformities that are the result of magnetic field variations rather than anatomical differences³. I will then need to perform intensity normalisation across the different images. Finally, I will need to add the correct label to each patch using the segmentation provided with the original training images.

3. The next step will be to use the prepared and normalised patches to train a convolutional neural network using the Keras library. This will require hyperparameter tuning using cross-validation to avoid overfitting. I will then construct segmented images using my convolutional neural network to delimit the different segments to get a visual result that is easy to interpret.
4. During the last step I will evaluate how well my convolutional neural network performed using standard methods used to evaluate classifiers such as the confusion matrix, recall, precision and Dice scores. This evaluation can be done for different hyperparameters. Because the dataset has been used many times before as part of the Grand Challenge⁴ I will also be able to perform a quantitative comparison with different segmentation algorithms also using convolutional neural networks as well as other algorithms using different techniques.

Success criteria

The main success criteria for this project will be to have an algorithm that performs brain tumour segmentation into the 4 different segments as discussed earlier.

The primary aim is to achieve similar results to those obtained by Pereira et al [?], hopefully achieving 90% of the accuracies obtained in the paper. This means achieving the Dice⁵ scores summarised in the following table, where ‘complete’ refers to the complete tumour region (including classes 2–4), ‘core’ refers to all regions except for the edema structure (classes 3–4) and ‘enhancing’ includes only the enhancing tumour (class 4):

complete	core	enhancing
0.79	0.74	0.69

Possible extensions

Due to the recent development of this area, this project naturally leads to multiple possible extensions:

1. This process of segmenting MRI scans is very slow as each scan has to be cut into patches, one per pixel, and each patch then needs to be classified. Recent techniques have shown that it is possible to classify all pixels of a patch at once, which would drastically improve the speed of the segmentation. A possible extension would be to try to improve the segmentation speed using the suggested technique. It would then

³<http://brainsuite.org/processing/surfaceextraction/bfc/>

⁴<https://grand-challenge.org>

⁵https://en.wikipedia.org/wiki/Sorensen-Dice_coefficient

be interesting to compare the performance of the faster algorithm to the performance of the original one.

2. Experiment with the layout of the neural network, in particular change the number of layers and the type of the layers of the convolutional neural network to try to improve the accuracy.
3. Instead of using the Keras library, implement similar functionality myself using TensorFlow. I could then compare the results obtained by my implementation with those obtained using Keras. This will show and require a deeper understanding of how convolutional neural networks work.
4. Use different data preprocessing/normalisation techniques and analyse how they affect the final accuracy of the convolutional neural network.
5. Apply more recent techniques used to improve convolutional neural networks such as Dropout [?], Maxout [?] or Batch Normalization [?] in order to improve the accuracy of the classification.

Timetable

Planned starting date is 21/10/2016.

Michaelmas term

1. **Weeks 3–4** Learn about Convolutional Neural Networks by finishing the CS321n online course. Read papers about using Convolutional Neural Networks for brain tumour segmentation.

Milestone: Understand the theory behind convolutional neural networks and be familiar with some of the more recent applications of them on the brain tumour segmentation problem.

2. **Weeks 5–6** Become familiar with the Keras library and refresh my Python knowledge. Download and play with the dataset.

Milestone: Be comfortable enough with Keras and Python to be able to start the main part of the project.

3. **Weeks 7–8** Prepare the dataset for the implementation of the convolutional neural network. This includes performing the different normalisations.

Milestone: Have the dataset ready, that is split up into normalised patches. Each patch should have the corresponding label for the pixel that is located at its center.

4. **Christmas Vacation** Implement and train a convolutional neural network and perform the necessary cross-validation on the different hyperparameters. Create segmented brain images using my classifier.

Milestone: Have a working convolutional neural network that is able to classify the different patches with the required accuracy mentioned in the primary success criteria. Have some images that are segmented using the trained classifier.

Lent term

1. **Weeks 1–3** Write the progress report and prepare the presentation.

Milestone: Have the progress report submitted on time and be ready to give the presentation.

2. **Weeks 4–5** Evaluate the performance of my segmentation algorithm and look for possible improvements.

Milestone: Have the evaluation data of my convolutional neural network.

3. **Weeks 6–8** Compare the performance of my algorithm with the benchmarks available online and implement some of the extensions if time permits.

Milestone: Have all the necessary data for the final evaluation of my project.

4. **Easter Vacation** Write the main chapters of the dissertation. Implement some of the extensions if time permits.

Milestone: Have a complete first draft of my dissertation

Easter term

1. **Weeks 1–2** Improve the dissertation where necessary

Milestone: Have the dissertation in its final form

2. **Weeks 3–4** Proof read the dissertation and submit it.

Milestone: Have the dissertation submitted.