TP3
Introduction to Deep Learning

Théo Estienne

Wednesday 29th January, 2020

Goal : Run and get familiarized with Tensorflow library and a very popular deep learning architecture for 2D segmentation of medical data.

Part I
Background

1 Architecture

Image Segmentation is one of the most important problems for medical imaging but also other communities such as computer vision, robotics, surveillance etc. Image segmentation aims to generate maps or volumes on which every pixel correspond to a specific class. Generating automatically volumes with accurate segmentations of different organs and/ or anomalies can potentially provide tools to help doctors on their everyday life. This topic was quite popular for a lot of years, however with the entrance of deep learning the performance of the algorithms have been boosted significantly.

In this exercise, we will explore a state of the art architecture based on 2D fully convolutional architecture, namely U-Net. U-Net is currently one of the most popular architectures for the segmentation of different organs and medical diseases. The architecture is based on an encoder-decoder scheme contracting path that follow the typical architecture of a convolutional network. In particular, it consists of the repeated application of two 3×3 convolutions (unpadded convolutions), each followed by a rectified linear unit (ReLU) and a 2×2 max pooling operation with stride 2 for downsampling. At each downsampling step the number of feature channels are doubled. Every step in the expansive path consists of an upsampling of the feature map followed by a 2×2 convolution ("up-convolution") that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3×3 convolutions, each fol-lowed by a ReLU. Figure 1 depicts the different components of the U-Net architecture, for more details check the paper [1].

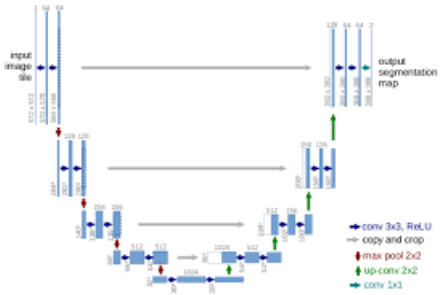

Figure 1: An illustration of the different layers of U-Net architecture.

To obtain and optimize the parameters of the network variety of loss functions can be used. For this exercise we will use one of the most commonly used losses for classification and segmentation problems namely categorical cross entropy.

$$Loss = -\sum_{i}^{C} t_i log(s_i)$$

[1] https://arxiv.org/pdf/1505.04597.pdf

where t_i and s_i are the ground truth and the CNN score for each class i and C is the set of the different classes.

2 Dataset

For this exercise we will train our model using the publicly available BraTS19 dataset [2]. In particular, the dataset consists of a multi-institutional routine clinically-acquired pre-operative multimodal MRI scans of glioblastoma (GBM/HGG) and lower grade glioma (LGG), with pathologically confirmed diagnosis. Annotations comprise the GD-enhancing tumor (ET — label 4), the peritumoral edema (ED — label 2), and the necrotic and non-enhancing tumor core (NCR/NET — label 1). An illustration of the dataset is presented in Figure 2

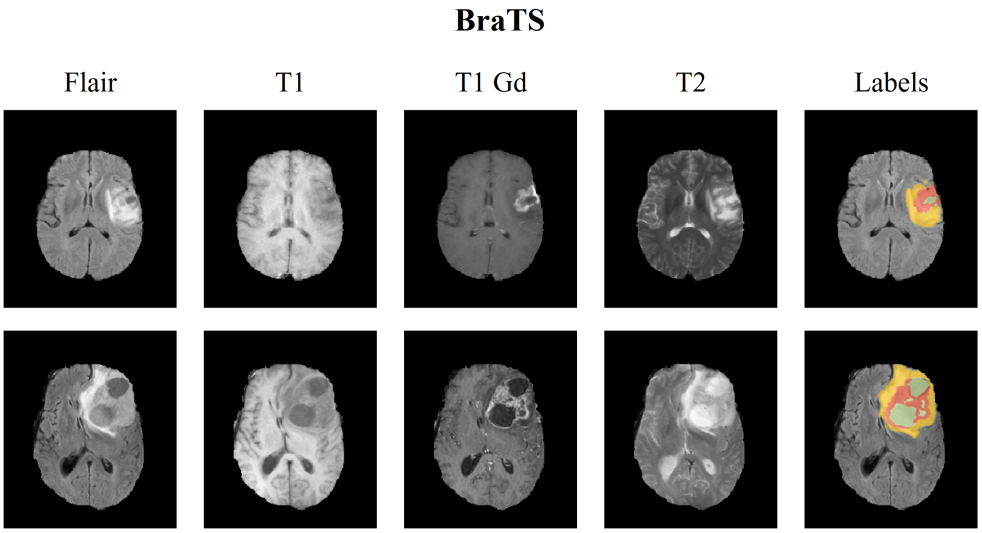

Figure 2: An illustration of two different subjects from the BraTS dataset.

Part II
Practical Exercise

3 Get familiarized with the data

The data is stored in the folder /content/TP3/data/. For a patient BraTS19_EXAMPLE, you will have a folder /content/TP3/data/BraTS19_EXAMPLE. Inside this folder you will have nifti files (.nii.gz) for each modalities and each z slice. The train, validation and test split are stored in the folder /content/TP3/datasets/. For each split, you will have a text files with a list of patient

Exercice 1 : Train, Validation and Test Set

Complete the following code to :

- Load the train, validation and test set
- Print the first 5 patients of the train set
- Print the lenght of the train, validation and test set

Exercice 2 : Content of the data folder

Complete the following code to :

- Print the first 5 patients in the data folder

[2] https://www.med.upenn.edu/cbica/brats2019/data.html

- Print the lenght of the data folder
- Select a patient and print the number of images inside the patient folder
- Print the number of z slices for one patient
- Print the 5 first images of a patient

Exercice 3 : Study of medical images

Medical images are stored in specific data type such as dicom or nifti. Theses data types contain not only the images but also medical information. There are many Python librairy to load medical data. We will use SimpleITK.

Complete the following code to :

- Print the first 5 patients in the data folder
- Print the lenght of the data folder
- Select a patient and print the number of images inside the patient folder
- Print the number of z slices for one patient
- Print the 5 first images of a patient

Exercice 4 : Comparaison between original data and processed data

In order to accelerate the calculation time and have good results quickly, we will work on small images of shape (96, 96). The original images of the dataset have a shape (155, 240, 240) where the 155 correspond to the number of slice on the z axis.

The step to pass from the original images to the images we will use are :

- Crop the images to a shape of (155, 192, 192)
- Downsample the images by interpolation of scale 0.5 [3] to a shape of (77, 96, 96)
- Save all the z slice independantly in a new array of shape (96, 96)

Complete the following code to plot side by side one slice of the original images and the preprocessed images. Print also the shape of the original images and the preprocessed images. You should use the functions : sitk.ReadImage, sitk.GetArrayFromImage, shape, plt.imshow, plt.title, plt.subplot

Change the z parameter to plot different slice of the images

4 Creation of the neural network

In this part, we will define our architecture and train our first network.

Exercice 5 : Create the network

- Complete the function DownConvBlock and UpConvBlock. You should use the different layers function from tensorflow which are imported in the code.
- Try to compare the following code with the Figure 1. Where are the DownConvBlock, the UpConvBlock?

[3] `https://scikit-image.org/docs/dev/auto_examples/transform/plot_rescale.html`

Exercice 6 : Study the model

To study and debug a neural network, keras offers to solution : the function summary() which print a summary of the model and the function plot_model which creates a graph of the neural network.

Execute the following code and answer the following question.

- Open the image model.png and compare it with the Figure 1
- What are the number of convolutionnal layer in our model ?
- What is the number of parameters in our model ?
- What is the smallest shape in the model ? Why ?
- What is the layer with the biggest number of parameters ? Try to find the formula to calculate the number of parameters.

Exercice 7 : Main - Run the model

To train the model in keras, you have different steps :

- Choose the optimiser, learning rate and loss function
- Create the model and compile it
- Define callbacks to save the model, plot it, change the learning rate and so on...
- Define the data in the form of a data generator or a numpy array
- Launch the training for a number of epochs

Complete the following code and launch a training for a small number of epochs (5 or 10 for instance). You should see the loss decreasing. Then open tensorboard to visualise the decrease of the loss and the prediction of the network at the different epochs

5 Data Augmentation

In this part we will use data augmentation to increase the performance of the network. The idea with data augmentation is to create new artificial sample so that the network see more data and overfit less.

Exercice 8 : Study of the data augmentation

Complete each of the following code to see the impact of the data augmentation. Compare the original data and the new artificial images.

Exercice 9 : Train with data augmentation

Relaunch a new training with data augmentation. The training will take more time than during the previous training.

Exercice 10 : Comparaison of training curves

The logs of two pre-trained for 200 epochs models are given in the folder /content/TP3/tensorboard_logs/. The only difference between the two models in that one is using data augmentation and the other not. Open tensorboard and compare the loss, accuracy and prediction for the two models.

6 Prediction and Evaluation

After training the model we have to perform two steps : the prediction of the segmentation using a model trained and the evaluation of the performance of the network.

The evaluation of the network should be done on the validation set (which have not been seen by the network during the training). The test set will only be used when all the parameters of the network are chosen.

Exercice 11 : Prediction and returning to original space

In this exercise, we will load a network and use the function predict() to calculate the predicted segmentation. The predicted segmentation will have a shape (96, 96, 4). We need to transform it back to the 3D shape (155, 240, 240) to perform the evaluation. The steps to calculate the prediction will be :

- Load an image X, with the function load_image
- Apply model.predict() on X
- Apply get_mask2original_shape() to the prediction

Questions

- Study the function load_image(patient). Try to understand what this function is doing. Try to guess the shape of all the different variable (temp_array, array, X). Then use the function print and shape to verify your guess.
- Study the function get_mask2original_shape(). What are doing the different functions inside it ? Try to guess the different shape

Exercice 12 : Visual Comparaison

We will do a visual comparaison between the predicted mask and the groundtruth. We will perform the comparaison with Matplotlib (python librariry for graph) and a software specialised for medical imaging vizualisation 3D Slicer [4].

Questions

- Load the groundtruth mask (in folder /content/TP3/origin_data/) and using the function plt.subplot, plt.imshow and plt.title, plot the predicted mask and the groundtruth side by side
- Change the value of the z_slice parameter to explore the segmentation slice by slice
- Study the function numpy2nifti. What does it do ? Apply it and save the predicted mask as a nifti files. Download the MRI, the groundtruth segmentation and the predicted segmentation and download the software 3D Slicer. Open the nifti files in 3D Slicer and compare the images.

Exercice 13 : Metrics

We will now calculate the evaluation metrics to assess the performance of our model. We will use 3 metrics : the sensitivity (also called true positive rate), the Specificity (also called true negative rate) and the dice score (different of the dice loss used during the training).

We will evaluate the metrics on the 5 patients in the folder /content/TP3/origin_data/. In the reality, we should evaluate the metrics for all the patient of the validation set (the test set will only be used when we choose the best model and we publish our results).

Questions :

- Using the previous predict_mask and orig_mask array, calculate the metrics. You just need to apply the function evalAllSample(). Print the result.
- Using, a for loop and all the functions defined in the part 4, calculate the metrics for the 5 patients in the folder /content/TP3/origin_data/. You should print the average Dice value for each category (WT, ET, TC)

7 Subsidiary Question

[4] https://www.slicer.org/