Theory

Given are the 3D medical images for 100 patients of the brain, and a tumor mask is provided for each image. Imagine that your task is to develop a neural network model that will solve the segmentation task.

1) Briefly discuss what steps you will perform to implement the solution and how you would evaluate the results of the model.

There are few steps to train a deep learning model:

a. DATA PREPARATION: image quality assessment and data integrity, i.e., it is fundamental to check if the images contain artifacts (for CT images we need to take in account all possible types of artifacts: hardware, physics or patient-based artifacts, such as motion, clothing and jewelry artifacts) or they are unnecessary for training the model, and in case of positive response exclude them from the data set. Once the data are ready for the model we need to split the data in 3 subfolders, train, validation and testing. The number of subjects for each folder can be selected in different way, for such scenario it is possible to use this splitting ratio 50/20/30, for train, validation and testing, respectively. However this splitting ratio can be adjusted accordingly with the data size of the data set.
b. DATA LOADER: create and test the data loader class/function. "Patching" approach in case of 3D model to allow the utilization of resources with limited computational capabilities, for example a discrete GPU without a large amount of memory.
   Data normalization: it is essential to normalize the data in the same range of values, commonly, [0,1]. Normalization techniques such as Z-score normalization or min-max normalization can be employed.
c. TRAINING:
   Learning rate: this is a critical parameter, and it has to be chosen carefully in order to allow the network to converge to the global minima. A high learning rate is inadvisable and most likely the model doesn't reach the global minima (due to the overshooting). On the contrary, a small learning rate could potentially lead to a local minima convergence, but, it is more favourable to let the model converge to the global minima, although this may take a very long training time.
   Network architecture: a 3D UNet model, or its flavours like Attention-UNet, UNet-MSS are the most common choices for the task of medical image segmentation.
   Activation function: for the current task, it is advisable to make use of the rectified linear activation function (ReLU) or even PReLU, parameterized-ReLU.
   Optimizer: "Adam" is the most widely used optimizer, but there are other optimizers such AdamW, AdaDelta can also be employed.
   Loss function: there are available several loss functions for the task of image segmentation, the most commonly used loss function for the task of image segmentation is a **pixel-wise cross entropy loss**. This loss examines *each pixel individually*, comparing the class predictions (depth-wise pixel vector) to our one-hot encoded target vector. However, the current experimentations show that using DICE loss or Focal Tversky loss yield better results.
   Batch size: this is strictly dependent on the resources available, GPU memory size.
   Number of epochs: until convergence.
d. TESTING or evaluation part: this is usually performed calculating DICE and/or intersection over union on the test set only.
e. FINE-TUNING: sometimes, the test set can have a different data distribution than the training set in which case fine-tuning the already trained model with a set closer to the testing set might improve the results.

2) You noticed that the score on the training set is 0.9, and the score on the test set is 0.6. The test set wasn't included in the training process. What actions will you take to improve the situation?

If training accuracy is much higher than testing accuracy (30% in this case) then, it is plausible that the trained model has overfitted. To avoid or limit overfitting there are the following techniques:

A. dropouts: it consists in dropping randomly connections between neurons, in this way it is possible to force the network to find new paths and generalize;
B. data augmentation;
C. regularization of data (L1, Lasso Regression or L2 Ridge Regression);
D. early stopping of the training.

Programming

Develop a generator(TF, Keras, or PyTorch) for a training dataset from DICOM dirs. You are not able to load all data to RAM. Use the following file as an example of DICOM [redacted] (Use several copies of the series). The solution must be written in Python.

Solution

In reply to this email, please, attach your answers to the theory questions and the programming solution (.py or .zip).

Deadline

24 hours after receiving the email.