Technical note for the liver challenge

This short report exposes the main problems the team Data Med Team faced during the JFR data challenge 2018 on liver challenge. Different options were tested and the main solutions and stages implemented are presented in the following.

1. Data exploration

First of all the data under study are ultrasound data in quite few number (less than 400 samples for the train set) and labelled with 8 classes. It appeared that the 8 classes are highly imbalanced (same classes have less than 10 samples). Another difficulty was pointed by our radiologist: the ultrasound is operator dependent, it means there is no standard cross-section for such an operation. Consequently this introduces much noise in the data.

Hence 4 issues needed to be tackled:

- Regularization technics to prevent over fitting
- Data augmentation to increase significantly the number of training data
- Over-sampling technics to make classes more balanced
- Tuning the neural network to obtain the highest possible accuracy

2. Pre-processing solutions

The first step for the pre-processing pipeline was to over-sample the classes which had the lowest samples. This was done by simply duplicating images. Thus we were back to a classic image classification problem with balanced classes. The second step was to use data augmentation technics with the library ImageDataGenerator (from Keras) such as rotating the image, flipping, scaling, shifting … Doing so we ended up with a total of 3000 samples of ultrasound data (instead of 400 initially) which seemed enough to us.

The pipeline was then usual since images were converted to 2D numpy arrays. These arrays were centered and reduced. The labels data were converted to one hot vectors.

3. Deep learning solutions

Dealing with an image processing classification problem, a convolutional neural network (CNN) classifier can bring very promising results. Given the few number of data provided, one should use anti over fitting techniques such as: cross-validation and dropout technique. The dropout technique is also considered as an efficient regularization solution.

The scientific formula of the 2-layer CNN chosen is exposed below:

$$y = softmax\{ ReLU(x * W_1 + b_1)W_2 + b_2\}$$

The first neural network is a convolutional network activated by a ReLU function. The second layer is a simple fully-connected graph. This classifier was implemented and tuned manually with Tensorflow. Because of time constraints we could not manage to practice fine tuning of the model.

Since we managed to get back to a balanced machine learning problem in terms of classes, we kept the accuracy score to track the improvement of the trained models. The loss function chosen was the cross-entropy function. Finally the network was fed with batchs of 40 samples since training iterations were not too computational costly.

Solène Lefèvre, Dali Ioudarene & Vincent Hardy

The CNN was trained with the 3000 samples in 500 iterations until we got a train accuracy of 0.80, a loss of 1.73 and a test accuracy of 0.67. This trained model allowed us to get a total AUC score of 0.76 on the test set we had initially chosen.

So now let's find out what score we really reached on the real test set released by the organization team today ☺ !