Ouadahi Fares
Cama Hector

LAB6 - Report

Task 1) : Regression problem – prediction of stock prices

The goal of this task is to introduce long short-term memory (LSTM) layers using a stock dataset. The goal is to predict the price of the opening of day D+1 knowing the price of the opening from day D-T to D with T the number of time steps.
We perform a regression with a simple model made of four LTSM layers with 20 units each.

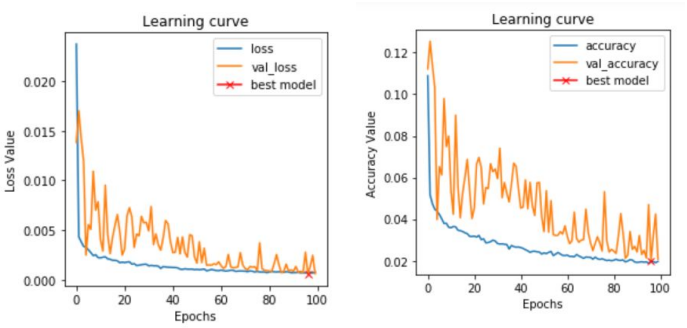

Figure 1 : Mean square error and mean absolute error for the model with 20 units LTSM layers ; 100 epochs

Then we try to double the units to 40.

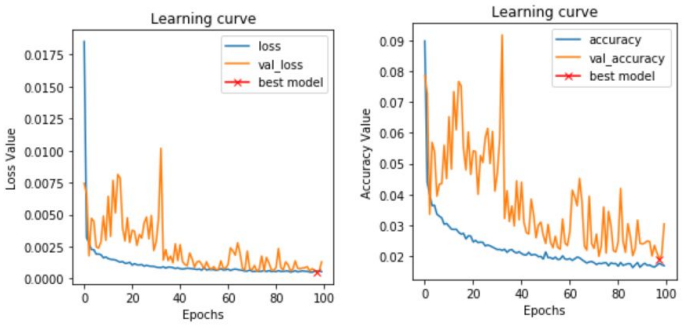

Figure 2: Mean square error and mean absolute error for the model with 40 units LTSM layers ; 100 epochs

The model is still good but there is no big difference between the two models. If we look at the prediction values it is a bit better with more units.

Task 2) : Classification problem – fiber bundle classification

This time the same model with LSTM layers is used to perform a classification problem, the classification of neural fiber tracts in the brain.
The layers have a particular characteristic; they are bidirectional because the input data are bidirectional. Moreover, the input size is not static so the batch size is set to one.

Then we obtained the following results for 50 epochs and a number of units decreased to 10.

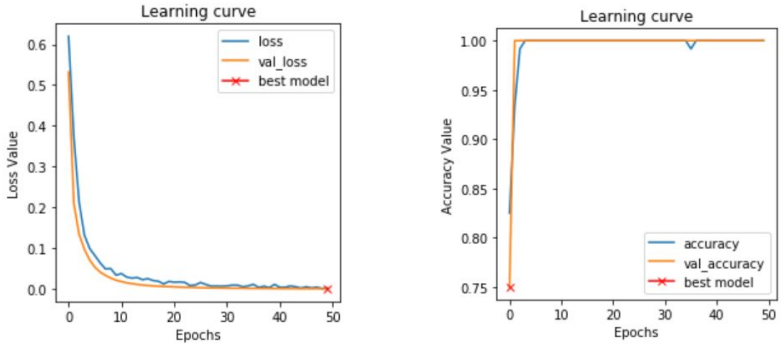

Figure 3 : Loss and accuracy curves for bundles classification ; 50 epochs

The perfect accuracy is obtained fastly. This is normal for this type of task. If we reduce the number of units we do not see any differences. It means that for a simple classification problem we only need one unit in the LTSM layers. If we use other bundles the task becomes more complicated.

Task 3 : Another U-Net extension

The final task was about adding LTSM layers to the U-net model before concatenating the layers from encoder and decoder. Because of the running time (very long) we chose to fit for 50 epochs.

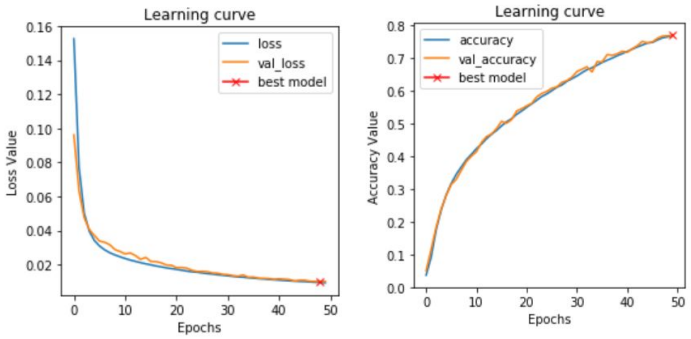

Figure 4: Loss and score curves for brain MRI segmentation with improved U-net with LTSM layers ; 50 epochs

The performance of the model is really good for this small number of epochs. We reach 0.8 accuracy and the loss values are very low.

Bonus task 1 :

See the code. We have implemented padding to avoid using batch_size=1. The results are the same as with the other method.

Bonus task 2 :

Even with a high number of classes the model accuracy reaches 1 after a few epochs.