AMAL - Master DAC (**due on November 30th 2021**)

Practical Works № 4 – 6

Professor: Patrick Gallinari

Student names: Pablo López Rivera, Patrick Lutz

Practical 4

The subject of this practical work was focused on Reccurrent Neural architectures, which are mainly designed for "sequential" data (time series, progressions, etc.). The first exercise consisted on coding a recurrent architecture, which is done as the class `RNN`, subclass of `nn.Module`, in the module `utils.py`.

For the second question, it was asked to apply the architecture coded in the first exercise, to classify metro stations of Hangzhou's network, based on the in and out flux of passengers. We tried lots of combinations of the hyperparameters (size of hidden states, learning rate, batch size), and the best result was for a RNN made of hidden units of size 100, a learning rate of 0.01 and a batch size of 32. The training curves of accuracy and lost can be seen in figure 1.

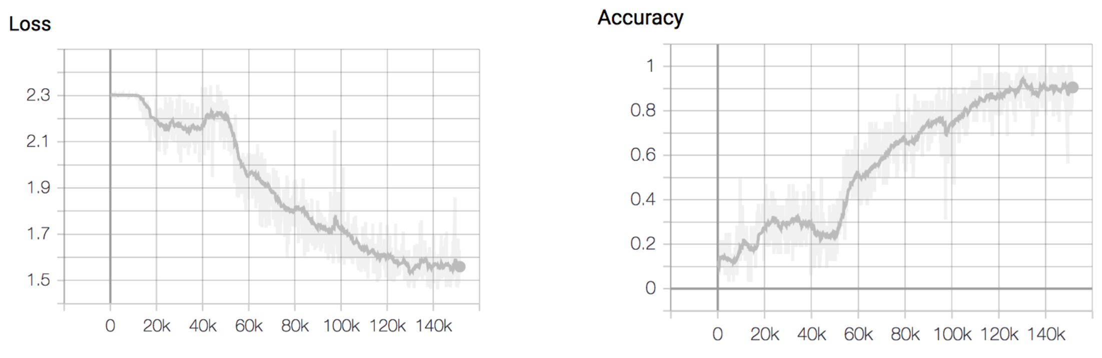

FIGURE 1 – Accuracy and loss for the training set.

On the other hand, this model was trained for 500 epochs; initially we intended for every model a training made of 1000 epochs, but we stopped the process if there was no significant improving of the accuracy.

Practical 5

In this practical we aim to continue the study of RNN, by using LTSM and GRU's. In the first exercise, is proposed to define a function `pad_collate_fn()`, in order to do a "pretreatment" to the phrases contained in the file, adding padding and end of file characters.

The second question, is oriented to code a proper version of the Cross Entropy criterion, in order to consider the padding formerly added by the exercise 1.

After that, for the third question is proposed to code the function `generate()`; in the fourth question is demanded to tune the code in order to use embeddings, via the module `nn.Embeddings`.

Practical 6

The first part of this practical aimed at implementing a the RNN variant long short-term memory in order to predict the part of speech for each word in a phrase. We used a linear encoder to reduce the input size of this network to 30. Figure 2 shows the convergence of the RNN. We achieved an accuracy of 87.7%.

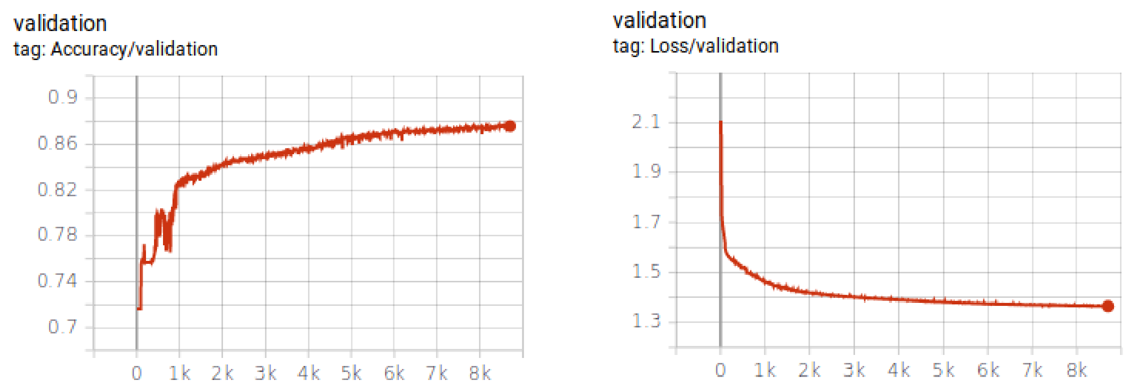

FIGURE 2 – Accuracy and loss on the validation during training for the part-of-speech problem.