Assignment 1: POS Tagging

Serban Cristian Tudosie
serban.tudosie@studio.unibo.it

Francesco Vannoni
francesco.vannoni2@studio.unibo.it

Mirko Del Moro
mirko.delmoro@studio.unibo.it

Abstract - Recurrent Neural Networks have been shown to be very effective for tagging sequential data, e.g. speech utterances or handwritten documents. In this study, we propose to use several Neural Network models for part-of-speech (POS) tagging task. The different combination of models exploits one or more RNN layers starting from the GloVe embeddings of the sentences words.

1 INTRODUCTION

Our task is to classify and label words from sentences into their parts of speech tag.
We start from a dataset provided by NLTK containg 200 dependency treebank files. We prepare the dataset in order to be trained on sentences. By using regular expressions we clean the file from unwanted blank spaces, tabs and noisy characters. After that we split the dataset into train, validation and test. The first 100 documents from the dataset are used as train set, the following fifthy as validation set and the remaining as test set.
Analyzing the dataset we find out that the average length of the sentences is around 25 and the variance is small. Therefore we look for outliers and we decide to remove from the training and validation set those having a length higher than 80.(RIGIRA CON DATASET NON PULITO E GUARDA QUANTE SONO)
Sentences are then tokenized by assigning to each word an identifier integer. Since we have variable length sentences we pad each sequence and we reserve the 0 token for the padding. After that we compute the embedding matrix for sentences terms. To get the embedding we use GloVe vocabulary but we have to handle the dataset's terms not contained in GloVe. Out of vocabulary embedding has to be affected only by the words of the split it belongs to, while it has to be independent from the words of the other splits.
We compute OOV terms embedding by averaging on the embeddings of all their previous and next word from the sentences of the split the words belong to. If all the neighbours are OOV terms the average can't be computed so the embedding elements are picked from a uniform distribution. We decide to assign the same kind of embedding also to those terms having only one neighbour contained in the vocabulary, otherwise the two embedding would have been the same. This process must be done in order for train, validation and test. The result embeddings are concatenated to the embedding matrix and the terms are added to the vocabulary. This allows to have a embedding matrix where if a word is present both in train and test set and not in GloVe vocabulary, the embedding is computed as OOV in the handling of the training set OOVs and it is considered as a vocabulary word from the test set.
The embedding matrix is used in order to train four models with different architectures shown in the next section (Sectio REF ALLA SEZIONE MODELLI). Those models are made of combinations of layers but all of them contain at least one RNN layer.
We train the models for 10 epochs with a batch size of 32 and we evaluate them on the validation set by using the accuracy score and the confusion matrices in order to select the two best models. Then we perform Bayesian Optimization hyperparamer tuning on the best model selected. In particular we try several combinations of number of units and l2 regularizer value. Lastly we train the best models for 50 epochs with a batch size of 32 and using "Early Stopping" in order to prevent overfitting. Then we evaluate them on the test by using the f1 score not considering the punctuation tags.

2 MODELS

We try four model architectures in order to obtain POS tagging:

1. two layers architecture: a Bidirectional LSTM layer and a Dense layer on top (baseline)
2. two layers architecture: a GRU layer and a Dense layer on top (gru)
3. three layers architecture: a Bidirectional LSTM layer, a LSTM layer and a Dense layer on top (two lstm)
4. three layers architecture: a Bidirectional LSTM layer, two Dense layer on top (two dense)

The inputs are the tokenized sentences of the training set, the outputs are the are the POS labels. The activation function of the top Dense layer is "softmax"; in the last model (where there are two dense layers) the other dense layers has "Relu" as activation function. As shown in the Section METTI INTRODUZIONE the l2 regularizer value and the number of RNN units is tuned throught a Bayesian Optimization hyperparameter tuning.
The optimizer chosen is Nadam with an initial learning rate equal to 0.01; the loss function is "categorical crossentropy". In order to prevent overfitting we exploit "Early Stopping" monitoring the accuracy on the validation set: if it doesn't decrese for four steps, the training is interrupted.

3 RESULTS EVALUATION

In this section we are going to evaluate the findings obtained. The left histogram shows the scores obtained on the validation set by computing F1 macro metric. As we can see in the Figure FIGURE1 REF, baseline and gru are the models with the highest scores on the validation set. We train gru and baseline for an higher number of epoch (50) and we use the resulting models on the test set.

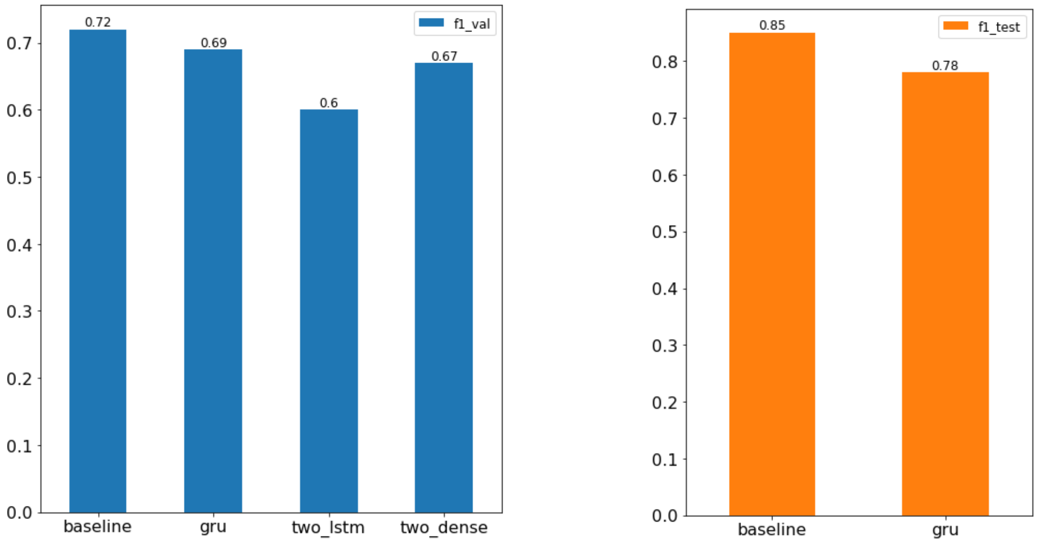
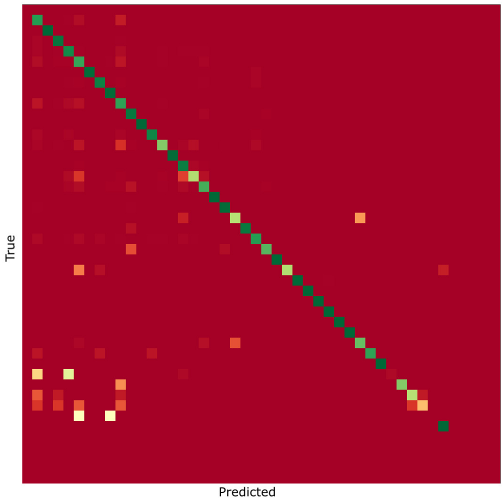


Figure 1: Models F1 score on validation set (left histogram). Best Models F1 score on test set (right histogram)

As expected the best one is baseline, the one having an higher number of parameter. Both models didn't train for fifty epochs because Early stopping interrupted the train after respectively 30 and 27 epochs.
Analyzing epoch by epoch the scores on the training set we can observe the gru is faster to converge. Maybe two lstm is been penalized by this property of gru and it could overtake gru performances by training on an higher number of epochs than 10.

4 ERROR ANALYSIS



As shown in the previous section we reach higher performances but there is still a percentage of errors. Therefore we try to find where the errors are. The figure on the left shows the heatmap of the confusion matrix computed on the test set. If an element is on the diagonal it means it is a correct prediction. The darker the green, the higher is the percentage of correct prediction. In the lower part a lot of rows are completely red because the corresponding POS tags are not present in the test set. Above that we can see some mistakes. The tags involved are: "