SNLP exercise set 05: word class prediction with neural networks

Ç. Çöltekin — SfS / University of Tübingen

Jul 28, 2017

In this set of exercises, we are going to use convolutional and recurrent neural networks for predicting whether a given word in German is a noun or a verb. As formulated here, this is a toy example. However, understanding this simple examples well will allow you to apply them to many other NLP tasks.

Data

The data for this exercise is from from the TüBA-D/Z treebank. The data is simply a list of words. Each word is either a *noun* or a *verb*. Data is presented in a tab-separated format, where the first field is the word class and the second field is the word itself. A short segment from the data file is presented in Figure 1.

You can get the data for this exercise from the course web page.[1]

```
noun aalstrich
noun abänderung
verb abarbeiten
noun abarbeitung
verb abdrehen
verb abfallen
```

Figure 1: A short excerpt from the data to be used in this exercise.

[1] http://www.sfs.uni-tuebingen.de/~ccoltekin/courses/snlp/

Exercises

Exercise 1. Read the data, map each Unicode character to a unique integer. and map each word a list of integers that correspond to their characters. Pad all integer arrays representing words with zeros such that all words are represented with the zero-padded integer sequences with the length of the longest word.

Reserve 10 % of the data as test set. Make sure that the class distribution is similar in both training and test parts of the data.

Exercise 2. Train a neural network with embeddings as the first layer, followed by a convolutional layer, a pooling layer that takes a single maximum of each filter defined by the convolutions, which is followed by a binary classifier. Experiment with a single layer classifier, as well as using multiple dense layers.

Tune the following hyperparameters:

- Embedding dimension
- Number of convolution filters
- Convolution filter length
- Number of hidden layers, number of units at each layer in the convolution

Exercise 3. Repeat the Exercise 2, but use a GRU layer instead of convolution/pooling. Tune the relevant parameters.

Exercise 4. Test and compare your best performing CNN and RNN networks from the previous two exercises on the test data reserved in Exercise 1.

Exercise 5. Calculate the hidden representation of the GRU network for 10 random verbs and 10 random nouns.

- Calculate the average cosine distance between/within word classes
- Transform the data using PCA, and plot and inspect first two principle components