Exercises
Natural Language Understanding
Spring 2018

Machine Learning Institute
Dept. of Computer Science, ETH Zürich
Prof. Dr. Thomas Hofmann
Dr. Massimiliano Ciaramita
Web http://www.da.inf.ethz.ch/teaching/2018/NLU/

Series 5, March 26th, 2018 (A simple PoS tagger in tensorflow)

The Model We want to build a simple feed-forward neural network to perform PoS-tagging. That is, given a word w_t and m previous words $w_{t-m}, \dots, w_{t-1}$ we want to predict the PoS tag y_t. We choose a simple approach consisting of the following steps:

- Embed words as vectors $x_i \in \mathbb{R}^d, i \in \{t-m, \dots, t\}$.
- Concatenate the target word vector and all vectors in the window into a single feature vector $x_t^{(1)} \in \mathbb{R}^{(m+1)d}$
- Obtain a hidden representation $x_t^{(2)} \in \mathbb{R}^{d'}$
- Apply a softmax over all PoS tags
- Optimize wrt. the cross entropy loss between our predictions $p(y)$ and the label $y^\star$

The Implementation

- Make yourself familiar with tensorflow [1] and install [2] it on your local machine (no GPU required, nor recommended). Please try to find solutions regarding installation on the web – most issues are common ones. Please understand that we cannot answer general tensorflow questions. You can be sure, the web is full of answers.
- Download the skeleton code[1] at [3]. It does some basic steps like loading the data for you and provides a simple code structure. You are welcome to change or ignore it.
- Download and unpack the two datasets (small and medium) at [4].
- Complete the implementation by following the hints in the code skeleton.

Experiments The purpose of this exercise is to get you familiar with tensorflow and deep learning routines. Start with the small dataset for training the network on your own machine[2]. The model is too simple and the data too small to see stellar[3] results, yet you should be able to see effects of different architectures. The following experiments are recommendations and we encourage you to try whatever you feel curious about.

- Run your model for $d = 50, m = 3, |V| = 20000$ using a single hidden layer of the form
$$x_t^{(2)} = \mathrm{relu}(W x_t^{(1)} + b)$$
. What accuracy do you get on the validation set? Use tensorboard [8] to watch the curves...
- Increase m (and optionally d) and watch training and validation error. Do you see overfitting?
- Increase $|V|$ to 50000. How does the accuracy change?
- Add another hidden layer with dimension d'' and vary the dimensions d' and d''. What effect do you observe on training and test accuracy?
- Vary the window size and plot the validation accuracy over m. When is the model saturated?
- How much information is in the target word w_t? Remove x_t from the feature vector to find out.
- How much information does the window carry? Find our by providing no window at all.

[1] Written for tensorflow version 1.6.0

[2] Feel free to use the large dataset. Depending on your memory, you might need to change the input processing to hold the corpus in memory only once and create `x,y` batches on the fly. You might want to decrease `dev_sample_percentage` as well.

[3] Feel free to check out [7] and use standard training sets such as WSJ.

Using the EULER cluster (CPUs) If you are curious to run longer experiments, you can ssh to and work on the EULER cluster. Please take a look at the wiki on how to use EULER [5], in particular on how to request certain amounts of memory and computing power using the batch system [6].
Run

```
module load new python/3.6.1
```

to get a running python tensorflow implementation (version 1.3)[4]. Before allocating dozens of cores, use `bjob_connect` and `top` to investigate how many cores tensorflow is actually using. Typically this not more than four. In any case you **must** set `inter_op_parallelism_threads` and `intra_op_parallelism_threads` in tensorflow to match the number of cores that you ordered when submitting jobs. The admins keep an eye on this.

References

[1] https://www.tensorflow.org/get_started/premade_estimators

[2] https://www.tensorflow.org/install/

[3] http://www.da.inf.ethz.ch/teaching/2018/NLP/material/pos-tagger.zip

[4] https://polybox.ethz.ch/index.php/s/pvSDsyyqRlDkaH3

[5] http://brutuswiki.ethz.ch/brutus/Getting_started_with_Euler (no registration required)

[6] https://scicomp.ethz.ch/wiki/Using_the_batch_system

[7] https://aclweb.org/aclwiki/POS_Tagging_(State_of_the_art)

[8] https://www.tensorflow.org/get_started/summaries_and_tensorboard

[4] If you want a newer version of tensorflow, locally install the package virtualenv with `pip3 install --user virtualenv` and follow the instructions in [2]