

Deep Learning - 89687

Ass2 – Part 3

The tagger of part3 is similar to the tagger of part1. We simply add an embedding/no embedding behavior. We just have to choose.

We add the vocabulary file and the vector words in the parameters as arguments.

POS

The parameters we use are:

- Adam Optimizer
- One hidden layer, as required, with 150 neurons
- Learning rate = 0.01
- Number of epochs = 25 (we stop when it converges).
- Batch size = 512 (to be faster)
- Loss = Cross Entropy (as required)
- Activation function = tanh (as required)

Accuracy = 93.5%.

NER

The parameters we use are:

- Adam Optimizer.
- One hidden layer, as required, with 100 neurons.
- Learning rate = 0.01
- Number of epochs = 25 (we stop when it converges).
- Batch size = 1024
- Loss = Cross Entropy (as required)
- Activation function = tanh (as required)

Accuracy = 74.5%

We get a very good accuracy from the first epochs

Pre-trained vectors

Pre-trained word vectors were given for a specific set of words.

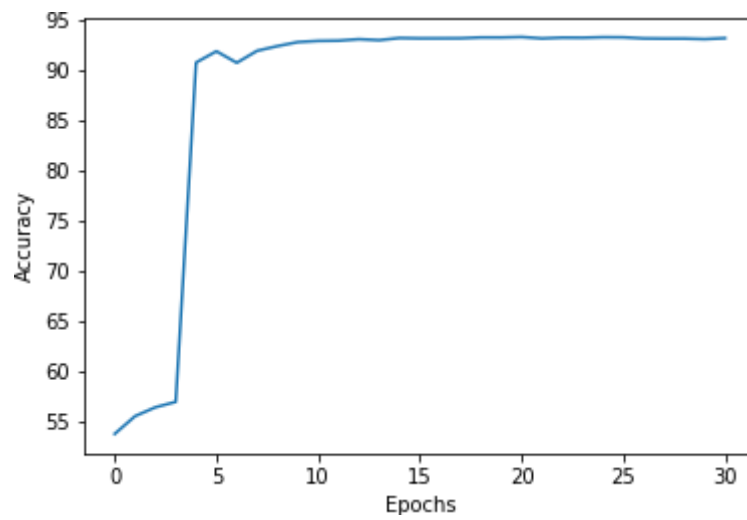
If a word is in the train set but not included in the pre-trained word vectors, or if a tag is not in the pre trained dictionary then we add them to the dictionary with a uniform initialization (-1, 1).

We also took into account the lower casing. Words that were not in lower case, we the Python function lower(), to turn the letters to lower cases.

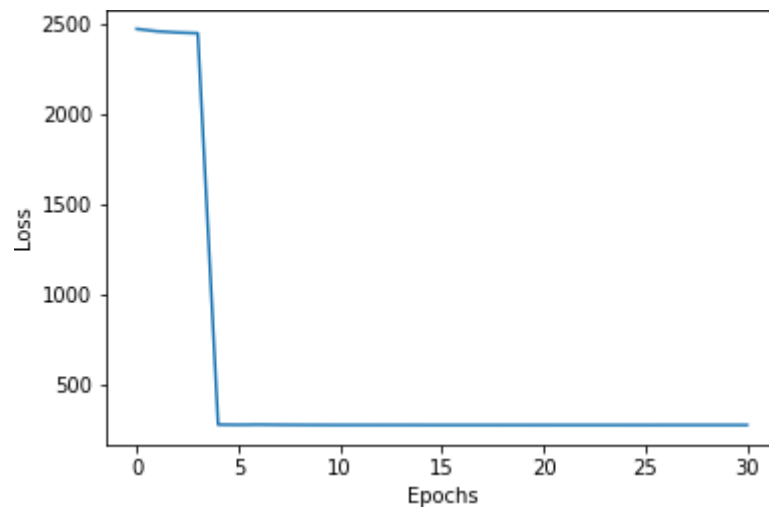
GRAPHS

POS accuracy:

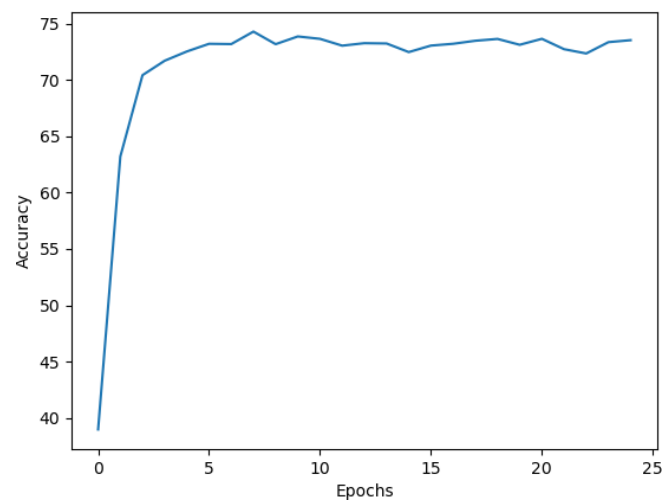
Lea Setruk
Yoel Benabou



POS Loss:



Ner Accuracy:



Lea Setruk
Yoel Benabou

NER Loss:

