Natural Language Processing Lab 9

Moritz Bohm, 140164022

1 Model Description

The language for this lab is the small dataset containing 5 sentences. Each sentence is then provided with a START and END token, so that they can be differentiated by the neural network. Upon entering the epoch loop, the inputs are prepared to be passed into the model, they're turned into indices and wrapped into variables. The words are then converted into word embeddings, with a dimensionality of 10. As such, each word is represented by a vector of size (1,10). The input layer for the neural network is the context for the word, containing the two previous words, stored in a vector of dimensionality (2, 10), which is flattened to dimensionality (1, 20). The inputs are then projected onto the hidden layer, by multiplying each input vector by a weight vector of dimensionality (20, 128). The result is then passed into the relu() function, which makes the results non-linear. This is what forms the "hidden layer" of the neural network, containing the first outputs, with dimensionality of (1, 128). These firs outputs are then multiplied by a second weight vector, with dimensionality (128, 17), with 17 being the size of the vocabulary. This results in the final outputs for the model, which have a dimensionality of (1, 17). The mathematical equation of this model is as follows:

$$I \cdot W_1 \leftrightarrow (1, 20) \cdot (20, 128)$$

$$O_1 = Relu(I \cdot W_1)$$

$$O_1 \cdot W_2 \leftrightarrow (1, 128) \cdot (128, 17)$$

$$O_2 = log_{softmax}(O_1 \cdot W_2) \leftrightarrow (1, 17)$$

$$P(X_n | X_{n-2}, X_{n-1}) = argmax(O_2)$$

2 Sanity Check

For the sanity check of the model, I implemented a dedicated loop, looping through the sentence "The mathematician ran to the store ." to see that the model predicts words correctly and with consistency. Upon tuning the hyper-parameters (Epoch number and learning rate) I find that an epoch of 85 and learning rate of 0.075 yields a 100% accuracy upon 5 repetitions of the sanity check. Setting the epoch number to 500, with a learning rate of 0.075 does not improve results, and thus it can be assumed that the epoch number ceases to have an effect after a threshold. Reducing the epoch number to below 28, with a learning rate of 0.075, leads to a reduction in accuracy. In terms of learning rate, with an epoch number of 85, the accuracy begins to decrease once the learning rate is below 0.0009, or above 0.3, the accuracy of 100% is lost. The reason that the model predicts the word "mathematician" instead of "physicist" is due to the fact that the word mathematician is the most common to follow "the", and as such will have the highest log probability.

3 Test

In order to test that the neural network successfully predicts the correct word, I use a measure of cosine similarity to compare whether "physicist" or "philosopher" is closer to the word which should actually

fill the gap, ”mathematician”. This is done by comparing the word embedding vectors for each word. As ”physicist” should be closer to ”mathematician”, you would expect to always have ”physicist” as the predicted word to fill the gap. However, upon running the test 5 times, ”physicist” is chosen 4 times, and ”philosopher” once. This would not work on the bigram ML from lab 2, due to the fact that the bigram ML will simply select the word that appears more often in the vocabulary, disregarding a measure of semantic similarity (which is what is used here).