For NLP course, 3rd assignment, we need to create a neural network for word prediction. In order to implement neural network in Python environment, Dynet framework used. For dataset, Trumph’s speeches are used.

Before beginning the implementation, whole dataset read by Python. The dataset consist of 163475 words. Then, a word vector is created using unique words by removing duplications and non-string words in dataset which is 16247 words.

For the neural network, 2 word vectors are used for input and a single word vector is used for output since we are building a bigram model. Also, one hidden layer is required in order to make the network simpler. However, the hidden neuron number is not defined in assignment. In theory, the hidden neuron number should be two times of input. Since 2\*16247 = 32494 hidden neurons is not feasible and generates memory error in my personal computer. Due to this fact, I used 162 hidden neurons which are around 0.01 \* 16247.

Also, in theory network should stop its training process when error is not decreasing enough, or validation error increases. However, for this case, this network cannot be trained for lots of epochs since an epoch takes a lot of time. For this case, I only trained for a single epoch since the hidden neuron number is large enough.

For activation functions in hidden layer, tangent hyperbolic is used. For the output layer, neurons are linear. Since the computation graph equation is;

Y = tanh(W\*X)+B where Y is output, X is input, W is weights and B is constant term

For the error function, squared distance is used since our inputs and outputs are all vectors. For the case which is explained in previous sections, the epoch error for the network is 0.6229877259568916. In first assignment, the smoothed MLE for bigram model that I generated is around 7.774864795101213e-07. Although, neural network seems worse than bigram language model, it is quite high result for a single epoch. Since the computation power of my personal computer is limited, the network is generated by using 162 hidden neurons and trained for a single epoch. If the network is created with larger hidden neurons, at least for 2 times higher than input number and trained for several epochs, the results would definitely be higher and accurate.