Lab 5: Neural Language Modelling

Registration Number: 180128251

April 29, 2019

Neural Network Language Model Description

The model implemented is a Multi-Layer Perceptron that has one hidden layer. It takes as input the index of words in the vocabulary, that are used to perform a look-up of their word embeddings/features and fed into the network. These features are sent to a layer that has 6 inputs and 128 outputs and used Rectified Linear Unit(ReLU) as an activation function. The output of this layer is then passed on to a final layer with 128 inputs and the vocabulary size (in this case 17) as an output with a softmax function. The model learns to predict the target word given a context of words. Since this model uses a n-gram language model where n=3; our context size is 2. The layers are basically linear models each of the form:

$$y = x * W^T + b$$

where x indicates the input and y indicates the output with b being the bias and W being the weights of the model. The model designed is therefore of the form:

$$y = b + Wx + ZReLU(d + Hx)$$

where **b** is the output layer weights, **d** - the hidden layer biases, **Z** - the hidden-to-output weights, **H** - the hidden layer weights and **W** - the output layer weights. The input vector **x** are the word features that consist of a concatenation of the different word features from the embedding layer of the context.

	Input	Output
Input Layer	2	6
Hidden Layer	6	128
Output Layer	128	17

Running a Sanity Check against the model

The model was trained using a learning rate of **0.03** and **400** epochs. After which, it was tested on its ability to correctly predict data from the training set. Using a context size of 2 (as was used to train our model), the model was able to predict correctly every the last word for every trigram in the sentence: "The mathematician ran to the store .". It should also be noted that, although the model predicts correctly the word "mathematician" as expected, it should also very well predict words like "physicist" or "philosopher" since they both have the same contexts (i.e. <s> The; where <s> represents the start of the sentence). However, because the word "mathematician" occurs more frequently in our training data than the other two, the model gives a higher prediction value to "mathematician" as opposed to "physicist" and "philosopher".

The sanity check was run five times to ensure that the model was well-trained to produce the same outcome every time.

```
Context: <s> The |=> prediction: mathematician
Context: The mathematician |=> prediction: ran
Context: mathematician ran |=> prediction: to
Context: ran to |=> prediction: the
Context: to the |=> prediction: store
Context: the store |=> prediction: .
Context: store . |=> prediction: </s>
```

Testing the model

The model was tested by providing it with a sentence with a gap, where it is to fill in the gap with a more likely word. The sentence being "The _____ solved the open problem.", with possible options being either "physicist" or "philosopher". The correct answer for this is, is "physicist". Using the bigram ML model from lab 2, that employed the use of probabilistic models, would not be able to correctly choose between either of the two because the model won't have the bigram "physicist solved" and "philosopher solved" in it and they will both have the same probability values for "The physicist" and "The philosopher" making prediction of the missing word difficult.

The Neural Network Model was able to predict the most appropriate word to be "physicist" after using the words in question ("philosopher" and "physicist") as context words to predict a known target word and using the probability of the target word to compute the overall probability of the sentence as a way of determining the model's capability of choosing the right word between *physicist* and *philosopher*. This is expected as per our training data, it can be observed that, *physicist* occurs in a similar context as *mathematician*. The similarity comparison can be done by comparing the cosine distances between the tensor value or *mathematician* and physicist being 0.793 and that of mathematician and philosopher being -0.234. Based on these values, it is noted that the embedding for *physicist* is closer to that of *mathematician* than that of *philosopher* since the value of physicist is close to the value of 1.

```
[*] Prediction from model is physicist
Cosine similarity between mathematician and philosopher: -0.234166
Cosine similarity between mathematician and physicist: 0.793120
```

Conclusion

A **Neural N-gram Language Model** proves to be a better language model as compared to a **Probabilistic N-gram Language Model** introduced in lab 2, since it can perceive similarities between words based solely on the context they appear. In this lab however, a lot of manual tuning of hyper-parameters had to be done to achieve our desired results which could be quite problematic, albeit this could be due to small amount of training data used to train the model.