E1246 - Natural Language Understanding
Assignment2 : Language Models

Shivank Gupta(14638)
shivankgupta@iisc.ac.in

Abstract

In Assignment-1 I had developed a trigram based language model on **Brown** and **Gutenberg** corpus and then calculated perplexity on both the corpus. In this assignment i have implemented Neural Network based language model using LSTM on **Gutenberg** corpus. I have build two language model i.e. Character level and word level Language model and compared their results with the previous N-gram Language models.

1 Introduction

In this assignment we have only one corpus **Gutenberg** and we have to implement Neural Network based language model using LSTM over the entire corpus or on some subsets files of the corpus(whichever possible). We have mainly 3 tasks to perform :

1. Build best word-level Language model over the given corpus and calculate perplexity using the same split that we have used in our previous assignment.

2. Build best character-level Language model over the given corpus and calculate perplexity using the same split that we have used in our previous assignment.

3. Compare the above obtain models with Assignment 1 Language models and generate a sentence of 10 tokens from the best model that is obtained after comparing these models

2 Language Model

I have implemented the LSTM(Long Short Term Memory) Neural Network based Language Model which is a variant of RNN(Recurrent Neural Network). A recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed graph along a sequence.RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations. This allows it to exhibit dynamic temporal behavior for a time sequence. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them very efficient for a language model generation. But because of vanishing gradient problem they fails so to handle this LSTM was introduced which solve this problem. It has a property that helps in keeping short memory for long period thast's why it is named as LSTM. A RNN composed of LSTM units is often called an LSTM network. The cell is responsible for ”remembering” values over arbitrary time intervals; hence the word ”memory” in LSTM. Each of the three gates can be thought of as a ”conventional” artificial neuron, as in a multi-layer (or feedforward) neural network: that is, they compute an activation (using an activation function) of a weighted sum. Intuitively, they can be thought as regulators of the flow of values that goes through the connections of the LSTM; hence the denotation ”gate”. There are connections between these gates and the cell.

3 Preprocessing

Since text data contains lots of unnecessary things like punctuation, etc which are not use full for Language Modelling therefore preprocessing of text data is necessary. In each Model that i have removed all the punctuation and then trained my model.

4 Model Description

4.1 Character level LSTM Model

In this model i have used single layer of 128 LSTM units and then dense layer of softmax is used to predict the probability of next character. I have used 8 characters as input and predicted the next character probability and thus calculated the loss based on this probability.
Stride : 2
Input sequence : 8
Batch_size : 128
Learning Rate : 0.01

4.2 Word level LSTM Model

In this model i have used single layer of 128 LSTM units and then dense layer of softmax is used to predict the probability of next word. In thi am learning word embeddings along the training and dimension for each word embedding used in 20. Model takes sequence of 8 words as a input and predict the probability of next word and thus loss is calculated based on the predicted probabilities.
Stride : None
Input sequence : 8
Batch_size : 128
Learning Rate : 0.01

5 Results

Due to lack of resources i have trained my model on subset of whole corpus i.e for character level language model i have used the biggest file from the gutenberg corpus and for word level language model i have used only starting three files from the gutenburg corpus. Therefore the results obtained below are not good due to less training data and less number of epochs.

5.1 Character level Model

Perplexity- 189.36

5.2 Word level Model

Perplexity- 256.95

6 Sentence Generation

6.1 Character level Model

Few examples of sentence generated from character level model:

1. you mean to say that he had been a very good sort
2. and taking the dimensions of the two of them and mrs weston
3. that whenever i am sure i had not been very much to
4. to point and still i am sure i had not been very
5. away before she had been a very good sort of thing and

6.2 Word level Model

Few examples of sentence generated from word level model:

1. and i am afried it will be the proposalt with it.
2. pointing the truth, i am afraid of a letter, she s
3. ry well to be best in the whole england, and heard
4. as a second it is not mind in any word being for h
5. e no doubt and time of the brother of herself as i

7 Acuuracy/Measures

7.1 Model Comparison

Perpexility is used as the measure for this task.

7.2 Sentence Generation

Human Evaluation.

8 Conclusion

I have observed that if the number of epochs are less or the amount of training data is not sufficient then LSTM doesn't turn out to be a good model since it has huge parameters to learn therefore it requires large amount of training data in order to learn those parameters. But if we have large training data and and run our model for certain number of epochs then it turns out to be the best among all the models that we have design till now.

9 Github Link

https://github.com/shibu38/Nlu_Assignment_2