E1 246
Natural Language Understanding
Assignment 2 : Neural Language Modeling

Monish Kumar Keswani
Department of Computer Science and Automation
monishkumar@iisc.ac.in

Abstract

A Neural Network based language model using LSTM is designed on **Gutenberg**(D2) corpus. It is then compared with the N-grams language model built in the previous assignment under the same setting. Two Neural language models are built. One is the Token based LSTM model and another is the character based LSTM model. Both of these models are compared among themselves and the best model is used to generate a sentence of 10 tokens.

1 Introduction

Following three tasks are performed while designing the Language model

- The **Gutenberg**(D2) dataset is divided into train, test and dev set. The model is evaluated using the perplexity as the evaluation measure under the following scenario
  - **S2 :** Train: D2-Train, Test: D2-Test
- The token/word level based LSTM language model is built based on the setting above
- The character level based LSTM language model is built based on the setting above
- The sentence of 10 tokens is generated using the model designed.

2 Language Model

Recurrent neural networks (RNNs) allow representing arbitrarily sized sequential inputs in fixed-size vectors,while paying attention to the structured properties of the inputs.It can be seen in Figure-1 that RNN when unrolled is a very deep neural network in which same parameters are

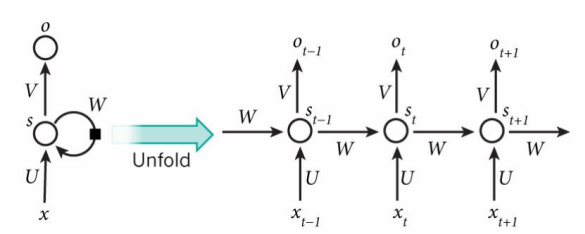

Figure 1: Unrolling of an RNN architecture

shared across many parts of the computation, and additional input is added at various layers.Training an RNN network is basically unrolling the RNN based on the input sequence and adding a loss to the node fo the unrolled graph, and then use the backward algorithm to compute the gradients with respect to that loss. This procedure is referred as backpropagation through time (BPTT). RNNs suffer from vanishing gradient problem which means it tends to forget about the past as it moves forward. The Long Short-Term Memory (LSTM) architecture was designed to solve the vanishing gradients problem, and is the first to introduce the gating mechanism. An LSTM has three of gates, to protect and control the cell state, an input gate, an output gate and a forget gate. The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit.

3 Preprocessing

The processing of the raw text is done to get the data which can be better understood by the model. The NLTK sentence tokenizer was used to split the raw data into corresponding sentences.The processed data was then divided into train, dev and test splits.

4 Model Description

The training was taking time and based on the resources available only 5 Gutenberg files were used for training. For both the models the hyperparameters were the Hidden Layer size, Number of layers, Input size, Time step length, Batch size

4.1 Character Model Description

For character level modeling the hyperparameters were tuned on dev set. The file of 1MB was trained to decide the parameters.

- Hidden Layer Size : 650
- Number of Layers : 2
- Input Size : Number of characters
- Time step length : 20
- Batch Size : 128

4.2 Word Model Description

For token/word level modelling the hyperparameters were set based on the same setting.

- Hidden Layer Size : 650
- Number of Layers : 2
- Input Size : 650
- Time step length : 20
- Batch Size : 128

5 Evaluation

Perplexity was used to evaluate the performance of the models. Each of the model was trained under same settings for them to be comparable.The following are the perplexity values based on the above models:

- The perplexity of Character Level LSTM model is 79.21
- The perplexity of Token level LSTM model is 121.21
- The perplexity of the N-grams model is 215.2

6 Sentence Generation

The sentences were obtained by taking random seeds. The sentences were found to be more natural if the number of epochs were increased. Better sentences can be obtained by training the model on GPU for more time. Some of the sentences that were generated in each of the model are listed below

6.1 Character LSTM Model

The examples generated using Character level LSTM modelling are given below

- According to him this is not corrsct
- She will be held thou shall not
- I was sent to be him of LORD him
- Mrs Weston thee will be him from that

6.2 Word LSTM Model

The examples generated using Word level LSTM modelling are given below

- I have been very deal to be him . Elton
- thee name , thee LORD , and I will give to him
- she a few of she her , be her she had done,
- I was not to be the same of the own man
- I heard great deal , be the LORD of thee.

7 Conclusion

The model training played a very important role in evaluation as well as sentences generation. If the model is trained for fewer epochs the perplexity was high and the high frequency words/characters were being generated like " the the the the ... ". It requires huge amount of resources to train.