NLU assignment-2(LSTM based models)

Sonu Dixit
sonudixit@iisc.ac.in

1 Introduction

LSTM(Long Short Term Memory networks) are a special kind of RNN, capable of learning long-term dependencies. A RNN(recurrent neural network) can be thought of as multiple copies of the same network, each passing a message to a successor. RNN's are weak at handling long distance dependencies. In the last assignment we built n gram based langugae models, in this we have to compare results of LSTM based models with n-gram models.

2 Task1: Character Level Model

Gutenberg corpus has been used. I have used 80-20 split for train and test data. That is from every book in the corpus 80-20 split was done.

At first data is converted to lower case and then I removed some special characters from the input data, using regular expression. The data contained 46 unique characters. I have used a sequence length of 50, that is every next character is being predicted according to last 50 characters context.

Every character is first encoded as a 128bit vector, a sequence of 50 such vectors are fed as input to 2 layered LSTM network. At the output of network, i get a probability distribution over such 128bit vectors(46 such vectors, since 46 unique characters are present). These vectors are then decoded back to its referring character.

PerPlexity Calculation: Output of LSTM is a probability distribution over all the characters. From this, probability of next character(in test data) is derived, hence perplexity is calculated. I got perplexity=6 in this model.

3 Task 2 : Word Level Model

Same train test split has been used as in Char based model. Here also, at first data is converted to lower case and then i removed some special characters. I replaced some training words with unknown (UNK) token. This step was not required in character level model, since there number of unique characters are very less. There is almost zero probability that i will see some new character. Word Vocabulary is very large, so probability of getting a unseen word during test time is very high as compared to unseen characters. Since characters are very limited in number. Among all the words which occur once in corpus, i have randomly selected 20 percent of them and replaced them with unknown token.

Here i have used sequence length of 20words. That is every word is being predicted based on a context of previous 20 words. Every word is encoded as a 128 bit vector before being fed to network. Here total unique words are 124888. As in the case of character level model, here also, in the output, network is giving a probability distribution over all the unique words in corpus.

PerPlexity Calculation: Output of LSTM is a probability distribution over all the unique words. From this, probability of next character(in test data) is derived, hence perplexity is calculated. I got perplexity=71 in this model.

4 Task 3 : Sentence Generation

I have selected a few words to start the sentence generation. Out of these words, one word is selected at random, and used as a starting word for sentence generation. After getting the probability distribution over all the ext characters, i am selecting top 5 charac-

ters, and out of them a character is randomly selected to be the next character. This randomness is necessary, otherwise it can form a repeating self loop kind of thing. If 'b' is most probable after 'a' and if 'a' is most probable after 'b', then if any of a/b is generated, then after the sequence 'abababab..' will be generated. Some of the generated sentences are: 1. a very dear beardedand with me down and made them him to me and present for the conclusionto me to the 2. serve hiscontent of judgment to set so with dance of his head of thecompanions of his journey in und 3. exis on hiscontest was found the world in the trees of life, and yet havebeen a remarkable and under

5 Observations

1. Perplexity decreased in these models as compared to n-gram models. Perplexity in character level model is better than that of word level model.

2. Sentence generated through word level model has better semantic meaning than character level model.

3. Time required for training LSTM based models are very high compared to that of n-gram model. Personal computers are not suitable for training these models.

6 Github Link

https://github.com/SonuDixit/nlu_second_assignment.git