A02(NLU)-LSTM BASED NEURAL NETWORK LANGUAGE MODEL

Chaikesh Chouragade
SR No. 14358
GITHUB LINK-https://github.com/chaikesh/LSTM-LANGUAGE-MODEL

ABSTRACT

In this assignment i have implemented neural network based language model .At last there is comparision between neural language model and classical(bigram and trigram) language models implemented in assignment-01 .LSTM recurrent neural network architecture is used to implement language model. Two variants namely word level and character level LSTM are implemented.

In word level LSTM i have evaluated the different model by varying the word embedding dimension, and the size of hidden state of LSTM cell.

1 DATASET

As adviced ,neural network needs to be trained on dataset preferably as large as possible as it avoids the problem of overfitting. due to computational limitations, we have restricted our data set to the following three files of gutenberg corpus: austen-emma.txt,austen-sense.txt and austen-sense.txt.

80-20 split is used for training and test data.

2 EVALUATION METRIC

For intrinsic evaluation perplexity measue is used to compare different language model on Test Set.Lower the perplexity implies better the language model.

For extrinsic evaluation random sentence is generated to compare different model,for this i have provided random seed sentence for initialization.

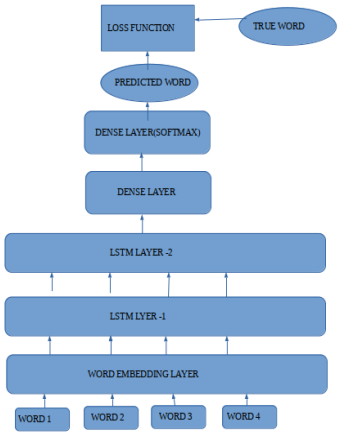
3 MODEL IMPLEMENTATION

3.1.WORD LEVEL LSTM MODEL

first pre-processing of the training data is done, where we convert all uppercase to lowercase, removed alphanumeric words and removed few special characters such as],/ etc as they do not hold much importance for our word level language models. Now we convert our pre-processed training data into fixed length of token of words depending on model design to train our network.

In this model we try to capture the contextof previous 50 words to predict the 51st word.Hence, we need our training data in the formatof 50 words as input and 51st word as output totrain the model. First, we fed these raw words asinput to embedding layer and we will be updating these embeddings corresponding to each word during training to get best embeddings for our task.

Next, we have a LSTM layer connected to another LSTM layer in encoder arrangement which outputs a final rep- resentation of our 50 words. This representation is further fed through two DNN and a softmax layer to predict the output probabilities of words present in the vocabulary. This predicted word is compared with true output and cross-entropy loss is calculated and further backpropagated to update the parameters.

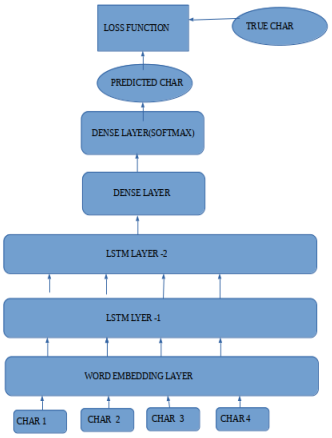


3.2.CHARACTER LEVEL LSTM MODEL

For building character level language model,we just

convert our training data into lowercase and convert into fixed length of token of characters depending on model design to train our network.

This model schematic is shown below ,it has now characters as input to character embedding layer. This model try to capture the context of previous 40 characters to predict the 41st character. Hence, we need our training data in the format of 40 characters as input and 41st character as output to train the model. First, we fed these raw characters as input to embedding layer and we will be updating these embeddings corresponding to each character during training to get best embeddings for our task.



4 MODELS WITH DIFFERENT HYPERPARAMETER

There are total of four variation of word level model,with different hyperparameter.in all the word level model i have rumn the model for total of 80 epochs.

M1
embedding dimension-300
hidden state dimension-100

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 50, 300)	2483400
lstm_1 (LSTM)	(None, 50, 100)	160400
lstm_2 (LSTM)	(None, 100)	80400
dense_1 (Dense)	(None, 100)	10100
dense_2 (Dense)	(None, 8278)	836078

Total params: 3,570,378
Trainable params: 3,570,378
Non-trainable params: 0

M2
embedding dimension-500
hidden state dimension-100

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 50, 500)	4139000
lstm_1 (LSTM)	(None, 50, 100)	240400
lstm_2 (LSTM)	(None, 100)	80400
dense_1 (Dense)	(None, 100)	10100
dense_2 (Dense)	(None, 8278)	836078

Total params: 5,305,978
Trainable params: 5,305,978
Non-trainable params: 0

M3
embedding dimension-100
hidden state dimension-100

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 50, 100)	827800
lstm_1 (LSTM)	(None, 50, 100)	80400
lstm_2 (LSTM)	(None, 100)	80400
dense_1 (Dense)	(None, 100)	10100
dense_2 (Dense)	(None, 8278)	836078

Total params: 1,834,778
Trainable params: 1,834,778
Non-trainable params: 0

M4
embedding dimension-300
hidden state dimension-50

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 50, 300)	2483400
lstm_1 (LSTM)	(None, 50, 50)	70200
lstm_2 (LSTM)	(None, 50)	20200
dense_1 (Dense)	(None, 50)	2550
dense_2 (Dense)	(None, 8278)	422178

Total params: 2,998,528
Trainable params: 2,998,528
Non-trainable params: 0

there is a single character level model

M5
embedding dimension-300
hidden state dimension-50

Layer (type)	Output Shape	Param #
embedding_2 (Embedding)	(None, 40, 50)	2750
lstm_3 (LSTM)	(None, 40, 100)	60400
lstm_4 (LSTM)	(None, 100)	80400
dense_3 (Dense)	(None, 100)	10100
dense_4 (Dense)	(None, 55)	5555

Total params: 159,205
Trainable params: 159,205
Non-trainable params: 0

5 RESULTS

5.1 PERPLEXITY

MODEL	PERPLEXITY	CROSS ENTROPY LOSS
M1(WORD LEVEL)	78	1.30
M2(WORD LEVEL)	87	2.29
M3(WORD LEVEL)	85	2.73
M4(WORD LEVEL)	89	2.34
M5(CHARACTER LEVEL)	4.56	1.15
ADD-K	261	----
KNESER NEY	183	----

5.2 RANDOM SENTENCES

M1 the two last lines and there is no reason why you should not write it into your book oh but those two lines are the best of all granted for private enjoyment and for private enjoyment keep them they are not at all the less written you know because you divide.....her a little host lady i have not one who is as often bred.

M2

mourning in secret over obstacles which must divide her for ever from the object of her love and that marianne was internally dwelling on the perfections of a man of whose whole heart she felt thoroughly possessed and whom she expected to see in every carriage which drove near their house .....and of temptation her own imagination and impetuous him that she knew her impatience.

M3

am as confident of seeing frank here before the middle of january as i am of being here myself but your good friend there nodding towards the upper end of the table has so few vagaries herself and has been so little used to them at hartfield that she cannot calculate.....yourself at randalls though i am sure i am sure you will be very much.

M4

t class us together harriet my playing is no more like her s than a lamp is like sunshine oh dear i think you play the best of the two i think you play quite as well as she does i am sure i had much rather hear you every body.....but i should not be tempted for she had often called at his son.

M5

all the tediousness of the many years of.....the same old from the conscious and seeing the door, and they were serious.

6 OBSERVATIONS

IMPLEMENTATION-

1.Due to computational limitation i restricted model to run for only 80 epochs,incresing epochs would have resulted in better model.One importatnt point to note is that as we have trained our word level model on around 1.15 lacs observations and number of parameters were around 40 lacs,which indicate that our model is overfitting.

2.But in case of chracter level model we have trained our word level model on around 6.24 lacs observations and number of parameters were around 1.59 lacs.hence model is good ,no overfitting is possible.

PERPLEXITY

1.Word level neural network model performs better than classical bigram ,trigram level model in terms of perplexity.As mentioned in implementation issue we found our best possible word level language model.

2.varying word embedding dimension has significant effect on accuracy. Among 100,300,500 dimension 300 was found to give optimal result.

3.varying hidden state size also has an effect on accuracy of model,among 50,100 dimension 100 dimension gave better results than 50 dimension.

4.character level language model was best in terms of perplexity,which is expected as we did not overfit the model ,which was the case in word level language model.

5.Overall the LSTM bsaed neural network ,perform better than classical language model.