Natural Language Understanding with Distributed Representations - Assignment 3

Urjitkumar Patel
up276@nyu.edu

In this assignment, we are experimenting with two most popular RNN Language model, one is LSTM and second is GRU.

1 Bag of Words

1.1 shortcomings of a bag of words model

The main disadvantage of Bag of words model is we don't consider the order of words during the classification of document. Hence If we consider Bag of words model for sentiment analysis, Below two sentences,

1. This is a good movie, its not a bad movie

2. This is a bad movie, its not a good movie

Both sentences represent totally opposite meaning but with bag of words model, model will not be able to differentiate between these two sentence as we don't use the information coming from previous word in Bag of words model. With LSTM model, if we take the example of the same case, model will learn that "not bad" actually means "good", which will help differentiate between both sentences.

1.2 LSTM to Bag of Words model

In LSTM, we mainly have three gates 1. Input, 2. output and 3. forget. All these gates are basically use sigmoid to decide what information to pass based on their specific functions. Forget gate decides what information to forget from the information coming from previous step. This gate multiplies value between 0 to 1 to the values coming from previous cell state. **If we make all multiplier values 0 in forget gate, we will not get anything from previous cell state**. In short, we will just use current word for the output. Which is if we think is the same case as Bag of words model.

2 RNN Language Model

2.1 Train a language model

With the help of code given, I was able to train a LSTM language model. Default LSTM parameters worked well for me. I used small model configuration for training my model. Here below are the configuration. (Results are attached with next section)

parameter	Default LSTM
no of rnn layers	2
max grad norm	5
No. of steps	20
Batch size	20
hidden size	200
lr decay	0.5
L2 Reg. LAMBDA	0
learning rate	1
no.of epochs	13 (Stopping Criteria)
Drop Out Keep Prob	-

Stopping Criteria : I tried to use early stopping (with patience of 2) as stopping criteria but sometimes I stopped too early (before 5 epochs), hence I just used number of epochs 13 as stopping criteria.

2.2 Which gate is the most important for this task?

I observed that, **removing output gate from LSTM impacted the performance the most** compare to forget and input gates. It makes sense because output gate is the one which directly contributes to make the prediction for the current state. Hence, removing the output gate would degrade the performance with high margin. On the other hand, even if we don't forget anything from previous information, model would not be able to use information which it observed very long ago, but still it will not impact the performance very badly because our prediction is mainly depends on current or relatively new information. Without Input gate, We will pass all the information coming from new state, which should not hurt the model as much as it hurts by removing other two gates (forget or output). I observed the same with my experiments. Below are the best results achieved for all LSTM Configuration.

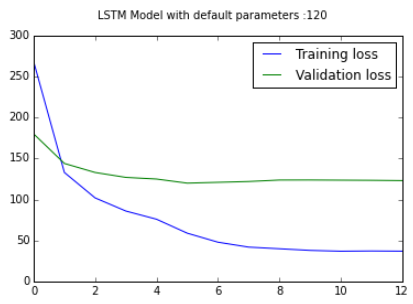

Figure 1: LSTM Default

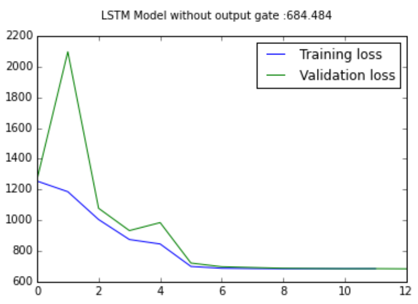

Figure 2: LSTM without output gate

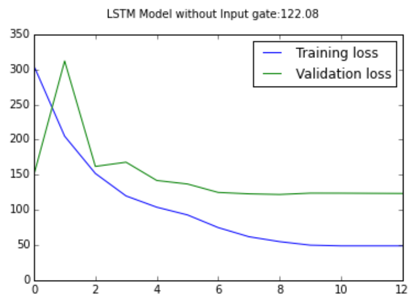

Figure 3: LSTM without input gate

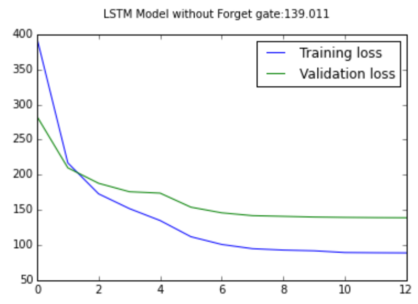

Figure 4: LSTM without forget gate

Network	Train Accuracy	Val Accuracy	Test Accuracy
LSTM Default	3̃7	1̃20	1̃17.551
LSTM Without Input gate	4̃9	1̃22	1̃17.875
LSTM Without Output gate	6̃83	6̃85	6̃39.169
LSTM Without Forget gate	8̃9	1̃39	1̃31.673

2.3 Gated Recurrent Unit

I noticed that with the same parameters as LSTM, GRU model was not able to converge well. I ended up around 600 perplexity which is very bad. I decreased the learning rate and increased the hidden layers size. With that I was able to reduce the perplexity to around 150.

I used the max number of epochs as stopping criteria. Here below are the configuration I used for GRU and comparison with LSTM.

parameter	Baseline Uni-gram
no of rnn layers	2
max grad norm	5
No. of steps	20
Batch size	20
hidden size	**500**
lr decay	0.5
L2 Reg. LAMBDA	0
learning rate	**0.1**
no.of epochs	20
Drop Out Keep Prob	-

Comparison Of GRU with LSTM

Network	Train Accuracy	Val Accuracy	Test Accuracy
LSTM	3̃7	1̃20	1̃17.875
GRU	1̃16	1̃54	1̃49.906

Even with my optimal configuration of GRU, LSTM performed better on test set. Also LSTM converges faster than GRU. But, I observe very less overfitting with GRU configuration. In terms of training time, both models took around 1000 seconds.

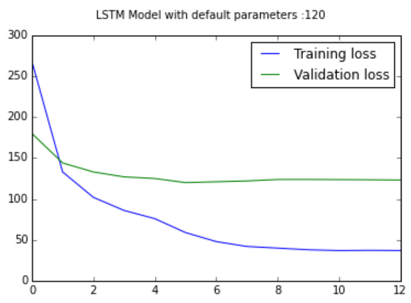

Figure 5: LSTM with default parameters

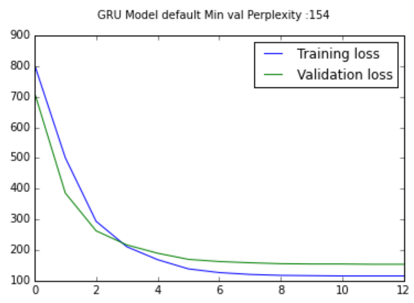

Figure 6: GRU with tuning

2.4 Setup evaluation

Please find attached evaluation script which uses the trained model and assigns the score out based on the rubric given in the assignment. Please note that I am assigning the score out of 9 because we don't have the word embedding for ",". Here below are the results.

Network	Evaluation score out of 9
GRU	9
LSTM	9

2.5 t-SNE Visualization

t-SNE representation with all words was very messy and not readable. Hence I just visualized 300 words. I have attached the images separately too with the submission as images below are not clear. Here is how it look like.

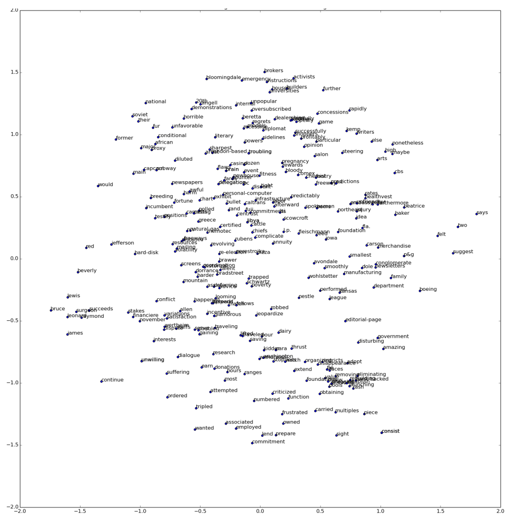
Figure 7: t-SNE representation - LSTM Model

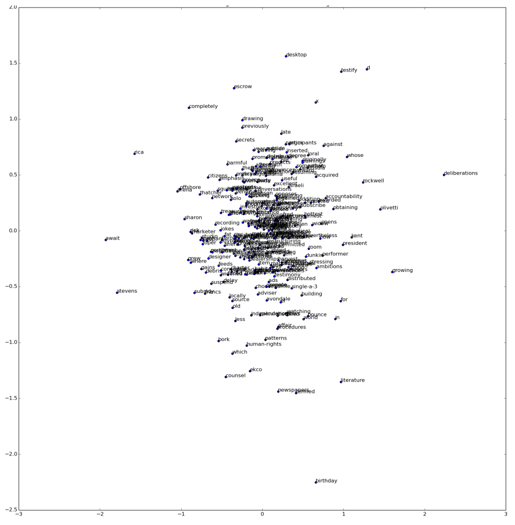
Figure 8: t-SNE representation - GRU Model