Piradon (Tien) Liengtiraphan
Final
Instructor: Dr. Pablo Rivas

CMPT 496L-111: Deep Learning w Tensor Flow
Due: 2017-12-9

Final

Exercise 1a.

Download this modified version (attached). After decompressing the file you will notice that it has a data directory. Such directory contains three main datasets; each is divided into train, validation, and test sets. The three datasets are 1) the Penn Tree Bank (PTB) with a great collection of general purpose sentences, 2) the Trump dataset that includes your professor's personal collection of presidential tweets, and 3) the Mark dataset which is the earliest account of Jesus of Nazareth according to the author Mark. These three datasets vary in size and vocabulary length, and to set the experiments up, you will have to follow this steps:

1. *edit the file reader.py around lines 71-73 to reflect your dataset choice*
2. *execute the file lanmod.py and terminate its execution after it displays 'Epoch 1 : Learning rate: 1.000' (if it doesn't crash by itself)*
3. *open a file generated automatically with the extracted dictionary, entitled word_to_id.csv and look at how many entries does the file has (how many rows) and write the number down somewhere, e.g., PTB has 10,000 rows*
4. *edit the file lanmod.py around line 38 where it says 'vocab_size = ' and set it equal to the number you obtained from the previous step*
5. *pat yourself in the back, you did well, now go get some coffee because the fun is about to begin*

Response.

Below are the enumerated parameters that will be the basis for my Final in CMPT 496L-11: Deep Learning w Tensor Flow.

1. **mark.txt** and later after experimentation with pre-processing **pre.mark.txt**
2. The line after printing "Epoch #: Learning Rate..." was modified with `exit()` to force the program to exit immediately after the above line was printed. Removed for following experiment.
3. Rows generated after 1 Epoch: 2722 and 1798 after pre-processing.
4. Made espresso to maintain consciousness.

Exercise 1b.

Except for the batch size, which should remain "batch_size = 1" in the last experiments (see the remarks section below), the flag "is_training=1" that also should remain the same, and the "vocab_size = " that will be updated before hand, you will vary any and all of the parameters as you see fit, according to your understanding and study of the lab you read. Run as many experiments as you can, there is no lower or *upper limit. The goal for you is to find the best set of parameters that yield the lowest perplexity in the validation and test sets. Record your experiment results.*

Response.

Parameter Combination (init_scale, learning_rate, max_grand_norm, num_layers, num_steps, hidden_size, max_epoch, max_max_epoch, keep_prob, decay)

Parameter Combination (See Above)	Valid Perplexity	Test Perplexity
(0.05, 1.0, 5, 5, 50, 256, 10, 13, 1, 0.5)	2192.244	1172.050
(0.05, 1.0, 5, 5, 50, 256, 20, 13, 1, 0.5)	1315.692	777.220
(0.05, 1.0, 5, 5, 50, 256, 4, 13, 1, 0.5)	924.889	575.621
(0.05, 1.0, 5, 5, 50, 512, 4, 13, 1, 0.5)	4477.708	1898.221
(0.1, 0.1, 5, 5, 20, 200, 4, 13, 1, 0.5)	312.598	359.779
(0.1, 0.1, 5, 5, 20, 200, 5, 13, 1, 0.25)	195.353	204.870
(0.1, 0.5, 5, 5, 20, 200, 5, 13, 1, 0.25)	571.569	299.741
(0.1, 1.0, 10, 5, 50, 256, 4, 13, 1, 0.5)	779.359	408.026
(0.1, 1.0, 5, 10, 20, 200, 4, 13, 1, 0.5)	270.019	303.856
(0.1, 1.0, 5, 4, 20, 200, 5, 13, 1, 0.25)	549.386	375.960
(0.1, 1.0, 5, 5, 20, 200, 4, 13, 1, 0.1)	278.853	256.845
(0.1, 1.0, 5, 5, 20, 200, 4, 13, 1, 0.25)	396.598	280.185
(0.1, 1.0, 5, 5, 20, 200, 4, 13, 1, 0.5)	727.568	433.751
(0.1, 1.0, 5, 5, 20, 200, 5, 13, 0.5, 0.25)	259.223	307.693
(0.1, 1.0, 5, 5, 20, 200, 5, 13, 1, 0.15)	421.270	279.855
(0.1, 1.0, 5, 5, 20, 200, 5, 13, 1, 0.25)	291.187	223.111
(0.1, 1.0, 5, 5, 20, 200, 5, 14, 1, 0.15)	274.428	310.168
(0.1, 1.0, 5, 5, 20, 200, 5, 15, 1, 0.25)	281.364	311.524
(0.1, 1.0, 5, 5, 20, 200, 6, 13, 1, 0.25)	285.093	232.291
(0.1, 1.0, 5, 5, 20, 205, 6, 14, 1, 0.25)	420.625	289.442
(0.1, 1.0, 5, 5, 21, 200, 5, 13, 1, 0.25)	451.812	236.453

(0.1, 1.0, 5, 5, 25, 200, 5, 13, 1, 0.25)	286.154	314.680
(0.1, 1.0, 5, 5, 50, 200, 4, 13, 1, 0.5)	348.596	319.131
(0.1, 1.0, 5, 5, 50, 256, 4, 13, 1, 0.5)	483.033	379.269
(0.1, 1.0, 5, 5, 50, 256, 4, 20, 1, 0.5)	903.603	417.186
(0.1, 1.0, 5, 6, 20, 200, 6, 13, 1, 0.25)	298.443	359.594
(0.1, 1.0, 6, 5, 20, 200, 5, 13, 1, 0.25)	613.280	341.681
(0.1, 1.0, 6, 6, 20, 200, 5, 13, 1, 0.25)	350.641	332.720
(0.1, 1.5, 5, 5, 20, 200, 5, 13, 1, 0.25)	293.364	349.758
(0.2, 1.0, 5, 5, 20, 200, 5, 13, 1, 0.25)	285.408	293.872
(1.0, 1.0, 5, 5, 20, 200, 6, 13, 1, 0.25)	96.125	78.181
(1.0, 1.0, 5, 5, 50, 256, 4, 13, 1, 0.5)	2016345.683	2382912.956
(0.1, 1.0, 5, 5, 20, 200, 5, 13, 1, 0.25)	115.769	74.413
(0.1, 1.0, 5, 5, 20, 200, 5, 13, 1, 0.25)	143.205	99.802
(0.1, 1.0, 5, 5, 20, 200, 5, 13, 1, 0.25)	146.245	84.101

Exercise 1c.

Carefully analyze your results, and answer the following questions:

- *What was your strategy in the search of the best set parameters? Explain.*
- *Based on this experience, which parameters you think are the most influential, crucial, or important to yield a good result? Explain.*
- *Based on this experience, which parameters you think are the least influential, crucial, or important to yield a good result? Explain.*
- *Observing the experiment that gave you the best results, discuss:*
  - *What happens to the perplexity in training, validation, and testing sets? Why do you think that is? Support your answer with a plot of the training and validation perplexity.*
  - *What are your thoughts on the quality of the sentences that it produces? Provide sample sentences to support your answer.*
  - *Is the quality of the sentences congruent with the perplexity on the validation set? Explain.*

Exercise 1ca.

What was your strategy in the search of the best set parameters? Explain.

Response.

The first step in ascertaining the best set of parameters is to run the language modeling neural network "as-is" meaning there are no change in parameters from the default given to us at the beginning of the assignment. This is used to establish a baseline from which we will measure our success. Any perplexity returned that is lower than the base perplexity will be considered an improvement, while any deviation higher than the base perplexity will considered as a worse parameter set.

The next step was to understand exactly what each of the different parameters control, by understanding this one could gain a better sense as to what truly mattered within this particular situation. Reading the RNN lab entitled "Language Modeling with RNNs/LSTMs" as suggested by Professor Rivas, is definitely a good start in gaining a cursory understanding of each parameter. A quick summary of the most pertinent part of the lab is located below

- In this network, the number of LSTM cells are 2. To give the model more expressive power, we can add multiple layers of LSTMs to process the data. The output of the first layer will become the input of the second and so on.
- The recurrence steps is 20, that is, when our RNN is "Unfolded", the recurrence step is 20.
- Each LSTM has 200 hidden units which is equivalent to the dimensionality of the embedding words and output.

The next important step in this strategy is to Google everything you don't understand. The benefit of this is that you are going to be more knowledgeable going in, rather than flailing around like a fish out of water. Once you understand the parameter and how they relate to the recurrent neural network we can move onto changing the parameters that we found are the most pertinent. Then measuring the resulting valid and test perplexity against the base case and/or the current best parameter set and continue to massage the parameter set to achieve the ideal setting.

After a certain amount of testing and discussing with fellow students, it was suggested that pre-processing the data might lead to better results. By this suggestions I wrote up the file `data_preprocessor.py` in order to achieve this. It does this by clearly separating punctuation from words and standardizing all the words, all lowercase, in an attempt to reduce the amount of noise within the data. Once this was accomplished and ran against the best parameter settings that I had previously noted down, I achieved my lowest perplexity of **74.413**.

Exercise 1cb.

Based on this experience, which parameters do you think are the most influential, crucial, or important to yield a good result? Explain.

Response.

From my reading and experiments modifying the data set, I came to the conclusion that the paramount parameter appears to be learning rate. Where modifying learning rate beyond the ideal rate of **0.1** leads to wildly varying perplexity rates. However the biggest degradation in quality, when learning rate is modified is the quality of the sentences. In either direction, the sentences seem to significantly drop in quality, ranging from being comprised entirely of only small sets or words or consisting of one character in its entirety.

Another important factor that leads to large changes in perplexity when modified beyond ideal point is the number of hidden states. This makes sense when you consider the fact that the number of hidden states should be equivalent to the dimensionality of the embedding words and output, and when this is not the case, the recurrent neural network will have a harder time rectifying the difference.

Decay rate seems to also have a big effect on the results. Given that another important contributing factor is learning rate, decay controls the amount by which the learning rate is decreased after the max learning rate epoch is passed. Meaning that all the following epochs will be trained at a lower learning rate than those that came before them. This is important with smaller data sets, as it helps prevent overfitting.

Exercise 1cc.

Based on this experience, which parameters do you think are the least influential, crucial, or important to yield a good result? Explain.

Response.

Surprisingly I discovered through different experiments the number of Epochs makes very little difference in lowering the overall number of the testing perplexity. While increasing the number of Epochs, did for a while decrease the perplexity, eventually this decrease flat-lined. However the overall problem was not in the perplexity, the problem in fact lies in the degradation in sentence quality. An overall trend for this was the fact that sentences were comprised entirely of ”choice words”. It would seem that by increase the amount of training for a data set that does not scale to size, the neural network over-trains on the training data, and is overfitted as a result.

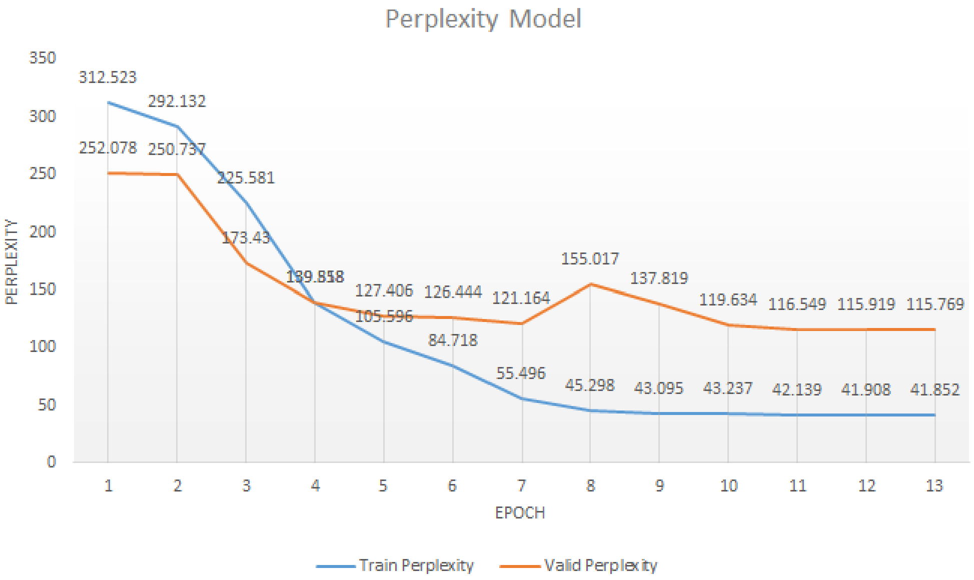
There is a correlation between max epoch and decay and learning rate. From general observation these should be proportional to each other, in that if one is increased then another should be decreased to compensate. Given this relationship, with decay and learning rate being identified as heavier influences, max epoch can take a back seat and have limited to minimal effect on the resulting perplexity, assuming the other two factors are modified properly.

Exercise 1cda.

Observing the experiment that game you the best results, discuss:

- *What happens to the perplexity in training, validation, and testing sets? Why do you think that is? Support your answer with a plot of the training and validation perplexity.*

Response.



As you can see from the graphic above the overall trend with regards to perplexity in the above data sets (train and valid) there is a general downwards slope for perplexity, which mean that the recurrent neural network, is getting better at predicting and understanding the input. This trend seems to exist regardless of the parameter sets, meaning that despite, bad parameters the recurrent neural network is still able to understand the input coming in to produce, somewhat valid, output. This trend continues until an eventual flat line near the end of training, with thirteen Epochs.

I suspect that this flat line will result in worse data over time, should the number of Epochs be increased beyond the scope of the training data set. It is normal to assume that at the beginning of training the perplexity of both the train and valid data sets would be relatively high when compared to the overall result of the recurrent neural network after the training has completed. This is due to the fact that at the beginning of the training the neural network is unfamiliar with the input and unable to produce valid output, resulting in the high amount of perplexity. As training progresses further we can see the steady decrease of perplexity as the neural network familiarizes itself with more of the input data set, and uses that familiarity to build its output, leading to an improvement in overall output. The flat line most likely begins when the network has reached the extent to which the input can provide significant improvement, from here further training would lead to overfitting. Here is where the max epoch would ideally be reached to prevent overfitting, and decay rate would allow the neural network to simply generate output.

Exercise 1cdb.

What are your thoughts on the quality of the sentences that it produces? Provide sample sentences to support your answer.

Response.

Sample Sentences

- the the right . and the the priests who the lord who the lord of the right , the lord
- the lord will be me . and they went out and went to the tomb , and they will not
- the son of man is . and he had not with him , and they were believe . and
- the son of god , and you will be believe ? but they will not believe him . and
- the son of god , and you will be me . but he said to them , why do you

Repetitive is the first word that comes to mind concerning the output. The sentences being produced at this point seem to follow a trend/pattern of starting with ”The” and have significantly less amounts of gibberish when compared to other parameter sets. This is most likely due to the fact that with the ideal parameter the neural network is able to process and minimize the input effectively enough, thereby maximizing the usage of neural network resources. The batch size which has remained static for this particular experiment is relatively small, therefore it could be reducing the chance of diluting the function's effect while processing the data.

Another possibility is that given the relatively small size of this data set, by keeping the number of hidden states small, we are allowing the neurons to specialize more at identifying specific patterns and/or trends within certain data inputs. By allowing this specialization, without further dilution with the addition of unnecessary neurons, we increase the certainty of each type of prediction, for each individual neuron, leading to an overall decrease in perplexity.

Exercise 1cdc.

Is the quality of the sentences congruent with the perplexity on the validation set? Explain.

Response.

Unlike other parameter sets, which produced lower perplexities but sentences that were obviously over trained on certain key words. The ideal parameter set that my recurrent neural network is currently operating on does seem to be congruent with the perplexity of the validation send and the quality of the sentence produced. Looking at the sentences we can see the general trend of the sentences becoming much more coherent, in the sense that specific word types are being ordered at appropriate intervals and locations. This is not to say that the sentences are "good". By a stretch one could infer meaning from the generated sentences, but they are by no means the words of a poet.

This corresponds well to the validation perplexity, in that the the number confirms that while the neural network may be improving on its understanding of the input, the overall output generated is still not inherently comprehensible. The best validation perplexity produced at the thirteenth Epoch of the recurrent neural network operating on the ideal parameter set is **115.769**, indicating that there is still a relatively high percent chance of the model having an error in prediction; which given the sample sentences produces, confirms the congruency of the associated values.
