Project 2 - Recurrent Networks and Sentiment Classification-Report

Parameters:

BATCH_SIZE: (128)

MAX_WORDS_IN_REVIEW: (400)

EMBEDDING_SIZE: (50)

Preprocessing:

1. Convert the input string into lower case, using string.lower() function, the GloVe embeddings are all in lowercase.
2. Remove all the punctuations and non-words word. Import re library, using re.compile('(\w+)') extract the pattern only contains a-z in lower case. Then using obj.findall() to find all the words only contains a-z digits, returned in a list.
3. Remove all the stop words that were provided, and only left the word has length longer than 2 digits, because stop words mostly don't have emotion trend, can misjudge the results.
4. Using random.shuffle method disorganize the word inside the list. Because we only need part of the words, so we need to make sure that possibility of each word been chosen is equal.
5. Append stops into the review which has less than MAX_WORDS_IN_REVIEW number of words.
6. Join all the words into a string, because finally the runner.py method need a string as input.

define_graph:

1. Define input_data, labels, dropout_keep_prob using tf.placeholder() as usual.
2. Define weights, biases using tf.Variable()
3. Define a lost cell using tf.contrib.rnn.BasicLSTMCell()
4. Using tf.contrib.rnn.DropoutWrapper() to make the last cell become a dropout cell.
5. Using tf.nn.dynamic_rnn() to run the cell and the input_data, returns outputs, final_state.
6. Multiply the final_state[1], weights and add the biases, and then using the tf.nn.sigmoid() activation function to calculate the results.
7. Using tf.nn.softmax_cross_entropy_with_logits() to calculate the difference of results and labels and using the tf.reduce_mean function to calculate the loss.
8. Using the tf.train.AdamOptimizer().minimize() to minimize the loss.
9. Using tf.equal() to return a list which contains the results that has true or false if the result and labels equal or not, and then convert the accuracy into decimal fraction using tf.cast() and tf.reduce_mean.