CSEE 5590 - Special Topics  
 Deep Learning – Lab 3

**Author:**   
Kalyan Kilaru  
Class Id: 22

**Configuration:**  
IDE: pycharm  
python: version 3

**Introduction:** By using the new set of Data we need to implement the text classifications with CNN, LSTM and RNN models. Apart from that we also need to compare the results of each model with the others.  
  
**Objective:** The main objective of this assignment is to study how each model works and how can we implement them. Also to know how the accuracy or loss is effected in each model that takes the same data.

**Approaches:** To implement text classification with CNN, LSTM and RNN models I have considered the mnist Data set for the 3 models mentioned above. The code is modified slightly from that of the given tutorials in order to create the accuracy or loss and also the graph in the Tensor Board. I used the same data set in order to check how the accuracy is affected when the parameters are changed. I observed the changes regarding the loss that occurs whenever the Hyper Parameters are modified.

**Workflow:**

* To get the same data set for all the models.
* To write the code that generates the Accuracy/Loss and the graph in the Tensor Board.
* To keep the parameters same for all the models and run them.
* Then we will compare the result for each model.
* To modify the Hyper Parameters and check how it effects the Accuracy for all the models.

**Datasets:** Same Data set is used for all the three models in order to compare the results effectively. The Data set used is “MNIST”.

**Parameters:**

* The Learning Rate is taken as 0.001
* The Training steps taken is 1000
* 128 is the Batch Size taken
* 200 is the Display step taken

**Evaluation:**I have changed the code that is given in class 4 and class 5 slightly for the CNN model.

**A screenshot of a cell phone

Description generated with very high confidence**

The Below mentioned is the Code for the RNN Model

def RNN(x, weights, biases):

x = tf.unstack(x, timesteps, 1)

#LSTM cell value

lstm\_cell = rnn.MultiRNNNCell([rnn.BasicLSSTMCell(num\_hidden), rnn.BasicLSTMCell(num\_hidden)])

#States, outputs

outputs, states = rnn.static\_rnn(lstm\_cell, x, dtype=tf. float32)

return tf.matmul(outputs[-1], weights['out']) + biases['out']

As both the RNN and LSTM are of similar Models some small changes are made for the RNN model in order to get the graph of LSTM.  
  
def LSTM(x,weights,biases):

x = tf.unstaack(x, timesteps, 1)

lstm\_cell = rnn.BasicLSTMCell(num\_hidden, forget\_bias=1.0)

outputs, states = rnn.static\_rnn(

lstm\_cell, x, dtype=tf.float32)

return tf.matmul(outputs[-1],

weights['out']) + biases['out']

**Output(Graphs):**Here is the output graph for the CNN model in the Tensor Board.

**A close up of a map

Description generated with very high confidence**

The graph or the output for the RNN Model.

A close up of a map

Description generated with very high confidence

The graph or the output for the LSTM Model which is quite similar to that of the RNN.

A close up of a map

Description generated with very high confidence

The below is the Step\_Loss\_Accuracy for the CNN Model

Step 1, Loss=69134.38228, Accuracy=0.190

Step 200, Loss=1349.51222, Accuracy=0.9245

Step 400, Loss=714.62136, Accuracy=0.9353

Step 600, Loss=529.60330, Accuracy=0.9253

Step 800, Loss=425.59349, Accuracy=0.9153

Step 1000, Loss=439.52534, Accuracy=0.9861

\

Optimization Done.

Testing Accuracy: 0.970668575

The below is the Step\_Loss\_Accuracy for the RNN Model

Step 1, Loss=2.9316, Accuracy= 0.128

Step 200, Loss=2.2572, Accuracy= 0.159

Step 400, Loss=1.8823, Accuracy= 0.417

Step 600, Loss=1.9841, Accuracy= 0.363

Step 800, Loss=1.7148, Accuracy= 0.441

Step 1000, Loss=1.44889, Accuracy= 0.514

/Optimization Finished!

Testing Accuracy: 0.536867

The below is the Step\_Loss\_Accuracy for the LSTM Model

Step 1, Loss=2.4718,Accuracy=0.064

Step 200, Loss=2.1881,Accuracy=0.409

Step 400, Loss=1.9338,Accuracy=0.567

Step 600, Loss=1.8280,Accuracy=0.467

Step 800, Loss=1.7814,Accuracy=0.564

Step 1000, Loss=1.3934,Accuracy=0.514

/Optimization Finished!

Testing Accuracy: 0.47125452

**Conclusion:**From the above results we can observe that the accuracy and the loss is high for the CNN Model when compared to that of the RNN and LSTM Models. When the Training steps has been increased for the RNN and LSTM the accuracy Increased and the loss has been decreased. In this case the LSTM performs best over RNN Model.

**References:  
https://www.tensorflow.org/programmers\_guide/  
https://stackoverflow.com/questions/44167134/**