**Introduction**

Implementing the Text Classification with CNN, RNN and LSTM model with new dataset and displaying the graphs in the tensor board. At the end compare results and decide which is the better model.

**Objectives**

1. To perform the text classification with Convolutional Neural Networks model
2. To perform the text classification with Recurrent Neural Networks model
3. To perform the text classification with Long Short-Term Memory
4. Then compare the results of three models and decide the best model

**Approaches/Methods**

In the beginning, I choose the MNIST as the input file. Here we will develop to perform the CNN, RNN and LSTM to perform the text evaluation and display the graphs in tensor board with filewriter and summaries. Then compare the model by their accuracies

**Workflow**

1-> Import the dataset

2-> Change some parameters in the code

3-> Find the accuracy and loss for each model

4-> Plot the graph in TensorFlow for each model

5-> Use same parameters in all the models and evaluate all the models

6-> The compare all the models and find the best model

**Dataset**

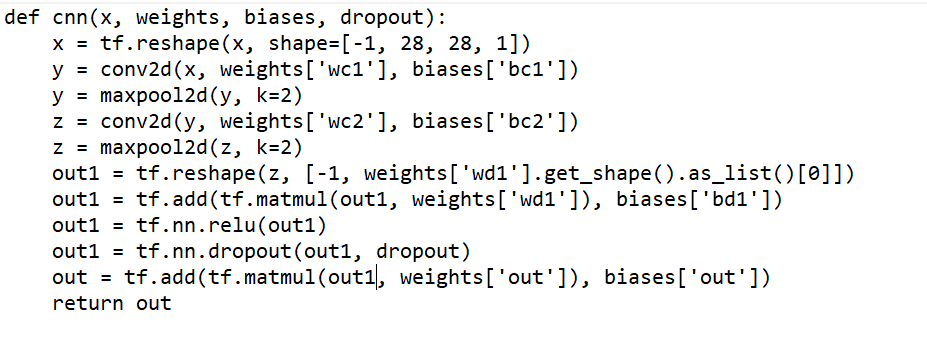
MNIST Dataset

**Parameters**

* Learning Rate
* Training Steps
* Batch Size
* Display Step

**Evaluation**

Evaluation of Convolutional Neural Networks Model:

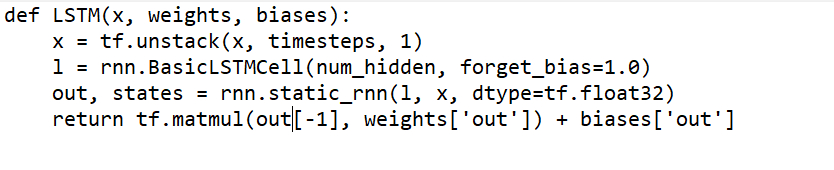


Evaluation of Recurrent Neural Network Model:

A screenshot of a cell phone

Description generated with high confidence

Evaluation of Long Short Term Memory: (LSTM units are building units of layers of a RNN)



The display step here is 200. So for every 200 steps it will display the output and learning rate is taken as 0.001 and training steps as 1000. The batch size is 200.

Graphs in Tensor Board for CNN, RNN and LSTM respectively:A close up of text on a white background

Description generated with high confidence

A screenshot of a cell phone

Description generated with high confidence

A screenshot of a cell phone

Description generated with high confidence

Output:

**CNN**

A screenshot of a cell phone

Description generated with very high confidence

**RNN**

A screenshot of a cell phone

Description generated with very high confidence

# for 10000 training steps

A screenshot of a cell phone

Description generated with high confidence

**LSTM**

A screenshot of a cell phone

Description generated with high confidence

# for 10000 training steps

A screenshot of a cell phone

Description generated with very high confidence

**Conclusion**

We conclude that Convolutional Neural Network Model is best among the three models depending on the accuracy and loss outputs of the models. If we increase more number of training steps, the accuracy increases, and loss increases but the time complexity also increases. LSTM stands in the second position and RNN in third position.