Deep Learning

LAB-3

Namrata Dutta |16247052

INTRODUCTION

This task is for the execution of Text Classification in various types of models like CNN, RNN and LSTM. We need to use a new data and then illustrating the same data for accuracy and loss. The next task is to analyze all the three models with the same data and reviewing them together to see which the best fit among all is.

The process of Text Classification is a function in Natural Language processing where the text or phrases given by a user is represented with sparse lexical features using linear or kernel methods. The examples for the same in neural systems are CNN, RNN and LSTM (long short term memory systems). For our task, CNN, RNN and LSTM models have been implemented on the same dataset to get and review the accuracy of the model.

CNN- Convolutional Neural Network Systems

Text classification in case of CNN comes with word embedding. Convolutional neural networks are very effective at document classification, because they are able to pick out the important features like tokens or sequences of tokens from the input sequences. CNN is also computationally very efficient. In CNN, a series of convolution + pooling operations, followed by a number of fully connected layers is done. If we want to perform multiclass classification, then the output is softmax. CNN is famous for image recognition and classification.

RNN-Recurrent Neural Networks

The idea behind RNNs is to make use of sequential information. The process of RNN is first, it allows us to score arbitrary sentences based on how likely they are to occur in the real world and then, a language model allows us to generate new text. RNNs are
called recurrent because they perform the same task for every element of a sequence, with the output depending on the previous computations. Some examples are Speech Recognition and Generating image Description.

(LSTMS)-Long Short Term Memory Systems

Long Short Term Memory networks called LSTMs – are a special kind of RNN, capable of learning long-term dependencies.

OBJECTIVE

The objective of this task is to apply classification of text into any of the data set of our choice and using CNN, RNN and LSTM. In this assignment, our approach and analyzation is shown. Also, which model out of CNN, RNN and LSTM is the best fit is analyzed and reported. The accuracy in all the cases is reviewed and according to that the best one is chosen. We are using the case of a crime investigation description as our data set.

APPROACHES/METHODS

1. The data generated from the police department according to the crimes announcement is uploaded. This data particularly has reports from different time intervals. This data file also contains attributes like date, category, address etc.
2. Then, this data set is entered to a protocol buffer format and after the data preprocessing, the data is separated.

WORKFLOW

1. Information of the class is generated after reading the data.
2. A vocabulary is built after the words in the data is analyzed.
3. Next is the data preprocessing, after which the chart is generated.
4. The placeholders needs to be created for the data sources and yields.
5. Initiation procedure makes the information received as non-linear.
6. The weights of the individual weights is used to duplicate the information variable.
7. The required model is created and the accuracy is reviewed.
8. Using tensor board, the map of the model that has been selected is generated.

DATASET

The data set used for our assignment was a criminal record categorized according to the city. It includes the records of Kansas City and and equally divided into training and test data.

PARAMETERS

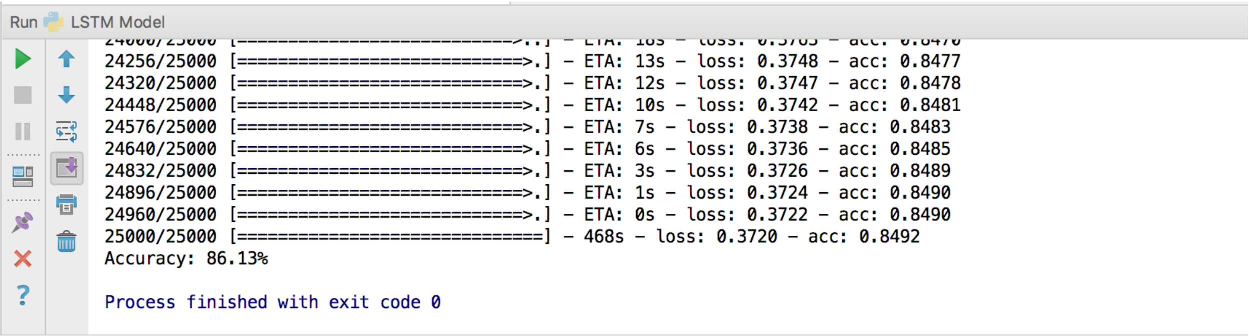
The parameters used for building these three models are:

1. For the functionality efficiency, an embedded layer has been generated and applied to the models.
2. Some of the parameters are number of words=5000, vector_length=32 and rev_len=500
3. The activation function used for making the data nonlinear was Sigmoid function.
4. The loss is evaluated in every iteration.
5. For the text classification, AdamOptimizer is used.
6. The CNN, RNN , LSTM model accuracy is generated.

EVALUATION & DISCUSSION

The Accuracies evaluated in all three models are shown below:

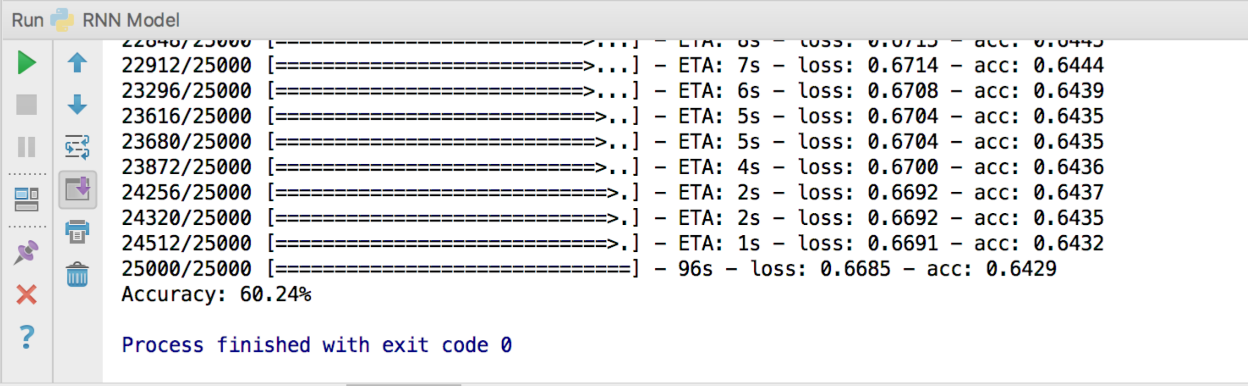
LSTM MODEL



```
24000/25000 [===========================>..] - ETA: 18s - loss: 0.3763 - acc: 0.8470
24256/25000 [============================>.] - ETA: 13s - loss: 0.3748 - acc: 0.8477
24320/25000 [============================>.] - ETA: 12s - loss: 0.3747 - acc: 0.8478
24448/25000 [============================>.] - ETA: 10s - loss: 0.3742 - acc: 0.8481
24576/25000 [============================>.] - ETA: 7s - loss: 0.3738 - acc: 0.8483
24640/25000 [============================>.] - ETA: 6s - loss: 0.3736 - acc: 0.8485
24832/25000 [============================>.] - ETA: 3s - loss: 0.3726 - acc: 0.8489
24896/25000 [============================>.] - ETA: 1s - loss: 0.3724 - acc: 0.8490
24960/25000 [============================>.] - ETA: 0s - loss: 0.3722 - acc: 0.8490
25000/25000 [==============================] - 468s - loss: 0.3720 - acc: 0.8492
Accuracy: 86.13%

Process finished with exit code 0
```

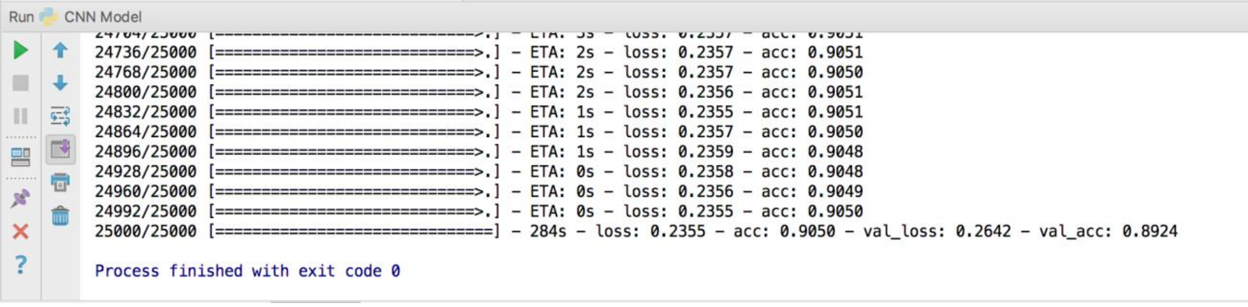
RNN MODEL



```
22848/25000 [==========================>...] - ETA: 8s - loss: 0.6715 - acc: 0.6445
22912/25000 [==========================>...] - ETA: 7s - loss: 0.6714 - acc: 0.6444
23296/25000 [==========================>...] - ETA: 6s - loss: 0.6708 - acc: 0.6439
23616/25000 [===========================>..] - ETA: 5s - loss: 0.6704 - acc: 0.6435
23680/25000 [===========================>..] - ETA: 5s - loss: 0.6704 - acc: 0.6435
23872/25000 [===========================>..] - ETA: 4s - loss: 0.6700 - acc: 0.6436
24256/25000 [============================>.] - ETA: 2s - loss: 0.6692 - acc: 0.6437
24320/25000 [============================>.] - ETA: 2s - loss: 0.6692 - acc: 0.6435
24512/25000 [============================>.] - ETA: 1s - loss: 0.6691 - acc: 0.6432
25000/25000 [==============================] - 96s - loss: 0.6685 - acc: 0.6429
Accuracy: 60.24%

Process finished with exit code 0
```

CNN MODEL



```
24704/25000 [============================>.] - ETA: 3s - loss: 0.2357 - acc: 0.9051
24736/25000 [============================>.] - ETA: 2s - loss: 0.2357 - acc: 0.9051
24768/25000 [============================>.] - ETA: 2s - loss: 0.2357 - acc: 0.9050
24800/25000 [============================>.] - ETA: 2s - loss: 0.2356 - acc: 0.9051
24832/25000 [============================>.] - ETA: 1s - loss: 0.2355 - acc: 0.9051
24864/25000 [============================>.] - ETA: 1s - loss: 0.2357 - acc: 0.9050
24896/25000 [============================>.] - ETA: 1s - loss: 0.2359 - acc: 0.9048
24928/25000 [============================>.] - ETA: 0s - loss: 0.2358 - acc: 0.9048
24960/25000 [============================>.] - ETA: 0s - loss: 0.2356 - acc: 0.9049
24992/25000 [============================>.] - ETA: 0s - loss: 0.2355 - acc: 0.9050
25000/25000 [==============================] - 284s - loss: 0.2355 - acc: 0.9050 - val_loss: 0.2642 - val_acc: 0.8924

Process finished with exit code 0
```

So, to conclude, the CNN turns out to be the best fit among all three models. We can see that the accuracy is more for CNN model and RNN , the least accuracy.

CONCLUSION

Our data set is reviewed by three models CNN, RNN, LSTM and accuracy was generated. So, we find that CNN model was better than the rest. It is accurate and efficient. LSTM takes the second place in accuracy.