**Python/Deep Learning**

**Deep Learning**

**Lab Assignment 3**

**Submitted By**

**Vamsi Krishna Challa**

**16242688 / Class id :10**

**Introduction:**

This lab assignment is all about implementing Text classification using CNN, RNN AND LSTM over a dataset that is not used in class and also comparing the results of them. We run the classification with text on convolutional neural network, feed forward neural network and LSTM. LSTM is basically a form of RNN, which comes into action when the need of overcoming vanishing gradient arises.

**Objective:**

* The primary objective is to perform text classification over CNN, RNN AND RNN using a new data set that is not used in class and then compare the results of the three classifications.
* We then analyze every classification over the neural network and compare the results.

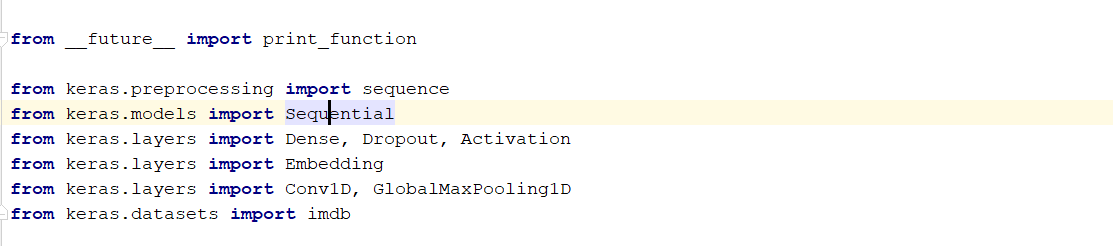
**Approaches and Methods:**

* We first load the data-set that can be found in the data folder into the program for performing classification.
* Then setup the training parameters to train the model
* Import the classes and functions that are needed to train the model.
* We then define the embedding layer to train the model.
* Here we consider using adam optimizer to construct CNN model.
* As the data might not be in a processable way, we need to process the data very much like removing the spaces, unwanted symbols etc before we train the model.
* We then run the models and analyze the results.

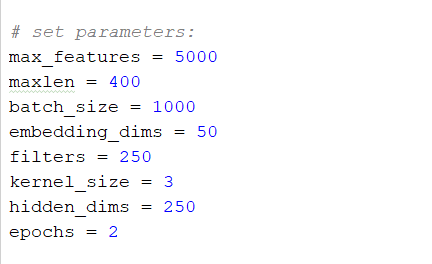
**Workflow:**

* We initially train and run the models to obtain accuracy and loss over various classifications on the same dataset.
* Load the data in train and test variables
* Set the parameters
* Compute accuracy and loss parameters.
* Repeat the process for all the three text classifications over CNN , RNN and LSTM.
* Every time you run the text classification over a neural network for the considered dataset, make note of the accuracy and loss values that are obtained when running the model.

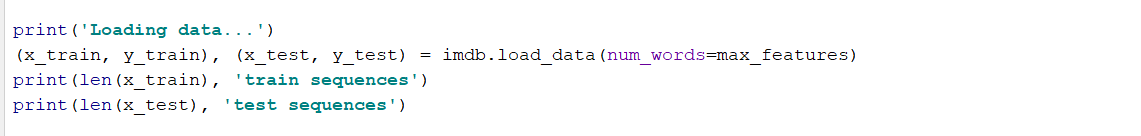
**Importing packages and datasets:**



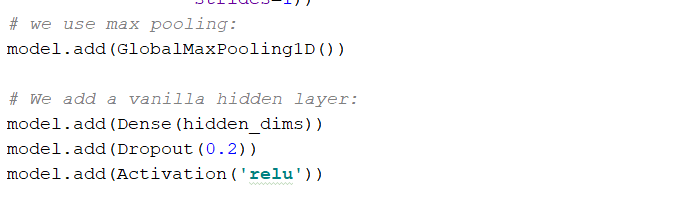
**Set and define the parameters:**



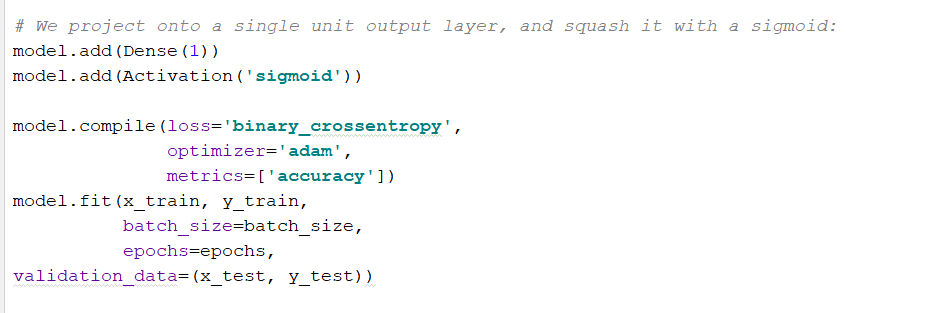
**Load the data:**



**Max pooling and implementing hidden layers:**



**Run the model:**

(the above code snippets are for cnn :) 

**Configuration:**

* PYCHARM : 3.4
* TensorBoard

**Data-Set:**

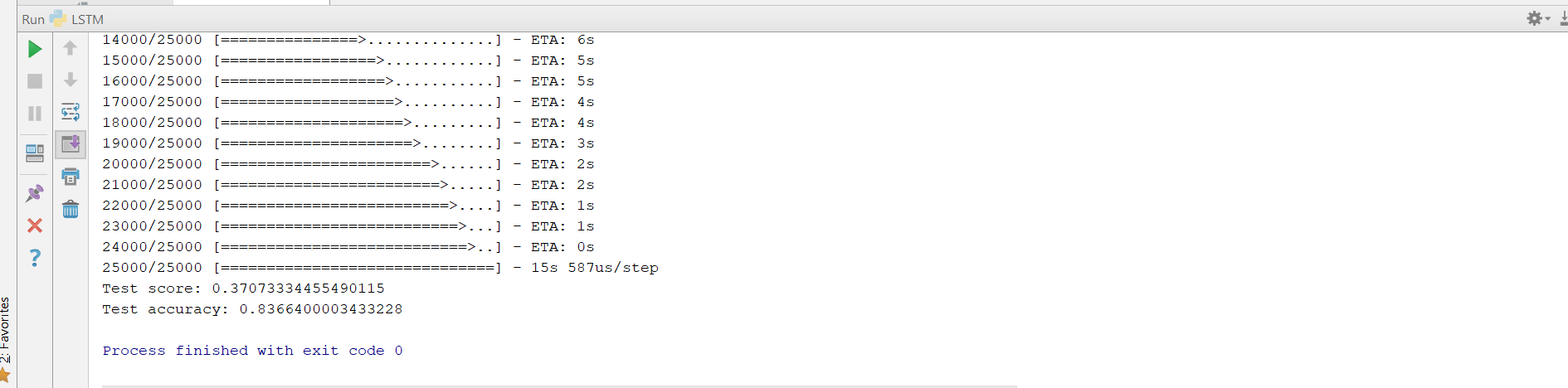
Here I have considered a IMDB dataset that can be obtained from inbuilt Keras package. It has 25000 sample data that can be used to train and test out model.

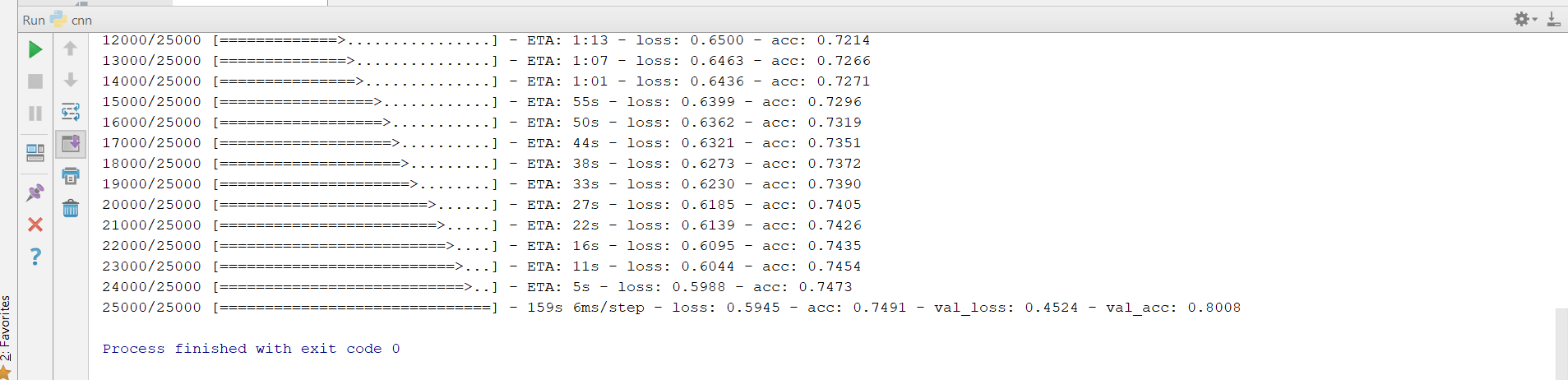
**Parameters:**

Below are the parameters that I have set for running this model

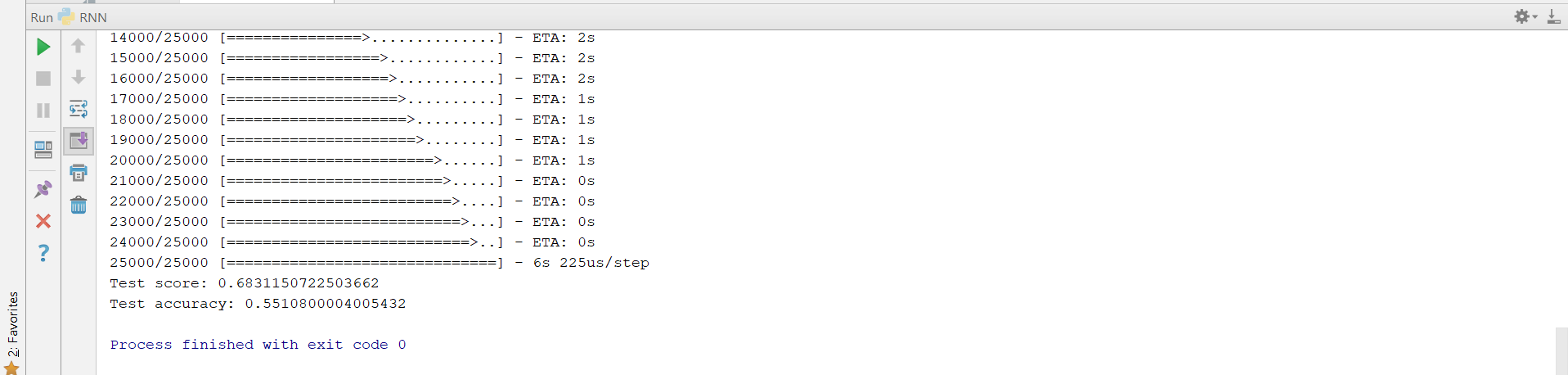
* max\_features
* maxlen
* batch\_size
* embedding\_dims
* filters
* kernel\_size
* hidden\_dims
* epochs

**Evaluation and discussion:**

When we perform lstm classification over the considered imdb data set, we obtain accuracy of about 0.836. 

When we perform cnn based text classification over the considered imdb dataset, we obtain an accuracy of about 0.804. 

When we perform rnn based text classification over the considered imdb dataset, we obtain an accuracy of about 0.551.



**Conclusion:**

Over this dataset we observe that the highest accuracy is obtained when we perform text classification over the imdb data set while considering the case of lstm. LSTM performs better than both rnn and cnn.

References:

* <https://www.tensorflow.org/programmers_guide/>
* <https://github.com/keras-team>