CSEE5590/490:

Python and Deep Learning Programming

(2018 Fall)

*Python Lab Assignment 3*

Submitted On:

07 November, 2018

Submitted By:

Name: Farid Uddin Ahmed

Class ID: 02

Name: Zarin Tasnim Sandhie

Class ID: 26



LIST OF THE DOCUMENTATIONS:

1. Author
2. Objective
3. Features
4. Configuration
5. Input/ Output Screenshots
6. Implementation & Code Snippet
7. Evaluation & Explanation
8. Conclusion
9. Links
10. References

AUTHORS

This report contains all the documents for the Lab Assignment #3. The assignment was done by Farid Uddin Ahmed (Class ID 2) and Zarin Tasnim Sandhie (Class ID 26), both are graduate student majoring in Electrical Engineering Department at University of Missouri Kansas City (UMKC).

OBJECTIVE

In the last three weeks of Python/deep learning class, we got to work with the basics of deep learning like neural network, keras, linear regression, word embeddings etc. The definition and working principle of the following topics were taught in the class:

* Keras
* TensorFlow
* Neural Networks
* Linear Regression
* Back-propagation
* Gradient Descent (Optimization Algorithm)
* Cost/Loss Functions
* Activation Function
* Bag of Words
* Word Embedding (representation of text data)
* word2vec model

This assignment includes all of the things mentioned above.

* The dataset chosen for problem 1 is Boston housing dataset. This dataset contains 13 different column which represents 12 input datasets and 1 output datasets. Depending on the input datasets, the price of the houses are given.
* The dataset for problem 2 is diabetes dataset. As second problem works with logistic regression, diabetes dataset is selected. We know, the results for logistic regression is binary (0 or 1), that is the reason for selecting this dataset.
* The first problem deals with the implementation of linear regression on the dataset. Then we have to observe the graph in TensorBoard and discuss the effect of changing different parameters in the model.
* The second problem is similar to first problem. Here we have to implement logistic regression on the same dataset and observe the change in accuracy by changing the parameters.
* For both the problems, we observed the effect of changing the parameters: learning rate, batch size, optimizer and activation function

FEATURES

The features of all the problems are discussed below:

Problem 1:

Implement the Linear Regression with any data set of your choice except the datasets being discussed in the class.

1. Show the graph in TensorBoard
2. Plot the loss and then change the below parameter and report your view how the result changes in each case
3. Learning rate
4. Batch size
5. Optimizer
6. Activation function

Problem 2:

Implement Logistic Regression with any data set of your choice.

a. Show the graph in TensorBoard.

b. Show the Loss in TensorBoard.

c. Use score = model.evaluate (x\_text,y\_test) and then print (‘test accuracy’, score[1]) to print the accuracy.

d. Change three hyperparameter and report how the accuracy changes.

CONFIGURATION

For executing the assignment, coding was done with Python software version 3.6. The simulation was in software: Anaconda (Spider).

INPUT/OUTPUT SCREENSHOTS

Problem 1:

* The “boston” datasets in the seaborn library is taken as input dataset.
* When simulated, the output plot shows the training and testing loss with the change of epochs.
* The plots were also seen in TensorBoard.
* The effect of changing the learning rate, batch size, optimizer and activation function are also observed on linear regression.

Output:

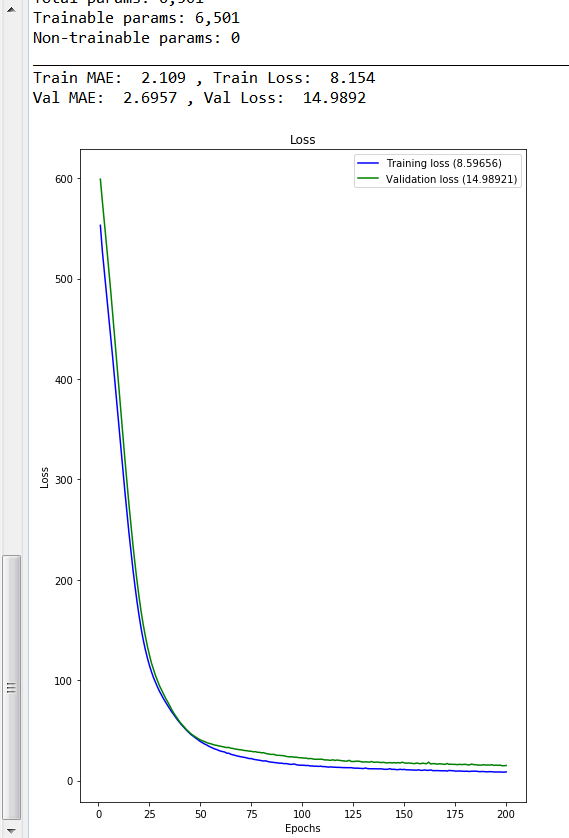
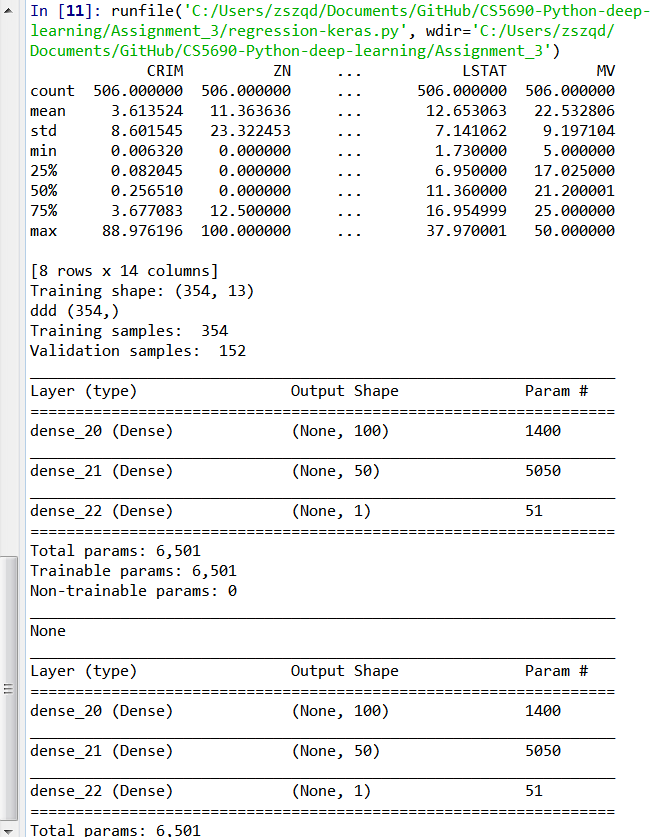


Figure 1: The model specification and training and testing loss for linear regression with Learning rate = 0.001, batch size = 128, optimizer = RMSprop and activation function = tanh

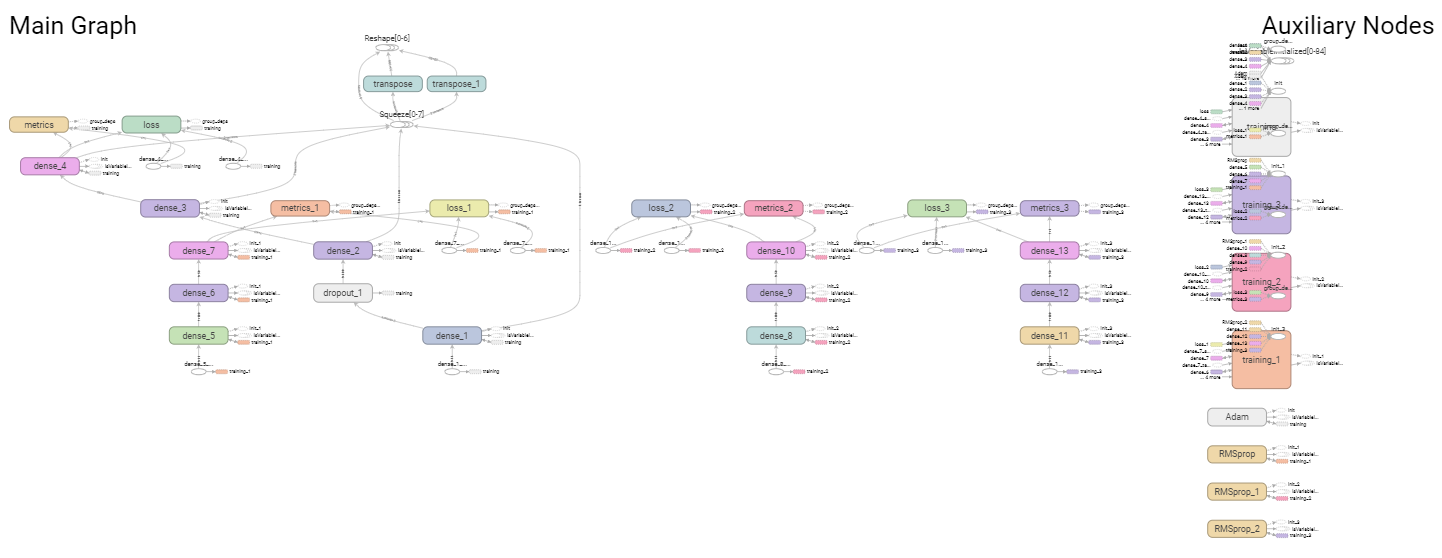


Figure 2: TensorBoard Main Graph

|  |  |
| --- | --- |
|  |  |

Figure 3: The TensorBoard graphs of loss and mean\_absolute\_error for linear regression with Learning rate = 0.001, batch size = 128, optimizer = RMSprop and activation function = tanh

The effect of changing different parameters on the loss are given below:

|  |  |
| --- | --- |
|  |  |
|  |  |

Figure 4: Effect of changing Learning rate: a) learning rate = 0.01 (Accurate learning) b) learning rate = 0.0001 (fast learning)

|  |  |
| --- | --- |
|  |  |
|  |  |

Figure 5: Effect of changing Batch size: a) batch size = 64 b) batch size = 256

|  |  |
| --- | --- |
|  |  |
|  |  |

Figure 6: Effect of changing Optimizer: a) RMSprop b) Adam

|  |  |
| --- | --- |
|  |  |
|  |  |

Figure 7: Effect of changing Activation function: a) tanh b) relu

Problem 2:

* The “boston” datasets is taken as input dataset.
* When simulated, the output plot shows the training and testing accuracy
* The plots were seen in TensorBoard.
* The effect of changing the batch size, optimizer and activation function are also observed on logistic regression.

Output:

|  |  |
| --- | --- |
|  |  |
|  |  |

Figure 8: The model specification and training and testing accuracy and loss for logistic regression with batch size = 128, optimizer = adam and activation function = tanh

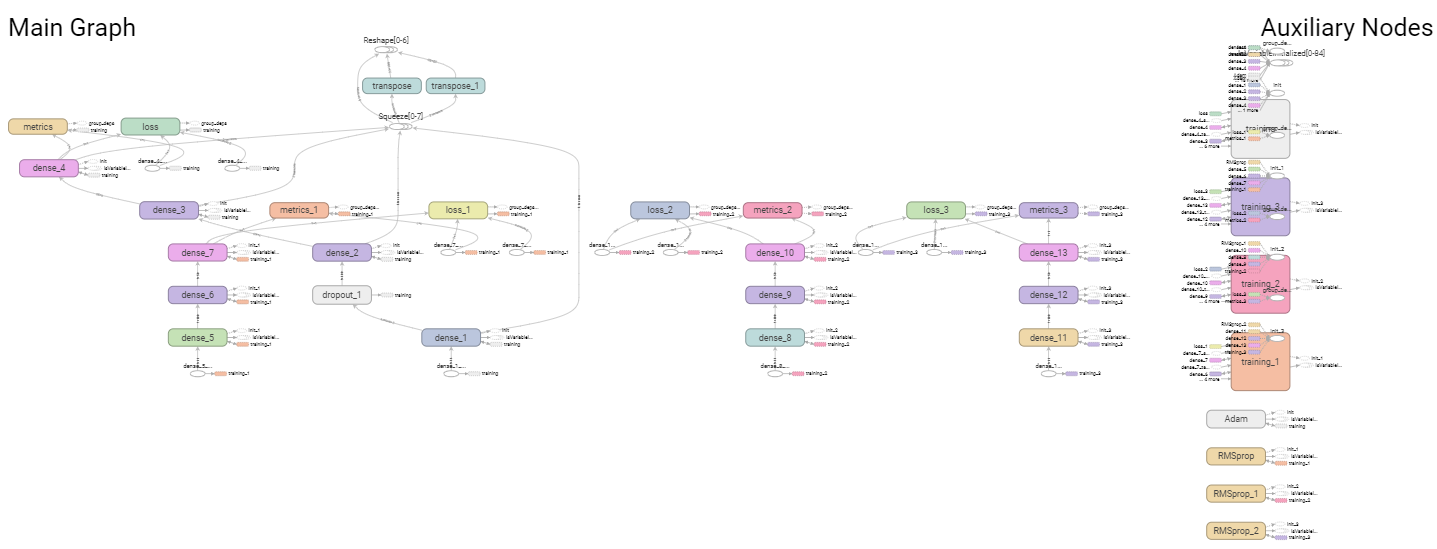


Figure 9: TensorBoard Main Graph

The effect of changing different parameters on accuracy are given in the following table (default parameters used here are: batch size=128, optimizer=adam, activation function=tanh):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Changing Batch Size | | | Changing Optimizer | | Changing Activation Function | |
|  | Batch size = 64 | Batch Size = 128 | Batch size = 256 | RMSprop | Adam | tanh | relu |
| Train Accuracy | 0.43 | 0.44 | 0.49 | 0.55 | 0.47 | 0.46 | 0.55 |
| Test Accuracy | 0.57 | 0.61 | 0.66 | 0.74 | 0.64 | 0.64 | 0.74 |

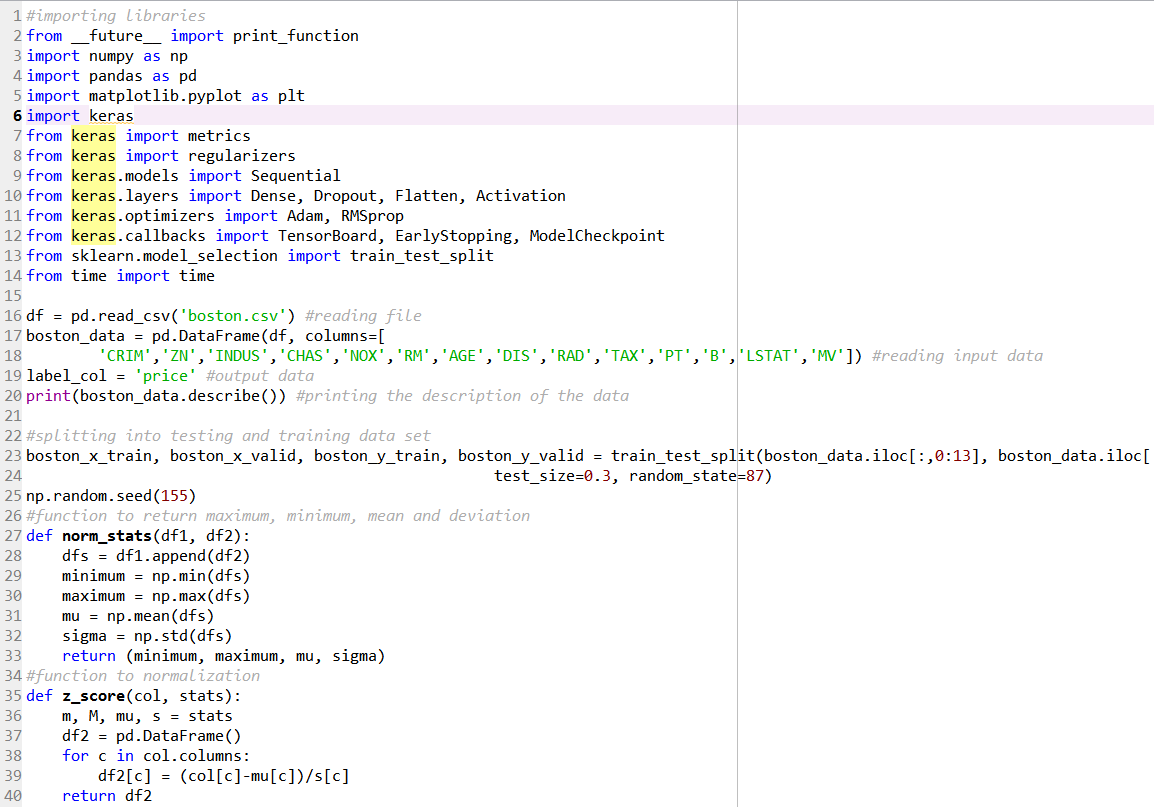
Table 1: Effect of changing parameters in logistic regression

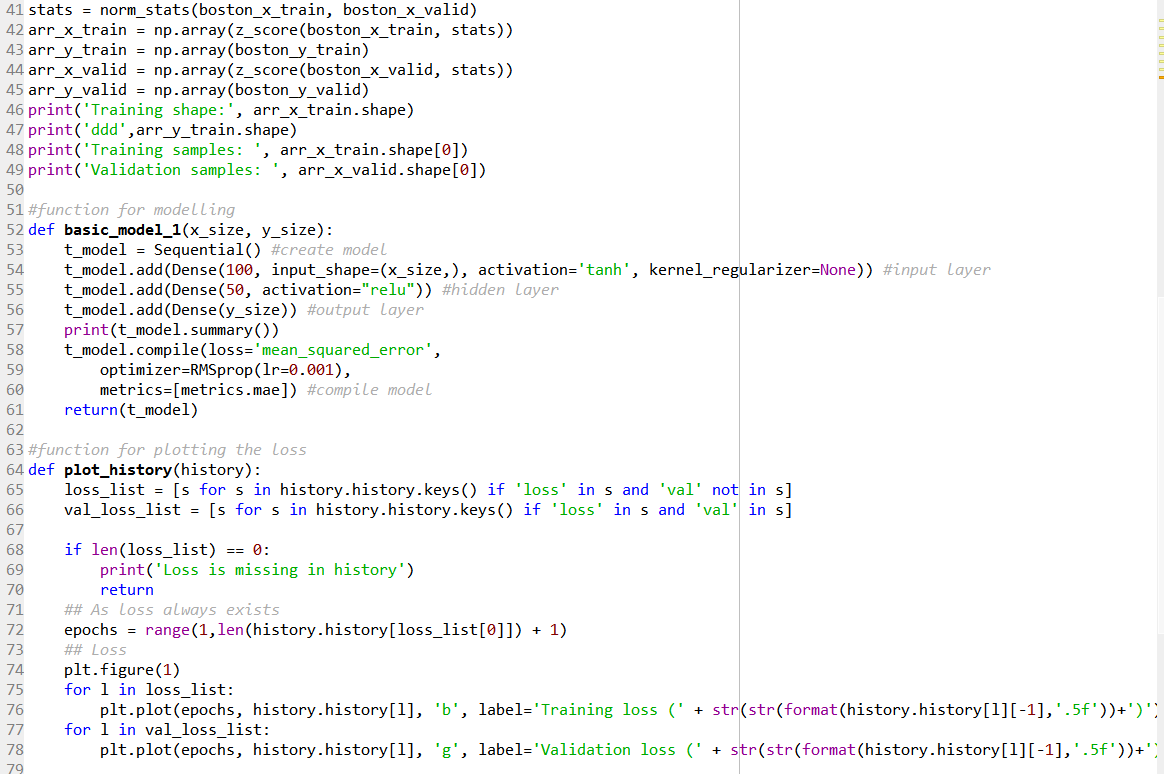
IMPLEMENTATION & CODE SNIPPET

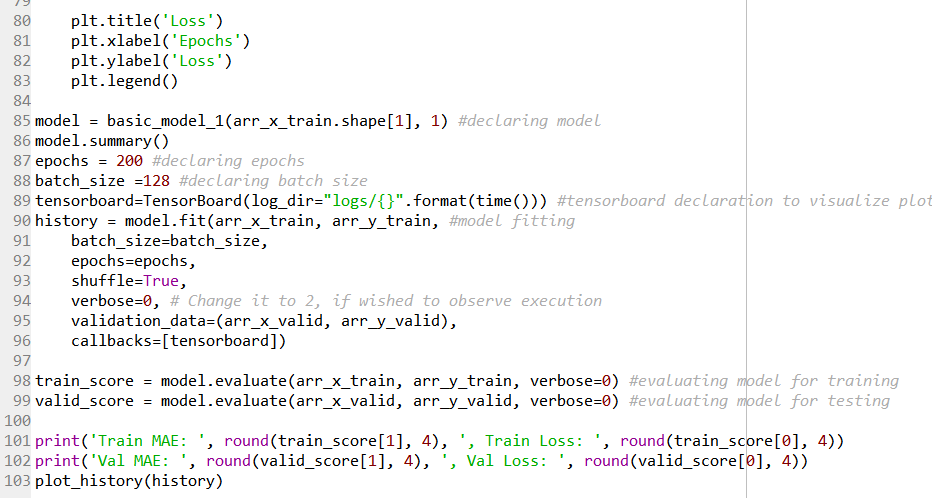
Problem 1:

* Libraries are imported.
* The code takes “boston” dataset as input dataset.
* The dataset is split into testing and training dataset.
* Functions are created for returning minimum, maximum, mean and deviation.
* Function is defined to create model with required layers and specifications.
* Model fitting is done.
* Then testing and training loss is shown.

Code:

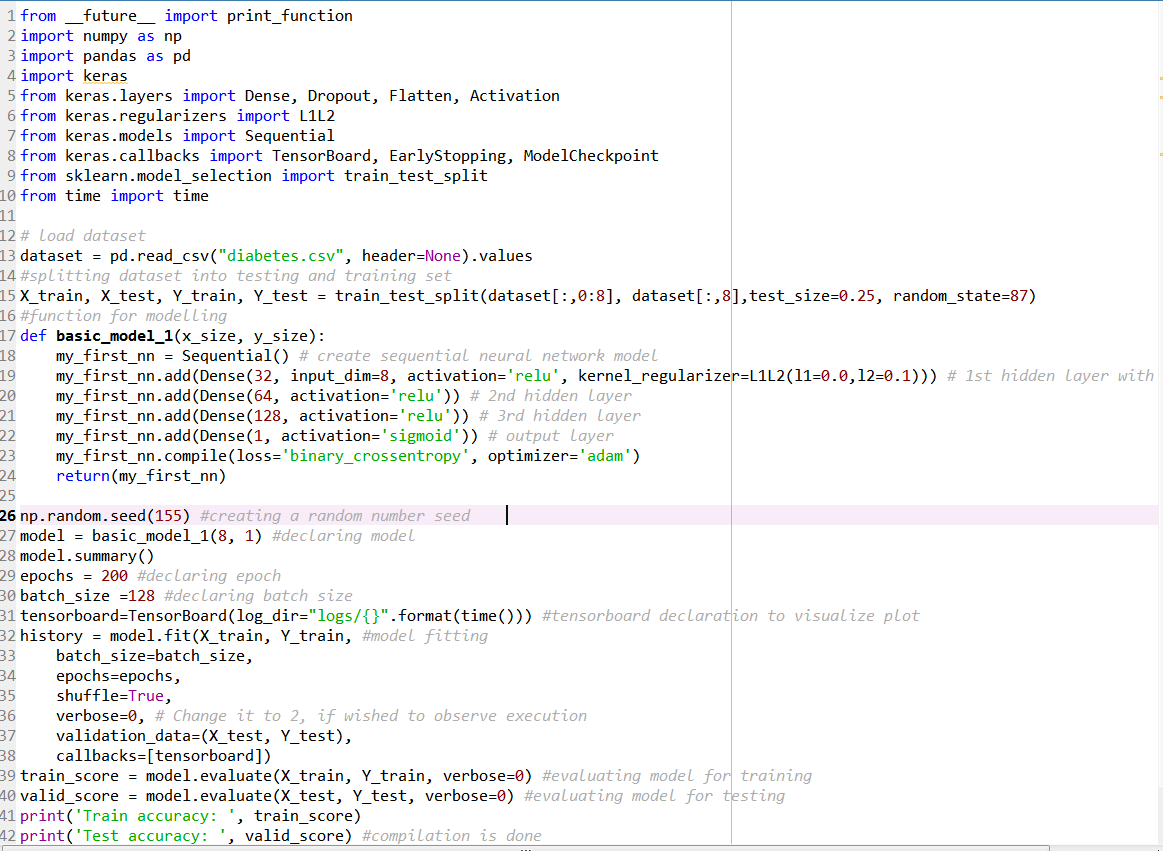






Problem 2:

* Required libraries are imported.
* The “diabetes” datasets is taken as input dataset.
* The dataset is split into testing and training dataset.
* Function is defined to create model with required layers and specifications.
* Model fitting is done.
* When simulated, the output plot shows the training and testing accuracy.
* The plots can be seen in TensorBoard.



EVALUATION & EXPLANATION:

Problem 1:

The accuracy found in the Naïve Bayes classification with digits dataset is around 80%. And it shows the number of each digits in the dataset. The total number of data is around 1700. And the number of each digit is around 170.

Problem 2:

This problem works on different types of kernel.

For linear kernel, the accuracy on the test set is around 98%. Though it vary slightly for different case. The value of C parameter was changed from 0 to 10. But the results were quite similar.

For poly kernel, the accuracy is in between 97%-99%. By changing the value of degree parameter, the change in accuracy can be observed. The accuracy rises at first with the increase of degree value and then starts to decrease. This decrement continues for a large value of degree (20).

The accuracy on test set is lowest with rbf kernel. The accuracy in the range of 10% for rbf kernel. By changing the value of gamma parameter within the range of 0 to 10, the accuracy increases slightly and then decreases.

CONCLUSION

All the required problems were solved successfully during this assignment. During this assignment, we learned the use and advantages of different kinds of classification techniques. We also learned about the NLP (Natural Language Processing) and the built-in functions of NLTK (Natural Language ToolKit).

LINKS

Video Link:

<https://www.youtube.com/watch?v=oy09gVxFOKo&feature=youtu.be>

Github Link:

<https://github.com/Sandhie177/CS5690-Python-deep-learning/tree/master/Assignment_2>

REFERENCES

1. <https://github.com/wade12/WikiScraper/blob/master/>
2. <https://www.w3resource.com/python-exercises/>
3. <https://www.learnpython.org/>
4. <https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/>