PROGRAMMING ASSIGNMENT 2 – Handwritten Digits Classification

CSE 574 Fall 2021 Introduction to Machine Learning

Group Number : 15

Submitted By :

S.No	Name	Email	UB Id Number
1	Lavanya Nadanasabapathi	lnadanas@buffalo.edu	50367936
2	Aravind Balakrishnan	balakri2@buffalo.edu	50412479
3	Anirudh Sowmyanarayanan	asowmyan@buffalo.edu	50415686

In this Project, we have implemented a Multilayer Perceptron Neural Network and evaluated its performance in classifying handwritten digits.

To update the weights of the model for predictive learning, our Machine Learning Model uses forward feed and back propagation methods.

This project examines the relationships between the number of hidden nodes, the regularization coefficient, and the maximum iterations of the conjugate gradient algorithm when performing an optimization task, as well as the impact of these parameters on the accuracy and training time of neural networks.

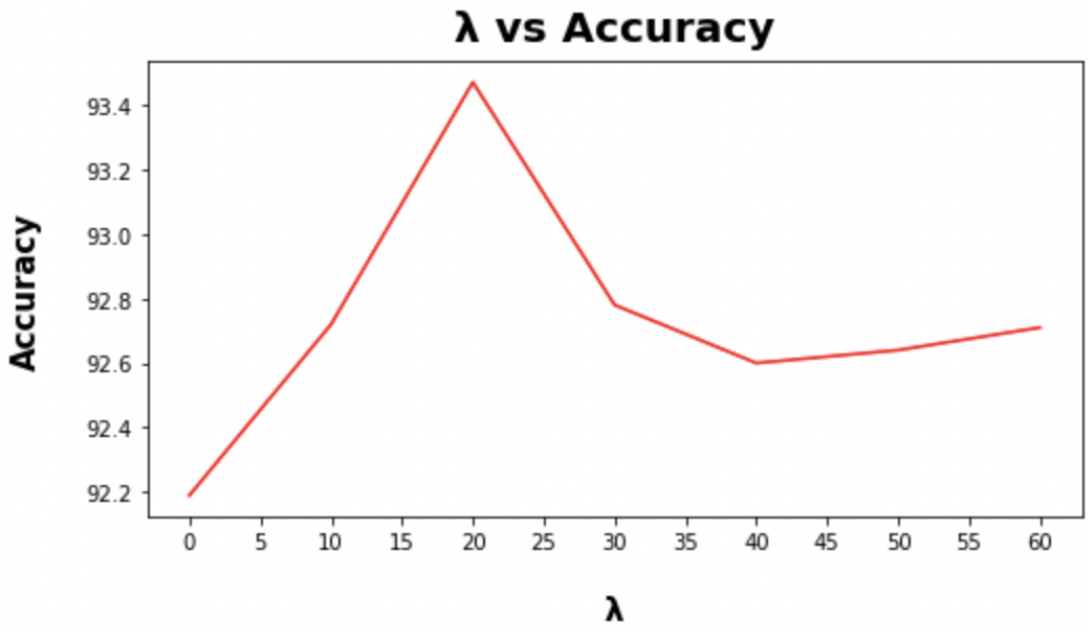
Selecting Hyper Parameters using nnScript.py :

The main aim is to compare the accuracy and time taken based on different values hidden units and lambda.

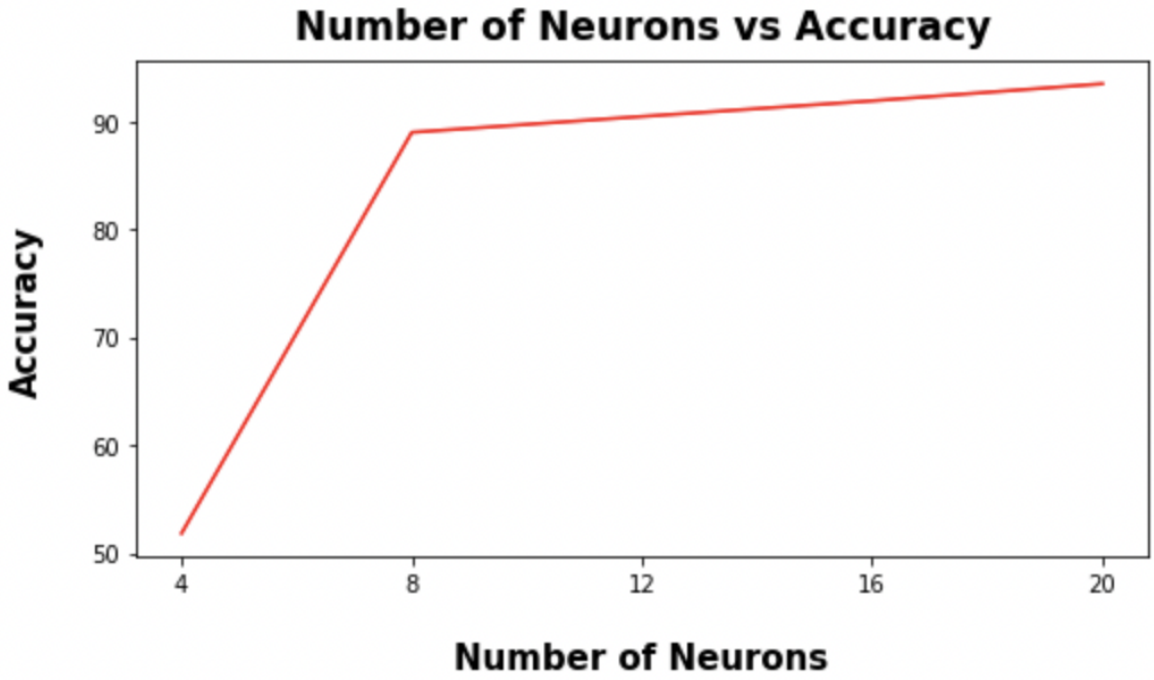
Neurons	Lambda	Training Accuracy	Validation Accuracy	Test Accuracy	Execution Time
4	0	75.11	74.33	75.9	61.476873
4	10	22.594	21.70999	22.36	73.900757
4	20	53.14	51.81	53.01000	72.0493
4	30	46.62	45.49	46.87	66.66914
4	40	68.848	67.29	68.25	67.20999
4	50	47.4	45.44000	47.5	65.1224
4	60	66.2839	64.41	65.57	69.695354
8	0	81.568	80.07	82.0	76.153369
8	10	86.68	85.91	86.47	70.513974
8	20	89.62	88.97	89.149	73.216768
8	30	89.034	88.8	88.29	71.68739
8	40	86.6859	85.87	86.2299	67.41144
8	50	90.2100	89.5399	90.22	69.446758
8	60	89.53	88.4900	89.35	65.68748
12	0	91.252	90.4	91.09	84.230311

12	10	90.286	90.09	90.62	83.19667
12	20	90.9540	90.44	90.92	81.1367640
12	30	92.096	90.41	91.79	83.388833
12	40	91.686	91.28	91.56	79.61507
12	50	91.298	90.2599	90.72	79.06721
12	60	91.688	91.66	91.78	84.44773
16	0	93.306	92.72	93.07	88.04918
16	10	92.706	92.2	92.11	79.4820845
16	20	92.964	91.89	92.41	81.93713
16	30	93.17	92.1799	92.57	81.370678
16	40	92.723	91.94	92.7899	78.700
16	50	92.658	92.15	92.92	74.66488
16	60	91.862	91.44	92.25	83.7624003
20	0	92.857	92.190	92.69	97.35811
20	10	93.37	92.72	93.08	91.889648
20	**20**	**93.988**	**93.47**	**93.65**	**86.419619**
20	30	93.708	92.78	93.21	91.73333
20	40	93.036	92.60	93.08	88.014763
20	50	92.85	92.64	92.99	88.743404
20	60	92.83	92.7100	92.86	83.924710

When Lambda() = 20 and Neurons = 20 are the optimal values for the given data set as we can see that the accuracy achieved is the highest.



For Optimal Value of Lambda where Lambda = 20, The accuracy peaks and the reduces.



The above graph indicates that as the number of Neurons increases, the accuracy will increase as well.

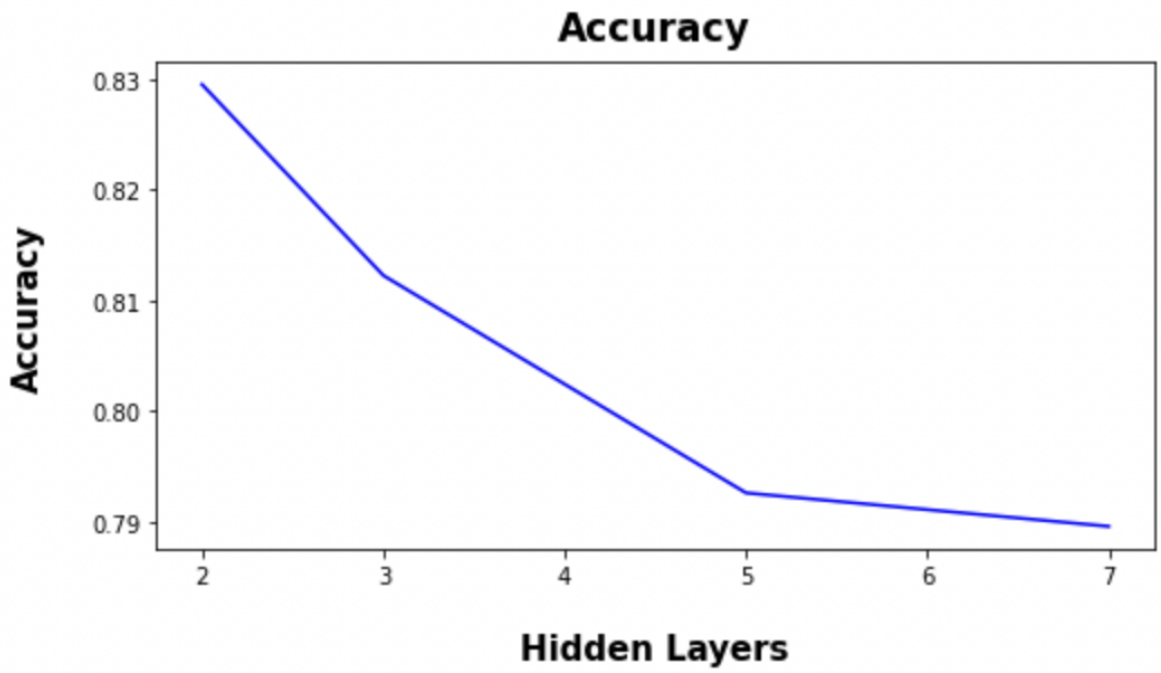
Comparing Single Layer and Multi Layer :

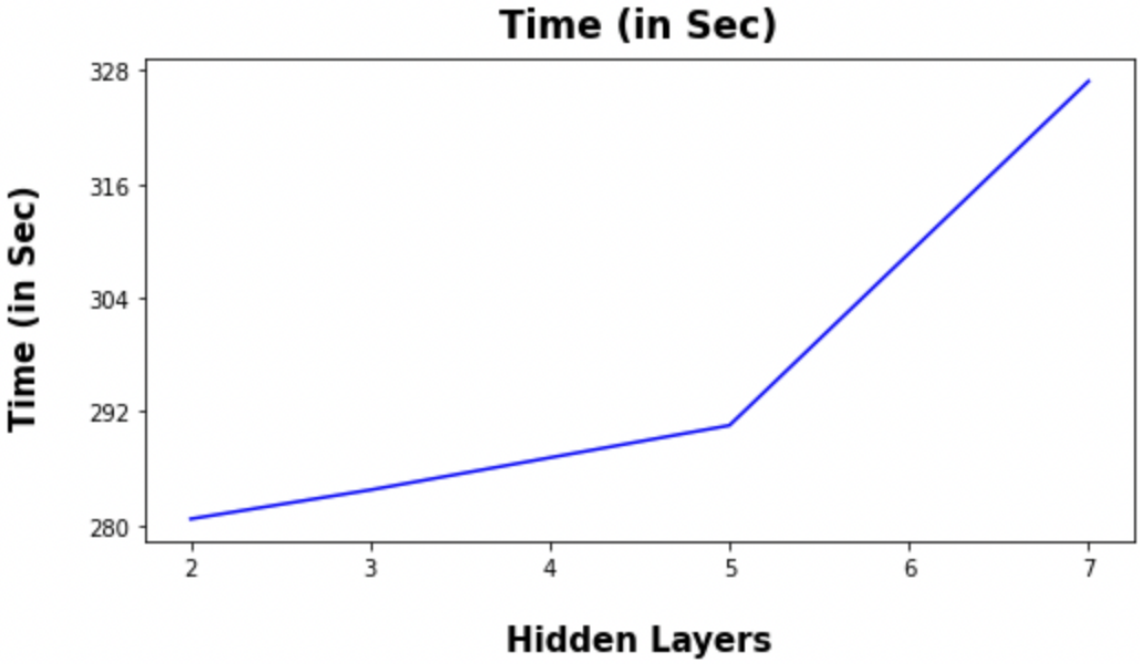
Accuracy of Single Neural Network on Mnist Dataset using facennScript.py

	facennScript.py
Training Set Accuracy	84.01421800947867
Validation Test Accuracy	82.5140712945591
Test Set Accuracy	84.74640423921271

Accuracy of deep neural network on celebA dataset using deepnnScript.py

No. of Layers	Accuracy	Training time (In sec)
2	0.829	280.68
3	0.822	283.76
5	0.925	290.53
7	0.789	326.81





Above graphs show Hidden Layers vs Accuracy and Hidden Layers vs time.

Running the cnnScript.py

After running the cnnScript the confusion matrix obtained is below :

```
Confusion Matrix:
[[ 970    0    1    0    0    5    1    1    2    0]
 [   0 1130    2    0    0    0    1    1    1    0]
 [   0    1 1018    4    1    0    0    4    4    0]
 [   0    0    0 1006    0    1    0    1    2    0]
 [   0    0    0    0  980    0    0    1    0    1]
 [   2    0    0    8    0  878    1    1    2    0]
 [   6    2    1    0    6   10  933    0    0    0]
 [   0    1    7    1    0    0    0 1017    2    0]
 [   2    0    3    4    3    3    0    3  953    3]
 [   1    4    0    4    8    5    0    4    2  981]]

Time usage: 1:00:01
Accuracy on Test-Set: 98.4% (9844 / 10000)
```

This is the Confusion Matrix that is obtained after 10,000 iterations.