

Introduction to Machine Learning (Spring 2020)
Programming Assignment 2 – Group 41
Report

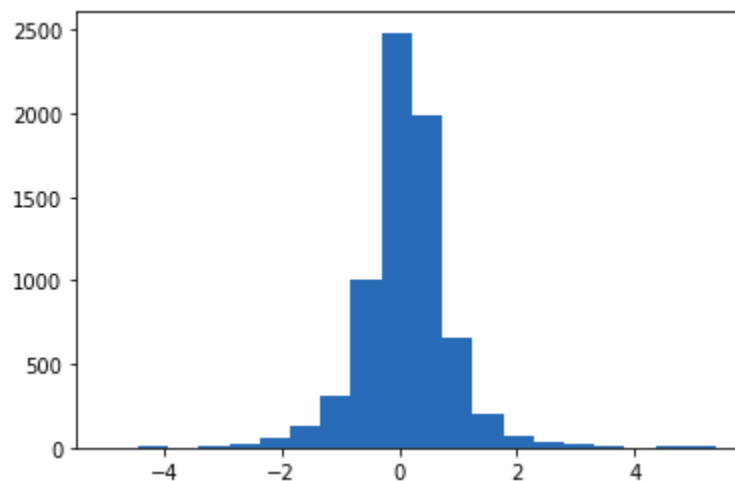
Vaibhav Chhajed (vchhajed)
Yeshwanth Badineni (ybadineni)
Hemant Koti (hemantko)

Part I – Sentiment Analysis

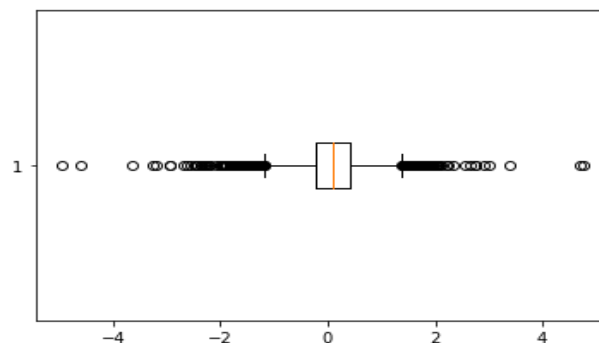
Approach 1 – Word Count Vectorization

Report 1: Describe your word features and the classification model and report the accuracy of your model on the test data.

- Describe your word features and the classification model and report the accuracy of your model on the test data.
- We have captured positive word list and negative word list from two files reviews.txt and labels.txt which correspond to features and labels respectively using the collection Counter library.
- We picked out words from review through split function and verified that they are in our vocabulary list and ignored others (i.e. ‘’, ‘’, “)
- Using the word count greater than 100 we calculated the positive to negative ratio of each word.
- To get into the specified range we converted into the log values.
- To understand the data more clearly, we plotted the histogram and boxplot.
- Histogram



- Boxplot



Introduction to Machine Learning (Spring 2020)
Programming Assignment 2 – Group 41
Report

Vaibhav Chhajed (vchhajed)
Yeshwanth Badineni (ybadineni)
Hemant Koti (hemantko)

- Through the plots, we understood that the deciding factor of the review was the use of strong (higher positive to negative ratio) positive words and strong (lower positive to negative ratio) negative words respectively.
- After verifying different percentiles range we came across the word feature which gave an accuracy of 77.1% to be in a range greater than 80 percentile and less than 10 percentiles as a threshold of positive and negative words respectively.

Approach 2 - Neural Network-Based Sentiment Classification

Report 2: Report the accuracy and running time of the vanilla neural network. Is it better or worse than your rule-based algorithm implemented in Approach 1?

- We have implemented ignore words function to compare the rule-based and machine learning approach.
- We have considered the same threshold (i.e. greater than 80 percentile and less than 10 percentile) to ignore words.
- We can observe that machine learning gives better accuracy than a rule-based approach. This is because machine learning can generalize the new data better, in case of a rule-based approach we need to add rules for every new data we see.
- Also, a machine learning-based approach is robust to outliers and anomalies. For instance, a statement like *"this is not a bad movie"* will be incorrectly classified as negative by a rule-based approach, whereas machine learning models can accurately classify these kinds of sentences due to its adaptability properties.

Report 3: Report the performance of the model (both in terms of accuracy and time using different values for the hidden layer width, number of hidden layers, number of epochs and a set of ignored words. What has the most impact on performance?

With CPU as background

```
Epochs:50, Weight=10, layer=1  
Train acc: 0.904375, Test_acc: 0.841250  
Time elapsed - 22.347902297973633 seconds.
```

With GPU as background

50 Epochs with different weight values

```
Epochs=50, Weight=5, layer=1  
Train acc: 0.920000, Test_acc: 0.855000  
Time elapsed - 13.598512172698975 seconds.
```

```
Epochs=50, Weight=10, layer=1  
Train acc: 0.905042, Test_acc: 0.851250
```

Introduction to Machine Learning (Spring 2020)
Programming Assignment 2 – Group 41
Report

Vaibhav Chhajed (vchhajed)
Yeshwanth Badineni (ybadineni)
Hemant Koti (hemantko)

Time elapsed - 13.796868562698364 seconds.

Epochs=50, Weight=50, layer=1

Train acc: 0.907167, Test_acc: 0.848750

Time elapsed - 14.32064151763916 seconds.

Epochs=50, Weight=100, layer=1

Train acc: 0.904333, Test_acc: 0.838750

Time elapsed - 14.629113674163818 seconds.

Performance impact with a different number of weights: We can observe that with increasing weights the accuracy of the neural network decreases.

100 Epochs with different weight values

Epochs=100, Weight=5, layer=1

Train acc: 0.923625, Test_acc: 0.852500

Time elapsed - 28.065927028656006 seconds.

Epochs=100, Weight=10, layer=1

Train acc: 0.906042, Test_acc: 0.845000

Time elapsed - 28.596012353897095 seconds.

Epochs=100, Weight=50, layer=1

Train acc: 0.911333, Test_acc: 0.841250

Time elapsed - 29.032610177993774 seconds.

Epochs=100, Weight=100, layer=1

Train acc: 0.909375, Test_acc: 0.840000

Time elapsed - 29.917771577835083 seconds.

Performance impact with a higher number of epochs: We can observe that with an increased number of epochs the accuracy of the neural network decreases compared to the previous weights.

50 Epochs with different hidden layers

Epochs=50, Weight=10 (h1->10->h2->10->out->2), layer=2

Introduction to Machine Learning (Spring 2020)
Programming Assignment 2 – Group 41
Report

Vaibhav Chhajed (vchhajed)
Yeshwanth Badineni (ybadineni)
Hemant Koti (hemantko)

Train acc: 0.909292, Test_acc: 0.853750

Time elapsed - 14.945465087890625 seconds.

Epochs=50, Weight=10 (h1->10->h2->10->h3->10->out->2), layer=3

Train acc: 0.887417, Test_acc: 0.817500

Time elapsed - 15.128944396972656 seconds.

Epochs=50, Weight(h1->10->h2->5->out->2), layer=2

Train acc: 0.910292, Test_acc: 0.855000

Time elapsed - 14.639018297195435 seconds.

Epochs=50, Weight(h1->10->h2->10->h3->5->out->2), layer=3

Train acc: 0.899792, Test_acc: 0.852500

Time elapsed - 15.199421882629395 seconds.

Performance impact with a different number of hidden layers: We can observe that with an increased number of hidden layers the accuracy of the neural network increases.

We tested with different intervals for ignore_words function and picked the most optimum interval between 10 and 80 to ignore words and observed performance of 84%.

Out of all the combinations, the maximum performance impact is observed with an increased number of hidden layers.

Part II - Image Classification on the AI Quick Draw Dataset

- 1. Run the evaluation of the 2 hidden layers neural network in the notebook - PA2-Part2.ipynb and report the test accuracy and the run time.**

Model summary for 28 x 28 image size with 500 epochs and 2 hidden layers are listed below.

Introduction to Machine Learning (Spring 2020)
Programming Assignment 2 – Group 41
Report

Vaibhav Chhajed (vchhajed)
Yeshwanth Badineni (ybadineni)
Hemant Koti (hemantko)

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 256)	200960
dense_2 (Dense)	(None, 32)	8224
dense_3 (Dense)	(None, 10)	330
Total params: 209,514		
Trainable params: 209,514		
Non-trainable params: 0		

```
Epoch 1/500
100000/100000 [=====] - 8s 82us/step - loss: 3.2113 - accuracy: 0.6147
Epoch 2/500
100000/100000 [=====] - 8s 85us/step - loss: 1.4194 - accuracy: 0.6948
Epoch 3/500
100000/100000 [=====] - 8s 85us/step - loss: 1.3944 - accuracy: 0.7077
Epoch 4/500
100000/100000 [=====] - 8s 83us/step - loss: 1.3804 - accuracy: 0.7148
Epoch 5/500
100000/100000 [=====] - 9s 87us/step - loss: 1.3802 - accuracy: 0.7207
Epoch 497/500
100000/100000 [=====] - 5s 53us/step - loss: 0.7294 - accuracy: 0.7989
Epoch 498/500
100000/100000 [=====] - 5s 53us/step - loss: 0.7674 - accuracy: 0.7985
Epoch 499/500
100000/100000 [=====] - 5s 49us/step - loss: 0.7316 - accuracy: 0.7987
Epoch 500/500
100000/100000 [=====] - 5s 46us/step - loss: 0.7454 - accuracy: 0.7986
Wall time: 47min 42s
```

```
Report 4 - Question 1 - 28 x 28 image with two hidden layers - Testing accuracy
```

```
predict_test = model.predict_classes(data_test)
acc_test = np.where(label_test1 == predict_test)[0].shape[0]/data_test.shape[0]
print('Testing accuracy {}'.format(acc_test))
```

Testing accuracy 0.68376

Training accuracy for the given model comes around 70% to 79% as shown in the epochs figure above. Testing accuracy for the given model comes around 69%. All the 500 epochs along with the accuracies, loss values are listed in part 2 of the Jupyter notebook.

Runtime for training the model is around 47 minutes 22 seconds as shown in the figure above.

Introduction to Machine Learning (Spring 2020)
Programming Assignment 2 – Group 41
Report

Vaibhav Chhajed (vchhajed)
Yeshwanth Badineni (ybadineni)
Hemant Koti (hemantko)

2. Compare the performance when the number of hidden layers is increased to 3 and 5.

- i. Model summary for 28 x 28 image size with 500 epochs and 3 hidden layers are listed below.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 256)	200960
dense_2 (Dense)	(None, 64)	16448
dense_3 (Dense)	(None, 32)	2080
dense_4 (Dense)	(None, 10)	330
Total params: 219,818		
Trainable params: 219,818		
Non-trainable params: 0		

```
Epoch 1/500
100000/100000 [=====] - 8s 84us/step - loss: 2.1831 - accuracy: 0.3665
Epoch 2/500
100000/100000 [=====] - 8s 78us/step - loss: 1.4936 - accuracy: 0.4839
Epoch 3/500
100000/100000 [=====] - 8s 81us/step - loss: 1.3660 - accuracy: 0.5339
Epoch 4/500
100000/100000 [=====] - 7s 73us/step - loss: 1.2877 - accuracy: 0.5610
Epoch 5/500
100000/100000 [=====] - 8s 77us/step - loss: 1.1979 - accuracy: 0.5901
Epoch 497/500
100000/100000 [=====] - 5s 47us/step - loss: 1.3942 - accuracy: 0.5261
Epoch 498/500
100000/100000 [=====] - 5s 50us/step - loss: 1.3223 - accuracy: 0.5196
Epoch 499/500
100000/100000 [=====] - 5s 52us/step - loss: 1.3587 - accuracy: 0.5239
Epoch 500/500
100000/100000 [=====] - 5s 52us/step - loss: 1.2943 - accuracy: 0.5255
Epoch 500/500
100000/100000 [=====] - 5s 52us/step - loss: 1.3851 - accuracy: 0.5263
Wall time: 44min 34s
```

Introduction to Machine Learning (Spring 2020)
Programming Assignment 2 – Group 41
Report

Vaibhav Chhajed (vchhajed)
Yeshwanth Badineni (ybadineni)
Hemant Koti (hemantko)

```
"""
Testing accuracy
"""
predict_test = model.predict_classes(data_test)
acc_test = np.where(label_test1 == predict_test)[0].shape[0]/data_test.shape[0]
print('Testing accuracy {}'.format(acc_test))

Testing accuracy 0.4288
```

Training accuracy for the given model comes around 45% to 55% as shown in the epochs figure above. Testing accuracy for the given model comes around 43%. Runtime for training the model is around *44 minutes 34 seconds* as shown in the figure above.

- ii. Model summary for 28 x 28 image size with 500 epochs and 5 hidden layers are listed below.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
=====		
dense_1 (Dense)	(None, 256)	200960
dense_2 (Dense)	(None, 128)	32896
dense_3 (Dense)	(None, 64)	8256
dense_4 (Dense)	(None, 32)	2080
dense_5 (Dense)	(None, 16)	528
dense_6 (Dense)	(None, 10)	170
=====		
Total params: 244,890		
Trainable params: 244,890		
Non-trainable params: 0		

```
Epoch 1/500
100000/100000 [=====] - 6s 55us/step - loss: 1.7857 - accuracy: 0.3662
Epoch 2/500
100000/100000 [=====] - 5s 54us/step - loss: 1.3902 - accuracy: 0.4831
Epoch 3/500
100000/100000 [=====] - 5s 54us/step - loss: 1.3496 - accuracy: 0.5013
Epoch 4/500
100000/100000 [=====] - 5s 55us/step - loss: 1.4021 - accuracy: 0.5012
Epoch 5/500
100000/100000 [=====] - 5s 55us/step - loss: 1.4594 - accuracy: 0.4896
```

Introduction to Machine Learning (Spring 2020)
Programming Assignment 2 – Group 41
Report

Vaibhav Chhajed (vchhajed)
Yeshwanth Badineni (ybadineni)
Hemant Koti (hemantko)

```
100000/100000 [=====] - 9s 87us/step - loss: 2.3027 - accuracy: 0.1007
Epoch 497/500
100000/100000 [=====] - 9s 90us/step - loss: 2.3028 - accuracy: 0.1005
Epoch 498/500
100000/100000 [=====] - 9s 93us/step - loss: 2.3028 - accuracy: 0.0997
Epoch 499/500
100000/100000 [=====] - 10s 104us/step - loss: 2.3028 - accuracy: 0.0985
Epoch 500/500
100000/100000 [=====] - 11s 107us/step - loss: 2.3028 - accuracy: 0.0983
Wall time: 50min 04s
```

```
"""
Testing accuracy
"""
predict_test = model.predict_classes(data_test)
acc_test = np.where(label_test1 == predict_test)[0].shape[0]/data_test.shape[0]
print('Testing accuracy {}'.format(acc_test))
```

Testing accuracy 0.1

Training accuracy for the given model comes around 9% to 10% as shown in the epochs figure above. Testing accuracy for the given model comes around 10%. Runtime for training the model is around 50 minutes 04 seconds as shown in the figure above.

Comparison between 3 and 5 hidden layers: Usually, the performance of a neural network with a higher number of hidden gives accuracy. But here we can observe that the performance of the model with 3 hidden is higher than 5 hidden layers. This is because there is not enough training data on which the model with 5 hidden layers can generalize, which results in the underfitting of the model with lower accuracy.

3. Use the `resize images()` function to reduce the resolution of the images to (20 x 20), (15 x 15), (10 x 10) and (5 x 5). Using the 2 hidden layer architecture, compare the performance at different resolutions, including the original (28 x 28) resolution, both in terms of test accuracy and time.

- i. Model summary for 28 x 28 image size with 500 epochs and 2 hidden layers are listed below.

Introduction to Machine Learning (Spring 2020)
Programming Assignment 2 – Group 41
Report

Vaibhav Chhajed (vchhajed)
Yeshwanth Badineni (ybadineni)
Hemant Koti (hemantko)

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 256)	200960
dense_2 (Dense)	(None, 32)	8224
dense_3 (Dense)	(None, 10)	330
Total params: 209,514		
Trainable params: 209,514		
Non-trainable params: 0		

```
Epoch 1/500
100000/100000 [=====] - 8s 82us/step - loss: 3.2113 - accuracy: 0.6147
Epoch 2/500
100000/100000 [=====] - 8s 85us/step - loss: 1.4194 - accuracy: 0.6948
Epoch 3/500
100000/100000 [=====] - 8s 85us/step - loss: 1.3944 - accuracy: 0.7077
Epoch 4/500
100000/100000 [=====] - 8s 83us/step - loss: 1.3804 - accuracy: 0.7148
Epoch 5/500
100000/100000 [=====] - 9s 87us/step - loss: 1.3802 - accuracy: 0.7207
Epoch 497/500
100000/100000 [=====] - 5s 53us/step - loss: 0.7294 - accuracy: 0.7989
Epoch 498/500
100000/100000 [=====] - 5s 53us/step - loss: 0.7674 - accuracy: 0.7985
Epoch 499/500
100000/100000 [=====] - 5s 49us/step - loss: 0.7316 - accuracy: 0.7987
Epoch 500/500
100000/100000 [=====] - 5s 46us/step - loss: 0.7454 - accuracy: 0.7986
Wall time: 47min 42s
```

```
Report 4 - Question 3 - 28 x 28 image with two hidden layers - Testing accuracy

predict_test = model.predict_classes(data_test)
acc_test = np.where(label_test1 == predict_test)[0].shape[0]/data_test.shape[0]
print('Testing accuracy {}'.format(acc_test))
```

Testing accuracy 0.68376

Training accuracy for the given model comes around 70% to 79% as shown in the epochs figure above. Testing accuracy for the given model comes around 69%. All the 500 epochs along with the accuracies, loss values are listed in part 2 of the Jupyter notebook.

Runtime for training the model is around 47 minutes 22 seconds as shown in the figure above.

Introduction to Machine Learning (Spring 2020)
Programming Assignment 2 – Group 41
Report

Vaibhav Chhajed (vchhajed)
Yeshwanth Badineni (ybadineni)
Hemant Koti (hemantko)

- ii. Model summary for 20 x 20 image size with 500 epochs and 2 hidden layers are listed below.

Model: "sequential_2"

Layer (type)	Output Shape	Param #
dense_4 (Dense)	(None, 256)	102656
dense_5 (Dense)	(None, 32)	8224
dense_6 (Dense)	(None, 10)	330
Total params: 111,210		
Trainable params: 111,210		
Non-trainable params: 0		

```
Epoch 1/500
100000/100000 [=====] - 5s 51us/step - loss: 10.5356 - accuracy: 0.3426
Epoch 2/500
100000/100000 [=====] - 5s 49us/step - loss: 10.2524 - accuracy: 0.3624
Epoch 3/500
100000/100000 [=====] - 5s 46us/step - loss: 10.2044 - accuracy: 0.3658
Epoch 4/500
100000/100000 [=====] - 5s 50us/step - loss: 10.1748 - accuracy: 0.3678
Epoch 5/500
100000/100000 [=====] - 5s 49us/step - loss: 10.1604 - accuracy: 0.3687
.
.
.
100000/100000 [=====] - 5s 46us/step - loss: 4.8865 - accuracy: 0.6967
Epoch 497/500
100000/100000 [=====] - 5s 46us/step - loss: 4.8984 - accuracy: 0.6959
Epoch 498/500
100000/100000 [=====] - 5s 46us/step - loss: 4.8874 - accuracy: 0.6966
Epoch 499/500
100000/100000 [=====] - 4s 44us/step - loss: 4.9329 - accuracy: 0.6937
Epoch 500/500
100000/100000 [=====] - 5s 46us/step - loss: 4.8963 - accuracy: 0.6959
Wall time: 51min 54s
```

```
"""
Testing accuracy
"""
predict_test = model.predict_classes(data_test)
acc_test = np.where(label_test1 == predict_test)[0].shape[0]/data_test.shape[0]
print('Testing accuracy {}'.format(acc_test))
```

Testing accuracy 0.67284

Introduction to Machine Learning (Spring 2020)
Programming Assignment 2 – Group 41
Report

Vaibhav Chhajed (vchhajed)
Yeshwanth Badineni (ybadineni)
Hemant Koti (hemantko)

Training accuracy for the given model comes around 65% to 70% as shown in the epochs figure above. Testing accuracy for the given model comes around 67%. Runtime for training the model is around *51 minutes 54 seconds* as shown in the figure above.

- iii. Model summary for 15 x 15 image size with 500 epochs and 2 hidden layers are listed below.

Training accuracy for the given model comes around 57% to 62% as shown in the epochs figure below. Testing accuracy for the given model comes around 60%. Runtime for training the model is around *41 minutes 24 seconds* as shown in the figure above.

Model: "sequential_3"

Layer (type)	Output Shape	Param #
=====	=====	=====
dense_7 (Dense)	(None, 256)	57856
dense_8 (Dense)	(None, 32)	8224
dense_9 (Dense)	(None, 10)	330
=====	=====	=====
Total params: 66,410		
Trainable params: 66,410		
Non-trainable params: 0		

```
Epoch 1/500
100000/100000 [=====] - 4s 42us/step - loss: 9.8873 - accuracy: 0.3824
Epoch 2/500
100000/100000 [=====] - 4s 41us/step - loss: 8.4520 - accuracy: 0.4719
Epoch 3/500
100000/100000 [=====] - 4s 44us/step - loss: 8.0744 - accuracy: 0.4958
Epoch 4/500
100000/100000 [=====] - 4s 42us/step - loss: 7.8797 - accuracy: 0.5087
Epoch 5/500
100000/100000 [=====] - 4s 41us/step - loss: 7.6945 - accuracy: 0.5207

100000/100000 [=====] - 4s 42us/step - loss: 6.2845 - accuracy: 0.6099
Epoch 497/500
100000/100000 [=====] - 4s 41us/step - loss: 6.2539 - accuracy: 0.6118
Epoch 498/500
100000/100000 [=====] - 4s 41us/step - loss: 6.2597 - accuracy: 0.6115
Epoch 499/500
100000/100000 [=====] - 4s 41us/step - loss: 6.2764 - accuracy: 0.6105
Epoch 500/500
100000/100000 [=====] - 4s 41us/step - loss: 6.2559 - accuracy: 0.6117
Wall time: 41min 24s
```

Introduction to Machine Learning (Spring 2020)
Programming Assignment 2 – Group 41
Report

Vaibhav Chhajed (vchhajed)
Yeshwanth Badineni (ybadineni)
Hemant Koti (hemantko)

```
"""
Testing accuracy
"""

predict_test = model.predict_classes(data_test)
acc_test = np.where(label_test1 == predict_test)[0].shape[0]/data_test.shape[0]
print('Testing accuracy {}'.format(acc_test))

Testing accuracy 0.5968
```

- iv. Model summary for 10 x 10 image size with 500 epochs and 2 hidden layers are listed below.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 100)	10100
dense_2 (Dense)	(None, 50)	5050
dense_3 (Dense)	(None, 10)	510
Total params: 15,660		
Trainable params: 15,660		
Non-trainable params: 0		

```
Epoch 1/500
100000/100000 [=====] - 5s 48us/step - loss: 1.6686 - accuracy: 0.5896
Epoch 2/500
100000/100000 [=====] - 4s 36us/step - loss: 1.0646 - accuracy: 0.6861
Epoch 3/500
100000/100000 [=====] - 4s 37us/step - loss: 1.0417 - accuracy: 0.7003
Epoch 4/500
100000/100000 [=====] - 4s 40us/step - loss: 1.0604 - accuracy: 0.7071
Epoch 5/500
100000/100000 [=====] - 4s 39us/step - loss: 1.0672 - accuracy: 0.7109

100000/100000 [=====] - 4s 41us/step - loss: 1.1487 - accuracy: 0.7136
Epoch 497/500
100000/100000 [=====] - 4s 43us/step - loss: 1.1315 - accuracy: 0.7129
Epoch 498/500
100000/100000 [=====] - 4s 43us/step - loss: 1.1206 - accuracy: 0.7125
Epoch 499/500
100000/100000 [=====] - 4s 41us/step - loss: 1.1353 - accuracy: 0.7128
Epoch 500/500
100000/100000 [=====] - 4s 43us/step - loss: 1.1392 - accuracy: 0.7122
Wall time: 49min 38s
```

Introduction to Machine Learning (Spring 2020)
Programming Assignment 2 – Group 41
Report

Vaibhav Chhajed (vchhajed)
Yeshwanth Badineni (ybadineni)
Hemant Koti (hemantko)

```
Testing accuracy
predict_test = model.predict_classes(data_test)
acc_test = np.where(label_test1 == predict_test)[0].shape[0]/data_test.shape[0]
print('Testing accuracy {}'.format(acc_test))

Testing accuracy 0.72948
```

Training accuracy for the given model comes around 71% to 76% as shown in the epochs figure above. Testing accuracy for the given model comes around 73%. Runtime for training the model is around 49 minutes 38 seconds as shown in the figure above.

Introduction to Machine Learning (Spring 2020)
Programming Assignment 2 – Group 41
Report

Vaibhav Chhajed (vchhajed)
Yeshwanth Badineni (ybadineni)
Hemant Koti (hemantko)

- v. Model summary for 5 x 5 image size with 500 epochs and 2 hidden layers are listed below.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 256)	6656
dense_2 (Dense)	(None, 64)	16448
dense_3 (Dense)	(None, 10)	650
Total params: 23,754		
Trainable params: 23,754		
Non-trainable params: 0		

```
Epoch 1/500
100000/100000 [=====] - 6s 61us/step - loss: 1.1103 - accuracy: 0.7017
Epoch 2/500
100000/100000 [=====] - 6s 58us/step - loss: 0.8202 - accuracy: 0.7631
Epoch 3/500
100000/100000 [=====] - 6s 57us/step - loss: 0.7949 - accuracy: 0.7729
Epoch 4/500
100000/100000 [=====] - 6s 57us/step - loss: 0.7825 - accuracy: 0.7760
Epoch 5/500
100000/100000 [=====] - 6s 60us/step - loss: 0.7693 - accuracy: 0.7812

Epoch 461/500
100000/100000 [=====] - 11s 108us/step - loss: 1.1085 - accuracy: 0.6954
Epoch 462/500
100000/100000 [=====] - 9s 94us/step - loss: 1.1057 - accuracy: 0.6938
Epoch 463/500
100000/100000 [=====] - 10s 95us/step - loss: 1.1052 - accuracy: 0.6945
Epoch 464/500
100000/100000 [=====] - 11s 114us/step - loss: 1.1073 - accuracy: 0.6944
Epoch 465/500
2048/100000 [.....] - ETA: 10s - loss: 1.0719 - accuracy: 0.7017
Wall time: 57min 11s
```

Introduction to Machine Learning (Spring 2020)
Programming Assignment 2 – Group 41
Report

Vaibhav Chhajed (vchhajed)
Yeshwanth Badineni (ybadineni)
Hemant Koti (hemantko)

```
"""
Testing accuracy
"""

predict_test = model.predict_classes(data_test)
acc_test = np.where(label_test1 == predict_test)[0].shape[0]/data_test.shape[0]
print('Testing accuracy {}'.format(acc_test))

Testing accuracy 0.6418
```

Training accuracy for the given model comes around *63% to 69%* as shown in the epochs figure above. Testing accuracy for the given model comes around *64%*. Runtime for training the model is around *57 minutes 11 seconds* as shown in the figure above.

Comparison between different image resolutions: We can observe that there is no much difference between the image resolutions (28 x 28), (20 x 20), (15 x 15) size images. This is because the model doesn't change much with minimal resolution changes. But as we further decrease the resolution of the images, we see a steep decrease in performance because the image properties fade away with a decrease in size.