COMP4107

Neural Networks - Assignment 1

Group Members: Krystian Wojcicki (101001444), Michael Kuang (101000485)

Date: October 7, 2018

Question 1

Running q1.py, we obtain:

```
(tensorflow) C:\Users\mkuang\Documents\Repos\COMP4107\Assignment1>python q1.py
prediction is: 5.35
Alice2d: [-0.64418534 -0.30125475]
Alicia2d: [-0.3593326  -0.36767659]
Bob2d: [-0.56750746 -0.08799758]
Mary2d: [-0.4428526  -0.56862492]
Sue2d: [-0.59388293  0.73057242]
Bob is closest to Alice in 2d
Alice2d: [-0.64418534 -0.30125475  0.63751178  0.96174681]
Alicia2d: [-0.3593326  -0.36767659  0.29605046 -0.80501437]
Bob2d: [-0.56750746 -0.08799758  0.62845599  0.52462823]
Mary2d: [-0.4428526  -0.56862492 -0.65902357  0.21502376]
Sue2d: [-0.59388293  0.73057242 -0.28824492 -0.17459057]
Bob is closest to Alice in 4d
```

We can replicate the slide entitled “Modelling “Person” – SVD” and obtained a prediction value of 5.35 (or 1.35 if the row average is not added) as expected. Projecting into a 2D space, we see that Bob is closest to Alice, and similarly Bob is closest to Alice in 4D.

Question 2

Running q2.py, we obtain the SVD values for u, s, v as:

```
(tensorflow) C:\Users\mkuang\Documents\Repos\COMP4107\Assignment1>python q2.py
u:  [[-0.33306893  0.73220483  0.37614814  0.45984101]
 [-0.48640367  0.34110504 -0.754835   -0.2779981 ]
 [-0.79307315 -0.44109455  0.37868687 -0.18184292]
 [-0.15333474 -0.39109979 -0.38122559  0.82352684]]
s:  [[1.10528306e+01 0.00000000e+00 0.00000000e+00]
 [0.00000000e+00 9.13748280e-01 0.00000000e+00]
 [0.00000000e+00 0.00000000e+00 1.30538231e-16]]
v:  [[-0.41903326 -0.56492763 -0.71082199]
 [-0.81101447 -0.11912225  0.57276996]
 [-0.40824829  0.81649658 -0.40824829]]
```

Question 3

Running q3.py, we see that $\left|\left|A - A_2\right|\right| = 1.33$.

Question 4

Running q4.py, we see that the least squares solution to $\min_{x \epsilon R^3} ||Ax - b||$ is [-0.14705882 0.05882353 0.26470588].

Running q4tables.py we get a table according to the question specifications:

```
(tensorflow) C:\Users\mkuang\Documents\Repos\COMP4107\Assignment1>python q4tables.py
Step Size       Iterations      x
0.010000        510     [-0.17866429  0.14271755  0.21666248]
0.050000        435     [inf inf inf]
0.100000        294     [-inf -inf -inf]
0.150000        249     [inf inf inf]
0.200000        225     [2.90474172e+307            inf             inf]
0.250000        210     [-inf -inf -inf]
0.500000        174     [-inf -inf -inf]
```

We see that there is a solution for when $\varepsilon = 0.01$ but no solutions for step size greater than that. This is because when the step size is at 0.05 or larger, the answer cannot converge as we are overshooting each time and thus will never reach a minimum.

Question 5

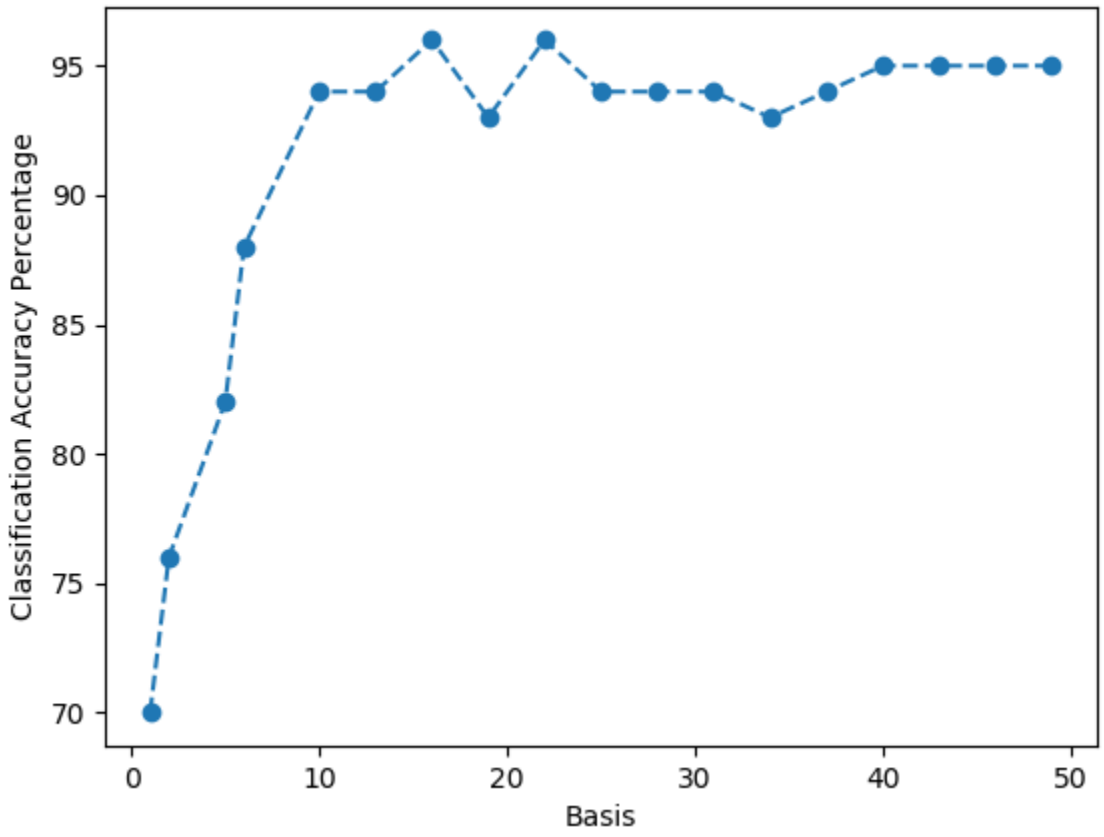
Running q5.py, we get 2 linearly independent vectors as described below:

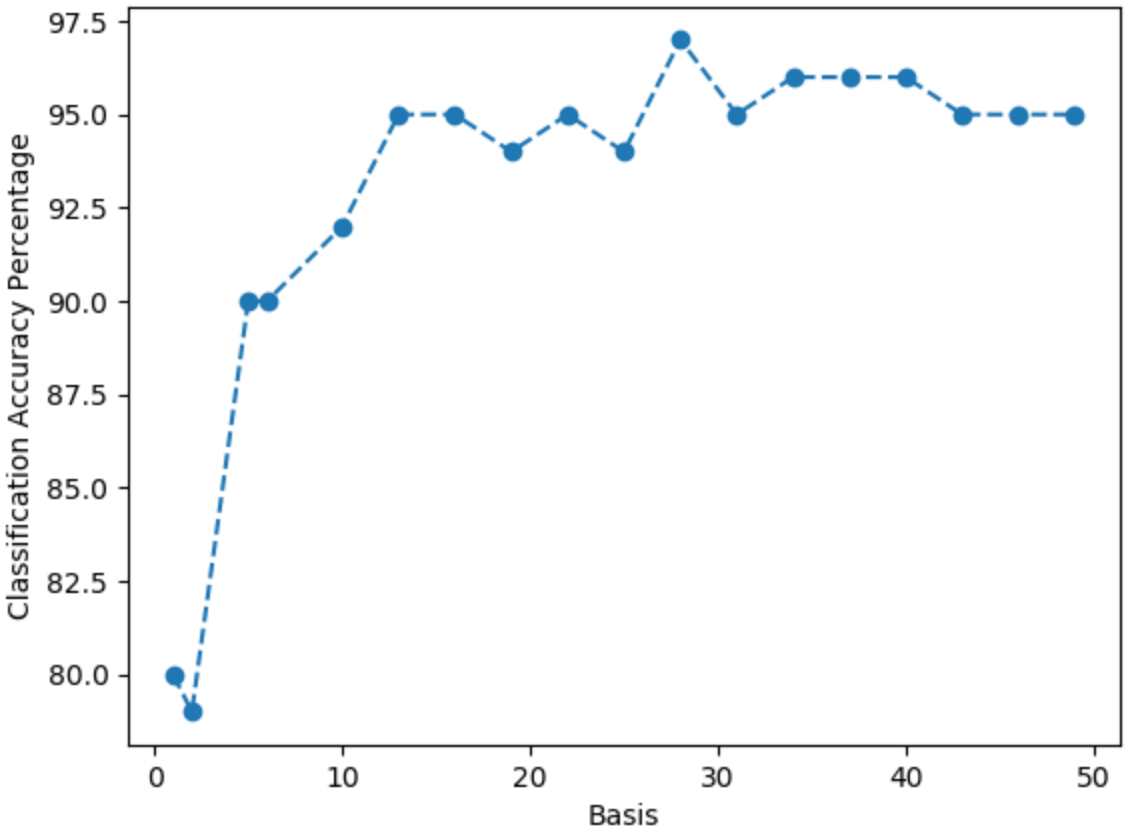
```
(tensorflow) C:\Users\mkuang\Documents\Repos\COMP4107\Assignment1>python q5.py
Two linearly independent vectors are:
Vector 1: [[-0.75878571]
 [ 0.59374776]
 [-0.02104018]
 [ 0.26695535]]
Vector 2: [[-0.33687873]
 [-0.65142615]
 [-0.50845005]
 [ 0.45125961]]
A has  4 columns of which 2  are linearly independent
A has  2  linearly independent rows
A inverse: [[ 0.06507304  0.01460823 -0.05046481]
 [ 0.03984064 -0.03187251 -0.07171315]
 [-0.00929615  0.14077025  0.1500664 ]
 [ 0.09561753  0.12350598  0.02788845]]
```

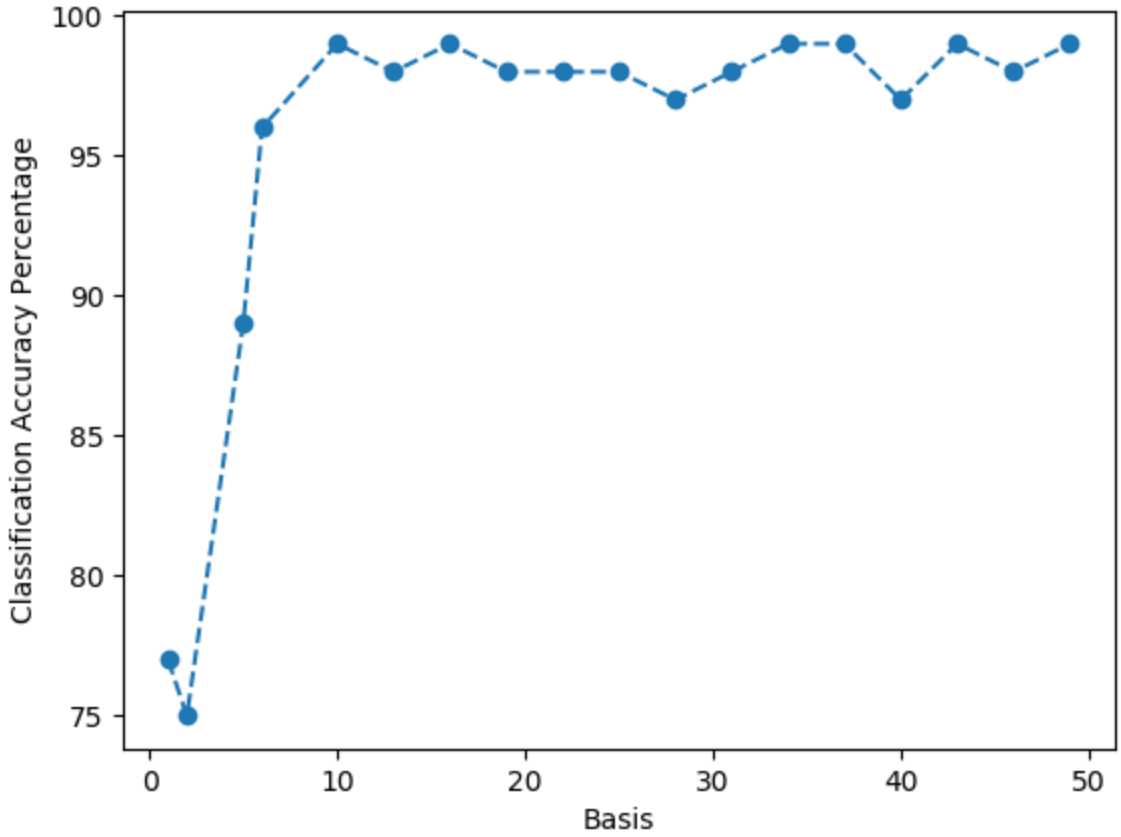
Since the columns of A only has 2 linearly independent columns, they are not linearly independent in R^3 because the rank is 2. Similarly, the rows of A only have 2 linearly independent rows, and therefore is not linearly independent in R^4. The inverse of A is as described above.

Question 6

Running q6.py with accordance to the specifications in question 6, we get a graph that closely resembles what is shown in figure 4 of the research paper where there is an exponential increase in accuracy then quickly plateaus after the first 10 basis: Each digits A matrix consisted of a randomly picked 500 images, and 100 unknown images (all of the digit 4) were used for accuracy testing.

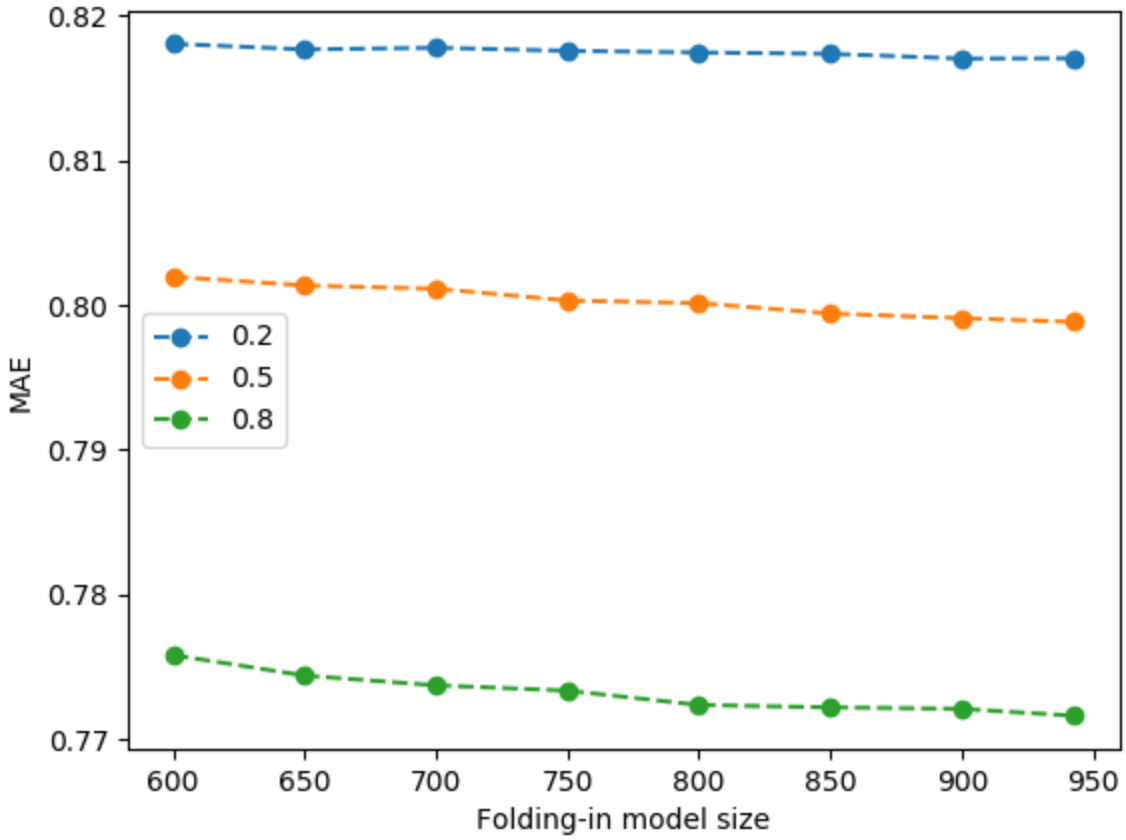


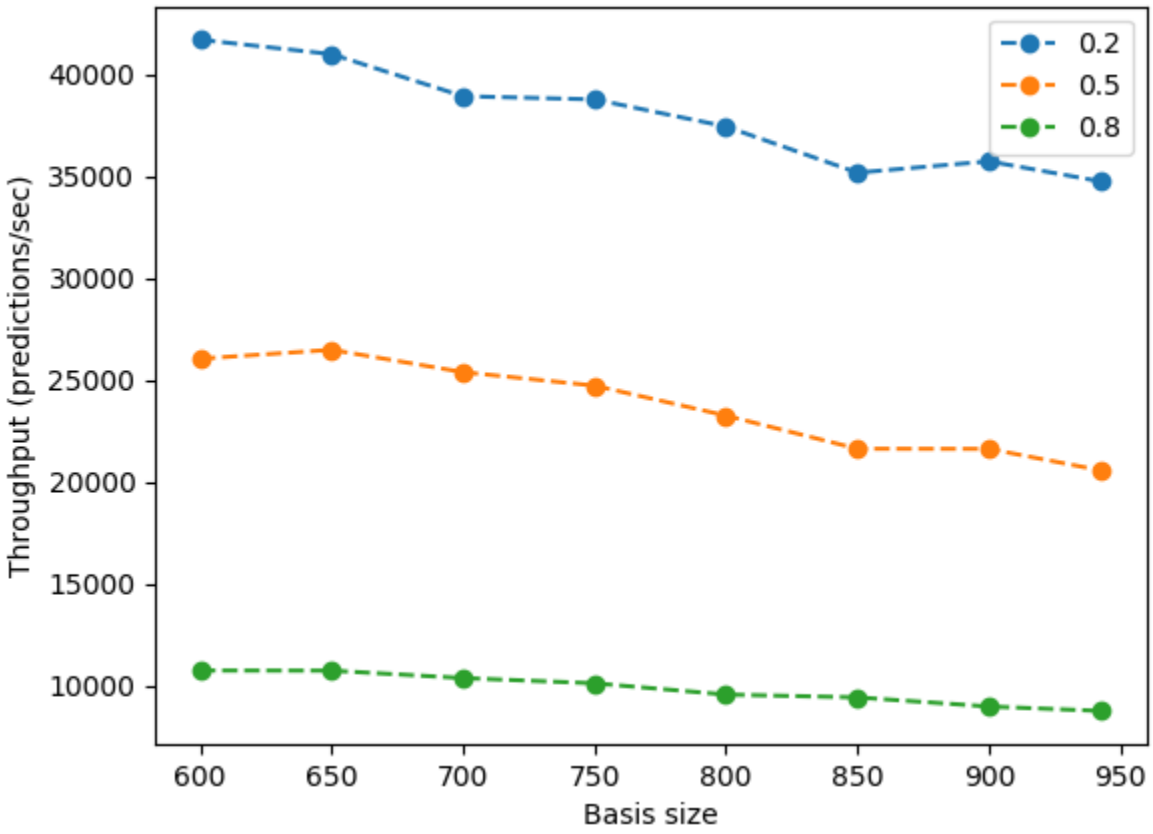

97.5
95.0
92.5
90.0
87.5
85.0
82.5
80.0
Classification Accuracy Percentage
0
10
20
30
40
50
Basis



Question 7

Following the specifications to the research paper, we were able to reproduce figure 3 and 4. Running q7.py will demonstrate the results:


0.82
0.81
0.80
0.79
0.78
0.77
MAE
0.2
0.5
0.8
600
650
700
750
800
850
900
950
Folding-in model size


0.2
0.5
0.8
40000
35000
30000
25000
20000
15000
10000
Throughput (predictions/sec)
600
650
700
750
800
850
900
950
Basis size