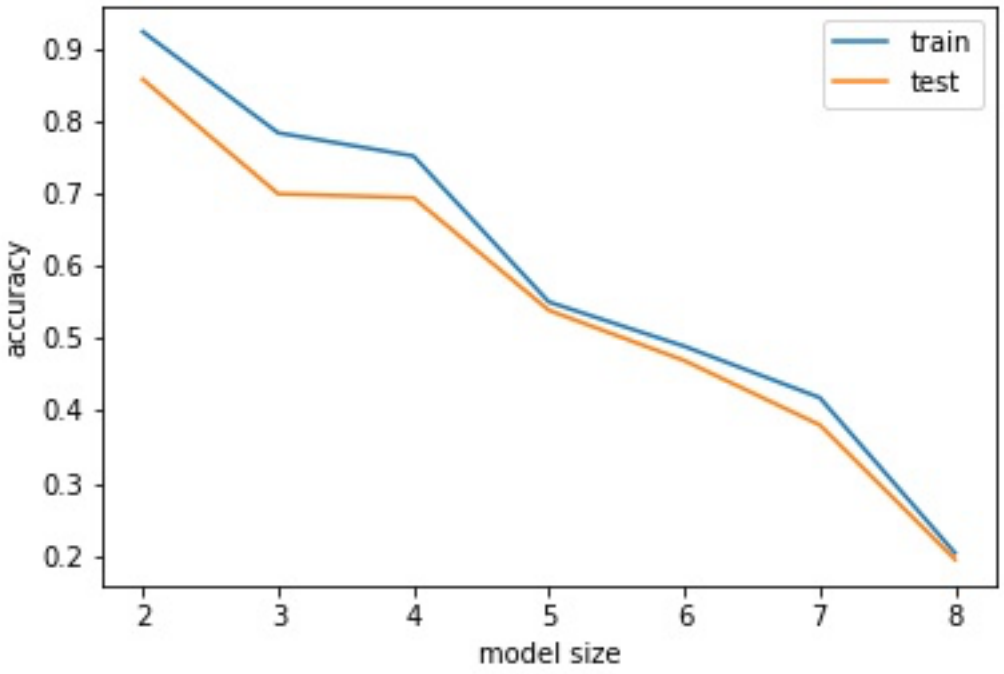
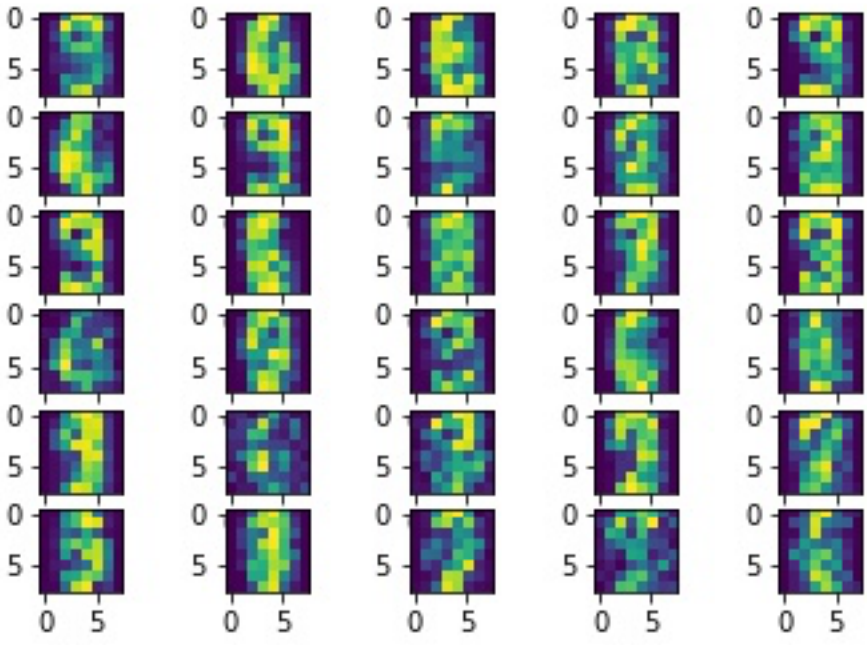
1. Question 1

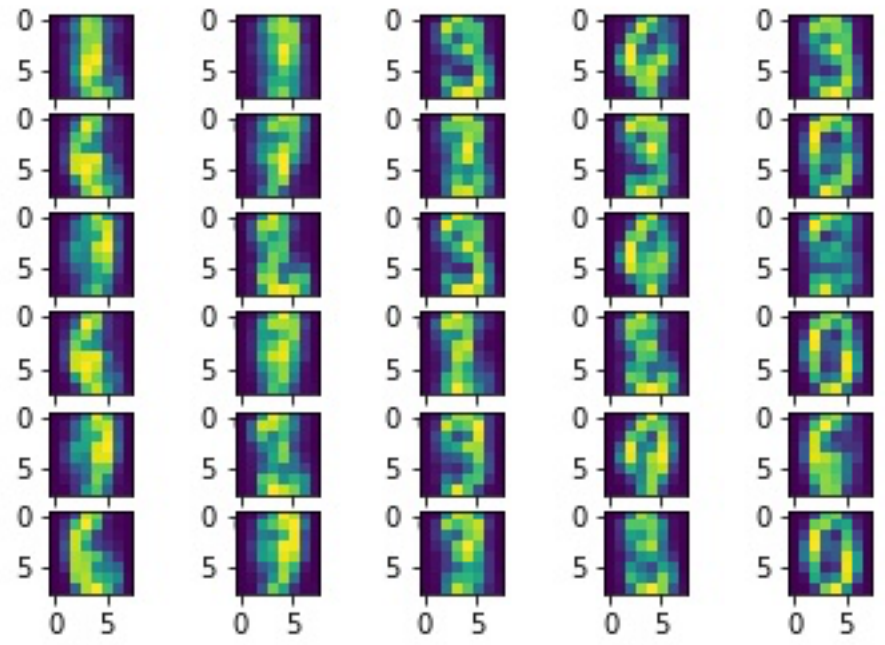
    a. The lower number of layers has the best training and testing accuracy. The more layers means the further distanced the later layers are from the actual data making the results noisier and noisier until 8 layers when the model is barely able to predict. Thus, the model with only 2 RBM layers is the best for both training and testing.



b. The gaussian produces results that are rather good in that the images can be reproduced with enough fidelity to often tell what number it is.



c. The non- Gaussian model performs very similarly to the non-gaussian. Some images are better reproductions that others but the non-Gaussian seems better at reproducing zeros.

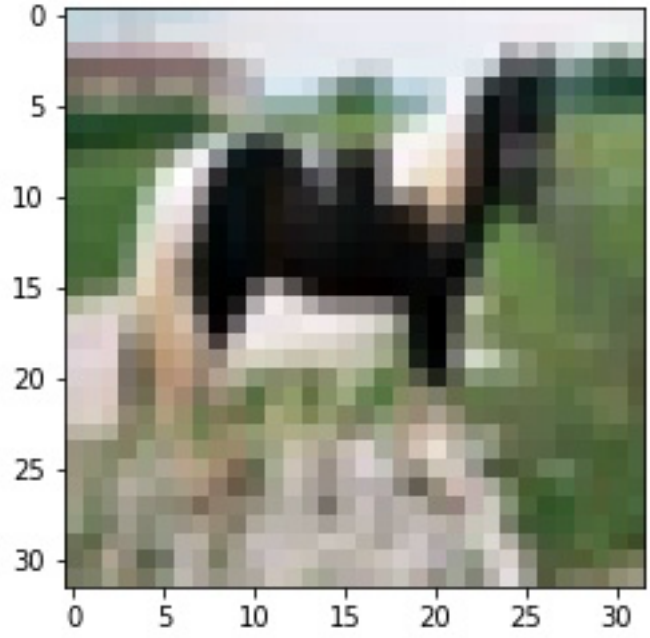


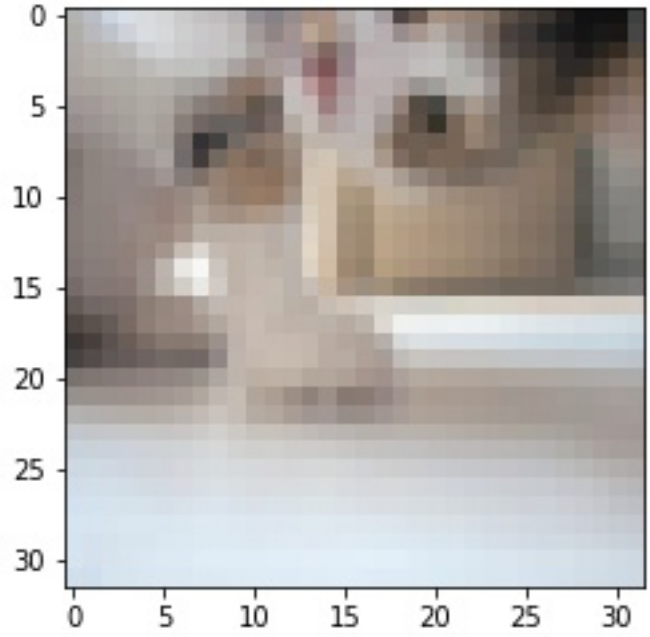
2. Convolutional NN.

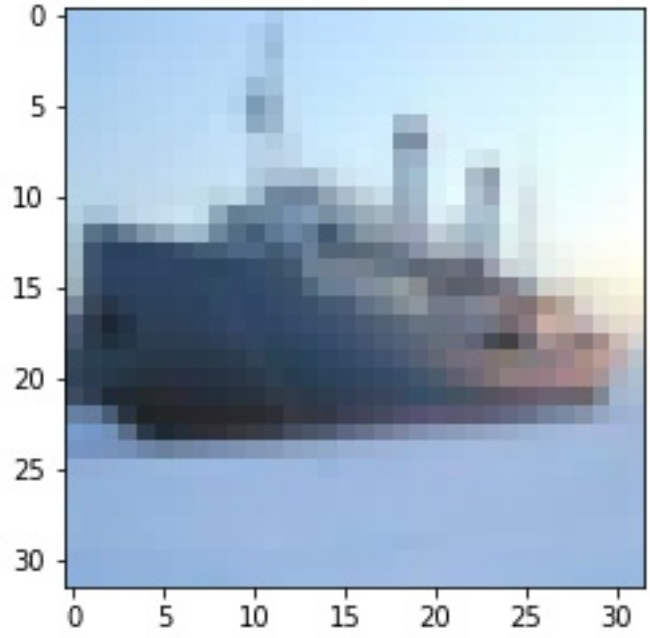
   a. Images selected by generating random indexes with a seed to be reproducible. The data is in red values, green values, blue values format which is unusual and requires some additional processing. This makes the format less than ideal for plugging in to a convolutional neural network, so after graphing the data are imported preformatted from Tensorflow.

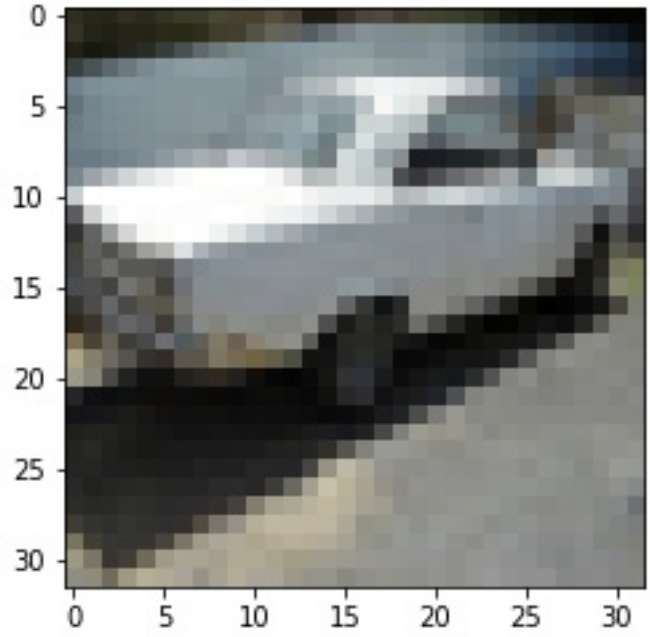
      i. Train indexes: 1746, 44103, 23887, 17865, 26390

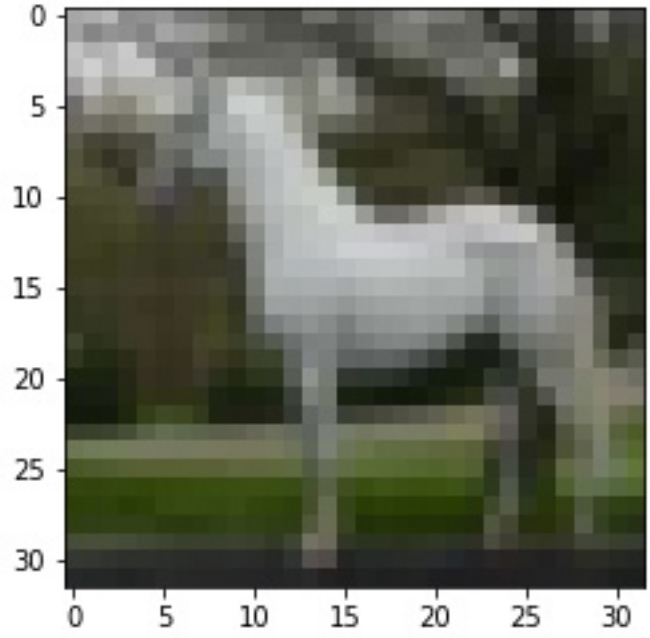
      ii. Test indexes: 1566, 1871, 495, 8550, 5330

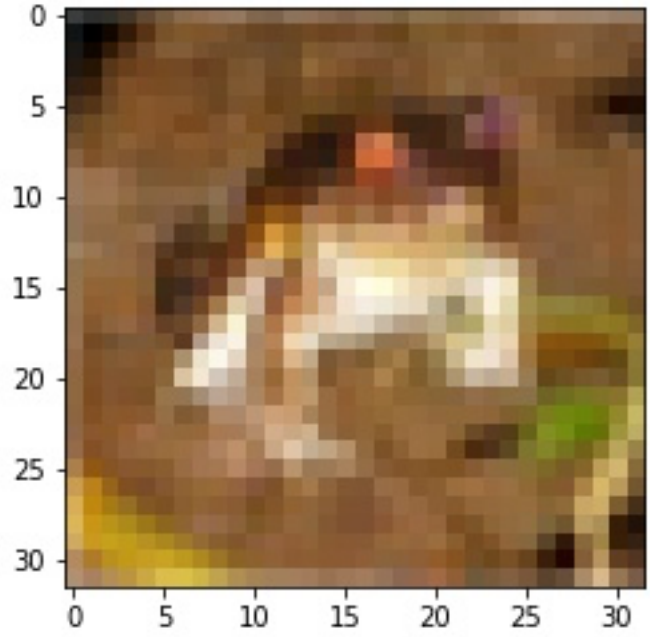

0
5
10
15
20
25
30
0
5
10
15
20
25
30

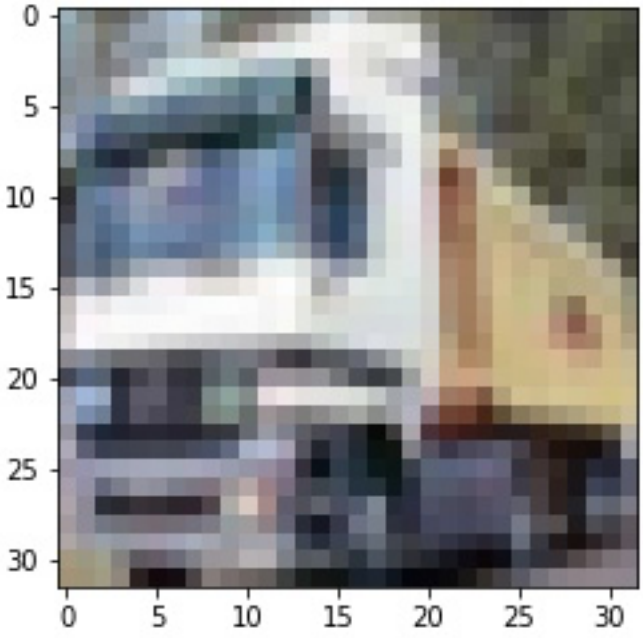

0
5
10
15
20
25
30
0
5
10
15
20
25
30

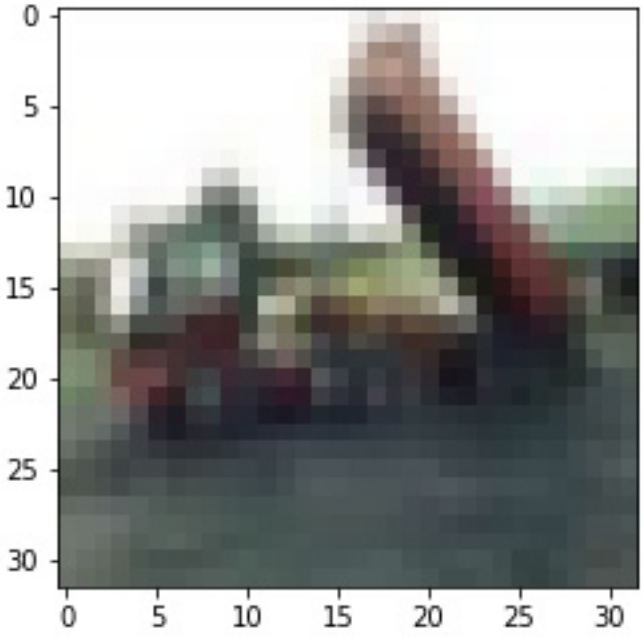

0
5
10
15
20
25
30
0
5
10
15
20
25
30

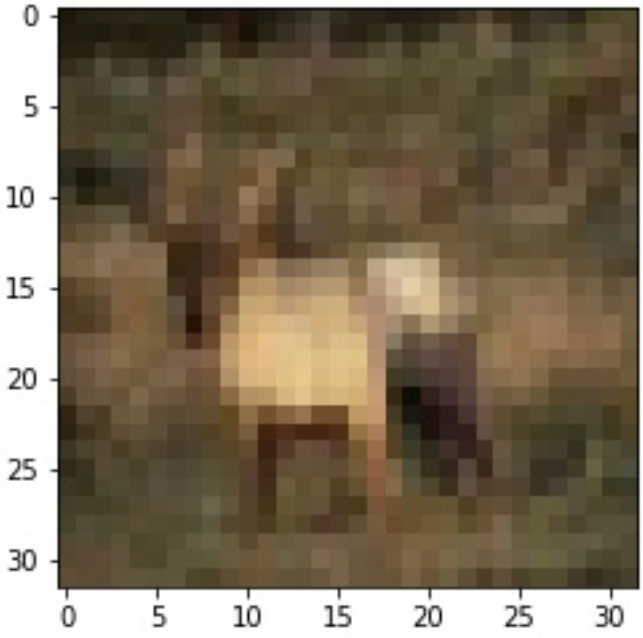

0
5
10
15
20
25
30
0
5
10
15
20
25
30

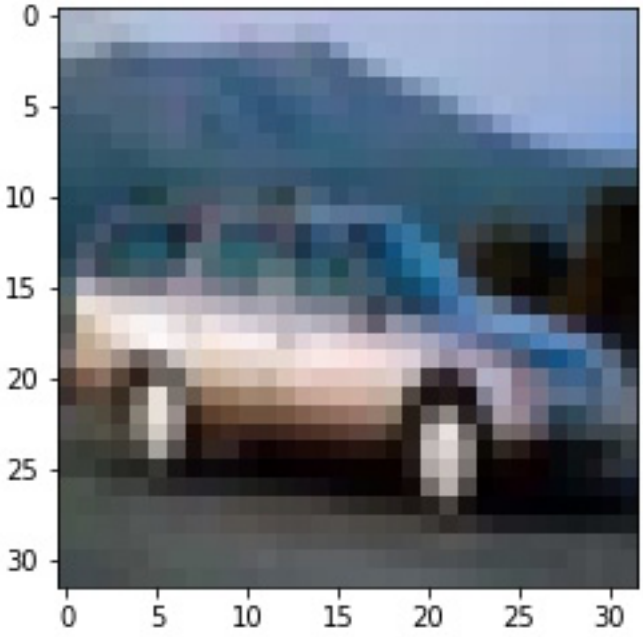

0
5
10
15
20
25
30
0
5
10
15
20
25
30


0
5
10
15
20
25
30
0
5
10
15
20
25
30


0
5
10
15
20
25
30
0
5
10
15
20
25
30


0
5
10
15
20
25
30
0
5
10
15
20
25
30

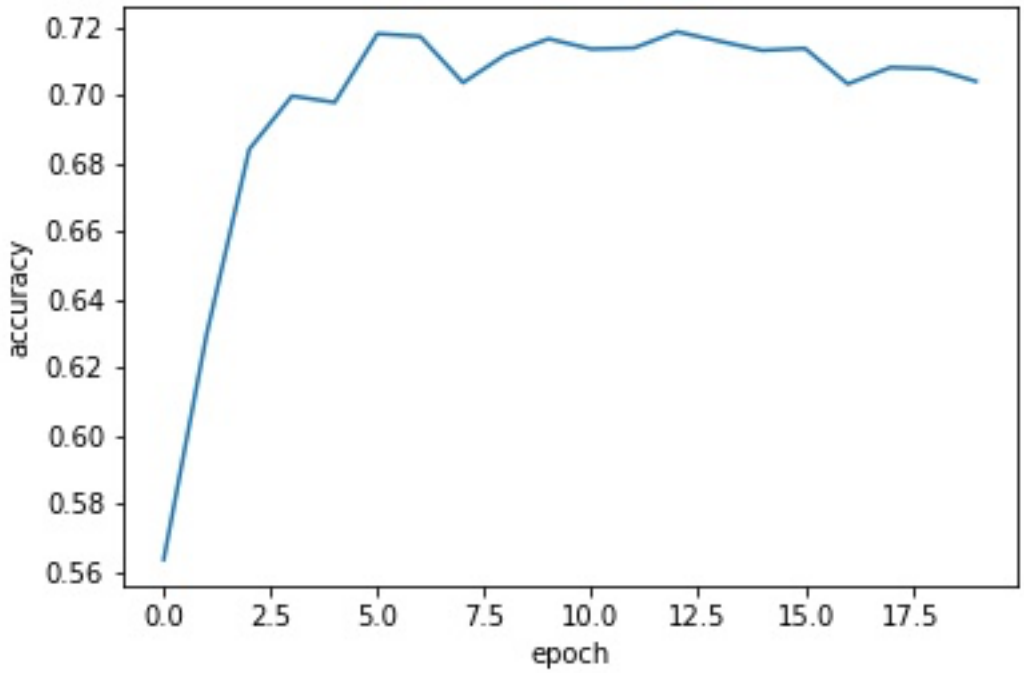

0
5
10
15
20
25
30
0
5
10
15
20
25
30

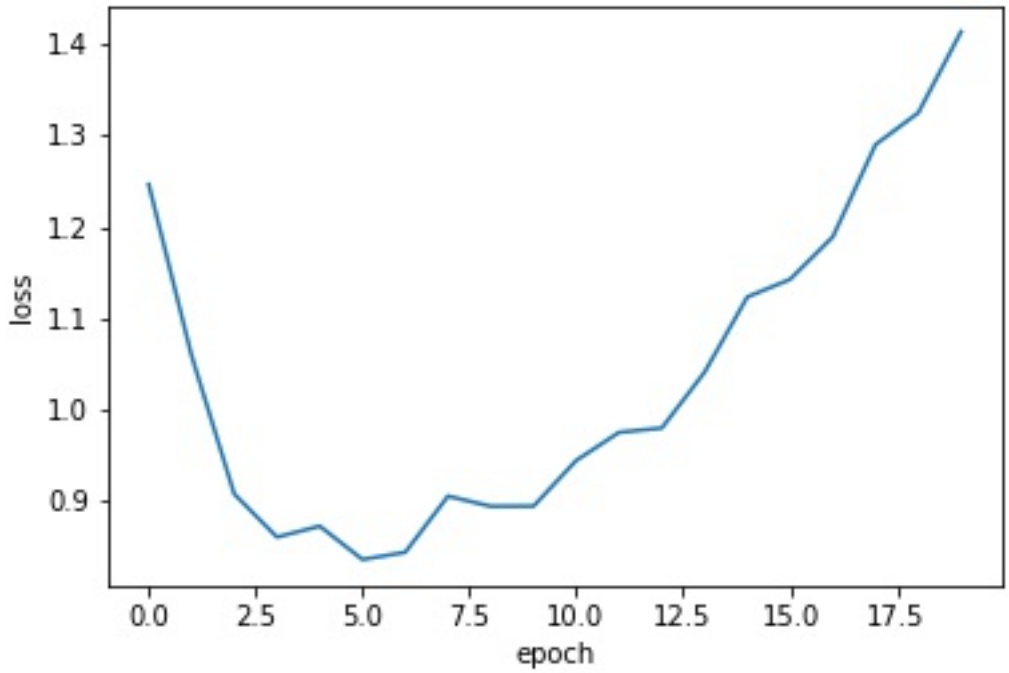

0
5
10
15
20
25
30
0
5
10
15
20
25
30

b. Results: L=2 is the best with Max Pooling. This is because more information is kept. Each pooling layer is approximating the data and forcing it to a smaller size. Thus, less compression means it is easier for the model to learn. All results are test accuracy.

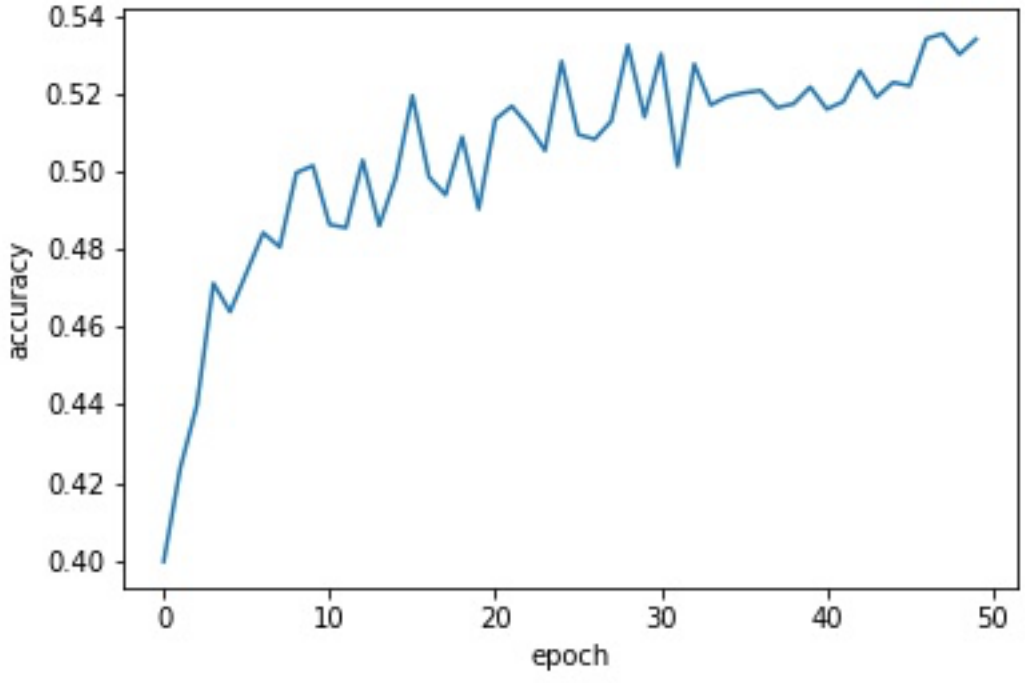
Accuracy	L = 2	L = 3	L = 4
Max Pooling	0.5578	0.4990	0.3594
Avg. Pooling	0.4901	0.4565	0.3126

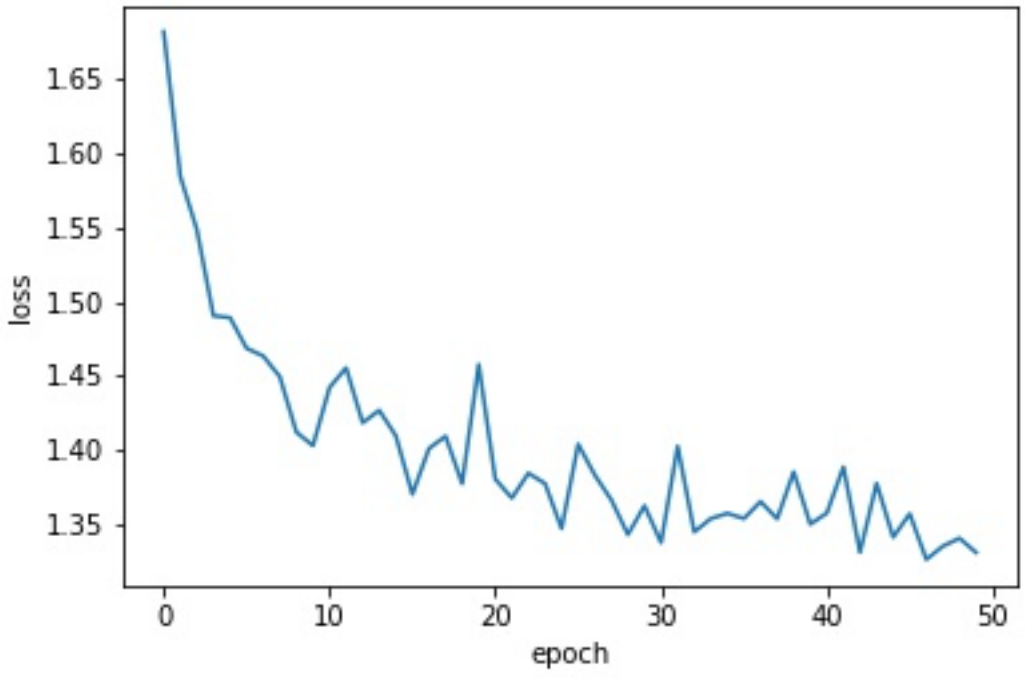
c. A very simple fix is to increase the number of epochs (iterations over the data during training) Using just 5 epochs and changing nothing else with L=2 and Max Pooling gives an accuracy of *0.6367*. Changing the number of nodes in each layer and not having a final pooling layer also helps. With 10 epochs, 32 (2,2), pooling, 64(2,2), pooling, 128 (2,2) and 10 epochs: accuracy = 0.7167. This changes the total number of parameters to be higher than before and increase the training time significantly compared to the original models. As can be seen from the graphs below it is better to stop training around epoch 10 to avoid overfitting.


0.72
0.70
0.68
0.66
0.64
0.62
0.60
0.58
0.56
accuracy
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
epoch


1.4
1.3
1.2
1.1
1.0
0.9
loss
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
epoch

d. Used a neural network classifier with the encoded images. The accuracy isn't great but considering there is less information there it isn't too bad





e. Confusion matrix

i. L1 confusion matrix:

```
[351,  67,  49,  24,  12,  52,  27,  52, 294,  72],
[ 27, 481,   5,  52,   1,  43,  34,  34, 118, 205],
[ 91,  23, 153,  41, 143, 210, 151, 128,  36,  24],
[ 45,  62,  51,  79,  30, 395, 105, 170,  17,  46],
[ 52,  24, 118,  65, 204, 177, 232,  96,  19,  13],
[ 39,  28,  50,  48,  37, 539,  55, 176,  15,  13],
[ 11,  49,  53,  65, 106, 128, 476,  78,  12,  22],
[ 55,  49,  42,  54,  29, 253,  44, 392,  17,  65],
[191, 143,  17,  26,   3,  59,  12,  14, 458,  77],
[ 64, 268,   4,  21,   2,  35,  36,  88,  86, 396]]
```

ii. L2 confusion matrix:

```
[[375,  52,  38,  34,  36,  29,  88,  11, 273,  64],
 [ 22, 563,   4,  30,   6,  21,  68,  10,  78, 198],
 [ 58,  19, 182, 109, 105, 154, 307,  13,  36,  17],
 [ 20,  22,  39, 303,  28, 225, 295,  13,  10,  45],
 [ 24,   8,  58,  70, 267,  82, 408,  29,  34,  20],
 [  8,  11,  32, 164,  44, 496, 182,  30,  17,  16],
 [  2,  14,  18,  47,  26,  32, 837,   3,   7,  14],
 [ 25,  31,  24,  90, 110, 173, 130, 318,  20,  79],
 [ 55,  68,  11,  27,  22,  30,  63,   2, 644,  78],
 [ 20, 162,   3,  40,   7,  31,  73,  13,  73, 578]]
```

iii. l2 confusion with regularization factor of 0.001

```
[471,  46,  38,   4,  76,   4,  46,  29, 205,  81],
[ 34, 578,   6,   1,  22,   2,  34,  23,  65, 235],
[ 78,  39, 199,  10, 285,  36, 220,  79,  32,  22],
[ 30,  49,  88,  70, 148, 105, 315, 102,  16,  77],
[ 35,  19,  58,   4, 510,   9, 243,  73,  22,  27],
[ 19,  19,  97,  46, 146, 232, 218, 171,  23,  29],
[  3,  22,  28,   5, 110,   4, 773,  25,  13,  17],
[ 28,  31,  28,  12, 166,  27,  85, 519,  16,  88],
[121,  75,  12,   4,  50,   7,  31,  12, 580, 108],
[ 35, 188,   5,   4,  16,   4,  52,  33,  57, 606]]
```

3. Question 3

   a. The correct approach to splitting the data does not shuffle the data, which is odd for machine learning but normal for time series problems. Shuffling the data causes the problem to be an interpolation problem rather than a time series problem. Thus, the data were split 80/20 for training/testing at the point of 80%

of the data. In other words, the first 80% of the data were selected for training and the last 20% were saved for testing/validation.

b. For time t+1 mse is not good. After much experimentation and going so far as to trying code straight from tensorflow's website, I cannot get a good result on the data given year, month, and day as input: MSE:17.103221893310547, MAE:3.40463924407959 Leaving out year, though, seems to work!! This makes the problem a little more like a seasonal prediction with multiple samples and doesn't account for the yearly trend at all. MSE:4.880945205688477, MAE:1.7468453645706177.

W/O Year	K=1	K=2	K=5	K=7	K=14	K=30	K=60
MSE	5.40	5.41	4.88	5.07	4.83	4.86	4.98
MAE	1.82	1.83	1.74	1.77	1.74	1.74	1.78

With Year	K=5	K=365	K=700
MSE	17.10	16.61	17.20
MAE	3.40	3.37	3.40

c. Gated recurrent unit is like LSTM but with a forgetting mechanism built into it. There is a vector that is multiplied by the previous estimate that determines how much information to keep from previous experiences. performance is very similar. The values are slightly lower for some values of "K" but slightly higher for others. Thus, no definite conclusion of the strength of one over the other can be made.

W/O year	K=2	K=5	K=14	K=60
MSE	4.98	4.88	5.01	4.92

MAE	1.76	1.74	1.77	1.76