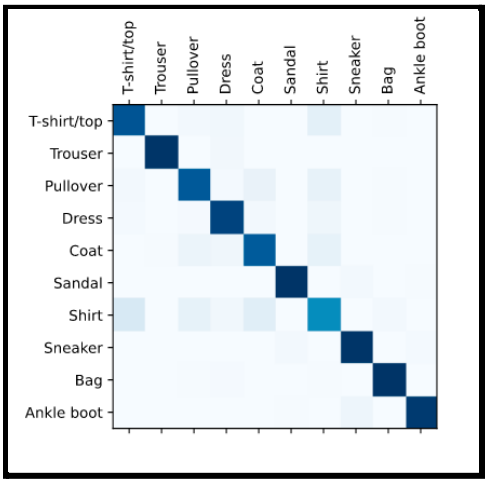
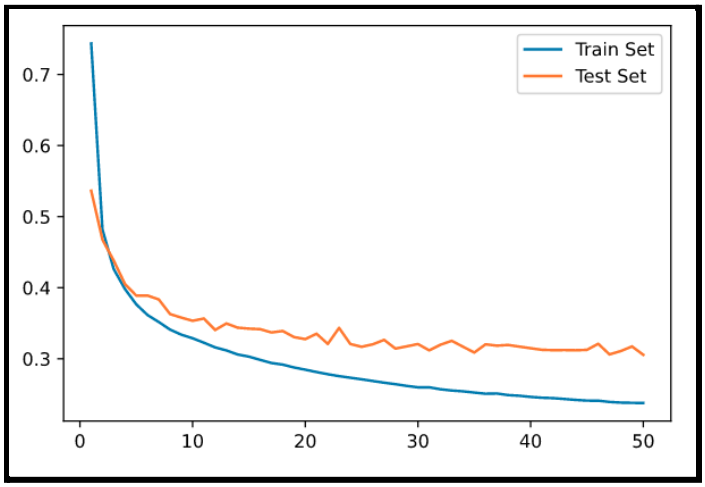
Assignment 2
Programming Questions

Answer 2:

Test Accuracy: 0.8921999931335449






```
[[857.   1.  18.  29.   2.   1.  85.   0.   7.   0.]
 [  1. 969.   0.  24.   3.   0.   1.   0.   2.   0.]
 [ 17.   1. 838.  13.  61.   0.  66.   0.   4.   0.]
 [ 13.   2.  10. 916.  18.   0.  36.   0.   5.   0.]
 [  2.   4.  52.  34. 834.   1.  70.   0.   2.   1.]
 [  0.   0.   0.   0.   0. 975.   1.  19.   1.   4.]
 [139.   2.  69.  29. 101.   0. 642.   0.  18.   0.]
 [  0.   0.   0.   0.   0.  18.   0. 970.   0.  12.]
 [  2.   2.   4.   8.   2.   1.   6.   3. 972.   0.]
 [  0.   0.   0.   0.   0.   6.   1.  44.   0. 949.]]
```

Answer 1:

Custom MLPClassifier

Architecture: [784, 256, 128, 64, 10]

batch_size: 64

weight_init: normal

n_layers: 5

epochs: 100

learning rate: 1e-4

Train Set in splitted into train/val set,

Train Set size(after split): 50,000

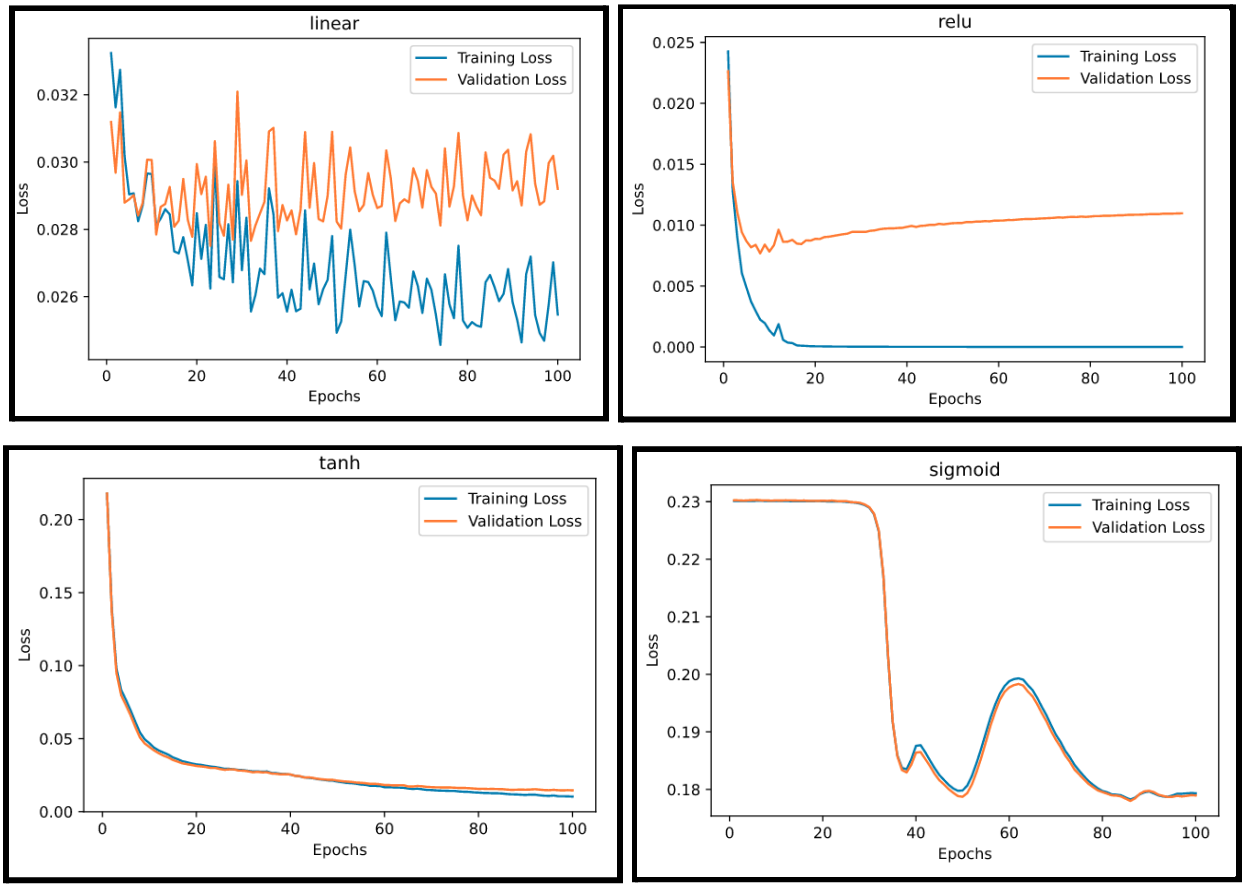
Val Set size(after split) : 10,000

Test Set size(after split): 10,000

Part 1: Model Saved

Part 2:

Loss Plots



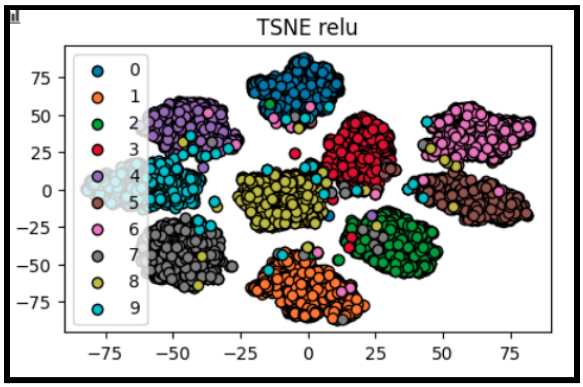
Part 3: For each case, the output activation function should be softmax, since it is a multiclass classification problem and output is expected to be probability. Therefore, softmax is the only acceptable choice for the output layer.

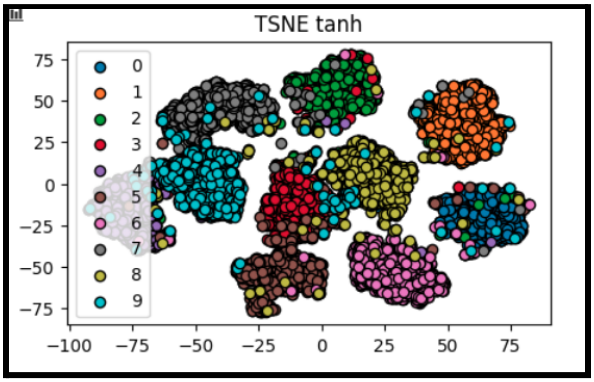
Part 4:

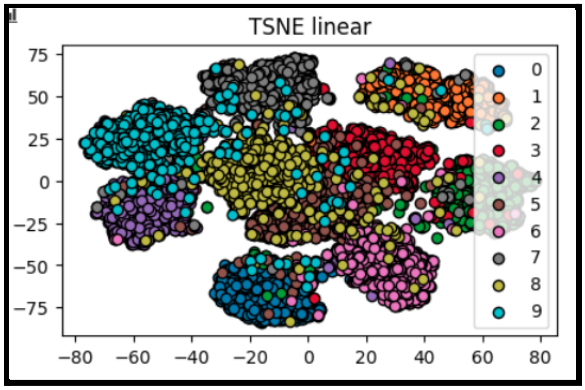
Total number of layers: 5

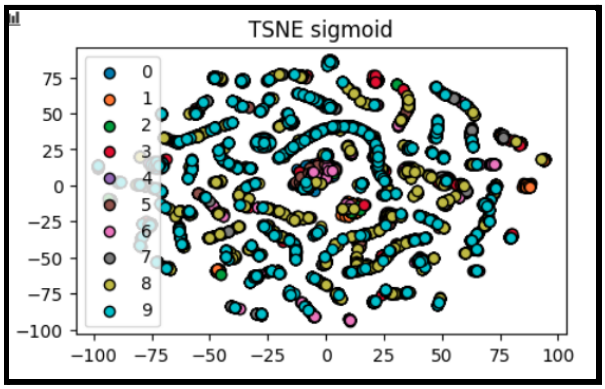
Total number of hidden layers: 3

Part 5:







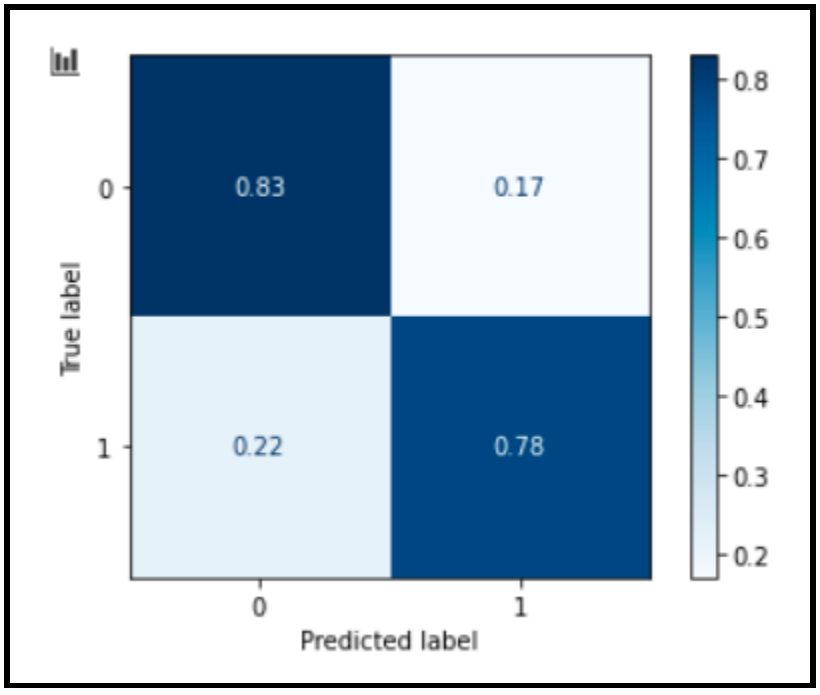


Model	Test Accuracy(After 100 Epochs)
Custom Model(ReLU)	0.982700
Custom Model(Tanh)	0.954700
Custom Model(linear)	0.917100
Custom Model(Sigmoid)	0.335800
Sklearn Model(ReLU)	0.965600
Sklearn Model(Tanh)	0.971600
Sklearn Model(Identity)	0.915400
Sklearn Model(Logistic)	0.970200

Logistic model of Sklearn does not suffer from a low accuracy score whereas the custom model(sigmoid) does.
One reason behind this could be vanishing gradients for custom models, and the Sklearn model is able to compensate for that as it uses adam optimizer(momentum term).

Answer 3:

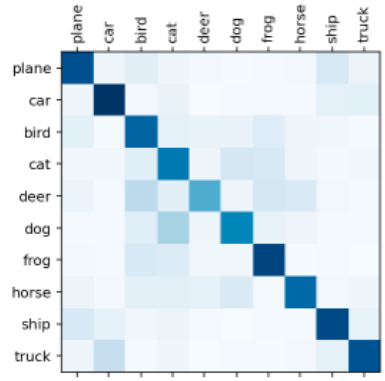
	precision	recall	f1-score	support
0	0.77	0.83	0.80	94
1	0.84	0.78	0.81	106
accuracy			0.81	200
macro avg	0.81	0.81	0.80	200
weighted avg	0.81	0.81	0.81	200



Answer 4:

Accuracy & Confusion Matrix

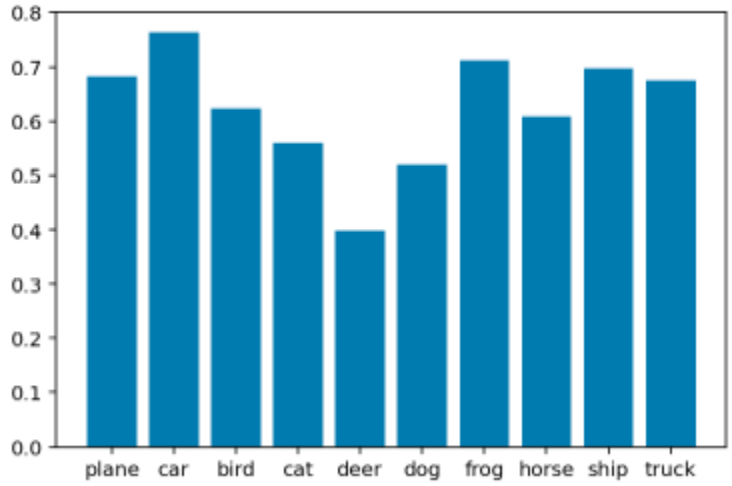
```
Test Accuracy: 0.6243000030517578
```



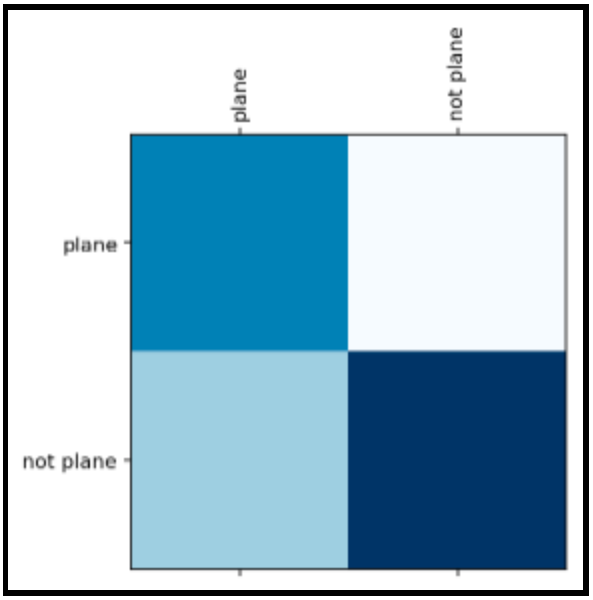

```
[[682.  25.  72.  24.  13.   4.  10.  17. 117.  36.]
 [ 24. 763.  13.  48.   0.   8.   6.   6.  63.  69.]
 [ 70.   6. 623.  59.  51.  49.  92.  26.  21.   3.]
 [ 23.  22.  79. 560.  24. 122. 108.  26.  15.  21.]
 [ 41.  10. 202.  72. 398.  31. 120. 102.  18.   6.]
 [  7.  10.  81. 250.  27. 521.  53.  34.  11.   6.]
 [ 13.  11. 112.  93.  21.  21. 713.   5.  10.   1.]
 [ 38.  17.  67.  67.  55. 103.   8. 609.  12.  24.]
 [113.  56.  21.  36.   7.   0.   9.   9. 698.  51.]
 [ 31. 168.   5.  29.   2.   7.   3.  15.  64. 676.]]
```

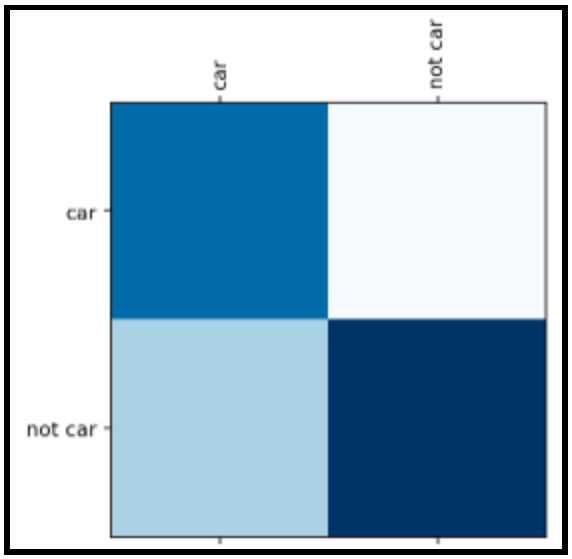
Class Wise Accuracy

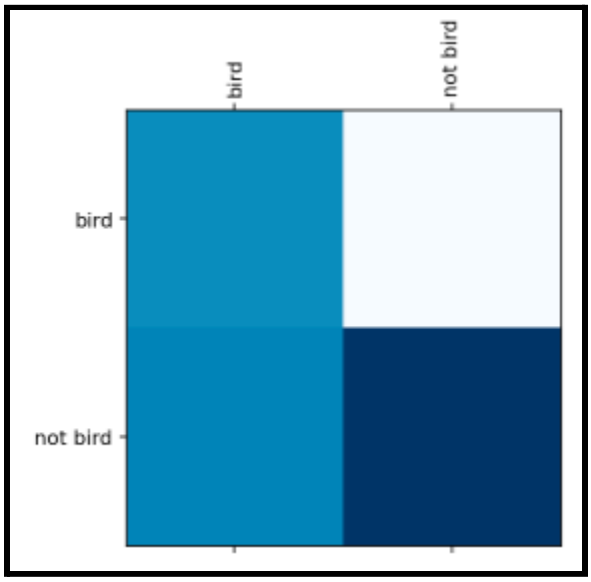
```
[0.682 0.763 0.623 0.56  0.398 0.521 0.713 0.609 0.698 0.676]
```

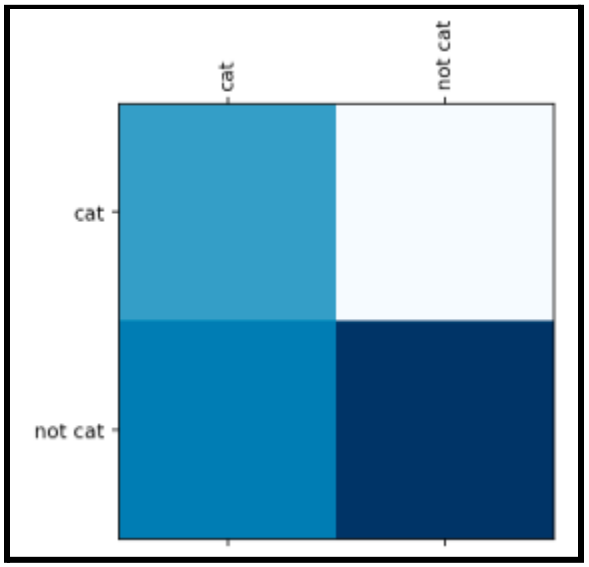


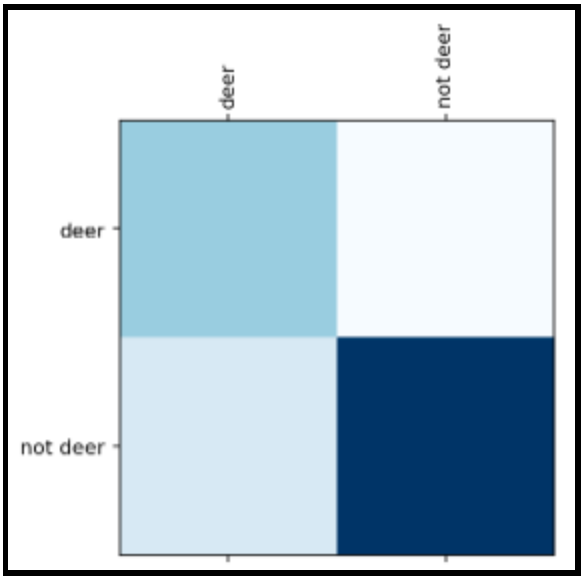
Class Wise Confusion Matrices & Loss Plot

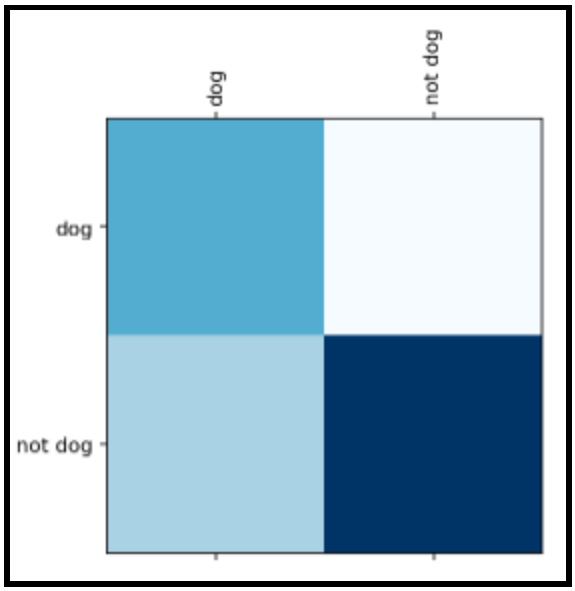

plane
not plane
plane
not plane

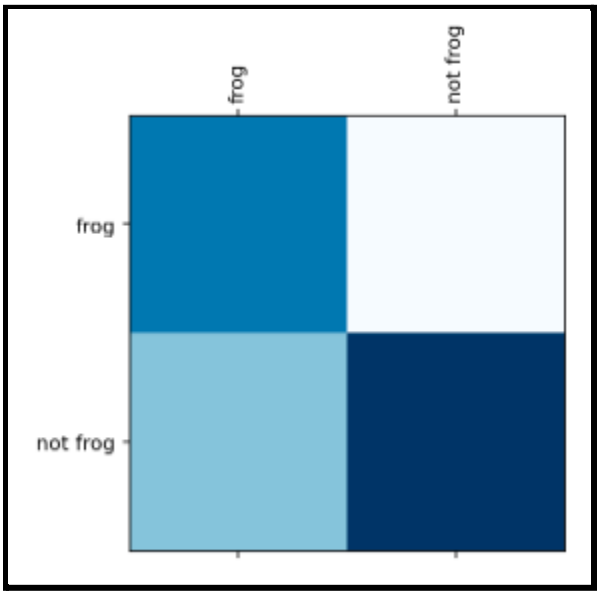

car
not car
car
not car

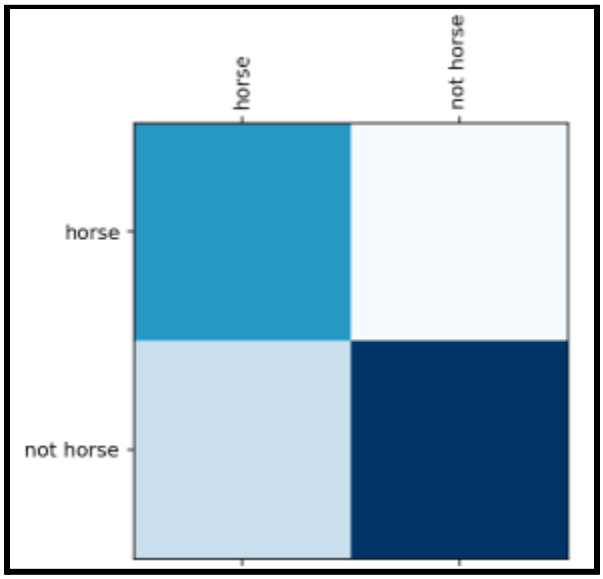

bird
not bird
bird
not bird

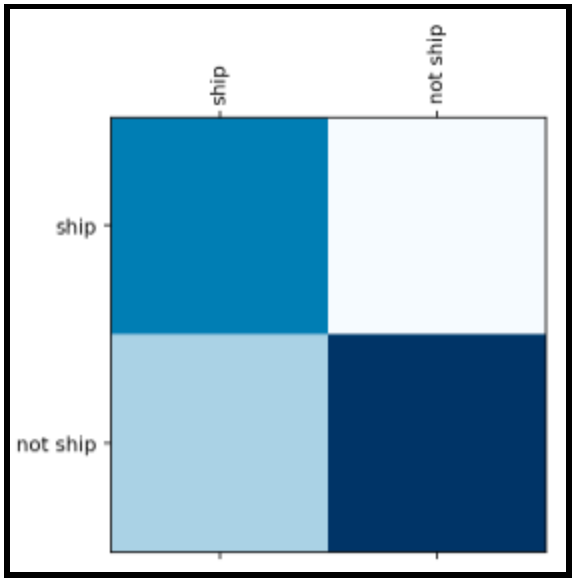

cat
not cat
cat
not cat

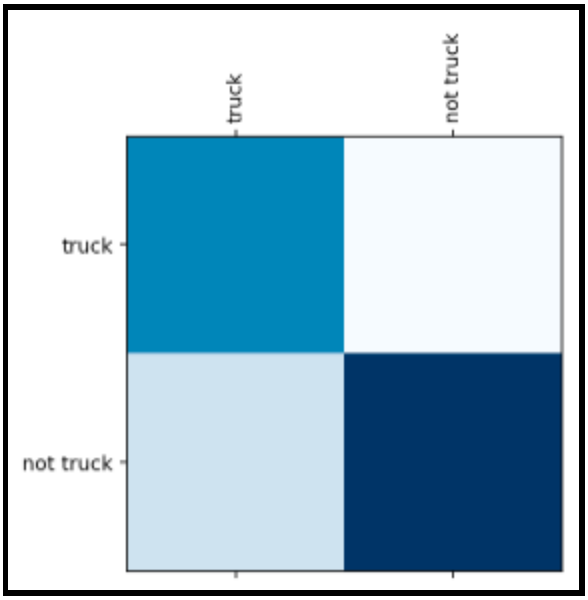

deer
not deer
deer
not deer

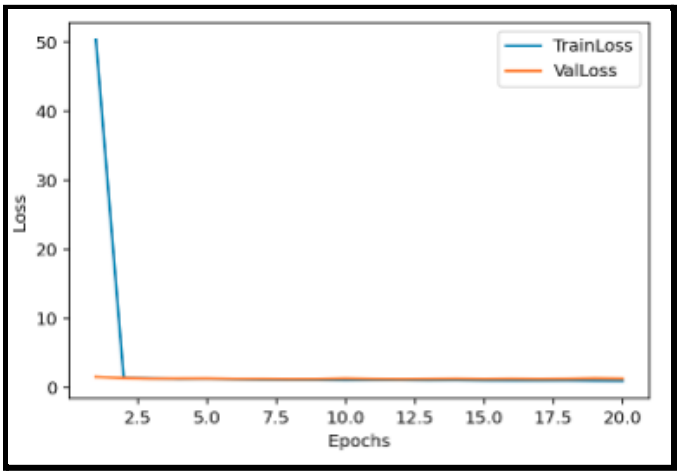

dog
not dog
dog
not dog


frog
not frog
frog
not frog


horse
not horse
horse
not horse






```
[[0.682      0.03533333]
 [0.36       0.96      ]]
[[0.763      0.02633333]
 [0.325      0.96388889]]
[[0.623      0.04188889]
 [0.652      0.92755556]]
[[0.56       0.04888889]
 [0.678      0.92466667]]
[[0.398      0.06688889]
 [0.2        0.97777778]]
[[0.521      0.05322222]
 [0.345      0.96166667]]
[[0.713      0.03188889]
 [0.409      0.95455556]]
[[0.609      0.04344444]
 [0.24       0.97333333]]
[[0.698      0.03355556]
 [0.331      0.96322222]]
[[0.676      0.036     ]
 [0.217      0.97588889]]
```