Homework 3:

Problem 1

Problem 1

To represent the operation as a fully connected layer, we can pad W with 0s making it a circulant matrix:

$$W = \begin{bmatrix} 1 & 3 & 0 & -1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 0 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 3 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 3 & 0 & -1 & 2 \end{bmatrix}$$

To prove that this is a valid circulant matrix, let $\text{vec}[x] = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{bmatrix}$

$$h = Wx = \begin{bmatrix} 1 & 3 & 0 & -1 & 2 & 0 & 0 & 0 \\ 0 & 1 & 3 & 0 & -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 3 & 0 & -1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 3 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{bmatrix} = \begin{bmatrix} 13 \\ 18 \\ 28 \\ 33 \end{bmatrix}$$

Unvectorized, this becomes $\begin{bmatrix} 13 & 18 \\ 28 & 33 \end{bmatrix}$

Applying the filter to the unvectorized x, we get

$x = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$, filter = $\begin{bmatrix} 1 & 3 \\ -1 & 2 \end{bmatrix}$

Feature map =

$$\begin{bmatrix} (1*1+2*3+4*-1+5*2) & (2*1+3*3+5*-1+6*2) \\ (4*1+5*3+7*-1+8*2) & (5*1+6*3+8*-1+9*2) \end{bmatrix}$$

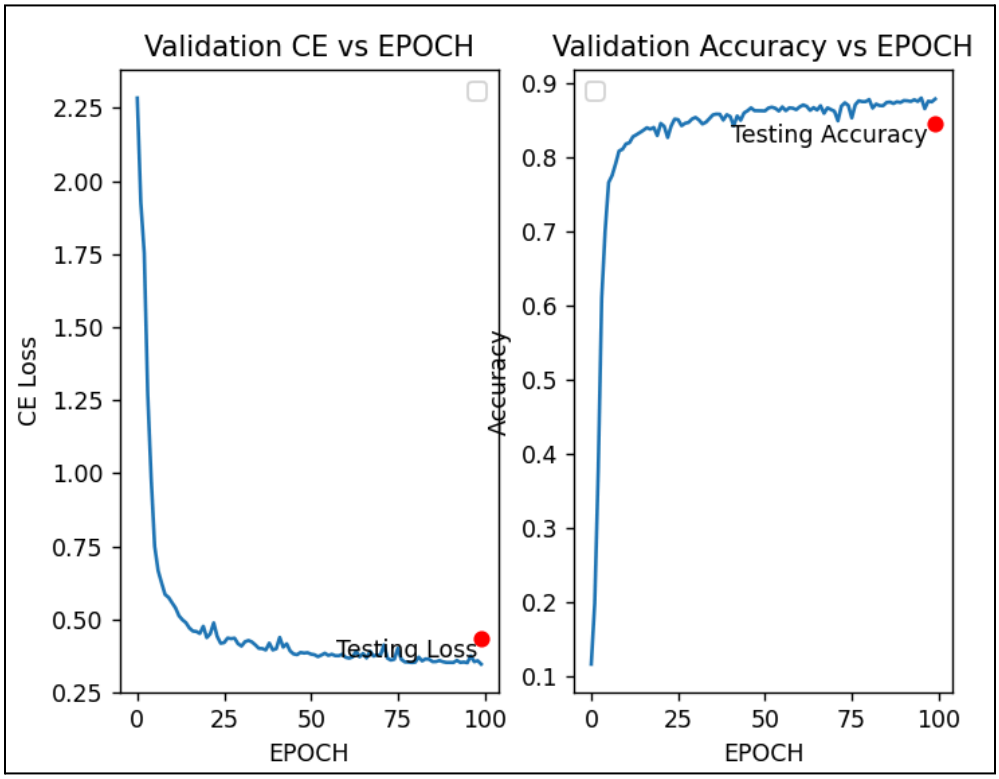
$$= \begin{bmatrix} 13 & 18 \\ 28 & 33 \end{bmatrix}$$

Problem 2

2A)

The generated model defined in the code, with EPOCHS=100, Learning Rate=0.01, L2_STRENGTH=0.01 and BATCH_SIZE=64 produced a final testing loss of 0.43 and a testing accuracy of 85%. Please Review the attached logs outputs for Question 2 and the provided Python File for all code implementations.

```
Testing Loss: 0.4360252523286244 and Testing Accuracy: 0.8464
```



2B)

The model could train and reached a HIGH testing accuracy, which was on par with an identical Tensorflow Model, but did not produce a check_grad score < 1e-1.

```
The check_grad value is:
0.4310760314538392
```

Problem 3

3A)

Please Review the attached logs outputs for Question 3 and the provided Python File for all code implementations.

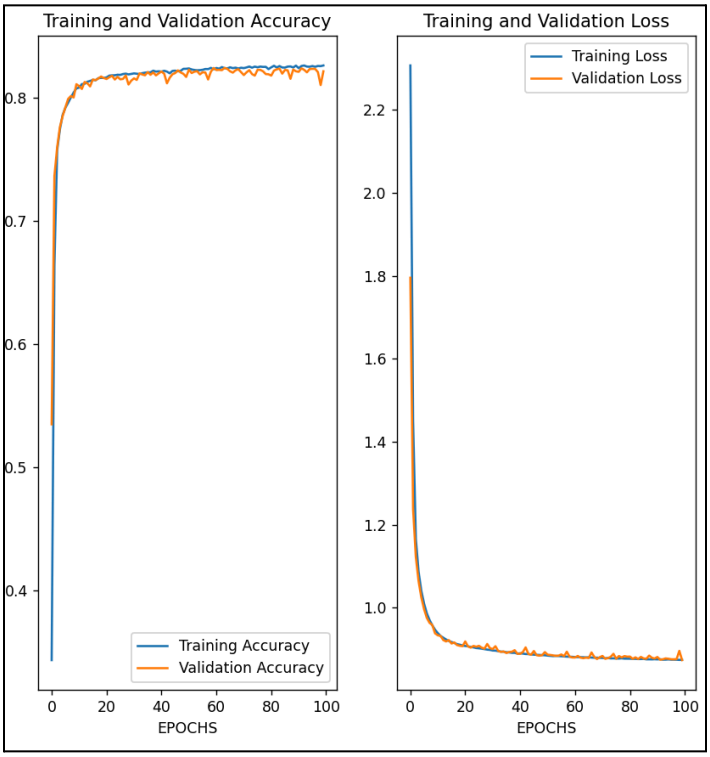


Figure 1: Simple Replicated NN Using Tensorflow

3B)

The best set of hyperparameters was {EPOCH: 220, LAYER_AMT: 5, NEURONS: 50, LR: 0.001, BATCH: 64, L2: 0.001. Which produced a validation result of {'validation_results': {'loss': 0.6098987460136414, 'accuracy': 0.8796666860580444}}. Please Review the attached logs outputs for Question 3 and the provided Python File.

3C)

The best model from the previous part (mentioned above) produced a Testing Accuracy of 86.75% and a Testing Loss of 0.64. Please Review the attached log outputs for Question 3 and the provided Python File for all code implementations. Figures 2 and 3 below shows the model Accuracy and Loss over each Training Epoch for the entire 220 Epoch span and the final 20 epochs. In Figure 3, the final 20 Epochs, the accuracy, and Loss are harder to view, since they are hidden behind the chart legend, but are visible in the bottom left and top right respectively. All results are available in the Q3 Logs.

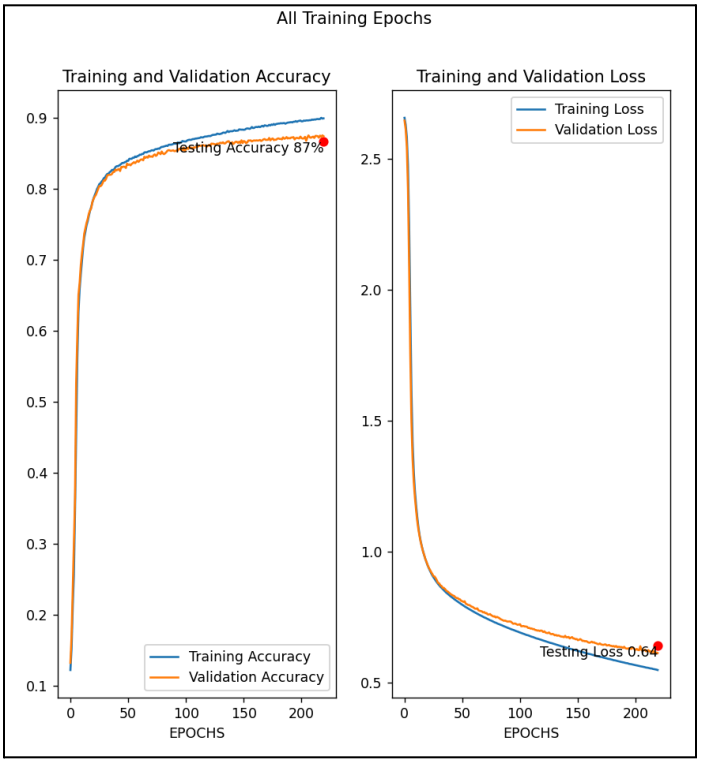


Figure 2: Best NN Training Epoch Results

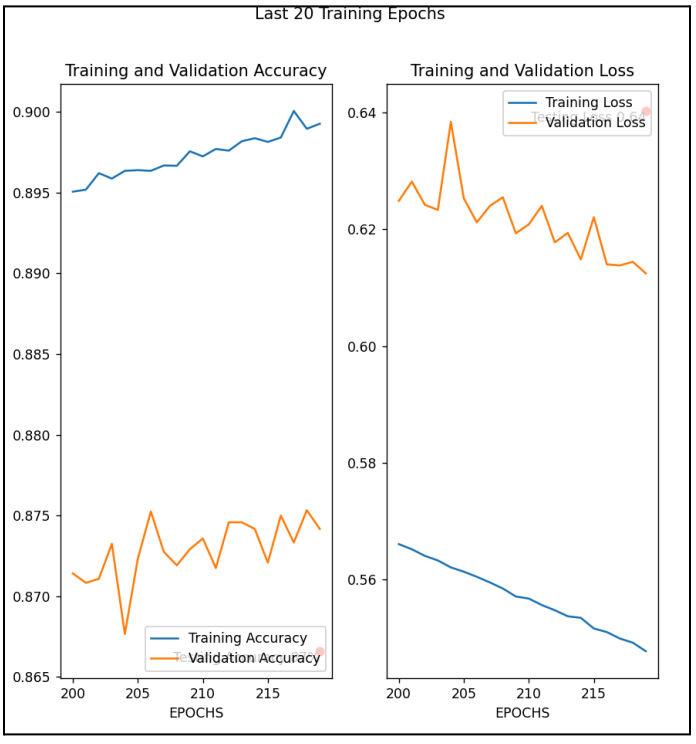


Figure 3: Best NN Training Final 20 Epoch Results

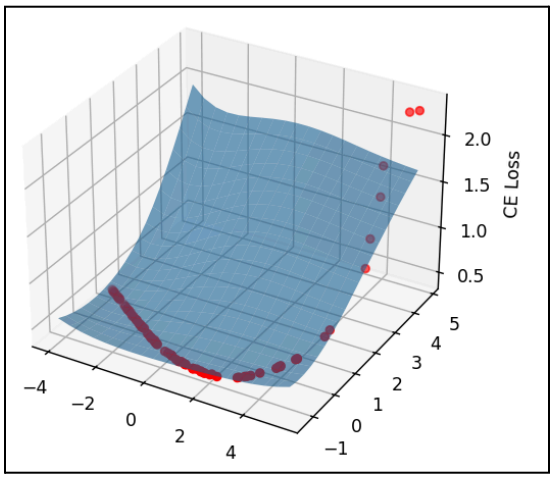
Problem 4

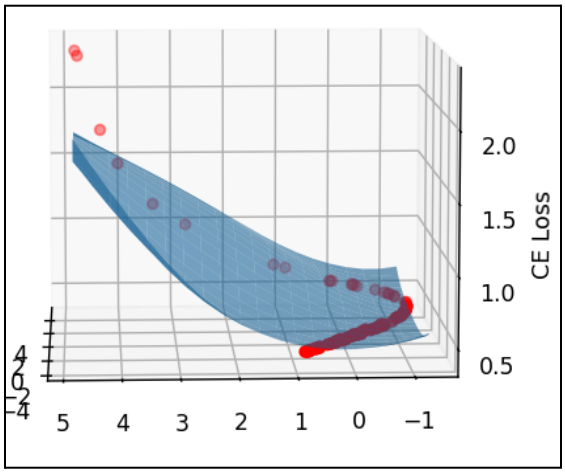
4A)

Please Review the screenshots provided below and the attached Python File for all code implementations.

4B)

Please Review the screenshots provided below and the attached Python File for all code implementations.





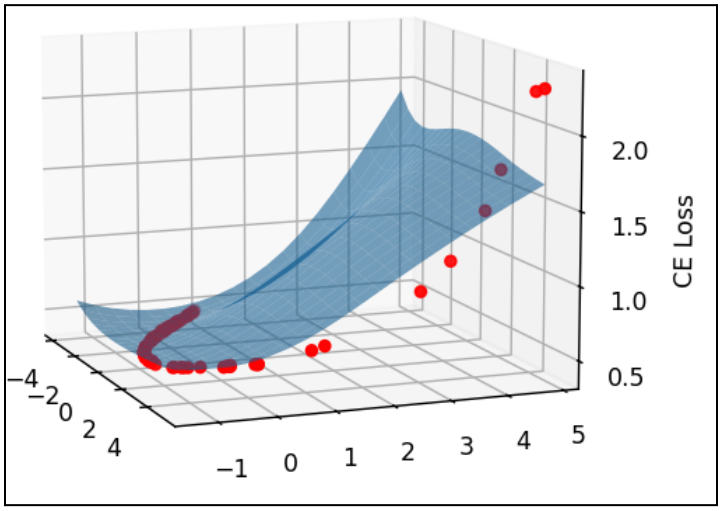


Figure 4, 5 & 6: SGD Space Plot and Weight Line