Fatima Ezzahra Safiri

CMPT 310

Dr. Lim

Report

Question 1:

The combination (NUM_CLUSTERS, MAX_ITER) I experimented with for this question are the following:
(2, 5) (2,10) (2,20) (3,5) (3,10) (3,20) (4,5) (4,10) (4,20)

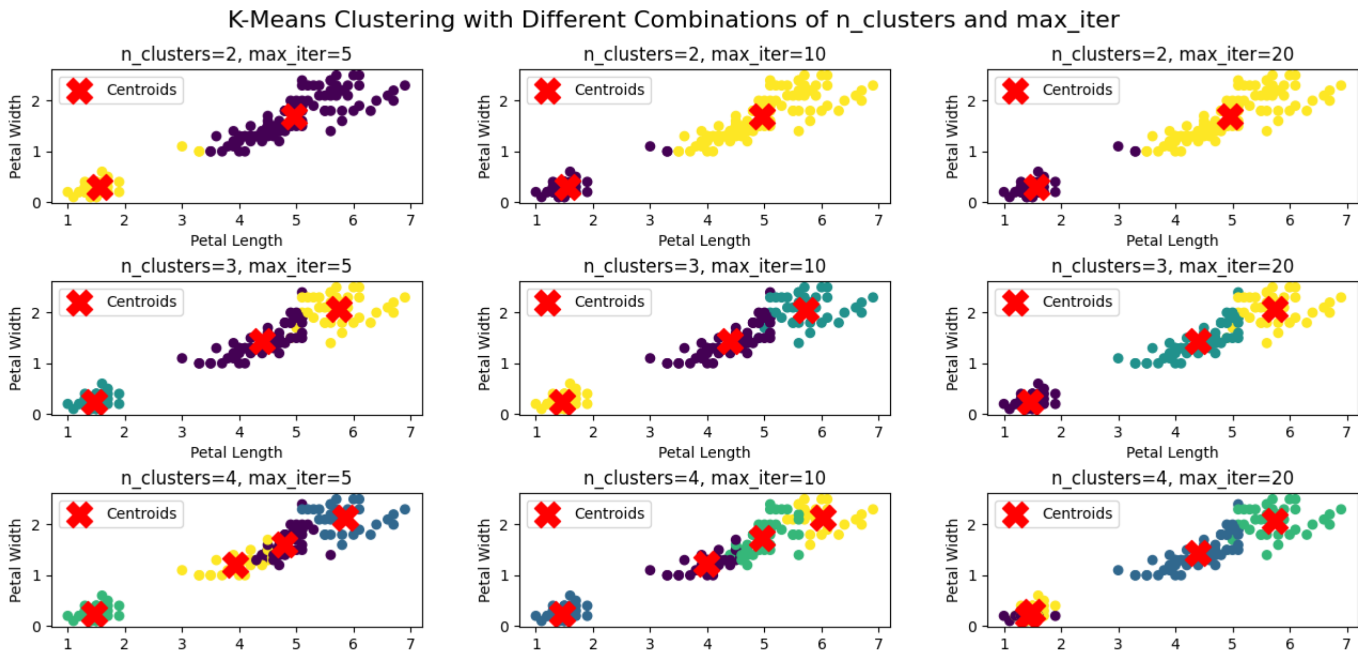


Figure 1 K-means clustering with different combinations

For the number of clusters parameter:

2 clusters split the data into two broad groups based on petal sizes, but it's too basic and misses finer details.

3 clusters work best for the Iris dataset, matching its three species. The clusters look more natural and less overlapping.

4 clusters split the data too much, making things overly complicated without adding much useful information.

Number of Iterations parameter:

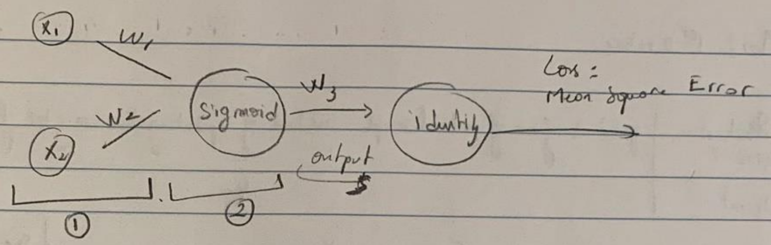
5 iterations, the clustering isn't great, and centroids don’t settle properly, so the groups are less accurate. We can see the overlapping of some data points.

10 iterations, centroids are better placed, clusters look cleaner, and the algorithm stabilizes.

20 iterations, there is no big difference from 10 iterations, meaning the algorithm already converged and more iterations don't help.

As we can see in *figure 1* the best results come from the combination of 3 clusters and at least 10 iterations. Too many clusters or iterations don’t necessarily make things better and might even overcomplicate the model. So using the (3,10) combination gives the best balance, fitting the Iris dataset well without overfitting.

Question 2:



Forward pass :

① $Z = W_1 \times X_1 + W_2 \times X_2 = (0.5 \times 1) + (\cancel{0.5 \times 0})$
$= 0.5$

② Apply Sigmoid act fun:

$$S = 1/1 + e^{-0.5} = \underbrace{0.62}$$

$$\text{output} = W_3 \times S = 0.5 \times 0.62 = \underbrace{0.31}$$

$$\hat{y} = 0.31$$

Find loss :

$$\text{Loss}_{\text{MSE}} = (1 - 0.31)^2 = \underbrace{0.47}$$

Backpropogation: For W_3 :

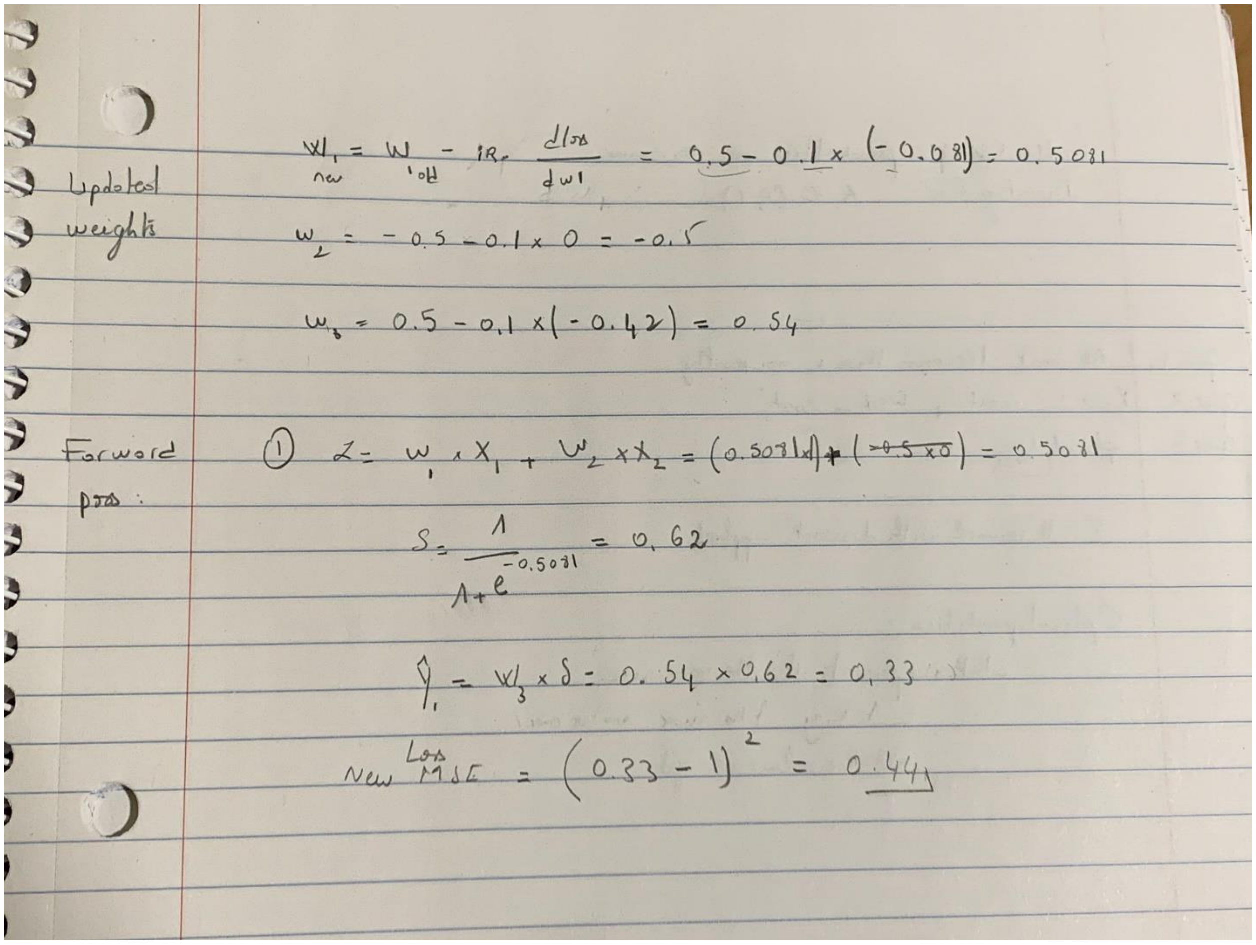
$$\frac{d\,\text{loss}}{dw_3} = \frac{d\,\text{loss}}{d\hat{y}} \times \frac{d\hat{y}}{dw_3} = (\hat{y} - y) \times S = -0.69 \times 0.62$$
$$= -0.42$$

For W_2 :

$$\frac{d\,\text{loss}}{dW_2} = 0$$

For W_1 :

$$\frac{d\,\text{loss}}{dW_1} = \frac{d\,\text{loss}}{d\hat{y}} \times \frac{d\hat{y}}{dw_1} = \underbrace{\frac{d\,\text{loss}}{d\hat{y}}} \cdot \underbrace{\frac{d\hat{y}}{dS}} \times \underbrace{\frac{dS}{dZ}} \times \underbrace{\frac{dZ}{dW_1}}$$
$$= -0.69 \times W_3 \times S(1-S) \times X_1$$
$$= -0.69 \times 0.5 \times [0.62 \times \underbrace{(1-0.62)}_{0.38}] \times 1$$
$$= -0.081$$



Updated weights

$$W_{1_{new}} = W_{1_{old}} - lR \cdot \frac{dloss}{dw1} = 0.5 - 0.1 \times (-0.081) = 0.5081$$

$$w_2 = -0.5 - 0.1 \times 0 = -0.5$$

$$w_3 = 0.5 - 0.1 \times (-0.42) = 0.54$$

Forward pass:

① $z = w_1 \times X_1 + w_2 \times X_2 = (0.5081 \times 1) + (\cancel{-0.5 \times 0}) = 0.5081$

$$S = \frac{1}{1 + e^{-0.5081}} = 0.62$$

$$\hat{y}_1 = W_3 \times S = 0.54 \times 0.62 = 0.33$$

$$\text{New } \overset{\text{Loss}}{\text{MSE}} = (0.33 - 1)^2 = \underline{0.44}$$

Question 3

1. Learning Rate Experiment:

Learning Rate: 0.1

Epochs: 50

Hidden Layers: [128]

Test Accuracy: 97.88%

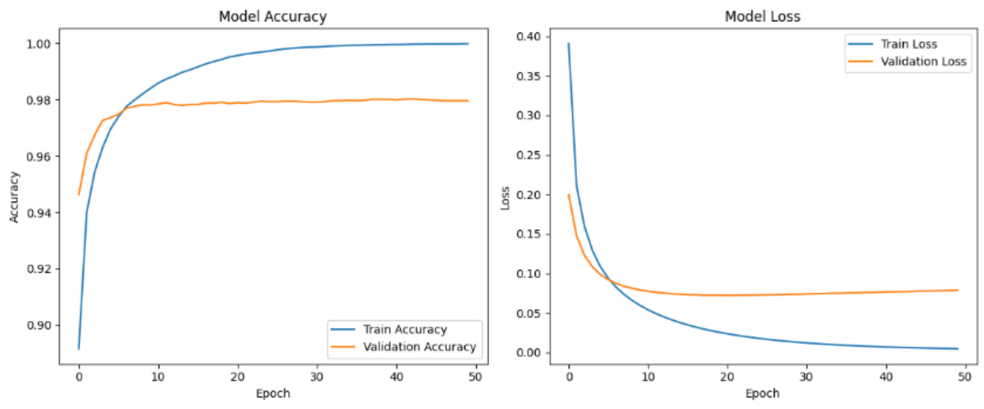


Figure 2

The training accuracy approaches 100%, indicating that the model learned the training data very well, and the validation accuracy stabilizes around 98%, which is strong but slightly lower than the training accuracy, suggesting some overfitting.

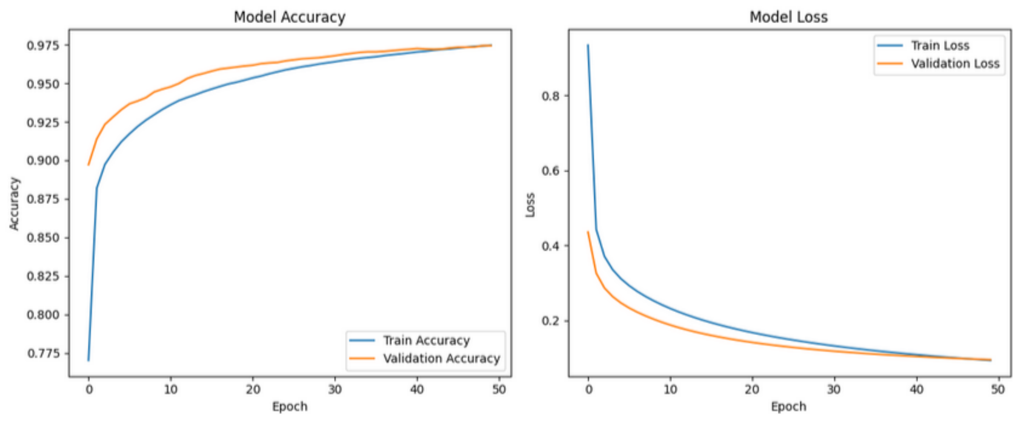
The training loss decreases rapidly and drops toward zero, which shows that the model is fitting the training data well. But, the validation loss stops decreasing after a point and even starts to increase. This indicates overfitting, so the model is starting to memorize the training data, but struggles to generalize to new data.

Learning Rate: 0.01

Epochs: 50

Hidden Layers: [128]

Test Accuracy: 97.04%



The training accuracy improves, reaching almost 98%, while the validation accuracy stabilizes around 97%. There's a small gap between the training and validation accuracy, which shows better generalization compared to the higher learning rate (0.1).

Both training and validation loss decrease progressively throughout the training process, with no sudden spikes in validation loss. This shows a stable learning process.

The learning rate of 0.01 resulted in slightly lower final accuracy compared to 0.1. But, the training was more stable, and there's less evidence of overfitting.

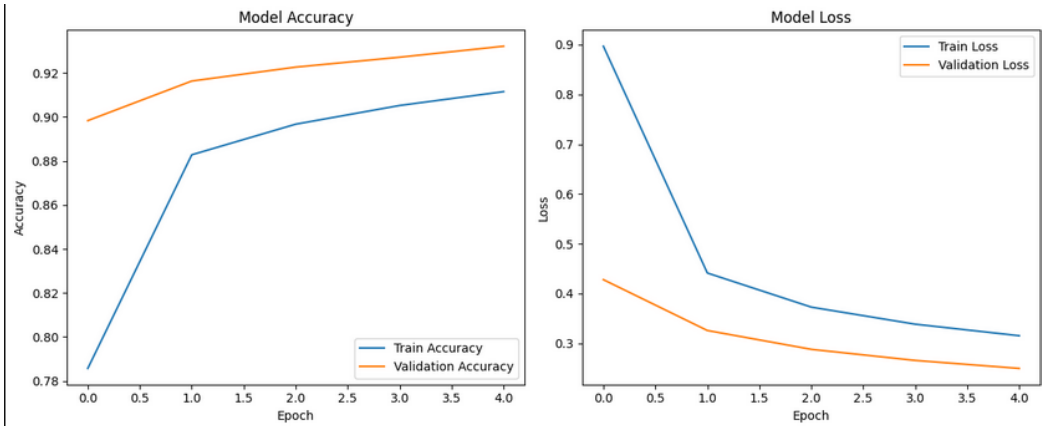
2. Epoch Experiment:

Learning Rate: 0.01

Epochs: 5

Hidden Layers: [128]

Test Accuracy: 92.18%



Training accuracy improves quickly in the first few epochs, reaching 92% by epoch 5.

The validation accuracy also reaches a stable value of around 92% by the end of the training, showing that the model has become stable in terms of learning at this point, and the model is not overfitting yet.

Both the training and validation loss decrease in the first few epochs and stabilize at low values by epoch 5. The validation loss follows a similar trend of training loss, which confirms that the model is learning and generalizing well.

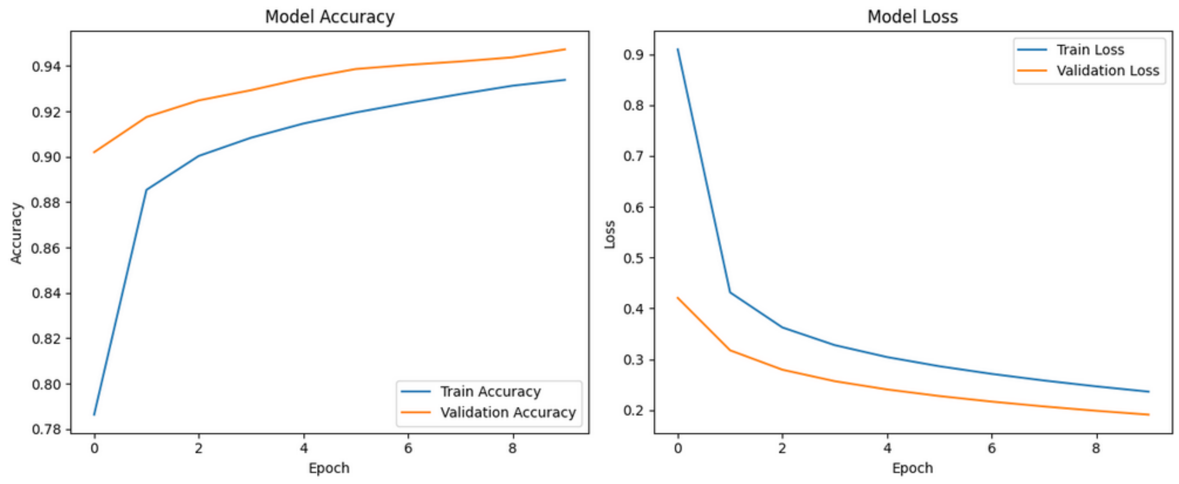
So, the model is performing properly with 92.19% accuracy, but more training epochs would likely improve performance as the loss is still decreasing.

Learning Rate: 0.01

Epochs: 10

Hidden Layers: [128]

Test Accuracy: 93.63%



The validation accuracy is slightly higher than training, stabilizing around 95%, this shows that the model is still learning without overfitting. Both training and validation losses decrease progressively, which shows improvement with each epoch. The validation loss lower than training, which is consistent with the higher validation accuracy.

By increasing the epochs from 5 to 10, the test accuracy went up, which shows the model is still learning, so training more epochs could improve it even further.

Learning Rate: 0.01

Epochs: 50

Hidden Layers: [128]

Test Accuracy: 97.04%

After 50 epochs, the model is performing almost at its best, with loss curves smoothing and only a small test accuracy improvement compared to 10 epochs, so 50 epochs seems enough. The training and validation accuracy are really close, and validation loss is still going down, which means there's no overfitting and the model is generalizing well. The test accuracy went from 92.19% at 5 epochs to 93.63% at 10, and finally 97.04% at 50 epochs, but most of the improvement happened between 5 and 10 epochs, with smaller gains after 20-30 epochs.

3. Layers Experiment:

Learning Rate: 0.01

Epochs: 50

Hidden Layers: [128]

Test Accuracy: 97.04%

One hidden layer with 128 units is enough to get good performance on this dataset. There's no overfitting because the validation loss is close to the training loss, and the accuracy curves match, showing that the model generalizes well to new data.

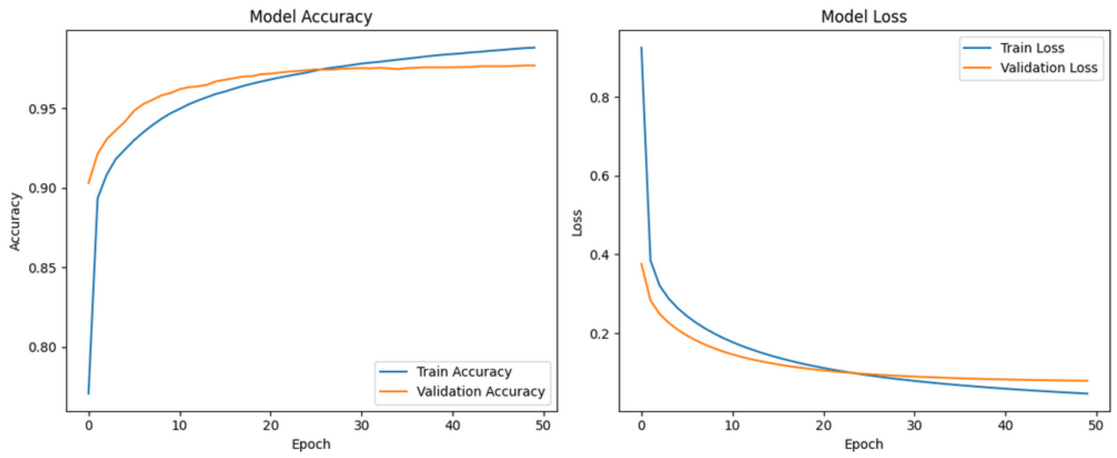
See figure 2

Learning Rate: 0.01

Epochs: 50

Hidden Layers: [128, 128]

Test Accuracy: 97.50%



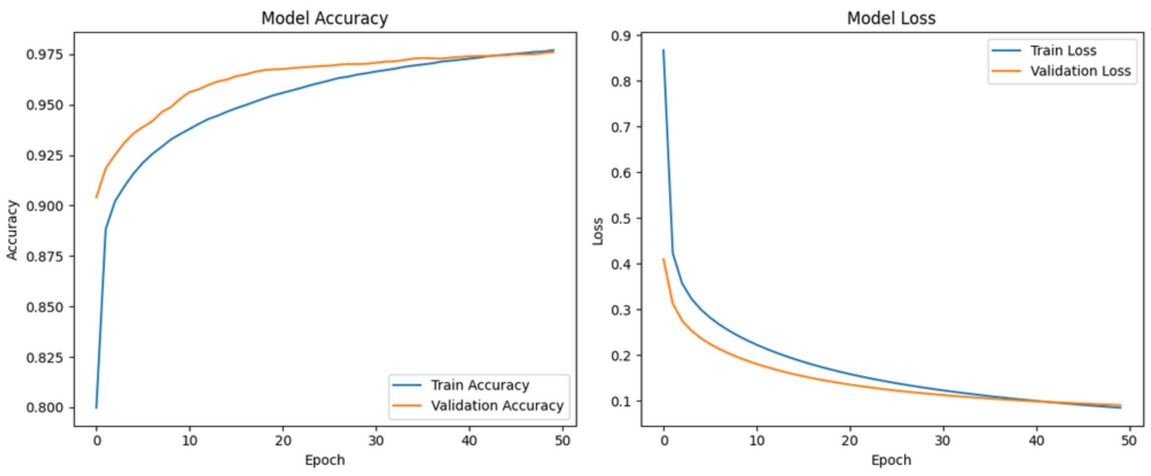
The model did a little better with two hidden layers compared to one, going from 97.04%to 97.50% accuracy. This shows that adding another hidden layer with 128 units helps the model learn more complex patterns in the data. There's no overfitting since the training and validation curves are still aligned, and the validation loss is lower than the training loss. Overall, the second hidden layer gave the model more capacity, which improved its ability to learn important features.

Learning Rate: 0.01

Epochs: 50

Hidden Layers: [256]

Test Accuracy: 97.14%



The final test accuracy is 97.14%, which is a bit lower than the 97.50% from the two hidden layer model, but still better than the single layer model with 128 units. Increasing the hidden units from 128 to 256 improved accuracy to 97.14%, but the gains may decrease as the model becomes complex. This shows that more units help the model learn more complex patterns. There's no overfitting but the experiment with two layers of 128 units had a better result, so in this case adding a second layer is more helpful than just increasing units in one layer.

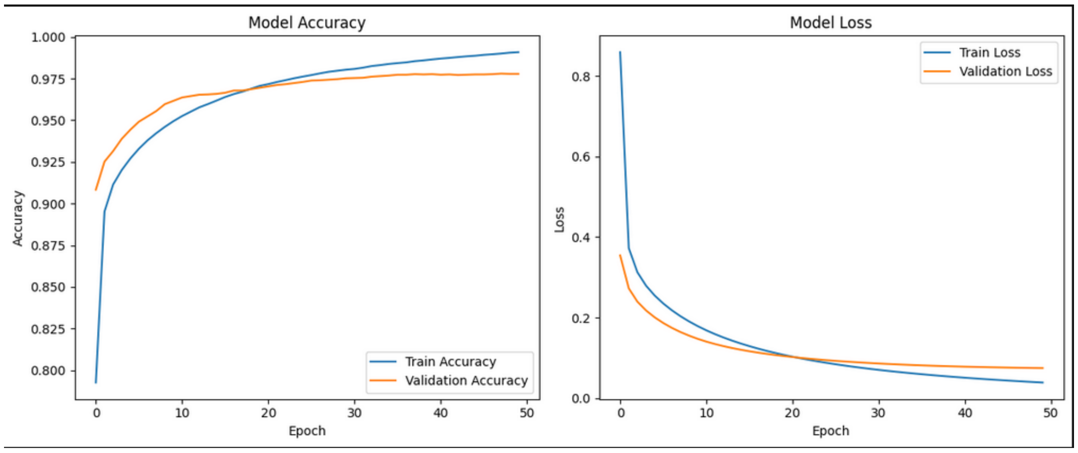
Learning Rate: 0.01

Epochs: 50

Hidden Layers: [256, 256]

Test Accuracy: 97.71%

The final test accuracy was 97.71%, the best so far, showing that two hidden layers with 256 units each help capture more complex patterns. The validation accuracy closely matches training, meaning the extra complexity is helping without causing overfitting. Even with more complexity, the model generalizes well. Test accuracy improved from 97.14% with one layer to 97.71% with two layers.


Model Accuracy
Accuracy
Epoch
Train Accuracy
Validation Accuracy
Model Loss
Loss
Train Loss
Validation Loss