Project 2 Report: Handwritten Digits Recognition using Neural Networks

Jordan S-H

https://github.com/jstebner/481-P2

[0] Overview

The aim of this project is to test the efficacy of several neural network architectures, optimized using gradient descent. The architectures to be tested are:

1. Multioutput perceptron, with varying learning rates.
2. Deep neural network, with varying depth.
3. Deep neural network, with varying learning rates.

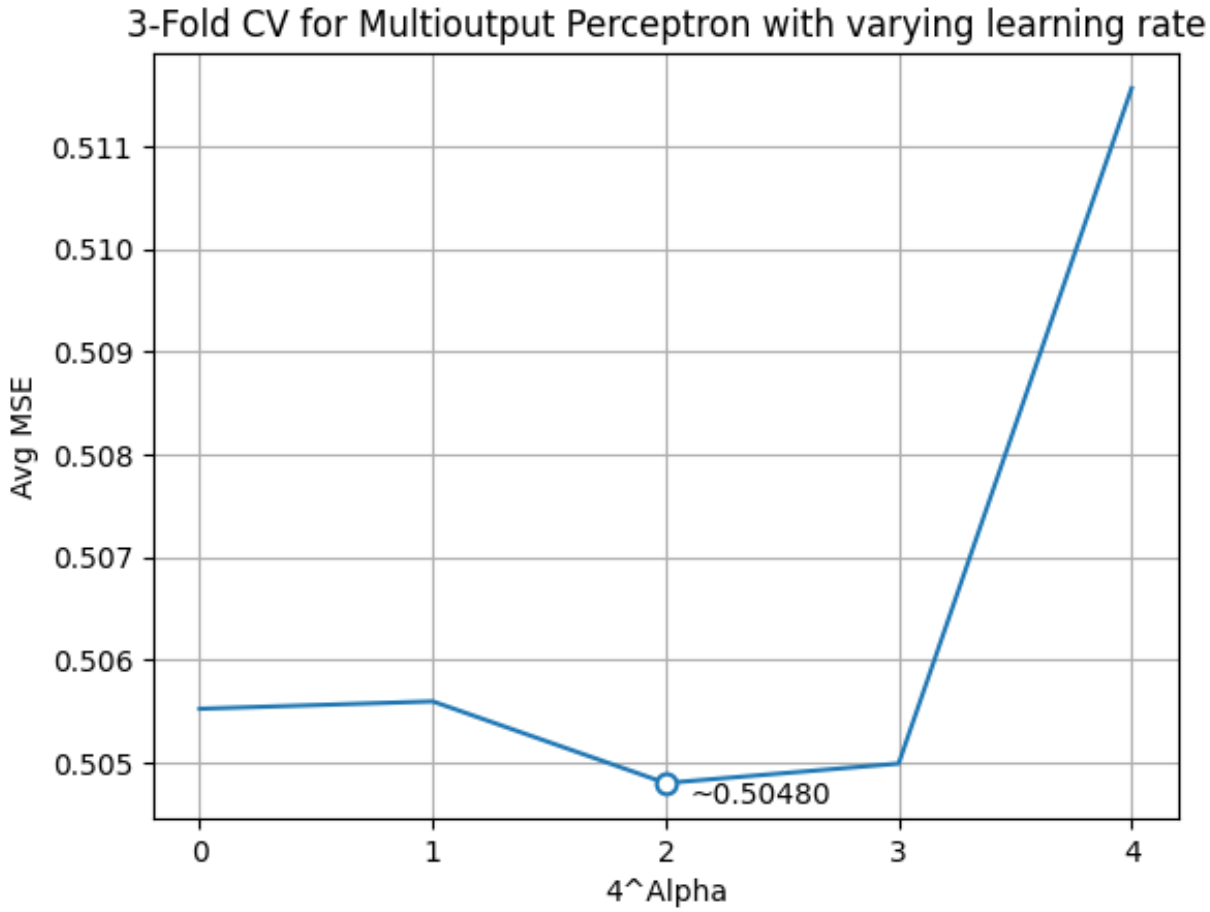
3-fold cross-validation will be used to select the optimal hyperparameter for each model, which will be used to train an optimal model for each architecture to see the effectiveness of each.

The data that the models will be trained on is of handwritten digits, where each pixel is represented with a 0 (black) or a 1 (white), where the dimensions of each image is 32 x 32. All data used for training and testing is stored in the '**data/**' directory.

The implementations for neural network classes and utilities were all provided by Professor Luis E. Ortiz in the file '**src/learn_nnet.ipynb**', with the implementations for this project added at the end of the same file.

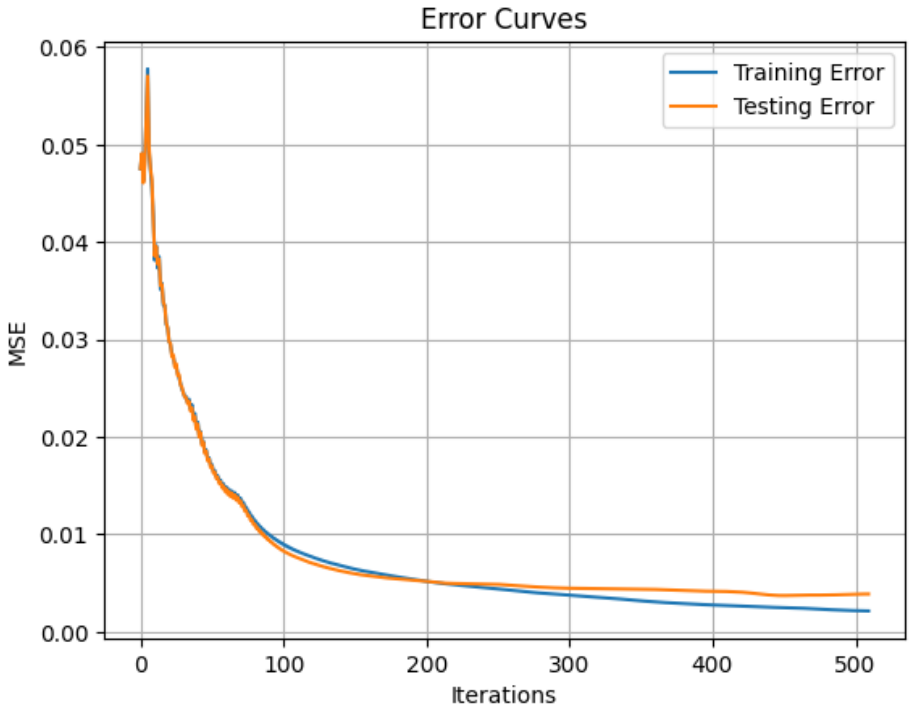
[1] Multi-output Perceptron

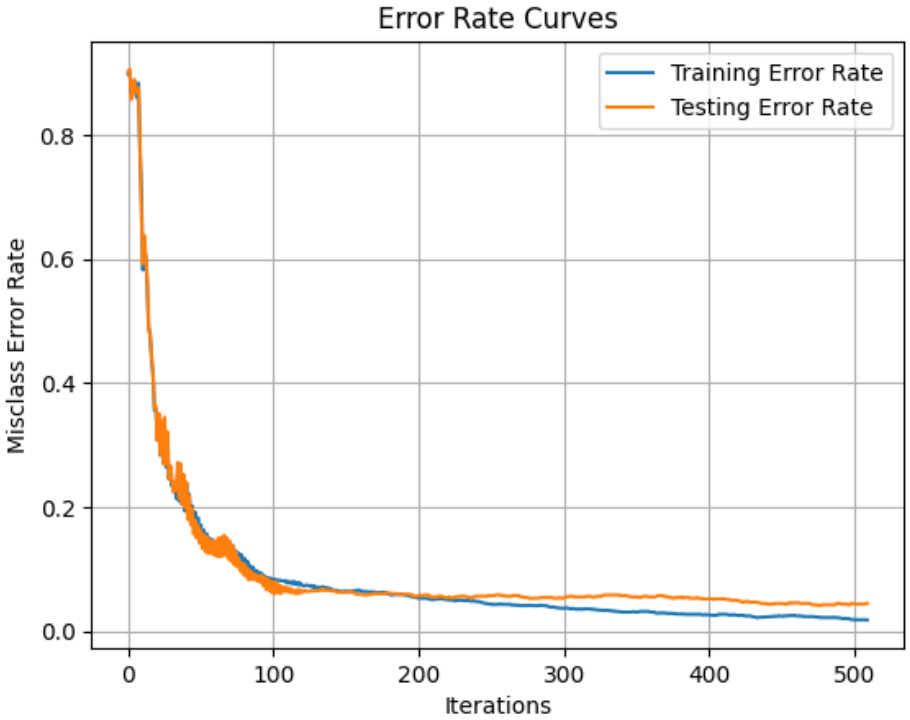
The multi-output perceptron has 1024 input units and 10 output units, with no hidden layers. Below depicts the graph of performing 3-fold cross-validation to select the optimal learning rate.



From the figure, we can see the optimal learning rate is 4^2 (or 16) with an average MSE of about 0.505. The optimal model will be trained using this learning rate.

Below are the curves for the test proxy function (first) and the misclassification error rate (second).





We can see in both that the training and testing error decline consistently, with a sharp drop until around 100 rounds. Also, the training curves stay below the testing curve, which is to be expected.

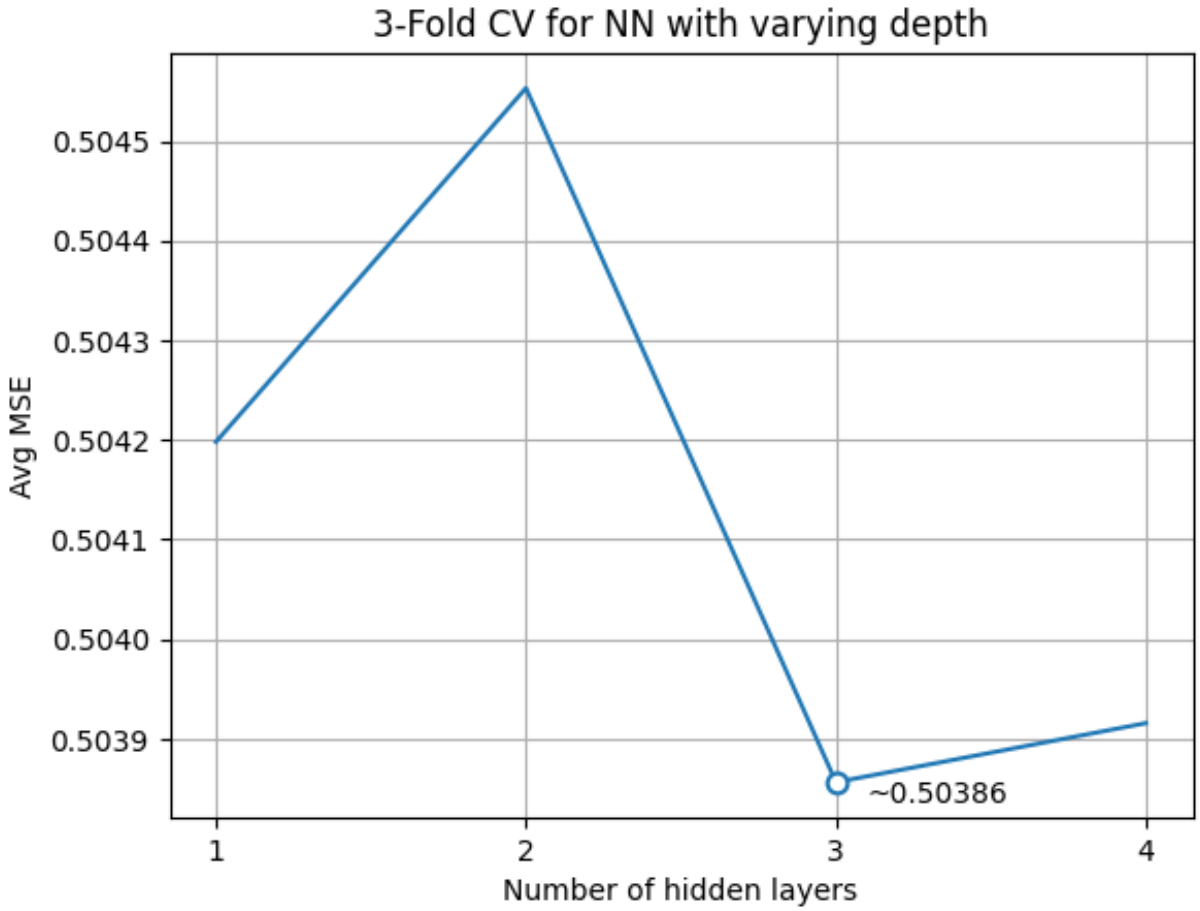
Below are the results of running the model using the trial dataset along with the confidence of the predictions.

```
Actual:    1
Predicted: 1      Confidence: 0.9484045139242051
Actual:    2
Predicted: 2      Confidence: 0.9952541579749694
Actual:    3
Predicted: 3      Confidence: 0.9734311607605
Actual:    4
Predicted: 4      Confidence: 0.6191694899587173
Actual:    5
Predicted: 5      Confidence: 0.9875683409382668
Actual:    6
Predicted: 6      Confidence: 0.9872908714572105
Actual:    7
Predicted: 7      Confidence: 0.9935059993137173
Actual:    8
Predicted: 8      Confidence: 0.9729356601915803
Actual:    9
Predicted: 9      Confidence: 0.9726205166724261
Actual:    0
Predicted: 0      Confidence: 0.8862707247042487
```

We can see that the model was accurate for all labels, each with pretty high confidence (except for classifying 4, which had a lower confidence than all other labels).

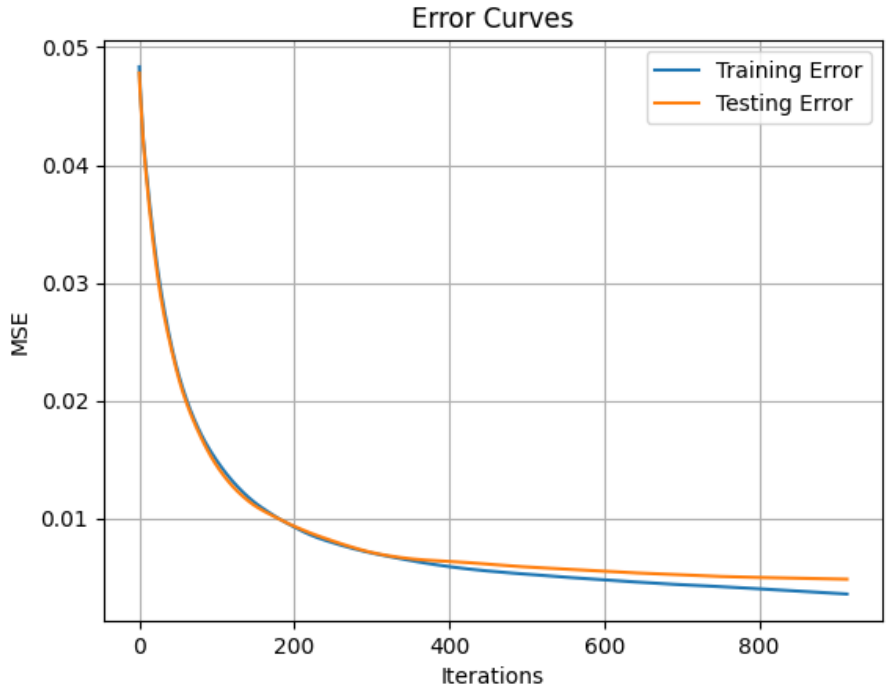
[2] Deep Neural Network with Varying Depth

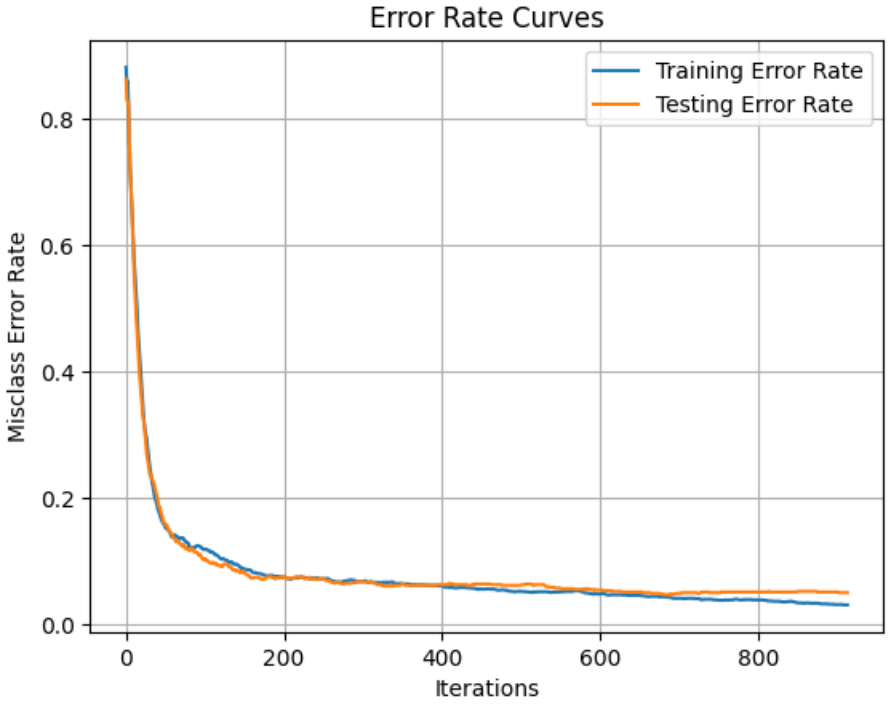
The first deep neural network architecture will have the same dimensions for input and output as the perceptron, however we will be varying the number of hidden layers present in the model (each with 64 units). Below depicts the graph of performing 3-fold cross-validation to select the optimal depth.



From the figure, we can see the optimal depth is 3 hidden layers with an average MSE of about 0.504. The optimal model will be trained using this depth.

Below are the curves for the test proxy function (first) and the misclassification error rate (second).





We can see in both that the training and testing error decline consistently, with a sharp drop until around 200 rounds (as opposed to 100 for the perceptron model). The training curves stay below the testing curve again, like the perceptron model. The model ran for many more rounds than the perceptron model as well (approx. 900, vs. approx. 500 for the perceptron).

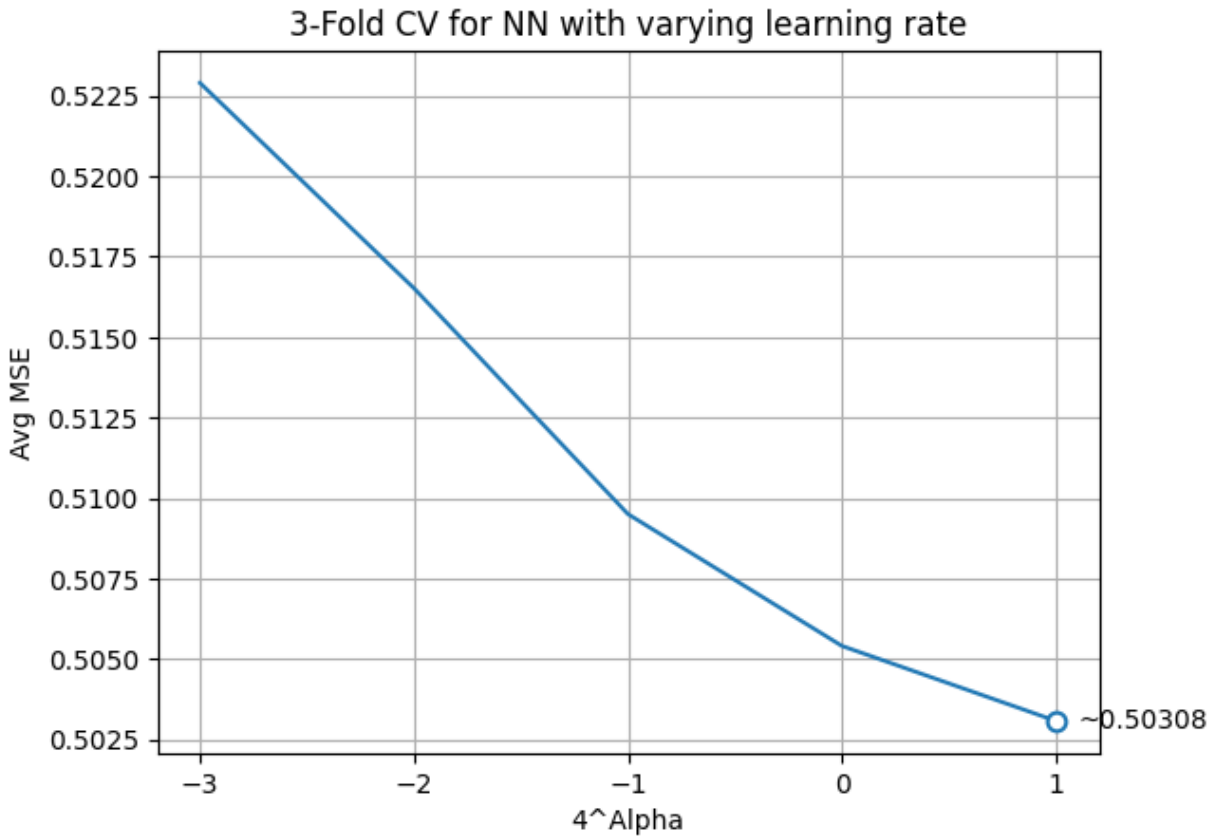
Below are the results of running the model using the trial dataset along with the confidence of the predictions.

```
Actual:    1
Predicted: 1      Confidence: 0.9020910182339072
Actual:    2
Predicted: 2      Confidence: 0.997960475657601
Actual:    3
Predicted: 3      Confidence: 0.9933097829146402
Actual:    4
Predicted: 4      Confidence: 0.7819241957383863
Actual:    5
Predicted: 5      Confidence: 0.9697171070932781
Actual:    6
Predicted: 6      Confidence: 0.9857734731145947
Actual:    7
Predicted: 7      Confidence: 0.9699866855418702
Actual:    8
Predicted: 8      Confidence: 0.962495634970271
Actual:    9
Predicted: 9      Confidence: 0.890421038639868
Actual:    0
Predicted: 0      Confidence: 0.9915151508883062
```

We can once again see that the model is accurate for all labels with high confidence, however the confidence for the label 4 is lower than all others (like the perceptron model).

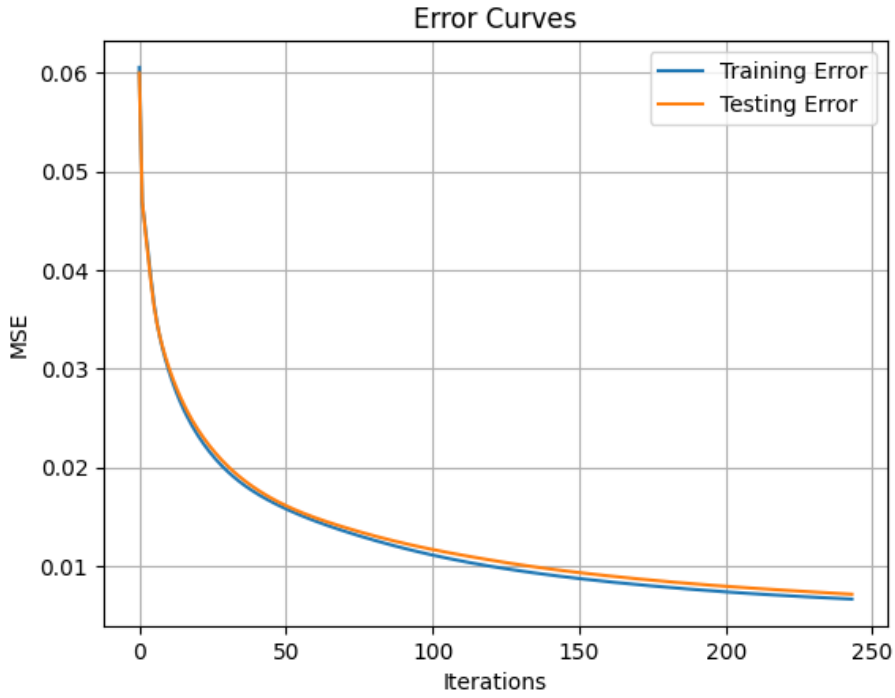
[3] Deep Neural Network with Varying Learning Rate

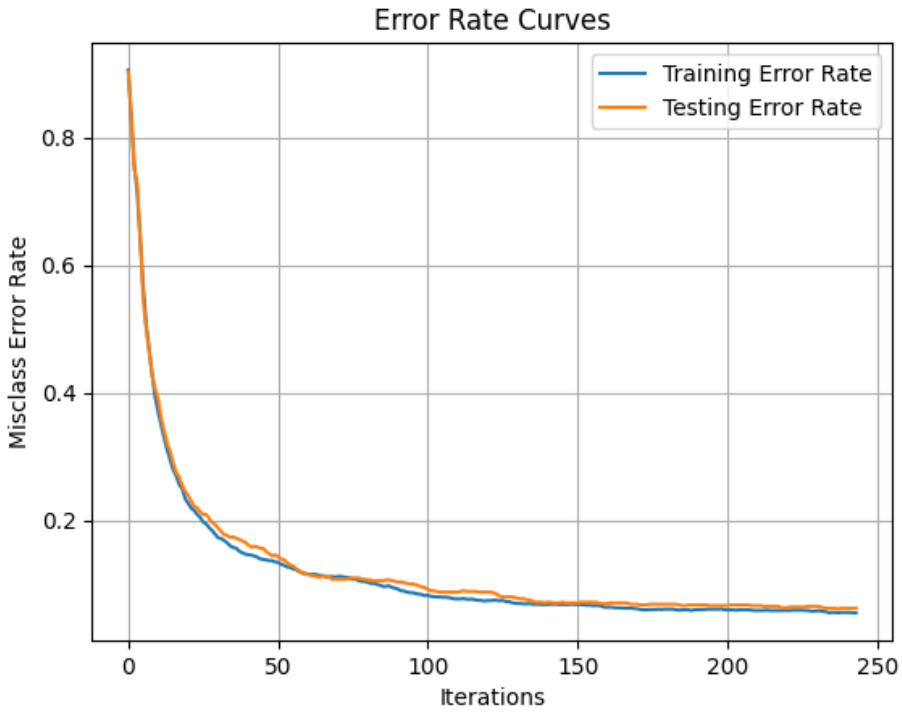
The second deep neural network architecture will have the same dimensions for input and output as both other models, however the model will contain 2 hidden layers (256 and 64 units each) and we will be varying the learning rate used to fit the model. Below depicts the graph of performing 3-fold cross-validation to select the optimal depth.



From the figure, we can see the optimal learning rate is 4 with an average MSE of about 0.503. The optimal model will be trained using this depth.

Below are the curves for the test proxy function (first) and the misclassification error rate (second).





We can once again see that the error curves decline sharply until roughly 50 rounds, then continue to steadily decline. This model also took the least number of rounds to train of the models evaluated.

Below are the results of running the model using the trial dataset along with the confidence of the predictions.

```
Actual:    1
Predicted: 1      Confidence: 0.8141534222524006
Actual:    2
Predicted: 2      Confidence: 0.9677721837636055
Actual:    3
Predicted: 3      Confidence: 0.9103231952409419
Actual:    4
Predicted: 9      Confidence: 0.28481163235744217
Actual:    5
Predicted: 5      Confidence: 0.5878644960476207
Actual:    6
Predicted: 6      Confidence: 0.9761102476204461
Actual:    7
Predicted: 7      Confidence: 0.9362983709095744
Actual:    8
Predicted: 8      Confidence: 0.8876768533809973
Actual:    9
Predicted: 9      Confidence: 0.6855157275476815
Actual:    0
Predicted: 0      Confidence: 0.8086707509492845
```

This model almost had perfect accuracy, where it mistook a 4 for a 9 (which makes sense since they have roughly the same shape). The confidence of the misclassified prediction is also incredibly low, which is consistent with the previous models.

[4] Conclusion

Each model performed well on classifying handwritten digits, and the training of each showed steady declines in error implying that gradient descent was a good optimizer.

The first model was very simple and had very good accuracy.

The second model was the most complex and took the longest to train, but its confidence in its predictions were also the highest.

The third model had a single misclassification error on the trial set, but it took the least number of rounds to train.