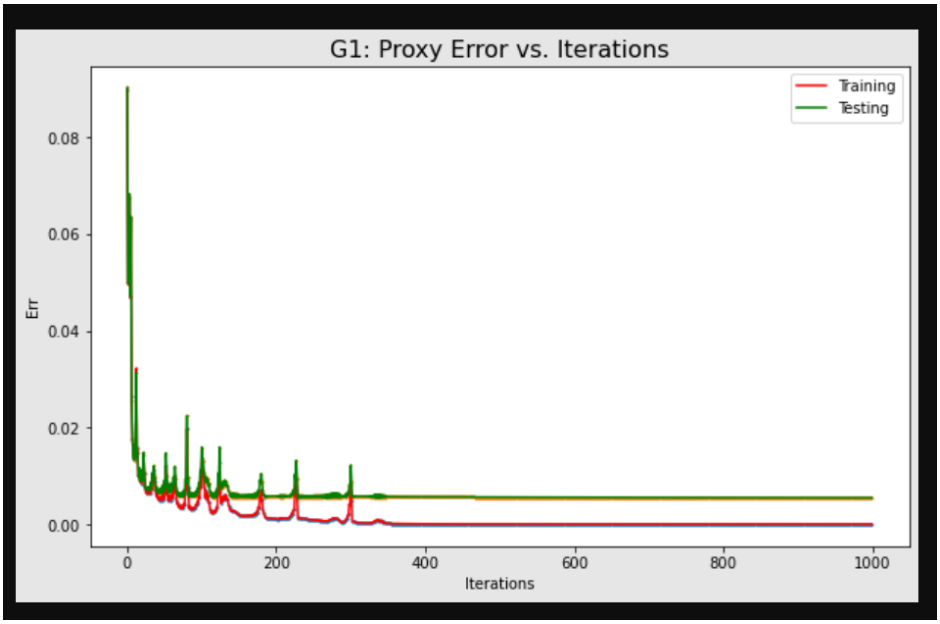
01: A (multi-output) perceptron

```
In [46]: best_nnet01 = best_nnet
         err_train_01, err_rate_train_01 = err_train, err_rate_train
         err_test_01, err_rate_test_01 = err_test, err_rate_test
         print("Proxy")
         print("(train, test)")
         print(err_train_01, err_test_01)
         print("Misclassification")
         print("(train, test)")
         print(err_rate_train_01, err_rate_test_01)
```

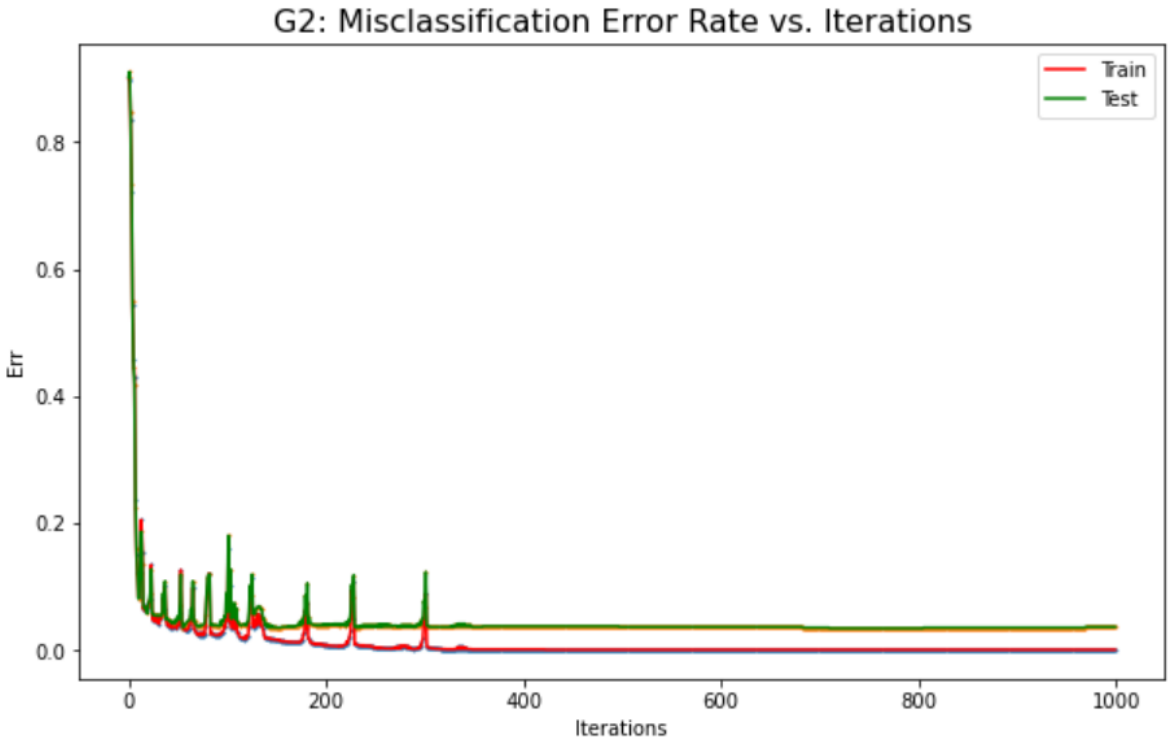
```
Proxy
(train, test)
2.4213598657037004e-06 0.005462738644109837
Misclassification
(train, test)
0.0 0.03594080338266381
```

From the above image, it can be seen that for best model of multi-output perceptron with learning rate=16, the proxy train and test errors are ~2.421e-06 and ~0.005 respectively and the misclassification errors are ~0.0 and ~0.0359 respectively for training and test dataset. These proxy errors indicates that the classifier/model is good at classifying the images as it has very small error values. The small misclassification errors also indicate that the classifier/model is good at correctly classifying most of the samples on the train and test dataset and there are very few samples that are misclassified.



From the graph above, it can be seen that for best model of multi-output perceptron with learning rate=16, the proxy errors of the training and testing dataset is high in the initial iterations and as

training progress the graphs shows the decline in the errors indicating that the difference between the predicted values and true values is very low. Similar trend can be seen for the testing proxy error. Also, the small distance between training and testing errors indicates that the predictions are accurate and model fits well. The errors almost become constant as somewhere near the iteration number 400 which also indicates that the training of the perceptron can be stopped at the iteration 400 or somewhat less. These errors are calculated for maximum iterations calculated.



From the graph above, it can be seen that for best model of multi-output perceptron with learning rate=16, the misclassification errors and its change of the training and testing dataset is high in the initial iterations and as training progress the graphs shows the decline in the errors indicating that the difference between the predicted values and true values is very low. Similar trend can be seen for the testing proxy error. Also, the small distance between training and testing errors indicates that the predictions are accurate and model fits well. The errors almost become constant as somewhere near the iteration number 400 which also indicates that the training of the perceptron can be stopped at the iteration 400 or somewhat less. These errors are calculated for maximum iterations calculated.

```
In [39]: output = best_nnet.forwardprop(X_trial.T)
         np.argmax(output,axis=0)

Out[39]: array([1, 2, 3, 4, 5, 6, 7, 8, 9, 0], dtype=int64)
```

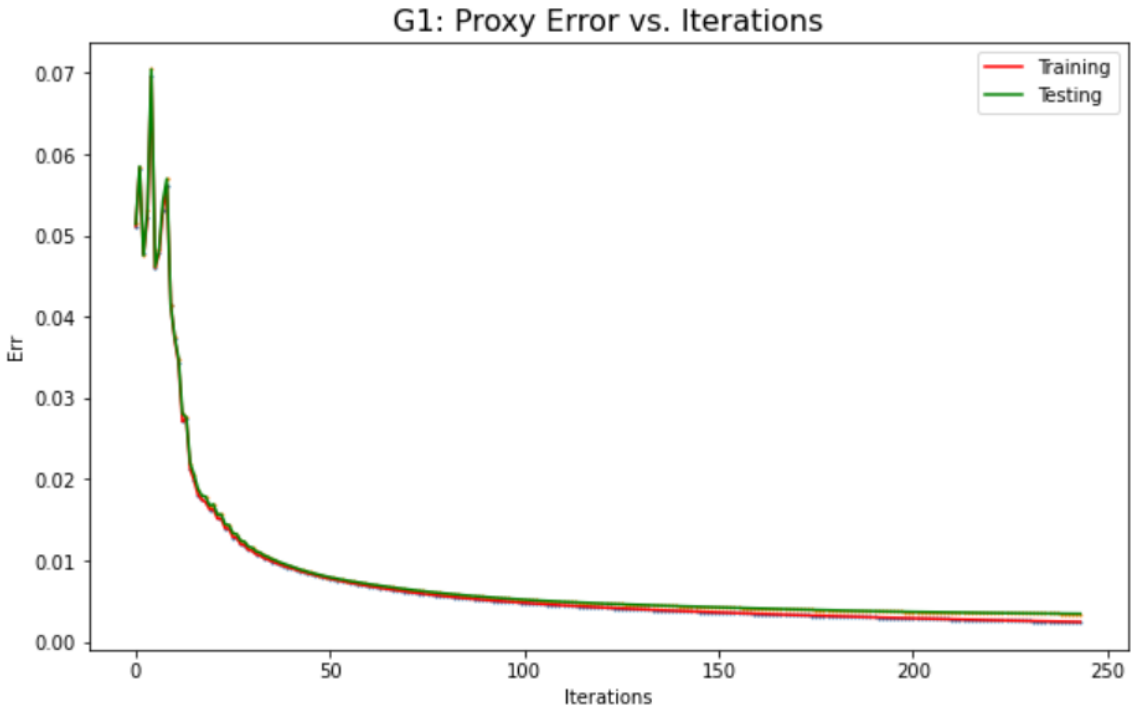
From the image above, it can be seen that for best model of multi-output perceptron with learning rate=4, the model exactly predicts the output labels of the trail dataset as same as the true labels. This indicates the that model has trained well.

03. A deep neural network with 2 hidden layers:

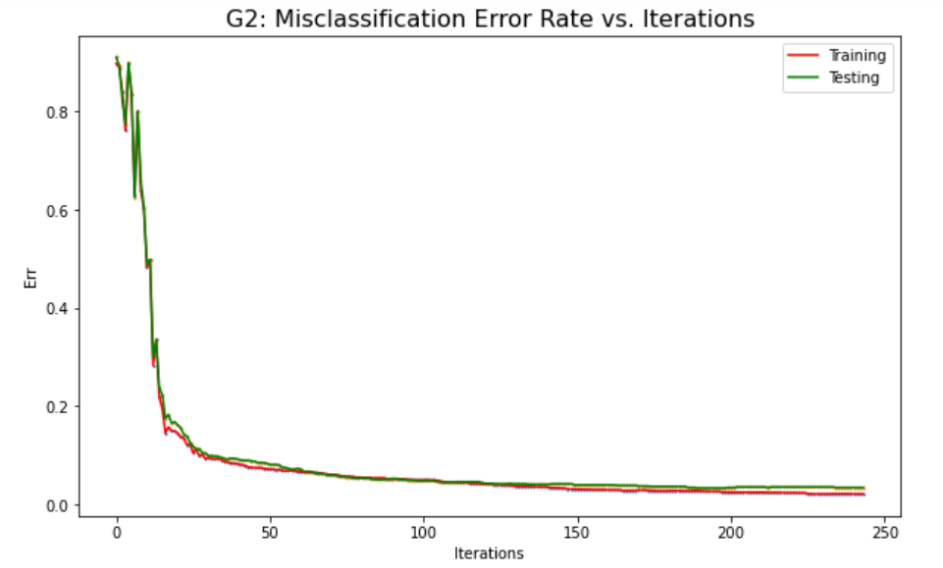
```
In [73]: print('Proxy')
         print('(train, test)')
         print(err_train_03, err_test_03)
         print('Misclassification')
         print('(train, test)')
         print(err_rate_train_03, err_rate_test_03)

Proxy
(train, test)
0.002456404260279807 0.0034602765819429795
Misclassification
(train, test)
0.02171664943123064 0.03488372093023251
```

From the above image, it can be seen that for best model of deep learning with 2 layers with 1st hidden layer units: 4^4, 2nd hidden layer units: 4^3 and learning rate=4, the proxy train and test errors are ~0.00245 and ~0.00346 respectively and the misclassification errors are ~0.0217 and ~0.0348 respectively for training and test dataset. These proxy errors indicates that the classifier/model is good at classifying the images as it has very small error values. The small misclassification errors also indicate that the classifier/model is good at correctly classifying most of the samples on the train and test dataset and there are very few samples that are misclassified.



From the graph above, it can be seen that for best model of deep learning with 2 layers with 1st hidden layer units: 4^4, 2nd hidden layer units: 4^3 and learning rate=4, the proxy errors of the training and testing dataset is high in the initial iterations and as training progress the graphs shows the decline in the errors indicating that the difference between the predicted values and true values is very low. Similar trend can be seen for the testing proxy error. Also, the small distance between training and testing errors indicates that the predictions are accurate and model fits well. These errors are calculated for maximum iterations calculated.



From the graph above, it can be seen that for best model of deep learning with 2 layers with 1st hidden layer units: 4^4, 2nd hidden layer units: 4^3 and learning rate=4, the misclassification error and its change also follows the down trend just as the proxy errors on both training and test dataset. This helps us to understand that as the training progresses the model is learning better and there are very few instances that are misclassified. Also the small distance between training and testing errors indicates that the predictions are accurate and model fits well. These errors are calculated for maximum iterations calculated.

```
In [76]:  output = best_nnet.forwardprop(X_trial.T)
          np.argmax(output,axis=0)

Out[76]: array([1, 2, 3, 9, 5, 6, 7, 8, 9, 0], dtype=int64)
```

From the image above, it can be seen that for best model of deep learning with 2 layers with 1st hidden layer units: 4^4, 2nd hidden layer units: 4^3 and learning rate=4, the model predicts the output labels of the trail dataset with the exception of the 4th label whose true value is 4 but the predicted value is 9.

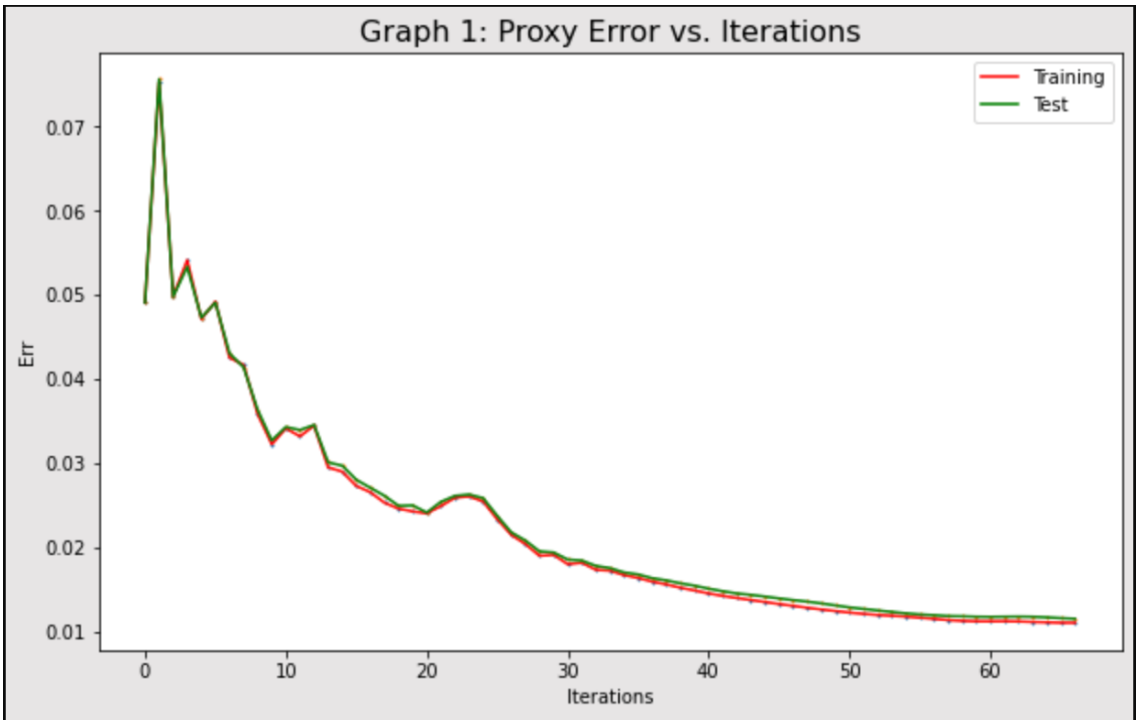
02. A deep neural nework with the same number of units per hidden layer

```
In [67]: print('Proxy')
         print('(train, test)')
         print(err_train, err_test)
         print('Misclassification')
         print('(train, test)')
         print(err_rate_train, err_rate_test)

Proxy
(train, test)
0.011003112366012459 0.011552640934699557
Misclassification
(train, test)
0.11427094105480873 0.13319238900634245
```

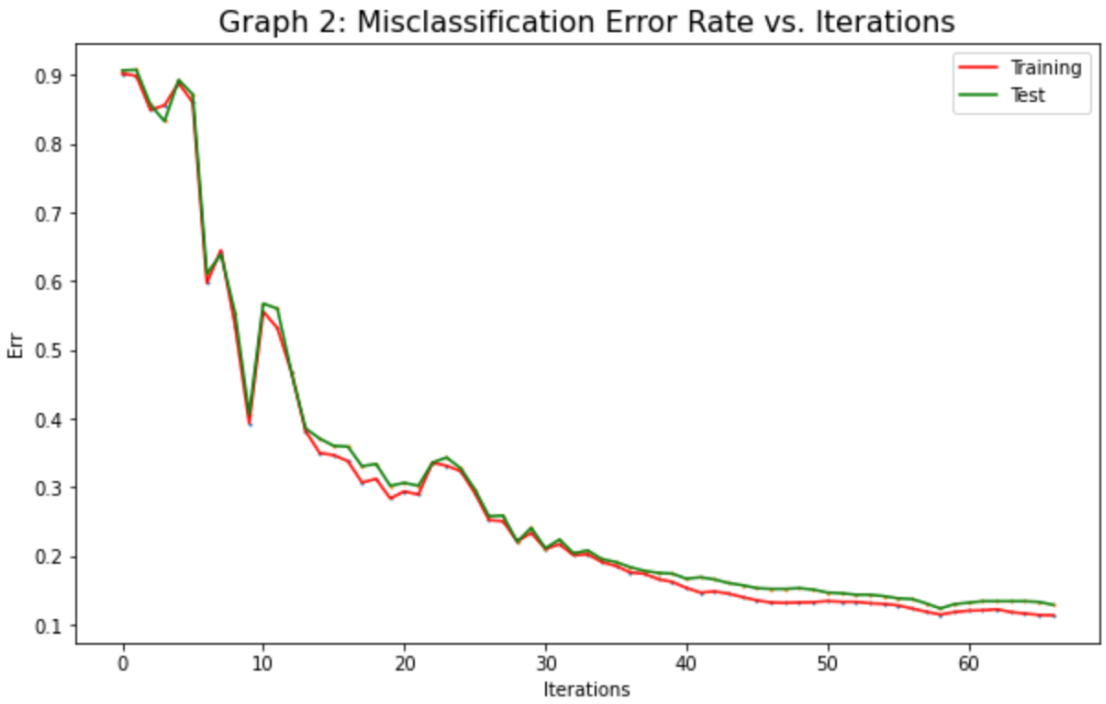
From the above image, it can be seen that for best model of deep learning with layer with 1st hidden layer units: 4^2, and learning rate=4, the proxy train and test errors are ~0.011 and ~0.0115 respectively and the misclassification errors are ~0.1142 and ~0.1331 respectively for training and test dataset. These proxy errors indicates that the classifier/model is good at classifying the images as it has very small error values. The small misclassification errors also

indicate that the classifier/model is good at correctly classifying most of the samples on the train and test dataset and there are very few samples that are misclassified.



From the graph above, it can be seen that for best model of deep learning with layer with 1st hidden layer units: 4^2, and learning rate=4, the proxy errors of the training and testing dataset is high in the initial iterations and as training progress the graphs shows the decline in the errors indicating that the difference between the predicted values and true values is very low. Similar trend can be seen for the testing proxy error. Also, the small distance between training and testing errors indicates that the predictions are accurate and model fits well. These errors are calculated for maximum iterations calculated.

[Here I have trained the model using default values of change_err_thresh and change_thresh and not 0. Thus as training progresses, the iterations are 60.]



From the graph above, it can be seen that for best model of deep learning with layer with 1st hidden layer units: 4^2, and learning rate=4, the misclassification error and its change also follows the down trend just as the proxy errors on both training and test dataset. This helps us to understand that as the training progresses the model is learning better and there are very few instances that are misclassified. Also, the small distance between training and testing errors indicates that the predictions are accurate and model fits well. These errors are calculated for maximum iterations calculated.

```
In [59]: output = best_nnet.forwardprop(X_trial.T)
         np.argmax(output,axis=0)

Out[59]: array([1, 2, 3, 4, 5, 6, 7, 8, 9, 0])
```

From the graph above, it can be seen that for best model of deep learning with layer with 1st hidden layer units: 4^2, and learning rate=4, the model exactly predicts the output labels of the trail dataset as same as the true labels. This indicates the that model has trained well.