Task 1:

Loading the input data and pre-processing:

i). Train files are read to data frame and divided into training set and validation set.

No. of samples in training set X_train = 3000 (1500 from Class 1 + 1500 from Class 2)

No. of samples in validation set X_val = 1000 (500 from Class 1 + 500 from Class 2)

ii). True class labels y_train and y_val are created for the samples with 0 for Class 1 samples and 1 for Class 2 samples.

iii). Test files are read in the similar fashion with true class labels created (X_test, y_test).

iv). The train, validation and test datasets are randomly shuffled to avoid any error caused by the bias.

Normalizing the data:

The train, validation and test data are normalized before training the model by calculating the mean and variance.

Initializing the model parameters:

A 2-n_H-1 MLP will have input layer with 2 nodes and 1 output layer. The value of n_H is varied among 2, 4, 6, 8, 10 to train the model.

This MLP required 2 weight matrices W1, W2 and 2 bias terms b1 and b2. (W1 and b1 between input-hidden layers and one W2 and b2 between hidden-output layers). These weights are randomly initialized, and bias terms are initialized to 0.

Dimensions:

W1: (input_dim x hidden_dim)

W2: (hidden_dim x output_dim)

b1: 1 x hidden_dim

b2: 1 x output_dim

where input_dim = no. of nodes in input layer, hidden_dim = no. of nodes in hidden layer, output_dim = no. of nodes in output layer

Training phase:

Stop Criteria: The model should be trained until the error in validation set does not decrease any longer. Due to this, the model is observed to converge in the local minima, so it is also trained using a finite number of epochs for each n_H to plotting the learning curve. Following are the hyper parameter values used for batch training.

Number of epochs = 4000, Learning rate = 0.7, Activation function = sigmoid, momentum = 0.05

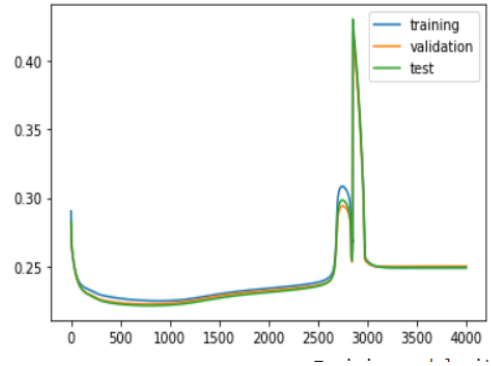
Steps in training:

1. In forward propagation step, the outputs of hidden layer and the output layer are computed using sigmoid activation function and the values are stored in cache.
2. In backward propagation, the gradients dW1, db1, dW2, db2 are computed.
3. In the next step, the weights are updated using the learning rate and the gradients computed in backward propagation step. Momentum is also used to reflect the previous epoch's gradient in the weights update of the current epoch.
4. The prediction output of test and validation sets are obtained by forward propagation step at every epoch.
5. Mean Square error is calculated for the training set, test set and validation set and saved to plot the learning curve.
6. The accuracy score of the test data is also saved for each epoch and the average of accuracy scores for each value of n_H is reported.

Observations: As the number of epochs increases the validation and test error reduced for all other n_H values, except when n_H = 2.

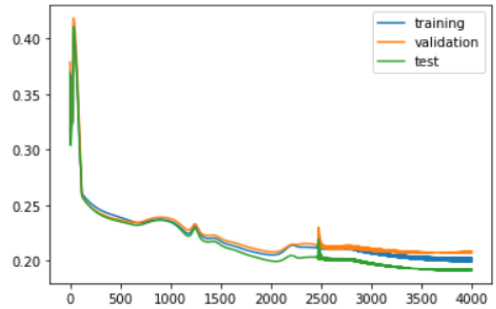
Learning curve for n_H = 2:

```
Test accuracy 66.53088544138751
Validation error at which the training ends 0.2501358888525168
Minimum validation error occured at epoch 776
Minimum test error occured at epoch 802
Minimum train error occured at epoch 888
```



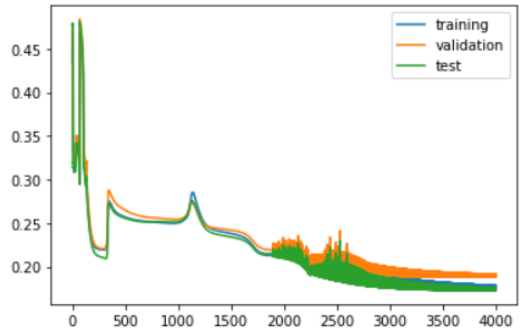
Learning curve for n_H = 4:

```
Test accuracy 66.88304552315988
Validation error at which the training ends 0.2081729743949217
Minimum validation error occured at epoch 3784
Minimum test error occured at epoch 3944
Minimum train error occured at epoch 3999
```



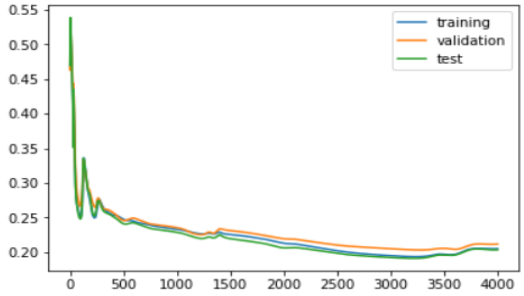
Learning curve for n_H = 6:

```
Test accuracy 66.84301392639459
Validation error at which the training ends 0.19117280550125057
Minimum validation error occured at epoch 3998
Minimum test error occured at epoch 3998
Minimum train error occured at epoch 3999
```



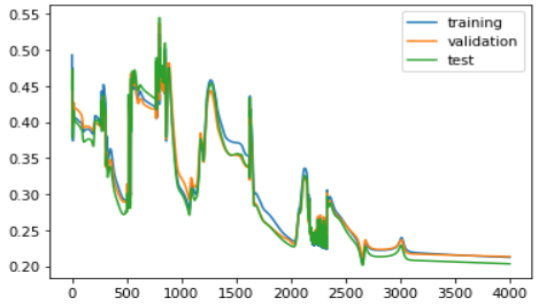
Learning curve for n_H = 8:

```
Test accuracy 66.8273796902934
Validation error at which the training ends 0.2115092109217838
Minimum validation error occured at epoch 3276
Minimum test error occured at epoch 3215
Minimum train error occured at epoch 3232
```



Learning curve for n_H = 10:

```
Test accuracy 66.71950375165011
Validation error at which the training ends 0.2132014888510633
Minimum validation error occured at epoch 2655
Minimum test error occured at epoch 2655
Minimum train error occured at epoch 2654
```



Accuracy:

The MLP with n_H = 4 gives the best classification accuracy for testing set with an average accuracy score of 66.88304552315988.

Task 2:

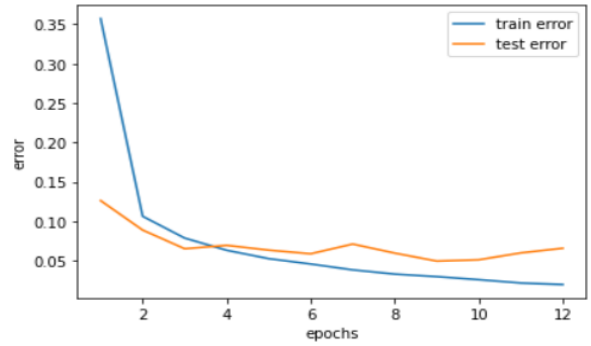
The given baseline code implementing Convolution Neural Networks is run with MNIST train and test datasets.

The test accuracy of the baseline: 0.9890000291824341

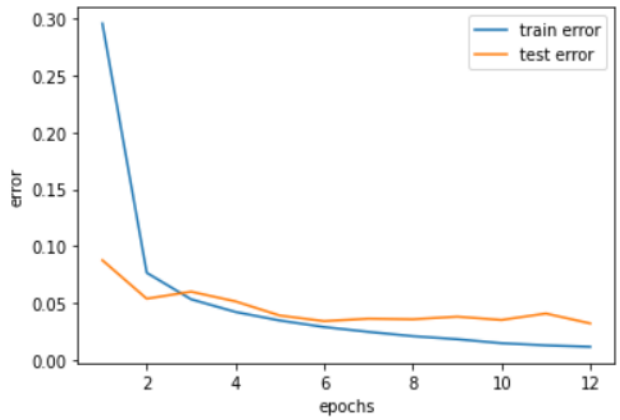
Observations:

1.Kernel Size in first convolution layer: The following is observed by changing the kernel size in the first hidden layer, which is convolutional layer.

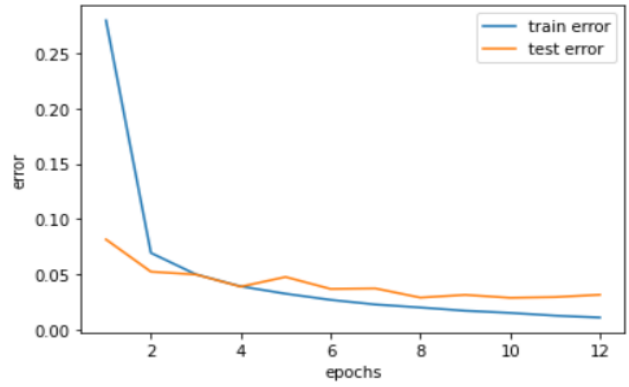
Test accuracy for kernel Size: (1x1) is 0.9811000227928162



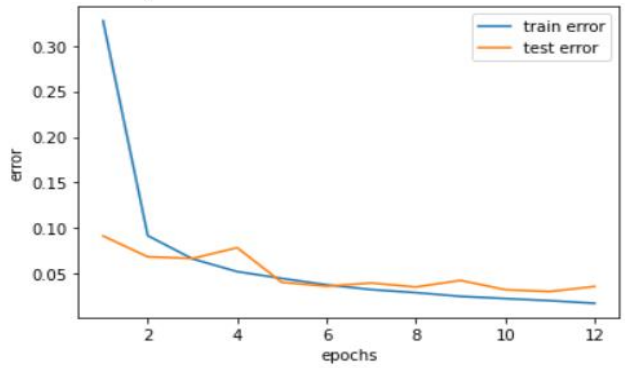
Test accuracy for Kernel Size (baseline) (3x3) is 0.9890000291824341



Test accuracy for Kernel Size: (5x5) is 0.9890999794006348



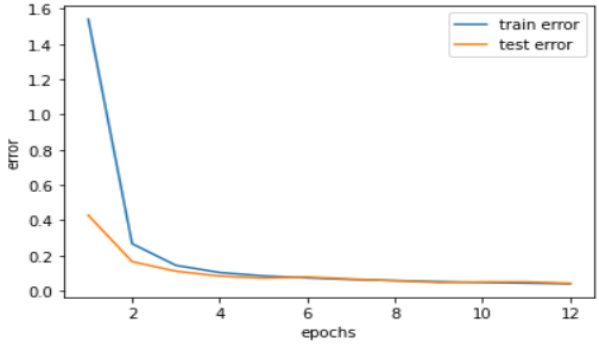
Test accuracy for Kernel Size: (7x7) is 0.989300012588501



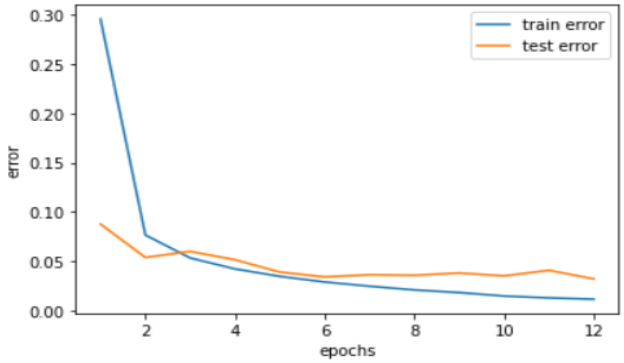
This kernel size should have odd dimensions. As the dimensions of the kernel size increases, the test accuracy score increases as we can see in the above figure. If the kernel size decreases, the accuracy decreased. As shown in above figures, the kernel size of (3x3) gives a better accuracy than (1x1). However, the test accuracy does not vary much in all the four cases.

2.Activation Function: The following changes are observed by updating the activation function in the two convolution layers with kernel size (3x3).

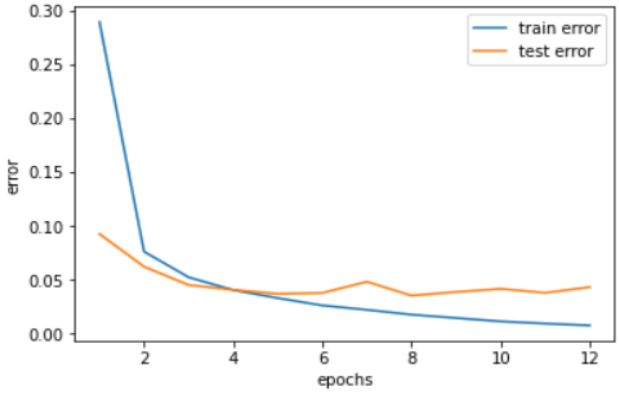
Sigmoid: Test loss: 0.04251803457736969, Test accuracy: 0.9865000247955322



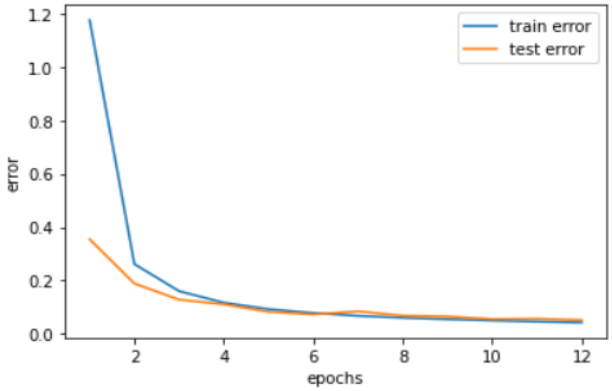
RELU: Test loss: 0.03570752963423729, Test accuracy: 0.9894000291824341



tanh: Test loss: 0.04309143126010895, Test accuracy: 0.9884999990463257



Softmax: Test loss: 0.05121622234582901, Test accuracy: 0.983299970626831

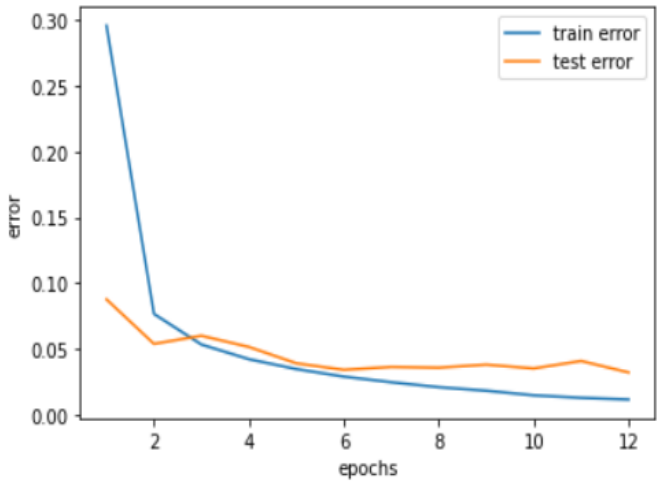


Although the test accuracies do not differ much between different activation functions, the graphs showing the errors has a significant difference. Out of all the activation functions, the test loss is lower for RELU function. In the learning curves, we can see that when sigmoid and softmax activation functions are used in Convolution layer the test error gradually declines and approaches close to the train error, hence converging to the global minima.

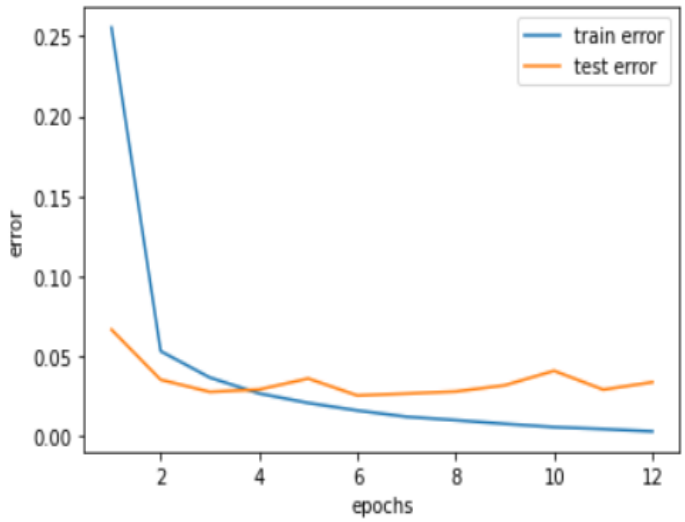
3.Feature map size in Convolution Layers:

The following graph shows the learning curve of CNN, when feature map size for a (3x3) kernel size in both Convolution layers are changed.

- Feature map size of 6 in first layer and 16 in second layer:
  Test loss: 0.03570752963423729
  Test accuracy: 0.9894000291824341



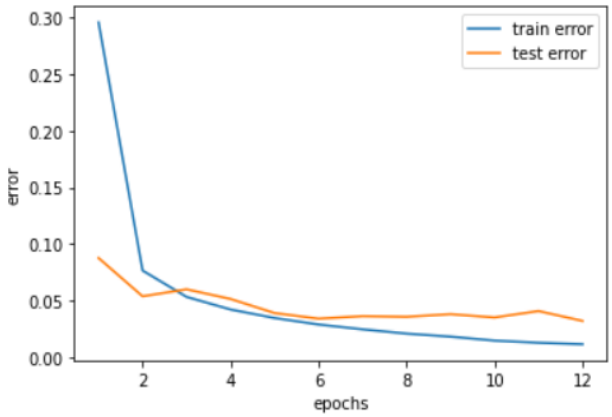
- With feature map size of 32 in first layer and 64 in second layer,
  Test loss: 0.0338655561208725
  Test accuracy: 0.9916999936103821



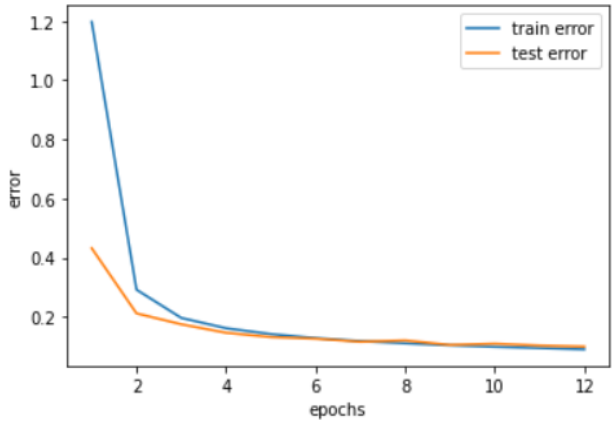
From the above graph it can be concluded that the accuracy of the testing increases with the increase in feature map size. If feature map size increases, it means that the number of filters in convolution layer increases. The graph indicates that the convolutional neural network detects more number of features as the feature map size increases, hence the testing error reduced.

4.Nodes in first and second fully Connected Layers

- With 120 number of nodes in first layer and 84 nodes in second layer
  Test loss: 0.03570752963423729
  Test accuracy: 0.9894000291824341



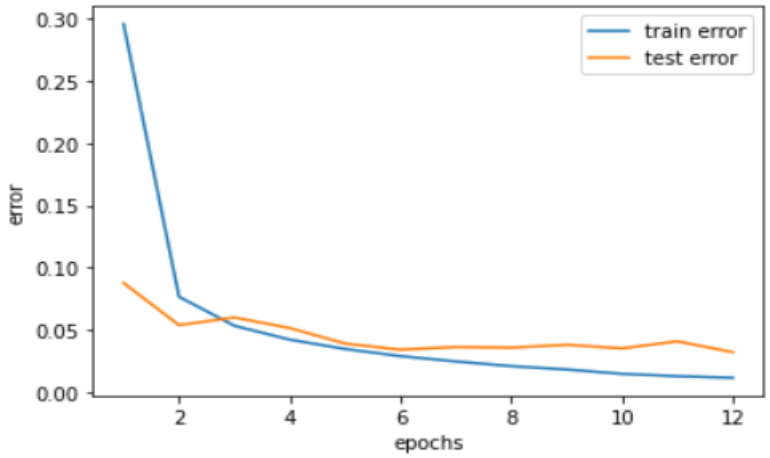
- With 6 number of nodes in first layer and 4 nodes in second layer
  Test loss: 0.09926041960716248
  Test accuracy: 0.972599983215332



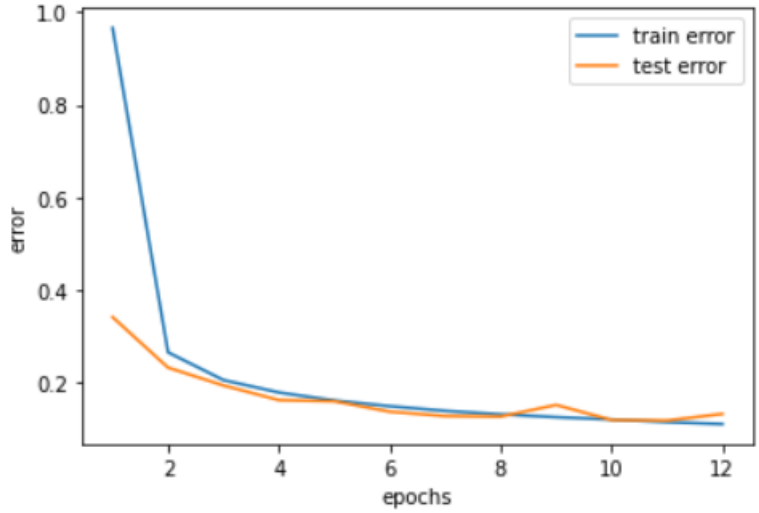
The decrease in the number of neurons in fully connected dense layer affects the output shape. Results show that the decrease in the number of nodes in fully connected layers leads to less test accuracy. Although the test accuracy has not changed by a large amount but reducing the nodes would certainly decrease the computational overhead.

5.Pooling size in both the Max pooling layers

- With pooling size of (2,2) in max pool layers
  Test loss: 0.03570752963423729
  Test accuracy: 0.9894000291824341



- With pooling size of (4,4) in max pool layers
  Test loss: 0.1322983056306839
  Test accuracy: 0.9616000056266785



As shown in the above figures, the increase in the pooling size, from (2,2) to (4,4) lead to poor performance of the CNN model. So, the pooling filters of dimensions 2x2 is well common and well suited for the model.