```
% ASSIGNMENT 4
% Akash Rout (21103080)
```

```
% Question 1
```

```
% Load IRIS dataset
data=readtable('Iris.csv');
```

```
X=table2array(data(:, 2:5)); % features
y=grp2idx(data{:, 6}) % label
```

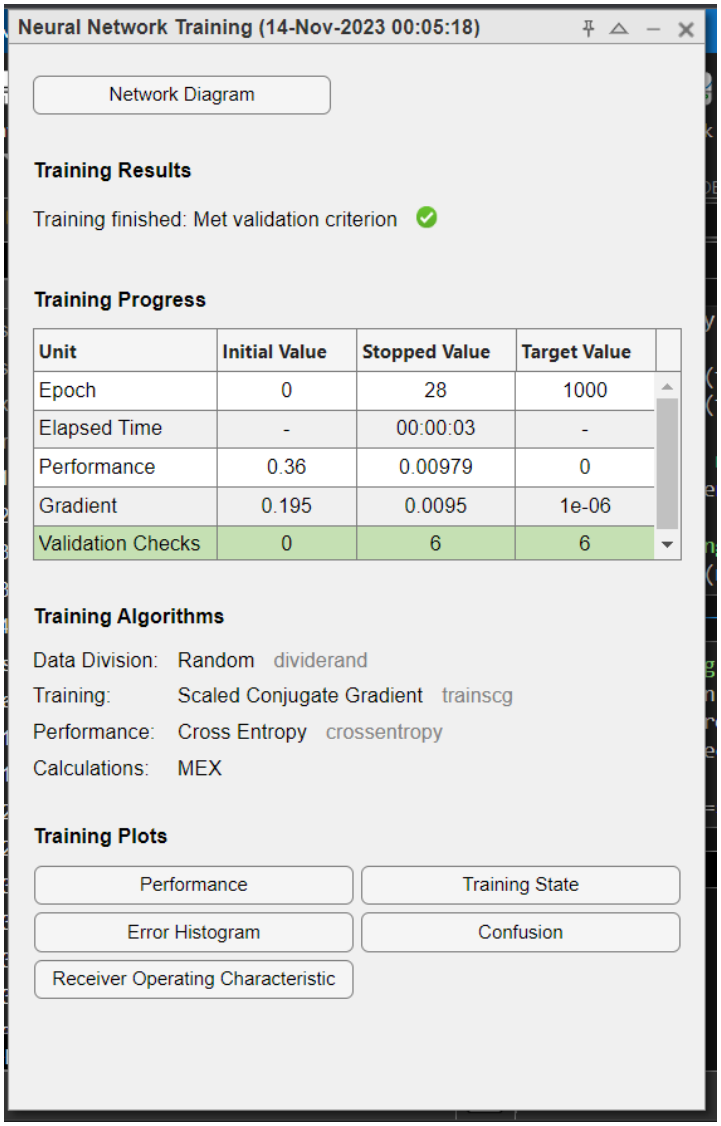
```
y = 150×1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     ⋮
```

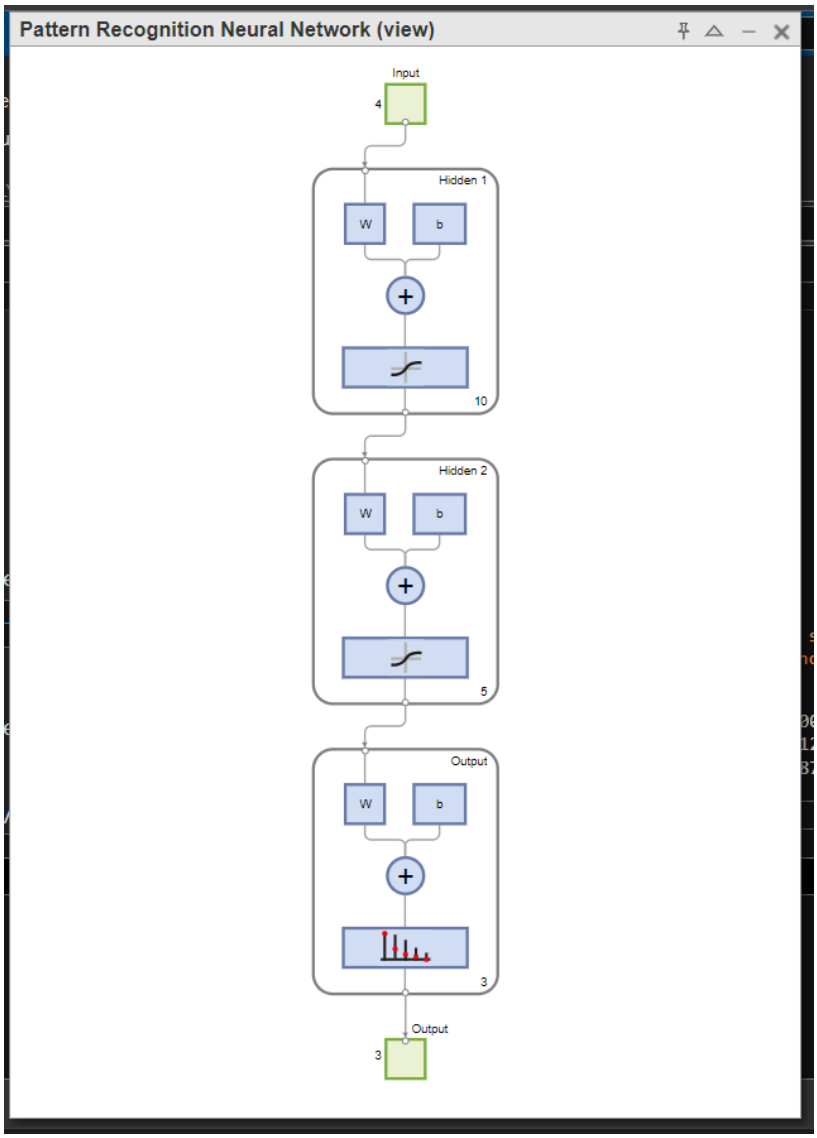
```
% train test split
rng(42);
ratio=0.8;
idx=randperm(size(X, 1));
trainIdx=idx(1: round(ratio*length(idx)));
testIdx=idx(round(ratio*length(idx))+1:end);

X_train=X(trainIdx, : );
y_train=y(trainIdx);

X_test=X(testIdx, :);
y_test=y(testIdx);

% neural network
net=patternnet([10 5]);

% training
nn=train(net, X_train', ind2vec(y_train'));
```

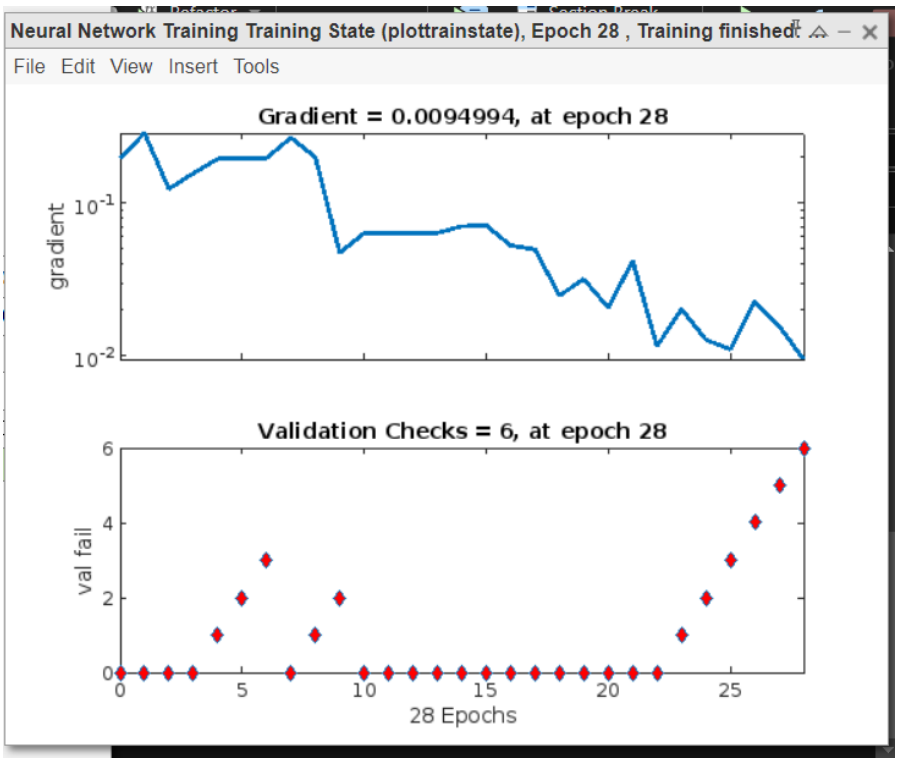
```
Warning: Converting sparse inputs to full for training. This can cause a large increase in memory
usage.
```

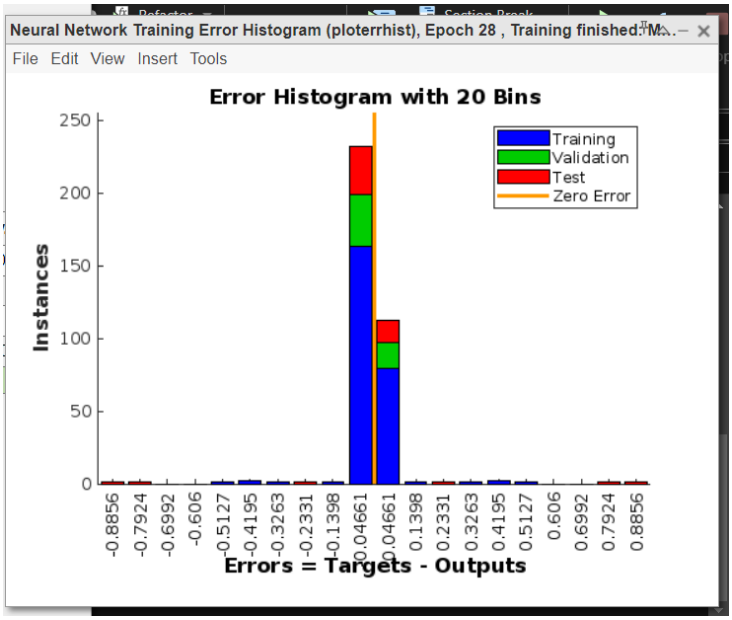
```
% testing
y_pred=nn(X_test')
```

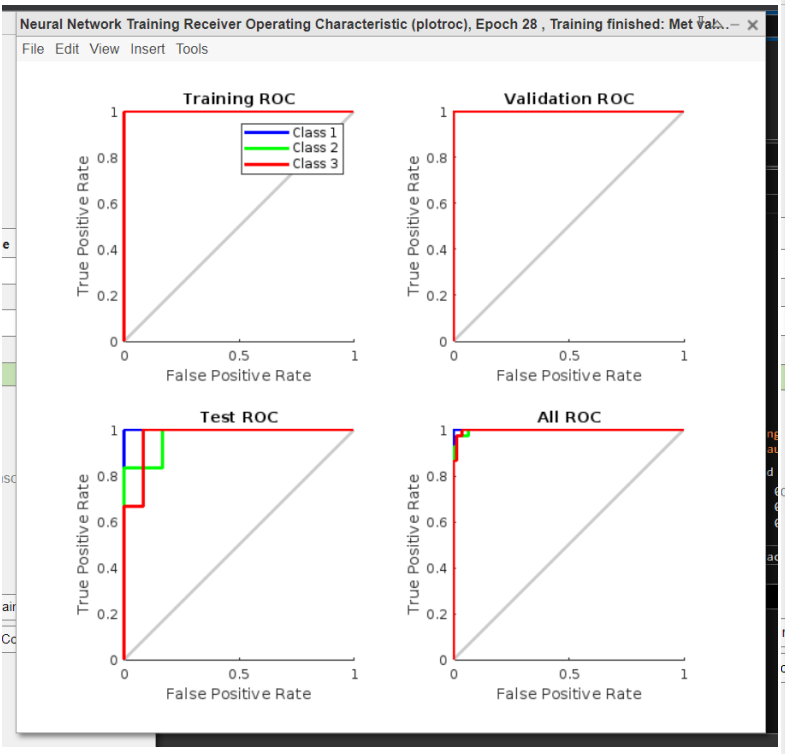
Neural Network Toolbox

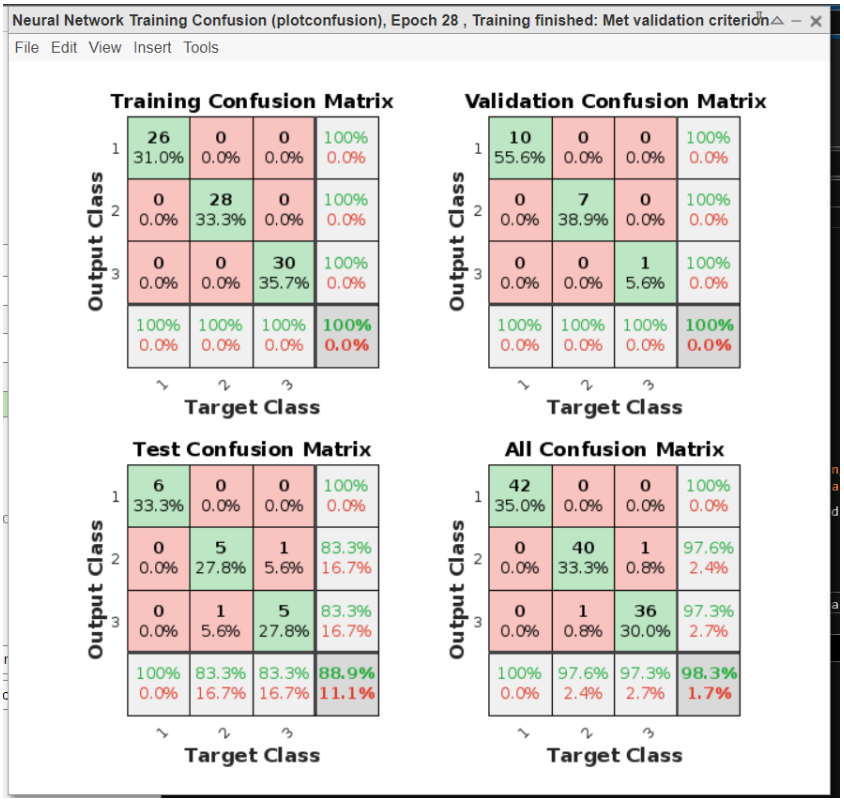


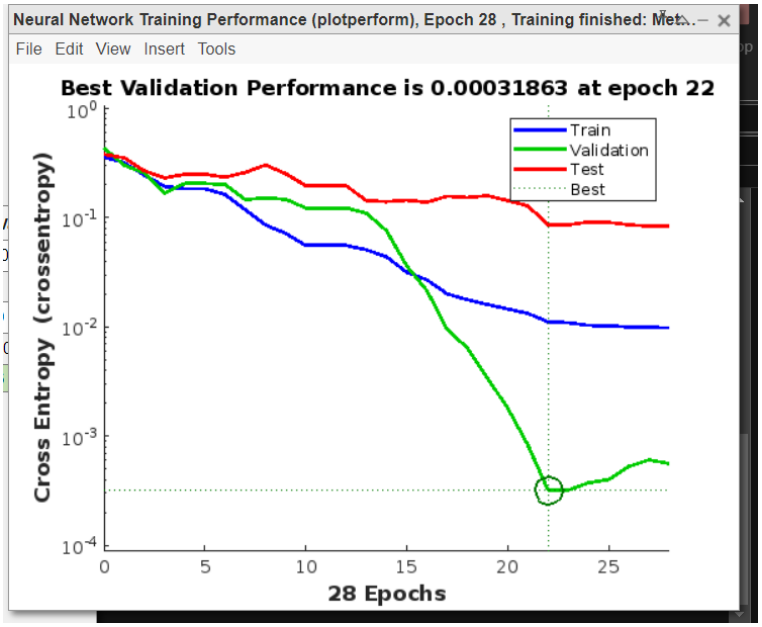





Neural Network Training Error Histogram (ploterrhist), Epoch 28 , Training finished: M...
File Edit View Insert Tools
Error Histogram with 20 Bins
Training
Validation
Test
Zero Error
Instances
250
200
150
100
50
0
-0.8856
-0.7924
-0.6992
-0.606
-0.5127
-0.4195
-0.3263
-0.2331
-0.1398
-0.04661
0.04661
0.1398
0.2331
0.3263
0.4195
0.5127
0.606
0.6992
0.7924
0.8856
Errors = Targets - Outputs


Neural Network Training Receiver Operating Characteristic (plotroc), Epoch 28 , Training finished: Met val...
File Edit View Insert Tools
Training ROC
Validation ROC
Test ROC
All ROC
Class 1
Class 2
Class 3
True Positive Rate
False Positive Rate
0
0.2
0.4
0.6
0.8
1
0.5


Neural Network Training Confusion (plotconfusion), Epoch 28 , Training finished: Met validation criterion
File Edit View Insert Tools
Training Confusion Matrix
Validation Confusion Matrix
Test Confusion Matrix
All Confusion Matrix
Output Class
Target Class
26 31.0%
0 0.0%
0 0.0%
100% 0.0%
0 0.0%
28 33.3%
0 0.0%
100% 0.0%
0 0.0%
0 0.0%
30 35.7%
100% 0.0%
100% 0.0%
100% 0.0%
100% 0.0%
100% 0.0%
10 55.6%
0 0.0%
0 0.0%
100% 0.0%
0 0.0%
7 38.9%
0 0.0%
100% 0.0%
0 0.0%
0 0.0%
1 5.6%
100% 0.0%
100% 0.0%
100% 0.0%
100% 0.0%
100% 0.0%
6 33.3%
0 0.0%
0 0.0%
100% 0.0%
0 0.0%
5 27.8%
1 5.6%
83.3% 16.7%
0 0.0%
1 5.6%
5 27.8%
83.3% 16.7%
100% 0.0%
83.3% 16.7%
83.3% 16.7%
88.9% 11.1%
42 35.0%
0 0.0%
0 0.0%
100% 0.0%
0 0.0%
40 33.3%
1 0.8%
97.6% 2.4%
0 0.0%
1 0.8%
36 30.0%
97.3% 2.7%
100% 0.0%
97.6% 2.4%
97.3% 2.7%
98.3% 1.7%


Neural Network Training Performance (plotperform), Epoch 28 , Training finished: Met...
File Edit View Insert Tools
Best Validation Performance is 0.00031863 at epoch 22
Train
Validation
Test
Best
Cross Entropy (crossentropy)
28 Epochs

```
y_pred = 3×30
    0.0000    0.0000    0.0000    0.9996    0.0001    0.0000    0.0001    0.9996 ···
    0.0173    0.0129    0.0129    0.0004    0.9998    0.1812    0.9999    0.0004
    0.9827    0.9871    0.9871    0.0000    0.0001    0.8188    0.0001    0.0000
```

```
y_pred_prob=exp(y_pred)./sum(exp(y_pred), 2);
y_pred=vec2ind(y_pred_prob);

accuracy=sum(y_pred==y_test')/length(y_test)
```

```
accuracy = 1
```

```
% Question 2
```

```
% Loading MNIST dataset
data = readtable('MNIST.csv');

% Feature Extraction
X = table2array(data(:, 2:end));
Y = categorical(data.label);

% Pixel Normalization
X = X / 255;

% Train Test Split
rng(42);
idx = randperm(size(X, 1), round(0.8 * size(X, 1))); % 80% for training, 20%
for testing

trainData = X(idx, :);
trainLabels = Y(idx, :);

testData = X(setdiff(1:size(X, 1), idx), :);
testLabels = Y(setdiff(1:size(X, 1), idx), :);

% Convert data to the format expected by the neural network
trainData = reshape(trainData', [28, 28, 1, numel(trainLabels)]);
testData = reshape(testData', [28, 28, 1, numel(testLabels)]);

% neural network architecture
layers = [
    imageInputLayer([28 28 1])
    convolution2dLayer(3, 32, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2, 'Stride', 2)
    convolution2dLayer(3, 64, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2, 'Stride', 2)
```

```
    fullyConnectedLayer(128)
    reluLayer
    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer];

% training options
options = trainingOptions('adam', ...
    'MaxEpochs', 10, ...
    'MiniBatchSize', 128, ...
    'ValidationData', {testData, testLabels}, ...
    'Plots', 'training-progress', ...
    'Verbose', false);

% training
net = trainNetwork(trainData, trainLabels, layers, options);

% testing
predictedLabels = classify(net, testData);
accuracy = sum(predictedLabels == testLabels) / numel(testLabels);
fprintf('Test Accuracy: %.2f%%\n', accuracy * 100);
```

```
Test Accuracy: 97.43%
```

Neural Network Toolbox