```
%I will not include LoadMNIST.m
%--------------------Network1.m ------------------------

[xTrain, tTrain, xValid, tValid, xTest, tTest] = LoadMNIST(3);

options = trainingOptions('sgdm', ...
    'Momentum',0.9,...
    'InitialLearnRate', 0.00100,...
    'MaxEpochs',60, ...
    'MiniBatchSize',8192, ...
    'ValidationData',{xValid,tValid}, ...
    'ValidationPatience',5 ,...
    'ValidationFrequency', 30,...
    'Shuffle', 'every-epoch',...
    'Plots','training-progress');

layers = [
    imageInputLayer([28 28 1])
    convolution2dLayer(5,20,'Padding',1, ...
    'WeightsInitializer','narrow-normal')
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    fullyConnectedLayer(100,'WeightsInitializer', 'narrow-normal')
    reluLayer
    fullyConnectedLayer(10, 'WeightsInitializer', 'narrow-normal')
    softmaxLayer
    classificationLayer];

net1 = trainNetwork(xTrain, tTrain, layers, options);

trainPred = classify(net1,xTrain);
validPred = classify(net1,xValid);
testPred = classify(net1,xTest);

cErrorTrain = ClassificationError(tTrain, trainPred);
cErrorVaild = ClassificationError(tValid, validPred);
cErrorTest = ClassificationError(tTest, testPred);

cErrorTrain
cErrorVaild
cErrorTest

%--------------------Network1.m  END --------------------

%--------------------Network2.m ------------------------

[xTrain, tTrain, xValid, tValid, xTest, tTest] = LoadMNIST(3);

options = trainingOptions('sgdm', ...
    'Momentum',0.9,...
    'InitialLearnRate', 0.0100,...
    'MaxEpochs',30, ...
```

```
    'MiniBatchSize',8192, ...
    'ValidationData',{xValid,tValid}, ...
    'ValidationPatience',5 ,...
    'ValidationFrequency', 30,...
    'Shuffle', 'every-epoch',...
    'Plots','training-progress');

layers = [
    imageInputLayer([28 28 1])

    convolution2dLayer(3,20,'Padding',1, ...
    'WeightsInitializer','narrow-normal')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,30,'Padding',1, ...
    'WeightsInitializer','narrow-normal')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,50,'Padding',1, ...
    'WeightsInitializer','narrow-normal')
    batchNormalizationLayer
    reluLayer

    fullyConnectedLayer(10,'WeightsInitializer', 'narrow-normal')
    softmaxLayer
    classificationLayer];

net1 = trainNetwork(xTrain, tTrain, layers, options);

trainPred = classify(net1,xTrain);
validPred = classify(net1,xValid);
testPred = classify(net1,xTest);

cErrorTrain = ClassificationError(tTrain, trainPred);
cErrorVaild = ClassificationError(tValid, validPred);
cErrorTest = ClassificationError(tTest, testPred);

cErrorTrain
cErrorVaild
cErrorTest

%--------------------Network2.m  END --------------------

function cError = ClassificationError(targets, outputs)
    sumAccurate = sum(targets==outputs);
    cError = 1 - sumAccurate/size(targets,1);
end
```

Published with MATLAB® R2020a