readSets.m

```
function [TrainSet, TestSet] = readSets(MAX_CLASS, TrainCount,
 TestCount, DATA_SIZE, Noise)
% READSETS Read the data for all classes
% INPT: MAX_CLASS: The # of classes to process
%       TrainCount: 1xC array. The amount of training sample required for
 each class
%       TestCount: 1xC array. The amount of testing sample required for
 each class
%       DATA_SZIE: The dimension of a sample
%       Noise: The magnitude of the variance of a gaussian noise to be
 added onto each pixel
% OUPT: TrainSet: 1xC cell. Each contains the training dataset of each class
%       TestSet: 1xC cell. Each contains the testing dataset of each class

for i = 1 : MAX_CLASS
 % Open the file containing the dataset of a class
 fileName = sprintf('./data/%c.data', i - 1 + 'a');
 fid = fopen(fileName);

    R = 0;
    TrainMatrix = zeros(TrainCount(i), DATA_SIZE);
    TestMatrix = zeros(TestCount(i), DATA_SIZE);

 % Read a sample from a file until we've reach the designated amount of
 samples
    while R < TrainCount(i) + TestCount(i)

        R = R + 1;
        % Read a sample
        tempData = readOneLine(fid, Noise);

  % Check EOL, which should never be evaluated true,
  % unless the TrainCount or TestCount is set incorrectly if
        tempData == -1
            break;
        end

        % Store the first MAX_TRAIN_SIZE data as train set if
        R <= TrainCount(i)
            TrainMatrix(R, :) = tempData;
        else
            TestMatrix(R - TrainCount(i), :) = tempData;
        end
    end

    TrainSet{i} = TrainMatrix;
    TestSet{i} = TestMatrix;
    fclose(fid);
end
end
```

readSets.m

```
function data = readOneLine(fid, Noise)
% READONELINE Read one line/sample from file
    tline = fgets(fid);
    if  tline  ==  -1
        data  =  -1;
        return;
    end
    tline = regexprep(tline, ' ', ''); % Remove whitespace
    data = tline - '0';  % ASCII to Int
    data = data(1:end-1); % Remove newline character
    data = data + Noise*randn(size(data)); % Apply gaussian noise
 noise

 % Limit the data range to 0.001 to 1
 % The bottom limit is due to the need of evaluate ln(x)
    data(data>1) = 1;
    data(data<=0) = 0.001;
end
```