Homework 3

load in data

```
clear
load("dataNew.mat");
noisy_image = double(imread("sampletest.png"));
comparable_image = double(imread("sampletrain.png"));
```

reshape matrices and normalize values

```
reshaped_imageTrain = reshape(imageTrain, 784, 5000) / 255;
reshaped_imageTest = reshape(imageTestNew, 784, 500) / 255;
reshaped_noisyImage = reshape(noisy_image, 784, 1) / 255;
reshaped_comparableImage = reshape(comparable_image, 784, 1) / 255;
```

calculate sample mean

```
sample_mean = zeros(784, 10);
class_length = zeros(1, 10);
for i = 0:9
    current_trainIndexes = find(labelTrain == i);
    class_length(i + 1) = length(current_trainIndexes);
    sample_mean(:, i + 1) = sum(reshaped_imageTrain(:, current_trainIndexes), 2) / class_length
end

N = reshaped_noisyImage - sample_mean(:, 4) .* reshaped_comparableImage;
```

calculate varaince for the sample_images

```
variance = zeros(784, 10);
class_testLength = zeros(1, 10);
for i = 0:9
    current_trainIndexes = find(labelTrain == i);
    current_testIndexes = find(labelTestNew == i);

    class_testLength(i + 1) = length(current_testIndexes);
    current_trainIndexes = current_trainIndexes(1:class_testLength(i+1));

    term_one = class_testLength(i + 1) .* sum((reshaped_imageTrain(:, current_trainIndexes) .^
    term_two = sum(reshaped_imageTrain(:, current_trainIndexes), 2) .^ 2;
    variance(:, i + 1) = term_one - term_two;
end
```

calculate covariance for the sample_images

```
covariance = zeros(784, 10);
for i = 0:9
    current_trainIndexes = find(labelTrain == i);
    current_testIndexes = find(labelTestNew == i);
```

```
    current_trainIndexes = current_trainIndexes(1:class_testLength(i+1));
    term_one = class_testLength(i+1) .* sum((reshaped_imageTrain(:, current_trainIndexes) .* re
    term_two = sum(reshaped_imageTrain(:, current_trainIndexes), 2) .* sum(reshaped_imageTest(:
    covariance(:, i + 1) = term_one - term_two;
end
```

calculate N using alpha and the sample images and then a*

```
N = reshaped_noisyImage - sample_mean(:, 4) .* reshaped_comparableImage;
alpha_star = sum(N .^ 2)
```

```
alpha_star = 18.5917
```

least squares distance classifier

```
least_SquaresDistances = inv(transpose(sample_mean) * sample_mean) * transpose(sample_mean) * r
[throw_away, predicted_labels] = min(least_SquaresDistances);
```

errors

```
given_classError = zeros(1, 10);
for i = 0:9
    given_Index = find(labelTestNew == i);
    given_Size = length(given_Index);
    pruned_labelTest = labelTestNew(given_Index);
    pruned_predictedLabels = transpose(predicted_labels(given_Index));
    given_classError(i + 1) = length(nonzeros(pruned_labelTest - pruned_predictedLabels)) / giv
end

total_Error = length(nonzeros(labelTestNew - transpose(predicted_labels))) / 500;
```

nearest neighbors classifier

```
predicted_labels = zeros(1, 500);
min_argIndexes = zeros(1, 500);

for i = 1:500
    compare_imageTest = repmat(reshaped_imageTest(:, i), 1, 5000);
    norm = sqrt(sum((compare_imageTest - reshaped_imageTrain) .^ 2));
    min_argIndex = find(norm == min(norm));
    min_argIndexes(i) = min_argIndex;
    predicted_labels(i) = labelTrain(min_argIndex);
end
```

Error given claass

```
given_NNclassError = zeros(1, 10);
```

```
for i = 0:9
    given_Index = find(labelTestNew == i);
    given_Size = length(given_Index);
    pruned_labelTest = labelTestNew(given_Index);
    pruned_predictedLabels = transpose(predicted_labels(given_Index));
    given_NNclassError(i + 1) = length(nonzeros(pruned_labelTest - pruned_predictedLabels)) / g
end

total_NNError = length(nonzeros(labelTestNew - transpose(predicted_labels))) / 500;
```