Họ và Tên: Trần Quang Khải

MHV: CH1502007

GitHub link: <https://github.com/tqkhai>

**BÀI TẬP THỰC HÀNH 2**

Q1. rand([1 200]);

Q2. A(3, 5);

Q3. A = zeros(100, 200);

Q4. size(A, 1);

Q5. A( : , 10);

Q6. A(10, : );

Q7. reshape(A’, 28, 28);

**BÀI TẬP THỰC HÀNH 3**

|  |  |
| --- | --- |
| n | Giá trị |
| 1 | 5 |
| 500 | 8 |
| 5000 | 2 |
| 10000 | 7 |
| 59000 | 4 |

Q1.

function ShowImageTrain(n)

fprintf('\n Load du lieu train');

imgTrainAll = loadMNISTImages('train-images.idx3-ubyte');

lblTrainAll = loadMNISTLabels('train-labels.idx1-ubyte');

figure;

img = imgTrainAll(:, n);

img2D = reshape(img, 28, 28);

strLabelImage = num2str(lblTrainAll(n));

imshow(img2D);

title(strLabelImage);

end

|  |  |
| --- | --- |
| n | Giá trị |
| 1 | 7 |
| 500 | 6 |
| 5000 | 0 |
| 9000 | 0 |

Q2.

function ShowImageTest(n)

fprintf('\n Load du lieu test');

imgTestAll = loadMNISTImages('t10k-images.idx3-ubyte');

lblTestAll = loadMNISTLabels('t10k-labels.idx1-ubyte');

figure;

img = imgTestAll(:, n);

img2D = reshape(img, 28, 28);

strLabelImage = num2str(lblTestAll(n));

imshow(img2D);

title(strLabelImage);

end

Q3.

function CountLabelOfTrainImage()

|  |  |  |  |
| --- | --- | --- | --- |
| Label | SL | Label | SL |
| 0 | 5923 | 5 | 5421 |
| 1 | 6742 | 6 | 5918 |
| 2 | 5958 | 7 | 6265 |
| 3 | 6131 | 8 | 5851 |
| 4 | 5842 | 9 | 5949 |

fprintf('\n Load du lieu train\n');

lblTrainAll = loadMNISTLabels('train-labels.idx1-ubyte');

lblTrainAllCount = size(lblTrainAll, 1);

index = 1;

a = zeros(10);

while index <= lblTrainAllCount

label = lblTrainAll(index);

a(label + 1) = a(label + 1) + 1;

index = index + 1;

end

for i = 1 : 10

fprintf('Label %d co %d anh. \n', i - 1, a(i));

end

end

Q4.

|  |  |  |  |
| --- | --- | --- | --- |
| Label | SL | Label | SL |
| 0 | 980 | 5 | 892 |
| 1 | 1135 | 6 | 958 |
| 2 | 1032 | 7 | 1028 |
| 3 | 1010 | 8 | 974 |
| 4 | 982 | 9 | 1009 |

function CountLabelOfTestImage()

fprintf('\n Load du lieu test\n');

lblTestAll = loadMNISTLabels('t10k-labels.idx1-ubyte');

lblTestAllCount = size(lblTestAll, 1);

index = 1;

a = zeros(10);

while index <= lblTestAllCount

label = lblTestAll(index);

a(label + 1) = a(label + 1) + 1;

index = index + 1;

end

for i = 1 : 10

fprintf('Label %d co %d anh. \n', i - 1, a(i));

end

end

|  |  |
| --- | --- |
| n | Giá trị |
| 5 | 4 |
| 500 | 6 |
| 900 | 8 |

Q5.

function label = TestRecognition(n)

fprintf('\n Load du lieu train');

imgTrainAll = loadMNISTImages('train-images.idx3-ubyte');

lblTrainAll = loadMNISTLabels('train-labels.idx1-ubyte');

Mdl = fitcknn(imgTrainAll', lblTrainAll);

fprintf('\n Load du lieu test \n');

imgTestAll = loadMNISTImages('t10k-images.idx3-ubyte');

lblTestAll = loadMNISTLabels('t10k-labels.idx1-ubyte');

imgTest = imgTestAll(:,n);

lblPredictTest = predict(Mdl, imgTest');

label = lblPredictTest;

end

Q6.

function ShowTestRecognition(n)

fprintf('\n Load du lieu train');

imgTrainAll = loadMNISTImages('train-images.idx3-ubyte');

lblTrainAll = loadMNISTLabels('train-labels.idx1-ubyte');

Mdl = fitcknn(imgTrainAll', lblTrainAll);

fprintf('\n Load du lieu test \n');

imgTestAll = loadMNISTImages('t10k-images.idx3-ubyte');

lblTestAll = loadMNISTLabels('t10k-labels.idx1-ubyte');

imgTest = imgTestAll(:,n);

lblPredictTest = predict(Mdl, imgTest');

figure;

img = imgTestAll(:, n);

img2D = reshape(img, 28, 28);

strLabelImage = num2str(lblTestAll(n));

imshow(img2D);

caption = ['Label: ', strLabelImage, ' | Predict: ', num2str(lblPredictTest)];

if(lblTestAll(n) == lblPredictTest)

caption = [caption, ' ~ KHOP'];

else

caption = [caption, ' ~ KHONG KHOP'];

end

title(caption);

end

Q6\*.

Q7.

function FailedTestRecognitionCount = FailedTestRecognition(n)

fprintf('\n Load du lieu train');

imgTrainAll = loadMNISTImages('train-images.idx3-ubyte');

lblTrainAll = loadMNISTLabels('train-labels.idx1-ubyte');

|  |  |  |  |
| --- | --- | --- | --- |
| n | Sai | n | Sai |
| 0 | 7 | 5 | 32 |
| 1 | 33 | 6 | 14 |
| 2 | 40 | 7 | 36 |
| 3 | 40 | 8 | 54 |
| 4 | 38 | 9 | 42 |

Mdl = fitcknn(imgTrainAll', lblTrainAll);

fprintf('\n Load du lieu test \n');

imgTestAll = loadMNISTImages('t10k-images.idx3-ubyte');

lblTestAll = loadMNISTLabels('t10k-labels.idx1-ubyte');

lblTestAllCount = size(lblTestAll, 1);

index = 1;

FailedTestRecognitionCount = 0;

while index ~= lblTestAllCount

if(lblTestAll(index) == n)

imgTest = imgTestAll(:,index);

lblPredictTest = predict(Mdl, imgTest');

if(lblPredictTest ~= n)

FailedTestRecognitionCount = FailedTestRecognitionCount + 1;

end

end

index = index + 1;

end

end

Q7\*.

function ToConfusionMatrix()

a = zeros(10, 10);

fprintf('\n Load du lieu train');

imgTrainAll = loadMNISTImages('train-images.idx3-ubyte');

lblTrainAll = loadMNISTLabels('train-labels.idx1-ubyte');

Mdl = fitcknn(imgTrainAll', lblTrainAll);

fprintf('\n Load du lieu test \n');

imgTestAll = loadMNISTImages('t10k-images.idx3-ubyte');

lblTestAll = loadMNISTLabels('t10k-labels.idx1-ubyte');

for index = 1:10000

imgTest = imgTestAll(:,index);

lblPredictTest = predict(Mdl, imgTest');

if(lblPredictTest ~= lblTestAll(index))

a(lblTestAll(index) + 1, lblPredictTest + 1) = a(lblTestAll(index) + 1, lblPredictTest + 1) + 1;

end

end

disp(a);

end

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CONFUSION MATRIX** | | | | | | | | | | |
|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **0** | 0 | 1 | 1 | 0 | 0 | 1 | 3 | 1 | 0 | 0 |
| **1** | 0 | 0 | 3 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| **2** | 7 | 6 | 0 | 5 | 1 | 0 | 2 | 16 | 3 | 0 |
| **3** | 0 | 1 | 2 | 0 | 1 | 19 | 0 | 7 | 7 | 3 |
| **4** | 0 | 7 | 0 | 0 | 0 | 0 | 3 | 5 | 1 | 22 |
| **5** | 1 | 1 | 0 | 12 | 2 | 0 | 5 | 1 | 6 | 4 |
| **6** | 4 | 2 | 0 | 0 | 3 | 5 | 0 | 0 | 0 | 0 |
| **7** | 0 | 14 | 6 | 2 | 4 | 0 | 0 | 0 | 0 | 10 |
| **8** | 6 | 1 | 3 | 14 | 5 | 13 | 3 | 4 | 0 | 5 |
| **9** | 2 | 5 | 1 | 6 | 10 | 5 | 1 | 11 | 1 | 0 |