

BÀI TẬP THỰC HÀNH 3





KHẢO SÁT BỘ DỮ LIỆU CHỮ SỐ VIẾT TAY

GV: Lê Đình Duy - Nguyễn Tấn Trần Minh Khang

Họ và tên: Nguyễn Anh Tú
MSHV: CH1601020





- Q1 - Hãy viết function hiển thị ảnh có thứ tự là n (n là tham số) cùng label tương ứng trong tập huấn luyện (train) của tập dữ liệu MNIST. Paste code vào bài thực hành và lập bảng cho biết kết quả khi chạy với n=1, 500, 5000, 10000, 59000.

```
function showNImageTrain(n)
%Q1- Hien thi anh co thu tu n va label tuong ung tap Train
imgTrainAll = loadMNISTImages('./train-images.idx3-ubyte');
lblTrainAll = loadMNISTLabels('./train-labels.idx1-ubyte');
imgTrain = imgTrainAll(:,n);
figure;
img2D = reshape(imgTrain,28,28);
imshow(img2D);
strLabelTrain = num2str(lblTrainAll(n));
title(strLabelTrain);
end
```

n	1	500	5000	59000
Image				
+				
Label				

- Q2 - Hãy viết function hiển thị ảnh có thứ tự là n (n là tham số) cùng label tương ứng trong tập test của tập dữ liệu MNIST. Paste code vào bài thực hành và lập bảng cho biết kết quả khi chạy với n=1, 500, 5000, 9000.

```
function showNImageTest(n)
%Q2- Hien thi anh co thu tu n va label tuong ung tap Test
imgTestAll = loadMNISTImages('./t10k-images.idx3-ubyte');
lblTestAll = loadMNISTLabels('./t10k-labels.idx1-ubyte');
imgTest = imgTestAll(:,n);
figure;
img2D = reshape(imgTest,28,28);
imshow(img2D);
strLabelTest = num2str(lblTestAll(n));
title(strLabelTest);
end
```

n	1	500	5000	9000
Image + Label	7 	6 	0 	0 

- Q3 - Hãy viết function thống kê số lượng các ảnh tương ứng với các label trong tập huấn luyện (train) của tập dữ liệu MNIST. Paste code vào bài thực hành và lập bảng kết quả khi chạy (nên xuất dưới dạng csv để tiện import thành bảng).

```
function countTrainImages()
    lblTrainAll = loadMNISTLabels('./train-labels.idx1-ubyte');
    rArray = [];
    imgNumber = 0;
    nNumLabel = size(lblTrainAll, 1);
    while(imgNumber<=9)
        countImgs = 0;
        for i = 1: nNumLabel
            if(lblTrainAll(i) == imgNumber)
                countImgs = countImgs + 1;
            end
        end
        rArray = [rArray, [imgNumber, countImgs]'];
        imgNumber = imgNumber+1;
    end
    csvwrite('Question3.csv', rArray);
end
```

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

- Q4 - Hãy viết function thống kê số lượng các ảnh tương ứng với các label trong tập test của tập dữ liệu MNIST. Paste code vào bài thực hành và lập bảng kết quả khi chạy (nên xuất dưới dạng csv để tiện import thành bảng).

```
function countTestImages()
%Q4- Thống kê số lượng ảnh có thu từ n và label tương ứng tập Test
    lblTestAll = loadMNISTLabels('./t10k-labels.idx1-ubyte');
    rArray = [];
    imgNumber = 0;
    nNumLabel = size(lblTestAll, 1);
    while(imgNumber<=9)
        countImgs = 0;
        for i = 1: nNumLabel
            if(lblTestAll(i) == imgNumber)
```

```

        countImgs = countImgs + 1;
    end
    end
    rArray = [rArray,[imgNumber,countImgs]'];
    imgNumber = imgNumber+1;
end
csvwrite('Question4.csv',rArray);
end

```

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

- Q5 - Hãy viết function trả về kết quả nhận dạng của ảnh trong tập test có thứ tự là n (n là tham số, nằm trong đoạn [1, 10000]). Paste code vào bài thực hành và lập bảng kết quả khi chạy với n = 5, 500, 900.

```

function RecognitionTestImages(n)
%Q5 - Function tra ve ket qua nhan dang cua anh trong tap Test co thu
%tu n ([1,10000]).
imgTestAll = loadMNISTImages('./t10k-images.idx3-ubyte');
lblTestAll = loadMNISTLabels('./t10k-labels.idx1-ubyte');
imgTrainImagesAll = loadMNISTImages('train-images.idx3-ubyte');
lblTrainLabelsAll = loadMNISTLabels('train-labels.idx1-ubyte');
Mdl = fitcknn(imgTrainImagesAll',lblTrainLabelsAll);
imgTest = imgTestAll(:,n);
lblPredictTest = predict(Mdl,imgTest');
lblImageTest = lblTestAll(n);
fprintf('\nBan dau: %s',num2str(lblTestAll(n)));
fprintf('\nDu doan: %s',num2str(lblPredictTest));
if(lblPredictTest==lblImageTest)
    fprintf('\n Ket qua dung');
else
    fprintf('\n Ket qua sai');
end
end

```

n	5	500	900
Ban đầu	4	6	8
Dự đoán	4	6	8
Kết quả	Đúng	Đúng	Đúng

- Q6 - Hãy viết function với tham số đầu vào n là thứ tự của ảnh trong tập test - sau đó hiển thị ảnh tương ứng - rồi hiển thị kết quả nhận dạng - rồi cho biết kết quả nhận dạng là đúng hay sai khi so khớp với label của tập test.

```
function RecognitionShowTestImages(n)
%Q6 - Nhập n, hiển thị ảnh và kết quả nhận dạng
imgTrainImagesAll = loadMNISTImages('train-images.idx3-ubyte');
lblTrainLabelsAll = loadMNISTLabels('train-labels.idx1-ubyte');
imgTestAll = loadMNISTImages('t10k-images.idx3-ubyte');
lblTestAll = loadMNISTLabels('t10k-labels.idx1-ubyte');
Mdl = fitcknn(imgTrainImagesAll, lblTrainLabelsAll);
imgTest = imgTestAll(:,n);
lblPredictTest = predict(Mdl, imgTest);
lblImageTest = lblTestAll(n);
figure;
img2D = reshape(imgTest, 28, 28);
imshow(img2D);
strLabelImage = 'Ban dau: ';
strLabelImage = [strLabelImage, num2str(lblTestAll(n)), '.'];
strLabelImage = [strLabelImage, 'Du doan: '];
strLabelImage = [strLabelImage, num2str(lblPredictTest), '.'];
if(lblPredictTest == lblImageTest)
    strLabelImage = [strLabelImage, ' Ket qua dung.'];
else
    strLabelImage = [strLabelImage, ' Ket qua sai.'];
end
title(strLabelImage);
end
```

- Q7 - Hãy viết function đếm số lượng các ảnh có label là n (n là tham số) bị nhận dạng sai theo thuật toán knn. Paste code của function đã chạy được vào bài thực hành và lập bảng kết quả khi chạy với n= 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

```
function CountWrongImages(n)
%Q7- Đếm số lượng ảnh có label n bị nhận dạng sai
imgTrainImagesAll = loadMNISTImages('train-images.idx3-ubyte');
lblTrainLabelsAll = loadMNISTLabels('train-labels.idx1-ubyte');
Mdl = fitcknn(imgTrainImagesAll, lblTrainLabelsAll);
imgTestImagesAll = loadMNISTImages('t10k-images.idx3-ubyte');
lblTestLabelsAll = loadMNISTLabels('t10k-labels.idx1-ubyte');
nNumTestImgs = size(imgTestImagesAll, 2);
count = 0;
for nNumber = 1 : nNumTestImgs
    img = imgTestImagesAll(:,nNumber);
    lblPredictTest = predict(Mdl, img);
    if(lblPredictTest ~= lblTestLabelsAll(nNumber))
        if(lblTestLabelsAll(nNumber) == n)
            count = count + 1;
        end
    end
end
```

```

end
fprintf('\n%d',count);
end

```

0	1	2	3	4	5	6	7	8	9
7	6	40	40	38	32	14	36	54	42

Q7* - Tương tự bài tập Q7 - nhưng lập bảng confusion matrix.

```

function CountWrongImagesConfussionMatrix()
%Q7*Dem so luong anh co label n bi nhan dang sai va lap Confusion matrix
imgTrainImagesAll = loadMNISTImages('train-images.idx3-ubyte');
lblTrainLabelsAll = loadMNISTLabels('train-labels.idx1-ubyte');
Mdl = fitcknn(imgTrainImagesAll', lblTrainLabelsAll);
imgTestImagesAll = loadMNISTImages('t10k-images.idx3-ubyte');
lblTestLabelsAll = loadMNISTLabels('t10k-labels.idx1-ubyte');
CMatrix = zeros(10,10);
nNumTestImgs = size(imgTestImagesAll,2);
for nNumber = 1 : nNumTestImgs
    imgTest = imgTestImagesAll(:,nNumber);
    lblImageTest = lblTestLabelsAll(nNumber);
    lblPredictTest = predict(Mdl, imgTest');
    CMatrix(lblImageTest+1,lblPredictTest+1)
=CMatrix(lblImageTest+1,lblPredictTest+1)+1;
end
csvwrite('CMatrix.csv',CMatrix);
end

```

	0	1	2	3	4	5	6	7	8	9
0	973	1	1	0	0	1	3	1	0	0
1	0	1129	3	0	1	1	1	0	0	0
2	7	6	992	5	1	0	2	16	3	0
3	0	1	2	970	1	19	0	7	7	3
4	0	7	0	0	944	0	3	5	1	22
5	1	1	0	12	2	860	5	1	6	4
6	4	2	0	0	3	5	944	0	0	0
7	0	14	6	2	4	0	0	992	0	10
8	6	1	3	14	5	13	3	4	920	5
9	2	5	1	6	10	5	1	11	1	967

- Q8** (Tương đương 50% đồ án môn học nếu kết hợp với Q7*) - Hãy viết function tính độ chính xác của thuật toán knn với các tham số khác nhau của hàm knn (ví dụ số lượng nearest neighbors, độ đo distance). Paste code vào bài thực hành và lập bảng kết quả.

```

% Accuracy(1,1): tham so NumNeighbors, k = 1
% Accuracy(1,3): tham so NumNeighbors, k = 3

```

```

% Accuracy(2,1): tham so Distance, do do minkowski
% Accuracy(3,1): tham so Distance, do do euclidean
function d = Accuracy(param,number)
    imgTrainImagesAll = loadMNISTImages('train-images.idx3-ubyte');
    lblTrainLabelsAll = loadMNISTLabels('train-labels.idx1-ubyte');
    imgTestImagesAll = loadMNISTImages('t10k-images.idx3-ubyte');
    lblTestLabelsAll = loadMNISTLabels('t10k-labels.idx1-ubyte');
    switch(param)
        case (1)
            Md1 = fitcknn(imgTrainImagesAll', lblTrainLabelsAll,
'NumNeighbors', number);
        case (2)
            Md1 = fitcknn(imgTrainImagesAll', lblTrainLabelsAll,
'Distance','minkowski');
        case (3)
            Md1 = fitcknn(imgTrainImagesAll', lblTrainLabelsAll,
'Distance','euclidean');
    end
    nNumTestImgs = size(imgTestImagesAll,2);
    count = 0;
    for nNumber = 1 : nNumTestImgs
        fprintf('\n%d ',nNumber);
        img = imgTestImagesAll(:,nNumber);
        lblPredictTest = predict(Md1, img');
        if(lblPredictTest == lblTestLabelsAll(nNumber))
            count = count + 1;
        end
    end
    d = (count / nNumTestImgs) * 100;
    fprintf('Do chinh xac: %8.3f',d);
end

```

	K=1	K=3
NumNeighbors	96.91%	97.06%

	minkowski	euclidean
Distance	96.91%	96.91%