

Bài tập 1

Họ tên học viên: Nguyễn Mạnh Phúc

MSHV: CH1601033

Link github: <https://github.com/nguyenmanhphuc/VRA-classroom>

Email tài khoản github: nguyenmanhphuc93@gmail.com

Bài tập 2:

- Q1 - Hãy cho biết câu lệnh tạo ngẫu nhiên một số trong đoạn [1 200].
`r = randi([1 200]);`
- Q2 - Hãy cho biết câu lệnh truy cập phần tử dòng 3, cột 5 của ma trận A.
`A(3, 5);`
- Q3 - Hãy cho biết lệnh tạo ma trận A kích thước 100x200 và các giá trị đều là zero (0).
`zeros(100, 200);`
- Q4 - Hãy cho biết lệnh lấy số dòng của ma trận A.
`size(A, 1);`
- Q5 - Hãy cho biết lệnh lấy vector cột 10 của ma trận A.
`A(:, 10);`
- Q6 - Hãy cho biết lệnh lấy vector dòng 10 của ma trận A.
`A(10, :);`
- Q7 - Hãy cho biết lệnh chuyển vector dòng thành ma trận 28x28.
`reshape(A, 28, 28);`

Bài tập 3:






- 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 ShowTrainImage(n)
    allTrainImages = loadMNISTImages('./train-images.idx3-ubyte');
    allTrainLabels = loadMNISTLabels('./train-labels.idx1-ubyte');
    figure;
    image = allTrainImages(:, n);
    label = allTrainLabels(n);
    image2D = reshape(image, 28, 28);
```

```

    label = ['Label: ', num2str(label)];
    imshow(image2D);
    title(label);
end

```

N	Kết quả
1	<div>Label: 5</div> 
500	<div>Label: 8</div> 
5000	<div>Label: 2</div> 
10000	<div>Label: 7</div> 
59000	<div>Label: 4</div> 





- 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 ShowTestImage(n)
    allTestImages = loadMNISTImages('./t10k-images.idx3-ubyte');
    allTestLabels = loadMNISTLabels('./t10k-labels.idx1-ubyte');
    figure;
    image = allTestImages(:, n);
    label = allTestLabels(n);
    image2D = reshape(image, 28, 28);
    label = ['Label: ', num2str( label)];
    imshow(image2D);
    title(label);

```

end

N	Kết quả
1	Label: 7 
500	Label: 6 
5000	Label: 0 
9000	Label: 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 m = CountTrainImagesByLabel()  
    allTrainLabels = loadMNISTLabels('./train-labels.idx1-  
ubyte');  
    %Number of images with zero label will be at m(10);  
    m = zeros(10, 1);  
    numberOfImages = size(allTrainLabels, 1);  
    for i = 1: numberOfImages  
        label = allTrainLabels(i);  
        m(label+1) = m(label+1) +1;  
    end  
  
    csvwrite('CountTrainLabels.csv', m);
```

end

Label	Số lượng ảnh
0	5923
1	6742

2	5958
3	6131
4	5842
5	5421
6	5918
7	6265
8	5851
9	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 m = CountTestImagesByLabel()
    allTrainLabels = loadMNISTLabels('./t10k-labels.idx1-ubyte');
    %Number of images with zero label will be at m(10);
    m = zeros(10, 1);
    numberOfImages = size(allTrainLabels, 1);
    for i = 1: numberOfImages
        label = allTrainLabels(i);
        m(label+1) = m(label+1) + 1;
    end

    csvwrite('CountTestLabels.csv', m);

end
```

Label	Số lượng ảnh
0	980
1	1135
2	1032
3	1010
4	982
5	892
6	958
7	1028
8	974
9	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 label = PredictTestImage (n)
    allTrainImages = loadMNISTImages('./train-
images.idx3-ubyte');
    allTrainLabels = loadMNISTLabels('./train-
labels.idx1-ubyte');

    mdl = fitcknn(allTrainImages', allTrainLabels);

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

    imageTest = allTestImages(:, n);
    label = predict(mdl, imageTest');
end
```

Label	Kết quả
5	4
500	6
900	8

- 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.

```
%Q6
function ShowImageWithLabels (n)
    allTrainImages = loadMNISTImages('./train-
images.idx3-ubyte');
    allTrainLabels = loadMNISTLabels('./train-
labels.idx1-ubyte');

    mdl = fitcknn(allTrainImages', allTrainLabels);

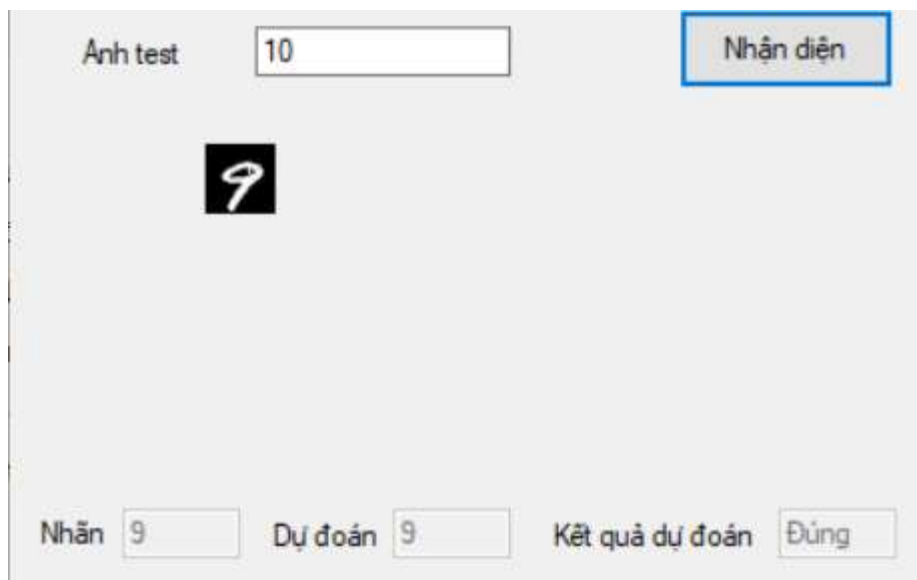
    allTestImages = loadMNISTImages('./t10k-images.idx3-
ubyte');
    allTestLabels = loadMNISTLabels('./t10k-labels.idx1-
ubyte');
    imageTest = allTestImages(:, n);
    label = predict(mdl, imageTest');
    correctLabel = allTestLabels(n);
    figure;
```

```

image2D = reshape(allTestImages(:,n), 28, 28);
imshow(image2D);
imageTitle = ['Predicted: ', num2str(label), '
Correct Label: ', num2str(correctLabel), ' => '];
if label ==correctLabel
    imageTitle = [imageTitle, 'Correct Prediction'];
else
    imageTitle = [imageTitle, 'Wrong Prediction'];
end
title(imageTitle);
end

```

- Q6* - Hãy viết thêm phần giao diện cho bài trên - tham khảo code ở đây: <https://bitbucket.org/intelligenceagent/cudacnn-public/wiki/Home>



- 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 count = CountWrongLabelPrediction(n)
    allTrainImages = loadMNISTImages('./train-images.idx3-
ubyte');
    allTrainLabels = loadMNISTLabels('./train-labels.idx1-
ubyte');

    mdl = fitcknn(allTrainImages', allTrainLabels);

```

```

allTestImages = loadMNISTImages('./t10k-images.idx3-
ubyte');
allTestLabels = loadMNISTLabels('./t10k-labels.idx1-
ubyte');
count = 0;
countTestImages = size(allTestImages, 2);
for i = 1: countTestImages
    correctLabel = allTestLabels(i);
    if(correctLabel~=n)
        continue;
    end
    imageTest = allTestImages(:, i);
    predictedLabel = predict mdl, imageTest');

    if predictedLabel ~= correctLabel
        count = count + 1;
    end
end
end

```

N	Số lượng ảnh bị nhận diện sai
0	7
1	6
2	40
3	40
4	38
5	32
6	14
7	36
8	54
9	42

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

```

function matrix = CreateConfusionMatrix()
    allTrainImages = loadMNISTImages('./train-
images.idx3-ubyte');
    allTrainLabels = loadMNISTLabels('./train-
labels.idx1-ubyte');

    mdl = fitcknn(allTrainImages', allTrainLabels);

```

```

allTestImages = loadMNISTImages('./t10k-images.idx3-
ubyte');
allTestLabels = loadMNISTLabels('./t10k-labels.idx1-
ubyte');
matrix = zeros(10);
countTestImages = size(allTestImages, 2);
for i = 1: countTestImages
    correctLabel = allTestLabels(i);

    imageTest = allTestImages(:, i);
    predictedLabel = predict mdl, imageTest');
    matrix(correctLabel+1, predictedLabel+1) =
matrix(correctLabel+1, predictedLabel+1) + 1;
end
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% đề 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.

```

function acc = CalAccurate(num, distance)
    allTrainImages = loadMNISTImages('./train-images.idx3-
ubyte');
    allTrainLabels = loadMNISTLabels('./train-labels.idx1-
ubyte');

    mdl = fitcknn(allTrainImages', allTrainLabels,
'Distance', distance, 'NumNeighbors', num);

```



```

    allTestImages = loadMNISTImages('./t10k-images.idx3-
ubyte');
    allTestLabels = loadMNISTLabels('./t10k-labels.idx1-
ubyte');

    countTestImages = size(allTestImages, 2);
    correct = 0;
    predictedLabels = predict mdl, allTestImages');
    correct = sum(predictedLabels==allTestLabels);
    acc = correct/countTestImages*100;
end

```

Loại khoảng cách	euclidean	cosine	minkowski	cityblock
k				
1	96.91	97.23	96.91	96.31
3	97.06	97.37	97.06	96.31
5	96.87	97.29	96.87	96.16
10	96.70	97.16	96.70	95.91
20	96.24	96.86	96.24	95.50
50	95.34	96.11	95.34	94.38