BÀI TẬP VÀ THỰC HÀNH 1

1.Tài khoản github: <https://github.com/thuynttk11/VRA>

2.Họ và tên HV: Nguyễn Thị Thu Thủy

MSHV: CH1601018

Email tài khoản github: [thuyntt.11@grad.uit.edu.vn](mailto:thuyntt.11@grad.uit.edu.vn)

3.

Vd1: Tạo ngẫu nhiên một số thực trong đoạn [0..1]

Tạo ngẫu nhiên một số nguyên trong doạn [1 10]

Tạo ngẫu nhiên một ma trận kích thước 1x10 các số nguyên nằm trong đoạn [-10,10]

function Example001()

a = rand();

fprintf('\n a = %8.3f',a);

r = randi([1 10]);

fprintf('\n Random tu 1 den 10: %d',r);

matran = randi([-10,10],1,10);

fprintf('\n Size ma tran: %d',size(matran,2));

fprintf('\n Ma tran duoc tao: ');

fprintf('[%2d]',matran);

end

Vd2: Tạo ma trận

function Example001()

strMessage = '\n Nhap m: ';

m = input(strMessage);

strMessage = '\n Nhap n: ';

n= input(strMessage);

a = ones(m,n); fprintf('[%d]',a);

b = zeros(m,n); fprintf('[%d]',b);

c=eye(m,n);

d=randi([-10,10],m,n);

a(1,1) =5

e = size(a);

end

Vd3:Viết chương trình: nhập số nguyên n và tính tổng từ 1 đến số nguyên n với vòng lặp while

function Example003()

strMessage = '\n Nhap n: ';

n = input(strMessage);

s = 0;

i = 1;

while i<=n

s = s+i;

i = i+1;

end

fprintf('\n Tong S = 1+2+...+%d la %d.\n',n,s);

end

Vd4:Viết chương trình: nhập số nguyên n và tính tổng từ 1 đến số nguyên n với vòng lặp for

function Example004()

strMessage = '\n Nhap n: ';

n = input(strMessage);

s = 0;

for i =1:n

s=s+i;

end

fprintf('\n Tong S = 1 + 2 + ... + %d la %d.\n',n,s);

end

Hàm isprime(n): kiểm tra số nguyên n có nguyên tố không (trả về 1 nếu nguyên tố, trả về 0 nếu không nguyên tố)

function m = isprime(n)

count = 0;

for i = 1:n

if(mod(n,i)==0)

count = count +1;

end

end

m=(count==2);

end

Hàm findnprime(n): trả về một mảng gồm n số nguyên tố đầu tiên.

function primeArray = findnprime(n)

count = 0;

i = 2;

primeArray = [];

while (count<n)

if(isprime(i)==1)

count = count +1;

primeArray = [primeArray,i];

end

i=i+1;

end

end

Hàm savenprime(n): tìm n số nguyên tố đầu tiên và lưu vào tập tin có tên là “prime<n>.mat”

function saveprime(n)

rArray = findnprime(n);

strFileName = ['prime',num2str(n),'.mat'];

save(strFileName,'rArray');

end

Hàm loadnprime(n): nạp n số nguyên tố từ tập tin có tên “prime<n>.mat”

function loadnprime(n)

strFileName = ['prime',num2str(n),'.mat'];

load(strFileName);

rArray

end

Hàm loadMNISTImages(filename): trả về ma trận chứa các ảnh của MNIST mỗi ảnh có kích thước 28x28

function images = loadMNISTImages(filename)

%loadMNISTImages returns a 28x28x[number of MNIST images] matrix containing

%the raw MNIST images

fp = fopen(filename, 'rb');

assert(fp ~= -1, ['Could not open ', filename, '']);

magic = fread(fp, 1, 'int32', 0, 'ieee-be');

assert(magic == 2051, ['Bad magic number in ', filename, '']);

numImages = fread(fp, 1, 'int32', 0, 'ieee-be');

numRows = fread(fp, 1, 'int32', 0, 'ieee-be');

numCols = fread(fp, 1, 'int32', 0, 'ieee-be');

images = fread(fp, inf, 'unsigned char');

images = reshape(images, numCols, numRows, numImages);

images = permute(images,[2 1 3]);

fclose(fp);

% Reshape to #pixels x #examples

images = reshape(images, size(images, 1) \* size(images, 2), size(images, 3));

% Convert to double and rescale to [0,1]

images = double(images) / 255;

end

Hàm loadMNISTLabels(filename): trả về ma trận chứa các nhãn ảnh của MNIST

function labels = loadMNISTLabels(filename)

%loadMNISTLabels returns a [number of MNIST images]x1 matrix containing

%the labels for the MNIST images

fp = fopen(filename, 'rb');

assert(fp ~= -1, ['Could not open ', filename, '']);

magic = fread(fp, 1, 'int32', 0, 'ieee-be');

assert(magic == 2049, ['Bad magic number in ', filename, '']);

numLabels = fread(fp, 1, 'int32', 0, 'ieee-be');

labels = fread(fp, inf, 'unsigned char');

assert(size(labels,1) == numLabels, 'Mismatch in label count');

fclose(fp);

end

Vấn đề 01: Hàm Recognition001\_Digits() nạp dữ liệu train và dữ liệu test của bài toán nhận dạng chữ viết tay

function Recognition001\_Digits()

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

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

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

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

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

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

fprintf('\n Ket thuc .\n');

end

Vấn đề 02: Hàm Recognition002\_Digits() cho biết số lượng ảnh train và ảnh test sau khi nạp dữ liệu train và dữ liệu test của bài toán nhận dạng chữ viết tay

function Recognition002\_Digits()

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

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

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

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

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

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

nTrainImages=size(imgTrainAll,2);

nTrainLabels=size(lblTrainAll,1);

nTestImages=size(imgTestAll,2);

nTestLabels=size(lblTestAll,1);

nSizeofImage =size(imgTrainAll,1);

fprintf('\n So luong anh train: [%d].',nTrainImages);

fprintf('\n So luong nhan anh train: [%d].l',nTrainLabels);

fprintf('\n So luong anh test: [%d].', nTestImages);

fprintf('\n So luong nhan anh test: [%d].',nTestLabels);

fprintf('\n Kich thuoc cua mot anh: [%d].',nSizeofImage);

end

Vấn đề 03: Hàm Recognition003\_Digits() hiển thị ảnh train thứ nhất và cuối cùng sau khi nạp dữ liệu train và dữ liệu test của bài toán nhận dạng chữ viết tay

function Recognition003\_Digits()

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

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

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

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

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

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

nTrainImages=size(imgTrainAll,2);

figure;

img=imgTrainAll(:,1);

img2D=reshape(img,28,28);

strLabelImage=num2str(lblTrainAll(1));

imshow(img2D);

title(strLabelImage);

figure;

imgLast=imgTrainAll(:,nTrainImages);

img2DLast=reshape(imgLast,28,28);

strLabelImage=num2str(lblTrainAll(nTrainImages));% lblTrainLabelsAll

imshow(img2DLast);

title(strLabelImage);

end

Vấn đề 04: Hàm Recognition004\_Digits() hiển thị ảnh ngẫu nhiên trong bộ dữ liệu train và bộ dữ liệu test sau khi nạp dữ liệu train và dữ liệu test của bài toán nhận dạng chữ viết tay

function Recognition004\_Digits()

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

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

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

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

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

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

nTrainImages=size(imgTrainAll,2);

nNumber=randi([1 nTrainImages]);

figure;

img=imgTrainAll(:,nNumber);

img2D=reshape(img,28,28);

strLabelImage=num2str(lblTrainAll(nNumber));

strLabelImage=[strLabelImage,'(',num2str(nNumber),')'];

imshow(img2D);

title(strLabelImage);

nTestImgs=size(imgTestAll,2);

nNumber=randi([1 nTestImgs]);

figure;

img=imgTestAll(:,nNumber);

img2D=reshape(img,28,28);

strLabelImage=num2str(lblTestAll(nNumber));

strLabelImage=[strLabelImage,'(',num2str(nNumber),')'];

imshow(img2D);

title(strLabelImage);

end

Vấn đề 05: Hàm Recognition005\_Digits()

-Nạp dữ liệu train của bài toán nhận dạng chữ viết tay

-Xây dựng một model từ tập dữ liệu train bằng cách sử dụng thuật toán kNN

-Nạp dữ liệu test của bài toán nhận dạng chữ viết tay. Lấy ngẫu nhiên một ảnh từ tập dữ liệu test và sử dụng model của câu trên để predict (phân tích).

-Kiểm tra kết quả tiên đoán có đúng không

function Recognition005\_Digits()

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');

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

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

nTestImgs=size(imgTestAll,2);

nNumber=randi([1 nTestImgs]);

imgTest=imgTestAll(:,nNumber);

lblPredictTest=predict(Mdl,imgTest');

lblImageTest=lblTestAll(nNumber);

figure;

img2D=reshape(imgTest,28,28);

imshow(img2D);

strLabelImage='Ban dau';

strLabelImage=[strLabelImage,num2str(lblTestAll(nNumber)),'.']; %lblTestLabelAll

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

BÀI TẬP VÀ THỰC HÀNH 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).

A=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 VÀ THỰC HÀNH 3

|  |  |
| --- | --- |
| Q1: Funtion hiển thị ảnh thứ n cùng label tương ứng trong tập train | |
| Code | KQ |
| function Q1\_TH3(number)  strData='train-images.idx3-ubyte';  strDataLabel='train-labels.idx1-ubyte';  [allTrainImages,allTrainLabels]=LoadData(strData,strDataLabel);    figure;  img=allTrainImages(:,number);  img2D=reshape(img,28,28);  lblData=num2str(allTrainLabels(number));  lblData=[lblData,'(',num2str(number),')'];  imshow(img2D);  title(lblData);  end |  |
| Lệnh gợi trong Command Window  Q1\_TH3(1);  Q1\_TH3(500);  Q1\_TH3(5000);  Q1\_TH3(10000);  Q1\_TH3(59000); | |

|  |  |
| --- | --- |
| Q2: Funtion hiển thị ảnh thứ n cùng label tương ứng trong tập test | |
| Code | KQ |
| function Q2\_TH3(number)  testData='t10k-images.idx3-ubyte';  testDataLabel='t10k-labels.idx1-ubyte';  [allTestImages,allTestLabels]=LoadData(testData, testDataLabel);    figure;  img=allTestImages(:,number);  img2D=reshape(img,28,28);  lblData=num2str(allTestLabels(number));  lblData=[lblData,'(',num2str(number),')'];  imshow(img2D);  title(lblData);  end |  |
| Lệnh gọi trong Command Window  Q2\_TH3(1);  Q2\_TH3(500);  Q2\_TH3(5000);  Q2\_TH3(9000); | |

|  |  |
| --- | --- |
| Q3: Thống kê số lượng ảnh tương ứng với label trong tập train | |
| Code | KQ |
| function Q3\_TH3()    allTrainLabels = loadMNISTLabels('./train-labels.idx1-ubyte');  nTrainLabels = size(allTrainLabels);    nResult = zeros([10 1]);    for i = 1 : nTrainLabels  k = allTrainLabels(i);  nResult(k+1) = nResult(k+1) + 1;  end    fprintf('Thong ke tap train:\n');  bangKQ=zeros([10 2]);  for i = 1:10  bangKQ(i,1)=i-1;  bangKQ(i,2)=nResult(i);  end  bangKQ  filename="Q3\_TH3.csv";  csvwrite(filename,bangKQ);  end | Nhãn Số lượng nhãn  0 5923  1 6742  2 5958  3 6131  4 5842  5 5421  6 5918  7 6265  8 5851  9 5949 |
| Lệnh gọi trong Command Window  Q3\_TH3(); | |

|  |  |
| --- | --- |
| Q4: Thống kê số lượng ảnh tương ứng với label trong tập test | |
| Code | KQ |
| function Q4\_TH3()    allTestLabels = loadMNISTLabels('./t10k-labels.idx1-ubyte');    nTestLabels = size(allTestLabels);    end | Nhãn Số lượng nhãn   |  |  | | --- | --- | | 0 | 980 | | 1 | 1135 | | 2 | 1032 | | 3 | 1010 | | 4 | 982 | | 5 | 892 | | 6 | 958 | | 7 | 1028 | | 8 | 974 | | 9 | 1009 | |
| Lệnh gọi trong Command Window  Q4\_TH3(); | |

|  |  |
| --- | --- |
| Q5: Kết quả thứ tự nhận dạng của ảnh thứ n trong tập test | |
| Code | KQ |
| function Q5\_TH3(number)  imgTrainAll= loadMNISTImages('./train-images.idx3-ubyte');  lblTrainAll= loadMNISTLabels('./train-labels.idx1-ubyte');    Mdl=fitcknn(imgTrainAll',lblTrainAll);    imgTestAll=loadMNISTImages('./t10k-images.idx3-ubyte');  lblTestAll=loadMNISTLabels('./t10k-labels.idx1-ubyte');    imgTest=imgTestAll(:,number);    lblPredictTest=predict(Mdl,imgTest');  lblImageTest=lblTestAll(number);    figure;  img2D=reshape(imgTest,28,28);  imshow(img2D);    lblData='Anh thu: ';  lblData=[lblData,num2str(number),'. Ket qua la: '];  title(lblData);  end |  |
| Lệnh gọi trong Command Window  Q5\_TH3(5);  Q5\_TH3(500);  Q5\_TH3(900); | |

|  |  |
| --- | --- |
| Q6: Kết quả nhận dạng ảnh thứ tự n đúng hay sai tập test | |
| Code | KQ |
| function Q6\_TH3(number)  imgTrainAll= loadMNISTImages('./train-images.idx3-ubyte');  lblTrainAll= loadMNISTLabels('./train-labels.idx1-ubyte');    Mdl=fitcknn(imgTrainAll',lblTrainAll);    imgTestAll=loadMNISTImages('./t10k-images.idx3-ubyte');  lblTestAll=loadMNISTLabels('./t10k-labels.idx1-ubyte');    imgTest=imgTestAll(:,number);    lblPredictTest=predict(Mdl,imgTest');  lblImageTest=lblTestAll(number);    figure;  img2D=reshape(imgTest,28,28);  imshow(img2D);    lblData='Anh thu: ';  lblData=[lblData,num2str(number),'. Du doan: '];    lblData=[lblData,num2str(lblPredictTest),'. Anh KQ: '];    lblData=[lblData,num2str(lblTestAll(number))];  title(lblData);  if(lblPredictTest==lblImageTest)  lblData=[lblData,'. Nhan dang dung.'];  else  lblData=[lblData,'. Nhan dang sai.'];  end  title(lblData);  end |  |
| Lệnh gọi trong Command Window  Q6\_TH3(5);  Q6\_TH3(500);  Q6\_TH3(900); | |

|  |  |
| --- | --- |
| Q7: Số lượng các ảnh có label thứ n (0🡪9) bị nhận dạng sai theo thuật toán knn | |
| Code | KQ |
| function Q7\_TH3(number)  imgTrainAll= loadMNISTImages('./train-images.idx3-ubyte');  lblTrainAll= loadMNISTLabels('./train-labels.idx1-ubyte');    Mdl=fitcknn(imgTrainAll',lblTrainAll);    imgTestAll=loadMNISTImages('./t10k-images.idx3-ubyte');  lblTestAll=loadMNISTLabels('./t10k-labels.idx1-ubyte');    nLabelTest=size(lblTestAll,1);  count=0;  for i=1:nLabelTest  if(lblTestAll(i) == number)  imgTest = imgTestAll(:, i);  lblPredictTest=predict(Mdl,imgTest');  if(lblPredictTest ~= lblTestAll(i))  count = count + 1;  end  end  end    fprintf('So luong anh co nhan thu %d bi nhan dang sai la %d: ',number,count);    end | Nhãn thứ Số lượng nhận dạng sai  0 7  1 6  2 40  3 40  4 38  5 32  6 14  7 36  8 54  9 42 |
| Lệnh gọi trong Command Window  Q7\_TH3(0); 🡢 Q7\_TH3(9); | |