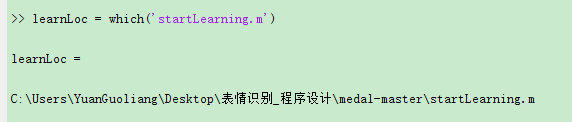
# MATLAB使用笔记

## 一 与路径相关

### 1which

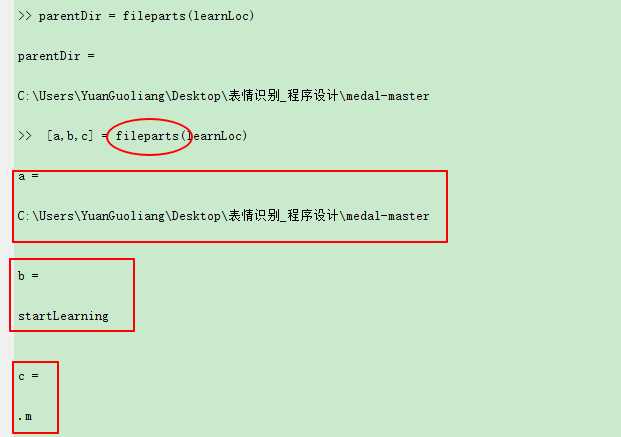
可以通过一个函数或脚本名称得到它的完整路径,同时还能处理函数重载的情况，

用于查找函数或文件的位置



### 2fileparts

用于将一个完整的文件名分割成3部分：路径，文件名，扩展名（返回一个值的话默认是路径）。例如：



### 3ddpath和rmpath

用于对matlab搜索路径的添加和删除。例如：

<<addpath(‘directory’) %将完整路径directory加入到当前搜索路径的最顶端

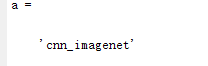
<<rmpath

### 4mfilename

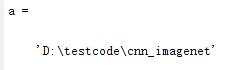
 mfilename是返回当前正在运行的函数所在文件的文件名的函数。

function [net, info] = cnn\_imagenet(varargin)

a=mfilename



a=mfilename('fullpath')



### 5fullfile

fullfile函数作用是利用文件各部分信息创建并合成完整文件名。

用法：

fullfile('dir1', 'dir2', ..., 'filename')

f = fullfile('dir1', 'dir2', ..., 'filename')

具体例子:

输入：f = fullfile('C:','Applications','matlab','fun.m')

得到：f =C:\Applications\matlab\fun.m

### imread

I = imread('..\mh\_gray.bmp'); ..表示当前目录的上一级目录

I = imread('.\photo\bmp1.bmp'); .表示当前目录，一般可以加子目录

## 二 函数

### 1 Imfilter

**功能：**对任意类型数组或多维图像进行滤波。  
**用法：**B = imfilter(A,H)  
　　　B = imfilter(A,H,option1,option2,...)  
　　　或写作g = imfilter(f, w, filtering\_mode, boundary\_options, size\_options)  
其中，f为输入图像，w为滤波掩模，g为滤波后图像。filtering\_mode用于指定在滤波过程中是使用“相关”还是“卷积”。boundary\_options用于处理边界充零问题，边界的大小由滤波器的大小确定。具体参数选项见下表：

|  |  |  |
| --- | --- | --- |
|  | 选项 | 描述 |
| filtering\_mode | ‘corr’ | 通过使用相关来完成，该值为默认。 |
|  | ‘conv’ | 通过使用卷积来完成 |
| boundary\_options | ‘X’ | 输入图像的边界通过用值X（无引号）来填充扩展 其默认值为0 |
|  | ‘replicate’ | 图像大小通过复制外边界的值来扩展 |
|  | ‘symmetric’ | 图像大小通过镜像反射其边界来扩展 |
|  | ‘circular’ | 图像大小通过将图像看成是一个二维周期函数的一个周期来扩展 |
| size\_options | ‘full’ | 输出图像的大小与被扩展图像的大小相同 |
|  | ‘same’ | 输出图像的大小与输入图像的大小相同。这可通过将滤波掩模的中心点的偏移限制到原图像中包含的点来实现，该值为默认值。 |

**举例：**originalRGB = imread('peppers.png');  
imshow(originalRGB)  
h = fspecial('motion', 50, 45);%创建一个滤波器  
filteredRGB = imfilter(originalRGB, h);  
figure, imshow(filteredRGB)

### 2 fspecial

|  |  |
| --- | --- |
| |  | | --- | | fspecial函数用于建立预定义的滤波算子，其语法格式为： h = fspecial(type) h = fspecial(type，para) 其中type指定算子的类型，para指定相应的参数； type的类型有： 1、'average' averaging filter 为均值滤波，参数为hsize代表模板尺寸，默认值为【3，3】。 H = FSPECIAL('average',HSIZE) returns an averaging filter H of size 2、 'disk' circular averaging filter 为圆形区域均值滤波，参数为radius代表区域半径，默认值为5. H = FSPECIAL('disk',RADIUS) returns a circular averaging filter 3、'gaussian' Gaussian lowpass filter  为高斯低通滤波，有两个参数，hsize表示模板尺寸，默认值为【3 3】，sigma为滤波器的标准值，单位为像素，默认值为0.5. H = FSPECIAL('gaussian',HSIZE,SIGMA) returns a rotationally 4、'laplacian' filter approximating the 2-D Laplacian operator 为拉普拉斯算子，参数alpha用于控制算子形状，取值范围为【0，1】，默认值为0.2.  H = FSPECIAL('laplacian',ALPHA) returns a 3-by-3 filter 5、'log' Laplacian of Gaussian filter 为拉普拉斯高斯算子，有两个参数，hsize表示模板尺寸，默认值为【3 3】，sigma为滤波器的标准差，单位为像素，默认值为0.5. H = FSPECIAL('log',HSIZE,SIGMA) returns a rotationally symmetric 6、'motion' motion filter 为运动模糊算子，有两个参数，表示摄像物体逆时针方向以theta角度运动了len个像素，len的默认值为9，theta的默认值为0； H = FSPECIAL('motion',LEN,THETA) returns a filter to approximate, once 7、'prewitt' Prewitt horizontal edge-emphasizing filter 用于边缘增强，大小为【3 3】，无参数 H = FSPECIAL('prewitt') returns 3-by-3 filter that emphasizes   [1 1 1;0 0 0;-1 -1 -1]. 8、'sobel' Sobel horizontal edge-emphasizing filter 用于边缘提取，无参数 H = FSPECIAL('sobel') returns 3-by-3 filter that emphasizes  9、'unsharp' unsharp contrast enhancement filter 为对比度增强滤波器。参数alpha用于控制滤波器的形状，范围为【0，1】，默认值为0.2. H = FSPECIAL('unsharp',ALPHA) returns a 3-by-3 unsharp contrast | |
|  |

### 3 isnan

Matlab中isnan函数

函数名称： isnan

函数功能： 判断数组的元素是否是NaN。NaN即Not a Number的缩写。

matlab中以下情况会出现NaN： 任何关于NaN的数值运算， 如sqrt(NaN) 加减法，如(+Inf)+(-Inf) 乘法, 如0乘以Inf 除法，如0/0、Inf/Inf Remainder, 比如rem(x, y)其中y为0或x是Inf。

语法格式： TF = isnan(A) 返回一个和A尺寸一样的数组， 如果A中某个元素是NaN， 则对应TF中元素是1， 否则TF中对应元素是0。 相关： isfinite、inf、nan、isinf Example

a = [-2 -1 0 1 2]

isnan(1./a) Warning: Divide by zero. ans = 0 0 0 0 0 isnan(0./a) Warning: Divide by zero. ans = 0 0 1 0 0

### 4 mod

简单的说mod(a,b)就是求的是a除以b的余数。比方说mod(100,3)=1,mod(17,6)=5

### 5 ceil fix floor round

ceil函数的作用是朝正无穷方向取整，即将m/n的结果向正无穷方向取整，如m/n=3.12，则ceil(m/n)的结果为4。  
类似的函数有如下几个：  
fix：朝零方向取整，如fix(-1.3)=-1; fix(1.3)=1;  
floor：朝负无穷方向取整，如floor(-1.3)=-2; floor(1.3)=1;  
round：四舍五入到最近的整数，如round(-1.3)=-1;round(-1.52)=-2;round(1.3)=1;round(1.52)=2。

### 6 eigs

d = eigs(A) %求稀疏矩阵A的6个绝对值最大特征值d，d以向量形式存放。

d = eigs(A,B) %求稀疏矩阵的广义特征值问题。满足AV=BVD，其中D为特征值对角阵，V为特征向量矩阵，B必须是对称正定阵或Hermitian正定阵。

d = eigs(A,k) %返回k个最大特征值

d = eigs(A,B,k) %返回k个最大特征值

d = eigs(A,k,sigma) %sigma取值：'lm' 表示绝对值最大的特征值；'sm' 绝对值最小特征值；对实对称问题：'la'表示最大特征值；'sa'为最小特征值；对非对称和复数问题：'lr' 表示最大实部；'sr' 表示最小实部；'li' 表示最大虚部；'si'表示最小虚部

d = eigs(A,B,k,sigma) %同上

d = eigs(A,k,sigma,opts) % opts为指定参数：参见eigs帮助文件。opts为一个向量

参数 描述 value

opts.issym

=1:如果A对称

=0：A不对称

{0|1}

opts.isreal

=1:A为实数

=0：otherwise

{0|1}

opts.tol

收敛？？？（没看懂）

\*\*估计<tol\*norm(A)

opts.maxit 最大迭代次数

opts.p lanczos向量个数？？（没看懂）

。。

。。

。。

。。

d = eigs(A,B,k,sigma,options) %同上。以下的参数k、sigma、options相同。

d = eigs(Afun,n) %用函数Afun代替A，n为A的阶数，D为特征值。

d = eigs(Afun,n,B)

d = eigs(Afun,n,k)

d = eigs(Afun,n,B,k)

d = eigs(Afun,n,k,sigma)

d = eigs(Afun,n,B,k,sigma)

d = eigs(Afun,n,k,sigma,options)

d = eigs(Afun,n,B,k,sigma,options)

[V,D] = eigs(A,…) %D为6个最大特征值对角阵，V的列向量为对应特征向量。

[V,D] = eigs(Afun,n,…)

[V,D,flag] = eigs(A,…) %flag表示特征值的收敛性，若flag=0，则所有特征值都收敛，否则，不是所有都收敛。

[V,D,flag] = eigs(Afun,n,…)

### 7 strcat

定义

strcat 即 Strings Catenate，横向连接字符串。

语法

combinedStr= strcat(s1, s2, ..., sN)

描述

将数组 s1,s2,...,sN 水平地连接成单个字符串，并保存于变量combinedStr中。如果任一参数是元胞数组，那么结果 combinedStr 是一个元胞数组，否则，combinedStr是一个字符数组。

其实它的作用很简单就是将多个字符连接成单个字符串，关键在于这个语句中括号里面的内容，

'D:\thesis\data\gamit\coordinate\' 加了单引号的，就说明单引号里面的内容为：字符串 ，下面的'\' 与'.org' 相同都只是字符串而已，原样输出即可。 置于接下来的 yr 和 model 肯定是先前人为定义的变量，里面肯定储存了某些字符串信息的。

那么这个语句最终实现的结果是：

将“ D:\thesis\data\gamit\coordinate\yr的值\model的值.org” 存于变量filename 中

看程序部能光看一句，这样这个语句事实上完成了 一个文件的创建，便于接下去程序中实现 调用路径中的文件数据 或 修改。

### 8 isa

功能说明

判断输入参量是否为指定类型的对象。

函数语法

K= isa(obj, 'class\_name')

参数解析

K = isa(obj, 'class\_name') 判断obj是否为class\_name类型。如果是，返回逻辑1（真）；如果不是，返回逻辑0（假）。

参量obj是一个MATLAB对象或者Java对象。参量class\_name是MATLAB（预定义的或用户定义的）对象或Java对象。预定义的MATLAB对象包括如下类型：

logical 逻辑数组

char 字符串数组

numeric 整型或浮点型数组

integer 有符号或无符号整型数组

int8 8位有符号整型数组

uint8 8为无符号整型数组

int16 16为有符号整型数组

uint16 16位无符号整型数组

int32 32位有符号整型数组

uint32 32位无符号整型数组

int64 64位有符号整型数组

uint64 64位无符号整型数组

float 单精度或双精度浮点数数组

single 单精度浮点数数组

double 双精度浮点数数组

cell 元胞数组

struct 结构体数组

function\_handle 函数句柄

'class\_name' MATLAB类或Java类

To check for a sparse array, use issparse. To check for a complex array, use ~isreal.

要判断输入参量是否为稀疏矩阵，请使用issparse。要判断输入参量是否为复数矩阵，请使用~isreal。

典型实例

实例1

isa(rand(3,4),'double')

ans =

1

实例2

下面的例子创建了一个用户定义的MATLAB类（名为polynom）实例。使用isa函数判断对象是否为polynom类。该实例仅介绍如何使用isa函数判断一个对象是否为MATLAB类，如果您没有定义名为polynom的类，该实例无法运行。

polynom\_obj = polynom([1 0 -2 -5]);

isa(polynom\_obj, 'polynom')

ans =

1

### 9 conv2

**功能：**是二维卷积运算函数（与convmtx2相似）。如果a和b是两个离散变量n1和n2的函数，则关于a和b的二维卷积运算数学公式如下：

用法：C = conv2(A,B)

C = conv2(hcol,hrow,A)

C = conv2(...,'shape')

C = conv2(A,B)计算数组A和B的卷积。如果一个数组描述了一个二维FIR滤波器，则另一个数组被二维滤波。当A的大小为[ma,na],B的大小为[mb,nb]时，C的大小为[ma+mb-1,mb+nb-1]。‘shape’见下表

参数值

含义

‘full’

默认值，返回全部二维卷积值。

‘same’

返回与A大小相同卷积值的中间部分

‘valid’

当all(size(A)>=size(B)),C的大小为[ma+mb-1,mb+nb-1]；否则，C返回[]。在n维卷积运算中，C的大小为max(size(A)- size(B)+1,0)

例子：

s = [1 2 1; 0 0 0; -1 -2 -1];

A = zeros(10);

A(3:7,3:7) = ones(5);

H = conv2(A,s);

mesh(H)

### 10 imag real （求复数的虚部，实部）

imag是求复数的虚数部分函数

用法：

Y=imag(Z)

求复数5+8i

a=5+8i

imag(a)

ans=

8

### 11 matlab神经网络newff函数的使用

设[P,T]是训练样本，[X,Y]是测试样本；

net=newrb(P,T,err\_goal,spread); %建立网络

q=sim(net,p);

e=q-T;

plot(p,q); %画训练误差曲线

q=sim(net,X);

e=q-Y;

plot(X,q); %画测试误差曲线

训练前馈网络的第一步是建立网络对象。函数newff建立一个可训练的前馈网络。这需要4个输入参数。

第一个参数是一个Rx2的矩阵以定义R个输入向量的最小值和最大值。

第二个参数是一个设定每层神经元个数的数组。

第三个参数是包含每层用到的传递函数名称的细胞数组。

最后一个参数是用到的训练函数的名称。

举个例子，下面命令将创建一个二层网络。它的输入是两个元素的向量，第一层有三个神经元(3)，第二层有一个神经元(1)。

第一层的传递函数是tan-sigmoid，输出层的传递函数是linear。

输入向量的第一个元素的范围是-1到2[-1 2]，输入向量的第二个元素的范围是0到5[0 5]，训练函数是traingd。

　　net=newff([-1 2; 0 5],[3,1],{'tansig','purelin'},'traingd');

这个命令建立了网络对象并且初始化了网络权重和偏置，因此网络就可以进行训练了。

我们可能要多次重新初始化权重或者进行自定义的初始化。

下面就是初始化的详细步骤。

　　在训练前馈网络之前，权重和偏置必须被初始化。初始化权重和偏置的工作用命令init来实现。这个函数接收网络对象并初始化权重和偏置后返回网络对象。

下面就是网络如何初始化的：

　　net = init(net);

　　我们可以通过设定网络参数net.initFcn和net.layer{i}.initFcn这一技巧来初始化一个给定的网络。

net.initFcn用来决定整个网络的初始化函数。前馈网络的缺省值为initlay，它允许每一层用单独的初始化函数。

设定了net.initFcn ，那么参数net.layer{i}.initFcn 也要设定用来决定每一层的初始化函数。

　　对前馈网络来说，有两种不同的初始化方式经常被用到：initwb和initnw。initwb函数根据每一层自己的初始化参数(net.inputWeights{i,j}.initFcn)初始化权重矩阵和偏置。前馈网络的初始化权重通常设为rands，它使权重在-1到1之间随机取值。这种方式经常用在转换函数是线性函数时。initnw通常用于转换函数是曲线函数。它根据Nguyen和Widrow[NgWi90]为层产生初始权重和偏置值，使得每层神经元的活动区域能大致平坦的分布在输入空间。

它比起单纯的给权重和偏置随机赋值有以下优点：

（1）减少神经元的浪费（因为所有神经元的活动区域都在输入空间内）。

（2）有更快的训练速度（因为输入空间的每个区域都在活动的神经元范围中）。

　　初始化函数被newff所调用。因此当网络创建时，它根据缺省的参数自动初始化。init不需要单独的调用。可是我们可能要重新初始化权重和偏置或者进行自定义的初始化。例如，我们用newff创建的网络，它缺省用initnw来初始化第一层。如果我们想要用rands重新初始化第一层的权重和偏置，我们用以下命令：

　　net.layers{1}.initFcn = 'initwb';

　　net.inputWeights{1,1}.initFcn = 'rands';

　　net.biases{1,1}.initFcn = 'rands';

　　net.biases{2,1}.initFcn = 'rands';

　　net = init(net);

IW: 输入层到隐含层的权重矩阵

LW: 隐含层和输出层间的权重矩阵

b: 阀值向量

如网络为net, 输入层和输出均为一个接点情况下，则用

net.IW{1,1}可以看到第一个输入接点到第一隐含层的权重向量；

net.LW{2,1}可以看到隐含层到输出层的权值向量；

net.b{1,1}是隐含层的阀值向量，

net.b{2,1}是输出接点的阀值；

在多输入输出下先用

net.IW

net.LW

net.b

查看各矩阵结构，再相应用net.IW{?,?}等语句查到相关的向量

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

clear all;

%define the input and output

p= [974 874 527;

388 466 1764;

1316 2439 2251;

1836 2410 1860;

1557 2301 1578;

1490 1877 2749;

1513 1278 2026;

1070 1561 2794;

1347 2415 3306;

1324 2746 1233;

1383 1463 1847;

1282 0 2347];

t=[19797 24282 34548];

% 创建bp网络和定义训练函数

% 这里是为了方便而建立一个矩阵，注意是12x2，不是3x2

pr=[ 527 974;

388 1764;

1316 2439;

1836 2410;

1557 2301;

1490 2749;

1278 2026;

1070 2794;

1347 3306;

1233 2746;

1383 1847;

0 2347]

net=newff(pr,[15,1],{'tansig' 'purelin'},'trainlm'); %这里要加入输出层的转移函数，一般是trainlm

net.trainparam.goal=50;

net.trainparam.epochs=5000;

%训练神经网络

[net,tr]=train(net,p,t);

%输出训练后的权值和阈值

iw1=net.IW{1};

b1=net.b{1};

lw2=net.LW{2};

b2=net.b{2};

%存储训练好的神经网络

save netkohler net

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

怎样知道matlab已经训练好的神经网络的权值、步长以及阙值

用matlab训练神经网络时不需输入权值、步长以及阙值，如果我想知道matlab已经训练好的神经网络的权值、步长以及阙值该怎末操作？

训练好的权值、阈值的输出方法是：

输入到隐层权值： w1=net.iw{1,1}

隐层阈值： theta1=net.b{1}

隐层到输出层权值： w2=net.lw{2,1};

输出层阈值： theta2=net.b{2}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

帮我看看matlab的这段程序(有关神经网络BP算法）

在一位老师的讲义上看到的程序，但是有些东西不太明白，请求帮助

程序如下：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

clf;

figure(gcf)

echo on

%NEWFF — 建立一个BP网络

%TRAIN — 对BP网络进行训练

%SIM — 对BP网络进行仿真

pause

P = -1:0.1:1;

T = [-.9602 -.5770 -.0729 .3771 .6405 .6600 .4609 ...

.1336 -.2013 -.4344 -.5000 -.3930 -.1647 .0988 ...

.3072 .3960 .3449 .1816 -.0312 -.2189 -.3201];

plot(P,T,'+');

title('Training Vectors');

xlabel('Input Vector P');

ylabel('Target Vector T');

pause

net=newff(minmax(P),[5 1],{'tansig' 'purelin'},'traingd','learngd','sse');

echo off

k = pickic;

if k == 2

net.iw{1,1} = [3.5000; 3.5000; 3.5000; 3.5000; 3.5000];

net.b{1} = [-2.8562; 1.0774; -0.5880; 1.4083; 2.8722];

net.lw{2,1} = [0.2622 -0.2375 -0.4525 0.2361 -0.1718];

net.b{2} = [0.1326];

end

net.iw{1,1}

net.b{1}

net.lw{2,1}

net.b{2}

pause

echo on

me=8000;

net.trainParam.show=10;

net.trainParam.goal=0.02;

net.trainParam.lr=0.01;

A=sim(net,P);

sse=sumsqr(T-A);

for i=1:me/100

if sse>net.trainparam.goal,i=i-1;break,end

net.trainParam.epochs=100;

[net,tr]=train(net,P,T);

trp((1+100\*(i-1)):(max(tr.epoch)+100\*(i-1)))=tr.perf(1:max(tr.epoch));

A=sim(net,P);

sse=sumsqr(T-A);

plot(P,T,'+');

hold on

plot(P,A)

hold off

pause

end

message=sprintf('Traingd, Epoch %%g/%g, SSE %%gn',me);

fprintf(message,(max(tr.epoch)+100\*(i-1)),sse)

plot(tr)

[i,j]=size(trp);

hold on

plot(1:j,net.trainParam.goal,'r--')

hold off

title('Error Signal')

xlabel('epoch')

ylabel('Error')

p = 0.5;

a = sim(net,p)

echo off

### 12 edge

BW = edge(I) 采用I作为它的输入，并返回一个与I相同大小的[二值化](http://baike.baidu.com/view/983308.htm)图像BW，在函数检测到边缘的地方为1，其他地方为0。

BW = edge(I,'[sobel](http://baike.baidu.com/view/4884628.htm)') 自动选择[阈值](http://baike.baidu.com/view/409216.htm)用[Sobel算子](http://baike.baidu.com/view/676368.htm)进行[边缘检测](http://baike.baidu.com/view/178264.htm)。

## 三、输出图像有关（画图等）

### 1 Subplot

subplot(4,4,2); 在一个窗口显示4\*4个子图，第2个子图；

imshow(fig/255);title(['Àà±ð£º' num2str(label)]) 给每个子图加标题

suptitle('怎么样'); 加总标题

figure,

a=1;

for s = 1:3

for j = 1:3

subplot(3,3,(s-1)\*3+j);

% title(sprintf('采样 Sample'));

% imshow( c{a})

imagesc(c{a}); colormap gray; axis image; axis off

a=a+1;

end

end

## 四、图像（2D）处理相关

### 1 imread，imshow（读图片函数）

img=imread('1.bmp'); title(sprintf('采样 Sample'));

### 2 rgb2gray（灰度处理）

img2=rgb2gray(img);

### 3 im2double

img3=im2double(img);

如果输入是 uint8 unit16 或者是二值的logical类型，则函数im2double 将其值归一化到0～1之间。

double 就是简单地把一个变量的类型转换成double型，数值大小不变，比如一个unit8类型变量a=1，double(a)的结果还是6，不过现在a已经是double型了

### 4 imresize

img=imread('1.tiff');

img2=imresize(img,[60,60]);

img2=imresize(img,2);

## 五那些随机函数

### 1 randperm

randperm是matlab函数，功能是随机打乱一个数字序列。  
　　函数功能：随机打乱一个数字序列。  
　　语法格式：  
　　y = randperm(n)  
　　y是把1到n这些数随机打乱得到的一个数字序列。

randperm(100,5) 将前100个数中，随机选择5个。所以randperm(n,m)中，n一定大于等于m。

train\_rowrank = randperm(size(train\_, 1)); % 随机打乱矩阵的行数

### 2 rand('state',0)

rand('state',0);

## 六活用打印输出函数

### 1printf()函数

a='this number is';

b=123456;

fprintf('%s %f,',a,b); %直接输出到屏幕；类似于C语言的输出格式

### 2disp

disp(['epoch ' num2str(i) '/' num2str(opts.numepochs) '. Took ' num2str(t) ' s']);

## 七与画图有关

### text

参考：[matlab中text 函数在显示字符串时的使用方法](http://blog.sina.com.cn/s/blog_d5d8dbfd0101g2sc.html)

### plpt

参考：[Matlab绘图](http://blog.sina.com.cn/s/blog_d5d8dbfd0101g2zv.html)

## 七GPU加速问题

1

%选择GPU设备

### 2 gpuDeviceCount

%返回设备上的GPU数目