

作业 3：利用LeNet-5实现手写数字识别

学号：19049100002 姓名：张泽群 任课老师：李宇楠 课程号：CS205201

1. 实现概要

1.1 编程语言

使用matlab(2021b)实现，重点使用matlab deep learning toolbox。

1.2 实现步骤

- (1) 对mnist数据集进行数据预处理。
- (2) 利用处理后数据集，基于matlab deep learning toolbox对LeNet-5网络进行训练，并计算准确率accuracy。
- (3) 自制几张手写数字png图片，对训练后网络进行样本测试。

2. Mnist数据集预处理

2.1 简介

THE MNIST DATABASE of handwritten digits

[Yann LeCun](#), Courant Institute, NYU
[Corinna Cortes](#), Google Labs, New York
[Christopher J.C. Burges](#), Microsoft Research, Redmond

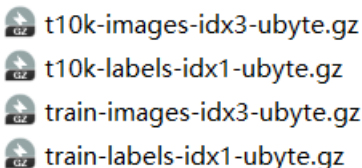
Please refrain from accessing these files from automated scripts with high frequency. Make copies!

The MNIST database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting.

MNIST 手写数字数据库是纽约大学柯朗研究所提供的一个，包含 60,000 个示例的训练集和一个包含 10,000 个示例的测试集，数字已经过大小标准化并在固定大小的图像中居中。

对于想要在现实世界数据上尝试学习技术和模式识别方法，同时在预处理和格式化上花费最少的精力的人来说，这是一个很好的数据库。它包含以下四个文件：



- t10k-images-idx3-ubyte.gz
- t10k-labels-idx1-ubyte.gz
- train-images-idx3-ubyte.gz
- train-labels-idx1-ubyte.gz

解压后为.idx3-ubyte的格式。

2.2 处理过程

我们利用斯坦福大学Andrew Ng教授提供的loadMNISTImages和loadMNISTLabels函数，可以获得一个 $28 \times 28 \times n$ 的返回.mat文件用于提取mnist数据集中Images信息，和一个 $n \times 1$ 的返回.mat文件用于提取mnist数据集中Labels信息。

因为matlab_2016b后支持，为图像数据集合创建 ImageDatastore 对象，该对象的子文件夹名即可代表该类图像的标签，所以我们可以将mnist数据预处理并创建 ImageDatastore 对象，以便于后续的训练。

根据loadMNISTImages和loadMNISTLabels函数获得Images数据和Labels数据，我们可以创建2个文件夹以及10个子文件夹并将数据分类存放。（实现过程如下所示）

2.3 实现以及结果

注：利用loadMNISTImages和loadMNISTLabels函数，获得Images数据和Labels数据。

```
%% load data
imageDim = 28;
% 训练数据
train_Images = loadMNISTImages('train-images.idx3-ubyte');
train_Images = reshape(train_Images, imageDim, imageDim, []);
train_Labels = loadMNISTLabels('train-labels.idx1-ubyte');
train_Labels(train_Labels == 0) = 10;
% 测试数据
test_Images = loadMNISTImages('t10k-images.idx3-ubyte');
test_Images = reshape(test_Images, imageDim, imageDim, []);
test_Labels = loadMNISTLabels('t10k-labels.idx1-ubyte');
test_Labels(test_Labels == 0) = 10;
```

注：对60000条训练数据和10000条测试数据预处理，使对应图像分别进入对应标签的子文件夹。

```
for i=1:60000
    if train_Labels(i) == 10

        imwrite(train_Images(:,:,i), sprintf('%s%d%s', './train_data/0/', i, '.png')
    );

        ...
    elseif train_Labels(i) == 9

        imwrite(train_Images(:,:,i), sprintf('%s%d%s', './train_data/9/', i, '.png')
    );

end
```

```

end
end

%% 对10000条测试数据处理，使对应图像分别进入对应标签的子文件夹
for i=1:10000
    if test_Labels(i) == 10

        imwrite(test_Images(:,:,i),sprintf('%s%d%s', './test_data/0/', i, '.png'))
    ;

        ...
    elseif test_Labels(i) == 9

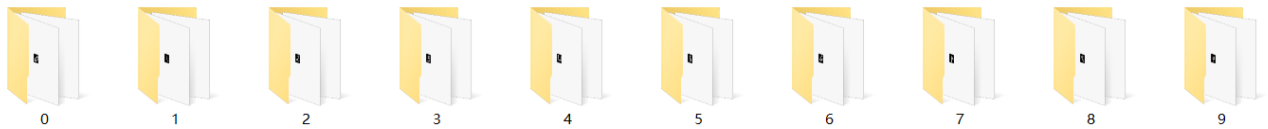
        imwrite(test_Images(:,:,i),sprintf('%s%d%s', './test_data/9/', i, '.png'))
    ;

    end
end
end

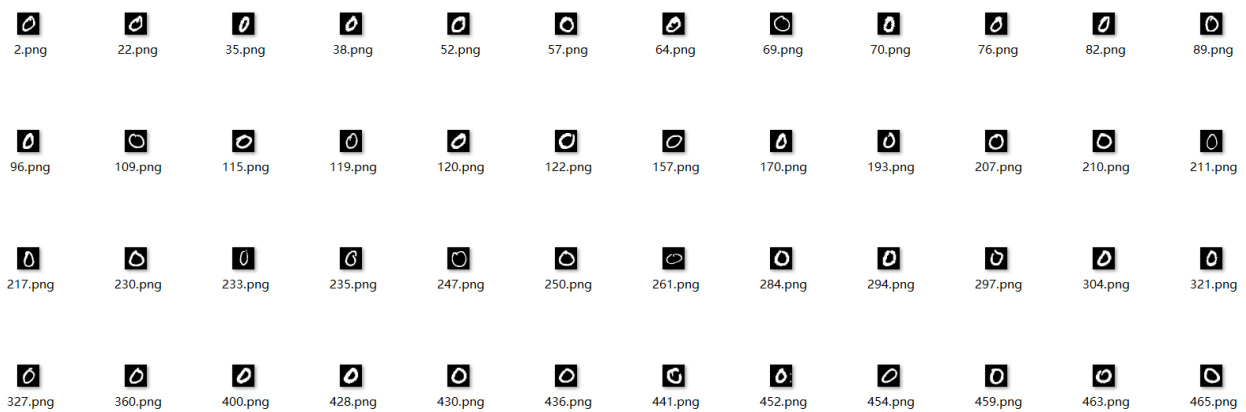
```

结果如下所示：

结果分为2个大文件夹，分别包含10个子文件夹(0-10)



以 0 为例：



3. 训练LeNet-5网络，计算准确率 *accuracy*

3.1 简介

LeNet-5 是 [Yann LeCun](#) 等人在1998年设计的用于手写数字识别的卷积神经网络，当年美国大多数银行就是用它来识别支票上面的手写数字的，它是早期卷积神经网络中最有代表性的实验系统之一。

3.2 原理与计算过程

LeNet-5共有7层，不包含输入，每层都包含可训练参数（连接权重）。输入图像为 32×32 大小，这要比数据集中最大的字符要大得多（最多 20×20 个像素位于 28×28 的中心区域）。

C1层是一个卷积层，由6个特征图Feature Map构成。有156个可训练参数（每个滤波器 $5 \times 5 = 25$ 个unit参数和一个bias参数，一共6个滤波器，共 $(5 \times 5 + 1) \times 6 = 156$ 个参数），共122,304个连接（ $28 \times 28 \times 28 \times 6$ ，每个神经元对应26个连接，每个feature map有 28×28 个unit，一共有6个feature map）。

S2层是一个下采样层，有6个 14×14 的特征图，特征图中的每个单元与C1中相对应特征图的 2×2 邻域相连接，输入到激活函数时它们共用一个系数加一个偏置，所需的训练参数为 $(1+1) \times 6 = 12$ 个，连接参数为 $(4+1) \times 6 \times 14 \times 14 = 5880$ 个。

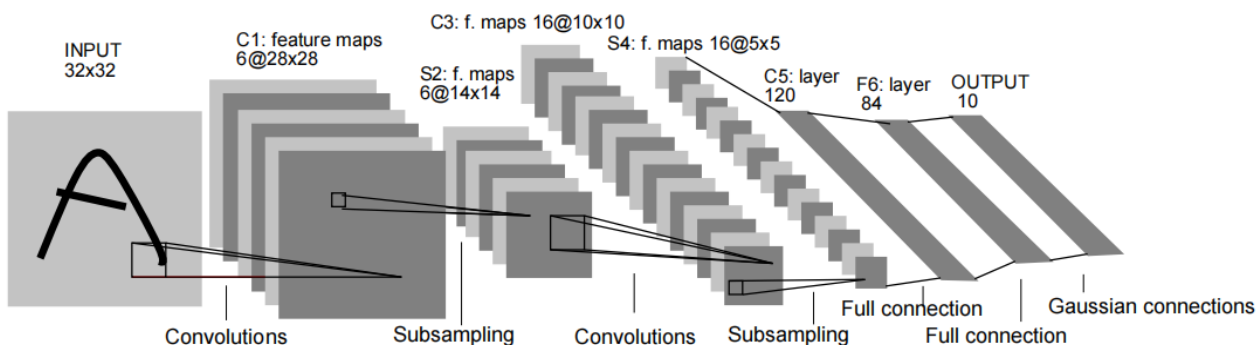
C3层是一个有16个特征图的卷积层，每个特征图对S2中的特征图并非是全连接的。共有 $(25 \times 3 + 1) \times 6 + (25 \times 4) \times 9 + (25 \times 6 + 1) = 1516$ 个训练参数，连接个数为 $1516 \times 10 \times 10 = 151600$ 个。

S4同样为下采样层，有 $16 \times (1+1) = 32$ 个训练参数，有 $(2 \times 2 + 1) \times 25 \times 16 = 2000$ 个连接。

C5层是一个卷积层，有120个特征图。每个单元与S4层的全部16个特征图的 5×5 领域相连。由于S4层特征图的大小也为 5×5 （同滤波器一样），故C5特征图的大小为 1×1 ，这构成了S4和C5之间的全连接。之所以仍将C5标示为卷积层而非全连接层，是因为如果LeNet-5的输入变大，而其他的保持不变，那么此时特征图的维数就会比 1×1 大。C5层有48120个可训练连接 $((5 \times 5 \times 16 + 1) \times 120)$ 。

F6层有84个单元，与C5层全相连。有10164个可训练参数。

最后的OUTPUT有10个，因为共有10种阿拉伯数字。



3.3 实现以及结果

注：加载预处理后的数据

%% 加载数据

```
imdsTrain=imageDatastore('train_Data', 'FileExtensions',  
{'.png','.jpg','.tif'},'IncludeSubfolders',true,'LabelSource','foldernames');  
imdsTest=imageDatastore('test_Data', 'FileExtensions',  
{'.png','.jpg','.tif'},'IncludeSubfolders',true,'LabelSource','foldernames')
```

注：LeNet-5网络的构建，与原理图中一致。

%% LeNet-5 网络

```
layers = [...  
    imageInputLayer([28 28 1],"Name","imageinput")  
  
    convolution2dLayer([5 5],6,"Name","conv1","Padding","same")  
    tanhLayer("Name","tanh1")  
  
    maxPooling2dLayer([2 2],"Name","maxpool1","Stride",[2 2])  
  
    convolution2dLayer([5 5],16,"Name","conv2")  
    tanhLayer("Name","tanh2")  
  
    maxPooling2dLayer([2 2],"Name","maxpool","Stride",[2 2])  
  
    fullyConnectedLayer(120,"Name","fc1")  
    fullyConnectedLayer(84,"Name","fc2")  
    fullyConnectedLayer(10,"Name","fc")  
    softmaxLayer("Name","softmax")  
    classificationLayer("Name","classoutput")  
];
```

注：设置训练参数并进行训练

%% 设置训练参数

```
options = trainingOptions('sgdm', ...  
    'LearnRateSchedule','piecewise', ... %学习率  
    'LearnRateDropFactor',0.2, ...  
    'LearnRateDropPeriod',5, ...  
    'MaxEpochs',20, ... %最大学习整个数据集的次数
```

```
'MiniBatchSize',128, ... %默认一个batch有128个样本
'Plots','training-progress',...
'Verbose',0);
```

%% 训练网络

```
net = trainNetwork(imdsTrain, layers, options);
```

```
save Minist_LeNet5 net
```

注：计算精确度

%% 计算精确度

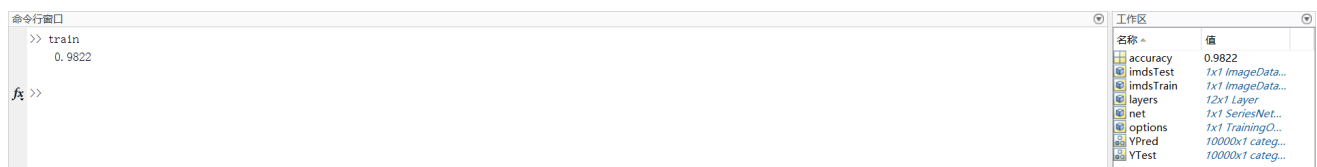
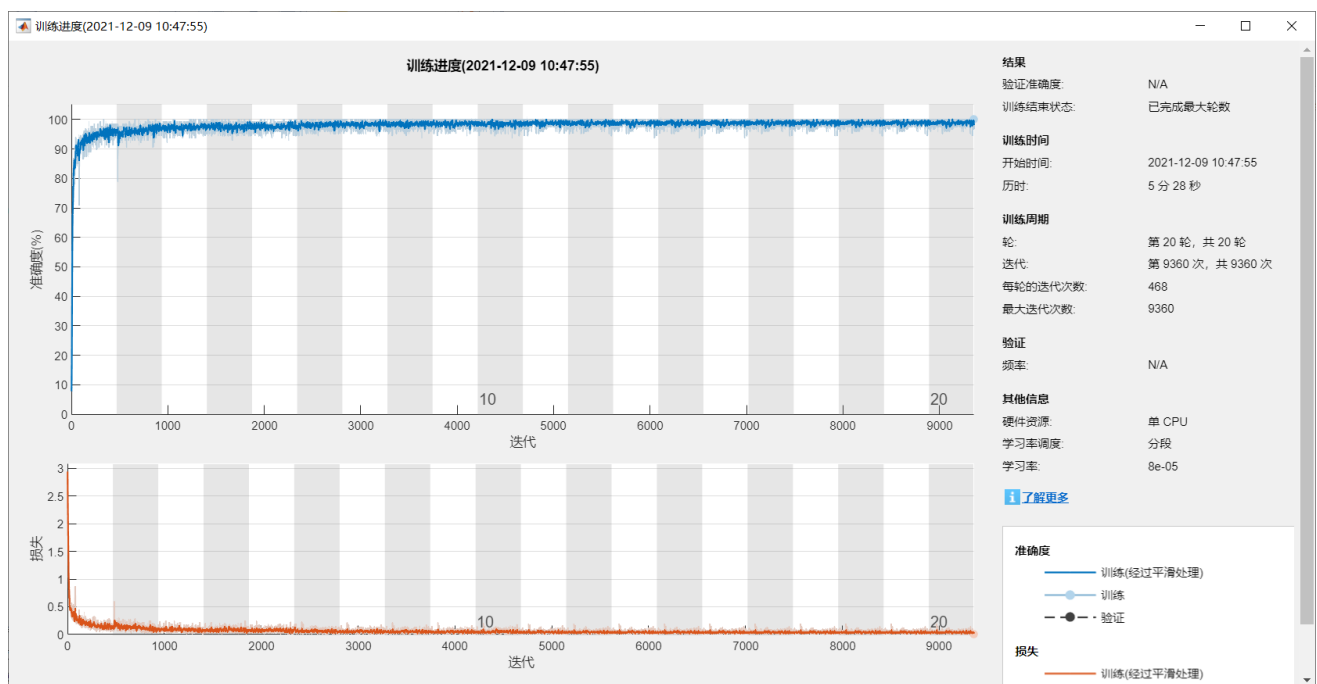
```
YPred = classify(net, imdsTest); %在测试集上输出网络预测结果
```

```
YTest = imdsTest.Labels;
```

```
accuracy = sum(YPred == YTest) / numel(YTest); %网络在测试集的精度
```

```
disp(accuracy);
```

训练过程和精确度结果图 (训练得出网络保存为Minist_LeNet5 net)



4. 样本测试

4.1 原理与计算过程

即在画图工具中手写部分数字(该部分数据存放在test_pic)，利用训练好的网络进行测试。

4.2 实现以及结果

注：注意需要把彩色图灰度化，使得图像为28x28，并且使得图像变为黑色背景，白色字体。

```
figure('Name','手写数字识别测试_19049100002_张泽群','NumberTitle','off');
for i = 1:count
    test_image = readimage(I_test,i);    %导入手写体数字图片
    subplot(3,3,i);
    imshow(I_test.Files{i});
    %输入图像处理
    %1. 把彩色图灰度化
    shape = size(test_image);
    dim=numel(shape);
    if dim > 2
        test_image = rgb2gray(test_image);
    end
    %2. 使得图像为28x28，并且使得图像变为黑色背景，白色字体
    test_image = imresize(test_image, [28,28]);
    test_image = imcomplement(test_image);
    result = classify(net, test_image);
    title(['识别结果: ' char(result)])
end
```

结果如下所示：

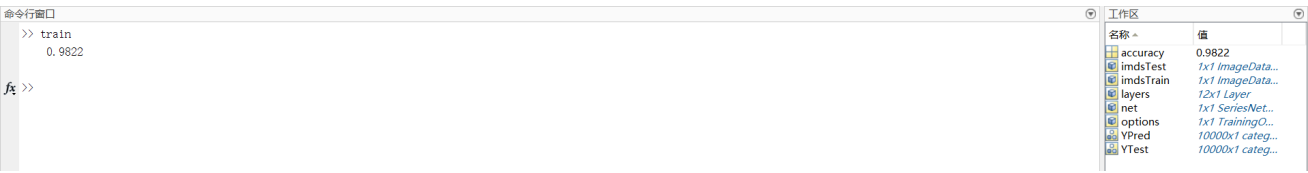


5. 实验结果及最终分析

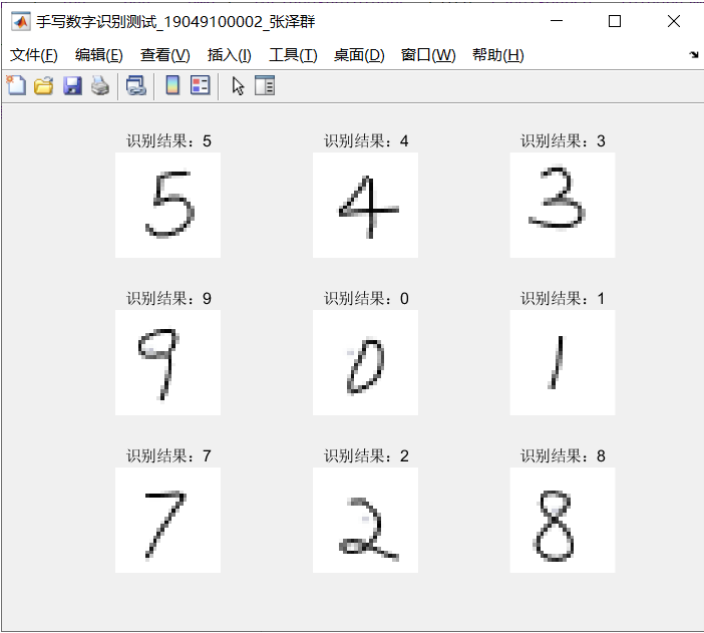
5.1 实验结果

实验最终结果(准确率和样本测试结果)如下图所示：

1. 准确率 $accuracy = 0.9822$



2. 样本测试结果



5.2 最终分析

分析：LeNet-5对mnist数据集训练的准确率没有到达99%，这可能是因为学习率不够小的原因。

样本测试结果正常。

6. 源程序附录

1. dealData.m (用于Mnist数据集预处理)

```
%% load data
imageDim = 28;
% 训练数据
train_Images = loadMNISTImages('train-images.idx3-ubyte');
train_Images = reshape(train_Images,imageDim,imageDim,[]);
train_Labels = loadMNISTLabels('train-labels.idx1-ubyte');
train_Labels(train_Labels == 0) = 10;
% 测试数据
test_Images = loadMNISTImages('t10k-images.idx3-ubyte');
test_Images = reshape(test_Images,imageDim,imageDim,[]);
test_Labels = loadMNISTLabels('t10k-labels.idx1-ubyte');
test_Labels(test_Labels == 0) = 10;

%% 对60000条训练数据处理，使对应图像分别进入对应标签的子文件夹
for i=1:60000
    if train_Labels(i) == 10

        imwrite(train_Images(:,:,i),sprintf('%s%d%s','./train_data/0/',i,'.png'
));
    elseif train_Labels(i) == 1

        imwrite(train_Images(:,:,i),sprintf('%s%d%s','./train_data/1/',i,'.png'
));
    elseif train_Labels(i) == 2

        imwrite(train_Images(:,:,i),sprintf('%s%d%s','./train_data/2/',i,'.png'
));
    elseif train_Labels(i) == 3

        imwrite(train_Images(:,:,i),sprintf('%s%d%s','./train_data/3/',i,'.png'
));
    elseif train_Labels(i) == 4

        imwrite(train_Images(:,:,i),sprintf('%s%d%s','./train_data/4/',i,'.png'
));
    elseif train_Labels(i) == 5

        imwrite(train_Images(:,:,i),sprintf('%s%d%s','./train_data/5/',i,'.png'
));
    elseif train_Labels(i) == 6

        imwrite(train_Images(:,:,i),sprintf('%s%d%s','./train_data/6/',i,'.png'
));
    elseif train_Labels(i) == 7
```

```

    imwrite(train_Images(:,:,i),sprintf('%s%d%s', './train_data/7/', i, '.png'
));
    elseif train_Labels(i) == 8

    imwrite(train_Images(:,:,i),sprintf('%s%d%s', './train_data/8/', i, '.png'
));
    elseif train_Labels(i) == 9

    imwrite(train_Images(:,:,i),sprintf('%s%d%s', './train_data/9/', i, '.png'
));
    end
end

```

%% 对10000条测试数据处理，使对应图像分别进入对应标签的子文件夹

```

for i=1:10000
    if test_Labels(i) == 10

    imwrite(test_Images(:,:,i),sprintf('%s%d%s', './test_data/0/', i, '.png'))
;
    elseif test_Labels(i) == 1

    imwrite(test_Images(:,:,i),sprintf('%s%d%s', './test_data/1/', i, '.png'))
;
    elseif test_Labels(i) == 2

    imwrite(test_Images(:,:,i),sprintf('%s%d%s', './test_data/2/', i, '.png'))
;
    elseif test_Labels(i) == 3

    imwrite(test_Images(:,:,i),sprintf('%s%d%s', './test_data/3/', i, '.png'))
;
    elseif test_Labels(i) == 4

    imwrite(test_Images(:,:,i),sprintf('%s%d%s', './test_data/4/', i, '.png'))
;
    elseif test_Labels(i) == 5

    imwrite(test_Images(:,:,i),sprintf('%s%d%s', './test_data/5/', i, '.png'))
;
    elseif test_Labels(i) == 6

    imwrite(test_Images(:,:,i),sprintf('%s%d%s', './test_data/6/', i, '.png'))
;
    elseif test_Labels(i) == 7

    imwrite(test_Images(:,:,i),sprintf('%s%d%s', './test_data/7/', i, '.png'))
;

```

```

elseif test_Labels(i) == 8

imwrite(test_Images(:,:,i),sprintf('%s%d%s', './test_data/8/', i, '.png'))
;

elseif test_Labels(i) == 9

imwrite(test_Images(:,:,i),sprintf('%s%d%s', './test_data/9/', i, '.png'))
;

end
end

```

2. loadMNISTImages.m (应用于dealData, 提取数据集图像信息)

```

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

```

3. loadMNISTLabels.m (应用于dealData, 提取数据集标签信息)

```
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
```

4. train.m (训练LeNet-5网络, 计算准确率)

```
%% 加载数据

imdsTrain=imageDatastore('train_Data', 'FileExtensions',
{'.png', '.jpg', '.tif'}, 'IncludeSubfolders', true, 'LabelSource', 'foldernames');
imdsTest=imageDatastore('test_Data', 'FileExtensions',
{'.png', '.jpg', '.tif'}, 'IncludeSubfolders', true, 'LabelSource', 'foldernames');

%% LeNet-5 网络
layers = [...
    imageInputLayer([28 28 1], "Name", "imageinput")

    convolution2dLayer([5 5], 6, "Name", "conv1", "Padding", "same")
    tanhLayer("Name", "tanh1")

    maxPooling2dLayer([2 2], "Name", "maxpool1", "Stride", [2 2])

    convolution2dLayer([5 5], 16, "Name", "conv2")
    tanhLayer("Name", "tanh2")

    maxPooling2dLayer([2 2], "Name", "maxpool", "Stride", [2 2])
```

```

    fullyConnectedLayer(120,"Name","fc1")
    fullyConnectedLayer(84,"Name","fc2")
    fullyConnectedLayer(10,"Name","fc")
    softmaxLayer("Name","softmax")
    classificationLayer("Name","classoutput")
];

%% 设置训练参数
options = trainingOptions('sgdm', ...
    'LearnRateSchedule','piecewise', ... %学习率
    'LearnRateDropFactor',0.2, ...
    'LearnRateDropPeriod',5, ...
    'MaxEpochs',20, ... %最大学习整个数据集的次数
    'MiniBatchSize',128, ... %默认一个batch有128个样本
    'Plots','training-progress',...
    'Verbose',0);

%% 训练网络
net = trainNetwork(imdsTrain, layers, options);

save Minist_LeNet5 net

%% 计算精确度
YPred = classify(net, imdsTest); %在测试集上输出网络预测结果

YTest = imdsTest.Labels;
accuracy=sum(YPred == YTest)/numel(YTest); %网络在测试集的精度
disp(accuracy);

```

5. test.m (样本测试)

```

%% 测试手写数字
load Minist_LeNet5 net; %导入训练好的LeNet5网络

I_test=imageDatastore('test_pic','FileExtensions',
    {' .jpg', ' .png', ' .bmp'}, 'IncludeSubfolders', false, 'LabelSource', 'foldernames');

count=numel(I_test.Files); %test_data文件夹中的文件个数

%% 画图并识别
figure('Name','手写数字识别测试_19049100002_张泽群','NumberTitle','off');
for i = 1:count
    test_image =readimage(I_test,i); %导入手写体数字图片
    subplot(3,3,i);

```

```
imshow(I_test.Files{i});

%输入图像处理

%1. 把彩色图灰度化
shape = size(test_image);
dim=numel(shape);
if dim > 2
test_image = rgb2gray(test_image);
end

%2. 使得图像为28x28，并且使得图像变为黑色背景，白色字体
test_image = imresize(test_image, [28,28]);
test_image = imcomplement(test_image);

result = classify(net, test_image);

title(['识别结果: ' char(result)])
end
```