

实验五 基于深度学习的模式识别实验

实验目的：

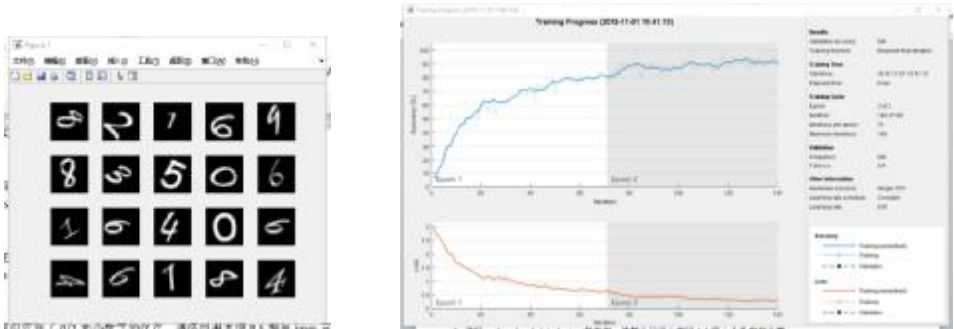
通过构建网络模式识别实例，理解深度学习的结构和原理。

实验内容：

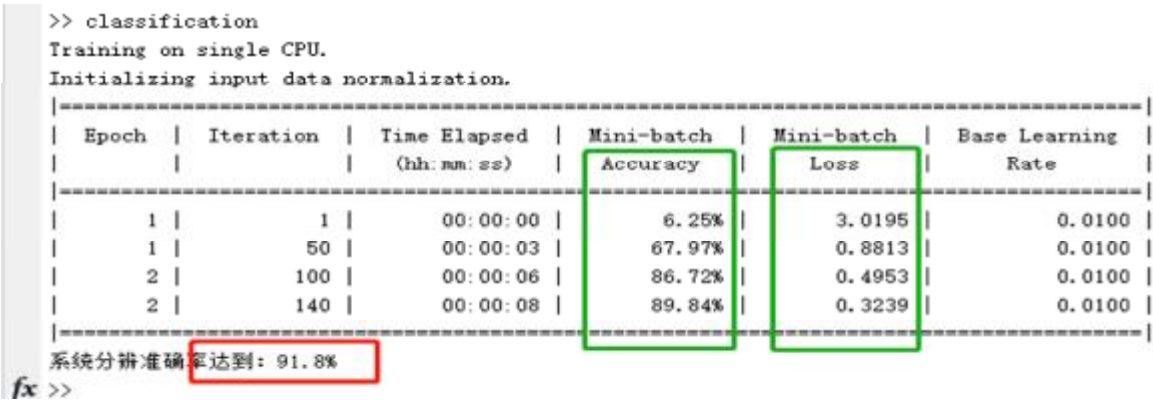
搭建深度学习模型，实现手写数字图像的分类。
配置参数并直接运行 `classification` 命令并观察结果。

实验要求：

- 1. 把 `rar` 文件所有内容复制到硬盘文件夹中，在 `matlab` 命令窗口进入到该文件夹
- 2. 执行命令 `classification` 运行程序，系统会弹出训练样本的图样以及训练过程的误差曲线



并可以在命令窗口观察到训练过程数据以及测试的结果输出（可以观察到下例中，绿色框训练准确率在不断提升，绿色框训练误差在不断减少；训练结束后最终红色框测试准确率为 91.8%）：



- 3. 保持其他参数不变，修改 `MaxEpochs` 的数值，观察并记录准确率的变化（`MaxEpochs` 数值的修改在原文件的 54-64 行附近），并填写表格一。（表格中已填写数值为根据上图观察数据的示例）

```
54 % 指定训练选项。使用具有动量的随机梯度下降
55 % 将最大训练轮数设置为 1。一轮训练是对整个t
56 % 通过指定验证数据和验证频率，监控训练过程
57 % 每轮训练都会打乱数据。软件基于训练数据训练
58 % 验证数据不用于更新网络权重。打开训练进度
59 Options = trainingOptions('sgdm', ...
60     'InitialLearnRate',0.01, ...
61     'MaxEpochs',2, ...
62     'Verbose',true, ...
63     'Plots','training-progress');
64
```

表格一	MaxEpochs								
	1	2	3	4	5	6	7	8	9
最后一次训练准确率		89.84							
测试准确率		91.8							

4. 把 MaxEpochs 的数值设置为 2，并把 **..., 'Plots', 'training-progress'** 的字样删除，如下图所示：

```
59 options = trainingOptions('sgdm', ...
60     'InitialLearnRate', 0.01, ...
61     'MaxEpochs', 2, ...
62     'Verbose', true);
63
```

修改卷积神经网络架构定义中 `convolution2dLayer` 的第一个参数 `filterSize` 和第二个参数 `numFilters`。（在卷积层中，第一个参数是 `filterSize`，它是训练函数在沿图像扫描时使用的卷积核的高度/宽度。在此示例中，数字 3 表示卷积核大小为 3×3 。第二个参数是卷积核数量 `numFilters`，它是连接到同一输入区域的卷积核数量。此参数决定了特征图的数量。使用 `'Padding'` 名称-值对组对输入特征图进行填充。对于默认步幅为 1 的卷积层，`'same'` 填充可确保空间输出大小与输入大小相同。）

```
40 %定义卷积神经网络架构
41 layers = [
42     imageInputLayer([dim dim 1])
43     convolution2dLayer(3, 8, 'padding', 'same')
44     batchNormalizationLayer
45     reluLayer
46 ]
```

把设置两个参数不同的数值，把观察到的测试准确率的数值填到表格二中。（提示：你可以另外写一段代码，把 `filterSize` 和 `numFilters` 作为变量，写循环语句执行并记录最终结果到一个矩阵中）

表格二 (测试准确率)		numFilters 卷积核数量								
		1	2	3	4	5	6	7	8	9
filterSize 卷积核 的高度/ 宽度	1									
	2									
	3									
	4									
	5									

5. 增加深度学习框架中隐含层的数量，如下左图所示：

```
40 %定义卷积神经网络架构
41 layers = [
42     imageInputLayer([dim dim 1])
43     convolution2dLayer(2, 4, 'padding', 'same')
44     batchNormalizationLayer
45     reluLayer
46 ]
47
48 maxPooling2dLayer(2, 'Stride', 2)
49
50 convolution2dLayer(2, 4, 'padding', 'same')
51 batchNormalizationLayer
52 reluLayer
53
54 fullyConnectedLayer(size(labelCount, 1))
55 softmaxLayer
56 classificationLayer];
```

```
40 %定义卷积神经网络架构
41 layers = [
42     imageInputLayer([dim dim 1])
43     convolution2dLayer(2, 4, 'padding', 'same')
44     batchNormalizationLayer
45     reluLayer
46 ]
47
48 maxPooling2dLayer(2, 'Stride', 2)
49
50 convolution2dLayer(2, 4, 'padding', 'same')
51 batchNormalizationLayer
52 reluLayer
53
54 maxPooling2dLayer(2, 'Stride', 2)
55
56 convolution2dLayer(2, 4, 'padding', 'same')
57 batchNormalizationLayer
58 reluLayer
59
60 fullyConnectedLayer(size(labelCount, 1))
61 softmaxLayer
62 classificationLayer];
```

上图的网络框架中，增加了红色部分（图中，红色部分与绿色部分一样），使得网络框架加深。也可以继续增加网络深度，如上右图所示。同样，每一行的 `convolution2dLayer` 的参数都可以修改成不同的数值。请自行观察不同设置下（深度+参数）对最终结果的影响。

6. 想深入一点学习的同学，可以理解代码中 `putNoiseDigit`，`toEditPredDigit`，`toShowDigitTest` 等变量的功能。注意：`toEditPredDigit` 设置为 1 时，系统容易出错，并影响原始数据，必要时，可以把所有程序及文件夹用 `rar` 文件恢复原样。