DDL: 1.9

本课程作业将使用MNIST手写数字数据集的子集（将提供），该数据集在实验室会话中由您准备（请参见附录中的实验材料），并且您将执行以下任务：

强制部分（您必须完成此部分）

设计并实现一个单层感知器，以区分数字6和其他9个数字，即当输入为6时，输出将为+1，否则为-1。按照以下准则进行实验：

• 将数据分为训练集和测试集。

• 绘制至少4张图表（例如，2组不同的初始权重，2种不同的训练速率），表示训练集上的随时间变化的错误（y轴上的错误分类百分比，x轴上的训练时期）。

• 为测试集绘制相应的图表。

解释您用于终止训练的停止准则。

提交内容

提交电子副本（代码和报告）。您的报告应包括：

• 您实施的简要解释

• 结果和结果解释

• 结论和反思（或通过完成此课程作业学到的关于目标识别的知识）

演示

您必须向助教演示您的实现。

可选部分（您不必完成此部分，但强烈建议尝试）

使用所需数量的感知器设计一个模式识别系统，将数字分为4组：(0, 1) (2, 3,) (4, 5, 6) (7, 8, 9)。解释您的实现和结果。

机器学习实验1

在实验室会话期间，我们将在MNIST数字数据集上进行连续实验。最终目标是使用Matlab构建一个数字识别系统。

任务1：开始使用Matlab

Matlab，即Matrix Laboratory的缩写，可能是最广泛使用的科学和工程数值软件。要开始使用，您可以观看以下视频：

http://www.mathworks.co.uk/products/featured/videos/index.html

或阅读以下书籍：

<http://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf>

任务2：加载并显示MNIST数据集

1. MNIST数字识别数据集是机器学习中最广泛使用的数据集之一。它包含60,000个训练样本和10,000个测试样本。您可以从http://yann.lecun.com/exdb/mnist/下载它。

2. 请仔细阅读其中关于数据存储结构的说明。尝试将数据加载到Matlab并显示。训练集中的前10个数字应该如下：



提示：

1. 在加载MNIST数据时，您可能会依赖Matlab中的以下函数：fopen、fread和fclose。

2. 在显示数据时，您将需要使用以下函数：reshape、imagesc和imshow。

3. 只需在Matlab中键入'help XXX'，即可查看如何使用上述函数。

4. 如果在Matlab中遇到“内存不足”问题，这意味着数据太大而无法适应内存，您必须在工作区中删除不必要的变量或将它们保存到硬盘中。

机器学习实验2

数据预处理：

1. 首先，请重复实验1，将MNIST数据集加载到Matlab中。由于原始MNIST数据集包含用于训练的60,000个数字，这对我们的实验室会话来说太大了。因此，我们只使用前1000个数字的子集进行训练。基于实验1的示例代码，您可以通过使用以下代码实现：

tr\_label= tr\_label(1:1000,1);

tr\_feats= tr\_feats(1:1000,:);

为了在将来的实验室中避免重复上述过程，您可以直接从以下网址下载这个小数据集

http://www.cs.nott.ac.uk/~qiu/Teaching/G53MLE/Labs/MNIST\_subset.mat

2. 为了对我们即将处理的数据有一个大致了解，您可以计算训练集中包含多少样本。您可以使用以下代码：

for i=0:9

num\_of\_sample(i+1)=size(find(tr\_label==i),1);

end

如果查看变量num\_of\_sample，您会看到每个数字的样本数量大致相同。您还可以尝试显示训练集中包含的每个数字的所有样本。(通过imshow实现？)

1. 有时需要对数据进行归一化，以使它们更具可比性。归一化的目的是使每个数字的欧几里德范数等于1。您可以在以后的实验室会话中自行判断是否需要进行这一标准化步骤。

什么是欧几里得范数？

http://en.wikipedia.org/wiki/Euclidean\_norm#Euclidean\_norm

机器学习实验3

设计感知器：

然后，您的任务是实现感知器算法。请尝试将您的算法封装为一个函数。函数的一个可能形式可以接受两个输入参数，一个是特征向量（tr\_feats），另一个是标签（tr\_label），它将输出学到的权重：

function weight=Perceptron(feats,label)

实现感知器函数后，尝试完成以下任务：

1. 将数字‘2’视为正类，数字‘5’视为负类，学习一个感知器。尝试可视化最有可能是‘2’或最有可能是‘5’的样本。还尝试可视化被错误分类的样本。
2. 现在考虑四个数字，‘2’、‘5’、‘1’和‘7’。如果我们认为‘2’和‘5’在视觉上更相似，‘1’和‘7’也是如此，那么我们可以设计一个‘父’感知器，将‘2’和‘5’视为正类，‘1’和‘7’视为负类。我们还可以设计两个‘子’感知器，一个用于区分‘2’和‘5’，另一个用于区分‘1’和‘7’。因此，总共我们需要三个感知器来分类这四个数字。您有其他想法吗？尝试实施它们并比较结果。
3. 现在考虑所有10个数字。请自行思考您需要多少感知器来识别它们。然后实施它！