**Python编程及算法导论**

**课题：创建MLP神经网络实现mnist数字手写体识别**

**一、python文件介绍**

network.py: 神经网络类，提供神经网络搭建，训练，测试方法。

train\_model.py：训练神经网络，并存储训练参数。

evaluate\_model.py：测试神经网络。

mnist\_loader.py：mist data loader

model\_biases400.pkl：存储训练了400 Epochs的biases参数。

model\_weights400.pkl：存储训练了400 Epochs的weights参数。

data/mnist.pkl.gz：手写数字训练数据

**二、Network神经网络方法描述：**

***-cost\_derivative***

基于当前训练参数(w, b), 针对一个训练数据(x, y)，通过前置神经网络计算会得到一个输出activation，尺寸为(1, 10)，向量中每一个位置的值[0, 1]代表神经网络对于输入为某一个数字的确信程度，最大值所代表的数字代表当前神经网络的最终预测值。

计算当前输出activation与实际标签y相比的损失，使用二次损失，公式如下：

(1)

计算损失相对于a的偏导数，公式如下：

(2)

***-evaluate***

依据当前训练的参数(w, b)，针对test中每一组数据(x, y)，进行前向传播，得到一个输出activation，尺寸为(1, 10)，向量中每一个位置的值[0, 1]代表神经网络对于输入为某一个数字的确信程度，最大值所代表的数字代表当前神经网络的最终预测值。比较这个预测值与标签值所代表的数字是否相等，得到本次预测是否正确。并计算所有正确预测的数字返回。

***-backprop***

计算损失函数关于每一个网络参数 (w, b)的偏导。

基于当前的训练参数(w, b)，针对训练数据中的一组输入(x, y)，进行前向传播，记录每个节点的z值及a值。

然后进行反向传播，从输出层开始，计算输出层“错误”值（BP1），同时计算损失函数对于输出层参数的偏导(BP2, BP4)。

(BP1)

(BP2)

(BP3)

(BP4)

开始反向传播，反向传播传播的是“错误”值，传播的公式为(BP2)，只要有了每一层每一个节点的“错误”值，都可以根据(BP3, BP4)进行计算。

***-gradient\_descent***

针对一个输入mini\_batch，反复计算mini batch中每一个training data作为输入时，损失函数相对于参数(w, b)的偏导，并求平均，得到这个mini\_batch下的平均偏导数，用来近似对于整体训练数据的偏导数。利用这个偏导数完成参数的更新。

(3)

(4)

***-SGD***

Mini-batch随机梯度下降最外层循环，控制Epoch数量，学习率，mini-batch尺寸等超参数。每个Epoch随机打乱全部训练数据（每个Epoch mini-batch数据不一样），并按照预定尺寸分成若干个mini-batch。循环调用gradient\_descent更新参数。每一个Epoch结束时，使用test数据集测试预测准确度，并进行下一次Epoch循环。如果准确度达到要求，可以Ctrl-C提前退出。

***-save\_params/load\_params***

使用pickle库保存当前训练数据或加载预训练的数据。

***-feedforward***

前置神经网络，针对一个输入数据x，计算输出值。

**三、Helper function帮助函数说明**

***-sigmoid***

计算输入numpy数组的sigmoid激活值（逐项计算），公式：

(5)

***-sigmoid\_derivative***

计算sigmoid函数在输入numpy数组z处的梯度，公式：

(6)

**四、mnist\_loader模块说明**

***-load***

加载给定的mnist.pkl.gz数据库。已经给定。

***-load\_data***

帮助函数，将load返回转换成易于训练的格式。如：训练数据由长度为50000个2-tuple组成，每个训练样本为一个2-tuple，第一个元素为shape (1, 784)的numpy数组，代表图片；第二个元素为one-hot编码的图片标签。

把标签数据转换为one-hot编码的标签，比如y=4，转换为np\_array [[0, 0, 0, 0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0]]。

返回Trainingdata和testdata进行训练与测试。

具体步骤：

1. 测试数据加载、转换与可视化。

mnist\_loader.py

1. 完成network类中，feedforward函数体，实现神经网路预测功能。测试随机初始化网络，test\_random.py脚本，请调整test\_idx测试多个数据。并统计准确率。
2. 完成MLP神经网络训练，并使用train\_model.py脚本训练。使用sigmoid激活函数，二次损失。