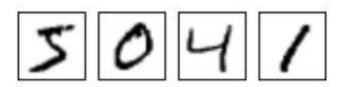
BP 算法识别 MNIST 手写数字集

目标

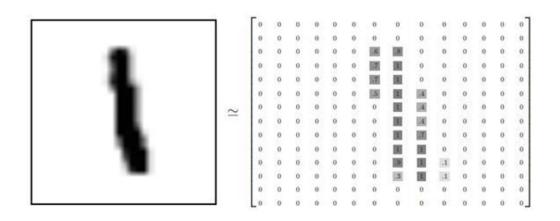
完善 BP 算法识别 MNIST 手写数字集代码

MNIST 介绍

MNIST 是深度学习的经典入门数据集,由 6 万张训练图片和 1 万张测试图片构成的,每张图片都是 28*28 大小(如下图),而且都是黑白色构成(这里的黑色是一个 0-1 的浮点数,黑色越深表示数值越靠近 1),这些图片是采集的不同的人手写从 0 到 9 的数字。如:



上图就是 4 张 MNIST 图片。这些图片并不是传统意义上的 png 或者 jpg 格式的图片,它们被处理成很简易的二维数组,如图:



数据格式:数据存储在一个非常简单的文件格式中,这种格式用来存储向量和多维矩阵。如下图所示:

train-images-idx3-ubyte: training set images
train-labels-idx1-ubyte: training set labels
t10k-images-idx3-ubyte: test set images
t10k-labels-idx1-ubyte: test set labels

更多 MNIST 信息参考: http://yann.lecun.com/exdb/mnist/

提供的文件

- ①MNIST data: 包含了上文数据格式部分介绍的四个文件;
- ②load samples.py: 加载 MNIST data 的数据,以便训练和测试;
- ③NeuralNet.py: BP 网络实现代码;
- ④main.py: 可执行文件,运行该文件可以直接调用 load_samples.py 和 NeuralNet.py.

作业内容

- ①完善 NeuralNet.py 中的 Sigmoid 函数 提示: z 为函数输入值, return 其 Sigmoid 函数计算结果。
- ②完善 NeuralNet.py 中的 sigmoid 导数函数 提示: z 为函数输入值, return 其导函数计算结果。
- ③完善 NeuralNet.py 中 BP 网络的 feedforward 函数

提示: NeuralNet 类中第一个函数__init__分别产生了两套 w 和 b, 存储在 self.w_和 self.b_中。self.w_的两个 w 维度分别为 40*784,10*40; self.b_的两个 b 维度分别为 40*1,10*1。return 内容为 feedforward 最后一层的计算结果。

前三题的代码请直接在 NeuralNet.py 中填写(需要补充代码的部分已用两行#隔出)

- ④解释 NeuralNet.py 中 BP 网络 backprop 函数的代码(已在文件中标明)
- ⑤完成①-③后运行 main.py(python main.py),给出最后的计算结果。 提示: py 脚本按 python3 版本编写, python2 运行可能会出错。

部分函数说明

def update mini batch(self, mini batch, eta): 更新权重

def cost_derivative(self, output_activations, y): 返回 output_activations 和 y 之差 def load_samples(dataset="training_data"): 根据参数加载数据集。并将图像和标签对应起来

def load_mnist(dataset="training_data", digits=np.arange(10), path="."): 根据参数 从文件中读取相关数据

def evaluate(self, test_data):根据测试数据集对模型进行评估,看有多少识别结果和标签一致

请在 3 月 27 日 10:00 前完成作业并发送到助教邮箱: 496674467@qq.com