

神经网络

作业四

181220076, 周韧哲, zhourz@smail.nju.edu.cn

2020 年 12 月 23 日

Problem 1

设计双月数据, 使其线性不可分, 用单层感知器和 BP 神经网络分别进行分类, 比较其结果。尝试分析 BP 网络隐藏神经元在非线形分类中所起的作用。

Solution.

如下图中的左图为产生的双月数据, 一共 2000 个点; 中间图为 BP 神经网络的分类结果; 右图为单层感知机的分类结果。我设计了一个拥有 3 层隐藏层的 BP 网络, 其中隐藏层激活函数为 *relu*, 隐藏层神经元个数为 32, 输出层激活函数为 *sigmoid*, 学习率设置为 0.0003, 迭代 200 轮后分类正确率为 100%。单层感知机的激活函数为 *sigmoid* 函数, 迭代 200 轮后分类正确率为 91.25%。命令行输入 `python double_moon.py --hidden_layer_num 3` 和 `python double_moon.py --hidden_layer_num 0` 运行可复现结果。

可以看出, 单层感知机无法处理线性不可分数据, 而多层 BP 网络可以很好地分类线性不可分数据。隐藏神经元或者说隐藏层的权重矩阵, 能够把数据特征抽象到另一个维度空间, 从而对输入特征进行了多层次的抽象: 矩阵和向量相乘, 本质上就是对向量的坐标空间进行一个变换。因此, 隐藏层的权重矩阵的作用就是使得数据的原始坐标空间从线性不可分, 转换成了线性可分。

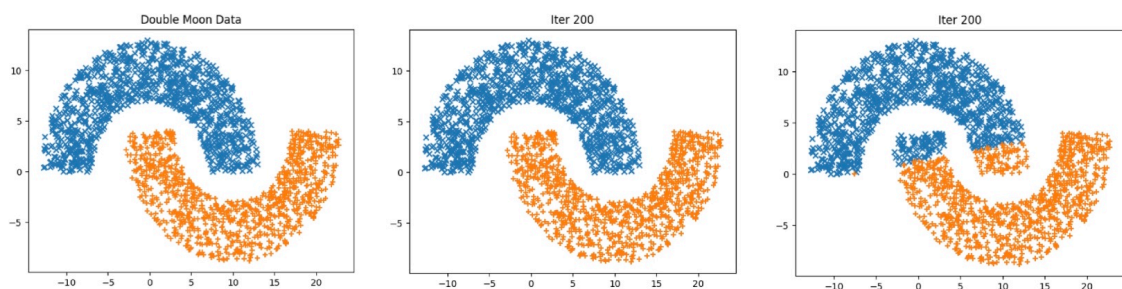


图 1: P1

Problem 2

从网上下载 `optdigits` 数据库, 这是一个 10 类手写数字库。建立一个 BP 网络用于这个 10 类手写数字的分类, 使用一层隐藏层。用 `training` 集对该网络进行训练, 用 `test` 集测试。

(1) 比较不同的隐藏层神经元个数对结果有什么影响。

(2) 思考如何进一步提高识别率。

Solution.

(1) 命令行输入 `python optdigits.py --hidden_dim 128` 运行可以训练网络并在测试集上测试。使用一层隐藏层, 固定迭代 100 轮, 当隐藏神经元个数为 16, 32, 64, 128, 256 时, 在测试集上的准确率分别为 0.9137, 0.9399, 0.9572, 0.9594, 0.9599。直观来看, 隐藏层神经元个数越多, 分类准确率越高, 因为神经元个数越多, 表示能力越强。

(2) 通过增加隐藏层数量、提高训练轮数、调节学习率, 可以进一步提高识别率。当迭代 120 轮, 隐藏层神经元个数为 128, 隐藏层数量为 2, 学习率为 0.002 时, 在测试集上的准确率达到了 0.9627。通过调参得到较好的学习率、迭代次数、不同激活函数, 或者通过归一化输入数据也可以提高网络模型的效果。

Problem3

训练一个 1-5-1 的神经网络来逼近函数 $\sin(x)$:

(1) 由 \sin 函数产生 100 个输入-输出对, 训练该神经网络使其能根据 x 预测 $\sin(x)$, 报告其精度 (误差)。

(2) 报告输入-隐藏层权值及隐藏-输出层权值。

(3) 报告作为 x 的函数的隐藏神经元输出 y , 报告输出神经元的输出 z , 找到每一个隐藏神经元输出函数的分界点, 讨论参与产生输出的隐藏神经元。

Solution.

(1) 命令行输入 `python sinx.py` 运行可得到结果, 最终迭代 500 轮, 均方误差约为 0.04216。

(2) 输入-隐藏层权值为:

$$w_0 = [0.91403577, -0.63624996, -0.02249133, 1.08801627, 0.18224804]$$

$$b_0 = [0.89582971, -0.80988285, -0.09036633, -1.35959951, 0.17857199]$$

隐藏-输出层权值为:

$$w_1 = [1.20082916, 0.89272928, -0.1035579, -1.65504323, 0.23501796]$$

$$b_1 = [-1.12173529]$$

(3) 隐藏神经元的激活函数为 `relu`, 输出 y 见文件 `data/layer0_output.txt`。输出神经元的激活函数为 `tanh`, 输出 z 见文件 `data/layer1_output.txt`。观察输出易知参与产生输出的包括了所有隐藏神经元。