

Homework1 报告

日期: 2023 年 04 月 09 日

姓名: 陈天悦 学号: 22210980031

1 任务分析

1.1 数据集分析

MNIST 手写数字数据集是一个经典的多分类问题, 通过神经网络可以很好的实现手写字体的识别。MNIST 数据集来自 National Institute of Standards and Technology, 训练集由来自 250 个不同人手写的数字构成, 其中 50% 是高中学生, 50% 来自人口普查局的工作人员, 测试集也是同样比例的手写数字数据, 但保证了测试集和训练集的作者集不相交。

MNIST 数据集一共有 7 万张图片, 其中 6 万张是训练集, 1 万张是测试集。每张图片是 28×28 的 0-9 的手写数字图片组成。每个图片是黑底白字的形式, 黑底用 0 表示, 白字用 0-1 之间的浮点数表示, 越接近 1, 颜色越白。

1.2 神经网络分类任务

神经网络通常被用于数据的特征提取, 并通过多层神经元的连接, 最后进行数据的分类。本次任务中, 由于数据量小, 使用 MLP (多层感知机), 通过梯度回传训练, 即可进行对于 MNIST 数据集的分类。

2 网络设计

使用 numpy 构建一个两层神经网络分类器并进行训练, 我们需要计算每层的梯度。在此任务中, 采用激活函数 ReLU 进行网络构建, 并通过交叉熵损失函数进行损失计算和梯度回传。

2.1 激活函数

神经网络通常通过矩阵乘法进行线性计算, 通过激活函数引入非线性, 本次采用 ReLU 作为激活函数:

$$\text{ReLU}(z) = \begin{cases} z & z > 0 \\ 0 & z \leq 0 \end{cases} \quad (1)$$

2.2 梯度计算

本次使用的两层全连接层是一种线性结构, 可以通过公式表示为:

$$f(X) = WX + b \quad (2)$$

其中 W 表示神经元权重, X 表述输入数据, $f(X)$ 表示输出值。

梯度求解可以表示为:

$$\nabla f(w_1, w_2, w_3) = (\partial f / (\partial w_1), \partial f / (\partial w_2), \partial f / (\partial w_3)) \quad (3)$$

2.3 交叉熵损失函数

本任务是经典的多分类问题，我们采用多分类的交叉熵损失函数进行计算：

$$-\sum_{c=1}^M y_{o,c} \log(p_{o,c}) \quad (4)$$

3 参数查找

3.1 参数搜索空间

在这个项目中，我们在超参数方面的探索主要是搜索隐藏层的神经元个数，学习率和 l2 正则化参数。具体搜索空间如下表：

表 1: 参数搜索空间

	Search space
Hidden	64, 128, 256, 512 1024
Learning rate	0.0001, 0.0005 0.001, 0.005, 0.01, 0.05, 0.1
L2 regular	0.001, 0.005, 0.01, 0.05, 0.1

3.2 搜索结果

通过实验遍历了以上搜索空间的所有组合，最后确定在隐藏层神经元个数为 512，学习率为 0.005，L2 正则化参数为 0.005 时训练效果最优。

4 实验结果

在最优超参数组合下进行训练，得到的训练过程的中间结果和最后的参数可视化如下：

4.1 损失函数曲线

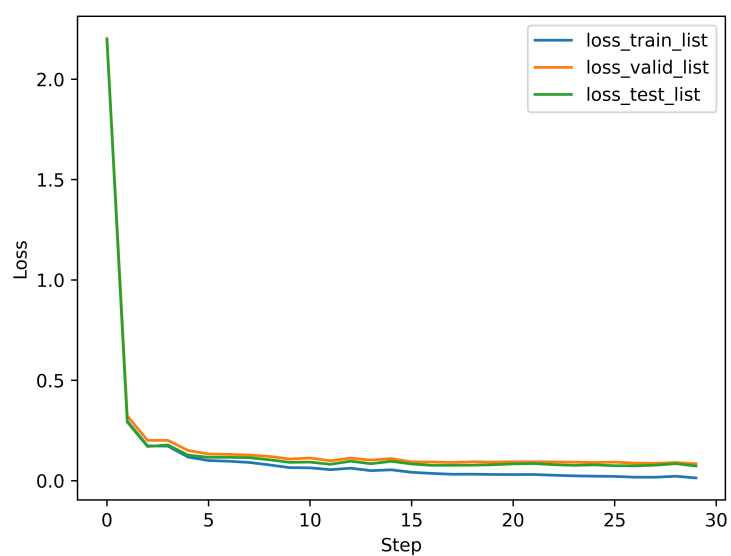


图 1: 训练损失曲线

通过损失函数曲线可以发现，训练在 5 轮之后逐渐收敛，模型在训练集、验证集和测试集上的表现也都趋于相同，模型的性能也比较稳定。

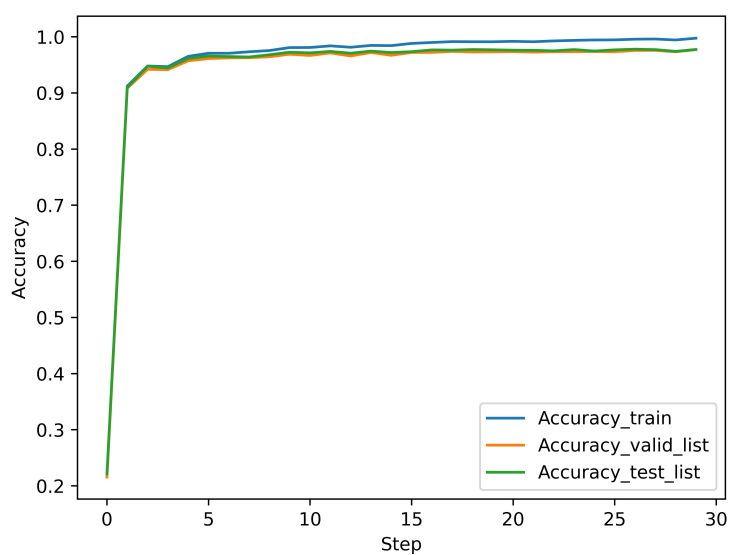


图 2: 训练准确率曲线

准确率和损失函数呈现出相同的变化趋势，都是在 5 轮之后逐渐收敛，同时，在训练集、验证集、测试集上的表现也比较统一。

4.2 参数可视化

以下给出了两层神经网络每层神经元的偏差和权重的直方图展示。

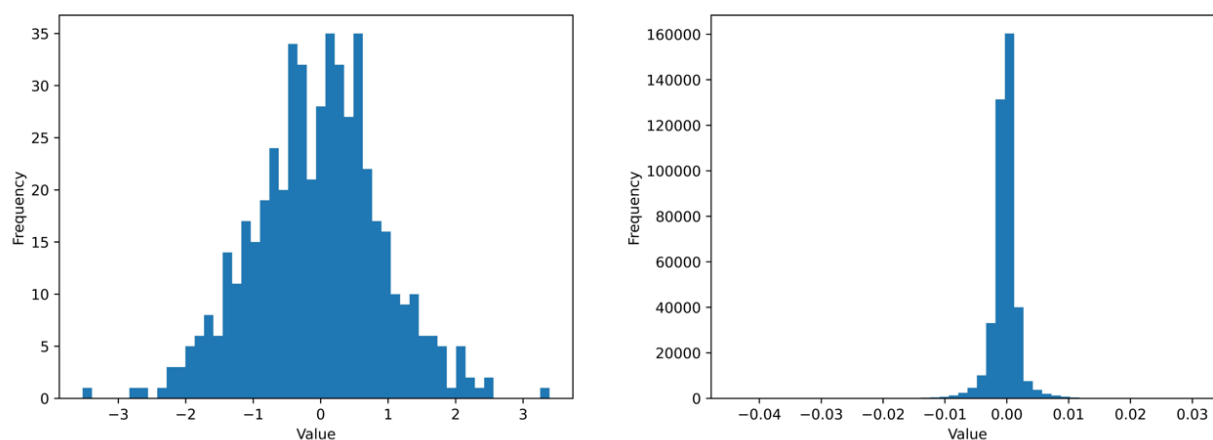


图 3: 第一层神经网络的偏差和权重直方图

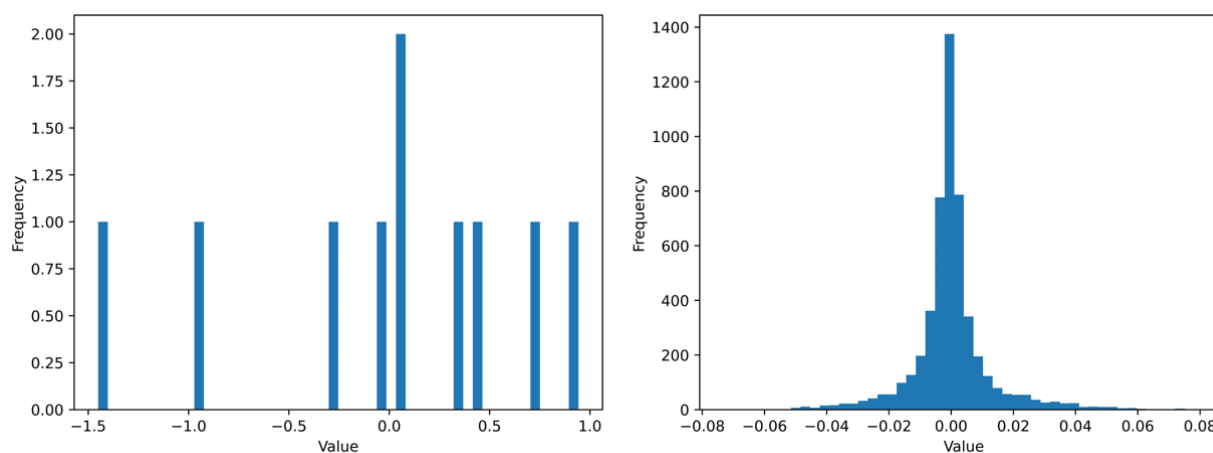


图 4: 第二层神经网络的偏差和权重直方图

从图中我们可以看出：不同网络层的权重分布比较相似，而偏差分布相异。

5 步骤说明

训练：运行 train.py

测试：运行 test.py

可视化：运行 draw.py

Github 链接：<https://github.com/tubao3104/HW1>

百度网盘地址：链接：https://pan.baidu.com/s/1aXATR4zNMzq1aXyeLM_oDQ 提取码：71pj