

《神经网络与深度学习》作业一

项目名称:	ReLU 神经网络拟合
姓 名:	朱励钊
学 号:	2053066
班 级:	21 级数据科学与大数据技术
课 程:	神经网络与深度学习
指导老师:	程大伟
完成日期:	2024.03.13

1 函数定义

定义目标函数为：

$$f(x) = 2x^2 + 3x + 5$$

该实验利用二层 ReLU 神经网络模型对该目标函数进行拟合。

2 数据采集

取目标函数最具明显特征的区间，即导数符号变换，区间为 $[-10, 10]$ 。

训练集在该区间随机取出 `num_points`（默认为 100，可以增减）个数据点，并代入目标函数得到标签值。考虑实际应用中的数据通常包含一些噪音或随机性，故在训练集标签值中加入了服从正态分布的随机噪音。

测试集在该区间以 0.2 的步幅取出 100 个数据点，并代入目标函数得到标签值。

故数据集为 $(\text{num_points}, 1)$ 的训练集， $(100, 1)$ 的测试集。

3 模型描述

3.1 Numpy 实现

定义一个两层的 ReLU 神经网络，图 3-1 如所示，隐藏层神经元个数为 64，输出层神经元个数为 1，用 ReLU 激活函数连接。

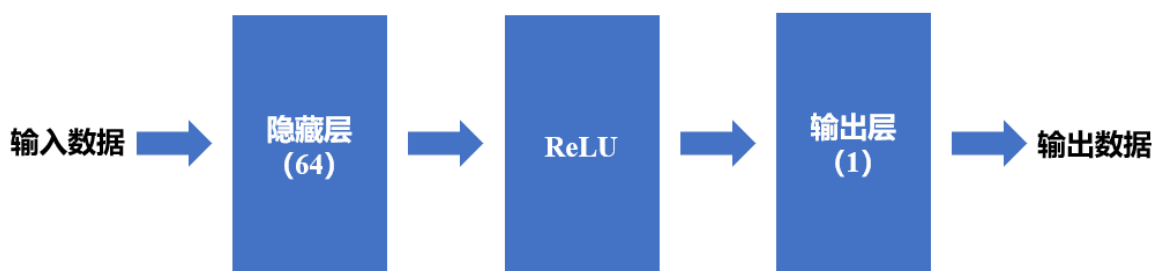


图 3-1 模型结构

利用 `numpy` 手动实现前向传播与后向传播。其中，设置损失函数为 `MSELoss`，优化算法取小批量梯度下降算法。此处的小批量是由于所有数据传入模型后的损失值过大，造成 `inf` 值，使得梯度回传出现 `nan` 值。而小批量可以通过调整 `batch_size` 对该问题进行改善。

3.2 Pytorch 实现

同样定义一个两层的 ReLU 神经网络，为便于对比，其结构与 3.1 的模型完全一致。

利用 `torch` 自动实现梯度回传。其中，设置损失函数为 `MSELoss`，优化算法改为更优秀的 `Adam` 算法。此处使用 `Adam` 优化器是因为其更新的步长能被限制在一定范围内且可自动调整学习率，这使得所有数据传入模型后几乎不会出现很大的损失值。

4 拟合效果

4.1 Numpy 实现

设置 `epochs=5000`、`learning rate=0.00001`、`batch size=10`，训练后的模型对测试集进行预测，其预测结果如图 4-1 所示。可以发现，拟合结果不尽人意，MSE 较大，但经过学生大量实践：不改变优化算法的情况下，对该目标函数进行拟合，十分容易出现损失值过大从而出现 `inf` 值的情况，即使将 `batch size` 调整为 1、增加训练轮次或调整学习率。所以该拟合结果整体还是不错的。

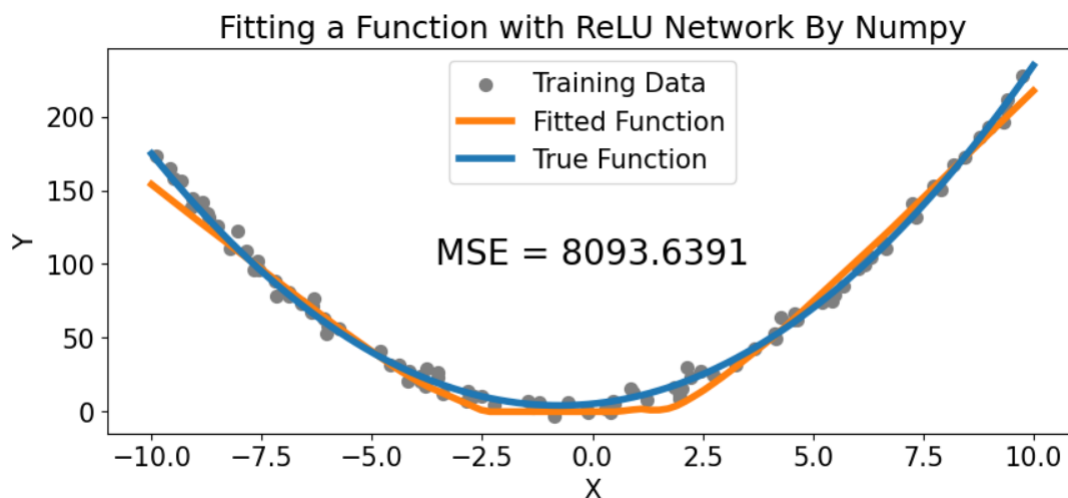


图 4-1 numpy 拟合结果图

4.2 Pytorch 实现

设置 `epochs=1000`、`learning rate=0.1`，训练后的的模型对测试集进行预测，其预测结果如图 4-2 所示。拟合结果十分完美，MSE 仅仅个位数。与 4.1 进行对比，在 `epochs` 更少、学习率更高的情况下，拟合出更佳的结果，这足以说明 Adam 优化器在此问题下的适合与优越。

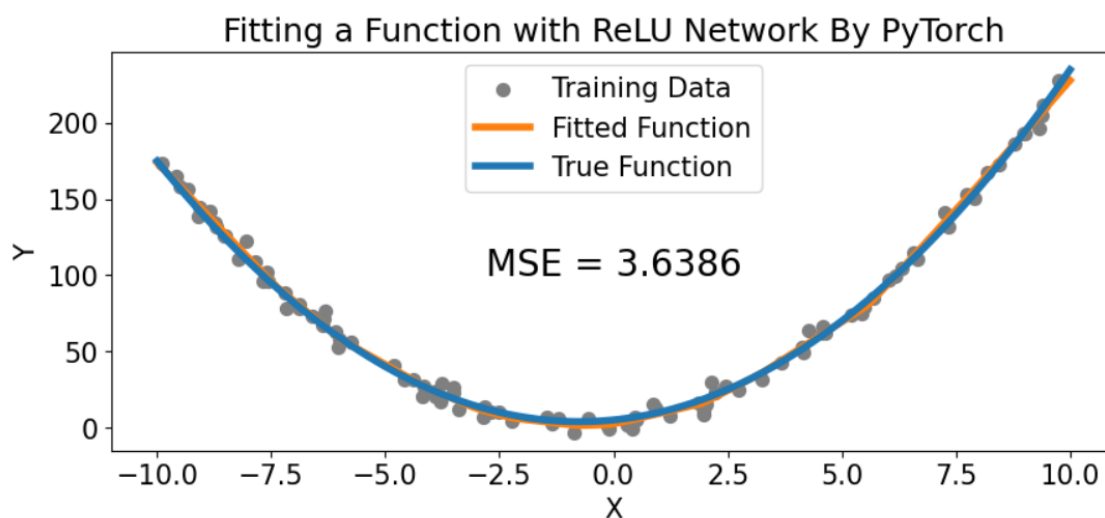


图 4-2 torch 拟合结果图