

使用基于ReLU的神经网络拟合多项式函数

1. 函数定义

我们选择了一个简单的多项式函数作为我们的目标函数：

$$f(x) = 2x^3 - 3x^2 + 2x - 1$$

这个函数具有明显的非线性特征，可以用于验证一个两层的ReLU网络可以模拟任何函数。

模型使用pytorch实现。

2. 数据采集

为了进行模型训练和测试，我们生成了一个数据集。我们从一个均匀分布中采样了1000个数据点作为训练集，并从一个等间隔的网格中取了100个数据点作为测试集。对于每个数据点 (x_i, y_i) ，其中 x_i 是在区间 $[-2, 2]$ 中均匀采样的， y_i 是通过目标函数 $f(x)$ 计算得到的，并且添加了一些高斯噪声以模拟真实数据。

3. 模型描述

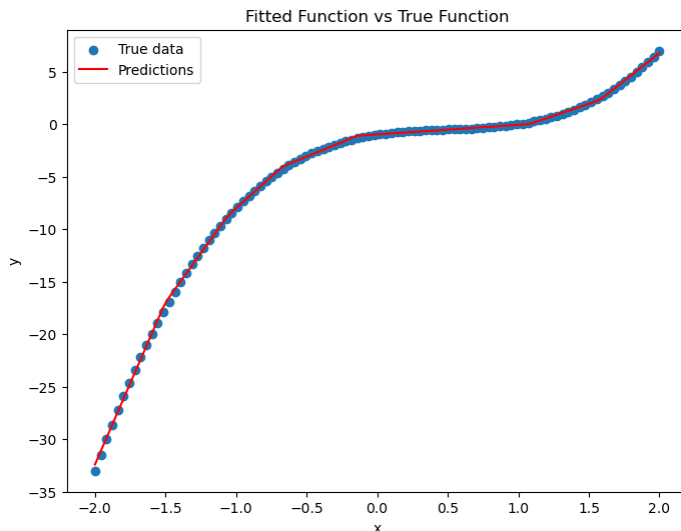
我们构建了一个简单的两层全连接神经网络。该网络有一个输入层、一个隐藏层和一个输出层。每个隐藏层节点使用ReLU激活函数，输出层没有使用激活函数。模型的架构如下：

- 输入层：1个神经元
- 隐藏层：10个神经元，ReLU激活函数
- 输出层：1个神经元，线性输出

我们使用均方误差作为损失函数，Adam优化器作为优化算法来训练模型。模型的目标是最小化预测值与真实值之间的均方误差。

4. 拟合效果

经过5000轮的训练后，我们对模型进行了测试，并将模型的预测结果与真实函数进行了比较。从结果来看，模型能够很好地拟合目标函数。在测试集上，模型的预测结果与真实函数的曲线非常接近，表明模型具有较好的泛化能力。



通过这个实验，我们验证了一个简单的两层ReLU神经网络可以很好地拟合复杂的非线性函数，这符合理论和实践中的结论。

[附] 代码: [function-fit-pytorch.ipynb](#)