

基于两层 ReLU 网络的函数拟合

徐志铭 2251804

一、问题说明

理论和实验证明，两层 ReLU 网络可以近似任意函数。本文使用 ReLU 作为激活函数，利用含有两个隐藏层的网络，拟合 $[-2\pi, 2\pi]$ 上的 $\sin x$ 函数。首先在区间上均匀取 300 个点，90% 作为训练集，其余作为测试集。

二、网络搭建

采用含有两个隐藏层的 ReLU 网络，输入数据维度 $x \in \mathbb{R}^{b \times 1}$ ，输出数据维度 $\hat{y} \in \mathbb{R}^{b \times 1}$ 。前向传播过程如下：

$$\begin{aligned} z^{[1]} &= xW_1 + b_1, W_1 \in \mathbb{R}^{1 \times h_1}, b_1 \in \mathbb{R}^{1 \times h_1} \\ a^{[1]} &= \text{ReLU}(z^{[1]}) \\ z^{[2]} &= a^{[1]}W_2 + b_2, W_2 \in \mathbb{R}^{h_1 \times h_2}, b_2 \in \mathbb{R}^{1 \times h_2} \\ a^{[2]} &= \text{ReLU}(z^{[2]}) \\ \hat{y} &= a^{[2]}W_3 + b_3, W_3 \in \mathbb{R}^{h_2 \times 1}, b_3 \in \mathbb{R}^{1 \times 1} \end{aligned} \tag{2.1}$$

其中 h_i 为第 i 层神经元的数量，网络参数为 $\{W_i, b_i\}_{i=1}^3$ 。

三、模型训练

3.1 计算图实现

采用numpy实现静态计算图。主要思路为：

1. 设计两种节点：数据节点DataNode和函数节点FunctionNode，都继承父类节点Node。函数节点实现正向和反向传播功能。
2. 设计静态计算图，在计算损失函数值之后，利用深度优先搜索遍历计算图，之后从叶子节点开始计算梯度，并调用函数节点的反向传播方法进行梯度回传。

3.2 拟合效果

展示拟合效果在图2.1中。最终模型在测试集上的拟合均方误差为 0.001。

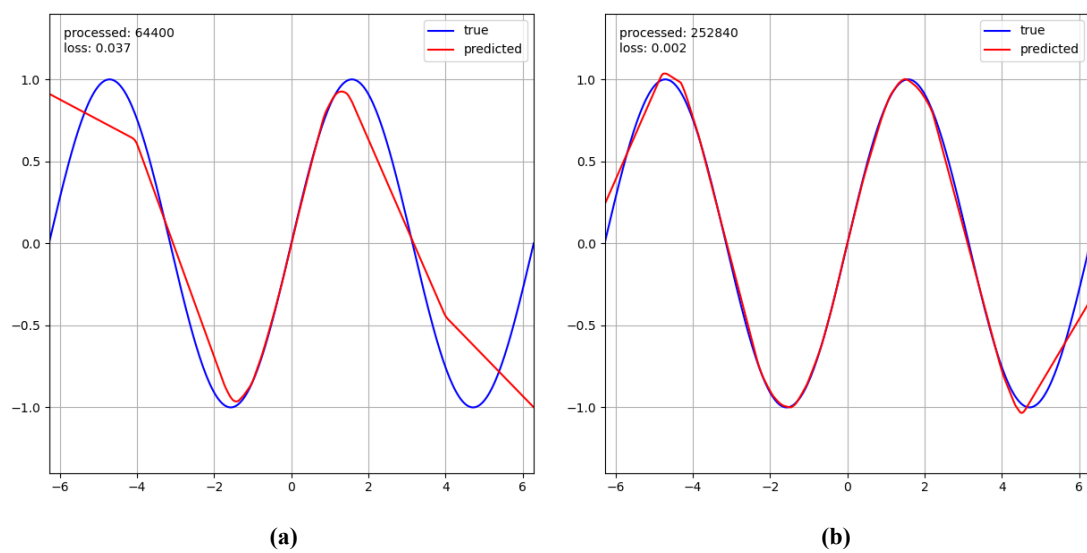


图 2.1 网络训练效果。(a) 训练初始阶段; (b) 训练结束阶段。在测试集上的拟合误差以及用于训练的样本数展示在图片左上角。红线代表模型输出，蓝线为拟合的函数真值。