

同濟大學

TONGJI UNIVERSITY

实验报告

成员

王欣玥 (2253300)

学院 (系)

国豪书院

专 业

信息安全

任课教师

程大伟

日 期

2025 年 3 月 22 日

1、函数定义

目标函数定义为一个非线性函数，包含正弦波动、二次项、线性项。

```
def target_function(x):
    return np.sin(2 * np.pi * x) + 0.5 * x**2+x
```

2、数据采集

训练集：

x_{train} ：在 $[-1,1]$ 之间均匀采样 1000 个数据点。

y_{train} ：计算 x_{train} 通过目标函数 $target_function(x_{train})$ 得到的真实值。

测试集：

x_{test} ：在 $[-1,1]$ 之间均匀采样 100 个数据点。

y_{test} ：计算 $target_function(x_{test})$ 。

并将 NumPy 数组转换为 PyTorch tensor

```
np.random.seed(42)
x_train = np.linspace(-1, 1, 1000).reshape(-1, 1)
y_train = target_function(x_train)

x_test = np.linspace(-1, 1, 100).reshape(-1, 1)
y_test = target_function(x_test)
torch.manual_seed(42)
x_train_tensor = torch.tensor(x_train, dtype=torch.float32)
y_train_tensor = torch.tensor(y_train, dtype=torch.float32)
x_test_tensor = torch.tensor(x_test, dtype=torch.float32)
y_test_tensor = torch.tensor(y_test, dtype=torch.float32)
```

3、模型描述

网络结构：

输入层：单个输入 x

隐藏层 1：100 维，全连接，使用 ReLU 激活。

隐藏层 2：50 维，全连接，使用 ReLU 激活。

输出层：1 维，全连接，直接输出预测值。

```
class ReLUNetwork(nn.Module):
    def __init__(self):
        super(ReLUNetwork, self).__init__()
        self.layer1 = nn.Linear(1, 100)
```

```
self.layer2 = nn.Linear(100, 50)
self.output_layer = nn.Linear(50, 1)
self.relu = nn.ReLU()

def forward(self, x):
    x = self.relu(self.layer1(x))
    x = self.relu(self.layer2(x))
    x = self.output_layer(x)
    return x
```

训练设置：

损失函数(MSELoss)：衡量预测值和真实值的误差（均方误差）。

优化器(Adam)：设置初始学习率 0.005。

训练轮数：10000

4、拟合效果

拟合结果如下，如图所示，模型成功拟合了目标函数：

Test MSE: 0.000024

