

# 函数拟合实验报告

2251079 隋建政

## 1.函数定义

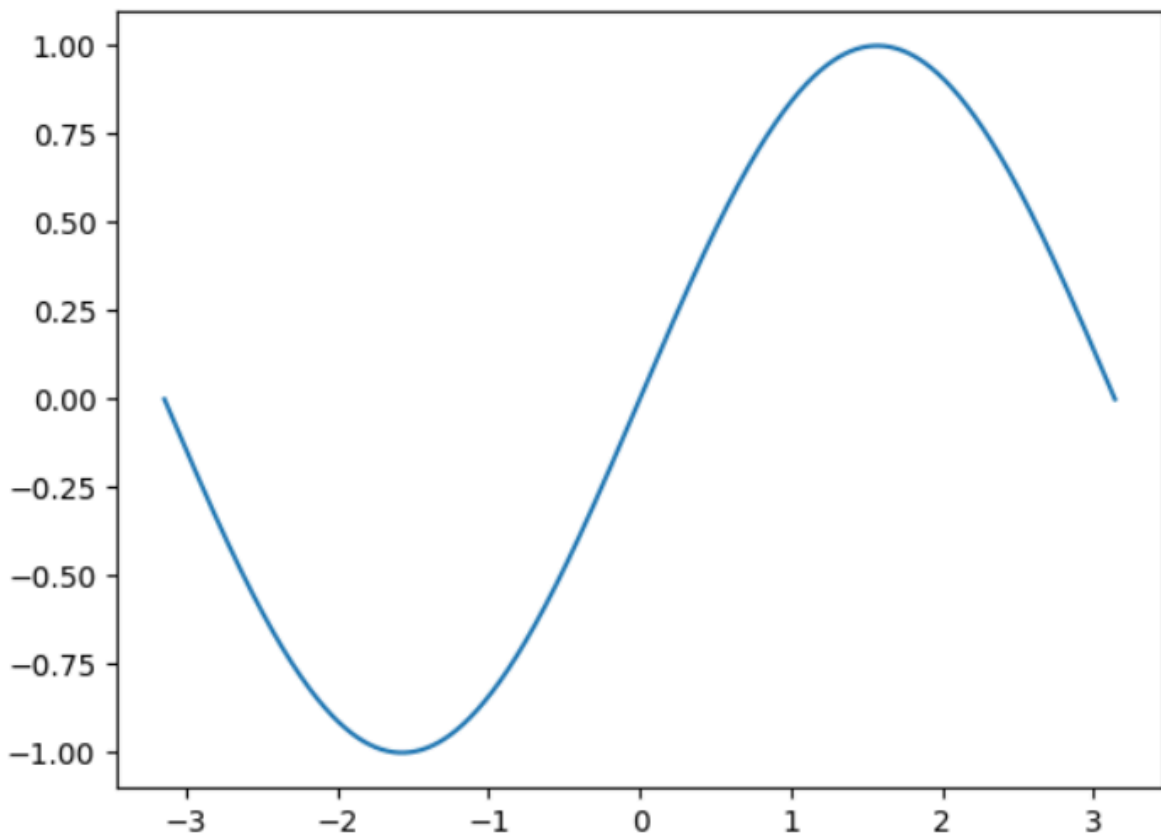
在本次实验中选取正弦函数进行拟合：

$$y = \sin(x)$$

给出代码如下：

```
x = torch.linspace(-math.pi, math.pi, 1000).unsqueeze(1)
y = torch.sin(x)

plt.plot(x, y)
```



## 2.数据采集

使用Pytorch框架中的TensorDataset和DataLoader采集数据，代码如下：

```
dataset = TensorDataset(x, y)
dataloader = DataLoader(dataset, batch_size=32, shuffle=True)
```

### 3.模型描述

在本次实验中，使用两层的ReLU神经网络对正弦函数进行拟合。神经网络的输入为1，输出也为1，具有一个隐藏层，拟合网络定义如下：

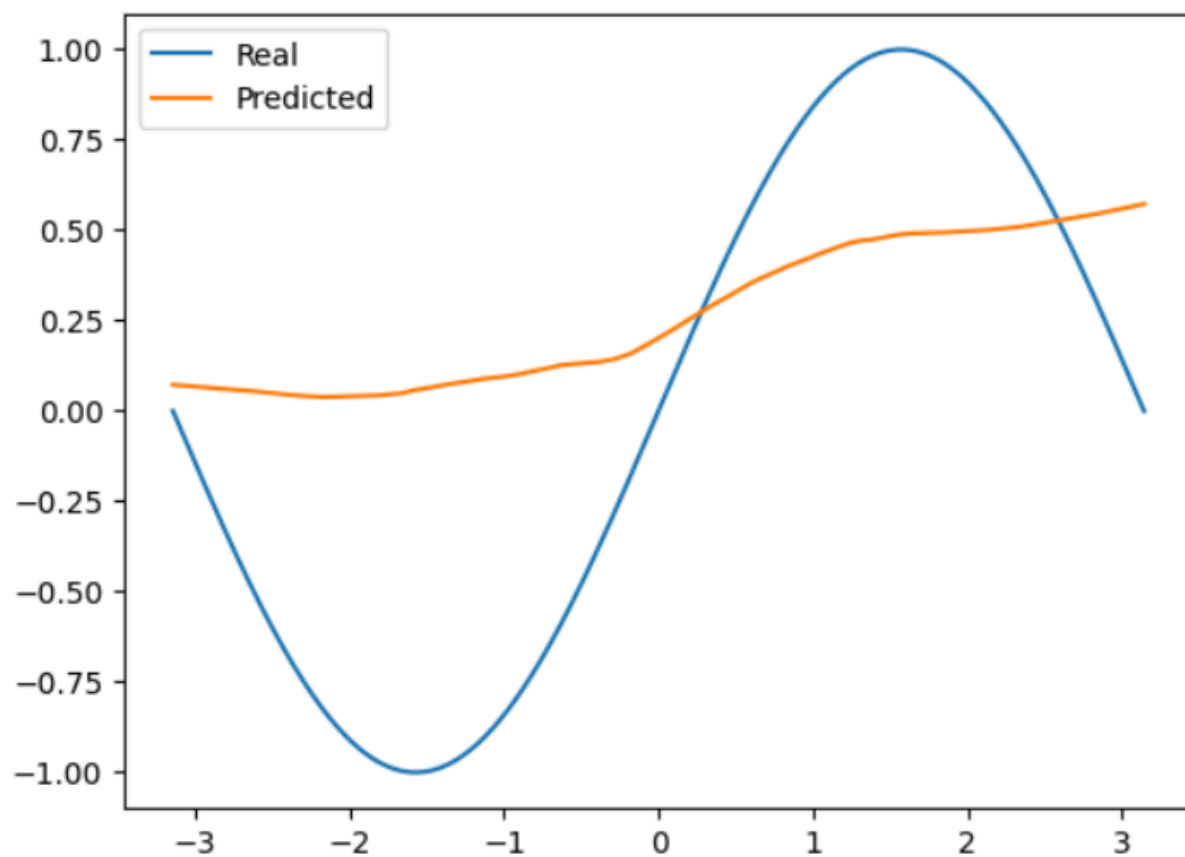
```
class myModel(nn.Module):  
    def __init__(self, hidden_size):  
        super().__init__()  
        self.net = nn.Sequential(  
            nn.Linear(1, hidden_size),  
            nn.ReLU(),  
            nn.Linear(hidden_size, 1)  
        )  
  
    def forward(self, x):  
        return self.net(x)
```

网络的隐藏单元数设置为256，学习率设置为1e-3，epoch的数量设置为2000。网络优化器选用Adam优化器，损失函数使用MSE。

```
hidden_size = 256  
learning_rate = 1e-3  
num_epochs = 2000  
  
model = myModel(hidden_size)  
optimizer = optim.Adam(model.parameters(), lr = learning_rate)  
loss_func = nn.MSELoss()
```

### 4.拟合效果

训练前网络结果：



训练后网络拟合的效果：

