使用Relu神经网络拟合函数

①确定要拟合的函数

在本次作业中，我尝试去拟合基本数学函数f=sin(x)

②在函数上采样生成训练集和测试集

为了训练和测试模型，用numpy库生成数据，其中训练集包含500个样本，测试集包括100个样本

```
# 训练数据 (x_train, y_train)
x_train = np.linspace(-10, 10, 500).reshape(-1, 1)  # 训练集：从 -10 到 10 生成 500 个数据点
y_train = func(x_train)  # 目标函数值

# 测试数据 (x_test, y_test)
x_test = np.linspace(-10, 10, 100).reshape(-1, 1)  # 测试集：从 -10 到 10 生成 100 个数据点
y_test = func(x_test)

# 转换为 PyTorch 张量
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构建两层的神经网络，使用Relu为激活函数，其中第一层有64个神经元，第二层输出1个值。

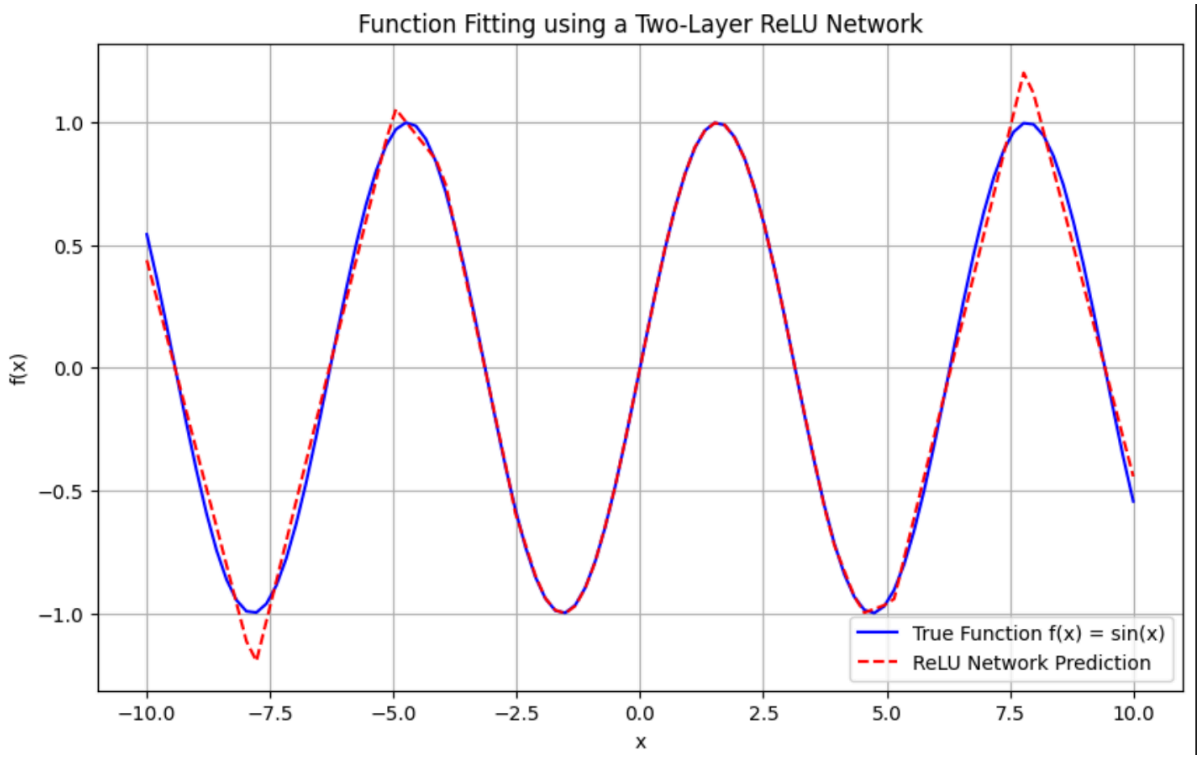
```
class ReLU_Network(nn.Module):
    def __init__(self):
        super(ReLU_Network, self).__init__()
        # 输入层到隐藏层
        self.layer1 = nn.Linear(1, 64)  # 1维输入，64个隐藏神经元
        # 隐藏层到输出层
        self.layer2 = nn.Linear(64, 1)  # 64个隐藏神经元到1维输出

    def forward(self, x):
        x = torch.relu(self.layer1(x))  # 使用 ReLU 激活函数
        x = self.layer2(x)  # 输出层
        return x
```

④使用 Adam 优化器和 均方误差 (MSE) 损失函数来训练网络。每经过 10 个 epoch 输出一次当前的损失，一共10000个epoch。

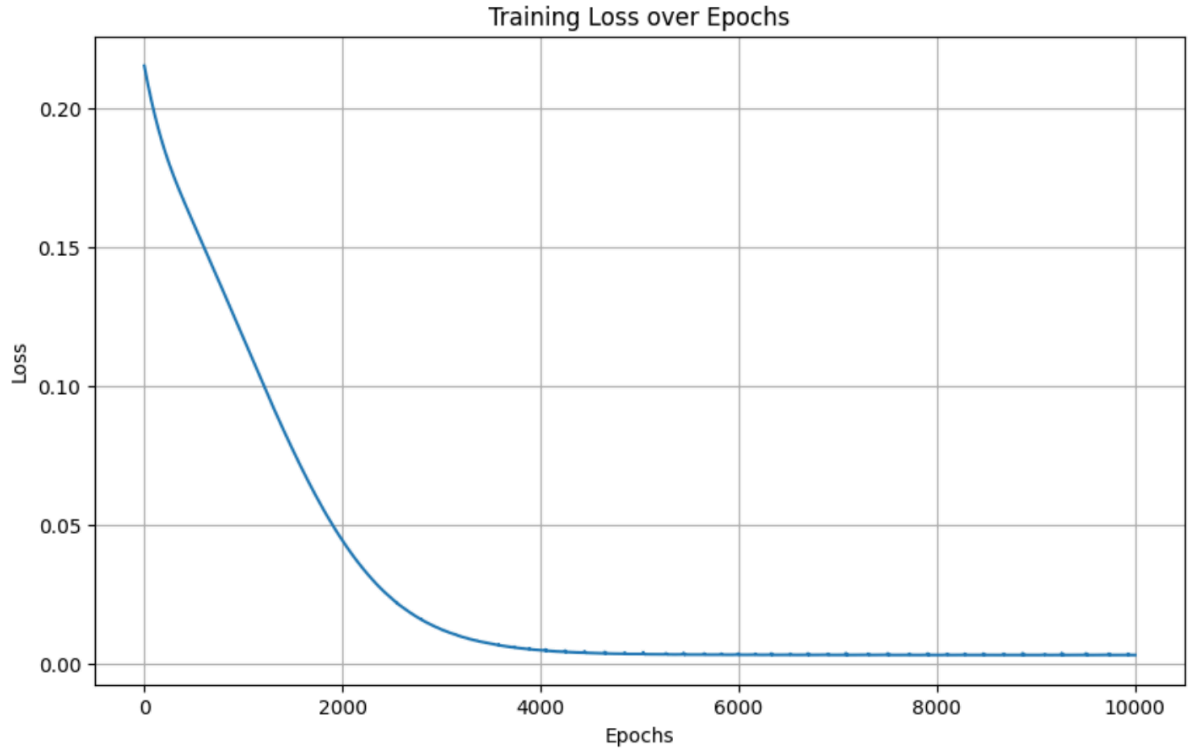
损失的变化情况如下：

```
Epoch [10/10000], Loss: 0.2135
Epoch [20/10000], Loss: 0.2117
Epoch [30/10000], Loss: 0.2099
Epoch [40/10000], Loss: 0.2081
Epoch [50/10000], Loss: 0.2064
Epoch [60/10000], Loss: 0.2047
Epoch [70/10000], Loss: 0.2031
Epoch [80/10000], Loss: 0.2015
Epoch [90/10000], Loss: 0.2000
Epoch [100/10000], Loss: 0.1985
Epoch [110/10000], Loss: 0.1970
Epoch [120/10000], Loss: 0.1956
Epoch [130/10000], Loss: 0.1942
Epoch [140/10000], Loss: 0.1929
Epoch [150/10000], Loss: 0.1916
Epoch [160/10000], Loss: 0.1903
Epoch [170/10000], Loss: 0.1891
Epoch [180/10000], Loss: 0.1879
Epoch [190/10000], Loss: 0.1867
Epoch [200/10000], Loss: 0.1855
Epoch [210/10000], Loss: 0.1844
Epoch [220/10000], Loss: 0.1833
Epoch [230/10000], Loss: 0.1822
Epoch [240/10000], Loss: 0.1812
Epoch [250/10000], Loss: 0.1801
...
Epoch [9970/10000], Loss: 0.0030
Epoch [9980/10000], Loss: 0.0030
Epoch [9990/10000], Loss: 0.0030
Epoch [10000/10000], Loss: 0.0030
```

⑤使用测试集评估模型，并绘制拟合效果



⑥展示模型训练曲线



可以看到模型在5000轮左右收敛，拟合效果良好