

函数拟合实验报告

1. 函数定义

这里选择函数： $y = \sin(2\pi x)$ 来进行拟合。

2. 数据采集

定义函数 $f(x)$ ：

```
def f(x):  
    return np.sin(2 * np.pi * x)
```

Python

在 $[0, 1]$ 上进行采样，得到训练数据与测试数据，然后进行线性变化扩大到 $[-1, 1]$ ，使模型能够学习到 \sin 函数从 $-\pi$ 到 π 的快速变化部分。

```
N_train = 100  
  
# 训练数据  
x_train = np.linspace(0, 1, 100).reshape(-1, 1)  
y_train = f(x_train)  
  
# 测试数据  
x_test = np.linspace(0, 1, 100).reshape(-1, 1)  
y_test = f(x_test)  
  
# 线性变化  
x_train = (x_train - 0.5) * 2  
x_test = (x_test - 0.5) * 2  
  
print(x_train.shape)  
print(x_test.shape)
```

Python

(100, 1)
(100, 1)

3. 模型描述

这里使用的是一个简单的双层 ReLU 神经网络，采用 `numpy` 实现。损失函数选择均方误差损失函数。

```
# 前向传播
z1 = np.dot(x_train, W1) + b1
a1 = relu.forward(z1)

# 输出层计算:  $z2 = a1 * W2 + b2$ 
z2 = np.dot(a1, W2) + b2
y_pred = z2
```

4. 拟合效果

学习率为 0.1，迭代次数为 7000，在测试集上的拟合结果如下图所示。

