

# 函数拟合

## 函数定义

$$f(x) = \sin(1.1\pi x) + 4.5 \cos(1.4\pi x)$$

## 数据采集

使用随机函数生成1000个x以及对应的y作为训练集，100个x以及对应的y作为测试集。

```
def generate_data(n_samples=100):  
    x = np.random.rand(n_samples)  
    y=test_function(x)  
    return x, y  
  
x,y=generate_data(1000)  
x_test,y_test=generate_data(100)
```

## 模型描述

使用Numpy实现相关函数以及函数拟合模型。具体地，搭建了一个两层的ReLU网络来进行拟合。其中，输入层大小为1，隐藏层大小为64，输出层大小为1，学习率设置为0.01。

```
def relu(x):  
    return np.maximum(0,x)  
  
#relu的导数  
def relu_derivative(x):  
    return np.where(x>0,1,0)  
  
class FunctionFitModel:  
    def __init__(self,input_dim,hidden_dim,output_dim,learning_rate=0.01):  
        self.learning_rate=learning_rate  
        self.w1=np.random.randn(input_dim,hidden_dim)*0.1  
        self.b1=np.zeros((1,hidden_dim))  
        self.w2=np.random.randn(hidden_dim,output_dim)*0.1  
        self.b2=np.zeros((1,output_dim))  
  
    def forward(self,x):  
        #...  
  
    def get_loss(self,y_pred,y_true):  
        return np.mean(np.square(y_pred-y_true))#均方误差  
  
    def backward(self,x,y_true,y_pred):  
        #...  
  
    def train(self,x,y_true,epochs,loss_threshold=-1.):  
        #...  
  
model=FunctionFitModel(1,64,1,0.01)
```

## 拟合效果

在自行定义的函数上，使用随机生成的1000组数据为模型进行了100000轮训练，虽然未达到预设的loss阈值0.005，但拟合效果已经相当不错，均方误差只有0.006~0.007。通过可视化展示可见预测的函数曲线与真实曲线已经相当拟合。因此，我认为拟合效果是不错的。

...

Epoch 99900, Loss: 0.006054884836661394

Test Loss: 0.00633938657861444

