



# 《神经网络与深度学习》实验报告

## 一、实验内容

自行定义一个函数，并使用基于 **ReLU** 的神经网络来拟合此函数。

## 二、实验要求

- 请自行在函数上采样生成训练集和测试集，使用训练集来训练神经网络，使用测试集来验证拟合效果。

- 可以使用深度学习框架来编写模型，如 **tensorflow**、**pytorch**、**keras** 等。

## 三、实验过程

选择拟合正弦函数  $y = \sin x$ ，使用 **pytorch** 实现。

### 1、函数定义

定义一个类 **Network** 作为神经网络模型。该类继承了 **nn.Model**，是一个具有两个隐藏层的前馈神经网络模型。

其中定义了两个函数。**\_\_init\_\_** 函数用于初始化，输入四个参数 **n\_in**, **n\_hidden1**, **n\_hidden2**, **n\_out** 分别表示输入层、两个隐藏层、输出层的神经元数量，在函数中建立各层的联系。**forward** 函数用于实现神经网络的前向传播过程，参数 **x** 为训练集数据。输入 **x** 经过第一个线性层 **self.layer1** 进行线性变换，然后通过 **ReLU** 激活函数进行非线性变换得到 **x\_1**。接着，**x\_1** 经过第二个线性层 **self.layer2** 进行线性变换，再次通过 **ReLU** 激活函数得到 **x\_2**。最后，**x\_2** 经过第三个线性层



`self.layer3` 进行线性变换后返回输出结果。

## 2、数据采集

使用 Numpy 的 `linspace` 在  $[0, 2\pi]$  之间取固定步长的 1000 个数据作为训练集 `x`，再对 `x` 做正弦运算得到 `y`。对 `x` 和 `y` 进行格式转换便于后续的计算操作。

使用 Numpy 的 `random.random` 在  $[0, 2\pi]$  之间取随机分布的 50 个数据组成测试集 `x_test`，再对 `x_test` 做正弦运算得到 `y_test` 用于检验拟合结果。

## 3、模型描述

模型结构：四层的神经网络模型，输入层和输出层各有一个节点，两个隐藏层各有 20 个节点；

激活函数：ReLU 函数；

训练集：0 到  $2\pi$  的 1000 个数据和它们的正弦值作为样本；

损失函数：均方误差损失函数 `MSELoss`

优化：Adam 优化器

训练过程：通过循环迭代对模型进行 5000 次训练，不断计算梯度更新参数使损失函数尽可能小，每迭代 1000 次就输出当前损失值并将训练结果可视化，输出预测结果和目标结果的对比图。

结果测试：在 0 到  $2\pi$  的区间内随机生成 50 个测试数据，与训练结果对比。



## 4、拟合效果

经过 5000 次训练后  $\text{loss} = 1.4334581464936491\text{e-}05$ ，预测曲线与目标曲线对比图、测试集与预测结果对比图如下所示，可见拟合效果较好。

