

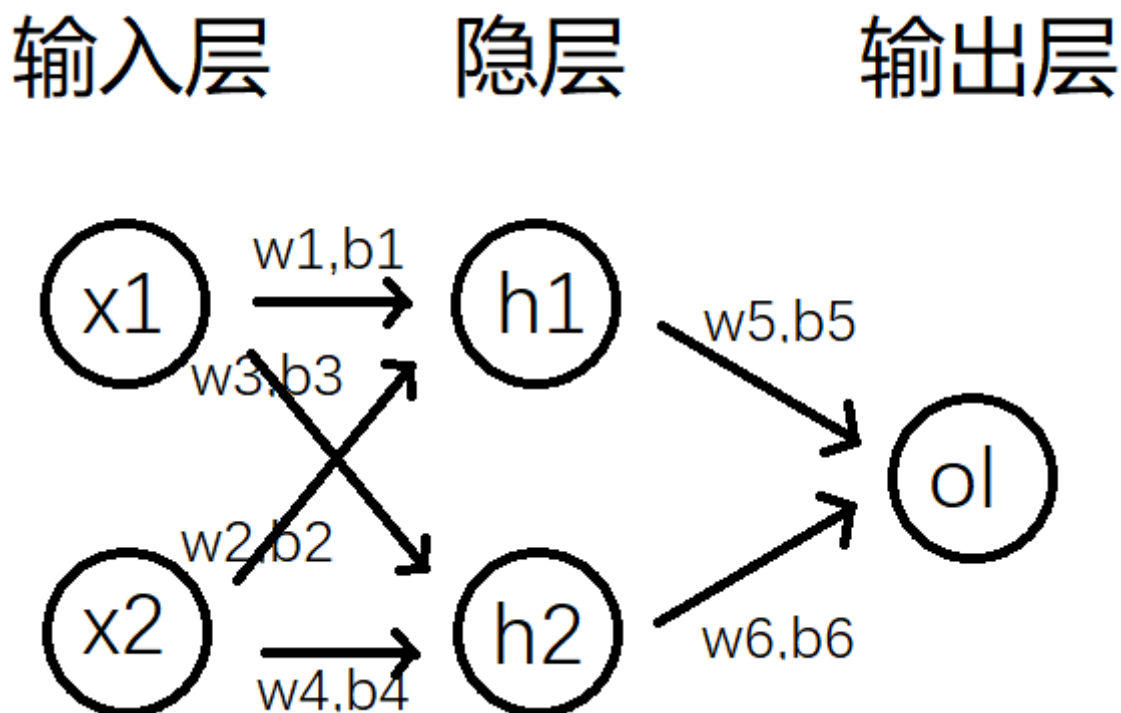
# 编程题指南——神经网络

## 题目

神经网络的训练和预测

## 问题描述

考虑如train\_feature.csv及train\_target.csv中的数据集，我们希望用如下图所示的简单神经网络对本问题进行分类。



$x_1, x_2, h_1, h_2, o_1$ 分别表示输入、隐层节点和输出节点，其中 $h_1, h_2, o_1$ 的激活函数都是sigmoid函数。

## 相关说明

你需要完成main.py中的函数sigmoid\deriv\_sigmoid\mse\_loss\神经网络NeuralNetwork\_221中的train函数。各函数的输入输出相关说明如下。

```
def sigmoid(x):           #输入为实值，输出为输入对应的sigmoid值。

def deriv_sigmoid(x):     #输入为实值，输出为sigmoid函数在x点的梯度

def mse_loss(y1,y2):      #输入为预测结果向量与真实标记向量，输出为这一组预测结果的均方误差
    (MSE)

def train(x,y):           #输入为训练样本的特征及其标记，本函数不输出而是更新神经网络的参数
    (w,b)
```

## 作业流程说明

1. 完成函数编写。
2. 参考main.py中main主程序部分完成神经网络的训练。
3. 读取测试样本特征并完成预测，并将分类结果输出为**学号\_ypred.csv**并提交。

## 评分标准

(10/10) 完成sigmoid及其梯度函数。

(20/20) 完成mse\_loss的编写

(30/30) 完成神经网络的训练

(50/50) 对训练样本的预测结果足够好

注：为了使训练样本的预测足够好，你可选择的方法**包括但不限于**以下改进：

1. 编写更复杂的神经网络
2. 使用更巧妙设计的损失函数（如交叉熵、加入正则项等）
3. 使用更复杂的优化方法（如Adam, Adaboost）
4. 设计合适的validation方法
5. 运用early stop、模拟退火等技术防止过拟合
6. 运用Batch Normalization等技术保证收敛性质

若选用以上方法或其他未提及的方法进行改进，请不要调包，务必亲自完成实现。请在**main.py**中以定义函数的形式实现改进方法。

## 提交清单

1. main.py
2. 学号\_ypred.csv