# 实验六 前馈神经网络（2）

班级： 学号： 姓名：

# 实验任务：基于前馈神经网络的二分类任务

**注意事项：不要粘贴源代码。实验报告中只需描述算法和操作流程，分析实验结果。**

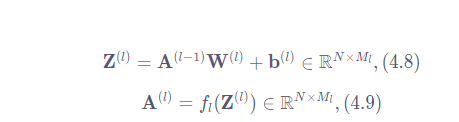
线性层算子-->logistic算子做激活函数--->实现一个两层的用于二分类任务的前馈神经网络，选用Logistic作为激活函数，可以利用上面实现的线性层和激活函数算子来组装--->测试现实例化一个两层的前馈网络，令其输入层维度为5，隐藏层维度为10，输出层维度为1。 并随机生成一条长度为5的数据输入两层神经网络------>定义二分类交叉熵损失函数----->误差反向传播算法来计算参数梯度【注意要在每个算子内添加backward函数】----->梯度下降更新参数放到另外的优化器中专门进行，此处的优化器需要遍历每层，对每层的参数分别做更新。

1数据集构建

使用第3.1.1节中构建的二分类数据集：Moon1000数据集，其中训练集640条、验证集160条、测试集200条。该数据集的数据是从两个带噪音的弯月形状数据分布中采样得到，每个样本包含2个特征。

2 模型构建

为了更高效的构建前馈神经网络，我们先定义每一层的算子，然后再通过算子组合构建整个前馈神经网络。



线性层直接调用函数（提一下下）

标出向量、矩阵

3 损失函数

二分类交叉熵损失函数见第三章。

4 模型优化

神经网络的层数通常比较深，其梯度计算和上一章中的线性分类模型的不同的点在于：

线性模型通常比较简单可以直接计算梯度，而神经网络相当于一个复合函数，需要利用链式法则进行反向传播来计算梯度。

（1）反向传播算法

第1步是前向计算，可以利用算子的forward()方法来实现；

第2步是反向计算梯度，可以利用算子的backward()方法来实现；

第3步中的计算参数梯度也放到backward()中实现，更新参数放到另外的优化器中专门进行。

（2）损失函数

二分类交叉熵损失函数；实现损失函数的backward()

（3）Logistic算子

为Logistic算子增加反向函数

（4）线性层

线性层输入的梯度；计算线性层参数的梯度；

（5）整个网络

实现完整的两层神经网络的前向和反向计算

（6）优化器

在计算好神经网络参数的梯度之后，我们将梯度下降法中参数的更新过程实现在优化器中。与第3章中实现的梯度下降优化器SimpleBatchGD不同的是，此处的优化器需要遍历每层，对每层的参数分别做更新。

5 完善Runner类：RunnerV2\_1

支持自定义算子的梯度计算，在训练过程中调用self.loss\_fn.backward()从损失函数开始反向计算梯度；

每层的模型保存和加载，将每一层的参数分别进行保存和加载。

6 模型训练

使用训练集和验证集进行模型训练，共训练2000个epoch。评价指标为accuracy。

7 性能评价

使用测试集对训练中的最优模型进行评价，观察模型的评价指标。

【思考题】

对比“基于Logistic回归的二分类任务”与“基于前馈神经网络的二分类任务”，谈谈自己的看法。

#### 实验结果：

XXX

# 实验总结：

XXX

REF：《神经网络与深度学习：案例与实践》