**深 圳 大 学 实 验 报 告**

**实验课程名称： 计算智能**

**实验项目名称： BP神经网络对蝴蝶花分类**

**学院： 计算机与软件 专业： 计算机科学与技术**

**报告人： 林士松 学号： 1800271013**

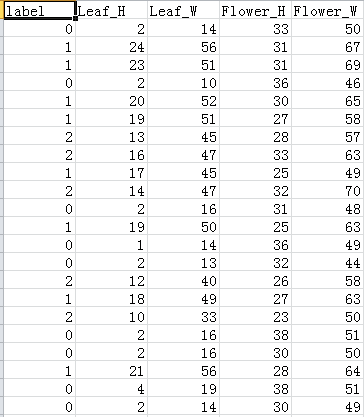
**授课老师： 朱安民**

1. **实验描述**

通过自己设计一个BP网络和调用Matlab的BP工具包对蝴蝶花进行分类。通过实验了解BP算法的基本原理。

1. **实验数据**

此实验用到的数据集总共有150个样本，总共有三个标签，每个样本包含四个特征，分别为叶长、叶宽、花长和花宽，具体格式如图（1）所示。我们将这150个数据集按照1:3的比例分为两部分，113个样本作为训练集，37个样本作为测试集。

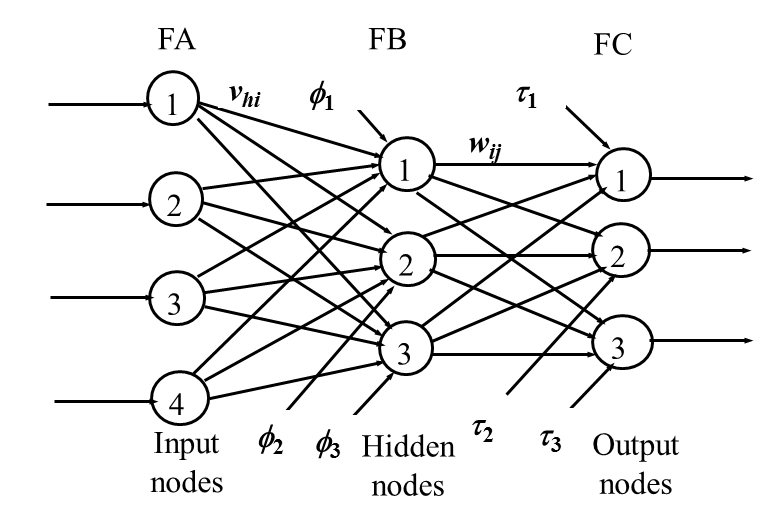


图（1）实验数据格式

1. **BP网络实现**

3.1、BP网络与设计

BP 算法是一种有监督式的学习算法，其主要思想是把学习过程分为两个阶段：第一阶段（正向传播过程），给出输入信息通过输入层经隐含层处理并计算每个单元的实际输出值；第二阶段（反向过程），若在输出层未能得到期望的输出值，则逐层递归地计算实际输出与期望输出之差值（即误差），以便根据此差值调节权值。 BP 神经网络的模型结构如图（2）所示。



图（2）BP神经网络模型

在本次实验中，每个样本有4个特征，于是我们定义输入层的节点数为4个；实验的学习难度不是很大，我们只采用了一层隐藏层，隐藏层的节点个数为3，最后总共有3个类别，因此输出层的节点个数为3.

3.2、算法步骤

BP算法的具体步骤如下：

第一步，网络的初始化：给层与层之间连接的权值和阈值分别赋予区间为（-1,1）内的随机数，设定误差函数为E，给定计算精度acc，最大学习次数Max\_epoch和最小误差TOR。

第二步，将数据集分为训练集和测试集，并对输入层的特征进行归一化处理。

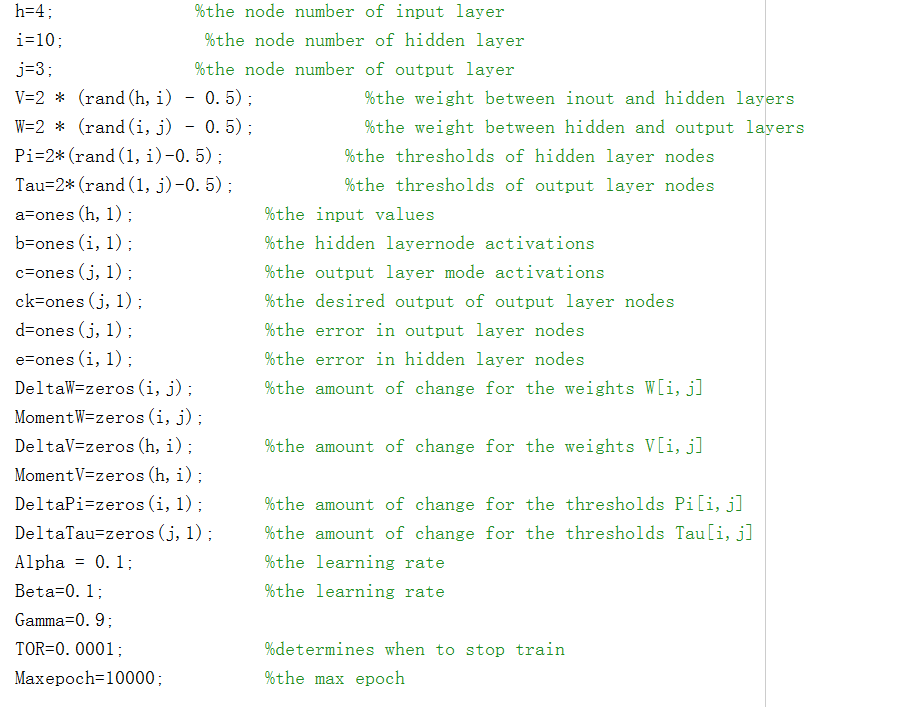
第三步，选取第k个样本作为输入，通过前向传播分别得到隐藏层神经元的数值和输出层节点的值。

第四步，利用期望输出和实际输出计算误差函数，计算误差对各层神经元的偏导数，并对权值做更新。

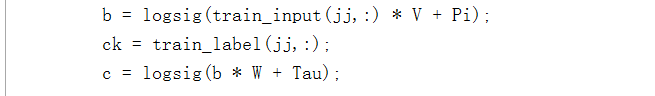
第五步，判断误差是否满足要求，当误差达到预设要求或者学习次数大于设定的最大次数时，结束算法。否则，选取下一个样本，返回第三步进行下一轮学习更新。

1. **自己设计BP神经网络：**

4.1、参数定义：



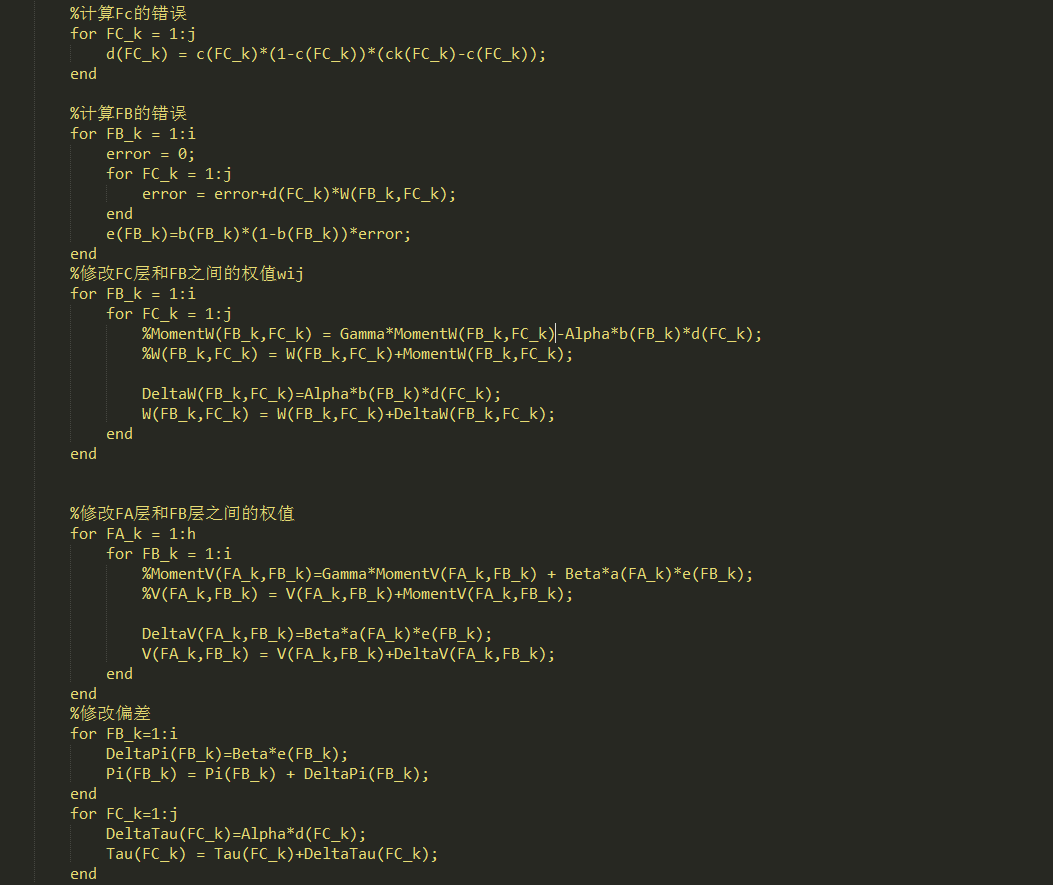
4.2正向传播



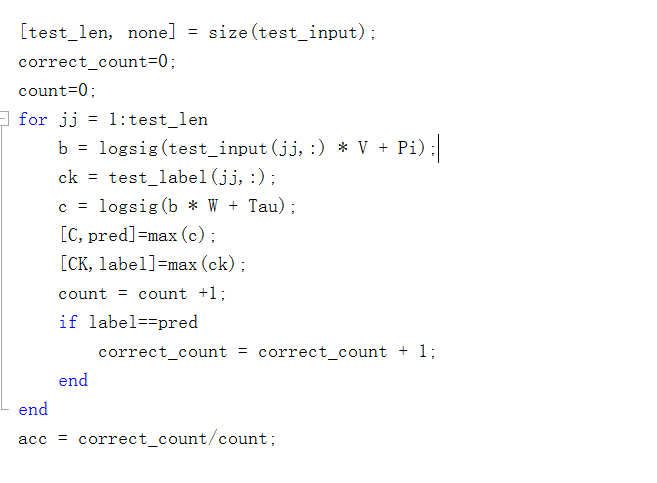
4.3、误差函数计算



4.4、反向传播



4.5、验证



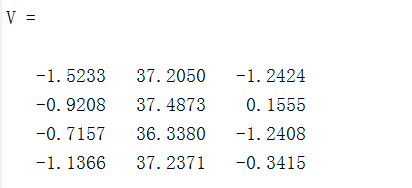
4.6、结果分析

下面经过修改隐藏层节点个数、修改学习率，修改最大迭代次数以及最小误差来分析对结果的影响，结果如下表所示。

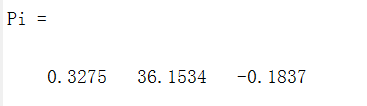
|  |  |  |  |
| --- | --- | --- | --- |
| Hidden nodes | Permitted error | Learning rate | Acc |
| 3 | 0.001 | 0.1 | 0.9189 |
| 6 | 0.001 | 0.1 | 0.8919 |
| 3 | 0.0005 | 0.1 | 0.8919 |
| 3 | 0.001 | 0.01 | 0.8919 |
| 3 | 0.0005 | 0.01 | 0.8648 |

表1 不同参数对结果的影响

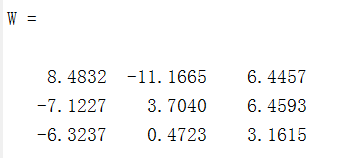
输入层到隐藏层的权值参数：



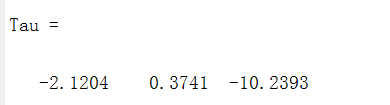
输入层到隐藏层的阈值参数：



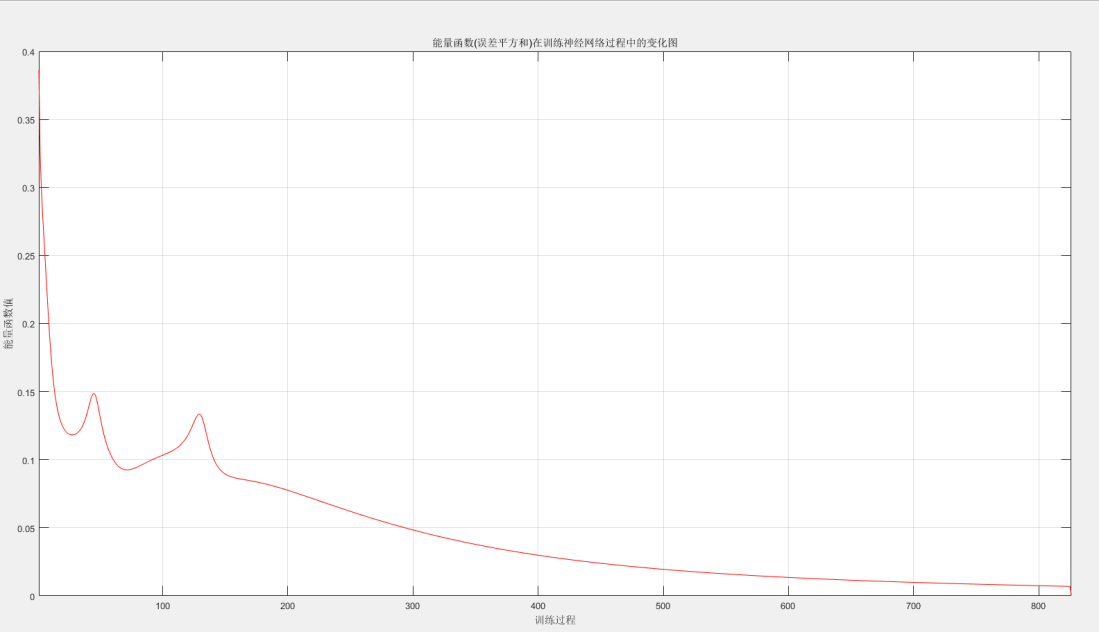
隐藏层到输出层的权值参数：



隐藏层到输出层的阈值参数：

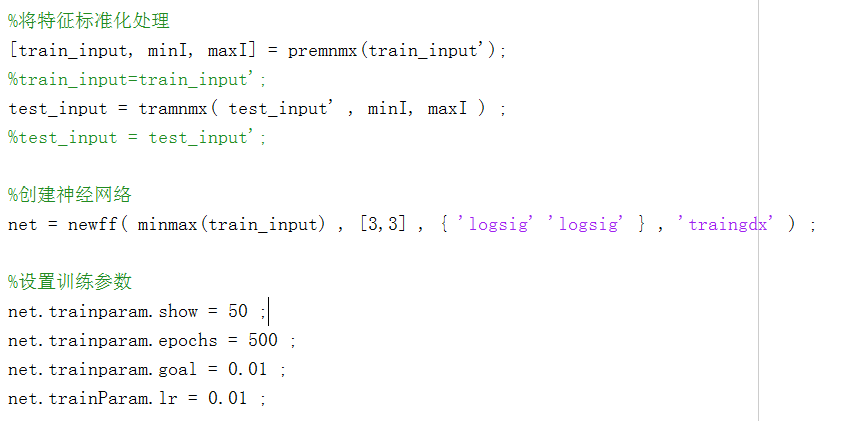


LOSS函数的变化：

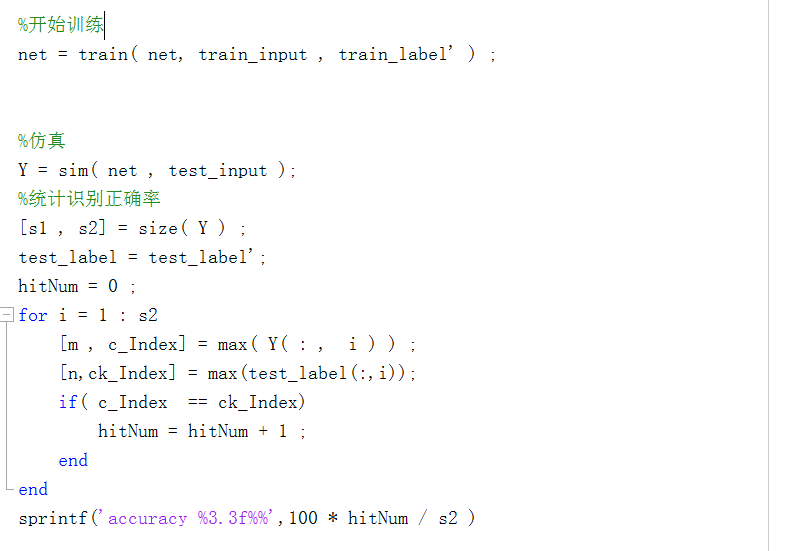


1. **调用BP或RBF工具包实现**

5.1、网络参数设定



5．2、网络设计：



5．3、实验结果

下面经过修改隐藏层节点个数、修改学习率、激活函数类别、训练学习方法来分析对结果的影响，结果如下表所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 训练学习方法 | 隐藏层节点 | 隐藏层激活函数 | 输出层激活函数 | 学习率 | 精度 |
| Traingdx | 3 | Sigmoid | Sigmoid | 0.01 | 0.94595 |
| Traingdx | 6 | Sigmoid | Sigmoid | 0.01 | 0.94595 |
| Traingdx | 3 | Sigmoid | Purelin | 0.01 | 1 |
| Traingdx | 3 | Purelin | Sigmoid | 0.01 | 0.59459 |
| Traingda | 3 | Sigmoid | Sigmoid | 0.01 | 0.94595 |
| Traingda | 3 | Sigmoid | Purelin | 0.01 | 0.97297 |
| Traingdx | 3 | Sigmoid | Sigmoid | 0.005 | 0.94595 |
| Traingdx | 3 | Sigmoid | Purelin | 0.005 | 0.89189 |

表（2）不同参数实验结果

从表（1）和表（2）可以看出，自己编写的bp算法相较于matlab调用的工具箱实现的bp算法在性能上会比较差。通过分析总结性能比较差的原因可能有下面几个方面：

自己编写的bp算法是通过随机梯度下降法反向传播的，收敛很慢。而调用的matlab工具箱实现了附加动量+自适应学习率的BP算法可以达到很好的性能；

在自己编写bp算法中学习率没能实现自适应，不可调，导致很难收敛；

1. **实验体会**

1）、BP 学习算法的基本思想

BP 算法是一种有监督式的学习算法，其主要思想是：输入学习样本，使用  
反向传播算法对网络的权值和偏差进行反复的调整训练，使输出的向量与期望向  
量尽可能地接近，当网络输出层的误差平方和小于指定的误差时训练完成，保存  
网络的权值和偏差。

2）、BP网络设计中结点个数如何确定?

第一、隐藏节点数不能太多，否则会加大计算量且可能出现过拟合现象；另外，隐藏层节点的数目也不能太少，否则会不够精确。经验可得隐藏层节点数课设计为Ni（输入层节点数）+No（输出层节点数）-2。

第二、输入层节点数为输入数据的特征数，比如该实验中的特征为花长、花宽、叶长、叶宽，因此输入节点数为4。

第三、输出层节点数为种类数，该实验中总共有三类，因此输出节点数为3.

3）、为什么要对数据进行通用标准化？

数据标准化主要是为了限定输入向量的最大值跟最小值不超过隐层跟输出层函数的限定范围，同时从而使数据具有可比性。

4）、为什么要对训练数据的输入顺序进行随机处理？

对训练数据的输入顺序进行随机化处理是为了保证能够有监督学习，同时使算法按照梯度下降法则进行学习。假如训练数据是有序的，那么会导致训练结果很难收敛到偏置值。

5）、为什么要将训练集合测试集分开？

将训练数据与测试数据分开主要是为了防止过拟合。如果把所有数据都用来训练模型，建立的模型自然是最契合这些数据的，测试表现也好。但是换了其它数据集测试这个模型效果可能就没有那么好的泛化性能。

6）、编程实现与调用工具包实现的不同与感受？

编程实现需要自己从头到尾编写网络，包括权值初始化、节点数设计、层数选择、正向传播过程和反向传播中的loss计算和权值更新等等；而调用工具包只需定义好一些超参数、节点数就可以了，其中的前向传播过程和反向传播梯度更新不需要我们自己编程实现，十分方便。另外，因为自己的编程能力不足，自己编程实现的算法的准确率往往回避调用工具包实现的要低。