教程 2 : BP-NN

提交日期：2020.7.18

提交人：詹紫琦

目录

[1. 题目 3](#_Toc27254)

[1.1 BP-NN(反向传播神经网络） 3](#_Toc31067)

[1.2 运行环境 3](#_Toc18800)

[2. 算法阐述或实验步骤说明 3](#_Toc2648)

[2.1算法阐述 3](#_Toc32220)

[2.3 实验步骤说明 5](#_Toc25484)

[3. 实验结果与截图 5](#_Toc2295)

[3.1 BP神经网络在鸢尾花数据上分类实验结果 5](#_Toc4019)

[4. 总结 8](#_Toc26426)

[4.1 总结 8](#_Toc27492)

[5. 参考文献 9](#_Toc5334)

# 题目

## BP-NN(反向传播神经网络）

BP算法[1]是一种神经网络更新权值参数的算法，假设这里有三层神经网络，其中包含一层输入层，一层隐含层，一层输出层，其中输入层与隐含层中都有一个偏置单元，通过设置每两层之间的权重参数来进行计算，相当于是多个感知机的计算，同时还需要一个激活函数，当输入层进入数据时，首先根据权重计算出到达隐含层的值，然后将该值进行激活函数激活，再根据权重计算出到达输出层的值，同样将该值进行激活函数激活，最终得到的输出就是这个三层神经网络模型得到的输出值，当然，这个输出值的取值关键就在于每两层之间的权重值，然后根据输出值与真实值的误差计算来进行反向传播更新每两层之间的权重值，依次迭代，这就达到了优化权重的效果，使误差更小，这就是反向传播算法的基本思想。

## 1.2 运行环境

系统：Ubuntu20.04，python3.7，Anaconda集成工具Jupyter编写。

# 算法阐述或实验步骤说明

## 2.1算法阐述

首先根据选定数据集，为UCI鸢尾花数据集，发现类别为3，输入特征数为4，确定神经网络结构，如下结构

其中输入层有5个单元（包含1个偏置单元），隐含层有6个单元（包含1个偏置单元），输出层有4个单元。

首先设置权重值W1，W1是输入层与隐含层之间的权重，权重值W2，W2是隐含层与输出层之间的权重，输入层偏置单元权值为B1，隐含层偏置单元权值为B2，然后进行前向传播。

对于任意一项数据，包含输入，得到的隐含层输入为，两者之间的关系如公式2-1

 （2-1）

隐含层输入经过激活函数sigmod函数激活后得到，激活函数为：如公式2-2

 （2-2）

同理，隐含层通过权重W2到达输出层，相关公式如2-3，公式2-4

 （2-3）

 （2-4）

至此前向传播完毕，由于权重W1，B1，W2，B2是随机初始化的权重值，需要进行调整，而通过初次的前向传播后，我们能得到输出层的误差值δ，如公式2-5：

 （2-5）

得到输出层的误差值之后，进行反向传播，依次得出隐含层的误差值，由于输出层不存在误差值，所以这里不计算输出层的误差。得到隐含层的误差值如公式2-6：

 （2-6）

得到各层的误差值之后，然后更新参数，这里使用的梯度下降更新，基本公式如2-7：

 （2-7）

由计算可知，损失函数相对于参数的偏导可以用公式2-8替代：

 （2-8）

然后进行更新权值。

## 2.3 实验步骤说明

首先读取数据集，根据数据集构建神经网络结构，随机初始化四个权值矩阵，设置学习率，学习率用来参数更新时作为梯度下降的学习率，然后进行前向传播，得到前向传播后的隐藏层激活后的数据以及各层的误差数据。将该数据用于反向传播中，然后编写反向传播函数，依次更新四个权值矩阵，计算出新的损失函数的值。然后用迭代完后的权值矩阵更新新的神经网络，用新的神经网络进行训练，输出结果，该结果是一个概率结果，取三个概率中最大的一类作为该分类的结果。然后将预测分类结果与实际结果进行对比，计算出分类正确率。

# 实验结果与截图

## 3.1 BP神经网络在鸢尾花数据上分类实验结果

首先基于控制变量法的原理，需要选取一个初始化权值矩阵，通过多次初始化，选取一个表现较好的权值矩阵，如图3-1：

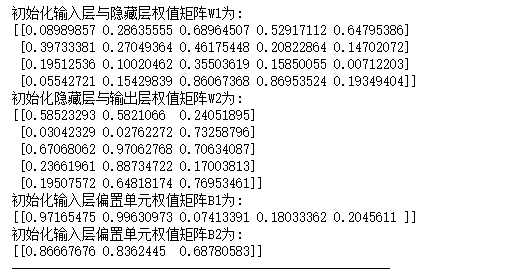


图3-1 初始化权重

将该权值矩阵作为之后实验中的初始化矩阵。在实验中由于存在梯度下降优化参数，存在两个可调整参数，一个是学习率alpha，另一个是迭代次数iteration。测试不同学习率下，不同迭代次数下，模型的分类效果如何。如表3-1。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| alpha  Iter | 100 | 500 | 1000 | 1500 | 2000 |
| 0.01 | 79.33% | 96.67% | 91.33% | 98.67% | 98.00% |
| 0.005 | 90.00% | 92.67% | 96.00% | 97.33% | 97.33% |
| 0.001 | 66.67% | 95.33% | 98.00% | 98.00% | 98.00% |

表3-1 学习率，迭代次数下的分类准确率

发现在alpha为0.01，迭代次数为1500时，能达到最高分类准确率，分类准确率为98.67%，输出的最终优化后的权重矩阵如图3-2.

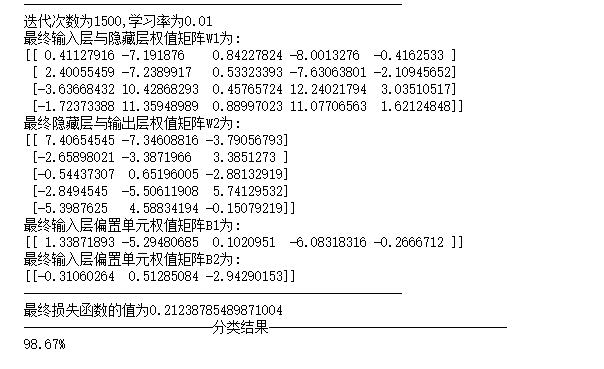


图3-2 设定alpha=0.01，iteration=1500时的结果输出

此时的损失函数随迭代次数变化图如图3-3

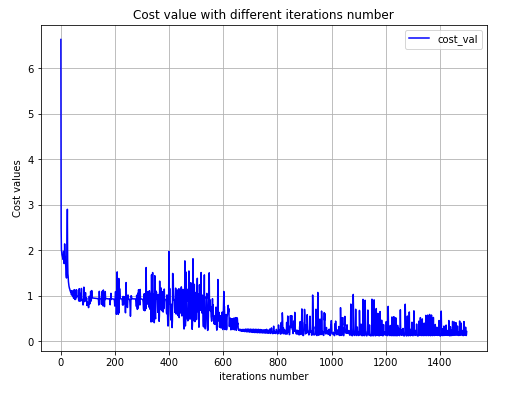


图3-3 alpha=0.01，iteration=1500时的损失函数

根据损失函数图可知，虽然有较好的分类效果，但损失函数存在着振荡的现象，需要将alpha参数降低，此时降低alpha为0.001，得到的损失函数图如图3-4，此时得到的分类准确率为98%，虽然准确率有所降低，但损失函数在迭代次数过多时不会产生振荡现象。损失函数图像较为平稳，模型的泛化能力能有所加强。

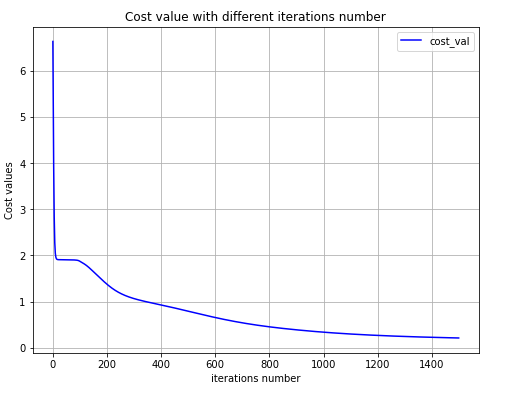


图3-4 alpha=0.001，iteration=1500时的损失函数变化

# 总结

## 4.1 总结

通过构建一个三层简单的神经网络，并且使用BP反向传播算法进行参数优化来实现鸢尾花分类器，最终能将其分类效果的准确率稳定在98%左右，同时也能得到较为平稳变化的损失函数图像。BP算法的使用充分的利用了神经网络的结构特性，通过对误差的反向传播，实现了参数的梯度下降操作来优化损失函数，最终能得到一个分类器，同时该BP网络的输出也是一个概率值，通过比较输出层三个单元的概率值大小来进行评判该输入的类别，同时这也是一个多分类问题。与逻辑回归中的单分类比较，如果逻辑回归需要进行多分类，则需要采用分而治之的方法设置多个二分类问题，而神经网络可以直接在输出层进行单元设置来一次实现多分类方法，大大简化了问题。

# 参考文献

[1] 周志华等.机器学习（西瓜书）