** **

**神经网络与反向传播算法**

院系名称 自动化科学与电气工程学院

专业名称 模式识别与智能系统

学生学号 15031184

学生姓名 李思奇

2018年5月27日

# 1. 简介

人工神经网络(ANN)的学习模式是一种受生物神经系统启发的方法，它在现实世界中执行异常复杂的计算，而不依赖于显式的定量操作。这项技术最初的灵感来自于生物大脑中由神经元及其突触形成的生物电网络。在神经网络模型中，一些简单的节点(称为“神经元”或“单元”)连接在一起形成一个节点网络，因此称为“神经网络”。

每个节点都有一组输入线，它们类似于生物神经元中的输入突触。每个节点也有一个“激活函数”，它告诉节点何时激活，类似于生物神经元。理想情况下，这个激活函数可以是生成一个“1”，或者是“0”，即是一个阶跃函数。“1”对应着神经元激活，而“0”对应着神经元抑制。然而，实际的激活函数并不如此简单。事实上，要创建有实用性的神经网络，它们几乎都要更加复杂，至少对于网络中的某些节点来说是如此。通常，神经网络至少有三层：输入层、隐藏层和输出层。输入层不进行处理，它只是将数据向量输入神经网络。数据经过输入层后输入隐藏层。隐藏层依次向输出层输入。神经网络中，实际的数据处理发生在隐藏层和输出层的节点上。

# 2. 基本原理

任何监督学习算法的目标是找到一个函数，它能过尽量好地将一组输入映射到它的正确输出。

在数学上，一个神经元的网络函数被定义为一些函数的组成，可以进一步定义为其他函数的组合。这可以方便地表示为一个网络结构，箭头描述变量之间的传递关系。一种广泛使用的组合类型是非线性加权和，其中：



其中K(通常称为激活函数)是一些预定义的函数，比如双曲正切。下面是一个简单的向量g = (g1, g2，…gn)的函数集合。反向传播需要一个已知的期望输出，以每个输入值的实际输出值来计算损失函数的梯度。因此，它通常被认为是一种有监督的学习方法。

定义平方误差函数为：



其中E为平方误差，t为训练样本的目标输出，y为输出神经元的实际输出。对于每个神经元j，它的输出被定义为。



神经元的输入网络是前一层神经元输出的加权总和。如果神经元是在输入层之后的第一层，那么输入层的就只是网络的输入。神经元的输入单元数为n，变量表示神经元i和j之间的权重。

激活函数是一般非线性和连续可微的。常用的激活函数是logistic函数，其形式如下：



logistic函数的导数有很好的性质，满足：



利用链式法则计算误差对权重的偏导数：



我们得到：



其中：



# 3. 利用BP算法训练神经网络

## 3.1 利用Iris dataset训练神经网络

### 3.1.1 模型及算法

Iris dataset是一个多分类数据集，包含训练样本150个，特征维度为4，输出类别数为3。构造一个3层神经网络，输入层神经元数为4，隐含层神经元数为5，输出层神经元数为3，神经网络结构简图如图3.1.1所示。

1

1

x1

x2

x3

y1

y2

y3

x4

Input layer

Hidden layer

Output layer

a1

a2

a3

a4

a5

**图3.1.1 神经网络结构简图**

设输入为：

 (3-1)

输出为：

 (3-2)

记样本数为m，为样本x通过神经网络前向传播得到的输出。为从第j层到第j+1层的权重矩阵，。

则有：





其中，为Sigmoid函数。

定义代价函数为：



其中，后一项相当于在logistic regression中的正则化项，目的是防止过拟合的现象出现。

训练神经网络的目标便是最小化损耗函数。我们使用误差反向传播算法(Back Propagation)来实现。

计算每层每个结点的误差：

 (3-3)

则令：



有：

 (3-4)

利用式(3-4)得到的梯度值，利用梯度下降算法，可以得到优化的权重矩阵。

### 3.1.2 权重矩阵初值设置

在神经网络算法实现的过程中，需要对权重矩阵进行初始化。若初始权重矩阵为0，则出现对称现象，神经网络性能极大的下降。为了打破这种对称，权重矩阵的初值采取生成随机数的形式。

利用Matlab，在[-0.12,0.12]的区间内随机生成初始权重矩阵：



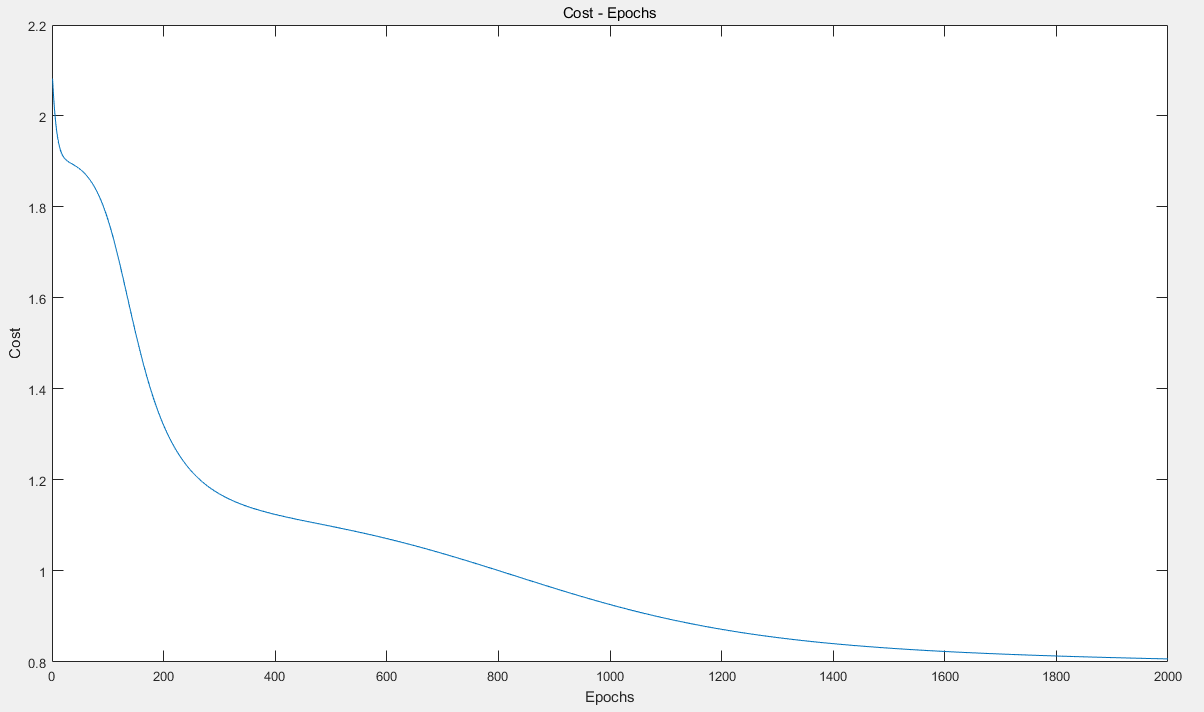


为了控制变量，在接下来的实验中，一直使用这组初始权重向量。

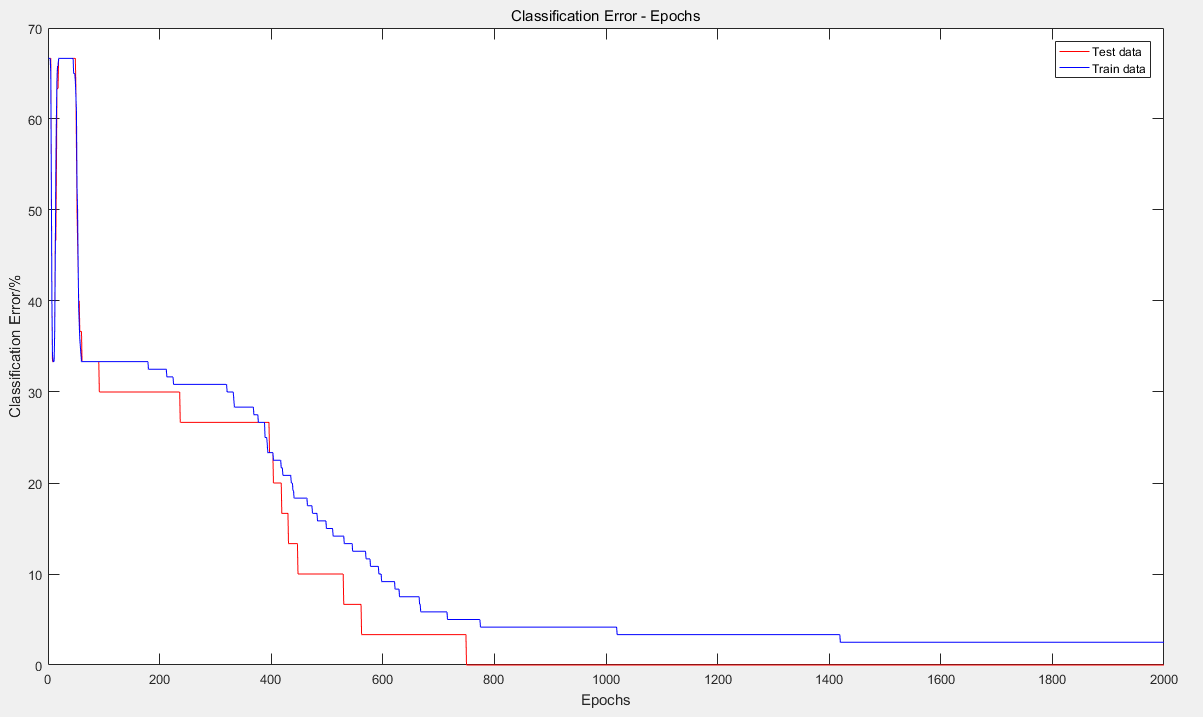
### 3.1.3 神经网络训练

对于Iris dataset，取每类的最后10个样本（总共30个样本）作为测试集，其余的120个样本作为训练集。

设置正则化系数、学习速率如下：。训练神经网络。做出代价函数与分类正确率关于epoch的曲线如图3.1.2和图3.1.3所示。



**图3.1.2 代价函数曲线**



**图3.1.3 错误率曲线**

经过2000个epoch的训练，得到的神经网络，对于训练集的分类正确率为97.5%，对于测试集的分类正确率为100%。可见训练出了一个分类效果较好的神经网络。

## 3.2 利用Wisconsin Cancer dataset训练神经网络

### 3.2.1 神经网络模型

Wisconsin Cancer dataset是一个多分类数据集，包含训练样本569个，特征维度为30，输出类别数为2。构造一个3层神经网络，输入层神经元数为30，隐含层神经元数为35，输出层神经元数为2，神经网络结构简图如图3.2.1所示。

1

1

x1

x2

y1

y2

x30

Input layer

Hidden layer

Output layer

a1

a2

a3

a35

……

……

**图3.2.1 神经网络结构简图**

设输入为：



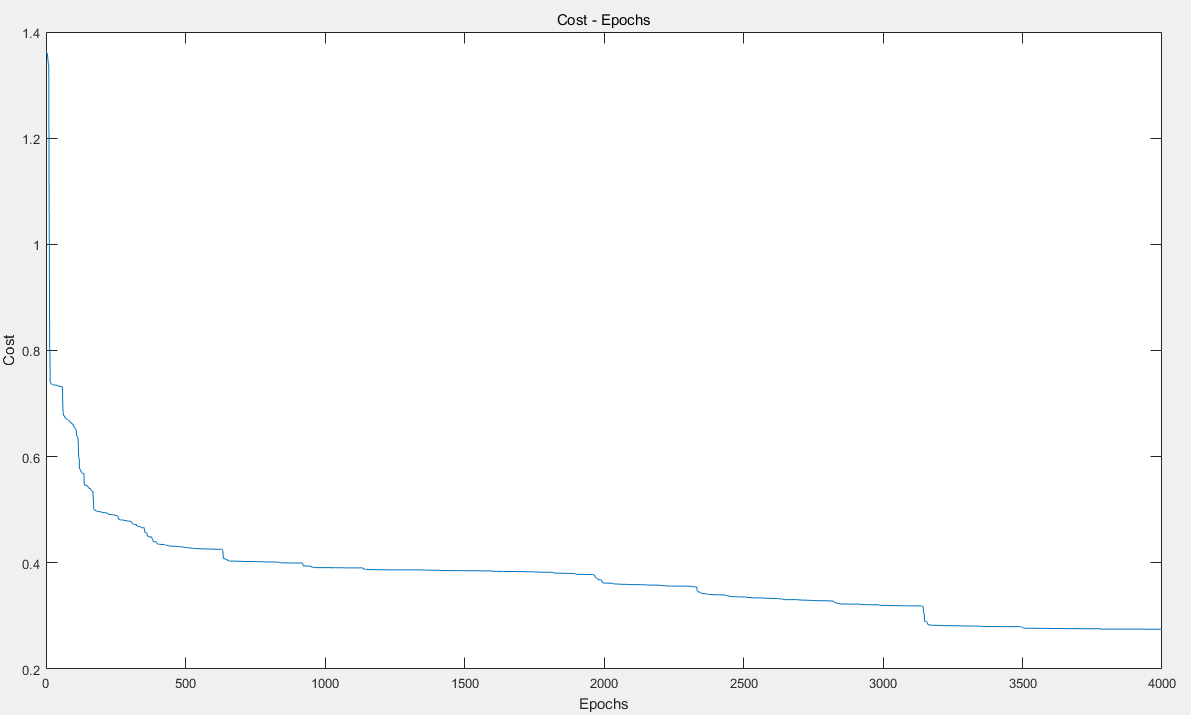
输出为：



### 3.2.2 神经网络训练

与3.1.3节中相同，训练神经网络。对于Wisconsin Cancer dataset，取其前80%的样本作为训练集，后20%的数据作为测试集。

做出代价函数关于epoch的曲线如图3.2.2所示。



**图3.2.2 代价函数曲线**

经过4000个epoch的训练，得到的神经网络，对于训练集的分类正确率为95.604%，对于测试集的分类正确率为91.228%。

## 3.3 与感知机对比

对于Iris dataset，利用Matlab中的Neural Network Training Toolbox工具箱 (nntraintool)来训练一个感知机。感知机是一个两层的神经网络，其输出层是M-P神经元，也就是“阈值逻辑神经元”，其简要结构如图3.3.1所示。

输入

θT

b

输出

**图3.3.1 感知机基本结构**

经过1000个epochs，训练得到的感知机分类正确率为82.3%，远远低于神经网络的分类正确率。

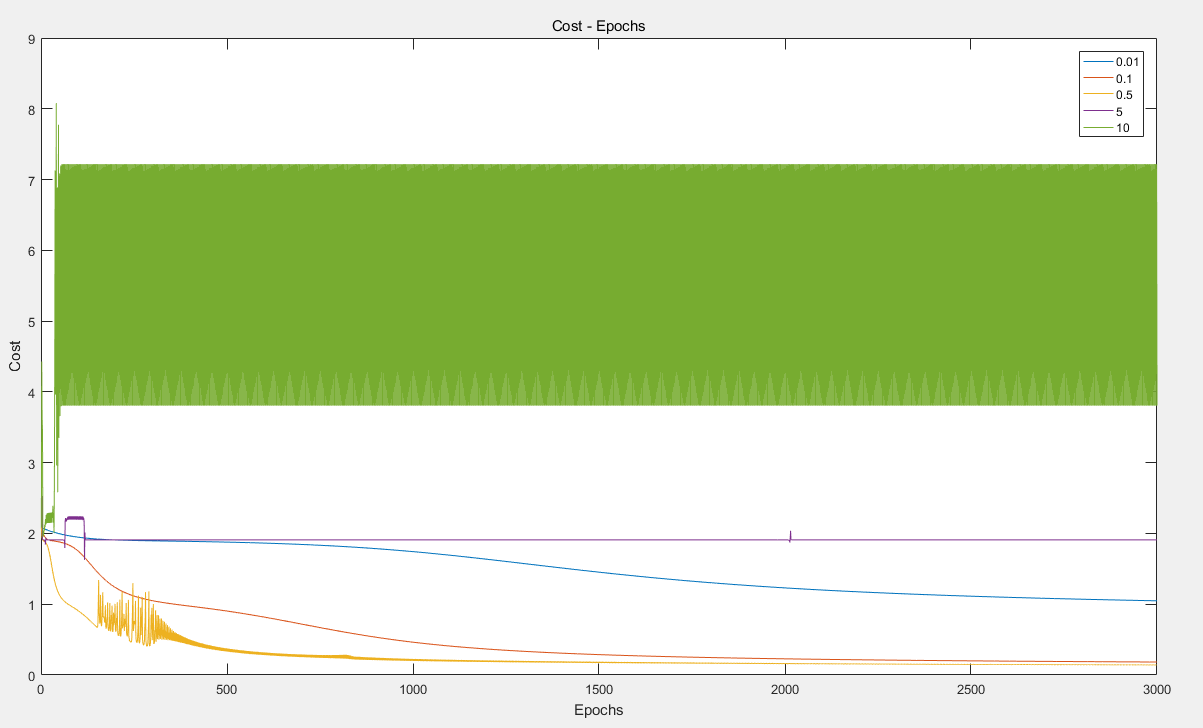
这是因为，感知机算法只有输出层神经元进行激活函数处理，即只拥有一层功能神经元，其学习能力非常的有限。若目标问题是线性可分的，则感知机算法可以收敛，从而求得适当的权向量。否则感知机的学习过程则会震荡发散。由此可见，神经网络算法由于多了一层功能神经元，可以以更高的准确率处理更多的数据。

## 3.4 改变训练参数

下面探究神经网络训练中参数变化带来的影响。仍以3.1节中对Iris dataset训练的神经网络模型为例。

### 3.4.1 改变学习速率

使用3.1.2中的权重矩阵初始值，改变学习速率，令，取训练epochs为3000，做出每种情况下的代价函数曲线如图3.4.1所示。



各个情况下，训练完成的神经网络在训练集和测试集上的分类准确率如表3.1所示。

**表3.1 各个情况的准确率**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0.01 | 0.1 | 0.5 | 5 | 10 |
| 训练集准确率 | 70% | 97.5% | 98.33% | 50.833% | 27.5% |
| 测试集准确率 | 73.33% | 100% | 100% | 50% | 23.33% |

由此可以看出，当学习速率较小时，训练的效率较低，需要迭代次数很多。并且可能会陷入局部最优解。而当学习速率过大的时候，可能会出现震荡发散的情况，分类准确率明显下降。

### 3.4.2 改变隐含层神经元数

改变中间隐含层的神经元个数，探究各种情况下的分类准确率，取学习速率，epochs为2000，结果如表3.2所示。

**表3.2 隐含层神经元数改变时分类准确率**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 隐含层神经元数 | 4 | 8 | 10 | 12 | 15 |
| 训练集准确率 | 98.33% | 97.5% | 97.5% | 97.5% | 97.5% |
| 测试集准确率 | 100% | 100% | 100% | 100% | 100% |

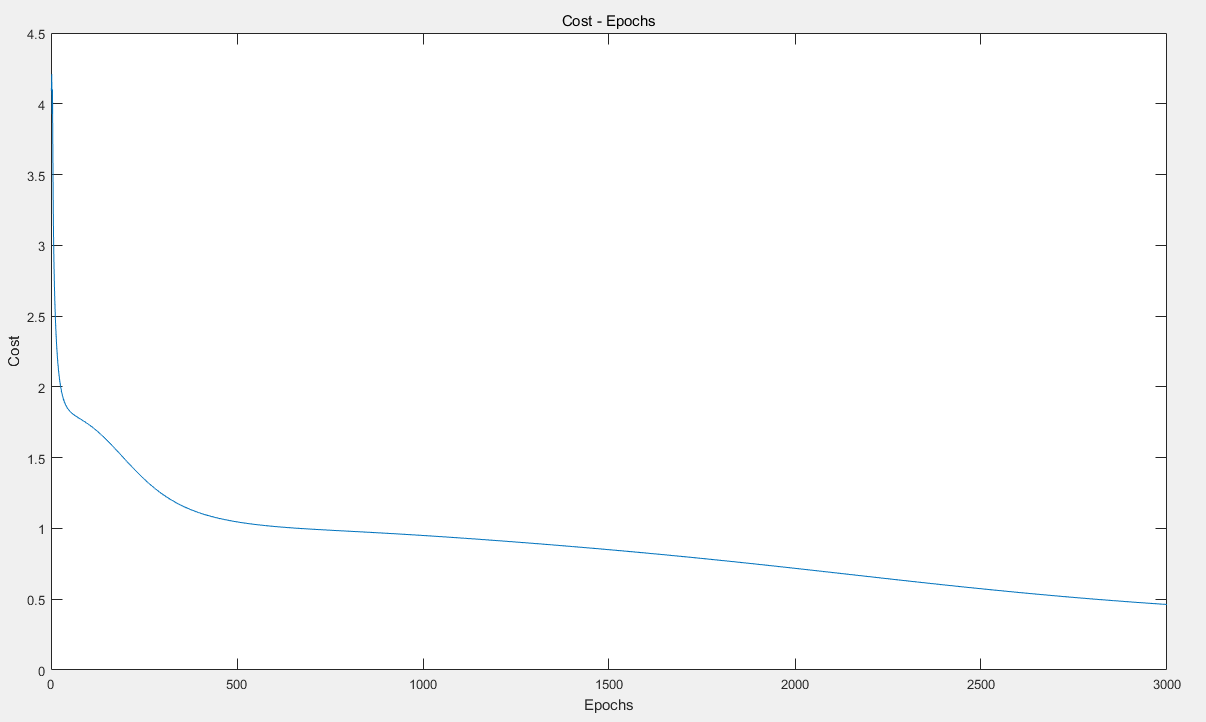
可见，隐含层神经元数改变时，对于神经网络的性能并没有太大影响。但是隐含层神经元数较大时，训练的运算量会增大很多。

# 4. 深入探究

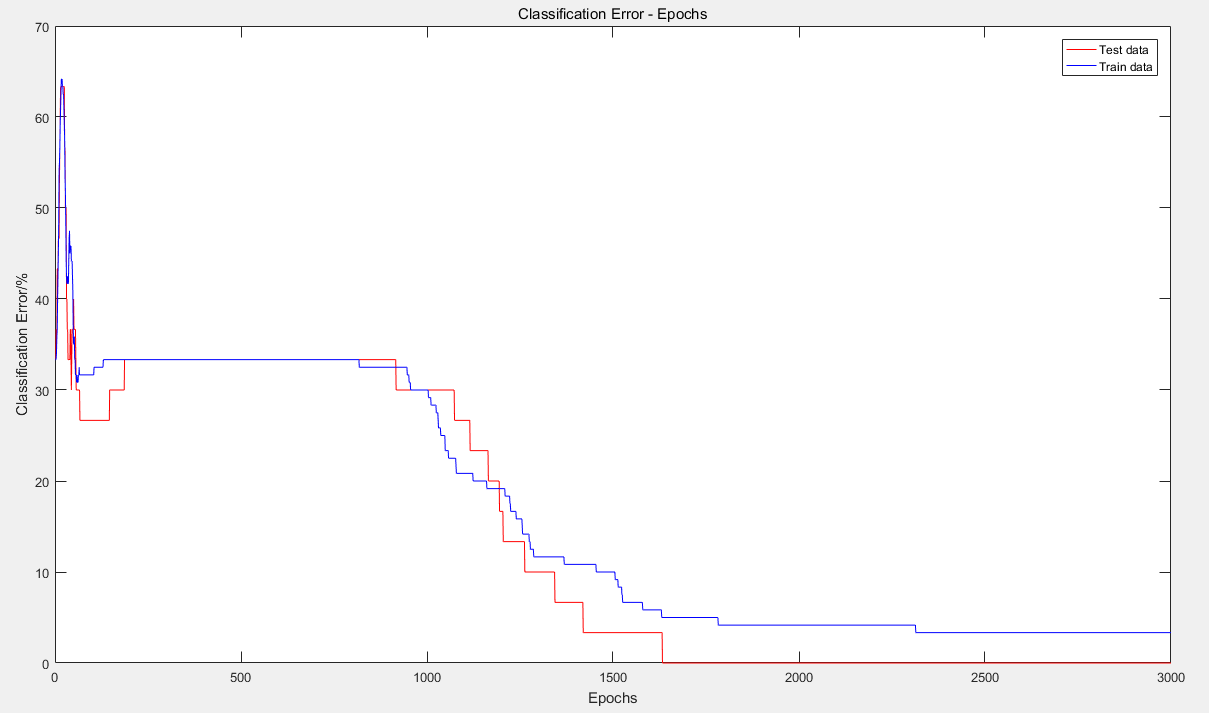
## 4.1 选择其他的激活函数

改变激活函数进行实验。除了上述的Sigmoid函数之外，选取双曲正切函数：，以及线性函数进行实验。

取，设置正则化系数、学习速率如下：。训练神经网络。做出代价函数与分类正确率关于epochs的曲线如图4.1.1和图4.1.2所示。



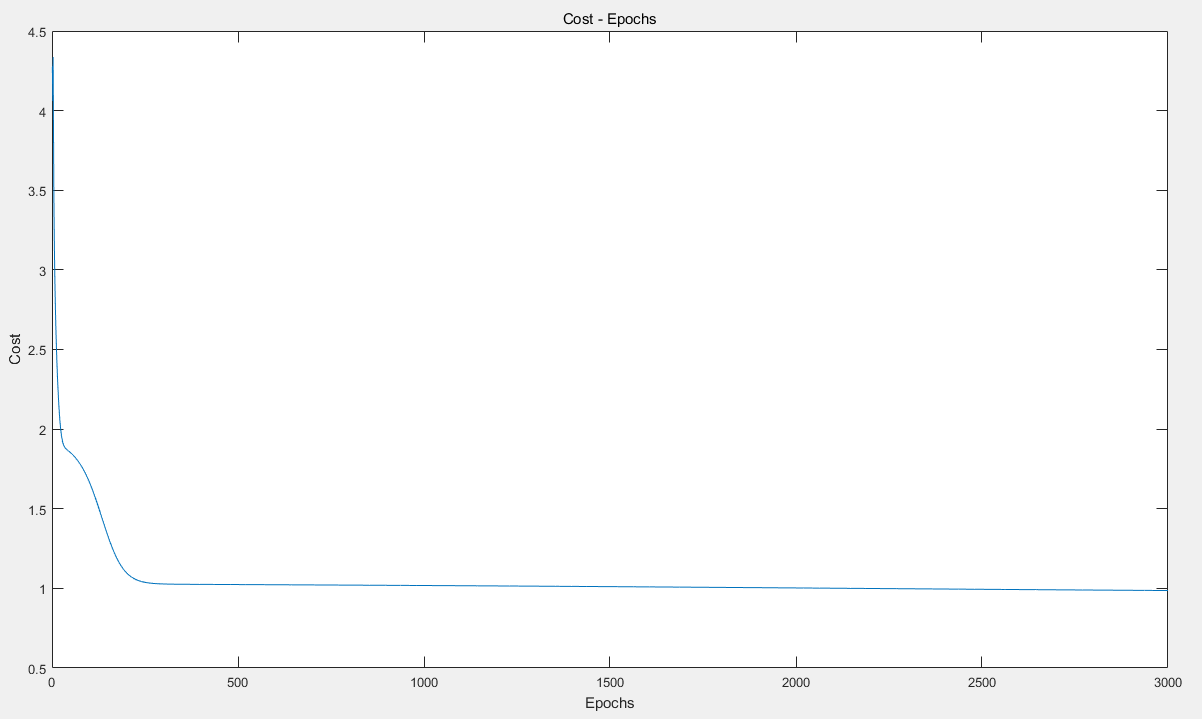
**图4.1.1 激活函数为f(x)=tanh(x)时的代价函数曲线**



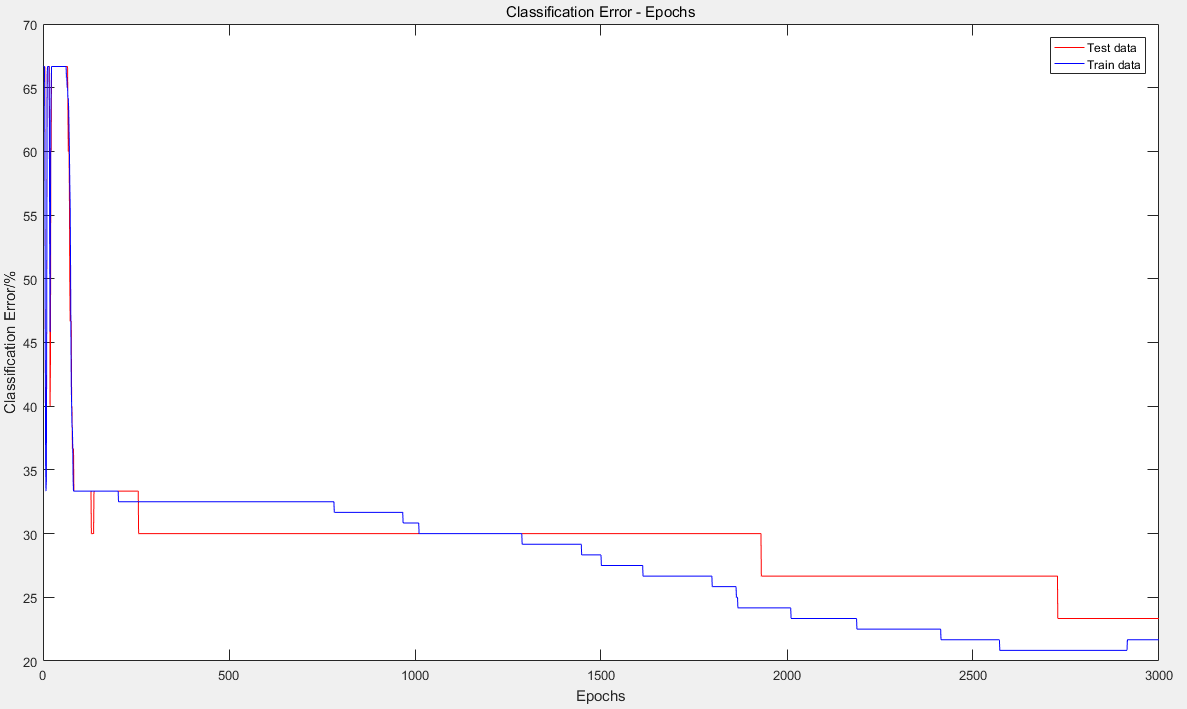
**图4.1.2 激活函数为f(x)=tanh(x)时的错误率曲线**

经过3000个epoch的训练，得到的神经网络，对于训练集的分类正确率为96.67%，对于测试集的分类正确率为100%。可见训练出了一个分类效果较好的神经网络。

取，设置正则化系数、学习速率如下：。训练神经网络。做出代价函数与分类正确率关于epochs的曲线如图4.1.3和图4.1.4所示。



**图4.1.3 激活函数为f(x)=x时的代价函数曲线**



**图4.1.4 激活函数为f(x)=x时的错误率曲线**

经过3000个epoch的训练，得到的神经网络，对于训练集的分类正确率为78.33%，对于测试集的分类正确率为76.67%。可见使用f(x)=0.25x作为激活函数的时候，训练出的神经网络分类效果较差。

## 4.2

## 4.3 训练自动编码器

利用Iris dataset训练一个自动编码器，其结构如图4.3所示。

1

x1

x2

x3

y1

y2

y3

x4

Input layer

Hidden layer

Output layer

a1

a2

a3

y4

**图4.3 自动编码器结构图**

系统的输入为Iris dataset的四维特征

# 附录：MATLAB程序代码