山东大学 SHANDONG UNIVERSITY

山东大学机器学习课程
实验报告

——实验六：神经网络的设计与实现

姓名：刘梦源
学院：计算机科学与技术学院
班级：计算机 14.4
学号：201400301007

一、 BP 神经网络

1.1 模型介绍

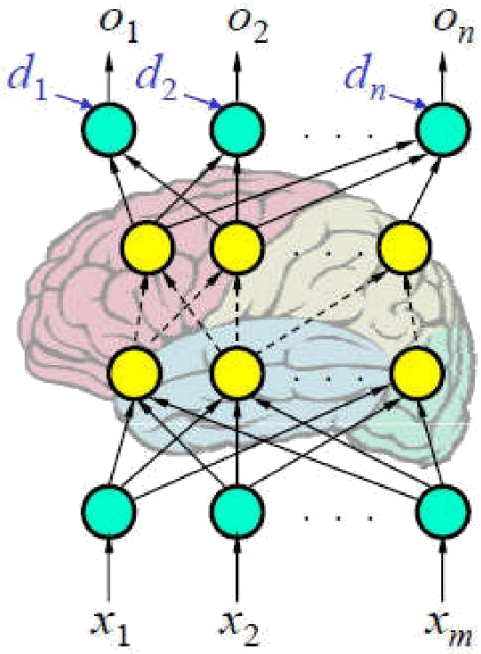

图 1.1　BP 神经网络模型

由图1.1可以看出神经网络由大量的节点和之间相互联接构成。每个节点代表一种特定的输出函数，称为激励函数。每两个节点间的连接都代表一个对于通过该连接信号的加权值，称之为权重。网络的输出则依据网络的连接方式，权重值和激励函数的不同而不同。而网络自身通常都是对自然界某种算法或者函数的逼近。而在本模型中反向传播是通过当前权值计算输出与实际输出的差值，进行梯度下降逼近，以获得逼近最终权值的效果。分为以下部分：

1) 初始化时，使用的是随机函数生成权值。

2) 正向传播时，利用当前权值进行计算。

3) 反向传播时，利用误差公式进行梯度下降逼近，计算修正量，更改权值矩阵。

4) 重复2)、3)的操作直到迭代次数足够或误差可以接受为止。

下面给出了BP神经网络的程序运行流程图：

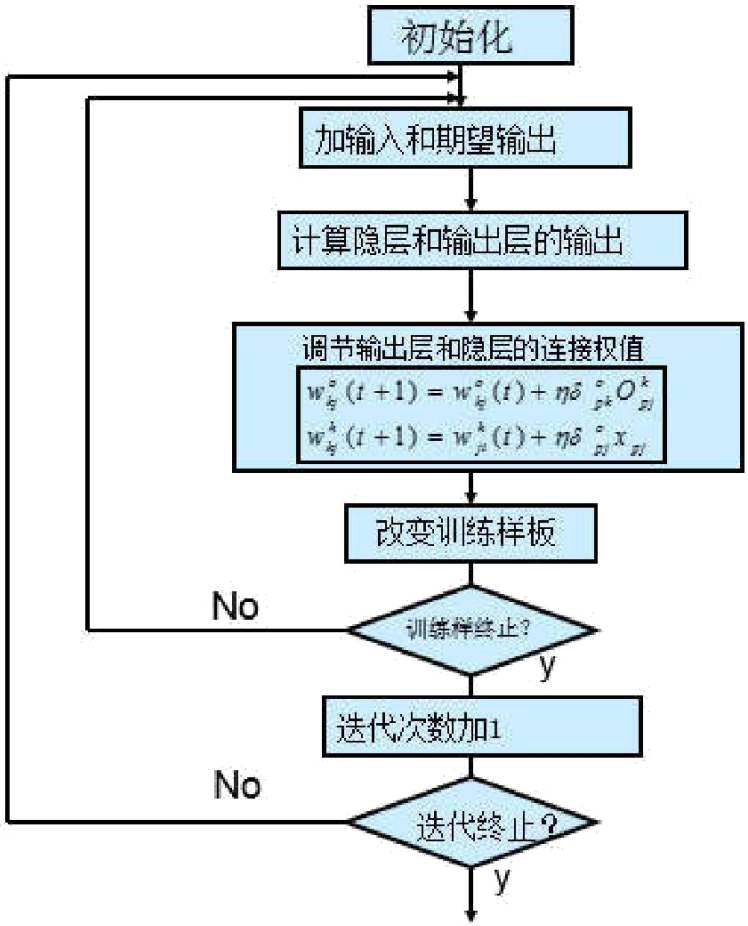

图1.2 BP神经网络的程序运行流程图

1.2符号设定[参考老师PPT]：

- Subscript i, j, k, represent different neurons, when j is the neuron of hidden layer, then i is on the left side of j, and k is on the right side of j.
- n is the iteration no.
- $E(n)$ is the sum of instantaneous error energy of the n_{th} iteration, its average is E_{av}.
- $e_j(n)$ is the error of the j_{th} neuron on the n_{th} iteration.
- $d_j(n)$ is the expected value of the j_{th} neuron on the n_{th} iteration.

- $y_j(n)$ is output of the j_{th} neuron on the n_{th} iteration; if the j_{th} neuron is the output layer, the $O_j(n)$ can be used.
- $w_{ji}(n)$ is the weight from i to j, its change is $\Delta w_{ji}(n)$.
- $v_j(n)$ is the internal state of the j_{th} neuron.
- $\varphi(.)$ is the activation function of the j_{th} neuron.
- θ_j is the threshold of the j_{th} neuron.
- $x_i(n)$ is the i_{th} element of the input sample.
- η is the learning rate.

1.3 反向传播

在这里着重说一下反向传播，这个是核心。在反向传播过程中，我们使用以下公式作为梯度逼近的应用公式：

$ej(n)=dj(n)-yj(n)$ ----------误差计算公式

$E(n)=12\Sigma ej2(n)j\in C$ ----------即时误差(Instantaneous error)公式

$vj(n)=\Sigma wji(n)yj(n)Pi=0$ -------输入净值公式

$yj(n)=\varphi j\ (vj(n))$---------- 激励输出公式

$\Delta wji(n)=\eta\delta j(n)\cdot yi(n)$----------变化计算公式

$\delta j(n)=(dj(n)-yj(n))\ \varphi j'(vj(n))$---输出层偏导公式

$\delta j(n)=\varphi j'(vj(n))\Sigma(dj(n)-yj(n))k\ \varphi'(vk(n))wkj(n)$隐含层偏导公式

$\varphi'(vj(n))=yj(n)[1-yj(n)]$ Sigmoid函数导数公式

1.4 模型参数的确定

为了达到更高的准确率，我们需要确定模型的参数与其确定方案如下：

- 网络层数：3层——此参数基于可行性与可实现性人工确定
- 各层神经元个数
  - 输入层：3个（二维的输入向量与一维常数向量）
  - 隐含层：6个（通过测试比较确定）
  - 输出层：1个（根据Label的值而确定）
- 权值矩阵：初始值由随机数生成
- 学习率：0.03——此参数基于测试比较决定
- 迭代最大次数：350——此参数基于测试比较决定

```
%BP神经网络参数的声明和设置
iterlimit=350;%迭代次数
alpha=0.03;%学习率
innum=3;%神经网络的层数
hidnum=4;%隐含层节点的数目
outnum=1;%输出层节点的数目
```

图 1.2 程序中模型参数的确定

基于以上的模型参数，我们可以得到正确率为 100%的结果。
那么如何用图形的方式表达出我们的分类结果呢？我首先画出了 500 个测试集的点，蓝色的‘o’和红色的‘o’分别代表了不同的 Label。
如下面两张图所示：其中，第一张图即图 1-3.1 用‘o’来表示真实的测试数据。而第二张图即图 1-3.2 用红色的‘X’和蓝色的‘X’来表示我们预测的分类结果。可以看出我们预测的正确率为 100%。同时为了视觉上的对比，图 1.4 给出了当迭代次数减少为 50 次时，模型对测试集合的分类情况，可以看出有些测试集分类发生了错误，蓝色的‘o’上出现了红色的‘X’

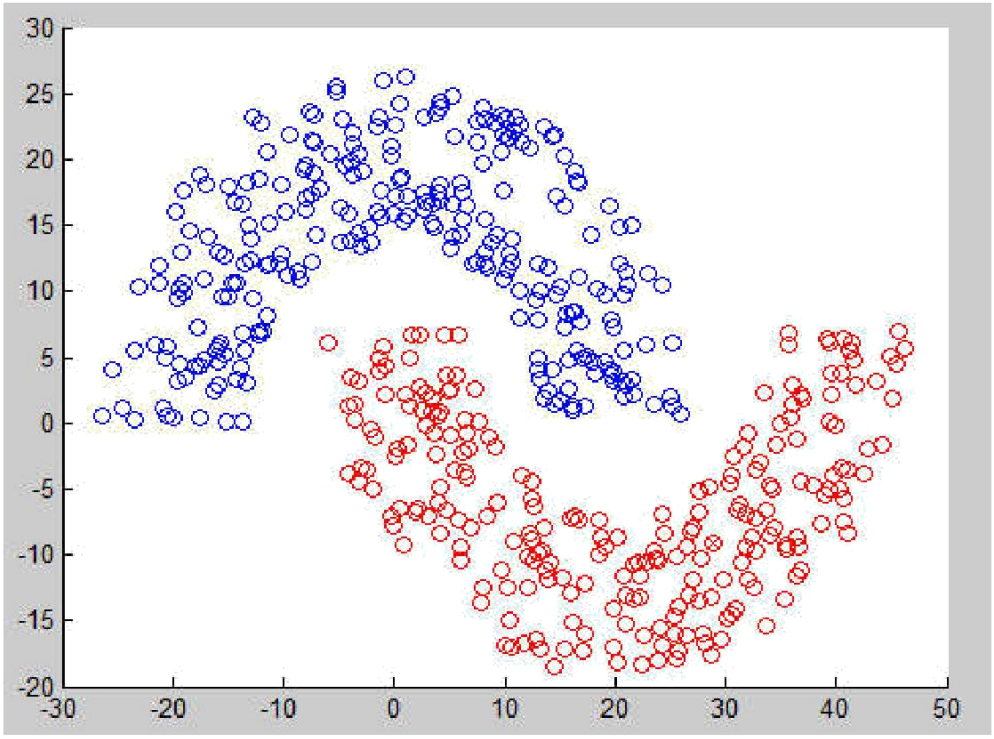

图 1-3.1 500 个测试集的真实 label

然后我们使用蓝色的‘X’完全匹配蓝色的‘o’表明预测的分类标签，分类结果如下图所示：

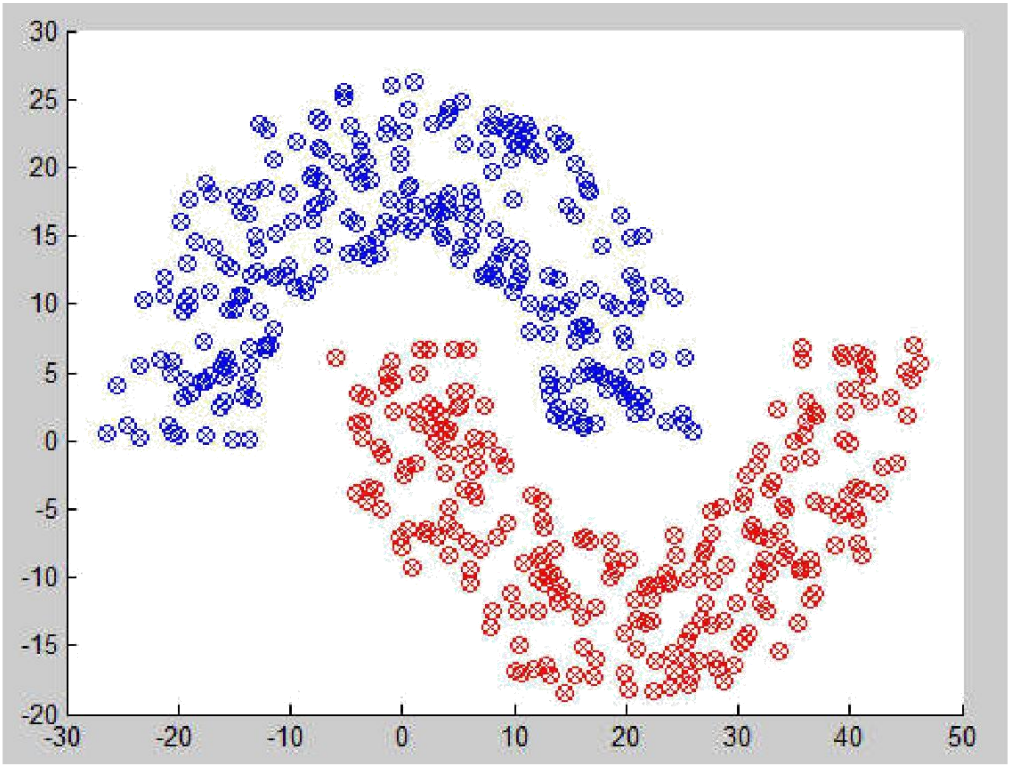

图 1-3.2 蓝色的‘X’完全匹配蓝色的‘o’表明分类全部正确

下图可以看出当迭代次数减到 50 次时：蓝色的‘o’上出现了红色的‘X’，红色的‘o’上出现了蓝色的‘X’，表明分类错误。

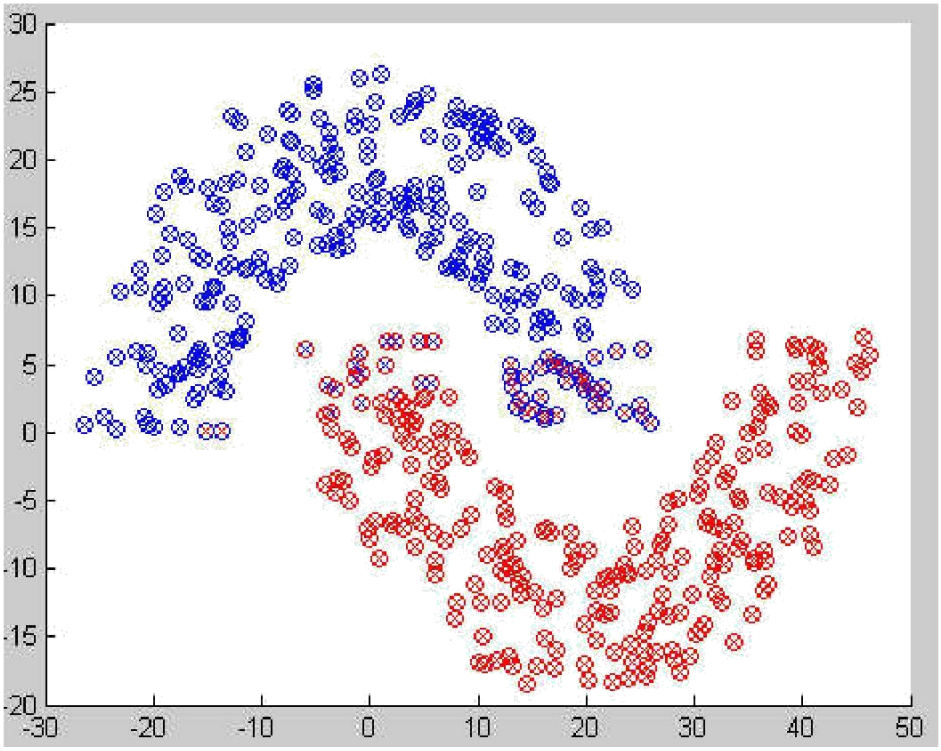

图1.4 当迭代50次时，错误率为5.40%

1.3 模型参数的验证与测试

1.3.1 隐含层神经原个数测试

条件与限定：为了测试隐含层神经原个数，我们需要使用固定变量法。即确保学习率 0.03 和最大迭代次数 350 不变。下面的数据均是通过 10 次运行程序分别对运行时间和错误率求均值得到：

隐含层个数	错误率	运行时间
1	47.60%	6.770534
2	10.40 %	6.995396
3	0.40 %	7.087463
4	0.01%	8.187613
5	0.01%	8.187613
6	0.00%	8.187612
7	0.00%	8.767734
8	0.00%	8.767731

表 1.1 BP 下隐含层个数变化对结果的影响

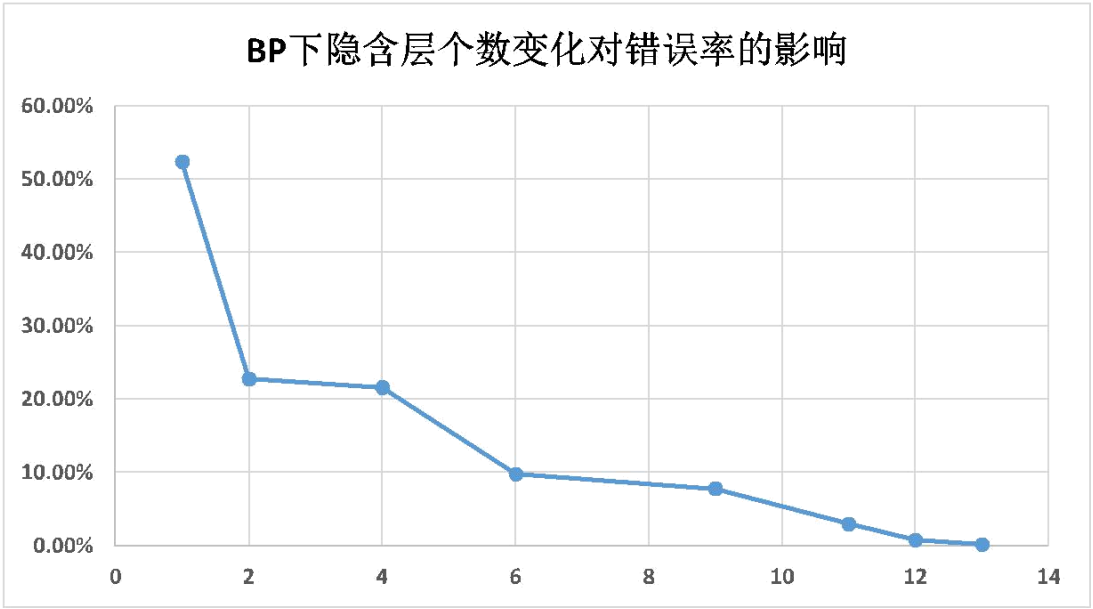

图 1.5　BP 下隐含层个数变化对错误率的影响

1.3.2 学习率 alpha 的测试

条件与限定：为了测试学习率 alpha，我们需要使用固定变量法。即确保隐含层神经原个数为 4,最大迭代次数 350。下面的数据均是通过 10 次运行程序对运行时间和错误率求均值得到：

学习率alpha	错误率	运行时间
0.001	9.40 %	7.874524
0.01	0.40 %.	8.019597
0.02	0.20 %	7.281995
0.03	0.00%	8.332000
0.04	0.01 %	7.866376
0.1	2.80 %	7.566376
0.15	3.40 %	7.238741
0.2	4.40 %	7.928436
0.4	3.60 %	6.888274
0.6	17.00 %	6.942554

表 1.2 学习率 alpha 变化对结果的影响

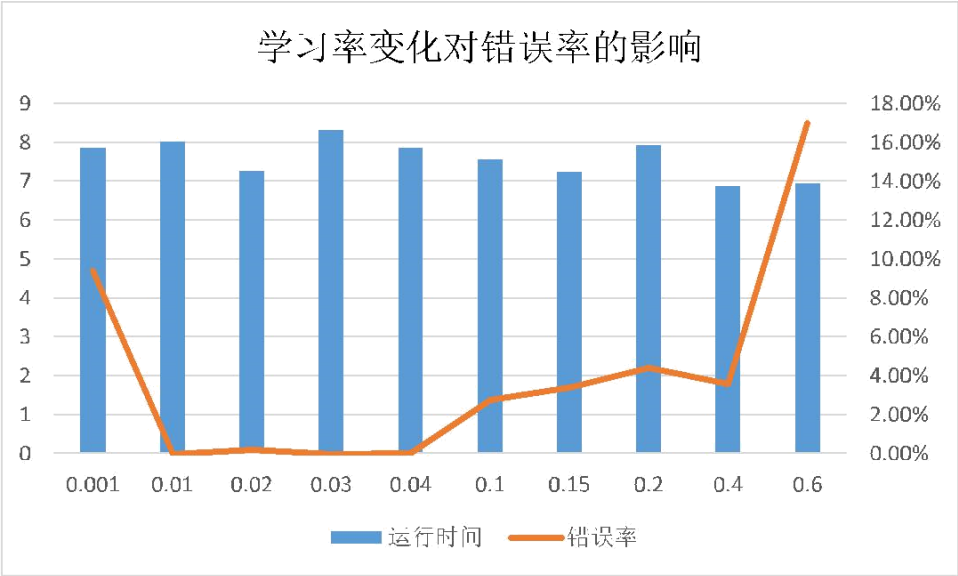

图 1.6 BP 学习率变化变化对错误率的影响

1.3.3 最大迭代次数变化的测试

条件与限定：为了测试最大迭代次数，我们需要使用固定变量法。即确保隐含层神经原个数为 4，学习率 alpha 为 0.03。下面的数据均是通过 10 次运行程序对运行时间和错误率求均值得到：

最大迭代次数	错误率	运行时间
50	5.40 %	1.032117
100	2.80 %	2.047297
150	0.80 %	3.139875
200	0.30%	4.092254
250	0.20 %	7.866376
300	0.10 %	7.566376
350	0.00 %	7.238741
400	0.00 %	7.928436
450	0.00 %	6.888274
500	0.00 %	6.942554

表 1.3 最大迭代次数变化对结果的影响

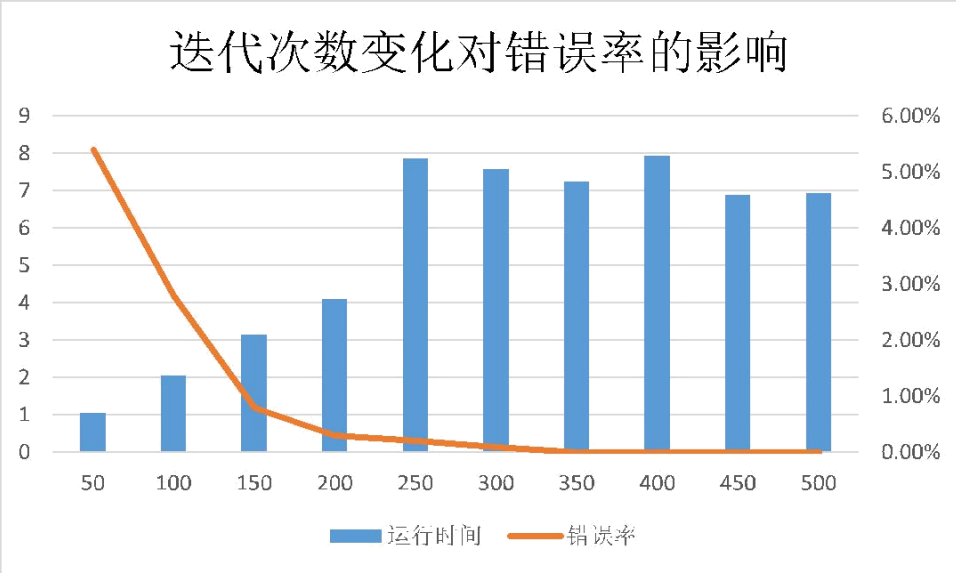

图 1.7 迭代次数变化变化对错误率的影响

从以上三个验证测试我们可以得出以下结论：

1.首先学习率 alpha 应保持较低水平最好小于 0.05，以保证梯度下降过程中不会出现明显的震荡情况，影响神经网络的效果。
2.然后隐含层个数应保证在输入层个数以上，使得整个网络可以比较合适地给出结果
3.最大迭代次数应该保证一定的数量，过少无法保证梯度下降到近乎收敛的程度，过多的话将会使得整个运行时间增加很多。本次实验的最大迭代次数选择了 350

二、RBF 网络模型

2.1 模型介绍

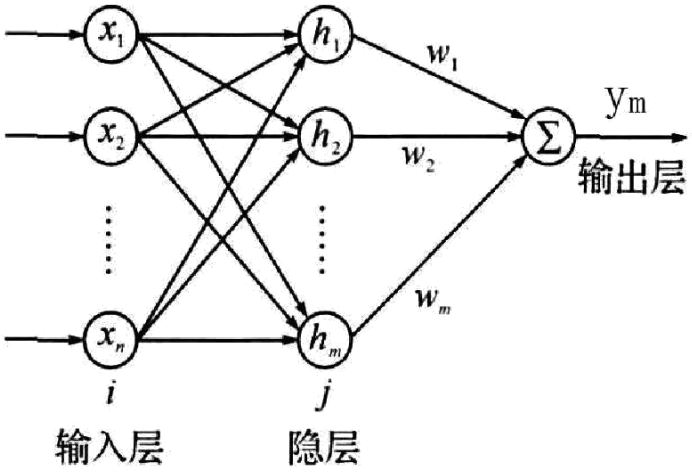

图 2-1 RBF网络结构

RBF网络的结构与多层前向网络类似，它是一种三层前向网络。第一层即输入层由信号源节点组成；第二层为隐含层，隐单元数视所描述的问题的需要而定，隐单元的变换函数是RBF，它是对称中心径向对称且衰减的非线性函数；第三层为输出层，它对输入模式的作用做出响应。由于输入到输出的映射是非线性的，而隐含层空间到输出空间的映射是线性的，从而可以大大加快学习速度并避免局部极小问题。本模型中径向基函数使用了基于高斯函数作为基函数的网络，并进行正则化与自组织选择扩展常数了，其中使用Green函数正则化了隐含层的权值，并修正了函数F的形式，最后对于输出层部分使用了瞬时梯度下降的方法收敛权值取值。具体分为以下部分：

1) 初始化时，使用的是随机函数生成权值。

2) 扩展常数的自组织选择与Green函数值的计算（正则化）。

3) 输出层计算输出，并利用误差公式进行梯度下降逼近，计算修正量，更改权值矩阵。

4) 重复2)、3)的操作直到迭代次数足够或误差可以忍受为止。

在整个过程中，我们使用以下公式作为径向基函数算法的应用公式：

i. $\delta j=\lambda \min i\|cj-ci\|$ 自组织选择扩展常数公式

ii. $G(X,Xp)=\exp(-12\sigma 2\|X-Xp\|2)$ Green函数在多元Gauss函数形式

iii. $F(X)=\Sigma wpG(X,Xp)Pp=1$ 修正的插值函数

iv. $E=12\Sigma ei2Pi=1$ 即时误差公式

v. $ei=di-(Xi)=di-\Sigma wjG(\|X-cj\|)Mj=1$ 误差计算公式

vi. $\Delta wj=\eta\Sigma ei(\|X-cj\|)Pi=1$ 变化计算公式

2.2 模型参数的确定

RBF 神经网络学习算法需要求解的参数有 3 个：基函数的中心、隐含层到输出层权值以及节点基宽参数。

对于本模型参数的确定，我们需要确定的参数与其确定方案如下：

- 网络层数：3层——此参数基于可行性与可实现性人工确定
- 各层神经元个数

输入层：3个（二维的输入向量与一维常数向量）

隐含层：12个（通过测试比较确定）

输出层：1个

- 权值矩阵：初始值由随机数生成
- 学习率：0.2 ——此参数基于测试比较决定
- 迭代最大次数：200——此参数基于测试比较决定

```
%RBF神经网络参数的声明和设置
iterlimit=200;%迭代次数
alpha=0.2;%学习率
innum=2;%神经网络的层数，RBF权相连得有两层
hidnum=12;%隐含层节点的数目
outnum=1;%输出层节点的数目
```

图 2-2 程序中 RBF 神经网络参数的设置

基于以上这些参数，我们可得结果为错误率为0/500=0%，运行时间0.706964秒

结果具体展示如下：

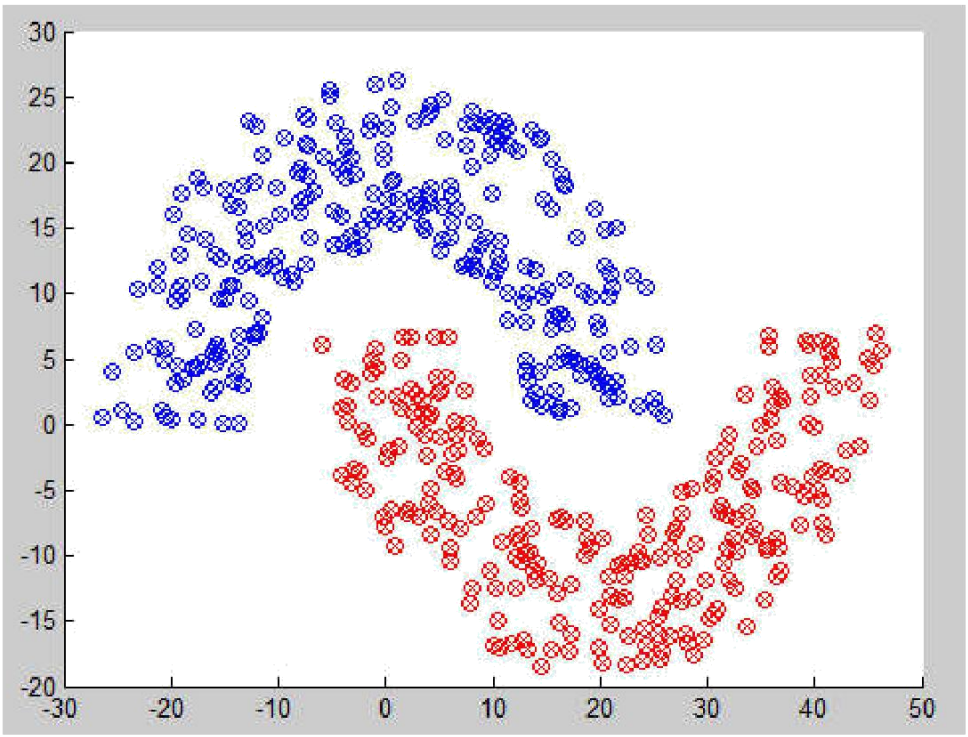

图 2-3　　按照以上参数利用 RBF 模型全部分类正确

2.2 模型参数的测试比较

2.3.1 隐含层神经原个数测试

条件与限定：为了测试隐含层神经原个数，我们需要使用固定变量法。即确保学习率 0.2 和最大迭代次数 200 不变。下面的数据均是通过 10 次运行程序对运行时间和错误率求均值得到：

隐含层个数	错误率	运行时间
1	52.10%	0.676694
2	22.80%	1.039694
4	21.60%	0.676694
6	9.80%	1.054543
9	7.80%	0.736899
11	3.00%	0.957462
12	0.80%	1.069193
13	0.20%	0.786694
14	0.00%	0.776694

表 2-1RBF 下隐含层个数变化对结果的影响

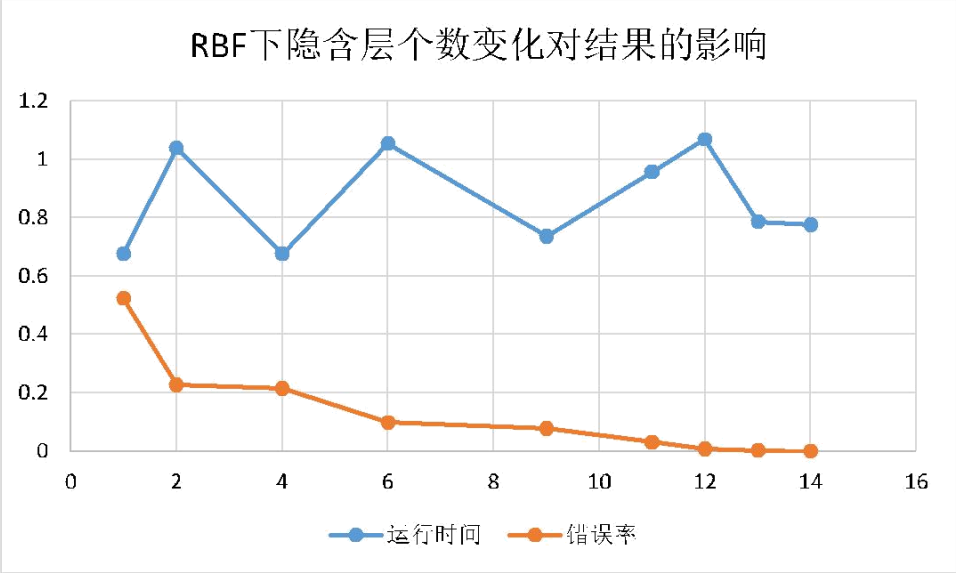

图 2-4 RBF 下隐含层个数变化对错误率的影响

2.3.2 学习率 alpha 的测试

条件与限定：为了测试学习率 alpha，我们需要使用固定变量法。即确保隐含层神经原个数为 12, 最大迭代次数为 200。下面的数据均是通过 10 次运行程序对运行时间和错误率求均值得到：

学习率alpha	运行时间	错误率
0.02	0.676694	0.20%
0.07	1.039694	0.30%
0.12	0.676694	0.20%
0.17	1.054543	0.10%
0.20	0.736899	0.00%
0.27	0.957462	0.20%
0.32	1.069193	0.60%
0.37	0.786694	0.20%
0.42	0.776694	0.70%

表 2-2 学习率 alpha 变化对结果的影响

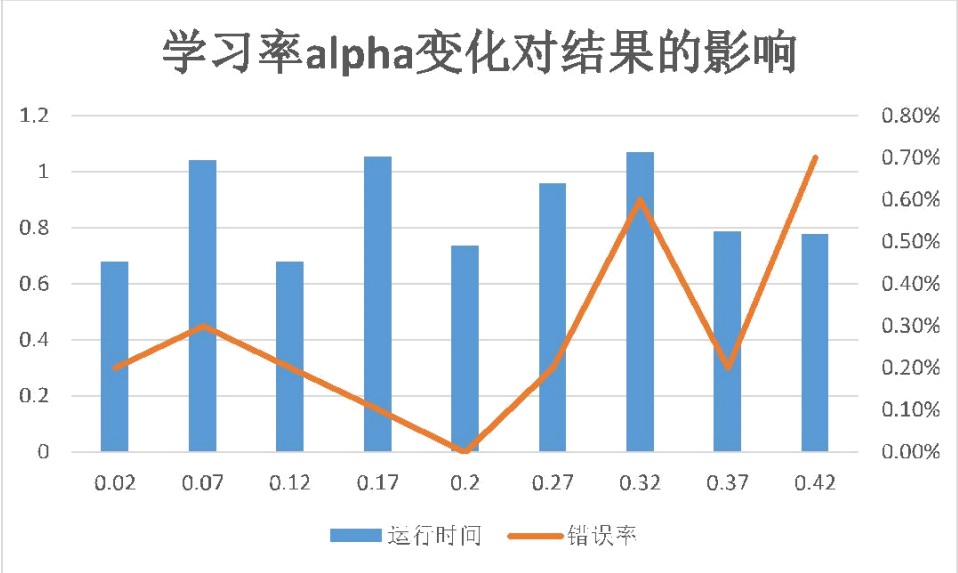

图 2-5 RBF 学习率变化变化对错误率的影响

2.3.3 最大迭代次数变化的测试

条件与限定：为了测试最大迭代次数，我们需要使用固定变量法。即确保隐含层神经原个数为 12, 学习率 alpha 为 0.2。下面的数据均是通过 10 次运行程序对运行时间和错误率求均值得到：

最大迭代次数	错误率	运行时间
50	0.866414	0.20%
100	1.165579	0.40%
150	1.975426	1.00%
200	1.657955	0.00%
250	1.513507	0.20%
300	1.305575	1.00%
350	1.520349	0.20%
400	1.720854	0.80%

表 2-3 最大迭代次数变化对结果的影响

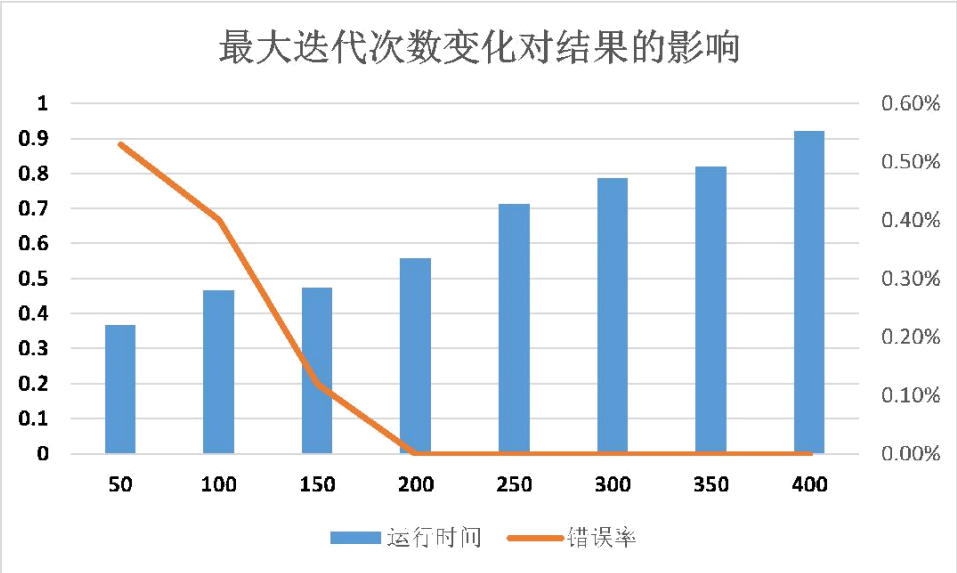

图 2-6 最大迭代次数变化对结果的影响

三、SVM 模型

3.1 libSVM 库的直接调用

此次使用的是LibSVM的SVM工具集，借用其中的matlab包进行运行与测试，其中使用的两个外调函数是：

1.训练函数：

model = svmtrain(training_label_vector, training_instance_matrix [, 'libsvm_options']);

2.测试函数

[predicted_label] = svmpredict(testing_label_vector, testing_instance_matrix, model [, 'libsvm_options']);

由于不需要参数设置，所以仅做了数据导入工作，故无相关比较。

结果为错误率为 0/500=0%，运行时间 0.675061 秒（核心算法部分运行时间，下同），结果具体展示如图 2-3（由于最后确定的参数下各方法结果均无错误，故图片不再展示）。

3.2 BP、RBF、SVM 三个模型的比较

- 结果的比较：均能够达到 100%的分类正确率。
- 性能的比较：我们基于正确率相同情况下的最低稳定参数要求，比较了效率情况（所用时间，单位：秒）：

模型	时间（秒）
BP	7.34
RBF	0.79
SVM	0.67

表 3-1 三个模型性能的比较

至此，实验得到了较为理性的结果。