

# BP 人工神经网络在渤海湾叶绿素预测中的应用

何恩业, 李 海, 任湘湘, 蔡 怡, 杨 静

(1. 国家海洋环境预报中心, 北京 100081)

摘 要: 利用误差反向传播(BP)人工神经网络模型建立了天津赤潮驴驹河赤潮监控区各种理化因子与叶绿素-a 之间的映射关系。模拟结果表明, 该模型能够较好的反映各种理化因子与叶绿素-a 浓度之间的非线性对应关系, 可间接地用于该地区的赤潮预测。

关键词: BP 人工神经网络; 叶绿素-a 浓度; 赤潮

中图分类号: P731 文献标识码: A 文章编号: 1003 - 0239 (2008) 2 - 0001 - 10

## 1 引言

有效地预测赤潮发生对于海洋防灾减灾具有重要的现实意义,但在对赤潮形成机理尚不明确而沿海海洋经济开发又切实需要的情况下,利用各种统计方法对赤潮发生进行预测就成了目前赤潮预测研究的一个重要方面。人工神经网络方法是对赤潮进行预测的各种统计方法中较为有效的一种非线性拟合方法,人工神经网络方法(ANN)是人脑及其活动的一个理论化的数学模型,是一个大规模的非线性自适应系统<sup>[1]</sup>。虽然人工神经网络已经发展了半个多世纪,但将其应用于赤潮预测却是近几年来才开始的。吴京洪等(2001)根据大亚湾澳头海域养殖区 1997~1999 年的赤潮监测数据,采用误差反向传播人工神经网络模型,研究了浮游植物总生物量与理化因子的关系,并对各因子的重要性作了分析<sup>[2]</sup>。蔡如钰(2001)利用石城岛和王家岛附近海域的调查数据,建立了当地海域各种理化因子与夜光藻密度间的人工神经网络映射模型,并利用该模型对各种理化因子与夜光藻密度的非线性对应关系的外向扩展和内插进行了研究,对了解夜光藻赤潮发生的机理具有一定的意义<sup>[3]</sup>。王洪礼等(2006)为了研究各种理化因子与赤潮藻类密度间非线性对应规律和有效地预测赤潮藻类密度,利用同样的数据构建了基于 BP 算法的一个四层模糊神经网络模型(FNN),并将该技术引入赤潮预测研究,较好的反演出各种理化因子与夜光藻密度的非线性对应变化规律<sup>[4]</sup>。

本文利用 2006 年 5 月下旬至 8 月末天津驴驹河赤潮监控区的连续赤潮监测资料,采用人工神经网络方法中应用最为广泛的一种算法-BP 算法,建立了监测区各海洋理化因子与叶绿素-a 间的 BP 人工神经网络模型,反演各理化因子与叶绿素-a 密度间的非线性对应关系。BP 网络是人工神经网络中的一种多层前馈网络的学习算法,它可以通过神经网络的自学习功能,确定神经元之间的耦合权值,从而使网络整体具有近似函数的功

收稿日期: 2007-12-19; 修订日期: 2008-04-03

基金资助: 国家“十五”科技支撑项目(2006BAV03B01)

作者简介: 何恩业(1981-),男,助理工程师,主要从事赤潮预报研究工作。

能,非常适用于非线性系统的建模研究。应用 BP 网络可以避开赤潮形成的复杂生态动力学机理而对直接反映浮游植物密度的叶绿素浓度进行预测。

## 2 双隐层 BP 网络结构

BP 网络,即误差反向传播网络,是一种非线性映射人工神经网络,具有很强的学习和记忆能力。人工神经网络一般由一个输入层、若干隐含层和一个输出层组成。它是一种有导师的学习算法,这种算法通常是应用最速下降法来调整各层权值,并最终达到预先给定的误差为止。网络的信号传送由正向传播和反向传播两部分组成:在正向传播阶段,学习样本送入输入层,经隐层逐层运算后传至输出层,每一层神经元的状态只影响下一层神经元的状态,如果输出层没有得到期望的输出结果,即网络的实际输出与期望输出之间存在较大的误差,则计算输出层的误差变化值,然后输入误差反向传播;在误差反向传播过程中,误差信号沿着原来的连接从输出层返回至输入层,并逐层调整耦合权值,以使误差达到最小。由于输出层对误差的调整首先必须通过改变隐层与输出层之间的联接权值来实现,而隐层要能对输出层反传过来的误差进行调整,也需调整前一隐层(或输入层)与它之间的联接权值,如此反复直到实际输出与期望输出之间差的绝对值小于预先给定的误差<sup>[5]</sup>。

### 2.1 BP 网络结构模型

具有两个隐层的 BP 网络结构(见图 1),该网络共分 4 层:

第 1 层为变量输入层,  $x_j, j=1, 2, \wedge, n_0$  为输入变量,  $n_0$  为输入变量的个数。若设  $x_0$  为第一隐层中激活函数的域值,则输入向量总共为  $n_0+1$  维。 $x_0$  一般取为 -1, 将其增广到输入量中, 作为一个分量, 则有  $x=(x_0, x_1, x_2, \wedge, x_{n_0})$ ;

第 2 层为第 1 隐层, 设有  $n_1$  个神经元, 则其输出向量为  $g=(g_0, g_1, g_2, \wedge, g_{n_1})$ , 其中  $g_0$  为第一隐层中激活函数的域值, 一般取为 -1;

第 3 层为第 2 隐层, 设有  $n_2$  个神经元, 其输出向量为  $h=(h_0, h_1, h_2, \wedge, h_{n_2})$ ,  $h_0$  为输出层中激活函数的域值, 一般取为 -1;

第 4 层为输出层, 设有  $m$  个神经元, 网络的输出向量为  $y=(y_0, y_1, \wedge, y_m)$ 。

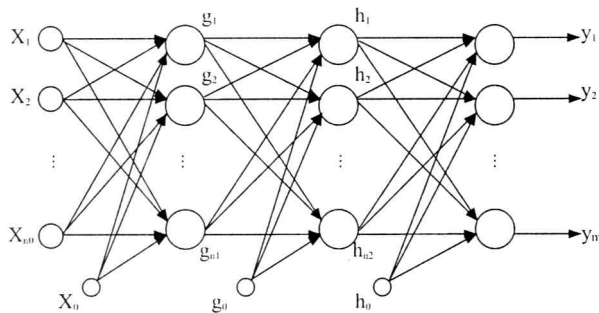


图 1 具有两个隐层的 BP 网络

第 1 隐层和第 2 隐层神经元采用  $S$  型对数激活函数 (单极性 Sigmoid 函数), 输出层神经元采用线性激活函数。  $S$  型对数激活函数表达式为:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

第 1 隐层输出向量为:  $g_j = f(\sum_{i=0}^{n_0} w_{ji} x_i - \theta_j), \quad j = 1, 2, \dots, n_1$

第 2 隐层输出向量为:  $h_k = f(\sum_{j=0}^{n_1} v_{kj} g_j - \theta_k), \quad k = 1, 2, \dots, n_2$  (2)

输出层输出向量为:  $y_l = f(\sum_{k=0}^{n_2} r_{lk} h_k - \theta_l), \quad l = 1, 2, \dots, m$

上面各式中,  $W$  是输入层到第 1 隐层间的耦合权值矩阵, 其分量为  $w_{ji}, j=1, \dots, n_1; i=0, 1, \dots, n_0$ ;  $V$  是第 1 隐层到第 2 隐层之间的耦合权值矩阵, 分量表示为  $v_{kj}, k=1, \dots, n_2; j=0, 1, \dots, n_1$ ;  $R$  是第 2 隐层到输出层之间的耦合权值矩阵, 分量表示为  $r_{lk}, l=1, \dots, m; k=0, 1, \dots, n_2$ , 为神经元的域值, 为了简便将其视为联接权值来处理, 即令  $\theta_j = w_{j0}, \theta_k = v_{k0}, \theta_l = r_{l0}, x_0 = g_0 = h_0 = -1$ , 于是上方方程可以改成如下形式:

$$\begin{cases} g_j = f(net_j), j = 1, 2, \dots, n_1; & net_j = \sum_{i=0}^{n_0} w_{ji} x_i, j = 1, 2, \dots, n_1 \\ h_k = f(net_k), k = 1, 2, \dots, n_2; & net_k = \sum_{j=0}^{n_1} v_{kj} g_j, k = 1, 2, \dots, n_2 \\ y_l = f(net_l), l = 1, 2, \dots, m; & net_l = \sum_{k=0}^{n_2} r_{lk} h_k, l = 1, 2, \dots, m \end{cases} \quad (3)$$

## 2.2 双隐层 BP 网络权值的动态学习算法

设一共有  $\alpha$  个样本对, 第  $p$  个样本对  $(x^p, t^p)$  输入模式后输出方差为:

$$E_p = \frac{1}{2} \sum_{l=1}^m (t_l^p - y_l^p)^2 \quad (4)$$

$\alpha$  个样本对经正向传播运算后网络总误差为:

$$E_\Sigma = \frac{1}{2} \sum_{p=1}^{\alpha} \sum_{l=1}^m (t_l^p - y_l^p)^2 \quad (5)$$

其中  $y_l^p$  表示网络的实际输出,  $t_l^p$  表示期望输出。取  $E_\Sigma$  为目标函数, 应用最速下降法, 反向调整各层联接权值, 使误差达到最小。设  $w_{uv}$  为网络任意两个神经元之间的联接权, 沿负梯度方向调整联接权  $w_{uv}$ , 调整的大小为  $\Delta w_{uv} = -\eta \frac{\partial E_\Sigma}{\partial w_{uv}}$ , 利用此方法可以得到各层之间权值

的修正量。第 2 隐层到输出层的联接权修正量为：

$$\Delta r_{lk} = \eta \sum_{p=1}^{\alpha} \delta_l^p h_k^p \quad \delta_l^p = f'_{netl} \Delta_l^p \quad \Delta_l^p = (t_l^p - y_l^p) \quad (6)$$

第 1 隐层到第 2 隐层的联接权修正量为：

$$\Delta v_{kj} = \eta \sum_{p=1}^{\alpha} \delta_k^p g_j^p \quad \delta_k^p = f'_{netk} \Delta_k^p \quad \Delta_k^p = \sum_{i=1}^m \delta_i^p r_{lk} \quad (7)$$

输入层到第 1 隐层的联接权修正量为：

$$\Delta w_{ji} = \eta \sum_{p=1}^{\alpha} \delta_j^p x_i^p \quad \delta_j^p = f'_{netj} \Delta_j^p \quad \Delta_j^p = \sum_{k=1}^{n_2} \delta_k^p v_{kj} \quad (8)$$

设  $n$  为迭代次数，应用梯度法可以得到网络各层联接权的迭代关系为：

$$\begin{cases} r_{lk(n+1)} = r_{lk(n)} + \Delta r_{lk}, & l=1, 2, \wedge, m; k=0, 1, 2, \wedge, n_2 \\ v_{kj(n+1)} = v_{kj(n)} + \Delta v_{kj}, & k=1, 2, \wedge, n_2; j=0, 1, 2, \wedge, n_1 \\ w_{ji(n+1)} = w_{ji(n)} + \Delta w_{ji}, & j=1, 2, \wedge, n_1; i=0, 1, 2, \wedge, n_0 \end{cases} \quad (9)$$

为了利于网络联接权加快脱离误差曲面的平坦区，引进动量项，则以上方程组变为：

$$\begin{cases} r_{lk(n+1)} = r_{lk(n)} + \Delta r_{lk} + m_c \Delta r_{lk(n)} \\ v_{kj(n+1)} = v_{kj(n)} + \Delta v_{kj} + m_c \Delta v_{kj(n)} \\ w_{ji(n+1)} = w_{ji(n)} + \Delta w_{ji} + m_c \Delta w_{ji(n)} \end{cases} \quad (10)$$

上面各式中， $\eta$  表示学习率，其值通常在 0.01 ~ 1.0 之间，学习率  $\eta$  选得太小，会导致网络参数(耦合权值)修改量过小，收敛缓慢；选得太大，虽然可以加快学习速度，但可能致使权值修改量在稳定点附近持续震荡，难以收敛， $m_c$  为动量因子，一般取 0.9 左右。动量项的作用在于记忆前一时刻联接权值的变化方向(即变化量的值)，增加动量项，利用其“惯性效应”来抑制可能产生的震荡，起到平滑作用<sup>[5]</sup>，这样可以采用较大的学习率  $\eta$ ，以提高学习速度。

### 2.3 计算步骤

双隐层 BP 网络反传学习算法的实现分为 5 步：

(1) 初始化。设置网络各联接权值(包括阈值)，置所有可调参数(权和阈值)为均匀分布的较小随机数，使每个神经元的净输入值较小，以保证工作在激活函数变化较灵敏的区域；

(2) 信息正向传递过程的计算。输入学习样本对  $(x^p, t^p)$ ，对每个输入样本计算各神经元的输出值；

(3) 误差反向传播。从输出层至输入层，反向依次计算各神经元的等效误差，然后

返回步骤(2),对其他学习样本对进行正向传播计算和误差反向传播,一直到所有 $\alpha$ 个学习样本对都进行类似的运算为止;

(4) 调整各层的联接权值。按照权值调整公式,修改各层的联接权值;

(5) 返回步骤(2),根据新的联接权值,进行正向计算。若对每一个学习样本对 $(x^p, t^p)$ 和输出层的每一个神经元 $l$ 均满足精度要求,即:

$$\|t_l^p - y_l^p\|^2 < \varepsilon \quad p=1, 2, \dots, \alpha; \quad l=1, 2, \dots, m \quad (11)$$

其中 $\varepsilon$ 为预先给定的误差,若 $\|t_l^p - y_l^p\|^2 < \varepsilon$ 得到满足,则网络的学习完成,否则重新进行新一轮的正向传递和误差的反向传播计算,并调整各层的联接权,直到条件(11)得到满足为止<sup>[5]</sup>。

### 3 样本选取和网络训练

#### 3.1 样本选取和样本数据预处理

利用国家“十五”863 重点项目“赤潮重点监控区监控预警系统”项目组于 2006 年夏季在天津驴驹河赤潮监控区的连续监测资料作为样本数据,对上面建立的 BP 神经网络模型进行训练。天津驴驹河赤潮监控区监测面积为 300km<sup>2</sup>,监控站点数为 6 个(见图 2),项目执行期间项目组从 2006 年 5 月 22 日~8 月 28 日进行了连续 99 天的赤潮环境条件监测。监测要素包括表层水温、水色、透明度、海况、风速、风向、简易天气现象、气温、气压、化学需氧量、硝酸盐浓度、亚硝酸盐浓度、铵盐浓度、磷酸盐浓度、硅酸盐浓度、悬浮物浓度、盐度、PH 值、浊度、溶解氧浓度、叶绿素-a 密度、海洋浮游生物种类及密度等。监测频率为 1、2、4 号站每天一次,3、5、6 号站每三天一次。除去由于海况恶劣等原因造成监测中断,各站位获得的数据量如下:1 号站 72 条记录,2 号站 70 条记录,3 号站 26 条记录,4 号站 73 条记录,5 号站 25 条记录,6 号站 25 条记录。

一般来说,叶绿素-a 含量升高,意味着浮游植物数量增大,故通过对叶绿素-a 密度的预测可以间接对由浮游植物引发的赤潮进行预测。根据对监控区浮游植物生长影响的

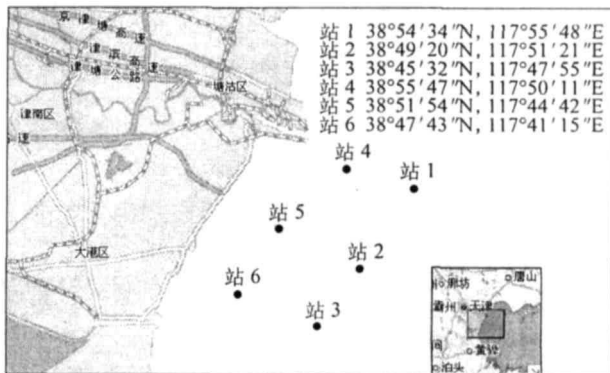


图 2 天津驴驹河赤潮监控区站位分布图

大小,选取监控区 8 个监测量作为上述人工神经网络模型的输入变量,将叶绿素-a 密度( $\mu\text{g/L}$ )作为预测量。选取的 8 个输入变量分别为表层水温( )、PH 值、盐度(ppt)、溶解氧浓度( $\text{mg/L}$ )、磷酸盐浓度( $\text{ug/L}$ )、亚硝酸盐-氮浓度( $\mu\text{g/L}$ )、硝酸盐-氮浓度( $\mu\text{g/L}$ )和氨-氮浓度( $\mu\text{g/L}$ )。一共选取了 48 个样本(见表 1)。

表 1 人工神经网络模型输入变量和叶绿素-a 样本

样本	表层水温	pH 值	盐度 ppt	溶解氧 mg/L	磷酸盐 ug/L	亚硝酸盐 ug/L	硝酸盐 ug/L	氨盐 ug/L	叶绿素 ug/L
1	17.4	8.07	33.15	7.74	44.77	26.1	261.2	34.5	10.84
2	18.8	8.02	33.01	7.23	16.54	52.2	430	72.75	5.99
3	21.6	8.14	32.65	7.6	37.9	11.8	233	16.4	7.08
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
46	26.9	8.03	30.60	6.58	6.34	95.4	347.7	92.3	8.21
47	27.3	8.08	30.59	6.49	6.81	54.3	321	27.4	12.2
48	27.7	8.13	30.10	6.79	6.34	73.7	337	21.274	18.08

如果变量的取值变化范围很大,经过网络学习后,联接权的幅值变化理论上也会很大,但是实际网络学习很难做到这一点,为此必须一开始就使各变量的重要性处于同等地位。从原理上说网络通过调整各权值的大小可以学习到变量的相对重要性,因此我们可以通过把输入数据归一化到能使网络所有权值都在一个不太大的范围之内,以此来减轻网络训练的难度。另一方面,S 型对数激活函数输出值在 0~1 之间,为了避免数据溢出,也需要将数据进行标准化、归一化处理<sup>[6]</sup>。数据行标准化的方法为:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad \sigma^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n} \quad y_i = \frac{x_i - \bar{x}}{\sigma} \quad (12)$$

其中  $x_i$  为原始样本值,  $i=1, 2, \dots, n$ ,  $n$  为样本数目,  $\bar{x}$  为样本平均值,  $\sigma$  为样本标准偏差,  $y_i$  为标准化后的样本。对数据进行标准化后,需对标准化后的数据进行归一化,归一化方法为:

$$z_i = 0.8 \times \frac{y_i - y_{\min}}{y_{\max} - y_{\min}} + 0.1 \quad (13)$$

其中  $y_{\min}$ 、 $y_{\max}$  分别为  $y_i$  的最小值和最大值,  $z_i$  为归一化后的变量值。

通过上述处理,就将原始数据标准化归一化到 0.1~0.9 的区间内,归一化后的数据(见表 2)。为了得到理想的训练集,接下来我们需要把标准归一化后的样本进行类间交叉处理。这要求训练时不同类别的样本要交叉输入,相同的样本在训练顺序上要尽量离得远些。若同类样本都聚集在一起,则某一类样本对权值集中修改的结果会破坏权值对前一类样本的映射关系,使 BP 网络对前一类样本的映射能力“前功尽弃”。这样反复修改的结果会导致震荡或延长训练时间<sup>[6]</sup>。通过打乱样本序列,得到类间交叉效果,可以得到更好的收敛速度。

表 2 标准化归一化后的数据

样本	表层水温	pH 值	盐度	溶解氧	磷酸盐	亚硝酸盐	硝酸盐	氨盐	叶绿素-a
1	0.375	0.863	0.799	0.753	0.775	0.120	0.251	0.144	0.341
2	0.873	0.536	0.316	0.312	0.263	0.489	0.900	0.156	0.147
3	0.388	0.790	0.802	0.398	0.233	0.145	0.100	0.178	0.339
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
46	0.696	0.354	0.710	0.436	0.189	0.336	0.246	0.152	0.309
47	0.755	0.463	0.517	0.560	0.601	0.900	0.518	0.134	0.155
48	0.880	0.681	0.307	0.494	0.268	0.479	0.685	0.181	0.425

3.2 网络训练结果及预测

通过对网络隐含层节点数以及网络参数反复试验 ,最后确定BP网络结构为 8-15-15-1 (即输入节点数为 8 个 ,两个隐含层节点数各为 15 个 ,输出层节点数为 1 个) ,网络学习率  $\eta=0.15$  ,动量系数  $m_c=0.8$  ,网络拟合最大误差预设值  $E_{\Sigma}$  为  $3.0\times10^{-4}$  ,网络初始联接权值赋予随机小量。

利用上面确定的网络结构对样本数据进行交叉训练 ,每次预留 5 个样本作为测试集 ,其余用作拟合集循环试验 ,选取预测误差最小的一个拟合训练集和预测集组合。网络训练完成后 ,即确定了网络各节点之间的耦合权值 ,这样便建立了表层水温、pH 值、盐度、溶解氧、磷酸盐、亚硝酸盐、硝酸盐、氨盐 8 个理化因子与叶绿素-a 密度之间的非线性人工神经网络模型。网络训练结束后的各层节之间耦合权值(见表 3、4、5)。利用训练好的网络模型对剩余的 5 个样本进行预测 ,拟合值和预测值与实际值的对比如图 3 所示 ,从图中可见 ,该人工神经网络能够较好地拟合并预测监控区叶绿素密度的变化情况。

表 3 输入层到隐层 1 的权值矩阵 W

节点 j	$W_{j0}$	$W_{j1}$	$W_{j2}$	$W_{j3}$	$W_{j4}$	$W_{j5}$	$W_{j6}$	$W_{j7}$	$W_{j8}$
1	0.815	-2.842	-1.461	1.578	4.032	-5.647	1.006	-0.138	0.957
2	2.671	0.900	1.034	-1.220	1.258	2.877	6.139	-1.884	4.918
3	0.459	-0.679	-0.671	-0.481	0.226	0.306	0.056	-0.256	0.384
4	0.411	-0.464	-0.851	-0.736	-0.216	0.260	-0.078	-0.229	-0.223
5	-0.130	2.304	-2.509	0.805	-9.292	4.839	3.469	3.571	-4.684
6	-0.671	0.271	-2.424	-2.046	1.108	0.468	-0.597	-0.902	0.895
7	1.128	-0.533	-0.090	2.087	1.089	-6.488	0.128	-2.545	0.822
8	-0.089	-0.015	-1.667	-1.468	0.564	0.433	-0.420	-0.501	-0.675
9	0.505	-3.479	4.618	1.079	-0.473	0.252	0.300	-0.604	2.435
10	2.253	0562	-1.178	5.862	-8.833	3.607	8.566	2.553	0.792
11	0.805	-0.754	-0.103	0.102	0.788	-0.776	-0.241	0.056	1.342
12	0.003	-0.356	-1.203	-1.335	0.289	0.735	-0.090	-0.450	-0.384
13	0.547	-0.599	-0.642	-0.624	-0.112	0.335	0.063	-0.220	-0.036
14	-1.293	2.781	2.324	-6.538	0.289	-1.935	-2.432	-1.819	-0.350
15	0.051	-0.201	-1.278	-1.191	-0.108	0.334	-0.474	-0.229	-1.024

表 4 隐层 1 到隐层 2 的权值矩阵 V

节点 k	$V_{k0}$	$V_{k1}$	$V_{k2}$	$V_{k3}$	$V_{k4}$	$V_{k5}$	$V_{k6}$	$V_{k7}$
1	0.395	-1.742	-1.432	0.021	0.269	-2.112	0.528	-1.663
2	1.115	-0.501	-0.335	-0.202	-0.180	-0.289	-0.327	-0.343
3	-1.123	0.546	-1.987	0.694	0.890	-6.110	2.270	-0.371
4	0.562	-1.960	-1.596	-0.051	0.138	-1.573	0.300	-1.771
5	0.699	0.622	-0.478	-0.310	-0.241	-0.388	-0.392	0.874
6	0.730	-2.135	-1.654	0.006	0.036	-1.398	0.249	-1.852
7	0.920	0.289	-0.344	-0.286	-0.258	-0.135	-0.422	0.473
8	2.060	-0.173	1.924	-1.112	-0.608	2.699	-1.199	-0.129
9	0.789	-1.491	-0.842	-0.057	0.02	-1.146	0.082	-1.404
10	0.264	-2.063	-2.309	-0.077	0.203	-1.795	0.614	-1.847
11	1.016	-1.888	-0.910	0.010	-0.070	-0.734	0.032	-1.619
12	0.886	0.718	-0.374	-0.338	-0.345	0.579	-0.509	0.979
13	0.254	-1.691	-1.831	0.013	0.281	-1.614	0.747	-1.763
14	0.848	-1.769	-0.931	-0.067	-0.019	-1.090	0.131	-1.514
15	-0.935	-1.371	-0.616	-0.091	-0.099	-0.761	0.001	-1.271

节点 k	$V_{k8}$	$V_{k9}$	$V_{k10}$	$V_{k11}$	$V_{k12}$	$V_{k13}$	$V_{k14}$	$V_{k15}$
1	0.473	-1.955	0.214	-0.670	0.350	0.161	0.309	0.495
2	-0.204	-0.292	-0.132	-0.307	-0.242	-0.191	-0.253	-0.331
3	1.660	-2.467	-8.212	0.416	1.202	0.650	6.062	1.335
4	0.257	-0.766	1.775	-0.399	0.171	0.061	-0.035	0.286
5	-0.359	-0.159	-0.972	-0.191	-0.434	-0.225	-0.027	-0.341
6	0.122	-0.184	2.121	-0.353	0.208	0.121	-0.223	0.140
7	-0.354	-0.410	-0.693	-0.238	-0.328	-0.254	-0.144	-0.374
8	-0.963	-4.096	0.063	-1.285	-1.143	-0.765	-0.417	-0.642
9	0.005	-0.529	0.870	-0.349	-0.011	0.000	-0.155	0.046
10	0.397	-1.276	2.707	-0.417	0.367	0.112	0.189	0.468
11	0.007	0.330	1.207	-0.246	0.048	0.036	-0.404	-0.123
12	-0.475	-0.621	-0.816	-0.396	-0.567	-0.271	-0.108	-0.391
13	0.549	-1.699	1.528	-0.534	0.419	0.226	0.281	0.614
14	0.071	-0.203	1.159	-0.298	0.038	0.021	-0.266	-0.015
15	-0.050	-0.196	0.669	-0.276	0.008	-0.101	-0.284	-0.078

表 5 隐层 2 到输出层的权值矩阵 R

节点 m	$R_{m0}$	$R_{m1}$	$R_{m2}$	$R_{m3}$	$R_{m4}$	$R_{m5}$	$R_{m6}$	$R_{m7}$
1	1.954	2.184	0.075	4.133	1.821	-1.208	1.808	-0.792

节点 m	$R_{m8}$	$R_{m9}$	$R_{m10}$	$R_{m11}$	$R_{m12}$	$R_{m13}$	$R_{m14}$	$R_{m15}$
1	-3.688	1.180	2.388	1.258	-1.146	2.211	1.275	0.939

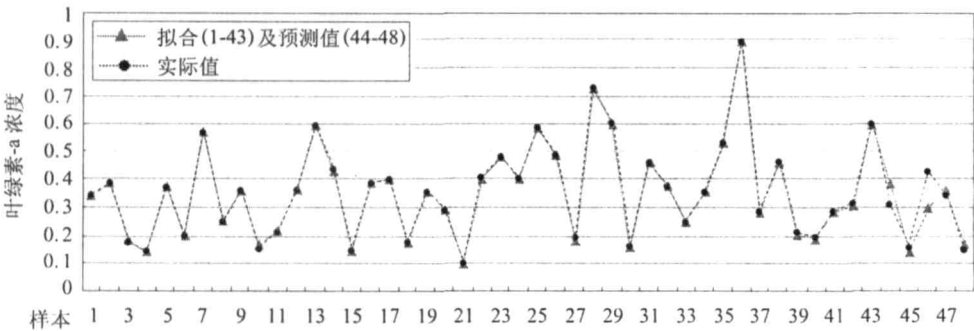


图 3 BP 网络的拟合值和预测值与实际值的对比曲线



## 4 讨论

本文利用误差反向传播(BP)人工神经网络模型建立了天津驴驹河赤潮监控区主要理化因子与叶绿素-a之间的映射关系。模拟结果表明:

(1) 该模型能够较好的反映各种理化因子与叶绿素-a浓度之间的非线性对应关系,具有较好的赤潮预测能力,这说明利用人工神经网络对目前发生机理尚不明晰的赤潮进行预测是可行的。

(2) 人工神经网络虽然能够对大量的输入输出数据进行精确的拟和运算,但是它有一些不利的因素,特别是当人们希望根据训练后的网络各节点联接权值的大小来分析网络根据哪些因素做出决策时,会发现实际上网络权值的解释性很差,一般很难通过对权值的分析得到各要素的重要性,因此要想对赤潮做深入研究仍离不开对其形成机理的剖析。

(3) 目前对于BP网络结构设计(如隐含层节点数的确定)、网络参数的选取(如学习率、动量系数、初始网络权值、要求拟合精度等)还没有统一的标准,因此这种人为因素对网络的训练结果带来很大的影响。此外为了提高网络的泛化能力,对训练集的设计也显得极为重要(如变量的选取以及类间均衡、样本分离及交叉等),因此必须通过大量的试验获取满意的结构和参数,这种网络模型难免带有缺陷。

(4) 渤海赤潮生物主要是浮游植物,根据2001~2006年监测数据统计,由浮游植物引发的渤海赤潮占到赤潮爆发总面积的93.4%,总次数的83.2%,而叶绿素-a密度与浮游植物密度的变化趋势吻合度是一致的,因此通过对叶绿素-a密度的预测可以间接地对渤海湾由浮游植物引发的赤潮进行预测。但是如果赤潮是由原生动植物引起的,这些赤潮生物会摄食浮游植物而降低叶绿素密度,这时需要结合其他预报方法对其进行预测<sup>[7]</sup>。

(5) 基于人工神经网络的赤潮预报方法作为一种新技术,有着很大的开发潜力与广泛的应用前景。随着人工神经网络方法的不断发展,结合对赤潮形成机理研究的不断深入,该预报模型还将不断完善。

### 参考文献:

- [1] 蒋宗礼. 人工神经网络导论[M]. 北京: 高等教育出版社, 2001, 8: 10 ~ 13.
- [2] 吴京洪, 杨秀环, 唐宝英, 李锦蓉, 张展霞. 大亚湾澳头养殖区赤潮与环境的关系研究 I. 浮游植物总生物量与环境因子的关系[J]. 中山大学学报(自然科学版), 2001, 40(3): 37 ~ 40.
- [3] 蔡如钰. 赤潮生物密度与环境因子的 ANN 网络映射预测模型[J]. 福建环境, 2001, 18(3): 30 ~ 32.
- [4] 王洪礼, 葛根, 李悦雷. 基于模糊神经网络(FNN)的赤潮预警预测研究[J]. 海洋通报, 2006, 25(4): 36 ~ 41.
- [5] 陈祥光. 人工神经网络技术及应用[M]. 北京: 中国电力出版社, 2003, 9: 22 ~ 31.
- [6] 王文成. 神经网络及其在汽车工程中的应用[M]. 北京: 北京理工大学出版社, 1998, 7: 232 ~ 268.
- [7] 黄良民, 钱宏林, 李锦蓉. 大鹏湾赤潮多发区的叶绿素-a分布与环境关系初探[J]. 海洋与湖沼, 1994, 25(2): 197 ~ 204.

## The Application of BP Neural Network in the Prediction of Chlorophyll-a Concentration in the Bohai Gulf

HE En-ye , LI Hai , REN Xiang-xiang , CAI Yi , YANG Jing

(1. National Marine Environmental Forecast Center , Beijing 100081 China)

**Abstract :** By using BP Neural Network method, a nonlinear relationship between various physical-chemical factors and the chlorophyll-a concentration in the Bohai Gulf is established. The simulated results show that the predicted chlorophyll-a concentration fits the observed data well, indicating that this model can be indirectly used for local red tide prediction.

**Key words :** BP neural network ; chlorophyll-a concentration ; red tide