

基于人工神经网络的粮食产量预测模型

禹建丽¹, 黎 娅²

(1. 中原工学院数理系, 河南 郑州 450007; 2. 河南经贸职业学院, 河南 郑州 450053)

摘要: 研究了人工神经网络在经济预测中的应用问题, 探讨利用人工神经网络进行农业粮食产量预测的方法。提出一种基于多层前馈 BP 神经网络的农业粮食产量预测模型, 可以得到影响粮食产量的主要因子和粮食产量之间的非线性映射关系。并通过实例验证了神经网络模型的预测精度明显高于线性回归模型的预测精度。

关键词: 粮食产量; 人工神经网络; 预测; 模型

中图分类号: F326 **文献标识码:** A **文章编号:** 1004-3268(2005)07-0044-03

人工神经网络是一个大规模自组织、自适应的非线性动力系统。它能较好的模拟人的形象思维, 具有大规模并行协同处理能力, 具有较强的容错能力、联想能力和学习能力, 能根据外界环境的变化修

改自己的行为, 并且能依据一定的学习算法自主地从训练实例中学习^[1]。BP 模型是一种用于前向多层神经网络的反传学习算法, 由鲁梅尔哈特和麦克莱伦德于 1985 年提出。目前, BP 算法已经成为应

收稿日期: 2004-12-01

作者简介: 禹建丽(1960—), 女, 河南洛阳人, 教授, 博士, 主要从事系统科学、计算智能、非线性预测等教学与研究工作。

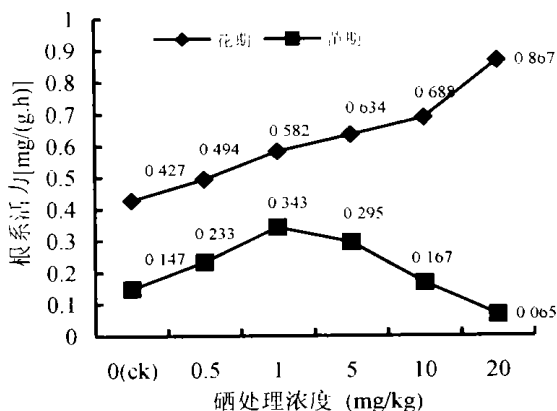


图2 硒在不同时期对大豆根系活力的影响

3 讨论

3.1 硒对大豆根系发育的促进作用

从上述结果可以看出, 硒可以促进植物的根系发育。不管是在砂壤, 还是在粘壤中, 适宜浓度的硒都能增强植株的根系还原能力, 使其表现出旺盛的生命活力。不仅如此, 适宜的硒浓度对于植株的生长发育是明显有利的, 能增加干物质积累和可溶性蛋白质、叶绿素的含量以及硒含量, 提高千粒重和产量(将作另文讨论)。因此, 可将硒用作微肥在广大

缺硒和低硒地区推广, 一方面可以提高作物产量、改善品质, 另一方面可以以此来改善农牧业生产中的硒生态环境, 提高人畜硒营养水平。

3.2 硒对植物的毒性及不同土质补硒的差异性

虽然硒有上述诸多积极作用, 但是也可以看到, 砂壤中栽培的大豆, 在硒浓度达到 20mg/kg 时, 根系活力显著降低, 而且在试验过程中观察到, 豆苗有萎蔫甚至死亡的现象, 即表现出一定的中毒症状。而在本试验设计的硒浓度范围内, 粘壤栽培大豆未表现出明显不良现象, 这说明粘壤部分阻碍了植物根系对硒的吸收, 有效利用度不如砂壤中的高。因此, 作物的补硒栽培应该根据不同的土壤情况来确定适宜的浓度范围, 因地制宜。

参考文献:

- [1] 吴军, 刘秀芳, 徐汉生. 硒在植物生命活动中的作用[J]. 植物生理学通讯, 1999, 35(5): 417-423.
- [2] 尚庆茂, 李平兰. 硒在高等植物中的生理作用[J]. 植物生理学通讯, 1998, 34(4): 284-287.
- [3] 李合生. 植物生理生化实验原理和技术[M]. 北京: 高等教育出版社, 2000. 119-120.

用最多且最重要的一种训练前向神经网络的学习算法,亦是前向网络得以广泛应用的基础,它是一种由教师示教的网络,通过对有代表性例子的学习和训练,能掌握事物的本质特征。利用神经网络建立预测模型可以有效地提高预测精度^[4]。我们以粮食播种面积、成灾面积、单位面积化肥用量为输入神经元,以全国粮食总产量为输出神经元,建立 BP 神经网络结构,根据历史数据进行网络学习,从而实现对未来全国粮食总产量的预测。

1 建立预测模型的方法

1.1 神经网络结构

采取具有多输入节点和单输出节点的 3 层前向神经网络结构,含 1 层隐层。当有信息向网络输入时,信息先由输入层传至隐层节点,再传至输出层。其间每经过一层都要由相应的特性函数进行变换。节点的特性函数选用:

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

1.2 BP 学习算法

BP 算法的学习过程是由正向传播与反向传播组成的。正向传播用于网络计算,对某一输入求出它的输出;反向传播用于逐层传递误差,修改连接权值和阈值。

BP 算法的步骤如下:

(1)设置变量和参量:

$X_k = [x_{k1}, x_{k2}, \dots, x_{kM}]$ ($k = 1, 2, \dots, N$), x_k 为输入向量,即训练样本, N 为训练样本的个数。

$W_{MI}(n) = (w_{ij})_{M \times I}$, 为第 n 次迭代时输入层与隐层 I 之间的权值向量。

$W_{IJ}(n) = (w_{ij})_{I \times J}$, 为第 n 次迭代时隐层 I 与隐层 J 之间的权值向量。

$W_{JP}(n) = (w_{ij})_{J \times P}$, 为第 n 次迭代时隐层 J 与输出层之间的权值向量。

$Y_k(n) = [y_{k1}(n), y_{k2}(n), \dots, y_{kP}(n)]$ ($k = 1, 2, \dots, N$), $Y_k(n)$ 为第 n 次迭代时网络的实际输出。

$d_k(n) = [d_{k1}(n), d_{k2}(n), \dots, d_{kP}(n)]$ ($k = 1, 2, \dots, N$), $d_k(n)$ 为期望输出。

(2)初始化。赋较小的随机非零值于 $W_{MI}(0)$, $W_{IJ}(0)$, $W_{JP}(0)$

(3)输入样本 X_k , $n = 0$

(4)对输入样本 X_k , 前向计算 BP 网络每层神经元的输入信号 u 和输出信号 v 。

(5)由期望输出 d_k 和上一步求得的实际输出 $Y_k(n)$ 计算误差 $E(n)$, 判断其是否满足要求, 若满足转至(8); 不满足转至(6)。

(6)判断 $n + 1$ 是否大于最大迭代次数, 若大于转至(8), 否则, 对输入样本 X_k , 反向计算每层神经元的局部梯度 δ 其中,

$$\delta_p = y_p(n) \left[1 - y_p(n) \right] \left[d_p(n) - y_p(n) \right],$$
$$(p = 1, 2, \dots, P)$$

$$\delta_j = f' \left(u_j^J(n) \right) \sum_{p=1}^P \delta_p w_{jp}(n) \quad (j = 1, 2, \dots, J)$$

$$\delta_i = f' \left(u_i^I(n) \right) \sum_{j=1}^J \delta_j w_{ij}(n), \quad (i = 1, 2, \dots, I)$$

(7)按下式计算权值修正 ΔW , 并修正权值, η 为学习速率。 $n = n + 1$, 转至(4)。

$$\Delta W_{jp}(n) = \eta \delta_p(n) v_j^J(n),$$

$$W_{jp}(n+1) = W_{jp}(n) + \Delta W_{jp}(n),$$

$$(j = 1, 2, \dots, J; p = 1, 2, \dots, P);$$

$$\Delta W_{ij}(n) = \eta \delta_j(n) v_i^I(n),$$

$$W_{ij}(n+1) = W_{ij}(n) + \Delta W_{ij}(n),$$

$$(i = 1, 2, \dots, I; j = 1, 2, \dots, J);$$

$$\Delta W_{mi}(n) = \eta \delta_i(n) x_{km}(n)$$

$$W_{mi}(n+1) = W_{mi}(n) + \Delta W_{mi}(n),$$

$$(m = 1, 2, \dots, M; i = 1, 2, \dots, I);$$

(8)判断是否学完所有的训练样本, 是则结束, 否则, 返回(3)。

2 神经网络预测模拟试验

以我国粮食播种面积 x_1 、成灾面积 x_2 、单位面积化肥用量 x_3 为输入神经元, 以我国粮食总产量 y 为输出神经元, 利用上面给出的方法建立 BP 神经网络模型, 根据 1978 年到 1995 年的历史数据^[3] 进行神经网络训练和预测模拟试验。

2.1 数据的标准化

先对数据进行标准化处理。即同一因子最小值定为 0, 最大值定为 1, 中间值用下列公式计算:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

标准化后的数值如表 1。

2.2 网络训练

以 x_1 、 x_2 、 x_3 作为网络输入层的 3 个神经元, y 作为网络输出层的 1 个神经元, 设定 1 个隐层, 隐层有 3 个神经元, 构造 1 个 3—3—1 的网络结构。取 1978 ~ 1992 年的数据作为学习样本集, 逐年进行学习。利用训练后的网络, 进行实际预测。神经网络预

测结果及线性回归预测结果列于表 2。

表 1 1978~1995 年粮食总产与诸因素标准化数值

年份	x_1	x_2	x_3	y
1978	1	0.395077	0	0
1979	0.970922	0	0.0448874	0.13689716
1980	0.926405	0.461057	0.0875303	0.07908502
1981	0.876429	0.248226	0.1040205	0.10135896
1982	0.843604	0.086727	0.1436127	0.24901769
1983	0.856438	0.097777	0.177332	0.41319418
1984	0.830902	0.036191	0.1942069	0.5133018
1985	0.742248	0.512388	0.2036639	0.37220012
1986	0.788067	0.566591	0.2358712	0.43426684
1987	0.795428	0.353477	0.2486794	0.500430797
1988	0.770287	0.577333	0.2797311	0.47325875
1989	0.815983	0.591713	0.3206581	0.54894011
1990	0.843674	0.166389	0.3639346	0.7362146
1991	0.818381	0.765411	0.4046089	0.68663814
1992	0.779881	0.655297	0.433309	0.69024201
1993	0.778761	0.498175	0.4858105	0.75946643
1994	0.75758	1	0.5187702	0.70250519
1995	0.768935	0.425495	0.5689955	0.81021102

表 2 神经网络训练后回检结果

年份	粮食总产 (万 t)	神经网络 预测值 (万 t)	相对 误差 (%)	线性回归 预测值 (万 t)	相对 误差 (%)
1978	30 475	30 897	1.38	29 826	-2.13
1979	33 210	32 392	-2.46	33 392	0.55
1980	32 055	31 929	-0.39	32 140	0.27
1981	32 500	33 370	2.68	34 387	5.81
1982	35 450	36 190	2.09	36 588	3.21
1983	38 730	38 470	-0.67	37 747	-2.54
1984	40 730	40 107	-1.53	39 143	-3.90
1985	37 911	37 131	-2.06	37 464	-1.18
1986	39 151	38 803	-0.89	38 355	-2.03
1987	40 473	40 972	1.23	40 547	0.18
1988	39 930	40 711	1.96	40 496	1.42
1989	41 442	42 032	1.42	41 703	0.63
1990	45 184	45 164	-0.4	45 405	0.49
1991	44 193	43 298	-2.03	43 354	-1.90
1992	44 265	44 412	0.33	45 152	2.00

2.3 预测结果与分析

用上述经过训练的神经网络对 1993~1995 年的粮食总产进行预测,其预测结果与线性回归预测值进行比较,列于表 3。

表 3 1993~1995 年粮食总产预测比较

年份	粮食总产 (万 t)	神经网络 预测值 (万 t)	相对 误差 (%)	线性回归 预测值 (万 t)	相对 误差 (%)
1993	45 648	45 963	0.69	47 458	3.97
1994	44 510	44 629	0.27	45 868	3.05
1995	46 662	47 314	1.40	50 485	8.19

由表 3 可以看出,经过训练的人工神经网络的预测结果具有较高的精确度,1993~1995 年预测值精度分别达到 99.31%,99.73%,98.6%。而相应的线性回归预测值精度分别为 96.03%,96.95%和 91.81%。将线性回归和神经网络预测结果相比较可以看出,神经网络预测模型优于线性回归预测模型。

3 结论

人工神经网络 BP 模型能自主地从训练实例中学习,建立自变量与因变量之间的非线性映射关系。预测指标的选择对神经网络学习和预测有较大的影响,因此,在应用神经网络模型进行预测时,应结合实际对预测指标进行适当的取舍,力求所建立的神经网络预测模型有较高的预测精度。本研究中,选取粮食播种面积、成灾面积、单位面积化肥用量 3 个预测指标,在神经网络训练后,得到了令人满意的回检精度和预测精度,表明 BP 神经网络模型可以应用于农业生产预测。

参考文献:

[1] 阎平凡,张长水. 人工神经网络与模拟进化计算[M]. 北京:清华大学出版社,2000. 17-21.
[2] 刘永中. 冷冻干燥过程的神经网络预测模型[J]. 农业机械工程[J]. 2002 33(4): 129-137.
[3] 国家统计局. 中国统计年鉴[M]. 北京:中国统计出版社,1997.