

人工神经网络与故障诊断

宋京伟 佐山隼敏 铃木和彦

(机械工程系) (日本冈山大学工学部)

摘 要

本文介绍了人工神经网络的基本概念、模型与算法,并介绍了人工神经网络在故障诊断领域中的轴承故障诊断和化工流程异常诊断应用例子。

关键词: 人工神经网络; BP 算法; 故障诊断

0 引 言

1943年,由心理学家 W. McCulloch 和数理逻辑学家 W. Pitts 首次提出了模仿人类大脑中的神经元与连接方式以构成并行算术和逻辑运算的人工神经网络模型(ANN),简称 M—P 模型。在此之后,各国的专家、学者相继对这一领域开展了各种研究,并取得了许多突破与成果。但在 60 年代,神经网络的研究基本上停留在单层结构的水平。

1982年,物理学家 Hopfield 提出了 HNN 模型,从而有力地推动了神经网络的研究。近几年来,神经网络理论引起了美国、欧洲与日本等各国科学家和企业家的巨大热情。在机械故障诊断领域,对于智能诊断问题,神经网络所具有的特性有很大吸引力,为此我们探讨了故障诊断领域的有关应用问题。

1 人工神经网络

1.1 人工神经网络的特性

人的思维是通过人脑来完成的,神经元是组成人脑的基本单元。神经科学表明,中枢神经的主要部份——大脑皮层是由大量(约 $10^{11} \sim 10^{12}$ 个)神经元组成。神经元的生理结构可简化成如图 1 所示,其中心有一实体团,称之为细胞体,它能对接收到的信息进行处理,细胞体周围的纤维分为两类,一类是接收信息的称为树突,另一类是发出信息的,称之为轴突。树突与轴突——对接,称之为突触,从而把众多的神经元连成一个神经网络。

人工神经网络^[1]是试图从结构上模拟人的生物神经来处理信息的,其基本成份也称之为神经元(有时也称之为计算单元或节点)。一个神经元的特性可以用图 2 表示,它有 n 个输入

本文于 1994 年 3 月 23 日收到

$X_1, X_2, X_3, \dots, X_n$, 是该神经元接收到的信息; 一个输出 Y , 是该神经元的输出信息。

对于某一神经元, 设来自神经元 i 对它的输入为 $X_i (i = 1, 2, \dots, n)$, 它的输出为 Y , 神经元 i 和该神经元之间的连接权为 W_i , 该神经元阈值为 θ , 则

$$Y = f\left(\sum_{i=1}^N W_i \cdot X_i - \theta\right)$$

(1)

f 为该神经元的输入输出函数, 神经元的非线性表现在它的输入输出函数 f 上。常用的非线性输入输出函数有 3 种:

(1) 继电型 (hard limiter) (参见图 3a)

$$f(x) = \begin{cases} 1 & (x > 0), \\ 0 & (x \leq 0). \end{cases} \quad (2)$$

(2) 阈值逻辑型 (threshold logic) (参见图 3b)

$$f(x) = \begin{cases} 0, & (x \leq 0), \\ ax, & (0 < x \leq 1/a), \\ 1, & (1/a \leq x). \end{cases} \quad (3)$$

(3) S 型 (sigmoid) (参见图 3c)

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (4)$$

其它还有更复杂的结构。

1.2 神经网络特点

当众多神经元连成一个网络并动态运行时, 则构成一个高度非线性的动力学系统, 作为一个非线性动力学系统, 人工神经网络除具有一般非线性动力学系统的一切特征外, 又有许多自身的特点:

(1) 信息传递的并行性 在人工神经网络系统中, 信息的存贮与处理是合而为一的, 即信息的存贮体现在神经元互连的分布上, 并进行大规模并行分布处理;

(2) 容错性 任何局部的损坏不会影响整体结果, 还能有效地支持不确定或不完全知识

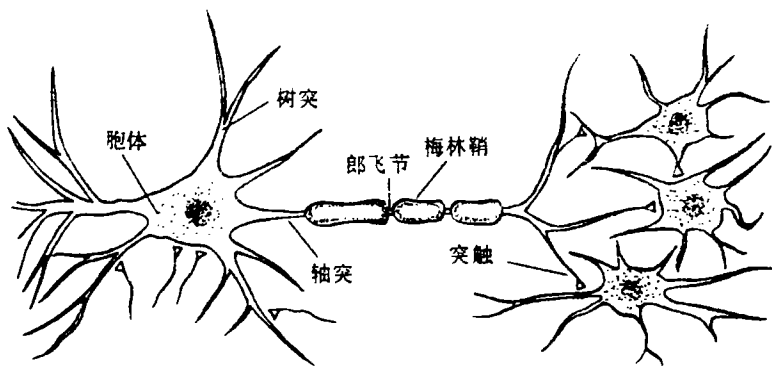


图 1 神经元的生理结构

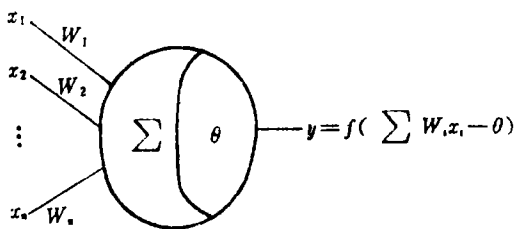


图 2 神经元的模型

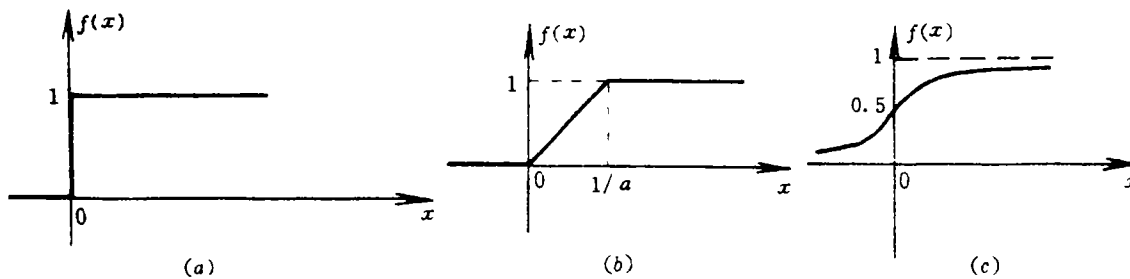


图3 非线性输入输出函数

的处理;

(3) 自学习性自组织性 神经网络能很好地进行知识的获取,它可通过内部的自组织能力,从外界环境中获取信息,使自己更为有效地适应环境以完成指定的各种计算与推理工作,具有创新特点,这不同于 AI 中的专家系统。

2 神经网络的常用模型与 BP 学习算法

人们根据不同的需要,提出了许多神经网络模型和相应的算法。这里只介绍几种最常用的模型和应用最广的 BP 学习算法。

2.1 M—P 模型与 Hebb 学习规则

神经网络最早的模型是 M—P 模型。在 M—P 模型中,神经元的性质是简单的,而网络的计算能力却是巨大的。设有 N 个神经元相互连接,每个神经元的状态为 S_i ($i=1, 2, \dots, N$),只能取 1 或 0,分别代表兴奋与抑制,状态间的关系为:

$$S_i = f_i \left[\sum_{j=1}^N W_{ij} + \theta_i \right] \quad (5)$$

式中 W_{ij} 是神经元 j 到神经元 i 的连接权; θ_i 是神经元的阈值。学习过程就是调节 W_{ij} 和 θ_i 的过程。

Hebb 提出了突触联结强度可变的假设:学习过程最终发生在神经元之间的突触部处,突触的联结强度随突触前后神经元的活动而变化。此假设称为 Hebb 学习,它为神经网络的学习算法奠定了基础。Hebb 学习规则即调节 W_{ij} 的原则: $\Delta W_{ij} = \alpha S_i S_j$ ($\alpha > 0$),式中 S_i, S_j 是神经元 i, j 的输出,这一规则已得到神经细胞学说的证实。

2.2 多层神经元和 BP (反向传播) 学习算法

一般单层神经元模型和图 2 所示相同,其输入信号可以是连续信号,也可以是二值信号,输出是二值的。

由于单层神经元模型的局限性,它无法完成一些复杂的分类问题和逻辑运算,如异或问题也不能解决;双层模型虽然解决了异或问题,但并没有解决其他复杂的分类问题;而 3 层

或多层神经元模型克服了上述限制,多层网络模型参见图 4. 从理论上讲,对于给定的任意复杂的分类问题,只要适当选择隐节点的数目(主要是凭经验和试探),再加上适当的学习算法,就可以构造出所需要的非线性映射.

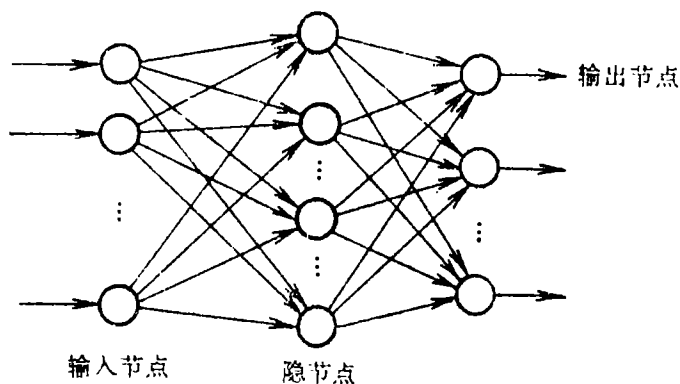


图 4 多层网络的结构

1985 年 Rumelhart 等人提出了多层网络的误差反向传播 (BP) 学习算法. BP 算法使用一种最速下降搜索技术,将误差信号从输出层向较低层传播而得名. 其主要思路是:如果求出误差 E 对多个神经网络元输出的偏导数,那么就可以算出 E 对所有连接权的偏导数,因而可用梯度下降法来修改各个连接权.

BP 算法的学习过程是,首先将输入样本从输入层经过加权由隐层通过输出输入函数处理输出给下一层,以此逐层处理传向输出层,得到一个输出. 如果在输出层不能得到期望的输出,则转入反向传播,将误差信号沿原来的连接通路返回,逐层修改神经元的连接权值,以减少输出误差,经若干次这样反复修改连接权值,使得误差在给定的允许范围,学习过程则结束.

这里具体就 BP 算法作一介绍. 设如图 4 所示 n 层网络,在样本 P 输入时,网络中第 K 层的各节点 u_j^k 的输出输入关系用下式表示

$$O_{pj}^k = f_j^k(i_{pj}^k) \quad (6)$$

$$i_{pj}^k = \sum_{i=1}^{N_{k-1}} W_{i,j}^{k-1,k} O_{pi}^{k-1} + \theta_j^k \quad (7)$$

式中

O_{pj}^k ——样本 P 输入时第 k 层的第 j 节点 u_j^k 的输出;

i_{pj}^k ——样本 P 输入时第 k 层的第 j 节点 u_j^k 的输入;

$W_{i,j}^{k-1,k}$ ——第 $k-1$ 层的第 i 节点 u_i^{k-1} 与第 k 层的第 j 节点 u_j^k 的连接权;

f_j^k ——第 k 层的第 j 节点 u_j^k 的输出输入函数;

θ_j^k ——对第 k 层的第 j 节点 u_j^k 输入的阈值;

N_k ——第 k 层的节点数.

为评价网络, 把网络之实际输出 O_{μ} 与期望输出 t 的二乘误差 E 作为评价函数.

$$E = \sum_p E_p, \quad (8)$$

$$E_p = \frac{1}{2} \sum_{i=1}^{N_p} (t_{\mu}^i - O_{\mu}^i)^2, \quad (9)$$

t_{μ}^i ——样本 P 输入时输出层(第 n 层)节点 i 的期望输出.

BP 算法为使 $E_p \rightarrow \min$, 是通过修改连接权来实现, 而且希望 E_p 是按照梯度下降的, 即

$$\Delta_p W_{i,j}^{k-1,k} \propto - \frac{\partial E_p}{\partial W_{i,j}^{k-1,k}}. \quad (10)$$

上式的右边可分解为

$$\frac{\partial E_p}{\partial W_{i,j}^{k-1,k}} = \frac{\partial E_p}{\partial i_{pj}^k} \cdot \frac{\partial i_{pj}^k}{\partial W_{i,j}^{k-1,k}}, \quad (11)$$

(11) 式右边的第一部份反映了节点 u_j^k 的输入 i_{pj}^k 的变化对 E_p 的影响, 第二部份反映了与节点 u_j^k 连接的连接权 $W_{i,j}^{k-1,k}$ 的变化对节点 u_j^k 输入 i_{pj}^k 的影响.

由式 (7) 可得

$$\frac{\partial i_{pj}^k}{\partial W_{i,j}^{k-1,k}} = \frac{\partial}{\partial W_{i,j}^{k-1,k}} \sum_{i=1}^{N_{k-1}+1} (W_{i,j}^{k-1,k} O_{\mu}^{k-1}) = O_{\mu}^{k-1}. \quad (12)$$

令

$$\delta_{pj}^k = \frac{\partial E_p}{\partial i_{pj}^k}, \quad (13)$$

所以

$$- \frac{\partial E_p}{\partial W_{i,j}^{k-1,k}} = \delta_{pj}^k O_{\mu}^{k-1} \quad (14)$$

$$\Delta_p W_{i,j}^{k-1,k} = \eta \delta_{pj}^k O_{\mu}^{k-1}, \quad (15)$$

其中 η 是学习率.

此外, 式(13) 还可以用节点 u_j^k 的输出 O_{pj}^k 来表示, 即

$$\delta_{pj}^k = - \frac{\partial E_p}{\partial i_{pj}^k} = - \frac{\partial E_p}{\partial O_{pj}^k} \cdot \frac{\partial O_{pj}^k}{\partial i_{pj}^k}. \quad (16)$$

由式 (6) 可得

$$\frac{\partial O_{pj}^k}{\partial i_{pj}^k} = \frac{\partial}{\partial i_{pj}^k} f_j^k(i_{pj}^k), \quad (17)$$

对输出节点:

$$\frac{\partial E_p}{\partial O_{pj}^k} = - (t_{pj}^k - O_{pj}^k), \quad (18)$$

$$\delta_{pj}^k = - (t_{pj}^k - O_{pj}^k) \frac{\partial}{\partial i_{pj}^k} f_j^k(i_{pj}^k). \quad (19)$$

对非输出节点:

$$\begin{aligned}
\frac{\partial E_p}{\partial O_{pj}^k} &= \sum_{i=1}^{N_{k+1}} \left(\frac{\partial E_p}{\partial i_{pi}^{k+1}} \cdot \frac{\partial i_{pi}^{k+1}}{\partial O_{pj}^k} \right) \\
&= \sum_{i=1}^{N_{k+1}} \left(\frac{\partial E_p}{\partial i_{pi}^{k+1}} \cdot \frac{\partial}{\partial O_{pj}^k} \sum_{i=1}^{N_{k+1}} (W_{i,j}^{k,k+1} O_{pi}^k) \right) \\
&= \sum_{i=1}^{N_{k+1}} \left(\frac{\partial E_p}{\partial i_{pi}^{k+1}} \cdot W_{i,j}^{k,k+1} \right) \\
&= - \sum_{i=1}^{N_{k+1}} (\delta_{pi}^{k+1} W_{i,j}^{k,k+1})
\end{aligned} \quad (20)$$

$$\delta_{pj}^k = \frac{\partial}{\partial i_{pj}^k} \cdot f_j'(i_{pj}^k) \sum_{i=1}^{N_{k+1}} (\delta_{pi}^{k+1} \cdot W_{i,j}^{k,k+1}) \quad (21)$$

因此, BP 算法可总结出三个式子:

学习过程:

$$\Delta_p W_{i,j}^{k-1,k} = \eta \delta_{pj}^k O_{pi}^{k-1}, \quad (22)$$

$$(i = 1, 2, \dots, N_{k-1} + 1; j = 1, 2, \dots, N_k; k = 2, 3, \dots, n)$$

对输出节点:

$$\delta_{pj}^n = - (t_{pj}^n - O_{pj}^n) \frac{\partial}{\partial i_{pj}^n} f_j'(i_{pj}^n), \quad (23)$$

对非输出节点:

$$\delta_{pj}^k = \frac{\partial}{\partial i_{pj}^k} f_j'(i_{pj}^k) \sum_{i=1}^{N_{k+1}} (\delta_{pi}^{k+1} W_{i,j}^{k,k+1}). \quad (24)$$

$$(k = 2, 3, \dots, n-1)$$

在 BP 算法中, 输入输出函数一般都采用 Sigmoid 函数, 因此

对输出节点:

$$\begin{aligned}
\delta_{pj}^n &= - (t_{pj}^n - O_{pj}^n) f_j'(i_{pj}^n) (1 - f_j(i_{pj}^n)) \\
&= (t_{pj}^n - O_{pj}^n) O_{pj}^n (1 - O_{pj}^n).
\end{aligned} \quad (25)$$

对非输出节点:

$$\begin{aligned}
\delta_{pj}^k &= f_j(i_{pj}^k) (1 - f_j(i_{pj}^k)) \sum_{i=1}^{N_{k+1}} (\delta_{pi}^{k+1} W_{i,j}^{k,k+1}) \\
&= O_{pj}^k (1 - O_{pj}^k) \sum_{i=1}^{N_{k+1}} (\delta_{pi}^{k+1} W_{i,j}^{k,k+1}).
\end{aligned} \quad (26)$$

为使 E_p 最小而修改连接权 $W_{i,j}^{k-1,k}$ 时, 修改后的连接权为

$$W_{i,j}^{k-1,k}(n+1) = W_{i,j}^{k-1,k}(n) + \eta \delta_{pj}^k O_{pi}^{k-1}. \quad (27)$$

为加快学习速度, 可在式 (27) 上加惯性项

$$\Delta_p W_{i,j}^{k-1,k}(n+1) = \eta \delta_{pj}^k O_{pi}^{k-1} + \alpha \Delta_p W_{i,j}^{k-1,k}(n), \quad (28)$$

即, 加上第 n 次计算得到的变化量 $\Delta W(n)$ 的 α 倍这一项, 求出第 $(n+1)$ 次的变化量 $\Delta W(n+1)$, 从而加快了学习的速度。

如果网络共有 M 层, 而第 M 层仅含输出节点, 第一层为输入节点, 则 BP 算法可描述为

(1) 选定初权值 $W_{i,j}^{1,k}$;

(2) 从第 1 层到第 M 层计算 $i_{pj}^k, O_{pj}^k, \delta_{pj}^k$ (正向过程);

(3) 从 M 层到第 2 层反向计算修正权值 $W_{i,j}^{l-1,l}$ (反向过程);

(4) 用修正后的权值重复 (2) ~ (3) 过程, 直到收敛。

使用 BP 算法的模型把一组样本的 I/O 问题变为一个非线性优化问题, 使用了优化中最普通的梯度下降法, 用迭代法运算求解权相应于学习记忆问题, 加入隐节点使优化问题的可调参数增加, 从而可得到更精确的解。这一模型已成为神经网络的重要模型之一。

3 神经网络故障诊断的应用实例

故障诊断神经网络通常均采用多层结构, 输入层为故障征兆层, 其节点数等于所要识别的模式特征参数的个数。隐层是中间层, 用于提取信号中的高阶相关特性, 节点数取决于所要求的网络复杂程度, 现在一般还是凭经验或试验确定。输出层为故障类别层, 其节点数等于待识别的故障源数, 每一个输出节点对应着某一具体故障。

3.1 滚动轴承故障诊断例

本例采用的神经网络为三层结构, 即输入层、隐层、输出层。输入层节点数与反映轴承状态的各特征参数相对应, 所选用的参数为峭度、波峰因数、冲击脉冲的峰值和均值、解调谱的特征频率及高次谐波 (到五次为止), 因此, 节点数定为 29; 输出层根据轴承状态, 定为 6 个节点, 即与正常以及外圈缺陷、内圈缺陷、滚动体缺陷、保持架缺陷、磨损等 6 种状态相对应; 隐层节点数为 18。

为了确定网络中各单元的连接权, 先使用学习样本对网络进行学习训练。通过适当选择学习样本对网络进行学习训练后, 用 NU204 单列圆柱轴承, 分别在轴承外圈、内圈、滚子上做成微小的伤痕, 形成各种故障轴承, 通过检测仪器得到各种振动特征参数, 送入诊断网络中, 对轴承状态进行诊断, 其诊断结果全部都能正确判断。

3.2 化学反应过程的异常诊断

图 5 所示的连续搅拌反应装置, 其生产过程是, 把原料装入反应槽, 夹层里通入冷却水, 由此来调节反应中槽内的温度, 然后取出槽内的成品。反应槽的入口流量、出口流量、槽内温度三者都是闭环控制。由于设备或部件的故障、人的误操作、外部对控制系统的干扰等原因, 会引起成品出口流量过大和出口压力过高等反应异常, 可能发展成整个反应装置起火、爆炸事故。

为防止事故, 对成品出口流量过大这一异常现象进行监测。通过异常原因的故障树分析, 造成上述异常共有 16 种情况, 可由 7 个中间参数反映, 它们又可以通过在四处安装传感器 P2、F3、P7、P10 检测反应过程参数, 即原料压力、原料入口流量、原料阀门控制压力、成品阀门控制压力来推断 (参见图 5)。为判断异常原因, 构筑了一个异常诊断神经网络。网络为三层结构, 输入层节点数为 4, 输入 4 个传感器的检测数据; 输出层节点数为 7, 与 7 个反映异常原因的中间参数相对应; 中间层选为五个节点。这个网络用学习数据训练后, 对其进行了仿真测试, 它可以通过输入所安装的传感器对反应异常发生时的检测值对反应过程进行正确的诊断。这个网络可作为异常诊断的支援系统使用。

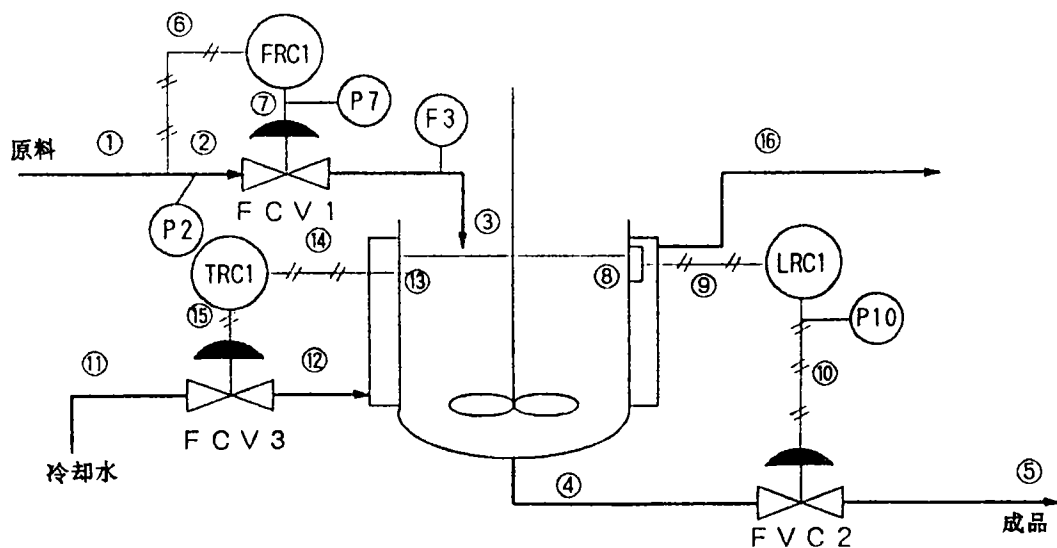


图 5 连续搅拌反应装置

4 结 论

从以上的论述及应用实例可知, 人工神经网络是一种并行分布信息处理结构, 网络的特性由网络节点(处理单元)和连接方式所确定. 它具有高度的并行处理、自组织与自学习能力, 在状态识别与故障诊断领域具有广泛的应用范围. 本文所介绍的人工神经网络在滚动轴承故障诊断和化工流程异常诊断中的应用, 对工程应用具有一定的实用价值.

以上所介绍的应用例子取自于第一作者在日本冈山大学作访问学者时与日方指导、合作教授的共同研究工作, 作为一种探索还有待于进一步深入研究.

参 考 文 献

- [1] 安居院猛, 长桥宏, 高桥裕树. ニューラルプログラム. 昭晃堂株式会社, 1993
- [2] 中野馨. 入門と実習ニューロコンピュータ. 技術评论社, 1989
- [3] 焦李成. 神经网络系统理论. 西安: 西安电子科技大学出版社, 1990
- [4] 丰田利夫. 回转机械診断の進め方. 日本プラントメンテナンス協会, 1991
- [5] 宋京伟, 佐山, 鈴木, 島田. ニューラルネットワークによる转がり軸受の故障診断に関する研究. 日本设备管理学会秋季研究发表大会论文集, 1993. 10
- [6] SAYAMA, SHIMADA, Zongxiao YANG, Jingwei SONG, SUZUKI. Artificial Neural Network Approach to Process Fault Diagnosis. THE SECOND ASIAN/PACIFIC INTERNATIONAL SYMPOSIUM ON INSTRUMENTATION, MEASUREMENT AND AUTOMATIC CONTROL, AUG. 1993. 8

Artificial Neural Network and Fault Diagnosis

Song Jingwei Hayatoshi SAYAMA Kazuhiko SUZUKI

ABSTRACT

This article introduces the basic concept, model and algorithm of artificial neural network, and discusses the examples of its application to diagnosis of bearing fault and chemical industry process abnormal.

Key words : Neural network; BP algorithm; Fault diagnosis