

多层前馈神经网络的学习和综合算法*

张 铃 吴福朝 张 钹 韩 玫

(安徽大学人工智能所,合肥 230039)

(清华大学计算机系,北京 100084)

摘要 本文提出多层前馈网络的一种新的学习和综合算法——FP算法,并证明由此算法得到的网络作为通用联想记忆器时,具有如下优点:(1)每个样本都是吸引中心;(2)每个样本的吸引半径达到最大值;(3)网络没有假吸引中心;(4)网络具有最少的元件个数;(5)学习的复杂性达到最优(就其复杂性的阶而言),故此网络在性能、结构、计算复杂性等方面均达到很好状态。

关键词 神经网络,多层前馈网络,吸引中心,吸引半径,学习算法。

对多层前馈神经网络,目前最流行的学习算法是BP算法,但此算法存在几个主要缺点:一是学习收敛速度慢;二是可能产生假吸引中心;三是所得网络容错能力差。Arai在文献[1]中证明了用三层网络可以完成任何样本的对应,并证明作为通用联想记忆器(p 个样本)的三层网络,其隐含的元件至少要有 $(p-1)$ 个,同时给出这种网络的求解方法。可惜所给的方法其计算量还比较大,且所得的网络性能不佳。故到目前为止对NN网络尚无有效的综合方法。

本文针对现有学习算法的不足,主要讨论作为通用联想记忆器的神经网络的一种新的学习和综合算法,这个算法在学习过程中同时对网络的结构进行综合,当学习结束时就给出网络的最优结构。我们暂且称之为前向传播(Forward Propagation)算法。

在介绍FP算法之前,先谈一谈本文总的思路。作为联想记忆器,最主要的目的是使网络能正确地进行分类(这与求优化问题不同),故不一定要使用 δ -律或BP算法。我们采用向前传播的方法(故称之为FP算法),即先从第一层元件开始,确定其元件个数和各元件的权、阈值。然后以第一层元件的输出为第二层元件的输入,再确定第二层元件的个数和权、阈值……。最后给出整个网络的结构及其各元件的权、阈值。

多层前馈网络,实际上只要三层(即二个元件层)就够了(下面将证明三层是最优结构,超过三层都是多余的)。

当给定 p 个样本时,我们取第一层元件共 $p-1$ 个,每个元件有 n 个输入(设样本输入是 n 维,输出是 m 维),其功能是将 p 个样本输入变换成 $(p-1)$ 维空间中的正交 $(p-1)$ 维单

* 本文1994-02-25收到,1994-08-22定稿

本项目得到攀登计划和863高技术计划的资助。作者张铃,1937年生,教授,研究所所长,主要研究领域为人工智能,应用数学。吴福朝,1956年生,副教授,主要研究领域为应用数学,人工智能。张钹,1935年生,教授,主要研究领域为人工智能,计算机应用技术。韩玫,女,1969年生,博士生,主要研究领域为计算机科学与技术。

本文通讯联系人:张钹,北京100084,清华大学计算机系

纯形的 p 个顶点,这与文献 [1] 中的方法的区别在于:在文献 [1] 中只要求变换成 p 个线性独立的向量,而我们的方法要求变换成正交 $(p-1)$ 维单纯形的顶点,这个特殊性使我们的算法大为简化(几乎只需将样本数据记入元件的权、阈值之中即可)。从第二层到第三层,取 m 个元件。通过第二层元件再将这正交坐标系的原点、单位向量变换成 p 个 m 维的样本输出向量,这样就得到对应于该样本集的联想记忆器的神经网络(这一点也与文献 [1] 中的方法不同,文献 [1] 中第二层是采用线性元件,用求线性方程组的方法求解。这表面上似乎更简单,其实这将使所得的网络容错能力降低。我们在第二层采用非线性元件,并直接给出元件的权和阈值,这既简化了计算又可提高所得网络的容错能力)。

1 FP 算法

设神经元元件 A_i 有 n 个输入,一个输出。下面先讨论各分量只取 $\{-1, 1\}$ 二值的情况。再设第一层网络 A 的输入 x 与输出 z 的对应关系为:

$$z = \text{Sgn}(W * x - \theta) \quad (1)$$

其中 W 是第一层网络 A 的权系数矩阵, θ 是阈值向量, $\text{Sgn}(x)$ 表符号函数。

给定训练样本集 $K = \{r^0 = (x^0, y^0), r^1 = (x^1, y^1), \dots, r^{p-1} = (x^{p-1}, y^{p-1})\}$ 作图 1 的网络结构。

令第一层元件的对应关系为 F , 第二层元件的对应关系为 G 。令 A_i 的权向量为 w^i , 阈值为 θ_i 。 B_i 的权向量为 u^i , 阈值为 ξ_i 。

现在来讨论如何将 $\{x^0, x^1, \dots, x^{p-1}\}$ 通过第一层元件,变换成 $(p-1)$ 维正交单纯形的顶点。

命题 1. x^0, x^1, \dots, x^{p-1} 是 p 个互不相同的 n 维向量,作 p 个 $(p-1)$ 维向量如下:

$$x^0 = (-1, -1, \dots, -1)$$

$$x^1 = (+1, -1, \dots, -1)$$

...

$$x^{p-1} = (-1, -1, \dots, +1)$$

则存在 $(p-1) \times n$ 维权矩阵 W 及 $(p-1)$ 维阈值向量 θ , 使得:

$$z^i = F(x^i), \quad i = 0, 1, \dots, p-1.$$

$$\text{证明: 令 } w^i = (x^i)^T \quad i = 1, 2, \dots, p-1. \quad (2)$$

$$\theta_i = \begin{cases} n - d_i + 1 & (\text{当 } d_i \text{ 是偶数时}) \\ n - d_i & (\text{当 } d_i \text{ 是奇数时}) \end{cases} \quad i = 1, 2, \dots, p-1. \quad (3)$$

其中 $d_i = \min d(x^i, x^j), j \neq i, i = 1, \dots, p-1, j = 0, 1, \dots, p-1, d(x, y)$ 表 x, y 的海明距离

$$\text{先证, 当 } i \neq 0 \text{ 时 } \langle w^i, x^i \rangle - \theta_i > 0 \quad (4)$$

$$\text{当 } i \neq j \text{ 时 } \langle w^i, x^j \rangle - \theta_i < 0 \quad (5)$$

其中 $\langle x, y \rangle$ 表 x 与 y 的内积。

$$\text{因为, 对任意 } x, y \text{ 有 } \langle x, y \rangle = n - 2 * d(x, y) \quad (6)$$

$$\text{故 } \langle w^i, x^i \rangle - \theta_i = \begin{cases} n - (n - d_i + 1) = d_i - 1 > 0, & d_i \text{ 为偶数} \\ n - (n - d_i) = d_i > 0, & d_i \text{ 为奇数} \end{cases}$$

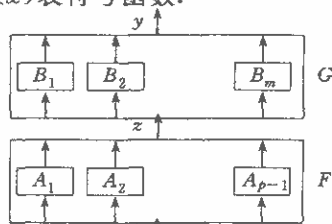


图 1

(4)式得证.

$$\begin{aligned} \text{当 } i \neq j \text{ 时 } \quad & \langle w^i, x^j \rangle - \theta_i = \langle x^i, x^j \rangle - \theta_i \\ & = (n - 2d(x^i, x^j)) - \theta_i \leq (n - 2d_i) - (n - d_i) = -d_i < 0 \end{aligned}$$

(5)式得证.

$$\begin{aligned} \text{由(4),(5)式易得: } & F(x^i) = z^i, \quad i = 1, 2, \dots, p-1. \\ & F(x^0) = z^0. \end{aligned}$$

□

下面讨论第二层元件的学习和综合.

命题 2. 任给 p 个 m 维输出向量 $\{y^0, y^1, \dots, y^{p-1}\}$, 则存在 $m \times (p-1)$ 维权矩阵 U 和阈值向量 ξ , 使得 $y^j = G(z^j)$, $j = 0, 1, \dots, p-1$. (7)

证明: 因(7)式成立的充要条件是 $y_i^j (\langle u^i, z^j \rangle - \xi_i) > 0 \quad i = 1, 2, \dots, m$ (8)

将(8)式写成方程组形式, 取右边均为 1, 得

$$y_i^0 (-u_1^i - \dots - u_{p-1}^i - \xi_i) = 1 \tag{9}$$

$$y_i^1 (+u_1^i - \dots - u_{p-1}^i - \xi_i) = 1$$

.....

$$y_i^j (-u_1^i - u_2^i \dots + u_j^i \dots - u_{p-1}^i - \xi_i) = 1 \tag{10}$$

.....

$$y_i^{p-1} (-u_1^i - \dots + u_{p-1}^i - \xi_i) = 1 \tag{11}$$

当 $y_i^0 = y_i^j$ 时, (10) - (9) 得 $2y_i^j u_j^i = 1 - 1 = 0$, 得 $u_j^i = 0$

当 $y_i^0 \neq y_i^j$ 时, (10) + (9) 得 $2y_i^j u_j^i = 1 + 1 = 2$, 得 $u_j^i = y_i^j$

$$\text{即 } \begin{cases} u_j^i = \begin{cases} 0, & \text{当 } y_i^0 = y_i^j \\ y_i^j, & \text{其他} \end{cases} \\ \xi_i = -(y_i^0 + u_1^i + \dots + u_{p-1}^i) \quad i = 1, 2, \dots, m; \quad j = 0, 1, \dots, p-1. \end{cases} \tag{12}$$

故取 $U = (u^i), \xi = (\xi_i)$. □

结合命题 1, 2 得

定理 1. 当给定训练样本集 $K = \{r^0, r^1, \dots, r^{p-1}\}$, 按(2)、(3)式定义 W 和 θ , 按(12)式定义 U 和 ξ . 则图 1 对应的网络将 x^i 变换成 $y^j, i = 0, 1, \dots, p-1$. 即每个样本输入向量都是吸引中心.

证明: 由命题 1 得对所有 i , 有 $z^i = F(x^i)$

再由命题 2 得对所有 i , 有 $y^j = G(z^i)$

故有 $y^j = G(z^i) = G(F(x^i)) = N(x^i)$. □

2 网络性能分析

2.1 样本的吸引域

设样本 $r^i = (x^i, y^i)$, 且网络将 x^i 变换为 $y^j = N(x^i)$, 则 x^i 的吸引域定义为:

$$D(x^i) = \{x | N(x) = y^i\}$$

其中 $y = N(x)$ 是网络的对应关系.

定理 2. 设样本集 K 有 p 个样本, 若按 FP 算法求其网络, 则样本输入的吸引域为

$$D(x^i) = \{x | d(x^i, x) < d_i/2\}, \quad i = 1, 2, \dots, p-1.$$

$$D(x^0) = \{x | d(x^0, x) \geq d_i/2\}, \quad i=1, 2, \dots, p-1.$$

且网络没有假吸引中心.

$$\text{证明: 先证明当 } i \neq 0, x \in D(x^i) \text{ 时有 } \langle w^i, x \rangle - \theta_i > 0 \quad (13)$$

$$\text{当 } d(x^i, x) \geq d_i/2 \text{ 时有 } \langle w^i, x \rangle - \theta_i < 0 \quad (14)$$

证当 $x \in D(x^i)$ 设 $d_i = 2k$ 是偶数, 则 $d(x^i, x) < d_i/2 = k$

$$\text{有 } \langle w^i, x \rangle - \theta_i = (n - 2 * d(x^i, x)) - \theta_i \geq (n - 2(k-1)) - (n - 2k + 1) = 1 > 0$$

$$\text{当 } d(x^i, x) \geq d_i/2 \text{ 时, } \langle w^i, x \rangle - \theta_i \leq (n - 2k) - (n - 2k + 1) = -1 < 0$$

$$\text{当 } d_i = 2k + 1 \text{ 是奇数, 且 } x \in D(x^i) \text{ 时有 } \langle w^i, x \rangle - \theta_i \geq (n - 2k) - (n - (2k + 1)) = 1 > 0$$

$$\text{当 } d(x^i, x) \geq d_i/2 \text{ 时, 由 } d(x^i, x) \geq k + 1/2, \text{ 则 } d(x^i, x) \geq k + 1.$$

$$\text{所以 } \langle w^i, x \rangle - \theta_i \leq (n - 2(k+1)) - (n - (2k+1)) = -1 < 0$$

故(13)、(14)式得证.

$$\text{又当 } x \in D(x^i), i \neq j \text{ 时, 有 } d(x^j, x) \geq d(x^i, x^j) - d(x^i, x) \geq \max\{d_i, d_j\} - d_i/2 \geq d_j/2$$

$$\text{由(13)式得 } F^i(x) = \text{Sgn}(\langle w^i, x \rangle - \theta_i) = 1 = z_i^i, \quad i \neq 0$$

$$\text{由(14)式得 } F^j(x) = \text{Sgn}(\langle w^j, x \rangle - \theta_j) = -1 = z_j^j, \quad i \neq j$$

得 $F(x) = z^i, \quad i=1, 2, \dots, p-1.$

即 x^i 的吸引域为: $D(x^i)$.

同理可得 x^0 的吸引域为: $D(x^0)$.

$$\text{又因为 } D(x^0) \cup D(x^1) \cup \dots \cup D(x^{p-1}) = X$$

其中 X 是输入向量全空间, 故知网络没有假吸引中心. □

应该指出:

(1) 是否存在有假吸引中心, 曾困扰着许多学习算法, 这里我们成功地解决了这个问题.

(2) 易知, 从平均角度来看, 样本的吸引半径的上界是 $d_i/2$, FP 算法达到了这个上界. 这里 x^i 的吸引半径定义为 $r = \max\{d | \text{当 } d(x^i, x) < d, \text{ 则 } x \in D(x^i)\}$

(3) FP 算法共用了 $(m+p-1)$ 个元件, 就元件个数看, 我们的网络也达到了最优状态.

2.2 FP 算法的计算复杂性

计算矩阵 W 的计算量为: $n \times (p-1)$, 而计算 θ 的计算量为: $(p-1) \times (p-1)$,

计算矩阵 U 的计算量为: $m \times (p-1)$, 计算 ξ 的计算量为: $m \times (p-1)$,

故总的计算量为: $(p-1) \times (n+2m+p-1)$.

这个计算量就其阶而言也达到最优.

2.3 知识的扩充

当我们已经学习了 p 个样本后, 现在再学习第 $(p+1)$ 个样本, 这种情况我们称之为知识的扩充. 下面的算法能很快完成这个任务.

知识扩充算法. (1) 在第一层增加一个元件 A_p .

$$\text{取 } w^p = (x^p)^T,$$

$$\theta_p = \begin{cases} (n - d_p + 1), & \text{当 } d_p \text{ 为偶数时} \\ (n - d_p), & \text{当 } d_p \text{ 为奇数时} \end{cases}$$

(2) 令 $d_i(t)$ 是根据前 $t+1$ 个样本计算得的 d_i 值.

$$\text{令 } d_i(p) = \min\{d_i(p-1), d(x^i, x^p)\}, \quad i=1, \dots, p-1.$$

$$d_p(p) = \min(d(x^i, x^p)) \quad i \neq p$$

(3) 在第二层各元件增加一个输入线, 其对应的权向量分量 u_p^i 为:

$$u_p^i = \begin{cases} 0, & \text{当 } y_i^0 = y_i^p \\ y_i^p & \text{当 } y_i^0 \neq y_i^p, \quad i = 1, 2, \dots, m. \end{cases}$$

(4) 将 ξ_i 改为 $\xi_i - u_{p-1}^i, i = 1, 2, \dots, m$.

经过这样修改后的网络, 就将第 $(p+1)$ 个样本也学习进去了。

实际上, 知识扩充算法给出了增量式的 FP 算法, 即 FP 的迭代算法。

容易计算知识扩充的计算量不超过 $(n+2m+p-1)$ 。

本算法的优点之一是在学习新知识时, 对已学过的旧知识基本上保持原来状态, 即保持知识的稳定性。而以往的算法常常会发生“学新的, 忘旧的”现象。

这种增量式的学习方法合乎人类学习的过程, 而文献[1]中的算法没有这种优点。

另外算法的计算量也达到最优(就阶而言)。

3 网络结构的分析

(1) 本网络共用 $(m+p-1)$ 个元件, 故是最少元件网络(证明见文献[1])。

(2) 利用上述的知识扩充算法, 对本网络极易进行新知识的学习, 且其学习的计算量也达到最优情况。

(3) FP 算法求得的 W, U 矩阵, 其分量只取 $\{-1, 0, +1\}$ 三种值, 阈值分量也只取不多于 n 的整数值, 故易于硬件实现。

(4) 从我们所得的网络来看, 每个样本的信息基本是独立地存放在第一层的一个元件上和第二层的各元件的不同的输入线上。从这点看, 其存贮象是集中存贮, 其优点在于便于增加新知识和对旧知识的修改。另一方面在每个元件上又以权向量形式存放, 这又象是分布式存贮, 其优点在于增强容错能力。各样本信息之间的联系由各层元件的阈值表示出来, 故我们的网络的存贮是集中式和分布式相结合的形式。它具有两者的优点。

(5) “拒识”状态的设置

在识别的应用中, 有时不要求对任何输入都要给出分类, 如在文字识别中, 对错字和模糊不清的字, 就不要要求给出对应的分类, 即允许有“拒识”的情况。

用 FP 算法可以很容易设置“拒识”状态, 方法如下:

取 $r^* = (x^*, y^*)$, 且 r^* 与所有 r^i 均不同, 取 r^* 为第零个样本。

再将 r^i 重新排成 r^1, r^2, \dots, r^p 。

这样, 因为 $D(r^*) = \{x | d(x^i, x) \geq d_i/2, i = 1, 2, \dots, p-1\}$, 故不在 x^i 的吸引域中的点必在 $D(x^*)$ 中, 这些点可以看成是拒识的点, 其输出均为 y^* , 即凡对应的输出是 y^* 的输入均视为拒识点。

4 实 例

FP 算法可以推广到输入向量的分量是一般实数的情况, 输出向量仍为二值向量。为了节省篇幅, 其证明省去。

现举一例子,求 XOR 函数对应的网络:

令 XOR 的真值表如下:

	x_1	x_2	y
x^0	-1	-1	-1
x^1	+1	-1	+1
x^2	-1	+1	+1
x^3	+1	+1	-1

按公式(2),(3)作权、阈值如下:

$$W^1 = (+1, -1), \quad \theta_i = 1, i=1, 2, 3.$$

$$W^2 = (-1, +1)$$

$$W^3 = (+1, +1)$$

作第二层权、阈值

$$u_1 = u_2 = 1, \quad u_3 = 0$$

$$\xi = -1$$

作对应网络如图 2.

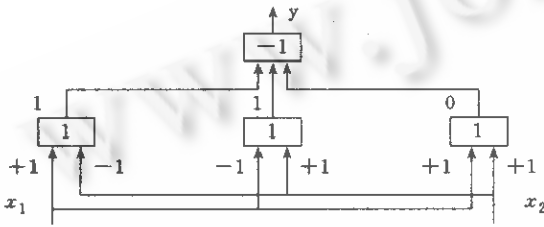


图2

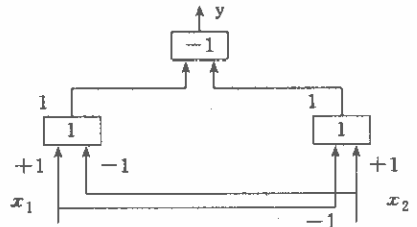


图3

图 2 中 $u_3=0$, 故可将第一元件层的第三个元件删去得图 3.

图 3 与文献[2](p. 64)中的 XOR 网络相同.

容易证明,用类似方法可证: n 维的奇偶问题的对应网络需要 $2^{n-1}+1$ 个元件,第一层(共 2^{n-1} 个元件)的权、阈值如下:

$$w^i = (x^{p_i})^T, \quad i = 1, 2, \dots, 2^{n-1}.$$

$$\theta_i = 1, \quad i = 1, 2, \dots, 2^{n-1}.$$

其中 $x^{p_i} \in \{x | x \text{ 的正分量之和为奇数}\}$.

第二层(共一个元件)的权阈值为:

$$u_i = 1, \quad i = 1, 2, \dots, 2^{n-1}.$$

$$\xi = -2^{n-1} + 1$$

对二变量的奇偶问题,若用 BP 算法,一般每个样本要学习几千次方可求到其解,对 n 个变量的奇偶问题,若用 BP 算法其计算量更大了. 而用 FP 算法,我们几乎不花什么计算就得到所要的网络. 但对 n 维奇偶问题直接用 FP 算法求得的网络,其隐含层的元件个数还嫌太多. 如何降低隐含层的元件个数问题,我们将在另文讨论.

5 与已有的结果的比较^[1,3-6]

(1) 与 BP 算法的比较

BP 算法也许是目前用得最多的算法之一,但是它有几个固有的缺点:

① 因为用的是梯度下降法,故(除非能量函数是凸函数时)不可避免地将产生局部极小的现象,使算法不是完备的,同时还可能产生假吸引中心,这就使在识别时可能会陷入假稳态,致使识别错误,降低识别的正确率.这些都是其固有的缺点.

② 因为 BP 算法是对样本逐个进行学习的,又因元件的权向量在迭代调整过程中是相互关联的,这样一来算法的收敛速度必然是很慢的.(从直观上看,其计算的复杂性基本是元件个数的指数阶).

③ 若元件的功能函数取符号函数,那么算法的迭代过程是从“非解”到“解”的迭代过程,即最后一步的“非解”到“解”的差 $=\alpha\delta x_j$,即只有一步之遥.故所得的网络必然容错能力很差,因为当“解”受到噪音干扰时,就很容易变为“非解”(因为两者之差只有一步之遥).

以上所述的缺点对 BP 算法都是本质的,对 BP 算法的任何改进,若不是本质的都将无法克服上述的缺点.

而我们用 FP 算法得到的网络,没有以上的缺点,且在计算量、容错能力、元件个数(指作为通用联想记忆器而言)等方面都达到最优.就网络无假吸引中心这一点,也是目前其它算法所难以做到的.

(2) 与 Hopfield 网络的比较

① H 网络的学习速度与 FP 算法相仿,但因 H 网络是自反馈网络,固其在进行识别时的计算量比 FP 算法大得多.

② H 网络的容量很小,其容量的上界 $<0.138N$ (N 是网络的规模).

令 $N = p + m$ (p 是样本个数, m 是目标向量的维数),则 FP 算法的容量 v 为:

$$v = (p / (p + m)) * N = a * N$$

当 p 很大时,接近 1,当 p 与 m 相当时, $a \sim 0.5$,故用 FP 算法所得的网络的容量比 H 网络的容量大得多.

③ H 网络一般情况下,其容错能力很差(甚至有时连样本都可能记不住).

④ H 网络也具有一般梯度法的共同缺点(可能产生局部极小和假吸引中心).

凡此种种,可以看出 H 网络若作为联想记忆器不是一个理想的模型.

(3) 与 Arai 方法比较

Arai 在文献[1]中给出的方法在元件个数达到最少这点上与我们的方法相同,文献[1]中的方法的计算量比 FP 算法大得多,且所得的网络的容错能力比我们的网络差.

总之,我们的网络在计算量、容错能力、元件个数等方面都达到了最优.当然,在引入构造性方法的同时,也付出了一些代价,如网络对元件损坏的容错能力降低了.

6 模拟结果

我们取 10 个标准的字母 A, a, B, b, C, c, D, d, E, e 为样本,要求将它们分为 5 类: (A, a), (B, b), (C, c), (D, d), (E, e) (见图 4).

网络结构为:第一元件层 10 个元件,第二元件层 5 个元件.

我们对标准的样本随机地增加或减少“黑色”像素,从而得到带噪声的检验样本,样本的噪声率为增加或减少的像素点数占总像素点数的比例值.分别用 FP 算法、BP 算法、H 网络和文献[1]中的方法进行学习,比较其学习复杂性,然后用带噪声的测试样本进行输入,统计

其识别的正确率. 结果见下表. 表中的学习步数是指 10 个标准样本的平均学习步数, 识别率和叠代步数是对 100×10 个检验样本进行平均的结果. 从表中容易看出 FP 算法比 BP 算法及文献[1]中的方法, 无论在计算量上还是在识别的正确率上都优越得多.

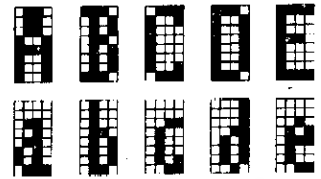


图 4

学习 算法	稳定 样本数	学习 步数	5%噪声		10%噪声		15%噪声		20%噪声	
			识别率	叠代步数	识别率	叠代步数	识别率	叠代步数	识别率	叠代步数
FP	10	0	100%	1	95%	1	65%	1	49%	1
BP	10	313	87%	1	65%	1	45%	1	20%	1
Hopfield	6	0	55%	2.94	44%	3.34	39%	3.45	35%	3.54
Arai	10	0	100%	1	28%	1	24%	1	25%	1

7 结 论

本文给出一个新的学习和综合的算法——FP 学习综合算法, 具有以下几个优点: (1) 计算量最少(就阶而言达到最少). (2) 网络结构最简单(作为通用联想记忆器, 就元件个数而言也达到最少). (3) 每个样本均是吸引中心, 其吸引半径(就平均而言)达到最大值. (4) 网络没有假吸引中心, (还可人为增加“拒识”状态). (5) 网络极易进行知识扩充. (6) 网络的元件的权矩阵的系数只取 $-1, 0, +1$ 三个值, 阈值也只取不多于 n 个的整数, 且元件的权和阈值不必在学习过程中不断地变化, 只要在学习开始时一次(按 FP 算法要求)性将权与阈值存入即可, 这样就克服了元件硬件化的最主要的困难. 故便于硬件实现. (7) 本算法融学习与综合于一体, 这是其他算法所不具备的.

凡此种种, 可以说, 对于作为通用联想记忆器的神经网络(这里“通用”是指对任给 p 个样本, 网络在不改变其拓扑结构情况下, 都能进行联想记忆)我们给出了一个极好的学习和综合的算法.

我们的方法与已有的学习方法(如 BP 算法、 δ -算法等)不但有以上所述的优点, 更主要的是在方法论上的区别, 一般的学习方法是在局部上进行调整搜索, 已经有理论证明(见文献[2]); 即使在树上进行启发性搜索, 在一般情况下其计算量也是指数阶的, 故按这种思路进行求解很难得到好的结果. 而我们的学习方法是在全局上进行规划求解, 事实证明使用全局的观点能得到好的结果.

其次, 我们是将学习和综合融合在一起的, 而过去的方法是将两者截然分开.

最后, 我们的方法吸收了传统线路设计的思想, 即将总的线路功能分成几个各自完成不同任务的子模块来设计的, 本文多层前馈网络的 FP 算法, 将总线路分成两层(元件层), 第一层完成由输入向量到 $(p-1)$ 维单纯形的变换, 第二层完成由 $(p-1)$ 维单纯形到输出向量的变换. 当然, 上述网络对通用联想记忆器来讲, 具有元件最少的性质, 对具体给定的样本集, 其隐含层的元件个数一般来说不是最少的. 对这个问题我们将在另文讨论(即对具体给

定的样本集,如何构造网络使其隐含的元件尽可能少).

另外,对反馈网络的学习问题我们也将另文讨论.

参考文献

- 1 Arai M. Mapping abilities of three-layer neural networks. Proc. of IEEE IJCNN-1989 Vol. I, 1989. 419-423.
- 2 Pearl J. Heuristics. Addison-Wesley Company, 1984.
- 3 Enhutsu I *et al.* A neural network model for learning. Proc. of IJCNN-1991 Vol. II, 1991. 461-465.
- 4 Fu L M. Fuzzy rule extraction from a multilayered neural network. Proc. of IJCNN-1992 Vol. I, 1992. 343-348.
- 5 Zurada J M. Introduction to artificial neural systems. West Publishing Company, St. Paul, U. S. A., 1992.
- 6 Rumelhart D E, McClelland J L. Parallel distributed processing. Cambridge: MIT Press, MA, 1986.

A LEARNING AND SYNTHESIS ALGORITHM OF MULTILAYERED FEEDFORWARD NEURAL NETWORKS

Zhang Ling Wu Fuchao

(Institute of Artificial Intelligence, Anhui University, Hefei 230039)

Zhang Bo Han Mei

(Department of Computer Science, Tsinghua University, Beijing 100084)

Abstract A new learning algorithm—forward propagation (FP) of multilayered feed-forward neural networks is presented in this paper. The authors show that as an associative memory the network constructed by the FP algorithm has several advantages. (1) Each training sample is an attractive center. (2) The attractive radius of each training sample reaches the maximum. (3) There is no spurious attractive center in the network. (4) The network has minimal number of elements. (5) The order of its learning complexity is optimal. The FP learning algorithm is also an effective synthesis tool, i. e., the network architecture can be constructed during its learning process.

Key words Neural network, multilayered feedforward network, attractive center, attractive radius, learning algorithm.