

# 求解线性不定方程组所展现的BP与Hopfield 类型神经网络的学习同质性研究\*

张雨浓<sup>1</sup>, 谭宁<sup>2</sup>, 李展<sup>1</sup>, 莫锦辉<sup>1</sup>

(1. 中山大学信息科学与技术学院, 广东 广州 510275; 2. 中山大学软件学院, 广东 广州 510275)

**摘要:** 主要针对求解线性不定方程组的两种并行计算神经网络模型(BP和Hopfield类型神经网络)进行了探讨。BP神经网络和Hopfield类型神经网络尽管在起源、网络定义、拓扑结构和学习模式上有较大的不同,但这两类人工神经网络在相同学习率条件下求解线性不定方程组中却可以表现出相同的数学公式、学习本质和计算能力,即学习同质性。此外,分别在零初值、相同但非零初值和不同随机初值三种情况下,针对两类人工神经网络求解线性不定方程组体现出来的学习同质性进行了计算机仿真验证,并证实了两类神经网络算法的有效性。

**关键词:** 不定方程; BP神经网络; Hopfield神经网络; 学习同质性

**中图分类号:** TP183 **文献标志码:** A **文章编号:** 0529-6579(2010)02-0001-08

## Common Nature of Learning in BP and Hopfield-type Neural Networks Solving an Underdetermined System of Linear Equations

ZHANG Yunong<sup>1</sup>, TAN Ning<sup>2</sup>, LI Zhan<sup>1</sup>, MO Jinhui<sup>1</sup>

(1. School of Information Science and Technology, Sun Yat-sen University, Guangzhou, 510275, China;

2. School of Software, Sun Yat-sen University, Guangzhou, 510275, China)

**Abstract:** The online solution of underdetermined linear equations is investigated by using two types of artificial neural networks (i. e., BP and Hopfield-type neural networks). Although they differ from each other in terms of origins, network definition, structures and learning patterns, the BP and Hopfield-type neural networks could be exploited for solving online such underdetermined linear equations and even possess a common mathematical formulation of learning and common computational abilities. In addition, based on zero initial values, the same but nonzero initial values and different random initial values, computer-simulation and verification results are given. These substantiate well the efficacy and commonness of such two types of neural networks on solving underdetermined linear equations.

**Key words:** underdetermined equation; BP neural network; Hopfield neural network; common nature of learning

很多工程实践中所遇到的计算问题可以最终归结或转化为求解线性方程组<sup>[1-5]</sup>,而线性不定方程组便是其中常见的一大类<sup>[4-8]</sup>。通常的不定方程也被称为丢番图方程<sup>[6,8]</sup>,是指方程中未知变量的个数多于方程的个数、且未知数的取值范围受到某些限制(如整数、正整数或有理数)的方程<sup>[8-9]</sup>。

不定方程大类中有一些特殊的方程一直以来受到众多学者的关注与研究,其中如勾股方程<sup>[10-11]</sup>、佩尔(Pell)方程和费马(Fermat)大定理<sup>[8-13]</sup>等等。值得提及,形如 $x^2 + y^2 = z^2$  ( $x, y, z \in Z^+$ , 即,均为正整数)的二次三元方程称为勾股方程,其正整数解称为商高数、勾股数或毕达哥拉斯

\* 收稿日期: 2009-06-12

基金项目: 国家自然科学基金资助项目(60775050); 新世纪优秀人才支持计划基金资助项目(NCET-07-0887)

作者简介: 张雨浓(1973年生),男,教授,博士生导师, E-mail: zhynong@mail.sysu.edu.cn

数<sup>[10-11]</sup>。《周髀算经》中有“勾广三，股修四，经隅五”的记载，即(3, 4, 5)是勾股方程的一个解<sup>[10-11]</sup>，它的全部正整数解已在16世纪前被发现；而形如 $x^2 - dy^2 = \pm 1$  ( $x, y \in \mathbb{Z}, d \in \mathbb{Z}^+$  且不含平方因子)的二次方程称为佩尔方程<sup>[8,12]</sup>；另外，费马大定理是指三元不定方程 $x^n + y^n = z^n$  ( $n \geq 3$  为整数)无正整数解<sup>[8,13]</sup>，其证明曾是数学界的难题，在1995年被普林斯顿大学的怀尔斯(A. Wiles)教授完全解决<sup>[8,13]</sup>。

不同于上述经典有趣的非线性不定方程求解，本文主要研究线性不定方程组的人工神经网络求解方法以及展示出来的不同网络模型共有的学习同质性现象。从数学表达而言，待求解的线性不定方程组可以简单地用如下矩阵向量的形式定义为<sup>[5-8]</sup>

$$Ax = b \quad (1)$$

其中 $A \in \mathbb{R}^{m \times n}$  ( $m < n$ ) 假定为行满秩的系数矩阵，系数向量 $b \in \mathbb{R}^{m \times 1}$ ， $x \in \mathbb{R}^{n \times 1}$  为待求解的未知向量，该线性不定方程组有 $m$ 个一次方程和 $n$ 个未知变量(也称 $n$ 元)。对于该不定线性方程组(1)，其解往往为无穷多个(或称多解)。

一般来说，求解线性不定方程组(1)的常规解法是传统的串行数值方法，如迭代算法和矩阵解法等<sup>[5,7]</sup>。当待处理变量维数较小的情况下，串行数值方法往往奏效；但待处理数据维数大到一定程度时，传统的数值方法就可能达不到所需的计算速度和精度要求。因此，针对数值算法的串行计算效率低下问题，不少学者先后提出和发展了众多的并行计算方法，其中也包括分布式存储的人工神经网络方法等<sup>[14-20]</sup>。

本文在前人工作的基础上首先构造了一种两层结构、线性神经元激励的BP神经网络<sup>[17-20]</sup>用于求解线性不定方程组，并详细探讨其拓扑结构、训练样本集组成以及权值修正公式等特色；接着对同样的线性不定方程组求解任务，换一种人工神经网络的求解思路，分析和采用了一种连续的Hopfield类型神经网络动力学模型<sup>[1-2,15-16]</sup>，并通过离散化得到在线求解线性不定方程组的离散Hopfield类型神经网络。对比这两类人工神经网络模型，可以发现：虽然BP和Hopfield类型神经网络在原理和拓扑结构上存在着很大的差异，但它们就求解线性不定方程组任务进行网络学习时能表现出了相同的数学本质和计算能力。

另外，值得指出的是：线性不定方程组的一个重要特点就是其解通常有多个乃至无穷多个，为了比较不同初值时两类人工神经网络的求解效果，在

计算机仿真验证中本文采用设定零初值、相同非零初值和不同随机初值这三种情况下对两者的仿真结果进行了比较。我们发现BP与Hopfield类型人工神经网络能求得一致的结果，这进一步验证了二者之间的学习本质和深层次的联系(相关性)。

## 1 BP神经网络求解模型

BP(误差回传)神经网络<sup>[17-20]</sup>是一种按误差反向传播算法训练的多层前向神经网络，是目前应用最为广泛的神经网络模型之一，其于1986年由Rumelhart和McClelland等人提出<sup>[21-22]</sup>。该网络的学习过程主要包含两个方面：样本信号的正向传递和误差信号的反向传播。

### 1.1 网络结构描述

为了求解线性不定方程组 $Ax = b$ ，可以构建一个结构简单的两层BP神经网络模型，如图1所示<sup>[17-20]</sup>。通常可以定义该神经网络的输入输出层之间的连接权值向量为 $w \in \mathbb{R}^{n \times 1}$ ，网络输入向量为 $y \in \mathbb{R}^{1 \times n}$ ，网络输出值为 $z \in \mathbb{R}$ 。出于快速运算和表达简洁的目的，在本任务中网络所有输入和输出神经元均可采用线性激励函数 $f(u) = u$ ，且可固定阈值均为零。根据图1，权值列向量 $w$ 定义如下：

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \in \mathbb{R}^{n \times 1}$$

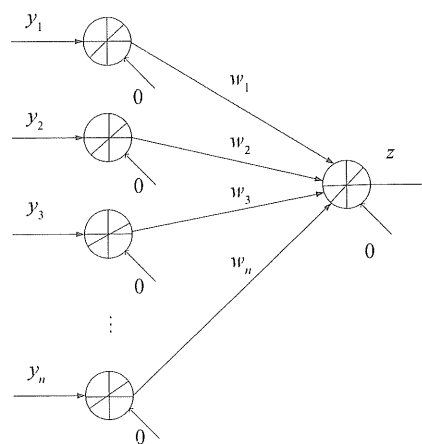


图1 求解线性不定方程组 $Ax = b$ 之两层BP神经网络模型

Fig. 1 A two-layer BP neural network model solving an underdetermined system of linear equations,  $Ax = b$

输入行向量  $y$  可定义为  $y \in [y_1, y_2, \dots, y_n] \in \mathbf{R}^{1 \times n}$ , 因此该神经网络输入  $y$  和输出  $z$  之间的关系可表述如下式:

$$z = f\left(\sum_{i=1}^n y_i w_i\right) = \sum_{i=1}^n y_i w_i = yw$$

### 1.2 网络训练样本

为了求解上述线性不定方程组 (1), 我们首先把系数矩阵  $A \in \mathbf{R}^{m \times n}$  拆分写成如下向量表述形式:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$$

其中行向量  $a_i := [a_{i1}, a_{i2}, \dots, a_{in}] \in \mathbf{R}^{1 \times n}$ 。

根据图 1 所示的 BP 神经网络的输入输出关系, 我们可以设定行向量  $a_i$  作为神经网络的第  $i$  次的输入样本  $y^{(i)}$ , 而其对应的目标输出样本  $z^{(i)}$  则可定义为  $z^{(i)} = b_i$ , 其中  $b_i$  是线性不定方程组 (1) 中系数向量  $b$  的第  $i$  个元素。因此, 可以把样本输入  $y^{(i)}$  和样本输出  $z^{(i)}$  配对写成如下样本集的形式:  $\{(a_i, b_i), i = 1, 2, \dots, m\}$ , 也即:

$$y^{(i)} = a_i, z^{(i)} = b_i \quad (2)$$

基于前述的神经网络输入  $y$  和实际输出  $z$  之间的映射关系, 针对 (2) 式给出的训练样本对集合, 我们可以定义 BP 神经网络的学习/训练误差函数  $E(w)$ , 从而推导出 BP 神经网络的权值修正公式, 施加对网络权值  $w$  的调节以达到求解线性不定方程组的目的, 具体见如下 1.3 节。

### 1.3 网络权值修正

为了获得权值公式, 我们首先定义图 1 所示 BP 神经网络的误差函数  $E(w)$  (即, 批处理方式下的误差函数) 为

$$\begin{aligned} E(w) &= \frac{1}{2} \sum_{i=1}^m (z^{(i)} - \sum_{i=1}^n y_i^{(i)} w_i)^2 = \\ &= \frac{1}{2} \sum_{i=1}^m (z^{(i)} - y^{(i)} w)^2 = \\ &= \frac{1}{2} \sum_{i=1}^m (b_i - a_i w)^2 = \\ &= \frac{1}{2} \|Aw - b\|_2^2 \end{aligned}$$

此处的权值向量  $w$  可以认为 (或等同为) 待求解的线性不定方程组之未知向量  $x$ , 其中  $\|\cdot\|_2$  表示向量的二范数。BP 神经网络的权值向量  $w$  调整至稳态, 也即求解得到了上述线性不定方程组 (1) 之

解向量  $x$ 。

依据经典负梯度下降法来减小 BP 神经网络的批处理误差  $E(w)$ , 该简单结构 BP 神经网络的连接权值向量  $w \in \mathbf{R}^{n \times 1}$  可以采用如下公式进行修正:

$$\begin{aligned} \Delta w &= w(k+1) - w(k) = -\eta \left. \frac{\partial E(w)}{\partial w} \right|_{w=w(k)} = \\ &= -\eta A^T (Aw(k) - b) \end{aligned}$$

也即

$$w(k+1) = w(k) - \eta A^T (Aw(k) - b) \quad (3)$$

其中  $\eta > 0$  为该 BP 神经网络的学习率 (应足够小), 迭代次数  $k = 0, 1, 2, 3, \dots$ 。经过对连接权值向量  $w$  的足够多次修正迭代之后,  $\Delta w \approx 0$  (即连接权值向量  $w$  将收敛至某一稳定值而不再变化), 此时图 1 所示的 BP 神经网络的连接权值向量  $w$  即为方程 (1) 的解  $x^* : \lim_{k \rightarrow \infty} w(k) = x^*$ , 换言之, 经过足够多次的权值迭代修正后, BP 神经网络能够求得不定方程组 (1) 的解。值得注意及指出的是<sup>[23]</sup>, 因 (1) 为线性不定方程组, 它往往有无穷多个解: 当网络初始权值  $w(0)$  为零时, 上述 BP 神经网络求得的是  $A^+ b$  (即线性方程组的伪逆解, 或称极小范数解); 当网络初始权值  $w(0)$  为非零向量值时, 该 BP 神经网络将收敛到其它邻近初值的解上去, 并理论上满足  $\lim_{k \rightarrow \infty} E(w(k)) = 0$ 。关于该 BP 神经网络解的收敛特性, 将在第三节通过更多仿真示例验证。

## 2 连续与离散 Hopfield 神经网络求解模型

Hopfield 类型人工神经网络是一种递归神经网络模型<sup>[1-2, 15-16]</sup>, 于 1982 年由 Hopfield 等人提出<sup>[21-22]</sup>, 该类型网络从输出至输入均有反馈连接, 并可分为连续型和离散型两种分支。基于负梯度法的连续 Hopfield 类型神经网络也可推广和应用用于求解线性不定方程组 (1)。其设计方法简述为: 首先定义该连续型 Hopfield 神经网络的能量函数为  $\varepsilon = \|Ax(t) - b\|_2^2 / 2$ , 然后利用负梯度法的设计思想来最小化能量函数  $\varepsilon$  (其也是一种误差函数), 从而可以得到求解该线性不定方程组 (1) 的连续 Hopfield 类型神经网络为

$$\frac{dx(t)}{dt} = -\mu A^T (Ax(t) - b) \quad (4)$$

其中  $\mu > 0$  为网络设计参数 (应足够大)。该 Hopfield 类型神经网络 (4) 求解线性不定方程组 (1) 的网络结构如图 2 所示。

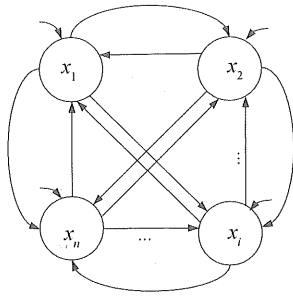


图 2 求解线性不定方程组  $Ax = b$  的 Hopfield 类型神经网络模型结构

Fig. 2 Architecture of Hopfield-type neural network solving an underdetermined system of linear equations,  $Ax = b$

为了探讨 Hopfield 网络结构和实现上的特点, 我们可定义  $\dot{x}_i$  为  $dx/dt$  的第  $i$  个元素, 则公式 (4) 进一步展开为

$$\begin{aligned} \dot{x}_i = & -\mu(\bar{a}_{i1}x_1 + \bar{a}_{i2}x_2 + \dots + \bar{a}_{in}x_n) + \\ & \mu(a_{i1}b_1 + a_{i2}b_2 + \dots + a_{im}b_m) = \\ & -\mu(\bar{a}_{i1}x_1 + \bar{a}_{i2}x_2 + \dots + \bar{a}_{in}x_n) + \mu\bar{b}_i \end{aligned}$$

其中标量  $\bar{a}_{ij}$  表示矩阵  $A^T A$  的第  $ij$  个元素,  $i, j \in \{1, 2, \dots, n\}$ , 标量  $\bar{b}_i$  表示  $a_{i1}b_1 + a_{i2}b_2 + \dots + a_{im}b_m$ 。由此可得该连续型 Hopfield 神经网络 (4) 的电路框图见图 3 (a) 所示。

为了更好地与上述求解线性不定方程组 (1) 的 BP 神经网络模型 (见第一节) 作对比, 我们可以进一步地由连续 Hopfield 类型神经网络模型 (4) 推导出其离散化的 Hopfield 类型神经网络模型。采用欧拉前向差分公式可得其离散 Hopfield 类型神经网络模型如下 (其中  $h > 0$  为采样间隔, 应足够小):

$$\frac{x(k+1) - x(k)}{h} = -\mu A^T(Ax(k) - b)$$

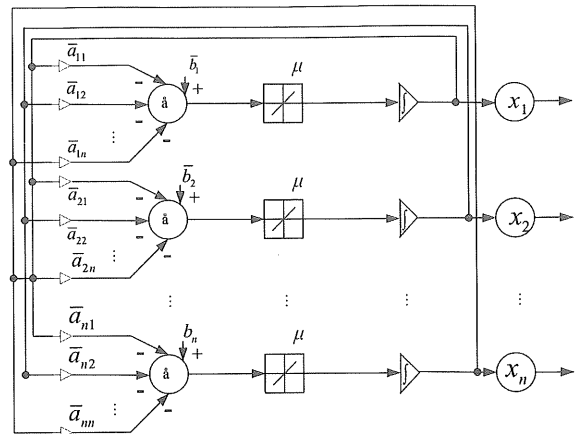
也即,

$$x(k+1) = x(k) - \eta A^T(Ax(k) - b) \quad (5)$$

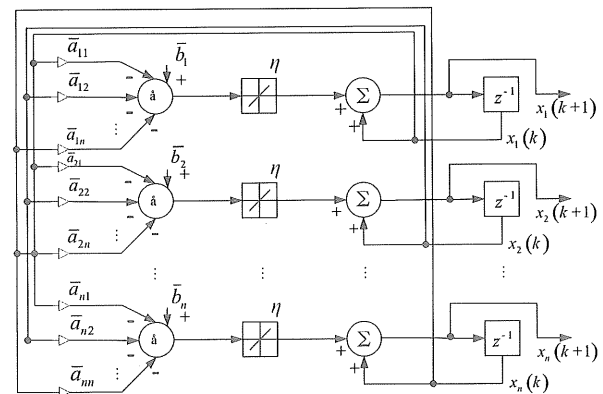
其中  $\eta := \mu h > 0$  为迭代步长 (也应足够小), 迭代次数  $k = 0, 1, 2, 3, \dots$ 。同理也可给出图 3 (b) 所示的离散 Hopfield 类型神经网络模型 (5) 的电路框图。

在经过足够多次的状态迭代/变换之后, 状态向量  $x(k+1) \approx x(k)$ , 即 Hopfield 神经网络系统 (5) 达到稳态, 因而可推出为  $\lim_{k \rightarrow \infty} x(k) = x^*$ , 这说明离散 Hopfield 类型神经网络模型 (5) 中的状态  $x$  在经过足够多次的变换后, 也可以最终收敛到线性不定方程组 (1) 的理论解  $x^*$ 。同样只有在初值  $x(0)$  为零的情况下上述离散 Hopfield 类型神

经网络才收敛到伪逆解  $A^+b$ , 否则将会收敛到其它解上去<sup>[23]</sup>。



(a) 连续 Hopfield 类型神经网络模型 (4) 的电路框图



(b) 离散 Hopfield 类型神经网络模型 (5) 的电路框图

图 3 求解 (1) 式的 Hopfield 类型人工神经网络的电路框图

Fig. 3 Circuit schematics of Hopfield-type neural networks solving (1)

(a) Continuous-time Hopfield-type neural network model (4)

(b) Discrete-time Hopfield-type neural network model (5)

对比上述的离散 Hopfield 类型神经网络模型 (5) 和前述的 BP 神经网络的权值修正公式 (3), 我们可以发现这两个迭代公式在数学形式和本质上是一致的! 进一步对照可以发现, 尽管 BP 神经网络和 Hopfield 类型神经网络在起源、设计原理、网络拓扑和电路结构上存在着很大的差异, 但它们表达学习过程的本质数学计算公式是一致的 (即都来自于误差最小原理的、可以一致描述的权值/状态更新公式)。为了能更直观地说明上述两类人工神经网络模型对求解线性不定方程组 (1) 的简单有效性和学习同质性, 本文下一节将进行丰富的计算机仿真验证和展示。

### 3 计算机仿真验证

为了验证 Hopfield 类型人工神经网络和 BP 神经网络模型对求解线性不定方程组 (1) 的可行性、有效性以及学习同质性, 我们利用离散 Hopfield 类型神经网络模型 (5) 和 MATLAB 软件的神经网络工具箱所构建的 BP 神经网络模型分别对上述线性不定方程组 (1) 进行求解并仿真<sup>[22]</sup>。通过零初值、相同非零初值和不同随机初值这三种情况下的仿真, 比较两类神经网络的最终收敛值。本节以如下线性不定方程组系数矩阵  $A \in \mathbb{R}^{2 \times 3}$  [其为行满秩, 即  $\text{rank}(A) = 2$ ] 和系数向量为  $b \in \mathbb{R}^{2 \times 1}$  作为计算机仿真示例:

$$A = \begin{bmatrix} 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix}, b = \begin{bmatrix} 4 \\ 1 \end{bmatrix}$$

1) 构建 Hopfield 类型神经网络求解上述线性不定方程组, 其主要程序代码如下:

```
A = [1 -2 1; 1 1 -2]; b = [4; 1];
Initial = zeros(3,1); % Initial = rand(3,1);
eta = 1.2/trace(A' * A); x = Initial;
error0 = 0.5 * norm(b - A * x)^2;
for i = 1:60
    x = x + eta * A' * (b - A * x);
    error(i) = 0.5 * norm(b - A * x)^2;
end
```

其中 `Initial = zeros(3, 1)` 表示初始值为零, 也可根据需要设为非零的状态或随机产生的初值向量 [即 `Initial = rand(3, 1)`]。

2) 构建 BP 神经网络的主要程序代码如下:

```
net = network(1,1);
net.inputs{1}.size = 3;
net.layers{1}.size = 1;
net.biasConnect = [0];
net.inputConnect = [1];
net.layerConnect = [0];
net.outputConnect = [1];
net.targetConnect = [1];
net.layerWeights{1}.learn = 1;
net.layers{1}.transferFcn = 'purelin';
net.performFcn = 'mse';
net.trainFcn = 'traingd';
net.layerWeights{1}.learnFcn = 'learngd';
net.trainParam.goal = 1e-16;
```

在上面的程序代码中, 我们调用了 MATLAB 神经网络工具箱中内置的基于标准 BP 算法 (负梯度下降法) 的学习函数 `learngd`、训练函数 `traingd` 和线性激励函数 `purelin`, 并设置了上述 BP 神经网络

的学习误差函数类型为 `mse` (即最小均方误差), 其目标值设定为  $10^{-16}$  (一个较高的精度目标)。

#### 3.1 零初值仿真

下面对离散 Hopfield 类型神经网络 (5) 求解线性不定方程组 (1) 进行仿真, 网络的初始输入向量  $x(0)$  设为零向量。其对应的仿真结果见图 4 和 5 所示。图 4 是状态轨迹曲线, 可以看到随着迭代次数  $k$  的增加, Hopfield 类型神经网络每个状态向量  $x_i(k)$  分别收敛至某一稳态值, 如迭代 60 次后,

$$x(60) = [x_1(60), x_2(60), x_3(60)]^T = [1.6667, -1.3333, -0.3333]^T$$

其 (此处保留小数点后四位数字) 准确地收敛到了线性不定方程组 (1) 的伪逆解

$$x^* = A^+ b = \begin{bmatrix} 1.6667 \\ -1.3333 \\ -0.3333 \end{bmatrix}$$

图 5 为误差下降曲线, 可以看到其在迭代十余步后已很好地收敛至零了。对照上述结果可以发现, 在初值为零的情况下, 离散 Hopfield 类型神经网络模型 (5) 能够收敛到该线性不定方程组 (1) 的伪逆解  $x^* = A^+ b$  上去。

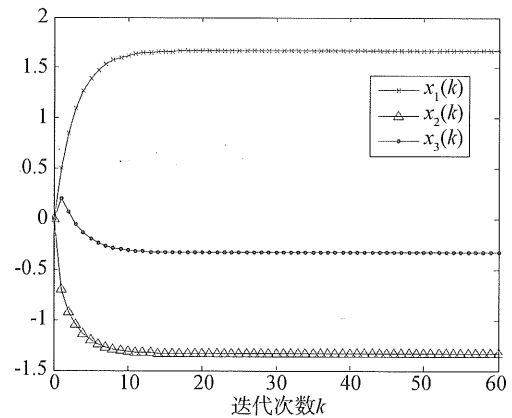


图 4 零初值离散 Hopfield 类型神经网络模型 (5) 求解线性不定方程组 (1)

Fig. 4 Solving an underdetermined system of linear equations (1) via discrete-time Hopfield neural network model (5) with a zero initial state

同样地, 我们可以采用 MATLAB 神经网络工具箱对线性不定方程组 (1) 进行零初值 BP 神经网络求解仿真。根据图 1 如前所示使用神经网络工具箱构建了一个两层线性激励的 BP 神经网络模型, 其中输入层含有 3 个神经元, 输出层含有 1 个神经元。样本输入输出对分别由上述满秩系数矩阵  $A \in \mathbb{R}^{2 \times 3}$  和系数向量  $b \in \mathbb{R}^{2 \times 1}$  依照 (2) 式拆分方法生成。

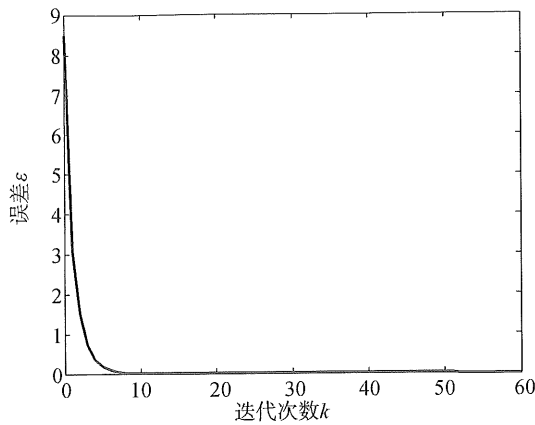


图 5 零初值离散 Hopfield 类型神经网络 (5) 的误差变化曲线

Fig. 5 Computational error  $\varepsilon$  of discrete-time Hopfield neural network model (5) solving an underdetermined system of linear equations (1)

从图 6 所示的误差变化曲线可以看出, BP 神经网络在训练了 55 次时 (见图的横坐标所示), 其均方误差  $E(w)$  已经下降为  $5.7104 \times 10^{-17}$ , 满足了设定的目标精度  $10^{-16}$ , 并在零初值的情况下当网络训练完毕之后的权值向量  $w$  为

$$w(55) = \begin{bmatrix} w_1(55) \\ w_2(55) \\ w_3(55) \end{bmatrix} = \begin{bmatrix} 1.6667 \\ -1.3333 \\ -0.3333 \end{bmatrix}$$

再次通过与上述伪逆解的比较, 可以发现采用神经网络工具箱构建的简单 BP 网络也能有效地求解线性不定方程组 (1)。

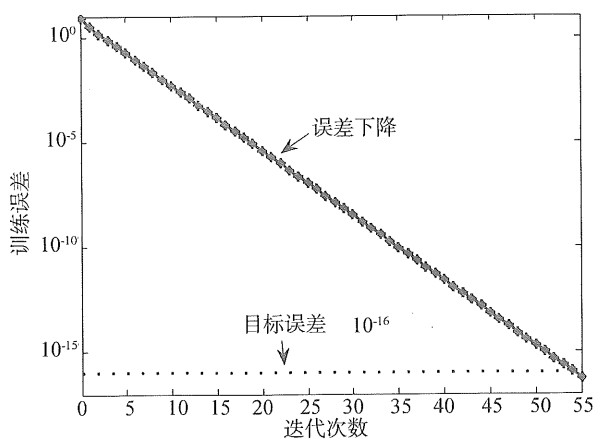


图 6 基于神经网络工具箱构建的 BP 网络的训练误差变化曲线

Fig. 6 Training error of a BP neural network constructed by MATLAB neural network toolbox for solving underdetermined linear equations (1)

### 3.2 相同非零初值仿真

下面进行相同但非零初值条件下的计算机仿真。首先设定 Hopfield 神经网络模型 (5) 的状态初值

$$x(0) = \begin{bmatrix} 1.0000 \\ 0.0000 \\ 3.0000 \end{bmatrix}$$

作为展示例子。经过 60 次迭代后, 得到如下状态终值

$$x(60) = \begin{bmatrix} x_1(60) \\ x_2(60) \\ x_3(60) \end{bmatrix} = \begin{bmatrix} 3.0000 \\ 0.0000 \\ 1.0000 \end{bmatrix}$$

可知该离散 Hopfield 类型神经网络 (5) 收敛到了线性不定方程组 (1) 的一个解上, 但不同于上述的伪逆解  $x^* = A^+b$ 。

对图 1 所示的 BP 神经网络, 给定与 Hopfield 神经网络相同的非零初始权值 (即  $w(0) = [1, 0, 3]^T$ ); 经过 55 次迭代训练之后, BP 网络达到了目标精度, 程序中中止并输出:

$$w(55) = \begin{bmatrix} w_1(55) \\ w_2(55) \\ w_3(55) \end{bmatrix} = \begin{bmatrix} 3.0000 \\ 0.0000 \\ 1.0000 \end{bmatrix}$$

对比两者在同样初值的情况下, 可以看到: Hopfield 神经网络和 BP 神经网络的解是一致的。也就是说, 在同样初值的情况下, 在经过足够多次的迭代后, 两类神经网络均能收敛到同一个解上。与第 3.1 小节的比较可知, 当采用两类神经网络进行求解仿真时, 通过调整不同的初值可得到不同的线性不定方程组 (1) 的解; 当且仅当零初值时会收敛到伪逆解。

### 3.3 不同随机初值仿真

从以上两小节观察到了, 零初值和相同非零初值情况下 BP 和 Hopfield 类型神经网络最终的收敛结果均是相同的, 在这一小节中我们验证在随机产生的不同初值时两类神经网络对线性不定方程组 (1) 的求解情况。

对于 Hopfield 类型神经网络模型 (5), 可启动原本注释了的指令 `Initial = rand(3, 1)` 随机产生  $[0, 1]$  区间内的初始值

$$x(0) = \begin{bmatrix} 0.4898 \\ 0.4456 \\ 0.6463 \end{bmatrix}$$

经过 60 次迭代后得到如下状态终值结果

$$x(60) = \begin{bmatrix} x_1(60) \\ x_2(60) \\ x_3(60) \end{bmatrix} = \begin{bmatrix} 2.1939 \\ -0.8061 \\ 0.1939 \end{bmatrix}$$

同样地, 对 BP 网络取  $[0, 3]$  区间内随机产生的不同初值

$$w(0) = \begin{bmatrix} 2.2966 \\ 2.3856 \\ 0.5606 \end{bmatrix}$$

经过 55 次迭代训练后, 得到如下稳定的权值结果

$$w(55) = \begin{bmatrix} w_1(55) \\ w_2(55) \\ w_3(55) \end{bmatrix} = \begin{bmatrix} 3.4143 \\ 0.4143 \\ 1.4143 \end{bmatrix}$$

通过对比两类神经网络的求解结果, 我们可以发现: 当取随机产生的不同初值时, BP 和 Hopfield 类型神经网络由于初值的不同而分别收敛到线性不定方程组 (1) 的不同解上。

总的来说, Hopfield 类型神经网络模型 (5) 与 MATLAB 神经网络工具箱构建的 BP 神经网络都可以用来求解线性不定方程组 (1)。当且仅当取零初值时, 二者都会收敛至线性不定方程组 (1) 的伪逆解  $x^* = A^+b$ ; 在相同非零初值情况下, BP 和 Hopfield 网络会收敛到不定方程组的相同但非伪逆解上; 在取不同的随机初值验证时, 可以看到两类神经网络各自收敛到与其初值对应的不同的理论解上去。

#### 4 结束语

本文探讨了 BP 和 Hopfield 类型神经网络对线性不定方程组  $Ax = b$  的求解问题。对比 BP 神经网络的权值修正公式 (3) 和离散 Hopfield 类型神经网络状态更新公式 (5), 可以得出如下结论: BP 神经网络与 Hopfield 类型神经网络虽然在起源、设计原理、网络拓扑结构、电路实现以及系统的物理意义上均有很大的不同, 但其二者的学习本质可以是相同的, 如最终推导出的公式 (3) 和 (5) 是一样的 (除了所用符号及其物理意义的不同), 具有本文所言的学习同质性共同点。计算机仿真验证结果更进一步地说明, 通过设定不同的初始值, 两类神经网络均能有效地收敛到与初值对应的线性不定方程组的理论解; 初始值相同时, 两类神经网络能有效且等效地收敛到相同的与初值对应的线性不定方程组的理论解上去, 从而进一步证实了 BP 和 Hopfield 类型人工神经网络共通的学习同质性现象 (本质上都是基于误差最小原理)。

#### 参考文献:

- [1] ZHANG Y, GE S. Design and analysis of a general recurrent neural network model for time-varying matrix inversion [J]. IEEE Transactions on Neural Networks, 2005, 16 (6): 1477 - 1490.
- [2] ZHANG Y, JIANG D, WANG J. A recurrent neural network for solving Sylvester equation with time-varying coefficients [J]. IEEE Transactions on Neural Networks, 2002, 13 (5): 1053 - 1063.
- [3] 付朝江. 基于工作站机群并行求解有限元线性方程组 [J]. 计算机工程与应用, 2008, 29 (24): 6441 - 6443.
- [4] 张元标. 基于一次不定方程的数字图像加密技术 [J]. 成都理工大学学报: 自然科学版, 2006, 33 (6): 645 - 648.
- [5] DENG Y. A matrix solution for linear indeterminate equation group with integral coefficients [J]. Henan Science, 2008, 26 (5): 520 - 524.
- [6] 王春岩, 田晓霞. 关于不定方程组  $x - 1 = 3py^2$ ,  $x^2 + x + 1 = 3z^2$  [J]. 数学理论与应用, 2008, 28 (4): 109 - 112.
- [7] 晏林. 多项式环上一次不定方程组的矩阵解法与程序设计 [J]. 科技通报, 2005, 21 (4): 373 - 378.
- [8] 柯召, 孙琦. 谈谈不定方程 [M]. 上海: 上海教育出版社, 1980.
- [9] 王迪吉, 高福康, 段芳. 关于不定方程的整数解及其解数的讨论 [J]. 新疆师范大学学报: 自然科学版, 2008, 27 (3): 5 - 8.
- [10] 杨丽英, 庄清寿. 由商高不定方程看整边直角三角形 [J]. 安庆师范学院学报: 自然科学版, 2008, 14 (2): 113 - 115.
- [11] 王美艳, 齐恩风. 素数  $p$  与勾股定理 [J]. 阜阳师范学院学报: 自然科学版, 2007, 24 (4): 43 - 44.
- [12] 吴文良, 孙骏. 关于 Pell 方程最小解计算公式的 2 个定理 [J]. 云南民族大学学报: 自然科学版, 2008, 17 (2): 131 - 132.
- [13] 李宏棋. 费马大定理的初等证明 [J]. 西安工程大学学报, 2008, 22 (5): 650 - 662.
- [14] 李立平, 韩兵欣, 刘利贤. 一种新型动力神经网络的理论算法研究 [J]. 微电子学与计算机, 2009, 26 (1): 45 - 47.
- [15] 张雨浓, 张禹珩, 陈轲, 等. 线性矩阵方程的梯度法神经网络求解及其仿真验证 [J]. 中山大学学报: 自然科学版, 2008, 47 (3): 26 - 32.
- [16] 李晓东, 杨国华. 反馈神经网络对非线性时变离散系统的近似 [J]. 中山大学学报: 自然科学版, 2002, 41 (5): 28 - 30.

#### 2.4 与单膜散射法和阻止柱双散射法的比较

通过 Monte Carlo 计算, 得到单膜散射法的束流利用率为 5%<sup>[10]</sup>, 单阻止柱双散射体方法的束流利用率为 17%, 而双环双散射体方法的束流利用率为 27%。双环双散射体方法的束流利用率明显提高。单膜散射法通常用来扩展半径为 2 cm 以下的束流, 双散射体方法可以用来扩展半径为 10 cm 以下的束流, 是单膜散射法的 5 倍。双环双散射体方法的形成距离仅为 300 cm, 比单阻止柱双散射体方法小, 便于安装在转动机架上。因此, 双环双散射体方法是被动散射体方法中比较好的方法。

### 3 结 论

利用 Monte Carlo 方法, 计算双环双散射体方法的束流分布。结果表明: 随着膜厚的增加, 由 Fluka 计算得到第一个散射体的厚度和平均能量损失与解析计算的结果相差越来越大。这种差别是由于解析计算过程中做了一些近似造成的。此外, 用液体散射体取代固体散射体在理论上是可行的。双环双散射体方法优于单膜散射法和挡光器双散射体方法。

#### 参考文献:

- [1] FONTENOT J D, NEWHAUSER W D, TITT U. Design tools for proton therapy nozzles based on the double-scattering foil technique [J]. *Radiation Protection Dosimetry*, 2005, 116: 211 - 215.
- [2] SMITH A R. Proton therapy [J]. *Physics in Medicine and Biology*, 2006, 51: R491 - R504.
- [3] JONES D T L, SCHREUDER A N, SYMONS J E, et al. Proton therapy at iThemba LABS [J]. *Radiation Physics and Chemistry*, 2004, 71: 983 - 984.
- [4] POLF J C, HARVEY M C, TITT U, et al. Initial beam size study for passive scatter proton therapy I. Monte carlo verification [J]. *Medical Physics*, 2007, 34: 4213 - 4218.
- [5] POLF J C, HARVEY M C, SMITH A R, et al. Initial beam size study for passive scatter proton therapy II. Changes in delivered depth dose profiles [J]. *Medical Physics*, 2007, 34: 4219 - 4222.
- [6] HERAULT J, IBORRA N, SERRANO B, et al. Monte Carlo simulation of a proton therapy platform devoted to ocular melanoma [J]. *Medical Physics*, 2005, 32: 910 - 919.
- [7] KOEHLER A M, SCHNEIDER R J, SISTERTSON J M. Flattening of proton dose distributions for large-field radiotherapy [J]. *Medical Physics*, 1977, 4: 297 - 301.
- [8] TAKADA Y. Dual-ring double scattering method for proton beam spreading [J]. *Japan Journal Applied Physics*, 1994, 33: 353 - 359.
- [9] FERRARI A, SALA P R, FASSO A, et al. A multi-particle transport code Fluka [M]. CERN European Organization for Nuclear Research, 2005.
- [10] 鞠志萍, 曹午飞, 刘小伟. 质子散射角分布的蒙特卡罗模拟 [J]. *物理学报*, 2009, 58: 174 - 177.
- [11] 余建国, 郁庆长. 质子治疗装置的束流扩展研究 [J]. *高能物理与核物理*, 1997, 21: 851 - 859.
- [12] 郁庆长, 罗正明. 质子治疗技术基础 [M]. 北京: 原子能出版社, 1999: 100 - 103.
- [17] YANG S, HO C, LEE C. HBP: improvement in BP algorithm for an adaptive MLP decision feedback equalizer [J]. *IEEE Transactions on Circuits and Systems*, 2006, 53 (3): 240 - 244.
- [18] 张雨浓, 陈扬文, 刘巍, 等. 改造前向神经网络结构以求网络权值直接确定——Jacobi 正交基神经网络实例 [J]. *自动化与信息工程*, 2008, 29(1): 1 - 5.
- [19] ZHANG Y, LI W, YI C, et al. A weights-directly-determined simple neural network for nonlinear system identification [C]. *Proceedings of IEEE International Conference on Fuzzy Systems*, 2008: 455 - 460.
- [20] 张雨浓, 刘巍, 易称福, 等. Legendre 正交基前向神经网络的权值直接确定法 [J]. *大连海事大学学报*, 2008, 34 (1): 32 - 36.
- [21] 邹阿金, 张雨浓. 基函数神经网络及应用 [M]. 广州: 中山大学出版社, 2009.
- [22] 周开利, 康耀红. 神经网络模型及其 MATLAB 仿真程序设计 [M]. 北京: 清华大学出版社, 2005.
- [23] WANG J. Recurrent neural networks for computing pseudoinverses of rank-deficient matrices [J]. *SIAM Journal on Scientific Computing*, 1997, 18(5): 1479 - 1493.

(上接第 7 页)