

改进的 Hopfield 网络图像复原

王 宇, 何小海

(四川大学电子信息学院, 成都 610064)

摘要: 基于对图像恢复 Paik 的 Hopfield 网络模型的分析, 对图像复原提出了一种改进的基于连续函数的全并行自反馈 Hopfield 网络, 通过引入计算的参数, 而不是按照实验或经验获得, 可以使网络收敛速度更快。改进后的 Hopfield 网络模型对退化图像的复原结果与 J.K. Paik 的方法比较, 结果表明此算法使图像复原处理更快, 并且图像恢复效果好。与固定参数 比较显示, 计算出的参数 对网络有更好的收敛速度。

关键词: 连续函数; Hopfield 网络; 图像复原

Improved Hopfield Network for Image Restoration

WANG Yu, HE Xiao-hai

(College of Electronics and Information Engineering, Sichuan University, Chengdu 610064)

【Abstract】 Based on the analysis of Paik's Hopfield network for image restoration, this paper presents a modified full parallel self-feedback continuous Hopfield network model to restore degraded image. And by introducing a calculated parameter rather than obtained by experiments or experience, the modified network converge can be made faster. The result of the model in restoring blurred image is compared with that of J.K. Paik's method, which shows the method makes the restoration process faster and output better restoration quality. In experiment 2, compared with the constant parameter, it testifies that the calculated parameter can give better convergence rate for the network.

【Key words】 continuous function; Hopfield network; image restoration

传统的图像复原方法通常都建立在一些假设的基础上, 例如系统是一个线性空间不变系统, 图像服从一定的概率密度分布, 图像是平稳的, 其目的在于便于进行数学处理。而实际上, 大多数成像系统或多或少存在某种程度的非线性和非平稳, 而且图像的先验概率也是未知的^[1]。此外, 在传统复原方法的求解过程中, 均采用解巨大的方程组或迭代方法进行求解, 其计算速度和收敛速度均比较慢, 不适合对图像的快速或实时处理。

神经网络由于其固有的自学习、自适应性、强鲁棒性和推广能力, 致使其应用领域越来越多。同时由于神经网络在并行处理及大容量计算方面存在着巨大的潜能, 因此被用于解决图像处理领域内的多种问题。用神经网络进行退化图像的复原便是其中的应用之一。由于它在很多方面表现出了比传统图像复原方法^[1]更加优越的性能, 因此 Zhou 等人于 1988 年提出了神经网络图像复原方法^[2], 后来经过 Paik 的改进, 再加上近年来, 关于神经网络图像复原方法的研究也越来越多^[3,4], 其用于图像复原的神经网络模型在不断地丰富, 算法在不断改进, 运算速度也越来越快。

考虑到神经网络的运算能力和对图像复原的效果, 这里提出将改进的 Hopfield 网络用于显微图像的复原中, 以改进图像复原的效果, 加快运算速度, 减少运算量。

1 基本原理

1.1 图像恢复的 Hopfield 网络建立

一个线性位移不变的具有 M 个灰度电平的 $L \times L$ 离散图像退化模型通常可以用矩阵形式表示如下:

$$Y = H \cdot X + N \quad (1)$$

其中, H 为相应于模糊函数的模糊矩阵; N 为加性噪声; X 和 Y 相对应于原始图像和退化图像的一维矢量^[3]。

若设模糊矩阵具有线性不变性且图像具有周期边界性, 则模糊矩阵 H 是一个由系统点扩展函数生成的分块循环矩阵的 Toeplitz 矩阵。

$$H = \begin{bmatrix} H_0 & H_{L-1} & \cdots & H_1 \\ H_1 & H_0 & \cdots & H_2 \\ \cdots & \cdots & \cdots & \cdots \\ H_{L-1} & H_{L-2} & \cdots & H_0 \end{bmatrix}$$

$$\text{其中, } H_i = \begin{bmatrix} h_{i,0} & h_{i,L-1} & \cdots & h_{i,1} \\ h_{i,1} & h_{i,0} & \cdots & h_{i,2} \\ \cdots & \cdots & \cdots & \cdots \\ h_{i,L-1} & h_{i,L-2} & \cdots & h_{i,0} \end{bmatrix} \quad (2)$$

图像恢复就是根据 Y 和 H 求原始图像 X , 恢复过程常用的约束误差方程为

$$E = \frac{1}{2} \|Y - HX\|^2 + \frac{1}{2} \lambda \|DX\|^2 = \min \quad (3)$$

其中, λ 为调和参数, 作用是抑制噪声和平滑原始图像, 当 λ 大时, 解趋于平滑, 反之则数据拟和误差变小; D 是高通滤波器。将式(3)展开, 得

$$\begin{aligned} E &= \frac{1}{2} (Y^T Y - Y^T H X - X^T H^T Y + X^T H^T H X) + \frac{1}{2} \lambda (X^T D^T D X) \\ &= \frac{1}{2} X^T T X - B^T X + \frac{1}{2} Y^T Y \end{aligned} \quad (4)$$

基金项目: 国家自然科学基金资助项目(60372079)

作者简介: 王 宇(1975 -), 男, 博士研究生, 主研方向: 数字图像处理, DSP 应用; 何小海, 教授、博士生导师

收稿日期: 2006-09-18 **E-mail:** scuwangyu@163.com

舍去常数项 $\frac{1}{2}Y^T Y$ ，得到

$$E = \frac{1}{2}X^T T X - B^T X \quad (5)$$

其中， $T = H^T H + \lambda D^T D$ ， $B = H^T Y$ 。而 Hopfield 的能量函数为

$$E_{hop} = -\frac{1}{2}V^T W V - B^T V \quad (6)$$

其中， V 为网络状态矢量； W 为网络权值；网络的状态变化为

$$v_i(t+1) = G\left(\sum_j w_{ij} v_j(t) + b_i\right) \quad (7)$$

其中， $G(u) = \begin{cases} 1, & u \geq 0 \\ 0, & u \leq 0 \end{cases}$ ，将式(4)和式(5)比较，可知若令

$$W = -T = -(H^T H + \lambda D^T D) \quad (8)$$

$$B = H^T Y \quad (9)$$

则

$$V = X \quad (10)$$

即当网络达到稳定状态， $E_{hop} = E$ 达到最小时对应的 V 就是要求的原始图像 X 。然而考虑图像复原的特点：图像的像素灰度值为 $0 \sim 255$ ，而 Hopfield 网络节点状态具有二值性；又由于 T 正定，即 $w_{ii} = -t_{ii} < 0$ ，网络能量递减的条件得不到满足，不能保证收敛性。所以不能直接应用 Hopfield 网络，需要对其进行修正^[4,5]。

1.2 Paik 的 Hopfield 神经网络的图像复原

为克服 Hopfield 神经网络的上述限制，Zhou 的 Hopfield 神经网络模型，在很大程度上克服了传统恢复方法中必须求解高维方程组的缺陷。尽管 Zhou 的神经网络图像复原方法在一定程度上克服了传统方法的缺陷，然而，从图像复原判断准则出发构造出能量函数后，其推导出的网络权矩阵的对角元素并不等于 0。由于 Hopfield 网络能量单调递减的条件得不到满足，因此，在 Zhou 的串行算法中每一步都需要判断能量是增或减；同时由于该模型中每个像素的灰度均被表示成神经元状态(0 或 1)的简单代数之和，因此，网络规模大，且运行时间长。

针对这两个缺点，Paik 等率先对神经网络的模型进行了改进^[6]，该网络每个神经元对应一个取值在 $0 \sim 255$ 间的像素灰度值。其神经元状态变化规则为 UR1。

$$\text{UR1: } x_i(t+1) = g(x_i(t) + \Delta x_i), i = 1, \dots, n$$

$$g(v) = \begin{cases} 0, & v < 0 \\ v, & 0 \leq v \leq 255 \\ 255, & v > 255 \end{cases} \quad (11)$$

$$\Delta x_i = f(u_i) = \begin{cases} -1, & u_i < -\theta_i \\ 0, & u_i \leq |\theta_i| \\ 1, & u_i > \theta_i \end{cases} \quad (12)$$

其中， $\theta_i = -\frac{1}{2}w_{ii} > 0$ ； $u_i = b_i - \sum_j w_{ij} x_j(t)$ 。

网络能量为

$$E = -\frac{1}{2}x^T W x - B^T x \quad (13)$$

当第 i 个神经元的状态从 x_i 变换到 $x_i + \Delta x_i$ 时，导致网络能量变化为

$$\Delta E_i = \frac{1}{2}w_{ii}(\Delta x_i)^2 - u_i \Delta x_i = -\Delta x_i \left(u_i - \frac{w_{ii}}{2} \Delta x_i\right) \quad (14)$$

在 Paik 的文章中已经证明只要保证 Δx_i 与 $u_i - \frac{w_{ii}}{2} \Delta x_i$ 同号就能保证 $\Delta E_i < 0$ 。

2 改进算法

虽然 Paik 的算法是对 Zhou 的算法的改进，但不是最佳的。在迭代过程中，神经元的值经常在终值振荡。在 Zhou 的算法中，一个神经元能量可能需要 100 或者更多的状态变化才能减小到最小。为避免 Paik 模型的缺点，可采用神经元连续变化的模型，如文献[7]将式(12)阈值函数变为

$$\Delta X = f(U) = U = B - TX \quad (15)$$

这里利用一个参数 γ 改进这个算法，于是令 $\Delta X = \gamma \times U$ ， $\gamma > 0$ 。在文献[7,8]中， γ 的选择并没有一定的标准，往往通过实验比较获得。利用投影网络应用^[9-11]，在本文中，通过目标函数和约束条件，可以计算 γ 而不是通过实验或经验获得。于是全并行自反馈改进 Hopfield 算法如下。

算法

(1) 退化图像像素值归一化， $Y = Y - 0.5$ ；初始值 $X(0) = H^T Y$ ； $t = 0$ ；网络收敛误差取 $err = 10^{-5}$ ；

(2) 判断是否 $err < 10^{-5}$ ，如果是，转(6)；否则转下一步；

(3) $TEMP = g(X(t) + \Delta X)$ ；

$$\Delta X = \gamma \times U；U = B - TX \quad (16)$$

$$g(v) = \begin{cases} -0.5, & v < -0.5 \\ v, & 0.5 \leq v \leq 0.5 \\ 0.5, & v > 0.5 \end{cases} \quad (17)$$

$$(4) \quad err = \frac{1}{N} \sum_{i=1}^N |temp_i(t) - x_i(t)| = \frac{1}{N} \|TEMP - X(t)\|；t = t + 1；$$

$X(t+1) = TEMP$ ；

(5) 计算 γ ，回到(2)步；

(6) $X = X + 0.5$ ，得到恢复图像 $X \in [0, 1]^n$ 。

重要的问题是 γ 如何确定？由式(12)的网络能量函数对 γ 求导，得 $\frac{d}{d(\gamma)} E(X(t+1)) = 0$ 时，网络能量变化最快。且有

$$\frac{d}{d(\gamma)} E(X(t+1)) = E'(X(t+1))^T \frac{d}{d(\gamma)} X(t+1) = E'(X(t+1))^T r(t) \quad (18)$$

式中， $r(t)$ 为残值误差。由网络能量式(5)，得

$$E = \frac{1}{2}X^T T X - B^T X$$

如果 $\frac{d}{d(\gamma)} E(X(t+1)) = 0$ ，并且计算残留误差为

$$r(t+1) = -E'(X(t+1)) = -TX(t+1) + B \quad (19)$$

为了确定 γ ，由式(19)改写式(18)，得

$$\frac{d}{d(\gamma)} E(X(t+1)) = E'(X(t+1))^T r(t) = -r(t+1)^T r(t) = 0 \quad (20)$$

并且 $(-TX(t+1) + B)^T r(t) = 0$ ，由式(19)，得到

$$r(t) = -TX(t) + B = U \quad (21)$$

于是，得到公式 $X(t+1) = X(t) + \gamma \bullet U = X(t) + \gamma \bullet r(t)$ ，因而有

$$(-TX(t+1) + B)^T r(t) = ((-TX(t) + B) - \gamma \bullet Tr(t))^T r(t) = 0$$

即

$$(r(t) - \gamma \bullet Tr(t))^T r(t) = 0 \quad (22)$$

根据式(21)， γ 能计算如下：

$$\gamma = \frac{r^T(t)r(t)}{r^T(t)Tr(t)} \quad (23)$$

注意当 $t=1$ 时， γ 应设为一个常数，并且 $0 < \gamma \leq 1$ ，而不是由式(23)计算获得。

3 仿真与结果分析

在这一节，实验 1 在相同条件下应用 Paik 的算法和本文

算法 1。实验用的原始清晰图为具有 256 个灰度电平 128×128 的 Miss 图。复原效果的优劣可借助信噪比的改善来判断：

$$\Delta_{SNR} = 10 \log_{10} \frac{\|Y - X\|}{\|X(t) - X\|} \quad (24)$$

其中 Y , X 和 $X(t)$ 分别表示原始清晰图像、退化图像以及第 t 步迭代后的复原图像。

设点扩展函数为

$$H = \begin{bmatrix} \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \end{bmatrix}$$

用 H 对图 1(a) 进行模糊，然后加上 $SNR=35dB$ 的 Gauss 白噪声，得到图 1(b) 为模糊退化后的图像。图 1(c) 为经 Paik 方法复原后的图像， $\Delta_{SNR} = 4.01, t_{stop} = 60$ ，图 1(d) 为经本文方法得到的复原图像， $\Delta_{SNR} = 6.21, t_{stop} = 16$ 。其中， t_{stop} 是网络收敛时的迭代次数。

实验 1

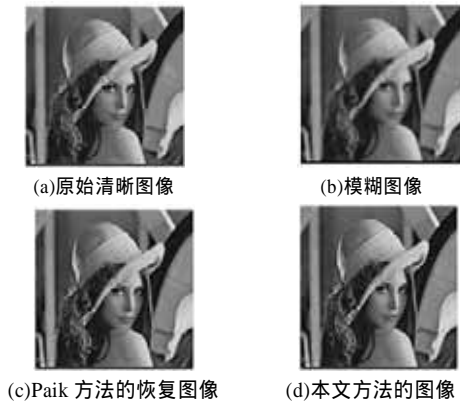


图 1 图像处理结果

通过图 1(c) 及图 1(d) 结果对比可以看出，本文的图像复原方法和 Paik 方法在相同的收敛条件下，达到的复原效果比 Paik 方法好，并且收敛更快。主要因为 Paik 算法网络中节点状态单位阶跃变化的缺点，直接影响了复原图像的质量和收敛速度。

实验 2

每次迭代的能量变化 ΔE 绘制于图 2，为了方便计算，从原始清晰图像采样 64×64 的图像。 $X \in [-0.5, 0.5]$ ， E 按式(2)计算。

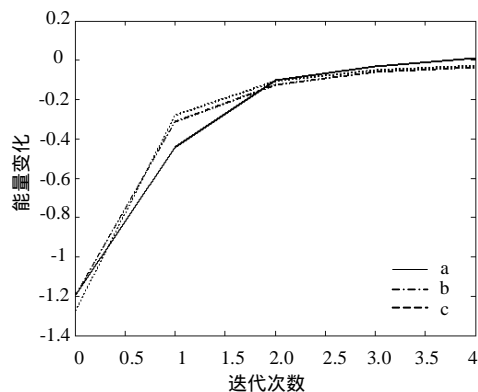


图 2 本文算法在不同 γ 条件下每次迭代能量变化

其中：

- (1) 曲线 a 中 γ 由式(22)计算获得；
- (2) 曲线 b 中 γ 每次迭代不变，设为 0.9；
- (3) 曲线 c 中 γ 每次迭代不变，设为 1.0。

图 2 显示曲线 a 比曲线 b 和曲线 c 在第 1 次迭代后每次迭代具有更大的斜率。(本文算法当 $t=1$ 时， γ 要设为一个常数， $0 < \gamma \leq 1$ ，而不是由式(23)计算获得)。

图 2 说明每次迭代时计算获得的 γ 使网络的能量变化比 γ 设为常数每次迭代变化最大，使网络收敛更快。实验也证明了本文方法中的 γ 在相同条件下能以更少的迭代次数收敛。

4 结论

通过分析文献[6]中的算法，一种改进基于连续状态变化函数的 Hopfield 网络模型用于图像复原，理论分析和实验结果表明此方法得到的复原图像信噪比显著提高。并且新算法对阈值函数的系数 γ 每次迭代通过计算获得，使得 Hopfield 网络的收敛速度更快，大大减少了图像复原的计算时间。但是在处理大图像时依然需要消耗大量的时间，如果在处理大图像时采用并行算法的话，可以进一步节省计算时间，将此算法应用在三维显微光学切片图像处理，是笔者以后的重要工作。

参考文献

- 1 陶青川. 计算光学切片显微三维成像技术研究[D]. 成都: 四川大学, 2005.
- 2 Zhou Y T, Chellappa R, Vaid A, et al. Image Restoration Using a Neural Network[J]. IEEE Transactions on ASSP, 1988, 36(7): 1141-1151.
- 3 Castleman K R. Digital Image Processing[M]. [S.L.]: Electronic Industries Publishing Company, 1998-09.
- 4 Sun Y, Zhou M H. Reconstructing Binary Section from Few Projection[J]. Transaction of Shanghai Jiaotong Univ, 1993, 27(3).
- 5 Yu S Y, Sun Y. A New Algorithm for Bilevel Section Reconstruction and Its Neural Network Implementation[J]. Automation, 1994, 20(4).
- 6 Paik J K, Katsaggelos A K. Image Restoration Using a Modified Hopfield Network[J]. IEEE Transactions on Image Processing, 1992, 1(1): 49-63.
- 7 王 磊, 戚飞虎. 精确复原退化图像的连续 Hopfield 网络研究[J]. 上海交通大学学报, 1997, 31(12): 43-46.
- 8 韩玉兵, 吴乐南. 基于状态连续变化的 Hopfield 神经网络的图像复原[J]. 信号处理, 2004, 20(5): 431-435.
- 9 Aiyer S V B, Niranjana M, Fallside F. A Theoretical Investigation into the Performance of the Hopfield Model[J]. IEEE Trans. on Neural Network, 1990, 1(3): 204-215.
- 10 Gee A H, Prager R W. Polyhedral Combinatorics and Neural Network[J]. Neural Computation, 1994, 6(2): 161-180.
- 11 Maa C Y, Shanblatt M A. Linear and Quadratic Programming Neural Network Analysis[J]. IEEE Trans. on Neural Networks, 1992, 3(4): 580-594.
- 12 Yalmoz T, Altun H, Engin T J E. Comparison of Simulation Algorithms for the Hopfield Neural Network: An Application of Economic Dispatch[J]. Turkish Journal of Electrical Engineering & Computer Sciences, 2000, 8(1): 67-80.