文章编号:1674-7070(2010)01-0006-07

Li-Hopfield 神经网络用于汉字字符识别

张煜东[1]　吴乐南[1]　王水花[1]　闵锐[1,2]

摘要

提出了一种 Li-Hopfield 网络,解决了 Hopfield 神经网络能量函数存在积分项、振荡收敛、计算复杂等问题,并继承了 Hopfield 网络的结构与 Li 网络的优点.同时发现基于部首拆分的汉字字符识别可以忽视甚至利用该网络的虚假稳定点,从而使样本存储量降为传统 Hopfield 网络的 1.44%,表明该算法有效且快速.

关键词

Hopfield 神经网络;虚假稳定点;汉字字符识别

中图分类号 TN911.73

文献标志码 A

0　引言
Introduction

虽然可用 Hopfield 神经网络(HNN)求解 TSP 问题,但由于 HNN 容易收敛到局部极值,收敛过程震荡、计算复杂且存在积分项,使得应用受限[1].随后,Li 等[2]采用定义在一个超立方体内部的一阶线性常微分方程,当解位于该超立方体的边界时,系统性能上等价于 HNN.实验证实,Li 网络(LNN)计算简单、直接收敛且能量函数更精确[2].

鉴于 HNN 与 LNN 的等价性与 HNN 的流行性,本文提出一种快速求解 HNN 设置参数的算法,基本思想是首先构建性能等价的 LNN,通过奇异值分解与矩阵相乘等快速运算求得 LNN 的参数,然后将 LNN 的参数转化为 HNN 的参数,最后得到想要的 HNN.

这样得到的 Li-Hopfield 网络虽然既保留了 HNN 的结构,又保留了 LNN 的优点,但仍存在不稳定平衡点(Unstable Equilibrium Point, UEP)[3]及设计者不希望的虚假稳定点(Spurious Stable Points, SSP)[4],使得研究者几乎抛弃该网络.

UEP 对网络影响不大,但 SSP 对大多数"模式联想"问题却是致命的[5].因此,学者们往往希望减少 SSP 的数目.然而在某些情况下,SSP 反而非常有用.例如汉字的两个部首(Indexing Component)组合成一个汉字,正是 SSP 在汉字空间中的体现.在同等条件下,它仅有传统的汉字字符识别网络 1.44% 的存储量,但识别效果令人满意.

1　Li-Hopfield 网络
Li-Hopfield network

1.1　背景

一般而言,神经网络的作用不外乎 3 类:函数逼近、数据分类、数据聚类[6].有些特殊的神经网络能够起到特殊的作用,例如脉冲耦合神经网络主要用于特征提取.

同样,HNN 与 LNN 属于特殊神经网络,作用是"模式联想(Pattern Recall)",也称"数据归类"[7].即存储设计者预先设定好的数据中心,使每一个数据中心变成一个吸引子.网络运行时,将给定的输入向量收敛到最匹配的某一吸引子,即得到对应的数据中心.

将"模式联想"与"数据分类"、"数据聚类"的区别列于表 1.可见,

收稿日期 2009-09-14

资助项目 国家自然科学基金(60872075);高等学校科技创新工程重大项目培育项目(706028);江苏省自然科学基金(BK2007103);东南大学优秀博士学位论文基金(YBJJ0908)

作者简介

张煜东,男,博士生,研究方向为图像识别. zhangyudongnuaa@gmail.com

1　东南大学　信息科学与工程学院,南京,210096

2　解放军理工大学　理学院,南京,210096

表 1　3 种不同功能的网络

Table 1　3 Types of neural networks

网络	作用	训练需要数据	训练时间
BP 网络,RBF 网络	数据分类(有监督)	输入向量、目标向量	长
竞争神经网络	数据聚类(无监督)	输入向量、类数	长
Hopfield 网络(HNN)	数据归类(模式联想)	类数、类中心位置	无

模式联想需要预先知道类数与对应的数据中心,这限制了它的使用范围. 但是,在"含噪字符识别"这个领域,每一个字符与对应受污染的字符集组成一类,而无污染的字符就是类中心,这恰好满足了必须预知类数与类中心的条件. 因此,HNN 适用于字符识别.

1.2　Hopfield 神经网络(HNN)

1.2.1　神经元模型

Hopfield 神经元可以按照如图 1 所示的电路实现,假设不存在输入电容与输入电阻,则非线性放大器的输入输出关系为

$$v_i = g_i(\lambda u_i) = \frac{2}{\pi}\arctan\left(\frac{\pi}{2}\lambda u_i\right). \tag{1}$$

式中:u_i、v_i 是输入与输出;参数 λ 是非线性放大器的增益,并假设其响应时间为 0.

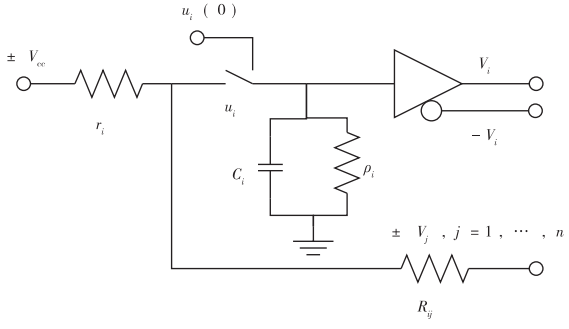


图 1　Hopfield 神经元的实现

Fig. 1　Realization of neuromime of Hopfield

n 个 Hopfield 神经元组成 HNN,神经元之间彼此相连. 图 1 中:V_{cc}/r_i 是第 i 个神经元的输入电流;$u_i(0)$是初始条件;R_{ij}表示从第 j 个至第 i 个神经元之间的权值电阻.

1.2.2　方程形式

上述 Hopfield 神经元可用如下微分方程表示

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \frac{1}{C_i}\left\{\sum_{j=1}^{n}\frac{\pm v_j}{R_{ij}} - \left(\frac{1}{\rho_i}+\frac{1}{r_i}+\sum_{j=1}^{n}\frac{1}{R_{ij}}\right)u_i + \frac{\pm V_{cc}}{r_i}\right\}. \tag{2}$$

定义如下符号,令

$$T_{ij} = \begin{cases} 1/R_{ij}, & \text{若 } R_{ij} \text{ 和 } V_j \text{ 相关;} \\ -1/R_{ij}, & \text{若 } R_{ij} \text{ 和 } -V_j \text{ 相关.} \end{cases} \tag{3}$$

$$R_i = \frac{1}{\rho_i}+\frac{1}{r_i}+\sum_{j=1}^{n}\frac{1}{R_{ij}}. \tag{4}$$

$$I_i = \frac{\pm V_{cc}}{r_i}. \tag{5}$$

因此,式(2)可以写成如下形式:

$$C_i\left(\frac{\mathrm{d}u_i}{\mathrm{d}t}\right) = \sum_{j=1}^{n}T_{ij}v_j - \frac{u_i}{R_i} + I_i. \tag{6}$$

这就是 Hopfield 模型,它最大的优势在于能够较方便地采用电路器件实现.

1.2.3　缺陷

然而,Hopfield 模型存在下述缺陷.

1) 输入向量 $\boldsymbol{u} = (u_1,\cdots,u_n)^{\mathrm{T}}$ 一般取值较大,归一化会对电路造成影响,不归一化则会导致网络性能大幅下降.

2) 当改动 R_{ij}来跟踪对应的 T_{ij}时,R_i 的值也会跟着改变,这是不希望的.

3) 神经网络的平衡点对应着如下方程的解:

$$\sum_{j=1}^{n}T_{ij}g(\lambda u_j) - \frac{u_i}{R_i} + I_i = 0, \quad i = 1,\cdots,n. \tag{7}$$

然而上述方程不易求解. 因此网络设计困难,需要专业的计算软件加以辅助.

4) 神经网络的能量函数

$$E(\boldsymbol{v}) = -\frac{1}{2}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{v} - \boldsymbol{v}^{\mathrm{T}}\boldsymbol{I} + \frac{1}{\lambda}\sum_{i=1}^{n}\int_0^{v_i}\frac{1}{R_i}g_i^{-1}(\boldsymbol{\rho})\mathrm{d}\boldsymbol{\rho}. \tag{8}$$

为计算方便,一般将式(8)中右边第 3 项忽略,因此导致平衡点的位置不够精确.

5) 网络收敛时存在严重的振荡.

1.3　Li 神经网络(LNN)

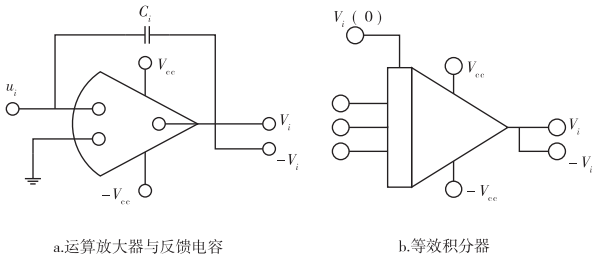
1.3.1 运算放大器

为了解决上述缺陷,Li 提出用运算放大器来替代传统的非线性放大器[8]. 运算放大器的输入输出关系如下:

$$v_i = \begin{cases} V_{cc}, & u_i > V_{cc}/\lambda; \\ \lambda u_i, & -V_{cc}/\lambda \leqslant u_i \leqslant V_{cc}/\lambda; \\ -V_{cc}, & u_i < -V_{cc}/\lambda. \end{cases} \tag{9}$$

式中,λ 是运算放大器的增益. 若第 i 个运放与反馈

电容 C_i 结合,则可得如图 2 所示的等效积分器.



a.运算放大器与反馈电容　　b.等效积分器

图 2　运算放大器

Fig. 2　Operational amplifier

1.3.2　Li 神经元

Li 神经元如图 3 所示. 图中: $\pm V_{cc}/r_i$ 表示第 i 个积分器的输入电流; $v_i(0)$ 是初始条件; R_{ij} 表示从第 j 个神经元输出至第 i 个神经元输入之间的权值电阻.

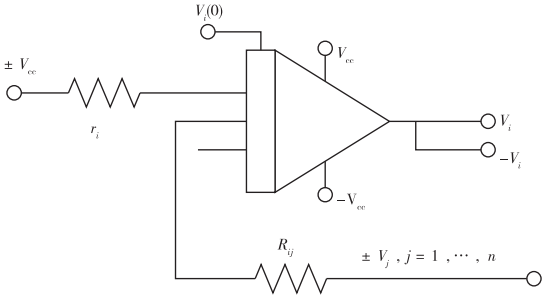


图 3　Li 神经元的实现

Fig. 3　Realization of neuromime of Li

1.3.3　方程形式

将 V_{cc} 看作 1,则矢量 $\boldsymbol{v}=(v_1,\cdots,v_n)^{\mathrm{T}}$ 的任一分量 v_i 的值均在 -1 与 1 之间,图 3 对应的系统可用如下带约束条件的常微分方程表示

$$\frac{\mathrm{d}v_i}{\mathrm{d}t}=\frac{1}{C_i}\left\{\sum_{j=1}^{n}\frac{\pm v_j}{R_{ij}}+\frac{\pm V_{cc}}{r_i}\right\},\quad i=1,\cdots,n;$$
$$-1<v_i<1,\quad i=1,\cdots,n. \tag{10}$$

若令 $C_i=1$, $\boldsymbol{x}=(x_1,\cdots,x_n)^{\mathrm{T}}=\boldsymbol{v}$, $\boldsymbol{T}$ 与 $\boldsymbol{I}$ 的设置同(3)和(5),则系统可简化为

$$\mathrm{d}\boldsymbol{x}/\mathrm{d}t=\boldsymbol{T}\boldsymbol{x}+\boldsymbol{I},\quad -1<x_i<1(i=1,\cdots,n). \tag{11}$$

1.3.4　优势

LNN 与传统 HNN 结构相似,但克服了 HNN 的内在缺点. 最重要的优势是能量函数为

$$E(\boldsymbol{x})=-(1/2)\boldsymbol{x}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{x}-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{I}. \tag{12}$$

其次,已证明 LNN 收敛过程中不会出现 HNN 的振荡现象[9]. 另外,实践表明,增益较大的传统放大器容易失稳,而同样条件下的高增益运算放大器可以保持稳定.

LNN 与传统 HNN 的比较可见表 2.

表 2　HNN 与 LNN 比较

Table 2　Comparison between HNN and LNN

	HNN	LNN
能量函数	二次项 + 积分项	二次项
收敛过程	振荡收敛	直接收敛
收敛精度	低	高
参数设置	复杂	简单
计算模块	解微分方程	SVD、矩阵相乘

1.4　LNN 的参数设置

1.4.1　Hopfield 网络参数设置

如何设置网络参数,传统的 HNN 需要求解式(7),实际中会面临下述 4 个问题:

1) 该方程是超越方程,必须具备较好的数学知识;

2) 必须采用专业的数学软件;

3) 方程系数往往奇异,采集器的量化误差、计算过程的舍入误差等都会造成解的波动;

4) 如果采用简化形式,则虽然求解较快,但是得到的网络设置与精确解存在误差.

1.4.2　Li 网络参数设置

LNN 只需简单的 4 步即可得到. 假设在 n 维空间中,存在 m 个期望类中心向量 $\boldsymbol{\alpha}=[\boldsymbol{\alpha}_1,\cdots,\boldsymbol{\alpha}_m]$,希望构造后得到的网络对这 m 个点是渐进逼近的.

步骤 1　计算 $\boldsymbol{Y}$ 矩阵如下,大小为 $n\times(m-1)$.

$$\boldsymbol{Y}=[\boldsymbol{\alpha}_1-\boldsymbol{\alpha}_m,\boldsymbol{\alpha}_2-\boldsymbol{\alpha}_m,\cdots,\boldsymbol{\alpha}_{m-1}-\boldsymbol{\alpha}_m]. \tag{13}$$

步骤 2　对 $\boldsymbol{Y}$ 作奇异值分解,得矩阵 $\boldsymbol{U}$、$\boldsymbol{V}$、$\boldsymbol{\Sigma}$,满足 $\boldsymbol{Y}=\boldsymbol{U}\boldsymbol{V}\boldsymbol{\Sigma}$. $\boldsymbol{U}$、$\boldsymbol{V}$ 都是酉矩阵, $\boldsymbol{\Sigma}$ 是对角阵. 记

$$\boldsymbol{Y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_{m-1}],$$
$$\boldsymbol{U}=[\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n],$$
$$k=\dim(\mathrm{span}(\boldsymbol{Y})). \tag{14}$$

这里, $\mathrm{span}(\boldsymbol{Y})$ 是指用 $\boldsymbol{Y}$ 的每一个列向量生成新的空间, dim 指维数. 显然, $k=\mathrm{rank}(\boldsymbol{\Sigma})$. 因此, $\{\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_k\}$ 是 $\mathrm{span}(\boldsymbol{Y})$ 的一个正交基底, $\{\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n\}$ 是 $\mathbf{R}^n$ 空间的一个正交基底.

步骤 3　计算 $\boldsymbol{T}^+$、$\boldsymbol{T}^-$ 矩阵如下:

$$\boldsymbol{T}^+=\sum_{i=1}^{k}\boldsymbol{u}_i\boldsymbol{u}_i^{\mathrm{T}},\quad \boldsymbol{T}^-=\sum_{i=k+1}^{n}\boldsymbol{u}_i\boldsymbol{u}_i^{\mathrm{T}}. \tag{15}$$

步骤 4　选取参数 τ,计算

$$\boldsymbol{T}_\tau = \boldsymbol{T}^+ - \tau\boldsymbol{T}^-, \quad \boldsymbol{I}_\tau = \boldsymbol{\alpha}_m - \boldsymbol{T}_\tau\boldsymbol{\alpha}_m. \tag{16}$$

需要注意的是,尽管 LNN 也不能摆脱 UEP 与 SSP 的干扰,但是通过选取足够大的 τ 值,可以保证设计者希望的吸引子的吸引力足够大,而虚假吸引子的吸引力尽量小.

可见,LNN 参数设置较为简单,只需一个 SVD 模块与一个矩阵相乘模块即可,便于用电路实现.

1.4.3 基于 Li 的 Hopfield 网络快速设置

由于 LNN 参数设置特别便捷,且 LNN 与 HNN 一一对应. 因此,对于 HNN,可以按图 4 所示变换.

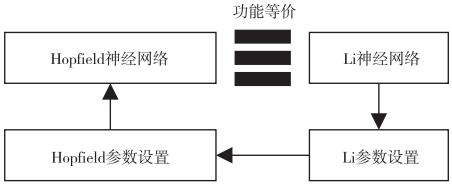


图 4 参数设置

Fig. 4 The setting of parameters

首先构建同样性能的 LNN,接着求出其对应的 LNN 的参数,即可得到 HNN 的参数,最后即可构建出期望的 HNN.

$$\boldsymbol{\Phi}_\tau = \boldsymbol{U}\begin{bmatrix} \mathrm{e}^{h}\boldsymbol{I}_k & 0 \\ 0 & \mathrm{e}^{-\tau h}\boldsymbol{I}_{n-k} \end{bmatrix}\boldsymbol{U}^{\mathrm{T}},$$

$$\boldsymbol{\Gamma}_\tau = \boldsymbol{U}\begin{bmatrix} (\mathrm{e}^{h}-1)\boldsymbol{I}_k & 0 \\ 0 & (\mathrm{e}^{-\tau h}-1)\boldsymbol{I}_{n-k}/(-\tau) \end{bmatrix}\boldsymbol{U}^{\mathrm{T}}. \tag{17}$$

式中,h 是连续空间的采样间隔. 这里,$\boldsymbol{\Phi}_\tau$、$\boldsymbol{\Gamma}_\tau$ 就是 HNN 的参数.

2 实验

Experiment

2.1 实验 1——LNN 优于 HNN

2.1.1 模式联想

假设数据中心为[1,-1]、[-1,1],则构建满足上述条件的 LNN 需要 2 个神经元,网络参数设置如下:

$$\boldsymbol{LW} = \begin{bmatrix} 1.045\,3 & -0.304\,5 \\ -0.304\,5 & 1.045\,3 \end{bmatrix}, \tag{18}$$

偏置 $b=0$. 随机产生 50 个输入向量送入网络,τ 设为 1,h 设为 0.3,迭代步数设为 20,结果见图 5. 图中实心点表示初始值,空心点表示每次迭代后的输出,可见,经过 20 代之后,所有输入均归类到左上角(-1,1)与右下角(1,-1).

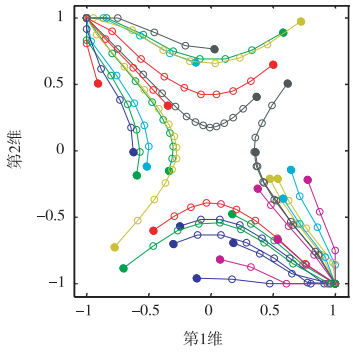


图 5 Li-Hopfield 网络收敛示意

Fig. 5 The convergence of Li-Hopfield network

2.1.2 不稳定平衡点

数据空间中存在许多 UEP,仍以上述 HNN 为例,将输入向量设为到点(-1,1)与(1,-1)距离相等,即位于两个实际类中心吸引区域的分界面,则归类结果示于图 6. 可见,此时网络收敛到另一个虚假类(Spurious Class)的中心(0,0).

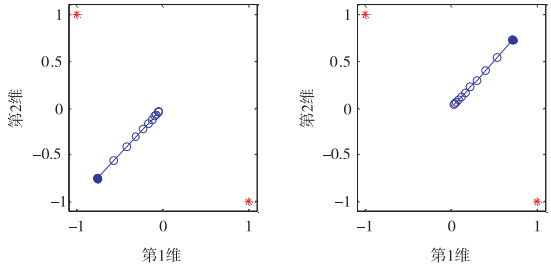


图 6 UEP 的平衡性

Fig. 6 Equilibrium of UEP

需要注意的是,之所以称其为“UEP”,是因为其“平衡性”仅在分界面上存在,对于不在分界面上的点,其吸引力变得非常弱(通过对 τ 的设置),表现出“不稳定性”. 此时若数据存在微小的扰动,则网络会迅速偏离(0,0),最终收敛到距两个数据中心最近的一个.

例如,若网络输入为[0.800 1,0.777 9],τ 设为 1 000,h 设为 0.5,则网络输出如图 7 所示. 网络在一开始迅速接近(0,0)后,最终被(1,-1)吸引过去.

UEP 对网络性能几乎无影响,这是因为:

1) 位置处于不同类中心吸引区域的边界处,例如上例中处于直线 $y=x$ 上,此时到(-1,1)与(1,-1)等距,原理上来说也无法正确归类;

2) 实际输入样本总会受噪声污染,产生位于分

界面上的输入样本的概率为零.

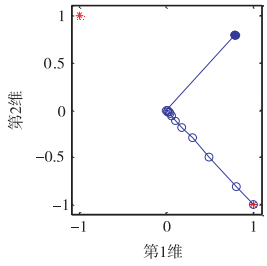


图 7　UEP 的不稳定性

Fig. 7　Instability of UEP

2.1.3　虚假稳定点

真正对网络产生不良影响的是 SSP,它是一种稳定的 UEP. 可理解为构建网络后生成的虚假类的类中心,具有很强的吸引力,且 SSP 的能量也是全局极小的,因此无法从能量上来区别. 由于无法解决 SSP 问题,HNN 逐渐被学者淡忘. Li 采用了新的网络结构后,证明其 SSP 的个数远远小于同样性能 HNN 的 SSP 个数,但仍然无法彻底消除 SSP.

目前已经证明的最好结果是,当 τ 足够大时,HNN 的渐进稳定点集合为期望类中心点的"仿射空间" $\mathrm{Affine}(\boldsymbol{\alpha}) = \{\sum \omega_i x_i : \sum \omega_i = 1\}$ 与"超立方体" $\boldsymbol{B}^n = \{x \in \mathbf{R}^n : x_i = 1 \text{ or } -1, i = 1, \cdots, n\}$ 的交集,即 $\mathrm{Affine}(\boldsymbol{\alpha}) \cap \boldsymbol{B}^n$.

这就是 HNN 的"仿射能力",它能够将预置的数据中心自动进行仿射变换,那些结果位于超立方体边界上的点,就会自动保留下来,成为一个虚假类的中心. 目前学者们致力于如何消除 HNN 的仿射能力.

例如,在 5 维空间中,有如下 4 个期望中心,分别是

$$\boldsymbol{\alpha}_1 = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \end{bmatrix}, \quad \boldsymbol{\alpha}_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \\ -1 \end{bmatrix},$$

$$\boldsymbol{\alpha}_3 = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{\alpha}_4 = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{19}$$

然而,当训练好网络之后,发现存在下列 2 个 SSP:

$$\boldsymbol{\beta}_1 = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}; \quad \boldsymbol{\beta}_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{20}$$

显然,$\boldsymbol{\beta}_1 = \boldsymbol{\alpha} \times [1 \quad -1 \quad 1 \quad 0]$, $\boldsymbol{\beta}_2 = \boldsymbol{\alpha} \times [-1 \quad 1 \quad 0 \quad 1]$,表明 $\boldsymbol{\beta}$ 均属于 $\boldsymbol{\alpha}$ 的仿射空间,同时 $\boldsymbol{\beta}$ 的元素取值均为 ±1,表明属于正立方体边界 $\boldsymbol{B}^5$. 因此,$\boldsymbol{\beta}$ 渐进稳定. 此时采用该网络实现数据联想,若输入数据接近 $\boldsymbol{\beta}$,则网络最终收敛到 $\boldsymbol{\beta}$,造成误归类.

需要注意的是,SSP 与 UEP 一样,都处于两个数据中心的分界面上. 但是 UEP 在两个数据中心的连线方向上是不稳定的,微小噪声就会使其最终接近某一个中心. 而 SSP 是稳定的,它能吸附附近的数据,且吸引力与给定数据中心点的吸引力一样大.

然而,通过研究发现,SSP 可以化害为利,例如用于汉字识别. 因为 SSP 可以视作已知类中心的仿射空间(权值和为 1 的线性组合),而同样地,汉字可以视作一种部首的组合(左偏旁 + 右偏旁,或上偏旁 + 下偏旁). 因此,可以认为 HNN 不仅存储了期望数据,而且还可以存储期望数据的组合. 假设网络存储了部首"弓"与"长",那么即使网络输入"张",也能识别出来. 这是别的网络所不具备的能力.

2.2　实验 2——基于部首的字符识别

目前世界上的文字有 5 000 多种,其中只有汉字有部首,这也体现了汉字之美. 对汉字采用 HNN 识别可以大幅减少训练样本量. 传统方法是将每个汉字作为一个记忆数据,但是鉴于本文讨论的 SSP,可以将汉字的部首作为记忆数据,其好处在于,只需要用 300 个部首(201 个主部首,99 个附型部首)[10],就可以表达20 902个常用汉字[11]. 换言之,传统的字符识别网络需要记住20 902个汉字数据,而采用基于 SSP 的组合方法,只需记住 300 个部首数据即可,只需传统网络 1.44% 的存储量.

2.2.1　预处理

为了简便,假设网络训练集中有 20 个部首,示于图 8. 规定黑色表示 −1,白色表示 1. 训练时引入 $\boldsymbol{\alpha}_{21}$ = 全白图像,表示"没有字符";以及 $\boldsymbol{\alpha}_{22}$ = 全黑图像,表示"噪声大到无法识别". 测试字符选取"计"与"亩",用 $\boldsymbol{P}_1$ 与 $\boldsymbol{P}_2$ 表示. 可见

$$\boldsymbol{P}_1 = \frac{2}{3}\boldsymbol{\alpha}_5 + \frac{2}{3}\boldsymbol{\alpha}_9 - \frac{1}{3}\boldsymbol{\alpha}_{21}, \tag{21}$$

$$\boldsymbol{P}_2 = \frac{2}{3}\boldsymbol{\alpha}_{13} + \frac{2}{3}\boldsymbol{\alpha}_{20} - \frac{1}{3}\boldsymbol{\alpha}_{21}. \tag{22}$$

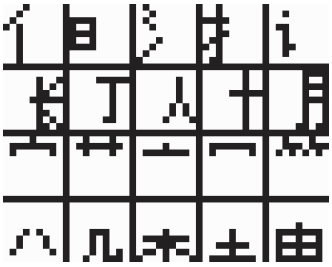


图 8　20 个简单部首

Fig. 8　20 simple radicals

由于系数满足 2/3 + 2/3 − 1/3 = 1,因此 $\boldsymbol{P}_1$、$\boldsymbol{P}_2$ 属于 $\boldsymbol{\alpha}$ 的仿射空间,同时像素由黑白二色组成(−1、1),因此 $\boldsymbol{P}_1$、$\boldsymbol{P}_2 \in \mathrm{Affine}(\boldsymbol{\alpha}) \cap \boldsymbol{B}^n$,是 SSP.

2.2.2　识别过程

由于 HNN 权值无需通过训练过程逐渐更新,因此直接按照本文方法利用训练数据构造网络即可,在 Matlab R2009a 仿真显示,全过程仅用0.000 2 s.

“计”原始图像的像素分布在[−1 1]区间,如下式所示.

$$\boldsymbol{P}_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \end{bmatrix}. \quad (23)$$

为了展现识别效果,对其置[−2 2]均匀分布的随机噪声,得到结果如图 9 所示.

$$\begin{bmatrix} 1.982\,2 & -0.714\,2 & 0.8242 & 2.045\,7 & 0.352\,4 & 1.829\,0 & -0.568\,5 & -0.794\,7 & -0.530\,0 \\ 1.945\,1 & -0.030\,1 & -2.593\,3 & 1.524\,3 & 0.175\,9 & 2.125\,5 & -0.035\,0 & -0.708\,5 & 1.562\,9 \\ 1.247\,4 & -0.785\,0 & 2.981\,6 & -0.640\,4 & 1.985\,3 & 0.151\,9 & -2.580\,7 & -0.645\,9 & 0.315\,3 \\ -0.263\,2 & -1.233\,1 & -1.671\,6 & -0.676\,6 & -2.958\,7 & -0.229\,9 & -2.488\,4 & 0.193\,4 & -0.384\,8 \\ 1.388\,8 & -0.946\,9 & -1.810\,6 & 2.109\,0 & -0.806\,2 & 1.226\,7 & -0.801\,8 & 2.772\,0 & 1.996\,5 \\ 0.199\,7 & 2.588\,8 & -2.751\,8 & 2.620\,5 & 1.671\,7 & 0.586\,1 & -1.05\,91 & 1.734\,9 & 1.332\,7 \\ -0.463\,5 & -0.213\,4 & -1.807\,0 & -0.864\,9 & 1.413\,9 & -0.753\,6 & 0.561\,9 & -0.471\,7 & 1.960\,1 \\ -0.149\,6 & -0.626\,5 & -2.814\,6 & -0.563\,4 & 1.104\,4 & 2.120\,7 & 0.195\,8 & 1.890\,9 & -0.060\,7 \\ 2.579\,8 & 0.229\,5 & 1.021\,7 & 2.303\,2 & 1.918\,8 & 0.350\,3 & -0.062\,6 & -0.558\,6 & 1.939\,8 \end{bmatrix}.$$

图 9　加噪图像的像素

Fig. 9　Pixels of images with noise

将其送入 HNN 内,发现第 18 步网络就已经收敛,即 $\boldsymbol{P}_1(18) = \boldsymbol{P}_1$. 收敛过程如图 10 所示.

再以“亩”为例,其污染图像的识别过程可如图 11 所示. 其在第 20 步收敛,即 $\boldsymbol{P}_2(20) = \boldsymbol{P}_2$.

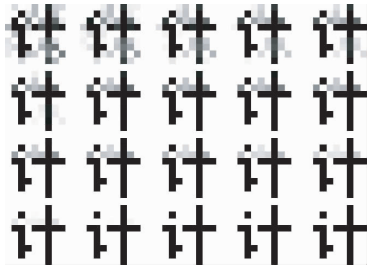


图 10　被污染的“计”图像识别过程

Fig. 10　Recognition of the corrupted Chinese character “计”

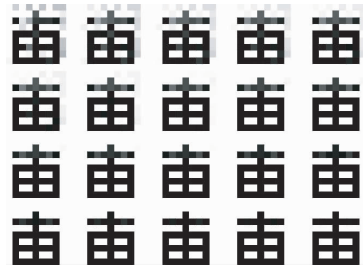


图 11　被污染的“亩”图像识别过程

Fig. 11　Recognition of the corrupted Chinese character “亩”

可见,尽管预先设置的数据中没有“计”和“亩”2 个字,然而,通过 HNN 的仿射能力,使得网络能够存储这 2 个字. 究其原因,在于“计”是部首“言”与“十”以及“全白图像”这 3 者生成的仿射空间. 换言之,本文利用了学者们曾想方设法去除的 SSP.

2.2.3　运行时间

HNN 不仅不需要训练,而且运行速度也非常快. 以本节构建的 HNN 为例,输入10 000幅随机 9 × 9 的汉字图像,采用 Matlab 运算,平均每幅图像的识别仅需0.015 747 s.

若用 C 语言实现,或者采用基于 GPU 的并行处理方式,则运算速度还能进一步提高. 若将其固化到硬件上,还能进一步提速.

3　结论

Conclusion

本文首先提出一种结合 Hopfield 与 Li 的新型神经网络,分析了其专用于“数据联想”的功能,并给出其优于 HNN 的 5 个原因. 接着,分析了其缺陷,即 SSP. 但是,对汉字字符识别而言,该缺点变成了优点,可以大幅减少存储数据. 最后,采用简单的部首结合方法对 SSP 的应用进行了验证. 表明基于 HNN 的汉字识别,主要优势体现在存储量仅有传统方法的 1.44%. 然而,真正实现一个基于 HNN 的汉字字

符识别系统,需要考虑更多的实际问题.

本文另一个贡献在于,证明了 HNN 的仿射能力在某些特殊情况下是有用的. 因此,今后的研究可以从如何消除 HNN 的 SSP(仿射能力),转移到寻找某些需要这一能力的问题上.

参考文献

References

[1] Wen U P, Lan K M, Shih H S. A review of Hopfield neural networks for solving mathematical programming problems[J]. European Journal of Operational Research, 2009, 198(3): 675-687

[2] Li J H, Michel A N, Porod W. Analysis and synthesis of a class of neural networks: linear systems operating on a closed hypercube [J]. IEEE Trans on Circuits and Systems, 1989, 36(11): 1405-1422

[3] Fu X L, Li X D. Global exponential stability and global attractivity of impulsive Hopfield neural networks with time delays[J]. Journal of Computational and Applied Mathematics, 2009, 231(1): 187-199

[4] Ma W B, Saito Y, Takeuchi Y. M-matrix structure and harmless delays in a Hopfield-type neural network[J]. Applied Mathematics Letters, 2009, 22(7): 1066-1070

[5] Ali M S, Balasubramaniam P. Stability analysis of uncertain fuzzy Hopfield neural networks with time delays[J]. Communications in Nonlinear Science and Numerical Simulation, 2009, 14(6): 2776-2783

[6] Udpin S, Niamsup P. Robust stability of discrete-time LPD neural networks with time-varying delay[J]. Communications in Nonlinear Science and Numerical Simulation, 2009, 14(11): 3914-3924

[7] Crook N, Goh W J, Hawarat M. Pattern recall in networks of chaotic neurons [J]. Biosystems, 2007, 87(2): 267-274

[8] Gokcen A, Cam U. MOS-C single amplifier biquads using the operational transresistance amplifier [J]. AEU-International Journal of Electronics and Communications, 2009, 63(8): 660-664

[9] Zhang Y P, Wang Q G. Stationary oscillation for high-order Hopfield neural networks with time delays and impulses[J]. Journal of Computational and Applied Mathematics, 2009, 231(1): 473-477

[10] 教育部国家语言文字工作委员会. 汉字部首表[S]. 北京:语文出版社,2009
State Commission on Language and Characters under Ministry of Education. Table of radicals of Chinese characters[S]. Beijing: Chinese Press, 2009

[11] 教育部国家语言文字工作委员会. GB13000.1 字符集汉字部首归部规范 [S]. 北京:语文出版社,2009
State Commission on Language and Characters under Ministry of Education. Classification standard of character radicals in GB13000.1 character set [S]. Beijing: Chinese Press, 2009

Li-Hopfield neural network used for Chinese character recognition

ZHANG Yudong[1] WU Lenan[1] WANG Shuihua[1] MIN Rui[1,2]

1 School of Information Science & Engineering, Southeast University, Nanjing 210096

2 Institute of Sciences, PLA University of Science and Technology, Nanjing 210096

Abstract In order to solve the problems in Hopfield neural networks, such as integration terms in energy function, convergence with oscillation, and complexity in computation, a novel Li-Hopfield neural network is proposed. It can not only solve the above-mentioned problems, but also inherit the structure of Hopfield neural network and the merits of Li neural network. Meanwhile, we find Chinese character recognition based on indexing components can ignore, even can utilize, the spurious stable points (SSP) in this network. Experiments demonstrate that the sample storage for this method is reduced to 1.44% of that for the traditional method. Thus, the proposed method is effective and fast.

Key words Hopfield neural network; spurious stable points; Chinese character recognition