

基于笔划特征和 MCLRNN 模型的联机手写藏文识别

王维兰¹, 陈万军²

WANG Wei-lan¹, CHEN Wan-jun²

1. 西北民族大学 中国民族信息技术研究院, 兰州 730030

2. 西安理工大学 信息科学系, 西安 710048

1. China Minorities Information Technology Institute, Northwest University for Nationalities, Lanzhou 730030, China

2. Department of Information Science, Xi'an University of Technology, Xi'an 710048, China

E-mail: wangweilan@xbmu.edu.cn

WANG Wei-lan, CHEN Wan-jun. MCLRNN model for online handwritten Tibetan character recognition based on stroke characteristics. *Computer Engineering and Applications*, 2008, 44(14): 91-93.

Abstract: This paper presents a new type of recurrent neural network called multi context layers recurrent neural network for online handwritten Tibetan character recognition. The new network has a multi context layer, which can keep more states in memory, to better describe stroke characteristics and spatial structure relationships between each stroke. The cross entropy rule and an efficient learning algorithm derived from the gradient descent method are adopted for training the network. The simulation experiment has achieved satisfying results.

Key words: recurrent neural network; online handwritten Tibetan character recognition; cross entropy

摘要:提出了一种新的多层联系子层递归神经网络(MCLRNN)模型并融合藏文字的空间结构特征来进行联机手写藏文识别。改进后的网络结构具有多层联系子层来保留若干时刻的网络内部状态,从而可以更好地表征藏文字的各笔划特征以及笔划间的空间结构关系,同时,采用更适用于模式分类的交叉熵准则和改进的梯度下降算法来训练网络,加快了网络的收敛速度并增强其分类能力。仿真实验取得了令人满意的结果。

关键词:递归神经网络;联机手写藏文识别;交叉熵

DOI: 10.3778/j.issn.1002-8331.2008.14.025 **文章编号:** 1002-8331(2008)14-0091-03 **文献标识码:** A **中图分类号:** TP391

1 引言

手写识别是指将书写者的笔迹由空间图形表示方式转变为其语义符号表示方式的过程。相对于脱机手写识别,联机手写识别则要求书写的过程与转换过程同步。由于联机手写识别的输入符合人的自然手写方式,有着巨大的学术研究和实用价值,已引起越来越多的研究者的广泛关注,使其得以不断的发展^[1,2]。

藏文是世界上优秀的成熟文字,要在计算机上使用藏文,首先遇到的问题就是如何有效地把藏文输入到计算机中。近年来,已经成功地研究和开发了藏文智能输入系统^[3],但如何快速高效方便的将藏文输入计算机仍然是一个难题。手写输入是各种输入方式中最直接最自然的输入方式,因此,本文针对藏文的联机手写识别做了一些初探性的工作。

目前,联机手写识别方法主要分为三类:统计法、结构法和统计与结构相结合的方法^[2],如弹性匹配法^[4]、时延神经网络^[5]和隐马尔科夫模型(HMM)^[6]等。基于结构的方法虽然

具有较强的描述字符二维空间结构的能力,但也有它自身的弱点,如抗干扰能力差,不够稳定,要求基元的分割、抽取和识别都能保证较高的准确率,不易适应手写字符的多变性。相比于基于结构的方法,基于统计的方法具有更强的鲁棒性和更高的稳定性。然而,一般的统计方法,如曲线/特征匹配法、马尔科夫模型和时延神经网络方法均有各自的缺点。曲线/特征匹配法由于计算量大而不适合于大字符集语种的手写识别。HMM首先是在语音识别方面取得了很大成功,后来被应用到序列学习问题,如联机手写识别。要得到一个稳定可靠的HMM,需要选择合适的状态数和模型的初始参数的估计值以及各状态之间的转移概率,通常这是一件很困难的事。基于反向传播算法训练的时延神经网络虽然要求的参数比较少,但它的固定时间窗使得它不能处理动态长度的序列。另一类神经网络是递归神经网络(Recurrent Neural Network, RNN)^[7],它能够成功处理如形式语言学习问题等这样的时序问题,同时,递归神经网络还克服了时延神经网络固定时间窗

基金项目:甘肃省科技攻关项目(the Key Technologies R&D Program of Gansu(Province), China under Grant No. 2GS064-A52-035-04); 国家人事部 2007 年度留学人员科技活动项目择优资助(No. 国人厅发[2007]170 号)。

作者简介:王维兰(1961-),女,教授,研究方向:图像处理、模式识别、藏文信息处理等;陈万军(1979-),男,硕士,助教,主要研究方向:模式识别和数字图像处理、中文信息处理。

收稿日期: 2007-09-03 **修回日期:** 2007-10-29

的缺点,并能用于作为识别未知模式的分类器。通常有两种类型的递归神经网络,它们是 Elman 网络和全递归神经网络。Elman 网络主要缺点是它的联系单元的规模较小,只能保存前一时刻隐层节点的状态,不具有长时间的记忆功能。全递归神经网络由于其学习算法 - RTRL (Real Time Recurrent Learning) 计算复杂度大而使其不适合于联机手写识别。

综合考虑以上方法的优缺点并结合藏文字的笔划数相对较少、长弧线笔划较多的特点和实际书写习惯,提出了一种新的以笔划为结构基元的多层联系子层递归神经网络 (Multi Context Layers Recurrent Neural Network, MCLRNN) 模型来进行联机手写藏文识别。该网络结构具有多层联系子层来保留若干时刻的网络内部状态,从而可以更好地表征藏文字的各笔划特征以及笔划间的空间结构关系,同时,采用更适用于模式分类的交叉熵准则和改进的梯度下降算法来训练网络,在加快网络收敛的同时最大化的减少输出误差并增强其分类能力。

2 笔划结构基元的提取

由于藏文字(也称为字丁)的笔划数较少,大多数字丁的笔划数均在 6 画左右,极少数的笔划数超过 10 画。同时,长而弯曲度较大的弧线笔划也是字丁的一个特点,如图 1 所示为部分采集的样本字样例。在结构基元的描述时采用笔划类型以及笔划间的位置关系,而不是采用构成长弧线笔划的不稳定的笔段,有利于更稳定的描述字丁的空间结构信息。

笔划的提取包括预处理、笔段的提出与合并及笔划的识别。

在实际书写时,每一个字丁 T 均可表示为由起笔与落笔分割开的笔划序列的集合,即 $T = \{s_1, s_2, \dots, s_m\}$, m 为笔划数。预处理包括笔划平直处、拐角处、起始处和终止处及量化噪声的消除和基于字丁线密度等分的大小规范化处理,其详细内容可参考文献 [8]。预处理后的字丁表示为 $T' = \{s'_1, s'_2, \dots, s'_m\}$, 结果如图 2 所示。

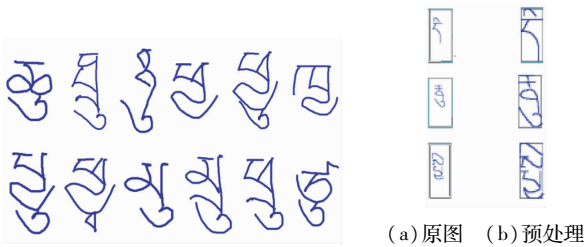


图1 部分样本字样例

图2 预处理

字丁笔段的抽取,关键在于笔划中转折点的抽取,也即前后方向变化较大,夹角较小的点的抽取。通过有约束的内接三角形来度量曲线的曲率,并采用二次搜索和角度阈值来确定转折点的准确位置。同时,为了尽可能准确的检测出折点,并最大可能避免冗余折点的出现,对检测出来的笔段再次进行了合并处理。至此,每一笔划 s'_i 可用笔段序列表示为: $s'_i = \{s'_{i1}, s'_{i2}, \dots, s'_{in}\}$, 其中 $1 \leq i \leq m, n$ 为笔段数。

笔划的识别即为笔段序列 s'_i 的识别,为此,在平面内进一步定义字丁笔段的方向属性 θ , 根据 θ 的取值不同将其量化为 8 个方向 $d_j, d_j \in \{0, 1, \dots, 7\}$ 。从而,每一个笔段序列 s'_i 可用 8 方向码表示为 $s'_i = \{d_{i1}, d_{i2}, \dots, d_{in}\}$ 。同时,根据字丁的笔划构成特点和实际书写习惯,定义了 16 种标准笔划,从而,笔划

的识别就转变为方向码串的匹配。考虑到实际书写时构成笔划的笔段数不稳定的情况,采用动态规划来进行串的匹配以消除由于笔段数的不等及方向码类型的不匹配而引起的误差。

3 MCLRNN 网络及其学习算法

3.1 MCLRNN 网络

为了使网络能够保存更多的过程状态信息和更强的动态特性,本文在 Elman 网络的基础上提出了 MCLRNN 网络^[9], 结构如图 3 所示。

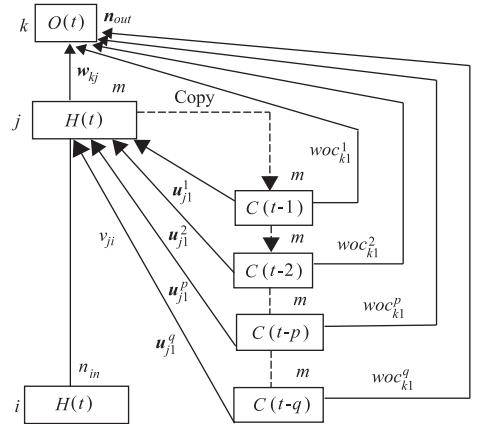


图3 多层联系子层 RNN 网络

增加多层反馈子层可以保留若干个时刻的网络内部状态,以便更好地表征过程对象的内部特征。联系子层 i 保留第 $(t-i)$ 时刻的隐含层节点的输出,所以在网络中可以保存 q 个时刻的隐含层节点的输出。在第 t 个时刻,第 i 联系子层的内容将送入第 $(i+1)$ 联系子层 ($i=1, \dots, q$), 而联系子层 1 的内容将被隐含层单元的输出而取代。各联系子层可进行短时的记忆,以保存一些必要的网络内部信息以表征过程的一些空间状态变量。同时,在网络中增加联系子层到输出层的前馈连接,这样可以大大减少网络中隐含层神经元的数目。

采用 softmax 函数和 logistic 函数分别作为输出层和隐含层的激励函数:

$$f_{SM}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}} \quad f(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

则隐含层的输入与输出分别表示为:

$$\tilde{h}_j(t) = \sum_{i=1}^{n_{in}} I_i(t)v_{ji}(t) + \sum_{p=1}^{n_{con}} \sum_{l=1}^m C_l(t-p)u_{jl}^p(t) \quad (2)$$

$$H_j(t) = f(\tilde{h}_j(t)) \quad (3)$$

其中

$$C_j(t-p) = C_j(t-p+1), p=2, \dots, q \quad (4)$$

$$C_j(t-1) = H_j(t) \quad (5)$$

网络输出层的输入与输出分别为:

$$\tilde{o}_k(t) = \sum_{j=1}^m H_j(t)w_{kj}(t) + \sum_{p=1}^{n_{con}} \sum_{l=1}^m C_l(t-p)w_{okl}^p(t) \quad (6)$$

$$O_k(t) = f_{SM}(\tilde{o}_k(t)) \quad (7)$$

3.2 交叉熵函数准则及学习算法

在 BP 算法中用误差平方和作为目标函数对分类问题并不是最合理的,误差平方和准则是在应有输出满足高斯分布

的条件下由最大似然原则导出的,而在分类问题中,输出并不满足高斯分布,而且传统 BP 算法收敛速度慢、产生局部极小点的缺陷也与采用误差平方和准则有关。本文选用更适用于模式分类的交叉熵准则^[10]和式(1)作为激励函数,应用于 MCLRNN 网络的训练。

最大似然方法是最优估计方法之一,以下就由最大似然原则导出交叉熵函数准则。

对于 N 个独立同分布的训练样本 $x(n), 1 \leq n \leq N, t(n)$ 为相应的目标值,其似然函数为

$$L = \prod_{n=1}^N p(x(n), t(n)) = \prod_{n=1}^N p(t(n) | x(n)) p(x(n)) \quad (8)$$

使似然函数 L 最大等价于使 L 的负对数最小,也就是式使(9)最小:

$$E = -\ln L = -\sum_{n=1}^N \ln p(t(n) | x(n)) - \sum_{n=1}^N \ln p(x(n)) \quad (9)$$

由于式(9)中的第二项($p(x)$)与网络参数无关,故在求最小时可省去。从而得到

$$E = -\ln L = -\sum_{n=1}^N \ln p(t(n) | x(n)) \quad (10)$$

设用作分类器的神经网络有 C 个输出,每个输出表示样本属于相应某一类的概率。令 $t_k(\cdot)$ 表示第 k 个神经元的期望输出,当 $x(n)$ 属于第 l 类(用 w_l 表示)时, $t_k(n) = \delta_{kl}$ 。给定输入 $x(n)$,第 k 个神经元的实际输出即是 $y_k = p(w_k | x(n))$,所以有

$$P(t(n) | x(n)) = \prod_{k=1}^C (y_k(n))^{t_k(n)} \quad (11)$$

将式(11)代入式(10),即得

$$E = -\sum_{n=1}^N \sum_{k=1}^C t_k(n) \ln y_k(n) \quad (12)$$

这就是输出 $y(x)$ 与目标 t 的分布间的交叉熵,当且仅当 $y_k(n) = t_k(n)$ 时,式(12)取最小值。

可以证明,基于交叉熵准则的 BP 算法与基于误差平方和准则的 BP 算法相比,对微小变化的反应灵敏度高,收敛速度更快,不易产生振荡,且易跳出局部极小点^[11];并且,交叉熵函数是从模式分类的角度推导出来的,因而更适于作为网络分类器的目标函数。

令 $O_k(t)$ 和 $d_k(t)$ 分别为网络输出层的第 k 个神经元在 t 时刻的实际输出和期望输出,则 t 时刻的网络交叉熵误差为:

$$E(t) = -\sum_{k=1}^{n_{out}} d_k(t) \ln O_k(t) \quad (13)$$

设网络从时间步 1 工作到时间步 T ,则每个周期的总误差函数为:

$$E_{total} = \sum_{t=1}^T E(t) \quad (14)$$

其中, T 为网络输出步数。

学习算法的目标是通过调节网络权值,使总误差函数 E_{total} 逐步减小。将 $E(t)$ 对连接权 $w_{kj}(t), woc_{kl}^p(t), u_{jl}^p(t)$ 和 $v_{ji}(t)$ 分别求偏导,由梯度下降法可得多层联系子层 RNN 网络的学习算法^[12]:

$$\Delta w_{kj}(t) = \eta_1 \frac{\partial E(t)}{\partial o_k(t)} H_j(t) \quad (15)$$

$$\Delta woc_{kl}^p(t) = \eta_2 \frac{\partial E(t)}{\partial o_k(t)} C_l(t-p) \quad (16)$$

$$\Delta u_{jl}^p(t) = -\eta_3 \sum_{j'=1}^m \left[\frac{\partial E(t)}{\partial \bar{h}_{j'}(t)} \frac{\partial H_{j'}(t)}{\partial u_{jl}^p(t)} + \sum_{p'=1}^{n_{con}} \sum_{l'=1}^m \frac{\partial E(t)}{\partial C_{l'}(t-p')} \delta_{j'l'} \frac{\partial H_{j'}(t-p')}{\partial u_{jl}^p(t)} \right] \quad (17)$$

$$\Delta v_{ji}(t) = -\eta_4 \sum_{j'=1}^m \left[\frac{\partial E(t)}{\partial \bar{h}_{j'}(t)} \frac{\partial H_{j'}(t)}{\partial v_{ji}(t)} + \sum_{p'=1}^{n_{con}} \sum_{l'=1}^m \frac{\partial E(t)}{\partial C_{l'}(t-p')} \delta_{j'l'} \frac{\partial H_{j'}(t-p')}{\partial v_{ji}(t)} \right] \quad (18)$$

其中, $\eta_1, \eta_2, \eta_3, \eta_4$ 分别是 $w_{kj}(t), woc_{kl}^p(t), u_{jl}^p(t), v_{ji}(t)$ 的学习步长; δ_{ab} 为 Kronecker 函数。

在式(15)~式(18)中

$$\frac{\partial E(t)}{\partial o_k(t)} = d_k(t) - O_k(t) \quad (19)$$

$$\frac{\partial E(t)}{\partial \bar{h}_{j'}(t)} = \sum_{k=1}^{n_{out}} \frac{\partial E(t)}{\partial o_k(t)} w_{kj'}(t) \quad (20)$$

$$\frac{\partial E(t)}{\partial C_{l'}(t-p)} = \sum_{k=1}^{n_{out}} \frac{\partial E(t)}{\partial o_k(t)} woc_{kl}^p(t) \quad (21)$$

$$\frac{\partial H_{j'}(t)}{\partial u_{jl}^p(t)} = f'(\bar{h}_{j'}(t)) \left[\delta_{jj'} \sum_{p''=1}^{n_{con}} \delta_{pp''} H_l(t-p'') + \sum_{p''=1}^{n_{con}} \sum_{j''=1}^m \sum_{l''=1}^m u_{j''l}^p(t) \delta_{j''j'} \frac{\partial H_{j''}(t-p'')}{\partial u_{jl}^p(t)} \right] \quad (22)$$

$$\frac{\partial H_{j'}(t)}{\partial v_{ji}(t)} = f'(\bar{h}_{j'}(t)) \left[\delta_{jj'} I_i(t) + \sum_{p'=1}^{n_{con}} \sum_{j'=1}^m \sum_{l'=1}^m u_{j'l}^p(t) \delta_{jj'} \frac{\partial H_{j'}(t-p')}{\partial v_{ji}(t)} \right] \quad (23)$$

4 联机手写藏文识别的实验结果

采用第 2 章介绍的特征描述方法提取出字丁的笔划特征序列 $F = \{f_0, \dots, f_{m-1}\}$,对每一个特征 $f \in F$ 采用以下 4 个属性进行描述: type(笔划类型), length(笔划长度), x -center 和 y -center(笔划几何中心的 x 坐标和 y 坐标)。将特征序列 F 作为神经网络的输入并采用前面给出的学习算法来评估 MCLRNN 模型在联机手写藏文识别中的应用。

实验中使用 60 套训练样本对 MCLRNN 模型进行训练,在 560 字丁字集上测试识别率。测试集为独立的 5 套测试样本,样本质量大致从工整书写到有约束的连笔书写变化。输入层与输出层的节点个数分别为 4 和 560,分别与特征描述的属性个数和字丁字集的个数相同。

表 1 给出了 MCLRNN 模型在 5 套测试样本上的前十选识别率。从表中可以看出,隐层节点数为 40 时,所有样本的平均识别率为 84.96%。当隐层节点数增加,识别率虽然在继续提高,但增幅不是特别大,仅仅提高了近 5.7%,随之而来的却是更复杂的网络结构和更长的学习时间。

表 1 多层联系子层 RNN 模型识别率 %

样本集	隐层节点数				
	40	60	80	100	120
样本集 1	87.18	90.66	89.99	92.52	93.23
样本集 2	83.94	86.34	88.14	87.96	88.79
样本集 3	86.21	87.79	89.64	91.13	91.35
样本集 4	84.91	88.02	89.58	90.29	90.48
样本集 5	82.57	85.71	88.46	89.44	89.67
平均识别率	84.96	87.70	89.16	90.27	90.70