DOI:10.16451/j.cnki.issn1003-6059.2011.01.011
第 24卷 第 1期 模式识别与人工智能 Vol.24 No.1
2011年 2月 PR&AI Feb 2011

基于 LSTM型 RNN的 CAPTCHA识别方法

张 亮 黄曙光 石昭祥 胡荣贵

(解放军电子工程学院 网络系 合肥 230037)

摘 要 全自动区分计算机和人的图灵测试(CAPTCHA)是一种基于人工智能难题的网络安全机制.研究 CAPTCHA的识别能够使其变得更加安全,并能促进一些人工智能难题的求解.文中首先对现有的 CAPTCHA识别方法进行总结和分析,然后提出一种基于长短时记忆(LSTM)型递归神经网络(RNN)进行识别的方法,并对 CAPTCHA识别中的特征提取问题进行研究.最后,为进一步提高 RNN的识别率,提出一种解码算法.实验结果表明,文中方法是有效的,灰度值对于 RNN是一种较好的特征,提出的解码算法能够取得较高的识别率,又有较低的时间复杂度.

关键词 人工智能,脱机文字识别,全自动的区分计算机和人的图灵测试(CAPTCHA),长短时记忆(LSTM)
中图法分类号 TP393.08

CAPTCHA Recognition Method Based on RNN of LSTM

ZHANG Liang HUANG Shu-Guang SHI Zhao-Xiang HU Rong-Gui
(Department of Network, PLA Electronic Engineering Institute, Hefei 230037)

ABSTRACT

Completely automated public turing test to tell computers and humans apart (CAPTCHA) is a kind of network security mechanism based on hard artificial problems. Study of recognition of CAPTCHA impels it to become more secure, and some hard atifical problems to be solved. Firstly, CAPTCHA recognition methods of state of the art are analyzed. Then, a recognition method is brought up based on recurrent neural network (RNN) which is composed by long short-term memory (LSTM) blocks. Thirdly, feature extraction for CAPTCHA recognition is studied. Finally, a decoding algorithm is proposed to improve the recognition rate. Experimental results show that the proposed recognition method is efficient. Gray value of images is proved to be a kind of good feature for RNN. Furthermore, the proposed decoding algorithm gets high recognition rates with low time complexity.

Key Words Artificial Intelligence, Offline Character Recognition, Completely Automated Public Turing Test To Tell Computers and Humans Apart (CAPTCHA), Long Short-Term Memory (LSTM)

收稿日期:2010-01-04 修回日期:2010-05-31
作者简介 张亮,男,1982年生,博士研究生,主要研究方向为图像处理与模式识别、人工智能、信息安全技术.E-mail:mathfun@163.com.黄曙光,男,1961年生,教授,博士生导师,主要研究方向为信息安全技术.石昭祥,男,1945年生,教授,博士生导师,主要研究方向为图像处理与模式识别、人工智能、信息安全技术.E-mail:zhxshi@gmail.com.胡荣贵,男,1966年生,教授,主要研究方向为信息安全技术.

1 引 言

传统的网络服务在设计时通常会假定它的使用者是自然人,例如电子邮件、网络论坛、银行交易等.但是没有人能够保证与服务器进行交互的就一定是个自然人,由此导致本该属于自然人使用的资源被非人类使用,大规模垃圾邮件发送机、论坛灌水机、游戏外挂等就是典型的机器人使用人类资源进行非法行为的例子,因此需要一种安全机制解决机器人滥用自然人资源的问题.全自动区分计算机和人的图灵测试(Completely Automated Public Turing Test to Tell Computers and Humans Apart, CAPTCHA)[1-2]就是一种这样的机制.

CAPTCHA是一种可视化的人机交互证明(Visual Human Interaction Proof, VHIP)[3],它来自于著名的图灵测试.CAPTCHA属于图灵测试的一种,但是它又与图灵测试有区别.裁判员在图灵测试中通常是人,而在实用的 CAPTCHA中,是另外一个机器人,也就是说由机器人来判断测试者是自然人还是机器人.自然人很容易通过 CAPTCHA,而机器人却很难通过.

CAPTCHA最早由 AltaVista的科学家 Broder等于 1997年投入使用,以解决 AltaVista搜索引擎被机器人滥用的问题.使用后,他们发现 AltaVista的滥用减少 95%.2000年,Yahoo寻求一种程序以阻止广告软件在聊天室里大量发布广告.Carnegie Mellon University的研究人员承接该项目,研制出 Gimpy类型的 CAPTCHA,满足当时的要求.目前 CAPTCHA已被广泛应用到互联网的各个领域,包括电子邮件、在线投票、网络论坛、网络博客、网上银行等,已成为互联网安全的一个标准防范措施[4].鉴于它良好的应用前景,目前很多学术机构以及商业公司等单位的研究人员都在对其进行研究.

目前 CAPTCHA的实现形式有文字识别型、邮件验证型、手机验证型、声音识别型等多种形式.由于文字识别型安全系数较高,可用性好,并且不依赖于除键盘和屏幕之外的其它硬件,得到广泛应用.这种类型 CAPTCHA在 Internet上的表现形式是,服务器向客户方发送一张图片,客户方对这张图片进行识别并且将识别结果返回给服务器.服务器根据客户方的识别结果正确与否判断客户方是否是自然人,从而决定是否向其提供某项服务.

与 RSA加密体系中利用大质数难以进行因数分解相类似,CAPTCHA主要利用目前人工智能领域一些公认的难题来实现(例如图像识别问题)[1-2].提出 CAPTCHA的研究人员认为如果基于某个人工智能难题的 CAPTCHA不能够被破解,那么说明找到一种有效区分机器人和人的方法,网络安全得到维护.如果提出的 CAPTCHA被破解,那么说明一个人工智能难题被解决,从而推动人工智能的发展,总之 CAPTCHA是一种双赢机制[1].本文研究 CAPTCHA识别的目的在于推动一些人工智能领域难题的求解.其次,由于 CAPTCHA是一种建立在人工智能问题上的加密协议[1],而在密码学领域,研究解密算法能够发现加密算法的缺陷,从而促使其变得更加安全,因此本文研究 CAPTCHA的识别同样有助于发现 CAPTCHA自身的缺陷,促使其变得更加安全.由于非粘着型 CAPTCHA已被实践证明非常容易被识别,目前的识别难点主要是粘着型 CAPTCHA,本文主要针对这类 CAPTCHA的识别进行研究.

2 现有的 CAPTCHA识别方法及分析

2.1 现有 CAPTCHA识别方法

目前针对文字型 CAPTCHA识别的主要方法有模板匹配方法[5]、BP神经网络方法[6-7]、SVM方法[7]等,假设我们不是图灵机而是人类的对手(Pretend We're Not a Turing Computer but a Human Antagonist, PWNTCHA)[8]是一个著名的基于模板匹配的 CAPTCHA识别器.PWNTCHA的工作过程可分为 4个阶段:模板库建立、图像预处理、图像分割、图像识别.模板库建立阶段主要是建立组成 CAPTCHA图像的各个字符的模板,这通过分析大量的 CAPTCHA图片进行实现.图像预处理阶段主要进行图像的灰度化、二值化和去除噪声.图像分割阶段主要是将各个待识别字符划分开来以供识别.在识别阶段,分割出来的待识别字符与模板库中各个模板进行匹配,取匹配程度最大的模板对应的字符作为识别结果.实验结果表明,PWNTCHA对 slashdot, phpbb和 linuxfr等很多网站早期使用的 CAPTCHA取得较高的识别率.

基于 BP神经网络方法和基于 SVM方法这两种方法的主要处理流程:图像预处理、图像分割、特征提取、分类器训练,分类器识别等.图像预处理和图像分割与 PWNTCHA的目的基本相同,只是预处理和分割的方法上存在不同而已.在特征提取阶段,根

据所使用的分类器不同,提取出相应的特征数据.在训练阶段,将特征数据和目标类别输入到分类器中进行训练.在识别阶段,将特征数据输入到分类器进行识别.

2.2 现有识别方法的分析

分析目前使用的识别方法可知,它们存在一个共同的阶段——图像分割.图像分割对于早期的CAPTCHA来说并不是一个难题,但是随着CAPTCHA技术的发展,高级CAPTCHA已普遍使用粘着严重的字符,而粘着字符的分割在文字识别领域已被证明是一个非常难的问题,从光学字符识别(Optical Character Recognition, OCR)提出到现在,都没有得到很好解决.设单个字符被成功分割的概率为 p,被成功识别的概率为 q,则对于长度为 L位的CAPTCHA,分割法的识别率 P为 $P=(pq)^L$.取经典值 $p=0.4$, $q=0.99$, $L=8$,则

$$P=(0.4\times 0.99)^8=0.000\,604\,73.$$

对应的识别失败概率 P'为

$$P'=1-P=0.999\,4.$$

因此虽然对单字符识别率为0.99,接近100%,但由于把高级CAPTCHA中的字符全部完整分割开来是几乎不可能的事情,由此导致基于分割的识别方法识别率非常低,在典型值下,是0.000 604 73.实际推断原理认为,概率很小的事件在一次实验中实际上几乎是不发生的,因此在典型值下,分割识别对高级CAPTCHA进行识别几乎不可能,这被著名的安全网站 http://securitylabs.websense.com所证实.该网站检测到有2台主机同时对Google的CAPTCHA进行大量的识别,其中一台使用基于分割的识别方法.监测显示,该主机基本上没有一次识别成功.因此对于高级CAPTCHA来说,以上提到的识别方法基本上很难取得成功,必须寻找不基于分割的识别方法.

3 基于LSTM型RNN的CAPTCHA识别方法

长短时记忆(Long Short-Term Memory, LSTM)型递归神经网络(Recurrent Neural Network, RNN)首先由Hochreiter等[9]提出,然后由Graves进行改进[10-11].LSTM型RNN突破HMM[12-14]在理论上的各种限制,能够建立输入值之间的长相关联系.实验表明,LSTM型RNN在上下文相关语言学习[15]、语音识别[10]等方面显示出较HMM模型更为优越的性能,并且不需要依赖词典,适合CAPTCHA识别的特点.因此本文采用LSTM型RNN(简称为RNN)对CAPTCHA进行识别.

3.1 LSTM型RNN

LSTM型RNN是一种用于解决梯度消亡现象的改进型RNN,一个典型的LSTM单元如图1所示[11].在该图中,NET INPUT用于接受网络 t时刻的输入以及 $t-1$时刻本层所有LSTM单元的输出结果.1个LSTM单元中存在1个或多个细胞核(Cell),用于描述LSTM单元的当前状态.图1中存在3个控制门,分别是INPUT GATE, OUTPUT GATE和FORGET GATE,3个门的输出分别连接到1个乘法单元上,从而分别控制网络的输入、输出以及cell单元的状态.处理完 O_t以后,只要保持INPUT GATE处于关闭状态(相当于乘法系数是0),FORGET GATE处于打开状态(相当于乘法系数是1),网络的输出就持续受到 O_t的影响.

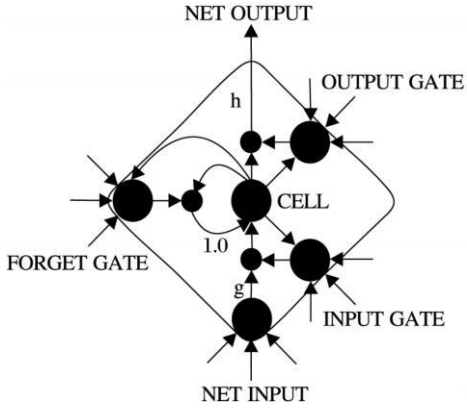


图1 LSTM单元结构图

Fig. 1 Architecture of long short-term memory block

典型的LSTM型RNN为3层结构:输入层、输出层和1个隐层.输入层的输入为特征值序列 $O_1,O_2,\cdots,O_T$(T为时间步数),输出层的输出是 $O_1,O_2,\cdots,O_T$对应的字符序列 $z_1,z_2,\cdots,z_L$(L为字符序列的长度).输入序列的长度和输出序列的长度并不存在固定的对应关系,同样长度的2个不同特征值序列可能产生不同长度的输出序列.每个隐层包含多个LSTM单元.t时刻隐层中第 i个LSTM单元的输入不仅包括输入层的输入,还包括隐层中包括 i单元在内所有LSTM单元在 $t-1$时刻的输出.

3.2 RNN的训练和识别

在基于HMM, BP, SVM等分类器的识别系统中,一般需要进行字符分割,然后使用分割以后的样本训练出字符模型或单词模型.但在基于RNN的

CAPTCHA识别中, 存在 1个优势在于训练和识别时不需要对 CAPTCHA中的字符进行分割. 得到一个 CAPTCHA图片后, 直接使用滑动窗口在图片上提取特征值序列, 然后与图片对应的字符序列构成一个训练样本, 即可使用这个样本对 RNN进行训练. 设特征值序列为 $x= O_1 O_2 \cdots O_T$, 对应的字符序列为 $z= z_1 z_2 \cdots z_L$, 那么, 训练的目标是使所有训练样本 (设为 S)的字符序列在对应的特征值序列之上的概率之积最大. 为防止溢出, 可以使用 ln函数将其描述为

$$O=-\ln\left(\prod_{(x,z)\in S} P(z\mid x)\right)=-\sum_{(x,z)\in S}\ln P(z\mid x).$$

通过使用前后向算法[10] 可计算 $P(z\mid x)$, 再计算目标函数相对于各个权值的导数, 然后使用随时间反向传播 (Back Propagation Through Time, BPTT) 或实时递归学习 (Real Time Recurrent Learning, RTRL)算法即可对网络进行训练.

训练好 RNN以后, 即可用于识别 CAPTCHA图片. 对于某个未知图片, 首先使用滑动窗口机制提取特征值序列, 然后将这个特征值序列输入到 RNN中, RNN即会输出对应的字符序列. 整个过程中不需要进行分割操作, 从而有效避免分割难题.

3.3　特征提取

特征选择和提取对识别系统至关重要, 它基本上决定系统所能达到的识别精度和其它一些性能. 针对文字识别, 研究人员已提出许多特征, 但是还没有理论或实验证明某种特征在各种情况下都能适用. 特别是在滑动窗口下, 很多被理论和实验证明的单字符特征并不适合, 因此本文使用多组特征对 CAPTCHA的识别进行研究.

在手写识别中, 一般在滑动窗口上提取一组基于矩特征和结构特征形成的混合特征. 因此与文献 [12] 类似, 本文提取窗口的平均灰度值、窗口重心、窗口的二阶矩、上下黑点位置及其相对变化率、黑白转换次数、上下黑点之间的黑像素比率等 9个参数作为 1组特征值, 将这组特征称为混合特征.

但是由于很难确定哪些特征是有效的, 而研究表明神经网络的隐层能够发现有用的中间表示, 能够创造出设计者没有明确引入的特征[16]. 因此除使用混合特征外, 还可以将原始图像数据直接输入到网络中, 让网络自动进行特征提取. 这需要先将图像归一化成相同的高度, 使得所有的图像具有相同的输入维数, 然后使用滑动窗口提取窗口内各点的灰度值作为特征向量. 将这组特征称为灰度值特征.

图像数据的维数一般很大, 将其直接作为特征输入, 会带来很大的计算量, 并可能引发维数灾难, 因此可考虑使用数据降维方法. 二维主分量分析 (Two Dimensional Principal Component Analysis, 2DPCA)[17] 是一种较新的专门针对图像信号的降维方法, 在人脸识别方面获得了较大成功. 2DPCA可表示为 $Y= AX$, 其中, A是图像矩阵, X是最佳投影轴, Y称为图像的投影特征向量.

使用 2DPCA进行 CAPTCHA图像的特征提取时, 首先需要计算协方差矩阵 G_t:

$$G_t = E\{[A- E(A)]^T[A- E(A)]\}.$$

在训练集 $B=\{B_1, B_2, \cdots, B_M\}$上, B_i代表一幅 CAPTCHA图像, M是训练集的大小, E(A)的无偏估计:

$$\mu = \frac{1}{M}\sum_{i=1}^{M} B_i$$

因此 G_t估计为

$$G_t = \frac{1}{M}\sum_{i=1}^{M}(B_i-\mu)^T(B_i-\mu).$$

估计出 G_t以后, 即可计算它的特征值和特征向量, 然后依照一定的选择方法选取前 k个特征向量 $\{X_1, X_2, \cdots, X_k\}$做为投影矩阵 X. 常用的选择依据是, 选择的前 k个特征向量对应的特征值之和至少占总特征值之和的 90%, 即

$$\frac{\sum_{j=1}^{k}\lambda_j}{\sum_{i}\lambda_i}\geqslant 0.9.$$

对于 CAPTCHA图像 A, 将其依次在 X_k进行投影即得到特征值序列 $\{Y_1, Y_2, \cdots, Y_k\}$, 其中 $Y_i = AX_i$, $1\leqslant i\leqslant k$但是 Y_i并不能直接作为 RNN的特征值序列, 因为它对应图像 A在 X_i上的整个投影. 基于滑动窗口机制的分类器 (例如 HMM, RNN等)对滑动窗口的一个基本要求是, 窗口移动的方向是字符逐步形成的方向[13], 可以对 Y进行一次转置实现该要求, 即

$$Y^T = (AX)^T = X^T A^T.$$

转置后 $(Y^T)_i = X^T(A^T)_i$, 其中 $(A^T)_i$是矩阵 A的第 行. 因此应该首先将 A旋转 90°, 然后提取 Y矩阵的行向量作为特征值序列, 这样可保证对图像的扫描是在字符逐步形成的方向.

4　RNN解码研究

4.1　RNN解码分析

RNN输出层的目的是求出对应 $x= O_1 O_2 \cdots O_T$的概率最大的字符序列 l, 也称为解码, 即

$$l^* = \arg\max_l p(l \mid x). \tag{1}$$

输出层共有 $M+1$个输出端, 其中, M为CAPTCHA中包含的字符类别数(也称为字典大小), 1对应空白字符. 对于每个时刻 t每个输出端都会有输出, 若使用 y_k^t表示第 k个输出单元在 t时刻的输出, 则

$$y_k^t = \frac{\exp(a_k^t)}{\sum_{k'} \exp(a_{k'}^t)}, \quad 1 \leqslant k \leqslant M+1, \quad 1 \leqslant t \leqslant T,$$

其中, a_k^t是第 k个输出单元在 t时刻的输入之和. 实质上 y_k^t表示在 t时刻输出字符 k的归一化概率.

在每个时刻 $t=1, 2, \cdots, T$选择一个输出端作为当前时刻的输出结果. 将 T个时刻的输出结果连接起来, 则形成一条路径 π, 但是路径并不是要输出的字符序列, 还需要经过函数 $B(\pi)$的处理. $B(\pi)$的主要作用是先将 π中相同的相邻字符合并成一个字符, 然后移除字符间的空格, 最后返回字符序列 l:

$$l = B(\pi).$$

不同的路径可得到相同的字符序列 l, 使用“-”表示1个空白输出, 则路径“A-DD-C”、“AA-DC”、“A-DD-C”经过 B函数处理以后都得到相同的字符序列 ADC, 因此在式(1)中 $p(l \mid x)$为

$$p(l \mid x) = \sum_{\pi \in B^{-1}(l)} p(\pi \mid x).$$

上式的计算需要使用前后向算法, 可参见文献[10].

解码是一个复杂的问题. 如果直接依据式(1), 则需要遍历所有可能的输出序列, 然后选择概率最大的 l. 对于时间长度为 T的输入, 其可能的输出有 M^T种, 取典型值 $T=80$, $M=56$, 则为 56^{80}. 完全遍历如此大的解空间基本不可能, 必须寻找一种有效的解码方法.

一种最直接的解码方法是最优路径解码(Best Path Decoding)[10-11]. 该方法认为由最大概率的路径形成的 l是最可能的输出. 最大概率的路径通过在每个时刻 t选择最大概率输出端 $y_{\max}^t$获得. 最优路径解码:

$$l^* \approx B(\arg\max_\pi p(\pi \mid x)),$$

$$p(\pi \mid x) = \prod_{t=1}^{T} p(\pi_t, t \mid x) = \prod_{t=1}^{T} y_{\max}^t.$$

但是最优路径解码会发生错误, 例如图2该图中时刻1和时刻2输出空白字符 blank的概率显著高于输出 A的概率, 因此最优路径解码将输出 blank, 但是实际上输出为 blank的概率为

$$p(l = \text{blank}) = p(--) = 0.6 \times 0.7 = 0.42$$

而输出为 A的概率为

$$p(l = A) = p(AA) + p(A-) + p(-A)$$
$$= 0.3 \times 0.4 + 0.3 \times 0.6 + 0.7 \times 0.4$$
$$= 0.58.$$

由于 $p(l = A) > p(l = \text{blank})$, 因此在图2中, 最可能的输出字符序列应该是 A, 而不是 blank.

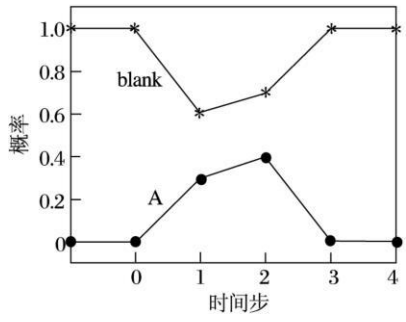


图2 最优路径解码发生错误

Fig. 2 Error case for best path decoding

为了弥补最优路径解码的缺陷, 文献[10]提出前缀搜索解码(Prefix Search Decoding). 其基本原理是将可能的字符输出形成一棵树, 然后计算每层节点的孩子结点的累计概率, 选择概率最大的节点向下扩展. 前缀搜索解码能够避免最优路径解码发生的错误, 但是算法时间复杂度随 x的长度指数增长, 并且计算过程难以理解[10], 因此很少使用. 一般应用时仍然使用最优路径解码[11].

4.2 邻域解码算法

鉴于最优路径解码和前缀搜索解码的不足, 本文基于禁忌搜索中的邻域思想提出一种解码算法——邻域解码. 在禁忌搜索中, 最优解是通过在当前解的邻域中迭代搜索获取的. 同样在邻域解码中, 首先求出最优路径解码的输出 $(\pi_{\max}^1, \pi_{\max}^2, \cdots, \pi_{\max}^T)$作为初始解, 然后在初始解的邻域中寻找更优解, 定义初始解的邻域 Neighbor为

$$\text{Neighbor} = \{(\pi_{(1-m_1)\max+m_1\sec}^1, \cdots, \pi_{(1-m_T)\max+m_T\sec}^T) \mid \sum_{i=1}^{T} m_i = 1, m_i \in \{0, 1\}\}, \tag{2}$$

其中

$$\pi_{(1-m_i)\max+m_i\sec}^t = \begin{cases} \pi_{\max}^t, & \text{if } m_i = 0 \\ \pi_{\sec}^t, & \text{if } m_i = 1 \end{cases}$$

上式的含义是, 在 t时刻, 当 m_i等于0时, 选择概率最大的输出端 $\pi_{\max}^t$, 否则取概率为次优的输出端 $\pi_{\sec}^t$.

分析式(2)可知, Neighbor集合的大小为 T,但这 T条路径并不是每条都需要考虑.为进一步的减少候选路径,引入路径置信度的概念,路径 $\pi = \pi^1_{sel}, \pi^2_{sel}, \cdots \pi^T_{sel}$的置信度定义为

$$Confidence(\pi) = \prod_{t=1}^{T} \frac{y^t_{sel}}{y^t_{max}},$$

其中, y^t_{sel}表示在 t时刻选择的输出端对应的概率, y^t_{max}表示 t时刻最大概率输出端的概率.分析可知,置信度的最大值为 1,对应最优路径解码输出的路径.置信度越小,则对应的路径越不可靠.

从邻域中获取多条路径以后,首先依据路径的置信度对路径进行排序,然后选择前 m个路径进行下一步求解.

如果使用 pathArray表示 l个存储多条路径的链表,初始值为空, π^t_{sel}表示在 t时刻选择的输出端,则邻域解码算法的基本步骤可描述如下.

step 1　计算最优路径解码得到的路径 $\pi = \pi^1_{max}\pi^2_{max}\cdots\pi^T_{max}$,置其置信度为 1,并且将 π加入到 pathArray中.

step 2　k= 1;

step 3　confidence= 1, π= Ø;

for t= 1 : T{　//遍历邻域

if(t= k) $\pi^t_{sel} = \pi^t_{se}$;　//选择次优输出端

else $\pi^t_{sel} = \pi^t_{max}$;　//选择最优输出端

$\pi = \pi + \pi^t_{se}$;　//连接各个时刻的输出端形成路径

confidence= confidence* (y^t_{sel}/y^t_{max});

}

step 4　将 π加入到 pathArray,并记录它的 confidence;

k++;

step 5　if k> T,转 step 6, else转 step 3;

step 6　依据置信度由大到小对 pathArray中的路径进行排序,选择前 m个;

step 7　for i= 1 : m

pathArray[i] = B(pathArray[i]);

依据前后向算法计算 pathArray[i]的后验概率;

step 8　输出 pathArray中后验概率最大的字符序列,算法结束.

分析算法可知,当 m= 1时,邻域解码等效于最优路径解码.由于选择最优路径解码的解作为初始解,并且在其邻域中只选择后验概率更大的字符序列,因此邻域解码算法的识别率只会大于等于最优路径解码的识别率.

邻域解码算法的时间复杂度为

$$O(T + T^2 + T\log(T) + mT_0 + m),$$

其中,第一项 T为 step 1所需时间, T^2为 step 2 ~ step 5的时间复杂度. $T\log(T)$为 step 6的时间复杂度. mT_0为 step 7时间复杂度, T_0为前后向算法的时间复杂度,最后一项 m为 step 8的时间复杂度.与最优路径解码对应的 O(T)时间复杂度相比,邻域解码算法的时间复杂度有所上升,但是远低于前缀搜索解码的指数级时间复杂度.

5　实验和结果分析

5.1　实验数据

测试一个识别系统的识别能力最好是在一个公共的数据集上进行,但是对于 CAPTCHA识别,并不存在相应的数据集.公布一个网站的 CAPTCHA识别程序会给该网站带来安全风险,同时也会给公布者带来法律问题.因此,与文献[18]相似,本文使用合成数据进行实验,使用一个开源 CAPTCHA生成程序产生粘着变形的样本,图 3是样本示例.

图 3　本文识别的 CAPTCHA示例

Fig. 3　Samples for CAPTCHA recognized by proposed method

5.2　实验设置

提取混合特征时,不对图像大小进行归一化,直接在每个滑动窗口上进行特征提取.提取灰度值特征时,将所有样本归一化成高度为 25、宽高比不变的图像.提取 2DPCA特征时,将所有的图像归一化为 96×36.

对于 3组特征,滑动窗口的宽度都为 1,重叠区域为 0.使用 3层结构的 RNN网络进行识别:1个输入层,输入单元数在使用混合特征时是 9,使用灰度值特征是 25,使用 2DPCA特征时是 8;1个中间层,含有 100个 LSTM单元;1个输出层,含有 57个输出单元.训练时为了提高 RNN的抗噪能力,对训练数据加入 μ= 0、σ= 1的高斯噪声. RNN网络使用带有冲量项的随机梯度下降法进行训练,学习速率为 1e-4,冲量为 0.9.每 10 epoch检测 1次验证集上的识别率,如果在检测 10次之后验证集上没有出现更好的结果,则停止训练.训练集大小为 1000,验证集为 500,测试集为 1000.邻域解码时取 m= 5.

特征提取和 CAPTCHA识别程序都使用 C++语言编写,由 VS2003进行编译.测试平台是 Windows

表 1 3组特征使用 3种算法的实验结果

Table 1 Experimental results of 3 algorithms using 3 kinds of features

使用特征	验证集识别率/%			测试集识别率/%			单个样本平均识别时间/s		
	方法 1	方法 2	方法 3	方法 1	方法 2	方法 3	方法 1	方法 2	方法 3
混合特征	19.4	20.8	20.8	21.1	22.4	22.4	0.346	0.558	2.06
灰度值	56.6	58.0	58.0	58.3	60.0	60.0	0.362	0.531	2.13
2DPCA	40.2	41.4	41.4	39.3	40.7	40.7	0.344	0.562	2.05

XP SP2,内存 512MB,CPU双核 1.6GHz.

5.3 实验结果及分析

表 1是基于混合特征、灰度值特征和 2DPCA特征的实验结果.其中方法 1、方法 2、方法 3分别表示最优路径解码、邻域解码和前缀搜索解码.

从表 1可以看出,识别率方面,邻域解码和前缀搜索解码都比最优路径解码高.识别时间方面,最优路径解码最短,邻域解码次之,前缀搜索解码最长,邻域解码算法的识别时间在可接受的范围内.实验说明邻域解码算法在取得较高识别率的同时,又有较低的时间复杂度,具有一定的优越性.

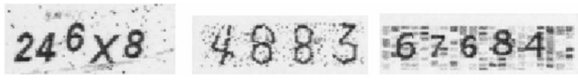
从实验结果可以看出,混合特征、灰度值特征和 2DPCA特征都取得一定的识别率,而 CAPTCHA设计时要求被识别率不能超过 5%,因此以上特征都达到识别 CAPTCHA的要求,说明本文使用 LSTM型 RNN进行粘着 CAPTCHA的识别是一种较好的方法.基于灰度值特征的识别率显著高于基于混合特征和基于 2DPCA特征的识别率,这说明灰度值对于 RNN是一种较好的特征,同样也说明 RNN具有较强地自动从原始图像数据中提取特征的能力.

5.4 与现有 CAPTCHA识别方法的对比

在脱机文字识别领域,在不能获取被比较方法的源程序的情况下,比较不同方法的识别率通常是较困难的.这是由于不同的预处理、后处理方法将极大地影响识别率,同时测试集的不同也导致识别率比较没有意义.对于 CAPTCHA识别,同样也存在这个问题,但是可通过分析识别的 CAPTCHA类型来比较识别方法的性能.图 4是文献[5]、[6]和[19]识别的 CAPTCHA.图 4(a)中第一幅图片虽然存在一定程度的粘着现象,但是属于点粘着,没有线粘着和字符重叠.后两张图片基本没有字符粘着.(b)、(c)中的 CAPTCHA虽然存在噪声和变形,但是字符之间都没有粘着或者粘着不严重.文献[19]也指出该文识别的 CAPTCHA字符之间的粘连都是几个点之间的粘连,并没有整条边都粘在一起的情况.因此,可以找到一种较好的分割算法将这 3个图的 CAPTCHA分割成单字符,而单字符的识别在文字识别领域已不是难题.但在图 3中,字符不仅存在变形,而且字符之间存在严重的点粘着、线粘着甚至重叠现象,因此很难找到分割算法对其进行分割,现有的方法很难成功,而本文方法却取得较好的识别率.

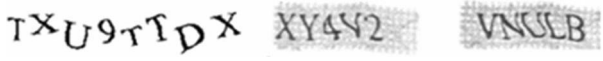
(a)文献[5]方法

(a) Method in reference [5]



(b)文献[6]方法

(b) Method in reference [6]



(c)文献[19]方法

(c) Method in reference [19]

图 4 3种方法识别的 CAPTCHA对比

Fig. 4 Samples for CAPTCHA recognized by 3 methods

6 结束语

CAPTCHA是目前互联网普遍使用的安全机制,目的是使用人工智能难题解决网络安全问题.研究 CAPTCHA的识别有助于发现 CAPTCHA的缺陷,使其变得更加安全,并有助于一些人工智能难题的求解.本文提出一种基于 LSTM型 RNN进行 CAPTCHA识别的方法,取得较好的识别率,能够识别现有方法难以识别的高粘着型 CAPTCHA.对 CAPTCHA识别的特征提取进行研究,通过实验证明灰度值对于 RNN是一种有效的特征.对 RNN进行改进,提出一种解码算法.相对于目前使用的最优路径解码算法,该算法能够取得更高的识别率,又有较低的时间复杂度.

但是实验中 CAPTCHA的识别率仍然不是很高,研究和发现更有效的特征,结合型 2模糊技术与 RNN神经网络,提高字符识别算法的鲁棒性是今后的研究方向.

参 考 文 献

[1] von Ahn L, Blum M, Hopper N J, et al. CAPTCHA: Using Hard AI Problems for Security // Proc of the 22nd International Conference on Theory and Applications of Cryptographic Techniques. Warsaw, Poland, 2003: 294-311

[2] Rusu A, Thomas A, Govindaraju V. Generation and Use of Handwritten CAPTCHAs. International Journal on Document Analysis and Recognition, 2010, 13(1): 49-64

[3] Rusu A, Govindaraju V. Handwritten CAPTCHA: Using the Difference in the Abilities of Humans and Machines in Reading Handwritten Words // Proc of the 9th International Workshop on Frontiers in Handwriting Recognition. Tokyo, Japan, 2004: 226-231

[4] von Ahn L, Maurer B, McMillen C, et al. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. Science, 2008, 321(5895): 1465-1468

[5] Xu Ming. Recognition and Anti-Recognition of Verification Code. Master Dissertation. Nanjing, China: Nanjing University of Science and Technology, College of Computer Science and Technology, 2007 (in Chinese)
(许 明. 验证码的识别和反识别. 硕士学位论文. 南京:南京理工大学. 计算机科学与技术学院, 2007)

[6] Chellapilla K, Simard P. Using Machine Learning to Break Visual Human Interaction Proofs (HIPs) // Weiss Y, Schölkopf B, Platt J, eds. Advances in Neural Information Processing Systems. Cambridge, USA: MIT Press, 2004, 17: 265-272

[7] Jiang Peng. Investigation on Verification Code and Its Implementation as Web Service. Master Dissertation. Nanjing, China: Nanjing University of Science and Technology, College of Computer Science and Technology, 2007 (in Chinese)
(姜 鹏. 验证码识别及其 Web Service的实现研究. 硕士学位论文. 南京:南京理工大学. 计算机科学与技术学院, 2007)

[8] Hocevar S. PWNTCHA—Pretend We're Not a Turing Computer But a Human Antagonist [EB/OL]. [2010-02-15]. http://sam.zoy.org/wikiy/PWNtcha

[9] Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Computation, 1997, 9(8): 1735-1780

[10] Graves A. Supervised Sequence Labelling with Recurrent Neural Networks. Ph.D Dissertation. Manno, Switzerland: Technical University of Munich, Dalle Molle Institute for Artificial Intelligence, 2008

[11] Graves A, Liwicki M. A Novel Connectionist System for Unconstrained Handwriting Recognition. IEEE Trans on Pattern Analysis and Machine Intelligence, 2009, 31(5): 855-868

[12] Varga T. Off-line Cursive Handwriting Recognition Using Synthetic Training Data. Ph.D Dissertation. Bern, Switzerland: University of Bern, Institute of Computer Science and Applied Mathematics (IAM), 2006

[13] Su Tonghua. Off-line Recognition of Chinese Handwriting: From Isolated Character to Realistic Text. Ph.D Dissertation. Harbin, China: Harbin Institute of Technology, School of Computer Science and Technology, 2008 (in Chinese)
(苏统华. 脱机中文手写识别——从孤立汉字到真实文本. 博士学位论文. 哈尔滨:哈尔滨工业大学. 计算机科学与技术学院, 2008)

[14] Zhao Wei, Liu Jiafeng, Tang Xianglong, et al. Cascaded HMM Training Algorithm for Continuous Character Recognition. Chinese Journal of Computers, 2007, 30(12): 2142-2150 (in Chinese)
(赵 巍, 刘家锋, 唐降龙, 等. 连续字符识别的级联 HMM训练算法. 计算机学报, 2007, 30(12): 2142-2150)

[15] Gers F A, Schmidhuber J. LSTM Recurrent Networks Learn Simple Context-Free and Context-Sensitive Languages. IEEE Trans on Neural Networks, 2001, 12(6): 1333-1340

[16] Mitchell T M. Machine Learning. New York, USA: McGraw-Hill, 1997

[17] Yang Jian, Zhang D, Alejandro F, et al. Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition. IEEE Trans on Pattern Analysis and Machine Intelligence, 2004, 26(1): 117-129

[18] Wachenfeld S, Klein H V, Jiang Xiaoyi. Recognition of Screen-Rendered Text // Proc of the 18th International Conference on Pattern Recognition. Hongkong, China, 2006, II: 1086-1089

[19] Li Ying. Investigation on Generation and Recognition of Verification Code. Master Dissertation. Nanjing, China: Nanjing University of Science and Technology, College of Computer Science and Technology, 2008 (in Chinese)
(李 颖. Web验证码的生成与识别. 硕士学位论文. 南京:南京理工大学. 计算机科学与技术学院, 2008)