



# (12)发明专利申请

(10)申请公布号 CN 107330480 A

(43)申请公布日 2017. 11. 07

(21)申请号 201710532593.X

(22)申请日 2017.07.03

(71)申请人 贵州大学

地址 550025 贵州省贵阳市花溪区贵州大学(北区)科技处

(72)发明人 杨观赐 杨静 苏志东 陈占杰  
袁庆霓 蓝伟文

(74)专利代理机构 贵阳东圣专利商标事务有限公司 52002

代理人 袁庆云

(51)Int.Cl.

G06K 9/68(2006.01)

G06N 3/04(2006.01)

G06N 3/08(2006.01)

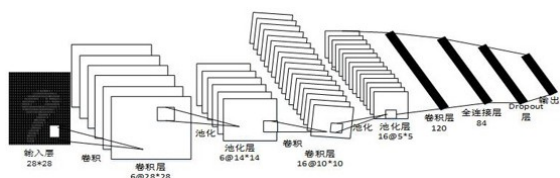
权利要求书1页 说明书7页 附图2页

## (54)发明名称

手写字符计算机识别方法

## (57)摘要

本发明公开了一种手写字符计算机识别方法,其特征在于:基于二次卷积神经网络结构模型,该模型有9层:包括输入层、5个由卷积层和池化层交错构成的隐含层、全连接层、输出层,其中全连接层之后具有一个Dropout层,具体步骤包括:预训练滤波器、输入用于训练的图片数据集、使用Relu激活函数,输出特征向量等。本发明具有能加快收敛速度,解决过拟合问题,减少累积误差,提高识别率的特点。



1. 一种手写字符计算机识别方法,其特征在于:基于二次卷积神经网络结构模型,该模型有9层:包括输入层、5个由卷积层和池化层交错构成的隐含层、全连接层、输出层,其中全连接层之后具有一个Dropout层;具体包括下述步骤:

步骤1:预训练滤波器,并初始化滤波器尺寸的像素;

步骤2:输入用于训练的图片数据集,将训练集中的图片处理成与滤波器尺寸相同的图片,并读入数据形成图片数据矩阵X;

步骤3:初始化权重 $w^{(1)}_{i,j}$ 、偏置 $b_i$ ;

步骤4:进行池化操作得到第1层卷积特征矩阵 $X^{(1)}$ ;

步骤5:运用ADAM优化器函数公式获得自顶向下的调节优化器的学习率,并调用Tensorflow中的权重与偏重更新接口更新权重 $w_i$ 和偏置 $b_i$ ,从而获得特征矩阵 $X^{(2)}$ ;

步骤6:按照步骤4与步骤5进行第二次卷积,得到特征矩阵 $X^{(3)}$ ;

步骤7:将特征矩阵 $X^{(3)}$ 合并为一个列向量作为神经元在全连接层的输入,然后将之与权重矩阵相乘,加上偏置,并对其使用Relu激活函数,得到特征向量 $b_1$ ;

步骤8:将全连阶层的特征向量作为dropout层的输入,并计算神经元在dropout层中的输出概率,得到特征向量 $b_2$ ;

步骤9:将特征向量 $b_2$ 作为输入,利用分类器输出识别结果,即可。

2. 如权利要求1所述的手写字符计算机识别方法,其特征在于:所述步骤4中,卷积特征计算中步长为2,边距设置为0;池化操作使用 $3 \times 3$ 小的矩阵。

3. 如权利要求1或2所述的手写字符计算机识别方法,其特征在于:所述步骤5中,运用ADAM优化器函数公式为:

$$J_{\theta}(\theta) = -\sum_i q_i \log(\theta_i)$$

其中

$J_{\theta}(\theta)$  为成本函数,  $\theta$  是预测的概率分布,  $\theta$

是实际的分布。

## 手写字符计算机识别方法

### 技术领域

[0001] 本发明属于图像识别领域,具体地说是涉及一种手写字符计算机识别方法。

### 背景技术

[0002] 卷积神经网络(Convolutional neural network,CNN)因其在目标检测、图像分类、知识获取、图像语义分割等领域的成功应用而备受关注,提高改进其性能是一个研究热点。现有技术中,在解决图像目标检测问题时,采用CPU控制卷积神经网络的整体流程和数据调度,运用GPU提高神经网络单元中的卷积计算和全连接层合并计算单元的运算速度,虽然神经网络学习速度得到了改善,但是也因为CPU和GPU之间的数据转换和调度导致了时间成本的增加,而且弱的GPU平台容易出现进程中断的问题。基于免疫机制获得了改进的卷积神经网络,虽然在MNIST数据集上取得了108.501s的较短识别时间,但是却导致识别率仅有81.6%。研究者提出了特征映射模型的多输入Sigmoid激活函数神经网络权值共享的最优网络结构算法。研究者使用不同大小的神经元内核组成特征图,将神经元元素引入到最大汇集层,从而将不同大小的神经元映射到汇集层。该方法在特征字符集中虽然获得了96.33%的识别率,但是其消耗的时间也较大。改进的MLP-CNN模型(MLP-CNN)通过增加神经网络特征数、使用随机梯度下降算法优化“交叉熵”,从而提高模型的性能。

[0003] 在缓和过拟合问题方面,研究者设计了一种伴随目标函数,并建立了基于卷积滤波器和非线性激活函数的辅助监听机制与规范化的辅助监听策略,从而提出了一种可以缓和卷积神经网络过拟合问题的伴随目标函数的正则化策略机制。此方法须采用端到端监督学习以微调正则化策略,进行辅助监听的卷积滤波器和非线性激活函数需要额外时间开销。研究者运用Laplace-Beltrami算子优化网络参数,在解决加权的磁共振图像识别问题中取得了很好效果,但针对小样本数据引起的过拟合问题需要采用监督学习的方式微调训练网络。研究者利用卷积神经网络无监督学习功能,将双线性插值引入卷积神经网络结构,并引入细粒度美学质量进行预测分类,从而实现对照片进行自动化美学评价。虽然解决了因高质量图片太大而导致卷积神经网络无法完整提取整个图片特征的问题,但是对较小的数字摄影图片集引起的过拟合问题,没有提出相应的解决方案。研究者通过在每一个卷积层嵌入一个求局部平均与二次特征提取的计算层,获得了一种改进的卷积神经网络,其特有的两次特征提取结构可以减小特征分辨率,但当数据集规模增大、卷积神经网络的层数增加时,容易导致过拟合;最近邻规则的神经网络模型(WCNN)使用Sigmoid函数作为激活函数,通过复合多个“卷积层”和“池化层”实现输入信号的加工处理;同时,在连接层与输出目标间建立了映射关系,并运用聚类算法对特征进行分类。研究者在卷积神经网络的每层附加两个连续的卷积操作,通过加倍特征提取数量提高图像分类的识别率,对系统的内存要求高。

[0004] 此外,针对卷积神经网络的收敛速度问题,研究者将多变量最大乘积和插值算子理论引入到卷积神经网络结构中,以算子理论中的运算符作为激活函数。此研究给出了详细的数学公式推导,但是没有给出应用测试结果。卷积神经网络技术,为图像的特征提取提

供了新的途径。虽然许多学者就提高经典CNN的性能方面做了许多工作,但是在解决图像特征提取问题时,存在以下不足:

[0005] 1) 采用Sigmoid函数作为激活函数时,有可能使得原本在较大范围内变化的输入值挤压到(0,1)范围内输出,当训练数据集规模较大时,Sigmoid函数容易造成梯度饱和与收敛速度慢的问题。

[0006] 2) CNN中常采用两种策略缓解过拟合问题:早停与正则化策略。早停策略中数据集被划分成训练集和测试集,前者用来计算梯度、更新连接权和阈值,后者用来评估误差,训练停止的标志是训练集误差降低而测试集误差升高;正则化策略则是其误差目标函数考虑了用于描述复杂度的因素。当CNN的学习层数增加时,他们解决过拟合问题的表现会变差。

[0007] 3) 在训练和评估阶段,CNN采用梯度下降策略,通过反向传播算法调整目标梯度的最小化训练集的累积误差,这个过程中并非每一层训练都评估累积误差,而是在给定的间隔层后才进行评估,虽然时间开锁会有所减少,但会导致累积误差增加。

## 发明内容

[0008] 本发明的目的在于克服上述缺点而提供一种能加快收敛速度,解决过拟合问题,减少累积误差,提高识别率的手写字符计算机识别方法。

[0009] 本发明的一种手写字符计算机识别方法,其中:基于二次卷积神经网络结构模型,该模型有9层:包括输入层、5个由卷积层和池化层交错构成的隐含层、全连接层、输出层(Softmax),其中全连接层之后具有一个Dropout层;具体包括下述步骤:

[0010] 步骤1:预训练滤波器,并初始化滤波器尺寸的像素;

[0011] 步骤2:输入用于训练的图片数据集,将训练集中的图片处理成与滤波器尺寸相同的图片,并读入数据形成图片数据矩阵X;

[0012] 步骤3:初始化权重 $w_{i,j}^{(1)}$ 和偏置 $b_i$ ,并调用Tensorflow提供的核函数def Kernel()初始化并行操作;

[0013] 步骤4:利用公式计算第1层卷积特征矩阵 $X^{(1)}$ ,其公式为: $x_j^l = f(\sum_j x_j^{(l-1)} * w_{i,j}^{(l)} + b_i^{(l)})$ ;

[0014] 步骤5:运用ADAM优化器函数公式获得自顶向下的调节优化器的学习率,并调用Tensorflow中的权重与偏置更新接口更新权重 $w_i$ 和偏置 $b_i$ ,从而获得特征矩阵 $X^{(2)}$ ;其公式:

$$y_{\theta'}(\theta) = -\sum_i \theta'_i \log(\theta_i);$$

[0015] 步骤6:按照步骤4与步骤5进行第二次卷积,得到特征矩阵 $X^{(3)}$ ;

[0016] 步骤7:将特征矩阵 $X^{(3)}$ 合并为一个列向量作为神经元在全连接层的输入,然后将之与权重矩阵相乘,加上偏置,并对其使用Relu激活函数,得到特征向量 $b_1$ ;

[0017] 步骤8:将全连阶层的特征向量作为dropout层的输入,并通过公式计算神经元在dropout

层中的输出概率,得到特征向量 $b_2$ ;其公式为:
$$p = (p_a = 1 | x)$$
$$= \sum_{i,j \in B_a} \exp(w_{i,j}^{(l)} * x_j^{(l-1)} + b_i^{(l)}) / (1 + \exp(\sum_{i,j \in B_a} w_{i,j}^{(l)} * x_j^{(l-1)} + b_i^{(l)}));$$

[0018] 步骤9:将特征向量 $b_2$ 作为输入,利用Softmax分类器输出识别结果。

[0019] 上述的手写字符计算机识别方法,其特征在于:所述步骤4中,卷积特征计算中步长为2,边距设置为0;池化操作使用 $3 \times 3$ 大小的矩阵。

[0020] 本发明与现有技术的相比,具有明显的有益效果,由以上方案可知,所述的二次卷积神经网络结构模型,该模型有9层,包括输入层、5个由卷积层和池化层交错构成的隐含层、全连接层、输出层(Softmax),其中全连接层之后具有一个Dropout层;改进了卷积神经网络结构。在具体的步骤中,通过引入基于Relu的激活函数以避免梯度饱和问题、提高收敛速度;通过在全连接层和输出层之间加入Dropout层解决过拟合问题,并设计了ADAM优化器的最小化“交叉熵”,将Dropout和ADAM与CNN相结合,形成了基于Dropout与ADAM优化器的改进卷积神经网络的手写字符计算机识别方法,并基于TensorFlow平台进行了并行化实现。该方法通过改进激活函数,避免了神经元节点输出恒为0的问题,在识别准确性、与时间开销方面得到了改善。

[0021] 以下通过具体实施方式,进一步说明本发明的有益效果。

### 附图说明

[0022] 图1为本发明的二次卷积神经网络结构模型;

[0023] 图2为实施例中的非线性函数;

[0024] 图3为实施例中的手写字符图像样本示例;

[0025] 图4为实施例中的MCNN-DA算法在ADAM不同学习率下的识别率;

[0026] 图5为实施例中的各种算法针对MNIST数据集的识别结果。

### 具体实施方式

[0027] 以下结合附图及较佳实施例,对依据本发明提出的手写字符计算机识别方法具体实施方式、特征及其功效,详细说明如后。

[0028] 如图1所示,本发明的一种手写字符计算机识别方法,基于所设计的二次卷积神经网络结构、Relu激活函数、以及基于Dropout与ADAM的过拟合防止方法,形成一种基于Dropout与ADAM优化器的卷积神经网络算法(aconvolution neural network algorithm based on Dropout and ADAM optimizer,MCNN-DA),主要流程如下:

[0029] 步骤1:预训练滤波器,并初始化滤波器尺寸的像素为: $P_1 \times P_2$ ;

[0030] 步骤2:输入用于训练的图片数据集,将训练集中的图片处理成与滤波器尺寸相同的图片,并读入数据形成图片数据矩阵X;

[0031] 步骤3:初始化权重 $w_{i,j}^{(1)}$ 和偏置 $b_i$ ,并调用Tensorflow提供的核函数def Kernel()初始化并行操作;

[0032] 步骤4:利用公式(1)计算第1层卷积特征矩阵 $X^{(1)}$ 。此步骤中,卷积特征计算中步长为2,边距设置为0;同时,为了保证经特征提取后输入和输出的图片具有相同的尺寸;池化操作使用 $3 \times 3$ 大小的矩阵;

[0033] 步骤5:运用ADAM优化器函数公式(3)获得自顶向下的调节优化器的学习率,并调用Tensorflow中的权重与偏重更新接口更新权重 $w_i$ 和偏置 $b_i$ ,从而获得特征矩阵 $X^{(2)}$ ;

[0034] 步骤6:按照步骤4与步骤5进行第二次卷积,得到特征矩阵 $X^{(3)}$ ;

[0035] 步骤7:将特征矩阵 $X^{(3)}$ 合并为一个列向量作为神经元在全连接层的输入,然后将之与权重矩阵相乘,加上偏置,并对其使用Relu激活函数,得到特征向量 $b_1$ ;

[0036] 步骤8:将全连阶层的特征向量作为dropout层的输入,并通过公式(2)计算神经元

在dropout层中的输出概率,得到特征向量 $b_2$ ;

[0037] 步骤9:将特征向量 $b_2$ 作为输入,利用Softmax分类器输出识别结果。

[0038] 1.二次卷积神经网络结构

[0039] 如图1所示的9层的二次卷积神经网络模型,它包含输入层、5个由卷积层和池化层交错构成的隐含层、全连接层、输出层(Softmax)。在这个结构中,全连接层之后具有一个Dropout层,他以给定的概率 $p$ 丢神经元节点(后文的测试中,训练阶段 $p=0.5$ ,在试阶段 $p=1$ );除了输出层之外,其余各层的激活函数均为ReLU函数。进行池化操作(max polling)的计算按公式(1)进行。

$$[0040] \quad x_j^{(l)} = f\left(\sum_j x_j^{(l-1)} * w_{i,j}^{(l)} + b_i^{(l)}\right) \quad (1)$$

[0041] 式中, $w_{i,j}^{(l)}$ 表示第 $l$ 层第 $i$ 类中第 $j$ 个神经元的权重, $b_i$ 代表第 $i$ 类的偏置,“\*”为卷积操作; $x_j^{(l)}$ 为第 $l$ 层卷积中第 $j$ 个神经元的输出; $x_j^{(l-1)}$ 表示第 $l-1$ 层第 $j$ 个神经元的输出,即第 $l$ 层的输入数据; $f(\dots)$ 是模型的激活函数,具有非线性特征。

[0042] 2.基于Relu的激活函数

[0043] 传统卷积神经网络CNN常采用的非线性函数tanh函数和sigmoid函数为激活函数(如图2(a)与2(b)所示)。

[0044] sigmoid函数将一个实数输入映射到 $[0,1]$ 范围内,作为激活函数存在以下两个问题:1)存在梯度饱和问题。当函数激活值趋近极值0或者1时,函数的梯度趋近于0。对于第 $l$ 层神经元反向传播产生的成本 $\delta^{(l)}$ 的公式为 $\delta^{(l)} = (w^{(l)})^T \delta^{(l+1)} * f'(x_j^l)$ ,  $(w^{(l)})^T$ 表示第 $l$ 层的第 $T$ 个神经元的权重,当第 $l+1$ 层神经元反向传播成本 $\delta^{(l+1)}$ 趋近于0时,计算所得的梯度也趋近于0,达到不调整更新参数的目的。2)导致权重恒为正。函数输出的均值非0,这会导致后一层神经元获得非0均值的信号输入,从而导致输入神经元的数据为正,最终导致其权重恒为正。这些问题会导致参数收敛速度慢,影响训练的效率与模型的识别效果。tanh函数可以将一个实数输入映射到 $[-1,1]$ 范围内,但它实际上是sigmoid函数的变形,即 $\tanh(x) = 2\text{sigmoid}(2x) - 1$ ,tanh函数也存在梯度饱和问题。

[0045] 事实上,Relu(The Rectified Linear Unit)函数 $f(x) = \max(0, x)$  ( $x \in (0, +\infty)$ ) (见图2(c))具有以下特点:(1)梯度不饱和。梯度的计算公式为: $I\{x > 0\}$ 。因此在反向传播过程中,减轻了梯度弥散的问题,神经网络前几层的参数也可以很快的更新。(2)计算复杂性低。Relu函数仅需要设置阈值,即如果 $x < 0$ ,  $f(x) = 0$ ,如果 $x > 0$ ,  $f(x) = x$ 。

[0046] 正是考虑到Relu函数的优势,采用Relu函数作为激活函数,以解决梯度饱和问题,提高收敛速度。

[0047] 3.基于Dropout与ADAM的过拟合防止方法

[0048] 为了提高网络的范化能力,将Dropout引入到卷积神经网络中。对于神经网络中的任一神经元,以概率 $p$ 将其暂时从网络中丢弃,其公式如下:

$$[0049] \quad p = (p_a = 1 | x) \\ = \sum_{i,j \in B_a} \exp(w_{i,j}^{(l)} * x_j^{(l-1)} + b_i^{(l)}) / (1 + \exp \sum_{i,j \in B_a} w_{i,j}^{(l)} * x_j^{(l-1)} + b_i^{(l)}) \quad (2)$$

[0050] 同时,为了训练模型,首先定义一个表示模型是“坏”的指标作为成本,并使用“交叉熵”作为成本函数 $y_{\theta^*}(\theta)$ ,其定义如下:

[0051] 
$$y_{\theta'}(\theta) = -\sum_i \theta'_i \log(\theta_i) \quad (3)$$

[0052] 其中,  $\theta$  是预测的概率分布,  $\theta'$  是实际的分布。在模型训练阶段, 使用ADAM优化器以不同的学习率优化“交叉熵”。

[0053] 有益效果分析如下:

[0054] 1. 测试环境

[0055] 为了评估所设计算法的性能, 在ubuntu14.04.4的64位操作系统中采用python语言编程实现了二次卷积神经网络学习算法, 并安装了TensorFlow 0.9.0。对不同ADAM学习率的情况下, 测试、分析了基于Dropout与ADAM优化器的卷积神经网络算法的性能, 并将其与四种算法进行了对比。所得结果都是在多台同一批次购进的同型号电脑上测得, 其配置为:i5-5200U CPU@2.20GHz×4, GPU为NVIDA GeForce 940M, 8.00GB内存。

[0056] 2. 比较算法

[0057] 用于比较的四种算法详情如下:

[0058] 1) 算法1: 最近邻规则的神经网络模型(WCNN)。此算法使用Sigmoid函数作为激活函数, 通过复合多个“卷积层”和“池化层”实现输入信号的加工处理; 同时, 在连接层与输出目标间建立了映射关系, 并运用聚类算法对特征进行分类。

[0059] 2) 算法2: 改进的MLP-CNN模型(MLP-CNN)。此算法通过增加神经网络特征数、使用随机梯度下降算法优化“交叉熵”, 从而提高模型的性能。

[0060] 3) 算法3: 极速学习机与支持向量机结合的算法(SVM-ELM), 此算将极速学习机与支持向量机结合, 削减隐层节点数为类别数, 通过SVM优化每个节点的线性决策函数。

[0061] 4) 算法4: 基于多分辨率直方图和梯度下降算法(GFGN)。

[0062] 3. 数据集及设置

[0063] 测试使用的数据集包括2类: MNIST手写体数字集和HCL2000手写汉字数据集, 它们均为灰度图像, 详情见表1, 示例见图3。MNIST是美国国家标准和技术研究所提供的一个专门用于手写体数字识别研究的数据库, 包含60000个训练样本和10000个测试样本, 每个样本为28px×28px的bmp图片。HCL2000是在国家863计划的资助下, 由北京邮电大学模式识别实验室建立的国家脱机手写汉字标准数据库, 搜集了1300个书写者针对3755个一级汉字的书写笔迹形成1300×3755个样本, 每个汉字样本为64px×64px个二值像素。本实验中, 选取随机其中的3000个汉字样本构成训练集或测试集, 并将该数据集分为10个类别, 同时将其调整为28px×28px的尺寸, 并把像素归一化到[0, 1]。

[0064] 将从MNIST数据集中选取的60000测试样本和10000训练样本划分为10类: 即0~9共10个字符分别对应一个训练集与测试集; 将HCL2000数据集中选取的2000个测试样本、1000个训练样本划分为10类, 即每一个汉字对应一个训练集与测试集; 2个卷积层的卷积核大小分别为7×7、5×5, 池化因子为3×3, 最大迭代次数为20000, 实验次数为10次, Dropout在训练阶段的概率 $p=0.5$ , 在测试阶段的概率 $p=1$ 。

[0065] 表1测试与训练数据集

	数据集	训练集的规模	测试集规模	图片分类数量
[0066]	MNIST	60000	10000	10
	HCL2000	2000	1000	10

#### [0067] 4. 不同学习率下的识别性能实验结果与分析

[0068] 为了发现ADAM优化器的不同学习率对算法性能的影响,根据上节的实验设置,给定不同的学习率以测试分析算法的性能。ADAM优化器最小化“交叉熵”的学习率设置为:0.02、0.03、0.04、0.05、0.06、0.07、0.08、0.09、0.1。每一个学习率下的10次平均识别率的统计结果如图4所示。

[0069] 从图4可以看出,对于所测试的2个数据集,当学习处于 $[0.2, 0.4]$ 内时,模型的识别率随着学习率的增加而提高,学习率与识别率表现出正相关。当ADAM优化器的学习率分布 $0.04 \sim 0.08$ 之间时,模型的识别率处于相对比较稳定的较高状态,此时,随着学习率的增加模型的识别率表现出微弱的增长趋势。随着学习率增大到0.8以上后,模型的识别率随着学习率的增加而显著下降,特别是当学习率达到1时,模型的识别率达到最低水平:20%左右。基于此数据集的测试表明:ADAM优化器的引入,可以影响系统的识别率。当学习率处于 $0.04 \sim 0.08$ 时,系统具有较好的表现。在后文的对比实验中,ADAM优化器的学习率设置为0.06。

#### [0070] 5. 与四种算法的比较与分析

[0071] 每个算法针对每个测试集运行10次,不同算法的识别率统计结果如表2所示,不同数据规模下各算法在MNIST样本库中的识别率统计结果如图5所示。值得说明的是:由于算法SVM-ELM与SVM-ELM没有针对此数据集的测试结果,因此他们的数据没有体现在图中。

[0072] 观察图5可知,本文算法对于不同的数据规模的识别率均高于算法MLP-CNN与WCNN。随着数据规模的增长,MLP-CNN算法的识别性能有较大的波动。WCNN算法的识别率,在数据规模小于3.6万条时,随着数据规模的增大,识别率有上升的趋势,但是当数据规模大于3.6万条后,识别率反而有下降的趋势。本文算法对应的识别率曲线处于MLP-CNN与WCNN算法对应曲线的上方,这表明本文算法获得了更高的识别率。特别,随着数据规模的增大,本文算法的识别率波动较小,表现出了较强的鲁棒性。

[0073] 观察表2可知,对于MNIST测试集,在最低识别率方面,WCNN、MLP-CNN、SVM-ELM、GFCN及本文算法的10次运算的平均值分别为95.11%、97.82%、89.5%、91.36%和98.02%,本文算法的性能表现排第1,MLP-CNN算法排第2,SVM-ELM算法的表现最差。在最高识别率及平均识别率方面,本文算法的性能表现排第1,WCNN算法排第2,MLP-CNN算法的表现最差。事实上,这些数据表明:基于卷积神经网络模型的WCNN、MLP-CNN及本文算法在特征识别方面的表现,最差时均比SVM-ELM和GFCN算法特征提取与融合的效果更佳。就表中的数据而言,本文算法的识别率表现最好,最高时可以达到99.21%。此外,正是因为对算法并行化的实现,本文算法在时间开销方面也同样具有优势,本文算法的时间开销是WCNN算法的17.42%,MLP-CNN算法的5.81%,SVM-ELM算法的12.72%,GFCN算法的9.69%。

[0074] 对于HCL2000测试集,本文算法与SVM-ELM和GFCN算法相比,同样具有优势,在最低识别率方面,本文算法比SVM-ELM和GFCN算法分别提高了9.16%和7.45%。在最高识别率方



面,本文算法比SVM-ELM和GFGN算法分别提高了3.73%和4.17%。平均识别率方面,本文算法比SVM-ELM和GFGN算法分别提高了3.98%和5.31%。在时间开销方面,本文算法的时间开销是SVM-ELM算法的21.39%,GFCN算法的31.83%。

[0075] 综上所述:本文算法在识别时间开销与识别率方面均比比较算法表现优越,算法具有良好的鲁棒性。

[0076] 以上所述,仅是本发明的较佳实施例而已,并非对本发明作任何形式上的限制,任何未脱离本发明技术方案内容,依据本发明的技术实质对以上实施例所作的任何简单修改、等同变化与修饰,均仍属于本发明技术方案的范围内。

[0077] 表2算法识别性能比较

数据集名称	指标	WCNN[29]	MLP-CNN[30]	SVM-ELM[31]	GFGN[32]	MCNN-DA (本文算法)
MNIST	最低识别率 (%)	95.11	97.82	89.5	91.36	98.02
	最高识别率 (%)	95.71	98.96	91.35	94.13	99.67
	平均识别率 (%)	95.36	96.32	90.26	93.18	99.21
	平均识别时间 (ms)	96878	290491	132634	174132	16877
HCL2000	最低识别率 (%)	--	--	83.60	84.93	91.26
	最高识别率 (%)	--	--	90.00	89.62	93.36
	平均识别率 (%)	--	--	88.63	87.51	92.16
	平均识别时间 (ms)	--	--	316372	212633	67686

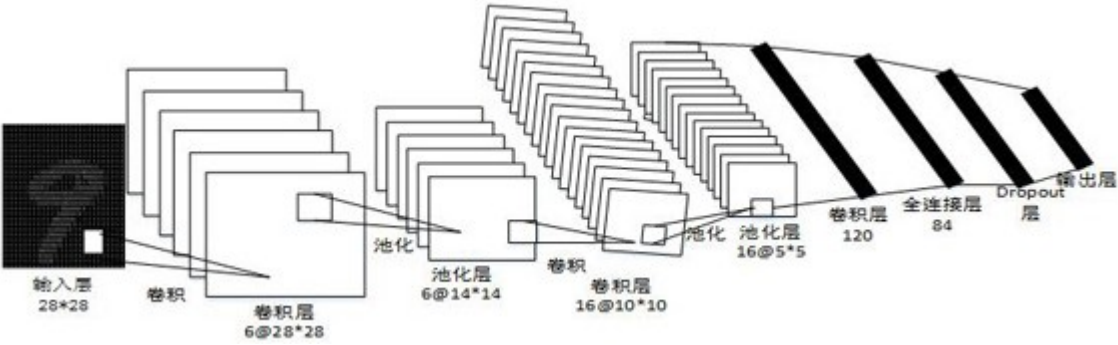
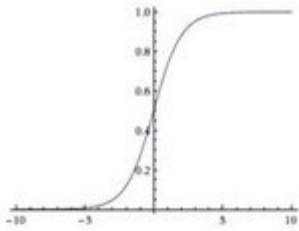
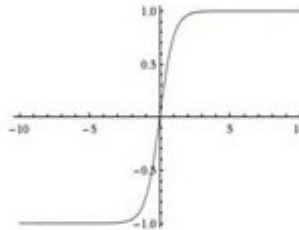


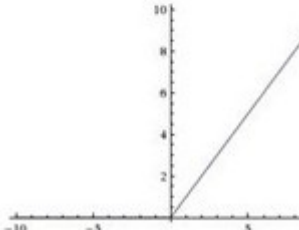
图1



(a) sigmoid 函数



(b) tanh 函数



(c) Relu 函数

图2

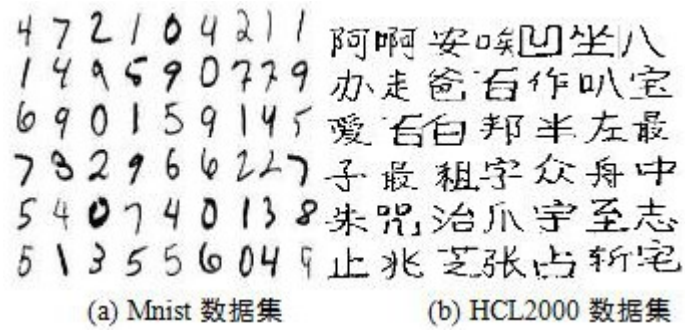


图3

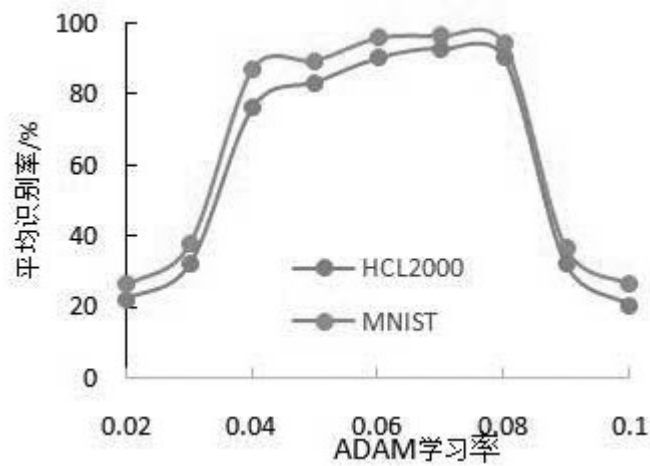


图4

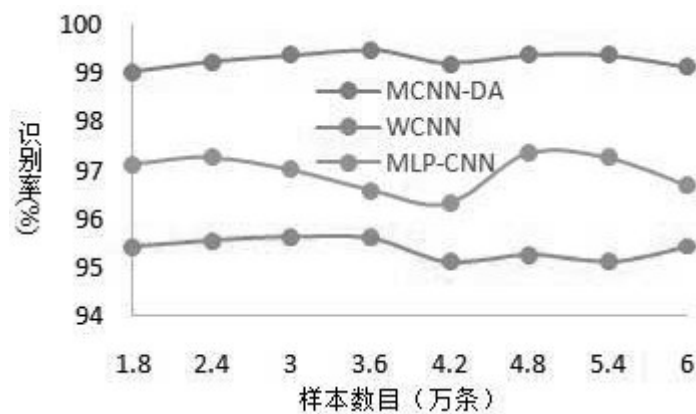


图5