

DOI:10.15923/j.cnki.cn22-1382/t.2019.1.05

# 基于 PSO 算法与 Dropout 的改进 CNN 算法

王金哲, 王泽儒, 王红梅\*

(长春工业大学 计算机科学与工程学院, 吉林 长春 130012)

**摘要:** 设计卷积升级网络结构, 引入 PSO 算法减小了误差的反向传播, 避免了滞后误差与图像的过拟合, 提高了收敛速度。将该方法应用到数据集 HCL2000 和 MNIST 上, 并进行了与 WCNN、MLP-CNN、SVM-ELM 的实验对比, 证明了改进算法的正确性。

**关键词:** 粒子群算法; CNN; Dropout; 过拟合

**中图分类号:** TP 311; TP 183 **文献标志码:** A **文章编号:** 1674-1374(2019)01-0026-05

## An improved CNN algorithm based on PSO and dropout

WAN Jinzhe, WANG Zeru, WANG Hongmei\*

(School of Computer Science & Engineering, Changchun University of Technology, Changchun 130012, China)

**Abstract:** A convolution upgrade network structure is designed, and then PSO algorithm is introduced to reduce the error back propagation. Accordingly, the over-fit between delayed error and image is avoided for improving convergent speed. The method is applied to data set HCL2000 and MNIST, and experiments compared with WCNN, MLP-CNN and SVM-ELM are carried.

**Key words:** Particle swarm optimization; CNN; Dropout; over-fit.

## 0 引言

近年来,伴随着互联网技术的飞速发展,人工智能已经应用到生活的不同方面,其中自然语言处理、机器人学、人脸识别、人工神经网络、图像理解和智能搜索等是目前人工智能领域的研究热点。由于卷积神经网络(CNN)可以很好地应用于这些领域,因此,针对 CNN 性能的改进成为许多学者关注的焦点<sup>[1]</sup>。CNN 通过局部感知与参

数共享,有效降低网络复杂度。该设计在针对图像识别、变形、扭曲、倾斜等变形形式具有较高的适应性和优越性<sup>[2]</sup>,通过将图像直接输入,避免了特征提取和图像重建等过程。因而 CNN 在语音识别、图像分析和文本分类等很多问题中表现出色,而且在很多具体的领域得到了普及应用。

CNN 结构特点如下:

1) CNN 包含卷积层、池化层、全连接层这三层基本网络结构,卷积层运算后,池化层开始进

收稿日期: 2018-09-23

基金项目: 吉林省科技厅科技发展计划基金资助项目(20160203010GX)

作者简介: 王金哲(1994—),男,汉族,山东滨州人,长春工业大学硕士研究生,主要从事数据挖掘方向研究,E-mail:1044135771@qq.com。\* 通讯作者:王红梅(1968—),女,汉族,吉林长春人,长春工业大学教授,博士,主要从事数据挖掘和分布式计算方向研究,E-mail:wanghm@ccut.edu.cn。

行,两者是交替进行计算的,同时将特征提取和分类过程相互结合在一起,有效地简化了传统算法中特征提取的步骤;

2) CNN 使用局部连接,单个的神经元仅仅和一小部分神经元相连,通过权重共享机制,可以有效地减少训练权重的数目,一定程度上防止过拟合问题;

3) 多层感知器包括输入层、隐含层和输出层这三个重要层,多层感知器对于图像平移、缩放、扭曲有良好的鲁棒性。

改进 CNN 算法<sup>[3]</sup>通过使用 GPU 来提升运算速度,但在 CPU 与 GPU 之间数据切换和进程的调度可能会出现进程中断的情况;文献[4]对于原有 CNN 训练时间较长和过拟合等问题提出基于免疫系统改进 CNN 网络,但是,这种改进在数据集测试仅有 81.6% 识别率;文献[5]对 MLP-CNN 这一经典模型进行改进,通过使用梯度下降算法和增加神经元特征数等来提高模型性能,但是这种改进会出现误差收敛缓慢的情况;文献[6]提出在 CNN 中加入波尔兹曼机制,并进行图像分类,该方法提升了图像分类的准确率,但是对系统内存要求比较高;文献[7]对正则化策略进行微调,以侦听卷积的滤波器和非线性激活函数,将会在一定程度上使得训练的时间有所增加;文献[8]在 CNN 中引入了双线性插值,可以对图像进行预测和分类,不过针对图片集过拟合现象没有提出切实可行的方案。

CNN 算法一般存在以下问题<sup>[9]</sup>:

1) CNN 采用梯度下降算法,在反向传播中的调节网络连接权重的过程中,误差将会缓慢下降,网络的收敛速度会受到不小的影响;

2) Sigmoid 激活函数使原来在较大范围内变化的输入数据映射到 $[0, 1]$ 内输出,在数据集的规模逐渐增大时,参数不易得到很好的训练,网络权重不能及时更新,从而减缓收敛的速度;

3) CNN 中常采用的 weight decay 方法与提前停止的方式缓解过拟合问题,不过解决过拟合现象的能力会随着学习层数目的增多逐渐下降。针对上述问题,为了可以让误差反传方法具有可靠的理论意义和重要的实用价值,采用 PSO 算法进行优化,提高 CNN 收敛速度,同时基于 Dropout 的改进 CNN 算法在很大程度上能够减少过拟合现象的发生。

## 1 相关理论

### 1.1 粒子群算法(PSO 算法)<sup>[10]</sup>

PSO 是一种比较经典的群体优化算法,它通过模仿鸟群行为来寻找最优值。PSO 算法中每个求解问题的解都作为多维搜寻空间的一个粒子,设定适应值函数,根据单个粒子自身获得的最优解,以及整体群簇所找到的最优值,通过不停地改变各粒子的位置和速度,最后通过迭代获得最优解。PSO 算法具有搜索速度快、算法易实现、精度较高等特点。基本 PSO 公式如下:

$$v'_{ij} = w \cdot v_{ij} + c_1 \cdot \text{rand}() \cdot (p_{ij} - x_{ij}) + c_2 \cdot \text{rand}() \cdot (p_{gj} - x_{ij}) \quad (1)$$

$$x'_{ij} = x_{ij} + v'_{ij} \quad (2)$$

### 1.2 卷积神经网络(CNN)

CNN 一般由输入层、卷积层、激励层、池化层、全连接层和输出层组成。它通过卷积运算和降采样等方法实现对有用信息的提取<sup>[11]</sup>。

输入层,将所要用到的数据或者图像进行输入,为方便卷积层运算,需要对输入数据做预处理。

卷积层,卷积层的卷积核主要对输入信息进行特征提取。

激励层,激活函数主要对运算的结果进行平滑处理,同时激活函数可以对网络稀疏化,清除冗余数据,保留数据主要特征。

池化层,整理卷积层输出信息,减少网络计算量,另外对特征进行压缩,提取更为重要的特征,防止过拟合现象。

全连接层,通过特征变换,将有用信息进行整合,对提取结果进行分类和预测等处理。

输出层,将前面运算的所得结果进行输出处理。

## 2 改进着眼点

### 2.1 PSO 算法在反向传播的改进

#### 2.1.1 CNN 算法前向传播

首先 CNN 进行前向传播,包含卷积和池化操作,对于卷积层的计算:

$$x_j^n = f\left(\sum_{i \in M_j} x_i^{n-1} K_{ij}^n + b_j^n\right) \quad (3)$$

式中: $M_j$  ——输入特征图集;

$x_j^n$  ——第  $n$  层的特征值  $j$ ;

$K_{ij}^n$  ——卷积核函数;

$f()$ ——激活函数,使用 Sigmoid 函数;

$b_j^n$ ——偏置参数。

卷积层与池化层进行交替计算,卷积层之后是池化层,对于池化层的计算为:

$$x_j^{n+1} = f\left(\sum_j x_j^n \omega_j^{n+1} + b_j^{n+1}\right) \quad (4)$$

式中: $\omega_j^n$ ——池化层特征图的权值常数。

输出层得到结果前,在  $n$  层的卷积神经中,作为输入样本,用  $f_n$  表示各层池化的激活函数, $w^{(n)}$  表示每个层的连接权重,这个过程的计算可以表示为:

$$y = f_n(\dots(f_2(f_1(x \cdot w^1)w^2))\dots)w^n \quad (5)$$

计算出前向传播的结果,和已给出的养殖的标签值作误差计算,误差函数表示为:

$$E = \frac{1}{n} \sum_{i=1}^N \sum_{j=1}^C (y_{ji}^d - y_{ji})^2 \quad (6)$$

式中: $N$ ——CNN 输入图像训练样本数;

$C$ ——网络输出层神经元数;

$y_{ji}^d$ ——第  $i$  个样本的第  $j$  个输出节点的期望输出值;

$y_{ji}$ ——第  $i$  个样本的第  $j$  个输出节点的实际输出值。

### 2.1.2 CNN 算法的后向传播

CNN 算法的后向传播是为了在计算期望输出和实际输出的误差后,将需要运算的参数作为

PSO 算法的粒子,利用式(6)获得粒子的局部最优 Pbest 和全局最优值 Gbest,然后使用式(1)和式(2)来更新粒子。迭代运算后,更新后的粒子是 CNN 权值,再对 CNN 网络重新进行前向传播,直至误差阈值收敛到最小范围之内,然后停止算法。

算法的实现过程如下:

输入: $m$ (粒子个数), $c_1$ (加速常数), $c_2$ (加速常数), $\omega$ (惯性权重), $x$ (粒子的位置), $v$ (粒子的速度)。

输出: $c\omega$ [(优化后的权值)。

1)对于群体内的每一个粒子,在 CNN 前向传播中使用式(5)计算,再由式(6)计算误差。

2)如果已经达到误差阈值的小值,停止算法,如果没有收敛,则执行式(1)与式(2)对粒子进行更新:

①将更新的粒子的信息传回 CNN 中,同时更新需要训练的权值,再次进行前向传播误差计算;

②回到 2)。

## 2.2 CNN 算法设计

### 2.2.1 CNN 结构设计

构建的 9 层 CNN 模型如图 1 所示。

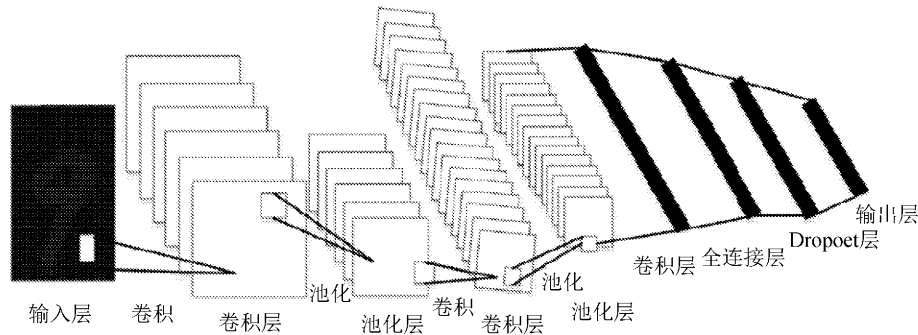


图 1 神经网络模型设计

图 1 由输入层、5 个由卷积层和池化层构成的隐含层、全连接层、Dropout 层和输出层组成。在这个新模型中,为了改善过拟合问题,将 Dropout 层添加到全连接层的后面,通过概率  $p$  摒弃神经元结点。

### 2.2.2 基于 Dropout 防止过拟合问题

为了提升网络的泛化能力,将会在 CNN 网络中增加 Dropout 层,在网络的训练过程中,对于神经网络中任意的一个神经元,它都是使用相同的概率  $p$  将其从网络中临时抛弃,从而减小了网络的大小,其公式为:

$$p = (p_a = 1 | x) = \sum_{i,j \in B_a} \frac{\exp(\omega_{ij}^n x_j^{n-1} + b_i^n)}{1 + \exp \sum_{i,j \in B_a} \omega_{ij}^n \cdot x_j^{n-1} + b_i^n} \quad (7)$$

式中: $p_a$ ——样本  $a$  生成 1 的概率;

$B_a$ ——第  $i$  类神经元中属于样本  $a$  的数量。

文中所提出的改进 CNN 结构,具体算法如下:

输入:训练集,测试集;

输出:识别结果。

1)对神经网络滤波器进行训练,同时初始化滤波器尺寸像素为  $P_1 * P_2$ ;

2)对图片集进行训练,将训练集中的图片经过处理,转换成尺寸大小一致的图片,同时把图片读入到数据矩阵  $P$  中;

3)初始化权重  $w_{ij}^n$  和偏置  $b_i$ ;

4)使用式(4)中所给的公式对第一层卷积特征矩阵  $P^{(1)}$  进行计算;

5)通过调用 PSO 算法,来对权值  $w_i$  进行更新,从而获得特征矩阵  $P^{(2)}$ ;

6)通过 4)与 5)进行第二次卷积计算,这样会得到一个新的特征矩阵  $P^{(3)}$ ;

7)将特征矩阵  $P^{(3)}$  作为一个列向量神经元,输入到全连接层,而后将这个输入乘以权重矩阵  $w$ ,然后添加偏置,对其运用激活函数,从而获得特征向量  $b_1$ ;

8)将全连接层得到的结果作为 Dropout 层的输入,通过式(7)运算获得神经元在 Dropout 层中的输出概率,得到特征向量  $b_2$ ;

9)将 8)得到的特征向量  $b_2$  作为输出层的输入,通过输出层 Softmax 分类器进行分类,最后输出识别结果。

### 3 实验与分析

#### 3.1 测试环境

为了评估所改进算法的性能,在 ubuntu16.04 的操作系统中采用 Python3 语言编程来实现文中所提出 PD-CNN 算法。测试平台为 6 台相同配置的电脑,配置为:AMD10CPU@2.5 GHz, GPU 为 AMD Radeon HD 8670M + AMD Radeon HD 8610G,8.00 GB 内存。文中使用的测试数据集包括 MNIST 手写体数据集与 HCL200 书写汉字数据集。实验次数为 10 次。

#### 3.2 比较算法

文中改进算法和 WCNN 算法、MLP-CNN

算法及 SVM-ELM 算法进行比较。其中 WCNN 算法为最近邻规则的神经网络模型。为了实现对输入的处理,使用了多个卷积层和池化层,并在连接层和输出结果之间建立了映射关系,最后使用聚类算法对特征分类。算法 MLP-CNN 主要是增加网络特征数和使用梯度下降算法进行优化,来提高网络模型的识别能力。SVM-ELM 是基于支持向量机的优化算法,可以减少隐层节点数,通过 SVM 优化每个结点的线性决策函数。以上 3 个算法都是比较经典的 CNN 改进算法,通过与以上 3 个 CNN 改进算法进行测试比较,可以有效地说明文中提出的改进算法在性能上有很大的提高。

#### 3.3 测试结果与分析

针对 MNIST 训练数据集运行 10 次,在不同数据规模下各算法的识别结果如图 2 所示。

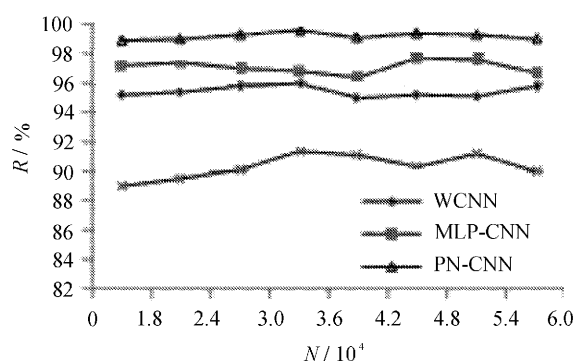


图 2 各算法针对 MNIST 数据集的识别结果

由图 2 可知,PD-CNN 算法对于不同的数据规模识别率高于 SVM-ELM、MLP-CNN 与 WCNN 算法。数据规模的增大,使得 MLP-CNN 算法对图片的识别率影响较大,识别性能波动幅度增加。当数据集的规模小于等于  $3.6 \times 10^4$  时,随着数据规模的增大,WCNN 算法与 PD-CNN 算法识别率呈上升趋势,但是,当数据规模大于  $3.6 \times 10^4$  时,识别率有下降趋势,随着数据集的增大,WCNN 算法与 MLP-CNN 算法识别率波动大于 PD-CNN 算法,说明了 PD-CNN 算法具有较好的鲁棒性。

对每个测试数据集运行 10 次,在最小识别率  $R_M$ 、最大识别率  $R_X$ 、平均识别率  $R$  与运行时间  $t$  进行比较算法性能结果统计。

算法性能统计结果见表 1。

表 1 算法性能统计结果

数据集	指标/%	WCNN	MLP-CNN	SVM-ELM	PD-CNN
MNIST	$R_m$	95.1	96.4	89.1	98.9
	$R_x$	96.0	97.7	91.3	99.6
	$R$	95.45	97.15	90.26	99.35
	$t/s$	193.86	301.45	532.63	126.32
HCL2000	$R_m$	84.2	88.5	83.6	92.1
	$R_x$	91.6	92.3	90.0	93.3
	$R$	89.61	90.77	88.63	92.96
	$t/s$	367.86	789.62	1132.45	254.53

通过观察表 1 可知,在数据集 MNIST 上,相比较于 WCNN、MLP-CNN 和 SVM-ELM 算法,PD-CNN 算法在最小识别率  $R_m$ 、最大识别率  $R_x$  与平均识别率  $R$  上最高,表明了文中改进算法的可行性以及正确性。使用 PSO 算法在梯度下降中训练 CNN,在识别精度与收敛速度高于常规的训练方法,通过在模型上添加一层 Dropout 层,防止可能出现的过拟合问题,同时可以使得 CNN 的正确率得到较大的提升。最后在运行时间上与另外三个 CNN 改进算法进行比较,可以看到 PD-CNN 算法时间小于三种算法的时间,说明 PSO 算法在反向传播中提高了收敛速度。

#### 4 结 语

针对 CNN 算法的收敛速度较慢、过拟合等问题,将 PSO 算法中需要训练的参数作为粒子进行优化,将更新的参数用于 CNN 算法的前向传播,调整网络连接权矩阵迭代,直到误差收敛,停止算法。同时设计新的神经网络模型,将一层 Dropout 层添加到输出层与全连接层之间,用于缓解过拟合风险,在识别率上得到了不小的提升,同时具有良好的鲁棒性。最后,通过与经典改进 CNN 算法 WCNN、MLPCNN 和 SVM-ELM 算法在数据集 MNIST 与 HCL200 进行实验比较,PD-CNN 算法在识别率、收敛速度均高于另外三个算法,证明了改进算法的正确性。

#### 参考文献:

[1] 周飞燕,金林鹏,董军.卷积神经网络研究综述[J].计算机学报,2017,40(6):1229-1251.  
 [2] Vieira S, Pinaya W H L, Mechelli A. Using deep learning to investigate the neuroimaging correlates

of psychiatric and neurological disorders: methods and applications[J]. Neuroscience and Biobehavioral Reviews,2017,74:58-75.

- [3] Li S, Dou Y, Niu X, et al. A fast and memory saved GPU acceleration algorithm of convolutional neural networks for target detection[J]. Neurocomputing,2016,230:48-59.  
 [4] Gong T, Fan T, Guo J, et al. GPU-based parallel optimization of immune convolutional neural network and embedded system[J]. Engineering Applications of Artificial Intelligence,2016,36(25):226-238.  
 [5] Singh P, Verma A, Chaudhari N S. Deep convolutional neural network classifier for handwritten Devanagari character recognition[C]// Proc of Information Systems Design and Intelligent Applications. New Delhi: Springer,2016:551-561.  
 [6] 石祥滨,房雪健,张德园,等.基于深度学习混合模型迁移学习的图像分类[J].系统仿真学报,2016,28(1):167-173.  
 [7] Sun W, Su F. A novel companion objective function for regularization of deep convolutional neural networks[J]. Image and Vision Computing,2016,56(36):110-126.  
 [8] Chen L C, Papandreou G, Kokkinos I, et al. Semantic image segmentation with deep convolutional nets and fully connected CRFs[J]. Computer Science,2016,26(4):357-361.  
 [9] Almási A, Woniak S, Cristea V, et al. Review of advances in neural networks:Neural design technologystack[J]. Neurocomputing,2016,174:31-41.  
 [10] 王泽儒,李芬田,王红梅.粒子群算法中惯性权重参数[J].长春工业大学学报,2017,38(4):354-360.  
 [11] 杨斌,钟金英.卷积神经网络的研究进展综述[J].南华大学学报:自然科学版,2016,30(3):66-72.