

引用格式:李仲德,卢向日,崔桂梅. 数字图像识别的代价函数选择和性能评价[J]. 电光与控制, 2019, 26(12):44-48, 68. LI Z D, LU X R, CUI G M. Cost function selection and performance evaluation for digital image recognition[J]. Electronics Optics & Control, 2019, 26(12):44-48, 68.

数字图像识别的代价函数选择和性能评价

李仲德, 卢向日, 崔桂梅
(内蒙古科技大学信息工程学院, 内蒙古 包头 014010)

摘要: 针对传统二次代价函数在卷积神经网络训练过程中图像识别准确率不高的问题, 提出基于交叉熵代价函数的卷积神经网络算法。经数学推导, 证明交叉熵代价函数较二次代价函数在图像识别中精度更高, 并应用 MNIST 数据集和 CIFAR-10 数据集, 使用 AlexNet 卷积神经网络, 分别采用二次代价函数和交叉熵代价函数对图像识别模型进行训练, 当数字图像识别精确率和损失值稳定后, 使用测试数据对代价函数进行多次测试, 对比识别准确率。仿真结果表明, 此方法不仅能提高数字图像识别的准确率, 而且相较于传统的代价函数, 训练模型速度更快, 明显缩减了训练深度神经网络模型的过程。

关键词: 卷积神经网络; 交叉熵代价函数; 二次代价函数; 手写数字数据集; CIFAR-10 数据集
中图分类号: TP301.6 **文献标志码:** A **doi:**10.3969/j.issn.1671-637X.2019.12.009

Cost Function Selection and Performance Evaluation for Digital Image Recognition

LI Zhong-de, LU Xiang-ri, CUI Gui-mei

(School of Information Engineering, Inner Mongolia University of Science and Technology, Baotou 014010, China)

Abstract: In order to solve the problem that the accuracy of image recognition is not high in the training process of convolutional neural network by using traditional quadratic cost function, a convolutional neural network algorithm based on cross-entropy cost function is proposed. By mathematical derivation, it is proved that the cross-entropy cost function is more accurate than the quadratic cost function in image recognition. Based on MNIST dataset and CIFAR-10 dataset, and using AlexNet convolutional neural network, the quadratic cost function and the cross-entropy cost function are adopted to train the image recognition model respectively. When the recognition accuracy and loss value of the digital image are stable, the cost function is tested several times by using the test data, and comparison is made to the recognition accuracy of the two functions. The simulation results show that the proposed method can not only improve the accuracy of digital image recognition, but also has a faster model training speed than the traditional cost function. The process of training deep neural network model is obviously shortened.

Key words: convolutional neural network; cross-entropy cost function; quadratic cost function; MNIST dataset; CIFAR-10 dataset

0 引言

近年来,深度学习在图像处理、语音识别等领域取得了前所未有的发展^[1],这种成就要归功于 HINTON 教授根据人脑认知机理,在 2006 年提出深度神经网络 (Deep Neural Network, DNN);在 2012 年 ImageNet 大规

模视觉识别挑战赛 (ILSVRC) 中, HINTON 及其学生 KRIZHEVSKY 设计的 AlexNet 神经网络在图像识别中准确率达 83.6%, 远超其他队, 获得冠军, 使得深度学习再一次兴起; 2015 年, 微软亚洲研究院提出了深度残差网络, 并在 ImageNet 的分类比赛中取得了 3.57% 的错误率; 2016 年, GAN 用于对视频输入进行降维然后用低维表达和 LSTM 处理图像画面^[2-3]。随着图像采集设备像素的不断提高以及工作场景的日益复杂, 图像中所包含的信息量越来越大, 研究如何从这些信息中更加准确、快速地提取出所关注目标的特征是当

收稿日期: 2019-01-02 修回日期: 2019-02-21

基金项目: 国家自然科学基金(61563041)

作者简介: 李仲德(1976—), 男, 河北顺平人, 博士, 讲师, 硕导, 研究方向为机器人自主控制、计算机控制。

今研究的热点^[4]。现今的神经网络与最初的浅层神经网络相比,虽然在整体性能上优于以往的初级神经网络,但仍存在神经网络训练陷入局部最优、图像识别准确率不高的问题,对于解决上述描述的客观实际情况提出了交叉熵代价函数模型。

1 概念介绍

1.1 视觉识别数据集的介绍

1) MNIST 数据集。

MNIST 数据集是美国 Google 实验室和纽约大学柯朗研究所创建的一个手写数字数据库,如图 1 所示,其训练库有 60 000 幅手写数字图像,测试库有 10 000 幅数字图像,每幅数字图像在计算机中均以字节形式存放^[5]。



图 1 MNIST 数据集
Fig. 1 MNIST dataset

2) CIFAR-10 数据集。

该数据集共有 60 000 幅彩色图像,分为 10 类,每类 6000 幅图。如图 2 所示,CIFAR-10 数据集中 50 000 幅

用于训练,另外 10 000 幅用于测试。

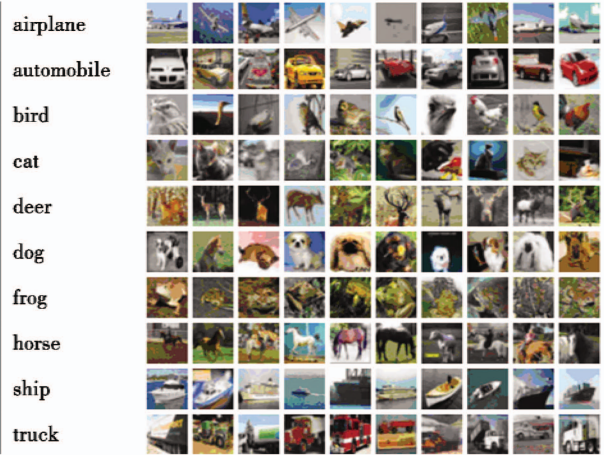


图 2 CIFAR-10 数据集
Fig. 2 CIFAR-10 dataset

MNIST 数据集和 CIFAR-10 数据集是深度学习算法,如 AlexNet, GoogleNet, VGGNet, ResNet 图像识别算法研究领域有效的小型数据集检验对象,故本文使用 AlexNet 卷积神经网络,在 MNIST 数据集和 CIFAR-10 数据集中训练图像识别模型^[6]。

1.2 卷积神经网络模型

AlexNet 神经网络属于卷积神经网络的一种,其结构如图 3 所示^[5-6]。卷积层提取图像特征,经代价函数计算,最终输出误差进行评价并计算出图像识别的准确率与损失值。

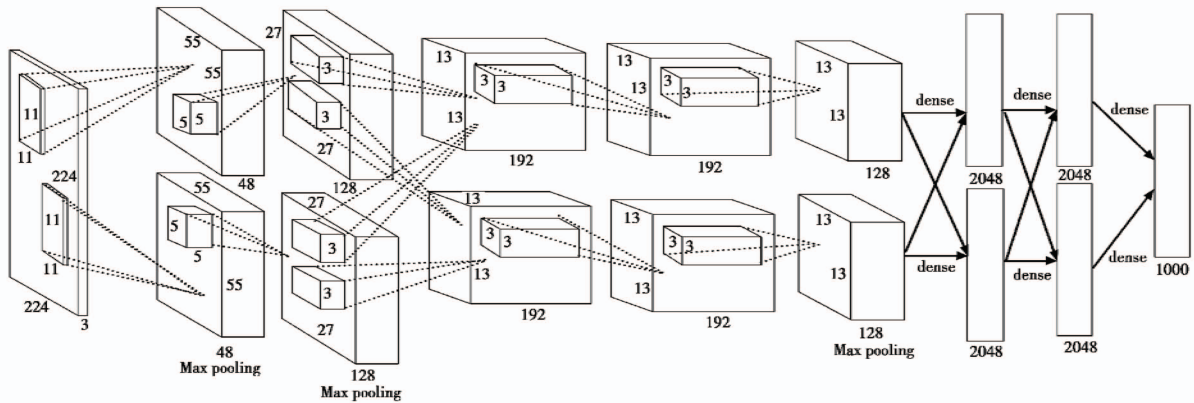


图 3 AlexNet 结构图
Fig. 3 AlexNet structure diagram

2 卷积神经网络数据预处理

在数字图像模型训练中,首先要经过 AlexNet 神经网络卷积层对数据进行预处理,即图像的像素矩阵和卷积层的核函数矩阵进行卷积^[7],利用卷积提取图像的局部特征,卷积处理如图 4 所示。

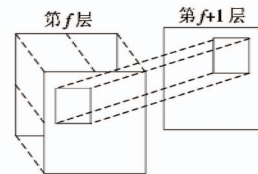


图 4 卷积处理示意图
Fig. 4 Schematic diagram of convolution processing

第 $f+1$ 层的神经元内部算式为

$$z = \sum_m W_{mn}^f x + b_m^{f+1} \quad (1)$$

式中: z 为第 $f+1$ 层图像特征图的输入; W_{mn}^f 表示在第 f 层第 m 个神经元与第 n 个神经元之间的权重连接; b_m^{f+1} 表示在 $f+1$ 层的第 m 个神经元的偏置; x 表示初始层的输入。

特征图的算式为

$$L \times W = ((L_m - l) / s + 1) \times ((W_m - w) / s + 1) \quad (2)$$

式中: $L \times W$ 表示输出特征图的尺寸; L_m 表示第 j 个神经元的长; W_m 表示第 j 个神经元的宽; s 表示滑动窗口的步长。

为了实现非线性函数映射的任意逼近,并使图像识别准确率输出在 $[0, 1]$ 之间,本文采用 Sigmoid 函数作为 AlexNet 神经网络卷积层的激活函数^[8-9],即

$$\sigma(z) = 1 / (1 + e^{-z}) \quad (3)$$

其导函数为

$$\sigma'(z) = \sigma(z)(1 - \sigma(z))。 \quad (4)$$

当 Sigmoid 函数的输出值接近 1 或者 0 时,网络权重和偏置项会出现梯度消失、识别精确率不高的情况,这时,交叉熵函数在梯度消失的情况下仍然可以学习到数据,进而解决网络陷入局部最优问题^[10]。

3 代价函数原理与算法

3.1 代价函数原理

传统的 AlexNet 神经网络图像识别算法大多采用二次代价函数

$$C = (y - a)^2 / 2 \quad (5)$$

式中: y 是期望输出即实际值; a 是神经元的输出, $a = \sigma(z)$, $z = wx + b$; C 是输出的概率值。

使用链式法则求权重和偏置的导数如

$$\frac{\partial C}{\partial w} = (a - y) \sigma'(z) x \quad (6)$$

$$\frac{\partial C}{\partial b} = (a - y) \sigma'(z)。 \quad (7)$$

从式(6)、式(7)的链式法则可以得出,激活函数持续为 0 或 1,其导数为 0,神经网络停止学习。对于上述问题,通过消除式(6)和式(7)中的导数项 $\sigma'(z)$,可以在激活函数接近或持续为 0 或者 1 时,梯度存在且能保持较高的学习速率对网络参数进行更新。

首先将式(6)和式(7)中的导数项去掉,可得

$$\frac{\partial C}{\partial w} = (a - y) x \quad (8)$$

$$\frac{\partial C}{\partial b} = (a - y)。 \quad (9)$$

联立式(5)、式(7)可得

$$\frac{\partial C}{\partial b} = (a - y) \sigma'(z) = \frac{\partial C}{\partial a} \sigma'(z) \quad (10)$$

由 $\sigma'(z) = \sigma(z)(1 - \sigma(z)) = a(1 - a)$ 可得

$$\frac{\partial C}{\partial b} = \frac{\partial C}{\partial a} a(1 - a) \quad (11)$$

联立式(9)、式(11)可得

$$\frac{\partial C}{\partial a} = \frac{a - y}{a(1 - a)}。 \quad (12)$$

对式(12)左右两端以 a 为变量进行积分,可得单个神经元的交叉熵代价函数为

$$C = -[y \ln a + (1 - y) \ln(1 - a)] + C_v \quad (13)$$

式中, C_v 是常数。

若是 n 个神经元进行训练,则所有的训练样本需进行平均,此时代价函数为

$$C = -\frac{1}{n} \sum_x [y \ln a + (1 - y) \ln(1 - a)] + C_v。 \quad (14)$$

3.2 数字图像识别代价函数的伪代码实现

根据第 2 章的介绍,数字图像识别代价函数的伪代码算法^[11-12]实现步骤如下所述。

```

{
    损失值 = (期望输出 - a)2 / 2 #二次代价函数
    损失值 = -[期望输出 ln a + (1 - 期望输出) ln(1 - a)] #交叉熵代价函数
    If
        期望输出 = 1
        损失值 = -[期望输出 ln a]
    else
        损失值 = -[(1 - 期望输出) ln(1 - a)]
    #使用梯度下降法
    训练过程 = SGD(学习率 = 0.2)
    #求准确率
    准确率 = 求平均(total(1 - 损失值))
    for 周期 in range(20):
        for batch in range(训练样本总数):
            MNIST/CIFAR-10. train/test. (批量训练数量)
    End
}

```

4 算法仿真研究和性能分析

将本文提出的数字图像识别的代价函数算法在 Intel(R) Core(TM) i7-2670QM, CPU 主频为 2.20 GHz, 内存为 8 GiB 的硬件实验平台和软件 Jupyter Notebook 仿真平台上采用 Python 语言编程^[13]实现,并利用 MNIST 数据集和 CIFAR-10 数据集中的样本对算法的有效性进行分析研究。

4.1 仿真结果定量分析

利用交叉熵代价函数和二次代价函数分别对 MNIST 和 CIFAR-10 数据集的标准数字图像训练样本进行训练研究^[14]。

4.1.1 验证 MNIST 数据集

图 5 是分别采用两种算法训练 MNIST 数据集图像的识别准确率对比;图 6 是 MNIST 数据集图像识别损失值结果对比。

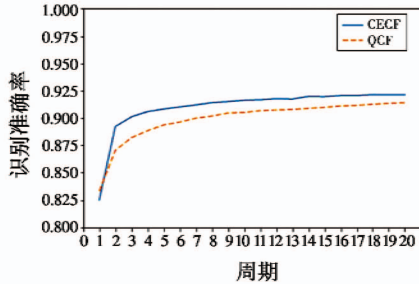


图 5 MNIST 数据集图像识别准确率结果对比
Fig. 5 Comparison of the accuracy of image recognition on MNIST dataset

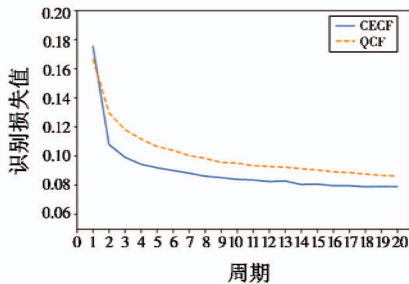


图 6 MNIST 数据集图像识别损失值结果对比
Fig. 6 Comparison of loss values of image recognition on MNIST dataset

从仿真结果可以看出,交叉熵代价函数比二次代价函数具有更高的识别准确率和更低的损失值,而且算法的收敛速度更快,学习能力更强。

利用训练好的神经网络对 MNIST 数据集集中的测试样本进行图像识别研究。测试结果如表 1 中 MNIST 数据集多周期识别准确率所示。

表 1 MNIST 数据集多周期识别准确率

Table 1 Recognition accuracy of MNIST dataset in multiple tests

次数	二次代价函数	交叉熵代价函数	次数	二次代价函数	交叉熵代价函数
1	0.9131	0.9103	6	0.9078	0.9177
2	0.9171	0.9178	7	0.9162	0.9231
3	0.9178	0.9198	8	0.9078	0.9298
4	0.9064	0.9293	9	0.9131	0.9237
5	0.9063	0.9289	10	0.9121	0.9267

4.1.2 验证 CIFAR-10 数据集

图 7 是分别采用两种算法训练 CIFAR-10 数据集

图像的识别准确率对比;图 8 是 CIFAR-10 数据集图像识别损失值结果对比。

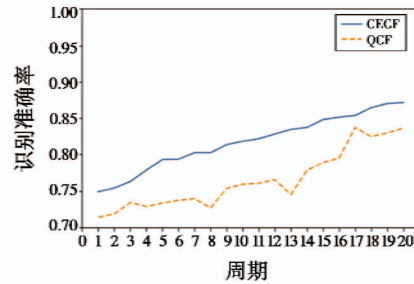


图 7 CIFAR-10 数据集图像识别准确率结果对比
Fig. 7 Comparison of accuracy of image recognition on CIFAR-10 dataset

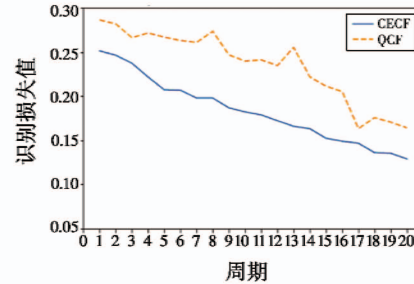


图 8 CIFAR-10 数据集图像识别损失值结果对比
Fig. 8 Comparison of loss values of image recognition on CIFAR-10 dataset

由仿真结果同样可以看出,交叉熵代价函数比二次代价函数具有更高的识别准确率和更低的损失值,而且算法的运算过程稳定性更强。

利用训练好的神经网络,对 CIFAR-10 数据集集中的测试样本进行图像识别研究。测试结果如表 2 中 CIFAR-10 数据集多周期识别准确率所示。

表 2 CIFAR-10 数据集多周期识别准确率

Table 2 Recognition accuracy of CIFAR-10 dataset in multiple tests

次数	二次代价函数	交叉熵代价函数	次数	二次代价函数	交叉熵代价函数
1	0.8141	0.8212	6	0.8078	0.8277
2	0.8121	0.8278	7	0.8174	0.8331
3	0.8148	0.8298	8	0.8058	0.8298
4	0.8064	0.8193	9	0.7931	0.8237
5	0.7943	0.8189	10	0.8141	0.8267

仿真结果表明,当 MNIST 数据集将测试图像扩展为全体样本且加入 CIFAR-10 数据集作为测试对象时,交叉熵代价函数依然能够保持很高的识别准确率,相较于传统方法更具有优势。

4.2 统计学定性分析

根据 4.1 节算法仿真研究,利用统计学分析的方法分别对两种算法测试不同数据集的准确率进行分析。

4.2.1 MNIST 数据集仿真结论定性分析

MNIST 数据集仿真结果如表 3、表 4 所示。

表3 MNIST数据集描述性统计量

Table 3 Descriptive statistics of MNIST dataset

对比项	二次代价函数	交叉熵代价函数
均值	0.911770	0.922710
标准差	0.0044512	0.0063311
极小值	0.9063	0.9103
极大值	0.9178	0.9298

由表3可知:两种代价函数算法的标准差值很小,说明数据可靠性好,而且标准差均不符合正态分布规律,则采用Friedman检验。Friedman检验假设如下:1) H_0 表示代价函数的MNIST数据集数据有效性无明显差异;2) H_1 表示代价函数的MNIST数据集数据有效性有明显差异。

对上述两个神经网络的测试结果进行分析,得到表4中Friedman检验统计量。

表4 MNIST数据集Friedman检验统计量

Table 4 Friedman test statistics of MNIST dataset

样本数量	卡方分布值	自由度	渐进显著性
10	6.400	1	0.011

根据常用显著性水平 $\alpha = 0.05$, 计算 $\chi_{0.05}^2(9) = 16.919 > 6.400$, 所以两组数据之间相互独立且本组数据特征明显。

4.2.2 CIFAR-10数据集仿真结论定性分析

CIFAR-10数据集仿真结果如表5、表6所示。

表5 CIFAR-10数据集描述性统计量

Table 5 Descriptive statistics of CIFAR-10 dataset

对比项	二次代价函数	交叉熵代价函数
均值	0.807990	0.825800
标准差	0.0084523	0.0048206
极小值	0.7931	0.8189
极大值	0.8174	0.8331

通过交叉熵代价函数和二次代价函数对CIFAR-10数据集仿真研究可知,虽然均值下降,但是由于CIFAR-10数据集图像信息量相较于MNIST数据集增大造成的。二次代价函数的标准差增加,但是交叉熵代价函数比传统方法要稳定,而且也证明了交叉熵代价函数在遇到复杂图像环境时的鲁棒性更好。

表6 CIFAR-10数据集Friedman检验统计量

Table 6 Friedman test statistics of CIFAR-10 dataset

样本数量	卡方分布值	自由度	渐进显著性
10	10.000	1	0.002

同理,根据常用显著性水平 $\alpha = 0.05$, 计算 $\chi_{0.05}^2(9) = 16.919 > 10.000$, 所以两组数据之间相互独立且本组数据特征明显。

综上所述,相较于传统的代价函数方法,虽然交叉熵代价函数的编程实现更繁琐,而且消耗硬件内存更

多,但是采用交叉熵代价函数的神经网络训练速度更快,在处理MNIST数据集时图像识别准确率平均提高1.1%;在处理CIFAR-10数据集时图像识别准确率平均提高2.2%。交叉熵代价函数在数字图像识别上更具优势。

5 结束语

对模型训练中常用的二次代价函数和交叉熵函数进行理论推导后,本文采用交叉熵代价函数和二次代价函数分别对AlexNet卷积神经网络进行训练,从图像识别的准确率以及损失值的结果来看,交叉熵代价函数更适合作为神经网络识别MNIST数据集和CIFAR-10数据集图像的代价函数。结果表明,交叉熵代价函数相比于传统的方法有更快的收敛速度,加快了训练神经网络的过程,并且交叉熵代价函数的运用有效地提升了MNIST数据集和CIFAR-10数据集图像识别的准确率。

参考文献

- [1] 周志华. 机器学习[M]. 北京:清华大学出版社,2016.
- [2] 刘鹏,张燕,赵海峰. 深度学习[M]. 北京:电子工业出版社,2018.
- [3] 张旭,谢鉴. 基于深度学习的人脸大数据系统应用[J]. 电子技术与软件工程,2019(3):161.
- [4] 叶韵. 深度学习与计算机视觉—算法原理、框架应用与代码实现[M]. 北京:机械工业出版社,2017.
- [5] 柯岩,林小竹,廖蕊,等. 卷积神经网络的“深度”研究[EB/OL]. [2019-02-13]. <https://kns.cnki.net/KCMS/detail/31.1289.tp.20190124.1507.007.html>.
- [6] 罗浩,姜伟,范星,等. 基于深度学习的行人重识别研究进展[EB/OL]. [2019-02-13]. <https://kns.cnki.net/KCMS/detail/11.2109.TP.20190122.0919.007.html>.
- [7] NIELSEN M. Neural networks and deep learning [EB/OL]. [2019-01-02]. <http://neuralnetworksanddeeplearning.com>.
- [8] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition [R/OL]. (2015-12-10) [2019-01-02]. <https://arxiv.org/abs/1512.03385>.
- [9] 王力,周志杰,赵福均. 基于BP神经网络和证据理论的超声检测缺陷识别[J]. 电光与控制,2018,25(1):65-69.
- [10] 王灵刚,张蕾,普杰信,等. 改进BP神经网络在物体识别中的应用[J]. 电光与控制,2012,19(4):68-71.

(下转第68页)

- on Pattern Analysis and Machine Intelligence, 2016, 38 (10):2096-2109.
- [4] KALAL Z, MIKOLAJCZYK K, MATAS J. Tracking-learning-detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7):1409-1422.
- [5] BOLME D S, BEVERIDGE J R, DRAPER B A, et al. Visual object tracking using adaptive correlation filters [C]//IEEE Conference on Computer Vision and Pattern Recognition, 2010:2544-2550.
- [6] HENRIQUES J F, CASEIRO R, MARTINS P, et al. High-speed tracking with kernelized correlation filters [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3):583-596.
- [7] DANELLJAN M, HAGER G, KHAN F S, et al. Accurate scale estimation for robust visual tracking[C]//Proceedings of British Machine Vision Conference, 2014:1-11.
- [8] DANELLJAN M, ROBINSON A, KHAN F S, et al. Beyond correlation filters: learning continuous convolution operators for visual tracking [C]//Proceedings of European Conference on Computer Vision, 2016:472-488.
- [9] 韩中庚. 数学建模方法及其应用[M]. 北京:高等教育出版社,2005.
- [10] 逯程,徐廷学,王虹. 基于改进 GM(1,1)模型的装备故障预测[J]. 火力与指挥控制,2018,43(10):135-138.
- [11] WU Y, LIM J, YANG M H. Online object tracking: a benchmark[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2013. doi: 10.1109/CVPR.2013.312.
- [12] WU Y, LIM J, YANG M H. Object tracking benchmark [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9):1834-1848.
-
- (上接第 48 页)
- [11] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[C]//Neural Information Processing Systems Conference, 2012. doi:10.1145/3065386.
- [12] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition [R/OL]. (2015-04-10) [2019-01-02]. <https://arxiv.org/abs/1409.1556>.
- [13] 塔里克·拉希德. Python 神经网络编程[M]. 林赐,译. 北京:人民邮电出版社,2018.
- [14] SZEGEDY C, LIU W, JIA Y Q, et al. Going deeper with convolutions[R/OL]. (2014-09-17) [2019-01-02]. <https://arxiv.org/abs/1409.4842>.