文章编号: 1001-0920(2019)06-1259-06 DOI: 10.13195/j.kzyjc.2017.1556

基于GoogLeNet模型的剪枝算法

彭冬亮, 王天兴†

(杭州电子科技大学 自动化学院, 杭州 310018)

摘 要: GoogLeNet包含多个并行的卷积层和池化层,极具表现力,但也导致其参数数量冗余和计算量大,解决该问题的根本途径是将网络稀疏化. 剪枝算法通过训练网络、修剪低权重连接和再训练网络三步操作,只保留卷积层和完全连接层中的强相关连接,实现简化网络结构和参数数量的效果,获得近似的网络模型,不影响网络后验概率估计的准确性,达到压缩效果. 传统计算方式不适合非均匀稀疏数据结构,所提出的阈值剪枝算法设定合适的阈值,将原始GoogLeNet模型中将近1 040万参数减少到65万,大约压缩了16倍. 原始网络在进行剪枝处理后,准确率会有所降低,但经过少数次迭代,网络的准确率与原始模型不相上下,达到了压缩模型的效果,验证了阈值剪枝算法对改进GoogLeNet模型训练过程的有效性.

关键词: 剪枝算法; GoogLeNet; Inception模块; 权重阈值; 参数冗余; 过拟合

中图分类号: TP3 **文献标志码:** A

Pruning algorithm based on GoogLeNet model

PENG Dong-liang, WANG Tian-xing†

(College of Automation, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: GoogLeNet contains a number of parallel convolutional layers and pooled layers, which makes the network highly expressive and leads to redundant and computational quantity of GoogLeNet parameters. The fundamental way to solve the problem is to rarefly network. A pruning algorithm consists of three steps: training the network, pruning the low-weight connection, and retraining the network, retaining the strong correlation between the convolution layer and the fully connected layer. It reduces the number of network structure and parameters to obtain an approximate network model and does not affect the accuracy of the network posterior probability estimation in order to achieve compression. The traditional calculation methods are not suitable for the non-uniform and sparse data structures. The threshold pruning algorithm is proposed, which sets a suitable threshold and reduces the original 10.4 million parameters in the original GoogLeNet model to 650,000, approximately 16 times compressed. After pruning the original network, the accuracy rate will be reduced. Then the accuracy of the network is comparable with the original model through a few iterations, and the effect of the compressed mode is achieved. It is proved that the proposed threshold pruning algorithm can effectively improve the GoogLeNet model training process.

Keywords: pruning algorithm; GoogLeNet; Inception module; weight threshold; parameter redundancy; overfitting

0 引 言

2014年ILSVRC挑战赛中,GoogLeNet[1]因其特有的Inception模块获得了冠军. Inception中包含多个并行的卷积层和池化层,使得神经网络极其具有表现力,可以学习和映射其输入与输出之间非常复杂的关系. 为了提高深层神经网络的性能,人们不断扩充网络的层级数量以延伸网络深度,或者增加每个层级中的单元数量以拓展网络宽度. 然而,更大的尺寸通常意味着更多的参数,扩大的网络更容易过拟合,特别是当训练集中标记示例的数量有限时,创建高质量的训练集可能非常棘手,这已经成为神经网络发展的一个主要瓶颈.

当使用有限的训练数据时,许多采样噪声也对网络模型训练生成的复杂映射产生了影响,即使它们与训练集都服从同分布,这些关系也将在后续的数据测试中产生误差. 针对这一问题,研究人员已经提出了许多优化方法,其中包括训练集与测试集准确率偏差变大前停止训练,或是引入诸如L_1和L_2正规化以及权重共享和各种权重惩罚项[2].

解决这个问题的根本途径是将网络架构从完全连接转变为稀疏连接. 除了模仿生物系统外,Arora

收稿日期: 2017-11-17; **修回日期:** 2018-05-31.

基金项目: 国家自然科学基金项目(61703128); 近地探测重点实验室基金项目(614241404030717).

†通讯作者. E-mail: hdu_wtx@163.com.

等[3]所做的开创性工作成为了该方法的坚实理论基础,其主要结果表明,如果数据集的概率分布可以表示为庞大而稀疏的深层神经网络,则可以通过分析最后一层激活的相关统计和具有高度相关输出的聚类神经元,逐层构建最优网络拓扑. 赫布学习规则指出,神经元连接间的激活水平能够改变权值,而变化的量与两个神经元的活性之和成正比,这种学习的结果是使网络能够提取训练集的统计特性,从而将输入信息按照其相似性程度划分为若干类. 两者共同阐明了,即使严格的数学证明需要非常苛刻的条件,但在不太严格的现实条件下,实践应用中也是适用的.

网络剪枝是一项网络稀疏化技术,可以防止过拟合,为拓展网络规模和加快训练速度提供了可能性. 网络剪枝是指当训练一个较大神经网络时,移除部分不需要连接的方法. 通过移除一个单元,将其和其所有的传入传出连接从网络中删除. 同时,较大尺寸的网络架构对初始训练条件较为敏感,而修剪网络能够降低网络的复杂度,有利于提升泛化能力. 不足之处在于,当对非均匀稀疏数据结构进行数值计算时,传统的计算基础设施非常低效,即使算术运算的数量减少了100倍,查找和缓存的消耗也非常巨大,以至切换到稀疏矩阵得不偿失. 通过使用改进的高度调优的数据库,允许极快的密集矩阵乘法,利用底层CPU或GPU硬件的特性,导致这一差距进一步扩大[4-5]. 此外,不均匀稀疏模型需要更复杂的工程和计算基础设施. 本文调用了CUDA的cuSPARSE库,该库主要是为了优化稀疏矩阵的运算,提供了很多方便易用的接口.

本文介绍了GoogLeNet模型以及其特有的Inception模块,阐明了采用的剪枝原理和操作细节,分别对其卷积层、全连接层和Inception模块进行剪枝操作. 通过对比,剪枝网络在保持原有模型性能的前提下,降低了计算量和存储空间,甚至在某些方面,剪枝后的模型比原始模型更好. 由于剪枝算法主要针对卷积层和全连接层,它可以普遍地应用于各种卷积神经网络模型. 给出了仿真实验结果,将不同层级进行剪枝,并与原始模型进行比较,分析了剪枝算法对GoogLeNet模型性能的影响. 最后总结全文,证明了剪枝算法对优化神经网络模型性能和训练过程的合理性和有效性.

1 剪枝原理

1.1 剪枝算法

在Cun等[6]提出的最优脑损伤(Optimal brain damage)算法中,将参数的显着性定义为由于删除该参数而引起目标函数的变化. 从这个定义直接评估每个参数显着性,即暂时删除每个参数并重新评估目标函数,简化目标函数E,有

$$E=\frac{1}{2}\sum_i h_{ii}u_i^2. \tag{1}$$

其中:h_{ii}为对角二阶导数,u_i为待测参数. 计算每个参数的收益s_i,依照显着性对参数进行排序,并删除一些低显着性参数,有

$$s_i=\frac{h_{ii}u_i^2}{2}. \tag{2}$$

这一过程相当费时费力,需要一种有效计算h_{ii}的方法. OBS算法在OBD算法上做了一定的改进[7],由于Hessian矩阵上有许多非对角线项与其对角线项相当,OBS算法采用对角线的值作为近似值代替,简化了计算过程和计算量. 同时注意到,在该算法中,基于幅度的方法假设Hessian矩阵各项同性;OBD算法假设H为对角矩阵;FARM算法[8]假设线性网络表达,并且只更新最后两层之间的权重. 事实表明,这些假设都不是有效的,也不足以达到最佳的剪枝效果.

虽然隐藏单元的数量在很大程度上决定了网络准确估计后验概率的能力,但是神经网络中的连接数随着隐藏层级和单元数量的增加而迅速增加,并且该数量受到实际可用计算资源的约束. 参数数量的确是网络性能重要因素,可是了解它们如何投入使用,获取网络中可用的训练参数和连接更为重要.

为了评估完全连接层结构的效率,本节研究了其连接权重的幅值的统计特性. 该想法基于权重是训练完成的网络中连接的显着性指标. 图1为完成训练后GoogLeNet模型卷积层中连接权重的直方图,细线为正态分布,平均值为−0.01,标准偏差为0.056.

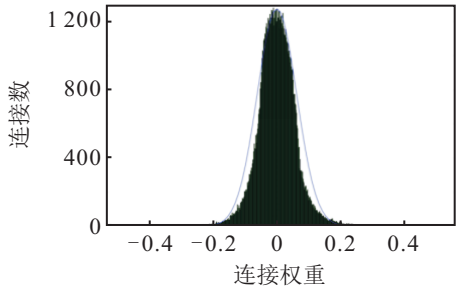

图1 GoogLeNet模型Conv层中连接权重直方图

由图1可见,大部分权重接近于零,可以对经过训练后的神经网络进行稀疏化仿真实验. 如果小幅值权重可删减的假设是正确的,则由图1可以看出,许多连接可能会被修剪,这将大大减少训练网络过程的计算量. 由于修剪减少了自由参数的数量,可以提高网络的泛化能力. 文献[9]仅使用反向传播计算中

收集的信息(权重和梯度)来度量剪枝进程. 本文实验将采用类似策略,由两个参数控制剪枝进程,当权重w符合下式时,即可剪去:

$$\begin{cases} |w_i| < \alpha \max\limits_i(w_i), \\ \left|w_i \dfrac{\partial E}{\partial w_i}\right| < \beta. \end{cases} \tag{3}$$

其中:α的敏感值低于0.10;β取决于网络和训练数据,但可以调整以获得所需的修剪准确性. 实验表明,第2个约束比第1个约束弱得多. 通过考虑权重的绝对值,实现相同数量的修剪得到几乎相同的网络性能. 因此,本文仅使用式(3)评估中的第1个标准,或者等效的,令$\beta = \infty$. 如果$|w| < \alpha$,则权重被简单地去除. 在这种情况下,不断调整以获得最优的修剪量.

用于权重更新的梯度下降法会产生许多小的权重. 假设对于小权重,如果单次更新后新参数是具有相同分布的独立随机变量,则所得权重是正态分布. 依据图1的权重分布可见,其实际分布可以通过正态分布很好地近似,均值接近于零,但对于幅值约0.1的权重,拟合度并不好. 由此得出结论,至少不是所有的0.1左右权重都是训练过程的人为因素产生的,这对参数α的选择是极具参考价值的.

修剪后,网络被重新训练,可以找到新的、更具相关性的最优网络权重,并且再训练的收敛速度比原来的训练快得多. 这是因为网络规模较小,通常几次迭代即可满足. 虽然重新训练的主要作用是纠正被删除连接中所产生的误差,但可能对离开误差函数的局部最小值也有影响.

文献[6]中OBD的一个遗憾是,在再训练前,将OBD方法与简单的基于幅度的方法进行比较较为优越,但在再训练后没有比较. 这并不能令人信服,因为再训练的影响远大于修剪后直接对比两种方法之间的差异. 第2节对剪枝前后网络再训练,实验结果表明,尽管采用简单的修剪标准,再训练后的网络性能可以媲美甚至比原始网络更好.

1.2 剪枝流程

剪枝方法需要进行三步操作: 首先,通过正常网络训练开始更新权重,与传统训练不同,本文并不关心迭代后权重的最终值,而是聚焦于训练前后层级单元之间关系是否重要;然后,修剪低权重连接,从网络中删除权重低于阈值的所有连接,将密集网络转换为稀疏网络;最后,重新训练网络,以获取剩余稀疏连接的最终权重,这一步十分关键,如果修剪后的网络在没有再训练的情况下直接进行使用,则其精度会受到极大影响.

1.2.1 Blob结构的修改

Caffe使用Blob结构来存储、交换和处理网络中正向和反向迭代的数据和导数信息,Blob是Caffe的标准数组结构,提供了Caffe的统一内存接口.

一般对于批量图形数据,Blob的维度为像数量$N\times$通道数$K\times$图像高度$H\times$图像宽度W. Blob按行为主存储,四维Blob数组中,坐标为(n,k,h,w)的值的物理位置为$((n\times K+k)\times H+h)\times W+w$. 若该层为全连接层,则Blob为二维数组,维度分别为输入数量和输出数量.

在Caffe中,Blob包括data_、diff_、count_、capacity_、shape_和shape_data_6个属性. 将原始矩阵由密集存储转化为稀疏形式,需要添加新的属性来存储稀疏化后的矩阵参数. 本文采用CSR (Compressed sparse row)稀疏矩阵的存储方式,CSR格式使用3个一维数组A、IA、JA,按行为主存储$m\times n$的稀疏矩阵M. 设nnz表示M中非零项的数量,数组A的长度为nnz,按从左到右从上到下的顺序保存M的所有非零项. 数组IA的长度为$m+1$,IA的前m个元素存储M每一行中第1个非零元素A的偏移索引值,最后一个元素$IA[m]$存储nnz. 数组JA的长度为nnz,存储A每个元素在M中的列索引. 这种格式允许进行快速的行访问和矩阵向量乘法. 因此,Blob的数据结构中添加了3个向量val_、rowptr_、colind_,这3个变量分别存储着所有非零元素值、非零元素行指针、非零元素列索引. 此外,还需要添加3个变量nnz_、sparse_、mask_,nnz_用来存储非零元素的个数,sparse_用来判断是否需要进行稀疏存储,mask_用来标记该元素是否需要进行梯度更新.

1.2.2 正则化与弃置层

选择正确的正则化方式会影响剪枝和再训练的表现. L_1正则化惩罚非零参数,导致更多的参数接近于零,因此在修剪后再训练之前可以提供更好的准确性. 然而,训练留下的连接不如L_2正则化,再训练后精度有所降低.

弃置层广泛用于防止过度拟合,也适用于再训练,但在再训练期间必须调整弃置率,以应对网络架构的变化. Hinton构建的弃置算法[10]可以被认为是一个“软弃置”,因为每个参数都以概率丢弃,而不是绝对地丢弃. 修剪可以被认为是一个“硬弃置”,如果参数在修剪后被置零,则没有机会回来. 随着矩阵稀疏化,分类器将选择最多的信息预测因子,因此具有更少的预测方差,减少了过度拟合. 由于剪枝已经提高了模型的泛化能力,再训练时弃置率应该更小.

令C_i为第i层中的连接数,C_{io}为原始网络,C_{ir}为再训练后网络,N_i为第i层中神经元的数量,有

$$C_i = N_i N_{i-1}, \tag{4}$$

$$D_r = D_o\sqrt{\frac{C_{io}}{C_{ir}}}. \tag{5}$$

其中:D_o为原始弃置率,D_r为再训练时的弃置率. 由于弃置层的作用,且C_i与N_i具有平方关系,根据式(4),修剪参数后的弃置率应遵循式(5).

1.2.3 局部剪枝和参数协同适应

在再训练期间,应将训练阶段得到的权重作为剪枝后剩余连接的权重,而不是将剪枝网络的参数重新初始化. 文献[11]表明,卷积神经网络拥有脆弱的协同适应特征:网络模型在原始训练权重下,进行梯度下降能够继续优化,恢复精度,但是重新初始化一些层级参数后,进行重新训练则没有达到预期效果. 所以如果重新初始化剪枝网络,则必须重新整理;如果在剪枝网络上再训练,则需要保留剩余连接的原始参数.

从保留权重开始重新训练已修剪的层需要较少的计算,因此不需要通过整个网络传播. 此外,随着网络越来越深,神经网络容易遭受消失的梯度问题[12],使得深层网络中的修剪错误难以恢复. 为了防止这种情况发生,修复了网络的一部分参数,只通过重新使用幸存参数来重新训练浅网络,这些参数在初始训练期间已经与未修剪的层共同适配.

1.2.4 修剪神经元

修剪连接后,可以安全地修剪具有零输入连接或零输出连接的神经元. 通过去除或从修剪神经元的所有连接来进一步修剪. 再训练阶段,到达具有零输入连接和零输出连接的死神经元时,系统会执行自动赋值,这是由于梯度下降和正则化,具有零输入连接(或零输出连接)的神经元将不会对最终损耗产生贡献,导致其输出连接(或输入连接)的梯度为零. 只有正则化才能将权重置为零. 因此,在再训练期间,死神经元将被自动去除.

1.3 卷积神经网络的定量分析

在典型的卷积神经网络中,假定第k层卷积层,其输入特征图尺寸为$D_M \times D_M \times M$,M为输入图像通道数,经卷积操作后得到的输出特征图尺寸为$D_N \times D_N \times N$,N为输出通道数,则可知该卷积层由N个卷积核组成. 假设每个卷积核的尺寸为$D_k \times D_k \times M$,则有

$$D_N = \frac{D_M - D_k + 2P}{S} + 1. \tag{6}$$

其中:P为边缘填充的宽度,S为卷积核的滑动步长. 本文GoogLeNet网络模型中的卷积核长宽均相等,且不带偏置项,第k层卷积层的参数数量p_k与计算开销c_k分别为

$$p_k = M \times N \times D_k \times D_k, \tag{7}$$

$$c_k = M \times N \times D_k^2 \times D_N^2 = p_k \times D_N^2. \tag{8}$$

由式(7)和(8)可见,通道数与参数数量、计算开销存在正相关关系,表明降低通道数能够减少网络模型的参数规模. 对比网络修剪前后的参数数量和计算开销,作为剪枝算法有效程度的量化参考.

2 仿真实验

2.1 GoogLetNet训练数据分析

本文采用ILSVRC2012数据集,由于时间和硬件性能等原因,使用前100个类,每类训练集有1 300张图片,格式为JPEG. 这100个类一一对应于WorldNet的100个同义子集,100个同义子集互相不重复,也不是父类.

模型训练250个周期,最小批次为64,GoogLeNet开始学习速率α为0.01,学习速率变化因子γ为0.96,学习策略为每个步长迭代后,将α乘以γ. 每16 k次迭代时,降低学习速率,最大迭代次数为500 k,动量μ为0.9,权重衰减率为0.000 2.

每经过16 k次迭代,学习率α便会降低,防止精度振荡,加快收敛过程. 在第1个16 k次迭代,Top-1和Top-5的精度上升最快,此时神经网络之间的连接关系已经形成,之后的迭代训练是为了不断地强化相关性和消除噪声干扰,因此选择第16 k训练得到的模型进行剪枝.

为了帮助理解每个结构单元对整个GoogeLeNet模型性能的影响,将给定的12个结构分别进行剪枝处理,并分别冻结它们;检测不同阈值下的相应性能指标,并进行迭代. 仅经过少数次的迭代,其指标即可回升至剪枝前的数值. 有人会认为,剪枝算法仅去除了神经网络本身对噪声拟合的相关连接,从而达到了良好泛化效果. 卷积层剪枝后的网络模型性能对比如图2所示. 由图2可知,同时对所有或者大部分架构进行剪枝操作,会影响整个网络架构的整体性能. 正因为如此,如果输入中提取的特征被严重破坏,则即使深度足够的神经网络也不能很好地映射图像特征. 因此,不同结构单元之间存在的依赖程度越大,临界阈值越小;结构单元越接近输出层,临界阈值越大.

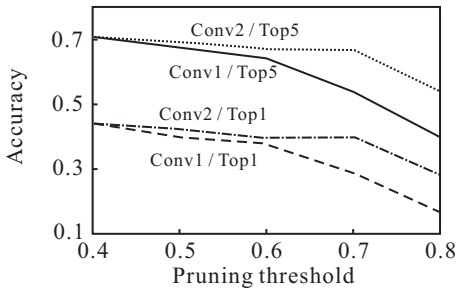

图 2　卷积层剪枝后的网络模型性能对比

2.2　Inception模块的剪枝

依照上述方法流程,分别对9个Inception模块(见图3)进行一系列实验,研究不同阈值下剪枝算法对最终性能的影响. 每个Inception模块包含4个并行通道,每个通道都有1×1的卷积核,用来降维以减少计算量,每个卷积核后面都有ReLU激活层,用来进行线性整流.

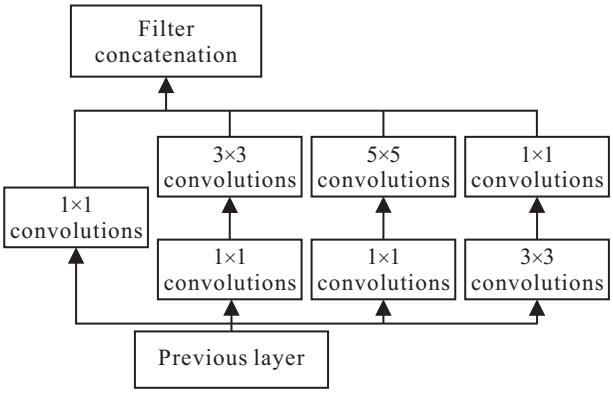

图 3　Inception结构

每个具体的Inception模块具有不同数量的输出尺寸,包括$28\times28\times256(3a)$, $28\times28\times480(3b)$, $14\times14\times512(4a,b,c)$, $14\times14\times528(4d)$, $14\times14\times832(4e)$, $7\times7\times832(5a)$, $7\times7\times1\,024(5b)$.

每经过16 k次迭代,即一个周期后,为神经网络训练样本生成一个相应的caffemodel,对其进行剪枝并继续训练. 实验结果表明,剪枝网络的剩余连接可以很好地模拟原始神经网络,保持其相应的性能. 同时,本文只对原始网络和剪枝网络进行训练,对剪枝网络进行再剪枝操作是多此一举. 因为进行再次剪枝操作,极容易超过临界修剪阈值,严重破坏提取到的特征. 同时,这对模型存储、收敛速度以及计算量等指标的改善均无足轻重. 所以,本文所有性能指标对比仅且仅需与原始网络对比.

仿真结果表明,任何一个Inception模块中的任意一条降维通道都可以完全剪去,但最好保留至少2个并行通道进行训练,否则反而会降低收敛速度和整体网络性能. 表1为各Inception模块剪枝后的网络模型性能对比(修剪阈值为0.9). 由表1可见,在靠近输入层的Inception中,剪去高卷积核通道效果较好;在靠近输出层的Inception中,剪去低卷积核通道效果较好. 因为低层模块主要提取图像的低维特征,而较为抽象的特征更容易在高层捕获.

表1　各Inception模块剪枝后的网络模型性能对比

	loss	Top-1	Top-5
Inception(3a)	2.491 04	0.379 0	0.649 6
Inception(3b)	2.497 64	0.370 4	0.648 8
Inception(4a)	2.376 49	0.397 4	0.673 0
Inception(4b)	2.244 78	0.424 6	0.700 6
Inception(4c)	2.284 73	0.422 4	0.691 0
Inception(4d)	2.268 65	0.432 2	0.702 2
Inception(4e)	2.251 23	0.432 2	0.704 8
Inception(5a)	2.235 94	0.431 8	0.710 8
Inception(5b)	2.213 21	0.442 2	0.721 8

仅对Inception模块进行剪枝前后,网络性能对比状况与仅修剪卷积层的情形类似. 过小的阈值没有达到简化网络的要求;过大的阈值会损坏网络架构,经过再训练也不能恢复到原来的性能. 设定合适的阈值,网络在进行剪枝处理后,准确率会有所降低,但经过少数次迭代,网络的准确率与原始模型不相上下,此时已经减少了参数量和计算量,达到了网络剪枝的目的.

2.3　网络剪枝

根据上述特性,对GoogLeNet模型不同结构作了不同程度的剪枝处理,前两个卷积层仅修剪60 %的参数,尽可能保留原始网络提取低维特征的功能. Inception模块剪去1或2个降维通道,其余并行通道剪枝阈值为70 %～90 %. 作为输出的3个全连接层模块几乎占据全部参数的一半,由于loss1和loss2对最终结果仅有部分参考价值,在训练前就已经剪去;剩余的loss3也仅保留5 %～10 %的参数. 因此,最后剪枝GoogLeNet模型中大约仅有65万参数,压缩了16倍,主要集中在卷积层.

图4为剪枝前后网络性能比较. 由图4可见,在修剪前后经过400次迭代后,剪枝网络的准确率已达到同时期的原始网络. 在上述剪枝策略下,模型性能指标没有明显下降,甚至比原始模型稍好,收敛速度更快. 这支持了剪枝算法可以抑制过度拟合、增强泛化性的观点.

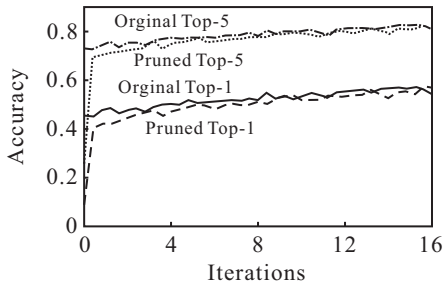

图 4 剪枝前后网络性能比较

3 结 论

本文针对网络参数量和计算量冗余问题,通过剪枝算法简化网络结构,减少参数数量,获得近似的网络模型,以达到压缩效果. 由于卷积层和完全连接层中强相关连接很大程度上决定了网络准确估计后验概率的能力,对原始网络模型进行训练、剪枝、再训练等一系列操作来减少计算量和存储,保留了强相关连接,不影响网络的其他性能. 从理论和实验效果上证明了该方法的可行性和有效性. 研究结果表明,对GoogeLeNet模型不同结构进行不同程度的剪枝,经过少量迭代后,能够保留原始性能,大大减少网络计算量. 此外,应该分析过度剪枝的网络,以理解在深度神经网络中各个层级之间连接协同适应和相互依赖的关系.

参考文献(References)

[1] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]. IEEE Computer Vision and Pattern Recognition. Boston: IEEE, 2015: 1-9.

[2] Nowlan S J, Hinton G E. Simplifying neural networks by soft weight-sharing[M]. Boston: MIT Press, 1992: 473-493.

[3] Arora S, Bhaskara A, Ge R, et al. Provable bounds for learning some deep representations[C]. Proc of the 31th Int Conf on Machine Learning. Beijing: JMLR, 2013: 584-592.

[4] Song F, Dongarra J. Scaling up matrix computations on shared-memory manycore systems with 1000 CPU cores[M]. New York: ACM, 2014: 333-342.

[5] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]. Int Conf on Neural Information Processing Systems. Nice: Curran Associates Inc, 2012: 1097-1105.

[6] Cun Y L, Denker J S, Solla S A. Optimal brain damage[M]. San Francisco: Morgan Kaufmann Publishers Inc, 1990: 598-605.

[7] Hassibi B, Stork D G. Second order derivatives for network pruning: Optimal brain surgeon[J]. Advances in Neural Information Processing Systems, 1993, 5(1): 164-171.

[8] Kung S Y, Hu Y H. A frobenius approximation reduction method (FARM) for determining optimal number of hidden units[C]. IJCNN-91-Seattle Int Joint Conf on Neural Networks. Seattle: IEEE, 1991: 163-168.

[9] Strm N. Phoneme probability estimation with dynamic sparsely connected artificial neural networks[J]. The Free Speech J, 1997, 1(5): 1-41.

[10] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting[J]. J of Machine Learning Research, 2014, 15(1): 1929-1958.

[11] Yosinski J, Clune J, Bengio Y, et al. How transferable are features in deep neural networks?[C]. Int Conf on Neural Information Processing Systems. Boston: MIT Press, 2014: 3320-3328.

[12] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult[J]. IEEE Trans on Neural Networks, 2002, 5(2): 157-166.

作者简介

彭冬亮(1977–), 男, 教授, 博士, 从事多传感器信息融合技术等研究, E-mail: dlpeng@hdu.edu.cn;

王天兴(1993–), 男, 硕士生, 从事深度学习的研究, E-mail: hdu_wtx@163.com.

(责任编辑: 郑晓蕾)