第47卷 第2期 计 算 机 工 程 2021年2月
Vol.47 No.2 Computer Engineering February 2021

·人工智能与模式识别· 文章编号：1000-3428(2021)02-0111-07 文献标志码：A 中图分类号：TP183

基于参数子空间和缩放因子的YOLO剪枝算法

杨民杰，梁亚玲，杜明辉

（华南理工大学 电子与信息学院，广州 510641）

摘 要：为保证YOLO网络在嵌入式设备上正常运行，需采用剪枝算法精简滤波器以减小网络存储空间和计算量，而现有剪枝算法耗时较长且剪枝精度较低。提出一种基于参数子空间和批量归一化(BN)层缩放因子的双准则剪枝算法。将卷积层滤波器通过k均值聚类得到不同参数子空间，在子空间内使滤波器按权重排序并去除权重较低的滤波器，同时采用BN层缩放因子剪枝算法避免剪枝精度下降。实验结果表明，采用该算法剪枝后的YOLOv3网络在精度不变的情况下，占用的内存减少5/6且计算时间缩短1/3，与PF、CP等剪枝算法相比，该算法在保持较高网络精度的情况下计算量更少。

关键词：模型压缩；剪枝；目标检测；均值聚类；缩放因子

开放科学(资源服务)标志码(OSID)：

中文引用格式：杨民杰，梁亚玲，杜明辉.基于参数子空间和缩放因子的YOLO剪枝算法[J].计算机工程，2021，47(2)：111-117.

英文引用格式：YANG Minjie，LIANG Yaling，DU Minghui. YOLO pruning algorithm based on parameter subspace and scaling factor[J].Computer Engineering，2021，47(2)：111-117.

YOLO Pruning Algorithm Based on Parameter Subspace and Scaling Factor

YANG Minjie，LIANG Yaling，DU Minghui

（School of Electronic and Information Engineering，South China University of Technology，Guangzhou 510641，China）

【Abstract】In order to ensure the normal operation of YOLO network on embedded devices，it is necessary to use pruning algorithm to simplify the filter to reduce the network storage space and the amount of calculation. However，the existing pruning algorithms are time-consuming and have low pruning precision. This paper proposes a bi-criteria pruning algorithm based on parameter subspace and Batch Normalization（BN）layer scaling factor. The convolution layer filter is clustered into different parameter subspaces by k-means clustering. In the subspace，the filters are sorted by weight，and the filters with lower weight are removed. The BN layer scaling factor pruning algorithm is used to avoid the degradation of pruning precision. Experimental results show that the memory occupied by the network which is pruned by the proposed algorithm is reduced by 5/6 while the precision remains unchanged and the computing time is reduced by 1/3. Compared with PF，CP and other pruning algorithms，the proposed algorithm requires less computation while maintaining high network precision.

【Key words】model compression；pruning；object detection；mean clustering；scaling factor

DOI：10. 19678/j. issn. 1000-3428. 0056932

0 概述

自2012年AlexNet[1]获得ImageNet大型视觉识别比赛（ImageNet Large Scale Visual Recognition Competition，ILSVRC）冠军以来，以AlexNet为代表的基于深度学习的卷积神经网络受到广泛关注并得到深入发展，一系列深度学习神经网络相继出现，例如用于分类任务的Vgg[2]、GoogleNet[3]和ResNet[4]等网络以及用于目标检测任务的Rcnn[5]、FastRcnn[6]、FasterRcnn[7]、YOLOv3[8]和SSD[9]等网络。随着神经网络逐渐加深，其应用于视觉任务的网络精度已逐渐接近甚至超过人类的视觉识别水平[10]，而网络模型占据的存储空间、计算时长和计算量等均不断增加，对嵌入式设备的计算性能提出更高要求，所需设备成本与电量能耗均增大。然而嵌入式设备在计算能力、存储能力以及电量能耗等方面均有一定限制[10]，为了

基金项目：国家自然科学基金(61701181)；广东省自然科学基金(2017A030325430)；广州市科技计划项目(201707010070)。

作者简介：杨民杰(1996—)，男，硕士研究生，主研方向为目标检测及模型优化；梁亚玲(通信作者)，副教授、博士；杜明辉，教授、博士。

收稿日期：2019-12-17 **修回日期：**2020-01-19 **E-mail：**ylliang@scut.edu.com

在嵌入式设备中成功部署相关的深度学习神经网络，需在保持神经网络分类或检测精度的同时对其进行压缩。

神经网络压缩算法主要包括低秩近似算法、剪枝算法、量化算法、知识蒸馏算法和紧凑型网络设计算法等。低秩近似算法是将稠密的满秩矩阵表示为若干低秩矩阵的组合，低秩矩阵又分解为小规模矩阵的乘积，从而达到简化的目的。文献[11]提出一种线性组合卷积核基底算法，用$f\times1+1\times f$卷积核替代$f\times f$卷积核进行低秩近似。剪枝算法是通过修剪神经网络中冗余滤波器进行网络优化，删除神经网络权重矩阵中不重要的部分权重，仅保留有用部分，再重新对网络进行微调。量化算法是用低精度参数权值代替神经网络中32 bit的浮点型参数权值，目前大多数低精度的方案采用INT8型参数代替FP32型参数，通过牺牲部分精度降低占用空间。知识蒸馏算法是指通过迁移学习将预先训练好的复杂网络的输出作为监督信号去训练简单的网络，以达到压缩网络的目的。紧凑型网络设计算法是重新构建可达到原精度的小型网络，例如MobileNet和ShuffleNet网络。

由于剪枝算法无需考虑应用领域、网络架构和部署平台，符合神经网络应用于嵌入式平台的实际需求，因此本文采用剪枝算法压缩网络。现有关于剪枝算法的研究大部分基于图像分类网络，但在人工智能应用场景中，目标检测网络的应用领域更广泛，在该网络上进行剪枝更困难。YOLO网络是一种端到端的目标检测网络，属于One-stage网络，One-stage网络包括特征提取和训练分类器2个阶段，而Two-stage网络目标检测方法包括区域选择、特征提取和训练分类器3个阶段，该网络进行区域选择后产生大量冗余的候选区域，会耗费较多推理时间。与Two-stage网络相比，One-stage网络耗时更短，因此YOLO网络推理速度更快。在嵌入式设备端，推理速度是评价实用性的重要参考指标，YOLO网络中的YOLOv3网络与同样作为One-stage的SSD网络准确率相同，但是其运算速度比SSD网络快3倍[8]，因此，本文选择YOLOv3网络来设计剪枝算法。

剪枝算法分为结构化剪枝算法和非结构化剪枝算法。结构化剪枝算法是通过从深度神经网络中剪去整个滤波器对网络进行简化来减少推理时间。非结构化剪枝是单独对每一层参数剪枝，这会造成不规则内存访问情况，从而降低推理效率。文献[12]利用二阶泰勒公式展开并选择参数进行剪枝，将剪枝看做正则项来改善训练和泛化能力。文献[13]根据神经元连接权值大小修剪训练后网络中不重要的连接，以减少网络参数。上述研究均采用非结构化剪枝算法基于单个权重进行剪枝，剪枝后的网络需配置专门的软件或硬件来加速运行。为使剪枝后的网络能在普适平台上运行，文献[14]引入结构化稀疏性，每个卷积单元根据其对网络在验证数据集上准确率的影响程度分配分值，通过去除分值低的卷积单元进行剪枝，然而该方法耗时较长，仅适用于小模型。文献[15]通过移除网络中权重值在0附近的滤波器及其连接特征图来降低计算成本，且无需稀疏卷积库支持，但当网络的权重值的分布非常集中或者权重值大部分不在0附近时，该方法剪枝效果较差。

文献[16]利用批量归一化(Batch Normalization，BN)层缩放因子γ在训练过程中衡量通道的重要性，去除不重要的通道压缩模型提升计算速度。在此基础上，本文提出一种利用参数子空间和缩放因子的YOLO剪枝算法，使用参数子空间避免卷积层滤波器权重剪枝范数标准差过大，采用双准则剪枝策略分别利用卷积层和BN层对YOLOv3网络进行剪枝。

1 双准则剪枝算法

剪枝算法的核心思想是寻找一种合适的评价指标去除冗余滤波器。现有剪枝算法大部分基于滤波器范数的大小，如果滤波器所产生对应特征图的L2范数接近0，则表明滤波器范数较小，该特征图对网络贡献较少，即该滤波器对网络的重要性较低。根据上述原理可对网络中滤波器按照重要性排序，并删除重要性较低的滤波器。该做法的前提是滤波器符合两个理想条件：1)滤波器范数标准差较大；2)滤波器最小范数接近0。但是大部分滤波器不符合上述条件，特别是目标检测网络滤波器。为此，本文提出基于参数子空间和BN层缩放因子双准则的剪枝算法，其具体流程如图1所示。

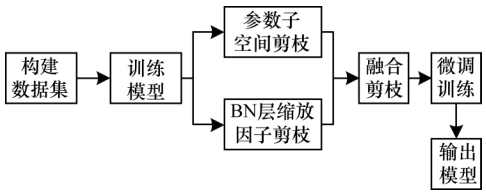

图1 本文剪枝算法流程

Fig.1 Procedure of the proposed pruning algorithm

1.1 基于参数子空间的剪枝算法

参数子空间是采用k均值聚类算法将卷积层滤波器聚类得到的不同子空间，在其中进行剪枝可避免滤波器范数标准差较大以及滤波器最小范数接近0。k均值聚类算法是一种迭代求解的聚类分析算法，也是基于样本集合划分的聚类算法。该算法将样本集合划分为k个子集并构成k个类，将n个样本分到k个类中，每个样本与其所属类中心的距离最小

且仅属于1个类。k均值聚类算法的复杂度为$O(mnk)$，m为样本维数。k均值聚类算法的步骤为：1）随机选取k个样本作为k个类的中心，将样本逐个指派到与其最近中心的类中，得到1个聚类结果；2）更新每个类的样本均值作为新的类中心；3）重复步骤1和步骤2，直到收敛或符合停止条件（没有（或最少）样本被重新分配给不同的聚类、没有（或最少）聚类中心发生变化、误差平方和局部最小）为止。

k均值聚类算法以n个样本集合X为输入，以样本集合的聚类C^*为输出，具体过程如下：

1）初始化。令$t=0$，随机选择k个样本点作为初始聚类中心$m^{(0)}=(m_1^{(0)}, m_2^{(0)}, \cdots, m_l^{(0)}, m_{l+1}^{(0)}, \cdots, m_k^{(0)})$。

2）对样本进行聚类。对固定的类中心$m^{(t)}=(m_1^{(t)}, m_2^{(t)}, \cdots, m_l^{(t)}, m_{l+1}^{(t)}, \cdots, m_k^{(t)})$，其中$m_l^{(t)}$为$G_l$的中心，计算每个样本到类中心的距离并将其指派到预期最近的中心的类上，构成聚类结果$C^{(t)}$。

3）计算新的类中心。对聚类结果$C^{(t)}$，计算各类中样本均值作为新的类中心$m^{(t+1)}=(m_1^{(t+1)}, m_2^{(t+1)}, \cdots, m_l^{(t+1)}, m_{l+1}^{(t+1)}, \cdots, m_k^{(t+1)})$。

4）如果迭代收敛或者符合停止条件，则输出$C^*=C^{(t)}$；否则令$t=t+1$，返回步骤2。

在剪枝过程中，本文使用的YOLOv3网络有较多连续的残差结构，如果对残差结构的每一层进行剪枝，则会造成部分网络层的通道数不同，无法通过捷径（shortcut）进行相加运算，导致网络不能正常运行。因此，本文避开残差结构中相连的卷积层进行剪枝，但是这些卷积层参数量仍会随着1×1卷积（conv）层通道数的减少而降低，如图2和图3所示。图2为未剪枝的残差结构，其中第一层参数量为128×256×1×1+128=32 896，第二层参数量为256×128×3×3+256=295 168。图3为已剪枝的残差结构，其中第一层参数量为37×256×1×1+37=9 509，第二层参数量为256×37×3×3+256=85 504。可以看出，在避开相连卷积层进行剪枝后，未剪枝的卷积层参数量会出现明显下降，从而简化算法并减少计算量。

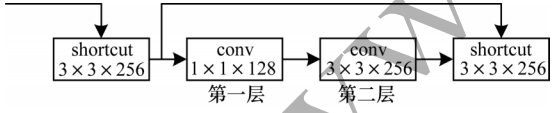

图2　未剪枝的残差结构

Fig.2　Residual structure without pruning

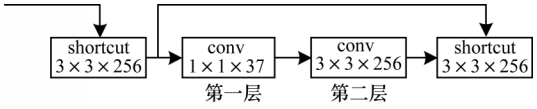

图3　剪枝后的残差结构

Fig.3　Residual structure after pruning

选择非残差结构需要相加的卷积层，对每个卷积层的滤波器进行k均值聚类分析，使用肘部法则确定每个卷积层的k值。k均值聚类是以最小化样本与质点平方距离误差（loss）作为目标函数，若将各簇质点与簇内样本点的平方距离误差和称为畸变程度，则一个簇的畸变程度越低表明簇内结构越紧密，一个簇的畸变程度越高表明簇内结构越松散。畸变程度通常随聚类类别的增加而减小，但对于有一定区分度的数据样本，在k值达到某个临界点时畸变程度将急剧减小，此时聚类性能较好。基于畸变程度的变化，YOLOv3不同卷积层可训练不同k均值聚类模型，从而得到每个卷积层最适合的聚类类别数。图4和图5分别表示第63层和第103层卷积层在不同k值下各簇质点与簇内样本点的平方距离误差。

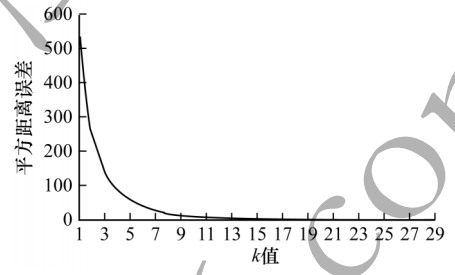

图4　第63层卷积层的聚类loss曲线

Fig.4　Cluster loss curve of the 63rd convolution layer

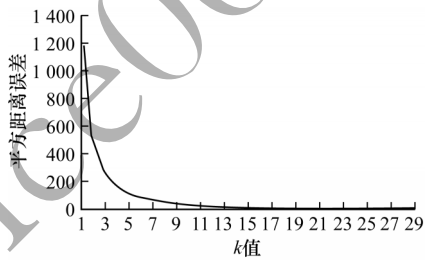

图5　第103层卷积层的聚类loss曲线

Fig.5　Cluster loss curve of the 103rd convolution layer

由于本文在聚类分析前先将卷积层每个滤波器的权重进行求和，再对求和后的权重进行聚类分析，因此聚类分析针对单层网络进行，在单层网络上使用肘部法则确定k值。使用该方法对每个卷积层用不同k值进行聚类分析得到各卷积层聚类结果。计算卷积层每个聚类类别中滤波器的权重，将其按照由大到小排序，并根据一定比例去除权重较小的滤波器。本文使用基于参数子空间的方法对单层卷积层进行剪枝实验，剪枝率设置为50%，采用平均精准度均值（Mean Average Precision，MAP）作为剪枝效果的评价指标。图6为第63层卷积层在不同k值下MAP的变化，可以看出，当k为4时剪枝所得MAP值最大，k=4即为图4中聚类loss曲线的畸变临界点。图7为第103层卷积层在不同k值下MAP的变化，可以看出，当k为4时剪枝所得MAP值最大，k为4即为

图5中聚类loss曲线的畸变临界点。由此可证明,使用肘部法则选出的k值有效。先用k均值聚类再根据权重排序进行剪枝,可避免滤波器范数标准差较大以及滤波器最小范数接近0。

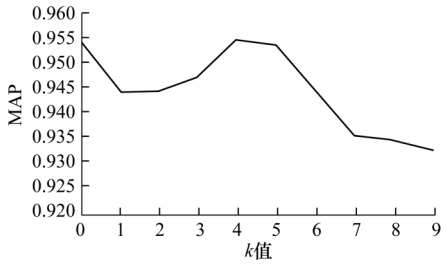

图6 第63层卷积层在不同k值下的MAP变化曲线

Fig.6 MAP variation curve of 63rd convolution layer with different k values

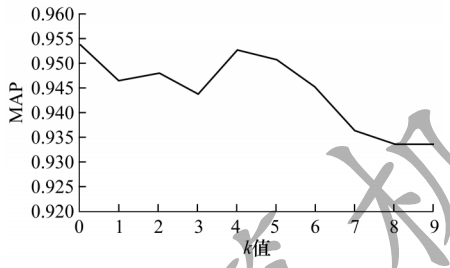

图7 第103层卷积层在不同k值下的MAP变化曲线

Fig.7 MAP variation curve of 103rd convolution layer with different k values

1.2 基于BN层缩放因子的剪枝准则

由于BN层可有效防止梯度爆炸并加速网络收敛,因此其被应用于各种卷积层神经网络中,位于卷积层之后,以对卷积层后的特征图完成归一化操作。然而若BN层仅在卷积层后进行归一化,再直接将数据送入下一层进行卷积计算,则网络将无法学习输出的特征分布。由于BN层后有ReLu激活层,若BN层后的特征图数据大部分小于0,那么其经过ReLu激活层后将失去大部分特征值,因此,BN层需通过优化缩放系数γ和偏移系数β对数据进行归一化处理,使网络能学习到输出的特征分布。

算法1 BN层缩放因子剪枝算法

输入 Values of x over a mini-batch: $B=\{x_{1,2,\cdots,m}\}$; Parameters to be learned: γ,β

输出 $\{y_i=\mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$1.\mu_{\mathrm{B}}\leftarrow\frac{1}{\mathrm{m}}\sum_{\mathrm{i}=1}^{\mathrm{m}}x_{\mathrm{i}}\text{//mini-batch mean}$$

$$2.\sigma_{\mathrm{B}}\leftarrow\frac{1}{\mathrm{m}}\sum_{\mathrm{i}=1}^{\mathrm{m}}(x_{\mathrm{i}}-\mu_{\mathrm{B}})^2\text{//mini-batch variance}$$

$$3.\widehat{\mathrm{x}_{\mathrm{i}}}\leftarrow\frac{\mathrm{x}_{\mathrm{i}}-\mu_{\mathrm{B}}}{\sqrt{\sigma_{\mathrm{B}}^2+\xi}}\text{//normalize}$$

$$4.\mathrm{y}_{\mathrm{i}}\leftarrow\gamma\widehat{\mathrm{x}_{\mathrm{i}}}+\beta\equiv\mathrm{BN}_{\gamma,\beta}(\mathrm{x}_{\mathrm{i}})\text{//scale and shift}$$

在算法1中,μ_B和σ_B^2分别为输入的均值和方差。卷积层每个滤波器均会产生1个特征图,每个特征图BN层在归一化时有唯一的缩放系数γ与其对应,通过缩放值可选择冗余的特征图,再由特征图选出冗余的滤波器,从而根据缩放系数大小来判断滤波器的重要性。本文利用缩放系数γ从BN层角度更全面地判断滤波器的冗余程度。

1.3 双准则融合剪枝

双准则融合剪枝是结合参数子空间和BN层缩放因子两种剪枝算法去除冗余的滤波器以实现最精简的网络结构。参数子空间剪枝算法是从卷积层角度寻找网络中冗余的滤波器,BN层缩放因子剪枝算法是从BN层角度寻找不重要的滤波器,本文将原始YOLOv3网络结构通过上述两种算法进行剪枝,以最大化压缩网络。

算法2 双准则融合剪枝算法

双准则融合剪枝算法将训练数据、原始YOLOv3网络结构和初始化权重以及剪枝率P为输入,以剪枝后网络为输出,具体过程如下:

1)原始YOLOv3网络在训练数据上进行训练。

2)选取可剪枝的网络卷积层。

3)对选取的卷积层运用肘部法则,得到每个卷积层剪枝需要的k值。

4)对选取的卷积层进行聚类操作,得到参数子空间。

5)在参数子空间进行权重排序,根据剪枝率得到需要删除的滤波器。

6)在步骤2后,对每个卷积层的BN层缩放因子进行排序,根据剪枝率得到需去除的滤波器。

7)将步骤5和步骤6中得到的滤波器求并集。

8)在步骤2后去除步骤7中的滤波器总和,得到精简的网络结构。

9)在训练数据上对新的网络结构进行微调,得到最终需要的网络模型。

10)若剪枝结果不理想,则重新选择剪枝率并返回步骤5。

在上述过程中,剪枝率由实验获得,在保持网络精度下结合参数子空间和BN层缩放因子两种剪枝方法可最大限度地压缩网络。

2 实验与结果分析

将本文提出的基于参数子空间和BN层缩放因子的双准则剪枝算法(以下称为本文双准则剪枝算法)与其他剪枝算法在CIFAR10分类数据集上的网络精度和浮点计算量进行对比,并将参数子空间剪枝算法、BN层缩放因子剪枝算法和本文双准则剪枝

算法在YOLOv3网络中得到的实验结果进行对比，以分析本文方法的剪枝性能。

2.1　不同剪枝算法的对比

将不同剪枝算法在CIFAR10数据集上实验结果进行对比，如表1所示（“—”表示数据不详，60U40是双准则融合剪枝算法的剪枝率，其中60%的剪枝率属于BN层缩放因子剪枝算法，40%的剪枝率属于参数子空间剪枝算法）。可以看出，PF[17]算法的浮点计算量仅减少27.6%，网络精度却下降1.73%，这是因为该算法主要考虑滤波器权重的绝对值问题，默认了权重分布需遵循滤波器范数标准差较大以及滤波器最小范数接近0这两个条件，对剪枝网络和数据有一定的限制。CP[18]算法采用LASSO回归方式选择剪枝通道，但由于是逐层剪枝，因此剪枝后的网络精度下降较多。LCCT[19]算法由于在卷积结构中加入LCCL加速卷积层，增加了训练难度，因此浮点计算量无明显减少。SFP[20]算法采用动态剪枝方式，但由于其剪枝策略仍根据权重的范数并保留BN层的偏置系数，因此造成剪枝精度随机下降。AMFSF[10]算法引入注意力机制进行剪枝，并采用范数计算注意力的重要性，但其会忽略部分剪枝细节从而降低网络稳定性。与其他算法相比，本文双准则剪枝算法从权重和BN层两个角度进行剪枝，在保证剪枝精度基本不变的情况下，其在ResNet56网络中浮点计算量减少46.3%，在ResNet110网络中浮点计算量减少45.6%。与其他剪枝算法相比，本文双准则剪枝算法在分类过程中具有较好的剪枝效果。

表1　不同剪枝算法在CIFAR10数据集上实验结果的对比

Table 1　Comparison of experimental results of different pruning algorithms on CIFAR10 dataset

网络	算法	剪枝率/%	基本精度/%	剪枝网络精度/%	下降网络精度/%	剪枝网络浮点计算量	剪枝网络浮点计算量减少幅度/%
ResNet56	PF	—	93.04	91.31	1.73	9.09×10^7	27.6
ResNet56	CP	—	92.80	90.90	1.90	—	50.0
ResNet56	SFP	30	93.59	93.10	0.49	5.94×10^7	41.1
ResNet56	AMFSF	30	93.59	93.54	0.05	5.94×10^7	41.1
ResNet56	本文算法	60U40	93.59	93.55	0.04	5.42×10^7	46.3
ResNet110	PF	—	93.53	92.94	0.59	1.55×10^8	38.6
ResNet110	LCCT	—	93.63	93.44	0.19	—	34.2
ResNet110	SFP	30	93.68	93.38	0.30	1.50×10^8	40.8
ResNet110	AMFSF	30	93.68	94.13	−0.45	1.50×10^8	40.8
ResNet110	本文算法	60U40	93.68	93.60	0.08	1.38×10^8	45.6

2.2　YOLOv3剪枝效果对比

本文以Ubuntu18.04软件为实验平台，采用i9-9900K CPU和2080ti显卡，使用Deer数据集，其中训练集有4 367张图像，测试集有519张图像。将参数子空间剪枝算法、BN层缩放因子剪枝算法和本文双准则融合剪枝算法在YOLOv3网络中不同剪枝率下的剪枝效果进行对比，结果如表2～表4所示（70U40是双准则融合剪枝算法的剪枝率，其中70%的剪枝率属于BN层缩放因子剪枝算法，40%的剪枝率属于参数子空间剪枝算法）。可以看出：参数子空间剪枝算法可去除所选卷积层50%的滤波器，MAP值下降约0.029，计算时间从每张图像0.010 3 s下降到每张图像0.008 0 s，网络参数量下降约50%；BN层缩放因子剪枝算法可去除所选卷积层80%的滤波器，MAP值仅下降约0.007，计算时间下降到每张图像0.007 3 s，网络参数量下降约80%；本文双准则剪枝算法的MAP值下降约0.032，计算时间下降到每张图像0.006 6 s，网络参数量下降约85%。

表2　YOLOv3网络中参数子空间剪枝算法在不同剪枝率下的实验结果

Table 2　Experimental results of parameter subspace pruning algorithm in YOLOv3 network at different pruning rates

剪枝率/%	基本MAP	剪枝后MAP	下降MAP	基本计算时间/s	剪枝后计算时间/s	下降计算时间/s	基本参数量	剪枝后参数量	下降参数量
10		0.946 719	0.007 114		0.009 9	0.000 4		54 497 476	7 026 258
20		0.939 827	0.014 006		0.009 3	0.001 0		47 976 446	13 547 288
30	0.953 833	0.942 566	0.011 267	0.010 3	0.008 9	0.001 4	61 523 734	41 875 478	19 648 256
40		0.933 820	0.020 013		0.008 4	0.001 9		35 439 111	26 084 623
50		0.924 740	0.029 093		0.008 0	0.002 3		30 049 842	31 473 892

表3 YOLOv3网络中BN层缩放因子剪枝算法在不同剪枝率下的实验结果

Table 3 Experimental results of BN layer scaling factor pruning algorithm in YOLOv3 network at different pruning rates

剪枝率/%	基本MAP	剪枝后MAP	下降MAP	基本计算时间/s	剪枝后计算时间/s	下降计算时间/s	基本参数量	剪枝后参数量	下降参数量
10	0.953 833	0.954 298	−0.000 465	0.010 3	0.010 0	0.000 3	61 523 734	54 386 955	7 136 779
20		0.952 684	0.001 149		0.009 4	0.000 9		47 474 468	14 049 266
30		0.951 774	0.002 059		0.008 7	0.001 6		40 755 094	20 768 640
40		0.949 080	0.004 753		0.008 1	0.002 2		34 454 090	27 069 644
50		0.950 129	0.003 704		0.007 6	0.002 7		28 640 973	32 882 761
60		0.951 205	0.002 628		0.007 4	0.002 9		23 321 302	38 202 432
70		0.954 223	−0.000 390		0.007 4	0.002 9		17 720 609	43 803 125
80		0.947 328	0.006 505		0.007 3	0.003 0		12 831 220	48 692 514

表4 YOLOv3网络中本文双准则融合剪枝算法在不同剪枝率下的实验结果

Table 4 Experimental results of double criteria fusion pruning algorithm in YOLOv3 network at different pruning rates

剪枝率/%	基本MAP	剪枝后MAP	下降MAP	基本计算时间/s	剪枝后计算时间/s	下降计算时间/s	基本参数量	剪枝后参数量	下降参数量
70U40	0.953 833	0.930 827	0.023 006	0.010 3	0.007 3	0.003 0	61 523 734	16 590 764	44 932 970
70U50		0.923 583	0.030 250		0.007 1	0.003 2		15 375 294	46 148 440
80U40		0.927 648	0.026 185		0.006 8	0.003 5		11 579 405	49 944 329
80U50		0.922 038	0.031 795		0.006 6	0.003 7		9 105 432	52 418 302

3 结束语

为保证目标检测YOLOv3网络在嵌入式设备上正常运行，本文提出一种结合参数子空间和BN层缩放因子的双准则剪枝算法。采用参数子空间剪枝算法避免权重分布过于集中，使用BN层缩放因子剪枝算法去除不重要的滤波器，同时利用卷积层和BN层进行剪枝以最大化压缩网络。实验结果表明，与PF、CP等剪枝算法相比，该算法可保持较高网络精度且计算量更少。下一步将对网络量化进行研究，在保证网络精度的同时进一步压缩网络并提升计算速度。

参考文献

[1] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[C]//Proceedings of 2012 Conference and Workshop on Neural Information Processing Systems. Cambridge, USA: MIT Press, 2012: 1097-1105.

[2] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[EB/OL]. [2019-11-02]. https://arxiv.org/abs/1409.1556.

[3] SZEGEDY C, IOFFE S, VANHOUCKE V, et al. Inception-V4, Inception-ResNet and the impact of residual connections on learning[C]//Proceedings of the 31st AAAI Conference on Artificial Intelligence. Washington D. C., USA: AAAI Press, 2017: 1-8.

[4] HE Kaiming, ZHANG Xiangyu, REN Shaoqing, et al. Deep residual learning for image recognition[C]// Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2016: 770-778.

[5] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2014: 580-587.

[6] GIRSHICK R. Fast R-CNN[C]//Proceedings of 2015 IEEE International Conference on Computer Vision. Washington D. C., USA: IEEE Press, 2015: 1440-1448.

[7] REN S Q, HE K M, GIRSHICK R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.

[8] REDMON J, FARHADI A. YOLOv3: an incremental improvement[EB/OL]. [2019-11-02]. https://arxiv.org/abs/1804.02767.

[9] LIU W, ANGUELOV D, ERHAN D, et al. SSD: single shot multibox detector[C]//Proceedings of ECCV'16. Berlin, Germany: Springer, 2016: 21-37.

[10] LU Haiwei, XIA Haifeng, YUAN Xiaotong. Dynamic network pruning based on filter attention mechanism and feature scaling factor[J]. Miniature Microcomputer System, 2019, 40(9): 1832-1838. (in Chinese)
卢海伟，夏海峰，袁晓彤. 基于滤波器注意力机制与特征缩放系数的动态网络剪枝[J]. 小型微型计算机系统，2019，40(9)：1832-1838.

[11] JADERBERG M, VEDALDI A, ZISSERMAN A. Speeding up convolutional neural networks with low rank expansions[C]// Proceedings of 2014 British Machine Vision Conference. Nottingham, UK: British Machine Vision Association, 2014: 21-26.

[12] HASSIBI B, STORK D G. Second order derivatives for network pruning: optimal brain surgeon[C]// Proceedings of 1993 Conference and Workshop on Neural Information Processing Systems. Cambridge, USA: MIT Press, 1993: 164-171.

[13] HAN S, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural network [C]// Proceedings of 2015 Conference and Workshop on Neural Information Processing Systems. Cambridge, USA: MIT Press, 2015: 1135-1143.

[14] ANWAR S, HWANG K, SUNG W. Structured pruning of deep convolutional neural networks[J]. ACM Journal on Emerging Technologies in Computing Systems, 2017, 13(3): 1-18.

[15] HE Yang, LIU Ping, WANG Zhiwei, et al. Filter pruning via geometric median for deep convolutional neural networks acceleration[C]//Proceedings of 2019 IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2019: 4340-4349.

[16] LIU Zhuang, LI Jianguo, SHEN Zhiqiang, et al. Learning efficient convolutional networks through network slimming[C]// Proceedings of 2017 IEEE International Conference on Computer Vision. Washington D. C., USA: IEEE Press, 2017: 2736-2744.

[17] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters for efficient convnets[EB/OL]. [2019-11-02]. https://www.researchgate.net/publication/307536925_Pruning_Filters_for_Efficient_ConvNets.

[18] HE Yihui, ZHANG Xiangyu, SUN Jian. Channel pruning for accelerating very deep neural networks[C]// Proceedings of 2017 IEEE International Conference on Computer Vision. Washington D. C., USA: IEEE Press, 2017: 1389-1397.

[19] DONG Xuanyi, HUANG Junshi, YANG Yi, et al. More is less: a more complicated network with less inference complexity[C]//Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2017: 5840-5848.

[20] HE Yang, KANG Guoliang, DONG Xuanyi, et al. Soft filter pruning for accelerating deep convolutional neural networks[EB/OL]. [2019-11-02]. https://arxiv.org/abs/1808.06866.

编辑　宋　圆