文章编号：2096-1618(2021)06-0610-05

基于 YOLOv4-Tiny 模型剪枝算法

曹远杰[1,2]， 高瑜翔[1,2]， 刘海波[1,2]， 吴美霖[1,2]， 涂雅培[1,2]， 夏朝禹[3]
(1. 成都信息工程大学通信工程学院，四川 成都 610225；2. 气象信息与信号处理四川省高校重点实验室，四川 成都 610225；3. 中国民用航空总局第二研究所，四川 成都 610041)

摘要：针对 YOLO 系列算法参数量大、算法复杂度高提出一种基于 BN(batch normalization)层剪枝方法。该方法先通过对 BN 层的缩放系数 γ 以及平移系数 β 添加正则化约束训练，根据 BN 层参数以及卷积层各通道对网络贡献度等指标设定合适阈值进行剪枝。该方法在基本没有精度损失的前提下对 YOLOv4-Tiny 模型压缩 11 倍，计算量减少 72%，在 CPU 和 GPU 处理器下推理速度分别增快 44% 和 29%。实验结果表明，该剪枝方法能保持模型良好性能的前提下压缩模型，减少参数，降低算法复杂度。

关 键 词：深度学习；卷积神经网络；YOLOv4-Tiny；YOLOv3-Tiny；模型剪枝；稀疏训练
中图分类号：TP183 **文献标志码**：A
doi：10.16836/j.cnki.jcuit.2021.06.005

0 引言

随着计算机视觉的发展，对深度学习网络性能要求提高，出现许多优秀的深层神经网络模型[1-3]，如 VGGNet、GoogleNet、ResNet、DenseNet 等。伴随性能提高的同时，网络层数也在不断增加。这些网络都因为算法复杂度太高难以在嵌入式平台等资源较少的设备应用。

针对网络模型太大、参数量太多、算法计算量(FLOPs)大等问题，出现许多解决方案。如 SqueezeNet 网络，网络采用 squeeze 和 expand 两部分，squeeze 部分由 1×1 的卷积组成，expand 部分是将 1×1 和 3×3 两种层的输出特征图进行合并(concat)；MobileNets 网络，网络主要采用深度可分离卷积[4-5]替代传统卷积。深度可分离卷积由通道卷积和点卷积组成，相较于传统卷积在卷积核越大的情况下能降低更多的参数量和计算量。轻量化模型在不同的处理器上展示性能也不同，由于采用深度可分离网络会增加模型访存量，也会降低模型性能。

深度学习网络模型有很大一部分参数对结果没有影响[6]，为压缩深度学习模型，除设计轻量级结构以外还可以对模型进行剪枝[7]，通过对每个卷积层进行结果贡献度排序，减去冗余的通道，可以有效地压缩模型，降低计算量。文献[8]提出对网络迭代剪枝，获得一个精简模型，最终在没有精度损失前提下 AlexNet 参数量减少 9 倍。有研究者提出不需要稀疏卷积库的支持，直接对权重大小排序进行剪枝。模型中加入稀疏约束项可以使卷积中的部分参数趋近于 0，获得稀疏权值[9]。姚巍巍等[10]提出针对 BN 层 γ 参数添加 L1 正则化稀疏约束，大部分的神经元输出为 0，通过对这些不重要神经元进行修剪来迭代优化网络。

基于以上方法，针对 YOLOv4-Tiny 模型提出一种更精确的剪枝方法完成轻量化模型设计。该方法在 BN 层参数 γ 和 β 添加 L1 正则化约束，然后对每个通道权值的绝对值均值大小排序，根据每个通道权重绝对值的均值、$|\gamma/\sqrt{\sigma_\beta^2+\varepsilon}|$ 和 $\beta-\mu_\beta\gamma/\sqrt{\sigma_\beta^2+\varepsilon}$ 这 3 个值判断是否满足剪枝条件。该方法无疑比单纯判断 γ 系数或者卷积层大小更可靠。模型剪掉大量参数后精度会下降，微调后即可恢复精度。

1 YOLOv4-Tiny

YOLO(you only look once)算法[11]是典型的目标检测 one stage 方法，将预测和分类通过一个网络直接得出结果，特点就是推理速度很快，检测效率高。2020 年提出了 YOLOv4[12]和 YOLOv4-Tiny 两种目标检测算法。YOLOv4-Tiny 相较于 YOLOv3-Tiny 算法结构更复杂，但精度有小提升，并用参数更少。与 YOLOv3-Tiny 类似，输出两种尺度大小的特征图[13]。YOLOv4-Tiny 在结构上采用 CSP 结构作为主干网络，该结构将第一个卷积的一半特征图用于结构内卷积使用，最终再进行合并(concat)操作。YOLOv4-Tiny 网络结构图 1 所示。

收稿日期：2021-04-18
基金项目：四川省教育厅高校创新团队资助项目(15TD0022)

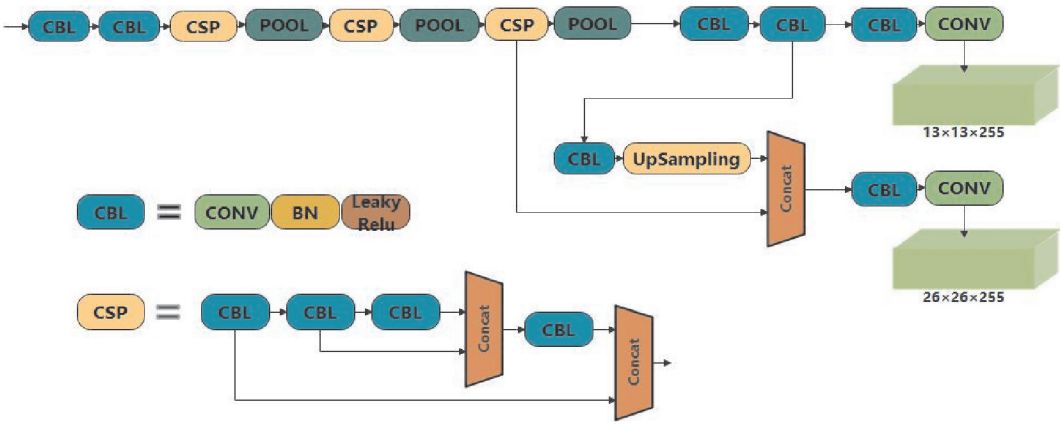


图 1　YOLOv4-Tiny 结构

2　YOLOv4-Tiny 的稀疏化剪枝

针对深度学习网络模型冗余参数过多导致算法复杂度高,推理速度慢等问题,先阐述模型的稀疏训练原理,再对稀疏后的权重和 BN 层进行大小排序,通过设定的阈值进行剪枝完成 YOLOv4-Tiny 的轻量化模型设计。

2.1　稀疏化训练

主要通过对 BN 层添加 L1 正则化约束对 BN 层参数稀疏化。Batch Normalization[14] 是 Google 提出的一种训练技巧,在模型训练时将 BN 层加入每个卷积后,可解决训练收敛慢等问题,还在一定程度上控制梯度爆炸。假设卷积层输出为 X,则经过 BN 层后输出为

$$Y_{bn}=\frac{\gamma(X-\mu_\beta)}{\sqrt{\sigma_\beta^2+\varepsilon}}+\beta \tag{1}$$

其中,Y_{bn}为 BN 层输出,γ 为 BN 层缩放系数,β 为平移系数,μ_β 和 σ_β^2 为均值和方差,ε 为常数,YOLOv4-Tiny 算法中一般取值为0.001。

LeakyRelu 层的输出为

$$Y_{\text{Relu}}=\text{LeakyRelu}\left(\frac{\gamma}{\sqrt{\sigma_\beta^2+\varepsilon}}\cdot X+\beta-\mu_\beta\frac{\gamma}{\sqrt{\sigma_\beta^2+\varepsilon}}\right) \tag{2}$$

其中 Y_{Relu}为激活层输出,LeakyRelu 为激活函数。当激活函数内的值趋近于 0 时可以看作输出为 0,对后面的卷积没有用。基于 LeakyRelu 激活函数,根据式(2),当参数 $a=\gamma/\sqrt{\sigma_\beta^2+\varepsilon}$ 和 $b=\beta-\mu_\beta\gamma/\sqrt{\sigma_\beta^2+\varepsilon}$ 较小时,通过激活函数后输出对网络贡献非常小,可以将其剪掉。由于均值,方差和 ε 都为定值,所以只需要判断 γ 和 β 两个参数足够小即可。为方便判断,对 BN 层 γ 和 β 参数都进行 L1 正则化约束。L1 范数公式为

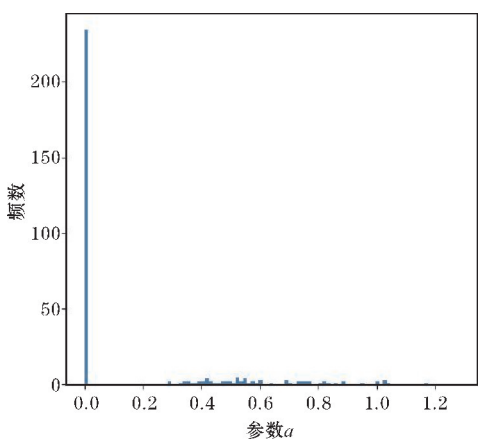
$$\Omega(\theta)=\sum_{i=0}^{n}||\theta_i||_1 \tag{3}$$

L1 范数是求各元素绝对值之和,假设目标函数为

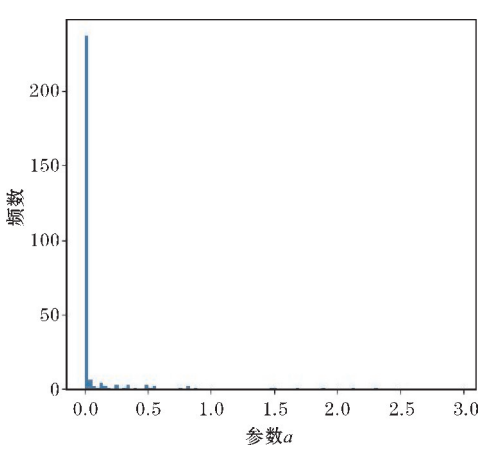
$$J'(\theta;Z,y)=J(\theta;Z,y)+\alpha\Omega(\theta) \tag{4}$$

其中,J 为原目标函数,J'为添加 L1 正则化约束后的目标函数,θ 为参数,Z 为输入,y 为标签,α 为正则化系数,控制约束力度。由式(4)可知,当 α 比较大时,要使 J'尽可能小,θ 向 0 趋近,因此 L1 正则化约束可以将参数稀疏化。

对 BN 层添加 L1 正则化约束后,参数 $a=\gamma/\sqrt{\sigma_\beta^2+\varepsilon}$ 和参数 $b=\beta-\mu_\beta\gamma/\sqrt{\sigma_\beta^2+\varepsilon}$ 稀疏度如图 2 所示。



(a) a 参数



(b) b 参数

图 2　BN 层参数频数分布直方图

图 2 是采用 α 为0.0001时稀疏训练后的第 15 个 BN 层参数频数直方图,由于 15 层参数较多,比较直观,更具普适性。由图 2 可知,添加 L1 正则化约束后的参数有大部分参数都趋近于 0,可以将大部分的冗余通道剪去。经过实验验证,由于采用 $\alpha=0.001$训练后精度下降比较大 ,故采用 $\alpha=0.0001$进行稀疏训练,训练后精度还提升了0.1个百分点。

2.2 剪枝和优化

模型的 BN 层通过稀疏化后对参数 a 和 b 进行大小排序,假设其中参数 a 的集合为 $A=\{a_1,a_2,a_3,\cdots,a_i\}$,参数 b 的集合为 $B=\{b_1,b_2,b_3,\cdots,b_i\}$。将卷积层的每层通道的权值的绝对值求和取平均后得到集合 $W=\{w_1,w_2,w_3,\cdots,w_i\}$。第一层 W 的频数分布直方图如图 3 所示,第一层参数较少。

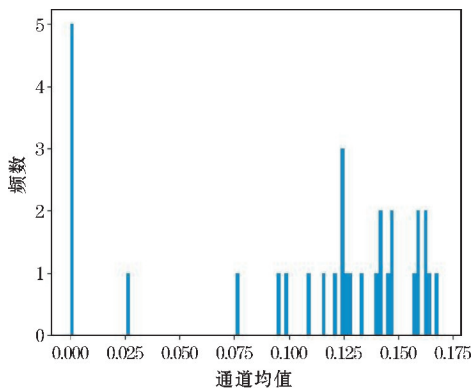


图 3 第一层卷积层通道均值频数分布直方图

由图 3 可知,稀疏训练后各通道均值差别较大。W 集合中可以剪掉的通道比普通通道均值小了约 10^7 倍。假设输入全为 1,权值、a 和 b 参数都为正,$(a_i\times w_i\times \boldsymbol{E}+b_i)$可以粗略地看作第 i 层 BN 层的输出特征图,此处 $\boldsymbol{E}$ 为和输出特征图大小一样的全 1 矩阵。W 集合各元素取均值为 $\bar{w}$,$\bar{w}$ 由于通道均值差别大的原因向 0 趋近,当 $a_i<\bar{w}$ 且 b_i 小于一个很小的数时,可以判别为第 i 层输出对结果贡献小。判别式为

$$(|a_i|<\bar{w}\times k)\&\&(b_i<k') \tag{5}$$

其中,$|a_i|$为第 i 层 BN 层参数$|\gamma/\sqrt{\sigma_\beta^2+\varepsilon}|$,$\bar{w}$ 为该层卷积层所有权值的均值,也就是各通道均值再取均值;k 和 k'为常数,k 用以调节剪枝率,k'设为一个很小的数,此处设为各通道大小的倒数;&& 为与上的意思;b_i 为第 i 层 BN 层的参数$\beta-\mu_\beta\gamma/\sqrt{\sigma_\beta^2+\varepsilon}$。

如式(2)所示,BN 层则是在做线性运算,剪枝后即可将 BN 层融入到卷积层当中以减少算法访存量,加快推理速度。假设第 i 层权重为 Q_i,则 BN 层合并到卷积层后的权重和偏置为

$$Q_i^{'}=\frac{\gamma\cdot Q_i}{\sqrt{\sigma_\beta^2+\varepsilon}} \tag{6}$$

$$\beta'=\beta-\frac{\gamma\cdot\mu_\beta}{\sqrt{\sigma_\beta^2+\varepsilon}} \tag{7}$$

其中,$Q_i^{'}$ 为融合第 i 层 BN 层后的权重,Q_i 为第 i 层原始权重,β'为融合 BN 层后的第 i 层偏置,其他参数都为对应 BN 层的参数。

3 实验结果及分析

为验证该剪枝算法,实验在 YOLOv4-Tiny 和 YOLOv3-Tiny 以及采用 GhostNet 结构所构造的轻量级网络(YOLO-GhostNet)[15] 3 种模型进行实验验证。

3.1 实验平台和数据

由于硬件设备有限,实验均采用 10097 张饮料数据集,数据集包含 4 个种类,分别为可乐、绿茶、雪碧、王老吉,每个种类大约 2500 张。其中测试集 908 张,训练集 8281 张。实验软件平台为 PyCharm,训练与验证框架为 Keras Tensorflow1.13-GPU 与 CPU 处理器下进行。硬件平台为 CPU (AMD Ryzen5 3500X),GPU (GTX1060 6 G),16 G内存(3600 MHz)。训练均采用 Batchsize=4。

3.2 实验结果及分析

根据式(5),当 $k=1$ 时,YOLOv4-Tiny 剪枝前后各卷积层对比如表 1 所示。

表 1 YOLOv4-Tiny 剪枝前后各卷积层大小

卷积层	剪枝前	剪枝后
Conv1	32	27
Conv2	64	47
Conv3	64	42
Conv4	32	31
Conv5	32	28
Conv6	64	58
Conv7	128	75
Conv8	64	39
Conv9	64	32
Conv10	128	54
Conv11	256	63
Conv12	128	49
Conv13	128	43
Conv14	256	48
Conv15	512	139
Conv16	256	51
Conv17	512	87
Conv19	256	28
Conv20	256	85

表 1 中前面的层剪枝相对较少,后面几层剪枝量

比较大,不同层根据冗余通道的多少剪枝率也不相同,整体剪枝率达到 91%。其中第 3、7、11 层卷积为如图 4 所示的 CSP 块中黄色部分的第一个卷积层,该层分为了两步,第一步是将卷积层的后半部分通道进行下一步卷积,如图 4 黄色解析部分的上面一条卷积路线。第二步是将所有通道进行卷积,如图4黄色解析部分的下面一条卷积路线。最终将两条路线输出的特征图进行合并。

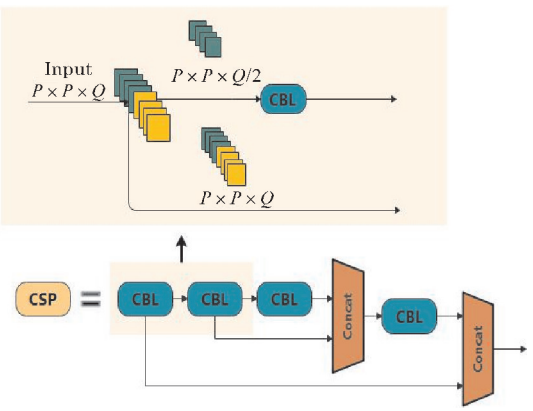


图 4　CSP 模块示意图

CSP 结构中的黄色部分解析如图 4 中下面黄色部分所示。对该层剪枝后分布会变得不均匀,所以剪枝后不能再将第一个卷积的一半通道进行后面卷积了。根据每层剪枝通道的不同,分出的通道也不一样。如第三个卷积层剪枝后只有 42 个输出通道,如果分一半通道进行图 4 所示的第一步卷积,那将会破坏之前训练的结构。根据剪枝通道编号可知,第三个卷积层的前半部分剪 15 个通道,后半部分剪 7 个通道。因此,第三个卷积层只分出后面的 25 个通道进行第一步卷积。

YOLOv3-Tiny 和 YOLOv4-Tiny 及以 GhostNet 卷积模块所构建的轻量级网络(YOLO-GhostNet)模型通过稀疏训练和稀疏训练后通过该剪枝方法后在饮料数据集下各项数据对比如表 2 所示。

表 2　YOLOv4-Tiny 和 YOLOv3-Tiny 及 YOLO-GhostNet 通过稀疏训练后模型与剪枝后模型参数对比

Network	mAP/%	Input	Run Time CPU/ms	Run Time GPU/ms	BFLOPs	Weight size/Mb
YOLOv4-Tiny	79.44	416	98.86	20.41	6.79	22.6
L1e4_YOLOv4-Tiny	79.54	416	98.86	20.41	6.79	22.6
L1e4_YOLOv4-Tiny-P(BN merger)	79.53	416	55.70	14.55	1.92	2.06
YOLOv3-tiny(Ciou)	77.38	416	87.55	20.72	5.45	33.2
L1e4_YOLOv3-tiny	77.14	416	87.55	20.72	5.45	33.2
L1e4_YOLOv3-tiny-P(BN merger)	77.14	416	49.10	15.57	1.41	1.43
YOLO-GhostNet	79.43	416	58.79	16.69	2.13	2.24
L1e4_YOLO-GhostNet	79.42	416	58.79	16.69	2.13	2.24
L1e4_YOLOv4-GhostNet-P (BN merger)	79.38	416	42.89	14.75	2.06	1.43

表 2 中以 L1e4 开头是代表以 $\alpha=0.0001$ 的正则化系数进行稀疏训练的模型,以-P 结尾为剪枝后的模型。BN merger 代表该模型将 BN 层进行合并。YOLOv3-Tiny 是采用 YOLOv4-Tiny 的损失函数所训练,所以精度有提升。据表 2,输入尺寸均采用 416×416,可以看到 YOLOv4-Tiny 剪枝后在精度方面相对于 L1e4_YOLOv4-tiny 模型值降低了0.01个百分点,剪枝率达 91%,计算量只有原来的 28%。在 CPU 推理速度加快 44%,在 GPU 环境下推理速度加快 29%。YOLOv3-Tiny 和 YOLO-GhostNet两种模型体积差别较大,这也与模型冗余通道的多少有关。在稀疏训练后通过该剪枝算法剪枝率分别为 96% 和 36%,精度损失都在0.05% 以内。3 种模型在稀疏训练后通过该剪枝算法都能维持原有性能的基础上降低算法复杂度和参数量,加快模型推理速度。

4　结束语

所提出剪枝算法在不同模型剪枝效果和剪枝率根据模型冗余通道的多少也会不同,通过对 BN 层参数添加正则化约束训练后再进行剪枝可减去大量参数的同时保持精度。该剪枝算法可以配合设计轻量级结构的模型进行使用,在保持精度不降低的前提下可以更大程度地轻量化模型。在以 GhosNet 所设计的 YOLOv4-Tiny 的轻量级模型进行剪枝后模型尺寸可降低至1.43 Mb,推理速度在不同处理器下可提高 29%~57%,可更好完成轻量化模型设计。

参考文献:

[1] Szegedy C,Liu W,Jia Y,et al. Going deeper with convolutions[C]. Proceedings of the IEEE confer-

ence on computer vision and pattern recognition. 2015: 1-9.

[2] K He, X Zhang, S Ren, et al. Deep Residual Learning for Image Recognition[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: 770-778.

[3] Huang G, Liu Z, Van Der Maaten L, et al. Densely Connected Convolutional Networks[C]. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017: 2261-2269.

[4] Howard A, Sandler M, Chu G, et al. Searching for MobileNetV3[C]. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019: 1314-1324.

[5] Chollet F. Xception: Deep learning with depthwise separable convolutions[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1251-1258.

[6] Denil M, Shakibi B, Dinh L, et al. Predicting parameters in deep learning[C]. Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2, 2013: 2148-2156.

[7] Molchanov P, Mallya A, Tyree S, et al. Importance estimation for neural network pruning[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2019: 11256-11264.

[8] Han S, Pool J, Tran J, et al. Learning both weights and connections for efficient neural networks[C]. Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1, 2015: 1135-1143.

[9] 叶会娟,刘向阳.基于稀疏卷积核的卷积神经网络研究及其应用[J].信息技术,2017,10(10):5-9.

[10] 姚巍巍,张洁.基于模型剪枝和半精度加速改进 YOLOv3-tiny 算法的实时司机违章行为检测[J].计算机系统应用,2020,29(4):41-47.

[11] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016: 779-788.

[12] Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: Optimal Speed and Accuracy of Object Detection[J/OL]. arXiv, 2020.

[13] Xu Z F, Jia R S, Liu Y B, et al. Fast method of detecting tomatoes in a complex scene for picking robots[J]. IEEE Access, 2020, 8: 55289-55299.

[14] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]. International conference on machine learning. PMLR, 2015: 448-456.

[15] 曹远杰,高瑜翔.基于 GhostNet 残差结构的轻量化饮料识别网络[J/OL]. https://doi.org/10.19678/j.issn.1000-3428.0059966.计算机工程,2021-04-18.

Model Pruning Algorithm based on YOLOv4-Tiny

CAO Yuanjie[1,2], GAO Yuxiang[1,2], LIU Haibo[1,2], WU Meilin[1,2], TU Yapei[1,2], XIA Chaoyu[3]

(1. College of Communication Engineering, Chengdu University of Information Technology, Chengdu 610225, China; 2. Meteorological Information and Signal Processing Key Laboratory of Sichuan Education Institutes, Chengdu 610225, China; 3. Second Institute of Civil Aviation Administration of China, Chengdu 610041, China)

Abstract: Due to the large number of parameters and high complexity of YOLO series algorithms, a pruning method based on the BN (batch normalization) layer is proposed. The method first adds regularization constraint training to the scaling coefficient γ and translation coefficient β of the BN layer, According to the parameters of the BN layer and the contribution of each channel of the convolutional layer to the network, an appropriate threshold is set for pruning. The proposed method comppresses the YOLOv4-Tiny model by 11 times with almost no loss of precision, reduces the computation amount by 72%, and increases the inference speed by 44% and 29% respectively under CPU and GPU processor. Experimental results show that the pruning method can compress the model, reduce the parameters and reduce the complexity of the algorithm while maintaining the good performance of the model.

Keywords: deep learning; convolutional neural network; YOLOv4-Tiny; YOLOv3-Tiny; model pruning; sparse training