网络出版时间：2019-06-27 11:51:18
网络出版地址：http://kns.cnki.net/kcms/detail/61.1450.TP.20190627.1115.086.html

基于深度神经压缩的 YOLO 优化

陈莉君，李 卓
（西安邮电大学，计算机学院，陕西，西安 710100）

摘 要： 从 AlphaGo 开始，深度学习渐渐的进入业内研究者的视线。深度学习近些年大热的主要原因是由于近些年设备的计算力的增加，尤其是图形处理器对于浮点数运算的有力支持。YOLO 在提出一种新的目标检测方法的同时，由于其过多的网络层数，带来对于存储空间巨大的需求。因此需要对于模型进行压缩，减少对于存储空间的需求。在传统压缩过程中单独使用剪枝或者量化方法，压缩后的模型依然存在一定的冗余。因此提出了一个结合剪枝和量化的方法对于模型进行压缩。本文针对在原始 YOLO 模型没有对于模型的测试方法以及对于模型稀疏度评估的手段，进行优化。在压缩的过程展示中，明确的标明每一层的稀疏度。实验证明 YOLO 模型在 VOC2012 数据集条件下，在保持接近原始模型的精度情况下，压缩了 10 倍。

关键词： 模型压缩；深度学习；目标检测；权重量化
中图分类号： TP 183 **文献标志码：** A

YOLO optimization based on deep neural compression

Chen Lijun，Li Zhuo
（School of Computer, Xi'an University of Posts & Telecommunications, Shaanxi ）

Abstract: Beginning with AlphaGo, deep learning has gradually entered the eyes of industry researchers. The main reason for deep learning in recent years is the increase in the computing power of devices in recent years, especially the strong support of graphics processors for floating-point operations. While YOLO proposes a new object detection method, it has a huge demand for storage space due to its excessive number of network layers. Therefore, it is necessary to compress the model to reduce the need for storage space. In the traditional compression process, pruning or quantification methods are used separately, and the compressed model still has some redundancy. Therefore, a method combining pruning and quantification is proposed to compress the model. This paper is optimized for the original YOLO model without the test method for the model and the means for model sparsity evaluation. In the process of compression process, the sparsity of each layer is clearly indicated. Experiments show that the YOLO model is compressed 10 times under the condition of VOC2012 data set while maintaining the accuracy close to the original model.

Keywords: deep compression; YOLO; pruning; quantification

0 引言

从 AlphaGo 开始，深度学习渐渐的进入业内研究者的视线。深度学习近些年大热的主要原因是由于近些年设备的计算力的增加，尤其是图形处理器（Graphics Processing Unit，缩写：GPU）对于浮点数运算的有力支持。YOLO[1]的提出了一种在目标识别领域新的方式，将检测和回归问题集合在一起，大大的增加了对于目标的检测速度。但是，YOLO 的网络层数相较于传统的网络也增加很多，导致训练和推理的计算量会大大增加。尤其是嵌入式设备和移动设备，这些设备可能并不能很好的支撑层数增加之后带来的计算量增加。

基金项目:陕西省科技厅自然科学基础研究计划（2018JM6047）

目前许多网络在设计的过程中，都会具有一些复杂的结构设计，例如全连接层，增加卷积层，这样会带来更高的精确度。相对的，这样的设计造成了网络模型中存在更多的冗余，增加了计算复杂度。优化压缩模型大致有三个方向:1) 更为精细的网络设计，简化卷积层和全连接层的形式 [2]。2) 对模型进行裁剪，在结构复杂的神经网络中，往往存在着大量的参数冗余，因此可以寻找一个适当的评定方法进行剪枝优化[3]。3) 为了保持原有模型中的数据精度，一般常见的网络中，都会以 32bit 的浮点类型保存模型权重[4]。这种保存方式，增加了数据存储的大小和重载模型的计算复杂度。对于以上的情况，可以将数据进行量化或者二值化。通过量化或者二值化之后，可以对于存储空间进行压缩。此外，还可以对卷积核进行稀疏化，将卷积核一部分裁剪为 0，从而减少再次加载模型的计算量。

本文着重讨论第二中、第三种方法，对 YOLO 模型进行剪枝，量化，降低模型对于存储空间的占用，减少计算复杂度。

1 相关工作

1.1 深度神经压缩

在计算机视觉领域，深度神经网络已经是一个最佳的实践方法。然而，由于有限的计算和存储资源，这些模型很难转移到嵌入式系统中。更多的研究者逐渐开始关注对于深度神经网络模型的压缩方法[5,6,7]上。对于压缩的方法，主要分为两种： 训练前压缩和训练后压缩。前一种的方法主要是在训练期间对于权重、激励和梯度进行压缩。例如 Courbariaux[8] 在训练过程中使用二值化权重和激励训练二值化神经网络，从而实现在多个数据集上的高压缩的效果。但是这样的压缩方法，需要压缩整个数据集，大大的增加了压缩深度模型的时间成本。另一种方法主要是侧重于压缩已经训练好的神经网络。例如 Denil [9]等人，使用权重矩阵的低秩分解来减少网络中的参数数量；Han[10]等人使用三阶段对于神经网络进行压缩，包括剪枝，训练量化和霍夫曼编码。以这样的方法减少网络的存储需求，并且不减少其准确度。本文中将使用 han 等人的思路对于 YOLO 进行压缩，并且对于压缩后的网络进行稀疏属性分析。

1.2 目标识别

以分类为方法的目标识别，即预测边界框和相关类的概率，现在已经成为目标识别领域中中常见的方法之一。DMP[11]这种经典的方法常用滑动窗口和分类器来检测窗口内是否存在对象。R-CNN[12]首先提出了使用 region proposal 生成潜在的边界框，然后再这些区域中运行分类器。这样大大的减少了搜索的空间和时间成本。Ouyang[13] 提出了一种基于 object proposal 的方法，提出了一种用于物体检测的可变形深度卷积神经网络。网络学习了大量物体的特征和变形特征。相较于 GoogleNet[14],准确率能高出 6.1%。但是，遗憾的是，这种网络的计算成本比较高，无法部署在嵌入式系统和设备上。为了解决计算成本问题，Redmon 等人提出一个命名为 YOLO 的深度神经网络。模型使用单个神经网络来预测边界框和类概率。后来 Redmon[15]等人又在 YOLO 的基础上，改进模型，达到了以 40 FPS 获得 78.6 mAP。本文中的压缩工作就是建立在 YOLO 模型之上。

2 方案

原始的 YOLO 模型在完成训练后，模型大小达到 200M，对于嵌入式设备和移动设备而言。200M 的模型进行内存预读，会对设备的性能造成较大的影响。本节将从压缩和量化两方面对于模型进行优化。并且对于压缩后的模型进行稀疏度评估。以达到优化模型对于存储空间的要求，并且可以直观的了解到每一层剪枝的情况。

2.1 模型压缩

模型压缩主要分为两个步骤，一个是剪枝，另外一个是对于权值进行量化存储。框架如图 2.1 所示

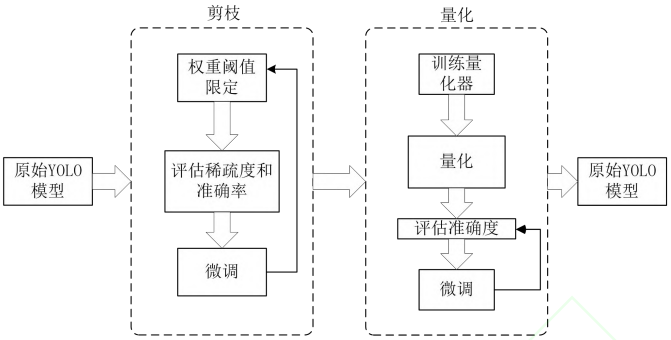

图 2.1 网络压缩的框架图

2.1.1 剪枝

剪枝主要目的是保存 YOLO 中重要的链接来达到降低存储数量和计算复杂度的目的。传统的神经网络训练过程中，在神经网络训练之前，神经网络的框架结构就已经被固定了。用户只需要进行数据的输入，就可以在迭代训练后获得需要的权重。但是这种固定框架结构的方式，导致不能在训练的过程中，随时对于神经网络的结构进行优化。因此，常规训练出的模型文件大小都不适合于部署在嵌入式设备或者移动设备上。YOLO 常规训练后的模型大小达到 200M，对于嵌入式设备，将模型预读进内存中，提供给推理过程使用，将会消耗设备的所有内存。所以，利用深度神经压缩的剪枝思路，对于 YOLO 模型进行压缩，是提高 YOLO 在各种设备上通用性的一个方式。剪枝过程分为四个步骤；

1. 通过训练找到权重小于阈值的神经网络链接
2. 删除权重小于阈值的神经网络链接
3. 重新对于神经网络进行训练

在训练的过程中使用公式

$$L(\mathcal{W})=\frac{1}{N}\sum_{i=1}^{N}\left(f\left(\mathcal{X}_i;\mathcal{W}\right)-y_i\right)^2+\frac{\lambda}{2}\|\mathrm{w}\|^2 \tag{1}$$

对权重进行计算，使一部分权重趋向于 0，然后将小于阈值的链接剪枝。使用公式的目的是能够减少在训练剪枝过程中的过拟合现象，并且可以保持较高的精度。在剪枝过程中，从大到小的对于阈值进行细化。流程图如图 2.2 所示：

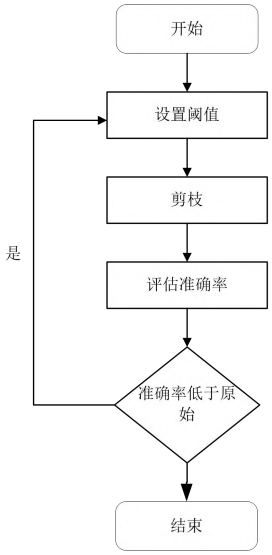

图 2.2 剪枝流程图

稀疏度评估步骤将在 2.2 节中介绍。稀疏度评估将会展示剪枝后，每一层中剩余的链接的数量。

2.1.2 量化

量化是以一个适度的误差为代价，使得精度很高的数值能够以更少的位数进行存储。量化主要分为三个步骤

1. k-means
2. 确定量化阈值
3. 微调

先分别介绍三个步骤的相关做法。量化步骤主要通过 k-means 实现。K-means 对于质心的选取对于最后量化的结果有很大的影响。选择质心有三种方法，均匀量化、随机量化和按密度量化。这里选择均匀量化的方法进行初始化。K-means 定义式如式 2 所示

$$c_k^{init} = w_{min} + k \times \frac{w_{max} - w_{min}}{2^n} \tag{2}$$

其中 n 为需要量化的位数，w 为权重。公式根据规定的量化的位数进行初值的选取，在过程中使用式 3 进行聚类

$$\arg\min_{c} \sum_{i=1}^{k} \sum_{\omega \in c_i} |\omega - c_i|^2 \tag{3}$$

在完成聚类之后，将进行稀疏度和准确率的评估，依据评估后的结果，实行微调。微调主要是对于 k-means 的质心进行调整。在微调过程中，因为在初始化之后，权重矩阵已经成为稀疏矩阵，有部分位置已经被置零，这些位置的梯度相应的在微调过程中也会被放弃。微调的对象为式 2 计算出的质心。

微调的方法为使用梯度矩阵对于执行进行微调。微调主要是将权值的梯度进行求和再乘以学习率，然后再将这部分的数值从质心中减去。公式如式 4 所示

$$c_n^k = c_k^{n-1} - lr \times \sum_{w_{ij} \in c_k} grad\left(w_{ij}\right) \tag{4}$$

其中c_k^n为第 n 次微调之后的结果，lr 为学习率，c_k为权重的集合，grad(w)为梯度。量化完成后，剪枝后的稀疏矩阵将成为一个稀疏矩阵加一个查找表。

2.2 测试和评估模型

2.2.1 测试模型

在 YOLO 原始模型中，并没有计算任何准确率的方法。这样就不能够将压缩后的结果和预期结果进行对比，因此，实验中需要加上对于模型的测试功能。测试将分为三个指标进行测试 True Positive 被模型预测为正的正样本、True Negative 负样本和 False Positive 被模型预测为正的负样本。以及精度 p 和召回率 R

使用下列公式进行精度 p（式 5）和召回率 R（式 6）

$$p = \frac{TP}{TP + FP} \tag{5}$$

和

$$R = \frac{TP}{TP + FN} \tag{6}$$

2.2.2 稀疏度评估

在剪枝和量化过程中，会将原始权重矩阵改变为稀疏矩阵。稀疏编码在神经网络模型压缩领域中有着很大的作用。在每一次的剪枝和量化中，可以量化每一次的结果，并且根据结果进行修正。代码实现如图 2.3 所示

```
for name,f in model.named_parameters():
        num = f.view(-1).shape[0]
        total_num += num
        sparse = pt.nonzero(f).shape[0]
        total_sparse+= sparse
        print("\t",name,(sparse)/num)
        result.append((sparse)/num)
```

图 2.3 修正代码

稀疏度评估函数主要是用现有的权重稀疏矩阵和原始网络的权重矩阵进行对比，算出每一层

的权重稀疏度和贝叶斯数值。

3 实验结果及分析

3.1 实验结果

实验采用 voc2012 数据集，并且在基于 PyTorch 的 YOLO 模型上进行实验。实验表明在阈值接近 0.9 时，准确率接近原始模型。在表 1 中展示权重比例从 0.9 到 0.6 模型的 mAP。并且将模型的部分层的稀疏度进行打印。

表 1 阈值变化对于 mAP 的影响

权重比例	mAP
0.9	83.5
0.8	83.1
0.7	81.2
0.6	70.5
0.5	65.2
均值	76.7

图 3.1 展示了部分层在剪枝和量化后的稀疏度，最后可以看出，压缩后的模型只有原始模型的 0.013 权重数量。

```
Layer sparse
      conv1.weight 0.9548611111111112
      conv1.bias 0.953125
      conv2.weight 0.5300767686631944
      conv2.bias 0.46875
      fc1.weight 0.0011419179488201 53
      fc1.bias 0.0
      fc2.weight 0.592578125
      fc2.bias 0.4
Total: 0.013573435928595692
```

图 3.1 部分层的稀疏度

图 3.2 展示了在压缩过程中的微调步骤，每一次微调都会引起 mAP 的抖动，在执行多次微调之后，可以将准确率逼近没有压缩前的模型

```
0 0.8828125
100 0.8828125
200 0.9296875
300 0.9296875
400 0.921875
0.9236550632911392
```

图 3.2 微调过程中的 mAP 展示

3.2 分析

在实验中可以看出，深度神经压缩对于 YOLO 模型的压缩效果比较明显。最后的结果可以接近原模型的 1%大小。模型大大的减少了对于存储空间的需求。并且可以满足部署在嵌入式设备上的需求。

对比表 1 可以得出，对于权重比例不是减少的越多越好。过多的修建权重，将会造成模型准确率的大幅度下降。模型在保持 80%~90%权重的情况下表现较好。

在图 3.1 站时的过程中，部分层的权重即将被置零，这点说明，YOLO 网络在设计过程中，部分网络因为全连接的原因，出现冗余，这些链接权重是压缩模型过程中需要去剪枝的部分。

在图 3.2 过程中，看出，对于阈值的每一次微调都会对于 mAP 有着一定的影响，所以在压缩的过程中，需要不断的对于阈值进行调整。并且阈值调整的步长越细小，得出的结果越好。

4 结语

本方案基于深度神经压缩，对于 YOLO 进行压缩。单独使用剪枝或者量化方法，压缩后的模型依然存在一定的冗余。因此提出了一个结合剪枝和量化的方法对于模型进行压缩。并且原始 YOLO 模型没有对于模型的测试方法以及对于模型稀疏度评估的手段。本文从这两方面对于整个压缩过程进行优化。可以在压缩后直观的观察到神经网络中某些层在整个神经网络中的荣誉情况。给与设计网络的过程中一定的指导作用。实验证明，基于深度神经压缩的混合压缩方法，能够有效的减少 YOLO 模型中存在的参数荣誉，减少了网络模型对于与存储空间的需求。进一步的工作将在物理存储上进行优化。

参考文献

[1].Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time ObjectDetection[C].IEEE Conference on Computer Vision and Pattern Recognition. IEEEComputer Society, IEEE,NYC, 2016:779-788.

[2].LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[3].Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. science, 2006, 313(5786): 504-507.

[4].Han S , Mao H , Dally W J . Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding[J]. Fiber, 2015, 56(4):3--7.

[5].肖学锋. 深度神经网络的参数压缩及前向加速[D].华南理工大学,2018.

[6].杨岱桦,雷菊阳.深度神经网络稀疏化压缩感知的研究[J].化工自动化及仪表,2018,45(10):798-801.

[7].纪荣嵘,林绍辉,晁飞,吴永坚,黄飞跃.深度神经网络压缩与加速综述[J].计算机研究与发展,2018,55(09):1871-1888.

[8].Schmidhuber J. Deep Learning in neural networks: An overview.[J]. Neural Netw, 2015, 61:85-117.

[9].Denil M, Shakibi B, Dinh L, et al. Predicting parameters in deep learning[C]//Advances in neural information processing systems, MIT Press,London. 2013: 2148-2156.

[10].Han S, Mao H, Dally W J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding[J]. Fiber, 2015, 56(4):3--7.

[11].Felzenszwalb P F, Girshick R B, McAllester D, et al. Object detection with discriminatively trained part-based models[J]. IEEE transactions on pattern analysis and machine intelligence,

2010, 32(9): 1627-1645.

[12].Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE,NYC. 2014: 580-587.

[13].Ouyang W, Zeng X, Wang X, et al. DeepID-Net: Object detection with deformable part based convolutional neural networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(7): 1320-1334.

[14].Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE,NYC. 2015: 1-9.

[15].Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE,NYC. 2017: 7263-7271.

作者简介：

李卓(1994-)，男，西安安邮电大学，硕士研究生，研究方向：计算机技术

陈莉君(1964-)，女，西安安邮电大学，教授，研究方向：Linux 内核；