计算机视觉课程报告

HetConv: Heterogeneous Kernel-Based Convolutions for Deep CNNs[14]

邱林山
21821199
计算机科学与技术学院
rsamyau@163.com

摘要（Abstract）

本文提出了一种全新的深度学习架构，其中的卷积运算利用了异构核。相比于标准的卷积运算，本文提出的 HetConv（基于异构核的卷积）能在减少计算量（FLOPs）和参数数量的同时维持表征的效率。为了展现本文所提出的卷积的有效性，作者在 VGG[1] 和 ResNet[2] 等标准卷积神经网络（CNN）上进行了广泛的实验并给出了实验结果。作者发现，使用他们提出的 HetConv 过滤器替换了这些架构中的标准卷积过滤器之后，能在 FLOPs 方面实现 3 到 8 倍的速度提升，同时还能维持（有时候能提升）准确度。作者将其提出的卷积与分组/深度方面的卷积进行了比较，结果表明它能在显著提升准确度的同时将 FLOPs 降低更多。

1. 简介

卷积神经网络（CNN）在视觉和自然语言处理领域都已经取得了卓越的表现。进一步提升性能的总体趋势使得模型越来越复杂且越来越深。但是，使用更深度的网络，通过提升模型复杂度来提升准确度并不是毫无代价的——计算成本（FLOPs）会大幅增长。因此，为了降低 FLOPs 以让模型更高效，研究者们已经提出了各种不同类型的卷积运算/卷积过滤器。

1.1. 动机

目前提高 CNN 性能的主要手段有：

1. 增加模型宽度和深度。这样带来的代价是计算量会变得非常大。
2. 设计更高效的卷积结构。比如 depthwise conv[3]、pointwise conv[4]、groupwise conv[5]等。这一类卷积结构的特点是计算量相比标准卷积小，代替原始的标准卷积方式可以较少一定的计算量，同时保持较高的精度。
3. 为追求更快更小的网络结构，用剪枝（model pruning）的手段进行模型压缩（model compression），包括 connection pruning[6]、filter pruning[7、8]、quantization[9]等方式。模型剪枝的方式在某些方面很有效，缺点是得到一个好的模型通常需要大量的训练时间。训练-剪枝-再训练（fine-tune）的方式带来的计算资源和时间成本非常大，且最后不一定能得到令人满意的压缩模型。

目前来看，性价比最高的就是采用高效的轻量级网络，代表性的有：Xception[10]、MobileNet[11] 系列、ShuffleNet[12]系列。为了取得更高的性能，必须要精心设计网络结构，实现 accuracy-speed trade-off。

设计一个新的有效的网络结构不是一件容易的事，需要不断的试错、总结，文章从卷积方式入手，提出了一种新的卷积方式来代替原来的卷积，取得了更好的实验效果。

1.2. 本文工作

不同于模型压缩方法存在准确度大幅下降的问题，本文提出的方法与 ResNet[2] 和 VGGNet[1] 等标准模型的当前最佳结果相比也具有很高的竞争力。不同于需要预训练模型的剪枝方法，使用 HetConv，我们可以从头开始训练我们的模型，同时无损准确度。如果我们增大 FLOPs 剪枝的程度，剪枝方法还会造成准确度极大下降。相比于 FLOPs 剪枝方法，使用本文提出的 HetConv 过滤器能在 FLOPs 方面达到当前最佳水平。另外，剪枝过程的效率也很低，因为在剪枝后还需要大量时间来进行训练和微调。本文提出的方法具有很高的效率，并且从头开始训练时也能得到与原始模型相近的结果。

这是首个异构的卷积/过滤器。这种异构设计有助于提升已有架构的效率（降低 FLOPs），同时无损准确度。作者在 ResNet[2] 和 VGG-16[13] 等不同架构上进行了广泛的实验——只是将它们的原始过滤器替换成了该文提出的过滤器。作者发现，无需牺牲这些模型的准确度，我们就能大幅降低 FLOPs（3 到 8 倍）。这样的降低程度甚至比已有的剪枝方法还好很多。

2. 模型

2.1. HetConv

像标准卷积、DW、PW、GW 式的卷积的一个共同点就是所有的卷积核大小一致，称为“Homogeneous Convolution”，比如 3x3x256 的 conv2d，每个卷积核的尺寸都是 3x3 大小。

文章提出的“Heterogeneous Convolution”，顾名思义，就是卷积核的尺寸大小不一。比如在有 256 个通道的卷积核中，一部分 kernel size 为 1，另一部分 kernel size 为 3。

HetConv 带来的好处是可以无缝替换 VGG、ResNet、MobileNet 等结构的卷积形式，这种新的卷积形式，可以向标准卷积一样，从新开始训练，得到比 pruning 更好的性能效果。文章还指出，HetConv 与标准卷积一样，实现 latency zero。图 1 和图 2 展示了标准过滤器与 HetConv 过滤器之间的差异。

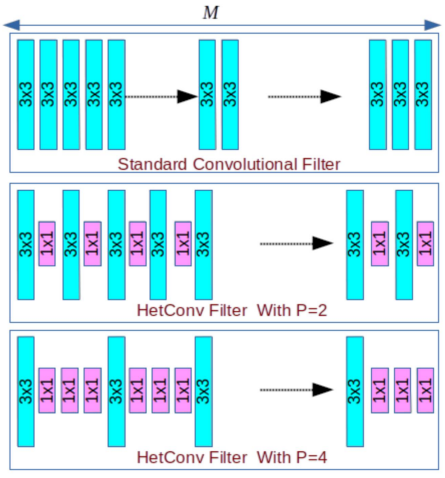

图 1:标准卷积过滤器(同构)和异构卷积过滤器(HetConv)之间的差异。其中 M 是指输入深度（输入通道的数量），P 是指 part（控制卷积过滤器中不同类型的核的数量）。在 M 个核中，M/P 个核的大小是 3×3，其余的都是 1×1。

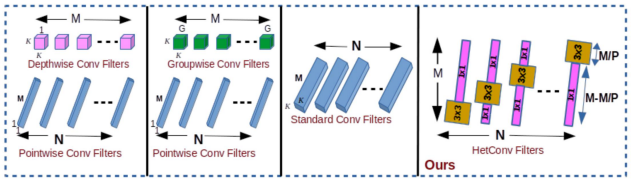

图 2：本文提出的卷积过滤器（HetConv）与其它高效卷积过滤器的比较。该文提出的异构过滤器的延迟为零，其它（GWC+PWC 或 DWC+PWC）则有一个单元的延迟。

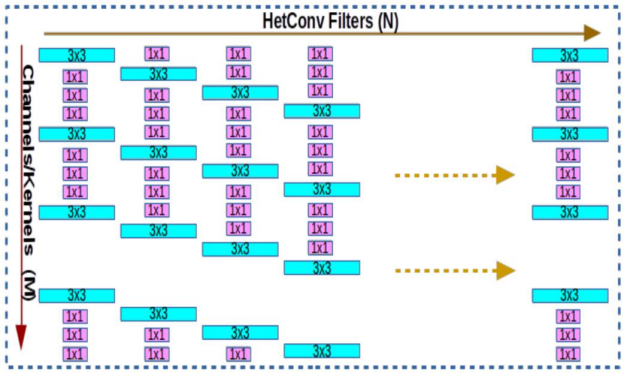

图 3：L 层处的卷积过滤器：本文提出的使用异构核的卷积过滤器（HetConv）。图中可以看到，每个通道都由 3×3 和 1×1 大小的异构核构成。在标准卷积过滤器中用 1×1 核替代 3×3 核能够在保持准确度的同时极大降低 FLOPs。一个特定层的过滤器排列成移位形式（即如果第一个过滤器从首个位置开始 3×3 核，则第二个过滤器从第二个位置开始 3×3 核，以此类推）。

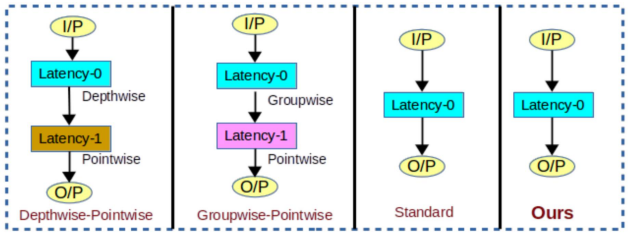

图 4：上图比较了不同类型的卷积的延迟情况。

2.2. 计算量分析

之后作者分别对比了 D+P、G+P 和 HetConv 的复杂度，计算过程比较简单。

标准卷积计算量：

$$FL_S = D_o \times D_o \times M \times N \times K \times K \rightarrow (1)$$

其中 Do 是卷积输出特征图的尺寸，M 是输入通道数，N 是输出通道数，K 是卷积核尺寸。HetConvolution：假设输入通道数为 M，有比例为 P 的卷积核尺寸为 K，这样的 kernel 数为 M/P，其他都是 1×1 大小，这样的 kernel 数为 (1-1/P) ·M，那么 K×K 卷积的计算量为：

$$FL_K = (D_o \times D_o \times M \times N \times K \times K)/P$$

1×1 卷积的计算量为：

$$FL_1 = \ (D_o \times D_o \times N) \ \times (1 - \frac{1}{P}) \times M$$

因此总的计算量为：

$$FL_{HC} = FL_K + FL_1 \rightarrow (2)$$

HetConvolution 与标准卷积的计算量之比：

$$R_{HetConv} = \frac{FL_k + FL_1}{FL_S} = \frac{1}{P} + \frac{1 - \frac{1}{p}}{K^2} \rightarrow (3)$$

当 P=1 时，HetConv 变为标准卷积，计算量之比为 1。

DW+PW 计算量：

$$FL_{MobNet} = D_o \times D_o \times M \times K \times K + D_o \times D_o \times M \times N \rightarrow (4)$$

DW+PW 与标准卷积的计算量之比：

$$R_{MobNet} = \frac{FL_{MobNet}}{FL_S} = \frac{D_o \times D_o \times M \times K \times K + D_o \times D_o \times M \times N}{D_o \times D_o \times M \times N \times K \times K} = \frac{1}{N} + \frac{1}{K^2} \rightarrow (5)$$

由公式(3)可知，增大 P，HetConv 变为标准卷积，控制 P 的大小，可以控制 accuracy 和 FLOPs。

极端情况下，P=M 的时候，公式（3）和（5）：

$$\frac{1}{M} + \frac{1 - \frac{1}{M}}{K^2} < \frac{1}{M} + \frac{1}{K^2} \rightarrow (6)$$

因此，MobileNet 比 HetConvolution 计算量更大。

GW+PW 计算量：

$$FL_G = (D_o \times D_o \times M \times N \times K \times K)/G + D_o \times D_o \times M \times N \rightarrow (7)$$

与标准卷积的计算量之比：

$$R_{Group} = \frac{FL_G}{FL_S} = \frac{1}{G} + \frac{1}{K^2} \rightarrow (8)$$

由公式（3）和（8）可知，P=G 的时候：

$$\frac{1}{P} + \frac{1 - \frac{1}{p}}{K^2} < \frac{1}{P} + \frac{1}{K^2} \rightarrow (9)$$

HetConv 的计算量比 GW+PW 更少。

3. 实验

作者选取了 VGG、ResNet、MobileNet 等网络，通过在 CIFAR-10、ImageNet 数据集上的实验验证 HetConv 的有效性。实验结果如表 1、表 2、表 3、表 4、表 5、表 6 所示。

Model	Acc%	FLOPs	FLOPs Reduced (%)	Parameters	Parameters Reduced (%)
VGG-16.P1	94.06	313.74M	–	15.00M	–
VGG-16.P1_SE	94.13	314.19M	–	15.22M	–
VGG-16.P2	93.89	175.23M	44.15	8.45M	43.68
VGG-16.P2_SE	94.11	175.67M	44.00	8.68M	42.99
VGG-16.P4	93.93	105.98M	66.22	5.17M	65.45
VGG-16.P4_SE	94.29	106.42M	66.08	5.41M	64.48
VGG-16.GWC4_PWC	92.76	107.67M	65.68	5.42M	–
VGG-16.P8	93.92	71.35M	77.26	3.54M	76.40
VGG-16.P8_SE	93.97	71.79M	77.12	3.77M	75.22
VGG-16.P16	93.96	54.04M	82.78	2.72M	81.86
VGG-16.P16_SE	93.63	54.48M	82.64	2.95M	80.59
VGG-16.P32	93.73	45.38M	85.54	2.31M	84.58
VGG-16.P32_SE	93.41	45.82M	85.39	2.54M	83.28
VGG-16.P64	93.42	41.05M	86.92	2.11M	85.95
VGG-16.P64_SE	93.33	41.49M	86.77	2.34M	84.63
VGG-16.DWC_PWC	91.27	38.53M	87.72	1.97M	–
VGG-16.PC	92.53	38.18M	87.83	1.93M	–
VGG-16.PC_SE	93.08	38.62M	87.69	2.15M	–

表 1：不同设置的 VGG-16 在 CIFAR-10 上的详细结果。

Method	Error%	FLOPs Reduced(%)
Li-pruned [21]	6.60	34.20
SBP [25]	7.50	56.52
SBPa [25]	9.00	68.35
AFP-E [3]	7.06	79.69
AFP-F [3]	7.13	81.39
VGG-16_P32 (Ours)	**6.27**	**85.54**
VGG-16_P64 (Ours)	**6.58**	**86.92**

表 2：在 CIFAR-10 数据集上，针对 VGG-16 架构的当前最佳模型压缩方法与本文的模型的比较。

Method	Error%	FLOPs Reduced (%)
Li-B [21]	6.94	27.6
NISP [41]	6.99	43.6
CP [11]	8.20	50.0
SFP [10]	6.65	52.6
AFP-G [3]	7.06	60.9
ResNet-56_P1	6.41	–
ResNet-56_P2	**6.40**	**44.30**
ResNet-56_P4	**6.71**	**66.45**
ResNet-56_P1_SE	7.16	–
ResNet-56_P2_SE	6.75	44.27
ResNet-56_P4_SE	7.79	66.42

表 3：在 CIFAR-10 上，针对 ResNet-56 架构的当前最佳模型压缩方法与本文的不同设置的模型的详细结果和对比。

Method	Accuracy (%)	FLOPs
MobileNet [12]	91.17	46.36M
MobileNet_P32	**92.06**	55.94M
MobileNet_P32_SE	**92.17**	56.91M

表 4：不同设置的 MobileNet 在 CIFAR-10 上的详细结果。

Method	Acc%(Top-1)	Acc%(Top-5)	FLOPs Reduced %
RNP (3X)[22]	–	87.57	66.67
ThiNet-70 [24]	69.8	89.53	69.04
CP 2X [11]	–	89.90	50.00
VGG-16_P1	71.3	90.2	–
VGG-16_P4	**71.2**	**90.2**	**65.8**

表 5： VGG-16 在 ImageNet 上的结果。与当前最佳的剪枝方法相比，本文的模型的准确度无损失，同时 FLOPs 也降低了很多。

Method	Error (top-1)%	FLOPs	FLOPs Reduced(%)
Li-B [21]	27.83	2.7G	24.2
NISP [41]	27.69	–	43.76
ResNet-34_P1	26.80	3.6G	–
ResNet-34_P4	**27.00**	1.3G	**64.48**
ResNet-34_P4_SE	**26.50**	1.3G	**64.48**

表 6：ResNet-34 在 ImageNet 上的结果。与当前最佳的剪枝方法相比，本文提出的模型的准确度无损失，同时 FLOPs 降低了显著更多。

4. 总结

文章提出了一种新的卷积方式，通过计算 FLOPs 和实验证明，HetConv 可以在更少计算量的上面取得更高的精度，文章也和 model conpression 进行了对比，从实验结果来看，效果也挺明显。HetConv 可以和现有的网络结构结合，操作简单方便。

对于 HeConv 的实用性方面可能还需要时间来证明，毕竟理论计算量和实际情况还是有些差距，另外作者没有在 detection、segmentation 任务做实验，但从分类任务来说，缺少一定的可信度。

References

[1] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. ICLR, 2015.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, pages 770–778, 2016.

[3] VincentVanhoucke.Learningvisualrepresentationsatscale. ICLR invited talk, 2014.

[4] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1–9, 2015.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, pages 1097–1105, 2012.

[6] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. ICLR, 2016.

[7] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In CVPR, pages 5058–5066, 2017.

[8] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In ICCV, page 6, 2017.

[9] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. ICLR, 2016.

[10] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In AAAI, volume 4, page 12, 2017.

[11] Saining Xie, Ross Girshick, Piotr Dolla´ r, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In Computer Vision and Pattern Recogni- tion (CVPR), 2017 IEEE Conference on, pages 5987–5995. IEEE, 2017.

[12] Francois Chollet. Xception: Deep learning with depthwise separable convolutions. CVPR, 2017.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. ICLR, 2015.

[14] Pravendra Singh, Vinray Kumar Verma, Piyush Rai, Vinay P.Namboodiri. HetConv: Heterogeneous Kernel-Based Convolutions for Deep CNNs. CVPR, 2019.