RepVGG: Making VGG-style ConvNets Great Again

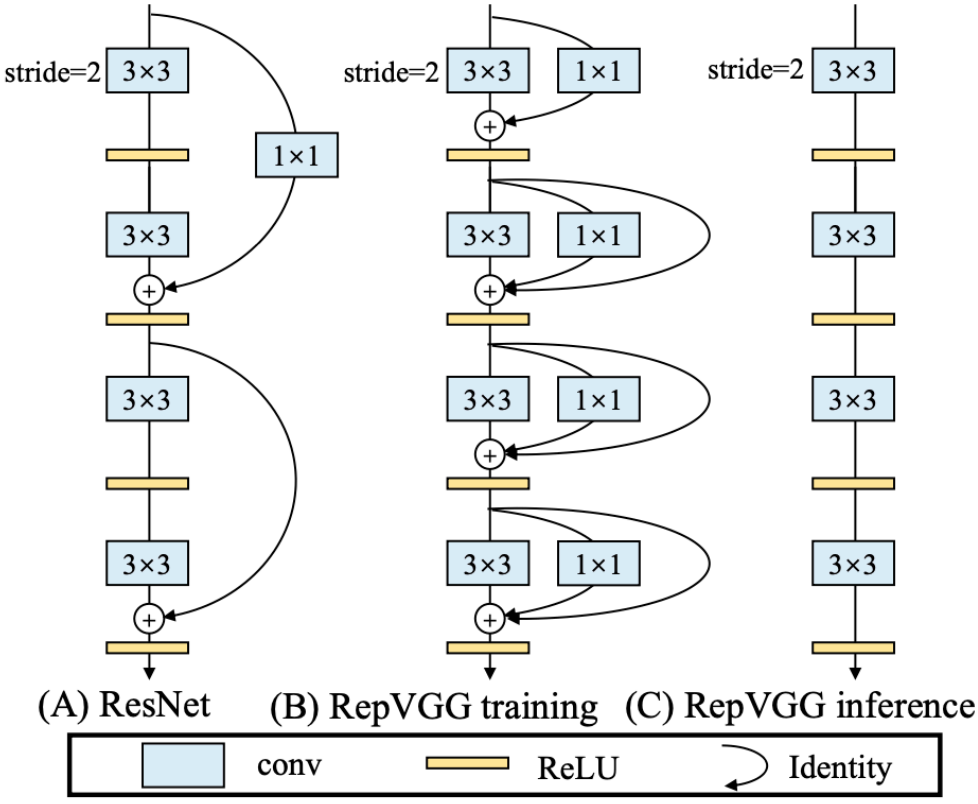
论文地址： https://arxiv.org/abs/2101.03697

代码地址： https://github.com/megvii-model/RepVGG

文章主要创新点

- 在VGG网络的Block块中加入了Identity和残差分支，相当于把ResNet网络中的精华应用 到VGG网络中
- 模型推理阶段，通过Op融合策略将所有的网络层都转换为Conv3*3，便于网络的部署和加速

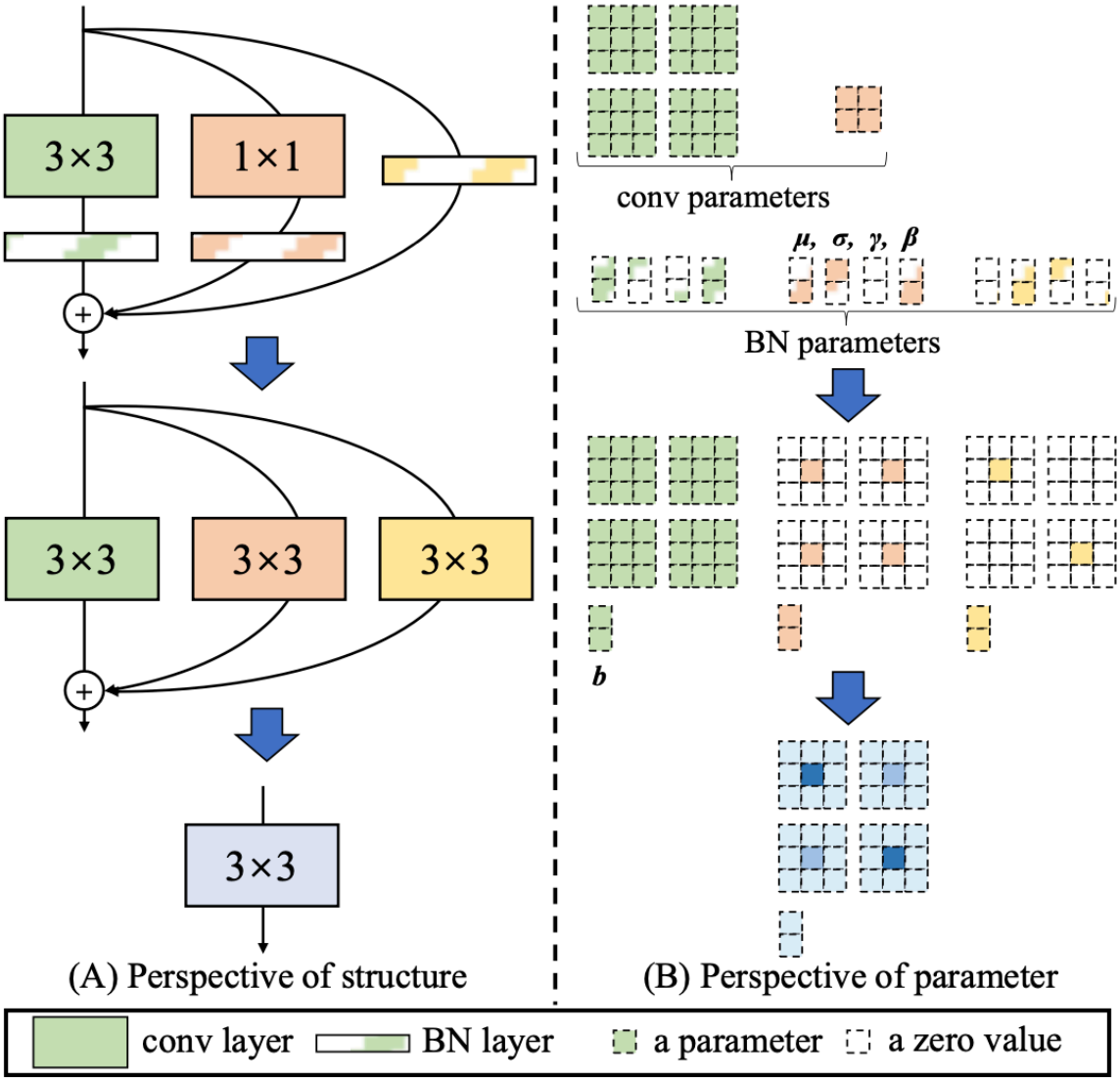
算法简介



上图展示了部分RepVGG网络，图A表示的是原始的ResNet网络，该网络中包含着Conv1*1的残差结构和Identity的残差结构，正是这些残差结构的存在解决了深层网路中的梯度消失问题，使得网络更加易于收敛。图B表示的是训练阶段的RepVGG网络架构，整个网络的主体结构和ResNet网络类似，两个网络中都包含残差结构。两个网络中的主要差异如下所述：（1）RepVGG网络中的残差块并没有跨层。（2）整个网络包含2种残差结构，第一种残差结构仅仅包含Conv1*1残差分支；第二种不仅包含Conv1*1的残差结构，而且包含Identity残差结构。由于残差结构具有多个分支，就相当于给网络增加了多条梯度流动的路径，训练一个这样的网络，其实类似于训练了多个网络，并将多个网络融合在一个网络中，类似于模型集成的思路，不过这种思路更加简单和高效。（3）模型的初始阶段使用了简单的残差结构，随着模型的加深，使用了复杂的残差结构，这样不仅仅能够在网络的深层获得更鲁邦的特征表示，而且可以更好的处理网络深层的梯度消失问题。图C表示的是推理阶段的RepVGG网络，该网络的结构非常简单，整个网络均是由Conv3*3+Relu堆叠而成，易于模型的推理和加速。

这种架构的主要优势包括：（1）当前大多数推理引擎都对Conv3*3做了特定的加速，假如整个网络中的每一个Conv3*3都能节省3ms，如果一个网络中包含30个卷积层，那么整个网络就可以节省3*30=90ms的时间，这还是初略的估算。（2）当推理阶段使用的网络层类别比较少时，我们愿意花费一些时间来完成这些模块的加速，因为这个工作的通用性很强，不失为一种较好的模型加速方案。（3）对于残差节点而言，需要当所有的残差分支都计算出对应的结果之后，才能获得最终的结果，这些残差分支的中间结果都会保存在设备的内存中，这样会对推理设备的内存具有较大的要求，来回的内存操作会降低整个网络的推理速度。而推理阶段首先在线下将模型转换为单分支结构，在设备推理阶段就能更好的提升设备的内存利用率，从而提升模型的推理速度。总而言之，模型推理阶段的网络结构越简单越能起到模型加速的效果。

模型推理阶段的重参数化过程



- 将残差块中的卷积层和BN层进行融合，该操作在很多深度学习框架的推理阶段都会执行
- 将融合后的卷积层转换为Conv3*3
- 合并残差分支中的Conv3*3

RepVGG算法实现步骤

- 获取并划分训练数据集，并对训练集执行数据增强操作
- 搭建RepVGG训练网络，训练分类网络，直到网络收敛为止
- 加载训练好的网络，对该网络执行重参数化操作

- 加载重参数化后的模型，执行模型推理

实验结果

Model	Top-1 acc	Speed	Params (M)	Theo FLOPs (B)	Wino MULs (B)
RepVGG-A0	72.41	3256	8.30	1.4	0.7
ResNet-18	71.16	2442	11.68	1.8	1.0
RepVGG-A1	74.46	2339	12.78	2.4	1.3
RepVGG-B0	75.14	1817	14.33	3.1	1.6
ResNet-34	74.17	1419	21.78	3.7	1.8
RepVGG-A2	76.48	1322	25.49	5.1	2.7
RepVGG-B1g4	77.58	868	36.12	7.3	3.9
EfficientNet-B0	75.11	829	5.26	0.4	-
RepVGG-B1g2	77.78	792	41.36	8.8	4.6
ResNet-50	76.31	719	25.53	3.9	2.8
RepVGG-B1	78.37	685	51.82	11.8	5.9
RegNetX-3.2GF	77.98	671	15.26	3.2	2.9
RepVGG-B2g4	78.50	581	55.77	11.3	6.0
ResNeXt-50	77.46	484	24.99	4.2	4.1
RepVGG-B2	78.78	460	80.31	18.4	9.1
ResNet-101	77.21	430	44.49	7.6	5.5
VGG-16	72.21	415	138.35	15.5	6.9
ResNet-152	77.78	297	60.11	11.3	8.1
ResNeXt-101	78.42	295	44.10	8.0	7.9

- **相同测试条件下，最小的模型RepVGG-A0与ResNet-18相比，各项指标都有显著的提升，RepVGG-A0网络不仅具有更少的参数量，更快的推理速度，而且获得了更高的分类精度**
- **与EfficientNet-B0相比，RepVGG-B1g4不仅具有更快的执行速度，而且获得了更高的分类精度，当然该模型也更大一些**
- **与VGG-16网络相比，RepVGG-B2在各个指标上面都有一定的性能提升**

Backbone	Mean IoU	Mean pixel acc	Speed
RepVGG-B1g2-fast	78.88	96.19	10.9
ResNet-50	77.17	95.99	10.4
RepVGG-B1g2	78.70	96.27	8.0
RepVGG-B2-fast	79.52	96.36	6.9
ResNet-101	78.51	96.30	6.7
RepVGG-B2	80.57	96.50	4.5

- 与**ResNet-50**网络相比，**RepVGG-B1g2-fast**网络不仅获得较高的精度，而且在速度上也有一些优势
- 与**ResNet-101**网络相比，**RepVGG-B2-fast**网络不均获得了较高的速度，各项指标上也都有所提升