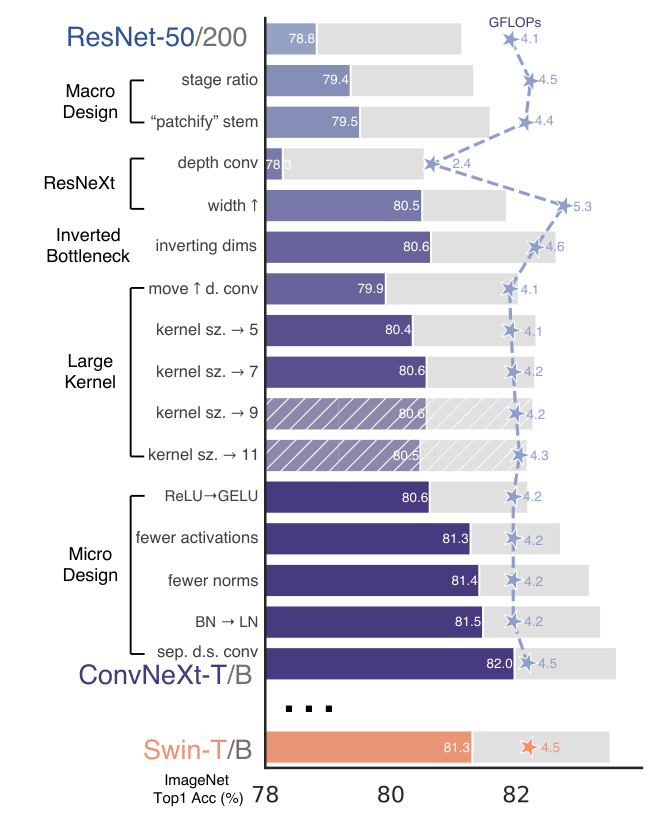
# 背景

目前很多视觉任务都使用基于Transformer的模型，而且一些榜上排名靠前的很多都是使用了Transformer。那么纯卷积网络就一定比使用了Transformer的网络性能差么？使用Vision Transformers相比CNN要更难训练。比如Transformer需要更多的训练数据，需要迭代更多的轮数，需要更多的数据增强，且对数据增强很敏感。而且算力要求太高。

这篇文章就提出了纯卷积神经网络ConvNeXt，模型是基于ResNet50/200模型，一开始使用的训练策略是基于ViT的，训练轮数从原来的90轮改为300轮，使用的优化器是AdamW，使用了Mixup，Cutmix，随机增强，随机擦除等数据增强手段，以及一些正则化方法，以这个作为实验基准，进行后续的改进。



上图展示了在相同的FLOPs下，ConvNeXt与Swin Transformer在ImageNet数据集上的对比，可以看出纯卷积网络的准确率已经超过了Swin Transformer。

# 方法

**①macro design**

(1)改变模块堆叠的比例。

从图1中红色框的ResNet50框架可以看出，原bottle neck-residual block的堆叠比例是[3, 4, 6, 3]。

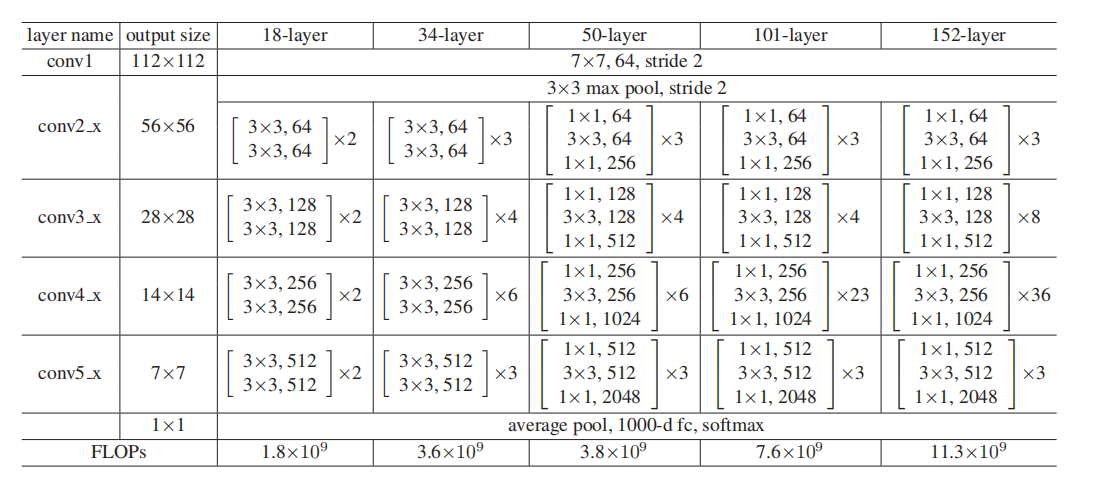


图 1 ResNet结构框架示意图

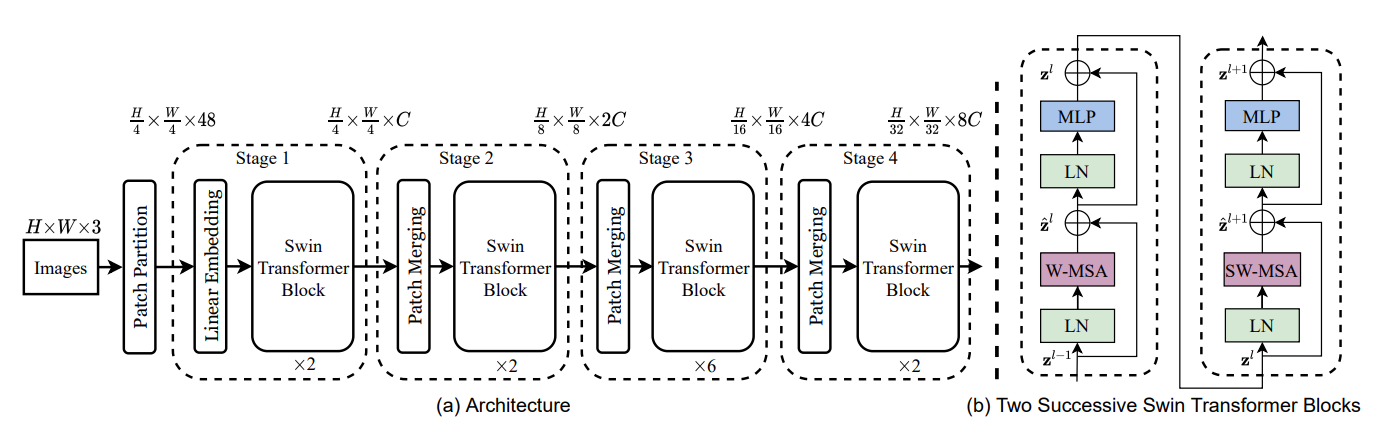
文章按照Swin Transformer中的block设计比例[1, 1, 3, 1]，将ResNet50的block堆叠比例进行了调整，调整为[3, 3, 9, 3]，准确率从78.8%提升到79.4%。

图 2 Swin Transformer

(2)改变下采样的方式

如图1蓝色方框中的所示，原ResNet50网络采用7×7，步长为2的卷积操作对图像进行下采样，但是在Swin Transformer中采用的是Patch Merging操作。

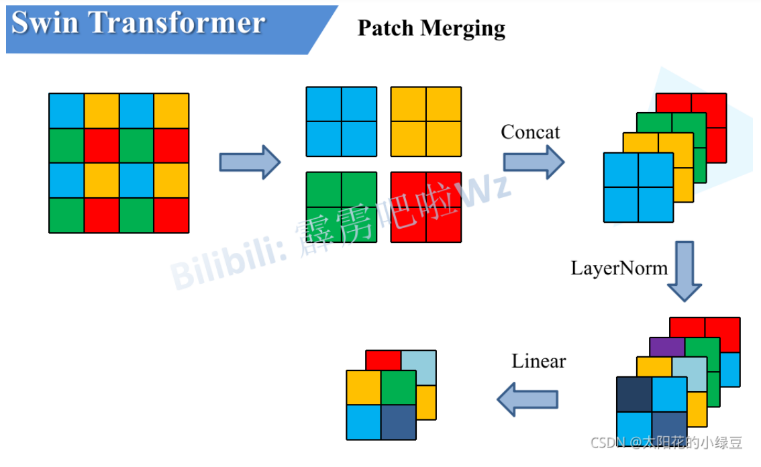


图 3 Patch Merging

假设输入特征图大小为4×4×1，Patch Merging会将每个2×2的相邻像素划分为一个patch，然后将每个patch中相同位置（同一颜色）像素给拼在一起就得到了4个特征图。接着将这四个特征图在深度方向进行concat拼接，然后在通过一个LayerNorm层。最后通过一个全连接层在特征图的深度方向做线性变化。

（原文链接：<https://blog.csdn.net/qq_37541097/article/details/121119988>）

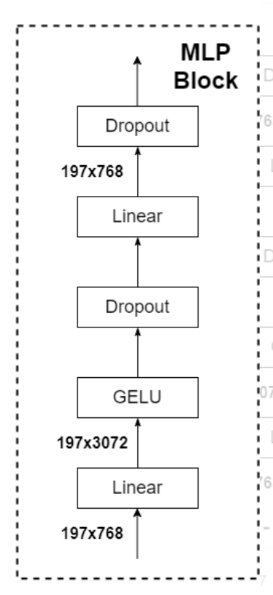
作者将原来的下采样层换为和Swin Transformer中一样的4×4，步长为4的patchify层。准确率有所提升。

**②ResNeXt风格**

ResNeXt网络每个模块使用组卷积，使用类似于Inception模块的设计，但不同的是每个路径使用相同的拓扑结构，作者采用的是深度可分离卷积，即卷积的组数和特征通道数相同。第一阶段最开始的通道数由原来的64改为和Swin-T一样的通道数96。准确率达到了80.5%

**③反向瓶颈结构**

图 4 Vision Transformers中的MLP结构示意图



从图4中可以看出Vision Transformers的MLP结构中间隐藏层部分的特征维度是输入图像的4倍，这样的结构和MobileNetV2里提出的反向残差结构类似。故作者也采用了反向瓶颈结构。

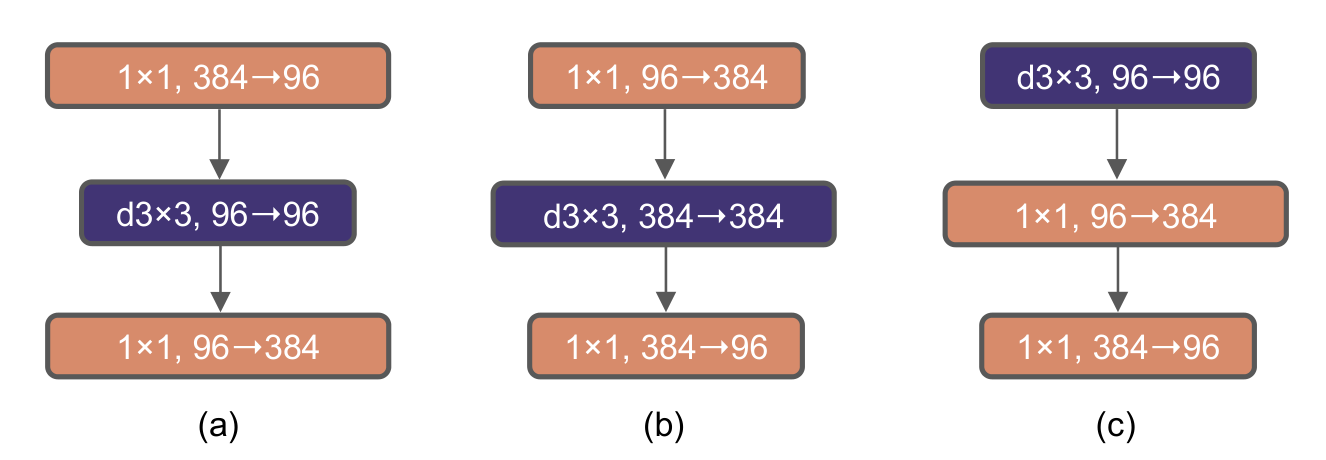


图 5 反向瓶颈结构

图5中(a)是ResNeXt使用的瓶颈结构，(b)是反向瓶颈结构，(c)是本文提出的反向瓶颈结构。模型准确率有所提升。

**④大卷积核**

在Transformer中一般都是对全局做self-attention，但现在主流的卷积神经网络都是采用3×3大小的卷积核，之前VGG中提出了“堆叠多个3×3卷积可以替代一个更大的卷积”，而且现在的GPU设备针对3×3大小的卷积核做了优化，所以运行更高效。

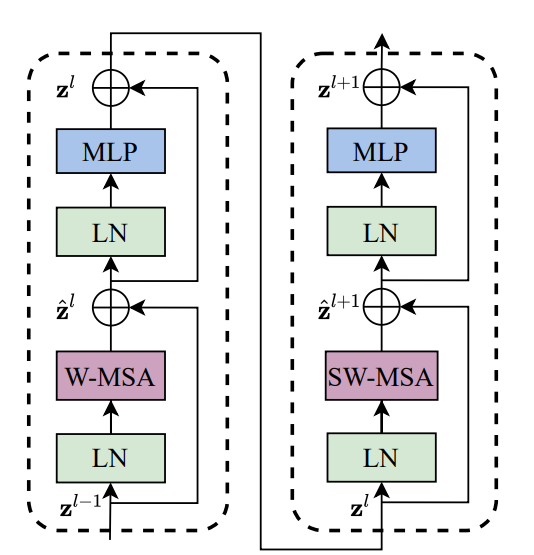


图 6 Swin Transformer Blocks

(1)调整深度可分离卷积的位置

参照Swin Transformer的结构，MSA这种复杂且低效的模块其输出通道数少，将其放在MLP模块之前。所以设计的反向瓶颈结构中，将深度可分离卷积的位置进行了迁移，从图5的(b)改为(c)，这样改动后，准确率和FLOPs都有所下降。

(2)增加卷积核尺寸

作者对深度可分离卷积的卷积核大小寸尺做了一系列选择，通过对比发现卷积核尺寸取到7时，准确率饱和。因此将原来3×3卷积改为7×7卷积。

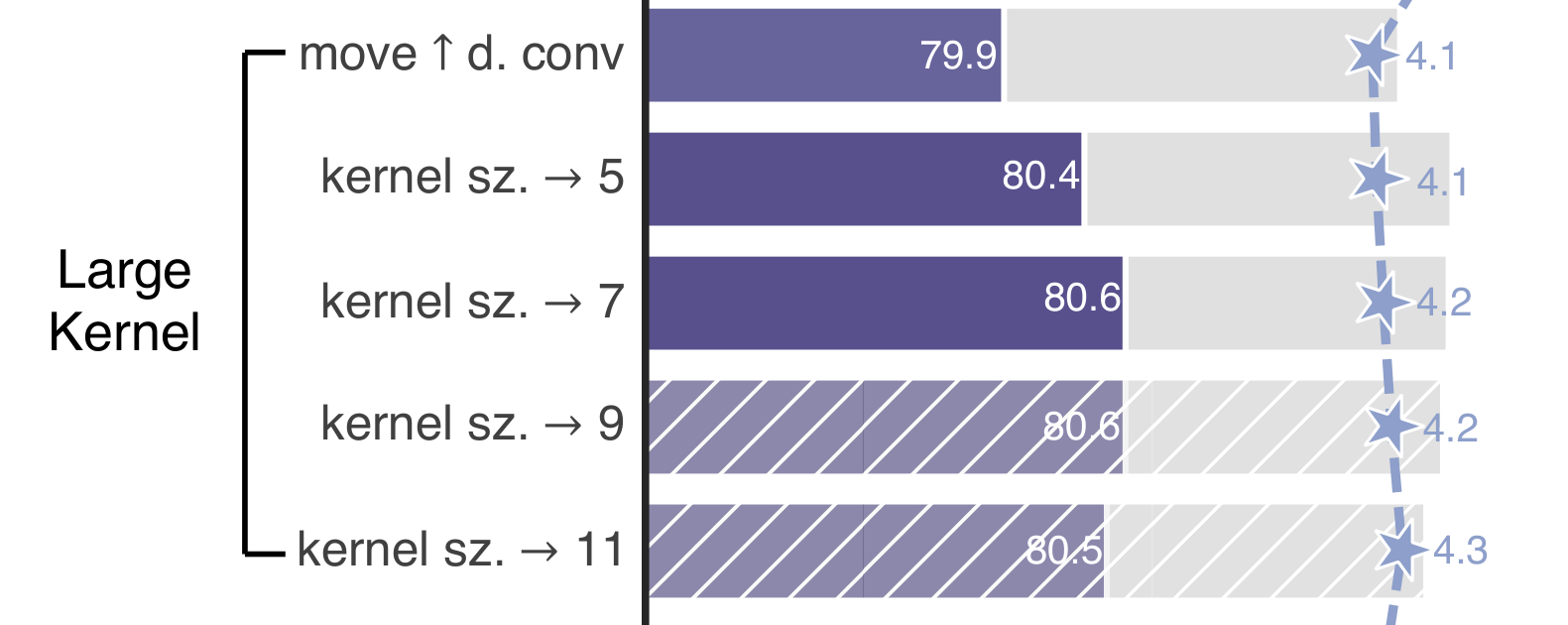


图 7 卷积核大小选择准确率对比

**⑤细节处的设计**

(1)激活函数的选择

通过和Transformer对比发现在Transformer中使用的激活函数都是GELU，因此将卷积神经网络中常用的ReLU激活函数替换为GELU激活函数，发现准确率并没有变化。

(2)激活函数个数的选择

在卷积神经网络中可以发现在卷积完后都会接一个激活函数，但是从图4的MLP结构中发现，只有在第一个全连接层后面使用了激活函数。因此作者也减少了激活函数的个数。发现采用这样的变动后，准确率有所提高。

(3)归一化层的个数选择

从Swin Transformer的结构中可以看出其归一化层的使用也很少，因此，作者也减少了归一化层的使用。此时准确率已经超过了Swin-T。

(4)将BN替换为Layer Normal

在Transformer中基本都用的Layer Normalization，因此作者也采用了同样的操作，准确率有提升。

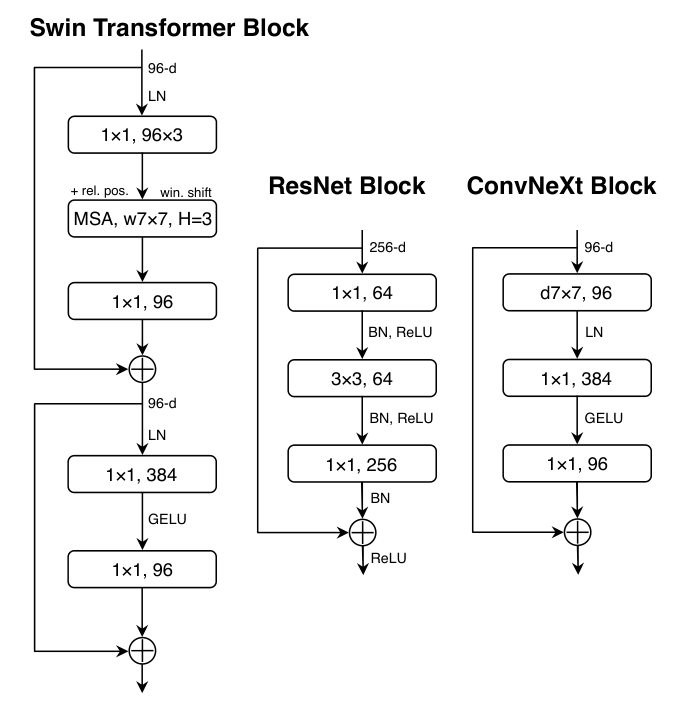


图 8 细节改变

(5)将下采样层分离出来

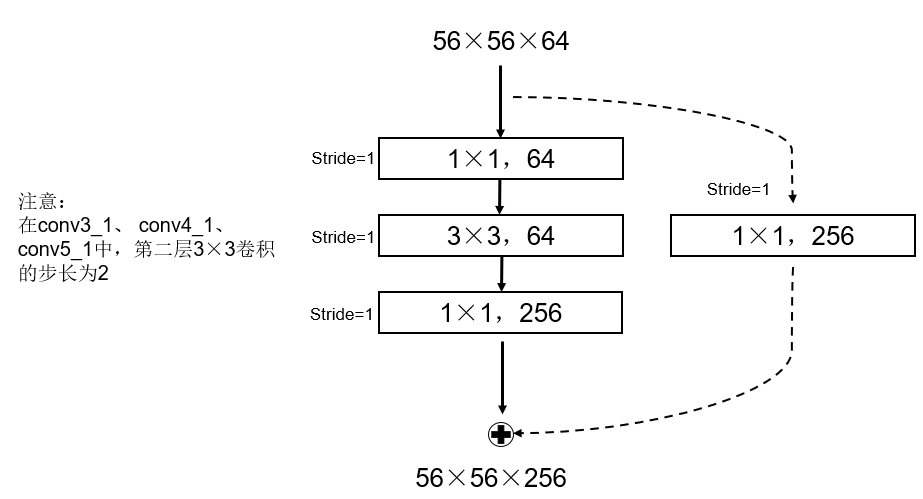


图 9 ResNet中的虚线残差连接

如图9所示，在ResNet50网络中conv3\_1、conv4\_1、conv5\_1的3×3卷积以及跳跃连接的步长都为2实现下采样。但是在Swin Transformer中下采样操作是通过单独的Patch Merging实现的，因此作者也为实现下采样单独设计了一个网络层结构，即一个Laryer Normalization加上一个卷积核大小为2步长为2的卷积。准确率就提升到了82.0%

最后，效仿Swin Transformer，也设计了四个版本，即T/S/B/L.

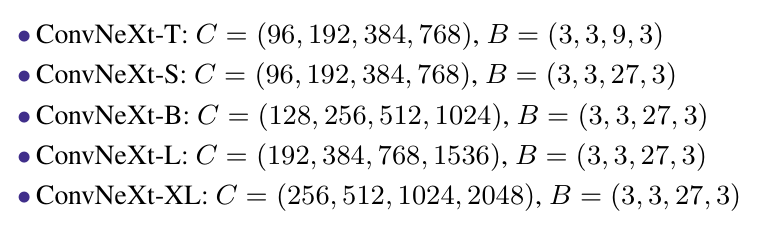


图 10 四种版本的参数配置