

背景

本文旨在 ImageNet1,000 分类任务上，对分辨率为 224×224 的图片进行分类，并将其计算量降至 6MFLOPs。但是仅一层输入特征图尺寸为 $112 \times 112 \times 3$ ，输出通道数为 8 的步长为 2 的卷积层所要求的计算量就有 2.7M Madds，那么用于设计其他卷积层和设计最终分类器的计算量就很有有限了。现有的高效的轻量网络都是减少网络的深度或者宽度来达到轻量化的目的，但是这样做会引起网络性能的下降，为了避免这样的情况发生，提出了两个低 FLOPs 网络的设计准则：

①降低节点之间的联结度从而避免降低网络的宽度；

②在每一层之间引入更多复杂的非线性激活函数来补偿网络深度减少引起的性能下降。

方法

降低节点之间的联结度从而避免降低网络的宽度

什么是节点之间的联结度？

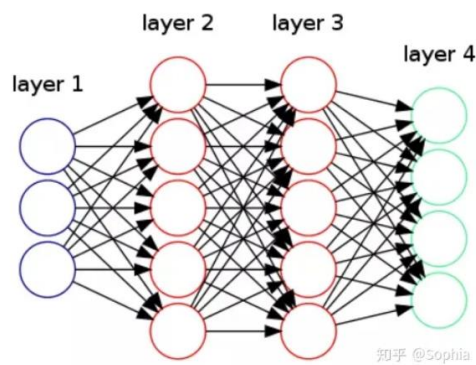


图 1

现在我们将卷积层想象成一个图，卷积层输入和输出之间的连接边其实就是卷积核的参数，由此将每个输出节点所连接的边的数量抽象地称为联结度。

现在我们假设该卷积层的计算量不变（假设输入特征图大小为 $H \times W \times C$ ，卷积核尺寸为 $K \times K$ ， c 个卷积核，输出特征图尺寸为 $h \times w \times c$ ）为：

$$K \times K \times C \times h \times w \times c = hwcCK^2 \quad (1)$$

那么假设输入特征图大小和卷积核大小不变，增大输出特征图通道数 $c \uparrow$ ，

为了保持计算量不变，只有减少输入输出节点之间的连接边数量。这样可以看出在固定的计算量下，节点之间的联结度和网络的宽度（特征通道数）之间是矛盾关系，那么平衡好两者之间的关系，有利于避免减少网络的宽度，提高每一层的特征表达能力。由此产生了第一条低 FLOPs 模型设计的准则：

降低节点之间的联结度从而避免降低网络的宽度。

实现的方法就是将逐点卷积和深度方向的卷积分解为低秩矩阵，分解为更为合适的尺度，用来平衡通道数目和输入输出联结度。

①将逐点卷积分解为自适应分组卷积。

自适应分组卷积的分组数和通道数构成算术平方根的关系。堆叠两个自适应分组卷积本质上是用一个分块矩阵近似一个点卷积操作。

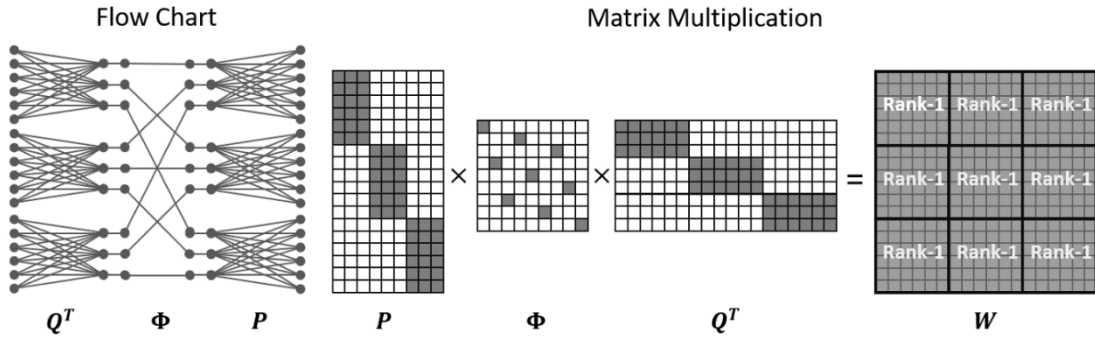


图 2

1×1 的逐点卷积带来的计算量太大，为了解决这样的情况，这里将 1×1 卷积使用的稠密的矩阵 w 分解为 3 个稀疏矩阵 Q 、 Φ 、 P ，且 $w = P\Phi Q^T$ ，以此来降低节点之间的联结度。其中矩阵 w 的尺寸是 $C \times C$ ；矩阵 Q 是对角块矩阵，其尺寸是 $C \times \frac{C}{R}$ ， R 是压缩比；矩阵 Φ 是置换矩阵，其尺寸是 $\frac{C}{R} \times \frac{C}{R}$ ；矩阵 P 也是对角块矩阵，其尺寸是 $C \times \frac{C}{R}$ 。

图 2 假设卷积矩阵 w 的尺为 18×18 ，压缩比 R 为 2，分组数 G 为 3（ G 为自适应分组数目 $G = \sqrt{\frac{C}{R}}$ ）。

首先根据压缩比 2 将 18 个节点映射为 9 个节点，这个过程分为 3 组卷积，因此每组卷积用到的矩阵大小为 6×3 ；

然后将得到的 9 个节点使用置换矩阵，进行 ΦQ^T 运算，得到的是 Q^T 矩阵所有的横行，相当于将 Q^T 矩阵重新排序；

最后将重新排序后的 9 个节点分 3 组，映射回 18 个节点，每组需要 3×6 大小的矩阵。

将 1×1 卷积使用的稠密的矩阵 w 分解为 3 个稀疏矩阵 Q 、 Φ 、 P ，相当于将原始 w 矩阵分成 $G \times G$ 个块，每个块的秩都为 1。从节点的联结度可以看出网络是十分稀疏的，每个输出节点只和 3 个隐藏层节点相连接，而如果使用原始逐点卷积的话，每个输出节点都要与 18 个输入节点相连接。

②将深度方向的卷积分解

将一个卷积核大小为 $k \times k$ 的深度方向的卷积直接分解为一个 $1 \times k$ 和一个 $k \times 1$ 的向量。其分解过程如图 3 所示。

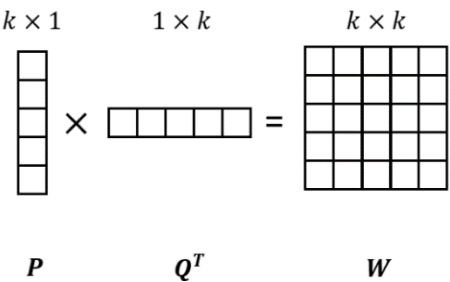


图 3

③将上述操作结合起来

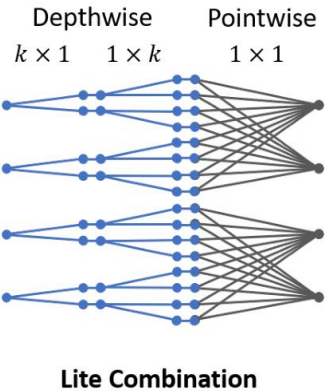


图 4

可以使用两种不同的方式将前述的两种操作结合起来。第一种是简单地将两

种卷积拼接在一起，第二种（如图 4 所示）是先用分解的深度方向卷积进行通道数扩增，再使用一组自适应分组卷积进行通道压缩，这样可以节省更多的计算资源用到学习更多的空间信息上，而非用于通道之间的融合上。

这样看来，网络的通道数量没有减少，计算量还降低了，达到了一开始的目的。

在每一层之间引入更多复杂的非线性激活函数来补偿网络深度减少引起的性能下降

之前改造过的逐点卷积只关注了每个分组卷积内部的连通性，忽略了组间的连通性。那么这里提出的一种 Dynamic Shift-Max 激活函数可以加强组间的信息流动。如图 5 所示，就是在原来的两组自适应分组卷积之间加入了一个静态的 Dynamic Shift-Max 激活函数——Group Shift 操作，它是 Dynamic Shift-Max 激活函数的特例。

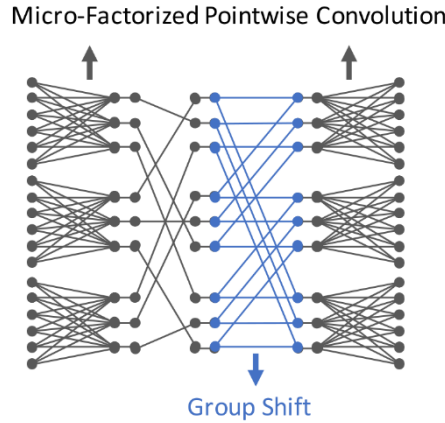


图 5

假设输入变量 $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_C\}$ ， \mathbf{x}_i 为输入的第 i 个通道的特征图，那么第 i 个通道在第 j 个组上的组循环偏移函数如下：

$$x_{\frac{C}{G}}(i, j) = x_{(i+j\frac{C}{G}) \bmod C}, \quad j = 0, \dots, G-1,$$

其中 C 为输入通道数， G 为分组数。

举个例子来解释一下上式。假设此时 $i=1, j=\{0, 1\}$ ，那么就意味着输入变量中的第 1 个通道的特征图要和第 4 个通道的特征图进行组间融合。 j 其实就是用来控制多少组之间的信息进行融合。

Dynamic Shift-Max 激活函数的定义如下：

$$y_i = \max_{1 \leq k \leq K} \left\{ \sum_{j=0}^{J-1} a_{i,j}^k(\mathbf{x}) x_{\frac{C}{G}}(i, j) \right\}$$

其中 $a_{i,j}^k(\mathbf{x})$ 是可学习参数，并且是依赖于输入样本 \mathbf{x} 的动态参数，类似于 SENet 中的动态通道加权。参数 $a_{i,j}^k(\mathbf{x})$ 可以简单地通过两个全连接层后加平均池化层实现。J 和 K 为人工设定的超参数，一般设置 J=K=2。Dynamic Shift-Max 激活函数的输出是第 i 个特征通道通过将 J 组 K 种不同方式融合后的最大响应值。最大限度地实现了输入特征映射与其组间通道偏移之间的多次融合，而且每次融合的参数都是动态的，这些特点为 Dynamic Shift-Max 提供了更多的特征表达能力，以补偿层数的减少。

MicroNet 模型

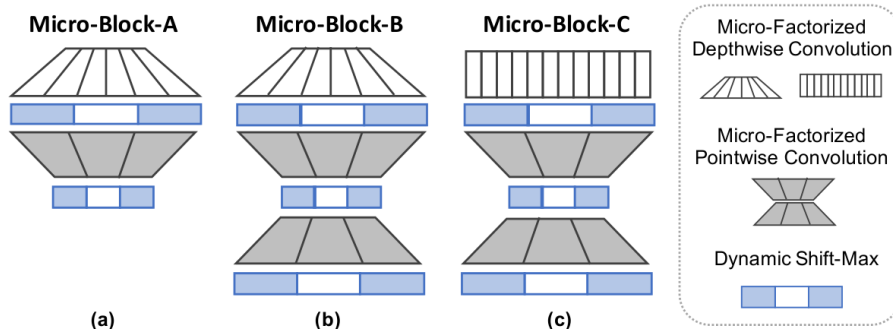


图 6

如上图所示是 3 种不同 MicroNet 模块，其计算量 FLOPs 从 6M 到 44M。它们都是由 Micro-Factorized Pointwise 卷积和 Micro-Factorized Depth-wise 卷积以不同的形式组合而来，并且它们都使用 Dynamic Shift-Max 作为激活函数。

图 6(a)首先是通过 Micro-Factorized Depth-wise 卷积扩展通道数，然后再使用一组自适应分组卷积进行通道压缩，还原通道数。图 6(b)和(a)的不同之处在于使用了两组自适应分组卷积。图 6(b)和(a) Micro-Factorized Pointwise 卷积和 Micro-Factorized Depth-wise 卷积之间的结合方式是使用图 4 的方法，而图 6(c)则是使用普通的拼接方法，一般该模块在网络较深的层中使用。

总结

个人感觉 Micro-Factorized Pointwise 部分和 ShuffleNetV1 使用的通道重排的想法很类似，通过分解卷积核，减少网络的联结度，使得网络更加稀疏，提高计算性能，又设计了动态的激活函数，增加网络的非线性，提高特征表达能力。