# 背景

通常我们去评价一个模型时，首先看的应该是它的精确度，当精确度达不到要求的时候，模型预测的速度再快，在后续模型部署的时候内存占用量再小，都无济于事。但如果模型达到一定的精确要求时，就需要更进一步的评价指标来评价模型，例如，前向传播时所需的计算力，它反应了对硬件如GPU性能要求的高低；参数量大小，它反应所占内存大小。

现在很多网络模型的设计都是以FLOPs为指标，浮点运算量是实际运算过程中的加减乘除计算过程中的计算次数，描述计算力。但这是一种非直接的度量网络计算复杂度的方式，还有许多直接度量的方式，例如模型的速度，但是速度又与其他因素有很大的关系，比如内存访问成本、并行度、硬件条件，FLOPs没有考虑这几个对速度有相当大影响的重要因素。

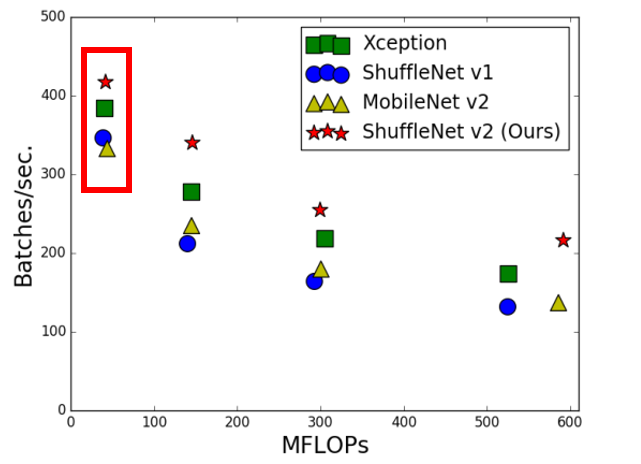


图 1

内存访问成本（MAC）是指计算机在进行计算时候要加载到缓存中，然后再计算，这个加载过程是需要时间的。其中，分组卷积是对MAC消耗比较多的操作（例如AlexNet将卷积分到多个GPU上）；在相同的FLOPs下，具有高并行度的模型可能比具有低并行度的另一个模型快得多，如果网络的并行度较高，那么速度就会有显著的提升；有的硬件平台会对卷积操作进行优化，比如：cudnn加强了对3×3卷积计算的优化，这样一来，不同平台下的FLOPs指标大小没有什么可比性。通过图1的红色框中可以看出，即使是有相同的FLOPs，在速度上也是有差异的，因此选用FLOPs作为计算复杂度的评价指标是非常片面的。

除此之外，通过将几个网络的运行时间拆分，可以看出卷积操作占据了大部分时间，而FLOPs指标只考虑了卷积部分的运行时间，其他的操作，例如数据I/O操作、数据打乱操作、逐元素操作等，也会占用大量的时间，因此从这个角度看FLOPs指标并不能够反应实际的运行时间，很片面。

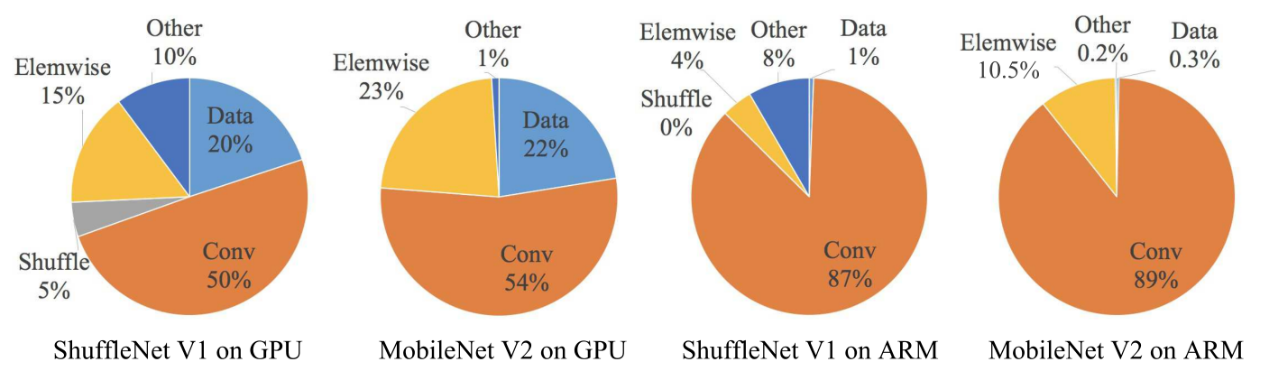


图 2

本文通过从不同的角度分析网络的运行时间，得出了几条高效网络的设计指南，与此同时也提出了ShuffleNetV2网络模型。

# 方法

## 高效的模型设计指南

### ①输入输出通道数相同时，内存访问成本最小

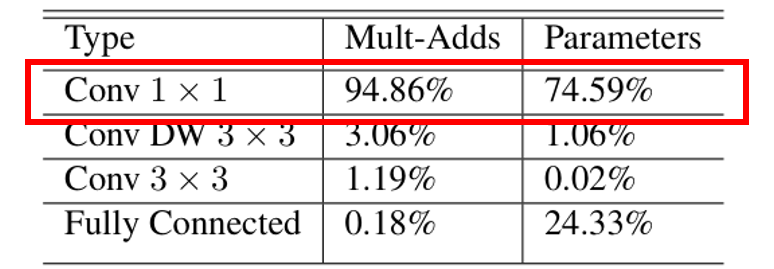


图 3

图3是对MobileNetV1各个操作所占用的计算量和参数量的分析，可以看出1×1卷积占总计算量的94.86%，占总参数量的74.59%，可见1×1卷积导致计算量很大。

ShuffleNetV1是提出了逐点分组卷积来解决上述问题，在本文中是通过分析1×1卷积的输入输出通道数，来设计使得MAC最小的卷积核形状。

🎈从数学角度证明一下：

以1×1卷积为例，假设输入通道数为，输出通道数为，输出特征图的大小为，那么该操作的计算量为：

 (1)

内存访问成本为：

输入部分：，输出部分：，权重：，所以MAC为三部分之和：，根据基本不等式“两个正实数的算术平均数大于或等于它们的几何平均数”，可以得出：

 (2)

设，那么，则代入(2)式中可得：

 (3)

可以看出当且仅当时等号成立，因此当输入通道数等于输出通道数时，MAC最小。

🎈实验证明一下：

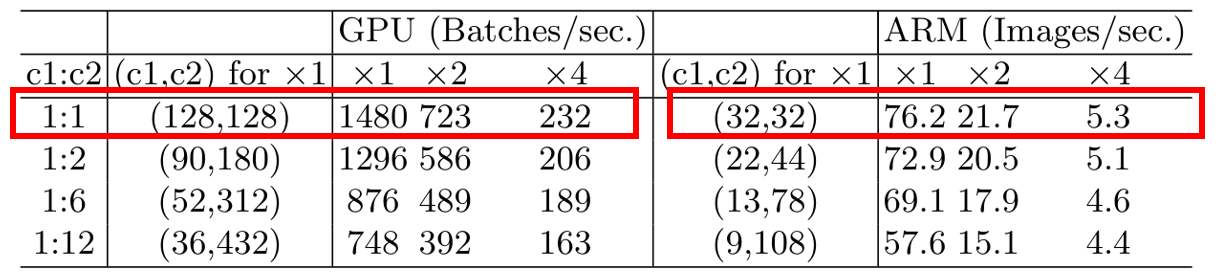


图 4

图4是在FLOPs相等的前提下进行的一系列对照实验，可以看出当输入输出通道数相等时，网络运行的最快。

### ②过多的分组会增加内存访问成本

使用分组卷积，当FLOPs固定的前提下，可以使用更多的特征通道，从而提高精度，但是更多的特征通道就意味着MAC的增加。

依然是从两方面进行证明。

🎈从数学角度证明一下：

在ShuffleNetV1中已经分析过分组卷积和普通卷积在计算量上的差异，可以得出一个一般性的结论，就是分组卷积的计算量是普通卷积计算量的，其中为分组的组数，那么(1)式就改写为：

 (4)

MAC为：

 (5)

设，那么，，则代入(5)式中可得：

 (6)

从(6)式中可以看出，MAC与分组数g成正比。

🎈实验证明一下：

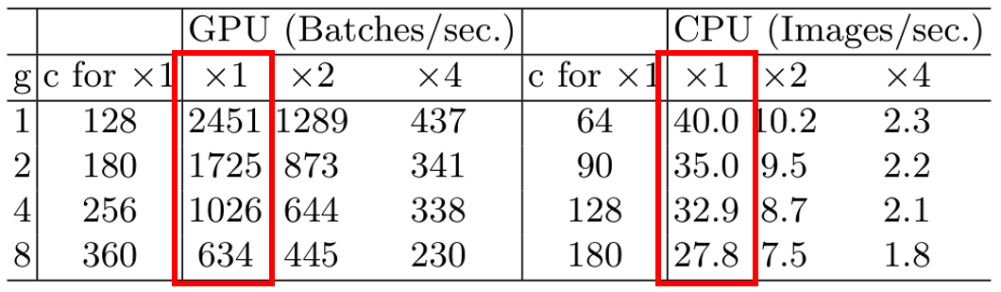


图 5

图5是通过改变通道数使得所有实验是在相同FLOPs下进行的，可以看出分组的组数越多，网络的运行速度就越慢。

### ③模型中的一些碎片化操作不利于网络并行化

这里的碎片化操作指的是在模型搭建时，每个block里面的卷积操作次数或者是池化操作的次数。虽然这些碎片化操作有利于精度的提升，但是不利于像GPU这样的有很强并行处理能力的硬件设备，因此导致了在GPU设备上效率不高。

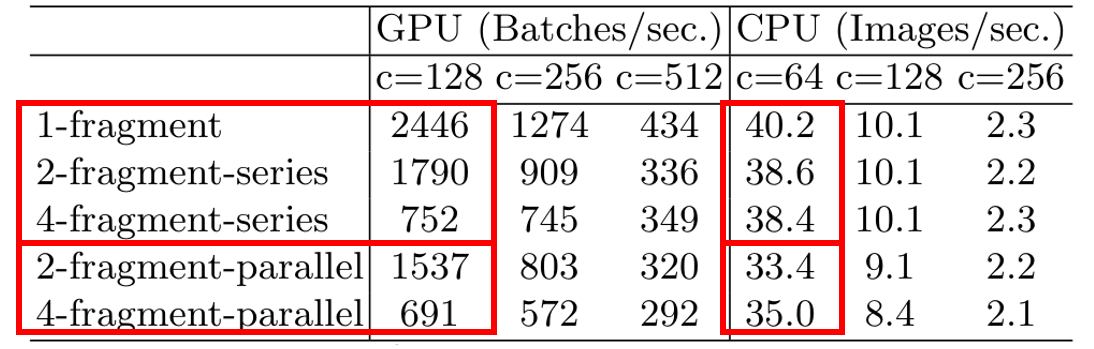


图 6

图6是在相同FLOPs下进行的对比实验，可以看出在GPU设备上随着block中串行的分支或是并行的分支数增加，网络运行速度会大幅下降，但是，同样的情况在CPU上网络的运行速度影响并不是很大。

### ④逐元素操作对FLOPs和MAC的影响是不可忽略的

这里的逐元素操作指的是ReLU、张量之间的相加操作、深度方向的卷积操作这类对FLOPs影响不大，但是对MAC影响很大的操作都称为逐元素操作。

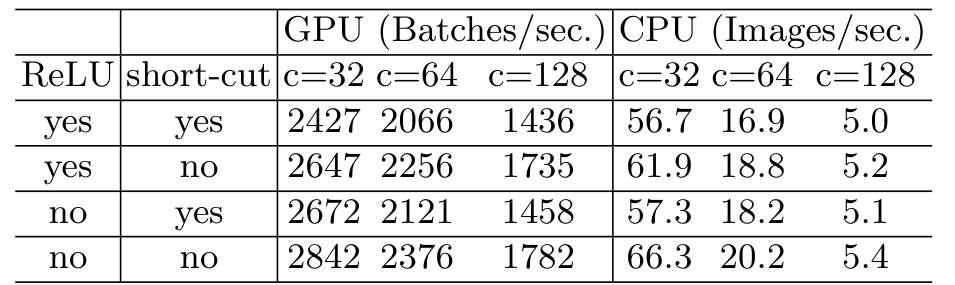


图 7

图7是通过将ResNet中瓶颈结构做了一系列对照实验，说明逐元素操作对网络速度有不可忽视的影响。

## ShuffleNetV2的单元结构

遵循以上四条高效网络的设计指南，一个高效的网络模型应该是：

①使用输入输出通道数相同的卷积操作；

②不过度依赖分组卷积操作来减少网络的计算量；

③网络的碎片化程度低；

④逐元素操作较少。

那么现在以这四个条件为准则，反观ShuffleNetV1（结构如图8a、b所示），其为了在不增加FLOPs的前提下增加特征通道数，使用了分组卷积以及瓶颈结构，现在看来这样的设计不符合条件1和2。

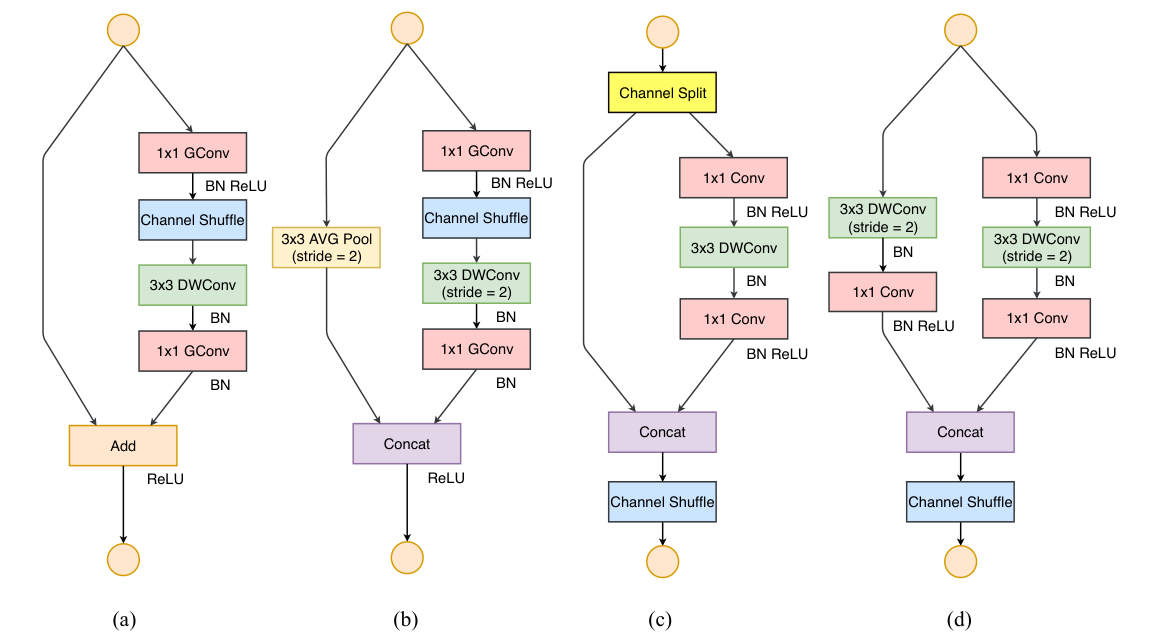


图 8

因此为了得到高效的模型，设计的关键就在于如何在保持特征通道数足够多的前提下既不增加block中卷积的次数也不增加分组的数量。在这样的准则下设计了ShuffleNetV2，其结构如图8(c)、(d)所示。

图8(c)中，首先是一个通道分离操作，在遵循设计指南3的前提下，将输入的通道数c一分为二。这样一分为二可以起到特征复用的效果。

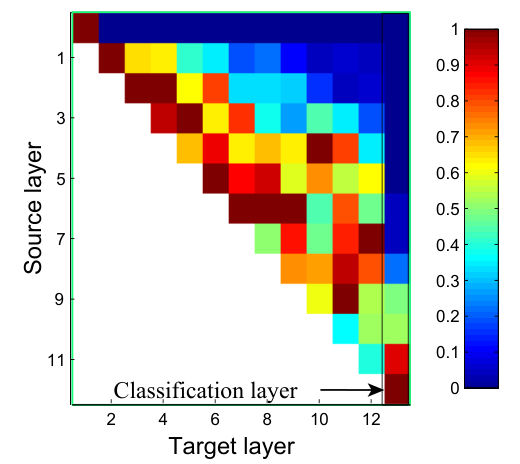


图 9

从图9可以看出相邻两层之间的关联性更强，这就意味着将所有层之间进行密集的连接是冗余的。

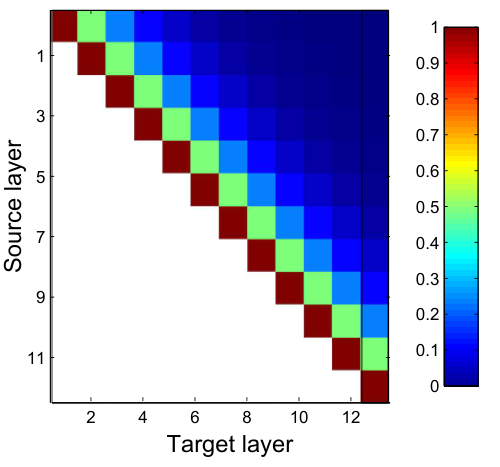


图 10

从图10可以看出block之间的间隔越多，其特征的相关性就越弱。

再看接下来的右侧分支，右侧分支是3个卷积操作，而且这3个卷积都是输入输出通道数相等的，也没有使用分组操作，因此符合指南1和2。最后将左侧的恒等映射和右侧分支进行concat拼接，没有使用逐元素相加操作，因此符合指南4。两部分内容拼接后进行了通道重排操作。

图8(d)是ShuffleNet2的下采样模块，和图8(c)不同的是将深度方向的卷积步长调整为2，并删除了一开始的通道分离操作，这样做的目的就是下采样，使特征图尺寸减半，通道数翻倍。

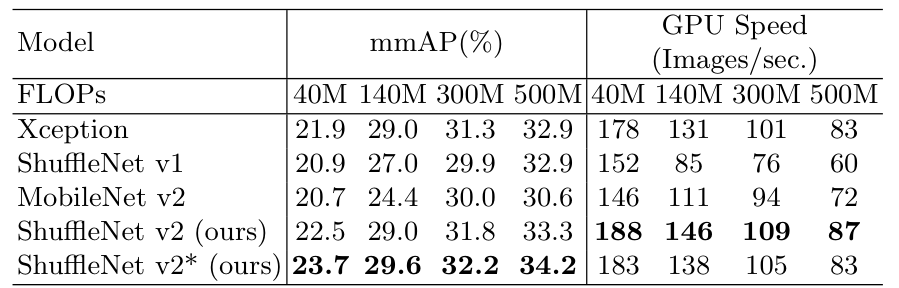


图 11

图11是在COCO目标检测任务上轻量级网络进行的对比，发现ShuffleNetV2网络更胜一筹。其中ShuffleNetV2\*是在每个block的第一个1×1卷积之前使用了额外的3×3深度方向的卷积，以此来增加感受野大小。