目录结构

一，模型计算量分析

> 终端设备上运行深度学习算法需要考虑内存和算力的需求，因此需要进行模型复杂度分析，涉及到模型计算量（时间/计算复杂度）和模型参数量（空间复杂度）分析。

为了分析模型计算复杂度，一个广泛采用的度量方式是模型推断时浮点运算的次数 （`FLOPs`），即模型理论计算量，但是，它是一个间接的度量，是对我们真正关心的直接度量比如速度或者时延的一种近似估计。

本文的卷积核尺寸假设为为一般情况，即正方形，长宽相等都为 `K`。

- `FLOPs`：floating point operations 指的是浮点运算次数,**理解为计算量**，可以用来衡量算法/模型时间的复杂度。
- `FLOPS`：（全部大写）,Floating-point Operations Per Second，每秒所执行的浮点运算次数，理解为计算速度,是一个衡量硬件性能/模型速度的指标。
- `MACCs`：multiply-accumulate operations，乘-加操作次数，`MACCs` 大约是 FLOPs 的一半。将 $w[0]*x[0]+...$ 视为一个乘法累加或 `1` 个 `MACC`。

注意相同 `FLOPs` 的两个模型其运行速度是会相差很多的，因为影响模型运行速度的两个重要因素只通过 `FLOPs` 是考虑不到的，比如 `MAC`（`Memory Access Cost`）和网络并行度；二是具有相同 `FLOPs` 的模型在不同的平台上可能运行速度不一样。

> 注意，网上很多文章将 MACCs 与 MACC 概念搞混，我猜测可能是机器翻译英文文章不准确的缘故，可以参考此链接了解更多。需要指出的是，现有很多硬件都将**乘加运算作为一个单独的指令。**

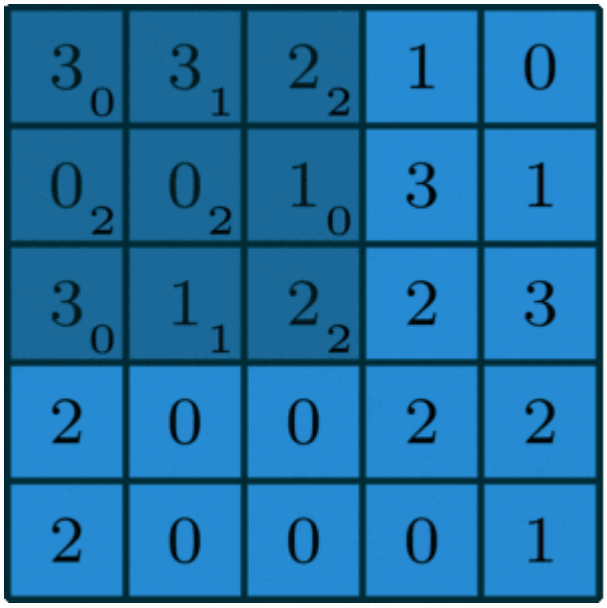
卷积层FLOPs计算

> 卷积操作本质上是个线性运算，假设卷积核大小相等且为 K。这里给出的公式写法是为了方便理解，大多数时候为了方便记忆，会写成比如 $MACCs=H\times W\times K^2\times C_i\times C_o$。

- $FLOPs=(2\times C_i\times K^2-1)\times H\times W\times C_o$（不考虑bias）
- $FLOPs=(2\times C_i\times K^2)\times H\times W\times C_o$（考虑bias）
- $MACCs=(C_i\times K^2)\times H\times W\times C_o$（考虑bias）

C_i 为输入特征图通道数，K 为过卷积核尺寸，H,W,C_o 为输出特征图的高，宽和通道数。 二维卷积过程 如下图所示：

> 二维卷积是一个相当简单的操作：从卷积核开始，这是一个小的权值矩阵。这个卷积核在 2 维输入数据上「滑动」，对当前输入的部分元素进行矩阵乘法，然后将结果汇为单个输出像素。



公式解释，参考这里，如下：

理解 `FLOPs` 的计算公式分两步。括号内是第一步，计算出`output feature map`的一个`pixel`，然后再乘以$H \times W \times C_o$，从而拓展到整个 output feature map。括号内的部分又可以分为两步：$(2 \times C_i \times K^2 - 1) = (C_i \times K^2) + (C_i \times K^2 - 1)$。第一项是乘法运算次数，第二项是加法运算次数，因为n个数相加，要加$n-1$次，所以不考虑`bias`的情况下，会有一个 -1，如果考虑`bias`，刚好中和掉，括号内变为$(2 \times C_i \times K^2)$。

所以卷积层的$FLOPs = (2 \times C_i \times K^2 - 1) \times H \times W \times C_o$ (C_i 为输入特征图通道数，K 为过滤器尺寸，H, W, C_o为输出特征图的高，宽和通道数)。

全连接层的 FLOPs 计算

全连接层的$FLOPs = (2I - 1)O$，I 是输入层的维度，O 是输出层的维度。

二，模型参数量分析

> 模型参数数量（params）：指模型含有多少参数，直接决定模型的大小，也影响推断时对内存的占用量，单位通常为`M`，`GPU`端通常参数用`float32`表示，所以模型大小是参数数量的`4`倍。这里考虑的卷积核长宽是相同的一般情况，都为`K`。

模型参数量的分析是为了了解内存占用情况，内存带宽其实比`FLOPs`更重要。目前的计算机结构下，单次内存访问比单次运算慢得多的多。对每一层网络，端侧设备需要：

- 从主内存中读取输入向量 / `feature map`；
- 从主内存中读取权重并计算点积；
- 将输出向量或`feature map`写回主内存。

`MAes`：memory accesse，内存访问次数。

卷积层参数量

卷积层权重参数量 = $C_i \times K^2 \times C_o + C_o$。

C_i 为输入特征图通道数，K 为过滤器(卷积核)尺寸，C_o 为输出的特征图的`channel`数(也是`filter`的数量)，算式第二项是偏置项的参数量 。(一般不写偏置项，偏置项对总参数量的数量级的影响可以忽略不记，这里为了准确起见，把偏置项的参数量也考虑进来。)

假设输入层矩阵维度是 96×96×3，第一层卷积层使用尺寸为 5×5、深度为 16 的过滤器（卷积核尺寸为 5×5、卷积核数量为 16），那么这层卷积层的参数个数为 5 ×5×3×16+16=1216个。

BN层参数量

`BN` **层参数量** = $2 \times C_i$。

其中 C_i 为输入的 `channel` 数（BN层有两个需要学习的参数，平移因子和缩放因子）

全连接层参数量

全连接层参数量 = $T_i \times T_o + T_O$。

T_i 为输入向量的长度， T_o 为输出向量的长度，公式的第二项为偏置项参数量。(目前全连接层已经逐渐被 `Global Average Pooling` 层取代了。) 注意，全连接层的权重参数量（内存占用）远远大于卷积层。

三，一些概念

双精度、单精度和半精度

`CPU/GPU` 的浮点计算能力得区分不同精度的浮点数，分为双精度 `FP64`、单精度 `FP32` 和半精度 `FP16`。因为采用不同位数的浮点数的表达精度不一样，所以造成的计算误差也不一样，对于需要处理的数字范围大而且需要精确计算的科学计算来说，就要求采用双精度浮点数，而对于常见的多媒体和图形处理计算， `32` 位的单精度浮点计算已经足够了，对于要求精度更低的机器学习等一些应用来说，半精度 `16` 位浮点数就可以甚至 `8` 位浮点数就已经够用了。
对于浮点计算来说， `CPU` 可以同时支持不同精度的浮点运算，但在 `GPU` 里针对单精度和双精度就需要各自独立的计算单元。

浮点计算能力

`FLOPS`：每秒浮点运算次数，每秒所执行的浮点运算次数，浮点运算包括了所有涉及小数的运算，比整数运算更费时间。下面几个是表示浮点运算能力的单位。我们一般常用 `TFLOPS(Tops)` 作为衡量 `NPU/GPU` 性能/算力的指标，比如海思 `3519AV100` 芯片的算力为 `1.7Tops` 神经网络运算性能。

- `MFLOPS`（megaFLOPS）：等于每秒一佰万（=10^6）次的浮点运算。
- `GFLOPS`（gigaFLOPS）：等于每秒拾亿（=10^9）次的浮点运算。
- `TFLOPS`（teraFLOPS）：等于每秒万亿（=10^12）次的浮点运算。
- `PFLOPS`（petaFLOPS）：等于每秒千万亿（=10^15）次的浮点运算。
- `EFLOPS`（exaFLOPS）：等于每秒百亿亿（=10^18）次的浮点运算。

硬件利用率(Utilization)

在这种情况下，利用率（Utilization）是可以有效地用于实际工作负载的芯片的原始计算能力的百分比。深度学习和神经网络使用相对数量较少的计算原语（computational primitives），而这些数量很少的计算原语却占用了大部分计算时间。矩阵乘法（MM）和转置是基本操作。MM 由乘法累加（MAC）操作组成。OPs/s（每秒完成操作的数量）指标通过每秒可以完成多少个 MAC（每次乘法和累加各被认为是 1 个 operation，因此 MAC 实际上是 2 个 OP）得到。所以我们可以将利用率定义为实际使用的运算能力和原始运算能力的比值：

$$utilization = \frac{used\ OPs/s}{raw\ OPs/s}$$

$$mac\ utilization = \frac{used\ Ops/s}{raw\ OPs/s} = \frac{FLOPs/time(s)}{Raw\_FLOPs} (Raw\_FLOPs = 1.7T\ at\ 3519)$$

四，参考资料

- PRUNING CONVOLUTIONAL NEURAL NETWORKS FOR RESOURCE EFFICIENT INFERENCE
- 神经网络参数量的计算：以UNet为例
- How fast is my model?
- MobileNetV1 & MobileNetV2 简介
- 双精度，单精度和半精度
- AI硬件的Computational Capacity详解