Report of latest found

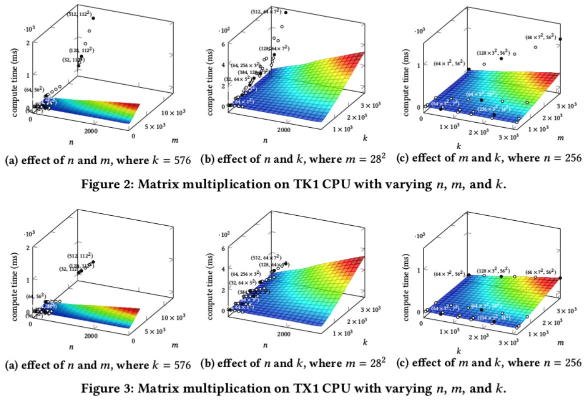
Ruzhuo Wang
rwang085@ucr.edu

1. What the paper do

1.1. Average forward time estimation of convolutional layer

Given the matmul of $[n \times k]$ and $[k \times m]$ (the number of FLOPs is $n \times m \times k$) performed by a CONV layer, n is the number of kernels, k is the size of a kernel in 3D ($width \times height \times depth$, where depth is the number of input feature maps), and m is the spatial size ($width \times height$) of output feature maps.

I find that the average forward time is linearly related to FLOPs ($n \times m \times k$)



(a) effect of n and m, where $k = 576$ (b) effect of n and k, where $m = 28^2$ (c) effect of m and k, where $n = 256$

Figure 2: Matrix multiplication on TK1 CPU with varying n, m, and k.

(a) effect of n and m, where $k = 576$ (b) effect of n and k, where $m = 28^2$ (c) effect of m and k, where $n = 256$

Figure 3: Matrix multiplication on TX1 CPU with varying n, m, and k.

Figure 1. The data they provide

1.2. Memory

The memory requirement to run a CNN comes from three major sources: (i) the memory that holds the parameters of the CNN; (ii) the memory that stores intermediate data of the CNN; and (iii) the workspace for computation.

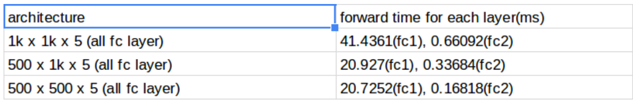
2. What I do

2.1. Average forward time estimation

2.1.1 For fully connected layer

The average forward time for a fc(fully connected) layer is determined by the number of nodes for adjacent layers, $e.g.$ one fc layer has m nodes, the adjacent fc layer has n nodes, then the average forward time is linearly related to $m \times n$.

In Figure 2, '1k x 1k x 5' means fc1 layer has 1000 nodes, the adjacent layer for fc1 is fc2 and fc2 has 1000 nodes, the adjacent layer for fc2 is fc3 and fc3 has 5 nodes.



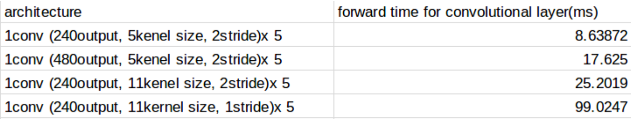
architecture	forward time for each layer(ms)
1k x 1k x 5 (all fc layer)	41.4361(fc1), 0.66092(fc2)
500 x 1k x 5 (all fc layer)	20.927(fc1), 0.33684(fc2)
500 x 500 x 5 (all fc layer)	20.7252(fc1), 0.16818(fc2)

Figure 2. The raw data I have for fc layer forward time

2.1.2 For convolutional layer

Assume the output number is x, kernel size is $y_1 \times y_2$, stride is z, the input batch size is k, input height is m, input width is n, input channel is c, then the average forward time should be linearly related to:

$$k \times x \times c \times y_1 \times ceil(\frac{m \times floor(\frac{y_1}{2}) \times 2}{z}) \times y_2 \times ceil(\frac{n \times floor(\frac{y_2}{2}) \times 2}{z})$$

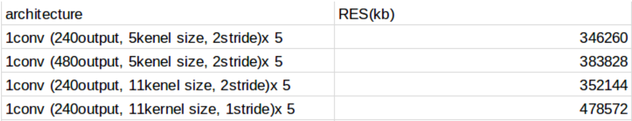


architecture	forward time for convolutional layer(ms)
1conv (240output, 5kenel size, 2stride)x 5	8.63872
1conv (480output, 5kenel size, 2stride)x 5	17.625
1conv (240output, 11kenel size, 2stride)x 5	25.2019
1conv (240output, 11kernel size, 1stride)x 5	99.0247

Figure 3. The raw data I have for convolutional layer forward time

2.2. Memory

I only have the raw data for memory estimation.



architecture	RES(kb)
1conv (240output, 5kenel size, 2stride)x 5	346260
1conv (480output, 5kenel size, 2stride)x 5	383828
1conv (240output, 11kenel size, 2stride)x 5	352144
1conv (240output, 11kernel size, 1stride)x 5	478572

Figure 4. The raw data I have for convolutional layer memory