SysDL Assignment 2

Ajay S Joshi(CS15B047)
SysDL CS6886
IITM

March 26, 2020

1 Machine Configuration

1. L1D Cache size: 32KB
2. L2 Cache size: 256KB
3. L3 Cache size: 3MB
4. RAM: 4GB

2 AVX Intrinsics

$\_m256\_add\_ps$, $\_m256\_mul\_ps$: Add and multiply 2 256 bit registers
$\_m256_{l}oadu\_ps$, $\_m256\_storeu\_ps$, $\_m256\_maskload\_ps$, $\_m256\_maskstore\_ps$: Unmasked/-Masked Load/store from/to memory
$\_m256\_setzero\_ps$: Set 256-bit register to all zeros
$\_m256_{c}mp\_ps$, $\_m256\_and\_ps$: Compare 2 registers based on some comparison operator and do a bitwise AND of 2 registers
$\_mm256\_extractf128\_ps$: Extract higher/lower 128 bits from 256-bit register
$\_mm\_permute\_ps$: Permute 32-bit data from the source to create result

3 Step 1: Correctness

Total execution time of optimal model for batch size = 1:

- Without compiler optimization: 6.90 sec
- With -O3 optimization in g++: 1.21 sec

99 % of the total time is taken by convolution operations in 5 layers. Execution time varies across machines, which is demonstrated by lower execution time in a more powerful machine.

4 Step 2: Bottleneck analysis

1. Execution times are proportional to the number of computations in each CONV or FC layer.

2. If all memory accesses are assumed to be from the main memory, then 4 memory operations are performed (accumulation = read + write in output, read input, read weights), each of 4 bytes for 1 operation(MAC).
   Operational intensity = $\frac{1}{16}$ MAC/byte.

4.1 CONV Layers

1. CONV 1: 0.13 sec, 1.09 x 10^8 computations
2. CONV 2: 0.39 sec, 4.48 x 10^8 computations
3. CONV 3: 0.16 sec, 1.50 x 10^8 computations
4. CONV 4: 0.24 sec, 2.24 x 10^8 computations
5. CONV 5: 0.16 sec, 1.50 x 10^8 computations

4.2 FC Layers

1. FC 1: 0.014 sec, 3.78 x 10^7 computations
2. FC 2: 0.006 sec, 1.68 x 10^7 computations
3. FC 3: 0.003 sec, 4.1 x 10^6 computations

Highest time is taken by CONV layer 2. Also, other computations such as input padding, ReLU activation, max-pooling take orders of magnitude lesser time than the CONV layer computations, which take the most time.

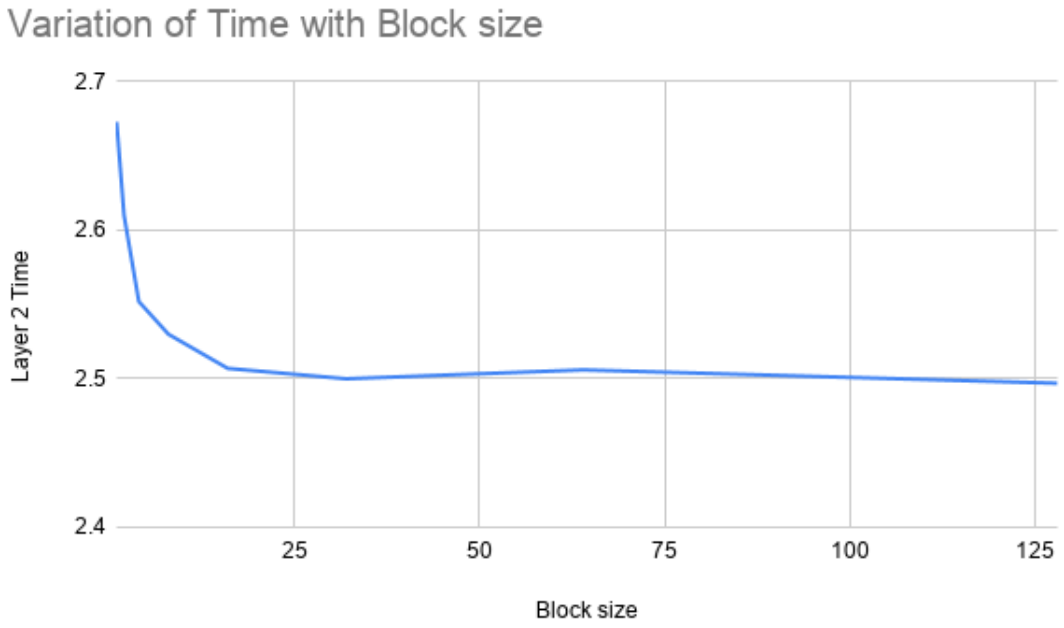
5 Design Space Exploration

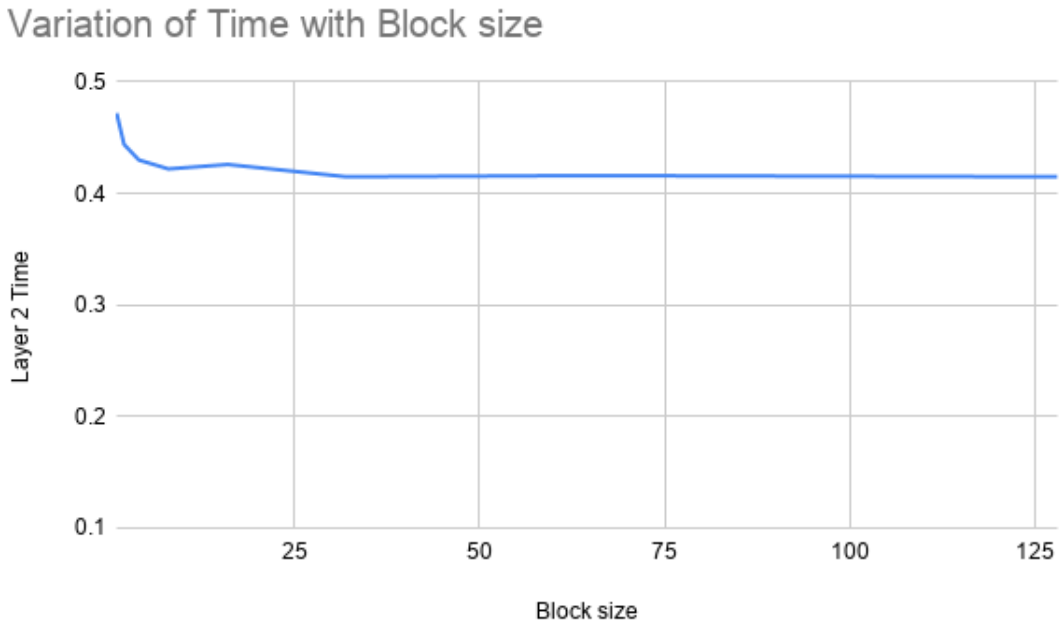
Variant Layer	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Input Stationary	0.31	1.002	0.542	0.810	0.454
Weight Stationary	**0.102**	**0.389**	0.206	**0.308**	**0.206**
Output Stationary	0.337	0.587	**0.165**	0.372	0.245

The best reuse patterns are highlighted in the above table for each layer. For most of the layers, weight stationary performs the best by good margins.

6 Tiling

The below graph shows variation in time with block size in tiling. The experiments are performed without and with using -O3 optimization in compiler respectively.





For both cases, tile size of **32** gives the best performance. The performance improvement is more without -O3 optimizations, as -O3 includes loop tiling along with lot of other optimizations.