CS5222 Advanced Architectures

Project 2 Part 2 Report

FPGA Lab

Nguyen Dang Phuc Nhat (A0184583U)

Table of Contents

1 Fixed-point Validation

I resorted to using the formulae for full-range quantization. It did not work as I expected: '127/range' is a proper scale but it didn't help with lowering accuracy. The value of 'range' can be determined by the actual value ranges of reg.coef_ and reg.intercept_

Therefore, I used '(2^18 - 1)/range' and the achieved validation error is 17.82%

```
Min/Max of coefficient values [-0.209865321585, 0.246998497499]
Min/Max of intersect values [-0.125379550542, 0.254461343394]
Misclassifications (float) = 13.77%
Misclassifications (fixed) = 17.82%
```

2 Fixed-point Design

Similarly to part 1, I have pipelined aggressively by pipelining the input/output reading loops as well as the matrix multiplication loop.

Arrays 'in_buf' and 'weight_buf' are partitioned into 32 blocks each. This is possible because we are using 8-bit integers, which are 4 times smaller than 32-bit floats in part 1. Through testing, I have found that 512 for 'TILING' provides the highest speedup.

Latency		Interval		Pipeline
min	max	min	max	Type
442919	442919	442920	442920	none

After normalization, the speedup is 532.5x compared to the baseline design.

Hardware utilization for this design:

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	129	-	-
Expression	-	-	0	4020
FIFO	-	-	-	-
Instance	0	0	8672	9184
Memory	274	-	4096	512
Multiplexer	-	-	-	8252
Register	-	-	9427	96
Total	274	129	22195	22064
Available	280	220	106400	53200
Utilization (%)	97	58	20	41

BRAM_18K is almost fully utilized. This will result in placement issues when compiling for the FPGA board.

For a realistic design that can be compiled successfully, I scaled down TILING to 128.

The speedup compared to baseline is still over 500x.

Latency		Interval		Pipeline
min	max	min	max	Type
443495	443495	443496	443496	none

To be precise, it is 531.82x faster than the baseline design.

Hardware utilization is also a bit lower than the 512-TILING design.

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	129	-	-
Expression	-	-	0	4001
FIFO	-	-	-	-
Instance	0	0	8672	9184
Memory	262	-	4096	512
Multiplexer	-	-	-	8252
Register	-	-	8895	96
Total	262	129	21663	22045
Available	280	220	106400	53200
Utilization (%)	93	58	20	41

BRAM_18K memory usage is now 93%. This is sufficient for placement in the compiling process.

3 FPGA Inference Run

When running on the FPGA, the validation error is 18.54%. As for speedup, the highest I have seen is a 55.21x speedup over the CPU. This is not as high as expected (60-70x). The reason might be the aggressive pipelining attempts.

```
FPGA accuracy: 18.54% validation error
CPU accuracy:  18.54% validation error
FPGA has a 55.21x speedup
```

4 Design Analysis

In this part, I will be writing about the 128-TILING design, where ‘in_buf’ and ‘weight_buf’ are partitioned into 32 blocks each.

There are 127 instances of multipliers. However, MACs should also be seen as a kind of multiplier. Therefore, in total, there are 256 multipliers instantiated in the design.

As for the pipelines, all of them have initiation interval II = 1. This, of course, is the optimal value.

* Loop:

Loop Name	Latency min	Latency max	Iteration Latency	Initiation Interval achieved	Initiation Interval target	Trip Count	Pipelined
- LOAD_OFF_1	5	5	2	1	1	5	yes
- LOAD_W_1	350	350	35	-	-	10	no
+ LOAD_W_2	32	32	2	1	1	32	yes
- LT	443136	443136	6924	-	-	64	no
+ LOAD_I_1	4480	4480	35	-	-	128	no
++ LOAD_I_2	32	32	2	1	1	32	yes
+ L1_L2	1287	1287	9	1	1	1280	yes
+ STORE_O_1	1152	1152	9	-	-	128	no
++ STORE_O_2	6	6	3	1	1	5	yes

As the II's of pipelines are all optimal, the only way to increase performance is by increasing the throughput of the AXI port. Therefore, the design is bandwidth-limited.