CMP SCI 635: Modern Computer Architecture

In-Datacentre Performance Analysis of a Tensor Processing Unit

Name: Kunjal Panchal Date: 21th Nov, 2019 Student ID: 32126469

Paper: 21 – Data Parallel [Jouppi17]

Strengths:

1. Machine learning is "fuzzy logic." What matters is the overall structure of the data. This means you don't have to have highly precise numbers for low-level calculations. Floating point numbers (decimal numbers) are not actually perfectly accurate in computer. The more bits we use to represent the number, the smaller the error will be. A 64-bit processor needs 4x as many wires and transistors as a 16-bit processor. Since Google doesn't care about high precision, these processing units are able to use less bits to do their calculations, saving on space and electricity.

2. Deep Learning Super Sampling is very tentative but the graphs make it look like that might really do something big performance wise. Probably more benefit/unit of area than more shaders etc. If it does it could make sense to ship the smaller chips with TPU's but not the ray tracing bits.

Weaknesses:

1. Many state of the art deep learning models are based on RNNs, especially for speech recognition and natural language processing, and RNNs do not have high enough operation intensity to efficiently use TPUs.

2. 300 W is being consumed by the tensor cores. It can do regular operations at the same time, plus NVlink alone consumes about 50 watts at full speed. Nvidia also claims the Xavier SoC can do 30 tflops at 30 W, and that's including the cpu in its power budget.

3. Googles is only tensor ops, not every other operation used in AI algorithms. The Google processor must be fed and use a lot of cpu for that too.

Questions/Assertions:

1. How difficult it'd be to build a circuit board with pretrained weights hardwired in, using analogue components to perform the non-linear transformations?

2. Google has a serious competitive advantage, though I don't see why others (including Nvidia) wouldn't be able to replicate it. Nvidia has way more experience with chip design, and this seems fairly simple as far as things go.

3. What is the difference between fp16 and integer8. Isn’t fp16 decimals and integer non-decimals, but what's the purpose of it in hardware?

4. Given that Software emulation < FPGA implementation < ASIC; is it safe to say that the future of computing is ASIC?

5. What happens when we get AMD AI cores, or maybe google or ARM AI cores in phones. If we have to train the network then how cross compatible is that training? If it isn't very compatible then that's a huge barrier entry for say ARM or AMD.