Do Large Batches Always Improve Neural Network Throughput?

Common benchmarks like ResNet-50 generally have much higher throughput with large batch sizes than with batch size =1.

For example, the Nvidia Tesla T4 has 4x the throughput at batch=32 than when it is processing in batch=1 mode.

Of course, larger batch sizes have a tradeoff: latency increases which may be undesirable in real-time applications.

Why do larger batches increase throughput for ResNet-50 and other common benchmarks?

ResNet-50 has the following:

- Input images of 224 x 224 pixels x 3 Bytes (RGB) = about 0.05MB
- Weights of 22.7MB

An Inference Accelerator also needs to store the activations that are the output of each layer of the model. In models like ResNet-50 the activations only need to be saved temporarily as they are used on the subsequent layer.

In ResNet-50, the largest activation is 0.8MB. Typically, one of the layers before or after will have half the activation size. Thus, if there is 1.2MB of SRAM for temporary activation storage, then no activations need to be written to DRAM: this is for batch=1. If all activations had to be written to DRAM, it would mean almost 10MB of activations written out and 10MB read back in when adding up the activation sizes for all layers. This means that 1.2MB of on-chip storage for temporary activations avoids 20MB of DRAM bandwidth per frame. This is almost as much as the 22.7MB of weights.

The idea of batching is actually quite simple. If performance is limited by reading in weights, then you need to process more images each time the weights are available to improve throughput. With a batch size of 10, you can read in the weights once for every 10 images, thus spreading the weight-loading slowdown across more workload. At 1.2MB of on-chip SRAM needed per image, just 12MB of SRAM for temporary activations allows batch=10 without using DRAM bandwidth for activations.

For many architectures and SRAM capacities, loading weights is the performance limiter. This is why larger batch sizes make sense for higher throughput (although they increase latency). This is true for ResNet-50 and many "simple" models because all of them use small image sizes.

What about MegaPixel CNNs?

Most people assume that every model will have higher throughput at larger batch sizes.

However, this assumption is based on the common benchmarks most chips quote today, which is ResNet-50, GoogleNet, and MobileNet. It is important to note that the thing all of them have in common is image sizes that are not real-world: 224x224 or 299x299 or 408x408 even. All of these are smaller than even a VGA, a very old computer display standard, which is 640x480. Typical 2 megapixel image sensors are >40 times larger than 224x224.

As image size grows, the weight size doesn't change. However, the size of intermediate activations grow proportionately with the larger input image size. The largest activation of any layer for ResNet-50 is 0.8MB, but when ResNet-50 is modified for 2 megapixel images, the largest activation grows to 33.5MB. Just for batch=1, on-chip SRAM storage for activations needs to be 33.5MB for the largest activation plus ½ that for the leading or following activation = 50MB. This itself is a large amount of on-chip SRAM and larger than the weights at 22.7MB. It might result in higher performance in this case to keep the weights on-chip and store at least larger activations in DRAM.

If you consider large batch sizes such as batch=10 for ResNet-50 at 2 Megapixel, there would need to be 500MB of on-chip SRAM to store all the temporary activations. Even the largest Chip C does not have enough SRAM for a large batch size. The increase in size in activations at megapixel images for ALL models is why large batch sizes don't get higher throughput for megapixel images.

Conclusion

All models that process megapixel images will use memory very differently than tiny models like ResNet-50's 224x224. The ratio of weights correspond to activation flips for large images. To really get a sense for how well an inference accelerator will perform for any CNN with megapixel images, a vendor needs to look at a benchmark that uses megapixel images. The clear industry trend is to larger models and larger images so YOLOv3 is more representative of the future of inference acceleration. Using on-chip memory effectively will be critical for low cost/low power inference.

Geoff Tate
CEO
Flex Logix