RNN Optimization on CPU, GPU, and a Custom Accelerator

Group Members: Kenny Chan & Ashwin Narkar

Goals

Previously we've explored the optimization of classifiers and CNNs by taking advantage of parallelism on GPUs, but we haven't explored implementing other algorithms or using custom hardware. We aim to simulate two RNNs, one small and one large, on a CPU, then optimize the same networks by parallelizing on a GPU and designing custom hardware. Then, we will compare the different latency, throughput, and energy efficiency capabilities between each platform to compare their advantages. Our goals are listed below.

- Simulate 2 RNNs with CPU code.
- Simulate the same 2 RNNs with GPU (CUDA) code.
- Simulate the same 2 RNNs on custom hardware.
- Measure/estimate and compare the latency of each approach.
- Measure/estimate and compare the throughput of each approach.
- Measure/estimate and compare the power efficiency of each approach.

Approach

We will first study how RNNs work and come up with two networks, one small and one large, that will be generic enough to best represent the algorithm. Then, we will work on writing and validating the CPU code to simulate the networks on the Seasnet servers. The latency and throughput will be measured by reporting the start and end times after each simulation. The energy will be estimated from the Seasnet server's CPU power specs and the simulation time. Following that, we will parallelize the code for GPUs using the V100 on the Tetracosa server, once again optimizing for our target metrics. The latency, throughput, and energy consumption will be measured the same way as the CPU.

Finally, we will design one ASIC that can operate on arbitrarily sized RNNs using Verilog. To verify, we will create testbenches that will pass in the same number of test inputs as our two RNN networks and simulate on Modelsim. The latency and throughput will be based on the number of clock cycles from Modelsim and the power will be measured using Synopsis for a 32nm process. Synopsis will also verify that the clock periods used for our latency and throughput measurements are synthesizable.