High Performance Machine Learning
Lab 3

Robert Benke
Department of Computer Architecture
Faculty of Electronics, Telecommunications and Informatics
Gdańsk University of Technology

Introduction

HPC is not just about computing power. Model training (or evaluation) is often limited by memory access (especially on external accelerators). One popular way to speed up predictions is to quantize the model. Using 8 bits or less requires much less data to store and read. Unfortunately, this leads to unstable and discretized gradients that cannot be used during training. While we will return to quantization in future labs, today we will focus on mixed precision computing. Mixed precision computation uses float32 for high precision computations (such as activations) and float16 (or bfloat16) everywhere else (`https://www.tensorflow.org/guide/mixed_precision`). Therefore, this fits perfectly with our training requirements.

1 Task 0: Measure peak memory usage (2 points)

Let us first measure the peak memory consumption and training step time for our float32 based code. Measuring GPU memory consumption is not an easy task because tensorflow always allocates all the gpu memory. One way to deal with this problem is to set a memory growth in the tensorflow configuration to change this behavior and watch nvidia-smi very carefully. Another (preferred) is to use tensorflow memory info (available in version 2.5+ of tensorflow - `https://www.tensorflow.org/api_docs/python/tf/config/experimental/get_memory_info`).).

Both techniques won't work for cpu. However, you can find memory peak of a process in the process status info (`https://man7.org/linux/man-pages/man5/proc.5.html`).

To pass the task: Measure peak memory usage for GPU (1 point) and CPU only (1 point). Save these results somewhere for further comparisons.

2 Task 1: Use mixed precision instead of float32 and adjust loss/gradients to lower precision (4 points)

Mixed precision should not have a negative impact on the performance of the model, but there are some issues that we have to deal with. One of them is float16 loss overflow (`https://www.tensorflow.org/guide/mixed_precision#loss_scaling`).

To pass the task: measure the peak memory usage and train step execution time for CPU (1 points) and GPU (1 point). Implement the loss scaling and unscaling (2 points)

3 Task 2: Compile computational graph with Accelerated Linear Algebra (XLA) (2 points))

It's the same as before :) `https://www.tensorflow.org/xla`

To pass the task: measure the peak memory usage and train step execution time for CPU (1 points) and GPU (1 point).