Introduction to High Performance Machine Learning (ECE-GY 9143)
Project Proposal
Ghulam Mujtaba gm2667

Project Title

Performance Comparison of Pytorch Python and C++ API.

Team members

Ghulam Mujtaba (gm2667)

I am currently a solo member. Given the scope of the project, that should be enough, I think.

Goal/Objective

Pytorch is primarily used through its python interface although most of the underlying high-performance code is written in C++. A C++ interface for Pytorch is also available that exposes the code underlying codebase. There are many cases where a need arises for the use of C++ instead of the primary python API to meet project needs, such as in Low latency operation for robotics, stock trading etc.

The goal of this project is to compare the performance of the two API and provide a quantitative comparison that can be used to make choices on when and where to use the different API's.

The primary reason to use Pytorch with C++ are as following:

1. **Low Latency Systems:** For real time applications such as self-driving cars, game AI and other real-time projects the need lower latency, pure C++ is a much better fit here as compared to the latency limitations of the python interpreter.
2. **Highly Multithreaded Environments:** The Global Interpreter Lock (GIL) in python prevents it from running more than one thread at a time. Multiprocessing is available but it is not as scalable and has some shortcomings. C++ is not limited by such constraints so for application where the models require heavy penalization such as Deep Neuroevolution, this can benefit.
3. **Existing C++ Codebases:** Many existing projects have existing C++ code bases, and it is more convenient to keep everything in on language as supposed to binding C++ and python. Using C++ throughout the project is a more elegant solution.

Challenges

The primary challenge is to install and setup the C++ distribution for Pytorch correctly. Getting to understand and use the new API in C++ which I have not used for ML work before.

Approach/Techniques

We will train and evaluate end-to-end a model with the C++ and Python frontends. We will be training a Deep Convolutional Generative Adversarial Networks (DCGAN) – to generate images of MNIST digits.

We will measure and evaluate the:

1. Data-loading time for each epoch.
2. Training (i.e., mini-batch calculation) time for each epoch.
3. Total running time for each epoch Run 5 epochs.
4. Inference time.

Implementation details: hardware (type of compute GPU/TPU etc, cloud based, edge devices), software (framework, existing code to reuse), dataset.

We will be using the Pytorch software framework in a ubuntu Linux operating system.

Introduction to High Performance Machine Learning (ECE-GY 9143)
Project Proposal
Ghulam Mujtaba gm2667

Code for the C++ part will be used from this tutorial, it will be modified to log the performance metrics we need:
https://pytorch.org/tutorials/advanced/cpp_frontend.html

And the Pytorch implementation from:
https://towardsdatascience.com/how-to-build-a-dcgan-with-pytorch-31bfbf2ad96a

Changes will be made to keep the code as similar as possible for best comparison results.

We will be using a NYU HPC Greene node to train and evaluate these models. Hardware specs for the nodes are as following:

- Intel Xeon Platinum 8268 24C 205W 2.9GHz Processor
- GPU: RTX8000 NVIDIA GPUs or V100 NVIDIA GPUs

Demo planned

This will basically be a presentation on the results and the code can be run to output the results.

References (if any):

https://pytorch.org/