CprE 487 Hardware Design for Machine Learning
Term Project Proposal

Tony Manschula and Henry Shires

Project Overview

Due to high demand in the fields of computer engineering and artificial intelligence, the tech industry is searching for faster ways to train, test, and deploy models. Given the recent explosion in popularity of powerful large-language and generative machine learning models, it seems reasonable to assume that more and more people will want to get into using and customizing machine learning models for their own purposes. An important consideration is that these individuals may not have an extensive amount of ML background, which makes a well-documented and easy-to-use framework a very beneficial asset.

Henry and I both had no machine learning background before coming into this class and found lab 1 to be a bit of a steep introduction given Tensorflow's relatively sparse documentation. Therefore, we propose a project that will redesign lab 1 to use PyTorch, as opposed to Tensorflow. We will implement the core learning points and activities of lab 1 using this new framework and compare the usability (in terms of documentation and relative difficulty of using the framework), training accuracy, and performance/resource utilization versus TensorFlow. If any core points are impractical or impossible to do so with PyTorch, those will be noted with the reason why there were not able to be implemented.

Background technology summary/initial related work

In the past, Tensorflow has been the go-to for researchers. In the past five to six years, however, PyTorch has seen a massive uptick in research use (O'Connor), indicating that PyTorch has been the preference for leading-edge research activity. Coupled with the fact that in 2021, 92% of the models available on HuggingFace.co were PyTorch exclusive (O'Connor), it makes sense to make the move to a more widely used and supported framework. A distinguishing factor of PyTorch is that it uses dynamic computation graphs, whereas Tensorflow's are static (Boesch). While this offers more flexibility in terms of input data sizes, its dynamic nature means that it's more difficult to optimize. PyTorch is typically regarded to be easier to work with due more closely following typical Python conventions.

Final Deliverables

Our deliverables will consist of a modified Lab 1 Jupyter notebook using PyTorch. Additionally, we will also benchmark against TensorFlow in several key areas in a writeup/presentation for the final demo:

- Usability
    - Ease of use of the API
    - Clarity and usefulness of API documentation
    - Any lab 1 activities that could not be implemented in PyTorch
- Training accuracy
    - Train a given model using a different number of epochs and report the number that achieved the highest validation accuracy
- Training time/performance
    - What hardware acceleration options do the frameworks support?
    - How much training time did each framework require to achieve its best validation accuracy?
- Resource utilization
    - Memory utilization and how each framework may lend itself to a given selection of hardware (embedded, etc.)

Tools Needed

- Python environment
    - VSCode vs. JetBrains?
- Device with a supported dedicated high-performance GPU
    - SSH into CUDA machines from lab 1
    - Alternatively, Tony's personal machine has a 3080Ti (not a flex)

Project Timeline

Week of Nov 18th: Lab 6 (if not done), set up development environment with required libraries, begin work
Week of Nov 25th (Fall Break): Implementation of core lab 1 points in PyTorch
Week of Dec 2nd: Implementation of core lab 1 points in PyTorch
Week of Dec 9th (Prep Week): Perform comparison of relevant metrics as listed above, begin report/presentation drafts
Week of Dec 16th (Finals): Complete report/presentation drafts

References

Boesch, Gaudenz. “Pytorch vs Tensorflow: A Head-to-Head Comparison.” *Viso.Ai*, 4 Dec. 2023, viso.ai/deep-learning/pytorch-vs-tensorflow/. Accessed 28 Oct. 2024.

O’Connor, Ryan. “Pytorch vs TensorFlow in 2023.” *AssemblyAI*, 14 Dec. 2021, www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2023/. Accessed 28 Oct. 2024.