Lab 1 – Debugging and Predicting Runtimes on Rosie

Course: CS3450
Author: Nigel Nelson
Date: 03/07/2022

Introduction:

- This lab acts as an exercise in accessing and using MSOE's supercomputer, Rosie, and understanding how running on different hardware specifications impacts the performance of training a neural network. Specifically, the hardware configurations compared in this lab include 1 CPU from a T4 node, 18 CPUs from a T4 node, 1 T4 GPU and 1 CPU from a T4 node, and 1 V100 GPU and 8 CPUs from a DGX node. Each one of these configurations is challenged to train a GAN (generative adversarial network) that is responsible for synthesizing random images of Pokémon characters based on existing images of Pokémon characters. In addition, the number of epochs are varied from 1, to 20, and the results from this change are observed to understand if training times scale linearly according to epochs, or if some other relationship is observed. Ultimately, this lab challenges students to understand how hardware choices will impact their network training times as well as how training parameters such as the number of epochs will impact their training times.

Predicting Relative Runtime from GPU specs:

- Given that the NVIDIA V100 has single precision throughput of 16.4 TFLOPS (floating point operations per second), and the NVIDIA T4 is capable of single precision throughput at a rate of 8.1 TFLOPS, I would expect the experimental times to train a neural network to take approximately twice as long on a single T4 than on a single V100. The reason for this is that the V100 is capable of about double the number of floating-point operations compared to the T4, and training a neural network involves back propagation, which is essentially all multiplication/addition, which both qualify as floating-point operations. Due to this, the V100 should be able to perform the same training of a neural network in half the time it takes the T4 to train the same neural network.

Experimentally Determine the Runtime

- In order to compare the impact that hardware has on training times, the gan.py script, which trains a Pokémon GAN, is ran on a single CPU, on 18 CPUS, on 1 T4 GPU and 1 CPU, and lastly on 1 V100 GPU and 8 CPUs.

Lab 1 – Debugging and Predicting Runtimes on Rosie

Course: CS3450
Author: Nigel Nelson
Date: 03/07/2022

Table 1

Times to train Pokémon GAN with various hardware configurations and number of epochs

	1 CPU	18 CPUs	1 T4 GPU & 1 CPU	1 V100 GPU and 8 CPUs
Time to train 1 epoch (seconds)	1682.0	538.1	68.6	33.4
Predicted time to train 20 epochs (seconds)	33,640.0	10,762.0	1,372.0	668.0
Time to train 20 epochs (seconds)	71,982.0	10,652.4	1,339.0	593.7

Discussion

- In this lab, various hardware configurations were used to train a Pokémon GAN in order to compare the resulting training times. Ultimately, it was found that the fastest configuration used 1 V100 GPU and 8 CPUs on a DGX node, followed by 1 T4 GPU and 1 CPU on a T4 node, followed by 18 CPUs on a T4 node, with the slowest configuration being 1 CPU on a T4 node. After reviewing the training times for the T4 GPU vs the V100 GPU, it appears that my initial hypothesis that the V100 would train a network in half the time needed by a T4 GPU was true. While the comparison is not completely equivalent, due to the fact that the DGX partition is allocated 8 CPUs for a single V100 GPU whereas the T4 partition is allocated 1 CPU for a single T4 GPU, a clear pattern can be seen between the training times. For example, to train a single epoch, a single T4 GPU took 68.6 seconds, whereas a single V100 GPU took 33.4 seconds. As for 20 epochs, a single T4 GPU took 1,339.0 seconds, whereas a single V100 GPU took 593.7 seconds. From this, it can be seen that in the experiments that the V100 GPU took approximately half the time to train the Pokémon GAN when compared to the T4 GPU. Furthermore, my predictions of the time required to train for 20 epochs appear to be correct as well, with the only outlier being the 1 CPU configuration. Besides this one hardware configuration, all the other hardware configurations took approximately 20 times as long to train 20 epochs as they took to train a single epoch. As for the 1 CPU configuration's time to train 20 epochs, it took approximately double my predicted time. What may have happened is

Lab 1 – Debugging and Predicting Runtimes on Rosie

Course: CS3450
Author: Nigel Nelson
Date: 03/07/2022

that in this training run, I may have been only using a single hyperthreaded core, and another cluster user was using the other hyperthreaded core on the same CPU, resulting in my task only receiving half of the CPU even though this task required 100% CPU usage. This could have been mitigated by assigning 2 CPUs in my testing as opposed to 1 CPU, reducing the risk that another cluster user's work would be assigned to the same physical core. If this was done, I am sure that the predicted pattern of 20 epochs taking 20 times as long as a single epoch would be seen for all hardware configurations.