Jackson Trexler

2000 Hours to Automate

Senior Project: P.A.N.T.S.

4/15/2023

Self Assessment

A

I participated in the collection of raw data, I was primarily responsible for the environments we used to test and run our models, as well as running the tests themselves. Surprisingly I gained quite a bit of experience with computer hardware and Linux based operating systems, which was not something I anticipated would be a part of the project. Part-way through the project, we were having difficulties using Google Colab using the free plan, and we learned the easy way that Colab pro is deceptively expensive. We started this project under the impression we may have to rely on our own hardware. A groupmemer had a computer with a dedicated AMD graphics card, but unfortunately it is difficult to use AMD cards with machine learning packages (if not impossible with many). I had a laptop with a dedicated nVidia graphics cards that had just high enough specs to train the models we wanted so we decided to use that. Initially we attempted running our pytorch code on the laptop as is, but we ran into a problem. Collabs is a linux based development environment, we spent hours trying to retrofit our previous code to work on Windows but Windows fundamentally handles multi-threading and child-processes differently. I then spent a considerable amount of time then trying to install and run linux from a USB device. Initially I wanted to run Ubuntu or Arch from installable boot media. The install process for Ubuntu actually involves flashing an ISO of a fully functional version of the OS onto a flash drive. This flash drive can be configured to have some amount of persistent storage. This storage, regrettably, is not included as part of the root directory but is partitioned separately. I was only able to install python and a few packages before I ran out of space on the root directory, and the root directory was not even considered persistent, any packages I installed were not guaranteed to still be around upon restart. I investigated ways to change the install location for the required packages, but this was not recommended for even intermediate Linux users. It can not be specified where a package is installed via apt-get install, the package itself specifies location. Changing install location would involve editing the packages themselves before they were installed, and even then there is no telling whether they would be able to locate their dependencies or be located as dependencies. Given that installing required packages was the easiest part of setting up our development environment, I decided to can the idea of running from boot media and instead install it as a proper, primary operating system on another USB device. The formatting and install process went well but no matter which way I configured it or my BIOS/UEFI settings on multiple computers, the stick I installed linux on was not recognized as bootable.

In retrospect I should have tried installing Ubuntu to a proper solid state from the beginning, but I learned much more from my initial though process. After swapping out solid states on the laptop, I installed Ubuntu 22.04. Following my usual process of setting up the OS, getting packages, and importing our previous code, we were finally able to run our project on the laptop. We ran into an additional problem, one that we had run into before. In order to accelerate machine learning, it is almost necessary to use a graphics card. The fact that the laptop had a NVidia card was the primary reason we were using it. Setting up a Linux device to be CUDA capable is a little less straight forward than it is on Windows. Pytorch only supports specific versions of CUDA, something we found out while setting up on Windows, but we also found out that while installing on Linux, factors such as the version of Ubuntu and which type of installer was used (local deb, network deb, local runfile) both had an impact on whether or not PyTorch would recognize the system as CUDA capable. Most tutorials were produced for either older Ubuntu distributions installed versions of the CUDA toolkit that were considered the latest at the time, which were generally installed differently from older versions. Eventually I did find the correct permutation, CUDA was recognized and the performance of our model improved dramatically. Finally, in the last few weeks of project work I worked to optimize our model, trying to reduce error as much as possible. Creating new iterations of the model was mostly trial and error: Which values can be tweaked without crashing the GPU, which values decrease error meaningfully. Unfortunately we realized a bit too late that the dataset we were using was not nearly large enough to generalize voice transcription across voices not included within the librespeech dataset. Even using the libre-500 dataset, which multiplied training time, the best performance we were able to get outside of the training / test datasets were vague approximations of certain sounds and syllables (“expo” recorded as “es po”). We were initially concerned this may be an issue of overfitting, but reducing epochs and layers, even to the point where error dramatically increase, did not result in generalization. I learned much about Linux, project setup, working in different operating systems, machine learning concepts, how to set up data to be used in machine learning, and how to use PyTorch in a greater capacity than what was demonstrated in my courses. I feel that I have enough practical understanding in machine learning to seek a career involving it.

B

Our group made two machine learning models that unfortunately did not perform to the level we desired, though we do have a function demo that serves as proof of concept. The division of labor regarding managing the models and environment themselves, creation of UI, proof of concept, and team management went nicely. Josh had more experience with machine larning as he completed Deep Learning previously, a course that I was actively taking. Justin created a functional UI, and is primarily responsible for putting together the demo we were able to show off at expo. Just was also instrumental for assembling summarization data from raw text I fetched that we were using to trian our summarization model. It can be difficult to work with people that you like, I had a difficult time shutting down other people's ideas if they were unlikely to work, and I am sure my teammates felt the same way about me. I very much enjoyed working with my group as both Josh and Justin are talented and driven, but we also share the same sense of humor, which is not conducive to productivity. I do feel that despite knowing a bit ways more about Linux than my group members, they were more in their element during this project than I was. I put in considerable effort towards the text to summarization model but it unfortunately didn not materialize to much.

My primary contributions that benefitted the team was while training the speech model and both debugging and setting up the appropriate environment for it. We are all a bit disappointed in how the project turned out given we wanted to train our own models. I understand if my group members are disapointed with the job I did on the summarization model but given the difficulty of the project we set out to do, I am personally happy with my own development and accomplishments, and I hope they are satisfied with theirs. I would like to give special recognition to both Josh and Justin as Josh had a very great depth of how to create deep learning models, I would not have been able to make necessary adaptations to the code to begin testing it without his help during pair programming. Justin is the primary reason we had a functioning demonstration at expo, and he was always very helpful while we were collecting training data.