Comparing GPU performance for deep learning between Pop!\_OS, Ubuntu, and Windows.

Does it really matter?

As a deep learning enthusiast, I always find myself stuck with the same question. ***Which OS should I choose for Deep Learning?*** The questions that follow are: Should I go for Windows, or should I go for Linux? If Linux, then what distro? Debian-based or Arch-based?

Some people think that today, in 2020, it doesn’t really matter whether you go for Windows or Linux; if you decide to go with Linux, it doesn’t really matter which distro (distribution) of Linux you choose. If you search for what OS should I choose for ML/DL, below are some of the answers you get:

* *IT DOESN’T MATTER.*
* *Ubuntu, because most libraries and tools make their way to Linux first.*
* *Ubuntu, because it has the largest community.*
* *It is a hassle to get CUDA and CuDNN working with Windows.*

Now with WSL (Windows Subsystem for Linux), it is possible to run any Linux distro directly in Windows 10 without needing a dedicated Virtual Machine (Virtual Box, etc.). Microsoft is working closely with NVIDIA to bring GPU computing to WSL2. It is already available for preview in Windows Insider Program. I am really looking forward to it.

Since I did not find any answers that helped me compare performance, I decided to do it myself. Hence, I am writing this post. In this post, I will be running the same model on Pop!\_OS, Ubuntu, and Windows 10. The performance will also vary from one PC to another depending on the configuration.

I have an:

**Alienware Aurora R9**

* **Processor: Intel ® Core ™ i7-9700 @ 3.00 GHz (Cores: 8, Threads: 8)**
* **RAM: 16 GB DDR4 (8x2)**
* **Graphics Card: NVIDIA GeForce RTX 2060 (6GB) –** 
  + **Driver version: 440.100\_Pop**
  + **CUDA: 10.2\_Pop**

**I know people will say RTX 2060 is not meant for machine learning, try 2070, 2070 Ti, or 2080, but I do not have enough money to invest in a new rig.**

**I am not going to get into detail in setting up the environment and downloading the libraries because if you are reading this article, I am sure you are way ahead on that.**

**Note: I am most comfortable using Anaconda as my main Python distribution. Hence, setting up Tensorflow with cuda-toolkit and cudnn takes only a single line of code: conda install tensorflow-gpu=(\*version\*) This works on any OS. The only drawback that I found on Windows is that for Windows, the latest version of TF is 2.1, whereas Ubuntu and Pop have 2.2\*.**

**Since this is just a performance comparison, I will not focus on the model and how good it is. However, I am planning to post an article on image classification with Tensorflow 2.x soon. The test comparison is based on two image datasets, i.e., cats-dogs-pandas and ASL (American Sign Language).**

**Cats-dogs-pandas have a 1000 images each. I divided this data to distribute 2400 images for training, 525 for validation, and 75 for testing (predictions). ASL, on the other hand, has 3000 images for 29 different classes for training, of which I took 450 for validation. Testing contains 29 images, one for each category.**

***The model for cats-dogs-pandas is a simple two-layered CNN model with Dropouts and a fully-connected (Dense) layer before the output. The model looks like this:***

***I used RMSprop as my optimizer with a learning rate of 0.01 and trained for 100 epochs.***

***The model for ASL is deeper compared to that of cats-dogs-pandas. It looks like this:***

***Here too, I used RMSprop as my optimizer with a learning rate of 0.01 and trained for 100 epochs.***

**I decided to do a *3-fold validation* with the comparison test (pun intended).**

**I decided to test three different models for each dataset, starting from a shallow Conv. Network to a deep network.**

**Here are the three models for the CDP dataset:**

**Here are the three models for the ASL dataset:**

**The below table shows the comparison results for different models on different OS:**

**Conclusion:**