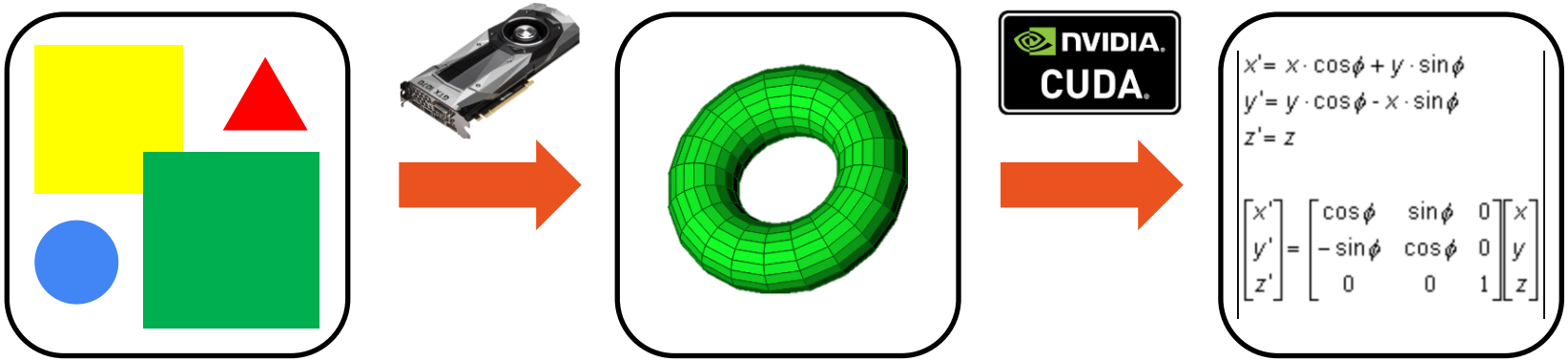
CUDA with WSL2 and Ubuntu

without Docker

윤성국(Geoff Yoon)
Epsilon-delta / 코딩이랑 무관합니다만,(NO(Not Only) conding)
김도형(DoHyeong Kim)
Forwiz system / FreeSoul / Gray hacker

CUDA

GPGPU(General-Purpose computing on Graphics Processing Units)



그래픽 처리
Graphic processing

더 무거운 그래픽 처리
Heavier Graphic processing

일반적인 목적의 계산
General-Purpose Computing

CUDA

CUDA 외의 GPGPU 라이브러리

Other GPGPU candidates

OpenCL

OpenMP

CUDA

Pytorch(딥러닝 프레임워크)는 이것들을 지원합니다.

Pytorch(Deep learning Framework) support these.

TORCH.BACKENDS

torch.backends controls the behavior of various backends that PyTorch supports.

These backends include:

- `torch.backends.cuda`
- `torch.backends.cudnn`
- `torch.backends.mps`
- `torch.backends.mkl`
- `torch.backends.mkldnn`
- `torch.backends.openmp`
- `torch.backends.opt_einsum`
- `torch.backends.xeon`

CUDA

But….

우리 Intel에서 엄청난
그래픽카드를 출시했습니다.

We, Intel, have released a great graphics card .

그래서 CUDA 됩니까?
So does CUDA work?

CUDA를 왜 여기서 찾아!

Why looking for CUDA here!

다른건… stackoverflow에서
안 알려주니까

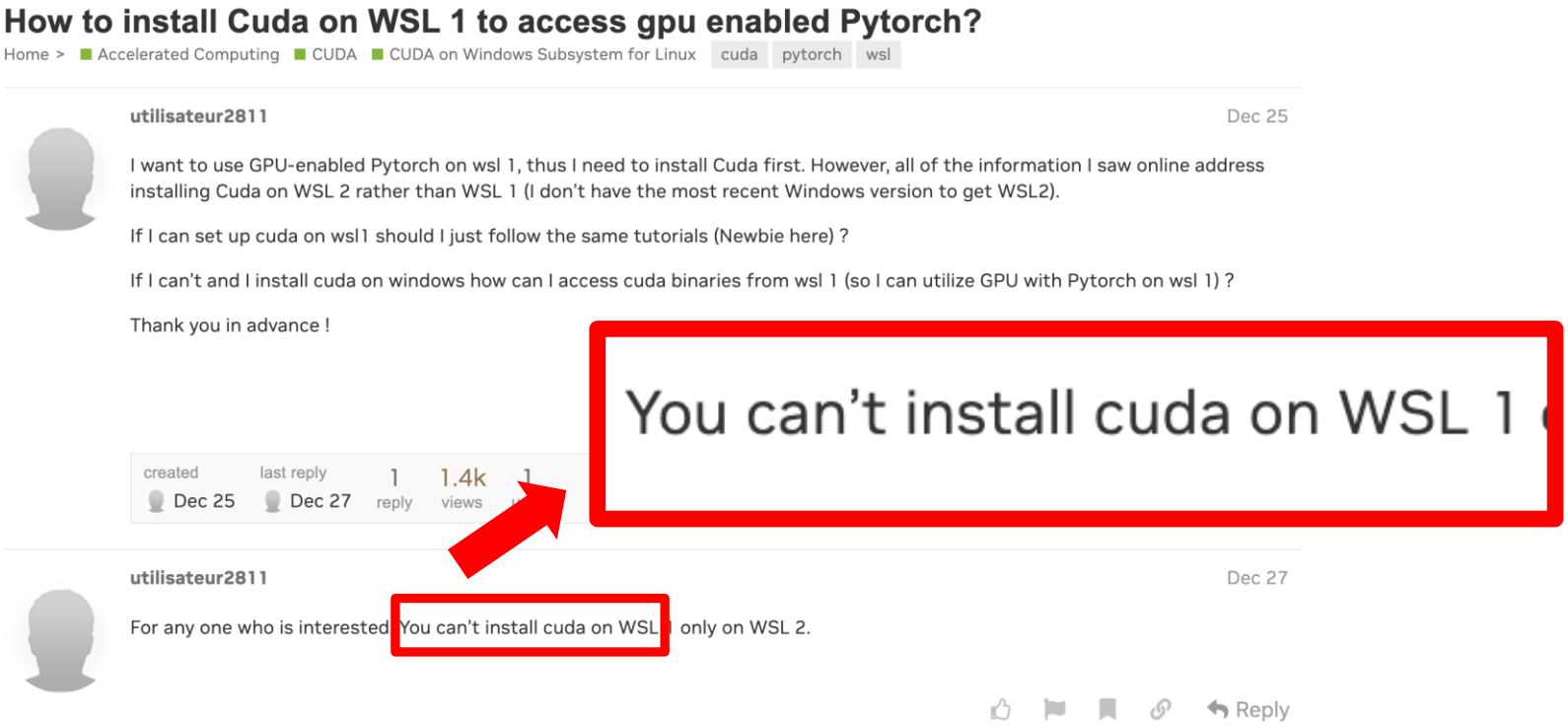
Anything else... Because
stackoverflow doesn't tell me

CUDA

Deep learning

NVIDIA
CUDA

CUDA on WSL1



How to install Cuda on WSL 1 to access gpu enabled Pytorch?

Home > Accelerated Computing > CUDA > CUDA on Windows Subsystem for Linux | cuda | pytorch | wsl

utilisateur2811 Dec 25

I want to use GPU-enabled Pytorch on wsl 1, thus I need to install Cuda first. However, all of the information I saw online address installing Cuda on WSL 2 rather than WSL 1 (I don't have the most recent Windows version to get WSL2).

If I can set up cuda on wsl1 should I just follow the same tutorials (Newbie here) ?

If I can't and I install cuda on windows how can I access cuda binaries from wsl 1 (so I can utilize GPU with Pytorch on wsl 1) ?

Thank you in advance !

created Dec 25 | last reply Dec 27 | 1 reply | 1.4k views | 1

You can't install cuda on WSL 1

utilisateur2811 Dec 27

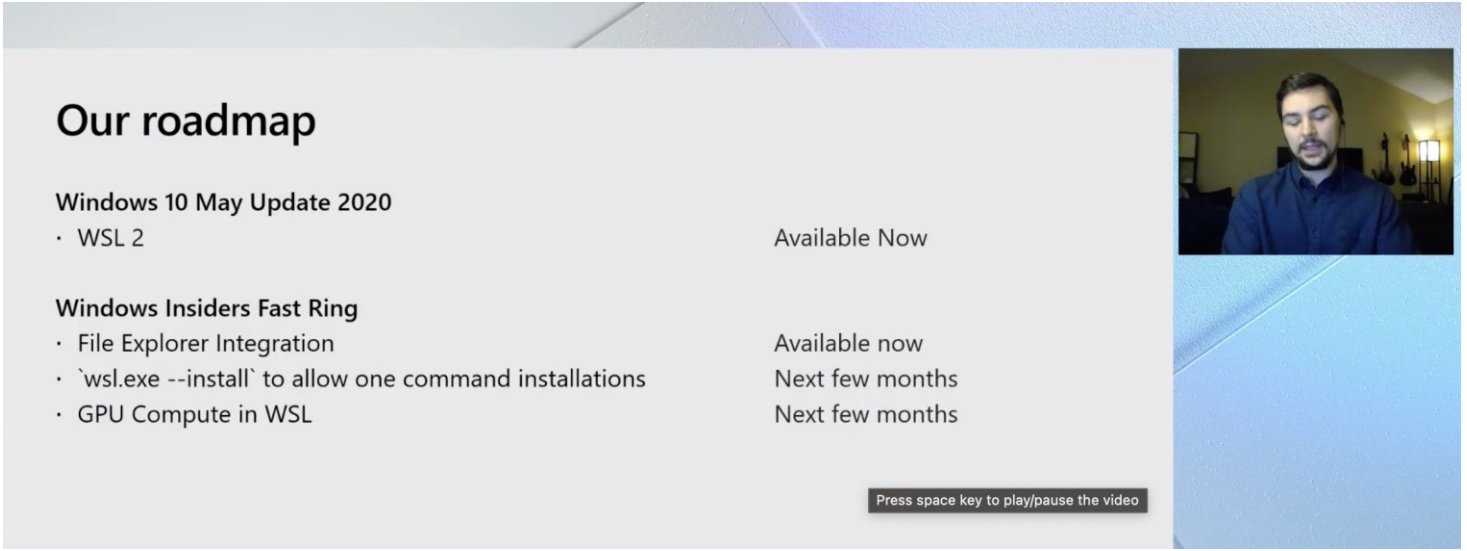
For any one who is interested You can't install cuda on WSL 1 only on WSL 2.

Reply

GPU compute on WSL2

build 2020에서 MS는 GPU compute에 대한 지원을 발표합니다.

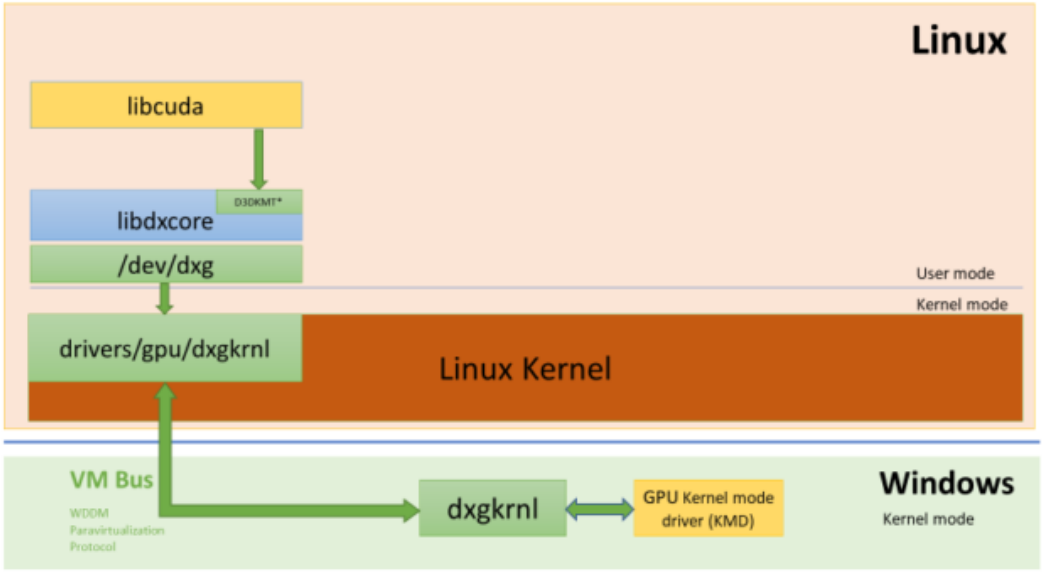
At Build 2020 MS announced support for GPU compute on WSL2.



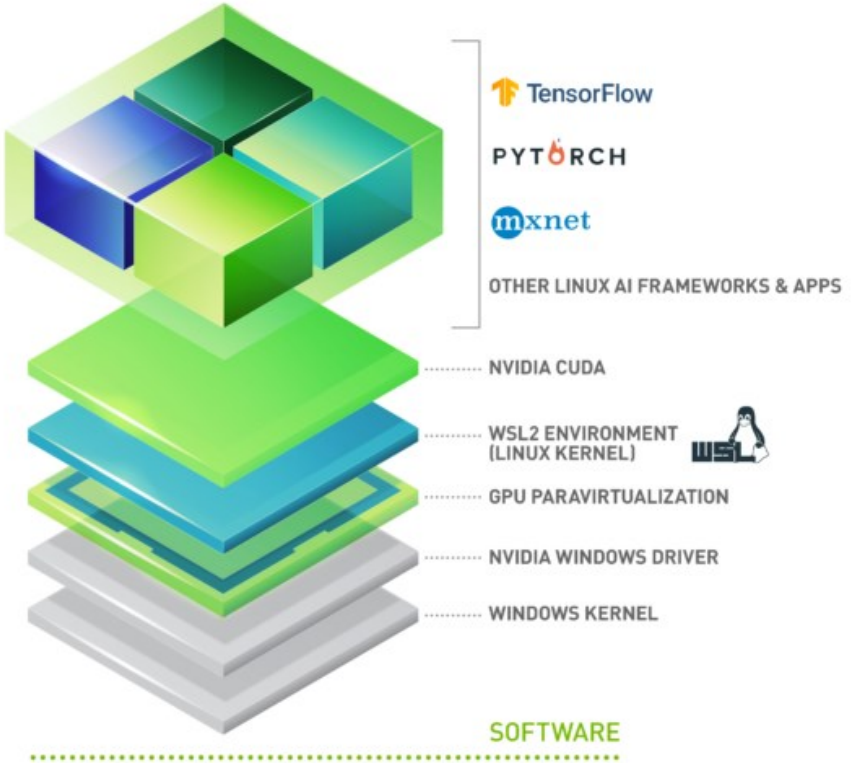
CUDA on WSL2

리눅스 게스트에서 실행되는 CUDA 유저모드 드라이버를 지원하는 WDDM 모델

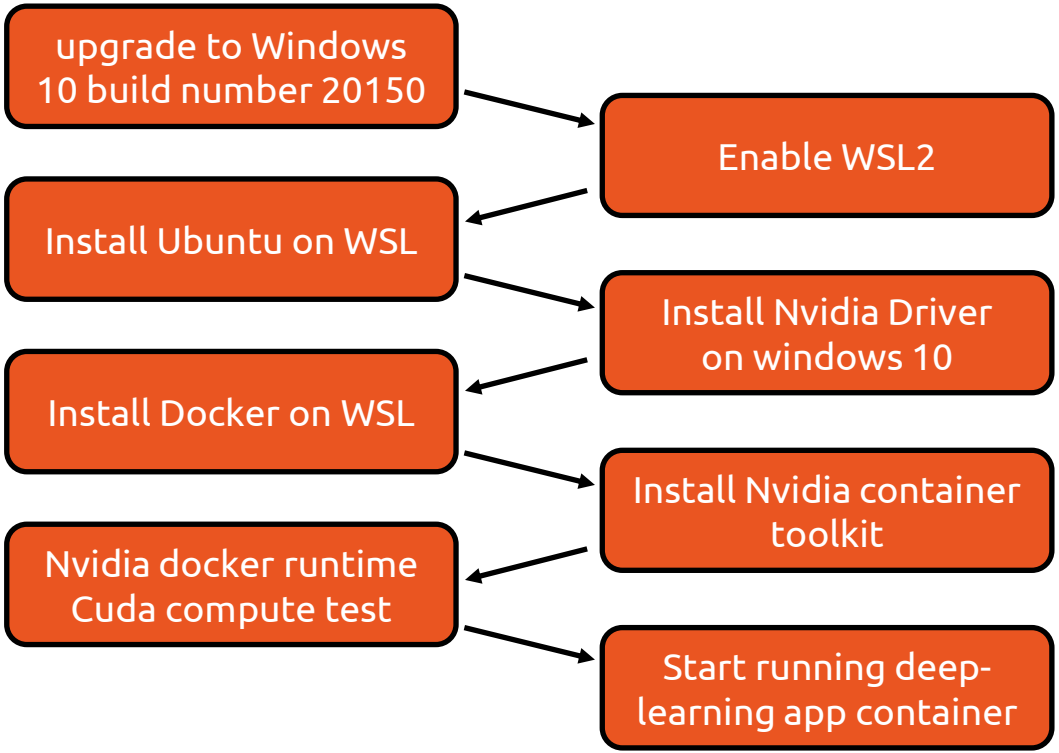
WDDM supporting CUDA user mode driver running inside Linux Geust

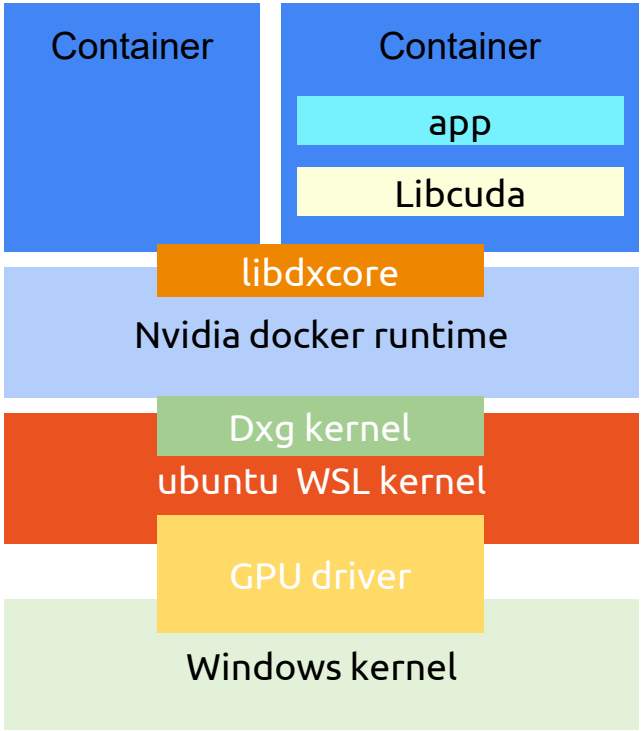


CUDA on WSL2

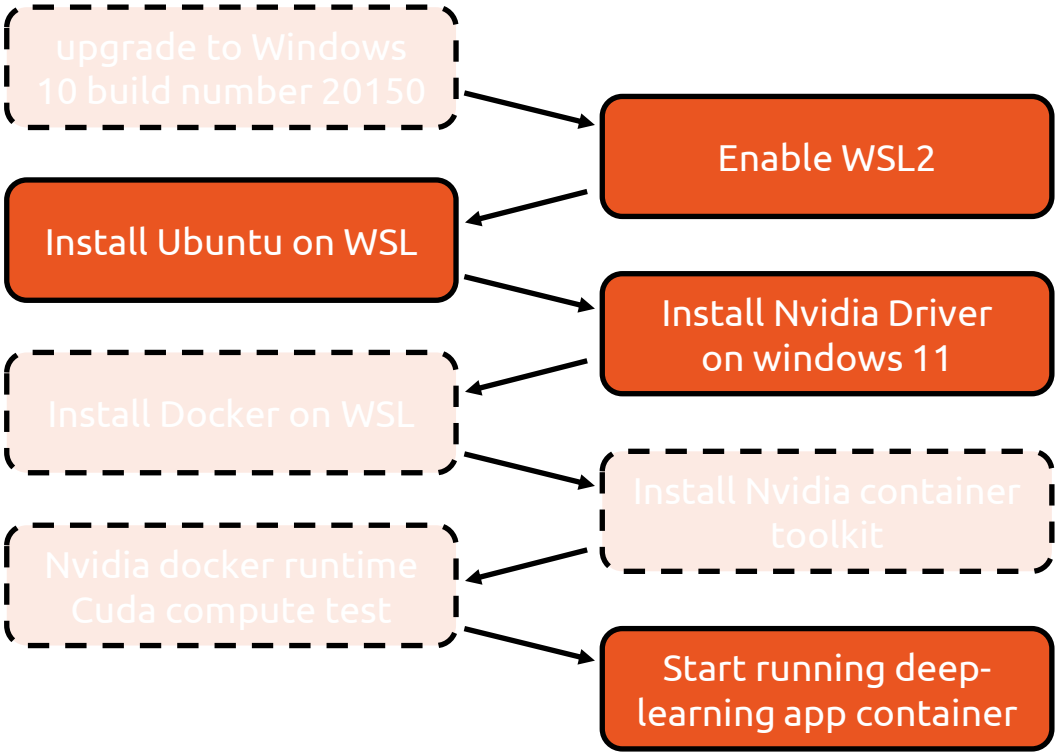

TensorFlow
PYTORCH
mxnet
OTHER LINUX AI FRAMEWORKS & APPS
NVIDIA CUDA
WSL2 ENVIRONMENT
(LINUX KERNEL)
WSL
GPU PARAVIRTUALIZATION
NVIDIA WINDOWS DRIVER
WINDOWS KERNEL
SOFTWARE

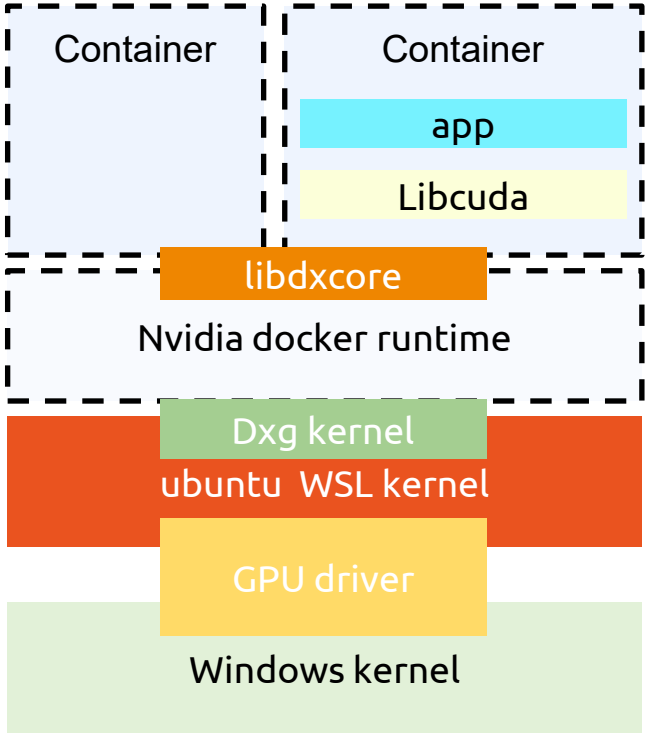
Setting deep learning env on wsl2 (windows 10)



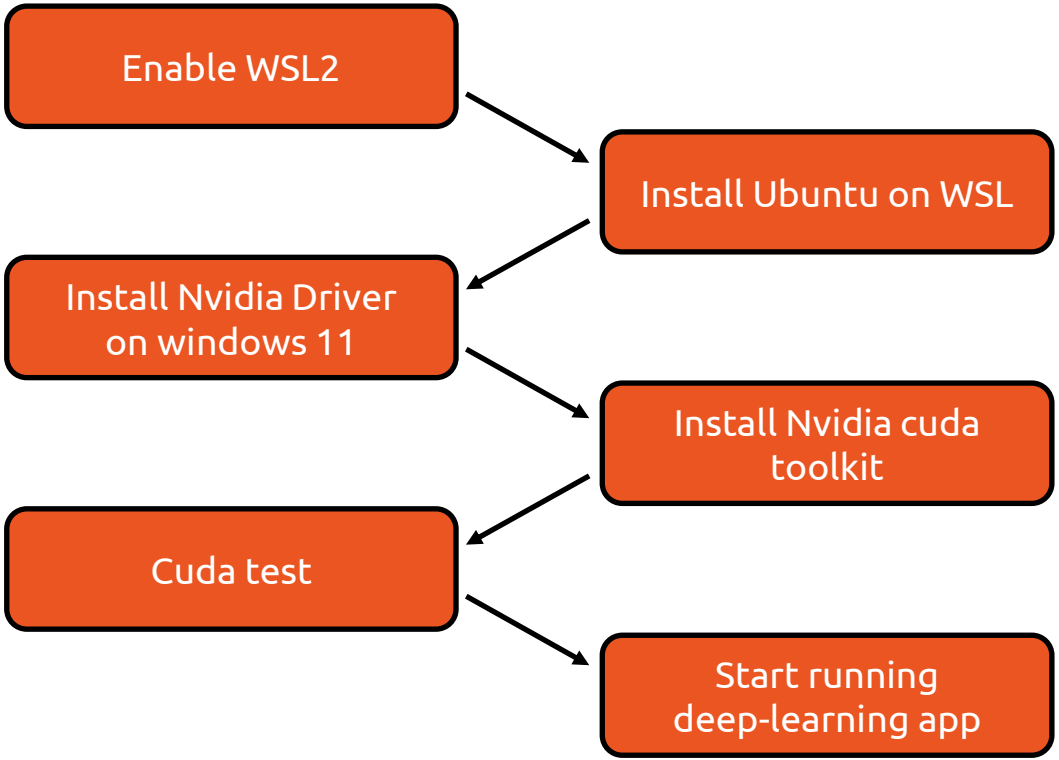


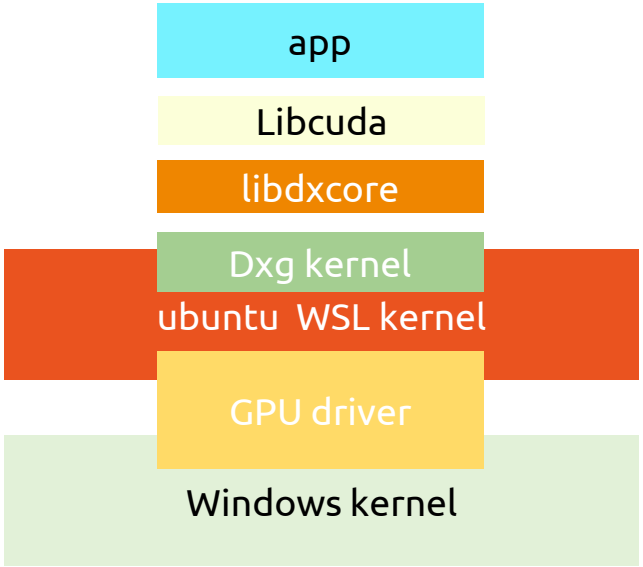
Setting deep learning env on wsl2 (windows 11)





Setting deep learning env on wsl2 (windows 11)





Deep learning on WSL2

The reason I get started cuda in wsl

GPU자원 좀 빌려줄 수 있나?
can i borrow your gpu resources?

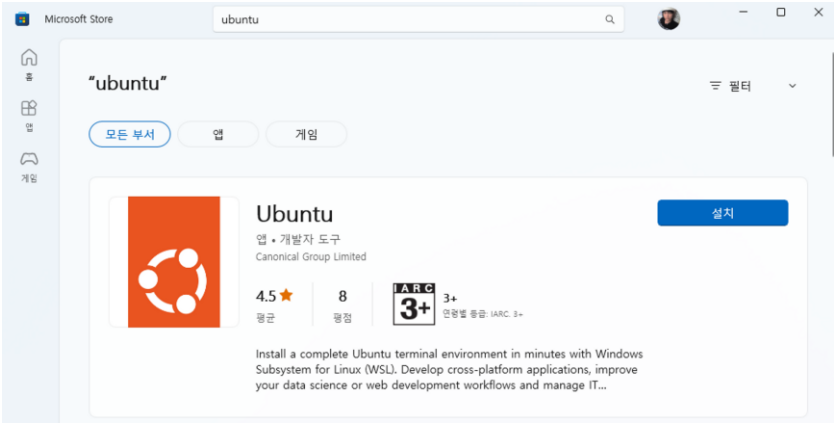
가능은 한데 그런데 나 윈도우인데
It's possible, but my OS is windows.

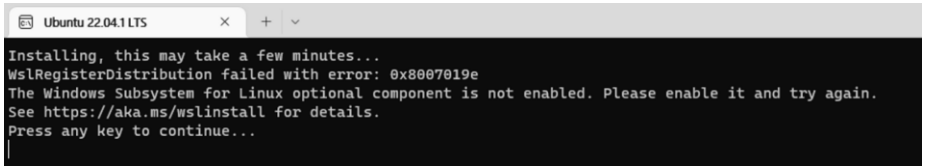
우리 팀 환경이 ubuntu라서 좀 곤란한데
My team env is ubuntu, so that's difficult

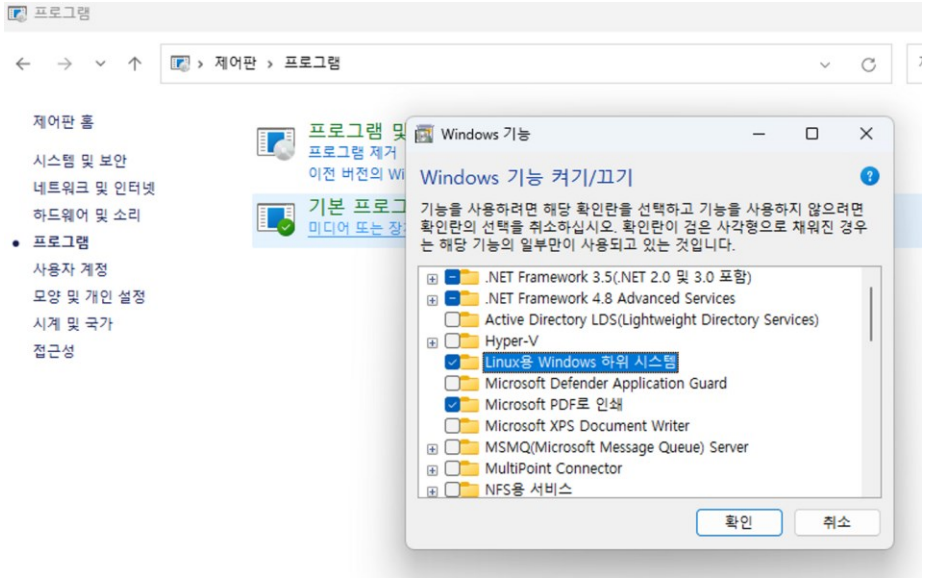
뭐가 문제야, wsl2에서 cuda 되잖아
What's the problem, cuda is working in wsl2

Deep learning on WSL2

Install ubuntu

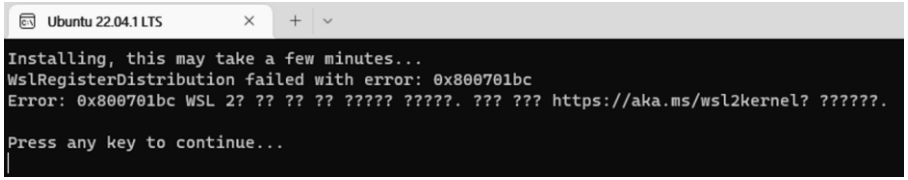


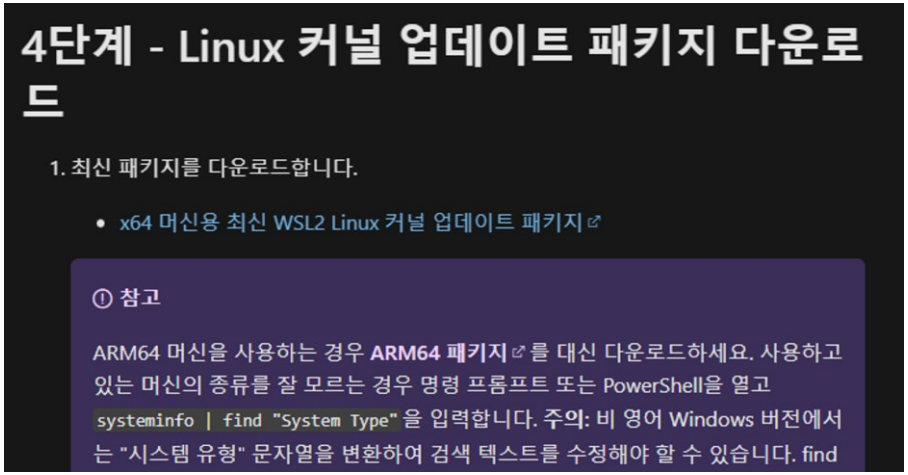


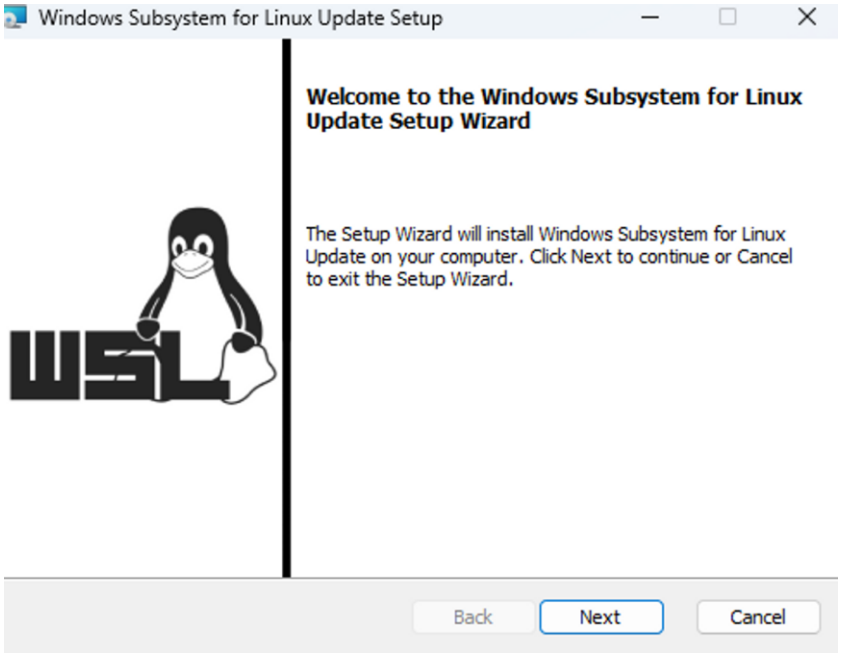


Deep learning on WSL2

Install ubuntu

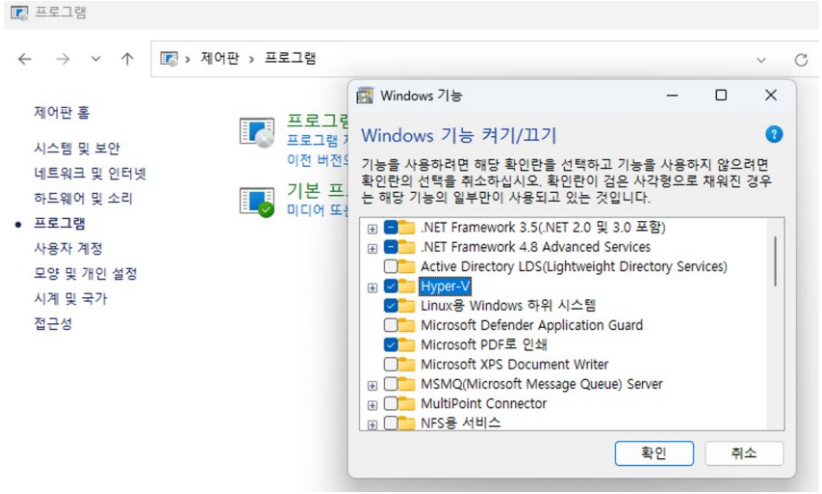


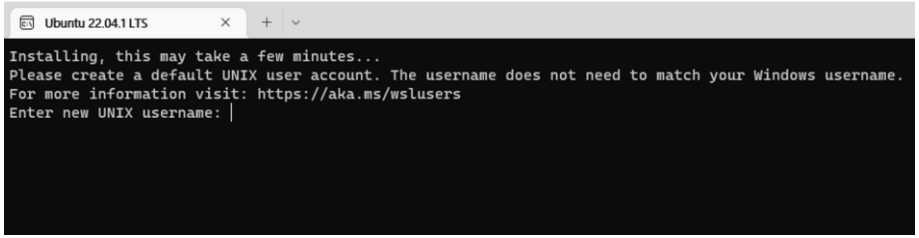




Deep learning on WSL2

Install ubuntu





Deep learning on WSL2

Setup remote SSH

```
sudo apt install update
sudo apt install openssh-server
sudo vi /etc/ssh/sshd_config
```

```
#PubkeyAuthentication yes
PasswordAuthentication yes
```

```
sudo service ssh --full-restart
```

Install and setup ssh server

```
sudo apt install net-tools
```

ports_wsl.ps1

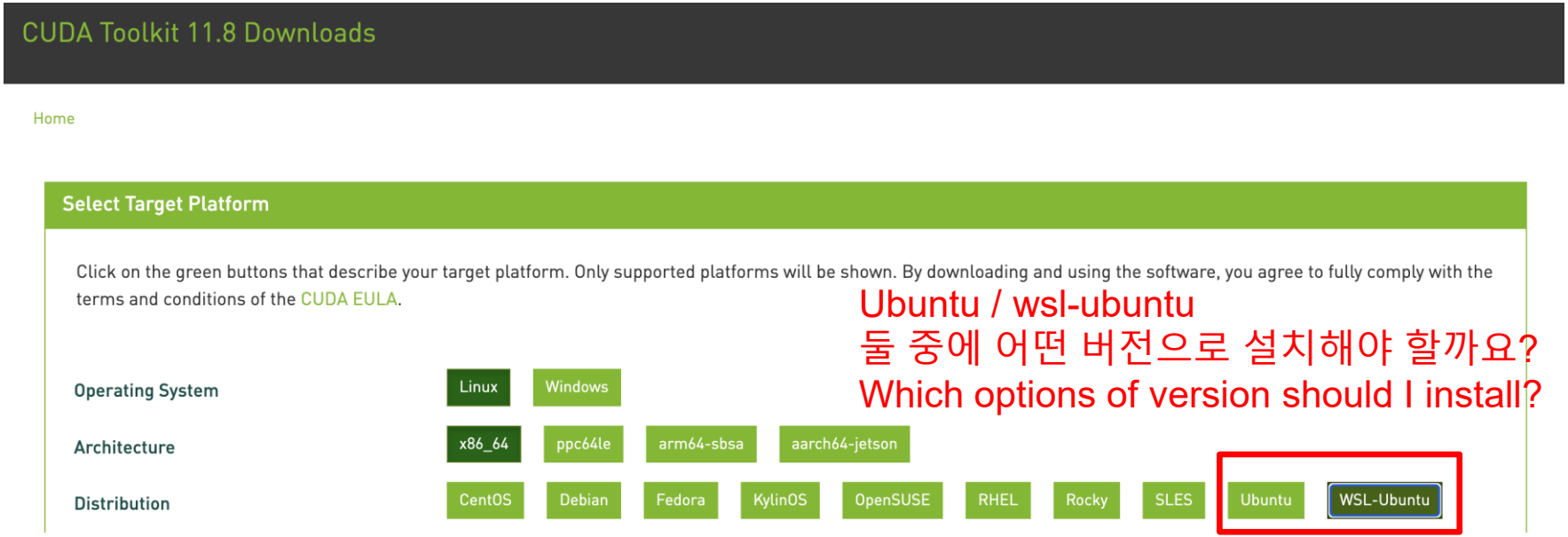
```
Users > epsilondelta > Library > Mobile Documents > com~apple~TextEdit > Documents > ports_wsl.ps1
If (-NOT ([Security.Principal.WindowsPrincipal][Security.Principal.WindowsIdentity]::GetCurrent())
        .IsInRole([Security.Principal.WindowsBuiltInRole] "Administrator")) {
  $arguments = "& '" + $myinvocation.mycommand.definition + "'"
  Start-Process powershell -Verb runAs -ArgumentList $arguments
  Break
}

$remoteport = bash.exe -c "ifconfig eth0 | grep 'inet '"
$found = $remoteport -match '\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}';

if ( $found ) {
  $remoteport = $matches[0];
}
else {
  Write-Output "The Script Exited, the ip address of WSL 2 cannot be found";
  exit;
}

$ports = @(22, 7861);

Invoke-Expression "netsh interface portproxy reset";

for ( $i = 0; $i -lt $ports.length; $i++ ) {
  $port = $ports[$i];
  Invoke-Expression "netsh interface portproxy add v4tov4 listenport=$port connectport=$port connectaddress=$remoteport";
}

Invoke-Expression "netsh interface portproxy show v4tov4";
```

```
PS C:\Users\ehqhd\Desktop> .\ports_wsl.ps1


ipv4 수신 대기:                 ipv4에 연결:
주소            포트            주소            포트
----------      ----------      ----------      ----------
                22              172.18.250.78   22
                7861            172.18.250.78   7861

PS C:\Users\ehqhd\Desktop>
```

Port-forwarding windows to wsl

Deep learning on WSL2

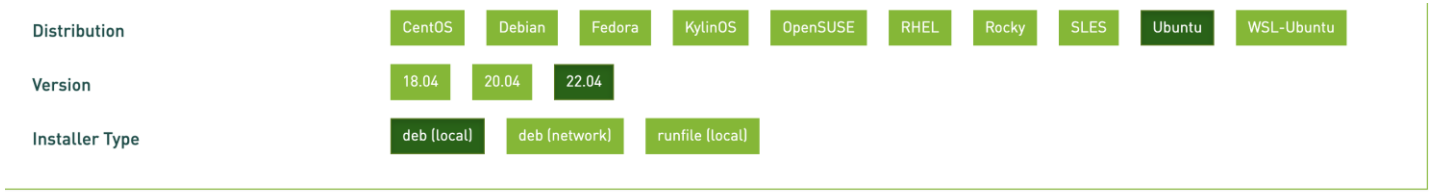
Install cuda toolkit



Deep learning on WSL2

Install cuda toolkit

사실 wsl 버전 아니어도 가능합니다.
Actually, it is possible even if not a wsl vervion.



Download Installer for Linux Ubuntu 22.04 x86_64

The base installer is available for download below.

Base Installer

Installation Instructions:

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/cuda-ubuntu2204.pin
$ sudo mv cuda-ubuntu2204.pin /etc/apt/preferences.d/cuda-repository-pin-600
$ wget https://developer.download.nvidia.com/compute/cuda/11.8.0/local_installers/cuda-repo-ubuntu2204-11-8-local_11.8.0-520.61.05-1_
amd64.deb
$ sudo dpkg -i cuda-repo-ubuntu2204-11-8-local_11.8.0-520.61.05-1_amd64.deb
$ sudo cp /var/cuda-repo-ubuntu2204-11-8-local/cuda-*-keyring.gpg /usr/share/keyrings/
$ sudo apt-get update
$ sudo apt-get -y install cuda
```

Deep learning on WSL2

Install cuda toolkit

```
geoff@KimDH_Works:~$ nvidia-smi
Sun Nov 20 16:32:06 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 520.56.06    Driver Version: 522.30       CUDA Version: 11.8     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA TITAN V      On   | 00000000:4F:00.0  On |                  N/A |
| 28%   38C    P2    32W / 250W |    865MiB / 12288MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

```
sudo apt install nvidia-utils-$version
```

Deep learning on WSL2

Setup anaconda environment

```
sudo apt install libgl1-mesa-glx libegl1-mesa libxrandr2 libxrandr2 libxss1 \
                 libxcursor1 libxcomposite1 libasound2 libxi6 libxtst6
wget https://repo.anaconda.com/archive/Anaconda3-2022.10-Linux-x86_64.sh
./Anaconda3-2022.10-Linux-x86_64.sh
```

```
Welcome to Anaconda3 2022.10

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
===================================================
End User License Agreement - Anaconda Distribution
===================================================

Copyright 2015-2022, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between you and Anaconda, Inc. ("Anaconda") and g
overns your use of Anaconda Distribution (which was formerly known as Anaconda Individual Edition).

Subject to the terms of this Agreement, Anaconda hereby grants you a non-exclusive, non-transferable license to:

  * Install and use the Anaconda Distribution (which was formerly known as Anaconda Individual Edition),
  * Modify and create derivative works of sample source code delivered in Anaconda Distribution from Anaconda's reposito
ry, and;
  * Redistribute code files in source (if provided to you by Anaconda as source) and binary forms, with or without modif
ication subject to the requirements set forth below, and;
```

Deep learning on WSL2

Setup anaconda environment (with setting cuda toolkit)

```
conda create -n diffusion_env python=3.10 conda
conda activate diffusion_env
conda install pytorch torchvision torchaudio pytorch-cuda=11.7 -c pytorch -c nvidia
```

```
  added / updated specs:
    - pytorch
    - pytorch-cuda=11.7
    - torchaudio
    - torchvision


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    cuda-11.7.1                |                0           1 KB  nvidia
    cuda-cccl-11.7.91          |                0         1.2 MB  nvidia
    cuda-command-line-tools-11.7.1|                0           1 KB  nvidia
    cuda-compiler-11.7.1       |                0           1 KB  nvidia
    cuda-cudart-11.7.99        |                0         194 KB  nvidia
    cuda-cudart-dev-11.7.99    |                0         1.1 MB  nvidia
    cuda-cuobjdump-11.7.91     |                0         158 KB  nvidia
    cuda-cupti-11.7.101        |                0        22.9 MB  nvidia
    cuda-cuxxfilt-11.7.91      |                0         293 KB  nvidia
```

```
(diffusion_env) geoff@KimDH_Works:~$ python
Python 3.10.8 (main, Nov  4 2022, 13:48:29) [GCC 11.2.0] (
Type "help", "copyright", "credits" or "license" for more
>>> import torch
>>> torch.cuda.is_available()
True
>>>
```

Deep learning on WSL2

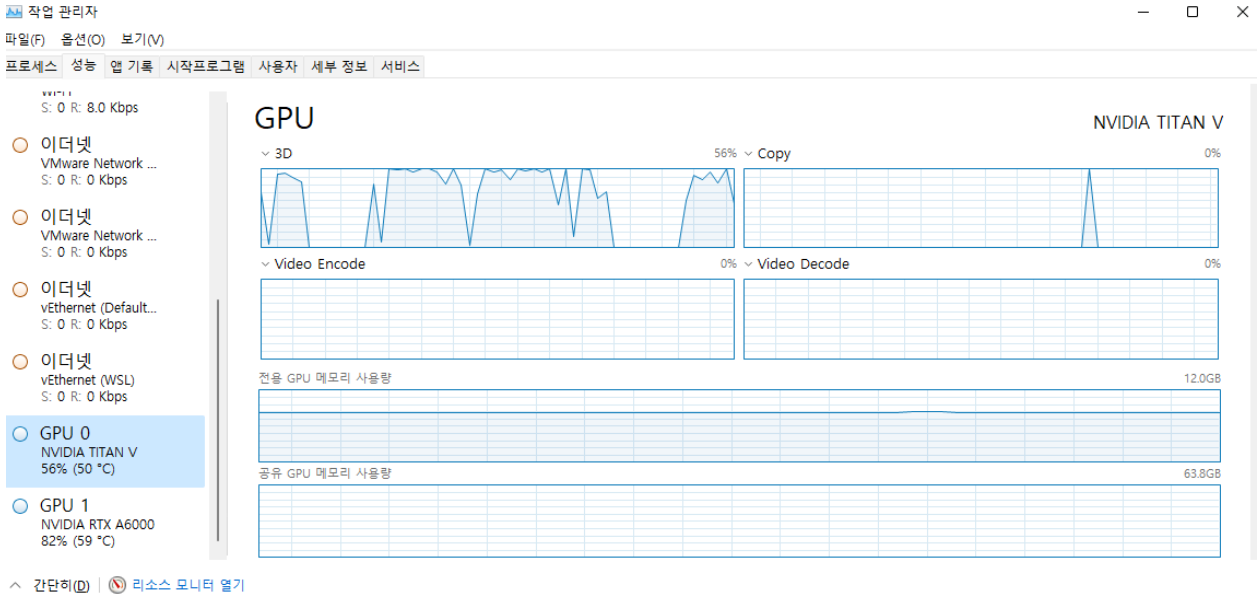
TEST : Train model using cuda

```
>>> class NeuralNetwork(nn.Module):
...     def __init__(self):
...         super().__init__()
...         self.flatten = nn.Flatten()
...         self.linear_relu_stack = nn.Sequential(
...             nn.Linear(28*28, 512),
...             nn.ReLU(),
...             nn.Linear(512, 512),
...             nn.ReLU(),
...             nn.Linear(512, 10)
...         )
...     def forward(self, x):
...         x = self.flatten(x)
...         logits = self.linear_relu_stack(x)
...         return logits
...
>>> model = NeuralNetwork().to(device)
>>> print(model)
NeuralNetwork(
  (flatten): Flatten(start_dim=1, end_dim=-1)
  (linear_relu_stack): Sequential(
    (0): Linear(in_features=784, out_features=512, bias=True)
    (1): ReLU()
    (2): Linear(in_features=512, out_features=512, bias=True)
    (3): ReLU()
    (4): Linear(in_features=512, out_features=10, bias=True)
  )
)
>>> loss_fn = nn.CrossEntropyLoss()
>>> optimizer = torch.optim.SGD(model.parameters(), lr=1e-3)
```

```
>>> epochs = 20
>>> for t in range(epochs):
...     print(f"Epoch {t+1}\n-------------------------------")
...     train(train_dataloader, model, loss_fn, optimizer)
...     test(test_dataloader, model, loss_fn)
...
Epoch 1
-------------------------------
loss: 2.302485  [    0/60000]
loss: 2.294812  [ 6400/60000]
loss: 2.269913  [12800/60000]
loss: 2.275481  [19200/60000]
loss: 2.257059  [25600/60000]
loss: 2.227393  [32000/60000]
loss: 2.237715  [38400/60000]
loss: 2.197939  [44800/60000]
loss: 2.199718  [51200/60000]
loss: 2.177548  [57600/60000]
Test Error:
 Accuracy: 40.6%, Avg loss: 2.165788

Epoch 2
-------------------------------
loss: 2.177885  [    0/60000]
```

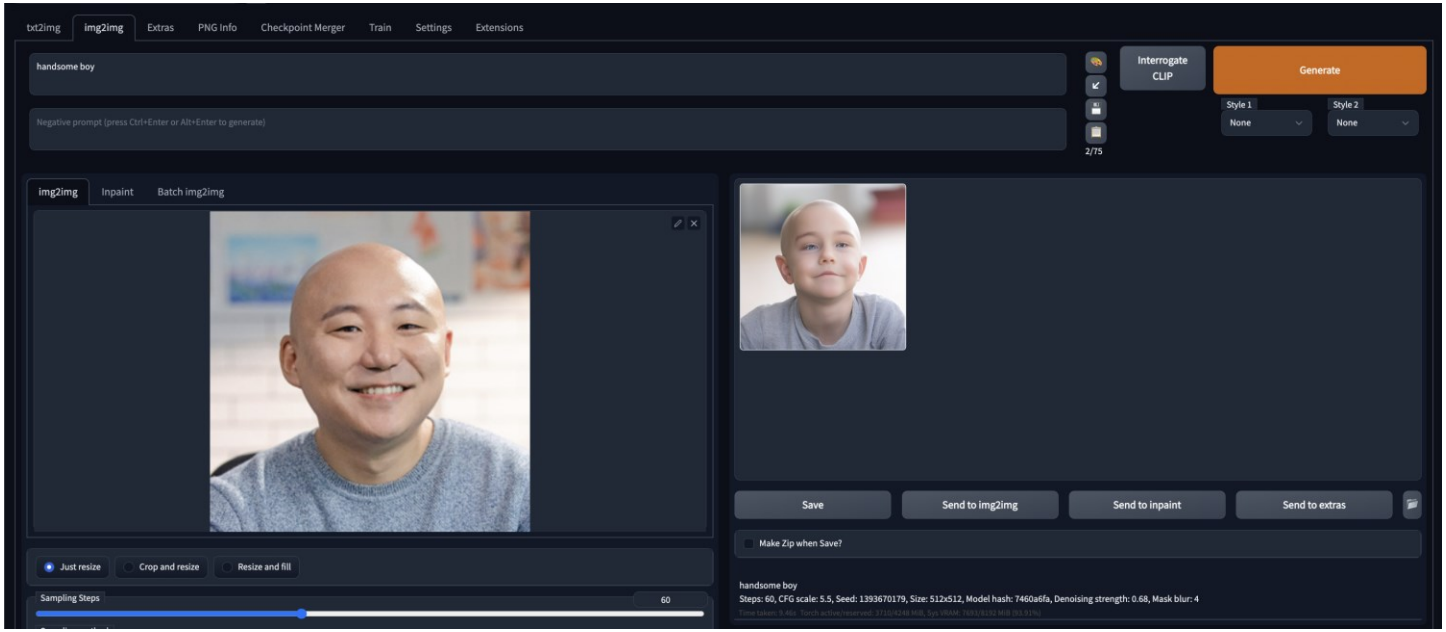
Deep learning on WSL2

TEST : Train model using cuda



Deep learning on WSL2

Demo : Stable diffusion web ui on WSL2



Conclusion – 이럴땐 하면 좋다
Do try this in this case

- 친구의 windows 컴퓨터에 달린 그래픽카드를 빌리고 싶은 경우
  If you want to borrow a graphics card from a friend's Windows computer
- Windows를 사용하면서 동시에 ubuntu환경에서 딥러닝 코드를 돌리고 싶은 경우
  if you want to run deep learning code in the ubuntu while using Windows
- 인터넷에서 갖다 쓰는 딥러닝 코드가 ubuntu에서만 도는데 재부팅하기 귀찮을 때
  My deep learning code get on the Internet runs only on ubuntu, but it is annoying to reboot
- 누가 쓰라고 칼로 협박하는 경우
  If someone threatens with a knife you to run deep learning on wsl

Conclusion – 왠만하면 하지마라
Don’t try this if you can

Why?

- Host ubuntu의 cuda 사용 경험과 차이가 날 수 있다.
  It may be different from Host ubuntu's cuda experience.
- Model inference server로 활용한다면 port-forwarding이 끊어지는 문제에 대비해야 할 수도 있다.
  If you use it as a model inference server, you may need to prepare for a problem where port-forwarding is disconnected.
- windows 쪽에서 grachic 작업을 같이 할 경우 예기치 못한 문제가 발생할 수 있다.
  Unexpected problems can occur when grachic work is done on the windows side.

Thank you!