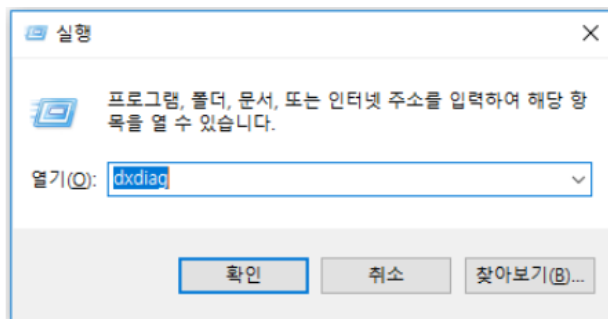
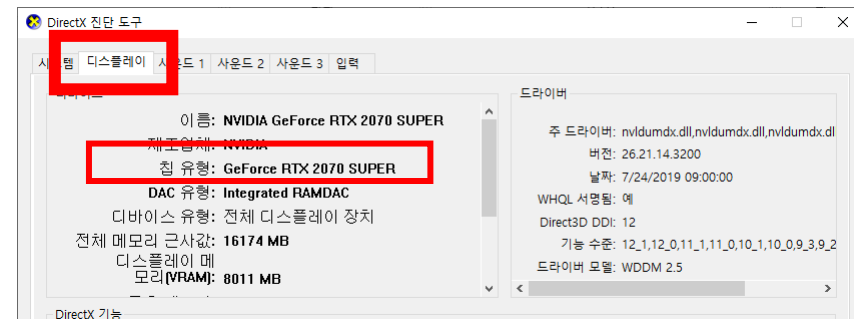


텐서플로우는 CPU 기반과 GPU 기반이 있음. GPU 텐서를 설치할수 있는지 NVIDIA 그래픽 카드를 확인해야함.

[1] win 로고 + R => dxdiag



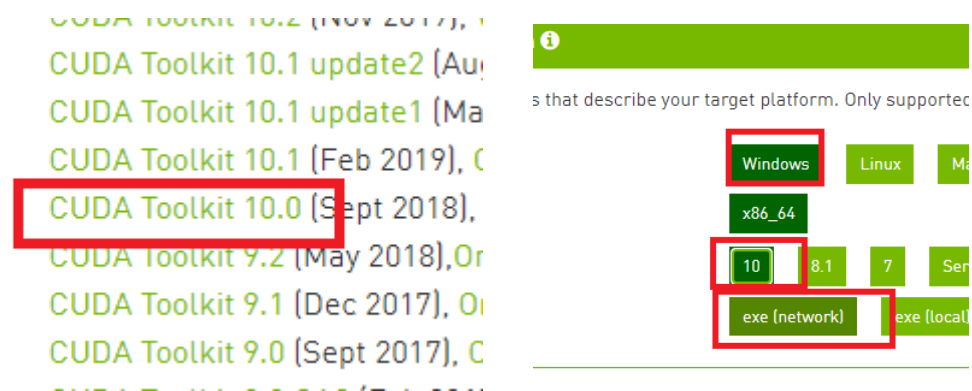
[2] 디스플레이 탭에서 확인



[3] CUDA Toolkit 다운로드

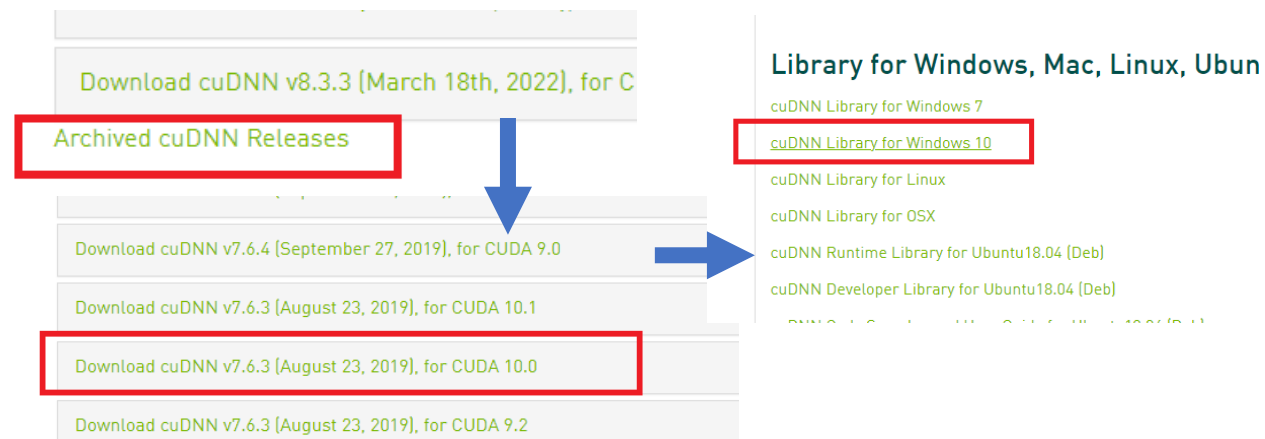
<https://developer.nvidia.com/cuda-toolkit-archive>

GPU 기반에서 텐서플로우 2.0을 먼저 설치하려면 Cuda 10.0
텐서플로우 1.0만 사용하려면 Cuda 9.0 설치함

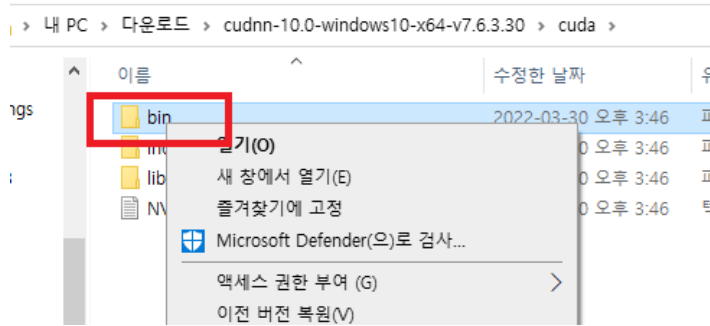


[4] cudnn다운로드 (cuda와 같은 버전으로 다운로드함) -회원가입로그인 필요

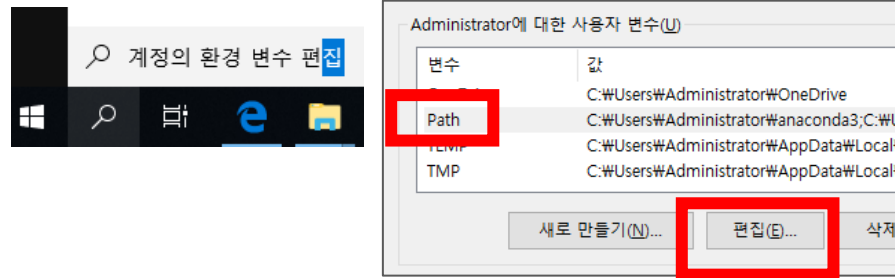
<https://developer.nvidia.com/rdp/cudnn-download>



[5] 다운로드 받은 cudnn 압축해제 (어느 위치든 무관) 후 폴더경로 복사

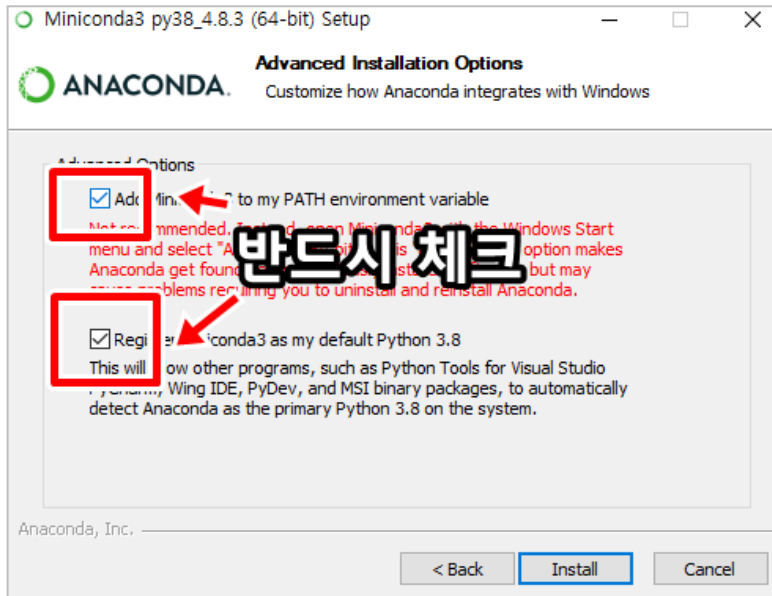


[6] 환경변수 편집 - 새로만들기에서 bin,lib,include추가

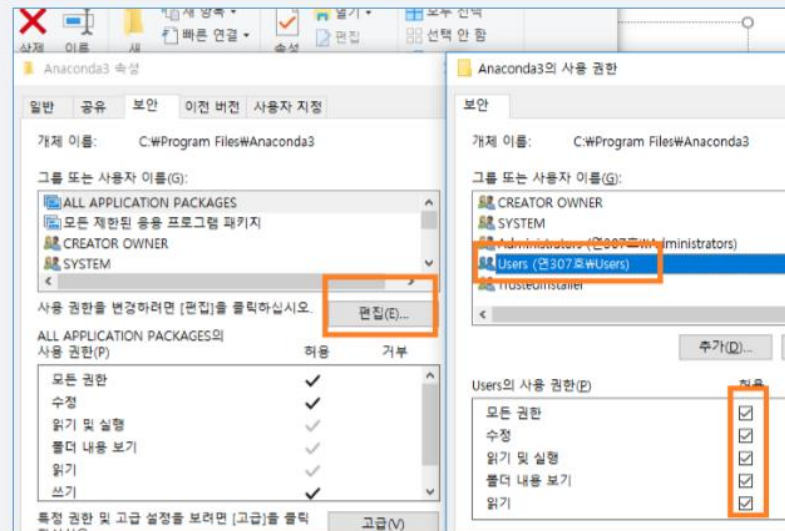


폴더명WcudaW**bin**
폴더명WcudaW**lib**
폴더명WcudaW**include**

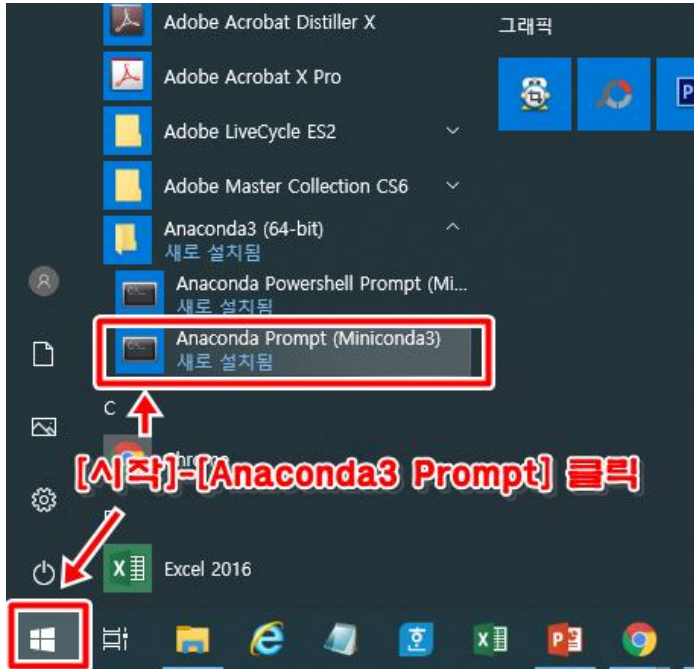
[7] 미니콘다 프로그램 설치 계정명이 한글일때는 반드시 all_user로 c드라이브에 설치
C드라이브 설치하면 추후 모듈 설치시 접근이 거부될수 있으므로 콘다 폴더의 접근 권한 해제해야함.
C드라이브 설치하면 탐색기에서 - 보기 - 자세히 - 숨긴항목 에 체크 해서 확인해야 할수도 있음.



■ 아나콘다 폴더권한 변경(C드라이브에 다운받은 경우만)



[8] 아나콘다 프롬프트에서 가상환경 설정



A

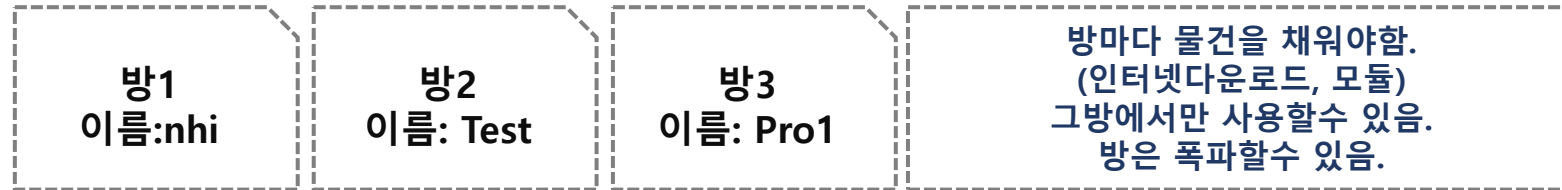
콘다프로그램 업데이트

```
(base) C:\Users\User> conda update conda  
Proceed ([y]/n)? y
```

B

가상환경 설정

가상환경: 작업별 공간임. 프로그램 다운로드(모듈) 및 저장등을 지정한 폴더에 모아놓는 공간임



```
(base) C:\Users\User> conda create -n tf_python=3.7
```

사용자 방이름

C

생성된 가상환경 확인

```
(base) C:\Users\User> conda info --envs
```

D

생성된 가상환경 삭제(진행하지 않음)

```
(base) C:\Users\User> conda remove --name tf --all
```

[9] 모듈 설치

E 가상환경 활성화

(base) c:\WUser\Wuser> **activate** **tf**

(tf) C:\WUsers\Wuser>

사용자 방이름

activate 방이름

(tf) C:\WUsers\Wuser> **conda install numpy**

activate 방이름

설치할 모듈이름(인터넷다운로드임)

```
Collecting package metadata (current_repodata.json): done  
Solving environment: |
```

Proceed ([y]/n)? y

```
(tf) C:\WUsers\Administrator> python  
Python 3.7.9 (default, Aug 14 2020, 12:23:15)  
Type "help", "copyright", "credits" or "license()" for more  
>>> tmp=[3,4]  
>>> mean(tmp)  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
NameError: name 'mean' is not defined  
>>> import numpy as np  
>>> np.mean(tmp)  
3.5  
>>>
```

python 실행

tmp=[3,4]
mean(tmp)

import numpy as np
np.mean(tmp)

*참고: 아나콘다 없이 파이썬만 설치되어 있을때

pip 최근자료로 업데이트 후 -> pip install 모듈명

(tf) C:\WUsers\Wuser> **pip install pip --upgrade pip**

(tf) C:\WUsers\Wuser> **pip install jupyter**

주로 사용하는 패키지 한번에 설치

(tf) c:\Wusers\Wuser> **conda install matplotlib seaborn pandas jupyter**

[10] 모듈 설치 텐서플로우 설치

모듈설치 (텐서플로우 2.0 GPU기반) cuda10.0 /gpu 기입하지않아도 됨

```
(tf) C:\Users\Wuser> conda install tensorflow-gpu==2.0
```

모듈설치 (텐서플로우 2.0 CPU 기반)

```
(tf) C:\Users\Wuser> conda install tensorflow-cpu
```

모듈설치 (텐서플로우 1.0 GPU기반)

```
(tf) C:\Users\Wuser> conda install tensorflow-gpu==1.15
```

모듈설치 (텐서플로우 1.0 GPU기반)

```
(tf) C:\Users\Wuser> conda install tensorflow==1.15
```

텐서2.0과
텐서1.0을
모두 사용하려면

가상환경을 작성하여 텐서버전
을 여러 개 설치할수 있음.

[11] 텐서플로우 GPU확인

텐서플로우 설치확인: 파이썬을 실행한뒤 텐서플로우를 import 하고 버전을 확인함.

(tf) C:\Users\user> **python**

```
>> import tensorflow as tf
>> tf.__version__
```

```
(tf) C:\Users\Administrator>python
Python 3.7.9 (default, Aug 31 2020, 17:10:11) [MSC
Type "help", "copyright", "credits" or "license" fo
>>> import tensorflow as tf
2020-09-10 00:56:14.243649: I tensorflow/stream_exe
dynamic library cudart64_101.dll
>>> tf.__version__
'2.1.0'
>>> _
```

텐서플로우 GPU 잘 설치되었는지 확인 (파이썬에서 실행함)

```
>> from tensorflow.python.client import device_lib
>> device_lib.list_local_devices()
```

**GPU 버전을 주피터노트북에서
사용시 반드시 한 개의 주피터
노트북만 실행해야함.**

```
incarnation: 4329773880141275865
, name: "/device:GPU:0"
device_type: "GPU"
memory_limit: 6588305899
locality {
  bus_id: 1
  links {
  }
}
incarnation: 1565322745674134630
physical_device_desc: "device: 0, name: GeForce RTX 2070 S
"
```

>> exit()로 종료