환경설정-Window 10

파이썬 설치

- https://www.python.org/downloads/
- 현재 3.7.4 버전 사용 중
- 명령프롬프트에서 사용하려면 환경변수 지정 해야함
  - 다운로드 할 때 기본값으로 설정 시 따로 환경변수 지정 필요 없음

아나콘다 설치

- https://www.anaconda.com/products/individual
- Python 3.7, 64-bit 설치

Visual Studio 설치

- Visual Studio 2019 다운로드
- https://visualstudio.microsoft.com/ko/downloads/

Table 2. Windows Compiler Support in CUDA 10.1

Compiler*	IDE	Native x86_64	Cross (x86_32 on x86_64)
MSVC Version 192x	Visual Studio 2019 16.x (Preview releases)	YES	NO
MSVC Version 191x	Visual Studio 2017 15.x (RTW and all updates)	YES	NO
MSVC Version 1900	Visual Studio 2015 14.0 (RTW and updates 1, 2, and 3)	YES	NO
	Visual Studio Community 2015	YES	NO
MSVC Version 1800	Visual Studio 2013 12.0	YES	YES
MSVC Version 1700	Visual Studio 2012 11.0	YES	YES

CUDA 설치1

1. Gpu 성능확인

https://developer.nvidia.com/cuda-gpus

GeForce GTX 960	5.2

2. CUDA version 확인

- CUDA Version은 아래에서 보이는 바와 같습니다.

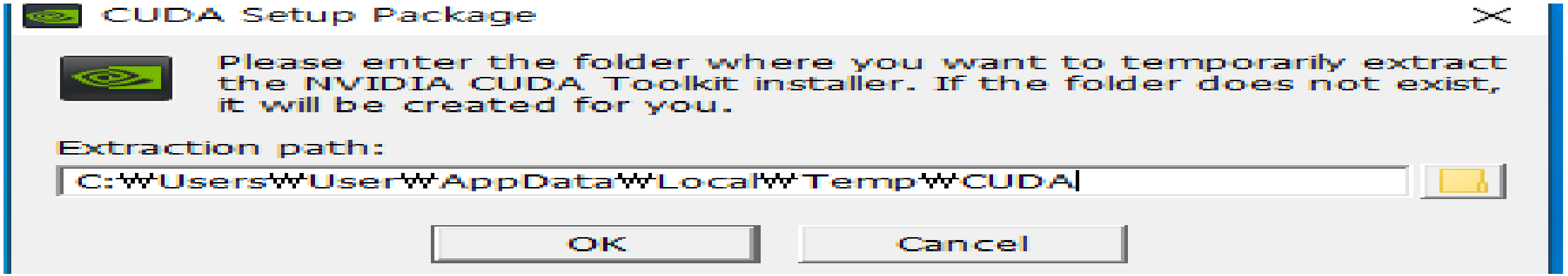
GPUs supported [edit]

Supported CUDA level of GPU and card. See also at Nvidia:

- CUDA SDK 1.0 support for compute capability 1.0 – 1.1 (Tesla)[20]
- CUDA SDK 1.1 support for compute capability 1.0 – 1.1+x (Tesla)
- CUDA SDK 2.0 support for compute capability 1.0 – 1.1+x (Tesla)
- CUDA SDK 2.1 – 2.3.1 support for compute capability 1.0 – 1.3 (Tesla)[21][22][23][24]
- CUDA SDK 3.0 – 3.1 support for compute capability 1.0 – 2.0 (Tesla, Fermi)[25][26]
- CUDA SDK 3.2 support for compute capability 1.0 – 2.1 (Tesla, Fermi)[27]
- CUDA SDK 4.0 – 4.2 support for compute capability 1.0 – 2.1+x (Tesla, Fermi, more?)
- CUDA SDK 5.0 – 5.5 support for compute capability 1.0 – 2.1+x (Tesla, Fermi, more?)
- CUDA SDK 6.0 support for compute capability 1.0 – 3.5 (Tesla, Fermi, Kepler)
- CUDA SDK 6.5 support for compute capability 1.1 – 5.x (Tesla, Fermi, Kepler, Maxwell). Last version with support for compute capability 1.x (Tesla)
- CUDA SDK 7.0 – 7.5 support for compute capability 2.0 – 5.x (Fermi, Kepler, Maxwell)
- CUDA SDK 8.0 support for compute capability 2.0 – 6.x (Fermi, Kepler, Maxwell, Pascal). Last version with support for compute capability 2.x (Fermi)
- CUDA SDK 9.0 – 9.2 support for compute capability 3.0 – 7.2 (Kepler, Maxwell, Pascal, Volta)
- CUDA SDK 10.0 – 10.1 support for compute capability 3.0 – 7.5 (Kepler, Maxwell, Pascal, Volta, Turing)

CUDA 설치2

- https://developer.nvidia.com/cuda-toolkit-archive
- Version 10.1 (Feb 2019) 다운로드
- 설치 경로 : C:₩Users₩User₩AppData₩Local₩Temp₩CUDA
- 제거방법 : C:Programfiles,(x86),프로그램 추가제거에서 NVIDIA 이름 다 제거 후 재부팅 후 재설치



cuDNN 설치

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

Download cuDNN v7.6.5 (November 5th, 2019), for CUDA 10.1 cuDNN Library for Windows 10

압축을 풀은 다음

3) 경로 이동

- 위 쿠다 파일의 bin, include, lib 안에 있는 것들을 아까 4-3에서 설치 한 CUDA의 경로에 알맞게 각각 넣어 줍니다.

- 즉, C:\Users\KMG\Downloads\cudnn-10.1-windows10-x64-v7.6.4.38\cuda 안에 bin, include, lib-x64의 하위에 있는 cudnn64_7.dll, cudnn.h, cudnn.lib 이 세개를 알맞는 C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.1 의 bin, include, lib-x64 안으로 넣어야 합니다.

OpenCV설치(version 3.4.0)

- https://sourceforge.net/projects/opencvlibrary/files/opencv-win/3.4.0/opencv-3.4.0-vc14_vc15.exe/download
- 압축 풀고 C:₩opencv로 옮기기
- 시스템변수 – Path – 새로만들기
  - C:₩opencv₩build₩x64₩vc14₩bin
- Visual studio 새 프로젝트 만들기 – 빈프로젝트 - cpp파일 생성 (ctrl+shift+A)
- 프로젝트 – 속성(참고 사이트에서 설정)

가상환경 만들기1

- 아나콘다 prompt 실행
- 경로에 폴더 만들기(test)
```
(base) C:₩Users₩User>
```
- 만들어준 test로 들어가기(cd test)
```
(base) C:₩Users₩User₩test>
```
- 가상환경 만들기
```
(base) conda create -n test python=3.7
```
- 가상환경 접속하기
```
(base) C:\projects\dl\test>activate test
```
- 라이브러리 설치하기

```
(test) conda install -n test tensorflow-gpu
```

```
(test) conda install -n test keras
```

```
(test) conda install -n test ipython notebook jupyter
```

```
(test) conda install -n test -c conda-forge lightgbm    # lightgbm
(test) conda install -n test -c conda-forge pydotplus   # pydotplus
(test) conda install -n test -c conda-forge pydot       # pydot
```

```
(test) conda install -n test numpy scipy matplotlib spyder pandas seaborn
scikit-learn h5py pillow matplotlib tqdm
```

```
(test) conda install -n test -c anaconda py-xgboost
```

```
(test) conda install -n test -c conda-forge catboost
```

참고 사이트 : https://theorydb.github.io/dev/2020/02/14/dev-dl-setting-local-python/

가상환경 만들기2

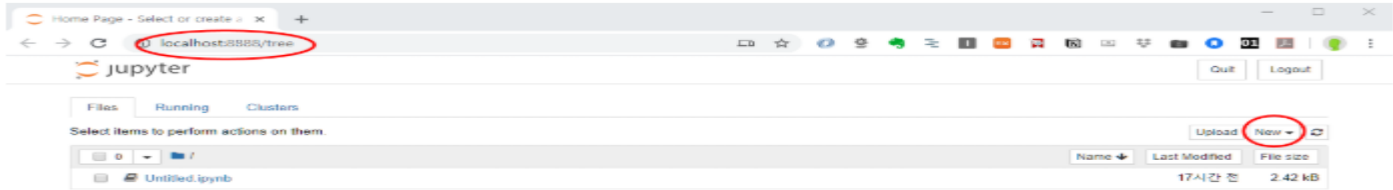
- 가상환경 kernel로 등록하기

```
python -m ipykernel install --user --name test
```

- 주피터 노트북 접속

```
(test) C:\projects\dl\test>jupyter notebook
```

- New-test 선택



- 버전 확인

```
In [1]: import tensorflow
        from tensorflow.python.client import device_lib

In [3]: print(device_lib.list_local_devices())

[name: "/device:CPU:0"
device_type: "CPU"
memory_limit: 268435456
locality {
}
incarnation: 3120156475130326648
, name: "/device:GPU:0"
device_type: "GPU"
memory_limit: 3167161548
locality {
  bus_id: 1
  links {
  }
}
incarnation: 17096388714978637142
physical_device_desc: "device: 0, name: GeForce GTX 960, pci bus id: 0000:01:00.0, compute capability: 5.2"
]
```

```
In [5]: import tensorflow
        from tensorflow import keras
        import pandas
        import sklearn
        import scipy
        import numpy
        import matplotlib
        import pydotplus
        import pydot
        import h5py

        print(tensorflow.__version__)
        print(keras.__version__)
        print(pandas.__version__)
        print(sklearn.__version__)
        print(scipy.__version__)
        print(numpy.__version__)
        print(matplotlib.__version__)
        print(h5py.__version__)

2.1.0
2.2.4-tf
1.0.5
0.23.1
1.5.0
1.18.5
3.2.2
2.10.0
```