AI – BOOK

-환경설정-

NVIDIA GPU, CPU CUDA 설치 방법

- 목차 -

1. 인공지능 정의 => 용규 & 재원 담당

2. 뉴럴 네트워크(2, 4진 분류), 이론( 구조, 함수 ) => all

3. RNN + LSTM ( 이론, 활성화 함수 ) => 명훈

-> 간단한 코드 예제 설명

4. CNN ( 이론 , 활성화 함수 ) => 우성

-> 간단한 코드 예제 설명

5. 출처

인공지능이란 무엇일까

인공지능은 일반적으로 인간 지능이 필요하거나 인간이 분석할 수 있는 범위를 벗어난 대규모 데이터를 포함하는 방식으로 추론, 학습 및 행동할 수 있는 컴퓨터와 머신을 빌드하는 과학 분야

텍스트, 스크린샷, 원, 폰트이(가) 표시된 사진

자동 생성된 설명

기계학습은 인간이 자연적으로 수행하는 학습 능력과 같은 기능을 컴퓨터에서 실현하려는 기술이나 방법이다.

인공지능 학습의 개념으로 예측(predict)은 경험을 바탕으로 지식을 습득하여 개인이 미래에 발생할 일을 생각하는 것이지만 인공지능은 경험이 아닌 데이터를 기반으로 ‘분석’하여 미래를 예견하는 예측(forecast)과는 개념이 다르며 딥러닝은 데이터를 통한 경험 바탕의 미래 예측을 수행한다.

위의 내용들을 기반하여 CPU 및 GPU를 활용한 인공지능 활용에 대해 소개해보려 한다.



Reference source : <https://cloud.google.com/learn/what-is-artificial-intelligence?hl=ko>

AI 사진 : <https://www.axis.com/blog/secure-insights-kr/2019/02/28/ai-video-surveillance/>

http://hsmpp.org/class/class\_view.asp?c\_code=la\_J040127

환경설정

GPU에 맞는 NVIDIA DRIVE 설치

드라이브 설치 전 Desktop/Laptop에 내장되어 있는 그래픽 카드를 확인 해야한다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

먼저 window 검색창에 ‘장치 관리자’를 검색 후

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명



디스플레이 어댑터를 보면 NVIDIA GeForce GTX 1080 그래픽 카드가 있는 것을 확인 할 수 있다.

또한 window 검색창에 ‘cmd’를 검색하거나 ‘명령 프롬프트’를 검색하여 실행한다.

텍스트, 소프트웨어, 컴퓨터 아이콘, 번호이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

위의 과정을 따라했다면 이제 명령어를 입력해 현재 자신이 어떤 GPU를 소유하고 있는지 알 수 있다.

Command -> nvidia-smi

위의 명령어를 입력하게 되면 현재 GPU 상태 모니터링, 프로세스 관리, 전력 및 클럭 설정 뿐만 아니라 정비 및 진단까지도 가능할 수 있게 한다. 텍스트, 스크린샷, 모노크롬이(가) 표시된 사진

자동 생성된 설명

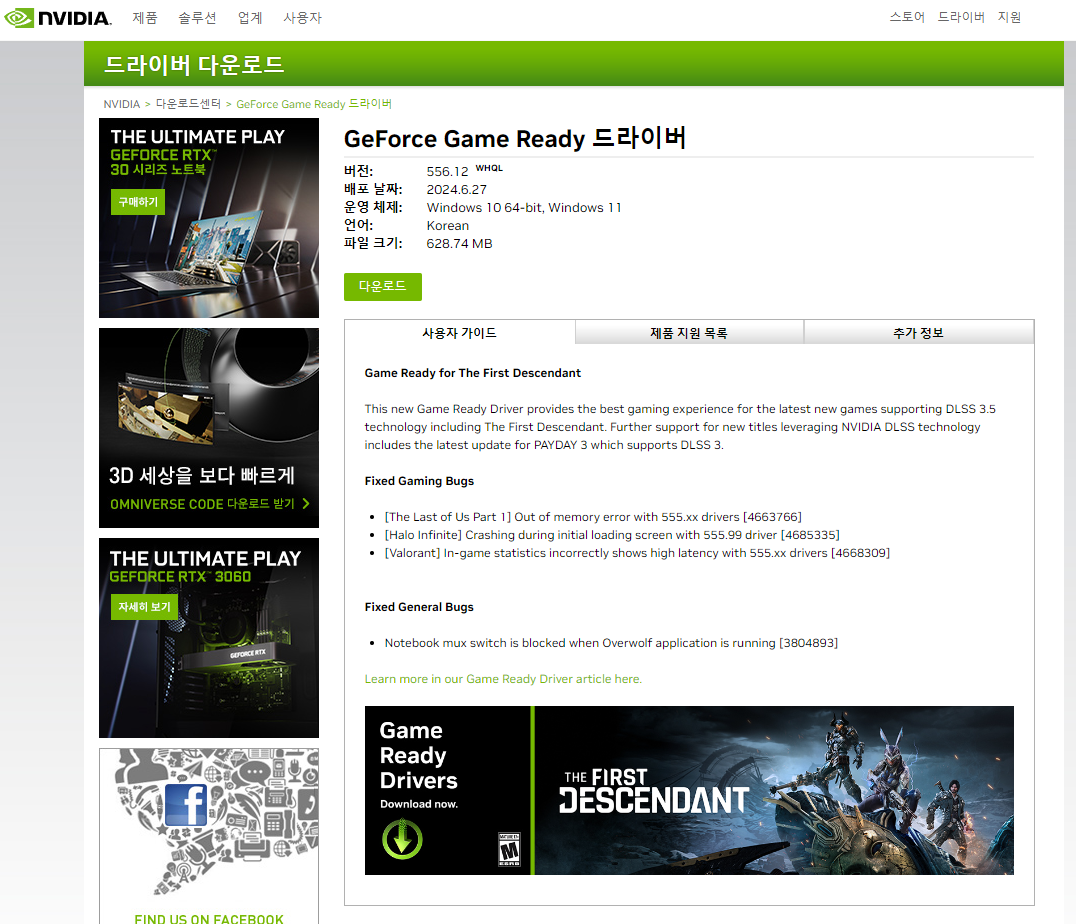
위의 사진은 명령어를 쳤을 경우 나오는 화면이다.

이제 GPU를 확인했으니 GPU에 맞는 드라이버를 설치할 차례이다.



NDIVIA DRIVE가 왜 필요할까요?

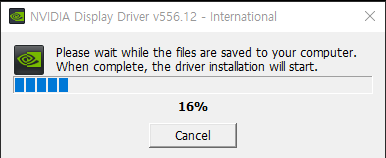
많은 연산을 CPU에서만 담당하게 되면 CPU가 해야 할 컴퓨터의 프로세스 관리 업무를 효율적으로 수행하지 못할 뿐더러 요구되는 연산 처리에 대한 소요시간이 증가하기 때문에 GPU를 활용해야 하고 이 때 필요한게 ‘GPU 드라이버’이다.



자신에게 맞는 드라이버를 선택하고 위의 사진과 같이 ‘다운로드’를 눌러 설치를 해준다.

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명 설치할 경로를 지정해주고

 경로 지정 후 다운로드

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명 위의 과정을 거치면 ‘NVIDIA 설치 프로그램’이 나오게 되고

‘동의 및 계속’을 누른 다음 순차적으로 진행한다.

텍스트, 스크린샷, 멀티미디어, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

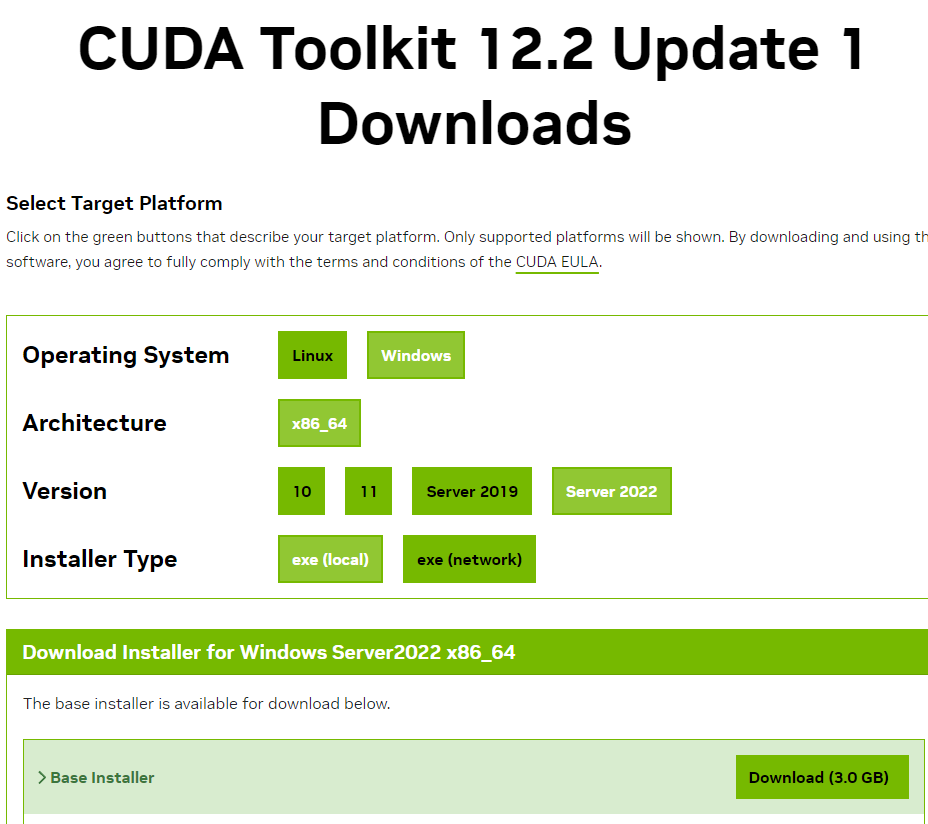
사진과 같이 설치 준비 중의 과정을 거쳐 드라이버 설치를 마치게 된다.



설치를 완료하고 난 뒤 체크박스는 사진처럼 체크하지 않아도 된다.

드라이버 다운 주소 :https://www.nvidia.com/ko-kr/geforce/drivers/

GPU를 사용하기 위해 CUDA 설치

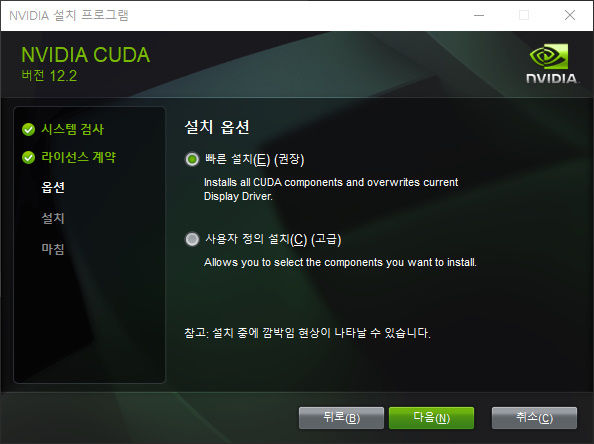


CUDA는 왜 필요한가요?

CUDA(Computed Unified Device Architecture) NVIDIA 사에서 개발한 GPU (Graphic Processing Unit)개발 툴루써 많은 양의 연산을 동시에 처리하기 위해 사용

자신의 컴퓨팅 환경에 맞게 옵션을 선택해 주고 Download를 눌러 설치해준다.

주소:https://developer.nvidia.com/cuda-12-2-0-download-archive?target\_os=Windows&target\_arch=x86\_64



드라이버 설치 때와 마찬가지로 순차적으로 진행해주면 된다.

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

(visual studio가 지원하는 버전이 아니기에 호환하는 버전을 찾을 수 없다 고로 버전에 맞는 것으로 설치해달라 라는 문구.

작성자 본인은 Pycharm을 사용할 것이기에 따로 설치를 하지 않을 것)

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 소프트웨어, 멀티미디어이(가) 표시된 사진

자동 생성된 설명

위의 과정들을 거치고 설치를 마치면 다음과 같은 창이 뜬다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어이(가) 표시된 사진

자동 생성된 설명

창이 뜨면 본인의 사정에 맞게 ‘지금 다시 시작’ 및 ‘나중에 다시 시작’을 눌러 적용해주면 된다.

이로써 CUDA의 설치가 완료 되었고 지정했던 경로에 맞게 잘 설치가 되었는지 확인해본다.

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v12.2

위의 경로를 따라 확인을 해보면 12.2 버전이 잘 설치되었음을 확인할 수 있다.

텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

Reference source :

<https://supersfel.tistory.com/entry/%ED%8C%8C%EC%9D%B4%EC%B0%B8%EC%97%90%EC%84%9C-GPU%EC%82%AC%EC%9A%A9>

<https://technfin.tistory.com/entry/%ED%8C%8C%EC%9D%B4%ED%86%A0%EC%B9%98Pytorch-%EC%84%A4%EC%B9%98%ED%95%98%EA%B8%B0-%EB%94%A5%EB%9F%AC%EB%8B%9D>

GPU를 사용하기 위해 cuDNN 설치



위의 사이트에 접속하여 자신이 설치한 cuda 버전에 맞게 cuDNN또한 설치를 해주어야 한다.

이 때 **cuDNN**이란?

딥 러닝 및 인공 신경망(ANN) 프레임워크를 가속화하기 위한 라이브러리

Reference source : https://developer.nvidia.com/rdp/cudnn-archive

텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

window환경에서 사용할 예정이기에 Window (Zip)을 설치하여

압축해제를 해준다.

텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 소프트웨어, 번호이(가) 표시된 사진

자동 생성된 설명

압축 해제 후 cuDNN 파일을 12.2버전이 설치된 CUDA 파일의 bin, include ,lib에 각 파일을 옮겨준다.

경로 : C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v12.2

텍스트, 스크린샷, 번호, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

(lib 파일을 옮기는 사진)

텍스트, 스크린샷, 번호, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

(include 파일을 옮기는 사진)

텍스트, 스크린샷, 번호, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

(bin 파일을 옮기는 사진)

위의 과정까지 문제없이 진행했다면 마지막으로 환경 변수 편집을 하면 끝이다.

먼저 window 검색 창에 ‘시스템 환경 변수 편집’을 검색하고 실행한다.

텍스트, 스크린샷, 소프트웨어, 번호이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 디스플레이, 번호이(가) 표시된 사진

자동 생성된 설명

환경 변수를 누르고

텍스트, 스크린샷, 디스플레이, 번호이(가) 표시된 사진

자동 생성된 설명

사용자 변수에 있는 Path를 눌러 환경변수를 편집해야한다.

텍스트, 전자제품, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명

Cuda toolkit을 설치한 경로에 있는 lib, include, bin디렉터리 경로를 ‘새로 만들기’를 누르고 추가해주면 된다.

그리고 ‘확인 버튼을 누르고 적용을 눌러 최종적으로 편집을 완료한다.

이로써 환경 변수 편집까지 하였으니 가상환경을 만들어 cuda를 사용할 수 있는 환경을 어느정도 구축했다.

Pycharm 설치 및 가상환경 구축

일반 버전은 일정 기간이 지나게되면 유료버전으로 바뀌게 되어 사용이 안된다.

텍스트, 폰트, 소프트웨어, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

하지만 Education으로 다운을 받게 되면 무료로 이용할 수가 있다.

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

‘Download’를 누르고 설치 경로를 지정한 다음 확인해보면

폰트, 그래픽, 상징, 텍스트이(가) 표시된 사진

자동 생성된 설명 이와 같이 설치가 완료된다.

위의 파일을 실행시키게 되면 다음 과정을 진행하게 된다.

텍스트, 스크린샷, 운영 체제, 폰트이(가) 표시된 사진

자동 생성된 설명

‘다음’버튼을 눌러 순차적 진행

텍스트, 스크린샷, 폰트, 디스플레이이(가) 표시된 사진

자동 생성된 설명

설치 경로를 지정하고 ‘다음’을 눌러준다.

텍스트, 스크린샷, 디스플레이, 번호이(가) 표시된 사진

자동 생성된 설명



1 : 바탕화면에 바로가기 생성

2 (필수권장) : 윈도우 명령 프롬프트에서 파이참에 접근 할 수 있도록 하며 자동으로 Path를 설정해주기 때문에 따로 설정을 하지 않아도 된다.

3 : 아무 폴더를 윈도우 탐색기에서 마우스 우클릭 시 “Open Folder as Project’옵션으로 표출된다.(폴더를 파이참에서 프로젝트로 바로 선택가능)

4 : .py파일과 연결

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

순차적으로 진행한 후 설치를 완료한다.

텍스트, 스크린샷, 소프트웨어, 운영 체제이(가) 표시된 사진

자동 생성된 설명

텍스트, 폰트, 스크린샷, 그래픽이(가) 표시된 사진

자동 생성된 설명 설치를 확인한 후 실행을 하게 되면 다음과 같은 문구가 나오게 되고

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

약관 동의를 체크하고 ‘계속’버튼을 눌러준다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

새로운 프로젝트를 만들기위해 ‘New Project’를 클릭한다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

0

Name : 생성하는 프로젝트의 이름을 설정

Location : 프로젝트의 경로 설정

Create Git Repository : 연결된 깃허브의 리포지토리를 생성

Create a welcome script : 프로젝트에 예제코드 삽입

Python Version : 프로젝트에 사용되는 파이썬 버전 설정

작성자 본인은 conda를 사용한 가상환경을 사용하기 때문에 miniconda를 설치해줍니다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

사전에 anaconda를 설치하지 않았기에 Pycharm에서 제공하는 miniconda3를 사용하려 한다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

노란색 박스 : .py등의 프로젝트 코드들이 들어가는 프로젝트의 위치

빨간 박스 : conda 환경

초록 박스 : Python버전설정 및 miniconda의 실행파일 위치(기본위치)

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

구성을 끝내고 기본 예제를 돌려보았을 때 잘 실행되는 것을 볼 수 있다.

위의 사진과 같이 Pycharm으로 가상환경을 만드는 방법이 있고 작성자 본인은 anaconda prompt를 사용하여 가상환경을 만들었다.

방법은 여러가지이며 그것들 중 하나를 선택해서 하면 되고

프롬프트로 가상환경을 만드는 방법은 다음과 같다.

‘명령 프롬프트’를 검색하여 실행했을 때와 마찬가지로 검색창에 anaconda prompt를 검색하여 실행한다.

텍스트, 소프트웨어, 컴퓨터 아이콘, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

위와 같이 miniconda의 프롬프트 창이 뜨게 되고 cmd 창에서

가상환경을 만들어보려 한다.

https://wooni-research.tistory.com/entry/Miniconda-Pycharm-%EA%B0%9C%EB%B0%9C%ED%99%98%EA%B2%BD-%EA%B5%AC%EC%B6%95%ED%95%98%EA%B8%B0

가상환경을 만들기 위한 명령어를 입력한다.

Command -> conda create –-name “프로젝트명”

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

(위의 명령어를 입력한 사진)

가상환경이 잘 만들어졌는지 확인하기 위해 명령어를 통해 확인한다.

그리고 현재 가상환경이 무엇이 있는지 확인하는 명령어를 통해 active상태로 만들 수 있는지 리스트를 확인해본다.

텍스트, 스크린샷, 폰트, 디스플레이이(가) 표시된 사진

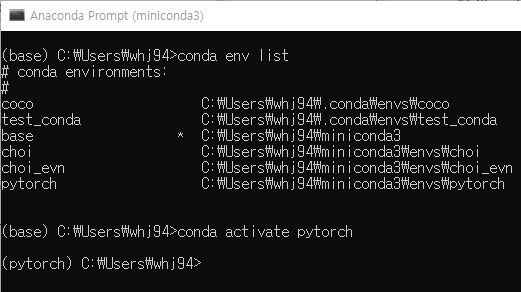
자동 생성된 설명

Command -> conda evn list (Conda 가상환경 목록들을 보여준다.)

방금 만들어낸 Pytorch가 잘 만들어졌음을 확인할 수 있다.

conda env list 명령어를 통해 현재 conda 가상환경 리스트를 알 수 있고 이 리스트들중 하나를 실행하여 다음 명령어를 통해 활성화 시킬 수 있다.

Command -> conda activate “가상환경명”



가상환경을 활성화 시키게 되면 맨 왼쪽 괄호에 활성화 하고자 하는 가상환경명이 기입되게 된다.

이때부터 설치하고 싶은 패키지나 하고자 하는 작업을 수행 할 수 있다.

여기까지 prompt로 가상환경을 만드는 과정이었으며 이제 Pycharm으로 돌아가 패키지 설치를 해보려한다.

먼저 Pycharm의 왼쪽 상단의 탭을 눌러주고 Open을 클릭한다.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

클릭 후 miniconda3\envs\pytorch경로를 확인하여 프로젝트를 열어준다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

(가상환경 목록들을 볼 수 있다.)

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Pytorch 파일을 열고 좌측 하단에 보면 5번째가 터미널 창이다.

터미널 창을 누르고 pytorch를 사용하기 위해 패키지를 설치해보려 한다.

Command -> python -m pip install troch torchvision torchaudio

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

파이썬 버전 및 cuda버전 호환성에 문제가 발생할 수 있지만 현재 설치한 cuda 12.2v은 웬만한 버전과 호환이 되고

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

파이썬 버전 또한 가장 최신의 것이어서 웬만한 패키지 버전과 호환이 가능하다.

이로써 문제없이 패키지 설치가 끝났고

https://pytorch.org/get-started/previous-versions/

위의 사이트를 참고하여 설치를 진행하였다.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

패키지 설치가 잘 되었는지는 pip list(파이썬 패키지 목록)를 통해 알 수 있다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

간단한 예제 코드를 사용하여 cuda를 사용할 수 있는지 실행해보았고 결과로써도 만족할만한 결과를 얻게 되었다.

PyTorch를 사용하여 간단한 인공 신경망을 만들어 MNIST 데이터셋에 대해 학습하고 평가하는 코드를 실행시켜보았다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Cuda를 사용하여 인공 신경망을 만들고 학습하고 평가를 하고 있는지 간단한 코드를 구성해 보았고 이로써 cuda를 사용하여 Pytorch 인공 신경망을 구성하기 위한 환경설정을 마무리한다.

작업도중 나타날 수 있는 변수

**1. conda에서 가상환경을 만들었는데 적용되지 않았을 경우**

텍스트, 멀티미디어 소프트웨어, 소프트웨어, 스크린샷이(가) 표시된 사진

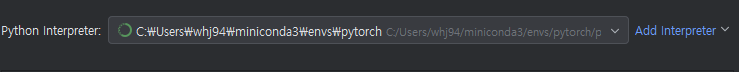
자동 생성된 설명

-> 우측 상단에 보이는 톱니바퀴를 눌러준다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

-> setting(설정)을 눌러준다.



-> Add Interpreter을 눌러준다.

텍스트, 소프트웨어, 멀티미디어 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명

-> Conda Environment를 클릭하고 Conda Executable을 연동할 수 있게 miniconda3 하위에 있는 conda.exe실행파일이 제대로 되어있는지 확인한다.

**2. Numpy 버전이 맞지 않아 버전을 낮추어 달라는 경우.**

-> pip uninstall numpy를 한 후 재설치를 하는 것이 제일 좋은 방법이다.

어떤 버전을 써야 하는지 헷갈려 이것저것 해보다가 엇갈려서 시간을 낭비하지 말고

<https://pytorch.org/get-started/previous-versions/>

위의 사이트를 참고하여 어떤 버전을 사용해야 하는지 한눈에 확인할 수 있다.

**3. CUDA Toolkit을 설치했는데 적용이 안되는 현상**

-> 현재 Pycharm에서 지원하는 cuda버전은 11.3이고 작성자 본인이 설치한 버전은 12.2이다. 11.3 cuda toolkit을 재설치하여 적용할 수 있지만 12.2와 11.3의 버전이 호환이 되어 cuda를 사용하는데에 있어서 문제가 없기에 재설치를 하지 않았다.

또한 앞서 설명했던 환경 변수 편집이나 cuDNN파일을 옮겼는지 여부를 확인해주고 python버전과 numpy, pytorch등의 버전을 잘 확인하여 시간이 낭비되는 일이 없도록 진행할 수 있다.