

Machine Learning Tutorial(인공지능의 요소와 도구)

이인호 (표준과학연구원, ihlee@kriss.re.kr)

인공지능의 역사와 필요성을 가장 먼저 소개한다. 그 다음 인공지능의 요소와 도구들을 골고루 학습하고 실습한다. 파이썬 언어를 활용하여 다양한 응용 사례들을 학습한다. 수강자분들께서는 참고자료 위치로 표시된 아래의 URL 에 게시된 아나콘다와 그 밖의 각종 패키지를 설치한 노트북을 지참해 주시기 바랍니다.

<http://incredible.egloos.com/7482936>

참고문헌: <http://incredible.egloos.com/7133499>

<http://incredible.egloos.com/7303943>

<http://incredible.egloos.com/7515137>

강의 사전 준비물

Anaconda3, tensorflow, keras, xgboost, lightgbm, catboost, bayesian-optimization, mlens, deslib, graphviz, pydot

아나콘다(Anaconda)는 **패키지 관리**와 **디플로이**를 단순케 할 목적으로 **과학 계산**(데이터 과학, 기계 학습 애플리케이션, 대규모 데이터 처리, **예측 분석** 등)을 위해 **파이썬**과 **R** 프로그래밍 언어의 **자유-오픈 소스** 배포판이다. 패키지 버전들은 **패키지 관리 시스템 conda**를 통해 관리된다.

Anaconda3 설치 방법 (윈도우즈 10 환경 기준) [맥킨토시, 리눅스도 유사하게 진행할 수 있음.]:

Anaconda | Individual Edition "URL

클릭 <https://www.anaconda.com/products/individual>"

→ 접속 후 "download" 문구 클릭

"64-Bit Graphical Installer (457 MB)" 문구를 확인하여 파일을 다운로드 받는다.

→ 내 PC 에 저장된 다운로드 파일을 클릭하여 실행시킨다. Anaconda3 설치를 할 때 아래 두가지 항목에 주의 한다. [두가지 선택 버튼들을 아래와 같이 한다.]

~~all-users 를 대상으로 사용할 수 있게 설치한다.~~

~~경로(path)가 잡히도록 설치한다. 기본 세팅을 유지한다.~~

Anaconda3 설치 방법 (윈도우즈 10 환경 기준, 2021.01.12, 정보전산실로부터의 Anaconda 사용 약관 변경 확인 후 재설치 방법):

③ 제어판 → 프로그램 및 기능 → 기존 Anaconda 제거

④ C 드라이브 → 사용자(Users) → ADMIN(사용자 이름 폴더) → ".conda" 폴더에 있는 것들 모두 삭제.

⑤ 원내 게시물 "원내 AI 교육 안내 (실습 코드, 재설치 파일 추가)"글에 업로드 된 "Anaconda3-2020.02-Windows-x86_64.zip"다운로드 및 압축 풀기

⑥ Anaconda3-2020.02-Windows-x86_64.exe 파일 실행/설치

⑦ 설치는 Default 로 설정되어 있는 대로 계속 설치하면 됨. (즉, 설치 중에, Selection Installation Type 에서 "Just Me (recommended)" 선택 Advanced Installation Options 에서 "Register Anaconda3 as my default Python 3.7" 선택) 설치 완료 후, 나머지 개발 환경 구축 과정은 기존 안내와 동일.

Jupyter notebook에서 kernel error 가 발생할 때 패키지 추가 설치가 필요합니다.: `pip install pywin32`

주피터 노트북 connection fail 에러가 발생할 경우 (웹브라우저 뜰 때 error 발생, 프로그램 실행이 안 될 때):

```
conda install -c conda-forge pywin32
```

```
conda install -c anaconda jupyter_client
```

```
conda install -c conda-forge jupyter_core
```

또는 아래의 두 명령어를 실행한다.

```
pip uninstall tornado
```

```
pip install tornado==5.1.1
```

Cf. Anaconda 를 cmd 상 다른 경로에서 실행시키고 싶은 경우엔 Anaconda 를 PATH 환경 변수에 추가하면 됩니다. 하지만 Anaconda 측에서는 PATH 환경 변수에 다른 소프트웨어를 방해할 수 있으므로 Anaconda 를 PATH 에 추가하지 않을 것을 권장하고 있습니다. (어차피 보통 Anaconda 를 Anaconda Prompt 에서 실행시키므로 환경 변수에 추가하지 않아도 됩니다.)

Anaconda3 설치 직후(가상환경 만들고 그곳에 필요한 패키지 설치한다.):
시작 → Anaconda Prompt (Anaconda3) 클릭한다. → 생성된 검은색 창 Anaconda Prompt (Anaconda3)에서 몇 가지 종류의 명령어들을 넣어 준다.
이 명령어들은 이름을 가지는 가상환경 생성 그리고 그 가상환경에 꼭 필요한 패키지 설치와 관련된 것들이다. 자신이 원하는 가상환경, "testAI"라는 이름으로 가정한다. [고유의 이름을 가지는 가상환경 생성 명령어] [여기에서는 특별히 AI 학습에 필요한 것들을 의미한다.] 입력해야 하는 명령어들의 순서는 아래와 같다.
이미 생성되어 있는 기본(base) 가상환경에서 출발하여 새로운 가상환경을 만든다. [모든 것을 base 에 설치할 수도 있다.] [cond install 또는 pip install 둘 중 하나로 패키지를 설치한다.]

```
conda create -n testAI
```

```
conda activate testAI
```

```
conda install tensorflow
pip install keras
pip install xgboost
pip install lightgbm
pip install catboost
pip install bayesian-optimization
pip install scikit-optimize
pip install mlens
pip install deslib
pip install graphviz
pip install pydot
pip install jupyter notebook
pip install nbconvert
```

```
{cuda를 이용할 경우, NVIDIA GPU 장비가 있는 경우.}
{
conda install -c anaconda cudatoolkit
conda install -c anaconda cudnn
conda install -c anaconda tensorflow-gpu

conda list cudatoolkit
conda list cudnn

nvidia-smi
nvcc --version
}
```

```
conda install -c conda-forge matplotlib
pip install opencv-python
conda install --channel conda-forge pymatgen
pip install pyxtal
pip install matminer
```

이미 설치된 패키지에 대해서는 기존 설치정보를 출력하고 설치작업을 자동으로 건너뛴다. 아래의 세 줄의 명령어들은 주피터 노트북과 관련된 명령어들이다.
[위의 명령어들에 연속해서 실행하는 것들이다.] [설치작업의 연속이다.]

```
pip install ipykernel
python -m ipykernel install --user --name=testAI
=부호 뒤에 testAI를 정확히 붙여서 작성해야 한다.
```

```
jupyter notebook --generate-config
```

[여기에서 가상환경 구성 작업이 종료된다. 사용자 폴더가 기준점이 된다.]

아래의 명령어를 입력하면 [Anaconda Prompt \(Anaconda3\)](#) 창에서 나올 수 있다. 즉, Anaconda 실행을 멈출 수 있다.

```
exit
```

[Anaconda Prompt \(Anaconda3\)](#) 창에서 나오기 전에 설치된 패키지를 볼 수 있는 명령어:

```
conda env list
```

현재 보유하고 있는 가상환경들의 목록을 확인할 수 있다. 명령어 exit를 이용하여 [Anaconda Prompt \(Anaconda3\)](#)에서 나올 수 있다.

Anaconda 에서 가상환경을 활성화하고 **jupyter notebook** 를 실행시킬 수 있다. 새롭게 Anaconda 에 들어가고자 할 때, 즉, 이번에는 Anaconda 설치가 아니고 실습을 진행하고자 한다. **jupyter notebook** 으로 프로그램을 만들 때, 인공지능 실습을 할 때:

시작 → **Anaconda Prompt (Anaconda3)** 클릭한다.

→

```
conda activate testAI
```

→

jupyter notebook

검은색 창과 브라우저가 동시에 생성된다. 웹브라우저 창은 창단기로 종료한다. 검은색 창에서 **jupyter notebook** 실행을 중지하는 방법: "**control + C**"를 입력하는 것이다.

jupyter notebook 실행 후, 아래의 파이썬 코드를 이용해서 텐서플로우의 설치 여부를 확인할 수 있다. 웹 브라우저 오른쪽 상단 → new → **testAI** 선택한다. 아래의 두 줄을 박스안에 타이핑하고, 위에 보이는 실행버튼을 누른다. [Run 버튼을 클릭해 준다.] 좌우로 길게 늘어진, 긴 직사각형 창(셀)에 아래 두 줄을 타이핑해서 넣어준다. [code 양식이다. markdown 양식이 아니다.]

```
import keras
print(keras.__version__)
2.4.3
import sklearn
print(sklearn.__version__)
0.23.2
import tensorflow as tf
print(tf.__version__)
2.3.0
```

testAI.zip 파일은 사용자 디렉토리 (예를 들어, C:/Users/ADMIN) 아래에서, 압축을 풀어놓으면 되는 파일입니다. (내 PC → 로컬디스크 (C:) → 사용자(또는 Users) → ADMIN(또는 각자 pc 관리자 이름))

위에서 설치한 graphviz 프로그램의 경우, 설치 후, 설치된 경로를 찾아 내야 한다. graphviz 는 인공지능망 구성을 나타내는 그림 그리기, decision tree 를 표시하는 그림 그리기에 사용된다. 반드시 설치해야 하는 것은 아니다. 이 경로를 환경변수 설정에 추가해주어야 한다.

1. conda install graphviz

2. conda install python-graphviz

graphviz 가 설치된 곳으로 이동하여, 경로를 복사한다. 환경변수의 PATH 에 추가하는 것이 목표이다. 시스템 어디에서나 dot.exe 를 실행할 수 있도록 합니다. 예를 들어, 아래와 같은 문구를 환경변수 PATH 에 추가한다.

C:\Program Files (x86)\Graphviz2.38\bin

(

위 설치만으로도 graphviz 는 문제 없이 실행될 것임. 하지만 만약 문제가 발생한다면, 아래와 같이 해결 시도:

① https://graphviz.gitlab.io/_pages/Download/windows/graphviz-2.38.msi

URL 을 통해 graphviz-2.38.msi 다운로드 및 설치

② 디폴트 설치 경로인, C:\Program Files (x86)\Graphviz2.38 에 설치

③ 환경변수 설정: 내 PC → 마우스 우 클릭 → 속성 → 고급 시스템 설정 → 환경 변수 → 시스템 변수 섹션에서 Path 환경 변수를 찾아 "편집" 클릭 → 찾아보기 → 위 graphviz 설치 경로의 bin 폴더 설정 후 확인 (C:\Program Files (x86)\Graphviz2.38\bin)

)

환경변수 설정:

내 PC → 마우스 우 클릭 → 속성 → 고급 시스템 설정 → 환경 변수 → 환경 변수를 누른다. 시스템 변수 섹션에서 PATH 환경 변수를 찾아 선택한다. 편집을 누른다. PATH 환경 변수가 존재하지 않을 경우 새로 만들기를 누른다.

jupyter notebook 작업관련 디폴트 작업위치: 사용자(유저) 디렉토리이다. 이곳이 기준점이 된다. 예를 들면 C:/Users/ihlee
이 디렉토리에 "testAI"라는 폴더를 만들 수 있다. [이것이 가장 간단한 방법이다.]
새롭게 생성된 인터넷 웹브라우저로 만들어진 jupyter notebook 에서, 우측 상단 → new → testAI [Markdown 타입은 Markdown 으로 셀의 내용을 작성할 수 있다. 코드로 실행되지는 않으며, 수식을 작성할 수 있다. 수식은 MathJax 에 의해 지원된다.]
수식이 레이텍 수준에서 적히기 때문에 계산결과를 수식과 함께 정리를 할 수 있다. jupyter notebook 이 진정한 공책이 되는 것이다. 따라서, 완성된 jupyter notebook 파일(확장자 *.ipynb)은 매우 다양한 양식으로 저장할 수 있다. [File → Download 에서 다양한 저장방식을 선택할 수 있다.] [슬라이드, html, pdf, latex, python 스크립트 형식으로 곧바로 변환할 수 있다.] jupyter notebook 을 잘 활용하면, 무료 CPU/GPU/TPU 를 제공하는 구글 코랩에서도 별 문제없이 적응할 수 있다.

jupyter notebook 사용법 [간략한 정리]

- * Shift + Enter 키를 누르면 셀이 실행되고 커서가 다음 셀로 이동한다.
- * Enter 키를 누르면 다시 편집상태로 돌아온다. [편집 모드]
- * ESC 키를 누르고 [명령어 모드]
- * a 키를 누르면 위에 셀이 추가된다
- * b 키를 누르면 아래에 셀이 추가된다.
- * dd 키를 누르면 셀이 삭제된다.
- * m 키를 누르면 문서 셀로 변경된다.
- * y 키를 누르면 코드 셀로 변경된다.

h 단축키 설명

스톱 버튼으로 실행 중지

kernel → clear output [출력 결과들만 제거한다.]

개발환경은 jupyter notebook 이 많이 활용된다. 하지만, jupyter notebook 대신에 spyder 를 사용하여 프로그램을 개발할 수 있다. spyder 를 선호하는 분들도 있다. 물론,

사용자가 하나를 선택하는 것이다. **jupyter notebook** 으로 프로그램을 개발했다면, 이것을 아주 손쉽게 파이썬 스크립트 형식으로 변환할 수 있다. File → Download → Python (*.py) 로 “내려받으면” PC 다운로드 폴더에 *.py 형식으로 파일이 저장된다. 이것을 복사하면 쉽게 spyder 에서 사용할 수 있다.

GPU 계산을 원하는 경우:

1. Anaconda 설치

- Python 3.7 version
- Anaconda Prompt 실행
- VirtualEnv 설치: `pip install virtualenv`
- VirtualEnv 생성: `conda create -n venv pip python=3.7`
 - VirtualEnv 진입: `conda activate venv`
 - VirtualEnv 해제: `conda deactivate`

2. CUDA 및 CuDNN 설치

- 현재 기준 (2019.07.14) Tensorflow (>= ver 1.13)은 CUDA 10 만 지원
 - (<https://developer.nvidia.com/cuda-toolkit-archive>)
 - CUDA 10 기준 CuDNN 설치
 - (<https://developer.nvidia.com/cudnn>)
 - 설치 후 PATH 설정 (윈도우 커맨드 창에 입력)
- 1)

```
SET PATH=C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\bin;%PATH%
```

2)

```
SET PATH=C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\extras\CUPTI\libx64;%PATH%
```

3)

```
SET PATH=C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\include;%PATH%
```

4) SET PATH=C:\tools\cuda\bin;%PATH%

* 위 4) 의 경우 CuDNN 압축 푼 폴더를 해당 디렉토리에 복사 후 PATH 등록

3. Tensorflow-gpu 설치

- VirtualEnv 진입 상태에서,
- `pip install tensorflow-gpu`





4. Keras 설치

- VirtualEnv 진입 상태에서,
- pip install keras

Path 설정대신에 아래 그림과 같은 절차를 밟을 수도 있다.

다운로드한 cuDNN 파일의 압축을 풀고, 폴더 내부에 있는 4 개 항목을 다른 곳으로 복사하는 방법이 있다.

bin, include, lib, NVIDIA_SLA_cuDNN_Support

 bin	파일 폴더
 include	파일 폴더
 lib	파일 폴더
 NVIDIA_SLA_cuDNN_Support	텍스트 문서

{cuda 를 이용할 경우, NVIDIA GPU 장비가 있는 경우.}

```
{
conda install -c anaconda cudatoolkit
conda install -c anaconda cudnn
conda install -c anaconda tensorflow-gpu
```

```
conda list cudatoolkit
conda list cudnn
```

```
nvidia-smi
nvcc --version
}
```

쿠다 툴킷 폴더에 cuDNN 내용물들을 모두 복사한다. 즉, cuda toolkit 설치, cuDNN 설치. 마지막으로 cuDNN 파일들을 cuda toolkit 폴더로 복사해버린다. tensorflow-gpu를 설치한다. 아나콘다 프롬프트에서 nvcc -version을 실행한다. [쿠다 툴킷 설치, cuDNN 폴더 복사, tensorflow-gpu 설치]

압축해제한 cuDNN 폴더에 있는 모든 파일을 복사해서 C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.3 폴더에 붙여넣기 해줍니다. 모두 진행하기로 덮어쓰기 작업을 진행합니다. 윈도우 시작, 우클릭 → 장치 관리자 → 디스플레이 어댑터 클릭 → NVIDIA GeForce GTX 750 Ti

장치 관리자로 디스플레이 어댑터 장치를 확인함. [→ 장치 확인 방법] NVIDIA 그래픽 장비가 있어야 함. PC 윈도우즈, Mac 은 NVIDIA 장치를 사용하지 않음. 따라서, GPU 가속화를 할 수 없음.

[주피터노트북에서 GPU 장치 확인 방법, 프로그램에서 확인하는 방법]

```
from tensorflow.python.client import device_lib
device_lib.list_local_devices()
import tensorflow as tf
tf.test.is_gpu_available()
tf.test.gpu_device_name()
tf.config.experimental.list_physical_devices(device_type='GPU')
gpus = tf.config.experimental.list_physical_devices('GPU')
```

```

for gpu in gpus:
print("Name:", gpu.name, " Type:", gpu.device_type)
# For tensorflow 2:
print("Num GPUs Available: ", len(tf.config.list_physical_devices('GPU')))

#Use only CPU
if False:
import os
os.environ["CUDA_VISIBLE_DEVICES"]="-1"

```

Linux/MacOS에서 Anaconda3 설치하기

Graphical installer를 다운받는다.

윈도우즈에서와 마찬가지로 설치한다.

터미널을 연다.

```
conda --version
```

만약 Linux/MacOS에서 conda 명령어가 먹히지 않으면

export PATH=~/anaconda/bin:\$PATH 로 anaconda를 경로에 추가한다. .bashrc 등에 넣어준다.

```
echo 'export PATH=~/anaconda/bin:$PATH' >> ~/.bashrc
```

```
conda list
```

```
conda update conda
```

또는 sudo conda update conda

conda create 명령어를 사용할 수 있으면 윈도우즈에서처럼 환경을 만들 수 있다.

차이점: conda activate 대신에 source activate를 사용한다.

그 다음부터는 pip, conda 를 이용하여 패키지들을 설치한다.

윈도우즈에서 Cuda 사용률을 확인하는 방법 (Cuda toolkit → cuDNN 순서대로 모두 설치한 후, 응용 프로그램 실행할 때):

1) 작업관리자 (Ctrl + Shift + ESC) 또는 Ctrl + alt + del

2) 성능 탭 → GPU

3) 선택 항목(↓표시) 중 1 개를 Cuda 로 변경하여 설정함. 실시간 그래프로 Cuda 사용률을 직접 확인할 수 있다. 또한, GPU 메모리 사용량을 직접 확인할 수 있다.

[Cuda toolkit → cuDNN → tensorflow-gpu 설치이후, 주피터 노트북에서 응용 프로그램 실행하면서 GPU 장치의 사용을 실제로 확인하는 방법]

'Windows 로고 + R'

→

실행창에 아래 명령어 입력

dxdiag

→

'DirectX 진단 도구' 화면 → 여기서 '디스플레이 1' 메뉴를 선택

→

제조업체가 NVIDIA 인지 확인함. [NVIDIA 제조사가 아니면 과학기술용 계산, 인공지능 프로그램의 가속화 기능을 사용할 사용하기가 어렵다.]

```
-----  
from tensorflow.python.client import device_lib  
print(device_lib.list_local_devices())  
from keras import backend as K  
K.tensorflow_backend._get_available_gpus()  
-----
```

최초작성일자: 2020년 12월 28일

최종수정일자: 2022년 5월 26일

작성자: 이인호 (ihlee@kriss.re.kr)