

AI융합전공 과목 공통

01 실습 도구 소개

AI융합전공 과목 공통

01

실습 도구 소개

1.0 실습 환경 구축 방법 옵션들

1.1 AWS

1.2 Colab

1.3 머신러닝 도구들

1.4 내 PC 에 실습 환경 구축하기 (Optional)

실습 환경 구축 방법 옵션들

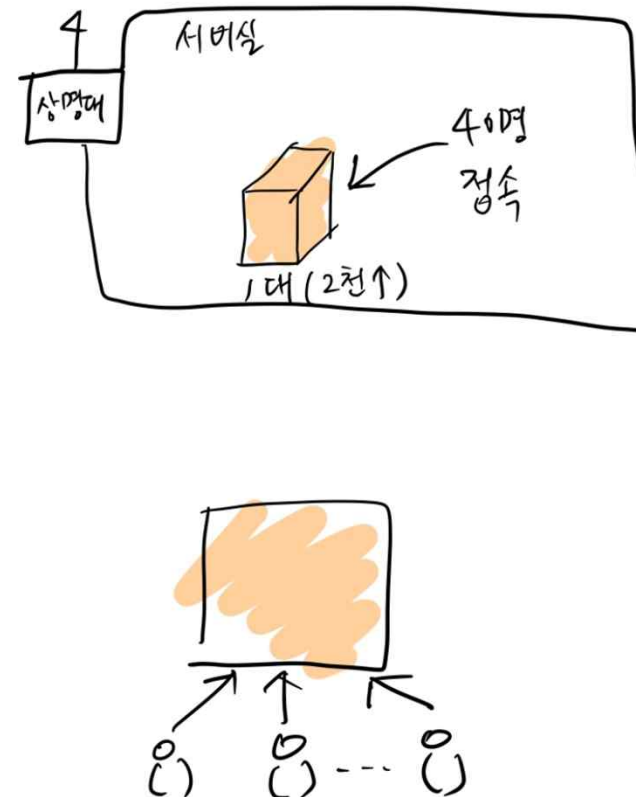
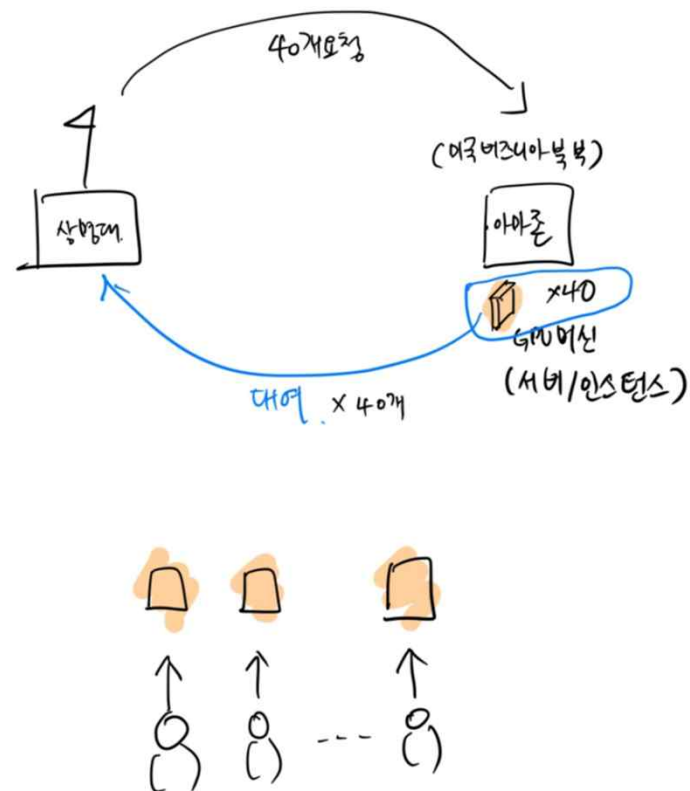
- 기계학습 및 심층학습 관련 내용 구현을 위해서는 실습 환경이 필요함
- (HW)GPU 머신: Nvidia 의 GTX(RTX), Titan, Tesla 계열 GPU 가 장착된 서버
- (SW)Ubuntu, CUDA, Jupyter notebook, 인공지능 프레임웍(PyTorch or TensorFlow & Keras), 각종 라이브러리들



© Nvidia

GPU 머신 사용 방식 비교

- 클라우드 서비스(대여) vs. GPU 머신(구입)



클라우드 서비스 vs. GPU 머신 vs. 학생 PC

	클라우드 서비스 (AWS SageMaker, Google Colab)	GPU 머신(Nvidia 사의 GTX(RTX), Titan, Tesla)	학생 PC (학습용 GPU 없는 경우)
과목 실습 목적	O	O	O
사업화 목적	O	O	X
비용	비쌈. 시간당 과금	비쌈	저렴
GPU 가속	선택가능	O	X
라이브러리 설치	불필요	필요	필요
셋팅하는 사람	AWS	조교	학생
접속시간	제한 있음	무제한	무제한
기타	서비스 명칭: AWS SageMaker 버즈니아 북부 ml.t2.medium. 예산: 한 학기 약 400만원	-	설치: 가상환경에서 Ubuntu, Jupyter Notebook, 각종 라이브러리

과목 수강하는
동안 사용

GPU가속이 필요한 작업

자연어처리

01

실습 도구 소개

1.0 실습 환경 구축 방법 옵션들

1.1 AWS

1.2 Colab

1.3 머신러닝 도구들

1.4 내 PC 에 실습 환경 구축하기 (Optional)

AWS 로그인

- <https://823623552154.signin.aws.amazon.com/console>



1. 링크 접속

계정:

823623552154

← 2. 계정 : 수정하지 말것!

사용자 이름:

s201511011

← 3. 사용자이름 : s학번
Ex> s201511011

암호:

.....

← 4. 암호 : ai2020

로그인

루트 계정 자격 증명으로 로그인

암호 찾기

5. 로그인 click

AWS 로그인

AWS 계정 823623552154

IAM 사용자 이름 s201511011

이전 비밀번호 ...

새 비밀번호

새 비밀번호 재입력

비밀번호 변경 확인

[루트 계정 자격 증명으로 로그인](#)

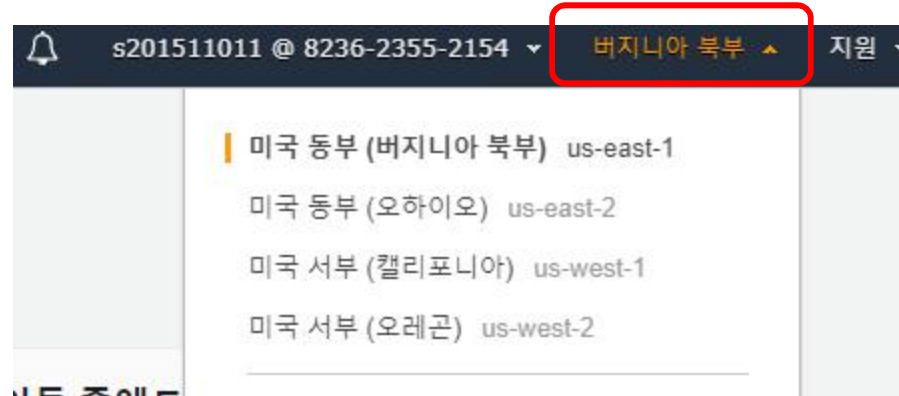
- 새 비밀번호
- 기존 비밀번호와 달라야 함
- 다른 학생들과 공유 금지

한국어 ▼

[이용 약관](#) [개인정보 보호정책](#) © 1996-2020, Amazon Web Services, Inc. 또는 자회사.

SageMaker

1. 버지니아 북부 선택



SageMaker

The screenshot shows the AWS Management Console interface. At the top, the AWS logo is on the left, and navigation links for '서비스' (Services), '리소스 그룹' (Resource Groups), and a search icon are in the center. On the right, there is a notification bell and the account ID 's201511011'. Below the header, the left sidebar contains links for '내역' (History), 'Amazon SageMaker', '콘솔 홈' (Console Home), and 'Support'. The main content area has a search bar with 'sagemaker' entered. Below the search bar, two columns of services are displayed. The first column, under the '컴퓨팅' (Computing) category, lists EC2, Lightsail, ECR, ECS, EKS, Lambda, Batch, and Elastic Beanstalk. The second column, under the 'Machine Learning' category, lists Amazon SageMaker, Amazon CodeGuru, Amazon Comprehend, Amazon Forecast, Amazon Fraud Detector, Amazon Kendra, Amazon Lex, and Amazon Machine Learning.

1. 서비스 click

2. Amazon Sagemaker 검색 & 선택

노트북 인스턴스

The screenshot shows the Amazon SageMaker console interface. The left sidebar contains a navigation menu with the following items: Amazon SageMaker Studio, 대시보드 (Dashboard), 검색 (Search), Ground Truth (with sub-items: 레이블링 작업, 데이터 세트에 레이블 지정, 레이블링 인력), **노트북** (Notebook), and 훈련 (Training). The '노트북' item is expanded, and '노트북 인스턴스' (Notebook Instances) is highlighted with a red box. A red arrow points from the text '1. 노트북 인스턴스 click' to this highlighted item. The main content area shows the 'Amazon SageMaker > 대시보드' (Amazon SageMaker > Dashboard) view. It features a blue banner for 'Amazon Elastic Inference' and a '개요' (Overview) section with two cards: 'Ground Truth' and '노트북' (Notebook). The '노트북' card describes creating training jobs and deploying models using AWS SageMaker SDK and sample notebooks.

노트북 인스턴스-생성

The screenshot shows the Amazon SageMaker console interface. On the left is a navigation sidebar with options like 'Amazon SageMaker Studio', '대시보드', '검색', 'Ground Truth', '레이블링 작업', '데이터 세트에 레이블 지정', '레이블링 인력', '노트북', '노트북 인스턴스' (highlighted), '수명 주기 구성', 'Git 리포지토리', and '훈련'. The main content area is titled 'Amazon SageMaker > 노트북 인스턴스'. At the top of this area is an 'Amazon Elastic Inference' informational banner. Below it, the '노트북 인스턴스' section contains a search bar, a refresh button, a '작업' dropdown, and a prominent orange '노트북 인스턴스 생성' button which is circled in red. A red text label '1. 노트북 인스턴스 생성 click' points to this button. Below the button is a table header with columns: 이름, 인스턴스, 생성 시간, 상태, and 작업. The table body currently displays the message '현재 리소스가 없습니다.' (No resources are currently available).

노트북 인스턴스-생성

Amazon SageMaker > 노트북 인스턴스 > 노트북 인스턴스 생성

노트북 인스턴스 생성

Amazon SageMaker는 Jupyter 노트북을 실행하는, 사전 구축된 완전 관리형 노트북 인스턴스를 제공합니다. 노트북 인스턴스에는 공통 모델 훈련 및 호스팅 연습을 위한 예제 코드가 포함되어 있습니다. [자세히 알아보기](#)

노트북 인스턴스 설정

노트북 인스턴스 이름

s201511011-notebook

1. 노트북 인스턴스 이름 : "s학번-notebook" 입력
s : 소문자 주의

최대 63자의 영숫자입니다. 하이픈(-)을 포함할 수 있지만 공백은 포함할 수 없습니다. AWS 리전의 계정 내에서 고유해야 합니다.

노트북 인스턴스 유형

ml.t2.medium

2. 노트북 인스턴스 유형 : ml.t2.medium 선택

탄력적인 추론 [자세히 알아보기](#)

없음

▶ 추가 구성

노트북 인스턴스-생성

권한 및 암호화

IAM 역할
노트북 인스턴스에는 SageMaker 및 S3 같은 다른 서비스를 호출할 수 있는 권한이 필요합니다. [AmazonSageMakerFullAccess](#) IAM 정책 첨부

AmazonSageMaker-ExecutionRole-20200221T164224 ▼

새 역할 생성

사용자 지정 IAM 역할 ARN 입력

기존 역할 사용

AmazonSageMaker-ExecutionRole-20200221T164224

사용자 지정 암호화 없음 ▼

▶ 네트워크 - 선택 사항

▶ Git 리포지토리 - 선택 사항

▶ 태그 - 선택 사항

3. IAM 역할의 “새 역할 생성” click

노트북 인스턴스-생성

IAM 역할 생성

IAM 역할을 전달하면 Amazon SageMaker가 사용자 대신 다른 AWS 서비스에서 작업을 수행할 수 있는 권한을 부여 받습니다. 여기서 역할을 생성하면 사용자가 생성하는 역할에 [AmazonSageMakerFullAccess](#) IAM 정책에 설명된 권한이 부여됩니다.
생성하는 IAM 역할은 다음에 대한 액세스를 제공합니다.

지정하는 S3 버킷 - 선택 사항

☐ 특정 S3 버킷

예 bucket-name-1, bucket-name-2, bu

임의로 구분. ARN, "*" 및 "/"는 지원되지 않습니다.

☒ 모든 S3 버킷

모든 인스턴스에 액세스할 수 있는 사용자에게 계정의 모든 버킷 및 해당 콘텐츠에 액세스할 수 있도록 허용합니다.

☐ 없음

☒ 이름에 "sagemaker"가 있는 모든 S3 버킷

☒ 이름에 "sagemaker"가 있는 모든 S3 객체

☒ 태그 "sagemaker" 및 값 "true"가 있는 모든 S3 객체

☒ SageMaker 액세스를 허용하는 버킷 정책이 있는 S3 버킷

객체 태그 지정 참조

S3 버킷 정책 참조

취소

역할 생성

4. 모든 S3 버킷 선택

5. 역할 생성 click

01. 실습 도구 소개

SW융합학부 양희경

15

노트북 인스턴스-생성

▶ 네트워크 - 선택 사항

▶ Git 리포지토리 - 선택 사항

▼ 태그 - 선택 사항

키	값	
<input type="text" value="SID"/>	<input type="text" value="s201511011"/>	<input type="button" value="제거"/>

태그 추가

취소

6. 태그에 입력
-키 : SID
-값 : s학번
* s 소문자 주의

7. 노트북 인스턴스 생성 click

노트북 인스턴스-Jupyter

☑ 성공적으로 노트북 인스턴스를 생성했습니다.
상태가 InService이면 노트북 인스턴스를 열고 템플릿 노트북을 열어 시작합니다.

세부 정보 보기

Amazon SageMaker > 노트북 인스턴스

노트북 인스턴스

🔍 노트북 인스턴스 검색

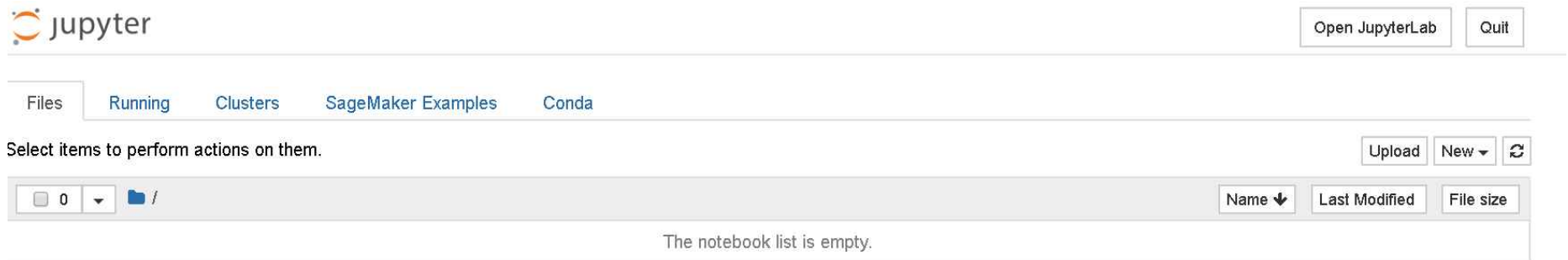
1. 이름, 인스턴스 확인

	이름	인스턴스	생성 시간	상태	작업
	s201511011-notebook	ml.t2.medium	Mar 03, 2020 01:01 UTC	🟢 InService	Jupyter 열기 JupyterLab 열기

2. 처음 노트북 인스턴스 생성 후 상태
- pending -> Inservice 까지 기다림 (약 5분)

3. Jupyter 열기 click

노트북 인스턴스-Jupyter



1. Jupyter 잘 들어가지나 확인

노트북 인스턴스-중지/시작

Amazon SageMaker > 노트북 인스턴스

노트북 인스턴스

Q 노트북 인스턴스 검색

이름	인스턴스	생성 시간	상태
s201511011-notebook	ml.t2.medium	Mar 03, 2020 01:01 UTC	InService

작업 ▼

- Jupyter 열기
- JupyterLab 열기
- 중지**
- 시작
- 설정 업데이트
- 태그 추가/편집
- 삭제

노트북 인스턴스 생성

< 1 > ⚙

열기 | JupyterLab 열기

***사용 후 노트북 인스턴스 중지하지 않는 경우 추가 과금 됨!**

***중지 전 Jupyter 반드시 저장하기**

- **상태 : Inservice -> stopping -> stopped로 바뀌면 중지된 것**
- **중지 후 다시 시작 : stopped 상태에서 작업의 "시작"click**
 - **Stopped -> Inservice 되면 노트북 인스턴스 시작된 것**

자연어처리

01

실습 도구 소개

1.0 실습 환경 구축 방법 옵션들

1.1 AWS

1.2 Colab

1.3 머신러닝 도구들

1.4 내 PC 에 실습 환경 구축하기 (Optional)

Colaboratory (Colab) 란?

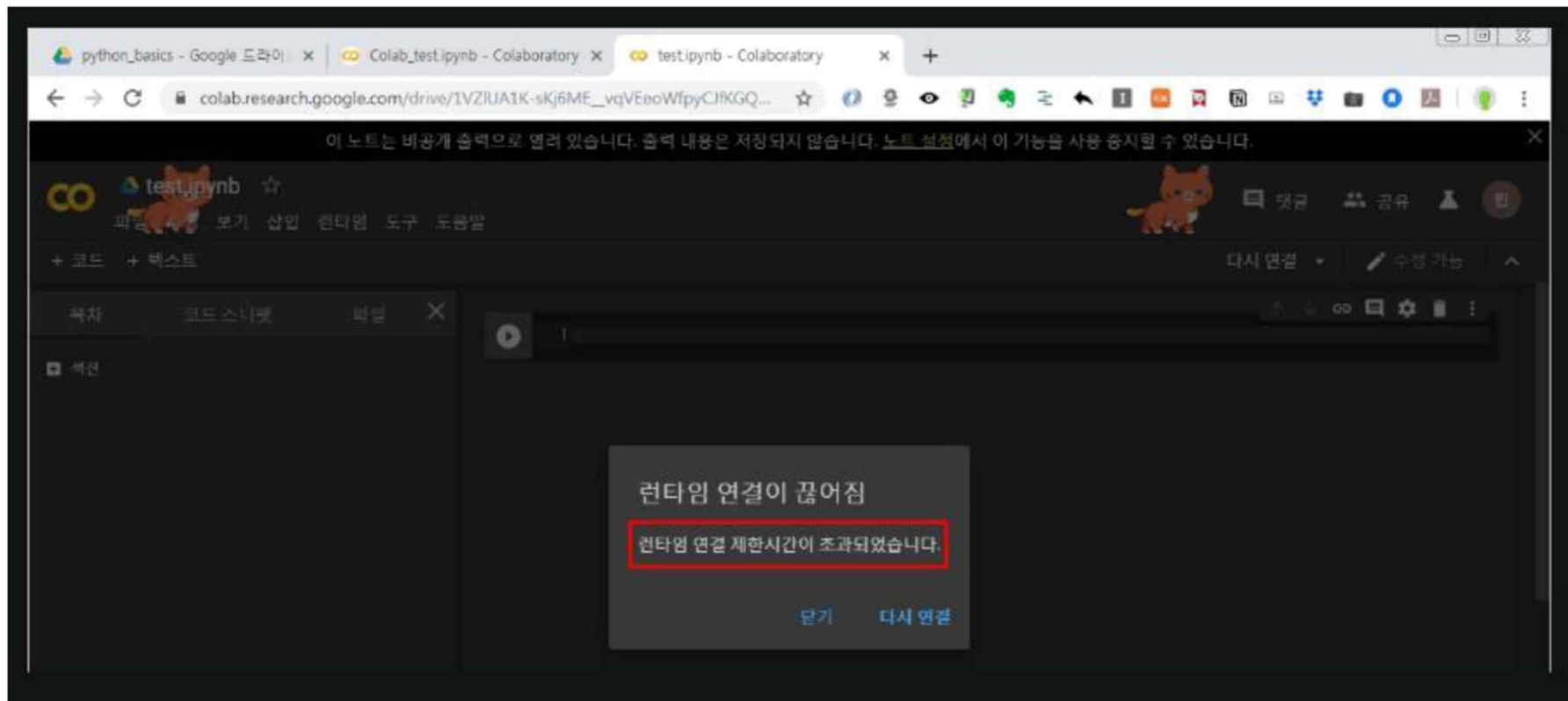
- Jupyter Notebook 이란?
 - 웹 브라우저에서 파이썬 코드를 작성하고 실행할 수 있는 개발 도구
- Google Colaboratory = Google Drive + Jupyter Notebook
- (Python2), Python3 을 지원하는 Jupyter Notebook 의 구글 커스텀 버전

Colab 의 장점

- 구글 드라이브와 연동이 가능하다.
- Git Hub 와 연동이 가능하다.
- 협업 가능
- **GPU 가속을 무료로 받을 수 있다.** (무료버전 12시간 제약)
- 본 과목 실습에서 사용되는 Pytorch 가 내장되어 있다.
- (-) 코드 실행 시 딜레이 존재

Colab 유의할 점

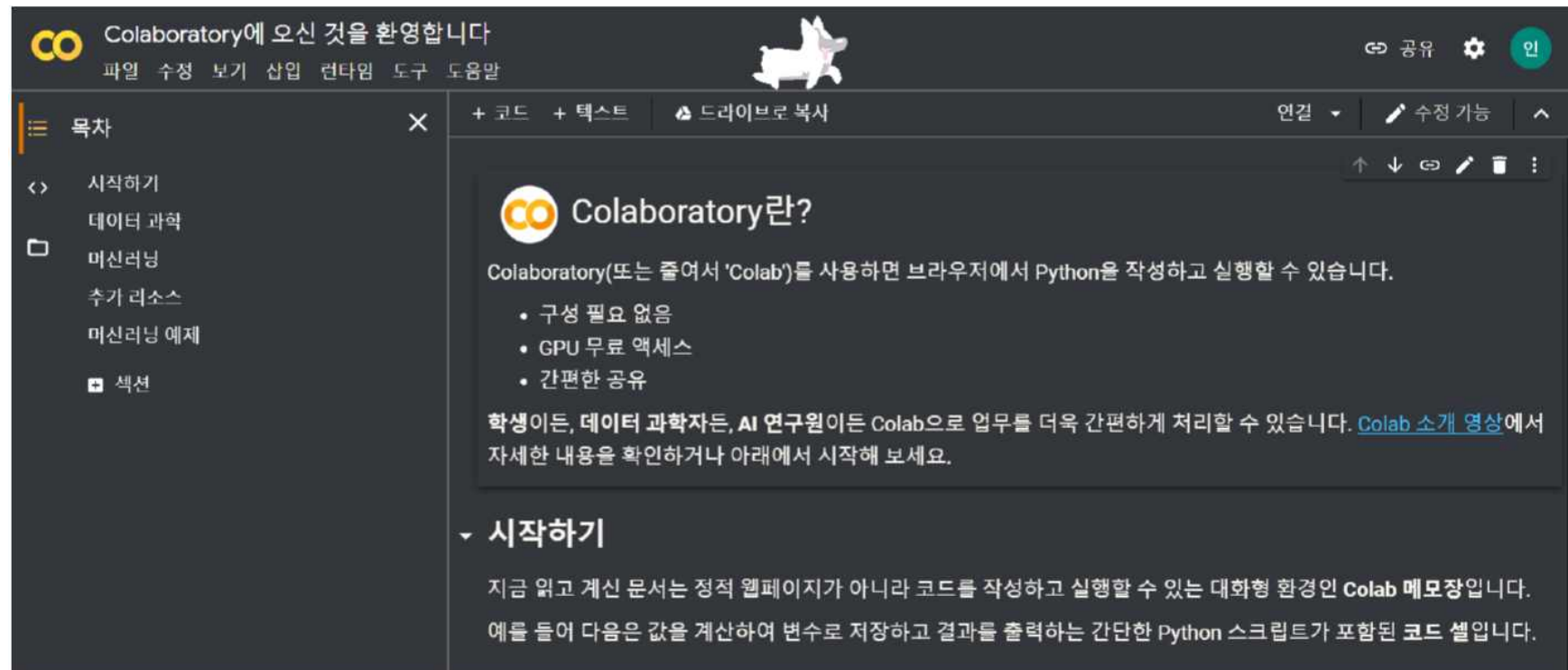
- 최대 세션 유지 시간은 **12시간**
(데이터는 소멸되나 소스코드는 자동 저장됨)



<https://theorydb.github.io/dev/2019/08/23/dev-ml-colab/>

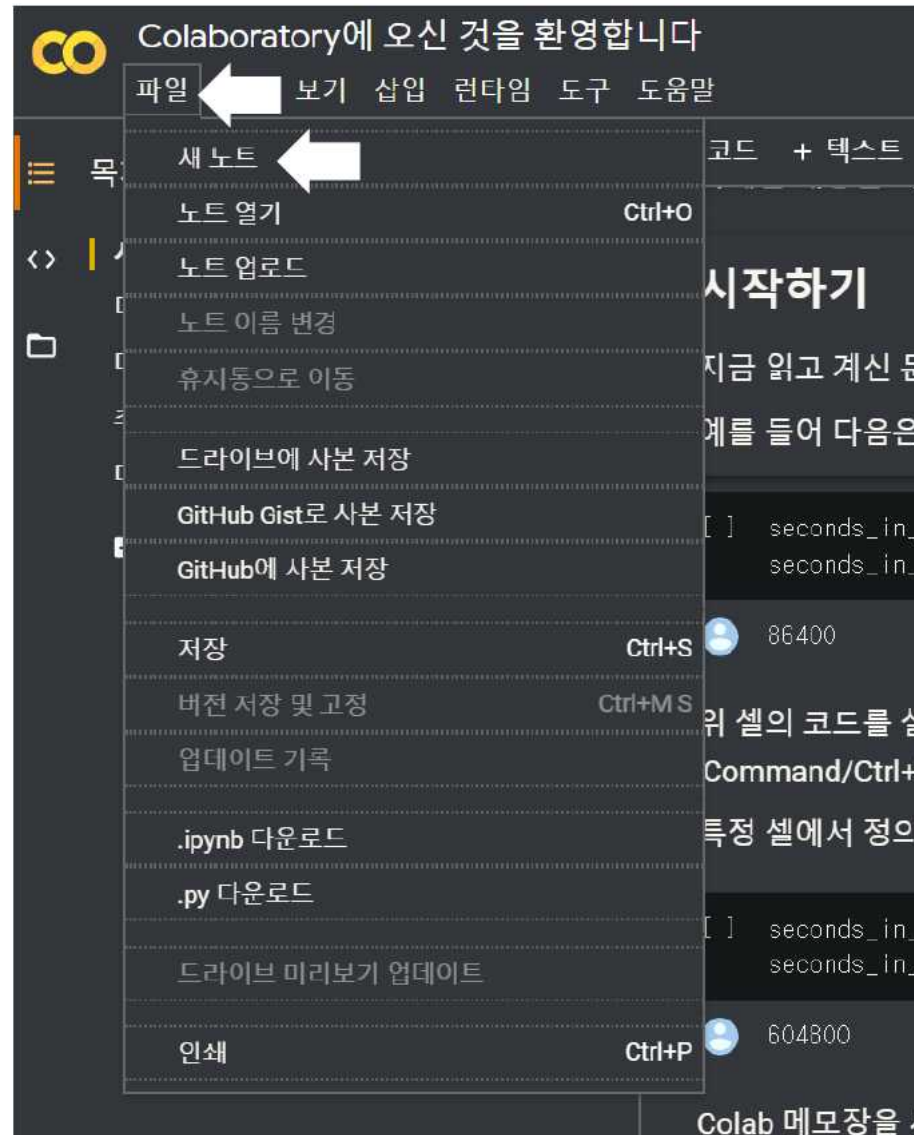
Colab 사용 방법

1. Google 로그인 후 <https://colab.research.google.com/> 접속



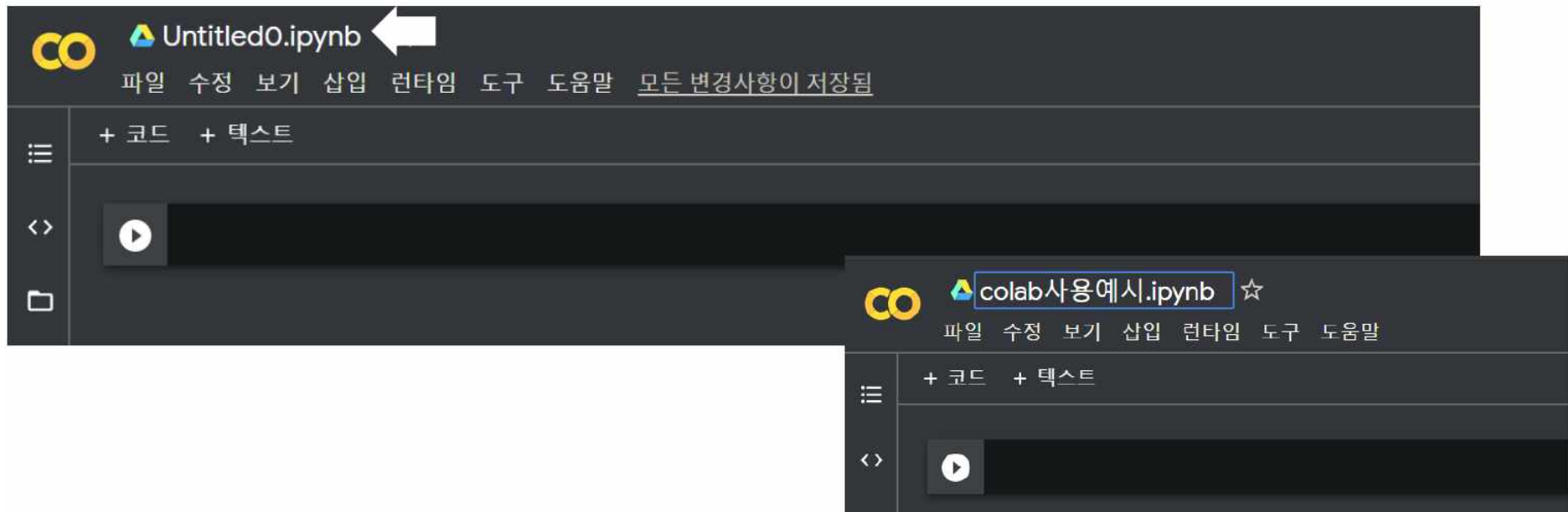
Colab 사용 방법

2. 파일 → 새 노트



Colab 사용 방법

3. 파일 이름을 클릭하여 이름 수정 가능 (Ctrl+S 로 저장)



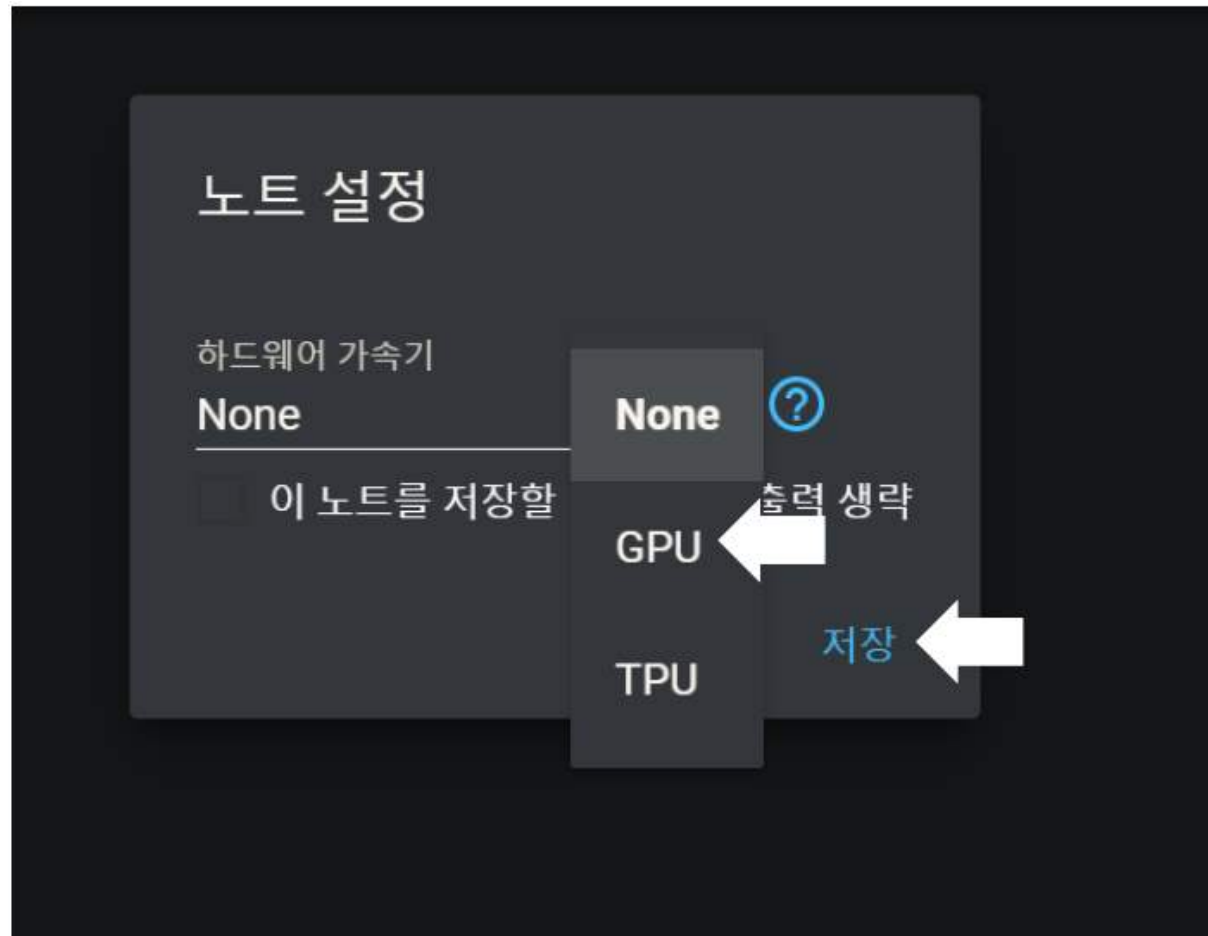
Colab 사용 방법

4. (Option) GPU 가속시, 수정 → 노트 설정(또는 런타임 → 런타임 유형 변경)



Colab 사용 방법

4. (Option) GPU 선택→ 저장



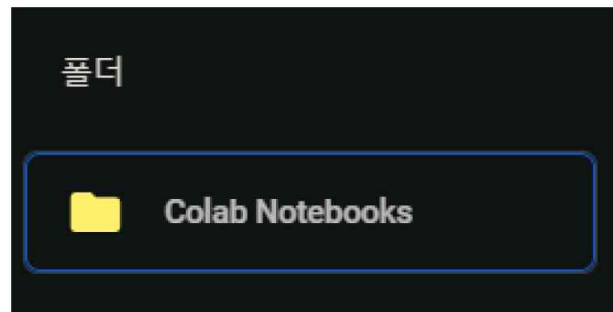
Colab 사용 방법

4. (Option) 교수님께서 주석 처리해 주신 `.cuda()` 를 주석 해제하여 Run

```
for j, [imgs, labels] in enumerate(dloader):  
    img = Variable(imgs,volatile=True).cuda()  
    #label = Variable(labels) #y  
    label = Variable(labels).cuda()  
    # .cuda() : GPU에 로드되기 위함. 만약 CPU로 설정되어 있다면 에러남
```

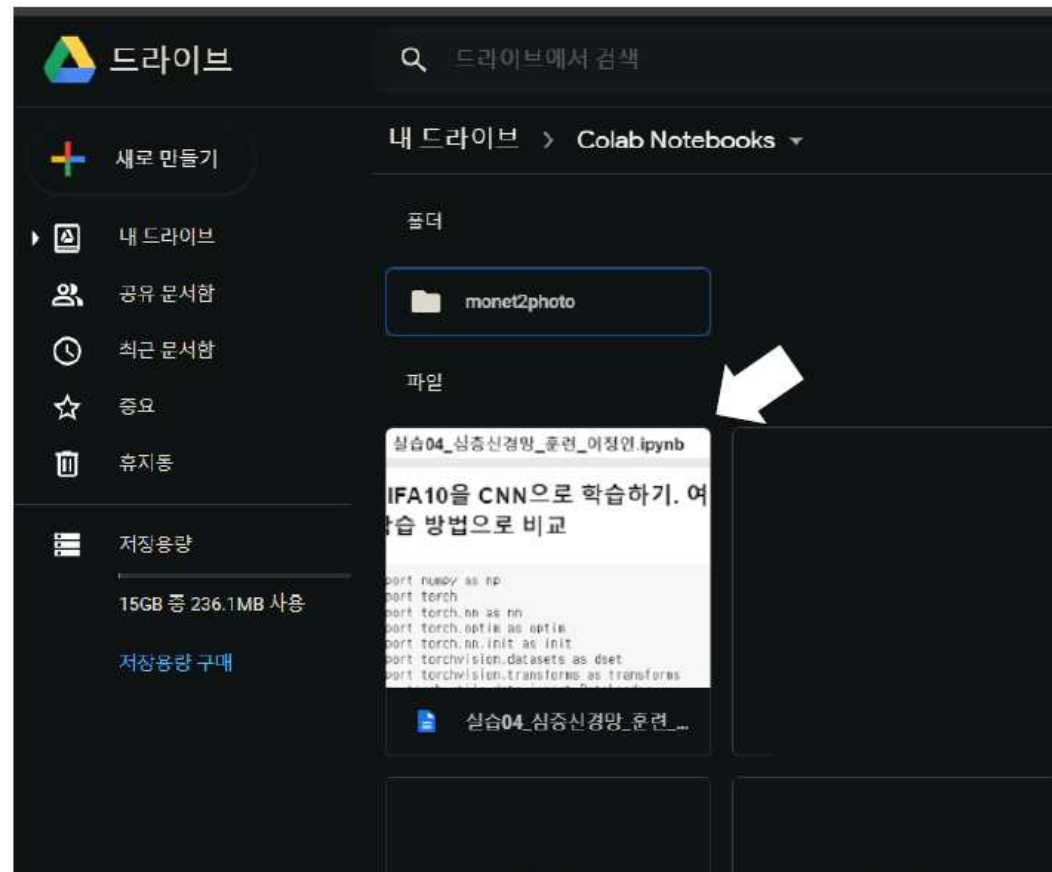
Colab 사용 방법

5. 구글 드라이브→ 내 드라이브→ Colab 폴더가 생성되었는 지 확인



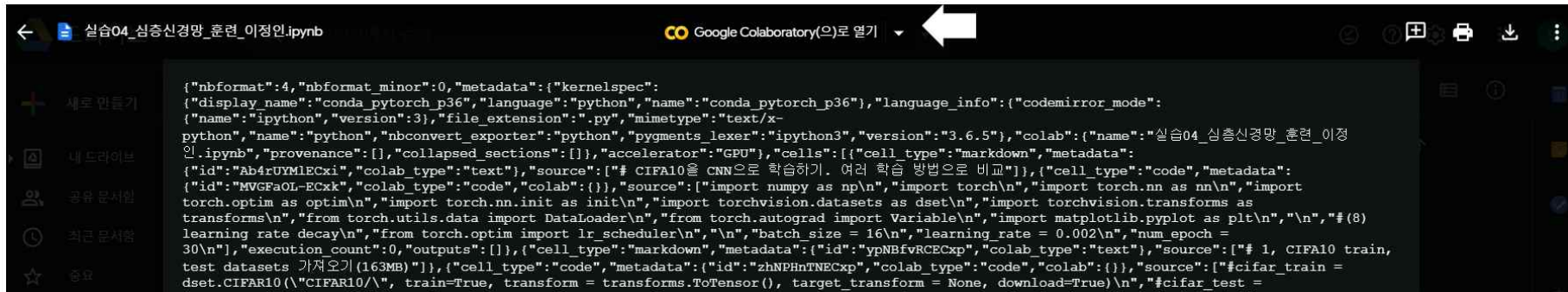
Jupyter 작업을 Colab 에서 사용할 경우

1. Jupyter에서 ipynb로 다운로드한 파일을 colab 폴더에 업로드



Jupyter 작업을 Colab 에서 사용할 경우

2. Google Colaboratory 로 열기 클릭!



Colab 관련 유용한 사이트

1. Colab 명령어

2. Google 드라이브 연동 방법

→ <https://theorydb.github.io/dev/2019/08/23/dev-ml-colab/>

3. GitHub 연동 방법

→ <https://league-cat.tistory.com/305>

자연어처리

01

실습 도구 소개

1.0 실습 환경 구축 방법 옵션들

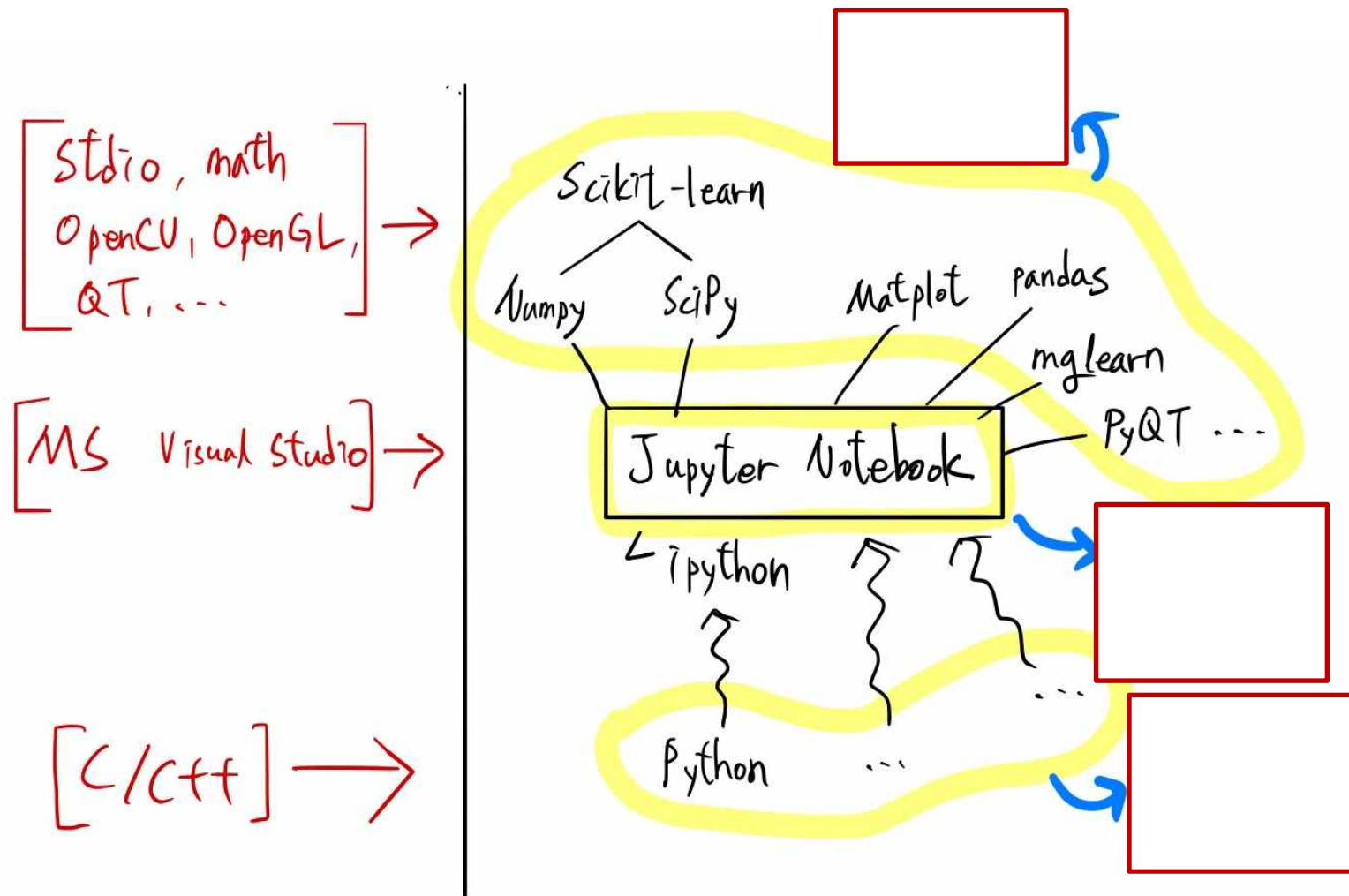
1.1 AWS

1.2 Colab

1.3 머신러닝 도구들

1.4 내 PC 에 실습 환경 구축하기 (Optional)

0) 머신러닝 프레임워크

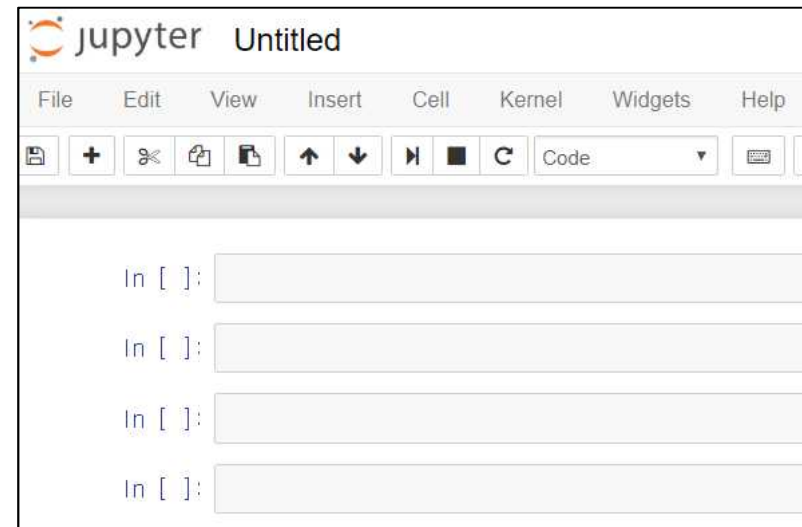


1) 왜 파이썬인가?

- 데이터 과학 분야를 위한 표준 프로그래밍 언어가 되어가고 있음
 - 기계학습 연구 분야에서 대체하기 어려운 프로그래밍 언어
- 데이터 적재, 시각화, 통계, 자연어 처리, 이미지 처리 등에 필요한 라이브러리들을 가짐
- 터미널이나 주피터 노트북 같은 도구로 대화하듯 프로그래밍 할 수 있음
- 복잡한 그래픽 사용자 인터페이스나 웹 서비스를 만들 수 있음
- 기존 시스템과 통합하기에 좋음

2) 필수 라이브러리와 도구들

- 주피터 노트북(Jupyter Notebook)
- 프로그램 코드를 브라우저에서 실행해주는 대화식 환경
- 데이터 분석에 매우 적합
- 코드와 설명, 이미지들을 쉽게 섞어 쓸 수 있음
- 역사
- IPython 노트북에서 여러 언어를 포괄하는 프로젝트인 주피터 노트북으로 이름이 바뀌었음
- Ipython 은 주피터 노트북의 파이썬 커널을 의미하게 됨
- 이름의 유래
- Julia, Python, R 의 합성어 Jupyter
- 목성의 발음과 같아 과학자들과 천문학자들에 대한 경의의 뜻
- 로고의 가운데 큰 원은 목성을, 주위 3개 작은 원은 목성의 위성3개를 최초로 발견한 갈릴레오 갈릴레이를 기리는 의미



2) 필수 라이브러리와 도구들- NumPy

- 다차원 배열을 위한 기능과 선형대수 연산과 푸리에 변환 같은 고수준 수학 함수와 유사 난수 생성기를 포함
- 핵심 기능: 다차원(n -차원) 배열인 ndarray 클래스
 - ndarray 클래스의 객체를 간단히 NumPy 배열 혹은 배열이라 부름
- <http://www.numpy.org/>

```
1 import numpy as np
2
3 x = np.array([[1,2,3],[4,5,6]])
4 print "x:\n", x
```

```
x:
[[1 2 3]
 [4 5 6]]
```

2) 필수 라이브러리와 도구들-SciPy

- 과학 계산용 함수를 모아놓은 파이썬 패키지
 - 고성능 선형 대수, 함수 최적화, 신호 처리, 특수한 수학 함수 및 통계 분포 등 기능 제공
- `scipy.sparse`
 - 희소 행렬 기능 제공
 - 희소 행렬(sparse matrix): 0을 많이 포함한 2차원 배열을 저장할 때 사용됨
- <http://www.scipy.org/scipylib>

2) 필수 라이브러리와 도구들-SciPy

```

1 from scipy import sparse
2
3 # 대각선 원소는 1이고 나머지는 0인 2차원 NumPy 배열을 만들.
4 eye = np.eye(4)
5 print "NumPy 배열:\n", eye

```

NumPy 배열:

```

[[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]]

```

```

1 # NumPy 배열을 CSR 포맷의 SciPy 희소 행렬로 변환
2 # 0이 아닌 원소만 저장됨
3 # (참고) CSR: Compressed sparse row. 행의 인덱스를 압축하여 저장
4 sparse_matrix = sparse.csr_matrix(eye)
5 print "SciPy의 CSR 행렬:\n", sparse_matrix

```

SciPy의 CSR 행렬:

```

(0, 0)      1.0
(1, 1)      1.0
(2, 2)      1.0
(3, 3)      1.0

```

```

1 data = np.ones(4)
2 row_indices = np.arange(4)
3 col_indices = np.arange(4)
4 eye_coo = sparse.coo_matrix((data, (row_indices, col_indices)))
5 print "COO 표현:\n", eye_coo

```

COO 표현:

```

(0, 0)      1.0
(1, 1)      1.0
(2, 2)      1.0
(3, 3)      1.0

```


2) 필수 라이브러리와 도구들-scikit-learn

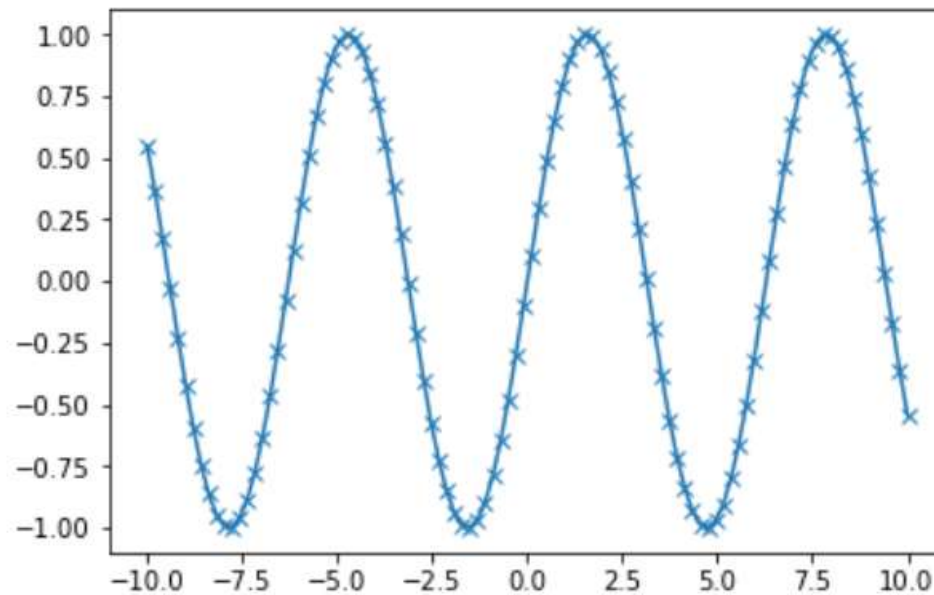
- 파이썬 머신러닝 라이브러리
- NumPy 와 SciPy 를 사용함
- <http://scikit-learn.org/stable/documentation>

2) 필수 라이브러리와 도구들-matplotlib

- 과학 계산용 그래프 라이브러리
- 선그래프, 히스토그램 등을 지원하며 출판에 쓸 수 있을 만큼의 고품질 그래프를 그려줌
- 데이터와 분석 결과를 다양한 관점에서 시각화해보면 통찰력을 얻을 수 있음
- <http://matplotlib.org/>

2) 필수 라이브러리와 도구들-matplotlib

```
1 import matplotlib.pyplot as plt
2
3 # -10에서 10까지 100개의 간격으로 나뉘어진 배열을 생성
4 x = np.linspace(-10, 10, 100)
5 # sin 함수를 사용하여 y배열을 생성
6 y = np.sin(x)
7 # plot 함수는 한 배열의 값을 다른 배열에 대응해서 선 그래프를 그림
8 plt.plot(x, y, marker='x')
9 plt.show()
```



2) 필수 라이브러리와 도구들-pandas

- 데이터 처리와 분석을 위한 파이썬 라이브러리
- DataFrame 이라는 데이터 구조를 기반으로 만들어짐
 - R 의 data.frame 을 본떠서 설계
 - 엑셀의 스프레드시트와 비슷한 테이블 형태
 - pandas 는 이 테이블을 수정하고 조작하는 다양한 기능을 제공
- SQL, 엑셀 파일, CSV 파일 등 다양한 파일과 데이터베이스에서 데이터를 읽어 들일 수 있음
- <http://pandas.pydata.org/>

2) 필수 라이브러리와 도구들-pandas

```

1  import pandas as pd
2
3  # 회원 정보가 들어간 간단한 데이터셋을 생성
4  data = {'Name': ["John", "Anna", "Peter", "Linda"],
5          'Location': ["New York", "Paris", "Berlin", "London"],
6          'Age': [24, 13, 53, 33]
7          }
8
9  data_pandas = pd.DataFrame(data)
10
11 # IPython.display 는 주피터 노트북에서 DataFrame 을 미려하게 출력해줌
12 from IPython.display import display
13 display(data_pandas)
14 #print (data_pandas)

```

	Age	Location	Name
0	24	New York	John
1	13	Paris	Anna
2	53	Berlin	Peter
3	33	London	Linda

2) 필수 라이브러리와 도구들-pandas

- 테이블에 질의하기

```
1 # Age 열의 값이 30 이상인 모든 행을 선택함  
2 display(data_pandas[data_pandas.Age > 30])
```

	Age	Location	Name
2	53	Berlin	Peter
3	33	London	Linda

2) 필수 라이브러리와 도구들

- 실습시 임포트하고 시작하자.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 from IPython.display import display
```

3) 소프트웨어 버전 확인

```
1 import sys
2 print "Python 버전:", sys.version
3
4 import pandas as pd
5 print "pandas 버전:", pd.__version__
6
7 import matplotlib
8 print "matplotlib 버전:", matplotlib.__version__
9
10 import numpy as np
11 print "NumPy 버전:", np.__version__
12
13 import scipy as sp
14 print "SciPy 버전:", sp.__version__
15
16 import IPython
17 print "IPython 버전:", IPython.__version__
18
19 import sklearn
20 print "scikit-learn 버전:", sklearn.__version__
```

Python 버전: 2.7.12 (default, Nov 12 2018, 14:36:49)
[GCC 5.4.0 20160609]
pandas 버전: 0.24.2
matplotlib 버전: 2.2.4
NumPy 버전: 1.16.5
SciPy 버전: 1.2.2
IPython 버전: 5.8.0
scikit-learn 버전: 0.17

자연어처리

01

실습 도구 소개

1.0 실습 환경 구축 방법 옵션들

1.1 AWS

1.2 Colab

1.3 머신러닝 도구들

1.4 내 PC 에 실습 환경 구축하기 (Optional)

- (Optional)Ubuntu in VirtualBox.pdf 참조 : Windows 위의 가상환경 설치 (GPU 가속 불가능)
- GPU 있고 내 PC 있을 때 (GPU 가속 가능)

1.4 내 PC에 실습 환경 구축하기(Optional)

1. NVIDIA GPU 를 구매한다. (200만원 이상)
2. PC 에 장착한다.
3. SW 설치
 - 운영체제(OS): Ubuntu
 - CUDA library, cudnn
 - Python(2.7 or 3.6)
 - 머신러닝, 딥러닝 프레임워크: Pytorch or TensorFlow

1.4 내 PC에 실습 환경 구축하기(Optional)

4. (참고) 가상환경 설정

– Option 1. Anaconda

- "아나콘다 설치 후 사용"
- `conda create -n 이름 python=2`
- (단, python 3 라면 'python=2' 생략)
- `conda env list`
- `conda remove --name 이름 --all`
- `conda activate 이름`
- `conda deactivate`
- <https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>
- (한글) <https://teddylee777.github.io/python/anaconda-%EA%B0%80%EC%83%81%ED%99%98%EA%B2%BD%EC%84%A4%EC%A0%95-%ED%8C%81-%EA%B0%95%EC%A2%8C>

– Option 2. Pyenv 와 virtualenv

- <https://lhy.kr/configuring-the-python-development-environment-with-pyenv-and-virtualenv>

1.4 내 PC에 실습 환경 구축하기(Optional)

4. (참고) 가상환경 설정

- AWS 서비스에서는 가상환경이 이미 셋팅되어 있음

