



모의경진대회 팀 대회 OT 및 2주차 코로나 감염여부 CT 이미지 분류 과제 특강

(주)마인즈앤컴퍼니 | 이녕민 매니저

Index

1. 팀 대회 소개
2. 과제 - 코로나 감염여부 CT 이미지 분류
 - 과제 및 평가지표 소개
 - 제한 사항
3. 서버 사용법
4. 베이스라인 소개
5. 팀 대회 참여 팁

모의 경진대회 과정

교육목표

- 실전 캐글 대회 참여를 대비하여 모의 캐글 문제를 풀 수 있다.
- 이미지/자연어/수치해석 Task 별로, 적절한 모델을 사용하여 추론 및 결과 제출 가능하다.
- 경진대회 진행을 위한 실험 설계 및 운영법을 터득한다.

교육 특징

- 실전과 유사한 모의경진대회 통한 실전 캐글에 대한 친숙도 증대
- 모의 경진대회 참여를 통하여 task별 AI 문제 해결 능력 향상
- 참가 팀별 별도 서버 제공하여, 원활한 경진대회 참여를 위한 자원 제공

진행 방식

- OT를 통한 과제 설명 및 베이스라인 교육
- 개인전 1회 실시 / 개인전 성적에 따른 팀 매칭 및 팀전 3회 실시
- 멘토링 세션 통한 질의응답 및 문제 해결 지원

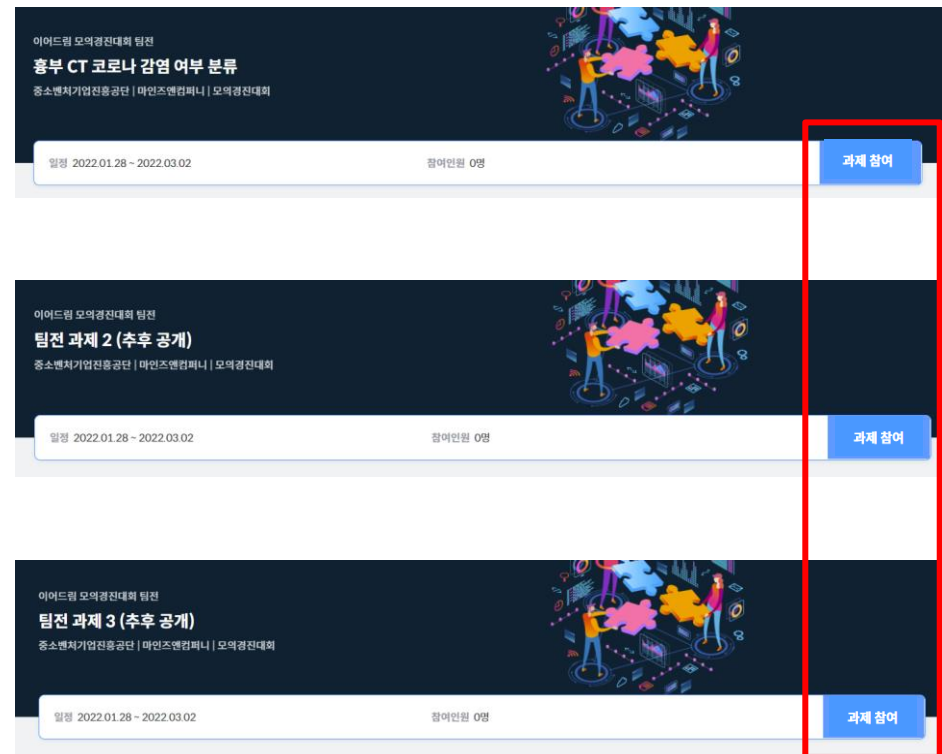
교육명		세부내용
모의 Kaggle 교육	개인전 (1회)	<ul style="list-style-type: none"> • 금융 & 정형 데이터 모의 Kaggle 경진대회 • AI CONNECT 사용법, 과제, Baseline에 대한 기본 교육 실시 • 실시간 리더보드 운영 및 개인 성적 산출
	팀전 (3회)	<ul style="list-style-type: none"> • 개인전 성적 통한 팀 매칭 및 팀별 서버 제공 • 과제, Baseline에 대한 기본 교육 실시 • 바이오 & 이미지 데이터 모의 Kaggle 경진대회 • 유통 & 정형 데이터 모의 Kaggle 경진대회 • 게임 & 자연어 데이터 모의 Kaggle 경진대회 • 실시간 리더보드 운영 및 팀별 성적 산출

AICONNECT 팀전 참여 방법

과제 3개 모두 참가하기

이후 과제는 추후 공개 예정이지만 **참가**는 꼭 오전까지 진행 필수!!

디스코드 팀 채널로 팀 구성 안내 완료
(임시)팀장이 과제 참여 진행해주세요!



제출 안내

과제개요

데이터

코드공유

과제문의

리더보드

결과제출

Step 1

파일 업로드

파일을 드래그하거나 선택하세요.

Step 2

제출설명

제출에 관련한 설명을 작성하세요.

제출하기

📧 제출 현황

결과확인

#	제출설명	Public Score	성공여부	최종선택	제출시간	CSV 파일
---	------	--------------	------	------	------	--------

< - >

Rows per page: 10

- 개인전과 동일하게 **결과제출** 탭에서 추론 결과 csv 파일 제출
- 팀으로 진행되기 때문에 실험 및 코드 버전 관리 등이 특히 중요! ('제출설명' 활용)
- 이후 최종제출 선택 및 코드 제출을 고려해 실험 기록(wandb 특강 참고) 및 팀 내 공유

Index

1. 팀 대회 소개
2. 과제 - 코로나 감염여부 CT 이미지 분류
 - 과제 및 평가지표 소개
 - 제한 사항
3. 서버 사용법
4. 베이스라인 소개
5. 팀 대회 참여 팁

흉부 CT 코로나 감염 여부 예측

흉부 CT 이미지로 코로나 감염 여부 예측 | 이미지

환자들의 흉부 CT 이미지로 코로나 감염 여부를 예측하는 문제

데이터셋

• 데이터 구조

- `train/`: CT 이미지 파일 646장
- train.csv (646 rows X 2 columns) : 이미지 파일명(file_name) 및 타겟값(COVID)
- `test/` : CT 이미지 파일 100장
- sample_submission.csv (100 rows X 2 columns) :
이미지 파일명(file_name) 및 타겟값(COVID): default 0



←0.png / train.csv→

	file_name	COVID
1	0.png	0
2	1.png	1
3	2.png	0
4	3.png	1

평가지표: Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1-Score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

		실제 정답	
		Positive	Negative
실험 결과	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Index

1. 팀 대회 소개
2. 과제 - 코로나 감염여부 CT 이미지 분류
 - 과제 및 평가지표 소개
 - 제한 사항
3. 서버 사용법
4. 베이스라인 소개
5. 팀 대회 참여 팁

본 과제의 제한 사항

- 외부 데이터 사용 불가
- 전이학습 불가

모델 라이브러리를 통하거나 .pt, .pth 등의 pre-trained 가중치 파일 업로드 후 사용 불허
다만, 모델 아키텍처는 가져와서 사용 가능하며 대회 종료 이후 코드 검증을 통해 확인 예정
(pre_trained=False 등으로 설정)

- 최종 점수가 가장 높은 제출 파일에 대한 코드를 코드 공유 탭에 게시
- 부정행위 적발 시 페널티 부여
- 결과 제출 제한: 1시간 1회 (1일 최대 24회)

Index

1. 팀 대회 소개
2. 과제 - 코로나 감염여부 CT 이미지 분류
 - 과제 및 평가지표 소개
 - 제한 사항
3. 서버 사용법
4. 베이스라인 소개
5. 팀 대회 참여 팁

서버 사양 소개

서버 사양

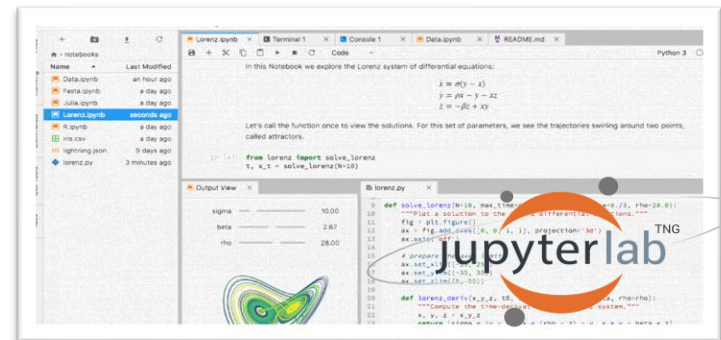
[제공 서버 * 팀당 3EA]
10 Core, 96GB mem, Nvidia T4(15GB)

[공통사양]

- Ubuntu : 18.04
- Python : 3.8.5
- Pytorch : 1.7.1
- cuda : 11.0
- nvidia driver : 455.32.00
- cudnn : 8.0.4

JupyterLab 접속법

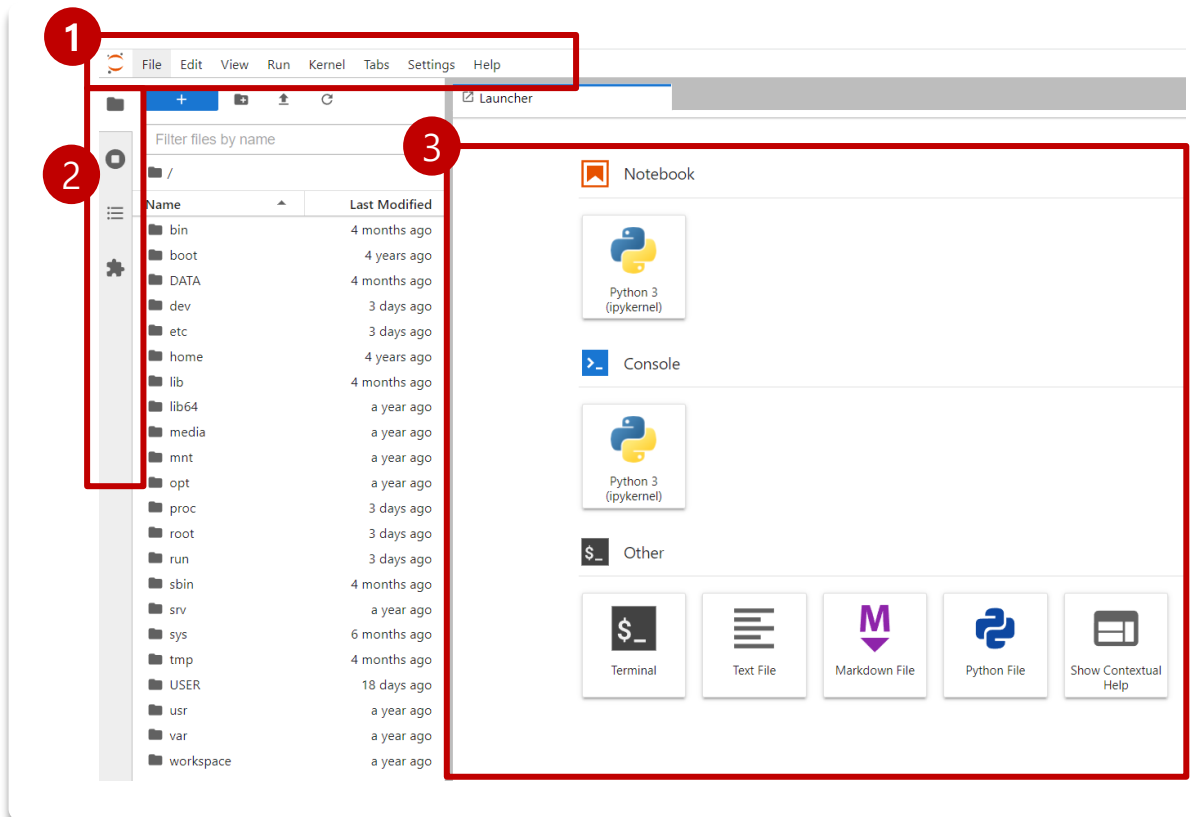
- 제공해 드린 서버 정보 사용(IP, port, P/W)
- 웹브라우저 창에서 [IP]:[port(Jupyter)] 입력
(예: 115.71.1.85:50000)
- JupyterLab 비밀번호 입력
- 접속 완료
- 서버 내 작업 공간
* /USER 폴더에 파일 용량 (약 90GB) 할당이 되어 있으니 해당 폴더에서 작업 부탁드립니다.



JupyterLab - 메인 화면

화면 예시

화면 설명



1 메뉴바

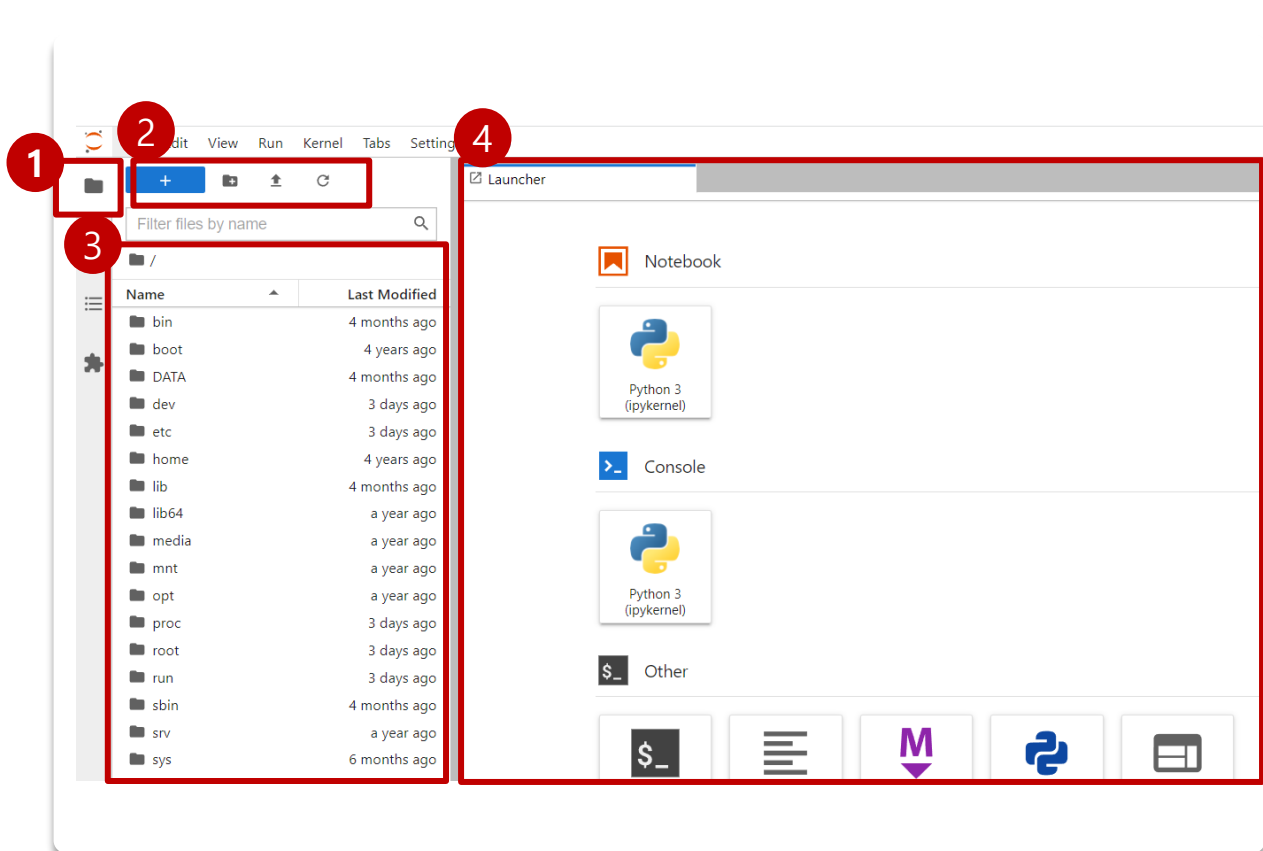
2 메뉴 아이콘

3 실행 런처
(Python, Teminal 만 주로 이용)

JupyterLab - 파일 관리 화면

화면 예시

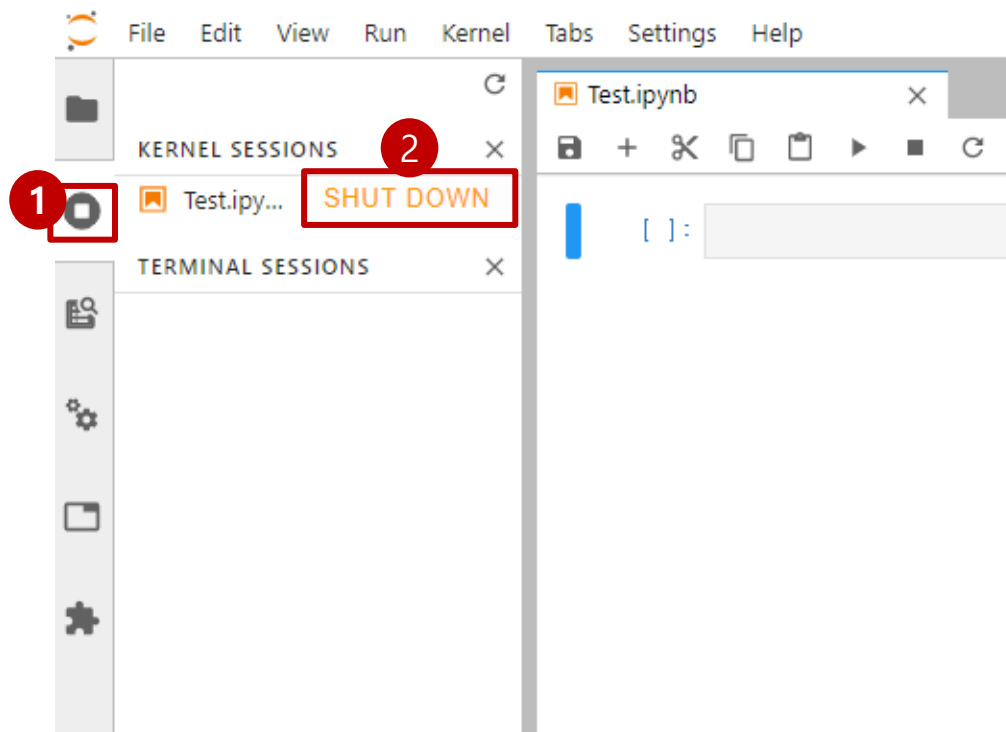
화면 설명



- 1 파일 아이콘 클릭
- 2 파일 관련 메뉴 아이콘
파이썬 파일 추가 / 폴더 추가 /
파일 업로드 / 새로그침
- 3 작업 공간(디렉토리)
- 4 작업 창

JupyterLab - 커널 관리 화면

화면 예시

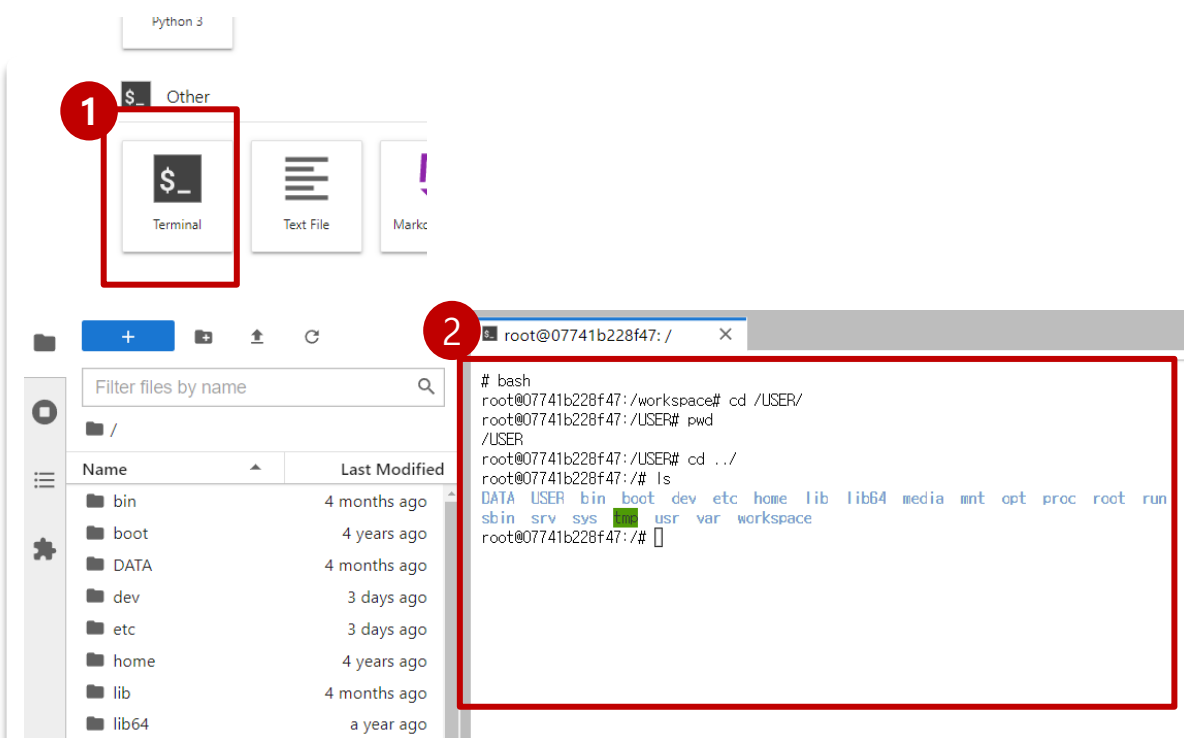


화면 설명

- 1 커널 관리 버튼
- 2 커널 종료 버튼

JupyterLab - 터미널 접속 방법

화면 예시



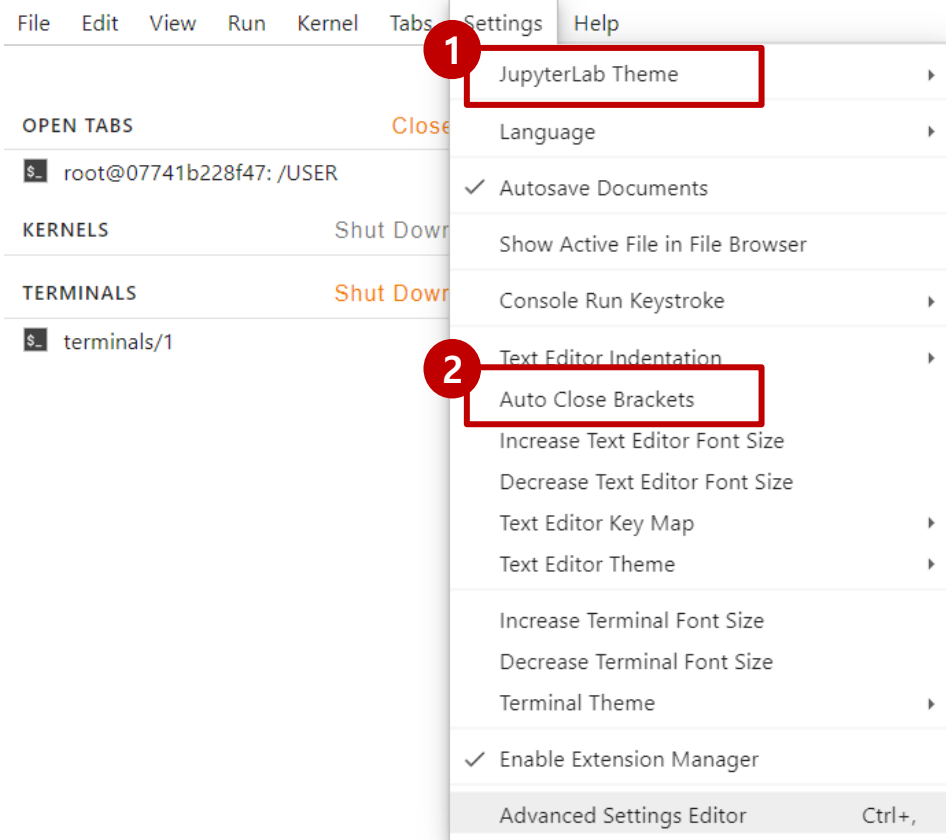
화면 설명

- 1 [실행 런처]에서 터미널 아이콘 클릭
- 2 리눅스 터미널 활용
bash 입력 후 bash shell 사용

작업은 /USER 에서 진행!!

JupyterLab - 터미널 접속 방법

화면 예시



화면 설명

- 1 다크모드 설정 가능
- 2 괄호 및 따옴표 자동 닫기

이외 개발환경 설정은 자유롭게 가능합니다. (해당 포트에만 영향)

라이브러리 설치 방법

Libraries 설치

- 리눅스: `apt-get install {library_name}`
- 파이썬: `pip install {library_name}`

acaconda 등 별도 환경 변경 권장하지 않음 (컨테이너 리셋해야 할 수 있음)

```
root@07741b228f47: /USER X
root@07741b228f47:/USER# pip install pandas
Collecting pandas
  Downloading pandas-1.4.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (
  11.7 MB)
    | 11.7 MB 17.9 MB/s
Requirement already satisfied: python-dateutil>=2.8.1 in /opt/conda/lib/python3.8/sit
e-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.8/site-packages
 (from pandas) (2020.5)
Requirement already satisfied: numpy>=1.18.5; platform_machine != "aarch64" and platf
orm_machine != "arm64" and python_version < "3.10" in /opt/conda/lib/python3.8/site-p
ackages (from pandas) (1.19.2)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.8/site-packages (fr
om python-dateutil>=2.8.1->pandas) (1.15.0)
Installing collected packages: pandas
Successfully installed pandas-1.4.0
root@07741b228f47:/USER#
```

```
root@07741b228f47: /USER X
root@07741b228f47:/USER# apt-get install screen
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libutempter0
Suggested packages:
  byobu | screenie | iselect
The following NEW packages will be installed:
  libutempter0 screen
0 upgraded, 2 newly installed, 0 to remove and 58 not upgraded.
Need to get 585 kB of archives.
After this operation, 1052 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

데이터 전송 방법

```
~/Documents/MNC_codes/yeardream
$ ls
dummy.txt

~/Documents/MNC_codes/yeardream
$ scp -P 50101 dummy.txt root@115.71.1.85:/USER
root@115.71.1.85's password:
Permission denied, please try again.
root@115.71.1.85's password:
dummy.txt                                100%   8    0.4KB/s   00:00

~/Documents/MNC_codes/yeardream
$ |
```

과제 데이터 탭에서 다운 후 각 서버에 업로드
(JupyterLab에서 drag & drop / scp 활용)

scp로 데이터 전송

- `scp -P {ssh_port_no} {source_dir} {dest_dir}`
(source/destination이 서버일 경우 - id@ip:/path/to/dir, ssh 정보 사용)
- e.g. 로컬 데이터를 서버로 전송하는 경우
 1. 로컬에서 터미널 열고 데이터 위치로 이동
 2. `scp -P {port_no} train.zip root@115.71.1.85:/USER/data`
 3. ssh 비밀번호 입력

자주 쓰는 명령어

- option: {command} -option ; 연달아 여러 개 붙여 사용 e.g. `ls -la`
- [16 Must-Know Bash Commands for Data Scientists](#)
- `clear` : 프린트된 터미널 아웃풋 모두 지우기
- `ctrl + c` : kill /// `ctrl + z` : suspend(stop)
- `watch -n # {command}` : {command}를 #초마다 반복 실행
- `ls` : 디렉토리 내 파일 보기
 - 주요 option [\[LINK\]](#)
 - `a` - 숨김파일까지 모두 보기
 - `l` - 파일 권한 함께 보기
- `zip` : 파일 압축 (`apt-get install zip`)
- `unzip` : 압축 풀기
 - 주요 option: [\[LINK\]](#)
 - `d` : 디렉토리 위치 / `q` : 내부 파일 출력 중지 (quiet)
- `jupyter nbconvert --to script {ipynb_file_path}` : .ipynb 노트북 파일을 .py 스크립트로 변경 (스크린과 함께 사용 시 실험에 유용함)

리소스 관리

- GPU 사용량 : `nvidia-smi` (정적 출력)
- CPU 사용량 : `top`

```
Mon Feb 7 09:31:21 2022
+-----+
| NVIDIA-SMI 455.32.00      Driver Version: 455.32.00      CUDA Version: 11.1      |
+-----+-----+
| GPU  Name      Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|=====+=====+
| 0 Tesla T4      On          | 00000000:10:00:0 Off |                    0 |
| N/A   33C    P8      9W /  70W | 0MiB / 15109MiB |      0%    Default  |
+-----+-----+

Processes:
GPU  GI  CI      PID  Type  Process name      GPU Memory
ID   ID
=====+=====
No running processes found
```

```
top - 06:20:41 up 76 days, 21:19, 0 users, load average: 3.46, 1.83, 0.95
Tasks: 8 total, 1 running, 7 sleeping, 0 stopped, 0 zombie
%Cpu(s): 19.0 us, 1.3 sy, 0.0 ni, 79.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 26359964+total, 16602273+free, 21222976 used, 76353920 buff/cache
KiB Swap: 33554428 total, 30164052 free, 3390376 used, 23746603+avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
 29402 root      20   0 32.719g  3.213g  986.1m S   761.0   1.3   19:15.54 python3
 25 root      20   0 1213232 364644  3346 S    0.7   0.2   23:22.56 jupyter-lab
 1 root      20   0  18496     0     0 S    0.0   0.0    0:00.07 bash
1998 root      20   0   72292     4     0 S    0.0   0.0    0:00.00 sshd
12767 root      20   0  11296    528   204 S    0.0   0.0    0:00.00 ssh-agent
18231 root      20   0    4624    664   572 S    0.0   0.0    0:00.69 sh
18232 root      20   0  18960   2416  1756 S    0.0   0.0    0:00.52 bash
29549 root      20   0  36832   1776  1324 R    0.0   0.0    0:00.06 top
```

- 주피터랩 사용 시 GPU 사용량에 프로세스가 안 잡히는 문제가 있음
- 주피터 커널이 제대로 종료되지 않은 경우 PID를 찾아 kill 해야할 수 있음

screen (가상 터미널) 사용법

- 터미널 동작 중 끊김 방지
- 오랜 학습 시 유용함
- 간단한 screen 사용법
 - 새 스크린 시작+진입: `screen -S {screen_name}`
 - 스크린 내부에서 (작동중인 채로) 바깥으로 나오기: `ctrl+a+d`
 - 스크린 terminate: `ctrl+d`
 - 띄워놓은 스크린에 다시 접속: `screen -r/x {screen_name}`
 - 스크린을 닫고 본 터미널에서 열기: `screen -D -r {screen_name}`
 - 스크린 리스트 확인: `screen -ls`

Index

1. 팀 대회 소개
2. 과제 - 코로나 감염여부 CT 이미지 분류
 - 과제 및 평가지표 소개
 - 제한 사항
3. 서버 사용법
4. 베이스라인 소개
5. 팀 대회 참여 팁

베이스라인 구성

대회 `코드 공유` 탭에서 .ipynb 다운로드 >> 서버로 업로드 후 열기!

코드 구성	설명
# Import Libraries	필요한 라이브러리 등을 로드 (필요 시 추가)
# Set Arguments & hyperparameters	여러 값들을 수정 (튜닝 시 주로 수정할 부분)
# Dataloader : CustomDataset()	데이터를 모델 인풋 형태로 가공 (augmentation 시 수정)
# Model : CustomCNN()	모델 아키텍처 선언 (모델 변경 시 수정)
# utils : LossEarlyStopper(), Trainer(), get_metric_fn()	배치 단위 학습 시 필요한 클래스들
# Train	에폭 단위로 학습 진행
# Inference	테스트 데이터에 대한 추론 진행 및 제출파일 생성

Seed값 고정

일관적인 성능 비교 및 재현에 필수적임

사용하는 라이브러리 중 랜덤 시드가 사용되는 항목들의 시드를 특정 값으로 고정
베이스라인의 경우 torch, numpy, random
torch.backends [참고 링크](#)

```
# 시드(seed) 설정

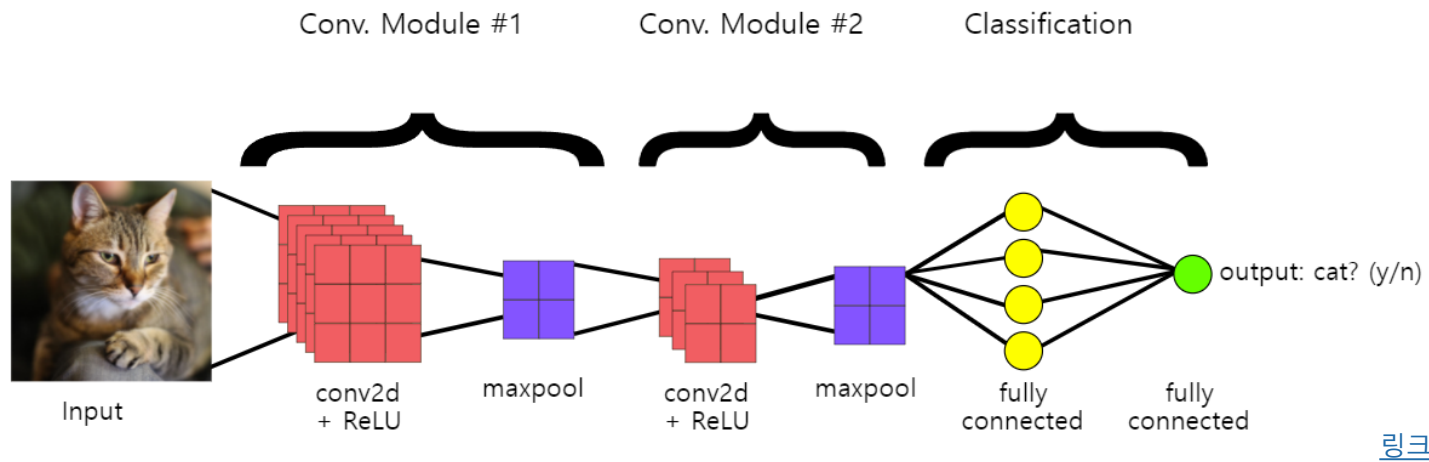
RANDOM_SEED = 2022

torch.manual_seed(RANDOM_SEED)
torch.backends.cudnn.deterministic = True
torch.backends.cudnn.benchmark = False
np.random.seed(RANDOM_SEED)
random.seed(RANDOM_SEED)
```

CustumDataset() 클래스

함수	기능
<code>def __init__()</code>	초기화 1. 변수 지정 2. train 과 validation 데이터로 분할 - 단순히 9:1로 분할 3. 이미지 변환 (torchvision.transforms) - 리사이즈, 텐서화, CNN 정규화만 진행
<code>def data_loader()</code>	csv 불러오기
<code>def __len__()</code>	데이터 개수 리턴
<code>def __getitem__()</code>	각 데이터의 이미지 텐서, 라벨값 리턴

Custum_CNN() 아키텍처



함수	기능
<code>def __init__()</code>	레이어 선언
<code>def forward()</code>	레이어 쌓는 부분 (선언된 모델이 실행되면 동작)

모델 output shape 계산

N : Batch size
 C : channel (in & out)
 H : image height
 W : image width
 $padding$: default = (0,0)
 $stride$:
 - Conv2d default = 1,
 - MaxPool2d default = kernel size
 $dilation$: default = (1,1)

[baseline]

1. conv1
2. pool
3. conv2
4. pool
5. linear1
6. linear2

<conv1>

$N = 32$

$H_{in} = W_{in} = 128$

$padding = (0,0)$

$kernel_size = (5,5)$

Shape:

Conv2d & MaxPool2d 계산방법 동일, [링크](#)

- Input: $(N, C_{in}, H_{in}, W_{in})$
- Output: $(N, C_{out}, H_{out}, W_{out})$ where

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times padding[0] - dilation[0] \times (kernel_size[0] - 1) - 1}{stride[0]} + 1 \right\rfloor$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times padding[1] - dilation[1] \times (kernel_size[1] - 1) - 1}{stride[1]} + 1 \right\rfloor$$

$$\begin{aligned}
 H_{out} &= W_{out} \\
 &= \left\lfloor \frac{128 + 2 \times 0 - 1 \times (5 - 1) - 1}{1} + 1 \right\rfloor \\
 &= 124
 \end{aligned}$$

⇒ Output: (32, 8, 124, 124)

모델 output shape 계산

N : Batch size
 C : channel (in & out)
 H : image height
 W : image width
 $padding$: default = (0,0)
 $stride$:
 - Conv2d default = 1,
 - MaxPool2d default = kernel size
 $dilation$: default = (1,1)

[baseline]

1. conv1
2. pool
3. conv2
4. pool
5. linear1
6. linear2

<pool>

$N = 32$

$H_{in} = W_{in} = 124$

$padding = (0,0)$

$kernel_size = (2,2)$

Shape:

Conv2d & MaxPool2d 계산방법 동일, [링크](#)

- Input: $(N, C_{in}, H_{in}, W_{in})$
- Output: $(N, C_{out}, H_{out}, W_{out})$ where

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times padding[0] - dilation[0] \times (kernel_size[0] - 1) - 1}{stride[0]} + 1 \right\rfloor$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times padding[1] - dilation[1] \times (kernel_size[1] - 1) - 1}{stride[1]} + 1 \right\rfloor$$

$$\begin{aligned}
 H_{out} &= W_{out} \\
 &= \left\lfloor \frac{124 + 2 \times 0 - 1 \times (2 - 1) - 1}{2} + 1 \right\rfloor \\
 &= 62
 \end{aligned}$$

⇒ Output: (32, 8, 62, 62)

모델 output shape 계산

N : Batch size
 C : channel (in & out)
 H : image height
 W : image width
 $padding$: default = (0,0)
 $stride$:
 - Conv2d default = 1,
 - MaxPool2d default = kernel size
 $dilation$: default = (1,1)

[baseline]

1. conv1
2. pool
3. conv2
4. pool
5. linear1
6. linear2

<conv2>

$N = 32$

$H_{in} = W_{in} = 62$

$padding = (0,0)$

$kernel_size = (5,5)$

Shape:

Conv2d & MaxPool2d 계산방법 동일, [링크](#)

- Input: $(N, C_{in}, H_{in}, W_{in})$
- Output: $(N, C_{out}, H_{out}, W_{out})$ where

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times padding[0] - dilation[0] \times (kernel_size[0] - 1) - 1}{stride[0]} + 1 \right\rfloor$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times padding[1] - dilation[1] \times (kernel_size[1] - 1) - 1}{stride[1]} + 1 \right\rfloor$$

$$\begin{aligned}
 H_{out} &= W_{out} \\
 &= \left\lfloor \frac{62 + 2 \times 0 - 1 \times (5 - 1) - 1}{1} + 1 \right\rfloor \\
 &= 58
 \end{aligned}$$

⇒ Output: (32, 25, 58, 58)

모델 output shape 계산

N : Batch size
 C : channel (in & out)
 H : image height
 W : image width
 $padding$: default = (0,0)
 $stride$:
 - Conv2d default = 1,
 - MaxPool2d default = kernel size
 $dilation$: default = (1,1)

[baseline]

1. conv1
2. pool
3. conv2
4. pool
5. linear1
6. linear2

<pool>

$N = 32$

$H_{in} = W_{in} = 58$

$padding = (0,0)$

$kernel_size = (2,2)$

Shape:

Conv2d & MaxPool2d 계산방법 동일, [링크](#)

- Input: $(N, C_{in}, H_{in}, W_{in})$
- Output: $(N, C_{out}, H_{out}, W_{out})$ where

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times padding[0] - dilation[0] \times (kernel_size[0] - 1) - 1}{stride[0]} + 1 \right\rfloor$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times padding[1] - dilation[1] \times (kernel_size[1] - 1) - 1}{stride[1]} + 1 \right\rfloor$$

$$\begin{aligned}
 H_{out} &= W_{out} \\
 &= \left\lfloor \frac{58 + 2 \times 0 - 1 \times (2 - 1) - 1}{2} + 1 \right\rfloor \\
 &= 29
 \end{aligned}$$

⇒ Output: (32, 25, 29, 29)

모델 output shape 계산

[baseline]

1. conv1
2. pool
3. conv2
4. pool
5. linear1
6. linear2

- a. linear1 전 shape : (32, 25, 29, 29)
- b. Flatten : a \rightarrow (32, 25*29*29)
- c. Linear1 : 25*29*29 \rightarrow 128
- d. Linear2 : 128 \rightarrow 2
- e. softmax : shape 유지 (output \rightarrow proba)

```
import torch.nn.functional as F

class custom_CNN(nn.Module):
    def __init__(self, num_classes):
        super(custom_CNN, self).__init__()
        self.conv1 = nn.Conv2d(in_channels=3, out_channels=8, kernel_size=5)
        self.pool = nn.MaxPool2d(kernel_size=2)
        self.conv2 = nn.Conv2d(in_channels=8, out_channels=25, kernel_size=5)

        self.fc1 = nn.Linear(in_features=25*29*29, out_features=128)
        self.fc2 = nn.Linear(in_features=128, out_features=num_classes)
        self.softmax = nn.Softmax(dim=1)

    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x))) # (32, 3, 128, 128) -> (32, 8, 62, 62)
        x = self.pool(F.relu(self.conv2(x))) # (32, 8, 62, 62) -> (32, 25, 29, 29)

        x = torch.flatten(x,1)
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))

        output = self.softmax(x)

        return output
```


Utils: 학습에 필요한 클래스들

클래스	설명
<code>LossEarlyStopper()</code>	일정 횟수(patience) 이상 CV loss 개선이 없을 경우 학습을 중단함
<code>Trainer()</code>	Epoch별 학습 및 검증 절차 train_epoch / validate_epoch (미니배치 단위 절차)
<code>get_metric_fn()</code>	메트릭 점수 계산

학습 및 추론

• 학습

- Trainer의 train_epoch, valid_epoch를 에폭마다 실행
- CV loss에 따른 early stop 여부 확인
- Loss 개선 시 가중치 저장

• 추론

1. 라벨을 제외하고 데이터 로드
2. 미니배치 단위로 추론
3. 추론 결과 csv로 저장

```

criterion = 1E+8
for epoch_index in tqdm(range(EPOCHS)):

    trainer.train_epoch(train_dataloader, epoch_index)
    trainer.validate_epoch(validation_dataloader, epoch_index)

    # early_stopping check
    early_stopper.check_early_stopping(loss=trainer.val_mean_loss)

    if early_stopper.stop:
        print('Early stopped')
        break

    if trainer.val_mean_loss < criterion:
        criterion = trainer.val_mean_loss
        check_point = {
            'model': model.state_dict(),
            'optimizer': optimizer.state_dict(),
            'scheduler': scheduler.state_dict()
        }
        torch.save(check_point, 'best.pt')
    
```

Index

1. 팀 대회 소개
2. 과제 - 코로나 감염여부 CT 이미지 분류
 - 과제 및 평가지표 소개
 - 제한 사항
3. 서버 사용법
4. 베이스라인 소개
5. 팀 대회 참여 팁

여러 명이 한 과제를 수행할 때

- 중요한 점
 - 효율적인 시간 및 리소스 관리
 - 실험 결과 기록 및 공유
 - Weights & Biases (<https://wandb.ai/>, 8일 특강 참고)
 - 구글 시트 (python 라이브러리 [box](#) 등 활용 가능)
 - 이 외 팀원들이 합의한 툴
- 고려할 이번 대회 환경
 - 팀당 GPU 3대 (T4, 분리되어 병렬처리 불가)
- 실험 순서
 - 시기별:
 - 초반 - EDA, 데이터 실험, backbone 모델 선정 등
 - 중반 - 모델 실험 (모델 크기 및 세부 구조, hyper-parameter 튜닝 등)
 - 후반 - 앙상블, 후처리, 시드 조정 등
 - 주로 시기별로 아이디어를 낸 후, 아이디어를 팀원들이 분담해 실험 진행



End of document