



**모의경진대회 오리엔테이션 및
1차 모의경진대회 베이스라인 특강:
흉부 CT 이미지 코로나 감염 여부 분류**

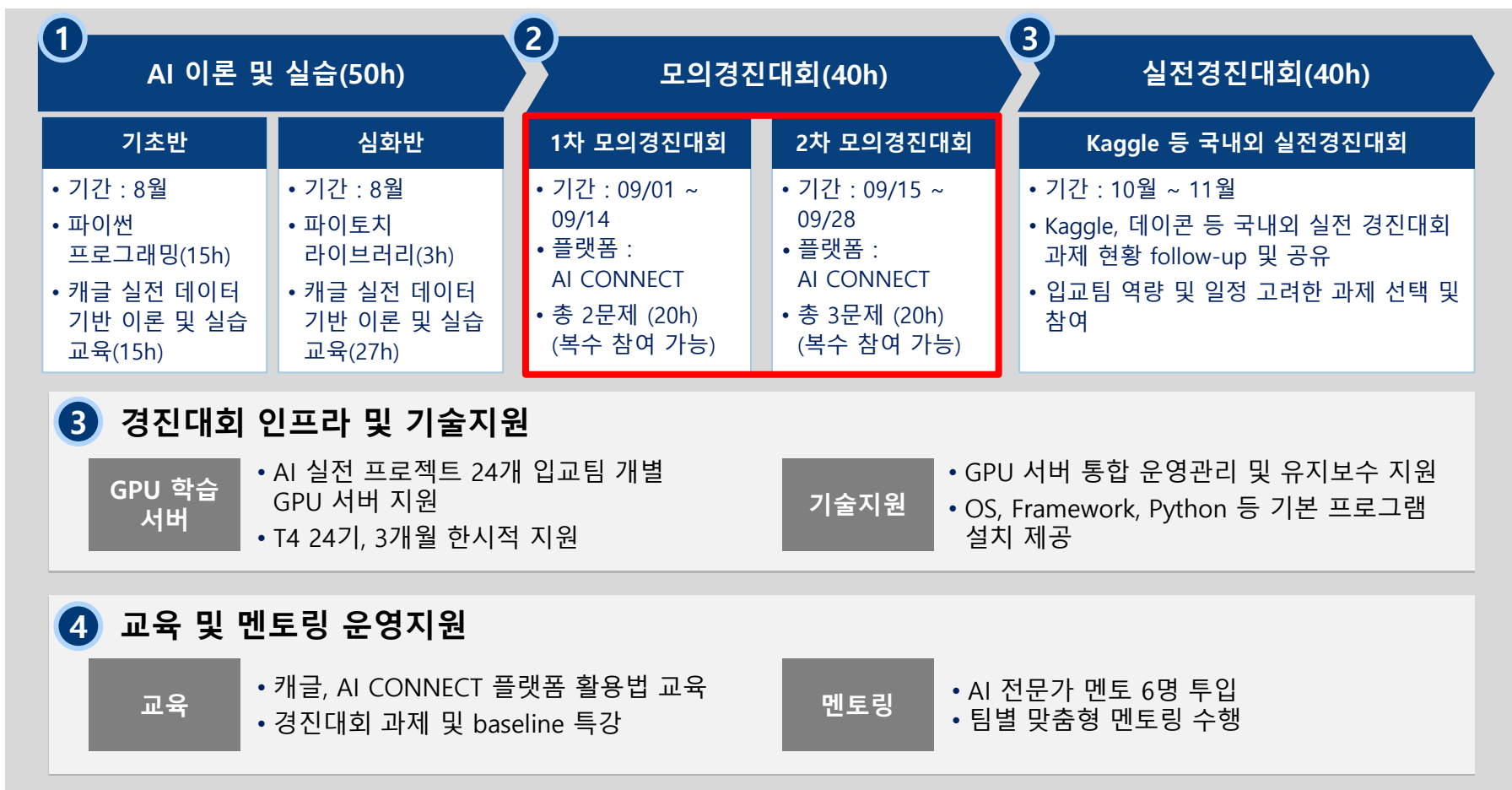
(주)마인즈앤컴퍼니 | 김태훈 매니저

2022.09.01

Index

1. 모의경진대회 및 실전경진대회 커리큘럼 및 일정
2. AI CONNECT 플랫폼 소개 및 사용법 안내
3. 1차 모의경진대회 – (이미지) 흉부 CT 이미지 코로나 감염 여부 분류 과제
 - 과제 소개 (데이터 / 평가지표 / 제한사항)
 - 서버 사용법
 - 베이스라인 코드

AI 실전 프로젝트 개요



모의경진대회 개요

교육목표

- 모의경진대회 참여를 통해 **실전경진대회를 위한 실전 감각을 획득**한다.
- 이미지/자연어/수치해석 Task 별로 **적합한 모델을 선택**하고 **학습 및 추론**을 진행할 수 있다.
- 모델의 성능을 높이기 위한 **실험 설계 방법**을 터득한다.

과정구성

- 1차 모의경진대회(기간 : 09/01 ~ 09/14)
- 2차 모의경진대회(기간 : 09/15 ~ 09/28)
- 모의경진대회 과제 및 베이스라인 특강
- 모의경진대회 우승자 코드 리뷰 세션
- 입교팀 개별 GPU 서버 제공 (~ 11월 말)
- 멘토링 세션(줌 소회의실)
- 전문가 특강

학점 이수

- 모의경진대회 과제 제출 당 2h
- 1,2차 모의경진대회 과제 제출로 획득 가능한 최대 학점은 각 20h
- 멘토링 세션 참여 당 1h
- 특강 참여 시 특강 시간에 대한 학점 부여 (베이스라인 / 전문가 / 코드)
- 9월 중 획득 가능한 최대 학점은 40h

교육명		세부 내용
모의 경진 대회 (40h)	1차 모의 경진 대회 (20h)	<ul style="list-style-type: none"> • (이미지) 흉부 CT 이미지 코로나 감염 여부 분류 • (정형/시계열) 뇌파 데이터 수면 단계 분류 • AI CONNECT 플랫폼, 과제 및 Baseline 특강 • 실시간 리더보드 운영 및 팀별 성적 산출 • 2문제 중 최소 1문제 제출 (복수 과제 참여 및 제출 가능)
	2차 모의 경진 대회 (20h)	<ul style="list-style-type: none"> • (이미지) 소도체 단면 이미지 객체 분할 • (정형/시계열) 도로별 물류 통행량 예측 • (자연어) 도서자료 데이터 기계독해(MRC) • AI CONNECT 플랫폼, 과제 및 Baseline 특강 • 실시간 리더보드 운영 및 팀별 성적 산출 • 2문제 중 최소 1문제 제출 (복수 과제 참여 및 제출 가능)

모의경진대회 일정

2022년 9월						
일요일	월요일	화요일	수요일	목요일	금요일	토요일
				1 1차 모의경진대회 OT - AI Connect 플랫폼 활용법 - 1차 모의대회 문제 설명 (Baseline 실습 진행) ※ 온/오프라인 병행	2	3
4	5	6 글장사 우수 졸업팀 특강 - (주)딥인사이트, 웨이브엑스 - 사업 소개 및 경진대회 수상 경험 공유 ※ 오프라인 단독	7 멘토링 세션 - Tabular : 곡대훈, 김동민 - Image : 영준하, 이상현 - NLP : 김태성, 최민석	8	9	10 추석
11	12 대체 휴일	13 글장사 우수 졸업팀 특강 - (주)담다디 창지현 대표 - 사업 소개 및 경진대회 수상 경험 공유 ※ 오프라인 단독	14 1차 모의경진대회 종료	15 2차 모의경진대회 OT - 2차 모의대회 문제 설명 (Baseline 실습 진행) ※ 온/오프라인 병행	16	17
18	19 1차 모의경진대회 코드 리뷰 ※ 온/오프라인 병행	20 MNC 특강 - 고석대 대표, 진방현 실장 - AI 기술 비즈니스 활용 - 스타트업 조직관리와 HR 전략 ※ 온/오프라인 병행	21 멘토링 세션 - Tabular : 곡대훈, 김동민 - Image : 영준하, 이상현 - NLP : 최민석	22	23	24
25	26	27 전문가 특강 - 서강대 구명환 교수 - AI Production 상용화 시스템 개발	28 2차 모의경진대회 종료	29	30	

1

1,2차 모의경진대회

- 정형/이미지/자연어 task 별 실전 Kaggle 출제 가능성이 높은 모의경진대회 과제 구성
- 과제 및 Baseline 특강 제공
- 참가자 별 서버 제공

2

멘토링 세션

- 매 2주 입교팀별 30분씩의 맞춤형 멘토링 세션 제공
- 총 6명의 KAIST AI 석박과정 멘토진
- 추가 멘토링 요청 시 일정 조정 가능

3

특강

- 주 1회 글장사 우수 졸업팀 / 전문가 특강 진행
- 오프라인 세션 확대를 통한 선후배 입교팀 간 네트워킹 장려

1차 모의경진대회 일정

일정



경진대회 세부 일정

- ✓ 오리엔테이션 및 과제 베이스라인 특강 : 09.01(목) 14:00 ~ 16:00
- ✓ 과제 참여 신청(복수 과제 참여 가능) : 09.01(목) 14:00 ~ 09.05(월) 24:00
- ✓ 추론 결과 제출 : 09.06(화) 00:00 ~ 09.14(수) 12:00
- ✓ 결과 발표 : 09.14(수) 18:00
- ✓ 과제별 우승팀 코드 리뷰 세션 : 09.19(월) 14:00 ~ 16:00

모의경진대회 우승팀 특전

글로벌★창업사관학교
1차 모의경진대회

뇌파 데이터 수면 단계 분류
뇌파 데이터를 이용해 수면 단계를 분류하는 과제

모의경진대회

0명 / 0팀

우승팀 : A사

글로벌★창업사관학교
1차 모의경진대회

흉부 CT 이미지 코로나 감염 여부 분류
흉부 CT 이미지로 코로나 감염 여부를 이진 분류하는 과제

모의경진대회

0명 / 1팀

우승팀 : B사

글로벌★창업사관학교
2차 모의경진대회

Make the Most of AI,
MINDs@company

과제1 (추후 공개)

재양

0명 / 0팀

우승팀 : A사

글로벌★창업사관학교
2차 모의경진대회

Make the Most of AI,
MINDs@company

과제2 (추후 공개)

재양

0명 / 5팀

우승팀 : C사

글로벌★창업사관학교
2차 모의경진대회

Make the Most of AI,
MINDs@company

과제3 (추후 공개)

재양

0명 / 5팀

우승팀 : D사

- 모의경진대회 과제별 우승 입교팀에게는 20만원 상당의 상품을 지급
- 1차 모의경진대회 과제 우승팀이 2차 모의경진대회 과제에서 우승하는 경우에는 중복해서 상품을 지급
- 하지만, 1, 2차 모의경진대회 안에서 복수 과제에서 동시에 우승하는 경우에는 제출 횟수가 가장 많은 과제에 대해서만 상품을 지급하고, 다른 과제의 경우 차순위자에게 상품을 지급
- 과제별 우승팀의 코드 리뷰 세션 예정

글창사 우수 졸업팀 / 전문가 특강 세션

특강일자	내 용	교육방식
9월 1일(화) 14:00~16:00 (2h)	1차 모의경진대회 오리엔테이션 - MNC 김태훈, 박성호 매니저 - AI CONNECT 플랫폼 활용법 / 1차 모의경진대회 과제 및 베이스라인	온/오프라인 병행
9월 6일(화) 14:00~17:00 (3h)	글창사 우수 졸업팀 특강 - 딥인사이트 오은송 대표, 지인찬 CTO - 사업 소개 / 경진대회 수상 경험 공유 - 웨이브텍 정해갑 대표 - 사업 소개 / 경진대회 수상 경험 공유	오프라인 단독
9월 13일(화) 14:00 ~ 16:00 (2h)	글창사 우수 졸업팀 특강 - 담다디 정지한 대표 - 사업 소개 / 경진대회 수상 경험 공유	오프라인 단독
9월 15일(화) 14:00 ~ 16:00 (2h)	2차 모의경진대회 오리엔테이션 - MNC 박성호, 김태훈, 도성진 매니저 - 2차 모의경진대회 과제 및 베이스라인	온/오프라인 병행
9월 19일(월) 14:00 ~ 16:00 (2h)	1차 모의경진대회 코드 리뷰 세션 - 과제별 우승팀	온/오프라인 병행
9월 20일(화) 14:00 ~ 17:00 (3h)	MNC 특강 - MNC 고석태 대표, 진방현 실장 - AI 기술 비즈니스 활용 사례 / 스타트업 조직관리와 HR 전략	온/오프라인 병행
9월 27일(화) 14:00 ~ 16:00 (2h)	전문가 특강 - 서강대 구명완 교수 - AI Production 상용화 시스템 개발	온라인

전문가 멘토링 세션

운영사 내/외부 전문 자문단 구성

멘토링 인력 Profile

주재걸 (Jaegul Choo)

2020- KAIST AI 대학원, 부교수

2019-2020 고려대학교 인공지능학과, 부교수

2015-2019 고려대학교 컴퓨터학과, 조교수

2013 Georgia Tech, Computational Science and Eng, 박사

2009 Georgia Tech, Electrical and Computer Eng, 석사



KAIST



곽대훈 멘토

김재철 AI 대학원
박사과정



김태성 멘토

김재철 AI 대학원
석박통합과정



최민석 멘토

김재철 AI 대학원
석박통합과정



형준하 멘토

김재철 AI 대학원
석박통합과정



이상현 멘토

김재철 AI 대학원
박사과정



김동민 멘토

김재철 AI 대학원
석사과정

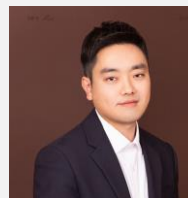
경진대회 전문 인력 Profile



김상만 상무

사업총괄

- 마인즈앤컴퍼니 AI Connect 사업부장
- NIPA 2022 인공지능 온라인 경진대회 사업총괄
- AI 스마트농업경진대회 사업PM
- 판교 AI Challenge 사업 PM
- 前 마인즈랩, AI 플랫폼 비즈니스 사업 및 영업 전담



김태훈 매니저

교육운영

- 마인즈앤컴퍼니 AI Data Scientist
- Kaggle Competitions Expert
- 서울대학교 경제학부 학사
- 2기 글창사/1기 이어드림 교육 운영
- NIPA 2022 인공지능 온라인 경진대회 운영



박성호 매니저

교육운영

- 마인즈앤컴퍼니 AI Data Scientist
- 前 UPenn Med 뇌과학 연구원, SCIE급 논문 1저자, SSCI급 논문 1저자
- UPenn 수학과 학사
- NIPA 2022 인공지능 온라인 경진대회 운영

실전경진대회

Kaggle Competition 수상

kaggle

Jane Street Market Prediction



- 참여 과제:
 - 주식 매수/매도 액션 예측
- 참여 기업명
 - 시너지에이아이
- 대회 결과
 - 총 참여 팀 4,245팀 중 123위를 기록

Cassava Leaf Disease Classification



- 과제
 - 병충해 이미지 판별 문제
- 참여 기업명
 - 딥인사이트
- 성과
 - 총 참여 팀 3,900팀 중 281위를 기록

PetFinder Pawpularity Contest



- 과제
 - 동물 사진 별 인기를 평가
- 참여 기업명
 - 담다디
- 성과
 - 총 참여 팀 3,537팀 중 121위를 기록

펀딩 및 비즈니스 확장



딥인사이트, 신용보증기금 '퍼스트펍권' 선정



[딥인사이트] '신용보증기금(KODIT)의 유망 스타트업 보증제도인 '퍼스트펍권' 창업기업에 선정, 3년간 최대 30억 원의 보증 및 각종 혜택을 지원' '글로벌창업사관학교 과정 중 '캐글' 대회 동메달 획득' <2022년 5월 10일>



[2021 인천창조경제혁신센터] 개인 맞춤형 소풍 매거진 '코코 쿠스' 서비스하는 담다디

매거진 한경

'글로벌창업사관학교 선정과 인천창조경제 혁신센터 스타트업 패스파인더 우수상을 선정', 'K스타트업 시애틀의 도움을 받아 현지 법인 설립 및 브랜드들과의 협업을 위한 미팅을 진행' <2021년 12월 13일>



이대목동병원 감염병 특화 개방형실험실 구축사업단 투자 유치 성공
인공지능 소프트웨어 개발 회사 시너지에이아이(대표 신태영) 투자 확정



[시너지에이아이] '총20억원의 투자 유치', '의료 영상으로부터 진단 정보 만들어 낼 수 있는 인공지능 소프트웨어 개발 스타트업', '글로벌창업사관학교 과정 중 '캐글'대회 참가 은메달 획득' <2022년 5월 20일>

Index

1. 모의경진대회 및 실전경진대회 커리큘럼 및 일정
2. AI CONNECT 플랫폼 소개 및 사용법 안내
3. 1차 모의경진대회 – (이미지) 흉부 CT 이미지 코로나 감염 여부 분류 과제
 - 과제 소개 (데이터 / 평가지표 / 제한사항)
 - 서버 사용법
 - 베이스라인 코드

AI CONNECT란?

AI Connect 마인즈앤컴퍼니의경진대회플랫폼



AI 경진대회
플랫폼



AI 생태계
활성화

AI 전문인력과 수요자들을 효율적으로 연결하여 AI 문제를 해결할 수 있는 AI 경진대회

특징1 AI 아이디어를 활용하려는 수요자와 다수의 아이디어를 제공하는 제안자들의 연결을 통해 객관적인 가치 평가가 가능함

특징2 클라우드소싱 기반 인공지능 문제해결 협업 플랫폼

AI 데이터 및 모델 활용 촉진, 개인/기업의 AI 역량을 증진시켜 전반적으로 AI 생태계 발전

특징3 AI 경진대회 플랫폼은 단순히 순위를 위한 경합의 장이 아닌 모델 제고를 통한 개인/기업 역량 증진과 다양한 모델을 테스트 가능한 기획의 장

- 문제해결이 필요한 분야에 대한 문제 정의 및 설계
- 데이터셋 제공 (필요에 따라 전처리 및 정제 진행)
- 상금 (수요기업별 협의)
- 인센티브 추가 제공 (상장, 채용기회 등)



- 해결 가능한 문제에 대하여 과제 참여 신청
- 문제 해결을 위해 최적의 AI 모델 개발 및 구축
- 과제 우승에 따른 상금 및 인센티브 획득
- AI 모델 학습 코드 및 문제 해결 노하우 제공



AI 경진대회 수행 이력

고객사	프로젝트명	기간
정보통신산업진흥원	NIPA 2022년 인공지능 온라인 경진대회	2022
중소벤처기업진흥공단	2021년 스타트업 청년인재 이어드림 프로젝트	2022
마인즈랩	NIA AI 데이터 구축 해커톤	2021
중소벤처기업진흥공단	글로벌창업사관학교 AI 실습 프로젝트 교육 운영	2021
농림수산물교육문화정보원	2021년 스마트농업 인공지능 경진대회	2021
경기도경제과학진흥원	판교 AI Challenge 운영 용역	2021
한국축산데이터	Animal Datathon Korea 2021	2021
정보통신산업진흥원	NIPA 2021년 인공지능 온라인 경진대회	2021
삼성서울병원	치아번호 식별 경진대회	2021
서울대학병원	Sleep AI Challenge	2021
아주대학교병원	구강암 판별 경진대회	2021
중소벤처기업진흥공단	AI 실습 경진대회 운영 용역	2020
IITP	2020년 인공지능 그랜드 챌린지	2020
정보통신산업진흥원	NIPA 2020년 AI 문제해결 공모전	2020

1,2기 글로벌창업사관학교 AI 실습 프로젝트 교육 운영

운영기간 '20.11~ '21.02
 '21.09~ '21.12
 발주처 중소벤처기업진흥공단



사업개요

- 기초/심화반 맞춤형 AI 이론 및 실습 교육
- 모의경진대회를 위한 문제 발굴 및 정의, 데이터셋 제작, 베이스라인 및 채점 코드 제작
- 글로벌 AI 경진대회(캐글) 참여 지원
- 참여하는 실전 과제에 최적화된 특강과 멘토링 제공

AI CONNECT 기능

1



참가자 접수 모집

- 플랫폼을 통하여 참가자 모집 및 접수 절차 수행
- 폐쇄형 경진대회 참가팀에게는 가상화된 별도의 개발 환경 자동으로 매핑 및 제공

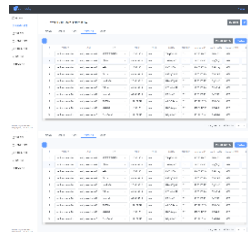


2



GPU 서버 연동

- 참가팀별 GPU 서버를 할당하여, 별도 서버 운영이 가능하며, 필요시 자체 보유 서버 지원



3



채점 서버

- 문제별 채점코드에 따른 설정 및 운영
- 제출 결과 값에 대한 평가 및 채점 진행
- 참가자별 제출 결과 히스토리 관리



4



공지 사항

- 문제 정의서, 데이터 설명, 평가지표, 순위 결정 규정 등의 다양한 공지사항을 참가자들에게 전달



5



실시간 리더보드

- 리더보드를 통해 참가자 스코어를 준 실시간 업데이트하여 관련 정보 제공
- 참가자는 본인의 스코어 및 순위 확인 지원



6

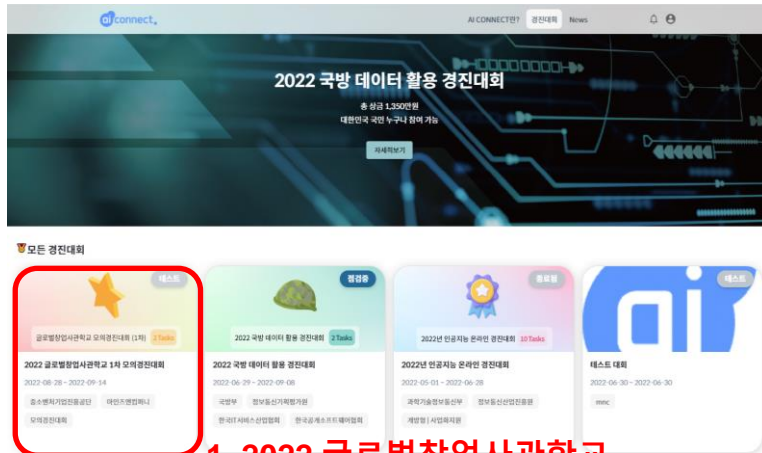


Q&A

- 플랫폼 내부 Q&A 게시판을 운영하여 참가자 문의사항 처리
- 대회별/과제별 각 별도의 게시판 운영으로 혼동 없이 질문 및 건의사항 응대



AI CONNECT 과제 참여 방법



1. 2022 글로벌창업사관학교
1차 모의경진대회 Click!



3. 과제 참여 Click!
(일단 모든 과제 참여 부탁드립니다)



2. 참여할 과제 Click!



AI CONNECT 결과 제출 방법

‘결과제출’ 탭을 통해
추론 결과 파일 제출

과제 참여중

HOME

과제 설명

리더보드

팀 구성

대회 공지/문의

데이터 설명/다운로드

코드공유

결과제출

과제 공지/문의

과제참여위소

일정 테스트

참여인원 2명

결과제출 1시간 1회(1일 최대 24회)의 ‘결과제출’ 제한

결과제출 제한사항

파일크기 최대 1MB

제출횟수 하루 최대 24회까지 제출 가능

파일첨부

최대 1MB

제출설명 실험 및 코드 버전 관리를 위해 ‘제출설명’ 기능 활용

제출 파일에 대한 설명을 작성하세요.

제출하기

제출현황 ↻

#	제출파일	Public Score	성공여부	제출시간	최종선택
제출이력이 없습니다.					

Public score / CV score 등을 고려하여 제출 결과물 중 2개의 결과물 ‘최종선택’ 선택하지 않을 시 Public score가 가장 높은 두 개의 결과물이 자동으로 최종선택됨

Index

1. 모의경진대회 및 실전경진대회 커리큘럼 및 일정
2. AI CONNECT 플랫폼 소개 및 사용법 안내
3. 1차 모의경진대회 – (이미지) 흉부 CT 이미지 코로나 감염 여부 분류 과제
 - 과제 소개 (데이터 / 평가지표 / 제한사항)
 - 서버 사용법
 - 베이스라인 코드

데이터셋

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

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

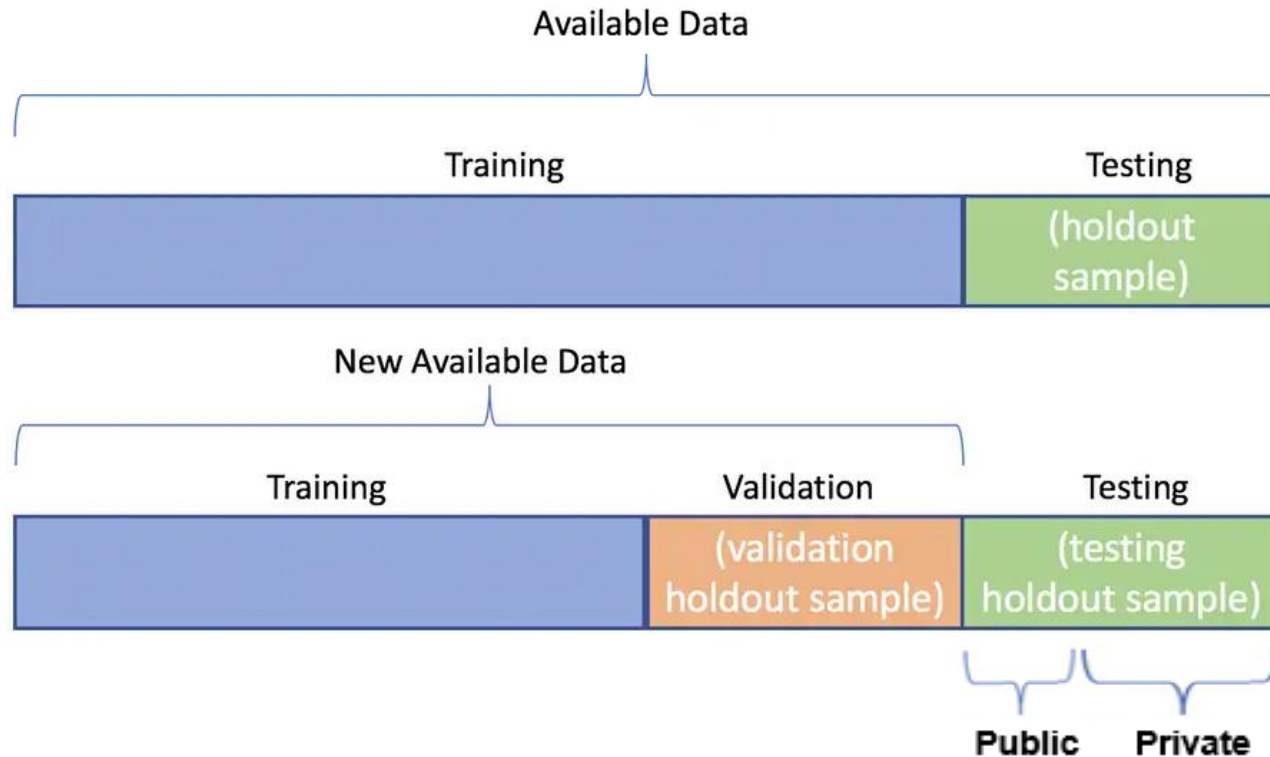
$$\text{Precision} = \frac{TP}{TP + FP}$$

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

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

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

Public / Private Score



- **Available Data -> Training / Testing Data** : 학습된 모델의 일반화 성능을 판단하기 위함
- **Training Data -> Training / Validation Data** : 일반화 성능을 높이는 방향으로 모델을 학습하기 위함
- **Testing Data -> Public / Private Data** : 리더보드를 통해 정답을 유추해내는 행위를 막기 위함

제한 사항

- 외부 데이터 사용 불가
- 이미 학습된 모델의 weight 사용 불가
 - 모델 라이브러리를 통해 이미 학습이 진행된 모델을 로드하여 사용할 수 없음
 - .pt, .pth, .h5 등의 pre-trained 가중치 파일을 업로드하여 사용할 수 없음
 - Transfer learning, Fine tuning 불가함
 - 다만, 모델 아키텍처는 가져와서 사용 가능함 (pretrained=False)
- 과제별 우승팀 코드는 코드 공유 탭에 게시 / 우승팀 코드 리뷰 세션
- 결과 제출 제한: 1시간 1회 (1일 최대 24회)

Index

1. 모의경진대회 및 실전경진대회 커리큘럼 및 일정
2. AI CONNECT 플랫폼 소개 및 사용법 안내
3. 1차 모의경진대회 – (이미지) 흉부 CT 이미지 코로나 감염 여부 분류 과제
 - 과제 소개 (데이터 / 평가지표 / 제한사항)
 - 서버 사용법
 - 베이스라인 코드

서버 사양 소개

서버 사양

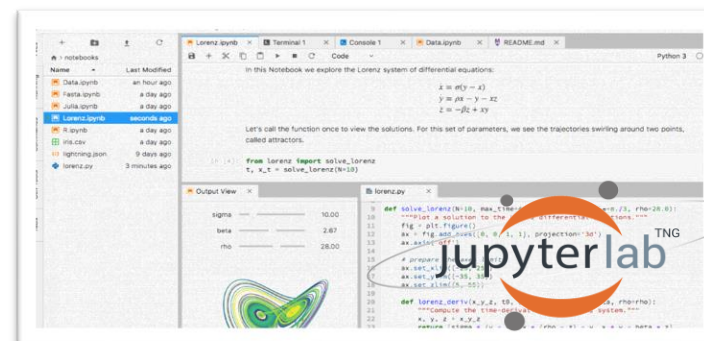
[제공 서버 * 팀당 1EA]
10 Core, 96GB mem, Nvidia T4(15GB)

[공통사양]

- Ubuntu : 18.04.5
- Python : 3.8.5
- Pytorch : 1.7.1
- cuda : 11.1
- nvidia driver : 455.32.00
- cudnn : 8.0.4

JupyterLab 접속법

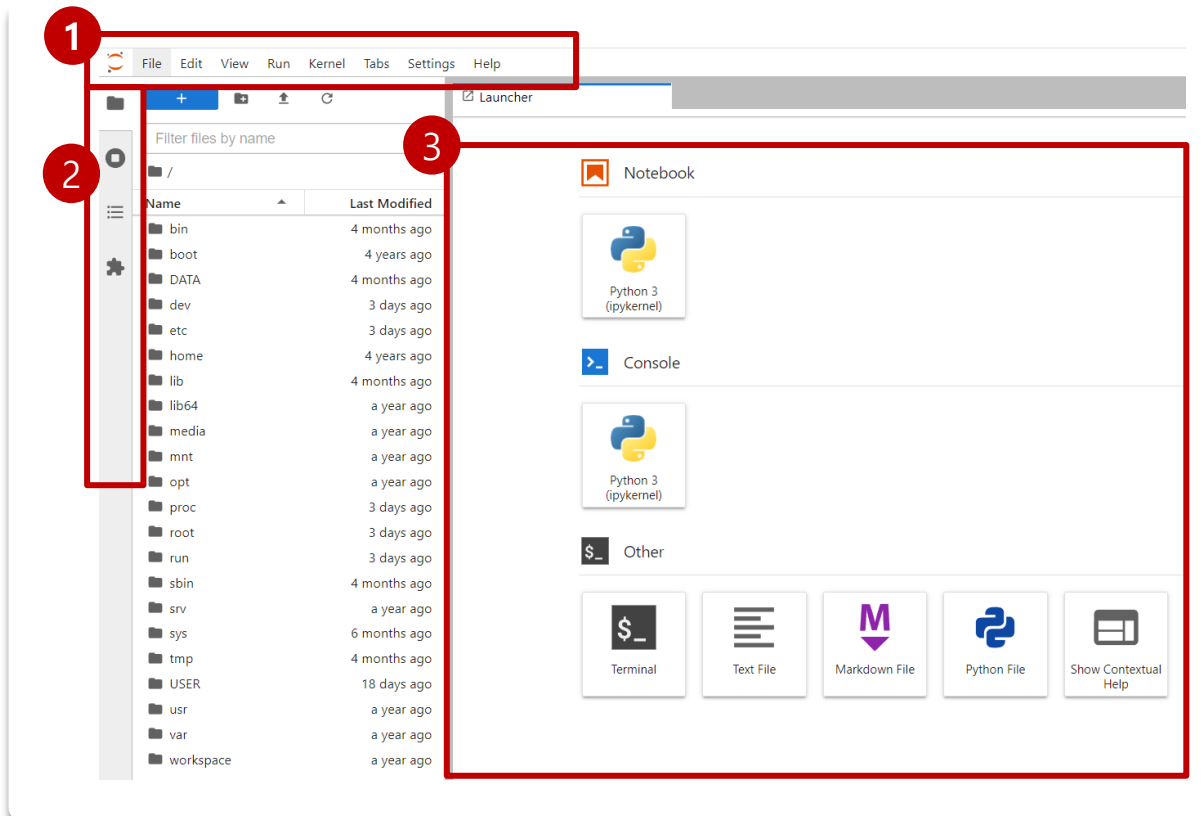
- 메일, 슬랙 DM으로 제공된 서버 정보 확인 (IP, port, P/W)
- 웹브라우저 창에서 [IP]:[port(Jupyter)] 입력 (예: 115.71.1.85:50000)
- JupyterLab 비밀번호 입력
- 접속 완료
- 서버 내 작업 공간
 - * /USER 폴더에 파일 용량 (약 90GB) 할당이 되어 있으니 해당 폴더에서만 작업 부탁드립니다.



JupyterLab - 메인 화면

화면 예시

화면 설명



1 메뉴바

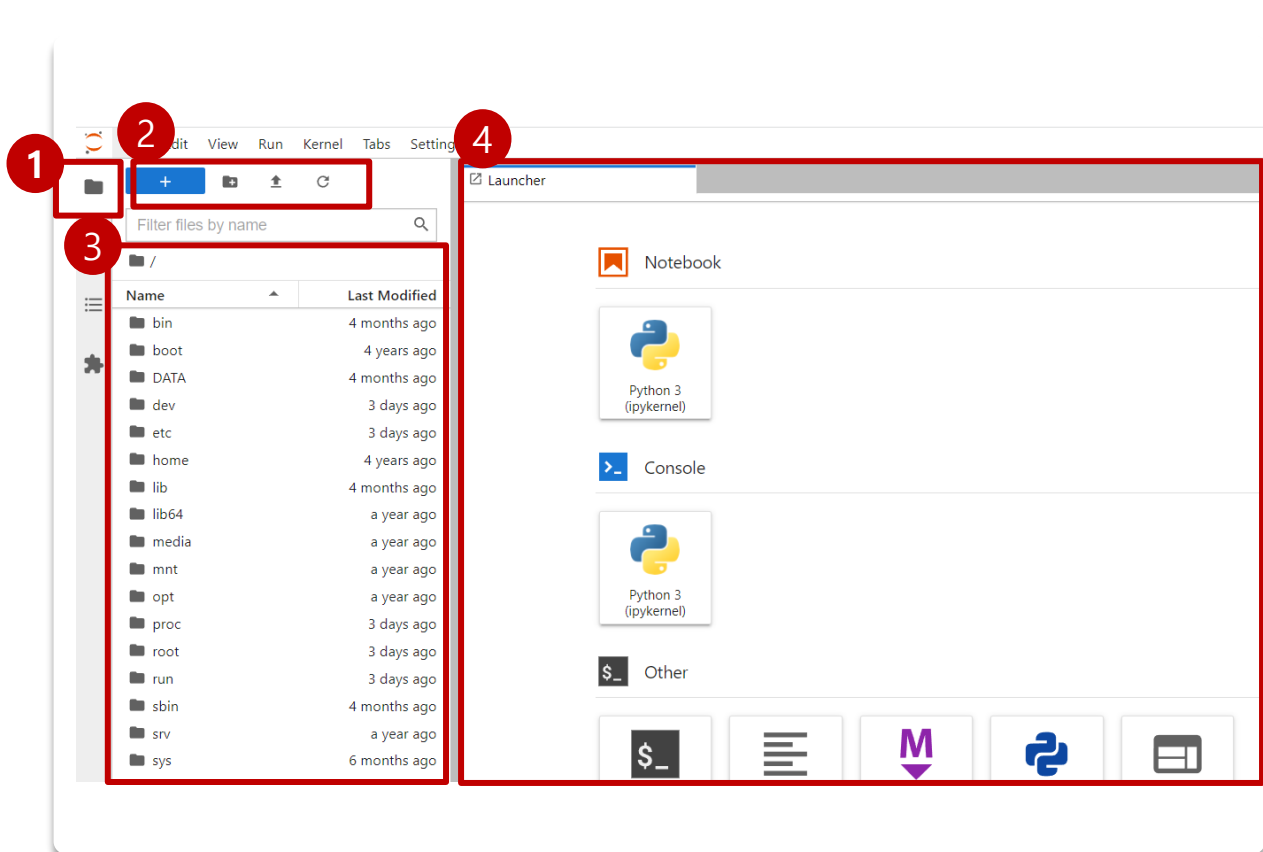
2 메뉴 아이콘

3 실행 런처
(Python, Terminal 만 주로 이용)

JupyterLab - 파일 관리 화면

화면 예시

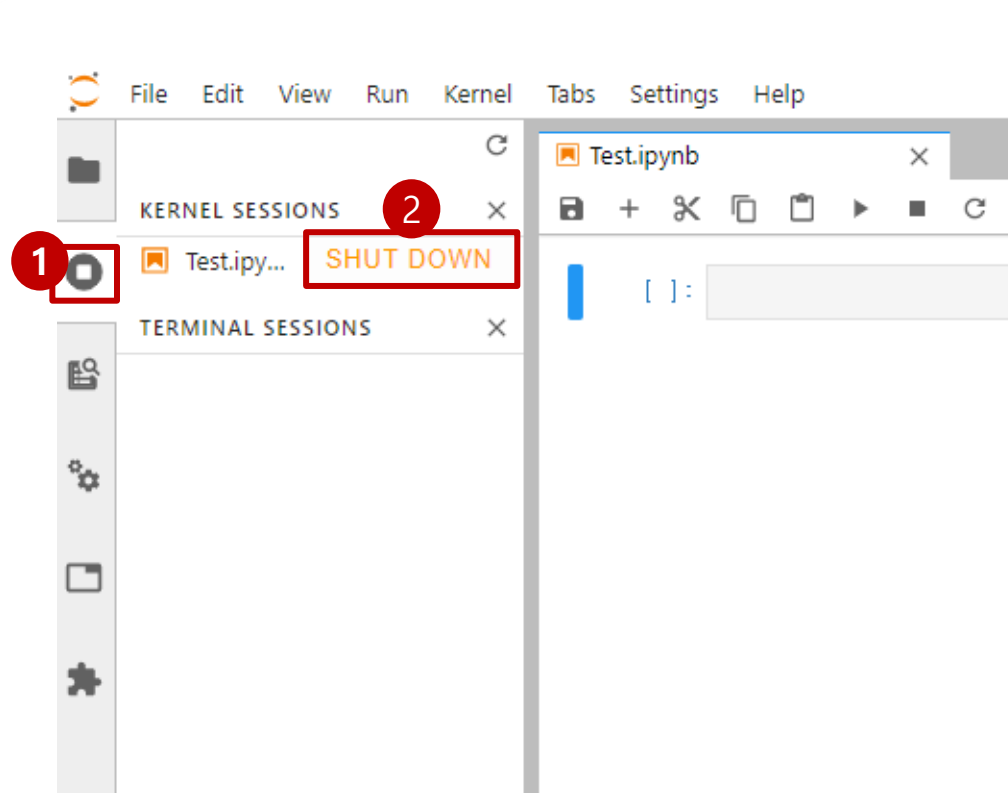
화면 설명



- 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

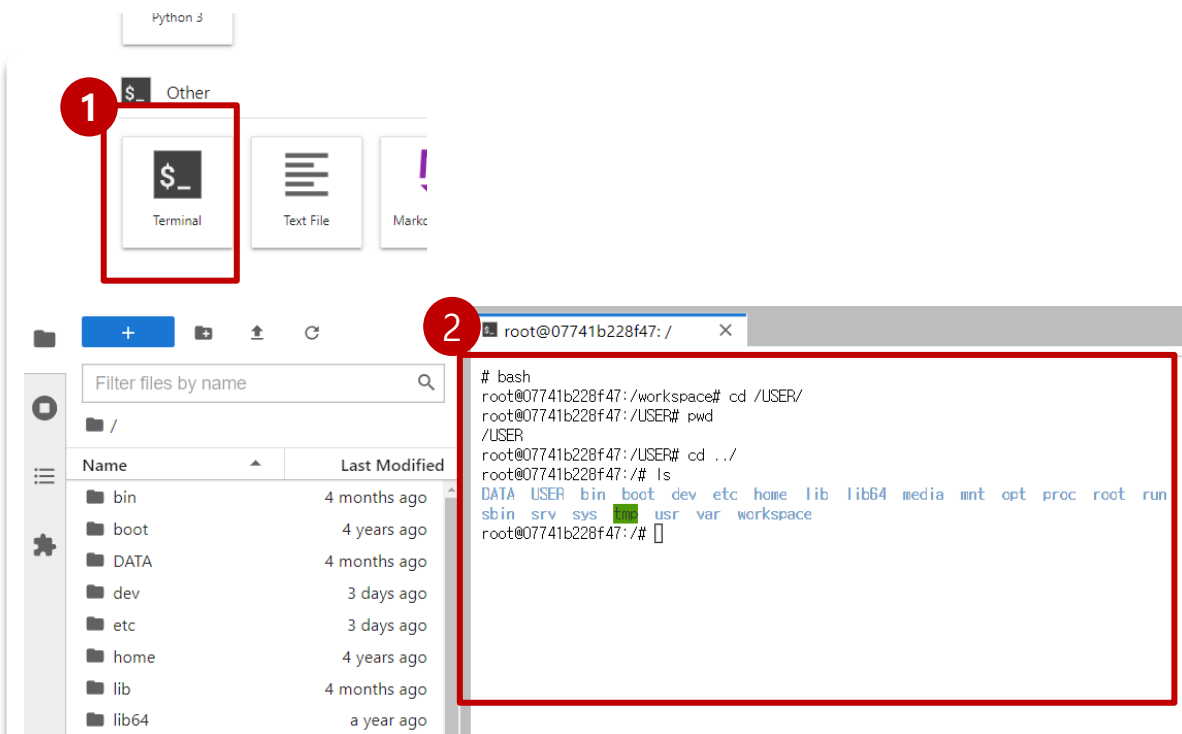


화면 설명

- 1 커널 관리 버튼
- 2 커널 종료 버튼

JupyterLab - 터미널 접속 방법

화면 예시



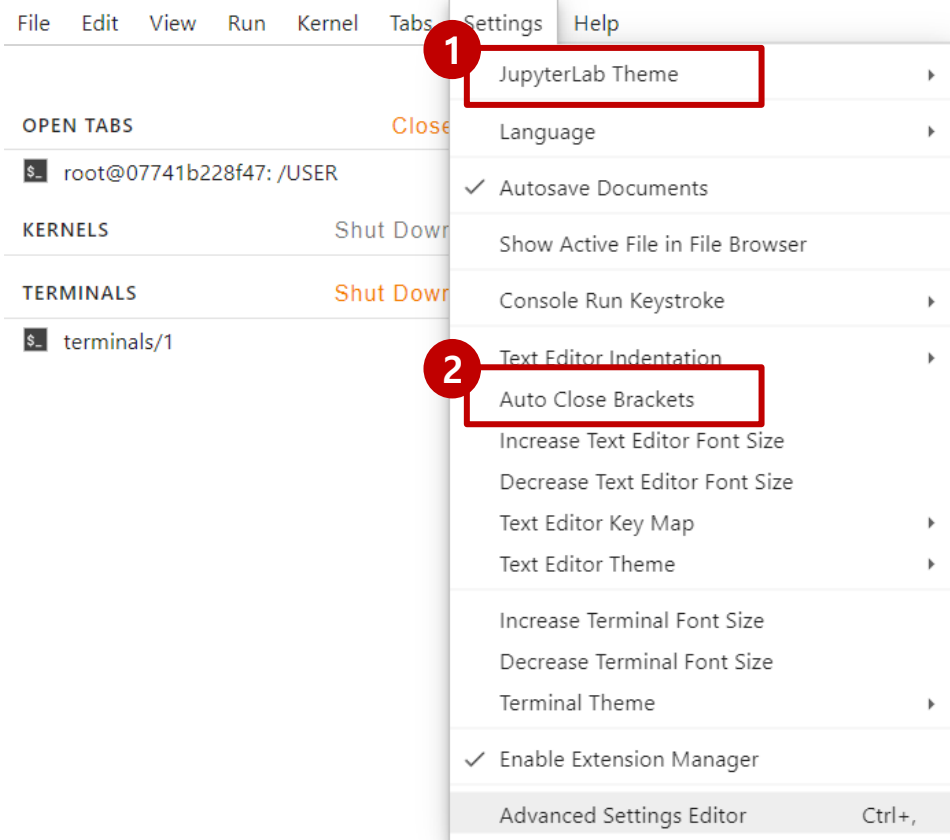
화면 설명

- 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
설정 가능합니다.
(해당 포트에만 영향)

라이브러리 설치 방법

Libraries 설치

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

anaconda 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/site-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 platform_machine != "arm64" and python_version < "3.10" in /opt/conda/lib/python3.8/site-packages (from pandas) (1.19.2)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.8/site-packages (from 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
```


리소스 관리

- GPU 사용량 : `nvidia-smi`

```
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      | 00000000:18:00.0 Off |   0          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
```

- CPU 사용량 : `top`

```
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
  29 root      20   0 1215252 584844 3948 S   0.7   0.2   237:22.96 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
```

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. 모의경진대회 및 실전경진대회 커리큘럼 및 일정
2. AI CONNECT 플랫폼 소개 및 사용법 안내
3. 1차 모의경진대회 – (이미지) 흉부 CT 이미지 코로나 감염 여부 분류 과제
 - 과제 소개 (데이터 / 평가지표 / 제한사항)
 - 서버 사용법
 - 베이스라인 코드

베이스라인 구성

베이스라인은 입교팀에게 할당된 서버에 업로드 되어 있고,
AI CONNECT 플랫폼 해당 과제 페이지에서도 다운로드 가능합니다.

코드 구성	설명
# 필수 라이브러리 불러오기	베이스라인에 필요한 라이브러리 로드
# 하이퍼파라미터 및 기타인자 설정	데이터 경로 / 시드 / 디바이스 / 하이퍼파라미터 등 설정
# Dataset 정의	데이터를 모델 인풋 형태로 가공
# 모델 정의	모델 아키텍처 선언
# Utils 정의	EarlyStopper, Trainer 클래스, 평가지표 함수 정의
# 모델 학습	에폭 단위로 학습 진행
# 추론	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

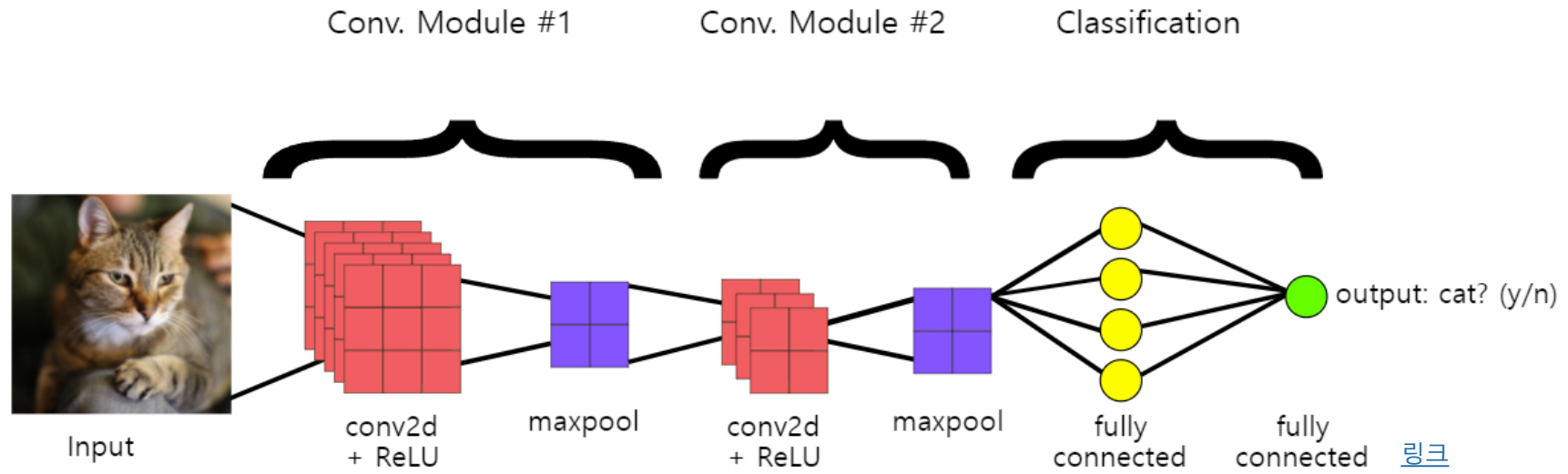
RANDOM_SEED = 2022

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

CustomDataset() 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>

$N = 32$

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

$padding = (0,0)$

$kernel_size = (5,5)$

$in_channels = 3$

$out_channels = 8$

Shape:

Conv2d & MaxPool2d 계산방법 동일, [링크](#)

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

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

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

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

(배치, 채널, 높이, 너비)

Input : (32, 3, 128, 128) \Rightarrow

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

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

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

$$H_{out} = W_{out}$$

$$= \left\lfloor \frac{124 + 2 \times 0 - 1 \times (2 - 1) - 1}{2} + 1 \right\rfloor$$

$$= 62$$

(배치, 채널, 높이, 너비)

Input: (32, 8, 124, 124) \Rightarrow

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>

$N = 32$

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

$padding = (0,0)$

$kernel_size = (5,5)$

$in_channels = 8$

$out_channels = 25$

Shape:

Conv2d & MaxPool2d 계산방법 동일, [링크](#)

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

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

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

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

(배치, 채널, 높이, 너비)

Input: (32, 8, 62, 62) \Rightarrow

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

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

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

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

(배치, 채널, 높이, 너비)

Input: (32, 25, 58, 58) \Rightarrow

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')
```



End of document