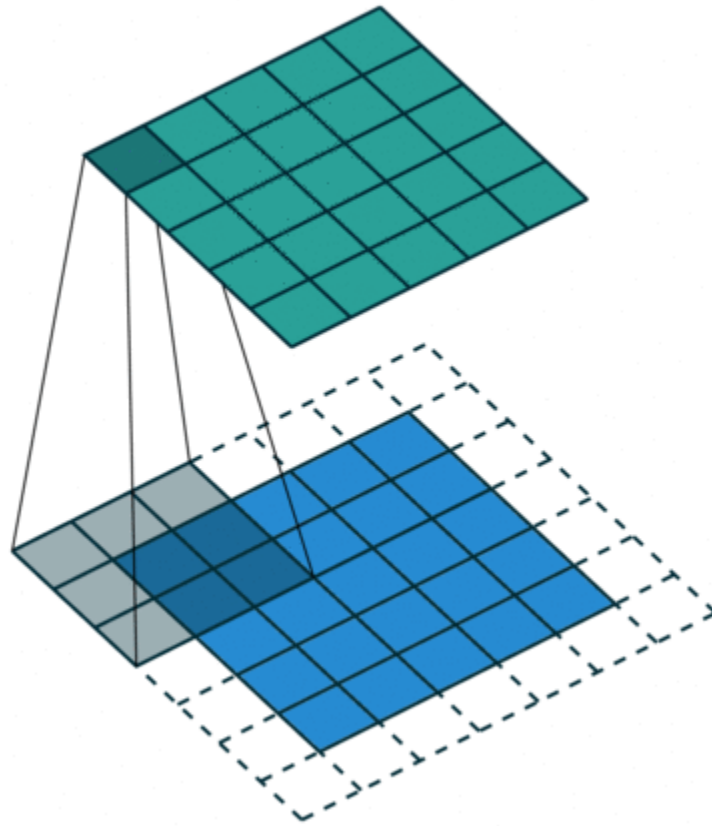


3주차 미팅 발표 자료

2021년 10월 6일 수요일

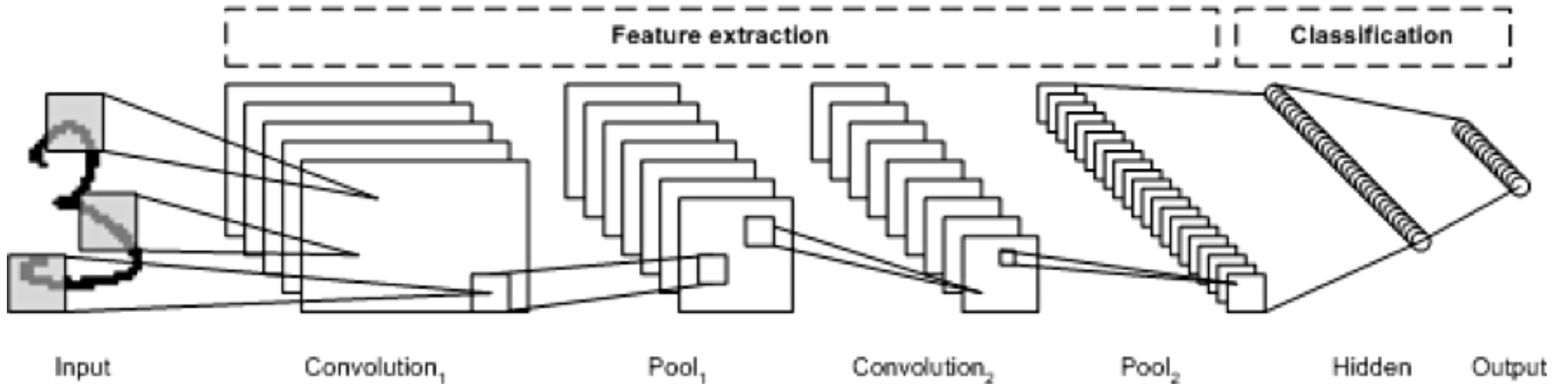
1. ROC 그래프 - 선형?
2. Channel-wise pooling 기능 (3~6)
3. Cross Entropy Loss (7~13)
4. BeatGAN, FRSKD model 코드 리뷰
5. 결합할 부분 : FRSKD의 knowledge distillation 코드 리뷰

Convolution 기본적인 형태



2D convolution using a kernel size of 3, stride of 1 and padding

CNN : 이미지의 특징 추출하는 부분 & 클래스 분류하는 부분



특징 추출 영역은 **Convolution Layer**와 **Pooling Layer**을 여러 겹 쌓는 형태로 구성
CNN 마지막 부분에는 이미지 분류를 위한 **Fully Connected** 레이어 추가

CNN 용어 - Channel

컬러 사진은 이미지 픽셀을 **RGB 3개**의 실수로 표현한 **3차원** 데이터
흑백 사진은 흑백 명암만을 표현하는 **2차원** 데이터로, **1개 채널**로 구성

Convolution Layer에 **n개**의 필터가 적용되면 출력 데이터는 **n개**의 채널을 갖게 된다.

RED Channel



Green Channel

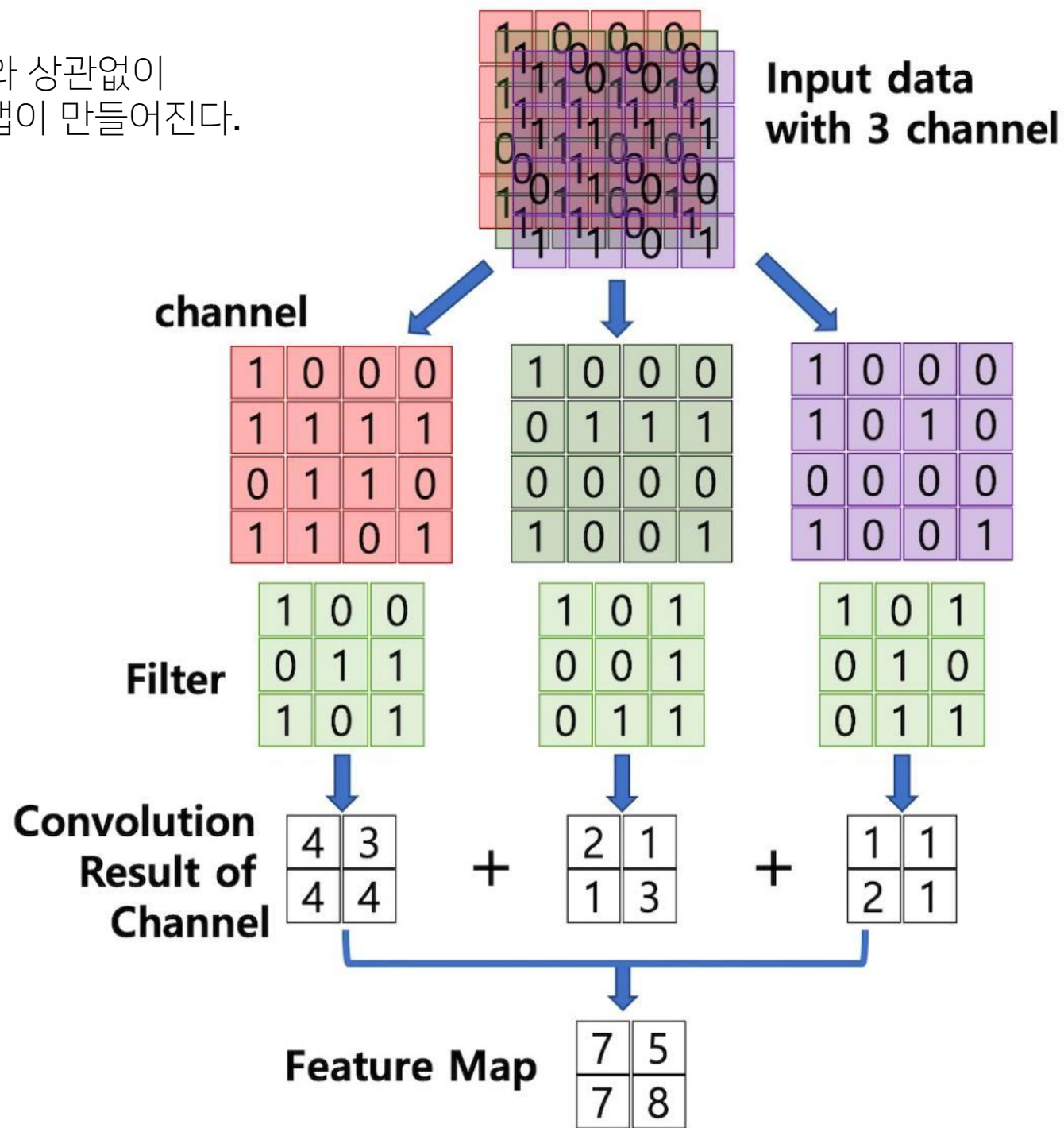


Blue Channel



이미지 출처: [https://en.wikipedia.org/wiki/Channel_\(digital_image\)](https://en.wikipedia.org/wiki/Channel_(digital_image))

입력 데이터는 채널 수와 상관없이
필터 별로 1개의 피쳐 맵이 만들어진다.



Cross-Entropy Loss Function

최적화 문제를 풀 때 목적함수를 정한다.

손실을 최소화하도록 목적함수를 표현 → 손실함수

Squared Loss $E(\mathbf{w}) \equiv \frac{1}{2} \sum_{d \in D} (y_d - \hat{y}_d)^2$

Mean Squared Loss $E(\mathbf{w}) \equiv \frac{1}{2} \frac{1}{|D|} \sum_{d \in D} (y_d - \hat{y}_d)^2$

$|D|$ 학습집합 크기

Cross-Entropy Loss Function

$$P(x)$$

확률변수

$$\frac{1}{p(x)}$$

사건이 드물게 발생
→ 정보가 커야 함

$$\frac{1}{p(x)} \times \frac{1}{p(y)}$$

두 사건에 대한 정보

$$\log \frac{1}{p(x)} + \log \frac{1}{p(y)}$$

곱하기보다 더하기가 직관적,
로그는 곱셈을 덧셈으로 변경

Cross-Entropy Loss Function

$$P(x)$$

확률변수

$$\frac{1}{p(x)}$$

사건이 드물게 발생
→ 정보가 커야 함

$$\frac{1}{p(x)} \times \frac{1}{p(y)}$$

두 사건에 대한 정보

$$\log \frac{1}{p(x)} + \log \frac{1}{p(y)}$$

곱하기보다 더하기가 직관적,
로그는 곱셈을 덧셈으로 변경

$$\log \frac{1}{p(x)}$$

$$\log p(x)^{-1}$$

$$-\log p(x)$$

Cross-Entropy Loss Function

$$E[X] = \sum_x xP(x)$$

확률변수X의 기댓값 계산 공식

$$E[X] = \int xp(x)dx$$

(확률질량함수, 확률밀도함수)

$$E[aX + b] = \sum_x (ax + b)p(x)$$

f(x)로 일반화 →

$$E[f(X)] = \sum_x f(x)p(x)$$

Cross-Entropy Loss Function

$$E[X] = \sum_x xp(x)$$

$$E[-\log p(x)] = \sum_x -\log p(x)p(x)$$

ENTROPY \rightarrow

$$\sum_x -p(x)\log p(x)$$

Cross-Entropy Loss Function

ENTROPY →

$$\sum_x -p(x) \log p(x)$$

Cross-Entropy? 다른 확률을 곱해서 Entropy 계산한 것

ex. 0 또는 1만 가지는 확률변수 X 에 대해,

$$\begin{aligned} \text{entropy} = & -p(X=0) \log(p(X=0)) - p(X=1) \log(p(X=1)) \\ & -p(X=0) \log(q(X=0)) - p(X=1) \log(q(X=1)) \end{aligned}$$

Cross-Entropy Loss Function

ENTROPY →

$$\sum_x -p(x) \log p(x)$$

Cross-Entropy? 다른 확률을 곱해서 Entropy 계산한 것

$$E(\mathbf{w}) \equiv \frac{1}{|D|} \sum_{d \in D} (-y_d \log(\hat{y}_d) - (1 - y_d) \log(1 - \hat{y}_d))$$

0~1사이 0 or 1

캡스톤디자인 연구를 위한 GPU 대여 요청 및 승인

경희대학교 GPU 대여 http://swedu.khu.ac.kr/html_2018/06/05.php
머신 타입 요청, 다음 날 승인됨

☰ Ocean-User

사용 가능한 머신

승인 g3.small (캡디) / 1 개 - [2021.09.30 ~ 2021.11.30]

승인 inst / 1 개 - [2021.09.30 ~ 2021.11.30]

요청한 머신

> [서버요청 관리 페이지로 이동](#)

인스턴스 외부에서 사용하기 위해 web 대쉬보드 url 전달하여 ocean 초기화, ocean 로그인 진행
가이드 : https://ocean-cli.herokuapp.com/Getting_Started.html#id4

인스턴스 생성 가이드

5 인스턴스 생성

- 3) **인스턴스 관리** 화면에서 인스턴스 리스트를 확인할 수 있습니다.
- 4) SSH Port를 통해 SSH 접속, 편집기 링크를 클릭하여 VS Code를 접속할 수 있습니다.

인스턴스 관리								새 인스턴스 추가하기
진행상태	인스턴스	사용목적	사용 이미지	SSH Port	서버사양	Volumes	편집기 링크	
생성완료	instance	캡스톤디자인2 (주제: 딥러닝 모델 푸르닝)	myaiocan/pytorch:latest	32003	inst	vol1	링크	삭제

< 1 >

- ▶ SSH 및 VSCode의 기본 비밀번호는 이메일 주소이며, 반드시 변경해주세요.
- ▶ 볼륨은 `/root/<볼륨이름>`으로 마운트 됩니다.
- ▶ 인스턴스 및 작업에 마운트 되는 볼륨 이외의 곳에 저장되는 파일은 인스턴스, 작업이 삭제 되면 삭제됩니다. 유지되어야 하는 파일은 반드시 볼륨에 저장해주세요.


인스턴스 생성

☰ Ocean-User



인스턴스 관리

새 인스턴스 추가하기

진 행 상 태	인스턴 스	사용목적	사용 이미지	SSH Port	서버사 양	Volumes	편 집 기 링 크
생 성 완 료	instance	캡스톤디자인1 (주제: ECG데이터 로부터 부정맥을 예측하는 딥러닝 알고리즘 개발 및 연구)	myaiocan/pytorch:latest	30122 	<div>inst</div>	<div>vol1</div>	링 크 <div>삭제</div>

SSH Port를 통해 SSH 접속, 편집기 링크를 클릭하여 VS Code 접속

Welcome to code-server

Please log in below. Check the config file at `~/.config/code-server/config.yaml` for the password.



SUBMIT

처음엔 편집기 링크 아무리 클릭해도 권한이 없다고 떴는데
많은 삽질을 하였고 (구글독서에 정리..) 결국 접속이 되었습니다!

작업 환경 확인 : OS - Ubuntu 18.04

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```

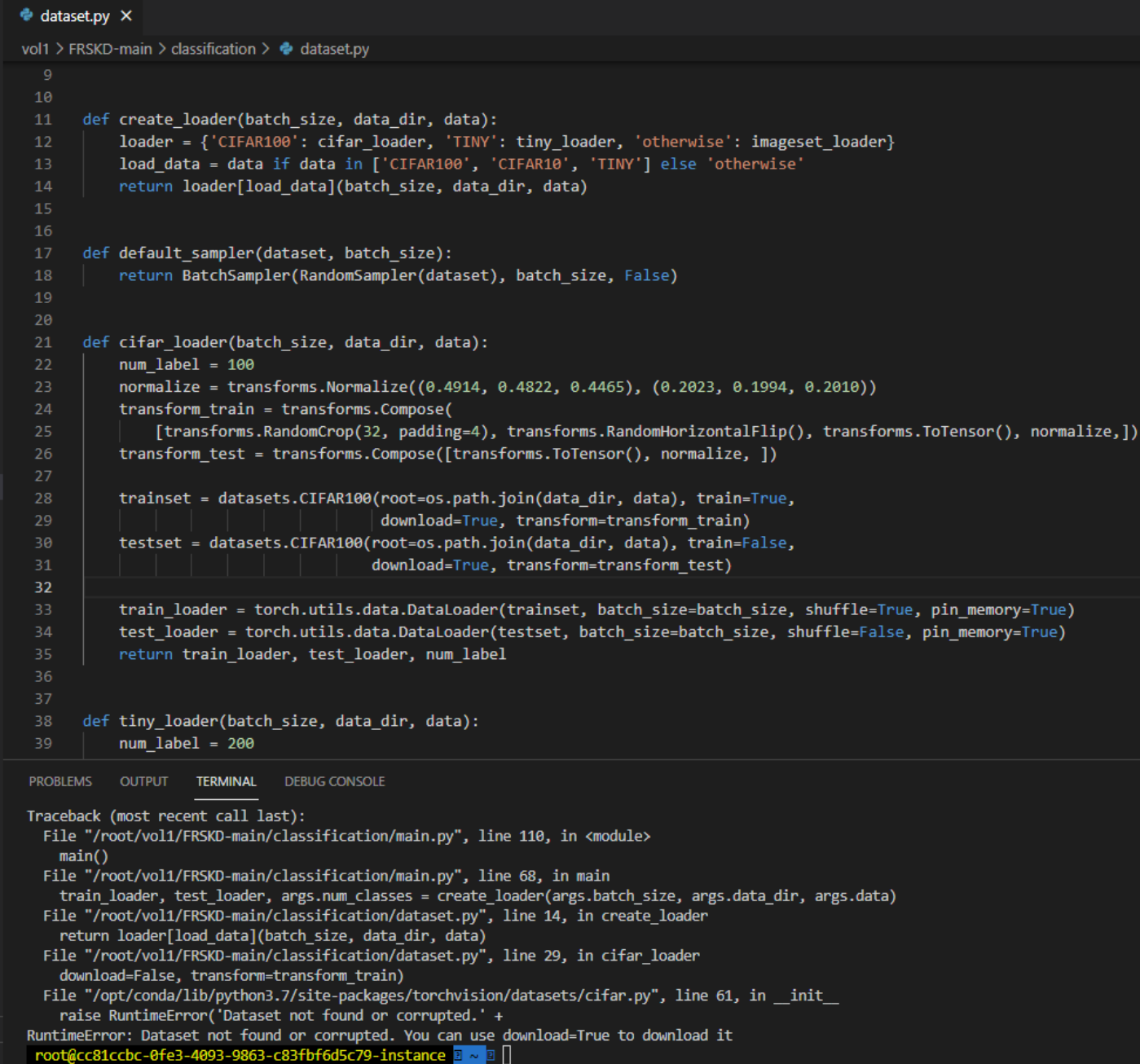
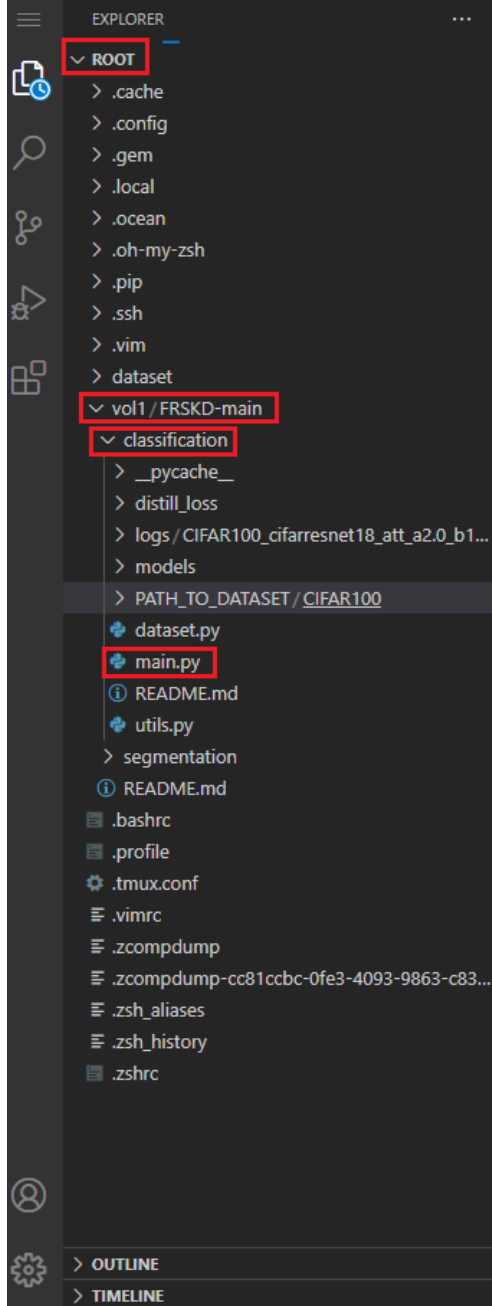
      .-/+00SSSS00+/- .
    `:+SSSSSSSSSSSSSSSS+:`
  -+SSSSSSSSSSSSSSSSSSyySSSS+-
    .0SSSSSSSSSSSSSSSSSSdMMMMySSSS0.
  /SSSSSSSSSSShdmmNNmmyNNMMhSSSSSS/
+SSSSSSSSShmydMMMMMMNddddySSSSSSSS+
/SSSSSSSSShNNMMyhhyyyymNNMMhSSSSSSSS/
.SSSSSSSSSdMMMNhSSSSSSSSShNNMMdSSSSSSSS.
+SSSSShhhyNNMMySSSSSSSSSSSyNNMMySSSSSSSS+
0SSyNNMMNyMMhSSSSSSSSSSSSShmmhSSSSSSSS0
0SSyNNMMNyMMhSSSSSSSSSSSSShmmhSSSSSSSS0
+SSSSShhhyNNMMySSSSSSSSSSSyNNMMySSSSSSSS+
.SSSSSSSSSdMMMNhSSSSSSSSShNNMMdSSSSSSSS.
/SSSSSSSSShNNMMyhhyyyymNNMMhSSSSSSSS/
+SSSSSSSSSdmydMMMMMMNddddySSSSSSSS+
/SSSSSSSSSSShdmmNNmmyNNMMhSSSSSS/
    .0SSSSSSSSSSSSSSSSSSdMMMMySSSS0.
  -+SSSSSSSSSSSSSSSSSSyySSSS+-
    `:+SSSSSSSSSSSSSSSS+:`
      .-/+00SSSS00+/- .

```

root@cc81ccbc-0fe3-4093-9863-c83fbf6d5c79-instance

OS: Ubuntu 18.04.3 LTS bionic x86_64
Host: G292-Z20-00 0100
Kernel: 4.15.0-136-generic
Uptime: 4 hours, 47 mins
Packages: 317
Shell: zsh 5.4.2
Terminal: vscode
CPU: AMD EPYC 7282 16- (16) @ 2.800GHz
Memory: 4922MiB / 193313MiB





VS Code에서 FRSKD-main 폴더 넣기

방법:

인스턴스 관리 →
편집기 링크 '링크클릭' →
접속한 code-server에서
/root/vol1/ 폴더에
깃헙에서 다운받은
FRSKD-main 폴더 삽입 →

작업 환경 확인 : OS - Ubuntu 18.04

우분투에 설치되어 있는 CUDA 버전 확인 (nvcc -V 명령어 이용)

```
root@cc81ccbc-0fe3-4093-9863-c83fbf6d5c79-instance ~/vol1/tmp$ nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Sun_Jul_28_19:07:16_PDT_2019
Cuda compilation tools, release 10.1, V10.1.243
```

cuda 10.1이므로,
PyTorch 사이트에서 해당 버전과 맞는 pytorch를 다운
<https://pytorch.org/get-started/previous-versions/>

```
conda install pytorch==1.7.1 torchvision==0.8.2 torchaudio==0.7.2 cudatoolkit=10.1 -c pytorch
```

작업 생성 가이드

6 작업 생성

작업은 인스턴스에서 작성한 코드 및 개발 환경을 그대로 GPU있는 버전으로 머신타입만 변경하여 동작하도록 합니다. 인스턴스와 유사하며 차이점은 실행 스크립트만 실행시키는 1회성 컴퓨팅 자원입니다. 인스턴스 내부의 Ocean-CLI에서도 간편하게 제출이 가능합니다.

- 1) 메뉴 바의 **작업 관리** ▶ **새 작업 추가하기** 버튼을 클릭하여 새 작업 요청 대화상자를 엽니다.
- 2) 인스턴스와 동일하게 입력하며 추가적으로 **실행 스크립트**, **반복 횟수**를 입력합니다. 실행 스크립트는 리눅스 커맨드에서 실행 가능한 모든 명령이 가능하며, 반복 횟수만큼 같은 작업이 여러 번 제출됩니다.

작업 생성



작업 관리

새 작업 추가하기

진행상태	이름	이메일	사용 목적	사용이미지	실행 명령어	서버사양	Volumes
생성 완료 0시간 9분	frskd	vlfdus0401@khu.ac.kr	캡스톤디자인	myaioccean/pytorch:latest	python /root/vol1/FRSKD-main/classification/main.py - -data_dir PATH_TO_DATASET --data CIFAR100 -- batch_size 128 --alpha 2 --beta 100 --aux none -- aux_lamb 0 --aug none --aug_a 0	g3.small (캡디)	vol1

삭제

작업 실행 - 로그



로그

2021-10-05T11:42:39.179707980Z : Downloading <https://www.cs.toronto.edu/~kriz/cifar-100-python.tar.gz> to
PATH_TO_DATASET/CIFAR100/cifar-100-python.tar.gz

2021-10-05T11:43:13.947549672Z : Extracting PATH_TO_DATASET/CIFAR100/cifar-100-python.tar.gz to PATH_TO_DATASET/CIFAR100

2021-10-05T11:43:16.211702198Z : Files already downloaded and verified

2021-10-05T11:43:16.570930586Z :

2021-10-05T11:43:16.570930586Z : 0it [00:00, ?it/s]


2021-10-05T11:43:16.570930586Z : 0% | 0/169001437 [00:01<?, ?it/s]


2021-10-05T11:43:16.570930586Z : 0% | 8192/169001437 [00:02<1:23:54, 33566.36it/s]


2021-10-05T11:43:16.570930586Z : 0% | 40960/169001437 [00:02<1:04:59, 43331.17it/s]

확인


작업 실행 - 로그


 로그


2021-10-05T11:43:16.571019244Z : 97% 164659200/169001437 [00:33<00:00, 5330371.76it/s]


2021-10-05T11:43:16.571019244Z : 98% 165339136/169001437 [00:33<00:00, 5619117.77it/s]


2021-10-05T11:43:16.571019244Z : 98% 165920768/169001437 [00:33<00:00, 5543204.41it/s]

2021-10-05T11:43:16.571019244Z : 99% 166502400/169001437 [00:34<00:00, 5602004.55it/s]

2021-10-05T11:43:16.571019244Z : 99% 167108608/169001437 [00:34<00:00, 5646736.34it/s]

2021-10-05T11:43:16.571019244Z : 99% 167682048/169001437 [00:34<00:00, 5445852.54it/s]

2021-10-05T11:43:16.571019244Z : 100% 168337408/169001437 [00:34<00:00, 5569193.88it/s]

2021-10-05T11:43:16.571019244Z : 100% 168902656/169001437 [00:34<00:00, 5503282.50it/s]--alpha 2.0

확인

하지만....

CUDA와 PyTorch 버전 충돌

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

RuntimeError: Dataset not found or corrupted. You can use download=True to download it

```
root@cc81ccbc-0fe3-4093-9863-c83fbf6d5c79-instance ~ conda install pytorch torchvision torchaudio cudatoolkit=11.1 -c pytorch -c conda-forge
```

Collecting package metadata (current_repodata.json): done

Solving environment: done

==> WARNING: A newer version of conda exists. <==

current version: 4.8.3

latest version: 4.10.3

Please update conda by running

```
$ conda update -n base -c defaults conda
```

GeForce RTX 3090 with CUDA capability sm_86 is not compatible with the current PyTorch installation.

CUDA와 PyTorch 버전 충돌

이제 간단히 Ubuntu를 설치하고, 코드를 돌리면.. 되는 줄 알았는데, 골치아픈 작업이 또 남았다고 했다. 바로 PyTorch 위에서 CUDA를 사용하기 위한 환경을 설정하는 것이다. CUDA는 그래픽카드를 학습에 활용할 수 있도록, Tensorflow나 PyTorch 같은 프레임워크에서 학습에 대한 연산을 CPU가 아닌 GPU가 처리하도록 위임하는 드라이버다.

Ubuntu의 고질적인 NVIDIA Driver와의 호환성 문제와, CUDA toolkit & NVIDIA Driver도 심심치 않은 충돌이 일어난다. 많은 설치법 가이드가 올라와 있지만 그 중에서는 오래된 배포판이거나, 각자의 그래픽카드 사양에 맞지 않는 설명도 섞여 있어 삽질하기 딱 좋다. 그래픽카드가 꼬이면 하는 수 없이 부팅 USB를 꺼내 재설치를 반복하니 더 지겨운 작업으로 여겨진다.

아나콘다 설치 후, 가상환경에서 python 3.6으로 세팅

```
conda create -n py36 python=3.6.10 anaconda
```

아나콘다 설치 후, 가상환경 **py36**을 만들어서 현재 기본 세팅인 파이썬 버전 **3.7.7**이 아닌 **3.6.10**으로 세팅

```
conda activate py36
```

가상환경 활성화 후 작업 진행

CUDA와 NVIDIA 드라이버 버전 확인

```
root@cc81ccbc-0fe3-4093-9863-c83fbf6d5c79-instance ~/vol1/tmp$ cat /proc/driver/nvidia/version
NVRM version: NVIDIA UNIX x86_64 Kernel Module 460.27.04 Fri Dec 11 23:35:05 UTC 2020
GCC version: gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04)

root@cc81ccbc-0fe3-4093-9863-c83fbf6d5c79-instance ~/vol1/tmp$ nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Sun_Jul_28_19:07:16_PDT_2019
Cuda compilation tools, release 10.1, V10.1.243
```

NVIDIA : 460.27.04
CUDA : 10.1.243

CUDA와 NVIDIA 드라이버 버전 확인

충돌하는 이유:

RTX 30- 시리즈는 **CUDA 11.0** 이상 버전이어야 하기 때문

NVIDIA : 460.27.04
CUDA : 10.1.243

CUDA 11.1로 설치

Operating System

Linux

Windows

Architecture

x86_64

ppc64le

sbsa

Distribution

CentOS

Debian

Fedora

OpenSUSE

RHEL

SLES

Ubuntu

WSL-Ubuntu

Version

20.04

18.04

16.04

Installer Type

runfile (local)

deb (local)

deb (network)

RTX3090은
CUDA 11.0 이상의 버전을 설치해야 한다.

다음과 같이 설정해서 다운받는다.

RTX30- 시리즈가 CUDA 11.0 이상의 버전을 설치해야 하는 이유 :

<https://medium.com/@dun.chwong/the-simple-guide-deep-learning-with-rtx-3090-cuda-cudnn-tensorflow-keras-pytorch-e88a2a8249bc>

CUDA 다운 : https://developer.nvidia.com/cuda-11.1.0-download-archive?target_os=Linux&target_arch=x86_64&target_distro=Ubuntu&target_version=1804&target_type=debnetwork

CUDA 사용을 위해 PATH 등록

> Base Installer

Installation Instructions:

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-ubuntu1804.pin
$ sudo mv cuda-ubuntu1804.pin /etc/apt/preferences.d/cuda-repository-pin-600
$ sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/7fa2af80.pub
$ sudo add-apt-repository "deb https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/ /"
$ sudo apt-get update
$ sudo apt-get -y install cuda
```

마지막 명령어는 `apt-get -y install cuda-11-1` 로 변경

`vi ~/.bashrc`

가장 아랫줄에 세 줄 추가 후 저장 (아래 링크 참조)

CUDA 11.4 폴더 삭제 명령어 `rm -rf /usr/local/cuda-11.4`

버전 최종 확인

중요 → <https://whiteglass.tistory.com/15>

11.4 폴더 삭제 → <https://eehoeskrap.tistory.com/525>

CUDA 11.1 성공!!!

```
root@cc81ccbc-0fe3-4093-9863-c83fbf6d5c79-instance ~ /vol1 ls usr/local 127 04:53:48
ls: cannot access 'usr/local': No such file or directory
root@cc81ccbc-0fe3-4093-9863-c83fbf6d5c79-instance ~ /vol1 ls /usr/local 2 04:53:51
bin cuda cuda-11 cuda-11.1 cuda-11.4 etc games include lib man sbin share src
root@cc81ccbc-0fe3-4093-9863-c83fbf6d5c79-instance ~ /vol1 /usr/local/cuda-11.4/bin/uninstall_cuda_11.4.pl
zsh: no such file or directory: /usr/local/cuda-11.4/bin/uninstall_cuda_11.4.pl
root@cc81ccbc-0fe3-4093-9863-c83fbf6d5c79-instance ~ /vol1 cd /usr/local/cuda-11.4/bin 127 04:54:46
root@cc81ccbc-0fe3-4093-9863-c83fbf6d5c79-instance ~ /usr/local/cuda-11.4/bin uninstall_cuda_11.4.pl
zsh: command not found: uninstall_cuda_11.4.pl
root@cc81ccbc-0fe3-4093-9863-c83fbf6d5c79-instance ~ /usr/local/cuda-11.4/bin cd ~ 127 04:55:09
root@cc81ccbc-0fe3-4093-9863-c83fbf6d5c79-instance ~ rm -rf /usr/local/cuda-11.4 17 04:56:23
root@cc81ccbc-0fe3-4093-9863-c83fbf6d5c79-instance ~ ls /usr/local 18 04:56:27
bin cuda cuda-11 cuda-11.1 cuda-11.4 etc games include lib man sbin share src
root@cc81ccbc-0fe3-4093-9863-c83fbf6d5c79-instance ~ nvcc -V 19 04:56:30
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2020 NVIDIA Corporation
Built on Mon_Oct_12_20:09:46_PDT_2020
Cuda compilation tools, release 11.1, V11.1.105
Build cuda_11.1.TC455_06.29190527_0
```