

Kakao Arena Melon Playlist Continuation

LunchPlay

이민호, 박정희, 지정재

목차

01

대회 소개 및 DataSet

02

환경 구축

03

BERT

04

CNN

05

Word2Vec

01

대회 소개 및 DataSet

The screenshot shows the Kakao Arena website. At the top, there's a navigation bar with 'kakaoarena', '대회 목록', '데이터셋', '포럼', '공지/문역', and '로그인'. The main banner features a cartoon bear character and text: '카카오 아레나는 머신러닝과 문제 해결을 즐기는 사람들을 위한 장소입니다.' Below this is a yellow button '로그인하고 대회 참가'. To the right, a '공지' (Notice) box contains information about a GitHub repository update and a contest. The main content area lists two competitions: '쇼핑몰 상품 카테고리 분류' (Shopping Mall Product Category Classification) and '브런치 사용자를 위한 글 추천 대회' (Article Recommendation Contest for Brunch Users). Each competition entry includes a logo, description, start date, and a progress bar. At the bottom, a '대회 참여 절차' (Competition Participation Process) diagram shows a sequence of steps: 1. Registration, 2. Data Download, 3. Model Training, 4. Submission, 5. Evaluation, 6. Winner Announcement.

kakaoarena

대회 목록 데이터셋 포럼 공지/문역 로그인

카카오 아레나는 머신러닝과 문제 해결을 즐기는 사람들을 위한 장소입니다.

로그인하고 대회 참가

공지 + 더 보기

[필독] 최종 제출물 재현을 위한 GitHub Repository 수정 기
한(7/27 23:59:59) 안내

[안내] 3회 대회 종료

[안내] 이벤트 설문결과-이번대회 HOT한 알고리즘은?

플레이그라운드 종료된 대회도 플레이그라운드에서 참여 가능합니다.

쇼핑몰 상품 카테고리 분류
쇼핑몰에 등록된 상품의 텍스트, 이미지 정보들을 활용해 카테고리를 분류하는 대회 (플레이그라운드)
10달 전 시작
2019.10.15

참여한 팀 수
59팀

참가하기

브런치 사용자를 위한 글 추천 대회
brunch 데이터를 활용해 사용자의 취향에 맞는 글을 예측하는 대회 (플레이그라운드)
10달 전 시작
2019.10.15

참여한 팀 수
48팀

참가하기

대회 참여 절차

1. 등록 2. 데이터셋 다운로드 3. 모델 학습 4. 결과 제출 5. 평가 6. FIN. 7. 결과 확인 8. 시상

대회 소개 및 DataSet

kaKao arena

대회 목적 | 데이터셋 | 포럼 | 공지/문의 | 로그인

종료 대회

Melon Playlist Continuation

플레이리스트에 있는 곡들과 어울리는 곡들을 찾아주세요

참여 팀 수: 786 | 총 예산: ₩14,080,000

4월 29일 4시 ~ 20일 00분 종료

개요 | 데이터 | 리더보드 | 포럼

다른 대회 선택 ▾

개요

상세 설명 |赛程 | 타임라인 | 상금 | 규칙

플레이리스트에 가장 어울리는 곡들을 예측할 수 있을까?

플레이리스트에 있는 곡들과 비슷한 느낌의 곡들을 계속해서 듣고 싶은 적이 있으셨나요?

이번 대회에서는 플레이리스트에 수록된 곡과 태그의 절반 또는 전부가 숨겨져 있을 때, 주어진지 않은 곡들과 태그를 예측하는 것을 목표로 합니다.

만약 플레이리스트에 들어있는 곡들의 절반을 보여주고, 나머지 숨겨진 절반을 예측할 수 있는 모델을 만든다면, 플레이리스트에 들어 있는 곡이 전부 주어졌을 때 이 모델이 해당 플레이리스트와 어울리는 곡들을 추천해 줄 것이라고 기대할 수 있을 것입니다.

데이터셋 구성

이 대회를 통해 다루게 될 데이터는 아래와 같습니다.

- 플레이리스트 메타데이터
 - 플레이리스트 제목
 - 플레이리스트에 수록된 곡
 - 플레이리스트에 달려있는 태그 목록
 - 플레이리스트 좋아요 수
 - 플레이리스트가 최종 수정된 시각
- 곡 메타데이터
 - 곡 제목
 - 앨범 제목
 - 아티스트명

- KaKao arena_ Melon Playlist Continuation
 - 목적 : 플레이리스트에 수록된 곡과 태그의 절반 또는 전부가 숨겨져 있을 때, 주어진지 않은 곡들과 태그를 예측하는 것
 - 목표 : 주어진 플레이리스트의 정보를 참조하여 해당 플레이리스트에 원래 수록되어 있었을 곡 리스트와 태그 리스트를 예측 (플레이리스트별로 원래 플레이리스트에 들어있었을 것이라 예상되는 곡 100개, 태그 10개를 제출).
 - 기간 : 2020. 4. 27 ~ 2020. 7. 26
 - 최종 제출물에 대하여 tensorflow, pytorch, keras로 모델 로드 및 재현 가능함을 기본 규칙으로 규정

01

대회 소개 및 DataSet

데이터

데이터셋 설명

이번 대회에서 제공하는 정보는 멜론 서비스에서 수집된 플레이리스트의 일부분입니다. 대회 설명에서 언급한 것과 같이, 곡에 대한 Mel-spectrogram 데이터는 용량이 방대합니다. 베이스라인 코드에서와 같이, 해당 데이터를 사용하지 않고도 문제를 해결하는 방법이 있으니, 컴퓨터의 용량과 컴퓨팅 파워가 충분할 때 사용하세요.

- song_meta.json: 곡 메타데이터
 - 총 707,989개의 곡에 대한 메타데이터가 수록되어 있습니다.
 - 개발 데이터와 평가 데이터에 수록된 모든 곡에 대한 메타데이터가 포함되어 있습니다.
 - 필드 설명
 - _id: 곡 ID
 - album_id: 앨범 ID
 - artist_id_basket: 아티스트 ID 리스트
 - artist_name_basket: 아티스트 리스트

데이터 (232.83 GB)

파일 목록

train.json 115.1K x 6
val.json 23.0K x 6
test.json 10.7K x 6
genre_gn_all.json
song_meta.json 708.0K x 9
arena_mel_0.tar
arena_mel_1.tar
arena_mel_2.tar
arena_mel_3.tar
arena_mel_4.tar

파일 상세

왼쪽 파일 목록에서 파일을 선택해주세요.

파일명

크기
0.00 Bytes (0 Bytes)

01

대회 소개 및 DataSet

● DataSet

1. Song_meta.json

- 총 707,989개의 곡에 대한 메타데이터가 수록
- 개발 데이터와 평가 데이터에 수록된 모든 곡에 대한 메타데이터가 포함
- _id: 곡 ID
- album_id: 앨범 ID
- artist_id_basket: 아티스트 ID 리스트
- artist_name_basket: 아티스트 리스트
- song_name: 곡 제목
- song_gn_gnr_basket: 곡 장르 리스트
- song_gn_dtl_gnr_basket: 곡 세부 장르 리스트
- issue_date: 발매일

2. genre_gn_all.json

- 곡 메타데이터에 수록된 장르에 대한 정보
- song_meta.json 에서 song_gn_gnr_basket 과 song_gn_dtl_gnr_basket 에 들어가는 정보들에 대한 메타데이터

3. arena_mel_{0~39}.tar

- 곡에 대한 mel-spectrogram 데이터를 담고있는 파일
- 각 곡 ID마다 npy 파일 1개가 배정
- numpy로 로드 가능.

대회 소개 및 DataSet

● DataSet

4. train.json

- 모델 학습용 파일로, 115,071개 플레이리스트의 원본 데이터가 수록
 - id: 플레이리스트 ID
 - playlist_title: 플레이리스트 제목
 - tags: 태그 리스트
 - songs: 곡 리스트
 - like_cnt: 좋아요 개수
 - updt_date: 수정 날짜

5. test.json

- 파이널 리더보드용 문제 파일로, 10,740개 플레이리스트에 대한 문제가 수록
- 모든 데이터가 수록되어있는 train 파일과는 다르게, 곡과 태그의 일부가 수록
 - id: 플레이리스트 ID
 - playlist_title: 플레이리스트 제목
 - tags: 태그 리스트
 - songs: 곡 리스트
 - like_cnt: 좋아요 개수
 - updt_date: 수정 날짜

환경 구축 및 구조

● 환경 구축

- OS : ubuntu 18.04
- Language ; Python 3.6
- Gcc 7.5
- Cudnn 7.6 (I'm using libcudnn7 cuda7.6 not libcudnn8)
- Cuda 10.1 + cuda10.0의 일부
- ananconda 최신 버전
- OpenMPI 4.0.4 (현재 안정된 버전)
- Horovod(Tensorflow, Keras, Pytorch, MXNet에서의 Multi-GPU를 활용한 Distributed Training을 지원하는 Framework)
- NVIDIA GPU 드라이버 418.113

(참고 자료 : https://docs.nvidia.com/deploy/cuda-compatibility/index.html#binary-compatibility_table-toolkit-driver)

- CUDA Toolkit 10.1
- tensorflow1.15-gpu

환경 구축 및 구조

● GCC, C++ 설치

- Gcc --version 을 터미널에 입력하여 버전이 7.5 인지 확인
- 만약 설치가 안되어 있는 경우 : <https://launchpad.net/~ubuntu-toolchain-r/+archive/ubuntu/test>
(설치가능한 버전을 확인 후 설치)

- 설치 방법

```
sudo add-apt-repository ppa:ubuntu-toolchain-r/test
```

```
sudo apt-get update
```

```
sudo apt-get install gcc-7 g++ make -y
```

02

환경 구축 및 구조

● Anaconda 설치

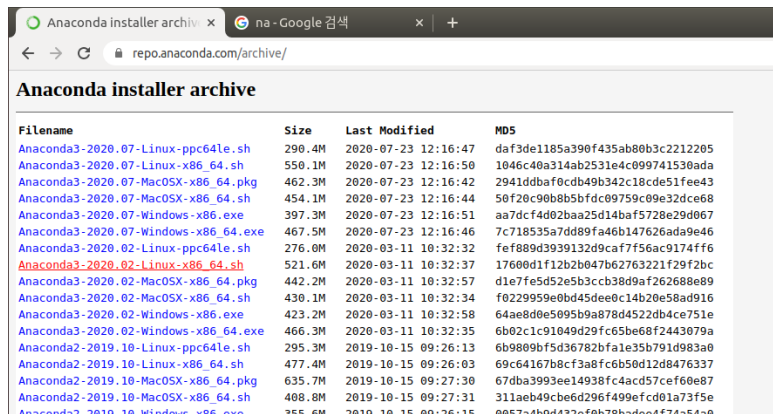
1. <https://repo.anaconda.com/archive/> 에서 anaconda release를 선택해서 다운로드 받은 후 Terminal에서 anaconda sh 파일이 있는 폴더로 이동한 후

chmod +x 파일명

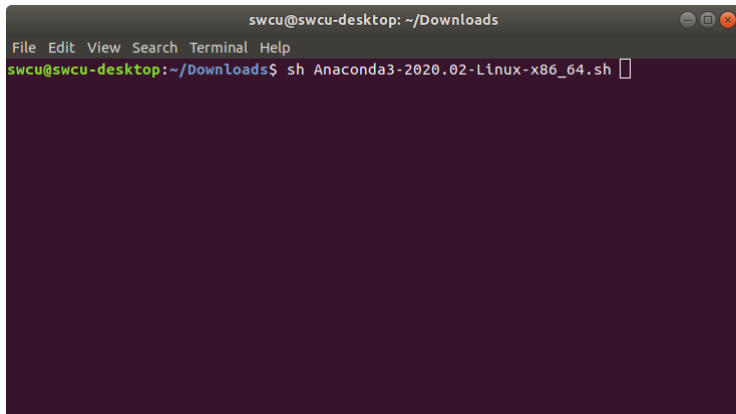
sh Anaconda3-20xx.xx-Linux-x86_64.sh (ex : # sh Anaconda3-2020.02-Linux-x86_64.sh)

(주의 : sudo로 설치를 진행하면 나중에 가상환경 만들때 chown 다시 실행해야 해서 힘들)

2. source ~/.bashrc 입력 (입력해야 환경 변수에 등록된 게 실행되서 conda 명령어 사용가능)



Filename	Size	Last Modified	MD5
Anaconda3-2020.07-Linux-ppc64le.sh	290.4M	2020-07-23 12:16:47	daf3de1185a390f435ab80b3c2212205
Anaconda3-2020.07-Linux-x86_64.sh	550.1M	2020-07-23 12:16:50	1046c40a314ab2531e4c099741530ada
Anaconda3-2020.07-MacOSX-x86_64.pkg	462.3M	2020-07-23 12:16:42	2941ddbaf0c0db49b342c18cde51fee43
Anaconda3-2020.07-MacOSX-x86_64.sh	454.1M	2020-07-23 12:16:44	50f20c908b8b5b5fde09759c09e32dce68
Anaconda3-2020.07-Windows-x86.exe	397.3M	2020-07-23 12:16:51	aa7dcf4d02baa25d14baf5728e29d067
Anaconda3-2020.07-Windows-x86_64.exe	467.5M	2020-07-23 12:16:46	7c718535a7dd89fa46b147626ada9e46
Anaconda3-2020.02-Linux-ppc64le.sh	276.0M	2020-03-11 10:32:32	fef889d393132d9caf7f56ac9174ff6
Anaconda3-2020.02-Linux-x86_64.sh	521.6M	2020-03-11 10:32:37	17600d1f12b2b047b62763221f29f2bc
Anaconda3-2020.02-MacOSX-x86_64.pkg	442.2M	2020-03-11 10:32:57	d1e7fe5d52e5b3ccb38d9af262688e89
Anaconda3-2020.02-MacOSX-x86_64.sh	430.1M	2020-03-11 10:32:34	f0229959e0bd45dee0c14b20e58ad916
Anaconda3-2020.02-Windows-x86.exe	423.2M	2020-03-11 10:32:58	64ae8d0e5095b9a878d4522db4ce751e
Anaconda3-2020.02-Windows-x86_64.exe	466.3M	2020-03-11 10:32:35	6b02c1c91049d29fc65be68f72443079a
Anaconda2-2019.10-Linux-ppc64le.sh	295.3M	2019-10-15 09:26:13	6b9809bf5d36782bfa1e35b791d983a0
Anaconda2-2019.10-Linux-x86_64.sh	477.4M	2019-10-15 09:26:03	69c64167b8cf3a8fcb50d12d8476337
Anaconda2-2019.10-MacOSX-x86_64.pkg	635.7M	2019-10-15 09:27:30	67dba3993ee14938fca4acd57cef60e87
Anaconda2-2019.10-MacOSX-x86_64.sh	400.8M	2019-10-15 09:27:31	311aeb49cbe6d296f499efcd01a73f5e
Anaconda2-2019.10-Windows-x86.exe	355.6M	2019-10-15 09:26:15	9057c4b0d432aef9b70b3dca4f74a54c0



```
swcu@swcu-desktop: ~/Downloads
File Edit View Search Terminal Help
swcu@swcu-desktop:~/Downloads$ sh Anaconda3-2020.02-Linux-x86_64.sh
```

02

환경 구축 및 구조

● Anaconda 설치

- 설명 글이 길게 나와서 넘기고 싶으면 q를 입력

```
swcu@swcu-desktop: ~/Downloads
File Edit View Search Terminal Help
Please, press ENTER to continue
>>>
=====
End User License Agreement - Anaconda Individual Edition
=====

Copyright 2015-2020, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between you and Anaconda, Inc. ("Anaconda") and governs your use of Anaconda Individual Edition (which was formerly known as Anaconda Distribution).

Subject to the terms of this Agreement, Anaconda hereby grants you a non-exclusive, non-transferable license to:

    * Install and use the Anaconda Individual Edition (which was formerly known as Anaconda Distribution),
    * Modify and create derivative works of sample source code delivered in Anaconda Individual Edition; and
    * Redistribute code files in source (if provided to you by Anaconda as source) and binary forms, with or without modification subject to the requirements set forth below.
```

```
swcu@swcu-desktop: ~/Downloads
File Edit View Search Terminal Help
Please, press ENTER to continue
>>>
=====
End User License Agreement - Anaconda Individual Edition
=====

Copyright 2015-2020, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between you and Anaconda, Inc. ("Anaconda") and governs your use of Anaconda Individual Edition (which was formerly known as Anaconda Distribution).

Subject to the terms of this Agreement, Anaconda hereby grants you a non-exclusive, non-transferable license to:

    * Install and use the Anaconda Individual Edition (which was formerly known as Anaconda Distribution),
    * Modify and create derivative works of sample source code delivered in Anaconda Individual Edition; and
    * Redistribute code files in source (if provided to you by Anaconda as source) and binary forms, with or without modification subject to the requirements set forth below.
```

02

환경 구축 및 구조

● Anaconda 설치

- license -> yes

```
swcu@swcu-desktop: ~/Downloads
File Edit View Search Terminal Help
pycryptodome
  A fork of PyCrypto. It is a self-contained Python package of low-level cryptographic primitives.
pycryptodomex
  A stand-alone version of pycryptodome.
libsodium
  A software library for encryption, decryption, signatures, password hashing and more.
pynacl
  A Python binding to the Networking and Cryptography library, a crypto library with the stated goal of improving usability, security and speed.
Last updated February 25, 2020

Do you accept the license terms? [yes|no]
[no] >>>
Please answer 'yes' or 'no':
>>>
Please answer 'yes' or 'no':
>>> 
```

- 환경 변수 설정 기본 설정으로 enter

```
swcu@swcu-desktop: ~/Downloads
File Edit View Search Terminal Help
and more.
pynacl
  A Python binding to the Networking and Cryptography library, a crypto library with the stated goal of improving usability, security and speed.
Last updated February 25, 2020

Do you accept the license terms? [yes|no]
[no] >>>
Please answer 'yes' or 'no':
>>>
Please answer 'yes' or 'no':
>>> yes

Anaconda3 will now be installed into this location:
/home/swcu/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/swcu/anaconda3] >>> 
```

02

환경 구축 및 구조

● Anaconda 설치

- Python 3.6
- # conda create -n [가상환경이름] python=3.6

- 가상환경 실행 # conda activate [가상환경이름]

```
swcu@swcu-desktop: ~  
File Edit View Search Terminal Help  
(base) swcu@swcu-desktop:~$ conda create -n KaKao python=3.6
```

```
swcu@swcu-desktop: ~  
File Edit View Search Terminal Help  
python-3.6.10      | 29.7 MB | ##### | 100%  
xz-5.2.5          | 341 KB  | ##### | 100%  
sqlite-3.32.3     | 1.1 MB  | ##### | 100%  
ncurses-6.2       | 817 KB  | ##### | 100%  
libedit-3.1.20191231 | 116 KB  | ##### | 100%  
ca-certificates-2020 | 125 KB  | ##### | 100%  
wheel-0.34.2      | 51 KB   | ##### | 100%  
openssl-1.1.1g    | 2.5 MB  | ##### | 100%  
readline-8.0      | 356 KB  | ##### | 100%  
tk-8.6.10         | 3.0 MB  | ##### | 100%  
Preparing transaction: done  
Verifying transaction: done  
Executing transaction: done  
#  
# To activate this environment, use  
#  
#   $ conda activate KaKao  
#  
# To deactivate an active environment, use  
#  
#   $ conda deactivate  
  
(base) swcu@swcu-desktop:~$ conda activate KaKao  
(KaKao) swcu@swcu-desktop:~$
```

환경 구축 및 구조

● tensorflow 및 관련 라이브러리 설치

- *tensorflow 공식 홈페이지 따라서 설치하시면 버전 다 꼬임
- Nvidia Driver의 경우 우분투 설치시 third-party driver 설치 옵션 체크후 os 설치하면 자동으로 설치
(이 방법으로 설치 안되어 있으면 넘기기)
- `nvidia-smi` #driver 상태 확인
- 하지만 driver와 라이브러리의 버전을 맞춰주려면 driver를 삭제해야 됩니다.
- `sudo apt-get remove --purge "*nvidia*"`
- `sudo apt-get autoremove`

02

환경 구축 및 구조

- tensorflow 및 관련 라이브러리 설치

```
swcu@swcu-desktop: ~/Downloads
File Edit View Search Terminal Help
(bert) swcu@swcu-desktop:~/Downloads$ sudo apt-get remove --purge "*nvidia*" -y
```

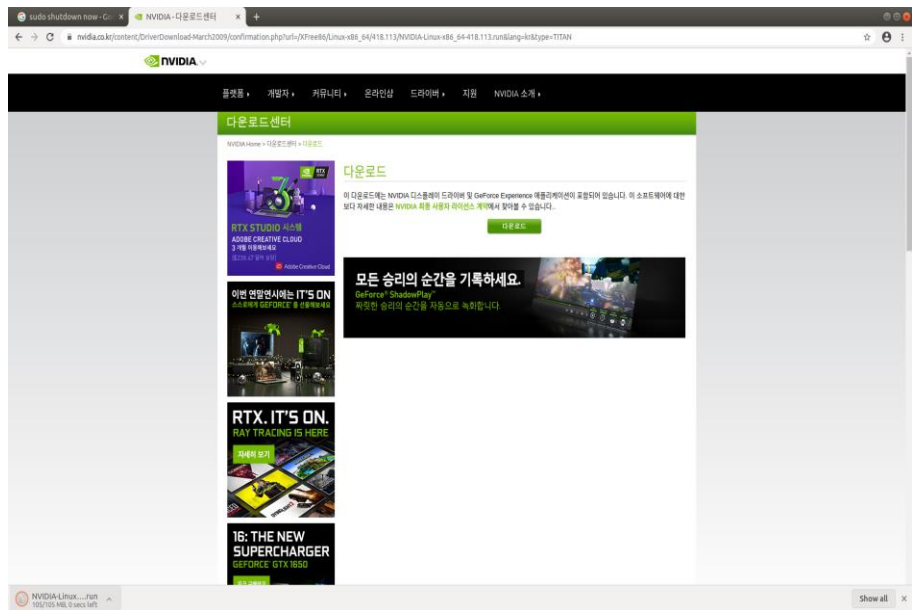
```
swcu@swcu-desktop: ~/Downloads
File Edit View Search Terminal Help
Package 'nvidia-dkms-440-server' is not installed, so not removed
Package 'nvidia-driver-418-server' is not installed, so not removed
Package 'nvidia-driver-440-server' is not installed, so not removed
Package 'nvidia-headless-418-server' is not installed, so not removed
Package 'nvidia-headless-440-server' is not installed, so not removed
Package 'nvidia-headless-no-dkms-418-server' is not installed, so not removed
Package 'nvidia-headless-no-dkms-440-server' is not installed, so not removed
Package 'nvidia-kernel-common-418-server' is not installed, so not removed
Package 'nvidia-kernel-common-440-server' is not installed, so not removed
Package 'nvidia-kernel-source-418-server' is not installed, so not removed
Package 'nvidia-kernel-source-440-server' is not installed, so not removed
Package 'nvidia-libopencl-331' is not installed, so not removed
Package 'nvidia-libopencl-331-updates' is not installed, so not removed
Package 'nvidia-libopencl-340' is not installed, so not removed
Package 'nvidia-libopencl-340-updates' is not installed, so not removed
Package 'nvidia-opencl-icd-331' is not installed, so not removed
Package 'nvidia-opencl-icd-331-updates' is not installed, so not removed
Package 'nvidia-opencl-icd-340-updates' is not installed, so not removed
Package 'nvidia-utils-418-server' is not installed, so not removed
Package 'nvidia-utils-440-server' is not installed, so not removed
Package 'xserver-xorg-video-nvidia-418-server' is not installed, so not removed
Package 'xserver-xorg-video-nvidia-440-server' is not installed, so not removed
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
(bert) swcu@swcu-desktop:~/Downloads$
```

만약 없다면 그림처럼 삭제할게 없다고 표시된다.

02

환경 구축 및 구조

- tensorflow 및 관련 라이브러리 설치



- 해당 경로에서 nvidia driver 418.113 버전 다운로드

- Downloads 경로에서 chmod +x 설치된 드라이버 설치파일

링크 : <https://www.nvidia.co.kr/download/driverResults.aspx/153774/kr>

환경 구축 및 구조

● tensorflow 및 관련 라이브러리 설치

- 모든 명령어는 관리자 권한 (sudo 나 sudo su)
- Nvidia driver를 설치하기 위해 기본 설치되어 동작하고 있던 nouveau kernel driver를 중지.
- 우선 /etc/modprobe.d/blacklist.conf 로 들어가서 끝 줄에 blacklist nouveau를 입력 후 저장 종료 후 다음 명령어 입력

```
echo options nouveau modeset=0 | sudo tee -a /etc/modprobe.d/nouveau-kms.conf
update-initramfs -u
reboot
```
- Reboot 된 후 Ctrl + alt + f3 로 gui가 아닌 화면으로 전환
- 현재 사용되고 있는 gui를 꺼주기 위해 systemctl isolate multi-user.target으로 다른 남아있는 모든 gui 화면을 끄.
- 다른 화면으로 전환 되었다면 Ctrl + alt + f3로 화면을 전환을 다시 입력해서 방금 명령어를 수행했던 화면으로 돌아오자
- 이 때 미리 설치해둔 드라이버 설치 파일 실행 후 reboot

02

환경 구축 및 구조

- tensorflow 및 관련 라이브러리 설치

```
root@swcu-desktop:/home/swcu/Downloads# nano /etc/modprobe.d/blacklist.conf
```

```
# replaced by p54pci
blacklist prism54

# replaced by b43 and ssb.
blacklist bcm43xx

# most apps now use garmin usb driver directly (Ubuntu: #114565)
blacklist garmin_gps

# replaced by asus-laptop (Ubuntu: #184721)
blacklist asus_acpi

# low-quality, just noise when being used for sound playback, causes
# hangs at desktop session start (Ubuntu: #246969)
blacklist snd_pcsp

# ugly and loud noise, getting on everyone's nerves; this should be done by a
# nice pulseaudio bing (Ubuntu: #77010)
blacklist pcspkr

# EDAC driver for amd76x clashes with the agp driver preventing the aperture
# from being initialised (Ubuntu: #297750). Blacklist so that the driver
# continues to build and is installable for the few cases where its
# really needed.
blacklist amd76x_edac

blacklist nouveau
```

02

환경 구축 및 구조

- tensorflow 및 관련 라이브러리 설치

```
root@swcu-desktop:/home/swcu/Downloads# echo options nouveau modeset=0 | sudo tee -a /etc/modprobe.d/nouveau-kms.conf
options nouveau modeset=0
root@swcu-desktop:/home/swcu/Downloads#
```

```
root@swcu-desktop:/home/swcu/Downloads# update-initramfs -u_
```

Reboot를 한 후 다음 명령어를 입력 root가 아니면 sudo 를 앞에 붙이기

```
(base) swcu@swcu-desktop:~$ sudo su
[sudo] password for swcu:
root@swcu-desktop:/home/swcu# systemctl isolate multi-user.target
```

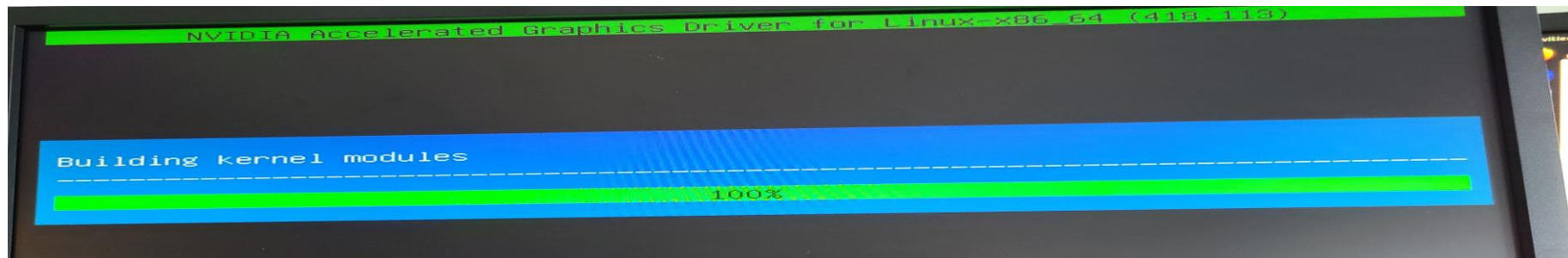
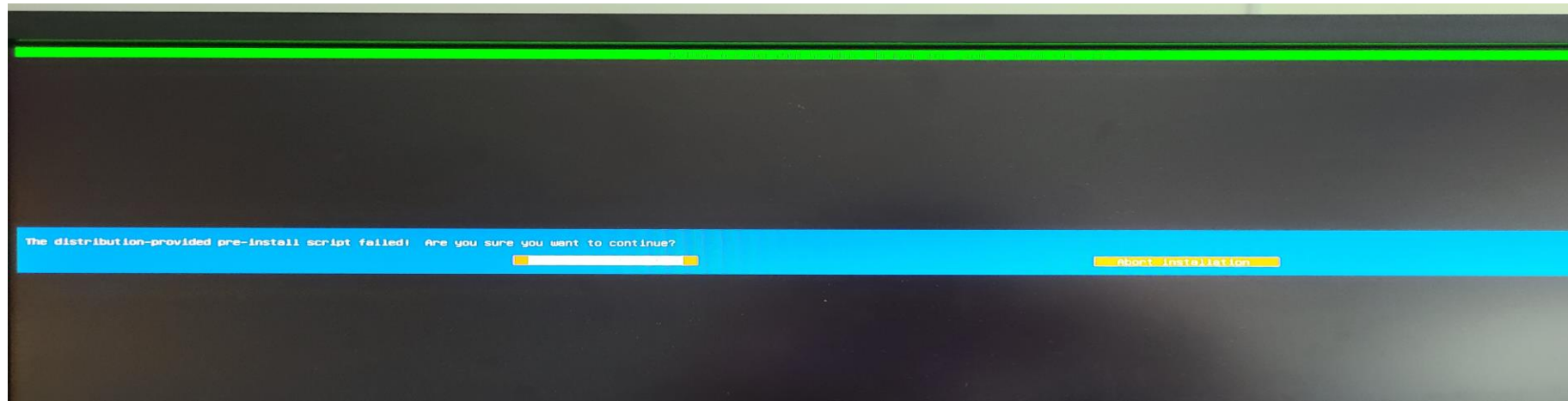
다른 화면으로 전환되었다면 ctrl + alt + f3

```
root@swcu-desktop:/home/swcu# cd Downloads/
root@swcu-desktop:/home/swcu/Downloads# ls
Anaconda3-2020.07-Linux-x86_64.sh  NVIDIA-Linux-x86_64-418.113.run
root@swcu-desktop:/home/swcu/Downloads# ./NVIDIA-Linux-x86_64-418.113.run _
```

02

환경 구축 및 구조

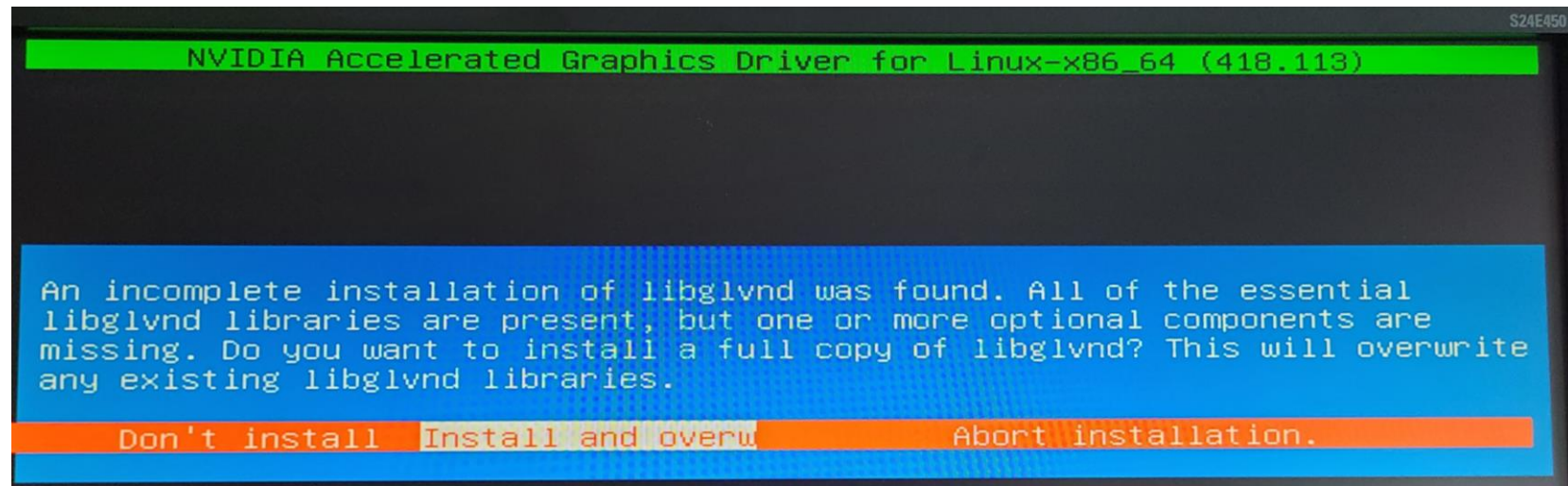
- tensorflow 및 관련 라이브러리 설치



02

환경 구축 및 구조

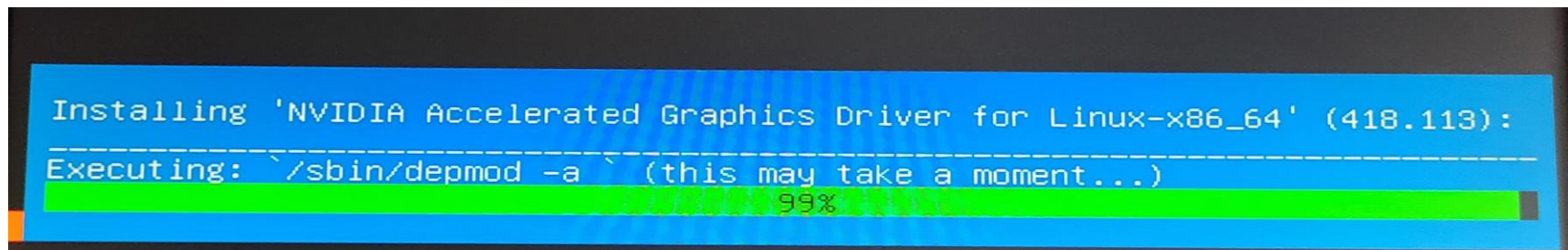
- tensorflow 및 관련 라이브러리 설치



```
S24E450
NVIDIA Accelerated Graphics Driver for Linux-x86_64 (418.113)

An incomplete installation of libglvnd was found. All of the essential
libglvnd libraries are present, but one or more optional components are
missing. Do you want to install a full copy of libglvnd? This will overwrite
any existing libglvnd libraries.

Don't install  Install and overw  Abort installation.
```



```
Installing 'NVIDIA Accelerated Graphics Driver for Linux-x86_64' (418.113):
-----
Executing: ` /sbin/depmod -a ` (this may take a moment...)
99%
```


02

환경 구축 및 구조

- tensorflow 및 관련 라이브러리 설치

Would you like to run the nvidia-xconfig utility to automatically update your X configuration file so that the NVIDIA X driver will be used when you restart X? Any pre-existing X configuration file will be backed up.

Yes

No

Your X configuration file has been successfully updated. Installation of the NVIDIA Accelerated Graphics Driver for Linux-x86_64 (version: 418.113) is now complete.

OK

root@swcu-desktop:/home/swcu/Downloads# reboot

환경 구축 및 구조

● CUDA 설치

- CUDA 10.1 설치 -
 - 링크 : <https://developer.nvidia.com/cuda-10.1-download-archive-base>
 - 위 사이트 접속하여 Linux_X86_64_Ubuntu_18.04_runfile (local) 선택 후 설치 파일 다운로드!
 - 설치한 파일을 'chmod +x' 명령어로 파일 접근 권한을 변경한 뒤에 './실행파일명' 로 실행
- CUDA 10.0 설치
 - 링크 : <https://developer.nvidia.com/cuda-10.0-download-archive>
 - 위 사이트 접속하여 Linux_X86_64_Ubuntu_18.04_runfile (local) 선택 후 설치 파일 다운로드!
 - 설치하는 파일을 'chmod +x' 명령어로 파일 접근 권한을 변경한 뒤에 './실행파일명' 로 실행

Q : Cuda 10.1 을 설치했는데 왜 설치하는가?

A : Cuda10.1의 라이브러리 외에도 cuda10.0 라이브러리를 필요로 하는 경우가 있었기 때문

02

환경 구축 및 구조

● CUDA 설치

- */usr/local/ 디렉토리에 cuda-10.1 과 cuda-10.0 이 설치되어 있으며 cuda-10.1을 주로 사용하겠다고 환경변수에 등록해놓았으므로 cuda-10.0 /lib64에 있는 라이브러리 파일들을 복사해서 cuda-10.1/lib64 붙여넣어줍니다.*

```
sudo cp -r /usr/local/cuda-10.0 /lib64/* /usr/local/cuda-10.1/lib64/
```

1-4) **~/.profile 파일 열기** (터미널 창에 입력)

```
sudo gedit ~/.profile
```

1-5) 아래의 내용을 **~/.profile 파일에 추가하고 저장하기**
(* cuda version이 다르면, 10.1을 설치할 버전에 맞게 수정하면 됨)

```
export PATH=/usr/local/cuda-10.1/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda-10.1/lib64:$LD_LIBRARY_PATH
```

1-6) **변경 사항 적용하기 & 재부팅** (터미널 창에 입력)

```
source ~/.profile
sudo reboot
```


02

환경 구축 및 구조

● CUDA 설치

- */usr/local/ 디렉토리에 cuda-10.1 과 cuda-10.0 이 설치되어 있으며 cuda-10.1을 주로 사용하겠다고 환경변수에 등록해놓았으므로 cuda-10.0 /lib64에 있는 라이브러리 파일들을 복사해서 cuda-10.1/lib64 붙여넣어줍니다.*

```
sudo cp -r /usr/local/cuda-10.0 /lib64/* /usr/local/cuda-10.1/lib64/
```

1-4) **~/.profile 파일 열기** (터미널 창에 입력)

```
sudo gedit ~/.profile
```

1-5) 아래의 내용을 **~/.profile 파일에 추가하고 저장하기**
(* cuda version이 다르면, 10.1을 설치할 버전에 맞게 수정하면 됨)

```
export PATH=/usr/local/cuda-10.1/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda-10.1/lib64:$LD_LIBRARY_PATH
```

1-6) **변경 사항 적용하기 & 재부팅** (터미널 창에 입력)

```
source ~/.profile
sudo reboot
```

02

환경 구축 및 구조

● CUDA 설치

- */usr/local/ 디렉토리에 cuda-10.1 과 cuda-10.0 이 설치되어 있으며 cuda-10.1을 주로 사용하겠다고 환경변수에 등록해놓았으므로 cuda-10.0 /lib64에 있는 라이브러리 파일들을 복사해서 cuda-10.1/lib64 붙여넣어줍니다.*

```
sudo cp -r /usr/local/cuda-10.0 /lib64/* /usr/local/cuda-10.1/lib64/
```

1-4) **~/.profile 파일 열기** (터미널 창에 입력)

```
sudo gedit ~/.profile
```

1-5) 아래의 내용을 **~/.profile 파일에 추가하고 저장하기**
(* cuda version이 다르면, 10.1을 설치할 버전에 맞게 수정하면 됨)

```
export PATH=/usr/local/cuda-10.1/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda-10.1/lib64:$LD_LIBRARY_PATH
```

1-6) **변경 사항 적용하기 & 재부팅** (터미널 창에 입력)

```
source ~/.profile
sudo reboot
```

02

환경 구축 및 구조

● CUDA 설치

The screenshot shows the NVIDIA Developer website for downloading the CUDA Toolkit 10.1. The browser window title is "CUDA Toolkit 10.1 original Archive | NVIDIA Developer - Mozilla Firefox". The address bar shows the URL: https://developer.nvidia.com/cuda-10.1-download-archive-base?target_os=Linux&target_arch=x86_64.

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System: Windows, Linux, Mac OSX

Architecture: x86_64, ppc64le

Distribution: Fedora, OpenSUSE, RHEL, CentOS, SLES, Ubuntu

Version: 18.10, 18.04, 16.04, 14.04

Installer Type: runfile (local), deb (local), deb (network)

Download Installer for Linux Ubuntu 18.10 x86_64

The base installer is available for download below.

> Base Installer

Download [2.4 GB]

Installation Instructions:

1. Run `sudo sh cuda_10.1.105_418.39_linux.run`
2. Follow the command-line prompts

The CUDA Toolkit contains Open-Source Software. The source code can be found [here](#).
The checksums for the installer and patches can be found in [Installer Checksums](#).
For further information, see the [Installation Guide for Linux](#) and the [CUDA Quick Start Guide](#).

[Documentation >](#) [Release Notes >](#) [Code Samples >](#) [Legacy Releases >](#)

https://developer.nvidia.com/compute/cuda/10.1/Prod/local_installers/cuda_10.1.105_418.39_linux.run

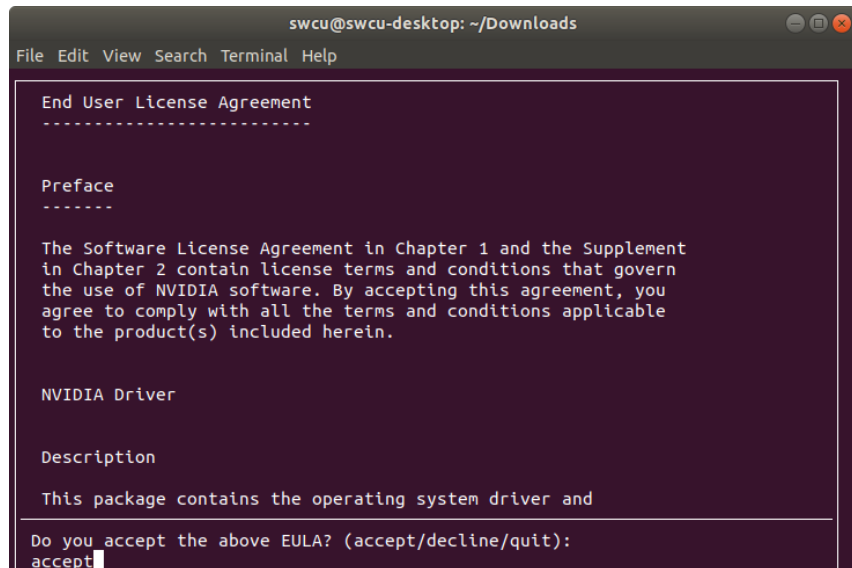
02

환경 구축 및 구조

● CUDA 설치

```
cuda_10.1.105_418.55_linux.run      cuda_10.1.243_418.87.00_linux.run
(base) swcu@swcu-desktop:~/Downloads$ chmod +x cuda_10.1.243_418.87.00_linux.run

(base) swcu@swcu-desktop:~/Downloads$ chmod +x cuda_10.0.130_410.48_linux.run
(base) swcu@swcu-desktop:~/Downloads$ ./cuda_10.1.243_418.87.00_linux.run
```



```
swcu@swcu-desktop: ~/Downloads
File Edit View Search Terminal Help

End User License Agreement
-----

Preface
-----

The Software License Agreement in Chapter 1 and the Supplement
in Chapter 2 contain license terms and conditions that govern
the use of NVIDIA software. By accepting this agreement, you
agree to comply with all the terms and conditions applicable
to the product(s) included herein.

NVIDIA Driver

Description

This package contains the operating system driver and

Do you accept the above EULA? (accept/decline/quit):
accept
```

02

환경 구축 및 구조

● CUDA 설치

```
swcu@swcu-des
File Edit View Search Terminal Help

CUDA Installer
- [X] Driver
  [X] 418.87.00
+ [X] CUDA Toolkit 10.1
  [X] CUDA Samples 10.1
  [X] CUDA Demo Suite 10.1
  [X] CUDA Documentation 10.1
Options
Install
```



```
swcu@swcu-des
File Edit View Search Terminal Help

CUDA Installer
- [ ] Driver
  [ ] 418.87.00
+ [X] CUDA Toolkit 10.1
  [ ] CUDA Samples 10.1
  [ ] CUDA Demo Suite 10.1
  [ ] CUDA Documentation 10.1
Options
Install
```

cuda 10.1 설치시 다음과 같이 뭘 설치할지 결정하는 것이 나오는데 하나 빼고 나머진 그렇게 필요한 것은 아니기 때문에 체크 해제 후 install

02

환경 구축 및 구조

● CUDA 설치

cuda 10.0 설치시 accept no

```
swcu@swcu-desktop: ~/Downloads
File Edit View Search Terminal Help
Preface
-----
The Software License Agreement in Chapter 1 and the Supplement
in Chapter 2 contain license terms and conditions that govern
the use of NVIDIA software. By accepting this agreement, you
agree to comply with all the terms and conditions applicable
to the product(s) included herein.

NVIDIA Driver

Description
This package contains the operating system driver and
fundamental system software components for NVIDIA GPUs.

Do you accept the previously read EULA?
accept/decline/quit: accept

Install NVIDIA Accelerated Graphics Driver for Linux-x86_64 410.48?
(y)es/(n)o/(q)uit: no
```

- 환경변수 설정
설정후 source ~/.profile을
해주어야 바로 해당 환경변수 적용됨

```
swcu@swcu-desktop: ~/Downloads
File Edit View Search Terminal Help
GNU nano 2.9.3 /home/swcu/.profile Modified

PATH="$HOME/bin:$PATH"
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/.local/bin" ] ; then
    PATH="$HOME/.local/bin:$PATH"
fi

export PATH=/usr/local/cuda/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

환경 구축 및 구조

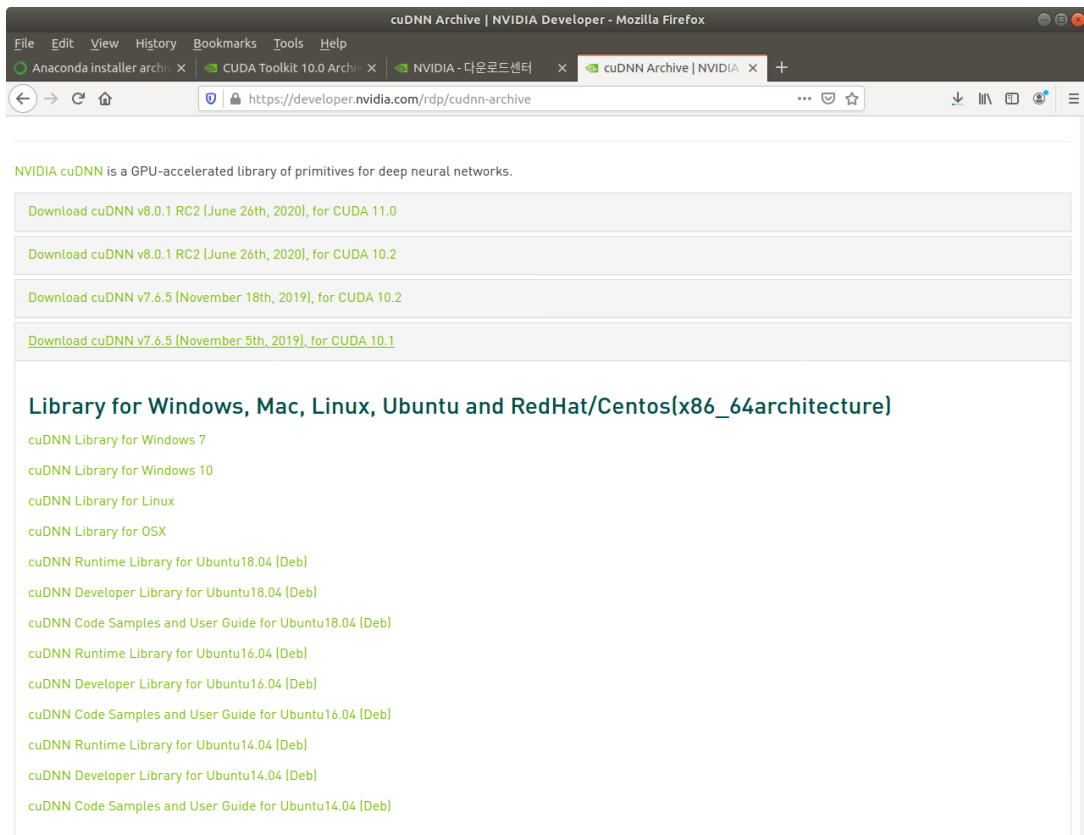
- CUDA 설치 확인

- Terminal 에서
- # nvcc -version
- 으로 CUDA version 확인하기

02

환경 구축 및 구조

- cuDNN



02

환경 구축 및 구조

- cuDNN

```
swcu@swcu-desktop: ~/Downloads
File Edit View Search Terminal Help
(base) swcu@swcu-desktop:~/Downloads$ ls
Anaconda3-2020.02-Linux-x86_64.sh  libcudnn7_7.6.5.32-1+cuda10.1_amd64.deb
cuda_10.0.130_418.48_linux.run      libcudnn7-dev_7.6.5.32-1+cuda10.1_amd64.deb
cuda_10.1.105_418.39_linux.run      NVIDIA-Linux-x86_64-418.113.run
cuda_10.1.243_418.87.00_linux.run   unloading.txt
(base) swcu@swcu-desktop:~/Downloads$ dpkg -i libcudnn7_7.6.5.32-1+cuda10.1_amd64.deb
dpkg: error: requested operation requires superuser privilege
(base) swcu@swcu-desktop:~/Downloads$ sudo dpkg -i libcudnn7_7.6.5.32-1+cuda10.1_amd64.deb
Selecting previously unselected package libcudnn7.
(Reading database ... 132513 files and directories currently installed.)
Preparing to unpack libcudnn7_7.6.5.32-1+cuda10.1_amd64.deb ...
Unpacking libcudnn7 (7.6.5.32-1+cuda10.1) ...
```

```
swcu@swcu-desktop: ~/Downloads
File Edit View Search Terminal Help
(base) swcu@swcu-desktop:~/Downloads$ sudo dpkg -i libcudnn7
libcudnn7_7.6.5.32-1+cuda10.1_amd64.deb
libcudnn7-dev_7.6.5.32-1+cuda10.1_amd64.deb
(base) swcu@swcu-desktop:~/Downloads$ sudo dpkg -i libcudnn7
libcudnn7_7.6.5.32-1+cuda10.1_amd64.deb
libcudnn7-dev_7.6.5.32-1+cuda10.1_amd64.deb
(base) swcu@swcu-desktop:~/Downloads$ sudo dpkg -i libcudnn7-dev_7.6.5.32-1+cuda10.1_amd64.deb
```

환경 구축 및 구조

● OPENMPI



Open MPI: Version 4.0

[About](#)
[Presentations](#)
[Open MPI Team](#)
[FAQ](#)
[Videos](#)
[Performance](#)
[Open MPI Software](#)
[Download](#)
[Current](#)
[Version 4.0](#)
[Upcoming](#)
[Version 4.1](#)
[Still supported](#)
[Version 3.1](#)
[Version 3.0](#)
[Older versions](#)
[Version 2.1 \(retired\)](#)
[Version 2.0 \(retired\)](#)
[Version 1.10 \(ancient\)](#)
[Version 1.8 \(ancient\)](#)
[Version 1.7 \(ancient\)](#)
[Version 1.6 \(ancient\)](#)
[Version 1.5 \(ancient\)](#)
[Version 1.4 \(ancient\)](#)
[Version 1.3 \(ancient\)](#)
[Version 1.2 \(ancient\)](#)
[Version 1.1 \(ancient\)](#)
[Version 1.0 \(ancient\)](#)
[Nightly snapshots](#)
[Documentation](#)

[Home](#)

Enter your e-mail address to be notified when new releases of Open MPI are released



Your email address will be used only to send you announcements about new releases of Open MPI and you will be able to un-subscribe at any time.

Changes in this release:

- [See this page](#) if you are upgrading from a prior major release series of Open MPI. It shows the Big Changes for which end users need to be aware.
- [See the NEWS file](#) for a more fine-grained listing of changes between each release and sub-release of the Open MPI v4.0 series.

See the [version timeline](#) for information on the chronology of Open MPI releases.

Current stable release downloads:

Release	File names	Size	Date	Checksums (GNU md5sum and sha1sum v5.2.1)
4.0.4 SRPM notes	openmpi-4.0.4-1.src.rpm	15.99 MiB	Jun 10, 2020	MD5: cb93295af2b74402fbd2dd91b9ffb7b SHA1: 7d974dfa287d3eca3d6b16f0e8a6231cd2950b14 SHA256: 9cd0b7d951b3643c01828441765d9a8ebc1c01398be515ce8e28df1be4418054
	openmpi-4.0.4.tar.bz2	9.3 MiB	Jun 10, 2020	MD5: 4b7e3efd337a614f0e661fc6faa4a1d7 SHA1: 50861c22a4b92ca2e069cd49d756dd96c659bfa8 SHA256: 47e24eb2223fe5d24438658958a313b6b7a55bb281563542e1afc9dec4a31ac4
	openmpi-4.0.4.tar.gz	16.36 MiB	Jun 10, 2020	MD5: 48890b54bd0b025918e84f0e47b583b0 SHA1: e3145347861243d7b0e3876cd196a067609c60be SHA256: dca264f420411f540a496bdd131bffd83e325fc9006286b39dd19b62d7368233

02

환경 구축 및 구조

- OPENMPI

```
swcu@swcu-desktop: ~/Downloads/openmpi-4.0.4
File Edit View Search Terminal Help
(base) swcu@swcu-desktop:~/Downloads$ tar -zxvf openmpi-4.0.4.tar.gz
openmpi-4.0.4/
openmpi-4.0.4/config/
openmpi-4.0.4/config/c_get_alignment.m4
openmpi-4.0.4/config/c_weak_symbols.m4
openmpi-4.0.4/config/libtool.m4
openmpi-4.0.4/config/ltoptions.m4
openmpi-4.0.4/config/ltsugar.m4
openmpi-4.0.4/config/ltversion.m4
openmpi-4.0.4/config/lt~obsolete.m4
openmpi-4.0.4/config/mpi_check_libfca.m4
openmpi-4.0.4/config/mpi_check_libhcoll.m4
openmpi-4.0.4/config/mpi_check_lustre.m4
openmpi-4.0.4/config/mpi_check_mxm.m4
openmpi-4.0.4/config/mpi_check_psm.m4
openmpi-4.0.4/config/mpi_check_psm2.m4
openmpi-4.0.4/config/mpi_check_pvfs2.m4
openmpi-4.0.4/config/mpi_check_ucx.m4
openmpi-4.0.4/config/mpi_config_files.m4
openmpi-4.0.4/config/mpi_configure_options.m4
openmpi-4.0.4/config/mpi_contrib.m4
openmpi-4.0.4/config/mpi_cxx_find_exception_flags.m4
openmpi-4.0.4/config/mpi_cxx_find_template_parameters.m4
openmpi-4.0.4/config/mpi_cxx_find_template_repository.m4
```

02

환경 구축 및 구조

- OPENMPI

```
swcu@swcu-desktop: ~/Downloads/openmpi-4.0.4
File Edit View Search Terminal Help
(base) swcu@swcu-desktop:~/Downloads/openmpi-4.0.4$ sudo ./configure --prefix=/usr/local
checking for perl... perl

=====
== Configuring Open MPI
=====

*** Startup tests
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /bin/grep
```

02

환경 구축 및 구조

- OPENMPI

```
swcu@swcu-desktop: ~/Downloads/openmpi-4.0.4
File Edit View Search Terminal Help
(base) swcu@swcu-desktop:~/Downloads/openmpi-4.0.4$ ./configure --prefix=/usr/local
```

```
swcu@swcu-desktop: ~/Downloads/openmpi-4.0.4
File Edit View Search Terminal Help
Portals4: no
Shared memory/copy in+copy out: yes
Shared memory/Linux CMA: yes
Shared memory/Linux KNEM: no
Shared memory/XPMEM: no
TCP: yes

Resource Managers
-----
Cray Alps: no
Grid Engine: no
LSF: no
Moab: no
Slurm: yes
ssh/rsh: yes
Torque: no

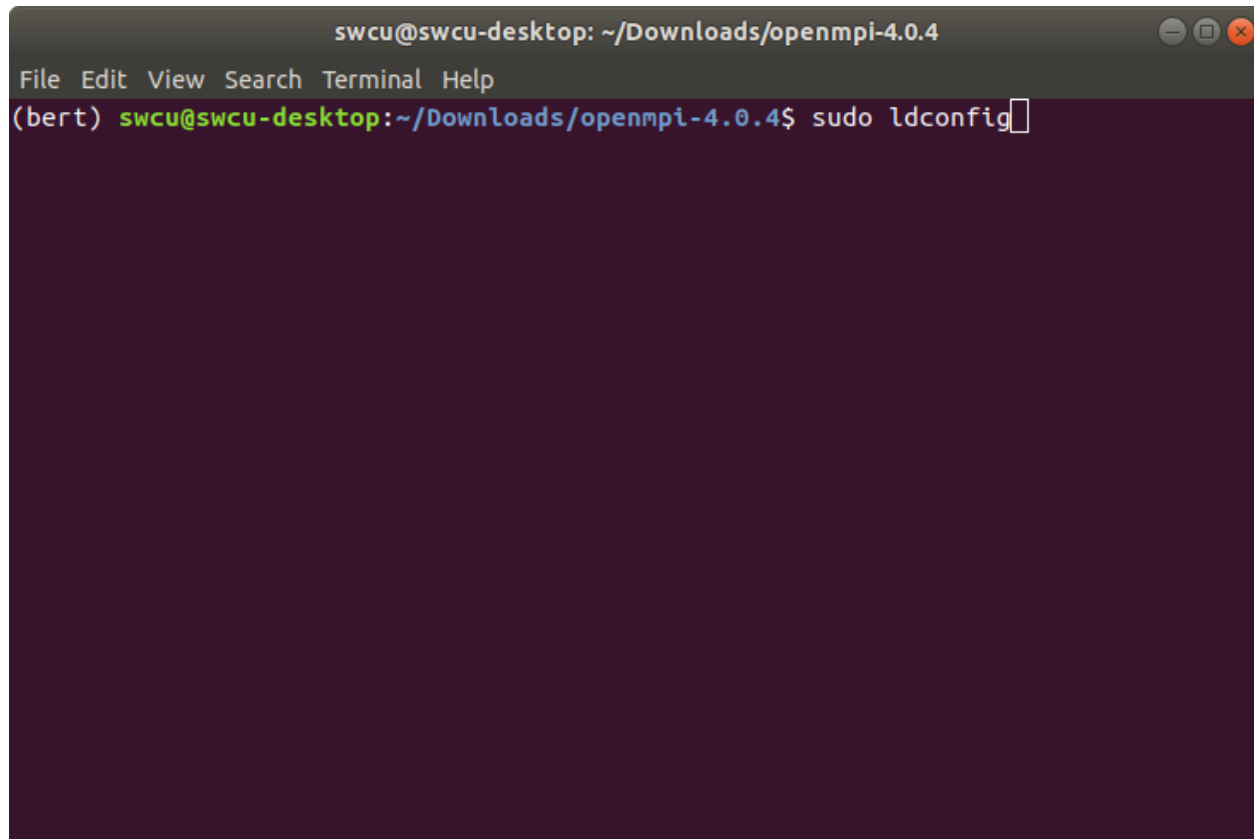
OMPIO File Systems
-----
Generic Unix FS: yes
Lustre: no
PVFS2/OrangeFS: no

(base) swcu@swcu-desktop:~/Downloads/openmpi-4.0.4$
```

02

환경 구축 및 구조

- OPENMPI



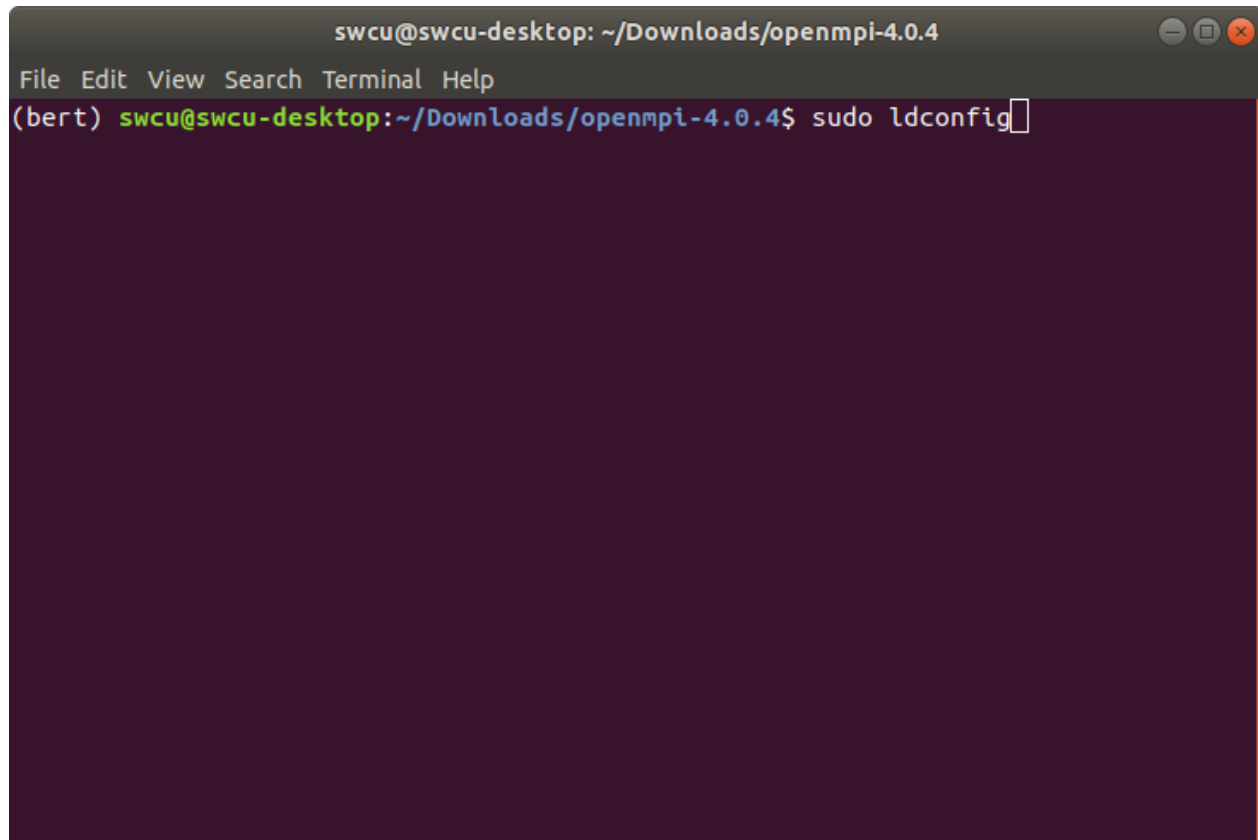
A terminal window titled "swcu@swcu-desktop: ~/Downloads/openmpi-4.0.4" with standard window controls. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The prompt "(bert) swcu@swcu-desktop:~/Downloads/openmpi-4.0.4\$" is followed by the command "sudo ldconfig" and a cursor.

```
swcu@swcu-desktop: ~/Downloads/openmpi-4.0.4
File Edit View Search Terminal Help
(bert) swcu@swcu-desktop:~/Downloads/openmpi-4.0.4$ sudo ldconfig
```

02

환경 구축 및 구조

- OPENMPI



A terminal window titled "swcu@swcu-desktop: ~/Downloads/openmpi-4.0.4" with standard window controls. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The prompt is "(bert) swcu@swcu-desktop:~/Downloads/openmpi-4.0.4\$". The command "sudo ldconfig" has been entered, and the cursor is at the end of the line.

```
swcu@swcu-desktop: ~/Downloads/openmpi-4.0.4
File Edit View Search Terminal Help
(bert) swcu@swcu-desktop:~/Downloads/openmpi-4.0.4$ sudo ldconfig
```

02

환경 구축 및 구조

- tensorflow 및 관련 라이브러리 설치

```
(bert) swcu@swcu-desktop:~/Downloads/openmpi-4.0.4$ pip install mxnet torch tensorflow-gpu==1.15
Collecting mxnet
  Downloading mxnet-1.6.0-py2.py3-none-any.whl (68.7 MB)
    | _____ | 44.8 MB 3.9 MB/s eta 0:00:07
```

```
swcu@swcu-desktop: ~/Downloads/openmpi-4.0.4
File Edit View Search Terminal Help
(bert) swcu@swcu-desktop:~/Downloads/openmpi-4.0.4$ HOROVOD_GPU_OPERATIONS=NCCL
pip install horovod3~
```


02

환경 구축 및 구조

- tensorflow 및 관련 라이브러리 설치 확인 코드

```
import tensorflow as tf
import numpy as np
import time
```

```
with tf.device("/gpu:0"):
    x_data = np.array([[0,0], [0,1], [1,0], [1,1]], dtype=np.float32)
    y_data = np.array([[0], [1], [1], [0]], dtype=np.float32)
```

```
X = tf.placeholder(tf.float32)
Y = tf.placeholder(tf.float32)
W = tf.Variable(tf.random_normal([2,1]), name='weight')
b = tf.Variable(tf.random_normal([1]), name='bias')
```

```
W1 = tf.Variable(tf.random_uniform([2, 2], -1.0, 1.0))
b1 = tf.Variable(tf.zeros([2]), name='Bias1')
L2 = tf.sigmoid(tf.matmul(X, W1) + b1)
```

```
W2 = tf.Variable(tf.random_uniform([2, 1], -1.0, 1.0))
b2 = tf.Variable(tf.zeros([1]), name='Bias2')
hypothesis = tf.sigmoid(tf.matmul(L2, W2) + b2)
```

02

환경 구축 및 구조

- tensorflow 및 관련 라이브러리 설치 확인 코드 2

여기서부터는 같은 코드를 사용합니다.

```
cost = -tf.reduce_mean(Y * tf.log(hypothesis) + (1-Y) * tf.log(1-hypothesis))
train = tf.train.GradientDescentOptimizer(learning_rate=0.1).minimize(cost)
predicted = tf.cast(hypothesis > 0.5, dtype=tf.float32)
accuracy = tf.reduce_mean(tf.cast(tf.equal(predicted, Y), dtype=tf.float32))
```

시작 시간 측정

```
start_time = time.time()
start = time.gmtime(start_time)
print("훈련 시작 : %d시 %d분 %d초"%(start.tm_hour, start.tm_min, start.tm_sec))
```

with tf.Session() as sess:

```
    sess.run(tf.global_variables_initializer())
```

```
    for step in range(20001):
```

```
        sess.run(train, feed_dict={X: x_data, Y: y_data})
```

```
        if step % 100 == 0:
```

```
            print(step, sess.run(cost, feed_dict={X: x_data, Y: y_data}), sess.run(W)) # W로 하나 W2로 하나 차이 없음.
```

```
    h, c, a = sess.run([hypothesis, predicted, accuracy], feed_dict={X: x_data, Y: y_data})
```

```
    print("\nHypothesis: ", h, "\nCorrect: ", c, "\nAccuracy: ", a)
```

```
    start_tm_min, end_start_tm_sec))
```

02

환경 구축 및 구조

- tensorflow 및 관련 라이브러리 설치 확인 코드 3

```
# 끝난 시간 측정
end_time = time.time()
end = time.gmtime(end_time)
print("훈련 끝 : %d시 %d분 %d초"%(end.tm_hour, end.tm_min, end.tm_sec))
```

```
# 소요 시간 측정
end_start = end_time - start_time
end_start = time.gmtime(end_start)
print("소요시간 : %d시 %d분 %d초"%(end_start.tm_hour, end_
```

환경 구축 및 구조

● Whole PipeLine

- NLP : BERT Model 이용

PlayList 이름

상위 10개의 Tag 예측

100개의
곡 예측

Vector간
거리 비교

Multi-Label
Classification

- Image Processing : CNN 이용해 학습

Melspectrogram
곡(음악)

Tag 예측

환경 구축 및 구조

- Multi-Label Classification
 - Supervised Learning(지도학습)
 - 가지고 있는 데이터가 Labeling이 되어 있는 경우
 - 라벨링 된 데이터(Labeled Data) : 데이터에 대한 답이 주어져 있는 것(평가가 되어 있는 것)
 - Classification
 - 지도학습 中 하나
 - 주어진 데이터를 정해진 카테고리에 따라 분류하는 문제

환경 구축 및 구조

- Multi-Label Classification

1. binary image classification problem.

- 이미지 분류할 수 있는 클래스가 2개인 경우
- 예) 자동차, 기차

2. *multi-label image classification problem*

- 이미지를 둘 이상의 클래스로 분류할 수 있는 경우

환경 구축 및 구조

- Multi-Label Classification

3. Multi-Class Image Classification



1. 이미지를 분류 할 수 있는 카테고리가 3 개 이상인 경우

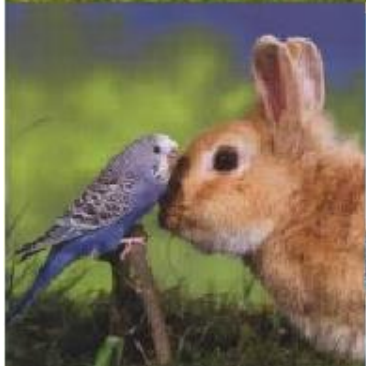
2. 이미지가 **둘 이상의** 카테고리에 속하지 않습니다

→ **Multi-Class Image Classification**

환경 구축 및 구조

- Multi-Label Classification

3. Multi-Label Image Classification



1. 이미지를 분류 할 수 있는 카테고리가 3 개 이상인 경우

2. 이미지가 **둘 이상의 카테고리에 속한다.**

→ **Multi-Label Image Classification**

03

BERT

- BERT

- Bi-directional Encoder Representations from Transformers의 약자
- Google에서 개발한 NLP(자연어 처리) 사전 훈련 기술
- 모든 자연어 처리 분야에서 좋은 성능을 내는 범용 Language Model
- 사전 훈련 언어 모델
- 대량 코퍼스로 BERT 언어모델을 적용하고, BERT언어모델 출력에 추가적인 모델(RNN, CNN 등의 머신러닝 모델)을 쌓아 원하는 Task를 수행



03

BERT

- BERT

예를 들어, 텍스트 분류모델을 만든다고 가정해 보겠습니다.

BERT를 사용하지 않은 일반 모델과정은,

: 분류를 원하는 데이터 -> LSTM, CNN 등의 머신러닝 모델 -> 분류

BERT를 사용한 모델링 과정,

: **관련 대량 코퍼스 -> BERT** -> 분류를 원하는 데이터 -> LSTM, CNN 등의 머신러닝 모델 -> 분류

대량의 코퍼스를 Encoder가 임베딩하고(언어 모델링), 이를 전이하여 파인튜닝하고 Task를 수행합니다.(NLP Task)

03

BERT

- Transformer

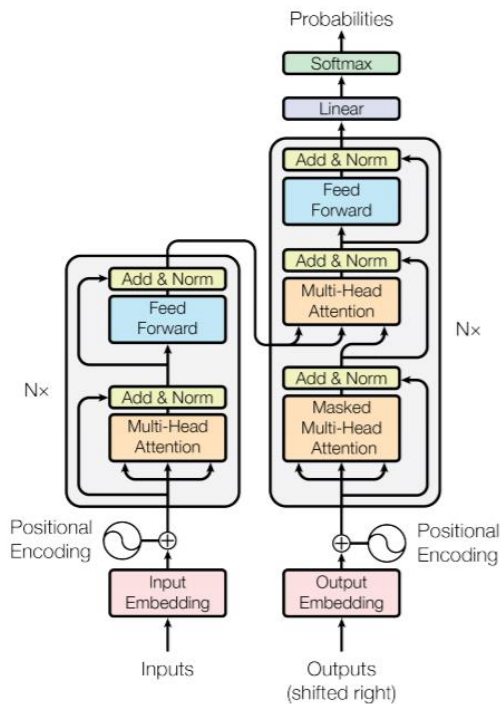


Figure 1: The Transformer - model architecture.

- Bert의 기본 구조는 Transformer라는 구조에서 Encoder를 적층시켜 만듦
- 기존의 인코더 디코더 모델을 발전시킨 딥러닝 모델
- RNN을 사용하지 않음(RNN 순차적 vs Parallelization)
-> 속도가 빠름
- Transformer 모델은 multi-head attention으로 이루어진 encoder를 여러 층 쌓아서 encoding을 수행 (encoder-decode 中 encoder만 사용)
- multi-head attention : 병렬 처리 attention layer
- Positional Encoding을 통해 위치 정보를 저장

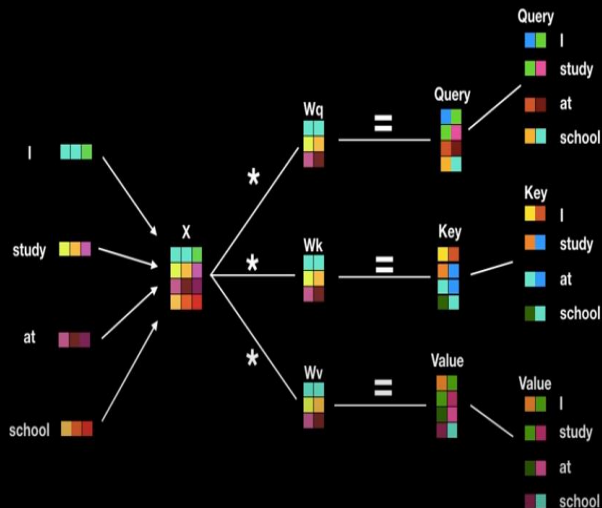
Attention is all we need !!

03

BERT

● Self-Attention

Self Attention



- Transformer에서 가장 중요한 구조
 - Encoder에서 일어나는 Attention 연산
 - Self Attention 과정
1. 입력된 토큰에 대해서 각각 독립적으로 학습 가능한 Query, Key, Value 라는 벡터를 할당
 2. 현재 단어와 다른 단어들의 상관관계를 구하기 위해 한 토큰의 Query와 나머지 토큰의 Key값을 모두 곱한하여 Attention score를 구한다. 결과(Attention score)는 각 토큰 별로 1개의 실수로 나온다. 결과가 클 수록 단어 간의 연관성이 높다.
 3. Score를 $\sqrt{\text{key vector의 차원 수}}$ 로 나눈다. (key vector의 차원이 늘어날수록 dot product의 값이 증대 되는 문제를 보완하기 위해 사용)
 4. 0~1까지의 확률의 개념으로 바꾸기 위해 Softmax를 사용하여 각 토큰별로 할당된 score 값을 확률 값으로 바꾼다.
 5. 확률 값을 각 토큰의 Value 값과 곱한다.
 6. 확률 값을 각 토큰의 Value 값과 곱들을 모두 더하여 단순한 단어가 아닌 문장 속에서 단어가 지닌 전체적인 의미를 지닌 벡터가 됨.

03

BERT

- Bert

1. Input

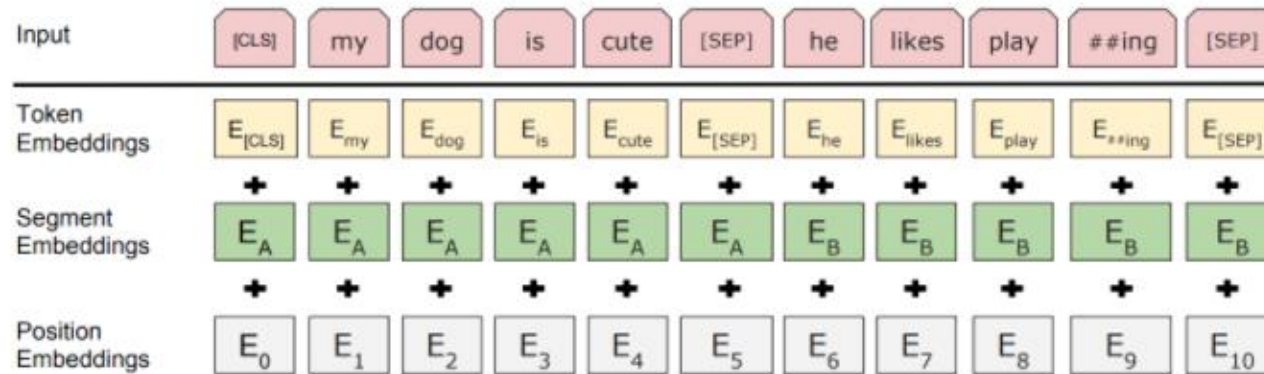


Figure 2: BERT input representation. The input embeddings is the sum of the token embeddings, the segmentation embeddings and the position embeddings.

03

BERT

- Input

- Token Embedding

- Word Piece Embedding 방식 사용, 각 문자 단위로 embedding을 하고, 자주 등장하면서 가장 긴 길이의 sub-word를 하나의 단위로 만든다.
- 자주 등장하지 않는 단어는 다시 sub-word로 만든다. -> 자주 등장하지 않은 단어를 'OOV'(Out Of Vocabulary) 처리하여 modeling의 성능을 저하했던 OOV 문제도 해결

- Segment Embedding

- 토큰으로 만든 단어들을 다시 하나의 문장으로 만드는 작업

- Position Embedding

- Self-Attention은 입력의 위치를 고려하지 않고 입력 토큰의 embedding 값을 결정
- Position Embedding : Token이 주어진 순서를 고려하여 embedding 하는 것

BERT

- Pre-Training

- MLM(Masked Language Model)

- 입력 문장에서 임의로 토큰을 버리고(Mask), Masked 토큰을 맞추는 방식으로 학습을 진행

- NSP(Next Sentence Prediction)

- 두 문장이 주어졌을 때, 두 문장의 순서를 예측하는 방식

- 두 문장 간 관련성이 고려 되어야하는 NLI와 QA의 fine tuning을 위한 두 문장의 연관을 맞추는 학습을 진행

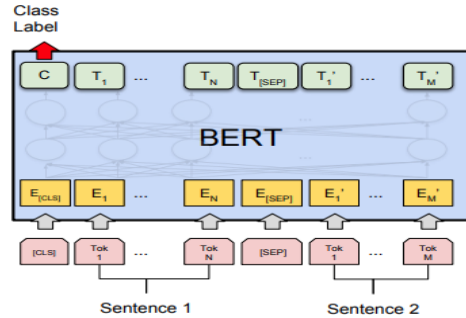
- Transfer Learning (or Fine Tuning)

학습된 언어 모델을 전이 학습(Transfer Learning)시켜 실제 NLP Task(SQUAD or GLUE, etc)를 수행하는 과정

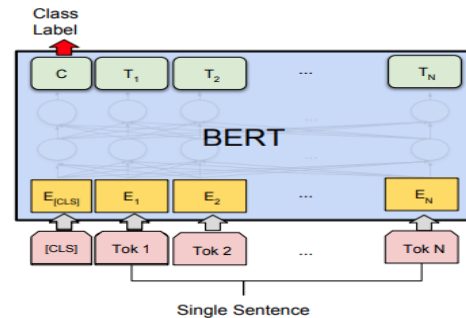
03

BERT

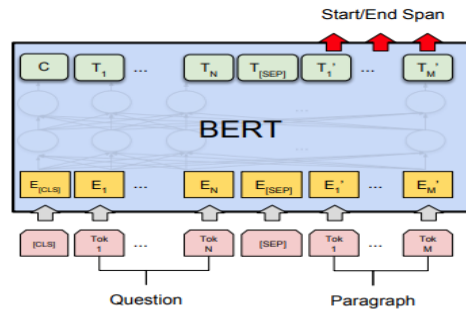
- Transfer Learning (Fine Tuning)



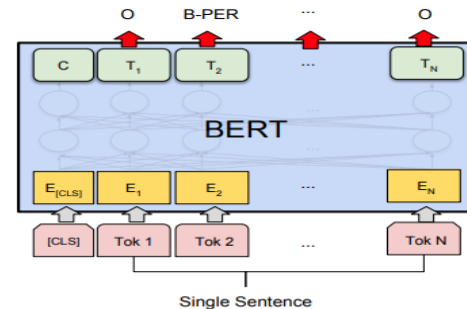
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



(b) Single Sentence Classification Tasks:
SST-2, CoLA



(c) Question Answering Tasks:
SQuAD v1.1



(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

03

BERT

- Bert 관련 출처

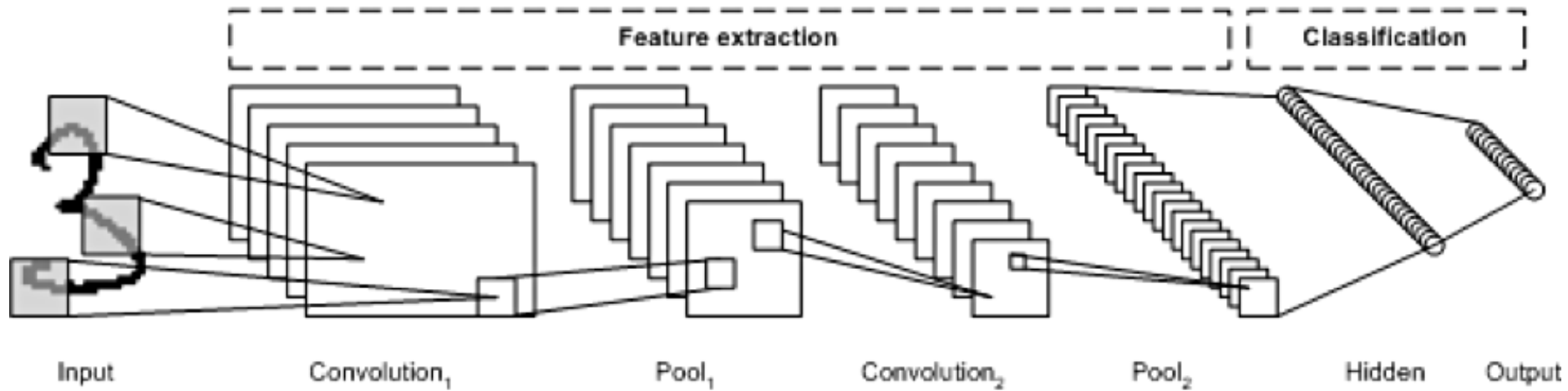
<https://ebbnflow.tistory.com/151>

<https://brunch.co.kr/@tristanmhhd/12>

<https://medium.com/platform/%EC%96%B4%ED%85%90%EC%85%98-%EB%A9%94%EC%BB%A4%EB%8B%88%EC%A6%98%EA%B3%BC-transformer-self-attention-842498fd3225>

04

CNN



04

CNN

- Fully Connected Layer 한계점 : Input Data가 1차원(배열) 형태로 한정
- Image - 2차원 이상의 Data -> 1차원으로 평면화 필요 => 공간 정보 손실
- Image 공간정보를 보존(평면화 하지 않고) 학습 가능한 모델이 필요 -> Convolution Neural Network(CNN)
- Convolution(합성곱)을 수행하는 Neural Network
- Image의 특징을 추출하는 부분과 Image를 분류하는 부분으로 구성
- Convolution Layer(Filter를 사용하여 parameter수는 최소화 시키면서 Image의 특징을 찾는 Layer)와 Pooling Layer(Image의 특징을 강화하고 모으는 Layer)를 여러 겹 쌓는 형태로 구성

- CNN 특징

1. 각 Layer의 입출력 Data의 형상 유지
2. Image의 공간 정보를 유지하면서 인접 Image와의 특징을 효과적으로 인식
3. 복수의 Filter로 Image의 특징 추출 및 학습
4. 추출한 Image 의 특징을 모으고 강화하는 Pooling Layer
5. Filter를 공유 parameter로 사용하기 때문에, 일반 NN과 비교하여 학습 parameter가 적음

04

CNN

- CNN 출처

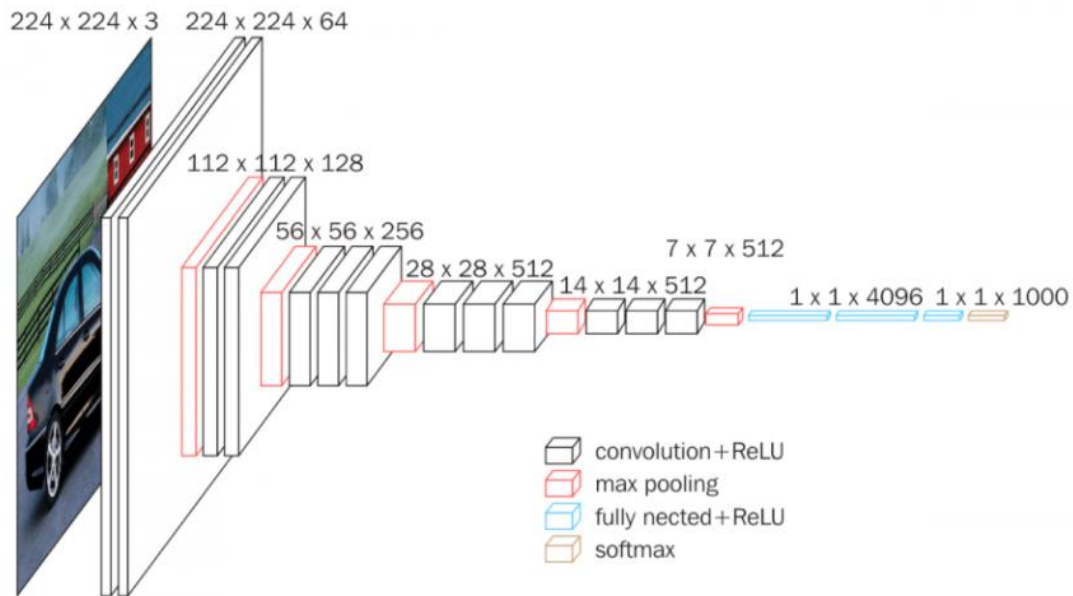
<https://medium.com/@hobinjeong/cnn-convolutional-neural-network-9f600dd3b395>

<http://taewan.kim/post/cnn/>

04

CNN

- VGG Net



VGG16 구조 [6]

- VGG Net
- Oxford 대학교의 Visual Geometry Group에서 만든 2014년 Image Net Visual Recognition Challenge(ILSVRC)대회에서 2등을 차지한 구조
- Model의 깊이가 성능에 얼마나 영향을 끼칠지에 집중하여 연구된 Model
- 간단한 구조로서 이해와 변형이 쉬움
- 단일 network에서 좋은 성능을 보여줌
- 최근에는 기본 Network model로 활용

- VGG Net 특징

- Conv Layer -> Filter의 크기 = 3×3 , stride = 1, zero padding = 1
- Pooling Layer -> Filter의 크기 = 2×2 , Max-pool
- Parameter 수가 과도하게 많아짐 -> 매우 많은 메모리를 이용해서 연산

Q. 왜 3×3 ?

A. 큰 필터 커널의 사이즈를 3×3 으로 설정했기 때문에 네트워크의 깊이를 깊게 만들 수 있다고 생각한다. 왜냐하면 필터커널의 사이즈가 크면 그만큼 이미지의 사이즈가 금방 축소되기 때문에 네트워크의 깊이를 충분히 깊게 만들기 불가능하기 때문

04

C N N

- D : VGG16, E :VGG19

A	A-LRN	B	C	D	E
11개의 레이어	11개의 레이어	13개의 레이어	16개의 레이어	16개의 레이어	19개의 레이어
conv 3, 64	conv 3, 64	conv 3, 64	conv 3, 64	conv 3, 64	conv 3, 64
	LRN	conv 3, 64	conv 3, 64	conv 3, 64	conv 3, 64
maxpool 2	maxpool 2	maxpool 2	maxpool 2	maxpool 2	maxpool 2
conv 3, 128	conv 3, 128	conv 3, 128	conv 3, 128	conv 3, 128	conv 3, 128
		conv 3, 128	conv 3, 128	conv 3, 128	conv 3, 128
maxpool 2	maxpool 2	maxpool 2	maxpool 2	maxpool 2	maxpool 2
conv 3, 256	conv 3, 256	conv 3, 256	conv 3, 256	conv 3, 256	conv 3, 256
conv 3, 256	conv 3, 256	conv 3, 256	conv 3, 256	conv 3, 256	conv 3, 256
			conv 1, 256	conv 3, 256	conv 3, 256
					conv 3, 256
maxpool 2	maxpool 2	maxpool 2	maxpool 2	maxpool 2	maxpool 2
conv 3, 512	conv 3, 512	conv 3, 512	conv 3, 512	conv 3, 512	conv 3, 512
conv 3, 512	conv 3, 512	conv 3, 512	conv 3, 512	conv 3, 512	conv 3, 512
			conv 1, 512	conv 3, 512	conv 3, 512
					conv 3, 512
maxpool 2	maxpool 2	maxpool 2	maxpool 2	maxpool 2	maxpool 2
conv 3, 512	conv 3, 512	conv 3, 512	conv 3, 512	conv 3, 512	conv 3, 512
conv 3, 512	conv 3, 512	conv 3, 512	conv 3, 512	conv 3, 512	conv 3, 512
			conv 1, 512	conv 3, 512	conv 3, 512
					conv 3, 512
maxpool 2	maxpool 2	maxpool 2	maxpool 2	maxpool 2	maxpool 2
FCN 4096	FCN 4096	FCN 4096	FCN 4096	FCN 4096	FCN 4096
FCN 4096	FCN 4096	FCN 4096	FCN 4096	FCN 4096	FCN 4096
FCN 1000	FCN 1000	FCN 1000	FCN 1000	FCN 1000	FCN 1000

04

C N N

- 참고 문헌

<https://wegonnamakeit.tistory.com/2>

<https://datascienceschool.net/view-notebook/47c57f9446224af08f13e1f3b00a774e/>

<https://blog.naver.com/laonple/220738560542>

04

CNN

- CNN 코드 설명(resize)
- 주어진 Mel-Spectrogram 의 shape = (48, ?), ? -> max =1876
- 학습하기 위해서는 Size 통일이 필요 -> (48, 1876) 으로 통일
- 주어진 데이터가 음악에 관련된 내용이므로 길이가 짧은 경우 부족한 부분을 0으로 채우는 것보다 앞에서부터 다시 중복에서 붙여오는게 곡을 더 잘 나타낸다고 생각해서 np.resize 메소드로 size를 조절

04

C N N

- CNN 코드 설명(resize)

- VGG 모델에서 기본적으로 Input으로 3차원 matrices가 필요함 + numpy에서 여러 matrices를 접합하기 위해서 미리 1차원을 준비해야함 -> Mel-Spectrogram을 (1, 48, 1876, 1)으로 resize

np.resize(ele,(1,48,1876,1)) (ele = 원본 Mel-Spectrogram 中 1곡의 npy 파일)

- np.append(append, item, axis=0)를 통해 하나의 numpy matrix로 접합하여 저장

append = np.append(append, item, axis=0) -> 접합

X = np.save("arrays"+str(num)+".npy", append) -> 저장

04

CNN

- Multi-Label classification

- 기존의 Keras 로 작성된 Image Multi-Label Classification 코드를 VGG-16 Net을 기반으로 변형

- 환경

Python = 3.7

Tensorflow =2.2

04

C N N

- Multi-Label classification

- GPU Out Of Memory 문제!

- Tensorflow는 기본적으로 거의 모든 GPU Memory를 mapping시켜서 사용 (Memory 파편화를 줄이기 위함)

- 그로 인해 GPU Memory 관련 문제가 자주 발생 -> sol) tensorflow가 사용하는 GPU Memory를 제한

```
# Solution
```

```
gpus = tf.config.experimental.list_physical_devices('GPU')
```

```
# 프로그램 시작시에 가상 장치가 설정
```

```
if gpus:
```

```
    try:
```

```
        tf.config.set_visible_devices(gpus[0], 'GPU')
```

```
        tf.config.experimental.set_virtual_device_configuration(  
            gpus[0],
```

```
[tf.config.experimental.VirtualDeviceConfiguration(memory_limit=10000)])
```

```
# 텐서플로가 첫 번째 GPU에 1GB 메모리만 할당하도록 제한
```

```
except RuntimeError as e:
```

```
    print(e)
```

- Multi-Label classification

• 기존 VGG 16-Net에서
 Conv 3x3, feature map 512
 Conv 3x3, feature map 512
 Conv 3x3, feature map 512
 Maxpool 2x2

4개의 층을 제외하고 Model을 쌓음

-> GPU Memory 초과

(class를 29160개로 잡고 진행해서 더 parameter가 폭발함)

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 46, 1874, 64)	640
conv2d_1 (Conv2D)	(None, 44, 1872, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 22, 936, 64)	0
dropout (Dropout)	(None, 22, 936, 64)	0
conv2d_2 (Conv2D)	(None, 20, 934, 128)	73856
conv2d_3 (Conv2D)	(None, 18, 932, 128)	147584
max_pooling2d_1 (MaxPooling2D)	(None, 9, 466, 128)	0
dropout_1 (Dropout)	(None, 9, 466, 128)	0
conv2d_4 (Conv2D)	(None, 7, 464, 256)	295168
conv2d_5 (Conv2D)	(None, 5, 462, 256)	590080
conv2d_6 (Conv2D)	(None, 3, 460, 256)	590080
max_pooling2d_2 (MaxPooling2D)	(None, 1, 230, 256)	0
dropout_2 (Dropout)	(None, 1, 230, 256)	0
dropout_3 (Dropout)	(None, 1, 230, 256)	0
flatten (Flatten)	(None, 58880)	0
dense (Dense)	(None, 4096)	241176576
dropout_4 (Dropout)	(None, 4096)	0
dense_1 (Dense)	(None, 1024)	4195328
dropout_5 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 29160)	29889000
=====		
Total params: 276,995,240		
Trainable params: 276,995,240		
Non-trainable params: 0		

04

CNN

- Multi-Label classification

- 모델 변형 부분 (변형 부분은 굵게 작성)

```
# model.add(Conv2D(filters=64, kernel_size=(3, 3), activation="relu", input_shape=(48,  
1876,1))) -> input shape = resize한 shape
```

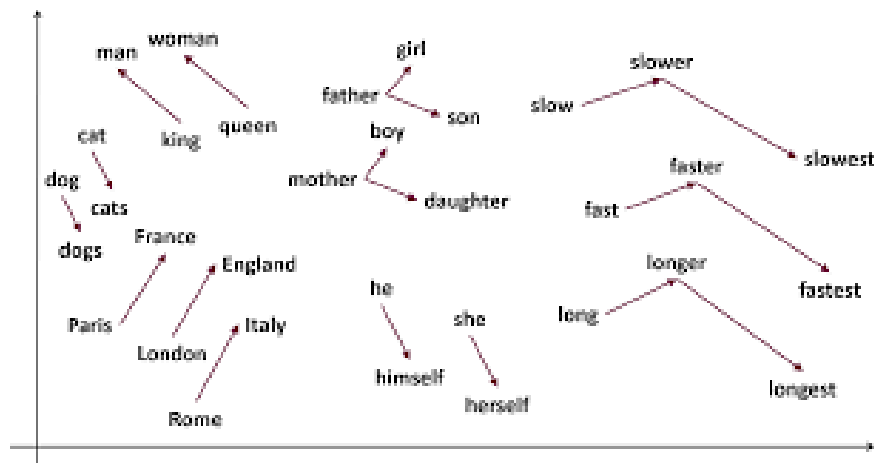
```
# model.add(Dense(29160, activation='sigmoid')) -> class 갯수 = 29160
```


05

Word2Vec

- word2vec

- 자연어 처리를 위한 기술
- 신경망 모델을 사용 하여 큰 텍스트 코퍼스에서 단어 연관성을 학습
- 단어 임베딩 을 생성하는 데 사용되는 관련 모델 그룹
- 속도가 느림



05

Word2Vec

- 워드 임베딩(Word Eembedding)
 - 밀집 벡터의 형태로 표현하는 방법
 - 밀집 표현(Dense Representation) : 벡터의 차원을 단어의 집합의 크기로 상정하는 것이 아니라 사용자가 설정한 값으로 모든 단어의 벡터 표현의 차원으로 표현
 - EX) LSA, Word2Vec, FastText, Glove 등

-	원-핫 벡터	임베딩 벡터
차원	고차원(단어 집합의 크기)	저차원
다른 표현	희소 벡터의 일종	밀집 벡터의 일종
표현 방법	수동	훈련 데이터로부터 학습함
값의 타입	1과 0	실수

05

Word2Vec

- 분산 표현

- 희소 표현 : 원-핫 벡터, 표현하고자 하는 단어의 인덱스의 값이 1이고 그 외는 0
- 분산 표현 : '비슷한 위치에서 등장하는 단어들은 비슷한 의미를 가진다' 라는 분포 가설이라는 가정 하에 만들어진 표현 방법
- 단어의 의미를 여러 차원에다가 분산하여 표현을 이용해 단어 간 유사도 계산

05

Word2Vec

● CBOW

- 주변에 있는 단어들을 가지고 중간에 있는 단어들을 예측하는 방법
- 슬라이딩 윈도우 : 윈도우 계속 움직여서 주변 단어와 중심 단어 선택을 바꿔가며 학습하는 방법

중심 단어 주변 단어

↓ ↓

The fat cat sat on the mat

The fat cat sat on the mat

The fat cat sat on the mat

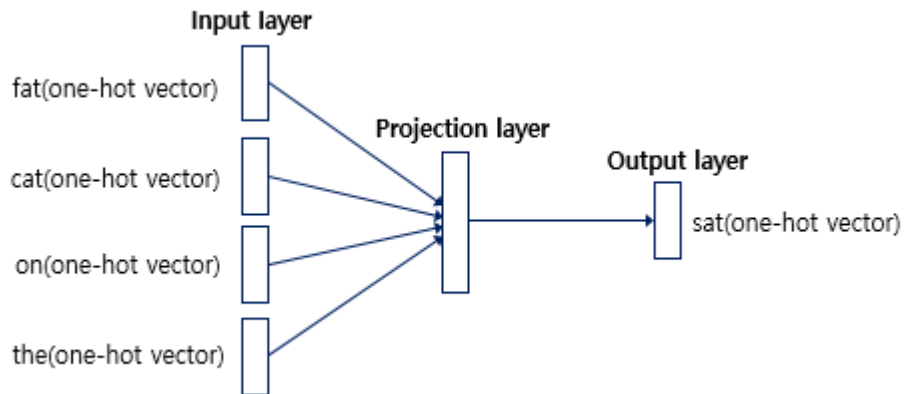
The fat cat sat on the mat

The fat cat sat on the mat

The fat cat sat on the mat

The fat cat sat on the mat

중심 단어	주변 단어
[1, 0, 0, 0, 0, 0, 0]	[0, 1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0]
[0, 1, 0, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0]
[0, 0, 1, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0, 0], [0, 1, 0, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0]
[0, 0, 0, 1, 0, 0, 0]	[0, 1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0]
[0, 0, 0, 0, 1, 0, 0]	[0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 1, 0]
[0, 0, 0, 0, 0, 1, 0]	[0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 0, 0, 1]	[0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 1, 0]

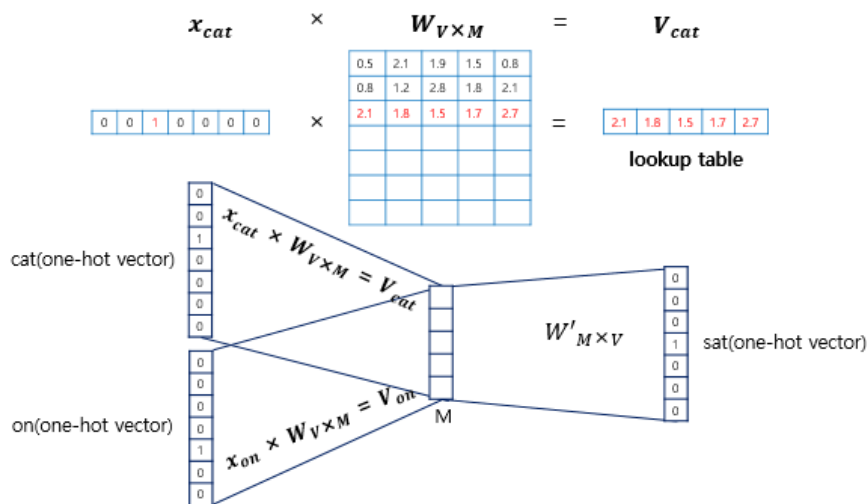
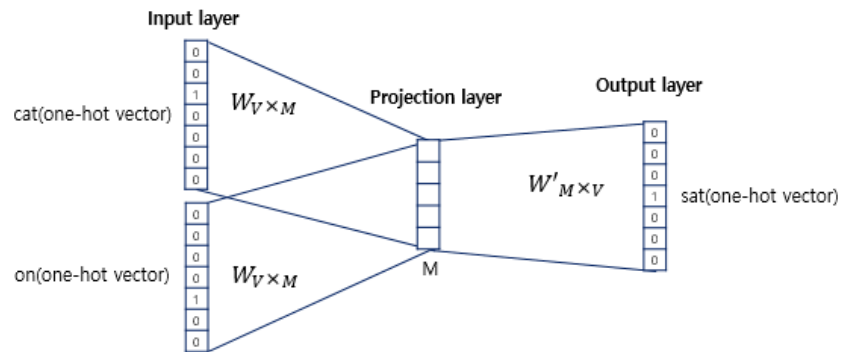


05

Word2Vec

● CBOW

- 인공 신경망의 훈련 전에 이 가중치 행렬 W 와 W' 는 대개 굉장히 작은 랜덤 값
- 주변 단어로 중심 단어를 더 정확히 맞추기 위해 계속해서 W 와 W' 를 학습해가는 구조.
- Projection layer : 룩업 테이블이라는 연산을 담당하는 층
- 룩업 테이블 : i 번째 인덱스에 1이라는 값을 가지고 그 외의 0의 값을 가지는 입력 벡터와 가중치 W 행렬의 곱은 사실 W 행렬의 i 번째 행을 그대로 읽어오는 작업

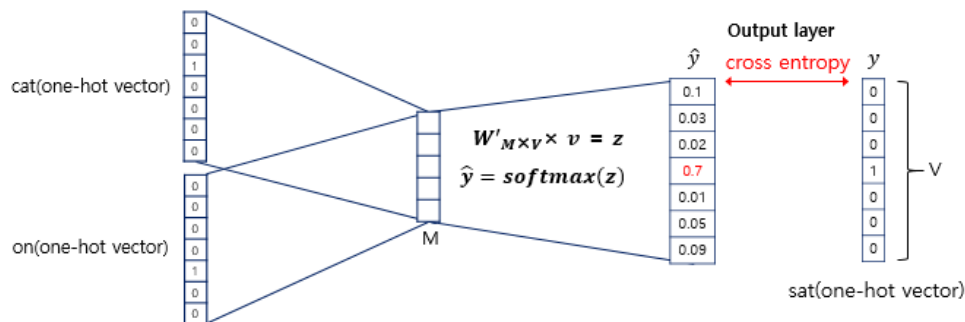
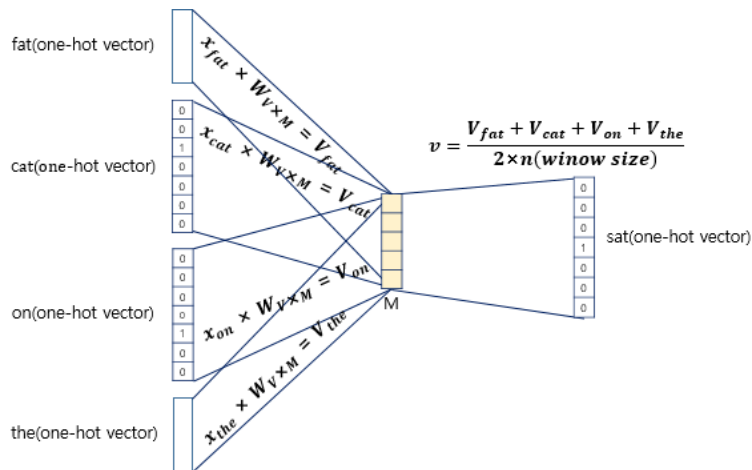


05

Word2Vec

● CBOW

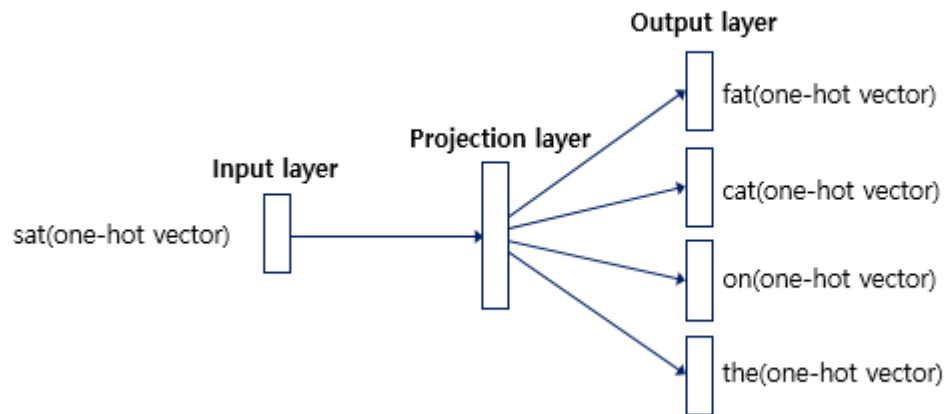
- 각 주변 단어의 원-핫 벡터에 대해서 가중치 W 가 곱해서 생겨진 결과 벡터들은 투사층에서 만나 이 벡터들의 평균인 벡터를 구한다.
- 스코어 벡터(score vector) : 구한 데이터에 softmax함수를 사용하여 출력값은 0과 1사이 값이고 총 스코어 벡터의 합이 1인 벡터



Word2Vec

- Skip-gram

- 중간에 있는 단어로 주변 단어들을 예측하는 방법
- CBOW와 달리 중심단어에서 예측하므로 벡터들의 평균을 구하는 과정이 없음
- 전반적으로 Skip-gram이 CBOW보다 성능이 좋다고 알려져 있음



05

Word2Vec

- 참고 문헌 및 자료

<https://wikidocs.net/33520>

<https://medium.com/analytics-vidhya/implementing-word2vec-in-tensorflow-44f93cf2665f>