1. 이 책에서 다루는 내용

이 책은 2018년 BERT가 발표된 이후의 자연어처리에 대한 내용을 담고 있다. 2018년은 자연어처리의 역사에 있어서 중요한 발전이 있었던 해이다. 2018년에는 Attention is all you need라는 논문이 발표되면서 기존의 RNN 계열의 인공지능 모델에서 트랜스포머 기반의 인공지능 모델로 변해가기 시작했다. 트랜스포머의 출연으로 언어 모델의 성능은 급격하게 발전하게 됐고, 파인튜닝을 통해 적은 데이터로 고성능의 모델을 쉽고 빠르게 만들 수 있게 됐다. 이 책에서는 확률 기반의 언어 모델인 N-Gram 언어 모델부터 최근에 발표된 GPT3까지의 발전 과정을 설명했다.

1장에서는 언어 모델에 관한 내용을 다뤘다. 언어 모델이 무엇인지 확률적으로 정의한 후, N-gram 기법을 이용해서 언어 모델을 설명했고 그 과정을 코드를 통해 구현했다. 또한 N-gram 이후에 오랫동안 사용됐던 인공지능 모델인 RNN 계열의 인공지능 모델도 설명했다. RNN 계열의 언어 모델도 간단하게 학습해볼 수 있도록 샘플 코드를 구현했다.

2장에서는 어탠션에 대해서 자세하게 다뤘다. 어탠션은 트랜스포머의 핵심을 이루는 연산이다. 어탠션 연산을 RNN 계열에 추가했을 때 어떠한 장점이 있는지 설명했고 관련된 샘플 코드도 공부해볼 수 있도록 구현했다.

3장부터 본격적으로 트랜스포머의 구조를 다뤘다. 2장에서 공부했었던 어탠션을 기반으로 트랜스포머를 어떻게 구현했는지 구조적으로 설명했다.

4장에서는 BERT 모델에 대해서 설명했다. BERT를 사전학습하는 방법에 대해서 소개했고, 사전학습된 모델을 파인튜닝하는 방법에 대해서도 소개했다. 또한 BERT 이후에 발표된 개선된 모델도 몇가지 소개했다.

마지막으로 5장에서는 GPT 계열의 모델을 설명했다. GPT2에서부터는 메타러닝의 개념이 추가된다. 자연어처리에서 메타러닝이 왜 필요하고 어떤 방법으로 학습되는지 소개했다. 다만 GPT2/GPT3의 정확한 학습 방법은 공개되지 않았기 때문에 아주 자세한 학습 과정은 코드로 구현하지 못했다.

2. 대상 독자

이 책은 어느 정도의 파이선 코드를 읽고 구현할 수 있으며 머신러닝에 대한 기초적인 역량을 가지고 있는 독자를 대상으로 저술했다. 파이선을 한번도 접해보지 않았거나 머신러닝을 처음 입문하는 독자가 공부하기에는 다소 난해한 내용일 수 있다.

3. 이 책의 활용법

주피터 노트북이나 파이선 실행 파일을 통해서 직접 코드를 실행시켜보면서 공부하길 권한다. 이 책은 많은 소스 코드를 레포지토리 형태로 제공하고 있다.(https://github.com/jinkilee/hello-transformer) 지금까지 머신러닝 공부를 하고 현업에서 일을 하면서 가장 구체적으로 도움이 됐었던 것은 수식과 그 수식을 구현한 코드였다.

3.1. 리눅스 환경에서 소스 코드 실행하기

리눅스 환경에서 소스 코드를 실행할 경우 [블록1]과 같이 진행하면 된다.

<블록 시작>

블록1: 리눅스 환경에서 가상환경 준비하기

# git과 가상환경 실행하기

$ sudo apt-get install git virtualenv

Reading package lists... Done

Building dependency tree

Reading state information... Done

git is already the newest version (1:2.25.1-1ubuntu3.2).

The following packages were automatically installed and are no longer required:

fonts-cantarell gdbserver libbabeltrace1 libdw1 libnuma1

Use 'sudo apt autoremove' to remove them.

The following additional packages will be installed:

python-pip-whl python3-appdirs python3-distlib python3-filelock python3-virtualenv

The following NEW packages will be installed:

python-pip-whl python3-appdirs python3-distlib python3-filelock python3-virtualenv virtualenv

0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.

Need to get 2005 kB of archives.

After this operation, 3367 kB of additional disk space will be used.

Do you want to continue? [Y/n] Y

Get:1 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates/universe amd64 python-pip-whl all 20.0.2-5ubuntu1.6 [1805 kB]

.....

Unpacking virtualenv (20.0.17-1ubuntu0.4) ...

Setting up python3-filelock (3.0.12-2) ...

Setting up python3-distlib (0.3.0-1) ...

Setting up python-pip-whl (20.0.2-5ubuntu1.6) ...

Setting up python3-appdirs (1.4.3-2.1) ...

Setting up python3-virtualenv (20.0.17-1ubuntu0.4) ...

Setting up virtualenv (20.0.17-1ubuntu0.4) ...

Processing triggers for man-db (2.9.1-1) ...

# 소스 코드 클론 받은 후 프로젝트 폴더로 이동하기

$ git clone https://github.com/jinkilee/hello-transformer

Cloning into 'hello-transformer'...

remote: Enumerating objects: 1596, done.

remote: Counting objects: 100% (1596/1596), done.

remote: Compressing objects: 100% (883/883), done.

remote: Total 1596 (delta 975), reused 1310 (delta 689), pack-reused 0

Receiving objects: 100% (1596/1596), 24.58 MiB | 36.97 MiB/s, done.

Resolving deltas: 100% (975/975), done.

$ cd hello-transformer

# 파이선 버전 3.6 이상으로 가상환경 생성

$ virtualenv -p /usr/bin/python3.6 venv

created virtual environment CPython3.8.10.final.0-64 in 312ms

creator CPython3Posix(dest=/home/jinkilee73/hello-transformer/venv, clear=False, global=False)

seeder FromAppData(download=False, pip=latest, setuptools=latest, wheel=latest, pkg\_resources=latest, via=copy, app\_data\_dir=/home/jinkilee73/.local/share/virtualenv/seed-app-data/v1.0.1.debian.1)

activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator

# 가상환경 실행

$ source venv/bin/activate

# 가상환경이 정상적으로 실행되면 아래와 같이 쉘에 (venv)라고 적혀있음.

(venv) $

# 가상환경에서 requirements.txt 설치하기

(venv) $ pip install -r requirements.txt

Collecting transformers==4.1.0

Downloading transformers-4.1.0-py3-none-any.whl (1.5 MB)

|████████████████████████████████| 1.5 MB 5.2 MB/s

Collecting jupyter

Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)

Collecting sklearn

Downloading sklearn-0.0.tar.gz (1.1 kB)

...

Successfully built sklearn

Installing collected packages: urllib3, charset-normalizer, certifi, idna, requests, pyparsing, packaging, joblib, regex, click, six, tqdm, sacremoses, tokenizers, filelock, numpy, transformers, pyzmq, entrypoints, tornado, traitlets, jupyter-core, python-dateutil, nest-asyncio, jupyter-client, matplotlib-inline, parso, jedi, pygments, pure-eval, asttokens, executing, stack-data, pickleshare, ptyprocess, pexpect, decorator, wcwidth, prompt-toolkit, backcall, ipython, debugpy, ipykernel, ipython-genutils, qtpy, qtconsole, jupyter-console, prometheus-client, MarkupSafe, jinja2, pycparser, cffi, argon2-cffi-bindings, argon2-cffi, attrs, pyrsistent, zipp, importlib-resources, jsonschema, nbformat, terminado, testpath, nbclient, webencodings, bleach, mistune, pandocfilters, defusedxml, jupyterlab-pygments, nbconvert, Send2Trash, notebook, jupyterlab-widgets, widgetsnbextension, ipywidgets, jupyter, scipy, threadpoolctl, scikit-learn, sklearn, pillow, cycler, fonttools, kiwisolver, matplotlib, typing-extensions, torch, nltk

Successfully installed MarkupSafe-2.1.0 Send2Trash-1.8.0 argon2-cffi-21.3.0 argon2-cffi-bindings-21.2.0 asttokens-2.0.5 attrs-21.4.0 backcall-0.2.0 bleach-4.1.0 certifi-2021.10.8 cffi-1.15.0 charset-normalizer-2.0.12 click-8.0.4 cycler-0.11.0 debugpy-1.5.1 decorator-5.1.1 defusedxml-0.7.1 entrypoints-0.4 executing-0.8.2 filelock-3.6.0 fonttools-4.29.1 idna-3.3 importlib-resources-5.4.0 ipykernel-6.9.1 ipython-8.1.0 ipython-genutils-0.2.0 ipywidgets-7.6.5 jedi-0.18.1 jinja2-3.0.3 joblib-1.1.0 jsonschema-4.4.0 jupyter-1.0.0 jupyter-client-7.1.2 jupyter-console-6.4.0 jupyter-core-4.9.2 jupyterlab-pygments-0.1.2 jupyterlab-widgets-1.0.2 kiwisolver-1.3.2 matplotlib-3.5.1 matplotlib-inline-0.1.3 mistune-0.8.4 nbclient-0.5.11 nbconvert-6.4.2 nbformat-5.1.3 nest-asyncio-1.5.4 nltk-3.7 notebook-6.4.8 numpy-1.22.2 packaging-21.3 pandocfilters-1.5.0 parso-0.8.3 pexpect-4.8.0 pickleshare-0.7.5 pillow-9.0.1 prometheus-client-0.13.1 prompt-toolkit-3.0.28 ptyprocess-0.7.0 pure-eval-0.2.2 pycparser-2.21 pygments-2.11.2 pyparsing-3.0.7 pyrsistent-0.18.1 python-dateutil-2.8.2 pyzmq-22.3.0 qtconsole-5.2.2 qtpy-2.0.1 regex-2022.1.18 requests-2.27.1 sacremoses-0.0.47 scikit-learn-1.0.2 scipy-1.8.0 six-1.16.0 sklearn-0.0 stack-data-0.2.0 terminado-0.13.1 testpath-0.6.0 threadpoolctl-3.1.0 tokenizers-0.9.4 torch-1.10.2 tornado-6.1 tqdm-4.62.3 traitlets-5.1.1 transformers-4.1.0 typing-extensions-4.1.1 urllib3-1.26.8 wcwidth-0.2.5 webencodings-0.5.1 widgetsnbextension-3.5.2 zipp-3.7.0

<블록 끝>

[블록1]의 내용을 모두 실행하면 이 책의 소스 코드를 실행할 수 있는 환경이 준비된 것이다. 파이선 파일을 실행할 경우에는 [블록2]와 같이 python 명령어를 통해서 실행하면 된다.

<블록 시작>

블록2: 가상환경에서 파이선 파일 실행하기

(venv) $ python hello-transformer.py

hello. transformer

<블록 끝>

주피터를 실행시킬 경우에는 [블록3]과 같이 실행하면 된다.

<블록 시작>

블록3: 가상환경에서 주피터 실행하기

(venv) $ jupyter notebook

[I 20:26:26.050 NotebookApp] Serving notebooks from local directory: /home/jinkilee73/hello-transformer

[I 20:26:26.050 NotebookApp] Jupyter Notebook 6.4.8 is running at:

[I 20:26:26.050 NotebookApp] http://localhost:8888/?token=922d72023eab7a6288beb8d95cfd488bab1b65ee16494650

[I 20:26:26.051 NotebookApp] or http://127.0.0.1:8888/?token=922d72023eab7a6288beb8d95cfd488bab1b65ee16494650

[I 20:26:26.051 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[W 20:26:26.055 NotebookApp] No web browser found: could not locate runnable browser.

[C 20:26:26.055 NotebookApp]

To access the notebook, open this file in a browser:

file:///home/jinkilee73/.local/share/jupyter/runtime/nbserver-5834-open.html

Or copy and paste one of these URLs:

http://localhost:8888/?token=922d72023eab7a6288beb8d95cfd488bab1b65ee16494650

or http://127.0.0.1:8888/?token=922d72023eab7a6288beb8d95cfd488bab1b65ee16494650

[I 20:28:01.885 NotebookApp] 302 GET /?token=922d72023eab7a6288beb8d95cfd488bab1b65ee16494650 (127.0.0.1) 0.740000ms

<블록 끝>

[블록3]과 같이 실행할 경우 주피터 노트북을 실행시킬 수 있다.

<그림 시작>

그림1

<그림 끝>

주피터 노트북을 실행시키면 [그림1]과 같이 브라우저를 통해서 접근할 수 있다.

3.2. Windows 환경에서 소스 코드 실행하기

Windows에서 소스 코드를 실행할 경우 Pycharm을 이용해서 실행하면 편리하다. Pycharm은 무료로 제공되는 Community 버전을 사용해도 된다.

<그림 시작>

그림2

<그림 끝>

Pycharm의 상단을 보면 [Git] >> [Clone]을 클릭해서 이 책의 소스코드 레포지토리를 클론 받을 수 있다.

<그림 시작>

그림3

<그림 끝>

이 후 [그림4]과 같이 자동으로 가상 환경을 설치하는 창이 열린다. 창이 열리면 [OK] 버튼을 눌러서 가상 환경을 설치하자.

<그림 시작>

그림4

<그림 끝>

Pycharm 환경에서 requirements.txt를 설치하려면 Pycharm 하단에 [Terminal]에 접근해서 직접 pip 명령어를(pip install -r requirements.txt) 실행시켜주면 된다.

<그림 시작>

그림5

<그림 끝>

requirements.txt가 모두 설치된 후에 파이선 파일을 실행시키려면 좌측 프로젝트 구조에서 실행시키고자 하는 파이선 파일을 클릭 후 파이선 인터프리터를 설정해서 실행시켜주면 된다. 파이선 인터프리터는 Pycharm의 우측 상단에 [Add Configuration]을 클릭해서 실행 환경 설정 창을 열고 [Add new]를 클릭해서 [그림10]에서 설치한 가상환경을 선택해주면 된다. 이 때 실행시킬 스크립트도 [Script path]에 추가한다. [그림6]를 참고하라.

<그림 시작>

그림6

<그림 끝>

마지막으로 주피터 노트북을 실행시키려면 [그림5]에서와 마찬가지로 [Terminal] 창에서 명령어를(jupyter notebook) 실행시켜주면 된다.

<그림 시작>

그림7

<그림 끝>

jupyter notebook을 실행시켰을 때 토큰 값이 나오지 않아서 브라우저에서 로그인을 할 수 없다면 [블록4]와 같이 password 부분을 없애고 로그인하면 로그인 없이 접근할 수 있다. 이렇게 사용할 경우 보안에 유의해서 사용하도록 하자.

<블록 시작>

블록4

jupyter notebook --NotebookApp.token='' --NotebookApp.password=''

<블록 끝>

4. 질의 응답

이 책에 내용에 대해서 질문이 있을 경우 레포지토리의 [Issue]를 통해서 등록하면 된다.

이슈 페이지 주소: https://github.com/jinkilee/hello-transformer/issues

위 링크로 접근 한 후에 우측에 초록색 [New Issue] 버튼을 눌러서 이슈를 등록하면 된다. 그림8과 같이 접근하면 된다.

<그림 시작>

그림8

<그림 끝>