세종대 캡스톤 디자인 2조- 16s

CTA(Celeb's Tweet Analysis)

희소행렬 처리를 통한 연산속도 증가

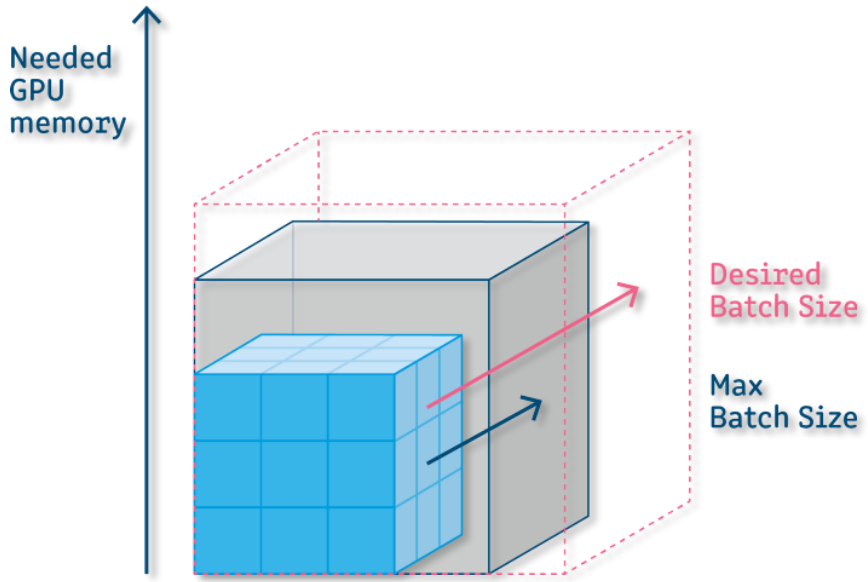
01.

목차

문제 인식

Batch size and GPU memory limitations in neural networks | Towards Data Science

개발 동기



딥러닝 모델 성능과 크기가 비례하는 경향으로

인해 생기는 메모리 부담 및 속도 저하 문제

문제점

1. 배치 사이즈의 증가는 메모리의 부담을 가져옴

2. 모델이 커짐에 따라 학습에 많은 시간 소요됨

3. GPU는 실용적인 측면에서 부담이 될 수도 있음

모델 압축(model compression) 작업 필요

문제 인식

개발 목적

01. 모델 성능 향상

convolution 연산 방식을 수정해 메모리 접근 효율을 증가

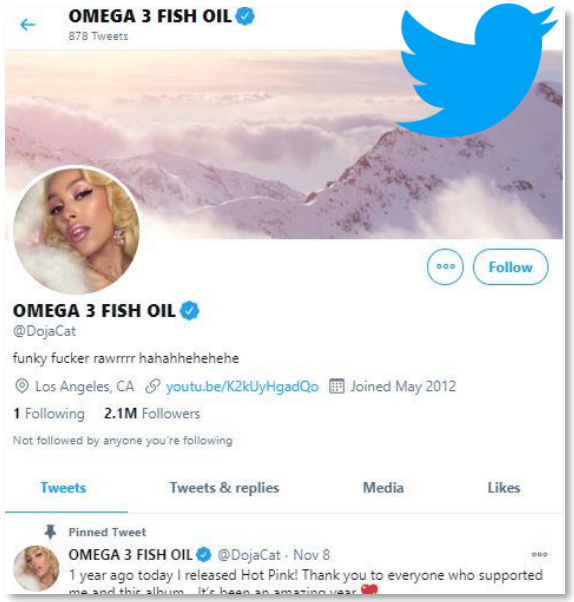
수행시간 감소, 메모리/cpu 사용량 감소

02. 딥러닝 활용 프로그램 개발

트위터 API 를 통해 유명인의 트윗 데이터 감성 분석

스낵 콘텐츠에 맞는 빠르고 재밌는 프로그램 개발

유명인의 트윗을 긍정/부정/중립 세 가지 감성으로 분류



OMEGA 3 FISH OIL (@DojaCat) / Twitter

각광받는 자연어처리(NLP)분야 모델을 활용해

누구나 재밌게 즐길 수 있는 콘텐츠 개발

문제 해결

기능 구성

문장 전처리

문장 전처리
: URL,소문자,@
,해시태그,이모지 변환

“Thank you @Spotify for making my mug the cover of Pop List playlist 💥 htt
ps://t.co/3wGHLaTIqm”

“thank you HASHTAG for making my mug the cover of pop list playlist
:collision: URL”

토큰화
문자열의 각 단어에
인덱스 부여

[[429], [1408], [4], [267], [747], [], [714], [321], [53], [], [403], [228], [321],
[429], [1], [994], [714],... [739], [4], [714], [739], [1], [403], [429]]

문제 해결

기능 구성

모델 구성

embedding layer(1)

convolution layer(4)

maxpooling layer(4)

dropoupt layer(4)

flatten(1)

dense(1)

compile : optimizer = 'SGD', loss ='mse'

모델 빌드

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 50, 32)            12800000
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 50, 128)           20608
_________________________________________________________________
max_pooling1d_4 (MaxPooling1 (None, 25, 128)           0
_________________________________________________________________
dropout_4 (Dropout)          (None, 25, 128)           0
_________________________________________________________________
conv1d_5 (Conv1D)            (None, 25, 64)            49216
_________________________________________________________________
max_pooling1d_5 (MaxPooling1 (None, 12, 64)            0
_________________________________________________________________
dropout_5 (Dropout)          (None, 12, 64)            0
_________________________________________________________________
conv1d_6 (Conv1D)            (None, 12, 32)            14368
_________________________________________________________________
max_pooling1d_6 (MaxPooling1 (None, 6, 32)             0
_________________________________________________________________
dropout_6 (Dropout)          (None, 6, 32)             0
_________________________________________________________________
conv1d_7 (Conv1D)            (None, 6, 32)             8224
_________________________________________________________________
max_pooling1d_7 (MaxPooling1 (None, 3, 32)             0
_________________________________________________________________
dropout_7 (Dropout)          (None, 3, 32)             0
_________________________________________________________________
flatten_1 (Flatten)          (None, 96)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 97
=================================================================
Total params: 12,892,513
Trainable params: 12,892,513
Non-trainable params: 0
_________________________________________________________________
```

문제 해결

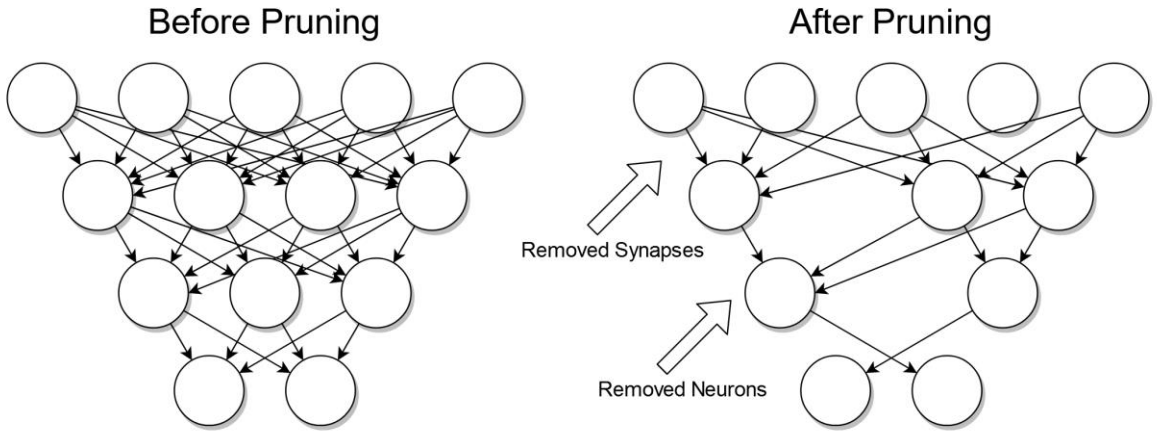
기능 구성

가중치 프루닝 적용

학습 프로세스 과정에서 가중치 프루닝을 통해 모델

가중치를 점차적으로 제로화하여 희소행렬을 생성

- 70, 75, 80, 85 총 4단계 진행



프루닝 적용 전후 신경망 이미지

문제 해결

기능 구성

NZB 인덱싱 적용 모델

NZB

weight의 0이 아닌 원소를 1, 0인 원소를 0인 비트맵으로 표현

비트맵 구성: 0이 아닌 원소개수 2진수 표현+ mask bit

실제 Convolution layer 1 에서 160개의 원소 중 1의 개수를 나타내는

bit 8개하고 + 5*32 = 160 원소에서 값이 존재하는 부분에 1을 표시

```
168
array([0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1,
       1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0], dtype=uint8)
```

(NZB Indexing 함수를 통해서 bitmap 저장한 결과)

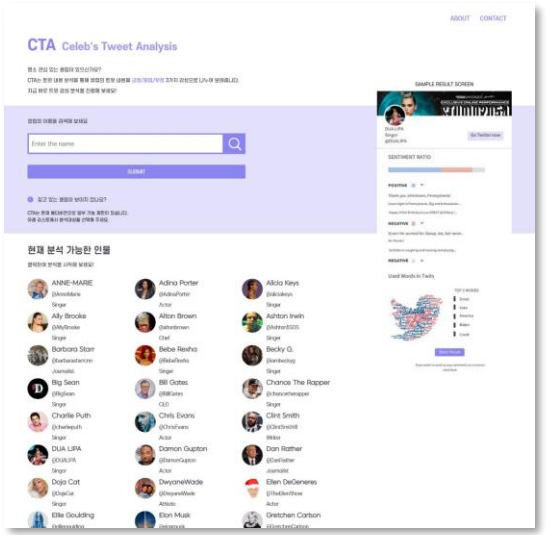
문제 해결

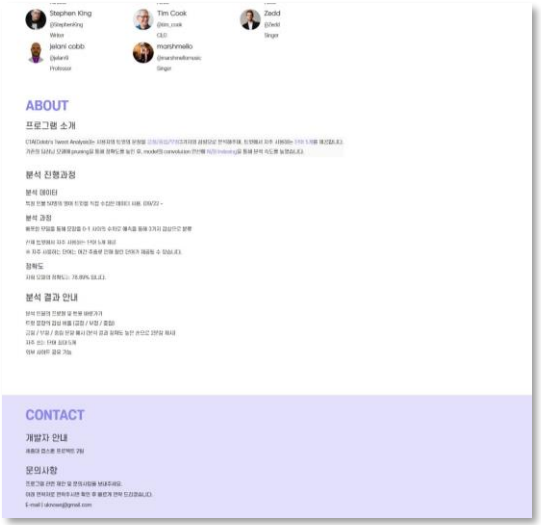
기능 구성

웹 서비스

메인 페이지

- 인물 검색
- 분석 가능 인물 보기 & 선택
- ABOUT 페이지
- CONTACT 페이지





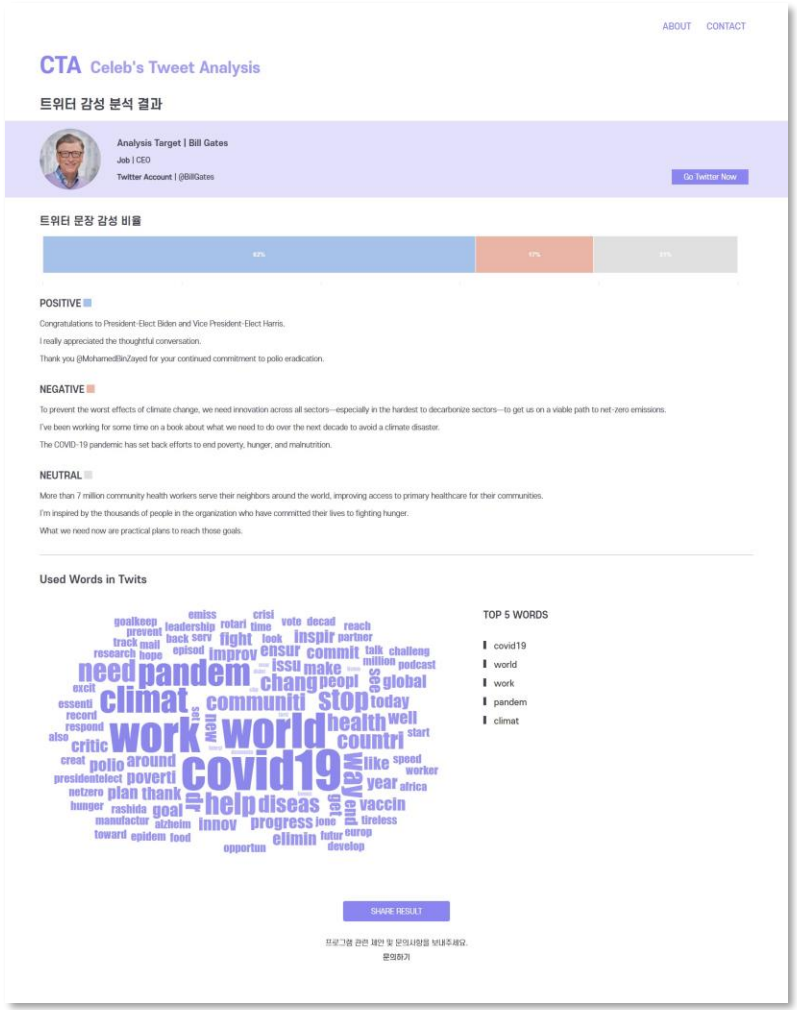
문제 해결

기능 구성

웹 서비스

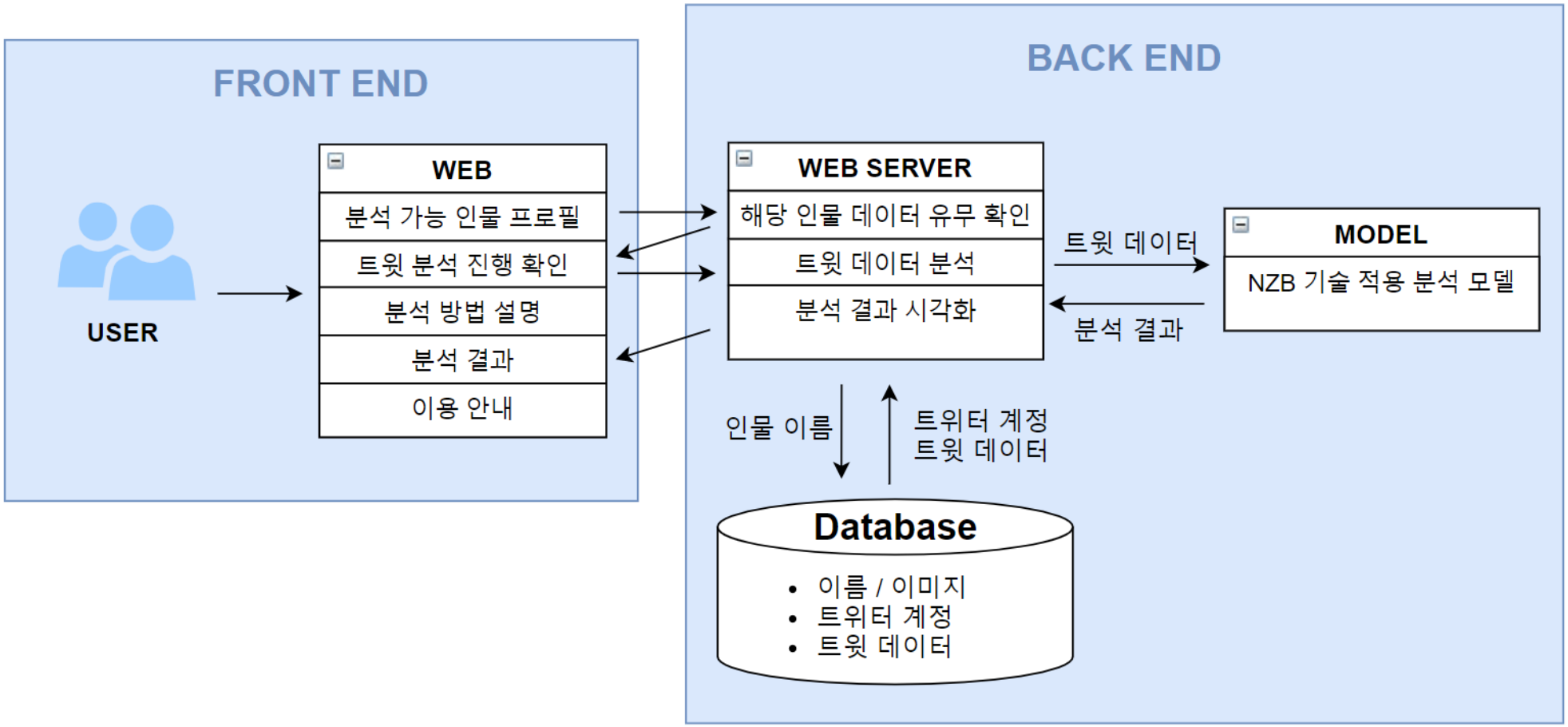
분석 결과 페이지

- 분석 인물의 프로필 및 트윗 바로가기
- 트윗 문장의 감성 비율 그래프 (긍정 / 부정 / 중립)
- 긍정 / 부정 / 중립 문장 예시 (분석 결과 정확도 높은 순으로 2문장 제시)
- 자주 사용하는 단어 5개
- 외부 사이트 공유 버튼



문제 해결

시스템 구조


FRONT END
USER
WEB
분석 가능 인물 프로필
트윗 분석 진행 확인
분석 방법 설명
분석 결과
이용 안내
BACK END
WEB SERVER
해당 인물 데이터 유무 확인
트윗 데이터 분석
분석 결과 시각화
트윗 데이터
분석 결과
MODEL
NZB 기술 적용 분석 모델
인물 이름
트위터 계정
트윗 데이터
Database
이름 / 이미지
트위터 계정
트윗 데이터

문제 해결

개발 환경

01. 언어

Python, JavaScript

02. 프레임워크

Flask

03. 네트워크

AWS

04. 플랫폼

Anaconda, Kaggle, Node.js

05. 라이브러리

Bootstrap, D3, Scikit learn,
Keras, TensorFlow, emoji

06. Editor

Pycharm, Jupyter Notebook,
VS Code, Colab

향후 일정

팀원 역할

01. 수민	02. 주희	03. 아현	04. 승주
PM 및 문서 작성	회의록 작성	Text-CNN 모델 구현	PPT 제작 및 대본 작성
분석 가능한 유명인 조사	분석 가능한 유명인 조사	학습 데이터 조사 및 수집	워드 임베딩 진행
데이터 수집 및 전처리	데이터 수집 및 전처리	모델 학습	학습 데이터 조사 및 수집
학습 데이터 조사 및 수집	학습 데이터 조사 및 수집	NZB 인덱싱 기술 자료조사	모델 학습
분석 데이터 전처리	이모지 테이터셋 제작	NZB 인덱싱 기술 적용	웹서버 설계 및 구축
문장 감성분석 결과 제공	분석 데이터 전처리	모델 평가 및 검증	데이터베이스 구축
자주 사용하는 단어 카운트	문장 감성분석 결과 제공		분석 결과 시각화
NZB 인덱싱 기술 적용	NZB 인덱싱 기술 적용		
모델 평가 및 검증	모델 평가 및 검증		

개발 추진 계획


일정
9 월
10 월
11 월
12 월
2
3
4
1
2
3
4
1
2
3
4
1
프로젝트 기획
요구사항 분석 & 개발환경 구축
데이터 수집 및 전처리
웹 & UI 설계
text-CNN 모델 구현
text-CNN 모델 평가 및 검증
NZB 기술 적용
웹 & UI 구현
데이터베이스 구축
NZB 적용 모델 평가 및 검증
시각화
모델 성능 비교
최종 문서 작성 & 발표

성능 향상

프루닝

```
1 weight = model.get_weights()
2 weight[1]
```

```
array([[[-0.08483198,  0.10879445, -0.0392587 , ..., -0.06532168,
         -0.01460742,  0.10556462],
        [-0.08496561, -0.00843406,  0.01690301, ..., -0.07297008,
          0.08761676, -0.12777297],
        [ 0.04901376, -0.05214268, -0.07659314, ...,  0.00484493,
         -0.02682856,  0.00608644],
        ...,
        [ 0.0069919 , -0.01444536, -0.05929992, ..., -0.03798576,
          0.04171865, -0.17185484],
        [-0.00214236,  0.03407577,  0.0150112 , ..., -0.0470692 ,
          0.01037525,  0.14364514],
        [ 0.04413214, -0.01114238,  0.07179853, ..., -0.0252963 ,
          0.02178209,  0.05814373]],

       [[-0.02932109,  0.03590417, -0.05922792, ...,  0.07371072,
          0.05312107, -0.00901164],
        [-0.05780687, -0.07679348, -0.05774446, ...,  0.03626138,
         -0.05005791,  0.01616147],
        [ 0.0739404 ,  0.07748745,  0.01142946, ...,  0.05761244,
          0.0391749 , -0.11382985],
        ...,
        [ 0.01689709, -0.11136347, -0.00276525, ...,  0.07495484,
         -0.03052371, -0.04141993],
        [-0.05014974,  0.24354735,  0.01009412, ..., -0.05476376,
          0.01429985,  0.19797589],
        [ 0.06921268,  0.09666787,  0.0505678 , ..., -0.03344753,
         -0.03850385,  0.10422182]],
```

80% pruning 결과

```
1 weight = model.get_weights()
2 weight[1]
```

```
array([[[-0.08490542,  0.13946635,  0.        , ..., -0.        ,
          0.        ,  0.14509021],
        [-0.08522931,  0.        ,  0.        , ..., -0.        ,
          0.10017765, -0.18178467],
        [-0.        , -0.        , -0.08196158, ..., -0.        ,
          0.        , -0.        ],
        ...,
        [ 0.        , -0.        ,  0.        , ...,  0.        ,
         -0.        , -0.19687262],
        [ 0.        ,  0.        , -0.        , ..., -0.        ,
         -0.        ,  0.18011148],
        [-0.        ,  0.        , -0.        , ...,  0.        ,
         -0.        ,  0.        ]],

       [[-0.        ,  0.        , -0.        , ...,  0.07085259,
          0.        , -0.        ],
        [ 0.        , -0.16063233,  0.        , ..., -0.        ,
         -0.        ,  0.        ],
        [ 0.0741232 ,  0.07681461, -0.        , ..., -0.        ,
         -0.        , -0.13656458],
        ...,
        [ 0.        , -0.14292516,  0.        , ...,  0.07939034,
         -0.        ,  0.        ],
        [-0.        ,  0.25387815, -0.        , ..., -0.        ,
         -0.        ,  0.21618742],
        [-0.        ,  0.12369737, -0.        , ...,  0.        ,
         -0.        ,  0.10505404]],

       [[-0.        , -0.        ,  0.05303088, ...,  0.07229692,
          0.        , -0.        ],
        [ 0.        , -0.        ,  0.        , ..., -0.        ,
         -0.        ,  0.        ],
        [ 0.        , -0.        , -0.        , ..., -0.        ,
          0.        ,  0.11715095],
```

Pruning(%)	X	70%	75%	80%	85%
fit(epochs=8)	79.08%	81.09%	81.01%	80.94%	80.85%
evaluate()	78.48%	78.37%	78.76%	78.89%	78.81%

연산처리 속도 향상과

evaluate 결과값 높은 모델 선택

Prunned 80 모델 사용

성능 비교

수행 시간

@elonmusk 분석 결과

```
트윗 문장 감정 비율
POSITIVE    51.66%
NEGATIVE    24.32%
NEUTRAL     24.02%
Name: label, dtype: object
                            text     label     score  elapsed_time
389              Congratulations!  POSITIVE  0.970972      0.046137
303  Congratulations SpaceX Team!  POSITIVE  0.969335      0.043981
434             Glad Jen is safe!  POSITIVE  0.960235      0.047925
                                                  text  ... elapsed_time
91   Mild sniffles & cough & slight fever past few ...  ...     0.036287
509                  Too bad that place got torn down.  ...     0.050213
380                                                 😢  ...     0.047230

[3 rows x 4 columns]
                                                  text  ...       cal
213                  Even the small details matter.  ...  0.001797
425  Will release order configurator probably in Jan  ...  0.001939
378                                  You never know  ...  0.002596

[3 rows x 5 columns]

자주 사용하는 단어 TOP5
[('test', 30), ('product', 27), ('tesla', 26), ('great', 26), ('need', 23)]

소요시간 32.002538204193115
```

NZB Indexing 적용 전

Layer 1 약 32.6초

```
자주 사용하는 단어 TOP5
[('test', 30), ('product', 27), ('tesla', 26), ('great', 26), ('need', 23)]

소요시간 32.600881814956665
```

Layer 1,2 약 27초

```
자주 사용하는 단어 TOP5
[('test', 30), ('product', 27), ('tesla', 26), ('great', 26), ('need', 23)]

소요시간 27.096725463867188
```

Layer 1,2,3 약 26.7초

```
자주 사용하는 단어 TOP5
[('test', 30), ('product', 27), ('tesla', 26), ('great', 26), ('need', 23)]

소요시간 26.729569911956787
```

NZB Indexing 적용 후

NZB Indexing 적용을 통해 convolution 수행 시간이 줄어들었다.

성능 비교

메모리

@elonmusk 분석 결과

```
 트윗 문장 감정 비율
POSITIVE   51.66%
NEGATIVE   24.32%
NEUTRAL    24.02%
Name: label, dtype: object
                          text     label     score  elapsed_time
389             Congratulations!  POSITIVE  0.970972      0.051141
303  Congratulations SpaceX Team!  POSITIVE  0.969335      0.045336
434            Glad Jen is safe!  POSITIVE  0.960235      0.043845
                                                 text  ... elapsed_time
91   Mild sniffles & cough & slight fever past few ...  ...     0.045404
509                 Too bad that place got torn down.  ...     0.045176
380                                                 😟  ...     0.052922

[3 rows x 4 columns]
                                               text  ...       cal
213                  Even the small details matter.  ...  0.001797
425  Will release order configurator probably in Jan  ...  0.001939
378                                 You never know  ...  0.002596

[3 rows x 5 columns]

자주 사용하는 단어 TOP5
[('test', 30), ('product', 27), ('tesla', 26), ('great', 26), ('need', 23)]

cpu usage              : 53.90.0 %
memory usage           : 0.61 %
```

NZB Indexing 적용 전

Layer 1 cpu: 30.4 / memory : 0.69 (%)

```
자주 사용하는 단어 TOP5
[('test', 30), ('product', 27), ('tesla', 26), ('great', 26), ('need', 23)]

cpu usage              : 30.4 %
memory usage           : 0.69 %
```

Layer 1,2 cpu: 44.7 / memory : 0.66 (%)

```
자주 사용하는 단어 TOP5
[('test', 30), ('product', 27), ('tesla', 26), ('great', 26), ('need', 23)]

cpu usage              : 44.7 %
memory usage           : 0.66 %
```

Layer 1,2,3 cpu: 56.4 / memory : 0.66(%)

```
자주 사용하는 단어 TOP5
[('test', 30), ('product', 27), ('tesla', 26), ('great', 26), ('need', 23)]

cpu usage              : 56.4 %
memory usage           : 0.66 %
```

NZB Indexing 적용 후

NZB Indexing 적용을 통해 CPU 사용량, Memory 사용량이 전반적으로 증가했다.

프로그램 시연

capstoneteam2.tk