

0409 진행상황

1. 지난주 이슈

- Text to Vector
 - CountVectorizer 클래스로 BoW 만듦
 - 패딩 처리
 - max_len 기준으로 `pad_sequences` 처리
- 부족한 경우는 0 패딩으로 같은 len을 가지는 시퀀스로 최종적으로 처

```
X_train
executed in 10ms, finished 13:59:59 2021-04-02
array([[ 0,  0,  0, ...,  67,   6, 248],
       [ 0,  0,  0, ..., 1439,  23,  22],
       [ 0,  0,  0, ...,  426, 136,   3],
       ...,
       [ 0,  0,  0, ...,  136,  77, 2806],
       [ 0,  0,  0, ...,  212,   2,  212],
       [ 0,  0,  0, ...,  239,   1,   10]])
```

모델 구축

```
model = Sequential()
model.add(Embedding(vocab_size, 50))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Bidirectional(LSTM(50)))
model.add(Dense(3, activation='softmax'))
```

- test 정확도 = 0.5041

1. 모델

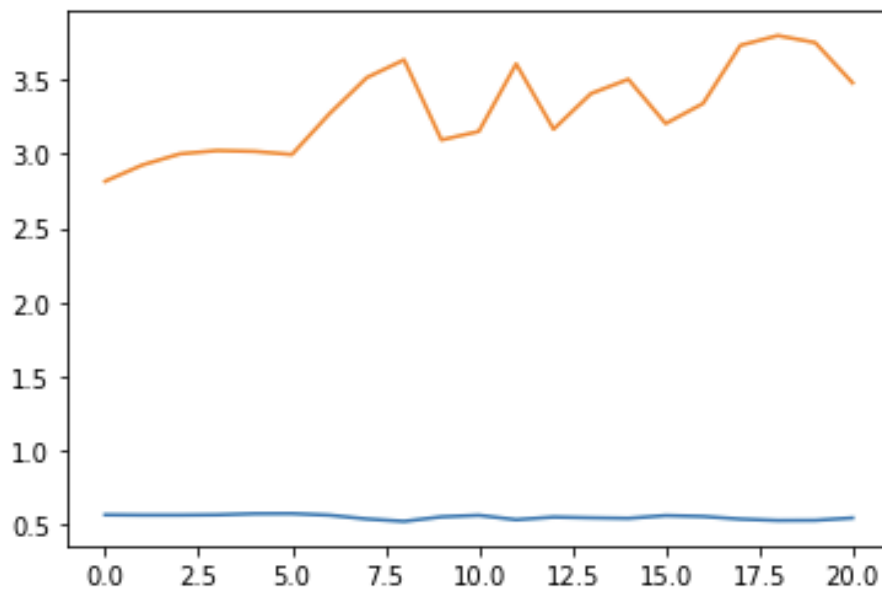
```
model = Sequential()
model.add(Embedding(vocab_size, 100))
model.add(BatchNormalization())
```

```

model.add(Dropout(0.5))
model.add(Bidirectional(LSTM(50, return_sequences=True)))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Bidirectional(LSTM(50, return_sequences=True)))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Bidirectional(LSTM(50)))
model.add(Dense(3, activation='softmax'))

```

- 테스트 정확도: 0.5638



2. 이번주 진행상황

1. Input 값 : Count 기반의 vector

패딩 후, 정규화 진행한 후 모델 수행하였지만 여전히 성능이 안 좋은 상황

- 모델 1: 은닉층 2개를 가지는 LSTM 구조

epoch: 100

Valid 정확도 : 0.43713 (train 데이터도 여전히 개선이 없는 상태)

- 모델2 : 1D-CNN

epoch : 100

Valid 정확도 : 0.43713 (train 데이터도 여전히 개선이 없는 상태)

2. Input 값: Word2Vec vector

- AI허브 데이터 6백만개 ⇒ word2vec 모델 학습
- 단어 당 100차원의 벡터가 부여됨
- 실제 train set을 토큰화 한 후, count 기반이 아닌 word2vec vector로 변환
- 이 때, 부여되지 않은 값 : 랜덤으로 작은 값으로 100차원 행렬 생성
- 한 문장에 존재하는 단어 행렬별로 더한 후, 갯수로 나누어 문장의 100차원 평균으로 input 값 지정
 - Input값 예시

```
sum(X_train[0]) / len(X_train[0])
executed in 11ms, finished 13:45:41 2021-04-09
array([ 1.27485409e-01,  8.23980290e-03, -1.32144257e-01, -1.31051645e-01,
        1.61492750e-01, -3.98410372e-02,  1.99769754e-02, -4.08526987e-01,
        2.38573626e-01,  9.18339863e-02, -7.69471228e-02,  3.01503632e-02,
        2.76218038e-02,  6.80007413e-02,  1.05558686e-01,  2.79601458e-02,
       -8.63816813e-02, -2.51677409e-02,  1.18635066e-01,  1.69561043e-01,
        6.05299696e-03, -1.78437233e-01, -6.79993853e-02,  3.51231880e-02,
        4.09121662e-02, -3.29676509e-01,  2.06857994e-01,  3.85349631e-01,
       -1.98827043e-01,  1.55974030e-01, -1.16328858e-01,  7.96150640e-02,
       -1.28055438e-01, -1.70631111e-01,  2.72211749e-02,  2.90114284e-01,
        6.96500242e-02,  2.44633555e-01,  8.81979465e-02,  1.34876311e-01,
        6.46285340e-02, -2.83093359e-02, -3.43858689e-01, -5.80019243e-02,
        2.45623246e-01,  3.07061791e-01,  4.39386368e-02,  3.40989679e-01,
        7.81106651e-02,  8.66883099e-02,  2.04399362e-01, -1.61611233e-02,
        8.27188045e-02,  1.59497008e-01, -9.37300548e-02,  2.63493598e-01,
        3.09644520e-01,  2.40241602e-01, -1.79686740e-01,  2.72785723e-02,
       -7.55629465e-02, -6.55498952e-02, -1.41496226e-01,  1.47061411e-03,
        1.80223718e-01,  2.79807043e-03,  9.50599741e-03,  2.46717647e-01,
       -1.36658192e-01, -5.49307652e-02, -1.31417960e-01,  4.01656419e-01,
       -7.99595937e-02,  1.23989535e-02, -4.20702696e-02,  6.97149485e-02,
        8.07519481e-02,  1.48915976e-01, -1.37721992e-03, -1.02999859e-01,
        3.73630315e-01,  3.37255448e-01,  2.11999938e-01, -1.64211765e-01,
       -2.69827042e-02, -1.00057699e-01,  1.80788934e-01, -1.30664065e-01,
       -1.79420948e-01, -2.55546778e-01, -1.58899933e-01,  1.60253868e-01,
        3.89023498e-02, -1.30451024e-02,  2.51350105e-01, -3.54798511e-04,
       -4.60654497e-02, -3.30065221e-01,  2.78480500e-01, -1.45261541e-01],
      dtype=float32)
```



















- 모델 성능

- 여전히 매우 낮은 성능

```
Epoch 66/100: val_acc did not improve from 0.45000
2/2 [=====] - 6s 3s/step - loss: 1.0901 - acc: 0.4000 - val_loss: 1.0765 - val_acc: 0.4500
Epoch 66/100
2/2 [=====] - ETA: 0s - loss: 1.0868 - acc: 0.4125
Epoch 00066: val_acc did not improve from 0.45000
2/2 [=====] - 7s 3s/step - loss: 1.0868 - acc: 0.4125 - val_loss: 1.0766 - val_acc: 0.4500
Epoch 67/100
1/2 [=====>.....] - ETA: 0s - loss: 1.0879 - acc: 0.4531
Epoch 00067: val_acc did not improve from 0.45000
2/2 [=====] - 7s 3s/step - loss: 1.0804 - acc: 0.4500 - val_loss: 1.0765 - val_acc: 0.4500
Epoch 68/100
2/2 [=====] - ETA: 0s - loss: 1.0886 - acc: 0.4125
Epoch 00068: val_acc did not improve from 0.45000
2/2 [=====] - 6s 3s/step - loss: 1.0886 - acc: 0.4125 - val_loss: 1.0764 - val_acc: 0.4500
Epoch 69/100
1/2 [=====>.....] - ETA: 0s - loss: 1.0878 - acc: 0.4219
Epoch 00069: val_acc did not improve from 0.45000
2/2 [=====] - 7s 3s/step - loss: 1.0831 - acc: 0.4375 - val_loss: 1.0763 - val_acc: 0.4500
Epoch 70/100
1/2 [=====>.....] - ETA: 0s - loss: 1.0947 - acc: 0.3906
Epoch 00070: val_acc did not improve from 0.45000
2/2 [=====] - 6s 3s/step - loss: 1.0813 - acc: 0.4125 - val_loss: 1.0760 - val_acc: 0.4500
Epoch 71/100
2/2 [=====] - ETA: 0s - loss: 1.0966 - acc: 0.3875
Epoch 00071: val_acc did not improve from 0.45000
2/2 [=====] - 6s 3s/step - loss: 1.0966 - acc: 0.3875 - val_loss: 1.0755 - val_acc: 0.4500
Epoch 72/100
1/2 [=====>.....] - ETA: 0s - loss: 1.0928 - acc: 0.4375
Epoch 00072: val_acc did not improve from 0.45000
2/2 [=====] - 6s 3s/step - loss: 1.1051 - acc: 0.4125 - val_loss: 1.0753 - val_acc: 0.4500
```

3. 논의

- 모델의 정확도를 향상시키기 위한 방법
- 현재 존재하지 않은 word2vec 의 행렬을 랜덤으로 부여하는게 맞는지 & input값 평균 적절할지
- 데이터 자체의 문제가 아닐까 (현재 이 데이터로 캐글에 공개되어있는 competition이며 score는 f1-score 기준)

#	Team Name	Notebook	Team Members	Score ?	Entries	Last
1	K Lee			0.61187	14	13d
2	sanl			0.60815	44	4mo
3	eunjin kim			0.60784	7	4mo
4	Kyeongpil Kang			0.60507	29	19h
5	Ji Hyung Moon			0.60419	1	8mo
6	kakao brain			0.60394	2	8mo
7	noowad93			0.59884	4	13h
8	Yeoun Yi			0.59716	13	8mo
9	Giantpanda			0.58508	1	4mo
10	ohjuhyun			0.58214	14	8mo
11	Naive Bayeji		   	0.57696	30	3mo
12	WonChul Kim			0.56991	3	4mo
13	SY			0.56655	5	8mo
14	SungheeJung			0.56530	10	10mo
15	Minyoung Park			0.56106	20	4mo

TODO

- word2vec 인풋값 개선 (존재하지 않는 값 0으로 대체?)
- 모델 개선
- 데이터 자체의 문제인지 파악 (offensive와 hate의 차이를 실제로 사람이 라벨링하면 주관적인 감정으로 데이터가 편향이 생길 수도 있지 않을까)