

0402 진행상황

1. 자연어처리 공부

■ 희소 표현(Sparse Representation)

단어를 벡터로 나타내는 가장 기본적인 방법이 One hot encoding 방법으로 나온 벡터 표현 방법이다. 즉, 표현하고자 하는 단어의 인덱스의 값만 1이고 나머지 인덱스는 모두 0으로 표현되는 방법이다. 그러나 이러한 경우 해당 단어의 인덱스를 제외하고 다른 단어는 모두 0으로 표현되어 Sparse Representation이 된다. 따라서 단어 간의 유사도도 파악하지 못함

즉, 단어가 10000개가 있다면 10000개의 차원의 벡터가 나오게 되고 너무 많은 공간을 차지하게 된다.

■ 밀집 표현(Dense Representation)

따라서, Dense Representation이 등장하게 되고 10000차원을 원하는 벡터로 밀집하여 나타내는 것을 뜻한다. 즉, 희소 행렬을 차원 감소를 통해 중요한 축을 찾아내어 더 적은 차원으로 다시 표현하는 것이다. 이처럼 단어를 밀집 벡터(Dense Vector)로 표현하는 것이 워드 임베딩이다.

■ 분산 표현(Distributed Representaion)

- 기본적으로 분포 가설이라는 가정 하에 만들어진 표현 방법
- 비슷한 위치에서 등장하는 단어들은 비슷한 의미를 가진다는 가정

대표적인 워드 임베딩 방법론 : LSA, Word2Vec, FastText, Glove 등

■ 케라스 임베딩 층(Keras Embedding Layer)

- 임베딩 층의 입력으로 사용되기 위해서는 입력 시퀀스의 각 단어들이 모두 정수 인코딩이 되어 있어야 한다.
- 임베딩 층 역할 : 입력 정수에 대해 밀집 벡터로 맵핑 ⇒ 밀집 벡터는 인공 신경망의 학습 과정에서 가중치가 학습되는 것과 같은 방식으로 훈련됨
- Embedding(vocab_size, output_dim, input_length)

1. Vocab_size : 텍스트 데이터의 전체 단어 집합의 크기

2. Output_dim : 워드 임베딩 후의 임베딩 벡터의 차원

3. input_length : 입력 시퀀스의 길이 (샘플의 길이가 500개의 단어로 구성 ÷ 값 : 500)

■ 사전훈련된 워드 임베딩

- 훈련 데이터가 적은 상황이라면, 케라스 Embedding()보다 더 효과적일 수 있음
- 훈련 데이터가 적다면 케라스의 Embedding()으로 해당 문제에 충분히 특화된 임베딩 벡터를 만들어내는 것이 쉽지 않음
⇒ 등장 빈도순대로 단어를 정렬하여 인덱스를 부여하였을 때의 장점은 등장 빈도수가 적은 단어의 제거

2. 구체적인 진행상황

데이터 전처리

- tokenized 변수 생성 (mecab 형태소 라이브러리 사용)

	comments	hate	tokenized
0	(현재 호텔주인 심정) 아18 난 마른하늘에 날벼락맞고 호텔망하게생겼는데 누군 계속...	2	[현재, 호텔, 주인, 심정, 아, 18, 난, 마른, 하늘, 날벼락, 맞, 호텔, ...]
1	...한국적인 미인의 대표적인 분...너무나 곱고아름다운모습...그모습위의 슬픔을...	0	[..., 한국, 적, 미인, 대표, 적, 분, ,, ..., 너무나, 곱, 아름다...
2	...못된 녀들...남의 고통을 즐겼던 녀들..이젠 마땅한 처벌을 받아야지...그래...	2	[..., 못한, 녀, ,, ..., 남, 고통, 즐겼, 던, 녀, ,, ..., 이젠, ...]
3	1,2화 어설프는데 3,4화 지나서부터는 갈수록 너무 재밌던데	0	[1, ,, 2, 화, 어설프, 는데, 3, ,, 4, 화, 지나, 서, 부터, 갈...
4	1. 사람 얼굴 손톱으로 긁은것은 인격살해이고2. 동영상에 몰카냐? 메갈리안들 생각...	2	[1, ,, 사람, 얼굴, 손톱, 으로, 긁, 것, 인격, 살해, 2, ,, 동영상...
...
7891	힘내세요~ 응원합니다!!	0	[힘내, 세요, ~, 응원, 합니다, !, !]
7892	힘내세요~~삼가 고인의 명복을 빕니다..	0	[힘내, 세요, ~~, 삼가, 고인, 명복, 빕, 니다, ,, ,]
7893	힘내세용 ^^ 항상 응원합니다 ^^!	0	[힘내, 세용, ^^, 항상, 응, 원함, 니, 닷, ^^, !]
7894	힘내소...연기로 답해요 나도 53살 인데 이런일 저런일 다 있더라구요.인격을 믿습...	0	[힘내, 소, ,, ..., 연기, 로, 답, 해요, ,, 나, 53, 살, 인데, ...]
7895	힘들면 관둬야지 그게 현명한거다	0	[힘들, 관둬, 어야지, 그게, 현명, 한, 거]

7896 rows x 3 columns

- 종속변수 원핫인코딩 (softmax 함수로 확률값 반환해주기 위해)

```

4 from tensorflow.keras.utils import to_categorical
dataset_y = to_categorical(train_data['hate'])
dataset_y = np.array(dataset_y, dtype=np.int32)
print(dataset_y)

```

executed in 7ms, finished 02:42:47 2021-04-02

```

[[1 0 0]
 [0 0 1]
 [1 0 0]
 ...
 [1 0 0]
 [1 0 0]
 [0 1 0]]

```

텍스트 to 시퀀스

1. 단어 기반 인코딩 (빈도수 기반)

- `from tensorflow.keras.preprocessing.text import Tokenizer` 사용

2. 시퀀스로 변환

- `tokenizer.texts_to_sequences` 를 통해 시퀀스로 변환

3. 패딩 처리

- `max_len` 기준으로 `pad_sequences` 처리

부족한 경우는 0 패딩으로 같은 len을 가지는 시퀀스로 최종적으로 처리

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

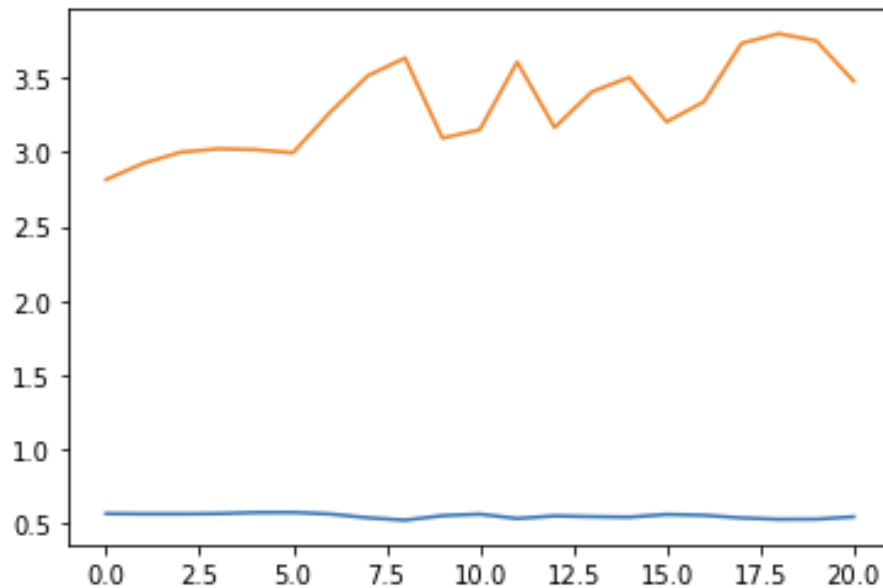
1. 모델

```

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



논의

- 모델의 정확도를 향상시키기 위한 방법
- 데이터 자체의 문제가 아닐까

#	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 Bayesji		   	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