**딥러닝과 인공지능 4차과제**

20152640

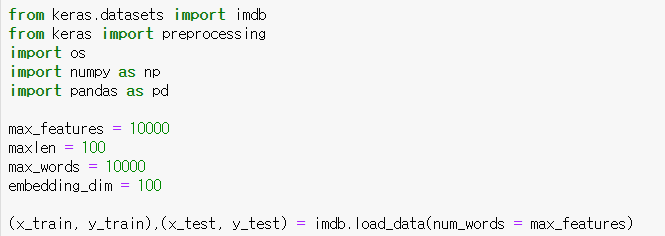
김태현

이번 딥러닝과 인공지능 4차과제의 내용은 단어 Embedding을 활용하여 Positive와 negative를 분류하는 것이다. 해당 분류를 위해 사용하는 데이터는 keras에 내장되어 있는 데이터인 IMDB 영화 리뷰 데이터를 활용하고자 한다. 이번 보고서에서 집중적으로 살펴볼 사항은 크게 2가지가 있고 세부적으로 2가지 비교 실험을 해보고자 한다. 이를 정리해 보면 다음과 같다.

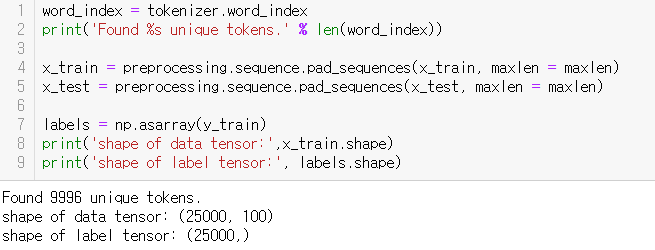
1. IMDB 데이터를 load했을 때 나오는 그대로 활용했을 때
   1. word\_index를 로드하고 embedding 작업을 진행 분류모델 만들기
   2. Glove를 활용 pre-train된 정보를 가져와 분류모델 만들기
2. IMDB 데이터를 text로 변환시킨 후 Tokenizing 진행 후 활용하기
   1. Embedding을 실행한 후 분류모델 만들기
   2. Glove를 활용 pre-train된 정보를 활용 분류모델 만들기

분석에 활용한 데이터는 train 25000개, test 25000개로 이루어져 있으며 각각 데이터는 긍정적 리뷰와 부정적 리뷰로 분리되어 있는 데이터이다.

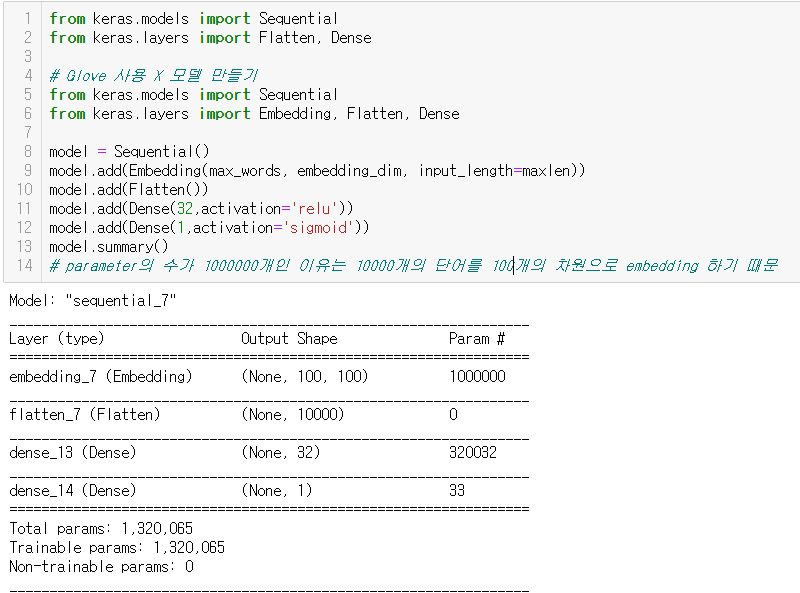
IMDB 데이터 load 및 활용할 패키지 load하기



Imdb 데이터의 word에 대한 정보 가져오기 및 문장길이 padding 작업하기.

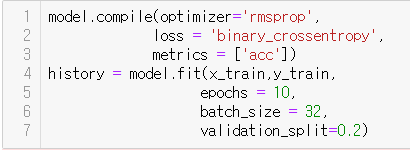


**1.1** 리뷰 분류모델 구성하기

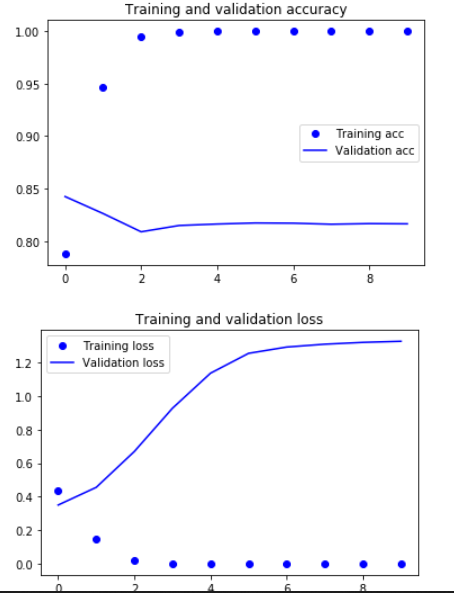


train되는 parameter수는 total parameter수와 동일함.

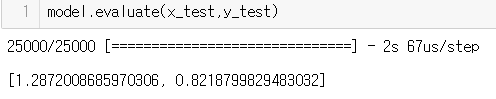
모델 학습시키기.



모델 성능확인하기.

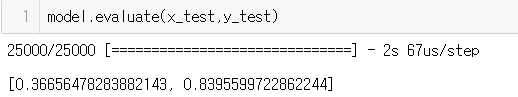


1epoch이후 accuracy는 물론 loss도 상승한다는 것을 알 수 있음. 10epcoch 학습 성능



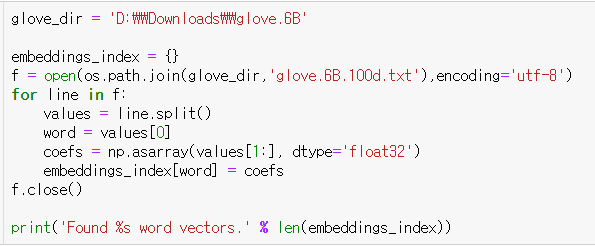
Loss는 1.2872 accuracy는 0.82188이다.

1epoch동안 학습시킨 모델의 성능

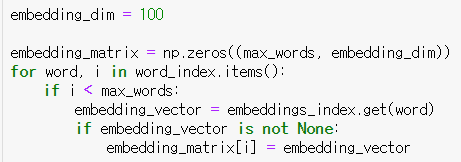


1epoch만 학습시켰을 때 loss는 0.3666 accuracy는 0.8396이다. 오히려 학습을 적게 시켰을 때 성능이 더 좋은 것을 알 수 있음 이러한 상황이 발생한다면 학습의 learning rate의 비율을 줄이면 된다는 사실을 저번 과제를 통해 배울 수 있었다. 이번 과제에서는 중요한 사항이 아니기 때문에 생략할 것이다.

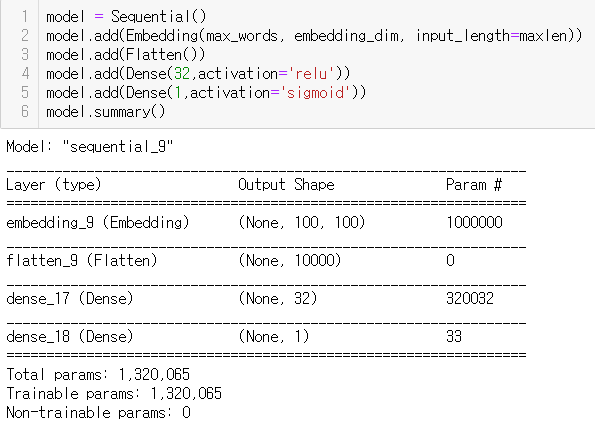
**1.2** Glove 모델불러와 embedding layer에 해당 값 사용하기.



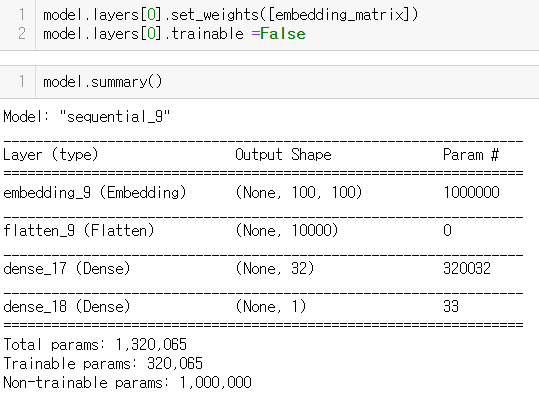
Glove 모델에 있는 값 중 imdb에 있는 데이터만을 가져와 embedding matrix만들기



학습을 위한 모델 구성하기

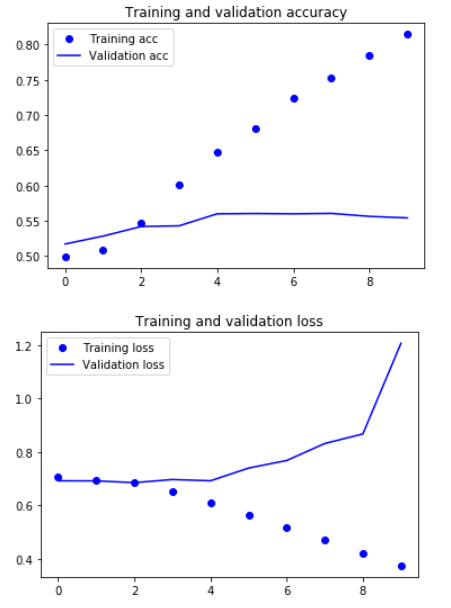
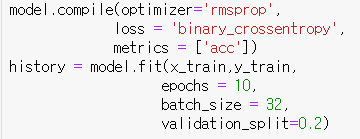


Glove 모델의 embedding 결과값 입력하고 학습 불가능하게 만들기



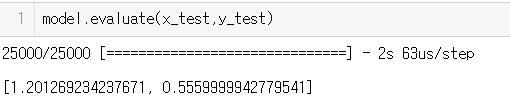
모델에서 Embedding matrix에서 학습되는 100만개의 데이터가 학습되지 않게 되었음을 알 수 있다. 또한 set\_weight함수를 활용 glove에 있는 embedding matrix의 값을 삽입하였음.

모델 compile및 성능 확인하기.

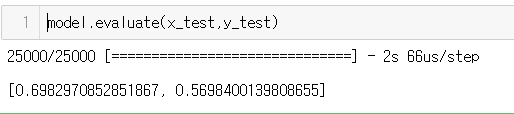


성능을 확인해보면 4epoch 정도에서 성능이 가장 좋았음을 알 수 있음 4epoch까지 실행했을 때의 성능과 10epoch로 학습했을 때 성능을 비교해 보겠음.

10 epochs

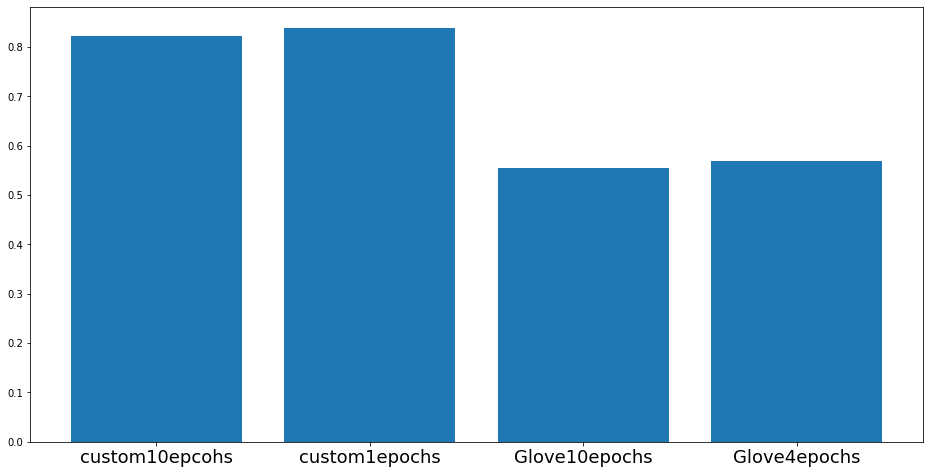


4epochs



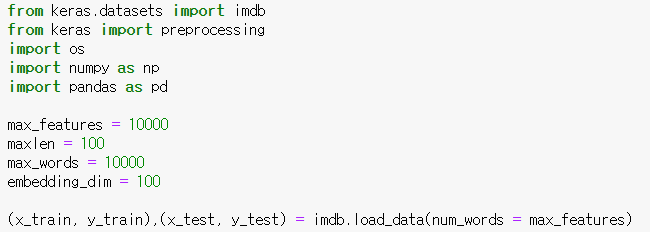
4epcoch일 때 성능과 Loss 모두 우수함을 알 수 있다. 이는 학습이 진행될수록 과적합이 일어났음을 알 수 있다.

성능 종합적인 비교

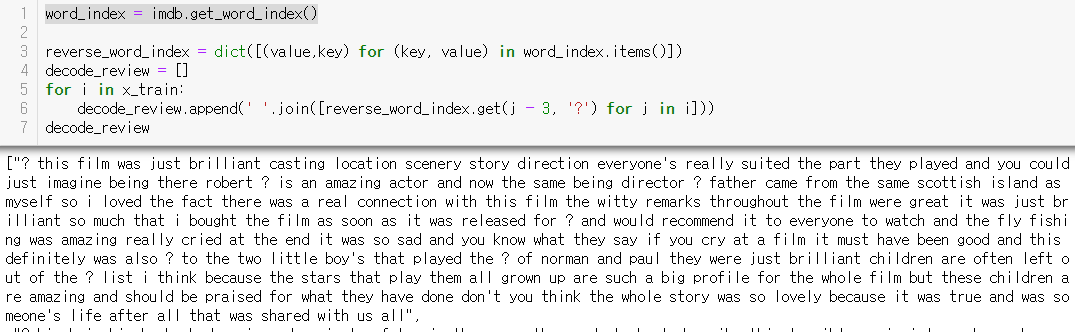


Glove 모델을 사용했을 때 성능이 오히려 더 좋지 않다. 이는 텍스트 분석을 진행함에 있어 미리 Pre Train Model을 사용하면 성능이 않좋게 나올 수 있다는 사실을 보여준다. CNN 모델 같은 경우는 Pretrain 된 모델을 사용하면 성능이 좋게 나오지만 RNN이나 텍스트 분석을 진행할 때에는 이와 같이 목적에 따라 Pretrain 모델을 적절히 선택해 사용하는 것이 중요하다고 생각한다.

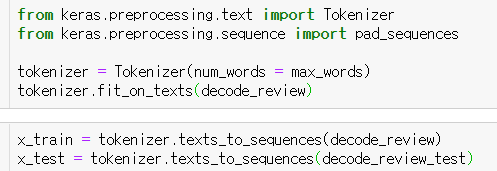
**2** IMDB데이터 텍스트로 변환 후 Tokenizing 진행



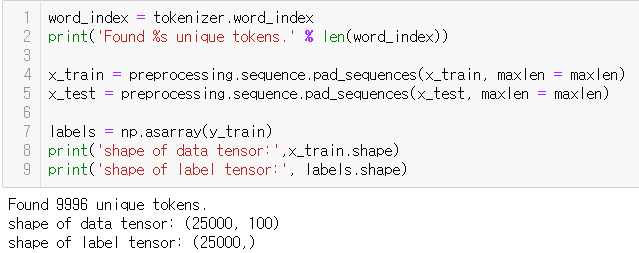
숫자로 나와있던 데이터를 문자형으로 변환



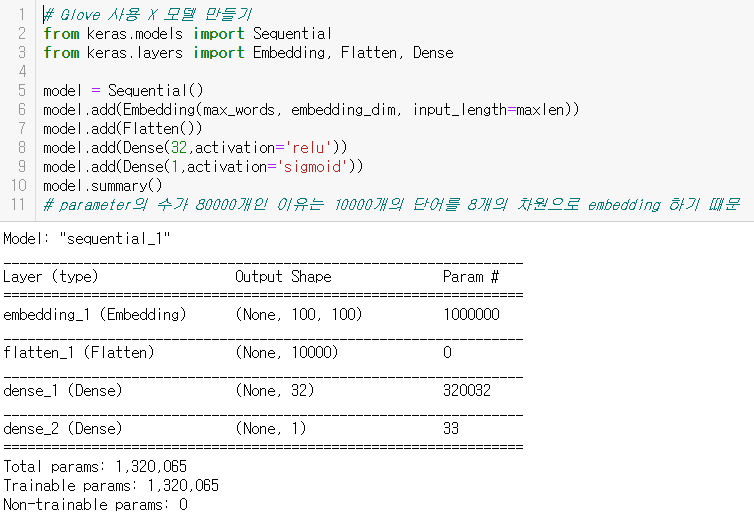
데이터 Tokenizing작업 시행



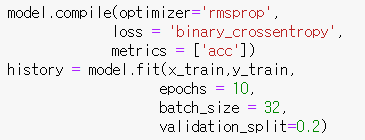
Input data의 개수를 가지게 하기 위한 padding 작업 시행



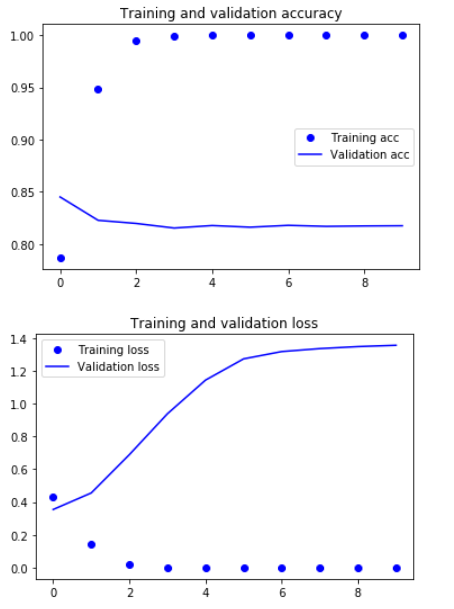
**2-1** Glove X 모델 만들기



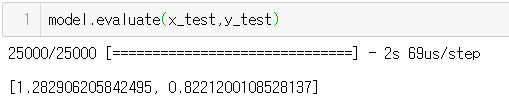
모델 compile 및 학습시키기



모델 결과 확인

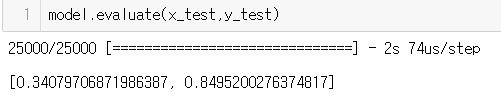


10 epoch일 때 성능 확인



Loss 1.2829, accuracy 0.8221나옴

1epoch 일때 성능 확인

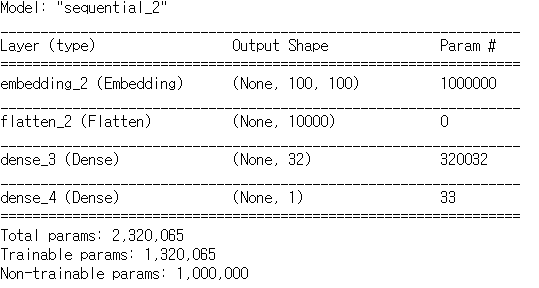


Loss 0.3408, Accuracy 0.8495 나옴 10epoch일 때 과적합 되었음을 알 수 있다. 또한 성능도 우수하다는 사실을 알 수 있음

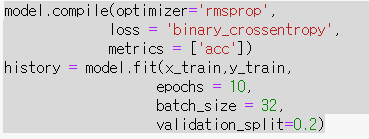
**2-2** Glove 모델 사용

모델을 불러오는 과정 및 적용하는 과정은 1단계에서와 동일한 과정을 거쳤기 때문에 생략

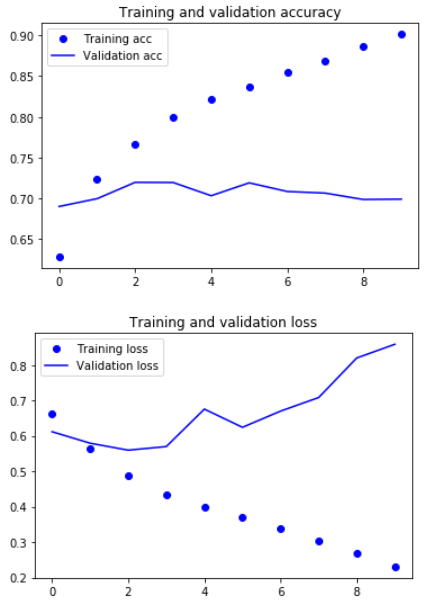
모델 구성



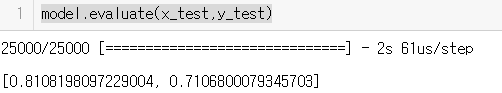
모델 compile 및 학습



모델 성능 확인하기

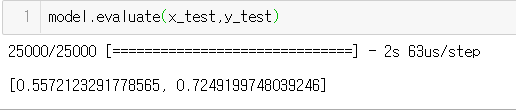


10 epoch일 때 성능 확인



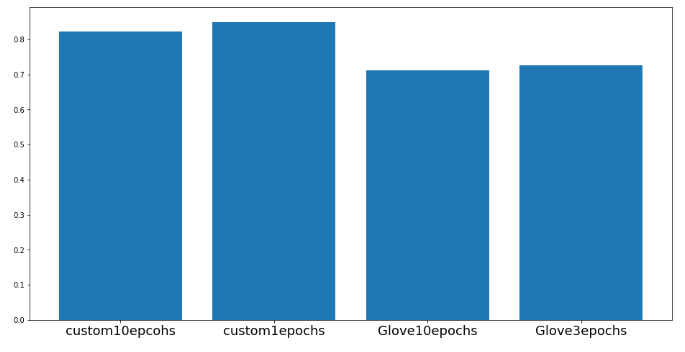
Loss 0.8108 Accuracy 0.7107나옴

3epochs일 때 성능 확인

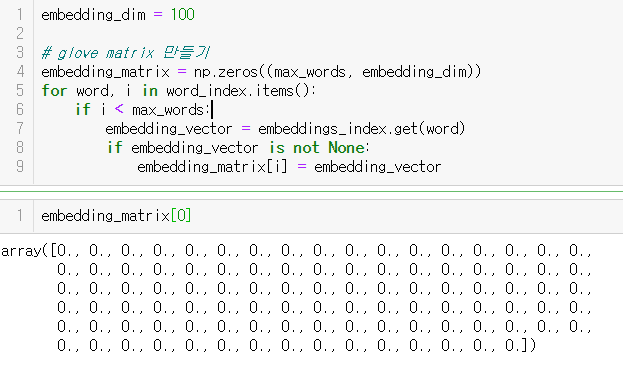


Loss 0.5572, Accuracy 0.7249 나옴 Glove를 사용할 때 역시 과적합이 된다는 사실을 알 수 있다.

Imdb 변환시킨 후 작업한 모델 성능 종합적 확인



기존에 있는 데이터를 그대로 활용했을 때에 비해 성능이 더 좋아짐을 알 수 있음.

결론: Glove모델을 사용하지 않았을 때 imdb 데이터를 변환하던 변환하지 않던 성능이 비슷했지 만 glove 모델을 사용할 때에는 두개의 차이가 확연하게 나타나는 모습을 보았다. 이에 대하여 자세히 살펴보았는데 이유를 알아낼 수 있었다. glove에는 우리가 변환시킨 데이터를 넣었을 때 자신의 단어에 맞추어 변환되는 모습을 보여주었다. 하지만 그냥 했을 때 word\_index에 있는 key값이 Glove에 없는 경우가 발생하였고 이때는 embedding의 값이 0으로 나오는 모습을 볼 수 있었다. 

이처럼 단어의 토큰화 정도가 모델에 영향을 더 미칠 수 있으니 많은 사항들을 고려해 모델을 만드는 것이 중요하다고 생각한다.