**인공지능 프로젝트 보고서**

**161655 허정화**

**1. 프로젝트 주제**

금융뉴스 헤드라인 감정 분석

**2. 기존 연구 소개(접근방법) 및 한계점**

현재 우리 사회는 금융위기에 대한 불안감이 무척 큽니다. 이러한 금융 위기 예측 및 대비를 위해 시장 정서 파악에 대한 다양한 연구가 필요합니다. 즉, 신뢰할 수 있으며 시기적절하게 시장의 정서를 파악할 수 있는 방법이 필요합니다. 기존의 방법으로는 정기적으로 설문조사를 하여 계산된 신뢰도 지표와 같은 방법이 있는데 이는 정보가 무척 제한되며 결과가 시간이 지날수록 빠르게 뒤처진다는 한계가 있습니다. 따라서 설문조사에 기반한 정서 조사의 한계를 극복하기 위해 경제발전을 실시간으로 모니터링할 수 있는 기술에 대한 관심이 높아지는 추세입니다.  
이를 극복하기 위한 가장 유망한 접근법 중 하나는 미디어를 투자자 입장에서 바라보는 것입니다. 월별로 최근 경제 발전에 대한 의견을 묻는 대신 뉴스를 활용하여 시장 정서의 진화를 시각화할 수 있습니다. 투자 정보는 주로 뉴스를 기반으로 하기에 미디어가 투자자의 감정과 시장 참여자의 행동에 영향을 준다고 가정하는 것은 매우 타당합니다.  
따라서 본 프로젝트에서는 발전된 자연어 처리 기술을 이용하여 금융뉴스 헤드라인의 감정을 분류해볼 것입니다.

**3. 사용 데이터 (수집 데이터)**

데이터 이름 : Sentiment Analysis for Financial News

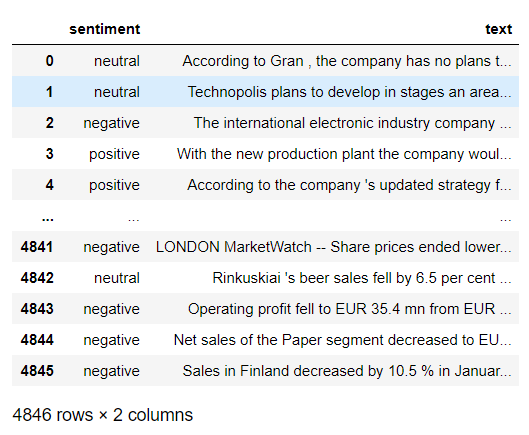
데이터 특징 : 소매 투자자 관점에서 금융 뉴스 헤드라인에 대한 감정이 라벨링 되어있음.

데이터 용도 : 텍스트 감정 분류 학습을 목적으로 사용

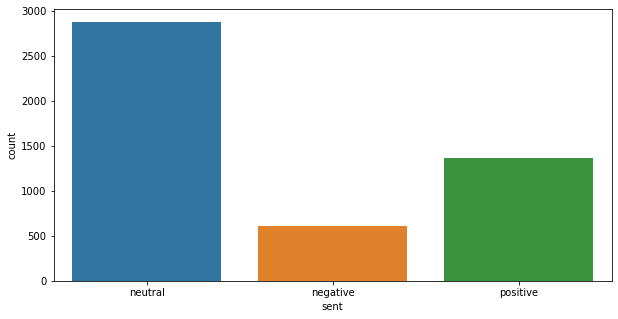
데이터 출처 : kaggle <https://www.kaggle.com/ankurzing/sentiment-analysis-for-financial-news>

참고 : Malo, P., Sinha, A., Takala, P., Korhonen, P. and Wallenius, J. (2014): “Good debt or bad debt: Detecting semantic orientations in economic texts.” Journal of the American Society for Information Science and Technology.

**4. 사용 데이터의 통계 (변수 별 평균, 분산 등) 및 전처리**



**-데이터 분포**

****

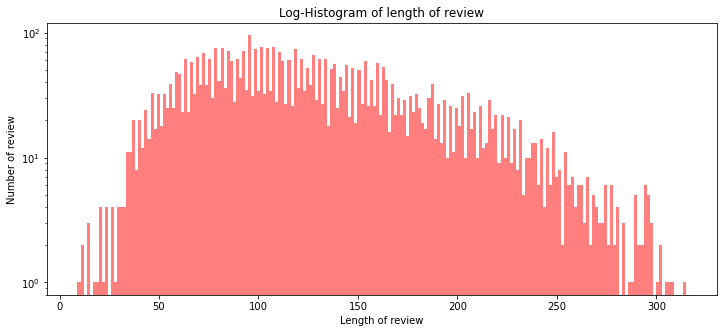
중립 개수: 2879

긍정 개수: 1363

부정 개수: 604

**-Text 길이 분포**

#그래프



길이 최대 값: 315

길이 최소 값: 9

길이 평균 값: 128.13

길이 표준편차: 56.52

길이 중간 값: 119.0

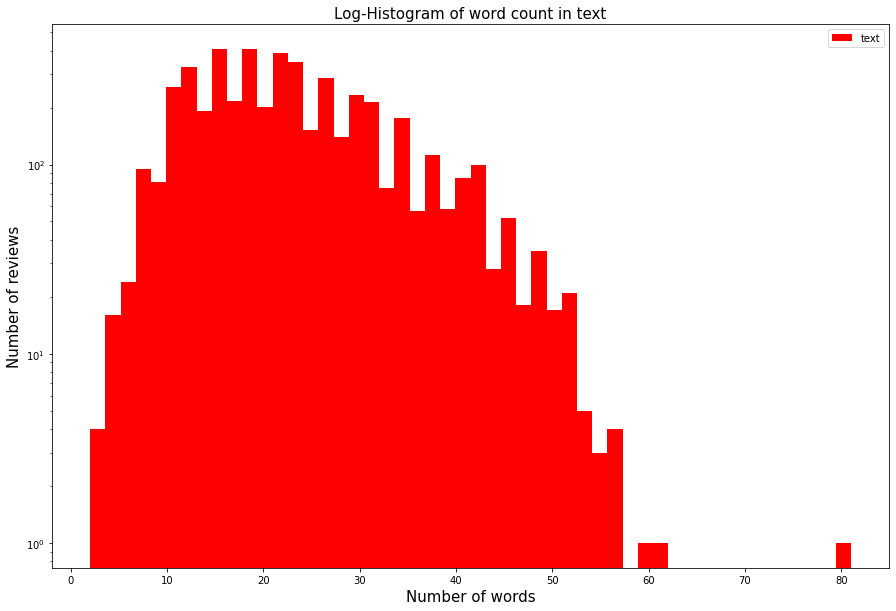
길이 제 1 사분위: 84.0

길이 제 3 사분위: 163.0

**-많이 사용된 단어**

****

**-리뷰 단어 개수 분포**

****단어 개수 최대 값: 81

단어 개수 최소 값: 2

단어 개수 평균 값: 23.11

단어 개수 표준편차: 9.96

단어 개수 중간 값: 21.0

단어 개수 제 1 사분위: 16.0

단어 개수 제 3 사분위: 29.0

**-특수문자 및 대소문자 비율**

물음표: 0.17%

마침표: 99.46%

첫 글자가 대문자: 92.88%

대문자: 99.83%

숫자: 53.32%

**# 학습에 방해되기 때문에 특수문자 제거 및 대문자->소문자 바꾸기가 필요함**

**-전처리**

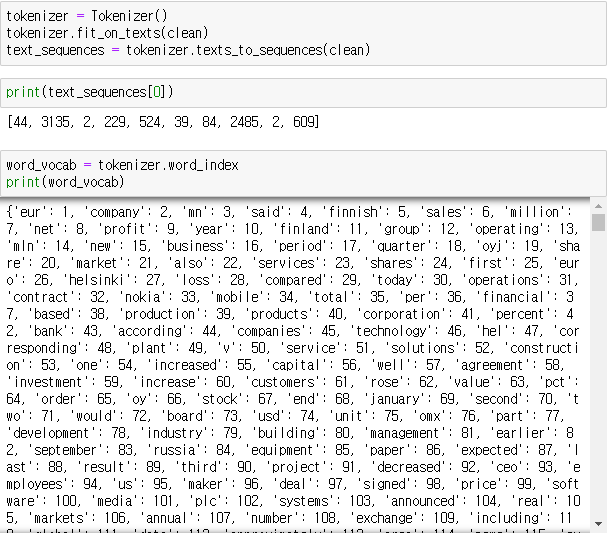


1. 특수문자 공백으로 바꾸기 //특수문자는 문장의 의미에 크게 영향을 미치지 않는다
2. 대문자 -> 소문자 바꾸기 //불용어 제거 및 학습에 방해가 되기 때문에 소문자로 바꾼다.
3. 불용어 제거 //불용어란 문장에서 자주 출현하나 의미에 큰 영향을 주지 않는 데이터이다.

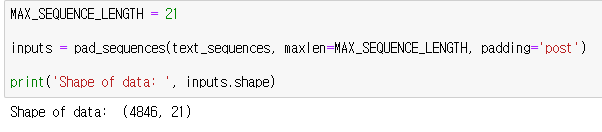


**🡪전처리한 데이터는 csv파일로 저장하여 머신러닝에 사용한다.**

**-벡터화(Tokenizer 사용) :** 각 단어를 인덱스로 벡터화한다.



**-패딩(pad\_sequences 사용) :** 일정한 길이의 입력값을 만들기 위해서 패딩한다.

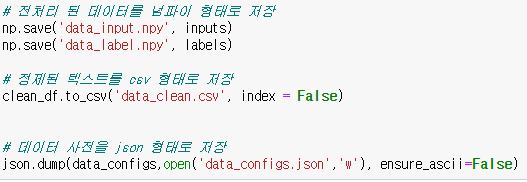


Max\_length는 단어개수의 중간값 사용한다.

이유 : 평균은 일부 데이터의 길이가 지나치게 길면 급격히 올라가기 때문에 자연어처리에서는 쓰이면 위험하기 때문이다.

**🡪벡터화 및 패딩까지 마친 데이터는 딥러닝에 사용한다.**

**-데이터 저장**



**5. 모델링 및 평가**

**<머신러닝>**

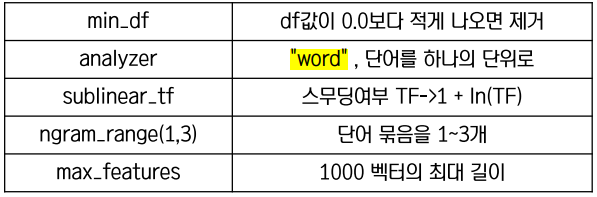
**1. TF-IDF로 text벡터화 한 경우**

**-TF-IDF** : TF(특정 단어가 하나의 데이터에서 등장한 횟수) X I(역수)DF(특정단어가 여러 데이터에서 틍장한 횟수)로 벡터화를 시키는 도구이다.

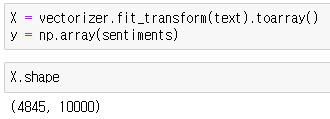
이를 통해 특정 단어가 하나의 데이터에서 아무리 많이 등장해도 여러데이터에서 많이 등장해버리면 값이 작아진다.



\*파라미터 설명



처음에는 analyzer를 char로 했으나 성능이 낮게 나와 word로 지정해주었다.

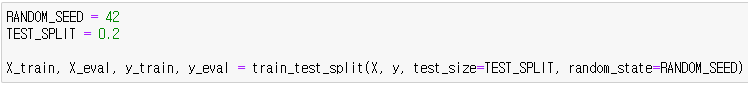
 -> text에 벡터화 적용

**\*이 때 데이터가 4845개로 줄어든 이유는 불용성 제거를 해주었을 때 한 개의 데이터가 null값이 되어 학습에 지장을 주기에 제거해주었다.**



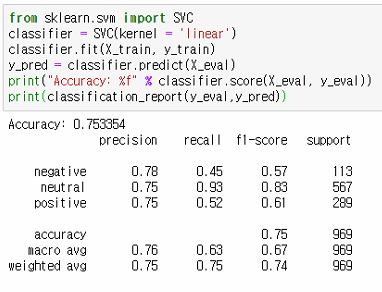


**-data 나누기**

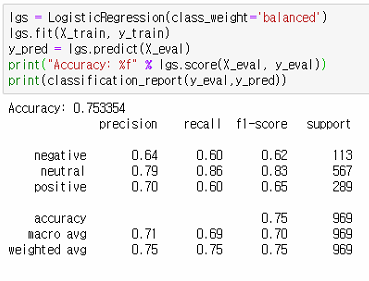


0.2의 비율로 train과 test data를 나누어 주었다.

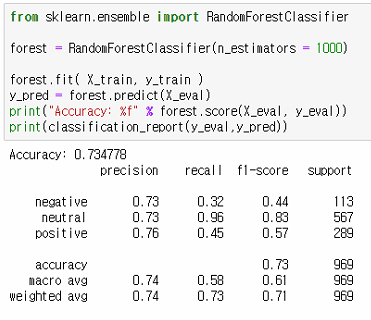
**-SVM**



**-LogisticRegression**



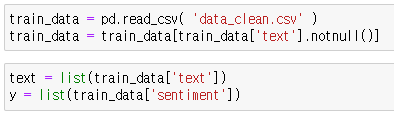
**-RandomForest**

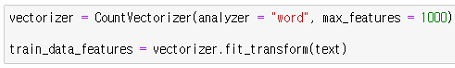


**2. CounterVectorizer로 text벡터화 한 경우**

**-CounterVectorizer**: 텍스트에서 횟수를 측정한 뒤 벡터로 만들어준다.

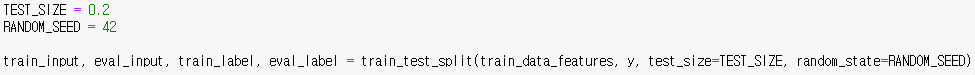
이때 횟수 기준을 word로 잡았다.





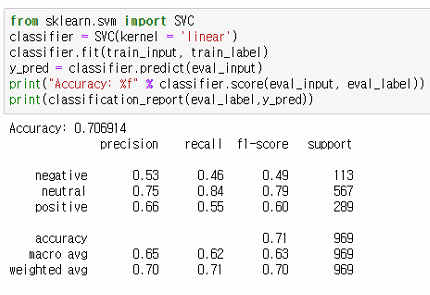


**-데이터나누기**

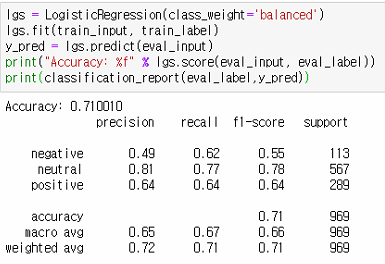


0.2의 비율로 train과 test data를 나누어 주었다.

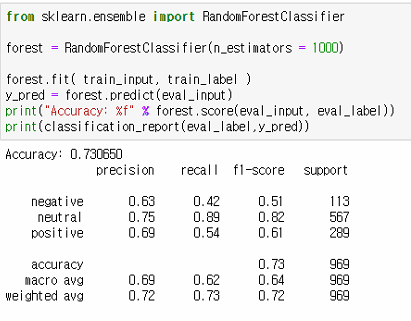
**-SVM**

****

**-LogisticRegression**

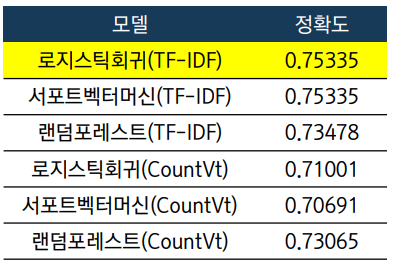
****

**-RandomForest**

****

**-결과분석**

TF-IDF로 벡터화 후 학습한 로지스틱회귀모델이 가장 정확도가 높았으며 f1스코어도 고루 높았다. 따라서 가장 성능이 좋은 모델이라고 보여진다.

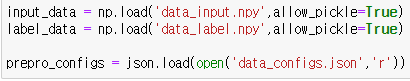


**<딥러닝>**

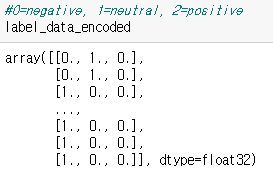
**1. 데이터 불러오기 및 전처리**

to\_categorical을 이용해 원핫인코딩을 해준다.

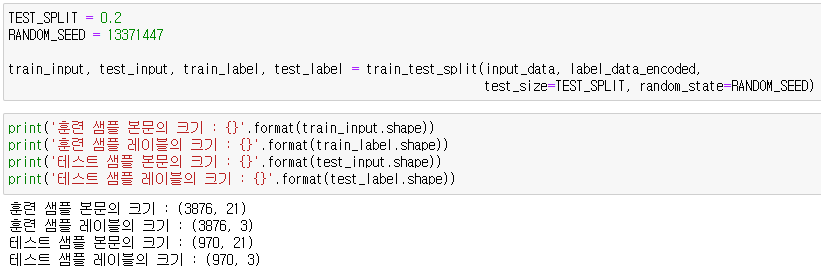
1이있는 인덱스가 0인경우 negative, 1인경우 neutral, 2인경우 positive 이다.







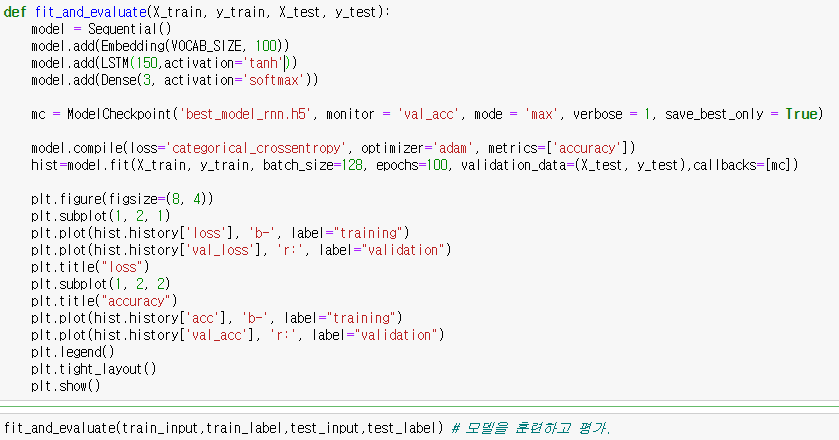
**2. 데이터 나눠주기**



**\*vocab\_size 지정(토크나이저로 단어 인덱스화할 때 쓰인 단어 개수)**



**-RNN**



🡪input층: embedding층, Embedding(input\_dim(첫번째 인자)=총 단어 개수, output\_dim(두번째 인자)=임베딩 벡터의 출력 차원)

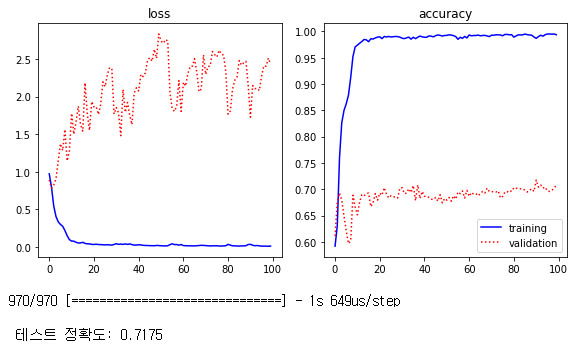
* input\_length=입력 시퀀스의 길이( 여기서는 21 ) 지정하던 안하던 결과는 같다.

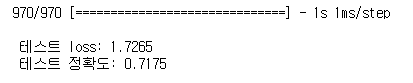
🡪output층: Denselayer로 softmax함수 활용

 이런 형태로 출력한다.

이 출력값은 인덱스 1의 probability가 크므로 neutral로 판단한 것이다.

학습완료 후 결과:



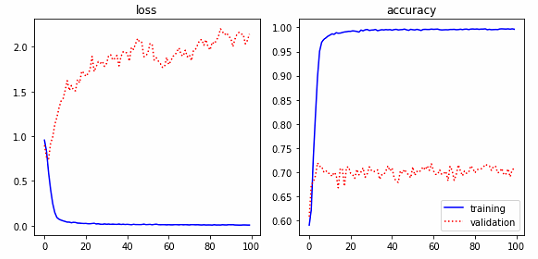


**-RNN+CNN**





학습완료 후 결과:

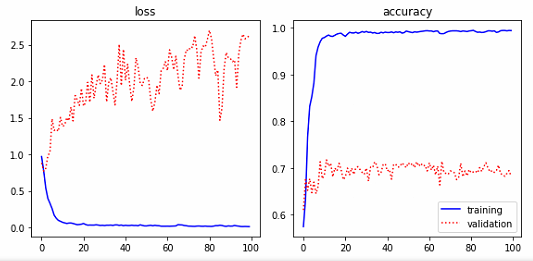
****

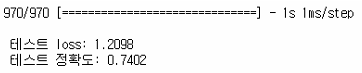
****

**-CNN(Conv1D 1개)**



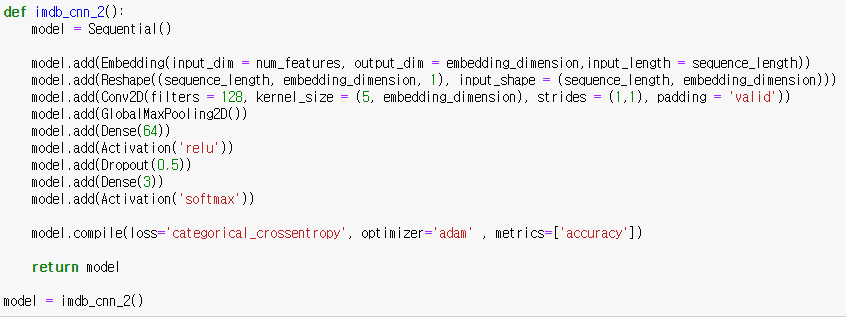
학습완료 후 결과:

****

****

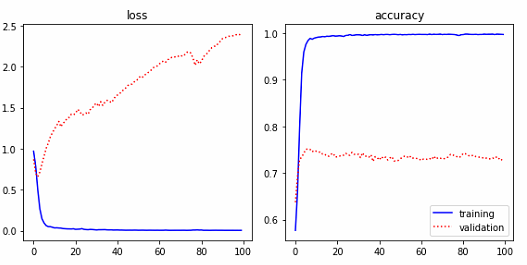
CNN만 사용시 이전보다 정확도가 크게 증가한 것을 볼 수 있다.

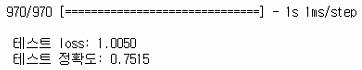
**- CNN(Conv2D 1개)**





학습완료 후 결과:

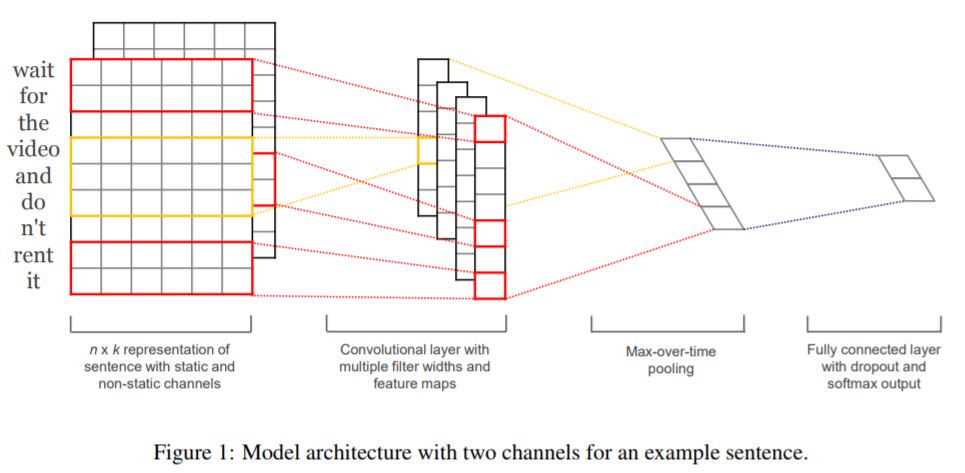
****

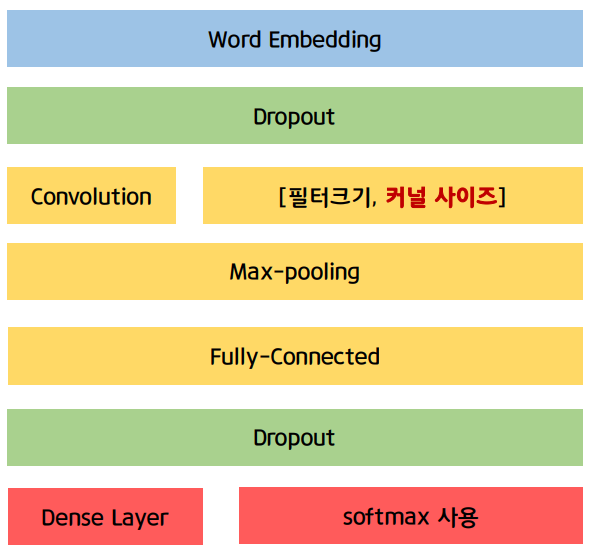
****

위와 같은 구조에서 Conv1D대신 Conv2D를 사용했더니 loss가 감소하고 정확도가 증가했다.

**-** **CNN(Yoonkim박사의 CNN for sentence classification)**

**출처:** Convolutional Neural Networks for Sentence Classification - Yoon Kim (New York University)



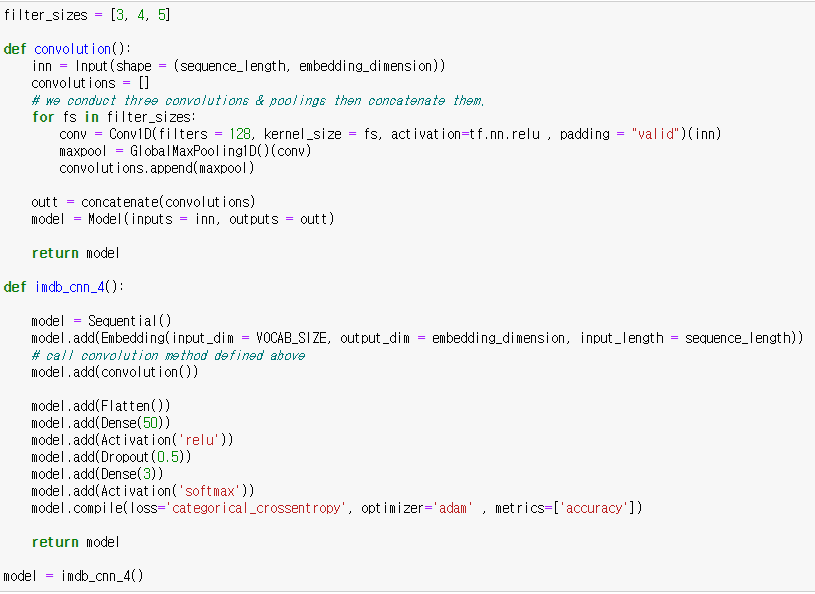


커널 사이즈를 다르게 줌으로 써 N-gram의 효과 즉, 다양한 각도에서 문장을 보는 효과가 있다고 한다.

이 모델을 직접 구현해 보았으며, convolution layer에 filters=128, kernel\_size=[3,4,5]로 주었다.

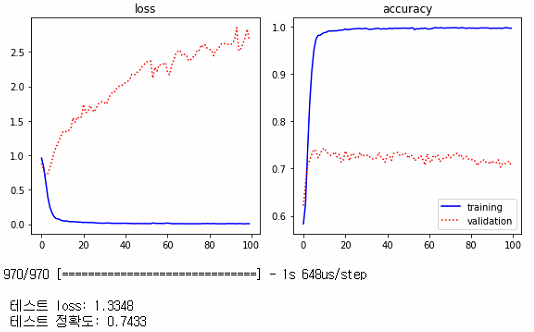
Conv1D와 Conv2D를 사용했을 때를 비교해 보았다.

**-Conv1D 3개 사용**



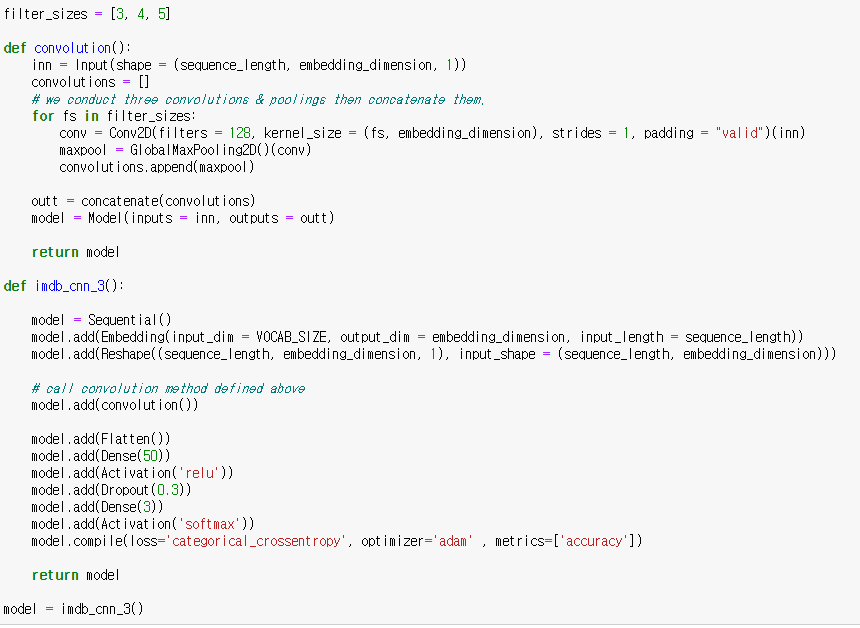


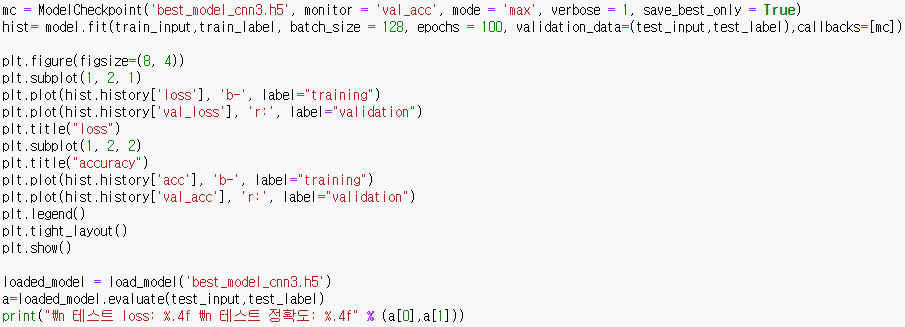
학습완료 후 결과 :



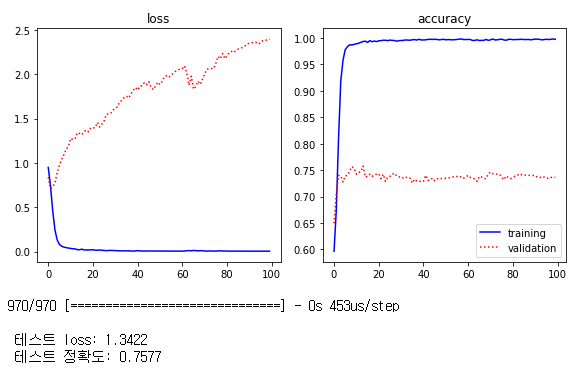
Conv1D를 1개 사용했을 때 보다 정확도가 올랐다.

**-Conv2D 3개 사용**





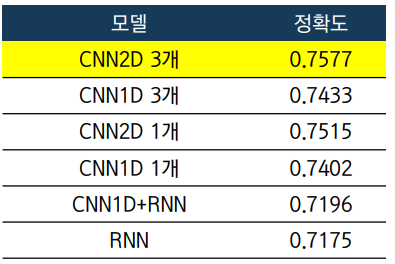
학습완료 후 결과 :



이 때까지 학습한 모델 중 가장 높은 정확도를 가졌다.

**-결과분석**

이 데이터의 감정분류는 RNN보다 CNN이 적합하며, Conv2D를 3개를 이용할 때 정확도가 가장 높다.



**6. 기대효과**

이 학습된 모델의 성능을 좀더 향상 시켜 실제로 사용한다면 지금도 무수히 쏟아져 나오고 있는 경제 뉴스에 대한 감정을 실시간으로 분류할 수 있을 것입니다.

더 나아가 분류된 데이터를 통해 앞으로 다가올 경제 위기 또는 호황을 예측할 수 있을 것이며 주가예측에도 사용될 수 있을 거라 기대됩니다.