

합성곱 신경망을 이용한 한글 텍스트 감성 분류기 설계

김원우[○], 박광현

광운대학교

asd130394@gmail.com, akaii@kw.ac.kr

Design of Korean Text Emotion Classifier Using Convolution Neural Network

Wonwoo Kim[○], Kwang-Hyung Park

Kwangwoon University

요약

기존의 자연어 처리 분야에서 감성을 분석하려는 연구가 많이 있었다. 자연어 처리 연구는 텍스트라는 심볼릭 데이터로부터 특징(Feature)을 추출해야 하는데, 특징들을 추출하는 방법과 추출된 특징을 조합하는 방법에 따라 성능이 달라진다. 최근 활발히 연구되고 있는 딥 러닝(Deep learning)은 비선형 특징을 추출하고 조합하는데 탁월한 성능을 보이고 있다. 본 논문에서는 딥 러닝 분야의 이미지 특징 추출에서 높은 성능을 보이는 합성곱 신경망을 텍스트 데이터에 적용하여 한글 텍스트 감성 분류기를 설계하였다.

1. 서론

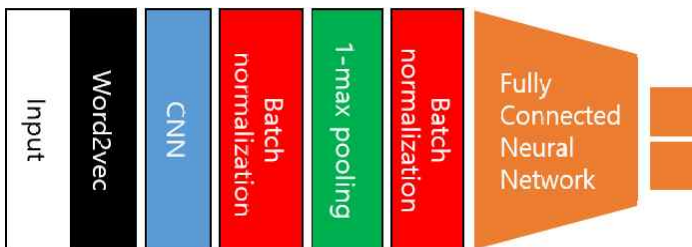
수집된 텍스트 데이터로부터 작성자의 주관적인 의견을 찾아 긍정인지 부정인지 분석하는 감성 분석 연구가 자연어 처리 분야에서 연구되고 있다. 또한, 딥 러닝의 순환 회귀 신경망을 이용하여 텍스트를 분석하고 생성하는 연구가 진행되고 있다[1][2]. 이미지 데이터 분석에서 탁월한 성능을 보이는 합성곱 신경망(CNN)은 데이터의 특징을 추출하고 복잡도를 줄이는 역할을 한다[3][4]. 최근에는 이러한 합성곱 신경망의 장점을 텍스트 데이터에 적용하는 연구가 진행되고 있다[5][6].

본 논문에서는 [5][6]에서 제안한 합성곱 신경망 모델을 기반으로 여러 가지 기법을 추가한 한국어 텍스트 분류기를 설계한다.

2. 본론

2.1 신경망 구조

[그림 1]은 본 논문에서 사용하는 신경망의 구조를 나타낸다.



[그림 1] 한글 텍스트 감성분석을 위한 신경망 구조

한국어 텍스트를 형태소 단위로 나누어 Word2vec의 입력으로 사용하였으며, Word2vec은 skip-gram 알고리즘을 사용하여 학습하였다[7]. 또한, 각 층의 입력 데이터 분포를

균일하게 하기 위해 배치 정규화(Batch normalization)를 수행하였다.

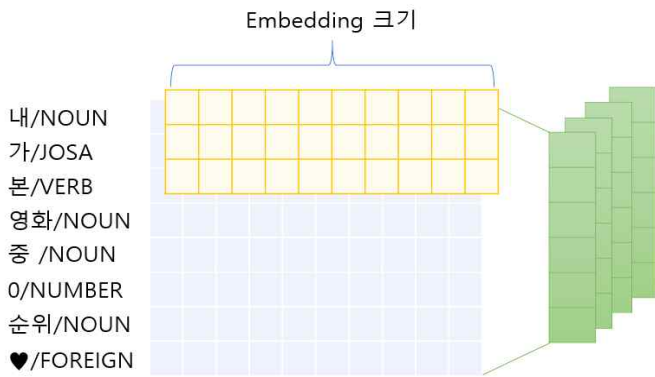
[표1]은 본 논문에서 사용하는 인공 신경망의 구조이다. 차원의 B는 배치의 크기, E는 임베딩의 크기, L은 문장의 길이, n은 필터의 크기, C는 채널의 수를 나타낸다. 괄호 안의 성분은 배치, 높이, 폭, 채널의 순서이다. 이렇게 구성된 인공 신경망은 역전파 알고리즘을 사용하여 학습하였다.

[표1] 인공 신경망의 구조

인공 신경망 구조	차원 / 값
Input	(B , L)
Word2vec	(B , L , E , 1)
CNN(n×200)(ReLU)/BN	(B , L-n+1 , 1 , C)
1-Max pooling	(B , 1 , 1 , C)
BN / FNN(ReLU)	(B , C)
Drop out, l2-Norm	0.7 , 0.1
Output	(B , 2)
Cost	Softmax cross-entropy

2.2 합성곱 신경망

본 논문의 합성곱 신경망이 기존의 신경망과 다른 점은 필터의 크기이다. 기존의 신경망은 (n × n) 필터를 사용하는데 본 논문에서는 [그림 2]와 같이 (n × E) 필터를 사용한다. 이미지는 각 픽셀이 의미를 갖지만 텍스트 데이터는 단어마다 임베딩 크기만큼 의미를 갖기 때문에 그 크기만큼 필터를 만든다. 합성곱 신경망 층에서 필터를 사용하여 여러 채널을 생성한다.



[그림 2] 문장과 필터의 합성 곱

2.3 1-Max pooling

기존의 Max pooling은 (n x n)의 형태로 pooling을 하지만 본 논문에서는 특징 벡터 전체를 (1 x L-n+1)의 형태로 Max pooling하여 총 C개의 pooling 결과를 얻는다.

2.4 배치 정규화

입력 데이터의 분포를 균일하게 하기 위해 배치 정규화를 적용하였다[8]. 배치 정규화는 미니 배치 단위의 특징들이 무관계성을 보인다고 가정하고 특징 각각에 대해 정규화한다. 식 (1)에서 B 는 미니배치에 대한 x값이다.

$$\begin{aligned}
 u_B &= \frac{1}{m} \sum_{i=1}^m x_i, \\
 \sigma_B^2 &\leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \\
 \hat{x}_i &\leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \\
 \therefore y_i &\leftarrow \gamma \hat{x}_i + \beta
 \end{aligned} \quad (1)$$

정규화를 통해 평균과 분산을 0, 1로 만드는 것은 활성화 함수층의 비선형성을 없앨 수 있기 때문에 식 (1)과 같이 기울기 인수 γ 와 스케일 인수 β 를 사용하여 보완한다. 본 논문에서는 합성 곱 신경망 층에서 1-Max pooling 층의 입력을 넣을 때와 pooling 된 결과를 Fully Connected Layer의 입력으로 넣을 때 배치 정규화를 적용하였다.

3. 실험

3.1 학습 데이터

학습 데이터는 깃헙(git-hub)의 네이버 영화 리뷰 데이터를 사용하였다[9]. 긍정, 부정 10만 개씩, 총 20만 개의 문장으로 구성되어 있다. 한글 형태소 분석기인 Twitter -

Korean - text를 이용하여 학습 데이터를 만들고, 그 중에서 3,000개의 문장을 무작위로 뽑아 검증 데이터를 구성하였다. 55,823개의 단어장을 구성하였으며 Word2vec을 이용하였다. Word2vec의 결과는 [표 2]에서 볼 수 있다.

[표 2] Word2vec으로 학습한 ‘좋다’와 ‘별로’의 유사 단어 예시

단어	유사 단어	거리 Cosine Distance
좋다/Adjective	괜찮다/Adjective	0.619
	훌륭하다/Adjective	0.596
별로/Noun	그닥/Noun	0.628
	그냥/Noun	0.5927

[표 2]는 ‘좋다’와 ‘별로’라는 단어 각각의 유사 단어와 코사인 거리를 보여준다. ‘좋다/Adjective’는 ‘괜찮다/Adjective’와 가장 유사하고 ‘별로/Noun’는 ‘그닥/Noun’과 가장 유사하다.

3.2 실험 환경

본 논문에서는 텐서플로우(Tensorflow) 라이브러리를 사용하여 합성 곱 신경망을 구현하였으며, GeForce GTX 970(GPU)의 환경에서 학습시켰다.

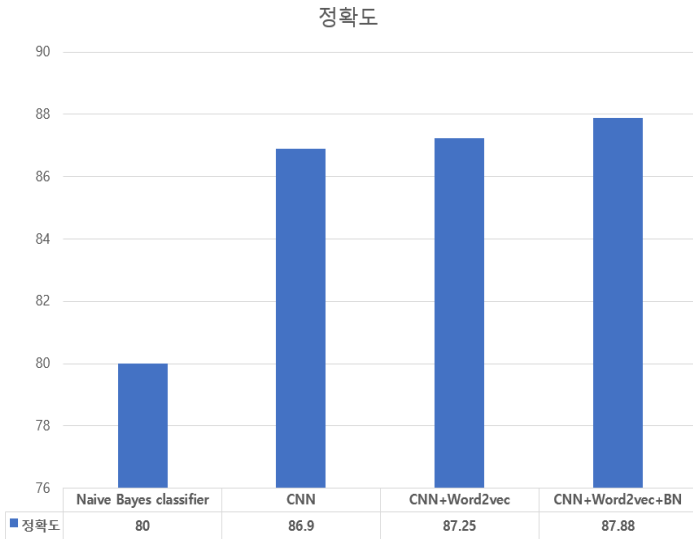
3.3 결과

기존의 Naive Bayes 모델, 합성 곱 신경망(drop-out, l2-norm), 합성 곱 신경망(drop-out, l2-norm)에 Word2vec을 적용한 경우, 합성 곱 신경망(drop-out, l2-norm)에 Word2vec과 배치 정규화를 적용한 경우를 실험하였으며, [표 3], [그림 4]와 [그림 5]는 결과를 보여준다.

[표 3] 분류결과

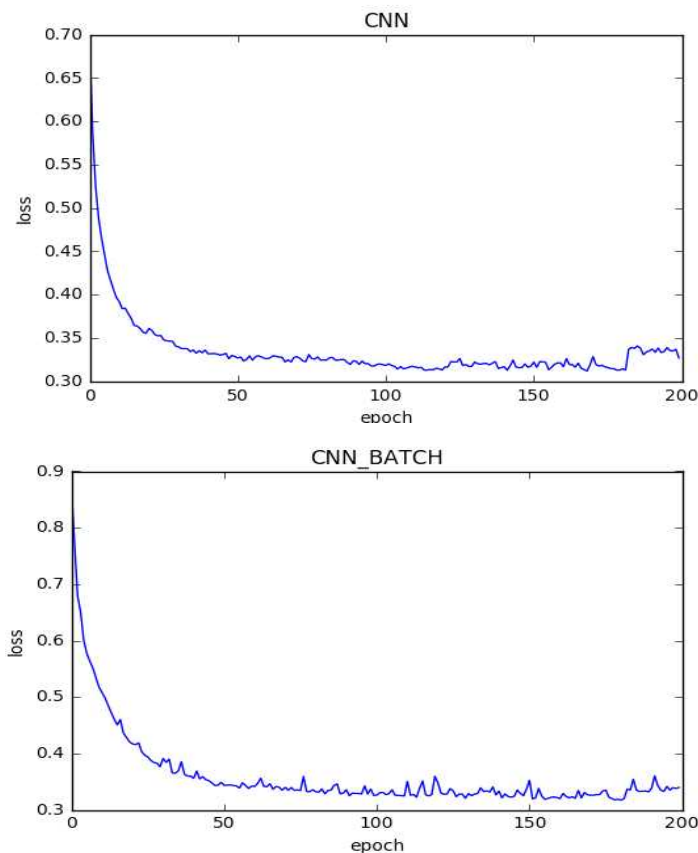
입력	출력	출력 라벨
이런스타일 액션영화 굿굿!!	긍정	긍정
....재미가없어요 시간이 아깝고	부정	부정
후속작 계획은 없나요..? ㄸ	부정	긍정

[표 3]을 보면 “후속작 계획은 없나요..?ㄸ”라는 입력에 대해서 잘못 분류를 하였다.



[그림 4] 분류 정확도

(왼쪽부터 순서대로 Naive Bayes, CNN, CNN+Word2vec, CNN+Word2vec+Batch Normalization)



[그림 5] CNN+Word2vec과
CNN+Word2vec+Batch Normalization의
과적합 비교

Drop out과 l2-norm을 고정하고 Word2vec과 배치 정규화가 성능에 미치는 영향을 실험하였다. [그림 4]는 Word2vec이 성능에 미치는 영향을 나타내고, [그림 4] 및

[그림 5]는 배치 정규화가 과적합 방지 및 성능에 미치는 영향을 보여준다. 배치 정규화는 깊은 층에서 발생하는 과적합을 방지하고 성능을 높이는 것으로 증명되었는데[8], [그림 5]에서 150 epoch이후의 손실(loss) 값을 통해 얇은 층에서도 효과가 있음을 확인할 수 있다.

4. 결론 및 향후 연구

본 논문에서는 합성곱 신경망을 사용하여 텍스트 감성 분류기를 설계하였으며, Word2vec과 배치 정규화를 통해 성능을 높일 수 있음을 보였다.

향후 연구로는 소설 등과 같이 등장인물이 존재하는 텍스트 데이터에서 인물 및 장소의 관계를 분석하는 연구가 필요하다.

참 고 문 헌

- [1] Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng and Christopher Potts, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," EMNLP, 2013.
- [2] Ilya Sutskever, Oriol Vinyals, Quoc V. Le, "Sequence to Sequence Learning with Neural Networks," NIPS, 2014.
- [3] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," CVPR, 2015.
- [4] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, "Going deeper with convolutions," CVPR, 2015.
- [5] Kim Yoon, "Convolutional neural networks for sentence classification," EMNLP 2014,
- [6] Ye Zhang, Byron C. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification," arXiv preprint arXiv:1510.03820.
- [7] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, "Distributed Representations of Words and Phrases and their Compositionality," NIPS 2013.
- [8] Sergey Ioffe, Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," TPAMI, 33, 2011.
- [9] 네이버 영화 리뷰데이터 깃헙(git-hub)페이지, <https://github.com/e9t/nsmc>