순환 신경망(LSTM)을 이용한 영화 평점 예측

강경필, 주재걸
고려대학교 컴퓨터학과
e-mail : rudvlf0413@korea.ac.kr
jchoo@korea.ac.kr

Predicting Movie Evaluation using Deep LSTM

Kyeongpil Kang and Jaegul Choo
Dept. of Computer Science, Korea University

요 약

소비자의 선호도 및 여론을 정량적인 방법으로 분석하기 위해 비정형 데이터의 분석은 필수적인 요소가 되고 있다. 하지만 비정형 데이터는 언어의 구조 및 모호성 등으로 인해 분석하기 어려운 형태이다. 따라서 본 연구는 최근 각광받고 있는 인공신경망, 특히 그 중에서도 순환 신경망의 한 모델인 Deep LSTM 을 이용하여 비정형 데이터를 분석하고 이를 활용하여 어순 및 어감 등의 언어의 구조적 문제에도 효과적인 정략적 모델을 설계하여 학습하고 이를 기존의 인공신경망 모델과 비교 분석하고자 한다.

1. 서론

마케팅 및 기업 경영 등에 있어서 중요한 것 중 하나는 바로 소비자의 선호도 및 호감도이다. 소비자의 선호도 및 호감도를 통해 제품의 재고 관리 및 새로운 제품을 개발하는데 있어 참조가 되기 때문이다. 하지만 소비자의 욕구 및 호감도를 측정하는 방법은 어려운 일이기 때문에 대다수의 기업에서는 설문조사 등을 통한 간접 평가를 하게 된다.

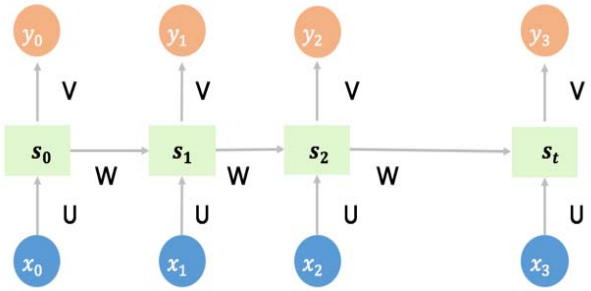
하지만 이러한 방법은 일부 소비자의 호감도만을 측정할 수 있을 뿐 전체를 일반화하기에는 한계가 있는 실정이다. 그러나 최근 기업들은 소셜 네트워크 서비스(SNS) 및 감정분석(sentiment analysis)을 이용한 비정형 데이터 마이닝 기법을 적용하고 있다. 이러한 대량의 데이터를 통해 다수의 소비자의 관점을 파악할 수 있기 때문이다. 하지만 이러한 데이터의 대다수는 비정형 데이터가 주를 이루고 있고, 텍스트 등으로 이루어진 비정형 데이터의 경우는 가공 및 처리, 분석하기 까다로운 단점이 있다.

다행히 최근 각광을 받고 있는 인공 신경망을 통해 비정형 데이터에 대한 가공 및 처리, 분석을 가능케 하는 연구들이 진행되고 있다. 특히 그 중 한 종류인 순환 신경망(RNN; recurrent neural networks)은 문장의 단어 순서 및 문맥을 고려하기 때문에 단어 간 순서 정보가 있는 비정형 데이터를 학습하기에 적합한 모델이다.

따라서 본 연구는 순환 신경망을 학습하여 관객이 남긴 리뷰를 통해 해당 관객의 평점을 예측하는 모델을 만들고, 이를 통해 문맥 및 어순을 갖는 비정형 데이터 분석을 통한 소비자의 호감도를 정량적으로 측정해보고자 한다.

2. 순환신경망(Recurrent Neural Networks)

순환 신경망이란 (그림 1)과 같이 현재의 입력값과 직전의 셀의 상태의 조합으로 현재의 결과값을 계산하는 신경망이다.



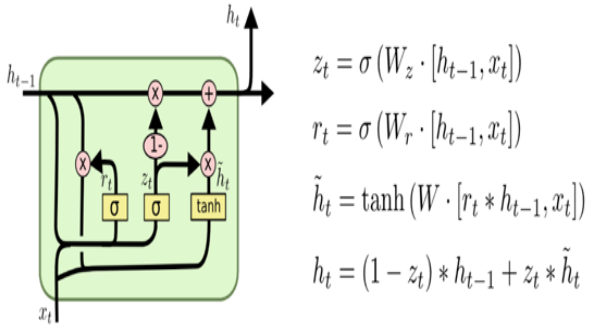
(그림 1) RNN 도식

각각의 선은 특성에 대한 가중치를 나타내며 다음과 같은 수식으로 나타낼 수 있다.

$$s_t = tanh(Ux_t + Ws_{t-1})$$
$$h_t = softmax(Vs_t)$$

순환 신경망은 학습 과정은 기존의 인공 신경망과 조금 다른데, 시간에 따라 나온 결과 값을 통한 경사하향법(BPTT; back propagation through time)을 통해 학습을 한다. 하지만 이 경우 미분값의 누적 곱이 0 에 수렴하기 때문에 길이가 긴 입력값을 받은 경우 gradient vanishing 문제와 같은 한계가 있게 된다. 이를 획기적으로 개선한 새로운 순환 신경망 모델이

LSTM (long short term memory)이고, 모델의 개괄은 (그림 2)과 같다.



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$
$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$
$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

(그림 2) LSTM 도식 및 수식 ([13]에서 발췌)

LSTM 의 경우 이전 상태를 이전 상태를 얼마나 망각할지, 현재의 입력값을 얼마나 받아들일지에 대해 조절하기 때문에 위에서 말한 Gradient Vanishing 문제를 해결할 수 있게 된다. 따라서 LSTM 의 경우 길이가 긴 입력값, 즉 긴 문장에 대해서도 학습이 용이하게 된다.

본 연구는 짧은 문장 뿐만 아니라 다양한 길이를 가진 영화 리뷰들을 학습하기 위해 LSTM 을 이용하여 모델을 설계하였다.

3. 데이터 및 전처리

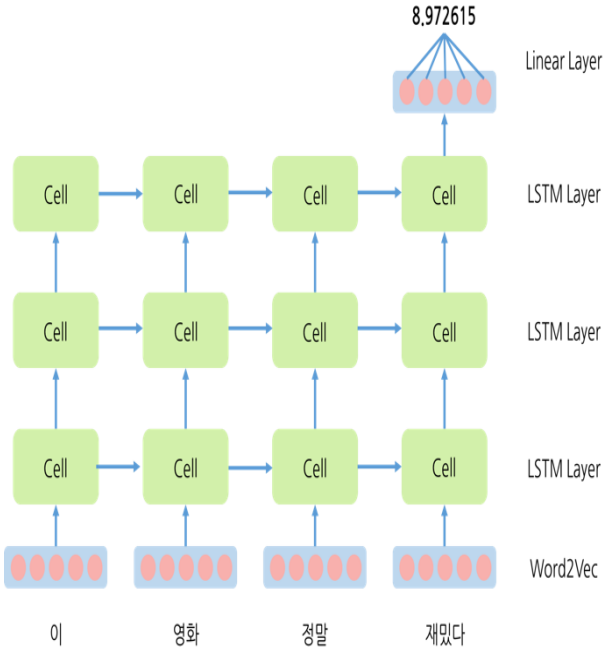
영화 평점 및 평가글, 특히 한글 자연어로 이루어진 데이터를 학습하기 위해 사용한 데이터는 네이버 영화(Naver Movie) 리뷰 데이터이다. 해당 데이터는 각각의 영화를 본 관객들이 리뷰와 함께 평점을 남기고, 다양한 영화에 대해 대량의 평점 데이터를 갖고 있기 때문에 평점 예측을 위한 모델을 학습하기 적합하다고 판단하였다.

전체 2,700 여 개의 영화 중 약 193 만 개의 영화 리뷰 및 평점 데이터를 수집하였다. 하지만 이 중 50% 이상이 10 점을 택하였기 때문에 학습된 모델 또한 이러한 경향을 가질 것이라 예상되어 10 점의 평점을 갖는 리뷰 데이터는 제외하였다. 그리고 대다수의 리뷰가 100 개 이하의 형태소를 포함하고 있고, 효율적인 학습을 위해 100 개 이하의 형태소를 갖는 리뷰들을 고려하였고, 빈도가 10 이상의 형태소만을 취급하였다. 이를 통해 977,858 개의 리뷰 및 평점(1~9 점 범위), 37,452 개의 형태소를 학습데이터로 사용하게 되었다.

또한 37,452 개의 단어를 온전히 벡터 형태의 입력값으로 사용하기에는 매우 많은 변수를 사용하게 되어 과적합(Over-fitting) 및 학습을 비효율적으로 하게 되는 문제가 있다. 따라서 이러한 문제를 해결하기 위해 Word2Vec 이라는 단어 임베딩(word embedding) 방법을 사용하였다. Word2Vec[6]이란 해당 단어가 사용된 주변의 문맥을 고려하여 특정 차원으로 사영하는 방법이다. 본 연구는 window size 를 5 로 설정한 Word2Vec 을 이용하여 37,452 차원의 입력 벡터를 256 차원의 벡터로 사영하여 학습을 진행하였다.

4. 모델 및 학습

본 연구는 순환 신경망 중 LSTM 을 이용하였고 이를 3 층의 LSTM layer 로 구성하여 모델을 좀더 유연하고 정확하도록 깊게 설계하였고 각각의 LSTM 셀(cell)은 128 dimension 의 결과로 출력하게 된다. 또한 마지막 입력에 대한 마지막 계층의 셀의 결과 벡터가 선형 레이어를 통해 계산되어 평점을 예측하게 된다. 전체적인 모델의 구성은 (그림 3)과 같다.



(그림 3) Deep LSTM 모델

각각의 레이어는 Dropout[8]과 L2 regularization 을 사용하여 과적합(over-fitting)을 방지하도록 하였다. 최적화 방법은 RMSprop 을 사용하였고, 계산된 평점과 실제 평점 간의 차이를 계산하는 loss 함수는 mean squared error 를 사용하였다. 매 epoch 마다 학습 데이터는 전체의 832,000 개를 사용하였고 테스트 데이터는 약 146,000 개(15%)를 사용하여 매 epoch 마다의 에러를 측정하여 너무 많은 횟수의 학습을 진행하여 발생하는 과적합을 방지(early stopping)하도록 하였고 이를 통해 15 epoch 동안 학습을 진행하였다. 그리고 복잡한 학습의 빠른 계산을 위해 GPGPU 를 사용하였고 batch size 는 128 로 하였다. 모델의 구현을 위해 언어로는 Python 2.7 를 사용하였고 딥러닝 라이브러리 중 Keras 라이브러리[12]를 사용하였다.

리뷰	실제 평점	예측한 평점
진짜 전쟁영화 중 아직도 잊혀지지 않는..감동 굿	9	8.12501144
전쟁보다 뜨거운 형제애.. 손에 꼽을 만큼 인상적이다	8	8.0163269
씁쓸한 현실에 통쾌한 한방	9	7.54761076
이게 뭐가 재미있는지...	1	3.60447502
스토리 상의 개연성도 떨어지고, 어울리지 않는 역할의 배우들도 영화에 몰입하는 걸 방해함	1	3.7998209

<표 2> 실제 리뷰를 통한 평점을 예측한 예

리뷰	예측 평점	
	Deep LSTM	DNN
영화가 재밌지만 배우가 연기를 너무 못한다	5.51341343	5.83494663
배우가 연기를 너무 못하지만 영화는 재밌다	6.48727274	5.8401289
연기는 볼만하지만 영화는 너무 지루하다	5.88856125	6.13369274
영화는 너무 지루하지만 연기는 볼만하다	7.2376647	6.03732109
재밌지만 결말이 너무 어처구니없다	3.29077196	4.01654625
결말이 너무 어처구니없지만 재밌다	5.20533323	4.01654625

<표 2> 어순에 따른 Deep LSTM 과 DNN 비교 예

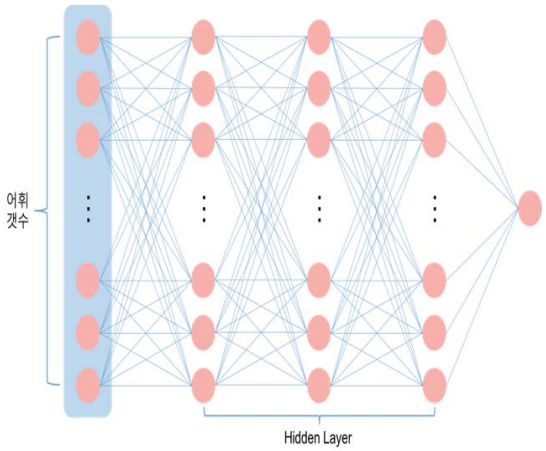
5. 결과

전체 학습을 끝냈을 때 1~9 점의 범위 중에서 관객들이 실제로 남긴 리뷰에 대해 본 연구가 제안한 모델이 예측한 점수들은 <표 2>와 같다.

본 연구의 모델(Deep LSTM)의 장점은 입력의 순서 및 문맥을 고려한다는 것이다. 따라서 리뷰의 어순 혹은 어감에 민감하게 반응할 수 있다. 이를 확인하기 위해 LSTM 과 비교할 대상으로 기존의 깊은 인공신경망(DNN; deep neural network)을 (그림 4)와 같이 설계하여 학습하였다. 해당 신경망은 본 연구의 모델과 같이 3 개의 비선형 계층 구조이고 각각의 hidden layer 는 본 연구의 모델의 변수의 개수와 비슷하게 하기 위해 노드의 개수는 12 개로 하였다. 그리고 각각의 hidden layer 의 활성함수(activation function)는 ReLU(rectified linear unit)을 사용하였고 제안한 모델과 마찬가지로 dropout 을 사용하였다. 본 모델과 다른 점은 비교 대상의 모델은 순환신경망의 구조처럼 순차적으로 입력을 받지 않기 때문에 Word2Vec 을 사용하지 않고 단어 어휘 개수를 입력 벡터의 차원으로 설정하였다.

DNN 모델에 대해서도 학습을 한 뒤, 어순이 반대되는 리뷰 쌍에 따른 두 모델(Deep LSTM, DNN)의 예측 결과는 <표 3>과 같다.

예컨대 "한국말은 끝까지 들어봐야 한다" 라는 말이 있다. 그만큼 한국어의 경우 중요한 문맥이 뒤에 온다. 이와 같이 위 결과에서 홀수 행은 긍정문이 앞에 나오고 부정문이 뒤에 나와서 전체적으로 부정적인 문장이고 짝수 행은 그 반대이다. 결과를 보면 알 수 있듯이 비교 대상인 DNN 모델의 경우 차이가 거의 없는 반면 Deep LSTM 모델의 경우 긍정/부정 어순에 대한 차이가 뚜렷함을 알 수 있다. 이를 통해 Deep LSTM 모델이 기존의 인공 신경망보다 더욱 정확하게 어감을 잡아낸 다는 것을 알 수 있다.



(그림 4) DNN 모델(비교 대상)

1~9 점을 범위로 한 테스트 데이터를 기준으로 Deep LSTM 과 DNN 의 평균 에러 점수는 <표 3>과 같다.

모델	에러(점수)
Deep LSTM	**2.9184**
DNN	2.9994

<표 3> Deep LSTM 과 DNN 에러 비교

6. 결론

본 연구는 순환 신경망(Deep LSTM)의 학습을 통해 영화의 리뷰, 특히 그 어순을 분석하여 평점을 효과적으로 예측하였다. 특히 문장의 어순 및 어감을 효과적으로 짚어낸다는 점에서 기존의 어순을 고려하지 않은 신경망을 통한 평점 예측보다 우위에 있다. 따

라서 이를 통해 비정형 데이터를 통한 소비자의 욕구 및 선호도, 여론 경향 파악 및 분석에 있어서 더 효과적이고 정량적인 분석할 수 있을 것으로 기대된다.

물론 본 연구의 모델의 고도화 및 예측 오차율에 있어서 더 개선되어야 할 점들이 존재한다. 따라서 추후 연구에서 언어의 구조적 정보를 모델에 적용, hidden layer 및 node 개수, Word2Vec 의 window size 등의 hyper-parameter 의 최적화, Word2Vec 의 전역적 정보 이외의 추가적인 지역적 정보 학습, 더 많은 비정형 학습 데이터 수집 및 정밀한 자연어 처리 및 가공 등을 통해 더 정확하고 효율적인 모델로 개선할 수 있도록 하고자 한다.

참고문헌

[1] Williams, Ronald J., and David Zipser. "A learning algorithm for continually running fully recurrent neural networks." Neural computation 1.2 (1989): 270-280.
[2] Mikolov, Tomas, et al. "Recurrent neural network based language model." Interspeech. Vol. 2. 2010.
[3] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.
[4] Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." arXiv preprint arXiv:1409.2329 (2014).
[5] Mikolov, T., and J. Dean. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems (2013).
[6] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).
[7] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.
[8] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." Journal of Machine Learning Research 15.1 (2014): 1929-1958.
[9] Tieleman, Tijmen, and Geoffrey Hinton. "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude." COURSERA: Neural Networks for Machine Learning 4.2 (2012).
[10] Diaz, Fernando, Bhaskar Mitra, and Nick Craswell. "Query Expansion with Locally-Trained Word Embeddings." arXiv preprint arXiv:1605.07891 (2016).
[11] Naver Movie
http://movie.naver.com
[12] Keras Deep Learning Library
https://keras.io
[13] LSTM 도식 및 수식
http://colah.github.io/posts/2015-08-Understanding-LSTMs/