Convolutional Neural Networks for Sentence Classification

- 2014 -

TASK

Positive	Negative
GREAT movie and the family will love it!! If kids are bored one day just pop the tape in and you'll be so glad you did!!! ~~~~Rube i luv raven-s!	The script for this movie was probably found in a hair-ball recently coughed up by a really old dog. Mostly an amateur film with lame FX. For you Zeta-Jones fanatics: she has the credibility of one Mr. Binks.
Did Sandra (yes, she must have) know we would still be here for her some nine years later? See it if you haven't, again if you have; see her live while you can.	I would love to have that two hours of my life back. It seemed to be several clips from Steve's Animal Planet series that was spliced into a loosely constructed script. Don't Go, If you must see it, wait for the video ...
Verry classic plot but a verry fun horror movie for home movie party Really gore in the second part This movie proves that you can make something fun with a small budget. I hope that the director will make another one	This is without a doubt the worst movie I have ever seen. It is not funny. It is not interesting and should not have been made.

Selection of reviews including all the formatting and typos.

하나의 문장으로 된 영화 리뷰를 긍정인지 부정인지 분류해보자 (Movie Review Data, Pang and Lee 2005)

Model Architecture

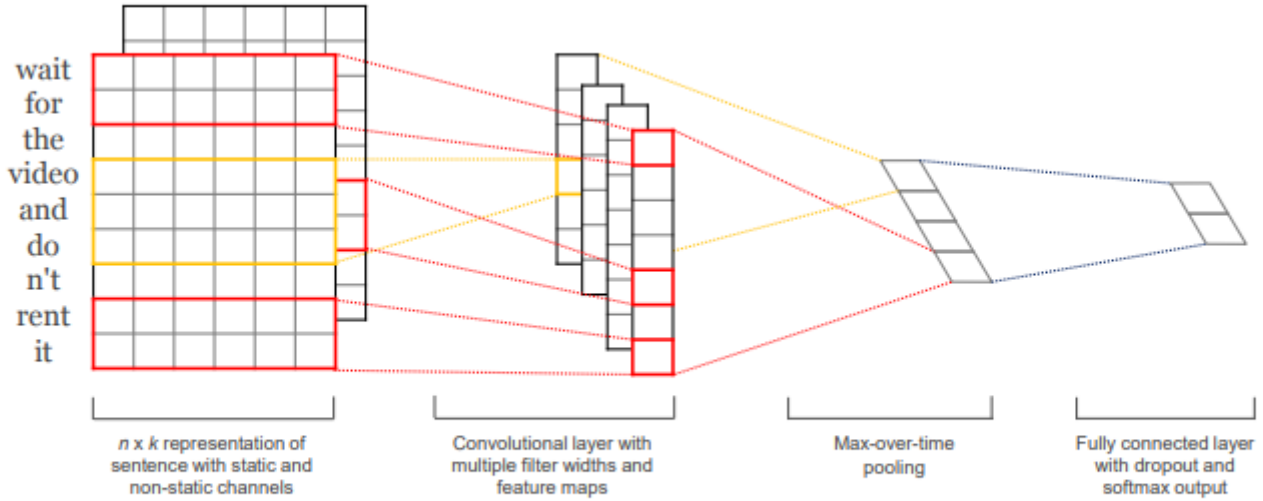
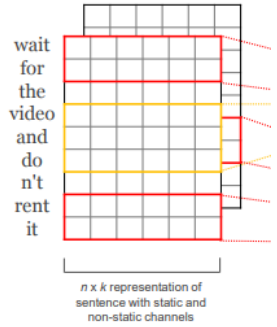


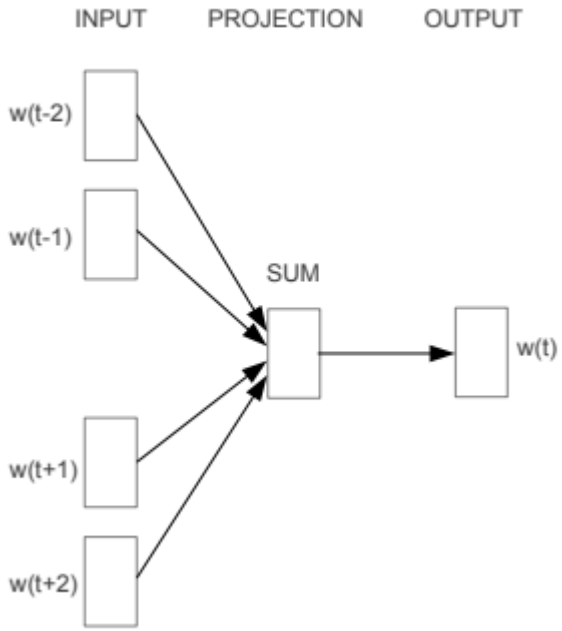
Figure 1: Model architecture with two channels for an example sentence.

여러 단어(하나의 단어는 K 길이의 벡터)들을 받아서 (1: 긍정, 0: 부정)으로 분류하는 구조

Word representation



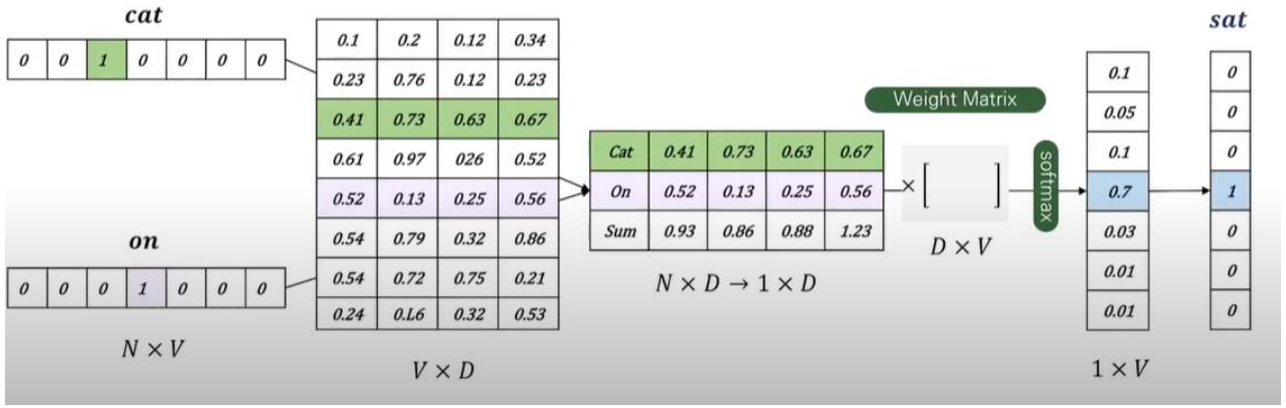
하나의 단어를 k 길이의 숫자로 이루어진 벡터로 만드는 작업



Efficient Estimation of Word Representations in Vector Space (2013)

- One Hot Encoding: 귤 [1,0,0], 오렌지 [0,1,0], 사과 [0,0,1]
  - 단어가 많아지면 엄청 벡터가 길어지고 유사한 단어가 유사한 벡터로 표현되지 않음
- Distributed Representation: 귤 [0.8,0.7,0.1], 오렌지 [0.7,0.8,0.1], 사과 [0.1,0.1,0.7]
  - 벡터의 길이는 k로 고정되어 있고, 비슷한 단어들이 비슷한 숫자로 표현되면 좋겠다

Word representation



우선은 모든 단어들을 one-hot Encoding하여 데이터를 만들어준다.

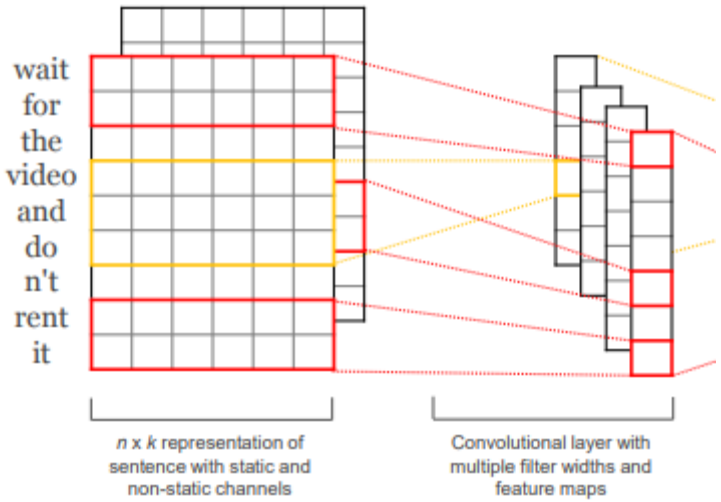
< TASK >

- ✓ y: 단어 하나
- ✓ x: 그 단어의 앞 뒤 단어들
- ✓ 앞 뒤 단어들을 input으로 넣어서 중간 단어를 예측한다

모델 구조의 중간에 (단어 개수 X Representation Dimension)의 매트릭스가 존재하는데,
학습이 다 끝난 다음에 이 매트릭스의 k번째 row를 전체 사전에서 k번째 단어의 representation으로 사용한다.

Reference: [Paper Review] Efficient Estimation of Word Representations in Vector Space

Convolutional Layer



Data
- ✓ k 길이의 representation을 concat해서 n * k 길이의 1차원 data

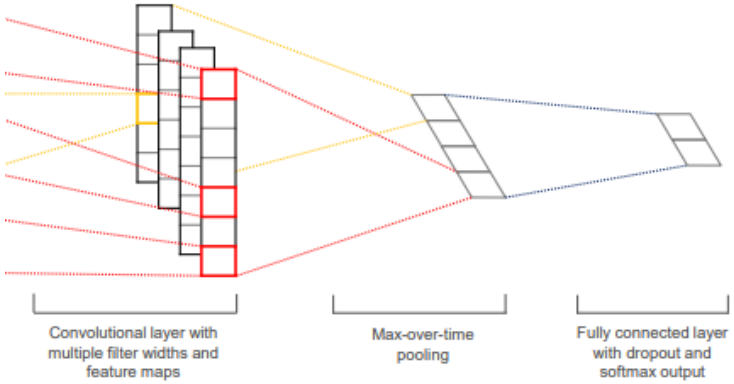
Feature map C: [c1, c2, c3, …,]
- ✓ Window size h에 따라서 feature map의 길이가 달라진다.

$$c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b).$$

- ✓ w는 window size * k 크기의 벡터, b는 bias
- ✓ f는 hyperbolic tangent 같은 non-linear function

Multichannel feature map:
- ✓ 여러 h를 사용하여 여러 feature map을 생성한다.

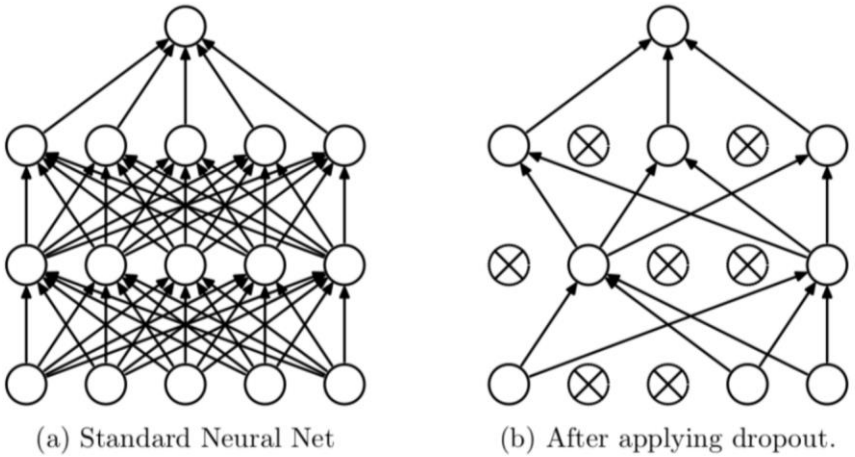
Classification



Max-over-time pooling
- ✓ Feature map C에서 가장 큰 값을 취한다.
- ✓ 컨셉
  - ✓ 하나의 filter에서 가장 중요한 하나의 feature만 추출한다.
  - ✓ 문장의 길이가 다름에서 오는 문제점을 해결한다.

FC layer with dropout and soft-max output
- ✓ y의 종류가 최종 layer dimension

Regularization



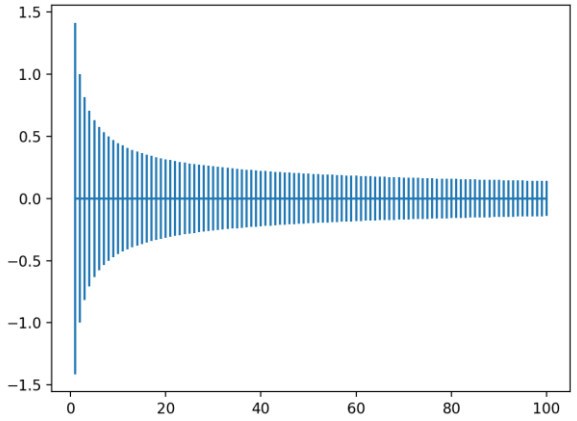
(a) Standard Neural Net (b) After applying dropout.

Dropout

- ✓ Pooling 이후의 Fully-connected layer에서 사용
- ✓ parameter p: 각 node가 dropout될 확률
- ✓ Multi channel의 개수가 m이라고 하면
  pooling 이후의 벡터 z의 dimension도 m.
  z에다가 0 혹은 1로 되어있는 벡터 r (dimension: m)을 곱함

$$y = \mathbf{w} \cdot (\mathbf{z} \circ \mathbf{r}) + b,$$

- ✓ 모델이 학습되고 test를 할 때는 dropout을 수행하지 않고
  weight matrix w에다가 parameter p를 곱해서 w = pw로 바꿈
  (train 된 w는 p만큼 적은 노드만 사용하는 것에 익숙해져 있으니깐)



Weight constraint

- ✓ 특정 weight가 너무 커지는 것을 방지한다.
- ✓ parameter s: l2 norm of weight의 상한선
- ✓ 만약 weight vector w의 l2 norm이 s보다 크다면
  l2 norm이 s가 되도록 scaling을 진행한다.

Experiments

Multi-Channel vs Single Channel

- ✓ Convolution을 여러 필터로 진행할 것인지 그냥 하나로만 할지에 따른 분류

Static vs Non-static Representation

- ✓ Word-representation을 학습 과정에서 업데이트 할지 말지에 따른 분류
- ✓ 그냥 word2vec 모듈을 static하게 사용하면
  bad랑 good이 유사하게 나옴
  (문맥적 위치가 유사해서)
- ✓ 긍정/부정 dataset에 적합해서 학습을 진행하면
  bad랑 good이랑 유사도가 낮아짐
- ✓ non-static이 항상 성능 향상을 가져오는 것은 아님

	Most Similar Words for	
	Static Channel	Non-static Channel
bad	*good* *terrible* *horrible* *lousy*	*terrible* *horrible* *lousy* *stupid*
good	*great* *bad* *terrific* *decent*	*nice* *decent* *solid* *terrific*
n't	*os* *ca* *ireland* *wo*	*not* *never* *nothing* *neither*
!	*2,500* *entire* *jez* *changer*	*2,500* *lush* *beautiful* *terrific*
,	*decasia* *abysmally* *demise* *valiant*	*but* *dragon* *a* *and*

Table 3: Top 4 neighboring words—based on cosine similarity—for vectors in the static channel (left) and fine-tuned vectors in the non-static channel (right) from the multichannel model on the SST-2 dataset after training.