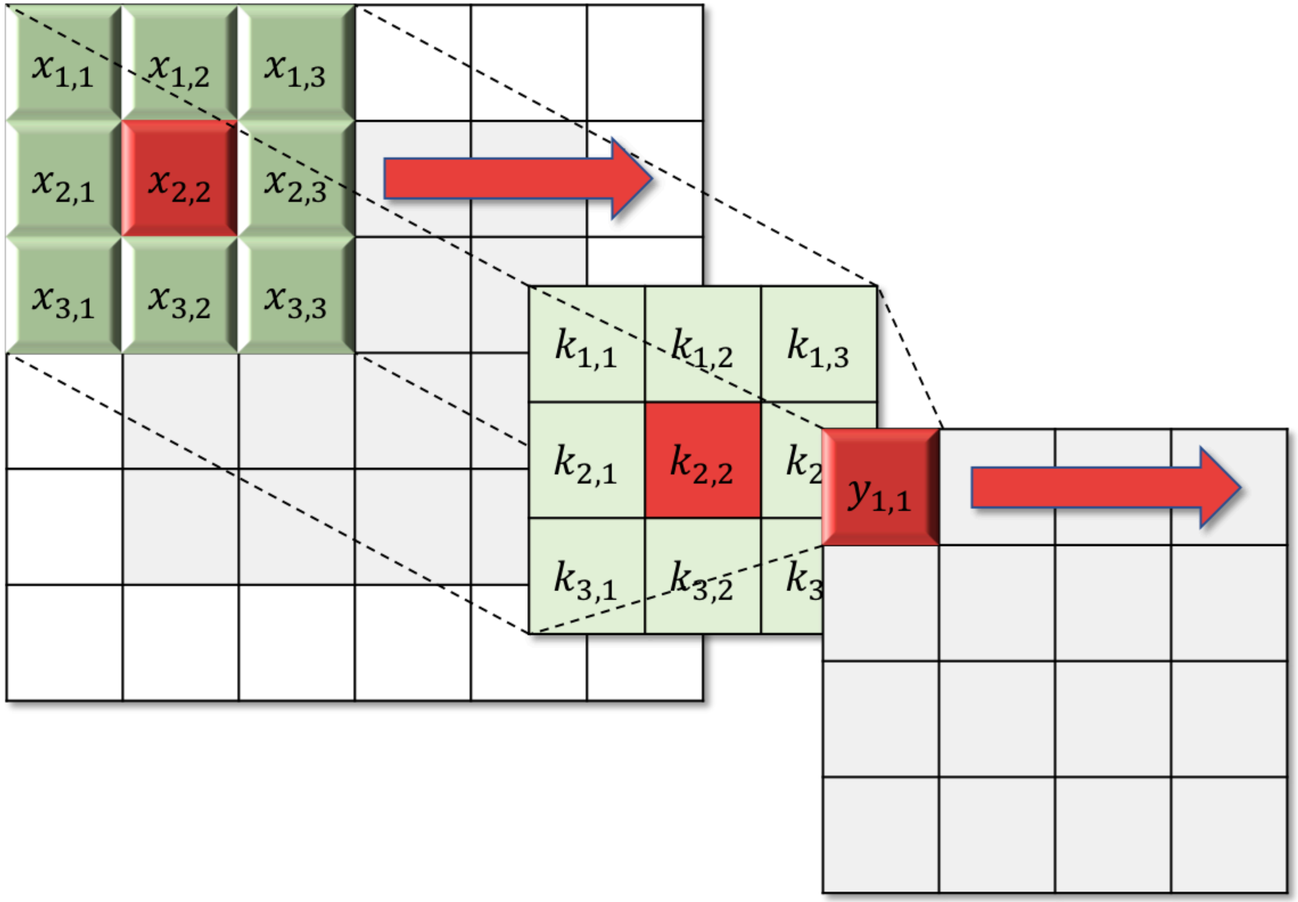
Text Classification using CNN

Ki Hyun Kim

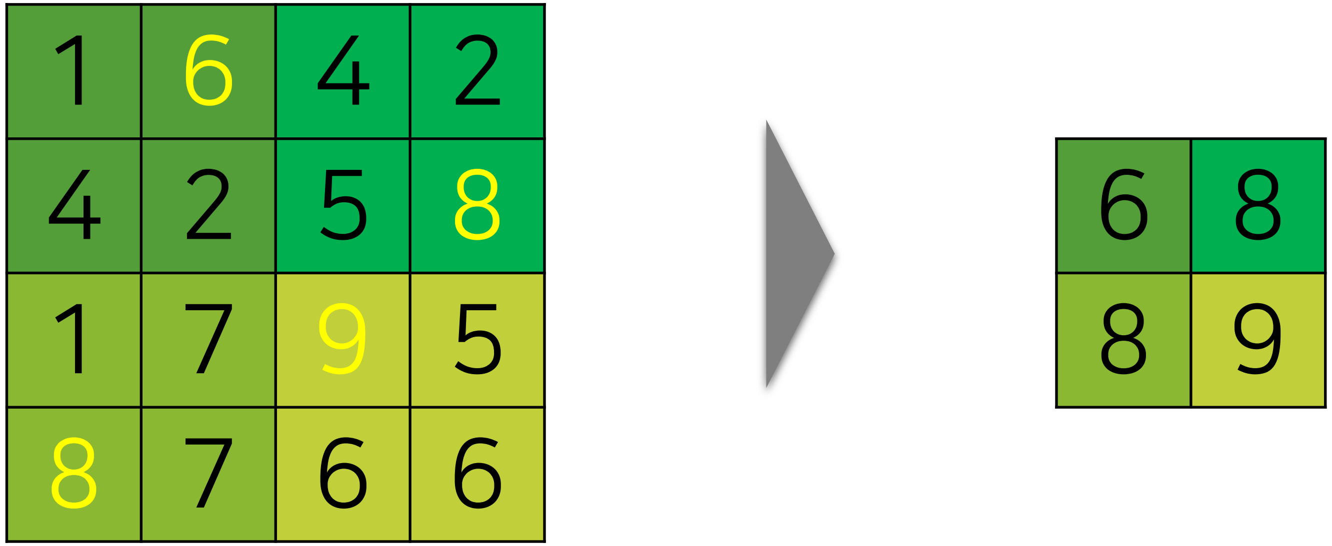
nlp.with.deep.learning@gmail.com

Convolutional Neural Networks



Max-pooling

- Down sampling 기법



Motivations

- 감성 분석(sentiment analysis)
  - 긍정: "너무 좋은", "기분이 행복", "품질에 만족", "맛이 훌륭한", "서비스가 좋은"
  - 부정: "형편 없는", "너무 느린", "완전 비추", "개판 직전"

- 클래스를 결정하는 일부 문구의 패턴이 있기 마련

- 이때, 문구 내의 단어들을 비슷한 의미의 단어들로 치환할 수 있을 것
  - 좋은 ≈ 훌륭한 ≈ 만족스러운 ≈ 대박

- 비슷한 의미의 문구의 임베딩 벡터의 패턴을 인식할 수 있다면?

Introduction

- Convolutional Neural Networks for Sentence Classification [Kim, 2014]

www.aclweb.org › anthology ▾ 이 페이지 번역하기

Convolutional Neural Networks for Sentence Classification ...

Convolutional Neural Networks for Sentence Classification · Yoon Kim. Anthology ID: D14-1181; Volume: Proceedings of the 2014 Conference on Empirical ...

Y Kim 저술 - 2014 - 7070회 인용 - 관련 학술자료

Convolutional Neural Networks for Sentence Classification

Yoon Kim
New York University
yhk255@nyu.edu

ract

of experiments with
networks (CNN)
e-trained word vec-
l classification tasks.
ple CNN with lit-
ning and static vec-
nt results on multi-
arning task-specific
uning offers further
e. We additionally
dification to the ar-
for the use of both
c vectors. The CNN
in improve upon the
ut of 7 tasks, which
alysis and question

1 Introduction

Deep learning models have achieved remarkable results in computer vision (Krizhevsky et al., 2012) and speech recognition (Graves et al., 2013) in recent years. Within natural language processing, much of the work with deep learning methods has involved learning word vector representations through neural language models (Bengio et al., 2003; Yih et al., 2011; Mikolov et al., 2013) and performing composition over the learned word vectors for classification (Collobert et al., 2011). Word vectors, wherein words are projected from a sparse, 1-of-V encoding (here V is the vocabulary size) onto a lower dimensional vector space via a hidden layer, are essentially feature extractors that encode semantic features of words in their dimensions. In such dense representations, semantically close words are likewise close—in euclidean or cosine distance—in the lower dimensional vector space.

Convolutional neural networks (CNN) utilize layers with convolving filters that are applied to local features (LeCun et al., 1998). Originally invented for computer vision, CNN models have subsequently been shown to be effective for NLP and have achieved excellent results in semantic parsing (Yih et al., 2014), search query retrieval (Shen et al., 2014), sentence modeling (Kalchbrenner et al., 2014), and other traditional NLP tasks (Collobert et al., 2011).

In the present work, we train a simple CNN with one layer of convolution on top of word vectors obtained from an unsupervised neural language model. These vectors were trained by Mikolov et al. (2013) on 100 billion words of Google News, and are publicly available.[1] We initially keep the word vectors static and learn only the other parameters of the model. Despite little tuning of hyperparameters, this simple model achieves excellent results on multiple benchmarks, suggesting that the pre-trained vectors are 'universal' feature extractors that can be utilized for various classification tasks. Learning task-specific vectors through fine-tuning results in further improvements. We finally describe a simple modification to the architecture to allow for the use of both pre-trained and task-specific vectors by having multiple channels.

Our work is philosophically similar to Razavian et al. (2014) which showed that for image classification, feature extractors obtained from a pre-trained deep learning model perform well on a variety of tasks—including tasks that are very different from the original task for which the feature extractors were trained.

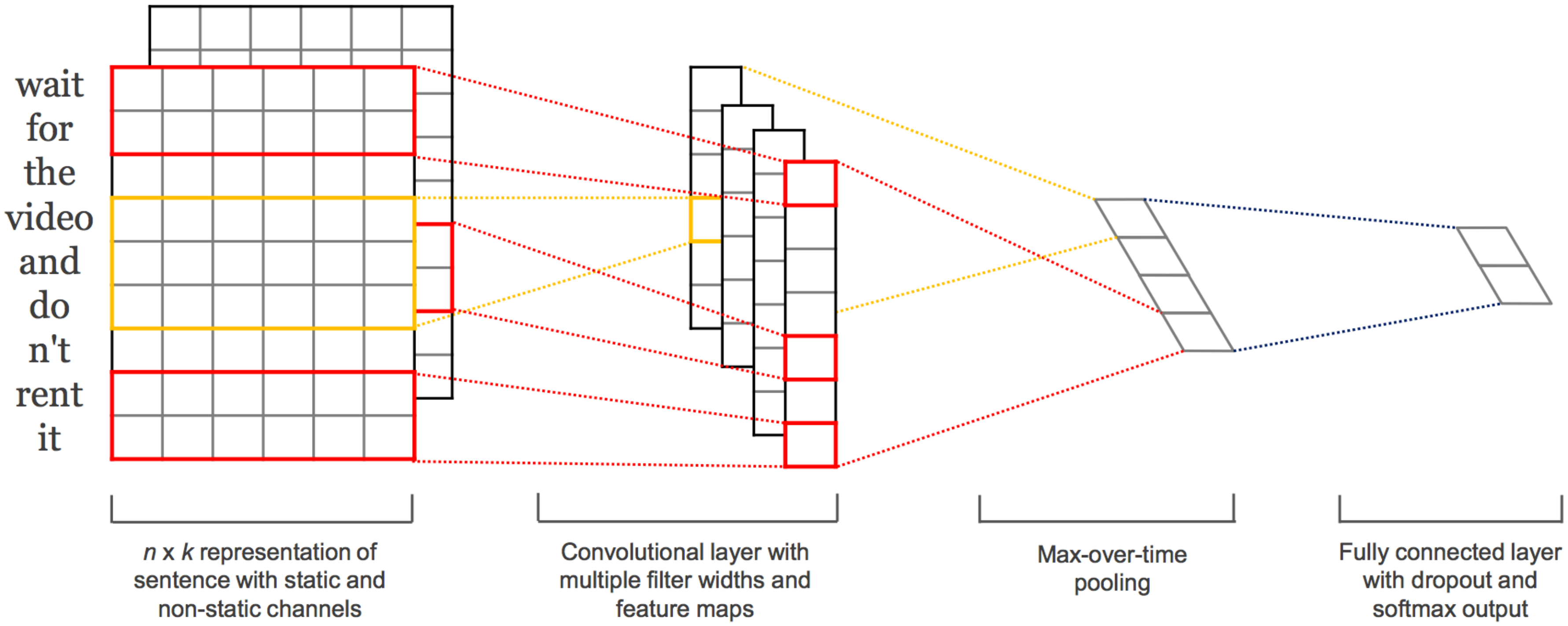
2 Model

The model architecture, shown in figure 1, is a slight variant of the CNN architecture of Collobert et al. (2011). Let $\mathbf{x}_i \in \mathbb{R}^k$ be the k-dimensional word vector corresponding to the i-th word in the sentence. A sentence of length n (padded where

[1] https://code.google.com/p/word2vec/

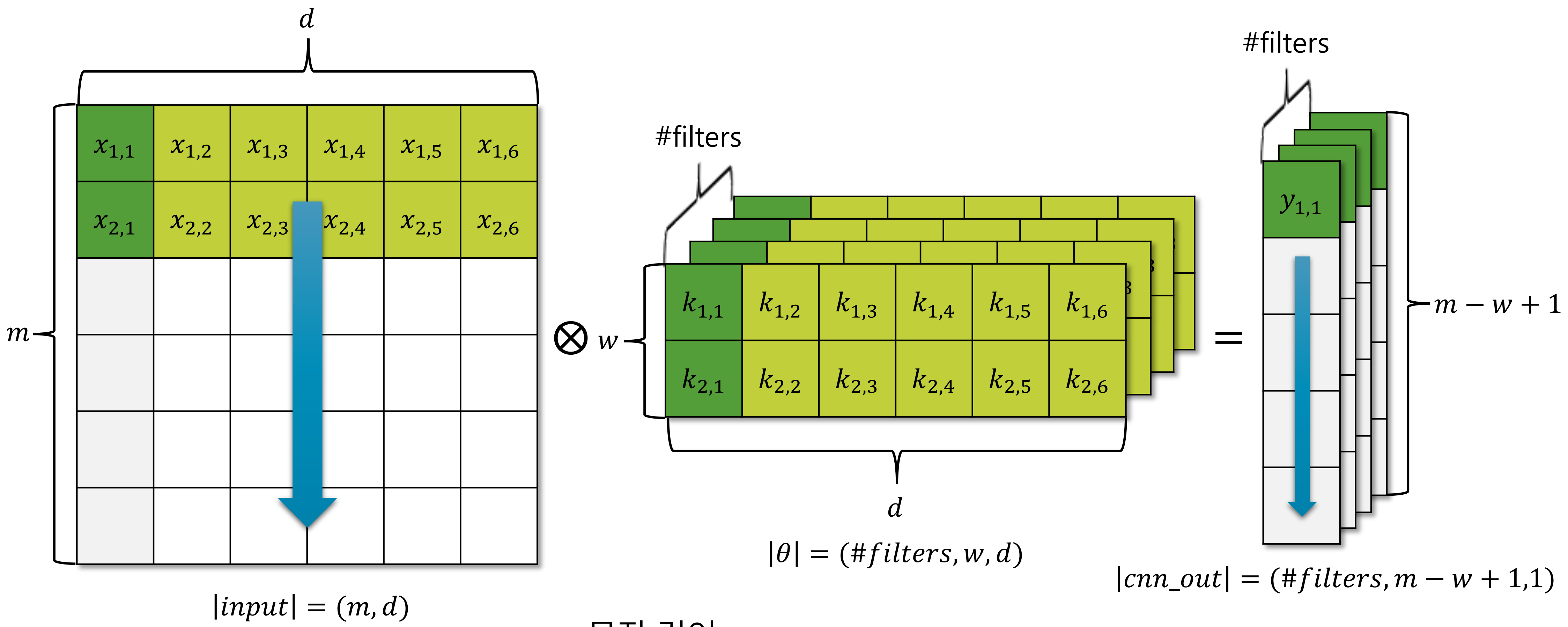
Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751,
October 25-29, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

Overview



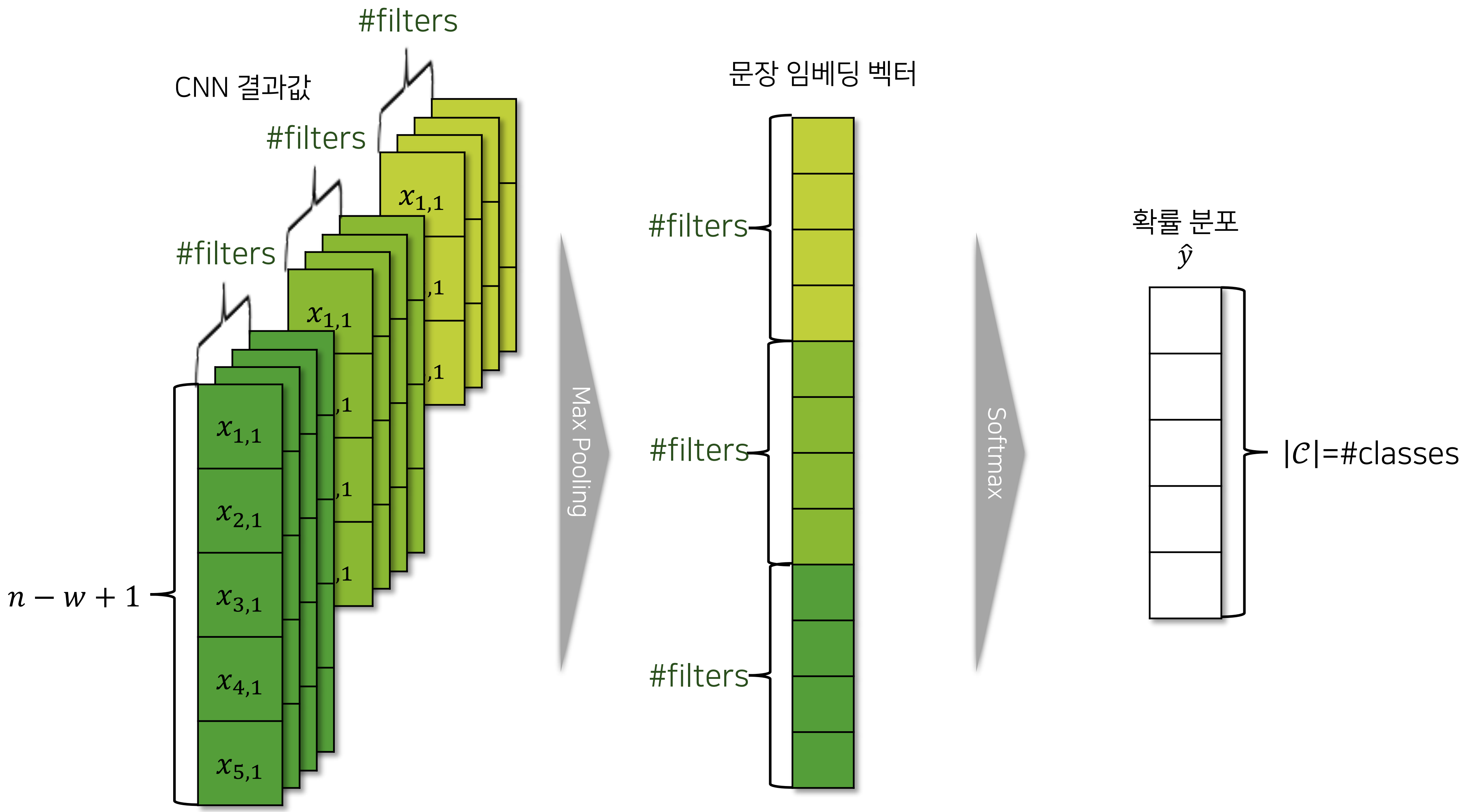
wait
for
the
video
and
do
n't
rent
it
n x *k* representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Text Classification with CNN [Kim, 2014]



m : 문장 길이
d : 임베딩 벡터 크기
w : 윈도우 크기 (패턴 내 단어의 갯수)

Text Classification with CNN [Kim, 2014]



Summary

- RNN에 비해 좀 더 직관적인 방법

- RNN은 문장의 문맥(context)을 이해한다면, CNN은 문장 내 단어의 패턴을 인식

- 당시 논문은 word embedding을 직접 수행한 이후에 넣어주는 것을 추천했지만, embedding layer에 one-hot vector를 넣어주어 구현할 것.