Convolutional Neural Networks for Sentence Classification

| 연세대학교 응용통계학과 이규민 |

NLP에 CNN을 써보자!

Yoon Kim

yoonkim@seas.harvard.edu
http://www.people.fas.harvard.edu/~yoonkim

Employment

Jun 2020 – Research Scientist
MIT-IBM Watson AI Lab

Education

2020 Ph.D., Computer Science
Harvard University
Advisor: Alexander Rush
Thesis: "Deep Latent Variable Models of Natural Language"

Notation

$\mathbb{x}_i \in \mathbb{R}^k$: k – dimension word vector of i^{th} word

$\mathbb{x}_{1:n} = \mathbb{x}_1 \oplus \ldots \oplus \mathbb{x}_n$: sentence of length n (n words)

$\mathbb{w} \in \mathbb{R}^{hk}$: filter applied to a window of h words

$b \in \mathbb{R}$: bias term

f : non-linear function (ex. hyperbolic function)

Model

$c_i = f(\mathbb{w} \cdot \mathbb{x}_{i:i+h-1} + b)$: non-linear function (ex. hyperbolic function)

Model

$c_i = f(\mathbb{w} \cdot \mathbb{x}_{i:i+h-1} + b)$: non-linear function (ex. hyperbolic function)

→ $\{\mathbb{x}_{1:h}, \mathbb{x}_{2:h+1}, \ldots, \mathbb{x}_{n-h+1:n}\}$ 이라는 문장에 대하여
$\mathbb{c} = [c_1, c_2, \ldots, c_{n-h+1}]$ 이라는 feature map 생성 ($\mathbb{c} \in \mathbb{R}^{n-h+1}$)
이후 max pooling을 통해 1개의 feature 획득

→ Window size 를 다르게 하며 multiple filter 로 학습
→ 그렇게 얻은 multiple feature 를 fully connected softmax layer 에 넣으면 label에 대한 distribution 얻을 수 있다.

Model

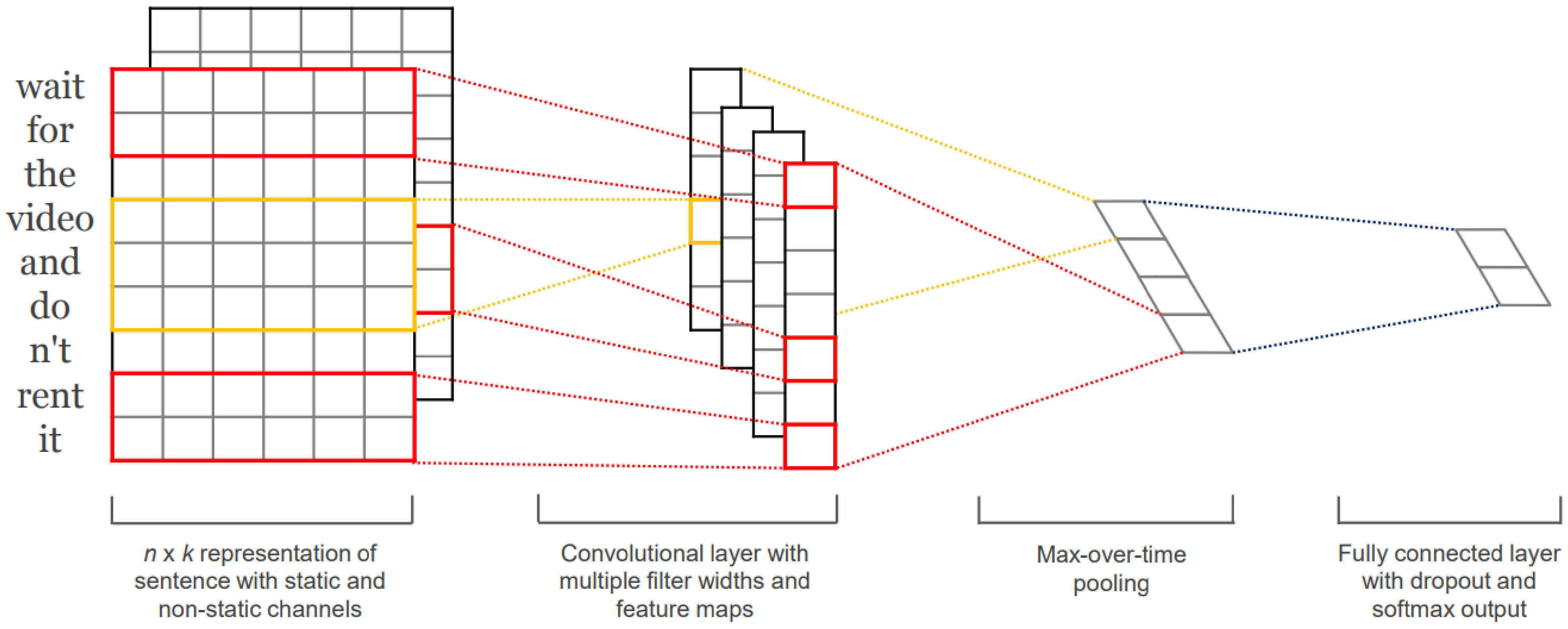
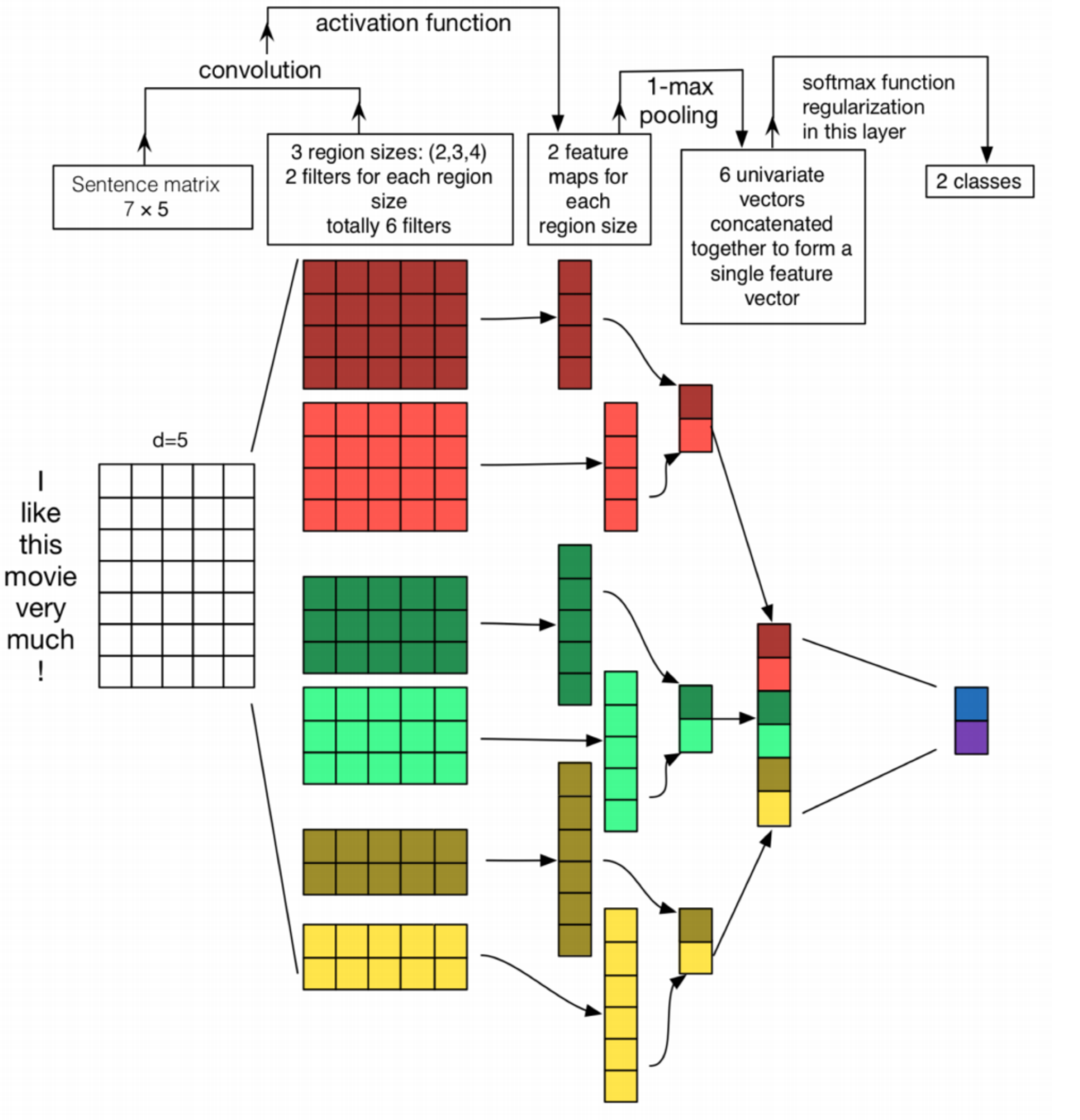


Figure 1: Model architecture with two channels for an example sentence.

Model

1. Filter 에 대해 convolution
2. f 라는 activation function에 대입
3. Max pooling

4. 위 1~3을 여러 filter에 대해 시행
5. 얻은 값을 fully connected softmax layer로 학습!



2.1 regularization

- Dropout

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

- $\mathbb{w} \cdot \mathbb{x} + b$ 대신 $\mathbb{w} \cdot (\mathbb{x} * \mathbb{r}) + b$ 로 학습 (*는 element-wise multiplication)

$\mathbb{r}$ 은 m dimension Bernoulli random variable with prob. P of being 1

- 이후 test 시 $\hat{\mathbb{w}} = p\mathbb{w}$ 를 weight 로 사용
- $||\mathbb{w}_c|| > s$ 이면 $||\mathbb{w}_c|| = s$ 로 rescale

Data set

Data	c	l	N	$\|V\|$	$\|V_{pre}\|$	*Test*
MR	2	20	10662	18765	16448	CV
SST-1	5	18	11855	17836	16262	2210
SST-2	2	19	9613	16185	14838	1821
Subj	2	23	10000	21323	17913	CV
TREC	6	10	5952	9592	9125	500
CR	2	19	3775	5340	5046	CV
MPQA	2	3	10606	6246	6083	CV

- Column 은 순서대로

target class의 수
평균 문장 길이
전체 data set size
단어 size
pre-trained word vector 에 있는 단어의 수
test set size

3.1 Hyperparameter & 3.2 pre-trained data

- Find hyperparameters based on dev set
- Nonlinearity: ReLU
- Window filter sizes h = 3, 4, 5
- Each filter size has 100 feature maps
- Dropout p = 0.5
- L2 constraint *s* for rows of softmax, *s* = 3
- Mini batch size for SGD training: 50
- Word vectors: pre-trained with word2vec, *k* = 300

3.3 Model variation

Static/ Non-static, Single channel / Multi-channel 에 따라

CNN-rand, CNN-static, CNN-nonstatic, CNN-multichannel 로 구분하여 학습

4. Result

Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	**89.6**
CNN-non-static	**81.5**	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	**88.1**	93.2	92.2	**85.0**	89.4
RAE (Socher et al., 2011)	77.7	43.2	82.4	–	–	–	86.4
MV-RNN (Socher et al., 2012)	79.0	44.4	82.9	–	–	–	–
RNTN (Socher et al., 2013)	–	45.7	85.4	–	–	–	–
DCNN (Kalchbrenner et al., 2014)	–	48.5	86.8	–	93.0	–	–
Paragraph-Vec (Le and Mikolov, 2014)	–	**48.7**	87.8	–	–	–	–
CCAE (Hermann and Blunsom, 2013)	77.8	–	–	–	–	–	87.2
Sent-Parser (Dong et al., 2014)	79.5	–	–	–	–	–	86.3
NBSVM (Wang and Manning, 2012)	79.4	–	–	93.2	–	81.8	86.3
MNB (Wang and Manning, 2012)	79.0	–	–	**93.6**	–	80.0	86.3
G-Dropout (Wang and Manning, 2013)	79.0	–	–	93.4	–	82.1	86.1
F-Dropout (Wang and Manning, 2013)	79.1	–	–	**93.6**	–	81.9	86.3
Tree-CRF (Nakagawa et al., 2010)	77.3	–	–	–	–	81.4	86.1
CRF-PR (Yang and Cardie, 2014)	–	–	–	–	–	82.7	–
SVM_S (Silva et al., 2011)	–	–	–	–	**95.0**	–	–

4.1 Multi vs single channel

→ Mixed results…….

4.2 Static vs non-static channel

→ more specific to the task-at-hand!

4.3 Others……

→ Dropout : effective!

→ Random initialization : can be improved!

	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*

5. conclusion

In the present work we have described a series of experiments with convolutional neural networks built on top of word2vec. Despite little tuning of hyperparameters, a simple CNN with one layer of convolution performs remarkably well. Our results add to the well-established evidence that unsupervised pre-training of word vectors is an important ingredient in deep learning for NLP.

→ simple CNN works good for NLP tasks, especially sentence classification!

Thank You :)

감사합니다