A Convolutional Neural Network for Modelling Sentences

윤훈상

CONTENTS

끝!

0. Abstract

문장을 표현하는 것은 언어 이해의 핵심
문장의 Semantic Modelling을 위한 DCNN 소개

This Network → Dynamic k-Max Pooling을 사용 (Global pooling operation over linear sequences)

→ Varying length of input sentences

→ 장단기 관계를 볼 수 있는 Feature Graph 도출

→ Parse Tree를 이용하지 않아 모든 언어에 사용 가능

4 Experiment: 1. (small scale) Binary class sentiment prediction

2. (Large scale) Multi-class sentiment prediction

3. Six-way question classification

4. Twitter sentiment prediction by distant supervision

Excellent in First Three task!

25% greater in Last Task!!

1. Introduction

Sentence model의 목적은

Classification or Generation을 위해

문장의 Semantic content를 표현하기 위함!

이 Sentence modelling 문제는 많은 자연어 이해 관련 문제 속에 있다.

→ 감정분석 (Sentiment Analysis)

→ Paraphrase detection

→ Entailment recognition

→ Summarisation

→ Discourse analysis

→ Machine translation

→ Grounded language learning

→ Image retrieval(?)

1. Introduction

각각의 '문장'은 낮은 빈도로 관찰된다. (같은 문장)
따라서 단어나 n-gram으로 문장을 바라 봐야한다.
즉, Sentence Modeling의 핵심은 문장의 Feature가
어떤 단어 및 n-gram의 Feature로 부터 왔는지 살펴봐야 한다.

의미에 관한 다양한 Model들이 제시되어 왔다.

→ Composition based methods : Co-occurrence statistics / Algebraic operations

→ Composition Function

→ Automatically extracted logical forms

→ (Central) Neural Networks : neural BOW / Bag-of-n-grams models 부터

Recursive neural networks or TDNN

1. Introduction

이 논문에서는 CNN을 문장의 Semantic modelling에 추가할 것이다.

1. 문장의 길이는 가변적
2. Network 사이에 1-d Convolutional layers와 Dynamic k-max pooling layers

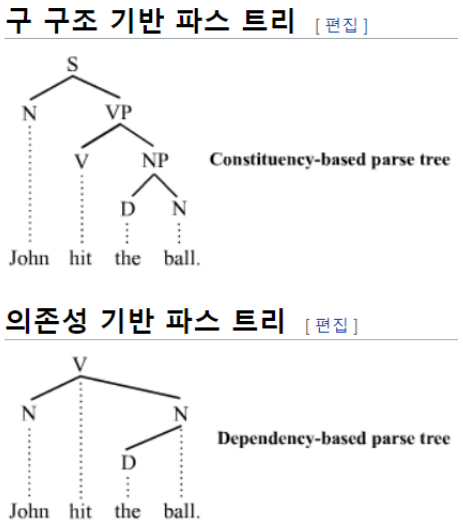
Dynamic k-max pooling은 Max pooling의 일반화

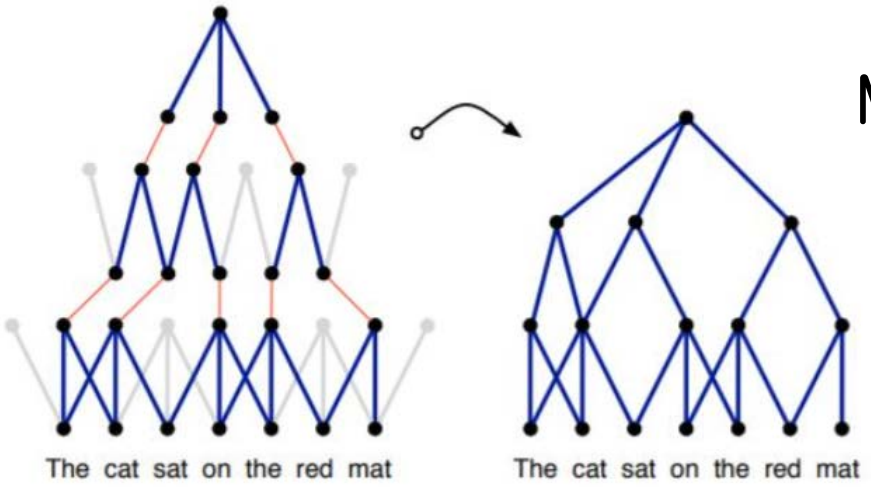
→ k-max pooling은 linear sequence의 단일 Max값이 아니라 k개의 max를 return

→ Network에 따라 k가 dynamically 하게 변한다!

1. Introduction

각 Convolutional layer는 sentence matrix에 1-d filter를 얹는다.
같은 filter를 n-gram씩 얹으면 feature들이 독립적으로 추출 가능





Multiple layers of convolutional & dynamic pooling operations

↓

Feature Graph

높은 층의 작은 Filter는 멀리 떨어져 있는 non-continuous한 phrase의 관계를 잡을 수 있다.

Syntactic parse tree와 유사해 보인다. 하지만 완벽히 같지는 않고 neural network쪽

1. Introduction

Section 2 : DCNN 개론

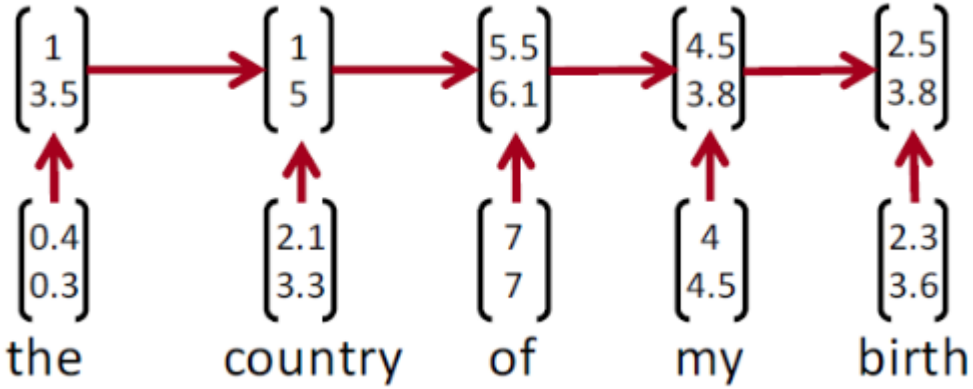
Section 3 : 관련 Operators와 Network Layer

Section 4 : Feature Graph와 Network 구성

Section 5: 실험 결과와 학습된 Feature detectors

Recurrent NN, Recursive NN, CNN

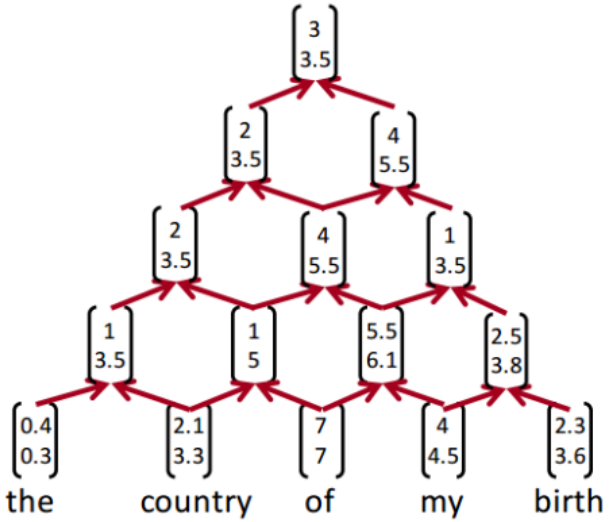
- Recurrent NN : 입력값을 순서대로 받아 하나씩 순차적으로 처리하는 네트워크 구조



마지막 히든 노드인 (2.5, 3.8)은 이전까지의 모든 맥락(the, country, of, my)과 함께 현재 입력 값(birth) 정보가 모두 반영된 것을 알 수 있습니다

CNN

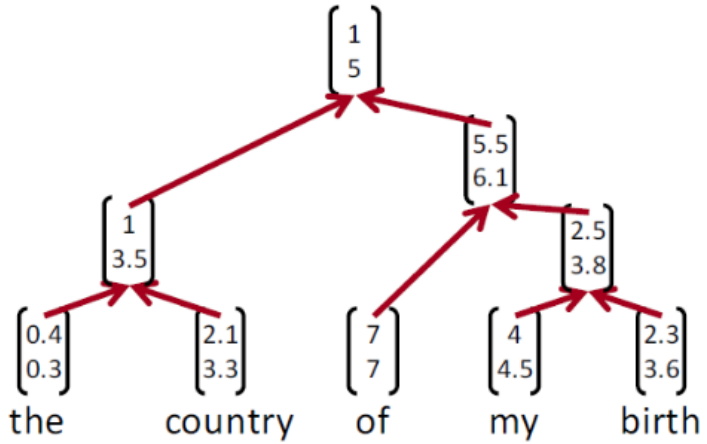
- 입력값을 생략없이 반영한다는 점에서 Recurrent Neural Network와 다를 바가 없다.

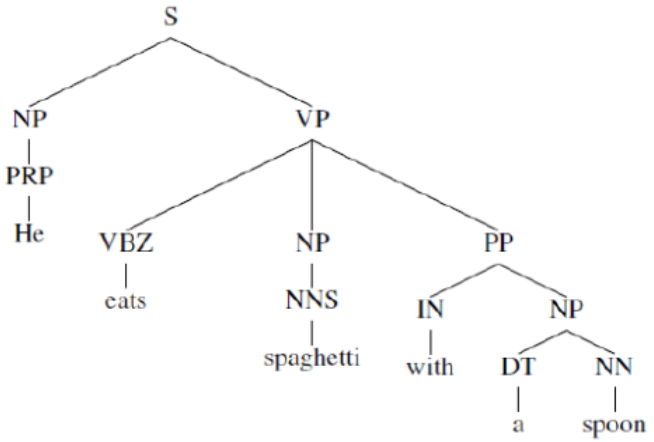


But CNN은 입력 값이 한번에 주어지고
Filter가 슬라이딩 하면서 문장의 지역적 정보를 반영

Recursive Neural Networks

- (The country) + (of my birth)의 수식을 받는 구조
- ‘the country’, ‘country of’, ‘of my’보다는 응집성이 높은 표현



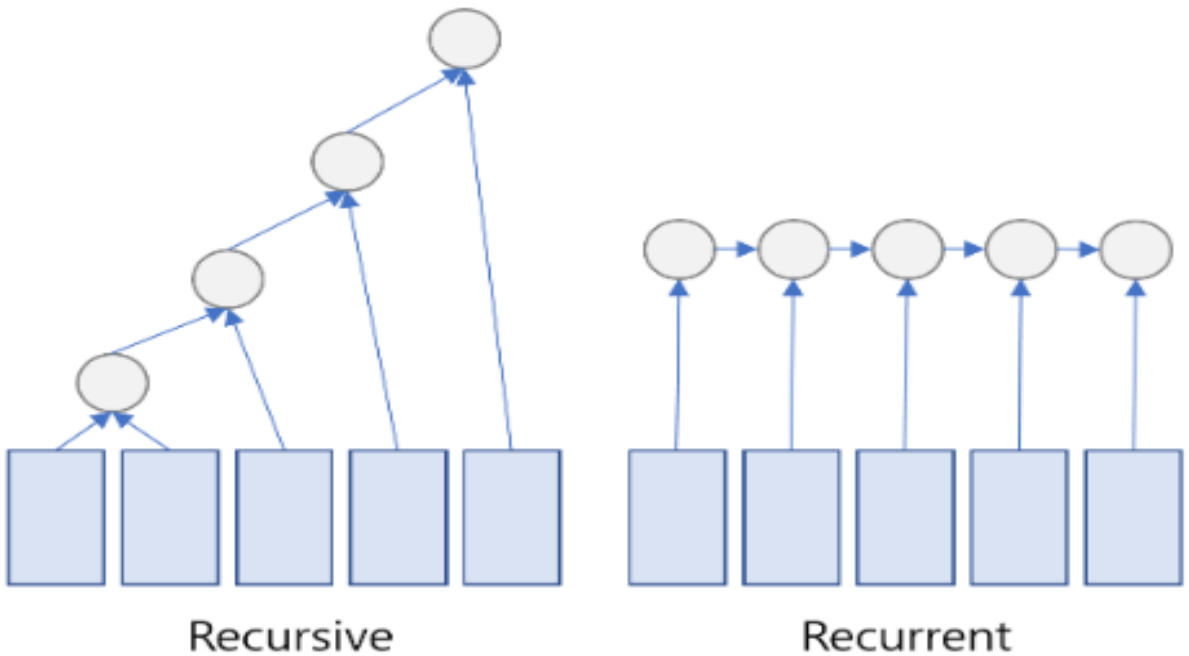


- 언어의 hierarchy한 성질을 네트워크 구조에 적극 차용
- https://ratsgo.github.io/deep%20learning/2017/04/03/recursive/

다만 RNN은 Recurrent Neural Networks나 CNN과 달리 트리 구조의 입력값을 반드시 필요로 합니다. 예컨대 아래와 같은데요. 이런 구조의 데이터를 생성하려면 대단히 많은 시간과 비용을 들여야 하는데다 계산도 복잡하기 때문에 RNN이 CNN이나 Recurrent Neural Networks에 비해 주목을 덜 받는 경향이 있는 것 같습니다.

> (((The)(actors))(((are)(fantastic))(.)))

마지막으로 Recurrent Neural Networks는 Recursive Neural Networks의 특수 케이스라는 점을 짚어보겠습니다. 만약 Recursive Neural Networks가 모든 지역정보를 순서대로 빠짐없이 반영한다고 하면 아래와 같이 구조를 그릴 수 있는데요, 이를 각도 회전해놓고 보면 본질적으로 Recurrent Neural Networks와 같습니다.



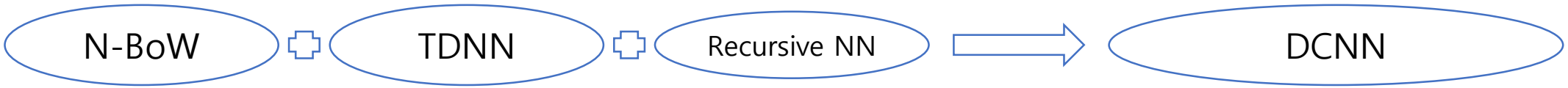
2. Background

DCNN은 Pooling operation 에 뒤따르는 Convolution operation 에 의해 구성

1. Related neural sentence models
2. 1-d convolution
3. TDNN (Network에 Max-pooling을 얹으면 Sentence Model이 될 수 있다.

2-1. Related Neural Sentence Models

- 가장 기본: N-BoW models
- 외부 Parse Tree 이용: Recursive Neural Network
- RNN이 Recursive NN의 simpler one
- Convolution operation 과 TDNN 구조



2.2 Convolution

Vector of weights $\mathbf{m} \in \mathbb{R}^m$ (Filter of the convolution)

Vector of inputs as a sequence $\mathbf{s} \in \mathbb{R}^s$ (ith word single feature value)

$$\mathbf{c}_j = \mathbf{m}^\intercal \mathbf{s}_{j-m+1:j}$$

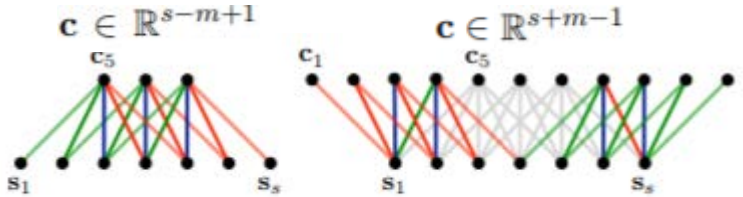


Figure 2: Narrow and wide types of convolution. The filter $\mathbf{m}$ has size $m = 5$.

Narrow Convolution : Requires s >= m, 즉 입력 벡터가 반드시 Filter보다 커야한다.

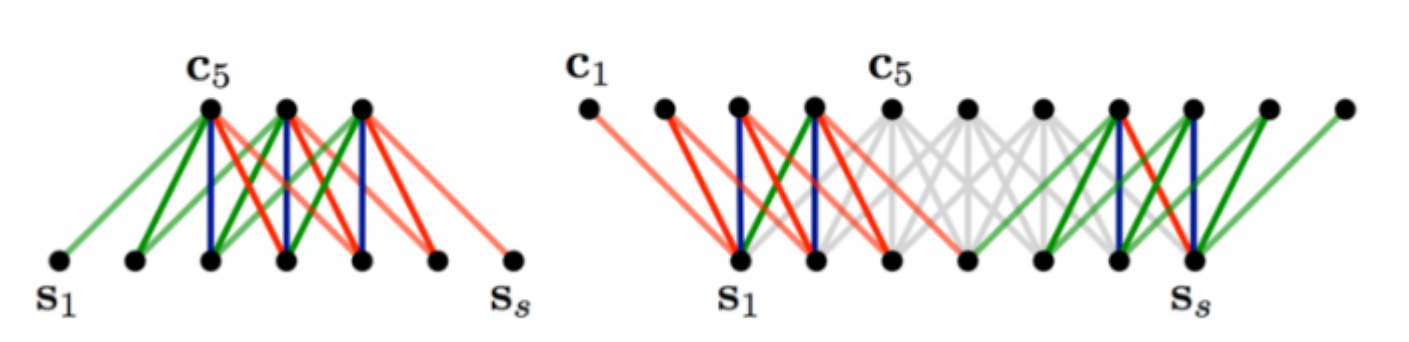
Wide Convolution: Doesn't requires s or m in specific length

Wide의 장점:

- 모든 가중치들이 전체 문장에 도달할 수 있도록 한다. Margin을 포함하여 /이는 Filter size가 상대적으로 클 때 유용
- Non-empty result c

Narrow vs. Wide convolution

위에서 컨볼루트를 설명할 때 필자는 필터를 적용하는 방법에 대해 약간의 부분을 무시했습니다. 행렬 중심에 3 × 3 필터를 적용하면 문제가 없지만 가장자리는 어떻게됩니까? 위와 왼쪽에 인접한 요소가없는 행렬의 첫 번째 요소에 필터를 어떻게 적용할까요? 이때, 제로 패딩을 사용할 수 있습니다. 행렬 외부에 있는 모든 요소는 0이됩니다. 이렇게하면 필터를 입력 행렬의 모든 요소에 적용하고 더 크거나 동일한 크기의 출력을 얻을 수 있습니다. 제로 패딩을 추가하는 것은 wide 컨볼루션이라고도하며, 반면 패딩을 사용하지 않는 것은 narrow 컨볼루션이됩니다. 1D의 예제는 다음과 같습니다.



input 크기와 관련하여 큰 필터가있을 때 와이드 컨볼루션이 유용하거나 심지어 필요한 경우를 볼 수 있습니다. 위의 경우, 좁은 컨볼 루션은 크기 (7-5) + 1 = 3의 출력을 가져 오며, 폭이 큰 컨볼루션은 크기 (7 + 2 * 4-5) + 1 = 11의 출력을 산출합니다.

https://m.blog.naver.com/rkdwnsdud555/221222217300

2.3 Time-Delay Neural Networks

Sequence s 는 time dimension을 갖는다.

$\mathbf{s}_j$ 는 vector of d $\mathbf{s} \in \mathbb{R}^{d \times s}$.

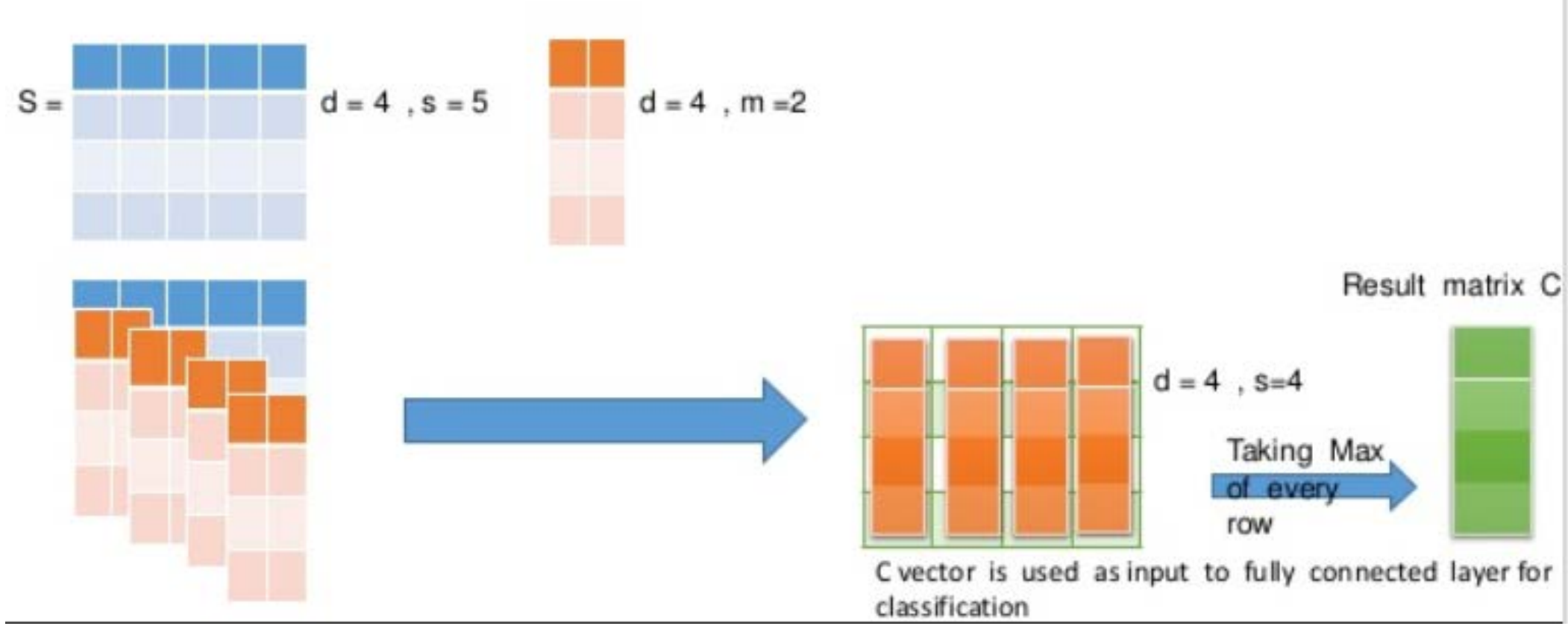
$$\mathbf{s} = \begin{bmatrix} | & | & | \\ \mathbf{w}_1 & \dots & \mathbf{w}_s \\ | & | & | \end{bmatrix} \times M = \mathbf{c}_{max} = \begin{bmatrix} \max(\mathbf{c}_{1,:}) \\ \vdots \\ \max(\mathbf{c}_{d,:}) \end{bmatrix}$$

No dependencies in external language-specific features

Narrow convolution을 사용하여 Margin(상대적 불필요) 정보를 제거하지만 오히려 neglec할 수도 있다.

Max-pooling은 Pooling값의 상대적 위치를 제공해주지 못한다.

Narrow one dim . convolution through time axis



해당 논문에서는 wide-convolution을 *k-max pooling* 을 사용하는 dynamic pooling layer로 대체한다. 이 모델의 input이 가변 길이의

3. Convolutional Neural Networks with Dynamic k-max Pooling

우리는 Wide convolutional Layers에 이어
Dynamic k-max pooling layer 사용

CNN for Modelling Sentences

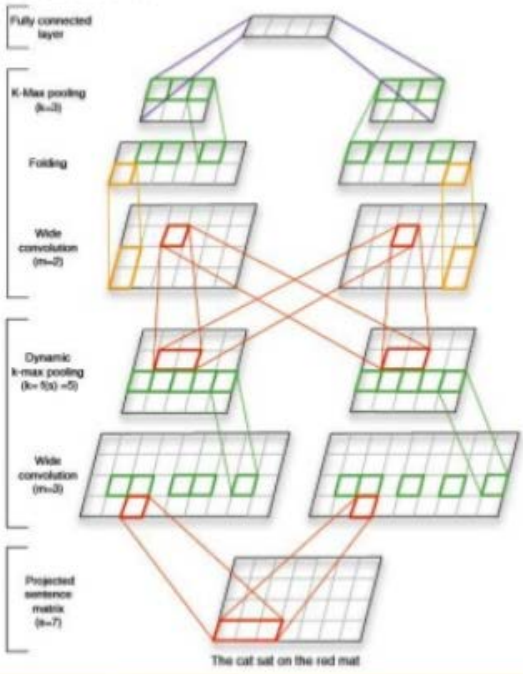
- Blunsom, Phil, Edward Grefenstette, and Nal Kalchbrenner. "**A Convolutional Neural Network for Modelling Sentences**." ACL 2014.

- Word vector: learning
- Supervisor depends on tasks
- **Dynamic k-Max Pooling**

$$k_l = \max(k_{top}, \lceil \frac{L-l}{L} s \rceil)$$

- Folding
  - Sums every two rows
- Multiple Feature Maps
  - Different features

- Word order, n-gram
- Sentence Feature
- Many tasks



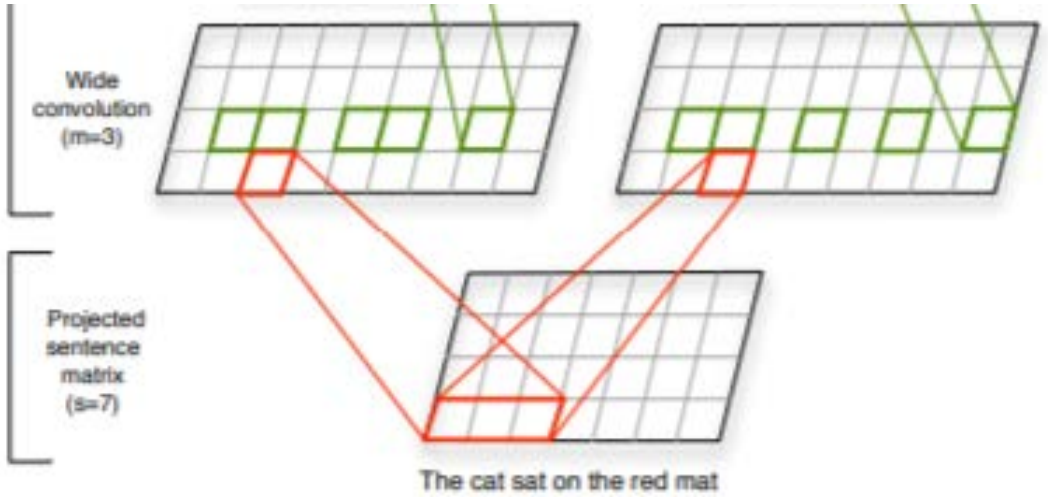
3.1 Wide Convolution

주어진 input 문장에 대해서 각 단어의 Embedding 값을 구한다

(w_i는 파라미터로서 Optimized during training) $\mathbf{w}_i \in \mathbb{R}^d$

만들어진 Embedding Vector를 Concat 해 문장 Matrix s를 만든다. $\mathbf{s} \in \mathbb{R}^{d \times s}$

$$\mathbf{s} \in \mathbb{R}^{d \times s} * \mathbf{m} \in \mathbb{R}^{d \times m} = \mathbf{c} \quad d \times (s + m - 1)$$



3.2 k-Max Pooling

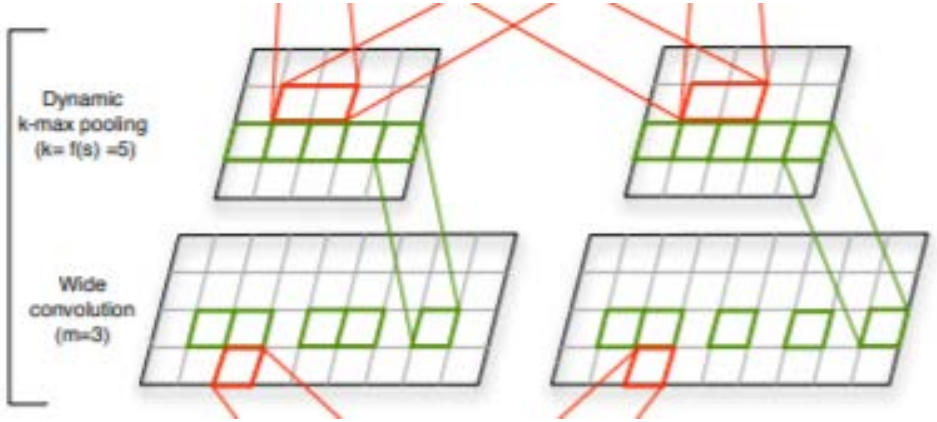
일반적인 Pooling과 다르다.

p>=k인 dimension을 갖는 sequence vector p에 대해

K개의 최대값을 선택해 subsequence를 만든다.

K-max pooling은 위치가 가까운 것에 신경 쓰지않고 p에서 k개의 active 한 값을 뽑아낸다.

연산 후에는 각 feature의 정확한 위치 정보는 손실되지만 feature들 간의 순서 정보는 보존



3.3 Dynamic k-Max Pooling

Dynamic은 k가 유동적으로 변한다는 뜻 (k 자체가 function!)

문장의 길이와 네트워크 총 깊이(전체 레이어의 수)에 의해 결정

If s = 문장 길이, L = 네트워크 깊이, 소문자 l = 현재 네트워크

then

$$k_l = \max(k_{top}, \lceil \frac{L-l}{L} s \rceil)$$

$$k_1 = \max(3, \lceil \frac{3-1}{3} \cdot 18 \rceil) = 12$$

$$k_2 = \max(3, \lceil \frac{3-2}{3} \cdot 18 \rceil) = 6$$

$$k_3 = k_{\max} = 3$$

Ex) K(top)은 가장 높은 층의 convolutional layer를 위한 고정된 Parameter(?)

K(1) / K(2) / k(3) 1,2,3층의 k

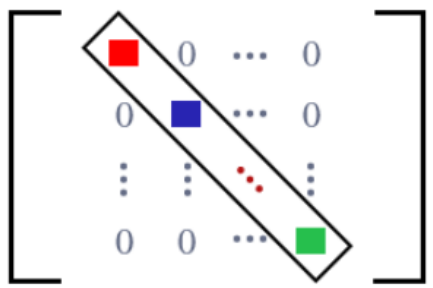
K값은 층이 올라갈 수록 수가 적어지는 구조인데 층이 낮은 곳보다 높은 곳에서 더 중요한

Feature를 고르는 과정으로 이해할 수 있다.

3.4 Non-linear Feature Function

Pooling이후, bias와 non-linear function (g) 가 적용된다.

$$\mathbf{M} = [\text{diag}(\mathbf{m}_{:,1}), \ldots, \text{diag}(\mathbf{m}_{:,m})]$$



$$a = g\left(\mathbf{M}\begin{bmatrix}\mathbf{w}_j \\ \vdots \\ \mathbf{w}_{j+m-1}\end{bmatrix} + \mathbf{b}\right)$$

3.5 Multiple Feature Maps

지금까지

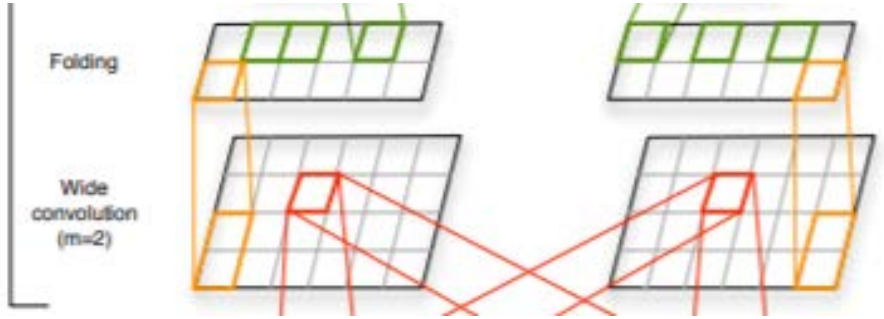
Wide Convolution → (dynamic) k-max pooling → non-linear function

을 통해 first order feature map을 얻어내었다. F_n^i

$$\mathbf{F}_j^i = \sum_{k=1}^{n} \mathbf{m}_{j,k}^i * \mathbf{F}_k^{i-1}$$

3.6 Folding

Fully connected layer에 도달하기 전, 다른 행의 Feature detectors는 각기 독립적이다.
위에서 살펴본 M을 대각 행렬이 아닌 Full 행렬로 만들면 이를 해소할 수 있지만 더 쉬운 방법이 Folding이다.



Folding : d행이 있을 경우 2개를 하나로 접어 d/2행으로 만든다.

4. Properties of the Sentence Model

DCNN의 특징

4.1 Word and n-gram order

4.2 Induced Feature Graph

4.1 Word and n-Gram Order

Input sentence의 단어 순서들에 민감하다.

잘 정제된 Feature detector를 만들기 위해:

- 1. 특정한 n-gram이 나타나는지
- 2. 연관성이 큰 n-gram의 관계

} 이 두 요점이 필요

DCNN으로 나타난 결과는 n-gram의 순서와 상대적 순서를 유지한다.

4.2 Induced Feature Graph

Sentence Model은 Internal이나 External structure를 사용하는데 DCNN은 Internal

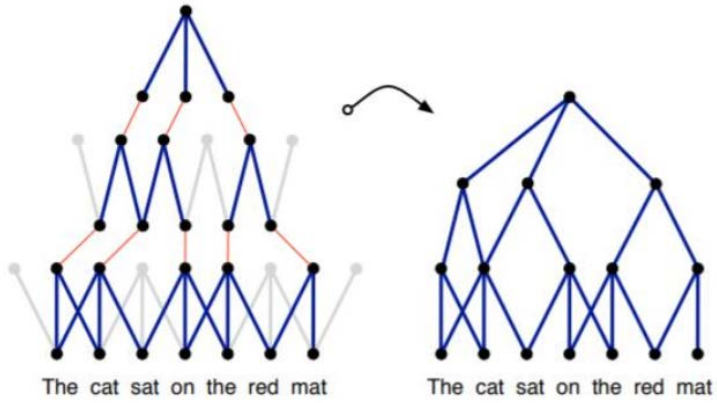
Pooling Operation에 사용되지 않은 Node는 Drop

Remaining Node가 Final Node를 구성하게 된다.

논문의 Fig 1. 이 Induced Feature Graph이다.

Folding layer가 없다면 모든 sub-graph들이 Root node에서 만나지만

Folding layer를 통해 sub-graph들이 중간에 결합할 수 있도록 한다.



5. Experiments

5.1 Training

각 실험 마다, Network의 최상층은 Softmax가 붙은 Fully connected layer

Network는 Cross-entropy를 감소시키기 위하여 훈련되며 L2 정규화를 포함
Mini-batch로 훈련되고 역전파가 시행
Adagrad update rule로 gradient based optimisation을 진행

Fast Frontier Transforms를 통하여 1차 선형 Convolution을 빠르게 입력할 수 있다.

GPU 사용

5.2 Sentiment Prediction in Movie Reviews

Stanford Sentiment Treebank Dataset으로 영화 리뷰 감성 분석 진행

1. Binary
2. Multiple 5 outcomes →

3. Neural 기반 분류기인 Max-TDNN, NBoW, DCNN outperform!

Sentiment Prediction in the movie reviews dataset

Classifier	Fine-grained (%)	Binary (%)
NB	41.0	81.8
BiNB	41.9	83.1
SVM	40.7	79.4
RecNTN	45.7	85.4
Max-TDNN	37.4	77.1
NBoW	42.4	80.5
DCNN	**48.5**	**86.8**

Neural

Max-TDNN : filter 6 (narrow) / Convolution → non-linearity → Max pooling → softmax classification

NBoW : sums the word vectors → non-linearity → softmax classification

DCNN : wide convolution → folding → dynamic k-max pooling → non-linearity

Filter 7 , 5 / k-max pooling k 4 /...

5.3 Question Type Classification

TREC의 질문 Dataset은 6개의 질문 유형 포함 (장소, 사람, 수치 정보)

Neural이 Non-Neural보다 우수한 성능

5.2 binary classifier와 비슷한 하이퍼 파라미터를 사용하고
Dataset이 작기 때문에 word-vector의 차원을 32로 지정.

- 사실 DCNN과 다른 분류기와의 차이가 유의하지 않다($p < 0.09$)
- Training set 자체로 Train하고 Test했으니 최신 기술의 성능과 비슷

Six-Way Question Classification on TREC

Classifier	Features	Acc. (%)
HIER	unigram, POS, head chunks NE, semantic relations	91.0
MAXENT	unigram, bigram, trigram POS, chunks, NE, supertags CCG parser, WordNet	92.6
MAXENT	unigram, bigram, trigram POS, wh-word, head word word shape, parser hypernyms, WordNet	93.6
SVM	unigram, POS, wh-word head word, parser hypernyms, WordNet 60 hand-coded rules	95.0
MAX-TDNN	unsupervised vectors	84.4
NBOW	unsupervised vectors	88.2
DCNN	unsupervised vectors	93.0

5.4 Twitter Sentiment Prediction with Distant Supervision

Tweets들을 모아모아 실험 진행
Label은 Tweet에 나타난 이모티콘을 통해 달았다.

Training set - 1.6 million // Test 400 hand annotated(주석)

Go et al.(2009) 방식대로 전처리를 진행했으며, 소문자화 진행
(e.g) huuuuuuungry → hungry로 단일 처리
DCNN Model은 영화평 감성분석(binary)에 사용된 모델

Non-neural n-gram based classifer보다 DCNN의 놀라운 성능

Twitter Sentiment Prediction

Classifier	Accuracy (%)
SVM	81.6
BINB	82.7
MAXENT	83.0
MAX-TDNN	78.8
NBOW	80.9
DCNN	**87.4**

N-BoW와 DCNN의 성능 차이로 보아 긴 n-gram과 hierarchical combine은 크게 도움이 된다고 판단할 수 있다.

5.4 Visualizing Feature Detectors

Feature Detectors

Binary Movie review Feature detectors

			POSITIVE								'NOT'		
lovely	comedic	moments	and	several	fine	performances	n't	have	any	huge	laughs	in	its
good	script	,	good	dialogue	,	funny	no	movement	,	no	,	not	much
sustains	throughout	is	daring	,	inventive	and	n't	stop	me	from	enjoying	much	of
well	written	,	nicely	acted	and	beautifully	not	that	kung	pow	is	n't	funny
remarkably	solid	and	subtly	satirical	tour	de	not	a	moment	that	is	not	false
			NEGATIVE								'TOO'		
,	nonexistent	plot	and	pretentious	visual	style	,	too	dull	and	pretentious	to	be
it	fails	the	most	basic	test	as	either	too	serious	or	too	lighthearted	,
so	stupid	,	so	ill	conceived	,	too	slow	,	too	long	and	too
,	too	dull	and	pretentious	to	be	feels	too	formulaic	and	too	familiar	to
hood	rats	butt	their	ugly	heads	in	is	too	predictable	and	too	self	conscious

Figure 4: Top five 7-grams at four feature detectors in the first layer of the network.

DCNN filter는 특정 단어들이 input으로 들어왔을 때의 active한 Neuron of Feature detector

첫 Layer는 입력 문장의 n-gram이지만 Layer가 높아질 수록 분리된 n-gram들이다.

Filter size가 7이므로, 288 feature detector 중 7-gram들의 랭크를 표에 담았다.

단일의 n-gram 학습을 넘어, syntactic, semantic, structural 중요도가 있는 n-gram 패턴을 찾을 수 있도록 학습된다.