A Convolutional Neural Network for Modelling Sentences

Nal Kalchbrenner Edward Grefenstette
Phil Blunsom

Department of Computer Science, Oxford University

Overview of Model

Represent sentences by extracting more abstract features

Input: sequence of word embeddings

Output: classification probabilities

Each layer involves

1. Convolution
2. Dynamic k-Max Pooling
3. Apply a non-linearity (tanh)

One-Dimensional Convolution

1. The *filter* $\mathbf{m} \in \mathbb{R}^m$
2. The sequence $\mathbf{s} \in \mathbb{R}^s$

Returns sequence $\mathbf{c} \in \mathbb{R}^{s-m+1}$

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

Takes a dot product between length m subsequences of $\mathbf{s}$ and the filter $\mathbf{m}$

Wide convolution pads $\mathbf{s}$ with $m - 1$ zeros on the left.

Convolution with Word Embeddings

Assume word embeddings of dimension d

Filter $\mathbf{m}$ will be in $\mathbb{R}^{d \times m}$

Sequence $\mathbf{s}$ will be in $\mathbb{R}^{d \times s}$

Each row of $\mathbf{m}$ will be convolved with the corresponding row of $\mathbf{s}$

k-Max Pooling (LeCun et al.)

Given k and sequence $\mathbf{p} \in \mathbb{R}^p$, $p \geq k$

1. Return k largest elements of $\mathbf{p}$
2. Keep elements in their original order

Denoted $\mathbf{p}^k_{max} \in \mathbb{R}^k$

Dynamic k-Max Pooling

"Smooth extraction of higher-order features"

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

- k_{top} is fixed parameter
- l is current layer
- L is total number of layers
- s is sentence length

Folding

Elementwise sum of pairs rows of a matrix

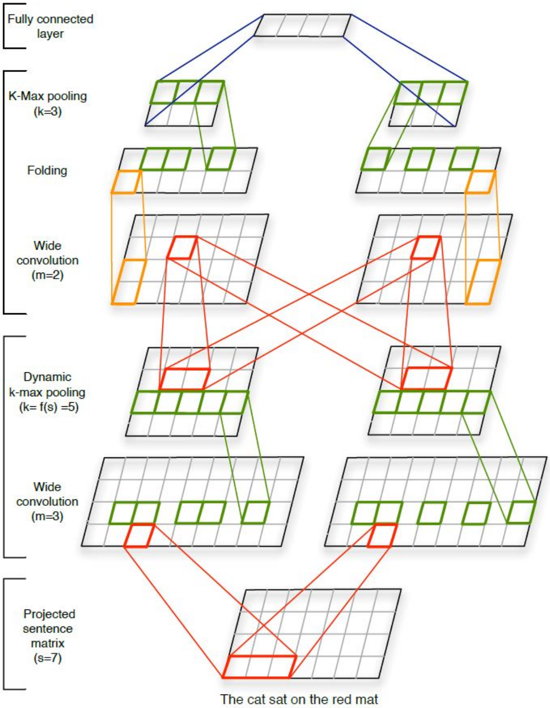
$$f : \mathbb{R}^{d \times n} \to \mathbb{R}^{d/2 \times n}$$

$f(M) = N$ where

$$N[i, j] = M[2i, j] + M[2i + 1, j],$$

$i = 0, ..., d/2 - 1$, $j = 0, ...n - 1$

- Introduces dependencies between different feature rows
- No added parameters


Fully connected layer
K-Max pooling (k=3)
Folding
Wide convolution (m=2)
Dynamic k-max pooling (k= f(s) =5)
Wide convolution (m=3)
Projected sentence matrix (s=7)
The cat sat on the red mat

Size of Network

Model	First Layer		Second Layer		
*	Width	Filters	Width	Filters	k-top
Binary	7	6	5	14	4
Multi-class	10	6	7	12	5

Training

Top layer is soft-max nonlinearity to predict probability distribution

L_2 regularization of parameters in objective function

Parameters are word embeddings, filter weights, & fully connected layers

Trained using Adagrad with mini-batches

"Processes multiple millions of sentences per hour on one GPU"

Experiments

1. Predicting sentiment of movie reviews - binary (Socher et al. 2013)
2. Predicting sentiment of movie reviews - multi-class (Socher et al. 2013)
3. Categorization of questions (Li and Roth 2002)
4. Sentiment of Tweets, labels based on emoticons(Go et al. 2009)

Feature embedding dimensionality chosen based on size of dataset

Movies accuracy

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**

First layer feature-detectors

POSITIVE

lovely	comedic	moments	and	several	fine	performances
good	script	,	good	dialogue	,	funny
sustains	throughout	is	daring	,	inventive	and
well	written	,	nicely	acted	and	beautifully
remarkably	solid	and	subtly	satirical	tour	de

NEGATIVE

,	nonexistent	plot	and	pretentious	visual	style
it	fails	the	most	basic	test	as
so	stupid	,	so	ill	conceived	,
,	too	dull	and	pretentious	to	be
hood	rats	butt	their	ugly	heads	in

'NOT'

n't	have	any	huge	laughs	in	its
no	movement	,	no	,	not	much
n't	stop	me	from	enjoying	much	of
not	that	kung	pow	is	n't	funny
not	a	moment	that	is	not	false

'TOO'

,	too	dull	and	pretentious	to	be
either	too	serious	or	too	lighthearted	,
too	slow	,	too	long	and	too
feels	too	formulaic	and	too	familiar	to
is	too	predictable	and	too	self	conscious

TREC 6-way classification accuracy

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

Twitter sentiment

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

Conclusion

Dynamic Convolutional Neural Networks

- Convolutions apply function to n-grams
- Dynamic k-max pooling extracts most active feature, and chooses k based on layer and sentence length
- Composing these two operations can be seen as feature detection
- Outperformed/stayed competitive with other neural approaches, baseline models, and state-of-the-art approaches without needing handcrafted features