Advanced Human Language Technologies

Convolutional Neural Networks

UPC
UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona
FIB

Outline

Motivation

Limitations of RNNs

- Long-distance relationships are faded due to vanishing gradient.
- Final representation carries to much weight for last words.
- Computations are no parallelizable since each word depends on the previous one.

Idea of CNNs

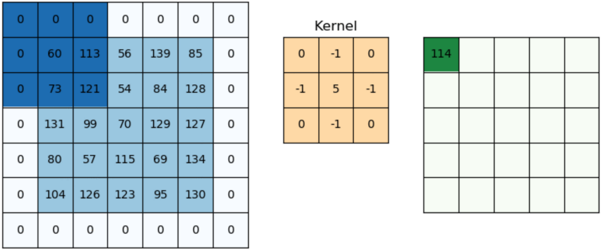
- Group subsequences of n-words
- Compute a vector for each subsequence
- Combine the obtained vectors in a global representation

CNNs introduce parallelism, and weight for distant words is not vanished

Outline

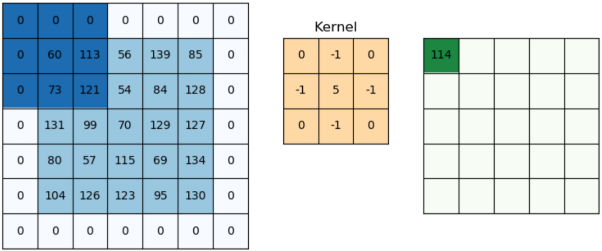
Outline

Discrete Convolutions



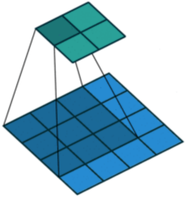
- Slide a small filter matrix (kernel, a.k.a filter) over the input matrix.
- At each position, compute the product of kernel and input values, and add them together.
- The output matrix is the concatenation of the application of the filter over the input matrix.
- kernel weights are *trained*

Discrete Convolutions

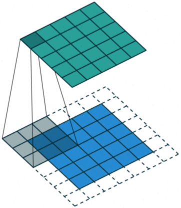


- Convolutional layers detect local features.
- Detection is invariant to feature position.
- Stacked convolutional layers detect more elaborate hierarchical features
- Application domains:
    - Text: Sequence of tokens - 1D convolution
    - Images: Matrix of pixels - 2D convolution
    - Video: Sequence of images - 3D convolution

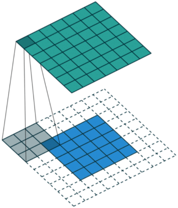
2D Discrete Convolutions



input size: 4×4
kernel size: 3
padding: 0
output size: 2×2



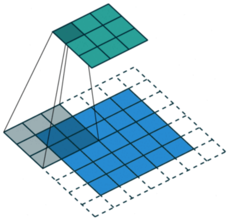
input size: 5×5
kernel size: 3
padding: 1
output size: 5×5



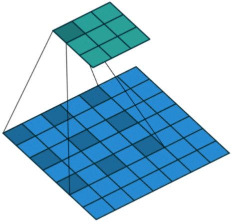
input size: 5×5
kernel size: 3
padding: 2
output size: 7×7

`https://raw.githubusercontent.com/vdumoulin/conv_arithmetic`

2D Discrete Convolutions



input size: 5×5
kernel size: 3
padding: 1
stride: 2
output size: 3×3

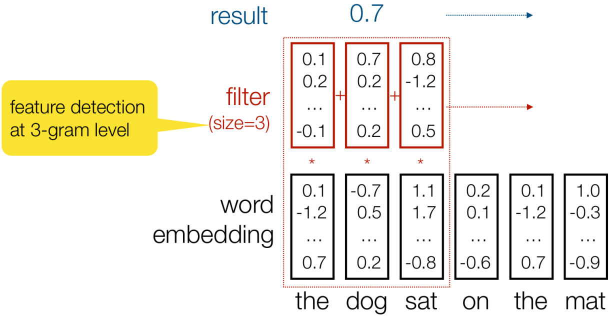


input size: 7×7
kernel size: 3
padding: 0
dilation: 2
output size: 3×3

https://raw.githubusercontent.com/vdumoulin/conv_arithmetic

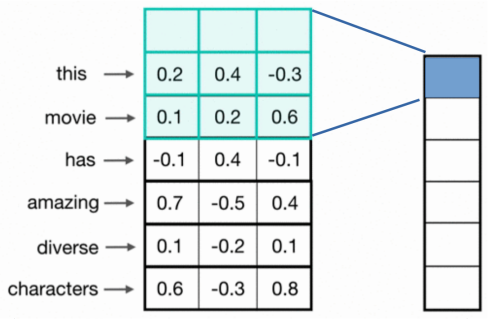
1D Discrete Convolutions

- Text is a sequence of tokens, each represented by a vector (e.g. embedding)
- We can see the sentence as a matrix, and use convolutions
- However, it does not make senses to capture pieces of an embedding, so the kernel will have fixed width (the embeding dimension)

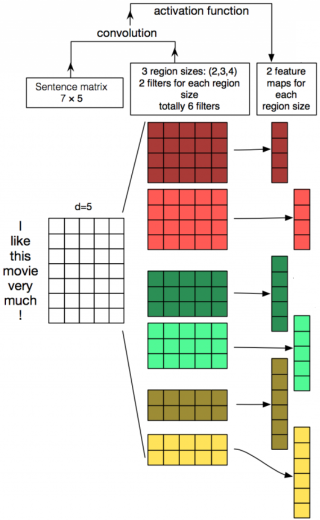


1D Discrete Convolutions

Embedding dim.: $d = 3$
Input size: $6 \times d$
Kernel size: 3
Padding: 1
Output size: 6



Multiple kernels



- We can use several kernels, with different sizes, padding, and stride
- Each kernel will learn to encode a different feature

https://github.com/m-peker/CNN-with-different-kernel-size-for-sentence-classification

1D convolution for text

Vanilla convolution

tentative	0.2	0.1	−0.3	0.4
deal	0.5	0.2	−0.3	−0.1
reached	−0.1	−0.3	−0.2	0.4
to	0.3	−0.3	0.1	0.1
keep	0.2	−0.3	0.4	0.2
government	0.1	0.2	−0.1	−0.1
open	−0.4	−0.4	0.2	0.3

t,d,r	−1.0
d,r,t	−0.5
r,t,k	−3.6
t,k,g	−0.2
k,g,o	0.3

Apply a **filter** (or **kernel**) of size 3

3	1	2	−3
−1	2	1	−3
1	1	−1	1

Source: Stanford cs224n course 2019. Chris Manning

1D convolution for text

With padding = 1

Ø	**0.0**	**0.0**	**0.0**	**0.0**
tentative	0.2	0.1	−0.3	0.4
deal	0.5	0.2	−0.3	−0.1
reached	−0.1	−0.3	−0.2	0.4
to	0.3	−0.3	0.1	0.1
keep	0.2	−0.3	0.4	0.2
government	0.1	0.2	−0.1	−0.1
open	−0.4	−0.4	0.2	0.3
Ø	**0.0**	**0.0**	**0.0**	**0.0**

Ø,t,d	−0.6
t,d,r	−1.0
d,r,t	−0.5
r,t,k	−3.6
t,k,g	−0.2
k,g,o	0.3
g,o,Ø	−0.5

Apply a **filter** (or **kernel**) of size 3

3	1	2	−3
−1	2	1	−3
1	1	−1	1

Source: Stanford cs224n course 2019. Chris Manning

1D convolution for text

With padding $= 1$, kernels $= 3$

Ø	**0.0**	**0.0**	**0.0**	**0.0**
tentative	0.2	0.1	−0.3	0.4
deal	0.5	0.2	−0.3	−0.1
reached	−0.1	−0.3	−0.2	0.4
to	0.3	−0.3	0.1	0.1
keep	0.2	−0.3	0.4	0.2
government	0.1	0.2	−0.1	−0.1
open	−0.4	−0.4	0.2	0.3
Ø	**0.0**	**0.0**	**0.0**	**0.0**

Ø,t,d	−0.6	0.2	1.4
t,d,r	−1.0	1.6	−1.0
d,r,t	−0.5	−0.1	0.8
r,t,k	−3.6	0.3	0.3
t,k,g	−0.2	0.1	1.2
k,g,o	0.3	0.6	0.9
g,o,Ø	−0.5	−0.9	0.1

Apply 3 **filters** of size 3

3	1	2	−3
−1	2	1	−3
1	1	−1	1

1	0	0	1
1	0	−1	−1
0	1	0	1

1	−1	2	−1
1	0	−1	3
0	2	2	1

Source: Stanford cs224n course 2019. Chris Manning

1D convolution for text

With padding $= 1$, kernels $= 3$, stride $= 2$

Ø	0.0	0.0	0.0	0.0
tentative	0.2	0.1	−0.3	0.4
deal	0.5	0.2	−0.3	−0.1
reached	−0.1	−0.3	−0.2	0.4
to	0.3	−0.3	0.1	0.1
keep	0.2	−0.3	0.4	0.2
government	0.1	0.2	−0.1	−0.1
open	−0.4	−0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Ø,t,d	−0.6	0.2	1.4
d,r,t	−0.5	−0.1	0.8
t,k,g	−0.2	0.1	1.2
g,o,Ø	−0.5	−0.9	0.1

Apply 3 **filters** of size 3

3	1	2	−3
−1	2	1	−3
1	1	−1	1

1	0	0	1
1	0	−1	−1
0	1	0	1

1	−1	2	−1
1	0	−1	3
0	2	2	1

Source: Stanford cs224n course 2019. Chris Manning

1D convolution for text

With padding = 1, kernels = 3, dilation = 2

Ø	0.0	0.0	0.0	0.0
tentative	0.2	0.1	−0.3	0.4
deal	0.5	0.2	−0.3	−0.1
reached	−0.1	−0.3	−0.2	0.4
to	0.3	−0.3	0.1	0.1
keep	0.2	−0.3	0.4	0.2
government	0.1	0.2	−0.1	−0.1
open	−0.4	−0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Apply 3 **filters** of size 3

3	1	2	−3
−1	2	1	−3
1	1	−1	1

1	0	0	1
1	0	−1	−1
0	1	0	1

1	−1	2	−1
1	0	−1	3
0	2	2	1

Ø,t,d	−0.6	0.2	1.4
t,d,r	−1.0	1.6	−1.0
d,r,t	−0.5	−0.1	0.8
r,t,k	−3.6	0.3	0.3
t,k,g	−0.2	0.1	1.2
k,g,o	0.3	0.6	0.9
g,o,Ø	−0.5	−0.9	0.1

1,3,5	0.3	0.0
2,4,6		
3,5,7		

2	3	1
1	−1	−1
3	1	0

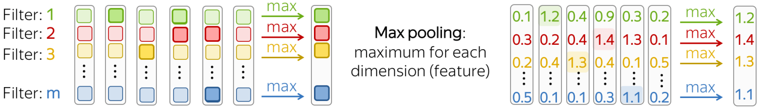
1	3	1
1	−1	−1
3	1	−1

Source: Stanford cs224n course 2019. Chris Manning

Outline

Pooling

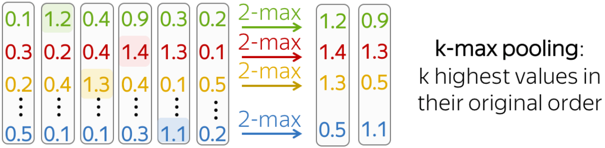
- Pooling is used after convolution to reduce complexity while capturing relevant information from previous layer
- A sliding window is moved along the convolution sequence, and a single value (e.g. maximum, average) is computed for each position.



https://lena-voita.github.io/nlp_course/models/convolutional.html

Pooling

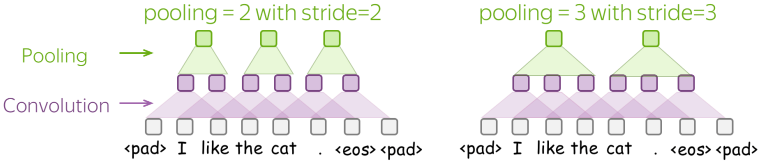
- Average pooling computes average of all positions instead of maximum.
- k-max pooling keeps the k highest values in the sequence, producing a k dimensions vector for each position, instead of a single value



https://lena-voita.github.io/nlp_course/models/convolutional.html

Pooling

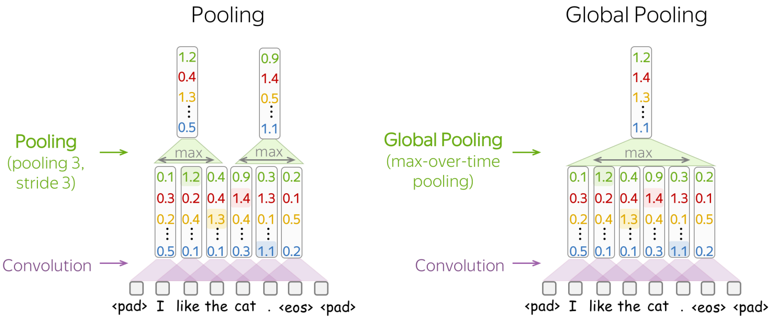
- The sliding window is moved along the sequence with a stride of the same size than the window, so there is no overlapping



https://lena-voita.github.io/nlp_course/models/convolutional.html

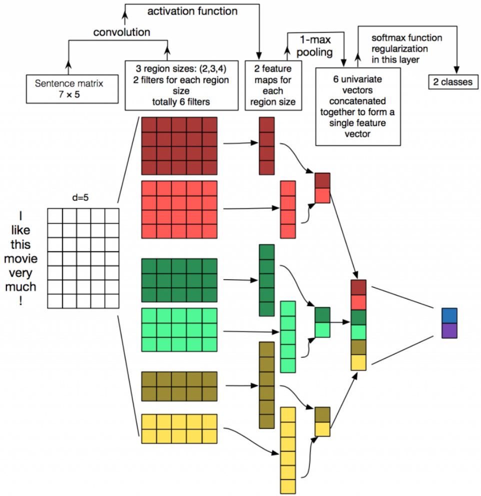
Pooling

- If the size of the window is the whole sequence, then we talk about global polling or max/average over time polling
- Note that not all sentences have the same lenght, so the window size for global polling is dynamic.



https://lena-voita.github.io/nlp_course/models/convolutional.html

Multiple kernels



- Using several kernels, we pool each one separately
- Results of each pooling are concatenated to form a feature vector used by the final layers

https://github.com/m-peker/CNN-with-different-kernel-size-for-sentence-classification

1D convolution for text

With padding = 1, kernels = 3, max pooling over time

Ø	0.0	0.0	0.0	0.0
tentative	0.2	0.1	−0.3	0.4
deal	0.5	0.2	−0.3	−0.1
reached	−0.1	−0.3	−0.2	0.4
to	0.3	−0.3	0.1	0.1
keep	0.2	−0.3	0.4	0.2
government	0.1	0.2	−0.1	−0.1
open	−0.4	−0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Ø,t,d	−0.6	0.2	1.4
t,d,r	−1.0	1.6	−1.0
d,r,t	−0.5	−0.1	0.8
r,t,k	−3.6	0.3	0.3
t,k,g	−0.2	0.1	1.2
k,g,o	0.3	0.6	0.9
g,o,Ø	−0.5	−0.9	0.1

max p	0.3	1.6	1.4

Apply 3 **filters** of size 3

3	1	2	−3
−1	2	1	−3
1	1	−1	1

1	0	0	1
1	0	−1	−1
0	1	0	1

1	−1	2	−1
1	0	−1	3
0	2	2	1

Source: Stanford cs224n course 2019. Chris Manning

1D convolution for text

With padding $= 1$, kernels $= 3$, avg pooling over time

∅	0.0	0.0	0.0	0.0
tentative	0.2	0.1	−0.3	0.4
deal	0.5	0.2	−0.3	−0.1
reached	−0.1	−0.3	−0.2	0.4
to	0.3	−0.3	0.1	0.1
keep	0.2	−0.3	0.4	0.2
government	0.1	0.2	−0.1	−0.1
open	−0.4	−0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

∅,t,d	−0.6	0.2	1.4
t,d,r	−1.0	1.6	−1.0
d,r,t	−0.5	−0.1	0.8
r,t,k	−3.6	0.3	0.3
t,k,g	−0.2	0.1	1.2
k,g,o	0.3	0.6	0.9
g,o,∅	−0.5	−0.9	0.1

ave p	−0.87	0.26	0.53

Apply 3 **filters** of size 3

3	1	2	−3
−1	2	1	−3
1	1	−1	1

1	0	0	1
1	0	−1	−1
0	1	0	1

1	−1	2	−1
1	0	−1	3
0	2	2	1

Source: Stanford cs224n course 2019. Chris Manning

1D convolution for text

With padding $= 1$, kernels $= 3$, stride $= 2$, local max pool

∅	0.0	0.0	0.0	0.0
tentative	0.2	0.1	−0.3	0.4
deal	0.5	0.2	−0.3	−0.1
reached	−0.1	−0.3	−0.2	0.4
to	0.3	−0.3	0.1	0.1
keep	0.2	−0.3	0.4	0.2
government	0.1	0.2	−0.1	−0.1
open	−0.4	−0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

Apply 3 **filters** of size 3

3	1	2	−3
−1	2	1	−3
1	1	−1	1

1	0	0	1
1	0	−1	−1
0	1	0	1

1	−1	2	−1
1	0	−1	3
0	2	2	1

∅,t,d	−0.6	0.2	1.4
t,d,r	−1.0	1.6	−1.0
d,r,t	−0.5	−0.1	0.8
r,t,k	−3.6	0.3	0.3
t,k,g	−0.2	0.1	1.2
k,g,o	0.3	0.6	0.9
g,o,∅	−0.5	−0.9	0.1
∅	−Inf	−Inf	−Inf

∅,t,d,r	−0.6	1.6	1.4
d,r,t,k	−0.5	0.3	0.8
t,k,g,o	0.3	0.6	1.2
g,o,∅,∅	−0.5	−0.9	0.1

Source: Stanford cs224n course 2019. Chris Manning

1D convolution for text

With padding = 1, kernels = 3, 2-max pool over time

Ø	0.0	0.0	0.0	0.0
tentative	0.2	0.1	−0.3	0.4
deal	0.5	0.2	−0.3	−0.1
reached	−0.1	−0.3	−0.2	0.4
to	0.3	−0.3	0.1	0.1
keep	0.2	−0.3	0.4	0.2
government	0.1	0.2	−0.1	−0.1
open	−0.4	−0.4	0.2	0.3
Ø	0.0	0.0	0.0	0.0

Ø,t,d	−0.6	0.2	1.4
t,d,r	−1.0	1.6	−1.0
d,r,t	−0.5	−0.1	0.8
r,t,k	−3.6	0.3	0.3
t,k,g	−0.2	0.1	1.2
k,g,o	0.3	0.6	0.9
g,o,Ø	−0.5	−0.9	0.1

2-max p	−0.2	1.6	1.4
	0.3	0.6	1.2

Apply 3 **filters** of size 3

3	1	2	−3
−1	2	1	−3
1	1	−1	1

1	0	0	1
1	0	−1	−1
0	1	0	1

1	−1	2	−1
1	0	−1	3
0	2	2	1

Source: Stanford cs224n course 2019. Chris Manning

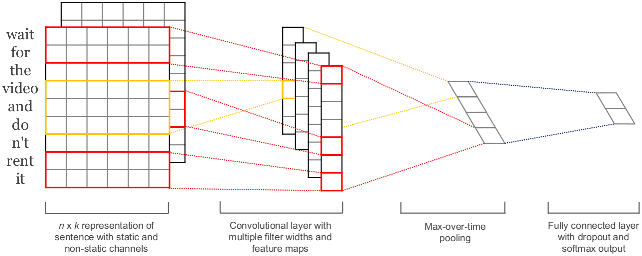
Outline

Sentence Classification

Sentence classification is a generic task that can be used for different goals, depending on the target classes.

- Sentiment analysis (positive/negative/neutral)
- Q&A: Question target identification (person/place/number/...)
- Relation extraction (e.g. Drug-drug interaction detection)
- Sentence similarity (equivalent/similar/unrelated/opposite)
- Textual entailment (yes/no)
- Textual inference (equivalent/entailment/unrelated/contradiction)
- ...

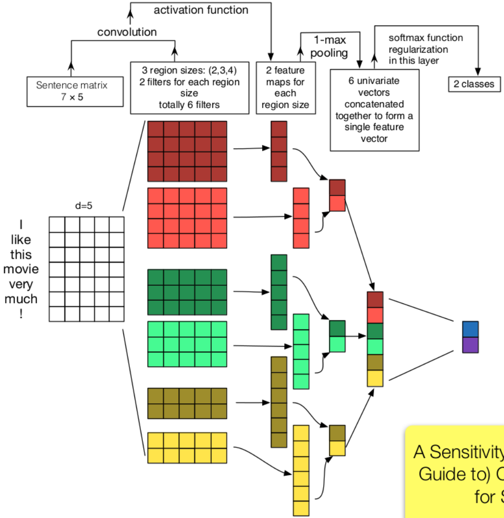
Sentence Classification



Convolutional Neural Networks for Sentence Classification

Kim, 2014

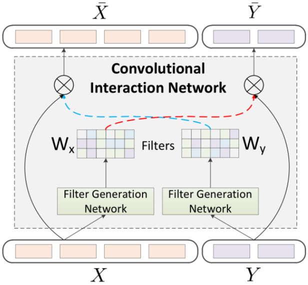
Sentence Classification

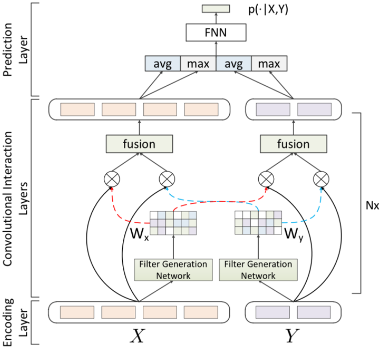


A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification

Zhang and Wallace, 2015

Textual Inference



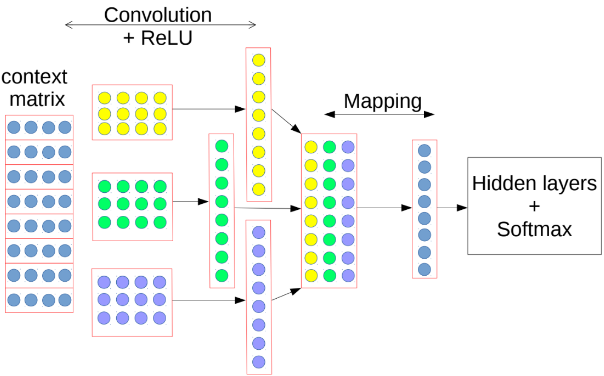


Convolutional Interaction Network for Natural Language Inference

Gong et al., 2018

Language Modeling

CNNs have been also used for language modeling



Convolutional Neural Network Language Models

Pham et al., 2016

Language Modeling

Motivation

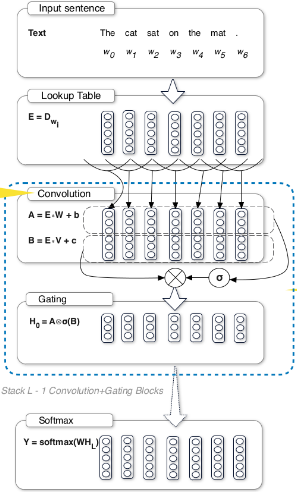
CNN basics

Application Examples

Conclusions

CNNs have been also used for language modeling

Use of padding to ensure causal dependencies



Gated Linear Unit (GLU)

Language modeling with gated convolutional networks

Dauphin et al., 2016

Outline

Conclusions

- Convolutions are local feature detectors.
- Convolutions are invariant to position.
- Stacked convolutions detect hierarchical features.
- Convolutions are fast
- Convolutions have been applied to multiple NLP tasks, including Machine Translation, Language Modeling, Text Classification, Word Representation, Textual Inference, etc.

Acknowledgements

- Slides in this session are based on images and ideas from lectures by
  - Noe Casas
  - Christopher Manning