Deep Learnig for NLP

Universität Bielefeld

Lecture 8 - Convolutional Neural Networks

Dr. Steffen Eger
steffen.eger@uni-bielefeld.de

Jonas Belouadi
jonas.belouadi@uni-bielefeld.de

Today

1. Convolution and pooling
2. Convolutional networks for NLP

- Lecture based on/inspired by:
  - Lecture Videos by Richard Socher and Andrej Karpathy

https://www.youtube.com/watch?v=vYJtZwoO9Rw

https://www.youtube.com/watch?v=AQirPKrAyDg

Motivation

Universität Bielefeld

- Problem 1: Variable-sized input
  standard MLP always expect the same input size

- Problem 2: Relevance of words
  “to” and “a” are not very informative, but content words like “kidnapping” are important for most tasks **independent of their position in the input**

- Problem 3: MLP may have too many parameters (“too complex models”) in certain situations

Idea of convolution

Universität Bielefeld

“A convolutional neural network is designed to identify ***indicative local predictors*** *in a large structure, and combine them to produce a fixed size vector representation of the structure, capturing these local aspects that are most informative for the prediction task at hand.“*

Yoav Goldberg

NLLG

Last lecture

Universität Bielefeld

- sentence embeddings
  - can easily be used for classification

- **Today:** use convolutional neural networks for sentence classification
  - a more "historical" approach (at least for NLP)

Convolution in image recognition

Example by Richard Socher:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix}$$

Image representation *f*

Filter *g*
(also known as kernel)

Convolved image representation

Convolved features

Apply the filter to the image, move over different filter regions:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Convolution operation

4	3	4
2	4	3
2	3	4

Convolved features!

Convolved features

Apply the filter to the image, move over different filter regions :

The task is to learn good filter weights!

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Convolution operation

4	3	4
2	4	3
2	3	4

Convolved features!

And in texts?

Universität Bielefeld

- Sentiment classification
  - The **movie** was **really good**.
  - We saw this **really good movie**.
  - The **movie**, which we saw yesterday with all the colleagues in this tiny movie theatre next to the bridge, was (despite my expectations) **really good**.

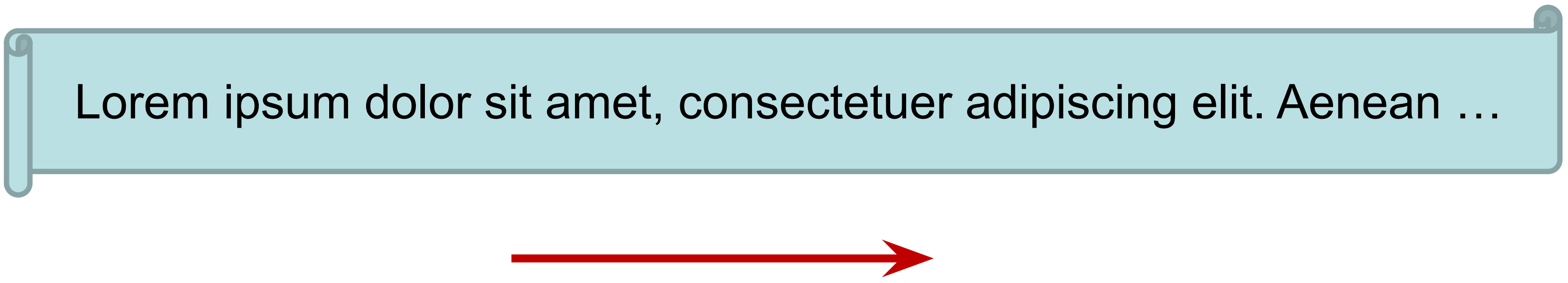
→ For this task, position information does not really matter.

Dimensionality

- Advantages of text flow:
  - Usually only one dimension

  → as opposed to two dimensions (or even three) in images

- one-dimensional convolutional networks are sometimes also called **time-delay neural networks** (TDNN)



The 1d convolution operation

Universität Bielefeld

$$(f * g)[i] = \sum_{m=-M}^{M} f[i-m]g[m]$$

9	7	2	4	8	7	3	1	5	9	8	4

- $*$ is the convolution operator
- f is the input representation
- i is the current position in the input representation
- M is the window size
- $g[m]$ is the weight for an input element with distance m to the current input
  → g it is also often referred to as the *filter* (or *kernel*)

- Careful! You will find many terminological alternatives in the literature:
  - *w* (for weights) instead of *g*, *n* or *t* (for time) instead of *i*, *a* (for age) instead of *m*, …

Convolutional Layer in NLP

Universität Bielefeld

- Input sentence $\boldsymbol{x}_{1:n}$ is a concatenation of the word vectors $\boldsymbol{x}_i \in \boldsymbol{R}^d$
- Convolutional filter: $\boldsymbol{w} \in \boldsymbol{R}^{hd}$

M = 1
$d = 2$

(0.8)

$\binom{1}{1}$ $\binom{7}{3}$ $\binom{2}{5}$ $\binom{1}{9}$ $\binom{3}{3}$ $\binom{1}{1}$

PAD The movie was awesome PAD

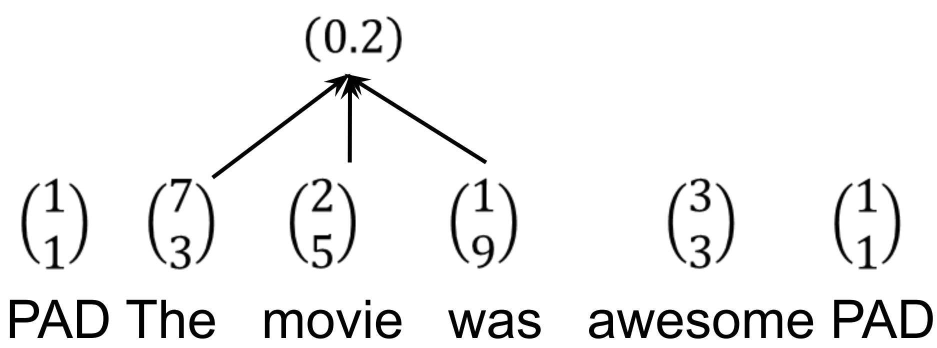
NLLG

Convolutional Layer in NLP

Universität Bielefeld

- Input sentence $\boldsymbol{x}_{1:n}$ is a concatenation of the word vectors $\boldsymbol{x}_i \in \boldsymbol{R}^d$
- Convolutional filter: $\boldsymbol{w} \in \boldsymbol{R}^{hd}$

M = 1
$d = 2$



Convolutional Layer in NLP

- Input sentence $\boldsymbol{x}_{1:n}$ is a concatenation of the word vectors $\boldsymbol{x}_i \in \boldsymbol{R}^d$
- Convolutional filter: $\boldsymbol{w} \in \boldsymbol{R}^{hd}$

$M = 1$
$d = 2$

$$(0.6)$$

$$\begin{pmatrix}1\\1\end{pmatrix} \quad \begin{pmatrix}7\\3\end{pmatrix} \quad \begin{pmatrix}2\\5\end{pmatrix} \quad \begin{pmatrix}1\\9\end{pmatrix} \quad \begin{pmatrix}3\\3\end{pmatrix} \quad \begin{pmatrix}1\\1\end{pmatrix}$$

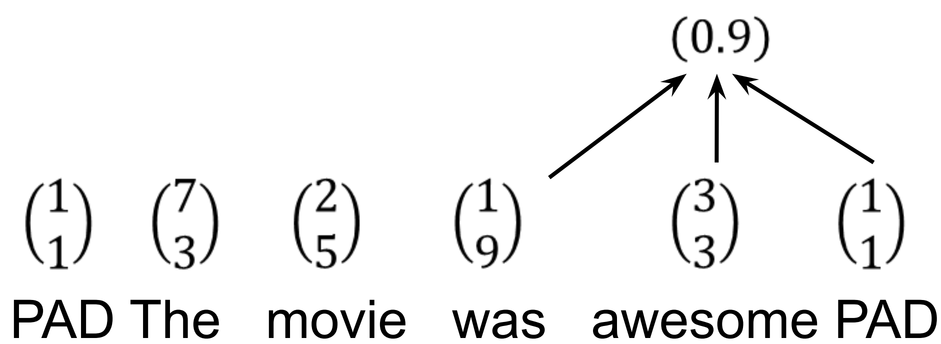
PAD The movie was awesome PAD

Convolutional Layer in NLP

Universität Bielefeld

- Input sentence $\boldsymbol{x}_{1:n}$ is a concatenation of the word vectors $\boldsymbol{x}_i \in \boldsymbol{R}^d$
- Convolutional filter: $\boldsymbol{w} \in \boldsymbol{R}^{hd}$

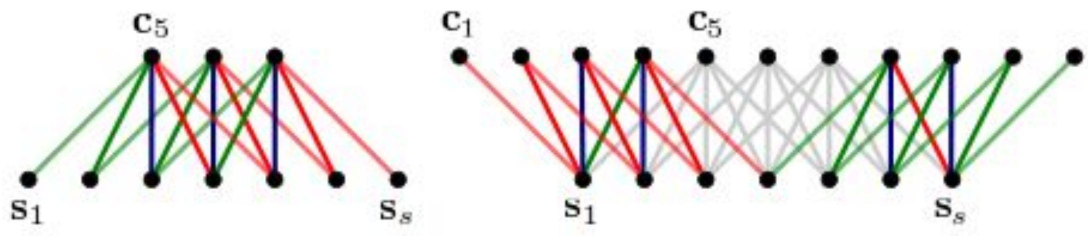
$M = 1$
$d = 2$



Wide vs narrow convolution

How do we handle the words on the margins of the sentence?

Kalchbrenner et al. 2014 recommend to use wide convolution for most NLP tasks



narrow wide

http://arxiv.org/pdf/1404.2188v1.pdf

Different viewpoints on convolution

Universität Bielefeld

- Is convolution for text in 1d or in 2d?
- Can also interpret that each input $f[i-m]$ lies in R^d
- and each weight $g[m]$ also lies in R^d

$$(\boldsymbol{f} * \boldsymbol{g})[\boldsymbol{i}] = \sum_{\boldsymbol{m}} \boldsymbol{f}[\boldsymbol{i} - \boldsymbol{m}]\boldsymbol{g}[\boldsymbol{m}]$$

Properties of convolutional networks

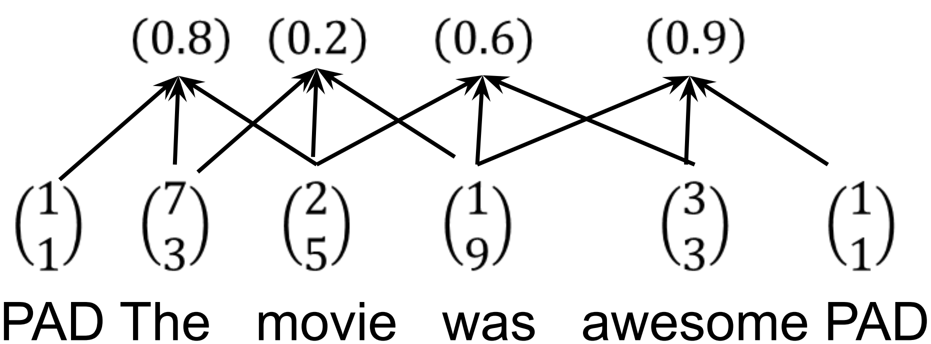
Universität Bielefeld

Not every input is connected to every output in the following layer

→ **sparse connectivity** (vs **fully-connected** layers)

For each window, we use the same weights and bias values

→ **parameter sharing**



Dense layer vs. Convolutional Layer

So, the main difference to our known dense layers is **parameter-sharing** and **sparse connectivity**

Why are these two properties important?

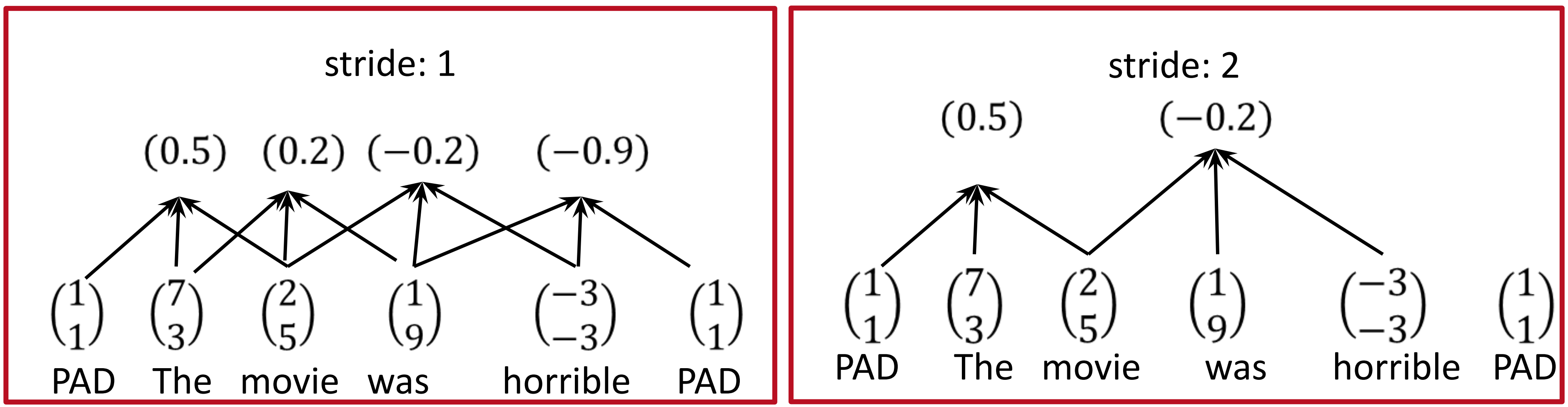
Stride

Universität Bielefeld

The stride specifies the steps size for moving over the sentence

In NLP, stride 1 is commonly used

In computer vision, other values might be more useful



Pooling layer

Universität Bielefeld

- Another new building block: **pooling layer**
  → Idea: capture the most important activation

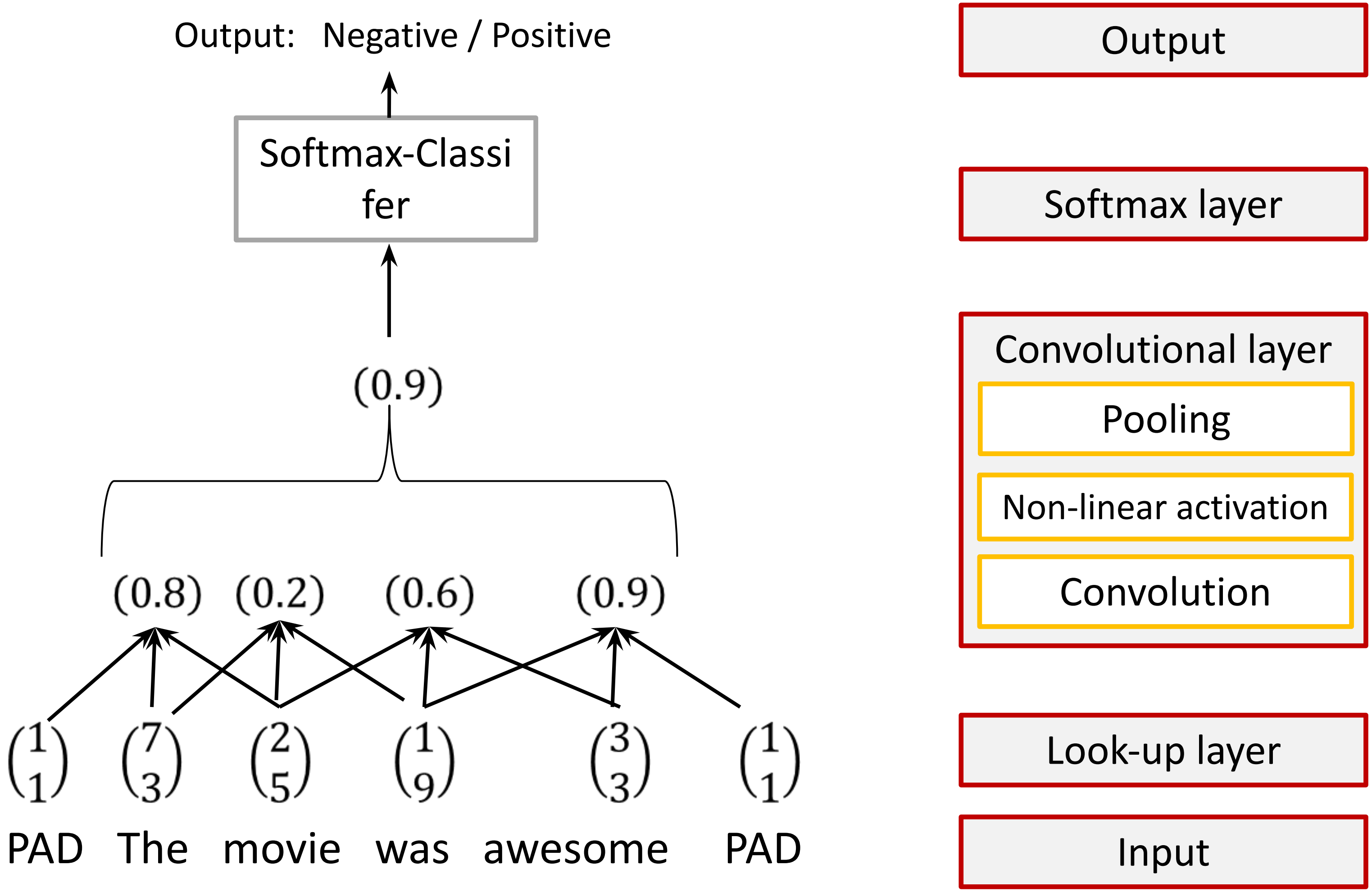
- Let $c_1, c_2, \ldots \in \boldsymbol{R}$ denote the output values for our convolutional filter
- Compute the output o for a **max-over-time pooling** layer:

$$\mathrm{o} = \max_i c_i$$

- Max-over-time pooling is most common in NLP. You can also find min-pooling and mean-pooling in other areas. Could also use some other averaging
- Note that there are no associated weights

Classification with convolution and pooling

Universität Bielefeld



Multiple filters

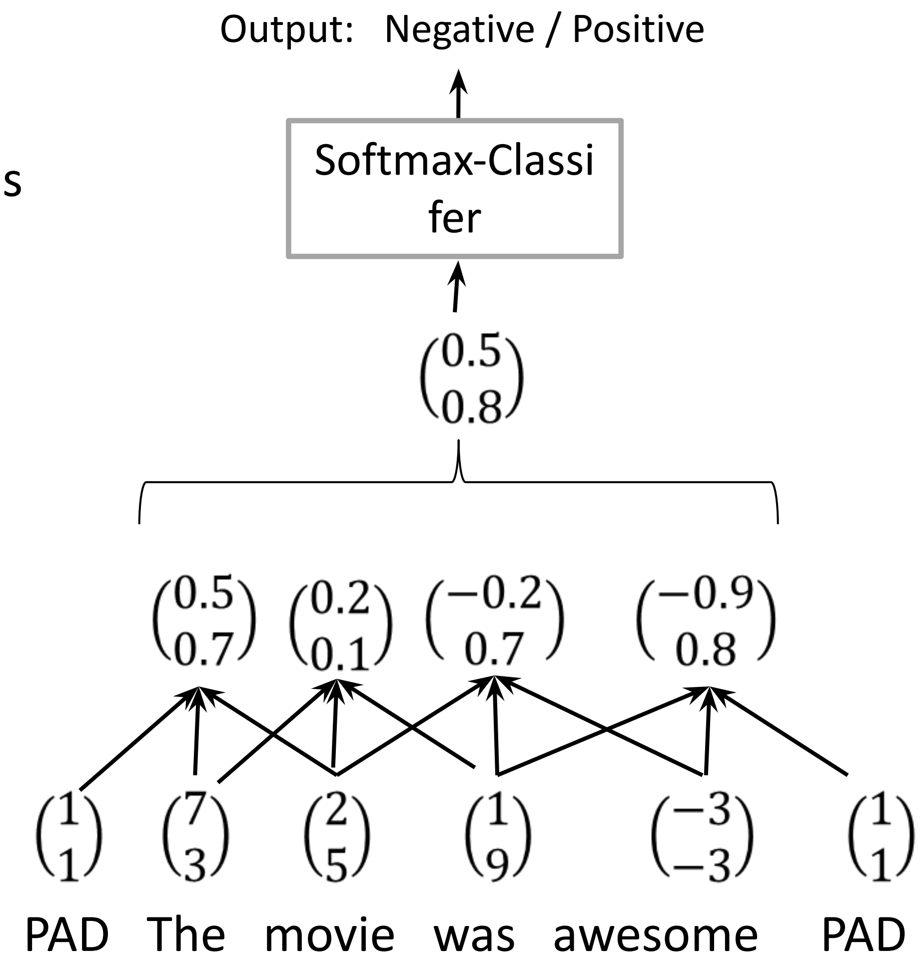
Usually we have **many** filters (hundreds or thousands), not just one

They may be of same or of different sizes

Multiple filters

Universität Bielefeld

- Further filters.
- The convolved representation is often called a feature representation.

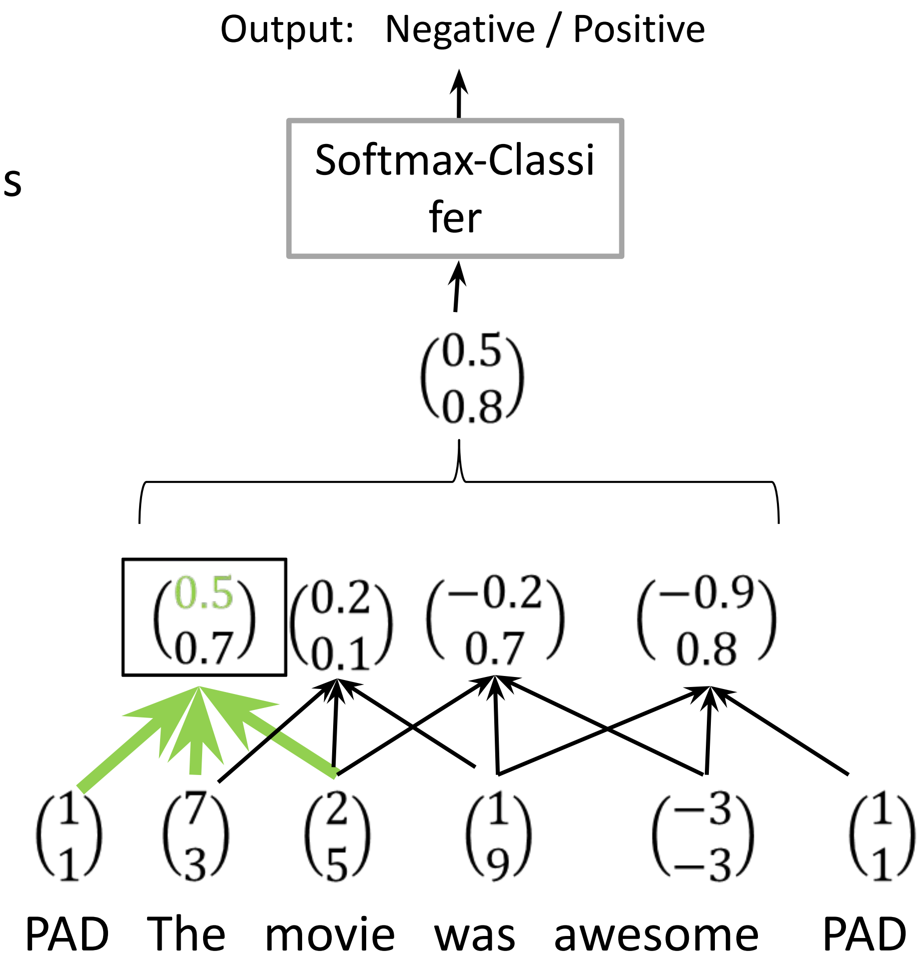


NLLG

Multiple filters

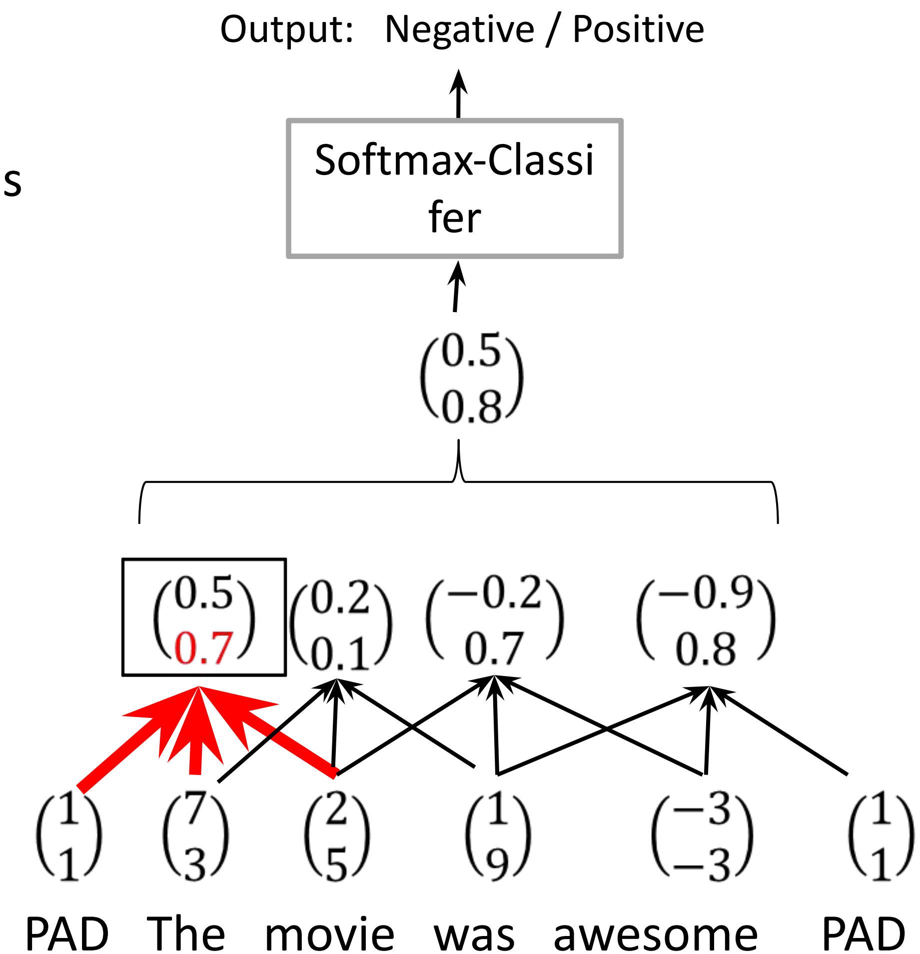
Universität Bielefeld

- Further filters.
- The convolved representation is often called a feature representation.

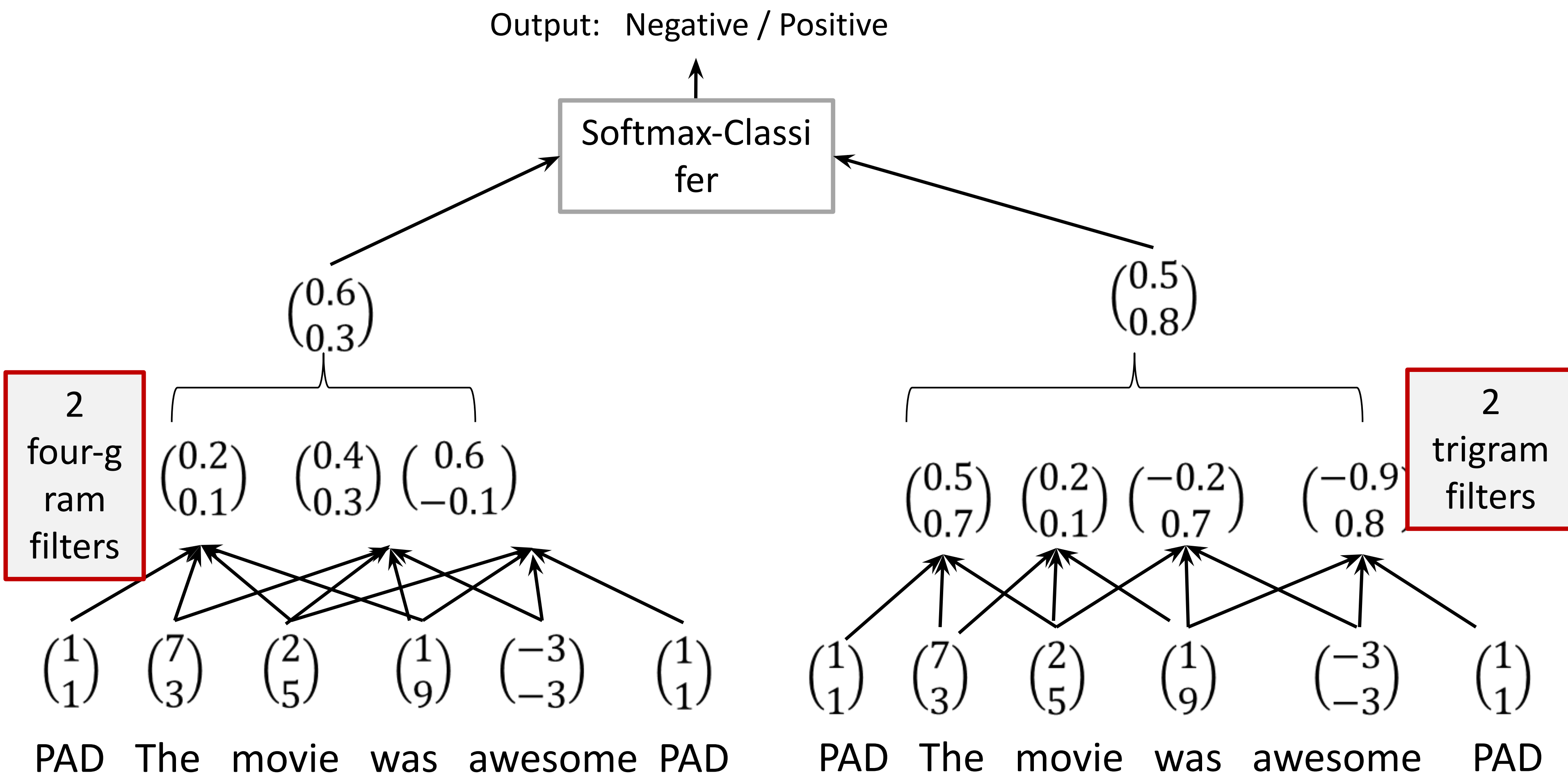


Multiple filters

- Further filters.
- The convolved representation is often called a feature representation.

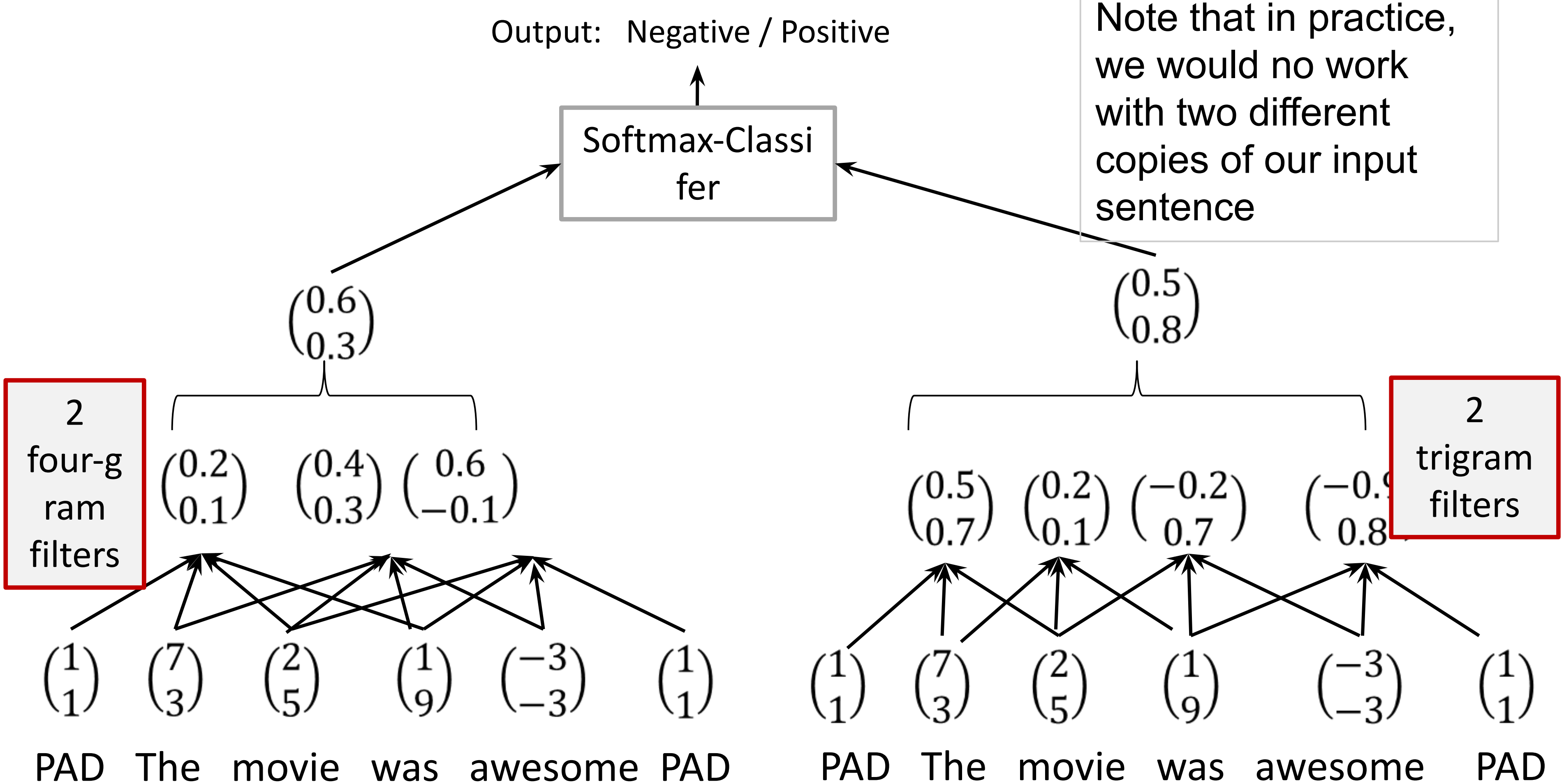


Combining different n-gram sizes



Combining different n-gram sizes

Note that in practice, we would no work with two different copies of our input sentence



Properties of pooling

Universität Bielefeld

- Idea: Extracting relevant features independent of their position in the input
- Problems:
  - Output remains the same if a feature occurs once or multiple times

    The music was great, but the cast was horrible, the plot was horrible and the costumes were horrible.

  - Order of features is not considered

    I don‘t love it, I hate it. vs *I don‘t hate it, I love it.*

Agenda

1. Convolution and pooling

2. **Convolutional networks for NLP**

Convolutional networks for NLP

Universität Bielefeld

- Sentence classification
  - Kalchbrenner, Grefenstette, Blunsom, 2014: *A Convolutional Neural Network for Modelling Sentences*
  - Kim, 2014: *Convolutional Neural Networks for Sentence Classification*
  - Zhang and Wallace, 2016: *A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification*
- Semantic role labeling (SRL)
  - The SENNA framework
    Collobert and Weston, 2011: *Natural Language Processing (Almost) from Scratch*
- Character-based approach
  - Zhang, Zhao, LeCun 2015: *Character-level Convolutional Networks for Text Classification*
  - Kim et al., 2016: *Character-aware Neural Language Models*

Sentence classification tasks

Universität Bielefeld

- Sentiment classification of movie reviews, product reviews, tweets

- Question classification into 6 question types: person, location, numeric information,…

- Classifying sentences into subjective / objective

  “I feel this work is not novel enough.“

- Classifying whether a sentence is ironic

Character-based approach

- Characters as units, smaller vocabulary:
  - 70 characters:26 English letters, 10 digits, 33 other characters and the new line character.

```
abcdefghijklmnopqrstuvwxyz0123456789
-,;.!?:'''/\|_@#$%^&*~`+-=<>()[]{}
```

  - plus uppercase letters, if required

- No embeddings, just one-hot vectors
  - Blanks and all other characters are all-zero vectors

Character-based approach for text classification

Universität Bielefeld

- Classifying
  - articles into topics
  - reviews into positive/negative

- Approach can compete with state-of-the-art, but requires huge datasets
  - Increase the training set by replacing words with their synonyms using a thesaurus

- Fascinating idea, might lead to more language-independent models, but currently word embeddings work better for most tasks

Summary

Universität Bielefeld

- Convolutional networks can deal with variable sized input
  - Sparse connectivity, parameter sharing
  - Narrow vs wide convolution
- Pooling makes it possible to focus on most relevant features
  - Max-over-time pooling
- Convolutional networks for NLP
  - Sentence classification
  - Character-based approaches

References

The lectures referenced on slide 2

www.deeplearningbook.org

Kalchbrenner, Grefenstette, Blunsom, (2014): A Convolutional Neural Network for Modelling Sentences, *arXiv preprint arXiv:1404.2188*

Kim(2014): Convolutional Neural Networks for Sentence Classification, in *Proceedings of the 2014 Conference on Empirical Methods in NLP (EMNLP)*: 1746–1751.

Zhang and Wallace, 2016: A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification, *arXiv preprint arXiv:1510.03820*.

Collobert and Weston (2011): Natural Language Processing (Almost) from Scratch, in *The Journal of Machine Learning Research 12*: 2493-2537.

Zhang, Zhao, LeCun (2015): Character-level Convolutional Networks for Text Classification, in *Advances in Neural Information Processing Systems: 649-657*.

Conneau et al. (2016), „Very Deep Convolutional Networks for Text Classification“