

**ERGO**

A Munich Re company

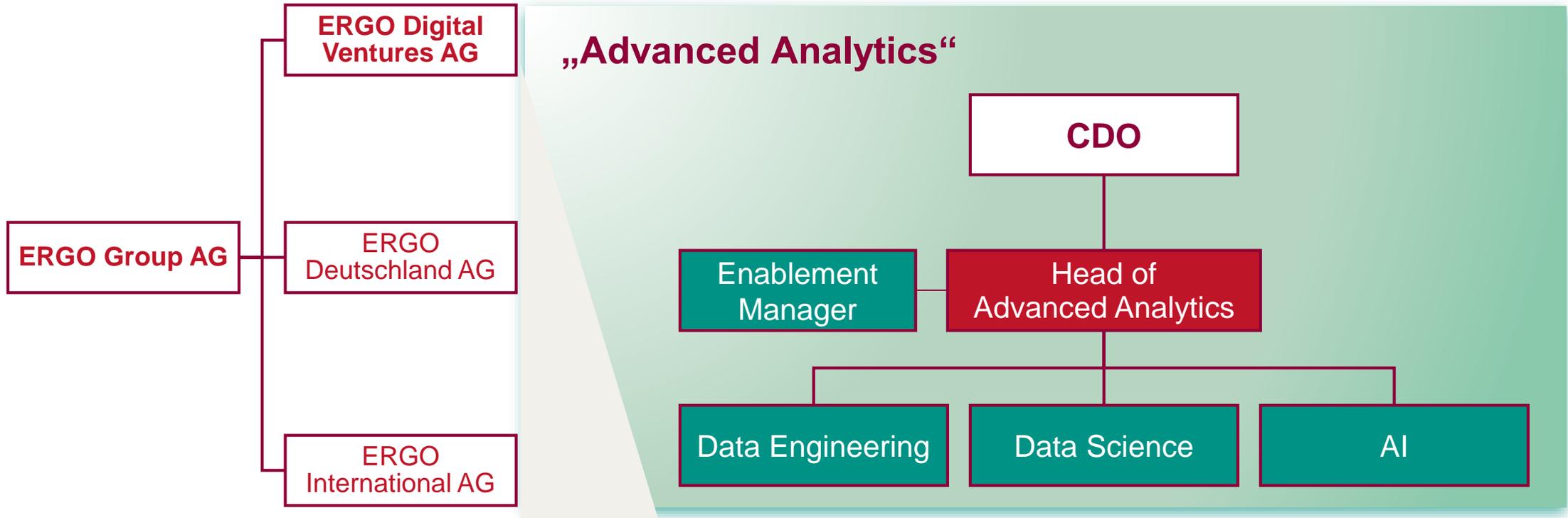
# Anwendung von Deep Learning im Eingangsmanagement

Dr. Sebastian Mikus, Dr. Lucas Hafner

DAV/DGVFM e-Herbsttagung 2020



# Einheit Advanced Analytics mit umfassendem Mandat für die ERGO Gruppe



✓ **Datenschutz und Compliance-Anforderungen** nach **höchstem Standard** berücksichtigt

✓ **Rechtliche Basis** für **ERGO Group** weite Zusammenarbeit **geschaffen**

✓ **Mandantenfähige, rollenbasierte IT-Plattform** nach **höchsten IT-Security Standards**

## Teil 1

Überblick über ML-Ansätze im Eingangsmanagement ERGO

## Teil 2

Ergebnisse der Textklassifikation bei einem konkreten Use Case

# Signifikantes Potential für Verbesserungen im Eingangsmanagement durch AI



The image shows three overlapping insurance forms from ERGO. The top form is a 'Krankenkasse bzw. Kostenträger' form for DKV, containing personal and contact information for Erika Mustermann. The middle form is an 'Unfall-Anzeige für die private Unfall-Versicherung' form, detailing an accident report. The bottom form is a 'Kranken-Versicherung' form, providing information about the policyholder and the insurance contract.

OCR

Klassifikation

Extraktion

Fachbereich

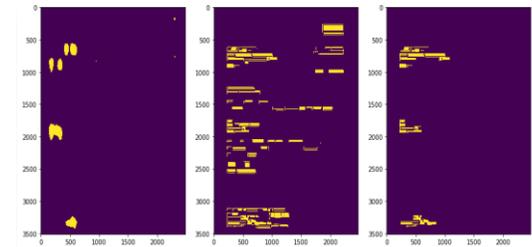
# Methodenüberblick für Fragestellungen rund um Extraktion



## Bilderkennungs-Ansätze

### z.B. AddressFinder

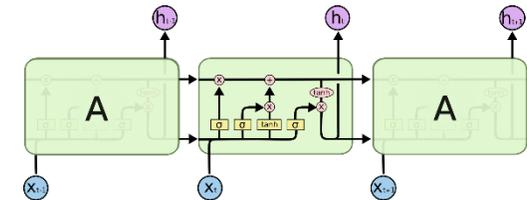
- Convolutional Neural Network
- Activation Map Algorithmus
- Extraktion von Adressen



## Texterkennungs-Ansätze

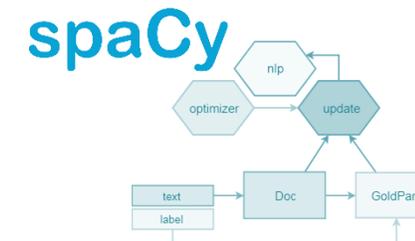
### Extractor & RoleExtractor

- Textbasierter bidirektionales LSTM
- Extraktion neuer Adressen, neuer IBANs



## Custom Extractor

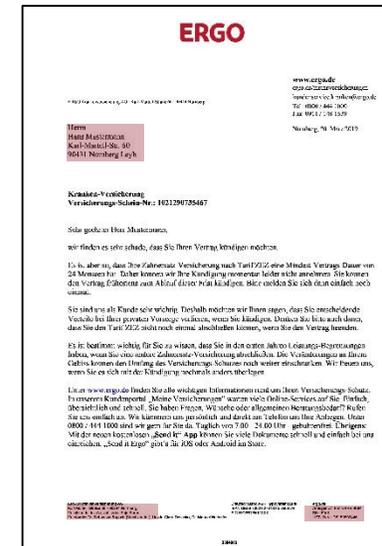
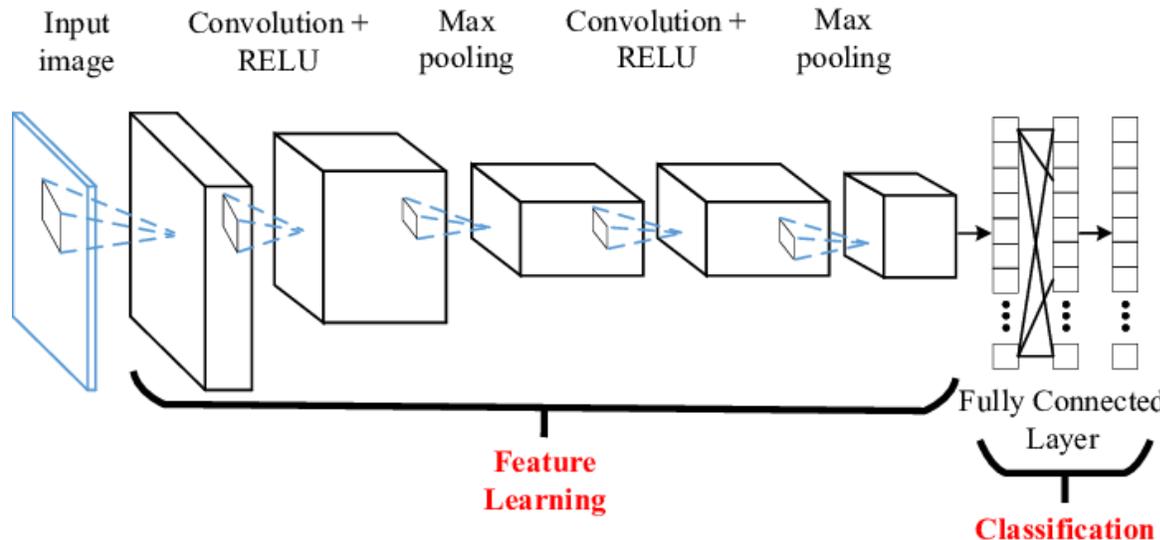
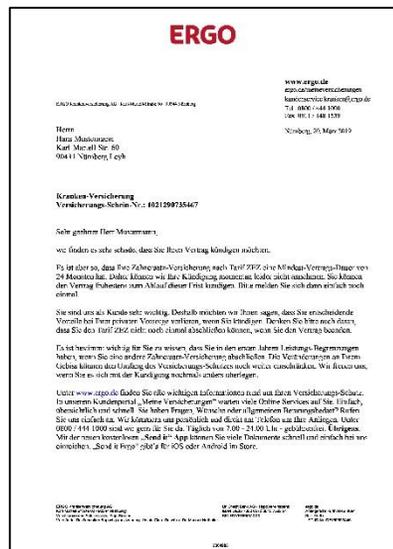
- Multilayer residual CNN Extraktion von Polizzennummern, SNR, VNR



# AddressFinder – CNN und Activation Map



A Munich Re company



Bildinformation

CNN

Activation Map

Bildinformation als Input für ein CNN



CNN klassifiziert Dokument: Adresse vorhanden (ja/nein)



Bildposition anhand derer das CNN Vorhandensein von Adresse ermittelt

# Wie funktioniert ein Convolutional Neural Network (CNN) (1/10)?

1	2	3	4	5	6
7	8	9	10	11	12
1	2	3	4	5	6
7	8	9	10	11	12
1	2	3	4	5	6
7	8	9	10	11	12

9	8	7
6	5	4
3	2	1



Input-Layer

Kantengewichte  
(gemeinsame  
Kantengewichte)

# Wie funktioniert ein CNN (2/10)?

1	2	3	4	5	6
7	8	9	10	11	12
1	2	3	4	5	6
7	8	9	10	11	12
1	2	3	4	5	6
7	8	9	10	11	12

9	8	7
6	5	-4
0	2	1

99

$$1 * 9 + 2 * 8 + 3 * 7 + 7 * 6 + 8 * 5 + 9 * (-4) + 1 * 0 + 2 * 2 + 3 * 1$$

Input-Layer

Kantengewichte  
(gemeinsame  
Kantengewichte)

# Wie funktioniert ein CNN (3/10)?

1	2	3	4	5	6
7	8	9	10	11	12
1	2	3	4	5	6
7	8	9	10	11	12
1	2	3	4	5	6
7	8	9	10	11	12

9	8	7
6	5	-4
0	2	1

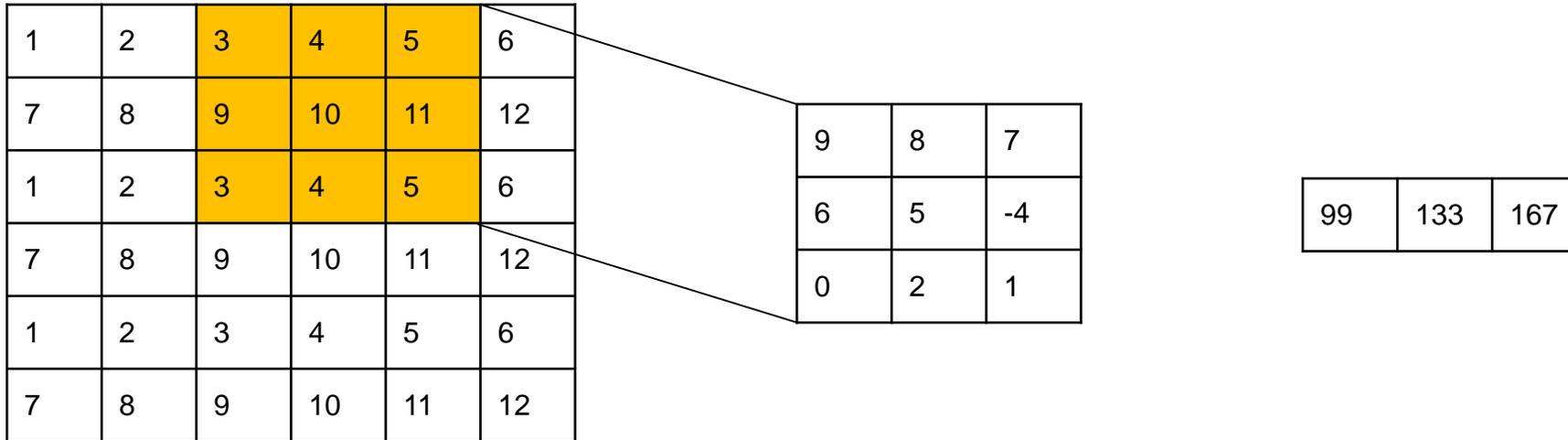
99	133
----	-----

$$2 * 9 + 3 * 8 + 4 * 7 + 8 * 6 + 9 * 5 + 10 * (-4) + 2 * 0 + 3 * 2 + 4 * 1$$

Input-Layer

Kantengewichte  
(gemeinsame  
Kantengewichte)

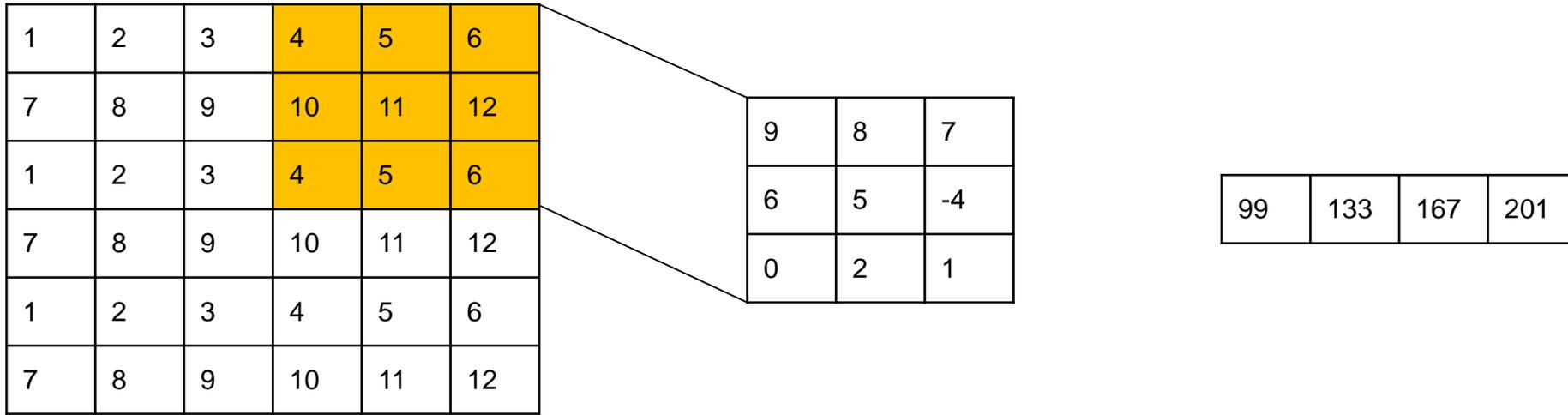
# Wie funktioniert ein CNN (4/10)?



Input-Layer

Kantengewichte  
(gemeinsame  
Kantengewichte)

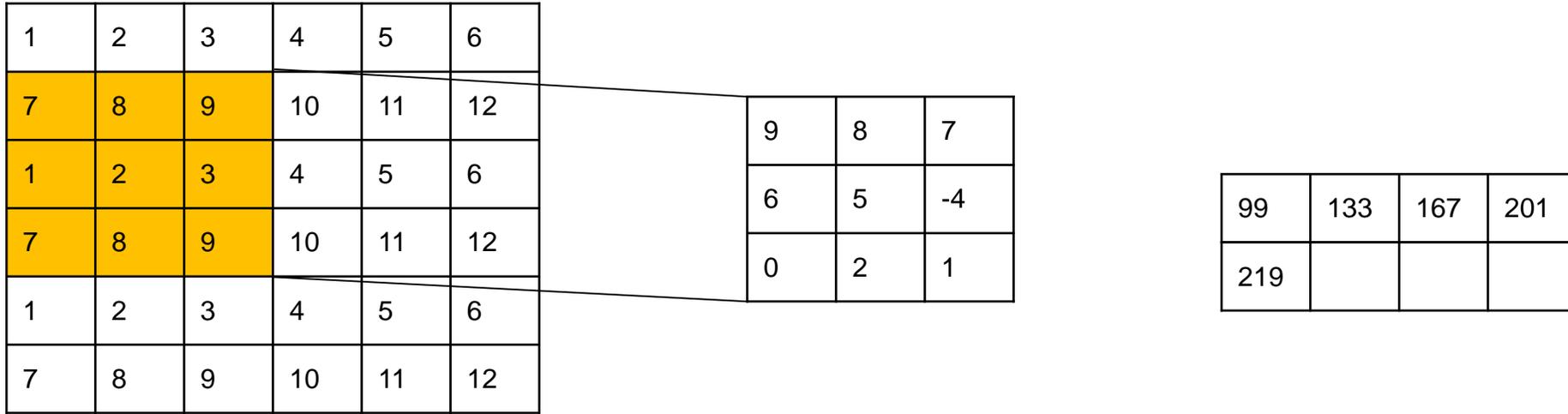
# Wie funktioniert ein CNN (5/10)?



Input-Layer

Kantengewichte  
(gemeinsame  
Kantengewichte)

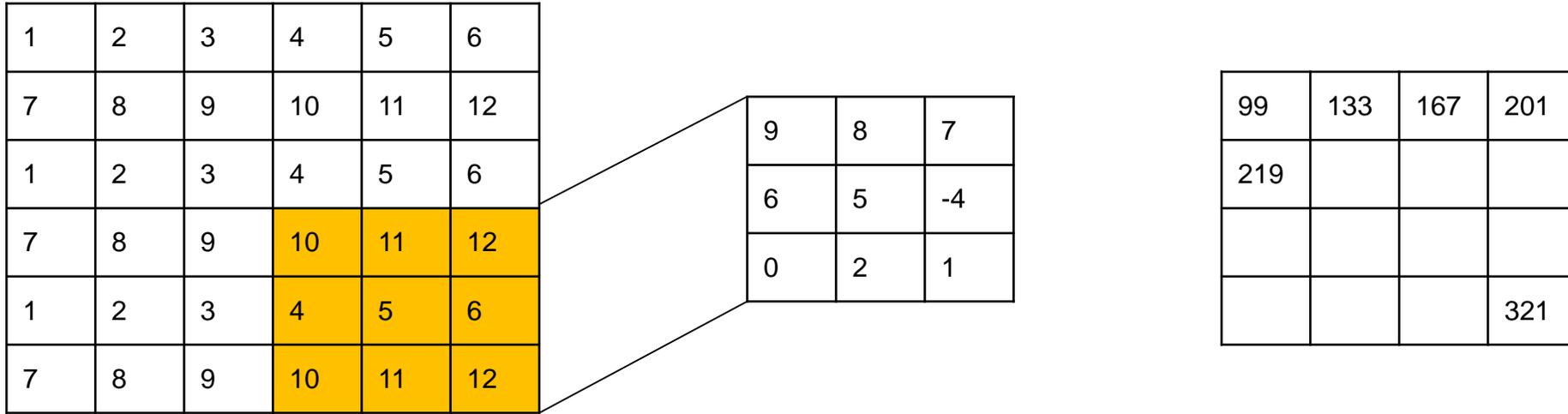
# Wie funktioniert ein CNN (6/10)?



Input-Layer

Kantengewichte  
(gemeinsame  
Kantengewichte)

# Wie funktioniert ein CNN (7/10)?



Input-Layer

Kantengewichte  
(gemeinsame  
Kantengewichte)

Output Layer

# Wie funktioniert ein CNN (8/10)?

10	10	10	10	0	0
10	10	10	10	0	0
10	10	10	10	0	0
10	10	10	10	0	0
10	10	10	10	0	0
10	10	10	10	0	0

Input-Layer

1	0	-1
1	0	-1
1	0	-1

„Filter“

0	0	30	30
0	0	30	30
0	0	30	30
0	0	30	30

Hidden Layer –  
erkennt vertikale  
Kanten

<https://setosa.io/ev/image-kernels/>

# Wie funktioniert ein CNN (9/10)?

10	10	10	10	0	0
10	10	10	10	0	0
10	10	10	10	0	0
10	10	10	10	0	0
10	10	10	10	0	0
10	10	10	10	0	0

Input-Layer

1	0	-1
1	0	-1
1	0	-1

„Filter“

0	0	30	30
0	0	30	30
0	0	30	30
0	0	30	30

Hidden Layer –  
erkennt vertikale  
Kanten

0	30
0	30

Hidden Layer –  
Maximum (Max  
Pooling)

<https://setosa.io/ev/image-kernels/>

# Wie funktioniert ein CNN (10/10)?

10	10	10	10	0	0
10	10	10	10	0	0
10	10	10	10	0	0
10	10	10	10	0	0
10	10	10	10	0	0
10	10	10	10	0	0

Input-Layer

$w_{11}$	$w_{21}$	$w_{31}$
$w_{12}$	$w_{22}$	$w_{32}$
$w_{13}$	$w_{23}$	$w_{33}$

„Filter“ + Bias

0	0	30	30
0	0	30	30
0	0	30	30
0	0	30	30

Hidden Layer –  
erkennt vertikale  
Kanten

0	30
0	30

Hidden Layer –  
Maximum (Max  
Pooling)

- Die Filter sind Parameter, die selbständig trainiert werden.
- Filter können selbständig spezielle Muster in den Bildern entdecken (z.B. Kanten).
- Pooling „vergisst“ die genau Position im Bild.
- Insgesamt gibt es wenig Parameter (verglichen mit einem vollständig verbundenen Layer).

# Visualisierung von CNNs

Layer 1



Layer 5

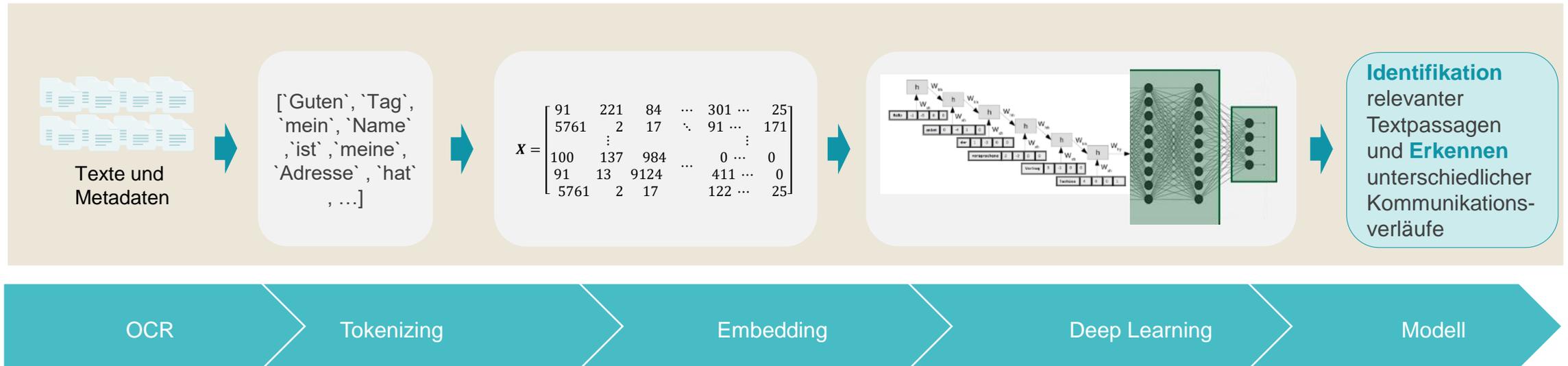


Original



Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European conference on computer vision*. Springer, Cham, 2014.

# Verarbeitung unstrukturierter Daten – vom Bild zum Modell



# Signifikantes Potential für Verbesserungen im Eingangsmanagement durch AI



The image shows three overlapping insurance forms from ERGO. The top form is a 'Krankenkasse bzw. Kostenträger' form for DKV, containing personal and contact information for Erika Mustermann. The middle form is an 'Unfall-Anzeige für die private Unfall-Versicherung' form, detailing an accident report. The bottom form is a 'Kranken-Versicherung' form, providing information about the insured person and the insurance policy.

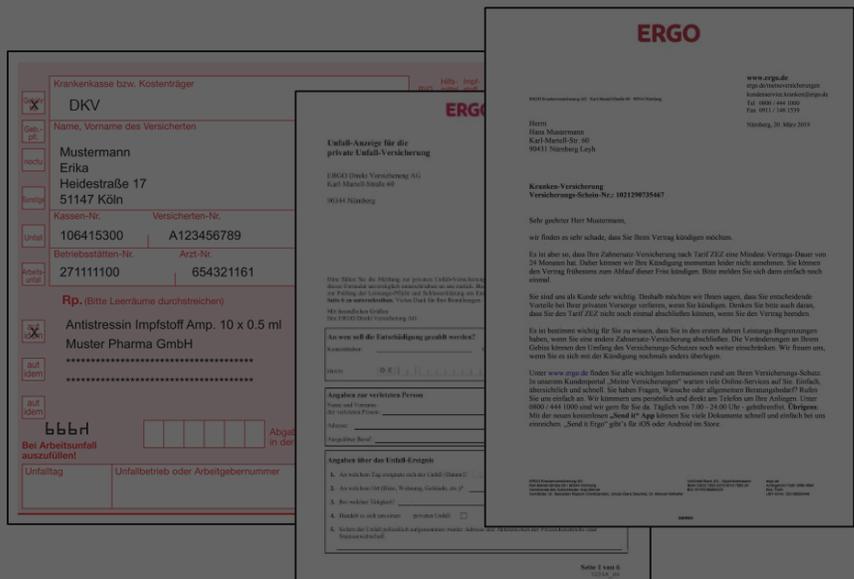
OCR

Klassifikation

Extraktion

Fachbereich

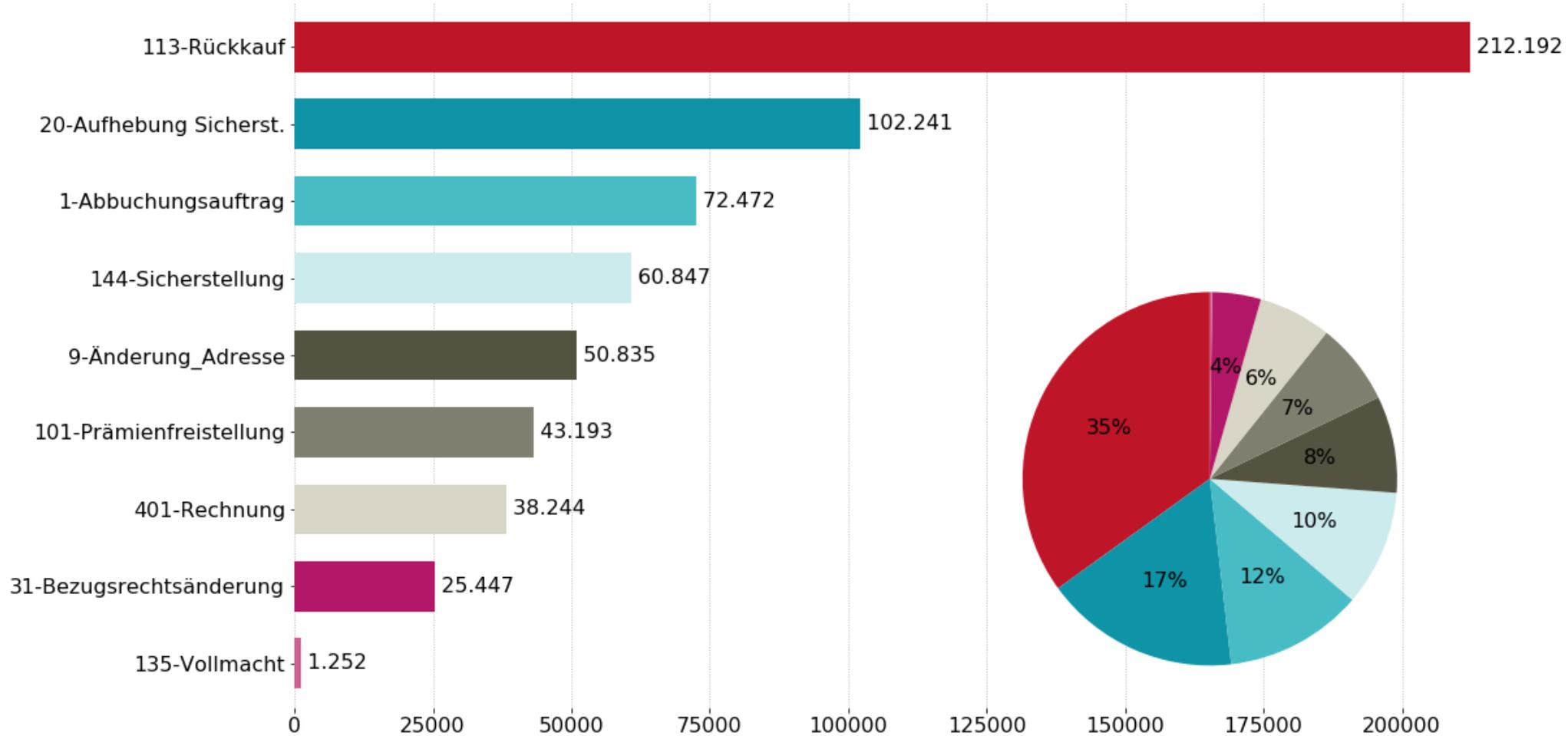
# Signifikantes Potential für Verbesserungen im Eingangsmanagement durch AI



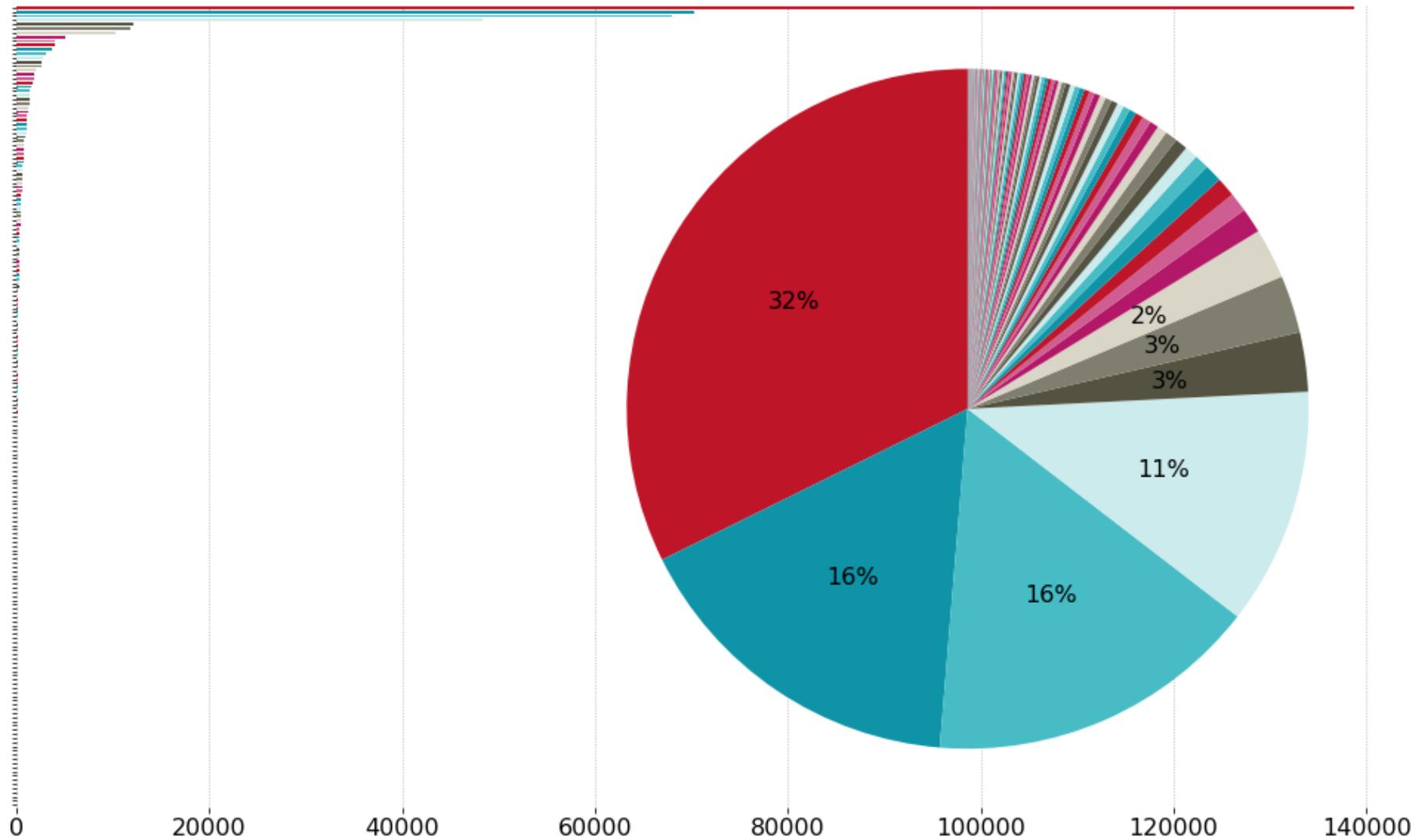
# Häufigkeitsverteilung der Eingangsklassen POC

**ERGO**

A Munich Re company



# Häufigkeitsverteilung der Eingangsklassen MVP



# Modellierungsansätze – Simple Word Embedding Models (SWEM)

## Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms

Dinghan Shen<sup>1</sup>, Guoyin Wang<sup>1</sup>, Wenlin Wang<sup>1</sup>, Martin Renqiang Min<sup>2</sup>

Qinliang Su<sup>3</sup>, Yizhe Zhang<sup>4</sup>, Chunyuan Li<sup>1</sup>, Ricardo Henao<sup>1</sup>, Lawrence Carin<sup>1</sup>

<sup>1</sup> Duke University <sup>2</sup> NEC Laboratories America <sup>3</sup> Sun Yat-sen University <sup>4</sup> Microsoft Research

dinghan.shen@duke.edu

### Abstract

Many deep learning architectures have been proposed to model the *compositionality* in text sequences, requiring a substantial number of parameters and expensive computations. However, there has not been a rigorous evaluation regarding the added value of sophisticated compositional functions. In this paper, we conduct a point-by-point comparative study between Simple Word-Embedding-based Models (SWEMs), consisting of parameter-free pooling operations, relative to word-embedding-based RNN/CNN models. Surprisingly, SWEMs exhibit comparable or even superior performance in the majority of cases considered. Based upon this understanding, we propose two additional pooling strategies over learned word embeddings: (i) a max-pooling operation for improved interpretability:

tic regularities and patterns (Bengio et al., 2003; Mikolov et al., 2013; Pennington et al., 2014). Leveraging the word-embedding construct, many deep architectures have been proposed to model the *compositionality* in variable-length text sequences. These methods range from simple operations like addition (Mitchell and Lapata, 2010; Iyyer et al., 2015), to more sophisticated compositional functions such as Recurrent Neural Networks (RNNs) (Tai et al., 2015; Sutskever et al., 2014), Convolutional Neural Networks (CNNs) (Kalchbrenner et al., 2014; Kim, 2014; Zhang et al., 2017a) and Recursive Neural Networks (Socher et al., 2011a).

Models with more expressive compositional functions, e.g., RNNs or CNNs, have demonstrated impressive results; however, they are typically computationally expensive, due to the need to estimate hundreds of thousands, if not millions, of parameters (Parikh et al., 2016). In contrast, models with simple compositional functions often compute a sentence or document embedding by

Shen, Dinghan, et al. "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms." *arXiv preprint arXiv:1805.09843* (2018).  
<https://arxiv.org/abs/1805.09843>

# Modellierungsansätze – Simple Word Embedding Models (SWEM)

## Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms

Dinghan Shen<sup>1</sup>, Guoyin Wang<sup>1</sup>, Wenlin Wang<sup>1</sup>, Martin Renqiang Min<sup>2</sup>

Qinliang Su<sup>3</sup>, Yizhe Zhang<sup>4</sup>, Chunyuan Li<sup>1</sup>, Ricardo Henao<sup>1</sup>, Lawrence Carin<sup>1</sup>

<sup>1</sup> Duke University <sup>2</sup> NEC Laboratories America <sup>3</sup> Sun Yat-sen University <sup>4</sup> Microsoft Research

dinghan.shen@duke.edu

### Abstract

Many deep learning architectures have been proposed to model the *compositionality* in text sequences, requiring a substantial number of parameters and expensive computations. However, there has not been a rigorous evaluation regarding the added value of sophisticated compositional functions. In this paper, we conduct a point-by-point comparative study between Simple Word-Embedding-based Models (SWEMs), consisting of parameter-free pooling operations, relative to word-embedding-based RNN/CNN models. Surprisingly, SWEMs exhibit comparable or even superior performance in the majority of cases considered. Based upon this understanding, we propose two additional pooling strategies over learned word embeddings: (i) a max-pooling operation for improved interpretability:

tic regularities and patterns (Bengio et al., 2003; Mikolov et al., 2013; Pennington et al., 2014). Leveraging the word-embedding construct, many deep architectures have been proposed to model the *compositionality* in variable-length text sequences. These methods range from simple operations like addition (Mitchell and Lapata, 2010; Iyyer et al., 2015), to more sophisticated compositional functions such as Recurrent Neural Networks (RNNs) (Tai et al., 2015; Sutskever et al., 2014), Convolutional Neural Networks (CNNs) (Kalchbrenner et al., 2014; Kim, 2014; Zhang et al., 2017a) and Recursive Neural Networks (Socher et al., 2011a).

Models with more expressive compositional functions, e.g., RNNs or CNNs, have demonstrated impressive results; however, they are typically computationally expensive, due to the need to estimate hundreds of thousands, if not millions, of parameters (Parikh et al., 2016). In contrast, models with simple compositional functions often compute a sentence or document embedding by

Shen, Dinghan, et al. "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms." *arXiv preprint arXiv:1805.09843* (2018).  
<https://arxiv.org/abs/1805.09843>

# Modellierungsansätze – Simple Word Embedding Models (SWEM)

## Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms

Dinghan Shen<sup>1</sup>, Guoyin Wang<sup>1</sup>, Wenlin Wang<sup>1</sup>, Martin Renqiang Min<sup>2</sup>

Qinliang Su<sup>3</sup>, Yizhe Zhang<sup>4</sup>, Chunyuan Li<sup>1</sup>, Ricardo Henao<sup>1</sup>, Lawrence Carin<sup>1</sup>

<sup>1</sup> Duke University <sup>2</sup> NEC Laboratories America <sup>3</sup> Sun Yat-sen University <sup>4</sup> Microsoft Research

dinghan.shen@duke.edu

### Abstract

Many deep learning architectures have been proposed to model the *compositionality* in text sequences, requiring a substantial number of parameters and expensive computations. However, there has not been a rigorous evaluation regarding the added value of sophisticated compositional functions. In this paper, we conduct a point-by-point comparative study between Simple Word-Embedding-based Models (SWEMs), consisting of parameter-free pooling operations, relative to word-embedding-based RNN/CNN models. Surprisingly, SWEMs exhibit comparable or even superior performance in the majority of cases considered. Based upon this understanding, we propose two additional pooling strategies over learned word embeddings: (i) a max-pooling operation for improved interpretability:

tic regularities and patterns (Bengio et al., 2003; Mikolov et al., 2013; Pennington et al., 2014). Leveraging the word-embedding construct, many deep architectures have been proposed to model the *compositionality* in variable-length text sequences. These methods range from simple operations like addition (Mitchell and Lapata, 2010; Iyyer et al., 2015), to more sophisticated compositional functions such as Recurrent Neural Networks (RNNs) (Tai et al., 2015; Sutskever et al., 2014), Convolutional Neural Networks (CNNs) (Kalchbrenner et al., 2014; Kim, 2014; Zhang et al., 2017a) and Recursive Neural Networks (Socher et al., 2011a).

Models with more expressive compositional functions, e.g., RNNs or CNNs, have demonstrated impressive results; however, they are typically computationally expensive, due to the need to estimate hundreds of thousands, if not millions, of parameters (Parikh et al., 2016). In contrast, models with simple compositional functions often compute a sentence or document embedding by

Model	Yahoo! Ans.	AG News	Yelp P.	Yelp F.	DBpedia
Bag-of-means*	60.55	83.09	87.33	53.54	90.45
Small word CNN*	69.98	89.13	94.46	58.59	98.15
Large word CNN*	70.94	91.45	95.11	59.48	98.28
LSTM*	70.84	86.06	94.74	58.17	98.55
Deep CNN (29 layer)†	73.43	91.27	95.72	64.26	98.71
fastText ‡	72.0	91.5	93.8	60.4	98.1
fastText (bigram)‡	72.3	92.5	95.7	63.9	98.6
SWEM-aver	73.14	91.71	93.59	60.66	98.42
SWEM-max	72.66	91.79	93.25	59.63	98.24
SWEM-concat	73.53	92.66	93.76	61.11	98.57
SWEM-hier	73.48	92.48	95.81	63.79	98.54

Table 2: Test accuracy on (long) document classification tasks, in percentage. Results marked with \* are reported in Zhang et al. (2015b), with † are reported in Conneau et al. (2016), and with ‡ are reported in Joulin et al. (2016).

Shen, Dinghan, et al. "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms." *arXiv preprint arXiv:1805.09843* (2018). <https://arxiv.org/abs/1805.09843>

# Modellierungsansätze – Simple Word Embedding Models (SWEM)

## Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms

Dinghan Shen<sup>1</sup>, Guoyin Wang<sup>1</sup>, Wenlin Wang<sup>1</sup>, Martin Renqiang Min<sup>2</sup>

Qinliang Su<sup>3</sup>, Yizhe Zhang<sup>4</sup>, Chunyuan Li<sup>1</sup>, Ricardo Henao<sup>1</sup>, Lawrence Carin<sup>1</sup>

<sup>1</sup> Duke University <sup>2</sup> NEC Laboratories America <sup>3</sup> Sun Yat-sen University <sup>4</sup> Microsoft Research

dinghan.shen@duke.edu

### Abstract

Many deep learning architectures have been proposed to model the *compositionality* in text sequences, requiring a substantial number of parameters and expensive computations. However, there has not been a rigorous evaluation regarding the added value of sophisticated compositional functions. In this paper, we conduct a point-by-point comparative study between Simple Word-Embedding-based Models (SWEMs), consisting of parameter-free pooling operations, relative to word-embedding-based RNN/CNN models. Surprisingly, SWEMs exhibit comparable or even superior performance in the majority of cases considered. Based upon this understanding, we propose two additional pooling strategies over learned word embeddings: (i) a max-pooling operation for improved interpretability:

tic regularities and patterns (Bengio et al., 2003; Mikolov et al., 2013; Pennington et al., 2014). Leveraging the word-embedding construct, many deep architectures have been proposed to model the *compositionality* in variable-length text sequences. These methods range from simple operations like addition (Mitchell and Lapata, 2010; Iyyer et al., 2015), to more sophisticated compositional functions such as Recurrent Neural Networks (RNNs) (Tai et al., 2015; Sutskever et al., 2014), Convolutional Neural Networks (CNNs) (Kalchbrenner et al., 2014; Kim, 2014; Zhang et al., 2017a) and Recursive Neural Networks (Socher et al., 2011a).

Models with more expressive compositional functions, e.g., RNNs or CNNs, have demonstrated impressive results; however, they are typically computationally expensive, due to the need to estimate hundreds of thousands, if not millions, of parameters (Parikh et al., 2016). In contrast, models with simple compositional functions often compute a sentence or document embedding by

Model	Yahoo! Ans.	AG News	Yelp P.	Yelp F.	DBpedia
Bag-of-means*	60.55	83.09	87.33	53.54	90.45
Small word CNN*	69.98	89.13	94.46	58.59	98.15
Large word CNN*	70.94	91.45	95.11	59.48	98.28
LSTM*	70.84	86.06	94.74	58.17	98.55
Deep CNN (29 layer)†	73.43	91.27	95.72	64.26	98.71
fastText ‡	72.0	91.5	93.8	60.4	98.1
fastText (bigram)‡	72.3	92.5	95.7	63.9	98.6
SWEM-aver	73.14	91.71	93.59	60.66	98.42
SWEM-max	72.66	91.79	93.25	59.63	98.24
SWEM-concat	73.53	92.66	93.76	61.11	98.57
SWEM-hier	73.48	92.48	95.81	63.79	98.54

Table 2: Test accuracy on (long) document classification tasks, in percentage. Results marked with \* are reported in Zhang et al. (2015b), with † are reported in Conneau et al. (2016), and with ‡ are reported in Joulin et al. (2016).

Shen, Dinghan, et al. "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms." *arXiv preprint arXiv:1805.09843* (2018). <https://arxiv.org/abs/1805.09843>

# Modellierungsansätze – Simple Word Embedding Models (SWEM)

## Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms

Dinghan Shen<sup>1</sup>, Guoyin Wang<sup>1</sup>, Wenlin Wang<sup>1</sup>, Martin Renqiang Min<sup>2</sup>

Qinliang Su<sup>3</sup>, Yizhe Zhang<sup>4</sup>, Chunyuan Li<sup>1</sup>, Ricardo Henao<sup>1</sup>, Lawrence Carin<sup>1</sup>

<sup>1</sup> Duke University <sup>2</sup> NEC Laboratories America <sup>3</sup> Sun Yat-sen University <sup>4</sup> Microsoft Research

dinghan.shen@duke.edu

### Abstract

Many deep learning architectures have been proposed to model the *compositionality* in text sequences, requiring a substantial number of parameters and expensive computations. However, there has not been a rigorous evaluation regarding the added value of sophisticated compositional functions. In this paper, we conduct a point-by-point comparative study between Simple Word-Embedding-based Models (SWEMs), consisting of parameter-free pooling operations, relative to word-embedding-based RNN/CNN models. Surprisingly, SWEMs exhibit comparable or even superior performance in the majority of cases considered. Based upon this understanding, we propose two additional pooling strategies over learned word embeddings: (i) a max-pooling operation for improved interpretability:

tic regularities and patterns (Bengio et al., 2003; Mikolov et al., 2013; Pennington et al., 2014). Leveraging the word-embedding construct, many deep architectures have been proposed to model the *compositionality* in variable-length text sequences. These methods range from simple operations like addition (Mitchell and Lapata, 2010; Iyyer et al., 2015), to more sophisticated compositional functions such as Recurrent Neural Networks (RNNs) (Tai et al., 2015; Sutskever et al., 2014), Convolutional Neural Networks (CNNs) (Kalchbrenner et al., 2014; Kim, 2014; Zhang et al., 2017a) and Recursive Neural Networks (Socher et al., 2011a).

Models with more expressive compositional functions, e.g., RNNs or CNNs, have demonstrated impressive results; however, they are typically computationally expensive, due to the need to estimate hundreds of thousands, if not millions, of parameters (Parikh et al., 2016). In contrast, models with simple compositional functions often compute a sentence or document embedding by

Model	Yahoo! Ans.	AG News	Yelp P.	Yelp F.	DBpedia
Bag-of-means*	60.55	83.09	87.33	53.54	90.45
Small word CNN*	69.98	89.13	94.46	58.59	98.15
Large word CNN*	70.94	91.45	95.11	59.48	98.28
LSTM*	70.84	86.06	94.74	58.17	98.55
Deep CNN (29 layer)†	73.43	91.27	95.72	64.26	98.71
fastText ‡	72.0	91.5	93.8	60.4	98.1
fastText (bigram)‡	72.3	92.5	95.7	63.9	98.6
SWEM-aver	73.14	91.71	93.59	60.66	98.42
SWEM-max	72.66	91.79	93.25	59.63	98.24
SWEM-concat	73.53	92.66	93.76	61.11	98.57
SWEM-hier	73.48	92.48	95.81	63.79	98.54

Table 2: Test accuracy on (long) document classification tasks, in percentage. Results marked with \* are reported in Zhang et al. (2015b), with † are reported in Conneau et al. (2016), and with ‡ are reported in Joulin et al. (2016).

Shen, Dinghan, et al. "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms." *arXiv preprint arXiv:1805.09843* (2018). <https://arxiv.org/abs/1805.09843>

Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

# Modellierungsansätze – Simple Word Embedding Models (SWEM)

**ERGO**

A Munich Re company

Anbei	sende	ich	Ihnen	meine	Rechnung	zum	Schaden
-------	-------	-----	-------	-------	----------	-----	---------

Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

# Modellierungsansätze – Simple Word Embedding Models (SWEM)

**ERGO**

A Munich Re company

Embedding-Dimension	Anbei	sende	ich	Ihnen	meine	Rechnung	zum	Schaden
1								
2								
:								
20								

# Modellierungsansätze – Simple Word Embedding Models (SWEM)

Embedding-Dimension	Anbei	sende	ich	Ihnen	meine	Rechnung	zum	Schaden
1	0.1	0.2	0.3	0.1	0.1	0.9	0.1	0.8
2	0.2	0.2	0.8	0.9	0.8	0.1	0.1	0.1
:	:	:	:	:	:	:	:	:
20	0.7	0.8	0.1	0.1	0.1	0.2	0.6	0.1

# Modellierungsansätze – Simple Word Embedding Models (SWEM)

Embedding-Dimension	Anbei	sende	ich	Ihnen	meine	Rechnung	zum	Schaden	SWEM-AVG
1	0.1	0.2	0.3	0.1	0.1	0.9	0.1	0.8	0.325
2	0.2	0.2	0.8	0.9	0.8	0.1	0.1	0.1	
:	:	:	:	:	:	:	:	:	:
20	0.7	0.8	0.1	0.1	0.1	0.2	0.6	0.1	

# Modellierungsansätze – Simple Word Embedding Models (SWEM)

Embedding-Dimension	Anbei	sende	ich	Ihnen	meine	Rechnung	zum	Schaden	SWEM-AVG
1	0.1	0.2	0.3	0.1	0.1	0.9	0.1	0.8	0.325
2	0.2	0.2	0.8	0.9	0.8	0.1	0.1	0.1	0.4
:	:	:	:	:	:	:	:	:	:
20	0.7	0.8	0.1	0.1	0.1	0.2	0.6	0.1	

# Modellierungsansätze – Simple Word Embedding Models (SWEM)

Embedding-Dimension	Anbei	sende	ich	Ihnen	meine	Rechnung	zum	Schaden	SWEM-AVG
1	0.1	0.2	0.3	0.1	0.1	0.9	0.1	0.8	0.325
2	0.2	0.2	0.8	0.9	0.8	0.1	0.1	0.1	0.4
:	:	:	:	:	:	:	:	:	:
20	0.7	0.8	0.1	0.1	0.1	0.2	0.6	0.1	0.34

Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

# Modellierungsansätze – Simple Word Embedding Models (SWEM)

Input Layer  $\in \mathbb{R}^{20}$



Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

# Modellierungsansätze – Simple Word Embedding Models (SWEM)

Input Layer  $\in \mathbb{R}^{20}$



Hidden Layer  $\in \mathbb{R}^{14}$

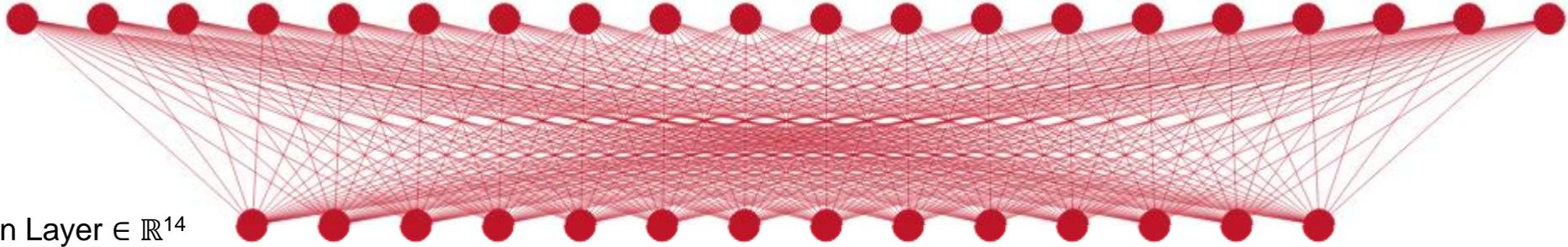


Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

# Modellierungsansätze – Simple Word Embedding Models (SWEM)

Input Layer  $\in \mathbb{R}^{20}$

Hidden Layer  $\in \mathbb{R}^{14}$



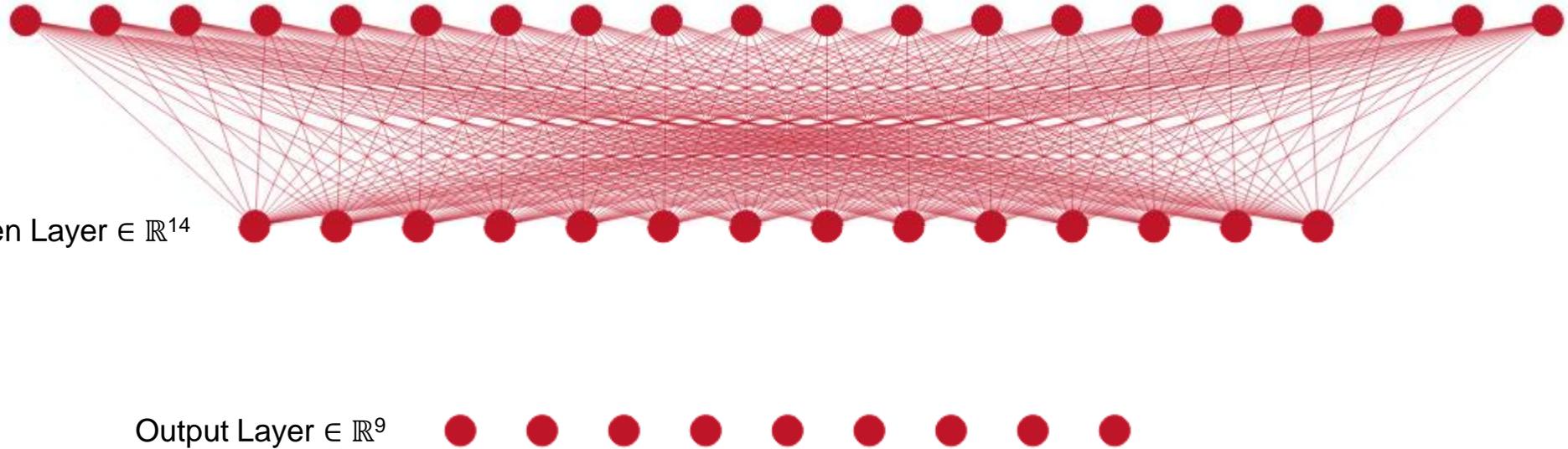
Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

# Modellierungsansätze – Simple Word Embedding Models (SWEM)

Input Layer  $\in \mathbb{R}^{20}$

Hidden Layer  $\in \mathbb{R}^{14}$

Output Layer  $\in \mathbb{R}^9$

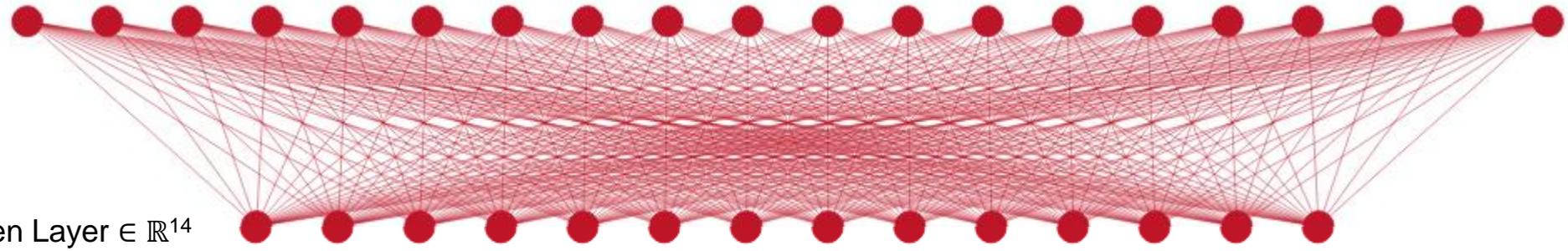


# Modellierungsansätze – Simple Word Embedding Models (SWEM)

Input Layer  $\in \mathbb{R}^{20}$

Hidden Layer  $\in \mathbb{R}^{14}$

Output Layer  $\in \mathbb{R}^9$



- Rückkauf
- Aufhebung Sicherstellung
- Abbuchungsauftrag
- Änderung Adresse
- Sicherstellung
- Prämienfreistellung
- Rechnung
- Bezugsrechtsänderung
- Vollmacht

# Modellierungsansätze – Simple Word Embedding Models (SWEM)

Input Layer  $\in \mathbb{R}^{20}$

Hidden Layer  $\in \mathbb{R}^{14}$

Output Layer  $\in \mathbb{R}^9$

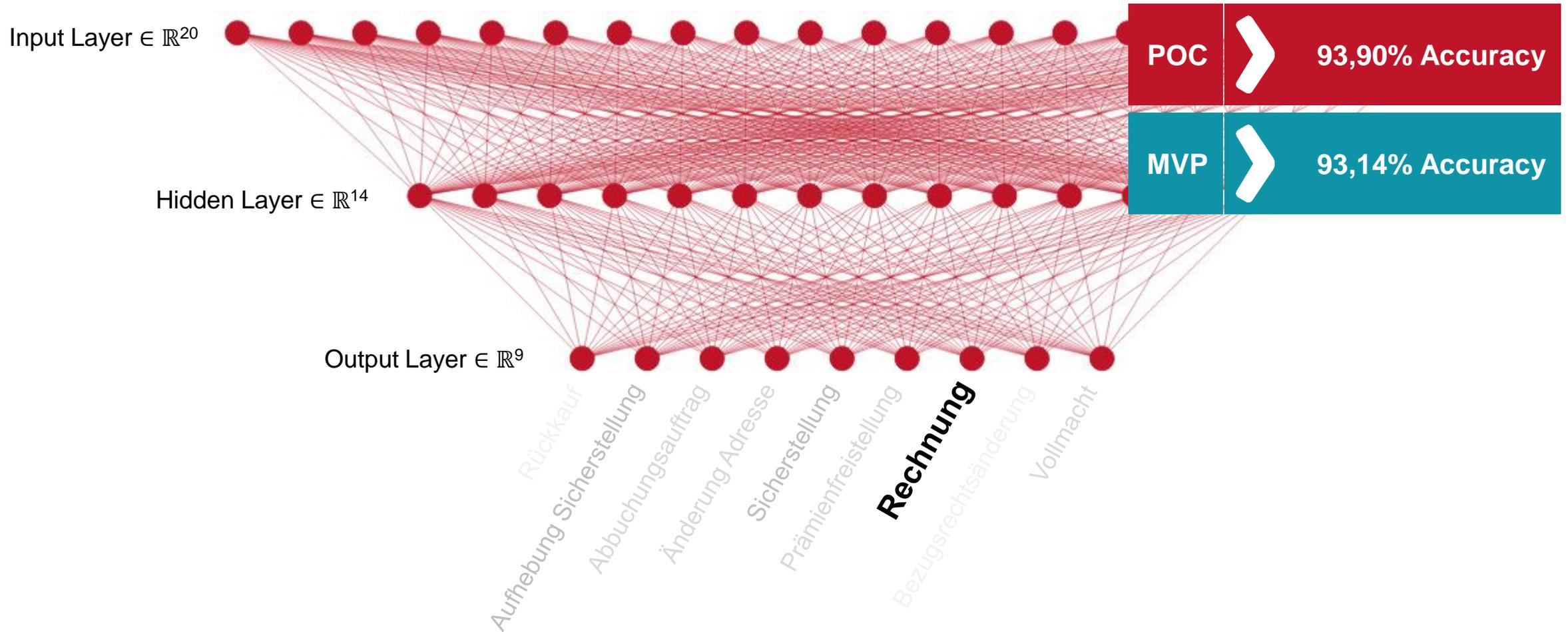
Rückkauf  
Aufhebung Sicherstellung  
Abbuchungsauftrag  
Änderung Adresse  
Sicherstellung  
Prämienfreistellung  
**Rechnung**  
Bezugsrechtsänderung  
Vollmacht

Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

# Modellierungsansätze – Simple Word Embedding Models (SWEM)

**ERGO**

A Munich Re company



# Formulare mit mehreren Eingangsklassen

## Änderung Adresse

**Der Versicherungsnehmer wünscht: (Beilage der Originalpolizze nicht erforderlich!)**  
(☒ Zutreffendes bitte ankreuzen)

**1 Namens-/Adressänderung**

für Versicherungsnehmer  
 für Versicherte Person

Bei Namensänderungen ist eine Fotokopie eines amtlichen Dokumentes (z.B. Heiratsurkunde) erforderlich.  
Adressänderungen werden ohne sonstige Dokumente vorgenommen.

Name:   
Akad. Grad, Nachname, Vorname

Adresse:   
Postleitzahl, Ort, Straße/Gasse/Platz, Hausnr., Türnr.

## Abbuchungsauftrag

**ERMÄCHTIGUNG ZUM EINZUG VON PRÄMIEN DURCH LASTSCHRIFTEN**

Name des Zahlungspflichtigen Polizzennummer Kontonummer des Zahlungspflichtigen

Kontoführendes Kreditinstitut des Zahlungspflichtigen Bankleitzahl

Hiemit ermächtige/n ich/wir/Sie widerruflich, die von mir/uns zu entrichtenden Zahlungen zu Lasten meines/unseres oben angeführten Kontos mittels Lastschrift einzuziehen. Damit ist auch meine/ unsere kontoführende Bank ermächtigt, die Lastschriften einzulösen, wobei für diese keine Verpflichtung zur Einlösung besteht, insbesondere dann, wenn mein/unser Konto die erforderliche Deckung nicht aufweist. Ich/Wir habe/n das Recht, innerhalb von 42 Kalendertagen ab Abbuchungstag ohne Angabe von Gründen die Rückbuchung bei meiner/unserer Bank zu veranlassen.

## Bezugsrechtsänderung

**2 Änderung des Bezugsrechtes: Keine weiteren Dokumente erforderlich.**  
Das bestehende Bezugsrecht zu Gunsten eines bestehenden Vinkulargläubigers bleibt unverändert.

bei Ableben:      
Nachname, Vorname Geb. Datum (unbedingt anführen) weiblich männlich

bei Erleben:      
Nachname, Vorname Geb. Datum (unbedingt anführen) weiblich männlich

## Prämienfreistellung

**9**  **Prämienfreistellung ab 01.**   
Ich nehme zur Kenntnis, dass der Versicherungsschutz aus allfälligen, im Versicheru  
erlischt.

**Reaktivierung des prämienfreigestellten Vertrages ab 01.**   
Bei Reaktivierung eines prämienfreigestellten Vertrages nach Tarif LifeInvest bzw.  
SinfoniInvest wurde das Kundeninformationsdokument (KID) des/der gewählten For

# Formulare mit mehreren Eingangsklassen

## Änderung Adresse

**Der Versicherungsnehmer wünscht: (Beilage der Originalpolizze nicht erforderlich!)**  
(☒ Zutreffendes bitte ankreuzen)

**1 Namens-/Adressänderung**

für Versicherungsnehmer  
 für Versicherte Person

Bei Namensänderungen ist eine Fotokopie eines amtlichen Dokumentes (z.B. Heiratsurkunde) erforderlich. Adressänderungen werden ohne sonstige Dokumente vorgenommen.

Name:   
Akad. Grad, Nachname, Vorname

Adresse:  00000, Musterstadt, Musterstraße 0  
Postleitzahl, Ort, Straße/Gasse/Platz, Hausnr., Türnr.

Relevantes „n-gram“

## Abbuchungsauftrag

**ERMÄCHTIGUNG ZUM EINZUG VON PRÄMIEN DURCH LASTSCHRIFTEN**

Name des Zahlungspflichtigen Polizzennummer Kontonummer des Zahlungspflichtigen

Kontoführendes Kreditinstitut des Zahlungspflichtigen Bankleitzahl

Hiemit ermächtige/n ich/wir/Sie widerruflich, die von mir/uns zu entrichtenden Zahlungen zu Lasten meines/unseres oben angeführten Kontos mittels Lastschrift einzuziehen. Damit ist auch meine/ unsere kontoführende Bank ermächtigt, die Lastschriften einzulösen, wobei für diese keine Verpflichtung zur Einlösung besteht, insbesondere dann, wenn mein/unser Konto die erforderliche Deckung nicht aufweist. Ich/Wir habe/n das Recht, innerhalb von 42 Kalendertagen ab Abbuchungstag ohne Angabe von Gründen die Rückbuchung bei meiner/unserer Bank zu veranlassen.

## Bezugsrechtsänderung

**2 Änderung des Bezugsrechtes: Keine weiteren Dokumente erforderlich.**  
Das bestehende Bezugsrecht zu Gunsten eines bestehenden Vinkulargläubigers bleibt unverändert.

bei Ableben:      
Nachname, Vorname Geb. Datum (unbedingt anführen) weiblich männlich

bei Erleben:      
Nachname, Vorname Geb. Datum (unbedingt anführen) weiblich männlich

## Prämienfreistellung

**9**  **Prämienfreistellung ab 01.**   
Ich nehme zur Kenntnis, dass der Versicherungsschutz aus allfälligen, im Versicheru erlischt.

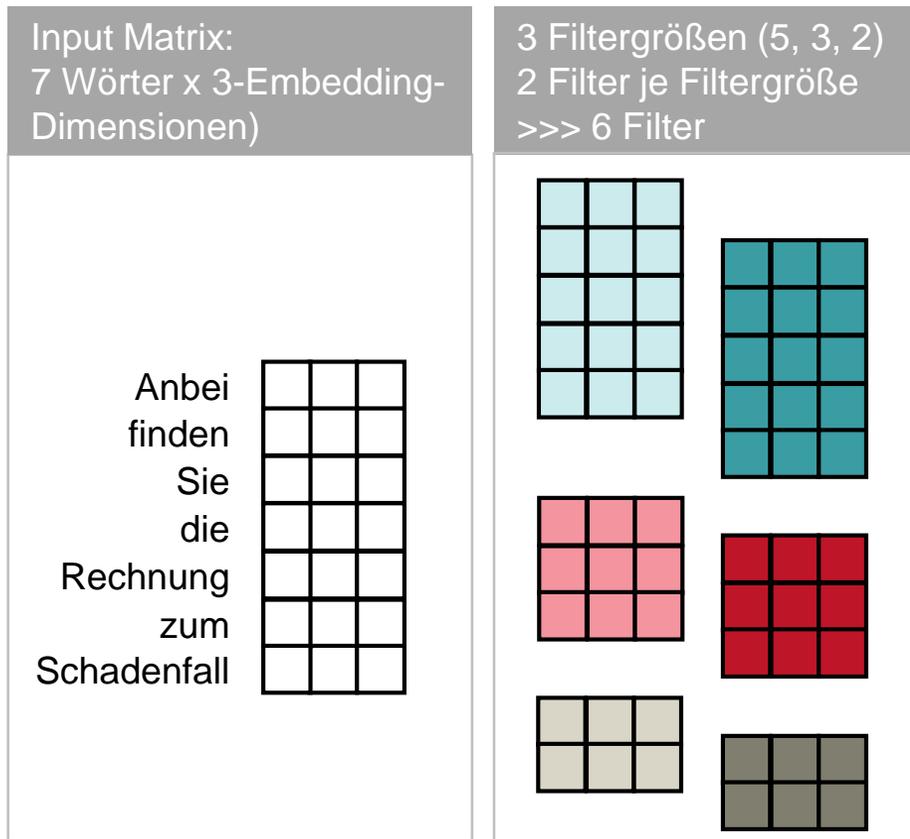
**Reaktivierung des prämienfreigestellten Vertrages ab 01.**   
Bei Reaktivierung eines prämienfreigestellten Vertrages nach Tarif LifeInvest bzw. SinfoniInvest wurde das Kundeninformationsdokument (KID) des/der gewählten For

# Modellierungsansätze – Convolutional Neural Networks (CNN)

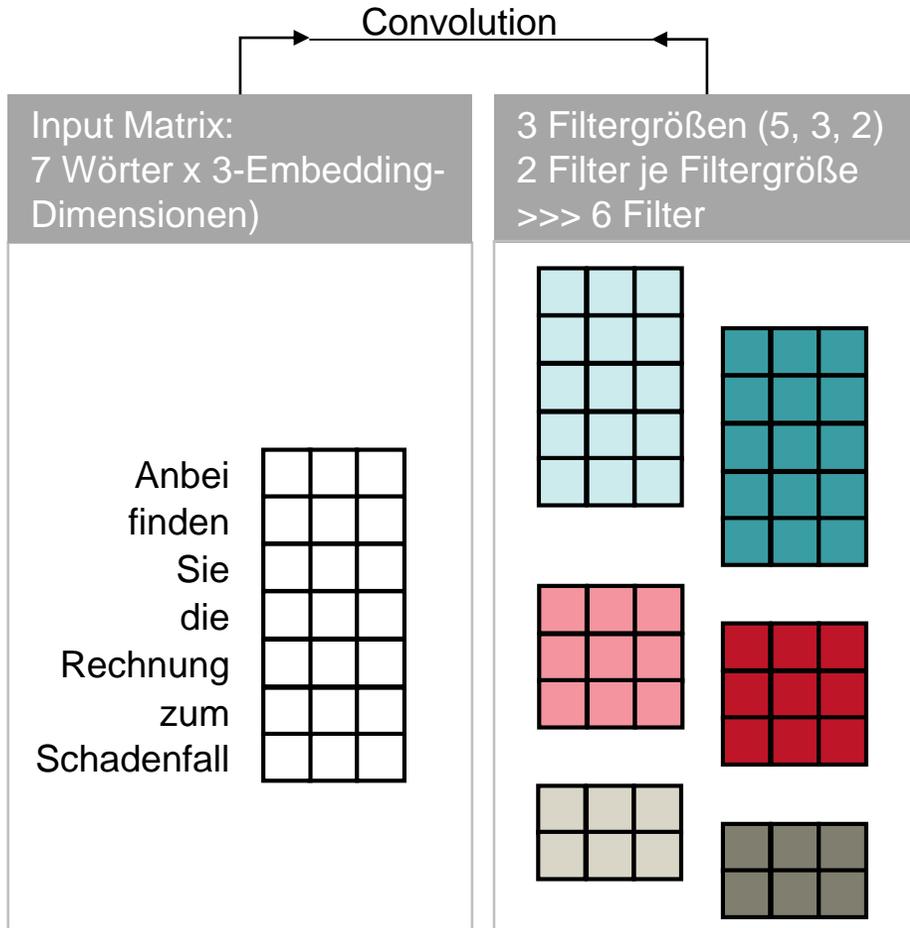
Input Matrix:  
7 Wörter x 3-Embedding-  
Dimensionen)

Anbei			
finden			
Sie			
die			
Rechnung			
zum			
Schadenfall			

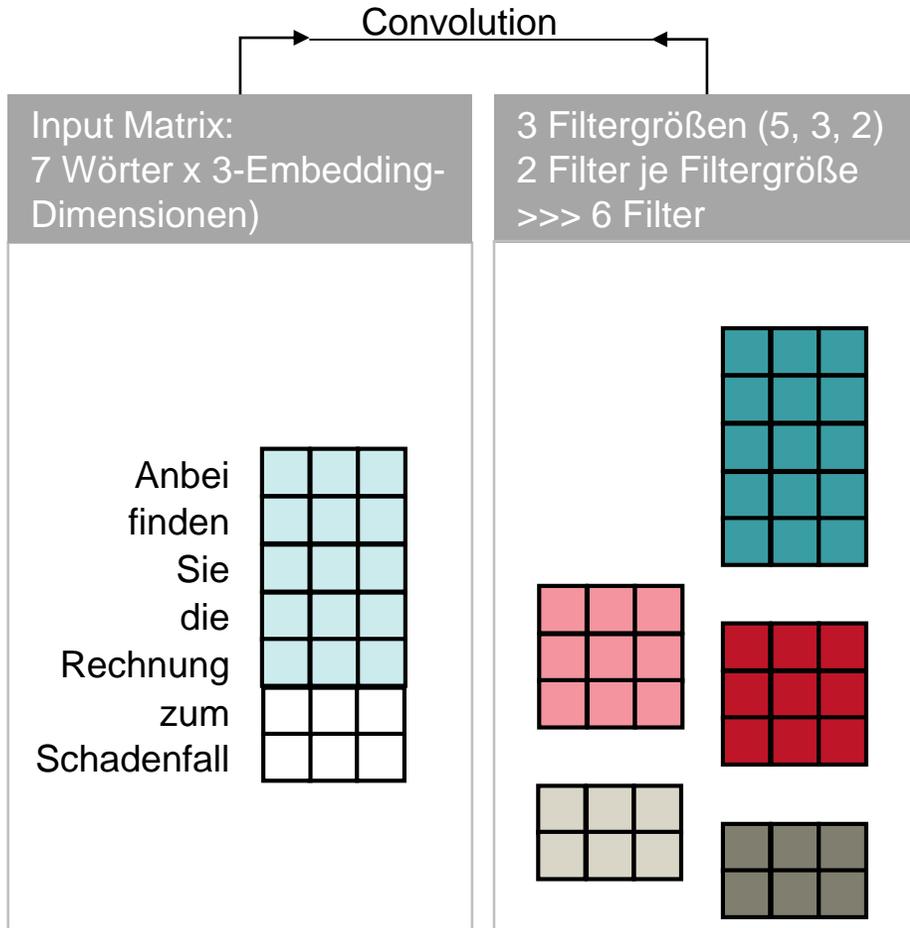
# Modellierungsansätze – Convolutional Neural Networks (CNN)



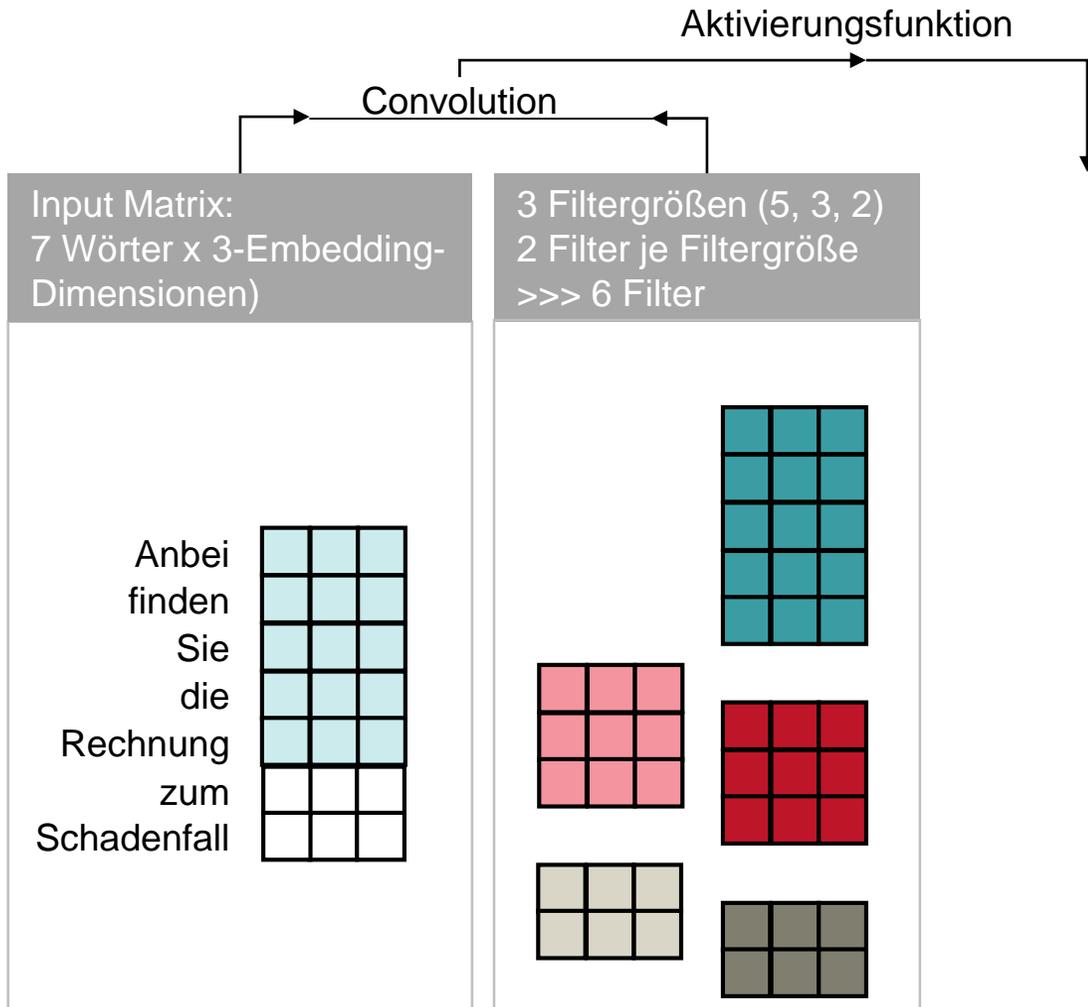
# Modellierungsansätze – Convolutional Neural Networks (CNN)



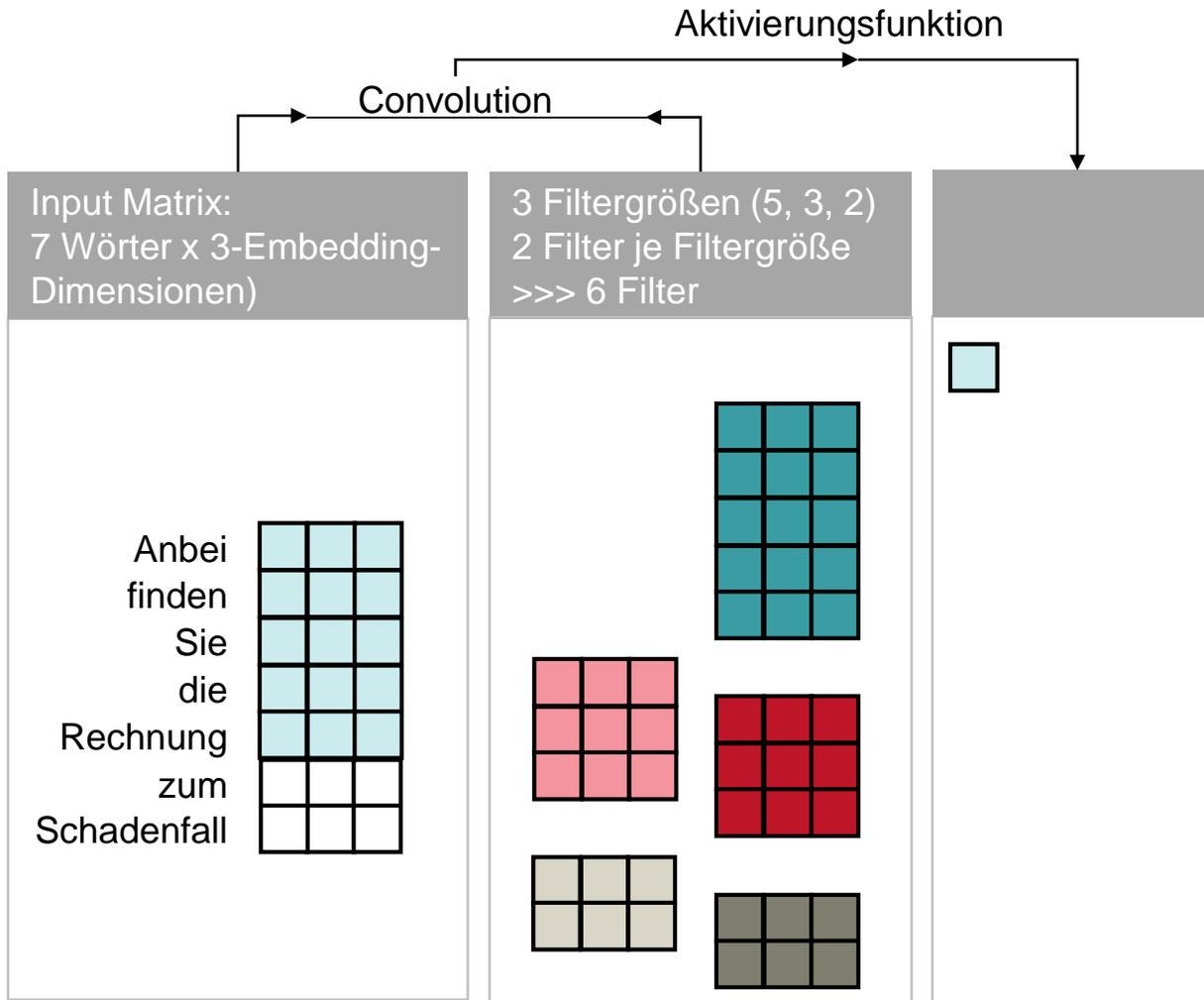
# Modellierungsansätze – Convolutional Neural Networks (CNN)



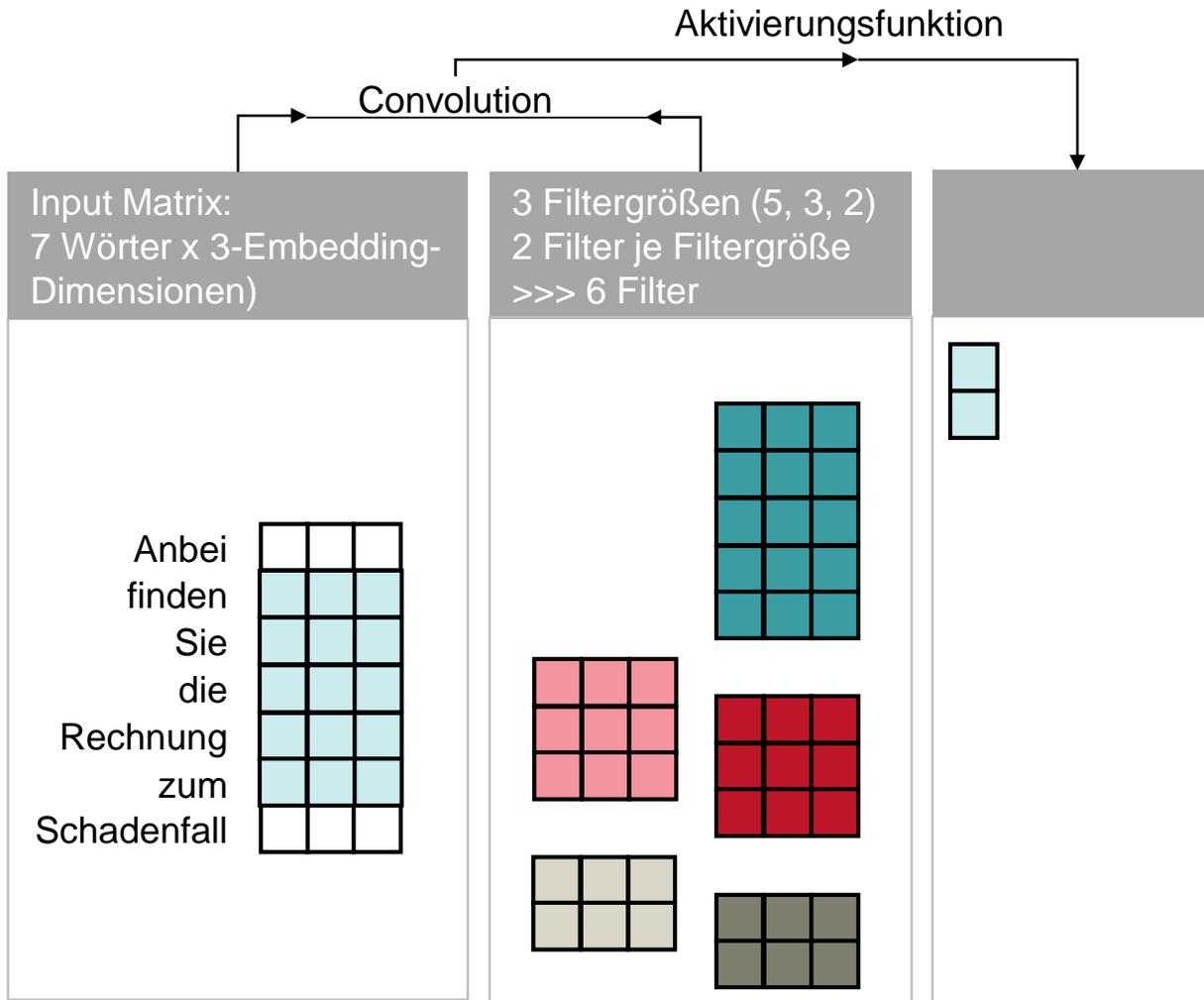
# Modellierungsansätze – Convolutional Neural Networks (CNN)



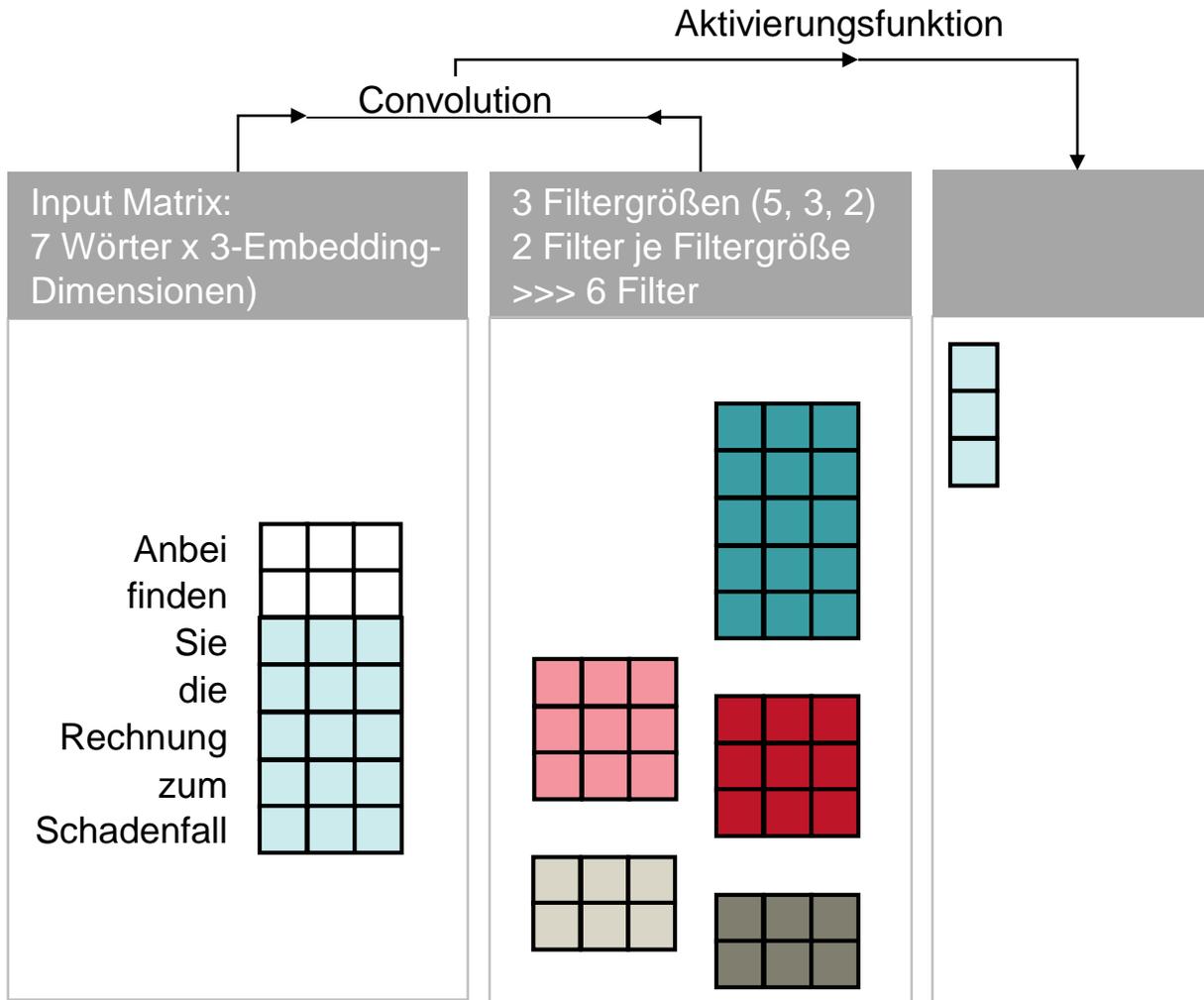
# Modellierungsansätze – Convolutional Neural Networks (CNN)



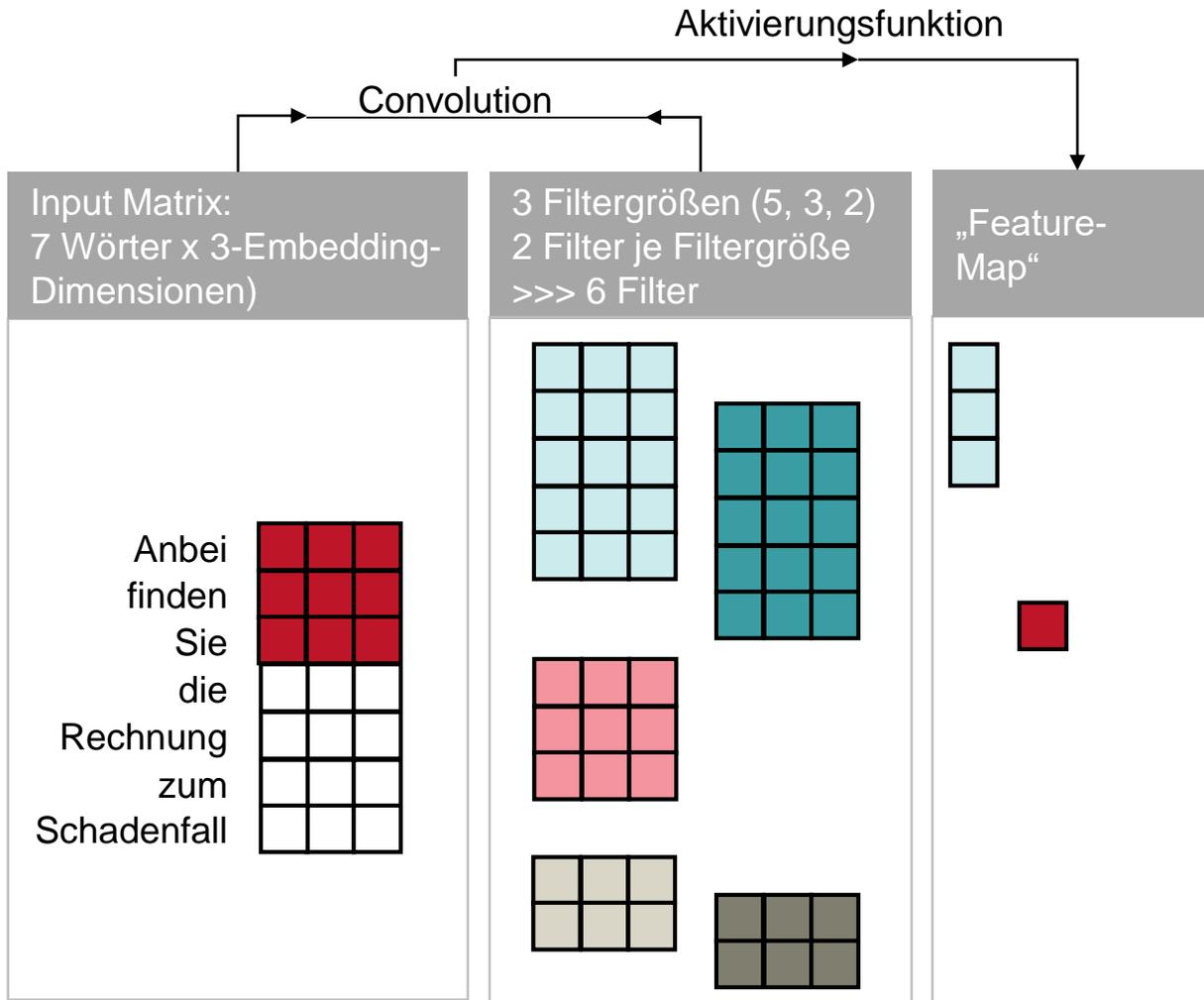
# Modellierungsansätze – Convolutional Neural Networks (CNN)



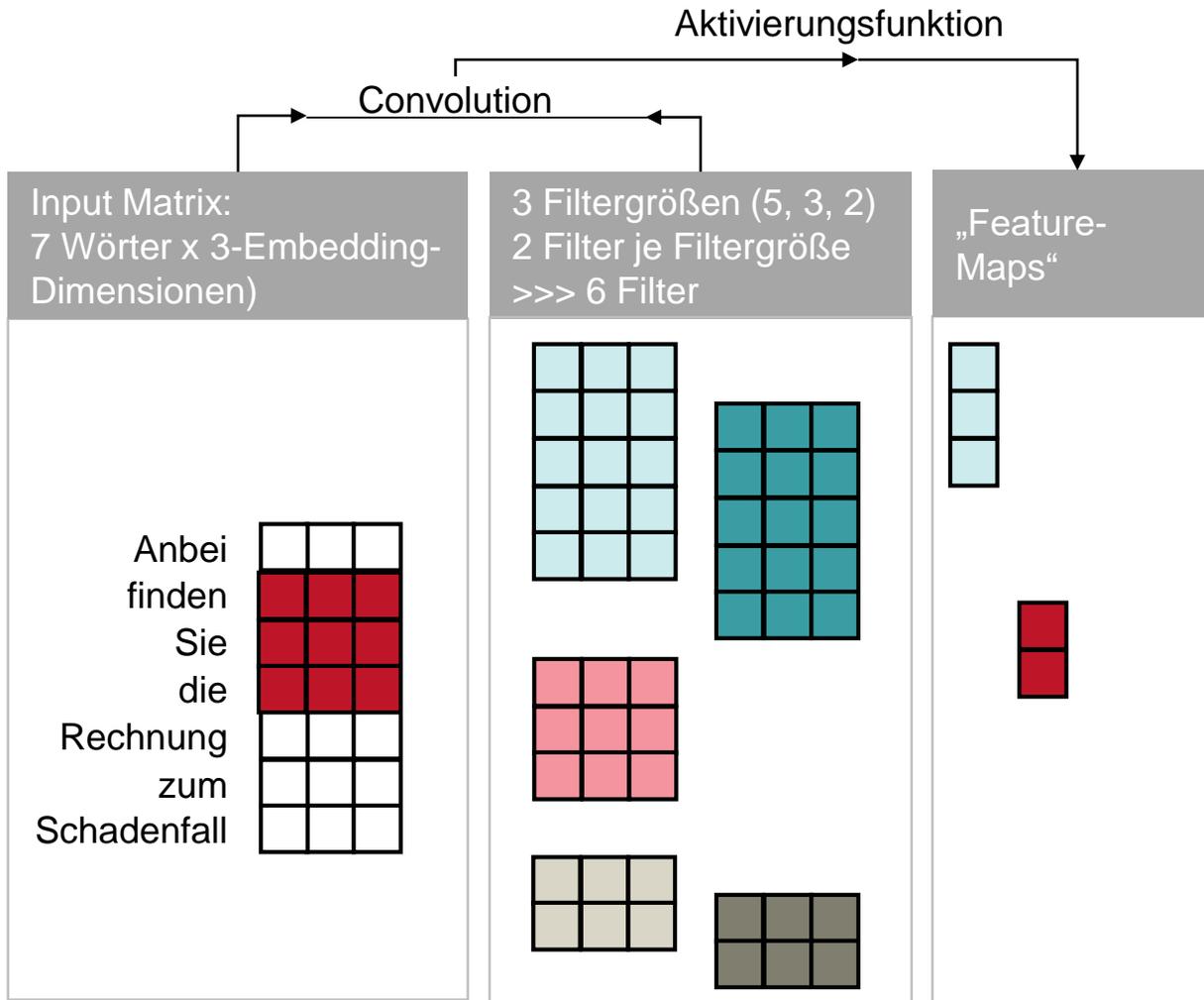
# Modellierungsansätze – Convolutional Neural Networks (CNN)



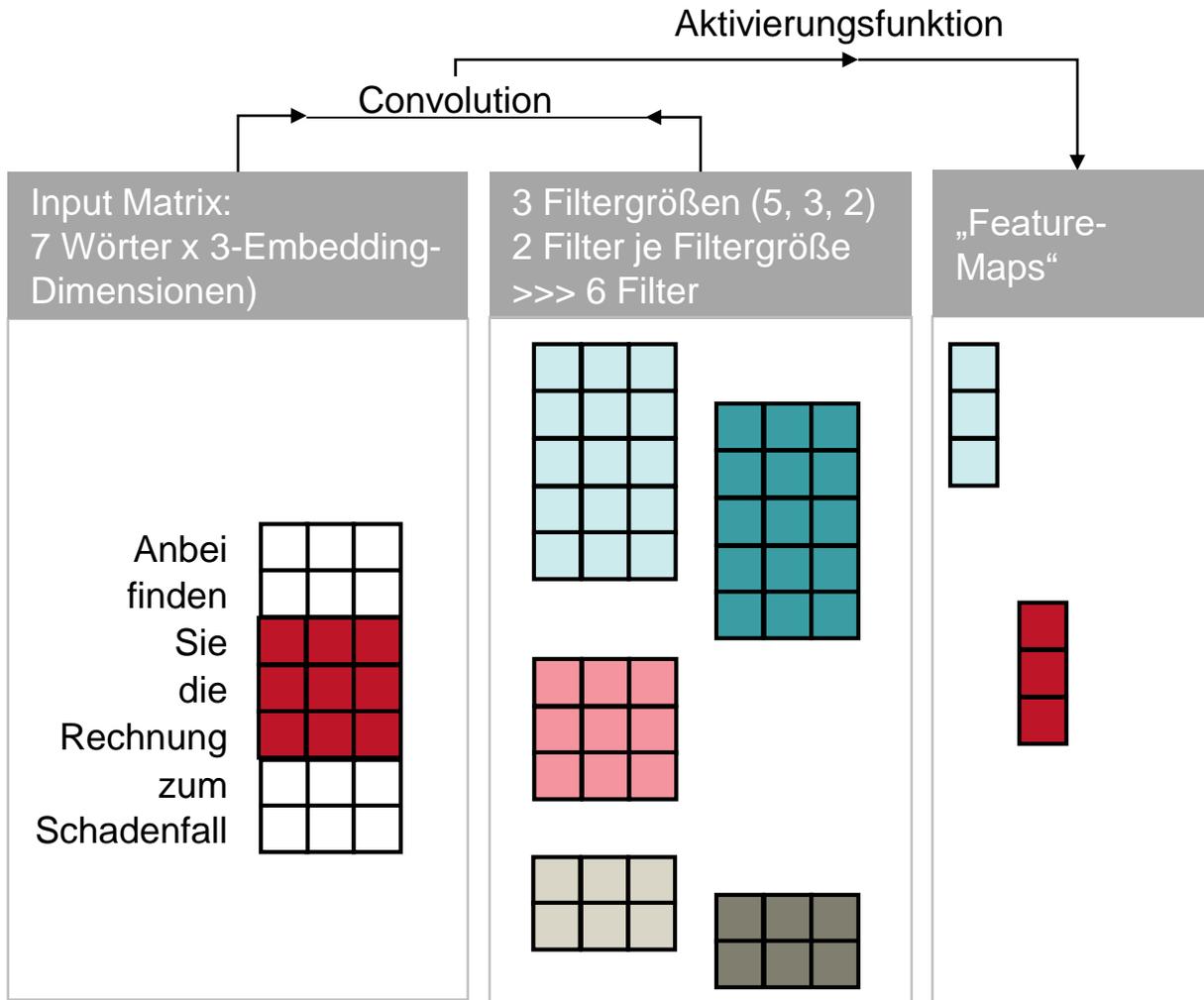
# Modellierungsansätze – Convolutional Neural Networks (CNN)



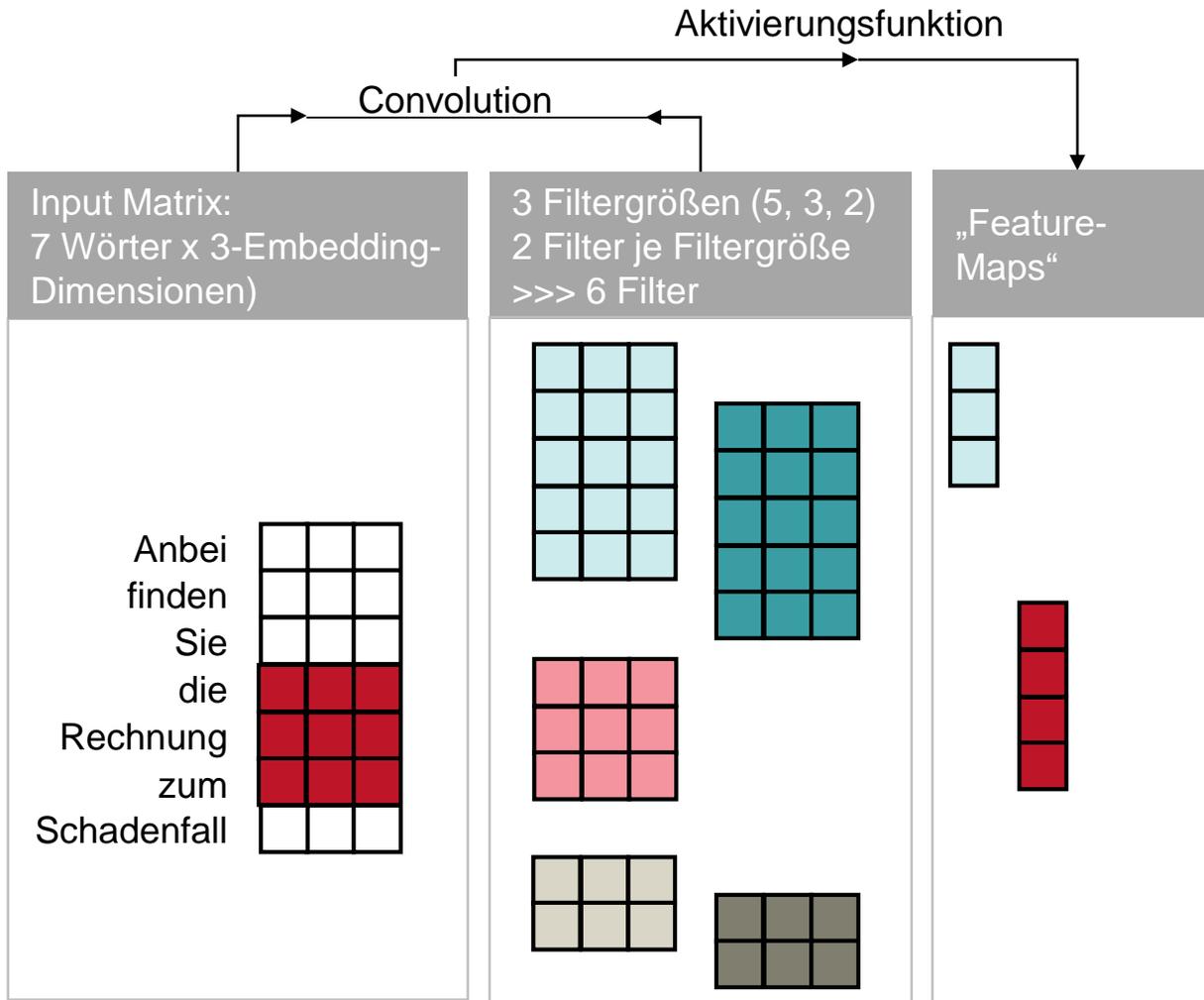
# Modellierungsansätze – Convolutional Neural Networks (CNN)



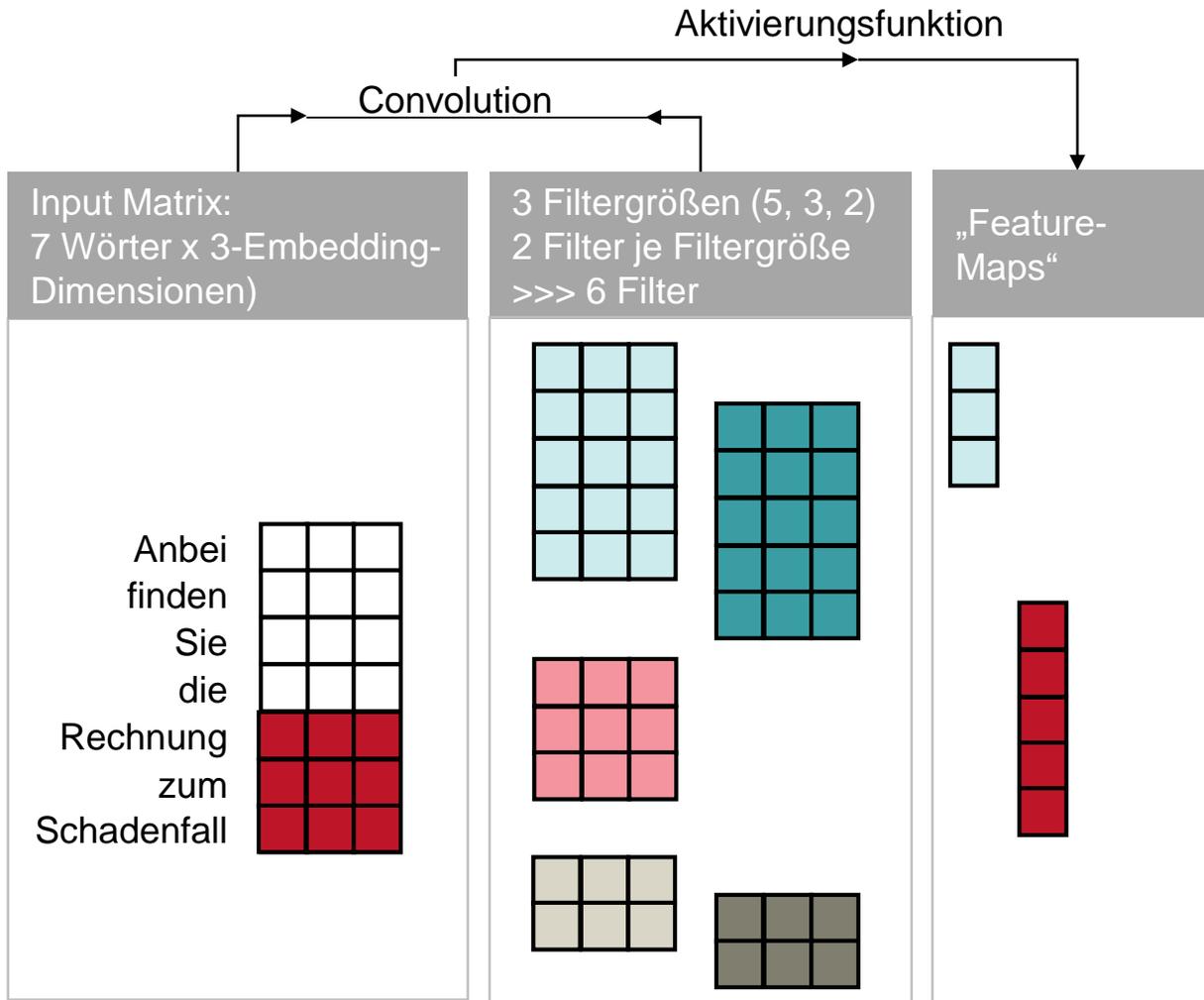
# Modellierungsansätze – Convolutional Neural Networks (CNN)



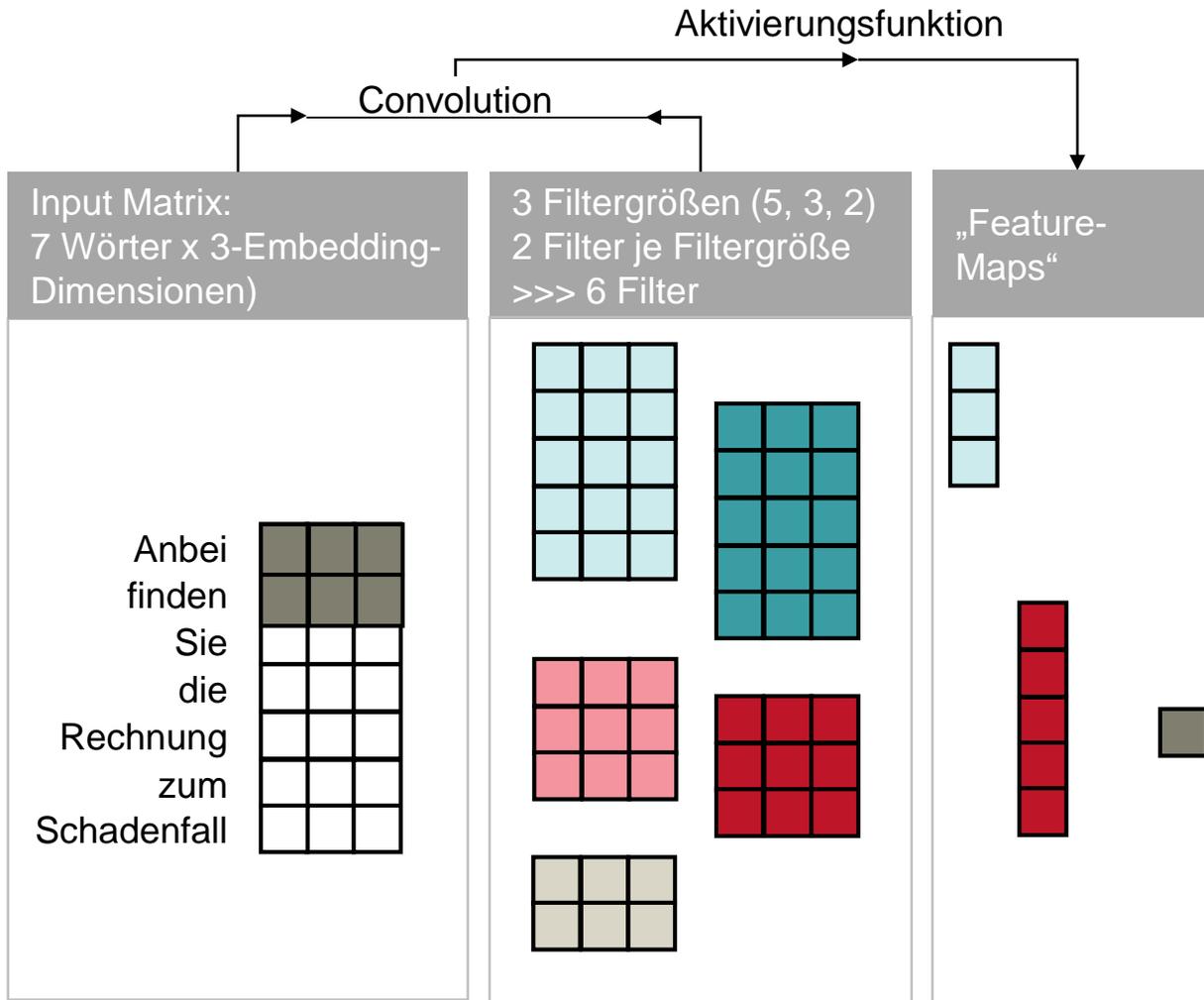
# Modellierungsansätze – Convolutional Neural Networks (CNN)



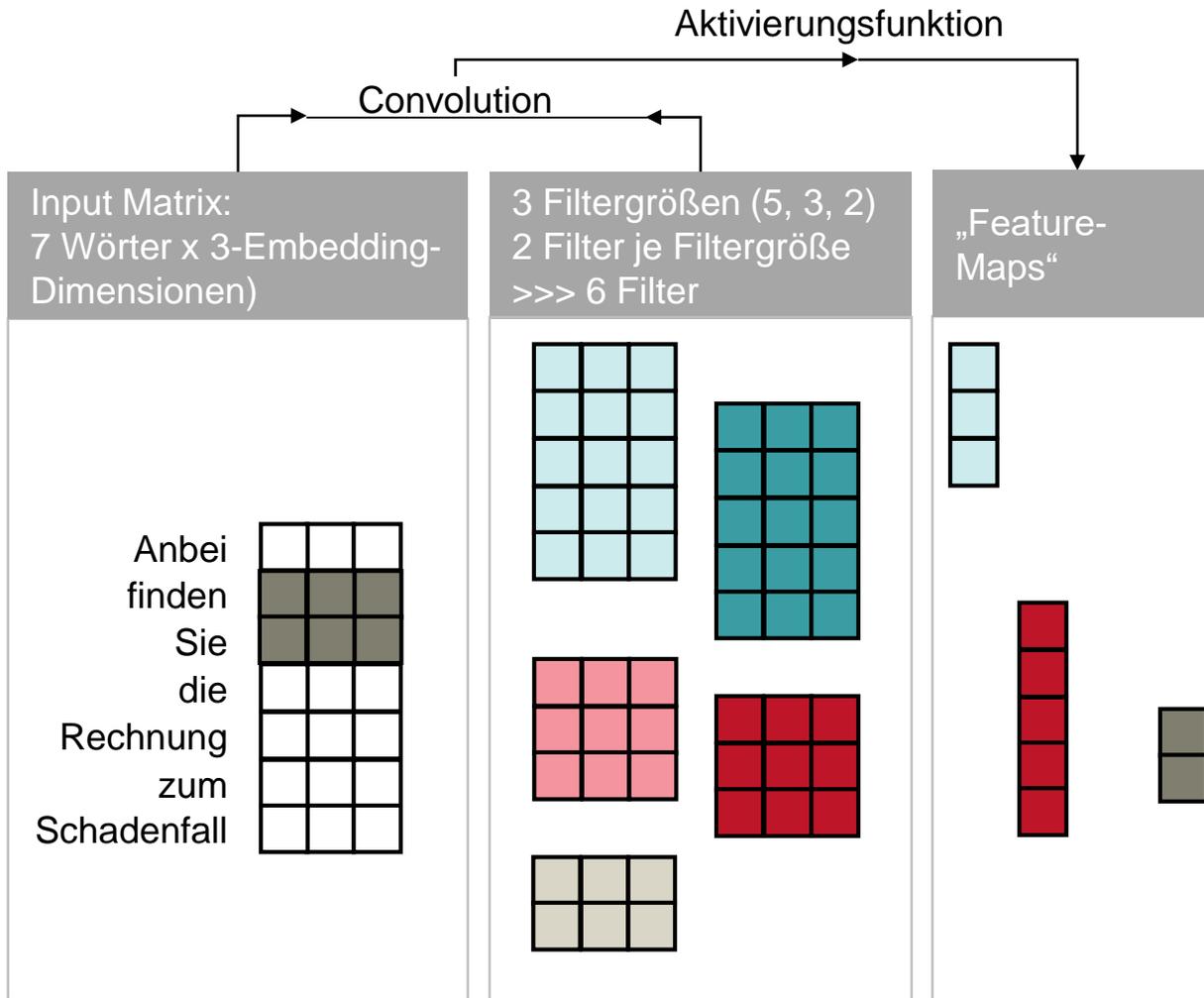
# Modellierungsansätze – Convolutional Neural Networks (CNN)



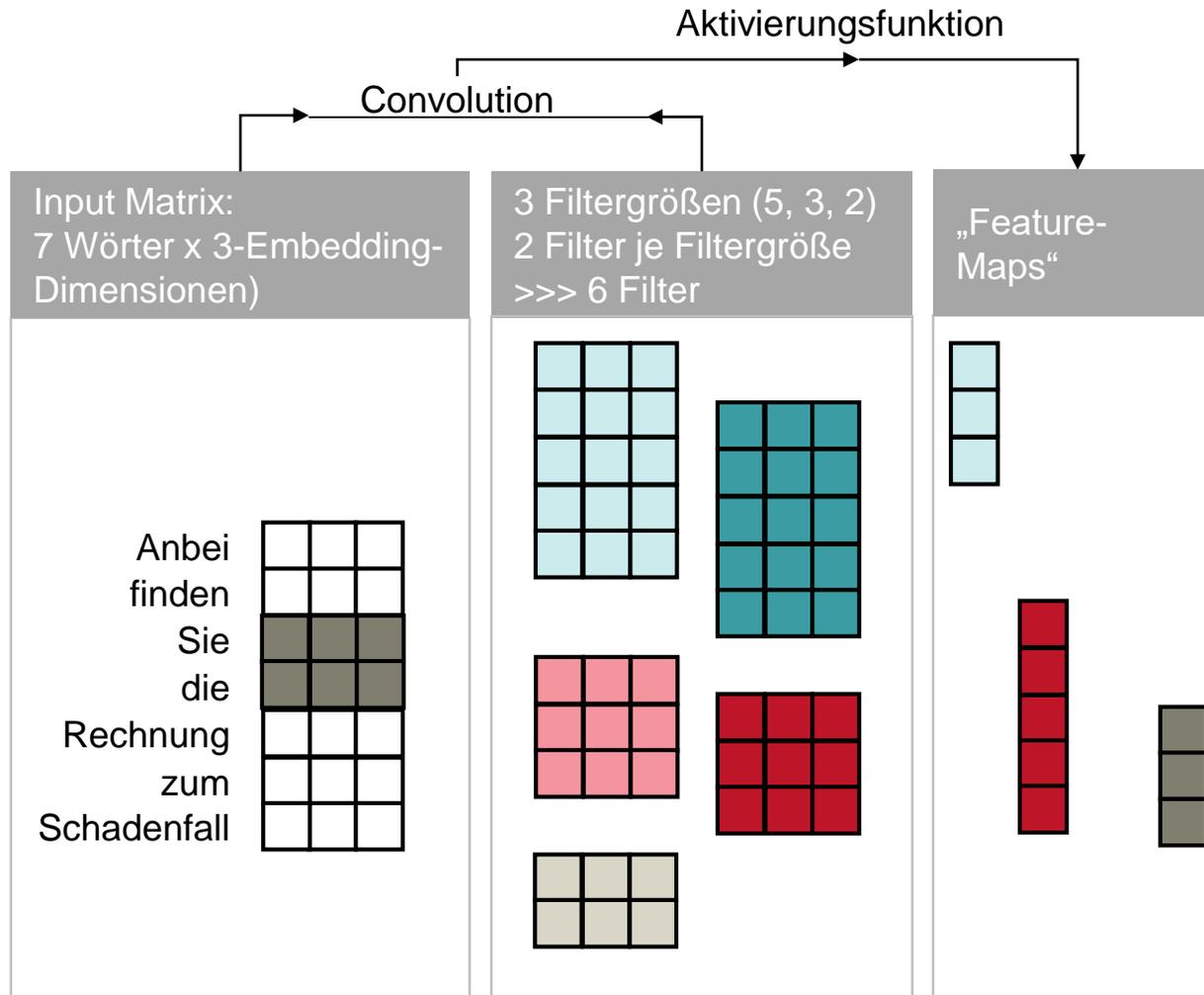
# Modellierungsansätze – Convolutional Neural Networks (CNN)



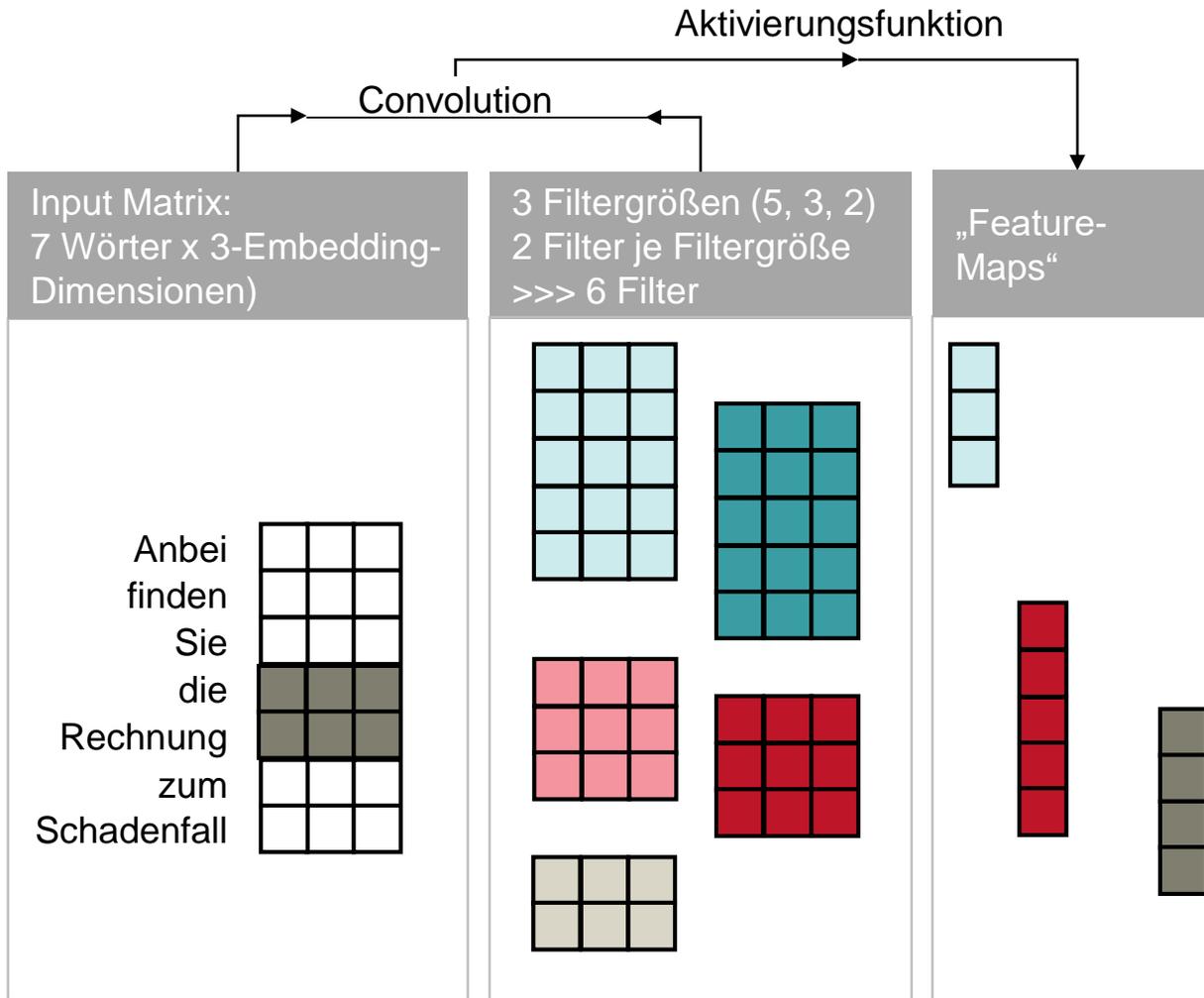
# Modellierungsansätze – Convolutional Neural Networks (CNN)



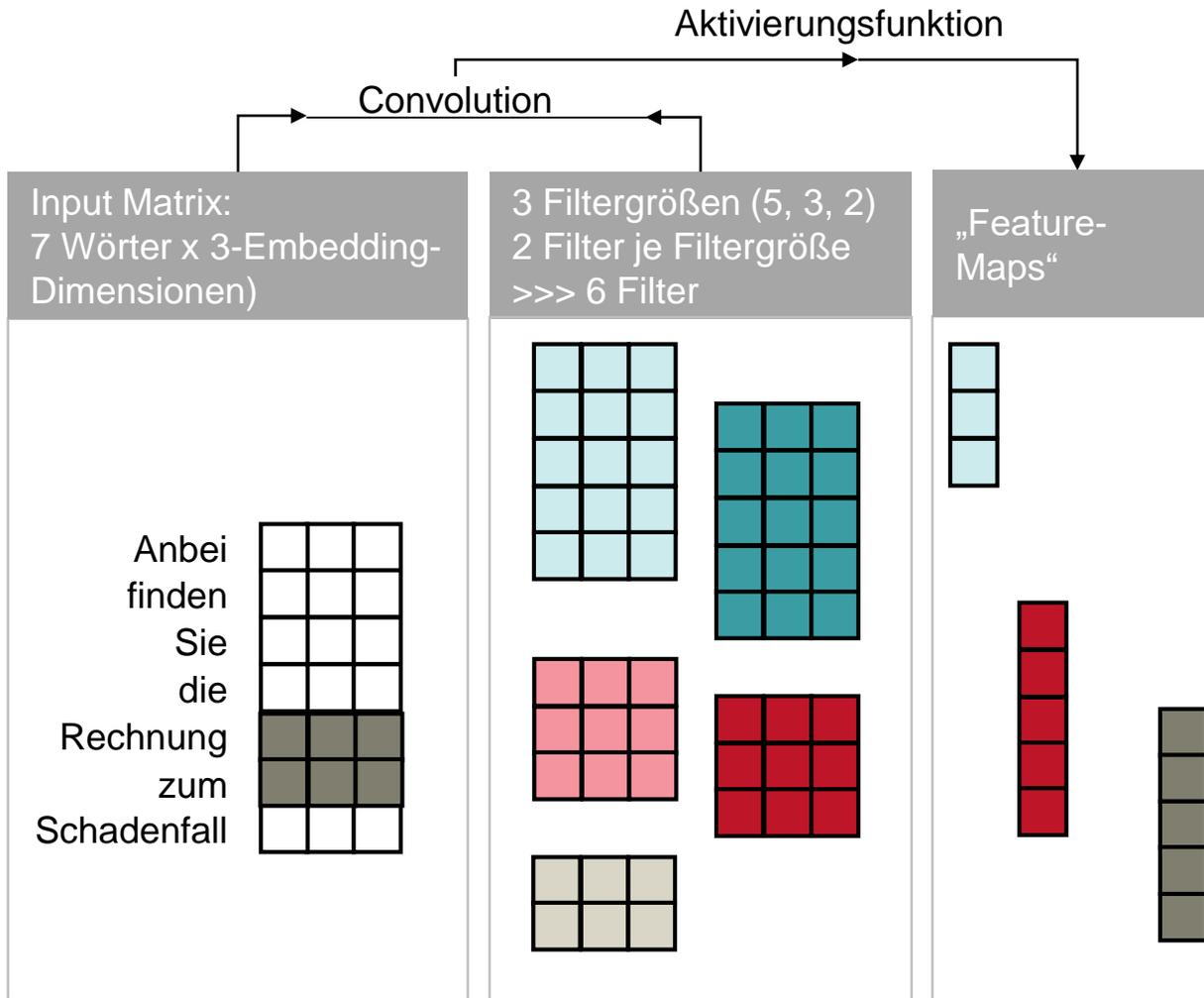
# Modellierungsansätze – Convolutional Neural Networks (CNN)



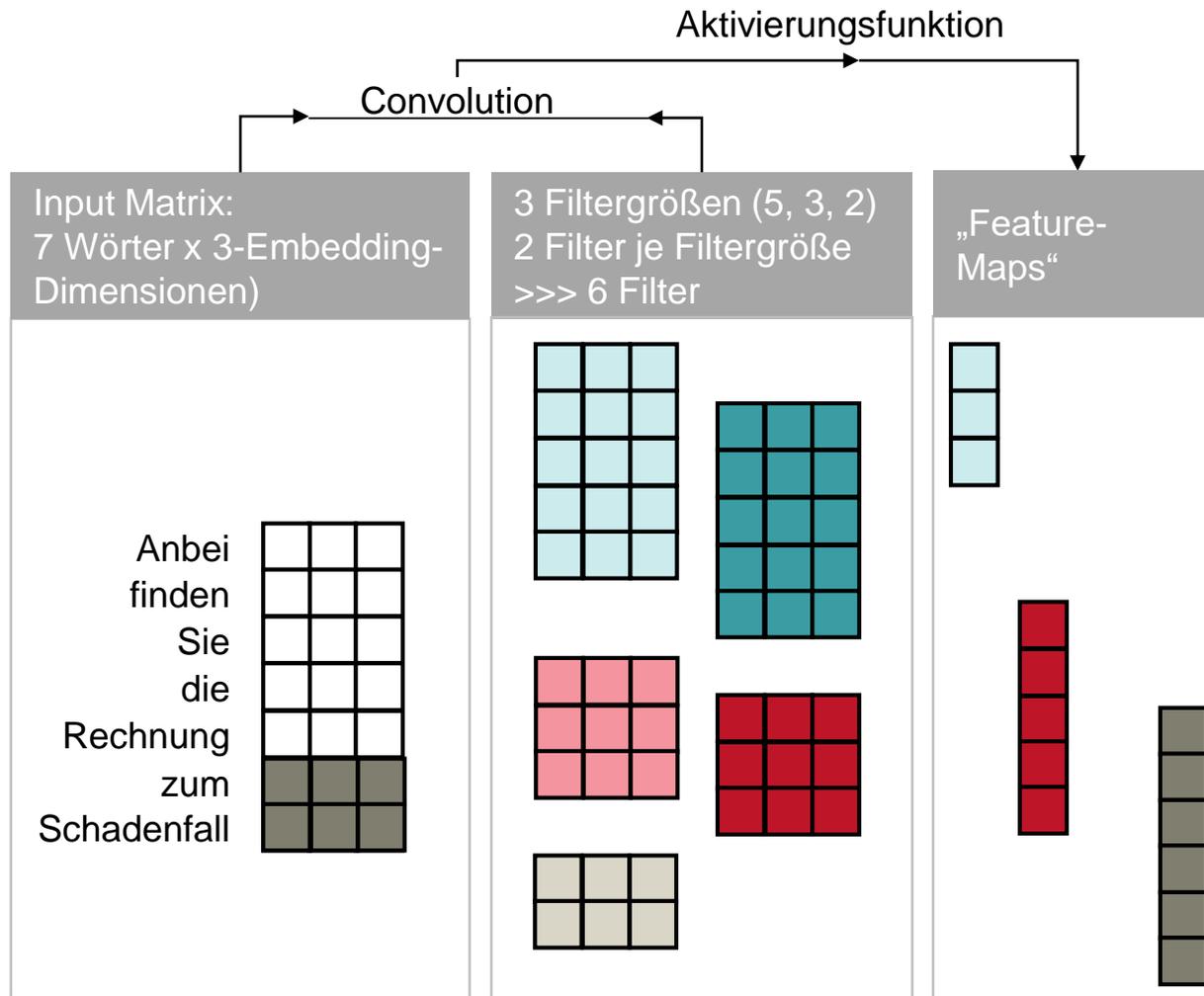
# Modellierungsansätze – Convolutional Neural Networks (CNN)



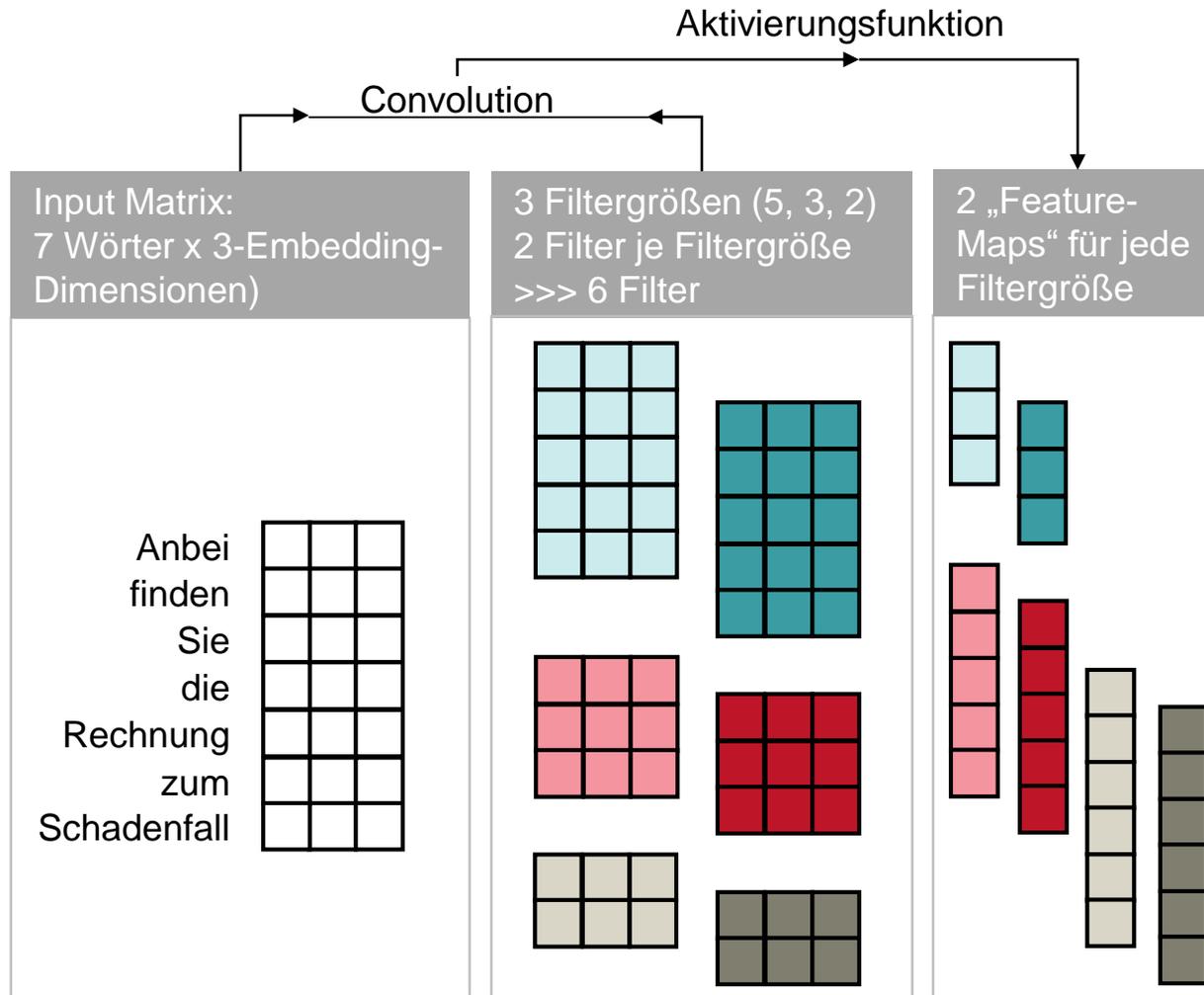
# Modellierungsansätze – Convolutional Neural Networks (CNN)



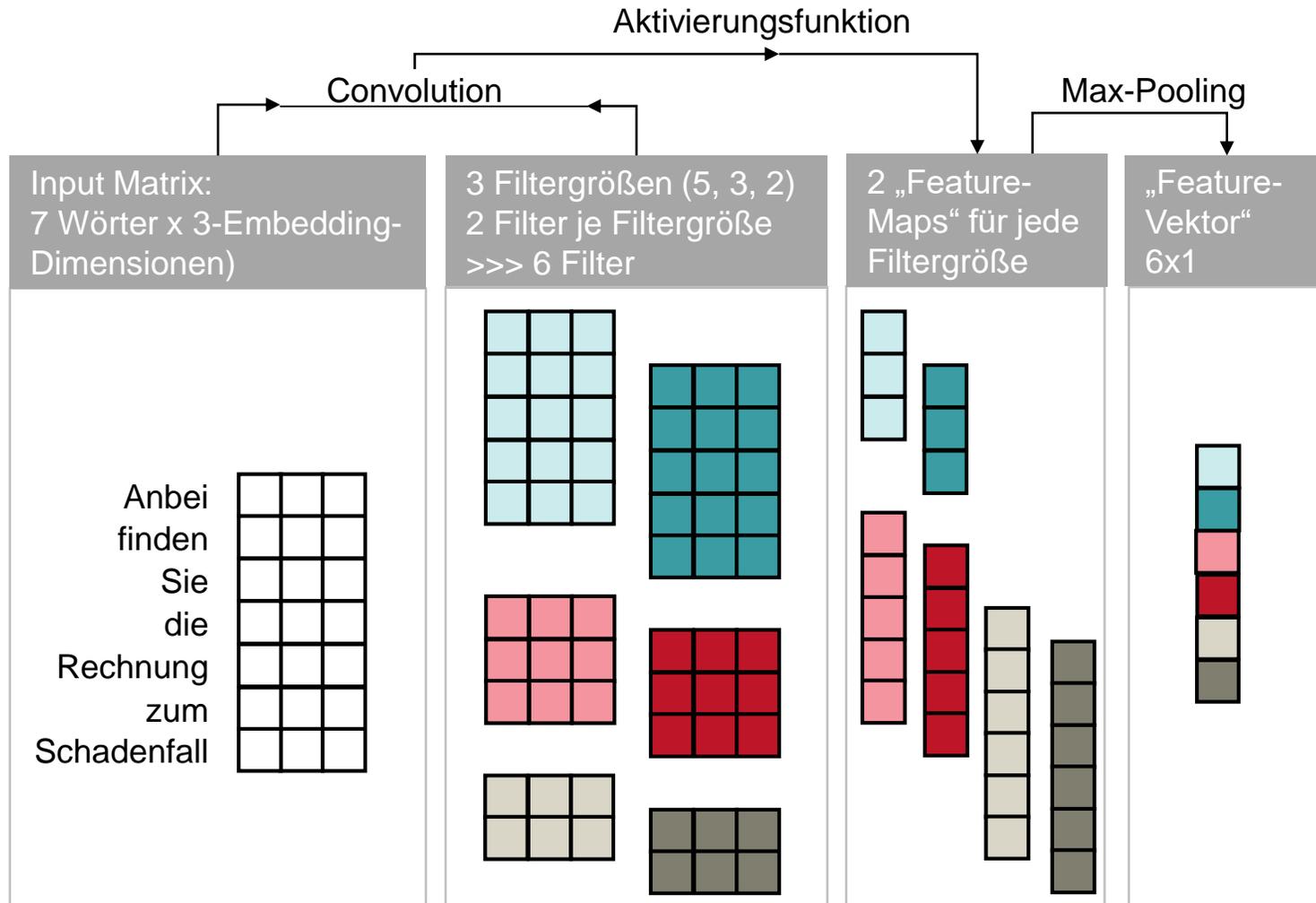
# Modellierungsansätze – Convolutional Neural Networks (CNN)



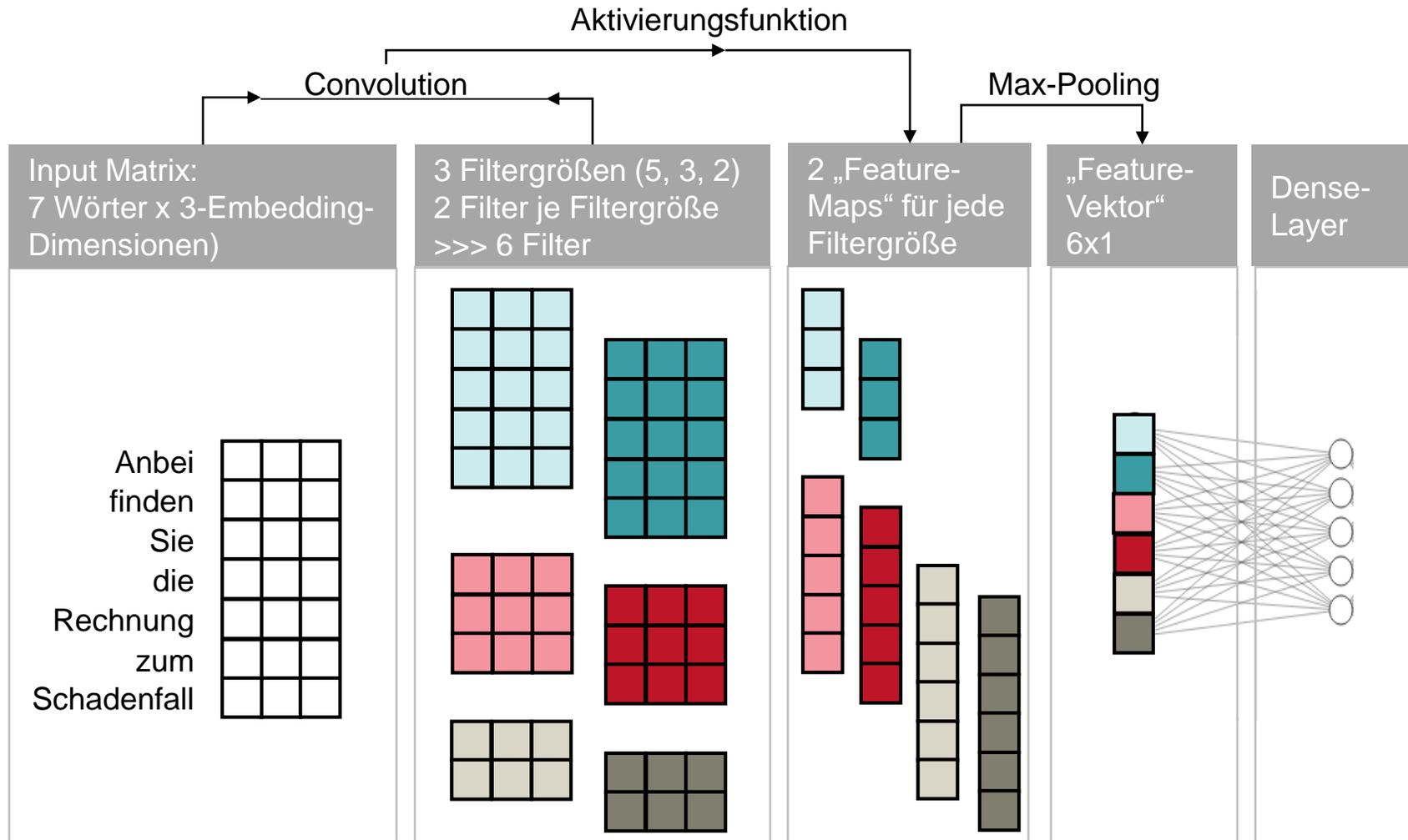
# Modellierungsansätze – Convolutional Neural Networks (CNN)



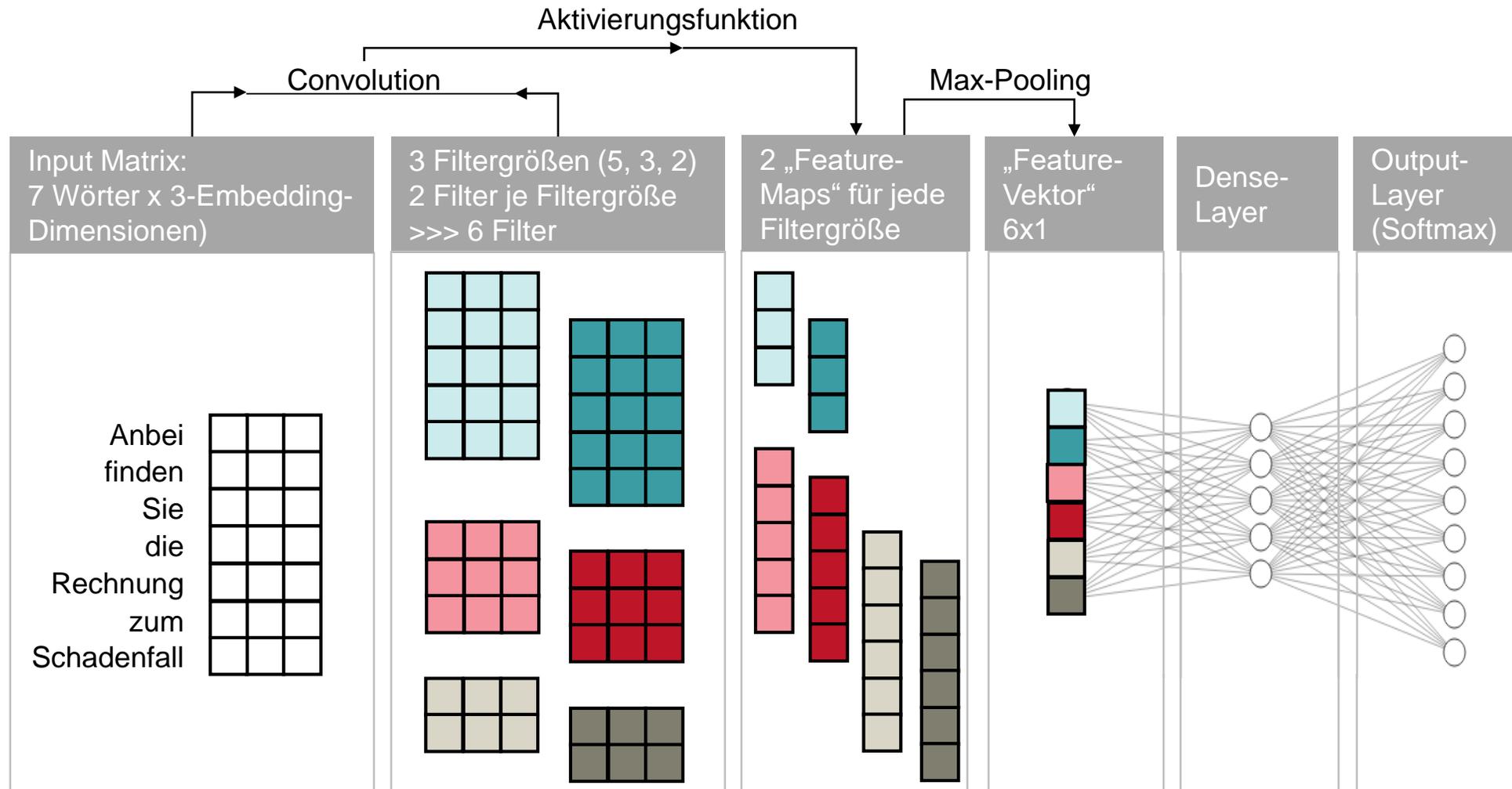
# Modellierungsansätze – Convolutional Neural Networks (CNN)



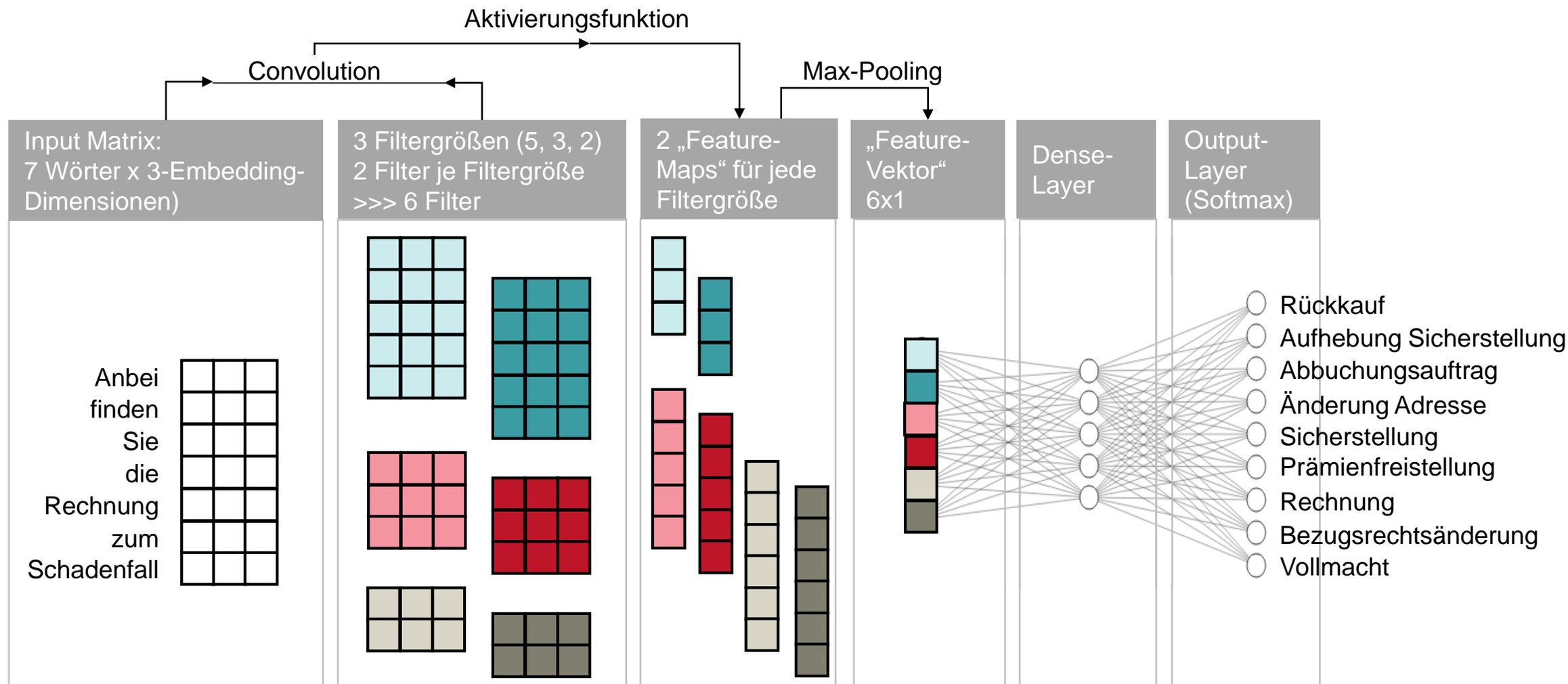
# Modellierungsansätze – Convolutional Neural Networks (CNN)



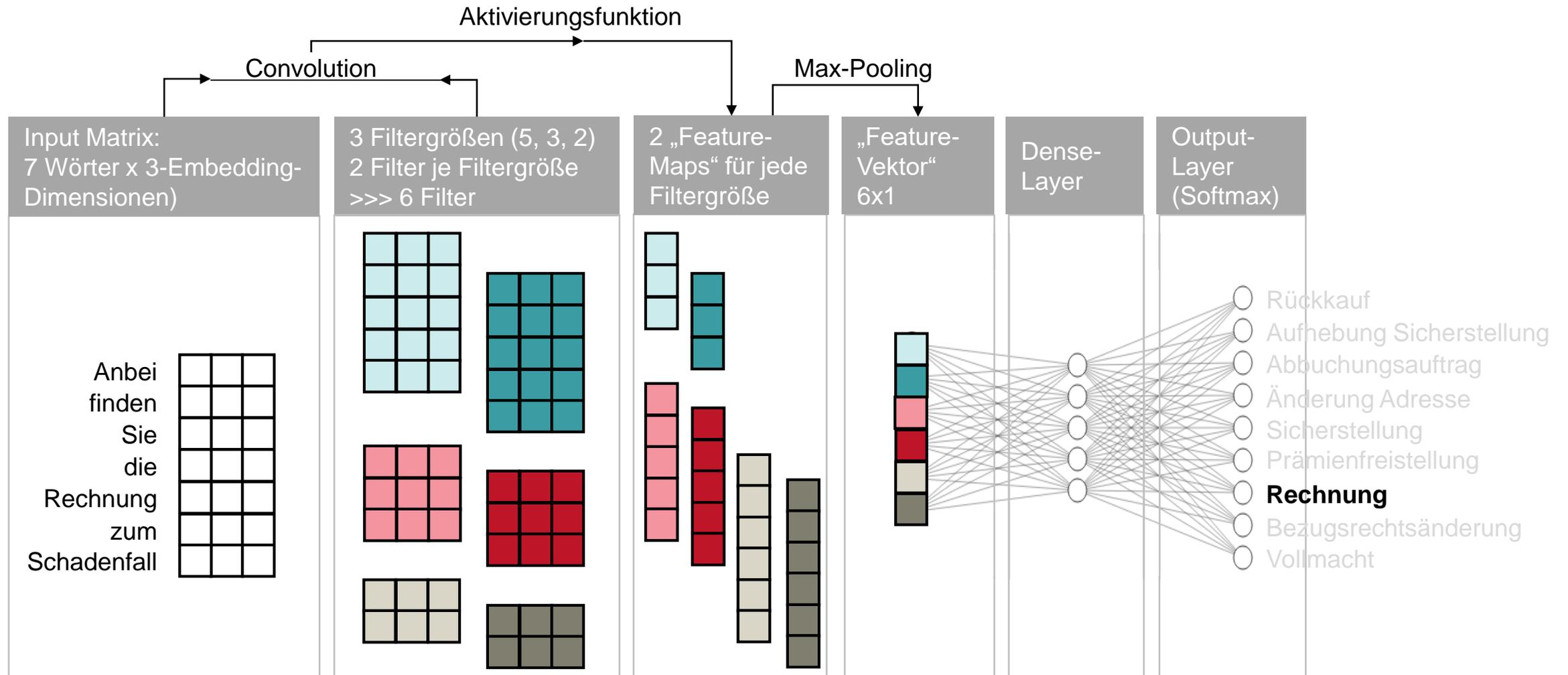
# Modellierungsansätze – Convolutional Neural Networks (CNN)



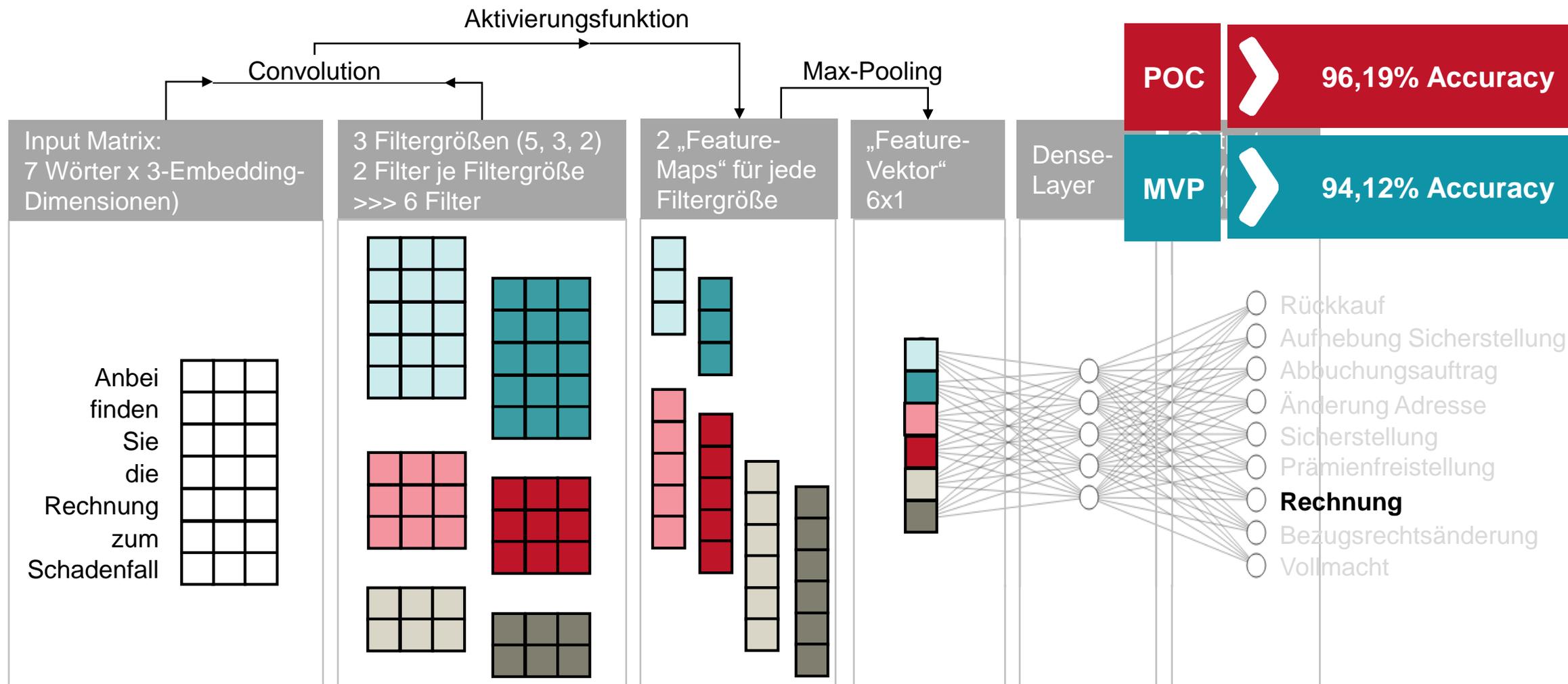
# Modellierungsansätze – Convolutional Neural Networks (CNN)



# Modellierungsansätze – Convolutional Neural Networks (CNN)

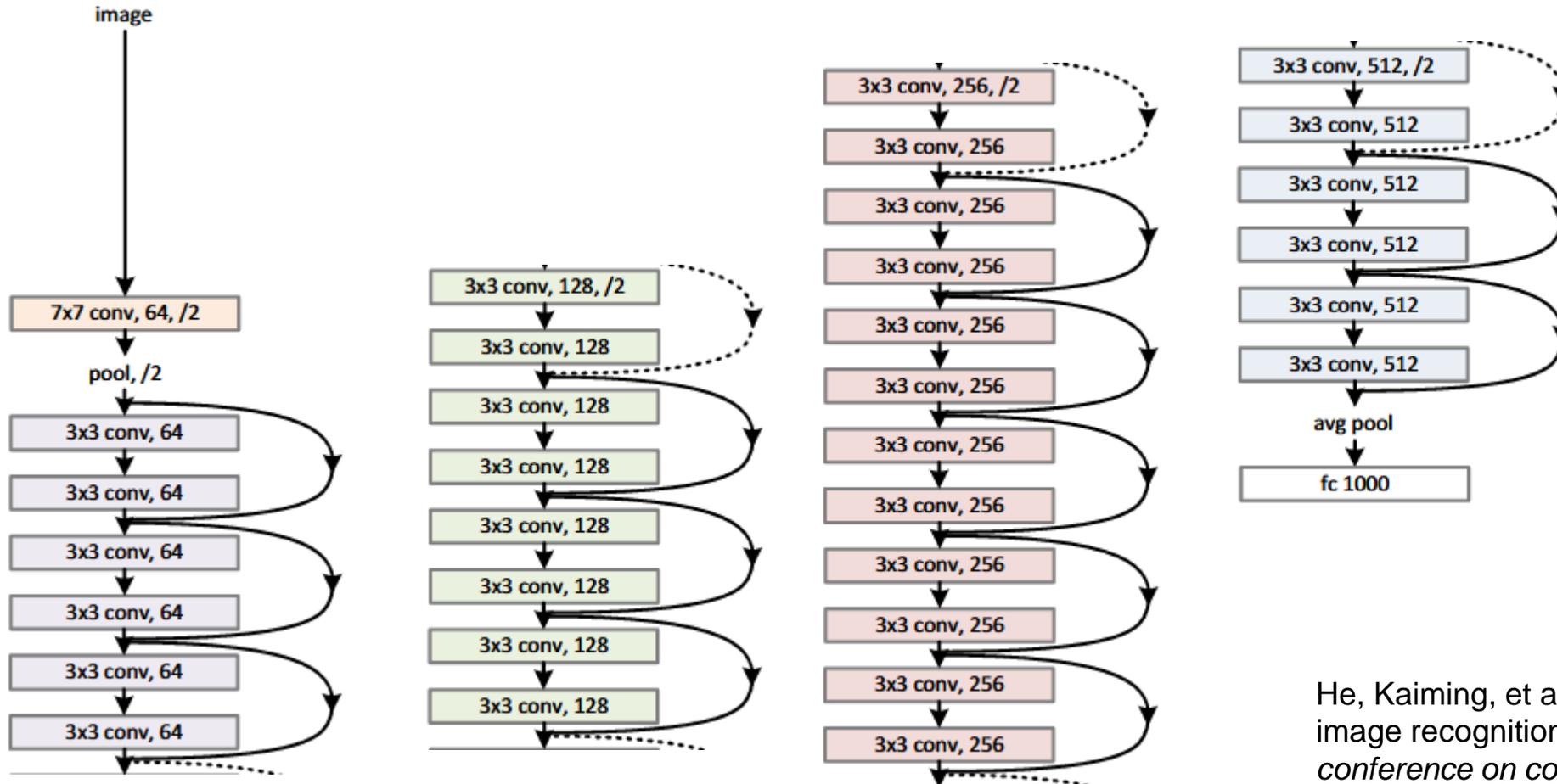


# Modellierungsansätze – Convolutional Neural Networks (CNN)



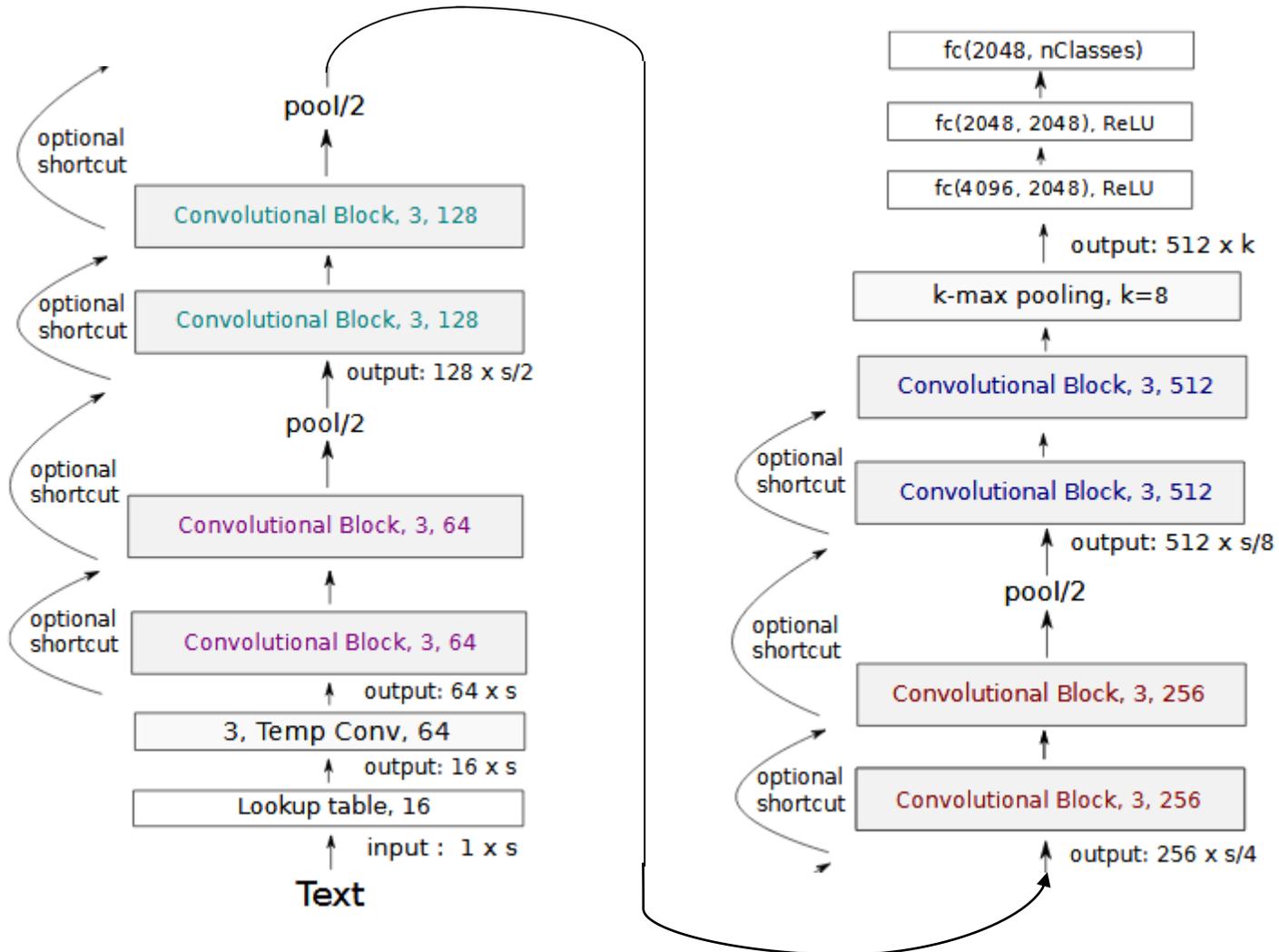
# Was in der Bilderkennung funktioniert...

## 34-layer residual



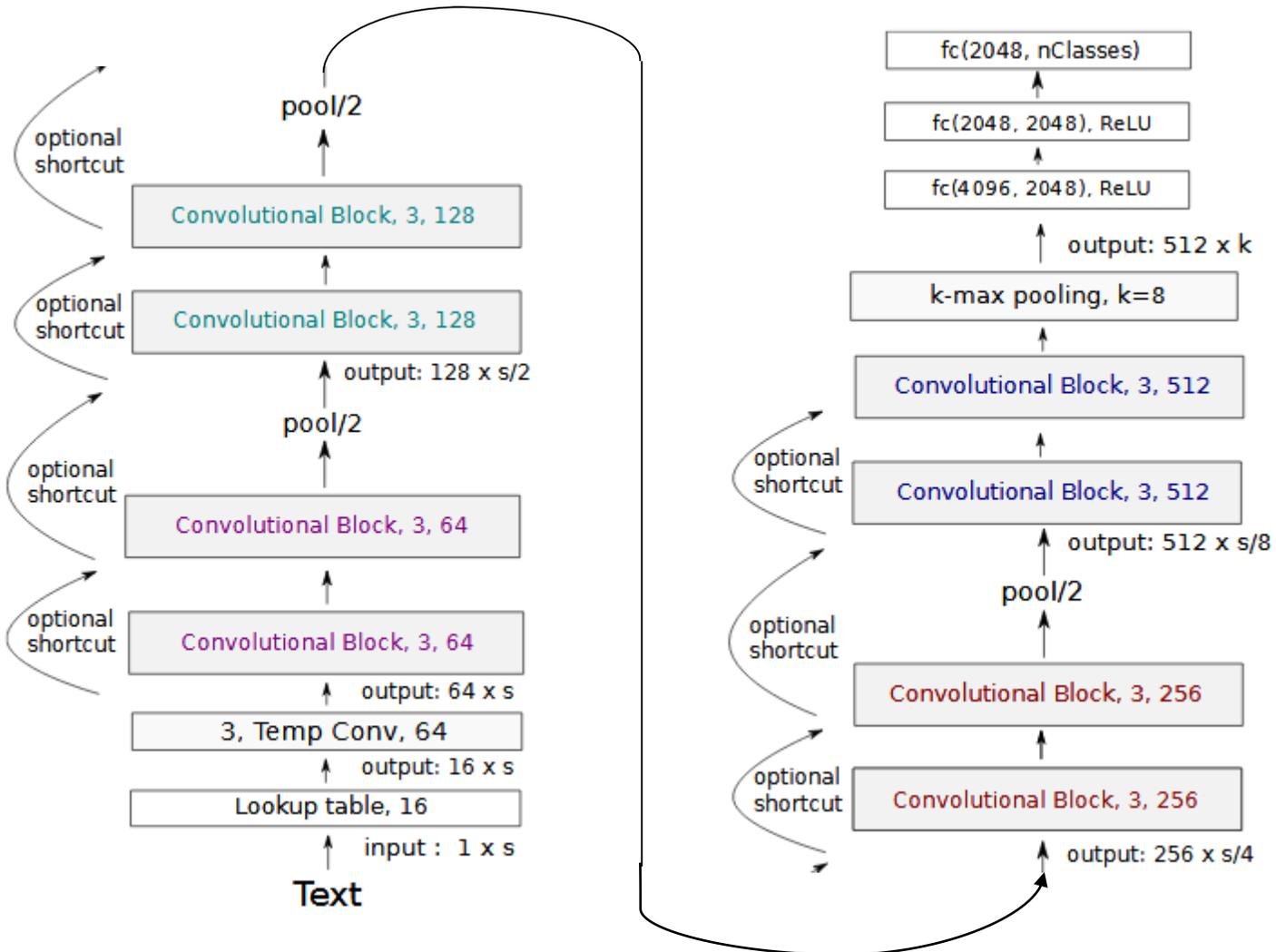
He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

# ...führt auch im NLP-Feld zu Erfolgen!



Conneau, Alexis, et al. "Very deep convolutional networks for text classification." *arXiv preprint arXiv:1606.01781* (2016).

# ...führt auch im NLP-Feld zu Erfolgen!

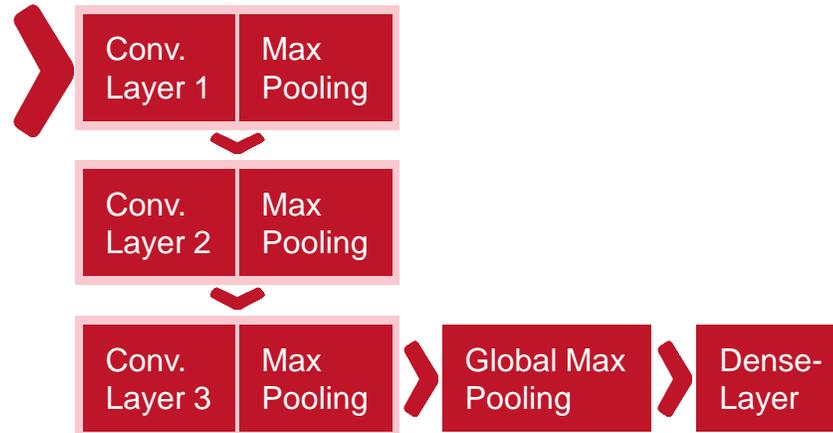
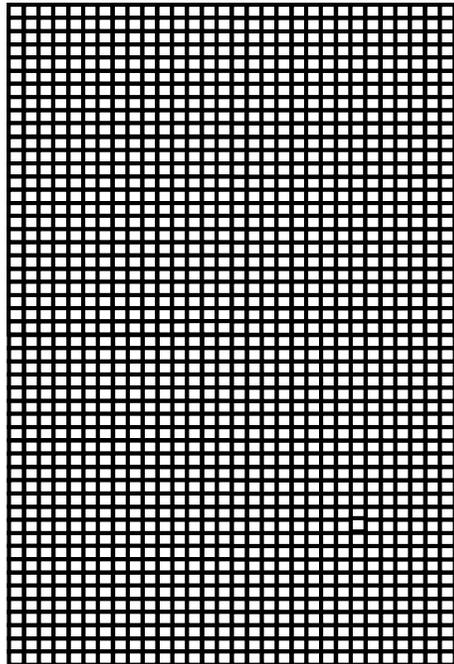


CNN mit 3 Convolutional Layer	
POC	➤ 96,74% Accuracy
MVP	➤ 94,39% Accuracy

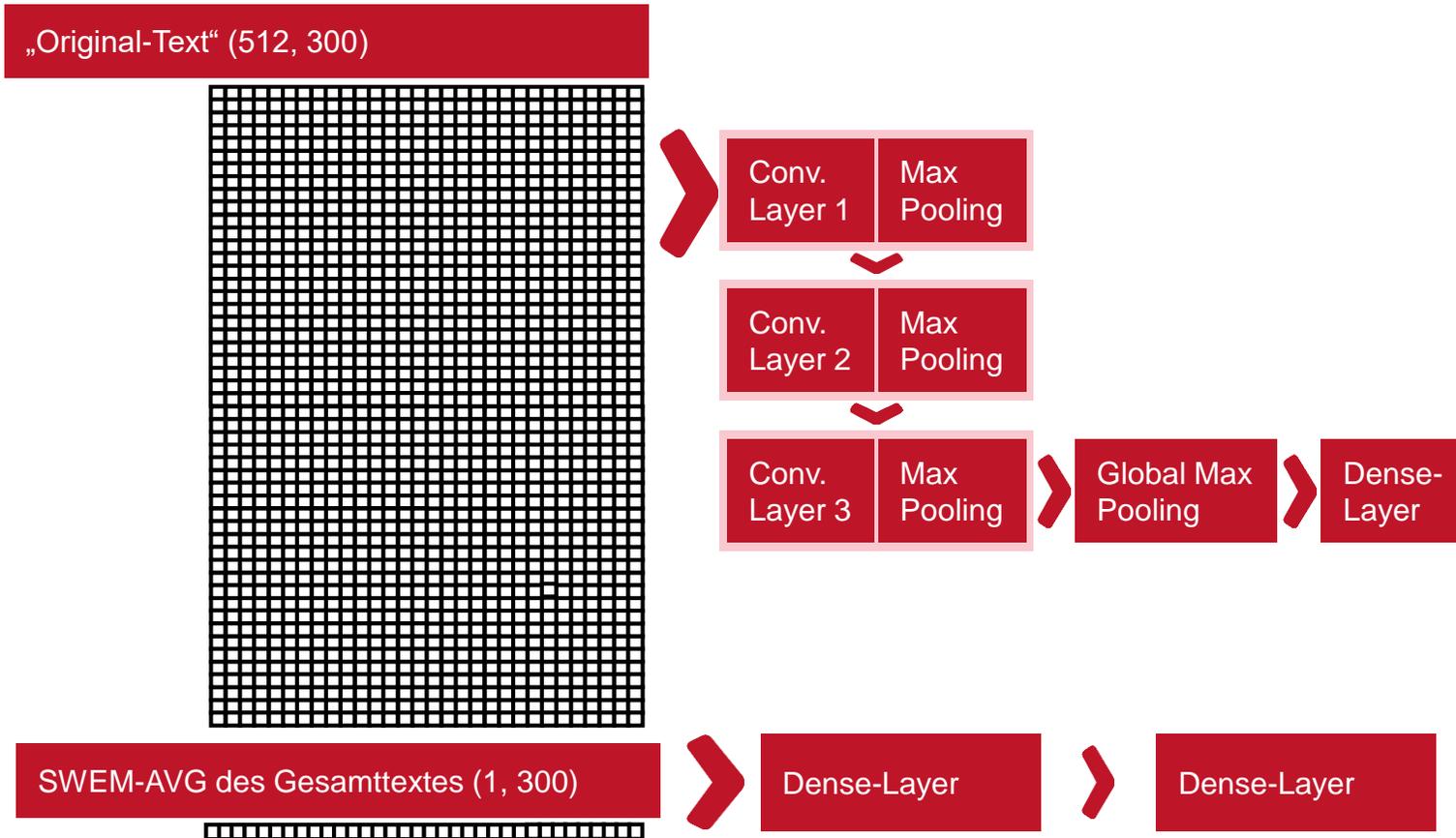
Conneau, Alexis, et al. "Very deep convolutional networks for text classification." *arXiv preprint arXiv:1606.01781* (2016).

# CNN-SWEM-Hybrid

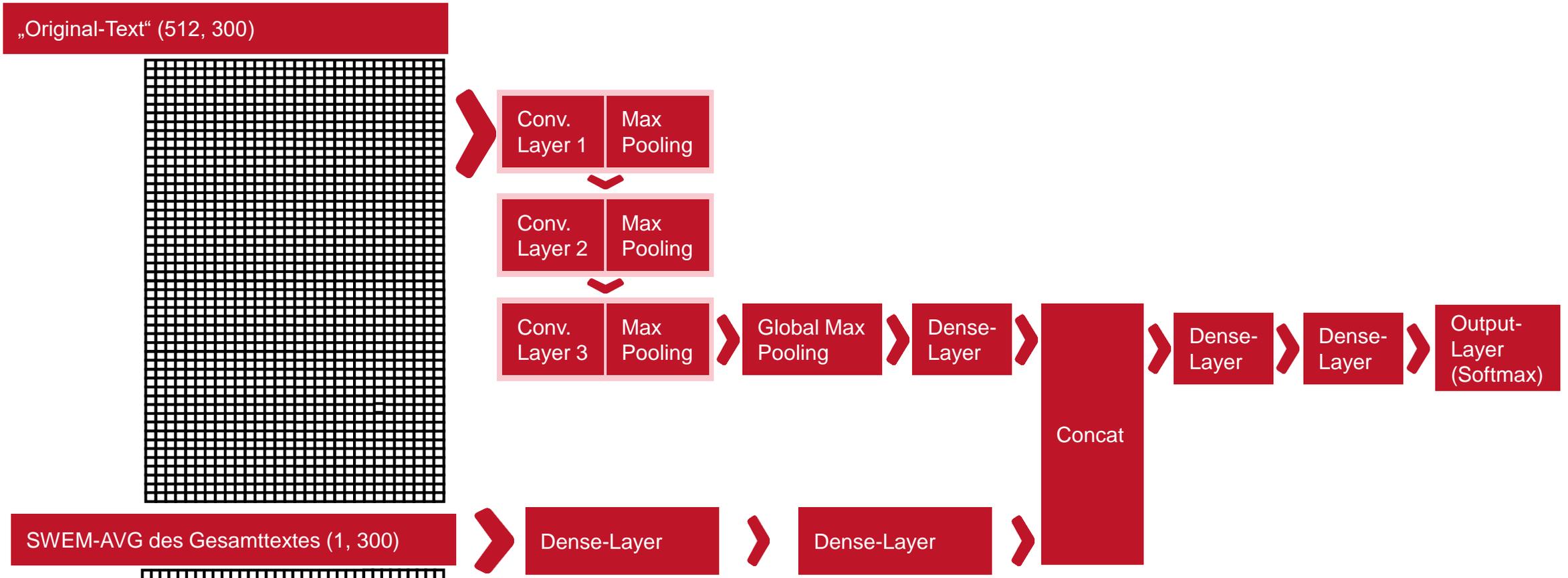
„Original-Text“ (512, 300)



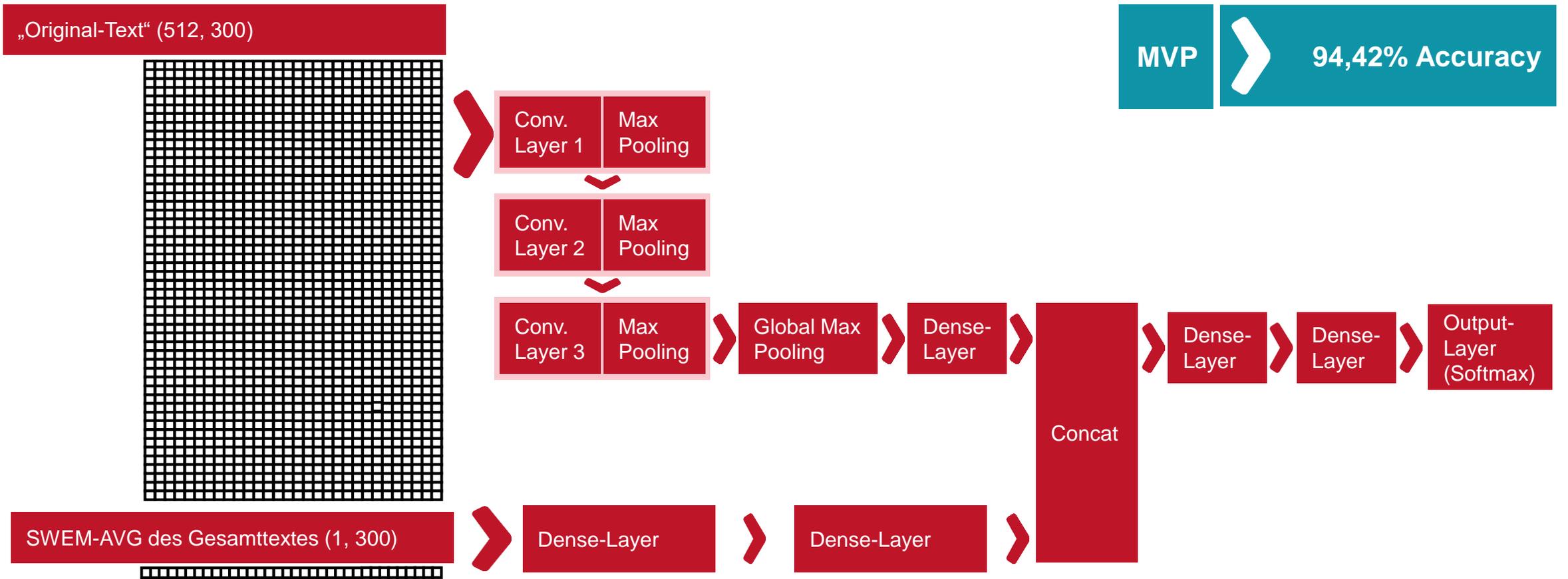
# CNN-SWEM-Hybrid



# CNN-SWEM-Hybrid



# CNN-SWEM-Hybrid



# Übersicht Modellperformance

Modell berücksichtigt...	Anzahl Dokumente	Accuracy auf Test-Daten
	43.214	
...Reihenfolge der Tokens NICHT	Bag of Words (1-gram)	93,71%
	SWEM AVG	93,14%
..."lokale" Reihenfolge der Tokens ("short range dependencies")	CNN (shallow, 1conv-layer)	94,12%
	CNN ("deep", 3conv-layers)	94,32%
..."globale" Reihenfolge der Tokens ("long range dependencies")	BiLSTM-Attention	94,46%
	CNN-SWEM-Hybrid	94.42%

# Übersicht Modellperformance

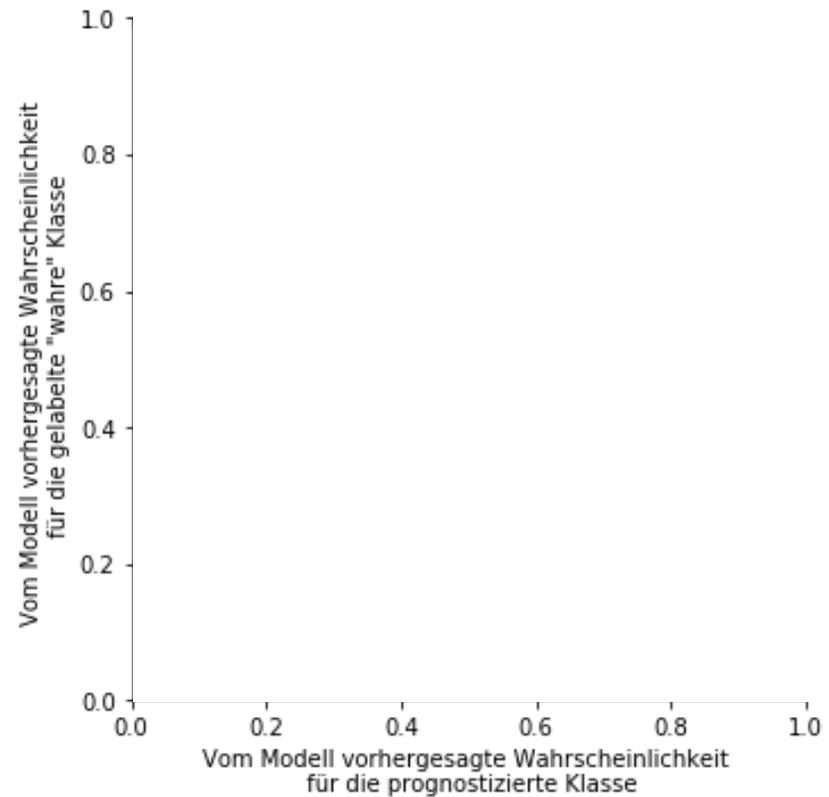
Modell berücksichtigt...	Anzahl Dokumente	Accuracy auf Test-Daten	Trainingszeit pro Epoche
	43.214		343.234
...Reihenfolge der Tokens NICHT	Bag of Words (1-gram)	93,71%	143s
	SWEM AVG	93,14%	1s
..."lokale" Reihenfolge der Tokens ("short range dependencies")	CNN (shallow, 1conv-layer)	94,12%	35s
	CNN ("deep", 3conv-layers)	94,32%	17s
..."globale" Reihenfolge der Tokens ("long range dependencies")	BiLSTM-Attention	94,46%	205s
	CNN-SWEM-Hybrid	94.42%	11s

Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

# Modellperformance & Fehleranalyse (2018 & 2019, Test-Sample)

**ERGO**

A Munich Re company

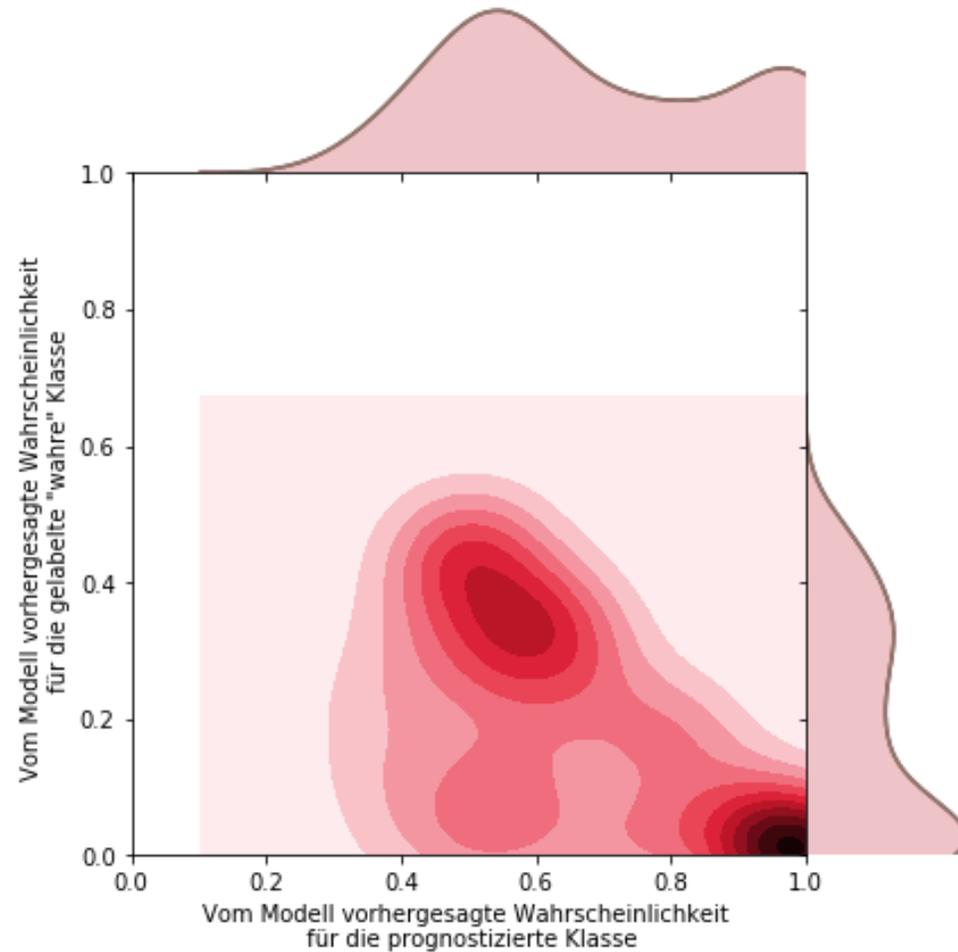


Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

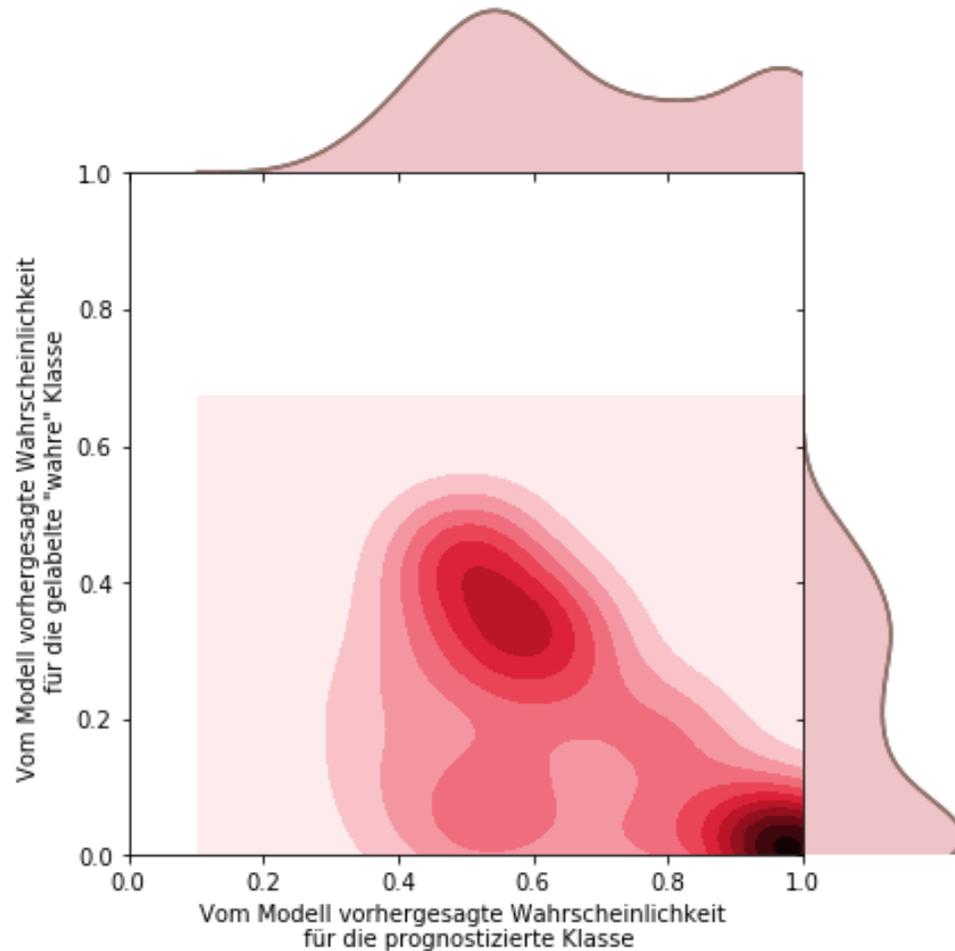
# Modellperformance & Fehleranalyse (2018 & 2019, Test-Sample)

**ERGO**

A Munich Re company



# Modellperformance & Fehleranalyse (2018 & 2019, Test-Sample)



**„Falschklassifizierte“ Dokumente zurückzuführen auf:**

- 1. Falsch gelabelte Daten**
- 2. Handschriftlich verfasste Dokumente**
- 3. Nicht eindeutige Klassifizierbarkeit**
- 4. Modellfehler**

**Vielen Dank  
für Ihre Aufmerksamkeit!**

[sebastian.mikus@ergo.de](mailto:sebastian.mikus@ergo.de)

[lucas.hafner@ergo.de](mailto:lucas.hafner@ergo.de)

**ERGO**

A Munich Re company

