

ERGO

A Munich Re company

Anwendung von Deep Learning zur Klassifikation eingehender Dokumente

Dr. Sebastian Mikus, Dr. Lucas Hafner

DAV/DGVFM e-Jahrestagung 2020



Teil 1

Eine kurze Einführung in Natural Language Processing mit Neuronalen Netzen

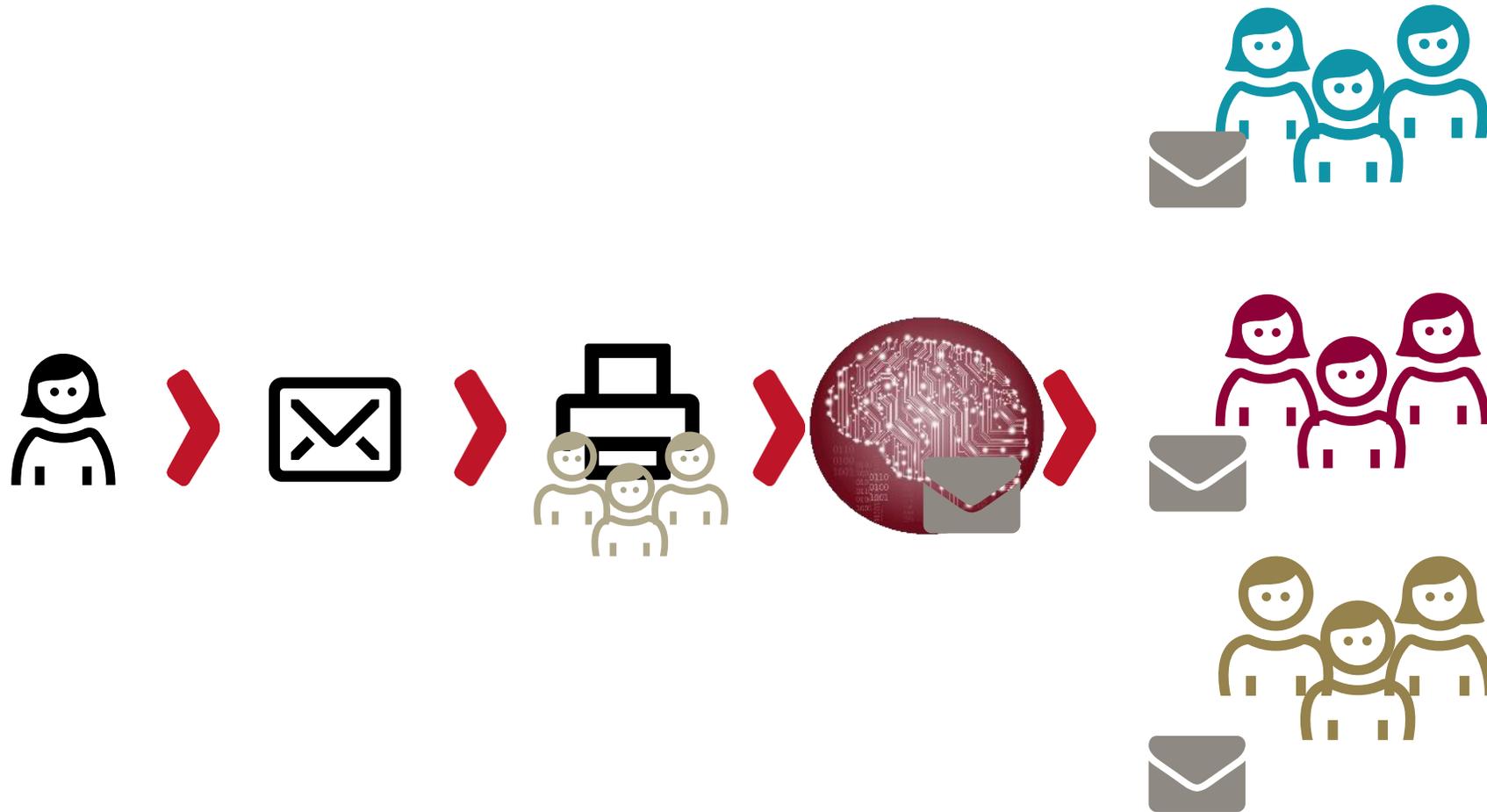
Teil 2

Ergebnisse der Textklassifikation bei einem konkreten Use Case

Signifikantes Potential für Verbesserungen im Eingangsmangement durch AI-Klassifizierung



Signifikantes Potential für Verbesserungen im Eingangsmangement durch AI-Klassifizierung



Review: Grundproblem des Machine Learning

ERGO

A Munich Re company

Lerne aus gegebene Daten einen Vorhersage zu machen

(höhe BAP, Alter, Beitrag, ...) → Storno(wahrscheinlichkeit)

(PZN, Alter VP, Beitrag, ...) → erwarteter Ablehnungsbetrag

(Tarif, Maßnahmenjahr, Abrechnungsjahr, ...) → erwarteter Schaden

Text aus Eingangspost → Anliegen

Text aus email → Bearbeitungort

Review: Grundproblem des Machine Learning

Lerne aus gegebene Daten einen Vorhersage zu machen

(höhe BAP, Alter, Beitrag, ...) → Storno(wahrscheinlichkeit)

(PZN, Alter VP, Beitrag, ...) → erwarteter Ablehnungsbetrag

(Tarif, Maßnahmenjahr, Abrechnungsjahr, ...) → erwarteter Schaden

Text aus Eingangspost → Anliegen

Text aus email → Bearbeitungort

Allgemeiner...

Gegeben:

$(x_1, y_1), \dots, (x_n, y_n)$

Ann: Es gibt einen Zusammenhang

$x \rightarrow f(x) + \epsilon$

Ziel: Lerne

f

Review: Grundproblem des Machine Learning

Lerne aus gegebene Daten einen Vorhersage zu machen

(höhe BAP, Alter, Beitrag, ...) → Storno(wahrscheinlichkeit)

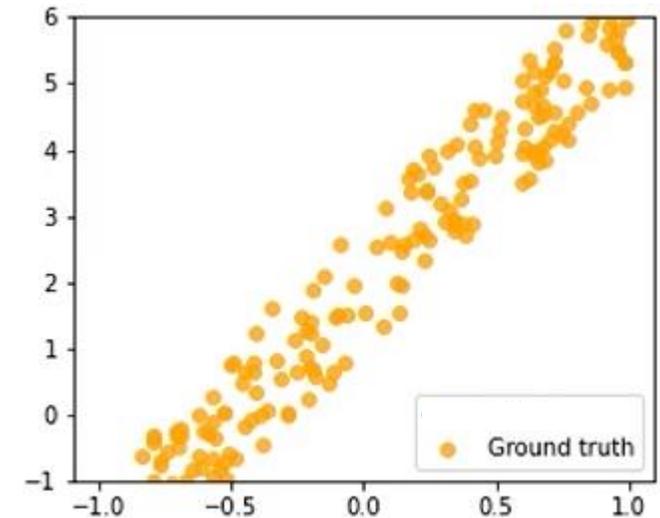
(PZN, Alter VP, Beitrag, ...) → erwarteter Ablehnungsbetrag

(Tarif, Maßnahmenjahr, Abrechnungsjahr, ...) → erwarteter Schaden

Text aus Eingangspost → Anliegen

Text aus email → Bearbeitungort

Beispiel



Allgemeiner...

Gegeben:

$(x_1, y_1), \dots, (x_n, y_n)$

Ann: Es gibt einen Zusammenhang

$x \rightarrow f(x) + \epsilon$

Ziel: Lerne

f

Review: Grundproblem des Machine Learning

Lerne aus gegebene Daten einen Vorhersage zu machen

(höhe BAP, Alter, Beitrag, ...) → Storno(wahrscheinlichkeit)

(PZN, Alter VP, Beitrag, ...) → erwarteter Ablehnungsbetrag

(Tarif, Maßnahmenjahr, Abrechnungsjahr, ...) → erwarteter Schaden

Text aus Eingangspost → Anliegen

Text aus email → Bearbeitungort

Allgemeiner...

Gegeben:

$(x_1, y_1), \dots, (x_n, y_n)$

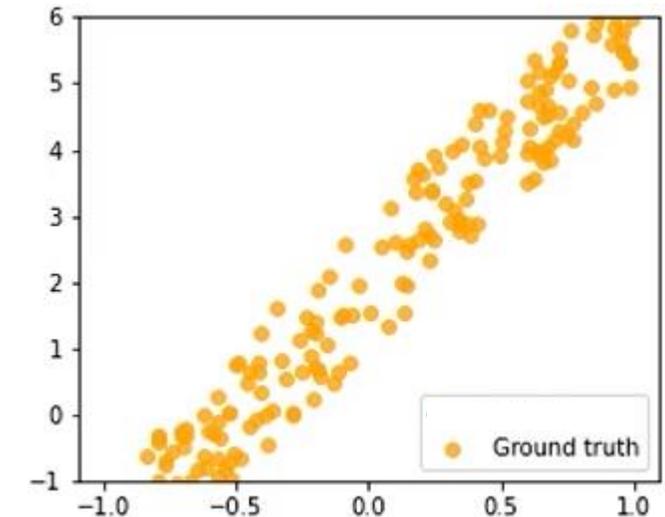
Ann: Es gibt einen Zusammenhang

$x \rightarrow f(x) + \epsilon$

Ziel: Lerne

f

Beispiel



$$Y = \beta_0 + \beta_1 * X + \epsilon$$

Review: Grundproblem des Machine Learning

Lerne aus gegebene Daten einen Vorhersage zu machen

(höhe BAP, Alter, Beitrag, ...) → Storno(wahrscheinlichkeit)

(PZN, Alter VP, Beitrag, ...) → erwarteter Ablehnungsbetrag

(Tarif, Maßnahmenjahr, Abrechnungsjahr, ...) → erwarteter Schaden

Text aus Eingangspost → Anliegen

Text aus email → Bearbeitungort

Allgemeiner...

Gegeben:

$(x_1, y_1), \dots, (x_n, y_n)$

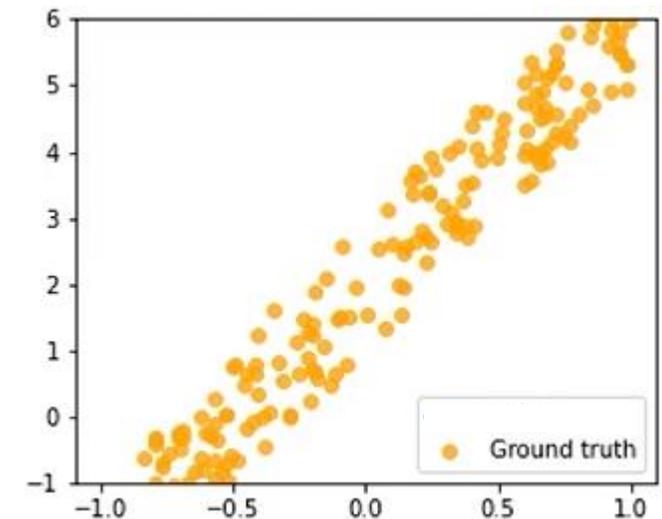
Ann: Es gibt einen Zusammenhang

$x \rightarrow f(x) + \epsilon$

Ziel: Lerne

f

Beispiel



$$Y = \beta_0 + \beta_1 * X + \epsilon$$

$$Loss(\beta_0, \beta_1) = \frac{1}{2n} \sum |y_i - \hat{y}_i|^2$$

Review: Grundproblem des Machine Learning

Lerne aus gegebene Daten einen Vorhersage zu machen

(höhe BAP, Alter, Beitrag, ...) → Storno(wahrscheinlichkeit)

(PZN, Alter VP, Beitrag, ...) → erwarteter Ablehnungsbetrag

(Tarif, Maßnahmenjahr, Abrechnungsjahr, ...) → erwarteter Schaden

Text aus Eingangspost → Anliegen

Text aus email → Bearbeitungort

Allgemeiner...

Gegeben:

$(x_1, y_1), \dots, (x_n, y_n)$

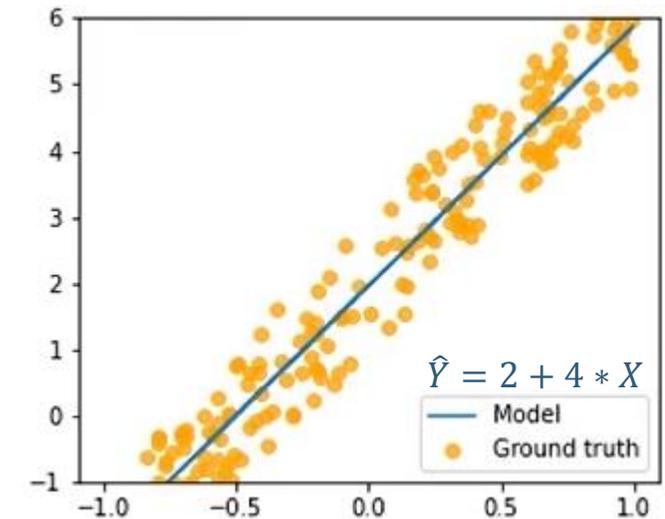
Ann: Es gibt einen Zusammenhang

$x \rightarrow f(x) + \epsilon$

Ziel: Lerne

f

Beispiel



$$Y = \beta_0 + \beta_1 * X + \epsilon$$

$$Loss(\beta_0, \beta_1) = \frac{1}{2n} \sum |y_i - \hat{y}_i|^2$$

Minimierung der Loss Funktion liefert Schätzer für f

Neuronale Netze: Eine kurze Einführung

ERGO

A Munich Re company

Neuronale Netze: Eine kurze Einführung

ERGO

A Munich Re company

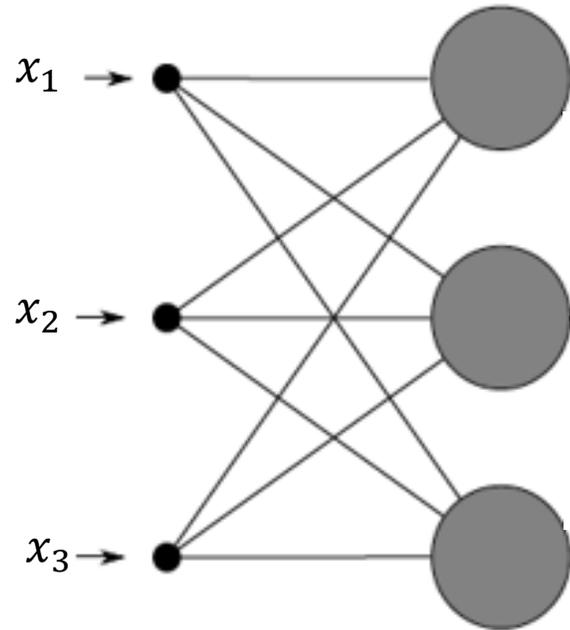
x_1

x_2

x_3

Input Layer

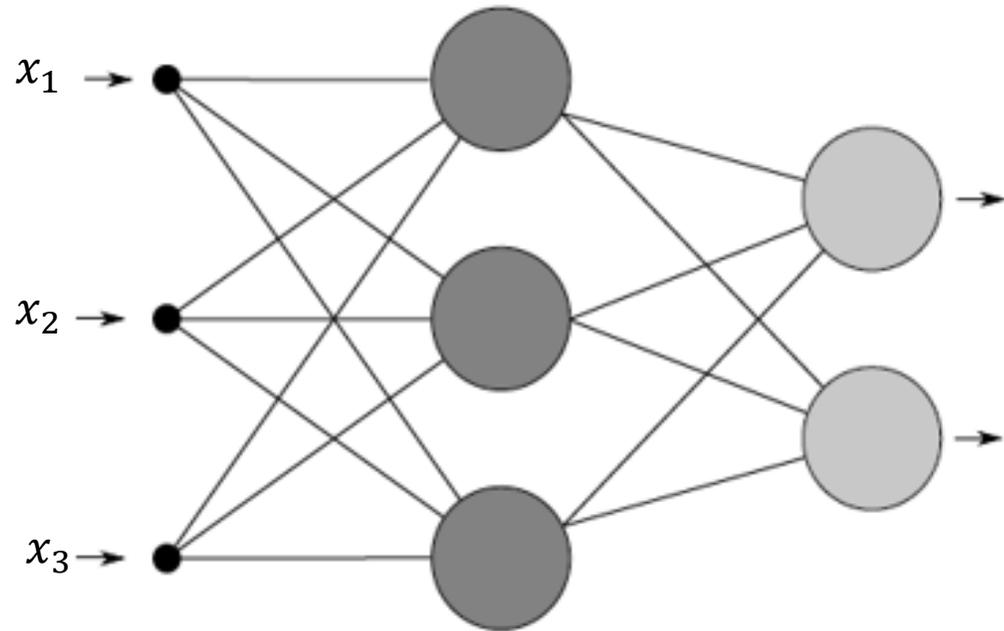
Neuronale Netze: Eine kurze Einführung



Input Layer

Hidden Layer

Neuronale Netze: Eine kurze Einführung

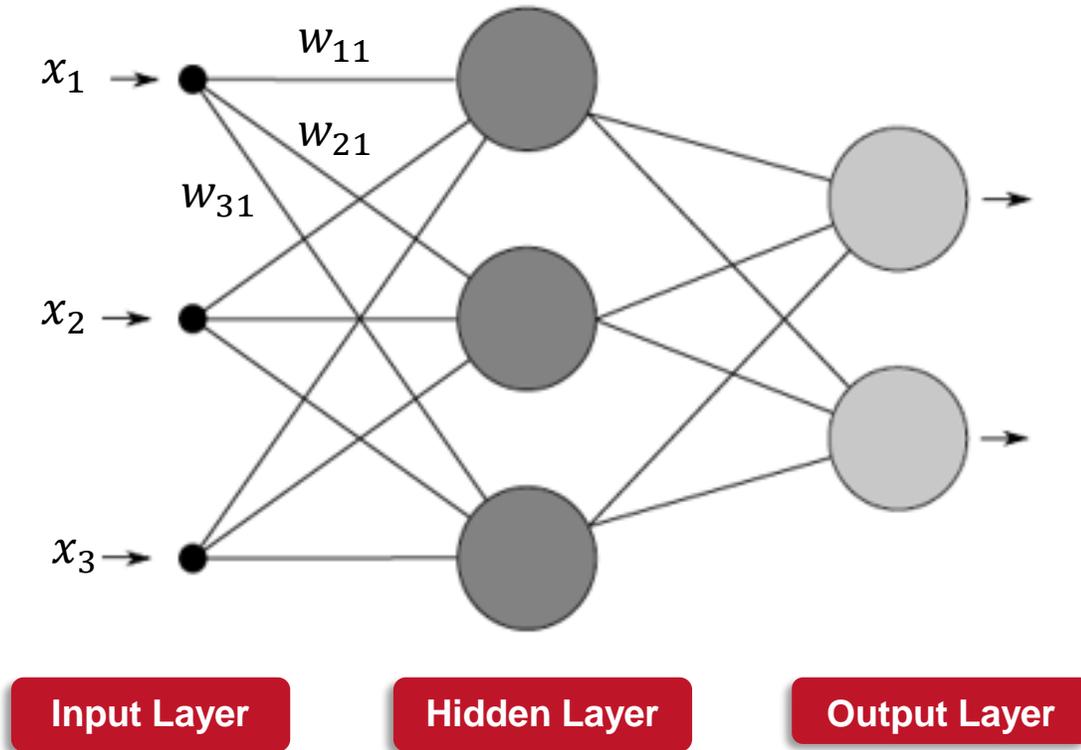


Input Layer

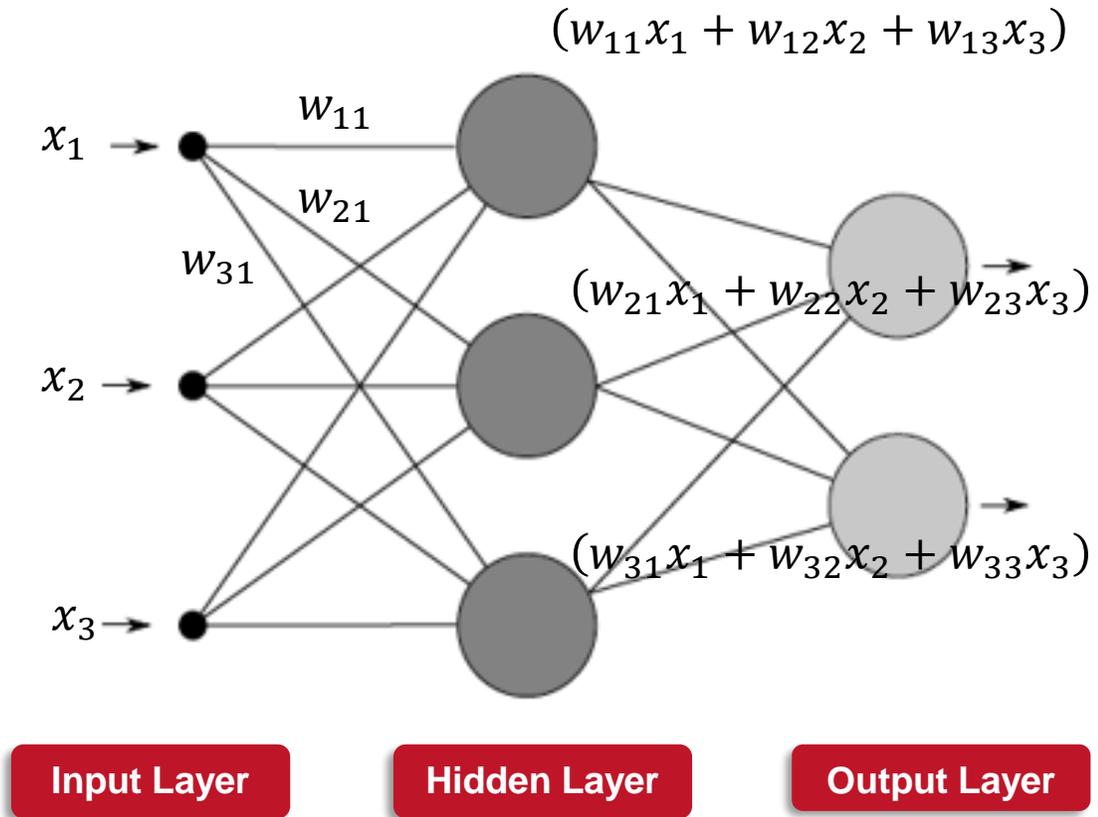
Hidden Layer

Output Layer

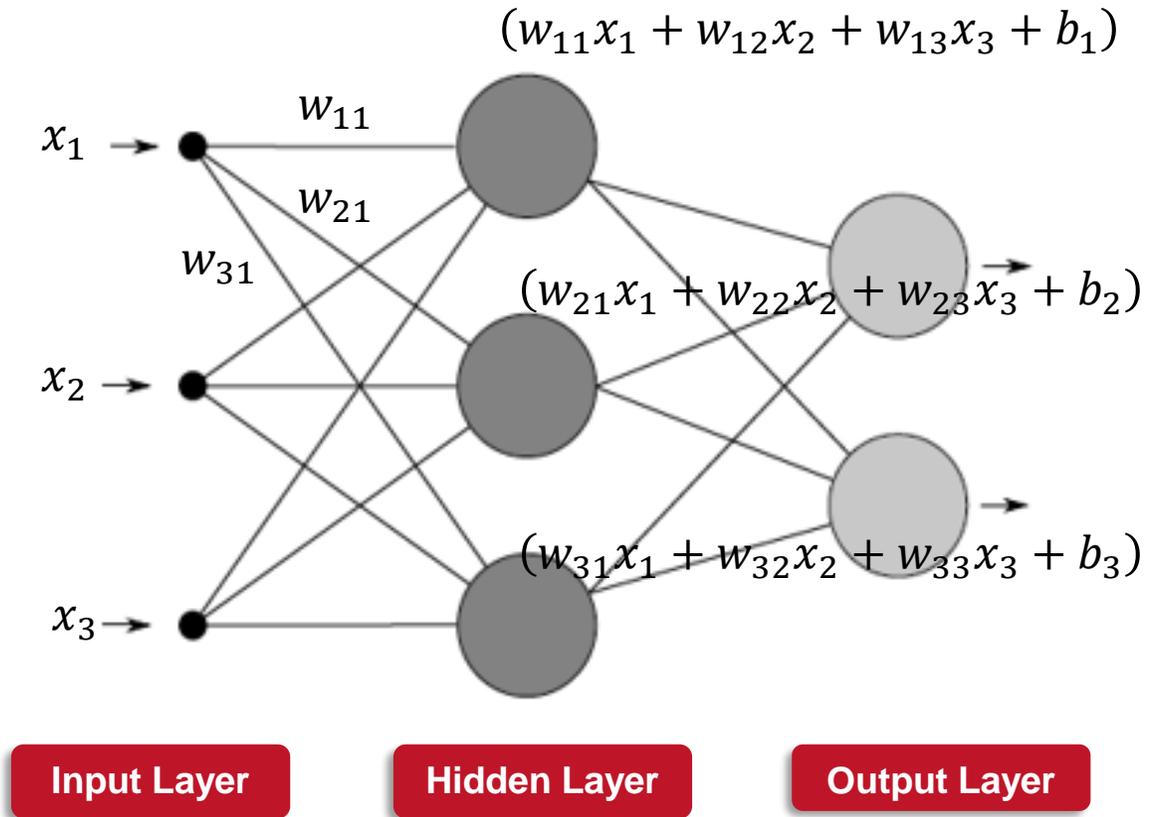
Neuronale Netze: Eine kurze Einführung



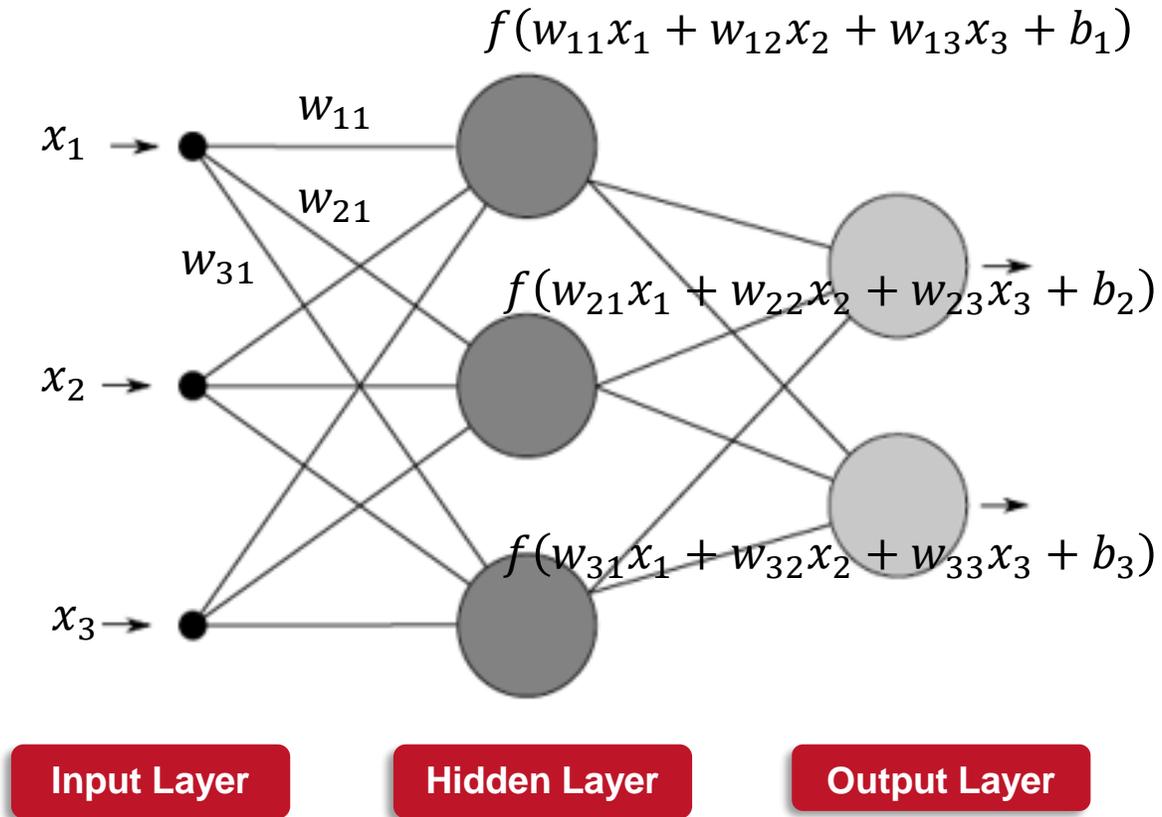
Neuronale Netze: Eine kurze Einführung



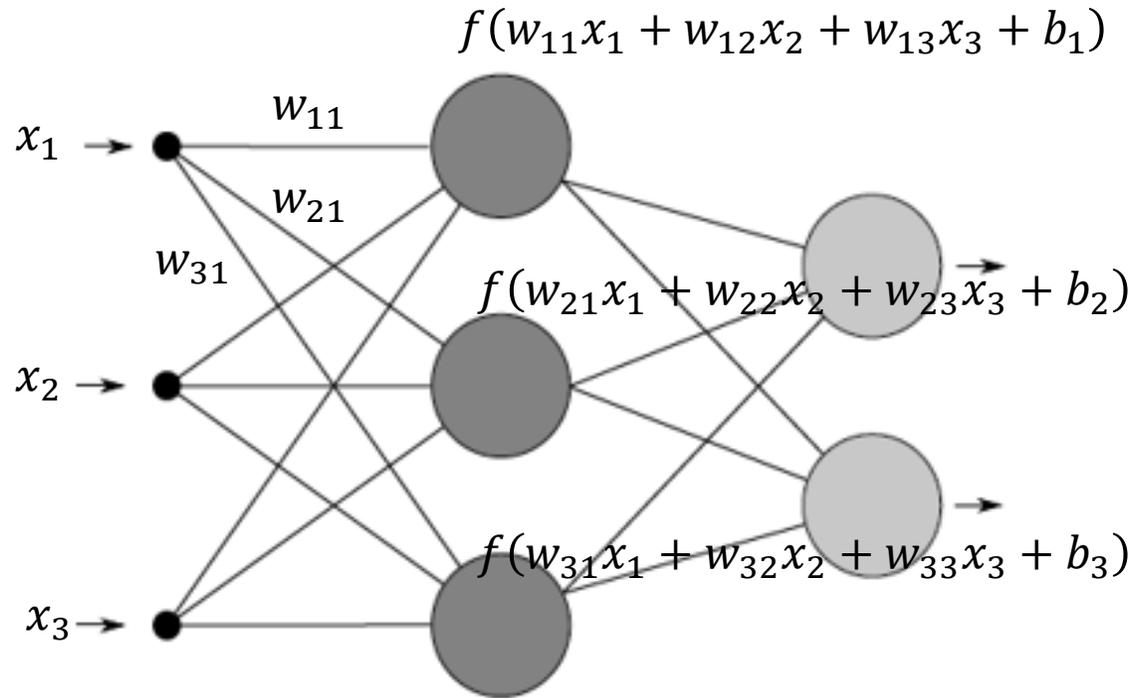
Neuronale Netze: Eine kurze Einführung



Neuronale Netze: Eine kurze Einführung



Neuronale Netze: Eine kurze Einführung



Input Layer

Hidden Layer

Output Layer

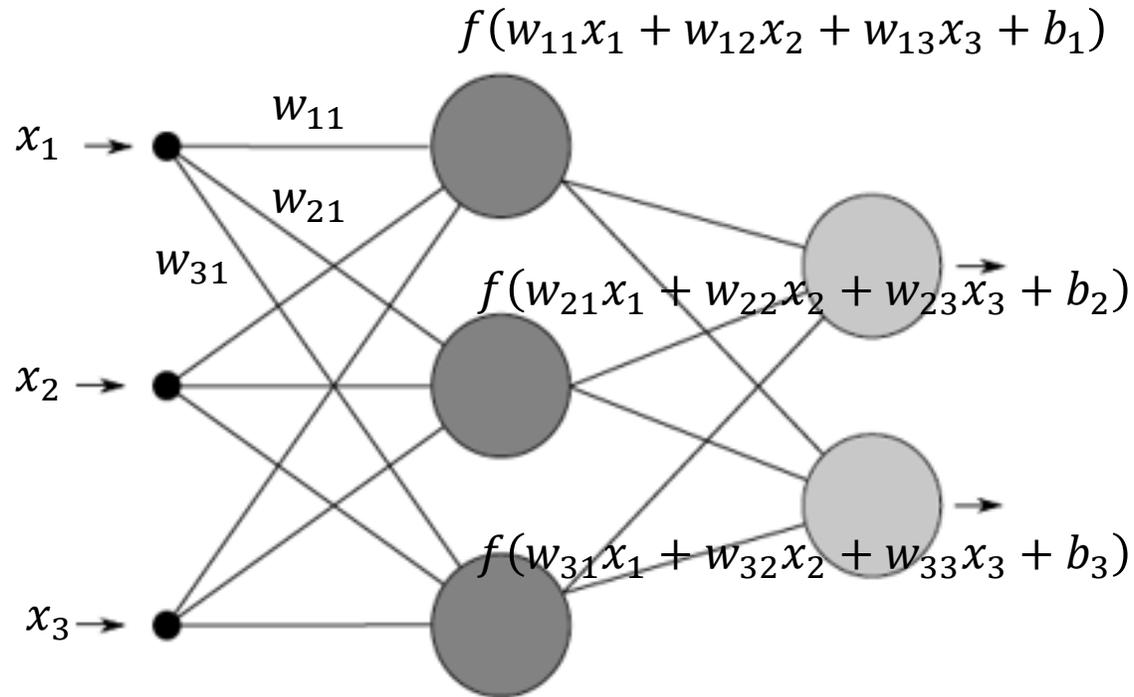
$$\text{relu}(x) = \max(0, x)$$

$$f(x) = \frac{1}{1 + e^x}$$

Beispiele für f

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum e^{x_j}}$$

Neuronale Netze: Eine kurze Einführung



$$L(w, b) = \frac{1}{2n} \sum \|y - a\|^2 \text{ min!}$$

Input Layer

Hidden Layer

Output Layer

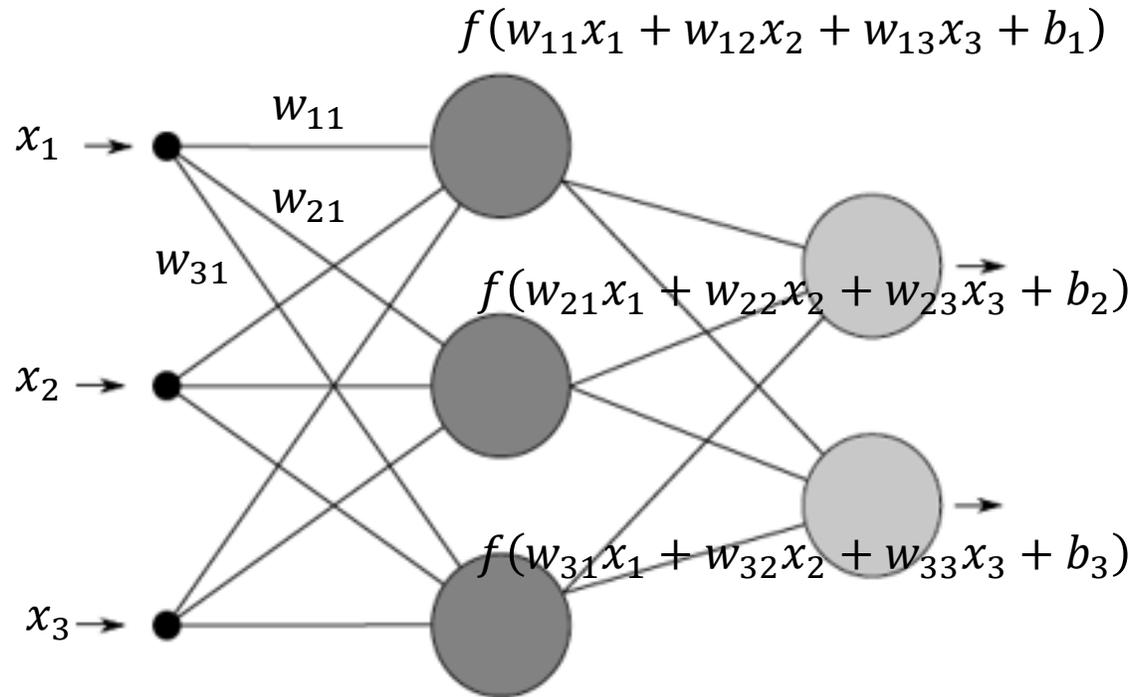
$$\text{relu}(x) = \max(0, x)$$

$$f(x) = \frac{1}{1 + e^x}$$

Beispiele für f

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum e^{x_j}}$$

Neuronale Netze: Eine kurze Einführung



$$L(w, b) = \frac{1}{2n} \sum \|y - a\|^2 \text{ min!}$$

Input Layer

Hidden Layer

Output Layer

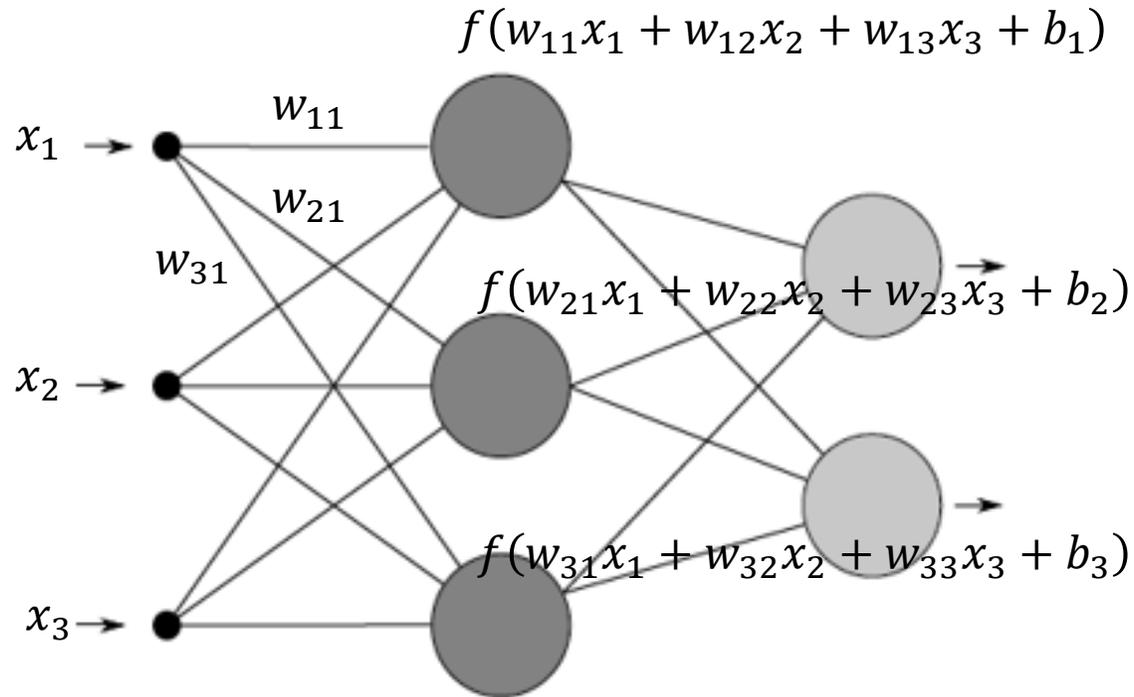
$$\text{relu}(x) = \max(0, x)$$

$$f(x) = \frac{1}{1 + e^x}$$

Beispiele für f

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum e^{x_j}}$$

Neuronale Netze: Eine kurze Einführung



$$L(w, b) = \frac{1}{2n} \sum \|y - a\|^2 \text{ min!}$$

Idee zur Lösung des Minimierungsproblems:

$\text{grad } L(w, b)$ zeigt in die Richtung des steilsten Anstiegs

D.h. ersetze sukzessive

$$w'_k = w_k - \mu \frac{\partial L}{\partial w_k}$$
$$b'_l = b_l - \mu \frac{\partial L}{\partial b_l}$$

Mini-Batch Stochastic Gradient Descent (SGD)

Input Layer

Hidden Layer

Output Layer

$$\text{relu}(x) = \max(0, x)$$

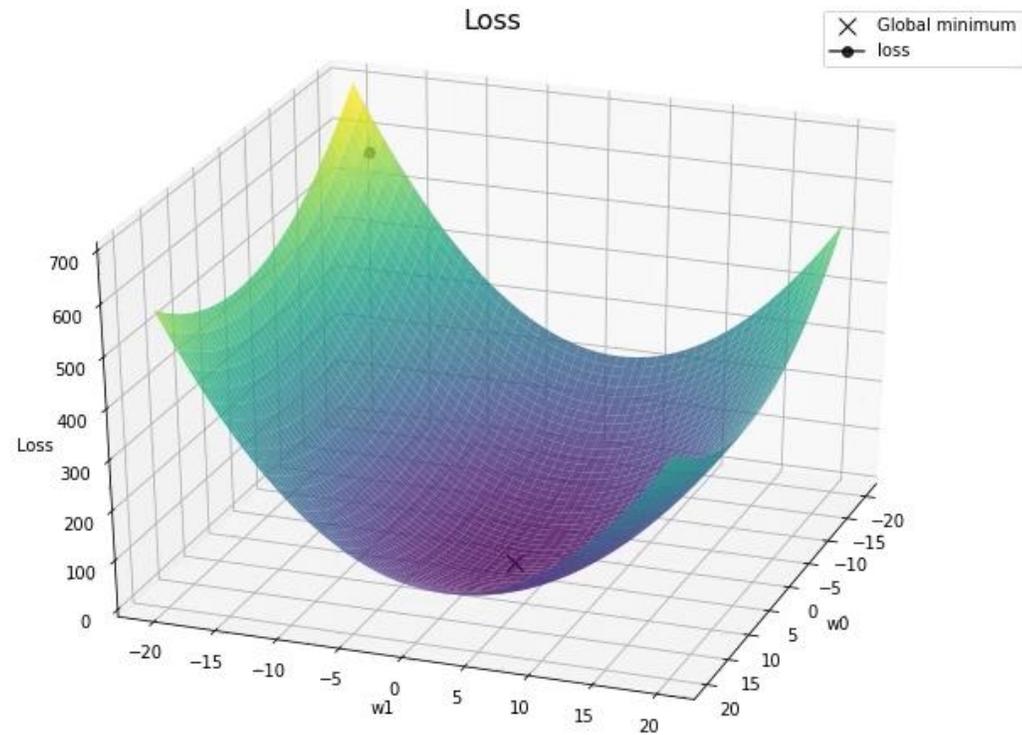
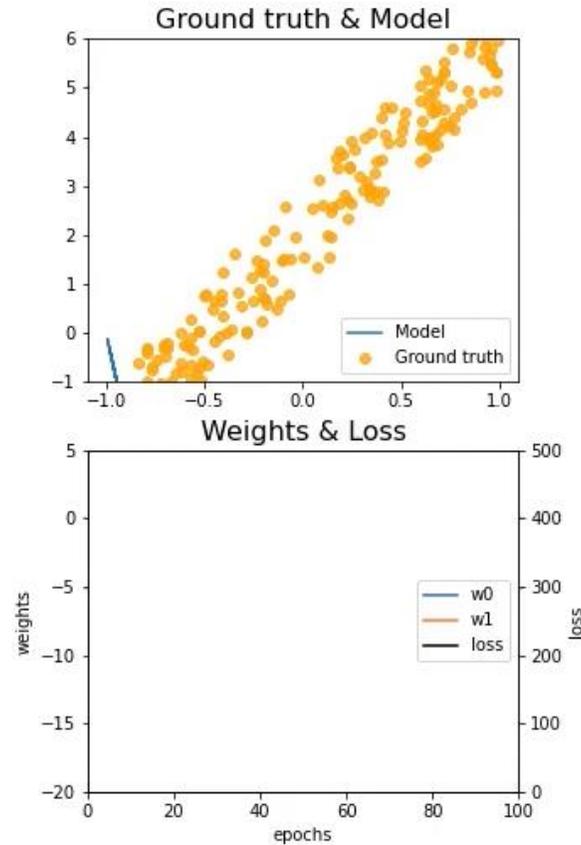
$$f(x) = \frac{1}{1 + e^x}$$

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum e^{x_j}}$$

Beispiele für f

Mini-Batch Stochastic Gradient Descent

Learning Rate: 0.01 - Epoch: 1/100



Vom Text zur Zahl: Modellierungsansätze, Bag of Words

ERGO

A Munich Re company

- 1) „Anbei sende ich Ihnen meine Rechnung zum Schaden...“
- 2) „Anbei sende ich Ihnen meine neue Kontoverbindung...“

Vom Text zur Zahl: Modellierungsansätze, Bag of Words

- 1) „Anbei sende ich Ihnen meine Rechnung zum Schaden...“
- 2) „Anbei sende ich Ihnen meine neue Kontoverbindung...“

a) Count 1_Gram

Dokument	Anbei	sende	ich	Ihnen	meine	Rechnung	zum	Schaden	neue	Kontoverbindung
1	1	1	1	1	1	1	1	1	0	0
2	1	1	1	1	1	0	0	0	1	1

Vom Text zur Zahl: Modellierungsansätze, Bag of Words

- 1) „Anbei sende ich Ihnen meine Rechnung zum Schaden...“
- 2) „Anbei sende ich Ihnen meine neue Kontoverbindung...“

a) Count 1_Gram

Dokument	Anbei	sende	ich	Ihnen	meine	Rechnung	zum	Schaden	neue	Kontoverbindung
1	1	1	1	1	1	1	1	1	0	0
2	1	1	1	1	1	0	0	0	1	1

b) Count 2_Gram

Dokument	(Anbei, Sende)	(sende, Ich)	(ich, Ihnen)	(Ihnen, meine)	(meine, Rechnung)	(Rechnung, zum)	(meine, neue)	(zum, Schaden)	(neue, Kontoverbindung)	
1	1	1	1	1	1	1	0	1	0	
2	1	1	1	1	0	0	1	0	1	

Vom Text zur Zahl: Word2Vec – die Grundidee

- Wörter in ähnlichen Kontexten sollten ähnliche Darstellungen zugeordnet werden
- Idee: Trainiere ein Netz, das angibt, wie wahrscheinlich ein Wort „in der Nähe“ eines anderen Wortes vorkommt.

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

Vom Text zur Zahl: Word2Vec – die Grundidee

- Wörter in ähnlichen Kontexten sollten ähnliche Darstellungen zugeordnet werden
- Idee: Trainiere ein Netz, das angibt, wie wahrscheinlich ein Wort „in der Nähe“ eines anderen Wortes vorkommt.

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

Vom Text zur Zahl: Word2Vec – die Grundidee

- Wörter in ähnlichen Kontexten sollten ähnliche Darstellungen zugeordnet werden
- Idee: Trainiere ein Netz, das angibt, wie wahrscheinlich ein Wort „in der Nähe“ eines anderen Wortes vorkommt.

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

Vom Text zur Zahl: Word2Vec – die Grundidee

- Wörter in ähnlichen Kontexten sollten ähnliche Darstellungen zugeordnet werden
- Idee: Trainiere ein Netz, das angibt, wie wahrscheinlich ein Wort „in der Nähe“ eines anderen Wortes vorkommt.

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

Trainingsbeispiele:

- (ich, sende), (ich, Ihnen)

Vom Text zur Zahl: Word2Vec – die Grundidee

- Wörter in ähnlichen Kontexten sollten ähnliche Darstellungen zugeordnet werden
- Idee: Trainiere ein Netz, das angibt, wie wahrscheinlich ein Wort „in der Nähe“ eines anderen Wortes vorkommt.

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

Trainingsbeispiele:

- (ich, sende), (ich, Ihnen)

Vom Text zur Zahl: Word2Vec – die Grundidee

- Wörter in ähnlichen Kontexten sollten ähnliche Darstellungen zugeordnet werden
- Idee: Trainiere ein Netz, das angibt, wie wahrscheinlich ein Wort „in der Nähe“ eines anderen Wortes vorkommt.

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

Trainingsbeispiele:

- (ich, sende), (ich, Ihnen)
- (Ihnen,ich), (Ihnen, meine)

Vom Text zur Zahl: Word2Vec – die Grundidee

- Wörter in ähnlichen Kontexten sollten ähnliche Darstellungen zugeordnet werden
- Idee: Trainiere ein Netz, das angibt, wie wahrscheinlich ein Wort „in der Nähe“ eines anderen Wortes vorkommt.

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

Trainingsbeispiele:

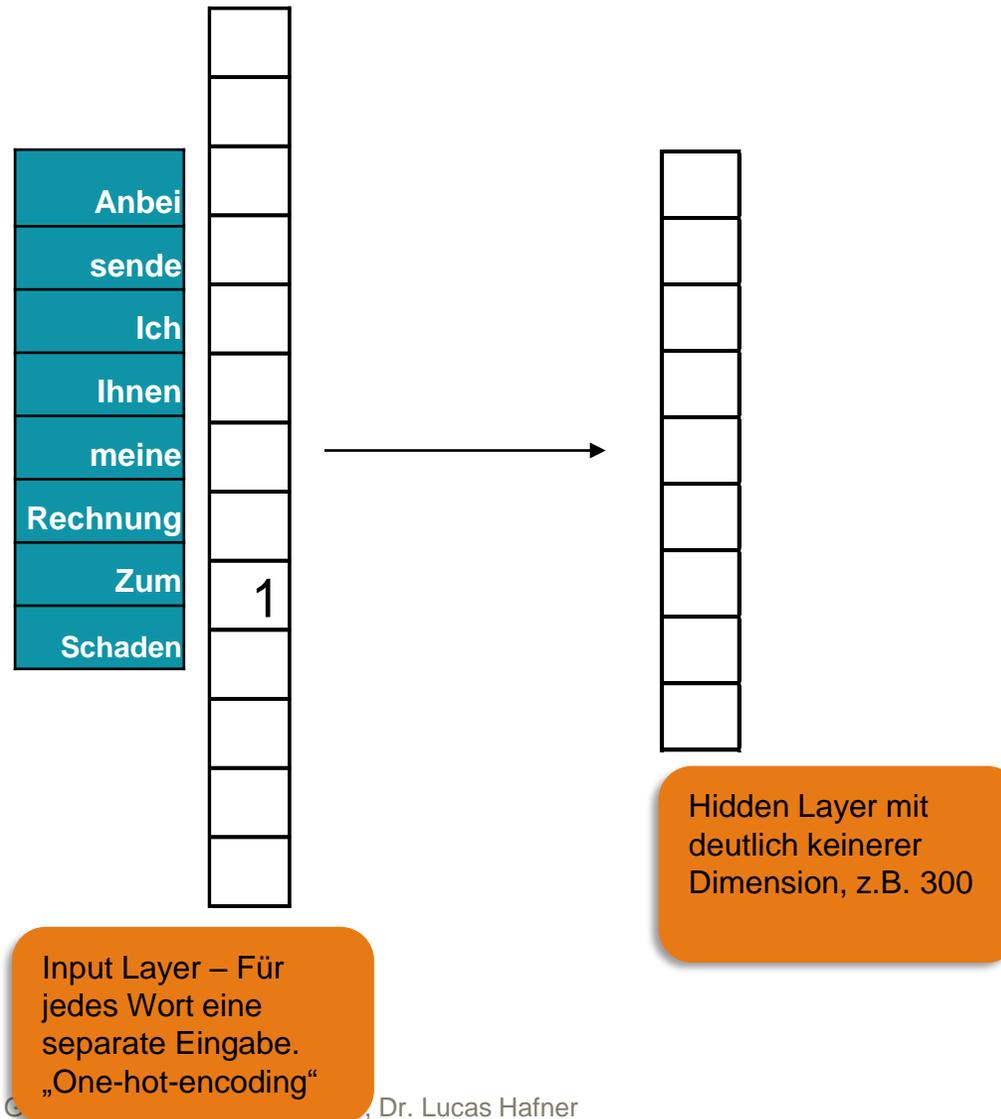
- (ich, sende), (ich, Ihnen)
- (Ihnen,ich), (Ihnen, meine)
- (meine, Ihnen), (meine, Rechnung)
-

Vom Text zur Zahl: Word2Vec – die Grundidee

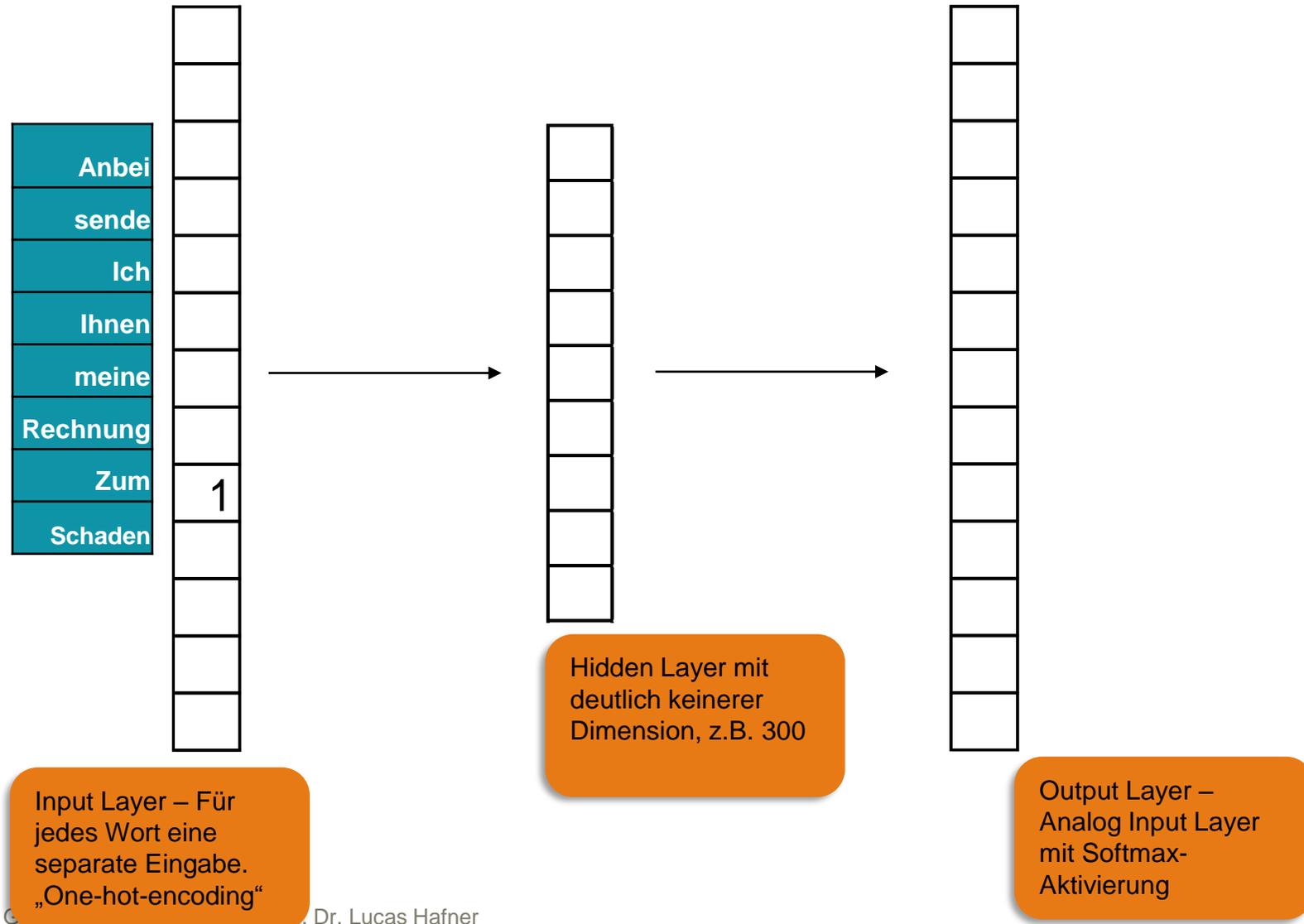


Input Layer – Für jedes Wort eine separate Eingabe. „One-hot-encoding“

Vom Text zur Zahl: Word2Vec – die Grundidee



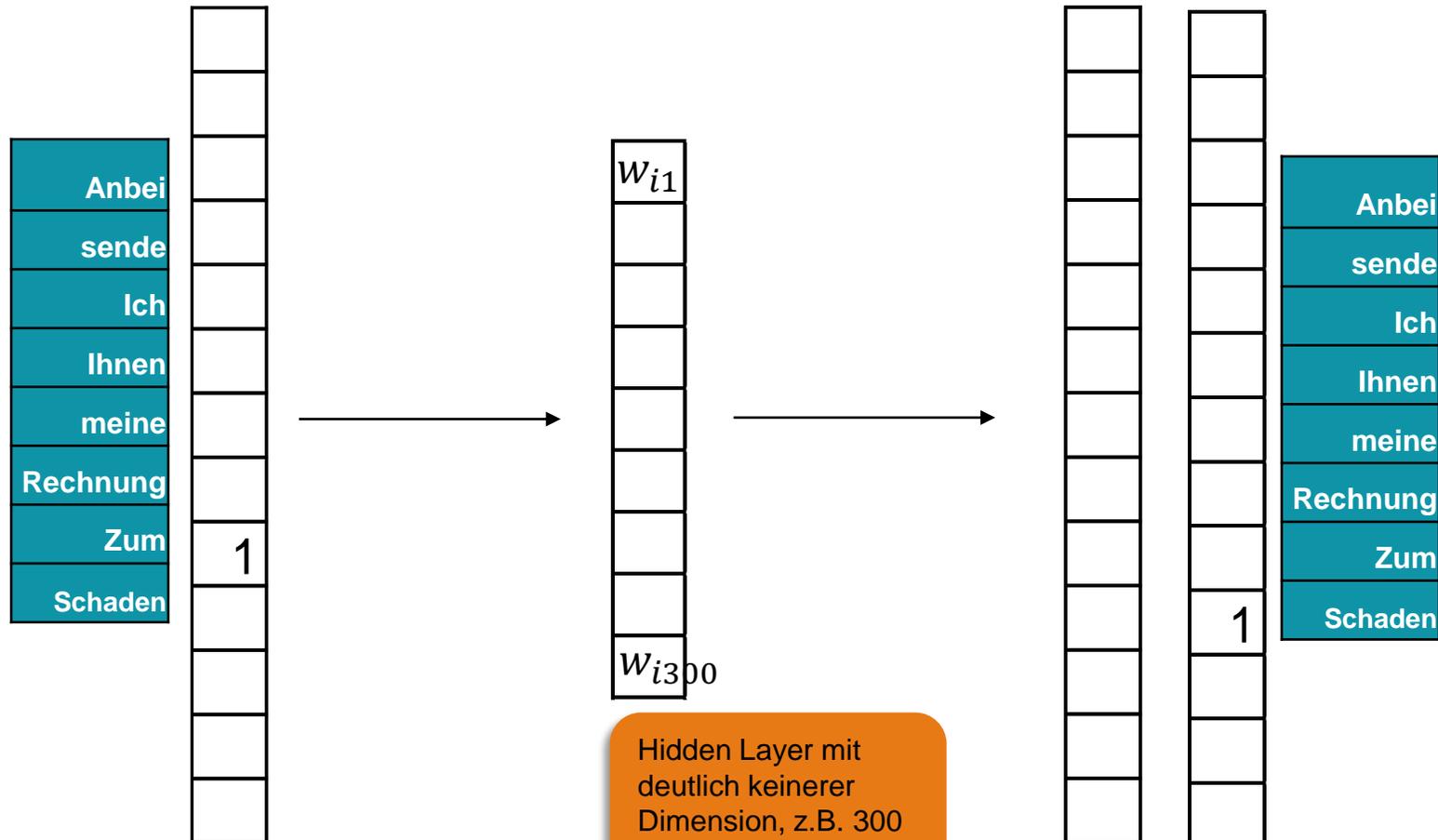
Vom Text zur Zahl: Word2Vec – die Grundidee



Vom Text zur Zahl: Word2Vec – die Grundidee

Trainings-
beispiel:
 $x_i = \text{„zum“}$
 $y_i = \text{„Schaden“}$

Durch die Softmax
Aktivierung lernt das
Netz die
Wahrscheinlichkeit
für dafür, dass y in
der Umgebung von x
vorkommt



Input Layer – Für
jedes Wort eine
separate Eingabe.
„One-hot-encoding“

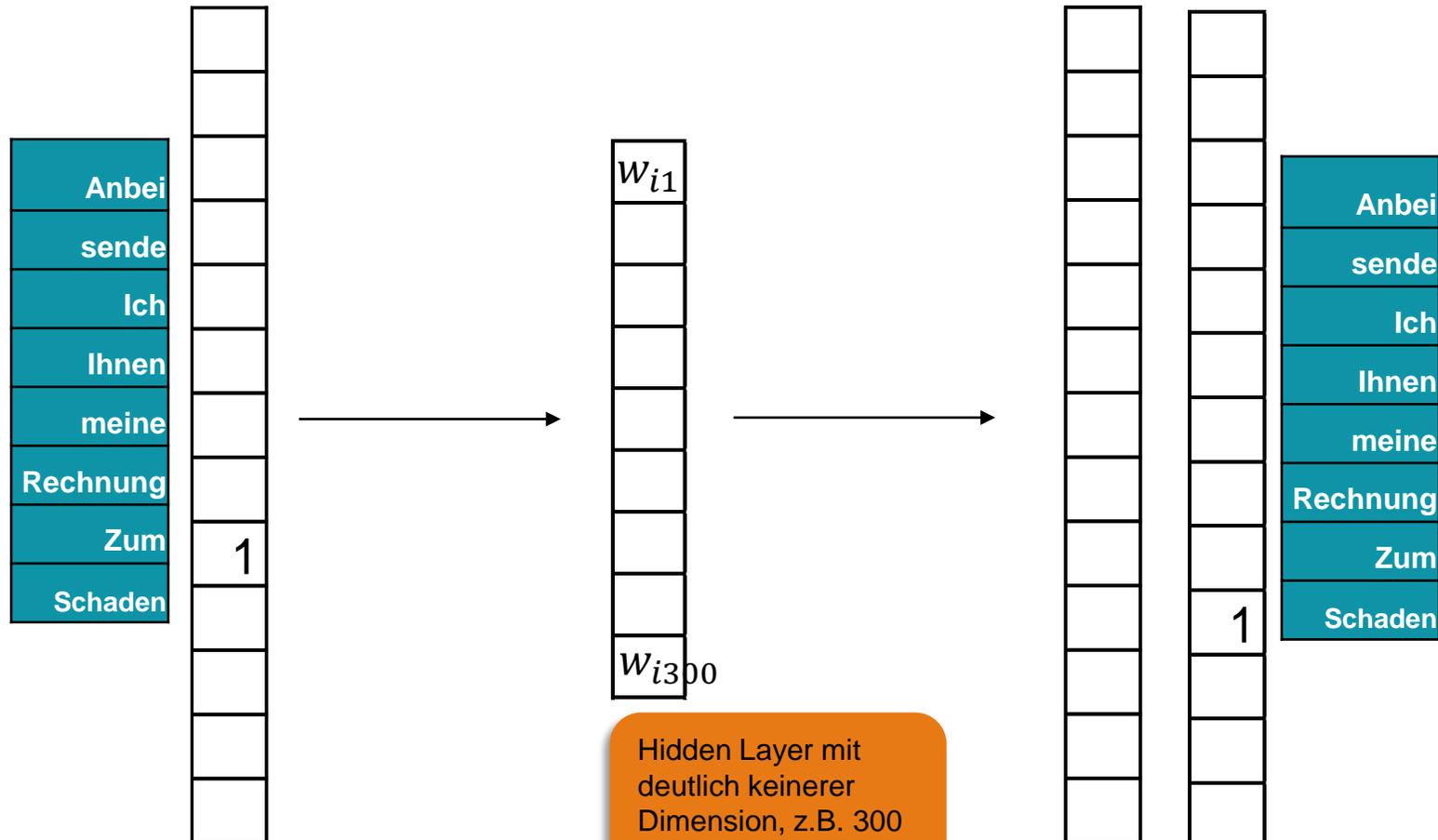
Hidden Layer mit
deutlich kleinerer
Dimension, z.B. 300

Output Layer –
Analog Input Layer
mit Softmax-
Aktivierung

Vom Text zur Zahl: Word2Vec – die Grundidee

Trainings-
beispiel:
 $x_i = \text{„zum“}$
 $y_i = \text{„Schaden“}$

Durch die Softmax
Aktivierung lernt das
Netz die
Wahrscheinlichkeit
für dafür, dass y in
der Umgebung von x
vorkommt

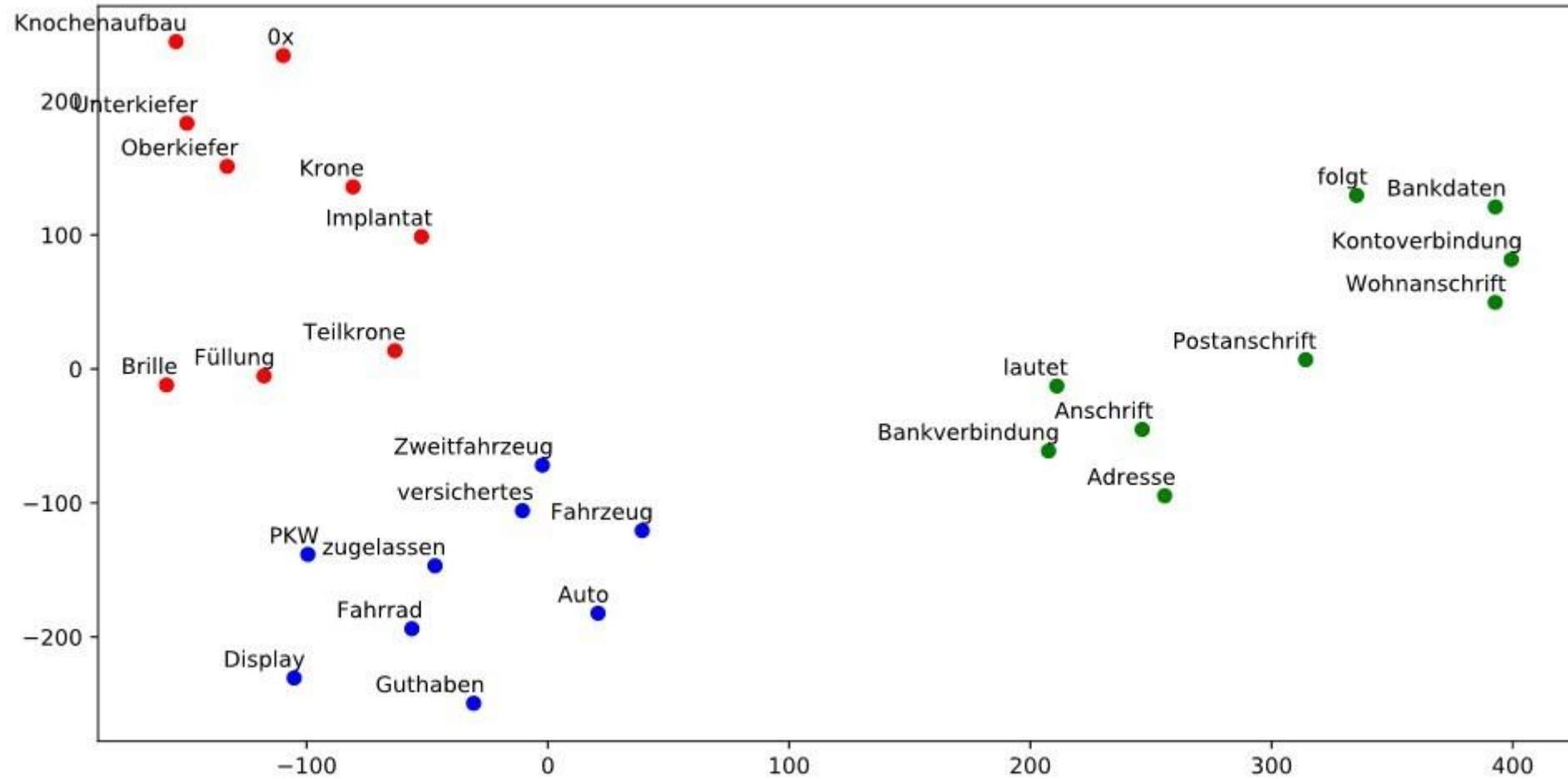


Input Layer – Für
jedes Wort eine
separate Eingabe.
„One-hot-encoding“

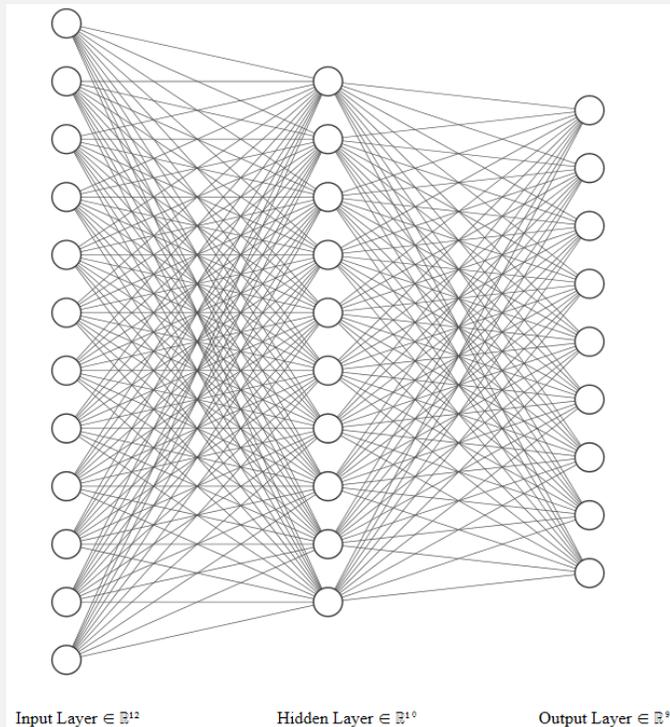
Hidden Layer mit
deutlich kleinerer
Dimension, z.B. 300

Output Layer –
Analog Input Layer
mit Softmax-
Aktivierung

Visualisierung des Word2Vec



Fully Connected Neural Networks



Quelle: <http://alexlenail.me/NN-SVG/index.html>

Convolutional Neural Networks

- LeCun, Yann, et al. "Gradient-based learning applied to document recognition." (1998)
- Wichtiger Baustein in der modernen Bilderkennung

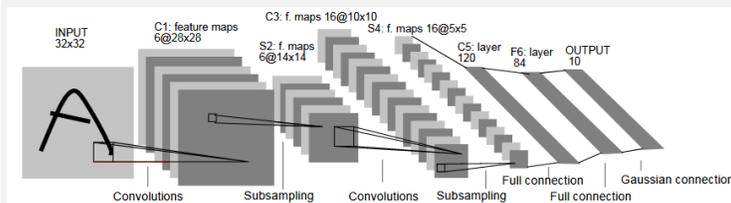
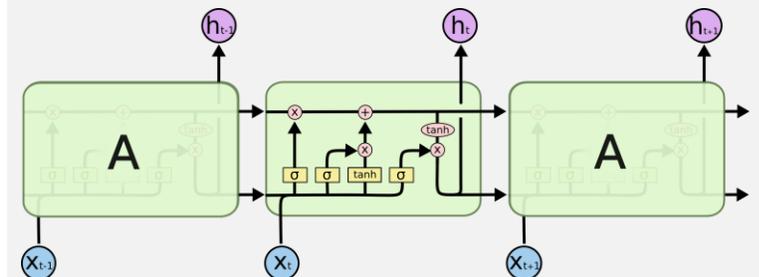


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Recurrent Neural Networks

- **LSTM (Long short-term memory)**
 - Sepp Hochreiter, Jürgen Schmidhuber: „Long short-term memory“ (1996)
 - Idee dahinter: Die Eingabe wird sequentiell verarbeitet.
 - Erlaubt Verarbeitung von langen **Sequenzen** (z.B. Texte, Audio)

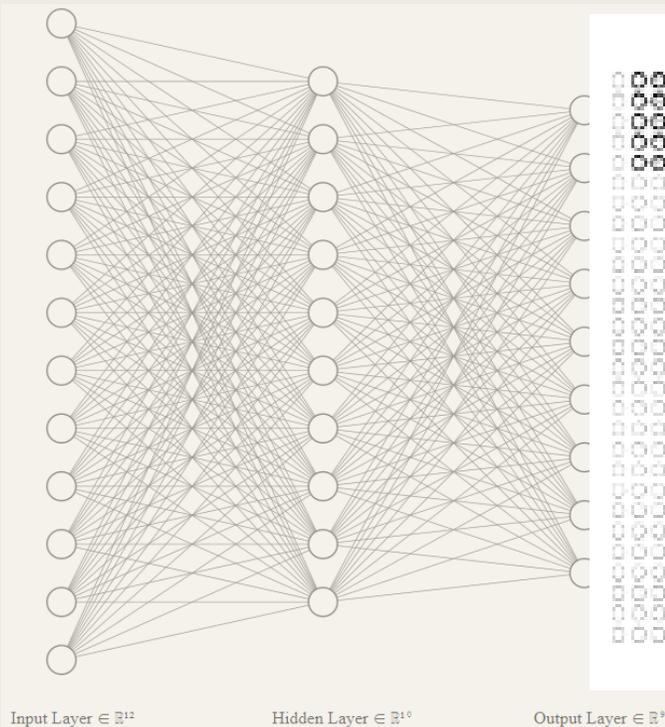
(Illustration LSTM Zelle):



Quelle: colah.github.io/posts/2015-08-Understanding-LSTMs/

Neural Network Architectures

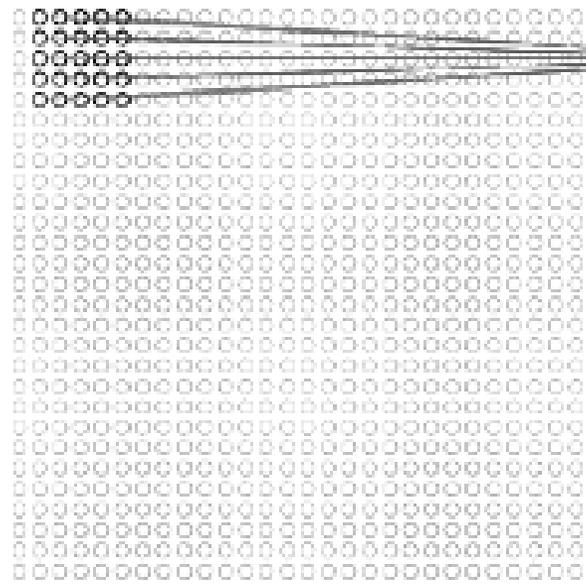
Fully Connected Neural Networks



Quelle: <http://alexenail.me/NN-SVG/index.html>

Convolutional Neural Networks

- LeCun, Yann, et al. "Gradient-

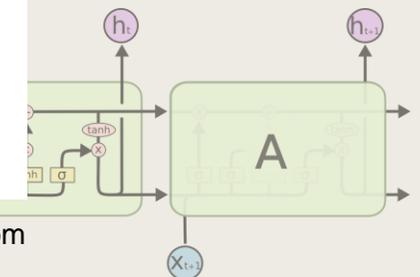


Recurrent Neural Networks

LSTM (Long short-term memory)

er, Jürgen
 „Long short-term
 5)
 Die Eingabe wird
 arbeitet.
 eitung von langen
 .B. Texte, Audio)

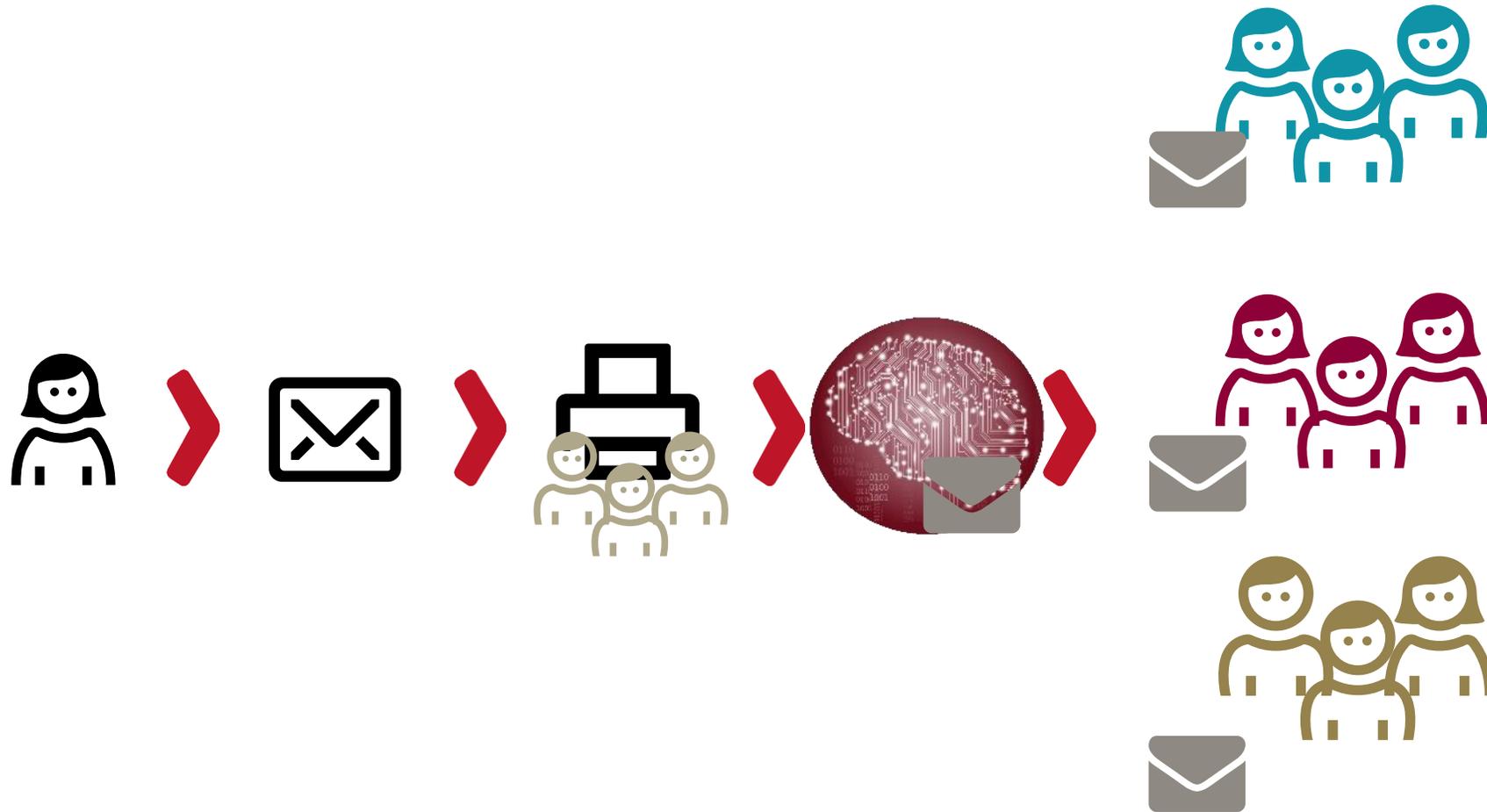
Zelle):



Quelle: Networks and Deep Learning
 Michael Nielsen
<http://neuralnetworksanddeeplearning.com>

Quelle: colah.github.io/posts/2015-08-Understanding-LSTMs/

Eingangsmangement



Häufigkeitsverteilung der Eingangsklassen

ERGO

A Munich Re company

113-Rückkauf

20-Aufhebung Sicherst.

1-Abbuchungsauftrag

144-Sicherstellung

9-Änderung_Adresse

101-Prämienfreistellung

401-Rechnung

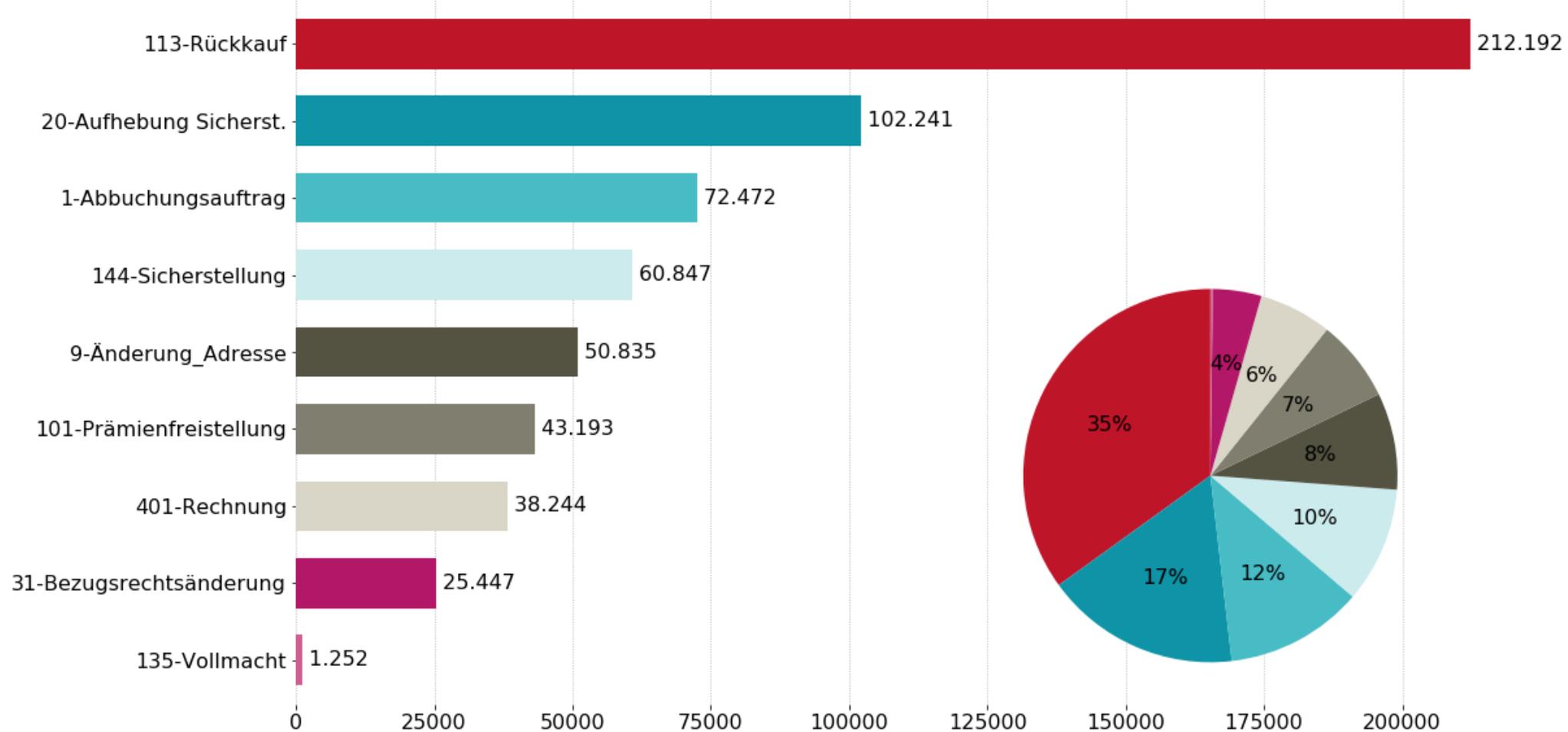
31-Bezugsrechtsänderung

135-Vollmacht

Häufigkeitsverteilung der Eingangsklassen

ERGO

A Munich Re company



Modellierungsansätze – Simple Word Embedding Models (SWEM)

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

Dinghan Shen¹, Guoyin Wang¹, Wenlin Wang¹, Martin Renqiang Min²

Qinliang Su³, Yizhe Zhang⁴, Chunyuan Li¹, Ricardo Henao¹, Lawrence Carin¹

¹ Duke University ² NEC Laboratories America ³ Sun Yat-sen University ⁴ 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 Shen¹, Guoyin Wang¹, Wenlin Wang¹, Martin Renqiang Min²

Qinliang Su³, Yizhe Zhang⁴, Chunyuan Li¹, Ricardo Henao¹, Lawrence Carin¹

¹ Duke University ² NEC Laboratories America ³ Sun Yat-sen University ⁴ 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 Shen¹, Guoyin Wang¹, Wenlin Wang¹, Martin Renqiang Min²

Qinliang Su³, Yizhe Zhang⁴, Chunyuan Li¹, Ricardo Henao¹, Lawrence Carin¹

¹ Duke University ² NEC Laboratories America ³ Sun Yat-sen University ⁴ 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)



A Munich Re company

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

Dinghan Shen¹, Guoyin Wang¹, Wenlin Wang¹, Martin Renqiang Min²

Qinliang Su³, Yizhe Zhang⁴, Chunyuan Li¹, Ricardo Henao¹, Lawrence Carin¹

¹ Duke University ² NEC Laboratories America ³ Sun Yat-sen University ⁴ 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>

09843v1 [cs.CL] 24 May 2018

Modellierungsansätze – Simple Word Embedding Models (SWEM)

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

Dinghan Shen¹, Guoyin Wang¹, Wenlin Wang¹, Martin Renqiang Min²

Qinliang Su³, Yizhe Zhang⁴, Chunyuan Li¹, Ricardo Henao¹, Lawrence Carin¹

¹ Duke University ² NEC Laboratories America ³ Sun Yat-sen University ⁴ 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)

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-MAX
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-MAX
1	0.1	0.2	0.3	0.1	0.1	0.9	0.1	0.8	0.9
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-MAX
1	0.1	0.2	0.3	0.1	0.1	0.9	0.1	0.8	0.9
2	0.2	0.2	0.8	0.9	0.8	0.1	0.1	0.1	0.9
:	:	:	:	:	:	:	:	:	:
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-MAX
1	0.1	0.2	0.3	0.1	0.1	0.9	0.1	0.8	0.9
2	0.2	0.2	0.8	0.9	0.8	0.1	0.1	0.1	0.9
:	:	:	:	:	:	:	:	:	:
20	0.7	0.8	0.1	0.1	0.1	0.2	0.6	0.1	0.8

Modellierungsansätze – Simple Word Embedding Models (SWEM)

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

Modellierungsansätze – Simple Word Embedding Models (SWEM)

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

Modellierungsansätze – Simple Word Embedding Models (SWEM)

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

Modellierungsansätze – Simple Word Embedding Models (SWEM)

Embedding-Dimension	Anbei	sende	ich	Ihnen	meine	Rechnung	zum	Schaden	SWEM-MAX	SWEM-AVG
1	0.1	0.2	0.3	0.1	0.1	0.9	0.1	0.8	0.9	0.325
2	0.2	0.2	0.8	0.9	0.8	0.1	0.1	0.1	0.9	0.4
:	:	:	:	:	:	:	:	:	:	:
20	0.7	0.8	0.1	0.1	0.1	0.2	0.6	0.1	0.8	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 $\in \mathbb{R}^{14}$

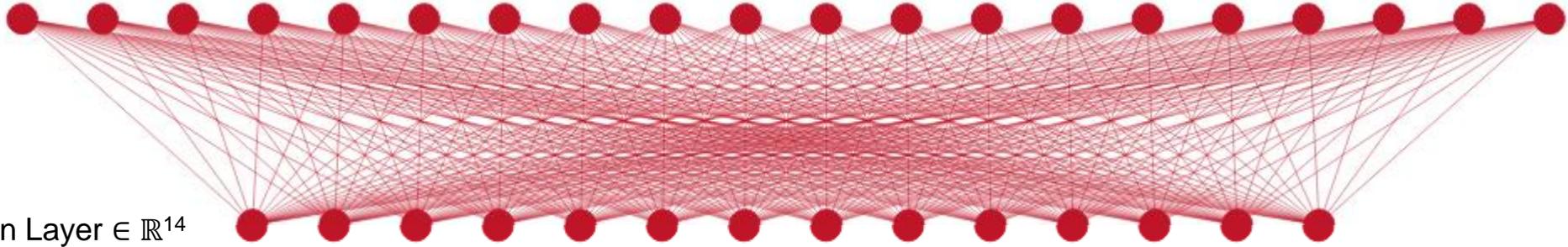


Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

Modellierungsansätze – Simple Word Embedding Models (SWEM)

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

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



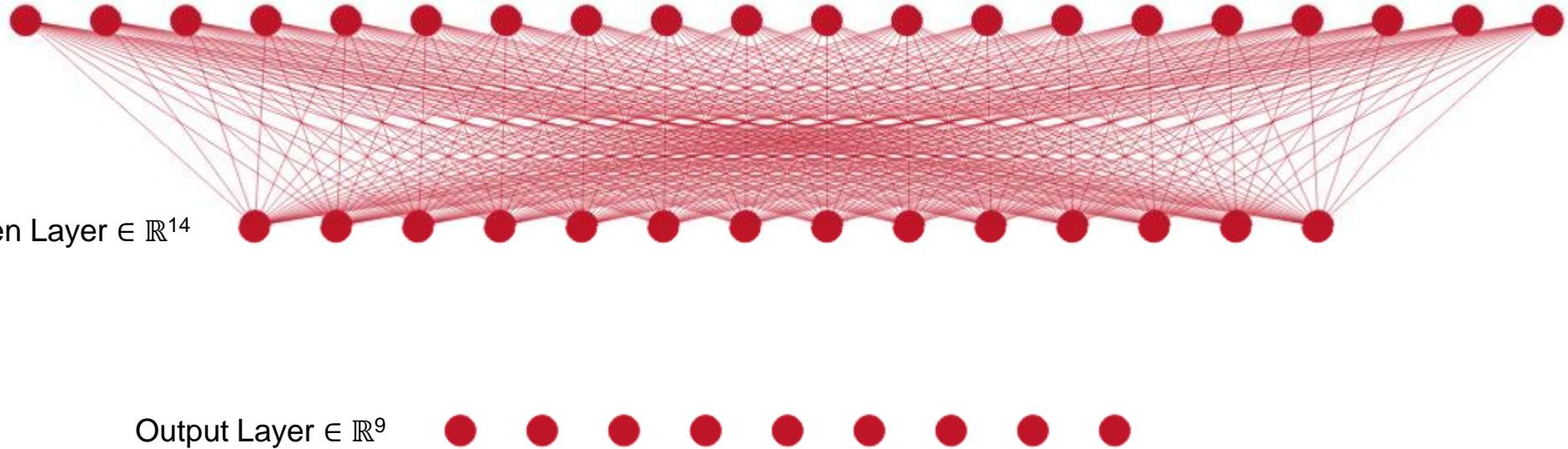
Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

Modellierungsansätze – Simple Word Embedding Models (SWEM)

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

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

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



Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

Modellierungsansätze – Simple Word Embedding Models (SWEM)

ERGO

A Munich Re company

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

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

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



93,90% Accuracy

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)

PROC. OF THE IEEE, NOVEMBER 1998

1

Gradient-Based Learning Applied to Document Recognition

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

Abstract—

Multilayer Neural Networks trained with the backpropagation algorithm constitute the best example of a successful Gradient-Based Learning technique. Given an appropriate network architecture, Gradient-Based Learning algorithms can be used to synthesize a complex decision surface that can classify high-dimensional patterns such as handwritten characters, with minimal preprocessing. This paper reviews various methods applied to handwritten character recognition and compares them on a standard handwritten digit recognition task. Convolutional Neural Networks, that are specifically designed to deal with the variability of 2D shapes, are shown to outperform all other techniques.

I. INTRODUCTION

Over the last several years, machine learning techniques, particularly when applied to neural networks, have played an increasingly important role in the design of pattern recognition systems. In fact, it could be argued that the availability of learning techniques has been a crucial factor in the recent success of pattern recognition applications such as continuous speech recognition and handwriting recognition.

LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278-2324.

Modellierungsansätze – Convolutional Neural Networks (CNN)

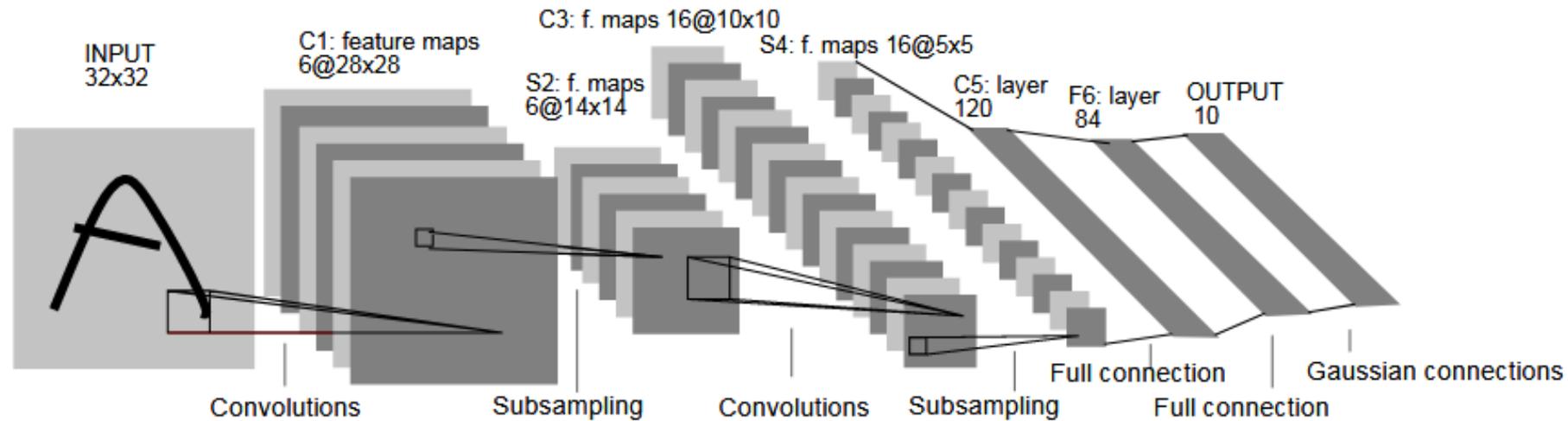


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

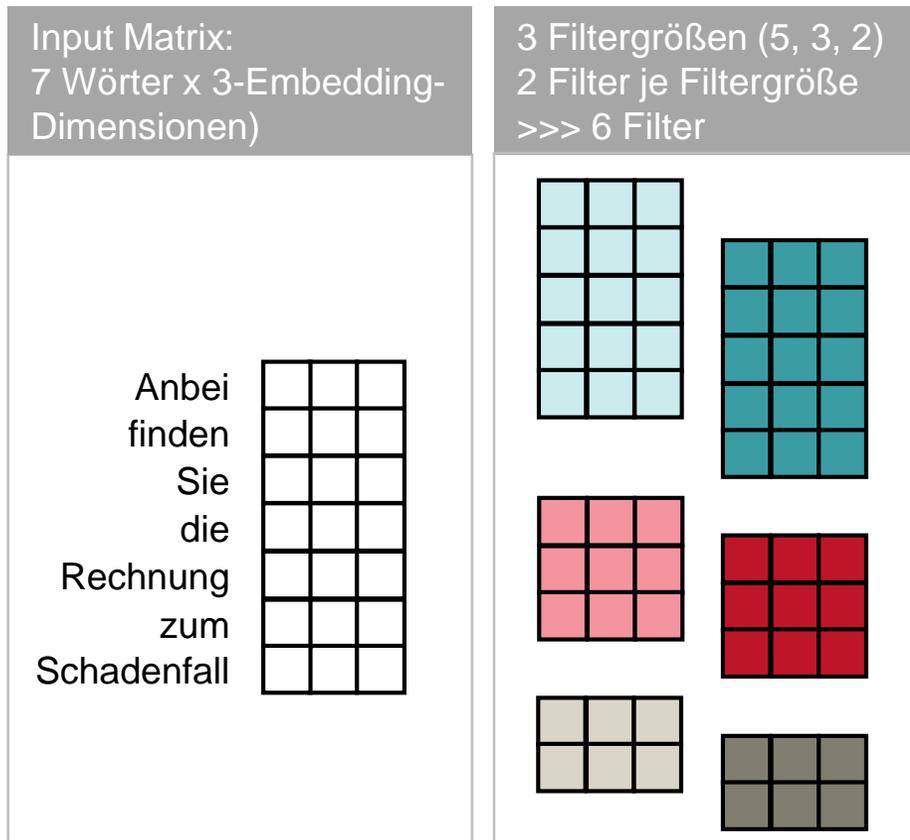
LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278-2324.

Modellierungsansätze – Convolutional Neural Networks (CNN)

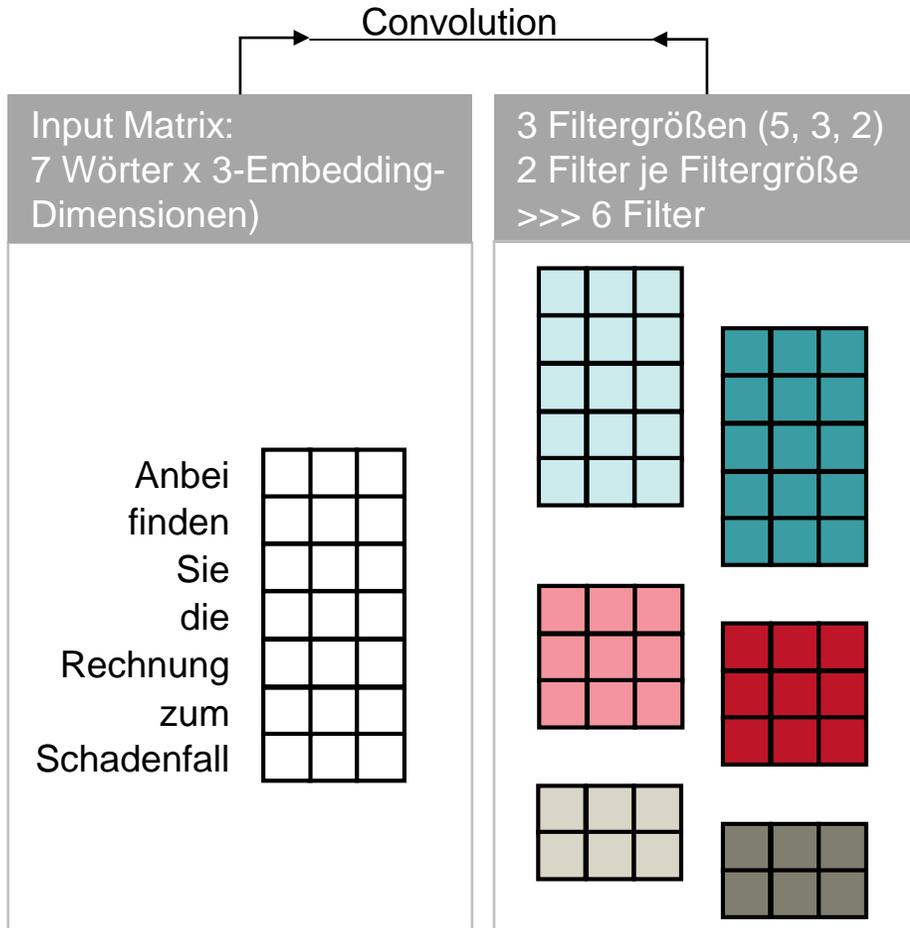
Input Matrix:
7 Wörter x 3-Embedding-
Dimensionen)

Anbei			
finden			
Sie			
die			
Rechnung			
zum			
Schadenfall			

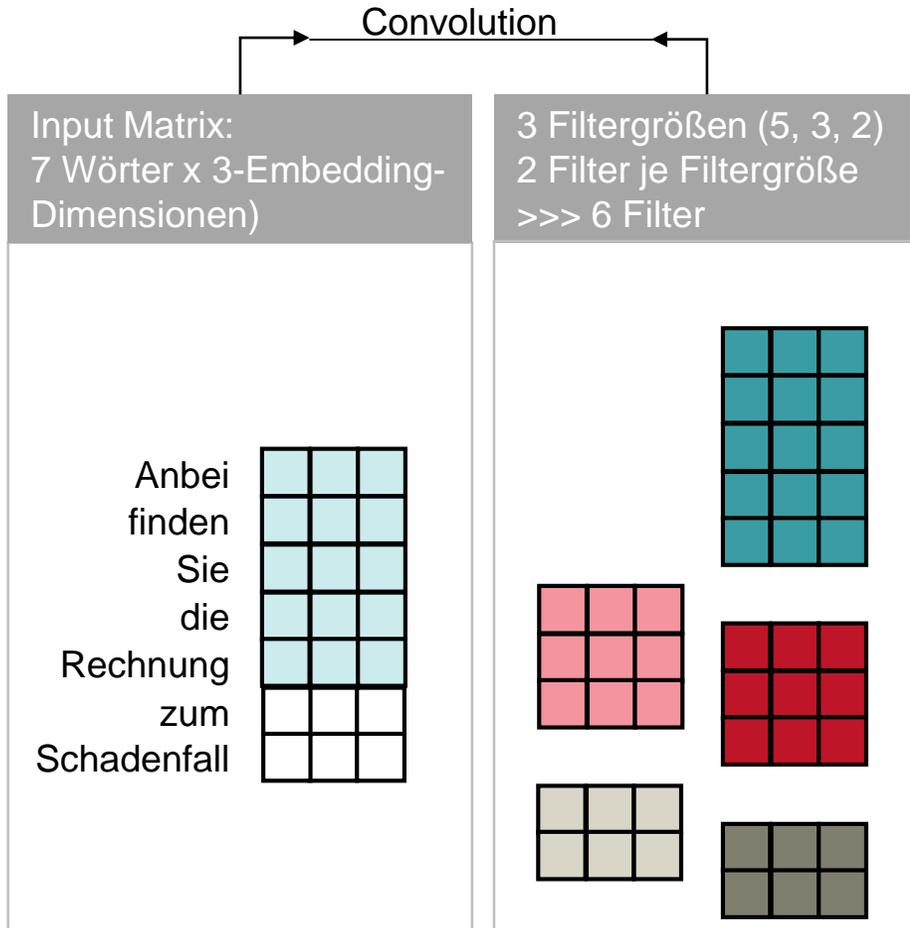
Modellierungsansätze – Convolutional Neural Networks (CNN)



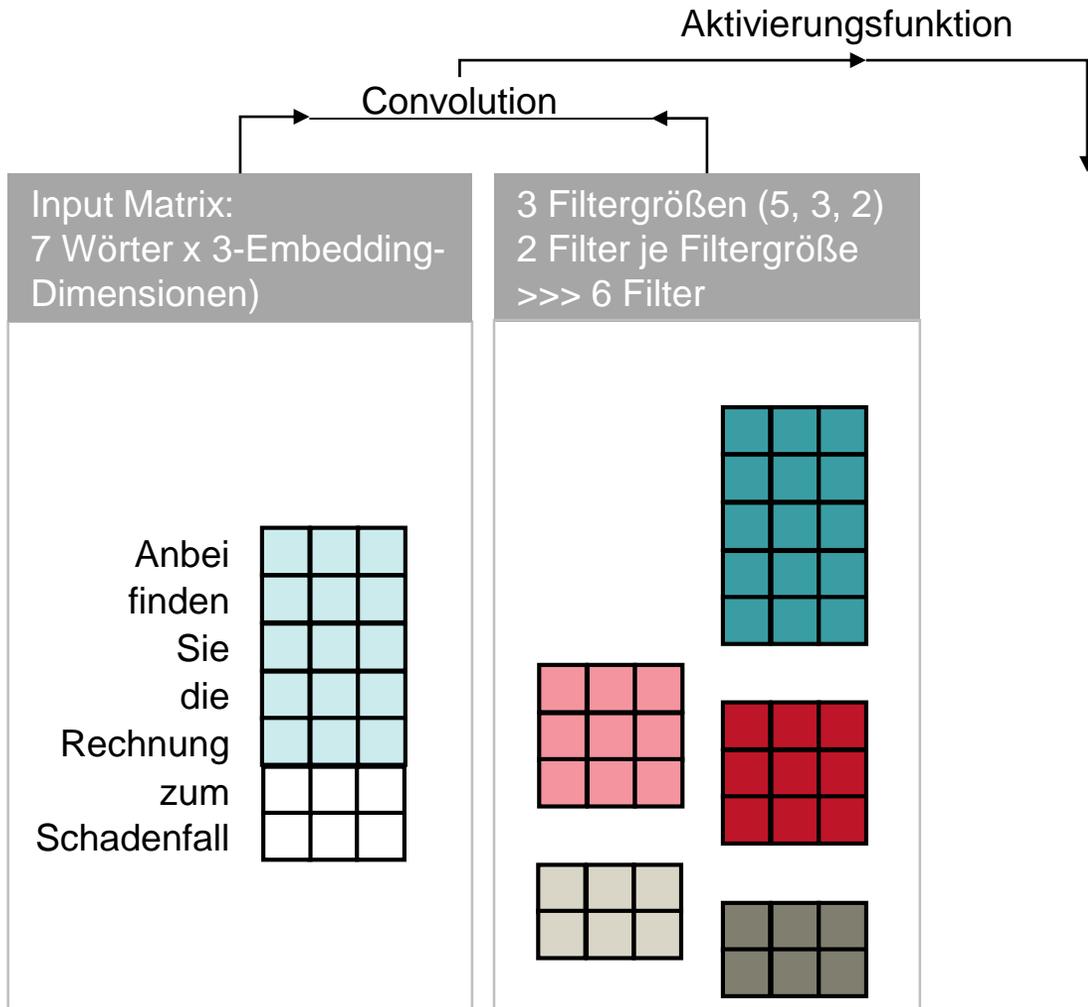
Modellierungsansätze – Convolutional Neural Networks (CNN)



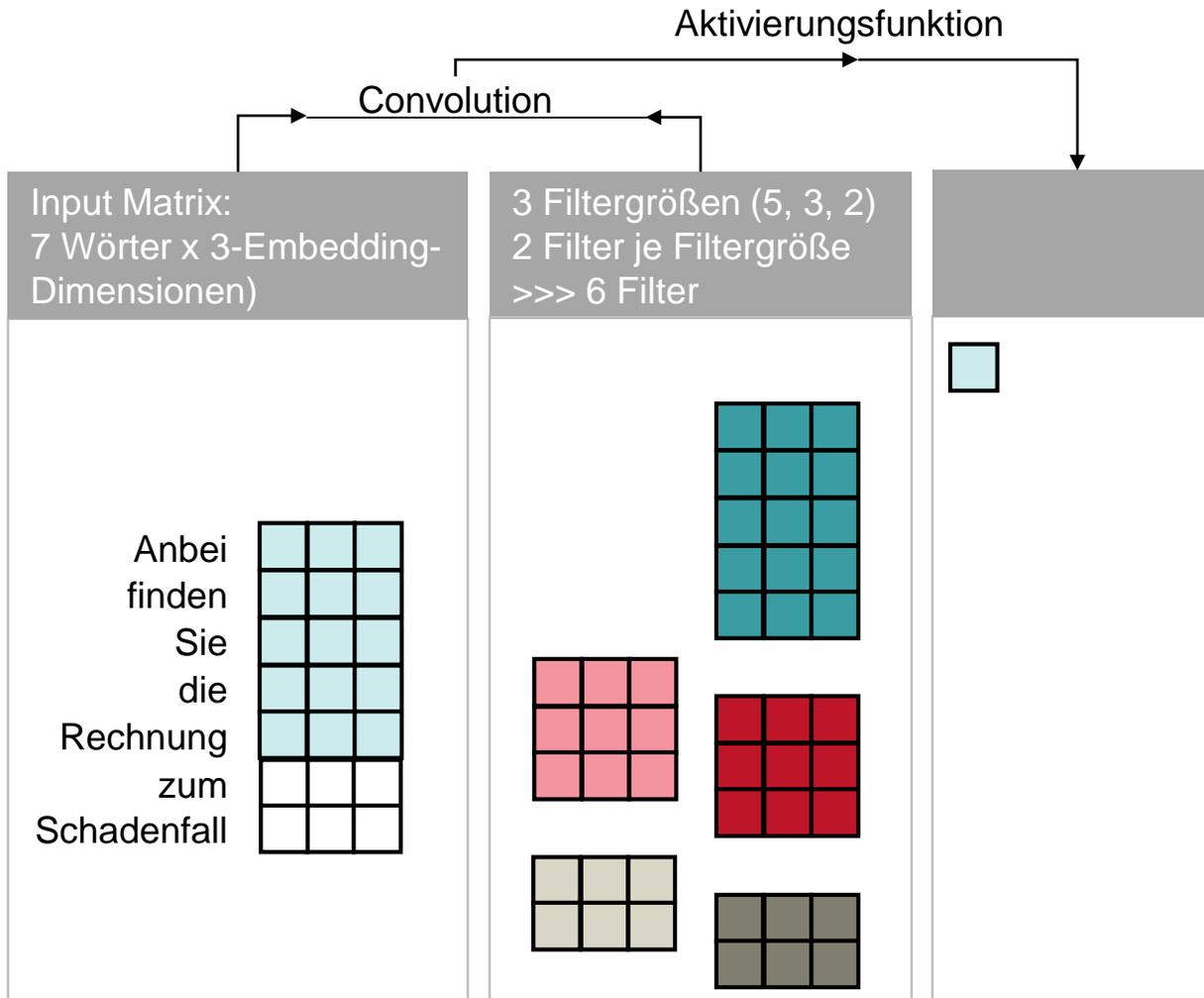
Modellierungsansätze – Convolutional Neural Networks (CNN)



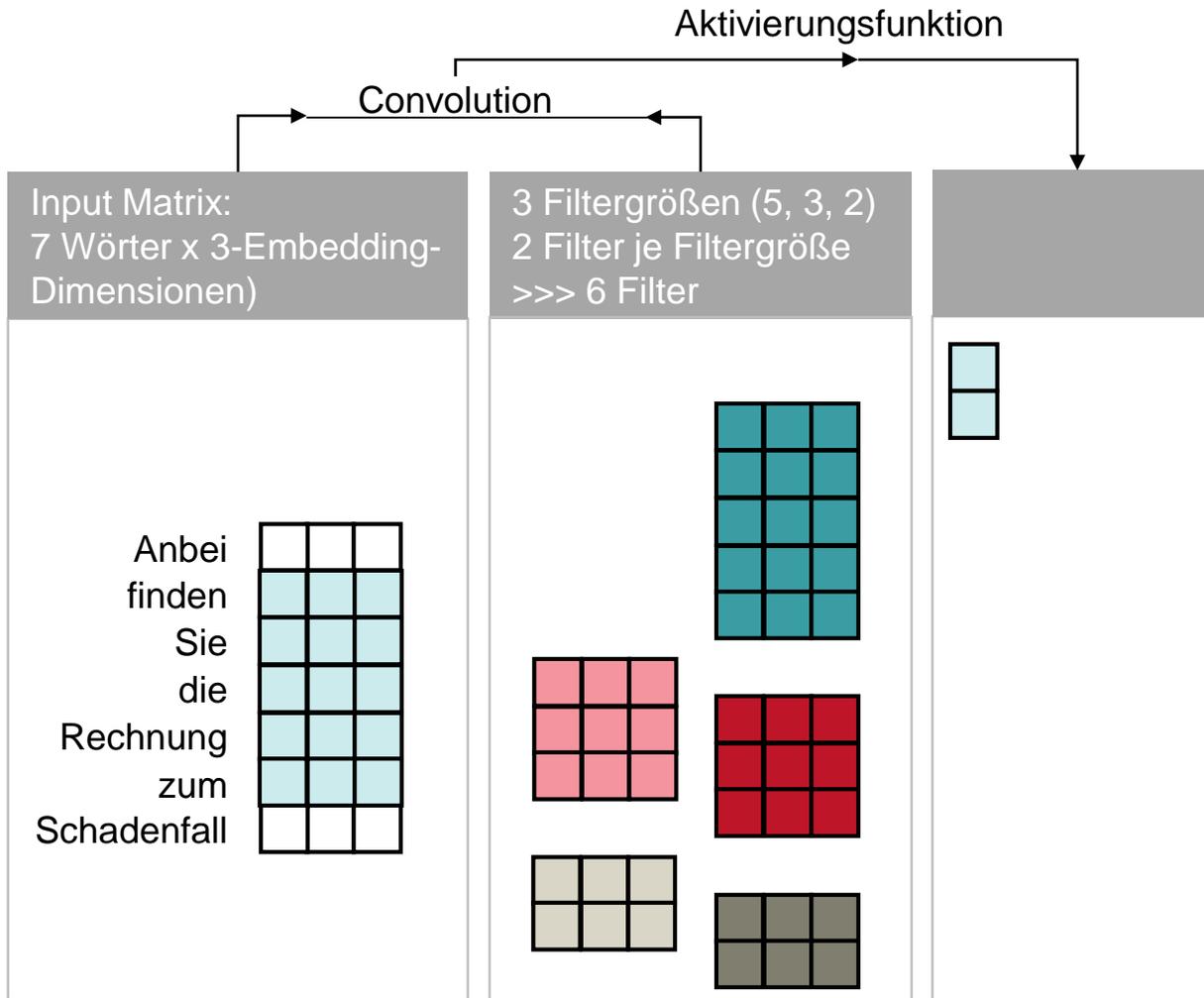
Modellierungsansätze – Convolutional Neural Networks (CNN)



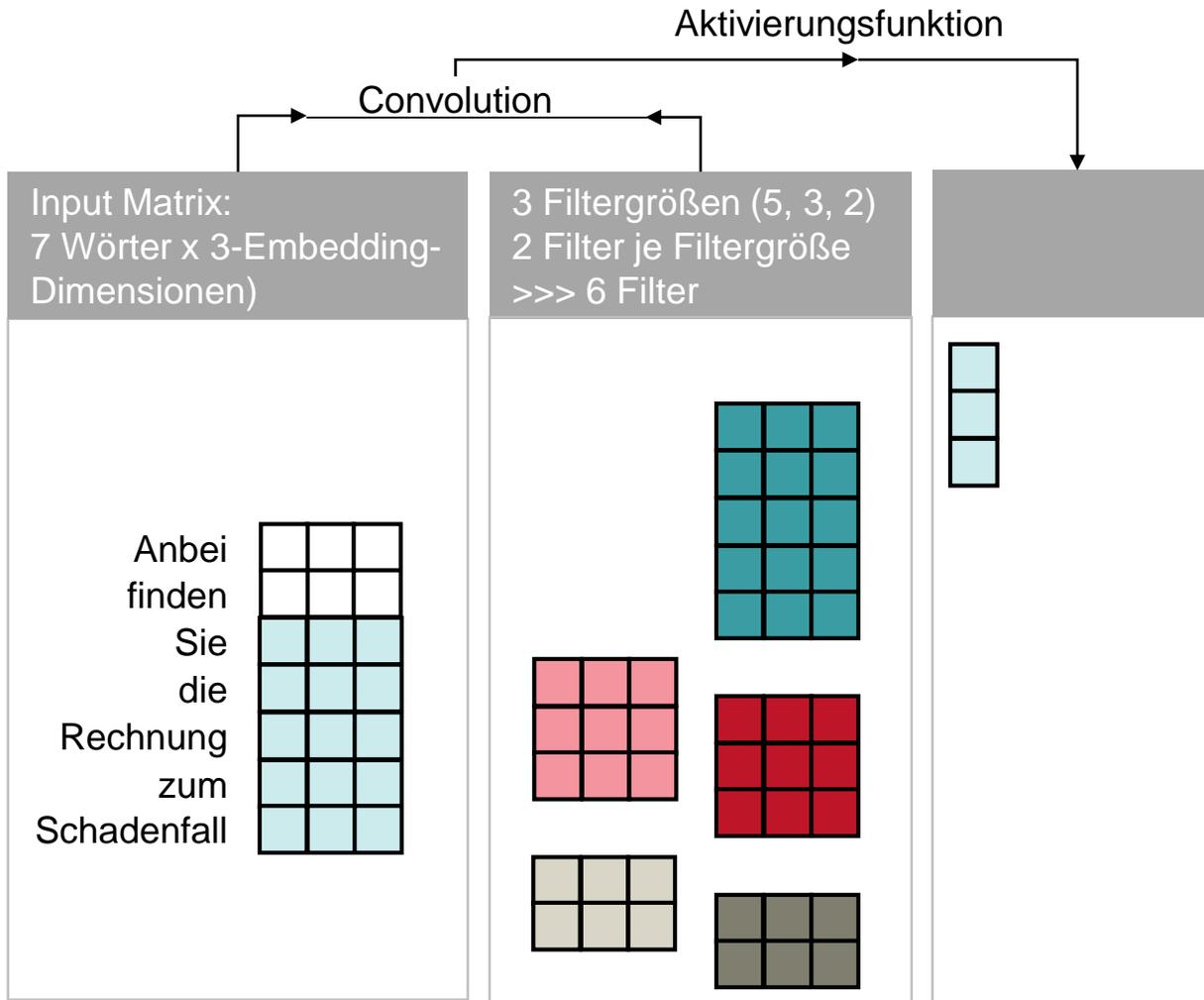
Modellierungsansätze – Convolutional Neural Networks (CNN)



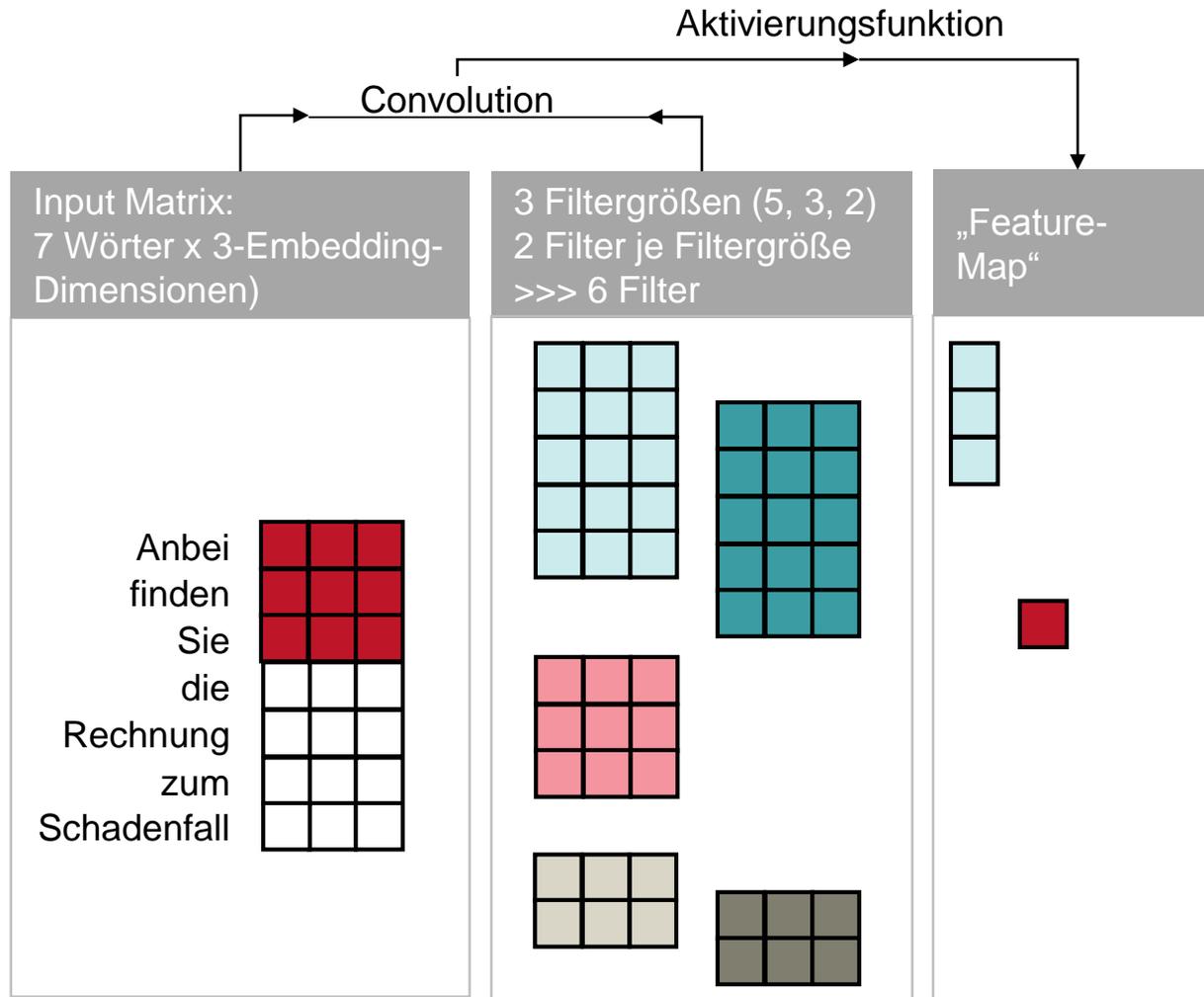
Modellierungsansätze – Convolutional Neural Networks (CNN)



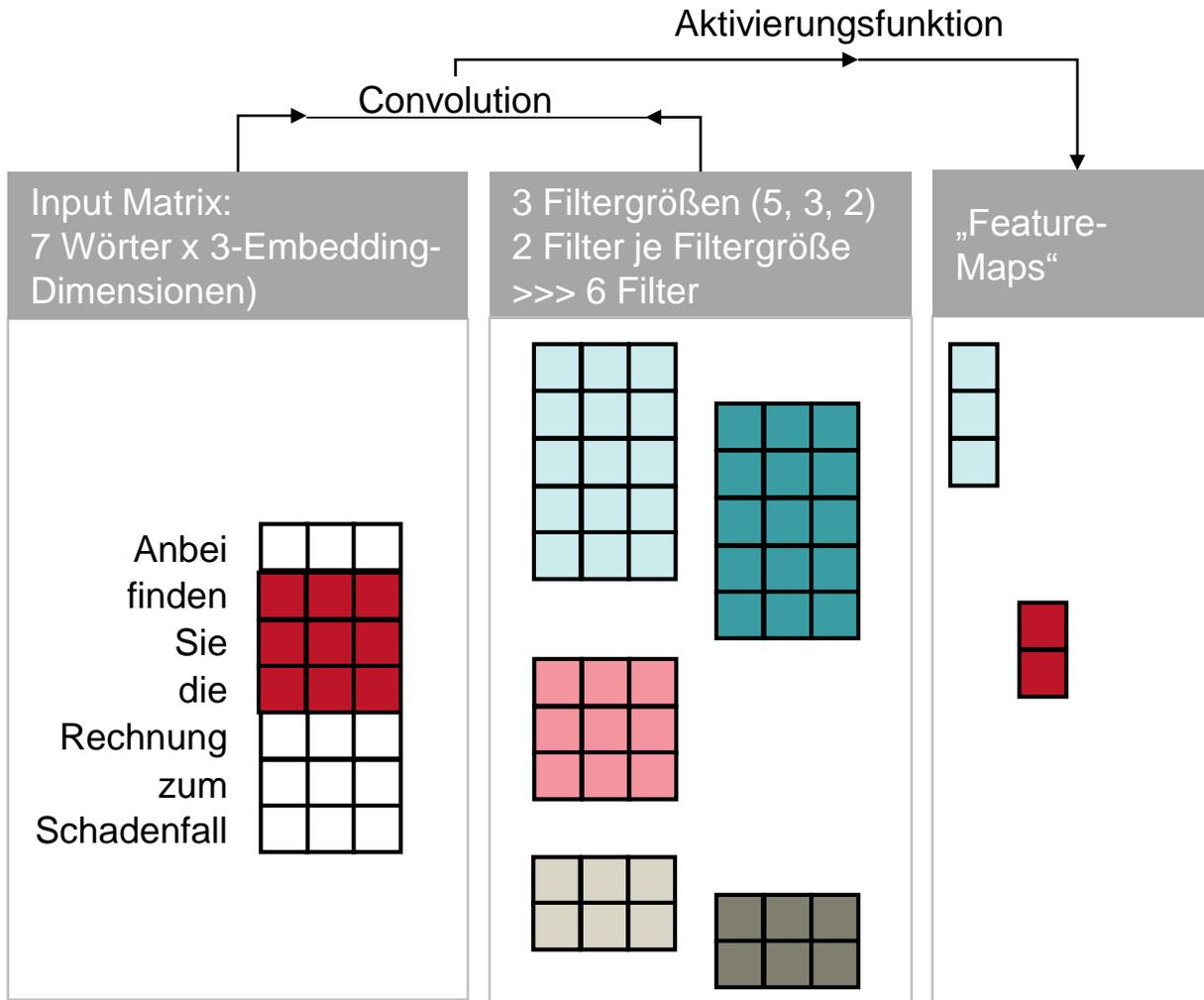
Modellierungsansätze – Convolutional Neural Networks (CNN)



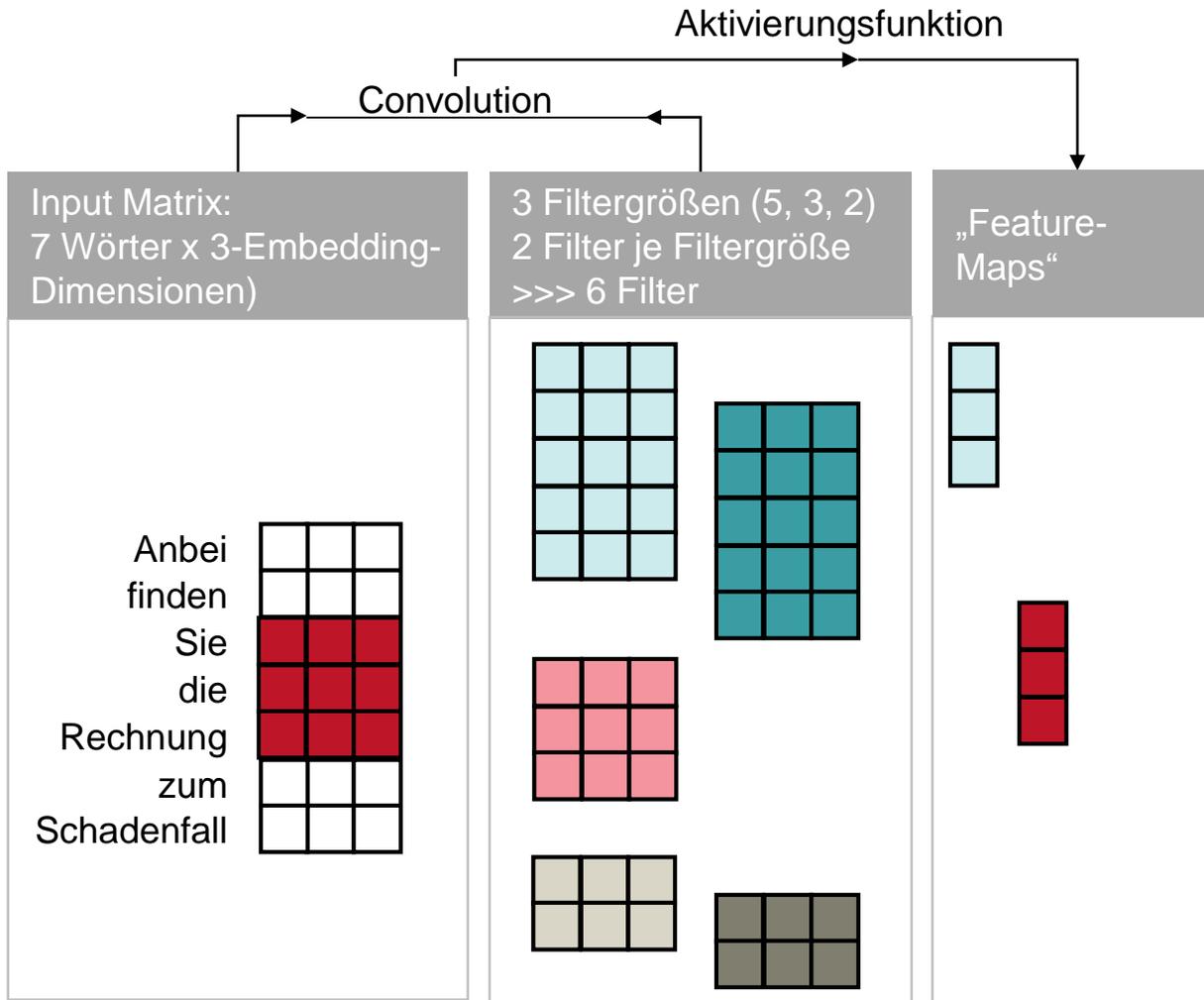
Modellierungsansätze – Convolutional Neural Networks (CNN)



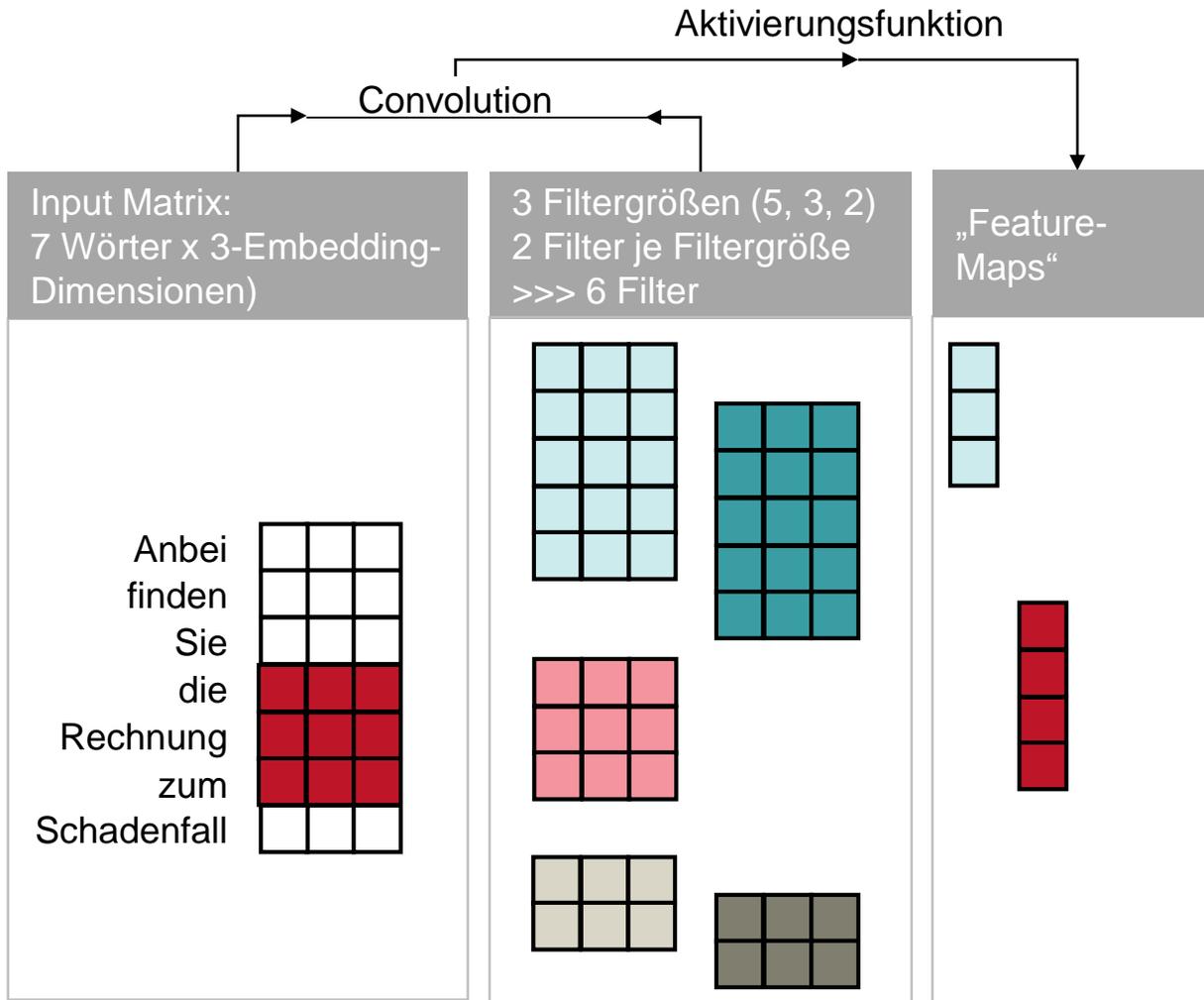
Modellierungsansätze – Convolutional Neural Networks (CNN)



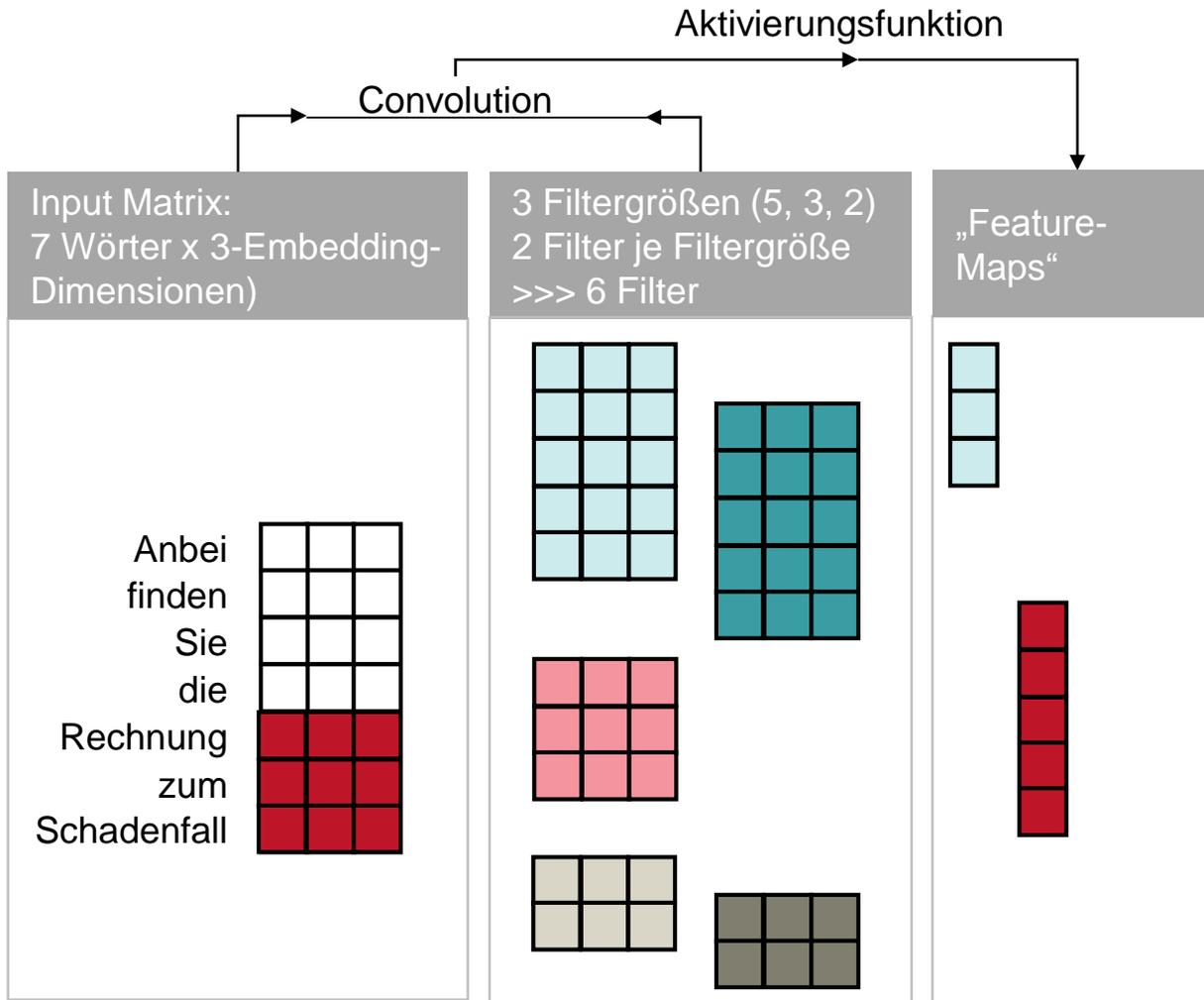
Modellierungsansätze – Convolutional Neural Networks (CNN)



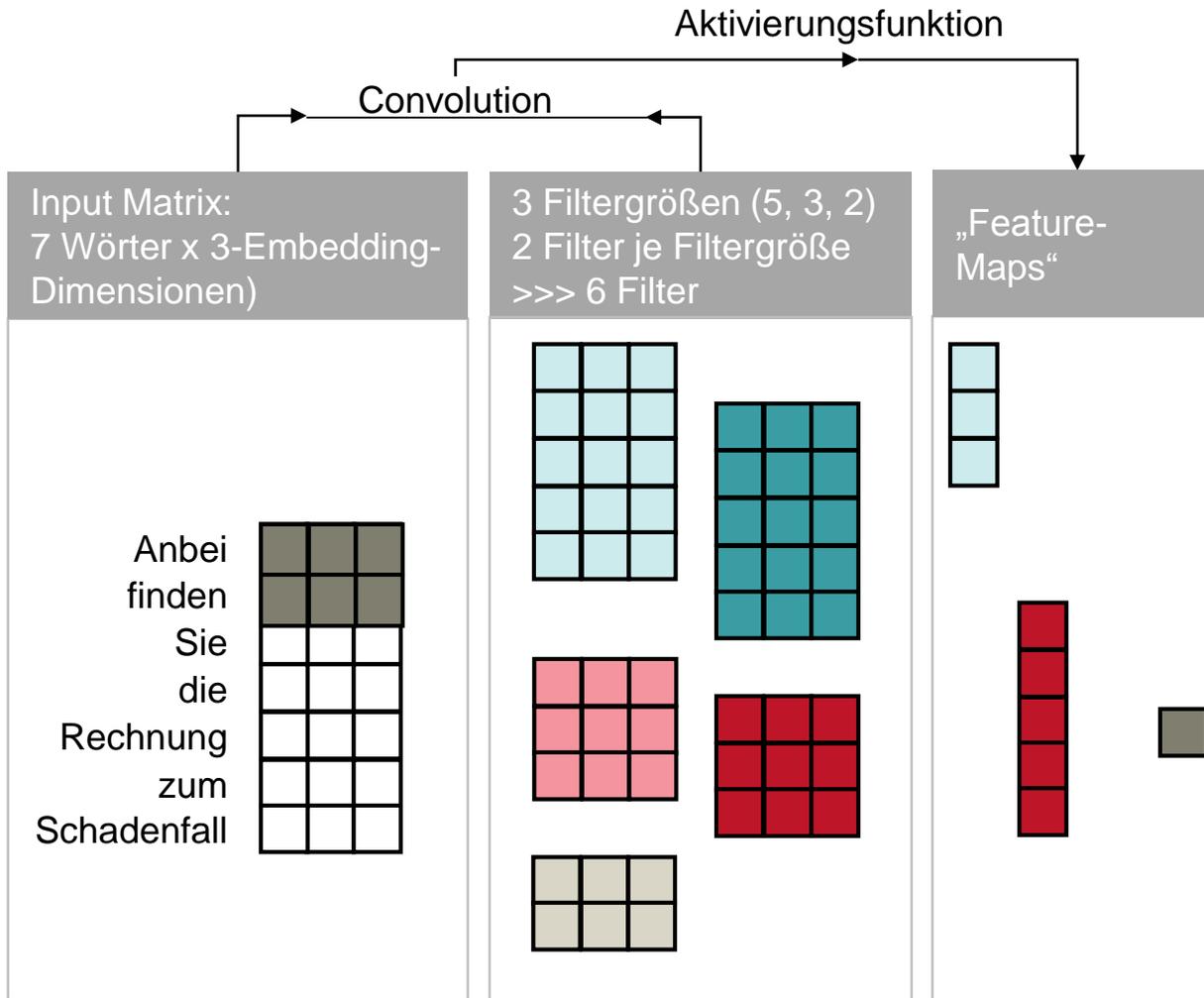
Modellierungsansätze – Convolutional Neural Networks (CNN)



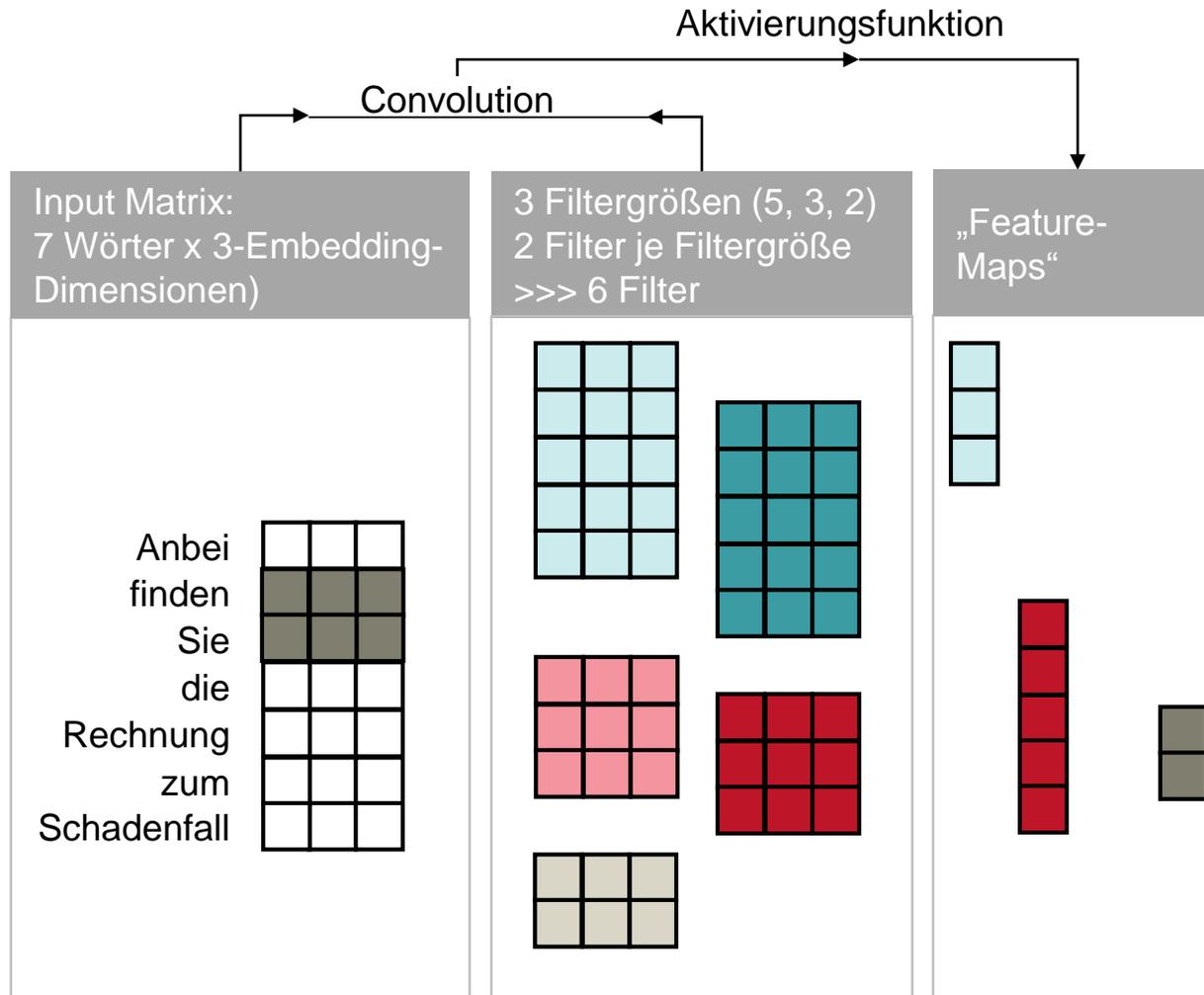
Modellierungsansätze – Convolutional Neural Networks (CNN)



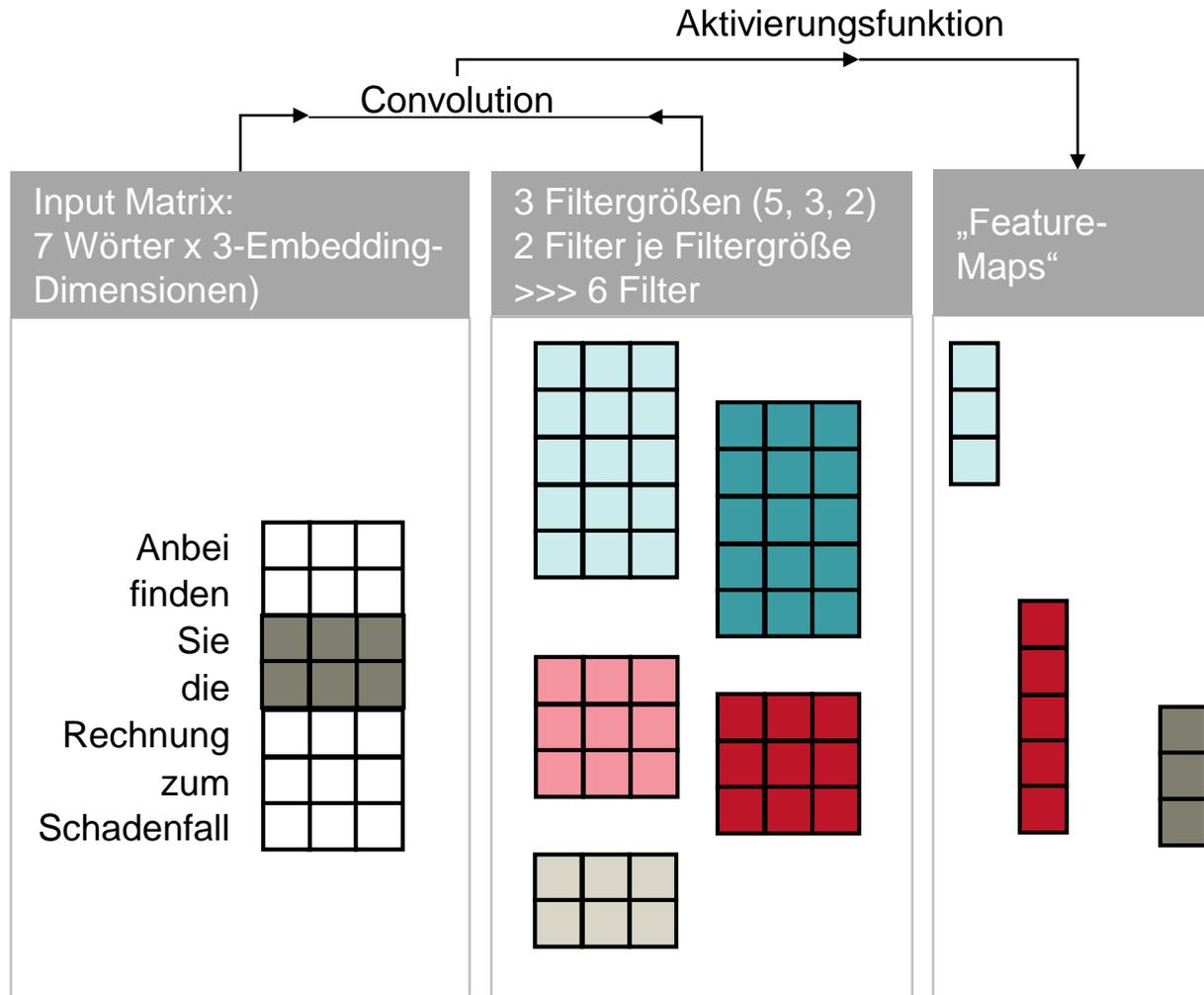
Modellierungsansätze – Convolutional Neural Networks (CNN)



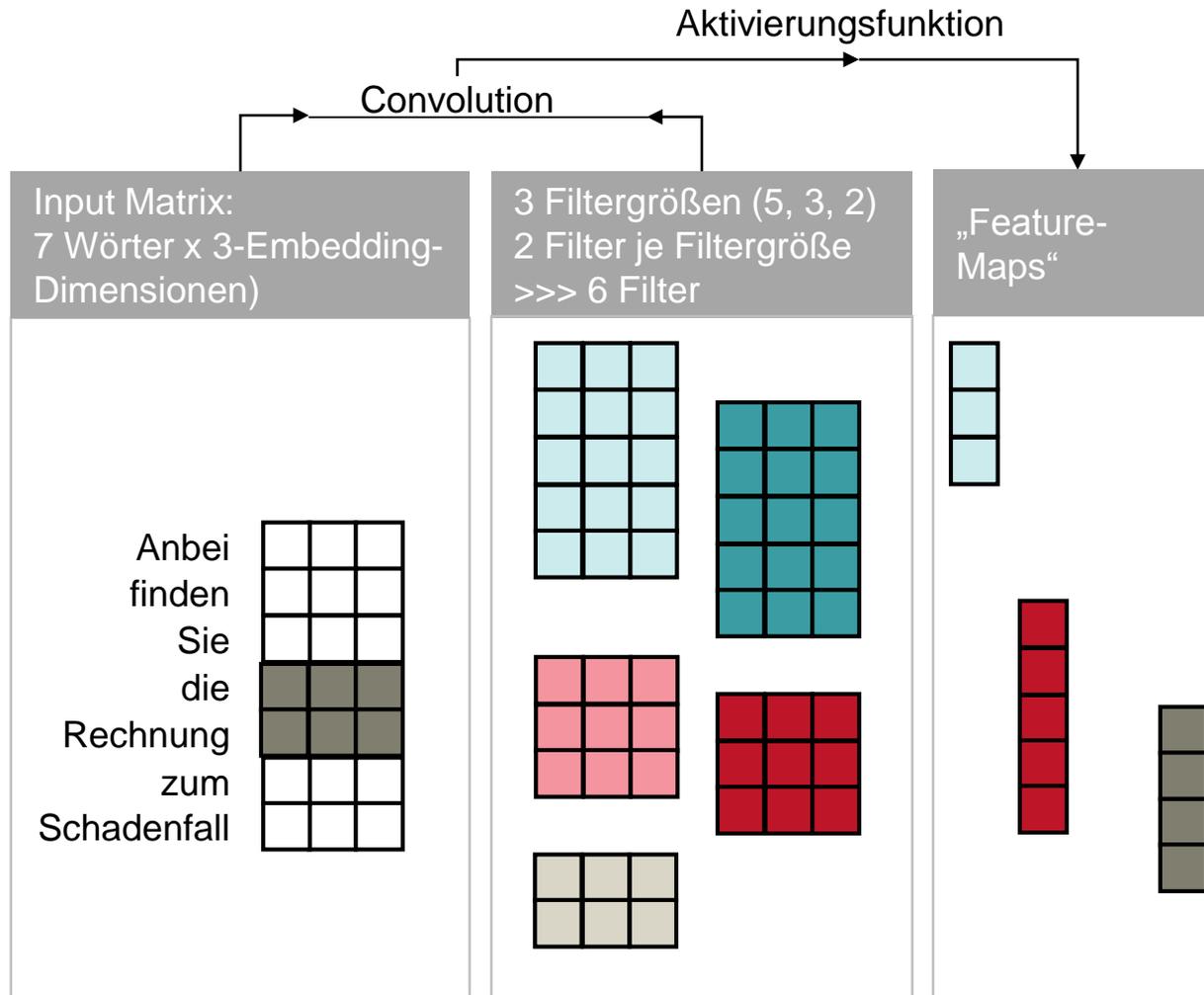
Modellierungsansätze – Convolutional Neural Networks (CNN)



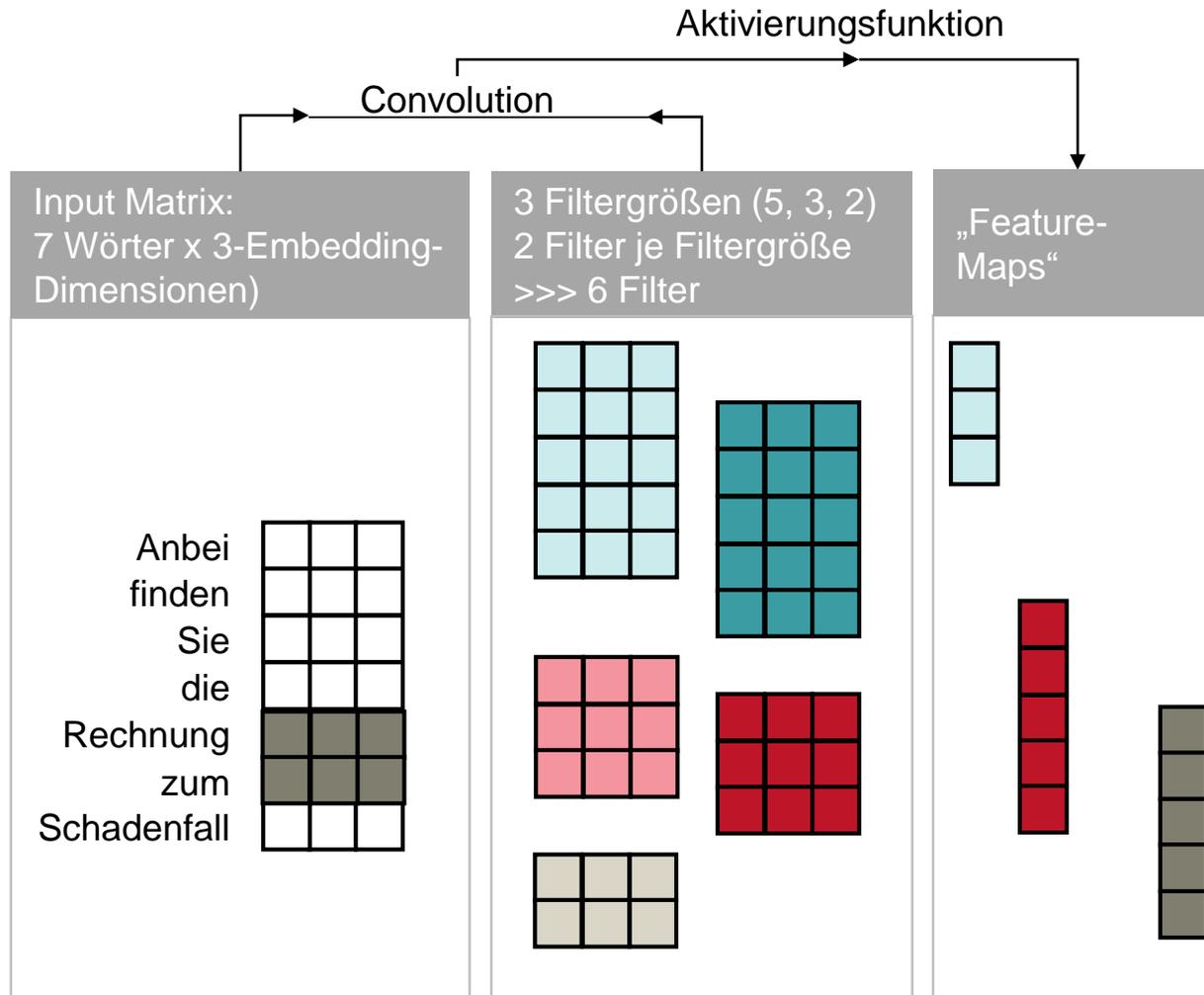
Modellierungsansätze – Convolutional Neural Networks (CNN)



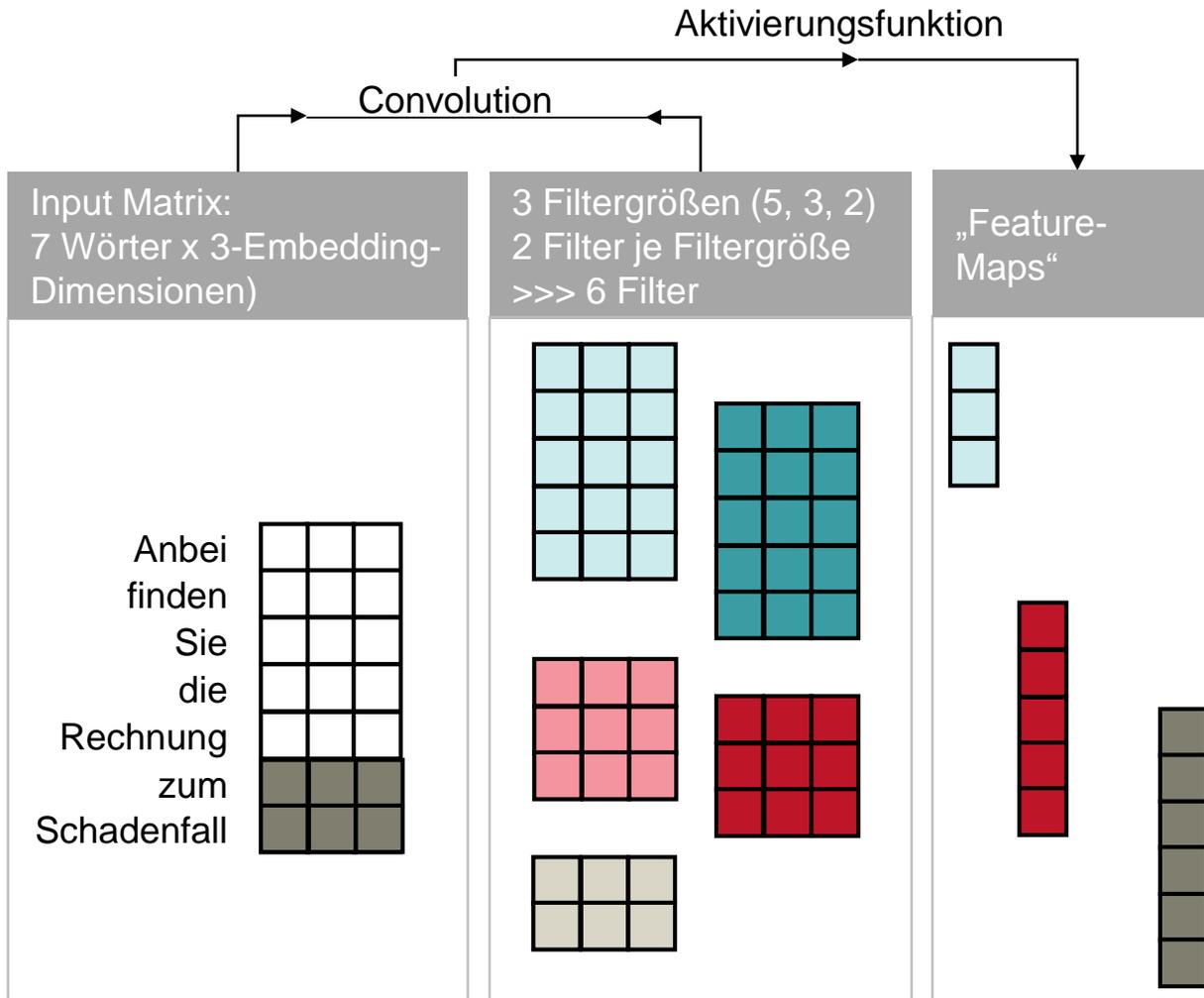
Modellierungsansätze – Convolutional Neural Networks (CNN)



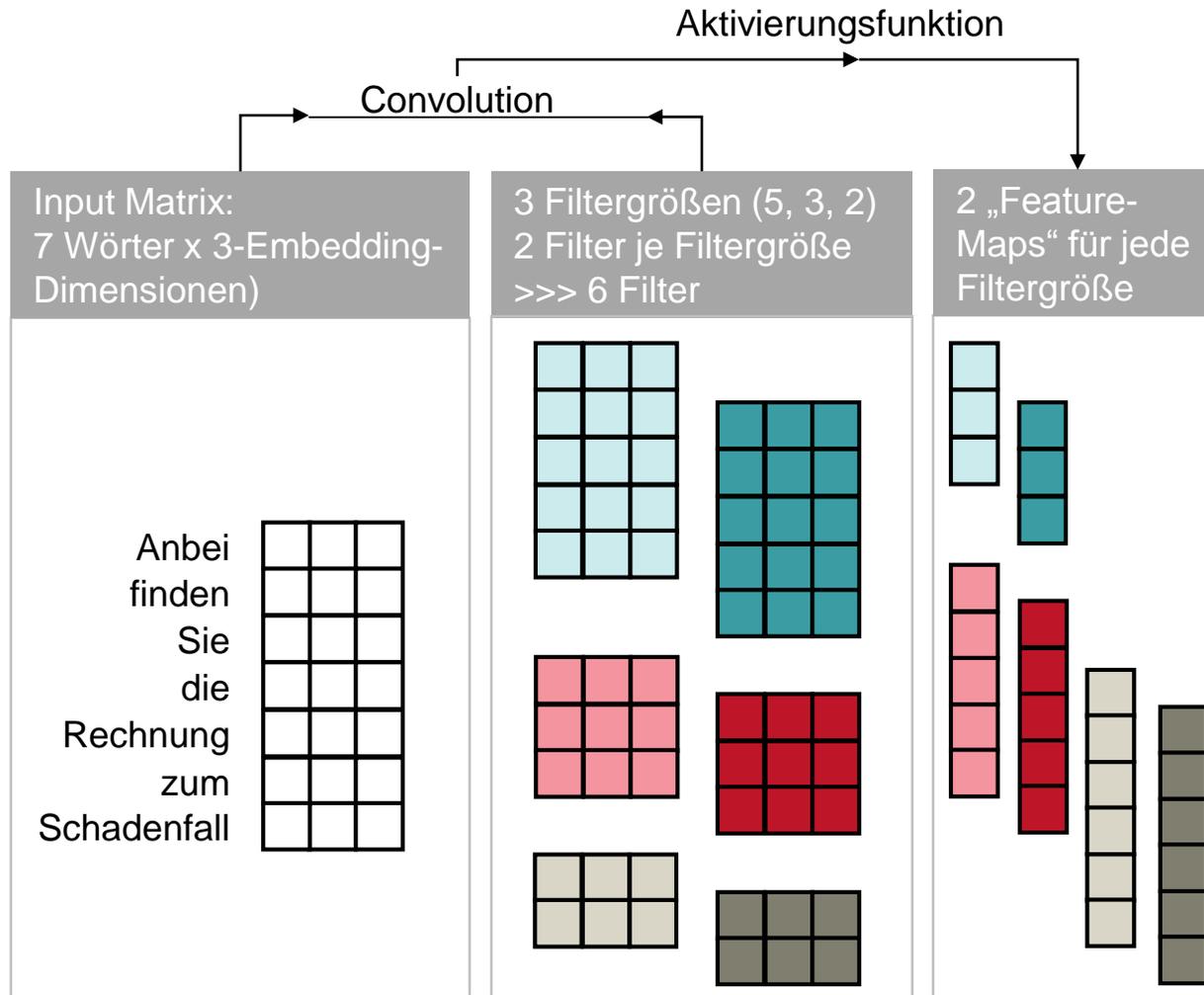
Modellierungsansätze – Convolutional Neural Networks (CNN)



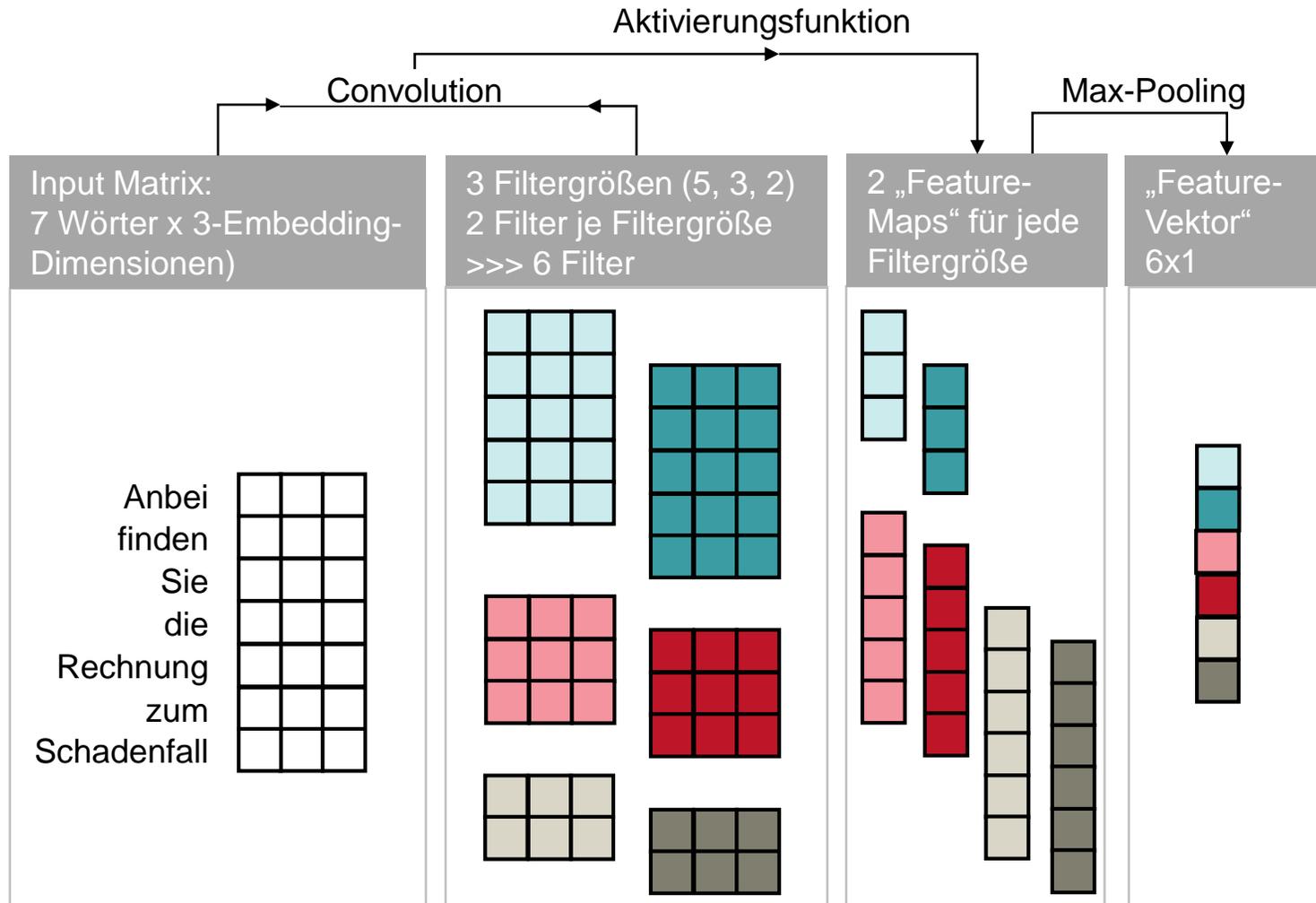
Modellierungsansätze – Convolutional Neural Networks (CNN)



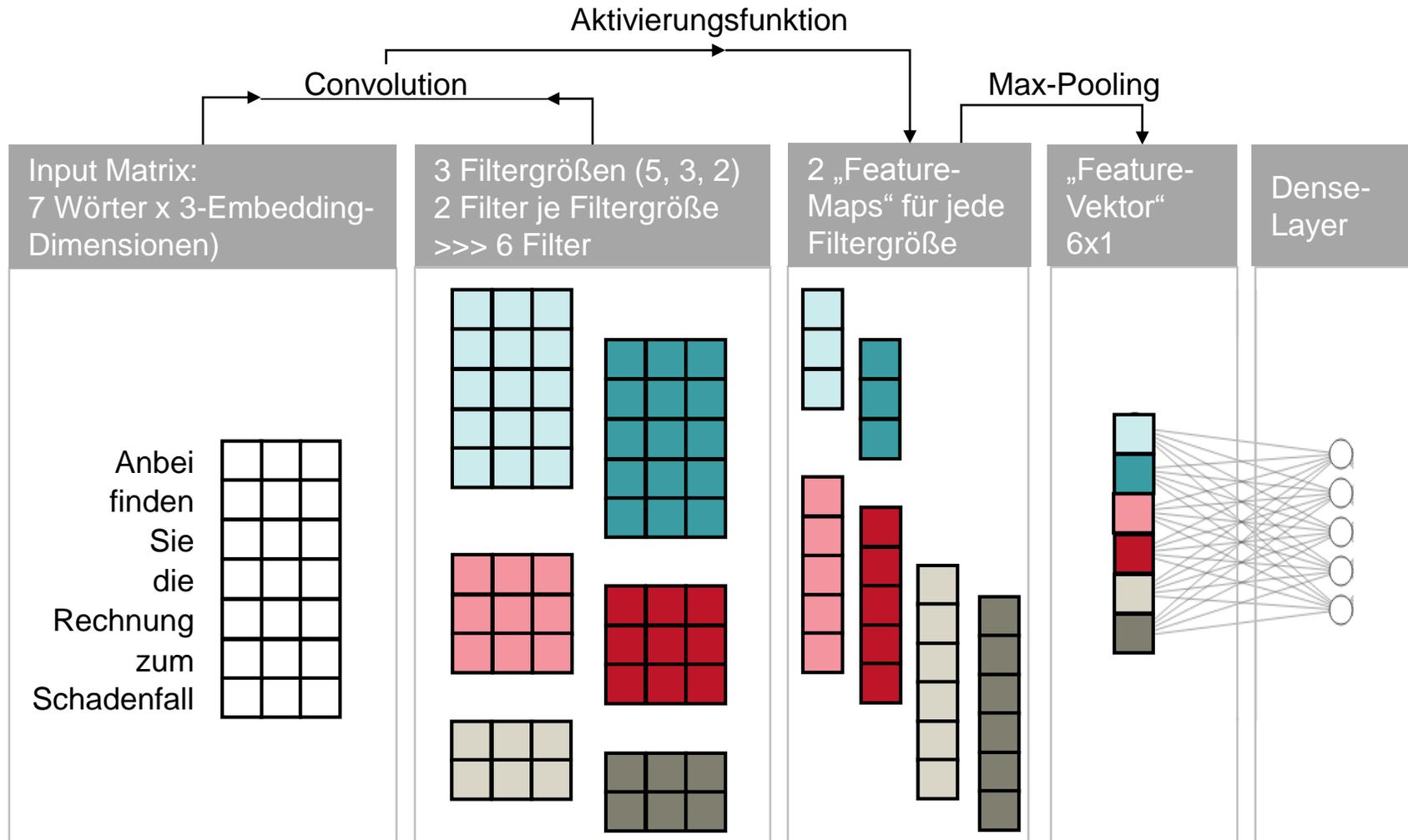
Modellierungsansätze – Convolutional Neural Networks (CNN)



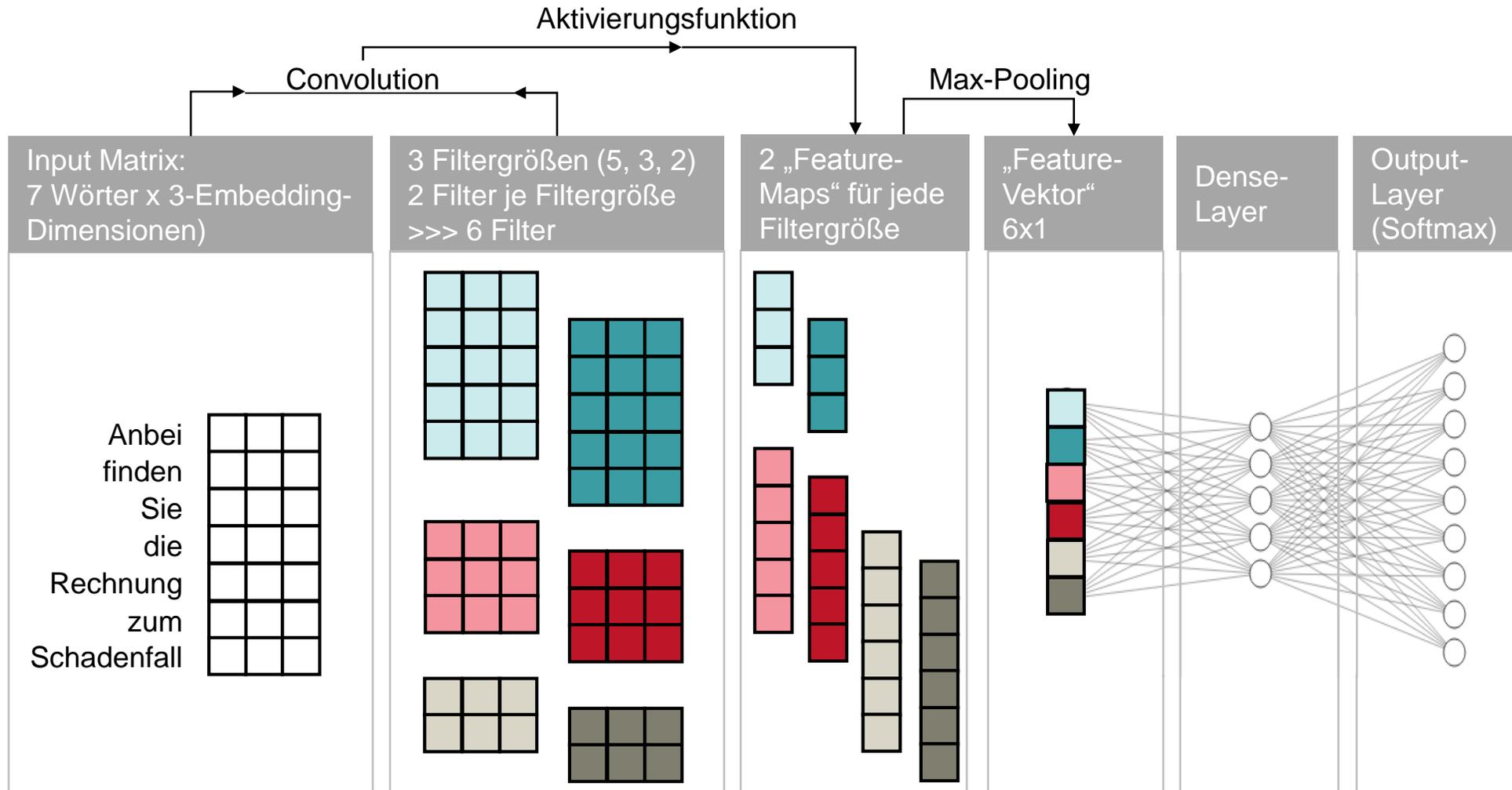
Modellierungsansätze – Convolutional Neural Networks (CNN)



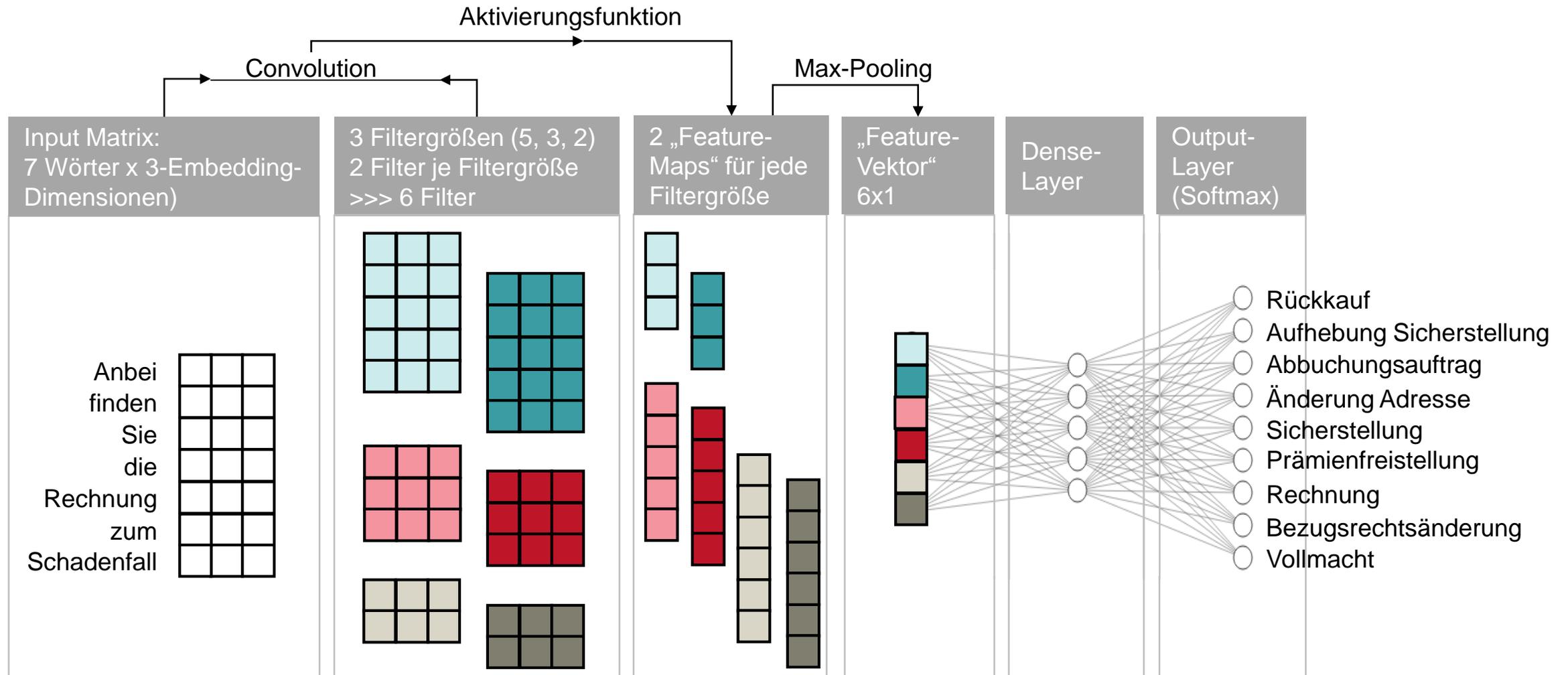
Modellierungsansätze – Convolutional Neural Networks (CNN)



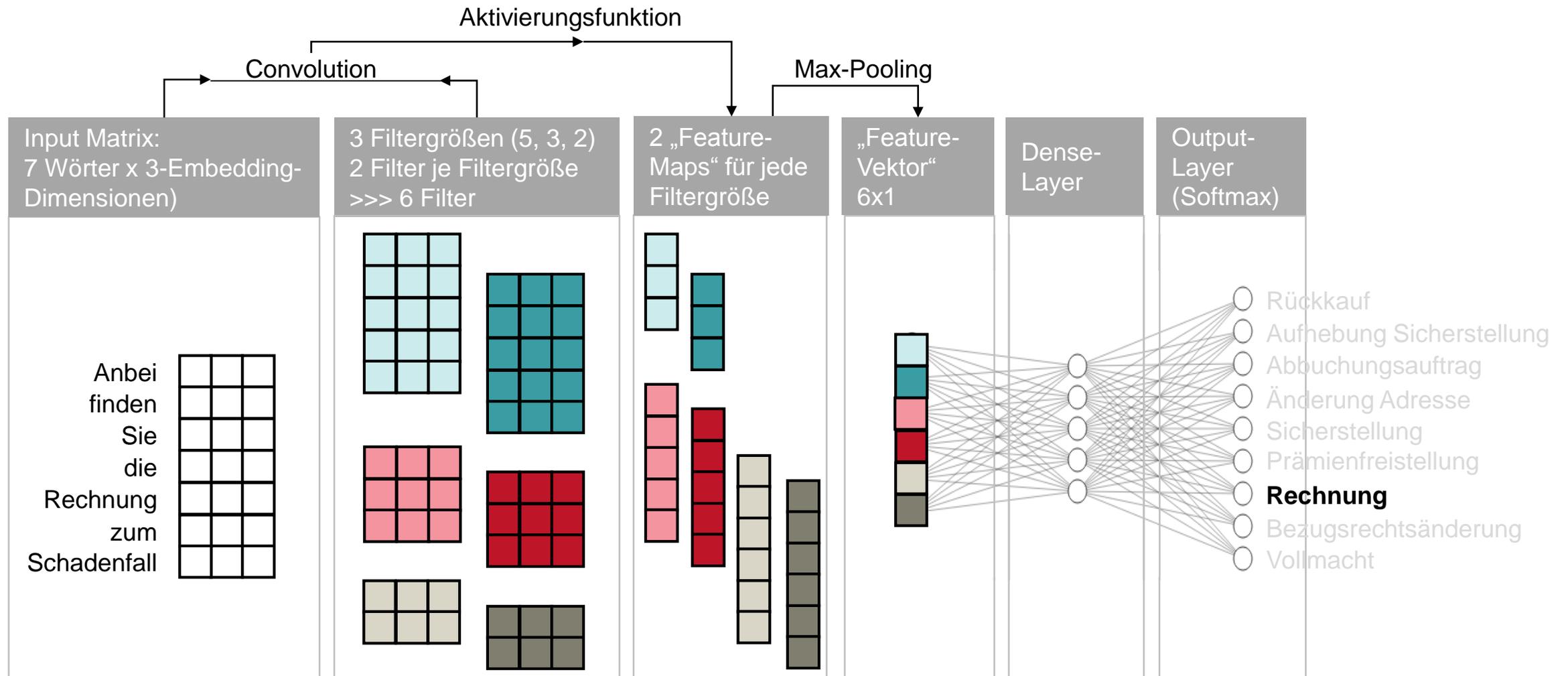
Modellierungsansätze – Convolutional Neural Networks (CNN)



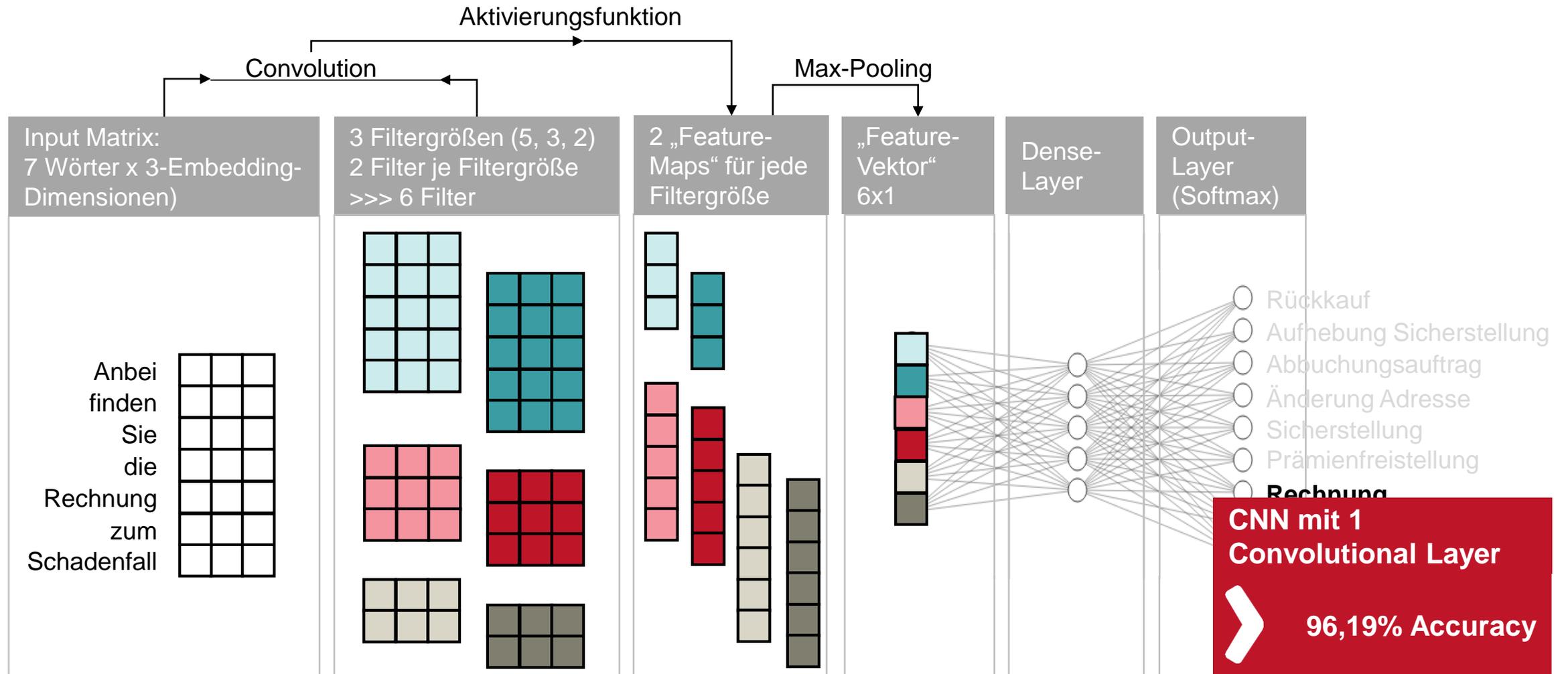
Modellierungsansätze – Convolutional Neural Networks (CNN)



Modellierungsansätze – Convolutional Neural Networks (CNN)



Modellierungsansätze – Convolutional Neural Networks (CNN)



Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

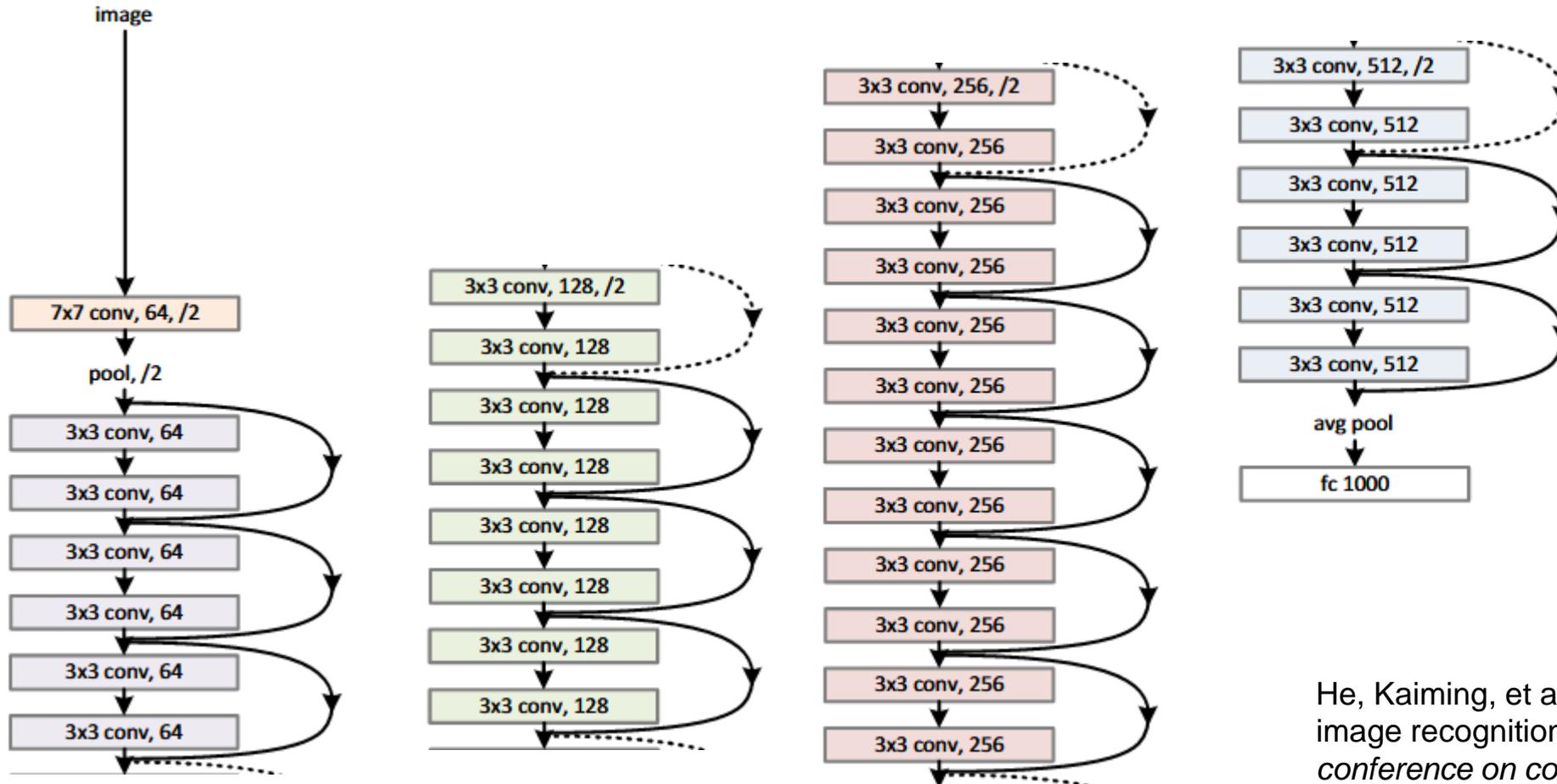
Was in der Bilderkennung funktioniert...

ERGO

A Munich Re company

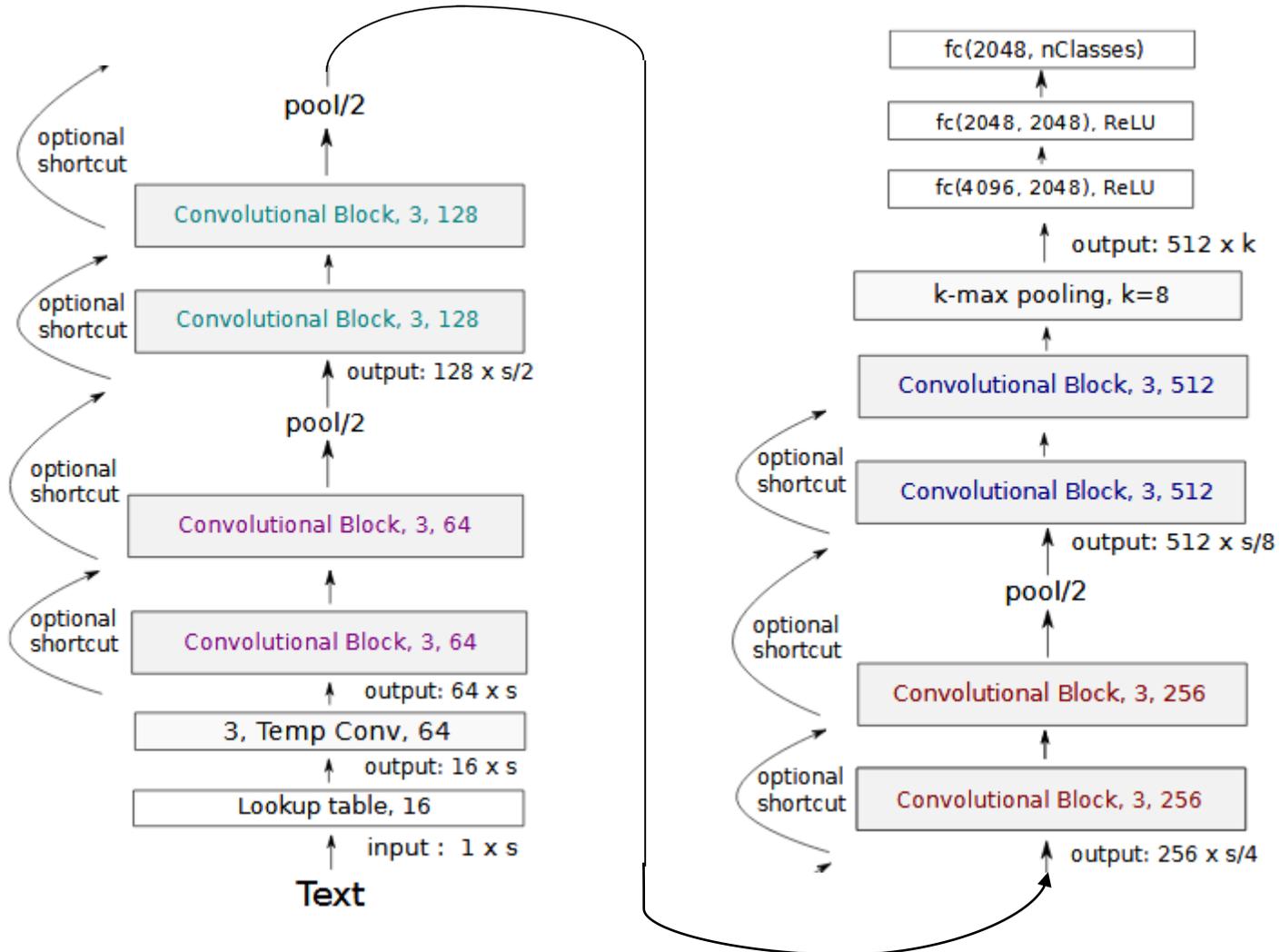
Was in der Bilderkennung funktioniert...

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).

Was in der Bilderkennung funktioniert...

Layer 1



Layer 5



Original



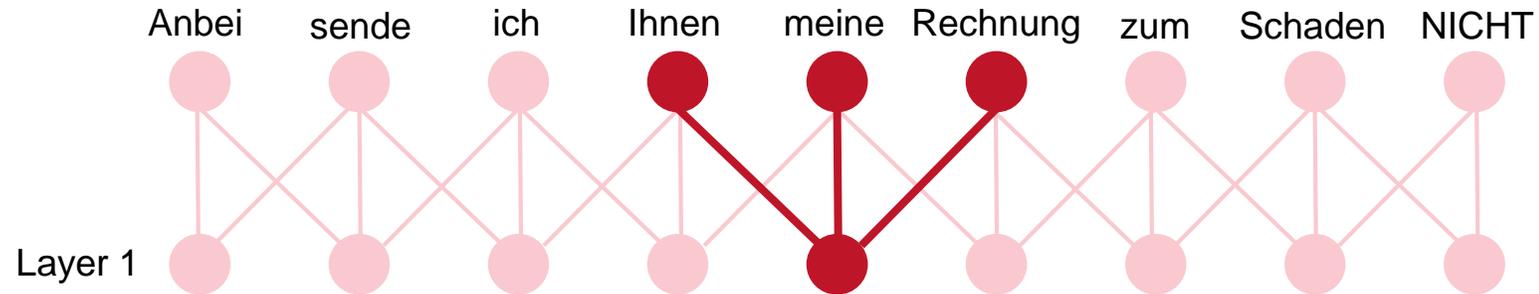
Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European conference on computer vision*. Springer, Cham, 2014.

Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

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

ERGO

A Munich Re company



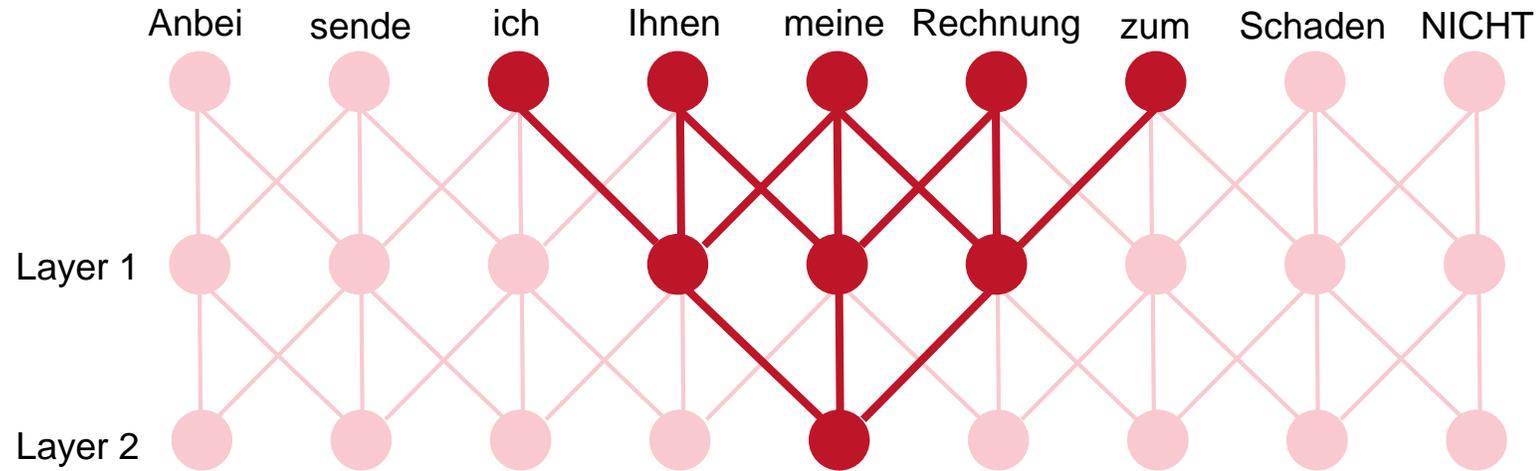
<https://www.youtube.com/watch?v=fhldbRp6xeg>

Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

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

ERGO

A Munich Re company

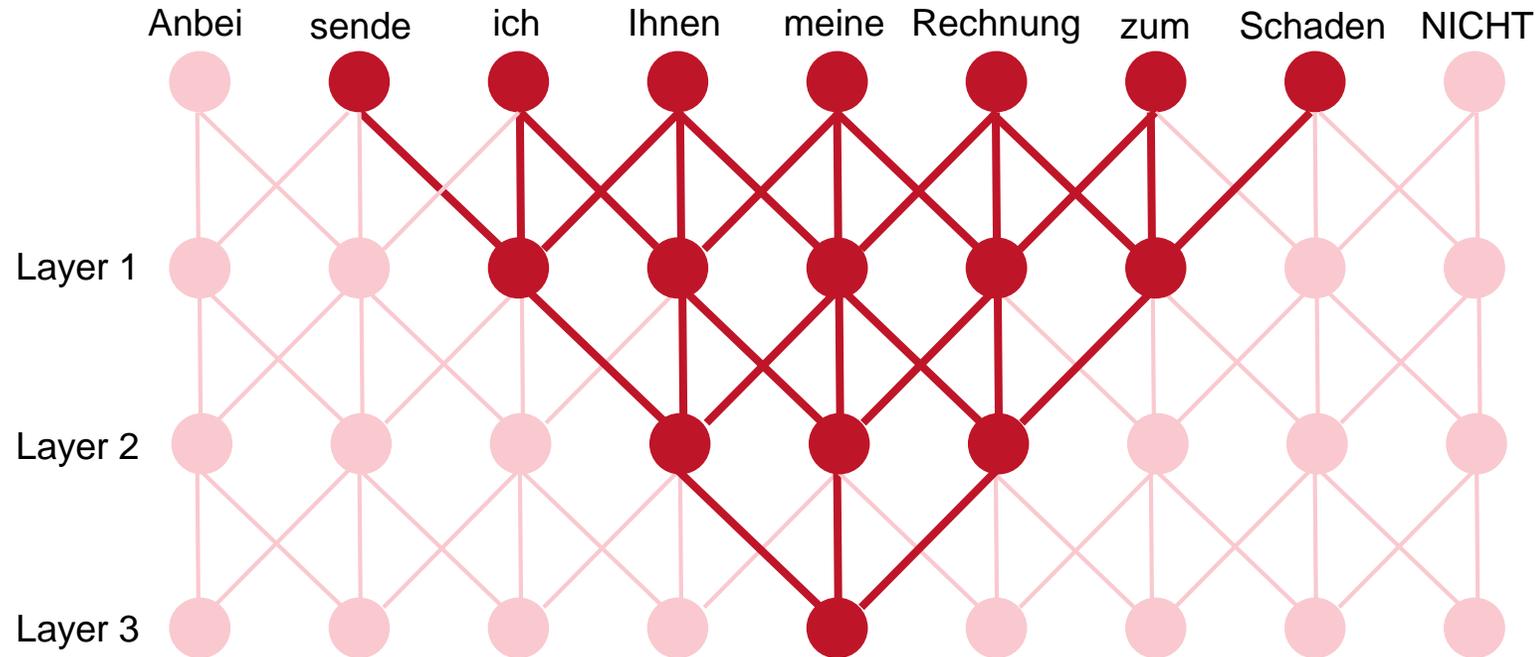


Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

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

ERGO

A Munich Re company

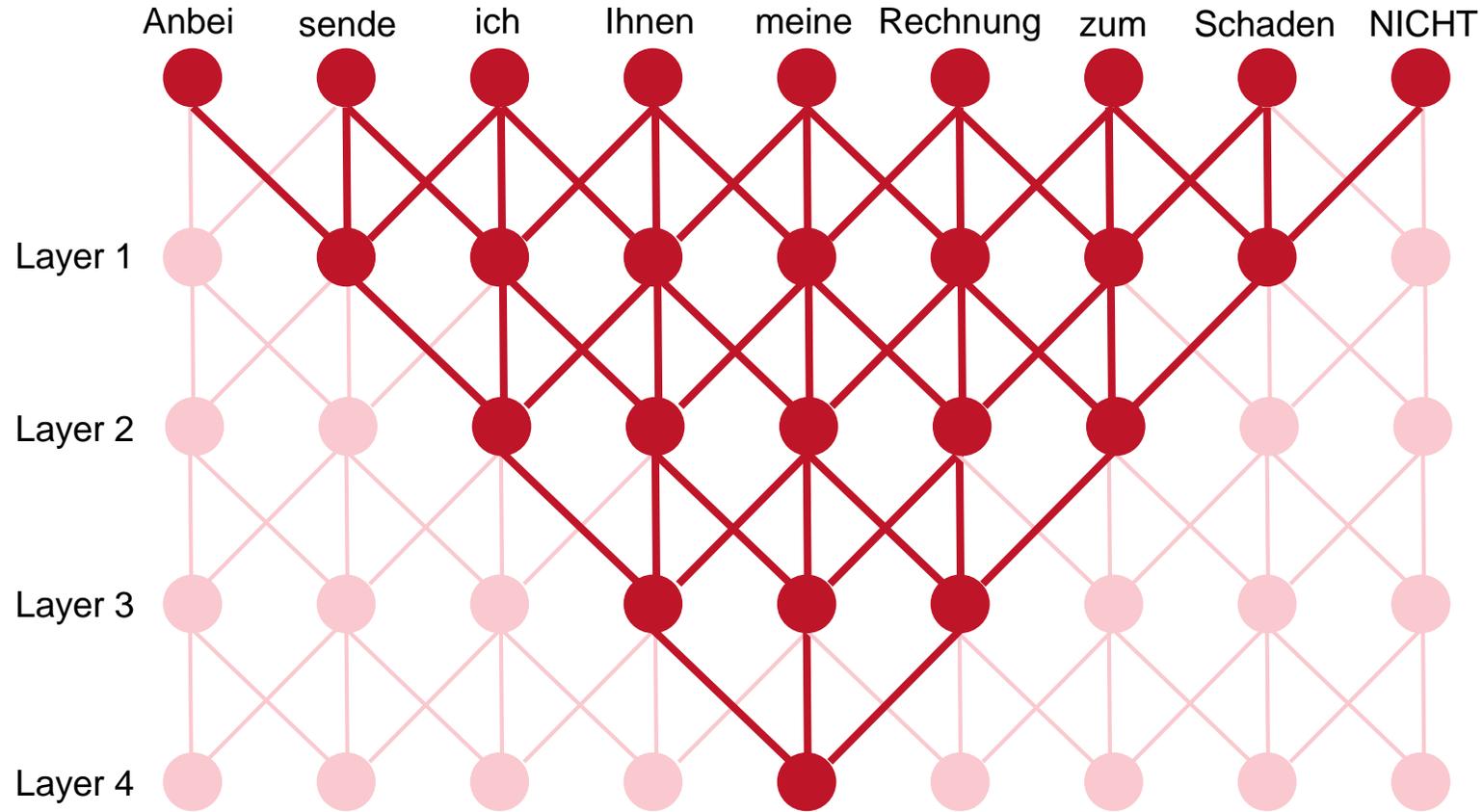


Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

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

ERGO

A Munich Re company

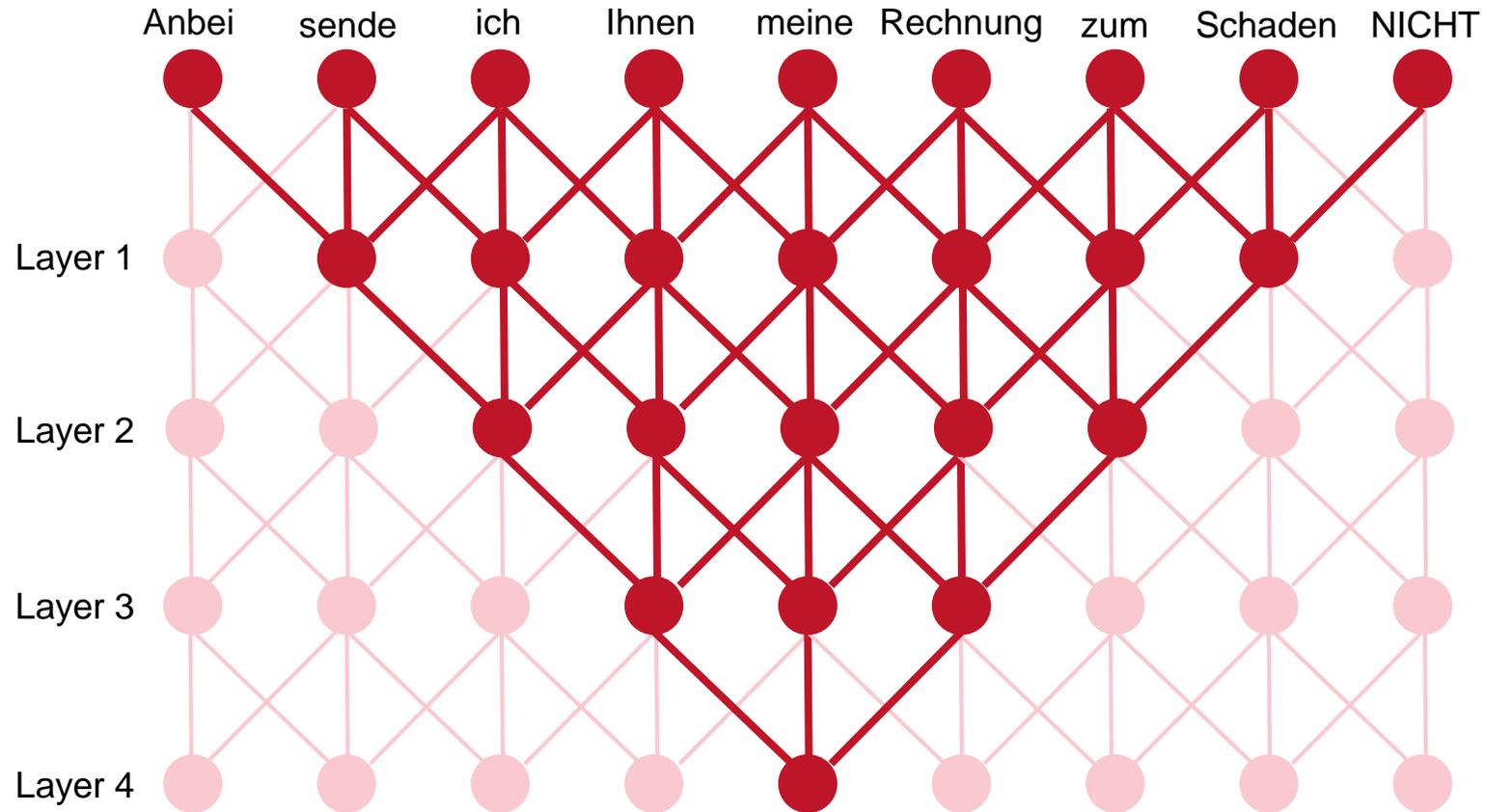


Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

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

ERGO

A Munich Re company



**CNN mit 3
Convolutional Layer**



96,74% Accuracy

Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

Übersicht Modellperformance

ERGO

A Munich Re company

	Accuracy auf Test-Daten
Anzahl Dokumente	5.794

Übersicht Modellperformance

		Accuracy auf Test-Daten
Modell berücksichtigt...	Anzahl Dokumente	5.794
...Reihenfolge	Bag of Words (1-gram)	91,37%
der Tokens NICHT	SWEM AVG	93,90%

Übersicht Modellperformance

Modell berücksichtigt...	Anzahl Dokumente	Accuracy auf Test-Daten
...Reihenfolge der Tokens NICHT	Bag of Words (1-gram) SWEM AVG	5.794 91,37% 93,90%
..."lokale" Reihenfolge der Tokens ("short range dependencies")	Bag of Words (2-gram) SWEM HIER CNN (shallow, 1conv-layer)	- 96,32% 96,19%

Übersicht Modellperformance

Modell berücksichtigt...	Anzahl Dokumente	Accuracy auf Test-Daten
	5.794	
...Reihenfolge der Tokens NICHT	Bag of Words (1-gram)	91,37%
	SWEM AVG	93,90%
..."lokale" Reihenfolge der Tokens ("short range dependencies")	Bag of Words (2-gram)	-
	SWEM HIER	96,32%
	CNN (shallow, 1conv-layer)	96,19%
..."globale" Reihenfolge der Tokens ("long range dependencies")	CNN ("deep", 3conv-layers)	96,74%
	BiLSTM-Attention	96,63%
	CNN-BiGRU-Attention	96,76%

Übersicht Modellperformance

Modell berücksichtigt...	Anzahl Dokumente	Accuracy auf Test-Daten	Trainingszeit pro Epoche*
...Reihenfolge der Tokens NICHT	Bag of Words (1-gram)	91,37%	302s
	SWEM AVG	93,90%	1s
..."lokale" Reihenfolge der Tokens ("short range dependencies")	Bag of Words (2-gram)	-	
	SWEM HIER	96,32%	
	CNN (shallow, 1conv-layer)	96,19%	
..."globale" Reihenfolge der Tokens ("long range dependencies")	CNN ("deep", 3conv-layers)	96,74%	
	BiLSTM-Attention	96,63%	
	CNN-BiGRU-Attention	96,76%	

* auf einer Nvidia Tesla V100 Grafikkarte

Übersicht Modellperformance

Modell berücksichtigt...	Anzahl Dokumente	Accuracy auf Test-Daten	Trainingszeit pro Epoche
...Reihenfolge der Tokens NICHT	5.794	596.872	
	Bag of Words (1-gram)	91,37%	302s
	SWEM AVG	93,90%	1s
..."lokale" Reihenfolge der Tokens ("short range dependencies")	Bag of Words (2-gram)	-	7.5h
	SWEM HIER	96,32%	55s
	CNN (shallow, 1conv-layer)	96,19%	73s
..."globale" Reihenfolge der Tokens ("long range dependencies")	CNN ("deep", 3conv-layers)	96,74%	
	BiLSTM-Attention	96,63%	
	CNN-BiGRU-Attention	96,76%	

Übersicht Modellperformance

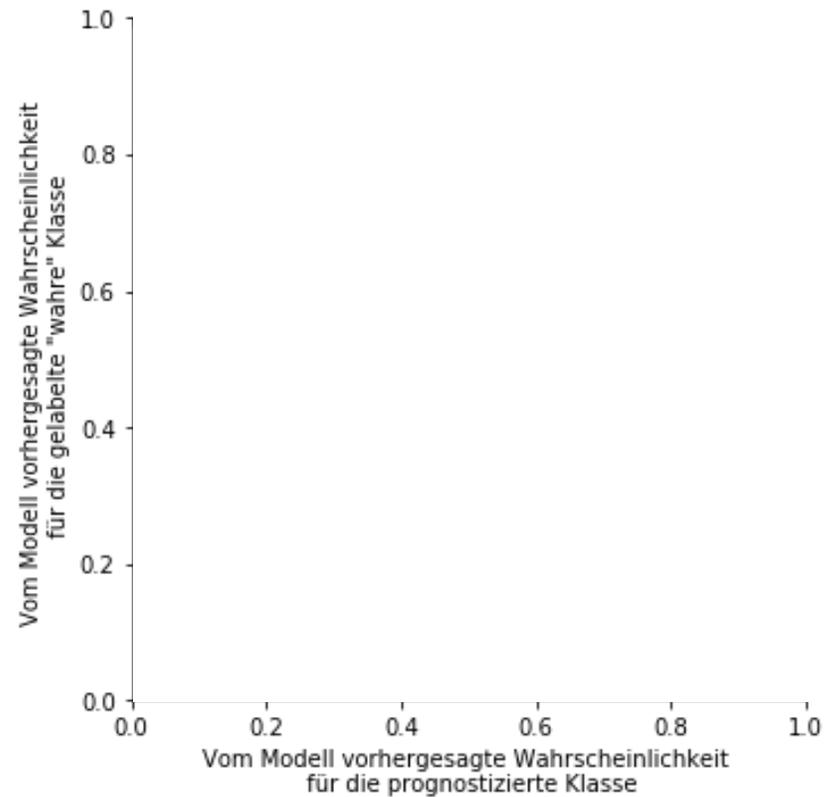
Modell berücksichtigt...	Anzahl Dokumente	Accuracy auf Test-Daten	Trainingszeit pro Epoche
...Reihenfolge der Tokens NICHT	5.794	596.872	
	Bag of Words (1-gram)	91,37%	302s
	SWEM AVG	93,90%	1s
..."lokale" Reihenfolge der Tokens ("short range dependencies")	Bag of Words (2-gram)	-	7.5h
	SWEM HIER	96,32%	55s
	CNN (shallow, 1conv-layer)	96,19%	73s
..."globale" Reihenfolge der Tokens ("long range dependencies")	CNN ("deep", 3conv-layers)	96,74%	35s
	BiLSTM-Attention	96,63%	350s
	CNN-BiGRU-Attention	96,76%	703s

Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

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

ERGO

A Munich Re company

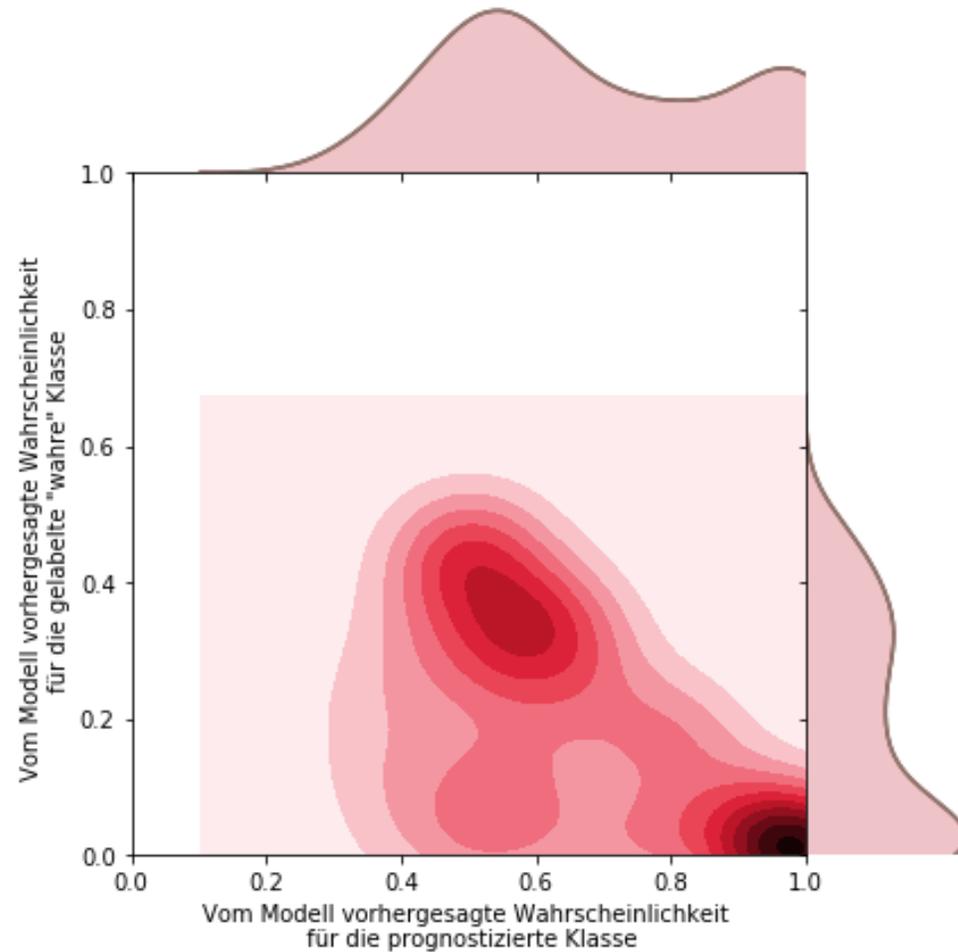


Klassifizierung von eingehenden Schriftstücken in die bestehenden Eingangsklassen

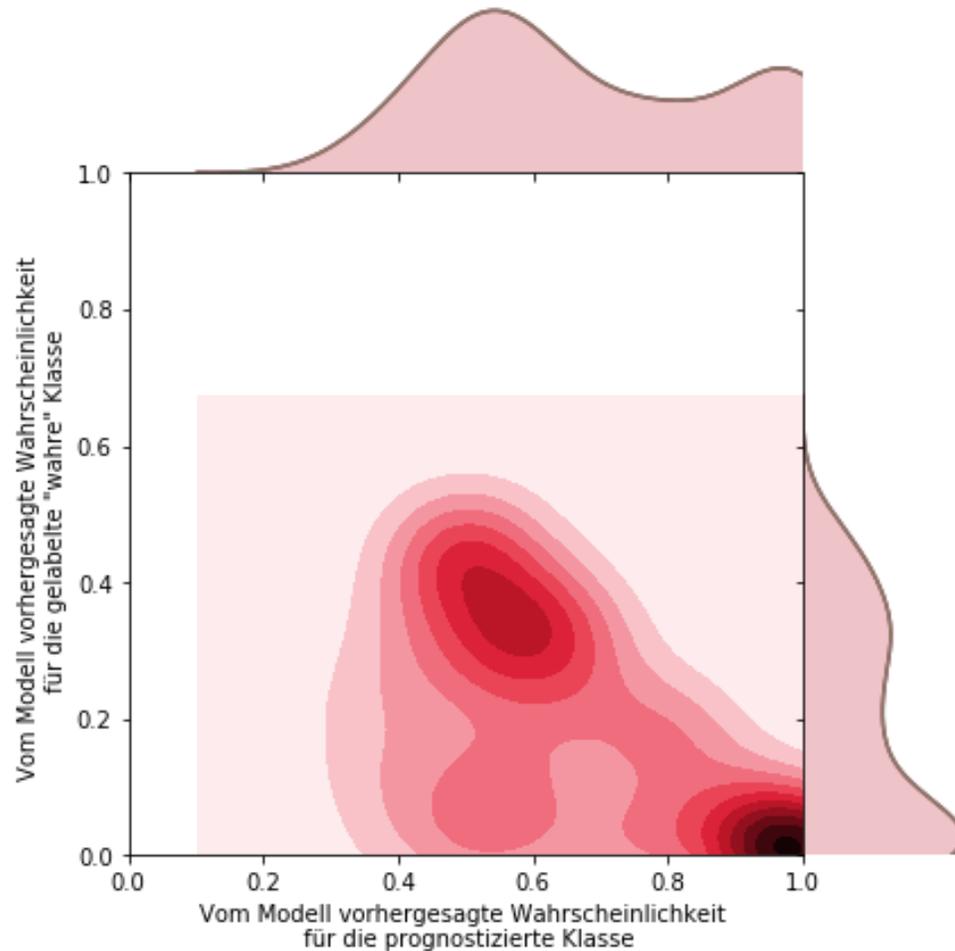
Modellperformance & Fehleranalyse (2018 & 2019, Test-Sample)

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**

ERGO

A Munich Re company

**Vielen Dank
für Ihre Aufmerksamkeit!**

sebastian.mikus@ergo.de

lucas.hafner@ergo.de

