

# Neural Networks: Types of convolutional layers and their application

Lukas Schmauch

Business Decision Support Techniques  
Prof. Dr. Johannes Ruhland, Sven Gehrke

Juni 2020

# Übersicht

Was ist ein CNN?

Warum CNNs statt klassischer Neuronaler Netze?

Architektur eines CNNs

Faltung (Convolution)

Kantenfilter

Zusammenfassung

# Was ist ein CNN?

- ▶ CNN - Convolutional Neural Network (**Faltungsnetzwerk**)
- ▶ nutzen **Faltung (engl. convolutional)** als spezielle lineare Transformation
- ▶ dienen primär zur **Klassifikation** gitterartig aufgebauter Daten (z.B. Bilder - 2D bzw. 3D Anordnung von Pixelwerten)
- ▶ **Besonderheit: geteilte Gewichte (weight-sharing)** und Invarianz gegenüber Verschiebungen der Eingabe (**Translationsinvarianz**)
- ▶ heutiger Erfolg: **moderne Hardware (GPUs) + Big Data**

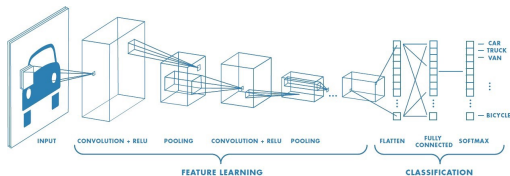


Figure: Architektur eines Convolutional Neural Networks [4]

# Warum CNNs statt klassischer Neuronaler Netze?

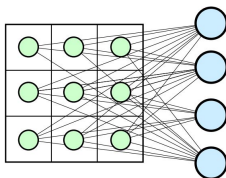


Figure: Ausschnitt erste Schicht eines MLPs mit Bilddaten [3]

- ▶ Annahme: Eingabe sei Grauwertbild mit **1 Megapixel** ( $1024 \times 1024 \times 1$ ), ein verborgener Layer mit **k Knoten**
- ▶ Anzahl der **zu schätzenden Parameter** der ersten Schicht:  
 $1024 \cdot 1024 \cdot 1 \cdot k \approx k$  **Millionen Parameter (Overfitting!!!)**
- ▶ **Herausforderung CNN:** drastische Reduktion der Parameter bei gleichzeitiger Invarianz ggü. Verschiebungen

# Architektur eines CNNs

- **Detection Stage:** Convolutional Layer + ReLU, Pooling Layer
- **Classification Stage:** Fully Connected Layer

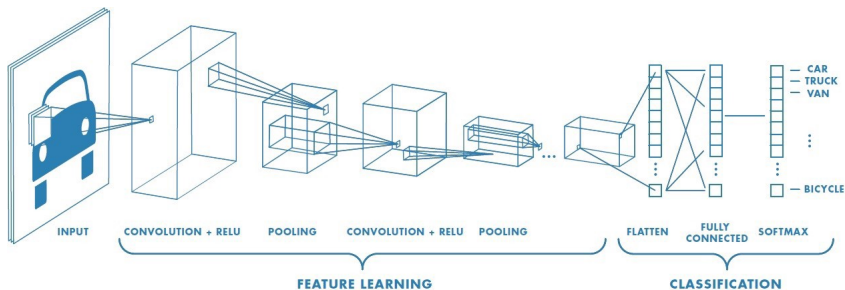
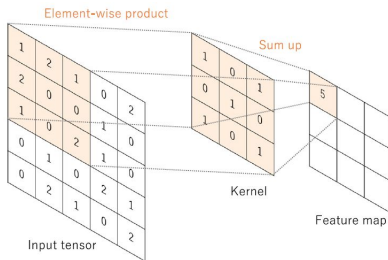


Figure: Architektur eines Convolutional Neural Networks [4]

# Faltungsoption (Convolution)

- **Faltung** mit einer **3x3 Faltungsmatrix** als elementweise Multiplikation und Addition (**Skalarprodukt**)
- Filter **gleitet schrittweise** über komplettes Eingabebild
- Resultierender **Pixelwert nach Maskenoperation** im gelben Bereich:  
 $(1 \cdot 1 + 2 \cdot 0 + 1 \cdot 1 + 2 \cdot 0 + 0 \cdot 1 + 0 \cdot 0 + 1 \cdot 1 + 0 \cdot 0 + 2 \cdot 1 = 5)$



**Figure:** Anwendung eines Filters bzw. Kernels auf den Input-Tensor und entstehende Feature Map [2]

# Einfache Kantenfilter

<table><tr><td>-1</td><td>-1</td><td>-1</td></tr><tr><td>2</td><td>2</td><td>2</td></tr><tr><td>-1</td><td>-1</td><td>-1</td></tr></table> Horizontal lines	-1	-1	-1	2	2	2	-1	-1	-1	<table><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr></table> Vertical lines	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1																	
2	2	2																	
-1	-1	-1																	
-1	2	-1																	
-1	2	-1																	
-1	2	-1																	
<table><tr><td>-1</td><td>-1</td><td>2</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>2</td><td>-1</td><td>-1</td></tr></table> 45 degree lines	-1	-1	2	-1	2	-1	2	-1	-1	<table><tr><td>2</td><td>-1</td><td>-1</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>-1</td><td>-1</td><td>2</td></tr></table> 135 degree lines	2	-1	-1	-1	2	-1	-1	-1	2
-1	-1	2																	
-1	2	-1																	
2	-1	-1																	
2	-1	-1																	
-1	2	-1																	
-1	-1	2																	

Figure: Verschiedene Kernelgewichte, zur Extraktion der jeweiligen Kantenformen [5]



Figure: Anwendung eines horizontalen Katendetektors [5]

- **bekannte Filter:** Sobel-Operator, Laplace-Filter, Scharr-Operator, Prewitt-Operator, Roberts-Operator, Kirsch-Operator,...

# Zusammenfassung

1. CNNs (Faltungsnetzwerke) sind spezielle Neuronale Netze mit **mindestens einer Faltungsoperation**, statt der generellen Matrizenmultiplikation.
2. Seit der Entwicklung **moderner Hardware (GPUs)**, sowie der **Sammlung und Etikettierung riesiger Datenmengen** sind sie erstmals effizient trainierbar.
3. CNNs werden erfolgreich u.a. im Bereich **Bildklassifikation**, sowie der **Handschrift- und Spracherkennung** eingesetzt.
4. Convolutional und Pooling Layer dienen der **automatischen Extraktion von Merkmalen**.
5. Diese Komponenten können sich beliebig oft wiederholen (tiefe Netze), um **hierarchisch komplexere Merkmale** zu identifizieren.
6. **Filter gleiten schrittweise** über ein **Eingabebild**, um **Feature Maps** mit den hervorgehobenen Mustern zu berechnen.
7. Die **Gewichte eines Filters** werden in der Trainingsphase **automatisch gelernt (Backpropagation)** und müssen nicht händisch bestimmt werden.
8. Die Besonderheit von CNNs sind die **geteilten Gewichte (weight-sharing)** eines Filters über das Eingabebild, sowie die **Invarianz gegenüber Verschiebungen der Eingabe**. (Idee: Muster können sich überall im Bild befinden)
9. Die eigentliche **Klassifikation der extrahierten Merkmale** findet in den letzten Schichten (**Fully Connected Layer**) statt.



# Literaturverzeichnis



Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, 2016



Convolutional neural networks: an overview and application in radiology, Yamashita, 2018



Script: Einführung in tiefe Lernverfahren - Faltungsnetzwerke, Prof. Joachim Denzler



[https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks\the-eli5-way-3bd2b1164a53](https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53)



<https://aishack.in/tutorials/image-convolution-examples/>