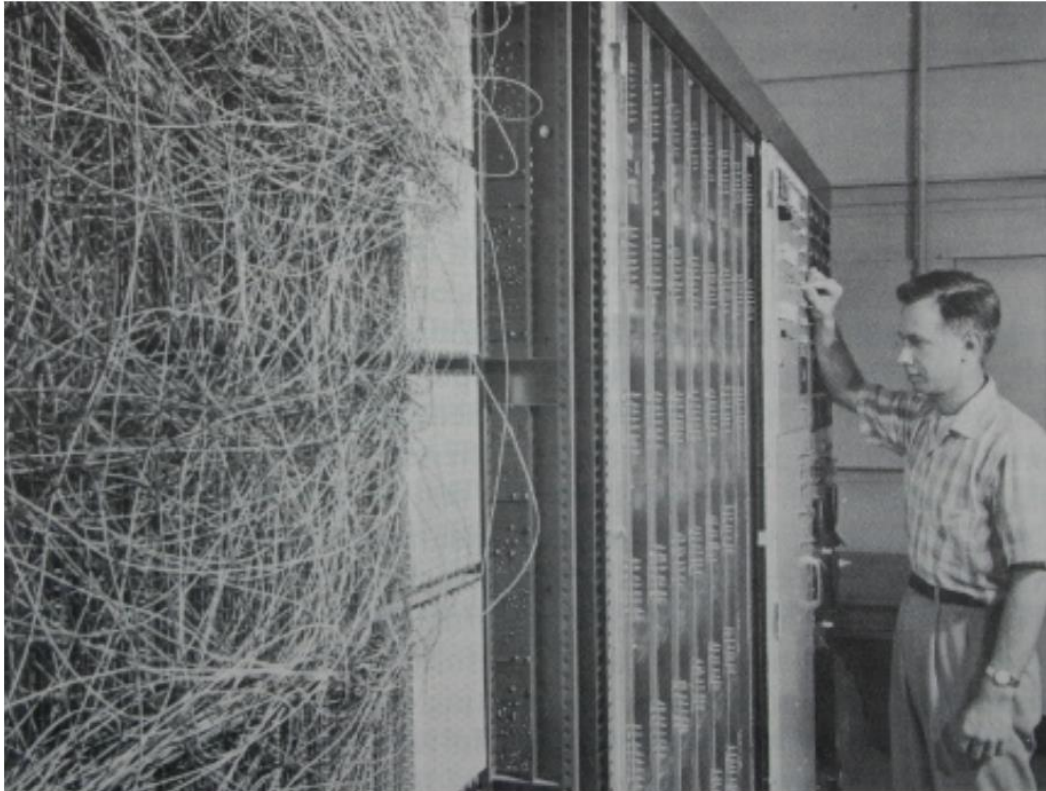


# Convolutional Neural Network

## -- Filters

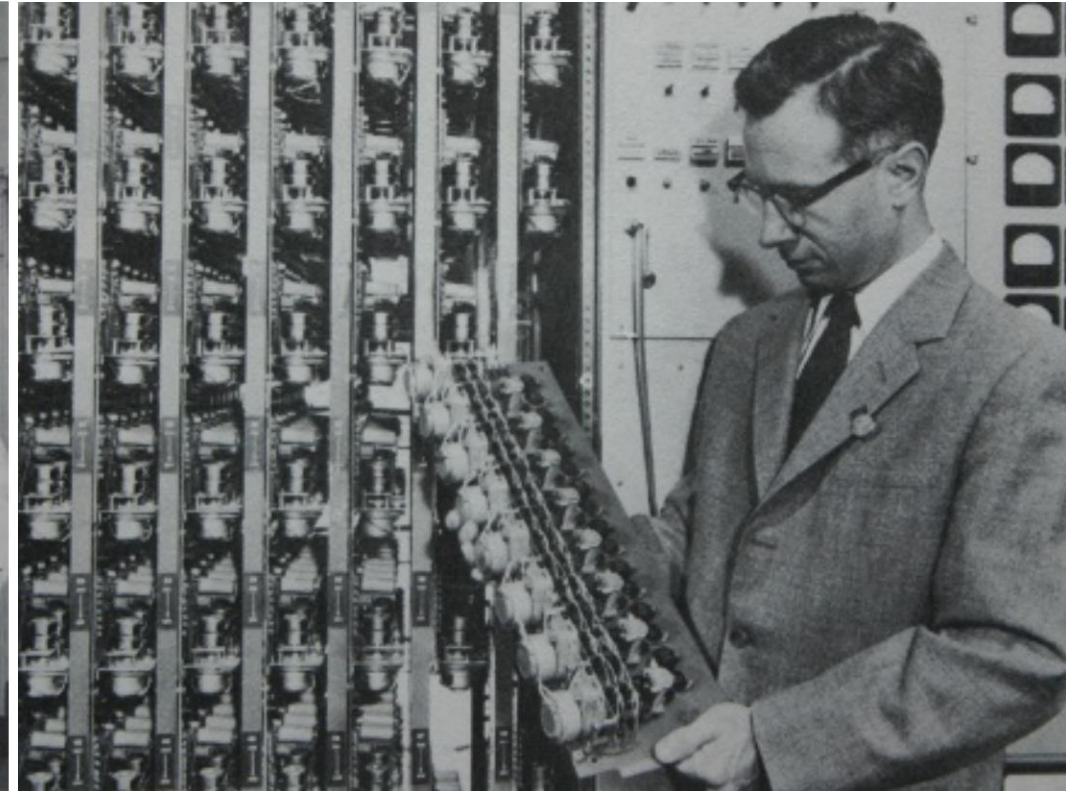
# Mark 1 Perzeptron-Hardware

Zufällige Verbindungen



„lineare Funktionen“

Potentiometer: drehbarer variabler Widerstand



„Aktivierungsfunktionen“

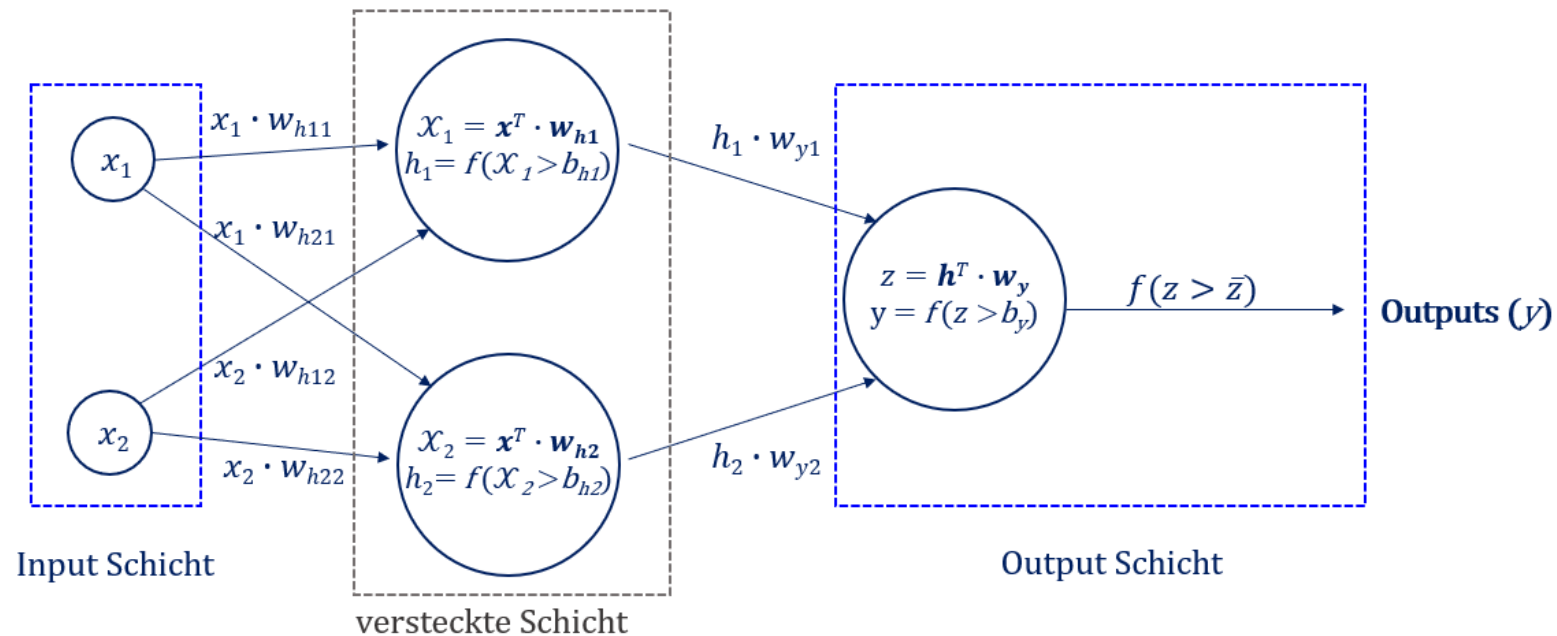
*Frank Rosenblatt, 1957*

# Perzeptron

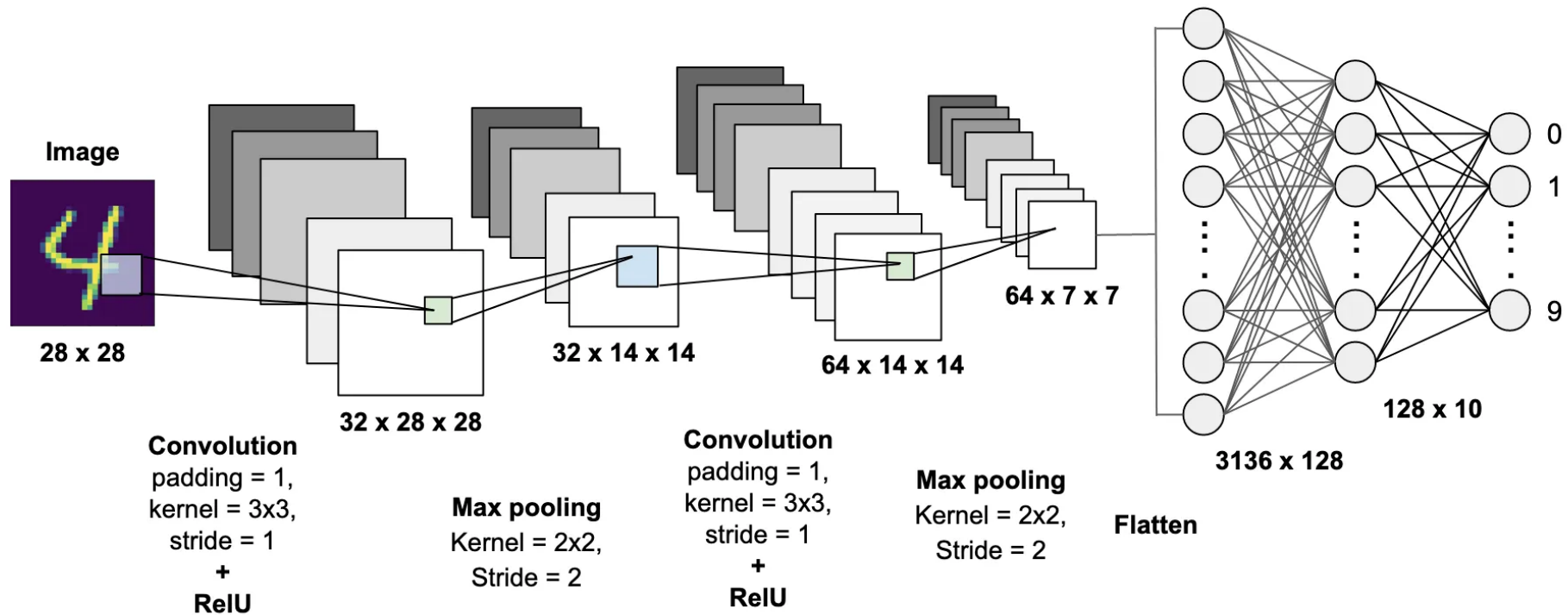
## AND, OR & XOR-Operationen

inputs		outputs		
$x_1$	$x_2$	AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Logikgatter



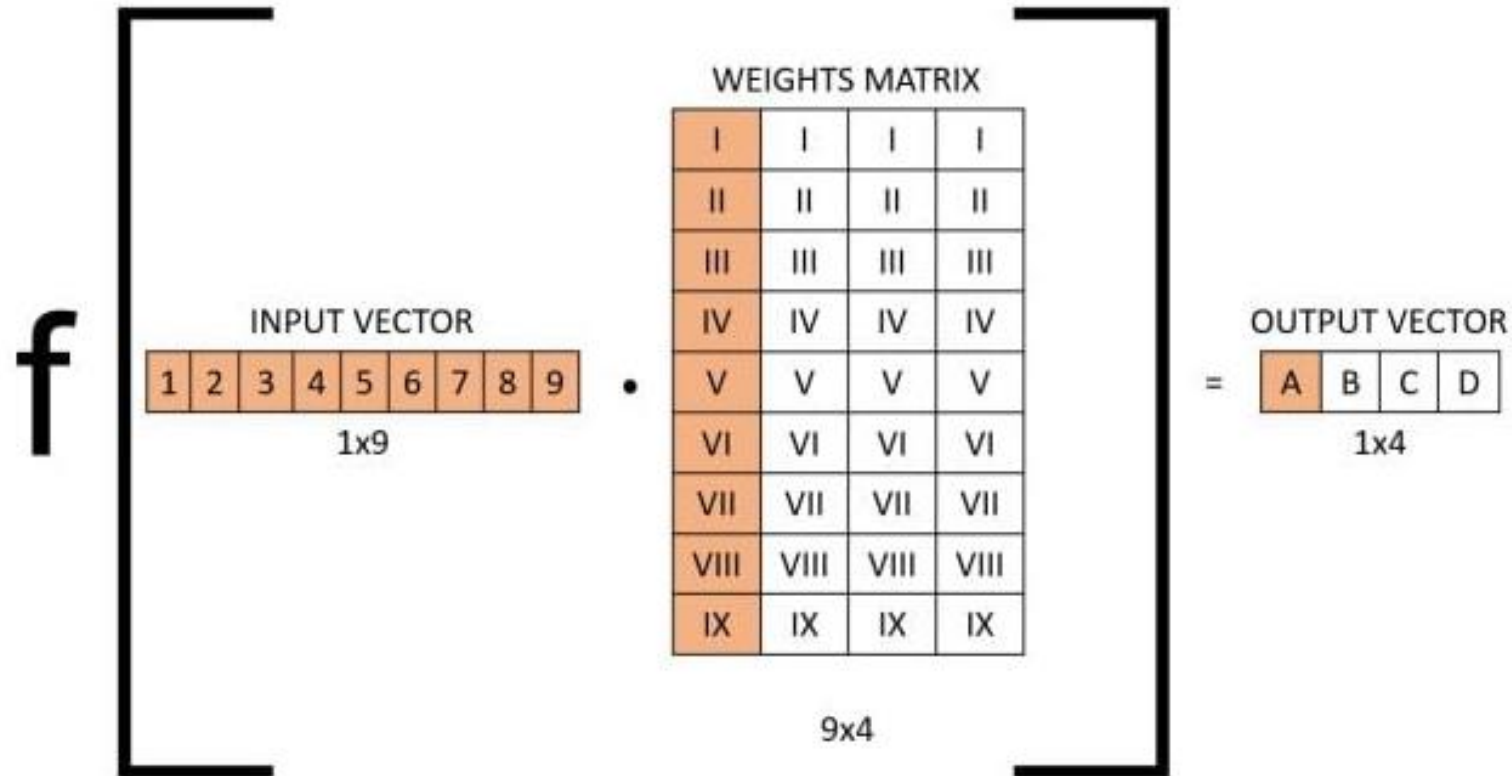
# Convolutional Neural Network



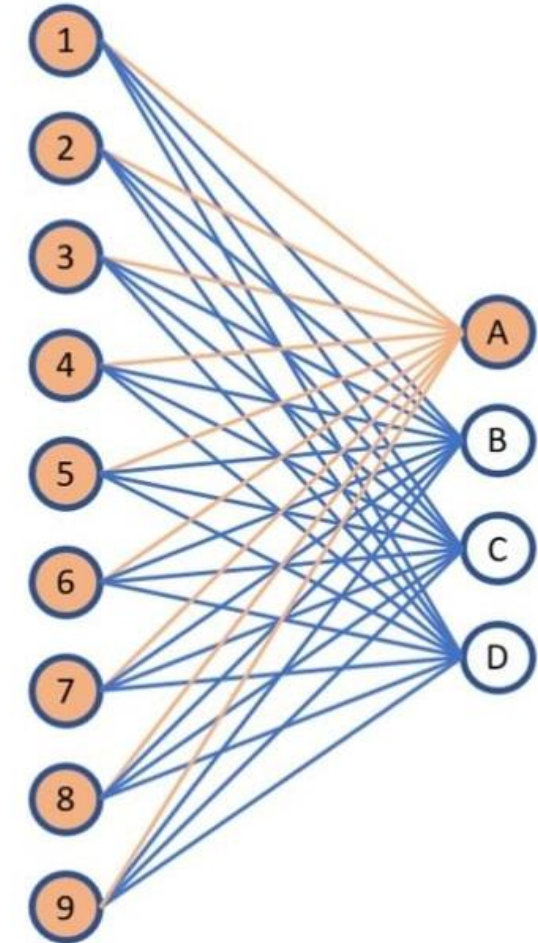
faltendes neuronales Netzwerk

Rumelhart, 1986

# Fully Connected Layer vs. Convolutional Layer



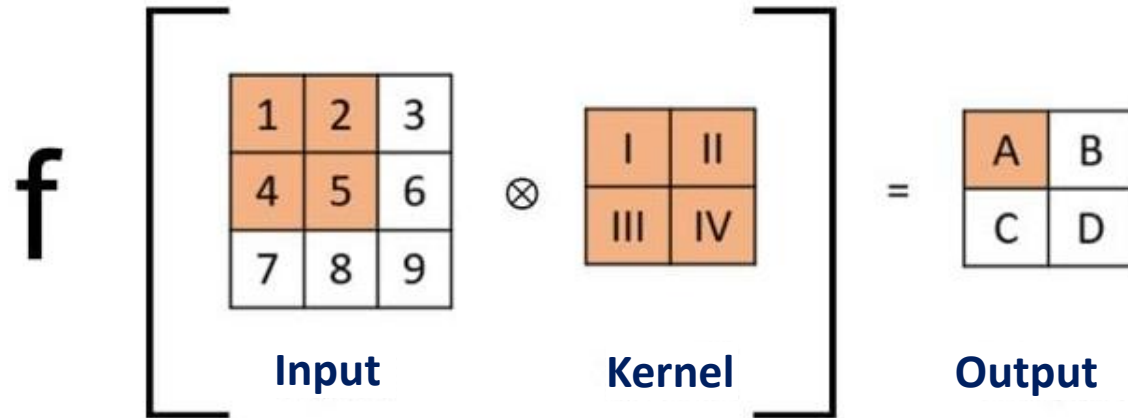
die Eingabe der vollverknüpften Schicht, multipliziert mit der Gewichtungsmatrix, um den Ausgabevektor zu erhalten.



eine vollständig verknüpfte Schicht (a fully connected layer)

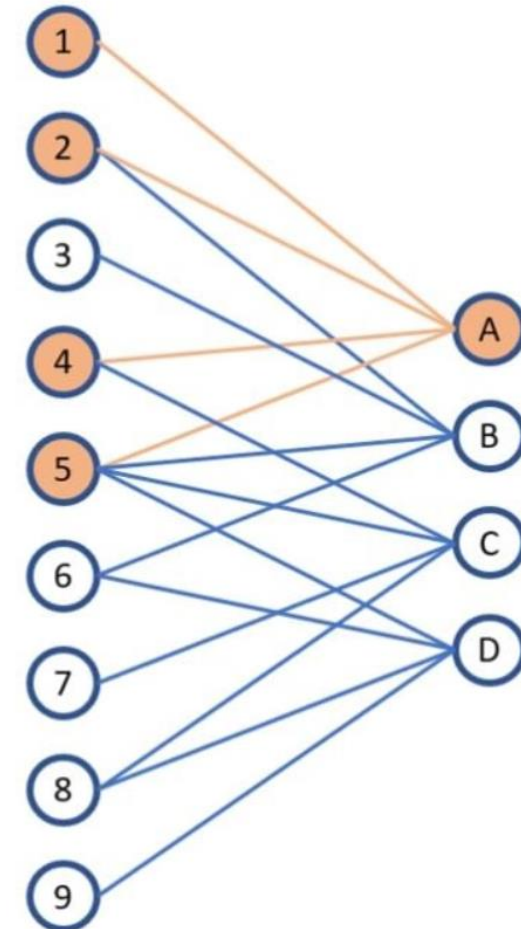
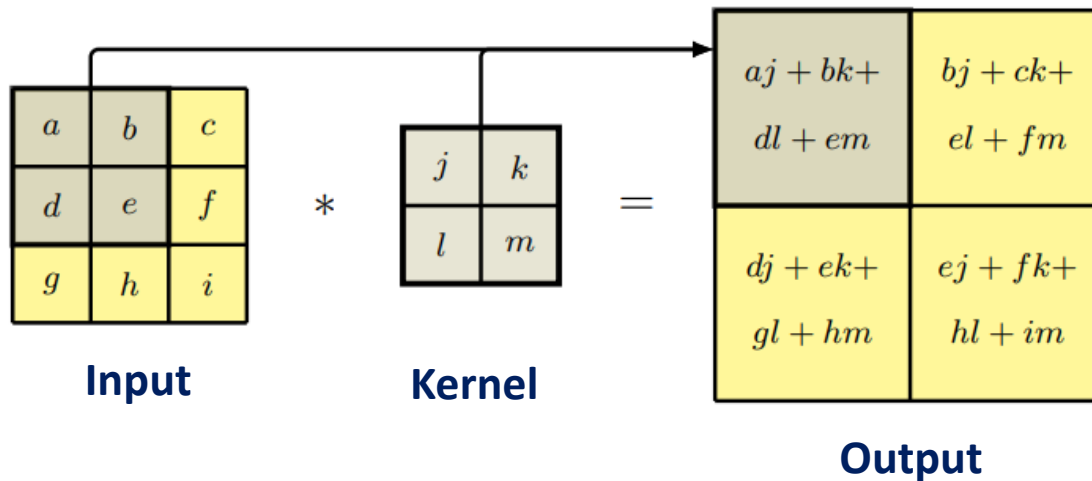


# Fully Connected Layer vs. Convolutional Layer



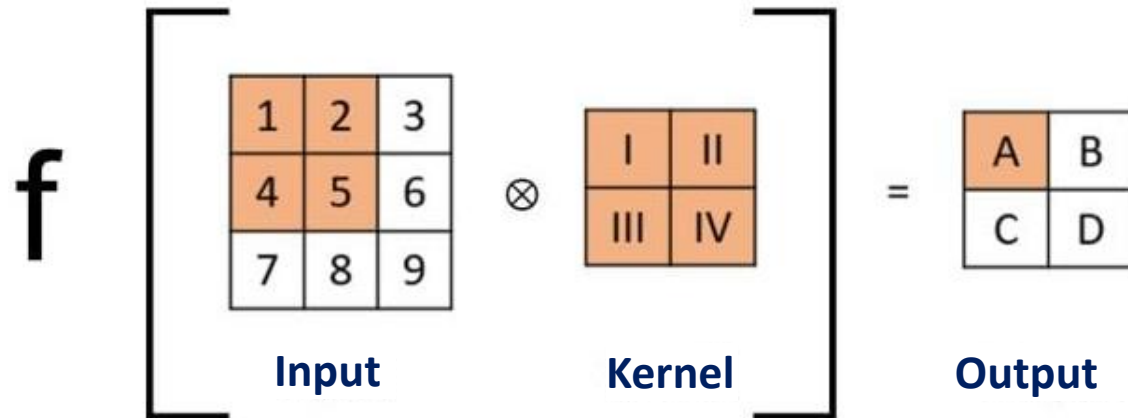
$$A = 1 * I + 2 * II + 4 * III + 5 * IV$$

Verfahren der Faltungsschicht

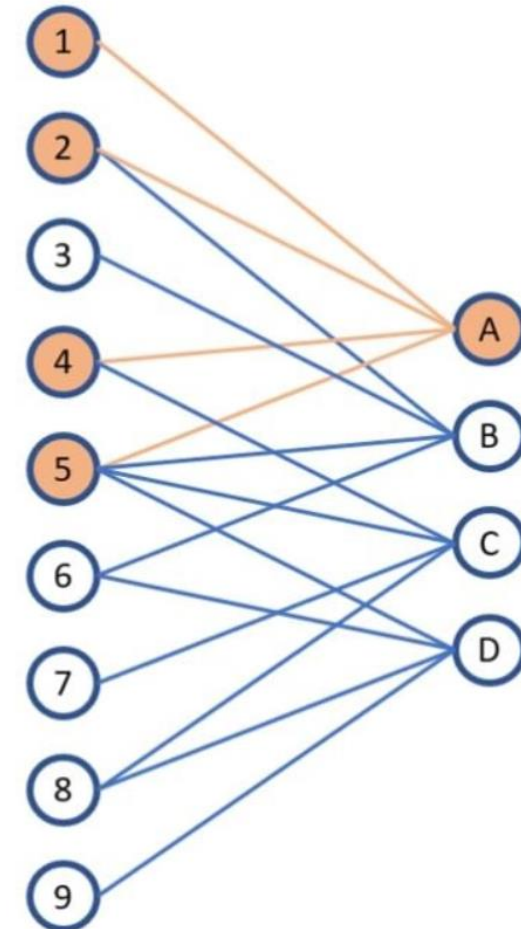


Die Art und Weise, wie die Knoten in einer Faltungsschicht verbunden sind.

# Fully Connected Layer vs. **Convolutional Layer**



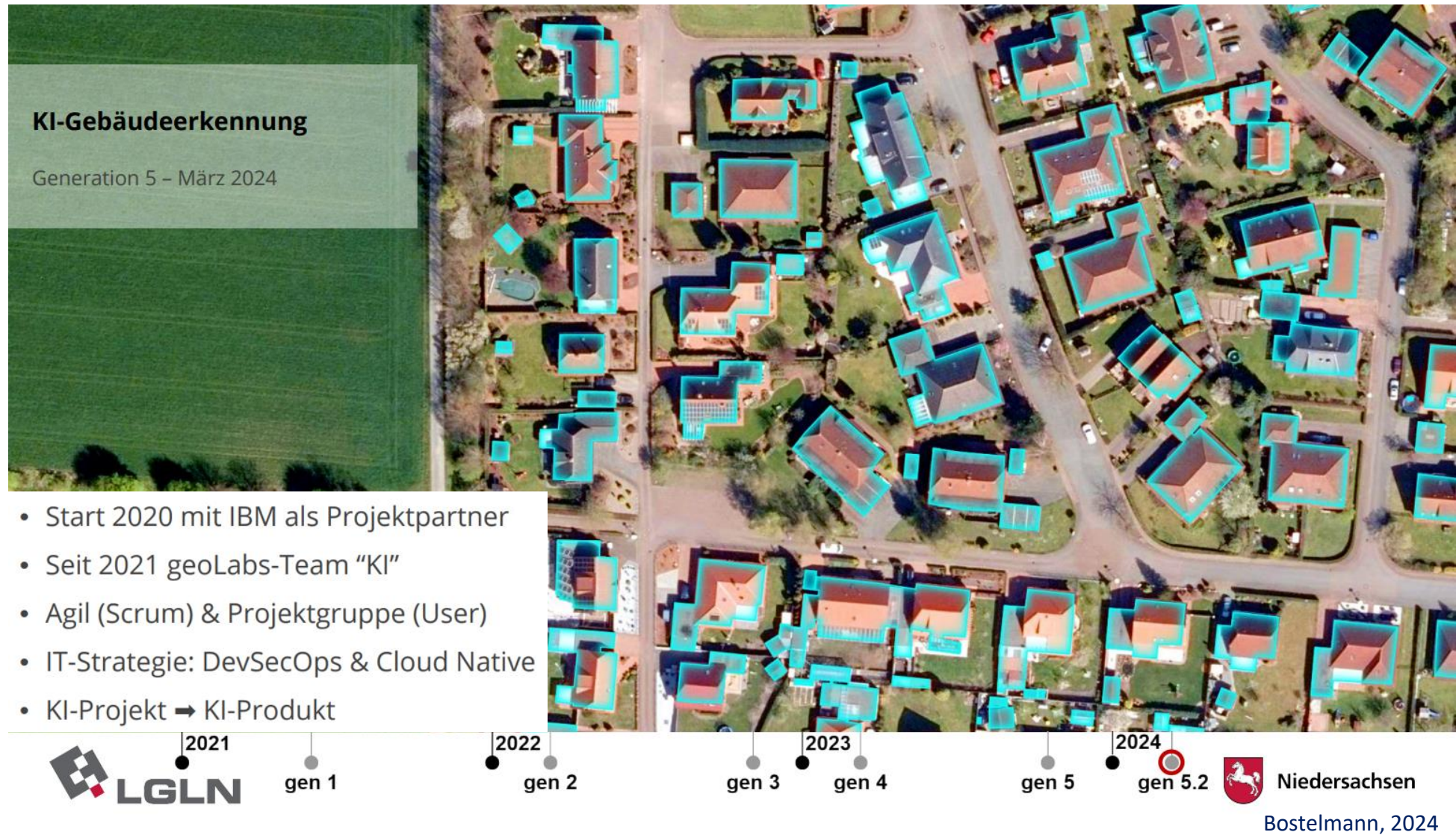
$A = 1 * I + 2 * II + 4 * III + 5 * IV$   
Verfahren der Faltungsschicht



Die Art und Weise, wie die Knoten in einer Faltungsschicht verbunden sind.



# Luftbilder





# Luftbilder

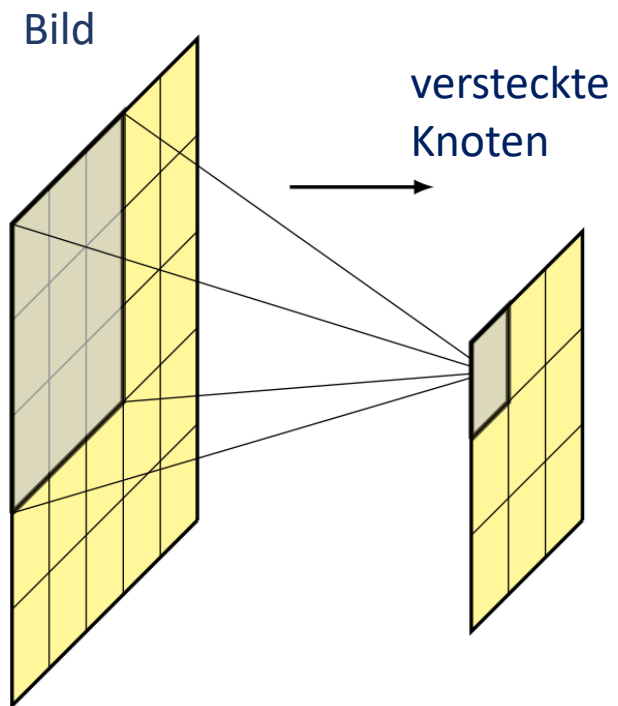
- ein Farbbild mit  $10^3 \times 10^3$  Pixeln, von denen jedes drei Werte hat, die den Intensitäten von Rot, Grün und Blau entsprechen.
  - Die erste versteckte Schicht des Netzes hat 1.000 versteckte Knoten, dann haben wir bereits  $3 \times 10^9$  Gewichte in der ersten Schicht.
- 
- Luftbild (DOP20, 1km \* 1km) mit  $5 \cdot 10^3 \times 5 \cdot 10^3$  Pixeln, von denen jedes drei Werte hat, die den Intensitäten von Rot, Grün und Blau entsprechen.
  - Die erste versteckte Schicht des Netzes hat 1.000 versteckte Knoten, dann haben wir bereits  $7,5 \times 10^{10}$  Gewichte in der ersten Schicht.

Digitale Orthophotos Bodenauflösung 20 cm (DOP20): mit einer Bodenauflösung von 20 cm (1 Bildpixel = 20 cm x 20 cm)

## Punktwolken...

# Filters in CNN

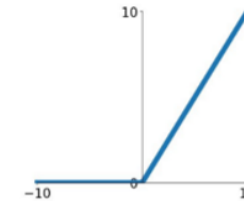
## ■ Feature-Detektoren



Grauwertbilder (d.h., Bilder mit einem einzigen Kanal)

- kleinen rechteckigen Bereich (Patch)
- den Begriff der Lokalität erfassen
- $t = \text{ReLU}(w^T x + w_0)$

**ReLU**  
 $\max(0, x)$



# Perzeptron

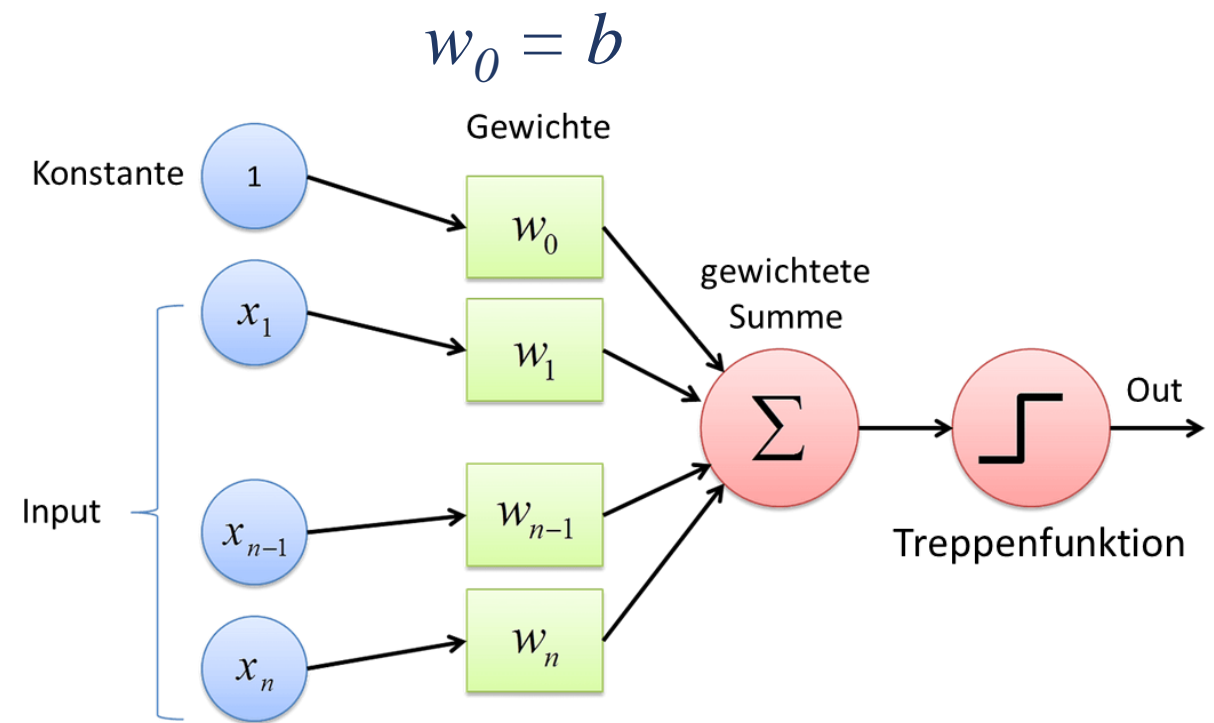
$$z = f\left(\sum_{i=1}^n w_i x_i + b\right) = f(\mathbf{w}^T \mathbf{x})$$

Dabei ist:

$$\mathbf{W} = [w_1 \ w_2 \ \dots \ w_n \ b]^T$$

$$\mathbf{X} = [x_1 \ x_2 \ \dots \ x_n \ 1]^T$$

$$f(n) = \begin{cases} 1 & n \geq 0 \\ 0 & \text{ansonsten} \end{cases}$$



**Begrenzungsline:**

$$a \cdot x + b \cdot y + c = 0 \rightarrow w_1 \cdot x_1 + w_2 \cdot x_2 + b = 0$$

**Begrenzungsfläche:**

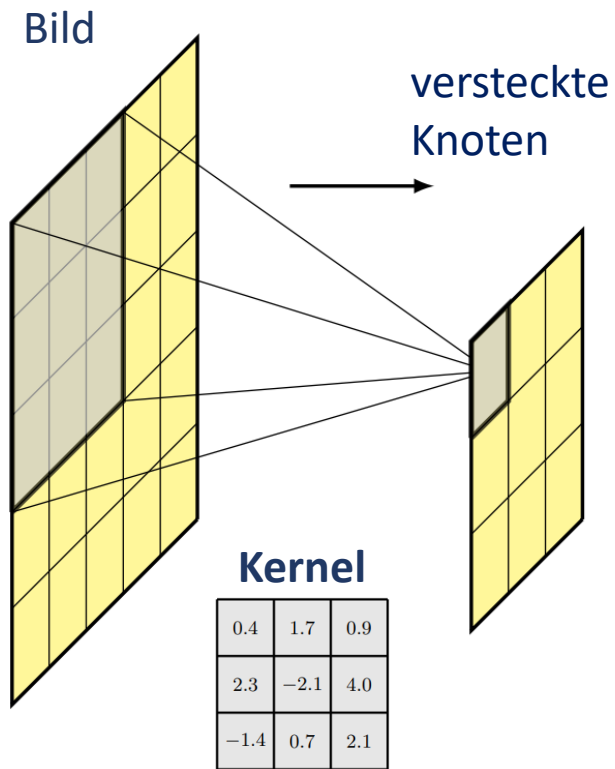
$$a \cdot x + b \cdot y + c \cdot z + d = 0 \rightarrow w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + b = 0$$

**Hyperfläche (n-1):**

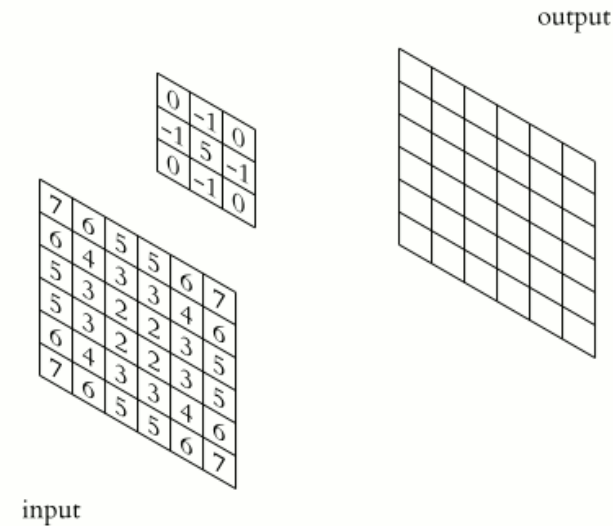
$$w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n + b = 0$$

# Filters in CNN

## Feature-Detektoren

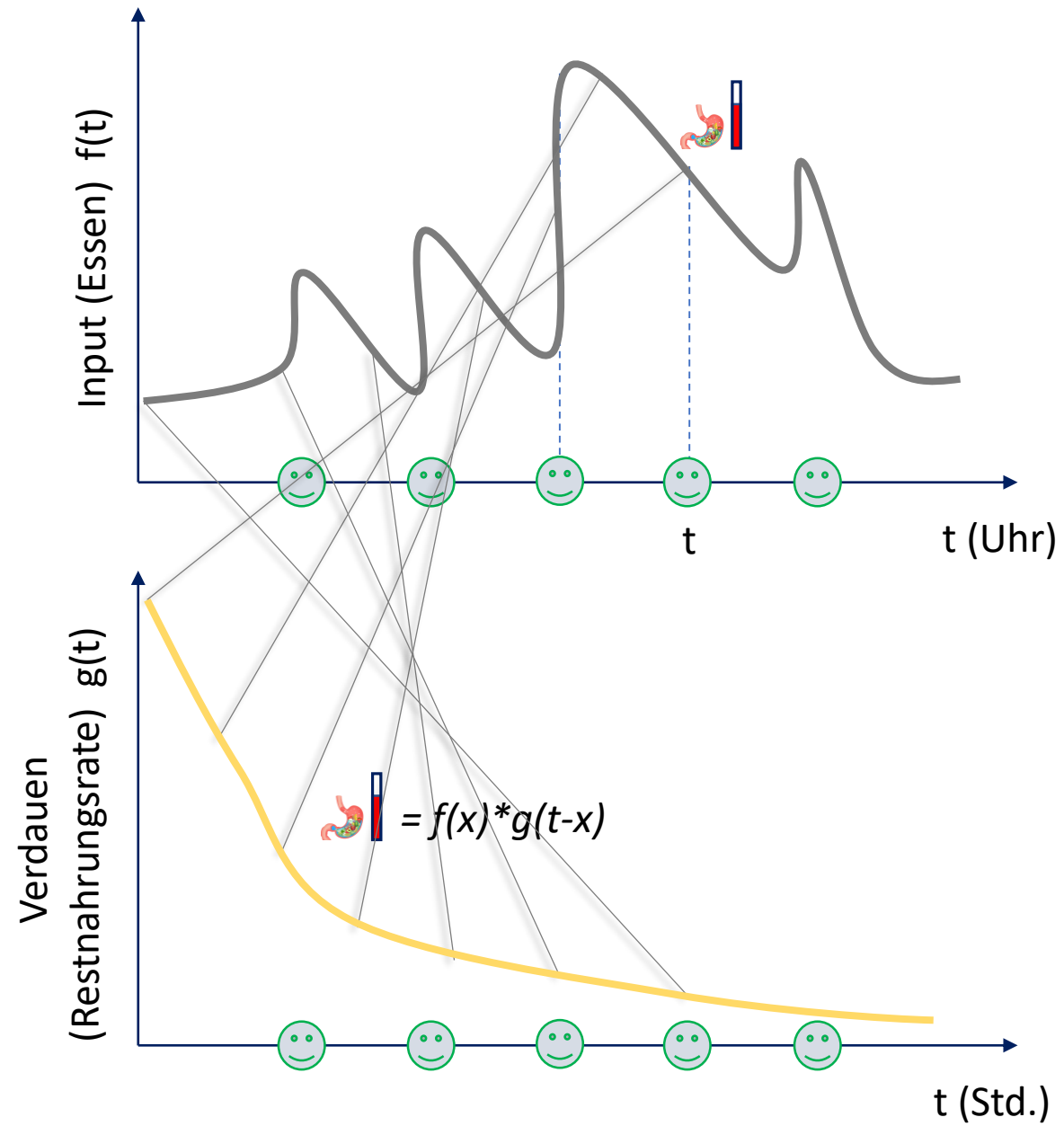


Grauwertbilder (d.h., Bilder mit einem einzigen Kanal)



- kleinen rechteckigen Bereich (Patch)
- er erfasst den Begriff der Lokalität
- $t = \text{ReLU}(w^T x + w_0)$ 
  - a gewichtete lineare Kombination der Eingangswerte
  - eine nichtlineare Aktivierungsfunktion
  - x ist ein Vektor von Pixelwerten für das rezeptive Feld
- Kernel (Filter) : Gewichte

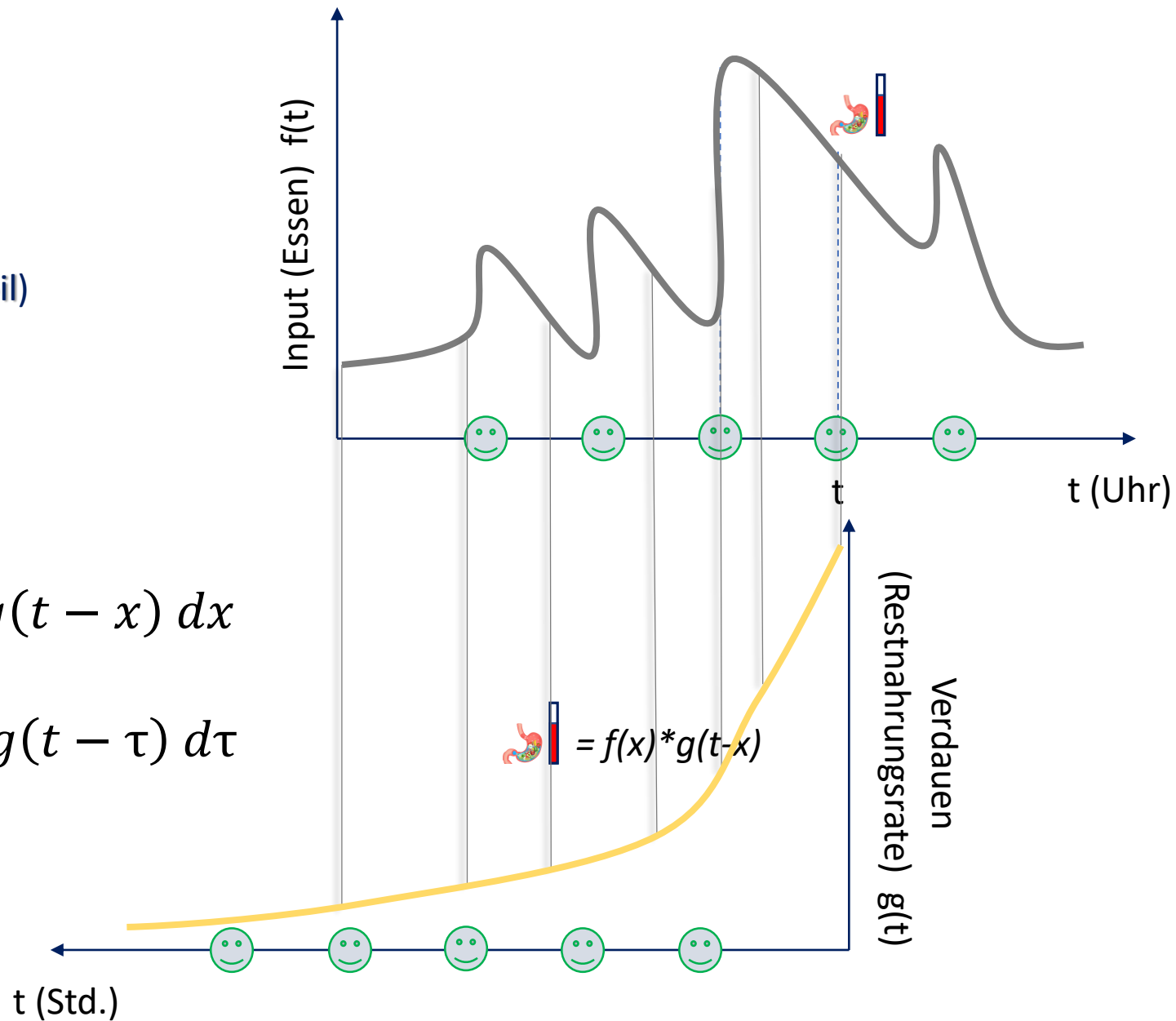




**System: Verdauen (stabil)**

$$(f * g)(t) = \int_0^t f(x) * g(t - x) dx$$

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) * g(t - \tau) d\tau$$

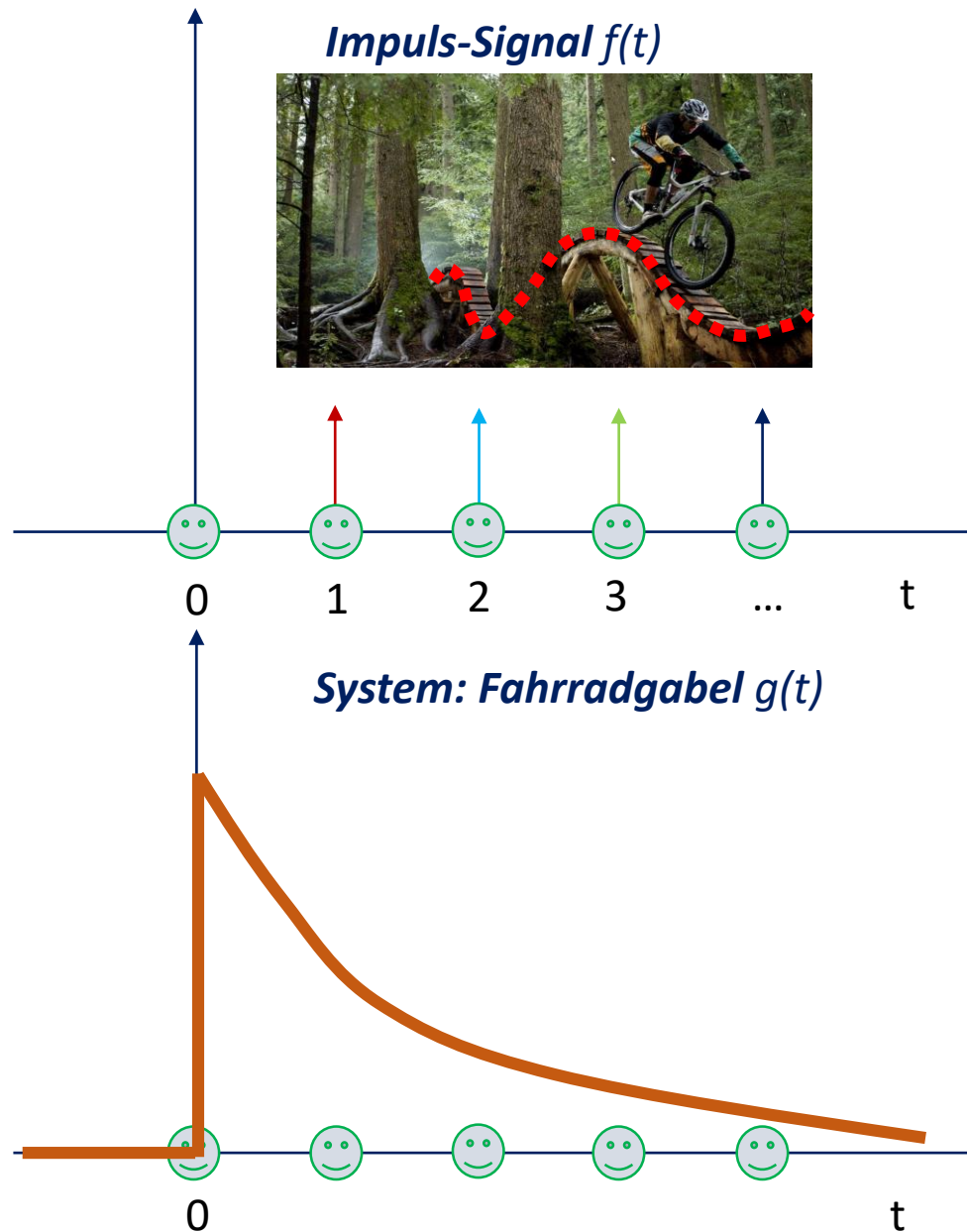


$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau.$$

**System:** Fahrradgabel



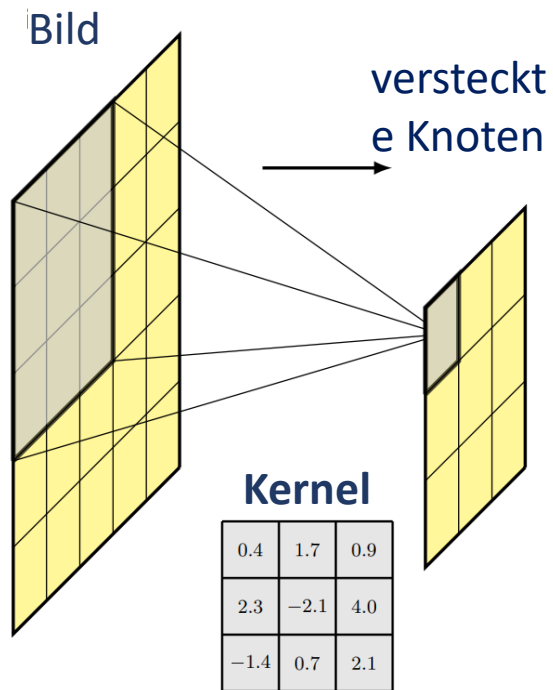
die Größe des Steins ~ Schwingungsamplitude (das Ausmaß der Verformung)



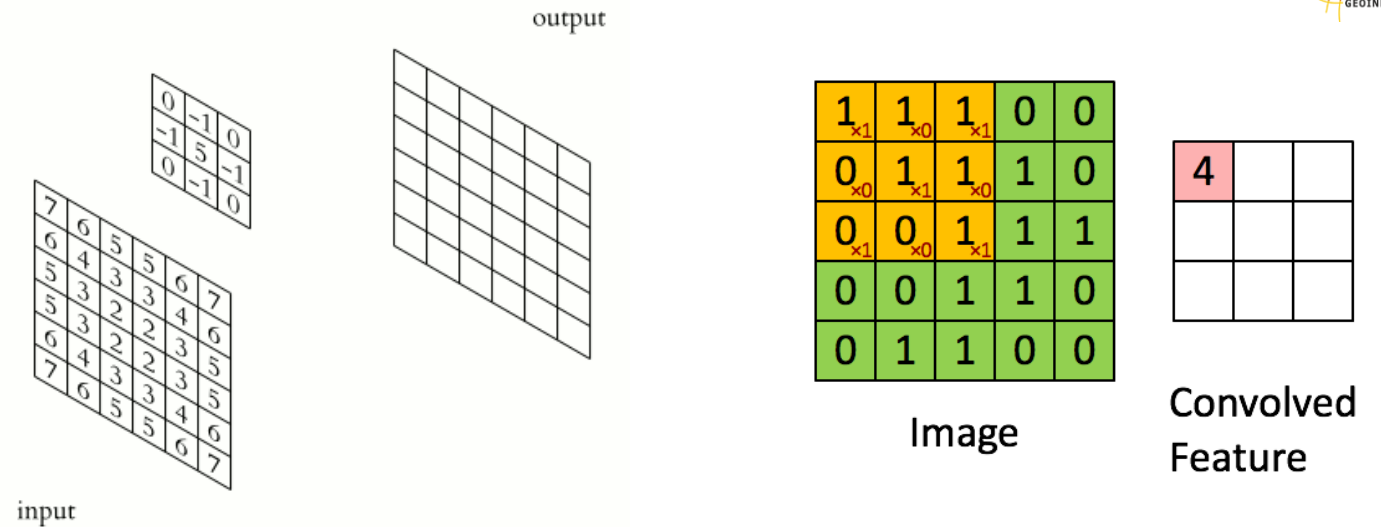
# Filters in CNN

## Feature-Detektoren

kleinen rechteckigen Bereich (Patch)



Grauwertbilder (d.h., Bilder mit einem einzigen Kanal)

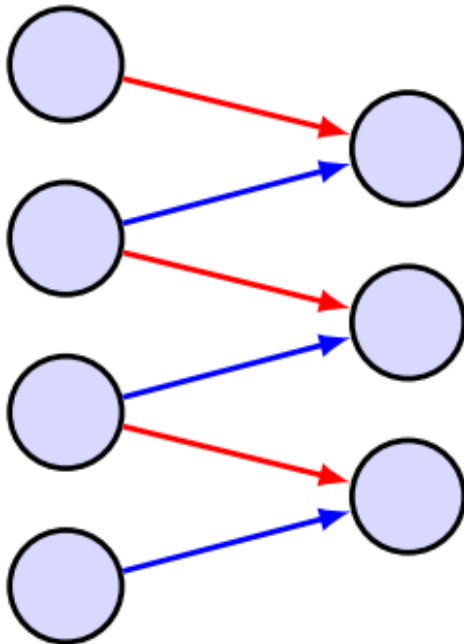


- den Begriff der Lokalität erfassen
- $t = \text{ReLU}(w^T x + w_0)$ 
  - a gewichtete lineare Kombination der Eingangswerte
  - eine nichtlineare Aktivierungsfunktion
  - x ist ein Vektor von Pixelwerten für das rezeptive Feld
- Kernel (Filter) : Gewichte
- Maximieren von  $w^T x$ : sieht aus wie das Kernel-Image
- Streng genommen wird die Faltung als Kreuz-Korrelation bezeichnet.



# Filters in CNN

## ▪ Äquivarianz der Verschiebung



- Illustration der Faltung für ein eindimensionales Feld von Eingangswerten und einem Kernel der Breite 2.
- Verbindungen mit der gleichen Farbe haben die gleichen Gewichtswerte.
- Dieses Netz hat also sechs Verbindungen, aber nur zwei unabhängige, lernbare Parameter.

# Filters in CNN

- Äquivarianz der Verschiebung
  - Vertikale / Horizontale Kanten erkennen: ein fester, handgefertigter Kernel

Vertikale

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



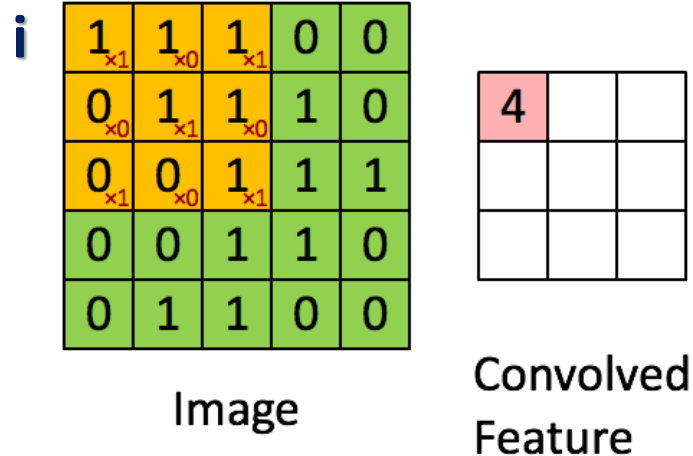
Horizontale

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



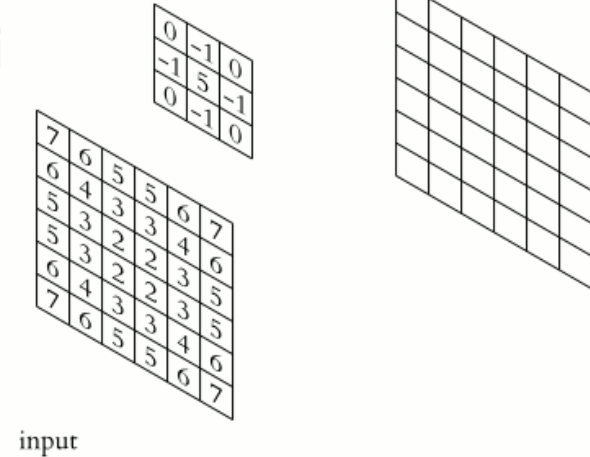
# Filters in CNN

## ■ Auffüllen (Padding)



die Faltungskarte ist kleiner als das Original Bild ( $5 < 3$ )

**ii**



## ■ Die Veränderungen der Dimensionalität ohne das Auffüllen:

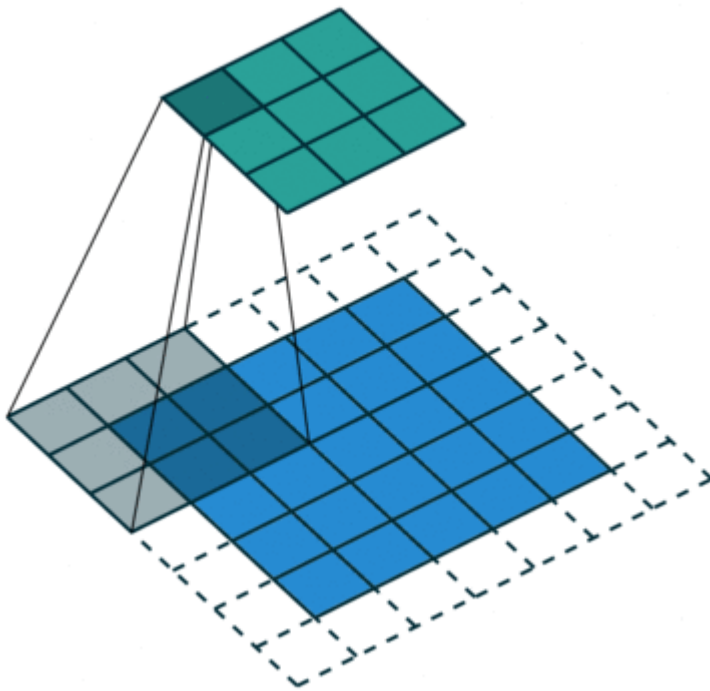
- die Dimensionalität des Bildes:  $M \times N$  Pixel
- die Dimensionalität des Kernels:  $K \times K$
- die Dimensionalität der resultierenden Feature Map (i):  
 $(M - K + 1) \times (N - K + 1) = (5 - 3 + 1) \times (5 - 3 + 1) = 3 \times 3$

## ■ Die Veränderungen der Dimensionalität mit dem Auffüllen:

- die Dimensionalität des Bildes:  $M \times N$  Pixel
- die Dimensionalität des Kernels:  $K \times K$
- Padding:  $P$  Pixel
- die Dimensionalität der resultierenden Feature Map (ii):  
 $(M + 2P - K + 1) \times (N + 2P - K + 1) = (6 + 2 \times 1 - 3 + 1) \times (6 + 2 \times 1 - 3 + 1) = 6 \times 6$

# Filters in CNN

## ■ Schrittwerte (stride)



- Feature-Maps zu verwenden, die deutlich kleiner sind als das Originalbild
- schrittweise Faltungen zu verwenden
- Die Veränderungen der Dimensionalität nach dem Auffüllen:
  - die Dimensionalität des Bildes:  $M \times N$  Pixel
  - die Dimensionalität des Kernels:  $K \times K$
  - Padding:  $P$  Pixel
  - den Kernel  $S$  Pixel nacheinander über das Bild zu führen
  - die Dimensionalität der resultierenden Feature Map:

$$\left[ \frac{M + 2P - K}{s} + 1 \right] * \left[ \frac{N + 2P - K}{s} + 1 \right]$$

$$= \left[ \frac{5 + 2*1 - 3}{2} + 1 \right] * \left[ \frac{5 + 2*1 - 3}{2} + 1 \right]$$

$$= 3 * 3$$



# Filters in CNN

## ■ Max Pooling

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

- den Maximalwert aus dem Feld der Eingabedaten auswählen

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

# Filters in CNN

## ■ Average Pooling

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

- den Durchschnittswert aus dem Feld der Eingabedaten auswählen

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

# Filters in CNN

## Max Pooling & Average Pooling

max pooling

(446, 450)



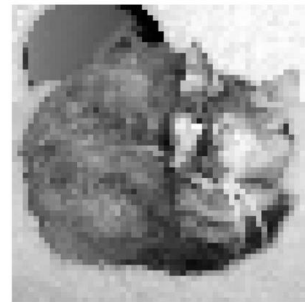
(223, 225)



(111, 112)



(55, 56)



- den Durchschnittswert aus dem Feld der Eingabedaten auswählen

average pooling

(446, 450)



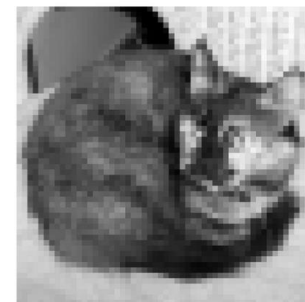
(223, 225)



(111, 112)



(55, 56)



## Take Home Messages

Im Vergleich dieser Faltungsstruktur mit einem standardmäßigen vollverknüpften Netz sehen wir mehrere Vorteile:

- **Verbindungen sind wenig zahlreich**, was selbst bei großen Bildern zu einer weitaus geringeren Anzahl von Gewichten führt.
- **Die Gewichtungswerte werden gemeinsam genutzt**, wodurch sich die Zahl der unabhängigen Parameter deutlich reduziert und folglich auch die erforderliche Größe der zum Training dieser Parameter benötigten Menge.
- **Dasselbe Netz kann auf Bilder unterschiedlicher Größe angewendet werden, ohne dass ein erneutes Training erforderlich ist.**
- **Durch die Änderung der Größe des Eingangsbildes wird lediglich die Größe der Feature-Map geändert, nicht aber die Anzahl der Gewichte oder die Anzahl der unabhängigen, lernbaren Parameter im Modell.**



## Take Home Messages

### Filters in CNN

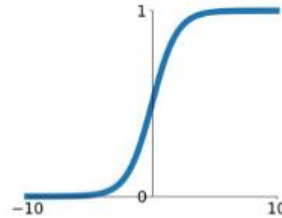
- Kernel
- Auffüllen (Padding)
- Schrittwerte (stride)
- Max Pooling & Average Pooling

# Nächste Schritte:

## Convolutional Neural Network (CNN) Aktivierungsfunktionen...

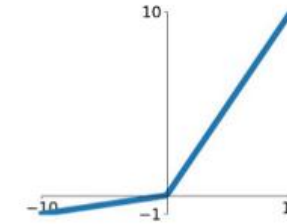
### Sigmoid

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



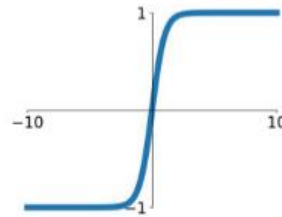
### Leaky ReLU

$$\max(0.1x, x)$$



### tanh

$$\tanh(x)$$

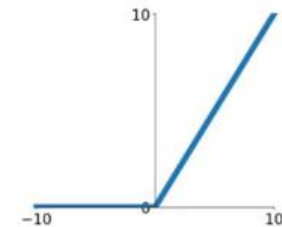


### Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

### ReLU

$$\max(0, x)$$



### ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

