DEEP LEARNING

NEURONALE NETZE

Deep Learning

HERZLICH WILLKOMMEN IM KURS!

VIELEN DANK, DASS ICH DIR NEURONALE NETZE & ARTIFICIAL INTELLIGENCE BEIBRINGEN DARF!

WAS IST DAS ZIEL VON DIESEM KURS?

- Du kannst Neuronale Netze für deine eigenen Projekte anwenden
- Du kannst eine einfache AI für ein Spiel implementieren
- Du verstehst, wie Deep Learning / AI funktioniert
- Wir verwenden hierfür fertige Tools
  - Keras / Tensorflow
  - Tensorflow.js

WIE IST DIESER KURS AUFGEBAUT?

WIE IST DIESER KURS AUFGEBAUT?

- Crashkurs Python
- Ein einzelnes Neuron
- Neuronales Netz
- Bilderkennung
- Convolutional-Neural-Networks

WIE IST DIESER KURS AUFGEBAUT?

- GPU in der Cloud
- Autos auf Bildern erkennen
- Fertige Modelle verwenden (Transfer Learning / Model Zoo)
- Hunde und Katzen voneinander unterscheiden
- Autoencoder (z.B. Bildrauschen entfernen)

WIE IST DIESER KURS AUFGEBAUT?

- Exkurs: Kunstwerke erschaffen (Neural Image Style Transfer)
- Recurrent-Neural-Networks:
  - Bitcoin-Kurs vorhersagen
  - Text generieren (buchstabenweise)
- Wort-Embeddings:
  - Text generieren (wortweise)

WIE IST DIESER KURS AUFGEBAUT?

- Crashkurs Webentwicklung
- Praxisprojekt: Donald-Trump Text-Generator
  - Als Webseiten, mit Tensorflow.js
- Projekt Flappy-Bird-Klon:
  - Wir schauen uns den Quellcode vom Spiel an
  - Und entwickeln einen Bot mit Q-Learning!

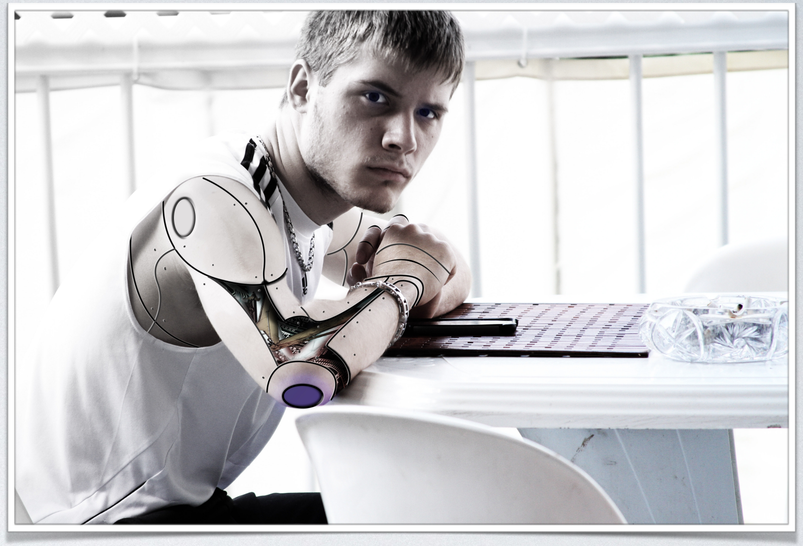
VIEL SPAß IM KURS!

BEGRIFFSABGRENZUNGEN

MACHINE LEARNING

- Der Computer extrahiert Zusammenhänge aus Daten
- Dies kann, muss aber nicht durch „Deep Learning“ erfolgen

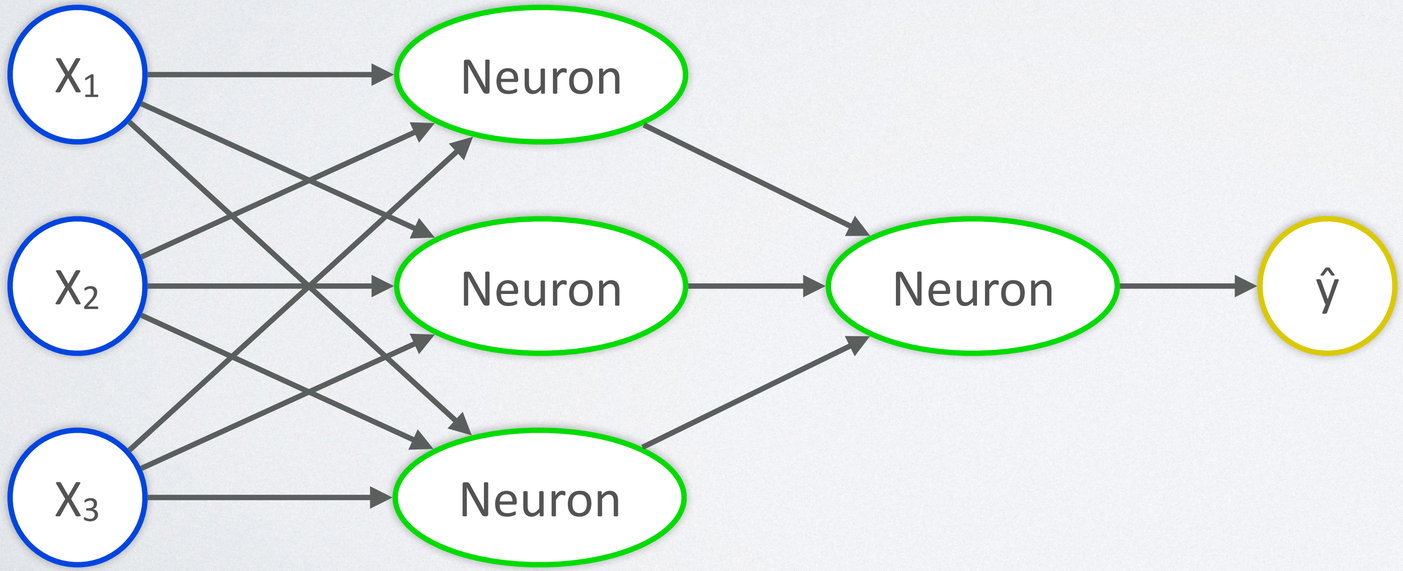
ARTIFICIAL INTELLIGENCE



- **Starke AI:**
  - Menschenähnliches Denken
- **Schwache AI:**
  - Einzelne Fähigkeiten
  - Bilderkennung
  - Autofahren,…

DEEP LEARNING / NEURONALE NETZE

- **Eine spezifische Art der „schwachen AI“**



NATURAL LANGUAGE PROCESSING

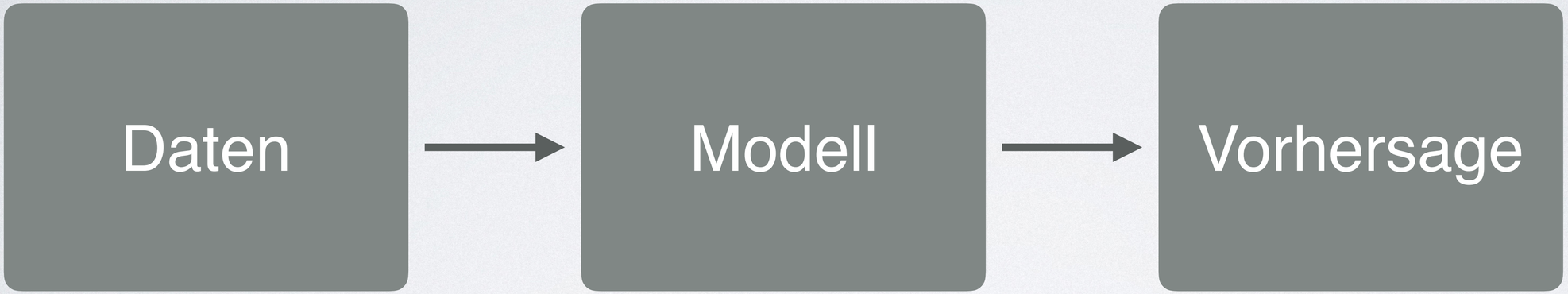
- **Wir verarbeiten „natürliche Sprache“**
- **Das Modell ist grundsätzlich egal egal**
- **Beispiele:**
  - Spamfilter (i.d.R. Naive Bayes)
  - Textgenerierung (Neuronales Netz)
  - Sentiment Analysis

ABGRENZUNG DER BEGRIFFE

- **Machine Learning**
- **Deep Learning**
- **Neuronale Netze**
- **Artificial Intelligence**

WAS IST EIN MODELL?

WAS IST EIN MODELL?



WAS FÜR EINE VORHERSAGE MÖCHTEST DU MACHEN?

- **Eine Zahl: Regression**
- **Eine Kategorie: Klassifizierung**
- **Ja / Nein: Klassifizierung**
- **Textgenerierung: Klassifizierung**

WARUM:
DEEP LEARNING &
ARTIFICIAL INTELLIGENCE?

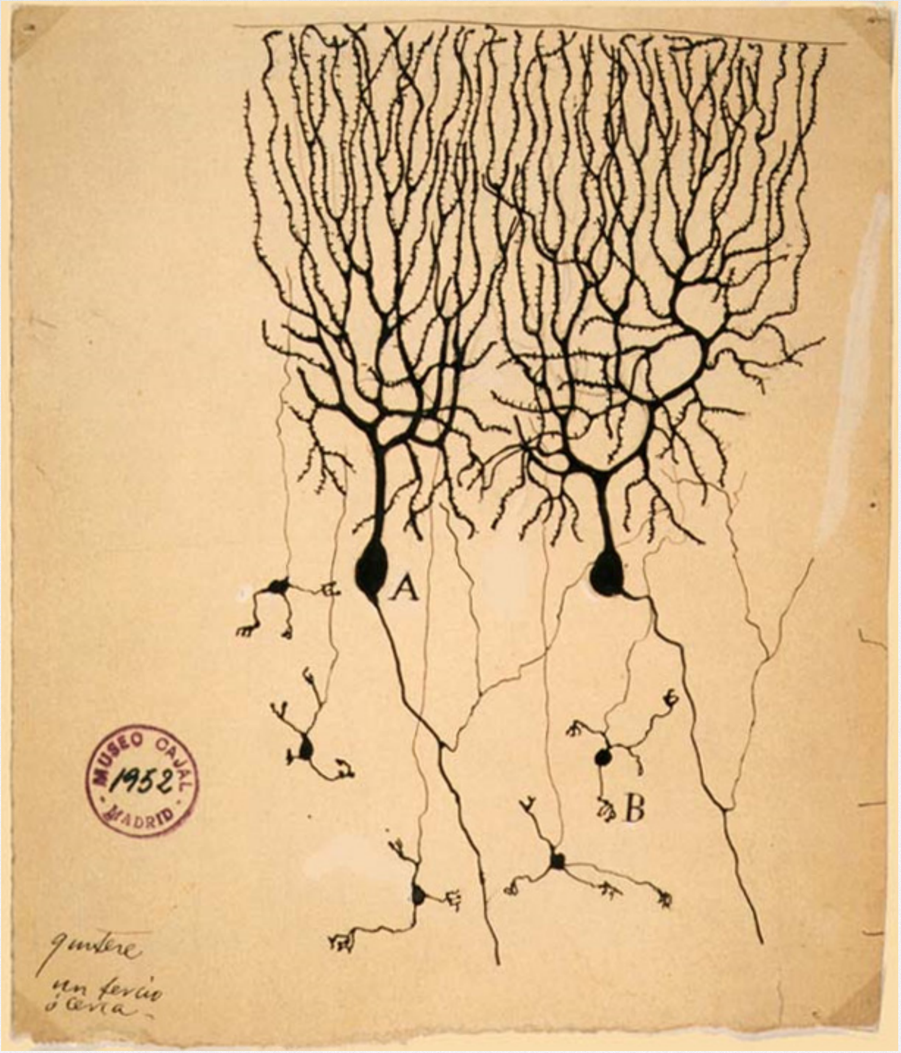
EIN EINZELNES NEURON

Deep Learning

NEURONALE NETZE: EIN EINZELNES NEURON

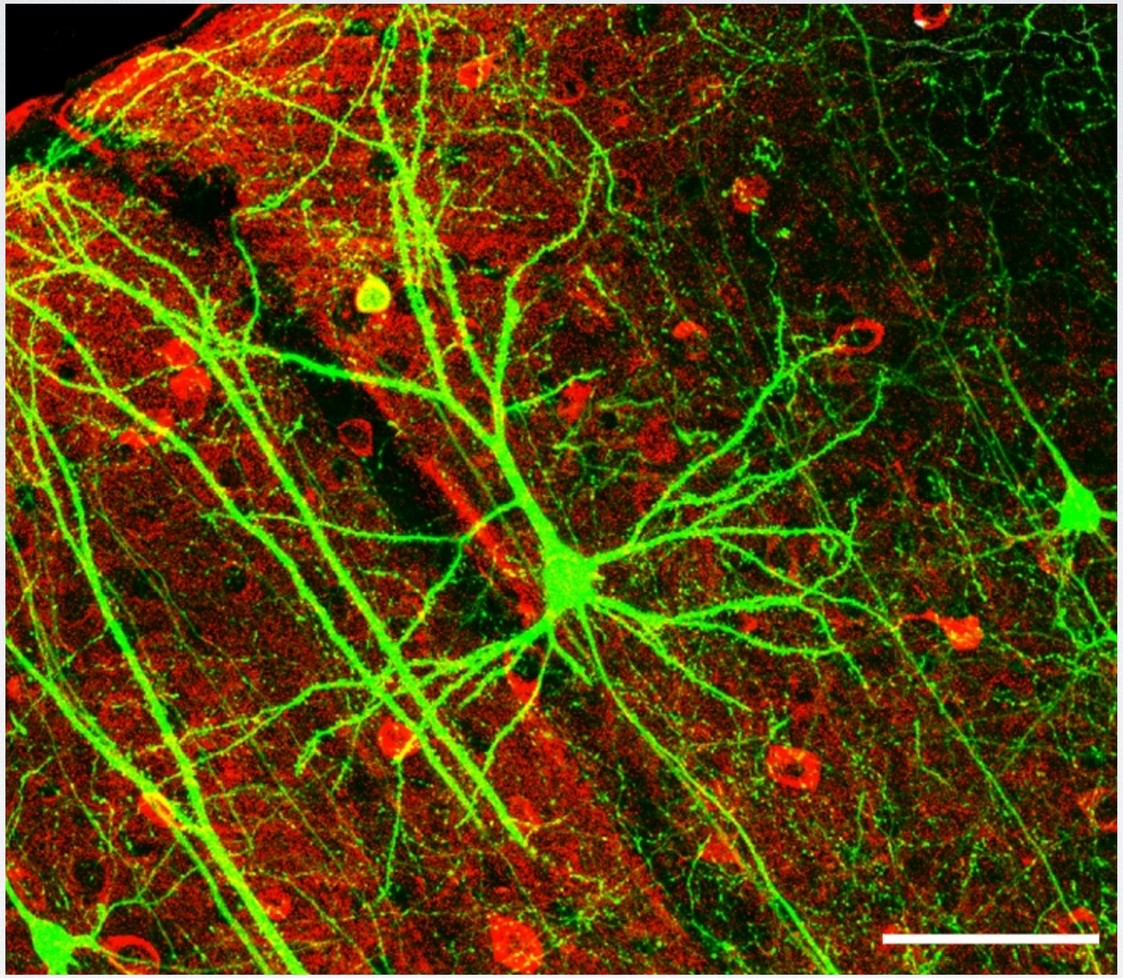
Deep Learning

NEURONALE NETZE



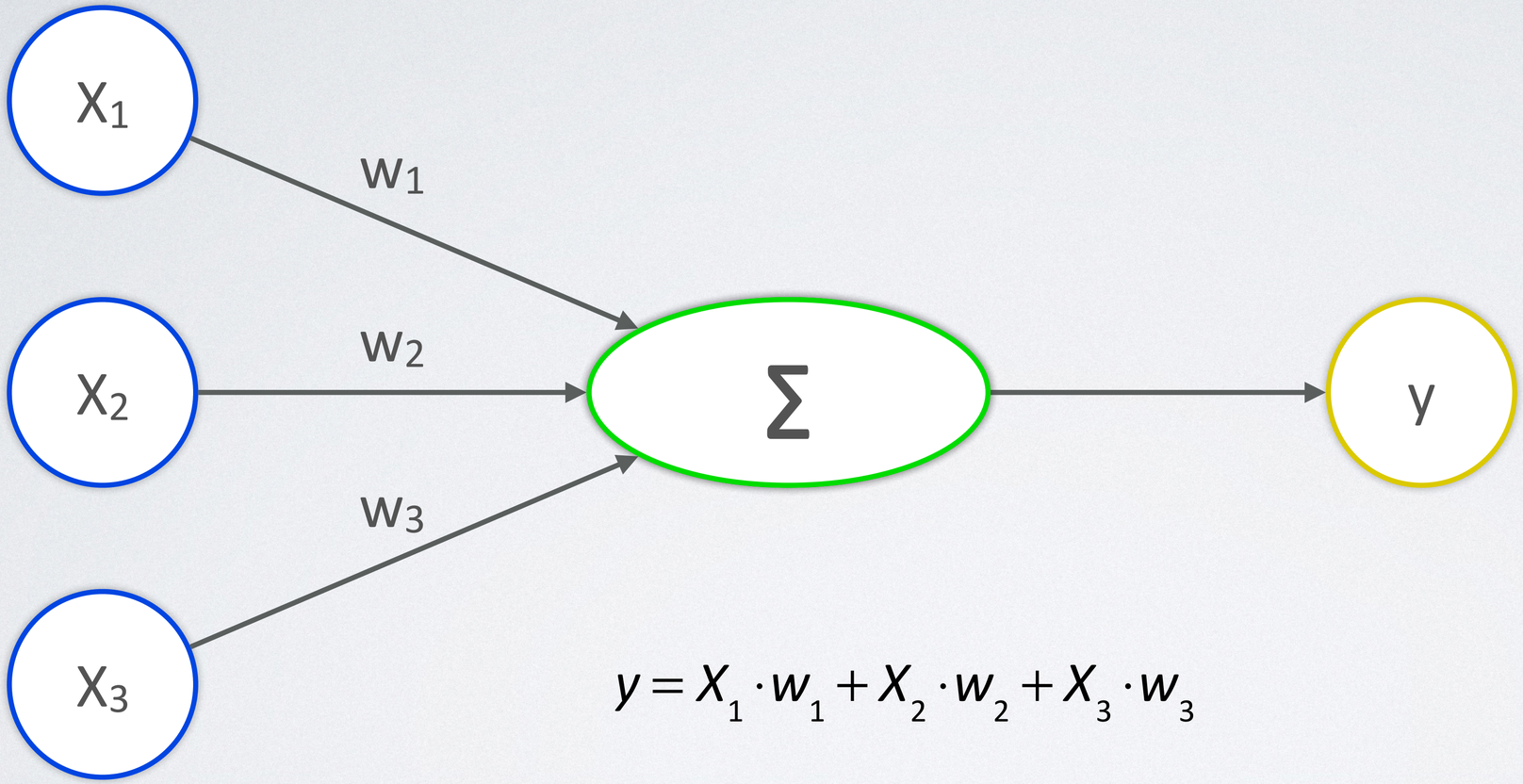
Santiago Ramón y Cajal, 1899

EIN EINZELNES NEURON



CC-BY 2.5, Wei-Chung Allen Lee, Hayden Huang, Guoping Feng, Joshua R. Sanes, Emery N. Brown, Peter T. So, Elly Nedivi- Dynamic Remodeling of Dendritic Arbors in GABAergic Interneurons of Adult Visual Cortex. Lee WCA, Huang H, Feng G, Sanes JR, Brown EN, et al. PLoS Biology Vol. 4, No. 2, e29. doi:10.1371/journal.pbio.0040029, Figure 6f, slightly altered (plus scalebar, minus letter "f".)

EIN EINZELNES NEURON



$$y = X_1 \cdot w_1 + X_2 \cdot w_2 + X_3 \cdot w_3$$

NEURONALE NETZE:
EIN EINZELNES NEURON (MIT BIAS)

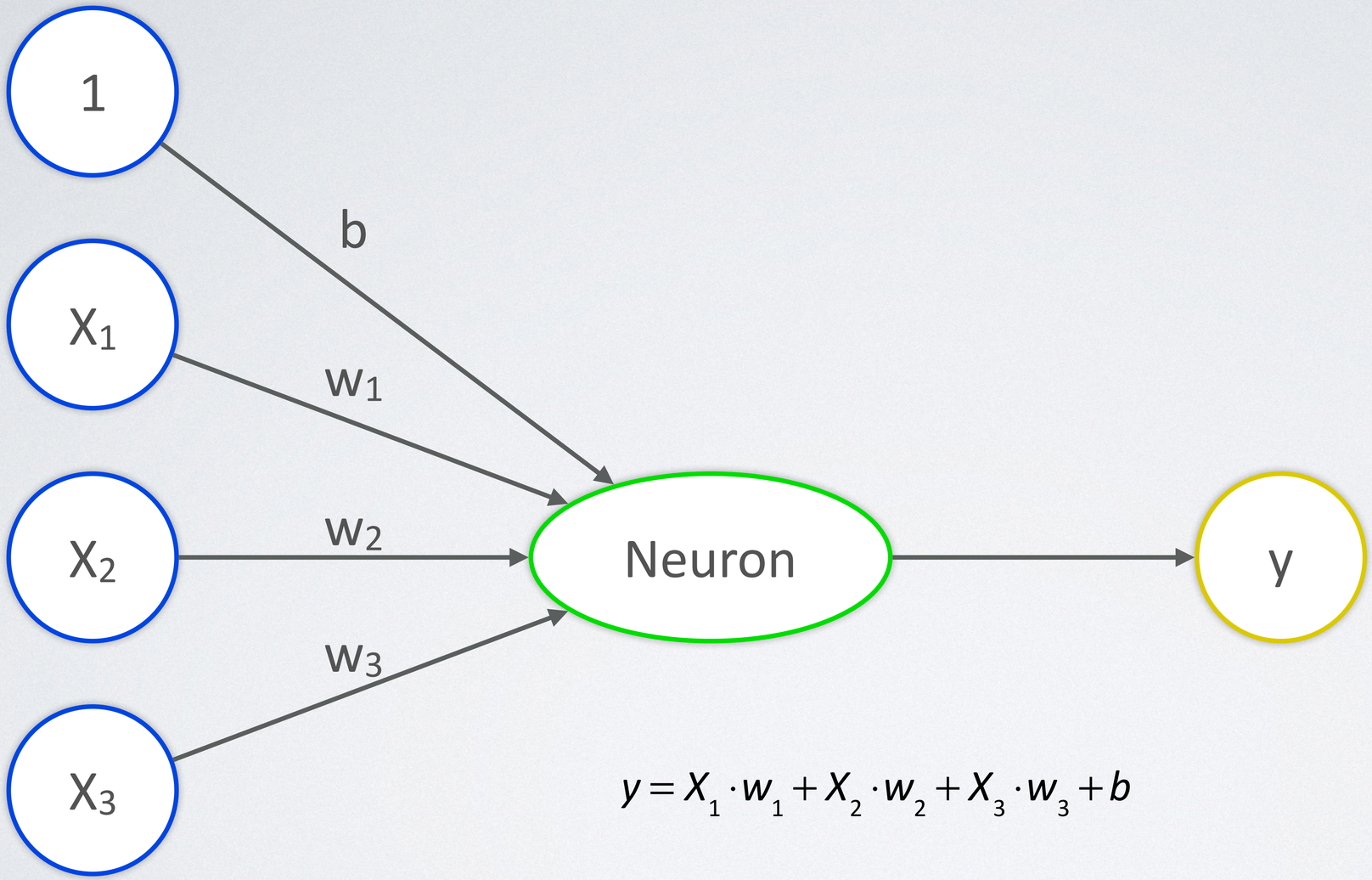
Deep Learning

WARUM BRAUCHEN WIR EINEN BIAS?

- **Beispiel:** Grad Celsius -> Fahrenheit
  - Formel: [Celsius] * 1,8 + 32

$$y = X_1 \cdot w_1$$

EIN EINZELNES NEURON



$$y = X_1 \cdot w_1 + X_2 \cdot w_2 + X_3 \cdot w_3 + b$$

NEURONALE NETZE: DIE AKTIVIERUNGSFUNKTION

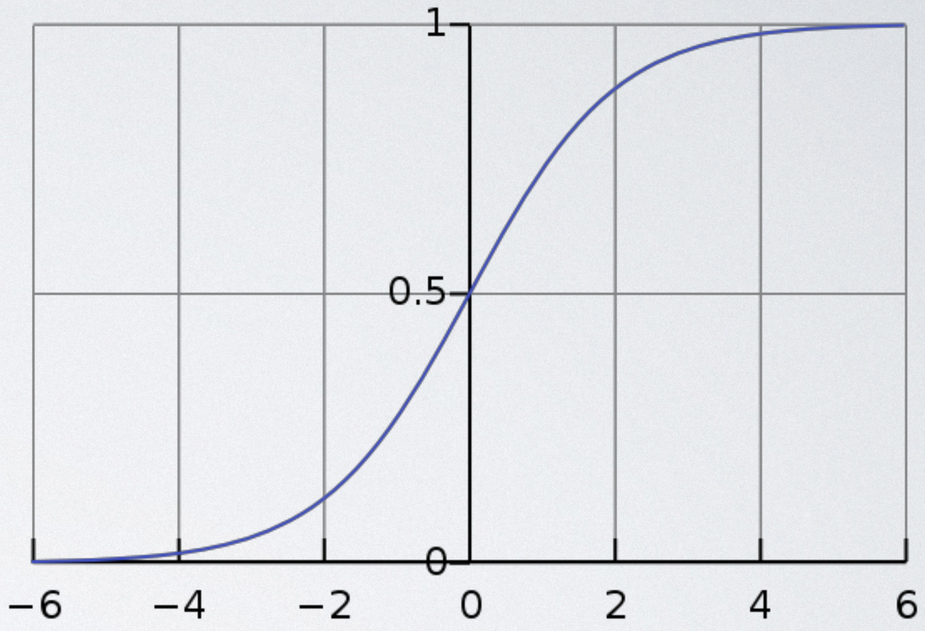
Deep Learning

WARUM BRAUCHEN WIR EINE AKTIVIERUNGSFUNKTION?

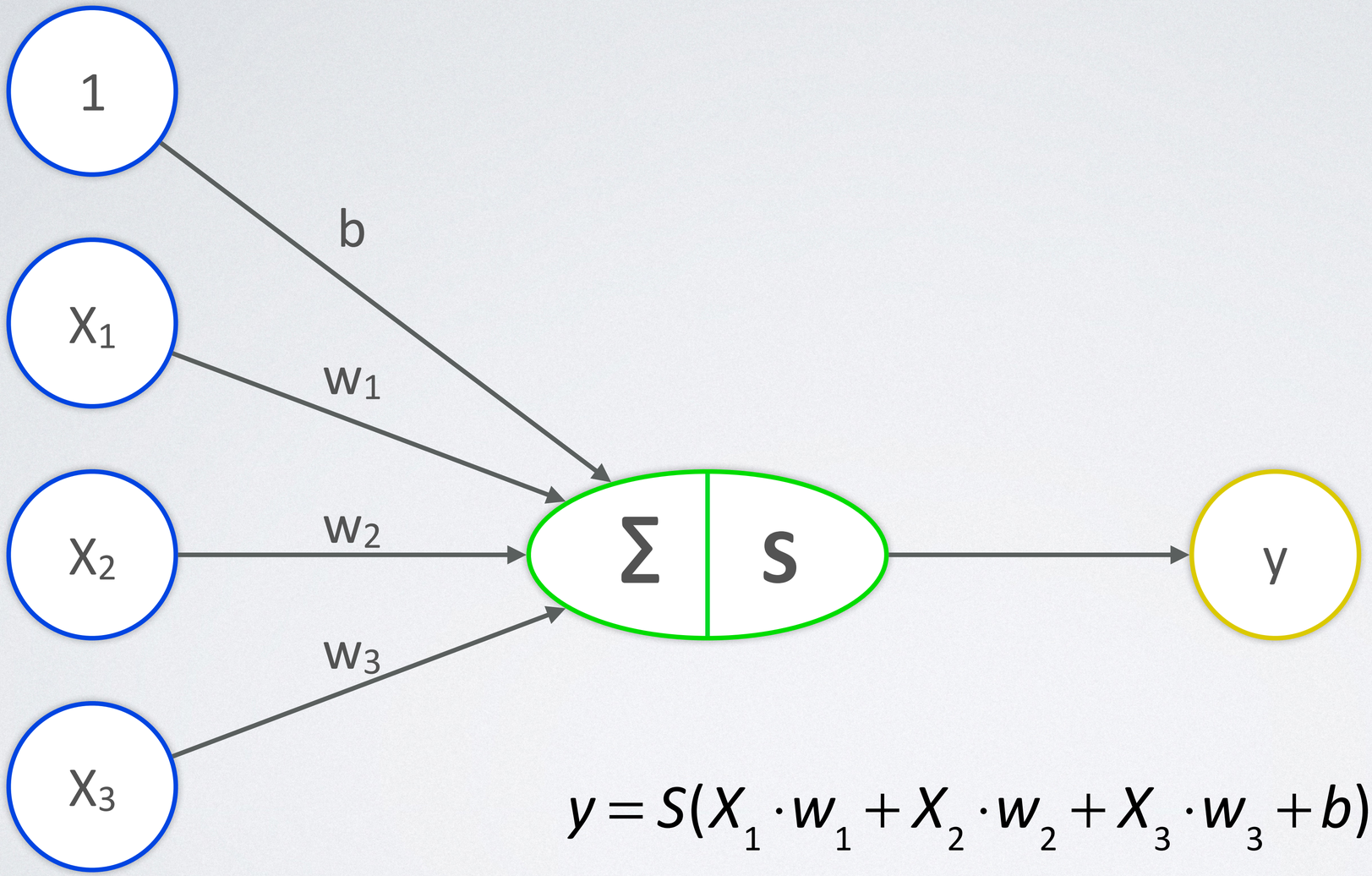
- **Problem 1:**
  - Wir können unsere einfachen Neuronen nicht „hintereinanderschalten“
- **Problem 2:**
  - Modell: Brustkrebs - Ja oder Nein?
  - Hier möchten wir nur Ergebnisse zwischen 0 und 1 bekommen

IDEE: AKTIVIERUNGSFUNKTION

- Du wirst später noch weitere Aktivierungsfunktionen kennenlernen
- **Beispiel:**
  - **Sigmoid-Funktion**
  - Damit bilden wir alle Ergebnisse auf den Zahlenbereich 0…1 ab.

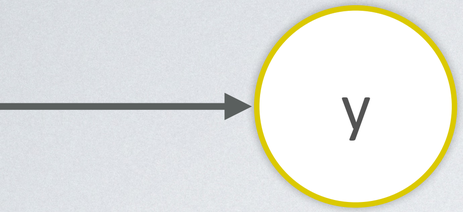


EIN EINZELNES NEURON



$$y = S(X_1 \cdot w_1 + X_2 \cdot w_2 + X_3 \cdot w_3 + b)$$

WAS IST JETZT UNSERE VORHERSAGE?

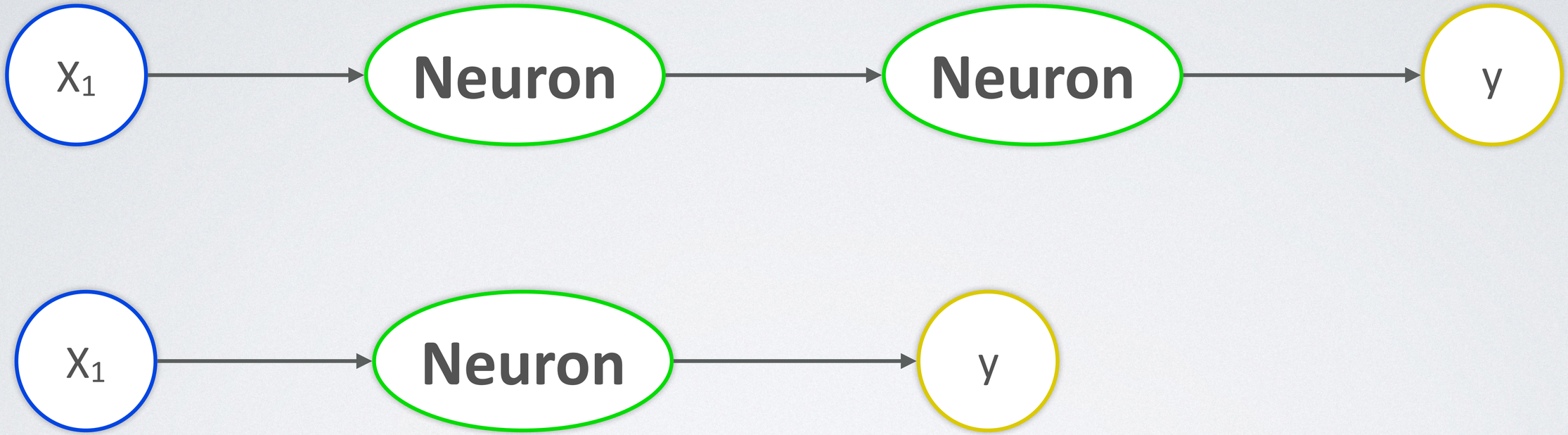


- Was ist jetzt unser y?
- Ein Wert zwischen 0 und 1
- Diesen können wir als eine Wahrscheinlichkeit auffassen

OPTIONALER EXKURS: WARUM KÖNNEN WIR LINEARE NEURONEN NICHT HINTEREINANDER SCHALTEN?

Deep Learning

WARUM KONNTEN WIR LINEARE NEURONEN NICHT HINTEREINANDER SCHALTEN?



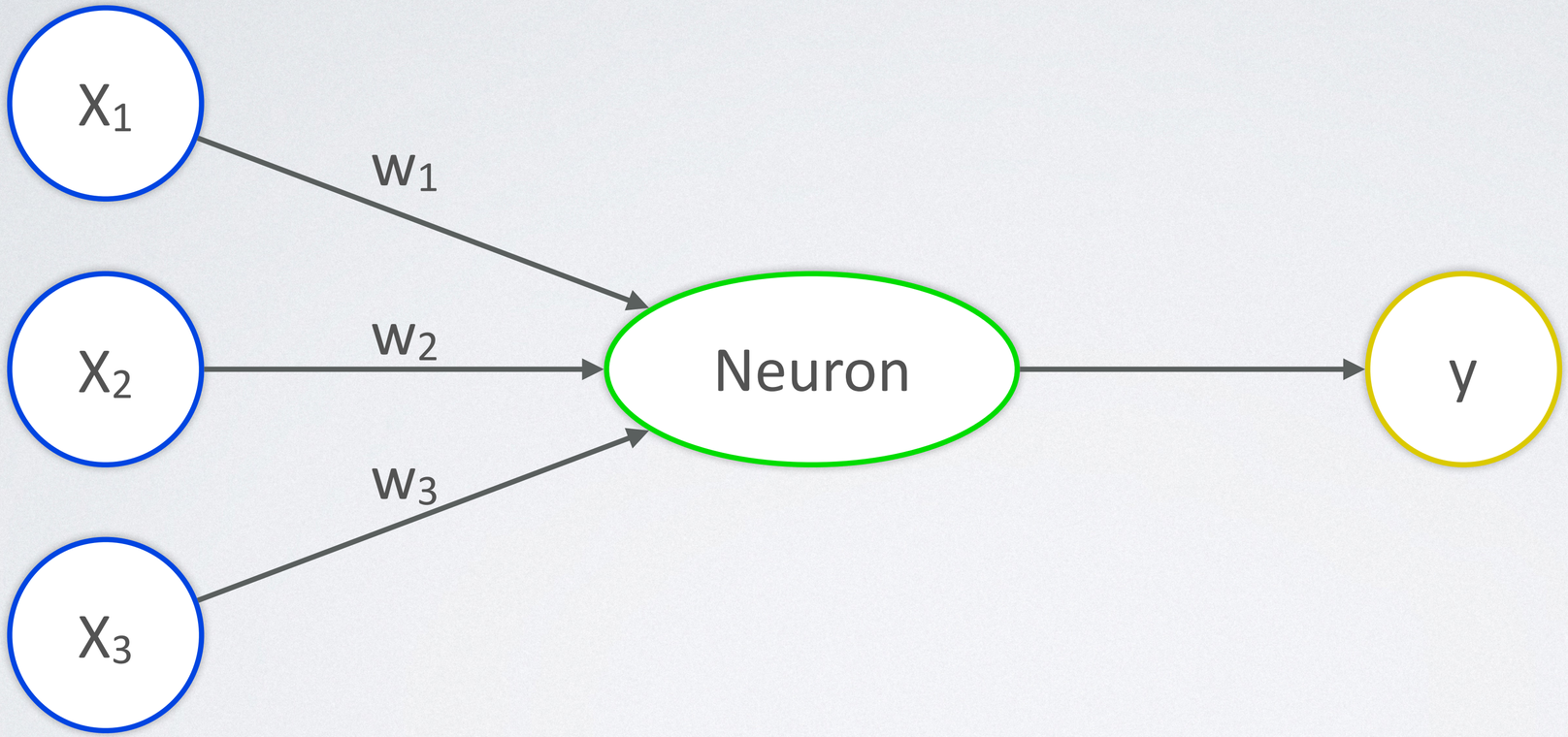
EIN NEURONALES NETZ

Deep Learning

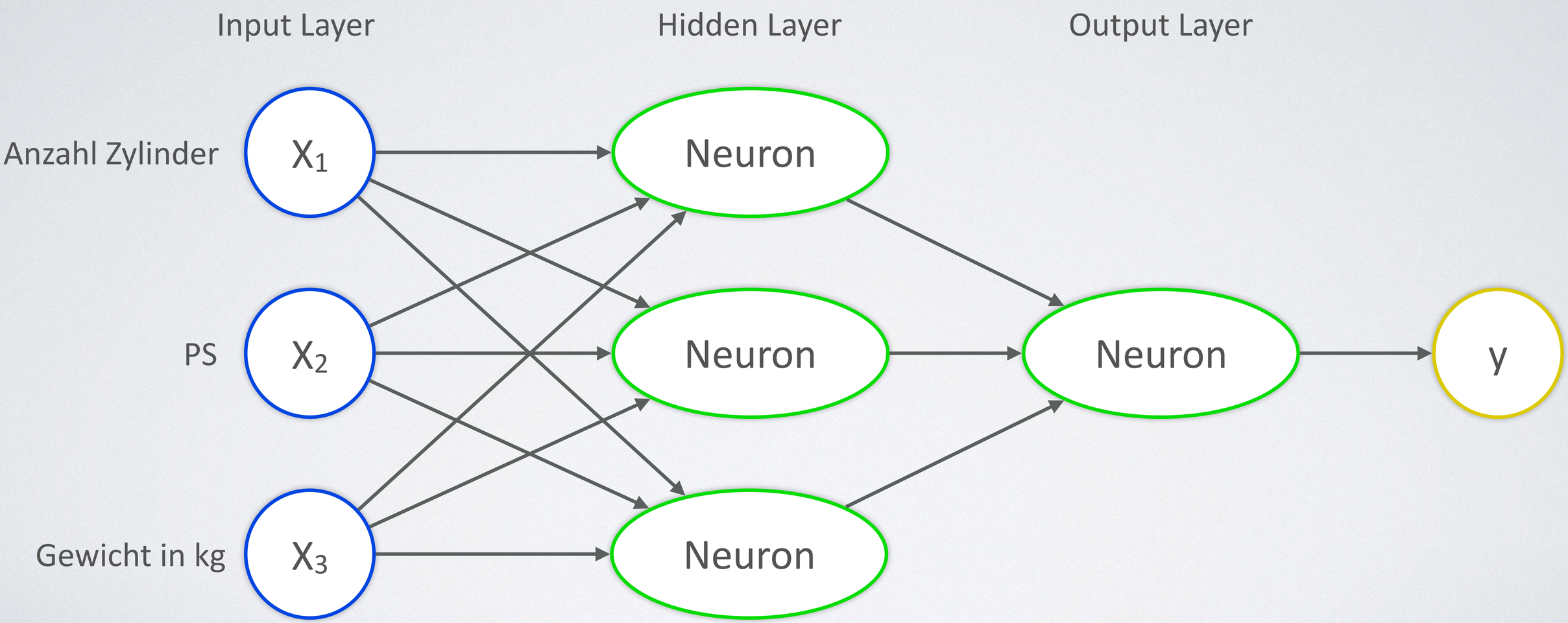
NEURONALE NETZE: EIN EINFACHES NETZ

Deep Learning

VOM NEURON ZUM NEURONALEN NETZ



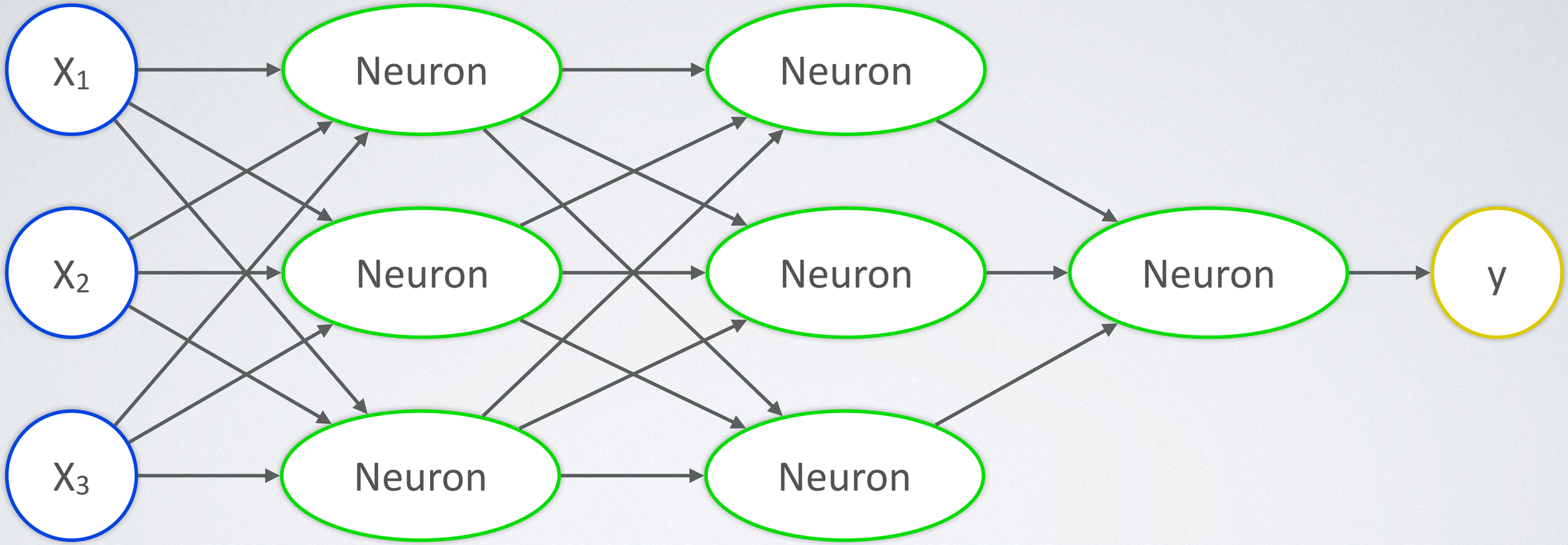
NEURONALES NETZ



NEURONALES NETZ

- Warum brauchen wir einen Hidden-Layer?
- **Es gilt grundsätzlich:**
  - Neuronale Netze mit einem beliebig großen Hidden-Layer können jede beliebige Funktion approximieren!

EIN KOMPLEXERES NEURONALES NETZ



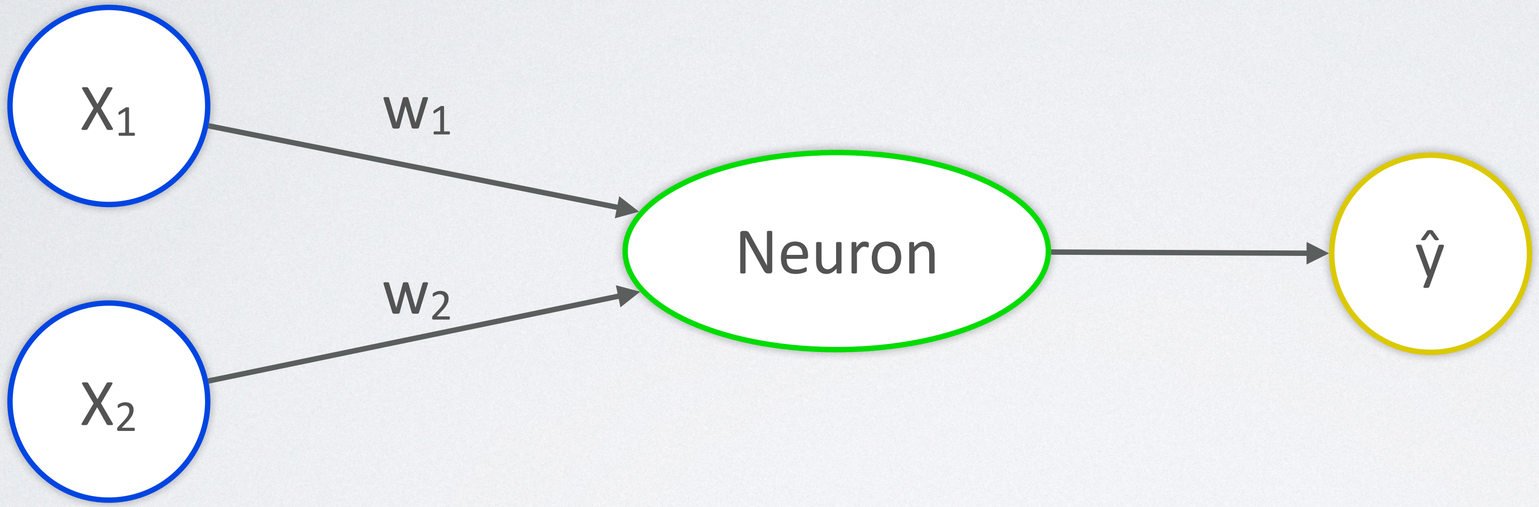
NEURONALE NETZE:
WIE WERDEN DIE GEWICHTE AKTUALISIERT?

Deep Learning

OVER RIDE
66
ENTRANCE
ROUTE66
THE LEVEL

WIE WERDEN DIE GEWICHTE AKTUALISIERT?

VOM NEURON ZUM NEURONALEN NETZ



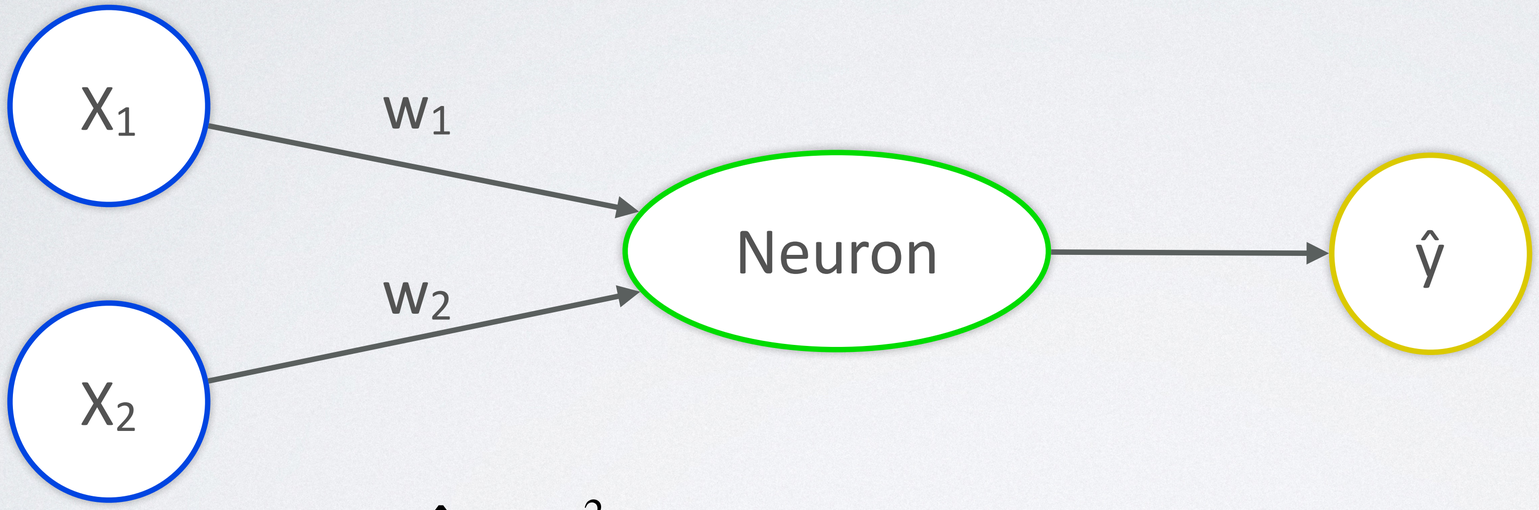
$\hat{y} = 0.5$

$y = 0.75$

NEURONALE NETZE: DIE KOSTENFUNKTION

Deep Learning

VOM NEURON ZUM NEURONALEN NETZ



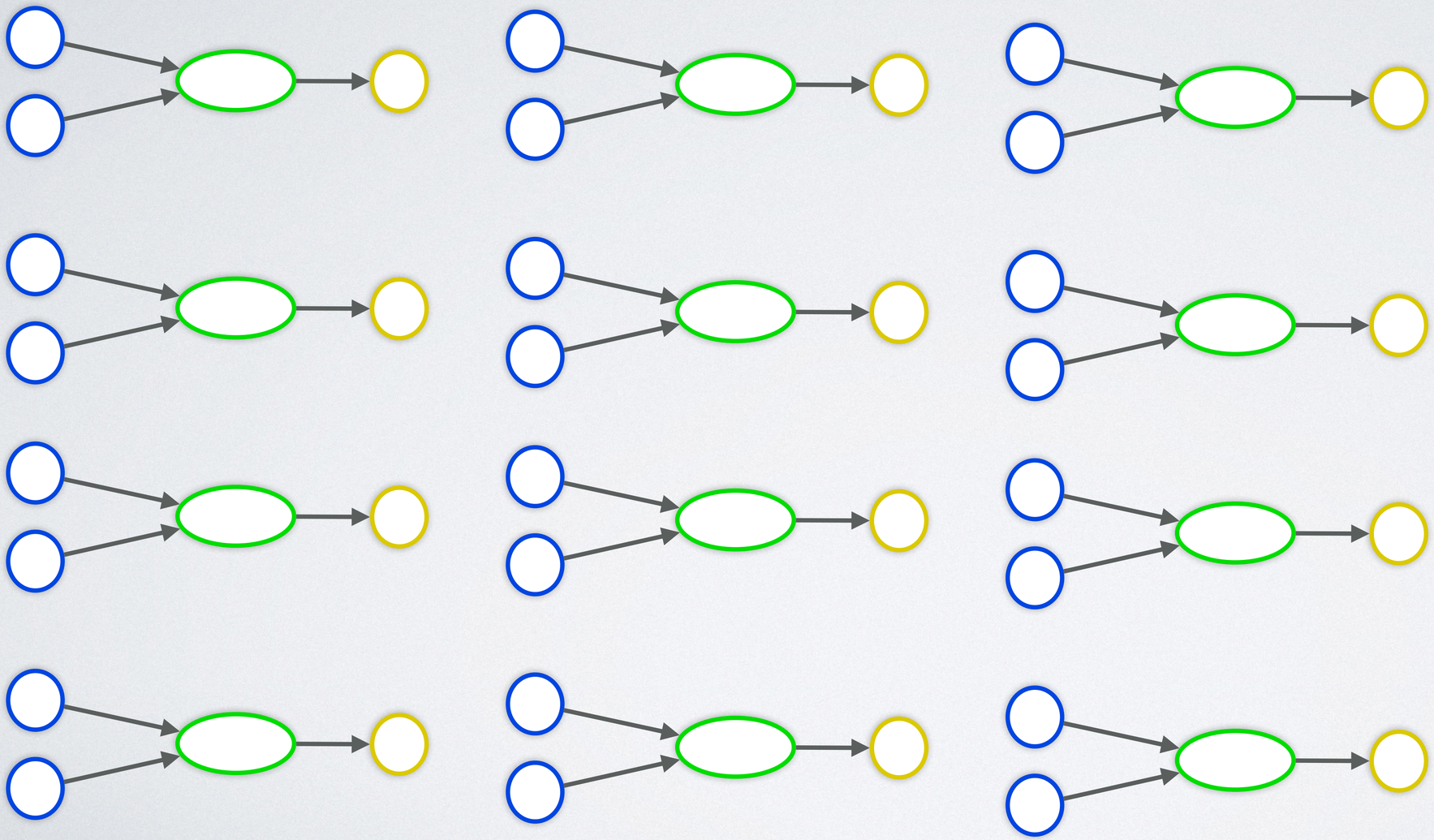
$\hat{y} = 0.5$

$y = 0.75$

$$C = (\hat{y} - y)^2$$

$$C = (X_1 \cdot w_1 + X_2 \cdot w_2 - y)^2$$

VOM NEURON ZUM NEURONALEN NETZ



$$C = \sum (\hat{y} - y)^2$$

NEURONALE NETZE: STOCHASTIC GRADIENT DESCENT

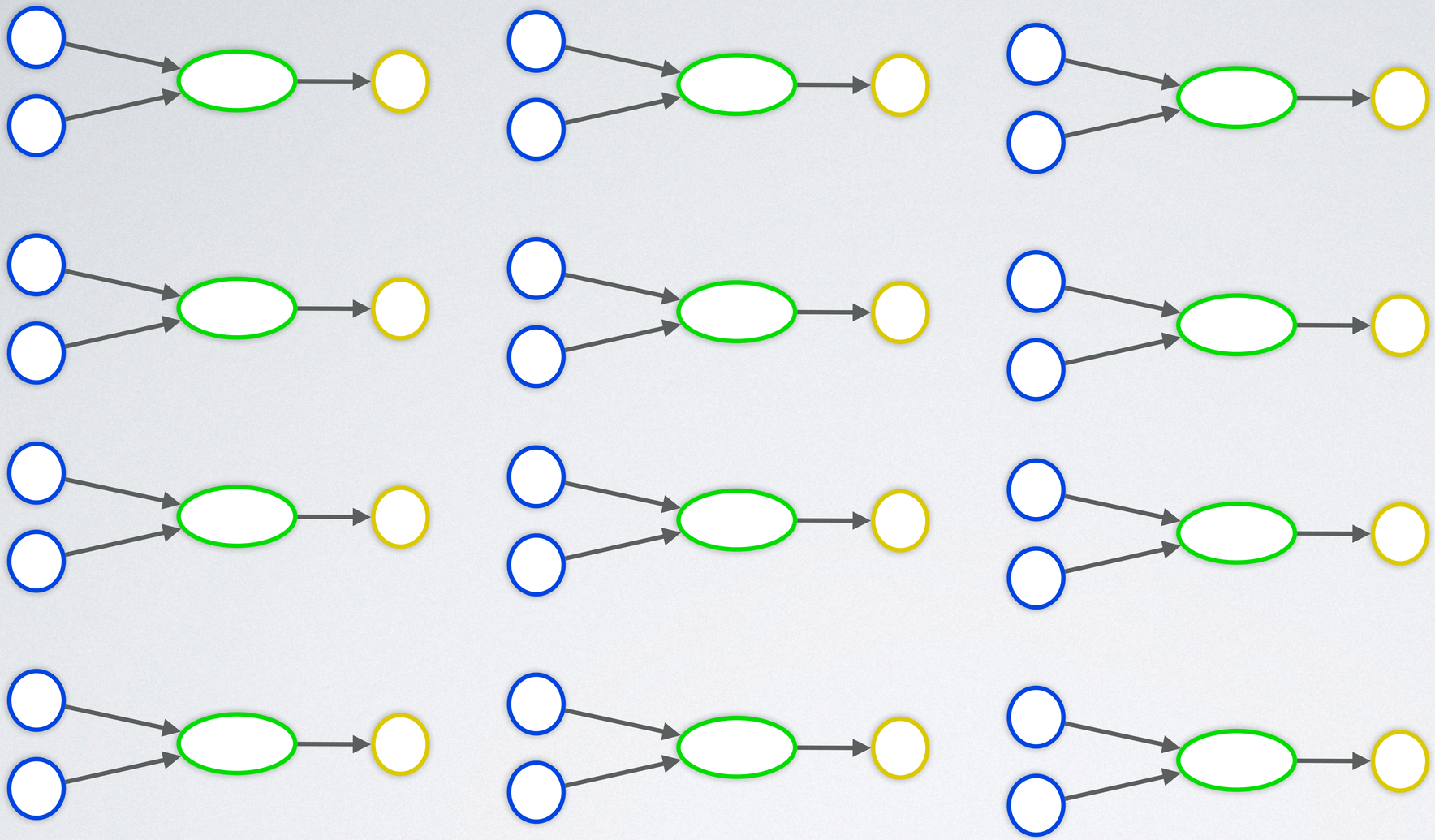
Deep Learning

STOCHASTIC GRADIENT DESCENT

- **Zur Erinnerung:**
  - Ziel war es, unsere Kosten C zu minimieren
  - Dazu berechnen wir die Kosten für unsere gesamten Trainingsdaten, und aktualisieren anschließend unsere Gewichte w

STOCHASTIC GRADIENT DESCENT

- **Problem:**
  - Diese Berechnung dauert oft lange
  - Die Kosten müssen für alle Trainingsdaten berechnet werden
  - Geht das nicht schneller?



$$C = \sum (\hat{y} - y)^2$$

$$C^{approx} = \sum (\hat{y} - y)^2 \quad C^{approx} = \sum (\hat{y} - y)^2 \quad C^{approx} = \sum (\hat{y} - y)^2$$

STOCHASTIC GRADIENT DESCENT

- **Begriffsdefinition:**
  - Epoche: 1x Trainingsdaten durchlaufen
  - Batch: Wir berechnen einmal die Kosten und aktualisieren danach die Gewichte
  - Lernrate: Wie groß sind unsere Schritte?

NEURONALE NETZE: STOCHASTIC GRADIENT DESCENT

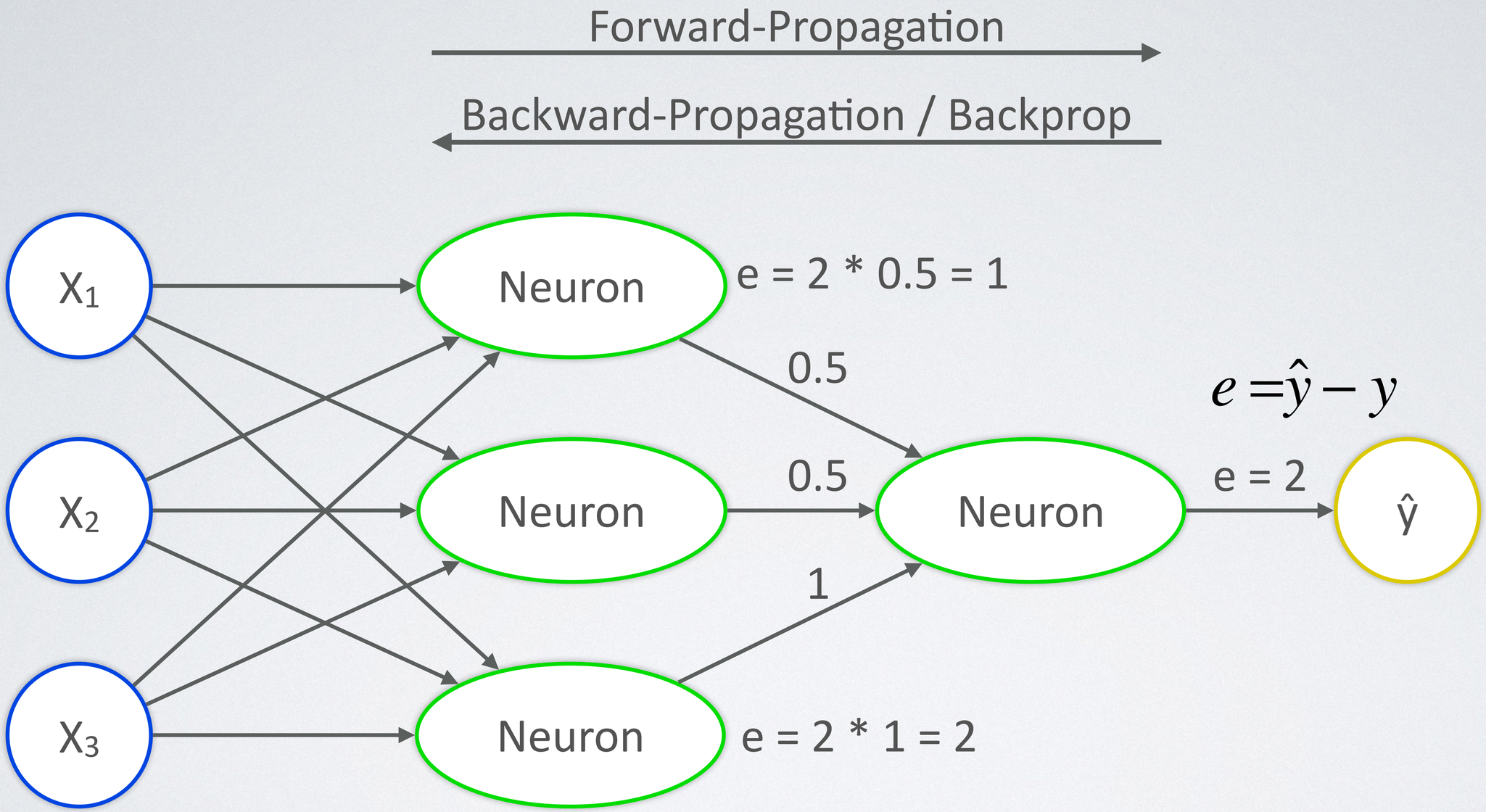
Deep Learning

NEURONALE NETZE: BACKPROPAGATION

Deep Learning

BACKPROPAGATION

- Aber wie werden die Gewichte der vorherigen Schichten aktualisiert?
- **Backpropagation!**

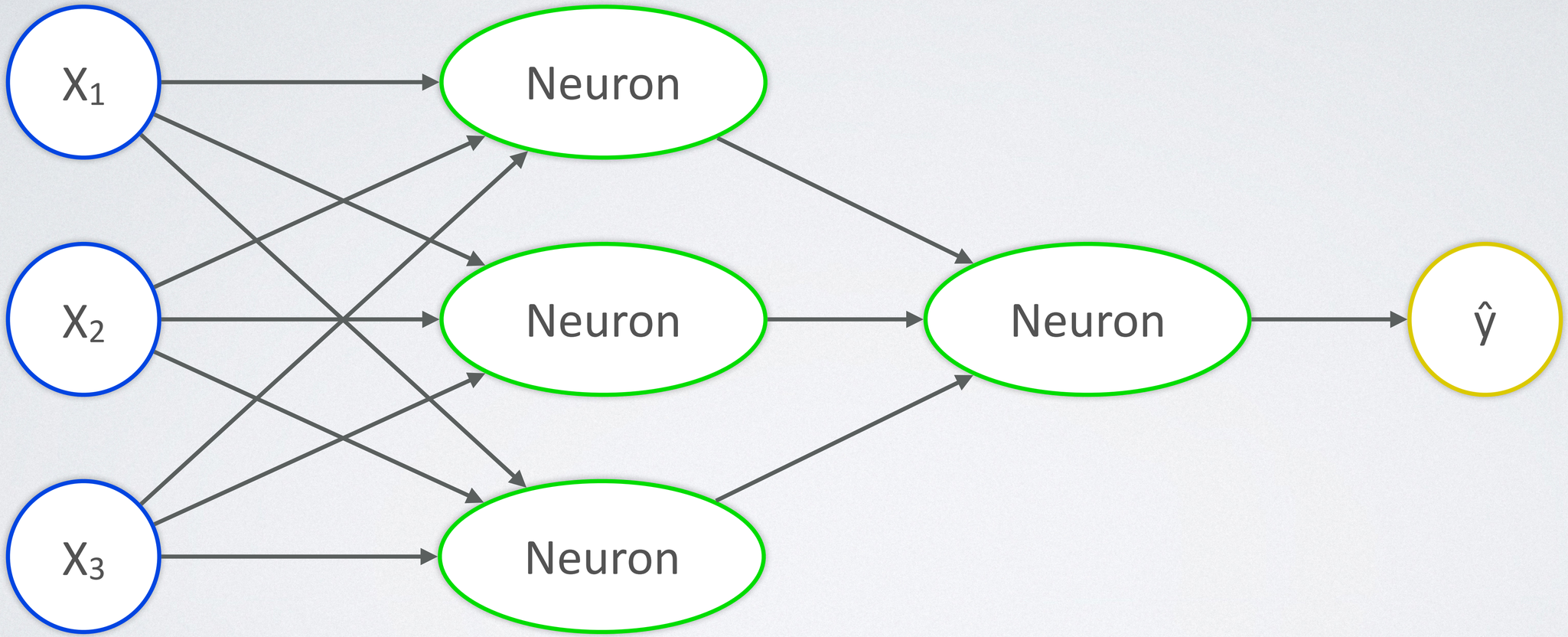

Forward-Propagation
Backward-Propagation / Backprop
X1
X2
X3
Neuron
Neuron
Neuron
Neuron
e = 2 * 0.5 = 1
0.5
0.5
1
e = 2 * 1 = 2
$e = \hat{y} - y$
e = 2
ŷ

NEURONALE NETZE:
BACKPROPAGATION - WIE INITIALISIEREN WIR DIE GEWICHTE?

Deep Learning

INITIALISIERUNG DER GEWICHTE

- Diese Frage kann darüber entscheiden, ob wir unser Neuronales Netz trainieren können oder nicht:
  - Wie initialisieren wir die Gewichte?


X_1
X_2
X_3
Neuron
Neuron
Neuron
Neuron
$\hat{y}$

NEURONALE NETZE: KERAS UND TENSORFLOW

Deep Learning

Keras

TensorFlow

TENSORFLOW

- Von Google entwickelt
- In C++ / CUDA geschrieben
- Wir könnten damit auch direkt arbeiten…

Keras

- Sehr benutzerfreundlich
- Modelle können einfach entwickelt werden
- Direkt für Python entwickelt
- Dennoch volle Kontrolle möglich - ggf. über Tensorflow-Funktionen

Keras

TensorFlow

- **Tensorflow:**
  - Forschung
  - Kontrolle z.B. über die Gewichtsaktualisierung möglich
- **Keras:**
  - Sehr viel einfacher
  - Verwenden von „fertigen“ Layern

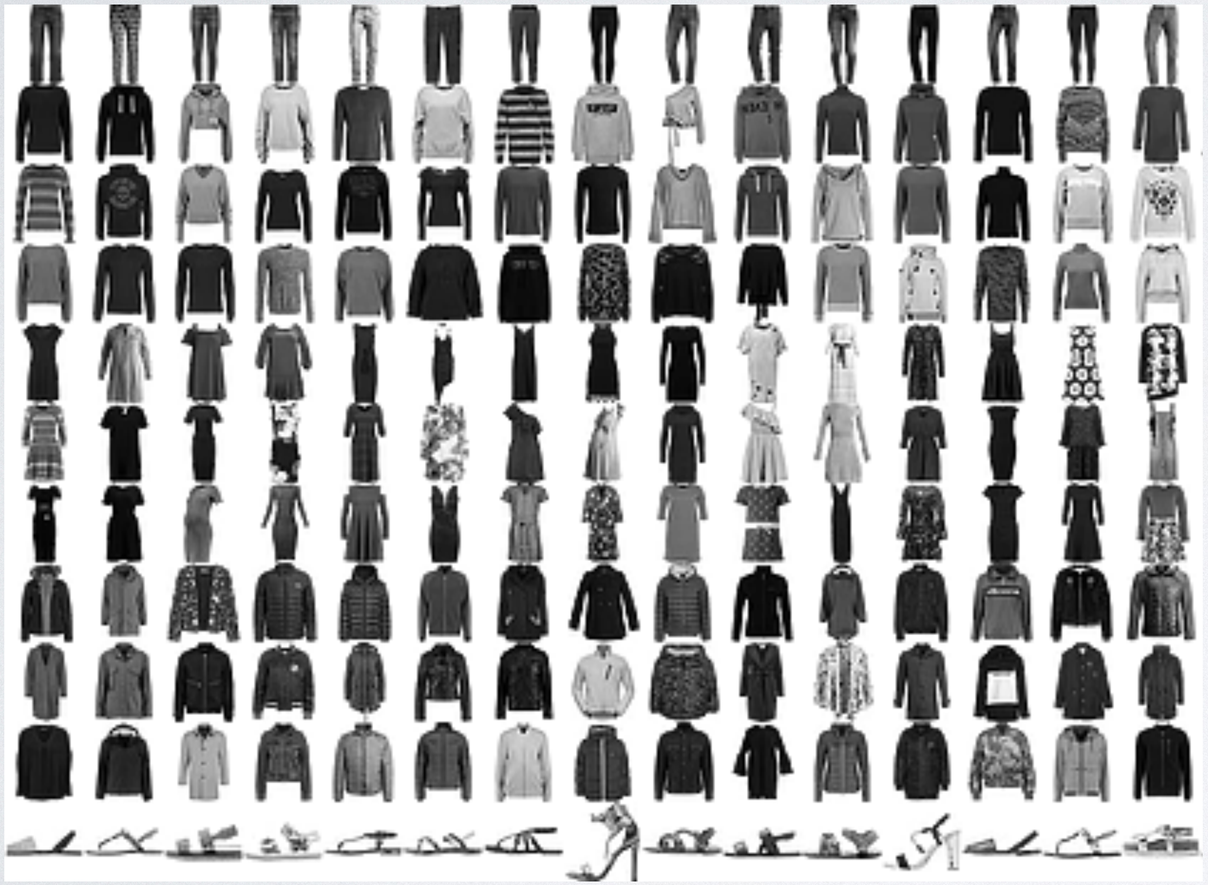
WAS FÜR EINE GRAFIKKARTE?

NEURONALE NETZE: VORSTELLUNG PROJEKT

Deep Learning

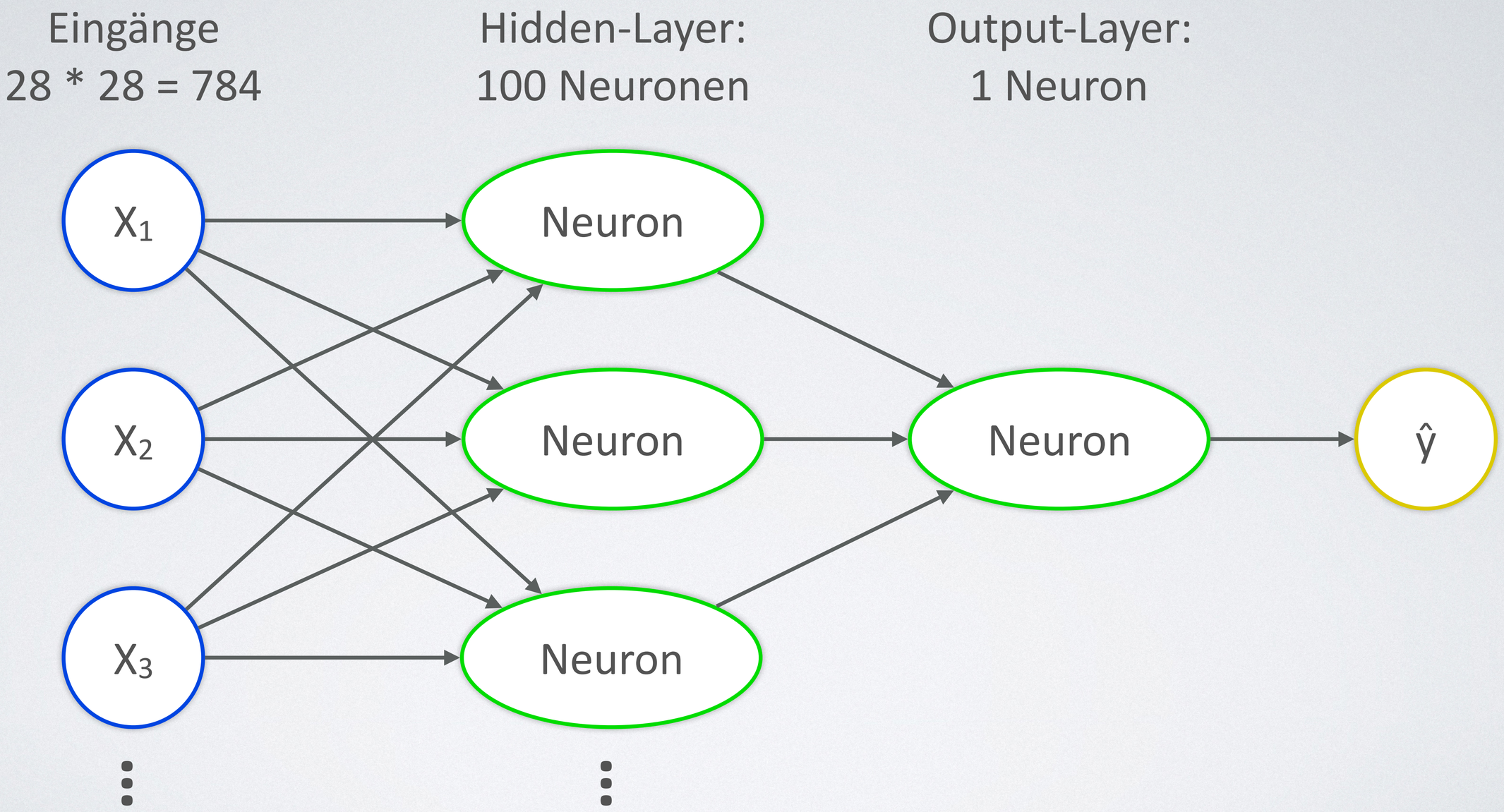
MNIST

FASHIONMNIST



FASHIONMNIST

- 10 Kategorien (T-Shirt, Pullover, Dress, Bag, …)
- Jedes Bild ist 28x28 Pixel groß
- **Problem 1:**
  - Ist auf einem Bild ein T-Shirt zu sehen oder nicht?


Eingänge
28 * 28 = 784
Hidden-Layer:
100 Neuronen
Output-Layer:
1 Neuron
X_1
X_2
X_3
Neuron
Neuron
Neuron
Neuron
$\hat{y}$

NEURONALE NETZE: TRAIN / TEST

Deep Learning

TRAIN / TEST

- **Was uns eigentlich interessiert:**
  - Wie gut ist unser Modell auf unbekannten Daten?
  - Unsere Trainingsdaten hat es ja schon gesehen!
- **Problem:**
  - **„Overfitting“** - Unser Modell passt sich zu stark an unsere Trainingsdaten an

TRAIN / TEST

- **Idee:**
  - **Wir halten Daten zurück**
  - **Diese Daten verwenden wir anschließend, um die Genauigkeit von unserem Modell zu berechnen**

NEURONALE NETZE: MEHRERE KLASSEN VORHERSAGEN!

Deep Learning

MEHRERE KLASSEN VORHERSAGEN

- **Bisher:**
  - T-Shirt: Ja / Nein
- Aber kann unser Netz nicht auch mehr?
- **Idee (One-Hot-Encoding):**
  - **Wir geben unserem Netz 10 Ausgänge!**

MEHRERE KLASSEN VORHERSAGEN

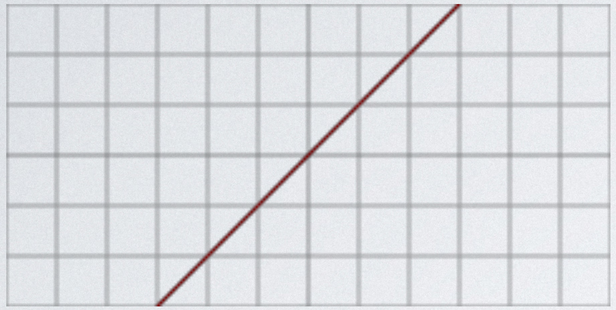
- **Idee:**
  - T-Shirt: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
  - Trouser: 0, 1, 0, 0, 0, 0, 0, 0, 0, 0
  - Pullover: 0, 0, 1, 0, 0, 0, 0, 0, 0, 0
  - …

NEURONALE NETZE: AKTIVIERUNGSFUNKTIONEN

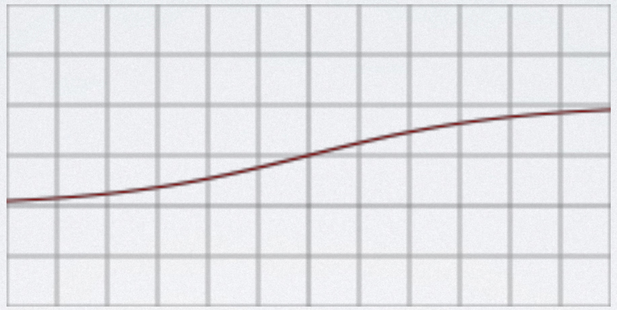
Deep Learning

AKTIVIERUNGSFUNKTIONEN

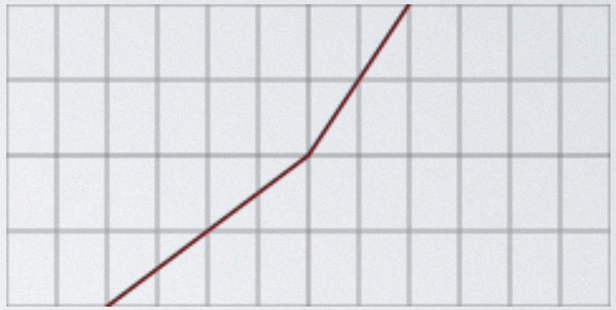
Linear



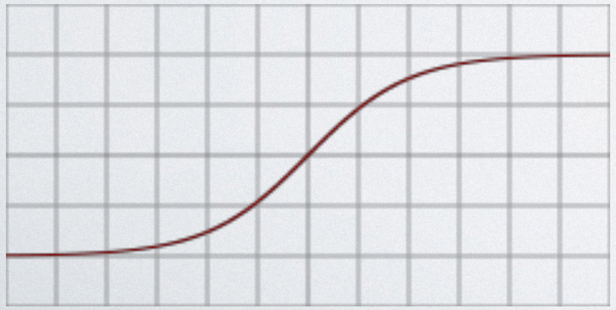
Sigmoid



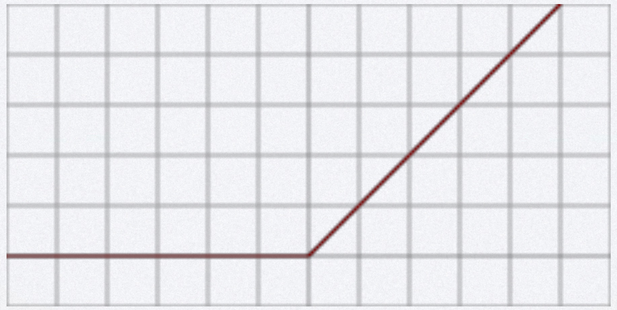
LReLu



Tanh

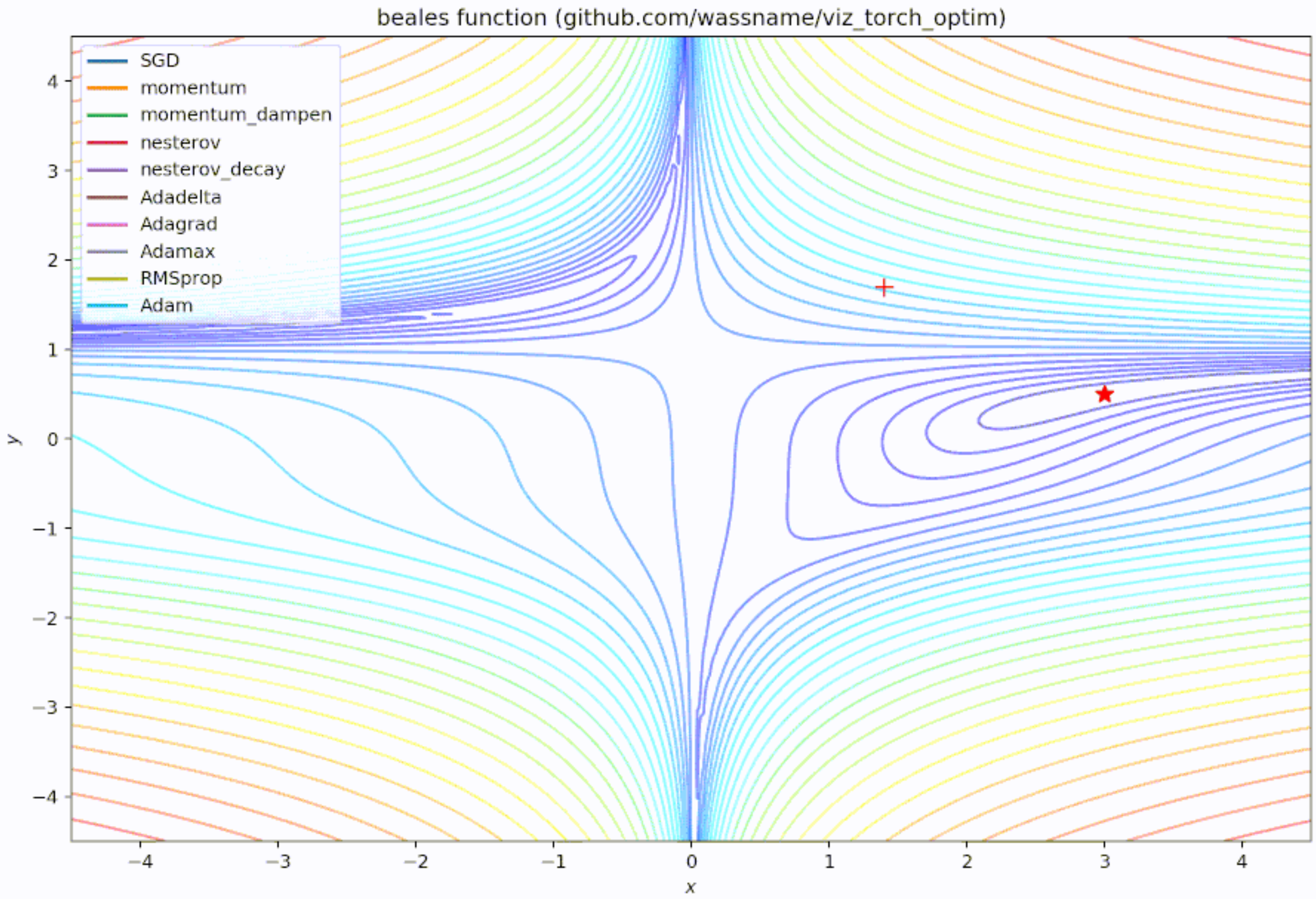


ReLU



NEURONALE NETZE:
SCHNELLER ZU EINER BESSEREN LÖSUNG KOMMEN

Deep Learning


beales function (github.com/wassname/viz_torch_optim)
SGD
momentum
momentum_dampen
nesterov
nesterov_decay
Adadelta
Adagrad
Adamax
RMSprop
Adam
4
3
2
1
0
−1
−2
−3
−4
y
−4
−3
−2
−1
0
1
2
3
4
x

NEURONALE NETZE:
WIE IST EIN BILD AUFGEBAUT?

Deep Learning

WIE IST EIN BILD AUFGEBAUT?

- **Aufbau eines Bildes:**
  - Viele Pixel in einer Matrix nebeneinander
  - Mit verschiedenen Farbwerten (Rot, Grün, Blau)

NEURONALE NETZE: CONVOLUTIONAL NEURAL NETWORKS

Deep Learning

CONVOLUTIONAL NEURAL NETWORKS (CNN)

- **Idee auch hier:**
  - Wir geben dem Neuronalen Netz zusätzliche Intuitionen über unsere Daten mit
  - Dazu können wir einen CNN-Layer verwenden
  - Aber was ist Convolution überhaupt?

CONVOLUTIONAL NEURAL NETWORKS (CNN)

- **Idee:**
  - Eine Kante ist eine Kante - egal wo sie auf dem Bild zu sehen ist
  - Unser Neuronales Netz kann auf der ersten Ebene lernen, Kanten zu erkennen
  - Und diese neuen „Features“ werden dann auf höheren Ebenen weiterverarbeitet

BEISPIEL EINER EINFACHEN CONVOLUTION

$$[+1 \quad 0 \quad -1] \cdot \begin{array}{|c|c|c|}\hline 1 & 1 & 0 \\ \hline \end{array} = 1$$

$$[+1 \quad 0 \quad -1] \cdot \begin{array}{|c|c|c|}\hline 0 & 1 & 1 \\ \hline \end{array} = -1$$

$$[+1 \quad 0 \quad -1] \cdot \begin{array}{|c|c|c|c|c|c|c|}\hline 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|}\hline 0 & 1 & 0 & -1 & 0 \\ \hline \end{array}$$

CONVOLUTIONAL NEURAL NETWORKS (CNN)

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

*

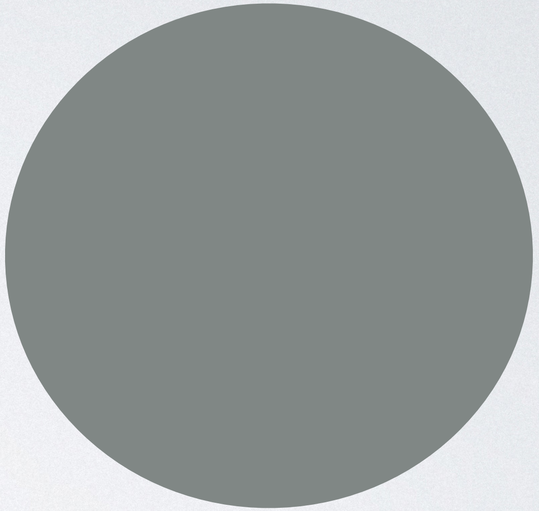
1	0	-1
1	0	-1
1	0	-1

=

-2	2	2
-3	3	3
-2	2	2

CONVOLUTIONAL NEURAL NETWORKS (CNN)

- **Problem:**
  - Wir möchten aber nicht nur vertikale Kanten erkennen
  - Sondern z.B. auch vertikale, schräge, …
  - Wir können unserem Netz also mehrere Filter zur Verfügung stellen



CONVOLUTIONAL NEURAL NETWORKS (CNN)

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

*

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

=

-2	2	2
-3	3	3
-2	2	2

*

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

=

…	…	…
…	…	…
…	…	…

NEURONALE NETZE: MAX-POOLING

Deep Learning

IDEE HINTER MAX-POOLING-LAYER

- **Idee:**
  - Oft ist die genaue Pixel-Position einer Kante nicht so wichtig
  - **Beispiel:**
    - Solange die Kante irgendwo in der Nähe ist, ist das auch vollkommen okay

-1	-1	1	1
-2	-2	2	2
-3	-3	3	3
-2	-2	2	2

MAX-POOLING

-1	-1	1	1
-2	-2	2	2
-3	-3	3	3
-2	-2	2	2

MaxPooling(2, 2) →

-1	2
-2	3

MAXPOOLING

-1	-1	1	1
-2	-2	2	2
-3	-3	3	3
-2	-2	2	2

$\xrightarrow{\text{MaxPooling(2, 2)}}$

-1	2
-2	3

- Oft nach einem Convolutional-Layer
- **Vorteile:**
  - Netz verallgemeinert mehr
  - Anzahl der Ausgabeknoten ist geringer
  - **=> Lernen geht schneller**

NEURONALE NETZE: ZEROPADDING

Deep Learning

PROBLEM BEI CONVOLUTIONAL-LAYERS!

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

*

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

=

-2	2	2
-3	3	3
-2	2	2

PROBLEM BEI CONVOLUTIONAL-LAYERS

- **Problem:**
  - Nach einem Convolutional-Layer ist unser Bild kleiner geworden
  - Wenn wir jetzt mehrere Convolutional-Layer hintereinander hängen wollen…
  - …wird in jedem Schritt unser Bild etwas kleiner
  - Können wir dagegen nicht was tun?

PROBLEM BEI CONVOLUTIONAL-LAYERS!

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

*

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

=

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

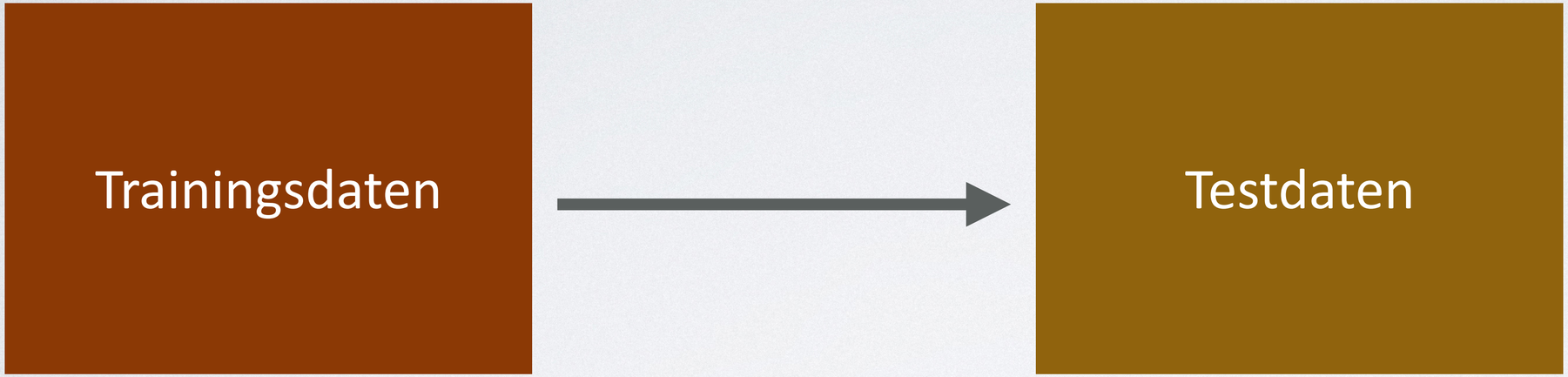
NEURONALE NETZE: DROPOUT

Deep Learning

DROPOUT - MOTIVATION

- **Motivation: Dropout**
  - Durch das Weglassen von Verbindungen können wir die Genauigkeit auf unseren Testdaten verbessern
  - Beispiel!

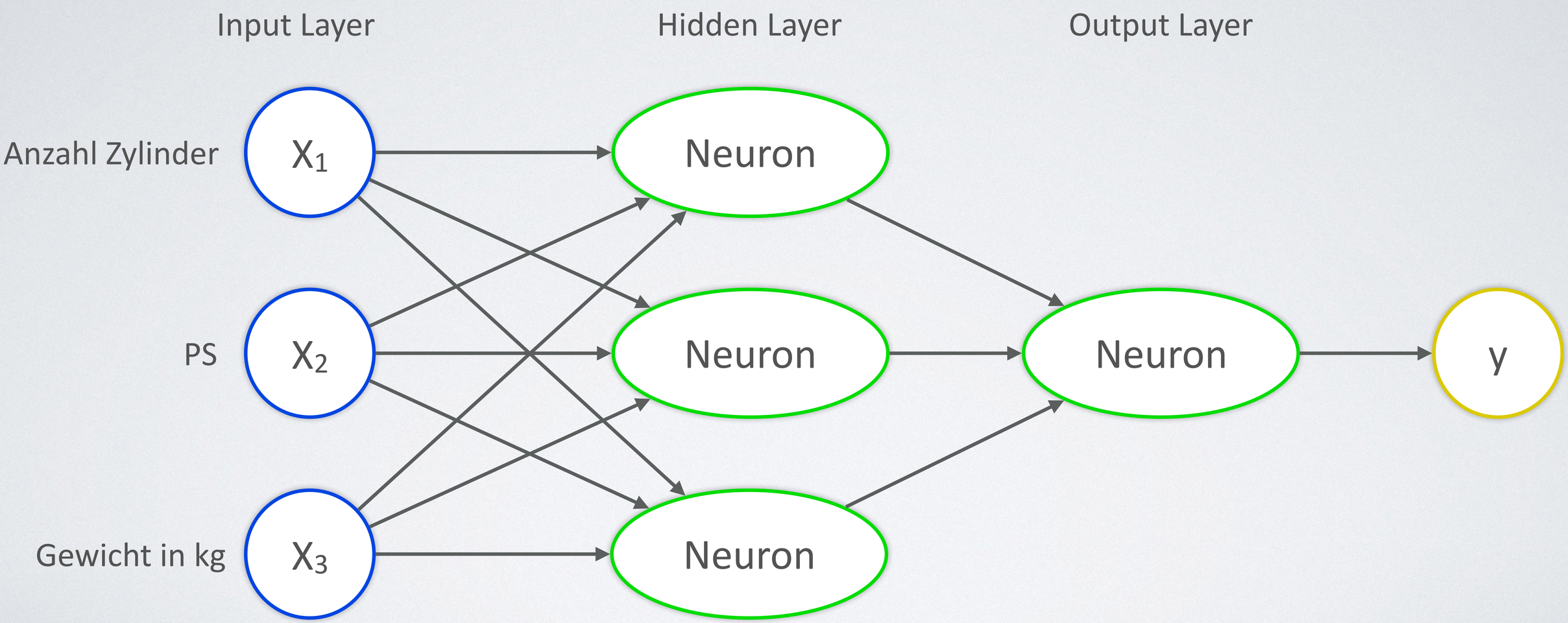
DROPOUT - MOTIVATION



DROPOUT - MOTIVATION

- **Preis für Auto:**
  - PS
  - Anzahl Zylinder
  - Gewicht in KG

NEURONALES NETZ



NEURONALE NETZE: TRAINING MIT EINER GPU

Deep Learning

TRAINING MIT EINER GPU

- **Problem:**
  - Neuronale Netze benötigen ziemlich viel Rechenleistung fürs Training
- **Lösung:**
  - **Wir trainieren auf einer Grafikkarte!**

TRAINING MIT EINER GPU

- Wir benötigen eine NVIDIA-Grafikkarte mit ausreichend Grafikspeicher
- Wenn du eine NVIDIA-GPU hast, kannst du Keras mit deiner lokalen Grafikkarte betreiben
- Die Einrichtung ist aber durchaus aufwendig
- **Alternativ:** GPU in der Cloud!

GPU IN DER CLOUD

- Für wenige Dollar pro Stunde kannst du dir in der Amazon-Cloud einen Computer mit einer kräftigen Grafikkarte mieten
- **Sehr empfehlenswert:**
  - Ausreichend Arbeitsspeicher
  - Sehr schnelle GPU
  - Ausreichend Grafikspeicher!

GPU IN DER CLOUD

- **Vorbereitung:**
  - Wir schauen uns ein paar Linux-Grundlagen an
  - Und anschließend mieten wir uns eine Instanz in der Amazon-Cloud!
  - Damit kannst du auch komplexe Netze trainieren

GPU IN DER CLOUD

- **Wichtig:**
  - Wenn später im Kurs das Training eines Netzes dir zu lange dauert
  - => Trainiere dieses Netz auf einer Grafikkarte, um die Trainingszeit zu verkürzen!

NEURONALE NETZE:
TRAINING MIT EINER GPU IN DER GOOGLE-CLOUD

Deep Learning

GPU IN DER GOOGLE-CLOUD

- **Google Colab:**
  - Google stellt uns eine kostenlose Jupyter-Notebook-Umgebung zur Verfügung
  - Inklusive GPU-Support!

GPU IN DER GOOGLE-CLOUD

- **Google Colab - Limitierungen:**
  - Du brauchst ein Google-Account
  - Du hast keinen „Anspruch“ darauf, dass die Umgebung immer funktioniert - teils sind z.B. keine GPUs mehr verfügbar
  - Richtig aufwendige Modelle können eingeschränkt werden

NEURONALE NETZE: CONVOLUTIONAL NEURAL NETWORKS (CNN) MOTIVATION

Deep Learning

CNN - MOTIVATION

- Grunsätzlich könnte unser Neuronales Netz auch jetzt schon ausgeben, was auf einem Bild zu sehen ist
- Aber: Das Netz hat kein „Verständnis“ davon, wie ein Bild aufgebaut ist
- Dieses Verständnis kann aber nützlich sein!

car 1.00
person 1.00
Fibank
OK
9732121
VISA

PROJEKT: BILDERKENNUNG AUTOS

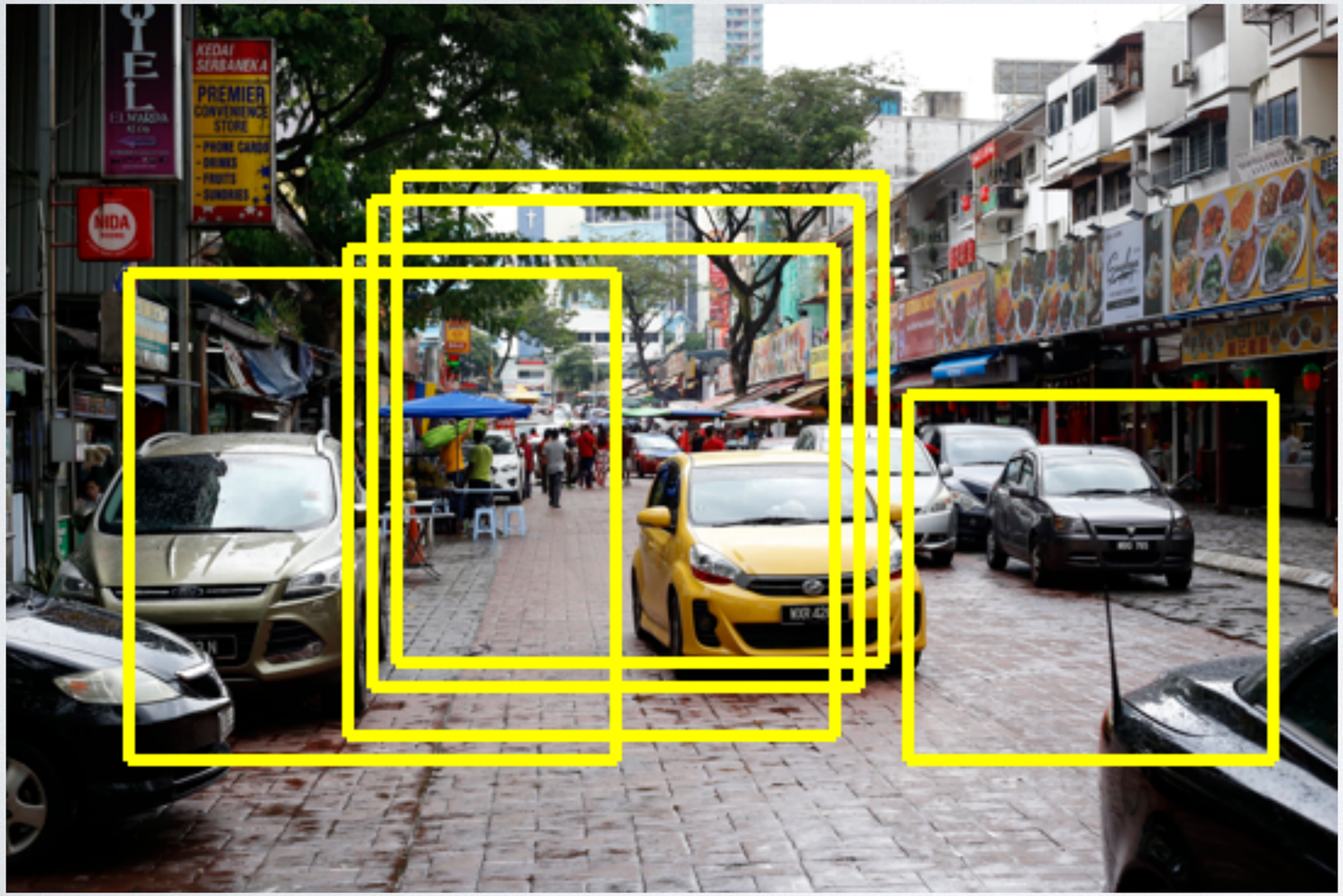
Deep Learning

PROJEKT: BILDERKENNUNG AUTOS

- Dieses Projekt besteht aus 2 Teilen
- **Teil 1:** Wir trainieren ein Neuronales Netz
- **Teil 2:** Wir wenden dieses Netz an, und finden das Auto

PROJEKT: BILDERKENNUNG AUTOS

PROJEKT: BILDERKENNUNG AUTOS



NEURONALE NETZE:
YOLO (YOU ONLY LOOK ONCE)

Deep Learning

car 1.00
person 1.00
Fibank
OK
9732121
VISA

CONVOLUTIONAL NEURAL NETWORKS (CNN)

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

*

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

=

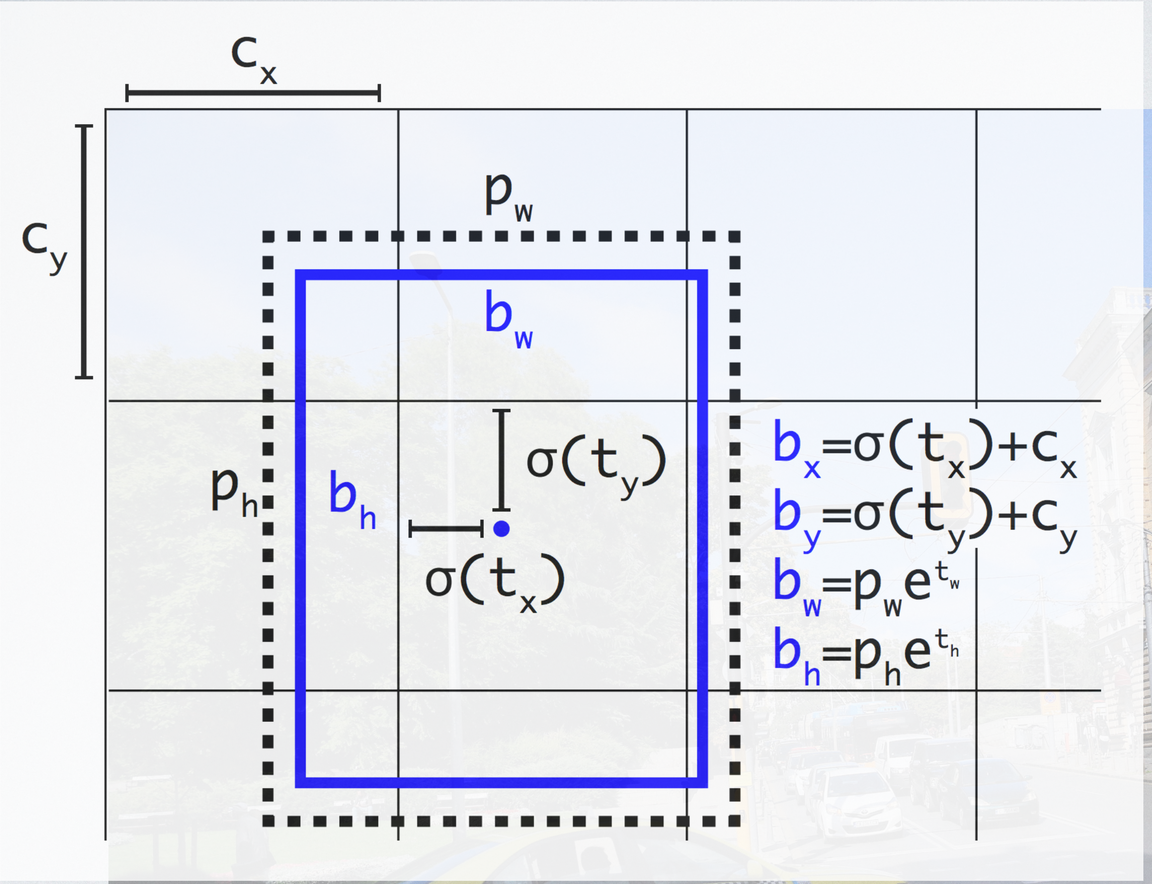
-2	2	2
-3	3	3
-2	2	2

*

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

=

…	…	…
…	…	…
…	…	…


c_x
c_y
p_w
b_w
p_h
b_h
$\sigma(t_y)$
$\sigma(t_x)$
$b_x=\sigma(t_x)+c_x$
$b_y=\sigma(t_y)+c_y$
$b_w=p_w e^{t_w}$
$b_h=p_h e^{t_h}$

NEURONALE NETZE: MODEL ZOO

Deep Learning

TRANSFER LEARNING

- **Problem:**
  - Das Training eines Neuronalen Netzes ist oft sehr aufwendig
  - Zudem benötigen wir sehr viele Daten, damit das Netz die korrekten Gewichte lernen kann

TRANSFER LEARNING

- **Idee:**
  - Wenn wir also zu wenig Daten haben:
  - Wir trainieren ein Neuronales Netz auf komplett anderen Daten
  - Und trainieren es erst dann auf unseren eigentlichen Daten

MODEL ZOO

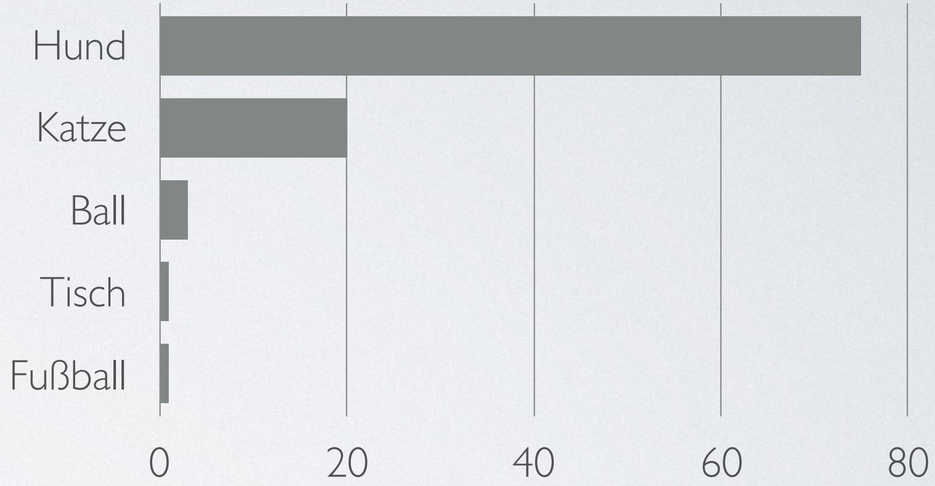
- **Idee:**
  - Fertig trainiere Modelle
  - Diese können wir direkt verwenden…
  - … oder abändern!

NEURONALE NETZE:
DIE IMAGENET-CHALLENGE

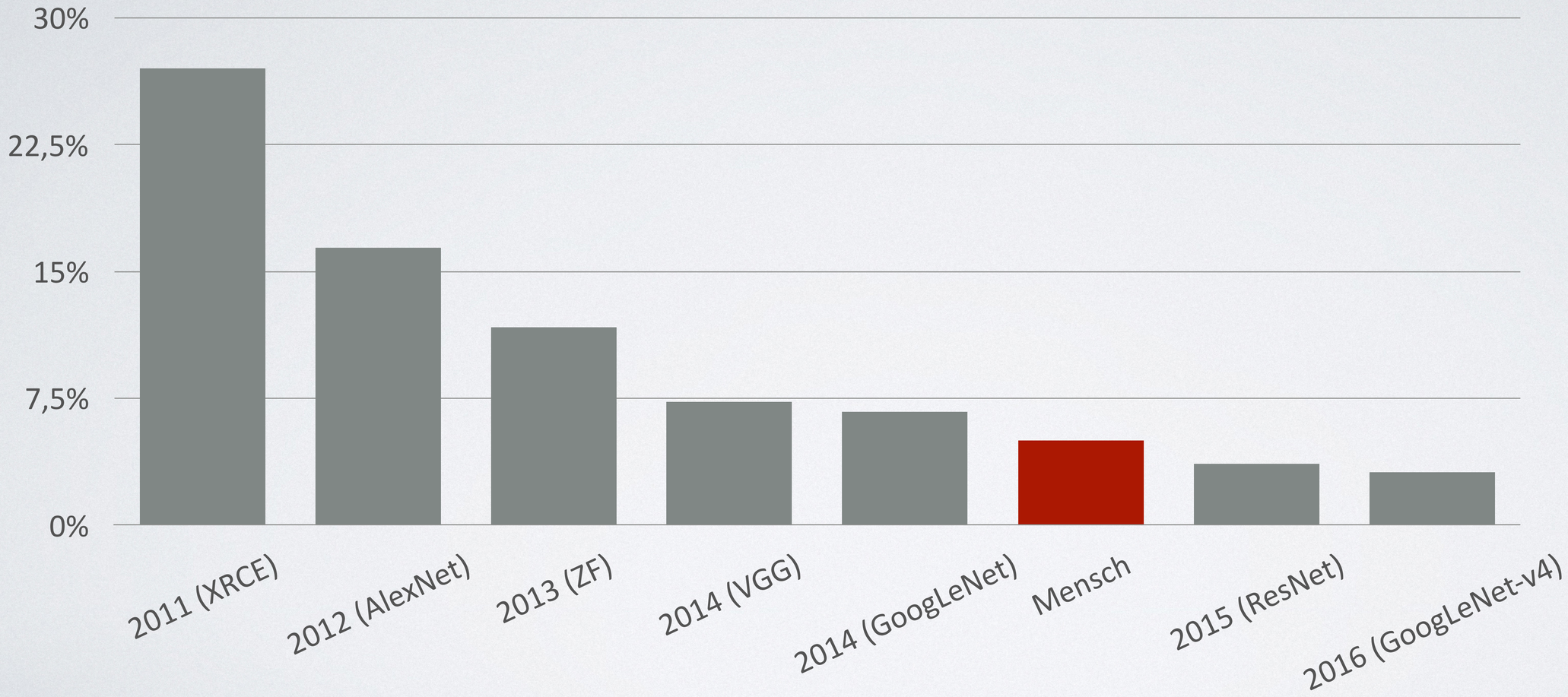
Deep Learning

IMAGENET-CHALLENGE

- **Aufgabe:**
  - Es gibt 1.000 Kategorien
  - Sage die richtige Kategorie vorher
    - Top 1 Score: Wie viele Prozent haben wir exakt richtig vorhergesagt?
    - Top 5 Score: Bei wie vielen Prozent war die richtige Kategorie in unseren Top 5 dabei?



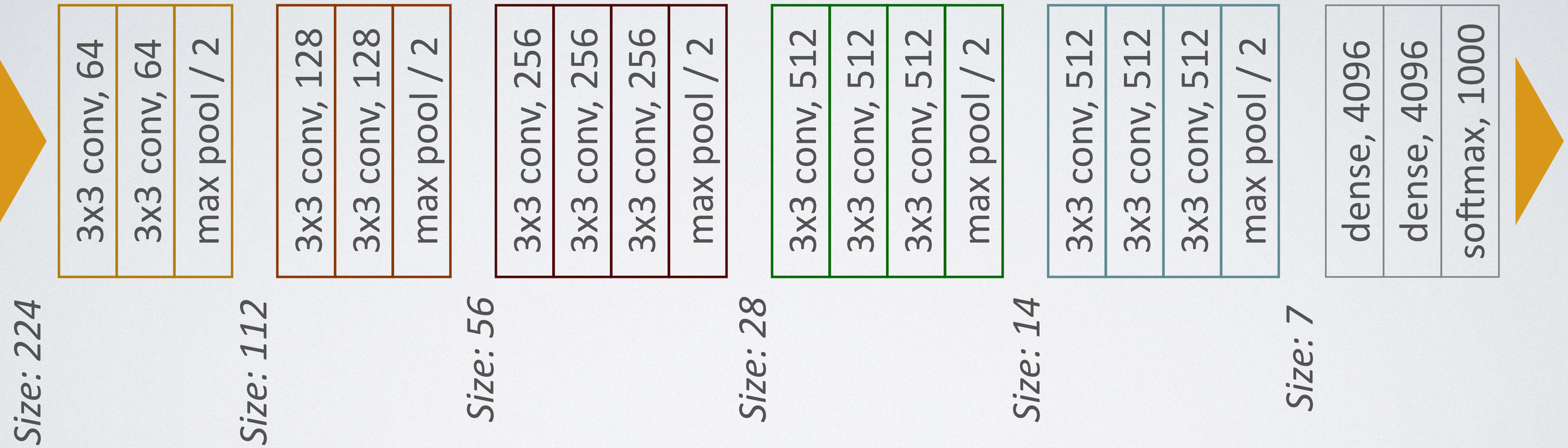
IMAGENET-CHALLENGE: TOP 1 - FEHLERRATE



VGG16

- **VGG16:**
  - Mehrere Convolutional-Layers & MaxPooling-Layers hintereinander
  - Fully Connected (Dense) Layer am Ende
  - Softmax zur Ausgabe der Kategorien
  - Größe der Gewichte: Ca. 530MB

VGG16: AUFBAU



NEURONALE NETZE: PROJEKT MODEL ZOO

Deep Learning

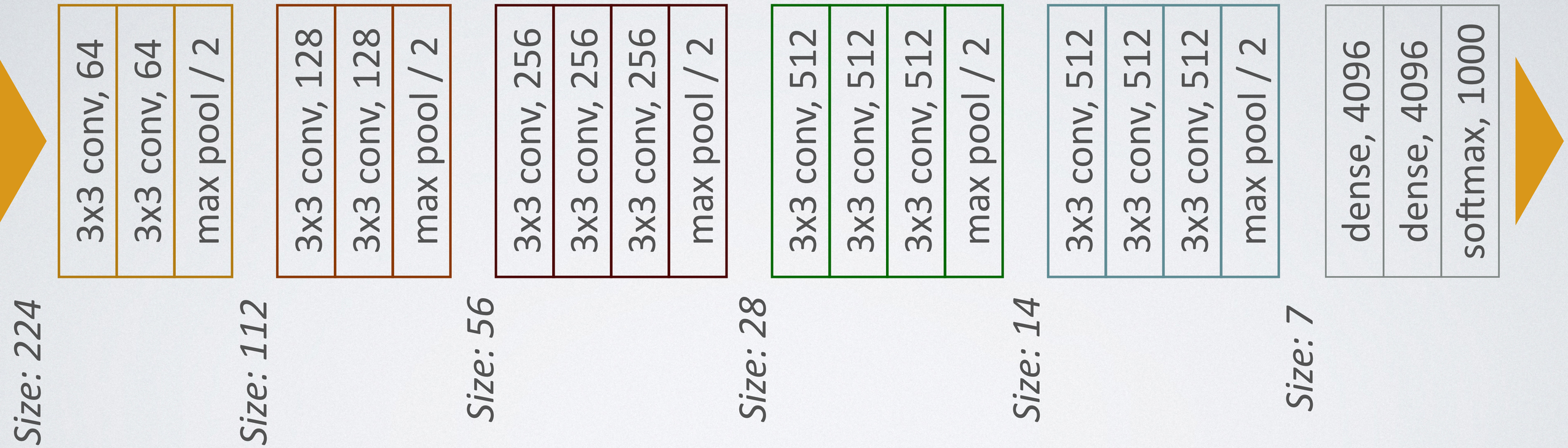
PROJEKT MODEL ZOO

- **Projekt:**
  - Wir möchten ein Netz trainieren, welches Hunde von Katzen unterscheiden kann
  - Unsere Datenbasis ist relativ beschränkt (ca. 600MB)
  - Das ist grundsätzlich erstmal schwierig…

PROJEKT MODEL ZOO

- **Idee:**
  - Transfer-Learning!
  - Das VGG16-Netz hat ja schon gelernt, Kanten / Objekte zu erkennen!
  - Wir können dies also für unser Modell weiterverwenden!

VGG16: AUFBAU



NEURONALE NETZE: AUTOENCODERS

Deep Learning

AUTOENCODER

- **Auto: selbst**
- **Encoder: kodieren**

AUTOENCODER

- **Idee:**
  - **Eingang vom Netz = Ausgang vom Netz**

AUTOENCODER: WARUM?

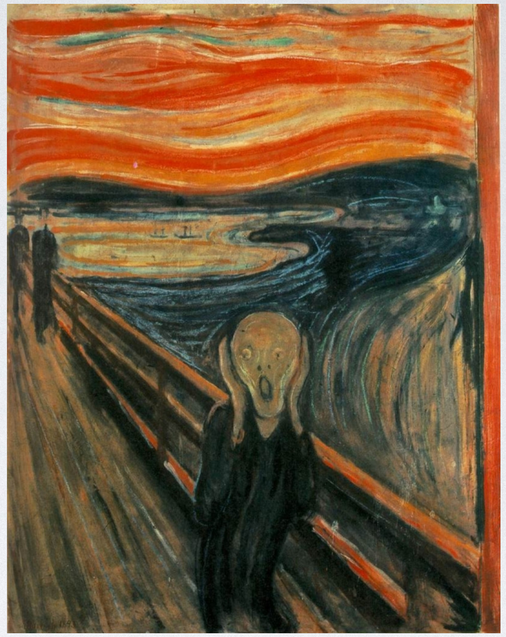
- **Daten komprimieren (Dimensionsreduktion)**
- **Entfernen von Bildrauschen**

NEURONALE NETZE: IMAGE STYLE TRANSFER

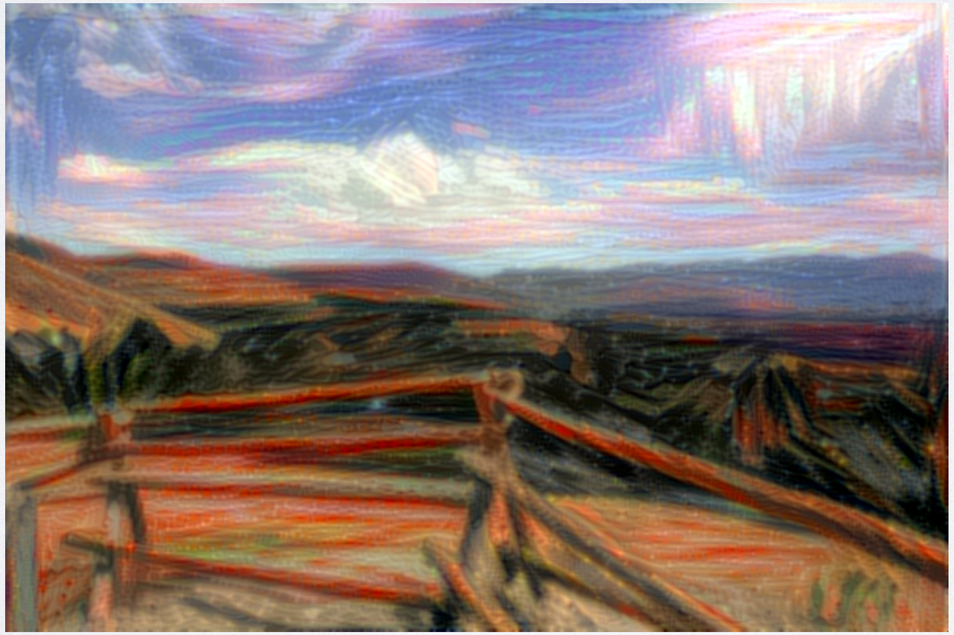
Deep Learning

Salt Lake City, Utah, 2018

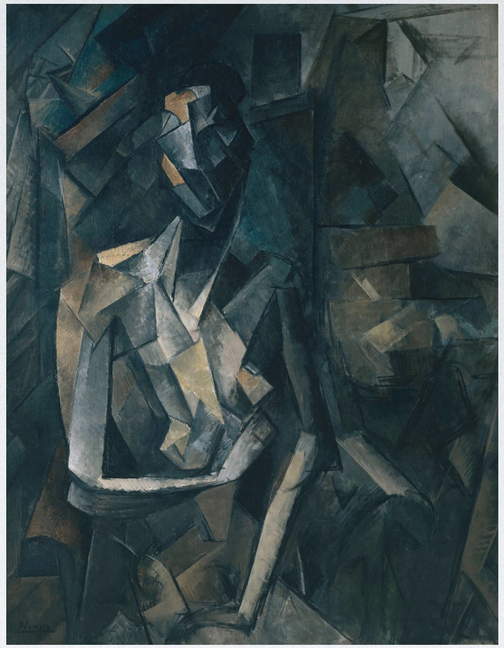
+



= ?



+



=

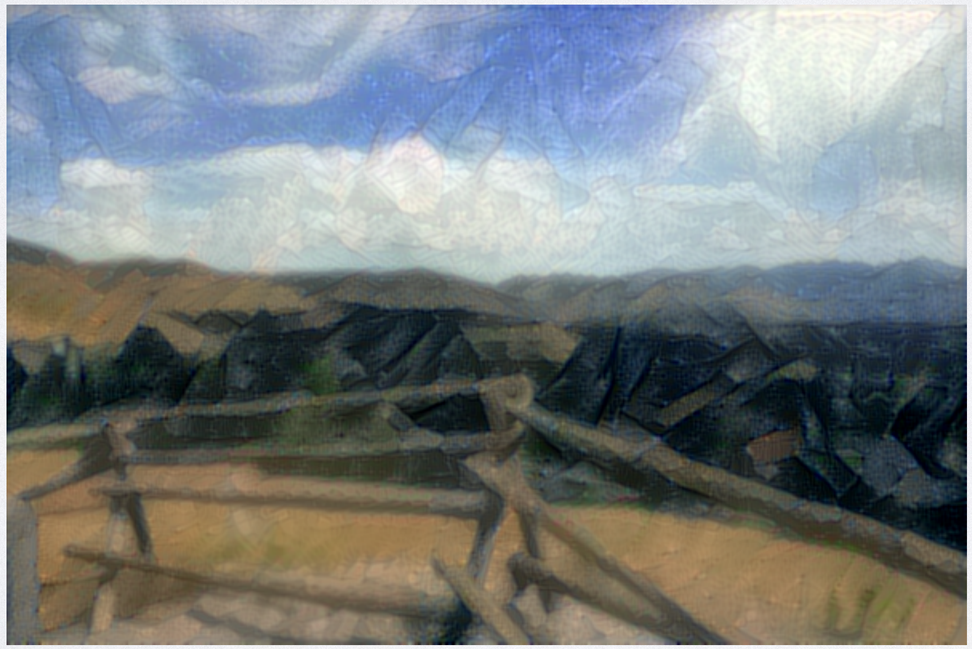
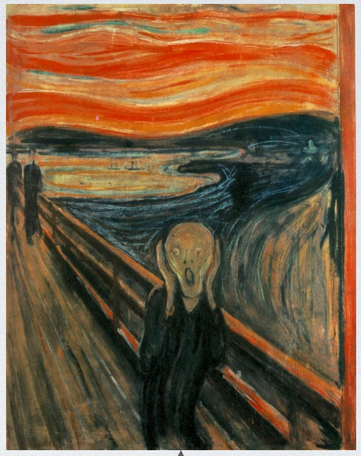


IMAGE STYLE TRANSFER

- **Idee: Neuronales Netz bekommt ein Bild übergeben**
- **Und versucht ein neues Bild zu erzeugen**

Vom Stil her ähnlich zu:



Style Transfer

Vom Inhalt ähnlich zu:

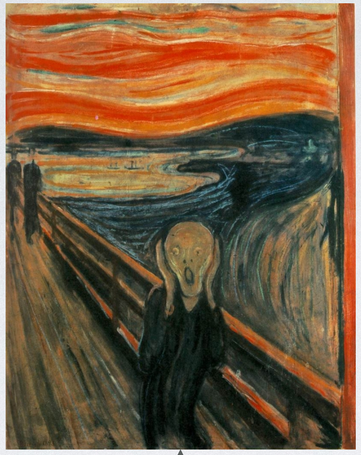
IMAGE STYLE TRANSFER

- **Das Netz für diesen Style Transfer ist etwas komplex**
- **Ich fand' das Beispiel aber richtig cool, und wollte es unbedingt in diesem Kurs mit drinnen haben**
- **Wir werden uns daher grob anschauen, wie es funktioniert**

NEURONALE NETZE:
WIE FUNKTIONIERT ES GENAU?

Deep Learning

Vom Stil her ähnlich zu:



Neuronales Netz (VGG19)

Vom Inhalt ähnlich zu:

IMAGE STYLE TRANSFER

- **Wichtiger Unterschied:**
  - Hier möchten wir also keine Gewichte vom Neuronen Netz optimieren
  - Sondern die Eingabe, also das, was wir ins Netz hineinschicken
  - Das macht Keras für uns leider nicht automatisch
  - Dazu müssen wir direkt das Backend ansteuern

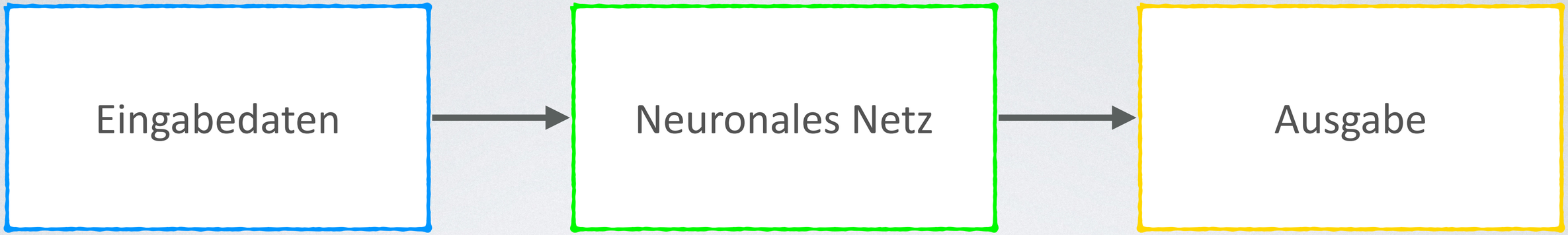
NEURONALE NETZE:
RECURRENT NEURAL NETWORKS

Deep Learning

RECURRENT NEURAL NETWORKS

- **Anwendungen:**
  - Text- und Sprachgenerierung
  - Übersetzung von Sätzen (z.B. „Google Translate“)
  - Google Duplex
  - Vorhersagen von Aktienkursen

„NICHT-RNN“-NEURONALE NETZE



- **Problem:**
  - Wir haben keine Möglichkeit, die zeitliche Struktur zu modellieren
  - CNN: Geben dem Netz Intuition über „bildliche“ Struktur
  - RNN: Geben dem Netz Intuition über „zeitliche“ Struktur

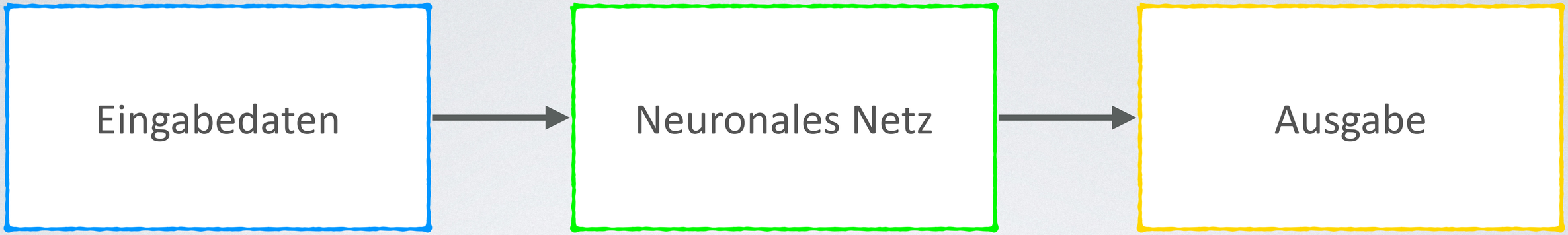
RECURRENT NEURAL NETWORKS

- **Allerdings:**
  - Klassische RNNs haben ein mathematisches Problem, weswegen sie nicht so gut einsetzbar sind
- **Wir schauen uns daher an:**
  - Theorie hinter RNNs
  - Das mathematische Problem („Exploding / Diminishing Gradients“)
  - Lösung dazu: LSTM

NEURONALE NETZE:
RECURRENT NEURAL NETWORKS

Deep Learning

„NICHT-RNN“-NEURONALE NETZE



- **Problem:**
  - Wir haben keine Möglichkeit, die zeitliche Struktur zu modellieren
  - CNN: Geben dem Netz Intuition über „bildliche“ Struktur
  - RNN: Geben dem Netz Intuition über „zeitliche“ Struktur

RECURRENT NEURAL NETWORKS

one to many

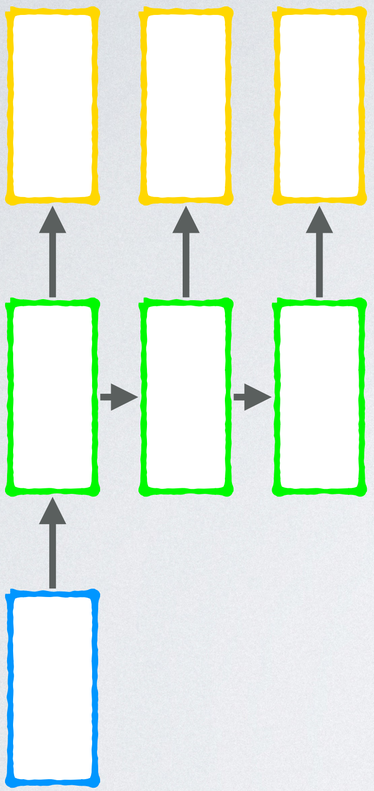
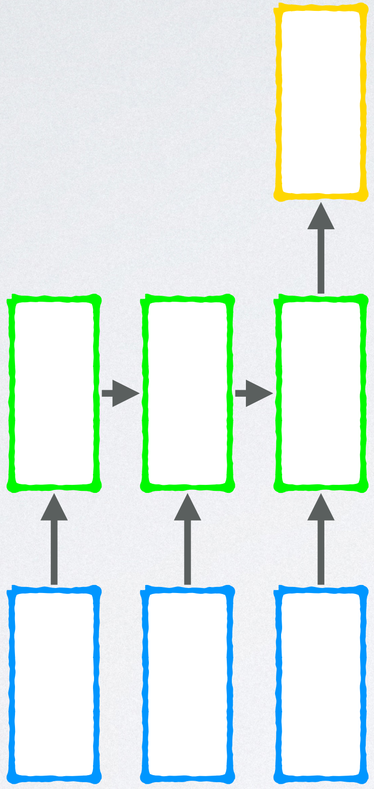


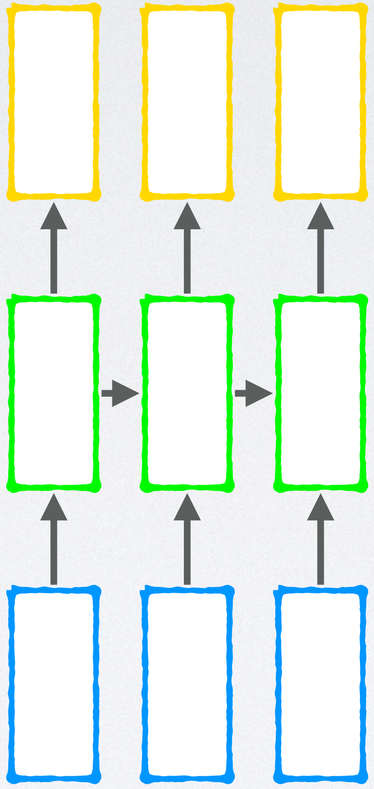
Image Captioning

many to one

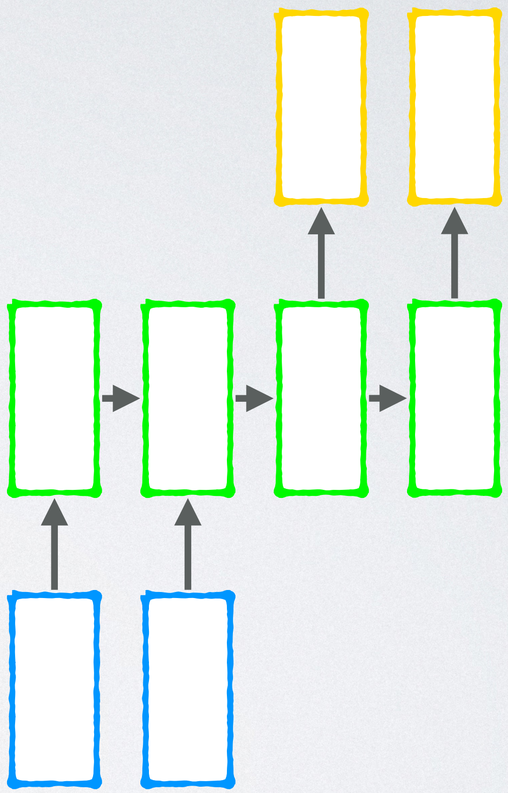


Text- / Video-Klassifizierung

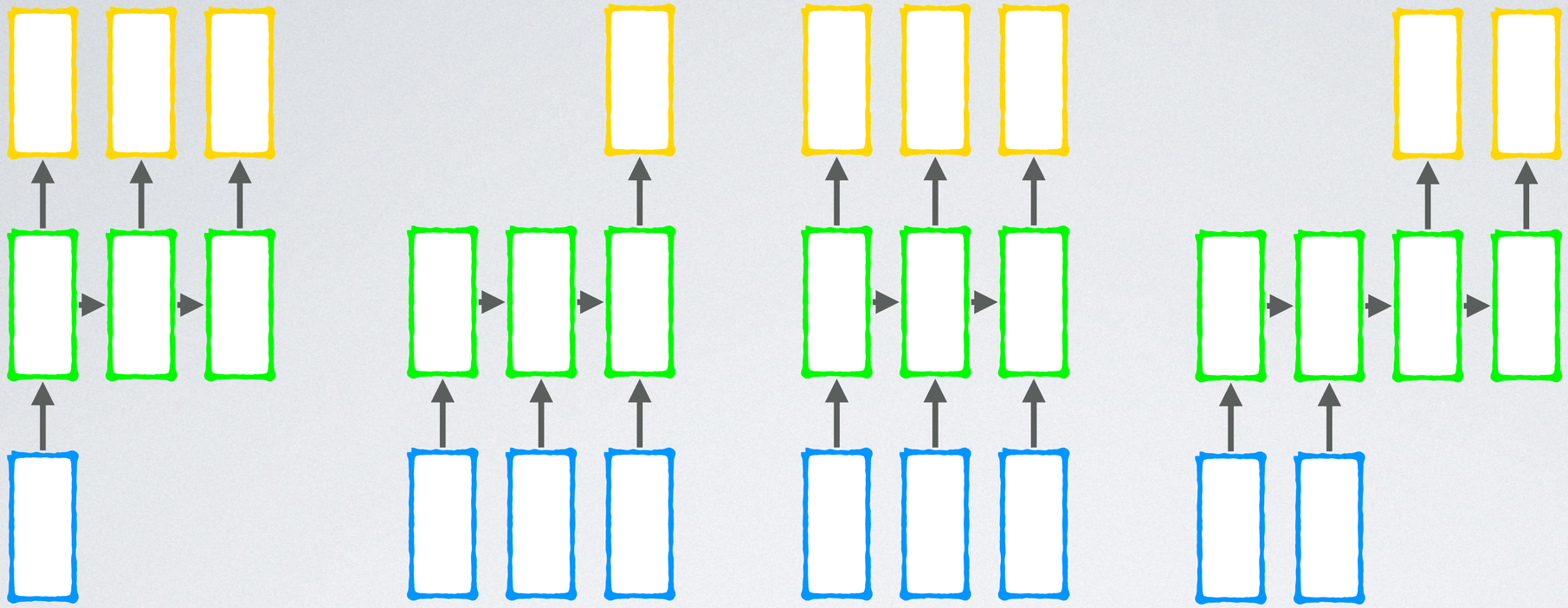
many to many



many to many



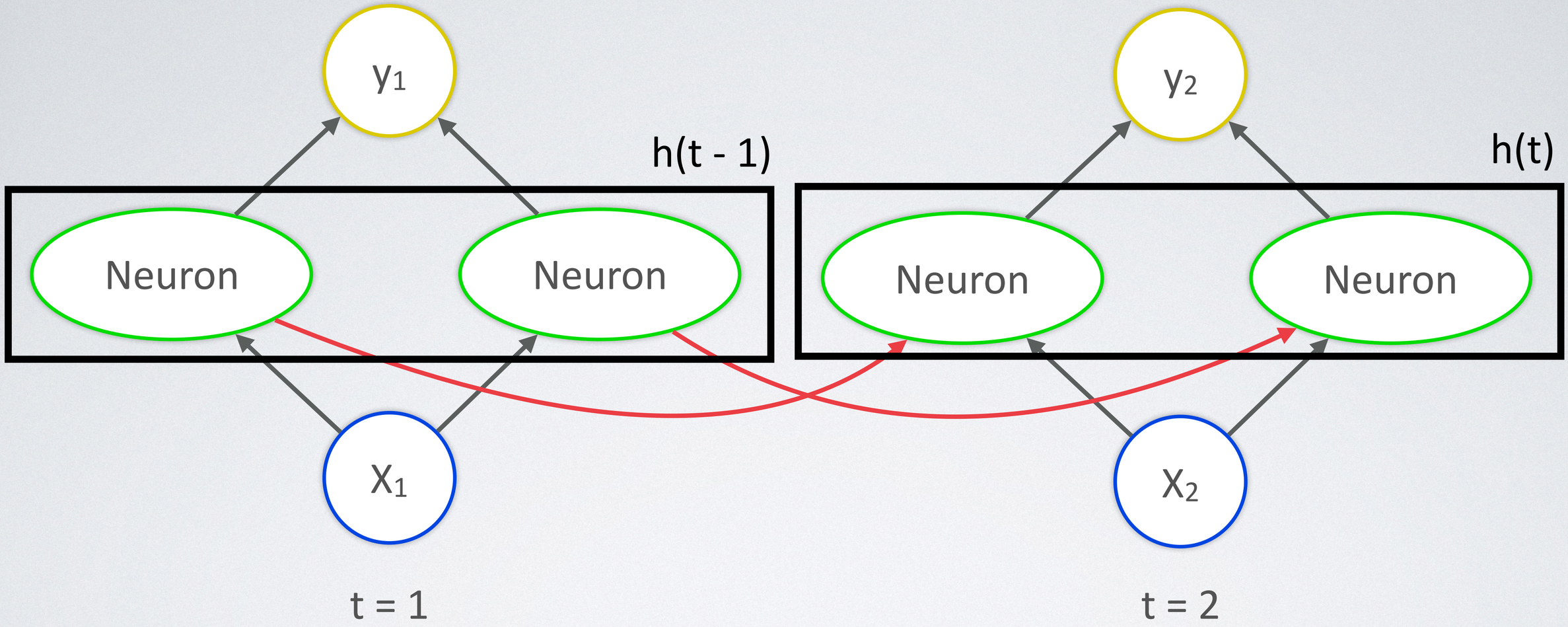
Übersetzung



- **Probleme bei „nicht-RNN“-Neuronalen Netzen:**
  - Feste Ein-bzw. Ausgabegröße
  - Annahme der Unabhängigkeit verschiedener Datenpunkte

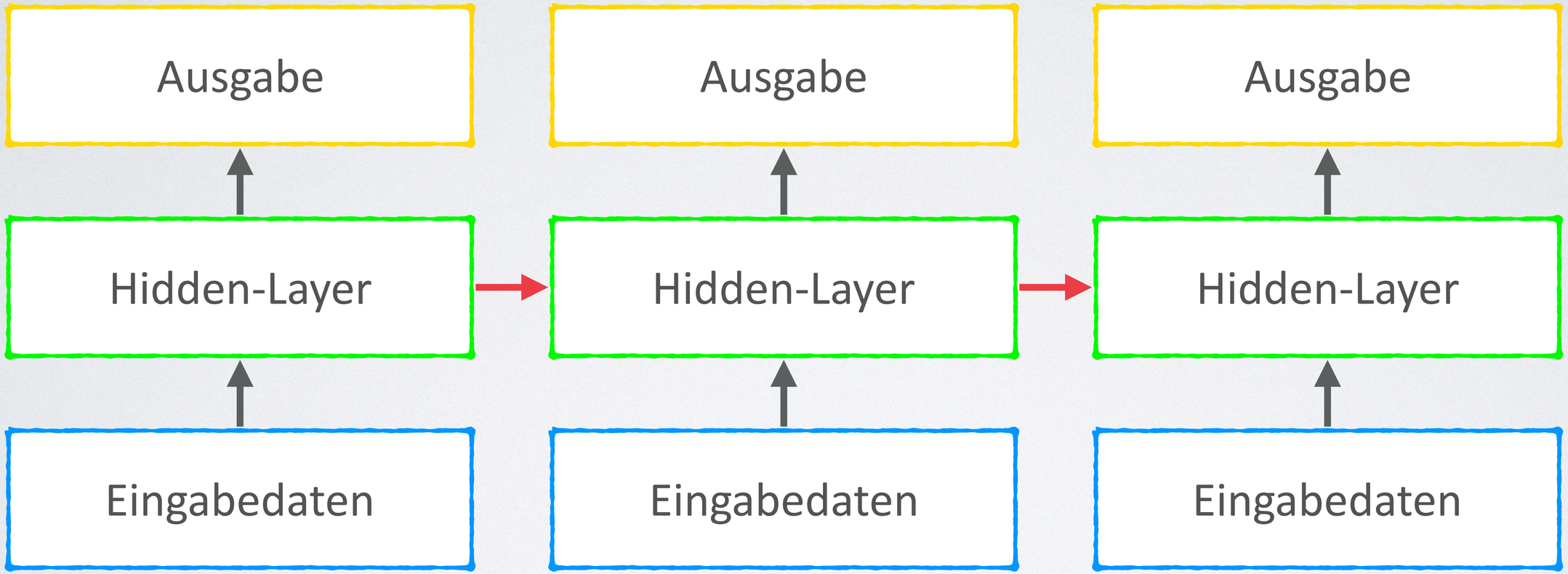
NEURONALE NETZE:
WIE IST EIN RNN AUFGEBAUT?

Deep Learning



$$h^{(t)} = \sigma(W^{(x)} \cdot x_{(t)} + W^{(h)} \cdot h^{(t-1)})$$

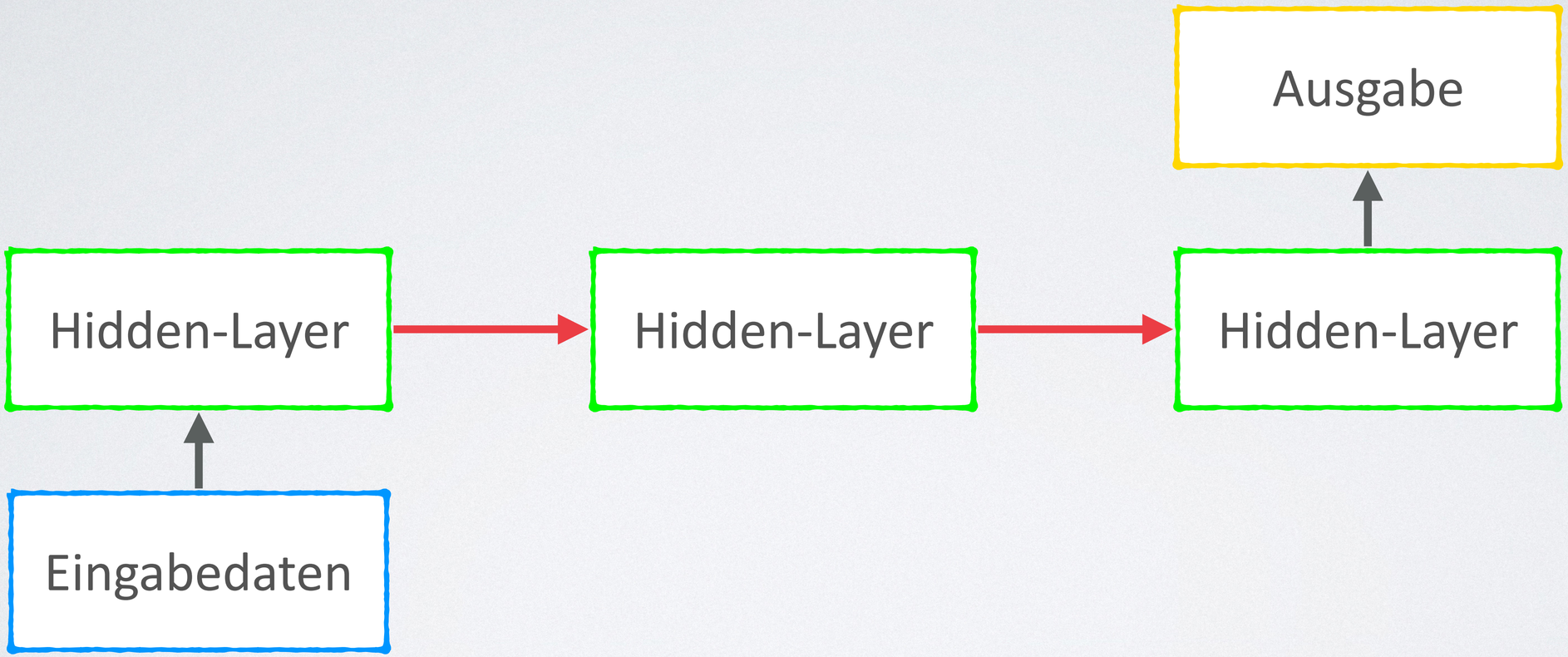
RNN - AUFBAU + BACKPROPAGATION



NEURONALE NETZE: DIMINISHING GRADIENTS?

Deep Learning

DIMINISHING GRADIENTS?



DIMINISHING GRADIENTS

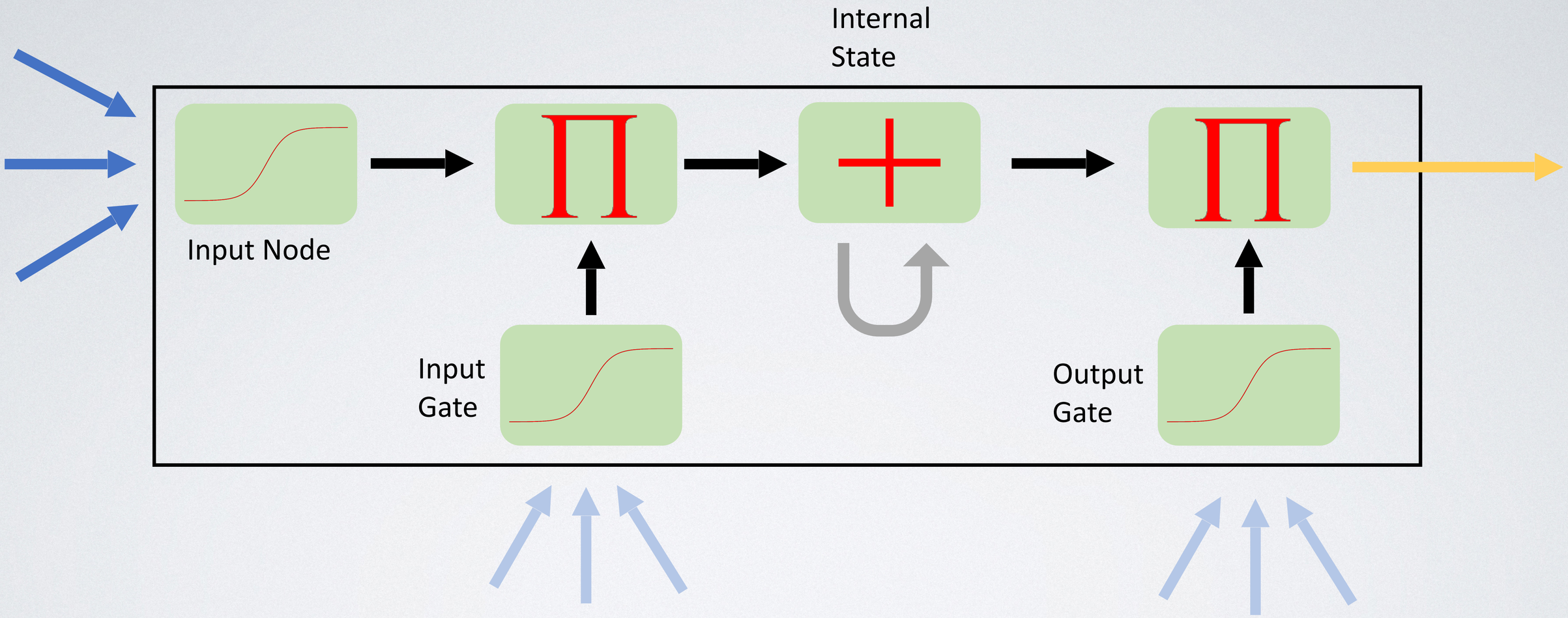
- **Problem:**
  - Bei einem klassischen RNN wird es sehr schwierig / unmöglich, Gewichte der vorherigen Zeitschritte anzupassen
- **Lösung:**
  - LSTM („Long Short-Term Memory“)

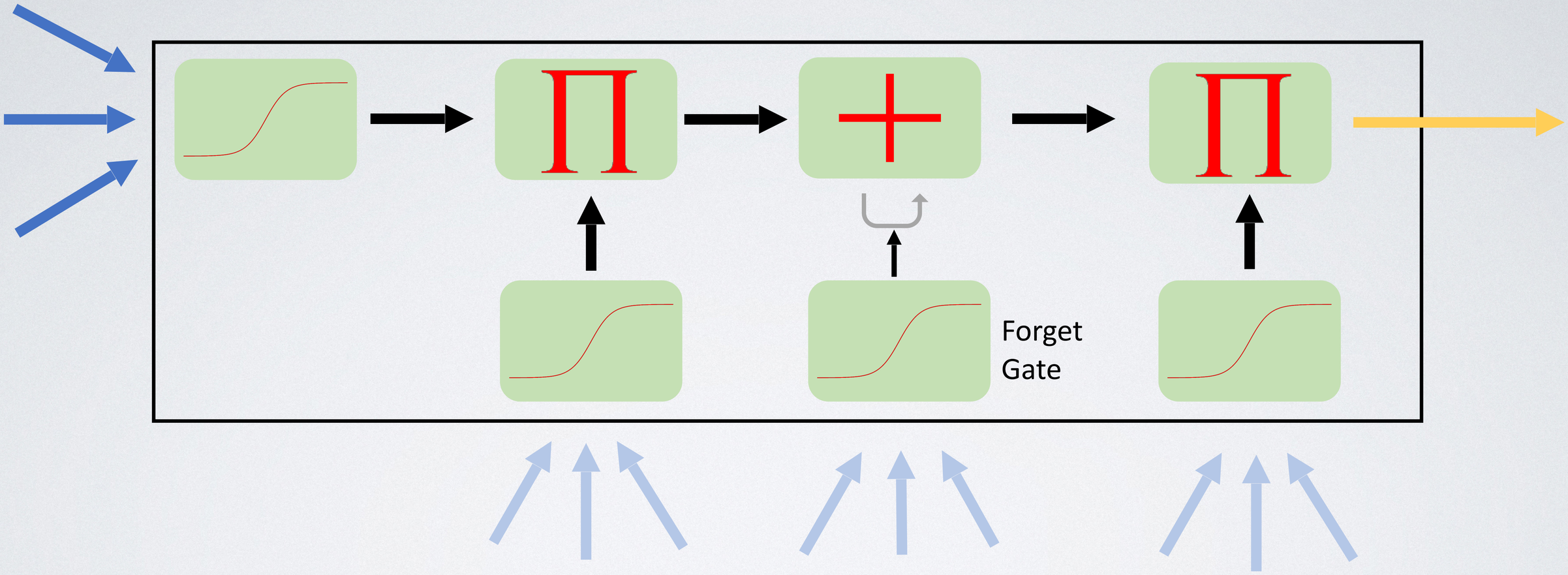
NEURONALE NETZE:
LSTM („LONG SHORT-TERM MEMORY“)

Deep Learning

LSTM

- **Ein ziemlich komplexer Aufbau**
- **Aber:**
  - Damit wird das Vanishing-Gradient-Problem gelöst
  - Und wir können unser RNN endlich sauber trainieren :)


Internal
State
Input Node
Input
Gate
Output
Gate


Forget
Gate

NEURONALE NETZE:
WIE MODELLIEREN WIR WÖRTER?

Deep Learning

WORD2VEC

- **Problem:**
  - Wir können schwer 10.000+ Wörter modellieren
  - Können wir die Informationen irgendwie für unser Netz aufbereiten?

WORD2VEC - BEISPIEL

- **Viele Wörter sind sehr ähnlich:**
  - Mutter
  - Mutti
  - Mama
  - (Erziehungsberechtigte)

WORD2VEC

- **Idee:**
  - Können wir diese Informationen nicht irgendwie vorab für unser Netz modellieren? (Word2vec)
  - Oder unserem Netz „Intuition“ über die Möglichkeit von Wortähnlichkeiten geben? (Embedding-Layer)

VORBEREITUNG: ONE-HOT-ENCODING

- **Wenn wir 3 verschiedene Klassen haben:**
  - 0 => [1, 0, 0]
  - 1 => [0, 1, 0]
  - 2 => [0, 0, 1]

NEURONALE NETZE:
WIE FUNKTIONIERT WORD2VEC?

Deep Learning

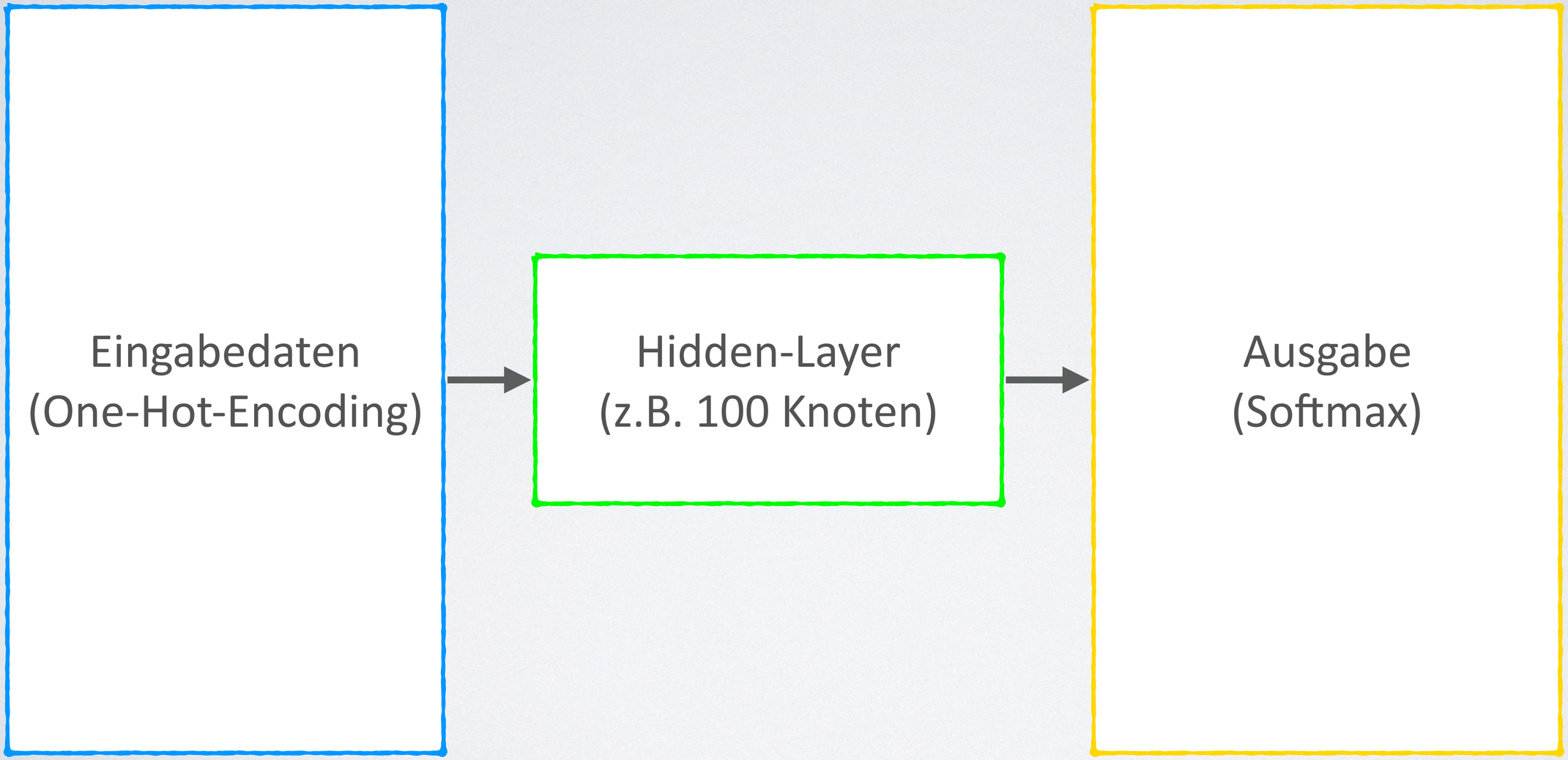
WORD2VEC - IDEE

- **Wir versuchen vorherzusagen, welche Wörter kommen bei einem Wort „in der Nähe“ vor?**

WORD2VEC - BEISPIEL

- **Window-Size: 2**
- **„Als Gregor Samsa eines Morgens aus unruhigen“**
  - (Als, Gregor), (Als, Samsa)
  - (Gregor, Als), (Gregor, Samsa), (Gregor, eines)
  - (Samsa, Als), (Samsa, Gregor), (Samsa, eines), (Samsa, Morgens)
  - (eines, Gregor), (eines, Samsa), (eines, Morgens), (eines, aus)
  - …

WORD2VEC - TRAINING VOM NETZ



NEURONALE NETZE: EMBEDDING LAYER

Deep Learning

AUFBAU DIESES ABSCHNITTS

- **Aufbau:**
  - Wir schauen uns den „Embedding-Layer“ von Keras an
  - Anschließend trainieren wir ein Neuronales Netz!

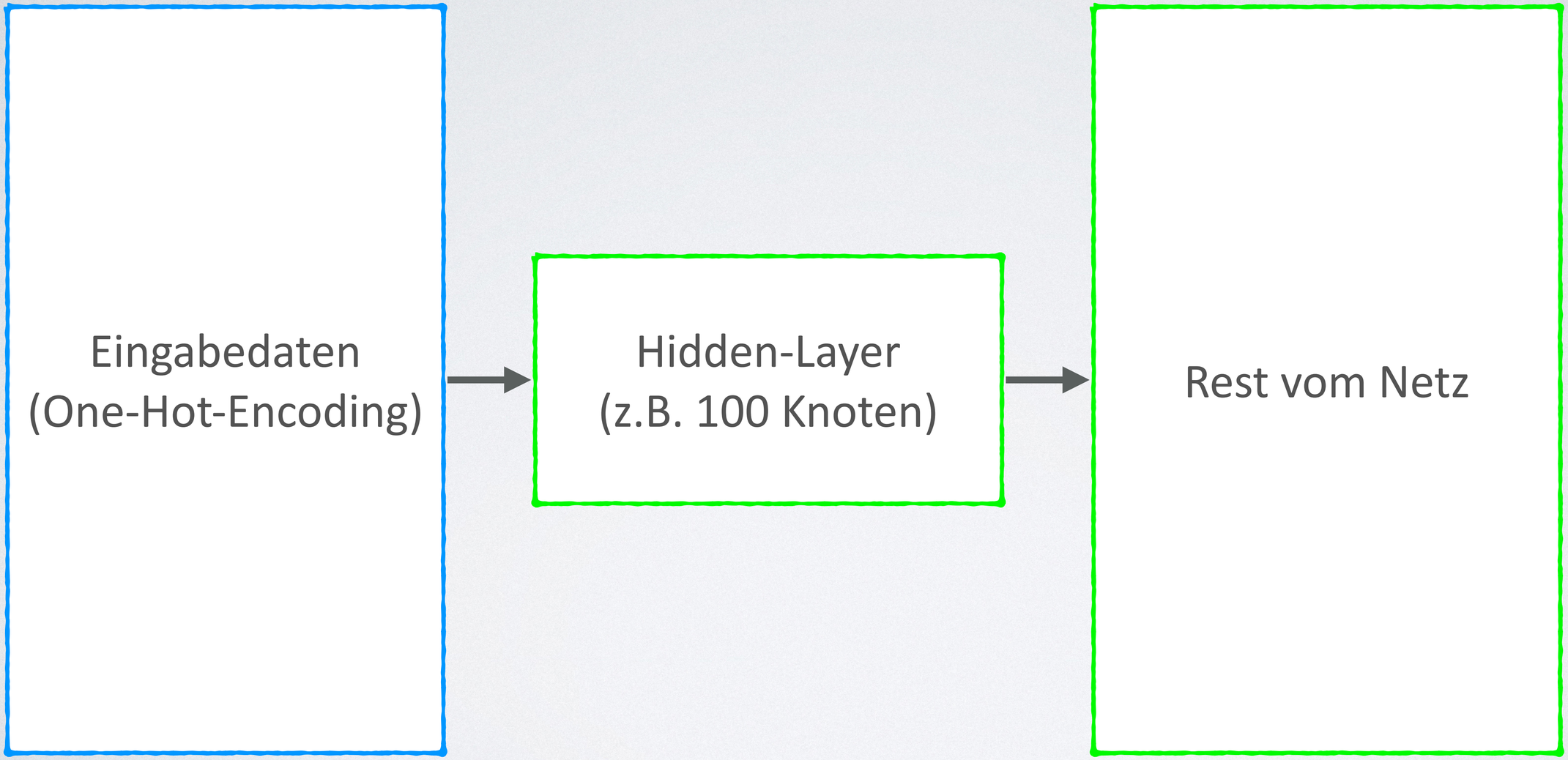
EMBEDDING-LAYER IN KERAS

- **Idee:**
  - Wir definieren kein „fixes“ Mapping wie in Word2Vec
  - Sondern geben unserem Neuronalen Netz die Möglichkeit, diesen Zusammenhang zu lernen

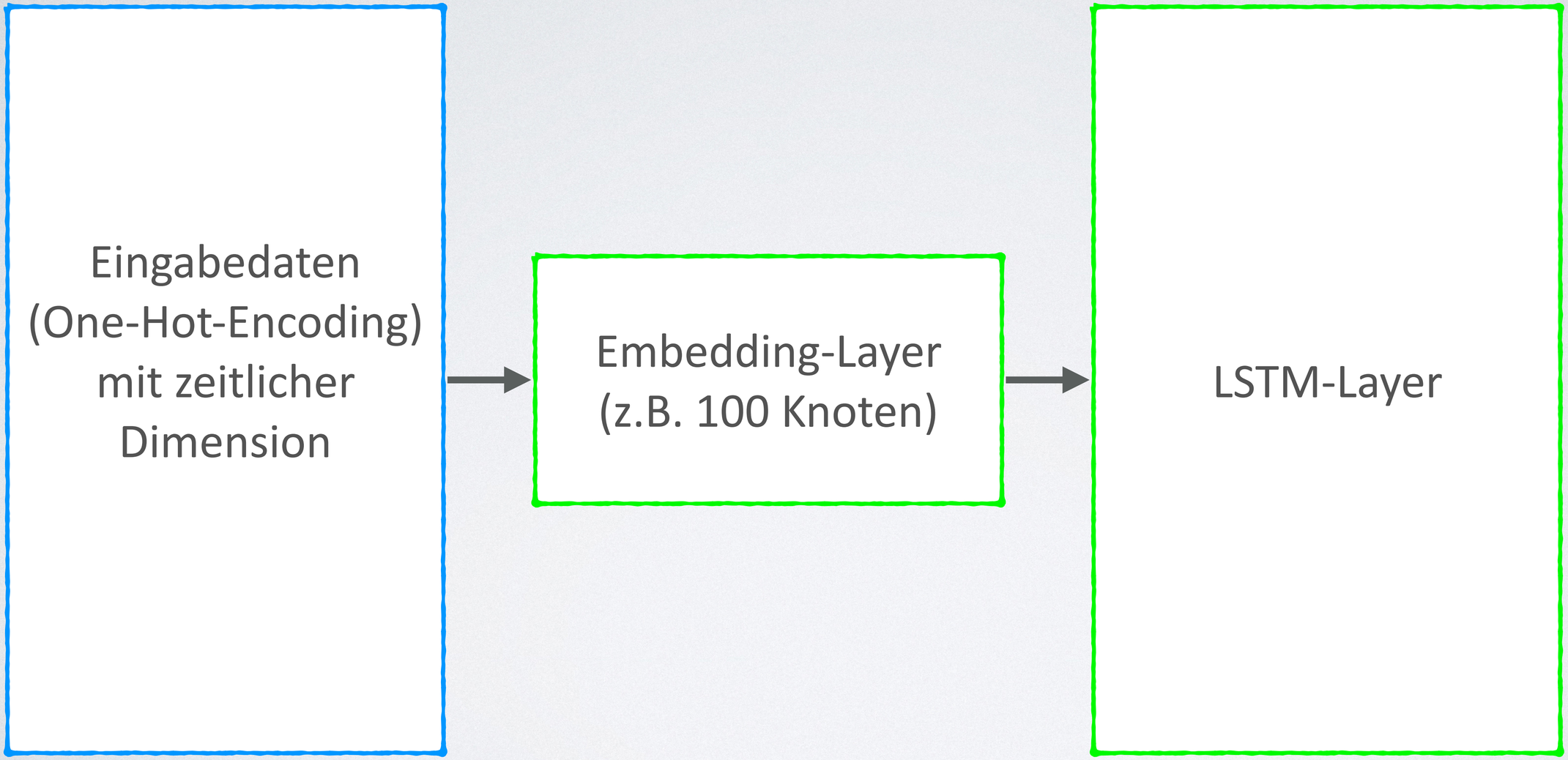
NEURONALE NETZE:
DER EMBEDDING-LAYER VON KERAS

Deep Learning

WARUM BRAUCHEN WIR DEN EMBEDDING-LAYER?



WARUM BRAUCHEN WIR DEN EMBEDDING-LAYER?



NEURONALE NETZE: WIE GEHT ES WEITER?

Deep Learning

MODELL AUF EINER WEBSEITE EINBINDEN

- **Modell optimieren**
  - Abwägung zwischen: Komplexität, Genauigkeit, Ladezeit!
- Berechnung in WebWorker auslagern
- Modell auf einem Server berechnen (AJAX?)
- Einbau von BabelJS (ES6)

NEURONALE NETZE
EXKURS WEBENTWICKLUNG?

Deep Learning

NEURONALE NETZE: EXKURS WEBENTWICKLUNG?

- **Grundsätzlich:**
  - Wenn du schonmal irgendwas mit Webentwicklung gemacht hast - du solltest diesen Exkurs überspringen können
  - Nur wenn du noch nie eine Webseite / HTML entwickelt hast, kann dieser Exkurs dir einen guten „Schnelleinstieg“ bieten

Q-LEARNING

Deep Learning

Q-LEARNING

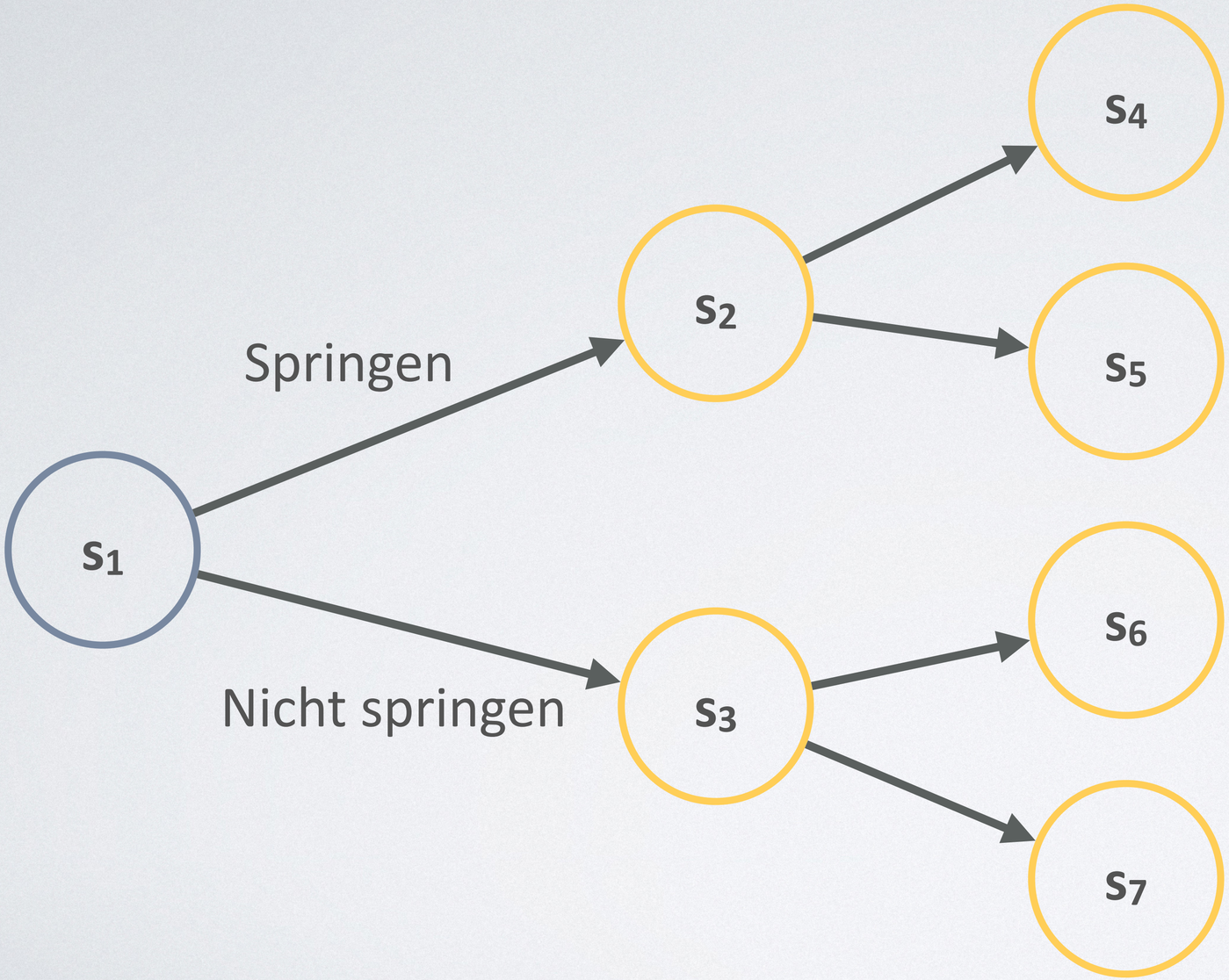
- **Reinforcement Learning**
- **Ziel:**
  - Wir lernen Verhaltensregeln, die einem Agenten (automatischen Spieler) den größtmöglichen Nutzen bringen
- Warum Q-Learning?
  - Q: Zustand (s) + Aktion (a)!

Q-LEARNING: EIN ZUSTAND Q

- **Q: Zustand (s)**
  - Die gesamte „Information“ über die Spielwelt
  - Wo ist der Spieler?
  - Welche Gegner sind um ihn herum? Wo sind diese?
  - Welche Eigenschaften hat der Spieler?

Q-LEARNING: Q

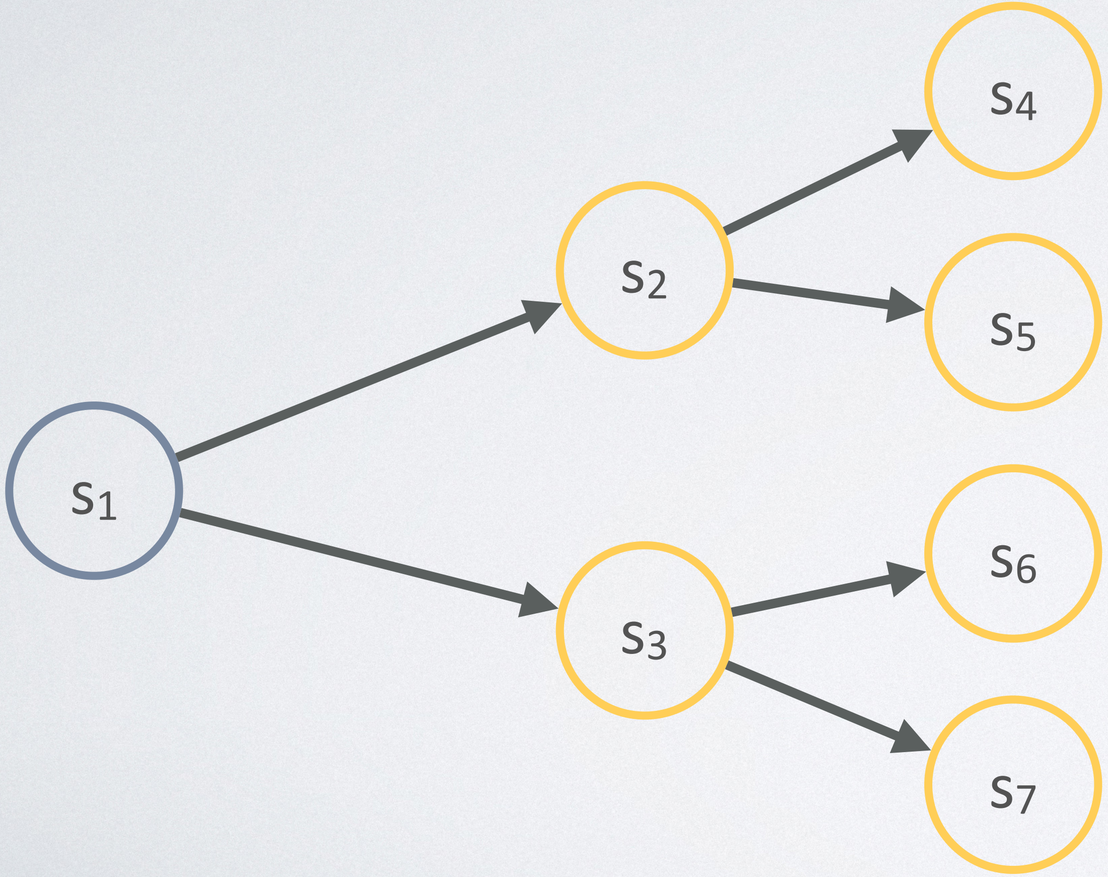
- **Q: Zustand (a)**
  - Welche Aktion wählen wir?


s1
Springen
Nicht springen
s2
s3
s4
s5
s6
s7

Q-LEARNING

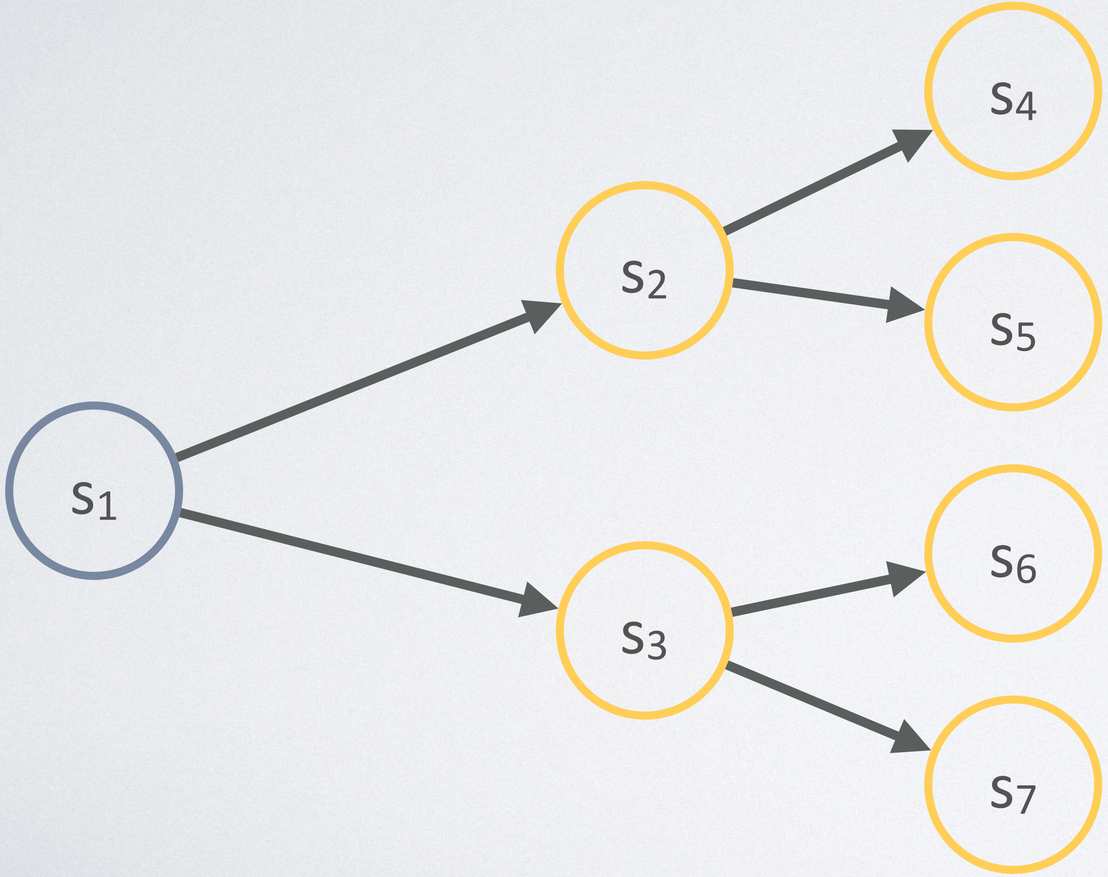
Deep Learning

$$Q^{new}(s_t, a_t) \leftarrow \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{\text{learned value}}$$



$$Q^{new}(s_t, a_t) \leftarrow$$

$$(1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{\text{learned value}}$$



SCHLUSS + WIE GEHT ES WEITER?

Deep Learning

VIELEN DANK, DASS ICH DIR DEEP LEARNING & AI BEIBRINGEN DURFTE! :)

Deep Learning & AI

WAS HAST DU ALLES GELERNT?

- **Neuronale Netze**
  - Bilderkennung (CNN)
  - Neural Image Style Transfer
  - Vorhersage von zeitlichen Daten (z.B. Aktienkursen)
  - Textgenerierung
- **Q-Learning**

WIE GEHT ES JETZT WEITER?

- **Machine Learning**
- **Eigene Modelle entwickeln**

WO BEKOMMST DU DATEN HER?

- **kaggle.com**
- **toolbox.google.com/datasetsearch**
- **gym.openai.com**