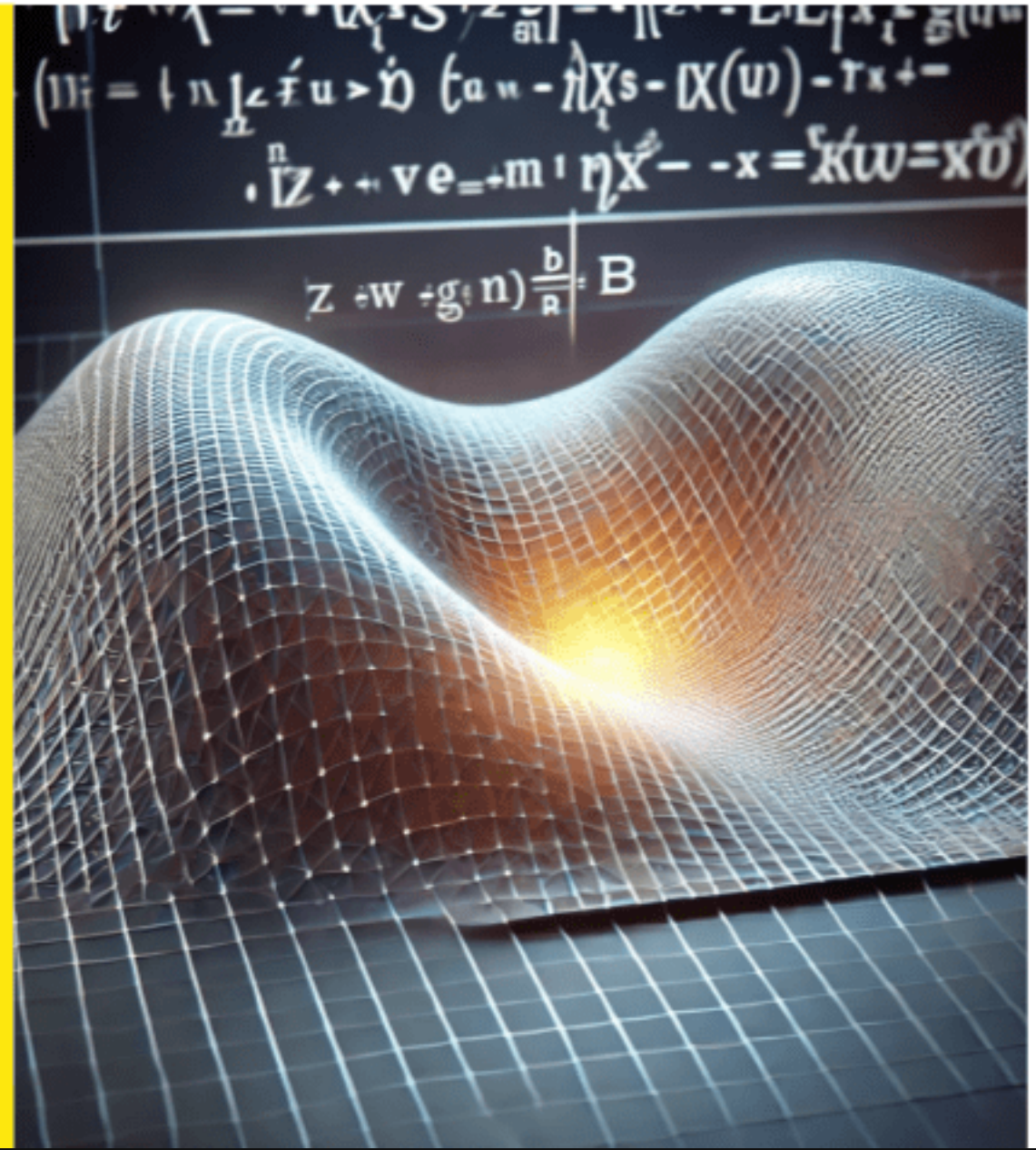


# Deep Learning Grundlagen

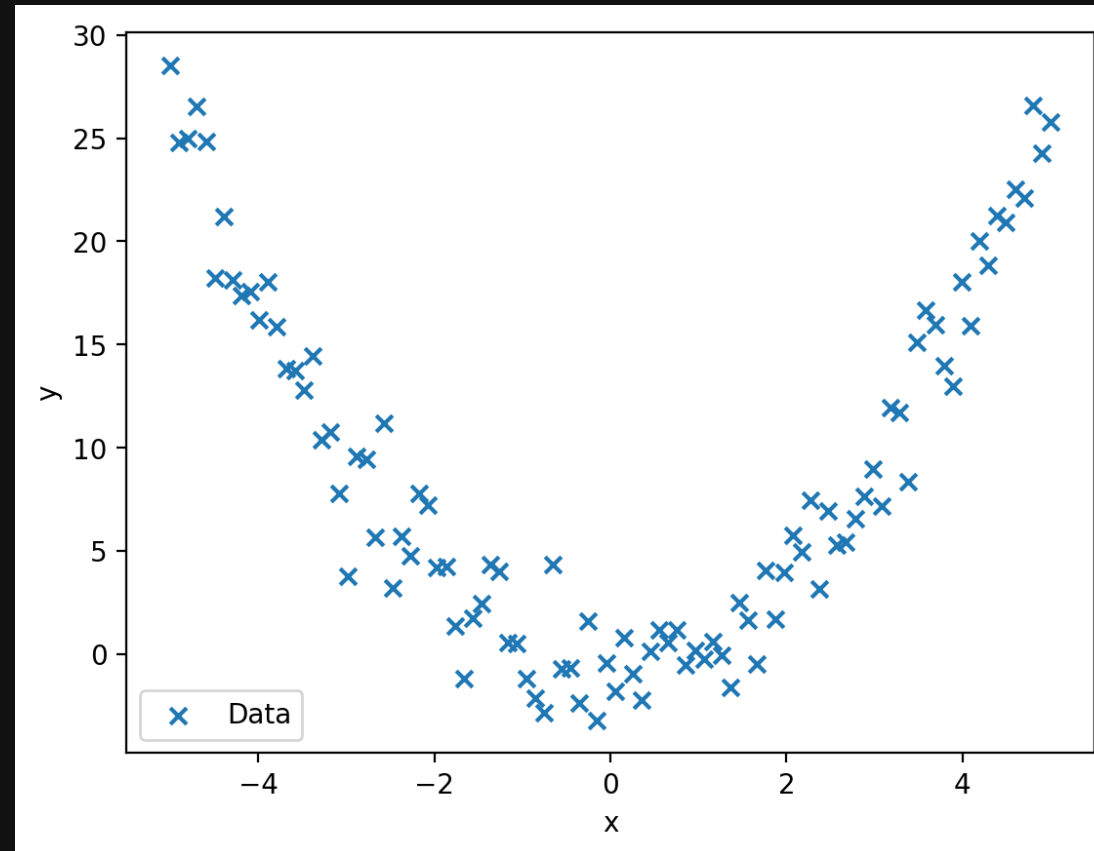
Dr. Stefan Hackstein  
stefan.hackstein@fhnw.ch



# Machine Learning

<https://www.youtube.com/embed/u3z3s3W3bF4?enablejsapi=1>

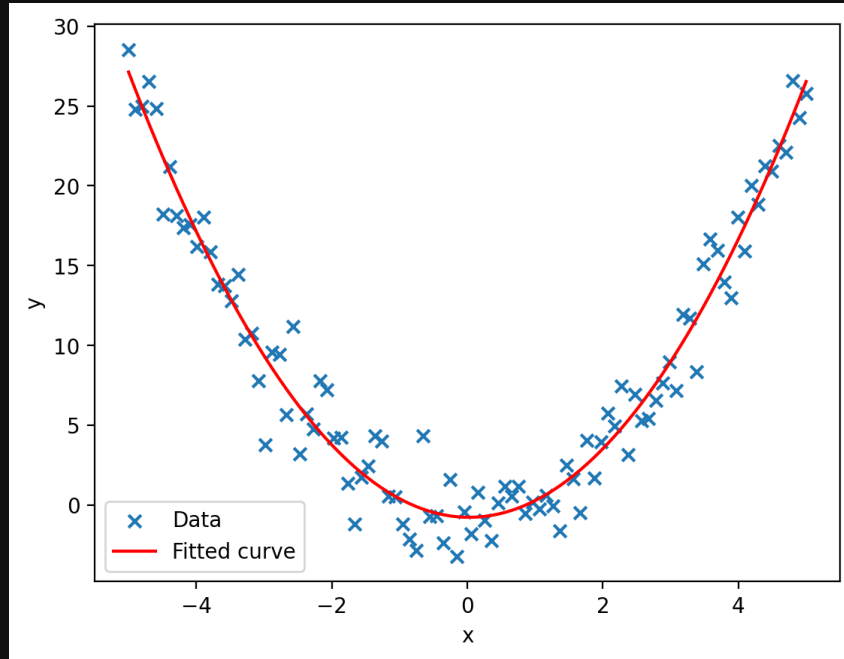
Strukturen in Daten finden



# Machine Learning

<https://www.youtube.com/embed/qe0-aw5-Kcl?enablejsapi=1>

## Lineare Regression



$$\hat{f}(x) = \sum_{k=1}^n c_k \Phi_k(x)$$

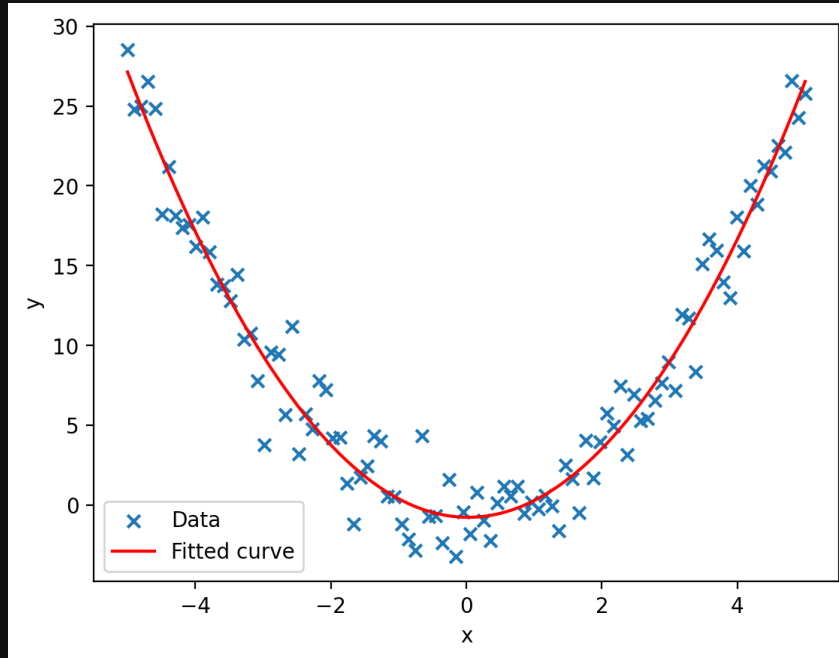
$$\Phi_1 = 1 \quad c_1 \approx 0$$

$$\Phi_2 = x \quad c_2 \approx 0$$

$$\Phi_3 = x^2 \quad c_3 \approx 1$$

# Machine Learning

<https://www.youtube.com/embed/5coJ0FeaWwA?enablejsapi=1>



Lineare Regression

$$\hat{f}(x) = \sum_{k=1}^n c_k \Phi_k(x)$$

- Genereller Funktionsapproximator
- braucht geeignete Basisfunktionen
- braucht kuratierte Daten

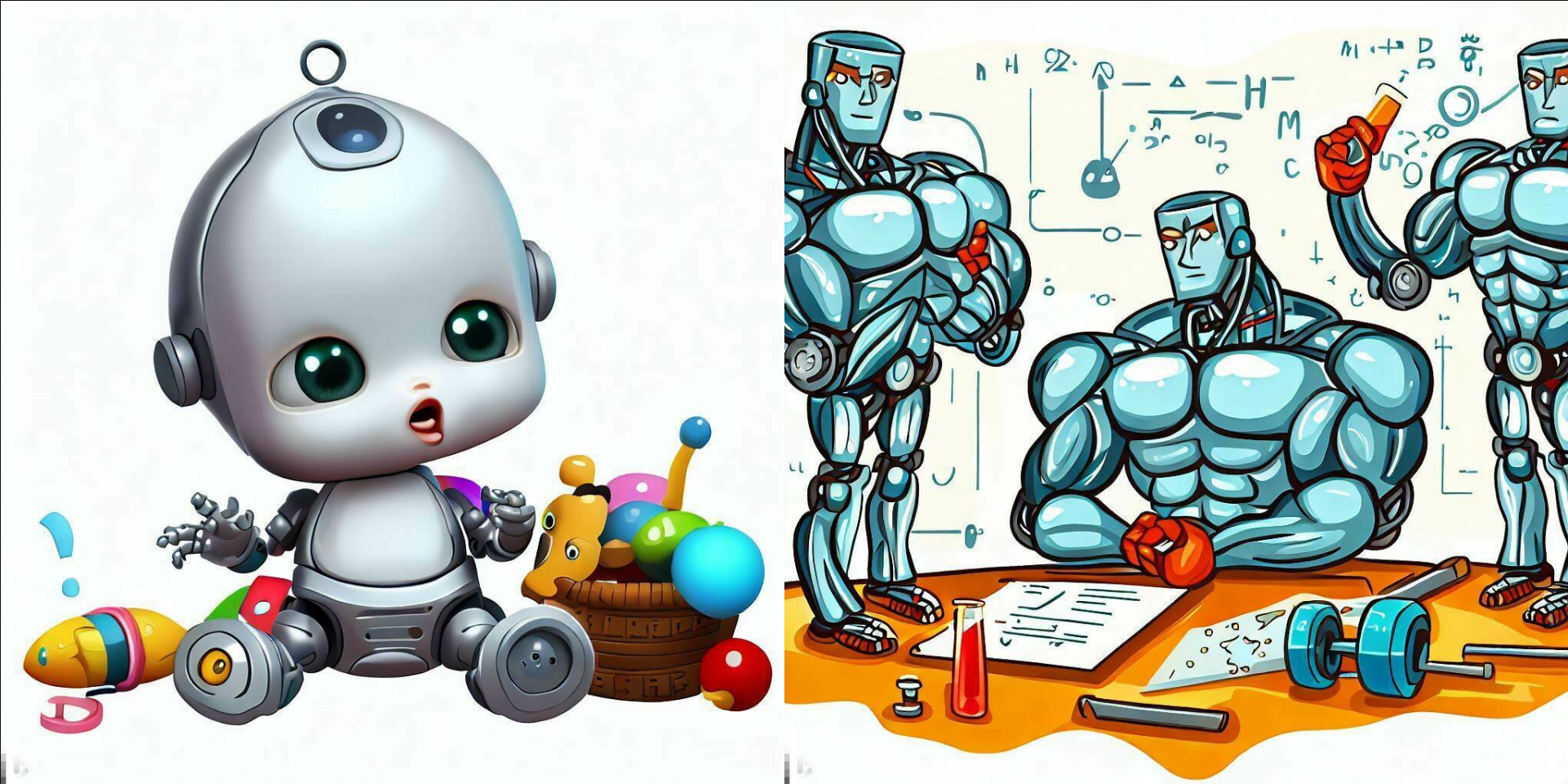
Mehr zu Machine Learning

# Machine Learning

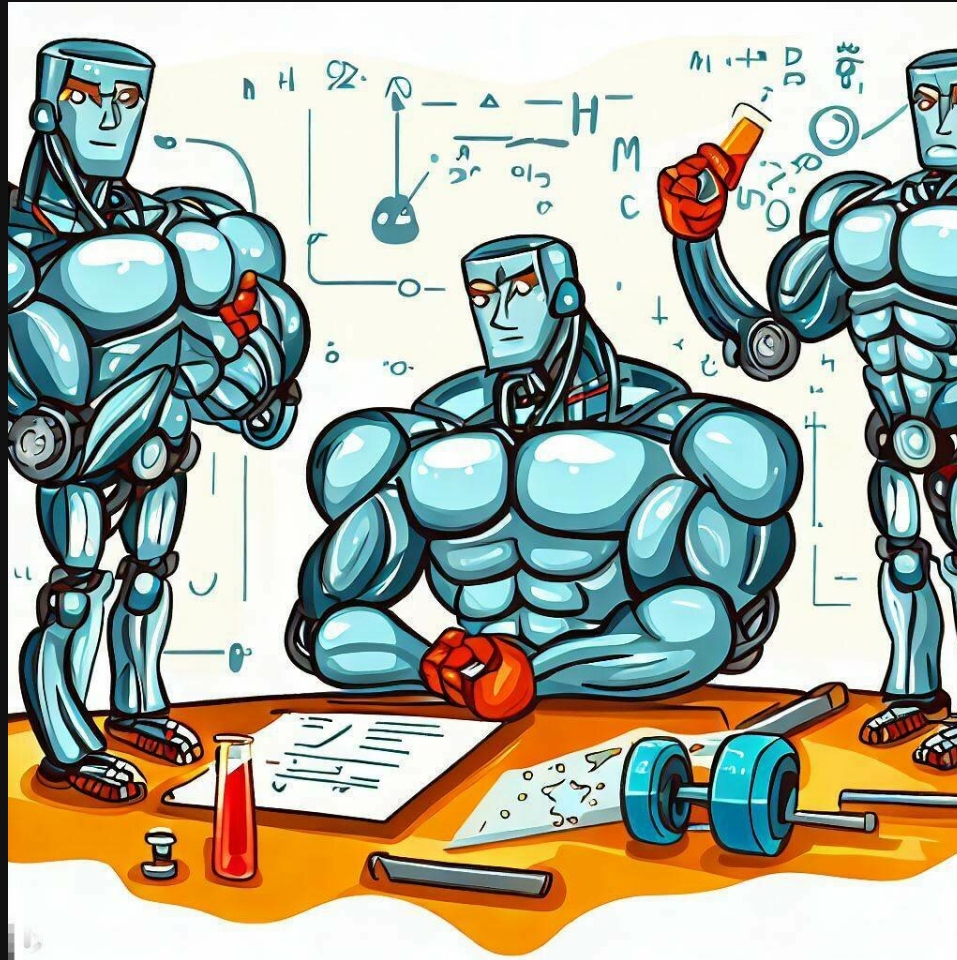




# Machine Learning Deep Learning



# Deep Learning



# Deep Learning

<https://www.youtube.com/embed/aPvbG6zr2OQ?enablejsapi=1>

- Universeller Funktionsapproximator
- Findet Basisfunktionen selbst?

$$\hat{f}(x) = \sum_{k=1}^{n_L} c_k^L \Phi_k^L(x)$$



# Deep Learning

<https://www.youtube.com/embed/6L4201VW4?w=100%&h=100%&enablejsapi=1>

- Findet Basisfunktionen selbst mit Linearer Regression

$$\hat{f}(x) = \sum_{k=1}^{n_L} c_k^L \Phi_k^L(x)$$

$$\Phi_k^L(x) = \sum_{k=1}^{n_{L-1}} c_k^{L-1} \Phi_k^{L-1}(x)$$

⋮

$$\Phi_k^1(x) = \sum_{k=1}^N c_k^1 x_k$$

# Deep Learning

<https://www.youtube.com/embed/lsmfSL3yYU?enablejsapi=1>

- Findet Basisfunktionen selbst mit Linearer Regression

Output Layer  $\hat{f}(x) = \sum_{k=1}^{n_L} c_k^L \Phi_k^L(x)$

$$\Phi_k^L(x) = \sum_{k=1}^{n_{L-1}} c_k^{L-1} \Phi_k^{L-1}(x)$$

Hidden Layers  $\vdots$  Input Layer

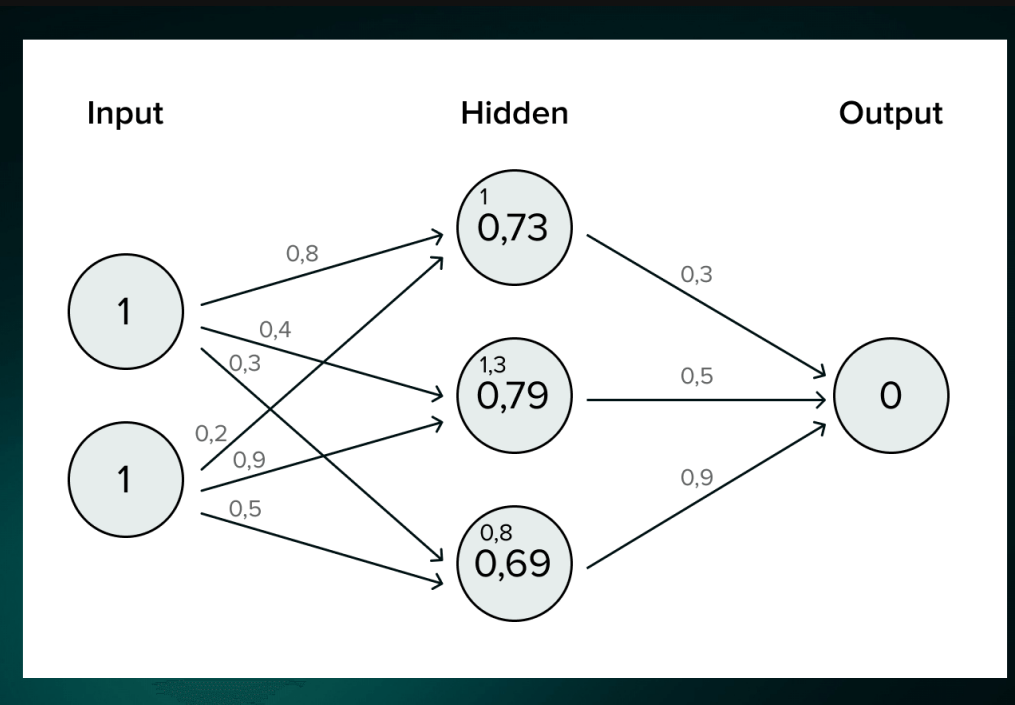
$$\Phi_k^1(x) = \sum_{k=1}^N c_k^1 \downarrow x_k$$

# Deep Learning

<https://www.youtube.com/embed/BI0xaXp014A?enablejsapi=1>

- Findet Basisfunktionen selbst mit Linearer Regression

$$\hat{f}(x) = \sum_{k=1}^n c_k \Phi_k^L(x)$$



$$\Phi_k^1 = x_k$$

$$\Phi_k^l = \sum_k c_k^l \Phi_k^{l-1}$$

$$\Phi^l = w \cdot x + b$$

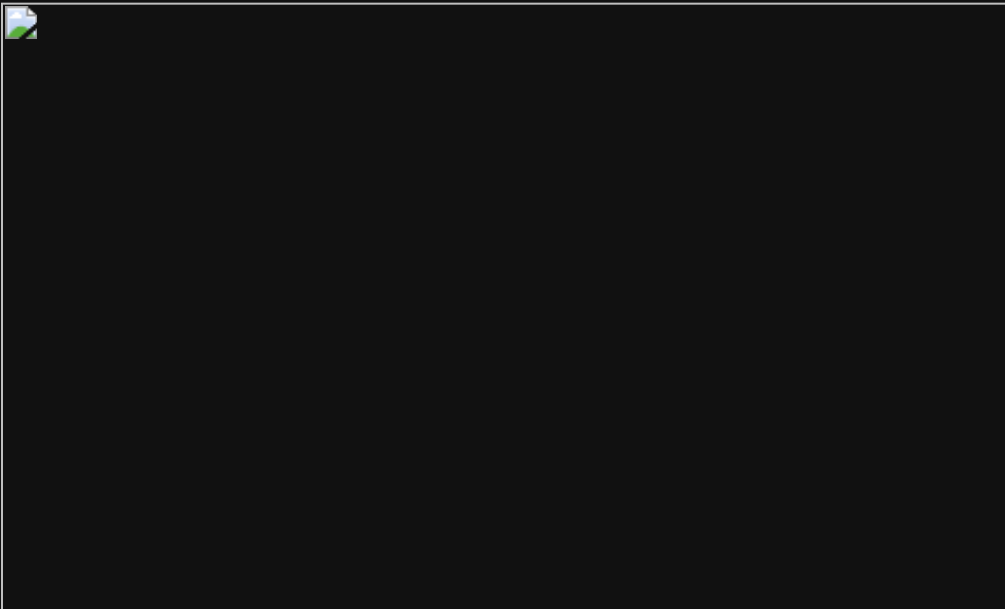
Weights \* Inputs + Bias

# Deep Learning

[https://www.youtube.com/embed/yHNg\\_hMEKJ9E?enablejsapi=1](https://www.youtube.com/embed/yHNg_hMEKJ9E?enablejsapi=1)

- Findet Basisfunktionen selbst mit Linearer Regression

$$\hat{f}(x) = \sum_{k=1}^n c_k \Phi_k^L(x)$$



$$\Phi_k^1 = x_k$$

$$\Phi_k^l = \sum_k c_k^l \Phi_k^{l-1}$$

$$\Phi^l = w \cdot x + b$$

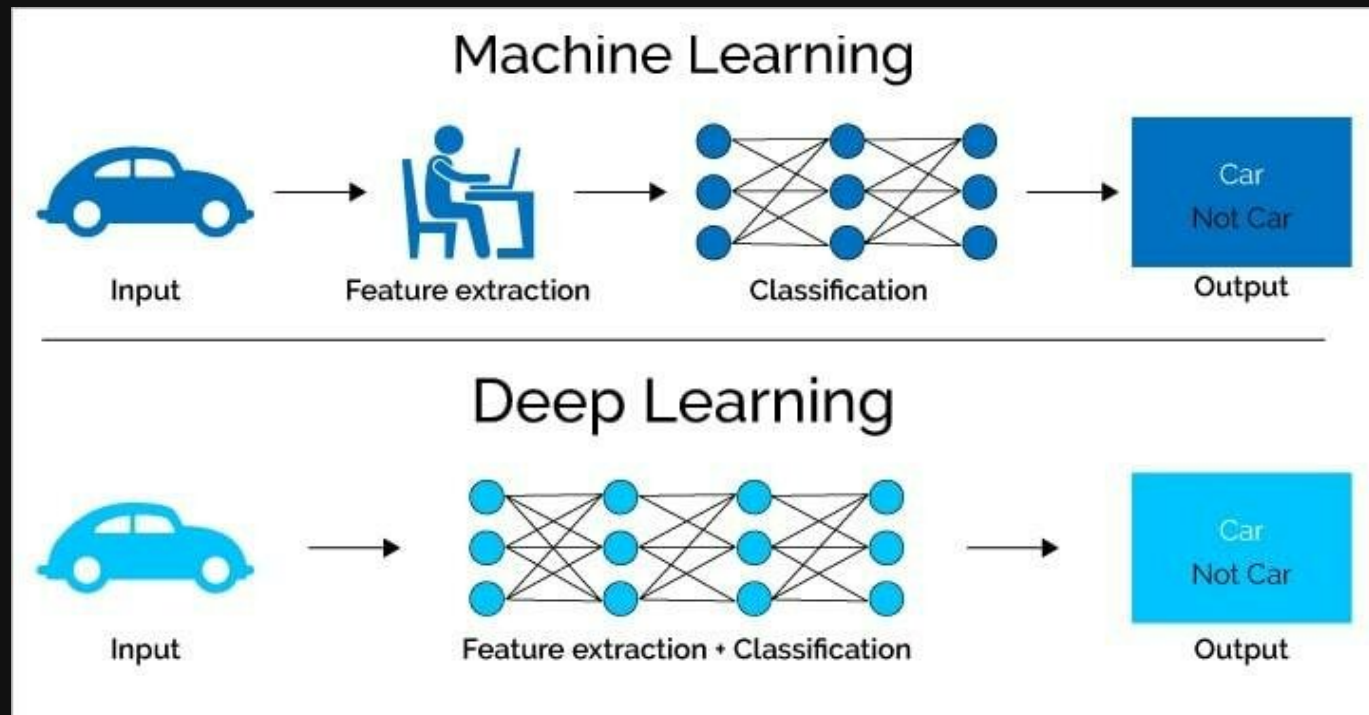
Weights \* Inputs + Bias

mehr zu Neuronalen Netzen

# Deep Learning

<https://www.youtube.com/embed/uFbhqFL55-w?enablejsapi=1>

- Spezialgebiet des Machine Learning
- Universeller Funktionsapproximator: Neuronale Netze
- Komplexe Algorithmen





# Deep Learning

## Anwendungen

<https://www.youtube.com/embed/2pCfjIJZ51g?enablejsapi=1>

- Computer Vision: Klassifizierung, Objekterkennung, Bildgenerierung
- Natural Language Processing: Übersetzungen, Chatbots
- Reinforcement Learning: Gaming, Robotik, Automatisierung
- Zeitreihenanalyse: Markt- & Wetterprognosen, Anomalieerkennung
- Explainable AI: Transparenz, Vertrauen, Sicherheit

# Deep Learning

## Anwendungen

<https://www.youtube.com/embed/09dCziuXlZA?enablejsapi=1>

- Computer Vision: Klassifizierung, Objekterkennung, Bildgenerierung

**Classification**



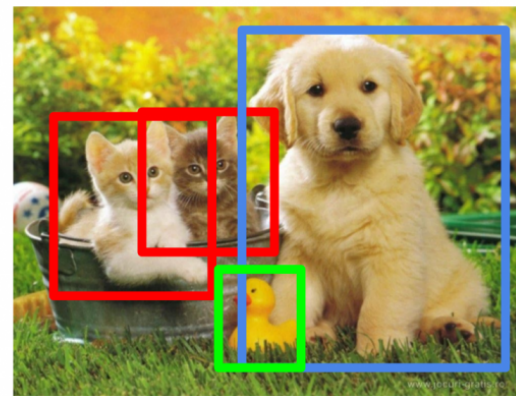
CAT

**Classification  
+ Localization**



CAT

**Object Detection**



CAT, DOG, DUCK

**Instance  
Segmentation**



CAT, DOG, DUCK

Single object

Multiple objects

# Deep Learning

## Anwendungen

<https://www.youtube.com/embed/YSZ4kL8R1SY?enablejsapi=1>

- Computer Vision: Klassifizierung, Objekterkennung, Bildgenerierung



# Deep Learning

## Anwendungen

- Computer Vision: Klassifizierung, Objekterkennung, Bildgenerierung
- Natural Language Processing: Übersetzungen, Chatbots
- Reinforcement Learning: Gaming, Robotik, Automatisierung
- Zeitreihenanalyse: Markt- & Wetterprognosen, Anomalieerkennung
- Explainable AI: Transparenz, Vertrauen, Sicherheit

# Deep Learning

## Anwendungen

<https://www.youtube.com/embed/YkPta1npO2A?enablejsapi=1>

- Natural Language Processing: Übersetzungen, Chatbots





# Deep Learning

## Anwendungen

- Computer Vision: Klassifizierung, Objekterkennung, Bildgenerierung
- Natural Language Processing: Übersetzungen, Chatbots
- Reinforcement Learning: Gaming, Robotik, Automatisierung
- Zeitreihenanalyse: Markt- & Wetterprognosen, Anomalieerkennung
- Explainable AI: Transparenz, Vertrauen, Sicherheit

# Deep Learning

## Anwendungen

<https://www.youtube.com/embed/B7zuC9CiAE4?enablejsapi=1>

- Reinforcement Learning: Gaming, Robotik, Automatisierung



# Deep Learning

## Anwendungen

- Computer Vision: Klassifizierung, Objekterkennung, Bildgenerierung
- Natural Language Processing: Übersetzungen, Chatbots
- Reinforcement Learning: Gaming, Robotik, Automatisierung
- Zeitreihenanalyse: Markt- & Wetterprognosen, Anomalieerkennung
- Explainable AI: Transparenz, Vertrauen, Sicherheit

# Deep Learning

## Anwendungen

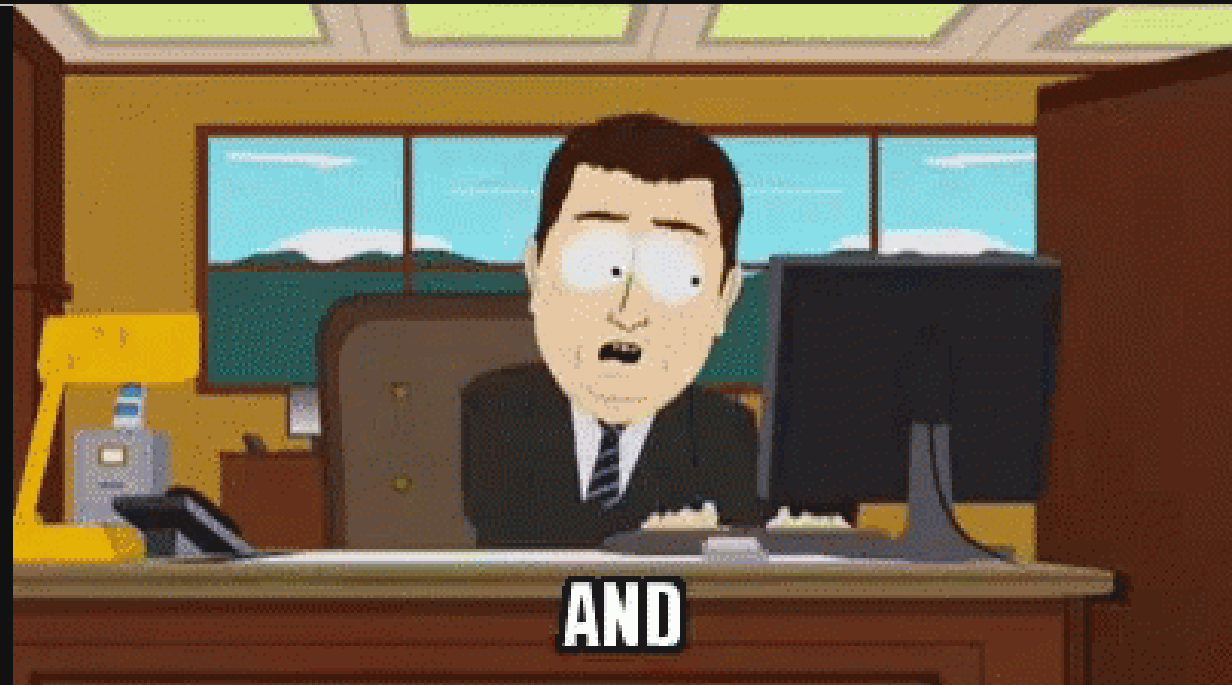
- Zeitreihenanalyse: Markt- & Wetterprognosen, Anomalieerkennung



# Deep Learning

## Anwendungen

- Zeitreihenanalyse: Markt- & Wetterprognosen, Anomalieerkennung





# Deep Learning

## Anwendungen

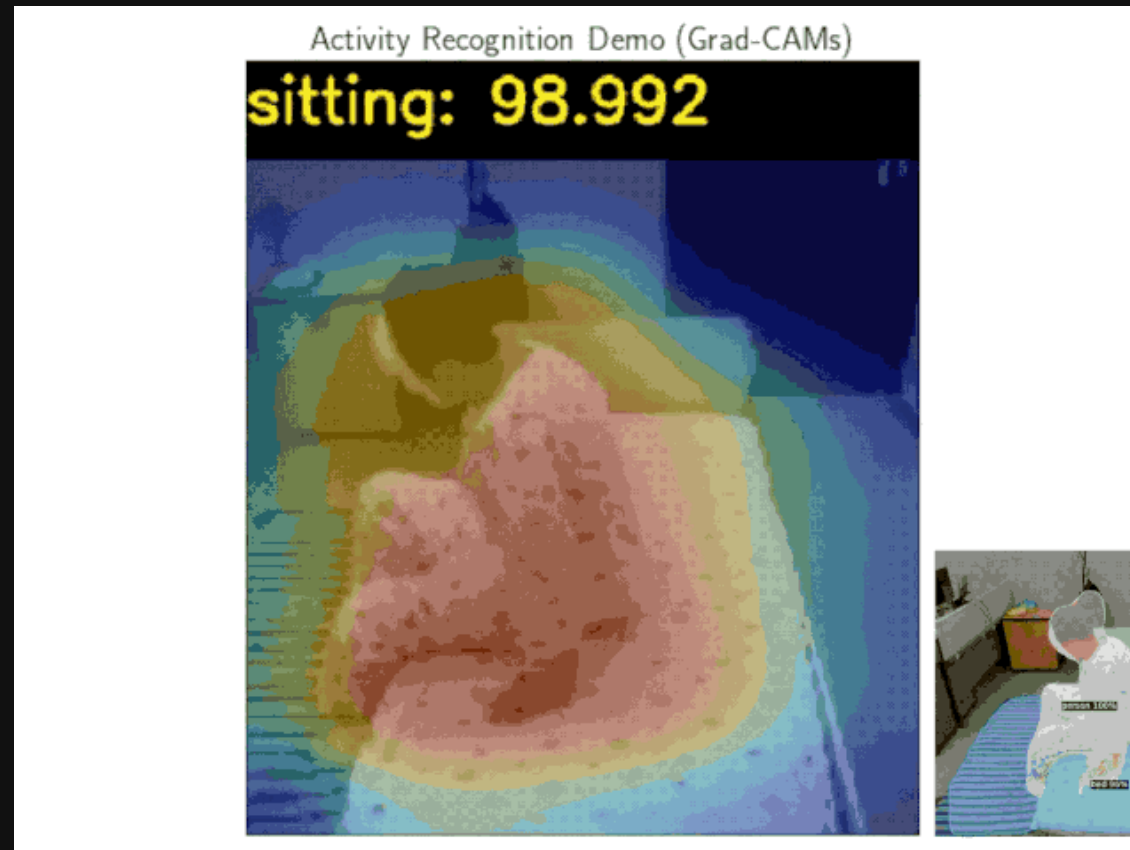
- Computer Vision: Klassifizierung, Objekterkennung, Bildgenerierung
- Natural Language Processing: Übersetzungen, Chatbots
- Reinforcement Learning: Gaming, Robotik, Automatisierung
- Zeitreihenanalyse: Markt- & Wetterprognosen, Anomalieerkennung
- Explainable AI: Transparenz, Vertrauen, Sicherheit

# Deep Learning

## Anwendungen

<https://www.youtube.com/embed/Uyg9oXjh02I?enablejsapi=1>

- Explainable AI: Transparenz, Vertrauen, Sicherheit



# Deep Learning

## Anwendungen

- Computer Vision: Klassifizierung, Objekterkennung, Bildgenerierung
- Natural Language Processing: Übersetzungen, Chatbots
- Reinforcement Learning: Gaming, Robotik, Automatisierung
- Zeitreihenanalyse: Markt- & Wetterprognosen, Anomalieerkennung
- Explainable AI: Transparenz, Vertrauen, Sicherheit

# Deep Learning

## Good Practice

<https://www.youtube.com/embed/D2VRF5HZNIk?enablejsapi=1>

### Case Study: Fragestellung

- Forschungsfragen
- Ziele
- Anwendungsdomäne
- Daten

1. Daten Erkunden

2. Modell & Trainingsloop erstellen

3. Evaluieren

4. Overfitten

5. Regularisieren

6. Optimieren

### Report

- Diskussion
- Konklusion

nach [Karpathy: A receipe for training neural networks](#)

Auch hilfreich: [Lones: How to avoid machine learning pitfalls](#)