

# Maschinelles Lernen – Einführung

Anton Winschel

- › Libraries, Frameworks, Cloud-Services
  - › Apple Core ML, Google TensorFlow, Amazon ML, Azure ML, ...
- › Hardwaremodule
  - › Apple iPhone X: Neural Engine Chip
  - › CPU, GPU, TPU (Tensor Processing Unit)
- › Anwendungen
  - › Recommendersysteme (Web-Suche, Netflix, Amazon, ...)
  - › Intelligente Bots
  - › Intelligente Features (Schrittzähler, Fortschrittsanzeige, ...)

# WAS IST MASCHINELLES LERNEN?

- › Generierung von Wissen aus Erfahrung
  - › **Erfahrung:** Lernen aus Beispieldaten
  - › **Wissen:** Erkennen von Mustern und Gesetzmäßigkeiten in Daten
- › Ursprung: Stochastische Optimierung

$$f(x, o) + e = y$$

$f$	Abbildung
$x \in X$	Eingabedaten
$y \in Y$	Ausgabedaten
$o \in \Omega$	Parameter der Abbildung
$e$	Fehler der Abbildung

- Ziel:**
1. Finde  $o$ , sodass  $e$  minimal ist (Optimierungsproblem)
  2.  $f$  soll auf realistische unbekannte Daten generalisieren

$$f(x, o) + e = y$$

## Feature

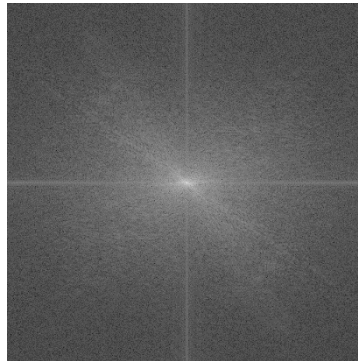
- › Eigenschaft von  $x$ , die sich mathematisch ausdrücken lässt
- › Beispiel: Objekterkennung



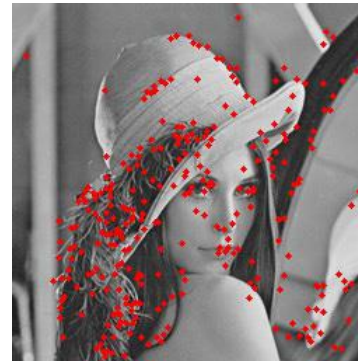
Original



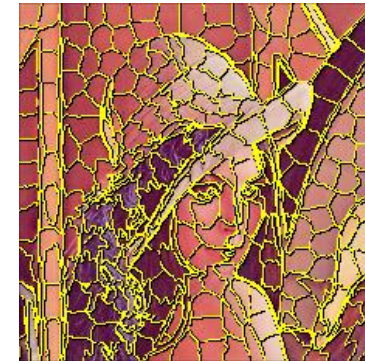
Kanten



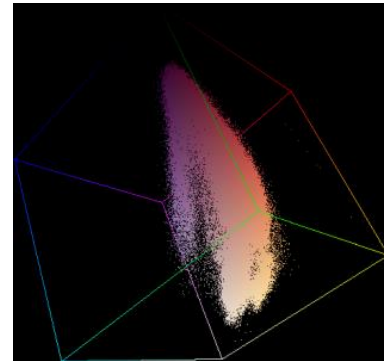
Frequenzen



Keypoints



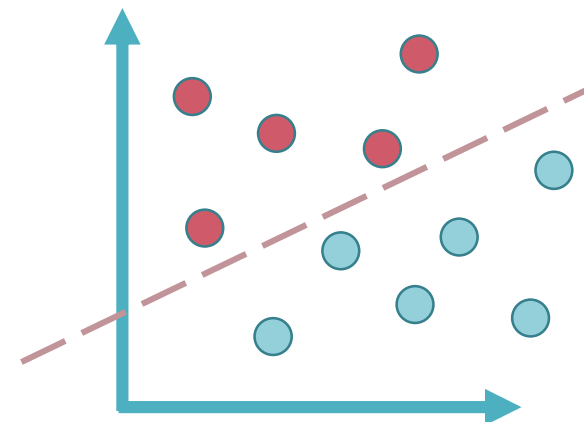
Segmentierung



Farbverteilung

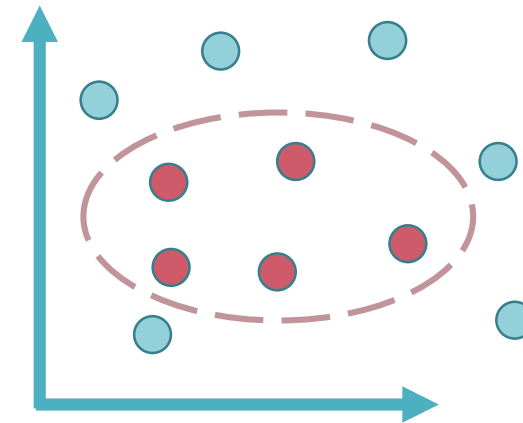
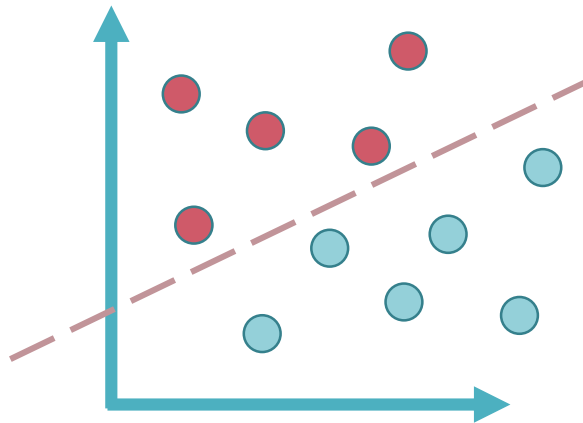
## Featureraum und Entscheidungsgrenzen

- › Hochdimensionaler Raum
- › Maschinelles Lernen: aus Stichproben die Struktur eines hochdimensionalen Raums erkennen

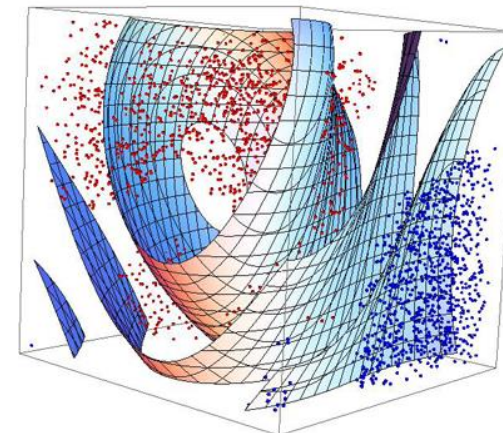
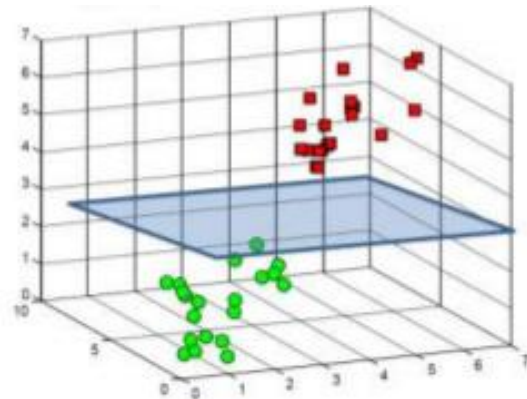


$$f(x, o) + e = y$$

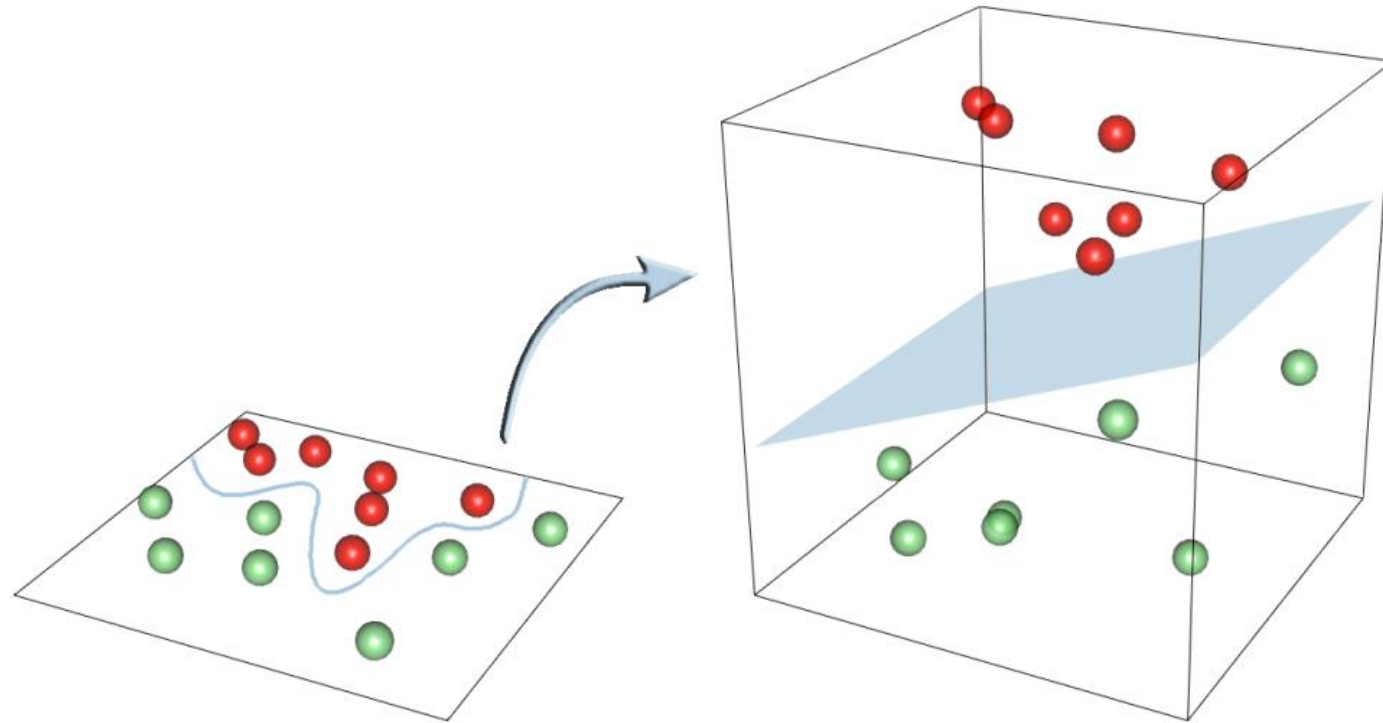
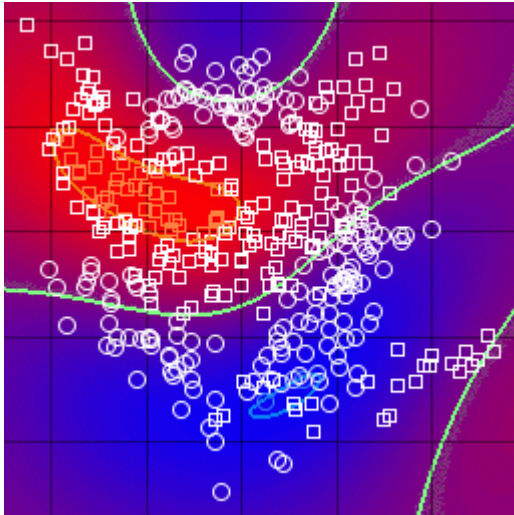
› Zusammenhang zwischen  $x$  und  $y$  (linear, nichtlinear)



› Dimension von  $x$  und  $y$

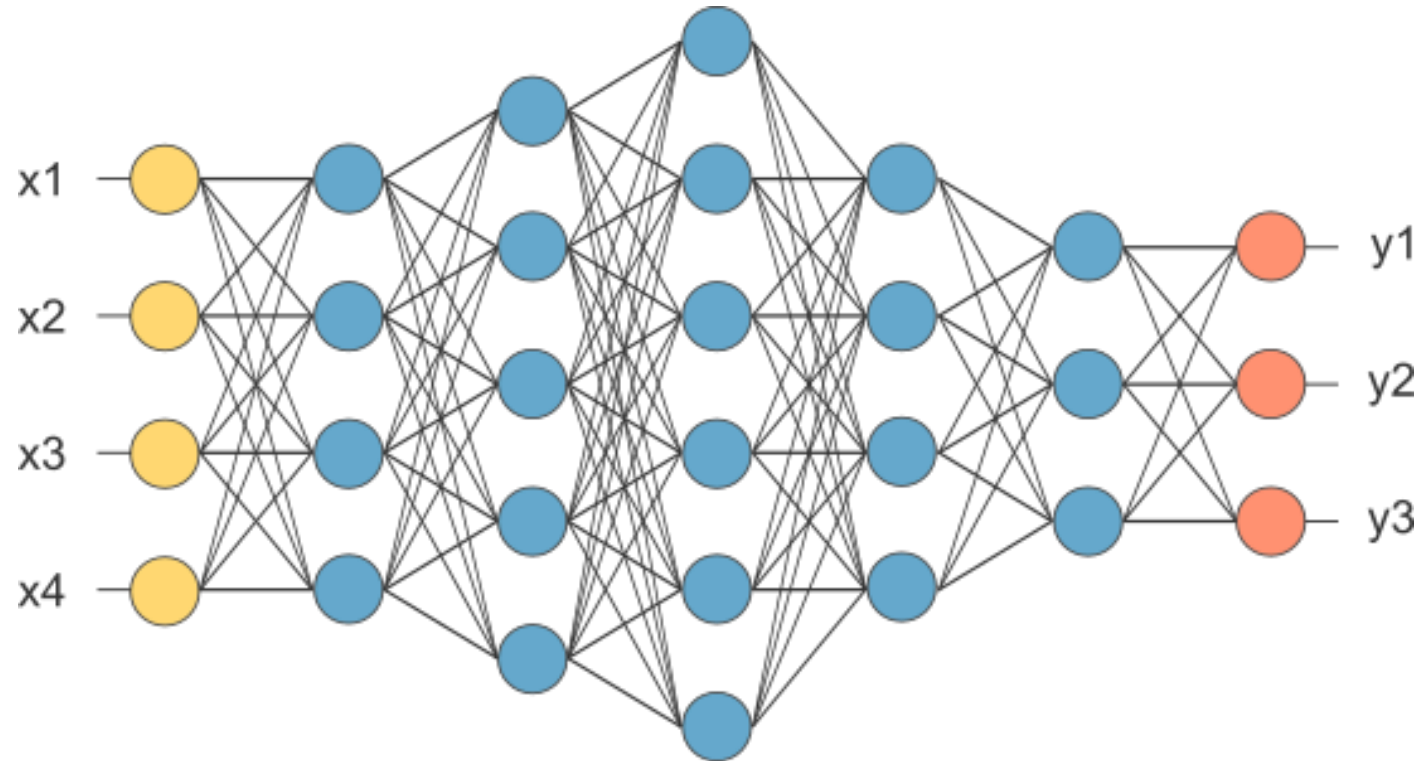


$$f(x, o) + e = y$$



- Klassifikation
- Regression

$$f(x, o) + e = y$$

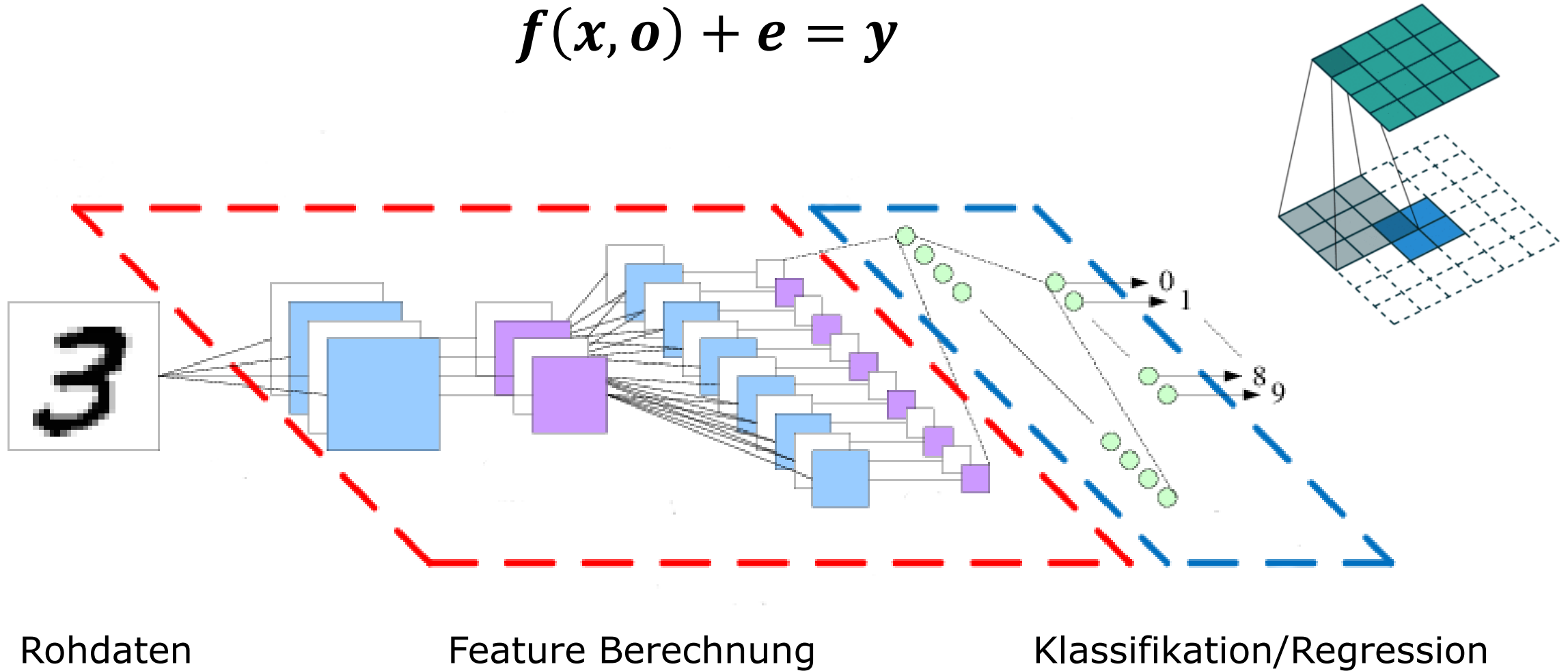


**Universales Approximationstheorem:**

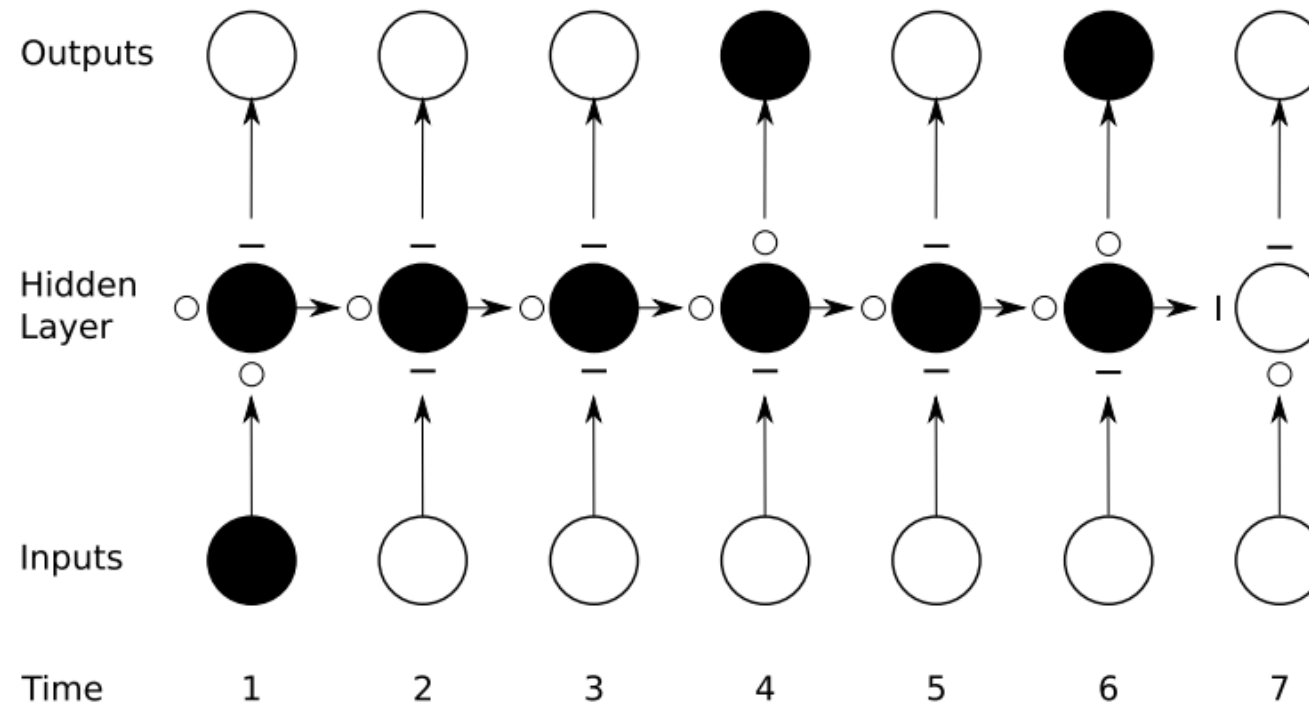
Ein neuronales Netz mit nur einer Zwischenschicht kann jede Funktion beliebig genau approximieren



$$f(x, o) + e = y$$

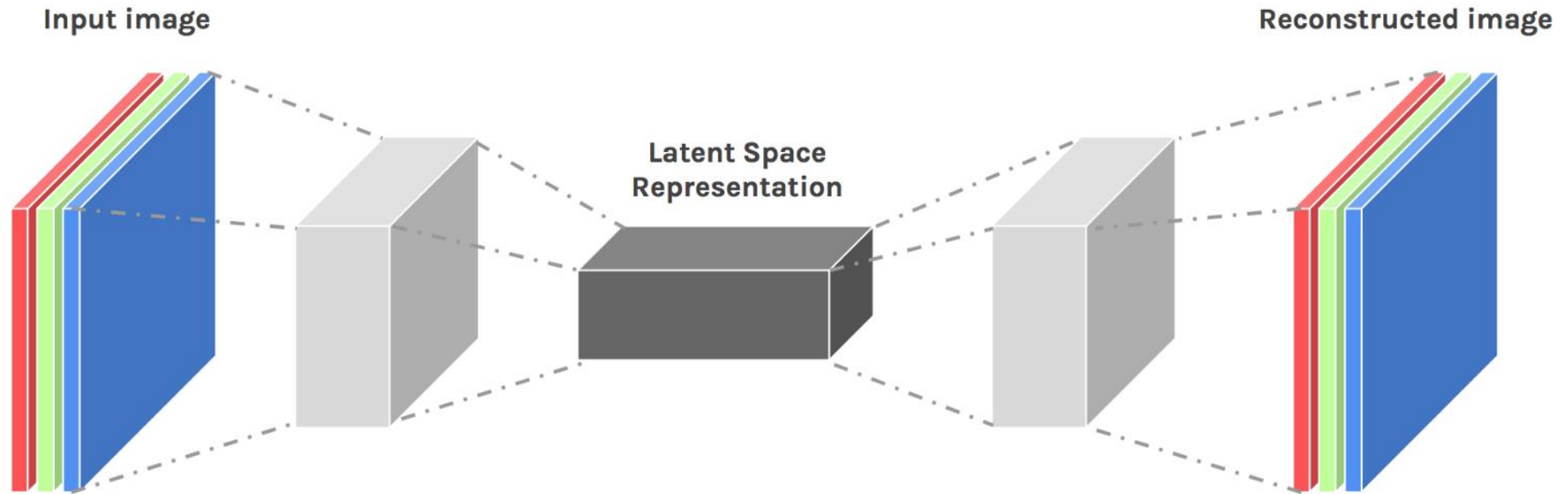


$$f(x, o) + e = y$$



- Zeitliche Vorhersage
  - Autovervollständigung von Texteingaben

$$f(x, o) + e = y$$



- Filter
  - Kompression
  - Graustufenbilder → Farbbilder
  - niedrige Auflösung → höhere Auflösung

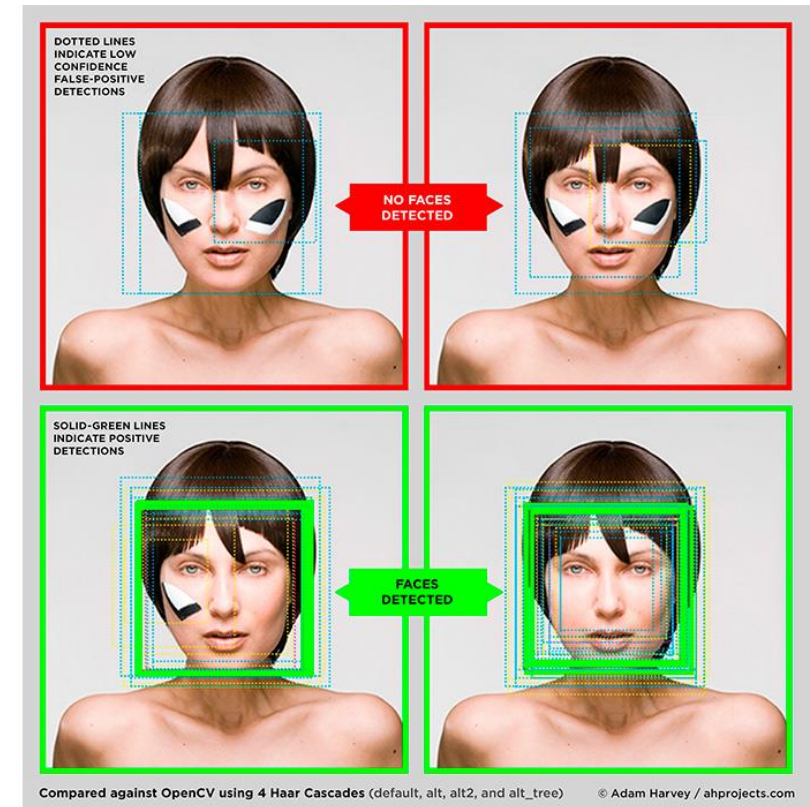
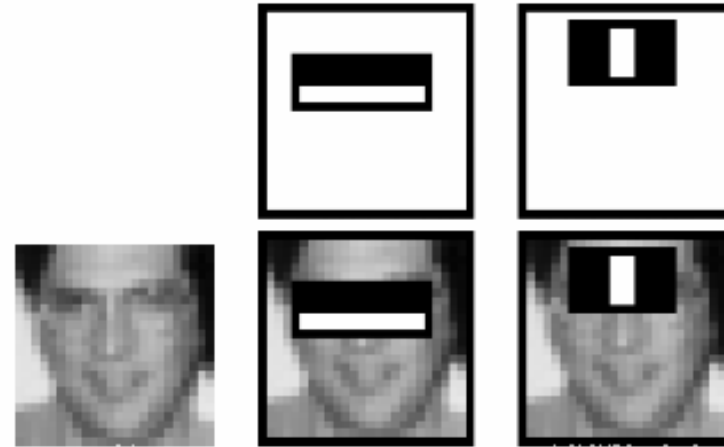
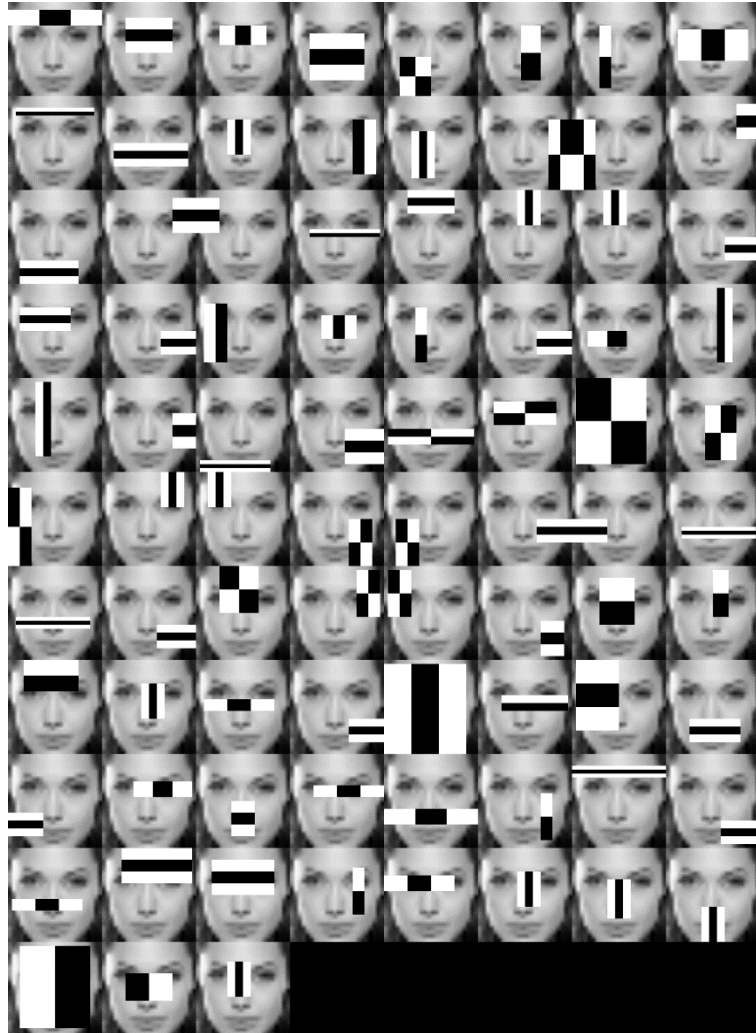
## > Algorithmische Schwächen

- > Abhängigkeit vom Trainingsdatensatz (realistisch, Ausreißer)
- > Trainingsdauer
- > Lokale Minima
- > Viele Parameter, die Erfahrung benötigen (oder langes Fine-Tuning)

## > Geometrische Schwächen

# ALGORITHMISCHE SCHWÄCHEN

## Viola Jones Detektor



# GEOMETRISCHE SCHWÄCHEN

Orthogonale Projektion auf Entscheidungsgrenzen

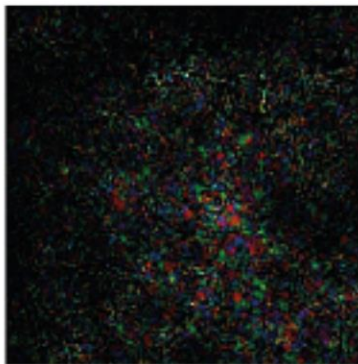
$x$



$x + r$

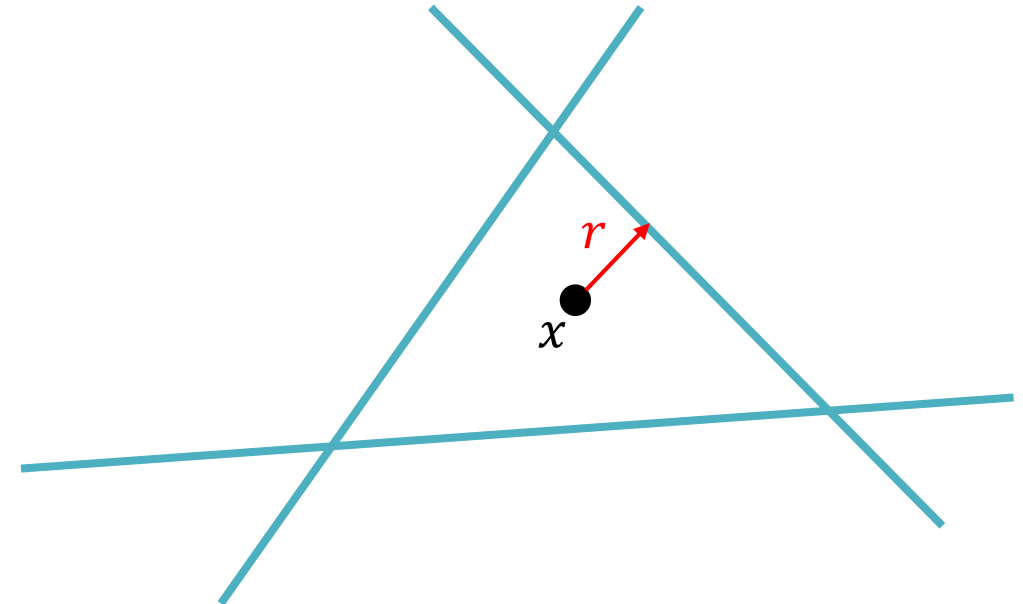


$r$



$f(x) = \text{"Wal"}$

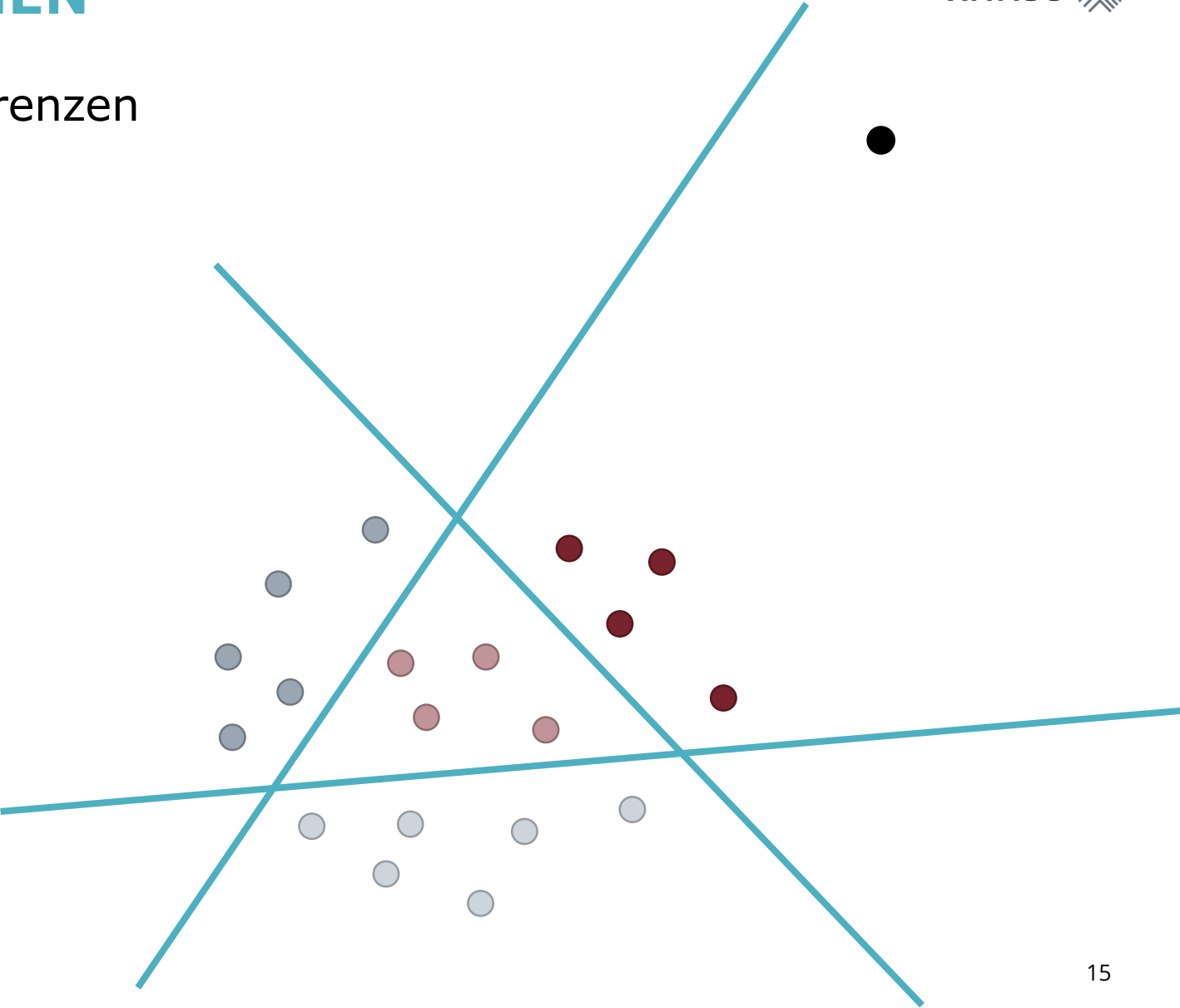
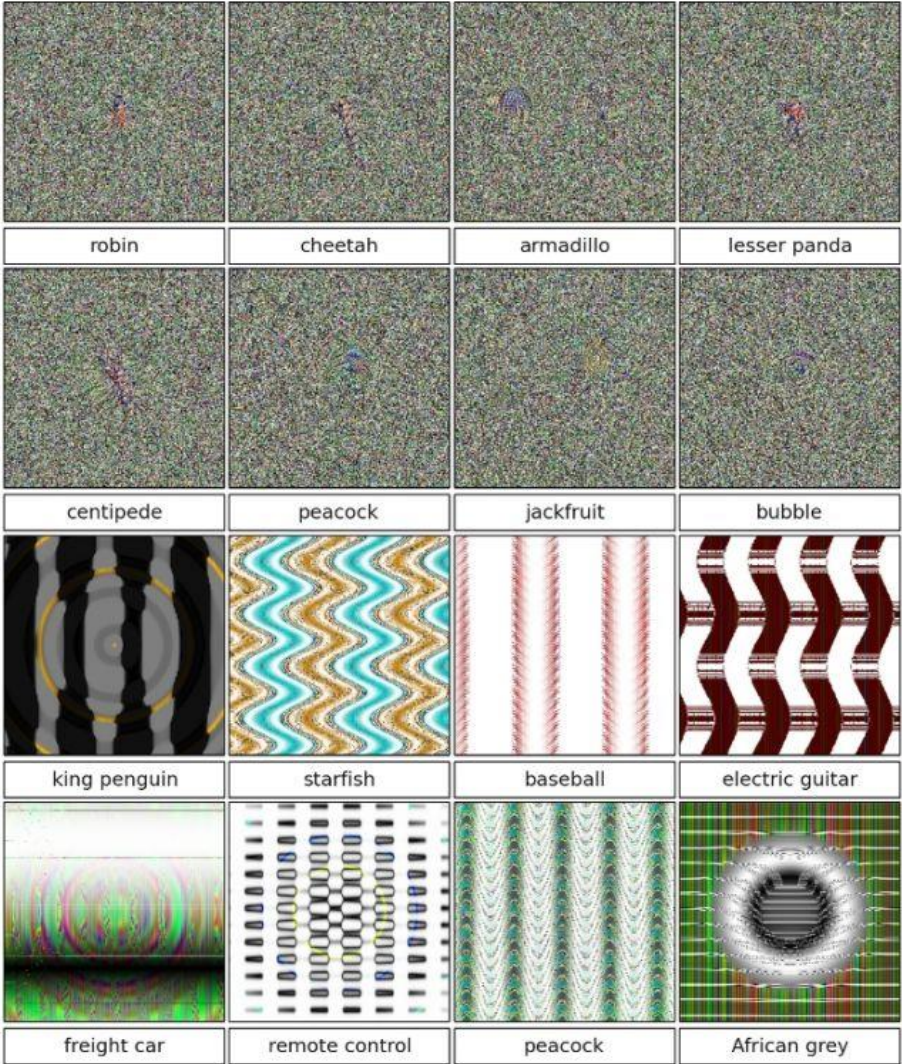
$f(x + r) = \text{"Schildkröte"}$



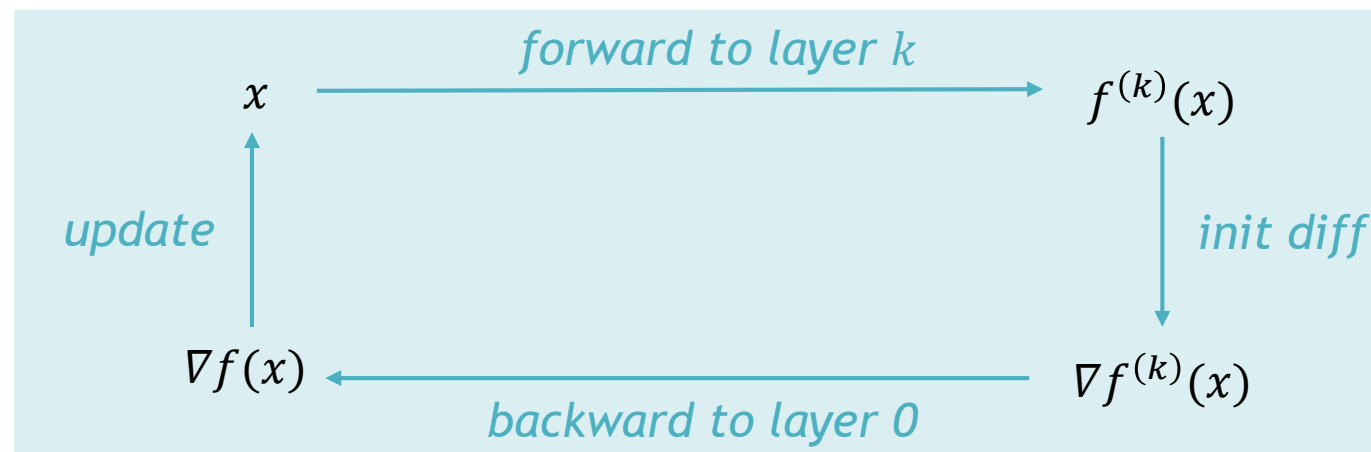
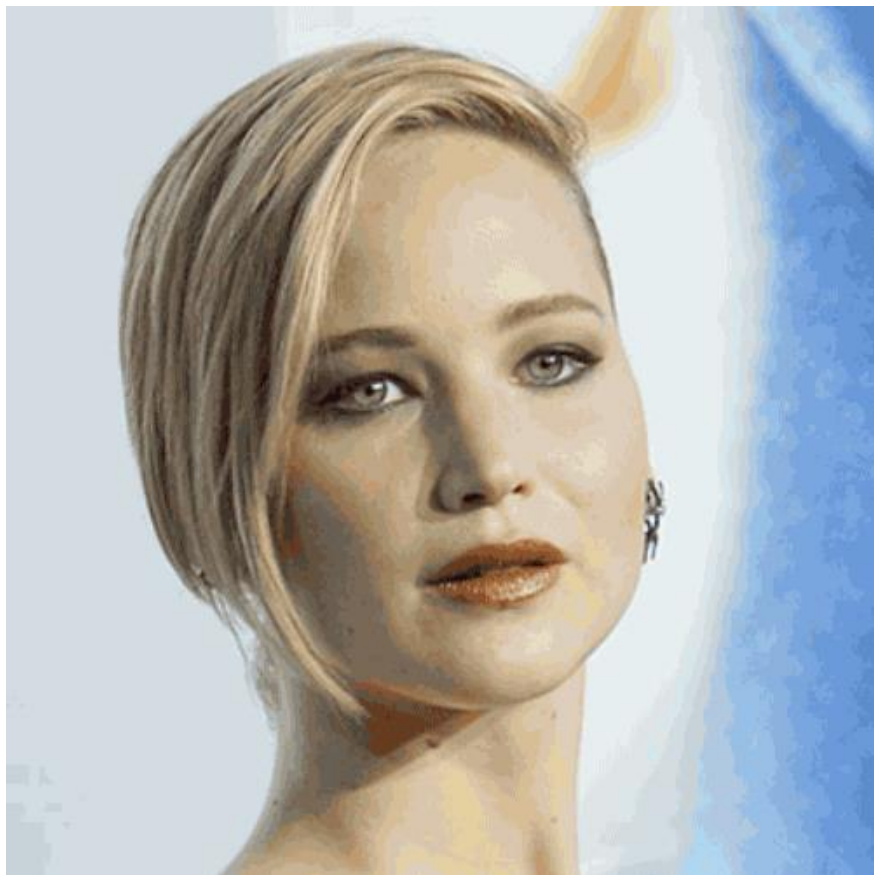


# GEOMETRISCHE SCHWÄCHEN

Daten jenseits von Entscheidungsgrenzen



# GOOGLE DEEP DREAM





- › Trainingsdaten sammeln
  - › Synthetische Daten
  - › Crowdsourcing
  - › Pre-Training
- › Features nicht selbst bestimmen, sondern lernen
  - › SVM → Faltungsnetze
- › CPU → GPU → TPU + Entwicklertools
  - › Training auf Rechnercluster/Cloud
  - › Test in Echtzeit → Mobile
  - › Berechnung im Frequenzraum

# DEEP LEARNING → DEEPER LEARNING

- › AlexNet (2012) 8 Schichten
- › VGGNet (2013) 19 Schichten
- › GoogLeNet (2014) 22 Schichten
- › Microsoft ResNet (2015) 152 Schichten
- › Stochastic ResNet (2016) 1200 Schichten

