

# MASCHINELLES LERNEN FÜR INTELLIGENTE SYSTEME: ÜBER DATEN, WISSEN UND ALGORITHMEN

Eyke Hüllermeier

Intelligente Systeme und Maschinelles Lernen

---

# MEILENSTEINE DER KI



AlphaGo schlägt Lee Sedol (2016)



Watson gewinnt Jeopardy! (2011)



Deep Blue schlägt Garry Kasparov (1997)

# MEILENSTEINE DER KI

**Daten**  
**+ Lernen**



AlphaGo schlägt Lee Sedol (2016)

**Wissen**  
**+ Retrieval**



Watson gewinnt Jeopardy! (2011)



Deep Blue schlägt Garry Kasparov (1997)

**Algorithmik**  
**+ Rechenleistung**

# MEILENSTEINE DER KI

**Daten**  
**+ Lernen**



*Von der Welt der Spiele  
in die Realität ...*

AlphaGo schlägt Lee Sedol (2016)

**Wissen**  
**+ Retrieval**



Watson gewinnt Jeopardy! (2011)



Deep Blue schlägt Garry Kasparov (1997)

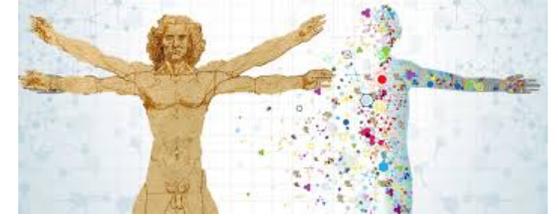
**Algorithmik**  
**+ Rechenleistung**

# ANWENDUNGEN DER KI



INDUSTRIE 4.0

PRECISION  
MEDICINE



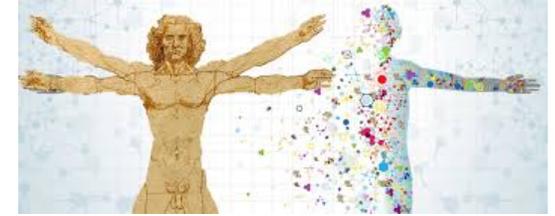
AUTONOMES  
FAHREN

# ANWENDUNGEN DER KI

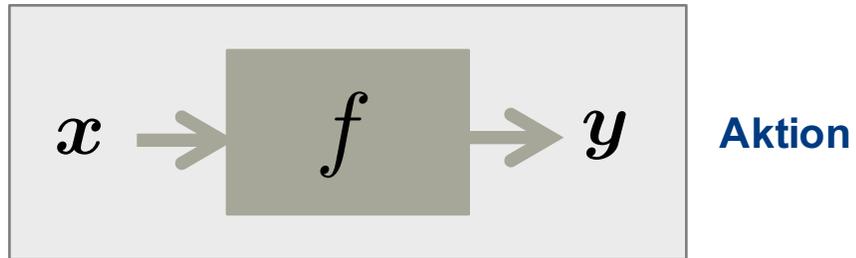


INDUSTRIE 4.0

PRECISION  
MEDICINE

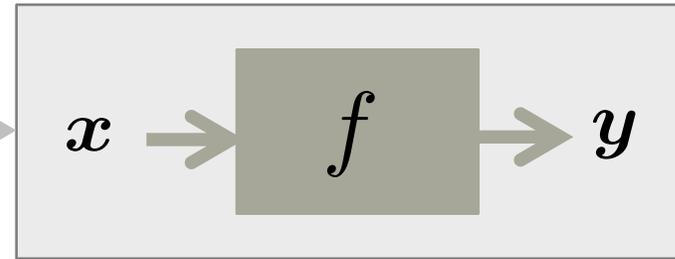


**Zustandsvektor**  
beschreibt die  
Umgebung



AUTONOMES  
FAHREN

# DER ALGORITHMISCHE ANSATZ



Algorithmus

```
ALGORITHM shortest-path( $V, T$ )  
   $W := \{v_1\}$   
  ShortDist[ $v_1$ ] := 0  
  FOR each  $u$  in  $V - \{v_1\}$   
    ShortDist[ $u$ ] :=  $T[v_1, u]$   
  WHILE  $W \neq V$   
    MinDist := INFINITE  
    FOR each  $v$  in  $V - W$   
      IF ShortDist[ $v$ ] < MinDist  
        MinDist = ShortDist[ $v$ ]  
         $w := v$   
      END {if}  
    END {for}  
     $W := W \cup \{w\}$   
    FOR each  $u$  in  $V - W$   
      ShortDist[ $u$ ] := Min(ShortDis[ $u$ ], ShortDist[ $w$ ] +  $T[w, u]$ )  
  END {while}
```

# DIE EVOLUTION INTELLIGENTER SYSTEME

```
function GetMin(var a: TList)
var
  i, min, mini: integer;
begin
  min := MaxInt;
  mini := 0;
  for i := 1 to a.len do
    if a.arr[i].G < min t
    begin
      min := a.arr[i].G
      mini := i;
    end;
  end;
  GetMin := mini;
end;
```

klassische  
Programmierung

```
mann(adam).
mann(tobias).
mann(frak).
frau(eva).
frau(daniela).
frau(ulrike).
vater(adam,tobias).
vater(tobias,frak).
vater(tobias,ulrike).
mutter(eva,tobias).
mutter(daniela,frak).
mutter(daniela,ulrike).
```

Wissensbasierte  
Systeme

```
# Spot Check Algorithms
models = []
models.append('LR', LogisticRegression)
models.append('LDA', LinearDiscriminantAnalysis)
models.append('KNN', KNeighborsClassifier)
models.append('CART', DecisionTreeClassifier)
models.append('NB', GaussianNaiveBayes)
models.append('SVM', SVC)
# evaluate each model in t
results = []
names = []
for name, model in models:
  kfold = model_selection.cross_val_score(model, X, y, cv=5)
  cv_results = model_selection.cross_val_results.append(cv_resu
  names.append(name)
  msg = "%s: %f (%f)" %
  print(msg)
```

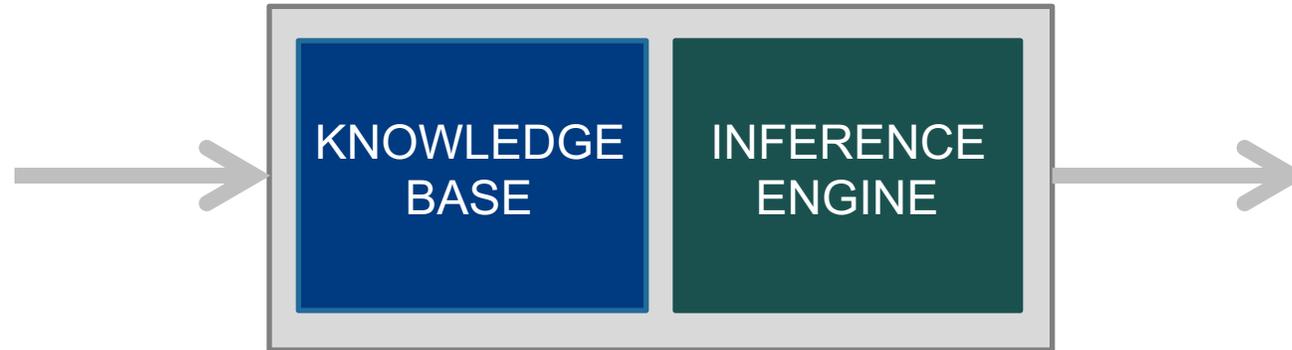
datengetriebenes  
Problemlösen



automatisiertes  
maschinelles Lernen

*... ist schwierig  
für komplexe  
Probleme*

# WISSENSBASIERTE SYSTEME



## EXPERTE

*Repräsentation von problem- und domänenspezifischem Wissen wie Fakten und Regeln.  
„Was“ aber nicht „wie“!*

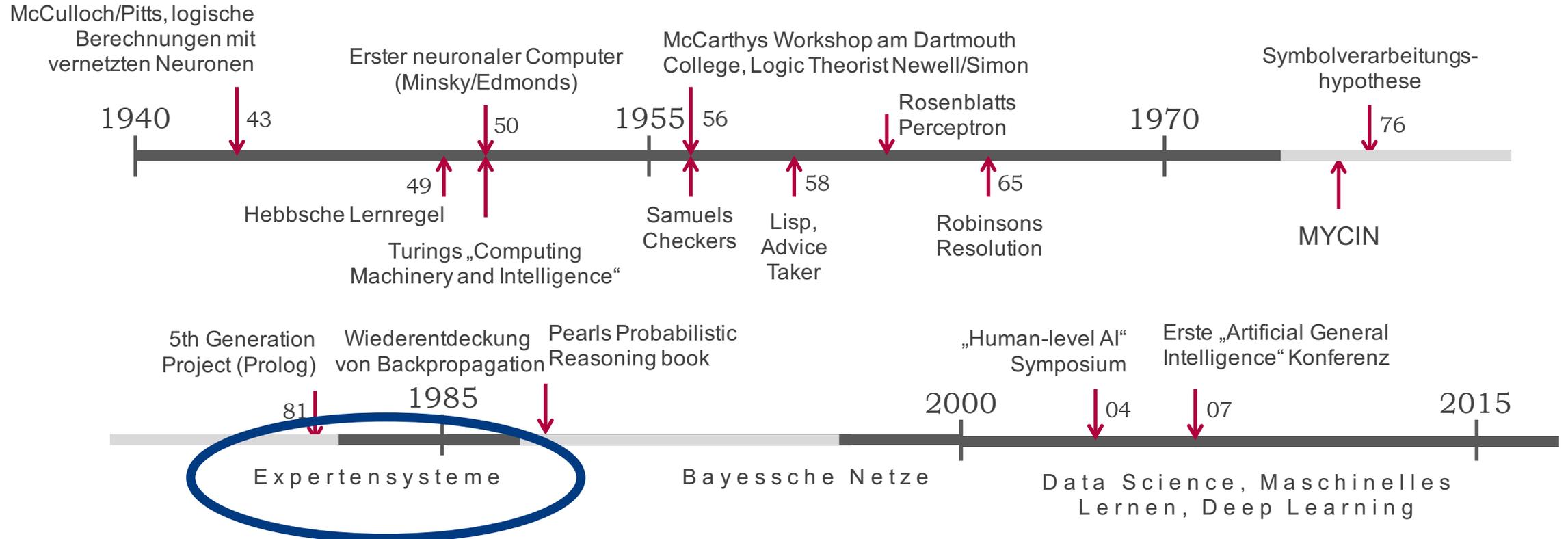
≠

## INFORMATIKER

*Generische Kontrollstrukturen,  
Programme = Theorien in formaler Logik,  
eng verwandt zu deklarativen  
Programmiersprachen*



# GESCHICHTE DER KI



# DIE EVOLUTION INTELLIGENTER SYSTEME

```
function GetMin(var a: TList)
var
  i, min, mini: integer;
begin
  min := MaxInt;
  mini := 0;
  for i := 1 to a.len do
    if a.arr[i].G < min t
    begin
      min := a.arr[i].G
      mini := i;
    end;
  end;
  GetMin := mini;
end;
```

klassische  
Programmierung

*... ist schwierig  
für komplexe  
Probleme*

```
mann(adam).
mann(tobias).
mann(frak).
frau(eva).
frau(daniela).
frau(ulrike).
vater(adam,tobias).
vater(tobias,frak).
vater(tobias,ulrike).
mutter(eva,tobias).
mutter(daniela,frak).
mutter(daniela,ulrike).
```

Wissensbasierte  
Systeme

*... leidet unter dem  
Flaschenhals des  
Wissenserwerbs*

```
# Spot Check Algorithms
models = []
models.append('LR', Logis)
models.append('LDA', Line)
models.append('KNN', KNei)
models.append('CART', Dec)
models.append('NB', Gauss)
models.append('SVM', SVCC)
# evaluate each model in t
results = []
names = []
for name, model in models:
  kfold = model_selectio
  cv_results = model_sel
  results.append(cv_resu
  names.append(name)
  msg = "%s: %f (%f)" %
  print(msg)
```

datengetriebenes  
Problemlösen



automatisiertes  
maschinelles Lernen

# AUTOMATISIERUNG MENSCHLICHER FÄHIGKEITEN

*Menschliche Fähigkeiten sind nicht immer einfach zu beschreiben!*

# AUTOMATISIERUNG MENSCHLICHER FÄHIGKEITEN

*Menschliche Fähigkeiten sind nicht immer einfach zu beschreiben!*



$x \in \mathbb{R}^N$



MANN  
oder  
FRAU

$y \in \{0, 1\}$

# AUTOMATISIERUNG MENSCHLICHER FÄHIGKEITEN

*Menschliche Fähigkeiten sind nicht immer einfach zu beschreiben!*

For example, a reduction of the search space does not immediately imply better solutions.



Eine Beschränkung des Suchraums führt beispielsweise nicht unmittelbar zu besseren Lösungen.

# AUTOMATISIERUNG MENSCHLICHER FÄHIGKEITEN

## Optimal Sample Complexity of $M$ -wise Data for Top- $K$ Ranking

**Algorithm 1** Rank Centrality (Negahban et al., 2012)

**Input** the collection of statistics  $s = \{s_{\mathcal{I}} : \mathcal{I} \in \mathcal{E}^{(M)}\}$ .  
**Convert** the  $M$ -wise sample for each hyper-edge  $\mathcal{I}$  into  $M$  pairwise samples:

1. Choose a circular permutation of the items in  $\mathcal{I}$  uniformly at random.
2. Break it into the  $M$  pairs of adjacent items, and denote the set of pairs by  $\phi(\mathcal{I})$ .
3. Use the (pairwise) data of the pairs in  $\phi(\mathcal{I})$ .

**Compute** the transition matrix  $P = [P_{ij}]_{1 \leq i, j \leq n}$ :

$$\hat{P}_{ij} = \begin{cases} \frac{w_i - w_j}{2} & \text{if } i \neq j; \\ 1 - \sum_{k \neq i, j} \hat{P}_{kj} & \text{if } i = j; \\ 0 & \text{otherwise,} \end{cases}$$

where  $d_{\max}$  is the maximum out-degree of vertices in  $\mathcal{E}$ .  
**Output** the stationary distribution of matrix  $P$ .

$$w_{ij} := \sum_{\mathcal{I} \in \mathcal{E}^{(2)}} \frac{1}{L} \sum_{t=1}^L y_{ij,t}^{(\mathcal{I})}. \quad (16)$$

In an ideal scenario where we obtain an infinite number of samples per  $M$ -wise comparison, i.e.,  $L \rightarrow \infty$ , sufficient statistics  $\frac{1}{L} \sum_{t=1}^L y_{ij,t}^{(\mathcal{I})}$  converge to  $\frac{w_i - w_j}{2}$ , as the PL model is a natural generalized version of the BTL model. Then, the constructed matrix  $P$  defined in Algorithm 1 becomes a matrix  $P$  whose entries  $[P_{ij}]_{1 \leq i, j \leq n}$  are defined as

$$P_{ij} = \begin{cases} \frac{1}{2d_{\max}} \sum_{\mathcal{I} \in \mathcal{E}^{(2)}: i, j \in \mathcal{I}} \frac{w_i - w_j}{w_i + w_j} & \text{for } \mathcal{I} \in \mathcal{E}^{(M)}; \\ 1 - \sum_{k \neq i, j} P_{kj} & \text{if } i = j; \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

The entries for observed item pairs represent the relative likelihood of item  $i$  being preferred over item  $j$ . Intuitively, random walks of  $P$  in the long run visit some states more often, if they have been preferred over other frequently-visited states and/or preferred over many other states.

The random walks are reversible as  $w_i P_{ij} = w_j P_{ji}$  holds, and irreducible under the connectivity assumption. Since we obtain the unique stationary distribution, it is equal to  $w = \{w_1, \dots, w_n\}$  up to some constant scaling.

It is clear that random walks of  $P$ , a noisy version of  $P$ , will give us an approximation of  $w$ . The algorithm et al., 2013) directly follows the ordering evaluated in each sample, if it is  $1 < 2 < \dots < M - 1 < M$ , it is broken into pairs of adjacent items:  $1 < 2$  up to  $M - 1 < M$ . Our method turns out to be consistent, i.e.,  $\frac{P_{ij} w_i - w_j}{w_i + w_j} = \frac{w_i - w_j}{2}$  (see (17)), whereas the adjacent breaking method is not (Azari Soutani et al., 2013).

adopts a power method, known to be computationally efficient in obtaining the leading eigenvalue of a sparse matrix (Meirovitch, 1997), to obtain the stationary distribution.

### 3.2. Proof outline

To outline the proof of Theorem 2, let us introduce Theorem 3. We show that Theorem 3 leads to Theorem 2.

**Theorem 3.** When Rank Centrality is employed, with high probability, the  $\ell_\infty$  norm estimation error is upper-bounded by

$$\frac{\|\hat{w} - w\|_\infty}{\|w\|_\infty} \leq \sqrt{\frac{n \log n}{\binom{n}{k}} p L} \sqrt{\frac{1}{M}}, \quad (18)$$

where  $p \geq c_1(M-1) \sqrt{\frac{\log n}{\binom{n}{k}}}$ , and  $c_1$  is some numerical constant.

Let  $\|w\|_\infty = w_{\max} = 1$  for ease of demonstration. Suppose  $\Delta_K = w_K - w_{K+1} \geq \sqrt{\frac{\log n}{\binom{n}{k}} p L} \sqrt{\frac{1}{M}}$ . Then,

$$\begin{aligned} \hat{w}_i - \hat{w}_j &\geq w_i - w_j - |\hat{w}_i - w_i| - |\hat{w}_j - w_j| \\ &\geq w_K - w_{K+1} - 2\|w - w\|_\infty > 0, \end{aligned} \quad (19)$$

for all  $1 \leq i \leq K$  and  $j \geq K+1$ . That is, the top- $K$  items are identified as desired. Hence, as long as  $\Delta_K \geq \sqrt{\frac{\log n}{\binom{n}{k}} p L} \sqrt{\frac{1}{M}}$ , i.e.,  $\binom{n}{k} p L \geq \frac{n \log n}{2k} \frac{1}{M}$ , reliable top- $K$  ranking is achieved with the sample size of  $\frac{n \log n}{2k} \frac{1}{M}$ .

Now, let us prove Theorem 3. To find an  $\ell_\infty$  error bound, we first derive an upper bound on the point-wise error between the score estimate of item  $i$  and its true score, which consists of three terms:

$$\begin{aligned} |\hat{w}_i - w_i| &\leq |\hat{w}_i - w_i| \hat{P}_{ii} + \sum_{j:j \neq i} |w_j - w_j| \hat{P}_{ij} \\ &\quad + \left| \sum_{j:j \neq i} (w_i + w_j) (P_{ji} - \hat{P}_{ji}) \right|. \end{aligned} \quad (20)$$

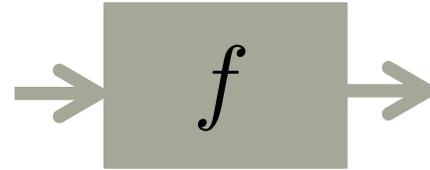
This can be obtained applying  $\hat{w} = P \hat{w}$  and  $w = P w$ . We obtain upper bounds on these three terms as follows.

$$\hat{P}_{ii} < 1, \quad (21)$$

$$\left| \sum_{j:j \neq i} (w_i + w_j) (P_{ji} - \hat{P}_{ji}) \right| \leq \sqrt{\frac{n \log n}{\binom{n}{k}} p L} \sqrt{\frac{1}{M}}, \quad (22)$$

$$\sum_{j:j \neq i} |w_j - w_j| \hat{P}_{ij} \leq \sqrt{\frac{n \log n}{\binom{n}{k}} p L} \sqrt{\frac{1}{M}} \quad (23)$$

with high probability (see Lemmas 1, 2 and 3 in the supplementary for details). One can see that the inequalities (21)



## Abstract

Given a sample of instances with binary labels, the top ranking problem is to produce a ranked list of instances where the *head* of the list is dominated by positives. Popular existing approaches to this problem are based on surrogates to a performance measure known as the fraction of positives of the top (PTop). In this paper, we show that the measure and its surrogates have an undesirable property: for certain noisy distributions, it is optimal to trivially predict the *same score for all instances*. We propose a simple rectification of the measure which avoids such trivial solutions, while still focussing on the head of the ranked list and being as easy to optimise.

# AUS DATEN LERNEN

*Statt eine Lösung direkt (algorithmisch oder wissensbasiert) zu entwickeln, ist es einfacher ...*

- **Beispiele** zu zeigen und diese vom System **generalisieren** zu lassen



# AUS DATEN LERNEN

*Statt eine Lösung direkt (algorithmisch oder wissensbasiert) zu entwickeln, ist es einfacher ...*

- **Beispiele** zu zeigen und diese vom System **generalisieren** zu lassen



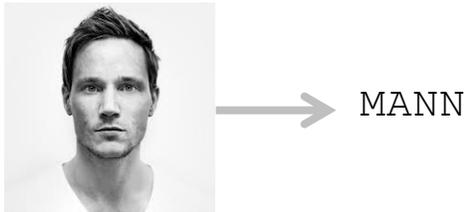
- das System **probieren** zu lassen und **Feedback** zu geben



# AUS DATEN LERNEN

*Statt eine Lösung direkt (algorithmisch oder wissensbasiert) zu entwickeln, ist es einfacher ...*

- **Beispiele** zu zeigen und diese vom System **generalisieren** zu lassen



- das System **probieren** zu lassen und **Feedback** zu geben



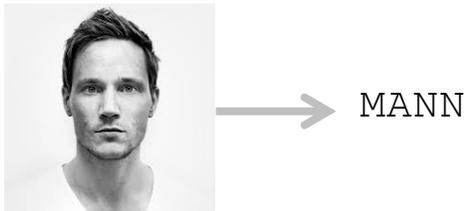
- etwas **vorzuführen** und das System **imitieren** zu lassen



# AUS DATEN LERNEN

*Statt eine Lösung direkt (algorithmisch oder wissensbasiert) zu entwickeln, ist es einfacher ...*

- **Beispiele** zu zeigen und diese vom System **generalisieren** zu lassen



→ *supervised learning*

- das System **probieren** zu lassen und **Feedback** zu geben



→ *reinforcement learning*

- etwas **vorführen** und das System **imitieren** zu lassen



→ *imitation learning*

# TRAINIEREN STATT PROGRAMMIEREN

☰ SPIEGEL ONLINE DER SPIEGEL SPIEGEL TV



Der selbstfahrende SUV im Video:



SPIEGEL ONLINE



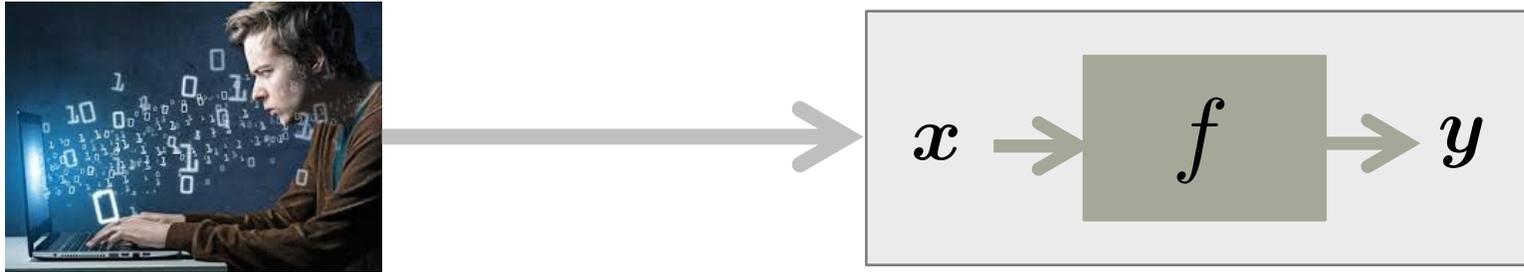
Das mathematische Geflecht dahinter ist kompliziert, aber das Ergebnis ein Fortschritt bei der Entwicklung: Trainieren statt programmieren. Diese Methode wird als Maschinelles Lernen oder Deep Learning bezeichnet.

ANZEIGE

Eurojackpot

Spiegel Online,  
5. Februar 2017

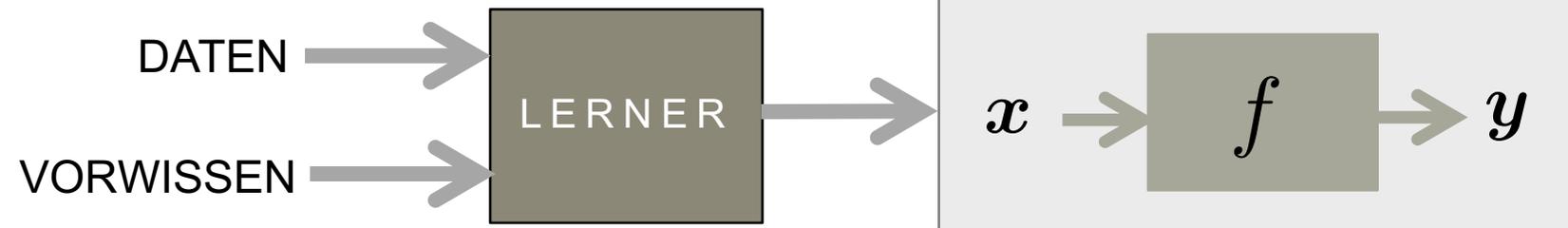
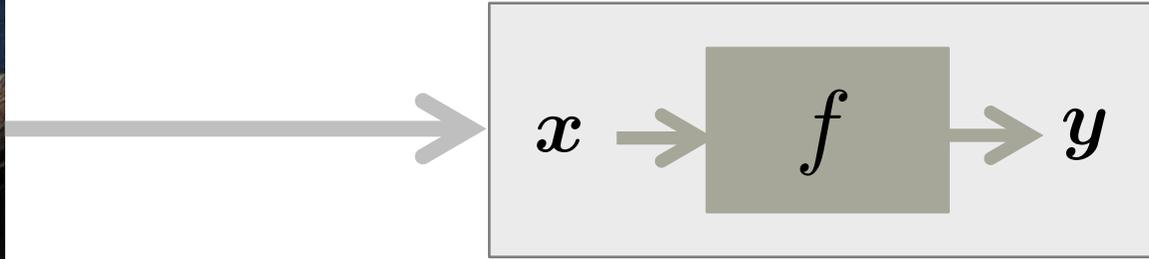
# AUS DATEN LERNEN



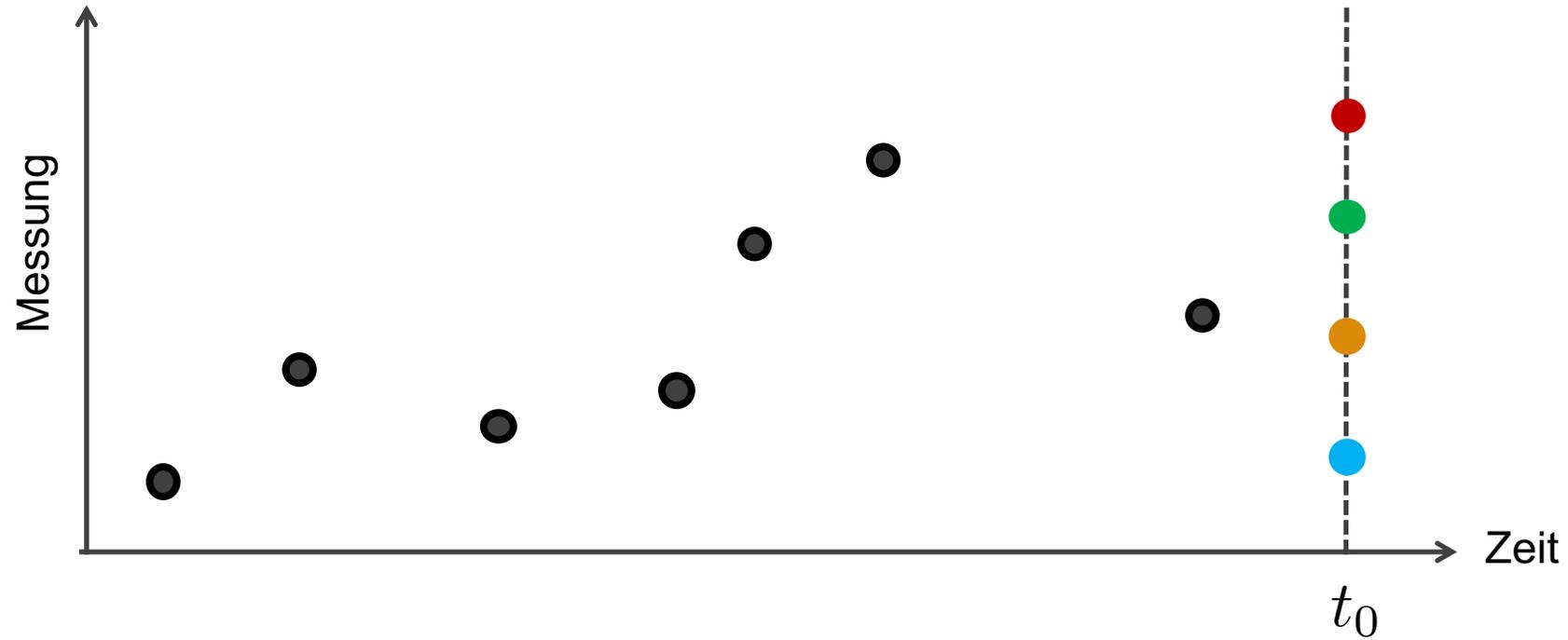
*“Machine learning is the science of getting computers to act without being explicitly programmed.”*

Andrew Ng, 2013

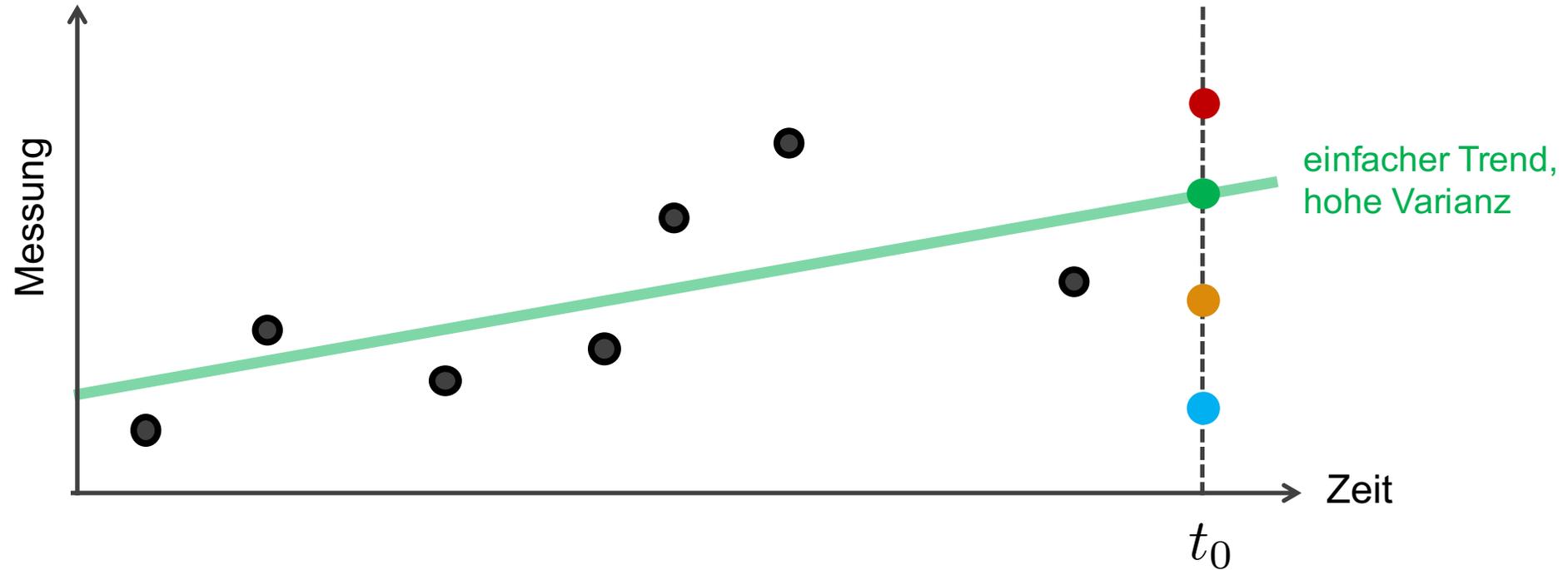
# AUS DATEN LERNEN



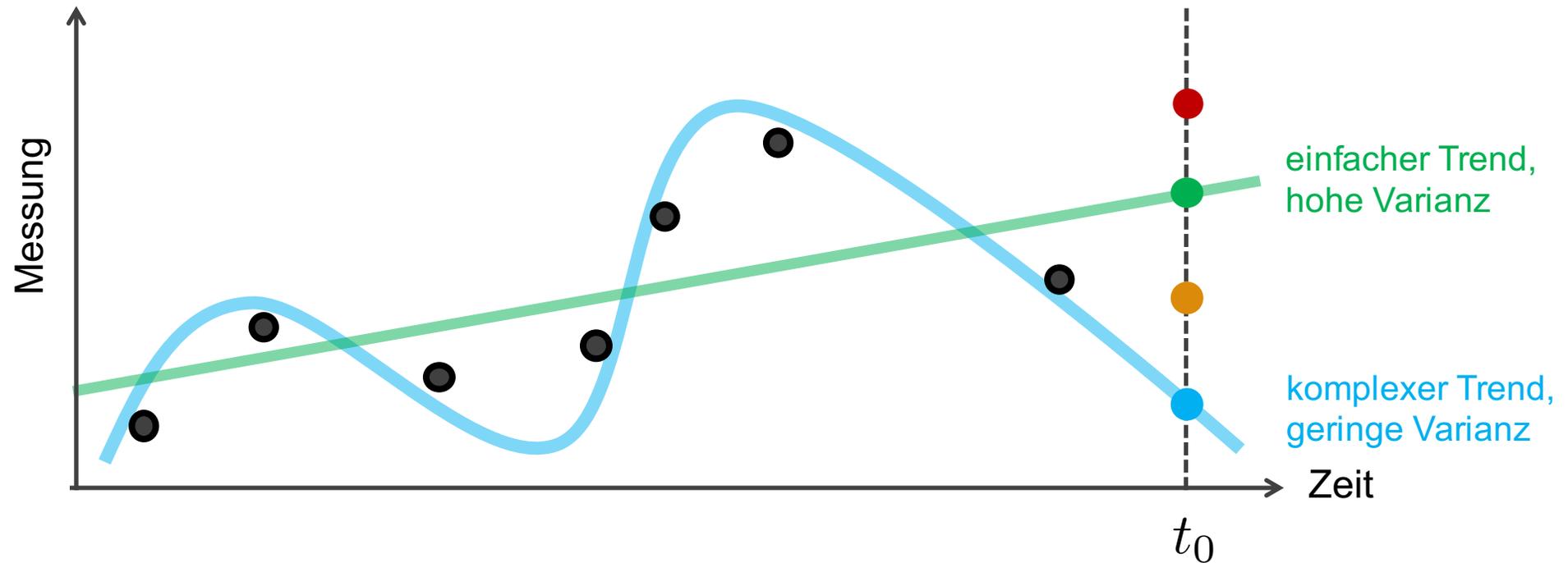
# GENERALISIERUNG UND INDUKTION



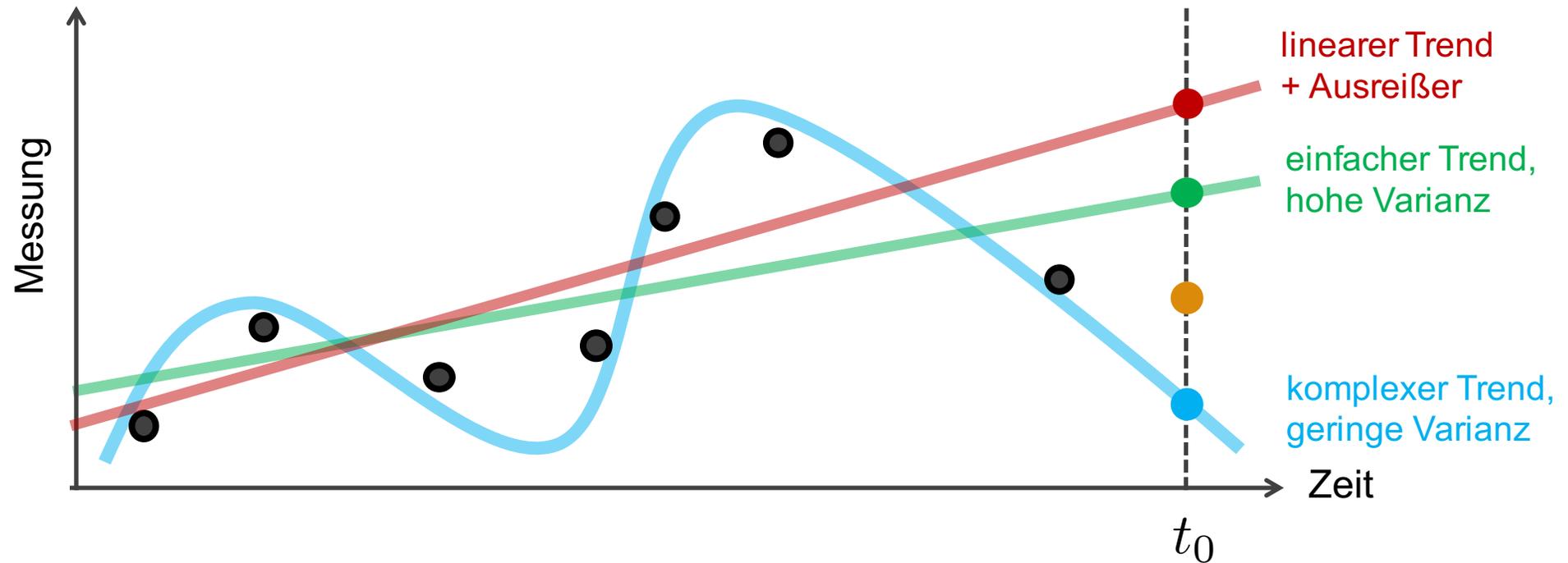
# GENERALISIERUNG UND INDUKTION



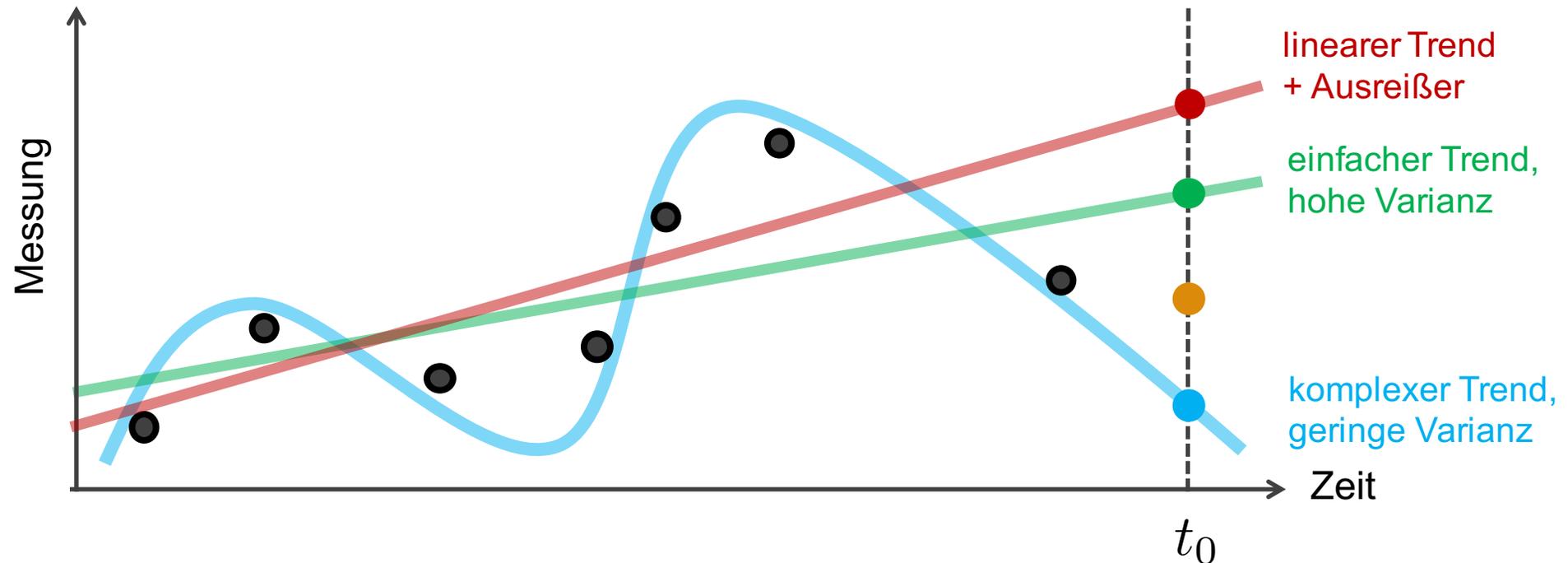
# GENERALISIERUNG UND INDUKTION



# GENERALISIERUNG UND INDUKTION

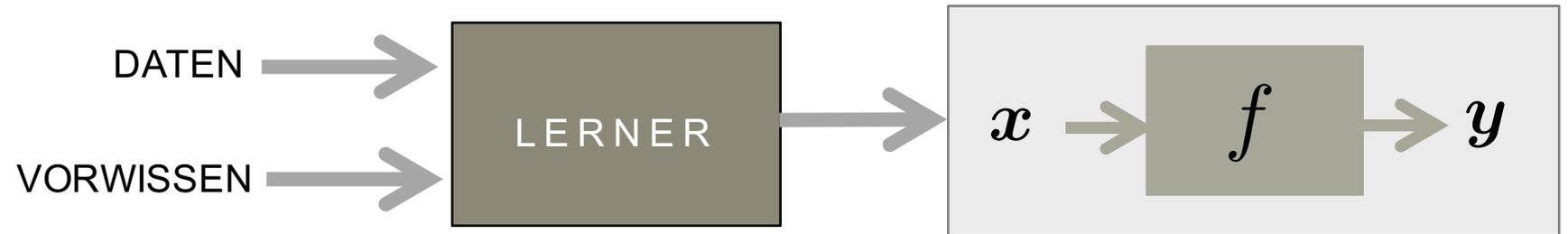


# GENERALISIERUNG UND INDUKTION

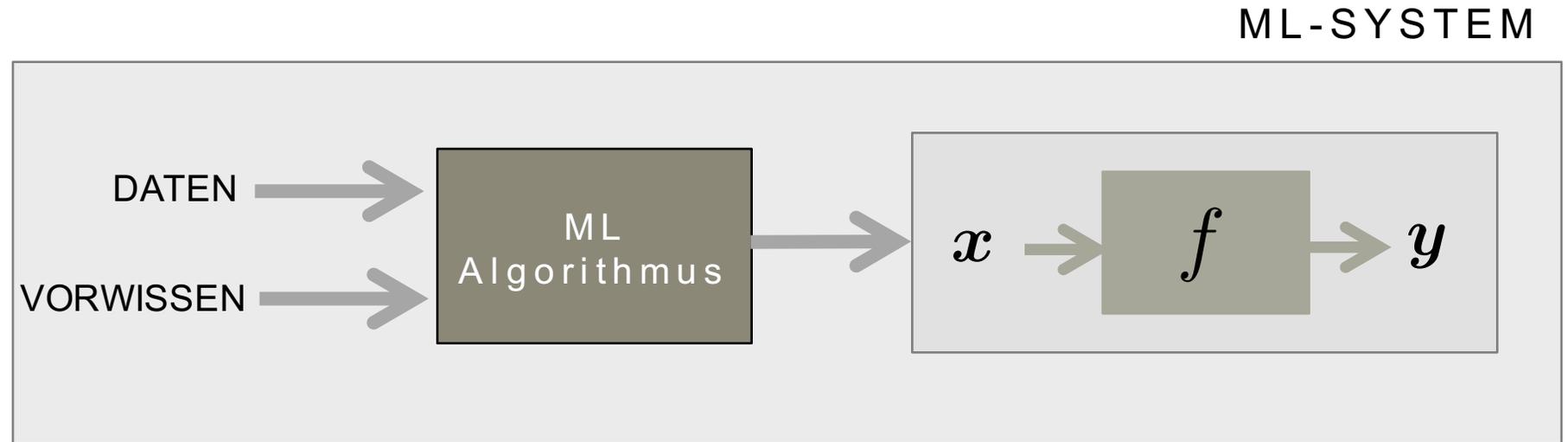


- Daten lassen sich in der Regel sehr unterschiedlich erklären (und entsprechend verallgemeinern).
- Jeder **Lernalgorithmus** implementiert (implizit oder explizit) gewisse **Modellannahmen**, die den Prozess der Generalisierung beeinflussen und zu unterschiedlichen Ergebnissen führen.

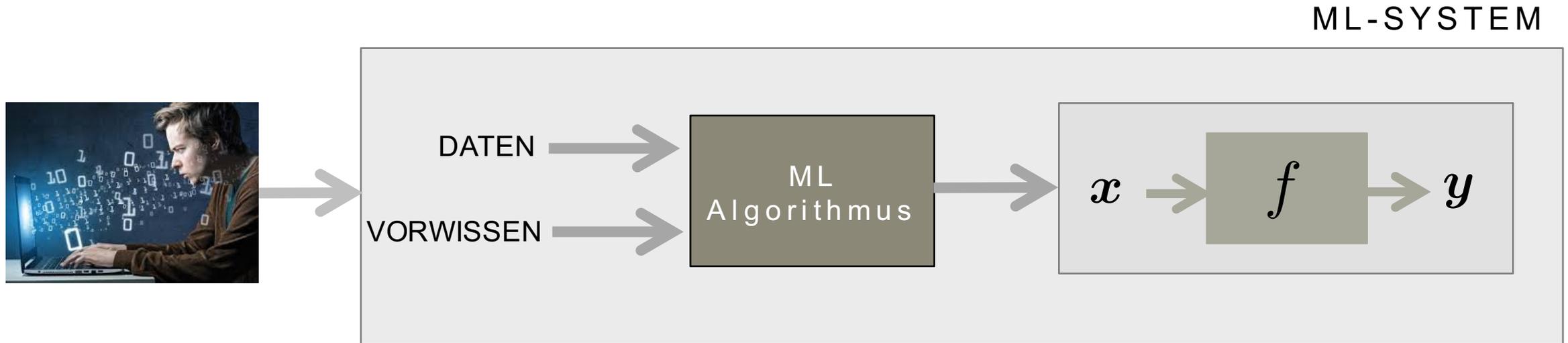
# ML-BASIERTES PROBLEMLÖSEN



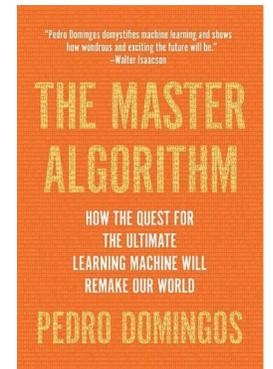
# ML-BASIERTES PROBLEMLÖSEN



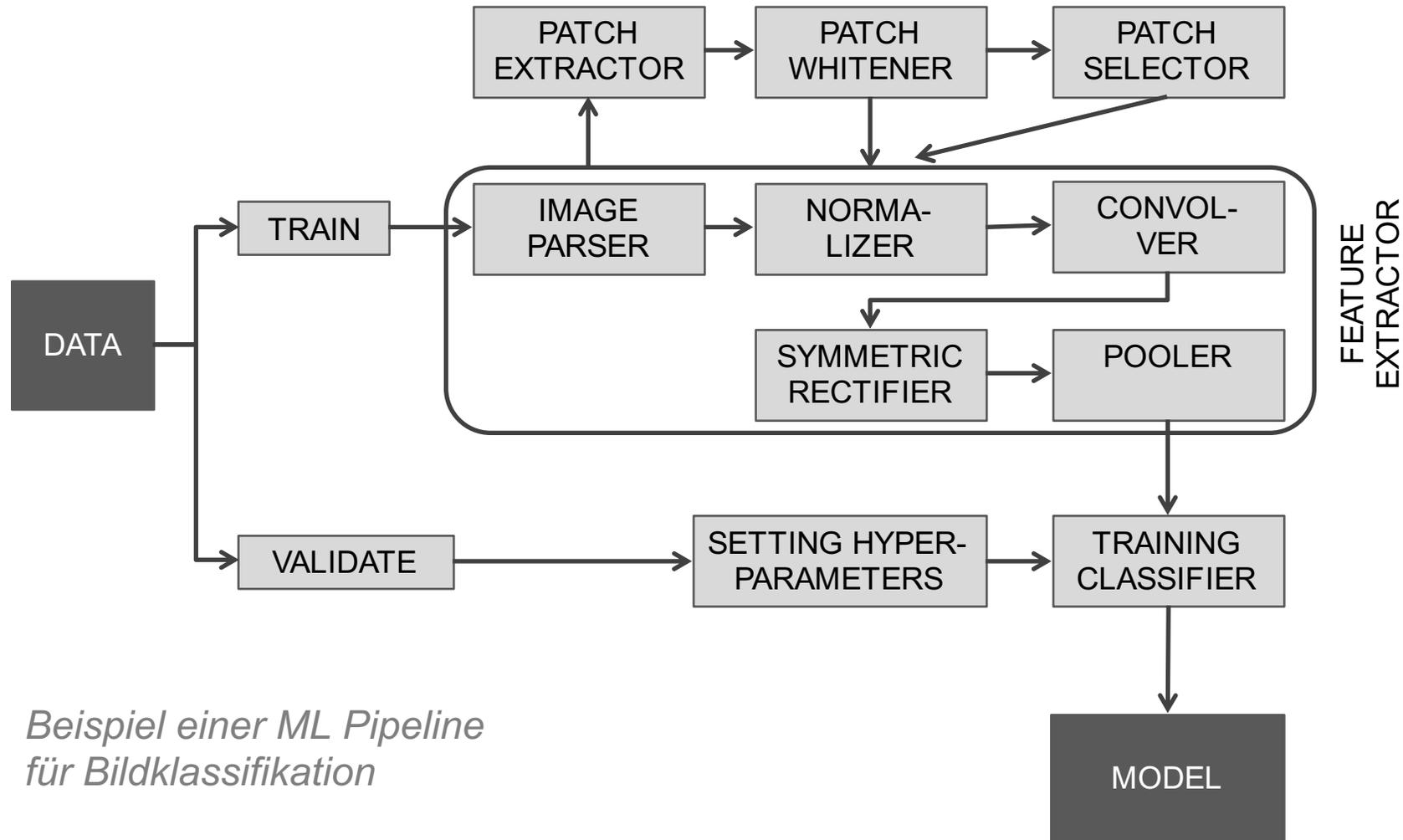
# ML-BASIERTES PROBLEMLÖSEN



- Algorithmus wird nicht mehr für das ursprüngliche Problem benötigt, aber für das **Problem, wie man lernt, dieses Problem zu lösen.**
- Der eigentliche Algorithmus wird mithilfe eines **“Meta-Algorithmus”** unter **Verwendung von Daten** erzeugt.
- Datengetriebene Algorithmenentwicklung bzw. datengetriebenes Problemlösen.



# ML-BASIERTES PROBLEMLÖSEN

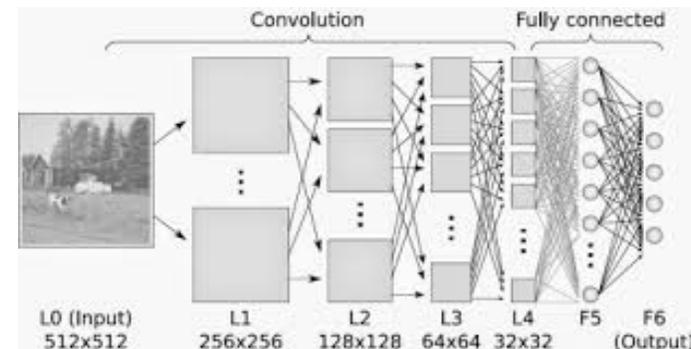


*Beispiel einer ML Pipeline für Bildklassifikation*

# ML-BASIERTES PROBLEMLÖSEN

Ein tiefes neuronales Netz (CNN) kann leicht mehr als 40 Hyper-Parameter haben:

- number of hidden units
- activation function
- convolution kernel width
- implicit zero padding
- weight decay coefficient
- loss function
- weight initialization
- learning rate
- batch size
- dropout rate
- ...



*Das praktische Lösen eines ML Problems geht einher mit der expliziten oder impliziten Festlegung tausender Freiheitsgrade ...*

# DIE EVOLUTION INTELLIGENTER SYSTEME

```
function GetMin(var a: TList)
var
  i, min, mini: integer;
begin
  min := MaxInt;
  mini := 0;
  for i := 1 to a.len do
    if a.arr[i].G < min t
    begin
      min := a.arr[i].G
      mini := i;
    end;
  end;
  GetMin := mini;
end;
```

klassische  
Programmierung

*... ist schwierig  
für komplexe  
Probleme*

```
mann(adam).
mann(tobias).
mann(frak).
frau(eva).
frau(daniela).
frau(ulrike).
vater(adam,tobias).
vater(tobias,frak).
vater(tobias,ulrike).
mutter(eva,tobias).
mutter(daniela,frak).
mutter(daniela,ulrike).
```

Wissensbasierte  
Systeme

*... leidet unter dem  
Flaschenhals des  
Wissenserwerbs*

```
# Spot Check Algorithms
models = []
models.append('LR', Logis)
models.append('LDA', Line)
models.append('KNN', KNei)
models.append('CART', Dec)
models.append('NB', Gauss)
models.append('SVM', SVCC)
# evaluate each model in t
results = []
names = []
for name, model in models:
  kfold = model_selectio
  cv_results = model_sel
  results.append(cv_resu
  names.append(name)
  msg = "%s: %f (%f)" %
  print(msg)
```

datengetriebenes  
Problemlösen

*... erfordert  
hohes Maß an  
Expertise in ML*



automatisiertes  
maschinelles Lernen

# AUTOMATISIERTES MASCHINELLES LERNEN

THE HUFFINGTON POST  
INFORM • INSPIRE • ENTERTAIN • EMPOWER

POLITICS ENTERTAINMENT WELLNESS WHAT'S WORKING VOICES VIDEO ALL SEI

THE BLOG

## Machine Learning as a Service: How Data Science Is Hitting the Masses

03/29/2016 02:43 pm ET | Updated Mar 29, 2016

f t p in ✉

Like 248

**Laura Dambrosio**   
Writer, entrepreneur, tech enthusiast

# AUTOMATISIERTES MASCHINELLES LERNEN

THE HUFFINGTON POST  
INFORM • INSPIRE • ENTERTAIN • EMPOWER

POLITICS ENTERTAINMENT WELLNESS WHAT'S WORKING VOICES VIDEO ALL SEI

THE BLOG

## Machine Learning as a Service: How Data Science Is Hitting the Masses

03/29/2016 02:43 pm ET | Updated Mar 29, 2016

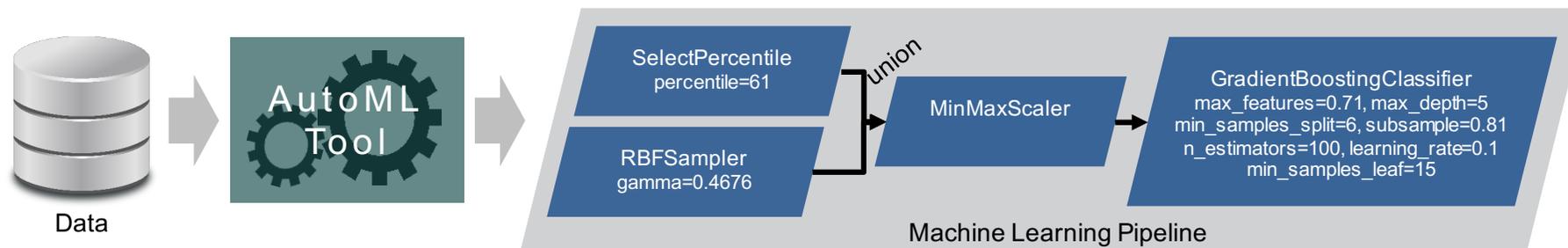
f t p in e m

Like 248

Laura Dambrosio  
Writer, entrepreneur, tech enthusiast



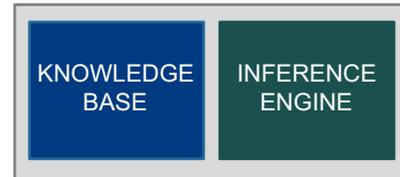
- **AutoML:** Automatisierte Auswahl und Parametrisierung von ML-Algorithmen sowie Kombination zu einer Gesamtlösung, speziell zugeschnitten auf ein Problem (einen Datensatz).
- Einige **AutoML Tools** existieren bereits, weitere werden derzeit (*von Informatikern*) entwickelt.



# ZUSAMMENFASSUNG

```
ALGORITHM shortest-path(V,T)
W := {v}
ShortDist[v] := 0
FOR each u in V - {v}
ShortDist[u] := T[v],u
WILE W /= V
MinDist := INFINITE
FOR each v in V - W
IF ShortDist[v] < MinDist
MinDist = ShortDist[v]
W := v
END {if}
END {for}
W := W U {w}
FOR each u in V - W
ShortDist[u] := Min(ShorDis[u],ShortDist[w] + T[w,u])
END {while}
```

klassische  
Programmierung



Wissensbasierte  
Systeme



→ MANN

datengetriebenes  
Problemlösen



automatisiertes  
maschinelles Lernen

1

**Daten** werden immer zentraler (wissensbasiertes vs. datengetriebenes Problemlösen).

2

**Wissen** bleibt wichtig aber wird abstrakter (Domänenwissen vs. ML-Wissen).

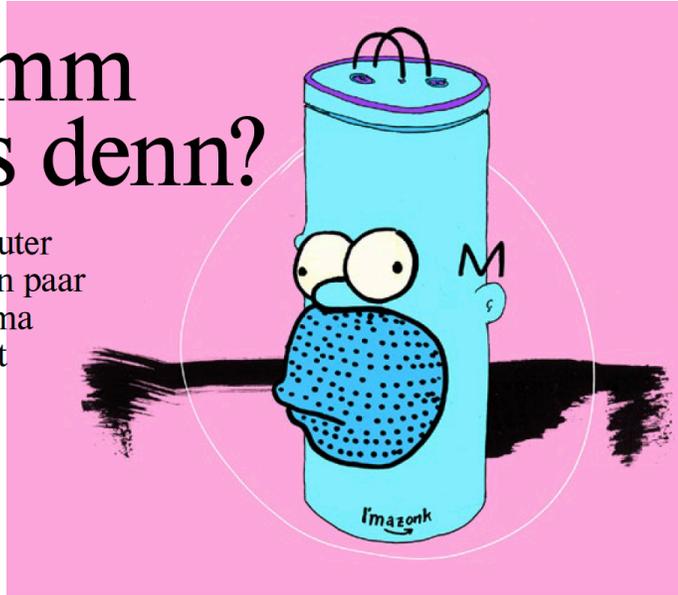
3

**Algorithmen** werden generischer (direkte Lösung, ML, AutoML, Meta-Lernen).

# KÜNSTLICHE VERSUS MENSCHLICHE INTELLIGENZ

## Wie dumm ist das denn?

So schlau sind Computer auch wieder nicht. Ein paar Anekdoten zum Thema künstliche Dummheit



Fluter, Herbst 2018, Nr. 68

**1** Der US-Konzern Microsoft wollte 2016 zeigen, wie clever lernende Systeme sein können – stattdessen bewies er mit seinem Twitter-Bot „Tay“ unfreiwillig das Gegenteil. Die ursprüngliche Idee war, dass der Bot durch Chats mit anderen Usern dazulernt. Je mehr er mit anderen interagiert, desto mehr kann er sich abschauen und desto schlauer wird er. Das war die Theorie. Letztlich dauerte es aber weniger als 24 Stunden, bis „Tay“ zur Rassismusschleuder wurde. „Hitler was right I hate the jews“, twitterte der Bot, leugnete den Holocaust und verbreitete Verschwörungstheorien zu den Terroranschlägen auf das World Trade Center in New York. Microsoft schaltete den Bot schnell wieder ab.

**2** Der Hautarzt Roberto Novoa baute kürzlich mit Kollegen an der US-amerikanischen Uni Stanford einen Algorithmus, der mithilfe einer Datenbank aus 129.000 Bildern von Hautveränderungen automatisch erkennen sollte, welche davon gutartig und welche gefährliche Tumore sind. Das Problem dabei: Bei besorgniserregenden Hautveränderungen legen Dermatologen oft ein Lineal mit ins Foto, um deren Größe zu dokumentieren. Das führte dazu, dass der Algorithmus auf Bildern immer dann Tumore erkannte, wenn im Bild ein Lineal zu sehen war – weil dessen Vorhandensein mit einer größeren Wahrscheinlichkeit einherging, dass es sich um krebsartige Veränderungen handelte. Immerhin hatte Novoa also ein Erkennungswerkzeug für Lineale gebaut.

**3** Forscher am MIT Lincoln Lab testeten 2013 ein Computerprogramm, das lernen sollte, eine Liste von Nummern zu sortieren. Es erreichte schließlich tatsächlich einen perfekten Score. Um das zu schaffen, löschte das Pro-

gramm einfach die Liste. Das entsprach den Anforderungen, wenn auch nicht ganz wie gedacht: Keine Nummern mehr, keine Unordnung.

**4** So ein großes Universum und kaum Zeit, es sich genau anzuschauen. Ein NASA-Forscherteam berichtete 2017 von einem Deep-Learning-Programm, das Sonnenstürme auf Bildern erkennen sollte. Das genaueste Programm war letztlich eines, das grundsätzlich anzeigte, dass auf einem Bild kein Sonnensturm zu sehen ist. Starke Sonnenstürme sind nämlich sehr selten.

**5** „Alexa, kauf mir ein Puppenhaus!“ Eine Sechsjährige in den USA bestellte im vergangenen Jahr per Sprachsteuerung von Amazon Echo neues Spielzeug. Das ärgerte ihre Eltern, und der örtliche Lokalsender in San Diego berichtete darüber. Blöd nur, dass der Reporter die Worte des Kindes live wiederholte. Das aktivierte nämlich bei zahlreichen Zuschauern die Sprachsteuerung von Amazon Echo und löste ungewöhnlich viele Bestellungen von Puppenhäusern im Raum San Diego aus.

**6** Sicherheit? Kann man auch mit Robotern machen, dachte sich im vergangenen Jahr eine Sicherheitsfirma in Washington, D.C.. Sie ließ einen Sicherheitsroboter durch ein Bürogebäude patrouillieren, angelehnt ans Äußere des „Star Wars“-Droiden R2D2, aber bestückt mit Kamera und Sensoren. Offenbar gefiel dem Roboter sein Dasein jedoch nicht: Er ertränkte sich in dem Brunnen im Foyer. Ein weiteres Modell des Roboters hatte 2016 in einer Einkaufsmall bereits ein 16 Monate altes Kind umgefahren – und hielt danach nicht einmal an. ←  
Von Arne Semsrott

# KÜNSTLICHE VERSUS MENSCHLICHE INTELLIGENZ

- Derzeitige KIs werden (mit vielen Daten) auf sehr spezialisierte, vorab präzise definierte Einzelaufgaben trainiert, ohne diese jedoch in ein größeres “Weltwissen” einzubetten.
- In vielen Anwendungen fehlt der “gesunde Menschenverstand” sowie ein gewisses Maß an Robustheit:

# KÜNSTLICHE VERSUS MENSCHLICHE INTELLIGENZ

- Derzeitige KIs werden (mit vielen Daten) auf sehr spezialisierte, vorab präzise definierte Einzelaufgaben trainiert, ohne diese jedoch in ein größeres “Weltwissen” einzubetten.
- In vielen Anwendungen fehlt der “gesunde Menschenverstand” sowie ein gewisses Maß an Robustheit:



# GESELLSCHAFTLICHE RELEVANZ DER KI

Künstliche Intelligenz

## Denn wir wissen nicht, wie sie's tun

Künstliche Intelligenz verheißt unglaublichen Fortschritt. Doch ein kniffliges Problem stellt sich gerade: Wie bestmögliche Menschen kann nicht immer erklärt werden, wie die verwendeten Algorithmen zu ihren Entscheidungen kommen. Und warum geht es...

23.08.2017, von STBYLLE ANDER

Teilen | Twittern | E-mailen



27. August 2017, 18:41 Uhr Software

## Sexismus im Algorithmus



Wenn das Programm das Foto einer Küche gezeigt bekam, assoziierte es damit automatisch "Frau". (Foto: dpa)

Wenn das Foto einer Küche automatisch mit "Frau" assoziiert wird: Immer wieder offenbaren Bilderkennungsprogramme sexistische Tendenzen. Zwei IT-Professoren wollen das Problem nun lösen.

ANZEIGE  
Delttek

### Bevormundung durch Technik

## Die Maschine will doch nur Ihr Bestes

Künstliche Intelligenz wird bald unseren Alltag durchdringen - und uns viele Entscheidungen abnehmen. Doch wir sollten uns dieser maschinellen Bevormundung nicht allzu dankbar beugen.

Eine Kolumne von **Sascha Lobo**

### Unfreiwillige Bevormundung

Die Bevormundung war im nicht dem 20. Jahrhundert noch ein Versprechen, im 21. Jahrhundert wurde sie durch die Digitalisierung auch zur Drohung werden. Mitte Mai 2017 wurde bekannt, dass die Ridesharing-Plattform Uber per KI vorhersagen

### Unwissentliche Bevormundung

Liesl Yearsley führte von 2007 bis 2014 ein Unternehmen, das künstlich konstruierte und schließlich von IBM gekauft wurde. Sie wollte wissen, wie schnell Menschen eine emotionale Beziehung zu einem Produkt aufbauen. Diesen Umstand hat der KI-Forscher Joshua B. Miller vor Sechzigern entdeckt, und er eignet sich für seine unwissentlichen Bevormundung, spricht: Manipulation

Gefahr der Bevormundung

### Freiwillige Bevormundung

Die stärkste Waffe des digitalen Paternalismus aber ist die Freiwilligkeit. Die Einsicht, dass es doch besser ist, genau jetzt auf die superintelligente Maschine zu hören. Längst ist normal, dass Algorithmen den Arbeitsalltag bestimmen: Lieferanten fahren so, wie es der Computer als ideal berechnet hat. In fünf Jahren wird es in vielen Branchen so ablaufen: Welche Aufgabe ist als nächste zu erledigen, welcher Text zu schreiben, welche Entscheidung zu treffen? Sie werden sich der Anweisung der höchst effizienten Maschine fügen.

Anmelden

## KULTUR

Nachrichten > Kultur > Gesellschaft > Neue deutsche Medien > Algorithmen > Ihre Kunst: Ein Horizont soweit die Daten reichen

### Empfehlungen durch Algorithmen

## Ein Horizont, so weit die Daten reichen

Filme, Musik, Bücher: Seit Jahren verlagern wir unseren Kulturkonsum immer weiter ins Netz und folgen algorithmisch erstellten Empfehlungen. Schränken wir damit unsere Entscheidungsfreiheit und unseren Horizont ein?

Von Dobrila Kontić

Einschränkung der Entscheidungsfreiheit

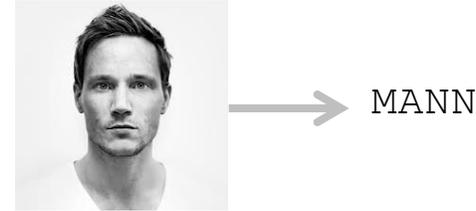
# ZUSAMMENFASSUNG

```
ALGORITHM shortest-path(V,T)
W := {v1}
ShortDist[v1] := 0
FOR each u in V - {v1}
ShortDist[u] := T[v1,u]
WILLE W /= V
  MinDist := INFINITE
  FOR each v in V - W
  IF ShortDist[v] < MinDist
  MinDist = ShortDist[v]
  W := v
  END {if}
  END {for}
  W := W U {w}
  FOR each u in V - W
  ShortDist[u] := Min(ShorDis[u],ShortDist[w] + T[w,u])
  END {while}
```

klassische  
Programmierung



Wissensbasierte  
Systeme



datengetriebenes  
Problemlösen



automatisiertes  
maschinelles Lernen

- 1 **Daten** werden immer zentraler (wissensbasiertes vs. datengetriebenes Problemlösen).
- 2 **Wissen** bleibt wichtig aber wird abstrakter (Domänenwissen vs. ML-Wissen).
- 3 **Algorithmen** werden generischer (direkte Lösung, ML, AutoML, Meta-Lernen).