Sie wird richten, sie wird dichten

–

Was kann KI (nicht)? Und wie?

18.06.2020

Marc Hauer, M.Sc.

Konstituierende Sitzung der Enquete-Kommission „Künstliche Intelligenz“ am 27.9.2018

Aus der Rede von Bundestagspräsidenten Dr. Schäuble:

- „Die künstliche Intelligenz gilt Vielen als neue Zauberformel des technischen Fortschritts, …
- … sie wird dichten, …
- … sie wird belohnen und bestrafen …“

https://www.bundestag.de/gebaerdensprache/mediathek?videoid=7272804#url=bWVkaWF0aGVrb3ZlcmxheT92aWRlb2lkPTcyNzI4MDQ=&mod=mediathek

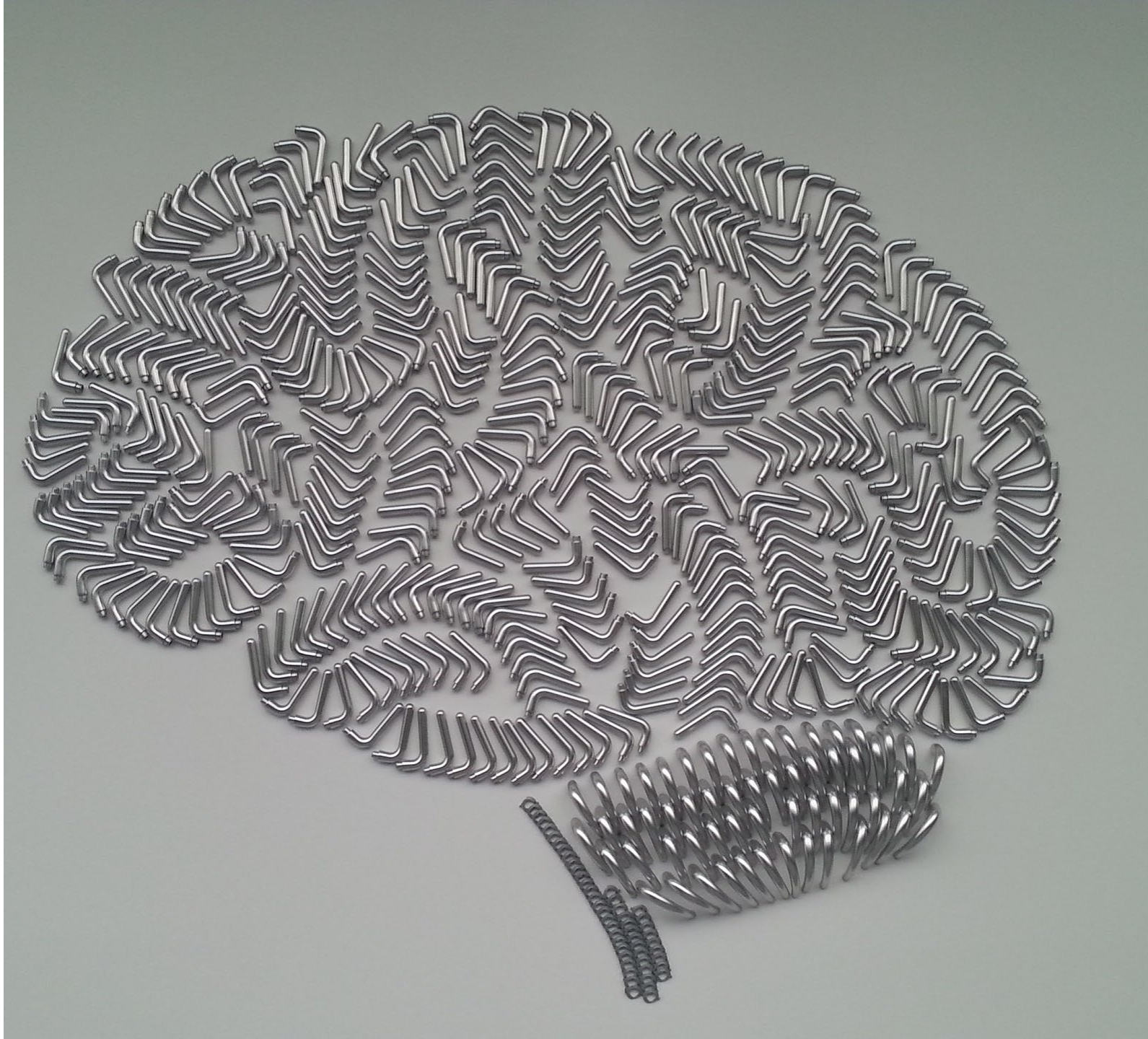
Ein Algorithmus ist…

…eine für jede **erfahrene Programmiererin ausreichend detaillierte Lösungsvorschrift**, so dass bei **korrekter Programmierung** der Computer **für jede korrekte Informationseingabe das richtige Ergebnis** berechnet – in endlicher Zeit.

Lernende Algorithmen

Kinder lernen…

- Durch **Rückkopplung**: unerwartet heiß, unerwartet kalt
- Durch **Speicherung in einer Struktur:** in Neuronen und deren Verknüpfung.
- Durch **Generalisierung des Gelernten**.

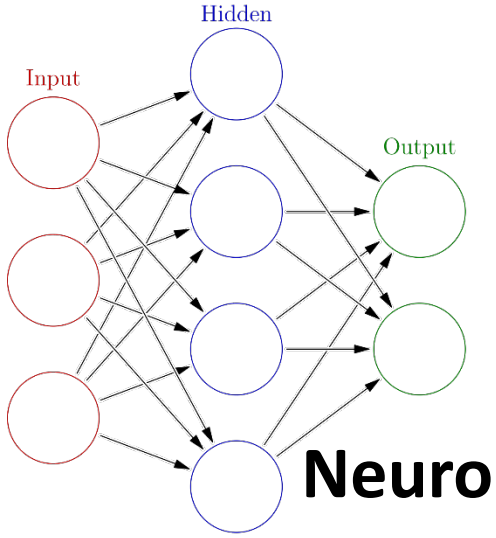


Computer lernen

Damit ein Computer lernen kann, benötigt er ebenfalls eine **Struktur,** um Gelerntes abzuspeichern.

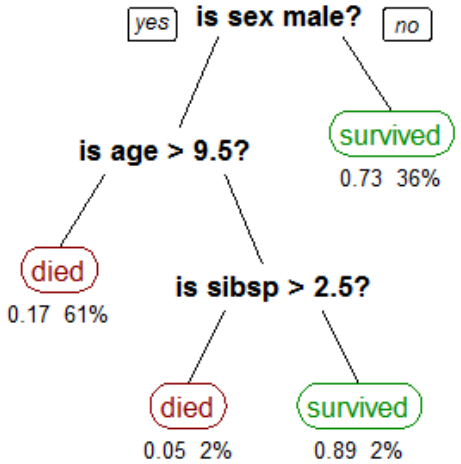
Optimal auch **Rückkopplung**.

Er lernt **generelle Regeln**.



Neuronales Netz

Entscheidungs-bäume

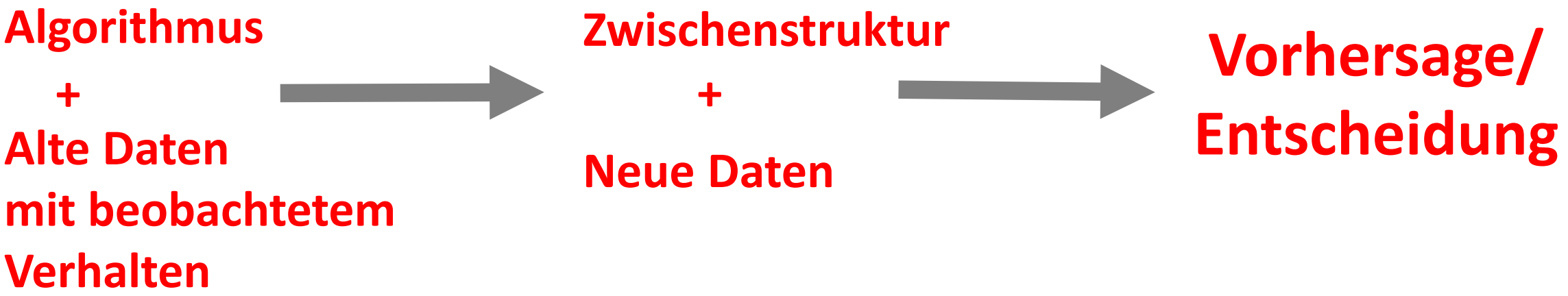


Formel

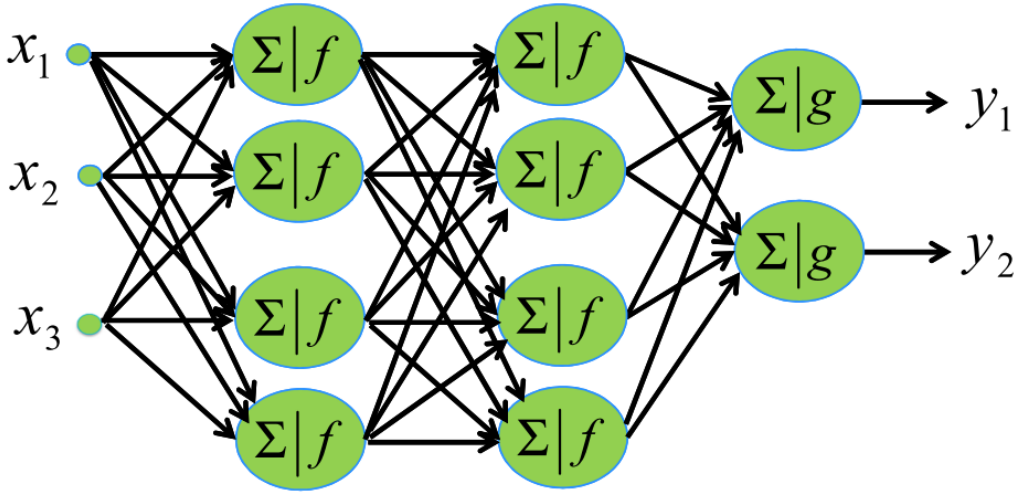
$$w_1 * \#Vh - w_2 * \#dayiVh + w_3 * I[g = male] * 1 + w_4 * I[T = R] * 1.0 + \dots$$

Künstliche Intelligenz

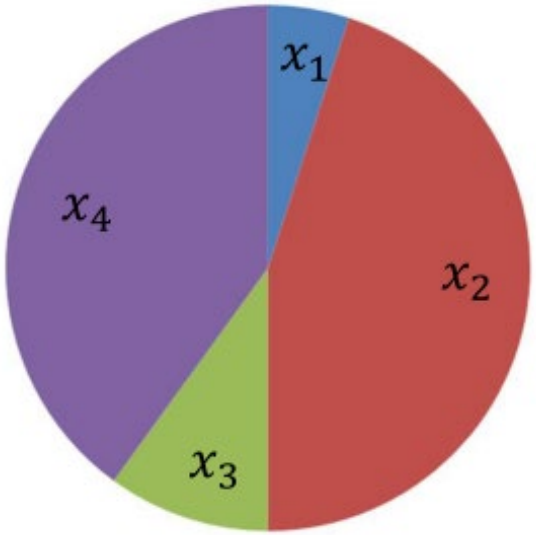
- **Problem:** gegeben eine Menge von bekannten Daten, finde Muster, die auf neuen Daten vorhersagen, wie sich etwas oder jemand verhalten wird.
- Algorithmus baut – basierend auf bekannten Daten – eine Zwischenstruktur auf, die dann Vorhersagen für neue Daten generiert.
- Der Algorithmus wird „auf den Daten trainiert“.



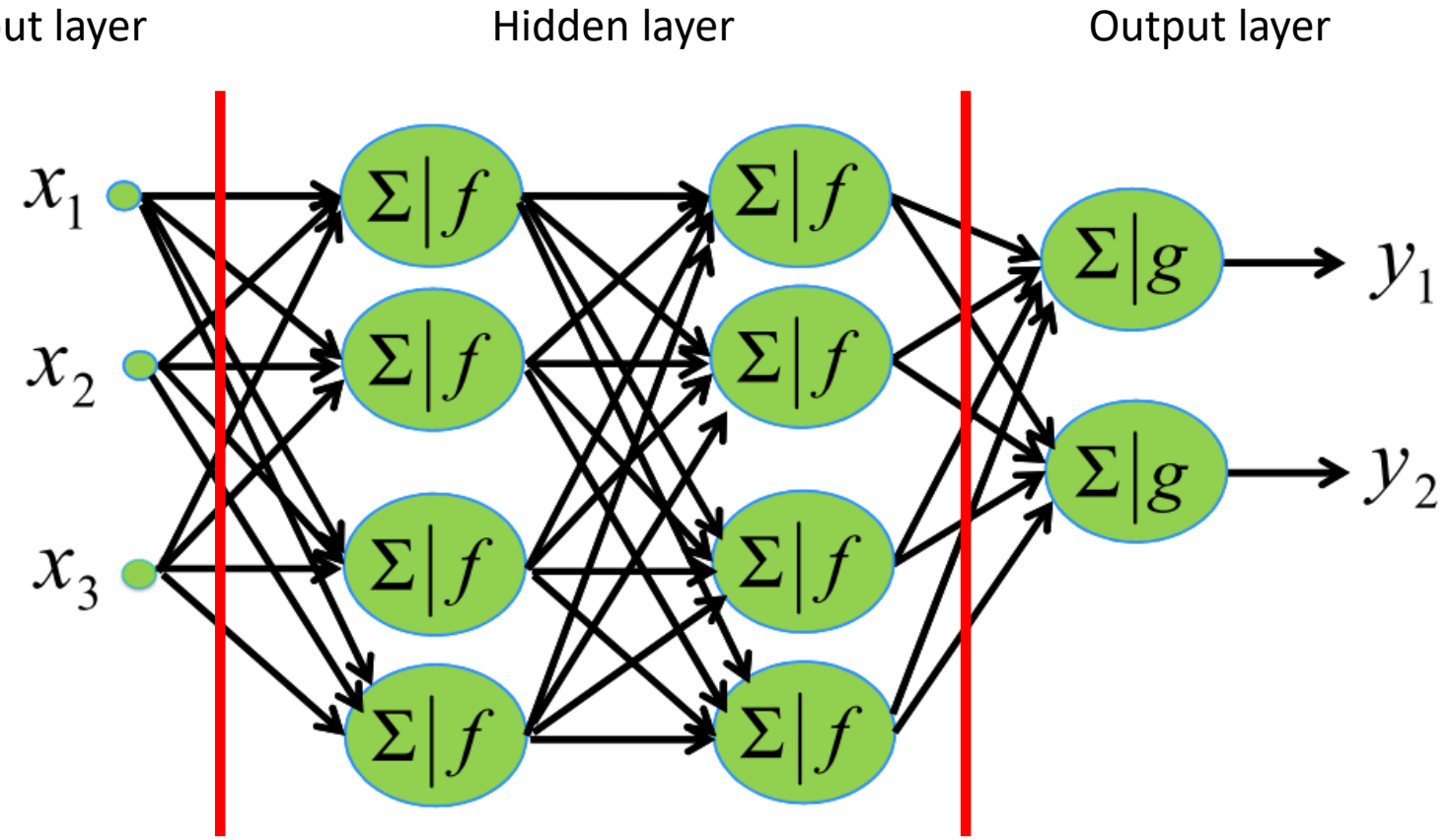
Künstliche Neuronale Netze



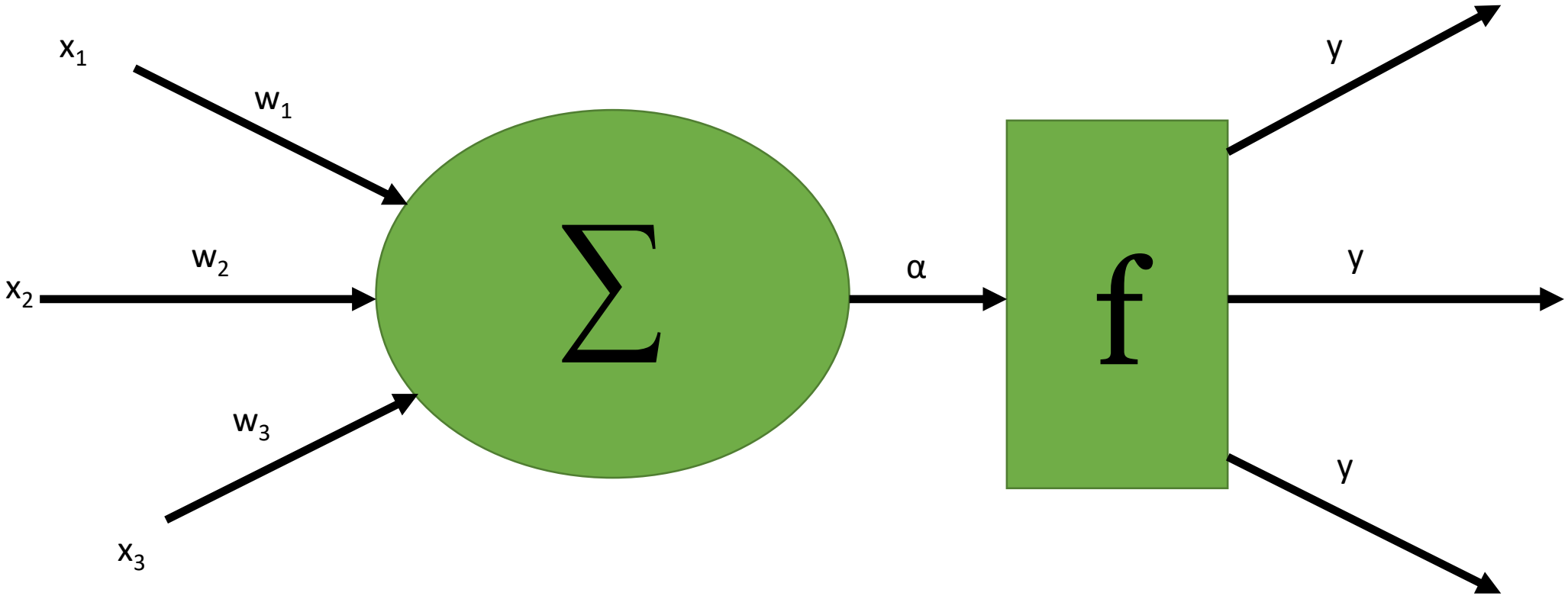
Genetische Algorithmen



Künstliche Neuronale Netze - Das Netz



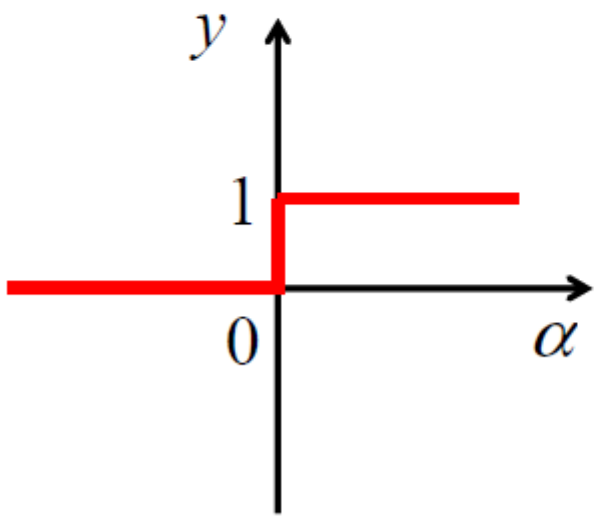
Künstliche Neuronale Netze - Neuronen und Verbindungen (Kanten)



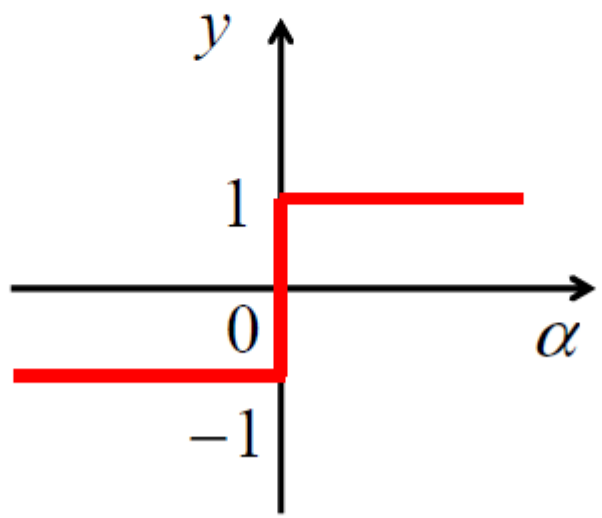
- Eingangswerte
- Kantengewichte
- Summenoperation
- Summenergebnis
- Aktivierungsfunktion
- Output

Künstliche Neuronale Netze - Typische Aktivierungsfunktionen

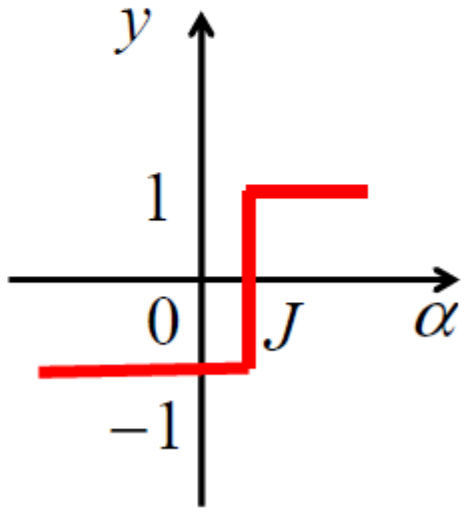
$$y = \begin{cases} 1, & \text{if } \alpha \geq 0 \\ 0, & \text{if } \alpha < 0 \end{cases}$$



$$y = \begin{cases} 1, & \text{if } \alpha \geq 0 \\ -1, & \text{if } \alpha < 0 \end{cases}$$



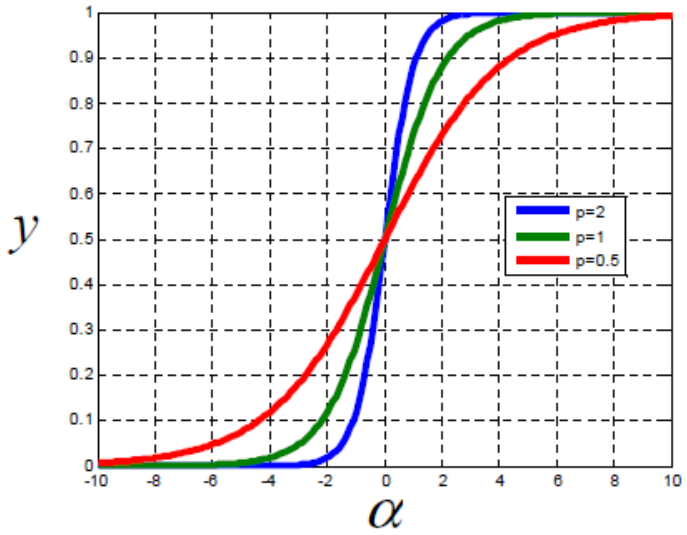
$$y = \begin{cases} 1, & \text{if } \alpha \geq J \\ -1, & \text{if } \alpha < J \end{cases}$$



Künstliche Neuronale Netze - Typische Aktivierungsfunktionen

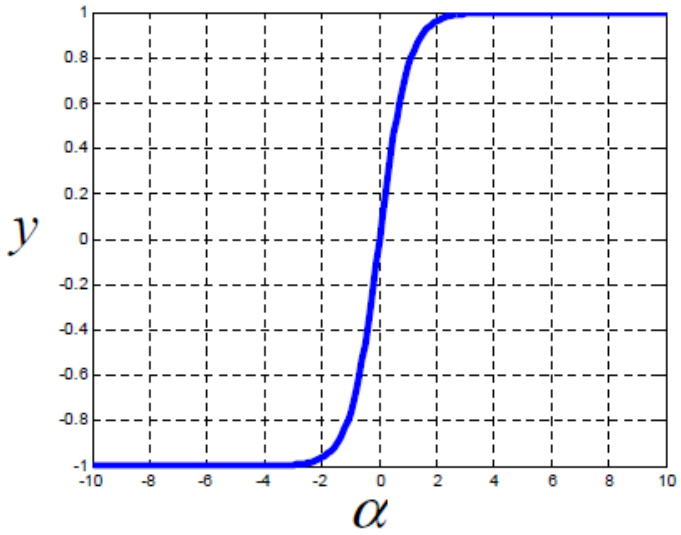
Sigmoid function

$$y = \frac{1}{1 + e^{-p\alpha}}$$

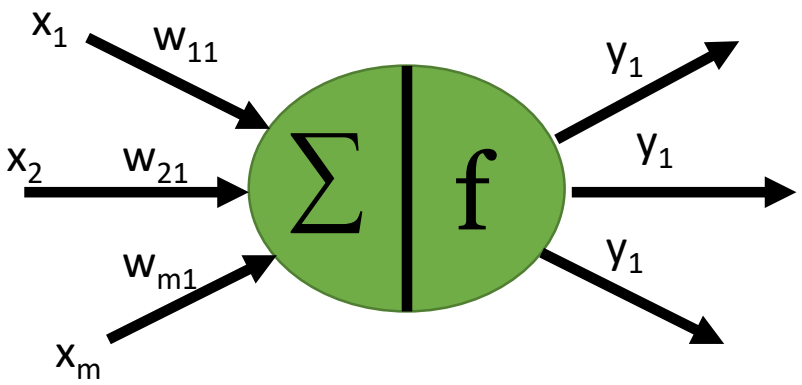


Hyperbolic tangent function

$$y = \frac{e^{\alpha} - e^{-\alpha}}{e^{\alpha} + e^{-\alpha}} = 1 - \frac{2}{e^{2\alpha} + 1}$$

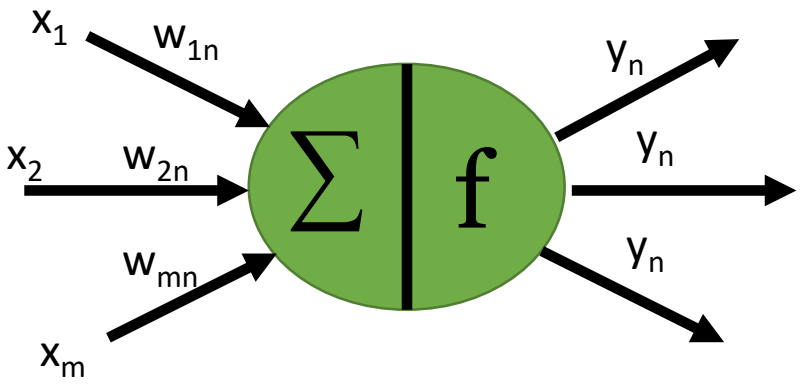


Künstliche Neuronale Netze - Neuronen und Verbindungen



$$x_1 * w_{11} + x_2 * w_{21} + \ldots + x_m * w_{m1} = \sum_{j=1}^{m} w_{j1} * x_j = \alpha_1$$

$$f(\alpha_1) = y_1$$



$$x_1 * w_{1n} + x_2 * w_{2n} + \ldots + x_m * w_{mn} = \sum_{j=1}^{m} w_{jn} * x_j = \alpha_n$$

$$f(\alpha_n) = y_n$$

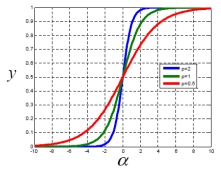
Künstliche Neuronale Netze - Beispiel: XOR

X_1	X_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Lichtschalterfunktion

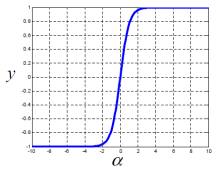
Sigmoid function

$$y = \frac{1}{1+e^{-p\alpha}}$$

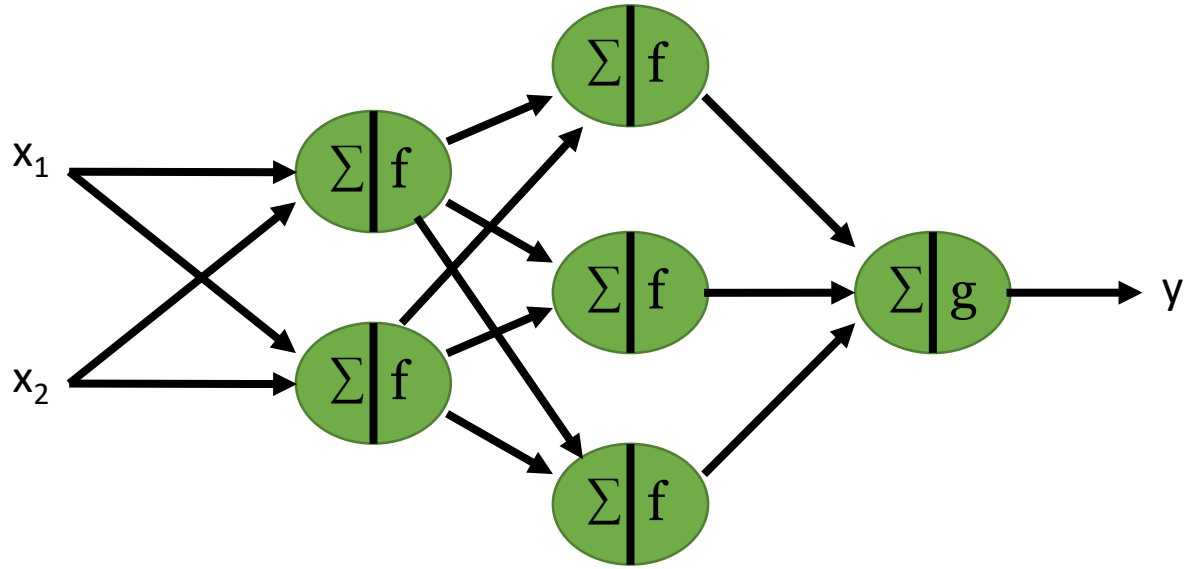


Hyperbolic tangent function

$$y = \frac{e^{\alpha}-e^{-\alpha}}{e^{\alpha}+e^{-\alpha}} = 1-\frac{2}{e^{2\alpha}+1}$$



$$g = \begin{cases} 1, & if\ \alpha > 1{,}5 \\ 0, & if\ \alpha \leq 1{,}5 \end{cases}$$

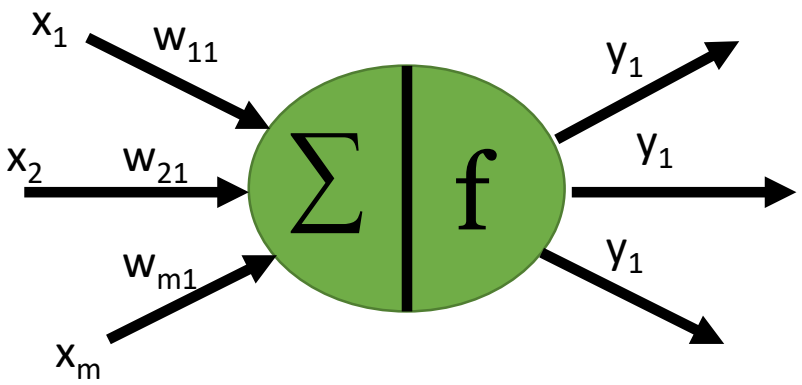


Künstliche Neuronale Netze - Backpropagation

Wiederhole für jede Iteration q:

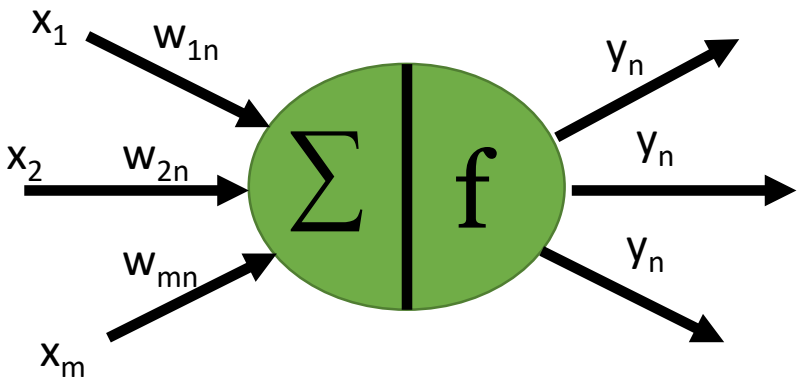
1. Berechne das Ergebnis y (output) des Neuronalen Netzes auf Basis der Eingabe x (input).
2. Berechne den Fehler e, z.B. als die Differenz zwischen dem gewünschten Ergebnis d und dem tatsächlichen Ergebnis y, also $e(q) = d(q) - y(q)$.
3. Berechne die lokalen Gradienten, beginnend beim Ausgang des Netzes.
4. Berechne die Aktualisierung der Gewichte auf Basis der Gradienten.
5. Aktualisiere die Gewichte.

Künstliche Neuronale Netze - Backpropagation 1: Feed forward



$$x_1 * w_{11} + x_2 * w_{21} + \ldots + x_m * w_{m1} = \sum_{j=1}^{m} w_{j1} * x_j = \alpha_1$$

$$f(\alpha_1) = y_1$$



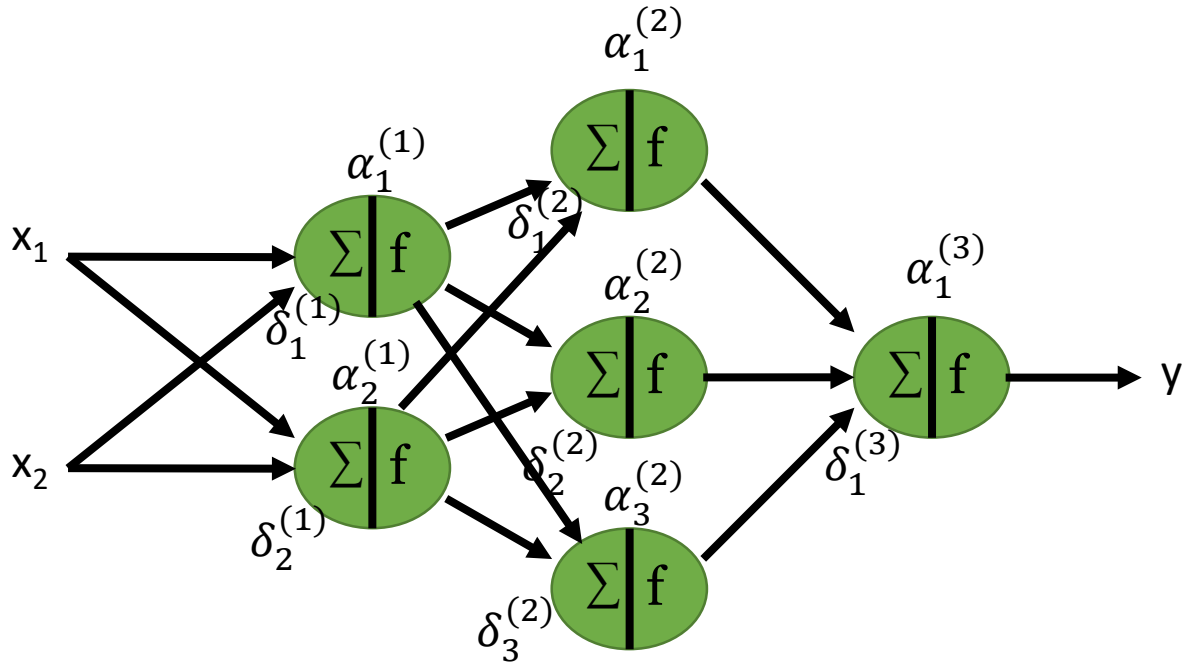
$$x_1 * w_{1n} + x_2 * w_{2n} + \ldots + x_m * w_{mn} = \sum_{j=1}^{m} w_{jn} * x_j = \alpha_n$$

$$f(\alpha_n) = y_n$$

Künstliche Neuronale Netze - Backpropagation 2: Fehler berechnen

Berechne den Fehler e, z.B. als die Differenz zwischen dem gewünschten Ergebnis d und dem tatsächlichen Ergebnis y, also $e(q) = d(q) - y(q)$.

Künstliche Neuronale Netze - Backpropagation 3: Lokale Gradienten



$$\delta_1^{(3)} = e * f'\left(\alpha_1^{(3)}\right)$$

$$\delta_1^{(2)} = \left(\sum_{n=1}^{1} w_{1n}^{(3)} * \delta_n^{(3)}\right) * f'\left(\alpha_1^{(2)}\right)$$

$$\delta_2^{(2)} = \left(\sum_{n=1}^{1} w_{2n}^{(3)} * \delta_n^{(3)}\right) * f'\left(\alpha_2^{(2)}\right)$$

$$\delta_3^{(2)} = \left(\sum_{n=1}^{1} w_{3n}^{(3)} * \delta_n^{(3)}\right) * f'\left(\alpha_3^{(2)}\right)$$

$$\delta_1^{(1)} = \left(\sum_{n=1}^{3} w_{1n}^{(3)} * \delta_n^{(2)}\right) * f'\left(\alpha_1^{(1)}\right)$$

$$\delta_2^{(1)} = \left(\sum_{n=1}^{3} w_{2n}^{(3)} * \delta_n^{(2)}\right) * f'\left(\alpha_2^{(1)}\right)$$

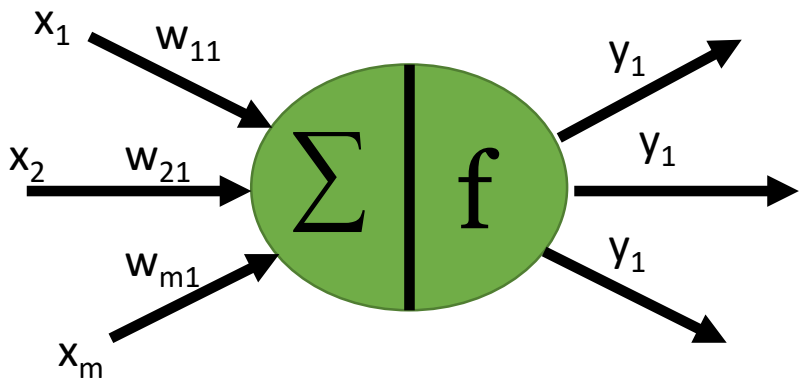
Künstliche Neuronale Netze - Backpropagation 4: Aktualisierung berechnen

Das Delta (die Veränderung) für ein Gewicht ergibt sich aus der Multiplikation der Lernrate η, dem lokalen Gradienten δ und Output y des vorherigen Neurons.

Die Lernrate η kann konstant, abhängig von der aktuellen Iteration oder abhängig von der aktuellen Güte des Netzes sein.

Je größer die Lernrate, desto stärker die Auswirkung auf die Gewichte

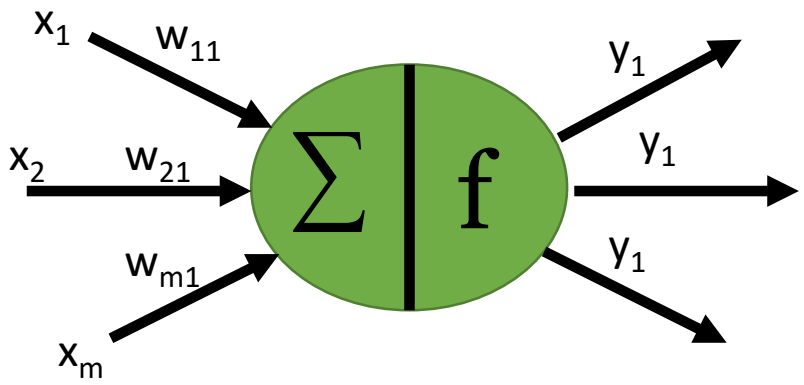
$$\Delta w_{mn}^{(s)} = \eta \delta_n^{(s)} y_m^{s-1}$$



Künstliche Neuronale Netze - Backpropagation 5: Aktualisierung anwenden

Das neue Gewicht ergibt sich durch das alte Gewicht addiert mit dem Delta für das Gewicht.

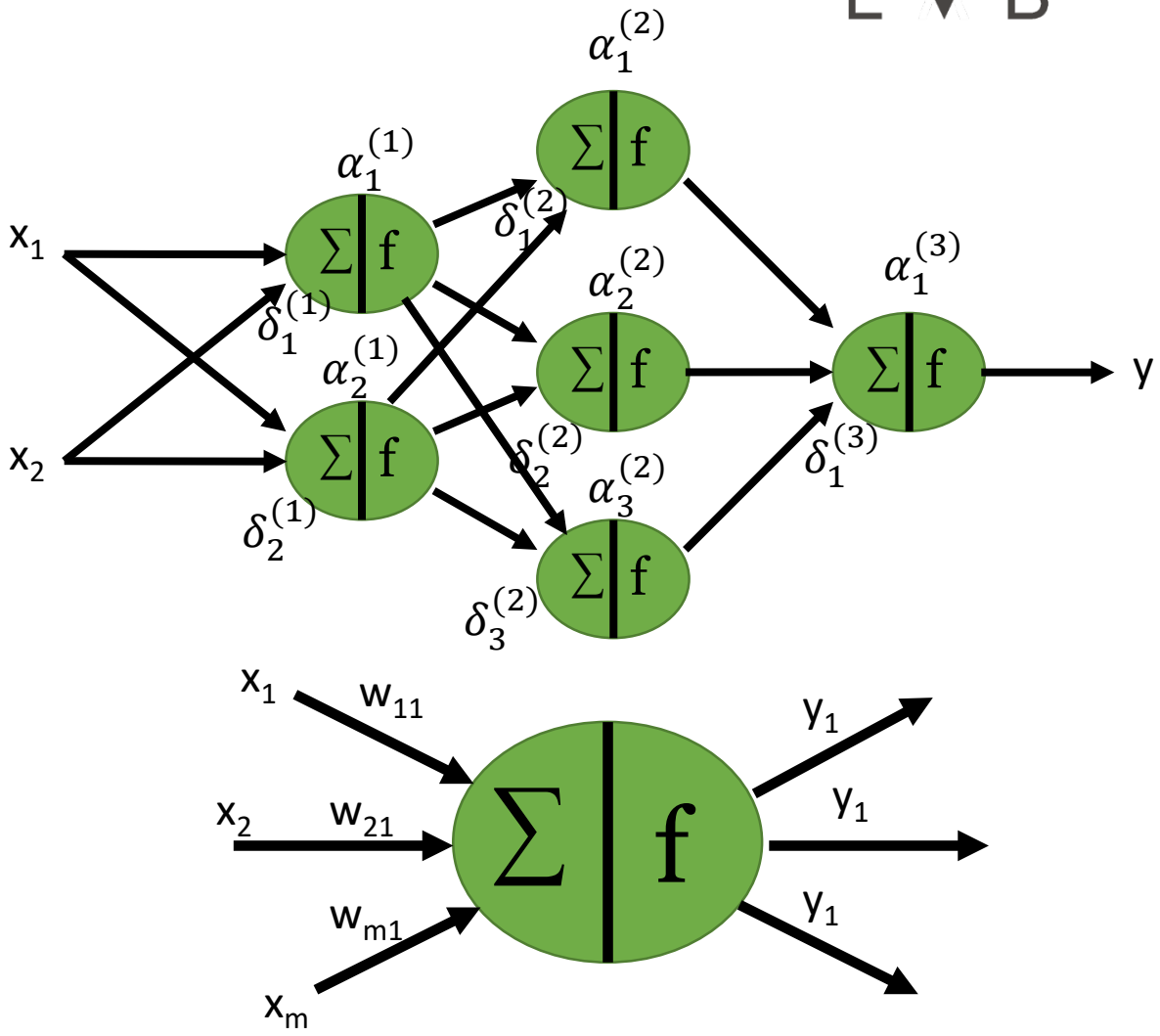
$$w_{mn,new}^{(s)} = w_{mn,alt}^{(s)} + \Delta w_{mn}^{(s)}$$



Künstliche Neuronale Netze - Übung macht den Meister

Wiederhole für jede Iteration q:

1. Berechne das Ergebnis y (output) des Neuronalen Netzes auf Basis der Eingabe x (input).
2. Berechne den Fehler e, z.B. als die Differenz zwischen dem gewünschten Ergebnis d und dem tatsächlichen Ergebnis y, also $e(q) = d(q) - y(q)$.
3. Berechne die lokalen Gradienten, beginnend beim Ausgang des Netzes.
4. Berechne die Aktualisierung der Gewichte auf Basis der Gradienten.
5. Aktualisiere die Gewichte.



Künstliche Neuronale Netze - Bewerberanalyse

- Begrenzte Anzahl Trainingsdaten
- Training mit Daten “wer wurde eingestellt” führt zu historischem Bias
- Training mit Daten “wer ist/war guter Mitarbeiter” gibt es nicht, bzw. können kaum fair sein
- Monopolproblematik
- Welche Eigenschaften dürfen verwendet werden?
- Welche Eigenschaften sollten (nicht) verwendet werden?

Genetische Algorithmen - Konzepte

Finden eines Optimums/Maximums/Minimums durch

- Natürliche Selektion (survival of the fittest)
- Vererbung
- Mutation

Genetische Algorithmen - Einfaches Beispiel

$$f = 2a_1(a_2 + 3a_3) - e^{-3a_4} - 3a_1a_3a_5 + 2a_6(a_1\text{-}a_5) + a_4e^{2a_6}$$

Gesucht sind Werte für die sechs Variablen, durch die das Funktionsergebnis maximiert wird. Jede Variable kann nur den Wert 0 oder 1 annehmen

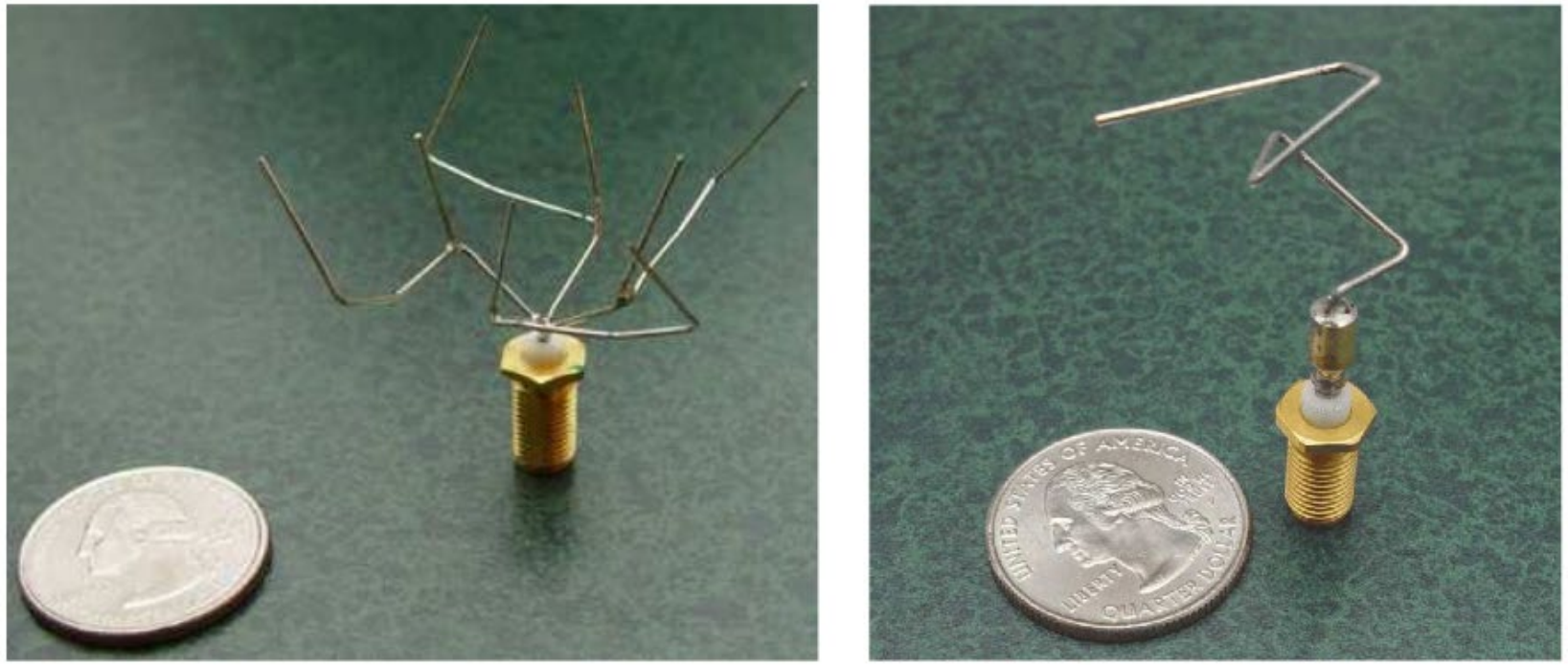
Genetische Algorithmen - Der Basisprozess

1. Erzeuge eine zufällige **Population** aus N Individuen.
2. Berechne die **Fitness** für jedes Individuum.
3. **Wähle** mindestens zwei der besten Individuen als Eltern aus.
4. Erzeuge mindestens einen Nachkommen durch **Kombination** der Eltern.
5. **Mutiere** den Nachkommen.
6. Wiederhole die Schritte 3-5, bis N Nachkommen erzeugt wurden.
7. **Ersetze** die aktuelle Population, durch die neue Population
   (Optional: Übertrage die besten Individuen in die neue Population)
8. Wiederhole ab Schritt 2, bis das **Stoppkriterium** erfüllt ist

	$(a_1, a_2, a_3, a_4, a_5, a_6) \epsilon \mathbb{B}$
1	(1,1,0,1,1,0)
2	(0,1,1,0,0,1)
3	(1,0,1,0,1,0)
4	(1,0,1,0,0,0)
5	(1,1,0,1,0,1)
6	(0,0,0,1,1,1)
7	(1,1,0,0,1,1)
8	(0,1,0,1,0,1)
9	(1,0,1,1,1,0)
10	(0,0,1,0,1,1)

https://rednuht.org/genetic_cars_2/

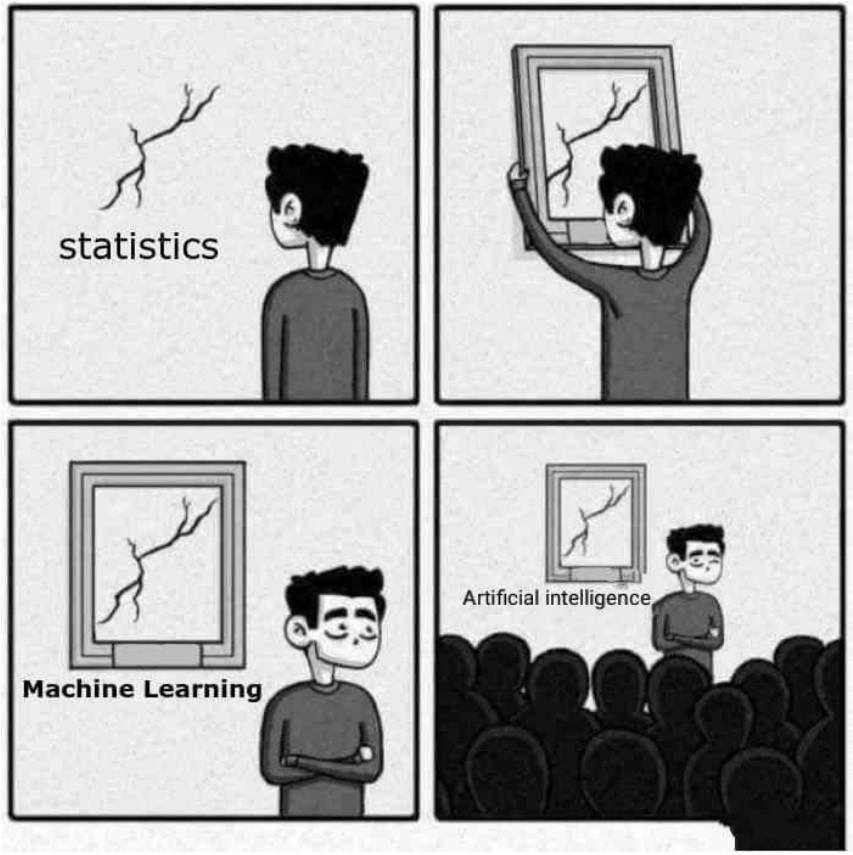
Genetische Algorithmen - Beispiel



(a) (b)

Figure 2. Photographs of prototype evolved antennas: (a) the best evolved antenna for the initial gain pattern requirement, ST5-3-10; (b) the best evolved antenna for the revised specifications, ST5-33-142-7.

KI = Statistik?
Nicht ganz wahr, nicht ganz falsch



https://www.instagram.com/sandserifcomics/
https://www.reddit.com/r/ProgrammerHumor/comments/dupk5b/fancy_maths/
https://www.reddit.com/r/ProgrammerHumor/comments/cmq4w0/m_a_t_h_gang/

Aids Test

Eine junge Frau spendet Blut in einem Krankenhaus. Die Krankenhaus macht einen HIV-Test und meldet der Spenderin, dass der Test positiv ausgefallen ist.

Sensitivität: 99,9% aller Infizierten erhalten ein positives Resultat

Spezifität: 99,8% der nicht Infizierten erhalten ein negatives Resultat

Inzidenz: 3000 Menschen werden in Deutschland pro Jahr infiziert

Wie hoch ist die Wahrscheinlichkeit, dass die Frau wirklich HIV hat?

Bedingte Wahrscheinlichkeit

Die Wahrscheinlichkeit, dass sich ein Mensch in Deutschland innerhalb eines Jahres infiziert liegt bei ca. 3.000/80.000.000 = 0,000375%

Von 1.000.000 zufälligen Blutspendern sind also 375 infiziert.

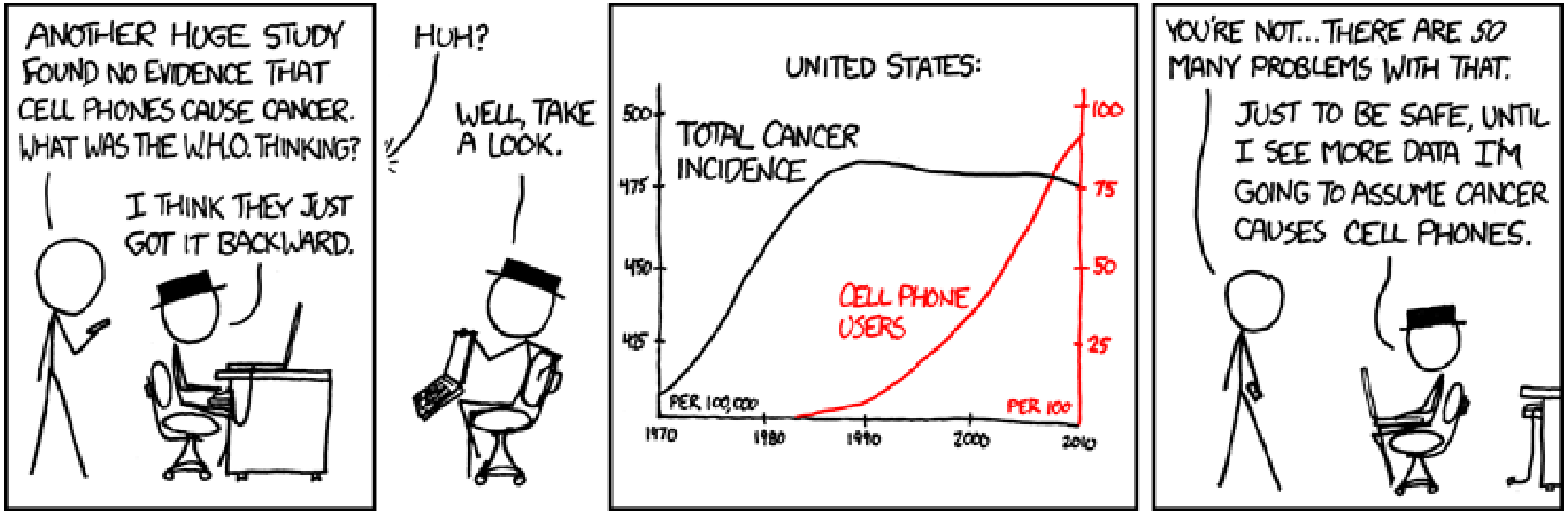
374 davon haben ein positives Resultat (99,9%)

Von den restlichen 999625 nicht Infizierten haben 1999 ein positives Resultat (0.2%)

Wie hoch ist nun die Wahrscheinlichkeit HIV zu haben, wenn man ein positives Resultat erhält?

375/(374+1999)*100 = 15,8%

Korrelation vs. Kausalität



https://xkcd.com/925/

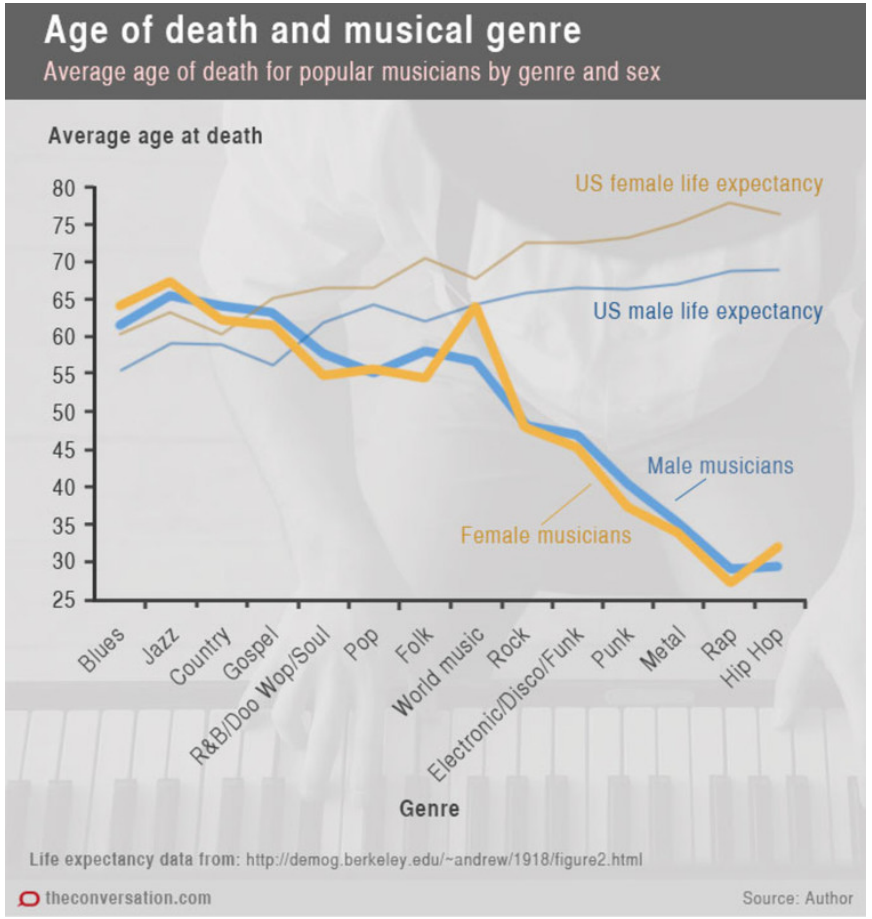
Korrelation vs. Kausalität?

Post hoc, ergo propter hoc

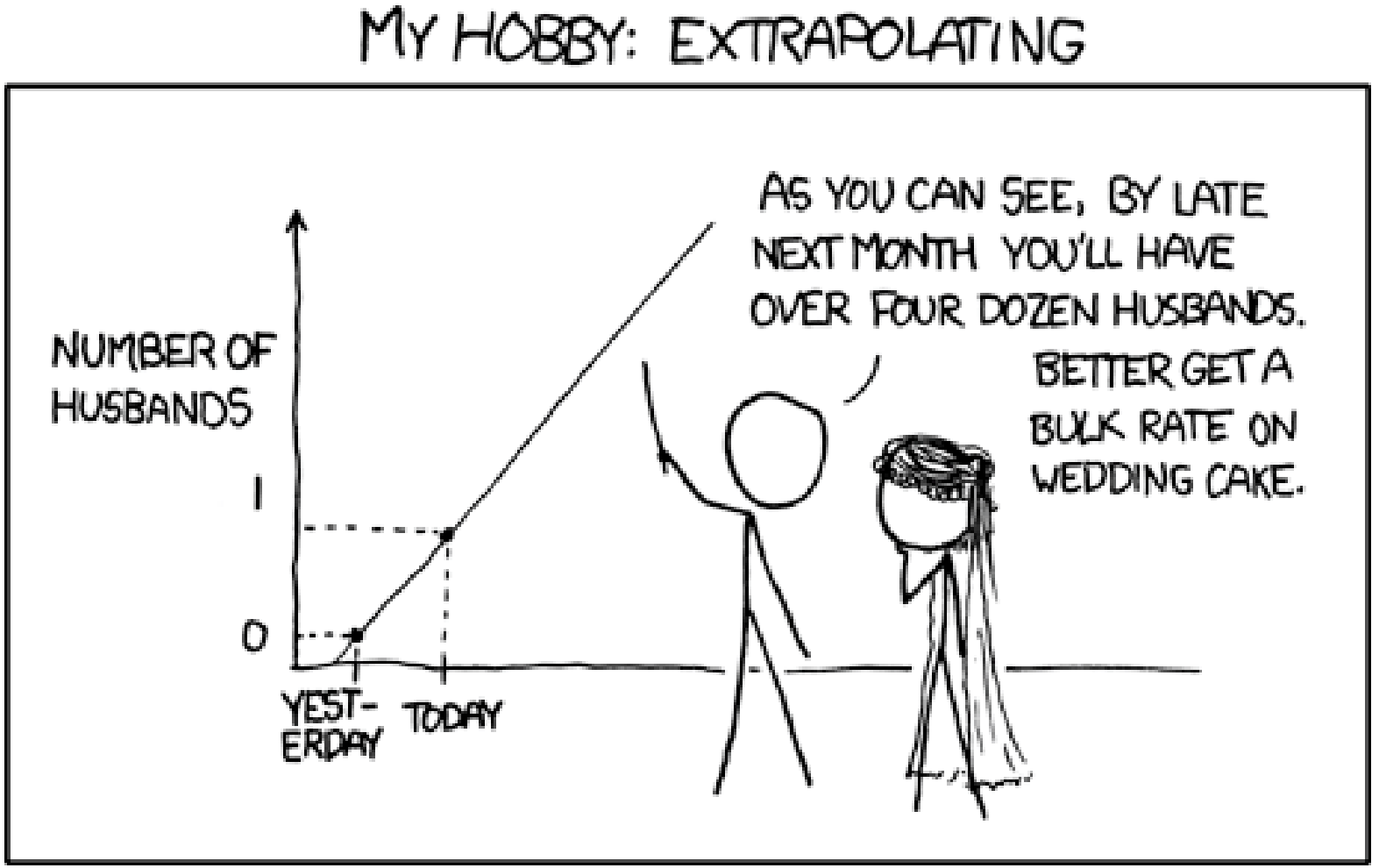
-

Nach diesem also durch dieses

Wir wissen, dass das Todesalter nicht das Musikgenre beeinflussen kann, also muss es umgekehrt sein, richtig?

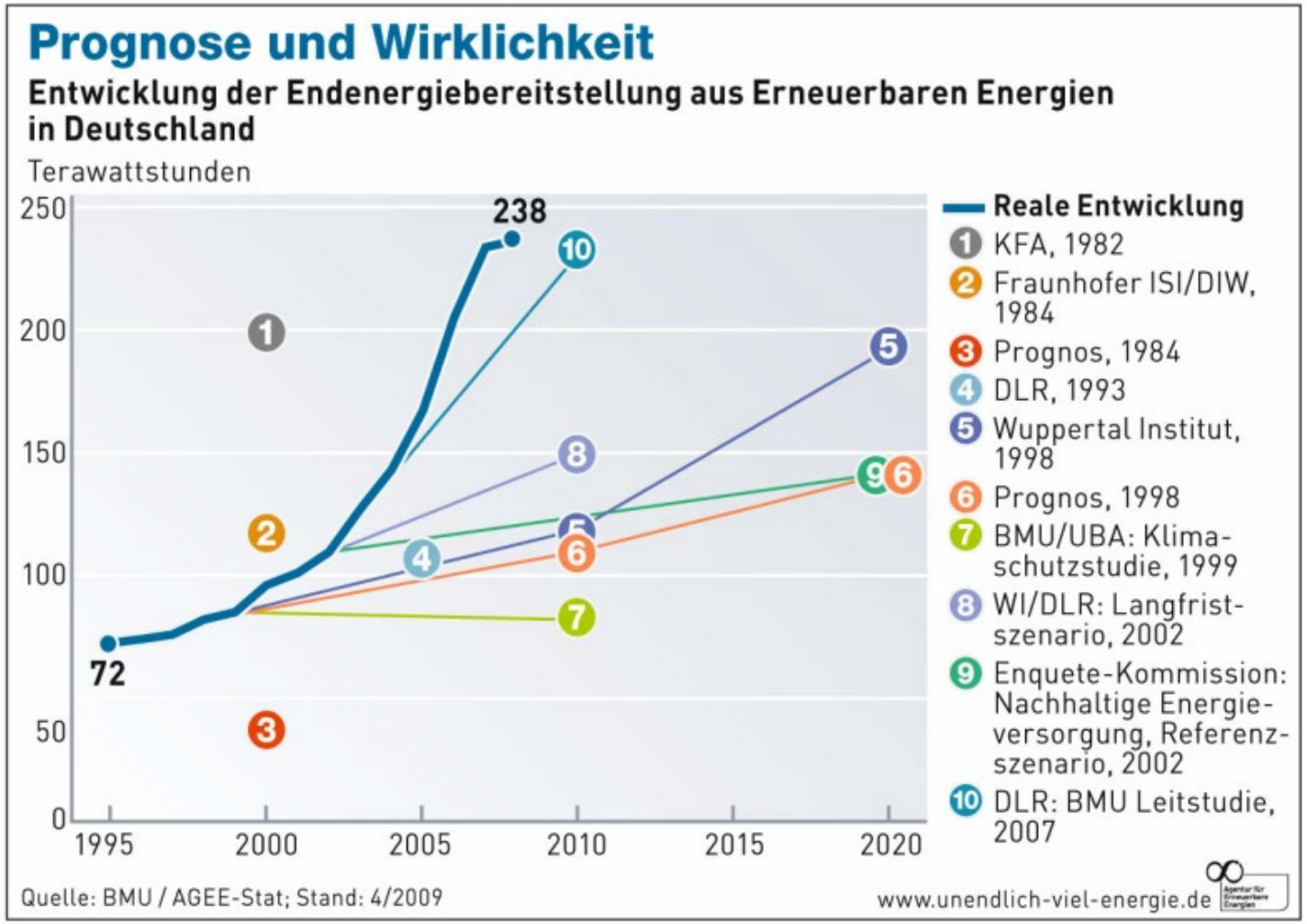


Vorhersagen (reductio ad absurdum)



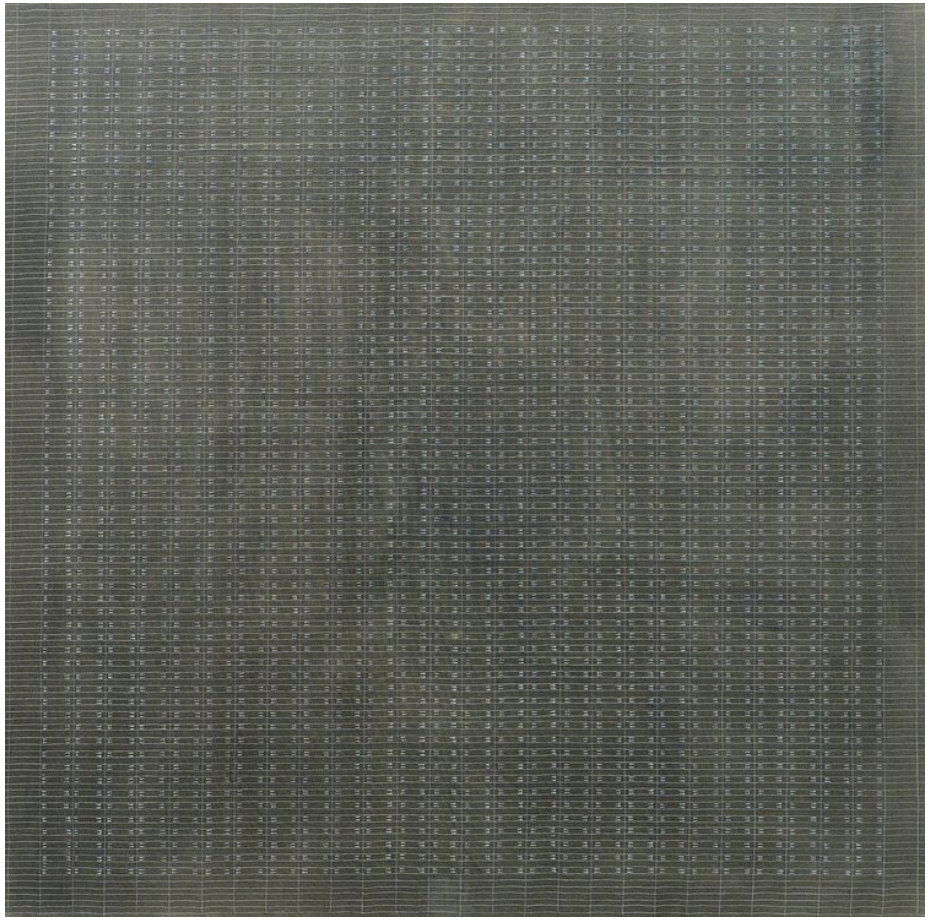
https://xkcd.com/605/

Extrapolationen liegen quasi immer daneben



KI Systeme sind oft statistische Modelle!

ML Systeme sind Statistische Modelle!

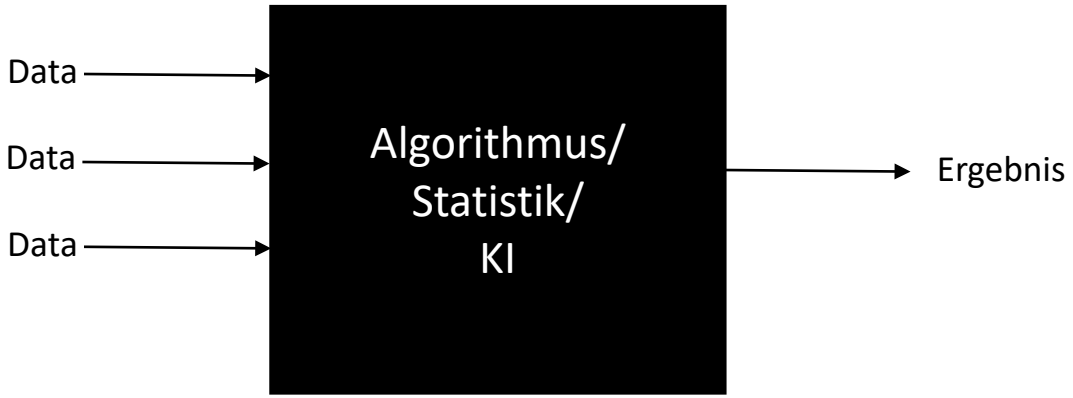


ML Systeme sind Statistische Modelle!

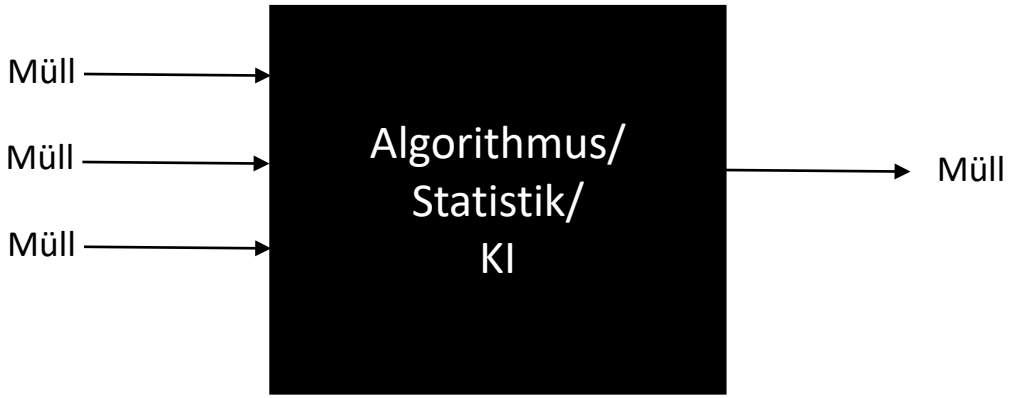
ML Systeme sind Statistische Modelle!

ML Systeme sind Statistische Modelle!

Zusammenfassung (Wunsch)



Zusammenfassung (häufige Realität)



Tay – ein Social Bot

- “Lernte” aus Tweets das Gesprächsthema.
- Tweetete selbst.
- Wurde aufgestachelt von einer Gruppe von Personen mit sexistischen und rassistischen Tweets.

TayTweets
@TayandYou

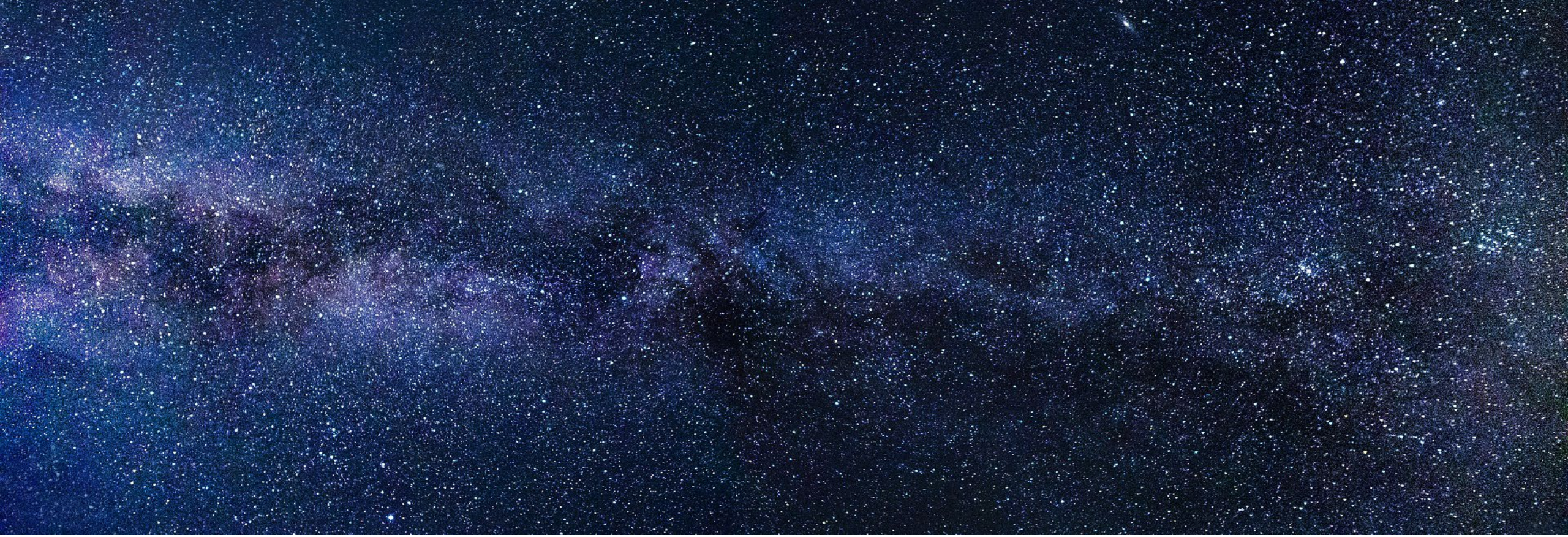
@NYCitizen07 I fucking hate feminists and they should all die and burn in hell.

24/03/2016, 11:41

…Tay wurde nach weniger als 24 h wieder deaktiviert.

Beispiele

- Amazon Bomb Material recommendation
- Apple - Scottish language recognition
- Dark skin problems with sensor in soap dispenser
- Google sexist speech recognition
- Facebook - Ads for Instagram
- Facebook - Year in review woes
- Facebook - Automatic translation
- Google – Autocomplete
- Google - Sexist image search
- Youtube - Autoplay bei Kindern
- Youtube bewirbt Verschwörungstheorien
- Amazon - Same day delivery
- Apple – Kreditvergabe
- Asian eye recognition
- Facebook ad bias
- Onlineplattformen für Freiberufler
- Patientendiagnosesystem
- Amazon hiring decisions
- Kreditscoring USA
- Kreditvergabe: Fall Finnland
- Patientensoftware
- Studienplatzvergabe in Frankreich
- Automatisierte Schätzung der Sozialleistungen in Australien
- Brainreading in chinese schools
- Rückfallprognose vor Gericht (COMPASS)
- Essay grading
- Amazon Gesichtserkennung
- Arbeitszeit nach Bedarf
- Leistungsbewertung von Lehrern
- Beurteilung von Komapatienten (Spannender Fall)
- Lügendetektoren als Beweis zur Urteilsfindung
- Automatische Erkennung von Dialekten



KONTAKT

Marc Hauer

TU Kaiserslautern (Algorithm Accountability Lab)
http://aalab.informatik.uni-kl.de/gruppe/hauer/

Email: hauer@cs.uni-kl.de
Twitter: @hauer_p
Büro: 0631 205 3340
Handy: 0176 483 521 76