

- THEMA

- Einführung: Maschinelles Lernen (ML)
- Einsatzgebiete und Beispiele
 - Wann ist ML einzusetzen?
 - Supervised, Unsupervised, Reinforcement
- Lernen aus Daten → Wie? → Ein Beispiel
 - Traditioneller Ansatz
 - ML Ansatz
 - Formalisierung & Terminologie
- Das Perzeptron - Ein einfaches Modell
- Der Perzeptron Lernalgorithmus

- WICHTIG

- $\mathcal{H} = \{\text{sgn}(w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n) : w_i \in \mathbb{R}\}$
- Δ = Perzeptron Lernalgorithmus

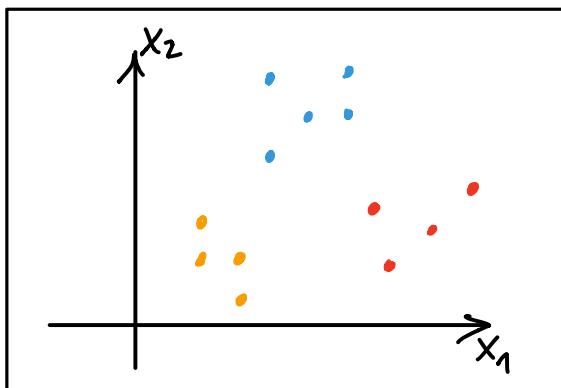
• MASCHINELLES LERNEN

- Wo wird es eingesetzt? Beispiele.
 - Produktempfehlungen. (Buch, Movie, ...)
 - Autonome Fahrzeuge, Roboter
 - SPAM-Filter
 - Bilderkennung / Objekterkennung / Face ID.
 - Bildgenerierung (DeepFake)
 - Spiele (DeepMind, Go, Atari)
 - Handschrifterkennung Spracherkennung → 
 - Medizinische Diagnosen
- Was ist es genau? Definition?
 - Fähigkeit zu lernen, ohne explizit programmiert zu werden. (Arthur Samuel, 1959)
 - ! - Wir sagen, ein Computer lernt aus Erfahrung E in Bezug auf eine Aufgabe T und einem Performancemaß P, wenn seine Performance in der Ausführung von T gemessen durch P, mit der Erfahrung E besser wird. (Tom Mitchell, 1998)
 - Kurz: Aus Erfahrung lernen und besser werden.
(in der Ausführung von T).

• Arten des Lernens

- Die üblichsten ML-Techniken können je nach der Art wie sie lernen, in drei Hauptkategorien klassifiziert werden (grob):

◦ Supervised Learning

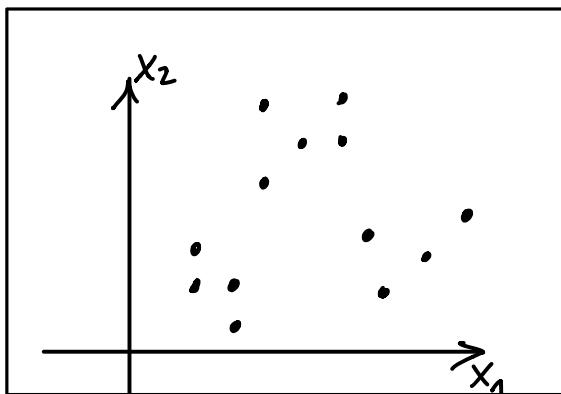


(Überwachtes Lernen)

- → Klassifizierung (Span-Fitter)
- → Regression (Hauspreis vorhersage)

- Kennzeichnet Daten (true labels)
- Ausgabe $\in \mathbb{R}$ bei Regression
- Direktes Feedback (durch true labels)

◦ Unsupervised Learning



(Unüberwachtes Lernen)

- → Clustering
- → PCA (Dimensionsreduktion)

- Keine Kennzeichnungen (no labels)
- Kein Feedback
- Erkennung von Mustern

◦ Reinforcement Learning



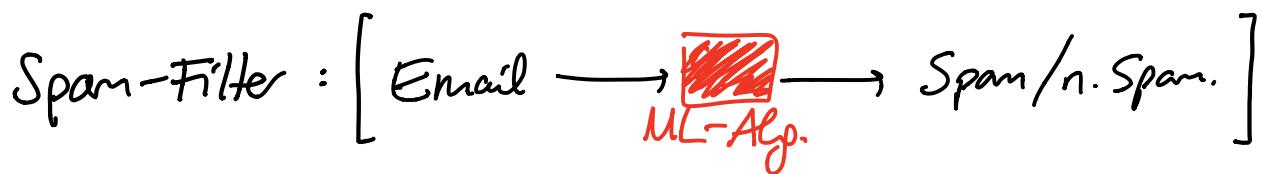
(Bestärkendes Lernen)

- → Fussball, Rubik's Cube
- → Autonome Fahrzeuge

- Lerne, gute Entscheidung zu treffen

- Bsp. AlphaGo, Deepdrive

BSP.



T = ?

Was ist die Aufgabe (Task T)

E = ?

Wie wird Erfahrung gewonnen?

P = ?

Wie wird der Erfolg gemessen?

- Welche Arten von Problemen können mittels ML gelöst werden?
→ sollten

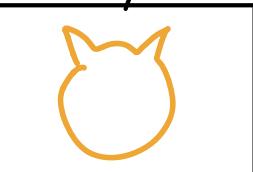
- Es gibt Daten (möglichst viel) MUSS
- Es gibt ein Muster (in den Daten)
- Muster mathematisch schwer erfassbar

Muster/Wissen aus den Daten zu extrahieren ist durch das Programmieren von expliziten Regeln nicht/schwer möglich. Keine analytische Lösung. Entwickle empirische Lösung.

BSP.

Objekterkennung

16	27	240	15
14	0	4	5
1	16	127	139
⋮⋮⋮			



Katze?

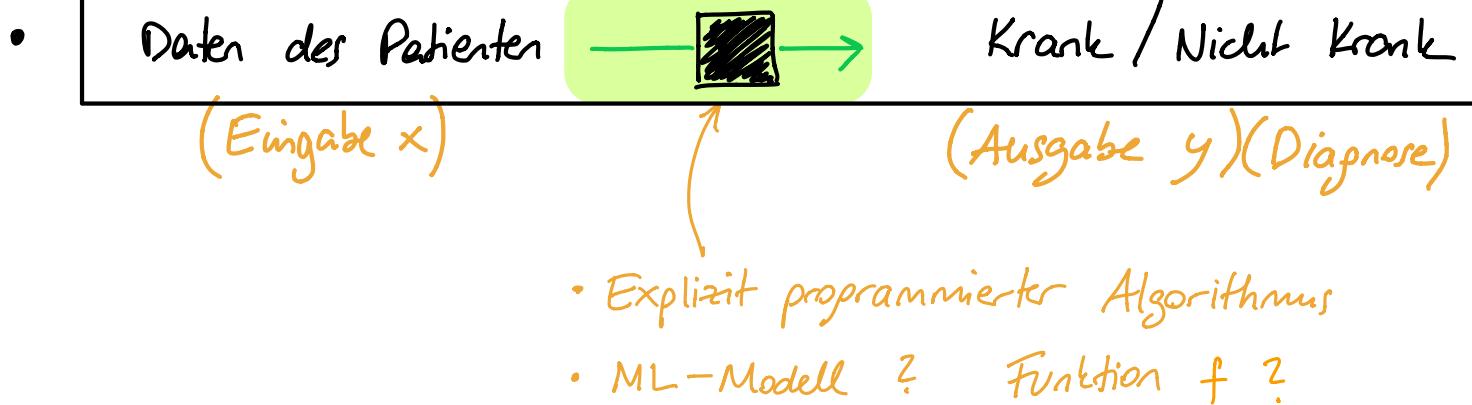
Spracherkennung



→ "Hallo"

• MEDIZINSCHE DIAGNOSE (Eine Fallstudie)

(1) (-1)



- EINGABE x : Biomedizinische Daten des Patienten als MERKMALE VEKTOR (Feature vector)

$$x = (\text{Geschlecht, Gewicht, Alter, Blutwert } 1, \text{Blutwert } 2, \dots)$$

$$= (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$$

- Tupel / Vektor der Länge n
- n : # Merkmale / Features

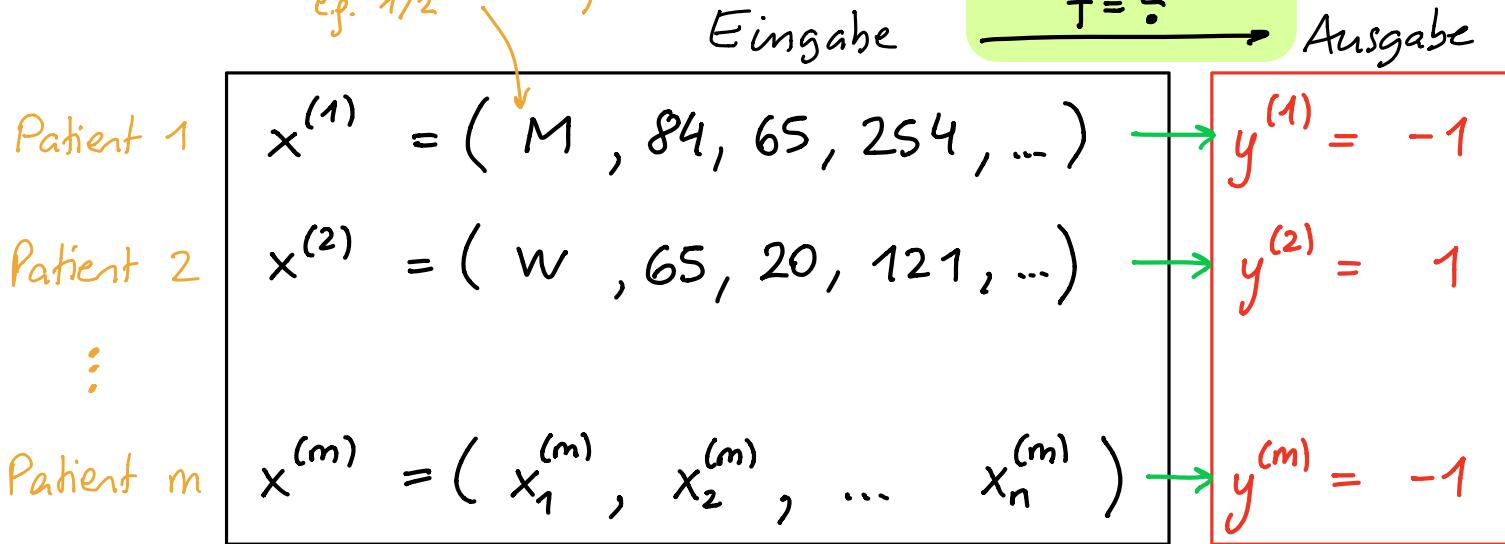
- AUSGABE y : Diagnose als 1 oder -1

- Binäre Klassifikation (zwei Klassen)

• DATEN : bisherige "Erfahrung" (Trainingsdaten)

= bekannte Eingabe - Ausgabe Paare

(müssen durch Zahlen erreicht werden)
e.g. 1/2

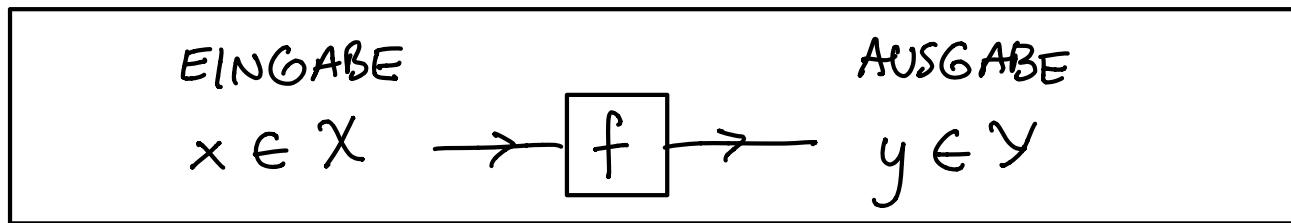


- m : # Beispiele / Datensätze
- n : # Merkmale

Labels

- "Traditionelle" Lösung ? Wie lernt der Mensch ?
- Wie könnte ein einfacher ML - Ansatz aussehen ?

• FORMALISIERUNG



- f : ZIELFUNKTION

(Ideale, unbekannte Funktion)

$$f : X \rightarrow Y$$

(bisher: explizit programmierter Algorithmus
 (eine Reihe von wohldefinierten Regeln))

- X : Input Space
 Merkmalraum

(Menge aller möglichen Eingaben)
 (engl. Feature space) (Bsp.: \mathbb{R}^n)

- Y : Output Space

(Menge aller möglichen Ausgaben)

- Bsp.: $\{-1, 1\}$
- Bsp.: {Hund, Katze}

- D : Datensatz

(Dataset. Daten bisheriger Patienten.)

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$$

- Jedes $x^{(i)}$ ist ein Vektor der Länge n !
- Index i zählt über alle Datenpunkte $[1 \leq i \leq m]$

- \mathcal{H} : Menge aller möglichen Hypothesen h (Kandidaten)
 - Bsp.: Lineare Funktionen, Polynome

- f ist die ideale Funktion und UNBEKANNT.
- $h \in \mathcal{H}$ sind mögliche Approximationen von f (Funktionen)

\mathcal{H} könnte z.B. die Menge aller Polynome zweiten Grades sein. Unter allen Kandidaten müsste nun der "beste" gefunden werden, der sich am besten an die Daten anpasst.

- h^* : Die endgültige Hypothese ($h^* \in \mathcal{H}$)
 - Der Lern - Algorithmus \mathcal{A} "wählt" ein $h^* \in \mathcal{H}$ das f möglichst gut approximiert (das "beste" h)
(bzw. eines der besten h)

ZIELFUNKTION
 $f : X \rightarrow Y$

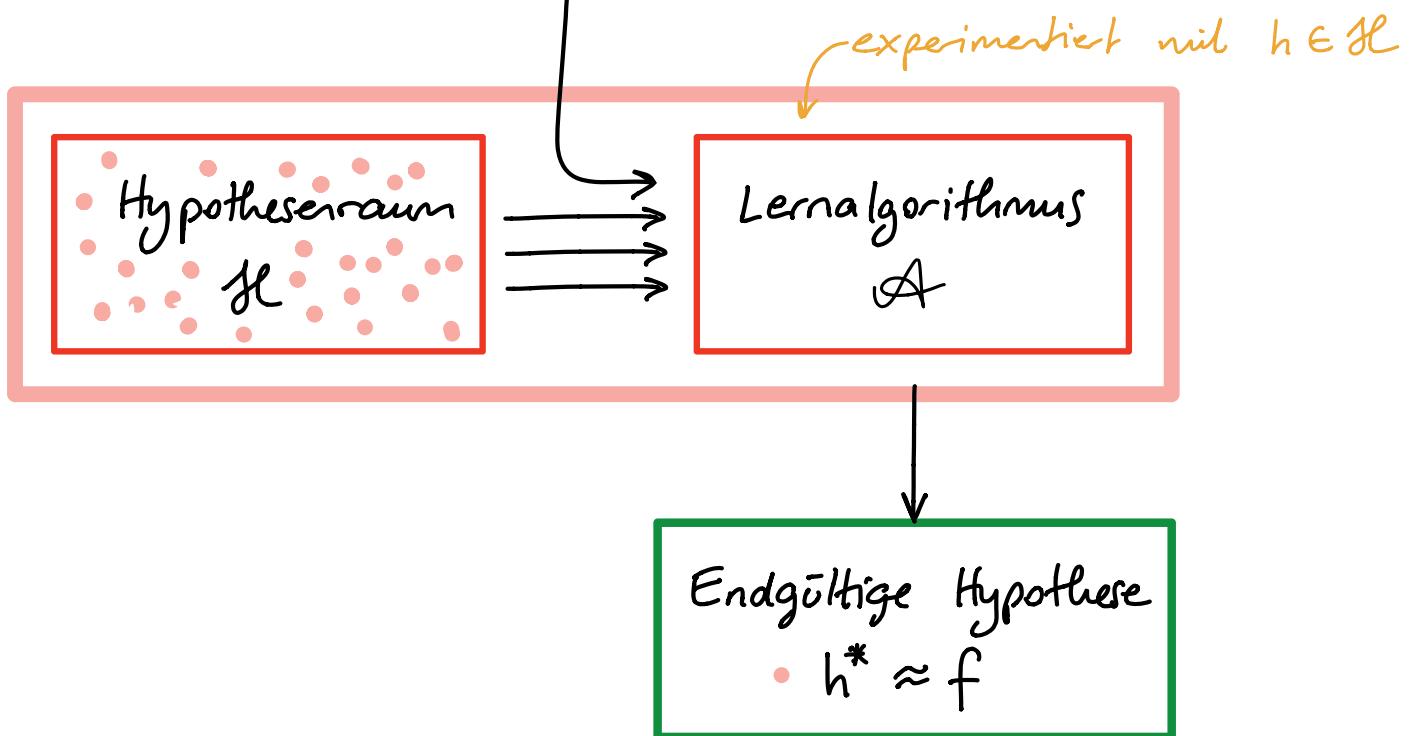
DATEN

$$\begin{aligned} & (x^{(1)}, y^{(1)}) \\ & (x^{(2)}, y^{(2)}) \\ & \vdots \\ & (x^{(m)}, y^{(m)}) \end{aligned}$$

- Beschreibt Zusammenhang zwischen Eingabe - Ausgabe

- Zielfunktion generiert Daten
- unbekannt

- (Eingabe, Ausgabe) Paare
- $x^{(i)} \in X$ (Eingaberaum)
- $y^{(i)} \in Y$ (Ausgaberaum)
- Beobachtungen. Bekannt und fix.



Quelle: Yaser Abu Mostafa, Learning from Data

- Wir als "ML Ingenieure" entscheiden:
 - Welche Art von Funktionen enthält H ? $H = ?$
 - Welche Funktionen werden ausprobiert und nach welchen Kriterium wird h^* ausgewählt? $\Delta = ?$

• EIN EINFACHES "MODELL": DAS PERCEPTRON



- Gebe den Merkmalen unterschiedliche "Gewichte", entsprechend ihrer "Relevanz" in der Entstehung der gegebenen Krankheit.
- Bilde "gewichtete Summe" der Merkmale
- Wenn Ergebnis größer als ein Schwellenwert $\Rightarrow 1$
Wenn Ergebnis kleiner als ein Schwellenwert $\Rightarrow -1$
 $\longrightarrow h = ?$
- Eingabemerkmale: $x = (x_1, x_2, x_3, x_4, \dots, x_n)$

$$h(x) = \begin{cases} 1, & \text{für } w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_nx_n \geq \theta \\ -1, & \text{für } w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_nx_n < \theta \end{cases}$$

- x bekannt!
- Gewichte w_i unbekannt! (θ unbekannt)
- $w_1x_1 + w_2x_2 + \dots + w_nx_n$ ist linear in w_i
(Lineare Gleichung) (Linearkombination der Merkmale)

$$\Rightarrow h(x) = \begin{cases} 1, & \sum_{i=1}^n w_i x_i \geq \theta \\ -1 & \sum_{i=1}^n w_i x_i < \theta \end{cases}$$

$$\Rightarrow h(x) = \begin{cases} 1, & \left(\sum_{i=1}^n w_i x_i \right) - \theta \geq 0 \\ -1, & \left(\sum_{i=1}^n w_i x_i \right) - \theta < 0 \end{cases}$$

$$\Rightarrow h(x) = \operatorname{sgn} \left[\left(\sum_{i=1}^n w_i x_i \right) + \underbrace{w_0 \cdot x_0}_{(w_0 := -\theta)} \right]$$

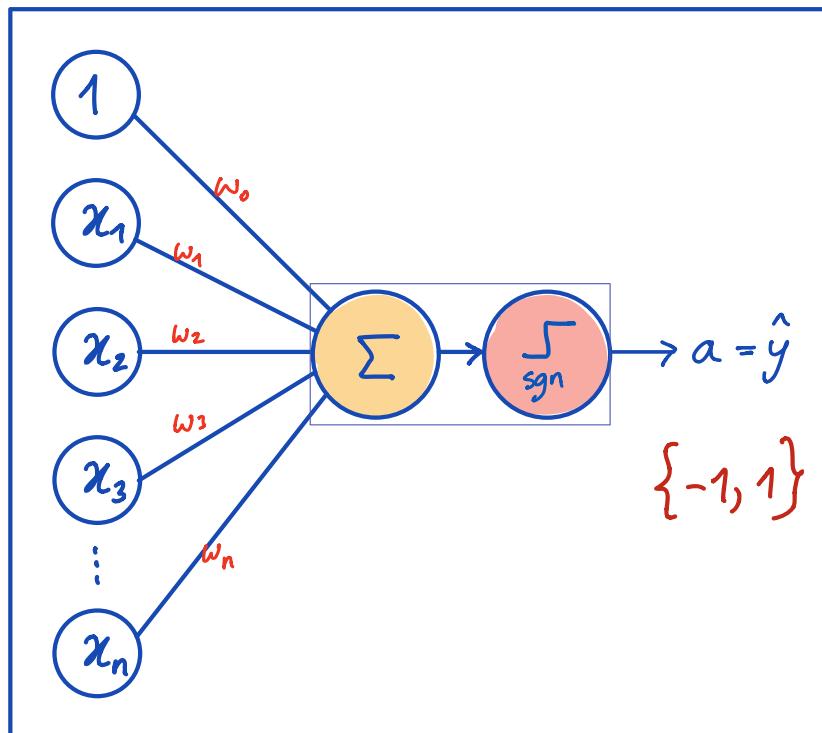
$$\Rightarrow h_w(x) = \operatorname{sgn} \left[\sum_{i=0}^n w_i x_i \right] \quad (\text{mit } x_0 := 1)$$

$$\Rightarrow h_w(x) = \operatorname{sgn} (w^\top x)$$

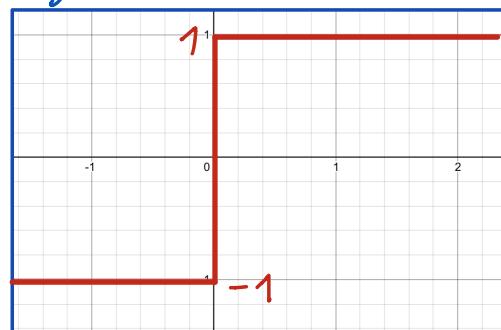
(Unbekannte Parameter von h)
mit rot hervorgehoben

da $\sum_{i=0}^n w_i x_i = w \cdot x = [w_0 \ w_1 \ \dots \ w_n] \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = w^\top x$

• Das Perzeptron als Graph (einschichtiges Neuronales Netz)



sgn - Funktion



Aktivierungsfunktion



$$a = \text{sgn} \left([\omega_0 \ \omega_1 \ \dots \ \omega_n] \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \right) = \text{sgn} (\omega^T x) = \text{sgn}(z)$$

BSP.

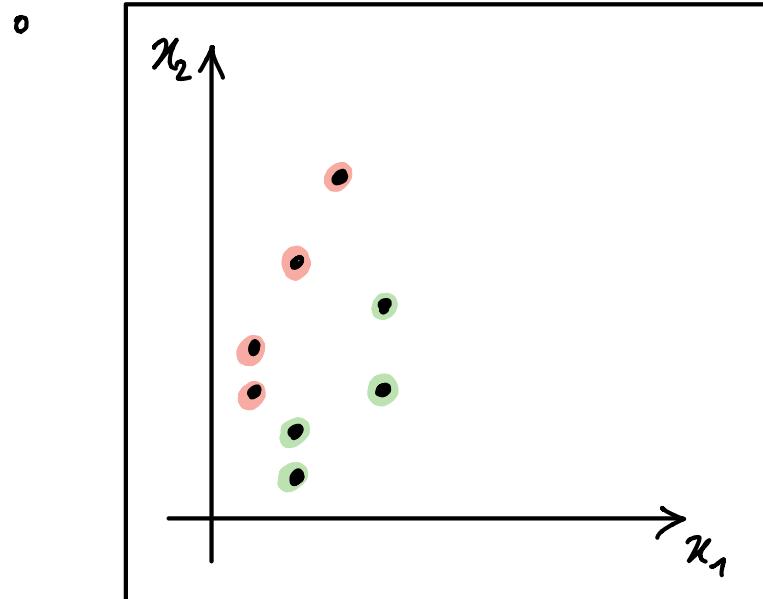
$$(x_1^{(n)}, x_2^{(n)}, y^{(n)}) \dots$$

- Daten D: $(2, 1, 1), (4, 3, 1), (4, 5, 1), (2, 3, 1)$
 $(1, 4, -1), (2, 6, -1), (1, 3, -1), (3, 8, -1)$
 $\dots (x_1^{(8)}, x_2^{(8)}, y^{(8)})$

- $m = 8$ (#Beispiele)

- $n = 2$ (#Merkmale)

- $X = \mathbb{R}^2$ (Merkmalsraum) $y = \{1, -1\}$



- Datenspunkte im Merkmalsraum, i.e. (x_1, x_2) -Ebene dargestellt.
- Klasse der Punkte mit grün / rot angegeben.

- $\mathcal{H} = \{ h_w(x) = \text{sgn}(\omega_0 + \omega_1 x_1 + \omega_2 x_2) \mid \omega_i \in \mathbb{R} \}$

Bemerkung: $2 + 2x_1 + 3x_2 = 0$ ist eine Gerade in \mathbb{R}^2
i.e. auf (x_1, x_2) -Ebene

- Einige zufällig gewählte Gewichte (ausprobieren ...)

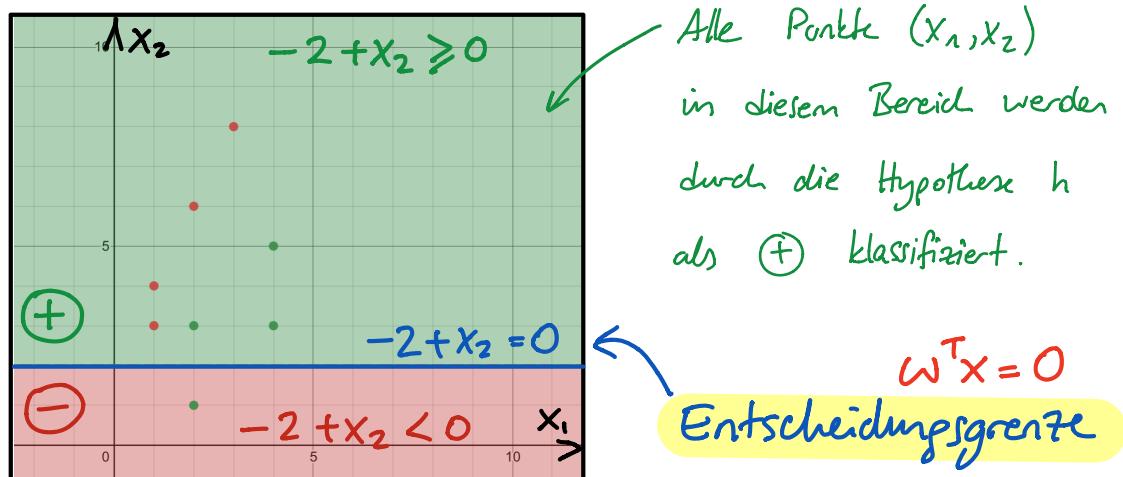
a) $w_0 = -2, w_1 = 0, w_2 = 1$

$$\omega = (-2, 0, 1) \quad x = (1, x_1, x_2)$$

$$h(x) = \text{sgn}(-2 + x_2) = \begin{cases} 1, & \underbrace{-2 + x_2}_{\omega^T x} \geq 0 \\ -1, & -2 + x_2 < 0 \end{cases}$$

⊕ ⊖

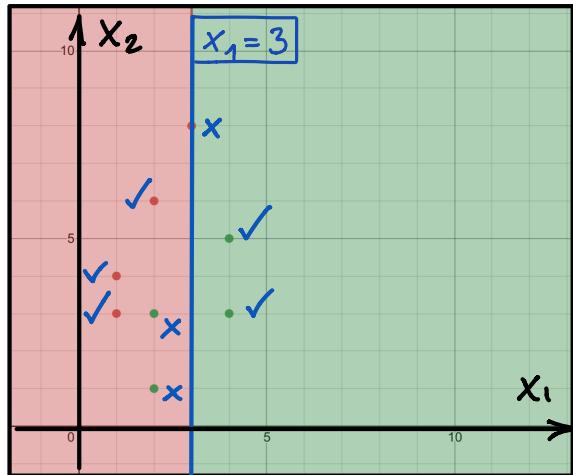
- $-2 + x_2 = 0$ ist Gerade im Merkmaltraum \mathbb{R}^2
- $-2 + x_2 \geq 0$ ist eine Seite dieser Geraden



- $h(1, 3) = -2 + 3 = 1 \geq 0 \Rightarrow \oplus \quad \text{FALSCH!} \quad \ominus$
- $h(2, 3) = -2 + 3 = 1 \geq 0 \Rightarrow \oplus \quad \text{RICHTIG!} \quad \oplus$
- $h(2, 1) = -2 + 1 = -1 < 0 \Rightarrow \ominus \quad \text{FALSCH!} \quad \oplus$
- 3/8 Punkte richtig
5/8 Punkte falsch klassifiziert.

b) $\omega_0 = -6, \omega_1 = 2, \omega_2 = 0$

$$h(x) = \begin{cases} 1, & -6 + 2x_1 \geq 0 \\ -1, & -6 + 2x_1 < 0 \end{cases}$$
⊕ ⊖



- 5/8 Punkte richtig
- 4/8 Punkte falsch

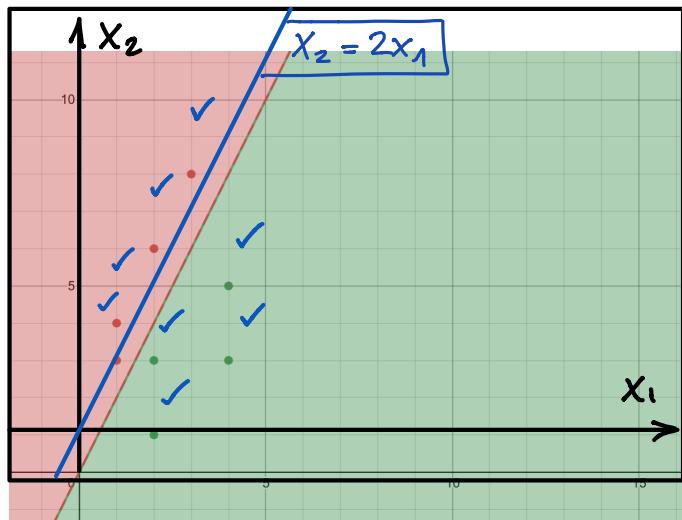
$$-6 + 2x_1 = 0 \Rightarrow x_1 = 3$$

c) $\omega_0 = 0, \omega_1 = 2, \omega_2 = -1$

$h^* = h(x)$

$$= \begin{cases} 1, & 2x_1 - x_2 \geq 0 \\ -1, & 2x_1 - x_2 < 0 \end{cases}$$
⊕ ⊖

$$\omega = \begin{bmatrix} 0 \\ 2 \\ -1 \end{bmatrix}$$



- 8/8 Punkte richtig

- $\omega^T x = 0$ ist eine Gerade
- Daten sind linear separierbar

} Das Perzeptron ist ein linearer Klassifizierer!

• Der Perceptron Lernalgorithmus (PLA)



Ein Algorithmus A, der systematisch Gewichte ausprobiert, um die "besten" Gewichte ω zu finden.

- Starte mit zufälligen ω_i
 - Solange es falsch klassifizierte Punkte gibt:
 - Wähle beliebigen Punkt $x^{(i)}$ mit:
 $h(x^{(i)}) \neq y^{(i)}$ (falsch klassifiziert)
 - Aktualisiere Gewichtsvektor mit:
- $\omega := \omega + y^{(i)} \cdot x^{(i)}$
- !
- Return $h^*(x) = \text{sign}(\omega^T x)$

Terminiert immer mit gültiger Lösung
(Beweis nicht trivial)

Ist ein Beispiel (x, y) falsch klassifiziert, liegt eines von zwei Fällen vor:

- $h(x) = 1 , y = -1$

$\omega := \omega - x$

- $h(x) = -1 , y = 1$

$\omega := \omega + x$

Gewichtsaktualisierung !

HIER FINDET DAS

LERNEN STATT

- INTUITION: Warum und wie "lert" dieser Algorithmus?
(Kein Beweis, Beispiel für $n=2$)
- Sei x ein falsch klassifizierter Punkt mit Label y :

$$h(x) \neq y \quad \text{Vorhersage} \neq \text{Label}$$

- Sei z.B.

$$y = 1$$

$$h(x) = -1$$

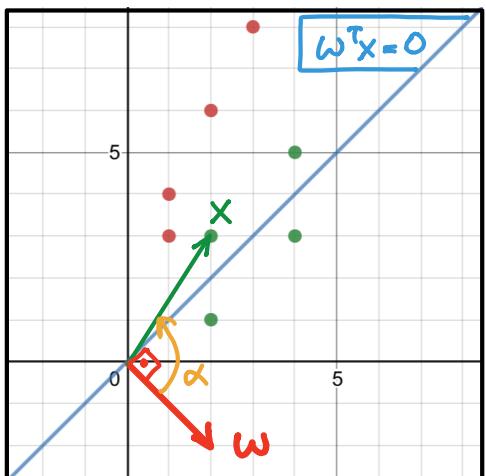
$$\Rightarrow \omega^T x < 0 \quad (\text{Skalarprodukt von } \omega \text{ und } x)$$

$$\Rightarrow \underbrace{|\omega| \cdot |x| \cdot \cos \alpha}_{\text{positiv}} < 0$$

$$\Rightarrow \cos \alpha < 0$$

$$\Rightarrow \alpha > 90^\circ \quad (\text{Winkel zwischen } \omega \text{ und } x)$$

Der Vektor ω steht senkrecht zu der Gerade $\omega^T x = 0$ und schaut in die positive Richtung (Seite mit $\omega^T x > 0$)

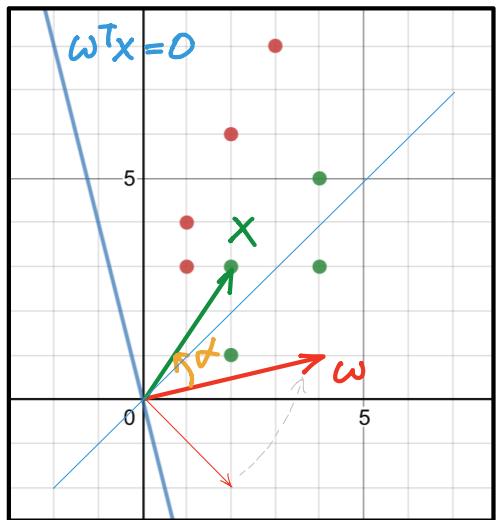


Der Punkt x liegt auf der anderen Seite der Gleichung (Seite mit $\omega^T x < 0$)

Der Winkel zwischen ω und x ist daher größer als 90° .

Beispiel für $w_1=2, w_2=-2$

- Der Vektor ω , und somit die Gerade $\omega^T x = 0$ sollen gedreht werden, sodass x danach auf der richtigen Seite landet.



- $x = (2, 3)$ war falsch klassifiziert (wahres Label: 1)
- Durch $\omega := \omega + x$ wird ω , und somit auch die Gerade gedreht, sodass x danach auf die richtige Seite fällt.

$$\omega := \omega + 1 \cdot x = (2, -2) + (2, 3) = (4, 1)$$

!

Ein Schritt des Lernalgorithmus Δ
(Gewichtsaktualisierung)

- Stumpfer Winkel ($90^\circ < \alpha < 180^\circ$) wird zu einem spitzen Winkel ($0^\circ < \alpha < 90^\circ$)
- Alle Punkte x mit dem Label $y=1$ müssen auf der gleichen Seite wie ω liegen und einen spitzen Winkel mit ω bilden.

- Drei Merkmale $x_1, x_2, x_3 \rightarrow$ Merkmalraum \mathbb{R}^3

Was ist, wenn wir x_1, x_2, x_3 haben?

- Wir haben drei Merkmale. Jeder Eingabe x ist ein Punkt im \mathbb{R}^3 :

$$(x_1, x_2, x_3) \in \mathbb{R}^3$$

- Visualisierung in 3D (GeoGebra)

$$h(x) = \text{sgn}(\omega_0 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3) = \text{sgn}(\omega^T x)$$

- Entscheidungsgrenze $\omega^T x = 0$ ist eine _____.

- Nichtlineare Entscheidungsgrenze → Polynomiale Klassifiz.

- Die Merkmale können auf nichtlineare Weise kombiniert werden:

$$x_1^2, x_1 \cdot x_2, x_2^2, x_1^2 \cdot x_2 \text{ usw.}$$

Diese neuen Werte kann man als zusätzliche, neue Merkmale betrachten.

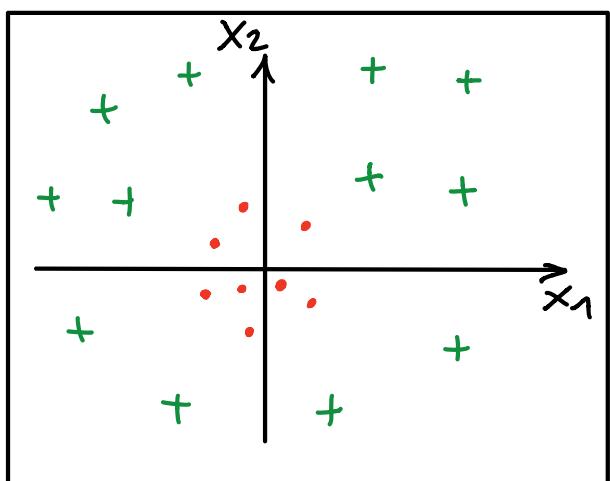
- Unter Einführung von x_1^2 und x_2^2 in unserem letzten Beispiel erhalten wir die Hypothesenfunktion:

$$h_w(x) = \operatorname{sgn} \left(w_0 + w_1 x_1 + w_2 x_2 + \underline{w_3 \cdot x_1^2} + \underline{w_4 \cdot x_2^2} \right)$$

linear in w_i !
 x_i sind Zahlen!!

zwei neue Merkmale

Bsp



- Daten nicht linear separierbar.
- Zu welcher Entscheidungsgrenze führen die Gewichte

$$(w_0, w_1, \dots, w_n) = (-4, 0, 0, 1, 1)^T$$

• Fragen

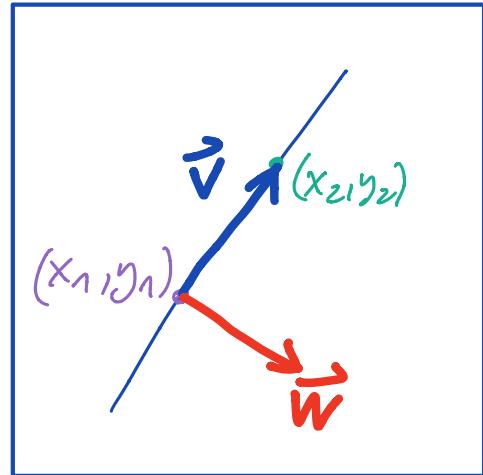
- Was macht das Perzepron, wenn die Daten nicht linear separierbar sind?
- Perzepron für das Logic Gate NOT.
 $x = ?$ $y = ?$
- Wie sieht die Entscheidungsgrenze in 3D aus?
→ Siehe GeoGebra Worksheet

Bemerkung

- Egal wo die Gerade liegt ($b=0$ oder nicht), es gilt:
 - Punkte (x_1, y_1) und (x_2, y_2) auf Gerade
$$\Rightarrow \omega_1 x_1 + \omega_2 y_1 + b = 0$$

$$\omega_1 x_2 + \omega_2 y_2 + b = 0$$
- Vektor $\vec{v} := (x_2 - x_1, y_2 - y_1)$ auf Gerade.

$$\begin{aligned}
 & \vec{v} \cdot (\omega_1, \omega_2) \\
 &= \omega_1 \cdot (x_2 - x_1) + \omega_2 (y_2 - y_1) \\
 &= (\omega_1 x_2 + \omega_2 y_2) - (\omega_1 x_1 + \omega_2 y_1) \\
 &= -b + b \\
 &= 0 \\
 &\Rightarrow \vec{v} \perp \vec{\omega}
 \end{aligned}$$



Bsp. $5x + 4y + 3 = 0 \Rightarrow \vec{\omega} = ?$