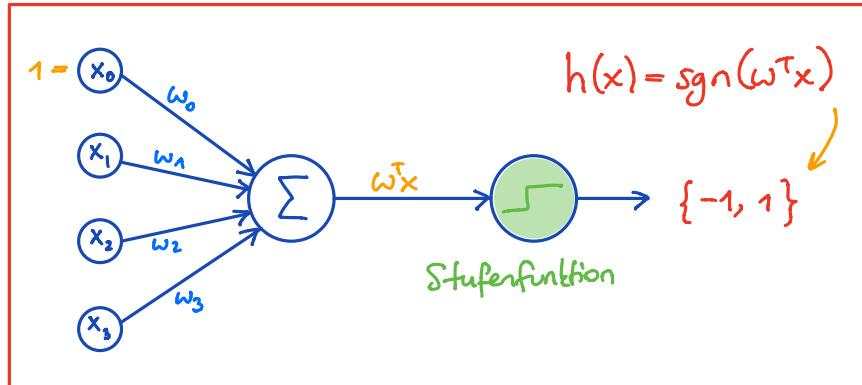
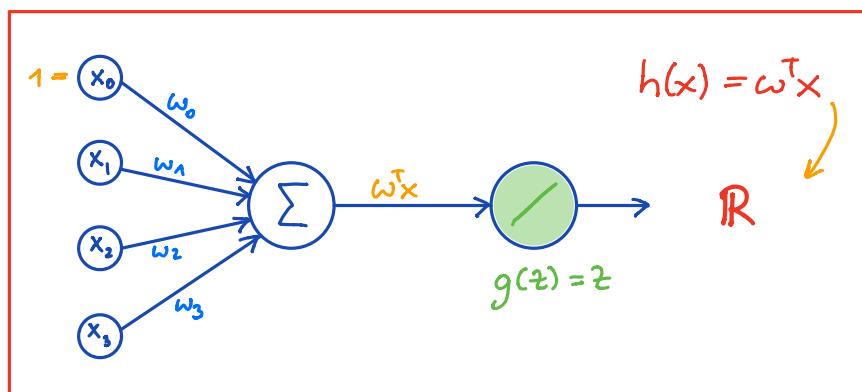


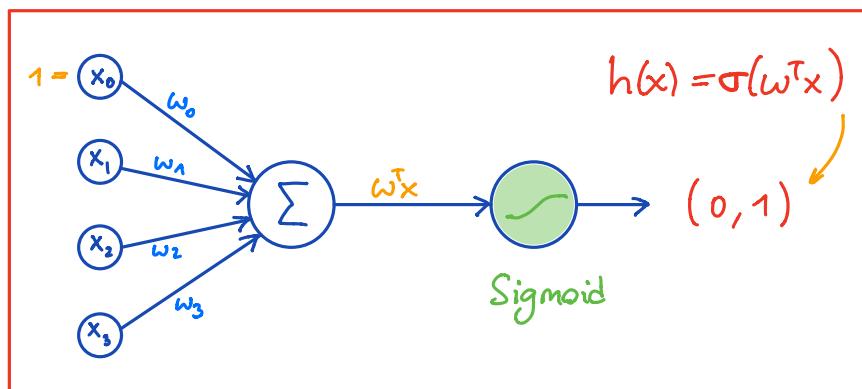
- LOGISTISCHE REGRESSION FÜR BINÄRE KLASSEIFIZIERUNG
- "GRAPH"ISCHE EINFÜHRUNG & ÜBERSICHT



- Perceptron
- Ausgabe ist -1 oder 1
(Zwei Klassen)
(Bsp.: Krank / Nicht Krank)
- $\omega^T x = 0$ Entscheidungsgrenze



- Lineare Regression
- Ausgabe ist reelle Zahl
(Bsp. Hauspreis)
- $\omega^T x$ Ausgabe

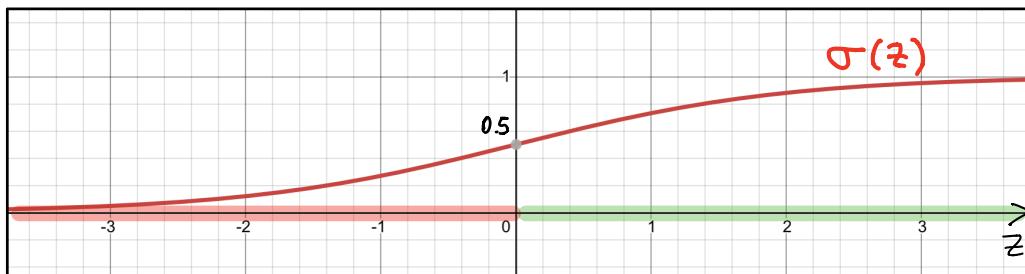


- Logistische Regression
- Ausgabe reelle Zahl in (0, 1)
(Wahrscheinlichkeit für Klasse 1)
(Bsp. 0.7 : 70% Klasse 1)
- $\omega^T x = 0$ Entscheidungsgrenze

Eingabe \vec{x}

Ausgabe y

- DIE LOGISTISCHE FUNKTION (als Aktivierungsfunktion)



$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Logistic Function
(Sigmoid Function)

$$\begin{array}{c|c} z < 0 & z > 0 \\ \sigma(z) < 0.5 & \sigma(z) > 0.5 \end{array}$$

$$\sigma(0) = 0.5$$

- Vergleiche mit der Vorzeichenfunktion

(siehe auch : Heaviside step function)



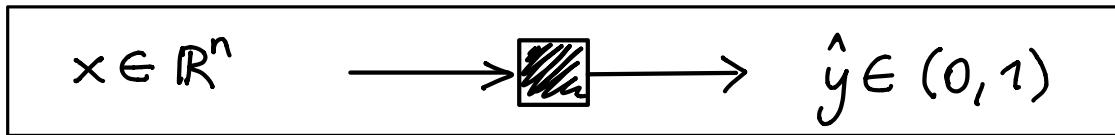
$$\text{sgn}(z) = \begin{cases} 1 & z \geq 0 \\ -1 & z < 0 \end{cases}$$

$$\begin{array}{c|c} z < 0 & z > 0 \\ \text{sgn}(z) = -1 & \text{sgn}(z) = 1 \end{array}$$

- Ableitung ?

• LOGISTISCHE REGRESSION - FORMAL ZUSAMMENGEFASST

- ## ◦ Das Problem : Binäre Klassifizierung



(Merkmalsvektor) Reelle Zahl in offenem Intervall $(0, 1)$

- ## ◦ D : m Input-Output Paare

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

$$y^{(i)} \in \{0, 1\}$$

- ## • H : Die Hypothese

$$h_{\omega}(x) = \sigma(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + \dots + w_n x_n) = \sigma(\omega^T x)$$

- \mathcal{L} : Cross-Entropy Loss (Verlustfunktion)

$$\mathcal{L}(\hat{y}, y) = -y \cdot \log(\hat{y}) - (1-y) \cdot \log(1-\hat{y})$$

"Distanz" zwischen \hat{y} und y

- J : Cross-Entropy Cost (Kostenfunktion)

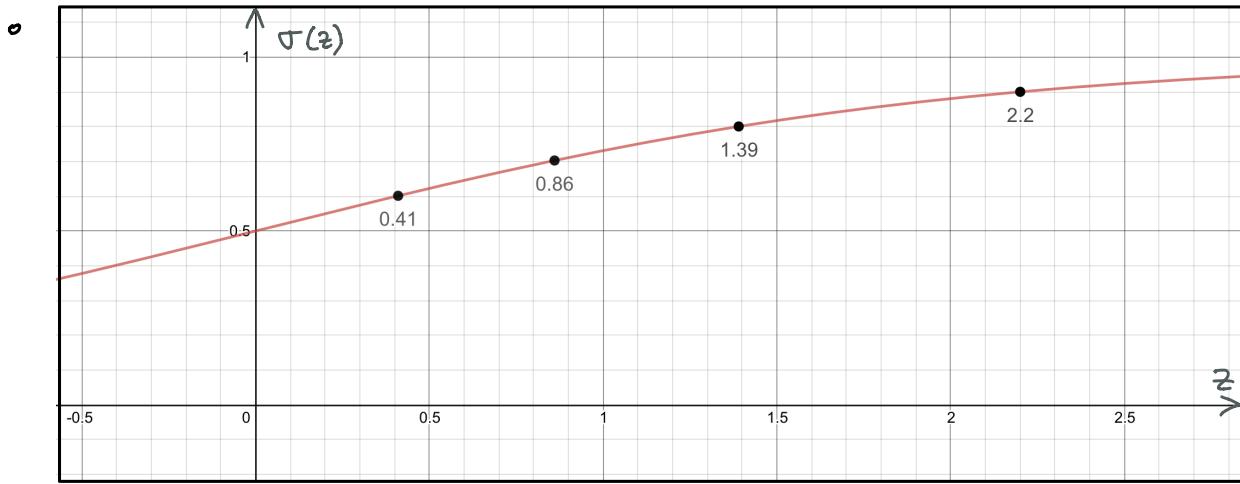
$$J(w) = -\frac{1}{m} \sum_{i=1}^m \left[y^{(i)} \cdot \log(h(x^{(i)})) + (1-y^{(i)}) \cdot \log(1-h(x^{(i)})) \right]$$

- A : Gradient Descent

• \mathcal{H} - DIE HYPOTHESENFUNKTION

- $h_{\omega}(x) = \sigma(\omega_0 + \omega_1 x_1 + \omega_2 x_2) = \sigma(\omega^T x) \in (0,1)$
- Interpretiere $h(x)$ als die Wahrscheinlichkeit, dass " $y=1$ " ist
- $h(x) = \sigma(\omega^T x) = 0.6 \Rightarrow \begin{cases} P(y=1 | x; \omega) = 60\% \\ P(y=0 | x, \omega) = 40\% \end{cases} \quad \text{Vorhersage : 1}$
- $h(x) = \sigma(\omega^T x) = 0.3 \Rightarrow \begin{cases} P(y=1 | x; \omega) = 30\% \\ P(y=0 | x, \omega) = 70\% \end{cases} \quad \text{Vorhersage : 0}$
- $h(x) = \sigma(\omega^T x) = 0.5 \quad \text{für} \quad \omega^T x = 0 \quad (\text{unentschlossen})$

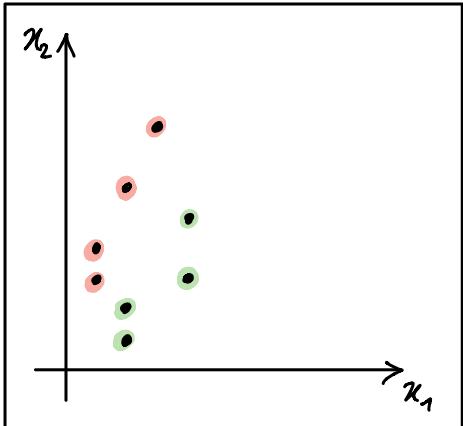
$\omega^T x = 0$ ist die Entscheidungsgrenze



<https://www.desmos.com/calculator/u2pictahys>

- $\sigma(0) = 0.5$
- $\sigma(0.41) \approx 0.6$
- $\sigma(0.86) \approx 0.7$
- $\sigma(1.39) \approx 0.8$
- $\sigma(2.2) \approx 0.9$

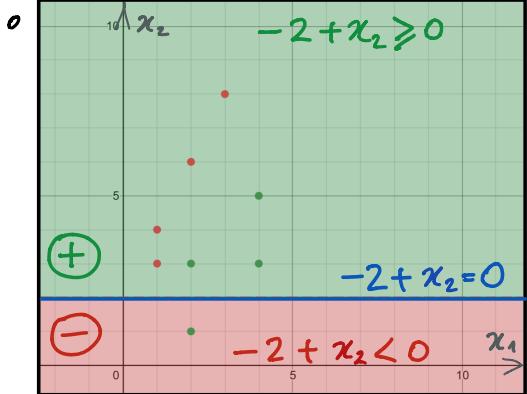
BSP. (Perzeptron Fallstudie)



- D: $(2, 1, 1), (4, 3, 1), (4, 5, 1), (2, 3, 1)$
 $(1, 4, -1), (2, 6, -1), (1, 3, -1), (3, 8, -1)$
 - $m = 8$
 - $n = 2$
 - $X = \mathbb{R}^2$
 - $y = \{0, 1\}$

$$a) \quad \omega = (-2, 0, 1) \quad x = (1, x_1, x_2) \quad \Rightarrow \quad \omega^T x = -2 + x_2$$

- $h(x) = \sigma(-2 + x_2)$
 - Entscheidungsgrenze : $-2 + x_2 = 0 \Rightarrow x_2 = 2$



$$h(1,3) = \sigma(1) = 0.73$$

$$h(2,1) = \sigma(-1) = 0.27$$

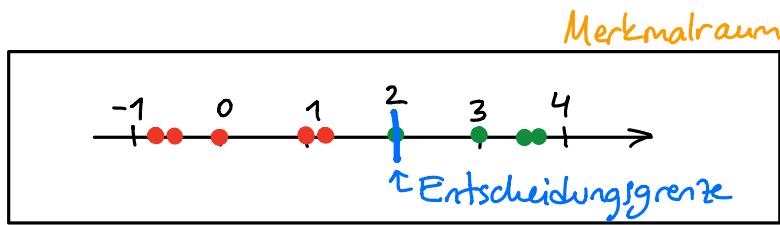
$$h(2,6) = \sigma(4) = 0.98$$

Entscheidungsgrenze

BSP. $h(x) = \sigma(\omega_0 + \omega_1 x_1)$, $\omega = (-4, 2)$. (n=1)

- $D = \{(2, 1), (3, 1), (3.5, 1), (3.6, 1), (1.2, 0), (1, 0), (0, 0), (-0.5, 0), (-0.7, 0)\}$

- $h(x) = \sigma(-4 + 2x_1)$
- Entscheidungsgrenze: $-4 + 2x_1 = 0 \Rightarrow x_1 = 2$



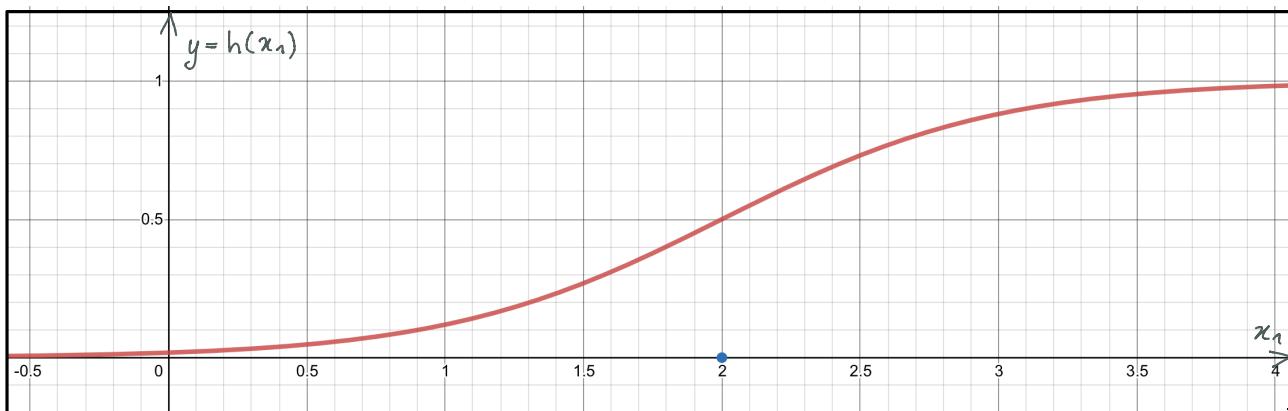
- Klasse 1:

$$-4 + x_1 \geq 0$$

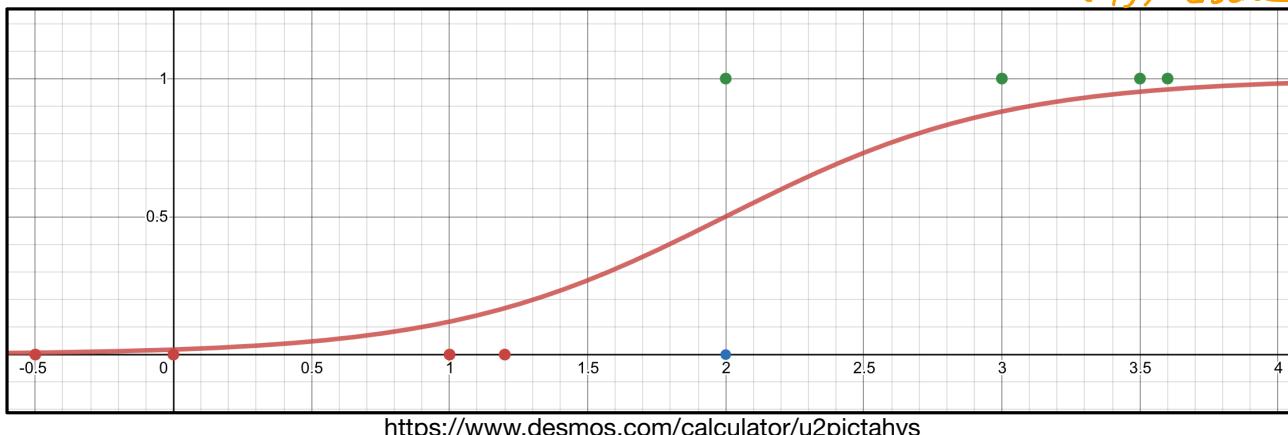
$$\Rightarrow x_1 \geq 4$$

- Hypothesenfunktion – Graphische Darstellung in (x, y) -Ebene

$$h(x_1) = \frac{1}{1 + e^{-4+2x_1}}$$

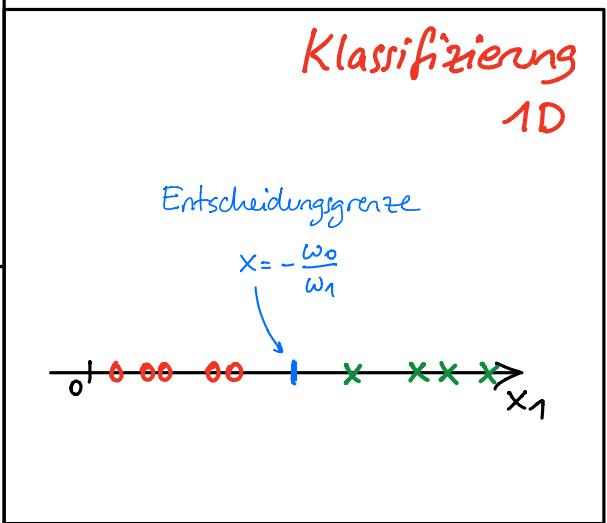
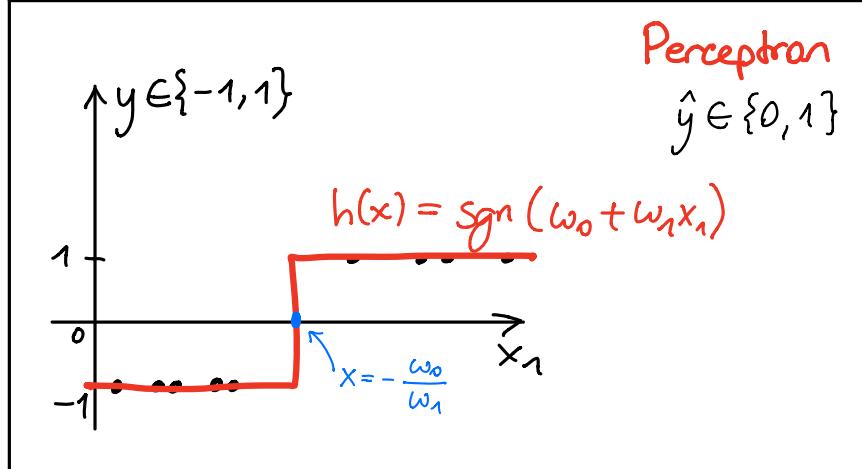
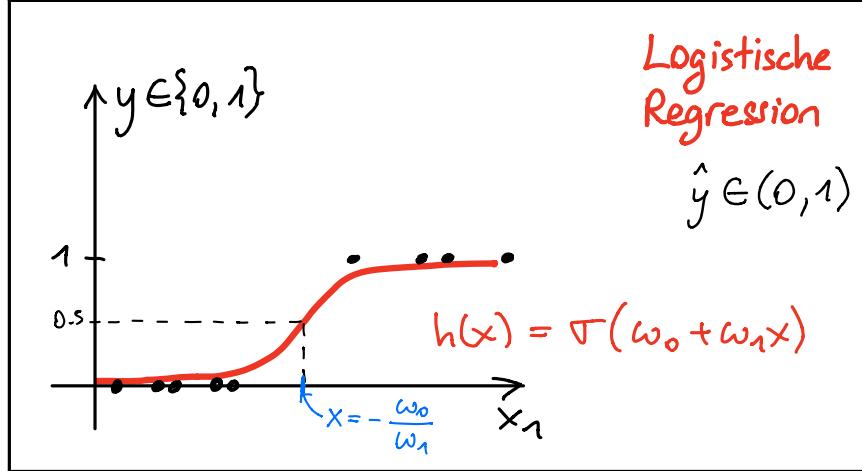
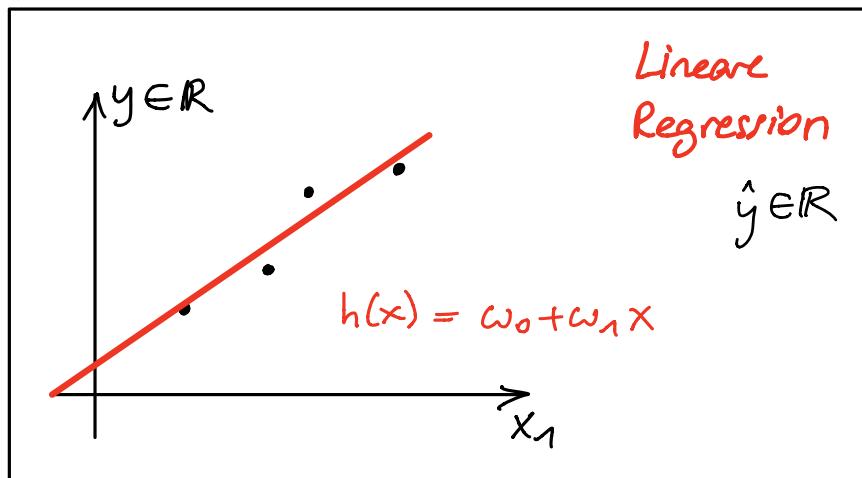


(x, y)-Ebene



• "Curve Fitting" und Klassifizierung

(n=1)



Merkmalraum
(Feature Space)

(x, y) - Ebene

• L & J - VERLUST & KOSTEN

- Der Verlust (Loss) ist der "Fehler" der Hypothesenfunktion für ein individuelles Beispiel (x, y)
- Zum Beispiel war L für die Lineare Regression als das Fehlerquadrat definiert:

$$L = \text{loss}(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2 = \frac{1}{2} (h(x) - y)^2$$

Anderer Loss Funktionen sind möglich, z.B. $|\hat{y} - y|$

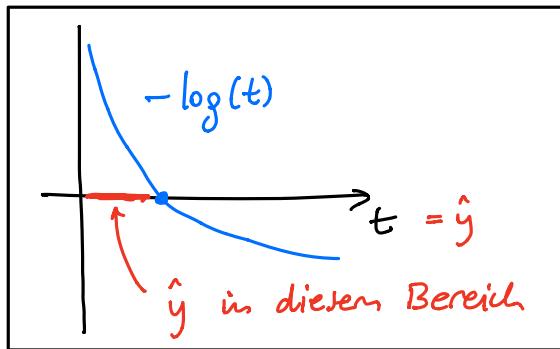
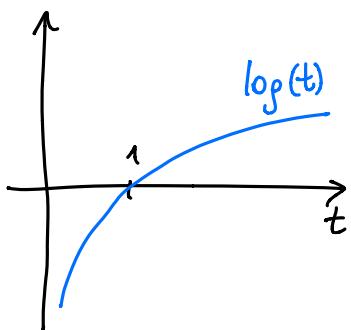
- Für die Logistische Regression wird die Cross-Entropy Verlustfunktion verwendet :

Cross Entropy Loss (Negative Log Likelihood)

$$L(h(x), y) = -y \log(h(x)) - (1-y) \cdot \log(1-h(x))$$

- Diese Funktion kann auch wie folgt abschnittsweise definiert werden :
 - Wenn $y=1 \Rightarrow \text{loss} = -\log(\hat{y})$
 - Wenn $y=0 \Rightarrow \text{loss} = -\log(1-\hat{y})$

- Wie misst diese Funktion den Verlust? (Intuition)



- $y = 1$ und $\hat{y} \approx 1 \Rightarrow -\log(\hat{y}) \approx 0$
 $\hat{y} \approx 0 \Rightarrow -\log(\hat{y}) \rightarrow \infty$
- $y = 0$ und $\hat{y} \approx 1 \Rightarrow -\log(1-\hat{y}) \rightarrow \infty$
 $\hat{y} \approx 0 \Rightarrow -\log(1-\hat{y}) \approx 0$

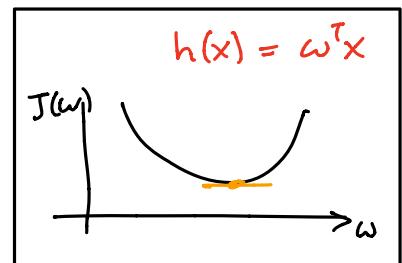
Cross Entropy Cost (Kostenfunktion)

$$J(\omega) = \frac{1}{m} \cdot \sum_{i=1}^m \left[-y^{(i)} \cdot \underbrace{\log(h(x^{(i)}))}_{\text{orange bracket}} - (1-y^{(i)}) \cdot \log(1-h(x^{(i)})) \right]$$

$$\log \left(\frac{1}{1 + e^{-(\omega_0 + \omega_1 x_1 + \dots + \omega_n x_n)}} \right)$$

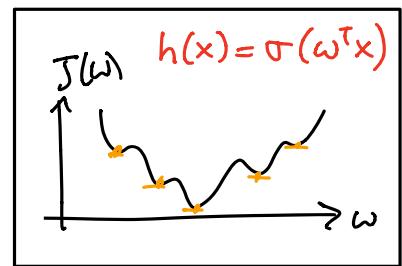
- Warum nicht die Fehlerquadrate verwenden?

$$\text{Linear: } (\hat{y} - y)^2 = (\omega^T x - y)^2 = \\ = (\omega_0 + \omega_1 x_1 - y)^2$$



$J(w)$ konvex

$$\text{Logistisch: } (\hat{y} - y)^2 = (\sigma(\omega^T x) - y)^2 \\ = \left[\frac{1}{1+e^{-\omega_0-\omega_1 x}} - y \right]^2$$

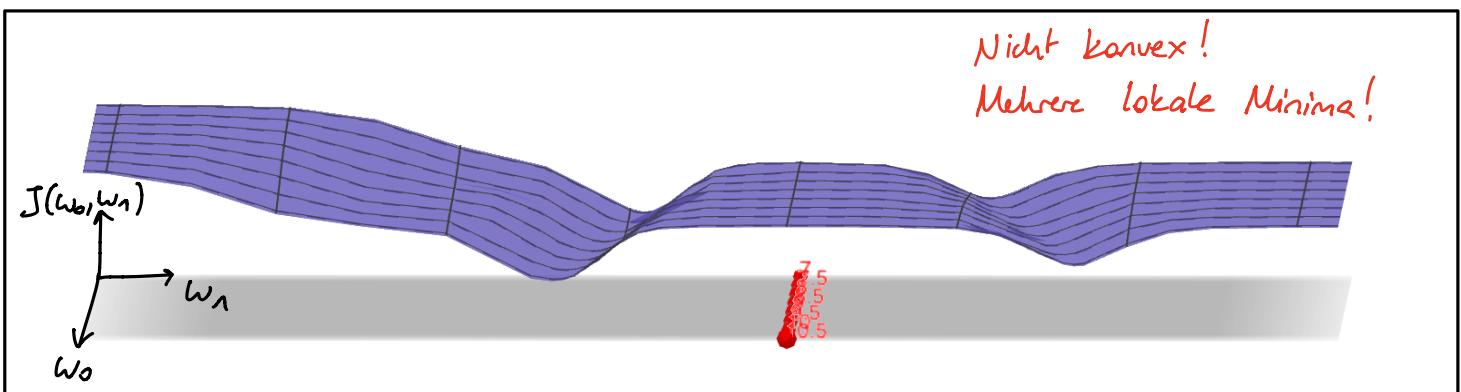


$J(w)$ nicht konvex!!

(Mehrere lokale Minima möglich!)

BSP.

$$J(\omega_0, \omega_1) = \frac{1}{5} \left[\left(\frac{1}{1+e^{-\omega_0-18\omega_1}} \right)^2 + \left(\frac{1}{1+e^{-\omega_0-7\omega_1}} - 1 \right)^2 + \left(\frac{1}{1+e^{-\omega_0+12\omega_1}} - 1 \right)^2 \right. \\ \left. + \left(\frac{1}{1+e^{-\omega_0-13\omega_1}} - 1 \right)^2 + \left(\frac{1}{1+e^{-\omega_0+13\omega_1}} \right)^2 \right]$$



<https://www.geogebra.org/classic/bt3grg8r>

• A - GRADIENTENABSTIEG (GRADIENT DESCENT)

• Gewichtsaktualisierungen

$$\omega := \omega_{\text{alt}} - \alpha \cdot \nabla J$$

$$\omega_j := \omega_j - \alpha \underbrace{\frac{\partial J(\omega)}{\partial \omega_j}}_{?} \quad (j=1, \dots, n)$$

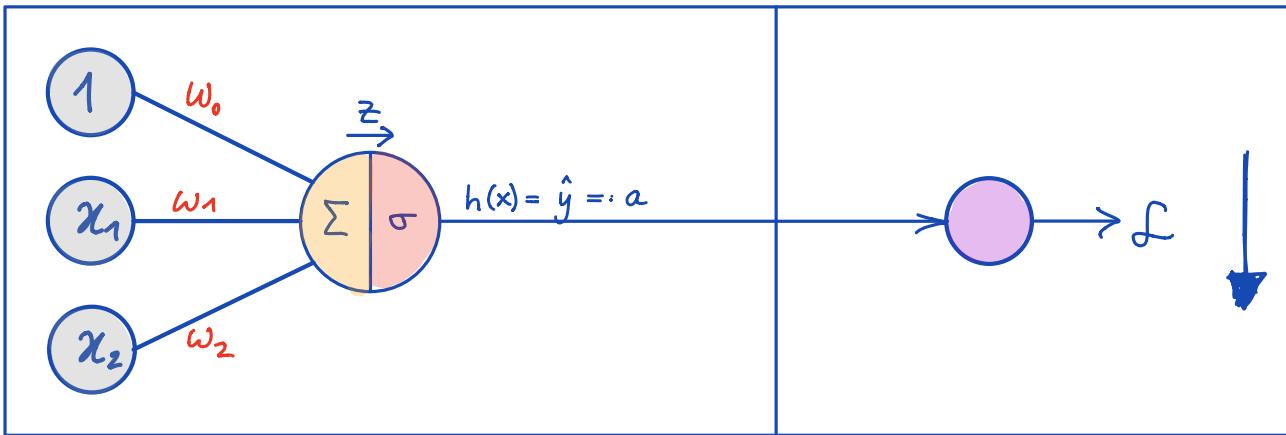
\Rightarrow Was ist $\frac{\partial J}{\partial \omega_j}$ für

$$J(\omega) = \frac{1}{m} \cdot \sum_{i=1}^m \left[-y^{(i)} \cdot \underbrace{\log(h(x^{(i)}))}_{?} - (1-y^{(i)}) \cdot \log(1-h(x^{(i)})) \right]$$

$$\log \left(\frac{1}{1 + e^{-(\omega_0 + \omega_1 x_1 + \dots + \omega_n x_n)}} \right)$$

• Die partiellen Ableitungen

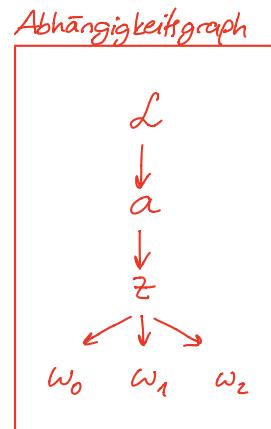
Ein Beispiel (x, y) mit $x = (x_1, x_2)$ ($m=1, n=2$)



- $J = L = - \left(y \cdot \log(a) + (1-y) \cdot \log(1-a) \right)$

$$a = \sigma(z)$$

$$z = \omega^T x = w_0 + w_1 x_1 + w_2 x_2$$



- Kettenregel :

$$\frac{\partial L}{\partial w_0} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_0} = [(a-y) \cdot 1]$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_1} = [(a-y) \cdot x_1] = \nabla L = \nabla J$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_2} = [(a-y) \cdot x_2]$$

wenn $m=1$

Vergleiche mit linearer Regression!

$$(\ln(x))' = \frac{1}{x}$$

- $L(a, y) = -y \cdot \log(a) - (1-y) \cdot \log(1-a)$

$$\begin{aligned} \frac{\partial L}{\partial a} &= \frac{-y}{a} - \frac{(1-y)}{(1-a)} \cdot (-1) = \frac{-y(1-a) + a(1-y)}{a(1-a)} = \\ &= \frac{-y + ya + a - ay}{a(1-a)} = \frac{a - y}{a(1-a)} \end{aligned}$$

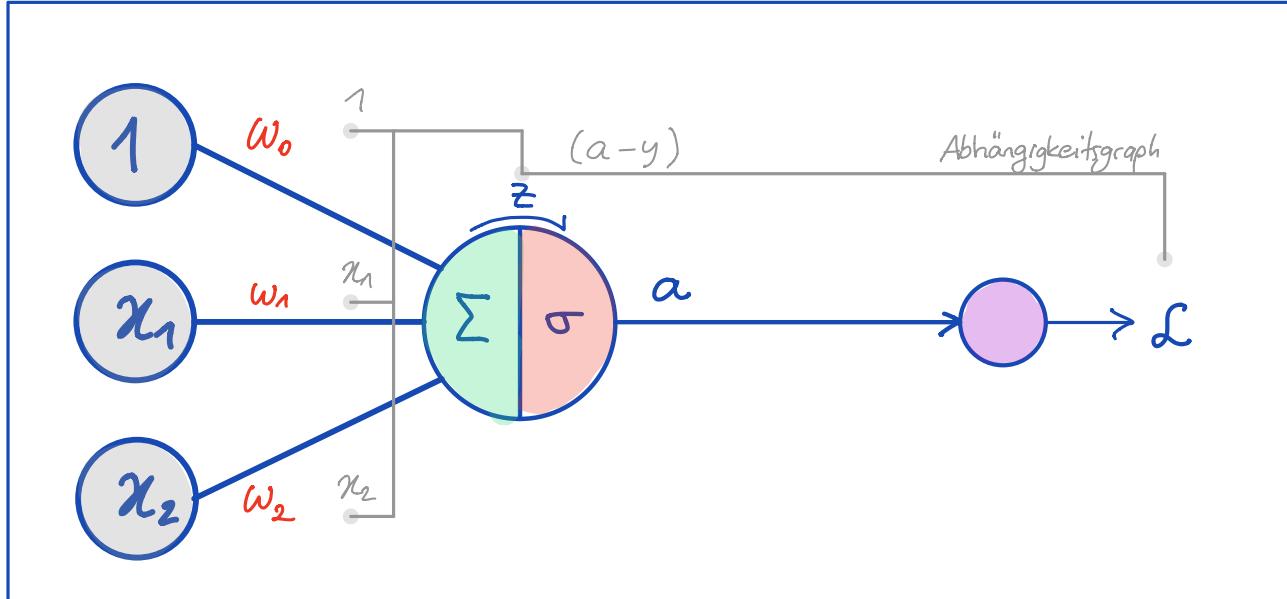
- $a = \sigma(z) = \frac{1}{1+e^{-z}} \Rightarrow \frac{1}{a} = 1+e^{-z}$

$$\begin{aligned} \frac{da}{dz} &= \frac{d}{dz} (1+e^{-z})^{-1} = -1 \cdot (1+e^{-z})^{-2} \cdot (-e^{-z}) \\ &= -1 \cdot a^2 \cdot \left(1 - \frac{1}{a}\right) \\ &= -a^2 \cdot \left(\frac{a-1}{a}\right) = a(1-a) \end{aligned}$$

- $\frac{\partial L}{\partial z} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial z} = \frac{a-y}{a(1-a)} \cdot a(1-a) = a-y$

Dies gilt, solange
 $a = \sigma(z)$

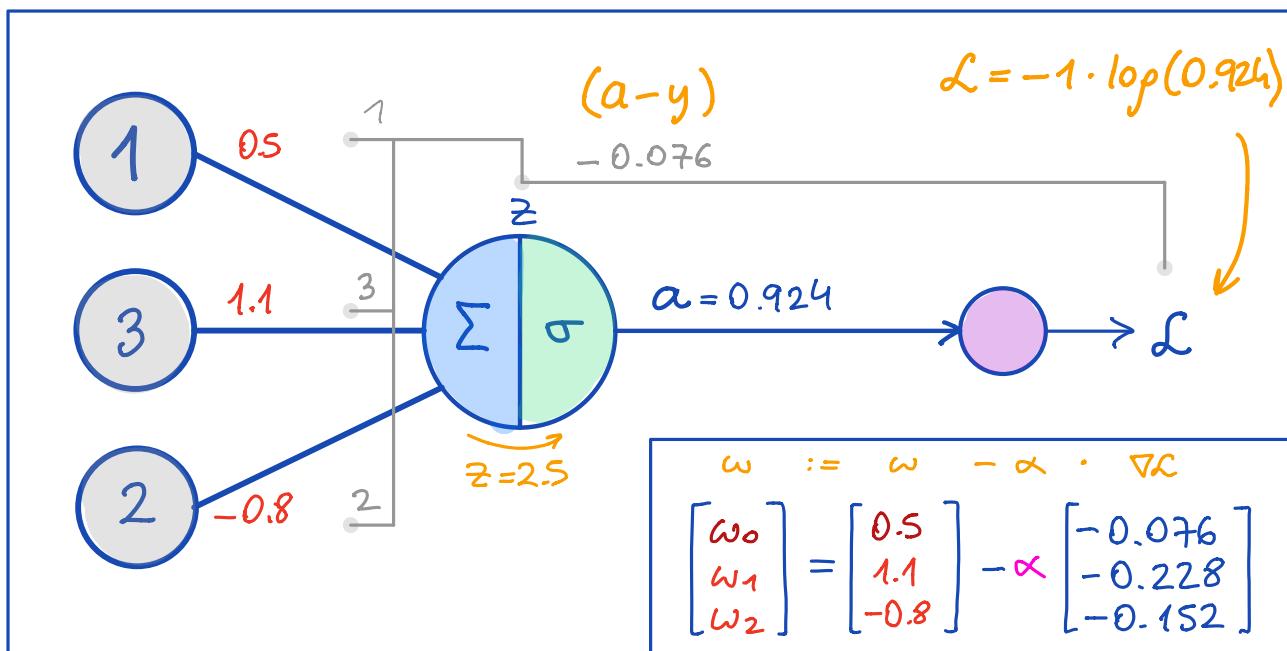
- Abhangigkeitsgraph - Overlay auf Netzwerk



Kettenregel : Multipliziere partielle Ableitungen auf dem "rückwärts weg", beginnend bei \mathcal{L} bis hin zum jeweiligen Punkt (Parameter).

BSP. $x = (1, 3, 2)$, $y = 1$, $\omega = (0.5, 1.1, -0.8)$

Forward pass: $z = \omega^T x = 2.5$, $a = \sigma(z) = 0.924$



- Vectorisierung (Ein Beispiel) ($m=1$) $x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$ $y = [\bullet]$

- $z = w^T x = \begin{bmatrix} w_0 & w_1 & w_2 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} z \end{bmatrix}$
- $\frac{\partial z}{\partial w} = x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$ (Koeffizienten von w)
(Gleiche Dimension wie w)
- $\frac{\partial J}{\partial w} = (a - y) \cdot x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \cdot \bullet$ $J = \mathcal{L}$

- Vectorisierung (Mehrere Beispiele) ($m=4$) $y = [\bullet^{(1)} \bullet^{(2)} \bullet^{(3)} \bullet^{(4)}]$

- $z = w^T X = \begin{bmatrix} w_0 & w_1 & w_2 \end{bmatrix} \begin{bmatrix} 1 & (1) \\ x_1 & (2) \\ x_2 & (3) \\ x_3 & (4) \end{bmatrix} = \begin{bmatrix} z^{(1)} & z^{(2)} & z^{(3)} & z^{(4)} \end{bmatrix}$
- $\frac{\partial z}{\partial w} = X = \begin{bmatrix} 1 & (1) \\ x_1 & (2) \\ x_2 & (3) \\ x_3 & (4) \end{bmatrix}$

- $\frac{\partial J}{\partial w} = \frac{1}{m} \cdot X \cdot (A - Y)^T = \frac{1}{4} \cdot \begin{bmatrix} 1 & (1) \\ x_1 & (2) \\ x_2 & (3) \\ x_3 & (4) \end{bmatrix} \cdot \begin{bmatrix} \bullet^{(1)} \\ \bullet^{(2)} \\ \bullet^{(3)} \\ \bullet^{(4)} \end{bmatrix}$
- $J = \frac{1}{4} \sum_{i=1}^4 \mathcal{L}(a^{(i)}, y^{(i)})$ Die einzelnen Verluste werden summiert!

$A - Y : 1 \times 4$
 $X : 3 \times 4$
 $\frac{\partial J}{\partial w} : 3 \times 1$

Gradient in Matrix-Form

$$\Rightarrow \nabla J = \frac{\partial J}{\partial w} = \frac{1}{m} \cdot X \left(J(w^T X) - Y \right)^T !$$

und

$$w_{\text{neu}} := w_{\text{alt}} - \frac{\alpha}{m} \cdot X \left(J(w^T X) - Y \right)^T !$$

Gewichtsaktualisierung

- Aktualisierung von Gewicht w_j

$$\frac{\partial J}{\partial w} = \frac{1}{m} \cdot X \cdot (A - Y)^T = \frac{1}{4} \cdot \underbrace{\begin{bmatrix} (1) & (2) & (3) & (4) \\ 1 & x_1 & x_2 & x_3 \end{bmatrix}}_{\text{Zeile } j} \cdot \underbrace{\begin{bmatrix} a-y \\ (1) \\ (2) \\ (3) \\ (4) \end{bmatrix}}_{\text{Zeile } j}$$

$$\Rightarrow \frac{\partial J}{\partial w_j} = \frac{1}{m} \cdot \sum_{i=1}^m x_j^{(i)} \cdot (a^{(i)} - y^{(i)})$$

$$\Rightarrow w_j := w_j - \frac{\alpha}{m} \cdot \sum_{i=1}^m (h(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}$$

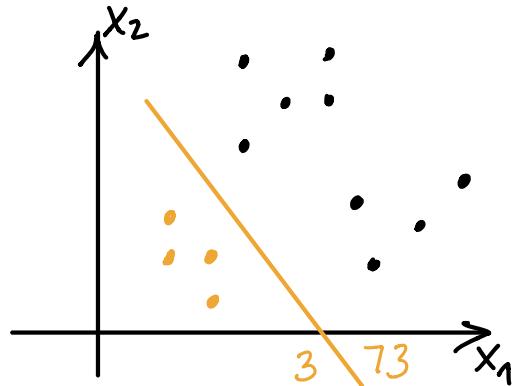
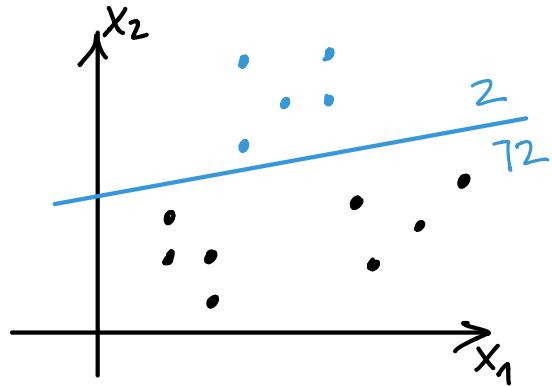
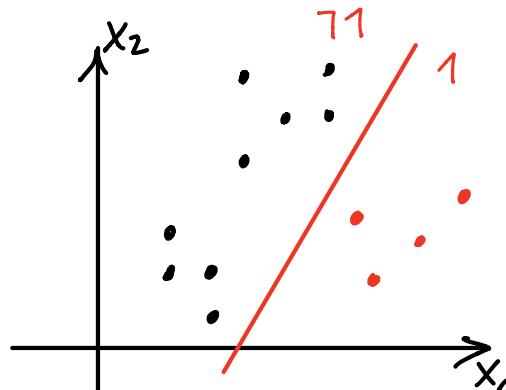
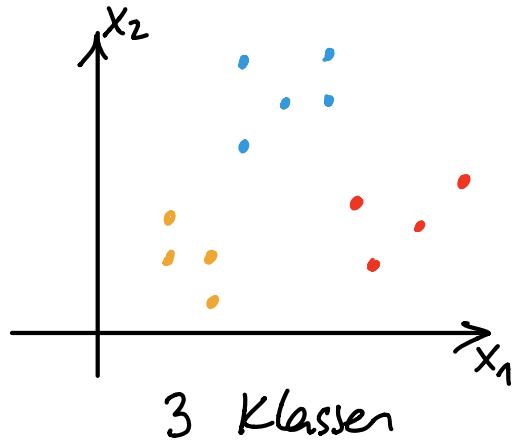
(Vergleiche mit linearer Regression)

- Andere Optimierungsmethoden

- Conjugate Gradient <<
 - BFGS <<
 - L-BFGS <<
- }
 - Etwas komplizierter.
 - Fertige Implementierungen verwenden.

Mehrklassige Klassifizierung

- Logistische Regression separat für alle k Klassen durchführen.
(Eines-versus-Rest) (one-versus-all)



- Klasse mit größter Wahrscheinlichkeit auswählen

$$\max_k h_k(x)$$

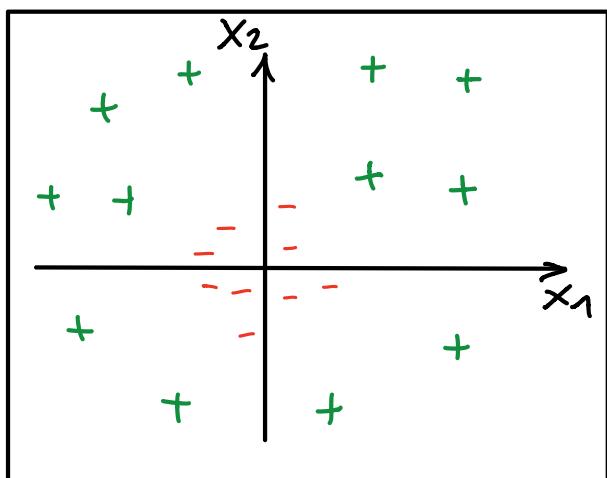
• Nicht-Lineare Entscheidungsgrenzen (Polynomiale Klassifizierung)

- Die Merkmale können auf nicht-lineare Weise kombiniert werden.

e.g. x_1^2 , $x_1 \cdot x_2$, x_2^2 , $x_1^2 \cdot x_2$ etc.
neue Merkmale [e.g. für einen Datenpunkt:
 $x_2 = 4 \Rightarrow x_2^2 = 16$]

BSP. Durch die Einführung der neuen Merkmale $x_3 := x_1^2$ und $x_4 := x_2^2$ erhalten wir die neue Hypothesenfunktion:

$$h_{\omega}(x) = \sigma(\omega_0 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 \cdot x_1^2 + \omega_4 \cdot x_2^2)$$



- Daten nicht linear separierbar.
- Sei $\omega = (-4, 0, 0, 1, 1)$
- Entscheidungsgrenze = ?