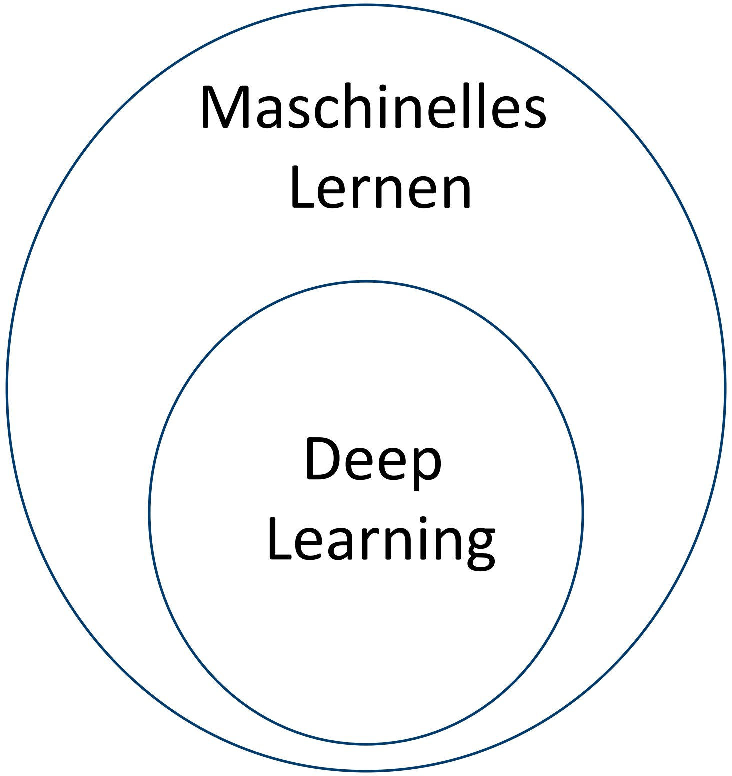
Hochschule Karlsruhe
University of
Applied Sciences

Deep Learning

Prof. Dr. Reimar Hofmann

Zum heutigen Vortrag

- Was ist Maschinelles Lernen?
  - Wie funktionieren insbesondere *Neuronale Netze*?
- Was versteht man unter *Deep Learning*?
  - *Rekurrente Neuronale Netze* als Anwendungsfall.
  - Das *Vanishing-Gradient-Problem*.
  - Wie wurde es gelöst? (LSTM-Modelle)
- Der jüngste Erfolg von Deep Learning: ChatGPT



Maschinelles Lernen

Klassische Softwareentwicklung:

Menschen legen fest,
wie auf welche Ereignisse
reagiert werden soll
(z.B. *wenn* – *dann* – Regeln)

Beispiel:

Online-Bankingsystem einer Bank

Maschinelles Lernen:

Menschen programmieren ein Verfahren,
wie der **Computer aus Trainingsbeispielen selbst erlernt**, wie er auf welche Inputs reagieren soll.

Beispiel:

Auf Kamerabildern andere
Verkehrsteilnehmer identifizieren.

Ein Beispiel für maschinelles Lernen

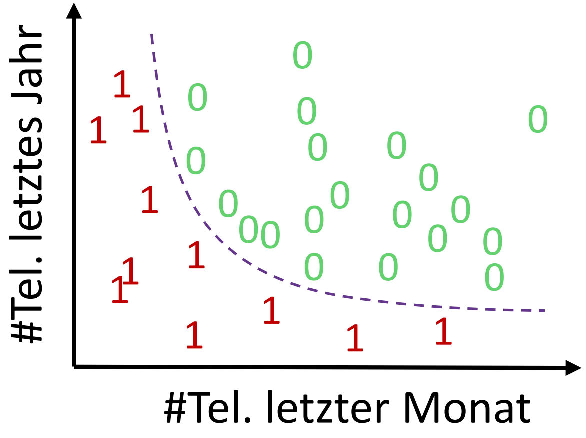
Vorhersage, welche Kunden ihren Handyvertrag kündigen werden:

Input-Vektor: $\begin{pmatrix} \text{Anzahl Telefongespräche im letzten Monat} \\ \text{Anzahl Telefongespräche im letzten Jahr} \\ \text{Datenvolumen im letzten Monat} \\ \text{Datenvolumen im letzten Jahr} \end{pmatrix}$

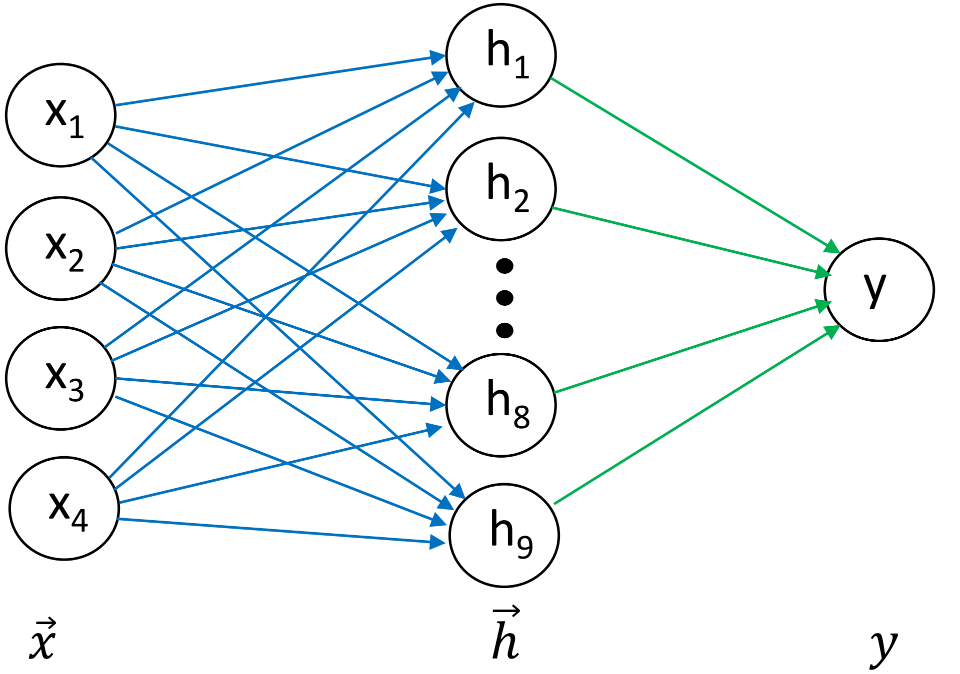
Output: 0 (kündigt nicht) oder 1 (kündigt)

Trainingsdaten: Liste von (Input, Output)-Paaren:

$$\underbrace{\left(\begin{pmatrix} 10 \\ 100 \\ 5 \\ 40 \end{pmatrix}, 0\right)}_{\text{Kunde 1}}, \underbrace{\left(\begin{pmatrix} 2 \\ 70 \\ 1 \\ 7 \end{pmatrix}, 1\right)}_{\text{Kunde 2}}, \underbrace{\left(\begin{pmatrix} 20 \\ 80 \\ 15 \\ 50 \end{pmatrix}, 0\right)}_{\text{Kunde 3}}, \ldots$$



Ein Ansatz für maschinelles Lernen: Neuronale Netze



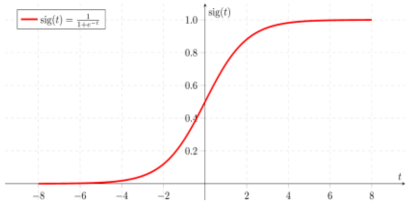
Architektur eines einfachen Neuronalen Netzes
(4 Inputs, 1 versteckte Schicht mit 9 Neuronen, 1 Output)

Berechnung des Outputs *y* aus gegebenen Inputwerten $x_1,\ldots,x_4$:

$$h_k \coloneqq sig\left(\sum_{j=1}^{4} x_j \cdot {}^{1}w_{kj}\right) \qquad (\text{für } k = 1..9)$$

$$y \coloneqq \sum_{j=1}^{9} h \cdot {}^{2}w_j$$

sig(*x*) : Sigmoid-Funktion



Von MartinThoma - Eigenes Werk, CC0, https://commons.wikimedia.org/w/index.php?curid=32970804

In Vektorschreibweise:

$$y = f(\vec{x}) = f_2(f_1(\vec{x}))$$ mit

$f_1(\vec{x}) \coloneqq sig({}^{1}W \cdot \vec{x})$, ${}^{1}W$: Gewichts-Matrix der ersten Schicht,

$f_2(\vec{h}) \coloneqq {}^{2}W \cdot \vec{h}$, ${}^{2}W$: Gewichts-Matrix der zweiten Schicht

Ein Neuronales Netz ist also eine durch die Gewichte parametrisierte Funktion.

Training von Neuronalen Netzen

Anschaulich:

-> Für jeden Parameter prüfen, wie stark eine geringfügige Erhöhung das Modell verbessern würde

-> Die Parameter jeweils einen *kleinen Schritt* in die richtige Richtung verbessern.

-> Wiederholen, bis das Modell stagniert.

Genauer: Lernen durch Gradientenabstieg

-> Ein Maß für die Menge an Fehlern auf den Trainingsdaten definieren (z.B. Summe der Fehlerquadrate)

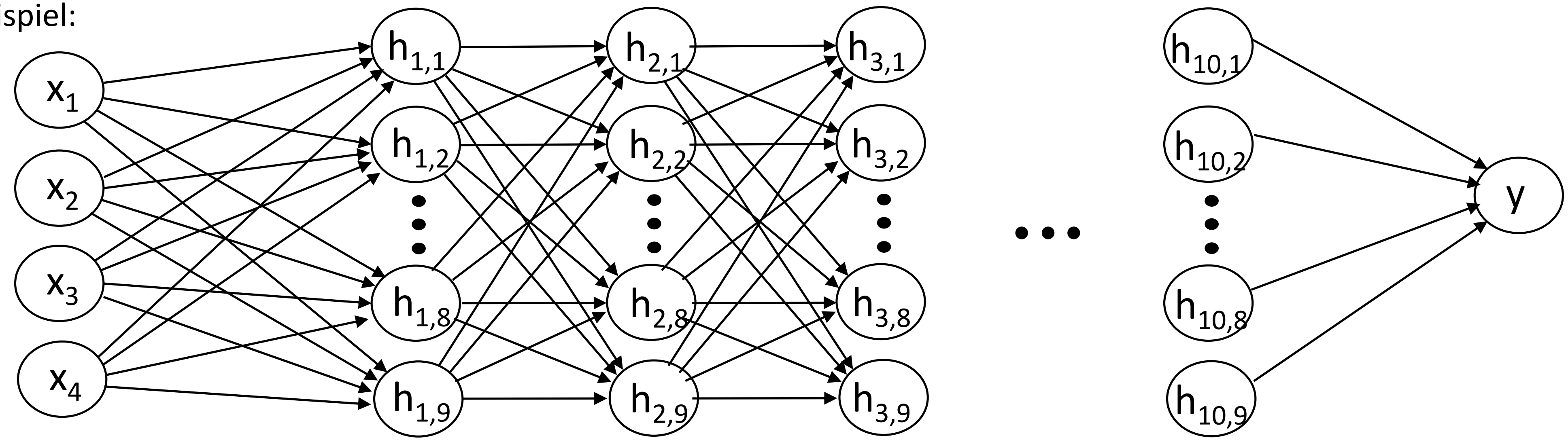
-> Für jeden Parameter die partielle Ableitung des Trainingsfehlers nach diesem Parameter berechnen.
(Der Vektor all dieser partiellen Ableitungen heißt Gradient.)

-> Den Parametervektor einen kleinen Schritt in Gegenrichtung des Gradienten verändern.

-> So lange wiederholen, bis das Modell stagniert.

Tiefe Neuronale Netze

Beispiel:



$$y = f(\vec{x}) = f_{10}\left(\dots\left(f_3\left(f_2\left(f_1(\vec{x})\right)\right)\right)\right)$$

- Tiefe Neuronale Netze **haben *viele Schichten***
- Können **mehrstufige Berechnungen** erlernen
- Sind insbesondere für **Bild- und Sprachverarbeitung** erfolgreich
- Treten z.B. auf, wenn Sequenzen variabler Länge als Input verarbeitet werden sollen

Wo treten tiefe neuronale Netze auf? (1)

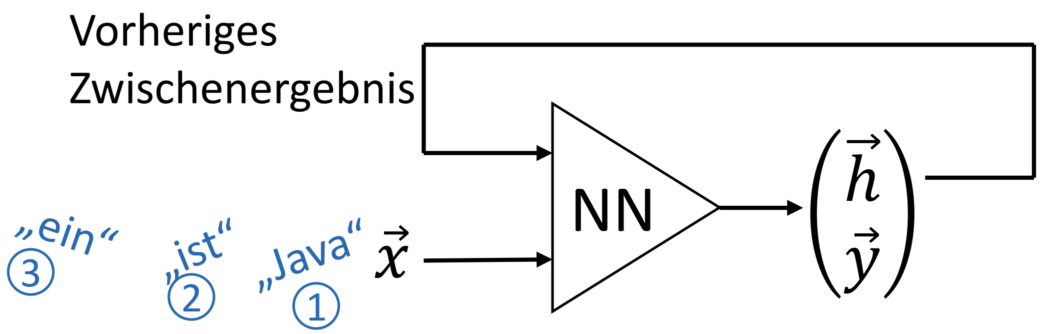
Rekurrentes Neuronales Netz als Chatbot:

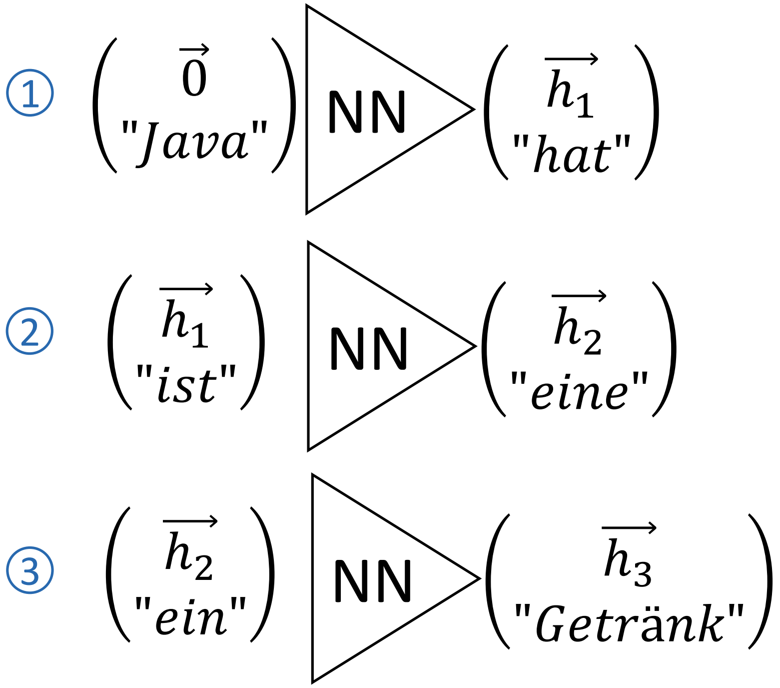
Input: $\begin{pmatrix} \text{letztes Zwischenergebnis } \vec{h} \\ \text{ein Wort (codiert als Vektor } \vec{x}\text{)} \end{pmatrix}$

Output: $\begin{pmatrix} \text{neues Zwischenergebnis } \vec{h} \\ \text{prognostiziertes nächstes Wort im Satz} \end{pmatrix}$

Die Worte eines Satzes werden nacheinander jeweils

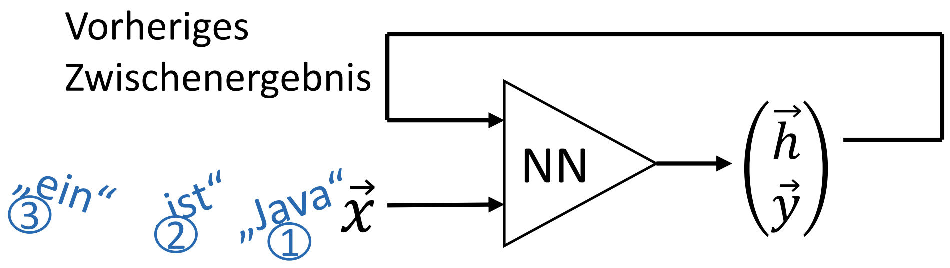
- als Input angelegt,
- das *Zwischenergebnis* aus dem vorherigen Schritt ebenfalls als Input,
- dann daraus berechnet:
  - das prognostizierte nächste Wort $\vec{y}$
  - das neue *Zwischenergebnis*



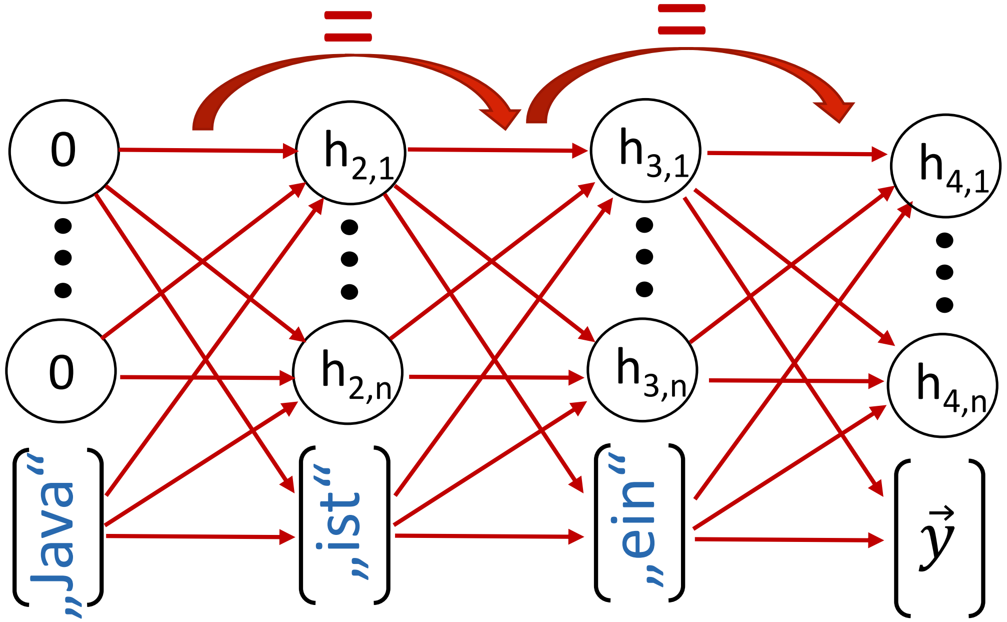


Wo treten tiefe neuronale Netze auf? (2)

Rekurrentes Neuronales Netz:



Ein Rekurrentes Neuronales Netz entspricht einem Tiefen Neuronalen Netz, bei dem die Gewichte in allen Schichten gekoppelt sind:



Die Tiefe entspricht der Länge der Input-Sequenz. So können auch **sehr** tiefe Netze entstehen.

Das „Vanishing Gradient“-Problem

Beispiel:

Berechnung der Ableitung des Outputs nach einem Parameter *w* aus der Schicht 3:
w beeinflusst also das Ergebnis von f_3. Wir zerlegen:

$$y = f_{10}\left(\left(f_9\left(\ldots\left(f_3\left(\overrightarrow{h_2}\right)\right)\right)\right)\right)$$

also

$$\frac{\partial y}{\partial w} = \frac{\partial y}{\partial \overrightarrow{h_{10}}} \cdot \frac{\partial \overrightarrow{h_{10}}}{\partial w} = \frac{\partial y}{\partial \overrightarrow{h_{10}}} \cdot \frac{\partial \overrightarrow{h_{10}}}{\partial \overrightarrow{h_9}} \cdot \frac{\partial \overrightarrow{h_9}}{\partial w} = \frac{\partial y}{\partial \overrightarrow{h_{10}}} \cdot \frac{\partial \overrightarrow{h_{10}}}{\partial \overrightarrow{h_9}} \cdot \frac{\partial \overrightarrow{h_9}}{\partial \overrightarrow{h_8}} \cdot \ldots \cdot \frac{\partial \overrightarrow{h_4}}{\partial \overrightarrow{h_3}} \cdot \frac{\partial \overrightarrow{h_3}}{\partial w}$$

Je mehr Schichten, desto mehr Faktoren. Die einzelnen Faktoren sind oft betragsmäßig klein, d.h. jede Multiplikation mit einem weiteren Faktor verkleinert das Ergebnis.

Der Gradient der vorderen Schichten des Netzes wird also mit zunehmender Anzahl an Schichten exponentiell kleiner. Folge: **Kein effizientes Lernen möglich**.

Lösung des Vanishing-Gradient-Problems

„LongShortTermMemory“-Netze (**LSTM**) brachten den Durchbruch:

- Die einzelnen Neuronen haben eine komplexere Struktur:
  - Der Wert wird von einem zum nächsten Zeitschritt im Normalfall beibehalten.
  - Explizite „Gates“ entscheiden abhängig von den Inpute, ob zu dem Wert etwas hinzuaddiert werden soll, oder ob er gelöscht werden soll.

In der Folge:

- Durchbruch bei der Bildverarbeitung, z.B. Gesichtserkennung auf Fotos.
- Durchbruch bei der Sprachverarbeitung

Jüngster Erfolg: ChatGPT

- Erster Chatbot, mit dem man sich realistisch unterhalten kann. (-> Turing-Test)
- Relativ breites Weltwissen.
- Von kommerzieller Firma (OpenAI) entwickelt; nicht alle Details offengelegt.
- > 175 Milliarden Parameter
- Trainiert
  - auf öffentlich zugänglichen Texten (-> Trainiert die Antwort auf die Frage *„Was haben Leute geantwortet, wenn in Blogs die Frage xy gestellt wurde?“*)
  - zusätzlich explizit mit manuell für ChatGPT erzeugten Trainingsbeispielen

ChatGPT Beispiele

RE

Was ist die lösungsmenge der Gleichung 2x^2 = 2x+3?

Um die Lösungsmenge der Gleichung 2x^2 = 2x + 3 zu bestimmen, bringen wir die Gleichung zuerst in die Standardform:

2x^2 - 2x - 3 = 0

Als nächstes können wir die quadratische Formel anwenden, um die Lösungen zu finden:

x = (-b ± sqrt(b^2 - 4ac)) / 2a

Dabei sind a = 2, b = -2 und c = -3. Wenn wir diese Werte in die Formel einsetzen, erhalten wir:

x = (2 ± sqrt(2^2 - 4(2)(-3))) / (2(2))
x = (2 ± sqrt(4 + 24)) / 4
x = (2 ± sqrt(28)) / 4

Wir können die Wurzel von 28 vereinfachen, indem wir 28 als 4 * 7 schreiben:

x = (2 ± sqrt(4 * 7)) / 4
x = (2 ± 2sqrt(7)) / 4
x = (1 ± sqrt(7)) / 2

Daher ist die Lösungsmenge der Gleichung 2x^2 = 2x + 3:

{x | x = (1 + sqrt(7)) / 2 oder x = (1 - sqrt(7)) / 2}

✓

ChatGPT Beispiele

RE Was ist die Lösungsmenge der Ungleichung (5-x)/(x-2)>1?

Um die Lösungsmenge der Ungleichung zu finden, müssen wir zunächst das Intervall bestimmen, in dem die Ungleichung erfüllt ist.

Wir können dies tun, indem wir die Ungleichung auf beiden Seiten mit (x-2) multiplizieren, unter Berücksichtigung, dass das Vorzeichen umgedreht wird, wenn wir eine Ungleichung mit einer negativen Zahl multiplizieren: ✓

(5-x) > (x-2) * 1 X

Durch Ausmultiplizieren der rechten Seite erhalten wir:

(5-x) > x-2 ✓

Durch Ausmultiplizieren der rechten Seite erhalten wir:

(5-x) > x-2

Durch Umstellen der Ungleichung erhalten wir:

3x > 7 X

x > 7/3 ✓

Das bedeutet, dass die Ungleichung erfüllt ist, wenn x größer als 7/3 ist.

Die Lösungsmenge der Ungleichung lautet daher:

{x | x > 7/3} ✓

Das bedeutet, dass alle Werte größer als 7/3 die Ungleichung erfüllen. Folgefehler

ChatGPT Beispiele

RE

Was ist die Lösungsmenge der Ungleichung 4x^2>3x+1?

Um die Lösungsmenge der Ungleichung 4x^2 > 3x + 1 zu finden, müssen wir die Ungleichung zuerst in Standardform bringen, d.h. alle Terme auf eine Seite bringen und die Ungleichung auf der anderen Seite:

4x^2 - 3x - 1 > 0

Die nächste Schritt ist, die Nullstellen dieser quadratischen Gleichung zu finden, indem wir sie faktorisieren oder die quadratische Formel verwenden:

x1 = (3 + sqrt(37)) / 8

x2 = (3 - sqrt(37)) / 8

Rechenfehler:

$$(-3)^2 - 4 \cdot 4 \cdot (-1) = 25 \neq 37$$

Diese Nullstellen teilen die x-Achse in drei Intervalle, und wir müssen feststellen, in welchem Intervall die Ungleichung erfüllt ist. Um dies zu tun, können wir eine Testzahl in jedem Intervall auswählen und überprüfen, ob sie die Ungleichung erfüllt oder nicht. Wir wählen beispielsweise x = 0 für das Intervall (-∞, x2), x = 1 für das Intervall (x2, x1) und x = 2 für das Intervall (x1, +∞). Dann haben wir:

X

Für x = 0:

4x^2 - 3x - 1 = 4(0)^2 - 3(0) - 1 = -1 < 0

Daher ist die Ungleichung für x < x2 nicht erfüllt.

Für x = 1:

4x^2 - 3x - 1 = 4(1)^2 - 3(1) - 1 = 0

Daher ist die Ungleichung für x zwischen x2 und x1 erfüllt, d.h. x2 < x < x1.

Für x = 2:

4x^2 - 3x - 1 = 4(2)^2 - 3(2) - 1 = 11 > 0

X

Daher ist die Ungleichung für x > x1 erfüllt.

Daher ist die Lösungsmenge der Ungleichung 4x^2 > 3x + 1:

{x | x < (3 - sqrt(37)) / 8 oder x > (3 + sqrt(37)) / 8}

Hochschule Karlsruhe
University of
Applied Sciences

Vielen Dank!

www.h-ka.de