



Künstliche Intelligenz – Perzeptron



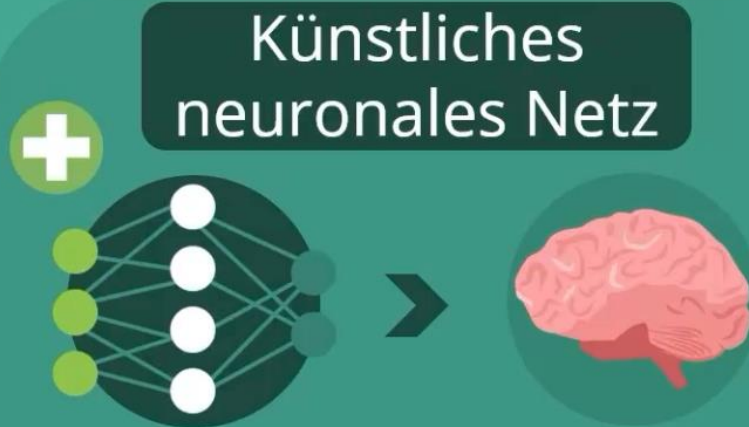
Künstliche Intelligenz

- Anwendungen Künstlicher Intelligenz als (Künstliches) Neuronales Netz (KNN)

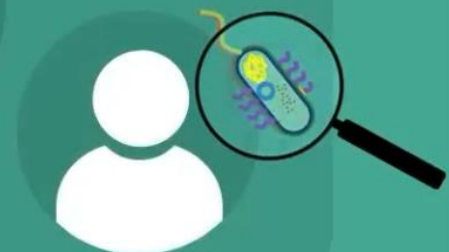


Künstliche Intelligenz – Neuronale Netze

Was ist ein neuronales Netz?



künstliche Intelligenz



Wollfaden-Perzeptron

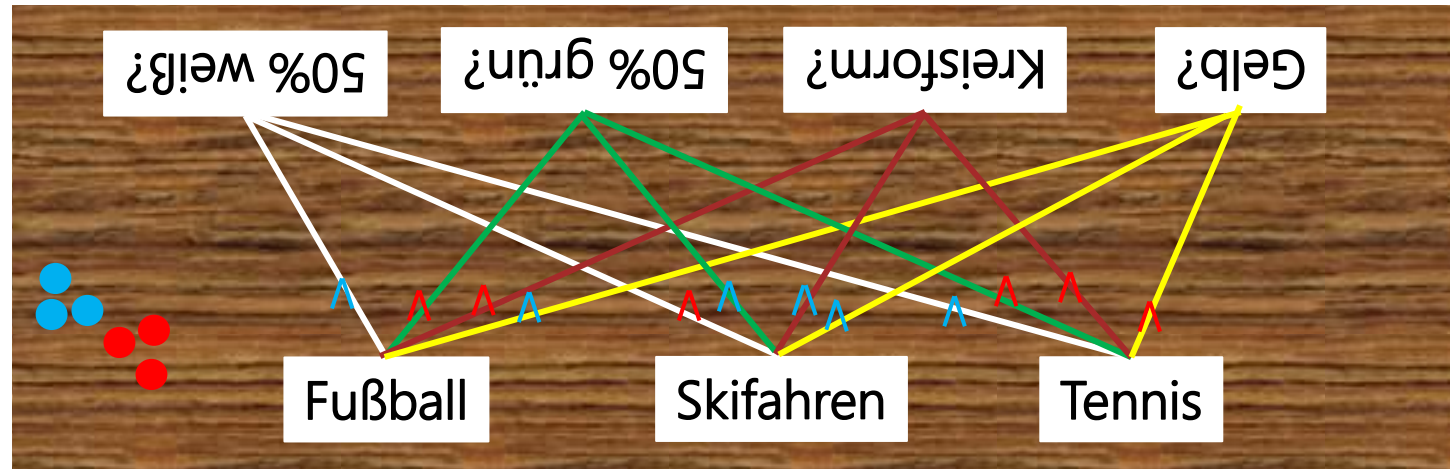
Material:

- Paketschnur / Wolle
- Wäscheklammern rot/blau
- Chips rot/blau
- Stellschilder mit Merkmalen
- Stellschilder mit Ergebnissen (0/1)
- Motivbilder
- Anleitungen für Spielleiter, Gehirn-Gruppe, Augen-Gruppe



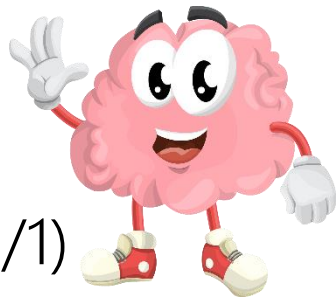
Augen-Gruppe

- Kann sehen, aber nicht denken
- Kann vorher festgelegte Merkmale erkennen



Gehirn-Gruppe

- Kann denken, aber nicht sehen
- Sind Ausgabeneuronen
- Jede/r hat eine Karte mit dem Ergebnis (0/1)



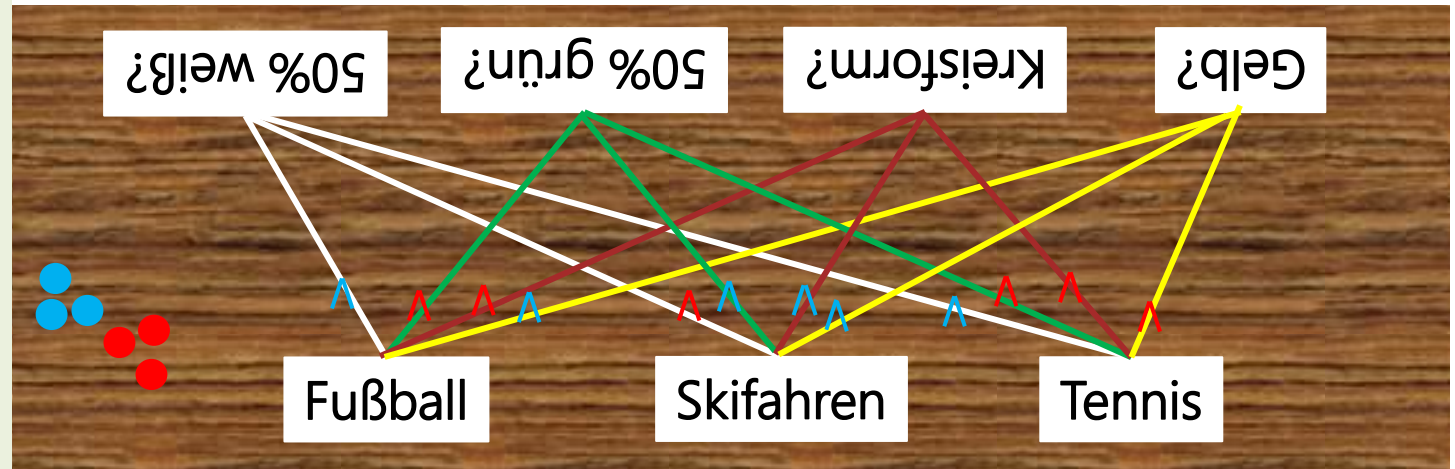
Wolffaden-Perzeptron - Bildverarbeitung

1. Hinter der Gehirn-Gruppe wird ein Bild hochgehalten.
2. Augen-Gruppe stellt für jedes sichtbare Merkmal das passende Stellschild auf.
3. Gehirn-Gruppe schaut, was durch die Verbindungen bei ihnen ankommt: Die Ausprägung jedes Merkmals (0/1) multiplizieren sie mit der Anzahl der Wäscheklammern auf der Schnur und legen Chips in der passenden Anzahl und Farbe unter die Klammern.
4. Gehirn-Gruppe zeigt Ergebnis: Wer mehr blaue als rote Chips hat, hält die blaue 0 hoch. Bei mehr roten als blauen Chips oder Gleichstand, wird die rote 1 gezeigt.



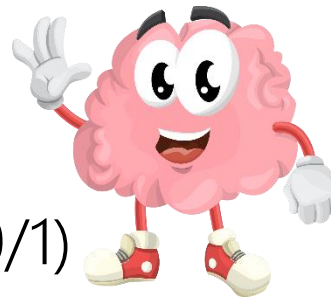
Augen-Gruppe

- Kann sehen, aber nicht denken
- Kann vorher festgelegte Merkmale erkennen



Gehirn-Gruppe

- Kann denken, aber nicht sehen
- Sind Ausgabeneuronen
- Jede/r hat eine Karte mit dem Ergebnis (0/1)



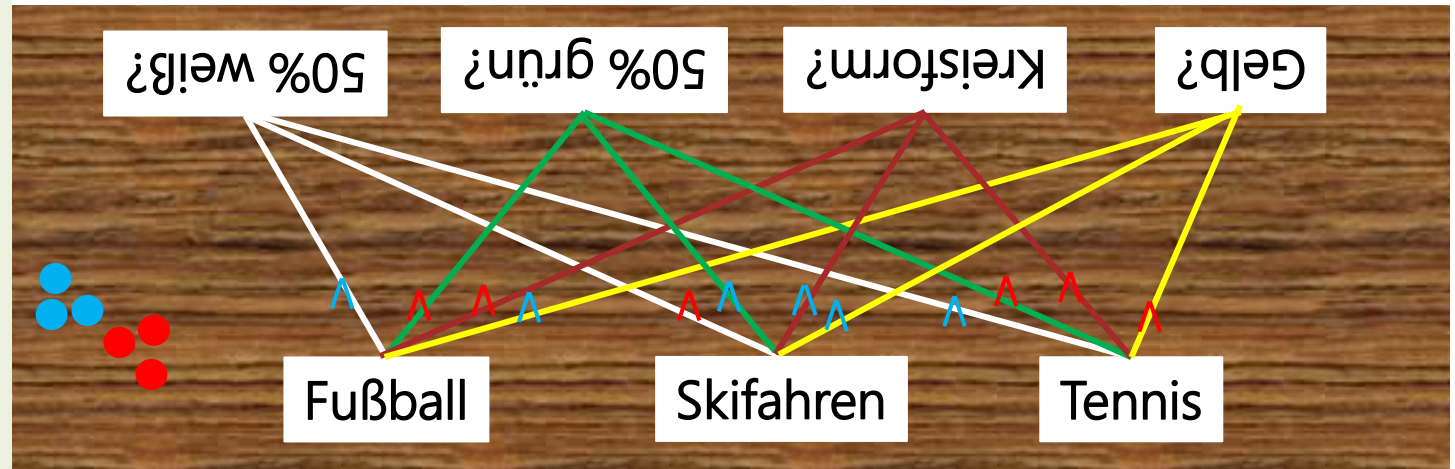
Wolffaden-Perzeptron - Aus Fehlern lernen

1. Alle Gehirn-Schüler, die richtig liegen, müssen nichts tun.
2. Alle anderen müssen umfärben.
 - Wurde **Rot statt Blau** gezeigt, dann werden alle Schnüre, durch die eine 1 angekommen ist, blauer. D.h. blaue Schnüre und leere Schnüre bekommen eine blaue Klammer, rote Schnüre verlieren eine rote Klammer.
 - Wurde **Blau statt Rot** gezeigt, dann werden alle Schnüre, durch die eine 1 angekommen ist, roter. D.h. rote Schnüre und leere Schnüre bekommen eine rote Klammer, blaue Schnüre verlieren eine blaue Klammer.



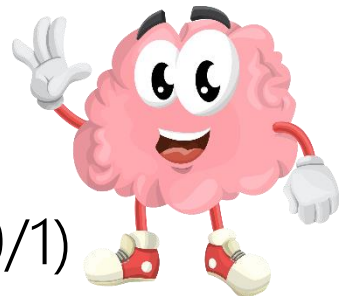
Augen-Gruppe

- Kann sehen, aber nicht denken
- Kann vorher festgelegte Merkmale erkennen



Gehirn-Gruppe

- Kann denken, aber nicht sehen
- Sind Ausgabeneuronen
- Jede/r hat eine Karte mit dem Ergebnis (0/1)



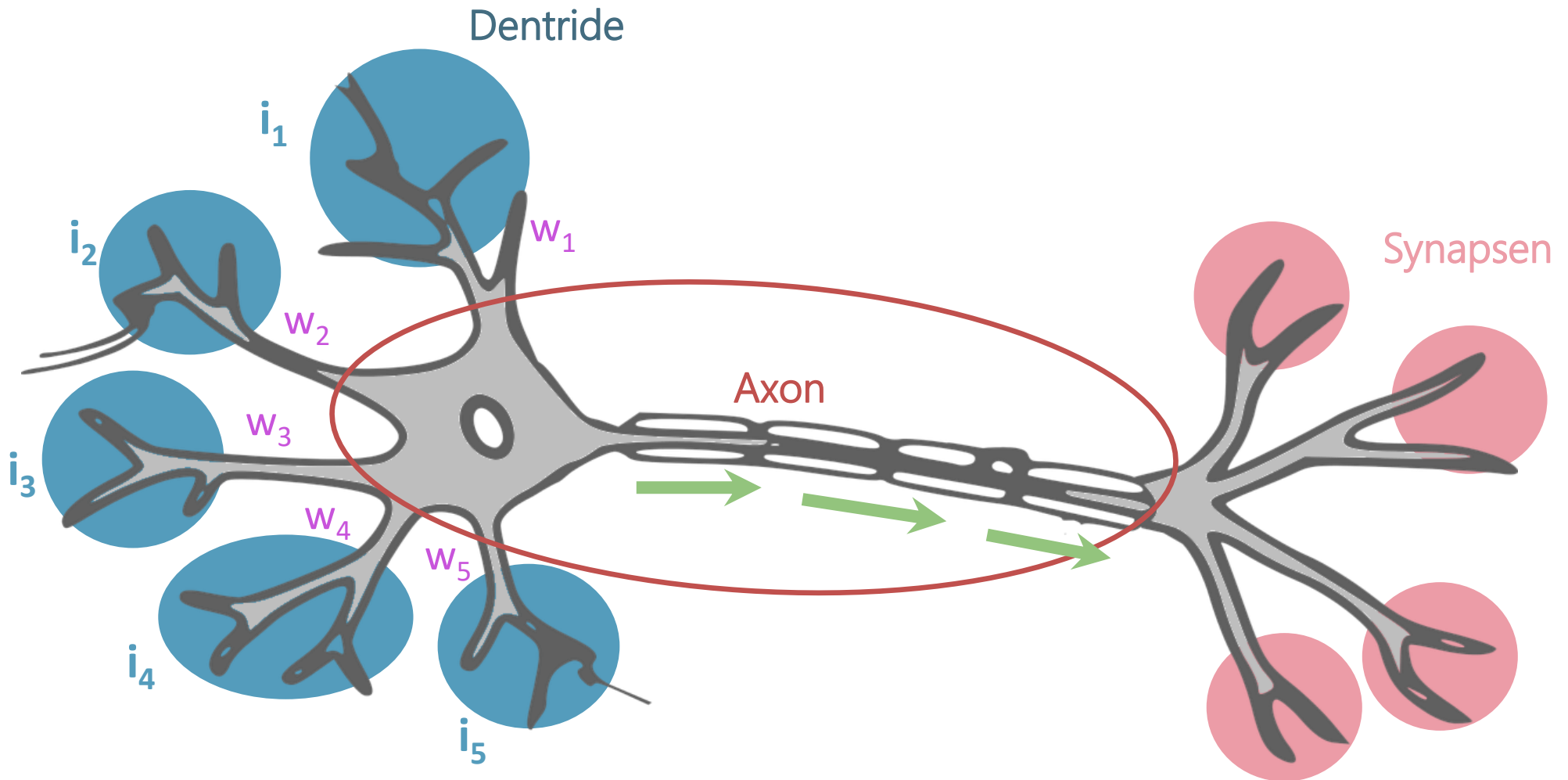
Übersicht

1. Hinter der Gehirn-Gruppe wird ein Bild hochgehalten.
 2. Augen-Gruppe stellt für jedes sichtbare Merkmal das passende Stellschild auf.
 3. Gehirn-Gruppe schaut, was durch die Verbindungen bei ihnen ankommt: Die Ausprägung jedes Merkmals (0/1) multiplizieren sie mit der Anzahl der Wäscheklammern auf der Schnur und legen Chips in der passenden Anzahl und Farbe unter die Klammern.
 4. Gehirn-Gruppe zeigt Ergebnis: Wer mehr blaue als rote Chips hat, hält die blaue 0 hoch. Bei mehr roten als blauen Chips oder Gleichstand, wird die rote 1 gezeigt.
1. Alle Gehirn-Schüler, die richtig liegen, müssen nichts tun.
 2. Alle anderen müssen umfärben.
 - Wurde **Rot statt Blau** gezeigt, dann werden alle Schnüre, durch die eine 1 angekommen ist, blauer. D.h. blaue Schnüre und leere Schnüre bekommen eine blaue Klammer, rote Schnüre verlieren eine rote Klammer.
 - Wurde **Blau statt Rot** gezeigt, dann werden alle Schnüre, durch die eine 1 angekommen ist, roter. D.h. rote Schnüre und leere Schnüre bekommen eine rote Klammer, blaue Schnüre verlieren eine blaue Klammer.
- Alle Schritte werden erst mit allen Bildern (nach jedem Bild Chips zurücklegen) wiederholt bis der Stapel durch ist.
 - Danach wird wieder von vorne begonnen und so lange wiederholt bis alle Bilder richtig erkannt werden.

Wollperzeptron – Nachbesprechung

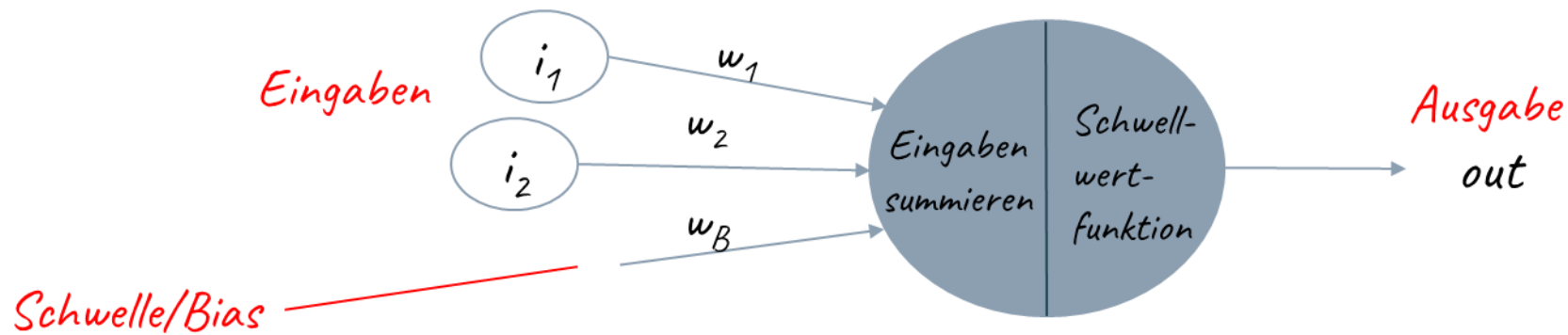
1. Was kann das Perzeptron vor und nach den Durchgängen?
2. Welches Ergebnis erhaltet ihr?
Seid ihr zufrieden?
3. Welche Bedeutung haben die Wäscheklammern und die farbigen Chips?

Das Perzeptron – Vorbild Nervenzelle

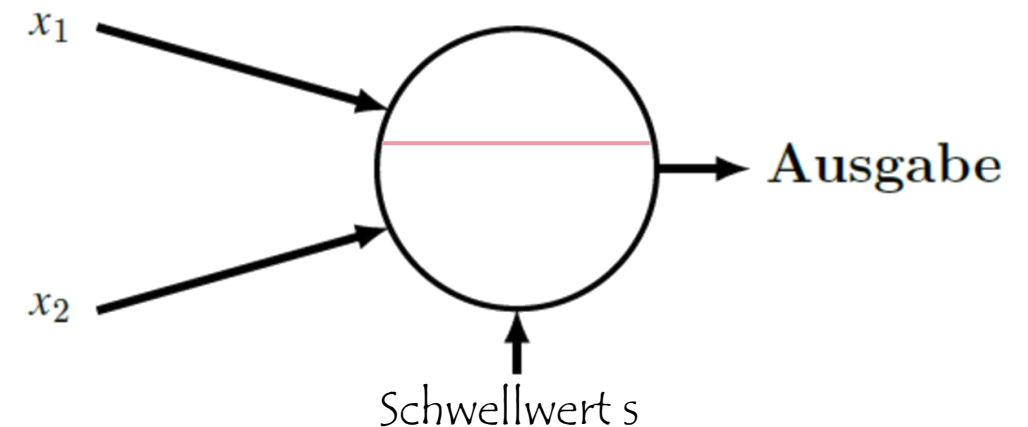


Künstliche Intelligenz – Perzeptron

- Reduktion auf zwei Eingaben



Übertragung in ein
mathematisches Modell

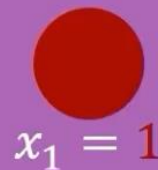


Funktionsweise Perzeptron

Funktionsweise



Neuron 1



$W_1 = 0,4$

Neuron 3



feuert mit 0,1

Neuron 2



$W_2 = -0,5$

$$x = 1 \cdot 0,4 + 0,6 \cdot -0,5 = 0,1$$



$$f(x) = \begin{cases} x & : x > 0 \\ 0 & : x \leq 0 \end{cases}$$

BlueJ-Projekt Perzeptron – Werte berechnen

Berechnete Ergebnisse

Sichtbares Koordinatensystem im Simulator:

$$0 < x < 4$$
$$0 < y < 4$$

Gerade einfügen

Gleichung: $x_2(x_1) = (-(-1.0/1.0)*x_1) + (0.0/1.0)$

Hinweis: x1 und x2 sind maximal 4 (für das Diagramm)

Tim Kleinöder

Berechne

x1: w1: -1.0

x2: w2: 1.0

s: 0.0

Formel: $x_1 * w_1 + x_2 * w_2 > S$

w1: +

w2: +

S: +

Berechne ein mal für zufällige x1 und x2

Berechne endlos für zufällige x1 und x2:

Start **Pause** **Ein weiterer Pu...** **Weiter** **Stop**

Geschwindigkeit

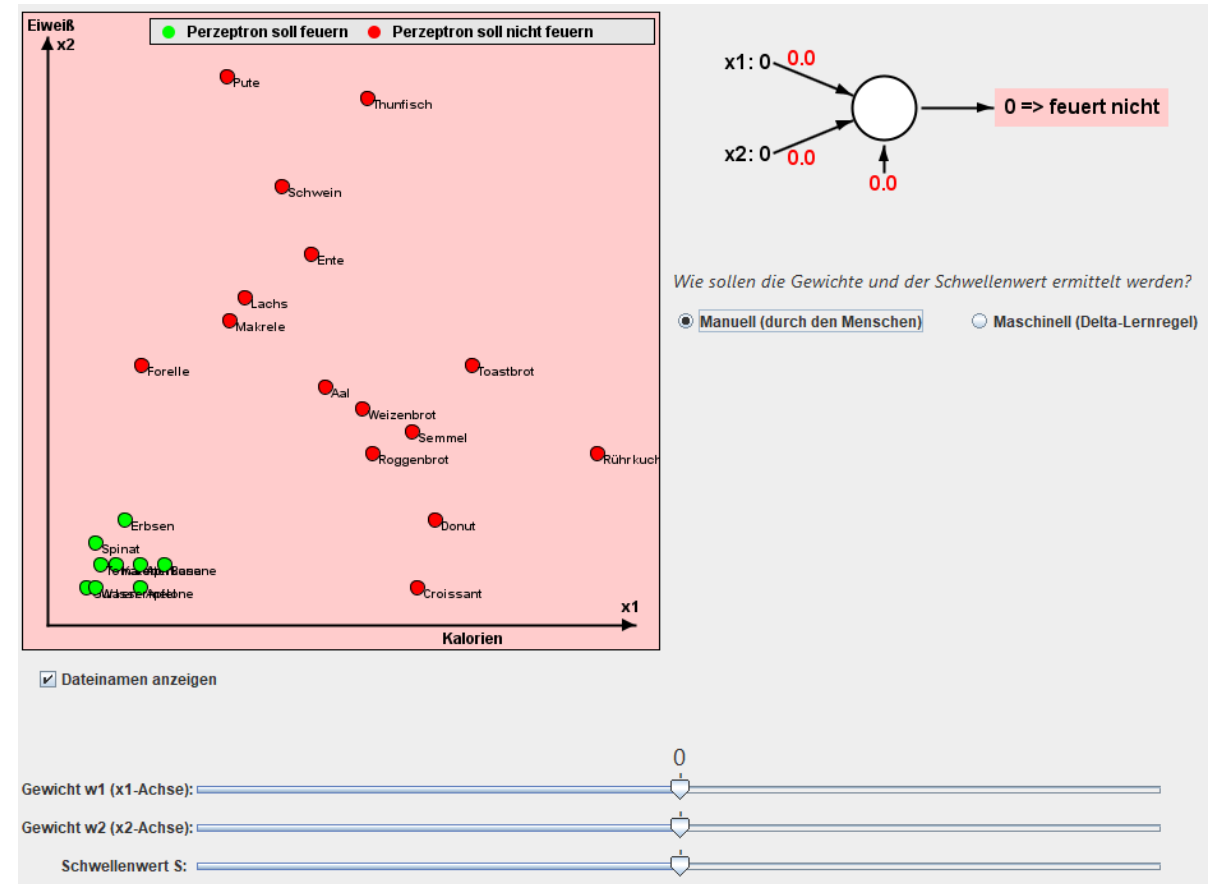
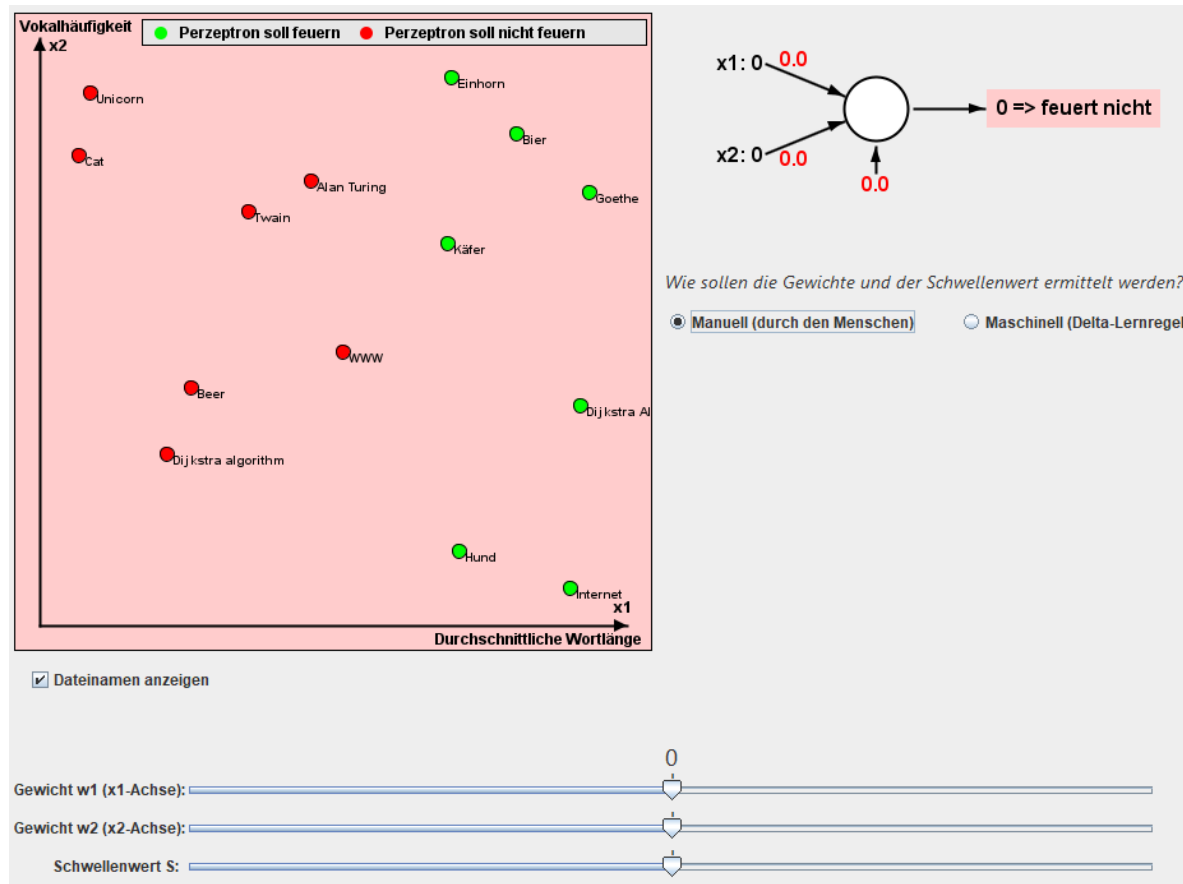
langsamer schneller

RESET

Diag. reset

BlueJ-Projekt Perzeptron – Gewichte einstellen 1+2

- Stelle die Gewichte und den Schwellenwert so ein, dass alle Datenpunkte richtig zugeordnet werden.



Perzeptron als lineare Funktion

- Zusammenhang mit linearer Funktion:

$$x_1 \cdot w_1 + x_2 \cdot w_2 = s \quad \begin{cases} \text{wenn größer } s, \text{ dann ja} \\ \text{wenn kleiner } s, \text{ dann nein} \end{cases}$$

$$x \cdot w_1 + y \cdot w_2 = s \quad | -x \cdot w_1$$

$$y \cdot w_2 = s - x \cdot w_1 \quad | : w_2$$

$$y = \frac{s - x \cdot w_1}{w_2}$$

$$= \frac{-w_1 \cdot x + s}{w_2}$$

$$= -\frac{w_1}{w_2} \cdot x + \frac{s}{w_2}$$

$$y = m \cdot x + t$$

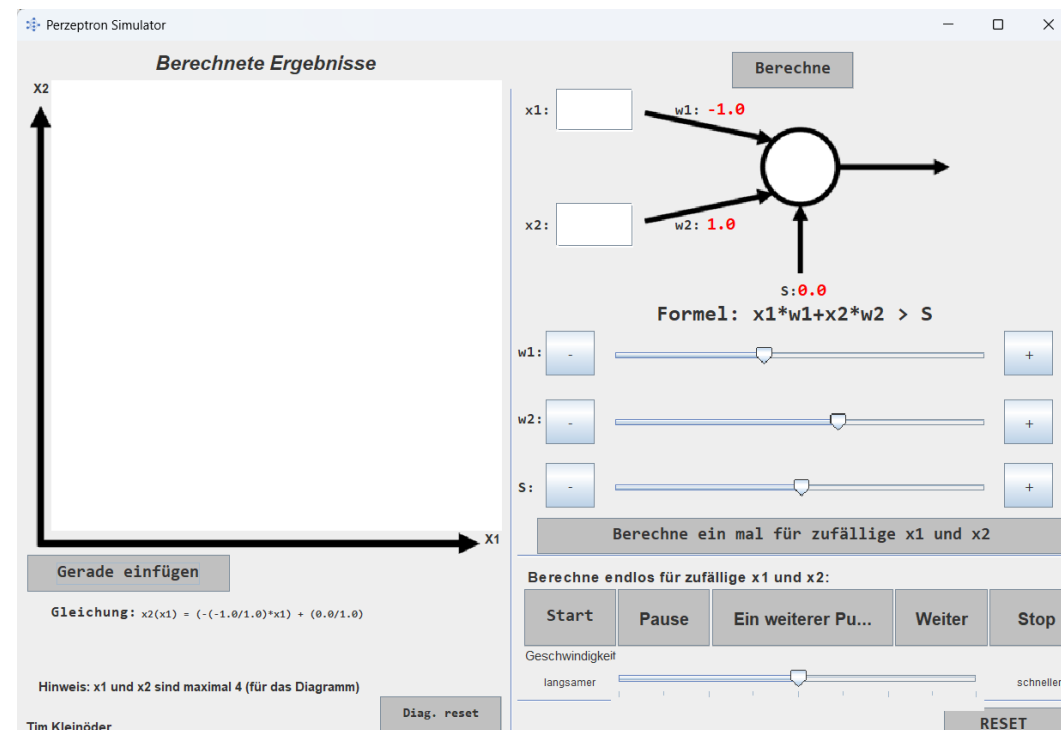
BlueJ-Projekt Perzeptron – Werte berechnen

- Berechne die Gewichte so, dass eine Trennung der Punkte von links oben nach rechts unten entsteht.
- Berechne die Gewichte so, dass eine Parallele zur x1-Achse entsteht.

Sichtbares Koordinatensystem
im Simulator:

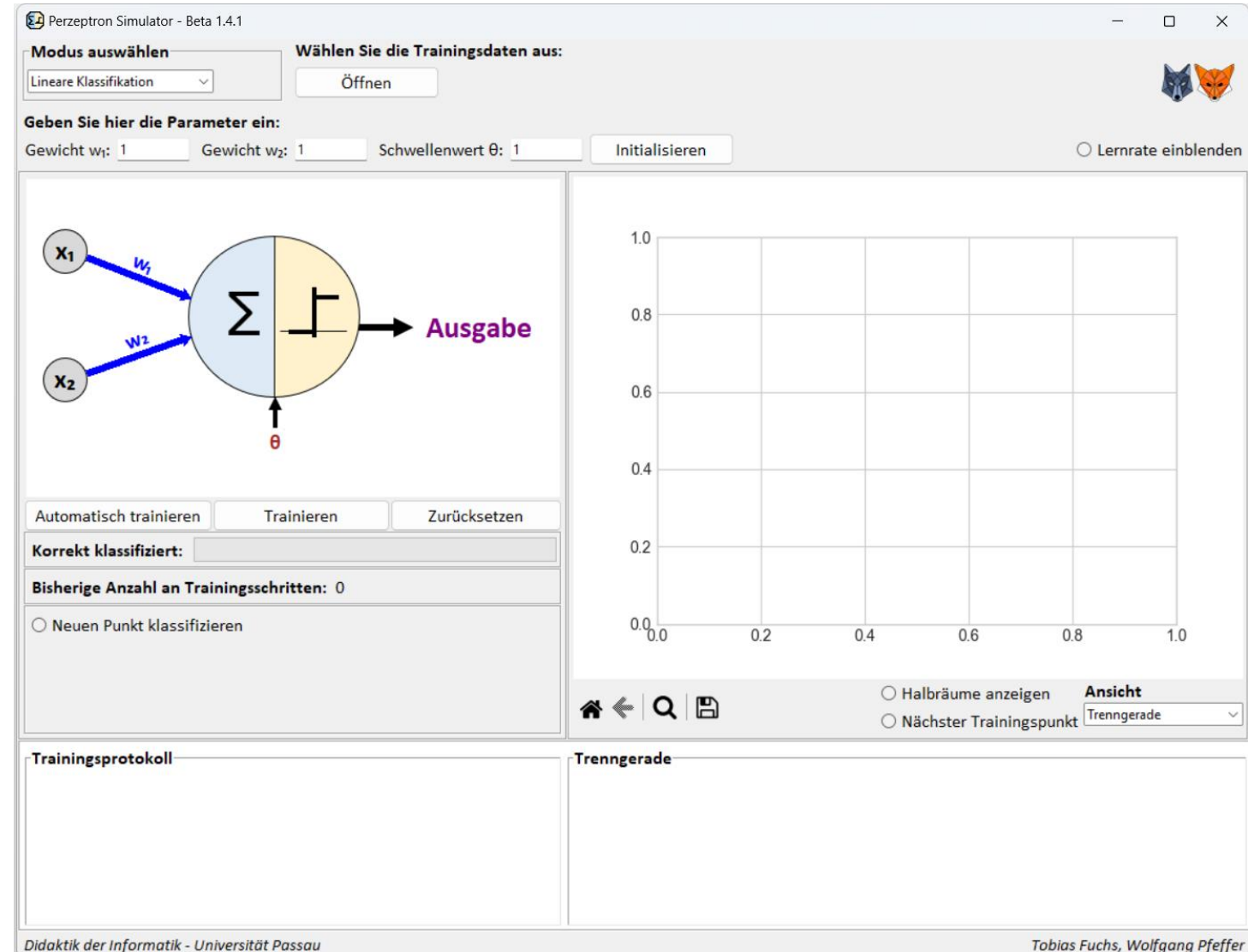
$$0 < x < 4$$

$$0 < y < 4$$



Perzeptron - Trainingsprozess

- Lernen – aber wie?
- Anpassung der Gewichte und des Schwellwerts



Perzeptron - Trainingsprozess

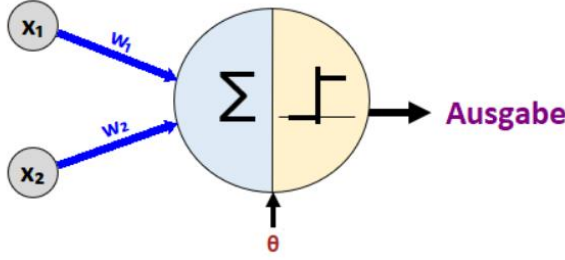
Perzeptron Simulator - Beta 1.4.1

Modus auswählen: Lineare Klassifikation

Wählen Sie die Trainingsdaten aus: **Öffnen**

Geben Sie hier die Parameter ein:

Gewicht w_1 : 1 Gewicht w_2 : 1 Schwellenwert θ : 1 Initialisieren ☐ Lernrate einblenden

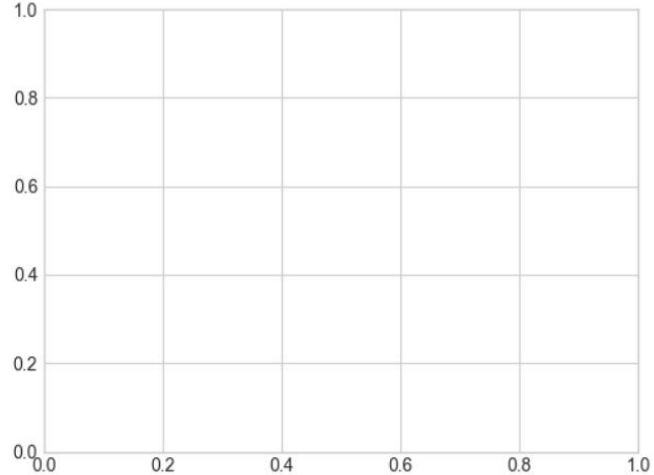


Automatisch trainieren Trainieren Zurücksetzen

Korrekt klassifiziert:

Bisherige Anzahl an Trainingsschritten: 0

☐ Neuen Punkt klassifizieren



Halbräume anzeigen Ansicht: Trenngerade

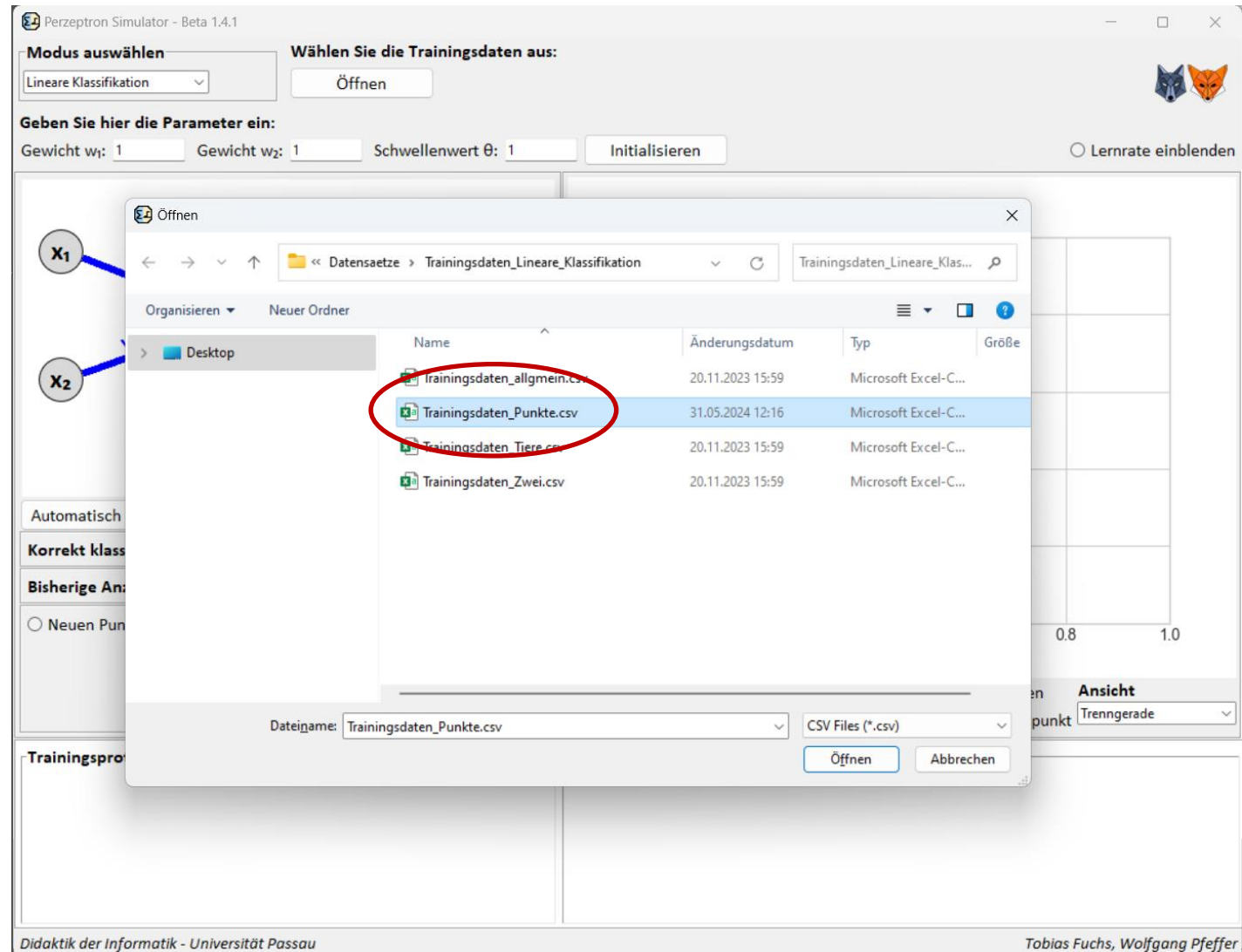
☐ Nächster Trainingspunkt

Trainingsprotokoll

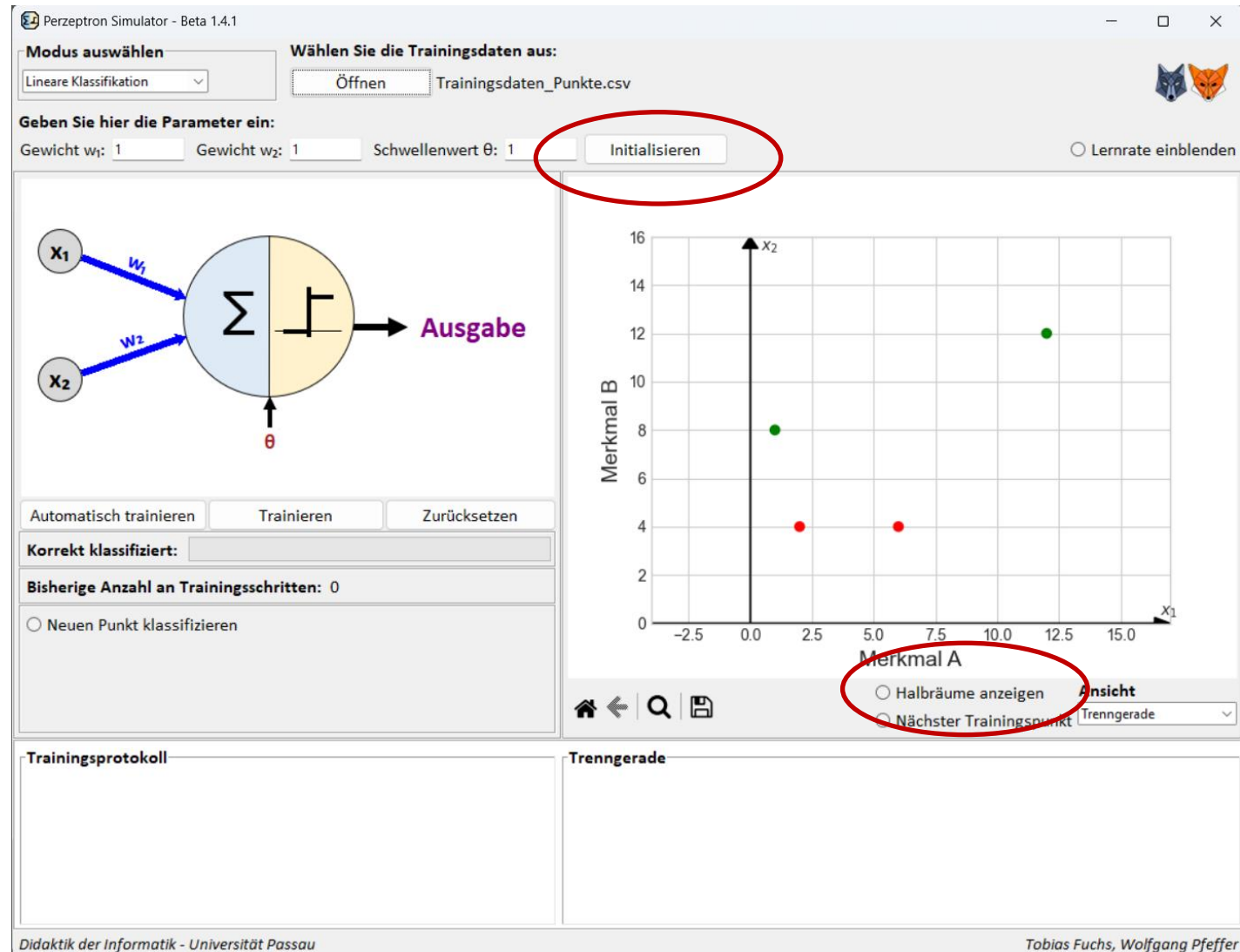
Trenngerade

Didaktik der Informatik - Universität Passau Tobias Fuchs, Wolfgang Pfeffer

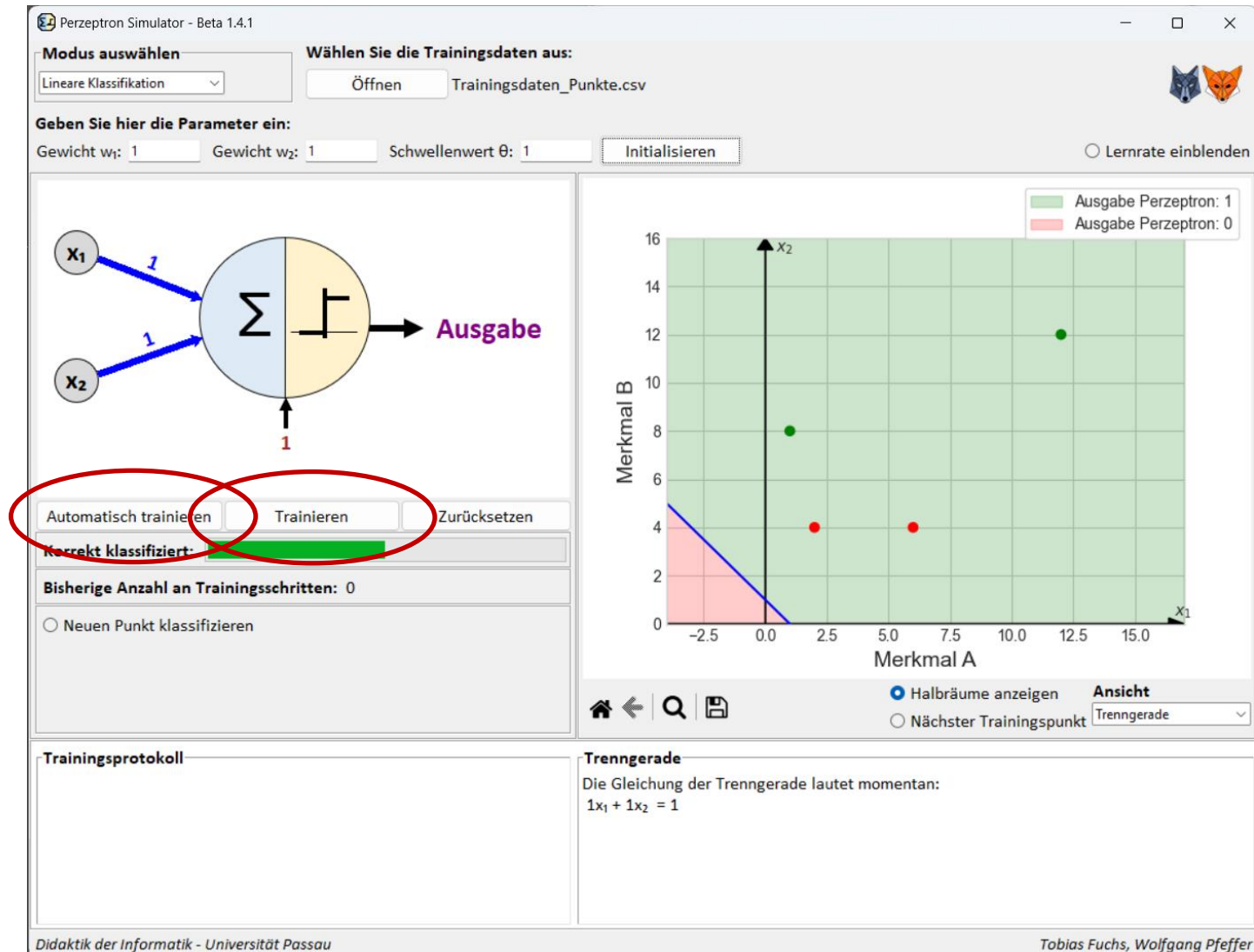
Perzeptron - Trainingsprozess



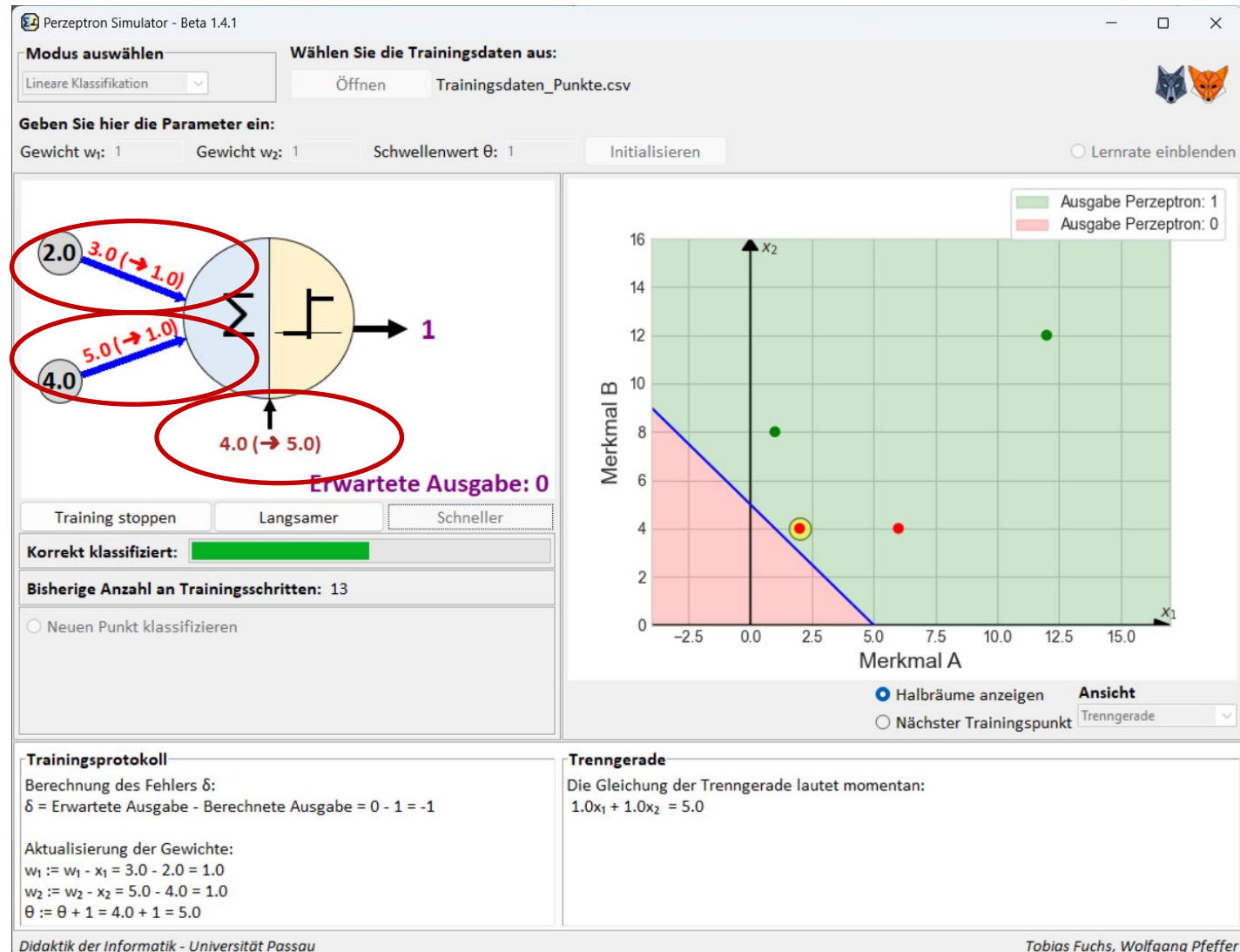
Perzeptron - Trainingsprozess



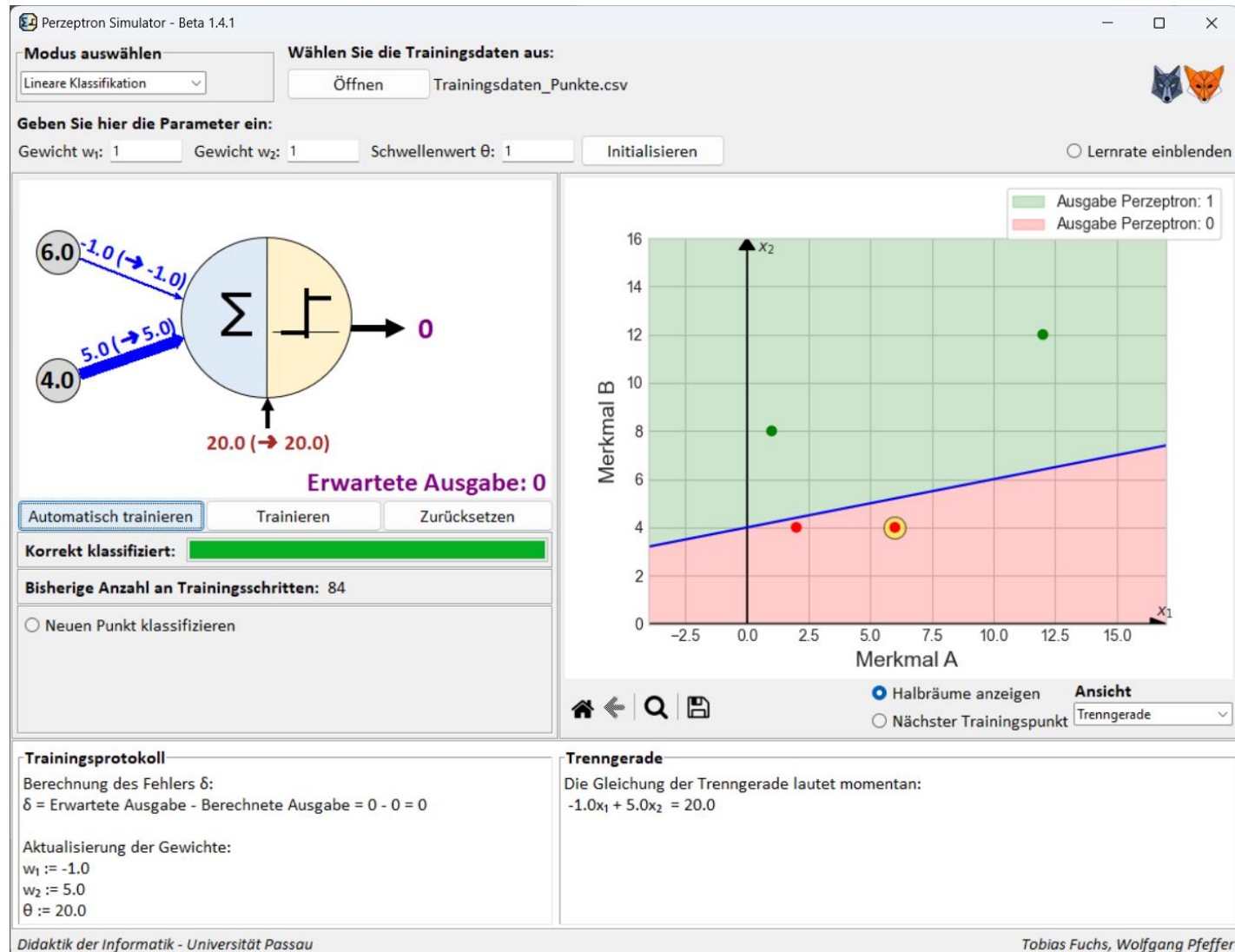
Perzeptron - Trainingsprozess



Perzeptron - Trainingsprozess



Perzeptron - Trainingsprozess



Keine weitere
Änderung, da alle
Punkte richtig
klassifiziert
werden!

$$w_1 = 1$$

$$w_2 = 1$$

$$s = 1$$

Perzeptron – Trainingsprozess Punkt A

- Perzeptron neu initialisieren und wir schauen uns den ersten Lernschritt mit dem ersten Punkt $A(0,5/1)$ mit Label 0 (also rot) nochmal im Detail an:

Berechnetes Label (output):

$$w_1 \cdot x_1 + w_2 \cdot x_2 \geq s$$

$$1 \cdot 0,5 + 1 \cdot 1 = 1,5 > 1 \rightarrow o = 1$$

Erwartetes Label (Wirklichkeit: Label des Punkts A)

$$\rightarrow t = 0$$

$$\rightarrow \text{Differenz: } \delta = t - o = 0 - 1 = -1$$

Der berechnete Wert des Perzeptron war zu hoch

→ Folge für Gewichte und den Schwellwert?

Änderung der Gewichte nach „unten“ und der Schwelle nach „oben“

$$w_1 = w_{1alt} - x_1 = 1 - 0,5 = 0,5$$

$$w_2 = w_{2alt} - x_2 = 1 - 1 = 0$$

$$s = s_{alt} + \delta = 1 + 1 = 2$$

$$w_1 = 0,5$$

$$w_2 = 0$$

$$s = 2$$

Perzeptron – Trainingsprozess Punkt B

- Wir schauen uns den zweiten Lernschritt mit dem ersten Punkt B(3/3) mit Label 1 (also grün) nochmal im Detail an:

Berechnetes Label (output):

$$w_1 \cdot x_1 + w_2 \cdot x_2 \geq s$$

$$0,5 \cdot 3 + 0 \cdot 3 = 1,5 < 2 \rightarrow o = 0$$

Erwartetes Label (Wirklichkeit: Label des Punkts B)

$$\rightarrow t = 1$$

$$\rightarrow \text{Differenz: } \delta = t - o = 1 - 0 = 1$$

Der berechnete Wert des Perzeptron war zu niedrig

→ Folge für Gewichte und den Schwellwert?

Änderung der Gewichte nach „oben“ und
der Schwelle nach „unten“

$$w_1 = w_{1alt} + x_1 = 0,5 + 3 = 3,5$$

$$w_2 = w_{2alt} + x_2 = 0 - 3 = -3$$

$$s = s_{alt} - \delta = 2 - 1 = 1$$

Perzeptron – Trainingsprozess Punkt C

Ausgangslage nach Initialisierung:

$$w_1 = 3,5$$

$$w_2 = 3$$

$$s = 1$$

- Wir schauen uns den dritten Lernschritt mit dem ersten Punkt C(0,25/2) mit Label 1 (also grün) nochmal im Detail an:

Berechnetes Label (output):

$$w_1 \cdot x_1 + w_2 \cdot x_2 \geq s$$

$$3,5 \cdot 0,25 + 3 \cdot 2 = 6,875 > 1 \rightarrow o = 1$$

Erwartetes Label (Wirklichkeit: Label des Punkts C)

$$\rightarrow t = 1$$

$$\rightarrow \text{Differenz: } \delta = t - o = 1 - 1 = 0$$

Der berechnete Wert des Perzeptron war passend

→ Folge für Gewichte und den Schwellwert?

Keine Änderung

$$w_1 = w_{1alt} = 3,5$$

$$w_2 = w_{2alt} = 3$$

$$s = s_{alt} = 1$$

$$w_1 = 3,5$$

$$w_2 = 3$$

$$s = 1$$

Perzeptron – Trainingsprozess Punkt D

- Wir schauen uns den vierten Lernschritt mit dem ersten Punkt D(1,5/1) mit Label 0 (also rot) nochmal im Detail an:

Berechnetes Label (output):

$$w_1 \cdot x_1 + w_2 \cdot x_2 \geq s$$

$$3,5 \cdot 1,5 + 3 \cdot 1 = 8,25 > 1 \rightarrow o = 1$$

Erwartetes Label (Wirklichkeit: Label des Punkts D)

$$\rightarrow t = 0$$

$$\rightarrow \text{Differenz: } \delta = t - o = 0 - 1 = -1$$

Der berechnete Wert des Perzeptron war zu hoch

→ Folge für Gewichte und den Schwellwert?

Änderung der Gewichte nach „oben“ und
der Schwelle nach „unten“

$$w_1 = w_{1alt} - x_1 = 3,5 - 1,5 = 2$$

$$w_2 = w_{2alt} - x_2 = 3 - 1 = 2$$

$$s = s_{alt} + \delta = 1 + 1 = 2$$

$$w_1 = 2$$

$$w_2 = 2$$

$$s = 2$$

Perzeptron – Trainingsprozess Punkt A

- Wir schauen uns den fünften Lernschritt mit dem ersten Punkt $A(0,5/1)$ mit Label 0 (also rot) nochmal im Detail an: usw.
- Das Prinzip bleibt das Gleiche und aber wir brechen hier mit der händischen Berechnung ab!
- Denn es bedarf meist **mehrerer Durchläufe** über alle Trainingsdaten bis das Perzeptron sich durch den Lernprozess so eingestellt hat, dass alle Label richtig berechnet werden!

Optimierung der Formeln

- Aus der Differenz δ lässt sich die „Richtung“ für Gewichtsänderung und Schwellenwert ableiten:

- $$\delta = t - o = \begin{cases} +1 & \rightarrow \text{"Gewichte hoch; Schwelle runter"} \\ 0 & \rightarrow \text{ohne Änderung} \\ -1 & \rightarrow \text{"Gewichte runter; Schwelle hoch"} \end{cases}$$

$$w_1 = w_{1alt} + \delta \cdot i_1$$

$$w_2 = w_{2alt} + \delta \cdot i_2$$

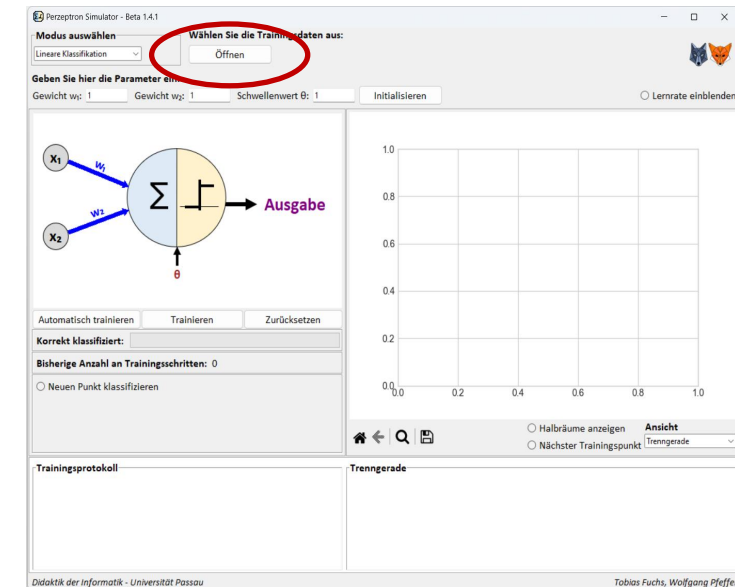
$$s = s_{alt} - \delta$$

i_1 und i_2 stehen
für den Input;
also die x- und y-
Werte der Punkte

- → Mit diesen Formeln bedarf es keiner Fallunterscheidung mehr!

Optimierung des Trainingsprozesses – Aufgabe

- Trainiere das Perzeptron mit verschiedenen Trainingsdaten:
Nutze hierfür nacheinander folgende Dateien:
 - Trainingdaten_Tiere.csv
 - Trainingsdaten_Punkte.csv
 - Trainingsdaten_Punkte-gestreckt.csv



- Was fällt dir für die verschiedenen Trainingsdaten auf?
- Wie wirken sich z. B. auch große Anfangswerte für die Gewichte/Schwelle auf den Trainingsprozess aus?

Delta-Lernregel mit Lernrate α

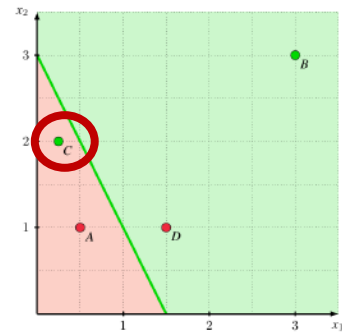
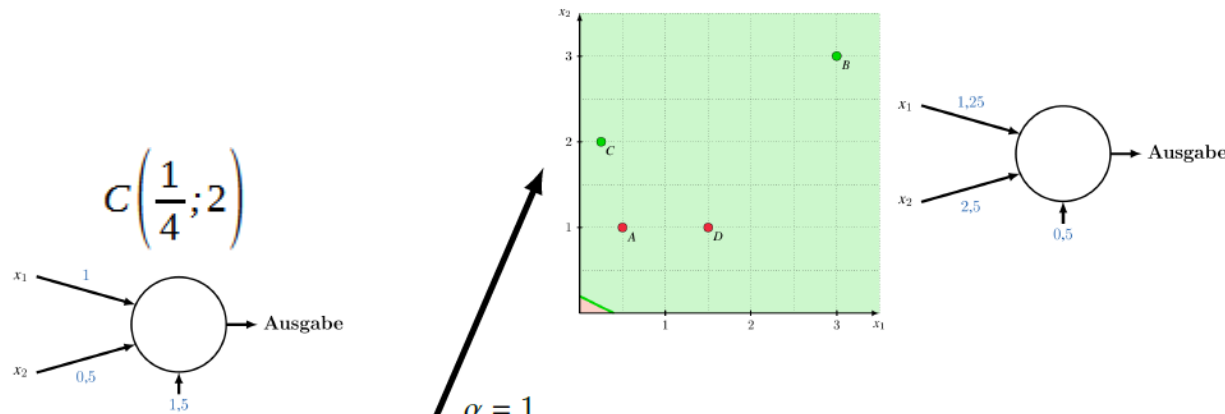
- Lösung: Lernrate α als zusätzlichen Parameter festlegen

$$w_1 = w_{1alt} + \delta \cdot \alpha \cdot i_1$$

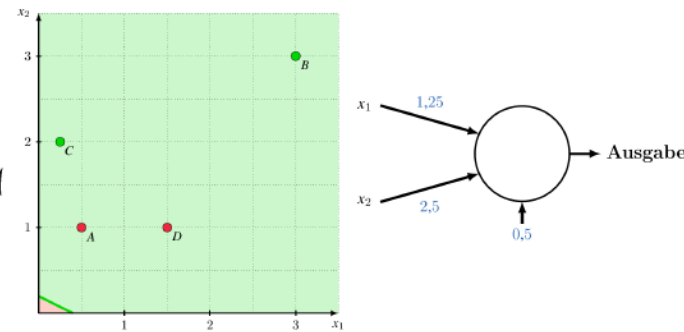
$$w_2 = w_{2alt} + \delta \cdot \alpha \cdot i_2$$

$$s = s_{alt} - \delta \cdot \alpha$$

Auswirkung der Lernrate auf den Trainingsprozess



$\alpha = 1$



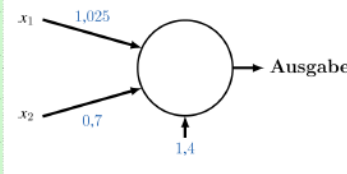
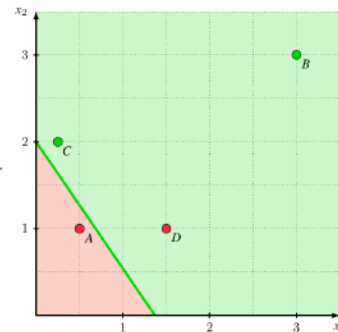
$$w_1 = 1 + 1 \cdot 1 \cdot \frac{1}{4} = 1,25$$

$$w_2 = 0,5 + 1 \cdot 1 \cdot 2 = 2,5$$

$$w_B = 1,5 - 1 \cdot 1 = 0,5$$

$\alpha = 1$

$\alpha = 0,1$



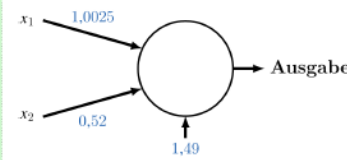
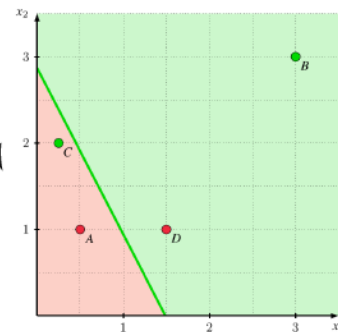
$$w_1 = 1 + 0,1 \cdot 1 \cdot \frac{1}{4} = 1,025$$

$$w_2 = 0,5 + 0,1 \cdot 1 \cdot 2 = 0,7$$

$$w_B = 1,5 - 0,1 \cdot 1 = 1,4$$

$\alpha = 0,1$

$\alpha = 0,01$



$$w_1 = 1 + 0,01 \cdot 1 \cdot \frac{1}{4} = 1,0025$$

$$w_2 = 0,5 + 0,01 \cdot 1 \cdot 2 = 0,52$$

$$w_B = 1,5 - 0,01 \cdot 1 = 1,49$$

$\alpha = 0,01$

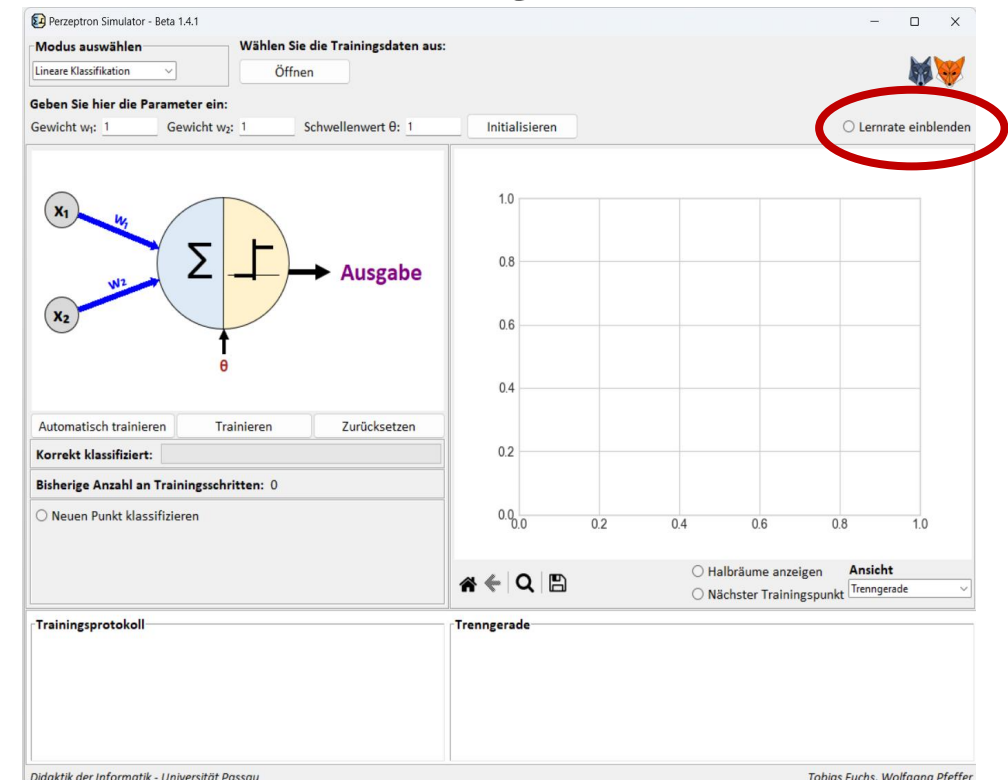
Ausgangslage

$$w_1 = 1 \quad w_2 = \frac{1}{2} \quad w_B = 1,5 \quad \delta = 1 - 0 = 1$$

Lernrate – Aufgabe

- Öffne die Trainingsdaten_Punkte-gestreckt.csv,
 - Blende die Lernrate ein
 - Wähle einen anderen Wert als 1 und initialisiere das Perzeptron neu und
 - Überprüfe, ob das Perzeptron mit dieser Lernrate schneller fertig trainiert ist

- Welche Lernratewert ist der Beste?
- Ist dieser Wert in jedem Szenario der Beste?



Güte des Perzeptron – Testen

- Bewertung mit Testdaten (sofern vorhanden) mittels Konfusionsmatrix:

Erwartetes Label (Realität)		Berechnetes Label (Vorhersage)		
		ja	nein	gesamt
	ja			
	nein			
	gesamt			

$$\text{Güte} = \frac{\text{Anzahl richtig klassifizierter Testdaten}}{\text{Gesamtanzahl aller Testdaten}}$$

Programmierung des Perzeptrons

Aufbau des Perzeptrons:

- Attribute
 - Gewichte w_1, w_2
 - Schwellwert s
 - Lernrate a
- Funktionalitäten/Methoden
 - Ausgabewert berechnen
Eingabe: Punktwerte x_1, x_2
Ausgabe: 0 oder 1 (rot oder grün)
 - Aus einem Datensatz lernen
 - Eingabe: Trainingsdatensatz mit erwartetem Label (x_1, x_2 und label/target)
 - Folge: Änderung der Gewichte bei Fehlklassifikation, sonst nichts

$$x_1 * w_1 + x_2 * w_2 \geq s$$

$$w_1 = w_{1alt} + \delta \cdot \alpha \cdot x_1$$

$$w_2 = w_{2alt} + \delta \cdot \alpha \cdot x_2$$

$$s = s_{alt} - \delta \cdot \alpha$$

Programmierung des Perzeptrons

- Öffne das BlueJ-Projekt Perzeptron Vorlage!
- Erweitere und implementiere nur in der Klasse **PERZEPTRON**!
- Das Perzeptron kann getestet werden, in dem eine neues Labor-Objekt erzeugt und die Methode **visualisiere()** aufruft.
- Mit Links- und Rechtsklick auf die Zeichenfläche können Datenpunkte eingefügt werden und durch die Methoden **trainiere...()** des Labors kann das Perzeptron trainiert werden!

Neuronale Netze – Ausblick 13. Jgst.



- Öffne das Tool Playground von tensorflow über den QR-Code oder per Doppelklick auf die URL-Verknüpfung [A Neural Network Playground](#) im Vorlagenverzeichnis!
- Betätige den Playbutton oben rechts und betrachte dabei schnell den Output.
- Verändere ein paar Parameter auf der linken Seite unterhalb von Data und lass jeweils das Perzeptron trainieren.
Wie hat sich die Änderung auf den Output ausgewirkt?
- Vergrößere nun schrittweise die Lernrate und vergleiche den Output.
- Erweitere nun das Perzeptron um Hidden Layers. Gehe hier schrittweise vor: erst ein, dann zwei usw. Lass jeweils das Perzeptron trainieren.
Wie macht sich die Hinzunahme weiterer Neuronen auf die Ausgabe bemerkbar?
- Verändere nun die Datenquelle zu der mit den vier Bereichen (rechts oben) und versuche mit möglichst wenig Neuronen insgesamt eine vollständige Klassifizierung zu erreichen, indem Du mit den Features experimentierst und die Anzahl der Layers bzw. jeweils der Neuronen einstellst.