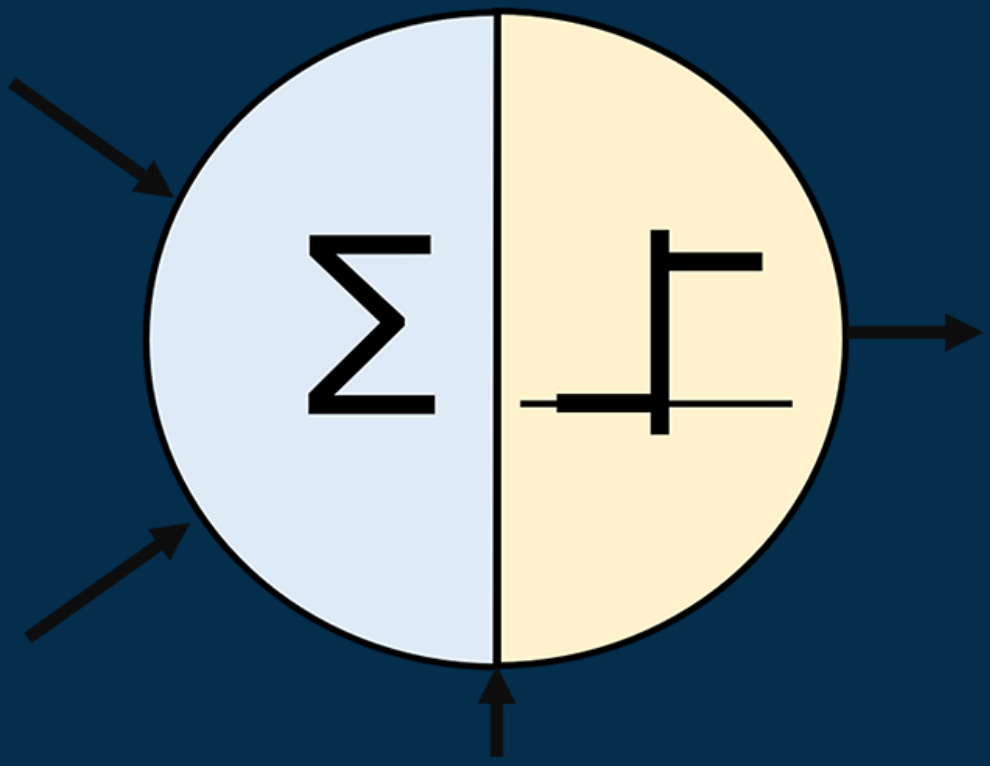
Handbuch

Simulationssoftware Perzeptron



Material zur Lehrerfortbildung KI@Informatik11

Konzeption des Handbuchs:

Das Handbuch dient zur Erklärung der für die Lehrerfortbildung „KI@Informatik 11 – was, wozu, wie, womit unterrichten“ der Didaktik der Informatik der Universität Passau zum Thema „Künstliche Intelligenz“ in der 11. Jahrgangsstufe entwickelten Software zur Simulation und didaktischen Visualisierung eines Perzeptrons.

Dr. Wolfgang Pfeffer

Dominicus-von-Linprun-Gymnasium Viechtach

E-Mail: schule@pfeffer-wolfgang.de

Tobias Fuchs

Universität Passau

E-Mail: fuchs_unipa@outlook.de

Weiteres Material findet sich im Mebis-Kurs „Material zu den KI-Fortbildungen der Universität Passau“ mit der `ID`-Nummer `1324361`. Den Einschreibeschlüssel zum Kurs erhalten Sie auf Anfrage.

Inhaltsverzeichnis

Überblick

Die Simulationssoftware *Perzeptron Simulator* ist im Rahmen der Fortbildungsinitiative KI für die Lehrerfortbildung „KI@Informatik 11 – was, wozu, wie, womit unterrichten“ der Didaktik der Informatik der Universität Passau als didaktische Software entwickelt worden. Der Simulator dient als Hilfsmittel für die Umsetzung des Themas *Perzeptron* des LehrplanPLUS-Kapitels 11.4 Künstliche Intelligenz des bayerischen Lehrplans.

Die Software bietet die Möglichkeit den Lernprozess eines Perzeptrons für zwei verschiedene Anwendungskontexte, die *Lineare Klassifikation* und die *Lineare Regression* zu simulieren. Neben einer schematischen Darstellung des Perzeptrons mit den jeweils aktuellen Parametern wird ein weiterer Fokus auf die graphische Darstellung der geometrischen Interpretation des Perzeptrons gelegt.
Für das Training des Perzeptrons können dabei mitgelieferte Trainingsdaten in Form von `.csv`-Dateien verwendet werden. Alternativ können auch eigene Trainingsdaten in den Simulator geladen werden.

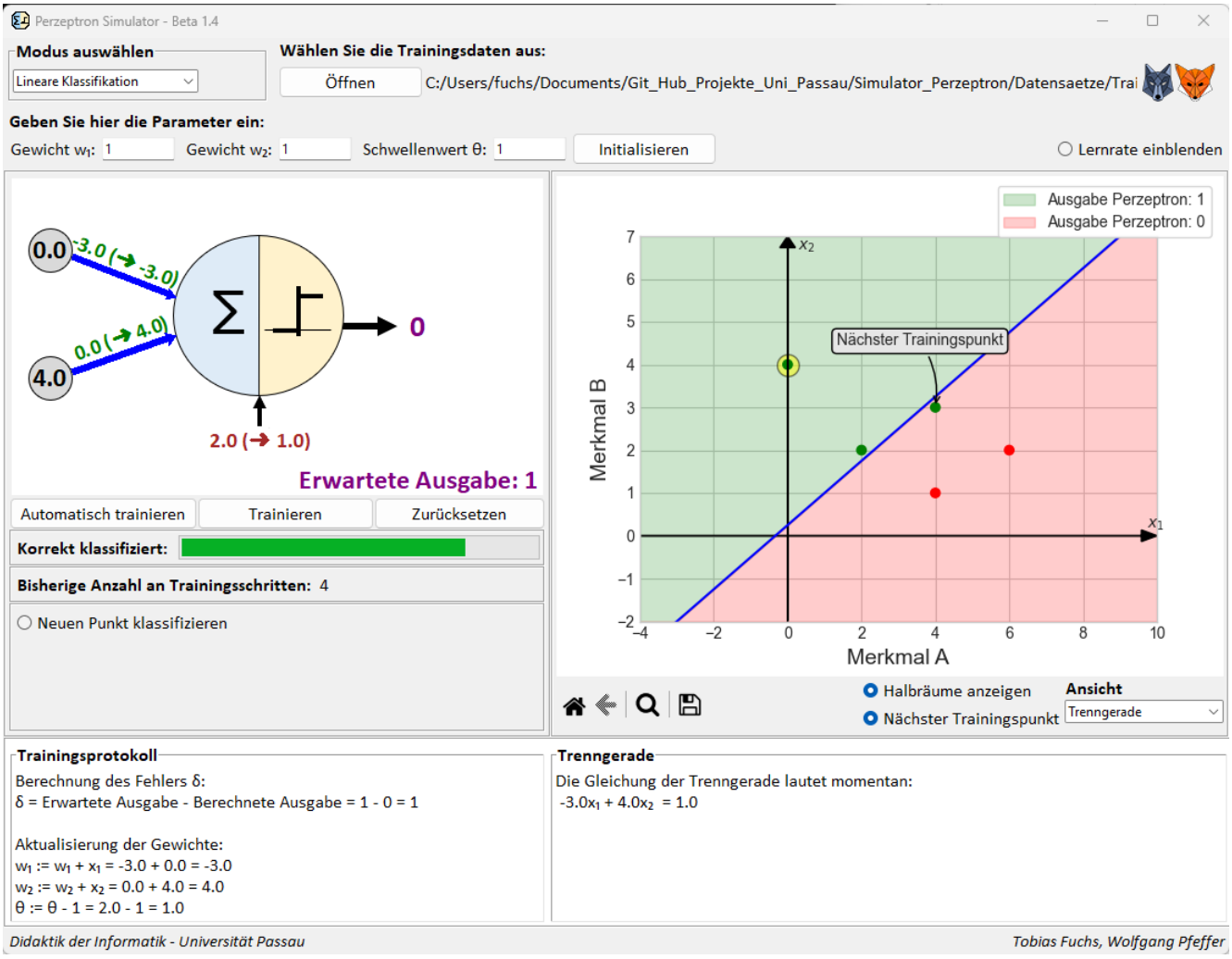


Abbildung 1: Trainingsprozess des *Perzeptron Simulator* im Modus *Lineare Klassifikation*

1. Verwendung des Simulators

Da die Simulationssoftware *Perzeptron Simulator* in Form von Python-Skripten zur Verfügung gestellt wird, ist diese unter Windows und MacOS/Linux (noch nicht optimiert) nutzbar. Damit Sie die Simulationssoftware *Perzeptron Simulator* ausführen können, muss auf Ihrem System lediglich **Python** mit den Paketen **Matplotlib** und NumPy (wird mit Matplotlib automatisch installiert) installiert sein.

Hinweis:

Es wird später auch kurz erläutert, wie Sie sich für Windows und MacOS eine ausführbare Datei des Simulators erstellen können, welche dann auf einem Rechner ohne vorherige Installation von Python und der oben genannten Pakete ausführbar ist.

1.1 Herstellen der Voraussetzungen: Python und das Paket Matplotlib

Installation von Python

Nutzen Sie für die Installation von Python am besten den auf der Python-Homepage zur Verfügung gestellten Installer. Dieser richtet, neben der Installation, Python auf Ihrem System ein.

1. Laden Sie sich den Python-Installer für Ihr System unter `https://www.python.org/downloads/` herunter.
2. Setzen Sie im ersten Fenster des Installers den Haken bei „*Add python.exe to PATH*“ (s. Abb. 2 Stelle 1.).
3. Klicken Sie dann auf „*Install Now*“ (s. Abb. 2 Stelle 2.). Dadurch wird Python installiert, die notwendigen Tools eingerichtet und Python für die Verwendung vorbereitet.

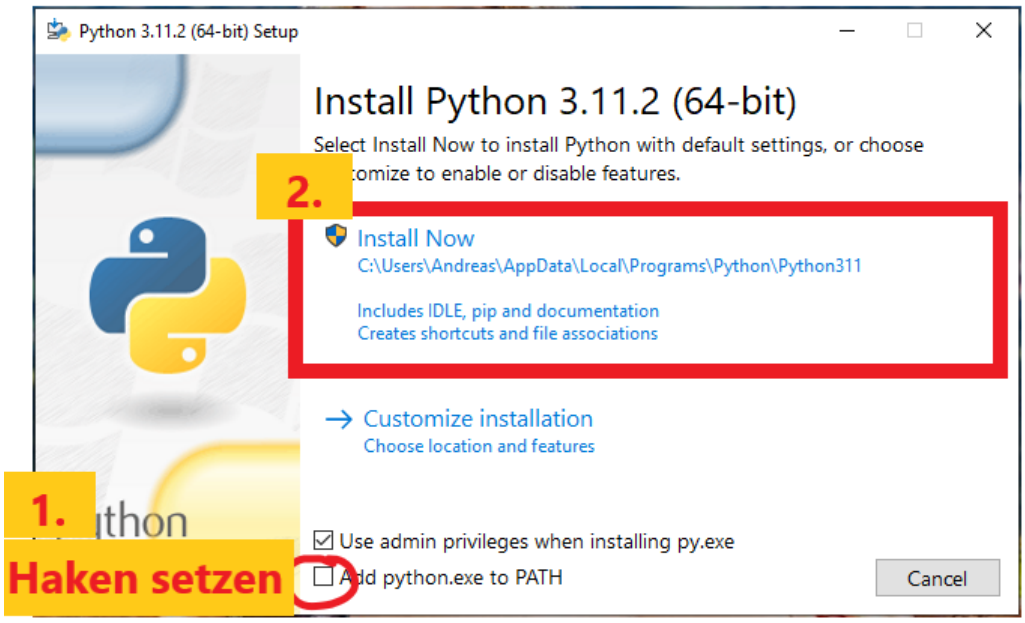


Abbildung 2: Screenshot des Python-Installers für Microsoft Windows

Installation von Paketen (Matplotlib)

Am einfachsten lassen sich Pakete für Python mit dem **Paket-Installer pip** installieren. Dieser wird bei Nutzung des Installers zur Installation von Python automatisch mitinstalliert.

1. Öffnen Sie die Kommandozeile/Eingabeaufforderung (Windows).

2. Geben Sie dort den Befehl

```
pip install matplotlib
```

ein und bestätigen Sie den Befehl mit der Eingabetaste. Das Paket Matplotlib (und ggf. für Matplotlib weitere benötigte Pakete) wird nun automatisch heruntergeladen und installiert.

Hinweis:

Durch die Installation von Matplotlib wird automatisch auch das Paket NumPy installiert, welches der Simulator zur Ausführung benötigt. Sollten Sie Matplotlib auf andere Art als via pip installieren, stellen Sie sicher, dass Sie auch das Paket NumPy installieren.

1.2 Ausführen des Perzeptron Simulators

Im Folgenden werden unterschiedliche Möglichkeiten aufgezeigt, wie der Simulator gestartet werden kann. Sie können darunter diejenige Variante wählen, welche Ihnen am liebsten ist und zu Ihrem System passt.

Hinweis:

Die aufgeführten Möglichkeiten stellen keine abschließende Liste an Startmöglichkeiten dar. Es gibt darüber hinaus noch viele weitere Möglichkeiten ein Python-Skript auszuführen. Wir haben uns in der folgenden Auflistung allerdings auf einzelne Möglichkeiten beschränkt, die für den Einsatz im Unterricht praktikabel und einfach umsetzbar sind.

1.2.1 Herunterladen der Software *Perzeptron Simulator*

(a) Laden Sie sich die aktuellste Version des *Perzeptron Simulators* unter `https://github.com/D3rFuch5/Simulator_Perzeptron` durch Klick auf *Code -> Download ZIP* herunter.
Alternativ finden Sie die ZIP-Datei auch unter `https://www.ddi.fim.uni-passau.de/ki-gym`

(b) Entpacken Sie die `.zip`-Datei. Der entpackte Ordner sollte folgende Struktur besitzen.

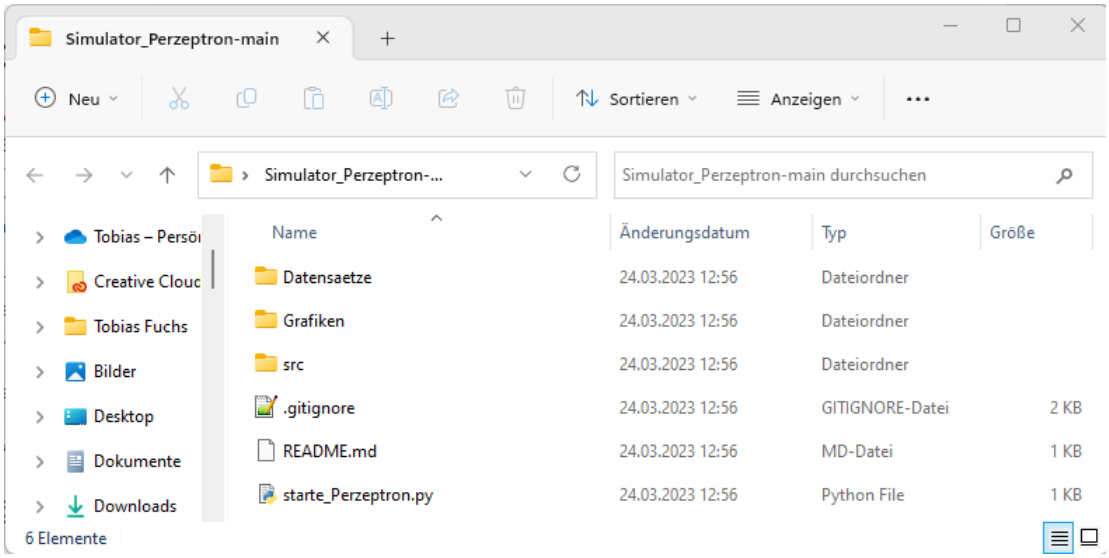
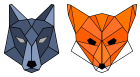


Abbildung 3: Screenshot des entpackten ZIP-Ordners des *Perzeptron Simulators*



1.2.2 Ausführen des *Perzeptron Simulators* (Windows):

Möglichkeit 1: Doppelklick auf `starte_Perzeptron.py`

Doppelklicken Sie auf die Datei `starte_Perzeptron.py`. Es öffnet sich die Oberfläche des *Perzeptron Simulators*.

Hinweis 1:

Durch Installation von Python mit dem Installer sollte Python als Standardprogramm für die Ausführung von `.py`-Dateien festgelegt sein.

Falls nicht, können Sie durch Rechtsklick auf die Datei `starte_Perzeptron.py` unter *Öffnen mit -> Python* Python für die Ausführung der Datei auswählen.

Hinweis 2:

Wollen Sie verhindern, dass sich neben der Oberfläche des *Perzeptron Simulators* auch eine Konsole öffnet, ändern Sie die Dateiendung der Datei `starte_Perzeptron.py` zu `starte_Perzeptron.pyw`.

Dies verhindert, dass beim Start, neben der Oberfläche des Simulators auch eine Konsole geöffnet wird.

Möglichkeit 2: Eingabe von `python starte_Perzeptron.py` in die Konsole

1. Öffnen Sie den entpackten Ordner.
2. Öffnen Sie eine Konsole für den **geöffneten** Ordner.
   Rechtsklicken Sie hierfür in den Ordner und wählen *In Terminal öffnen* aus. Es öffnet sich eine Konsole, welche bereits den Pfad des Ordners besitzt.
3. Geben Sie in der Konsole den Befehl
   ```
   python starte_Perzeptron.py
   ```
   ein und bestätigten Sie den Befehl mit der Eingabetaste. Es öffnet sich die Oberfläche des *Perzeptron Simulators*.

Hinweis:

Bei älteren Windows-Versionen ist die Auswahl *Im Terminal öffnen* nicht verfügbar.

Klicken Sie in einem solchen Fall in die Pfadzeile des geöffneten Ordners (s. Abb. 4 Teil 1.).

Geben Sie in die Pfadzeile `cmd` ein und bestätigen Sie den Befehl mit der Eingabetaste (s. Abb. 4 Teil 2.).

Es öffnet sich nun auch eine Konsole mit dem Dateipfad des geöffneten Ordners (s. Abb. 4 Teil 3.).

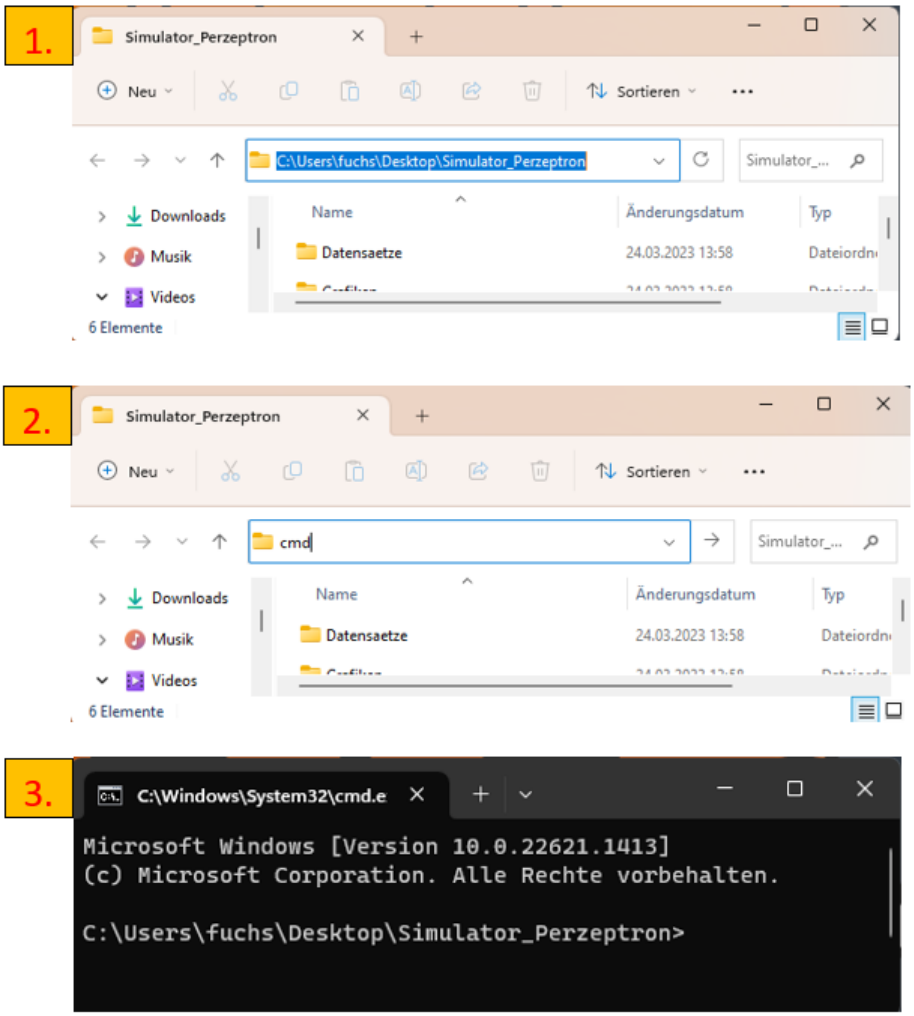


Abbildung 4: Öffnen einer Konsole für den aktuellen Ordner

Möglichkeit 3: Start mit IDLE

Die Abkürzung IDLE steht für *„Integrated Development and Learning Environment“* und bezeichnet eine Entwicklungsumgebung, welche bei einer normalen Python-Installation mit an Bord ist und nicht extra installiert werden muss.

1. Öffnen Sie IDLE aus den installierten Programmen.
2. Öffnen Sie die Datei `starte_Perzeptron.py` über *„File -> Open“*.
3. Drücken Sie `F5` um den Simulator zu starten.
   Alternativ klicken Sie im Fenster der geöffneten Datei auf *„Run -> Run Module“*.

Möglichkeit 4: Erstellen einer ausführbaren `.exe`-Datei mit dem Paket PyInstaller

Mit Hilfe des Pakets *PyInstaller* (`https://pyinstaller.org/en/v4.2/index.html`) ist es möglich aus Python-Skripten ausführbare Dateien für Windows und MacOS zu erstellen. Eine solche ausführbare Datei kann auf einem beliebigen Rechner des jeweiligen Systems ausgeführt werden, ohne dass dafür eine Installation von Python oder zusätzlicher Pakete notwendig ist, da diese in der ausführbaren Datei „gebündelt“ werden. Lediglich auf dem Rechner, auf dem die ausführbare Datei erstellt wird, muss Python mit den notwendigen Paketen installiert sein.

- *Installation des PyInstallers*:
  Öffnen Sie eine Konsole und geben dort den Befehl

  ```
  pip install pyinstaller
  ```

  ein und bestätigen Sie den Befehl mit der Eingabetaste. Das Paket Pyinstaller wird nun automatisch heruntergeladen und installiert.

- *Erstellen der ausführbaren `.exe`-Datei*:

  1. Öffnen Sie die Konsole im Ordner des Perzeptron-Simulators, d.h. in dem sich die Datei `starte_Perzeptron.py` befindet.
  2. Geben Sie den Befehl `pyinstaller --noconfirm --windowed --onefile --name Simulator_Perzeptron --icon Grafiken/Icon_simple.ico starte_Perzeptron.py` ein und bestätigen Sie den Befehl mit der Eingabetaste.
     Die ausführbare Datei wird nun erstellt. Das kann einige Minuten dauern.
  3. Im Ordner des Perzeptron-Simulators wurde automatisch ein Ordner `dist` erstellt, welcher die ausführbare Datei enthält.
     Kopieren Sie den Ordner *Grafiken* in den erstellten `dist`-Ordner.
  4. Doppelklick auf die `.exe`-Datei startet nun den *Perzeptron Simulator*.
  5. Wenn Sie die ausführbare Datei auf einem anderen Rechner nutzen wollen, kopieren Sie den `dist`-Ordner (inkl. des Ordners Grafiken) auf den anderen Rechner.

Hinweis:
Die Lizenzen der ggf. gebündelten Dateien sind zu beachten.

1.2.3 Ausführen des *Perzeptron Simulators* (MacOS):

Unter MacOS kann es, je nach verwendetem System, vereinzelt zu Darstellungsfehlern (u.a. Abschneiden eines Teils der Oberfläche) kommen. Der Fehler ist bekannt und es wird an einer zeitnahen Lösung gearbeitet. Sobald der Fehler behoben ist, wird eine neue Version des Simulators zur Verfügung gestellt und dieser Hinweis entfernt.

Möglichkeit 1: Verwenden des Python Launchers
Mit Python 3 wurde der *Python-Launcher*, ein Programm zum Ausführen von Python-Skripten über den Finder hinzugefügt.

1. Suchen Sie dazu die Python-Skriptdatei `starte_Perzeptron.py` im Finder.
2. Klicken Sie anschließend mit der rechten Maustaste auf die Datei und wählen Sie im Kontextmenü *„Öffnen mit -> Python Launcher“*.

Alternativ können Sie das Python-Skript `starte_Perzeptron.py` auch auf das Python Launcher-Symbol ziehen, sei es im Dock oder im Anwendungsordner.

Möglichkeit 2: Verwenden des Terminals

1. Öffnen Sie ein Terminal **für den Ordner**, welcher die Datei `starte_Perzeptron.py` enthält.
2. Geben Sie im Terminal den Befehl

   `python3 starte_Perzeptron.py`

   ein und bestätigten Sie den Befehl mit der Eingabetaste. Es öffnet sich die Oberfläche des *Perzeptron Simulators*.

Hinweis: Wie öffnet man ein Terminal im gewünschten Ordner aus dem Finder?

1. Öffnen Sie ein Finder-Fenster und navigieren Sie dann zu dem Ordner, den Sie verwenden möchten.
2. Wenn Sie die Pfadleiste nicht am unteren Rand des Finder-Fensters sehen, wählen Sie *„Darstellung“* -> *„Pfadleiste einblenden“*.
3. Klicken Sie bei gedrückter Taste „*ctrl*“ auf den Ordner in der Pfadleiste und wählen Sie dann „*In Terminal öffnen*“.

Möglichkeit 3: Erstellen einer `.app`-Datei mit dem Paket PyInstaller

Die Einrichtung des *PyInstallers* und die Erstellung einer ausführbaren `.app`-Datei erfolgt analog zu der Beschreibung der Erstellung einer `.exe`-Datei für Windows.
Es muss lediglich eine andere Icon-Datei im Format `.icns` verwendet werden. Alternativ kann auch auf ein Icon verzichtet werden. Dafür muss einfach der Befehlsteil `--icon Grafiken/Icon_simple.ico` aus dem obigen Befehl weggelassen werden.

Hinweis:
Die Lizenzen der ggf. gebündelten Dateien sind zu beachten.

2. Die Simulationssoftware

Die Simulationssoftware *Perzeptron Simulator* ermöglicht es ein Perzeptron für die Anwendungen *Lineare Klassifikation (2 Eingaben)* und *Lineare Regression(1 Eingabe)* zu simulieren. Für dieses beiden Anwendungen stehen zwei **Modi** in der Software zur Verfügung.

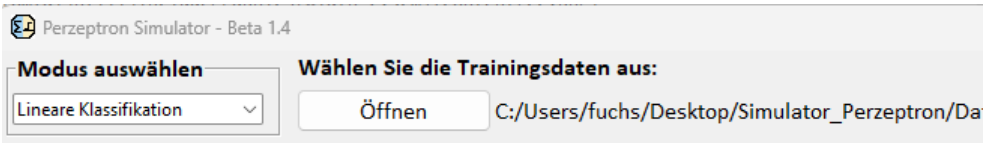


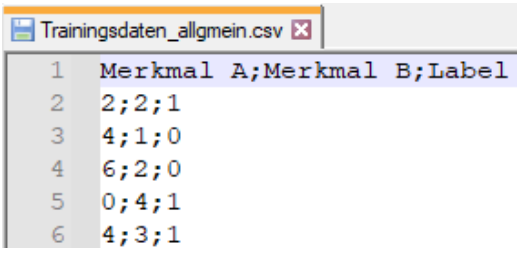
Abbildung 5: Auswahlmöglichkeit des Modus und der Trainingsdaten im *Perzeptron Simulator*

Modus wählen:
Den gewünschten Modus können Sie im linken, oberen Bereich der Software über das Drop-Down-Menü auswählen (Abb. 5 links). Standardmäßig ist der Modus *Lineare Klassifikation* ausgewählt.

Hinweis:
Ein Wechsel des Modus setzt den Simulator, d.h. u.a. die geladenen Trainingsdaten, einen ggf. bereits durchgeführten Lernprozess, ... zurück.

Auswahl der Trainingsdaten:
Durch Klick auf den Button *Öffnen* im Bereich *Wählen Sie die Trainingsdaten aus:* (Abb. 5 rechts) öffnet sich ein Kontextmenü über das Sie ihre Trainingsdaten auswählen können. Entsprechen die Daten dem notwendigen Format, werden diese automatisch geladen und im Koordinatensystem angezeigt.

Die Trainingsdaten müssen in Form einer `.csv`-Datei vorliegen. In der ersten Zeile stehen die Benennungen der betrachteten Merkmale/2 Eingaben(hier: *Merkmal A* und *Merkmal B*). Diese werden ausgelesen und dann im Koordinatensystem als Achsenbezeichnung angezeigt. Sie können somit passende Trainingsdaten eines beliebigen Kontexts (z.B. Gefährlichkeit von Tieren anhand von zwei Merkmalen; siehe mitgelieferter Beispieldatensatz) anzeigen. In den folgenden Zeilen stehen nun die Werte der einzelnen Trainingsdatenpunkte entsprechend der festgelegten Reihenfolge der Merkmale. Der letzte Eintrag jeder Zeile stellt dabei immer **0** oder **1** dar, je nachdem welcher der beiden Klassen der Trainingsdatenpunkt angehört.



Beachten Sie bitte, dass die einzelnen Einträge durch ein **;***(„Strichpunkt“)* voneinander getrennt sein müssen und bei der Eingabe von Dezimalwerten ein **.***(„Punkt“)* statt einem Komma verwendet werden muss. Für die lineare Regression wird nur eine Eingabe benötigt. Ein einzelner Trainingsdatenpunkt besteht also nur aus zwei Zahlen, der Eingabe und der zugehörigen Ausgabe.

Hinweis:
Sowohl für die lineare Klassifikation, als auch für die lineare Regression werden im Ordner *Datensaetze* bereits einzelne Beispieldatensätze mitgeliefert. Diese können Sie direkt verwenden oder sich an diesen bei der Erstellung

eigener Datensätze orientieren.

2.1 Modus: Lineare Klassifikation

Eingabe der initialen Gewichte w_1, w_2/des initialen Schwellenwerts θ des Perzeptrons:

Im Bereich *Geben Sie hier die Parameter ein:* können Sie die Initialwerte für die Gewichte w_1 und w_2 sowie für den Schwellenwert θ eingeben. Momentan werden hier nur ganzzahlige Eingaben akzeptiert.

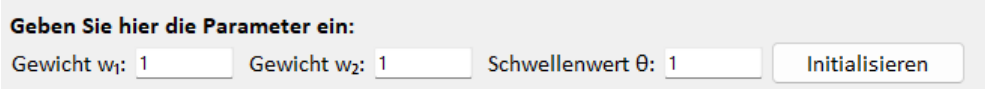


Abbildung 6: Eingabemöglichkeit der initialen Parameterwerte für w_1, w_2, θ

Nach Klick auf den Button *Initialisieren* werden die eingegebenen Werte geprüft und bei zulässigen Eingaben wird das Perzeptron mit diesen Werten initialisiert.

Links werden in der schematischen Darstellung des Perzeptrons die eingegeben Werte für die Gewichte w_1, w_2 und den Schwellenwert θ an den entsprechenden Pfeilen eingefügt (s. Abb. 7 Bereich 1.).

Rechts wird im Koordinatensystem die durch die Gewichte und den Schwellenwert festgelegte Trenngerade als blaue Gerade eingezeichnet (s. Abb. 7 Bereich 2.).

Des Weiteren wird die Gleichung der Trenngerade im rechten unteren Bereich *Trenngerade* angezeigt.

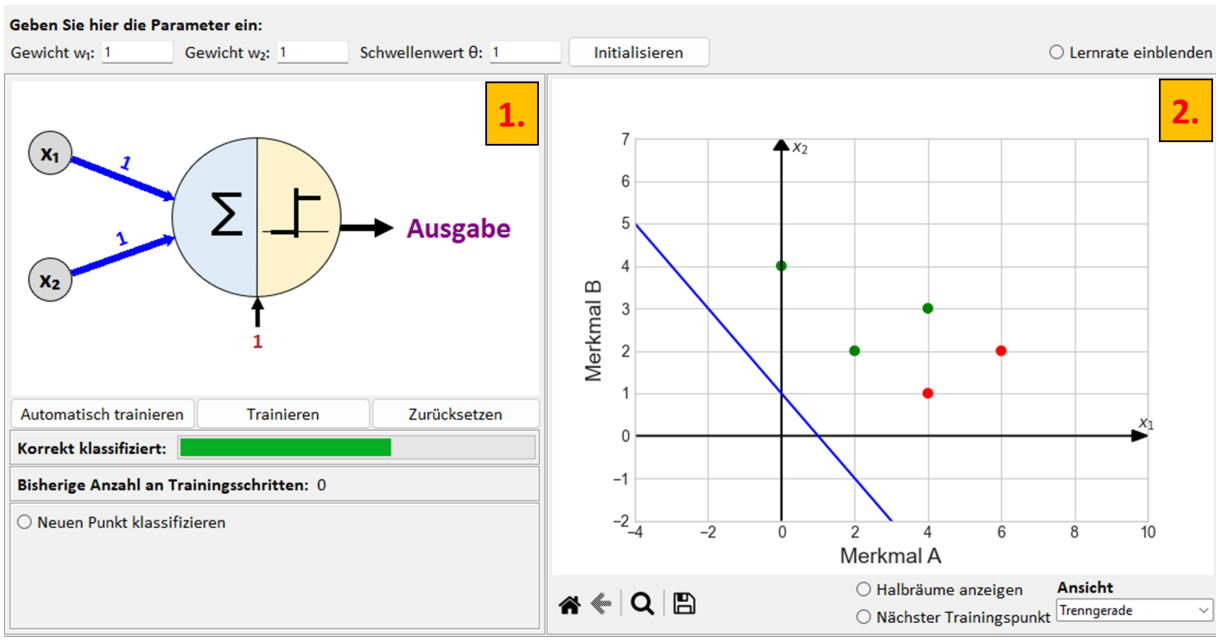


Abbildung 7: Anzeige des initialisierten Perzeptrons

Verwendung der Lernrate α:

Sie können das Perzeptron zusätzlich mit einer Lernrate α initialisieren. Aktivieren Sie dafür die Auswahl *Lernrate einblenden* (s. Abb. 8 oben) im rechten, oberen Bereich des Simulators.

Nun wird die Eingabemöglichkeit für die Lernrate eingeblendet (s. Abb. 8 unten). Hier können Sie nun einen Wert für die Lernrate eingeben.

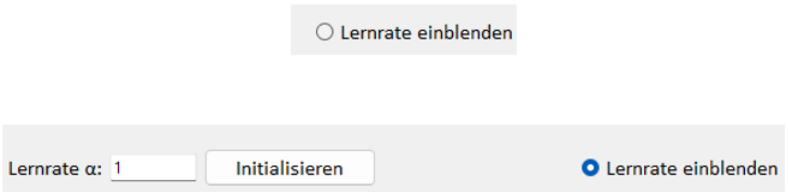


Abbildung 8: Verwendung der Lernrate α

Beachten Sie, dass bei Eingabe einer Dezimalzahl ein **.**(*Punkt*) als Trennzeichen verwendet werden muss.
Die Lernrate wird erst bei (Neu-) Initialisierung, d.h. Klicken des Buttons *Initialisieren* aktiv.

Trainieren des Perzeptrons:
Hauptzweck des *Perzeptron Simulators* ist die Simulation des Lernprozesses des Perzeptrons. Voraussetzung hierfür ist, dass bereits Trainingsdaten geladen sind und das Perzeptron initialisiert ist.

- Button *Trainieren*:
  Durch Klick auf den Button *Trainieren* wird ein einzelner Trainingsschritt durchgeführt.
  In der schematischen Darstellung des Perzeptrons links werden die Werte des aktuellen Trainingsdatenpunkts als Eingaben (graue Kreise) dargestellt.
  Rechts unten wird das Label des Trainingsdatenpunkts als *Erwartete Ausgabe* angezeigt(s. Abb 9).
  Rechts neben dem Neuron wird die vom Perzeptron *berechnete Ausgabe* angezeigt. Über den Pfeilen der Gewichte und des Schwellenwerts werden deren Werte **vor** und **nach** dem Lernschritt in der Form $Wert_{vorher}$ *(→* $Wert_{nachher}$*)* dargestellt.

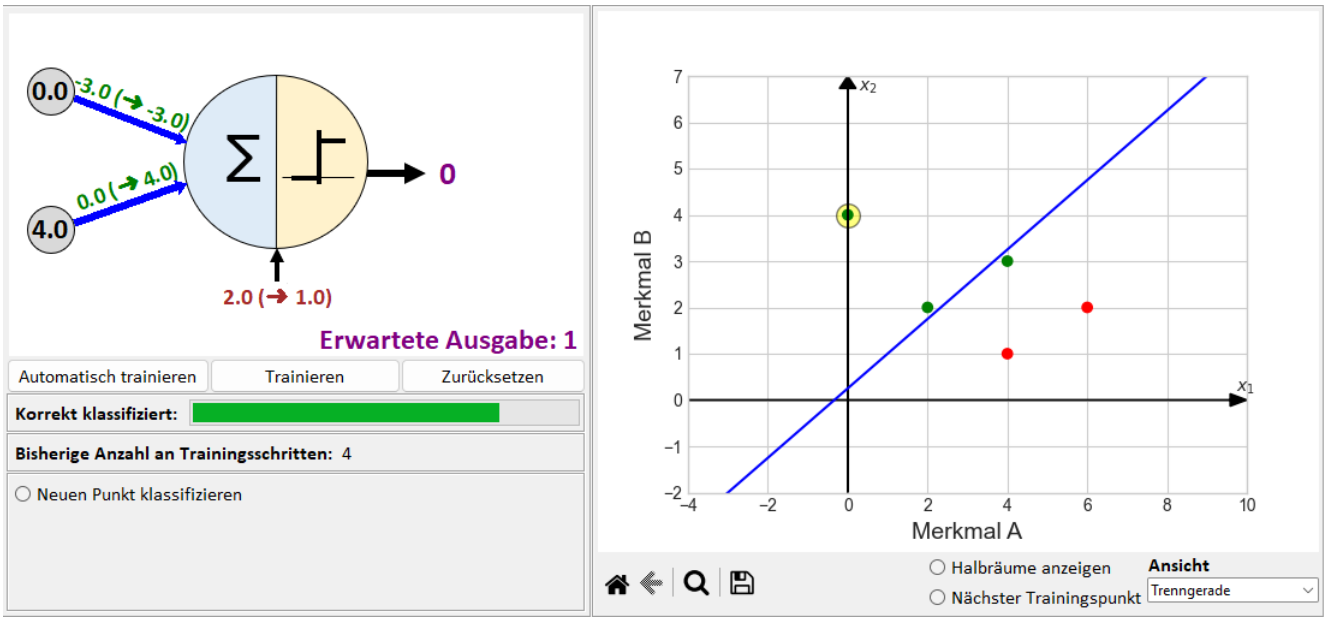


Abbildung 9: Trainingsschritt des Perzeptrons

In Abb. 9 hat gilt für den Schwellenwert 2.0 *(→ 1.0)*, d.h. vor dem Trainingsschritt hatte der Schwellenwert den Wert 2.0 und dieser wurde durch den Lernalgorithmus in diesem Schritt zu 1.0 angepasst.
Die Pfeildicken der Gewichte passen sich im Verhältnis ihrer zugehörigen Gewichtswerte zueinander an.

- Button *Zurücksetzen*:
  Durch Klick auf den Button *Zurücksetzen* wird der Lernprozess auf den Initialzustand zurückgesetzt, d.h. das Perzeptron besitzt nun wieder die von Ihnen eingegebenen Initialwerte für die Gewichte, den Schwellenwert, ...

- Button *Automatisch trainieren*:
  Durch Klick auf den Button *Automatisch trainieren* wird das automatische Training gestartet. Dabei wird das Perzeptron schrittweise mit den Trainingsdaten trainiert. Dies geschieht solange bis der Button *Training stoppen* geklickt wird. Die einzelnen Trainingsschritte des automatischen Trainings unterscheiden sich nicht von den Trainingsschritten, welche durch fortwährendes Klicken des Buttons *Trainieren* durchgeführt werden würden.
  Nach Aktivierung des automatischen Trainings besteht die Möglichkeit den Trainingsprozess zu beschleunigen oder zu verlangsamen. Zu Beginn findet zwischen den Trainingsschritten eine einsekündige Pause statt. Durch Klick auf *Schneller* wird die Pause halbiert, durch Klick auch *Langsamer* verdoppelt. Um ein reibungsloses automatisches Training sicherzustellen, wird eine Geschwindigkeitsänderung unterbunden, wenn der Lernprozess zu schnell oder zu langsam werden würde.

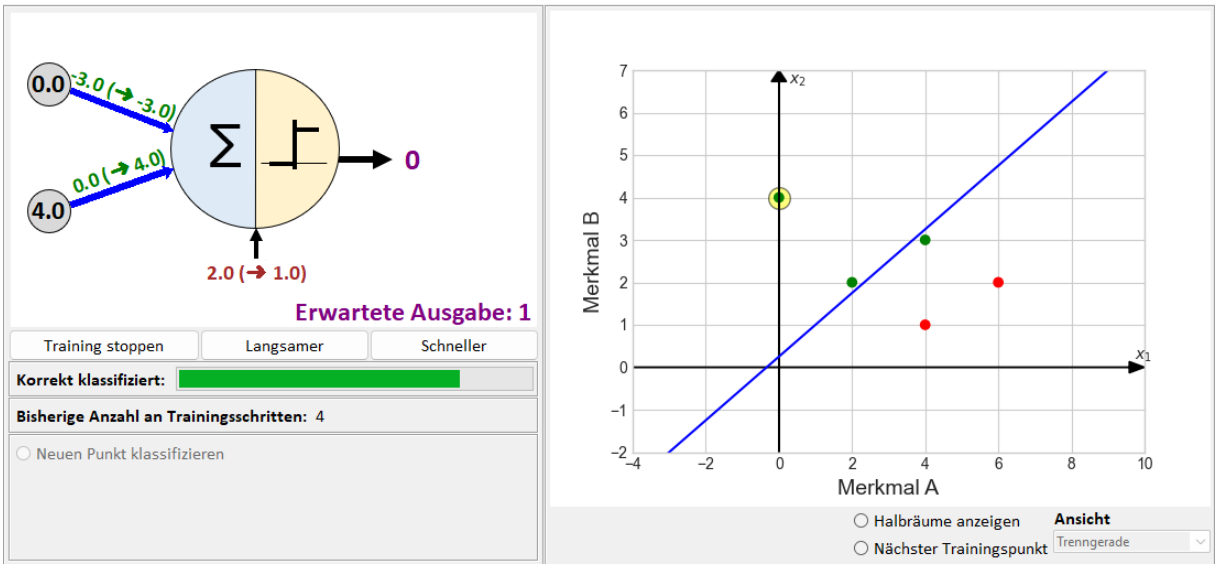


Abbildung 10: Oberfläche des Simulators bei aktiviertem automatischen Training

Unterhalb der schematischen Darstellung des Perzeptrons wird in einem Fortschrittsbalken immer angezeigt, wie viele Trainingsdaten mit dem aktuellen Perzeptron momentan korrekt klassifiziert werden. Des Weiteren wird die Anzahl der bisherigen Trainingsschritte angezeigt.

Im *Trainingsprotokoll* (Simulator unten) wird der aktuelle Trainingsschritt und die darin ggf. durchgeführten Anpassungen der Gewichte bzw. des Schwellenwerts genauer erläutert.
Im Bereich *Trenngerade* wird die Gleichung der aktuellen, durch das Perzeptron festgelegten Trenngerade angezeigt.

Im Koordinatensystem rechts wird immer die aktuelle geometrische Interpretation des Perzeptrons, d.h. die Trenngerade, der aktuell trainierte Datenpunkt(gelb hinterlegt),... dargestellt. Über die Auswahlbuttons *Halbräume anzeigen* und *Nächster Trainingsdatenpunkt* können weitere Informationen eingeblendet werden.

- *Halbräume anzeigen*:
  Durch Auswahl der Option *Halbräume anzeigen* werden die Halbräume bezüglich der Trenngeraden angezeigt, in denen alle Punkte liegen, für die das Perzeptron aktuell die Ausgabe 1 (grün) bzw. 0 (rot) liefern würde (s. Abb. 11).

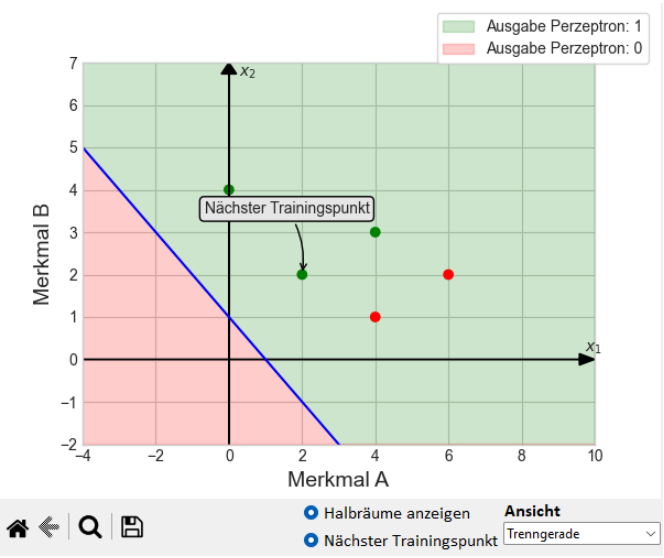


Abbildung 11: Anzeige der Halbräume und des nächsten Trainingsdatenpunkts

- *Nächster Trainingsdatenpunkt*:
  Durch Auswahl der Option *Nächster Trainingsdatenpunkt* wird der Datenpunkt mit einer Annotation versehen, welcher für den nächsten Trainingsschritt verwendet wird (s. Abb. 11).

Klassifikation eines neuen Datenpunkts mit dem momentanen Stand des Perzeptrons:
Der Simulator bietet die Möglichkeit einen neuen Datenpunkt mit dem momentanen Perzeptron zu klassifizieren. Dies kann u.a. in der Testphase des maschinellen Lernprozesses nützlich sein, da man somit die Testdatenpunkte der Reihe nach eingeben und vom Perzeptron klassifizieren lassen kann. Damit ist es möglich Aussagen über die Güte, also die Vorhersage-Qualität des momentanen (ggf. bereits trainierten) Perzeptrons zu machen.

Durch Klick auf den Auswahlbutton *Neuen Punkt klassifizieren* wird die Klassifikationsoberfläche eingeblendet. Alle anderen Eingabemöglichkeiten werden für die Dauer der Klassifikation gesperrt, da man den Punkt mit dem momentanen Stand des Perzeptrons klassifizieren will. Nach Deaktivierung der Klassifikation kann das Training bei Bedarf fortgesetzt werden.

Im Folgenden ist ein Klassifikationsduchlauf dargestellt:

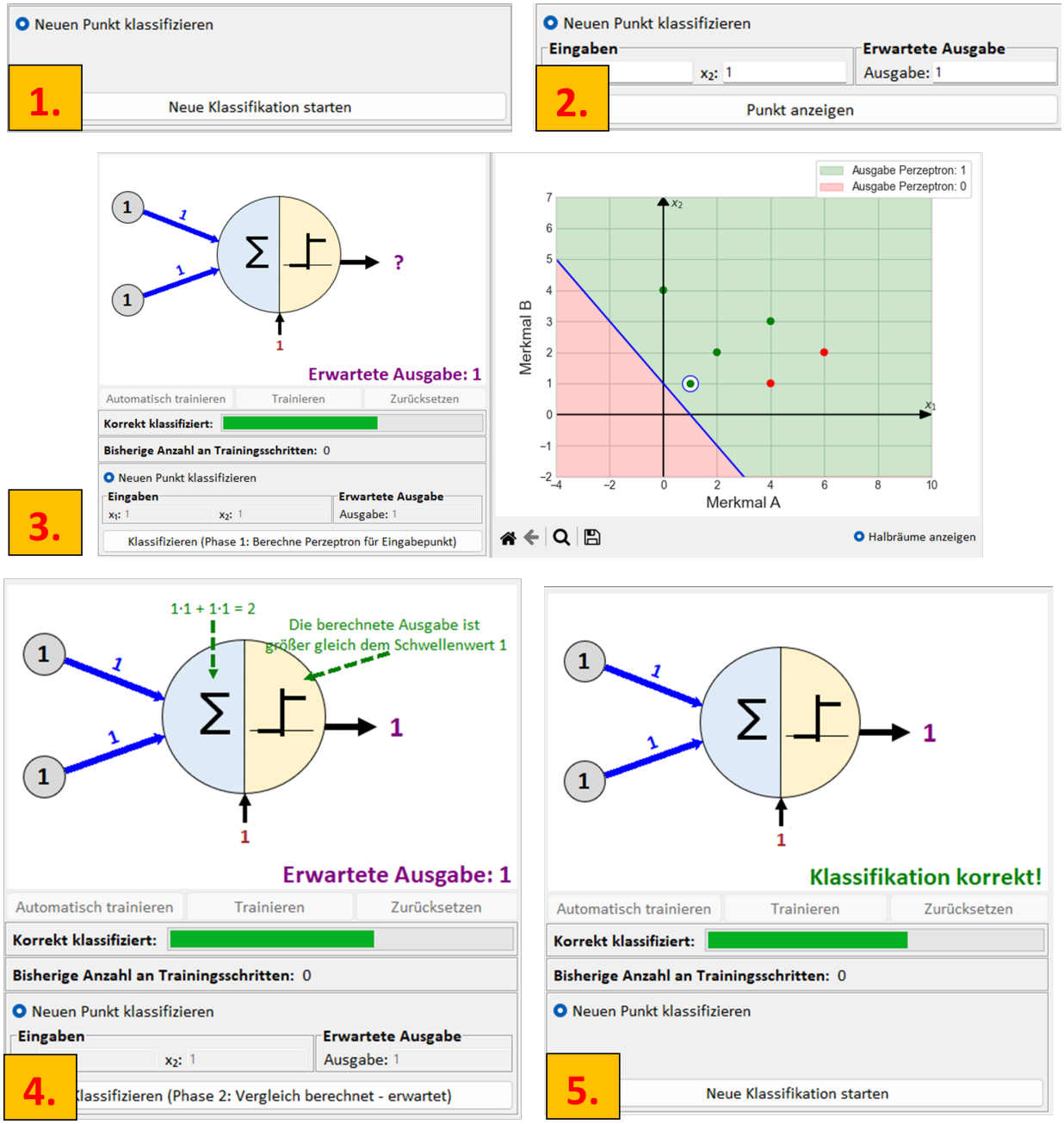


Abbildung 12: Darstellung des Ablaufs der Klassifikation eines neuen Datenpunkts

1. Durch Klick auf den Button *Neue Klassifikation starten* wird der Klassifikationsdurchlauf gestartet.

2. Im nächsten Schritt muss der zu klassifizierende Datenpunkt (Eingaben x_1, x_2) und die für den Punkt erwartete Klasse (*Ausgabe*) eingegeben werden.

3. Durch Klick auf *Punkt anzeigen* wird der Punkt im Koordinatensystem angezeigt.

4. In der schematischen Darstellung des Perzeptrons werden die Berechnungen des aktuellen Perzeptrons für den Datenpunkt angezeigt und die berechnete Ausgabe wird eingeblendet.

5. Die berechnete Ausgabe wird mit der erwarteten Ausgabe verglichen.
   Stimmen sie überein, wird *Klassifikation korrekt!* angezeigt. Andernfalls wird *Falsche Klassifikation!* angezeigt.

Ansichten:

- Ansicht *Trenngerade*:

  Diese Ansicht ist die Standardansicht, da sie die geometrische Interpretation des Perzeptrons darstellt.

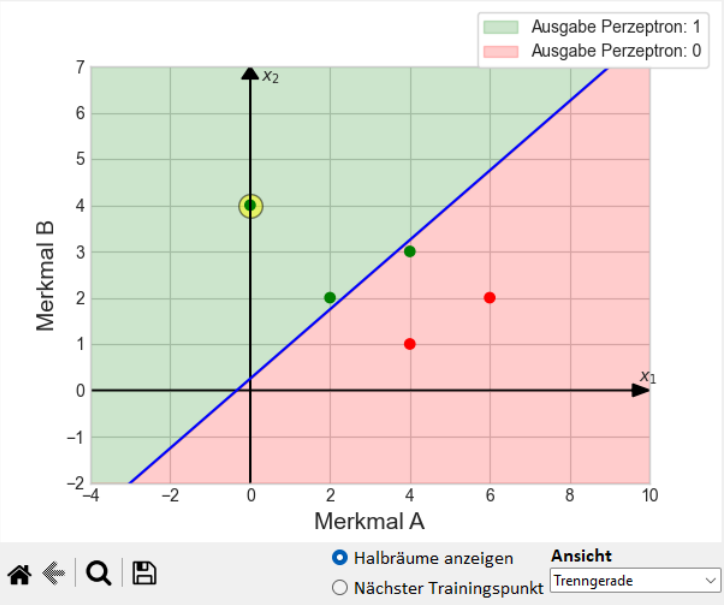


Abbildung 13: Ansicht Trenngerade des *Perzeptron Simulators* im Modus *Lineare Klassifikation*

- Ansicht *Gewichtsaktualisierung*:

  Die Ansicht *Gewichtsaktualisierung* zeigt die Entwicklung der Parameter des Perzeptrons während des Lernprozesses an. Eine Parameterkombination des Perzeptrons $(w_1|w_2|\theta)$ wird als dreidimensionaler Punkt(blau) angezeigt. Der grüne Halbraum symbolisiert den Raum, in dem alle Parameterkombinationen liegen, für die das Perzeptron den aktuell betrachteten Datenpunkt korrekt klassifizieren würde, der rote Halbraum die Parameterkombinationen, die zu einer falschen Klassifikation des aktuellen Datenpunkts führen würden.
  Mit dieser Ansicht kann visualisiert werden, dass die Gewichte immer so aktualisiert werden, dass man möglichst schnell zu einer korrekten Klassifikation des betrachteten Datenpunkts kommt (Stichwort **Gradientenabstieg**). Graphisch erkennt man dies daran, dass man sich durch eine Aktualisierung der Parameter(Gewichte/Schwellenwert) senkrecht auf die trennende Ebene zwischen falscher und richtiger Klassifikation zubewegt, sofern man vor der Aktualisierung im roten Bereich war.
  Diese Ansicht ist nur sinnvoll nutzbar, wenn nur ein einzelner Trainingsdatenpunkt(Trainingsdatensatz besteht nur aus einem Datenpunkt) trainiert wird, da sich ansonsten mit Wechsel des Trainingsdatenpunkts je Trainingsschritt auch die Halbräume ändern würden.
  In Abbildung 14 wurde der Trainingsdatenpunkt (2|3|1) mit den Initialparametern $w_1 = -4, w_2 = -4, \theta = 1$ trainiert. Man sieht, dass es zwei Trainingsschritte braucht um in den grünen Bereich zu kommen(dargestellt durch die Pfeile). Bei jeder Aktualisierung bewegt man sich auf schnellstem Wege(senkrecht) auf die trennende Ebene zu.

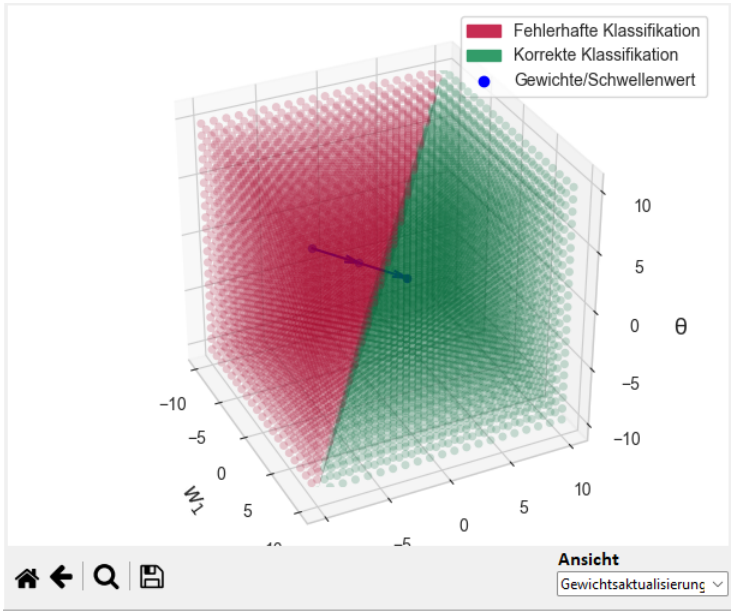


Abbildung 14: Ansicht Gewichtsaktualisierung des *Perzeptron Simulators* im Modus *Lineare Klassifikation*

Hinweis:

Der Anzeigebereich dieses 3D-Plots ist auf allen drei Achsen auf den Bereich $[-10; 10]$ beschränkt und fest. Eine dynamische Reskalierung bzw. Anpassung an die Parameterwerte ist in der aktuellen Version noch nicht implementiert.

2.2 Modus: Lineare Regression

Bei der linearen Regression soll eine Gerade so durch eine Punktmenge gelegt werden, dass diese möglichst gut zur Punktmenge passt, sprich dass der Gesamtabstand der Gerade zu den Punkten möglichst klein ist.

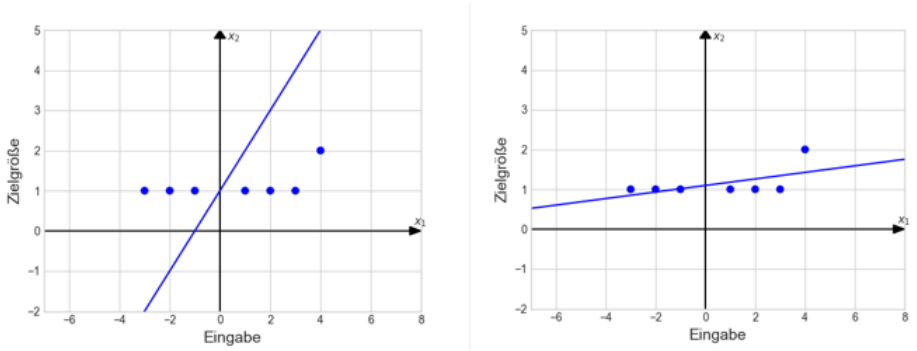


Abbildung 15: Darstellung der Idee der linearen Regression:(Links) Durch ein untrainiertes Perzeptron festgelegte Regressionsgerade; (Rechts) Durch ein trainiertes Perzeptron festgelegte Regressionsgerade

Eingabe der Initialparameter w_1, w_0 des Perzeptrons:

Im Bereich *Geben Sie hier die Parameter ein:* können Sie die Initialwerte für die Parameter w_1 und w_0 eingeben. Momentan werden hier nur ganzzahlige Eingaben akzeptiert.

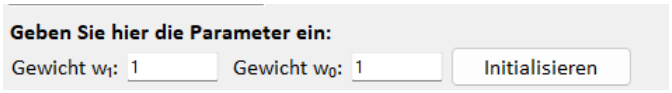


Abbildung 16: Eingabemöglichkeit der initialen Parameterwerte für w_1, w_0

Nach Klick auf den Button *Initialisieren* werden die eingegebenen Werte geprüft und bei zulässigen Eingaben wird das Perzeptron mit diesen Werten initialisiert.

Links werden in der schematischen Darstellung des Perzeptrons die eingegeben Werte für die Parameter w_1 und w_0 an den entsprechenden Pfeilen eingefügt (s. Abb. 17 Punkt 1.).

Rechts wird im Koordinatensystem die durch die Parameter festgelegte Regressionsgerade ($y = w_1 \cdot x_1 + w_0$) als blaue Gerade eingezeichnet (s. Abb. 17 Punkt 2.).

Des Weiteren wird die Gleichung der Regressionsgerade im unteren Bereich des Simulators angezeigt.

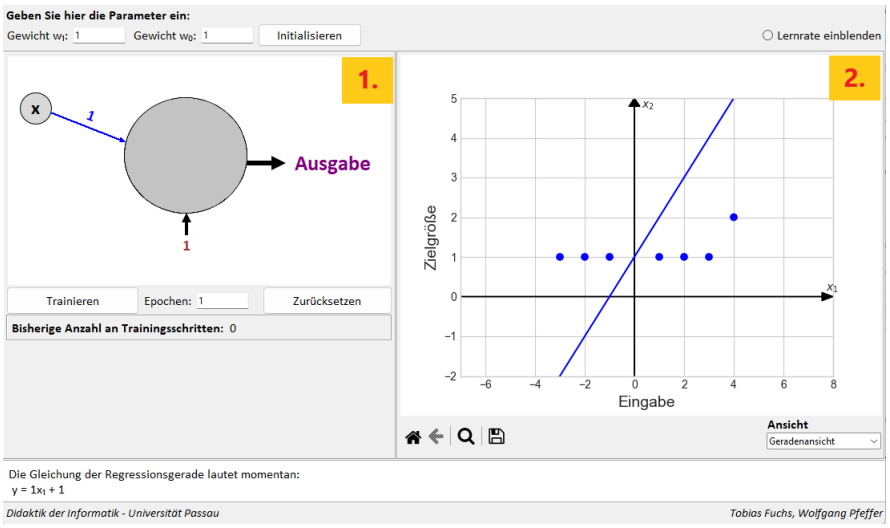


Abbildung 17: Anzeige des initialisierten Perzeptrons

Verwendung der Lernrate α:

Sie können das Perzeptron zusätzlich mit einer Lernrate α initialisieren. Aktivieren Sie dafür die Auswahl *Lernrate einblenden* (s. Abb. 18 oben) im rechten, oberen Bereich des Simulators.

Nun wird die Eingabemöglichkeit für die Lernrate eingeblendet (s. Abb. 18 unten). Hier können Sie nun einen Wert für die Lernrate eingeben.

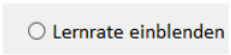


Abbildung 18: Verwendung der Lernrate α

Beachten Sie, dass bei Eingabe einer Dezimalzahl ein **.**(*Punkt*) als Trennzeichen verwendet werden muss.

Die Lernrate wird erst bei (Neu-) Initialisierung, d.h. Klicken des Buttons *Initialisieren* aktiv.

Trainieren des Perzeptrons:

Hauptzweck des *Perzeptron Simulators* ist die Simulation des Lernprozesses des Perzeptrons. Voraussetzung hierfür ist, dass bereits Trainingsdaten geladen sind und das Perzeptron initialisiert ist.

- Button *Trainieren*:
  Durch Klick auf den Button *Trainieren* wird ein Trainingsdurchlauf durchgeführt.
  Anders als im Modus *Lineare Klassifikation* wird in einem Trainingsschritt das Perzeptron nicht mit einem einzelnen Trainingsdatenpunkt trainiert, sondern hier werden für einen Trainingsschritt alle Trainingsdaten herangezogen. Die jeweils aktuellen Werte der Parameter werden an den entsprechenden Pfeilen in der schematischen Darstellung angezeigt(s. Abb. 19).

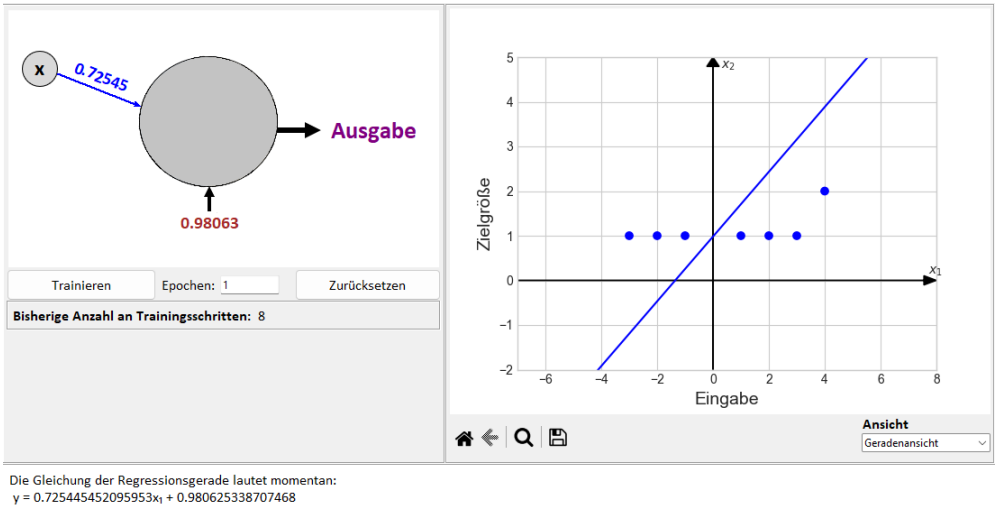


Abbildung 19: Training des Perzeptrons im Modus *Lineare Regression*

Die Anzahl der Trainingsschritte, welche in einem Trainingsdurchlauf durchgeführt werden, hängt von der festgelegten Epochenanzahl ab. Neben dem Feld *Epochen* kann diese Anzahl festgelegt werden. Die Eingabe muss dabei eine positive, ganze Zahl sein. Wird beispielsweise als Epochenanzahl 100 festgelegt, werden in einem Trainingsdurchlauf, d.h. durch einmaliges Klicken des Button *Trainieren*, 100 Trainingsschritte durchgeführt.

- Button *Zurücksetzen*:
  Durch Klick auf den Button *Zurücksetzen* wird der Lernprozess auf den Initialzustand zurückgesetzt, d.h. das Perzeptron besitzt nun wieder die von Ihnen eingegebenen Initialwerte für die Parameter.

Unterhalb der schematischen Darstellung des Perzeptrons wird die Anzahl der bisherigen Trainingsschritte angezeigt.
Im unteren Bereich des Simulators wird die Gleichung der momentanen, durch das Perzeptron festgelegten Regressionsgerade angezeigt.

Ansichten:

- Ansicht *Geradenansicht*:
  Diese Ansicht ist die Standardansicht, da sie die geometrische Interpretation der linearen Regression mit Hilfe des Perzeptrons darstellt.

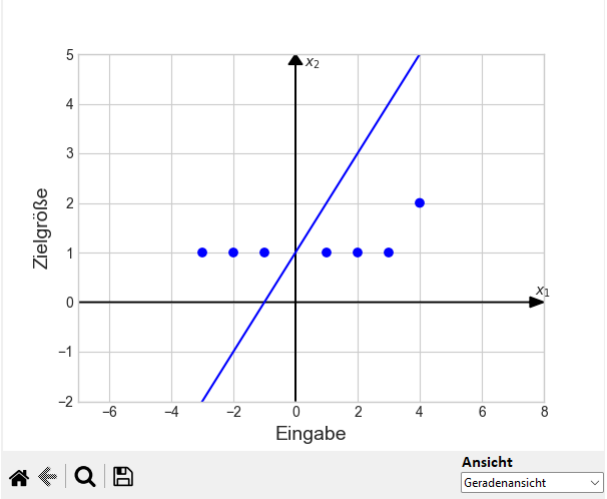


Abbildung 20: Ansicht *Geradenansicht* des *Perzeptron Simulators* im Modus *Lineare Regression*

- Ansicht *Gradientenabstieg*:
  Diese Ansicht dient dazu, das hinter dem Perzeptron-Lernalgorithmus stehende Verfahren des **Gradientenabstiegs** zu visualisieren. Da der Fokus des Handbuchs auf einer Anleitung zur Verwendung der Simulationssoftware *Perzeptron Simulator* liegt, wird die dafür benötigte Theorie an dieser Stelle nicht thematisiert.

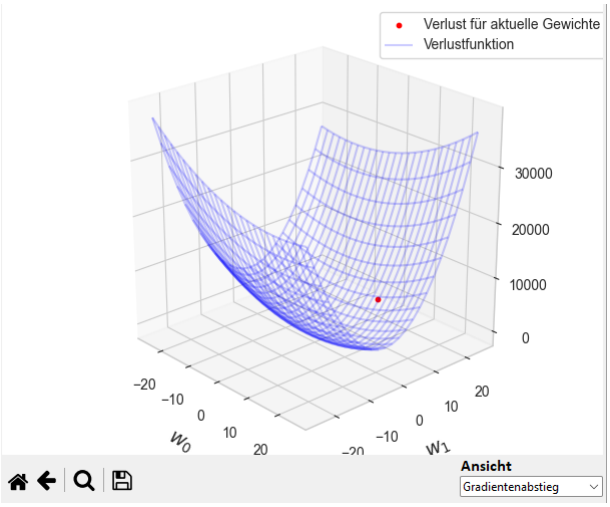


Abbildung 21: Ansicht *Gradientenabstieg* des *Perzeptron Simulators* im Modus *Lineare Regression*

Hinweis:

Der Anzeigebereich dieses 3D-Plots ist auf allen drei Achsen auf den Bereich $[-25; 25]$ beschränkt und fest. Eine dynamische Reskalierung ist in der aktuellen Version noch nicht implementiert.

3. Weitere Informationen

Die Simulationssoftware *Perzeptron Simulator* ist im Rahmen der Fortbildungsinitiative KI für die Lehrerfortbildung „KI@Informatik 11 – was, wozu, wie, womit unterrichten“ der Didaktik der Informatik der Universität Passau entwickelt worden. Der Simulator dient als ein Hilfsmittel für die Umsetzung des Themas *Perzeptron* des LehrplanPLUS-Kapitels 11.4 Künstliche Intelligenz des bayerischen Lehrplans.

Die aktuellste Version des *Perzeptron Simulators* sowie dieses Handbuchs finden Sie unter `https://github.com/D3rFuch5/Simulator_Perzeptron.git` sowie auf der Seite der Didaktik der Informatik der Universität Passau unter `https://www.ddi.fim.uni-passau.de/ki-gym`

Der Simulator wurde in der Programmiersprache Python entwickelt und verwendet neben der Python-Bibliothek(Version 3.11.3) die Pakete NumPy(Version 1.24) und Matplotlib(3.7.1), wobei das Paket NumPy bei der Installation von Matplotlib bereits mitinstalliert wird.

Die Simulationssoftware *Perzeptron Simulator* an sich und dieses zugehörige Handbuch sind unter der *APACHE LICENSE, VERSION 2.0* veröffentlicht. Die Python-Bibliothek sowie das Paket Matplotlib stehen unter PSF-Lizenz, das Paket NumPy unter BSD-Lizenz. Weitere von diesen Paketen verwendete Pakete/Bibliotheken können unter anderer Lizenz stehen und in diesen Fällen gelten diese Lizenzen.

- APACHE LICENSE, VERSION 2.0:
  Link: `https://www.apache.org/licenses/LICENSE-2.0`

  Apache License
  Version 2.0, January 2004
  http://www.apache.org/licenses/

  TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

  1. Definitions.
     “License“ shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

     “Licensor“ shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

     “Legal Entity“ shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, “control“ means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

     “You“ (or “Your “) shall mean an individual or Legal Entity exercising permissions granted by this License.

     “Source“ form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

     “Object“ form shall mean any form resulting from mechanical transformation or translation of a Source

form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, ßubmitted"means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.
   Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License.
   Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution.
   You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

(5.) Submission of Contributions.
Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

(6.) Trademarks.
This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

(7.) Disclaimer of Warranty.
Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

(8.) Limitation of Liability.
In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental,

or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

(8.) Limitation of Liability.
In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

(9.) Accepting Warranty or Additional Liability.
While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright yyyy name of copyright owner

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

- PSF LICENSE AGREEMENT FOR PYTHON 3.11.3 *(Stand: 07.04.2023)*:
  Link: https://docs.python.org/3/license.html#psf-license

1. This LICENSE AGREEMENT is between the Python Software Foundation (“PSF“), and the Individual or Organization (“Licensee“) accessing and otherwise using Python 3.11.3 software in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 3.11.3 alone or in any derivative version, provided, however, that PSF’s License Agreement and PSF’s notice of copyright, i.e., “Copyright © 2001-2023 Python Software Foundation; All Rights Reserved“ are retained in Python 3.11.3 alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python 3.11.3 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 3.11.3.
4. PSF is making Python 3.11.3 available to Licensee on an “AS IS“ basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 3.11.3 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 3.11.3 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 3.11.3, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.
8. By copying, installing or otherwise using Python 3.11.3, Licensee agrees to be bound by the terms and conditions of this License Agreement.

- License agreement for matplotlib versions 1.3.0 and later *(Stand: 07.04.2023)*:
  Link: https://matplotlib.org/stable/users/project/license.html

  License agreement for matplotlib versions 1.3.0 and later

  1. This LICENSE AGREEMENT is between the Matplotlib Development Team (“MDT“), and the Individual or Organization (“Licensee“) accessing and otherwise using matplotlib software in source or binary form and its associated documentation.
  2. Subject to the terms and conditions of this License Agreement, MDT hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use matplotlib alone or in any derivative

version, provided, however, that MDT's License Agreement and MDT's notice of copyright, i.e., "Copyright (c) 2012- Matplotlib Development Team; All Rights Reserved" are retained in matplotlib alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates matplotlib or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to matplotlib .
4. MDT is making matplotlib available to Licensee on an "AS IS" basis. MDT MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, MDT MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF MATPLOTLIB WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. MDT SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF MATPLOTLIB FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING MATPLOTLIB , OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between MDT and Licensee. This License Agreement does not grant permission to use MDT trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.
8. By copying, installing or otherwise using matplotlib, Licensee agrees to be bound by the terms and conditions of this License Agreement.

- NumPy License *(Stand: 07.04.2023)*:
  Link: https://numpy.org/doc/stable/license.html

  Copyright (c) 2005-2022, NumPy Developers.
  All rights reserved.

  Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
  * Neither the name of the NumPy Developers nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE