

KI und Programmierung

Künstliche Neuronale Netze mit dem Open Roberta Lab

Thorsten Leimbach, Geschäftsfeld Smart Coding and Learning

Version: 3.0



Neuronale Netze

Erläuterung

Diese Folien sollen einen Überblick über die Grundfunktionalitäten der Integration von Künstliche Neuronale Netzen im Open Roberta Lab geben. Es wird erläutert, wie ein Künstliche Neuronale Netz im Open Roberta Lab selbst erstellt und in einem eigenen NEPO (Open Roberta) Programm verwendet werden kann.

Diese Folien sollen dazu dienen, die grundlegende Handhabung Neuronaler Netze mit Open Roberta zu verstehen.

Übersicht

1. Einführung Perzeptron / Künstliches Neuronales Netz und Open Roberta xNN
2. Programmieren mit dem Open Roberta Lab
3. Beispiele
4. Ausblick

Kapitel 01

Einführung Künstliche Intelligenz

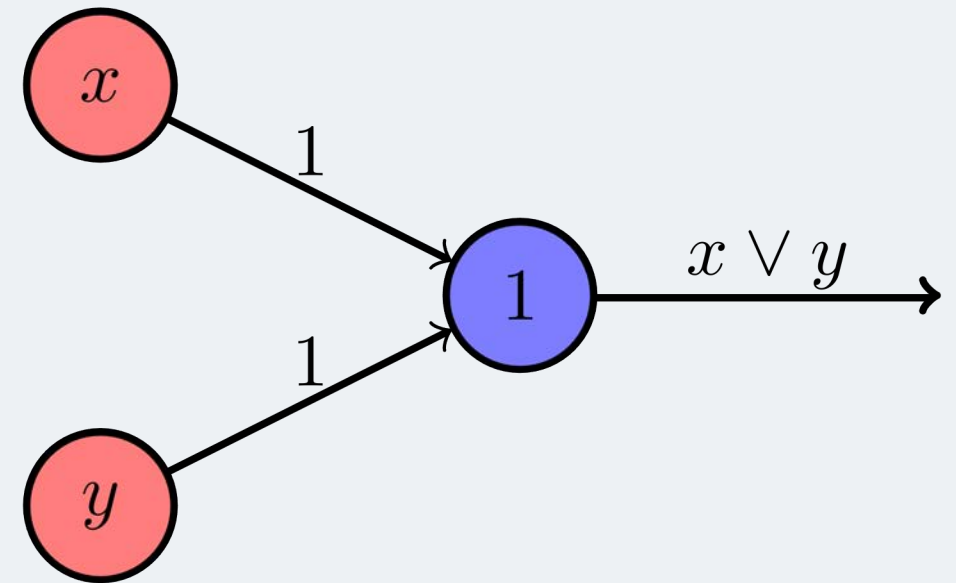
Neuronale Netze

Arten Künstlicher Neuronaler Netze

Perzeptron

Das **Perzeptron** (nach engl. perception, „Wahrnehmung“) ist ein vereinfachtes künstliches neuronales Netz, das zuerst von Frank Rosenblatt 1957 vorgestellt wurde. Es besteht in der Grundversion (einfaches Perzeptron) aus einem einzelnen künstlichen Neuron mit anpassbaren Gewichtungen und einem Schwellenwert. Unter diesem Begriff werden heute verschiedene Kombinationen des ursprünglichen Modells verstanden, dabei wird zwischen *einlagigen* und *mehrlagigen* Perzeptren (engl. *multi-layer perceptron*, MLP) unterschieden. Perzeptron-Netze wandeln einen Eingabevektor in einen Ausgabevektor um und stellen damit einen einfachen Assoziativspeicher dar.

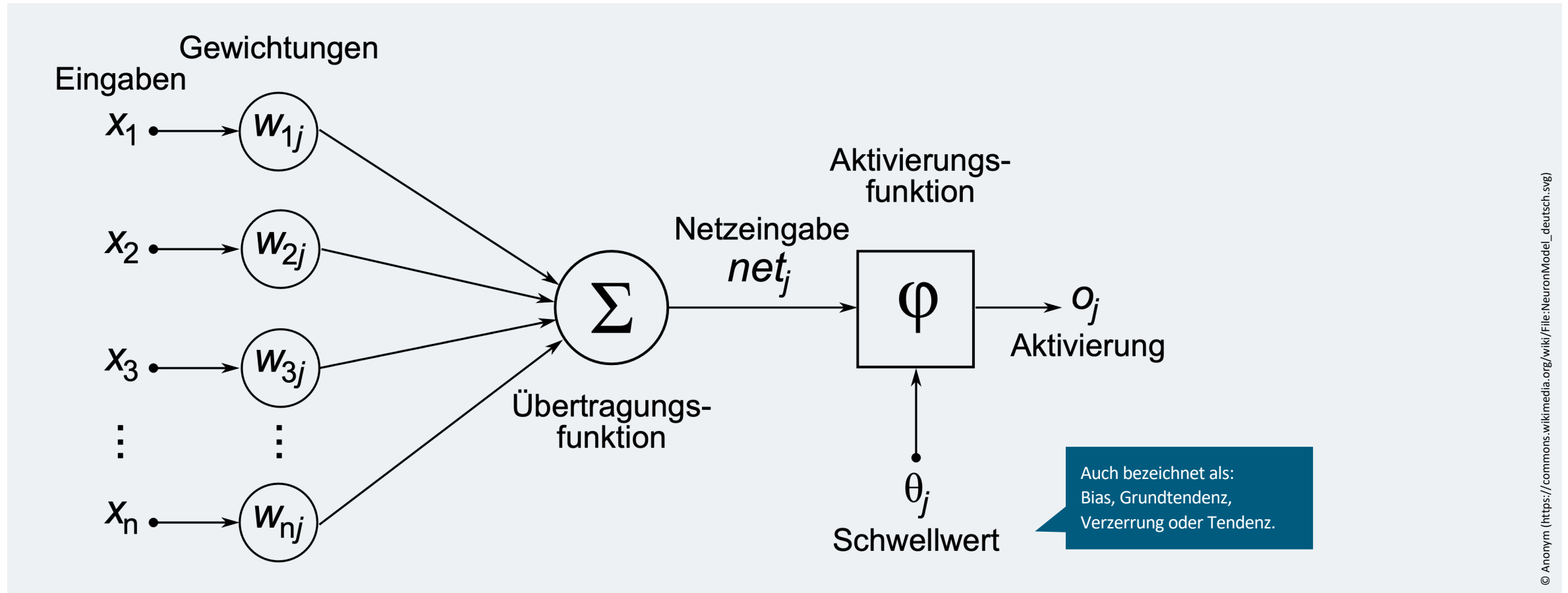
Quelle: Wikipedia_12-10-23
<https://de.wikipedia.org/wiki/Perzeptron>



© MartinThoma (<https://commons.wikimedia.org/wiki/File:Perceptron-or-task.svg>)

Neuronale Netze

Wie funktioniert ein Neuron?



© Anonym (https://commons.wikimedia.org/wiki/File:NeuronModel_deutsch.svg)

Neuronale Netze

Wie funktioniert ein Neuronales Netz im Open Roberta Lab?

Im Open Roberta Lab können seit 2022 Künstliche Neuronale Netze programmiert werden. Im »Plug-In : **Open Roberta xNN**« stehen hierzu Programmier-Blöcke zur Erstellung eines eigenen Künstlichen Neuronalen Netzes* und eine Konfigurationsansicht zur Verfügung, in der das eigentliche Neuronale Netz definiert werden kann.

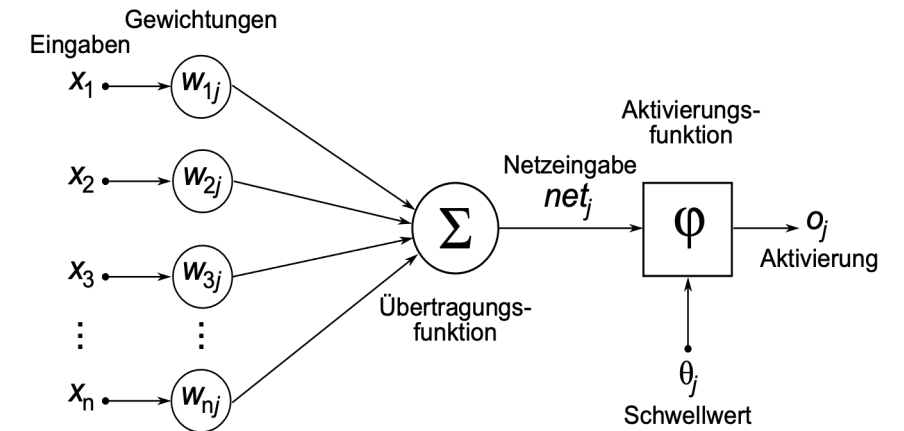
Ziel der Integration von Künstlichen Neuronalen Netzen in Open Roberta Lab ist es, die Grundlagen von KNN zu vermitteln und zu zeigen, wie diese zur Steuerung / zum Training eines Roboters eingesetzt werden können.

Die Integration von KNN wird Open Roberta xNN genannt. xNN steht für explainable Neuronal Networks.

Open Roberta xNN wird vom Ministerium für Schule und Weiterbildung NRW gefördert und in Zusammenarbeit mit Lehrkräften entwickelt:

<https://www.iais.fraunhofer.de/de/presse/presseinformationen/presseinformationen-2022/presseinformation-22mmtt2.html>

*Neuronales Netz und Künstliches Neuronales Netz werden hier synonym verwendet.



© Anonym (https://commons.wikimedia.org/wiki/File:NeuronModel_deutsch.svg)



Kapitel 02

Programmieren mit dem Open Roberta Lab

Open Roberta Lab kennenlernen

Einsteiger-Tutorial für Neulinge

Für Einsteiger*innen wird empfohlen die »Tour« aufzurufen.

Hierzu:

1. Open Roberta Lab: <https://lab.open-roberta.org>
2. Klick auf **Starte die Tour**, um ein interaktives Tutorial zu starten. Es zeigt die prinzipielle Handhabung des Open Roberta Lab und der Simulation.

Weitere Informationen gibt es im offiziellen **Open Roberta Wiki** und auf der Projektwebseite:

- <https://wiki.open-roberta.org>
- www.open-roberta.org

Bearbeiten • Roboter • Hilfe • Anmelden • Tutorials • Galerie • Sprachen

OPEN ROBERTA LAB

Programmiere deine Robots & Boards

Willkommen im Open Roberta Lab

Auf unserer Open-Source-Plattform »Open Roberta Lab« erstellst du im Handumdrehen deine ersten Programme per drag and drop.

Melde dich an
Logge dich ein und habe Zugriff auf deine gespeicherten Programme und Einstellungen. →

Importiere dein NEPO-Programm
Lade deine XML-Datei in unser Lab hoch und mache dort weiter, wo du aufgehört hast. →

Mache eine interaktive Tour
Möchtest du gleich loslegen, weißt aber nicht genau wie? Wir zeigen dir die ersten Schritte in einer interaktiven Anleitung. →

Beliebte Systeme • Alle Systeme und Filteroptionen • 8 Ergebnisse

Open Roberta Sim EV3 leJOS 0.9.1

Calliope mini

Open Roberta xNN

Spike Prime / Robot Inventor

Open Roberta Lab

Neuronale Netze programmieren mit Open Roberta xNN

Um die Programmierung Künstlicher Neuronaler Netze zu starten, muss in der Auswahl der vorhandenen Systeme **Open Roberta xNN** ausgewählt werden

Alternativ kann **Open Roberta xNN** direkt über folgenden Link aufgerufen werden:

- <https://lab.open-roberta.org/?loadSystem=xNN>

Bearbeiten ▾ Roboter ▾ Hilfe ▾ Anmelden ▾ Tutorials Galerie Sprachen ▾

OPEN ROBERTA LAB

Programmiere deine Robots & Boards

Willkommen im Open Roberta Lab

Auf unserer Open-Source-Plattform »Open Roberta Lab« erstellst du im Handumdrehen deine ersten Programme per drag and drop.

Melde dich an
Logge dich ein und habe Zugriff auf deine gespeicherten Programme und Einstellungen. →

Importiere dein NEPO-Programm
Lade deine XML-Datei in unser Lab hoch und mache dort weiter, wo du aufgehört hast. →

Mache eine interaktive Tour
Möchtest du gleich loslegen, weißt aber nicht genau wie? Wir zeigen dir die ersten Schritte in einer interaktiven Anleitung. →

Beliebte Systeme Alle Systeme und Filteroptionen 8 Ergebnisse

Open Roberta Sim EV3 leJOS 0.9.1

Calliope mini

Open Roberta xNN

Spike Prime / Robot Inventor

@slegen → loslegen → loslegen → loslegen →

Open Roberta Lab

Neuronale Netze programmieren mit Open Roberta xNN

Dies ist die Programmieransicht des Open Roberta Lab. Im Tab **NEURONALES Netz Definieren** wird das Neuronale Netz angezeigt und konfiguriert.

The screenshot displays the Open Roberta Lab programming environment. At the top, there is a navigation bar with menu items: 'Bearbeiten', 'Roboter', 'Hilfe', 'Anmelden', 'Tutorials', 'Galerie', and 'Sprachen'. The main header shows several tabs: 'PROGRAMM NEPOprog', 'ROBOTERKONFIGURATION XNNbasis', 'NEURONALES NETZ Definieren' (highlighted with a red circle), and 'NEURONALES NETZ Lernen'. The 'OPEN ROBERTA LAB' logo is in the top right corner. On the left, a vertical sidebar contains various programming blocks: 'Aktion', 'Sensoren', 'Kontrolle', 'Logik', 'Mathematik', 'Neuronales Netz', 'Text', 'Farben', and 'Variablen'. The main workspace shows a red box with '+ Start' and 'zeige Sensordaten' buttons. A faint 3D model of a robot is visible in the background. On the right side, there is a vertical toolbar with icons for undo, redo, help, and simulation. At the bottom right, there are icons for play, save, search, and delete.

Open Roberta Lab

Neuronale Netze programmieren mit Open Roberta xNN

So sieht das vorgenerierte Neuronale Netz aus.

The screenshot displays the Open Roberta Lab interface for configuring a neural network. At the top, there is a navigation bar with options: Bearbeiten, Roboter, Hilfe, Anmelden, Tutorials, Galerie, and Sprachen. The main menu includes PROGRAMM NEPOprog, ROBOTERKONFIGURATION XNNbasis, and NEURONALES NETZ (with sub-options Definieren and Lernen). The NEURONALES NETZ Definieren section is active, showing various configuration parameters:

- Anzeige und Ändern:** alles zeigen
- Form des Netzes:** 1,1
- Berechnung des Neurons:** (empty field)
- Stellen:** 2
- 0 verborgene Schichten:** + -
- Aktivierung:** Linear
- Zufallswerte:** von -1 bis 1

The central workspace shows a neural network diagram with two input nodes (x1 and x5) connected to a single hidden node (1). The output node is labeled 0. The diagram includes control buttons for adding (+), removing (-), and editing (edit icon) nodes. A toolbar at the bottom left contains icons for network structure, connections, undo, redo, home, and gallery. A SIM button is located at the bottom right.

Open Roberta Lab

Neuronale Netze programmieren mit Open Roberta xNN

Mit einem Klick auf + können weitere Eingabe- bzw. Ausgabe-Neuronen erzeugt werden.

Bearbeiten ▾ Roboter ▾ Hilfe ▾ Anmelden ▾ Tutorials Galerie Sprachen ▾

PROGRAMM NEPOprog ROBOTERKONFIGURATION XNNbasis **NEURONALES NETZ Definieren** NEURONALES NETZ Lernen

OPEN ROBERTA LAB

Anzeige und Ändern **alles zeigen** ▾

Form des Netzes 1,1 ✓

Berechnung des Neurons

Stellen 2 ▾

0 verborgene Schichten + -

Aktivierung Linear ▾

Zufallswerte von -1 bis 1 ✓

+ - ✎ + - ✎

x1 1 x5
0

© Markus Jürgens

SIM

Open Roberta Lab

Neuronale Netze programmieren mit Open Roberta xNN

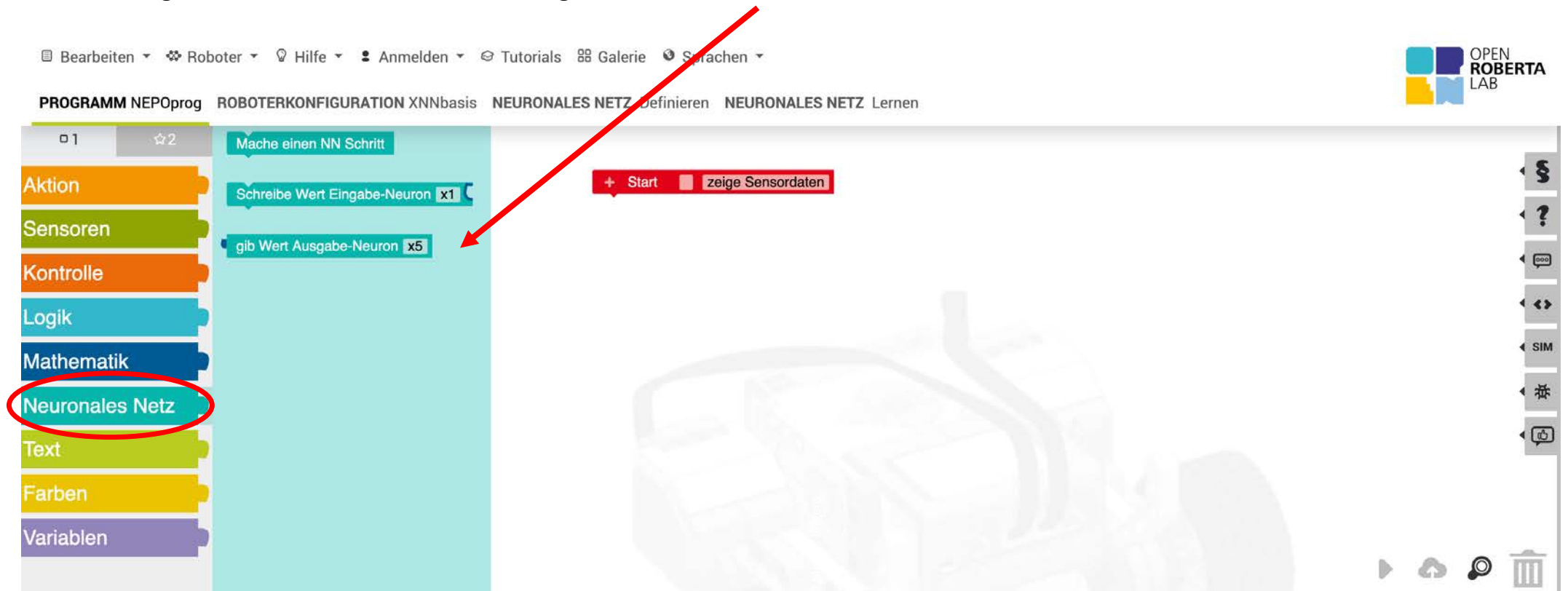
Mit einem Klick auf die gestrichelte Kante (hier: Verbindung zwischen Eingabe-Neuron $n1$ und Ausgabe-Neuron $n2$) wird das Kantengewicht festgelegt werden.

The screenshot displays the Open Roberta Lab interface for configuring a neural network. At the top, there are navigation menus and the Open Roberta Lab logo. Below this, a series of tabs indicate the current step: 'PROGRAMM NEPOprog', 'ROBOTERKONFIGURATION XNNbasis', 'NEURONALES NETZ Definieren', and 'NEURONALES NETZ Lernen'. The main workspace is divided into several control panels: 'Anzeige und Ändern' (set to 'alles zeigen'), 'Form des Netzes' (set to '1,1'), 'Berechnung des Neurons', 'Stellen' (set to '2'), '0 verborgene Schichten' (with '+' and '-' buttons), 'Aktivierung' (set to 'Linear'), and 'Zufallswerte' (set to 'von -1 bis 1'). The central workspace shows a neural network diagram with an input neuron 'x1' and an output neuron 'u'. A dashed blue line connects them, with a weight value of '1' displayed above it. A red arrow points to this dashed line, and a red circle highlights the weight input field, which contains the value '1' and has '+' and '-' buttons. At the bottom left, there are icons for various tools, and at the bottom right, there is a 'SIM' button.

Open Roberta Lab

Neuronale Netze programmieren mit Open Roberta xNN

Durch einen Klick auf den Tab **Programm NEPOprog** gelangt man zurück in die Programmansicht. Die grundlegenden NEPO-Blöcke, die für das Neuronale Netz benötigt werden, befinden sich in der Kategorie **Neuronales Netz**.



Open Roberta Lab

Neuronale Netze programmieren mit Open Roberta xNN

Es gibt noch mehr!

Mit einem Klick auf die Kategorie ★2, bekommt man Zugriff auf weitere Programmier-Blöcke in der Kategorie Neuronales Netz.

The screenshot shows the Open Roberta Lab interface. At the top, there are navigation menus: 'Bearbeiten', 'Roboter', 'Hilfe', 'Anmelden', 'Tutorials', 'Galerie', and 'Sprachen'. The main workspace is divided into several tabs: 'PROGRAMM NEPOprog', 'ROBOTERKONFIGURATION XNNbasis', 'NEURONALES NETZ Definieren', and 'NEURONALES NETZ Lernen'. The 'Neuronales Netz' category is highlighted with a red circle and a star icon. The workspace contains several blocks for defining and training a neural network:

- Mache einen NN Schritt** (teal block)
- Schreibe Wert Eingabe-Neuron x1** (teal block)
- gib Wert Ausgabe-Neuron x5** (teal block)
- Schreibe Gewicht von x1 nach x1** (teal block)
- Schreibe Bias x5** (teal block)
- gib Gewicht von x1 nach x1** (teal block)
- gib Bias x5** (teal block)

At the bottom of the workspace, there are two red blocks: **Start** and **zeige Sensordaten**. The interface also features a sidebar with various categories: 'Aktion', 'Sensoren', 'Kontrolle', 'Logik', 'Mathematik', 'Neuronales Netz', 'Text', 'Listen', 'Farben', and 'Variablen'. The 'Neuronales Netz' category is highlighted. The interface also includes a toolbar with icons for undo, redo, search, and delete, and a 'SIM' button.

Open Roberta Lab

Neuronale Netze programmieren mit Open Roberta xNN

Eingabe- und Ausgabe-Neuronen benötigen einen eindeutigen Namen.

Eingabe-Neuronen können konstante Werte, Variablen, Sensor-Werte oder auch Rückgabewerte aus Funktionen erhalten.

- 1 Mit dem Block **Mache einen NN Schritt** wird die Berechnung im gesamten Neuronalen Netz durchgeführt

- 3 Die im KNN ermittelten Werte können über den Block **gib Wert Ausgabe-Neuron** genutzt werden.

The screenshot displays the Open Roberta Lab programming environment. The interface includes a top navigation bar with options like 'Bearbeiten', 'Roboter', 'Hilfe', 'Anmelden', 'Tutorials', 'Galerie', and 'Sprachen'. Below this, there are tabs for 'PROGRAMM NEPOprog', 'ROBOTERKONFIGURATION XNNbasis', 'NEURONALES NETZ Definieren', and 'NEURONALES NETZ Lernen'. A left sidebar lists various block categories: 'Aktion', 'Sensoren', 'Kontrolle', 'Logik', 'Mathematik', 'Neuronales Netz', 'Text', 'Farben', and 'Variablen'. The main workspace shows a sequence of blocks: a red 'Start' block, a red 'zeige Sensordaten' block (pointed to by red circle 1), a blue 'Schreibe Wert Eingabe-Neuron n1' block (pointed to by red circle 2), a green 'Machen einen NN Schritt' block (pointed to by red circle 2), a blue 'Zeige Text gib Wert Ausgabe-Neuron n2' block (pointed to by red circle 3), and a blue 'Warte ms 5000' block. The background features a faint image of a robot.

Open Roberta Lab

Neuronale Netze programmieren mit Open Roberta xNN

The screenshot displays the Open Roberta Lab interface. On the left, a block-based programming environment is shown with a sidebar containing categories like 'Aktion', 'Sensoren', 'Kontrolle', 'Logik', 'Mathematik', 'Neuronales Netz', 'Text', 'Farben', and 'Variablen'. The main workspace contains a sequence of blocks: 'Start' (with 'zeige Sensordaten'), 'Variable output : Zahl' (set to 0), 'Schreibe Wert Eingabe-Neuron n1' (set to 2), 'Mache einen NN Schritt', and 'Schreibe output' (with 'gib Wert Ausgabe-Neuron n2'). A red arrow points from the 'Schreibe Wert Eingabe-Neuron n1' block to the configuration panel on the right.

The configuration panel on the right is titled 'NEURONALES NETZ Definieren' and includes the following settings:

- Anzeige und Ändern: alles zeigen
- Form des Netzes: 1,1
- Berechnung des Neurons: (empty)
- Stellen: 2
- 0 verborgene Schichten: + -
- Aktivierung: Linear
- Zufallswerte: von -1 bis 1

Below these settings, a diagram shows a neural network with two input neurons (n1 and n2) and one output neuron (n2). A red dashed arrow indicates the flow from left to right.

Ablauf:

Mit dem Block **Schreibe Wert Eingabe-Neuron $n1$** wird ein konkreter Wert dem NN Eingabe Neuron $n1$ übergeben/zugewiesen. Dieser Wert steht nun im Neuronalen Netz zur Verfügung und wird durch den Block **Mache einen NN Schritt** im Neuronalen Netz berechnet. Mit diesem Block (Mache einen NN Schritt) wird das gesamte Neuronale Netz, unabhängig von der Anzahl der Neuronen, einmal von links nach rechts durchlaufen. Anschließend kann über den Block **gib Wert Ausgabe-Neuron $n2$** der Wert aus dem Neuronalen Netz abgerufen / geholt werden. Gestartet wird das Programm in der Simulation (SIM). Klick auf den Start-Knopf.

Neuronales Netz

Benennung der Neuronen bei Open Roberta xNN

- 1 Name des Eingabe-Neurons.
- 2 Name des Ausgabe-Neurons.
- 3 Bezeichnung der Neuronen in der verborgenen Schicht (engl. hidden layer)

The screenshot displays the Open Roberta xNN web interface. At the top, there is a navigation bar with options like 'Bearbeiten', 'Roboter', 'Hilfe', 'Anmelden', 'Tutorials', 'Galerie', and 'Sprachen'. Below this is a menu bar with 'PROGRAMM NEPOprog', 'ROBOTERKONFIGURATION XNNbasis', 'NEURONALES NETZ Definieren', and 'NEURONALES NETZ Lernen'. The main control panel includes a dropdown for 'Anzeige und Ändern' (set to 'alles zeigen'), a 'Form des Netzes' field with '1,3,1', a 'Berechnung des Neurons' field, a 'Stellen' dropdown (set to '2'), a '1 verborgene Schicht' section with '+' and '-' buttons, an 'Aktivierung' dropdown (set to 'Linear'), and a 'Zufallswerte' section with 'von -1 bis 1'. The central area shows a neural network diagram with an input neuron 'n1' (green), three hidden neurons 'h1n1', 'h1n2', and 'h1n3' (grey), and an output neuron 'n2' (orange). Red circles with numbers 1, 2, and 3 point to the input, output, and hidden neurons respectively. Weights are shown as '0' on the connections. A 'SIM' button is at the bottom right.

Beispiele mit Open Roberta xNN

Neuronales Netz

Beispiele

Die folgenden Folien zeigen verschiedene Beispiele. Diese sollen veranschaulichen, wie die Werte im Neuronalen Netz berechnet werden.

Da es sich bei den neuronalen Netzen im Open Roberta Lab bisher um sogenannte **Feedforward-Netze** handelt (das Netz wird mit jedem NN-Schritt einmal von links nach rechts durchlaufen), ist die Berechnung der einzelnen Neuronen trivial.

Ein Klick auf das jeweilige Neuron (mit Ausnahme der Eingabe-Neuronen) zeigt, wie der Wert des Neurons berechnet wird.

Für dieses Beispiel lautet die Berechnung: $1 * n1 + 2 * n3 = n2$

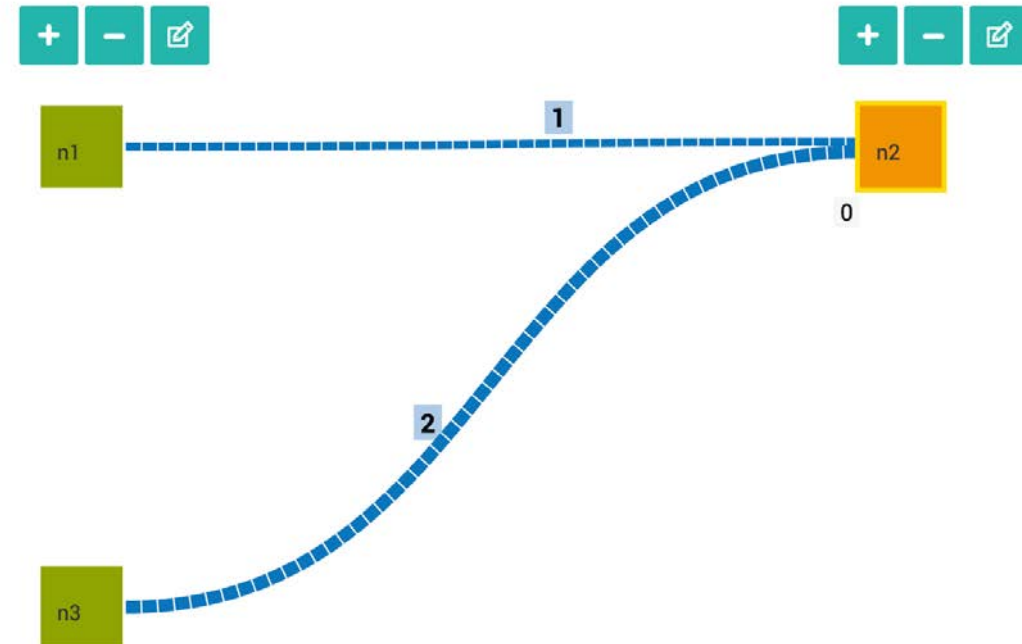
(1 mal Wert des Neurons $n1$ plus 2 mal Wert des Neurons $n3$ ergibt den Wert des Ausgabe-Neurons $n2$)

Berechnung des Neurons

$1*n1 + 2*n3$

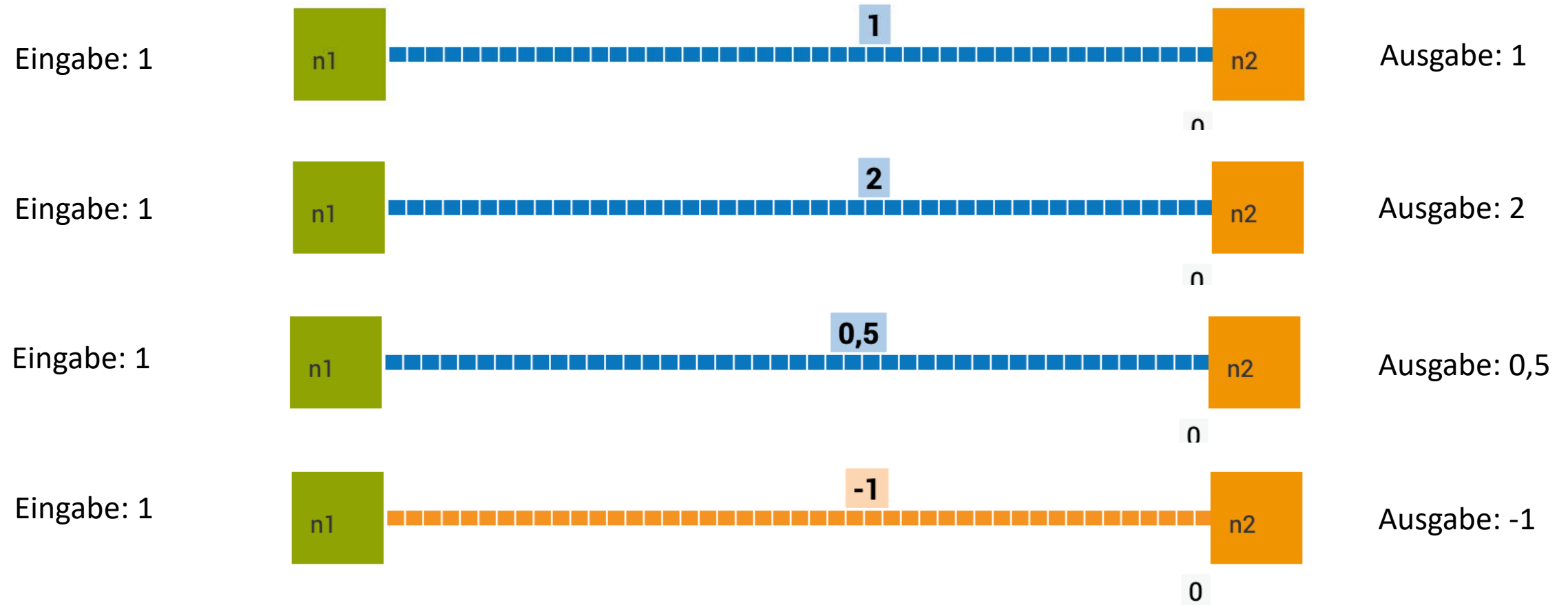
Nachkomma-Stellen

2



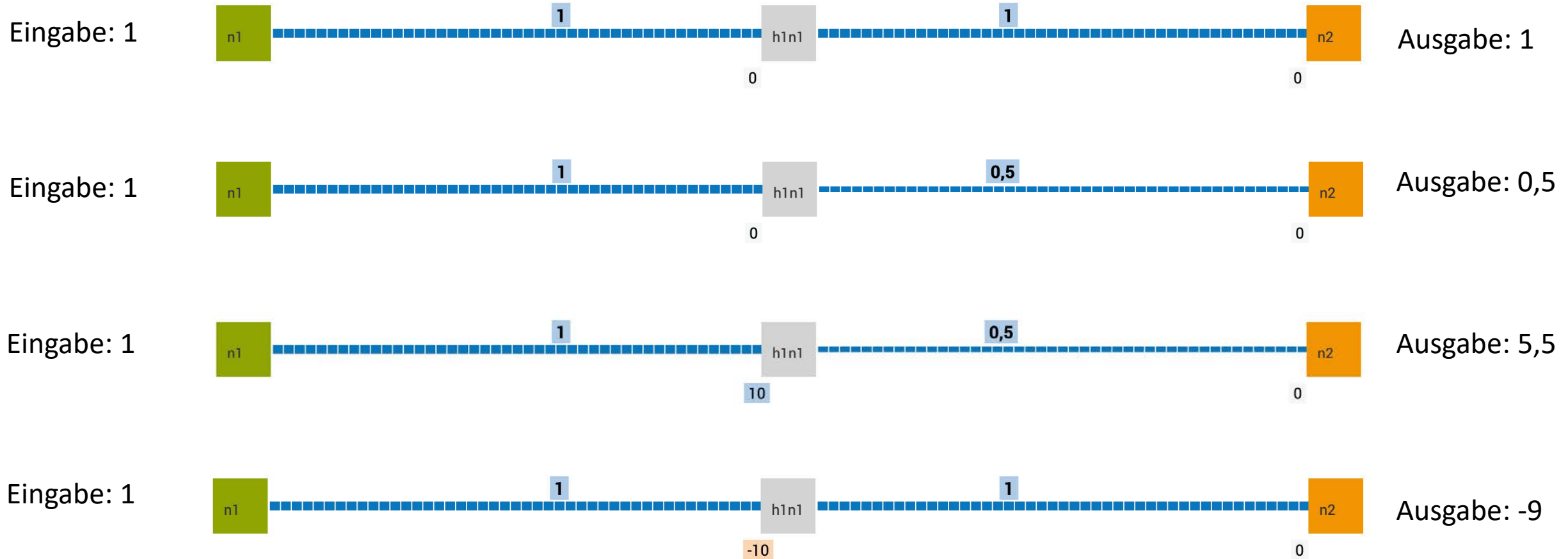
Neuronales Netz

Beispiele



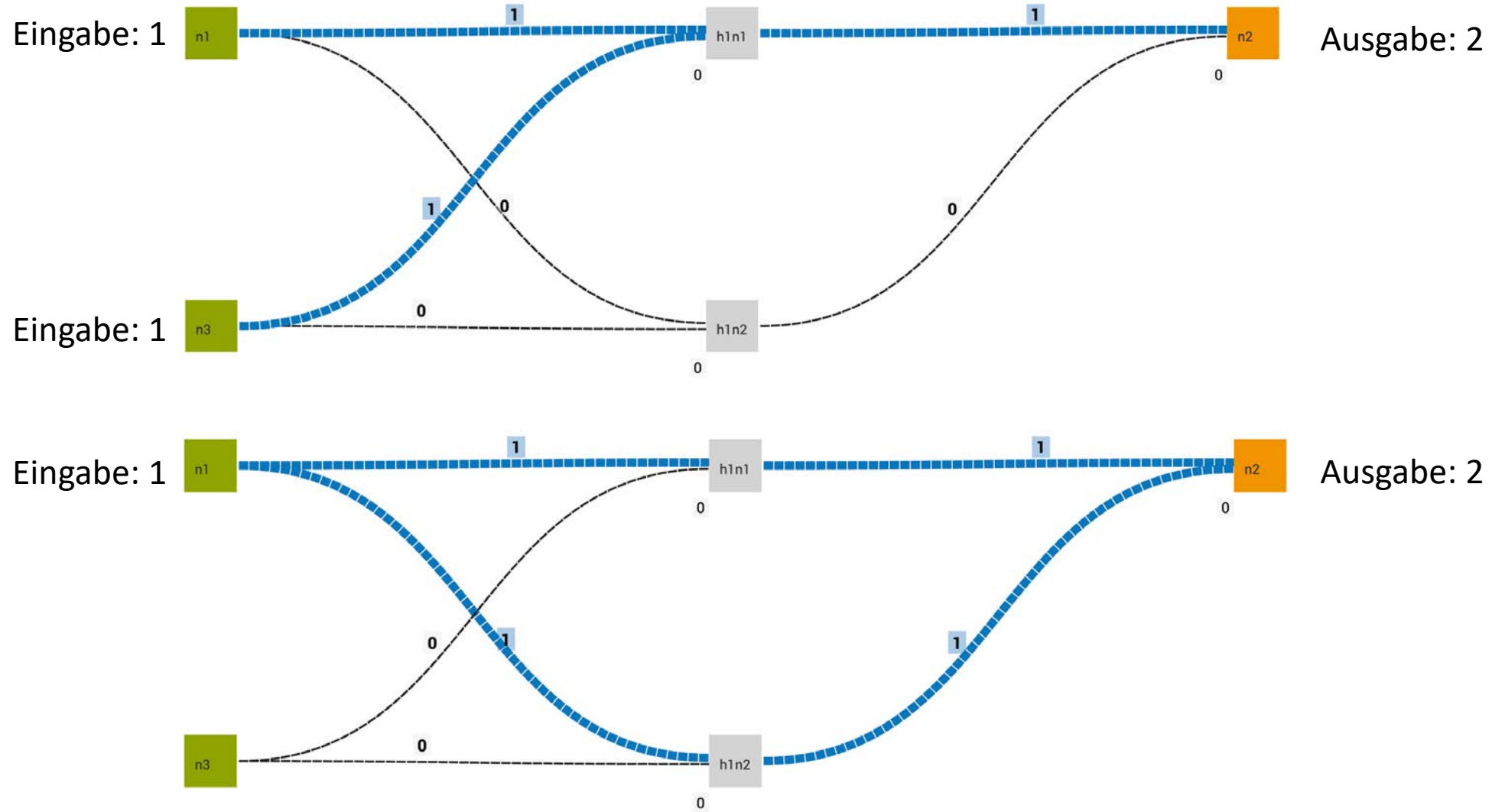
Neuronales Netz

Beispiele



Neuronales Netz

Beispiele



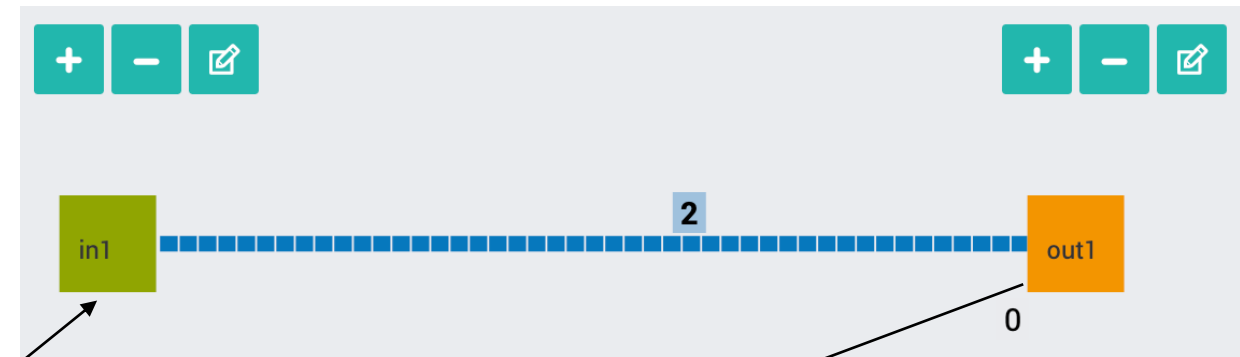
Neuronales Netz

Beispiel 1

Welchen Wert hat das Ausgabe-Neuron *out1* bzw. die Variable *output*?

- Neuronales Netz mit einem Eingabe-Neuron *in1*
- Ändern des Gewichts der Kante von *in1* zu *out1* = 2

Ergebnis Variable: *output* = 2



Neuronales Netz

Beispiel 1 – Erläuterung

1 Eingabe-Neuron

2 Ausgabe-Neuron

3 Kantengewicht

4 Bias

Bearbeiten ▾ Roboter ▾ Hilfe ▾ Anmelden ▾ Tutorials Galerie Sprachen ▾



PROGRAMM NEPOprog ROBOTERKONFIGURATION XNNbasis **NEURONALES NETZ** Definieren NEURONALES NETZ Lernen

Anzeige und Ändern

alles zeigen ▾

Form des Netzes

1,1 ✓

Berechnung des Neurons

Stellen

2 ▾

0 verborgene Schichten

+

-

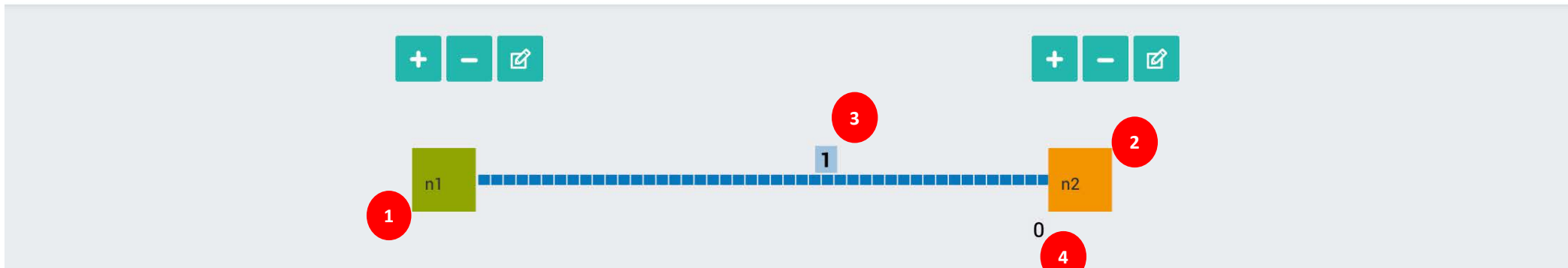
Aktivierung

Linear ▾

Zufallswerte

von -1 bis

1 ✓



Neuronales Netz

Änderung der Namen der Neuronen

1 Klick auf Eingabe-Neuron

2 Klick auf Ausgabe-Neuron

Bearbeiten ▾ Roboter ▾ Hilfe ▾ Anmelden ▾ Tutorials Gallerie Sprachen ▾



PROGRAMM NEPOprog ROBOTERKONFIGURATION XNNbasis **NEURONALES NETZ** Definieren NEURONALES NETZ Lernen

Anzeige und Ändern

alles zeigen ▾

Form des Netzes

1,1 ✓

Berechnung des Neurons

Stellen

2 ▾

0 verborgene Schichten

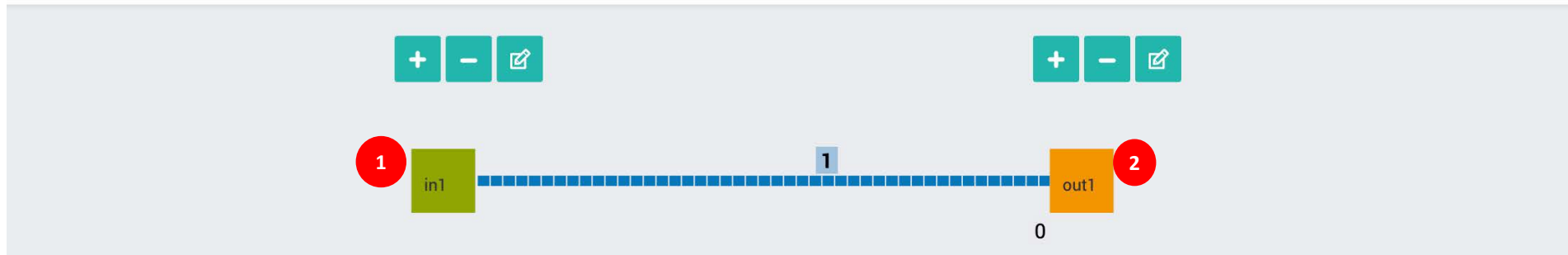
+ -

Aktivierung

Linear ▾

Zufallswerte

von -1 bis 1 ✓



Neuronales Netz

Beispiel 1 – Erläuterung

- 1 Eingabe-Neuron *in1* erhält den Wert 1
- 2 Diese steht dem NN zur Verfügung
- 3 Die Gewichtung multipliziert den Wert mit 2
- 4 Das Ausgabe-Neuron *out1* hat den Wert 2
- 5 Die Variable *output* erhält den Wert 2

PROGRAMM NEPOprogr ROBOTERKONFIGURATION XNNbasis **NEURONALES NETZ** Definieren NEURONALES NETZ Lernen

Anzeige und Ändern Form des Netzes Berechnung des Neurons Stellen 0 verborgene Schichten Aktivierung Zufallswerte

alles zeigen 1,1 2 + - Linear von -1 bis 1

in1 2 1 3 out1 4 0

+ Start zeige Sensordaten

- Variable output : Zahl ← 0

Schreibe Wert Eingabe-Neuron in1 1 1

Mache einen NN Schritt

Schreibe output gib Wert Ausgabe-Neuron out1 5

Neuronales Netz

Beispiel 2

Welchen Wert hat out1 bzw. die Variable *output*?

- Neuronales Netz mit zwei Eingabe-Neuronen *in1* und *in2*
- Wert und Gewichtung bei *in1* bleiben wie in Beispiel 1
- Ändere die Gewichtung der Kante von *in2* zu *out1* = -1

The screenshot shows the NEPO software interface for configuring a neural network. The top navigation bar includes options like 'Bearbeiten', 'Roboter', 'Hilfe', 'Anmelden', 'Tutorials', 'Galerie', and 'Sprachen'. The main menu shows 'PROGRAMM NEPOprogr', 'ROBOTERKONFIGURATION XNNbasis', and 'NEURONALES NETZ' (highlighted). Below the menu, there are several control panels: 'Anzeige und Ändern' (set to 'alles zeigen'), 'Form des Netzes' (set to '2,1'), 'Berechnung des Neurons' (empty), 'Stellen' (set to '2'), '0 verborgene Schichten' (with '+' and '-' buttons), 'Aktivierung' (set to 'Linear'), and 'Zufallswerte' (set to 'von -1 bis 1'). The main workspace displays a neural network diagram with two input neurons, 'in1' and 'in2', and one output neuron, 'out1'. A blue line connects 'in1' to 'out1' with a weight of '1'. An orange curved line connects 'in2' to 'out1' with a weight of '-1'. The output neuron 'out1' has a value of '0' next to it.

The code blocks are as follows:

- Start** (red block) with a checkbox for 'zeige Sensordaten'.
- Variable** (red block) 'output : Zahl' set to 0.
- Schreibe Wert Eingabe-Neuron** (teal block) 'in1' set to 1.
- Schreibe Wert Eingabe-Neuron** (teal block) 'in2' set to 1.
- Mache einen NN Schritt** (teal block).
- Schreibe** (purple block) 'output' to 'gib Wert Ausgabe-Neuron out1'.

Neuronales Netz

Beispiel 2 – Tipp

Wie der Wert eines Neurons berechnet wird, sieht man, wenn das Ausgabe-Neuron *out1* angeklickt wird.

Im Beispiel rechts: $out1 = 1 * in1 - 1 * in2$

Hinweis: Es wird immer das Neuron angezeigt, das gelb umrandet ist.

The screenshot shows the NEPOprog software interface for configuring a neural network. The top navigation bar includes options like 'Bearbeiten', 'Roboter', 'Hilfe', 'Anmelden', 'Tutorials', 'Galerie', and 'Sprachen'. The main menu shows 'PROGRAMM NEPOprog', 'ROBOTERKONFIGURATION XNNbasis', and 'NEURONALES NETZ' (highlighted). Below the menu, there are several control panels: 'Anzeige und Ändern' (set to 'alles zeigen'), 'Form des Netzes' (set to '2,1'), 'Berechnung des Neurons' (displaying the formula $out1 = 1 * in1 - 1 * in2$), 'Stellen' (set to '2'), '0 verborgene Schichten' (with '+' and '-' buttons), 'Aktivierung' (set to 'Linear'), and 'Zufallswerte' (set to 'von -1 bis 1'). The main workspace shows a diagram of a neural network with two input neurons, 'in1' and 'in2', and one output neuron, 'out1'. 'in1' is connected to 'out1' with a weight of 1, and 'in2' is connected to 'out1' with a weight of -1. The 'out1' neuron is highlighted with a red circle, and a red arrow points from it to the 'Berechnung des Neurons' field.

Neuronales Netz

Beispiel 2 – Lösung

- 1 Klicke auf Sim, um die Simulationsumgebung zu öffnen.
- 2 Klicke auf die Schaltfläche für Sensordaten.
- 3 Klappe Variablen-Werte auf.
- 4 Starte das Programm über die Start-Schaltfläche.

Ergebnis Variable: *output* = 1

Neuronales Netz

Kantengewichtung

Die Stärke der Kante visualisiert die Gewichtung.

Die Kante zwischen Eingabe-Neuron *in1* und Ausgabe-Neuron *out1* ist stärker als die Kante zwischen Eingabe-Neuron *in2* und Ausgabe-Neuron *out1*, somit ist auch das Kantengewicht höher.

PROGRAMM NEPOprog ROBOTERKONFIGURATION XNNbasis **NEURONALES NETZ** Definieren NEURONALES NETZ Lernen

Anzeige und Ändern Form des Netzes Berechnung des Neurons Stellen 0 verborgene Schichten Aktivierung Zufallswerte

Klick Gewichte oder Bias 2,1 Linear von -1 bis 1

in1 in2 out1

Kapitel 04

Ausblick

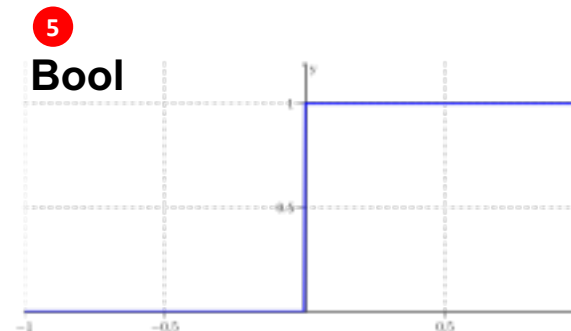
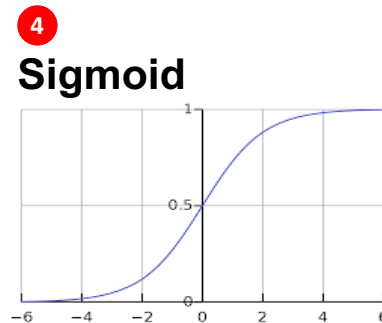
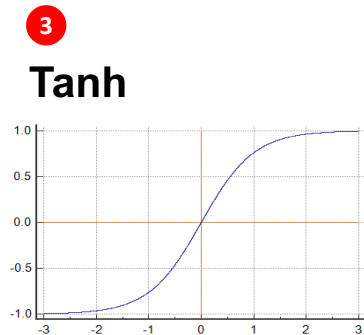
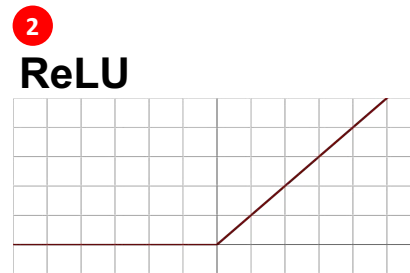
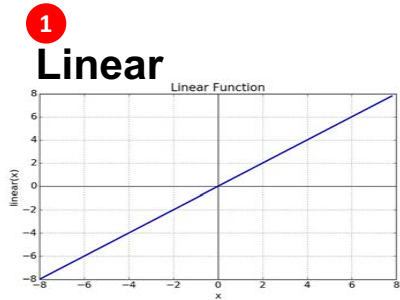
Neuronales Netz

Aktivierungsfunktion

Es lassen sich diverse Aktivierungsfunktionen einstellen.

Mehr zur Aktivierungsfunktion unter:

https://de.wikipedia.org/wiki/K%C3%BCnstliches_Neuron



Quelle: Wikipedia 12-10-23: https://de.wikipedia.org/wiki/K%C3%BCnstliches_Neuron

Bearbeiten ▾ Roboter ▾ Hilfe ▾ Anmelden ▾ Tutorials Galerie Sprachen ▾



PROGRAMM NEPOprog ROBOTERKONFIGURATION XNNbasis **NEURONALES NETZ** Definieren NEURONALES NETZ Lernen

Anzeige und Ändern Form des Netzes Berechnung des Neurons Stellen 0 verborgene Schichten **Aktivierung** Zu

alles zeigen 1,1 ✓ 2 ✓ + -

1 von -1 bis 1 ✓

2

3

4

5

+

-

+

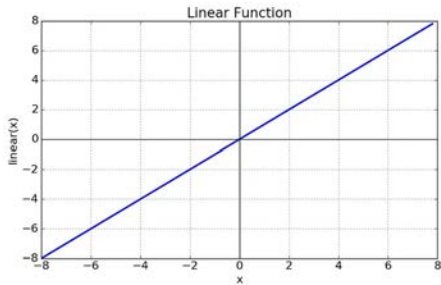
-

in1 1 out1 0

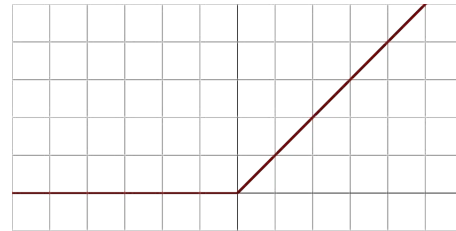
Aktivierungsfunktionen

Was hat es mit den Aktivierungsfunktionen auf sich?

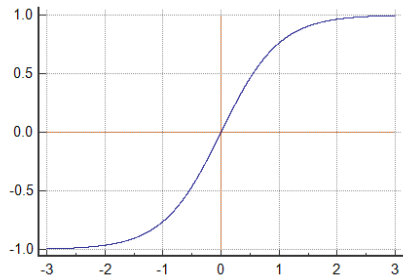
Linear



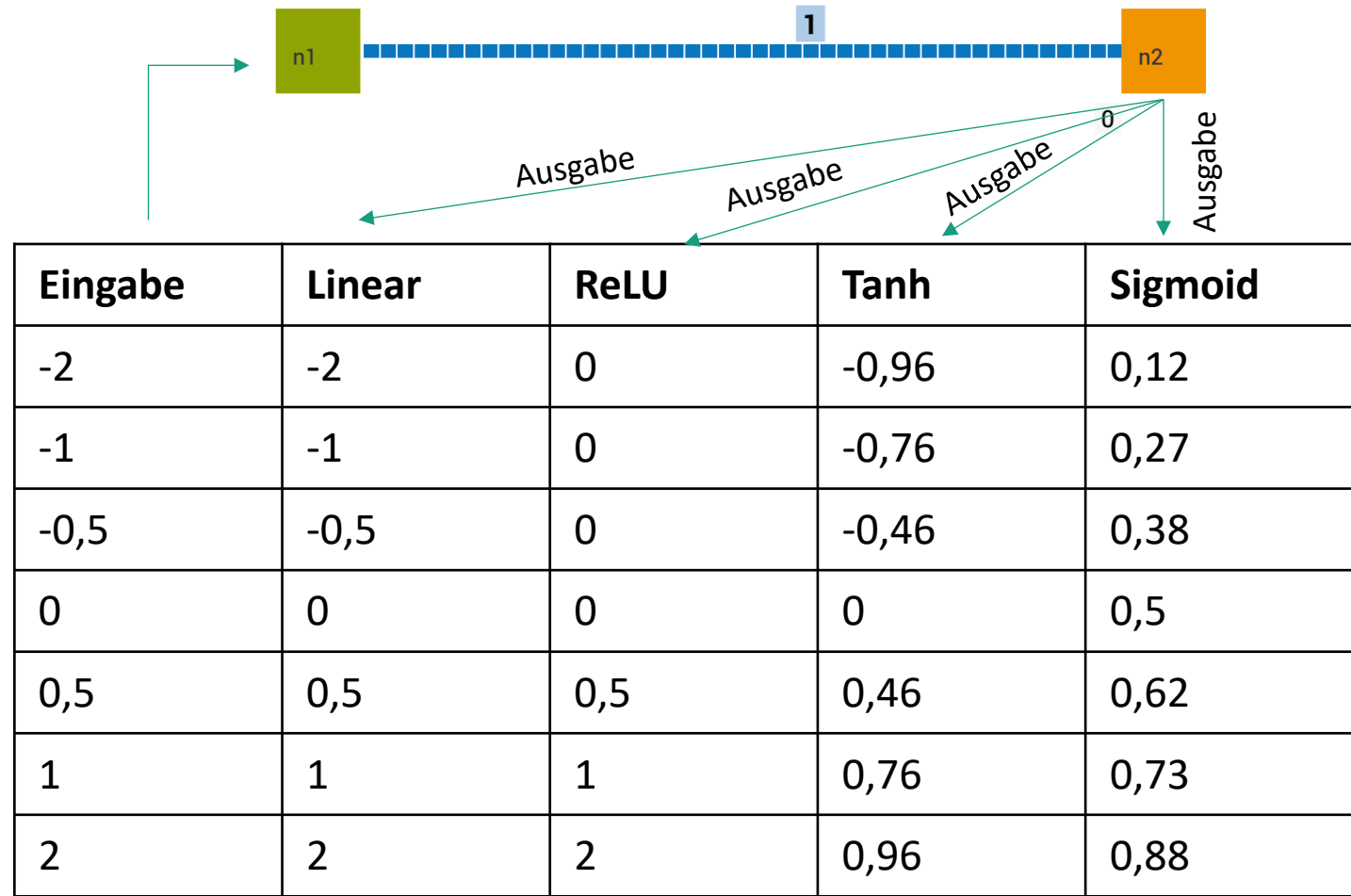
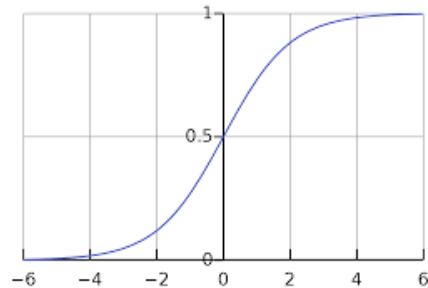
ReLU



Tanh



Sigmoid



Neuronales Netz

Aktivierungsfunktion

Um die verschiedenen Aktivierungsfunktionen zu verstehen, empfiehlt es sich, die Ausgabewerte mit verschiedenen Eingabewerten zu betrachten.

Die Variablenwerte können in der Simulation mit einem Klick auf die Schaltfläche Ansicht Sensordaten eingesehen werden.

The screenshot displays a software interface for programming a robot. The main workspace shows a sequence of programming blocks: a red 'Start' block, a red 'zeige Sensordaten' block, a red 'Variable output : Zahl' block, two cyan 'Schreibe Wert Eingabe-Neuron' blocks, a cyan 'Mache einen NN Schritt' block, a cyan 'Schreibe output' block, a cyan 'gib Wert' block, and an orange 'Warte ms 1000' block. Below the programming area is a simulation window showing a yellow robot on a track with four colored blocks (blue, yellow, green, red). A 'Sensordaten' (Sensor Data) window is open, displaying a list of values: 'System-Werte', 'Zeitgeber-Werte', 'Sensor-Werte', and 'Variablen-Werte'. The 'Variablen-Werte' section shows 'output : 3.5'. The interface includes a top menu bar with options like 'PROGRAMM NEPOprog', 'ROBOTERKONFIGURATION XNNbasis', 'NEURONALES NETZ Definieren', and 'NEURONALES NETZ Lernen'. A left sidebar contains various tool icons, and a bottom toolbar includes play, stop, and other control buttons.

Neuronales Netz

Ausblick

1. Wird ein Programm gespeichert, dann werden auch die Werte des KNN (Kantengewicht, Schwellenwert, Aktivierungsfunktion,...) mitgespeichert.
2. Neben den Werten der Ausgabe-Neuronen können auch die Kantengewichte und Schwellenwerte abgerufen werden. Dies erfolgt durch diese Blöcke:

gib Gewicht von nach

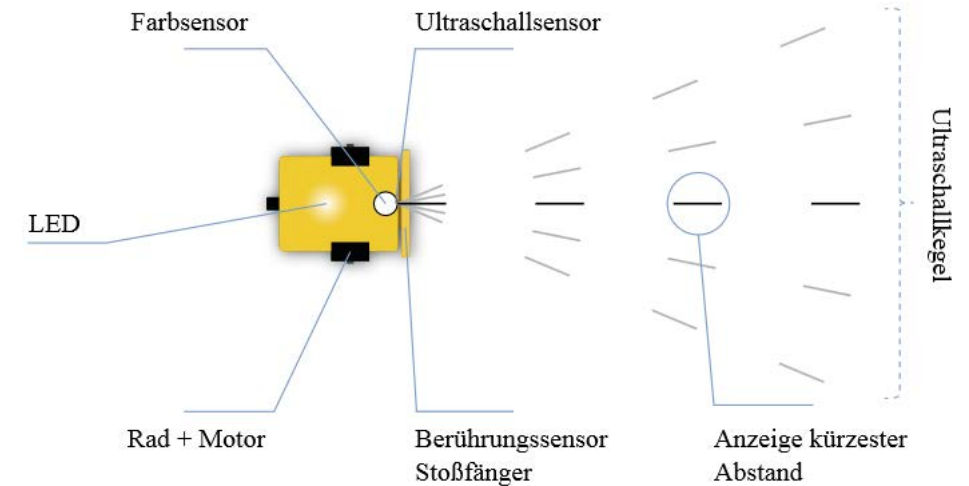
gib Bias

gib Wert Ausgabe-Neuron

Neuronales Netz

Ausblick – Roboter mit Open Roberta xNN programmieren

Statt eines festen Parameters können einem Eingabe-Neuron auch Sensorwerte übergeben werden. So können wir beispielsweise den Ultraschallsensor verwenden. Der Ultraschallsensor ist in der Mitte des Roboters angebracht.



Kontakt

Thorsten Leimbach
Roberta-Zentrale@iais.fraunhofer.de

Fraunhofer-Institut für Intelligente Analyse-
und Informationssysteme IAIS

Schloss Birlinghoven 1
53757 Sankt Augustin

www.iais.fraunhofer.de