

Eine Unterrichtssequenz zum Einstieg in das Themengebiet Neuronale Netze

Kerstin Strecker

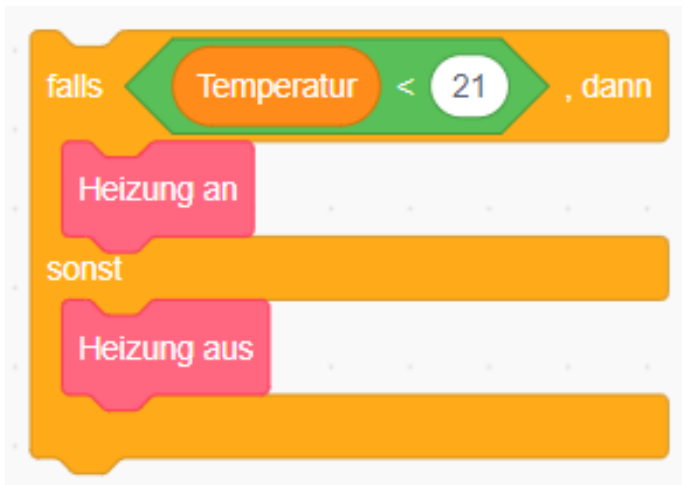
Anmerkung

- Die Inhalte dieser Folien bringen ausschließlich Ansichten und Meinungen der Autorin zum Ausdruck. Für die korrekte Ausführbarkeit der angegebenen Beispiel Quelltexte dieser Folien wird keine Garantie übernommen. Auch eine Haftung für Folgeschäden, die sich aus der Anwendung der Quelltexte dieser Folien oder durch eventuelle fehlerhafte Angaben ergeben, wird keine Haftung oder juristische Verantwortung übernommen.
- Alle Programme sind nur exemplarisch zu sehen und müssen angepasst werden

Maschinelles Lernen

Klassische Algorithmik:

- **Regeln** werden definiert, die Eingabedaten Ausgabedaten zuordnen.
- Beispiel: 18 °C => Heizung an



Überwachtes Lernen:

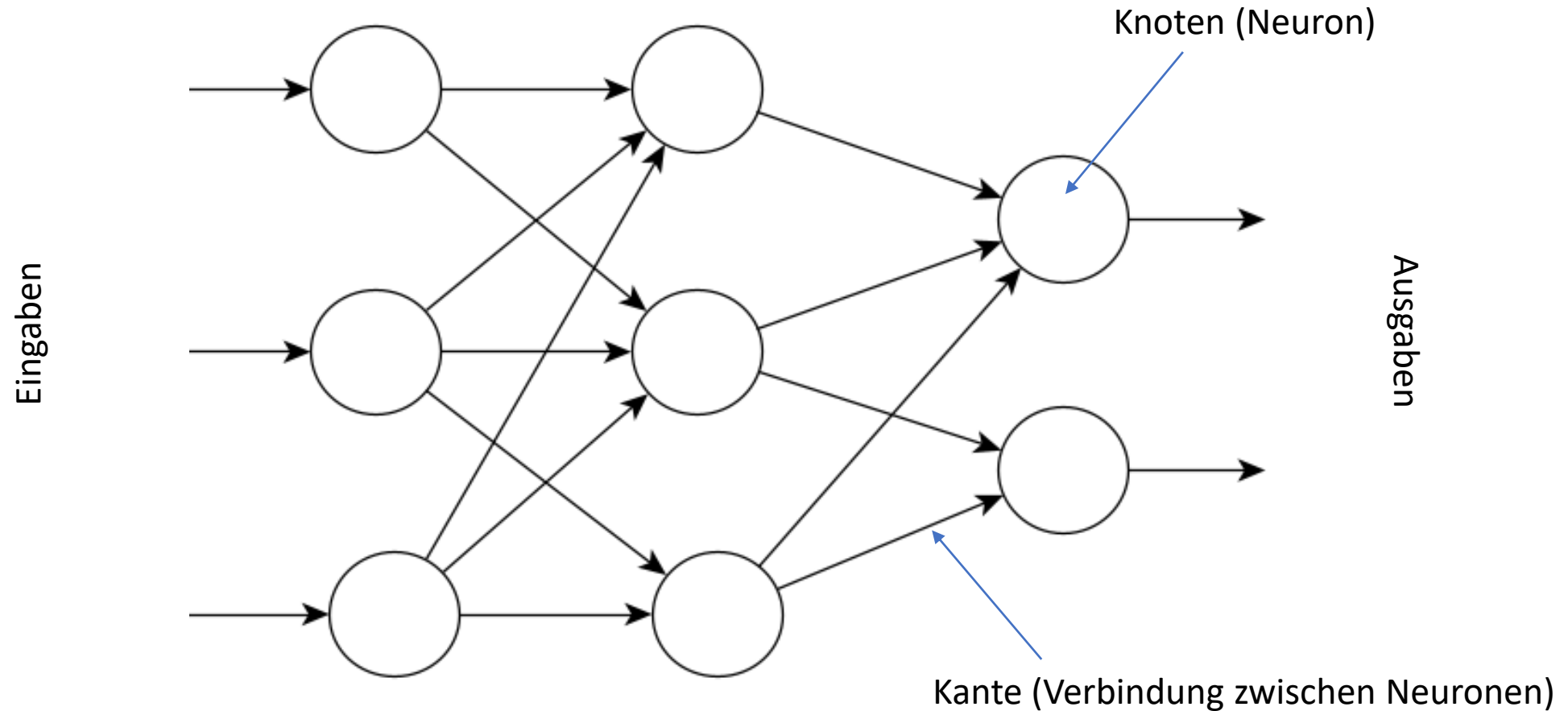
- Sehr viele Ein- und Ausgabedaten werden gegeben
- Muster werden erkannt
- 18°C => Heizung an

Eingabe	Ausgabe
13 °	Heizung an
23 °	Heizung aus
25 °	Heizung aus
14 °	Heizung an
17°	Heizung an
21 °	Heizung aus
20°	Heizung an
22 °	Heizung aus
19°	Heizung an
...	...

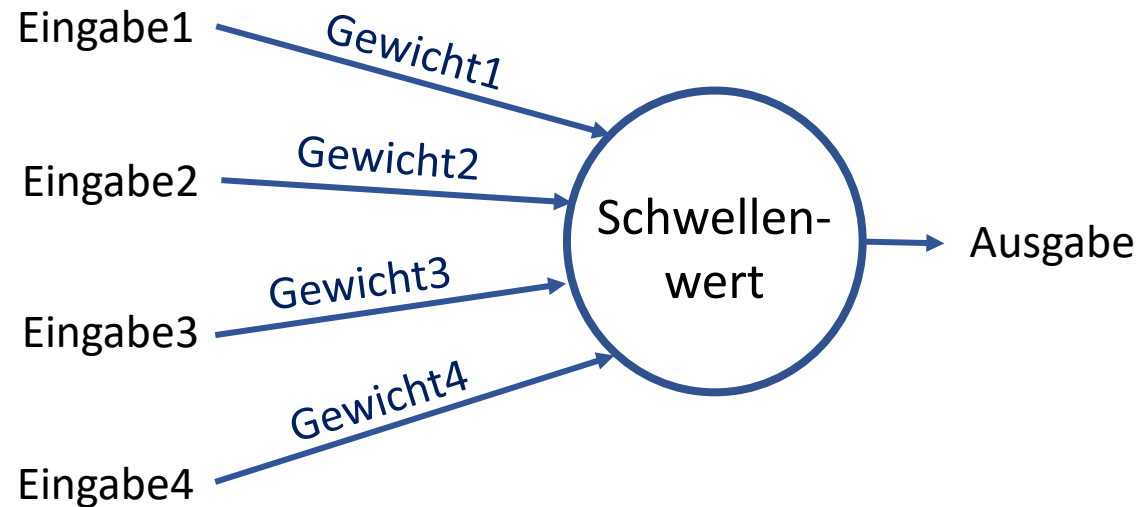
Muster

Exkurs: Neuronale Netze Grundlagen

Künstliches Neuronales Netz



Neuron



Beispiel:

Eingabe 1 = 2

Eingabe 2 = 4

Eingabe 3 = -3

Eingabe 4 = 8

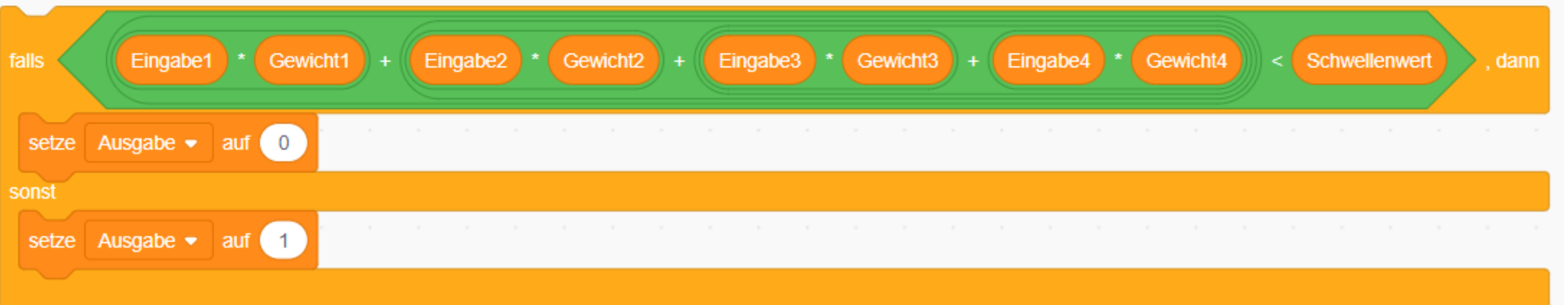
Gewicht 1 = 0

Gewicht 2 = 3

Gewicht 3 = 5

Gewicht 4 = 1

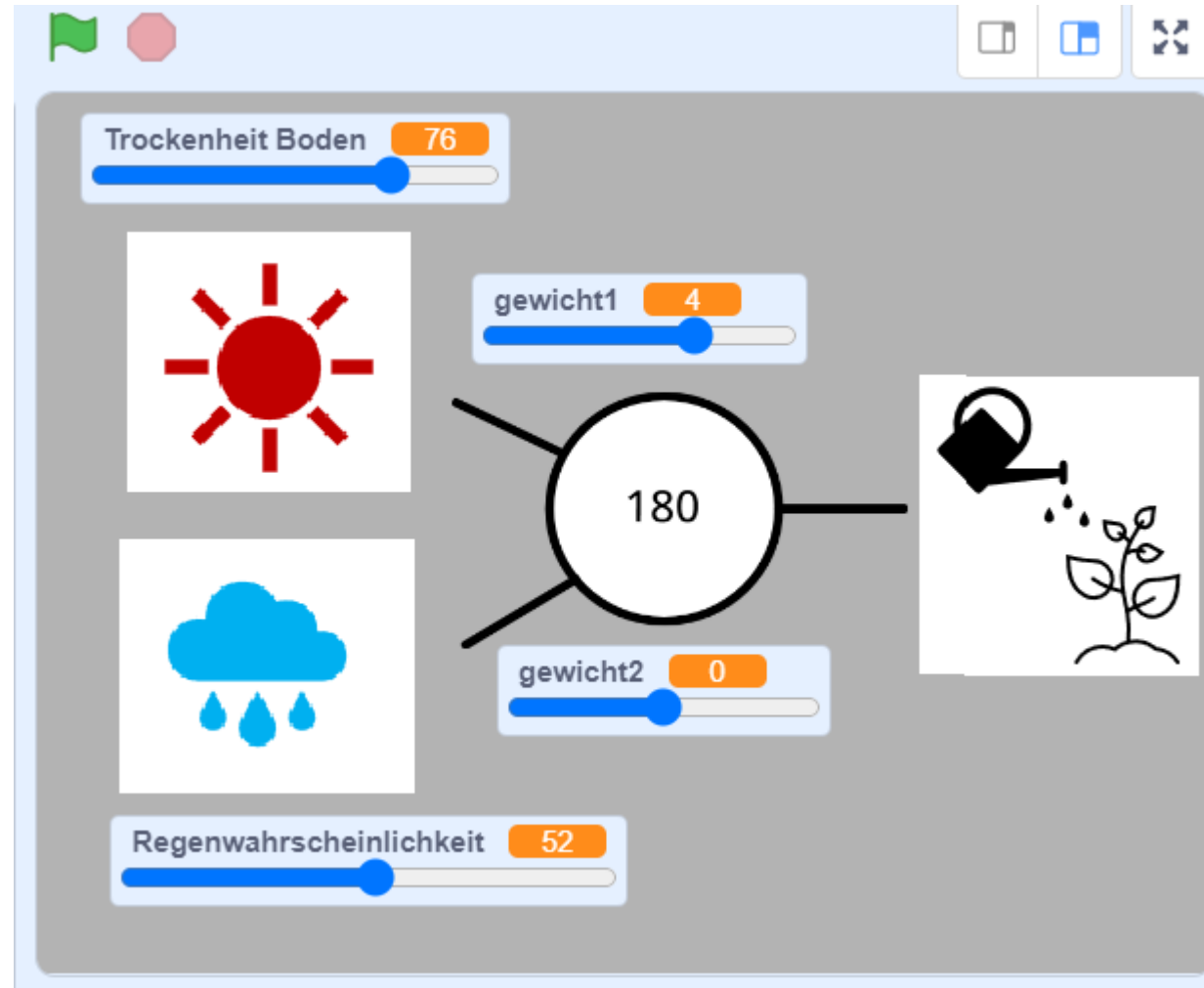
Schwellenwert = 10



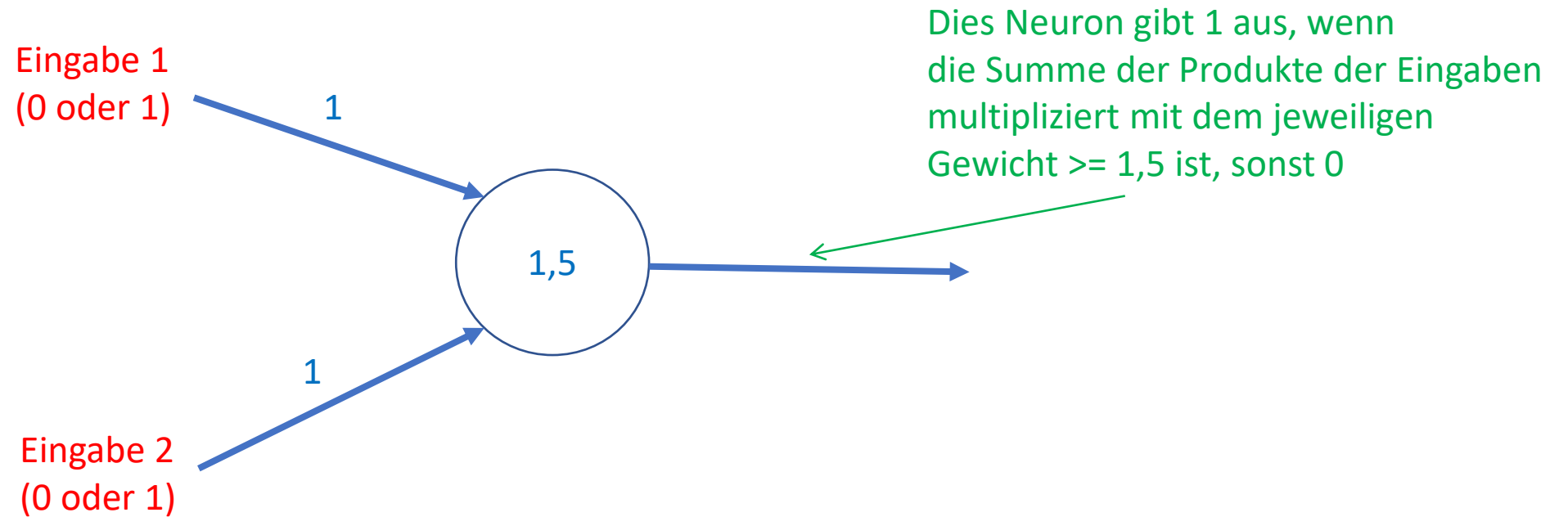
Künstliche Neuronale Netze

- **Gewichtung** an den **Kanten**
 - negatives oder positives Gewicht
- Im Knoten wird ein **Schwellenwert** notiert. Schwellenwert muss überschritten werden, bevor Nervenzelle „feuert“. Prinzip: „Alles oder Nichts“
- Beispiel für eine Ausgabefunktion eines Neurons:
- **Summe aller Produkte aus Eingaben multipliziert mit den zugehörigen Kantengewichten ist größer gleich dem Schwellenwert => Neuron feuert (z.B. Ausgabe 1), sonst nicht (z.B. Ausgabe 0)**

Erkunden der Funktionsweise eines Neurons

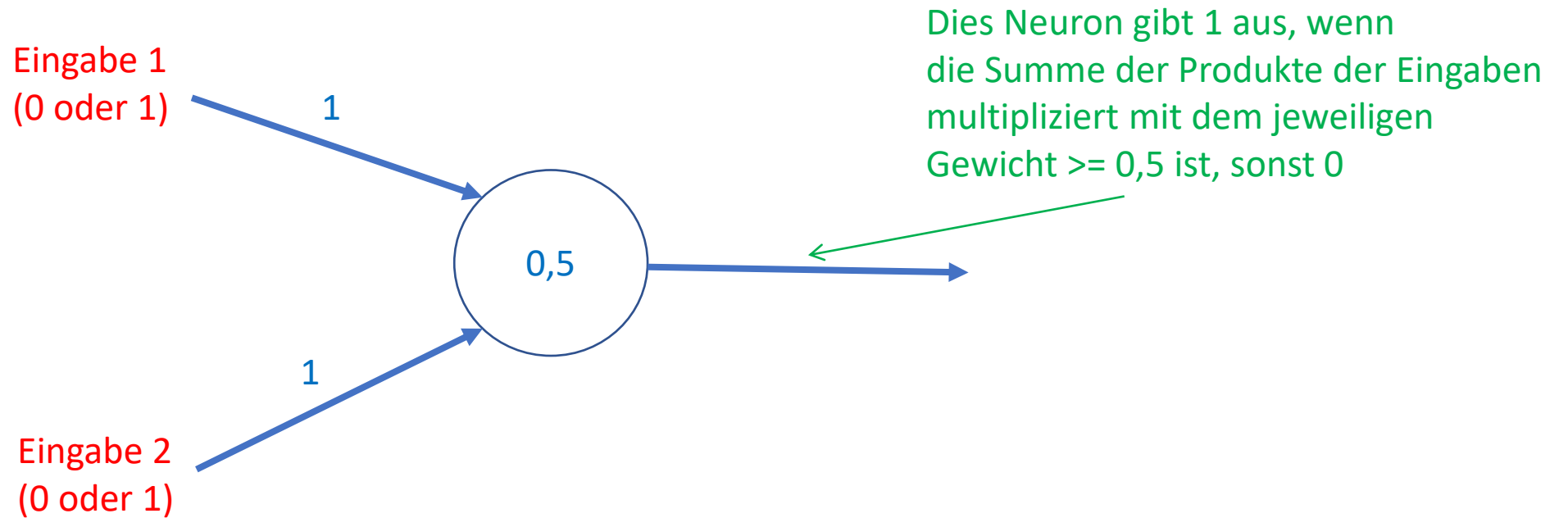


Beispiel UND



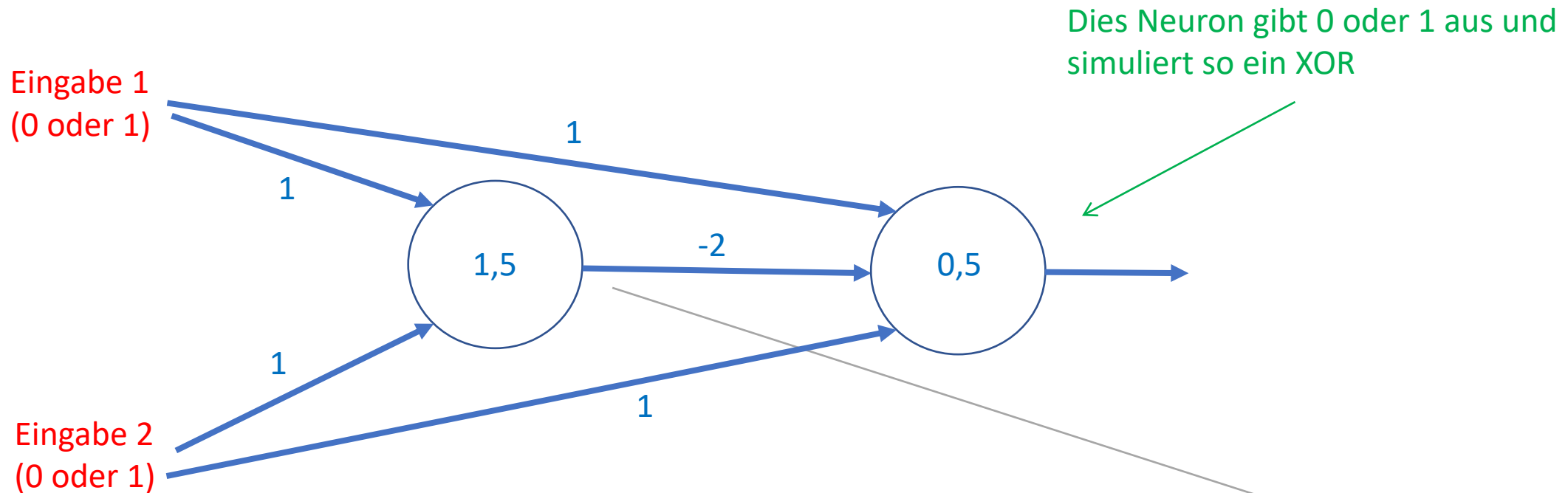
Eingabe1	Eingabe2	Ausgabe
0	0	0
0	1	0
1	0	0
1	1	1

Beispiel ODER



Eingabe1	Eingabe2	Ausgabe
0	0	0
0	1	1
1	0	1
1	1	1

Beispiel XOR



Eingabe 1	Eingabe 2		Ausgabe
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Wie lernt ein Neuronales Netz?

Prinzipiell:

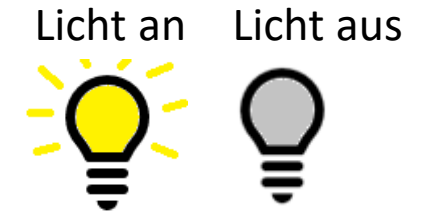
- Entwicklung neuer Verbindungen
- Löschen bestehender Verbindungen
- **Ändern der Gewichtung** (der Gewichte von Neuron i zu Neuron j)
- Anpassen der Schwellenwerte der Neuronen
- Hinzufügen von Neuronen
- Löschen von Neuronen

Didaktische Überlegungen

- Konzentration auf logische Ausdrücke => kleine mehrschichtige Netze, um auch mit Funktionen wie XOR umgehen zu können
- motivierendes Beispielszenario mit Bezug zur Lebenswelt der SuS
- NN sollten so klein sein, dass SuS die Änderung der Gewichte händisch nachvollziehen können
- NN sollten so klein sein, dass SuS sie selbst implementieren, zumindest die Implementierung erkunden können
- NN sollten so didaktisiert sein, dass die Änderung der Gewichte von den SuS selbst überlegt und implementiert werden kann
- Trotzdem soll der Einsatz von NN in diesen Beispielen **sinnvoll** sein (und nicht konstruiert erscheinen) und einen Bezug zur Lebenswelt haben

Der Kontext

- Eine Firma für Smart-Home-Anwendungen hat ein Lichtsystem entwickelt, das automatisch gesteuert wird.
- Je nach Wunsch des Kunden geht das Licht an oder aus, abhängig von den Werten zweier Sensoren.
 - Sensor 1 (Entfernungssensor): hilft zu erkennen, ob eine Person im Raum ist oder nicht
 - Sensor 2 (Lichtsensensor): hilft zu erkennen, ob es Tag oder Nacht ist
- Zusätzlich kann das Licht immer auch manuell an- und ausgeschaltet werden (später mehr dazu)



Sensoren und Aktoren



Eingabe **L** = **1** Tag

Eingabe **L** = **-1** Nacht



Eingabe **S** = **1** Person im Raum

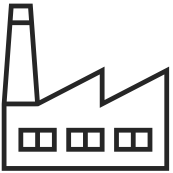


Eingabe **S** = **-1** keine Person im Raum

Ausgabe = **1** Licht an
Ausgabe = **-1** Licht aus



Geschichten und Anforderungen



- In den Büros der Fabrikhalle sollen die Lampen nachts immer leuchten aus Gründen des Einbrecherschutzes, und tagsüber nur, wenn die Mitarbeiter an ihren Plätzen sind.



- Im Bürogebäude der Softwarefirma sollen die Lampen nur nachts leuchten, wenn Mitarbeiter da sind. Tagsüber ist es durch die vielen Fenster immer hell genug.



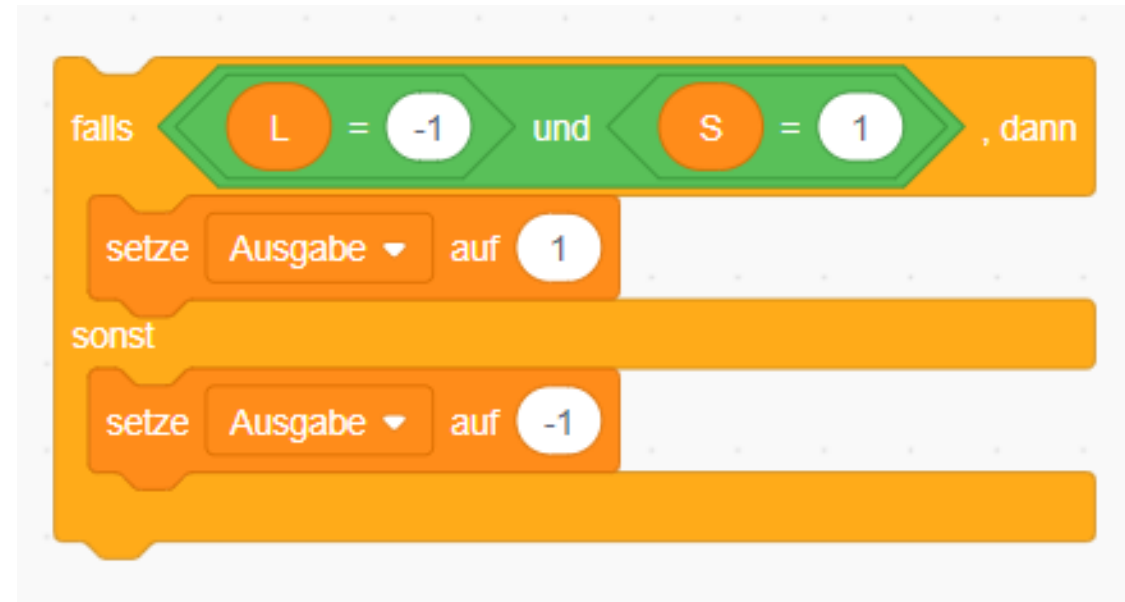
- Im alten Gebäude der Stadtverwaltung müssen die Lampen tagsüber angeschaltet sein, wenn Mitarbeiter da sind, da die Fenster zu wenig Licht hereinlassen. Sollten Mitarbeiter auch nachts arbeiten, müssen auch dann die Lampen eingeschaltet werden. Sonst können sie aus bleiben.



- Im Haus der Familie Schmidt sollen die Lampen tagsüber an sein, wenn jemand zuhause ist, und aus sein, wenn keiner da ist. Nachts sollen die Lampen aus sein, wenn die Bewohner im Haus sind und schlafen, und aus Gründen des Einbrecherschutzes an sein, wenn niemand da ist

Warum Neuronale Netze?

- Für jede Anforderung kann man mit einer Bedingung eine individuelle Lichtsteuerung programmieren.
- Beispiel: Licht geht nur an, wenn es Nacht ist ($L=-1$) und jemand im Raum ist ($S=1$), sonst bleibt das Licht aus:



Kerstin Strecker

Warum Neuronale Netze?

- Wir wollen das System (Lampe, Lichtsensor, Entfernungssensor) jetzt in unterschiedlichsten Gebäuden und bei verschiedensten Firmen einbauen. Aber: jeder Kunde hat andere Anforderungen! Und wir wollen nicht jedes Mal ein neues Programm entwerfen... (vor allem, wenn die Kunden ihre Wünsche ständig ändern)
- Schüler:innen werden mit einer Problemlösestrategie konfrontiert, wo ein und derselbe Algorithmus für verschiedene Anwendungen genutzt wird, indem er durch die Anpassung von Parametern (Gewichten) während der Laufzeit trainiert und damit konfiguriert wird. Damit geht es vor allem auch darum, zu erkennen, wie die Funktionalität des Neuronalen Netzes von den Trainingsdaten abhängt.

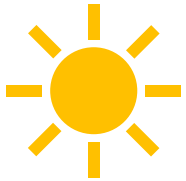
Weitere Beispiele

- Automatische Fenstersteuerung:
- Fenster geht automatisch auf oder zu in Abhängigkeit von Lichtsensorwert und Geräuschsensorwert
- Lichtsensor außen am Haus unterscheidet hell und dunkel
- Geräuschsensor außen am Haus unterscheidet laut und leise
- Zusätzlich kann das Fenster manuell geschlossen oder geöffnet werden

Sensoren und Aktoren

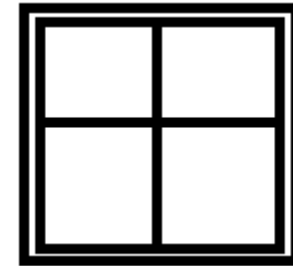
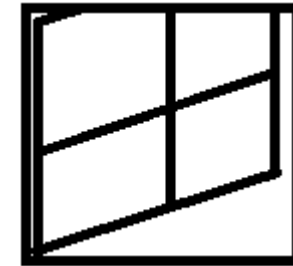


Eingabe S = 1 laut
Eingabe S = -1 leise



Eingabe L = 1 hell
Eingabe L = -1 dunkel

Ausgabe 1: Fenster auf
Ausgabe -1: Fenster zu



Ausgabe 1 (Neuron feuert)
Ausgabe -1 (Neuron feuert nicht)

Geschichten und Anforderungen



Wir betrachten das Haus der Familie Mustermann mit mehreren Schlafzimmern, in denen jeweils dasselbe neue intelligente Fensterschließsystem eingebaut werden soll. Aber die Bewohner unseres Hauses haben ganz unterschiedliche Vorlieben.



Onkel Peter schläft nachts immer bei offenem Fenster, egal wie laut oder leise es draußen ist. Tagsüber hat er sein Fenster aber immer geschlossen, weil er bei der Arbeit ist und ihm ein geschlossenes Fenster sicherer erscheint. Vor allem sollen tagsüber keine Insekten in sein Zimmer fliegen.



Mama Kate schläft nachts auch immer bei offenem Fenster, egal wie laut oder leise es draußen ist. Tagsüber macht sie das Fenster allerdings auch auf, wenn es draußen laut ist, weil dann die Kinder im Garten spielen und Mama Kate sie hören kann. Nur wenn es draußen leise ist, bleibt das Fenster tagsüber zu.

Geschichten und Anforderungen



Oma Emmy hat ganz besondere Eigenarten. Sie macht tagsüber das Fenster auf, wenn es laut ist, damit sie das Spielen ihrer Enkel und die Vögel im Garten hört. An leisen Tagen bleibt das Fenster zu. Nachts hat sie gerne ihre Ruhe. Daher öffnet sie nachts das Fenster nur, wenn es draußen leise ist. Ist es nachts aber laut, dann schließt sie auch nachts das Fenster, weil sie sonst nicht schlafen kann.



Papa Eric hat immer Nachtschicht. Deshalb hat er nachts das Fenster zu, weil er gar nicht im Haus ist. Tagsüber schläft Papa Eric. Dabei macht er das Fenster aber nur auf, wenn es draußen leise ist. Sonst bleibt das Fenster zu, damit er in Ruhe schlafen kann.

Sensoren



Tag:

$L = 1$ entspricht L

Nacht:

$L = -1$ entspricht $\bar{L}$



Anwesend:

$S = 1$ entspricht S

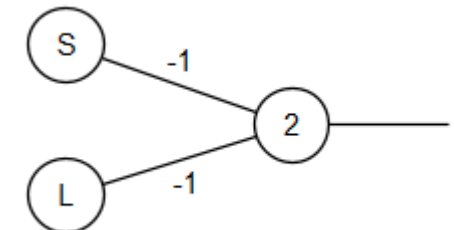
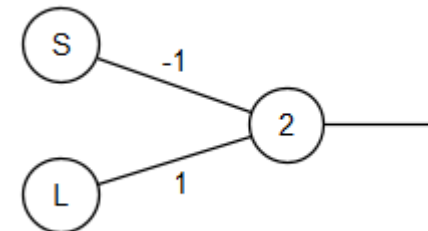
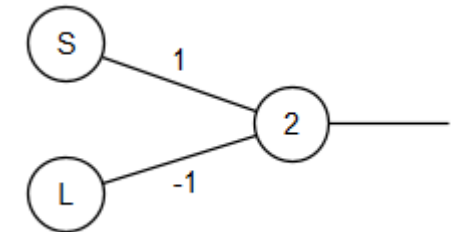
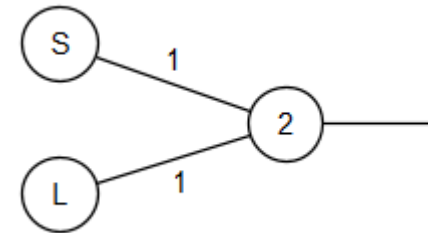
Nicht anwesend:

$S = -1$ entspricht $\bar{S}$

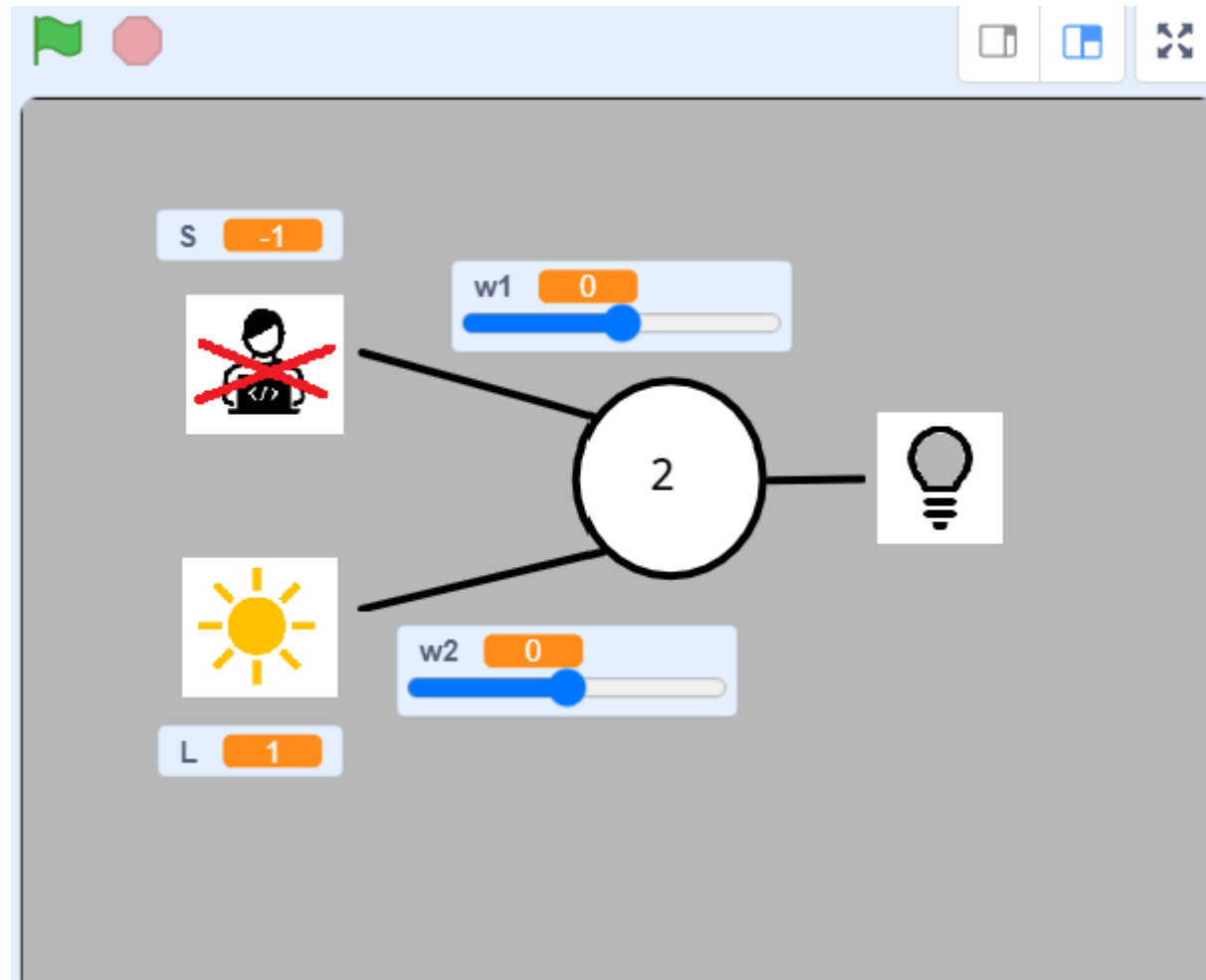


Funktionalität hängt von den Gewichten ab

Anforderung	
Lampe geht an, wenn es Tag ist und eine Person im Raum, sonst aus	$L \wedge S$
Lampe geht an, wenn es Nacht ist und eine Person im Raum, sonst aus	$\bar{L} \wedge S$
Lampe geht an, wenn es Tag ist und keine Person im Raum, sonst aus	$L \wedge \bar{S}$
Lampe geht an, wenn es Nacht ist und keine Person im Raum, sonst aus	$\bar{L} \wedge \bar{S}$

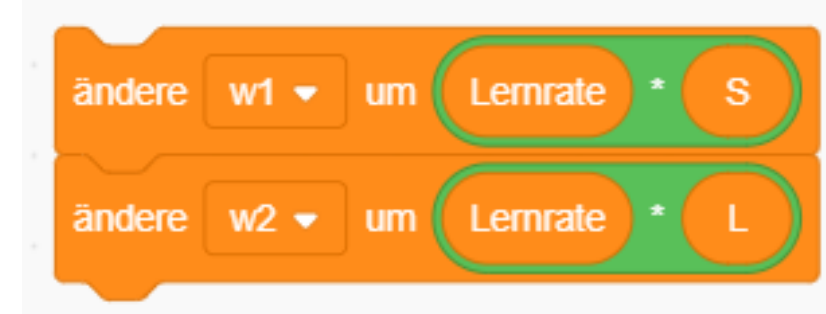


Ein Neuron und feste Gewichte



Ändern der Gewichte bei einem Neuron

- Immer wenn das Neuron nicht feuert, aber hätte feuern sollen
 - $w1 = w1 + \text{Lernrate} * S$
 - $w2 = w2 + \text{Lernrate} * L$

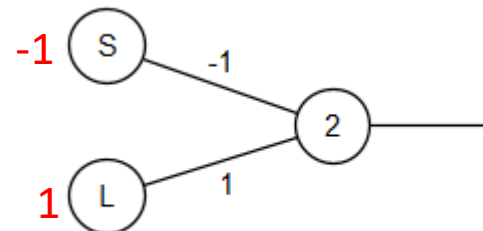
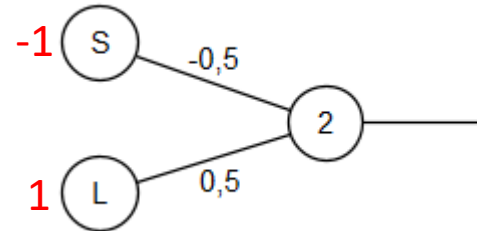
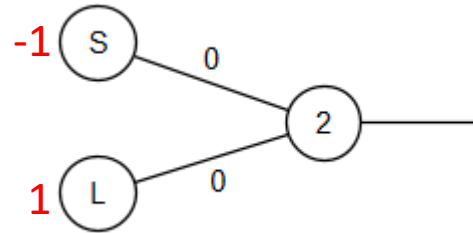


Ändern der Gewichte

Lernrate = 0,5

Aktuelle Gewichte:

$S=-1$, $L=1$



- Neuron feuert nicht, hätte aber feuern sollen=> **Gewichte erhöhen**

- $w1 = w1 + \text{Lernrate} * S$
- $w2 = w2 + \text{Lernrate} * L$

- Neuron feuert nicht, hätte aber feuern sollen=> **Gewichte erhöhen**

- $w1 = w1 + \text{Lernrate} * S$
- $w2 = w2 + \text{Lernrate} * L$

Ändern der Gewichte bei einem Neuron

- Immer wenn das Neuron feuert, aber nicht hätte feuern sollen
 - $w1 = w1 - \text{Lernrate} * S$
 - $w2 = w2 - \text{Lernrate} * L$

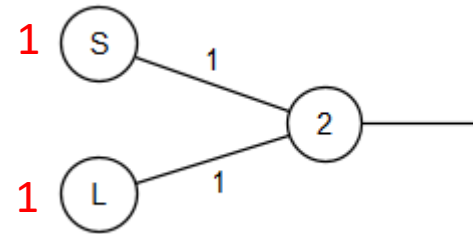


Ändern der Gewichte

Lernrate = 0,5

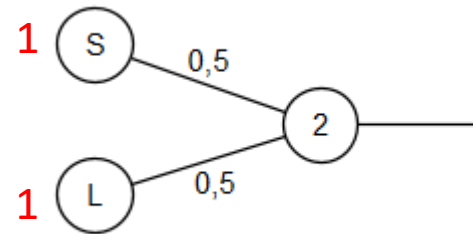
Aktuelle Gewichte:

$S=1, L=1$



- Neuron feuert, hätte aber nicht feuern sollen=> **Gewichte verringern**

- $w1 = w1 - \text{Lernrate} * S$
- $w2 = w2 - \text{Lernrate} * L$



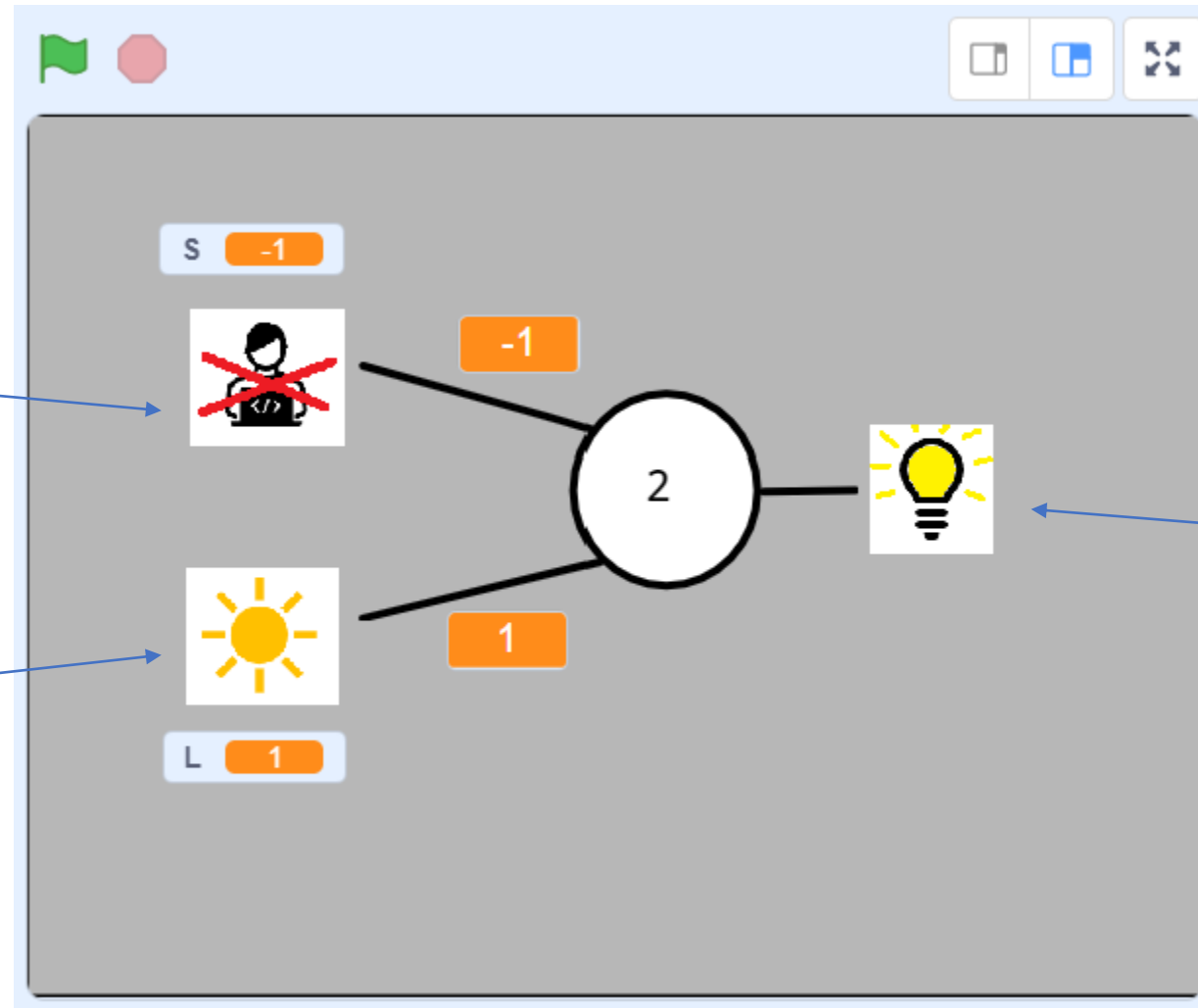
Fehlersignal zum Ändern der Gewichte

- Licht schaltet sich automatisch an oder aus bei bestimmten Werten des Lichtsensors und Entfernungssensors
- Wenn die Gewichte noch nicht optimal sind, funktioniert die automatische Lichtsteuerung nicht wie gewünscht.
- Dann wird das Licht manuell an- oder ausgeschaltet. Das simulieren wir, durch „Klick“ auf die Lampe
 - Licht aus -> „Klick“ auf die Lampe -> Licht an
 - Licht an -> „Klick“ auf die Lampe -> Licht aus
- Wird das Licht manuell gesteuert, kann man daraus schließen, wie die Gewichte verändert werden müssen

Ein Neuron, Gewichte werden automatisch angepasst

Zum Umschalten der Eingabewerte hier klicken

Zum Umschalten der Eingabewerte hier klicken



Fehlersignal: hier klicken

Zwischenergebnis

- Das automatische Lichtsystem wird mit anfangs zufälligen Gewichten eingebaut und schaltet das Licht dementsprechend an oder aus in Abhängigkeit der Werte von Licht- und Entfernungssensor.
- Durch die manuelle Steuerung des Lichts durch den Firmeninhaber, werden die Parameter (Gewichte) in der Folge immer weiter angepasst.
- Schließlich schaltet sich das Licht automatisch an oder aus entsprechend der Wünsche der Firma.
- Dasselbe System verhält sich nach einiger Zeit bei jeder Firma anders.
- ein und derselbe Algorithmus wird für verschiedene Anwendungen genutzt, indem er durch die Anpassung von Parametern trainiert und damit konfiguriert wird.
- Die Trainingsdaten beeinflussen somit die Funktionalität

Erste Reflexionsaufgabe

- Diskriminierende Algorithmen?

Notwendigkeit Neuronaler Netze

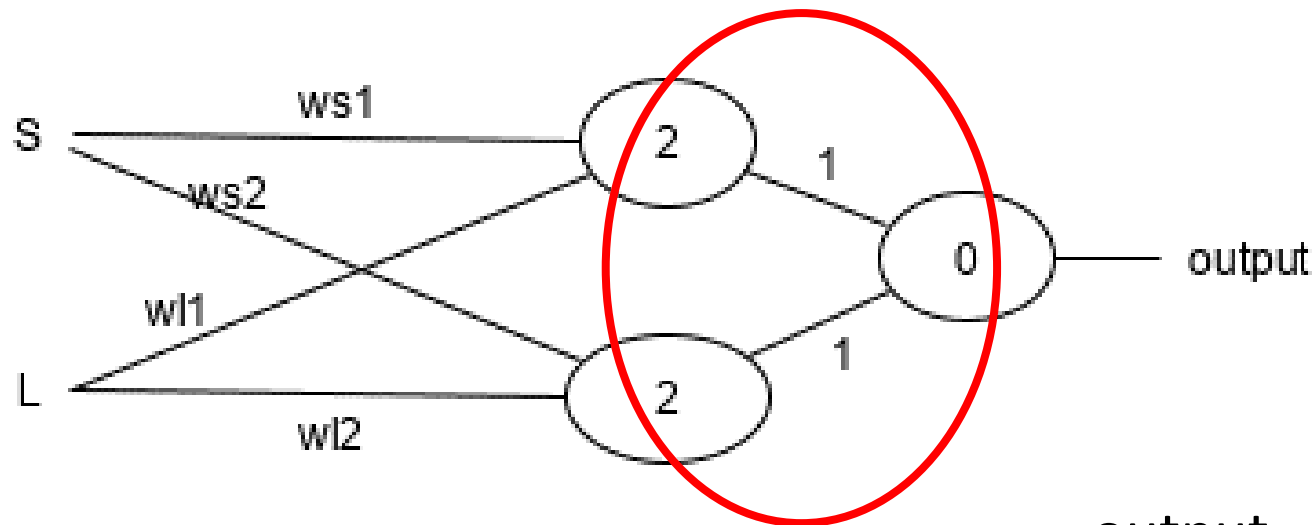


- Im Haus der Familie Schmidt sollen die Lampen tagsüber an sein, wenn jemand zuhause ist und aus sein, wenn keiner da ist. Nachts sollen die Lampen aus sein, wenn die Bewohner im Haus sind und schlafen und aus Gründen des Einbrecherschutzes an sein, wenn niemand da ist

$$(L \wedge S) \vee (\bar{L} \wedge \bar{S})$$

Neuronales Netz

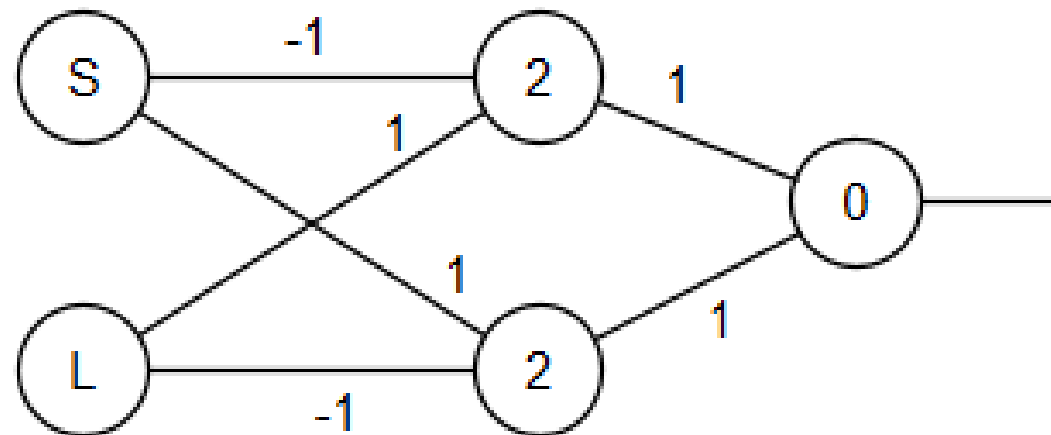
- Wir verwenden im weiteren Verlauf ein Neuronales Netz mit folgender Besonderheit:
 - Die letzte Schicht entspricht einem ODER



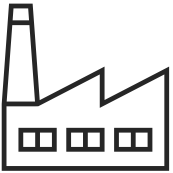
output = 1 => Lampe an
output = -1 => Lampe aus

Aufgabe

- Wann geht das Licht an?



Geschichten und Anforderungen



- In den Büros der Fabrikhalle sollen die Lampen nachts immer leuchten aus Gründen des Einbrecherschutzes, und tagsüber nur, wenn die Mitarbeiter an ihren Plätzen sind.



- Im Bürogebäude der Softwarefirma sollen die Lampen nur nachts leuchten, wenn Mitarbeiter da sind. Tagsüber ist es durch die vielen Fenster immer hell genug.

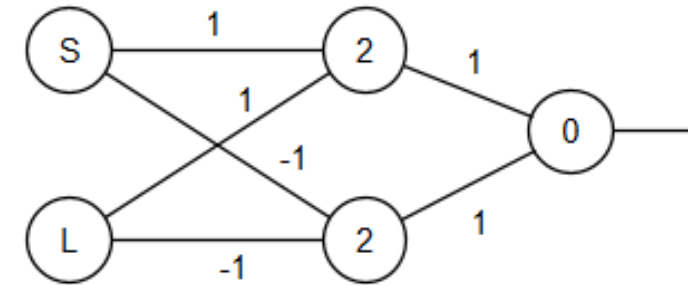
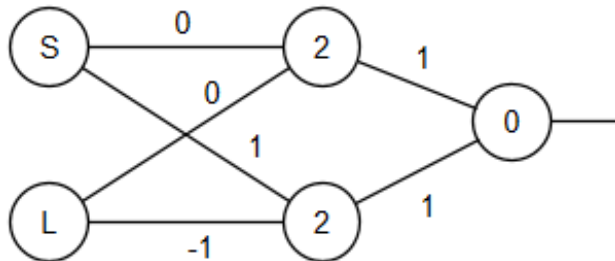
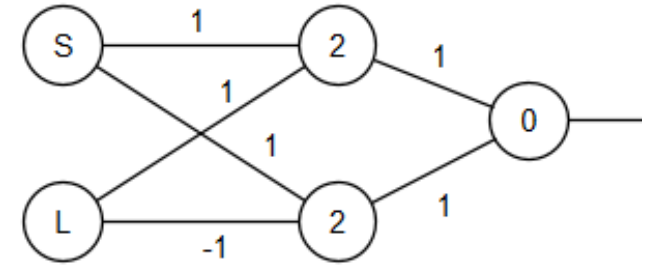
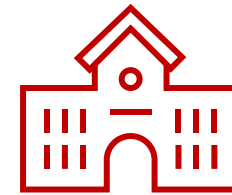
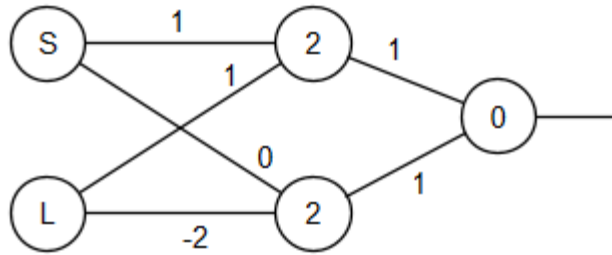
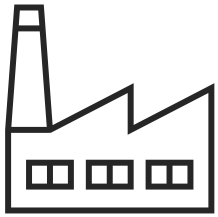


- Im alten Gebäude der Stadtverwaltung müssen die Lampen tagsüber angeschaltet sein, wenn Mitarbeiter da sind, da die Fenster zu wenig Licht hereinlassen. Sollten Mitarbeiter auch nachts arbeiten, müssen auch dann die Lampen eingeschaltet werden. Sonst können sie aus bleiben.

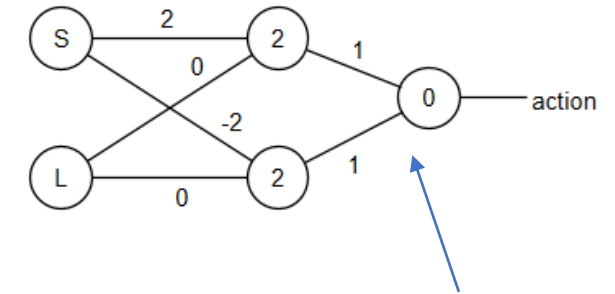


- Im Haus der Familie Schmidt sollen die Lampen tagsüber an sein, wenn jemand zuhause ist, und aus sein, wenn keiner da ist. Nachts sollen die Lampen aus sein, wenn die Bewohner im Haus sind und schlafen, und aus Gründen des Einbrecherschutzes an sein, wenn niemand da ist

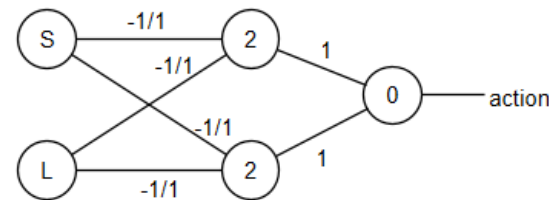
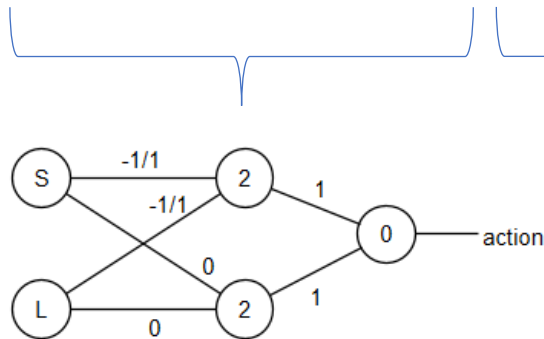
Konfigurationen der Firmeneinhaber



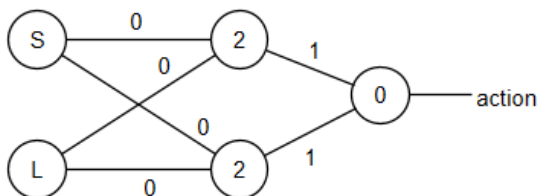
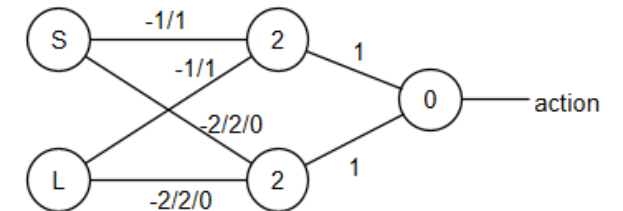
Warum reicht das für alle Ideen?



L	S																
-1	-1	-1	1	-1	-1	-1	1	1	1	-1	-1	-1	-1	1	1	1	1
-1	1	-1	-1	1	-1	-1	1	-1	-1	1	1	-1	1	-1	1	1	1
1	-1	-1	-1	-1	1	-1	-1	1	-1	1	-1	1	1	1	-1	1	1
1	1	-1	-1	-1	-1	1	-1	-1	1	-1	1	1	1	1	1	-1	1

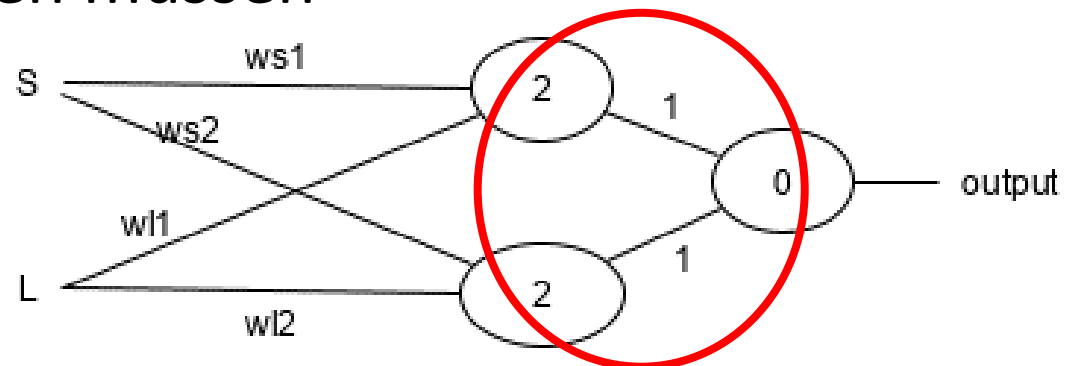


min. zwei Terme können zusammengefasst werden, entweder zu $S, L, \bar{S}$ oder $\bar{L}$



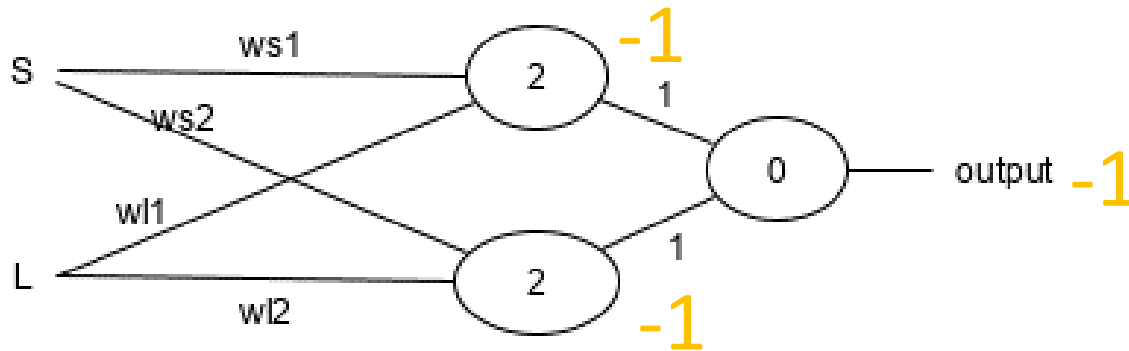
Neuronales Netz

- Besonderheit hier: **konstante** Kantengewichte in der letzten Schicht
- Vorteil: in der Trainingsphase müssen nur die Gewichte der ersten Schicht angepasst werden
- Prozesse wie backpropagation etc. müssen also (noch) nicht betrachtet werden
- die Schüler:innen können selber erarbeiten, wie die Gewichte in der Trainingsphase angepasst werden müssen



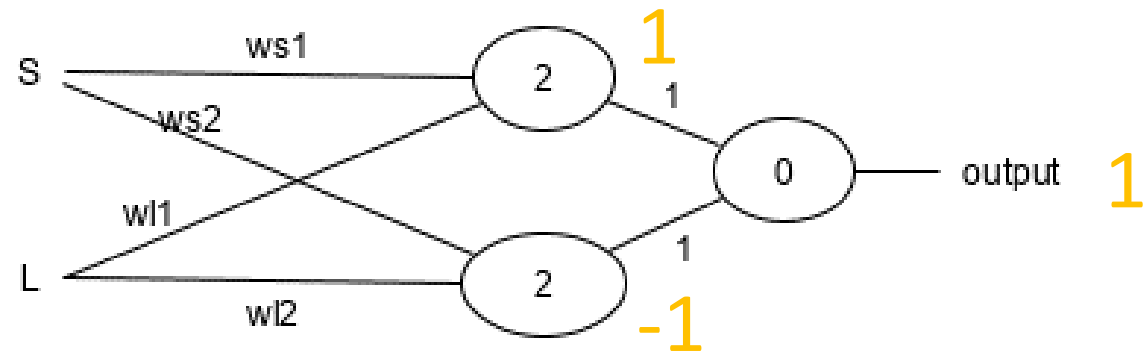
Änderung der Gewichte

- output = -1, gewünschter output = 1
- => beide Neuronen der ersten Schicht ergeben -1
- Für eins der Neuronen müssen die Gewichte der ersten Schicht verändert werden



Änderung der Gewichte

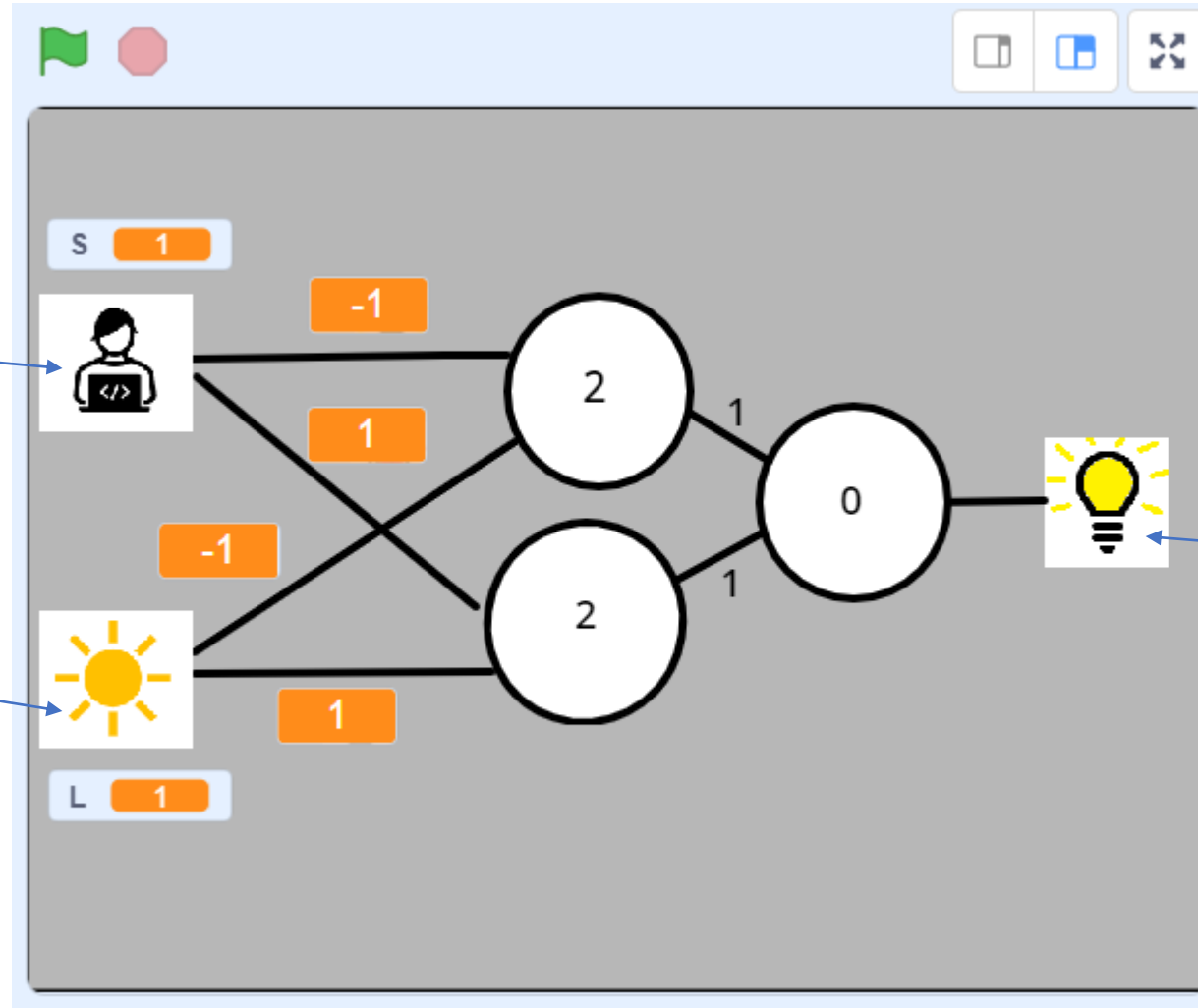
- output = 1, gewünschter output = -1
- => entweder wird das “fehlerhafte” Neuron der Zwischenschicht identifiziert (das, welches 1 ausgibt)
- oder beide Neuronen der ersten Schicht ergeben 1
- => entsprechende Gewichte verändern



Implementierung

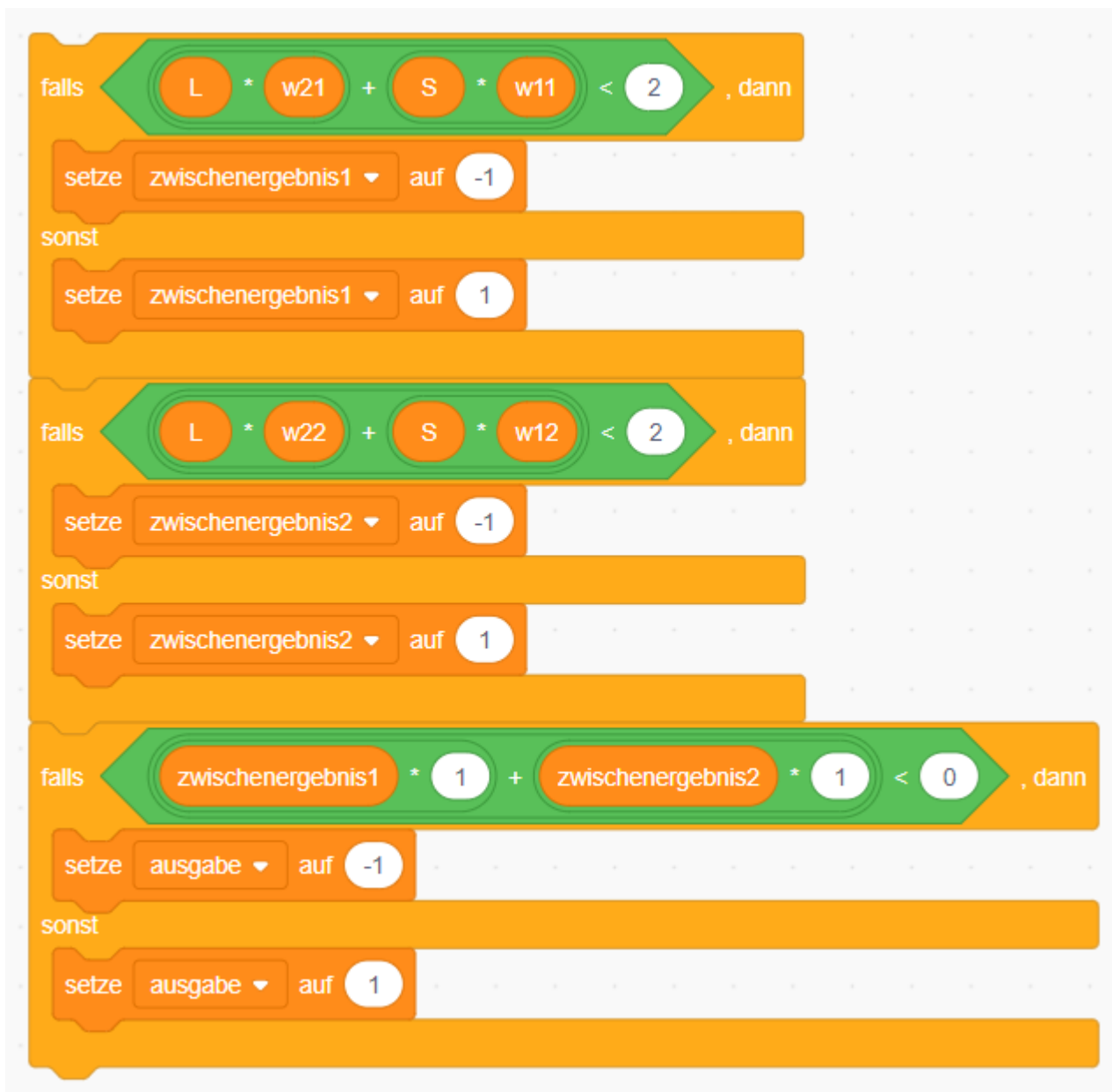
Zum Umschalten der
Eingabewerte hier klicken

Zum Umschalten der
Eingabewerte hier klicken



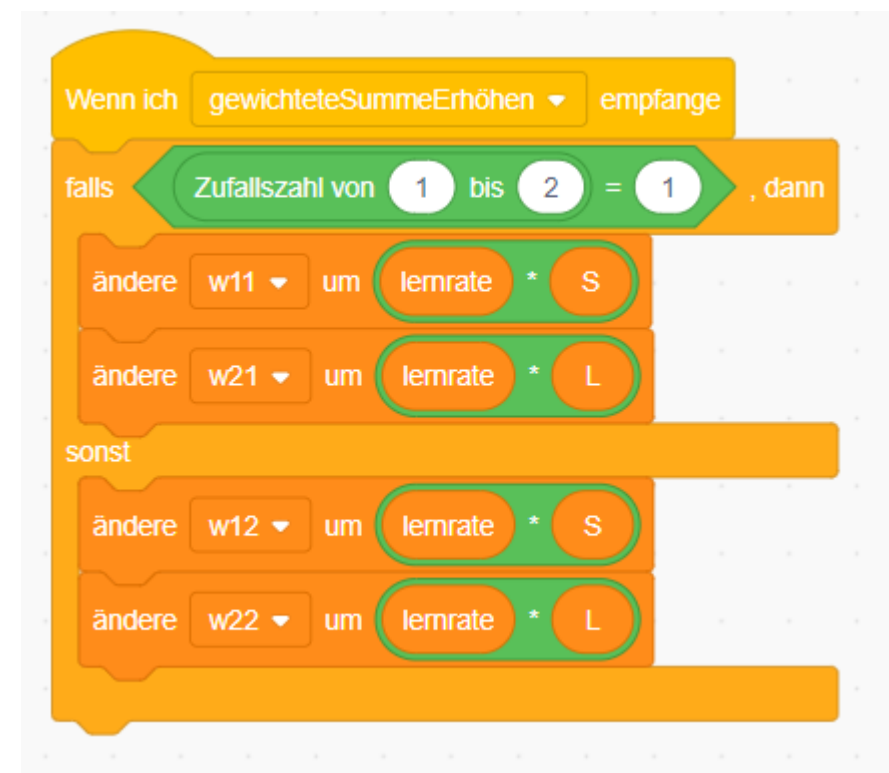
Fehlersignal: hier klicken

Exemplarisches Programm



NETZUM STEUER

Exemplarisches Programm



Exemplarisches Programm



Lernziele

- Die Schüler:innen können die Funktionsweise eines Perzeptrons beschreiben und implementieren und die Grenzen eines Perzeptrons benennen
- Die Schüler:innen können die Funktionsweise eines kleinen mehrstufigen Netzes beschreiben und implementieren (Besonderheit: feste Kantengewichte in der letzten Schicht)
- Sie erkennen, dass es sich bei der Implementierung Neuronaler Netze um einen deterministischen Algorithmus handelt, wie sie es gewohnt sind.
- Die Schüler:innen können die Anpassung der Gewichte eines Neuronalen Netzes durch ein Fehlersignal per Hand nachvollziehen und reflektieren

Lernziele

- Sie erkennen, dass durch die Trainingsphase bestimmte Parameter des Algorithmus während der Laufzeit angepasst werden und so der Algorithmus unterschiedlich konfiguriert werden kann
- Sie erkennen, dass die Parameteranpassung (je nach Feedback) unterschiedlich ausfallen kann und damit auch die zukünftigen Ergebnisse des Algorithmus entsprechend der Parametereinstellungen ausfallen
- Die Algorithmen erreichen keine Ebene des „Bewusstseins“ oder des „Verstehens“

Besonderheit: konstante Gewichte in der letzten Schicht

- Offene Fragen:
 - Wenn die letzte Schicht keine festen Kantengewichte hat, wie wird dann mit dem Fehler umgegangen?
 - Hier kann der Fehler ja gar nicht benannt werden
- Fehler des Neuronalen Netzes ist Funktion der Kantengewichte
- Fehler verringern
- Iteratives Gradientenverfahren, um das Minimum zu ermitteln (den Fehler so weit wie möglich minimieren)

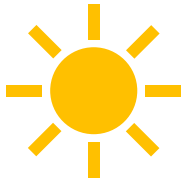
Weitere Beispiele

- Automatische Fenstersteuerung:
- Fenster geht automatisch auf oder zu in Abhängigkeit von Lichtsensorwert und Geräuschsensurwert
- Lichtsensor außen am Haus unterscheidet hell und dunkel
- Geräuschsensur außen am Haus unterscheidet laut und leise
- Zusätzlich kann das Fenster manuell geschlossen oder geöffnet werden

Sensoren und Aktoren

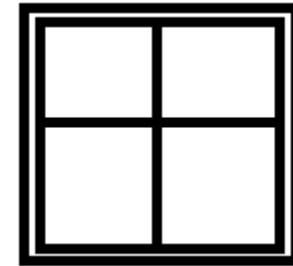
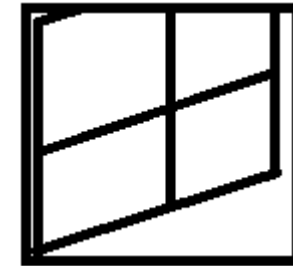


Eingabe S = 1 laut
Eingabe S = -1 leise



Eingabe L = 1 hell
Eingabe L = -1 dunkel

Ausgabe 1: Fenster auf
Ausgabe -1: Fenster zu



Ausgabe 1 (Neuron feuert)
Ausgabe -1 (Neuron feuert nicht)

Geschichten und Anforderungen



Wir betrachten das Haus der Familie Mustermann mit mehreren Schlafzimmern, in denen jeweils dasselbe neue intelligente Fensterschließsystem eingebaut werden soll. Aber die Bewohner unseres Hauses haben ganz unterschiedliche Vorlieben.



Onkel Peter schläft nachts immer bei offenem Fenster, egal wie laut oder leise es draußen ist. Tagsüber hat er sein Fenster aber immer geschlossen, weil er bei der Arbeit ist und ihm ein geschlossenes Fenster sicherer erscheint. Vor allem sollen tagsüber keine Insekten in sein Zimmer fliegen.



Mama Kate schläft nachts auch immer bei offenem Fenster, egal wie laut oder leise es draußen ist. Tagsüber macht sie das Fenster allerdings auch auf, wenn es draußen laut ist, weil dann die Kinder im Garten spielen und Mama Kate sie hören kann. Nur wenn es draußen leise ist, bleibt das Fenster tagsüber zu.

Geschichten und Anforderungen



Oma Emmy hat ganz besondere Eigenarten. Sie macht tagsüber das Fenster auf, wenn es laut ist, damit sie das Spielen ihrer Enkel und die Vögel im Garten hört. An leisen Tagen bleibt das Fenster zu. Nachts hat sie gerne ihre Ruhe. Daher öffnet sie nachts das Fenster nur, wenn es draußen leise ist. Ist es nachts aber laut, dann schließt sie auch nachts das Fenster, weil sie sonst nicht schlafen kann.

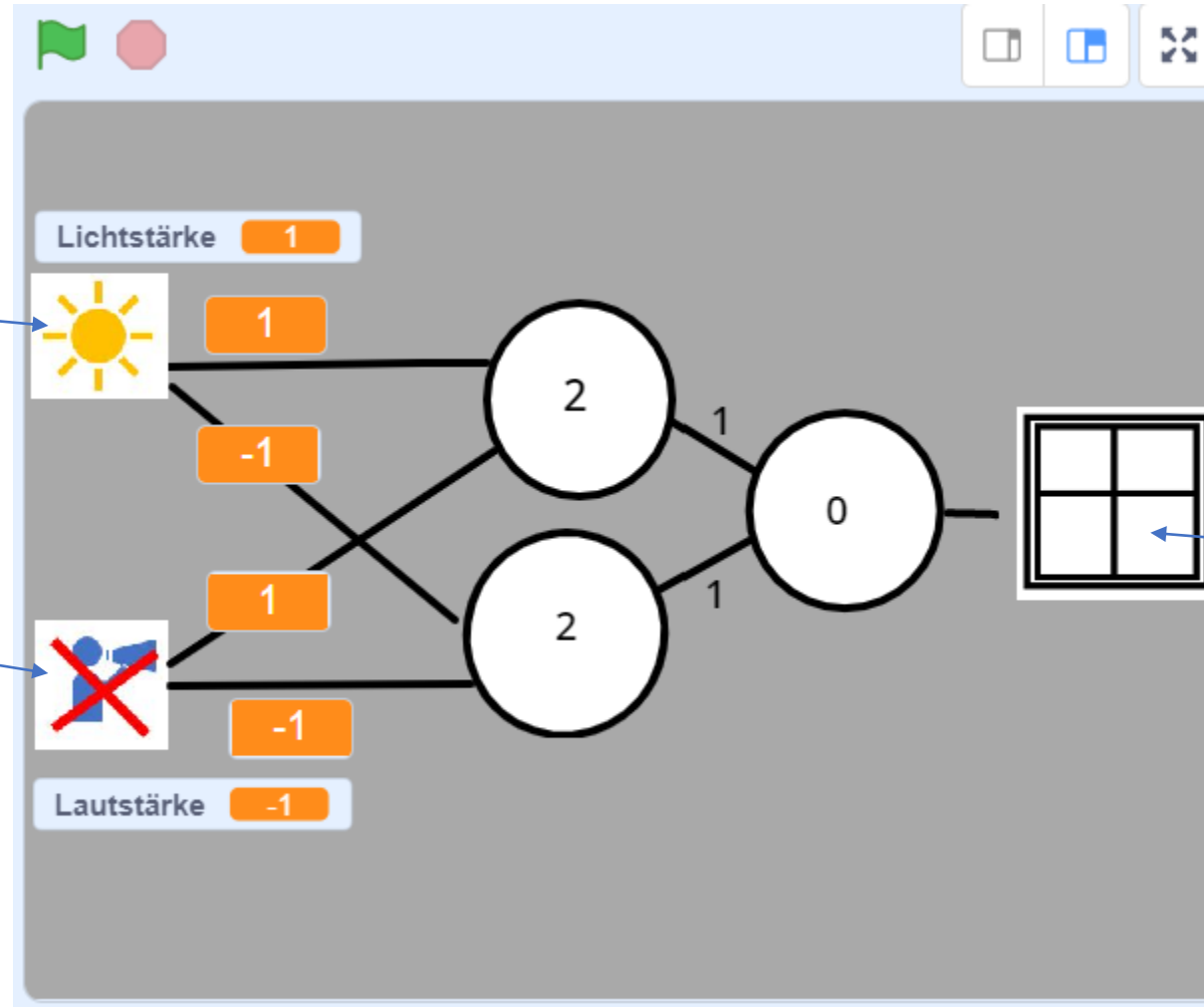


Papa Eric hat immer Nachtschicht. Deshalb hat er nachts das Fenster zu, weil er gar nicht im Haus ist. Tagsüber schläft Papa Eric. Dabei macht er das Fenster aber nur auf, wenn es draußen leise ist. Sonst bleibt das Fenster zu, damit er in Ruhe schlafen kann.

Implementierung

Zum Umschalten der
Eingabewerte hier klicken

Zum Umschalten der
Eingabewerte hier klicken



Fehlersignal: hier klicken

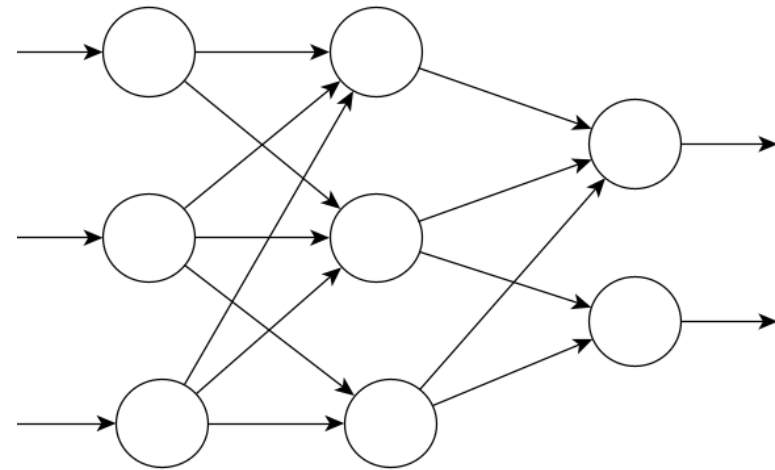
Weitere Beispiele

- Fenster öffnet und schließt sich automatisch, abhängig von Temperatur (warm oder kalt) und Tageszeit (Tag oder Nacht)
- Garagentor öffnet und schließt sich automatisch, abhängig von Entfernungssensor (Auto davor oder nicht) und Fernbedienung (gedrückt oder nicht)
- Pakete werden Lieferung 1 oder Lieferung 2 zugeordnet, abhängig von der Größe (groß und sperrig oder klein) und danach, ob sie einen prime-Aufkleber haben oder nicht.
- ...

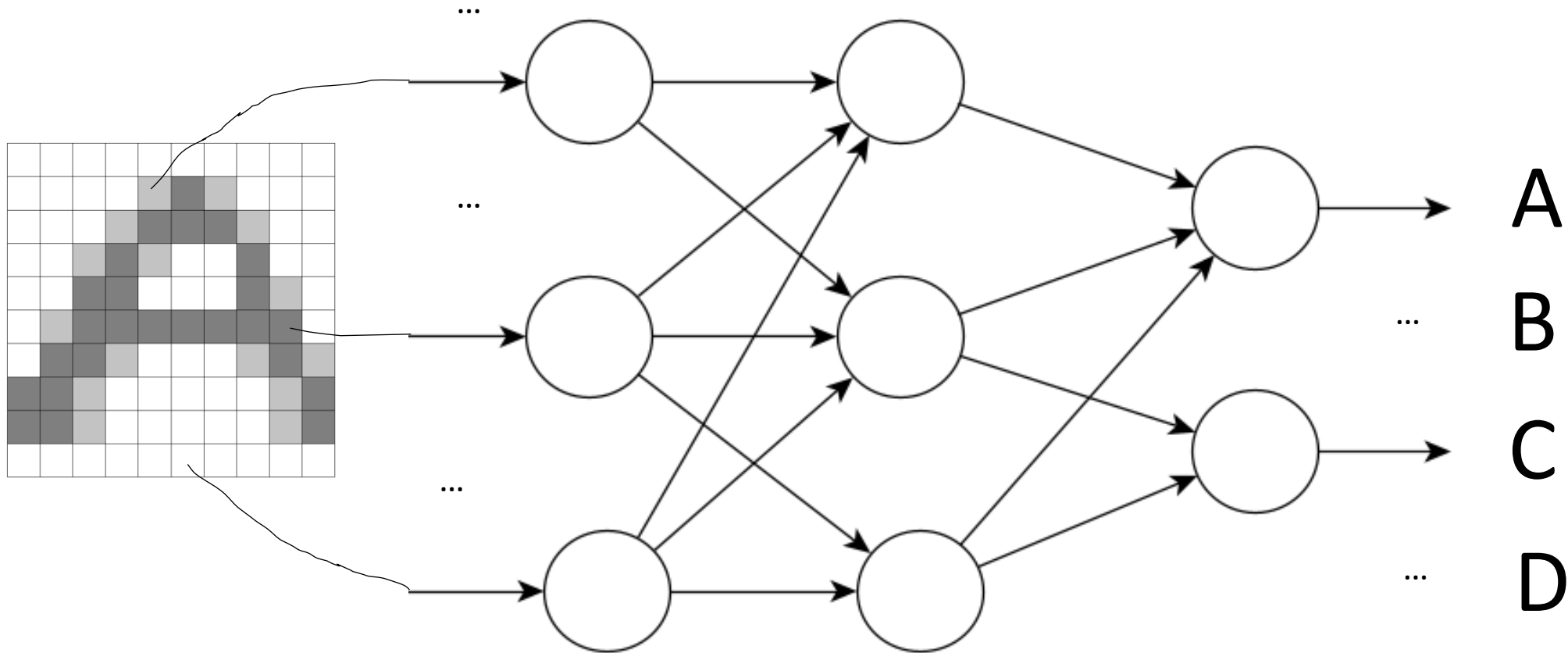
Neuronale Netze - Reflexion

Neuronale Netze allgemein

- **Trainingsphase** und **Anwendungsphase**
- Neuronale Netze liefern Vorhersagen für **unbekannte** Daten, mit denen sie in der Trainingsphase nicht trainiert worden sind.
- Gewichte an allen Kanten zwischen Neuronen



Neuronale Netze und Vorhersagen - Schrifterkennung



Reflexion über Neuronale Netze

- Was genau lernt eigentlich ein Neuronales Netz?
- Beispiel: Katze oder Elefant?

Reflexion über Neuronale Netze

- Neuronale Netze mit mehreren 100 Millionen Neuronen => keine Nachvollziehbarkeit der Ergebnisse
 - => Ethische Dilemma
- Treffsicherheit Neuronaler Netze
 - Aufgaben für die Neuronale Netze trainiert wurden, lösen sie meist sehr gut.
 - Ergebnisse können im individuellen Einzelfall falsch sein
- Chancen Neuronaler Netze
 - Mustererkennung
- Der Begriff der „künstlichen Intelligenz“

Vielen Dank für Ihre
Aufmerksamkeit!