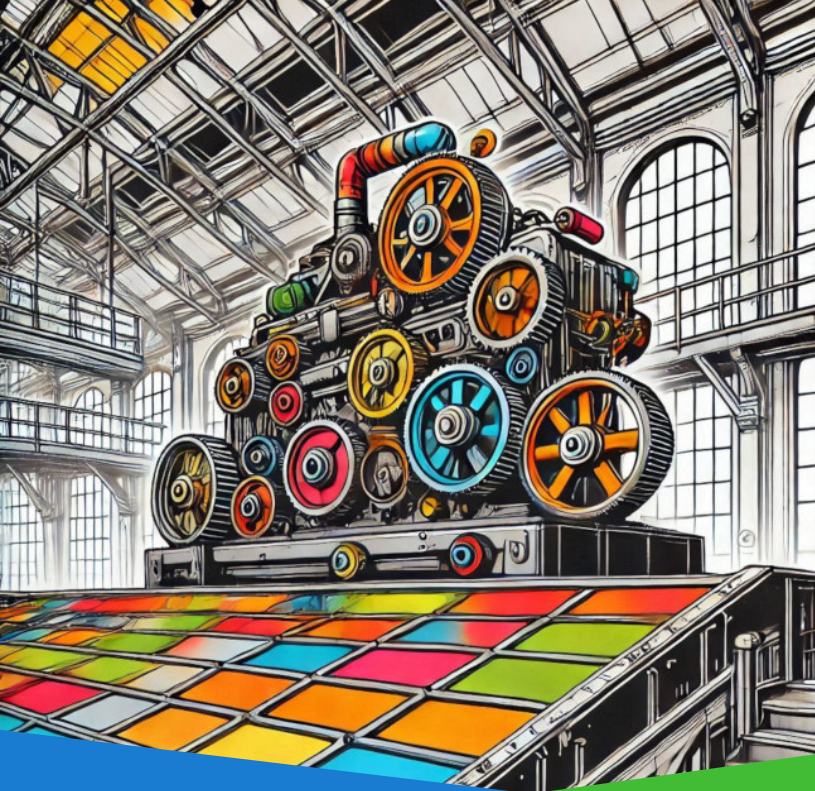


KI Entdecker Quest

## Die Rettung der Strichwelten

Neuralis in Gefahr! Systemfehler 0815 hat ein neuronales Netz zerstört und ihr seid auserwählt, es wieder aufzubauen. Eure Mission: Taucht Schritt für Schritt ein in die Welt der Künstlichen Intelligenz und entschlüsselt die Geheimnisse neuronaler Netze. Könnt ihr Neuralis retten?



# Impressum



taralino e.V.

Braunschweig

Erarbeitet von:

Dr. Daniel Scholz

daniel@taralino.de

Stand des vorliegenden Dokuments:

28. März 2025

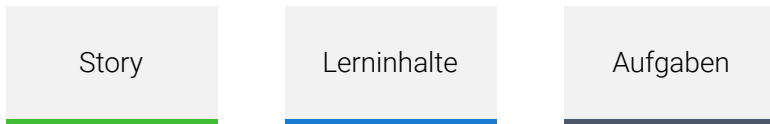


Dieses Werk ist lizenziert unter der Lizenz CC BY-SA 4.0. Das Material darf damit für beliebige Zwecke vervielfältigt und weiterverbreitet werden, auch im kommerziellen Kontext.

## Hinweise zur Einordnung

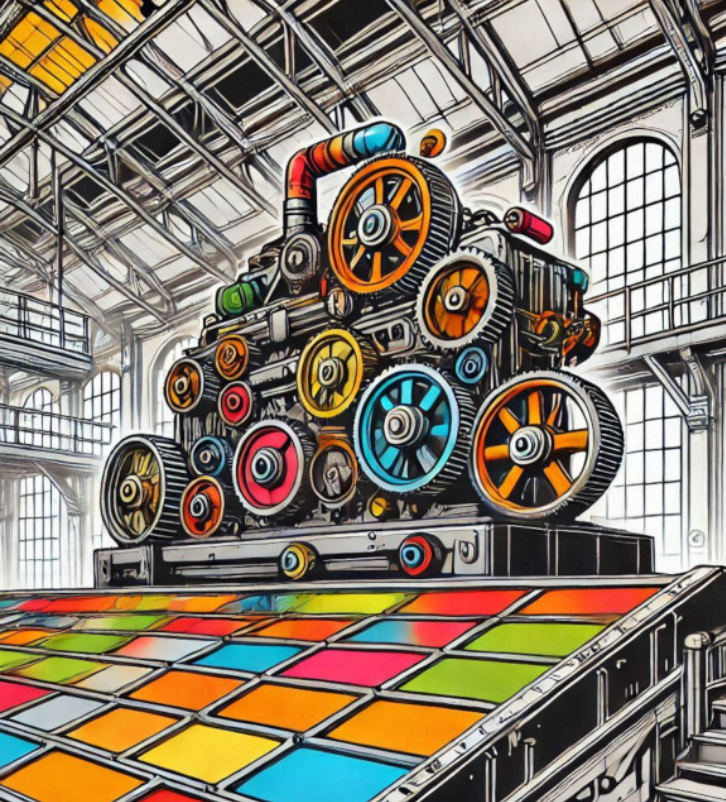


- Unsere KI Entdecker Quests wurde entwickelt, um grundlegende Verfahren der Künstlichen Intelligenz spielerisch, niederschwellig und ebenso unterhaltsam zu vermitteln.
- Während der Bearbeitung wird an ausgewählten Stellen eine Internetverbindung benötigt, um interaktive Anwendungen zu erproben.
- Die einzelnen Seiten sind klar aufgeteilt in begleitende Story, Lerninhalte und Aufgaben, was auch visuell hervorgehoben wird:



- Die Texte und Bilder zur begleitenden Story wurden mit Hilfe generativer KI erstellt.

Und nun: Los geht's!



## Die Herausforderung



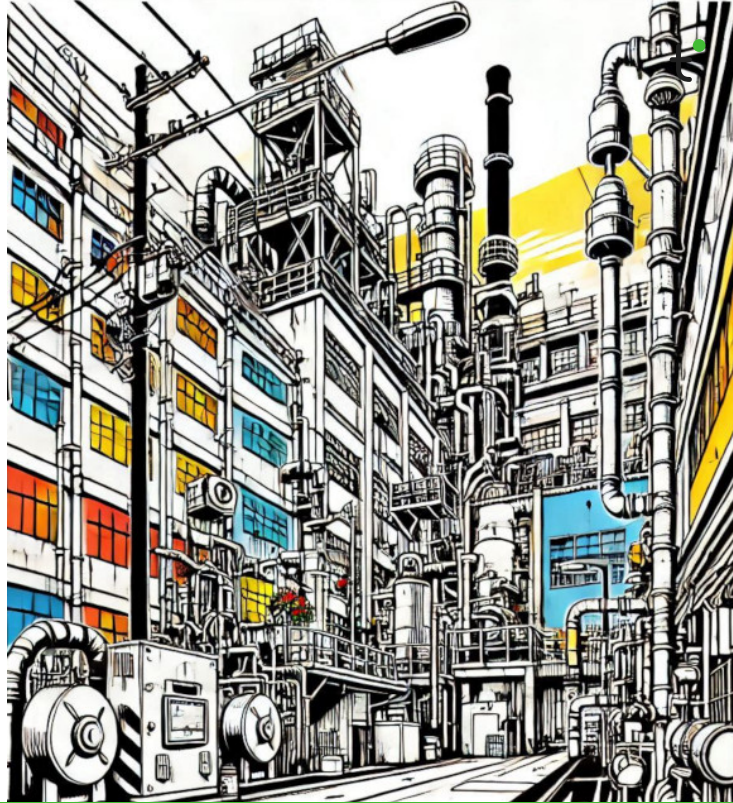
In einer futuristischen Welt wird die Menschheit von einer allgegenwärtigen Künstlichen Intelligenz namens Neuralis unterstützt, die Strichzeichnungen klassifizieren kann.

Das System wird in zahlreichen Lebensbereichen eingesetzt: Es hilft Ärzten, Röntgenaufnahmen zu analysieren, es unterstützt selbstfahrende Autos darin, Straßenmarkierungen und Hindernisse zu erkennen, und vieles mehr.

## Die Herausforderung

Doch ein Stromausfall hat dazu geführt, dass Neuralis nicht mehr funktioniert - ein fataler Ausfall, der katastrophale Folgen haben könnte.

Die ursprünglichen Entwickler sind spurlos verschwunden und niemand weiß, wie man das System wieder in Gang bringen kann.





## Die Herausforderung



Doch es gibt Hoffnung:

Dein Team und du, ihr seid eine Elitegruppe, die mit der geheimen Mission beauftragt wurde, Neuralis zu reparieren.

Begeht euch auf die Reise und trainiert Neuralis, bevor die Zeit abläuft. Ihr müsst eine Reihe von Aufgaben lösen, die euch durch die Funktionsweise neuronaler Netze führen.

## Bibliothek des Wissens

Obwohl ihr anfangs noch ziemlich ahnungslos seid, nehmt ihr die Mission gerne an.

Ihr begeben euch in die Bibliothek von Arcaria, in der nicht nur Bücher zu finden sind. Auch Datenträger und technische Zeichnungen sind dort archiviert.

Ihr findet sogar alte Entwicklungsunterlagen von Neuralis, die ihr aber leider noch nicht entschlüsseln könnt. Sicher ist aber, dass euch die Unterlagen noch nützlich sein werden...





## Bibliothek des Wissens



Ihr fangt ganz vorne an und habt die Idee, euch am Vorbild der Natur zur orientieren.

In der stillen, endlosen Weite der Bibliothek entdeckt ihr ein altes verstaubtes Biologiebuch. Ihr blättert ein wenig, bis euch ein Kapitel über neuronale Netze ins Auge fällt.

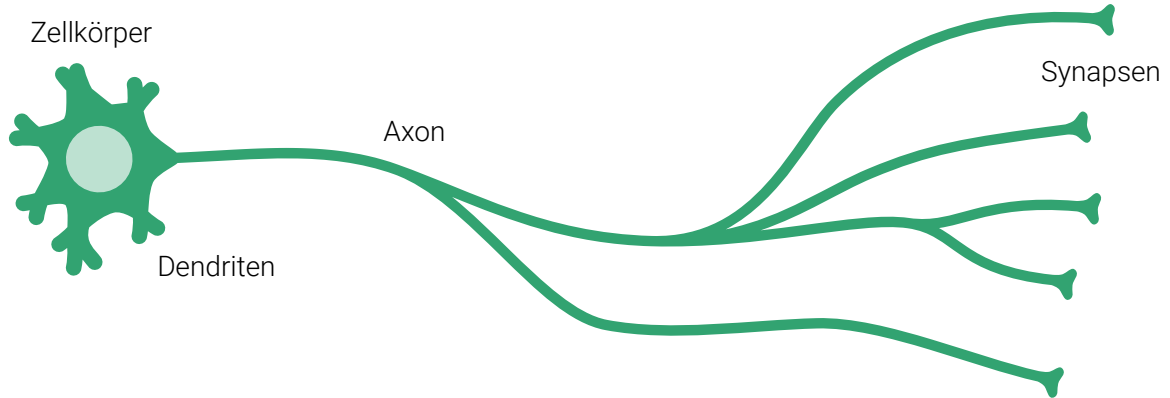
Wissbegierig könnt ihr es gar nicht abwarten, euch einzulesen.



# Neuronen



Das Nervensystem besteht aus unzähligen Nervenzellen, die auch als Neuronen bezeichnet werden. Nervenzellen sind auf die Weiterleitung einer elektrischen Erregung spezialisiert

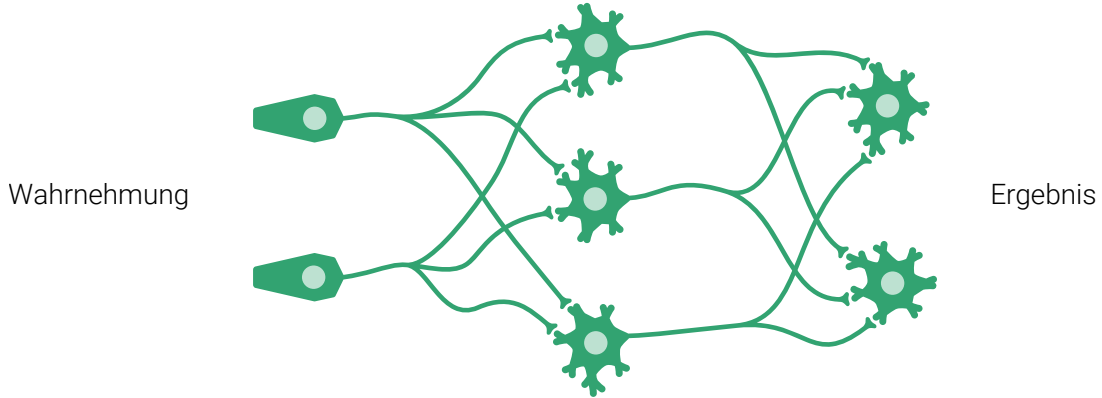


Genauer gesagt können Nervenzellen über ihre Dendriten aktiviert werden. Falls dies der Fall ist, so wird über das Axon ein elektrischer Impuls an alle Synapsen der Zelle übertragen.

# Neuronales Netz



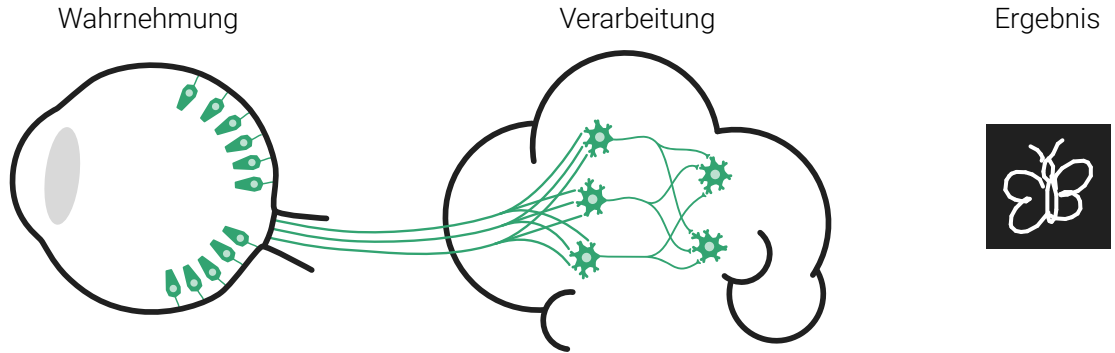
Die Synapsen einzelner Nervenzellen können mit den Dendriten nachfolgender Nervenzellen kommunizieren. Dadurch bildet sich ein großes Netzwerk, das neuronale Netz.



Im Netzwerk ganz links befinden sich Eingangsneuronen, die der Wahrnehmung dienen. Die Informationen werden im Gehirn verarbeitet und als abstraktes Ergebnis zusammengefasst.

## Beispiel Auge

Stäbchen- und Zapfenzellen sind Neuronen in der Netzhaut unserer Augen, welche der Wahrnehmung von Helligkeit und Farben dienen. Je nachdem, was wir gerade sehen, sind unterschiedliche dieser Neuronen aktiviert bzw. nicht aktiviert.



Die Informationen bzw. Zustände der Eingangsneuronen werden im Gehirn verarbeitet. Das Ergebnis kann beispielsweise sein, dass ein Schmetterling gesehen wird.



## Die Euphorie

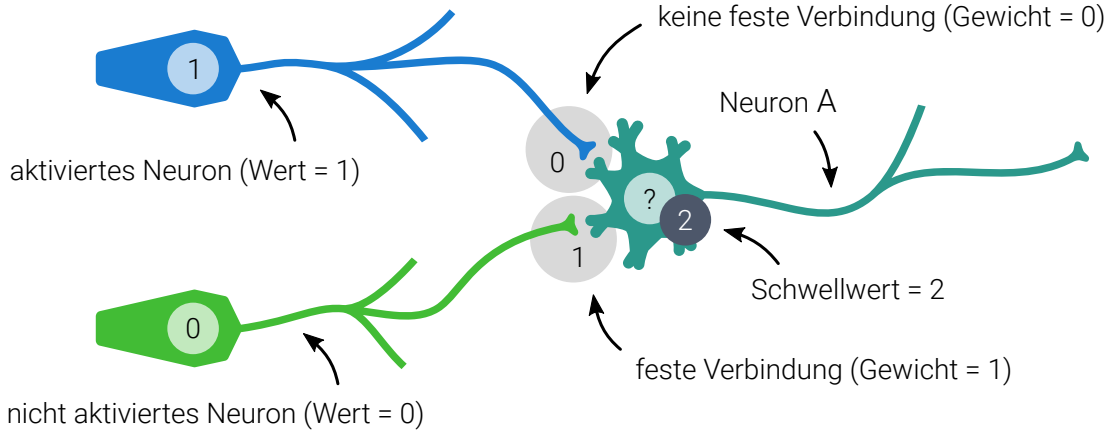


Begeistert von den neuen Erkenntnissen fühlt ihr euch unschlagbar - das Zusammenspiel der neuronalen Netze scheint plötzlich glasklar.

Mit funkelnden Augen macht ihr euch daran, das Wissen in Algorithmen zu gießen, überzeugt, dass der Weg zur Reparatur von Neuralis nun ein Kinderspiel ist.

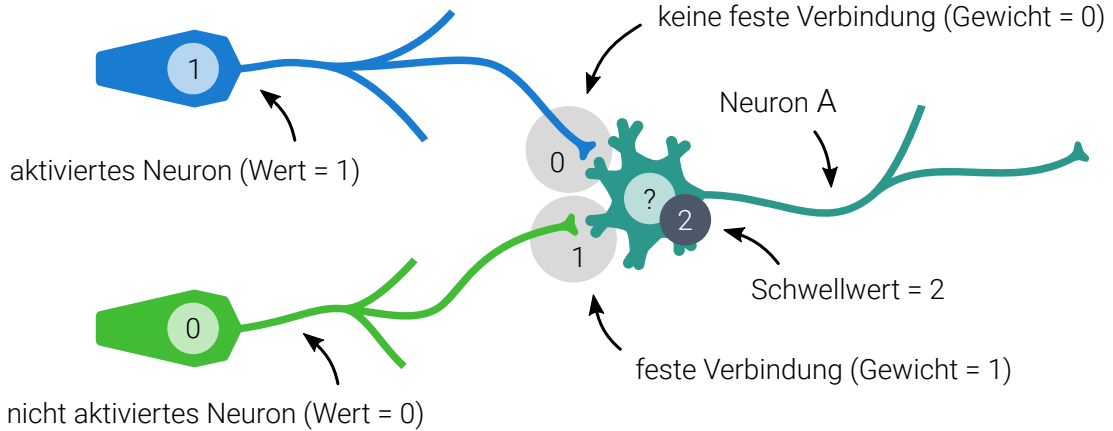
Ihr beginnt direkt mit der Arbeit und stellt ein erstes Modell auf.

# Modell eines Perzeptrons



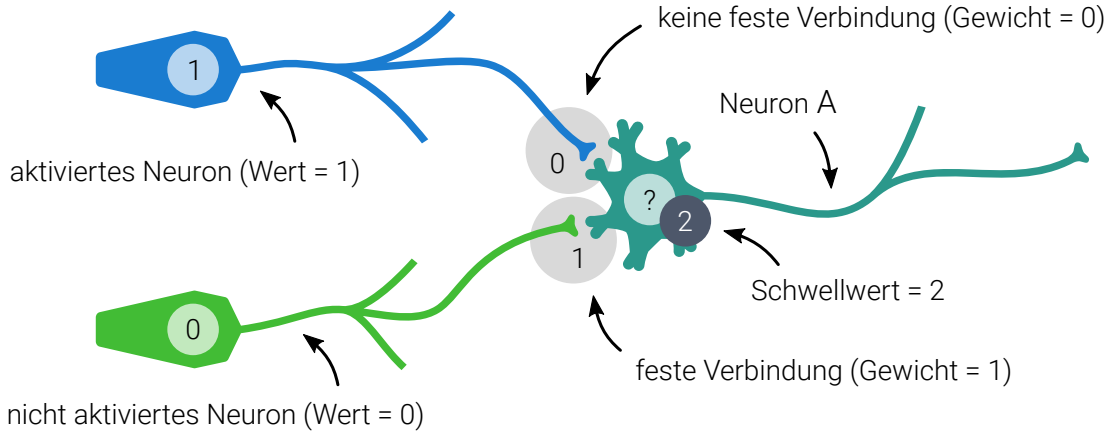
Neuronen, die aktiviert sind, erhalten den Wert 1 (und werden in blau dargestellt). Neuronen, die nicht aktiviert sind, erhalten den Wert 0 (und werden in grün dargestellt).

# Modell eines Perzeptrons



Zudem werden Gewichte zwischen je zwei Neuronen definiert: Ein Gewicht von 1 bedeutet, dass eine Verbindung besteht. Ein Gewicht von 0 bedeutet, dass keine Verbindung besteht.

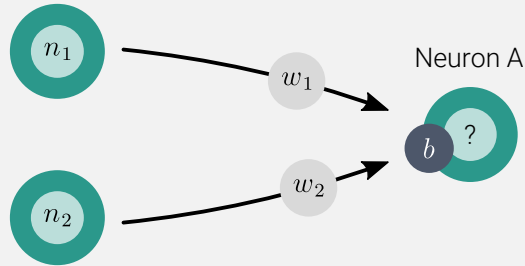
# Modell eines Perzeptrons



Die Idee dazu: Wenn die Summe der aktivierten Eingangsneuronen mit einer festen Verbindung einen gewissen Schwellwert erreicht, dann ist auch Neuron A aktiviert (anderenfalls nicht).

# Modell eines Perzeptrons

Zusammenfassend wurde die folgende Rechenvorschrift hergeleitet (in vereinfachter Darstellung):



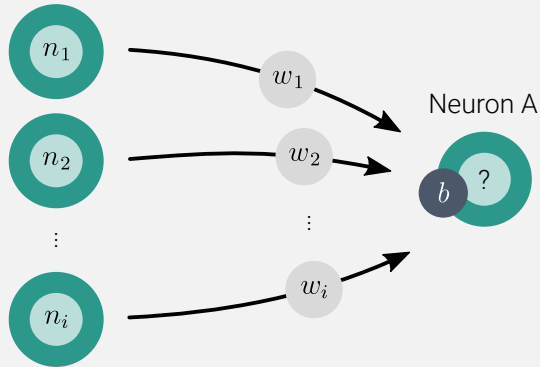
Neuron A ist genau dann aktiviert, falls  $n_1 \cdot w_1 + n_2 \cdot w_2 \geq b$  gilt.

Die Zustände der Neuronen ( $n_1$  und  $n_2$ ) sowie die Gewichte ( $w_1$  und  $w_2$ ) können wie zuvor eingeführt die Wert 0 und 1 annehmen. Der Schwellwert  $b$  ist eine (nicht negative) ganze Zahl.



# Modell eines Perzeptrons

Das allgemeine Modell eines Perzeptrons wird auf gleiche Art und Weise definiert:



Neuron A ist genau dann aktiviert, falls die Summe der gewichteten Neuronen größer oder gleich dem Schwellwert ist, falls also  $n_1 \cdot w_1 + n_2 \cdot w_2 + \dots + n_i \cdot w_i \geq b$  gilt.

# Aufgabe



Folge dem Link bzw. QR-Code und mache dich mit der Anwendung vertraut, in der die Rechenvorschrift eines Perzeptrons mit drei Eingangsneuronen anhand eines Beispiels erprobt werden kann:



[www.taralino.de/view/K1AQNNPZ](http://www.taralino.de/view/K1AQNNPZ)



Finde alle Konstellationen (d.h. Kombinationen der Zustände der drei Eingangsneuronen), sodass Neuron A ist.

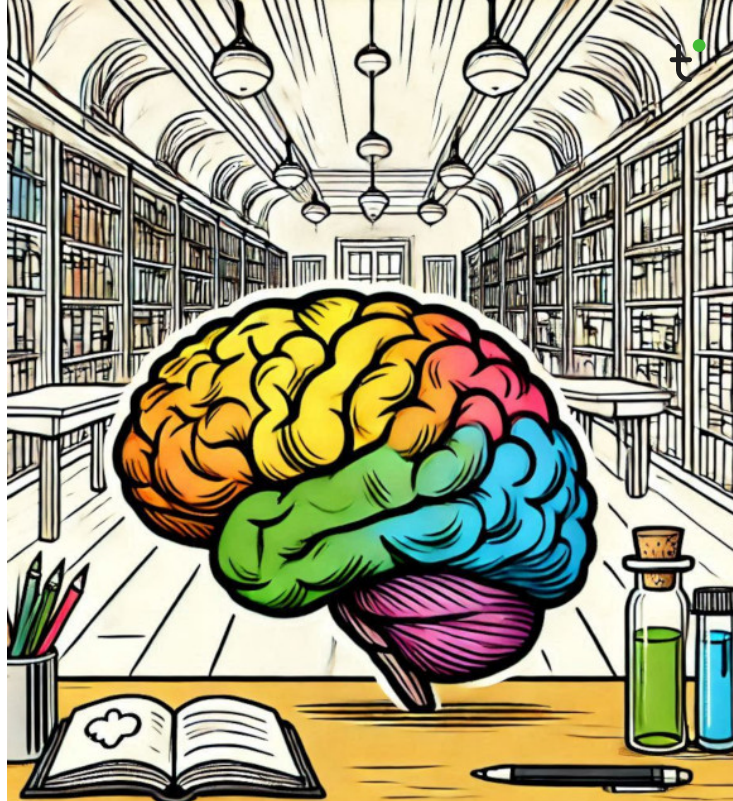
▶ zur Lösung

## Lernprozesse

Eure Zufriedenheit steigt, das Modell wirkt nicht nur logisch, sondern nimmt sich auch die Natur selbst zum Vorbild. Ihr erinnert euch:

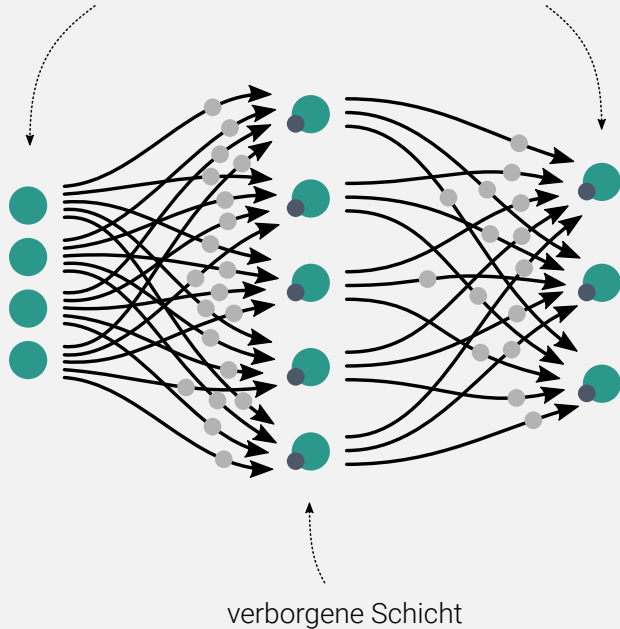
Lernprozesse in unserem Gehirn basieren darauf, neue Verbindungen zwischen Neuronen zu knüpfen oder alte Verbindungen aufzulösen. Genau diese Dynamik lässt sich durch die Gewichte im Modell simulieren.

Der nächste Schritt liegt vor euch: ein ganzes neuronales Netz zum Leben zu erwecken!



Eingangsschicht

Ausgangsschicht



## Künstliches neuronales Netz

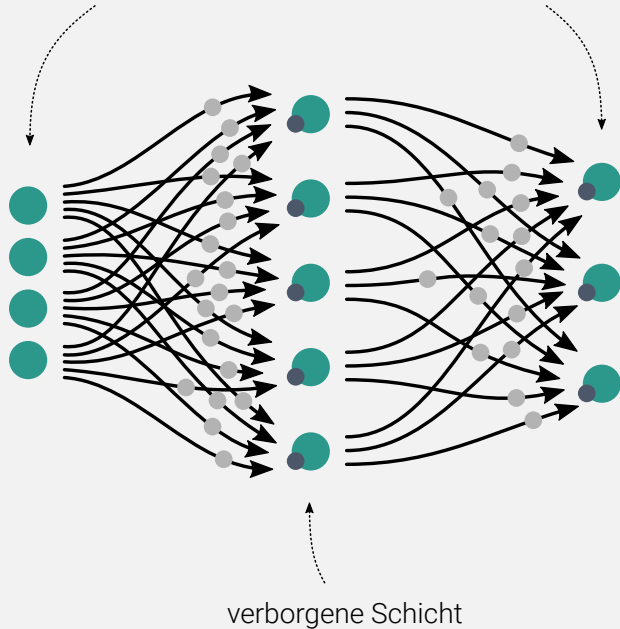


Viele Perzeptronen gemeinsam bilden ein neuronales Netz:

In einem künstlichen neuronalen Netz sind die Neuronen spaltenweise aufgebaut, in den sogenannten Schichten: Die Eingangsneuronen ganz links befinden sich in der Eingangsschicht, die Ausgangsneuronen rechts in der Ausgangsschicht. Zwischen zwei aufeinanderfolgenden Schichten sind alle Neuronen miteinander verbunden.

Eingangsschicht

Ausgangsschicht



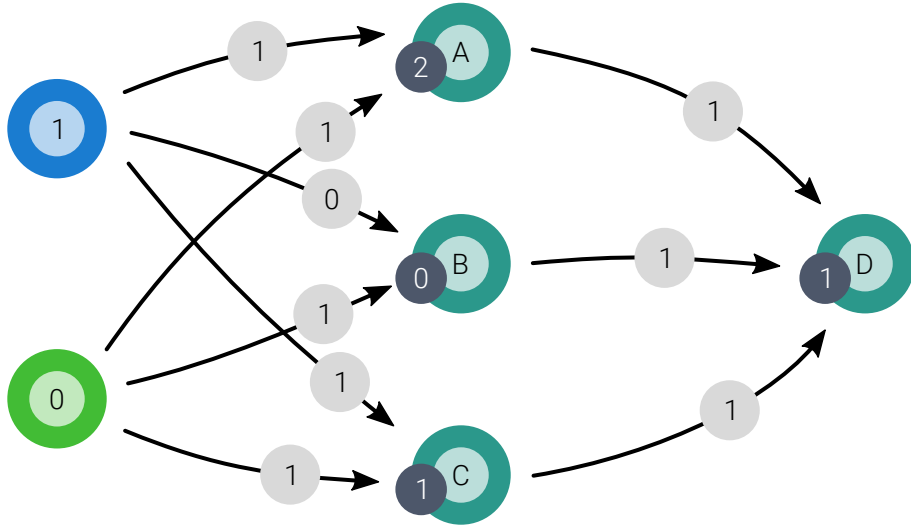
## Künstliches neuronales Netz



Von zentraler Bedeutung ist schließlich folgende Erkenntnis:

Sind in einem künstlichen neuronalen Netz sämtliche Gewichte und Schwellwerte bekannt und sind die Zustände bzw. Werte der Eingangsneuronen gegeben, so können auch die Zustände bzw. Werte der Ausgangsneuronen bestimmt werden.

# Aufgabe



Bestimme die Zustände bzw. Werte der Neuronen A bis D in Abhängigkeit der Eingangsneuronen.

► zur Lösung



## Strichzeichnungen



Mit ungebrochenem Siegeswillen seid ihr überzeugt, ein neuronales Netz geschaffen zu haben, das Strichzeichnungen klassifizieren kann.

Um eure Theorie zu prüfen, startet ihr den ersten Testlauf und erinnert euch an die Anfänge von Neuralis. Auch damals begann alles damit, einfachste Schwarz-Weiß-Bilder zu analysieren.

Doch eine Frage brennt euch unter den Nägeln: Wie genau können Schwarz-Weiß-Bilder von einem neuronalen Netz verarbeitet werden?

# Klassifikationsaufgabe



Das erste Anwendungsbeispiel zur Erprobung eines neuronalen Netzes ist folgendes:

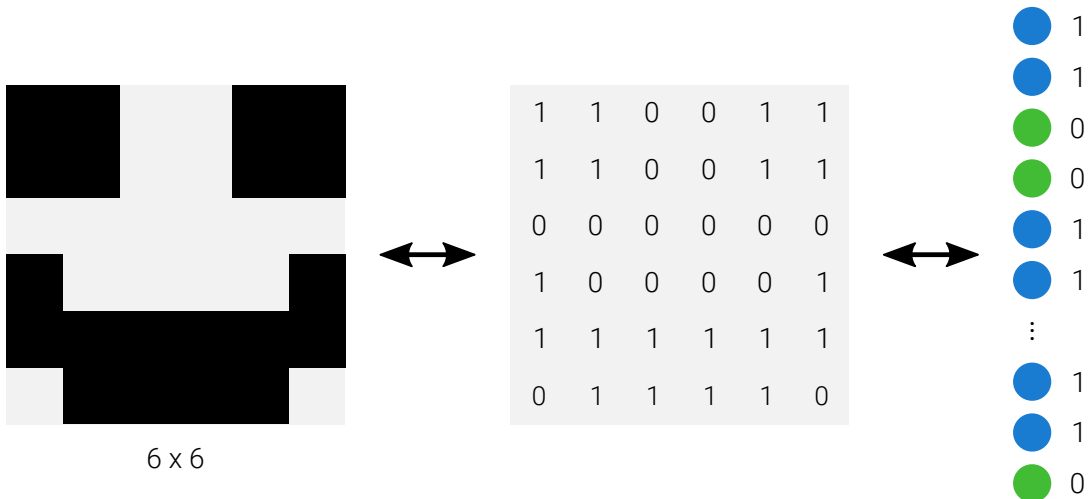
Die Aufgabe lautet, Schwarz-Weiß-Bilder bestehend aus 16 x 16 Pixel zu analysieren und eine Rückmeldung zu geben, ob es sich um einen Kreis (bzw. Buchstabe O) oder um ein Kreuz (bzw. Buchstabe X) handelt.

Die Bildpunkte bzw. Pixel eines Bildes sind entweder schwarz (kodiert als 1) oder weiß (kodiert als 0).



# Bilder als Eingangssignal

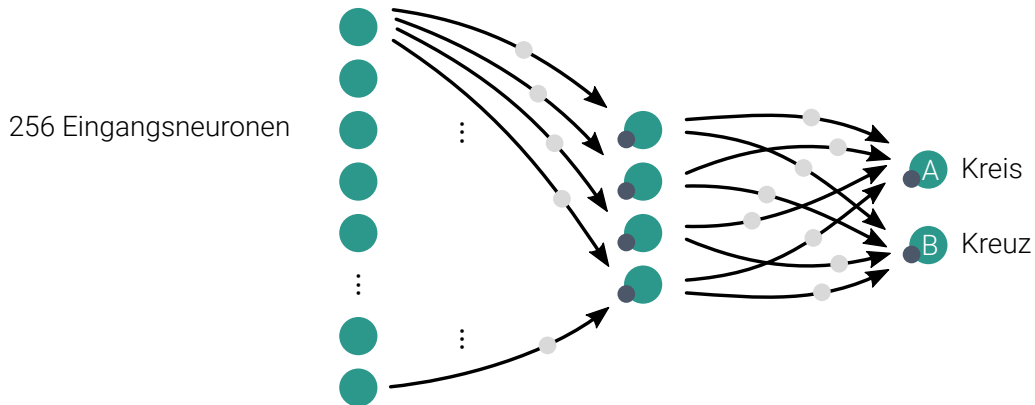
Jedes Schwarz-Weiß-Bild (bestehend ausschließlich aus schwarzen und weißen Pixel) kann als Eingangssignal eines neuronalen Netzes aufgefasst werden (und umgekehrt):



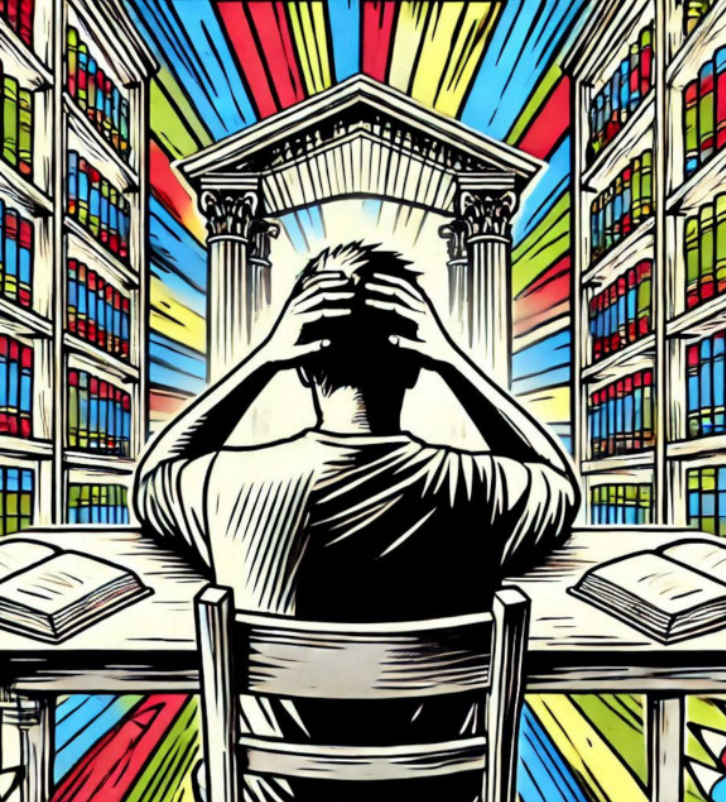
# Neuronales Netz zur Klassifikationsaufgabe



Zur Lösung der Klassifikationsaufgabe wird folgendes neuronales Netz verwendet:



Die Idee dabei ist, dass zu jedem Eingangssignal idealerweise genau ein Ausgangsneuron aktiviert wird, nämlich gemäß dessen, was im Eingangssignal (bzw. im 16 x 16 Bild) erkannt wurde.



## Wahl der Gewichte



Alles scheint reibungslos zu laufen, doch ihr geratet ins Grübeln. Ein Gedanke lässt euch nicht mehr los:

Das neuronale Netz kann die Aufgabe nur dann bewältigen, wenn die Gewichte und Schwellwerte sinnvoll gewählt wurden.

Die anfängliche Euphorie weicht langsam einem mulmigen Gefühl. Habt ihr etwas Entscheidendes übersehen? Oder steckt hier ein verstecktes Problem, das euch noch auf die Probe stellen wird?

## Wahl der Gewichte

Doch gerade, als der Zweifel an euch zu nagen beginnen, wendet sich das Blatt. In den alten Neuralis-Entwicklungsunterlagen findet ihr genau die Daten, die ihr braucht, um die Gewichte und Schwellwerte zu definieren.

Eure Anspannung weicht einer elektrisierenden Aufregung: Mit diesen Informationen könnt ihr endlich das zuvor beschriebene neuronale Netz in einem echten Testlauf zum Leben erwecken. Der Moment der Wahrheit ist gekommen!



# Experimentieraufgabe



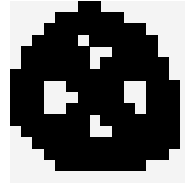
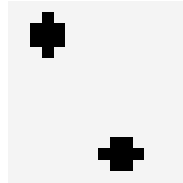
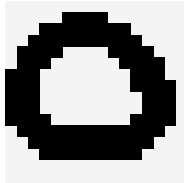
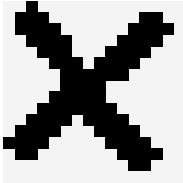
Folge dem Link bzw. QR-Code und mache dich mit der Anwendung vertraut:



[www.taralino.de/view/K1AQNNBD](http://www.taralino.de/view/K1AQNNBD)



Teste insbesondere Eingangssignale ähnlich zu den folgenden Eingangssignalen bzw. Bildern:





## Der Perspektivwechsel



Obwohl ihr mit eurem grundlegenden Modell die erste Klassifikationsaufgabe lösen konntet, macht sich Enttäuschung breit: Das Ergebnis ist nicht überzeugend. Was passiert zum Beispiel, wenn beide Ausgangsneuronen gleichzeitig aktiviert werden?

Doch Aufgeben ist keine Option! Bei einem kleinen Spaziergang lasst ihr euren Gedanken freien Lauf. Ideen fliegen hin und her und es dauert gar nicht lange, da kommt euch der erste Einfall, wie ihr das Modell verbessern könnt. Vielleicht seid ihr der Lösung näher, als ihr glaubt!

# Verallgemeinerungen



Obwohl es nicht mehr der Natur als Vorbild entspricht, sind aus mathematischer Sicht folgende Verallgemeinerungen sehr naheliegend:

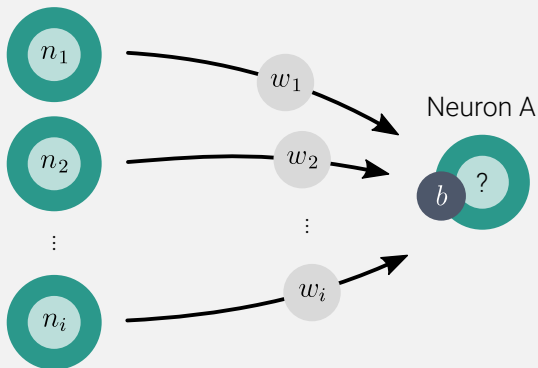
Für die Gewichte und Schwellwerte eines neuronalen Netzes werden sämtliche (reelle) Zahlen zugelassen (insbesondere auch negative Werte).

Ebenso können die Eingangsneuronen verallgemeinert werden:

Für die Eingangsneuronen werden nicht nur die Werte 0 und 1 zugelassen, sondern auch (reelle) Zahlen zwischen 0 und 1.

# Verallgemeinerungen

An der Rechenvorschrift hat sich trotz der Verallgemeinerungen rein gar nichts geändert:



Neuron A ist genau dann aktiviert, falls die Summe der gewichteten Neuronen größer oder gleich dem Schwellwert ist, falls also  $n_1 \cdot w_1 + n_2 \cdot w_2 + \dots + n_i \cdot w_i \geq b$  gilt.



# Verallgemeinerungen



Von zentraler Bedeutung in der Rechenvorschrift zuvor ist die Differenz zwischen der gewichteten Summe sowie dem Schwellwert. Daher wird folgende Definition eingeführt:

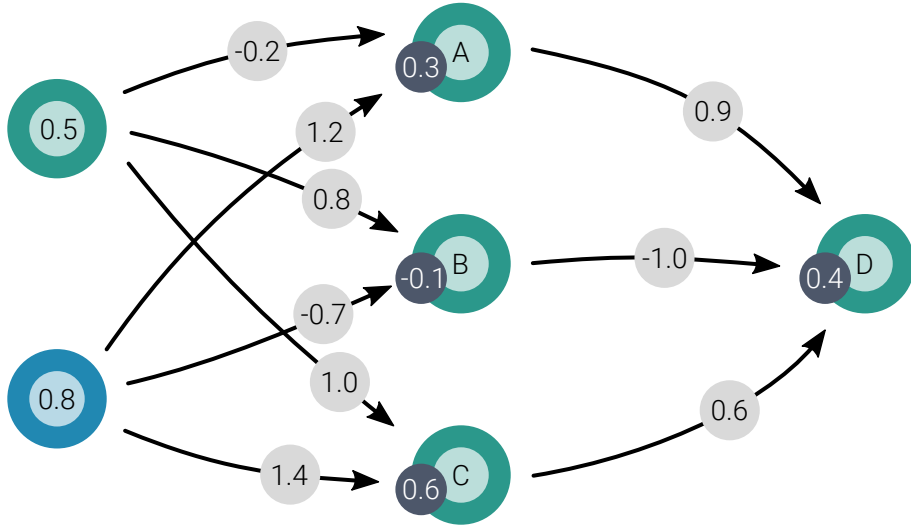
$$x = n_1 \cdot w_1 + n_2 \cdot w_2 + \dots + n_i \cdot w_i - b$$

Für den Wert von Neuron A gilt dank der Definition von  $x$  nun:

$$\text{Wert von Neuron A} = f(x) = \begin{cases} 1 & \text{falls } x \geq 0 \\ 0 & \text{falls } x < 0 \end{cases}$$

Bei der (reellwertigen) Funktion  $f(x)$  handelt es sich um eine sogenannten Aktivierungsfunktion.

# Aufgabe



Bestimme die Zustände bzw. Werte der Neuronen A bis D in Abhängigkeit der Eingangsneuronen.

► zur Lösung



## Die Aktivierungsfunktion



Etwas lässt euch keine Ruhe: Warum können alle Neuronen außer den Eingangsneuronen nur die Werte 0 und 1 annehmen? Wäre es nicht besser, diese Werte zu verallgemeinern?

Denn so könnten die Werte der Ausgangsneuronen auch als Vorhersagewahrscheinlichkeiten angesehen werden.

Ihr durchstöbert die Bibliothek und wühlt in den alten Unterlagen - bis euch Abbildungen mit den Graphen von Aktivierungsfunktionen ins Auge springen. Könnten sie der Schlüssel sein?

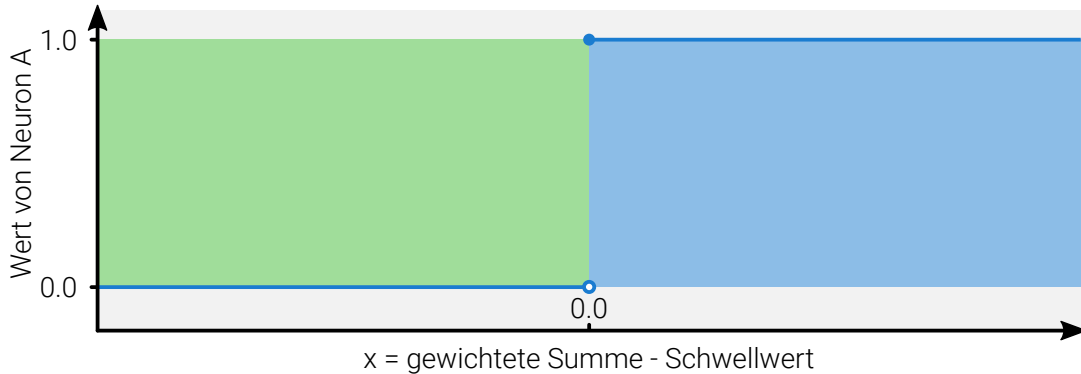
# Aktivierungsfunktion



Wie bereits bekannt, ergibt sich der Wert eines nachfolgenden Neurons in Abhängigkeit von

$$x = n_1 \cdot w_1 + n_2 \cdot w_2 + \dots + n_i \cdot w_i - b.$$

Entsprechend lässt sich der Graph der Aktivierungsfunktion folgendermaßen darstellen:



# Aktivierungsfunktion



Bei der Aktivierungsfunktion zuvor handelt es sich um die Stufenfunktion: Falls  $x \geq 0$  und damit falls die gewichtete Summe größer oder gleich dem Schwellwert ist, dann ist der Wert des nachfolgenden Neurons gleich 1 (und anderenfalls gleich 0).

Die Idee zur Verallgemeinerung der Aktivierungsfunktion ist folgende:

Das sprunghafte Verhalten bei  $x = 0$  soll durch eine geeignetere Aktivierungsfunktion verbessert werden. Weiterhin soll das nachfolgende Neuron A aber Werte zwischen 0 und 1 annehmen. Dabei soll der Wert ungefähr 1 sein soll, falls  $x$  groß ist, und ungefähr 0, falls  $x$  klein ist.

Zudem sollte es sich um eine stetige Funktion handeln (also keine Sprünge aufweisen, wie es bei der Stufenfunktion der Fall ist).

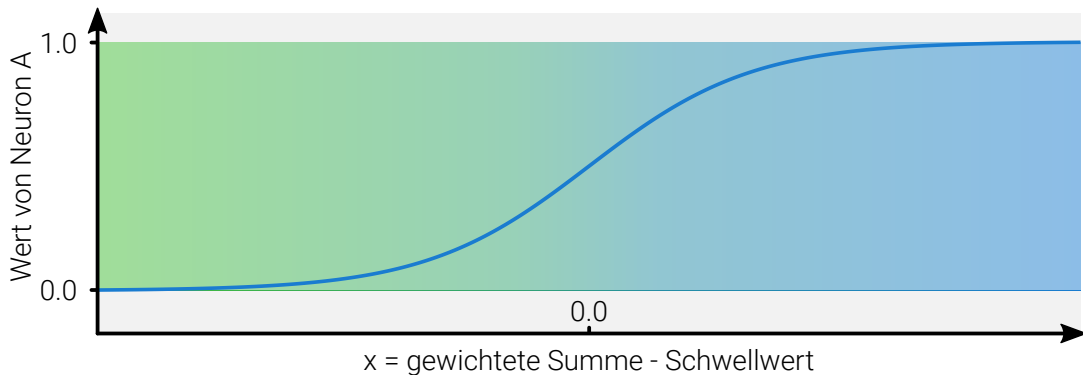
# Aktivierungsfunktion

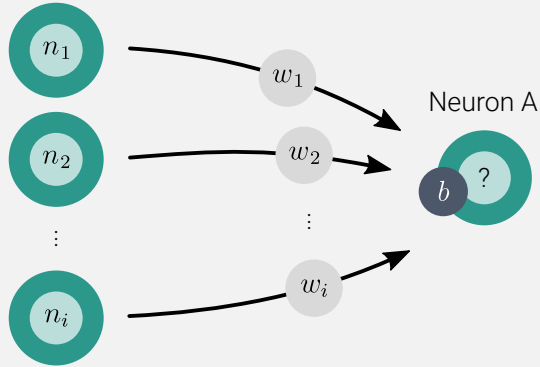


Eine Aktivierungsfunktion, die das zuvor beschriebene Verhalten aufweist, ist die Sigmoid-Funktion:

$$f(x) = \frac{1}{1 + \exp(-x)}$$

Der Graph der Sigmoid-Aktivierungsfunktion ist folgender:





Zusammenfassend ergibt sich im verallgemeinerten Modell eines Perzeptrons:

$$\text{Wert von Neuron A} = f(x) = \frac{1}{1 + \exp\left(- (n_1 \cdot w_1 + n_2 \cdot w_2 + \dots + n_i \cdot w_i - b)\right)}$$

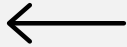
## Aufgabe zur Reflexion



Folge dem Link bzw. QR-Code und mache dich mit der Anwendung vertraut:



[www.taralino.de/view/K1AQNNCS](http://www.taralino.de/view/K1AQNNCS)



Reflektiere, ob du sämtliche Inhalte bisher verstanden hast:

Es sollte für dich keine Blackbox mehr sein, wie die Werte der Neuronen eines neuronalen Netzes in Abhängigkeit der Eingangsneuronen sowie in Abhängigkeit der Gewichte und Schwellwerte (schichtweise von links nach rechts) berechnet werden.





## Die Zwischenstation



Neuronale Netze sind für euch keine Blackbox mehr, im Gegenteil:

Ihr seid begeistert, wie klar und einfach sich die Auswertung gestaltet, solange alle Gewichte und Schwellwerte bekannt sind.

Mit wachsendem Selbstbewusstsein rückt euer großes Ziel in greifbare Nähe. Doch bevor ihr den finalen Sprung wagt, sammelt ihr weitere Erfahrungen mit kleinen, überschaubaren Klassifikationsaufgaben. Jeder Schritt bringt euch dem Erfolg ein Stück näher!

# Klassifikationsaufgaben



Um es nochmals formal zusammenzufassen:

Bei einer Klassifikationsaufgabe geht es darum, anhand gewisser Merkmale eines Objekts zu entscheiden, welcher Kategorie bzw. Klasse das Objekt (wahrscheinlich) angehört.

Um eine Klassifikationsaufgabe mit einem neuronalen Netz zu lösen, wird pro Merkmal ein Eingangsneuron und pro Kategorie bzw. Klasse ein Ausgangsneuron verwendet.

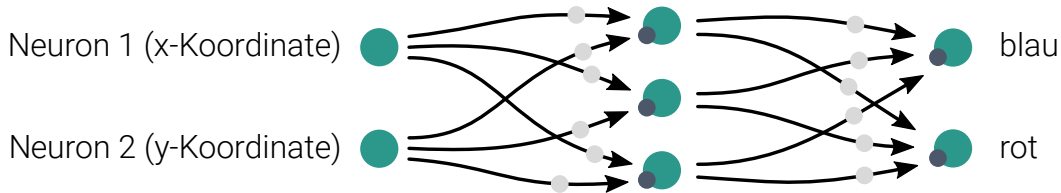
Jedes Objekt (z.B. ein Bild einer Strichzeichnung) kann nun am neuronalen Netz ausgewertet werden. Das Ausgangsneuron mit dem größten Wert liefert eine Vorhersage darüber, welcher Kategorie bzw. Klasse das Objekt (wahrscheinlich) angehört.

# Klassifikationsaufgaben

Ein kleines Beispiel, um weitere Erfahrungen zu sammeln, ist folgendes:

Die Aufgabe lautet, Punkte in der Ebene (gegeben durch x- und y-Koordinaten) bezüglich ihrer zugehörigen Farbe zu klassifizieren (z.B. in blau und rot).

Ein mögliches neuronales Netz, um diese Aufgabe zu lösen, ist folgendes:



# Aufgabe



Verwende nochmals die gleiche Anwendung wie zuvor zur Klassifikation in blau und rot:



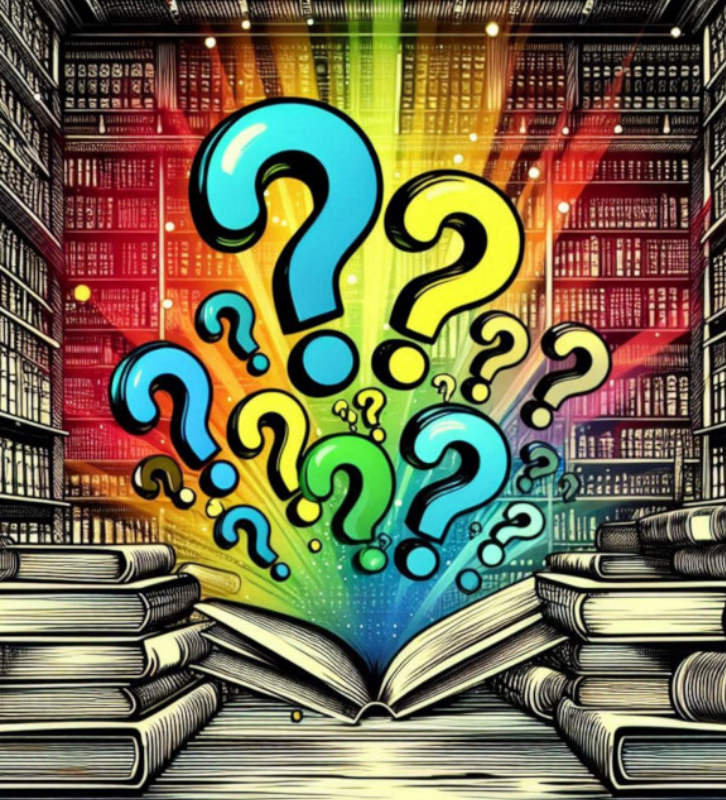
[www.taralino.de/view/K1AQNNCS](http://www.taralino.de/view/K1AQNNCS)



Bestimme das Klassifikationsergebnis bezüglich folgender Daten der Eingangsneuronen:

Neuron 1	0.6	0.8	0.3	0.7	0.1	0.8	0.2
Neuron 2	0.4	1.0	0.5	0.8	0.0	0.4	0.4
Ergebnis	blau						

▶ zur Lösung



## Lernen, Lernen, Lernen



Ihr habt es geschafft: Ihr versteht, wie neuronale Netze Klassifikationsaufgaben lösen können. Doch gerade, als die Euphorie auf ihrem Höhepunkt ist, gerät sie ins Wanken.

Ein Gedanke lässt euch keine Ruhe: Damit das Netz tatsächlich funktioniert, müssen Gewichte und Schwellwerte perfekt auf die Aufgabe abgestimmt sein. Doch wie soll das gelingen?

Eure Gedanken stocken und ein Hauch von Unsicherheit macht sich breit. Könnte dies die größte Hürde sein, die euch von eurem Ziel trennt?

## Lernen, Lernen, Lernen

Ihr beginnt, über das Wesen des Lernens nachzudenken. Schließlich haben auch wir das Erkennen von Mustern nicht von Geburt an gekannt. Unzählige Beispiele wurden uns gezeigt, bis wir im Kindesalter gelernt haben, wie ein Schmetterling aussieht oder wie sich ein Apfel von einer Blume unterscheidet.

Eure Überlegung führt zu einer klaren Erkenntnis: Wenn ein neuronales Netz lernen soll, müssen auch ihm Beispiele gezeigt werden. Für das präzise Einstellen der Gewichte und Schwellwerte sind Trainingsdaten der Schlüssel!



# Trainingsdaten



Um eine Klassifikationsaufgabe mit einem neuronalen Netz lösen zu können, müssen die Gewichte und Schwellwerte sinnvoll eingestellt werden. Und genau dazu werden Trainingsdaten benötigt:

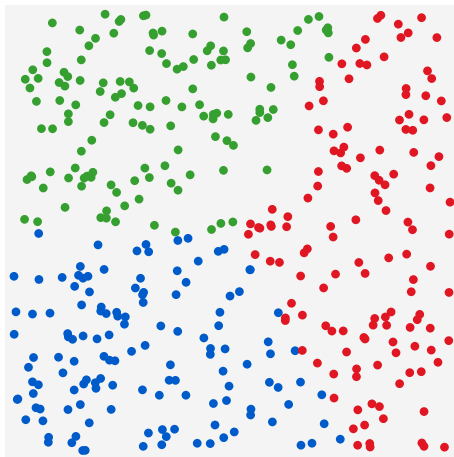
Bei den Trainingsdaten handelt es sich um eine große Menge an Objekten, die jeweils beschrieben werden durch ihre Merkmale und damit durch die Werte der Eingangsneuronen. Zu jedem Objekt muss zudem bekannt sein, was das zugehörige Klassifikationsergebnis ist.

Das Ziel ist es nun, die Gewichte und Schwellwerte des neuronalen Netzes so zu wählen, dass möglichst viele Trainingsobjekte korrekt klassifiziert werden.

# Trainingsdaten



Angenommen, die Klassifikationsaufgabe lautet, Punkte in der Ebene (gegeben durch x- und y-Koordinaten jeweils zwischen 0 und 1) bezüglich ihrer zugehörigen Farbe zu klassifizieren (in blau, rot und grün). Dann können mögliche Trainingsdaten folgendermaßen veranschaulicht werden:





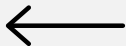
## Experimentieraufgabe



Folge dem Link bzw. QR-Code und mache dich mit der Anwendung vertraut:



[www.taralino.de/view/K1AQNNAA](http://www.taralino.de/view/K1AQNNAA)



Verändere die Gewichte und Schwellwerte derart, sodass ein möglichst großer Anteil der Trainingsdaten korrekt klassifiziert wird.

Hinweis: Es ist durchaus möglich, dass über 95 Prozent der Trainingsobjekte korrekt klassifiziert werden. Allerdings dürfte es sehr schwierig sein, ein derartig gutes Ergebnis ohne weitere Hilfe zu erzielen.



## Der Schlüssel



Wieder einmal macht sich Frustration breit. Ihr habt alles versucht, die Gewichte und Schwellwerte manuell anzupassen, doch das Ergebnis bleibt enttäuschend.

Es wird klar, dass ihr an eure Grenzen stoßt. Doch ihr gebt nicht auf. Es muss doch eine Möglichkeit geben, die Werte so zu bestimmen, dass das Netz einen möglichst großen Anteil der Trainingsdaten korrekt klassifiziert. Welche Hilfsmittel könnten euch jetzt noch helfen?

## Der Schlüssel

Tief in den Archiven der Bibliothek stoßt ihr auf eine seltsame Tafel, übersät mit Symbolen und Diagrammen. Als ihr sie entschlüsselt, wird euch klar:

Das Trainieren eines neuronalen Netzes ist ein mathematisches Optimierungsproblem. Ihr müsst die Gewichte so anpassen, dass der Fehler zwischen den Vorhersagen des Netzes und den tatsächlichen Ergebnissen minimiert wird.





## Der Schlüssel



Während ihr noch darüber nachdenkt, wie ihr das Problem lösen könnt, wird euch langsam etwas klar: Die Lösung liegt in Iteration und Anpassung.

Ihr erinnert euch an Methoden wie die Gradientensuche, die den Weg zu den optimalen Gewichten ebnen. Mit jeder Iteration, jedem kleinen Schritt, nähert ihr euch dem Ziel.

Nach und nach versteht ihr, dass es nicht um sofortige Perfektion geht, sondern darum, durch stetige Verbesserungen das Modell immer präziser zu machen.

## Der Schlüssel

Plötzlich macht es Klick! Mit einem Mal versteht ihr die Skizzen und Notizen in den alten Entwicklungsunterlagen. Alles ergibt Sinn, die scheinbar kryptischen Anmerkungen enthüllen ihre Bedeutung. Euch wird klar:

Für einen Computer ist es ein Leichtes, ein neuronales Netz anhand von Trainingsdaten zu trainieren. Die einst so abstrakte Theorie wird greifbar - ihr haltet den Schlüssel zur Lösung in euren Händen!



# Trainingsphase



Ohne an dieser Stelle auf die Details einzugehen.

Sofern Trainingsdaten mit bekanntem Klassifikationsergebnis zu jedem Trainingsobjekt gegeben sind, können mathematische Optimierungsverfahren eingesetzt werden, um ein neuronales Netz zu trainieren.

Mit anderen Worten: Ein iteratives Verfahren liefert Werte für sämtliche Gewichte und Schwellwerte, sodass ein möglichst großer Anteil der Trainingsobjekte korrekt klassifiziert wird.

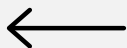
## Aufgabe zur Reflexion



Verwende nochmals die gleiche Anwendung wie zuvor zur Klassifikation in blau, rot und grün:



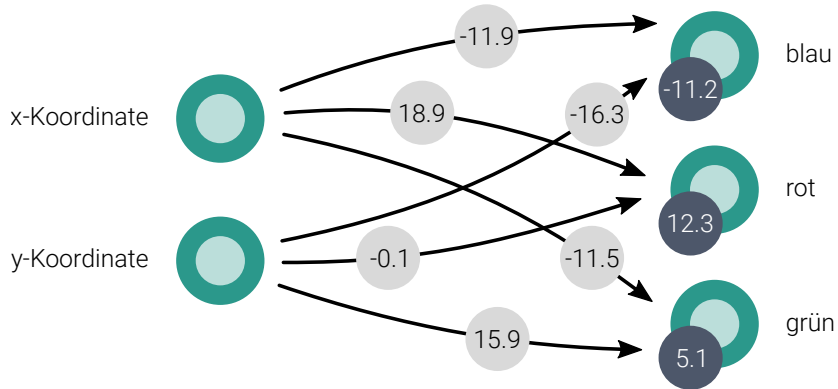
[www.taralino.de/view/K1AQNNAA](http://www.taralino.de/view/K1AQNNAA)



Die zugehörige Aufgabe befindet sich auf der nächsten Seite.

# Aufgabe zur Reflexion

Stelle die Gewichte und Schwellwerte gemäß der folgende Abbildung ein:



Wie groß ist damit der Anteil der korrekt klassifizierten Trainingsdaten?





## Das Datenarchiv



Ihr seid überwältigt. Wer hätte gedacht, dass schon so einfache neuronale Netze in der Lage sind, eine Klassifikationsaufgabe zu lösen? Es fühlt sich an wie Magie, die langsam Gestalt annimmt.

Alle Puzzlesteine, die ihr gesammelt habt, fügen sich zusammen. Der Moment rückt näher, in dem euer Werk vollendet ist. Es fehlt nur noch der letzte Schritt und euer neuronales Netz wird lebendig!

## Das Datenarchiv

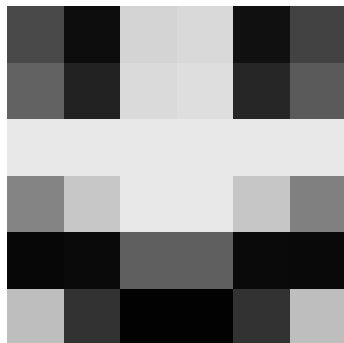
Tatsächlich ist eure Bibliothek eine einzigartige Schatzkammer - nicht nur für Bücher und Unterlagen, sondern auch für Daten.

Mit Vorfreude beginnt ihr eure Suche und schon nach kurzer Zeit werdet ihr fündig: ein Datensatz, perfekt geeignet, um euer neuronales Netz zur Klassifikation von Strichzeichnungen zu trainieren.



# Bilder als Eingangssignal

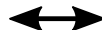
Aufgrund der Verallgemeinerung können nun Schwarz-Weiß-Bilder mit beliebigen Graustufen als Eingangssignal eines neuronalen Netzes aufgefasst werden (und umgekehrt):



6 x 6



0.7	0.9	0.2	0.1	0.9	0.7
0.6	0.9	0.1	0.1	0.9	0.6
0.1	0.1	0.1	0.1	0.1	0.1
0.5	0.2	0.1	0.1	0.2	0.5
1.0	1.0	0.6	0.6	1.0	1.0
0.3	0.8	1.0	1.0	0.8	0.3



- 0.7
- 0.9
- 0.2
- 0.1
- 0.9
- 0.7
- $\vdots$
- 1.0
- 0.8
- 0.3

# Klassifikationsaufgabe



Die finale Aufgabe kann folgendermaßen zusammengefasst werden:

Die Aufgabe lautet, Strichzeichnungen in Form von Schwarz-Weiß-Bildern bestehend aus 40 x 40 Pixel zu analysieren. Insbesondere sollen Strichzeichnungen bezüglich der folgenden vier Objekte klassifiziert werden:



Apfel



Schmetterling



Fisch

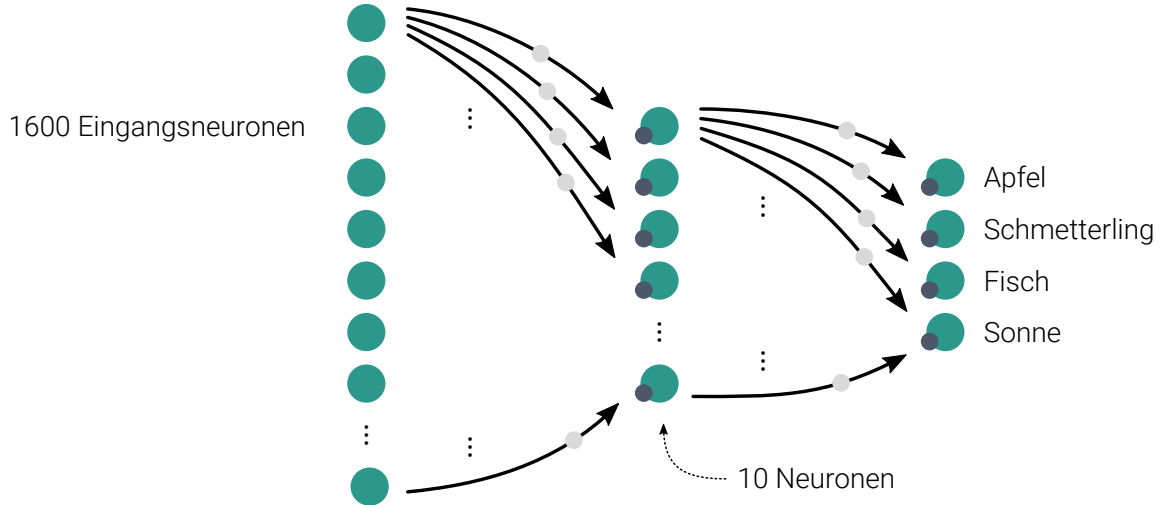


Sonne

# Klassifikationsaufgabe



Als neuronales Netz zur Lösung der Aufgabe wird folgendes Layout verwendet, wobei die Werte sämtlicher Gewichte und Schwellwerte auf Grundlage von Trainingsdaten bestimmt werden:



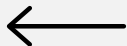
## Experimentieraufgabe



Folge dem Link bzw. QR-Code und mache dich mit der Anwendung vertraut:



[www.taralino.de/view/K1AQNDD](http://www.taralino.de/view/K1AQNDD)



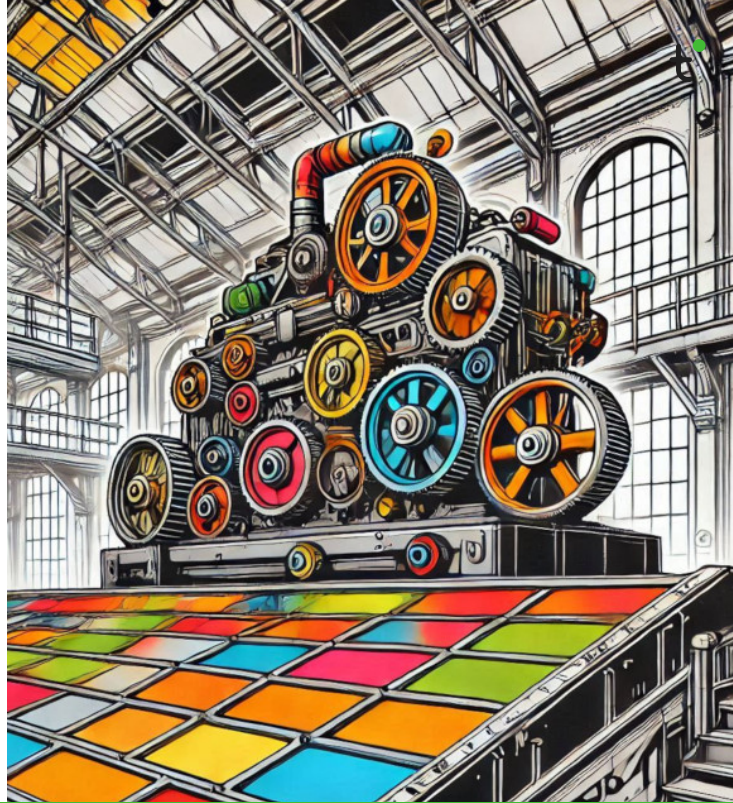
Fertige zahlreiche Strichzeichnungen der vier zu klassifizierenden Objekte an und schaue, wie gut das neuronale Netz die Aufgabe lösen kann.

## Herzlichen Glückwunsch!

Ihr habt es geschafft! Neuralis ist gerettet und mit eurem Erfolg habt ihr nicht nur eine wichtige Mission erfüllt, sondern auch etwas Wertvolles gelernt: Ihr versteht jetzt, wie neuronale Netze funktionieren.

Diese Erkenntnis ist die Grundlage vieler moderner KI-Modelle: von der Bildklassifikation bis hin zur generativen KI, die kreative Inhalte erschafft.

Eure Reise hat die Welt ein Stück schlauer gemacht. Und euch selbst auch!



# Anhang

Lösungen



# Lösung

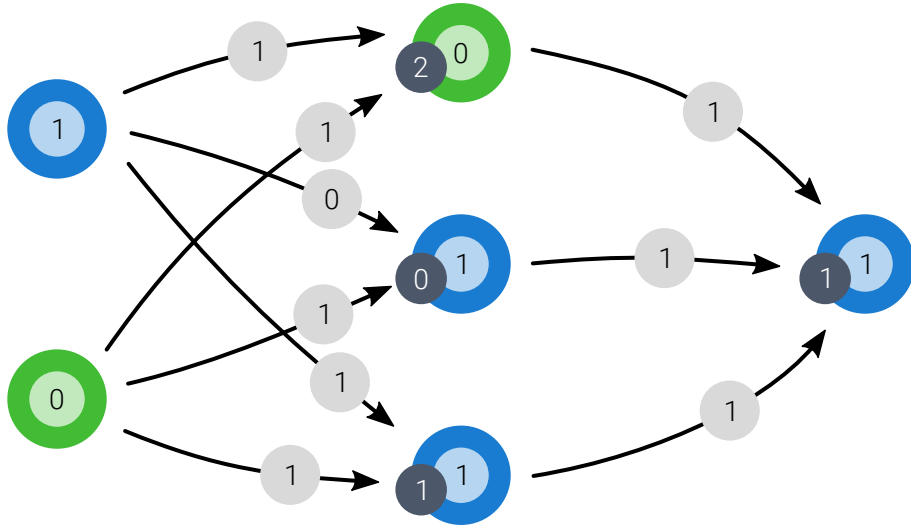


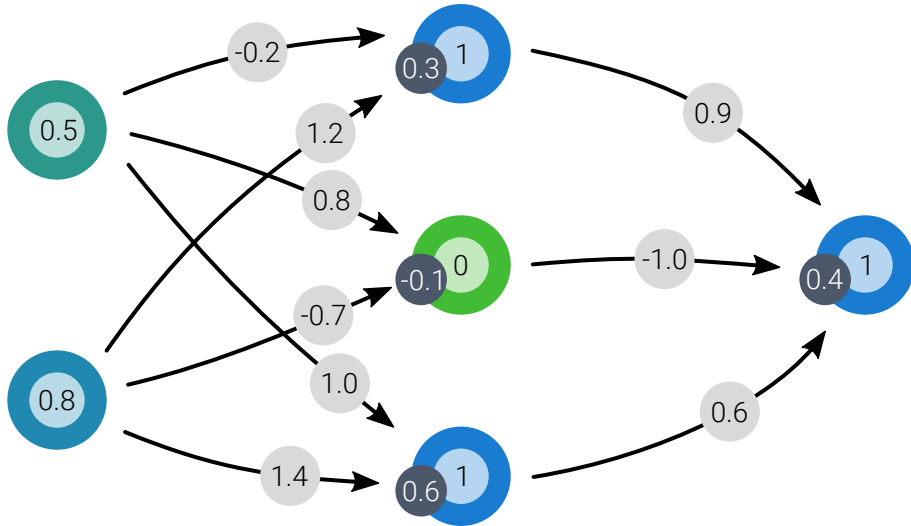
Es gibt genau zwei Konstellationen, sodass Neuron A aktiviert ist, nämlich

$$n_1 = 1, n_2 = 0, n_3 = 1 \quad \text{und} \quad n_1 = 1, n_2 = 1, n_3 = 1.$$

Mit anderen Worten: Die Neuronen 1 und 3 müssen aktiviert sein, damit der Schwellwert von 2 erreicht wird und folglich Neuron A aktiviert ist. Der Wert bzw. Zustand von Neuron 2 hat keinen Einfluss auf Neuron A, da für das zugehörige Gewicht  $w_2 = 0$  gilt.

▶ zurück zur Aufgabe





# Lösung



Unter Verwendung des neuronalen Netzes ergeben sich die folgenden Ergebnisse:

Neuron 1	0.6	0.8	0.3	0.7	0.1	0.8	0.2
Neuron 2	0.4	1.0	0.5	0.8	0.0	0.4	0.4
Ergebnis	blau	rot	blau	rot	rot	blau	rot