

# #7 TECHARTIKEL

## Künstliche neuronale Netze

---

August 22, 2022

„I get very excited when we discover a way of making neural networks better – and when that’s closely related to how the brain works.“ – Geoffrey Hinton

**Im vorausgegangenen Beitrag wurden die Basics zu künstlicher Intelligenz aufgezeigt. Moderne KI zählt zum Großteil zum Deep Learning und setzt somit künstliche neuronale Netze ein. Künstliche neuronale Netze sind folglich einer der wichtigsten Bausteine moderner KI. In diesem Beitrag sollen genau diese künstlichen neuronalen Netze betrachtet werden.**

Als Vorbild für künstliche neuronale Netze dient die Vernetzung von Neuronen im Nervensystem eines Lebewesens. Zumeist stellen künstliche neuronale Netze eine stark abstrakte Modellierung des Nervensystems dar. Dennoch sind diese Abstraktionen in der Lage gewisse Aufgaben signifikant besser zu lösen als der Mensch.

## Inhalt

1. Neuronen eines künstlichen neuronalen Netzes
2. Aufbau und Funktionsweise künstlicher neuronaler Netze
  - 2.1 Schichten eines künstlichen neuronalen Netzes
  - 2.2 Funktionsweise des künstlichen neuronalen Netzes
3. Sonderformen künstlicher neuronaler Netze
4. Künstliche neuronale Netze trainieren: Lernarten
5. Lernalgorithmen neuronaler Netze
6. Fazit

### 1. Neuronen eines künstlichen neuronalen Netzes

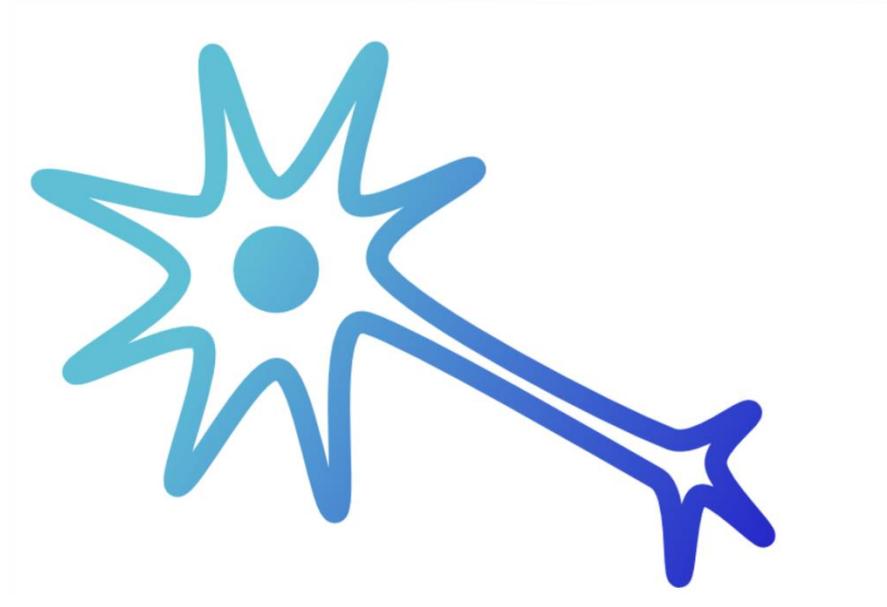


Abbildung: Schichten künstlicher neuronaler Netze

Das Gehirn besitzt eine extrem hohe Anzahl an Neuronen. Spannend dabei ist deren Aufbau und Funktionsweise, da diese den Neuronen eines künstlichen neuronalen Netzes nahezu gleichen. Unsere Abbildung veranschaulicht ein schematisches Neuron. Ein Neuron hat mehrere Eingabe- und Ausgabestellen. Bei unserer Abbildung sind die Eingabestellen links oben zu erkennen und die Ausgabestellen rechts unten. Sobald die einkommenden Signale eine bestimmte Schwelle übersteigen, feuert das Neuron auf alle Ausgabestellen. Ein neuronales Netz, wie eines im Gehirn, entsteht durch die Verknüpfung mehrerer Neuronen.

## 2. Aufbau und Funktionsweise künstlicher neuronaler Netze

Überträgt man dieses biologische neuronale Netz nun in die Informatik und baut daraus eine Datenstruktur, erhält man ein sogenanntes künstliches neuronales Netz. Ein künstliches neuronales Netz ist somit eine Datenstruktur bestehend aus einer beliebigen Anzahl an Neuronen bzw. in der Umsetzung dann eben Knoten.

Aufbau eines künstlichen neuronalen Netzes

### Welche Schichten gibt es?

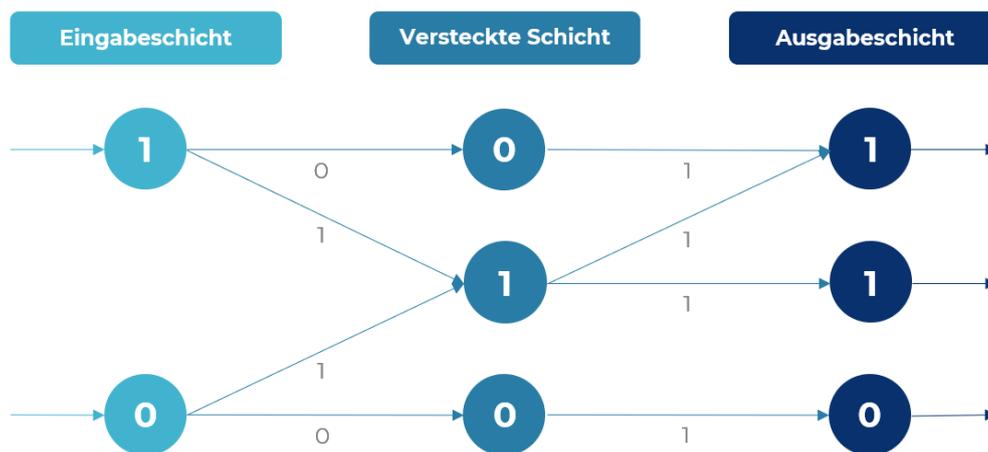


Abbildung: Schichten künstlicher neuronaler Netze

### SCHICHTEN EINES KÜNSTLICHEN NEURONALEN NETZES

Diese Knoten werden für die Übersichtlichkeit in Schichten eingeteilt: **Input Layer**, **Hidden Layer** und **Output Layer**.

#### Eingabeschicht bzw. Input Layer

Die Eingabeschicht nimmt die Daten der Umwelt auf.

#### Ausgabeschicht bzw. Output Layer

Diese Schicht gibt die Daten wieder nach außen ab.

#### Versteckte Schichten bzw. Hidden Layer

Von diesen Schichten gibt es beliebig viele oder auch keine. Diese verbinden üblicherweise die Eingabeschicht mit der Ausgabeschicht.

## FUNKTIONSWEISE DES KÜNSTLICHEN NEURONALEN NETZES

Die Knoten sind untereinander über gerichtete Verbindungen verbunden, wie die neuronalen Netze im Gehirn. Damit am Ende mit dem Netz auch Daten verarbeitet werden können, besitzt jede Verbindung einen Wert, ein sogenanntes Gewicht. Meist **zwischen 0 und 1** oder **-1 und 1**. Jeder Knoten besitzt auch selbst einen Wert. Das Input Layer erhält so die Daten von außen.

Diese Daten werden dann über die Verbindungen weitergegeben und mit den Gewichten dieser manipuliert, meist multipliziert. Der Folgeknoten berechnet dann auf alle eingehenden Werte eine Propagierungsfunktion an. Dies ist meist eine einfache Summenbildung.

Damit die Werte nicht immer größer werden, umso mehr Schichten ein Netz besitzt, wendet jeder Knoten auf das Ergebnis der Propagierungsfunktion eine **Aktivierungsfunktion** an. Dies kann die **Sigmoidfunktion** oder eine **binäre Funktion** sein. Hier wird meist eine Funktion gewählt, die Ergebnisse **zwischen 0 und 1** oder **-1 und 1** zurückgibt, damit es dem natürlichen Vorbild des aktiven oder nicht aktiven Status folgt. So gelangen die Werte dann Schicht für Schicht bis zum Output Layer, welches diese dann wieder ausgibt.

Doch damit ein KNN auch besser wird und nützlich ist, muss es trainiert werden. Beim Training eines KNN werden dann die einzelnen Stellschrauben angepasst. Hierbei können die Gewichte, der Schwellwert ab wann ein Knoten als aktiv gilt oder die einzelnen Funktionen angepasst werden. Dazu aber später mehr.

### 3. Sonderformen künstlicher neuronaler Netze

Es existieren zwei Spezialformen künstlicher neuronaler Netze:

Aufbau eines künstlichen neuronalen Netzes



### Sonderformen künstlicher neuronaler Netze

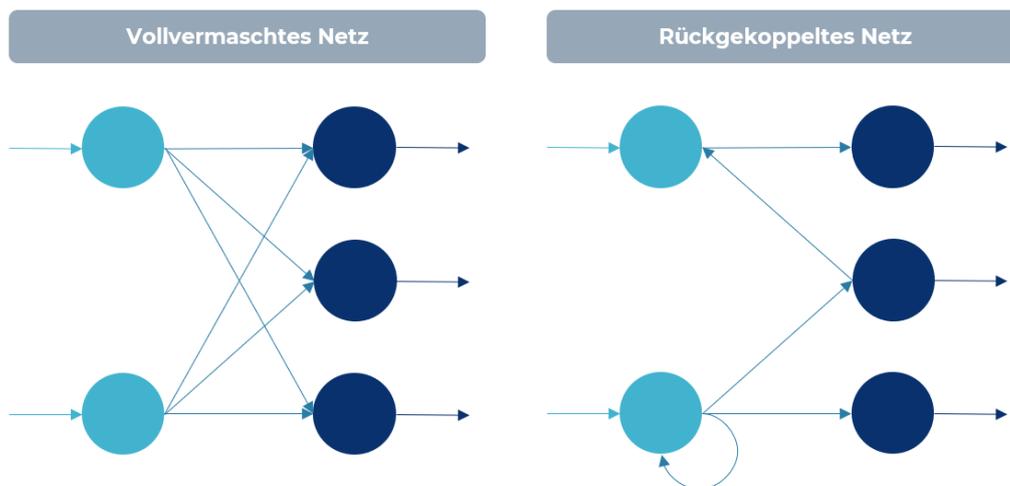


Abbildung: Sonderformen künstlicher neuronaler Netze

#### VOLLVERMASCHTES NETZ

Das vollvermaschte Netz kommt in der Praxis durchaus häufiger vor. Vor allem in der Bilderkennung. Hierbei existiert von jedem Knoten einer Schicht zu jedem Knoten der darauffolgenden Schicht eine Verbindung.

#### RÜCKGEKOPPELTES NETZ

Bei rückgekoppelten Netzen sind nun rückläufige Verbindungen vorhanden. Dies kann entweder von einer Schicht zu einer vorausgegangenen Schicht sein oder auch von einem Knoten zu sich selbst. Diese sind nicht so häufig in der Praxis, da bei diesen die Berechnung der Knotenwerte komplexer und rechenintensiver ist, da nicht mehr Schicht für Schicht berechnet werden kann.

#### 4. Künstliche neuronale Netze trainieren: Lernarten

Aber wie werden denn jetzt KNN besser? Bevor das beantwortet werden kann, muss entschieden werden, wie trainiert werden soll. Anhand des Nutzens des KNN oder der vorhandenen Daten muss die Lernart ausgewählt werden. Insgesamt gibt es drei Lernarten:

Lernarten künstlicher neuronaler Netze

### Welche Lernarten gibt es?

Supervised Learning	Unsupervised Learning	Reinforcement Learning
Model trainiert anhand von bekannten Daten und Beispielen. Es gibt eine klare Zielvariable, die vorhergesagt wird.	Model trainiert ohne Zielvariable und findet eigenständig Muster sowie Zusammenhänge in den Daten.	Model wird nicht gezeigt, welche Aktion oder Handlung in welcher Situation die Richtige ist, sondern es erhält eine positive oder negative Rückmeldung.
<b>Beispiele</b> Klassifikation: Objekterkennung & Texterkennung. Prognose: Wettervorhersage.	<b>Beispiele</b> Clusteranalyse: Segmentierung & Empfehlungssysteme. Dimensionsreduktion: Informationen komprimieren, Strukturen erkennen & Datenvisualisieren.	<b>Beispiele</b> Autonomes Fahren, Robotik, Verkehrssteuerung, Personalisierung von Werbung & Spiele KI.

Abbildung: Sonderformen künstlicher neuronaler Netze

#### ÜBERWACHTES LERNEN siehe [Supervised Learning](#)

Die erste Lernart wäre das überwachte Lernen. Hierbei bekommt das KNN Eingabedaten und zugehörige Label oder Informationen. Anhand dieser Eingabedaten versucht das KNN dann möglichst präzise die Label bzw. Informationen zu bestimmen. Diese Lernart kann für Klassifikation, also die Ausgabe von Klassen wie Ja/Nein, und Prognosen, also numerischen Ausgaben, verwendet werden.

#### Beispiel

Als konkretes Beispiel für eine Prognose: Ich habe eine KNN und gebe diesem Bilder von Hunden und als Label deren Gewicht. Das KNN wird dann darauf trainiert zu einem Hundebild dessen Gewicht anzugeben. Oder ein Beispiel für eine Klassifikation: Ich habe eine Datenbank mit Hunden- und Katzenbildern und zugehörigen Labels (Hund, Katze) und das KNN würde darauf trainiert werden bei einem Bild zu bestimmen, ob ein Hund oder eine Katze abgebildet ist.

## UNÜBERWACHTES LERNEN

Die zweite Lernart ist das unüberwachte Lernen. Hier gibt man dem KNN Eingabedaten, aber ohne Label und das KNN soll selbst Muster und Zusammenhänge in den Daten finden. Mit dieser Lernart erhält man ein KNN, welches eine Clusteranalyse oder Dimensionsreduktion durchführt. Also Daten werden in Cluster eingeteilt und gruppiert oder Verbindungen werden erkannt.

### Beispiel

Als Beispiel für eine Clusteranalyse könnte einem KNN als konkrete Eingabedaten Bilder von Tieren gegeben werden und das KNN clustert diese eigenständig in irgendeiner Art und Weise. Das kann nach Farbe, nach Tierart, aber auch nach irgendetwas abstrakterem sein, was einem Menschen gar nicht auffällt. Als Beispiel für eine Dimensionsreduktion könnte der Verkaufsverlauf von Kunden genommen werden, wodurch Verbindungen erkannt werden können z.B. viele die Produkt A kaufen, kaufen auch Produkt C.

## BESTÄRKENDES LERNEN siehe [Reinforcement Learning](#)

Die letzte Lernart ist das bestärkende Lernen. Hierbei wird dem KNN voller Spielraum gelassen und lediglich anhand von einer Fitnessfunktion oder einem Belohnungssystem gute Ergebnisse belohnt oder schlechte Ergebnisse bestraft (dabei handelt es sich jedoch einfach um eine „negative Belohnung“).

### Beispiel

Hier kann zum Beispiel personalisierte Werbung genommen werden. Klickt ein Nutzer auf die vorgeschlagene Werbung hat das KNN gute Arbeit geleistet und erhält eine Belohnung und kann somit immer besser einschätzen, welche Vorschläge und Zusammenhänge sinnvoll sind und welche nicht.

## 5. Lernalgorithmen neuronaler Netze

Beim Aufbau eines KNN wissen wir nun, dass Stellschrauben im Netz angepasst werden können. Um bei den Basics zu bleiben und nicht zu tief in die Mathematik einsteigen zu müssen, soll lediglich der Hintergedanke von Lernalgorithmen aufgezeigt werden. Grob lassen sich zwei Arten von Lernalgorithmen unterscheiden:

Lernarten künstlicher neuronaler Netze



### Welche Lernalgorithmen gibt es?

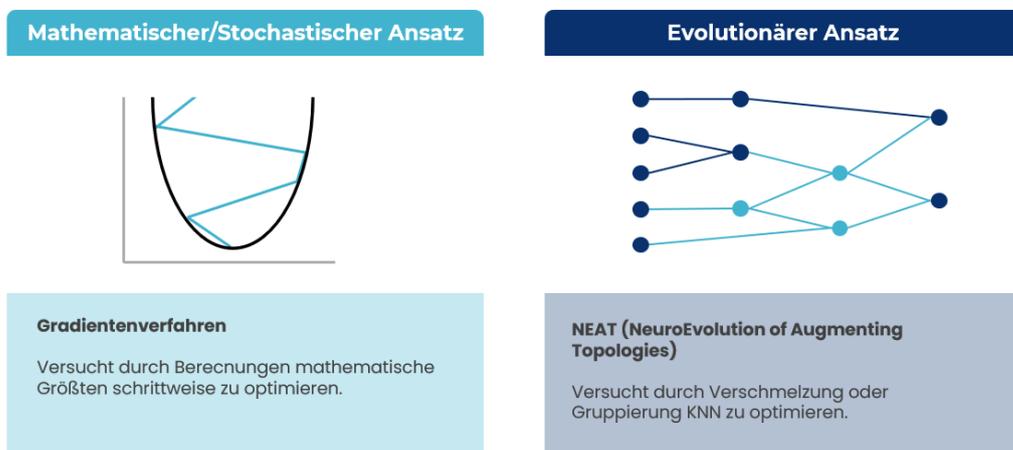


Abbildung: Lernalgorithmen künstlicher neuronaler Netze

### MATHEMATISCH/STOCHASTISCHE LERNALGORITHMEN

Als erstes sind mathematische bzw. stochastische Algorithmen zu nennen. Diese Algorithmen versuchen die Gewichte, Schwellwerte und/oder Propagierungsfunktionen eines KNN immer weiter zu optimieren. Solche Algorithmen werden meist im überwachten Lernen (Supervised Learning) angewandt.

#### Beispiel – Gradientenverfahren

Beim Gradientenverfahren wird eine Parabel genutzt, deren Wendepunkt (der Punkt ganz unten) das optimale Ergebnis des KNN darstellt. Die Ergebnisse werden dann anhand der Parabel ausgewertet und es wird erkannt, ob in die positive Richtung oder in die negative Richtung optimiert werden muss. Dies ist wiederum abhängig davon, ob man sich aktuell auf der linken oder rechten Seite der Parabel befindet.

Außerdem ob große oder lediglich kleinere Anpassungen gemacht werden können, was wiederum davon abhängig ist, ob man sich weiter oben oder weiter unten auf der Parabel befindet. So versucht man Stück für Stück näher an das Optimum zu gelangen.

## EVOLUTIONÄRER LERNALGORITHMUS

Evolutionäre Ansätze bedienen sich meist der natürlichen DNA-Weitergabe. Es werden KNN ausgewertet und die besten KNN miteinander verschmolzen bzw. weiterentwickelt.

### Beispiel – NEAT (NeuroEvolution of Augmented Topologies)

Das NEAT-Verfahren wurde 2002 in einem MIT Paper vorgestellt und bietet enormes Potenzial für einen recht geringen Aufwand. NEAT sieht nämlich die Struktur eines KNN nicht als fix an, sondern versucht zusätzlich zum Optimieren der Gewichtungen auch die Struktur zu optimieren, durch Hinzufügen und/oder Entfernen von Knoten und/oder Verbindungen. Diese evolutionären Algorithmen funktionieren primär bei bestärkendem Lernen (Reinforcement Learning) sehr gut. Hierbei werden mehrere KNN parallel laufen gelassen und wie bei der natürlichen Selektion in der Natur, würden immer die besten KNN „überleben“ bzw. ermittelt und diese können sich so immer weiter verbessern.

## 6. Fazit zu künstliche neuronale Netze

1. Künstliche neuronale Netze sind kein Hexenwerk. Häufig sind es einfache Abläufe und Strukturen, welche man durchaus einfach nachzuvollziehen sind. Meistens schreckt lediglich die relativ komplexe Mathematik ab. Hierbei ein Tipp von mir: Einfach ein Praxisbeispiel hernehmen und das damit durchspielen, dann wird die komplexe Mathematik verständlicher.
2. Künstliche neuronale Netze sind einer der wichtigsten Bausteine moderner KI und werden immer bedeutender, je weiter der Trend von künstlicher Intelligenz steigt.
3. Vor dem tatsächlichen Einsatz von KNN müssen grundlegende Fragen geklärt werden z.B. Was ist das Ergebnis des KNN? Denn davon ist die Lernart abhängig.

## 8. Quellen

1. Kriesel D. (2007). [Ein kleiner Überblick über Neuronale Netze](#)
2. Stanley K. O., Miikkulainen R. (2002). NEAT [Evolving Neural Networks through Augmenting Topologies](#)