

# Modul 7

## Neuronale Netze

"Jedes Neuron verfügt über ein sogenanntes **rezeptives Feld**, das ist eine Gruppe von **Neuronen** für deren **Aktivierung** es sensibilisiert ist."



# Über das Modul

Dieses Modul dient als grundlegende Einführung in das Thema Neuronale Netze (**NN**). Das Ziel dieser Lektion ist es, neuronale Netze vorzustellen und nicht nur zu zeigen, was sie können und wie sie verwendet werden, sondern auch einen Einblick zu geben, wie sie intern funktionieren.

## Ziele

Die Schüler\*innen können:...

- ...den grundlegenden Aufbau von Neuronalen Netzen erklären und die grundlegenden Elemente eines Neuronalen Netzes aufzählen (Teil 1)
- ...erklären wie die Eingabedaten dem Neuronalen Netz zugeführt werden können. (Teil 1)
- ...erklären, wie die Daten durch ein Neuronales Netzwerk fließen. (Teil 1)
- ...verstehen, dass die verschiedenen Schichten Verarbeitungsschritten entsprechen. (Teil 1)
- ...erklären, wie das Ergebnis des Verarbeitungsvorganges abgelesen werden kann. (Teil 1)
- ...erklären wo das Wissen eines Neuronalen Netzes gespeichert ist. (Teil 1)
- ...erklären welche Änderungen beim Lernvorgang passieren. (Teil 1)
- ...den Freiheitsgrad eines Neuronalen Netzwerks berechnen. (Teil 1)
- ...verstehen, dass es neben der Feed-Forward-Architektur eine Fülle weiterer Netzwerkarchitekturen für Neuronale Netze gibt.
- ...die Funktionsweise von Neuronen erklären. (Teil 2)
- ...die Funktionsweise einer Aktivierungsfunktion erklären. (Teil 2)
- ...erklären warum die Aktivierungsfunktion beschränkt werden muss und wie das umgesetzt wird. (Teil 1 und 2)
- ...verstehen, dass in verschiedenen Schichttypen verschiedene Aktivierungsfunktionen zum Einsatz kommen können. (Teil 2)
- ...den Begriff rezeptives Feld erklären. (Teil 2)
- ...verstehen, dass ein Neuronales Netz nur Ergebnisse liefern können, die aus der Datengrundlage ableitbar sind. (Teil 3)
- ...verstehen, dass kleinere Neuronale Netze mit überschaubarem Aufwand und mit verfügbaren Technologien relativ leicht umsetzbar sind. (Teil 3)
- ...verstehen, dass man für Neuronale Netze nicht unbedingt spezielle Bibliotheken oder andere Hilfsmittel einsetzen muss. (Teil 3)

...erklären wie Neuronale für grafische oder numerische Aufgaben eingesetzt werden können. (Teil 1-3)

## Agenda

<b>Zeit</b>	<b>Inhalt</b>
50 min	Theorie - Teil 1 - Grundlagen
30 min	Theorie - Teil 2 - Rezeptive Felder
20 min	Praxis - Simulator in der manuellen und automatischen Variante
25 min	Theorie - Teil 3 - praktisches minimales Beispiel
10 min	Praxis - Simulator
15 min	Praxis - praktische Übung

# Teil 1 – Grundlagen

Dieser Teil erklärt die Grundkonzepte und den Aufbau von neuronalen Netzen und zeigt die Funktionsweise am Beispiel einer Zeichenerkennung mithilfe eines einfachen Feed-Forward-Netzwerks.

## Einleitung und Vorstellung eines Feed-Forward-Netzwerkes

(Folien 2 – 5)

Neuronale Netze als vermutliche Grundlage aller Denkprozesse, künstliche Neuronale Netze als Versuch diesen Vorgang prinzipiell aber nicht im Detail nachzubilden. Neuronale Netze funktionieren grundsätzlich anders als herkömmliche Computer mit der üblichen von Neumann Architektur (Recheneinheit [Arithmetical Logical Unit], Steuereinheit [Control Unit], Speicher [Memory], Ein-/Ausgabeeinheit [I/O-Unit]). Begriffseinführung „Feed-Forward-Network“.

## Problembeschreibung Ziffern- und Zeichenerkennung

(Folien 2 – 5)

Warum ist Ziffern- und Zeichenerkennung anspruchsvoll? Was macht es so schwierig für herkömmliche Programmierung?

## Eingabedaten

(Folien 7 – 9)

Wie werden die Eingabedaten, also in unserem Fall die grafisch dargestellten Ziffern und Zahlen dem Netzwerk zugeführt? Die Pixelhelligkeiten werden als Aktivierungen von Neuronen der ersten Schicht (= Eingabeschicht) interpretiert.

## Schichtenaufbau

(Folien 11 – 14)

In der Feed-Forward-Architektur ist die erste Schicht die Eingabeschicht, die letzte ist die Ausgabeschicht und die Schichten dazwischen werden verborgene

Schichten genannt.

## **Datenfluss durch das Netzwerk**

(Folien 19 – 22)

Die Folien zeigen den Fluss der Daten durch das Netzwerk. Die Aktivierungen werden durch die Verbindungen weitergeleitet und werden zu den Aktivierungen der nächsten Schicht.

## **Verarbeitungsschritte**

(Folien 11 – 14)

Die Erkennung von Zeichen kann man sich schrittweise so vorstellen, dass zuerst einfache, grundlegendere Einheiten des Zeichens erkannt werden und dann größere, zusammenhängendere. Diese Einheiten werden jeweils farblich hervorgehoben. Ein Erklärungsansatz der Arbeitsweise von Neuronalen Netzen ist, dass es in einem erfolgreich trainierten Netz zu Arbeitsteilungen zwischen den Schichten kommt, sodaß die vorgeschalteten Schichten (linke Seite) auf die grundlegenden Einheiten und die nachgeschalteten Schichten (rechts) eher auf die größeren Einheiten reagieren.

## **Details der Durchführung und rezeptives Feld**

(Folien 23 – 25)

Der Zusammenhang zwischen Aktivierungen und Gewichtungen wird hier erklärt und der Begriff rezeptives Feld wird eingeführt. Jedes Neuron verfügt über ein sogenanntes rezeptives Feld, das ist eine Gruppe von Neuronen für deren Aktivierung es sensibilisiert ist. Das ist ein wichtiges Konzept zum Verständnis der Arbeitsweise von Neuronalen Netzen.

## **Beschränkungsfunktion**

(Folien 26 – 27)

Damit Werte von Neuronen nicht uferlos ins positive oder negative anwachsen (ein kleiner Bereich des Netzes würde genügen um das Gesamtergebnis unbrauchbar zu machen) kommt immer eine Beschränkungsfunktion zum Einsatz. Diese wird elementarer Bestandteil der Aktivierungsfunktion.

## **Vorerregung/Bias und gesamte Aktivierungsformel**

(Folien 28 – 31)

Erklärung welche Rolle der Bias in der Funktionsweise spielt und Angabe der vollständigen Aktivierungsformel bestehend aus der Beschränkungsfunktion, den Aktivierungen, den Gewichtungen und den Vorerregungen.

## Aufgaben und Lösungen

(Folien 32 – 33)

Anhand dreier Fragen werden noch einmal einige fundamentale Eigenschaften eines Neuronalen Netzwerks rekapituliert.

## Verweis auf andere Netzwerkarchitekturen und Links

(Folien 26 – 27)

Die Feed-Forward-Netzwerk-Architektur wurde hier vorgestellt weil sie eine sehr einfache und häufig verwendete Architektur ist. Die Grafik soll nur andeuten, dass eine Fülle anderer Architekturen eingesetzt und erprobt werden.

## Material

-  Neural\_Networks\_I\_-\_Basics.pdf

## Teil 2 – Rezeptive Felder

Dieser Teil vertieft den schon im Teil 1 eingeführten Begriff des rezeptiven Felds Anhand eines interaktiven Beispiels der Objekterkennung mit einer 2x2-Pixel-Kamera. Dieser Teil der Lektion ermöglicht mit dem Simulator ein interaktives Nachvollziehen der Arbeitsweise eines Neuronalen Netzerks.

### Kamera

(Folien 2 – 5)

Ein einfaches Neuronales Netz verleiht einer Kamera, die über einen 2x2-Pixel-Sensor verfügt die Fähigkeit vier verschiedene Muster zu erkennen. Wie schon zuvor bei der Zeichenerkennung ist das Problem anspruchsvoller als es zuerst klingt. Muster der gleichen Kategorie haben unter Umständen keine gemeinsamen Pixel. Wie schon zuvor wird die Helligkeit der Pixel zur Aktivierung der Neuronen in der Eingabeschicht.

### Arbeitsweise der Neuronen

(Folien 6 – 13)

In diesen Folien wird nochmals genau auf die Arbeitsweise der Neuronen eingegangen: Bildung der gewichteten Summe der Eingaben und Beschränkung der Werte. Die Details der Sigmoidfunktion bilden einen erweiterten Inhalt für fortgeschrittenere Schüler.

### Vervollständigung des Netzes

(Folien 14 – 16)

Die begonnene Schicht wird vervollständigt, eine zusätzliche Schicht vom gleichen Schichttyp wird hinzugefügt und schließlich eine weitere Schicht mit alternativer Aktivierungsfunktion. Für Aktivierungen und Gewichte werden in Folge nur 0, 1, -1 verwendet. Das ist eine drastische Einschränkung und funktioniert nur, weil das ein sehr einfacher Anwendungsfall ist. Es erleichtert aber das Rechnen sehr. Es ermöglicht uns den Rechengang durch das einfache Neuronale Netz zu verfolgen oder sogar selbst durchzuführen. In der Praxis wäre es keine gute Idee nur 0, 1, -1 zu verwenden. Das funktioniert nur bei diesem einfachen Beispiel.

# Alternative Aktivierungsfunktion

(Folie 17)

Die Details dieser Aktivierungsfunktion sind wieder als erweiterter Inhalt zu sehen. Beide benötigen ein sattelfestes Verständnis des Funktionsbegriffs und der grafischen Funktionsdarstellung. Wichtigste Eigenschaft der ReLU-Funktion in unserem Beispiel ist, dass sie keine negativen Werte durchlässt. Diese Eigenschaft wird im letzten Schritt zur erfolgreichen Ergebnisberechnung verwendet.

Die Anwendung der beiden Aktivierungsfunktionen vereinfacht sich erheblich durch den Trick, dass nur die Werte 0,1,-1 verwendet werden. Wichtigste Eigenschaft der hauptsächlich verwendeten Aktivierungsfunktion (Sigmoid) ist, dass Werte größer 1 auf 1 reduziert werden und Werte kleiner -1 auf -1. Wichtigste Eigenschaft der alternativen Aktivierungsfunktion (ReLU und leaky ReLU) ist, dass negative Werte auf 0 (oder nahe Null) reduziert werden.

## Das Netzwerk und die Kamera im Einsatz

(Folie 18)



Diese Folie zeigt das fertige System. Zugeführt und auch richtig erkannt wird ein vertikales Muster. Bei jedem Neuron ist das rezeptive Feld angeführt. Das ist für jedes Neuron die Menge der Eingabeneuronen von denen es direkt oder indirekt Signale erhalten kann. Die rezeptiven Felder wachsen von links nach rechts von der Größe 1 bis zur Größe 4 an. Jeweils ist auch von Bedeutung ob das Neuron auf positive (weißes Feld) oder negative Aktivierung (schwarzes Feld) in seinem rezeptiven Feld reagiert.

## Simulator

Im Simulator kann die Abarbeitung durch das Netzwerk durchgespielt werden. Den Simulator gibt es einmal in der manuellen und einmal in der automatischen Version. In der manuellen Version ermittelt und setzt man alle Aktivierungen selbst auf Grundlage der Eingabedaten (=Aktivierungen 0,1,-1) und Verbindungen (0,1,-1). Der automatisch arbeitende Simulator errechnet alle nötigen Aktivierungen und Ausgaben automatisch aus den Eingaben und ist in der Bedienung damit natürlich wesentlich bequemer handzuhaben. Die Herausforderung an den Bedienenden ist beim manuellen Simulator deutlich höher aber der Lerneffekt ist wesentlich größer. Begabte oder fortgeschrittene Schüler sollten zuerst mit dem manuellen Simulator vertraut gemacht werden.



## Material

-  Neural\_Networks\_II\_-\_receptive\_Fields.pdf
-  NN\_ii\_simulator.zip

## Teil 3 – praktisches minimales Beispiel

Dieser Teil stellt ein interaktives praktisches Beispiel eines aus nur 5 Neuronen bestehenden Neuronalen Netzes vor. Das Besondere an diesem Teil der Lektion ist, dass es sich zwar um ein minimales aber in allen Phasen in sich geschlossenes kleines Projekt handelt. Es werden keine externen Bibliotheken oder sonstige Hilfsmittel herangezogen. Trainingsdaten, Berechnungsvorgang, Training sind überschaubar, klar dokumentiert und im Pythonquellcode unmittelbar nachvollziehbar. Das Projekt dient sehr gut dem Zweck den Bereich der Künstlichen Intelligenz zu demystifizieren. Das Projekt kann auch als Grundlage für weitere Projekte dienen, zum Beispiel könnte man eine andere Datengrundlage verwenden oder das Netzwerk auch strukturell erweitern: zusätzliche Eingabe- und Ausgabemöglichkeiten, zusätzliche Schichten ... . Eine strukturelle Änderung bedingt aber auch eine Änderung des Trainingsvorgangs.

### Vorstellung des Neuronalen Netzwerks

(Folien 2 – 3)

Einleitung, grafische Darstellung und Erklärung des Aufbaus des Neuronalen Netzes.

### Datengrundlage und Ausgabe

(Folien 4 – 5)

Grafische Darstellung der Beziehung zwischen Körpergewicht, Körpergröße und Geschlecht. Damit soll nur der Zusammenhang dieser Größen verdeutlicht werden. Es handelt sich dabei aber nicht um die Trainingsdaten. Trainiert wird das Netzwerk lediglich mit 4 Datensätzen (siehe Python-Programm).

Durch die Überlappung der beiden Gruppen sind nicht immer eindeutige Resultate zu erwarten. Erklärung wie die Aktivierung der Ausgabeschicht als Ergebnis zu interpretieren sind.

### Rechengang

(Folie 6)

Hier wird der genaue Datenfluss durchs Netzwerk und der Rechengang erklärt.

# Beschränkungsfunktion

(Folie 7)

Hier wird die Notwendigkeit der Beschränkungsfunktion erklärt.

# Verweis auf Training und Simulator

(Folie 8)

Training stellt erweiterten Inhalt dar. Für fortgeschrittene Schüler stellt iii\_training.pptx aber einen guten Überblick dar um das Beispiel zu vervollständigen. Auch der Pythonquellcode gibt Aufschluss über die Details des Trainings.

Der Simulator ermöglicht es allen Schülern interaktiv Werte zu probieren und das Ergebnis abzulesen.

Über 2 Schieberegler werden die Eingaben vorgenommen. Die Aktivierung der Neuronen wird numerisch beim Neuron ausgegeben und zusätzlich mit Graustufen visualisiert. Schwarz steht hier für Aktivierung 0, weiß für Aktivierung 1. Die Stärke einer gewichteten Verbindung wird mit der Strichbreite symbolisiert. Positive Verbindungen sind schwarz, negative sind rot gefärbt.

# Praktische Übung

Die Teilnehmenden können mit dem erworbenen Wissen nun selbst eigene Beispieleingaben durchrechnen. Durch den minimalen, einfachen Aufbau des Netzwerkes ist eine manuelle Berechnung einfach durchführbar.



3 Wertepaare sind vorgegeben, für die die Teilnehmenden Ergebnisse berechnen können. In der untersten Zeile können eigene Werte verwendet werden.



# Quellenangaben und Verweis auf das Programm

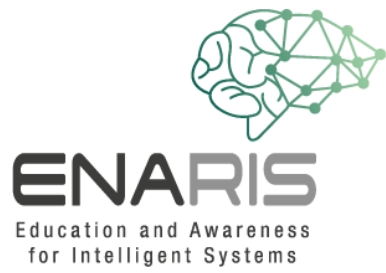
(Folie 9)

Das zugrundeliegende Pythonprogramm liegt in 2 Versionen vor, in denen verschiedene Aktivierungsfunktionen zum Einsatz kommen.

# Material

-  Neural\_Networks\_III\_-\_practical\_Example.pdf
-  NN\_iii\_simulator.zip

-  NN\_exercise.zip
-  NN\_additional\_content.zip



EUROPEAN UNION

