Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt
Fakultät Informatik und Wirtschaftsinformatik

**Seminararbeit**

# Bias of Neural Networks - Security implications

David Mödl & Sebastian Lober

16. Juni 2020

# Zusammenfassung

TODO

# Abstract

KI steht kurz für künstliche Intelligenz. Der Begriff KI ist jedoch irreführend. Eine 'KI' ist ein Programm, das versucht biologisches intelligentes Verhalten nachzuahmen. Die Begrifflichkeit Intelligenz in Verbindung mit Computern ist sehr umstritten, dennoch wird im Allgemeinen als auch in der Forschung das Wort 'Intelligenz' verwendet.
Aus diesem Grund und an Mangel an qualitativ hochwertigen Alternativen wird auch im Folgenden der Wortlaut KI verwendet, wohl wissend, dass die Bezeichnung nicht 100 Prozent korrekt ist.

# Inhaltsverzeichnis

# 1 Einführung

Künstliche Intelligenz(KI) oder auch artifizielle Intelligenz(AI) tritt allgegenwärtig in großen Teilen unserer Gesellschaft auf. Von Kaufvorschlägen auf Amazon, über Chat-Bots bis hin zu autonom fahrenden Autos spielt die KI eine große Rolle. Ein bekanntes Beispiel ist die Software älpha go", welche den internationalen GO Champion Lee Sedol besiegte[2]. Darüber hinaus ermöglicht die KI komplexe Sachverhalte zu simulieren und zu prognostizieren, wie zum Beispiel die vollautomatische Generierung hochaufgelöster, realistischer Videosequenzen auf der Grundlage simpler Eingaben[14].

Einerseits gibt es viele Erfolge, die für ein KI betriebenes System sprechen. Andererseits bestärken medienwirksame Verfehlungen, wie z.B. das frauenfeindliche Bewerbungssystem von Amazon[5], die Skeptiker solcher Systeme. Ziel dieser Arbeit ist es, die unterschiedlichen Ursprünge solcher algorithmischen Verzerrungen (engl. bias) zu erläutern und Präventionen zu schildern, welche diese vermeiden sollen.

Wir beginnen unsere Arbeit damit, Grundlagen für ein fundamentales Wissen spätere Kapitel aufzubauen. Danach möchten wir auf die Entstehung solcher Bias eingehen, die damit verbundenen Probleme und welche Präventionen gegen diese Fehlverhalten unternommen werden können.

# 2 Grundlagen

## 2.1 Bias

Wesentlicher Bestandteil der Arbeit ist das erläutern der "Biases", welche durch die Nutzung von künstlicher Intelligenz auftreten können. Das Wort Bias kommt aus dem Englischen und bedeutet im Wesentlichen:

1. Verzerrung – im statistischen Sinn als mittlere systematische Abweichung zwischen dem erwarteten („richtigen“) Modellergebnis und dem mittleren wirklich eingetretenen Modellergebnis.

2. Voreingenommenheit – je nachdem, wie wir die Welt aufgrund unserer Erfahrungen sehen, kommen wir zu unterschiedlichen Schlüssen.

Der Begriff Voreingenommenheit muss bei der Nutzung von KI vorsichtig behandelt werden, denn eine Maschine besitzt grundsätzlich keinerlei Vorurteile und weiß zu Beginn nicht was richtig oder falsch ist. Hier spricht man daher von einem Fehlverhalten oder einer Verzerrung, welche durch äußere Einflüsse wie z.B. dem Menschen verursacht wurden.

## 2.2 Künstliche Intelligenz

Künstliche Intelligenz (KI) oder englisch artificial intelligence (AI) ist der Oberbegriff für ein Teilgebiet der Informatik. Dieses Gebiet befasst sich nicht nur mit neuronalen Netzen, sondern generell mit jeglicher Form von maschinellen intelligenten Verhalten und dem maschinellen Lernen, siehe Abbildung 2.1. Generell wird bei der künstlichen Intelligenz versucht biologische Intelligenz auf einen Computer zu simulieren. Dies basiert meist auf simplen Algorithmen, wodurch die Begrifflichkeit 'Intelligenz' in Bezug auf einer Maschine öfter in Frage gestellt wird.

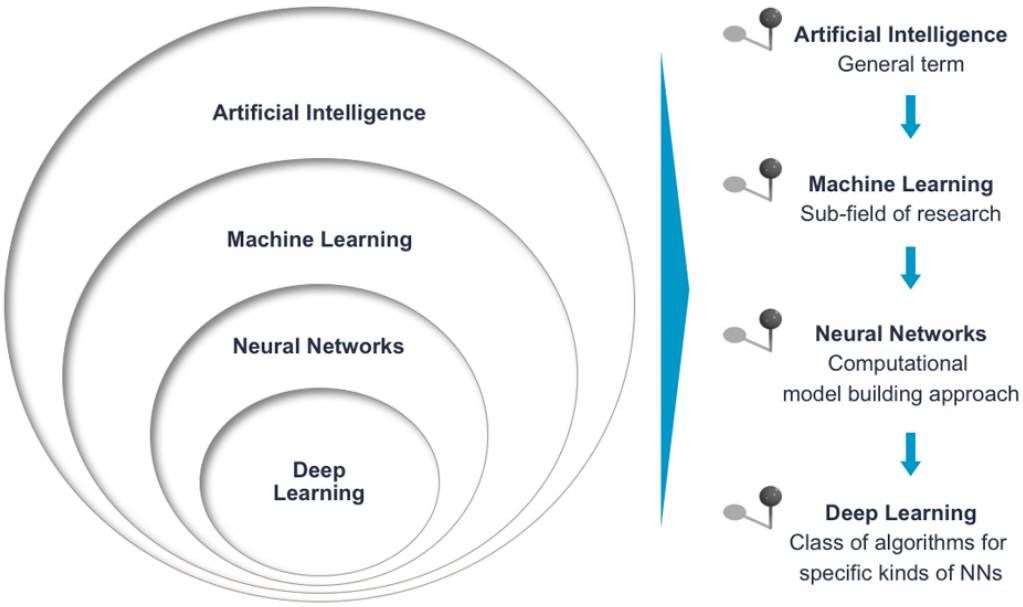


Abbildung 2.1: Verschiedene Abstraktionslevel von Artificial Intelligence in hierarchischer Ordnung

Damit ein Programm den Titel KI tragen darf, muss sie zum einen die Fähigkeit zu lernen besitzen, zum anderen die Fähigkeit intelligent Lösungen zu finden, auch bei nicht eindeutigen Eingaben.
KIs werden grob in zwei Kategorien aufgeteilt. Einmal die starke KI, die ebenbürtig mit Menschen zusammenarbeiten kann und schwacher KI, die lediglich das Arbeiten von Menschen unterstützen soll.

### 2.2.1 Maschine Learning

Machine Learning (ML) ist ein Teilgebiet der künstlichen Intelligenz. Es ist der Oberbegriff jeglicher Lernvarianten von KIs. Im Allgemeinen versucht eine KI neue Muster und Gesetzmäßigkeiten in Trainingsdaten zu erkennen, diese zu verallgemeinern und für neue Problemlösungen oder für die Analyse von bisher unbekannten Daten zu verwenden[12].

Diese Arbeit speziell konzentriert sich auf Deep Learning, welches eine Variante zum Trainieren von neuronalen Netzen darstellt.

#### Lernansätze

KIs bestehen aus vielen Algorithmen. Damit eine KI lernt, müssen diese Algorithmen angepasst werden. Dies kann überwacht geschehen mittels eines "Lehrers", der den Lernerfolg bei evolutionären Algorithmen-Änderungen überprüft.
Die andere Variante ist das unüberwachte Lernen. Hierbei erzeugt ein Algorithmus ein statisches Model aus den Trainingsdaten und erkennt Zusammenhänge zwischen den Daten und dessen Kategorie, sogenannte Features. Diese System gibt die Wahrscheinlichkeit zurück, zu welcher Kategorie die Eingabe gehört, abhängig von den erkannten Features der Eingabe.

Aus den Konzept dieser zwei Hauptlernvarianten wurden diverse Unterlernvarianten erstellt, wie teilüberwachtes Lernen, bestärktes Lernen oder unüberwachten Lernen ohne Kategorisierung.

### 2.2.2 Neuronale Netze

Künstliche neuronale Netze (KNN) bestehen aus künstlichen Neuronen, die untereinander verflechtet sind. Diese Konstrukte sind denen der Neuronen-Verbindungen im Nervensystem eines Lebewesens nachempfunden.
KNNs sind nicht dazu da das Nervensystem von Lebewesen nachzubilden, sondern abstrakt die Eigenschaften der Informationsverarbeitung und der Lernfähigkeit zu imitieren.

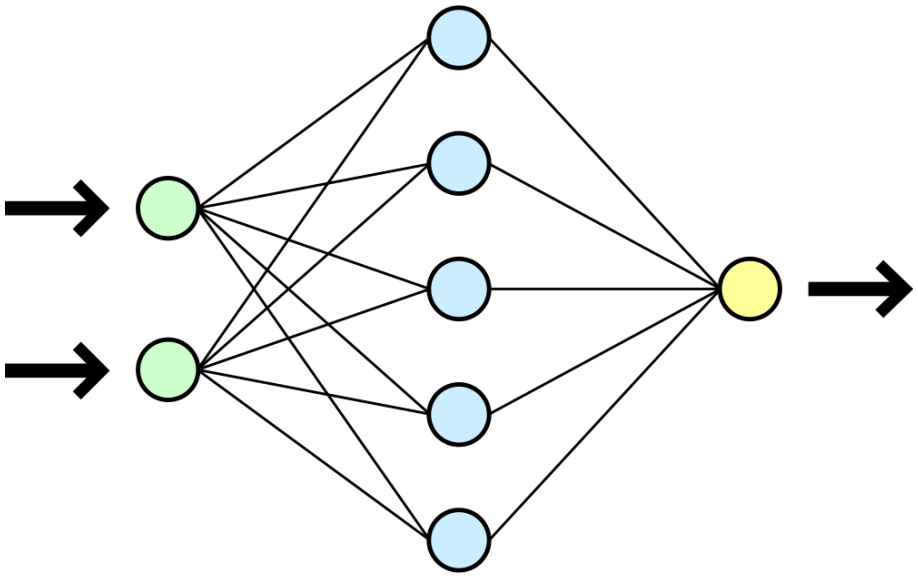

Abbildung 2.2: Vereinfachte Darstellung eines künstlichen neuronalen Netzes

KNNs sind meist in Schichten mit beliebig vielen künstlichen Neuronen aufgebaut. In der Regel besteht ein KNN aus drei Teilen die Eingangsschicht (grün), verdeckte Schicht (blau) im Englischen Hidden Layer und die Ausgabeschicht (gelb). In der Eingangsschicht fließen die Informationen in das Netz ein und in der Ausgabeschicht das Ergebnis der Berechnungen aus. Jede Schicht besteht aus beliebig vielen Neuronen je nach Komplexität des Zieles, die Hidden Layer sogar aus beliebig viele Schichten.

In der Regel arbeiten KNNs nach dem feedforward-Prinzip, bei dem die Informationen immer nur in eine Richtung fließt. Es gibt jedoch auch rekurrente Netze, bei denen durch rückgerichtete Kanten Rückkopplungen im Netz entstehen.
Die einfachste Netzstruktur ist das einschichtige feedforward-Netz. Dies besteht ohne Rückkopplungen aus nur einer Schicht, der Ausgabeschicht.

### Das künstliche Neuron

Künstliche Neuronen sind die Grundbestandteile eines künstlichen neuronalen Netzes. Ein künstliches Neuron ist die vereinfachte und abstrakte Version einer biologischen Nervenzelle und ist wie folgt aufgebaut.

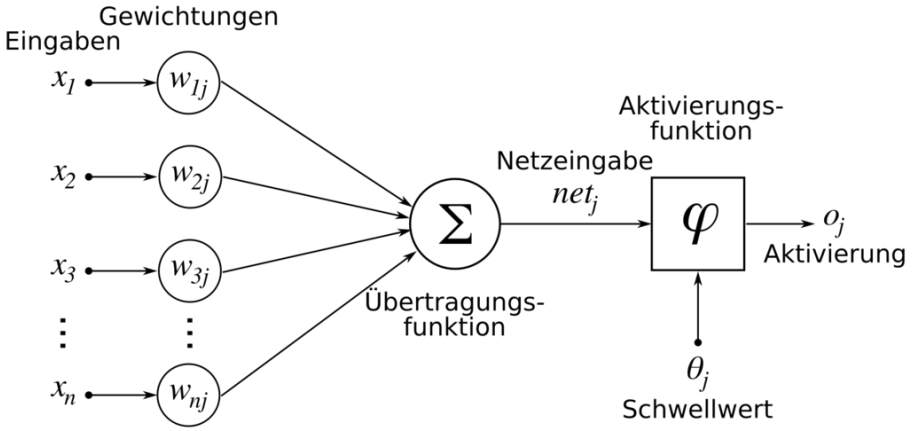


Abbildung 2.3: Ein künstliches Neuron

Ein künstliches Neuron besitzt $n$ Eingangskanäle und einen Ausgangskanal. $j$ repräsentiert hier die eindeutige Nummer des Neuron. Jede Eingabe $x_i$ besitzt ein dazugehörendes Gewicht $w_{1j}..w_{nj}$. Dieser spiegelt die Wichtigkeit der Eingabe wider, diese kann hemmend negativer oder erregend positiver Wert wirken. Die Übertragungsfunktion $\Sigma$ summiert alle multiplizierten Eingaben mit ihrem Gewicht und geht als Netzeingabe $net_j$ in die Aktivierungsfunktion $\varphi$ ein.

Ob das Neuron "feuert" oder kein Signal sendet, wird hier berechnet. Auf die Netzeingabe $net_j$ wird ein Schwellenwert $\theta_j$ addiert. Als mathematisch Vereinfachung wird der Schwellwert $\theta$ als $w_0$ bezeichnet und $x_0 = 1$ eingeführt und somit in der folgenden Formel immer auf die Netzeingabe $net_j$ addiert.

$$a = \sum_{i=0}^{n} x_i w_i$$

Die Variabel $a$ geht in die Aktivierungsfunktion $\sigma(a)$ ein, anhängig von dem Ergebnis wird das Neuron aktiv oder bleibt inaktiv. Der Ausgangskanal eines Neuron ist gleichzeitig ein Eingangskanal eines oder mehreren anderen Neuronen.

Dadurch dass ein Neuron mehrere Eingangskanäle besitzt, werden viele Eingangsinformationen auf ein Ergebnis reduziert. Durch mehrere Schichten und vielen Neuronen pro Schicht kann so eine große Menge an Daten schnell reduziert werden.

Jedoch muss jedes künstliche Neuron eines KNNs richtig eingestellt werden, damit das KNN dessen Ziel erfüllt. Eingestellt wird das Neuron durch Training. Trainieren bedeutet hier das Ermitteln der richtigen Werte für Gewichtungen und Schwellwerte, als auch das Einstellen der richtige Verbindungskombinationen der Neuronen untereinander. Diesen Prozess nennt man Deep Learning, eine Form von Machine Learning.

### 2.2.3 Deep Learning

Deep Learning ist eine Maschine Learning Variante, die speziell bei künstliche neuronale Netze eingesetzt wird. Beim Deep Learning werden die zahlreichen Zwischenschichten der Hidden Layer trainiert. Dabei wird eine umfangreiche und komplexe Struktur der Neuronen-Verbindungen aufgebaut. Wie das Programm endgültig die Aufgabe lösen soll, wird hierbei nicht vorgegeben, sondern wird bei diesem autonomen Prozess evolutionär ermittelt.

Ein künstliches neuronales Netz wird mit dem Zweck aufgebaut, eine bestimmte Aufgabe zu lösen. Extra dafür müssen Trainingsdaten aufbereitet werden. Diese Art der Daten, beispielsweise Bilder, soll das fertig trainierte Netz richtig interpretieren können. Trainings und Testdaten sind Daten, bei dem das korrekte Ergebnis bekannt ist.

Am Anfang ist das künstliche neuronale Netz meist mit relativ zufälligen Werten und Verbindungen vorbelegt. Trainingsdaten werden dem zu trainierenden Netzwerk an die Eingangsschicht übergeben. Diese durchlaufen das Netz. Das Ergebnis wird an der Ausgabeschicht überprüft. Die Ausgabeschicht besteht im einfachsten Fall aus zwei Neuronen, beispielsweise "Gesicht erkannt" oder "kein Gesicht", an dem gemessen wird, wie viel gewichtete Signale ankommen. Diese summiert, ergeben die Endergebnisse der Be-

rechnungen und das Neuron mit dem höchsten Gewicht, ergibt die Antwort.

Neuronale Netze sind für Menschen ab einer gewissen Größe nicht mehr nachvollziehbar. Somit kann nur die Eingabe mit der Ausgabe mit Hilfe von Testdaten verglichen werden, um auf die Korrektheit der Aufgabenlösung zu prüfen. Ziel ist es mit möglichst vollständigen Trainingsdaten das Netzwerk so einzustellen, dass diese nicht nur die Trainingsdaten und Testdaten richtig beantwortet, sondern auch unbekannt Daten korrekt interpretiert.

Um eine Aufgabe, wie "Gesicht in Bild erkennen", zu trainieren, wird nicht nur ein Netz mit Zufallswerten und Verbindungen generiert sondern tausende. Alle werden mit den gleichen Testdaten geprüft und für jedes Netz ein Mittelwert über die Korrektheit der Antworten erstellt. Da alle Netze Initial mit Zufallswerten belegt sind, haben die meisten Netze eine Erfolgsrate von ca. 50 Prozent. Die Netze mit den höchsten Erfolgsraten, mit beispielsweise mehr als 60 Prozent, werden behalten, der Rest wird verworfen. Die erfolgreichsten Netze werden mehrfach kopiert und bei jedem Kopiervorgang individuell leicht verändert und erneut getestet. Die Besten werden wieder genommen und leicht modifiziert kopiert und schlechteren verworfen.

Nach einer gewissen Anzahl an Iterationen entscheidet das Netz nicht mehr willkürlich, sondern scheint intelligent die Aufgabe zu lösen. Dieser Iterationsschritt kann unendlich oft laufen, jedoch empfiehl es sich, je nach Anwendungsfall, ab einer gewissen Erfolgsrate das Training zu beenden oder neue Trainings- und Testdaten zu verwenden. Als Ergebnis des Prozesses erhält man durch Deep Learning ein trainiertes künstliches neuronales Netz, das im Allgemeinen als KI bezeichnet wird.

### 2.2.4 Loss-Funktion

Alle Modelle benötigen eine Loss-Funktion(Verlustfunktion), um später trainiert werden zu können. Die Verlustfunktion bestimmt die Differenz zwischen der Prognose, die das Modell liefert, und dem vorgegebenen Label. Sie muss für die Menge aller Daten berechnet werden und beschreibt damit, wie gut das Modell die Trainingsdaten abbildet.

### 2.2.5 Informationsverlust zwischen Schichten

1. Erste Schicht verbunden mit letzter Schicht
a. Eingabe hoher Einfluss auf Endergebnis
2. Jede Schicht nur Verbindung zu der Nächsten
a. Hohe Informationsverlust

## 2.3 Architekturen

Arten
1. Full Connected
2. CNN
3. ResNet
4. Natural Network Connection
5. Dropout
6. ...

## 2.4 Daten

Erst durch eine Kombination aus Algorithmen und Daten wird die Entscheidungsfindung unterstützt. Wie ein menschlicher Entscheider können auch Algorithmen wegen unvollständiger oder fehlerhafter Daten zu fehlerhaften Entscheidungen gelangen. Eine gute Datenqualität und viele Daten ist daher unabdingbar. Auch die richtige Unterteilung zwischen Test- und Trainingsdaten spielt darf bei maschinelles Lernen nicht vergessen werden.

### 2.4.1 Datenaufbereitung

Der Erfolg einer KI hängt von den Daten ab mit welchen dieses Modell trainiert und getestet wird. Zuerst müssen Daten gesammelt werden, welche zu dem Zweck der KI passen. Soll eine System Äpfel und Birnen unterscheiden, müssen Bilder gesammelt werden, welche Birnen und Äpfel enthalten.

Um nun ein Modell trainieren zu können, müssen die Daten in Trainings- und Testdaten unterteilt werden. Die Testdaten werden nicht für das Training benutzt, sondern dienen dazu, das fertige System gegen unbekannte Daten zu testen, mit welchen es zuvor nicht trainiert wurde. Meist werden 10-20% der Daten als Testdaten reserviert. Die Trainingsdaten werden genutzt um das Modell zu trainieren.

Bei überwachtem Lernen müssen die Daten zuvor ein Label enthalten, welches der KI mitteilt um was es sich bei diesem Datensatz enthält, z.B. muss ein Birne auch als solche gekennzeichnet werden.

Ein Datensatz besteht grundsätzlich aus 2 Elementen. Beim überwachten Lernen wird wie bereits genannt, das Label welches man Vorhersagen möchte, benötigt. Anhand un-

seres Beispiels wäre ein Attribut welches dem Programm sagt, ob es sich auf dem Bild um eine Birne oder einen Apfel handelt. Damit die Maschine leichter Muster erkennen kann, werden dazu weitere Eigenschaften (z.B. Farbe, Größe) benötigt. Diese helfen dem Modell z.B. das Label richtig vorherzusagen.

### 2.4.2 Quantität

Je breiter, also je mehr unterschiedliche Eigenschaften in den Datensätzen existieren, umso komplexer wird das Modell. Und diese Komplexität der Probleme erfordert, dass die Menge an Daten entsprechend groß sein muss, damit das zu trainierende System immer besser reagiert.

Ein Beispiel hierfür findet sich in der Autoindustrie. Beim autonomen Fahren müssen Daten von Laser-, Kamera- und Radarsensoren im Auto zuverlässig und schnell verarbeitet und zusammengeführt werden. Dadurch verfügt das Fahrzeug jederzeit über ein präzises Abbild der realen Verkehrsbedingungen, kann sich selbst in diesem Umfeld verorten und darauf basierend in jeder Fahrsituation die richtige Entscheidung treffen [4].

Anhand dieses Beispiels erkennt man die Wichtigkeit der Quantität der Trainingsdaten, da die Anzahl möglicher Situationen im Straßenverkehr prinzipiell unendlich ist. Um gleichartige Strukturen im Verkehrsgeschehen zu erkennen, sind viele Trainingsdaten erforderlich, die ein immer genaueres Bild ergeben.

### 2.4.3 Qualität

Auch die Qualität der Daten ist essentiell, denn meist sind die Daten zu Beginn nicht gelabelt. Daher ist das Sammeln und vorbereiten das wichtigste beim maschinellen Lernen. Dabei sollten die Daten zusammenhängend und vollständig sein.

Dass heißt wenn nun die KI auf Bildern z.B. einen Panzer erkennen soll[9], muss die KI mit Bildern von Panzern trainiert werden und mit Bildern ohne Panzern.

# 3 Bias Entstehung

Bei der Nutzung von KI System können Verzerrungen(Bias) bzw. Fehlverhalten entstehen, diese können unterschiedlicher Natur sein und an unterschiedlichen Stellen, in der in Abbildung 3.1 gezeigten, vereinfachten Machine Learning Pipeline, auftreten. Dabei möchten wir auf die Daten eingehen, welche bei der Eingabe zu Bias führen können und menschliche Fehler verdeutlichen, welche bei der Verarbeitung und der Ausgabe auftreten können. Zuletzt möchten wir Adversial Attacks ansprechen, welche zu weiteren Verzerrungen führen können.

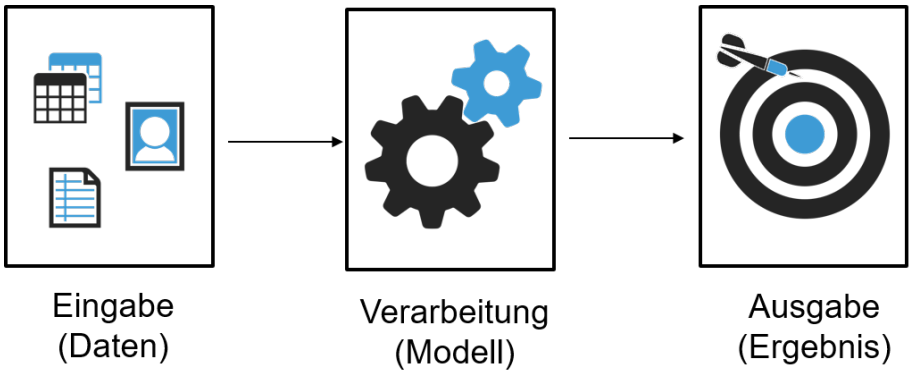


Abbildung 3.1: Machine Learning Pipeline: Eingabe, Verarbeitung, Ausgabe

## 3.1 Daten

Im ersten Kapitel möchten wir erläutern welche Bias, durch z.B. eine schlechte Datenqualität oder Datenquantität, entstehen können.

### 3.1.1 Unvollständigkeit der Daten

Zuerst möchten wir auf ein Problem aufmerksam machen, welches zu Verzerrungen führt, anhand des Beispiels aus 2.4.3.

Das Pentagon hatte eine Software angefordert, welche Panzer in der Natur erkennen sollte. KI Forscher haben daraufhin ihr neuronales Netz mit Fotos von getarnten Panzern trainiert, und mit Landschaftsfotos ohne getarnte Panzer. Dadurch sollte gewährleistet werden, dass die Software Panzer auf unbekannten Bildern erkennt.

Bei internen Test funktionierte das System sehr gut, doch bei den realen Test schien die Software nicht zu funktionieren. Das Problem hierbei lag daran, dass diese KI zuvor mit Trainingsdaten gefüttert wurde, welche nur bei schönem Wetter fotografiert wurden (siehe Auch die internen Testdateien erfüllten dieses Kriterium. Die realen Test hingegen wurden bei jedem Wetter ausgetragen.

Abbildung 3.2: Panzer bei bewölkten Wetter vs Landschaft ohne Panzer bei schönem Wetter

Die Software hatte somit trainiert schlechtes und gutes Wetter auseinander zu halten und nicht Panzer zu erkennen. Das Problem hierbei lag an unvollständigen Daten, hier wurden zu wenig unterschiedliche Fälle getestet und dadurch wurde ein Bias erzeugt, welcher die Nutzung der Software unmöglich machte.

### 3.1.2 Garbage in - Garbage out

Eine Maschine kennt grundsätzlich keinen Unterschied zwischen Schwarz und Weiß, Mann und Frau oder Jung und Alt. Erst durch eine KI lernt eine Maschine Verhalten und Muster kennen. Hierfür werden wie bereits in 2.4 beschrieben Daten benötigt, welche die richtige Qualität benötigen um die Ergebnisse zu bestimmen.

Bleiben fehlerhafte Daten unentdeckt, wird ein System trainiert, welches in Zukunft falsche Ergebnisse liefern wird. Das Beispiel aus 2.4.3 erläutert dieses Problem. Möchte ich ein System trainieren, welches Panzer identifizieren kann, muss ich diesem System beibringen Panzer zu erkennen. Füttere ich dieses nun mit Autos und markiere diese versehentlich als Panzer, identifiziert das System daraufhin diese nicht als Autos sondern als Panzer.

Bei der traditionellen Datenanalyse können solche schlechte Daten nachträglich entfernt werden. Hat allerdings eine Maschine durch maschinelles Lernen etwas gelernt, wird es schwer dies wieder zu verlernen. Denn ab einem Gewissen Grad wird es nahezu unmöglich, herauszufinden, auf welche Datenelemente die Vorhersagen basieren. Ähnlich wie beim menschlichen Gehirn.
Baut unser erlerntes Wissen in Teilen auf falsche Grundannahmen oder Informationsbausteinen auf, verliert der ganze Komplex seinen Wert und wir müssen von neu alles erlernen.

Diese Problem wird in der KI als "Garbage in - Garbage out"(Müll rein, Müll raus) bezeichnet.

### 3.1.3 Bias in Trainings-/Testdaten

Ca. 4 Jahre entwickelte Amazon einen Algorithmus, welcher unter mehreren Bewerbungstexten automatisch die besten Bewerber herausfiltern sollte. Dabei bezog die Software sich auf voran gegangene Bewerbungen, verdeutlichte dabei aber ein grundlegendes Problem des maschinellen Lernens in seiner aktuellen Form.

Der Algorithmus hatte mit den Datensätzen der angenommen Bewerber trainiert und lernte daraus welche Eigenschaften Amazon bevorzugt. Weil das Unternehmen aber Teil einer von Männern dominierten Industrie ist, waren in den zugrunde gelegten vergangenen zehn Jahren vor allem Männer eingestellt worden. Daraus resultierte, dass Frauen grundsätzlich schlechter bewertet wurden, selbst ohne die Angabe eines Geschlechtes und dieses z.B. nur durch Frauenvereine erkennbar wurde. Die KI blieb diesen Auswahlkriterien treu und bevorzugte vorwiegend Männer.[5]

Die Hoffnung solcher Anwendungen liegt eigentlich darin, Vorurteile zu vermeiden und Prozesse fairer zu gestalten, da eine Maschine wie in 3.1.2 bereits genannt keine Unterschiede kennt. Doch in diesem Beispiel beinhalteten die Trainingsdaten bereits Vorurteile und führten somit zu einem Fehlverhalten des Systems.

An diesem Beispiel wird deutlich wie zentral die Daten für eine KI sind. Meist ist es nicht möglich Daten zu finden, welche nicht bereits menschliche Bias enthalten. Solch verzerrte

Trainingsdaten, werden unter Bezug auf ihre Zusammensetzung auch als WEIRD Samples (western, educated, industrialized, rich and democratic societies) bezeichnet[6].

### 3.1.4 Under-/ Overfitting

Ein komplexer Datensatz wie in Absatz 2.4.2 kann beim überwachten Lernen dazu führen, wenn ein Modell zu gut oder zu schlecht trainiert wurde, nutzloses Wissen aufzubauen oder aus einem vorhandenem Trainingsdatensatz keine Relevanten Lerninformationen ziehen zu können. Diese Phänomene werden als Overfitting (Überanpassung) und Underfitting(Unteranpassung) bezeichnet.

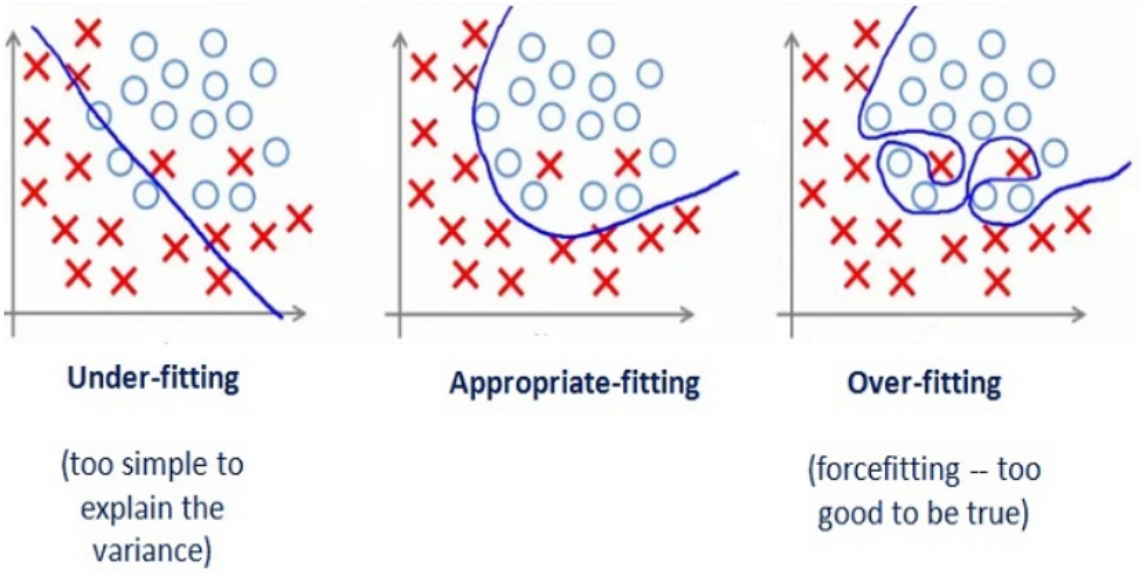


Abbildung 3.3: Over- und Underfitting

Die Abbildung 3.3 zeigt 3 Graphen, die das Problem erläutern. Jeder Graph visualisiert, wie gut das Modell anhand der Trainingsdaten die einzelnen Punkte in 2 Cluster unterteilt.

Das Modell ist unter angepasst, wenn wie im linken Graphen das Model zu schlecht mit den Trainingsdaten abgestimmt wurde. Hier wurden zu wenig Punkte richtig abgedeckt und es konnten zu wenige Muster zwischen den Eigenschaften und dem Label erkannt werden. Solche Modelle neigen zu hohen Verzerrungen der Vorhersagen.

Der Graph auf der rechten Seite hingegen sagt alle Punkte richtig vorher. Unter dieser Annahme könnte man denken es wäre ein sehr guter Graph. Hier spricht man allerdings von einer Überanpassung, da das Modell zu gut auf die Trainingsdaten abgestimmt ist,

und bei Testdaten schlechte Ergebnisse liefern wird. Ein Grund dafür ist, dass alle Punkte mit vorher gesagt werden, auch diese die Grundrauschen oder Outliner sind. Solche Modelle besitzen eine hohe Varianz der Vorhersagen[1].

Ein Beispiel für z.B. Overfitting ist, wenn durch maschinelles Lernen ein System einen Schrank auf einem Bild erkennen soll. Nun werden zu viele Eigenschaften mit erfasst, welche jede Art an Schränken erkennen soll, auch solche die z.B. Autos ähnlich sehen. Daraufhin könnte es passieren, dass das Modell die Trainingsdaten sehr gut erkennt und sobald man anfängt mit realen Daten zu testen, auf welchen auch Autos vorkommen können, könnte das Programm Autos als Schränke erkennen, da das Modell zuvor zu komplex war.

Am besten ist somit der mittlere Graph, welcher Outliner und das Grundrauschen ignoriert und eine gute Balance zwischen Verzerrung und Abweichungen besitzt.

### 3.1.5 Ähnlichkeit der Daten

Abbildung 3.4: Hund oder Bagel?

Es gibt Bilder, welche selbst für den Menschen schwer zu differenzieren sind (Beispiel Abbildung 3.4). Ist nun ein Mensch nicht aufmerksam und sagt der Maschine das ein Bagel ein Hund ist, erlernt die Maschine falsche Bilder und sagt zukünftig auch Bagel als

Hunde vorher. Aber auch durch richtiges Kennzeichnen der Bilder durch den Menschen, können solche Bilder zu Verzerrungen führen.

Ein weiters Beispiel hierfür kommt aus der Autoindustrie. In Testreihen für autonomes Fahren stuften die Probanden (weil sie nicht aufmerksam waren) immer wieder das bestimmte Bild eines Menschen als Bild einer Tonne ein. Das System reagiert folgerichtig und wertet in einer kritischen Verkehrssituation das Überfahren einer (vermeintlichen) Tonne als verhältnismäßige Alternative, die möglichst wenig Schaden anrichtet [8].

## 3.2 Menschliche Fehler

Den Faktor Mensch darf man bei der Bias Entstehung nicht vergessen. Konzeptionelle Fehler mangels an Wissen oder einem Missgeschick heraus fördern Fehlverhalten. Markante Defizite können ein funktionale KI-Entstehung komplett verhindern, sind in der Regel noch die besseren Missstände. Denn kleine Mängel, die nicht sofort auffallen, können in Produktion fatale Folgen haben.

### 3.2.1 Falsche Zielsetzung

Die Zielsetzung eines KNNs ist ein nicht zu unterschätzender Teil bei der KI Entwicklung. Setzt man hier den falschen Grundstein, können sich vermeidbare Fehlverhalten einer KI entwickelt.

#### Falsche Ziel Definition

Wenn man eine Aufgabe hat und diese Automatisieren möchte, greift man heute zutage gerne zur KI. KI ist modern und in aller Munde. Jedoch sollte einem im Klaren sein, dass künstliche Intelligenz kein Allheilmittel ist.

#### Zu viele Ziele

Viele Köche verderben den Brei. Diese Weisheit kann auch auf die Ziele von KNNs umgemünzt werden. Denn je mehr Ziele ein KNN hat, desto größer und komplexer muss ein KNN sein, um alle Fälle abdecken zu können. Je komplexer ein KNN ist, desto größer ist die Wahrscheinlichkeit, dass Fehler passieren. Auch ist sich das Netz deutlich unsicherer bei seinen Entscheidungen.

Wenn man eine nahe zu perfekte KI mit einer Aufgabe erweitern möchte, hat das meist zu Folge, dass die KI nach der Erweiterung zwar mehr kann, jedoch seine ehemalige Hauptaufgabe nicht mehr so gut meistert wie zuvor.

### 3.2.2 Falsche Modearchitektur Wahl

Grundgedanke 'Je mehr Neuronen und Schichten eine Netz hat desto besser' ist falsch.

### 3.2.3 Falsches Lernverhalten

KI lernt einfachste Unterschiede
i. Nicht Unterschied zwischen Auto und Boot sondern Untergrund(Wasser/Land)
ii. Sehr Fehleranfällig z.B. Auto fährt durch flaches Wasser (KI -¿ Boot)

# 4 Sicherheitsprobleme durch BIAS

## 4.1 Gefahren für Maschinen

Google KI -¿ Kühlung von Maschinen

## 4.2 Gefahren für Menschen

Tesla Autopilot
Etc.

## 4.3 Angriff auf KI

Ist eine Künstliche Intelligenz sehr gut trainiert, erzielt sie in ihrem Aufgabebereich überragende Ergebnisse. Doch auch nahe zu perfekte KIs sind nicht unfehlbar. Durch methodisch gestaltete Störungen der Daten kann eine KI bewusst getäuscht werden und die Computer-Wahrnehmung in die Irre führen. Diese Art der Ausnutzung von KI Schwächen nennt man Adversarial Attacks, zu deutsch gegensätzlicher Angriff.

#### Adversarial Attacks

Bildererkennungssoftware ist besonders anfällig von dieser Art von Angriff. Hierbei legt man über das Bild bestimmte Pixelmuster, die für das menschliche Auge in der Regel im Gesamtbild untergehen. KIs hingegen registrieren jegliches Muster und nehmen sie in ihre Berechnungen mit auf. Als Ergebnis werden Bilder zum Teil massiv fehlinterpretiert. Im Folgenden Angriff wird mit einer Brille und dessen speziellen Muster, eine KI überlistet. Im linken Bild sieht man die amerikanische Schauspielerin Reese Witherspoon. Dieses Bild wird auch korrekt als sie selbst von der KI erkannt. Setzt man nun ihr

Abbildung 4.1: Reese Witherspoon links korrekt identifiziert. Mitte Bild mit Adversarial Attack. Rechts Klassifizierung von Mitte als Russel Crowe

eine spezielle Brille auf, beginnt das neuronale Netz Fehler zu begehen. Plötzlich erkennt die KI Reese Witherspoon mit Brille als Russel Crowe (Bild rechts).

# 5 Prävention

## 5.1 Passender Algorithmus zu Daten

Im Abschnitt 3.1.2 wurden die Schwierigkeiten genannt, was passiert wenn ein Modell mit fehlerhaften Daten trainiert wurde und ab einem gewissen Grad es schwer wird diese fehlerhaften Vorhersagen zu verstehen und abzuändern.

In der Annahme das nun dieses System richtig trainiert wurde, aber die zeitliche Komponente nicht beachtet, wie z.B. dass ein Panzer in Zukunft anders ausschauen könnte, kann es zu dem Problem kommen, das in Zukunft ein Panzer nicht mehr erkannt wird. Wie Stöcker 2019([13]) bereits erwähnte, benötigt eine sich ändernde Gesellschaft auch kontinuierliche weiterlernende Algorithmen.

Solch ein Lernprozess wird auch als *adaptive learning* bezeichnet, womit dem Effekt sich ändernder, d. h. dynamischer Datenumgebungen, concept drift bezeichnet, Rechnung getragen wird [3]. In DL-Algorithmen ist solche eine Umsetzung nicht trivial, aber das Beispiel von Sahoo et al. [11] zeigt auch die Machbarkeit auch in diesem Feld.

Auch bei einem für Diskriminierung anfälligen Umfeld, wie im Beispiel von Abschnitt 3.1.3, muss besondere Sorgfalt bei der Wahl des passenden Algorithmus bedacht werden. Oftmals ist es unzureichend einzelne Attribute aus den Eingangsdaten auszuschließen, um beispielsweise eine algorithmische Diskriminierung bestimmter Personengruppen zu verhindern. Dies genügt jedoch in der Regel nicht, denn wie das Beispiel von Amazon zeigt, kann der Algorithmus Rückschlüsse anhand anderer Attribute ziehen(z.B. durch Frauenvereine).

In diesem Kontext sollten System bevorzugt werden, welche in der Entwicklung solcher Algorithmen zusätzliche Einschränkungen einführen, um eine Diskriminierung aufgrund eines bestimmten Attributs, zu verhindern [7].

## 5.2 Nur ein Ziel

Nimmt man das Beispiel aus dem Abschnitt 3.1.4 mit den Schränken und möchte der Maschine alle möglichen Schränke beibringen, kann dies wie in diesem Absatz bereits erläutert zu Overfitting führen. Das Ziel aus diesem Beispiel war zu komplex. Hier hätte man das Ziel einfacher halten und die Daten generalisieren sollen. Dass heißt sich auf Schränke beschränken die sofort als solche erkannt werden. Falls ein Schrank ein komplizierteres Aussehen hätte und durch den Algorithmus nicht erkannt worden wäre, wäre das besser als wenn der Algorithmus viele Autos als Schränke erkennt.

Solche Probleme treten häufig auf, da die Ziele meist zu komplex gehalten werden. Das führt zu einer großen Fehleranfälligkeit und somit zu Verzerrungen. Deswegen sollten die Ziele klar und einfach gehalten werden, um unnötige Komplexität zu vermeiden.

## 5.3 Verfahren zum Validieren

Damit Vorurteile, Outliner oder falsche Bilder in den Daten nicht zu Problemen führen sollte diese zuvor überprüft werden. Beispiele dafür könnten folgende sein:

1. Lektorat oder Peer-Review

2. Das Vier-Augen-Prinzip (gegenseitige Kontrolle)

3. Mehrheitsentscheide bei unterschiedlichen Ergebnissen

Zudem sollte das Modell ausreichend validiert werden und die Performance gemessen werden. Denn wenn der Algorithmus z.B. bei der Diagnose von Krebs helfen soll, sollten die Ergebnisse nicht falsch sein.

Würde man einen Datensatz in Trainings und Testdaten unterteilen, könnte es dazu kommen, das die Daten so perfekt aufgeteilt werden, dass er in dieser Aufteilung sehr gute Ergebnisse liefert. Falls nun aber ein Fall nur in den Testdaten vorhanden ist und somit nicht antrainiert wurde, jetzt aber in der realen Welt benötigt wurde, können die Ergebnisse trüben und das Modell könnte zu Problemen führen.

Eine Möglichkeit um das zu verhindern, bietet die Cross Validierung. Anstelle eines extra Datensatzes, welcher zum Validieren der Daten genutzt wird, wird der Datensatz in viele kleine Blöcke unterteilt. In der Abbildung 5.1 sind es 5. Danach werden 5 Durchläufe gemacht, in denen jeweils mit 4 unterschiedlichen aus diesen 5 Blöcken trainiert und mit einem davon getestet wird. Bei jedem Durchlauf entsteht ein Score,

welche angibt, wie gut das Modell ist. Am Ende wird dann ein durchschnittlicher Score über diese 5 einzelnen gebildet, welcher die gesamt Performance des Modells angibt.

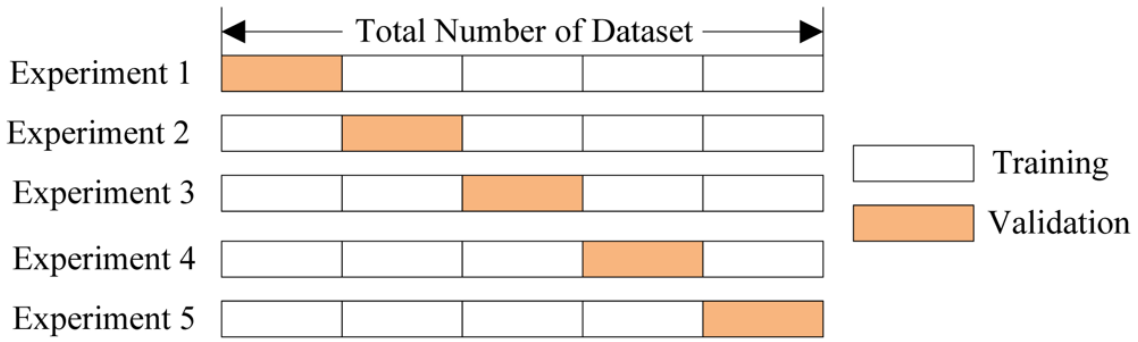


Abbildung 5.1: 5 fache Cross Validierung

Mit diesem Score lässt sich das beste Modell ermitteln und man verhindert unbewusste Nebenwirkungen.

## 5.4 Over- und Underfitting vermeiden

Underfitting und Overfittung sind oft Fehler der Test-/Trainingsdaten Aufbereitung. Um Overfitting zu verhindern, kann ein einfacheres Ziel wie bereits in Absatz 5.2 beschrieben, ausreichen. Dennoch gibt es Fälle wo das Ziel zu komplex ist oder zu wenig Daten vorhanden sind, um Underfitting oder Overfitting zu verhindern.

### 5.4.1 Underfitting

Anhand einer polynominialen Regression kann man verdeutlichen, wie Underfitting vermieden werden könnte. Um dies zu erreichen, muss der erwartete Wert einer Variable als Polynom n-ten Grades modelliert werden, welches dann wiederum das allgemeine Polynom ergibt. Wenn wir den Grad $n$ des Polynoms erhöhen, nimmt der Trainingsfehler generell ab. Die allgemeine Gleichung eines Polynoms ist folgende:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_n^n + \epsilon$$

Aber hier muss aufgepasst werden. Denn wie in der Abbildung 5.2 zu sehen sinkt zur gleichen Zeit der Cross Validierungsfehler solange wir n bis zu einem gewissen Punkt erhöhen, nach diesem Punkt steigt der Fehler und Overfitting wird das Problem.

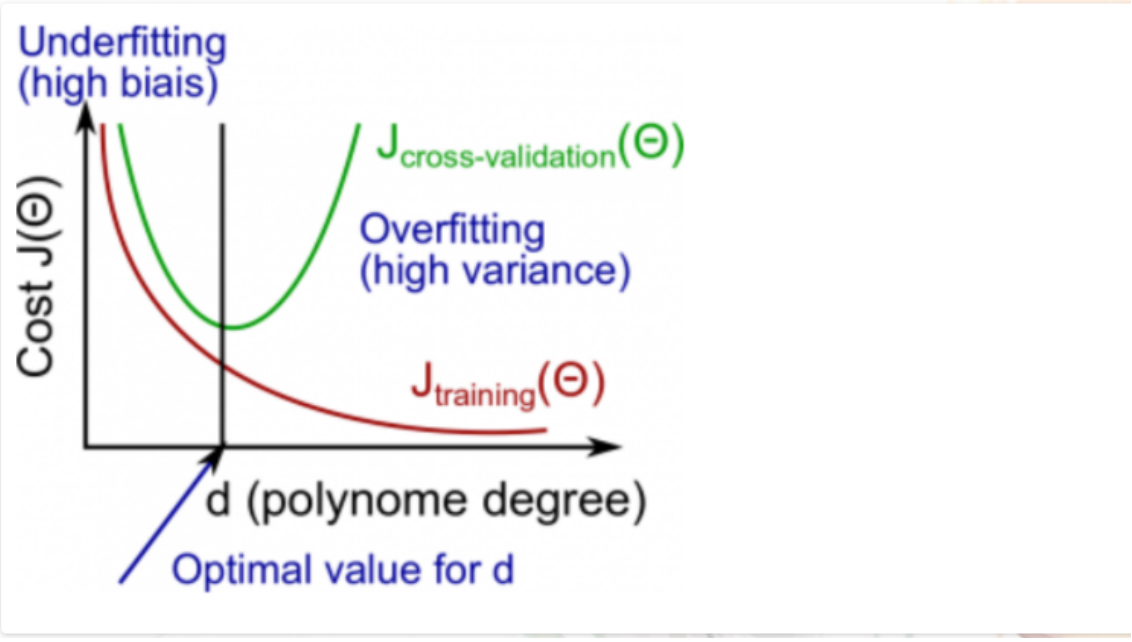


Abbildung 5.2: Polynominale Regression

### 5.4.2 Overfitting

Eine Möglichkeit um Overfitting zu vermeiden bietet die bereits in Abschnitt 5.3 Cross Validierung. Eine andere kann durch das erhöhen der Flexibilität des Models erreicht werden. Um die Flexibilität zu erhöhen können Regulierungstechniken benutzt werden.

Durch die Regulierung wird wenn die Komplexität eines Models zunimmt, eine Strafe angehängt. Der Regulierungsparameter ($\lambda$) bestraft alle Parameter mit Ausnahme des markierten Bereichs in Abbildung 5.3, sodass das Modell die Daten verallgemeinert und nicht überanpasst.

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n}\theta_j^2\right]$$
$$\min_\theta J(\theta)$$

Abbildung 5.3: Regularisierung in Loss Funktion

Wenn in der Abbildung 5.3 die Komplexität steigt, wird durch die Regulierung eine Strafe für höhere Terme hinzugefügt. Dass hat zur Folge, dass die Wichtigkeit solcher langen Terme sinkt und zugleich die Komplexität des Models weniger wird.

Beispiele für solche Techniken sind folgende:

1. L1 Regulierung (Lasso Regulierung)
2. L2 Regulierung (Ridge Regulierung)
3. Elastic net

# 6 Fazit

Thema ist größer als hier beschreibbar
Evtl. Deep Fake http://iphome.hhi.de/samek/pdf/LapNCOMM19.pdf https://ujjwalkarn.me/2016/08/
explanation-convnets/

# Literatur


[1] Anup Bhande. *What is underfitting and overfitting in machine learning and how to deal with it.* 2018. URL: https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76 (besucht am 14. 06. 2020).

[2] S Borowiec. *AlphaGo seals 4–1 victory over go grandmaster Lee Sedol.* 2016. URL: https://www.theguardian.com/technology/2016/mar/15/googles-alphago-seals-4-1-victory-over-grandmaster-lee-sedol (besucht am 08. 06. 2020).

[3] João Gama u. a. „A Survey on Concept Drift Adaptation". In: *ACM Comput. Surv.* 46.4 (März 2014). ISSN: 0360-0300. DOI: 10.1145/2523813. URL: https://doi.org/10.1145/2523813.

[4] Irina Hübner. *Wie KI autonomes Fahren sicherer macht.* 2019. URL: https://www.elektroniknet.de/elektronik-automotive/assistenzsysteme/wie-ki-autonomes-fahren-sicherer-macht-171811.html (besucht am 11. 06. 2020).

[5] Dastin Jeffrey. *Amazon scraps secret AI recruiting tool that showed bias against women.* 10. Okt. 2018. URL: https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G (besucht am 24. 05. 2020).

[6] Czihlarz Jochen. *Biases in Künstlicher Intelligenz (KI).* 21. Apr. 2020. URL: https://www.anti-bias.eu/allgemein/biases-in-kuenstlicher-intelligenz/ (besucht am 24. 05. 2020).

[7] Pechenizkiy M Kamiran F Calders T. *Discrimination aware decision tree learning.* 2010, S. 869–874.

[8] Jan Knupper. *Trainingsdaten für KI – Loesungen: Nicht nur die Masse macht's.* 2019. URL: https://www.clickworker.de/2019/03/19/trainingsdaten-ki-loesungen/ (besucht am 14. 06. 2020).

[9] Jaromir Konecny. *Von Jaromir KonecnyLesedauer ca. 14 Minuten148 Kommentare Künstliche Intelligenz, künstliche Dummheit und der gesunde Menschenverstand.* 2018. URL: https://scilogs.spektrum.de/gehirn-und-ki/kuenstliche-intelligenz-kuenstliche-dummheit-und-der-gesunde-menschenverstand/ (besucht am 11. 06. 2020).

[10] Obermeyer Z. und Powers B. und Vogeli C. und Mullainathan S. *Dissecting racial bias in an algorithm used to manage the health of populations.* 2019, S. 447–453.

[11] D Sahoo u. a. *Online deep learning: learning deep neural networks on the fly*. 2018, S. 2660–2666.

[12] E. Sengupta u. a. „Techniques to Elimenate Human Bias in Machine Learning“. In: *2018 International Conference on System Modeling Advancement in Research Trends (SMART)*. 2018, S. 226–230.

[13] Christian Stöcker. *Wer nur zurückschaut, schafft keine gute Zukunft*. 2019. URL: https://www.spiegel.de/wissenschaft/mensch/digitale-plattformen-wer-nur-zurueckschaut-schafft-keine-gute-zukunft-a-1261554.html (besucht am 14. 06. 2020).

[14] TC Wang u. a. *Video-to-video synthesis*. 2018, S. 1144–1156.

[15] West S.M. und Whittaker M. und Crawford K. *Discriminating Systems: Gender, Race and Power in AI*. 2019, S. 3. URL: https://ainowinstitute.org/discriminatingsystems.html (besucht am 24. 05. 2020).

[16] Peter-Michael Ziegler. *Im Namen des Algorithmus*. 2017. URL: https://www.heise.de/select/ct/2017/25/1512700333136715 (besucht am 24. 05. 2020).