



Duale Hochschule Baden-Württemberg  
Mannheim

Seminararbeit

## Algorithmen des maschinellen Lernens

### Studiengang Elektrotechnik

Studienrichtung Automation

Verfasser:	Finn Rasmus Schäfer
Matrikelnummer:	9512059
Kurs:	TEL19AT2
Studiengangsleiter:	Walther Berthold

# Inhaltsverzeichnis

1	Einleitung	3
2	Grundlagen des Maschinelle Lernen	4
3	Lernansätze	6
4	Algorithmen	7

# Abbildungsverzeichnis

1	Gegenüberstellung der Häufigkeit der Suchbegriffe „machine learning“ (blau) und „artificial intelligence“ (rot). Quelle: Google . . . . .	3
2	Gegenüberstellung von linearer und polynomialer Regression erstellt mit maschinellern Lernen Bildquellen: lineare Regression: Eigendarstellung, polynom Regression W3 School . . . . .	4
3	Beispieldarstellung für die Objektklassifizierung mittels maschinellern Lernen. Bildquelle: Amitrajit Bose . . . . .	5
4	Darstellung des Datenclustering für einen Datensatz mit drei verschiedenen Klassen. Bildquelle: Geeks for Geeks . . . . .	5
5	Visualisierung der Dimensionsreduktion zum effektiven Anwenden von machine Learning. Bildquelle: Kirkiles . . . . .	6
6	Visuelle Darstellung für einen maschinellen Lerndurchlauf mittels Gradientenabstieg. Bildquelle: Shuhao Cao . . . . .	8
7	Darstellung von zu kleiner und zu großer Lernrate an einem zweidimensionalen visuellen Beispiel . . . . .	9

# 1 Einleitung

Das maschinelle Lernen ist ein Teilbereich der künstlichen Intelligenz und findet Anwendung in einer Vielzahl von Industrie und Forschungsbereichen. Die in der Regel bekannten Anwendungsfälle sind vor allem die Computer Vision und die Robotik, des Weiteren findet das maschinelle Lernen aber auch Anwendung im Bereich der Data Science, der Bioinformatik und vor allem im natural language processing. Wobei das natural language processing das automatische Vervollständigen in zum Beispiel Suchmaschinen darstellt. Gerade seit ca. 2015 erfreut sich das maschinelle Lernen an einem sogar noch größerem Interesse als die künstliche Intelligenz an sich. Vergleicht man die Häufigkeit der Suchbegriffe über Google Trends, so fällt auf, dass gerade ab diesem Zeitpunkt deutlich mehr Interesse an maschinellem Lernen als an künstlicher Intelligenz im allgemeinen zu bestehen scheint. Auch der Durchschnitt ist seit, von 2004 bis heute betrachtet, mittlerweile höher (siehe Abbildung 1). Um zu verstehen, warum vor allem das Interesse am maschinellen Lernen so gestiegen ist muss zu erst einmal betrachtet werden, was das maschinelle Lernen überhaupt ist. Das maschinelle Lernen stellt das Lernen von Daten dar. In der Regel existieren eine bestimmte Anzahl von Datensätzen, diese werden verwendet um Auffälligkeiten zu klassifizieren oder Regelmäßigkeiten im allgemeinen zu verstehen. Anhand dieser Trainingsdaten kann dann eine Vorhersage für unbekannte Daten getroffen werden. Grundlegend lässt sich die Thematik in verschiedene Grundbereiche unterteilen, die sinnbildlich als ein Problemfeld gesehen werden kann. Neben den bekanntesten Bereichen des „Supervised Learning“, dem „Reinforcement Learning“ und dem „Unsupervised Learning“ gibt es in verschiedenen Forschungsbereichen auch das „Active Learning“, das „Learning to learn“ und das „Semi-Supervised Learning“ [1].



Abbildung 1: Gegenüberstellung der Häufigkeit der Suchbegriffe „machine learning“ (blau) und „artificial intelligence“ (rot). Quelle: Google

Jeder Teilbereich des maschinellen Lernens kann für einzelne Problem- und Anwendungsbereiche verwendet werden. Hauptsächlich unterscheidet man zwischen Regression, Klassifikation, Clustering und zum Beispiel Dimensionalitätsreduktion zur Vereinfachung von Problemen. In den Folgenden Kapiteln wird auf die einzelnen Probleme und Anwendungsbereiche eingegangen, vor allem aber auch die Algorithmik dahinter erklärt.

## 2 Grundlagen des Maschinelle Lernen

Wie im vorherigen Abschnitt bereits eingeführt lässt sich das maschinelle Lernen in verschiedene Teilbereiche aufteilen. Jeder Teilbereich deckt dabei eine eigene Problemstellung ab. Die in der Regel bekanntesten Probleme sind dabei die Folgenden:

- Regression
- Klassifizierung
- Clustering
- Dimensionsreduzierung

Die Regression lässt sich in zwei Teilbereiche unterteilen, zum einen die lineare Regression und zum Anderem die Polynomregression. Die lineare Regression wird verwendet um eine lineare Funktion durch eine gewisse Anzahl von Datenpunkten zu legen. Die Polynomregression hingegen wird dazu verwendet ein Polynom durch die Datenpunkte zu legen. Die grundlegenden Algorithmen für Regressionsprobleme sind der Gradientenabstieg und die Normalengleichung, die im nächsten Kapitel thematisiert werden.

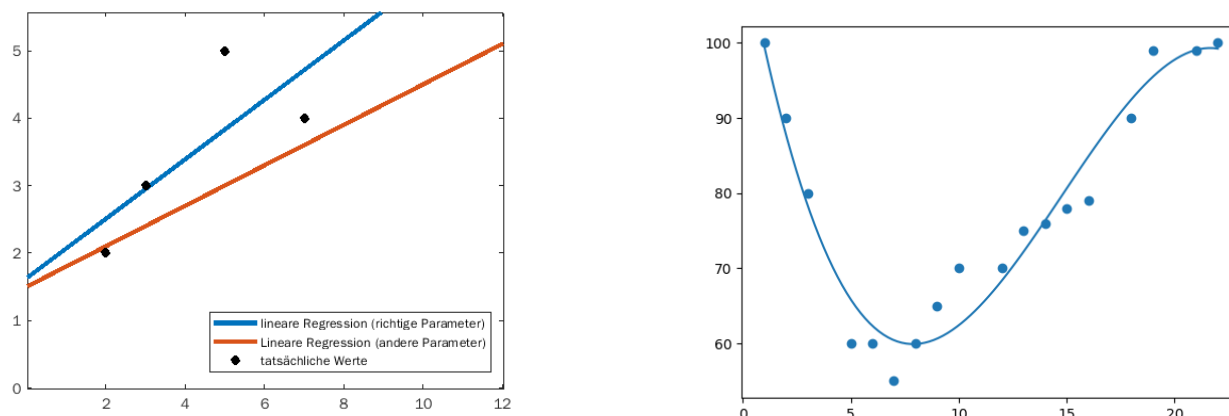


Abbildung 2: Gegenüberstellung von linearer und polynomialer Regression erstellt mit maschinellem Lernen Bildquellen: lineare Regression: Eigendarstellung, polynom Regression W3 School

Klassifizierungsprobleme werden vor allem in der Objektklassifizierung verwendet, darunter fallen zum Beispiel die Schrifterkennung, Tiererkennung oder zum Beispiel Transaktionsvalidität. Die Objektklassifizierung ist nicht mit der Objekterkennung zu verwechseln, die Objekterkennung fällt in einen detaillierten Unterbereich des maschinellen Lernens, dem Deep Learning [2]. Deep Learning ermöglicht das erkennen bestimmter Personen oder zum Beispiel Tiere. Die Klassifizierung ist nur in der Lage festzustellen ob es sich um einen Menschen oder ein Tier handelt.

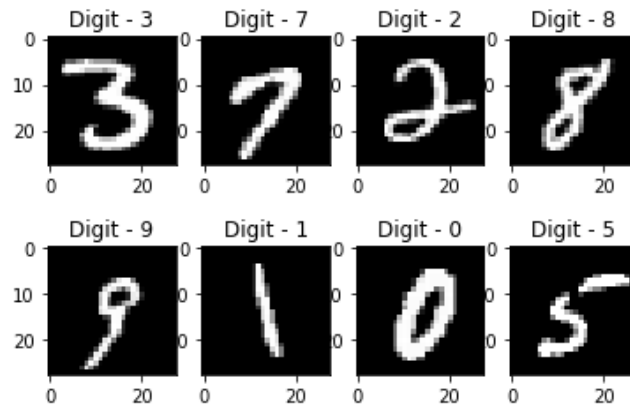


Abbildung 3: Beispieldarstellung für die Objektklassifizierung mittels maschinellem Lernen. Bildquelle: Amitrajit Bose

Hier sind vor allem die neuronalen Netze angesiedelt. Für die Klassifizierung werden vor allem bekannte Daten zum Trainieren verwendet. Es wird demnach versucht bekannte Objekte innerhalb der Daten zu erkennen. Unbekannte Zusammenhänge und zugehörigkeiten zu erkennen wird wiederum Clustering genannt. Eingangsgröße des Algorithmus sind dabei unstrukturierte Daten, aus denen Informationen gewonnen werden sollen. Ausgangsgröße des Algorithmus ist eine Gruppierung gewisser Datenpunkte in Klassen.

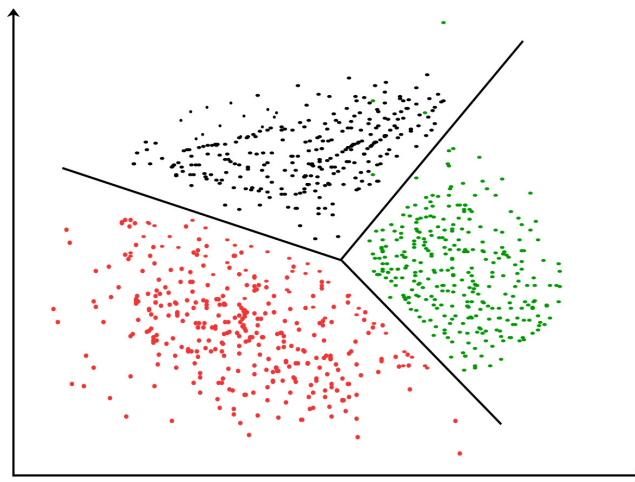


Abbildung 4: Darstellung des Datenclustering für einen Datensatz mit drei verschiedenen Klassen. Bildquelle: Geeks for Geeks

Die Dimensionsreduzierung wird vor allem auf große und komplexe Datensätze angewandt. Ziel dieser Reduktion ist die Reduktion der Komplexität des Problems, die in der Praxis oftmals Probleme bereitet. Über verschiedene Ansätze zum Beispiel Transformation und die Hauptkomponentenanalyse (PCA) wird versucht der Erhalt der Informationen bei niedrigerer Komplexität zu erhalten [3],[4]. Für diesen Ansatz erfolgt

die Erklärung ebenfalls ausführlicher im nächsten Abschnitt.

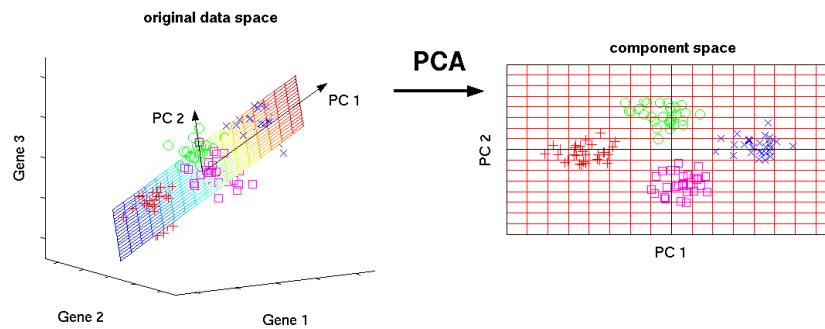


Abbildung 5: Visualisierung der Dimensionsreduktion zum effektiven Anwenden von machine Learning. Bildquelle: Kirkiles

### 3 Lernansätze

Wie im vorangegangenen Kapitel schon beschrieben gibt es verschiedene Lernansätze. Diese Lernansätze sind vor allem davon abhängig, welche Art von Datensatz man besitzt. Allgemein unterscheidet man zwischen strukturierten und unstrukturierten Daten. Bei den unstrukturierten Daten ist keine Vorkenntnis über die Daten vorhanden. Bei Strukturierten Daten liegen bereits wichtige Informationen vor. Strukturierte Daten werden zum Beispiel für Klassifizierungsprobleme verwendet. Innerhalb der strukturierten Daten werden verschiedene Features angegeben und schlussendlich die Klasse definiert. Anhand dieser Daten können dann Auffälligkeiten identifiziert werden und vorhersagen über unbekannte, unstrukturierte bzw. unklassierte Daten getroffen werden. Für das Clustering wird in der Regel unstrukturierte Big Data verwednet. Das bedeutet, dass hier verschiedene Daten vorhanden sind, jedoch noch keine geordneten Informationen über die Daten vorhanden sind. Diese werden erst nach dem Clustering geordnet sein. Die grundlegend verschiedenen Lernansätze sind dabei:

- Supervised Leraning
- Unsupervised Learning

Supervised Learning, deutsch „unterstütztes Lernen“, nutzt dabei für die Modellbildung strukturierte Daten. Zum Bilden des Modells werden die vorhandenen Daten gedrittelt und in der Regel  $\frac{2}{3}$  der Daten zur Modellgenerierung und  $\frac{1}{3}$  zur Modellvalidierung verwendet. Das Modell sollte erst nach der Validierung auf unbekannte Daten zur Prädiktion verwendet werden. Wie bereits angesprochen besitzt unsupervised learning keine Vorinformationen über die Daten. Die genauer Algorithmik wird hier jedoch nicht weiter betrachtet.

## 4 Algorithmen

Innerhalb dieses Kapitels werden beispielhaft einige Algorithmen des maschinellen Lernens vorgestellt, dabei handelt es sich vor allem um Algorithmen die auf der Theorie des supervised Learning beziehen. Für die Klassifizierung und die Regression werden vor allem der Gradientenabstieg und die Normalengleichung verwendet. Beim Gradientenabstieg handelt es sich um eine Potentialfeldmethode, mit der lokale Maxima oder Minima gefunden werden sollen. Der Gradientenabstieg findet vor allem Anwendung, wenn das Problem wenige Features hat, dafür aber eine große Menge an Testdaten. Die Normalengleichung verwendet man bei vielen Features und wenig Datensätzen. Bei der Normalengleichung handelt es sich um Matrixoperationen, die schlussendlich die gewünschten Parameter für zum Beispiel eine Funktion oder einen binären Entscheidungsbaum liefern.

Für den Gradientenabstieg (engl. Gradient Descent) wird über eine so genannte Lernrate und dann Schritt für Schritt ein Extrema im Gradientenfeld gesucht. Es ist dabei jedoch festzuhalten, dass das Gradientenverfahren nicht immer in einem Extrempunkt stagniert, oder aber auch in einem lokalen Extrempunkt stagnieren kann. Das Gradientenverfahren muss dabei nicht immer das ideale Ergebnis liefern. Fehler können dabei vor allem bei der Wahl der Anfangswerte oder aber in der Wahl der Lernrate gemacht werden. Um beispielsweise zwei Gewichtsparameter über eine Minimierung einer gewissen Gütefunktion (zum Beispiel die Standardabweichung) zu bestimmen, geht man beim Gradientenabstieg beispielsweise wie folgt vor:

1. Gütefunktion bestimmen
2. Bestimmung des Kosten über Gradientenalgorithmus

Der Gradientenalgorithmus berechnet sich dabei über das simultane Neuberechnen der Gütefunktionsparameter. Dazu wird die Multiplikation von Lernrate und der Ableitung der Gütefunktion vom aktuellen Wert des Parameters bestimmt. Das Ganze wird solange durchgeführt, bis die vorgegebenen Durchläufe abgearbeitet sind. Je nach Wahl der Lernrate wurde dann entweder ein Extrema gefunden, ein lokales Extrema gefunden oder aber kein sinnvolles Ergebnis gefunden. Abhängig ist das Ganze von der Wahl der Lernrate. Als Beispiel für den Gradientenabstieg wird Folgendes Beispiel betrachtet:

$$f(m, b) = \frac{1}{N} \sum_{i=0}^n (y_i - (mx_i + b))^2 \quad (1)$$

In der Gütefunktion, zu sehen in Gleichung 1, wird die Abweichung eines Punktes zum eigentlichen Datenpunkt berechnet. Die beiden Features sind dabei die Steigung  $m$  und der Anfangswert  $b$ . Über den Gradientenabstieg sollen nun simultan die idealen Werten für die Features bestimmt werden. Für die Berechnung der Parameter wird die Gütefunktion partiell nach dem anderen Feature abgeleitet und wie bereits erwähnt mit der Lernrate multipliziert. Für unser Beispiel sieht die Güteparameterberechnung wie folgt aus:

$$\begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} m - \alpha \frac{df}{dm} \\ b - \alpha \frac{df}{db} \end{bmatrix} = \begin{bmatrix} m - \alpha \left( \frac{-2x(y-mx+b)}{N} \right) \\ b - \alpha \left( \frac{-2(y-mx+b)}{N} \right) \end{bmatrix} \quad (2)$$

Das simultane Updaten der Gütefunktionsparameter wird  $N$ -mal durchlaufen. Der visuelle Gradientenabstieg sieht dabei beispielsweise wie folgt aus:

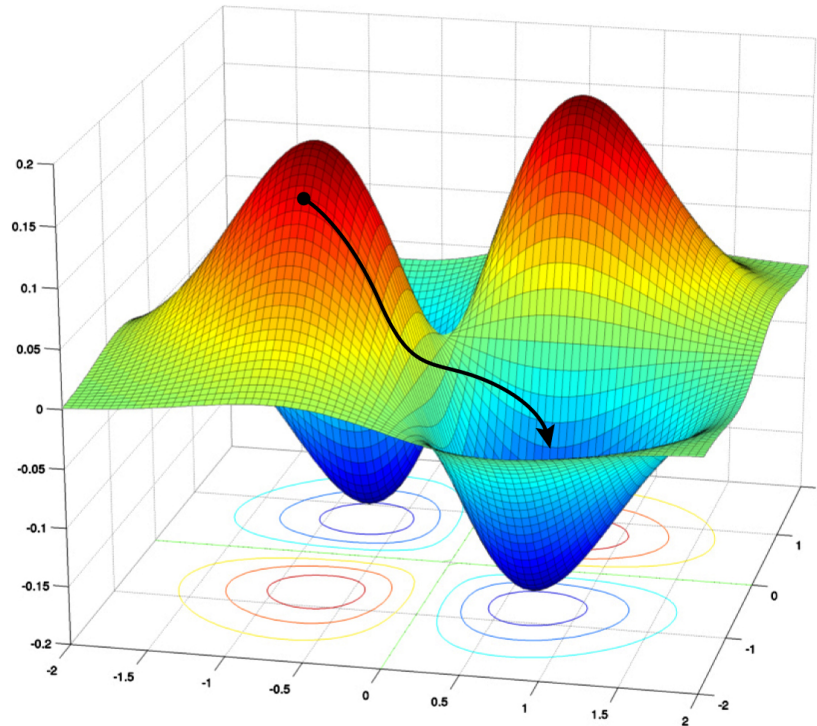


Abbildung 6: Visuelle Darstellung für einen maschinellen Lerndurchlauf mittels Gradientenabstieg. Bildquelle: Shuhao Cao

Die Nachteile des Gradientenabstiegs sind vor allem, schlecht gewählte Startpunkte können den Prozess des Gradientenverfahren enorm verlängern. Außerdem kann sich durch eine schlecht gewählte Lernrate ein nicht konvergierendes Verhalten einstellen. Beispielsweise ist dies in Abbildung 7 dargestellt.

Der Gradientenabstieg eignet sich vor allem für eine große Anzahl von Features (ca.  $10^6$ ). Als Alternative zum Gradientenabstieg dient oftmals die Normalengleichung. Diese ist in der Regel gut geeignet, wenn die gegebenen Features nicht so hoch sind (ca.  $\leq 1000$ ). Gerade weil die Normalengleichung eine Komplexität von  $O(n^3)$  besitzt, ist diese nicht für eine größere Anzahl von Features geeignet. Sie bietet jedoch den Vorteil, nicht von einem gewissen Startwert und auch nicht von einer Lernrate abhängig zu sein. Die gefundene Lösung ist daher immer die beste Wahl und kann nicht durch das Stagnieren in lokalen Extremas zum Beispiel durch zu große oder kleine Wahl der Lernrate beeinflusst



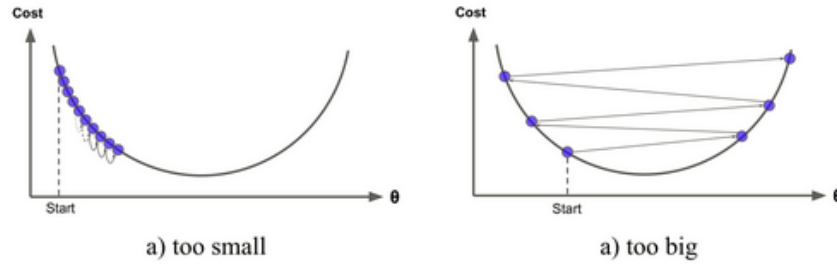


Abbildung 7: Darstellung von zu kleiner und zu großer Lernrate an einem zweidimensionalen visuellen Beispiel

werden. Bei der Normalengleichung handelt es sich um eine Gleichung in Matrixform. Mittels des Gleichungssystems lässt sich im Fall der linearen Regression zum Beispiel die Steigung und der Startwert der Funktion bestimmen. Die Gleichung bildet sich aus den vorhandenen Datensätzen. In den Datensätzen sind die Features und der zugehörige Endwert gegeben. Als Beispiel könnte man dabei eine Autovermietung nehmen. Mögliche Features bei einer Autovermietung wären beispielsweise:

- Hubraum
- Leistung
- Verbrauch
- Kauton

der Zugehörige Endwert wäre der Preis des Autos pro Tag. Über die aus der Normalengleichung ermittelten Werte der linearen Funktion könnte ermittelt werden, ob ein Auto im Vergleich sehr teuer oder eher billiger wäre bzw. welche Zusammenhänge dort zu erkennen wären. Die Grundlagen der Berechnung würden dabei wie folgt aussehen:

$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$y$
1	2100	10	1	55	420
1	2200	8	3	45	666
1	2000	9	6	35	404
1	1800	12	4	55	187

Table 1: Beispieldatensatz für die Normalengleichung

Für die Berechnung der Funktionsgrößen muss nun zu Beginn die Matrix  $X$  und der Vektor  $y$  erstellt werden. Die Matrix ist dabei eine  $m \times (n + 1)$  Matrix, wobei  $m$  die Anzahl der Testdatensätze ist und  $n$  die Anzahl der Features und der Vektor  $y$  logischerweise  $m$ -dimensional ist. Für dieses Beispiel ergibt sich demnach folgende Matrix  $X$  und

ein Vektor  $y$ :

$$X = \begin{bmatrix} 1 & 2100 & 10 & 1 & 55 \\ 1 & 2200 & 8 & 3 & 45 \\ 1 & 2000 & 9 & 6 & 35 \\ 1 & 1800 & 12 & 4 & 55 \end{bmatrix} \quad y = \begin{bmatrix} 420 \\ 666 \\ 404 \\ 187 \end{bmatrix}$$

Um nun die Parameter für die lineare Funktion zu berechnen muss Folgender Rechenalgorithmus angewandt werden, der Lösungsvektor  $\theta$  enthält die gewünschten Gewichtungen:

$$\theta = (X^T X)^{-1} X^T y \quad (3)$$

Eine einfache Berechnung mittels Matlab liefert für diesen Anwendungsfall einen Lösungsvektor von:

$$\theta = \begin{bmatrix} -3.2781 \\ 0.3406 \\ -155.2326 \\ 59.8837 \\ 21.8295 \end{bmatrix}$$

Über die über die Normalengleichung gegebene Gewichtung lässt sich somit eine Lösungsvorhersage treffen. Bevor jedoch eine Vorhersage getroffen werden soll, validieren wir unser Modell mit dem Datensatz aus Zeile 2 der Matrix. Die Rechnung liefert folgendes Ergebnis:

$$y(3) = 1 \cdot -3.2781 + 2000 \cdot 0.3406 - 9 \cdot 155.2326 + 6 \cdot 59.8837 + 55 \cdot 21.8295$$

$$404 \approx 404.1703$$

Das geschaffene Modell scheint Valide zu sein und kann in der Zukunft dazu genutzt werden Andere Ergebnisse vorherzusagen.

Bei der Hauptkomponenten Analyse kurz PCA findet in der Regel eine Koordinatentransformation statt. Die Koordinatentransformation wird dazu genutzt den Datensatz von einem Datenraum in einen anderen zu transferieren und ihn damit zugänglicher zum Beispiel für Gradientenverfahren oder Normalengleichung zu machen. Herausforderung bei dieser Thematik ist in der Regel das Finden der Transformationsmatrix. Da die Thematik den Rahmen dieser Arbeit sprengen würde soll das PCA-Verfahren hier nur symbolisch erwähnt und dargestellt werden.

## Literaturverzeichnis

- [1] Prof. Dr. Stephan Günnemann, *Machine Learning - Introduction*, Technical University Munich, Lecture, 2021.
- [2] L. Wuttke. „Machine learning vs. deep learning: Wo ist der unterschied?“ (o.D.), [Online]. Available: <https://datasolut.com/machine-learning-vs-deep-learning/> (visited on 06/01/2022).
- [3] Prof. Dr. Stephan Günnemann, *Machine Learning - Dimensionality Reduction & Matrix Factorization*, Technical University Munich, Lecture, 2021.
- [4] Philipp Helo Rehs, *Verfahren zur Dimensionsreduktion*, Heinrich Heine Universität Düsseldorf, Bachelorarbeit, 2014.