Fakultät Ingenieurwissenschaften
Labor für Kooperative, automatisierte Verkehrssysteme (KAV)

Verteidigungsmaßnahmen gegen Modell-Inversionsangriffe

Bachelorarbeit

von

Hannes Weber

Aschaffenburg, 28. November 2023

Autor:

Hannes Weber
Am Lindenbrunnen 17
D-97846 Partenstein

Matrikel-Nr.: 2220472
Studiengang Software Design (Bachelor)

Prüfer:

Prof. Dr.-Ing. Konrad Doll
Kooperative, automatisierte Verkehrssysteme (KAV)

Zweitprüfer:

Prof. Dr.-Ing. Ulrich Brunsmann

Technische Hochschule Aschaffenburg
Fakultät Ingenieurwissenschaften
Würzburger Straße 45
D-63743 Aschaffenburg

Ehrenwörtliche Erklärung

Hannes Weber

Am Lindenbrunnen 17
D-97846 Partenstein

Hiermit erkläre ich, dass ich die von mir vorgelegte Arbeit mit dem Thema „*Verteidigungsmaßnahmen gegen Modell-Inversionsangriffe*“ selbstständig verfasst habe, dass ich die verwendeten Quellen, Internet-Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen und Bildern –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Aschaffenburg, den 28. November 2023

Hannes Weber

Danksagung

Hiermit möchte ich mich ausdrücklich bei Herrn Prof. Dr. Konrad Doll und Herrn Prof. Dr. Ulrich Brunsmann für die Unterstützung während meiner Arbeit bedanken. Des Weiteren ...

Inhaltsverzeichnis

Abkürzungen

ML	*Maschinelles Lernen*
CNN	*Convolutional Neural Network*
KNN	*Künstlich-Neuronales Netzwerk*
NN	*Neuronales Netzwerk, eng.: Neural Network*

Kapitel 1

1 Einleitung

1.1 Motivation

Inmitten der fortscheitenden Digitalisierung greift man immer häufiger auf Lösungen zurück, die diverse Aufgaben in Arbeits- und Lebensbereich unterstützen, vereinfachen und sogar ersetzen. Von Medizin, in der beispielsweise Diagnoseverfahren zur frühzeitigen Erkennung von Krankheiten eingesetzt werden, bis zur Automobilindustrie, die schon jetzt auf selbstfahrende Kraftfahrzeuge setzt. Durch all die neuen Entwicklungen und Einsatzgebiete von künstlichen Intelligenzen gewinnt diese unaufhaltsam an Bedeutung.

Trotz des enormen Potenzials, das diese Technologien in ihren jeweiligen Anwendungsgebieten mit sich bringen, treten vermehrt Herausforderungen hinsichtlich Sicherheit und Robustheit der Systeme auf. Ein zentraler Aspekt ist die Notwendigkeit der Risikominimierung durch das Sicherstellen der Vertrauenswürdigkeit von KI-Systemen. Ein weiterer wichtiger Punkt in der Entwicklung dieser Systeme ist die Robustheit gegenüber Angriffen. KI-Systeme können anfällig für Manipulationen und gezielte Angriffe von außen oder durch Fehlkonfigurationen sein. Dabei spielt die Implementierung und Integration von bestimmten Sicherheitsmechanismen eine entscheidende Rolle, um die Integrität der Systeme zu gewährleisten. Insgesamt is die Sicherheit im Bereich des maschinellen Lernens von zentraler Bedeutung, wordurch das Vertrauen der Nutzer gestärkt und die breite Integration dieser Technologien in verschiedener Bereichen vorangetrieben wird. Diese Arbeit legt den Fokus auf die Herausforderung der Verdeutlichung von Sicherheits- und Robustheitsimplementierungen in Bild-Klassifikationsmodellen. In einem globalen System, in dem immer mehr Aspekte des täglichen Lebens in die 'Hände' von KI-Systemen gegeben werden, ist es umso wichtiger, diese mit Anbetracht auf Sicherheit zu implementieren, wie auch zu überwachen. Eine Unsicherheit eines Systems kann hierbei schon zu Verletzungen der Privatsphäre einzelner Personen führen. Daher wird diese Arbeit gesondert Angriffsvektoren und Bedrohungen von Klassifikationsmodellen behandeln. Darüber hinaus werden innovative Ansätze zur Bekämpfung möglicher Angriffsvektoren und Schwachstellen aufgezeigt, die sowohl Sicherheit, als auch Robustheit von neuronalen Netzen erhöhen. Insgesamt soll diese Arbeit dazu beitragen, dem Leser ein grundlegendes Verständis über Angriffe, deren Auswirkungen, wie auch Verteidigungsmaßnahmen zu liefern.

1.2 Aufgabenstellung

Hintergrund

Da KI-Systeme immer mehr Anwendung finden, werden diese zudem häufiger Ziele von Cyber-Angriffen. Es gibt viele verschiedene Angriffsvektoren auf diese Systeme, zu denen auch das Erlangen von zugrundeliegenden Informationen über Daten des Trainingsprozesses von Neuronalen Netwerken gehört. Da viele Modelle mit sensiblen Daten trainiert werden, ist es wichtig, dass Daten durch das Modell nicht an Angreifer übergeben werden. Der 'EU-AI Act' enthält unter anderem Anforderungen an KI-Systeme in Bezug auf Robustheit und Cybersicherheit. Daher bringt dieser nicht nur die Verantwortung sichere Modelle zu trainieren mit sich, sondern auch die Sicherstellung, dass genutzte Trainingsdaten privat gehalten werden. Um Modelle während der Bereitstellung abzusichern, müssen die verschiedenen Angriffsvektoren und die entsprechenden Abwehrmaßnahmen bekannt sein.

Ziel der Arbeit

Während der Thesis sollen folgende Fragen beantwortet werden:

- Welche Angriffsmöglichkeiten gibt es, um Daten von deployten Modellen zu extrahieren?
- Wie kann man sich gegen diese Attacken schützen?
- Showcase zu Verteidigungstechniken und deren Effektivität gegenüber Inversions-Angriffen.

Methodischer Ansatz

- Onboarding
- Recherche über verschiedene Attacken und Verteidigungsmöglichkeiten
- Vergleich von verschiedenen Angriffen auf unterschiedliche Modell-Architekturen
- Implementierung eines Showcases für mindestens einen Angriff auf mindestens eine Modell-Architektur:
  - Zeigen, wie ein solcher Angriff funktioniert.
  - Kann man sich gegen einen solchen Angriff verteidigen?
  - Wie effektiv sind die Verteidigungsstrategien?

1.3 Aufbau der Arbeit

Die Arbeit unterteilt sich in drei Hauptbestandteile. Dazu gehört zum einen der SStand der Technik", worin aktuelle Technologien und Modelle dargestellt werden. Hauptsächlich wurde der Wissenstand aus anderen Papern entnommen und spiegelt somit den "ResearchingTeil der Arbeit wieder. ....

Eine weitere Hauptkomponente der Arbeit ist die Implementierung eines Modells und dessen Angriffsvektoren. Mit der Implementierung der möglichen Verteidigungsmaßnahmen der Schwachstellen wird die Kehrseite aufgezeigt, mit der eine mögliche Absicherung stattfinden kann. ...

Durch die Darstellung und Auswertung der herausgefundenen Ergebnisse, geht die Arbeit zu Ende. Dabei werden die wichtigsten Erkenntnisse aus den Implementierungen und Researching-Tasks zusammengefasst, und anschaulich dargestellt. Dies soll den Vorteil mit sich bringen, dem Leser einen Überblick des neu erweorbenen und eine finale Aussicht zu bieten. ...

Der Aufbau der Arbeit lehnt sich an die Struktur von **Bar-Shalom** an.

Kapitel 2

2 Grundlagen

2.1 Maschinelles Lernen

Ein Teilgebiet der künstlichen Intelligenzen ist der Bereich des Maschinellen Lernens, welcher die Verarbeitung von Daten addressiert, um beispielsweise Vorhersagen in Situationen auf Basis von bestimmten Informationen zu treffen. Im Jahr 1959 wurde dieser von Arthur Samuel geprägt, der die Herausforderung annahm, ein Schachspiel mit Hilfe einer Maschine zu lösen. (Joshi 2020, S. 4) Das besondere hierbei ist das Erlernen des richtigen Lösungswegs, welcher nicht durch bestimmte Bedingungen gesteuert, sondern auf Basis verschiedener Informationen erlernt wird, die über diverse Quellen beigefügt werden. Maschinelles Lernen findet in vielen Bereichen Anwendung, von Spracherkennung, Bilderkennung bis hin zu selbstfahrenden Kraftfahrzeugen. Diese Technologie wird zunehmend wichtiger in unserer immer vernetzteren Welt und trägt dazu bei, komplexer werdende Probleme zu lösen und immer innovativere Lösungen in verschiedenen Branchen einzuführen.

Im Wesentlichen geht es um die Entwicklung von Algorithmen, die dem Computer das Erkennen von Mustern und Zusammenhängen in Daten ermöglichen, wordurch ohne menschliche Intervention Aufgaben erledigt werden können. Daher trainiert man Algorithmen auf Basis großer Datenbestände, um die erlernten Fähigkeiten auf neu erhobene Daten anzuwenden. Das Erlernen des nötigen Wissens basiert hierbei auf drei wesentlichen Faktoren, die die Qualität des Trainings beeinflussen. Neben den Informationen, die aus Daten gewonnen werden, nutzt man eine Kennzahl, die einen Vergleich zwischen dem aktuellen und idealen Verhalten herstellt, um mit dem dritten Faktor - einem Rückkopplungsmechanismus - das programm anzuleiten eine verbesserte Leistung in Folgeergebnissen zu erzielen. (Joshi 2020, S. 4)

Um ein Modell trainieren zu können, wird ein geeignetes künstliches Lernverfahren ausgewählt, das darauf abzielt, die Ausgaben eines künstlichen Systems in Bezug auf bestimmte Systemeingaben im Laufe des Lernprozesses zu optimieren. Unterschieden wird hautpsächlich in der Art und Weise, wie die 'Kritik' präsentiert wird, die zur Verbesserung des Verhaltens der jeweiligen künstlichen Systeme führen soll. Ein Training basiert auf sogenannten Hyperparametern, die Rahmenbedingungen des Prozesses festlegen. Dies kann zum Beispiel die Anzahl der Epochen, die Lern-Rate oder auch die Batch-Größe sein. Im Folgenden werden drei prägende Lernverfahren näher beschrieben.

2.1.1 Überwachtes Lernen

Diese Art des Lernens (eng.: supervised learning) stellt einen wesentlichen Bereich des maschinellen Lernenes dar, um aus Informationen, bestehend aus Datenpunkten $X = x_1, x_2, \ldots, x_n$ mit einem zugehörigen Label aus $Y = y_1, y_2, \ldots, x_n$, das nötige Wissen für bestimmte Anwendungsfälle zu erlangen. Anwendung findet dieses Verfahren in Systemen, die nicht-gelabelte Daten verarbeiten, um darauf basierend eine Vorhersage über die Zugehörigkeit abzugeben. Mögliche Arten von neuronalen Netzwerken, die über diese Methode trainiert wurden, sind beispielsweise Bildklassifikatoren. Dabei ist es die Aufgabe des Modells M, basierend auf einen bestimmten Input I mit Hilfe erlernter Muster und Zusammenhänge der Trainingsdaten eine Vorhersage über die Zugehörigkeit von I zu treffen.

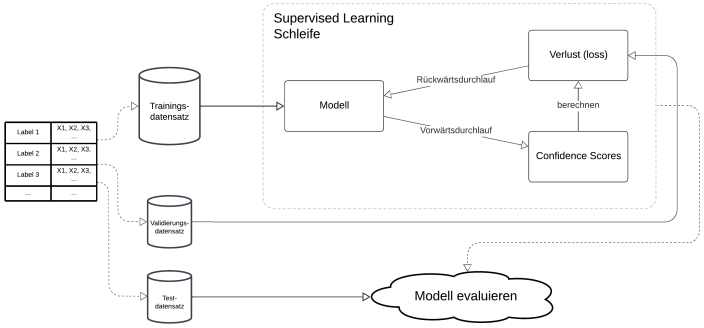

Bild 2.1: Prozessvisualisierung: überwachtes Lernen

Der *Ablauf* des Trainings lässt sich wie folgt beschreiben: nach Beenden von Datensammlung, Datenvorverarbeitung und Datensatztrennung in einen Trainings-, Validierungs- und Testdatensatz beginnt der eigentliche Teil des Trainings, wobei die Trainings- und Validierungsdaten für die aktualisierung der Modellparameter verwendet werden. Die Optimierung der Parameter wird auf Basis der Verlustfunktion mit Hilfe eines Rückwärtsdurchlaufes (eng.: Backward Propagation) durchgeführt, die mit der Differenz zwischen Vorhersagen des Modells aus dem Vorwärtsdurchlauf (eng.: Forward Propagation) und den realen Labeln berechnet wird.

Die *Qualität* des Trainings wird meist anhand der Genauigkeit von Vorhersagen bestimmt, welche ausdrückt, wie gut die Klassifikation von im Training noch nicht gesehenen Daten funktioniert.

$$\text{Accuracy} = \frac{\text{Anzahl korrekter Vorhersagen}}{\text{Gesamtanzahl der Vorhersagen}}$$

Weitere Metriken können die Präzision (eng.: Precision), der Recall oder auch der F1-Score sein. Neben Messungen der Modell-Genauigkeit kann man die Qualität des trainierten Modells auch anhand der Verlustfunktion (eng.: loss-function) oder der 'Area Under the Curve' (AUC), welche die Fläche unter der 'Receiver Operating Characteristic' (ROC) – dem Verhältnis zwischen der True- und False-Positive-Rate – messen.

2.1.2 Unüberwachtes Lernen

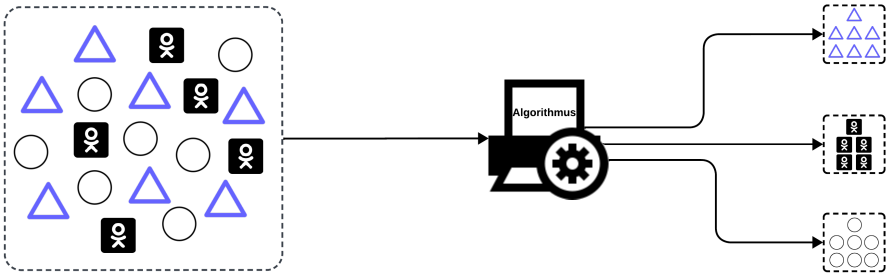

Bild 2.2: Resultat eines auf unüberwachtem Lernen basierenden Clusterings

Unüberwachtes Lernen (eng.: unsupervised learning) stellt einen anderen bedeutenden Bereich im maschinellen Lernen dar, der sich von überwachtem Lernen (2.1.1) dahingegen unterscheidet, dass keine expliziten Labels für die Trainingsdaten bereitgestellt werden. Diese Methode wird angewendet, wenn das Ziel darin besteht, Muster, Strukturen oder Zusammenhänge in den Daten zu entdecken, ohne dabei Kategorien oder Label vorzugeben, weshalb basierend auf Informationen eines Datensatzes, bestehend aus Datenpunkten $X = x_1, x_2, \ldots, x_n$ ohne dazugehöriges Label, trainiert wird. Das Modell soll dabei auf natürliche Art und Weise Strukturen, Muster und Zusammenhänge innerhalb der Eingabedaten ohne vorherige Kenntnisse der Zielvariable erlernen. „Aufgabe ist es hier, passende Repräsentationen zu finden, die z. B. die Erkennung von Charakteristika in Datenmengen, Wiedererkennung von Ausnahmen oder die Erstellung von Prognosen ermöglichen."(Lorenz 2020, S. 5) Im Gegensatz zu Überwachtem Lernen (2.1.1) geht es nicht um das Zurdnen von Mustern in vorhandenen Kategorien, sondern um das Auffinden von Clustern in einer bestimmten Datenmenge X. Dabei werden Modellparameter nicht über eine Verlustfunktion, die durch die Differenz zwischen tatsächlichem und vorhergesagtem Label berechnet wird, dargestellt, sondern die Parameter werden für die repräsentation inhärenter Muster in den Daten angepasst. Das Aufteilen des Datensatzes in 3 unabhängige Teile ist hierbei nicht notwendig, da keine Kategorien/Label für die jeweiligen Datenpunkte vorhanden sind. Es ist nur für die Auswertung des Algorithmus sinnvoll einen zweiten Datensatz zu erstellen, um damit die Qualität zu bewerten.

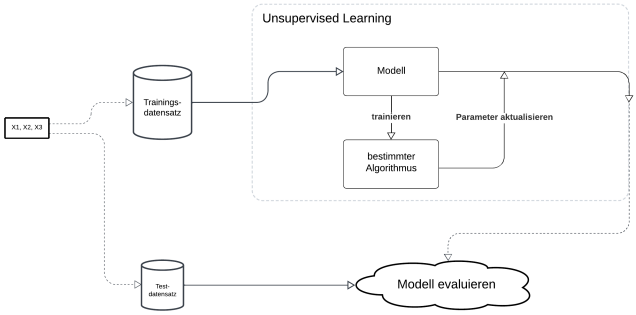

Bild 2.3: Prozessvisualisierung: unüberwachtes Lernen

Der *Ablauf* des Trainigs gestaltet sich wie folgt: Nach Abschluss von Datensammlung und Vorverarbeitung beginnt der eigentliche Prozess des unüberwachten Lernens. Im Gegensatz zum überwachten Lernen (2.1.1) gibt es hierbei keine vordefinierten Zielvariablen. Den Datenpunkten $X = x_1, x_2, \ldots, x_n$ ist also initial keine Kategorie beziehungsweise kein Label zugeordnet, da die Menge Y bist dato nicht existent ist. Im Gegensatz zu überwachtem Lernen nutzt man hier Algorithmen wie beispielsweise 'k-Nearest Neighbors (KNN)' (Joshi 2020, S. 38), um eine bestimmte Operation auf den zu behandelnden Datensatz auszuführen. Mit Hilfe des Algorithmus wird ein bestimmtes Modell dahingehend trainiert, dass es einen bestimmten Datensatz zum Beispiel in bestimmte Kategorien unterteilen kann. Dafür analysiert der Algorithmus die jeweiligen Datenpunkte und versucht diese mit Hilfe von Zusammenhängen und verschiedenen extrahierten Merkmalen zu interpretieren. Anhand des dabei erlangten Wissens werden Modellparameter aktualisiert, wonach das unüberwachte Lernen beendet ist. Ein Ergebnis kann hierbei beispielsweise ein kategorisierter Datensatz sein.

Die *Qualität* eines Modells ist schwieriger zu messen als bei überwachtem Lernen 2.1.1, da keine klar definierte Zielvariable vorliegt, mit Hilfe welcher eine Genauigkeitsanalyse durchgeführt werden kann. Dennoch lässt sich auch dieses mit verschiedenen Metriken evaluieren. Zum einen kann man mit dem sogenannten Silhouette Score (Shahapure und Nicholas 2020) Clustering-Algorithmen evaluieren, da dieser das Zusammenpassen der verschiedenen Datenpunkte innerhalb eines Clusters bewertet. Je höher dieser Wert, desto besser sind die jeweiligen Cluster definiert. Neben Metriken, die durch bestimmte Berechnungen repräsentiert werden, lässt sich die Qualität eines Modells auch durch visuelle Methoden wie der Auswahl geeigneter Plots messen, die die Zugehörigkeit verschiedener Datenpunkte darstellen.

2.1.3 Bestärkendes Lernen

Das bestärkende Lernen (eng.: reinforcement learning) ist ein maschinelles Lernverfahren, das auf Feedback der Umgebung angewiesen ist. Zudem besitzen Datenpunkte kein Label, weshalb dieser weder voll der Kategorie von überwachten noch der unüberwachter Lernverfahren zugeordnet werden. Die Interaktion mit der Umgebung findet hier mit Hilfe eines Agenten statt, welcher verschiedene Erfahrungen sammelt. Das Hauptziel des Algorithmus liegt darin, den Agenten so zu trainieren, dass er in seiner bestimmten Umgebung optimale Aktionen ausführt, die eine maximale Belohnung herbeiführen. Diese Aktionen werden innerhalb einer bestimmten Umgebung ausgeführt, welche – basierend auf die Tat – ein bestimmtes Feedback in Form von Belohung oder Strafe zurück gibt. Dadurch hat der Agent über die Dauer des Lernprozesses das Ziel, die maximale kumulative Belohung zu erzielen.

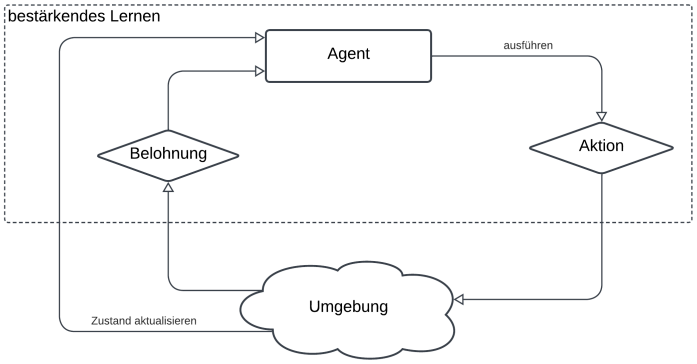

Bild 2.4: Prozessvisualisierung: bestärkendes Lernen

2.2 Bilderkennung /-klassifikation

2.3 Angriffsmöglichkeiten auf Neuronale Netzwerke

Hier soll geschildert werden, welche Angriffsmöglichkeiten es gibt. Im ersten Unterpunkt soll auf Model-Inversionsangriffe eingegangen werden.

2.3.1 Modell-Inversionsangriffe

Hier soll was über Modell-Inversionsangriffe stehen (Bsp. usw.) ...

2.3.1.1 Angriffsziel

Hier soll was über Ziel von diesem Angriff geschreiben werden (Bsp. usw.) ...

2.3.1.2 Angriffsvektoren

Hier soll geschrieben werden, wie der Angriff ausgeführt werden kann (Bsp. usw.) ...

2.3.1.3 Verteidigungsstrategien

Hier sollen mögliche Verteidigungsstrategien beleuchtet werden ...

Kapitel 3

3 Stand der Technik

3.1 Forschungsergebnisse

Kapitel 4

4 Implementierung

4.1 Funktionalität des Codes

Hier soll die Funktionalität des Codes beschrieben werden ...

```
import numpy as np

def incmatrix(genl1,genl2):
  m = len(genl1)
  n = len(genl2)
  M = None #to become the incidence matrix
  VT = np.zeros((n*m,1), int)  #dummy variable

#compute the bitwise xor matrix
  M1 = bitxormatrix(genl1)
  M2 = np.triu(bitxormatrix(genl2),1)

  for i in range(m-1):
    for j in range(i+1, m):
      [r,c] = np.where(M2 == M1[i,j])
        for k in range(len(r)):
          VT[(i)*n + r[k]] = 1;
          VT[(i)*n + c[k]] = 1;
          VT[(j)*n + r[k]] = 1;
          VT[(j)*n + c[k]] = 1;

          if M is None:
            M = np.copy(VT)
          else:
            M = np.concatenate((M, VT), 1)

          VT = np.zeros((n*m,1), int)

return M
```

Listing 4.1: Python example

4.2 Code-Beschreibung

Hier soll beschrieben werden, wie der Code strukturiert ist, nach welchen Pattern gearbeitet wurde und was der Code oberflächlich macht...

5 Kapitel 5

Ergebnisse

5.1 Beobachtungen

Hier sollen die Beobachtungen der Verteidigungsstrategien stehen ...

Bild einfügen: Das ist ein TEstBild Hier steht ganz viel Text... Hier

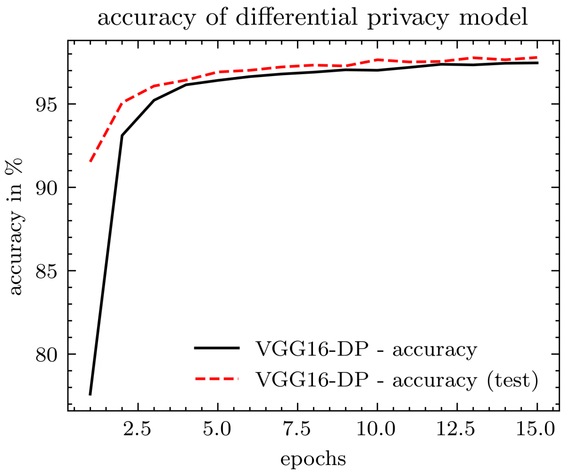

Bild 5.1: Beschreibung der Grafik

steht ganz viel Text... Hier steht ganz viel Text... Hier steht ganz viel Text... Hier steht ganz viel Text... Hier steht ganz viel Text... Hier steht ganz viel Text... Hier steht ganz viel Text... Hier steht ganz viel Text... Hier steht ganz viel Text... Hier steht ganz viel Text...

5.2 Rückschlüsse

Hier sollen die Rückschlüsse stehen ...

Kapitel 6

6 Zusammenfassung und Ausblick

Kapitel 7

7 Anhang

Hier sind noch LaTeX-Beispiele für die Verwendung:

...

Tabelle erzeugen:

Tabelle 7.1: Beispieltabelle

Spalte 1	Spalte 2	Spalte 3	Spalte 4
Inhalt 1	Inhalt 2	Inhalt 3	Inhalt 4
Inhalt 5	Inhalt 6	Inhalt 7	Inhalt 8

...

Quellen verlinken: Test **Bar-Shalom**. Test **Goldhammer**. Test **WHO-2004**.

...

Kapitel verlinken: Wie in Abschnitt 5 beschrieben wird ...

...

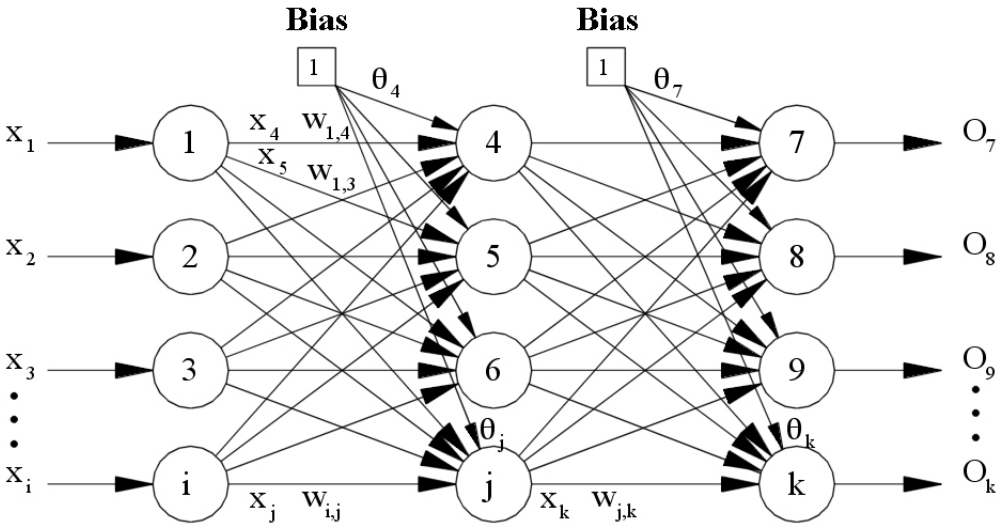

Bild 7.1: Beispielbild

Bilderverzeichnis

Tabellen

(Lorenz 2020)

Literatur

Bücher

Joshi, Ameet V (2020).
Machine Learning and Artificial Intelligence.
Cham: Springer International Publishing.
ISBN: 978-3-030-26621-9 978-3-030-26622-6.
DOI: 10.1007/978-3-030-26622-6.
URL: http://link.springer.com/10.1007/978-3-030-26622-6 (besucht am 24. 11. 2023).
Lorenz, Uwe (2020).
Reinforcement Learning: Aktuelle Ansätze verstehen - mit Beispielen in Java und Greenfoot.
Berlin, Heidelberg: Springer Berlin Heidelberg.
ISBN: 978-3-662-61650-5 978-3-662-61651-2.
DOI: 10.1007/978-3-662-61651-2.
URL: http://link.springer.com/10.1007/978-3-662-61651-2 (besucht am 04. 10. 2023).

Konferenzbeiträge

Shahapure, Ketan Rajshekhar und Charles Nicholas (Okt. 2020).
„Cluster Quality Analysis Using Silhouette Score“.
In: *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*.
2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA).
sydney, Australia: IEEE,
S. 747–748.
ISBN: 978-1-72818-206-3.
DOI: 10.1109/DSAA49011.2020.00096.
URL: https://ieeexplore.ieee.org/document/9260048/ (besucht am 28. 11. 2023).