

Master-Teamprojekt

Titel der Arbeit

Titel of the Thesis

Max Mustermann

Mat.Nr.: 420815

Betreuer:

Prof. Dr. rer. nat. E.-G. Haffner

Datum:

14. März 2022

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig verfasst habe und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht. Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen. Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Ort, Datum

Unterschrift

Danksagung

Ich bedanke mich bei all denjenigen, die mich während der Anfertigung dieser Arbeit unterstützt und motiviert haben. Ein ganz besonderer Dank gilt meinem Betreuer Prof. Dr. rer. nat. E.-G. Haffner. Ebenfalls möchte ich der Projektgruppe eTRainer der Hochschule Trier, sowie meinen Korrekturlesenden danken.

Abstract

This is a summary of all the important points and achievements of this work.

Zusammenfassung

Hierbei handelt es sich um eine Zusammenfassung aller wichtige Punkte und Errungenschaften dieser Arbeit.

Abkürzungsverzeichnis

ARS	Audience Response System
CGI	Common Gateway Interface
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
ML	Machine Learning

Inhaltsverzeichnis

1. Einleitung	1
2. Grundlagen	3
3. Hauptteil - 1	5
4. Hauptteil - 2	7
5. Hauptteil - 3	9
6. Evaluation	11
7. Fazit und Ausblick	13
7.1. Fazit	13
7.2. Ausblick	13
A. Anhang - Code	17
B. Anhang - Dokumentationen	19
Verzeichnisse	21
Literaturverzeichnis	21
Abbildungsverzeichnis	23
Tabellenverzeichnis	25
Code-Auszugs-Verzeichnis	27
Glossar	29

1

Einleitung

Hier wird in die Arbeit eingeleitet.

2

Grundlagen

Hier werden Grundlagen erläutert.

3

Hauptteil - 1

Hier wird der Inhalt der Arbeit präsentiert.

4

Hauptteil - 2

Hier wird der Inhalt der Arbeit präsentiert.

5

Hauptteil - 3

Hier wird der Inhalt der Arbeit präsentiert.



Evaluation

Hier wird die Arbeit evaluiert.

7

Fazit und Ausblick

Hier wird ein Fazit und ein Ausblick gegeben.

7.1. Fazit

Fazit.

7.2. Ausblick

Ausblick.

Generative Adversial Network

Generative Adversial Networks, kurz GANs, sind eine neu aufkommende Technologie im Bereich des maschinellen Lernens und künstlichen Intelligenz. Inspiriert von Ian Goodfellow und seinen Kollegen im Jahre 2014 bieten die GANs eine effiziente Möglichkeit tiefe Repräsentationen von Daten zu erlernen, ohne dass die Notwendigkeit besteht große Mengen von annotierten Trainingsdaten bereitzustellen. Dies wird durch das Backpropagationsverfahren und ein Wettbewerb zwischen zwei neuronalen Netzwerken, den Generator und den Diskriminator, erzielt. Dadurch werden viele neue Lösungsansätze zur Erzeugung von realistischen Inhalten angeboten. Ihre Anwendungen reichen von der Bildgenerierung bis zur Superresolution und der Erzeugung von Text.

Verfahren

Die Hauptkomponenten von einem GAN ist der Generator und Diskriminator. Beide neuronalen Netze werden gleichzeitig trainiert und stehen in einem Wettbewerb, bei dem der Generator versucht den Diskriminator durch die Erzeugung von synthetischen Inhalten zu täuschen. Das gesamte Netzwerk wird so trainiert, dass die Glaubwürdigkeit des Generators erhöht wird, so dass der Diskriminator nicht mehr zwischen den Eingaben unterscheiden kann. Die Netzwerke werden typisch von mehrschichtigen Netzwerken aus convolutional und fully connected layers implementiert.

Generator

Der Generator wird zum Generieren von unechten Daten, wie Bilder und Text, verwendet. Dieser besitzt keinen Zugriff auf den realen Datensatz und lernt in Folge dessen nur durch die Interaktion mit dem Diskriminator. Dieser gilt als optimal, falls der Diskriminator nur noch 50% der Eingaben richtig voraussagt.

Diskriminator

Die Aufgabe des Diskriminator besteht darin, zwischen den realen und unechten Eingaben unterscheiden zu können. Er kann sowohl auf die synthetischen Daten und auf den realen Datensatz zugreifen. Falls der Diskriminator nicht mehr richtig unterscheiden kann, so gilt er als konvergiert und optimal, falls seine Genauigkeit zur Klassifizierung maximiert wird. In Falle eines optimalen Diskriminators, wird sein Training gestoppt und der Generator wird allein weitertrainiert, um seine Genauigkeit wieder zu senken.

Training

Das Training erfolgt durch das Finden der Parameter für beide Netze. Dabei wird das Backpropagation auf beide Netze angewendet, um die Parameter zu verbes-

sern. Das Ziel ist die Optimierung beider Netze. Dabei wird das Training häufig als herausfordernd und instabil beschrieben, da einerseits das Finden der Konvergenz beider Modell schwierig erscheint. Andererseits kann der Generator für verschiedene Eingaben sehr ähnliche Muster erzeugen und der Diskriminatorverlust schnell gegen Null konvergiert, so dass kein zuverlässiger Weg für den Gradientenaktualisierung zum Generator existiert. Um den Problemen entgegenzukommen, wurden verschiedene Ansätze vorgeschlagen, wie das Verwenden von heuristischen Verlustfunktionen. Ebenso könnte ein zusätzliches Rauschen auf den Datensatz vor der Verwendung Abhilfe schaffen.

Anwendung

Ursprünglich ist die Entwicklung von GAN für das unüberwachte maschinelle Lernen gedacht. Jedoch weist die Architektur ebenso gute Ergebnisse in semiüberwachten und in Reinforcement Learning auf. Dadurch wird sie in umfangreichen Bereichen, wie im Healthcare, Mechanik und Banking verwendet. Beispielsweise werden GAN in der Medizin für die Identifizierung von chronischen Erkrankungen angewendet. Jedoch können durch die Nutzung von GANs ebenso 3D Objekte identifiziert und reale Bilder und Text generiert werden.

Limitationen

Aufgrund dessen, dass ein Generative Adversarial Network Inhalte erzeugen kann, die dem realen fast identisch aussehen, kann dies Probleme in der realen Welt schaffen, insbesondere bei der menschlichen Bildsynthese. Die Bilder können von Betrügern verwendet werden, um in den sozialen Medien falsche Profile zu erstellen. Diese kann ebenso durch den Einsatz von GANs verhindert werden, indem einzigartige und pragmatische Bilder von Personen erzeugt werden, die nicht existieren



Anhang - Code

Hier sehen Sie den gesamten Quellcode!

B

Anhang - Dokumentationen

Hier sehen Sie die gesamten Dokumentationen zu den erstellten Programmen.

Literaturverzeichnis

- [Haf18] HAFNER, E.G.: *Lineare Algebra für Dummies*. Wiley, 2018 (Für Dummies). <https://books.google.de/books?id=XfWADwAAQBAJ>. – ISBN 9783527819430

Abbildungsverzeichnis

Tabellenverzeichnis

Code-Auszugs-Verzeichnis

Glossar

- **C++:**
Hardwarenahe, objektorientierte Programmiersprache.
- **HTML:**
Hypertext Markup Language - textbasierte Auszeichnungssprache zur Strukturierung elektronischer Dokumente.
- **HTTP:**
Hypertext Transfer Protocol - Protokoll zur Übertragung von Daten auf der Anwendungssicht über ein Rechnernetz.
- **iARS:**
innovative Audio Response System - System mit zwei Applikationen (iARS-master-App; iARS-student-App), dass sich zum Einsetzen von e-TR-ainer-Inhalten in Vorlesungen eignet.
- **ISO:**
Internationale Vereinigung von Normungsorganisationen.
- **JavaScript:**
Skriptsprache zu Auswertung von Benutzerinteraktionen.
- **Konstruktor:**
Beim Erzeugen einer Objektinstanz aufgerufene Methode zum Initialisieren von Eigenschaften.
- **MySQL:**
Relationales Datenbankverwaltungssystem.
- **OLAT:**
Online Learning and Training - Lernplattform für verschiedene Formen von webbasiertem Lernen.
- **OOP:**
Objektorientierte Programmierung - Programmierparadigma, nach dem sich die Architektur eine Software an realen Objekten orientiert.
- **Open Source:**
Software, die öffentlich von Dritten eingesehen, geändert und genutzt werden kann.
- **PHP:**
Skriptsprache zur Erstellung von Webanwendungen.
- **Python:**
Skript- und Programmiersprache, die unter Anderem objektorientiertes Programmieren ermöglicht.

- **Shell:**
Shell oder auch Unix-Shell - traditionelle Benutzerschnittstelle von Unix-Betriebssystemen
- **Spyder:**
Entwicklungsumgebung für wissenschaftliche Programmierung in der Programmiersprache Python.
- **SymPy:**
Python-Bibliothek für symbolische Mathematik.