

Generative Adversarial Networks (GAN): Eine Übersicht und Beispiele zum Thema GAN

Marius Schmidt-Colberg
Hochschule der Medien
Stuttgart, Germany
Ms576@hdm-stuttgart.de

ABSTRACT

UPDATED—September 1, 2020. In letzter Zeit sind Generative Adversarial Networks (GANs) zu einem Forschungsschwerpunkt der künstlichen Intelligenz geworden. Inspiriert von einem Nullsummenspiel mit zwei Spielern, bestehen GANs aus einem Generator und einem Diskriminator, die beide nach der Idee des kontradiktorischen Lernens trainiert werden. Das Ziel von GANs ist es, die potenzielle Verteilung von realen Datenproben zu schätzen und neue Proben aus dieser Verteilung zu generieren. Seit ihrer Einführung wurden GANs aufgrund ihrer enormen Aussichten für Anwendungen, einschließlich Bild- und Bildverarbeitung, Sprachverarbeitung usw., umfassend untersucht. In diesem Papier wird die Entstehungsgeschichte von GANs erläutert und grundlegend erklärt. Darüber hinaus werden ein paar unterschiedliche Arten von GANs vorgestellt und grob erklärt. Zuletzt wird auf die Herausforderungen, Vorteile, Einfluss und ein Ausblick von GANs gegeben.

Author Keywords

Generative Adversarial Networks, Deep learning, Bild-zu-Bild-Übersetzung, CycleGAN, GauGAN, Face Image Synthesis, Imagepainting.

CSS Concepts

• **Human-centered computing~Human computer interaction (HCI); Haptic devices; User studies; Please use the 2012 Classifiers and see this link to embed them in the text:** https://dl.acm.org/ccs/ccs_flat.cfm

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

CHI 2020, April 25–30, 2020, Honolulu, HI, USA.

© 2020 Copyright is held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-6708-0/20/04...\$15.00.

DOI: <https://doi.org/10.1145/3313831.XXXXXXX>

*update the above block and DOI per your rightsreview confirmation (provided after acceptance)

INTRODUCTION

In den letzten Jahrzehnten hat sich das Wissen über künstliche Intelligenz und ihre Anwendung in verschiedenen Wissenschaften rasant erweitert, nachdem die Leistung von Computersystemen gestiegen ist und große Datensätze in verschiedenen Branchen entstanden sind. Das maschinelle Lernen [1], als einer der breiten und weit verbreiteten Zweige der künstlichen Intelligenz, befasst sich mit der Anpassung und Erforschung der Verfahren und Algorithmen, auf deren Grundlage Computer und Systeme ihre Lernfähigkeiten entwickeln. Algorithmen für maschinelles Lernen müssen Merkmale aus Rohdaten extrahieren. Bei bisherigen Verfahren wurden diese Merkmale manuell bereitgestellt und dem jeweiligen Algorithmus zugeführt, eine zeitaufwändige und unter Umständen unvollständige Aufgabe. Repräsentationslernen oder Feature-Lernen [2] bietet dem System die Möglichkeit, die für die Feature-Erkennung, Klassifizierung und andere Fragestellungen erforderlichen Repräsentationen automatisch zu entdecken. Mit anderen Worten, das Repräsentationslernen wandelt Eingabedaten in sinnvolle Ausgaben um. Deep Learning [3] ist eine Art des Repräsentationslernens, das darauf abzielt, superabstrakte Konzepte im Datensatz gemäß einer Reihe von Algorithmen zu modellieren. Dieser Prozess wird durch einen tiefen Graphen modelliert, der aus mehreren Schichten von linearen und nichtlinearen Transformationen besteht. Abb.1 veranschaulicht diese Definitionen in der Struktur der Hierarchie.

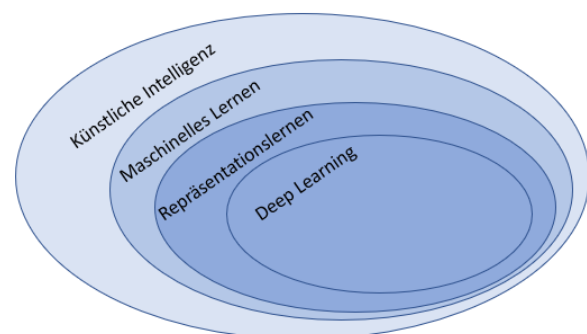


Figure 1 - Künstliche Intelligenz, maschinelles Lernen, Repräsentationslernen und Deep Learning auf einen Blick

Algorithmen für maschinelles Lernen lassen sich grob in zwei Hauptkategorien einteilen - überwachtes Lernen und

unüberwachtes Lernen. Beim überwachten Lernen wird ein Datensatz mit verschiedenen Merkmalen benötigt, wobei die einzelnen Daten beschriftet werden müssen. Diese Art von Algorithmen wird zur Lösung von Klassifizierungs- und Regressionsproblemen verwendet. Im Gegensatz dazu benötigt das unüberwachte Lernen Ein Datensatz ohne explizites Label. Bei dieser Art des Lernens wird dem Netzwerk nicht gesagt, nach welchem Muster es suchen soll, und es gibt keine klare Fehlermetrik. Einige gängige Beispiele für unüberwachtes Lernen sind generative Modelle, Dichteschätzung, Clustering, Rauscherzeugung und Rauschunterdrückung. Beim überwachten Lernen ist die manuelle Verwaltung/Erfassung von beschrifteten Daten kostspielig und zeitaufwändig, außerdem ist die automatisierte Datenerfassung schwierig und kompliziert. Der Erfolg des Deep Learnings ist der großen Datenmenge zuzuschreiben, die benötigte große Datenmenge, welche wie bereits erwähnt oft mit kostspieligen und zeitaufwändigen Methoden gesammelt wird, kann auch mithilfe von Datenerweiterungsmethoden erreicht werden. Die Anwendung dieser Methode auf das Modell erhöht die Fähigkeit des Modells, erzeugt einen regelmäßigen Effekt und reduziert den Generalisierungsfehler. Die Datenerweiterung erfolgt durch die Erstellung neuer und akzeptabler Stichproben des Trainingsdatensatzes, einschließlich der Anwendung von Operatoren, wie z. B. Drehen, Zuschneiden, Zoomen und andere einfache Transformationen auf Bilder. Allerdings können mit dieser Methode nur Daten mit begrenzten Informationen gewonnen werden. Die modernste Art der Datenanreicherung ist die Erzeugung von qualitativ hochwertigen Mustern durch generative Modelle. In Anbetracht der Fähigkeit generativer Netzwerke, Bilder in großem Umfang zu generieren, wird erwartet, dass der gravierende Mangel an beschrifteten Daten erheblich gemildert wird. Generative Modelle arbeiten üblicherweise auf Basis der „Markov-Kette“, der „Maximum-Likelihood-Schätzung“ (MLS) und der stochastische Inferenz. Die „Restricted Boltzmann Machine“ (RBM) und die daraus entwickelten Modelle wie das „Deep Belief Network“ (DBN) und die „Deep Boltzmann Machine“ (DBM) basieren auf MLS. Die von diesen Methoden erzeugten Muster vergleichen die Datenverteilung mit der experimentellen Verteilung der Trainingsdaten [4].

Diese Prototypen haben mehrere schwerwiegende Einschränkungen und sind möglicherweise nicht gut verallgemeinerbar. Generative Adversarial Networks (GANs) wurden von Ian Goodfellow [5] als Idee für halbüberwachtes und unbeaufsichtigtes Lernen vorgeschlagen. Yann LeCun, Leiter der IBC-Forschung bei Facebook, stellte adversariales Training als die interessanteste Idee der letzten zehn Jahre im maschinellen Lernen vor [6]. Abb.2 zeigt deutlich das rasante Wachstum der Anzahl veröffentlichter Artikel auf dem Gebiet der GANs in den letzten Jahren. GANs haben beeindruckende Verbesserungen gegenüber früheren generativen Methoden

wie Variations-Autocodierern oder Restricted Boltzmann Machines gezeigt. Abb.3 zeigt den Fortschritt der GANs über mehrere aufeinanderfolgenden Jahre für die Gesichtsgenerierung

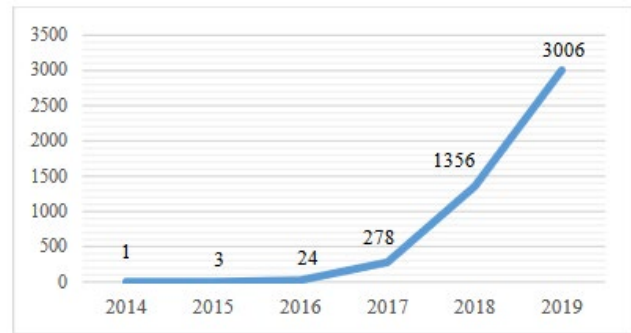


Figure 2 - Künstliche Intelligenz, maschinelles Lernen, Repräsentationslernen und Deep Learning auf einen Blick [7]



Figure 3 - Künstliche Intelligenz, maschinelles Lernen, Repräsentationslernen und Deep Learning auf einen Blick [8]

Bisher wurde Forschung durchgeführt, um Generative Adversarial Networks zu überprüfen. Insgesamt wurde die Einführung von GANs, ihre Anwendungen in verschiedenen Bereichen wie Computer Vision [9], Signalverarbeitung [10], Bildsynthese und -bearbeitung [11], Sprachverarbeitung [12], die Kombination von GANs mit einem Autoencoder [13], die Einführung der wichtigsten Architekturen von GANs [14] und die Untersuchung der Beziehung zwischen GANs und paralleler Intelligenz [15] behandelt. Die Hauptidee von GANs ist von einem Zwei-Personen-Nullsummenspiel inspiriert, bei dem die Gewinne (oder Verluste) eines Teilnehmers genau gleich den Verlusten (oder Gewinnen) des anderen sind. Die Gesamtgewinne der Teilnehmer minus die Gesamtverluste sind gleich Null. Anders und einfacher ausgedrückt kann man sich dieses Nullsummenspiel wie bei einem Wettbewerb zwischen Fälschern und der Polizei vorstellen. Die Fälscher wollen gefälschtes Geld herstellen und es echt aussehen lassen, und die Polizei will sich jeden einzelnen Geldschein ansehen und feststellen, ob er gefälscht ist [16].

Die GAN-Architektur besteht aus zwei Netzwerken, die gemeinsam trainieren: dem Generator und dem Diskriminator. Der Generator versucht, die statistische Verteilung von realen Daten zu lernen, um gefälschte Daten

zu erzeugen, die von realen Daten nicht zu unterscheiden sind, um den Diskriminator dazu zu verleiten, diese für echte Eingaben zu halten. Im Gegensatz dazu ist der Diskriminator ein Klassifikator, der unterscheidet, ob ein gegebener Inhalt wie echte Daten aus dem Datensatz oder wie künstlich synthetisierte Daten aussieht. Da beide Teilnehmer sich ständig optimieren, um ihre Fähigkeiten zu verbessern, und versuchen, aus ihren eigenen Schwächen zu lernen und die Schwächen des anderen auszunutzen, werden die neuronalen Netze während des Trainingsprozesses immer stärker. Der Optimierungsprozess zielt darauf ab, ein Nash-Gleichgewicht zwischen den beiden Teilnehmern herzustellen. In den Wirtschaftswissenschaften und der Spieltheorie ist ein Nash-Gleichgewicht ein stabiler Systemzustand, bei dem eine Interaktion zwischen verschiedenen Teilnehmern stattfindet. Unter diesen Umständen kann kein Teilnehmer profitieren, indem er einfach einseitig die Strategie ändert, ohne die Strategie der anderen Teilnehmer zu verändern, was genau das Ziel des GAN ist. Generator und Diskriminator erreichen einen Zustand, in dem der eine keinen Fortschritt machen kann, ohne den anderen zu verändern. Heutzutage sind GANs in verschiedenen Beispielen weit verbreitet, wie z. B. Text-zu-Bild-Synthese, Bild-zu-Bild-Übersetzung und viele mögliche medizinische Anwendungen. Abb. 4 zeigt den prozentualen Anteil der Gesamtzahl der bis 2019 veröffentlichten Artikel in verschiedenen Disziplinen

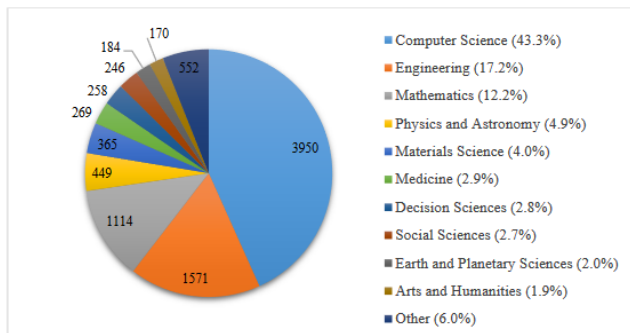


Figure 4 - Taxonomie der Anzahl der in Scopus indizierten Artikel basierend auf verschiedenen Disziplinen von 2014 bis 2019. Der Artikel aus[7].

Angesichts der Bedeutung von GANs und ihrer Anwendung in verschiedenen wissenschaftlichen Bereichen ist es notwendig, sie umfassend vorzustellen, die in diesem Bereich durchgeführte Forschung zu untersuchen und die Herausforderungen in diesem Bereich zu beschreiben. Daher hat sich diese Arbeit mit diesen Themen beschäftigt. Es ist erwähnenswert, dass ein besseres Verständnis von GANs die Wahrnehmung der Konzepte des Deep Learning erfordert.

Dieses Paper ist wie folgt aufgebaut. In Abschnitt II werden die wichtigsten Konzepte und die Theorie des GAN geklärt, sowie die Mathematik hinter den GANs erklären. Abschnitt III zeigt zwei fortgeschrittene GANs und Abschnitt IV beschreibt einige der bedeutenden Anwendungen von GANs auf dem Gebiet der Computer Vision.

2 - GENERATIVE ADVERSARIAL NETWORKS

Generativ Adversarial Networks sind eine potente Klasse von neuronalen Netzen, die einen intelligenten Ansatz zum unüberwachten Lernen verfolgen. Die GANs sind in der Lage, Muster zu generieren, die der realen Datenverteilung sehr ähnlich sind, indem sie automatisch die zugrunde liegende Struktur erforschen und die vorhandenen Regeln und Muster der realen Daten lernen [17]. Das GAN-Framework greift natürlich einen spieltheoretischen Ansatz auf. GANs enthalten normalerweise zwei neuronale Netze, die trainiert werden und gegeneinander antreten: einen Generator und einen Diskriminator. Der Grund für die Wahl des Wortes "adversarial" in GAN ist, dass diese beiden Netze während des gesamten Trainingsprozesses in ständigem Konflikt stehen. Diese beiden Netzwerke können mit Fälscher (Generator) und Polizei (Diskriminator) verglichen werden. Der Generator versucht, eine Form von Geld zu erzeugen, die dem echten Geld ähnlich ist, indem er die neuesten Tricks lernt, um die Polizei, d.h. den Diskriminator, zu täuschen. Umgekehrt muss die Polizei ihre Informationen ständig aktualisieren, um Falschgeld zu erkennen. Die beiden Netzwerke aktualisieren ständig ihr Wissen und erhalten Rückmeldungen über ihre erfolgreichen Änderungen. Dieser Kampf geht so lange weiter, bis die Polizei nicht mehr in der Lage ist, echte Daten von gefälschten Daten zu unterscheiden, was bedeutet, dass der Fälscher gültige Muster generiert [18]. Die Architektur des GAN ist in Abbildung 5 dargestellt. $Xdata$ und $G(z)$ sind die echten Proben im Trainingsdatensatz bzw. die vom Generator G synthetisierten gefälschten Proben. Der Diskriminator D beurteilt die Wahrscheinlichkeit, dass die Eingabedaten echt oder gefälscht sind. Im GAN nimmt der Generator zunächst den Rauschvektor z (einen Zufallsvektor mit gleichmäßiger Verteilung oder Gauß-Verteilung) mit einer festen Länge als Eingabe. Dann synthetisiert der Generator neue Daten $G(z)$ aus Standardnormalverteilungen $Xdata$. Um das Problem besser zu verstehen, muss man sich vorstellen, dass für die Generierung eines Bildes nicht die anfängliche Eingabe eines Bildes, sondern ein Vektor von Zufallswerten benötigt wird. Nach dem Training werden die Punkte dieses mehrdimensionalen Vektors mit den Punkten im Problembereich abgeglichen, was zu einer komprimierten Darstellung der Datenverteilung führt. Dieser Vektorraum wird als latenter Raum oder als ein Vektorraum aus latenten Variablen bezeichnet. Latente Variablen umfassen wichtige, aber unbeobachtbare Variablen direkt für eine Domäne [19]. Modelle des maschinellen Lernens können den statistischen latenten Raum von Bildern, Musik und Geschichten erlernen und anschließend eine Reihe neuer Kunstwerke mit Spezifikationen erstellen, die denen von realen Proben dieses Raums ähneln [20].

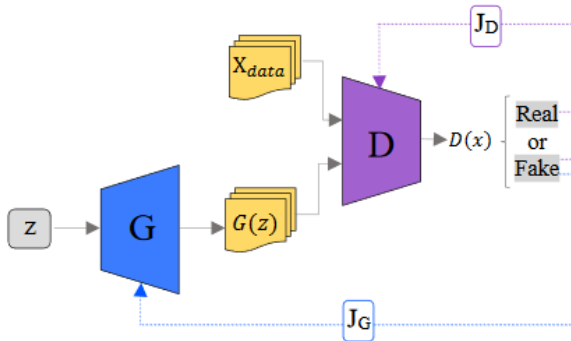


Figure 5 – Die Architektur von GANs. [25]

Ein Diskriminator arbeitet als binäre Klassifikation und unterscheidet gefälschte $G(z)$ Muster von echten X_{data} Mustern. Der Diskriminator wird so trainiert, dass er die Wahrscheinlichkeit maximiert, echten und gefälschten Daten die richtigen Labels zuzuweisen. Mit anderen Worten, wenn die Eingabe aus echten X_{data} Daten besteht, klassifiziert der Diskriminator sie als echte Daten und gibt einen numerischen Wert nahe 1 zurück. Andernfalls, wenn die Eingabe aus Daten besteht, die vom Generator generiert wurden, klassifiziert der Diskriminator sie als gefälschte Daten und gibt einen numerischen Wert nahe 0 zurück.

Der Generator und der Diskriminator können neuronale Netze, konvolutionelle neuronale Netze, rekurrente neuronale Netze und Autocoder sein. Daher benötigt der Diskriminator die Verlustfunktion J_D und der Generator die Verlustfunktion J_G , um die Netze zu aktualisieren (Abb.5). Der Generator aktualisiert seine Parameter nur durch die Backpropagationssignale der gefälschten Ausgabe. Im Gegensatz dazu erhält der Diskriminator mehr Informationen und aktualisiert seine Gewichte unter Verwendung der gefälschten und realen Ausgabe. GAN kann als ein Zwei-Spieler-Minimax-Spiel mit gleichzeitigem Training von Generator- und Diskriminatornetz modelliert werden. Minimax GAN Verlust wird als eine Optimierungsstrategie in Zwei-Spieler-Spielen betrachtet, bei der jeder Spieler seine Verluste reduziert oder die Ausgaben des anderen Spielers erhöht. Im GAN repräsentieren der Generator und der Diskriminator die beiden Spieler, die ihrerseits ihre Netzwerkgewichte aktualisieren. Minimax bezieht sich auf die Minimierung des Verlustes im Generator und die

Maximierung des Verlustes im Diskriminator [21]. Anders ausgedrückt: Der Diskriminator versucht, die Wahrscheinlichkeit zu maximieren, den Daten die richtigen Labels zuzuordnen. Im Gegensatz dazu versucht der Generator, eine Reihe von Stichproben zu erzeugen, die nahe an der realen Datenverteilung liegen, um die Kreuz-Entropie zu minimieren.

Ein Grund, der für Anfänger herausfordernd bleibt (siehe Abb. 7), ist das Thema der GAN-Verlustfunktionen. Die GAN-Optimierungsstrategie, als Minmax-Problem, wird als Gleichung 3 dargestellt. Zum besseren Verständnis wird die Gleichung 3 in die Gleichungen 1 und 2 zerlegt, wobei E die mathematische Erwartungsnotation ist. p_{data} steht für die Verteilung der realen Daten, während p_z die zufällige Rauschverteilung ist. Gemäß Gleichung 1, wenn $X=X_{data}$ (X sind die Eingabedaten für den Diskriminator), wird erwartet, dass der Diskriminator einen numerischen Wert nahe bei 1 in der Ausgabe anzeigt, d. h., es wird erwartet, dass X die realen Daten verteilt und $V(G,D)$ maximiert. Gemäß Gleichung 2, wenn $X=G(Z)$, gibt es zwei verschiedene Perspektiven, die das erste Kriterium des Problems aus der Perspektive des Diskriminators angehen. Vom Diskriminator wird erwartet, dass er es schafft, zu erkennen, dass die erzeugte Probe gefälscht ist, und dass er in seiner Ausgabe einen numerischen Wert nahe 0 anzeigt. $V(G,D)$ sollte auch unter diesen Umständen maximiert werden. Dies zeigt das zweite Kriterium des Problems aus der Sicht des Generators. Hier ist der Idealfall, dass der Generator in der Lage ist, den Diskriminator in die Irre zu führen, d. h. ein numerischer Wert nahe 1 wird in der Ausgabe angezeigt. Mit anderen Worten, der Generator wird so trainiert, dass er den Diskriminator täuscht, indem er $V(G,D)$ minimiert und eine reale Datenverteilung erhält. Schließlich zeigt Gleichung 3 aus mathematischer Sicht ein 2-Spieler-Minimax-Spiel mit der Wertfunktion $V(G,D)$. Abb. 6 veranschaulicht als Beispiel mehrere Schritte des gleichzeitigen Trainings von Generator und Diskriminator in einem GAN. In Abb.7(a) werden GANs durch gleichzeitiges Aktualisieren der Diskriminatorverteilung (blau, gestrichelte Linie) trainiert, so dass sie zwischen Stichproben aus der realen Datenverteilung (schwarz, gepunktete Linie) und der generierten Datenverteilung (grün, durchgezogene Linie) unterscheidet. In Abb. 7(b) wurde der Diskriminator trainiert, um zwischen echten und gefälschten Daten zu

$$\text{if } X = X_{data} \Rightarrow D(X) \rightarrow 1 \quad \Rightarrow \quad \max_D V(D,G) = E_{x \sim p_{data}(x)} [\log(D(x))] \quad (1)$$

$$\text{if } X = G(Z) \Rightarrow \begin{cases} D(X) \rightarrow 0; \text{ for } D & \Rightarrow \max_D V(D,G) = E_{z \sim p_z(z)} [\log(1 - D(G(z)))] \\ D(X) \rightarrow 1; \text{ for } G & \Rightarrow \min_G V(D,G) = E_{x \sim p_{data}(x)} [\log(D(x))] \end{cases} \quad (2)$$

$$\min_G \max_D V(D,G) = E_{x \sim p_{data}(x)} [\log(D(x))] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (3)$$

Figure 6 – GAN Verlustfunktion erklärt

unterscheiden, und er erfüllt seine Aufgabe problemlos. In Abb. 7(c) wird der Trainingsprozess für den Diskriminator gestoppt und nur der Generator wird trainiert, um die Verteilung der gefälschten Daten näher an die Verteilung der echten Daten heranzuführen. Diese Aktualisierungen werden fortgesetzt, bis der Diskriminator nicht mehr unterscheidet (Abb.7(d)). Es ist erwähnenswert, dass der Prozess des Trainings von GANs nicht so einfach und geradlinig ist wie der in Abb. 7 dargestellte Prozess. Die gefälschte Datenverteilung wird unter idealen Bedingungen vollständig mit der realen Datenverteilung überlagert, während es in der Praxis verschiedene Herausforderungen gibt

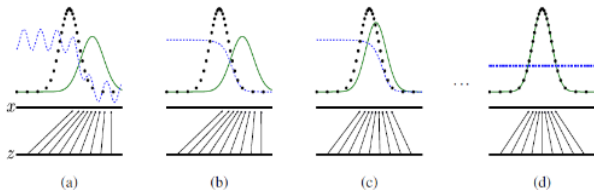


Figure 7 – Ein Beispiel für einen GAN-Trainingsprozess. Entwicklung der generierten Datenverteilung (grün) in Richtung der realen Datenverteilung (schwarz) und der Entscheidungsgrenze (blau). Die Abbildung stammt aus[7].

GANs sind eine Gruppe von Netzwerken mit einem sehr komplexen und anspruchsvollen Trainingsprozess, da sowohl Generator- als auch Diskriminator-Netzwerke gleichzeitig in einer adversen Art und Weise trainiert werden. Die gesamte Grundlage von GANs ist das Gleichgewicht zwischen den beiden Netzwerken. Mit anderen Worten, die Natur des Optimierungsproblems ändert sich jedes Mal, wenn die Parameter eines der Netzwerke aktualisiert werden, was zur Bildung eines dynamischen Systems führt. Die technische Herausforderung beim Training von zwei konkurrierenden neuronalen Netzen ist deren verzögerte Konvergenz [21].

3 - EINSATZ VON GANS IN COMPUTER VISION

GANs haben sich in vielen Bereichen der Computer Vision als enorm leistungsfähig gegenüber traditionellen Methoden erwiesen. Aufgrund der Eigenschaften des gegnerischen Mechanismus und der kontinuierlichen Selbstverbesserung sind GANs in der Lage, Merkmale aus vorhandenen stochastischen Verteilungen zu lernen und geeignete visuelle Merkmale zu erfassen. Dieser Abschnitt stellt einige führende Anwendungen von GANs vor, darunter die Bild-zu-Bild-Übersetzung, den CycleGAN, GauGAN, Face Image Syntesis, und Imagepainting.

3.1. BEISPIELE BILD-ZU-BILD-ÜBERSETZUNG UND CYCLEGAN

Da die automatische Sprachübersetzung als ein grundlegendes Problem des maschinellen Lernens gilt, wird das Problem der Bild-zu-Bild-Übersetzung in ähnlicher Weise gesehen. Die Bild-zu-Bild-Übersetzung ist ein unüberwachtes, kontrolliertes Verfahren zur Rückführung eines Bildes von einer Darstellung in eine andere. Die Übersetzung des Bildes von einer Ebene in eine andere ist

ein anspruchsvolles Problem, das oft ein spezifisches Modell und Verlustfunktionen für den Datensatz erfordert. Klassische Methoden verwenden Klassifizierung oder Regression pro Pixel. Bei diesen Methoden wird jedes der Ausgabepixel in Bezug auf das Eingabebild unabhängig von den früheren Pixeln vorhergesagt, was zum Verlust eines großen Teils des semantischen Inhalts des Bildes führt. Idealerweise wird jedoch ein allgemeiner Ansatz benötigt, der ein Modell und die Verlustfunktion für mehrere Bild-zu-Bild-Übersetzungsprobleme verwenden kann. Die Methode Pixels-to-Pixels (Pix2Pix) [22] wird unter Verwendung der cGAN-Architektur eingeführt, um dieses Problem zu lösen. Mit ihrem hohen Potenzial kann diese Architektur echte, hochauflösende Bilder in verschiedenen Bild-zu-Bild-Übersetzungsanwendungen erzeugen. Sie ermöglicht auch die Erstellung großer Bilder (z. B. 256×256) im Vergleich zu älteren GAN-Modellen. Abb.15 zeigt die Leistung von Pix2Pix. Sie ermöglicht auch die Erstellung großer Bilder (z. B. 256×256) im Vergleich zu älteren GAN-Modellen. Abb.15 zeigt die Leistung von Pix2Pix. In der Pix2Pix-Architektur ist der Generator durch das U-Netz [23] und der Diskriminator durch PatchGAN [24] inspiriert. Beide Netzwerke, die in dieser Architektur verwendet werden, sind tiefe Faltungsneuronale Netze. In PatchGAN wird die Klassifizierung in einem Schritt für alle Bilder durchgeführt. Dafür wird jedes Bild zunächst in $n \times n$ Abschnitte unterteilt. Dann wird für jedes Feld separat vorhergesagt, ob das Bild echt oder gefälscht ist. Schließlich wird die endgültige Klassifizierung durch Bildung des Mittelwerts aller Antworten durchgeführt. Das heißt, PatchGAN bewertet die Struktur auf der Skala der Patches. pix2pix hat seine Leistungsfähigkeit in einer Vielzahl von Anwendungen bewiesen, wie z. B. bei der Fleckennormalisierung von Histopathologie-Bildern, es wurde als Stain-to-Stain-Translation (STST) verwendet [25].



Figure 8 – Durch den pix2pix-Modus erzeugte Bildübersetzung [22]

Weitere Studien auf diesem Gebiet haben pix2pixHD [26] vorgeschlagen, um die Auflösung der generierten Muster zu erhöhen. Bei dieser Methode wird eine neue adversarische

Verlustfunktion verwendet, um Bilder mit einer Auflösung von 2048×1024 zu erzeugen. Pix2pix erfordert ein Training des gepaarten Datensatzes, was eine seiner Einschränkungen ist. Das heißt, es muss ein Datensatz aus den Eingabebildern vor der Übersetzung und den Ausgabebildern aus denselben Bildern nach der Übersetzung erstellt werden. Solche Bildpaare gibt es aber in vielen Fällen nicht. Die unpaarige Bild-zu-Bild-Übersetzungsmethode im zyklus-konsistenten GAN (cycleGAN)[27] kann verwendet werden, um dieses Problem zu überwinden. Diese Methode verwendet den Zyklus-Konsistenzverlust, der darauf abzielt, das Originalbild nach einem Translations- und inversen Translationszyklus zu erhalten. Dieser Zyklus benötigt keine Bildpaare für das Training (siehe Abb.16).

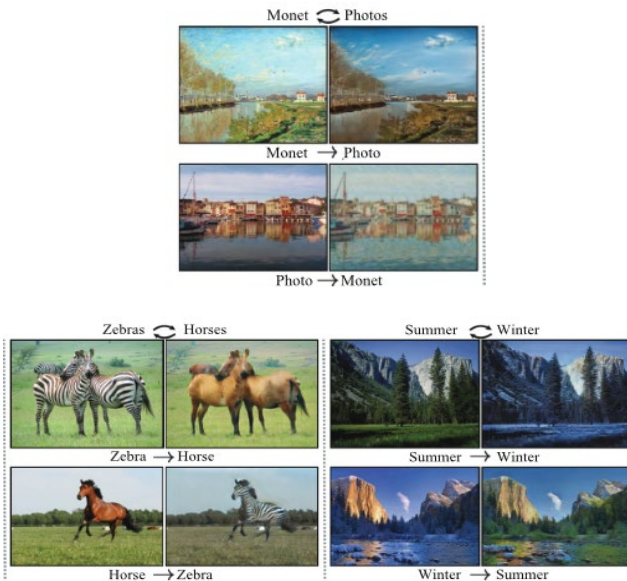


Figure 9 – Ein Beispiel Bildübersetzung, die mit dem CycleGAN-Modell[27] erzeugt wird. Dieses Modell übersetzt automatisch ein Bild von einem in das andere und umgekehrt.

3.2. BEISPIEL GAUGAN

Zwei Jahre später als Pix2PixHD im Jahr 2017 auf der IEEE Conference on Computer Vision and Pattern Recognition (CVPR) vorgestellt wurde, kam 2019 Nividias Vorstellung auf der CVPR GauGAN. Dieser neue Algorithmus, GauGAN, kann Kritzeleien effektiv in die Realität umsetzen. GANs werden typischerweise verwendet, um Daten zu generieren. Sie stellen typischerweise eine verrauschte Eingabe bereit, die das Netzwerk verwendet, um eine zugehörige Ausgabe zu erzeugen. Diese Art von GAN ist insofern nützlich, als dass es zur Generierung von Daten nichts anderes benötigt als zufälliges Rauschen, das mit jeder numerischen Software erzeugt werden kann. Auf der anderen Seite nehmen bedingte GANs im Allgemeinen eine bestimmte Eingabe, die die zu generierenden Daten bestimmt (mit anderen Worten, die generierten Daten sind von der Eingabe abhängig, die wir bereitstellen). In GauGAN zum Beispiel ist die Eingabe eine semantische Segmentierungskarte und das GAN generiert ein reales Bild,

das durch das Eingabebild bedingt ist, wie im folgenden Beispiel.

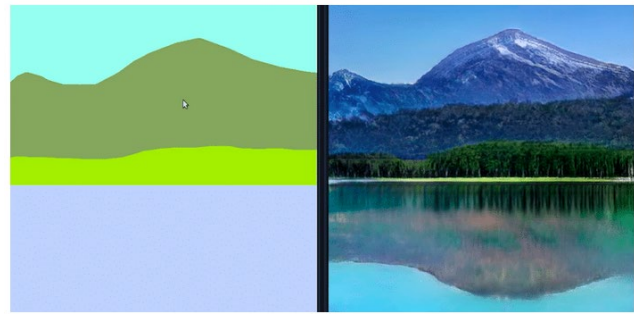


Figure 10 – Ein Beispiel von GauGAN, dieses Modell kann Kritzeleien effektiv in die Realität umsetzen. [28]

3.3. BEISPIEL IMAGESUPER-RESOLUTION(SR)

Imagesuper-resolution (SR) wurde häufig bei Satelliten-, medizinischen und militärischen Bildern eingesetzt. Deep Learning-Techniken halfen bei der Lösung dieses Problems, indem sie die hochfrequenten Details vorhersagten, die in Bildern mit niedriger Auflösung verloren gehen. Super-Resolution GAN (SRGAN)[29] wurde eingeführt, um die Bildauflösung zu verbessern. Bei dieser Methode wird ein niedrig aufgelöstes Bild empfangen und ein hochaufgelöstes Bild im 4-fachen Maßstab erzeugt. Die von SRGAN erzeugten Texturinformationen sind nicht real genug und werden von Bildrauschen begleitet. Um dieses Problem zu lösen, wurde das Enhanced Super-Resolution GAN (ESRGAN) [30] eingeführt. In dieser Netzwerkarchitektur wurden der adversarische Verlust und der Wahrnehmungsverlust verbessert. Darüber hinaus wurde ein neues Netzwerk namens „Residual-in-Residual Dense Block“ (RRDB) auf der Basis relativistischer GANs [31] eingeführt, das, wie in Abb. 14 gezeigt, eine bessere Leistung als SRGAN aufweist.

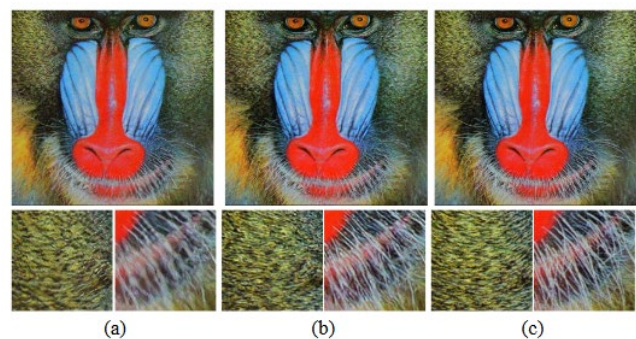


Figure 11 – Ein Beispiel mit verschiedenen GANs erzeugte Bild-Superauflösung. (a) SRGAN[29]; (b) ESRGAN[30]; (c) Ground truth

3.4. BEISPIEL FACE IMAGE SYNTHESIS

Die Gesichtssynthese ist weit verbreitet, auch in der Gesichtserkennung. Obwohl zahlreiche datenbasierte Deep-Learning-Ansätze dafür vorgeschlagen wurden, bleibt dieser Bereich eine Herausforderung. Da das menschliche Sehen

empfindlich auf Gesichtsverformungen und -deformationen reagiert, ist die Erzeugung echter Gesichtsbilder keine leichte Aufgabe. GANs haben gezeigt, dass sie in der Lage sind, qualitativ hochwertige Gesichtsbilder mit feiner Textur zu erzeugen.



Figure 12 – Ein Beispiel frontale Gesichtsbildsynthese durch Profilbilder mit der TP-GAN-Methode[32]

Two-Pathway GAN (TP-GAN) [32] kann ein Profilbild verwenden, um hochauflösende frontale Gesichtsbilder zu erzeugen (siehe Abb.17). Diese Technik kann lokale und globale Informationen wie beim Menschen berücksichtigen. Das mit dieser Methode erzeugte Gesichtsbild bewahrt gut die Merkmale der Identität einer Person. Es kann auch mehrere Bilder in verschiedenen Modi und Beleuchtungen verarbeiten. Es hat eine Zwei-Wege-Architektur. Ein globaler Generator wird trainiert, um globale Merkmale zu erzeugen, und ein lokaler Generator, um Details um die Gesichtsmarkierungen (markierte Punkte) zu erzeugen. Darüber hinaus kombiniert der Self-Attention GAN (SAGAN) [33] den Self-Attention-Block mit dem GAN für die Bildsynthese, um das Problem der weitreichenden Abhängigkeit zu lösen. Dadurch wird dem Diskriminator zugetraut, dass er die Abhängigkeit zwischen zwei weit entfernten Merkmalen bestimmen kann. Basierend auf SAGAN wird die BigGAN-Methode [35] vorgeschlagen, um die Diversität und Genauigkeit der generierten Muster zu erhöhen, indem die Stapelgröße erhöht und ein Trunkierungs-Trick verwendet wird. Im traditionellen Ansatz wird für die latente Verteilung z der Generator mit z als Eingabe gefüttert. In BigGAN wird z jedoch in mehrere Schichten des Generators eingebettet, um die Auflösungseigenschaften und verschiedene Ebenen zu beeinflussen. Wie in Abb.18 gezeigt, sind die generierten Samples realistisch



Figure 13 – Ein Beispiel welches eine Bildsynthese generiert durch das BigGAN-Modell [34]

Für die Gesichtsbildsynthese im neuen Zustand wurde das Disentangled Representation Learning GAN (DRGAN) Verfahren[35] eingeführt. Der Generator verwendet eine Encoder-Decoder-Architektur, die separate Repräsentationen für Gesichtsbilder, Encoder-Ausgang und Decoder-Eingang erlernt. Der Diskriminator enthält zwei Teile, d. h. Identitäts- und Zustandsklassifikation. Die Ergebnisse der Experimente zeigen, dass DRGAN die bestehenden Gesichtserkennungsverfahren im stationären Zustand übertrifft. Die Architektur des Face Frontalization GAN (FF-GAN) [36] verwendet ein 3D-Morphable Model (3DMM) [37] in der GAN-Struktur. 3DMM liefert Geometrie und Aussehen für Gesichtsbilder. Ebenso sind die 3DMM-Repräsentationen klein im Volumen. Schnelle FF-GAN-Konvergenz und hochauflösende Vollgesichtsbilder sind von hoher Qualität.

3.4. BEISPIEL IMAGE PAINTING

Beim Image Painting wird versucht, die verlorenen Teile eines Bildes so zu rekonstruieren, dass der Betrachter die rekonstruierten Bereiche nicht mehr erkennen kann. Diese Methode wird häufig verwendet, um unerwünschte Objekte aus einem Bild zu entfernen oder beschädigte Teile in alten Bildern wiederherzustellen. Bei traditionellen Techniken wurden die Löcher im Bild durch Duplizieren der Pixel des Originalbildes oder einer Bibliothek von Bildern gefüllt. Auf Deep Learning basierende Ansätze haben vielversprechende Ergebnisse für die Wiederherstellung großer, in einem Bild verloren gegangener Bereiche erbracht. Diese Methoden können annehmbare Bildstrukturen und Texturen erzeugen. Einige dieser Techniken wurden unter Verwendung von Faltungsnetzwerken vorgeschlagen, wobei die Leistung beim Füllen von Lücken mit den richtigen Merkmalen schlecht war. Daher wurden generative Modelle entwickelt, um die richtigen, im Trainingsprozess bekannten Merkmale zu finden. Die erste auf GANs basierende Bildrestaurationsmethode wird als Context-Encoder [38] vorgestellt. Diese Methode wird auf Basis der Encoder-

Decoder-Architektur trainiert, um aus der Bildsemantik beliebige fehlende große Regionen in Bildern abzuleiten. Dennoch kann bei dieser Methode eine vollständig verbundene Schicht keine genauen räumlichen Informationen speichern. Der Context-Encoder erzeugt manchmal unscharfe Texturen im Verhältnis zu den Bereichen um das Loch. In [39] wurde dann die Idee der "Stilübersetzung" mit dem Kontext-Encoder kombiniert und ein neuer Ansatz zur Wiederherstellung hochaufgelöster Bilder vorgeschlagen. Dieses Modell ist jedoch nicht leistungsfähig genug, um den fehlenden Bereich mit komplexen Strukturen zu füllen. In Abb.19 ist ein Beispielergebnis dieser Methode und des Context Encoders dargestellt.

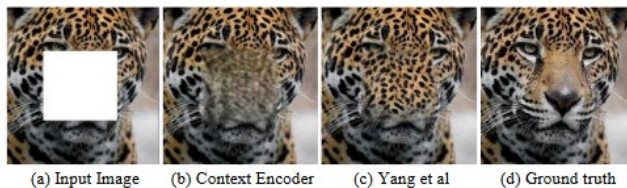


Figure 15 – Ein Beispiel Vergleich zwischen Context Encoder [76] und Yang et al.[39]

Ebenso verwendeten die Forscher in [40] DCGAN zur Wiederherstellung des Bildes, das verlorene Teile des Bildes erfolgreich generieren kann. In [41] wurde ein GAN-basierter Ansatz zur Bildwiederherstellung vorgeschlagen, der mit globalen und lokalen Umgebungen kompatibel ist. Die Eingabe ist ein Bild mit einer zusätzlichen binären Maske zur Darstellung des fehlenden Lochs. Die Ausgabe eines restaurierten Bildes hat die gleiche Auflösung. Der Generator verwendet die Encoder-Decoder-Architektur und erweiterte Faltungsschichten anstelle von Standard-Faltungsschichten, um einen größeren Raum zu unterstützen [42]. Es gibt zwei Diskriminatoren, einen globalen Diskriminator, der das gesamte Bild als Eingabe erfasst, und einen lokalen Diskriminator, der einen kleinen Bereich mit seinem Loch als Eingabe abdeckt. Die beiden Diskriminator-Netzwerke stellen sicher, dass das resultierende Bild sowohl auf der "globalen" als auch auf der "lokalen" Skala kompatibel ist. Dies führt zu einem wiederhergestellten Bild für hochauflösende Bilder mit beliebigen Löchern.

4. HERAUSFORDERUNGEN, VORTEILE, EINFLUSS, AUSBLICK UND VON GANS

4.1 HERHAUSFORDERUNGEN

Wie bei jeder anderen Technologie gibt es auch bei GANs einige Herausforderungen. Diese Probleme sind in der Regel mit dem Trainingsprozess verbunden, einschließlich des Zusammenbruchs des „Collapse Mode“ und Instabilität des Trainingsprozesses. Darüber hinaus wird die Auswertungstechnik, die Bildauflösung und die Ground Truth als weitere kontroverse Bereiche betrachtet. Eines der Hauptprobleme des Scheiterns beim GAN-Training ist der „Collapse Mode“. Dies bezieht sich auf den Zustand, in dem

der Generator beginnt, ähnliche Bilder zu erzeugen. Mit anderen Worten, die Vielfalt der generierten Samples ist auf verschiedene latente Räume beschränkt. Eine mögliche Lösung, um die Datenvielfalt zu erhöhen, ist die Verwendung einer Batch-Produktion von Samples anstelle der Generierung eines Samples. Ein anderer Ansatz besteht darin, mehrere Generatoren zu verwenden, um verschiedene Samples zu erhalten. In [43] wurden die generierten kombinatorischen Stichproben durch verschiedene Modelle untersucht, um das Problem des sogenannten „Collapse Mode“ aufzulösen. Die Verbesserung der Zielfunktion kann auch verwendet werden, um diese Herausforderung zu entschärfen, ähnlich wie bei den WGAN [44]-und weiteren GAN[45]-Modellen. Daher ist die Frage, wie die Diversität der generierten Stichproben erhöht werden sollte, eine entscheidende Frage, die in zukünftigen Arbeiten behandelt werden muss. "Instabilität des Trainingsprozesses" wird als eine weitere Herausforderung in diesem Bereich angesehen, die zu unterschiedlichen Ausgaben für ähnliche Eingaben führt. Obwohl die Batch-Normalisierung als eine Lösung für die GAN-Instabilität angesehen wird, ist sie nicht ausreichend, um die GAN-Leistung bei optimaler Stabilität zu verbessern. Es wurden zahlreiche Ansätze für ein nachhaltigeres Training vorgeschlagen [46], [47]. Nichtsdestotrotz sollten mehrere Lösungen vorgeschlagen werden, um ein stabileres GAN zu trainieren und auf das Nash-Gleichgewicht zu kommen. Das nächste Thema ist die GAN-Evaluation, ein komplexeres Thema als bei anderen generativen Modellen. In Abschnitt 2 wird eine Übersicht über mehrere derzeit häufig verwendete Evaluationsmetriken gegeben. Die Bereitstellung geeigneter, akzeptabler und umfassender Evaluierungsmethoden ist eine der wesentlichen Fragen, die weiter untersucht werden müssen. Eine weitere Einschränkung von adversen Netzwerken ist die Auflösung der generierten Bilder. Derzeit sind die meisten GAN-basierten Anwendungen für die Bildverarbeitung auf 256×256 begrenzt. Wenn dieses Netzwerk auf hochauflösende Bilder angewandt wird, werden in der Regel einige unscharfe Bilder erzeugt. Obwohl einige Forscher iterative Grob-zu-Fein-Methoden verwenden, um hochauflösende Bilder zu generieren, laufen diese nicht schnell. Chen und Koltun (2017) haben kaskadierte Verfeinerungsnetzwerke eingeführt, um eine Serie von 2-Megapixel-Bildern zu erzeugen, eine neue Perspektive für die Erzeugung hochauflösender Bilder [48]. Insgesamt ist es möglich, einen geeigneten Ansatz zur Verbesserung der Auflösung mit der Flexibilität der Bildgröße dank der hervorragenden Fähigkeiten der adversen Netzwerke anzubieten, und ein Bereich wird noch untersucht. Die Verwendung von Grundwahrheitsdaten für das Training ist eine weitere häufige Herausforderung. Das ist auch als ein entscheidendes Problem im Deep Learning bekannt. Diese Daten spielen eine wichtige Rolle bei der Synthese und Bearbeitung von GAN-basierten Bildern, da reale synthetisierte/bearbeitete Bilder nicht einfach gesammelt werden können. Die von CycleGAN[27] und Adversarial Inverse Graphics Networks (AIGNs) [49]

vorgeschlagenen Ansätze verwenden ungepaarte Daten für das Modelltraining. Diese Ansätze können als eine geeignete Lösung für ähnliche Probleme angesehen werden. Daher erfordert auch dieses Thema weitere Aufmerksamkeit, Untersuchung und Forschung für GAN-basierte Anwendungen.

4.2 VORTEILE VON GANS

GANs haben eine große Bedeutung für die Entwicklung von generativen Modellen. Als eine leistungsstarke Klasse generativer Methoden lösen GANs das Problem der Generierung von Daten, die natürlich interpretiert werden können. Insbesondere bei der Generierung von hochdimensionalen Daten ist die Generierungsdimension durch die verwendete neuronale Netzwerkstruktur nicht begrenzt, was den Umfang der generierten Datenmuster stark erweitert. Außerdem kann die Struktur des neuronalen Netzes verschiedene Verlustfunktionen integrieren, wodurch der Freiheitsgrad des Modelldesigns erhöht wird. Im Allgemeinen verwendet der Trainingsprozess von GANs zwei adversarial neuronale Netze als Trainingskriterium und kann durch Backpropagation trainiert werden. Das Training verlässt sich weder auf die ineffiziente Markov-Ketten-Methode noch auf approximative Inferenz. Es gibt keine komplexe Variationsuntergrenze, wodurch die Trainingsschwierigkeit erheblich reduziert und die Trainingseffizienz verbessert wird. Der Generierungsprozess von GANs erfordert keine langwierige Abtastsequenz, sondern kann direkt neue Proben abtasten und vorhersagen, was die Effizienz der Generierung neuer Proben verbessert. Das adversarialische Training verzichtet auf die direkte Replikation oder den Durchschnitt der realen Daten, wodurch die Diversität der generierten Samples erhöht wird. In der Praxis sind die von GANs generierten Samples für Menschen leicht verständlich. Zum Beispiel können GANs sehr scharfe und realistische Bilder erzeugen. Kurz gesagt, GANs bieten eine vielversprechende Lösung für die kreative Generierung von Daten, die für den Menschen aussagekräftig sind. GANs haben nicht nur einen großen Beitrag zur Entwicklung generativer Modelle geleistet, sondern sind auch für das semi-supervised Learning sinnvoll und lehrreich. Wie wir wissen, erfordert der Lernprozess von GANs keine Datenbeschriftungen, außer der Datenquelle. Obwohl das Ziel von GANs nicht das halbüberwachte Lernen ist, kann der Trainingsprozess von GANs verwendet werden, um ein Vortraining mit unmarkierten Daten durchzuführen. Zum Beispiel könnte zuerst eine große Menge an unbeschrifteten Daten verwendet werden, um GANs zu trainieren. Basierend auf dem Verständnis der trainierten GANs über die unbeschrifteten Daten können dann eine kleine Menge an beschrifteten Daten verwendet werden, um das diskriminative Modell für Klassifizierungs- und Regressionsaufgaben zu trainieren.

4.3. EINFLUSS UND AUSBLICK

GANs haben eine Menge Probleme für generative Modelle gelöst und andere KI-Methoden inspiriert, aber sie haben immer noch Einschränkungen. GANs verwenden die Idee

des adversen Lernens, aber die Konvergenz des Modells und die Existenz eines Gleichgewichtspunkts wurden noch nicht bewiesen. Der Trainingsprozess muss das Gleichgewicht und die Synchronisation von zwei adversen Netzwerken sicherstellen, da es sonst schwierig ist, gute Trainingsergebnisse zu erzielen. Es ist jedoch schwierig, die Synchronisierung der beiden adversen Netzwerke zu kontrollieren, sodass der Trainingsprozess instabil sein kann. Darüber hinaus haben GANs als generative Modelle, die auf neuronalen Netzen basieren, den üblichen Defekt (d. h. schlechte Interpretierbarkeit) von neuronalen Netzen. Und obwohl die von GANs generierten Muster vielfältig sind, gibt es das Problem des sogenannten „Collapse Mode“. „Collapse Mode“ bezieht sich auf Szenarien, in denen der Generator mehrere Bilder erzeugt, die dieselben Farb- oder Texturthemen enthalten und sich dadurch für das menschliche Verständnis kaum unterscheiden. Obwohl GANs einige Einschränkungen haben, ist es unbestreitbar, dass der Forschungsfortschritt von GANs ihre breiten Perspektiven offenbart hat. Neue Techniken, die sich der Reduzierung der Einschränkungen widmen, tauchen ständig auf. Beispielsweise überwindet der Wasserstein-GAN [41] das Problem der Trainingsinstabilität in hohem Maße und löst gleichzeitig teilweise das Problem des „Collapse Mode“. Wie man den „Collapse Mode“ vollständig vermeiden und den Trainingsprozess weiter optimieren kann, bleibt eine Forschungsrichtung von GANs. Darüber hinaus bleiben die Theorie der Modellkonvergenz und die Existenz des Gleichgewichtspunkts wichtige Forschungsthemen in der nahen Zukunft. Die oben genannten Forschungsrichtungen konzentrieren sich darauf, die Nachteile von GANs besser zu lösen. Aus der Perspektive der Entwicklung und Anwendung von GANs ist die Frage, wie eine Vielzahl von Daten, die mit Menschen interagieren können, aus einfachen zufälligen Eingaben generiert werden kann, eine wichtige Forschungsrichtung für die nahe Zukunft. Aus der Perspektive der Kombination von GANs und anderen Methoden ist die Integration von GANs mit Feature-Learning, Imitationslernen und Reinforcement-Learning sehr sinnvoll, um neue KI-Anwendungen zu entwickeln und die Entwicklung dieser Methoden zu fördern. Langfristig ist die Frage, wie GANs eingesetzt werden können, um die Entwicklung und Anwendung von KI zu fördern, die Fähigkeit von KI, die Welt zu verstehen, zu verbessern und sogar die Kreativität von KI zu stimulieren, ein wichtiges Problem, das von Forschern berücksichtigt werden sollte.

6. SCHLUSSFOLGERUNG

In den letzten Jahren wurden generative adversarialische Netzwerke (GANs) als einer der weit verbreiteten Deep-Learning-Algorithmen eingeführt und genutzt und wurden zu einer sehr beliebten Architektur für die Generierung hochrealistischer Inhalte. Diese Architektur versucht, Daten mit ähnlichen Eigenschaften wie die Eingabetrainingsdaten zu generieren, was dank seiner Resistenz gegen Überanpassung beim Lösen von Computer-Vision-Problemen die Aufmerksamkeit vieler Forscher erregt hat.

Da tiefgreifende neuronale Netze viele Daten zum Trainieren benötigen, haben sie eine schlechte Leistung, wenn die bereitgestellten Daten nicht ausreichend sind. GANs können dieses Problem überwinden, indem sie neuartige und realistische Daten erzeugen, ohne Tricks wie Datenvergrößerung zu verwenden. Auf diesem Gebiet wird eine Vielzahl wertvoller Forschungen und praktischer Anwendungen aktiv verfolgt. Zweifellos werden GANs mit einer Verbesserung der Netzwerkarchitekturen und Algorithmen in der Zukunft qualitativ hochwertige Bilder, Musikdateien, Filme und Texte erzeugen können, die für Menschen nur sehr schwer zu erstellen sind. Der Beitrag gibt einen Überblick über die wichtigsten Konzepte und die Theorie von GANs und zeigt Beispiele auf von GANs.

REFERENCES

1. K. P. Murphy, Machine learning: a probabilistic perspective. MIT press, 2012
2. Y. Bengio, A. Courville and P. Vincent, "Representation Learning: A Review and New Perspectives," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798-1828, Aug. 2013, d
3. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
4. Kunfeng Wang, Member, IEEE, Chao Gou, Yanjie Duan, Yilun Lin, Xinhua Zheng, and Fei-Yue Wang, Fellow, IEEE: "Generative Adversarial Networks: Introduction and Outlook", Oktober 2017, Retrieved February 1 2020 from <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8039016>
5. Ian J. Goodfellow, Jean Pouget-Abadie*, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair†, Aaron Courville, Yoshua Bengio: "Generative Adversarial Nets", 2014. Retrieved February 1 2020 from <https://arxiv.org/pdf/1406.2661.pdf>
6. Y. LeCun, 'RI Seminar: Yann LeCun : The Next Frontier in AI: Unsupervised Learning', 2016. [Online]. Available: <https://www.youtube.com/watch?v=IbjF5VjniVE> [Aufgerufen am 05.02.2021]
7. "Scopus database", 2019. [Online]. Abrufbar unter :www.scopus.com
8. Ian Godfellow "4.5 years of GAN progress on face generation." 15 Jan 2019, Retrieved February 1 2020 From https://twitter.com/goodfellow_ian/status/1084973596236144640?ref_src=twsrc%5Etfw%7Ctwcamp%5Etweetembed%7Ctwtterm%5E1084973596236144640%7Ctwgr%5E%7Ctwcon%5Es1_&ref_url=https%3A%2F%2Fmixed.de%2Fgeschichte-der-deepfakes-so-rasant-geht-es-mit-ki-fakes-voran%2F
9. Y.-J. Cao, L.-L. Jia, Y.-X. Chen, N. Lin, C. Yang, B. Zhang, Z. Liu, X.-X. Li, and H.-H. Dai, "Recent advances of generative adversarial networks in computer vision", *IEEE Access*, vol. 7, pp. 14985–15006, 2019 Retrieved February 1 2020 from <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8576508>
10. A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview", pp. 53–65, 2017, Retrieved February 1 2020 from <https://arxiv.org/pdf/1710.07035.pdf>
11. X. Wu, K. Xu, and P. Hall, "A survey of image synthesis and editing with generative adversarial networks", *Tsinghua Sci. Technol.*, vol. 22, no. 6, pp. 660–674, 2017. Retrieved February 1 2020 <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8195348>
12. T. Kaneko, "Generative adversarial networks: Foundations and applications", *Acoust. Sci. Technol.*, vol. 39, no. 3, pp. 189–197, 2018. Retrieved February 1 2020 https://www.jstage.jst.go.jp/article/ast/39/3/39_E183003/_pdf/-char/en
13. Y. Hong, U. Hwang, J. Yoo, and S. Yoon, "How generative adversarial networks and their variants work: An overview", *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–43, 2019. Retrieved February 1, 2020 <https://arxiv.org/pdf/1711.05914.pdf>
14. Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent progress on generative adversarial networks (GANs): A survey", *IEEE Access*, vol. 7, pp. 36322–36333, 2019. Retrieved February 1, 2020 <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8667290>
15. K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: introduction and outlook", *IEEE/CAA J. Autom. Sin.*, vol. 4, no. 4, pp. 588–598, 2017. Retrieved February 1, 2020
16. Jamie Beckett: „What’s a Generative Adversarial Network? Inventor Explains”, May 17, 2017 <https://blogs.nvidia.com/blog/2017/05/17/generative-adversarial-networks/>
17. K. Ganguly, "Learning Generative Adversarial Networks: Next-generation deep learning simplified", 2017

18. I. Goodfellow, “NIPS 2016 tutorial: Generative adversarial networks”, 2016, Retrieved February 1 2020, <https://arxiv.org/pdf/1701.00160.pdf>
19. I. Goodfellow, Y. Bengio, and A. Courville, “Deep learning,” MIT press, 2016.
20. F. CHOLLET, Deep Learning with Python. Manning Publications, 2017.
21. J. Brownlee, Generative Adversarial Networks with Python, Deep Learning Generative Models for Image Synthesis and Image Translation. 2019.
22. P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, ‘Image-to-image translation with conditional adversarial networks’, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017., Retrieved February 1, 2020 <https://arxiv.org/pdf/1611.07004.pdf>
23. O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation”, in International Conference on Medical image computing and computer-assisted intervention, pp. 234–241, 2015, Retrieved February 1, 2020 <https://arxiv.org/pdf/1505.04597.pdf>
24. C. Li and M. Wand, “Precomputed real-time texture synthesis with markovian generative adversarial networks”, in European conference on computer vision, pp. 702–716, 2016, Retrieved February 1, 2020, <https://arxiv.org/pdf/1604.04382.pdf>
25. P. Saleh and A. Chalechale, “Pix2Pix-based Stain-to-Stain Translation: A Solution for Robust Stain Normalization in Histopathology Images Analysis”, 2020., Retrieved February 1 <https://arxiv.org/ftp/arxiv/papers/2002/2002.00647.pdf>
26. T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-resolution image synthesis and semantic manipulation with conditional gans”, in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8798–8807, 2018. Retrieved February 1, <https://arxiv.org/pdf/1711.11585.pdf>
27. J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks”, in Proceedings of the IEEE international conference on computer vision, pp. 2223–2232, 2017. Retrieved February 1, 2020 <https://arxiv.org/pdf/1703.10593.pdf>
28. Kathuria Ayoosh, “Understanding GauGAN Part I: Unraveling Nvidia’s Landscape Painting GANs” Jan 19 2020, Retrieved February 1, 2020 <https://blog.paperspace.com/nvidia-gaugan-introduction/>
29. C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, and Z. Wang, “Photo-realistic single image super-resolution using a generative adversarial network”, in Proceedings of the IEEE conference on computer vision, Retrieved February 1 2020 <https://arxiv.org/pdf/1609.04802.pdf>
30. X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, ‘Esrgan: Enhanced super-resolution generative adversarial networks’, in Proceedings of the European Conference on Computer Vision Workshops (ECCVW), 2018. Retrieved February 1 2020 <https://arxiv.org/pdf/1809.00219.pdf>
31. A. Jolicœur-Martineau, “The relativistic discriminator: a key element missing from standard GAN”, 2018 Retrieved February 1, <https://arxiv.org/pdf/1807.00734.pdf>
32. R. Huang, S. Zhang, T. Li, and R. He, “Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis”, in Proceedings of the IEEE International Conference on Computer Vision, pp. 2439–2448, 2017, Retrieved February 1, 2020 <https://arxiv.org/pdf/1704.04086.pdf>
33. H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks”, in The 36th International Conference on Machine Learning (ICML), pp. 7354–7363, 2019. Retrieved February 1 2020, <https://arxiv.org/pdf/1805.08318.pdf>
34. A. Brock, J. Donahue, and K. Simonyan, “Large scale gan training for high fidelity natural image synthesis”, Int. Conf. Learn. Represent., 2019., Retrieved February 1 2020 <https://arxiv.org/pdf/1809.11096.pdf>
35. L. Tran, X. Yin, and X. Liu, “Disentangled representation learning gan for pose-invariant face recognition”, in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1415–1424, 2017. Retrieved February 1 2020, https://openaccess.thecvf.com/content_cvpr_2017/papers/Tran_Disentangled_Representation_Learning_CVPR_2017_paper.pdf
36. X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, “Towards large-pose face frontalization in the wild”, in Proceedings of the IEEE International Conference on Computer Vision, pp. 3990–3999, 2017, Retrieved February 1 2020, <https://arxiv.org/pdf/1704.06244.pdf>
37. V. Blanz and T. Vetter, “A morphable model for the synthesis of 3D faces”, in Proceedings of the

- 26th annual conference on Computer graphics and interactive techniques, pp. 187–194, 1999
38. D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting”, 2016. Retrieved February 1, 2020, <https://arxiv.org/pdf/1604.07379.pdf>
 39. C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, “High-resolution image inpainting using multi-scale neural patch synthesis”, Retrieved February 1 2020, <https://arxiv.org/pdf/1611.09969.pdf>
 40. R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, “Semantic image inpainting with deep generative models”, in Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2017, Retrieved February 1 2020, <https://arxiv.org/pdf/1607.07539.pdf>
 41. S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Globally and locally consistent image completion”, ACM Trans. Graph., vol. 36, no. 4, pp. 1–14, 2017
 42. F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions”, in International Conference on Learning Representations (ICLR), pp. 1–13, 2016, Retrieved February 1 2020, <https://arxiv.org/pdf/1511.07122.pdf>
 43. A. Ghosh, V. Kulharia, V. P. Namboodiri, P. H. S. Torr, and P. K. Dokania, “Multi-agent diverse generative adversarial networks”, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018., Retrieved February 1 2020, <https://arxiv.org/pdf/1704.02906.pdf>
 44. M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein Generative Adversarial Networks”, in Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 2017, Retrieved February 1 2020 <https://arxiv.org/pdf/1701.07875.pdf>
 45. L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, “Unrolled generative adversarial networks”, in proceedings international conference on learning representations, pp. 1–25, 2017, Retrieved February 1 2020, <https://arxiv.org/pdf/1611.02163.pdf>
 46. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans”, in Advances in neural information processing systems, pp. 2234–2242, 2016, Retrieved February 1, 2020 <https://arxiv.org/pdf/1606.03498.pdf>
 47. M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium”, in Advances in neural information processing systems, 2018, Retrieved 1 February 2020, <https://arxiv.org/pdf/1706.08500.pdf>
 48. Q. Chen and V. Koltun, “Photographic image synthesis with cascaded refinement networks”, in Proceedings of the IEEE international conference on computer vision, pp. 1511–1520, 2017, Retrieved February 1, 2020 <https://arxiv.org/pdf/1707.09405.pdf>
 49. H.-Y. F. Tung, A. W. Harley, W. Seto, and K. Fragkiadaki, “Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision”, in The IEEE International Conference on Computer Vision (ICCV), pp. 4364–4372, 2017. Retrieved February 1, 2020 <https://arxiv.org/pdf/1705.11166.pdf>