A decorative graphic on the left side of the slide consists of two overlapping parallelograms. The front one is blue and the back one is a light green. They are positioned diagonally, with the blue one partially covering the green one.

Generative Adversarial Networks

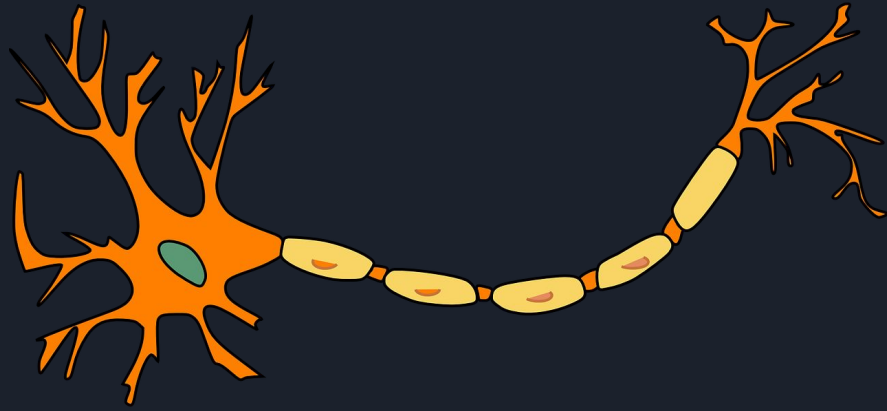
von Mirco Heck

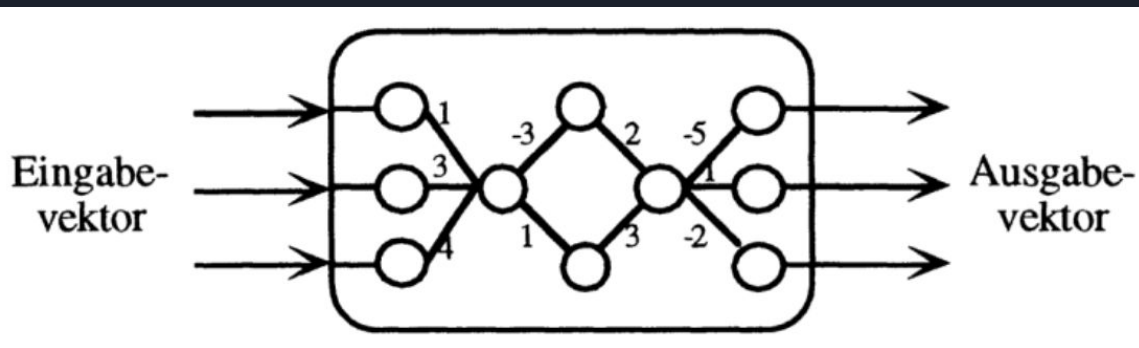
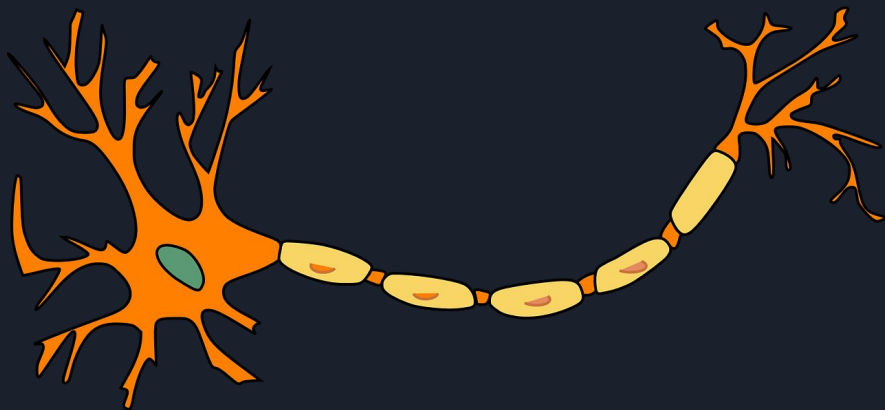


Agenda

1. Geschichte
2. Prinzip
3. Training
4. Wichtige Typen
5. Praxisbeispiele
6. Quellen





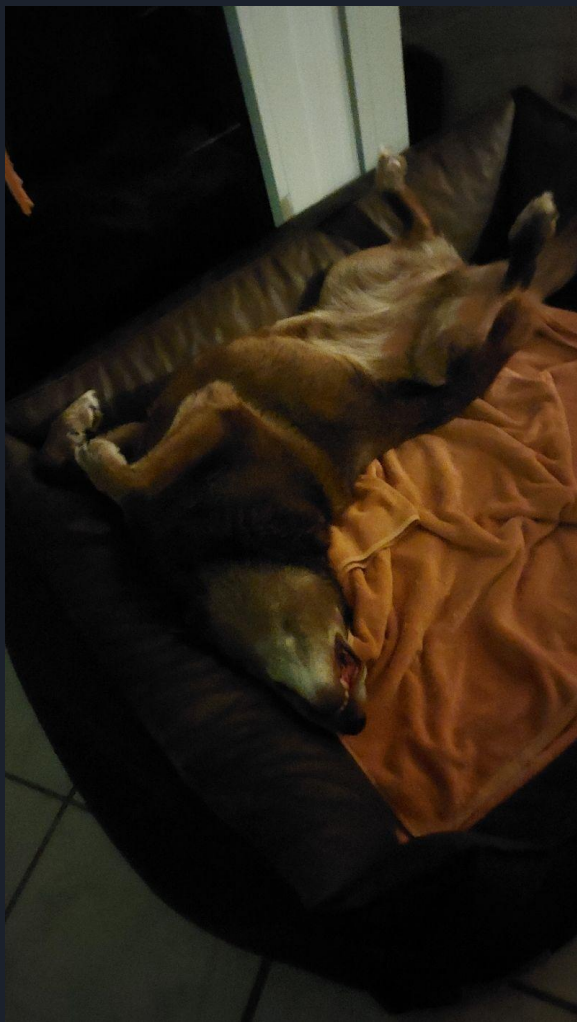



$$8 + x = 25$$

$$15 + x = 10$$

$$0 + x = -5$$



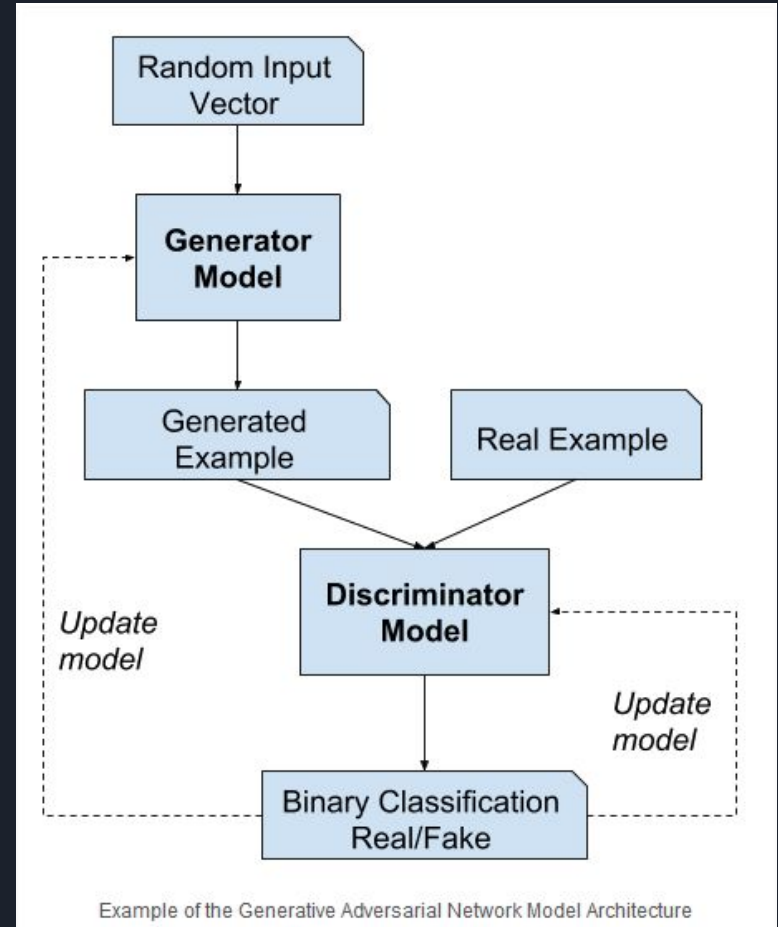




- Sehr neue Technologie
- Vorschlag der Verwendung kompetitiver neuronaler Netzwerke in 2013
- Vorschlag der GAN Architektur in 2014
- Vortrieb der Technologie durch NVIDIA

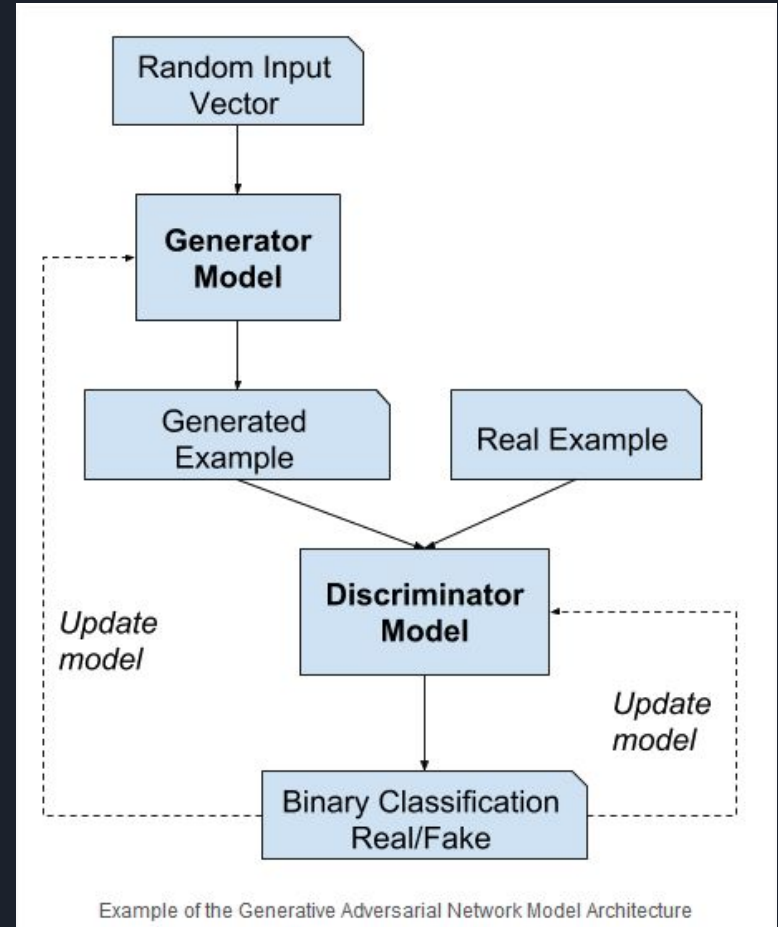
Prinzip

- **G**enerative **A**dversarial **N**etworks (Generative gegnerische Netze)
- Zwei Modelle, die sich gegenseitig trainieren
- Generator und Diskriminator
- Unüberwachtes Lernen
 - Machine Learning ohne vorher bekannte Zielwerte und ohne Belohnungen
 - Rechenintensiv und benötigt riesige Datensätze



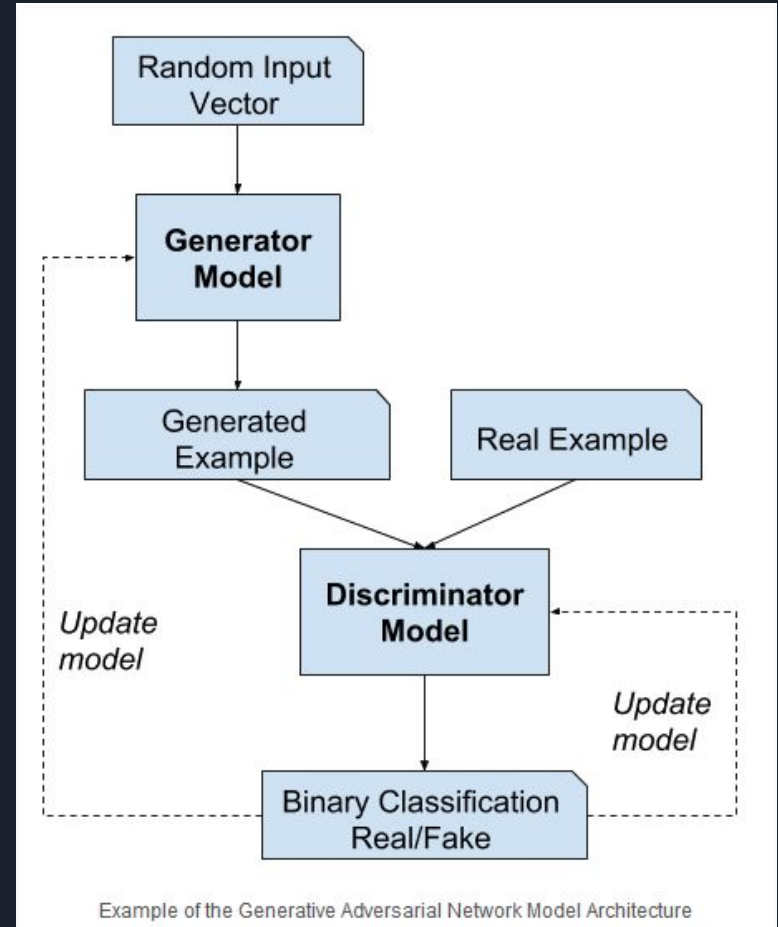
Generator

- Eingang: Zufälliges Rauschen(Latent Space)
- Lernt aus Rauschen ein Bild, Song oder Text zu erstellen
- Bekommt Feedback vom Diskriminator(Backpropagation)
- Ziel: Daten generieren die möglichst natürlich wirken



Diskriminator

- Eingang: Entweder vom Generator erzeugte Daten **ODER** Trainingsdaten/echte Daten
- Bestimmt Wahrscheinlichkeit, dass das Bild echt ist
- Binäre Klassifikation entscheidet, ob Diskriminator richtig liegt und gibt Feedback an beide Modelle zurück(Backpropagation)
- Lernt korrekt zu erkennen, ob die Daten künstlich erzeugt wurden oder echt sind
- Ziel: Möglichst präzise den Ursprung der Daten bestimmen können







VS





VS



VS





VS



VS



VS





Training

- Generator und Diskriminator werden immer abwechselnd trainiert
 - Während Training muss das jeweils andere Modell konstant bleiben
 - Sonst Risiko für fehlende Konvergenz wegen sich laufend ändernden Voraussetzungen
 - Diskriminator benötigt anfangs “Vorsprung”, da sonst der Trainingsprozess nicht starten kann
- Hin und Her erlaubt durch Selbstjustierung das Lösen von Problemen, an denen herkömmliche Lösungsansätze scheitern



Training

Konvergenz

- Je besser der Generator wird, desto schlechter wird der Diskriminator
- Bei perfektem Generator wäre Diskriminator nur noch max. 50% akkurat("Münzwurf")
 - Bei zu gutem Generator wird das Feedback des Diskriminators nutzlos
- Schlechter Diskriminator färbt zwangsmäßig auf Generator ab aufgrund von Backpropagation
 - Zu guter Generator kann zu schlechtem Generator führen
 - **Failure to converge**
- Großes Problem bei Training von GANs
 - Zeigt die Zerbrechlichkeit des GAN-Trainingsprozess



Wichtige Typen

Variationen der Architektur:

-Conditional GAN

- Zusätzliche Daten-Labels als Input für Generator und Diskriminator
- Bietet späteren Einfluss auf Parameter der generierten Daten
- z.B. kann Motiv eines generierten Bildes beeinflusst werden

-Deep Convolutional GAN

- Nutzung von Convolutional Neural Networks als Modelle
- CNNs bieten Convolutional Layer, die Bezug zwischen benachbarten Datenpunkten erlauben
- z.B. kann für Bilddaten benachbarte Pixel mit einbezogen werden
- nützlich für visuelle Einsatzzwecke

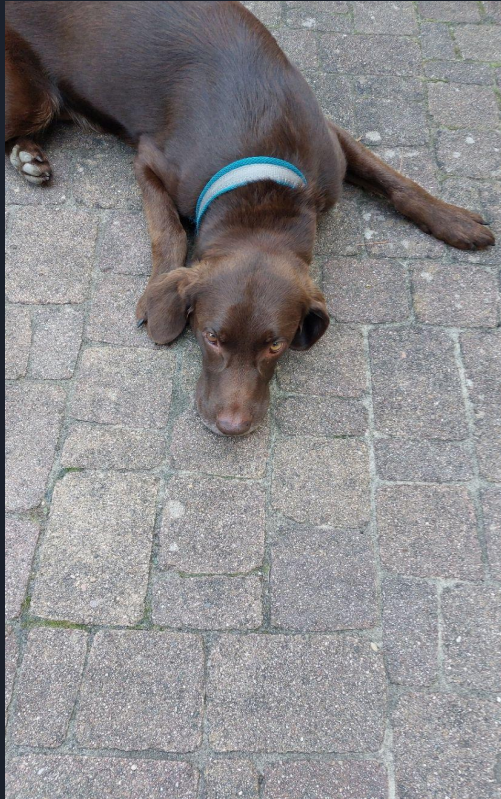


Beispiele

BigGAN(Artbreeder)

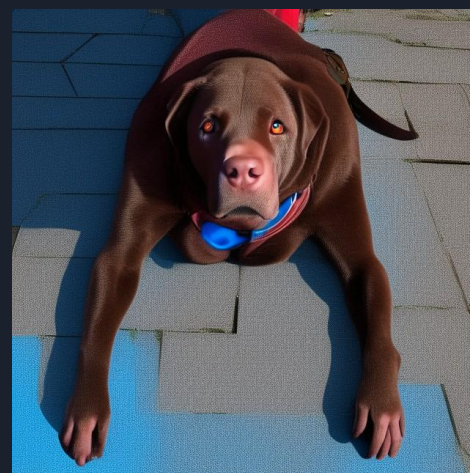
- GAN für die Bildsynthese mit Fokus auf das Erzielen hoher Bildauflösungen
- Conditional GAN auf Basis von 150 Gigabyte gelabelte Daten
- Findet Einsatz u.a. in bekannten Online-Tool *Artbreeder*(ehemals *GANbreeder*)
- Artbreeder bietet eine große Anzahl an Funktionen für die KI-unterstützte Bildbearbeitung, wie:
 - **Mixer**(mischt Bilder zusammen)
 - **Tuner**(erlaubt das Anpassen von Bildparametern)
 - **Pattern maker**(erlaubt das Einbetten von Formen in andere Bilder)
 - **Outpainter**(vergrößert den sichtbaren Ausschnitt eines Bilds)

Mixer:





Mixer:



Tuner:



Pattern maker:



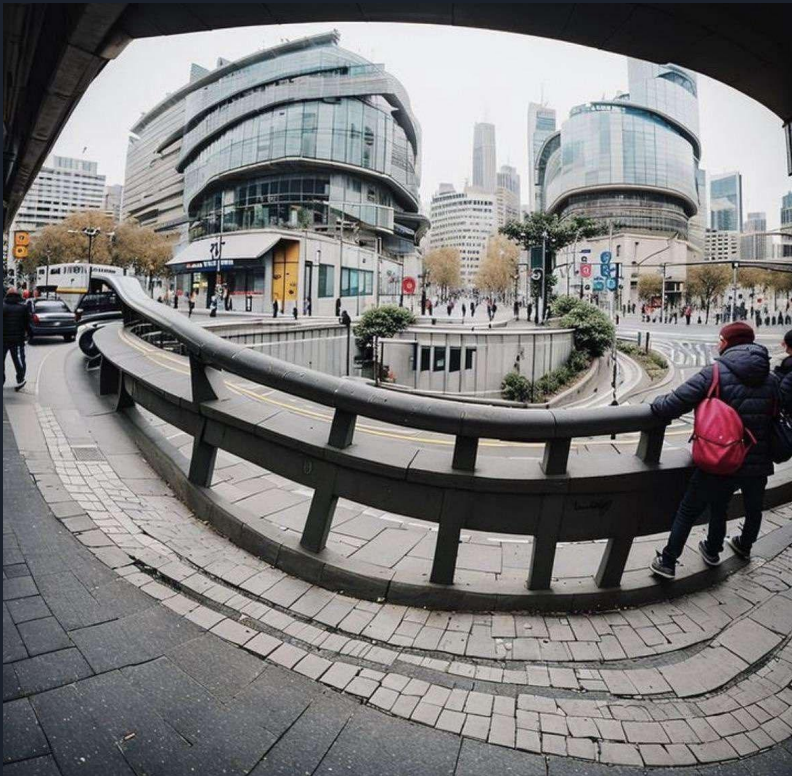
Pattern maker:



Pattern maker:



Pattern maker:



Pattern maker:



Pattern maker:





Beispiele

StyleGAN(this...doesnotexist.com)

- GAN für die Bildsynthese, besonders für die Synthese menschlicher Gesichter
- Wurde 2018 von Nvidia entwickelt(seit 2019 Open-Source)
- Funktioniert nur mit proprietären Nvidia-Produkten CUDA und Nvidia-GPUs
- StyleGAN 2 wurde 2020 veröffentlicht
- StyleGAN 2 wird genutzt für die Projektwebsite thispersondoesnotexist.com
- Website generiert bei jedem Aufruf ein photorealistisches Bild eines menschlichen Gesichts



This person does not exist

This person does not exist



This person does not exist



This person does not exist





Quellen

Inhalt:

- <https://developers.google.com/machine-learning/gan/problems>
- <https://books.google.de/books?hl=de&lr=&id=3irVBgAAQBAJ&oi=fnd&pg=PA2>
- <https://books.google.de/books?hl=de&lr=&id=b06qDwAAQBAJ&oi=fnd&pg=PP9>

Bilder:

- <https://ih1.redbubble.net/image.3640920496.6320/raf.360x360,075,t,fafafa:ca443f4786.jpg>
- <https://www.artbreeder.com>
- <https://thispersondoesnotexist.com/>
- https://static.wikia.nocookie.net/love-exalted/images/e/e1/Donkey_kong.png
- https://upload.wikimedia.org/wikipedia/en/9/9a/Trollface_non-free.png
- <https://www.goalcast.com/wp-content/uploads/2021/11/Untitled-design-48.jpg>

Vielen Dank für ihre
Aufmerksamkeit!