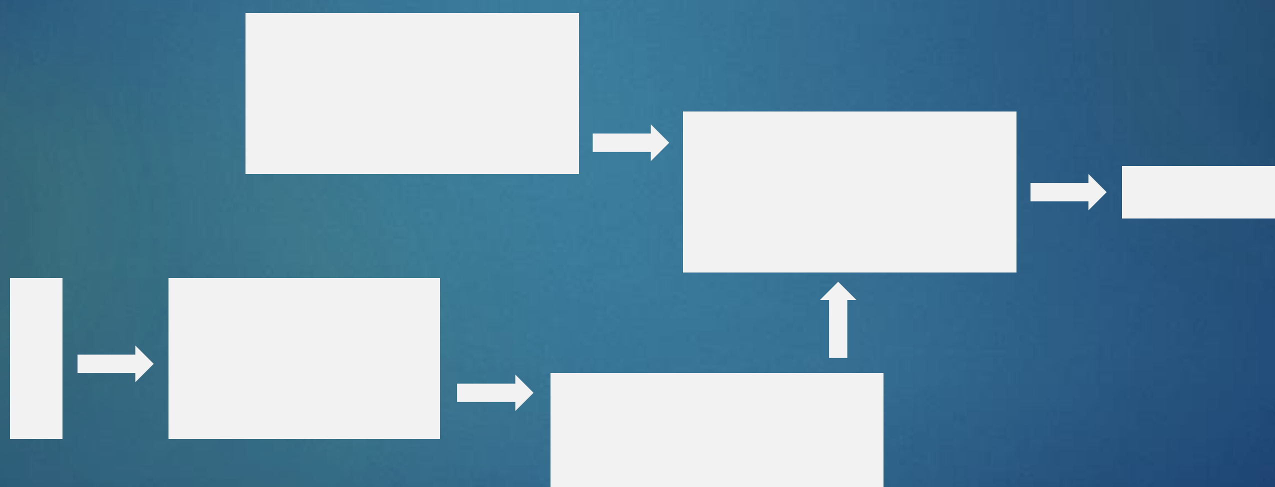
alitourani

Deep Learning from Scratch

Session #8: Generative Adversarial Networks



by: Ali Tourani – Summer 2021

Agenda

- Supervised vs. Unsupervised Learning
- Generative Modeling
- Latent Variable Models
- Autoencoders
- Variational Autoencoders
- Generative Adversarial Networks

Supervised vs. Unsupervised Learning

Supervised Learning

- Learning which takes place in the presence of a teacher!
- The algorithm learns from labeled training data
  - Refers to the observed labels (classes) when faces unforeseen data
- **Challenges?**
  - The tough process of building models
  - Needs technical data science expertise
  - Labeling many data is a disaster!

Regression **Classification**

Image source: https://www.pngitem.com

Supervised vs. Unsupervised Learning

Unsupervised Learning

- Let’s allow the model to work on its own and discover information!
- We do not need labeled data anymore!
  - Can find all kinds of unknown patterns in data
- **Challenges?**
  - More complex processing tasks
  - Easier data acquisition
  - Less limitations in building models

Association **Clustering**

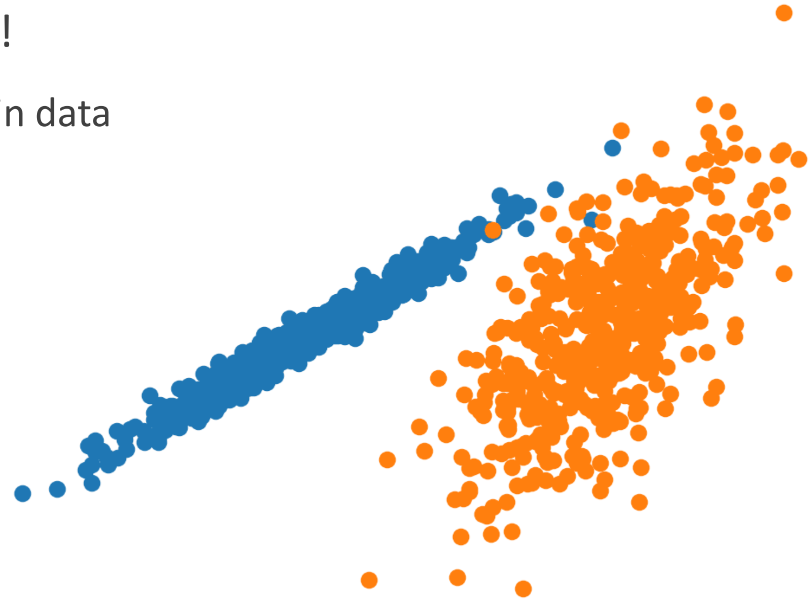
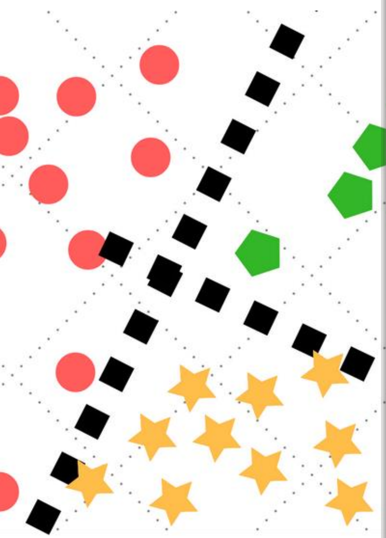


Image source: https://www.machinelearningmastery.com

Supervised vs. Unsupervised Learning

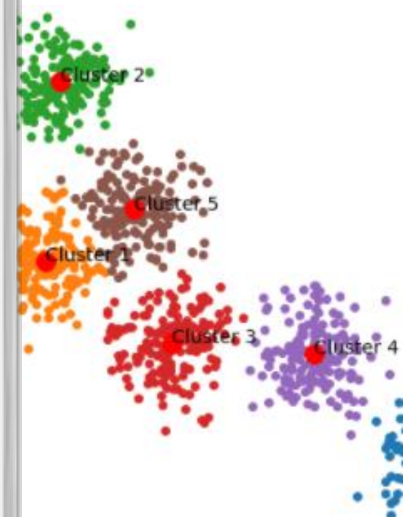


Supervised Leaning

- Access to the pairs of *(data, label)*
- Trying to find a way to map *data* to *label*
- Use cases in regression, classification, object detection, etc.

Unsupervised Leaning

- Access to only *data* and no *label*
- Trying to learn the underlying structures of data
- Use cases in clustering, feature reduction, etc.



Supervised vs. Unsupervised Learning

But, why is unsupervised learning important?

- In many cases, we do not know exactly what we are looking for! We just want to find patterns, no matters how!
- For instance, in the Cybersecurity applications

Supervised approach

Attacks may miss, because the machine has not seen it before

Unsupervised approach

The machine tries to detect any abnormal actions

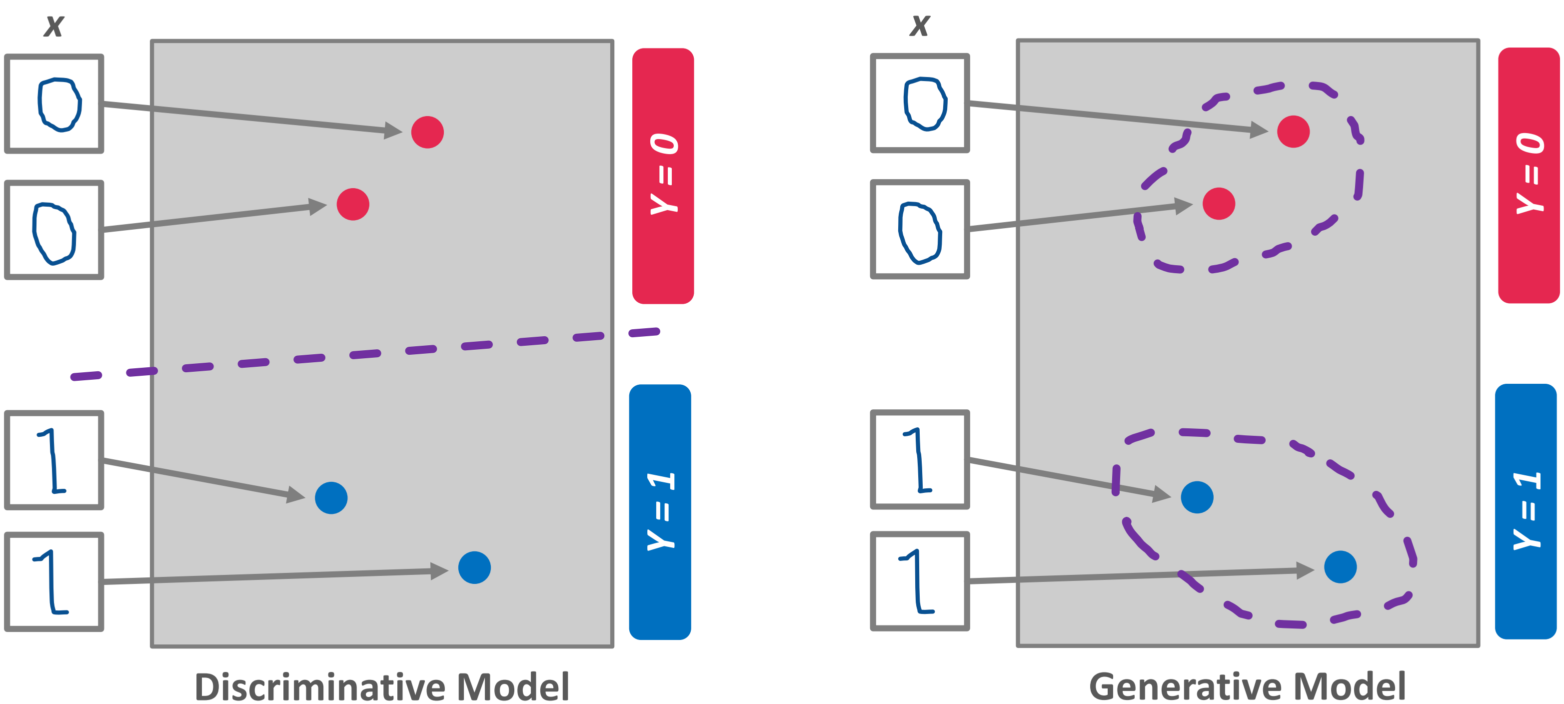
Image source: https://www.paralleltech.com

Generative Modeling

- Can we build models that can generate new samples of data?
  - **Answer**: Yes! By using **Generative Models**.
- Generative Modeling (GM):
  - Contrasts with discriminative modeling
  - Is the use of AI and statistics to produce representations of observed data
    - **Goal: Representing the distribution of the training data**
  - Is used in unsupervised approaches to help machines predict any probabilities
  - Utilizes reduced understandings of data for modeling
  - **Use case:** Models that predict the next word in a sequence

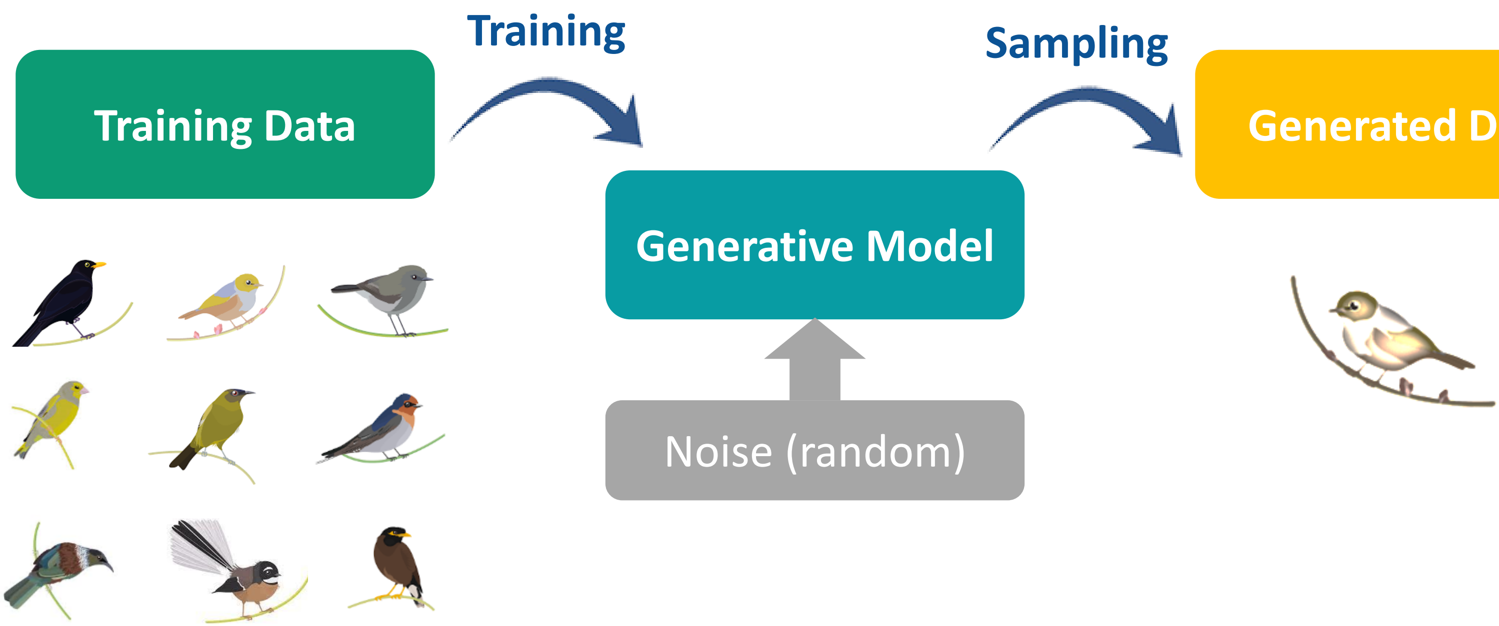
Generative Modeling

Generative vs. Discriminative modeling



Trying to distinguish 1's and 0's by generating samples that fall close to their real values

Generative Modeling



Observations

Image source: https://www.catalogue.data.govt.nz

Generative Modeling

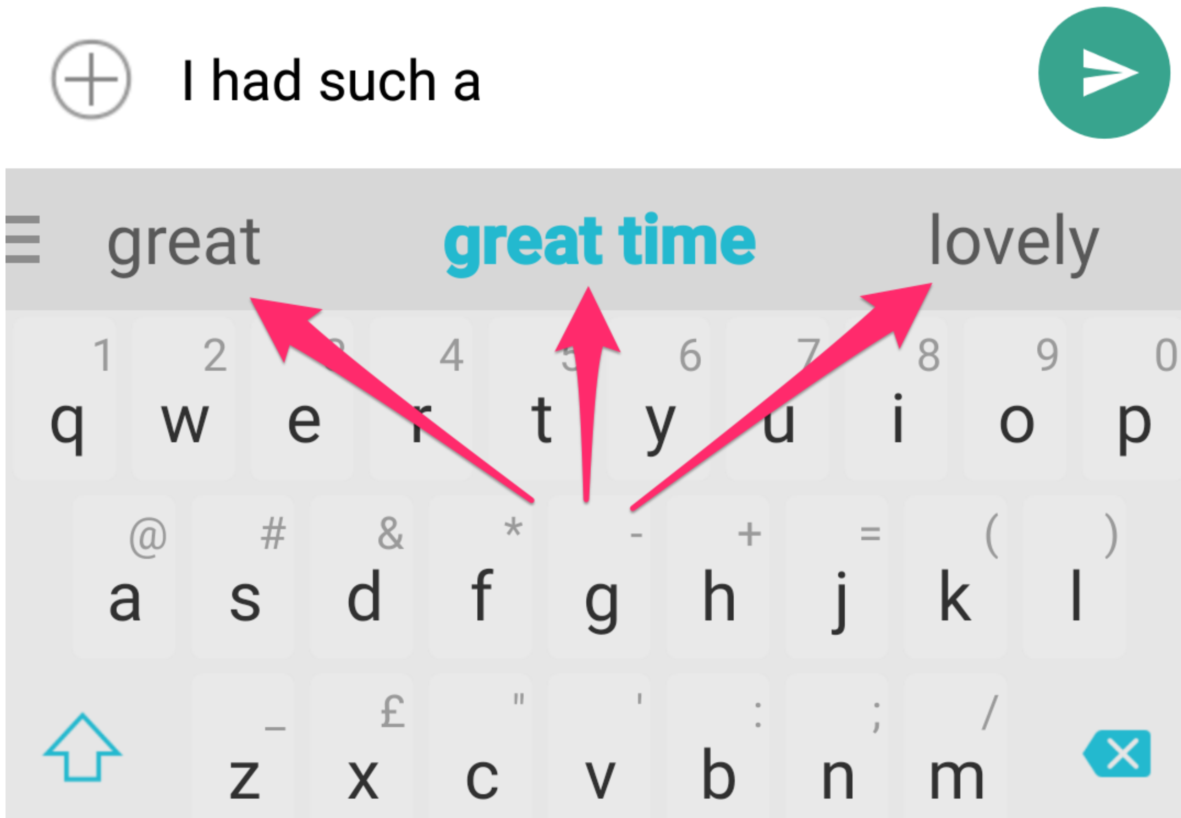
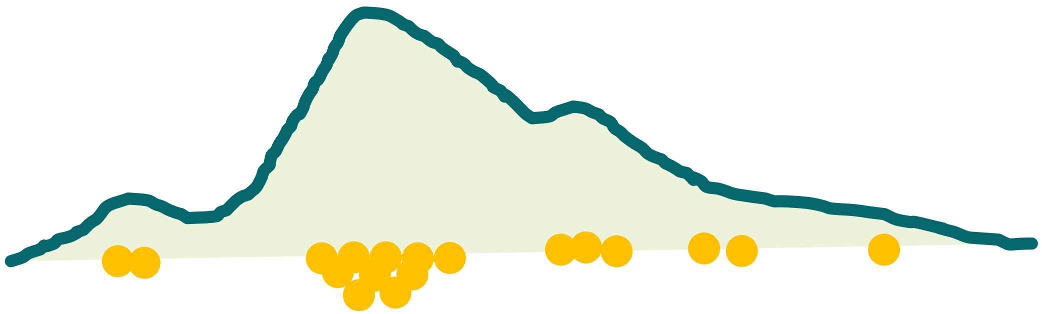


Image source: https://www.blog.jovian.ai

Generative Modeling

How to achieve this?

- Using the **features** extracted from training data
- The model should try to generate new sets of features
- Generated samples should follow the same (or similar) rules that created the training instances
- Building a model that **mimics the probabilistic distribution** in the training dataset



Generative Modeling

GMs can uncover underlying features in a dataset to create new items

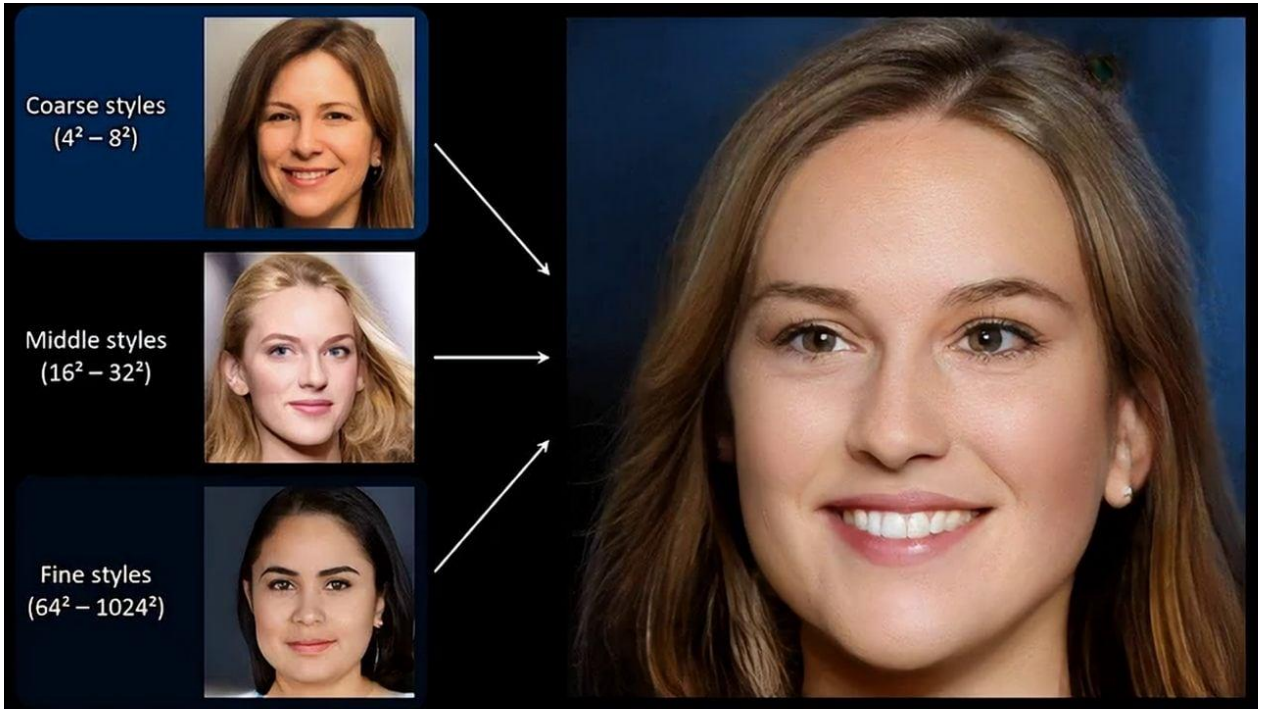


Image source: https://images.firstpost.com/

Generative Modeling

GMs can uncover underlying features in a dataset to create new items

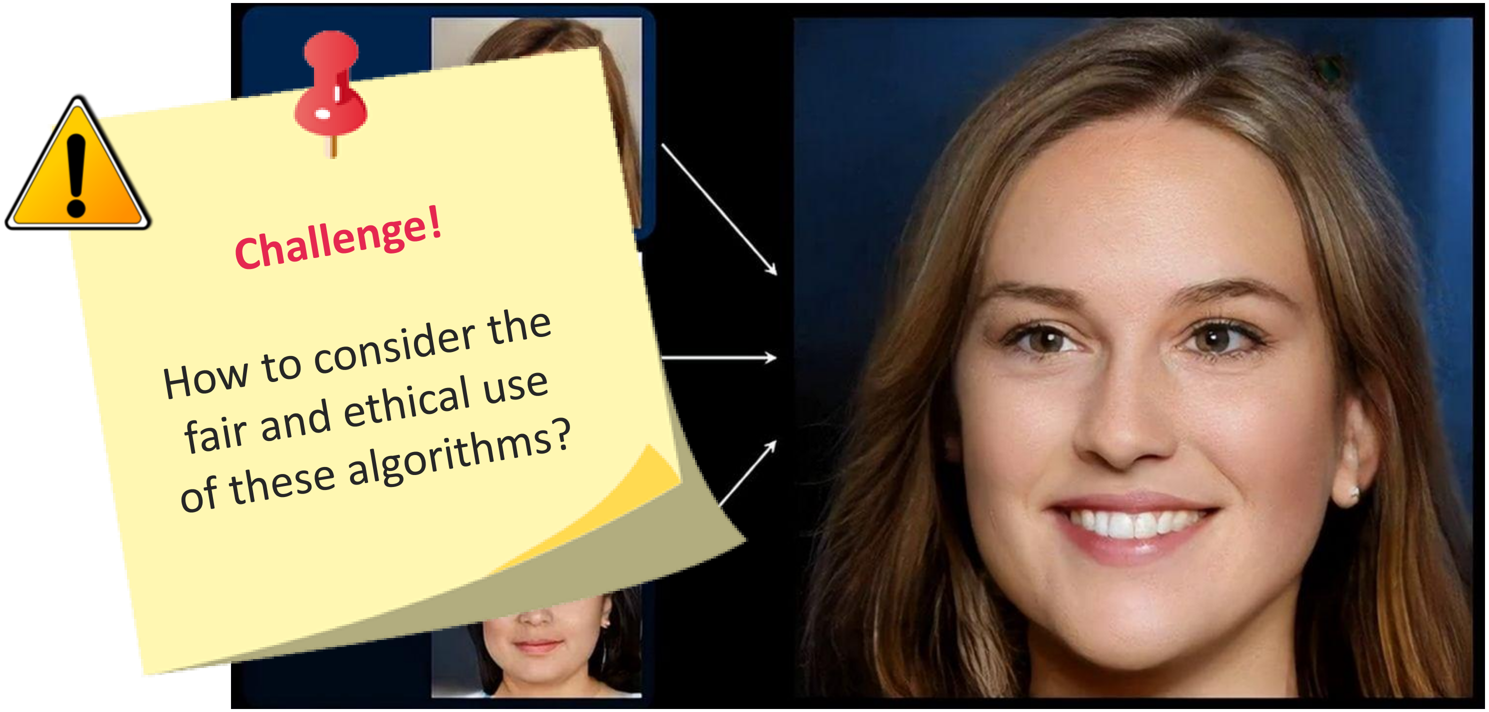
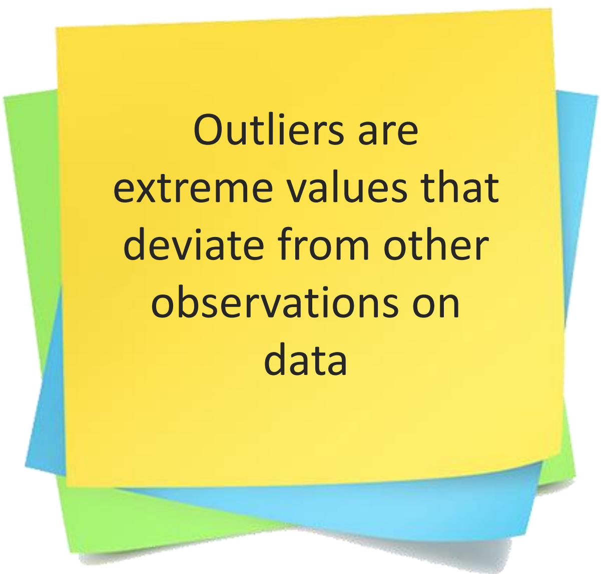


Image source: https://images.firstpost.com/

Generative Modeling

Outlier/anomaly detection using GMs

- Can also be used for outlier (anomaly) detection



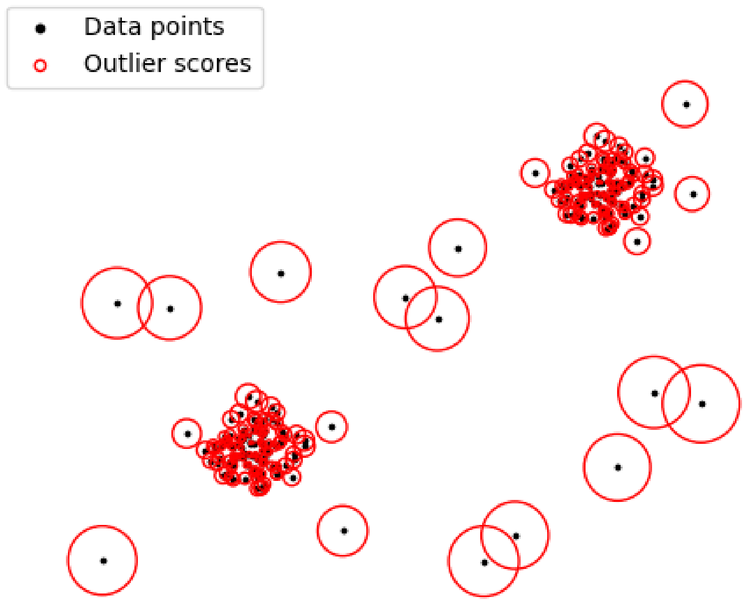


Image source: https://www.scikit-learn.org

Generative Modeling

Outlier/anomaly detection using GMs

- The goal is to detect rare or unseen items
- How to do that?
  - **Approach#1:** Leveraging generative models to detect outliers
  - **Approach#2:** Using outliers during training to improve accuracy

What we have trained

Some outliers with different features

Latent Variable Models

What are Latent Variables?

- Variables that cannot be observed
- They are detectable by their effects on observable data
- They can be modeled to be observed
- In Machine Learning:

A Latent Variable Model (LVM) is a probability distribution over x (observed at learning time in the dataset) and y (unseen data) variables

$$p(x, y)$$

Latent Variable Models

Why are LVMs important?

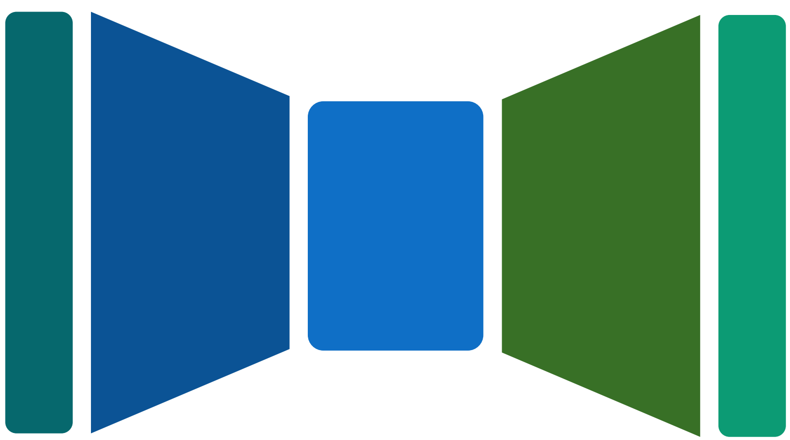
- Some data might be naturally unobserved (outliers)
- They enable us to leverage our prior knowledge when defining a model
- Using LVMs, we can learn the Explanatory Factors from observed data

We are trying to describe the animal using its shadow, while its physical appearances is not clear in this image

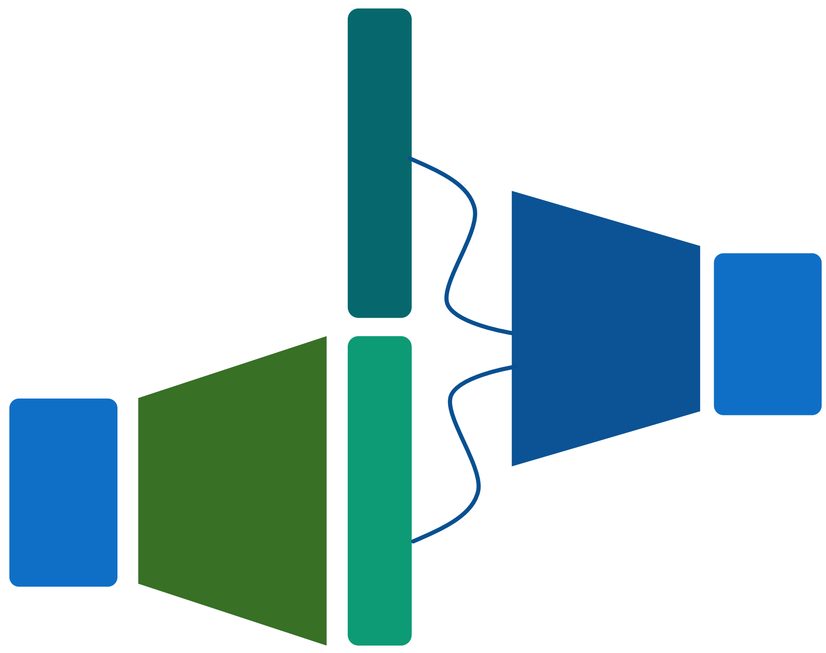
Image source: https://www.mymodernmet.com

Latent Variable Models

LVMs and different types of ANNs



Autoencoders (AEs)
Variational AEs (VAEs)



Generative Adversarial Networks (GANs)

Autoencoders

- A DNN used for Feature (representation) Learning in unlabeled data
  - Discovering the representations needed for feature detection
  - Learning a lower-dimensional feature representation
- Attempting to regenerate the input from its representation (encoding)
  - Ignoring the noise while training

Contains two modules:

- Encoder: maps raw data into vectors of LVs
- Decoder: reconstructs the observation

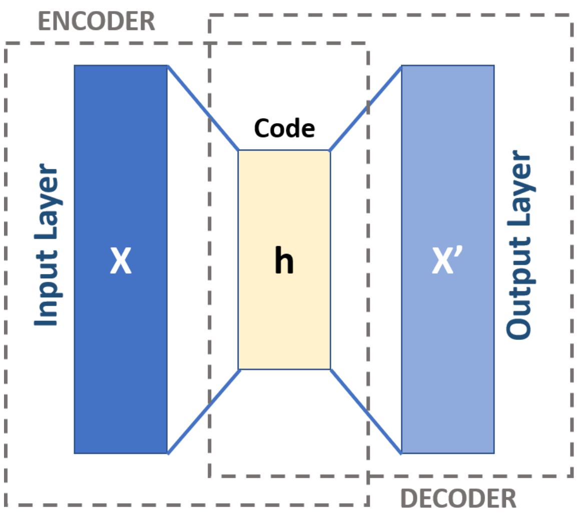
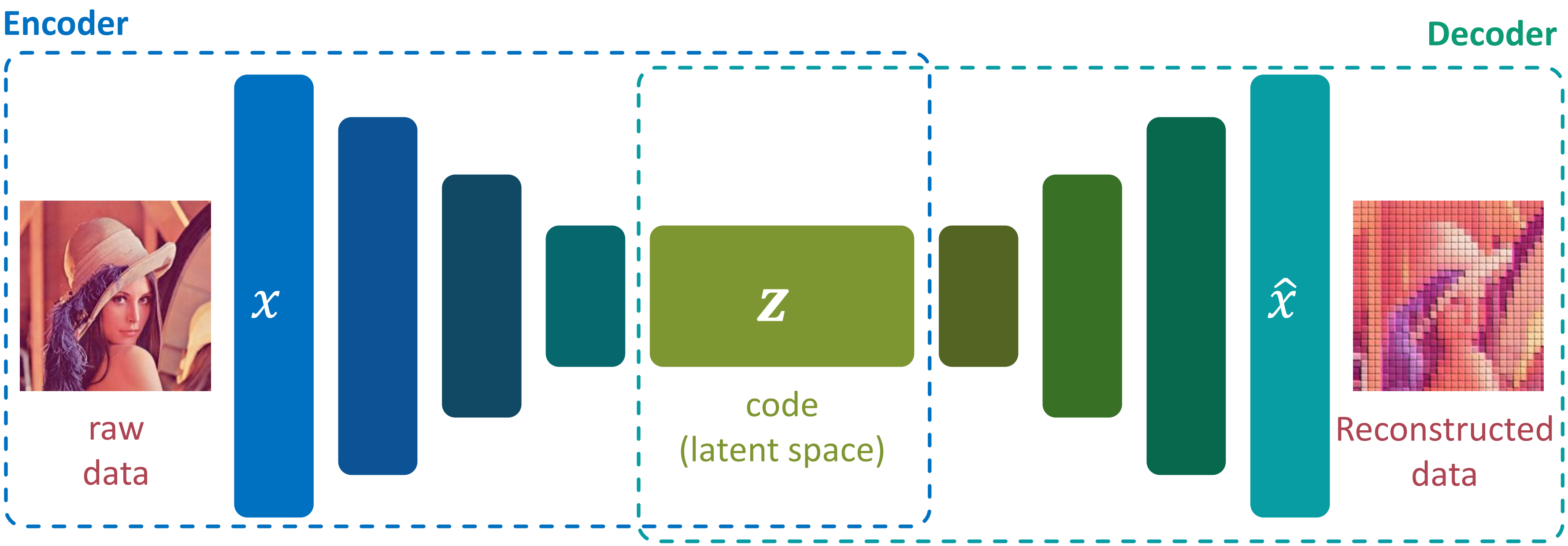


Image source: https://www.wikipedia.org

Autoencoders

The architecture of an autoencoder



Autoencoders

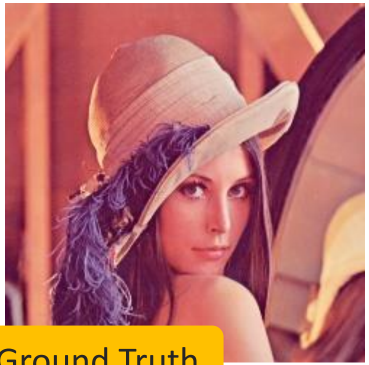
Important Notes on AEs

- The goal in an AE is to leverage NNs for representation learning
  - Having a low dimensional latent space (compression)
- AEs reconstruct the input **approximately** by keeping only the most relevant parts of data
- The code is actually a compressed knowledge representation
- Training process in AEs is a bit different!
  - The model is trained to use features to reconstruct the original data
  - Trying to minimize the difference between the input and reconstructed data
    - There are various approaches for this, like Mean Square Error (MSE)

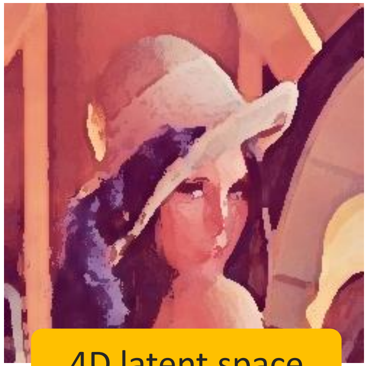
Autoencoders

Important Notes on AEs

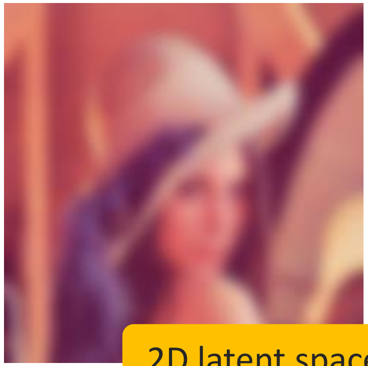
- In fact, AEs present a form of compression
  1. Compressing the input into much smaller latent space
  2. Building the input back (reconstruction)
- Lower dimensionality latent space ➔ Poorer reconstruction output



Ground Truth



4D latent space



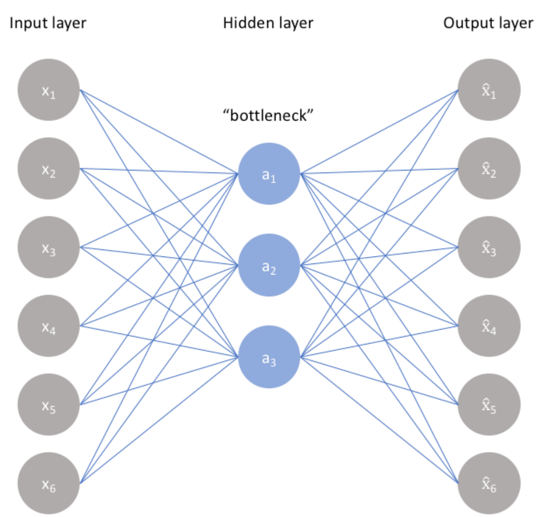
2D latent space

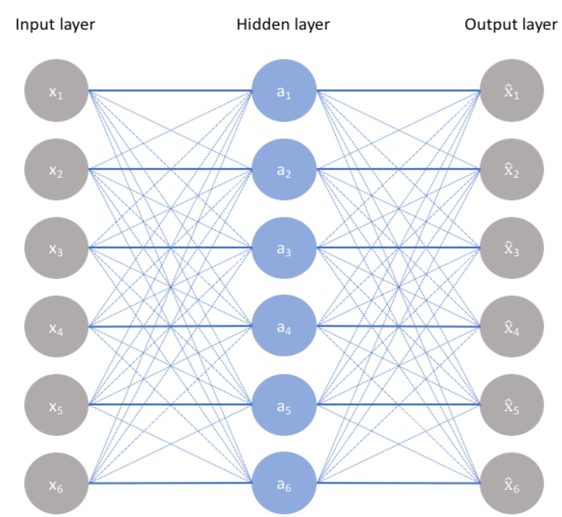
Autoencoders

Important Notes on AEs

- Bottleneck in AEs prevents the network from memorizing the input values instead of learning the compression of the input data

With Bottleneck





Without Bottleneck

Image source: https://www.jeremyjordan.me/autoencoders/

Autoencoders

Applications of AEs – Image Reconstruction

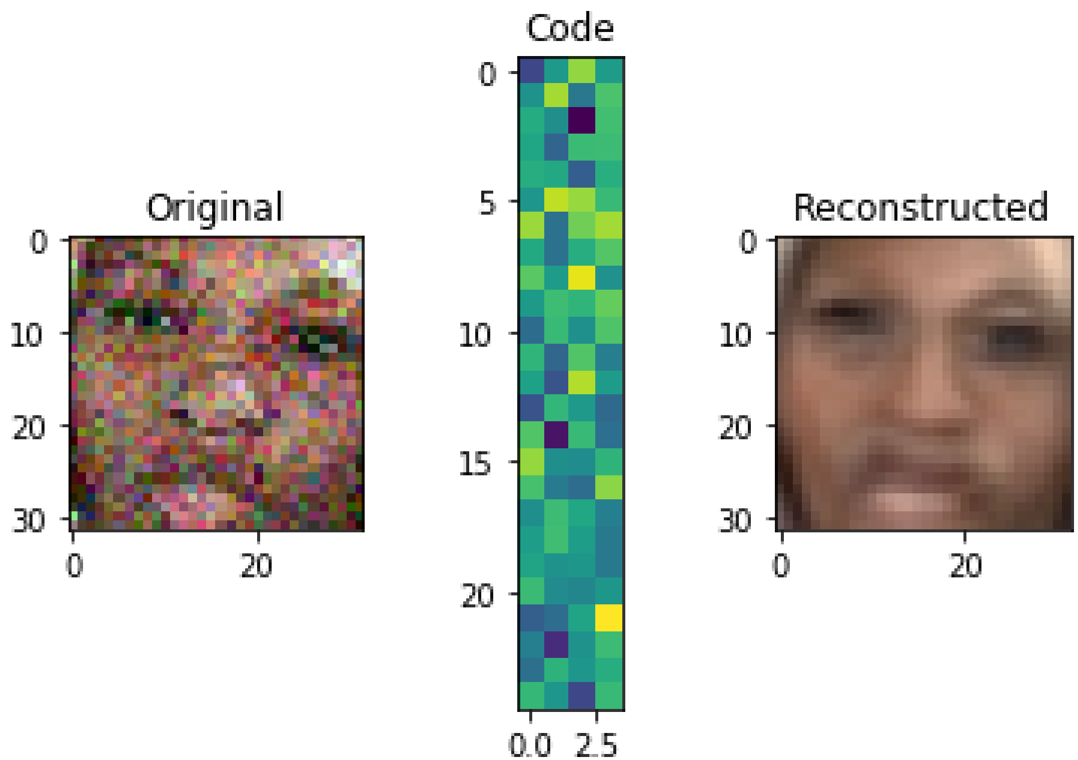


Image source: https://www.stackabus.com

Autoencoders

Applications of AEs – Image Compression

Image source: https://www.nytimes.com

Autoencoders

Applications of AEs – Dimensionality Reduction

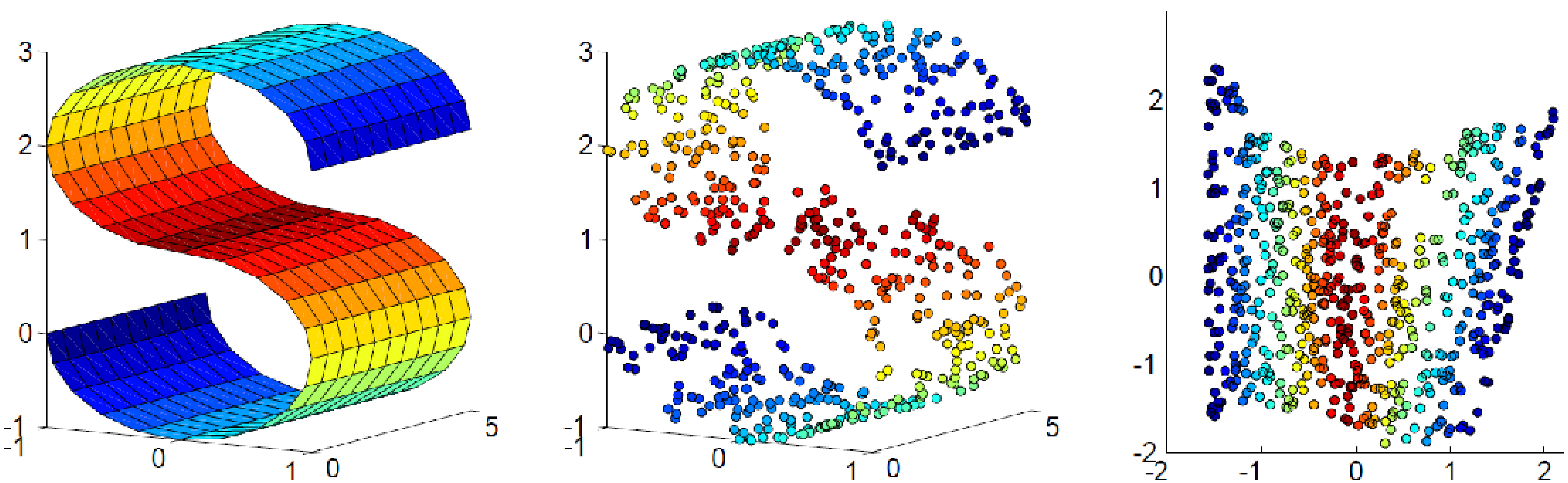


Image source: https://www.semanticscholar.org

Variational Autoencoders

- Unlike the traditional AEs, VAEs contain a variational twist for coding
  - A latent space with a mixture of distributions, instead of a fixed vector
  - Instead of direct learning of the latent variables, learns **Mean & Variance**
- A different mathematical formulation
- **Goal:** generating higher quality representations and samples
- Loss function in a VAE is a bit more complex:

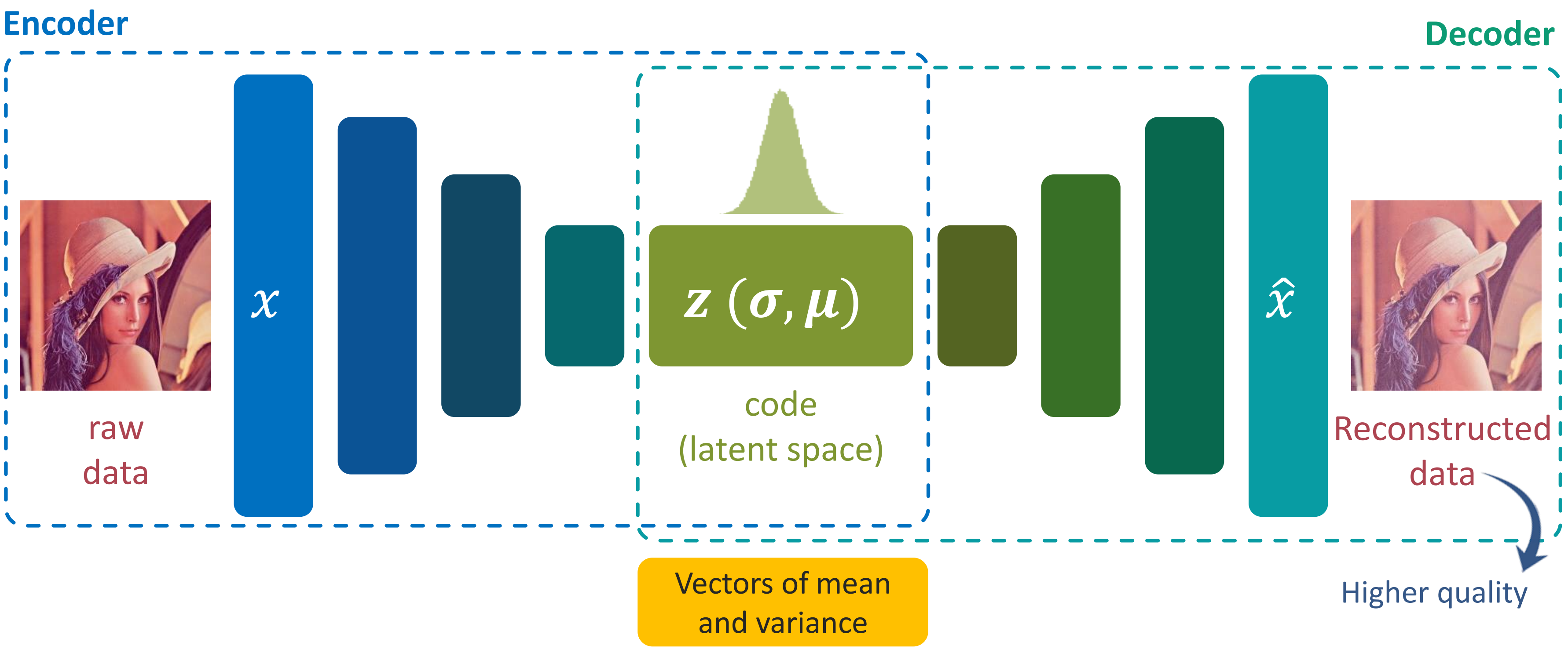
$$Loss = reconstruction\ loss + regularization\ loss$$

The difference between input data and the reconstructed output

Divergence between the inferred latent and fixed prior on latent distributions

Variational Autoencoders

The architecture of a variational autoencoder



Variational Autoencoders

Applications of AEs – Music Generation

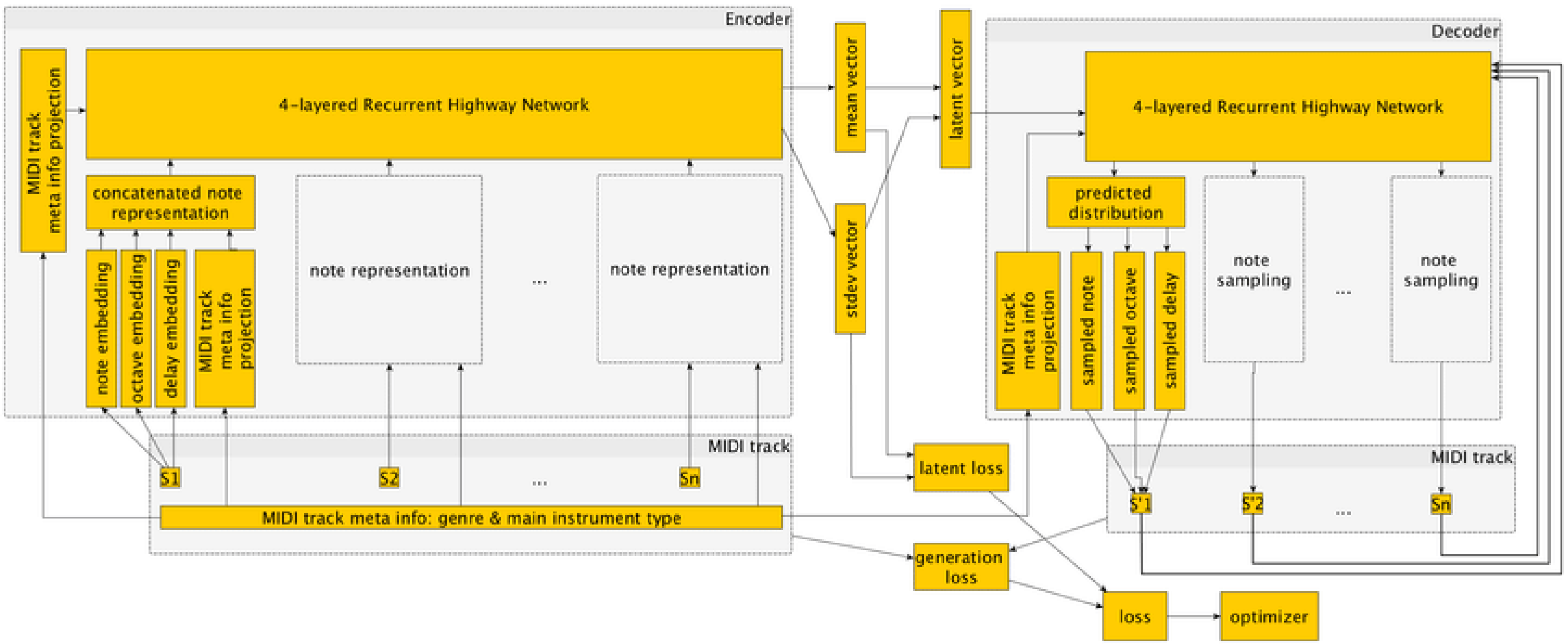


Image source: https://www.researchgate.org

Variational Autoencoders

Applications of AEs – Image Generation

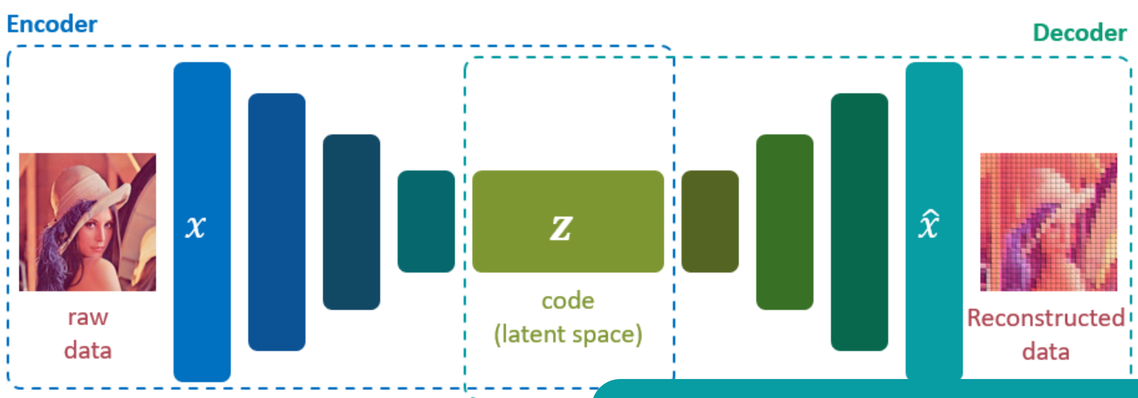
Image source: https://www.towardsdatascience.com

Variational Autoencoders

README.md

Autoencoders (AEs) and Variational Autoencoders (VAEs)

Autoencoders are deep neural network architectures used for **Feature (representation) Learning** in unlabeled data. The goal in an AE is to leverage NNs for representation learning. Autoencoders reconstruct the input approximately by keeping only the most relevant parts of data.



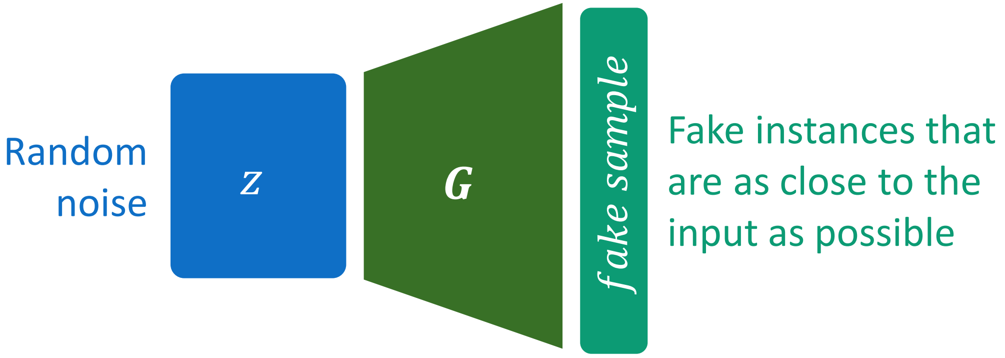
Codes

#	File	Description
0	Basic introduction	Introduction to Autoencoder codes in Keras

Full code on GitHub

Generative Adversarial Networks

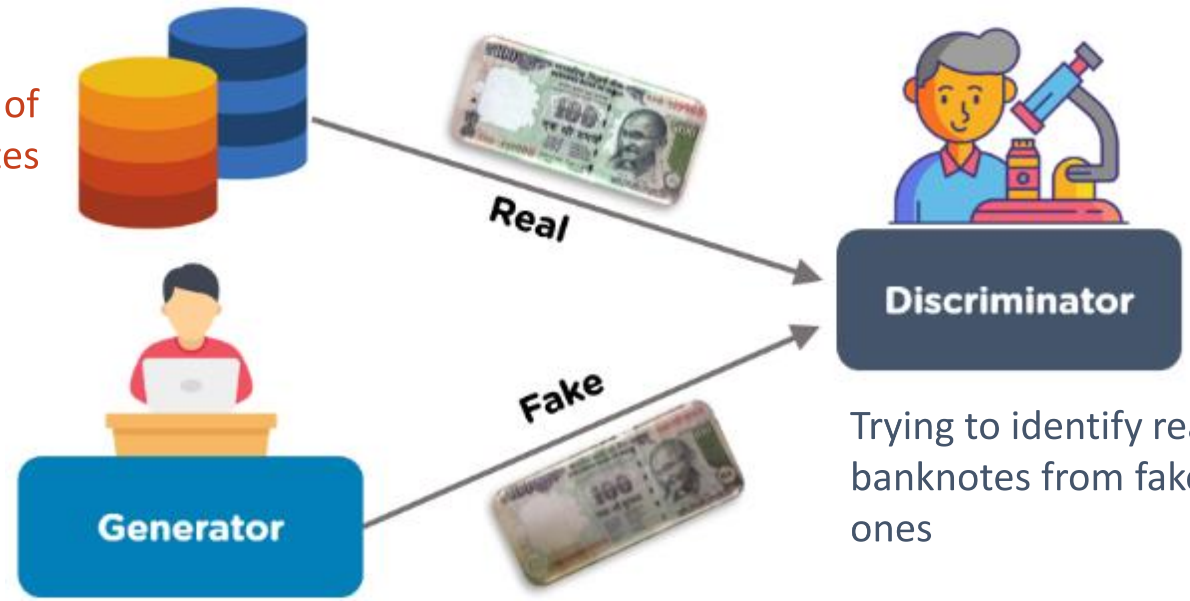
- A deep learning framework to generate new data similar to the training set
- **Goal:** sampling from a complex distribution
  - Building some approximations of this distribution
- **Idea:** starting from **random noise** (a simple data), then learning a functional **transformation** that goes from noise to the data distribution



Generative Adversarial Networks

Basic idea

Database of
Real Banknotes

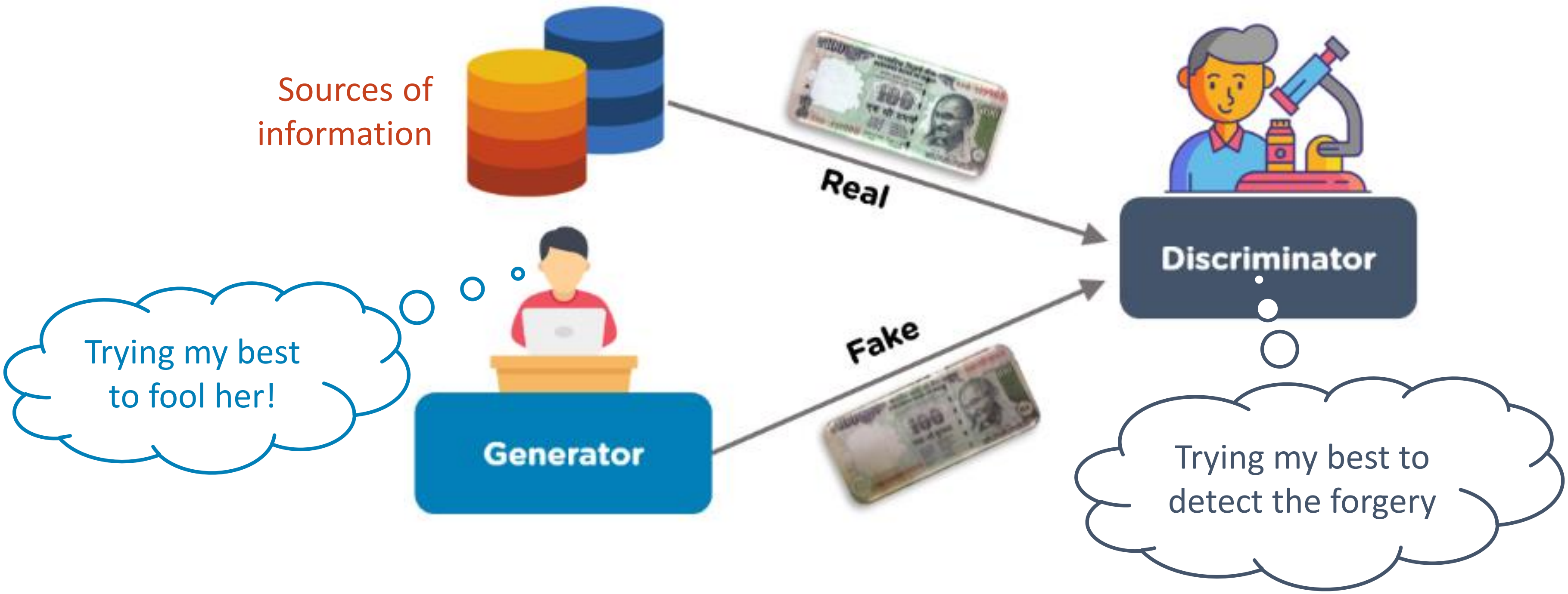


Generating new instances that are as close as possible to the fake data

Trying to identify real banknotes from fake ones

Generative Adversarial Networks

Basic idea



Generative Adversarial Networks

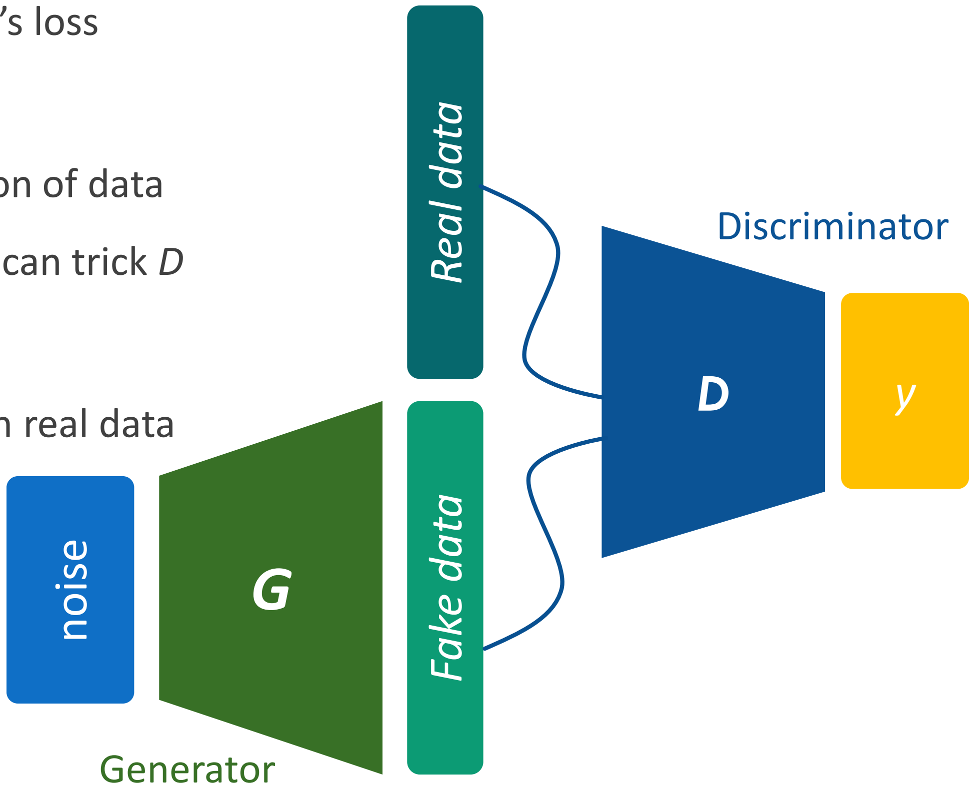
- Contains two NN that compete with each other in a game
  - One NNs' gain is another NN's loss

Generator

- Turning noise into an imitation of data
- **Goal:** generate samples that can trick *D*

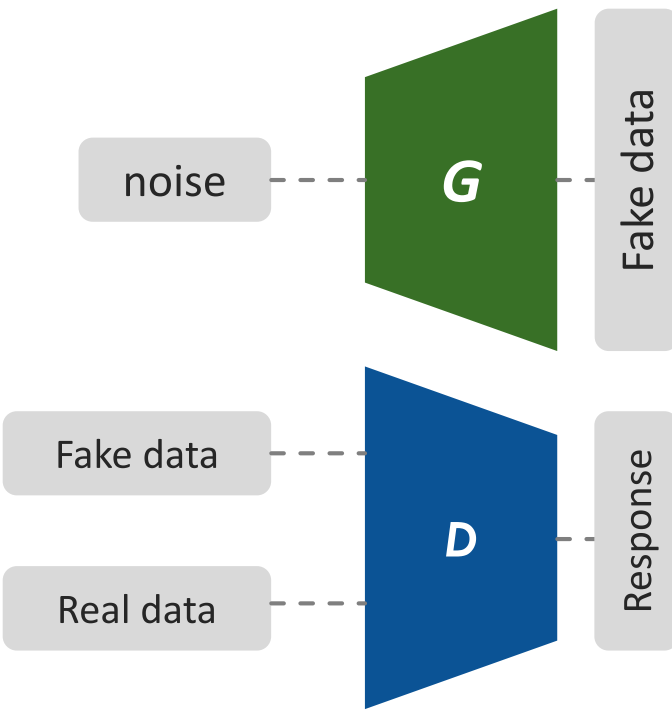
Discriminator

- Trying to distinguish between real data and fake (generated) ones

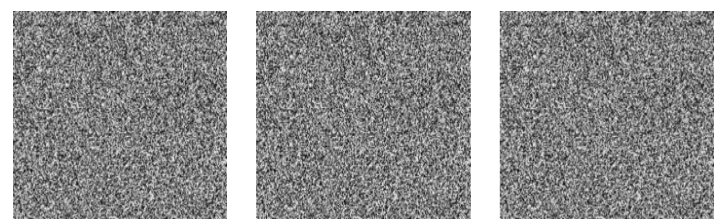


Generative Adversarial Networks

How do GANs work?

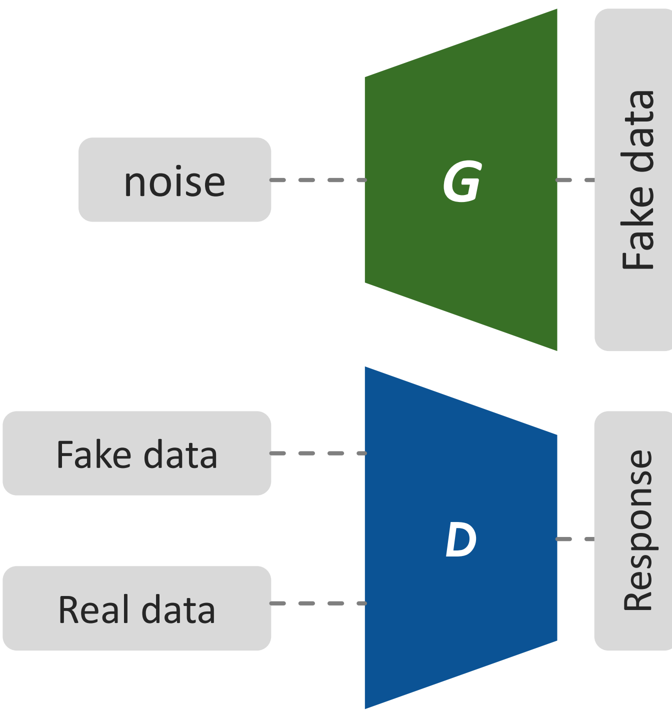


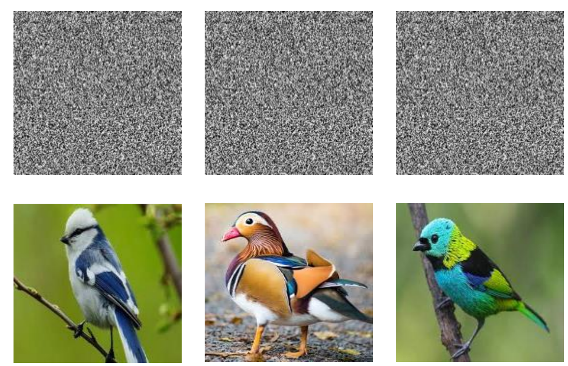
Generator: these are birds!



Generative Adversarial Networks

How do GANs work?





Fake data

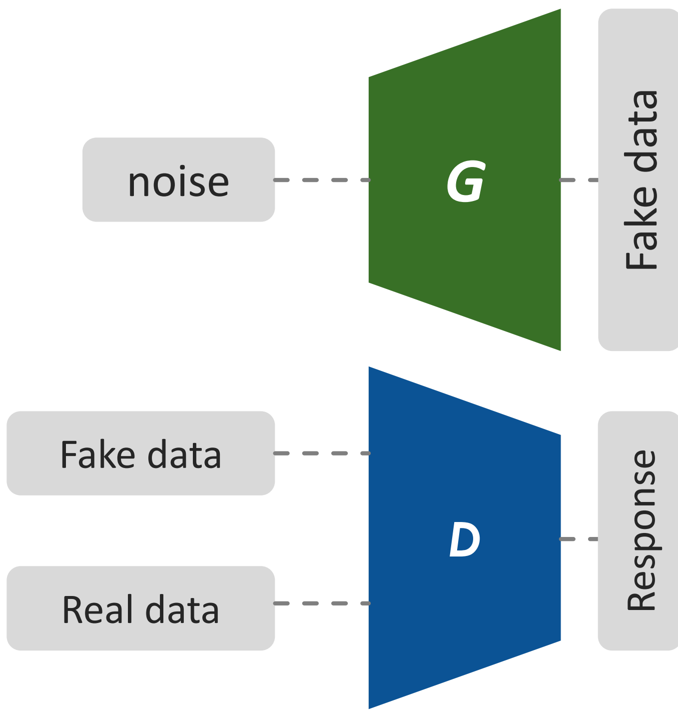
Real data

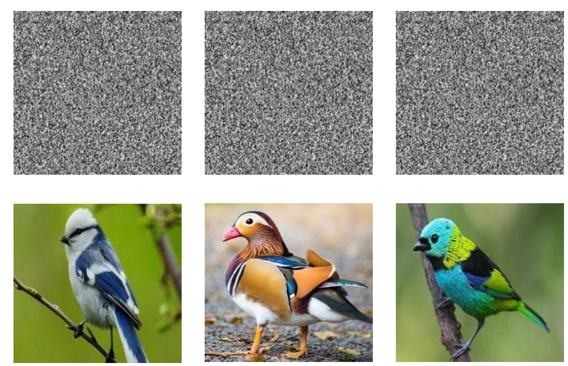
Discriminator: I don’t think so!

Generative Adversarial Networks

The discriminator needs to be trained over time to identify real and unreal items

How do GANs work?





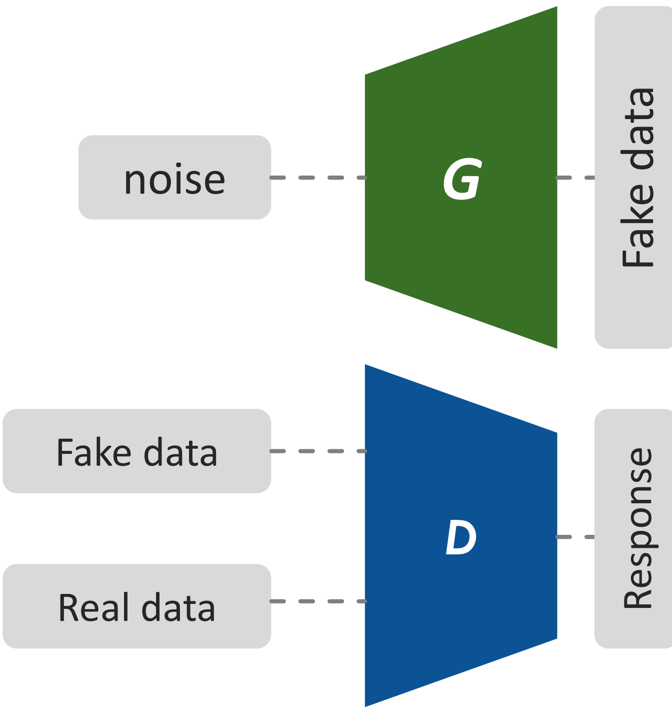
Fake data

Real data

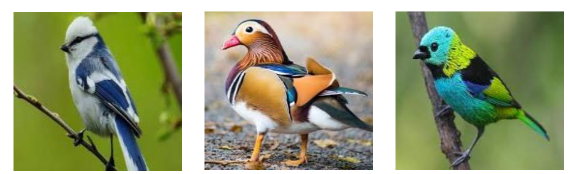
Discriminator: I don't think so!

Generative Adversarial Networks

How do GANs work?



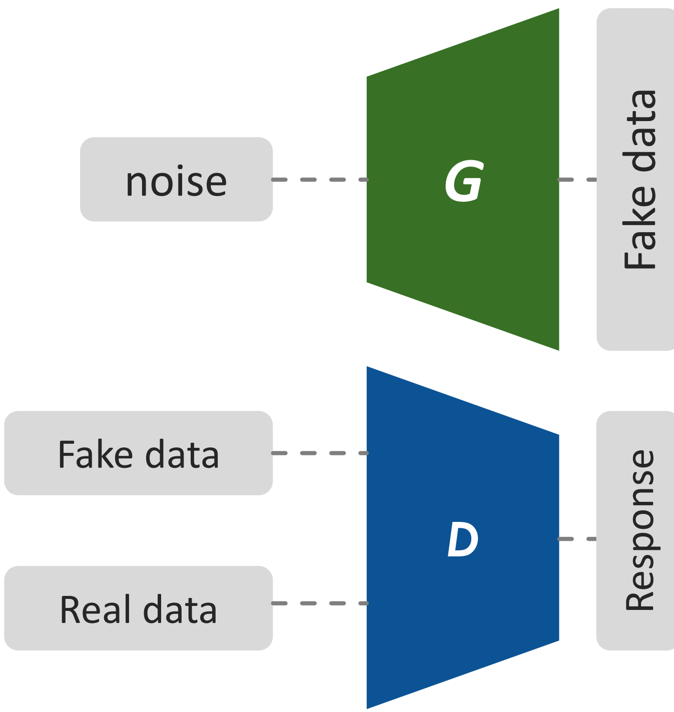
Generator: these are birds!

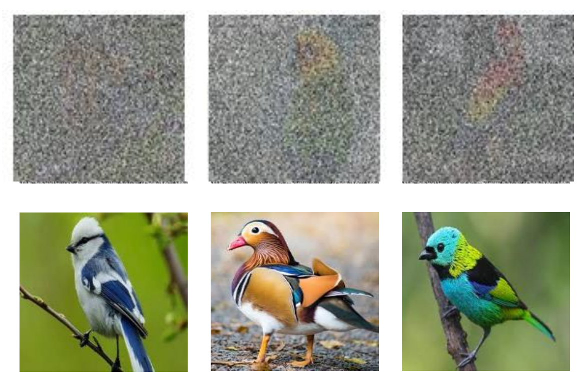


Real data

Generative Adversarial Networks

How do GANs work?





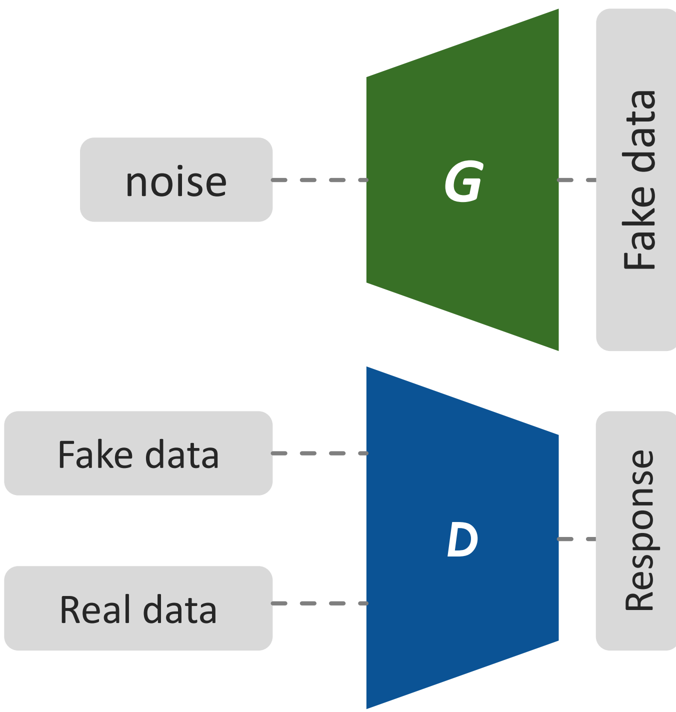
Fake data

Real data

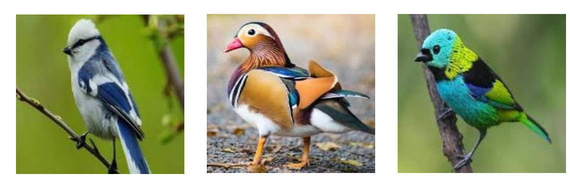
Discriminator: Hmm! Not sure yet!

Generative Adversarial Networks

How do GANs work?



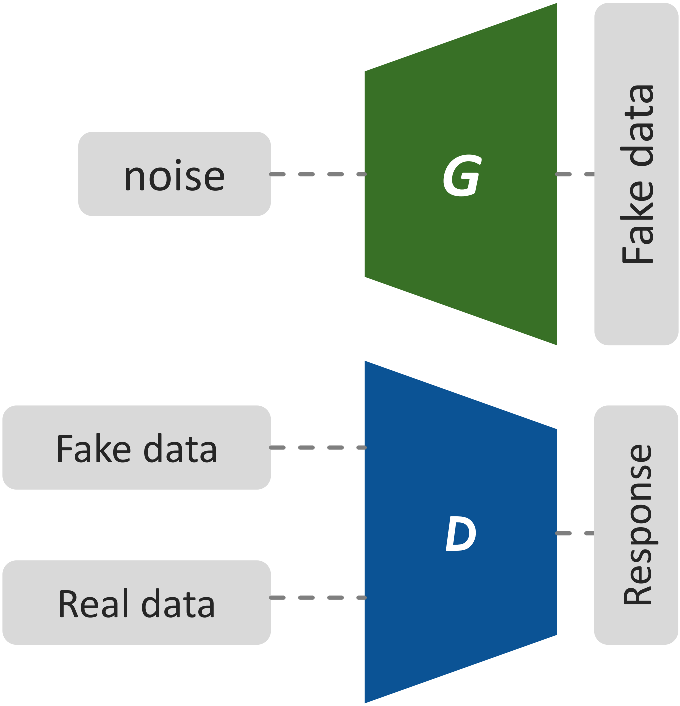
Generator: these are birds!

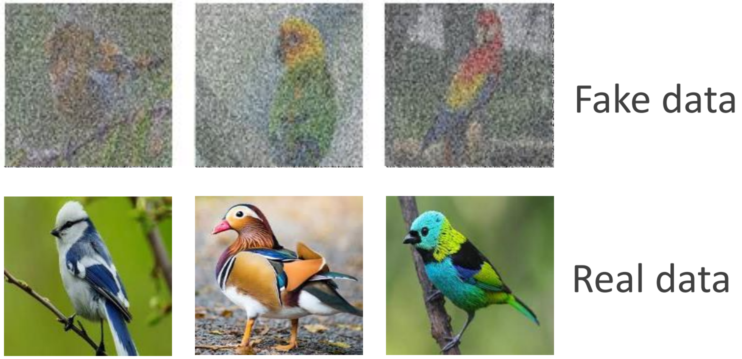


Real data

Generative Adversarial Networks

How do GANs work?

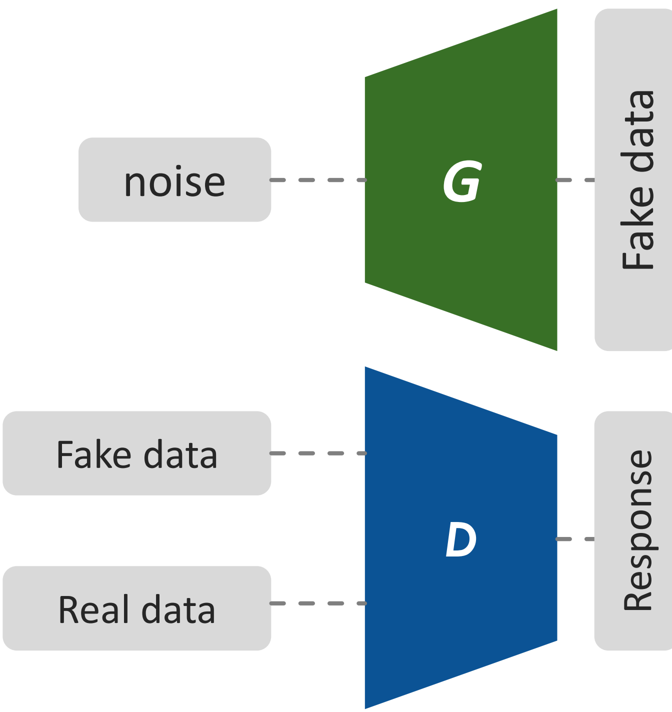




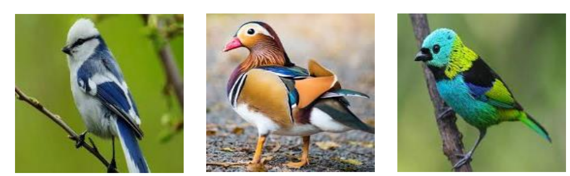
Discriminator: Maybe!

Generative Adversarial Networks

How do GANs work?



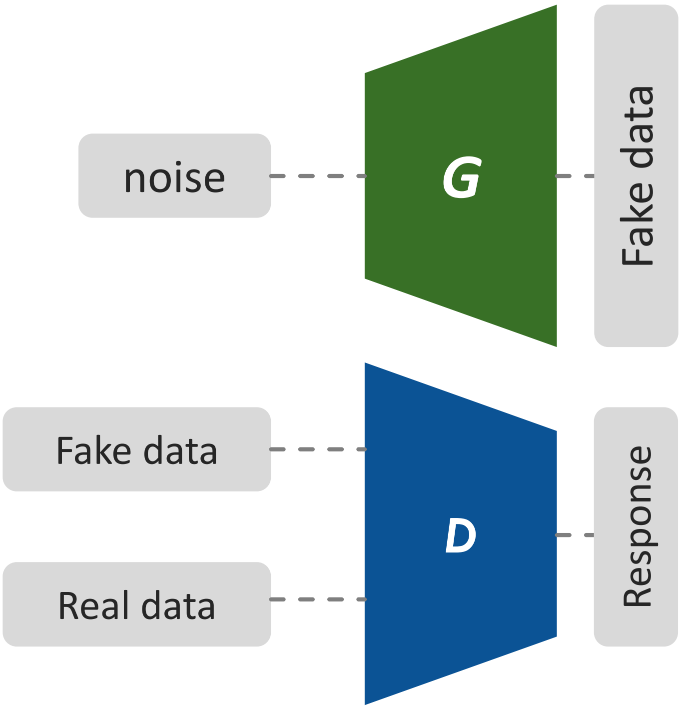
Generator: these are birds!



Real data

Generative Adversarial Networks

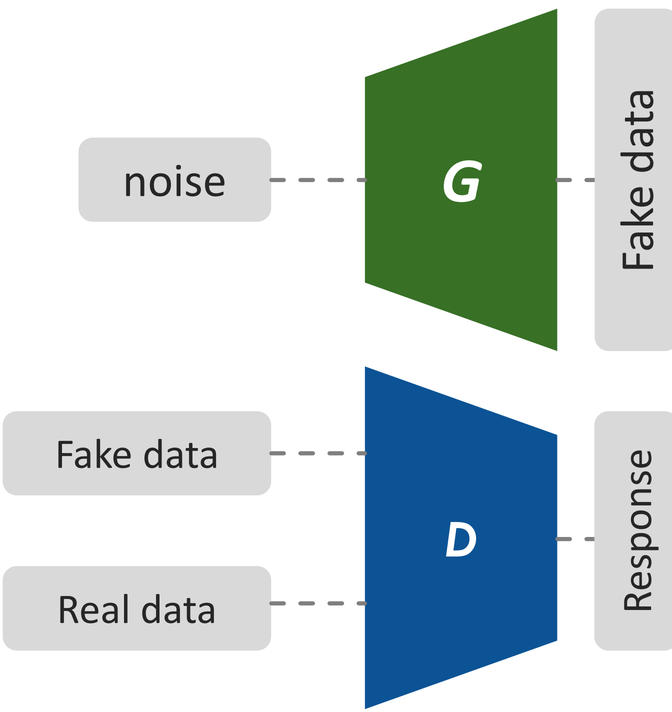
How do GANs work?



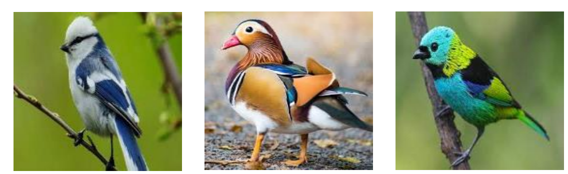
Discriminator: I can see something!

Generative Adversarial Networks

How do GANs work?



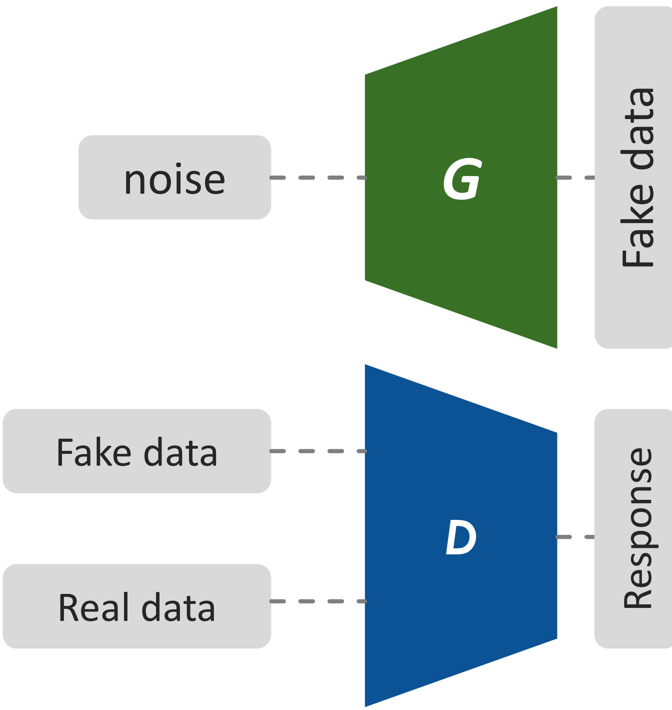
Generator: these are birds!



Real data

Generative Adversarial Networks

How do GANs work?



Fake data

Real data

Discriminator: Yes they are!

Generative Adversarial Networks

Important notes on GANs

- While training, discriminator becomes better to classify real and fake data, and thus, generator should try harder to trick it
- The optimum outcome is when the generator reproduces the true data distribution
- There are two perspectives for calculating loss function in GANs:
  - Loss (*D*): the max probability to correctly identify fake and real data
  - Loss (*G*): the min probability that ***D*** can distinguish real and fake data

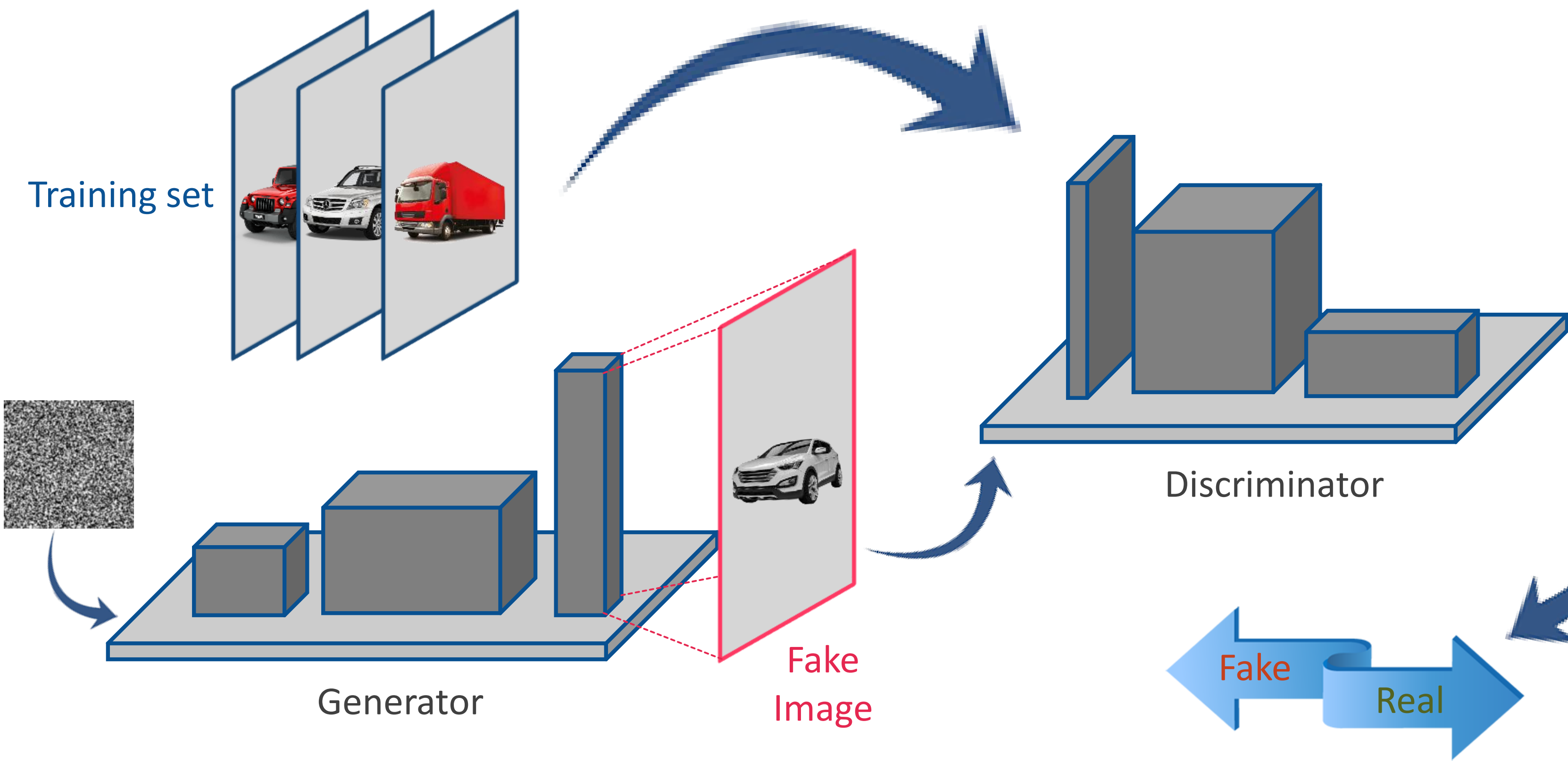
$$loss = \arg min_G \, max_D \, E_{z,x} \left[\log D\big(G(z)\big) + \log(1 - D(x))\right]$$

Generative Adversarial Networks

Important notes on GANs

- The output of a GAN can be a series of generated data that lies in the learned data distribution
- There are many different architectures for GANs:
  - Conditional GAN is a variant of GANs that can enable controlling the type of output using a conditioning factor
  - CycleGAN is another variant that learns a mapping for translation into another domain
  - StyleGAN, an extension to the GAN architecture that proposes large changes to the generator model

Generative Adversarial Networks



Generative Adversarial Networks

Applications of GANs – Human Face Generation

Image source: https://www.machinelearningmastery.com

Generative Adversarial Networks

Applications of GANs – Human Face Generation

StyleGAN

Image source: https://www.towardsdatascience.com

Generative Adversarial Networks

Applications of GANs – Face Aging

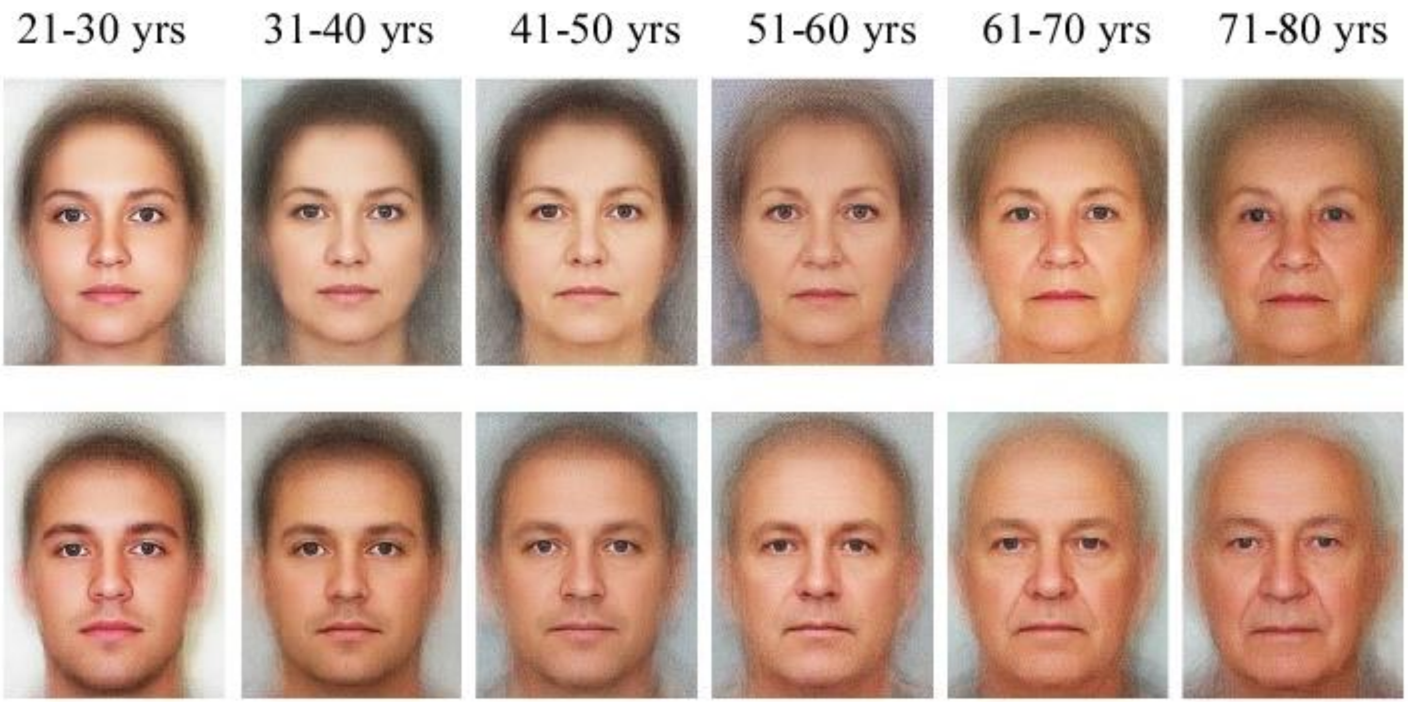


Image source: https://www.semanticscholar.org

Generative Adversarial Networks

Applications of GANs – Text-to-Image Translation

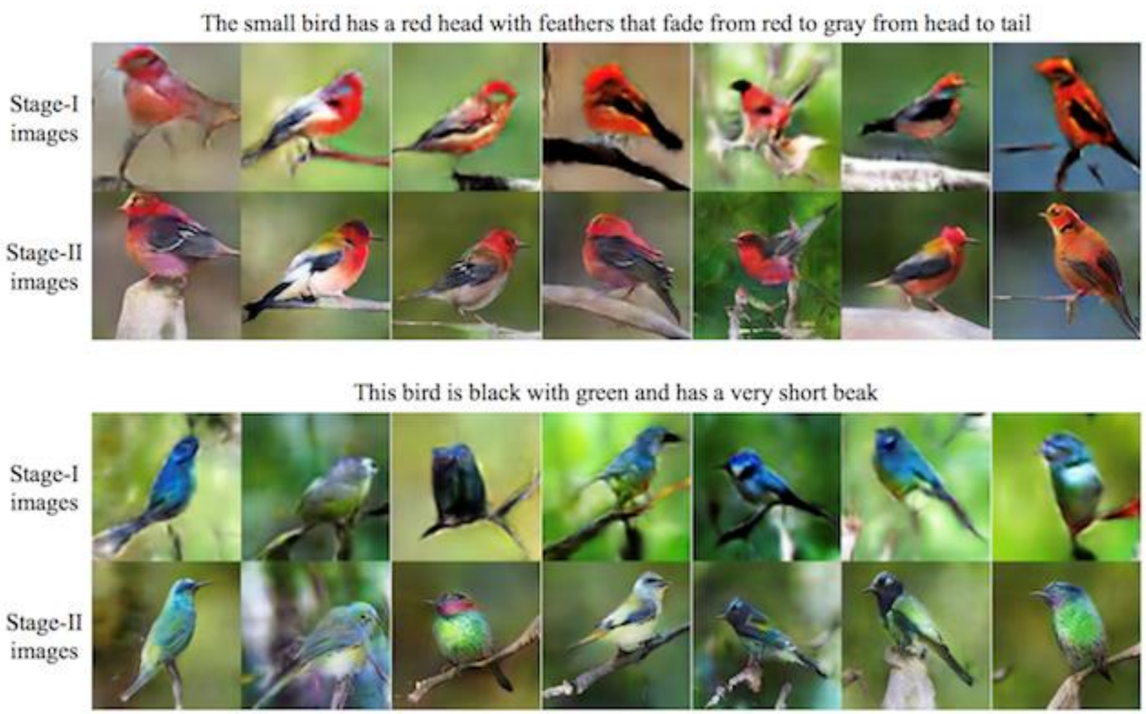


Image source: https://www.machinelearningmastery.com

Generative Adversarial Networks

Applications of GANs – 3D Object Generation

Image source: https://www.redsharknews.com

Generative Adversarial Networks

Applications of GANs – Image Completion

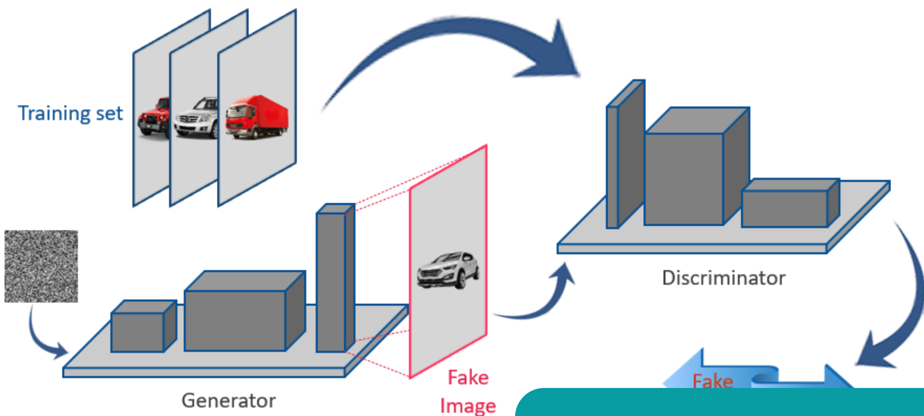
Image source: https://www.cs.brown.edu and https://www.paperswithcode.com

Generative Adversarial Networks

README.md

Generative Adversarial Networks (GANs)

GANs are deep learning frameworks to generate new data similar to the training set. The main goal here is to sample from a complex distribution and build some approximations of this distribution.



Codes

#	File	Description
0	Basic introduction	Introduction to GANs codes in Keras

Full code on GitHub

References

- http://introtodeeplearning.com/
- https://www.guru99.com/supervised-vs-unsupervised-learning.html
- https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/
- https://openai.com/blog/generative-models/
- https://ermongroup.github.io/cs228-notes/learning/latent/
- https://www.jeremyjordan.me/autoencoders/
- https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/
- https://towardsdatascience.com/understanding-generative-adversarial-networks-gans-cd6e4651a29

Questions?