Universität Konstanz

08 | Generative Deep Learning 1

Giordano De Marzo

https://giordano-demarzo.github.io/

Deep Learning for the Social Sciences

Recap

- Sequence Data
- Recurrent Neural Networks
- Long-Short Term Memories
- Applications

Outline

1. Generative Deep Learning
2. Variational Autoencoders
3. Generative Adversarial Networks
4. Applications


Universität
Konstanz
Generative Deep Learning

Supervised vs Unsupervised Learning

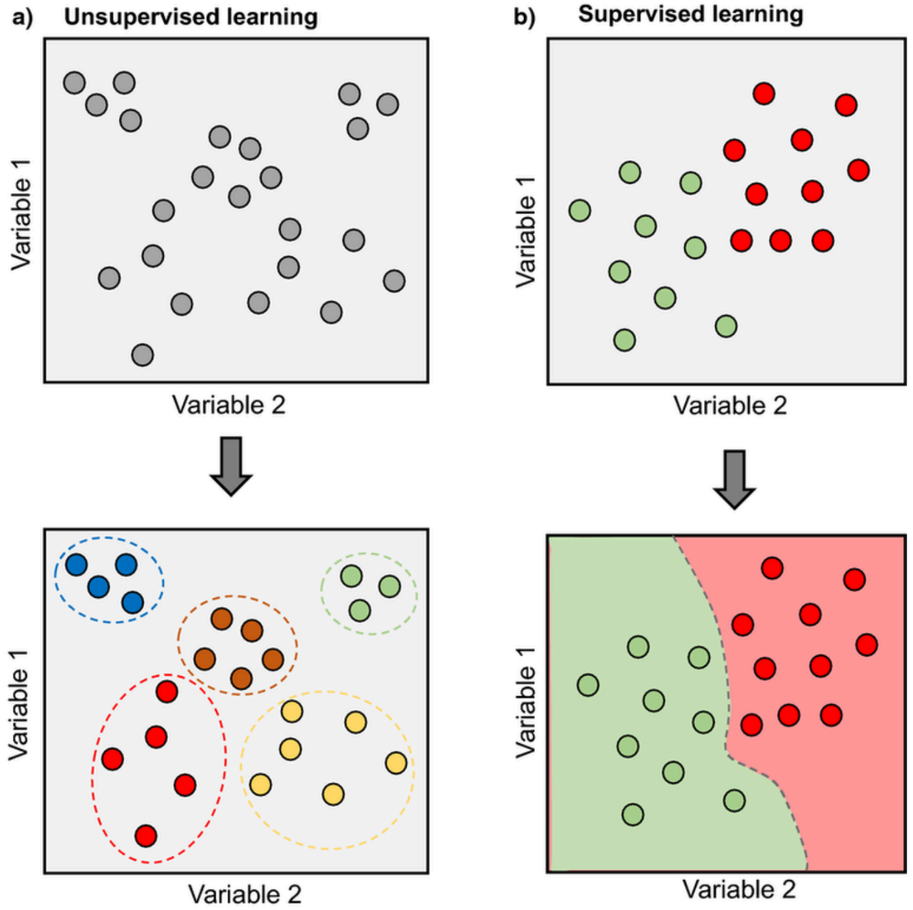
Up to now we mostly consider supervised deep learning models:

- MLP for classification and regression
- CNN for classification
- GNN for classification

We only considered one architecture that performs unsupervised learning, the Autoencoder.

Unsupervised learning tasks include

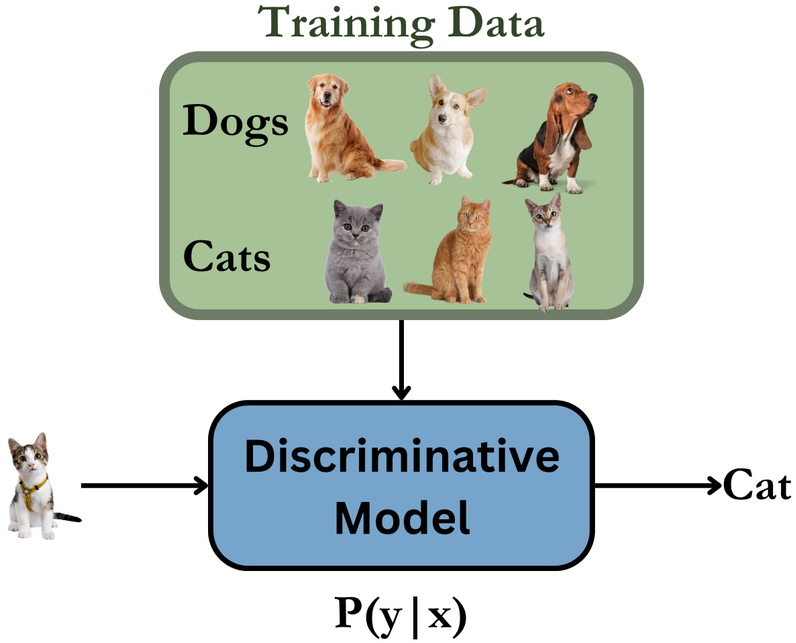
- clustering
- dimensionality reduction
- data generation

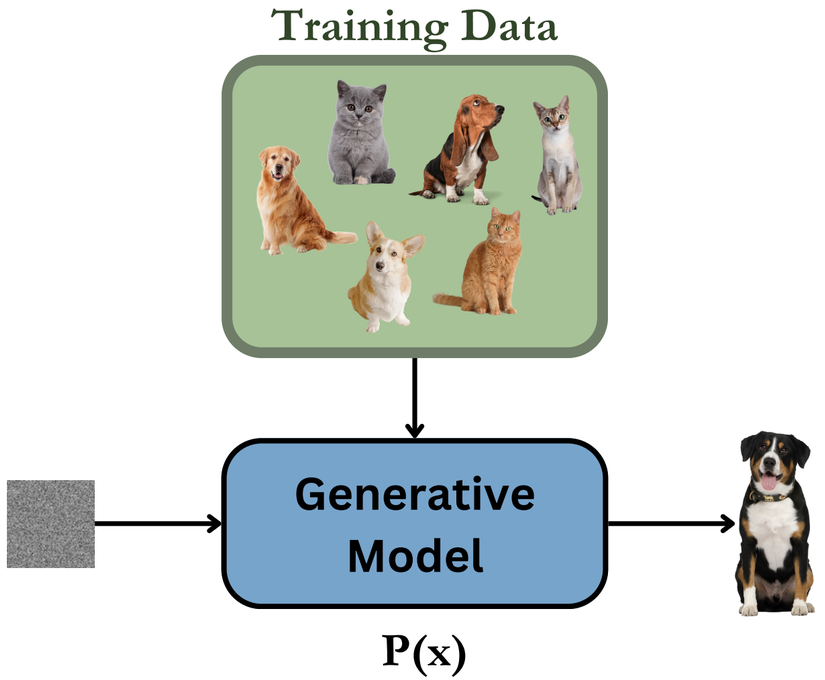


Discriminative vs Generative

Discriminative models assign labels to data: they reconstruct the probability P(y|x) of the label (y) given the data (x)

Generative models create new data: they reconstruct the probability P(x) of the data in our sample





Conditional Generative Models

The information about the label can be included in generative models getting the so called conditional generative models:

- they generate data given labels
- they reconstruct the conditional probability P(x|y)

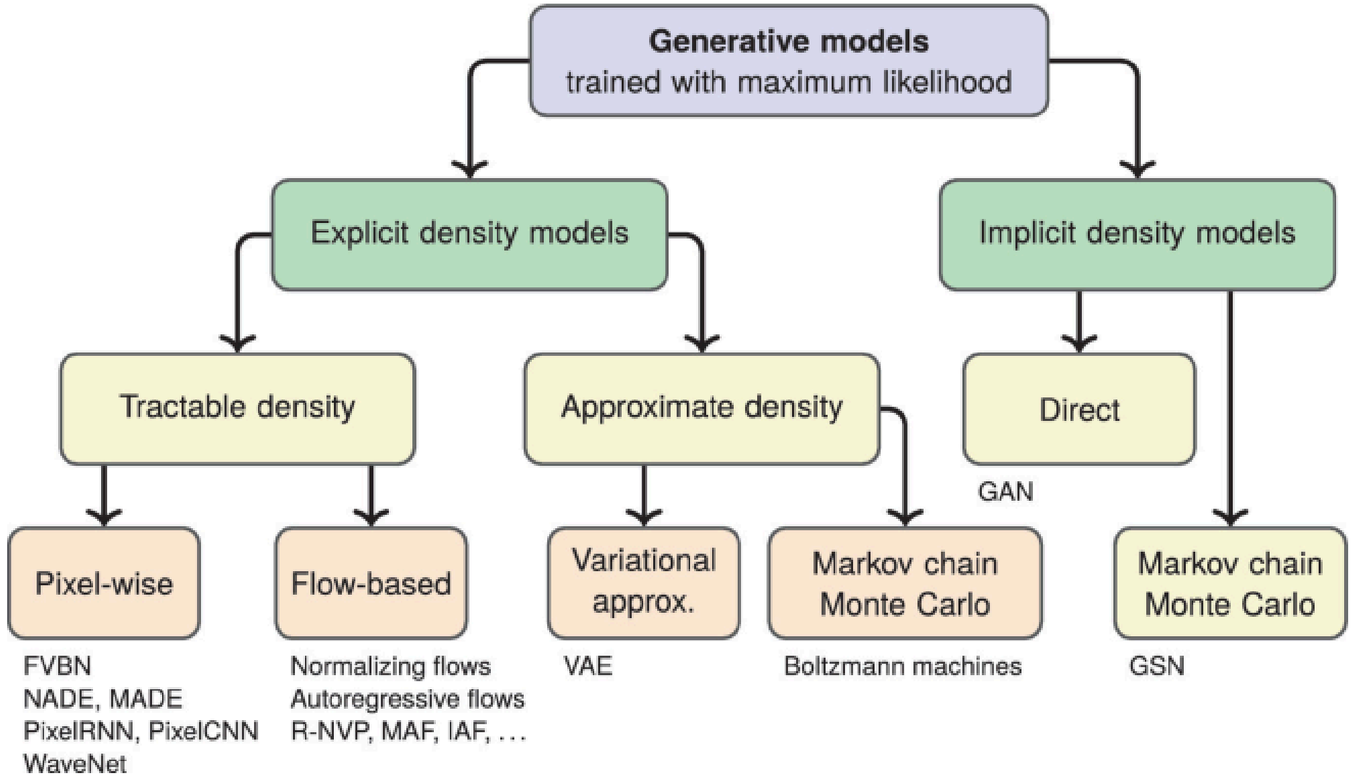
The 3 classes of problems are related by Bayes' theorem

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

A photo-realistic picture of a sailboat on a calm lake at sunset

Taxonomy of Generative Deep Learning

There is a jungle of generative deep learning models depending on if and how they reconstruct the data probability P(x). We will focus only on VAEs and GANs.



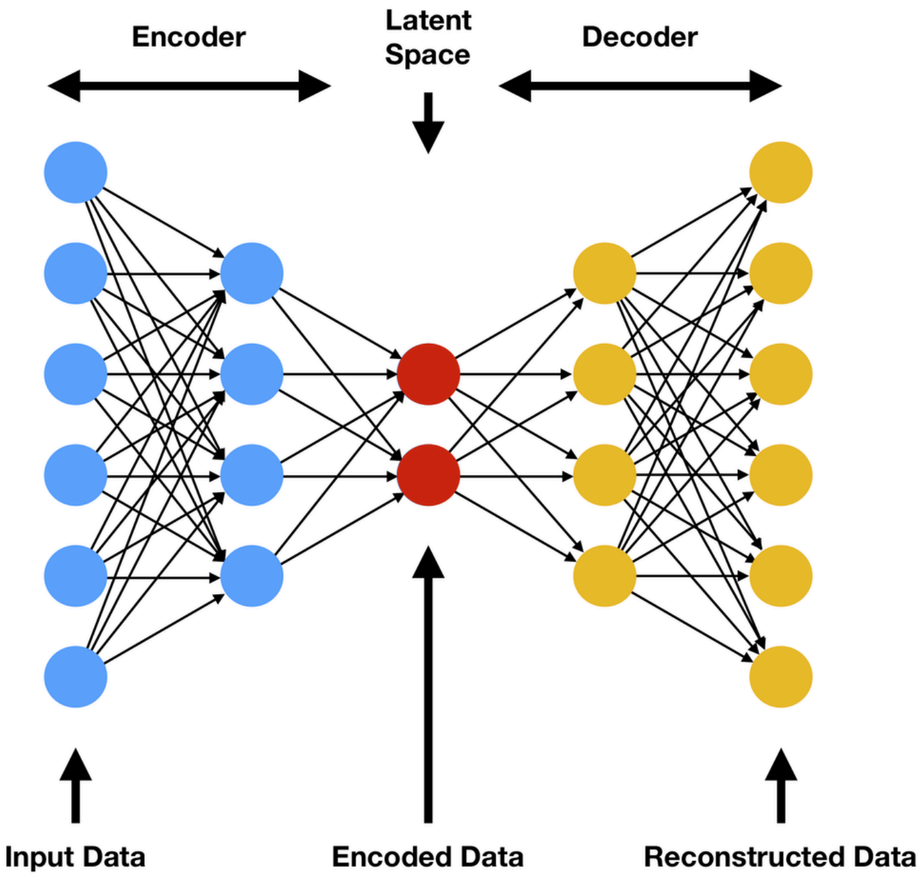
Strengths of Generative Deep Learning

Generative deep learning is an incredibly powerful technique

- Discriminative deep has a crucial weakness, the need of labelled data
- Generative deep learning only requires large amounts of unlabelled data
- Generative deep learning models can thus be trained on more data and have a larger number of parameters
- This allows generative deep learning models to better learn the hidden features and regularities of data
- The regularities learnt by such models can then be used for enhancing the performances of discriminative models

Variational Autoencoders

The Autoencoder



The Autoencoder is one of the most important MLP architectures for unsupervised learning. It is composed of three sections:

- **Encoder** Encodes the data into a latent representation
- **Latent Space** Space where the encoded data live (z)
- **Decoder** Convert back the data from the latent space to the standard representation

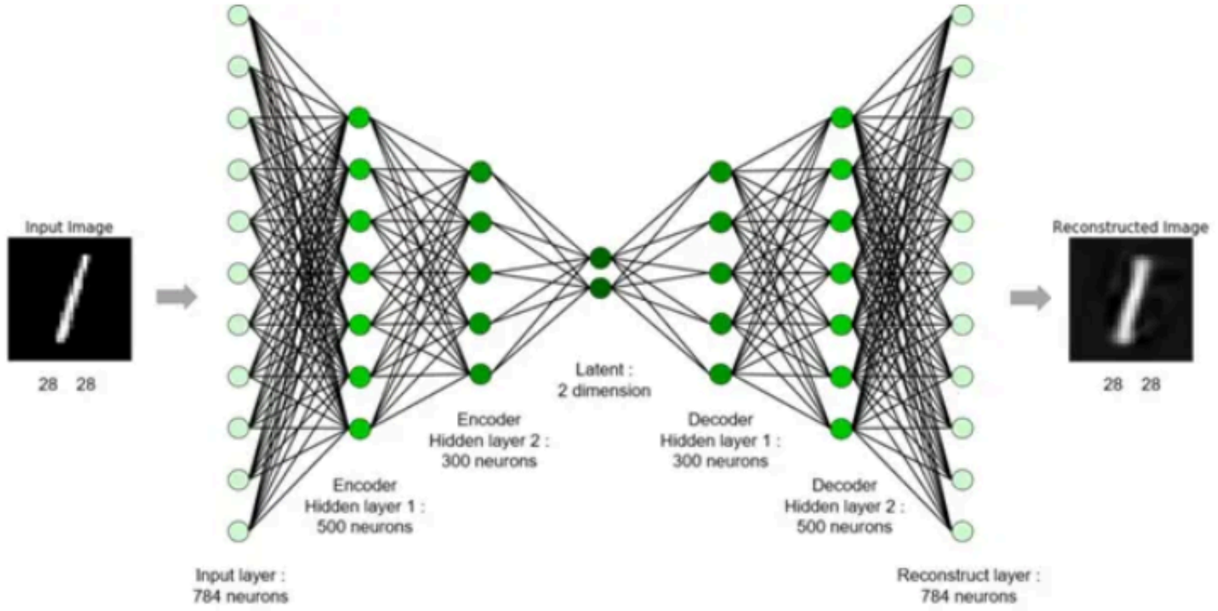
The Autoencoder can be used for several tasks

- Dimensionality Reduction
- Anomaly Detection
- Denoising

Unsupervised Learning

The Autoencoder is trained in an unsupervised way using a reconstruction loss

- it is trained to reconstruct in output exactly what it is given in input
- the loss quantifies how different is the output from the input
- practically speaking it is like a supervised regression, but there is no need of labelled data

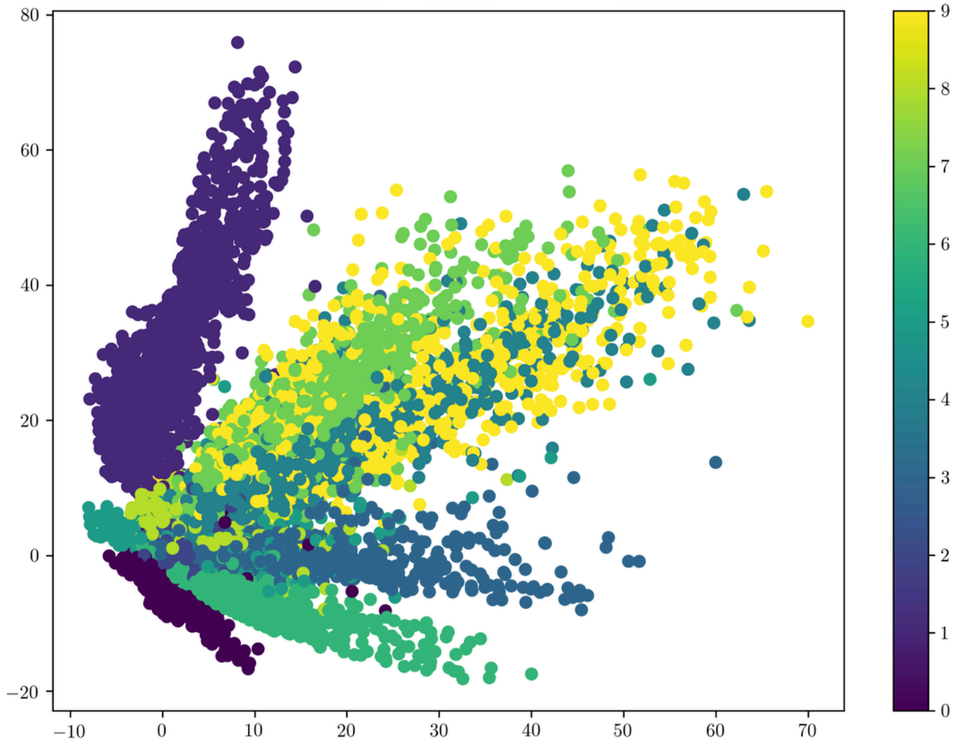


Dimensionality Reduction

The Autoencoder architecture has a bottleneck, the latent space:

- if the reconstruction loss is low, the autoencoder can successfully reconstruct the input
- this means that the latent space contains enough information to reconstruct the input
- the latent space contains a low dimensional representation of the input data

Therefore we can train an autoencoder and then use it encoder to get a dimensionality reduction of the data (similar to PCA or T-SNE)



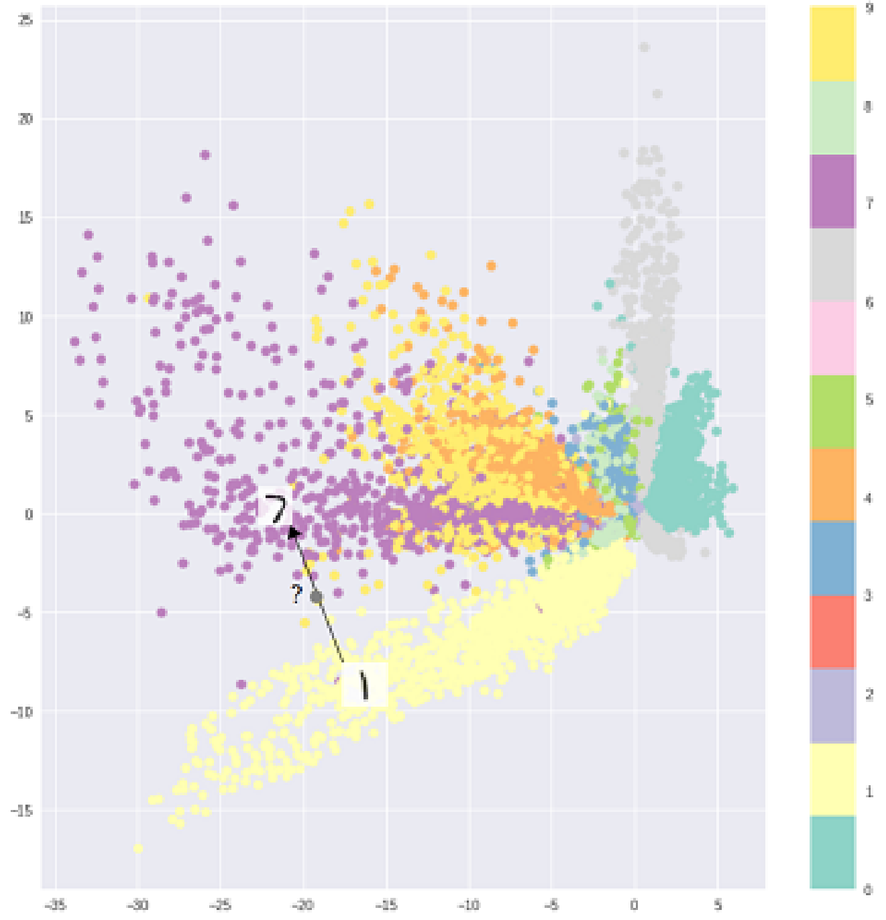
Problems of Generating Data with AEs

One possible approach to data generation involve an AE:

- we train the autoencoder as usual
- we then generate new samples by using the decoder section
  - we select a random point in the latent space
  - we input it in the decoder
  - we collect the output

This is in theory a valid approach, but fails due to the structure of the latent space

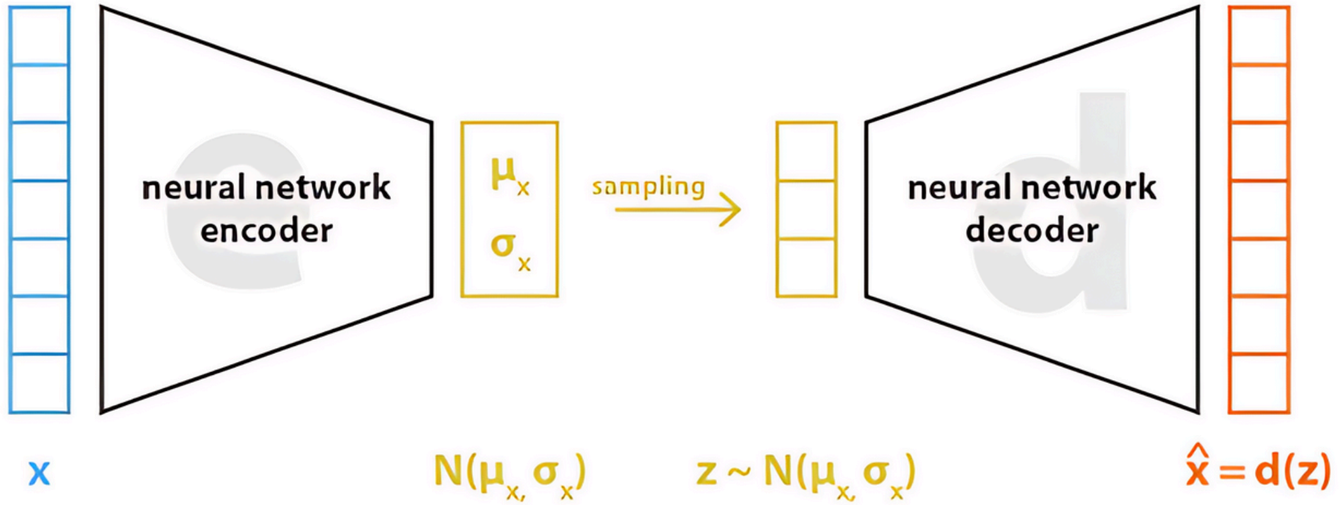
- there is no order
- there are large empty regions
- it is impossible to interpolate between training points



The Variational Autoencoder

The Variational Autoencoder (VAE) adds a twist of stochasticity to the standard AE

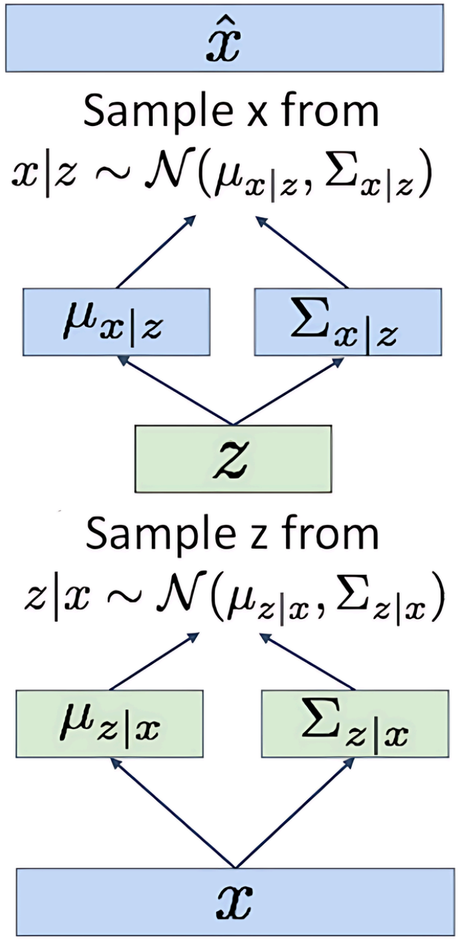
- same architecture, but now the encoder output consists of a mean vector and a variance vector
- the value of the latent variable z is computed sampling from a multivariate gaussian whose parameters are defined by the encoder's output
- the latent variable z in feed into the decoder, whose output is trained to be equal to the input



Probabilistic and Deterministic Decoder

Like the encoder, also the decoder can be either deterministic or probabilistic

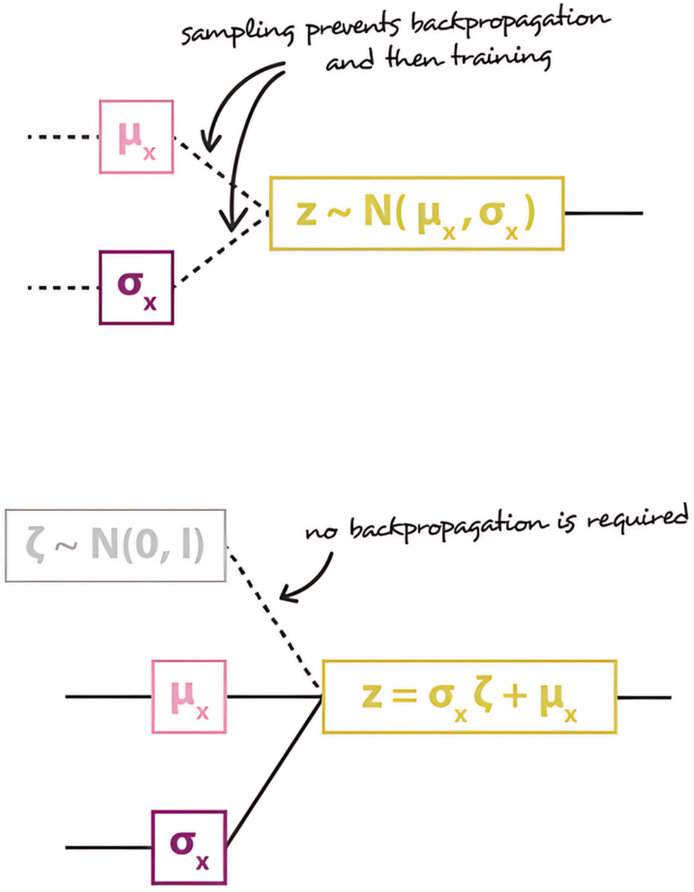
- a deterministic decoder works as in the standard AE, it simply outputs a vector or an image depending on the tasks
- in a probabilistic decoder the output consists of a mean and a variance vector
  - each component of the output vector is obtained by random sampling
  - the sampling is done using gaussians with mean values and variances defined by the output of the decoder
  - a deterministic encoder is a probabilistic encoder with null variance



Reparametrisation Trick

The introduction of stochasticity in the VAE architecture poses some challenges:

- there is no direct way to backpropagate through a stochastic variable
- in order to solve this issue we can exploit the reparametrisation trick
  - we sample from a fixed standard gaussian (mean 0 and variance 1)
  - we multiply the result for the standard deviation and we sum the mean value
  - in this way we can backpropagate since we can compute the derivative



Bayesian Formulation

Autoencoders have a well grounded mathematical foundation

- the assumption is that data x are generated starting from some hidden latent variable z
  - ex: for faces these could be gender, age, expression etc
- the distribution of the latent variables P(z) are independent gaussians
  - there is no correlation between the features
  - each feature will have a typical value with small fluctuations

We can use Bayes theorem to write the probability of real data as

$$P(x) = \frac{P(x|z)P(z)}{P(z|x)}$$

- P(x | z) is the decoder
- P(z) is the normal sampling of the latent variable
- P(z | x) is not directly available, but we train a neural network (the encoder) to approximate it as a function Q(z | x)

Loss and ELBO

The Bayesian formulation allows to define a loss function for training the VAE

- we want to maximize the probability of the observed data point p(x), so we use the log-likelihood as our loss

  $$\text{Loss} = \log[p(\{x_i\}_{i=1...N})]$$

- however this loss is impossible to compute directly, so we have to look at a lower bound, the ELBO (Evidence Lower Bound)

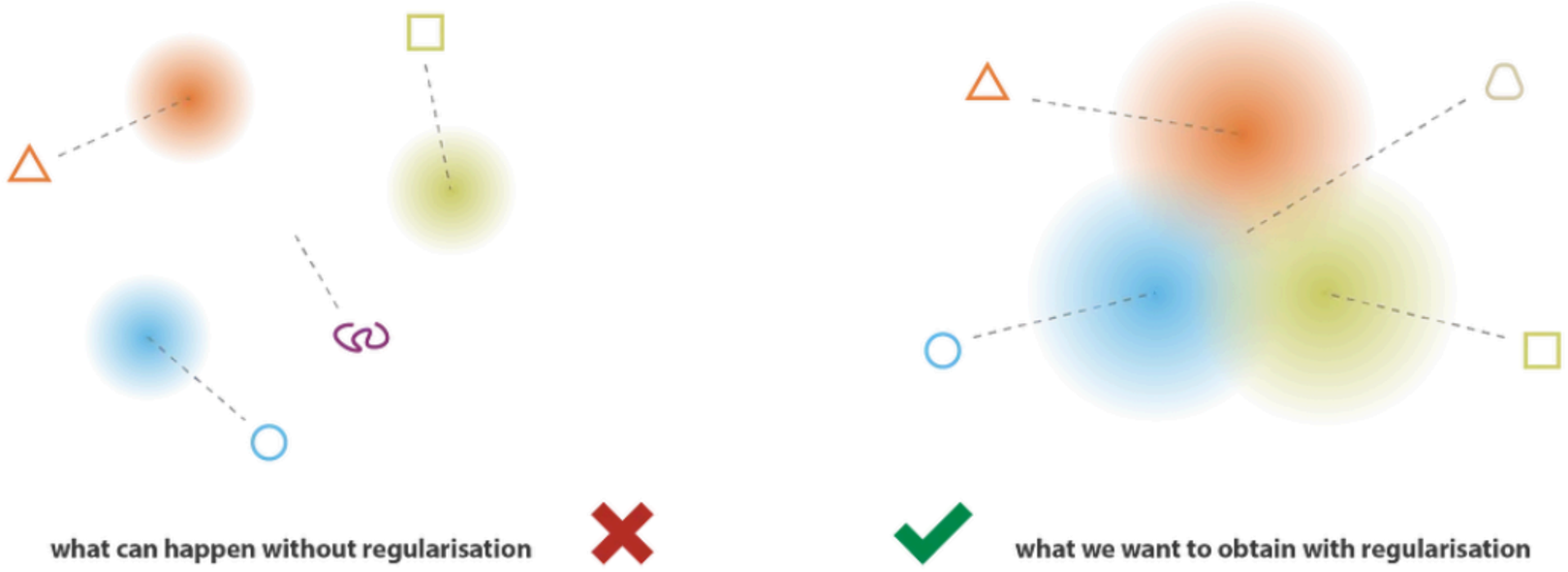
  $$ELBO = E_{z\sim Q(z|x)}[P(x|z)] - D_{KL}(Q(z|x), P(z))$$

  - the first term is the expectation value of the probability of the generated data computed over the latent variable (generated from the encoder)
  - the second term is the distance between the distribution of the encoder and the prior of the latent variable P(z) that is by definition a gaussian distribution

Latent Space and Regularization

This seems very complex, but the concept is very easy:

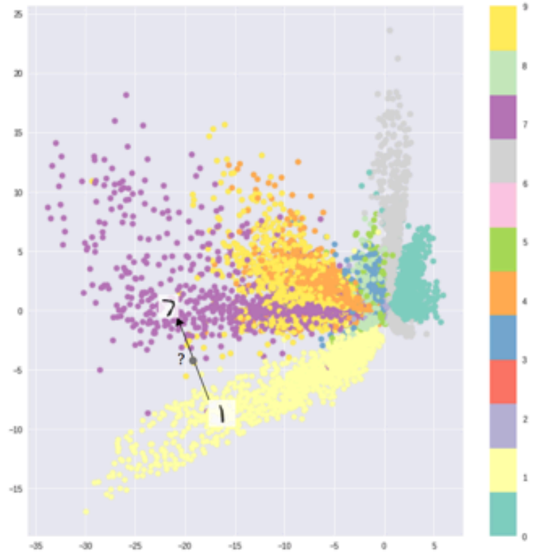
- the first term quantifies how well we can reconstruct the probability of the real data. For a deterministic decoder it is just the reconstruction error of the data
- the second term is a regularization that makes the latent space more regular



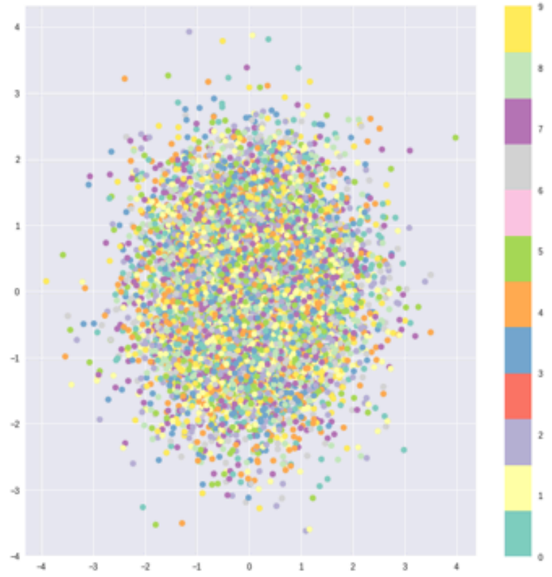
AE vs VAE Latent Space

The reconstruction term would like all points to be very far in the latent space to be easier to distinguish. The regularization term would like to force the points in the latent space to follow a gaussian, so to be very concentrated in the middle.

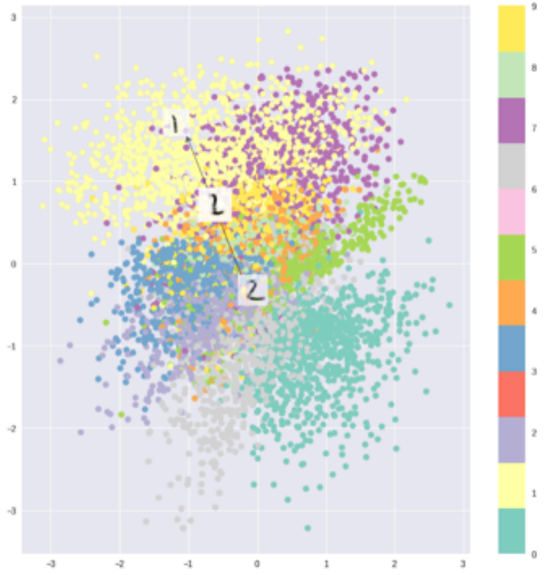
Only reconstruction loss



Only KL divergence



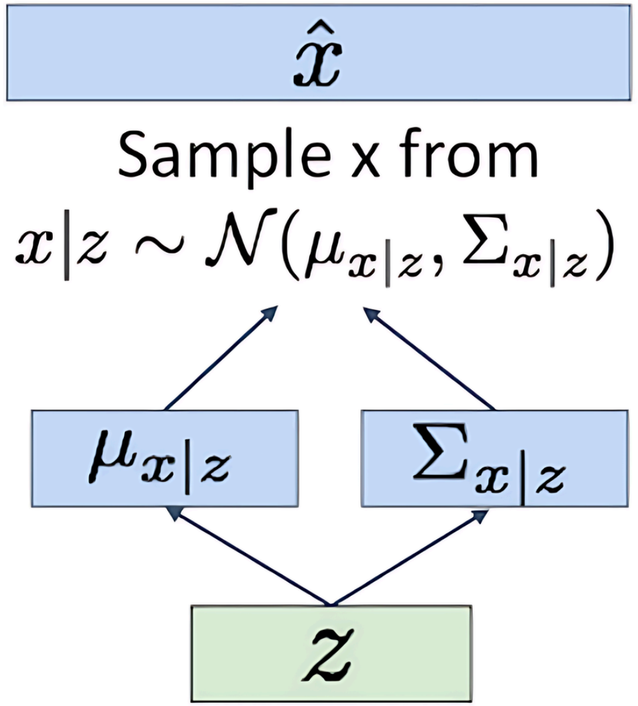
Combination



Generating New Data

Once we have trained a VAE using the ELBO we can use the decoder part for generating new data

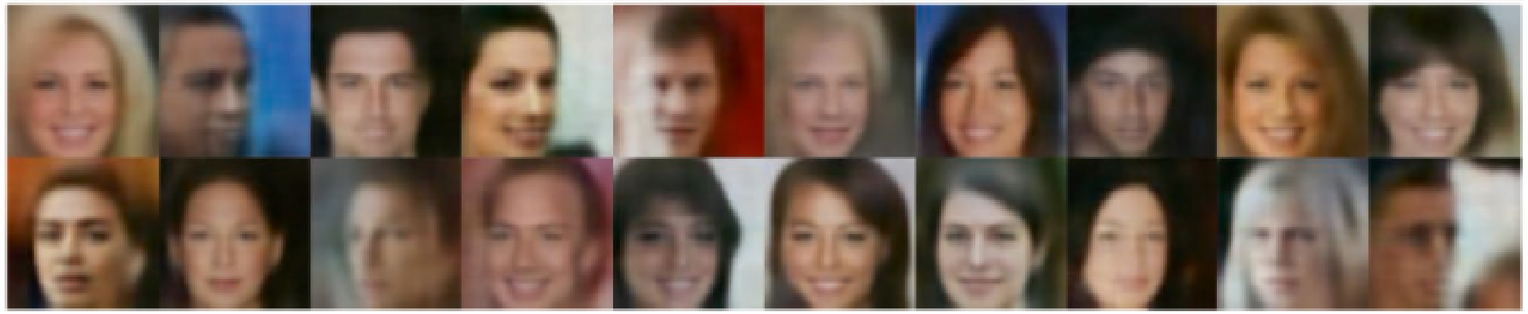
- we select a point in the latent space by sampling from a gaussian
- we process it using the decoder
  - if the decoder is deterministic that is our novel data point
  - if it is probabilistic we use its output to sample the new data
- since now the latent space is much more regular the neural network will be easy to interpolate



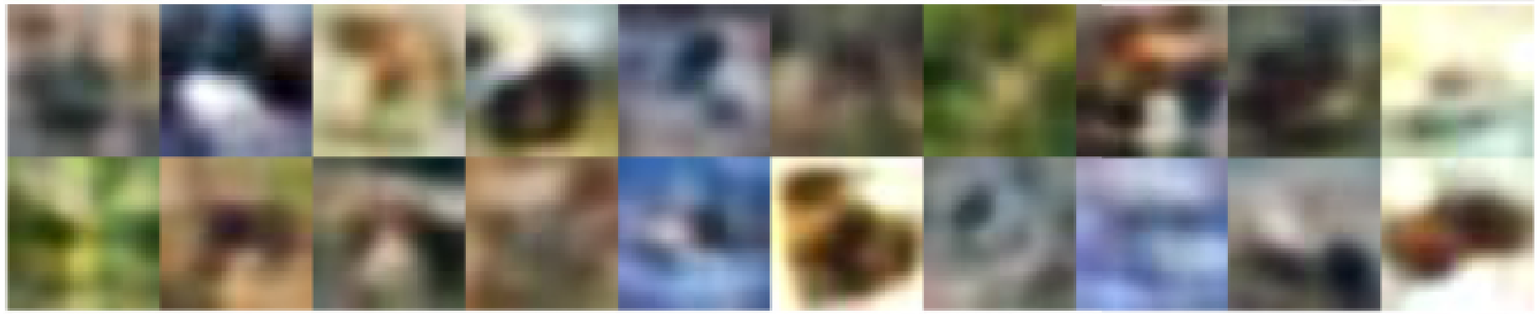
Examples of Generated Data

Human faces represent a standard benchmark for generative deep learning. We can very easily detect AI generated faces, while we may struggle with objects or animals.

Labelled Faces in the Wild Dataset



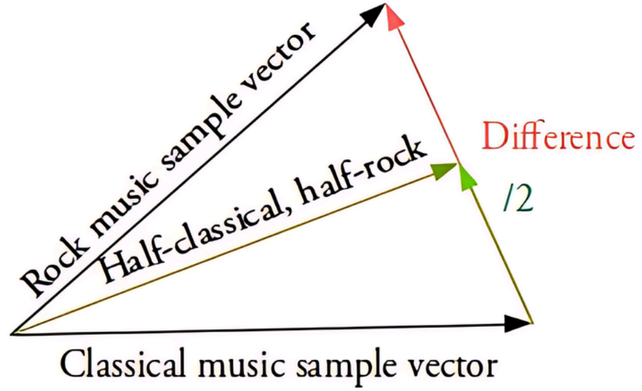
CIFAR-10 Dataset



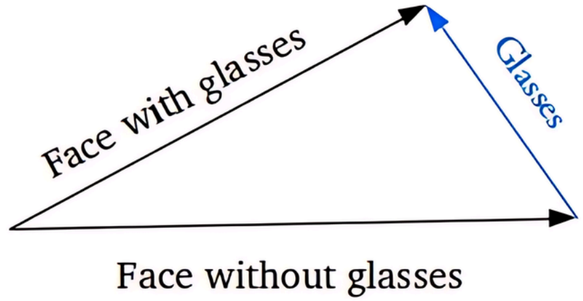
Vectors Arithmetic

The use of the KL divergence term in the loss not only regularizes the latent space, but makes it also very special:

- we are forcing the different directions of the latent space to be orthogonal by using independent gaussians (the covariance is null)
- the result is something similar to a world embedding space, with different directions encoding different concept
- we can subtract two point obtaining the vector that transform one into the other
- we can gradually transition from a data point to another by moving along the line connecting them



Interpolating between samples

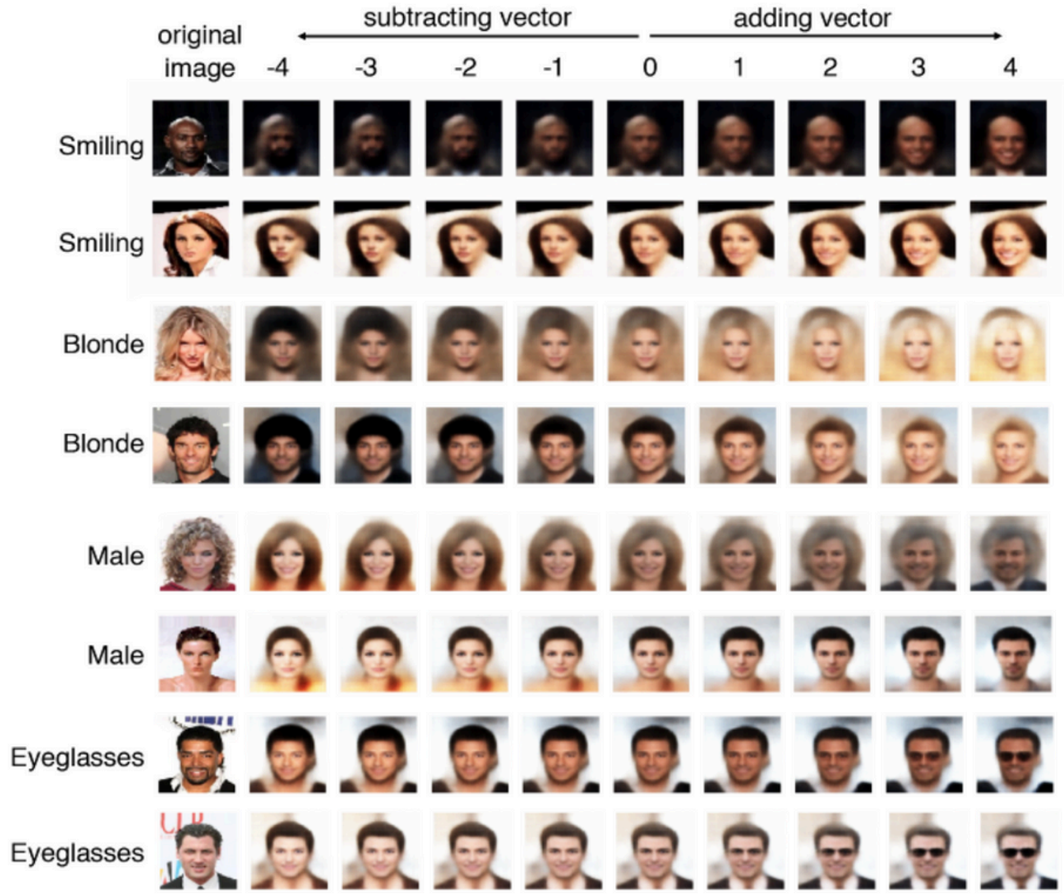


Adding new features to samples

Example: Arithmetic on Faces

In the example we show how it is possible to add different features to images by summing an appropriate latent vector

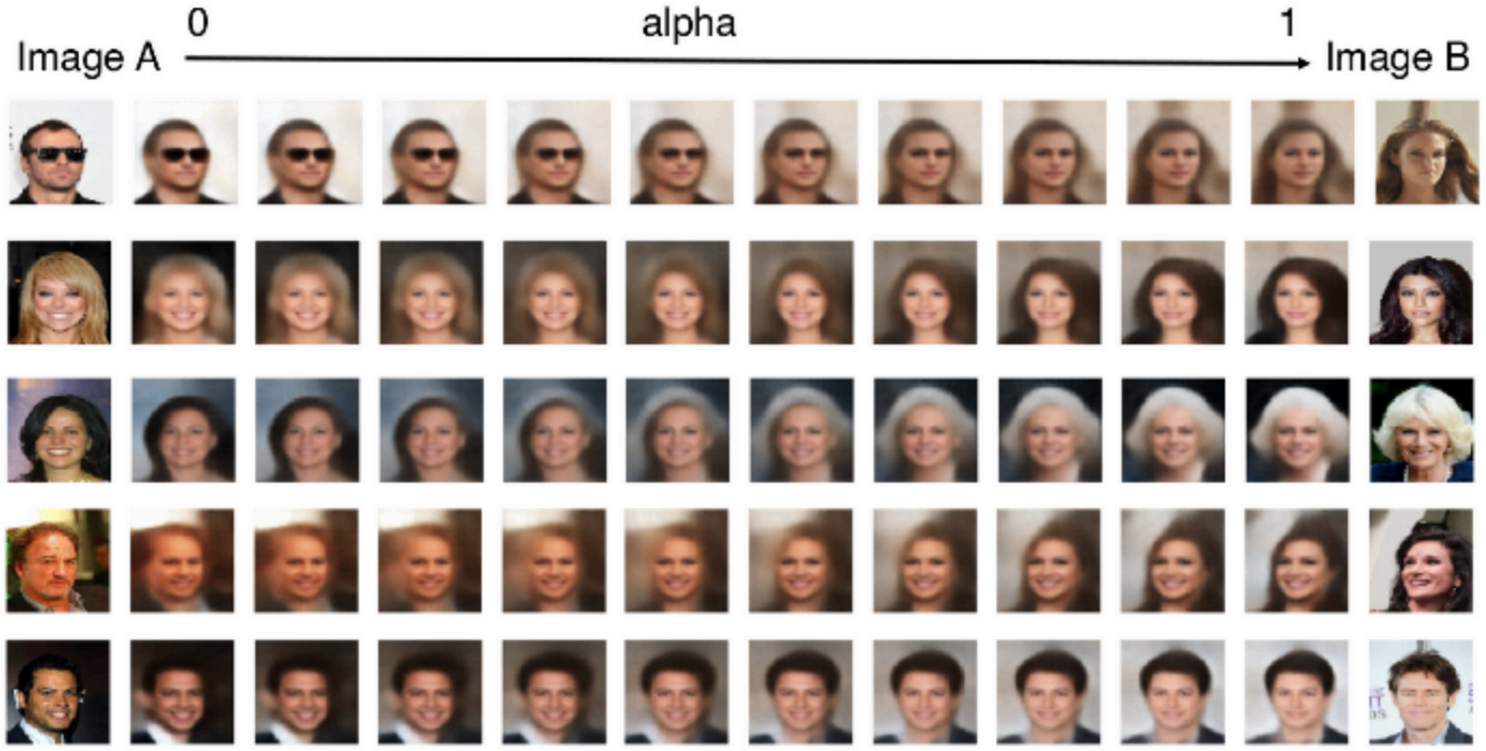
- we start from an original image
- we move along different directions corresponding to different properties
- the resulting output is similar to the original image, but modified including the new feature
- we can modulate the vector to weaken or strengthen the feature



D. Foster, Generative Deep Learning – Teaching Machines to Paint, Write, Compose and Play. O'Reilly Media, Inc., Jun. 2019.

Example: Face Morphing

Below we show an interpolation between pairs of images. This is obtained by moving along the line that connect the two images in the latent space and generate a smooth transition between the two.



D. Foster, Generative Deep Learning – Teaching Machines to Paint, Write, Compose and Play. O'Reilly Media, Inc., Jun. 2019.


Universität
Konstanz
Generative Adversarial Networks

Limits of VAEs

VAE are a very powerful tool

- they are most of the time just better autoencoders
- their latent space has all the good properties we would like it to have
- they are incredible in dimensionality reduction and denoising

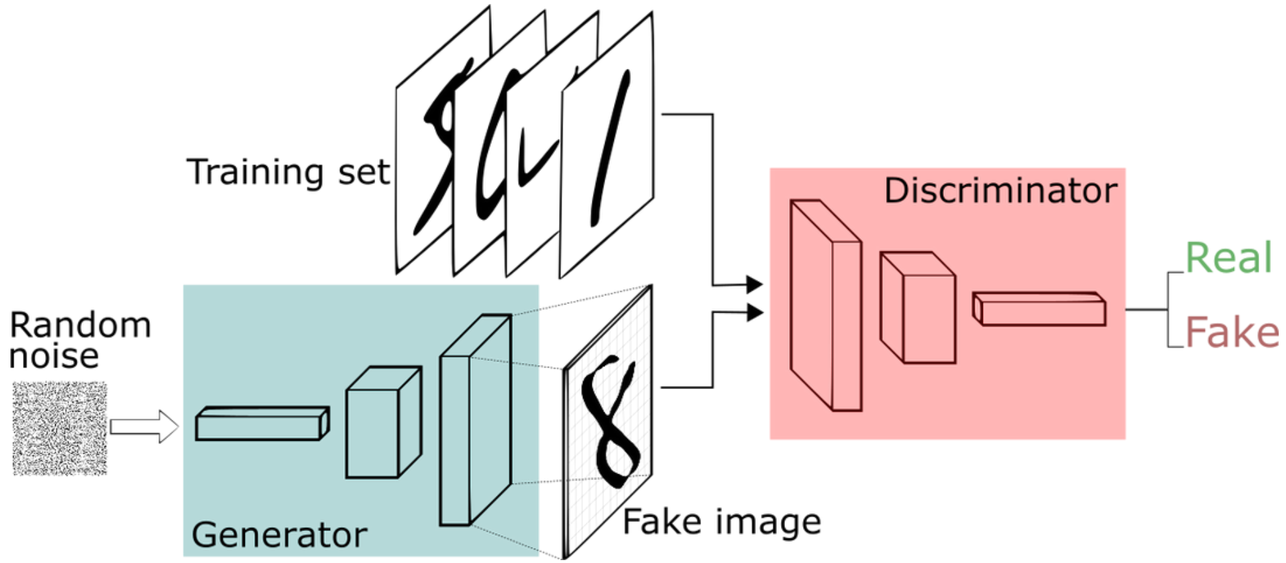
However they have strong limits in data generations, especially when it comes to images

- the quality is not bad
- however imaged are blurred and details are missing
- this derives from the presence of the gaussian regularization

Generative Adversarial Networks

Generative Adversarial Networks (GANs) are much more powerful when it comes to generate realistic images. They are conceptually very similar and are composed of two models

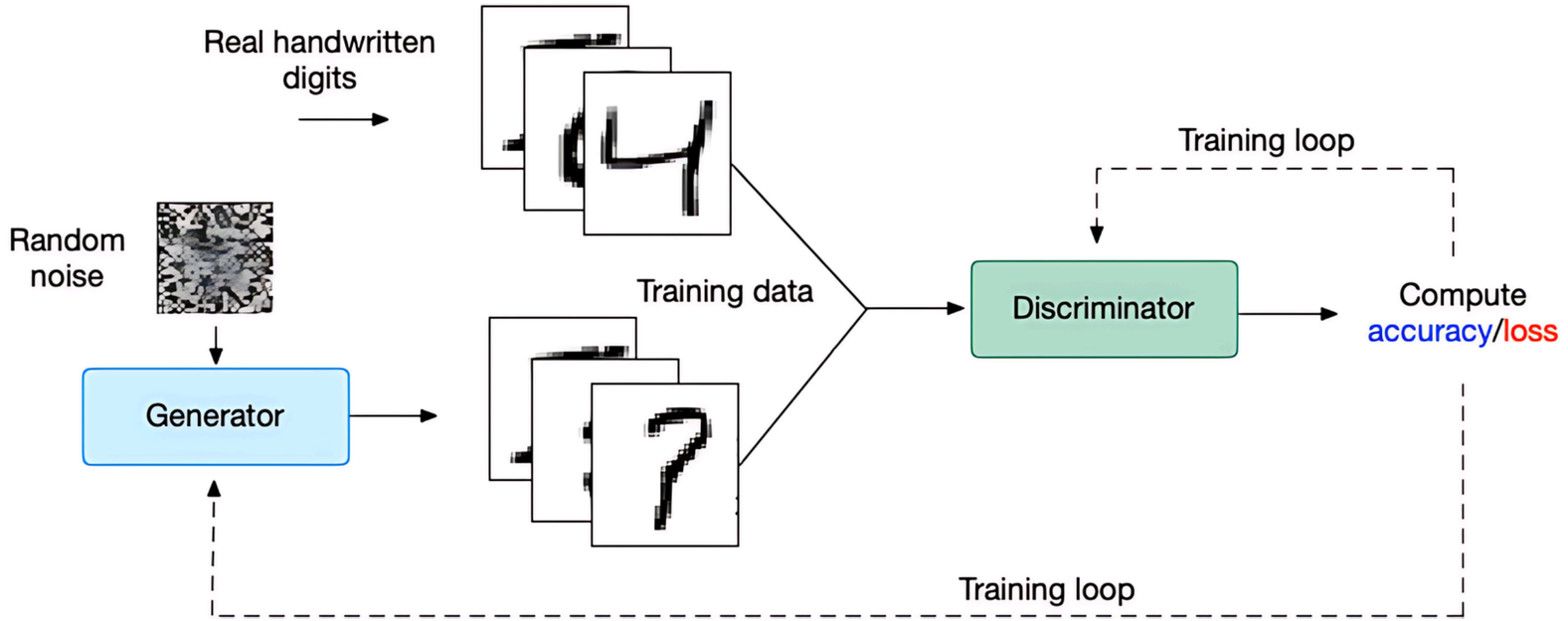
- a generator that is trained to generate artificial data
- a discriminator that is trained to distinguish artificial data from real data
- note that we do not need labelled data (we automatically know what is real and what is fake)



Training GANs

A GAN is trained by alternating between the Generator and the Discriminator

- the generator is used to generate fake images, that are then classified (together with real images) by the discriminator. The generator is then optimized by freezing the weights of the discriminator
- also the discriminator is then updated using the same set of images and the process is iterated



https://poloclub.github.io/ganlab/

MinMax Game

The loss of a GAN is not as simple as for other model

- the discriminator wants to maximize the ability to correctly classify the real and fake images

$$\min_G \max_D \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

- the generator wants to fool the discriminator on fake images, but doesn't care about the real ones

$$\min_G \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

Putting the two ingredient together we get the MinMax Game loss

$$\max_D \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

It can be proven that the optimal value is achieved when the distribution produced by the generator coincides with the distribution of real data. However, there is no guarantee that by iteratively updating the two neural networks we reach this optimum.

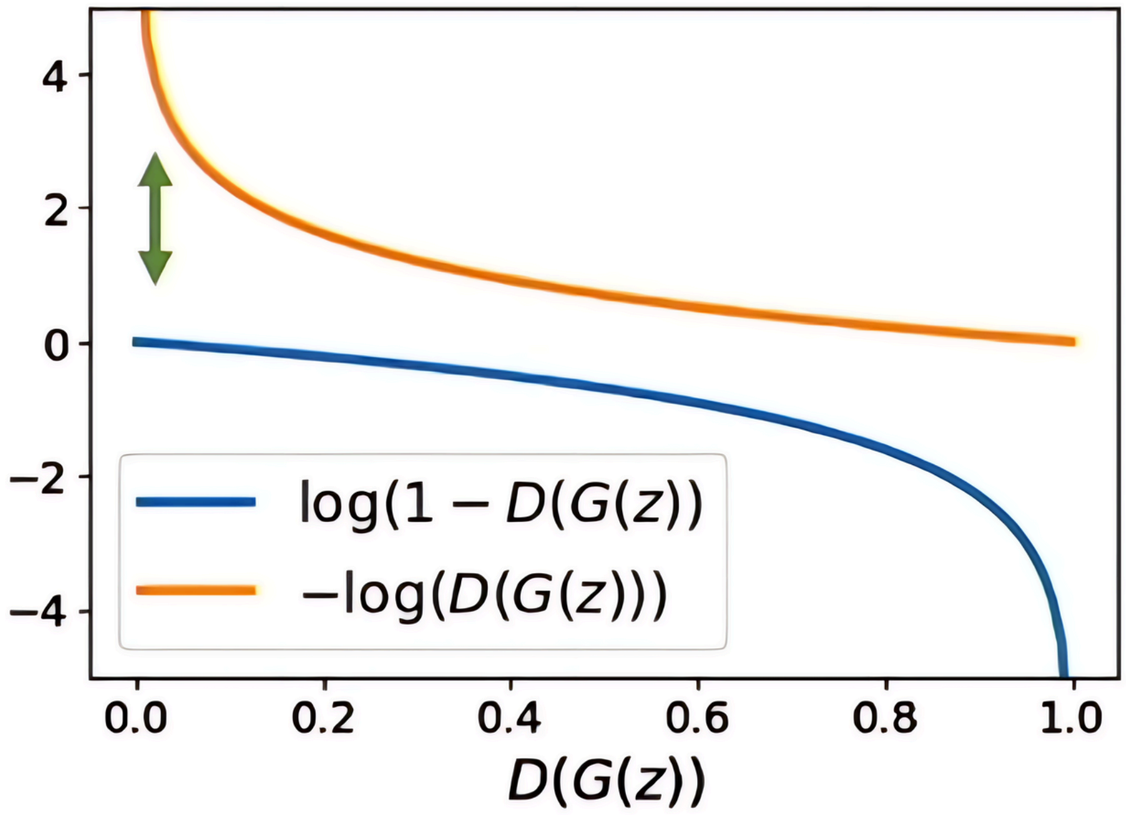
Gradient Vanishing Problem

Generating is much more difficult than classifying

- at the beginning the discriminator will perform much better then the generator
- the generator loss is flat when the generator performs poorly
- the generator gradient vanishes and training is impossible

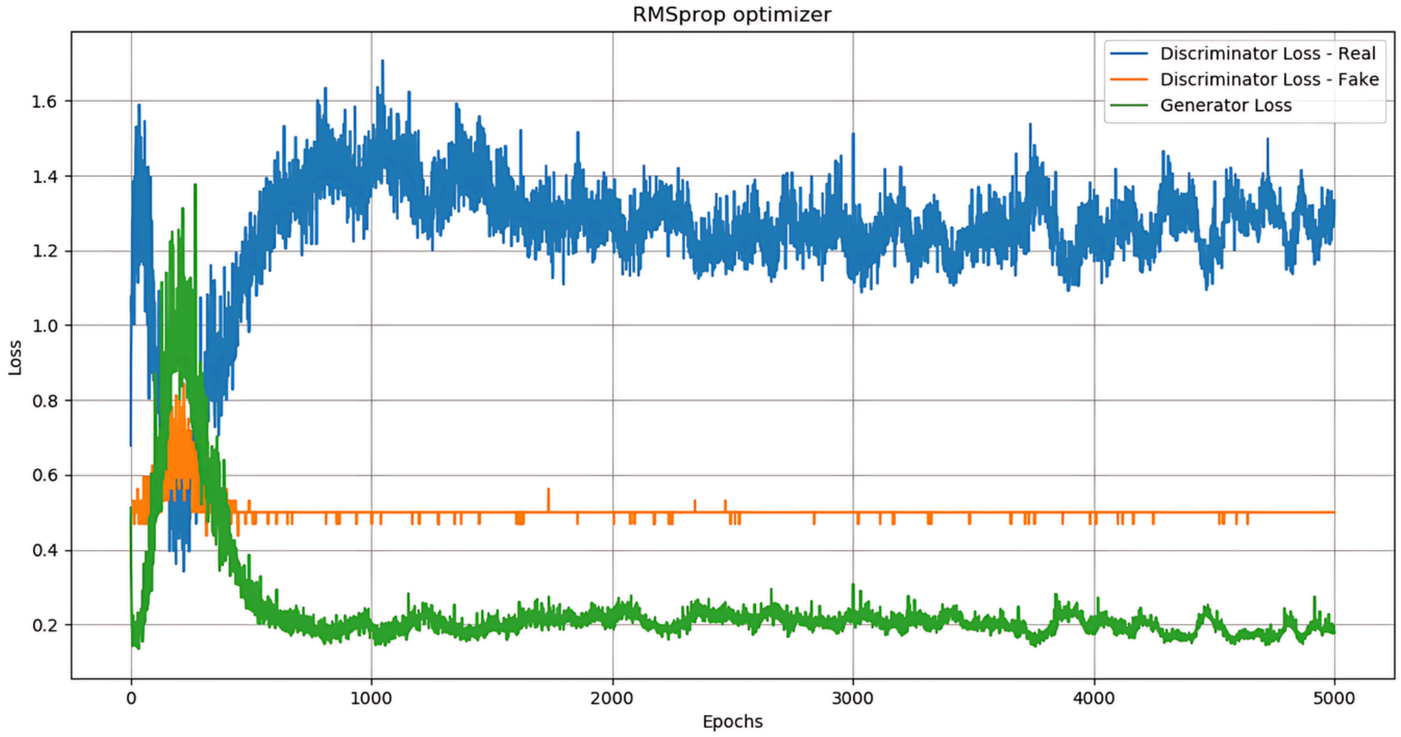
To solve this we modify the generator loss

- original loss
  $\min_G \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$
- modified (heuristic) loss
  $\min_G \mathbb{E}_{z \sim p_z(z)}[-\log D(G(z))]$

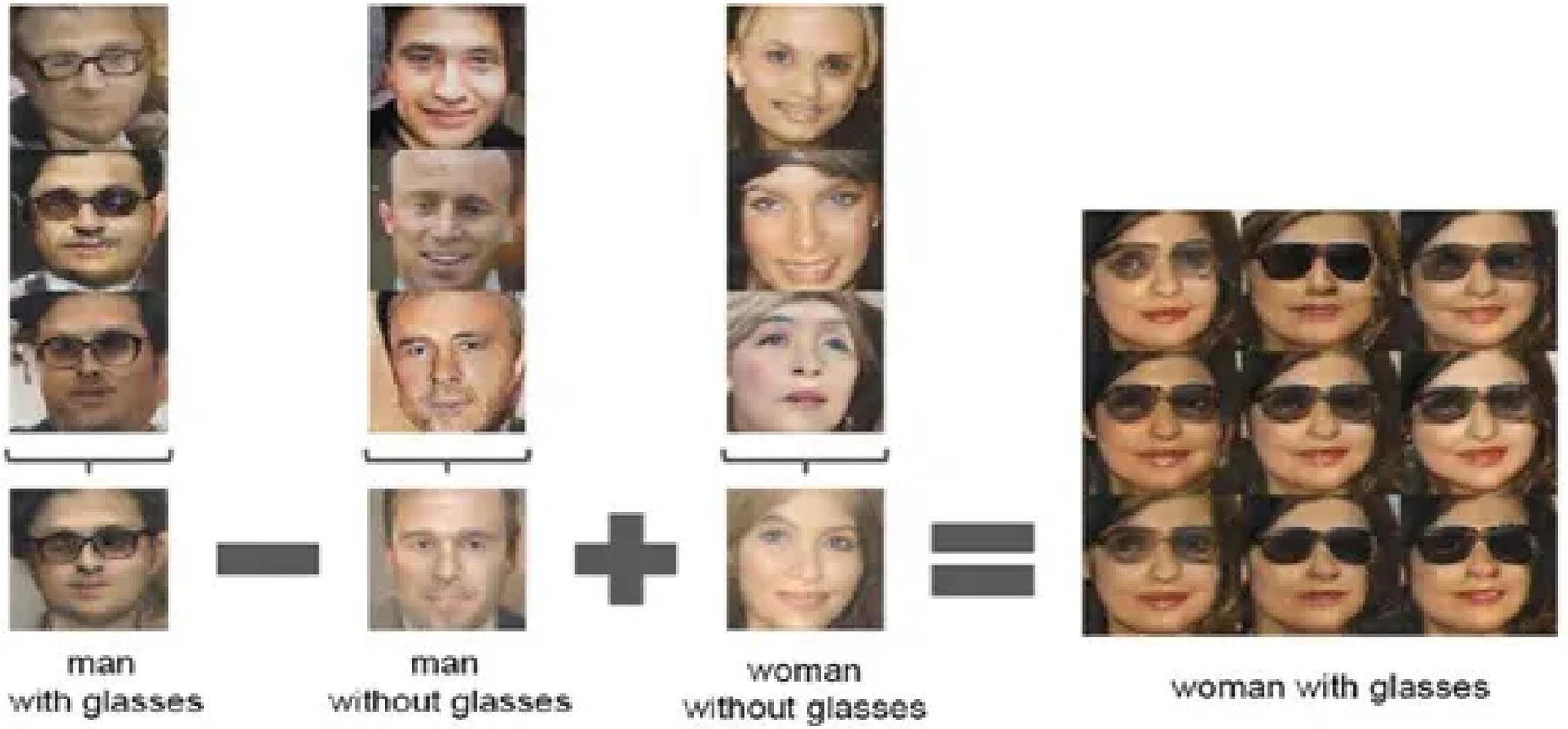


Visualizing the Loss in Training

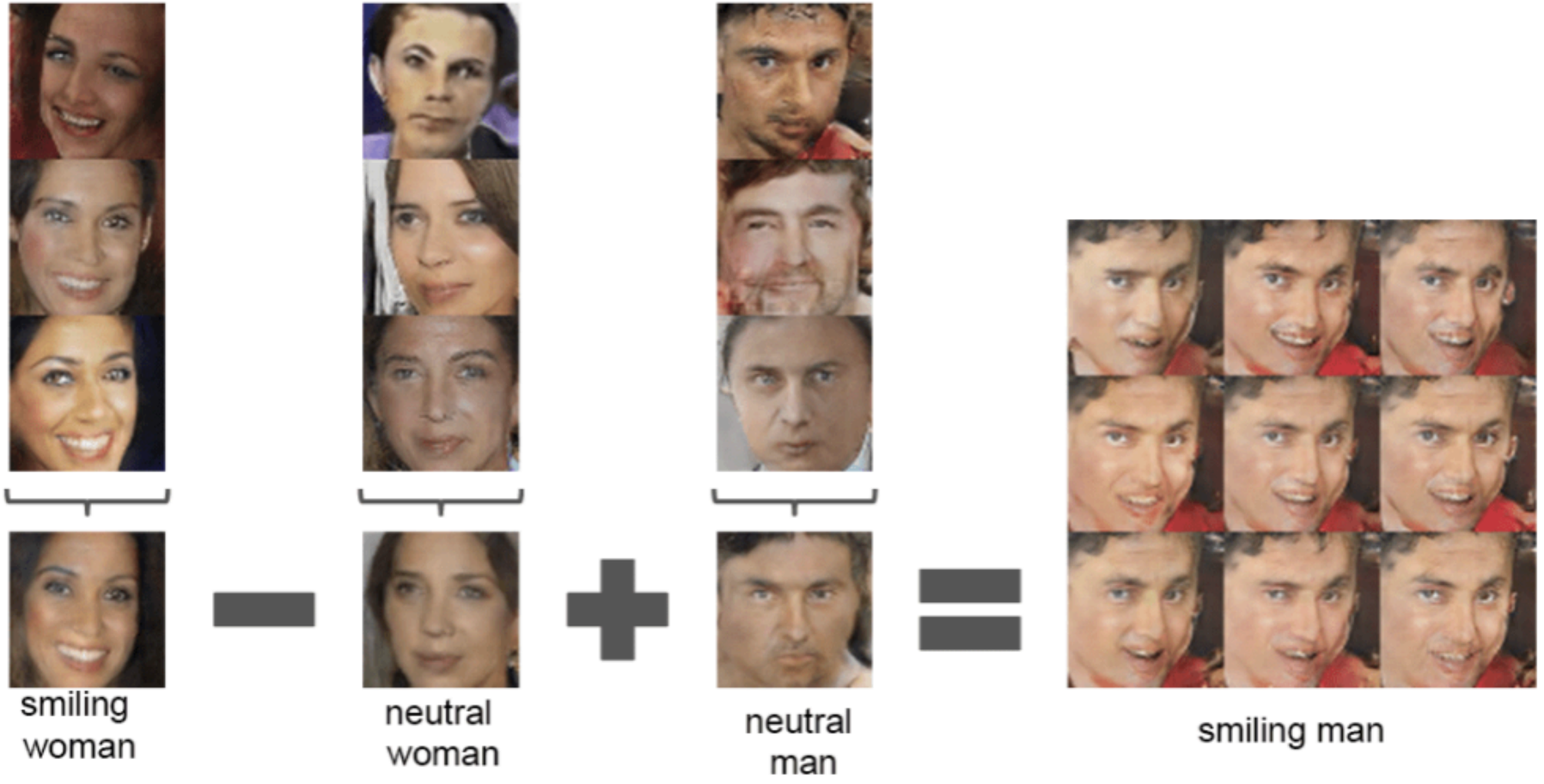
The loss of GANs is much more hard to interpret than the loss of other models. It will often show erratic fluctuations and it is hard to understand whether or not the model is learning.



Vectors in GANs



Vectors in GANs



Challenges of GANs

GANs are very elegant and conceptually simple, but training them is a nightmare:

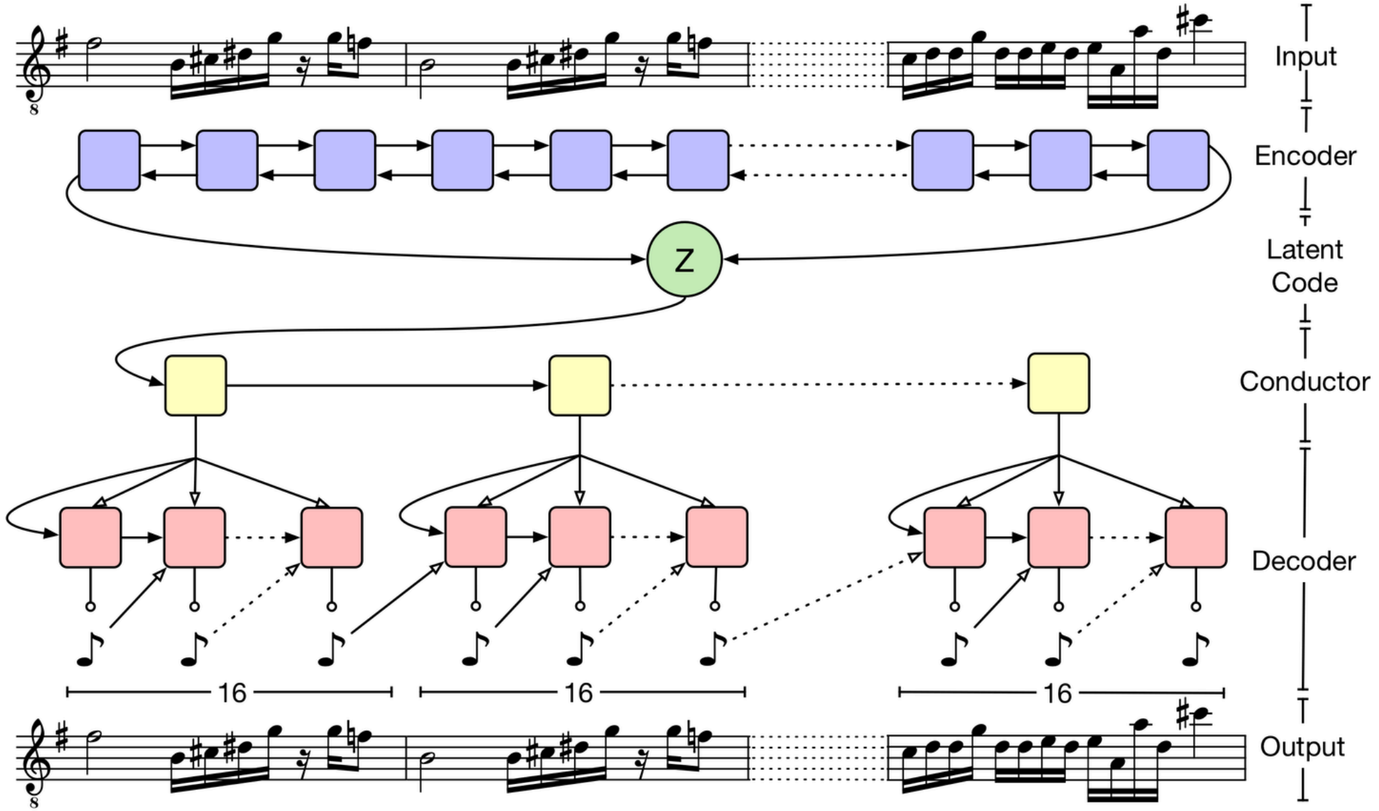
- typically generating is much more hard than discriminating, balancing discriminator and generator is a very hard task
- GANs involve two models and training two coupled models at the same time is not simple. The process is often unstable and convergence is never granted
- there is no universally accepted universal metric for evaluating GANs
- the generator may produce limited varieties of outputs, ignoring parts of the data distribution, still resulting is a good overall loss

These limitations are the main reasons why GANs are no more state of the art models, especially for image generation. In this field, diffusion based models such as Stable Diffusion or Midjourney, are now the new standard.

Universität Konstanz

Applications of Generative Deep Learning

MusicVAE

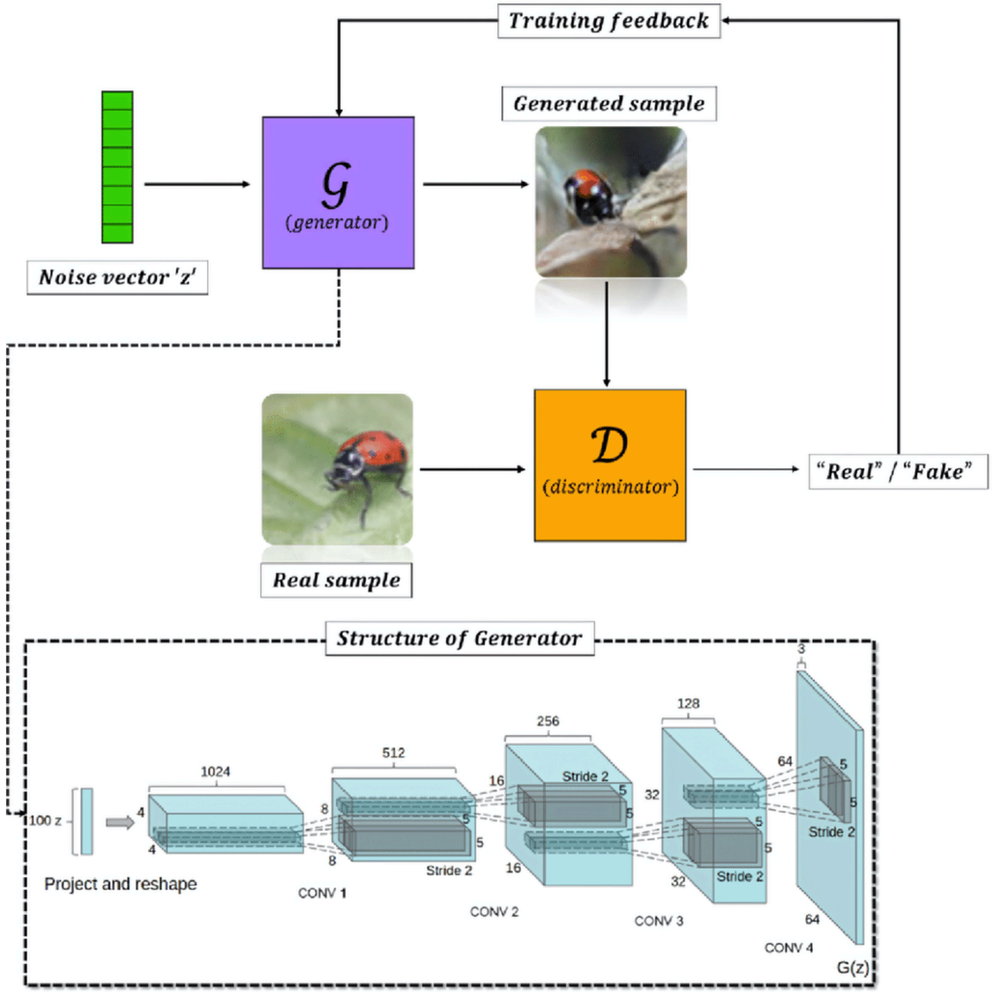


https://magenta.tensorflow.org/music-vae

DCGAN Architecture

One of the most relevant areas of application of GANs is the generation of images

- old models involved simple MLP architectures
- DCGAN improved the performances by using convolutional layers
  - the generator is a deconvolutional neural network
  - the discriminator is a neural network
- this allows to combine all the strengths of CNNs with the generative power of GANs

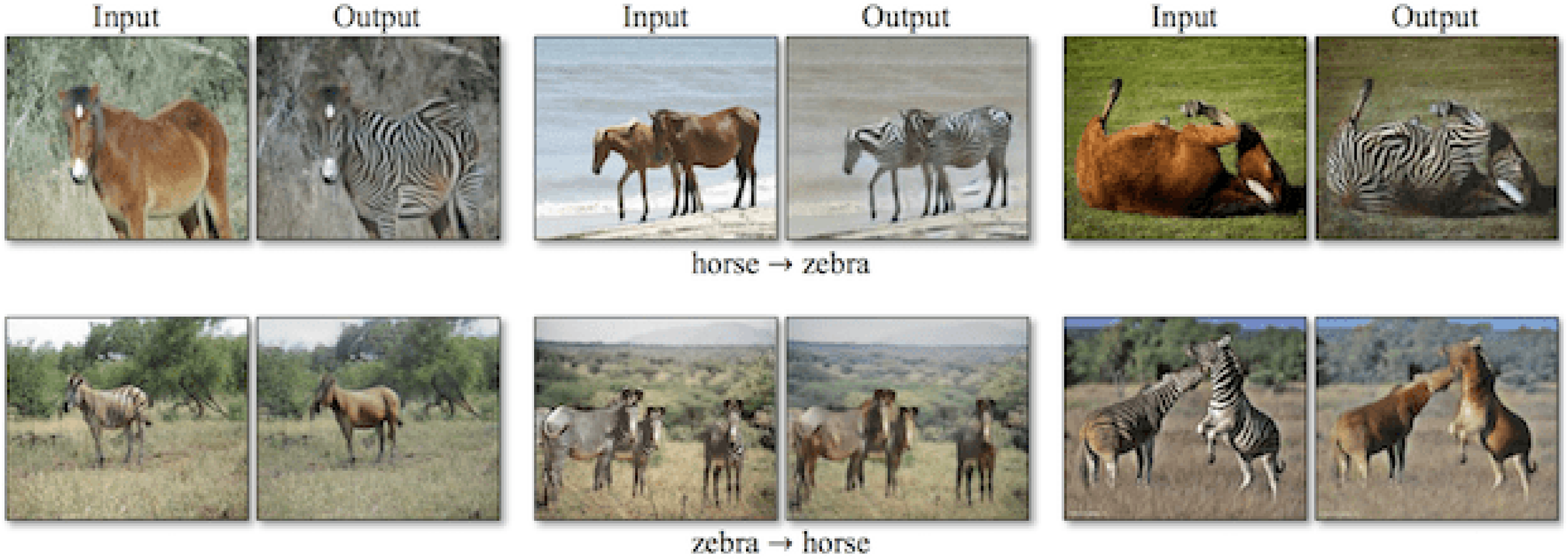


StyleGAN

StyleGAN (Nvidia, 2019) is one of the most powerful GAN model based on convolutional layers. It can generate faces and objects with different resolutions.

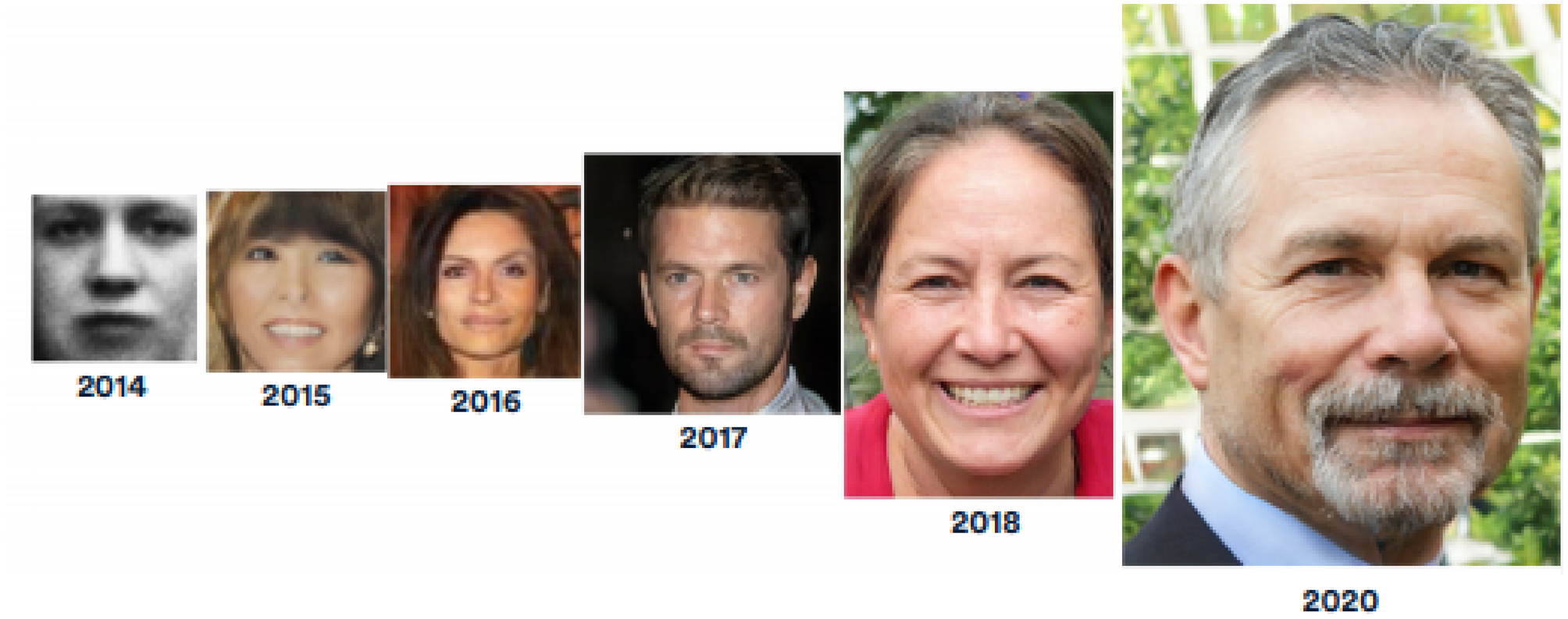
CycleGAN

By training two coupled GANs in the so called CycleGAN (2017) it is possible to perform domain transformation. In other words this corresponds to a sort of image-to-image unsupervised translation process, since we only need images from two domains, but no existing pairs.



Progress of GANs

GANs have reached incredible level of detail and resolution. Nowadays identifying AI generated GAN images of human faces is almost impossible, but there are some tricks.



Universität Konstanz

Can you Detect the AI Generated Images?

https://www.whichfaceisreal.com

https://detectfakes.kellogg.northwestern.edu

Summary

Generative Deep Learning

While discriminative deep learning reconstructs the conditional probability of the label given the data P(y|x), generative deep learning gives access to the distribution of data P(x)

Variational Autoencoders (VAEs)

VAEs are a modification of autoencoders that include a stochastic component. Their latent space is regualarized and this allows to interpolate between training data, generating new samples

Generative Adversarial Networks (GANs)

GANs consists of two models, a generator and a discriminator, that are trained together. The generator can then be used to produce artificial data.

Applications of Generative Deep Learning

VAEs and GANs find application in many fields, from music generation to images. Advanced GANs generate images that are almost impossible to distinguish from real ones