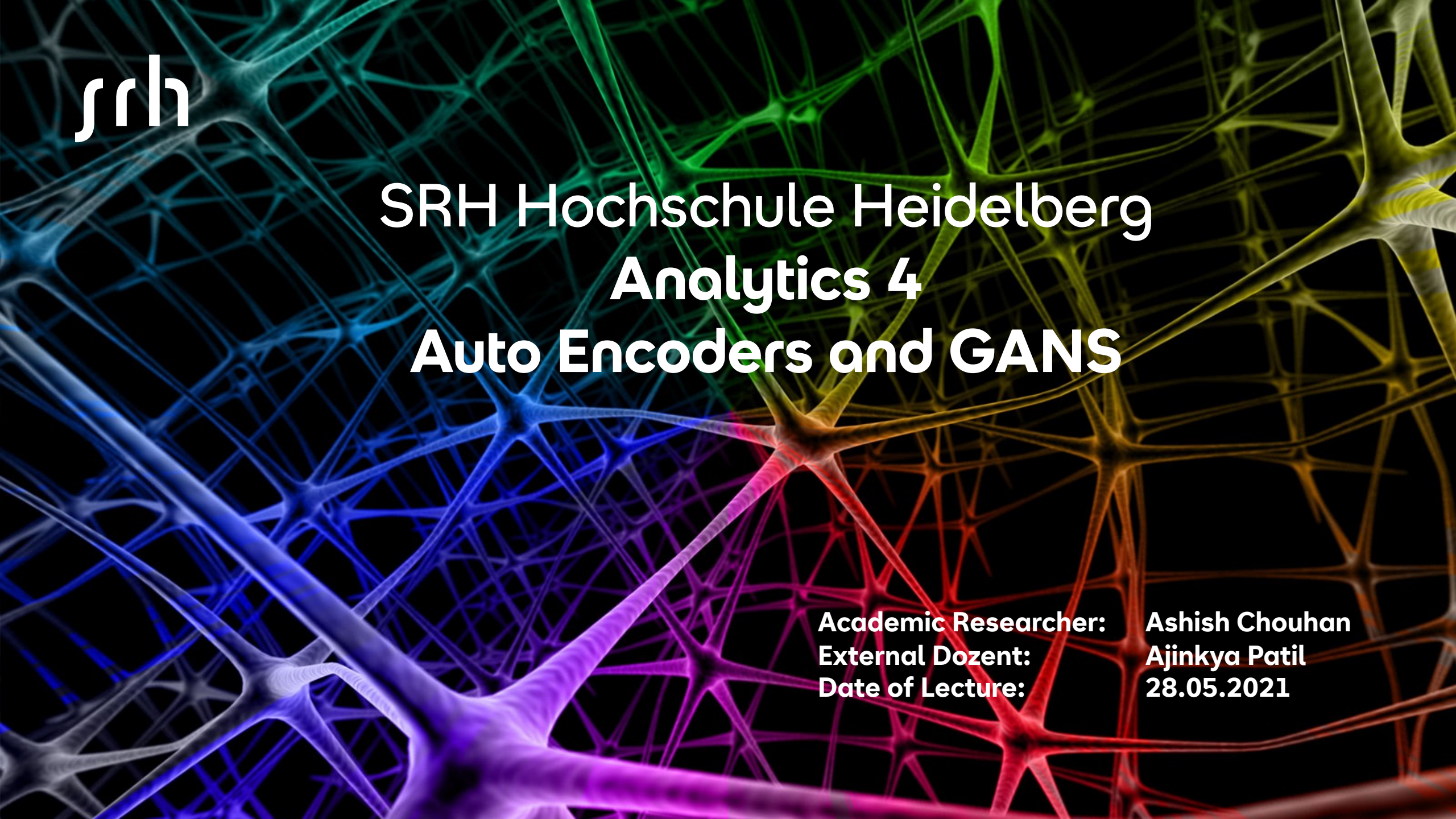


SRH Hochschule Heidelberg
Analytics 4
Auto Encoders and GANS

Academic Researcher: Ashish Chouhan
External Dozent: Ajinkya Patil

Date of Lecture: 28.05.2021

Agenda

- Supervised Vs. Unsupervised
- Generative models
- AutoEncoders (AE)
- Deep Auto Encoders (DAE)
- Denoising Auto Encoders
- Variational Auto Encoders (VAE)
- Generative Adversarial Networks (GANS)

Supervised vs unsupervised learning

Supervised Learning

- Data : (x,y)

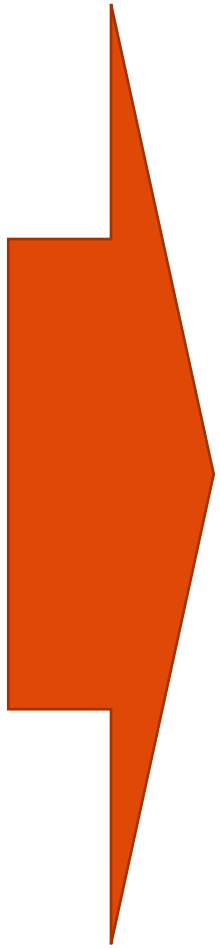
  X = data and y = label

- Goal

  Learn a function f(x) which will map x -> y

- Examples

  Classification, Regression, Object detection, etc.



Unsupervised Learning

- Data: x

  X is data, no labels

- Goal

  Learn the hidden underlying *structure* of the data

- Examples

  Clustering, Features or dimensionality reduction, etc.

Supervised vs unsupervised learning

Supervised Learning

- Data : (x,y)

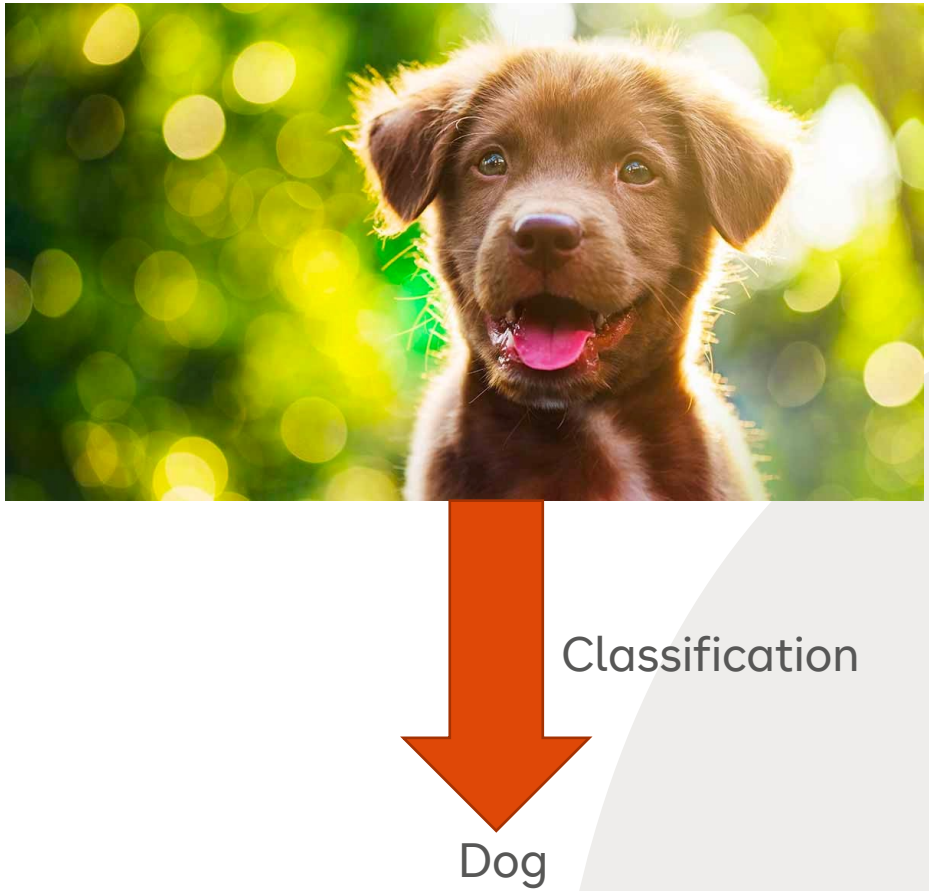
  X = data and y = label

- Goal

  Learn a function f(x) which will map x -> y

- Examples

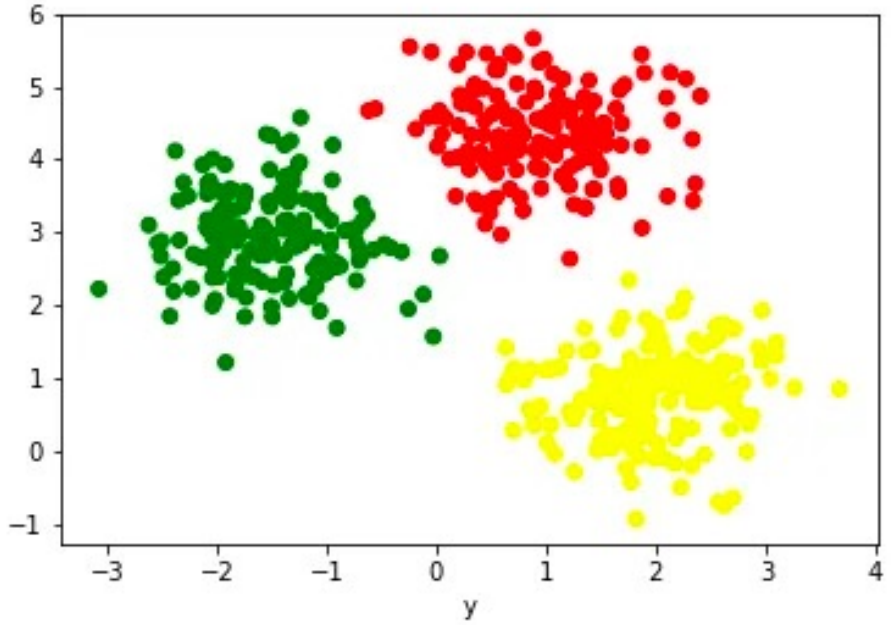
  Classification, Regression, Object detection, etc.



Supervised vs unsupervised learning

Unsupervised Learning

- Data: x
  X is data, no labels
- Goal
  Learn the hidden underlying *structure* of the data
- Examples
  Clustering, Features or dimensionality reduction, etc.



K-means clustering

Summary (Supervised vs unsupervised)

Parameters	Supervised machine learning	Unsupervised machine learning
Process	In a supervised learning model, input and output variables will be given.	In unsupervised learning model, only input data will be given
Input Data	Algorithms are trained using labeled data.	Algorithms are used against data which is not labeled
Algorithms Used	Support vector machine, Neural network, Linear and logistics regression, random forest, and Classification trees.	Unsupervised algorithms can be divided into different categories: like Cluster algorithms, K-means, Hierarchical clustering, etc.
Computational Complexity	Supervised learning is a simpler method.	Unsupervised learning is computationally complex

Summary (Supervised vs unsupervised)

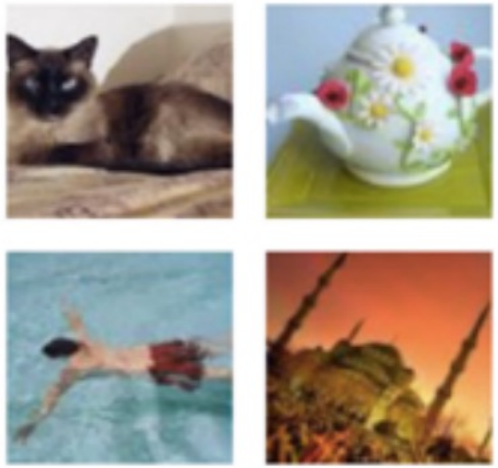
Parameters	Supervised machine learning	Unsupervised machine learning
Process	In a supervised learning model, input and output variables will be given.	In unsupervised learning model, only input data will be given
Input Data	Algorithms are trained using labeled data.	Algorithms are used against data which is not labeled
Algorithms Used	Support vector machine, Neural network, Linear and logistics regression, random forest, and Classification trees.	Unsupervised algorithms can be divided into different categories: like Cluster algorithms, K-means, Hierarchical clustering, etc.
Computational Complexity	Supervised learning is a simpler method.	Unsupervised learning is computationally complex

Generative Models

GOAL : Take as input training sample from some distribution and learn a model that represent a distribution.

How can a machine learn $P_{model}(x)$ similar to $P_{data}(x)$?

Input samples



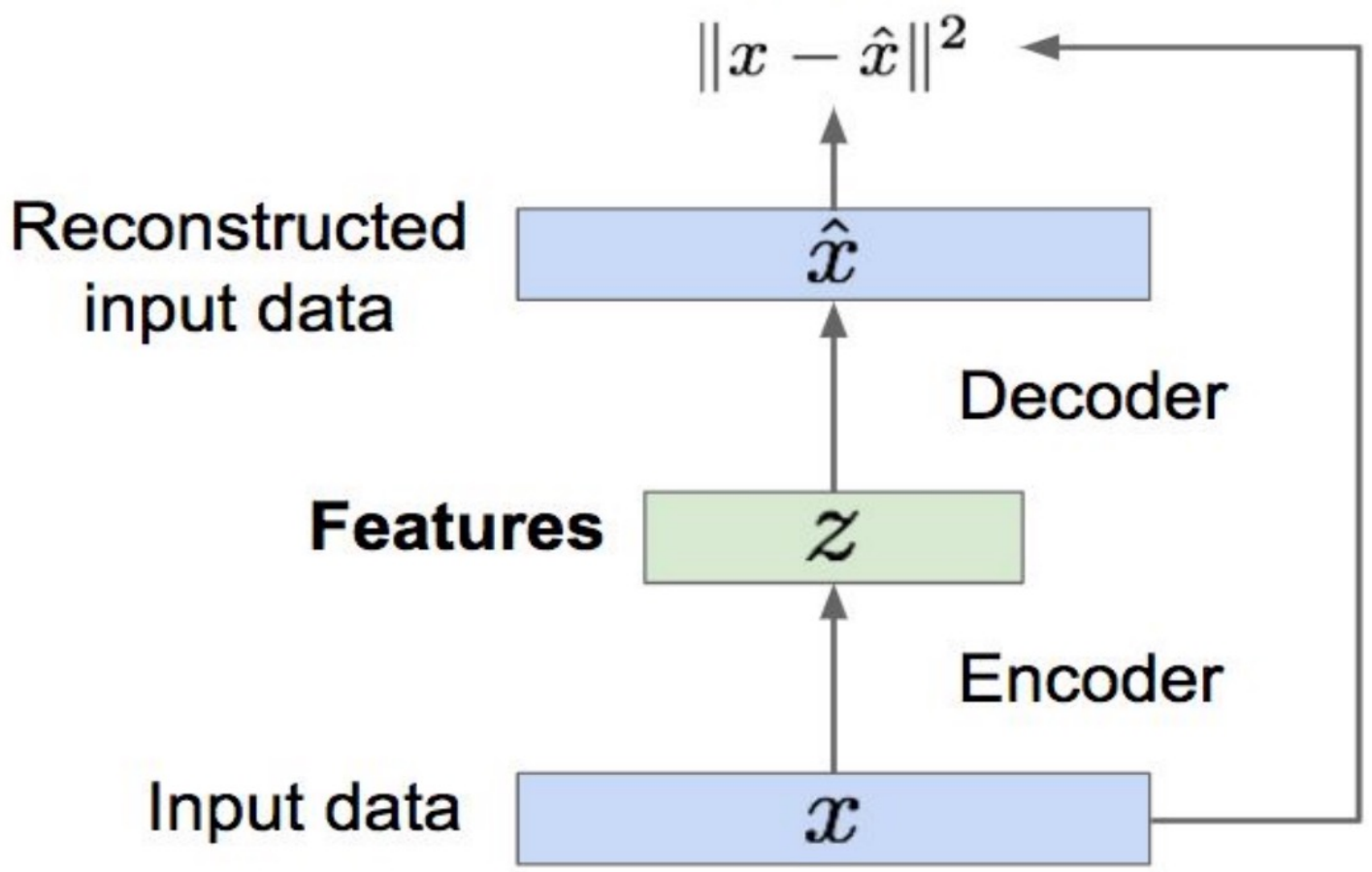
Generated samples

Training data = $P_{data}(x)$ while Generated data = $P_{model}(x)$

Auto Encoders

Unsupervised Learning

- Data: x
  X is data, no labels
- Goal
  Learn the hidden underlying *structure* of the data
- Examples
  Clustering, Features or dimensionality reduction, etc.

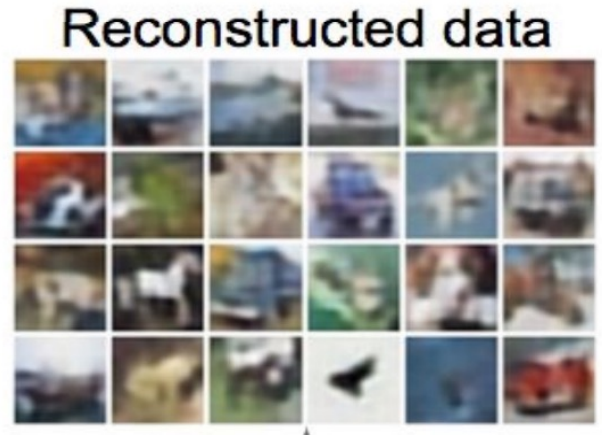


AutoEncoders (Feature Learning)

Auto Encoders

Unsupervised Learning

- Data: x
  X is data, no labels
- Goal
  Learn the hidden underlying *structure* of the data
- Examples
  Clustering, Features or dimensionality reduction, etc.



AutoEncoders (Feature Learning)

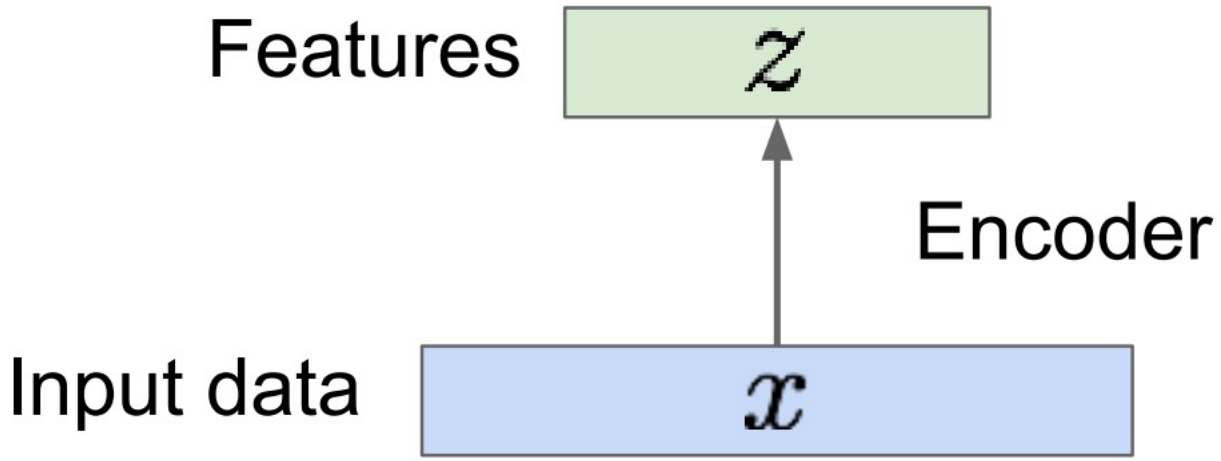
Autoencoders

Unsupervised approach for learning a lower dimensional feature representation for an unlabeled dataset.

Why feature reduction?

Z is usually smaller in structure than x (dimensionality reduction)

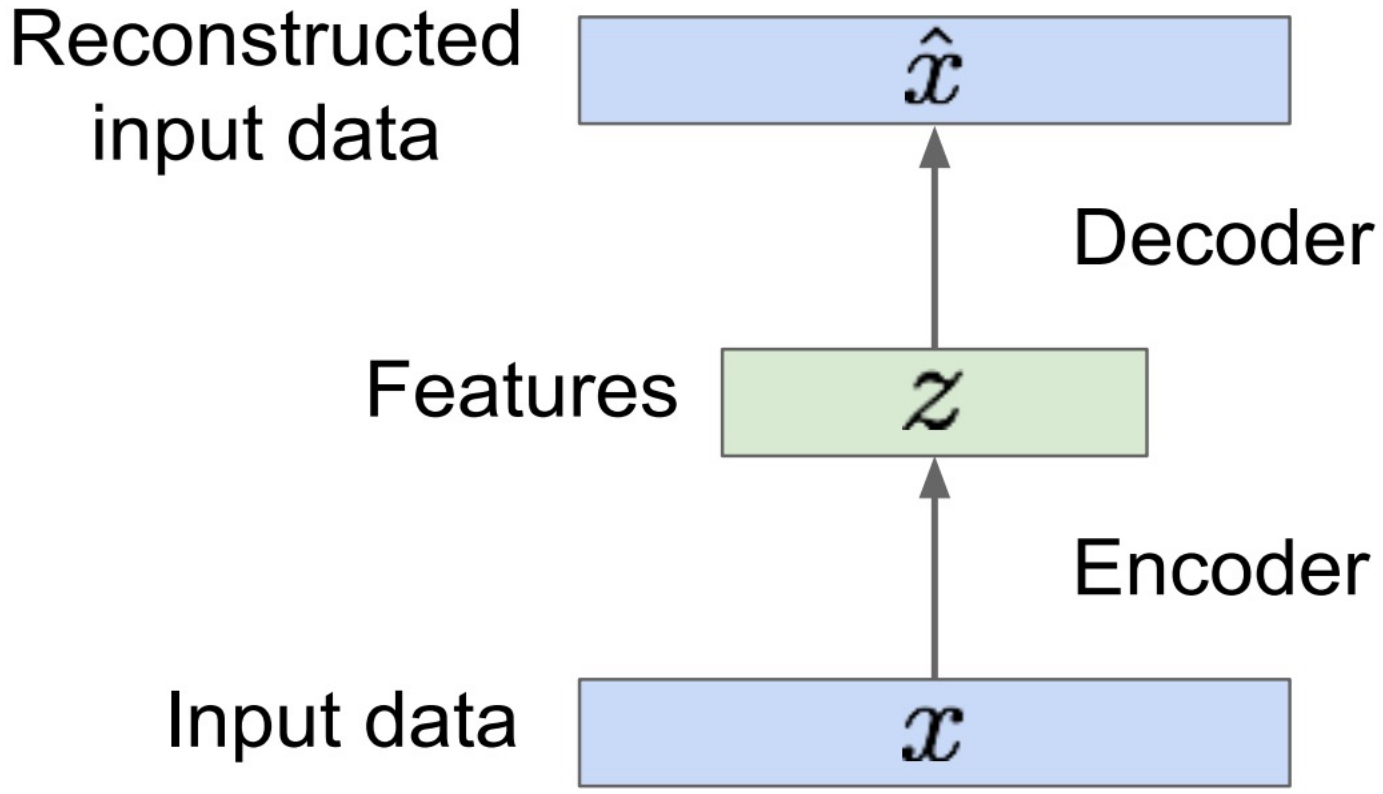
Because we want to capture the most meaningful factors of variation in data.



Autoencoders

How to learn this feature representation?

Training is done in such a way that features can be used to reconstruct original data by encoding itself -> “Auto Encoding”

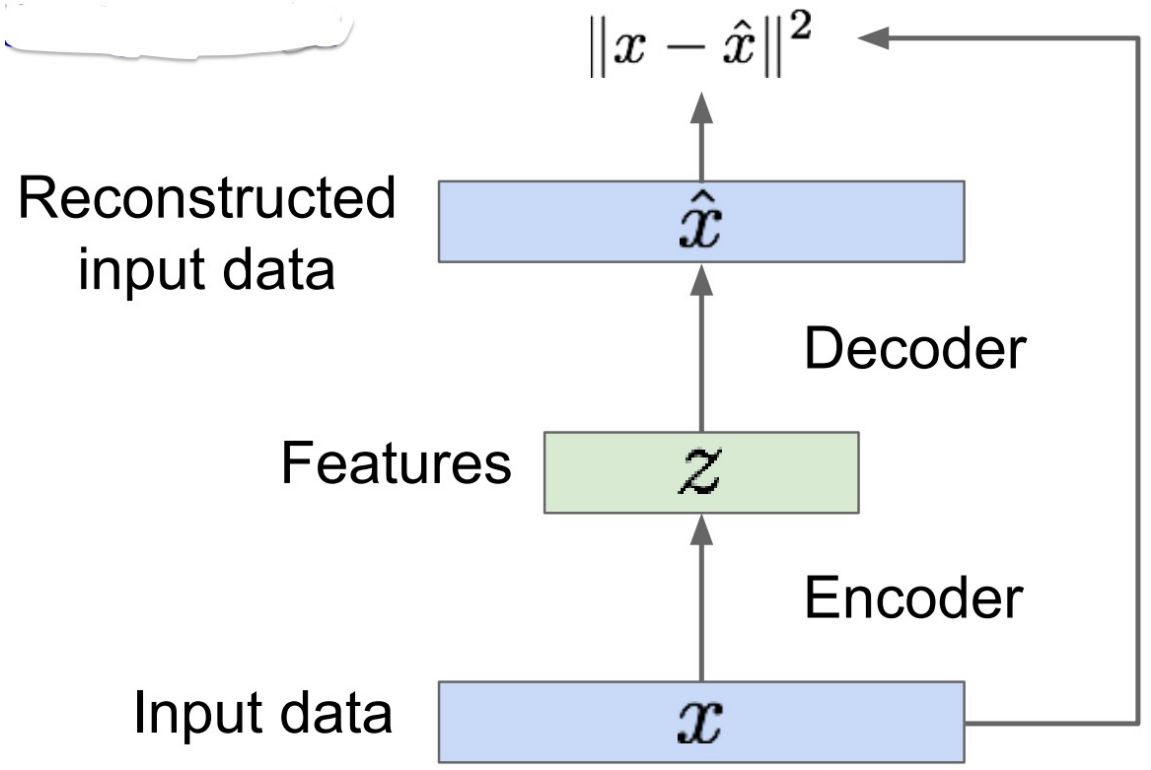


Autoencoders

- Train in such a way that features can be used to reconstruct the original data by decoder.

- The error calculated is the
- MSE (Mean Squared Error).

- Error is calculated without any
- Labels.

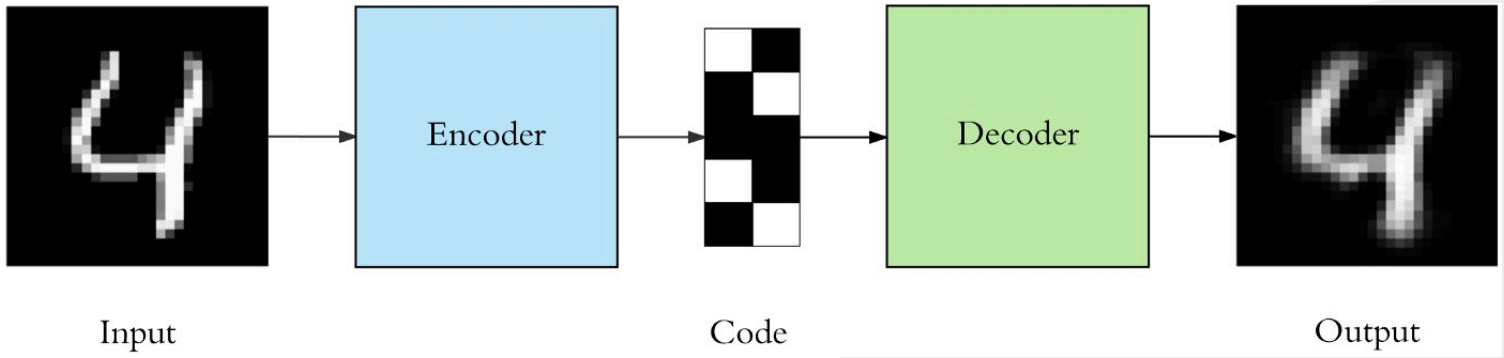


Autoencoders

Unsupervised approach for learning a lower dimensional feature representation for an unlabeled dataset.

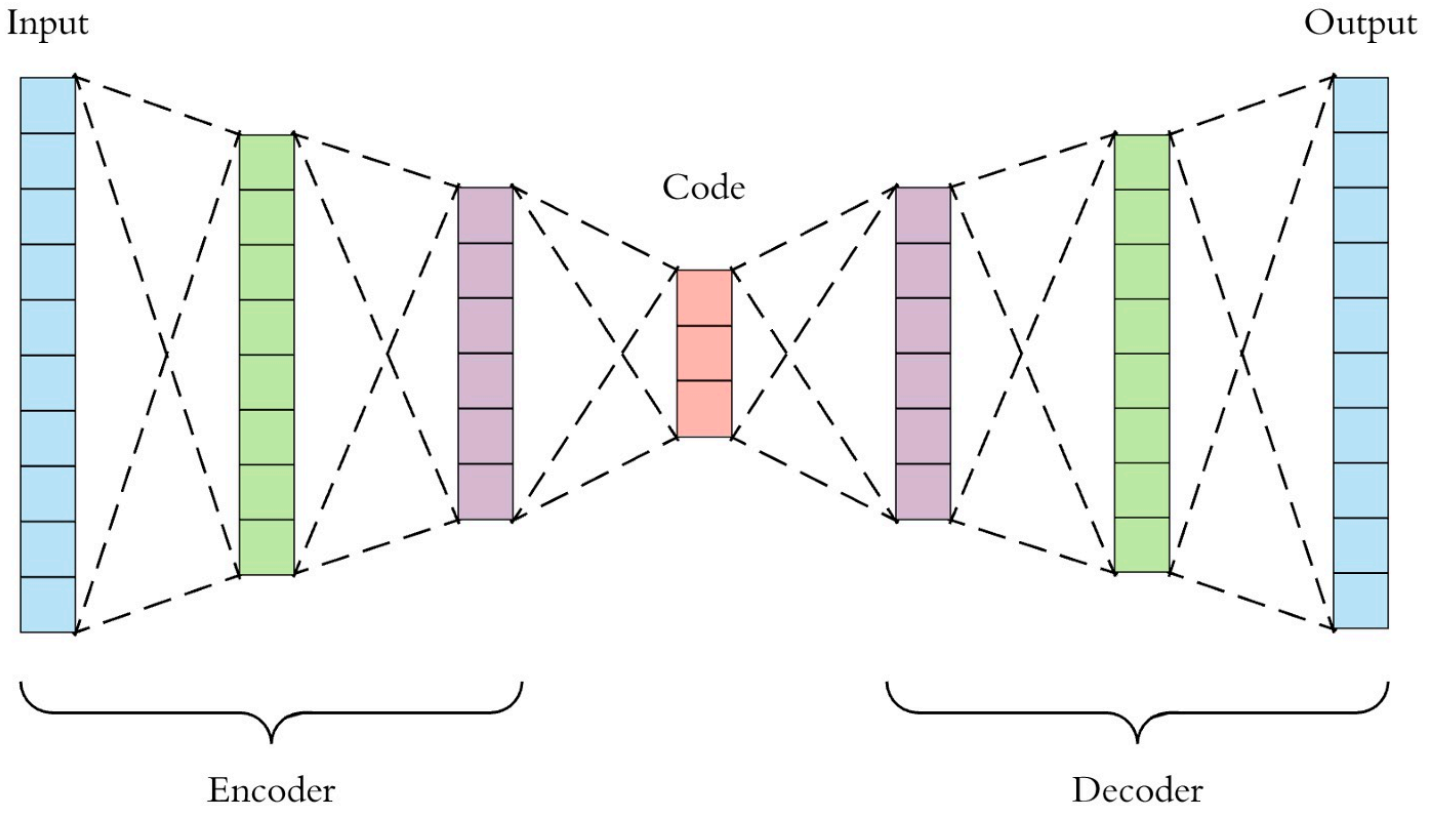
An autoencoder consists of 3 components:
1. Encoder
2. Code (Features)
3. Decoder.

The encoder compresses the input and produces the code (Features), the decoder then reconstructs the input only using this code(Features).



Deep Autoencoders

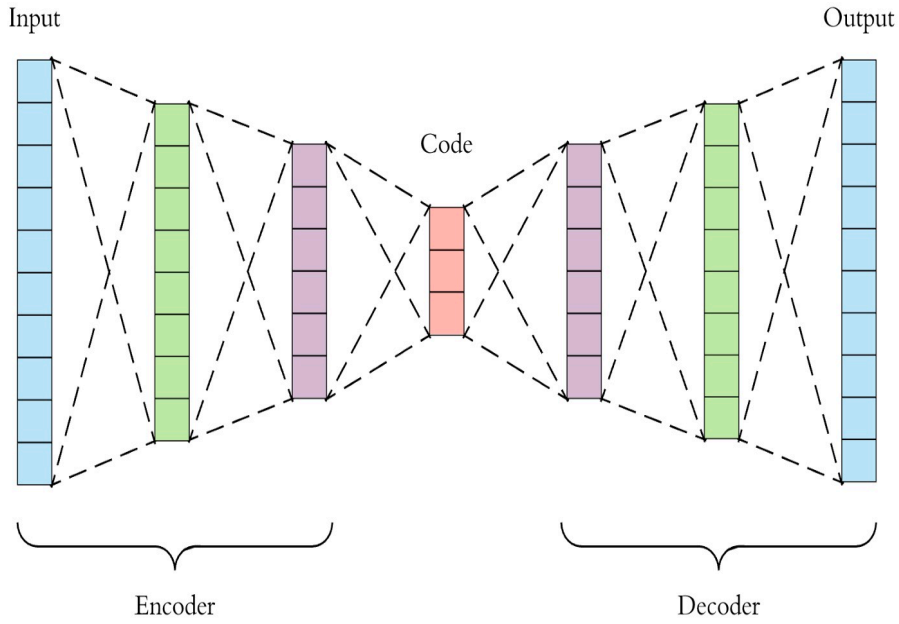
- Both the encoder and decoder are feedforward neural networks
- Code is a single layer of an ANN with the dimensionality of our choice.
- Autoencoders are trained the same way as ANNs via backpropagation.



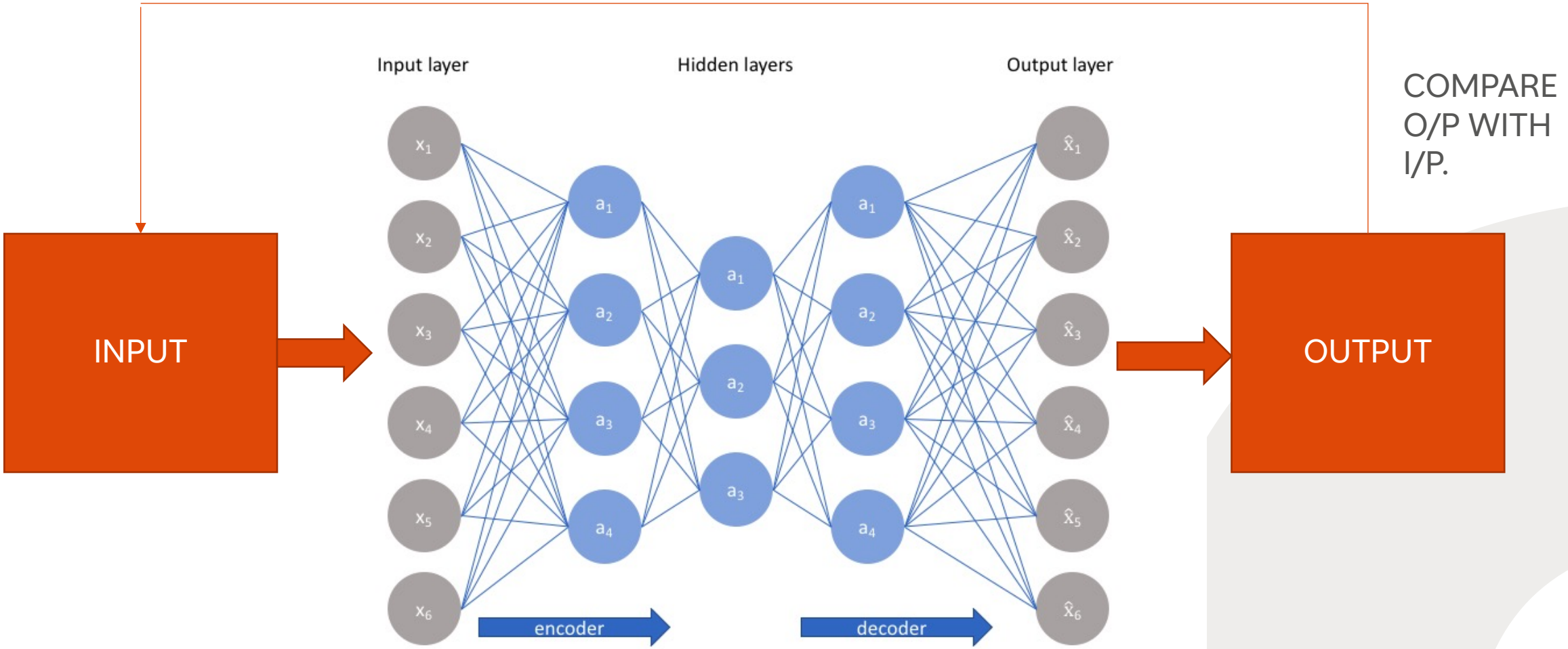
Deep Autoencoders

There are 4 hyperparameters that we need to set before training a deep autoencoder:

- Code size: number of nodes in the middle layer. Smaller size results in more compression.
- Number of layers: the autoencoder can be as deep as we like. In the figure adjacent we have 2 layers in both the encoder and decoder, without considering the input and output.
- Number of nodes per layer: Usually autoencoders look like a sandwich. The number of nodes per layer decreases with each subsequent layer of the encoder and increases back in the decoder. Also, the decoder is symmetric to the encoder in terms of layer structure. This is not always necessary.
- Loss function: we either use *mean squared error (mse)* or *binary cross entropy*. If the input values are in the range [0, 1] then we typically use cross entropy, otherwise we use the mean squared error.

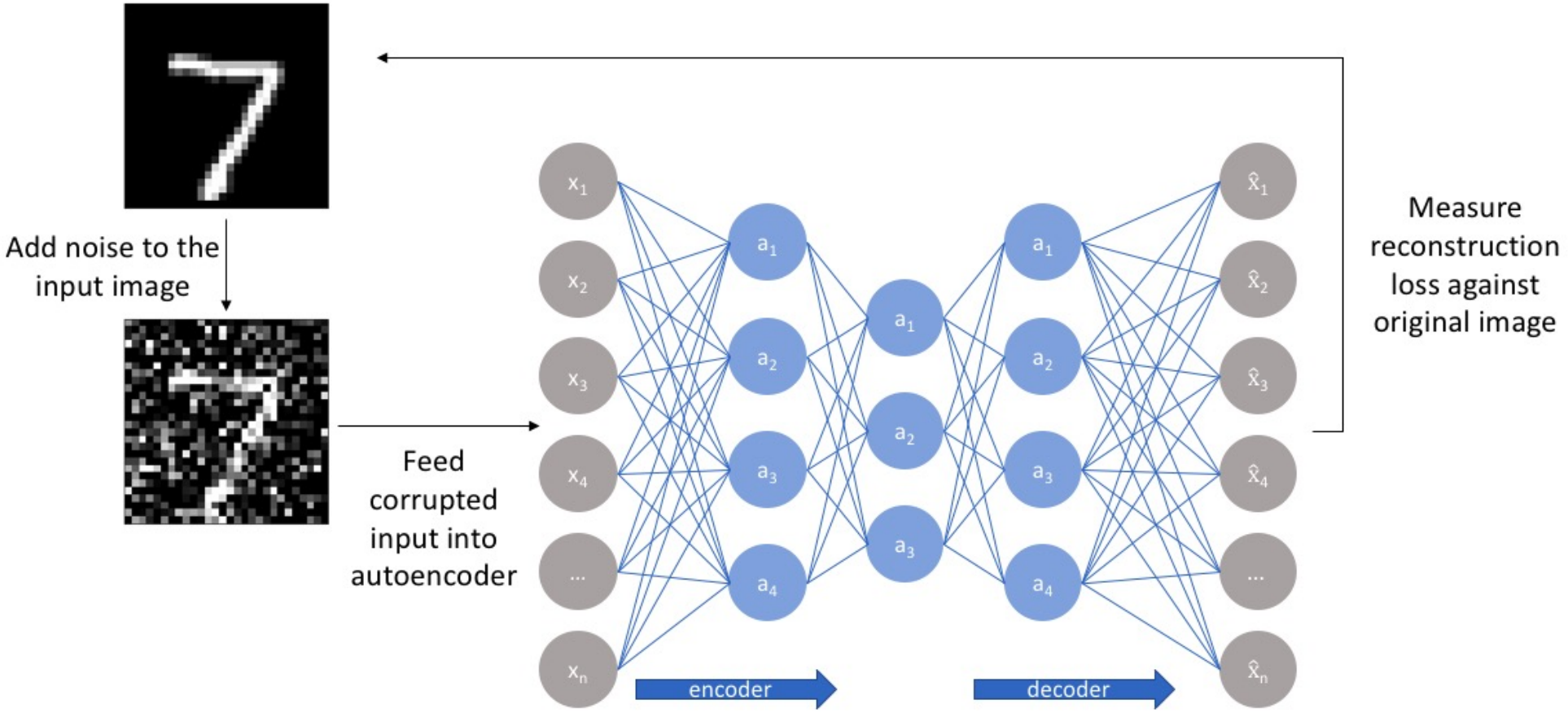


DEEP AUTO ENCODER



DEMO

DENOISING AUTOENCODERS



What could be “Noise”?

The term “noise” here could be:

- Produced by a faulty or poor-quality image sensor
- Random variations in brightness or color
- Quantization noise
- Poor paper quality (crinkles and folds) when trying to perform OCR

DEMO

Assignment

Optical Character Recognition (OCR) is the process of getting type or handwritten documents into a digitized format. If you've read a classic novel on a digital reading device or had your doctor pull up old healthcare records via the hospital computer system, you've probably benefited from OCR.
OCR makes previously static content editable, searchable, and much easier to share. But, a lot of documents eager for digitization are being held back. Coffee stains, faded sun spots, dog-eared pages, and lot of wrinkles are keeping some printed documents offline and in the past.

https://www.kaggle.com/c/denoising-dirty-documents/

Applications of Deep Auto Encoders

- Dimensionality Reduction
- Data compression (not recommended, as the compression is lossy)
- Image denoising
- Feature Extraction
- Recommendation Systems

Latest Space

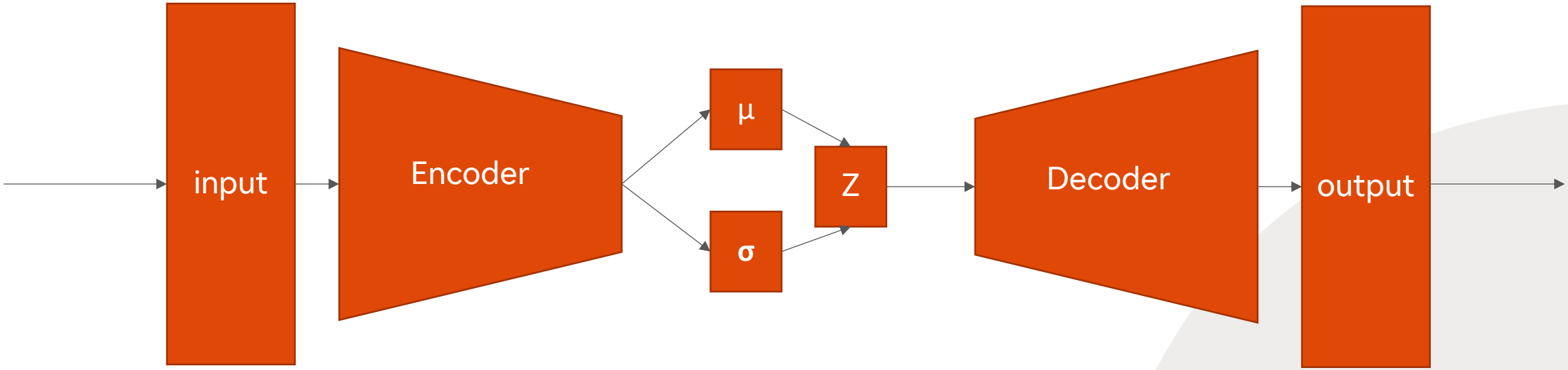
- Before you use a neural network for a task (classification, regression, image reconstruction), the usual architecture is to extract features through many layers (convolutional, recurrent, pooling etc.) which maps it to a n-dimensional hyperspace aka latent space. In other words, the latent space is the space where your features exist.

MNIST in Latent Space (2-D)

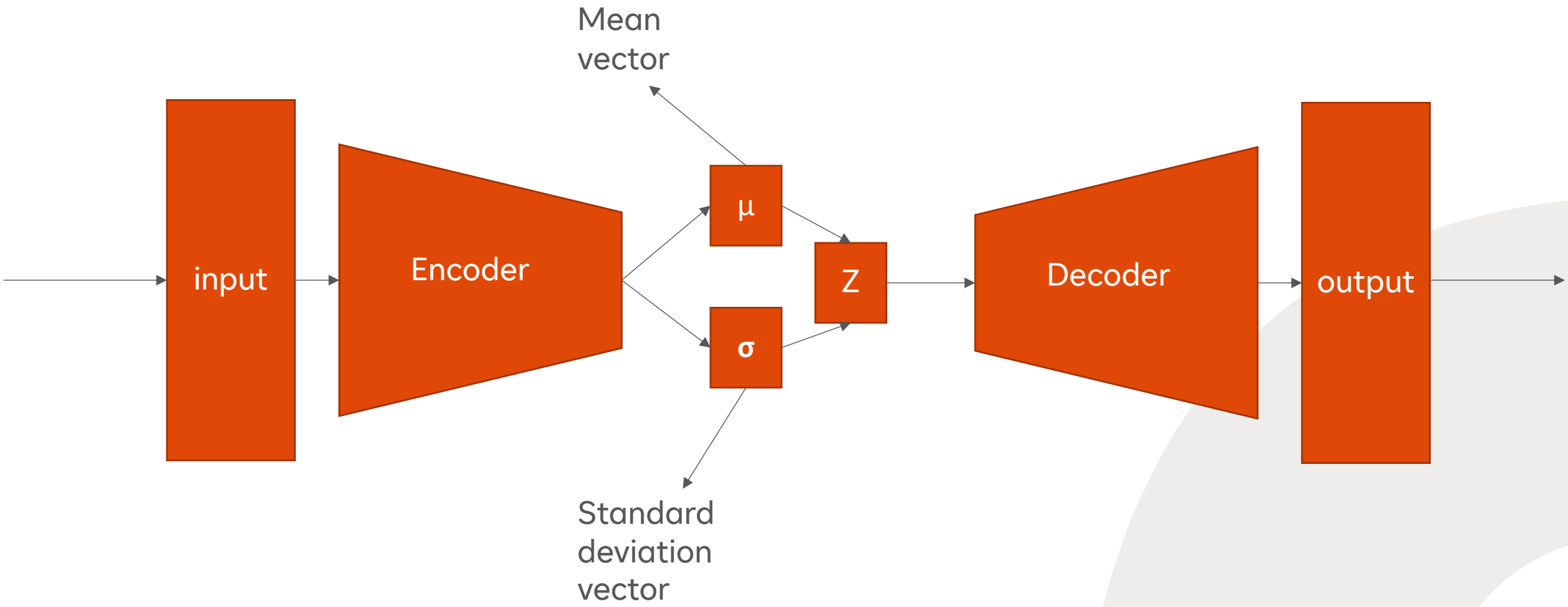
Variational Autoencoders

- Another category of Autoencoder introduced by Diederik Kingma and Max Welling in 2014 in their paper "Auto-Encoding Variational Bayes"
- They are *probabilistic autoencoders* which means their outputs are partially determined by chance.
- They are generative autoencoders, meaning they can generate new instances from the sample they were trained on.

Variational Autoencoders

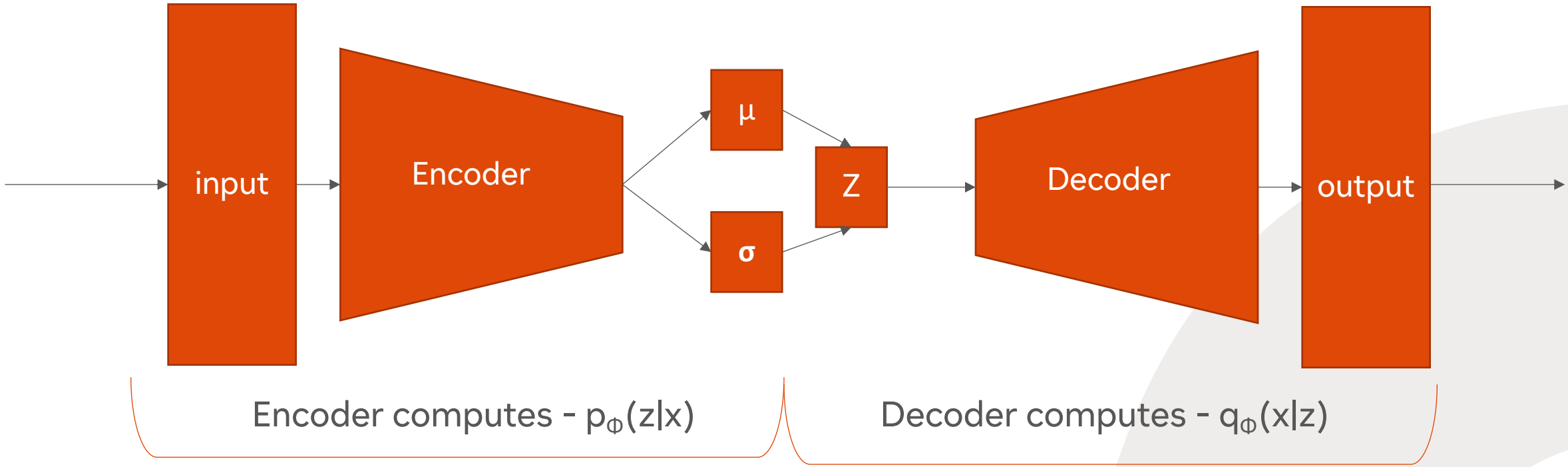


Variational Autoencoders



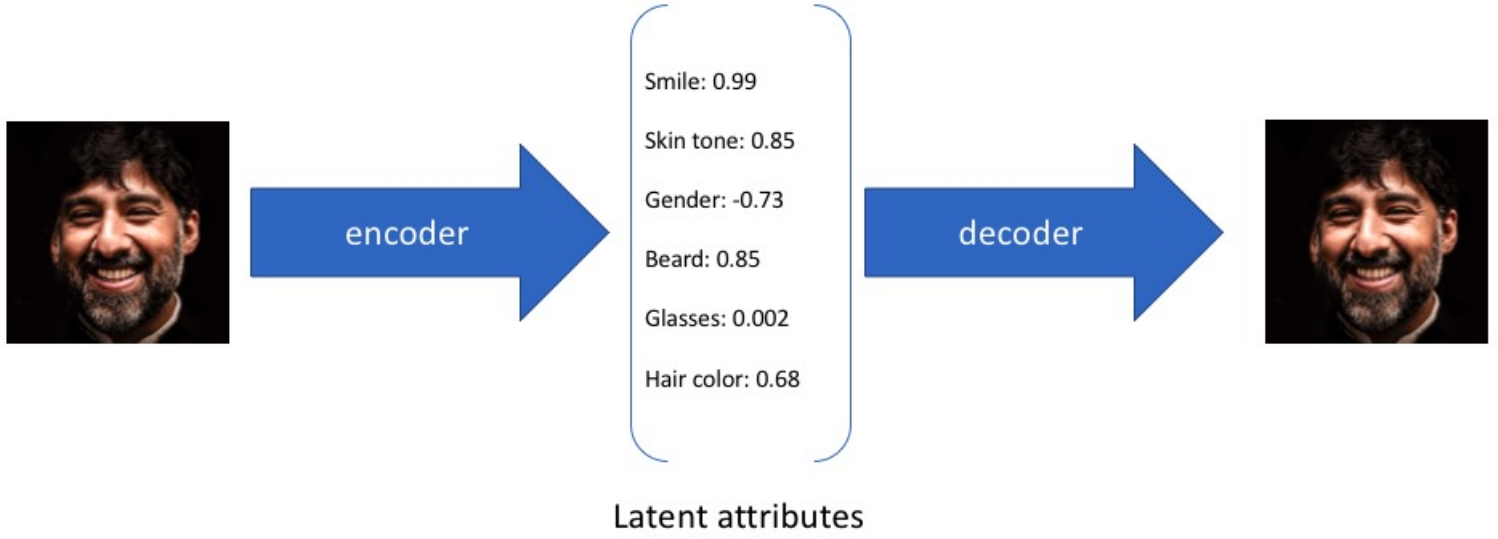
Variation DAEs are similar to DAE but with the probabilistic twist on the feature layer.

Variational Autoencoders



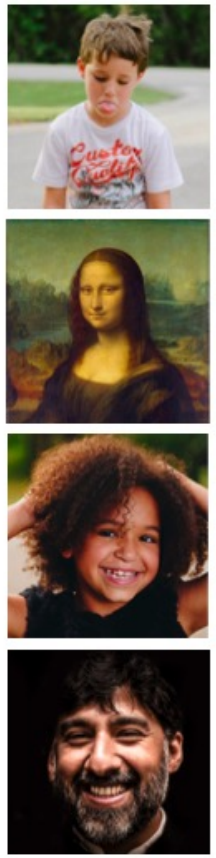
Variational AutoEncoders

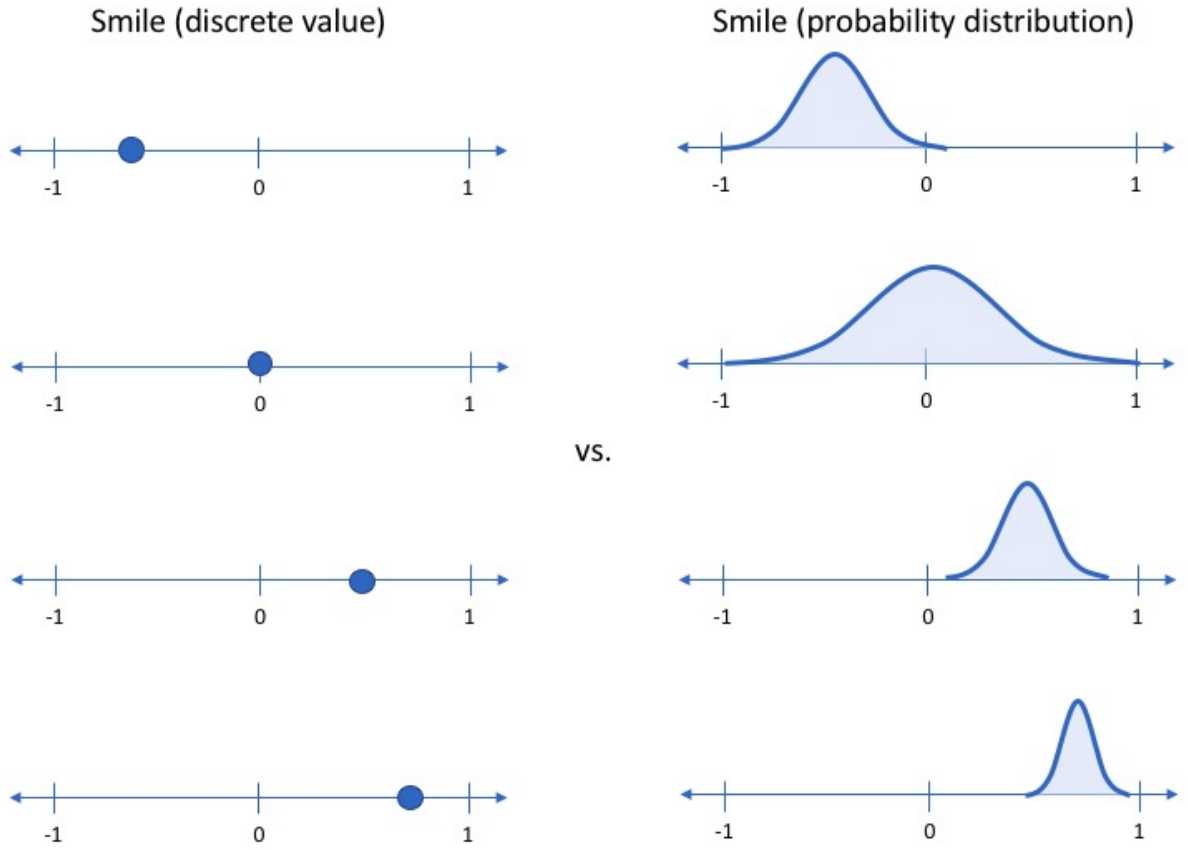
- How it would look in a normal DAE.
- All the features will have a simple feature vector and fixed values.



Variational AutoEncoders

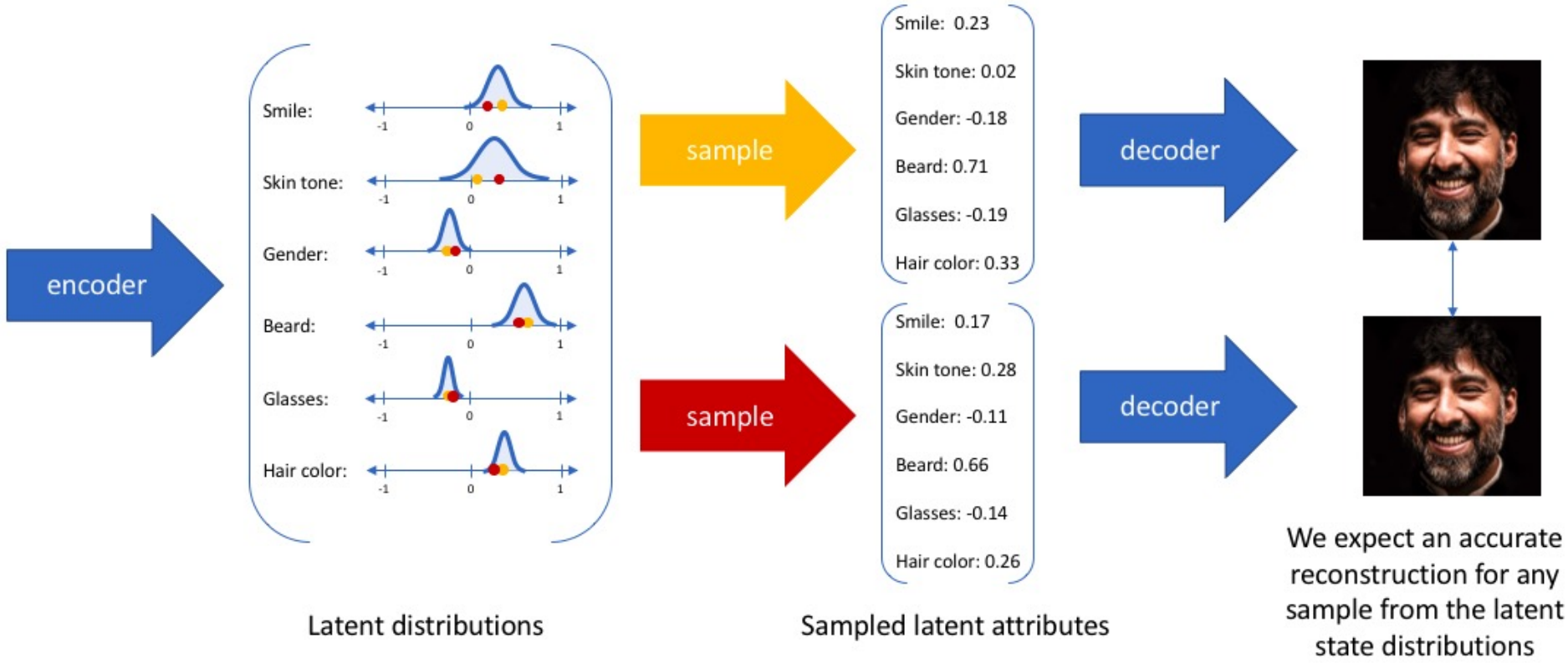
- With VDAE, Latent features exists on a probability distrubution.
- While decoding we will create samples from eact state's probability to generate vector as inputs for the deoder.





Variational Autoencoders

For each feature we calculate a probability distrubtion, e.g. smile or no smile, glasses or no glasses.



Variants of Autoencoders

1. Contractive AutoEncoder
2. Stacked Convolutional AutoEncoders
3. Generative Stochastic network
4. Winner Take all Autoencoder
5. Adversarial Autoencoders

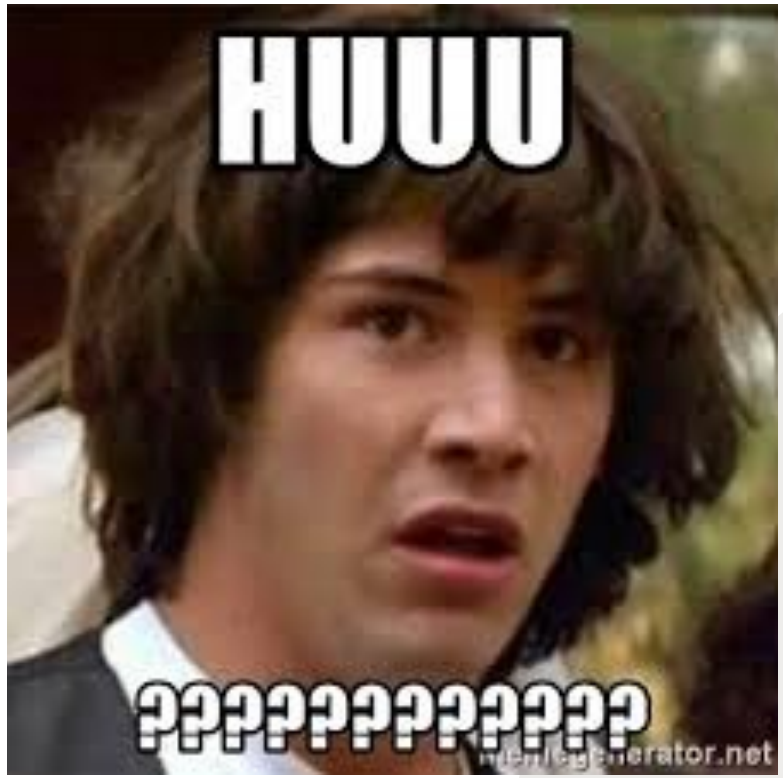
VAE Summary

- Compress the representation of inputs to interpret the latent distribution
- Reconstruction allows for unsupervised learning (No Labels)
- Generate new samples
- Allows reconstruction from latent distribution

VAE Question

- What if, there is another way to do this without having to go too much in depth of the latent space representation?

Answer - Generative Adversarial Networks (GANs)
GANS do not work with any explicit density function or
Instead, take a game-theoretic approach: learn to generate data from training sample throught a 2-player game.



GANS Introduction

- A **generative adversarial network** (**GAN**) is a class of machine learning frameworks invented by Ian Goodfellow and his colleagues in 2014 in this paper.

- Two neural networks contest with each other in a game (in the sense of game theory).

- Given a training set, this technique learns to generate new data with the same statistics as the training set.

- A GAN trained on photographs can generate new photographs that look at least superficially authentic to human observers, having many realistic characteristics.

Overview of gans

Informally:

- **Generative** models can generate new data instances.
- **Discriminative** models discriminate between different kinds of data instances.

A generative adversarial network (GAN) has two parts:

- The **generator** learns to generate plausible data. The generated instances become negative training examples for the discriminator.
- The **discriminator** learns to distinguish the generator's fake data from real data. The discriminator penalizes the generator for producing implausible results.

Overview of gans

- A generative model includes the distribution of the data itself, and tells you how likely a given example is.
- For example, models that predict the next word in a sequence are typically generative models (usually much simpler than GANs) because they can assign a probability to a sequence of words.
- A discriminative model ignores the question of whether a given instance is likely, and just tells you how likely a label is to apply to the instance.
- Note that this is a very general definition. There are many kinds of generative model. GANs are just one kind of generative model.

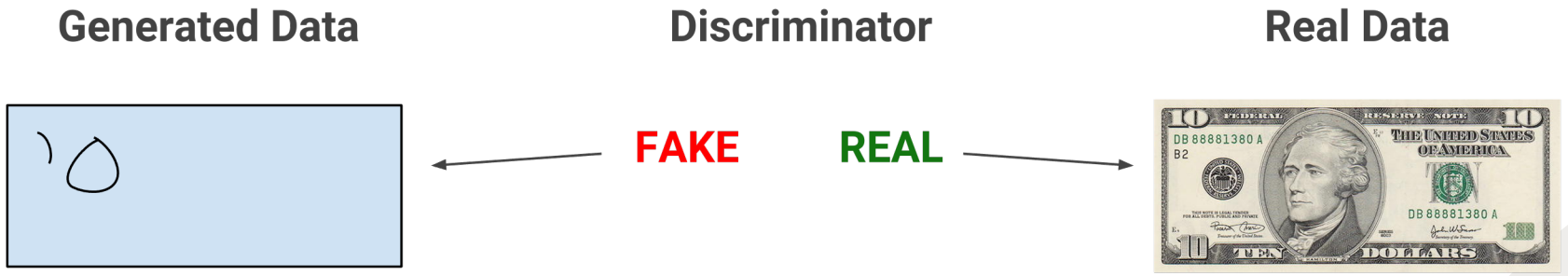
Overview of gans

- A generative model for images might capture correlations like "things that look like boats are probably going to appear near things that look like water" and "eyes are unlikely to appear on foreheads." These are very complicated distributions.

- In contrast, a discriminative model might learn the difference between "sailboat" or "not sailboat" by just looking for a few tell-tale patterns. It could ignore many of the correlations that the generative model must get right.

Overview of gans

When training begins, the generator produces obviously fake data, and the discriminator quickly learns to tell that it's fake:



As training progresses, the generator gets closer to producing output that can fool the discriminator

Overview of GANS

Finally, if generator training goes well, the discriminator gets worse at telling the difference between real and fake. It starts to classify fake data as real, and its accuracy decreases.

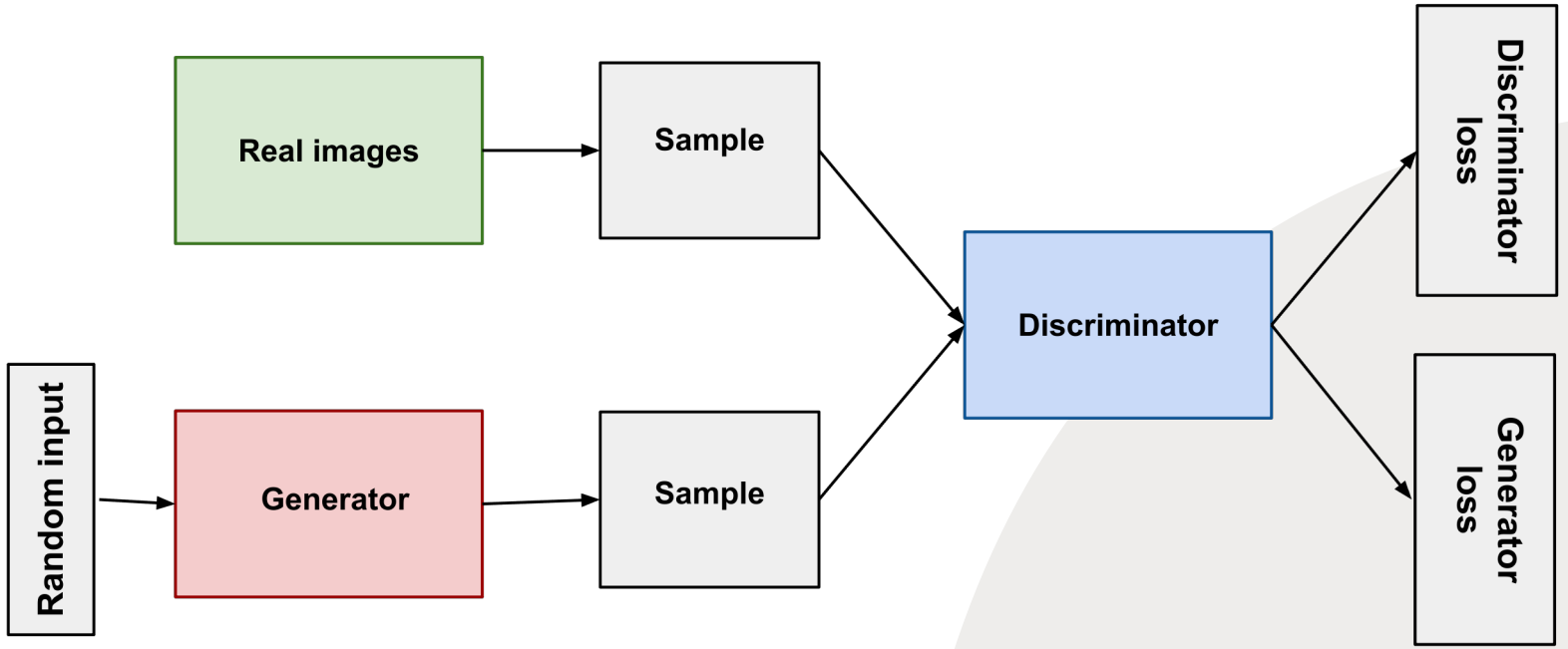
REAL

REAL

Overview of gans structure

A picture of the whole system

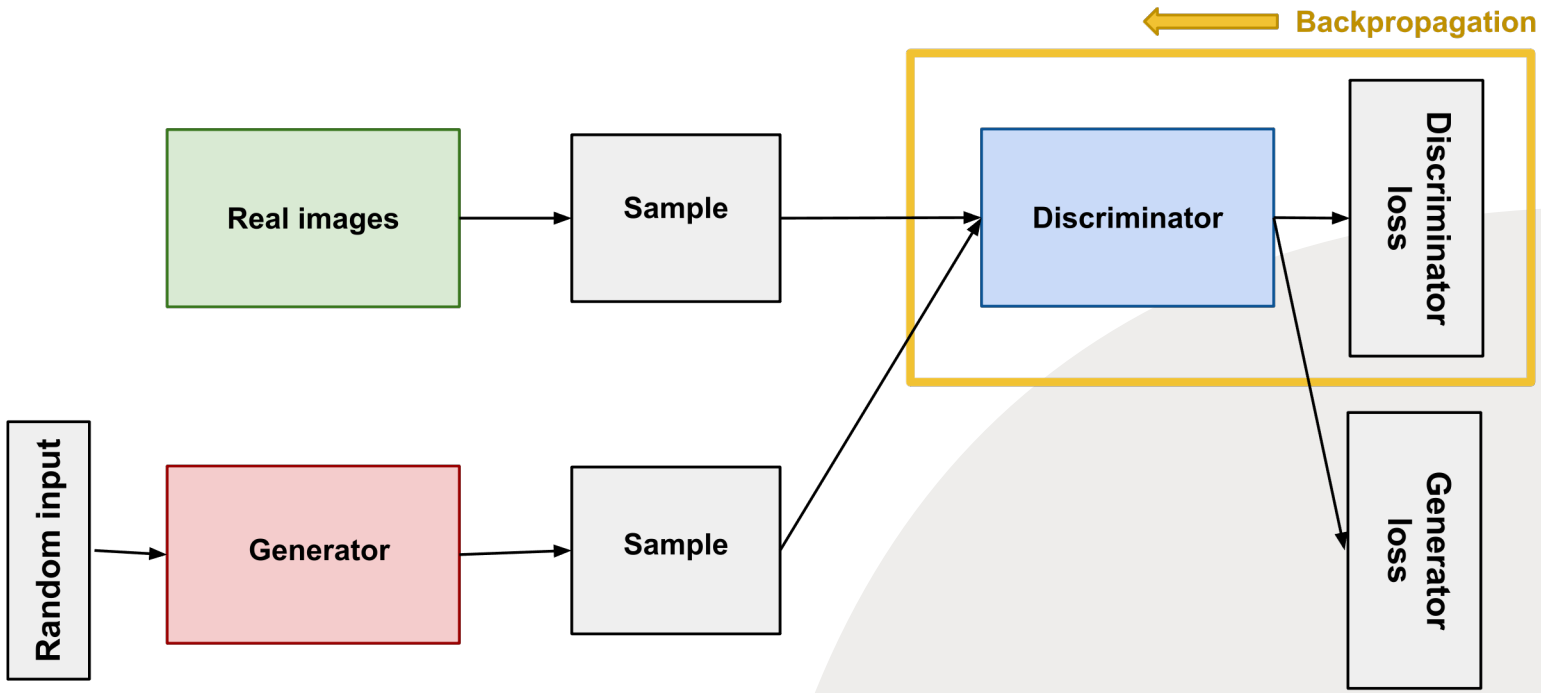
Both the generator and the discriminator are neural networks. The generator output is connected directly to the discriminator input. Through backpropagat ion, the discriminator's classification provides a signal that the generator uses to update its weights.



Overview of gans structure

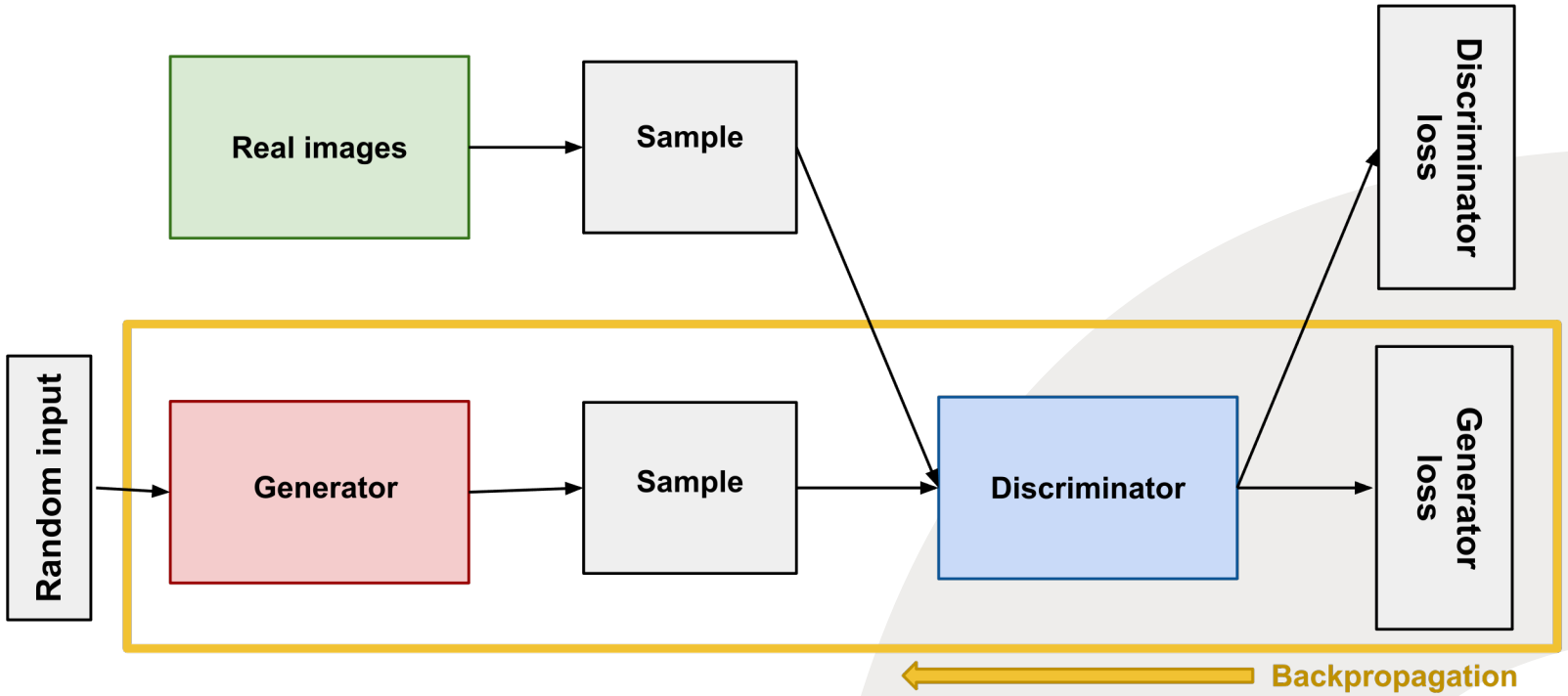
Training the Discriminator
-

The discriminator's training data comes from two sources:
Real data instances, such as real pictures of people. The discriminator uses these instances as positive examples during training.
Fake data instances created by the generator. The discriminator uses these instances as negative examples during training.



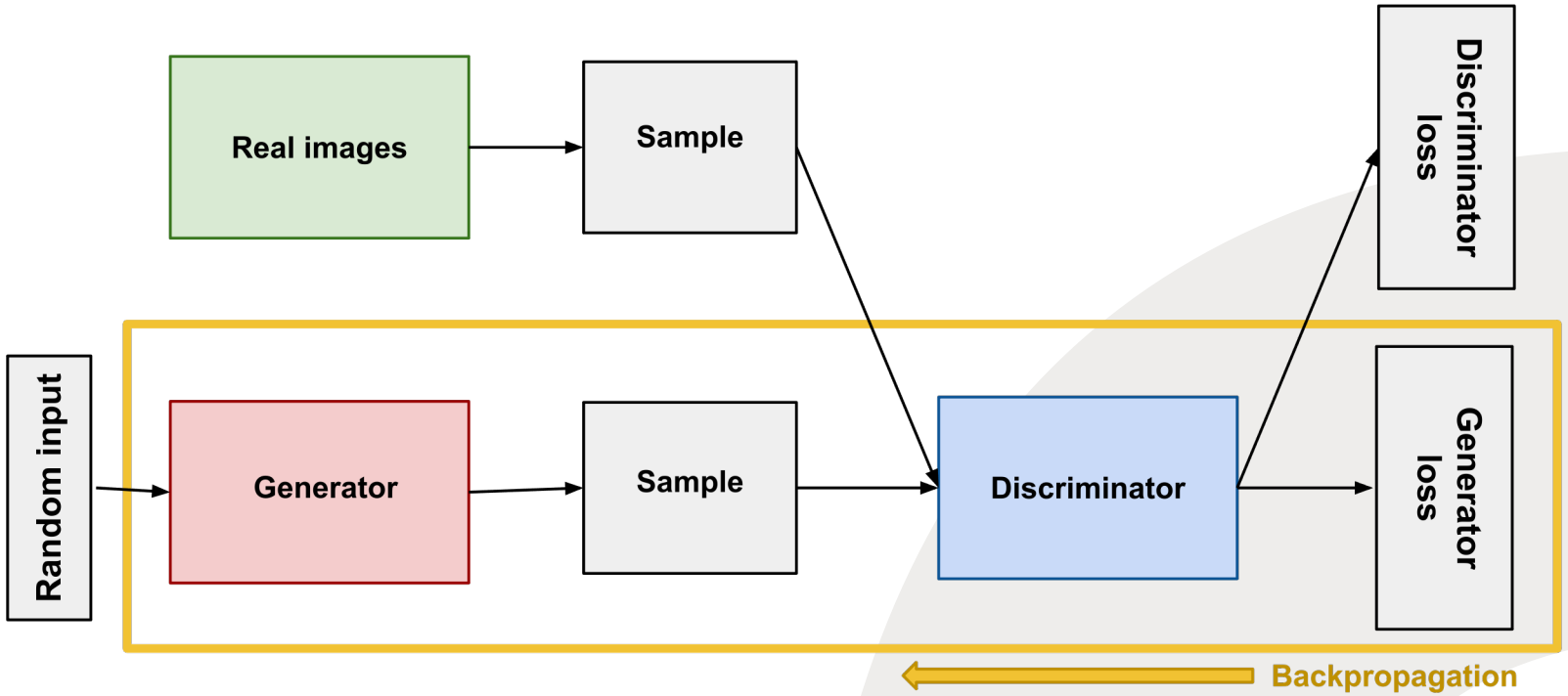
Overview of gans structure

The generator part of a GAN learns to create fake data by incorporating feedback from the discriminator. It learns to make the discriminator classify its output as real.



Using the Discriminator to Train the Generator

- Adjusting the weights in the generator NN by learning from backpropagation.
- We train the generator with the following procedure:
- Sample random noise.
- Produce generator output from sampled random noise.
- Get discriminator "Real" or "Fake" classification for generator output.
- Calculate loss from discriminator classification.
- Backpropagate through both the discriminator and generator to obtain gradients.
- Use gradients to change only the generator weights.
- This is one iteration of generator training.



Training

Let's visualize the working here.

Common Problems

- Vanishing Gradients
- Mode Collapse
- Failure to Converge

GANS Variations

Progressive GANs = In a progressive GAN, the generator's first layers produce very low resolution images, and subsequent layers add details. This technique allows the GAN to train more quickly than comparable non-progressive GANs, and produces higher resolution images.

Conditional GANs = Conditional GANs train on a labeled data set and let you specify the label for each generated instance. For example, an unconditional MNIST GAN would produce random digits, while a conditional MNIST GAN would let you specify which digit the GAN should generate.

Image-to-Image Translation = Image-to-Image translation GANs take an image as input and map it to a generated output image with different properties. For example, we can take a mask image with blob of color in the shape of a car, and the GAN can fill in the shape with photorealistic car details.

CycleGAN = CycleGANs learn to transform images from one set into images that could plausibly belong to another set. For example, a CycleGAN produced the righthand image below when given the left-hand image as input. It took an image of a horse and turned it into an image of a zebra.

GANS Variations

Text-to-Image Synthesis = Text-to-image GANs take text as input and produce images that are plausible and described by the text. For example, the flower image below was produced by feeding a text description to a GAN.

Super-resolution = Super-resolution GANs increase the resolution of images, adding detail where necessary to fill in blurry areas. For example, the blurry middle image below is a down sampled version of the original image on the left. Given the blurry image, a GAN produced the sharper image on the right:

Face Inpainting = GANs have been used for the *semantic image inpainting* task. In the inpainting task, chunks of an image are blacked out, and the system tries to fill in the missing chunks.

Text-to-Speech = Not all GANs produce images. For example, researchers have also used GANs to produce synthesized speech from text input.

Applications Of GANS

- Generate Examples for Image Datasets (Generate Cartoon Characters)
- Image-to-Image Translation
- Text-to-Image Translation
- Face Aging
- Photo Blending
- Clothing Translation
- DeepFakes

Deepfake 1

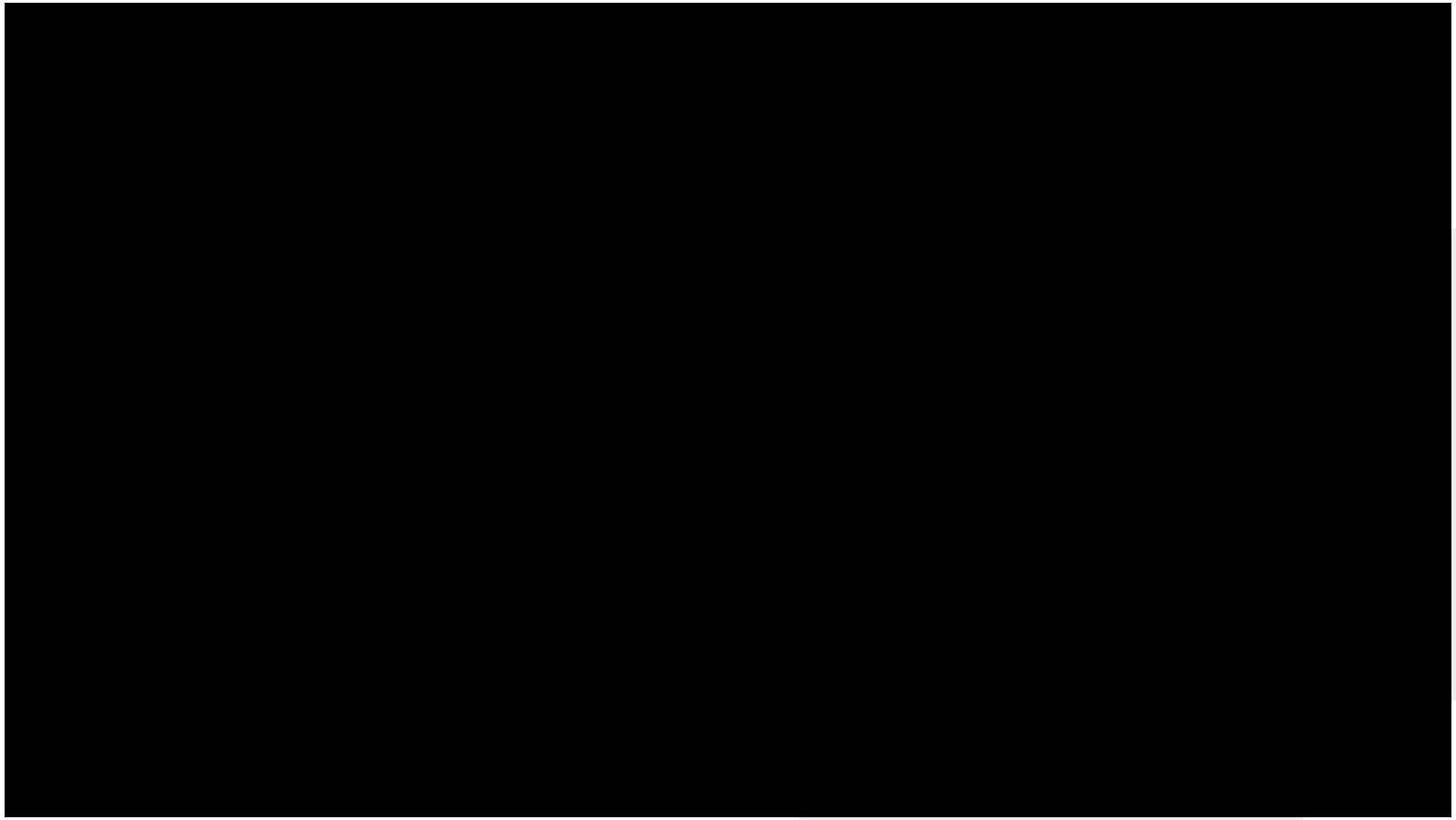
Better Call Trump:
Money Laundering 101
[DeepFake] =

The scene is from a series called Breaking bad where the lawyer Is explaining money laundering to his client.
Here we can see a subtle use of GAN to shift the faces of the Lawyer with President Trump and The criminal with Jared Kushner.

Deep Fake 2

Bill Hader Deepfake=

The scene is from a talk show where bill hader does some impressions of the great Al Pacino and Arnold Schwarzenegger and you can see the faces will change when the impressions start and then the face goes back to normal showing the sync that gans are capable of swapping data on runtime as well.



Questions

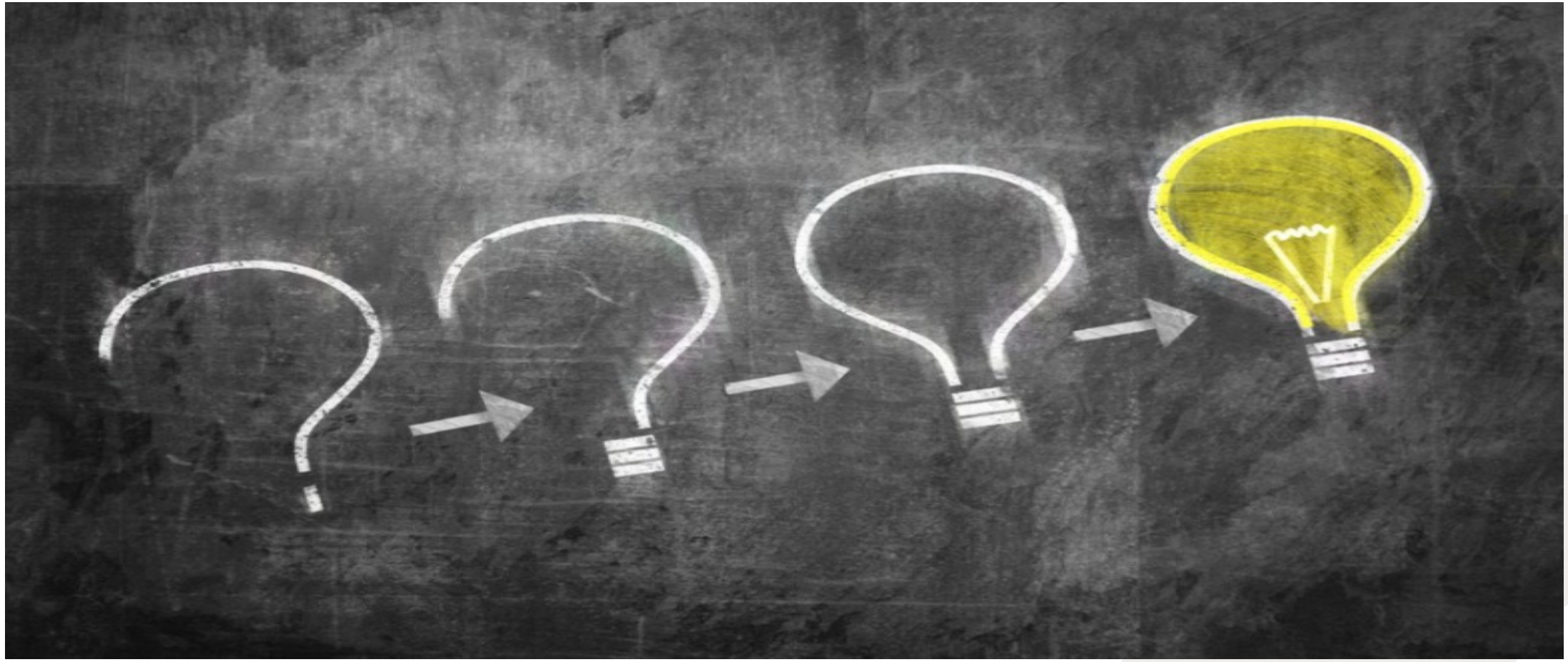
- Generate Examples for Image Datasets (Generate Cartoon Characters)
- Image-to-Image Translation
- Text-to-Image Translation
- Face Aging
- Photo Blending
- Clothing Translation
- DeepFakes

Assignment

- You will be predicting whether or not a particular video is a deepfake. A deepfake could be either a face or voice swap (or both). In the training data, this is denoted by the string "REAL" or "FAKE" in the label column. In your submission, you will predict the probability that the video is a fake.

- https://www.kaggle.com/c/deepfake-detection-challenge/data

Questions?



Vielen Dank für deine Aufmerksamkeit!

Kontakt:
Ashish Chouhan
SRH Hochschule Heidelberg
Ludwig-Guttmann-Straße 6
69123 Heidelberg
Phone: +49 6221 6799-224
Mail ID: ashish.chouhan@srh.de

Ajinkya Patil
Mail ID: ajinkya.patil.extern@srh.de
ajinkya.patil@sap.com