LECTURER: Nghia Duong-Trung

DEEP LEARNING

Introduction to Neural Networks and Deep Learning	1
Network Architectures	2
Neural Network Training	3
Alternative Training Methods	4
Further Network Architectures	5.1

UNIT 5

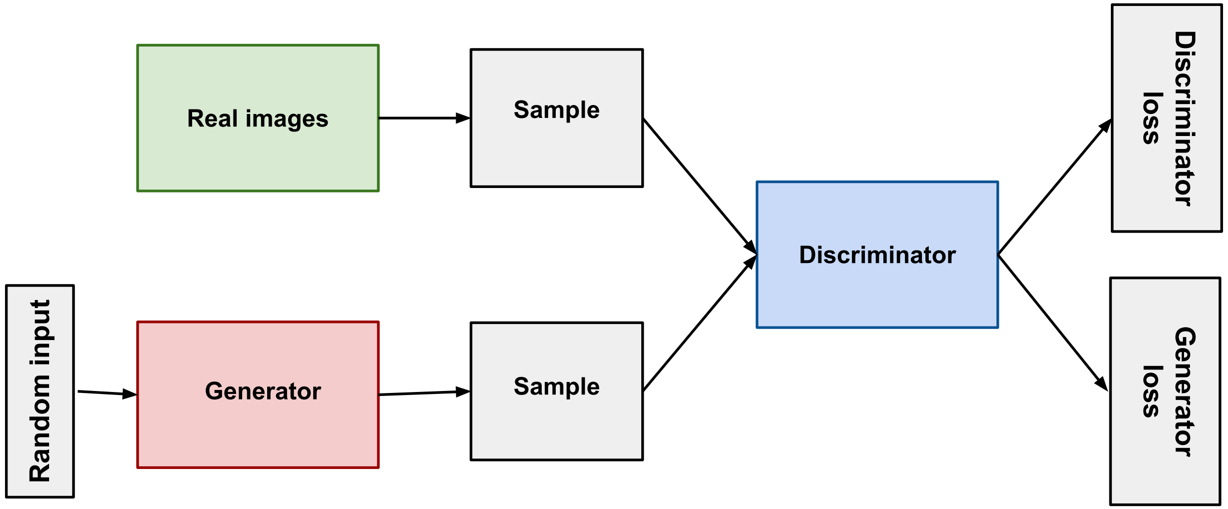
Further Network Architectures

STUDY GOALS

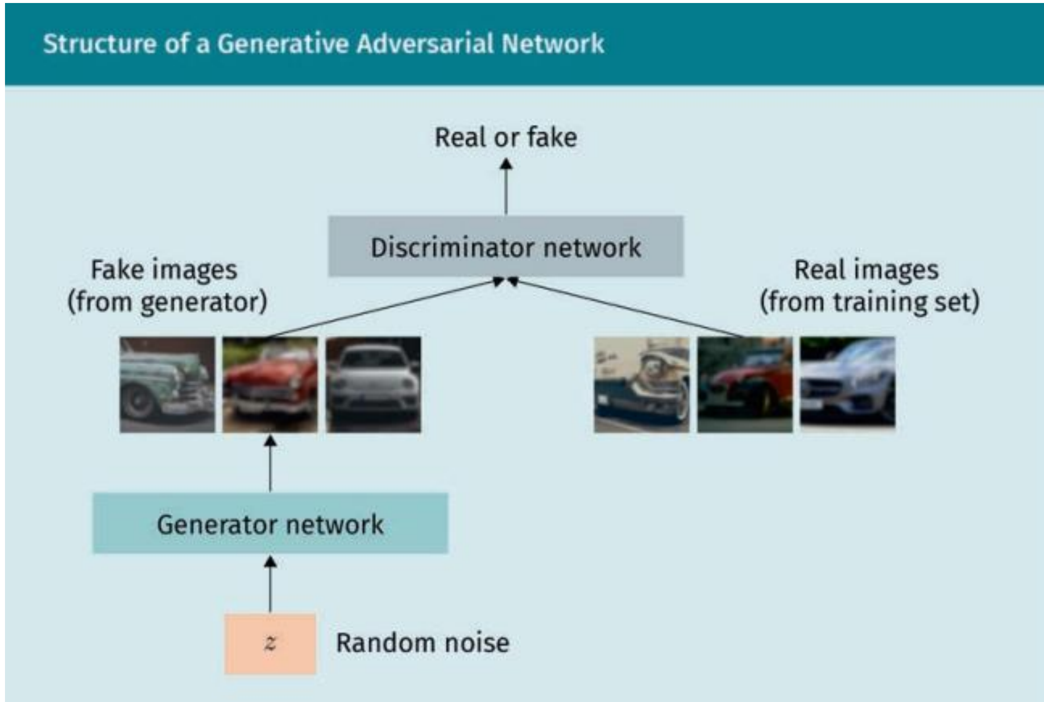
- After completing this unit you will be able to ...
  ... describe what generative adversarial networks (GANs) are and how to train them.
  ... build and train autoencoders and variational autoencoders (VAEs).
  ... explain what restricted Boltzmann machines (RBMs) are and how they are used for recommender systems.
  ... understand potential evolutions in deep learning in the form of capsule and spiking neural networks.

GENERATIVE ADVERSARIAL NETWORKS (GANS)

- The goal of a generative model is to generate new samples from the same distribution as a given set of training data.
- **Generative adversarial networks** (GANs), introduced by Goodfellow et al. (2014), are a specific class of generative deep learning model consisting of two opposing components:
  - Generator networks try to generate new samples from the same distribution as the training data.
  - Discriminator networks try to classify input data as either “real” (i.e., coming from the training data) or “fake” (i.e., outputted by the generator network).
- Conditional GANs: This is a GAN variant that allows the generation of samples from a desired class by using labels as input data.



https://developers.google.com/machine-learning/gan/gan_structure

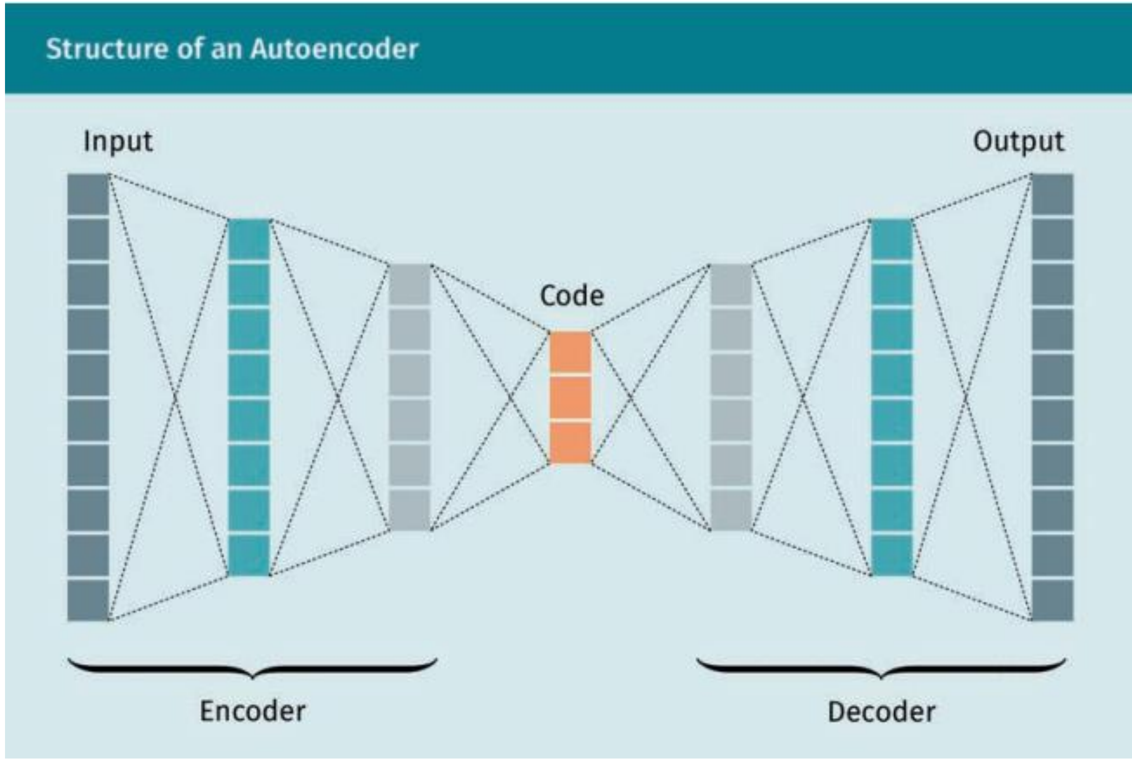


GENERATIVE ADVERSARIAL NETWORKS (GANS)

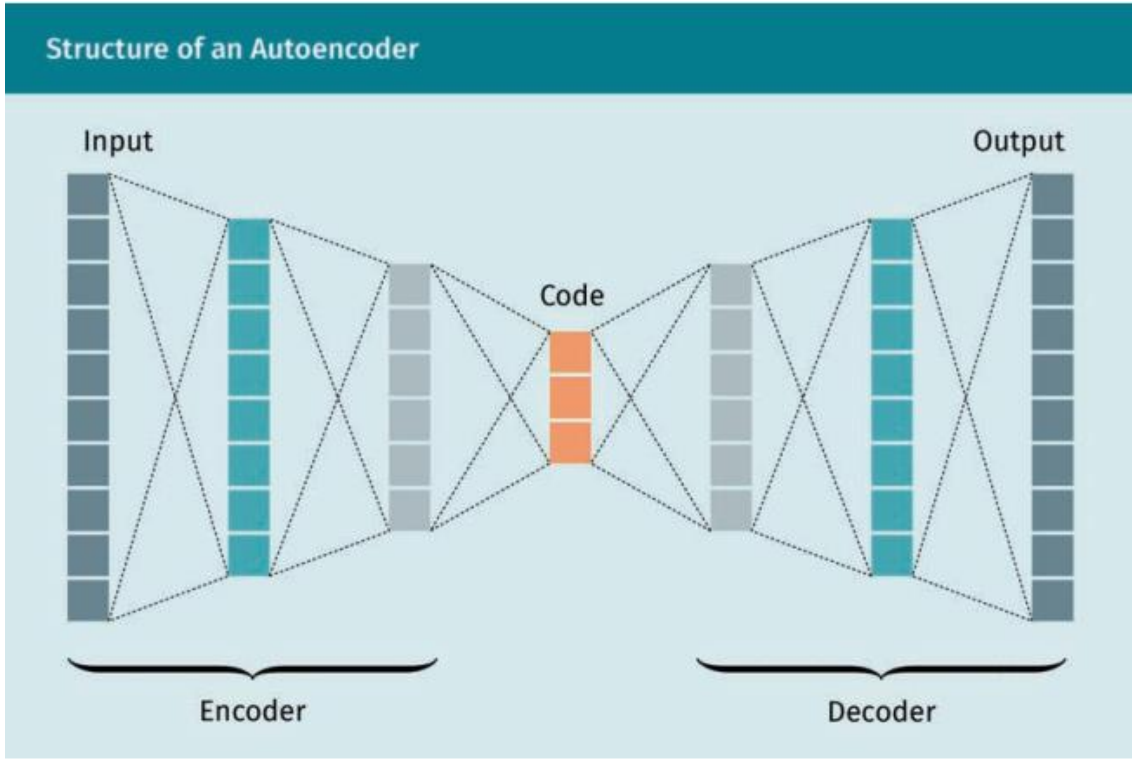
- https://salu133445.github.io/musegan/
- https://devpost.com/software/gan-music-generator
- https://www.projectpro.io/article/generative-adversarial-networks-gan-based-projects-to-work-on/530
- https://thispersondoesnotexist.com/

AUTOENCODERS

- **Autoencoders** are neural networks whose superficial task is to learn the identity function $f(x) = x$. In other words, autoencoders learn to reconstruct their input.
- The real purpose of the autoencoder, however, is to learn an efficient representation of the input data that, ideally, preserves only the information required to obtain a sufficiently faithful reconstruction.
- Autoencoders consist of two main sub-networks: an encoder and a decoder.
- Regularized autoencoders: These are autoencoders that use one or several techniques designed to decrease the complexity of the functions learned by the encoder and decoder.
- The autoencoder is trained to minimize reconstruction error simultaneously, and this has the effect of driving the code features that are unimportant for reconstruction toward zero.

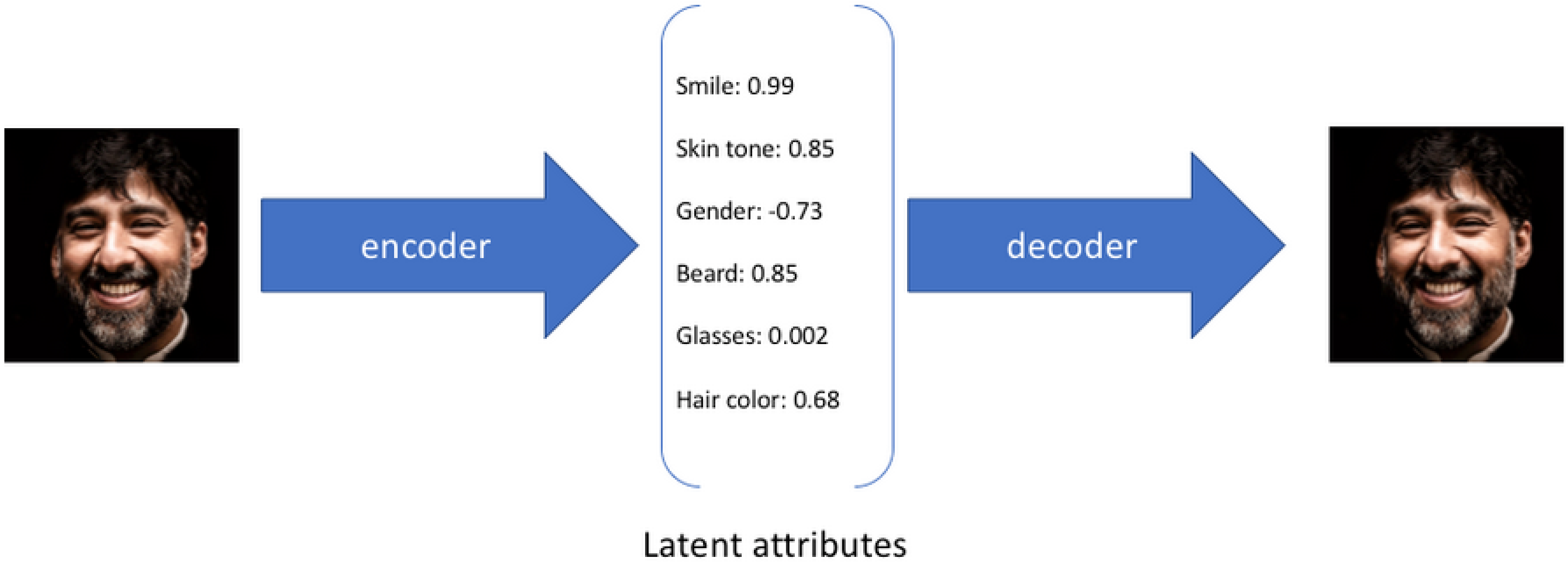


- How Do Autoencoders Function
- Encoder: The encoder layer compresses the input image into a smaller representation. The original image's distortion is clearly seen in the compressed version.
- Code: This portion of the network only represents the decoder's compressed input.
- Decoder: Using a lossy reconstruction and the latent space representation, this decoder layer restores the encoded image to its original dimension.



AUTOENCODERS

- An autoencoder with high capacity can end up learning an identity function (also called null function) where input=output.

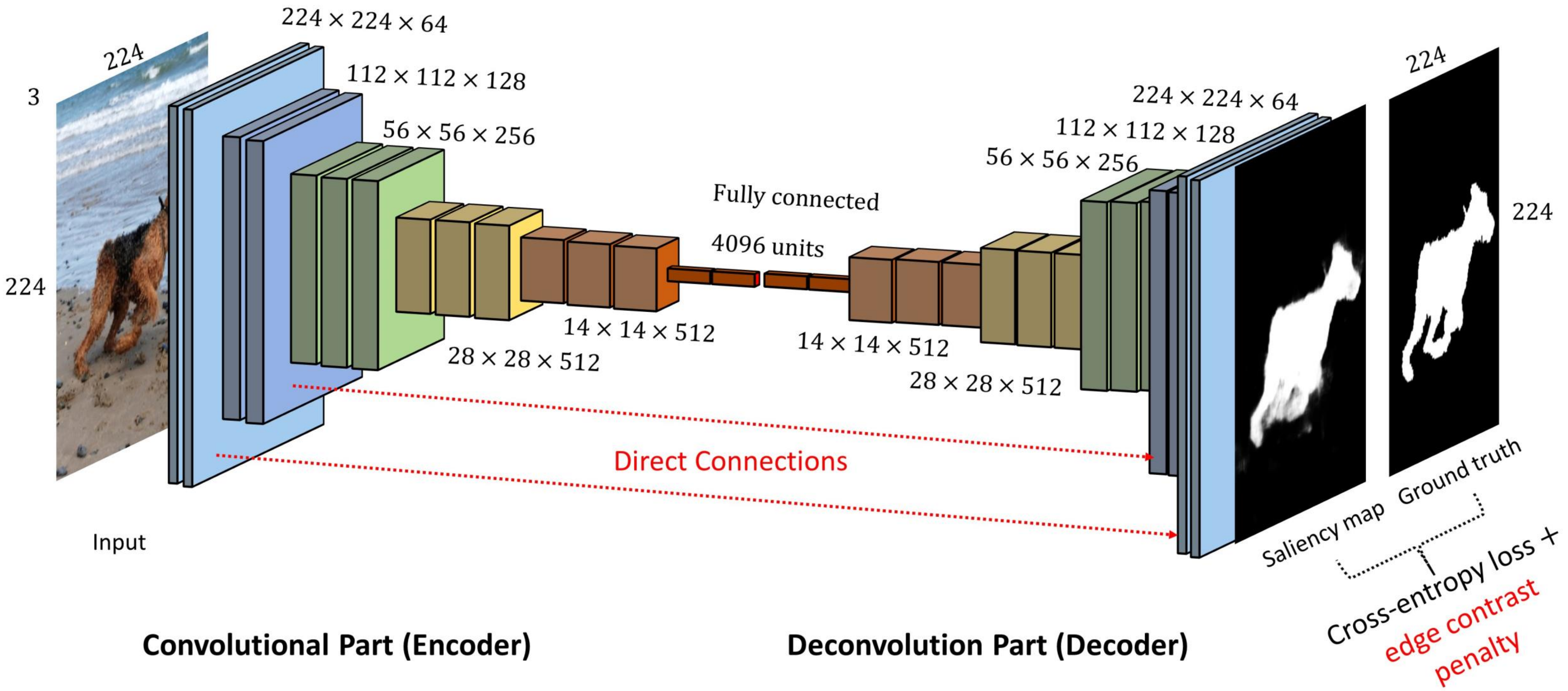


https://www.jeremyjordan.me/variational-autoencoders/

TRANSFER TASK

- Notebook AE > 01. PyTorch-Autoencoder.ipynb

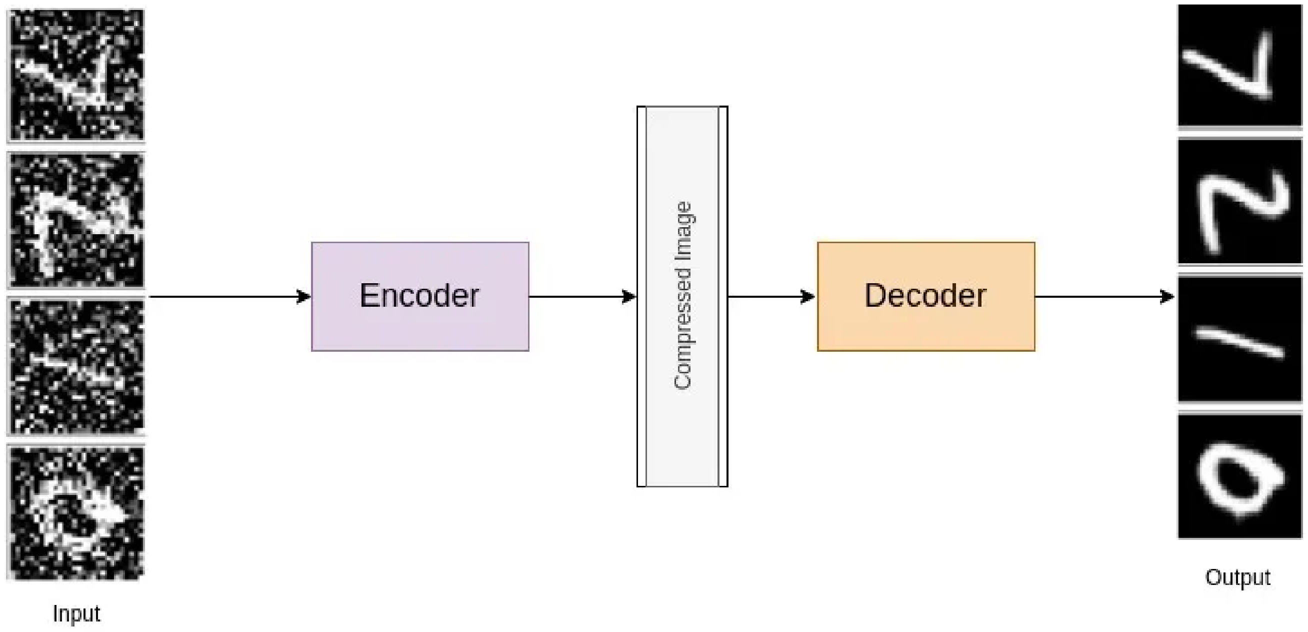
CONVOLUTIONAL AUTOENCODER



https://www.gabormelli.com/RKB/Convolutional_Autoencoder

DENOISING AUTOENCODER

- A denoising auto-encoder does two things:
  - Encode the input (preserve the information about the data)
  - Undo the effect of a corruption process stochastically applied to the input of the auto-encoder.

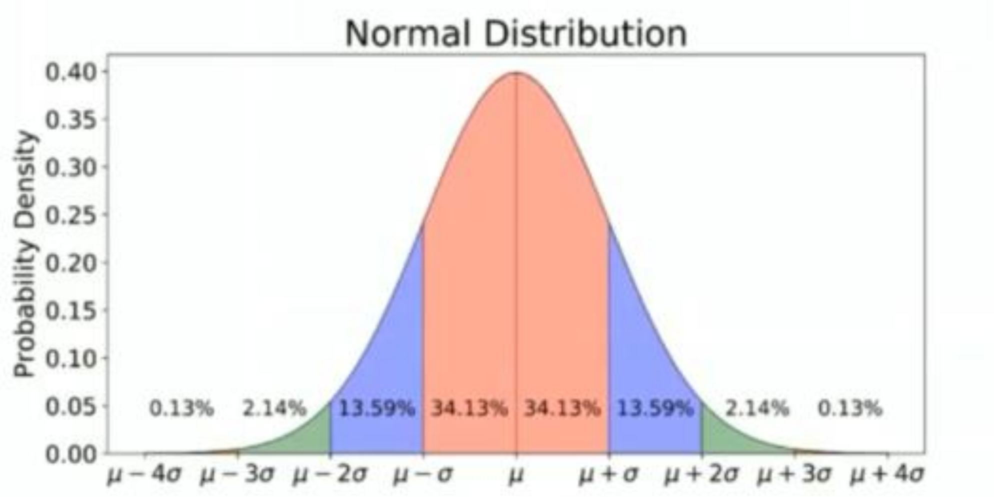


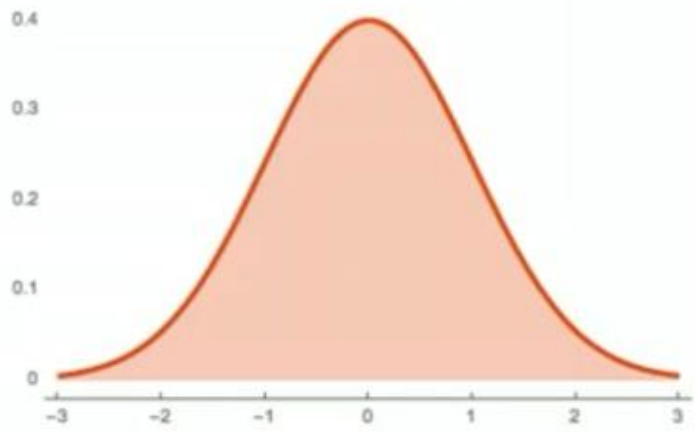
https://medium.com/analytics-vidhya/reconstruct-corrupted-data-using-denoising-autoencoder-python-code-aeaff4b0958e

TRANSFER TASK

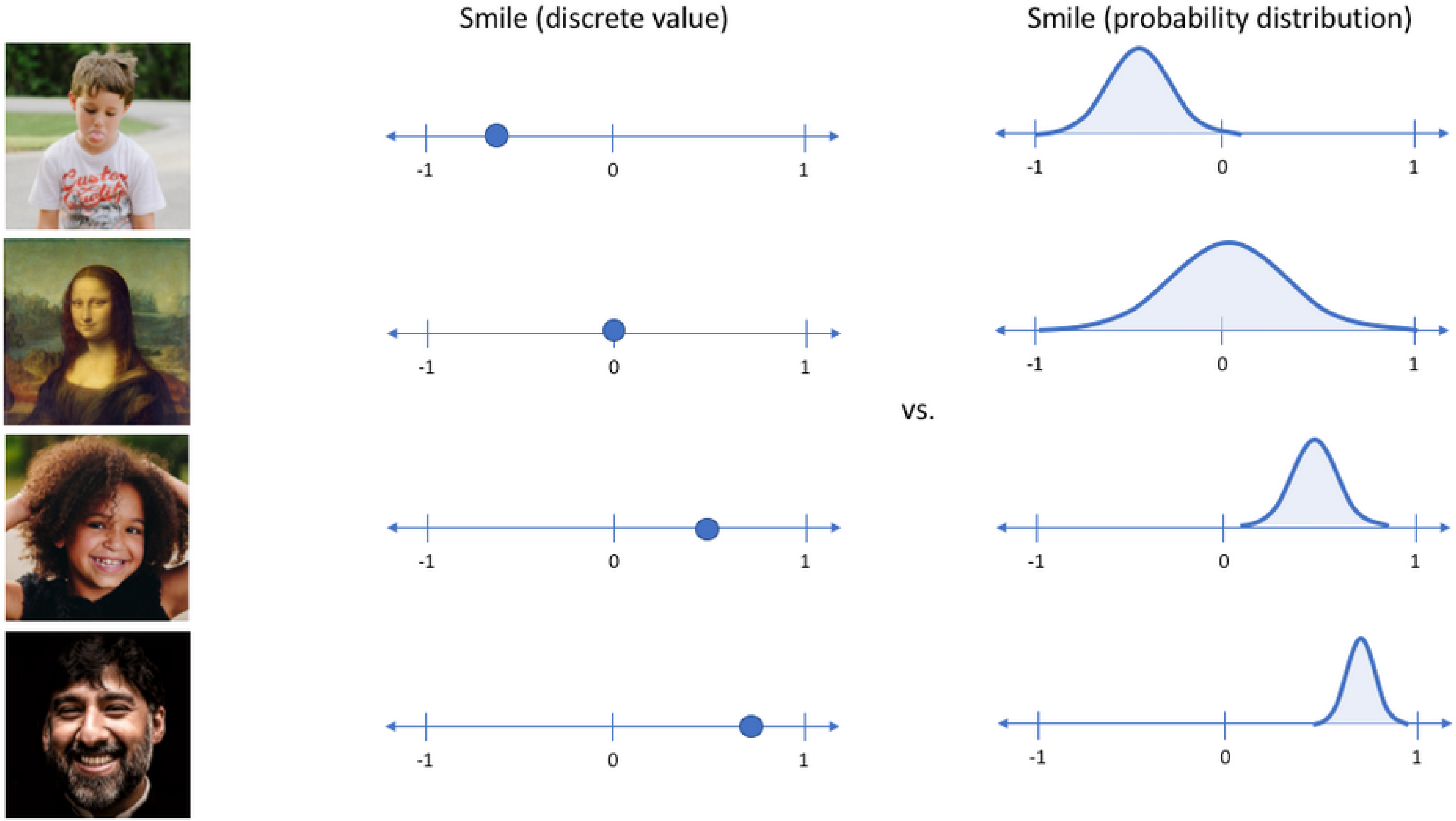
- Notebook AE > 02. Denoising_Autoencoder.ipynb

- What's the problem in Autoencoders?
  - The compressed representation does not know to which distribution it belongs to
  - At testing time, if we give random numbers to the decoder, we might get a random image.
  - Therefore, it is necessary to know which distribution to take the random numbers from and supply them to the decoder.
- Instead of forcing the encoder to produce a single encoding (as a normal autoencoder) with random numbers not know to which distribution they belong to, we want to force the encoder to produce a specific probability distribution (in practice, a Gaussian / normal distribution) over encodings. The decoder will then sample an encoding from that probability distribution, and try to reconstruct the original input -> variational autoencoder



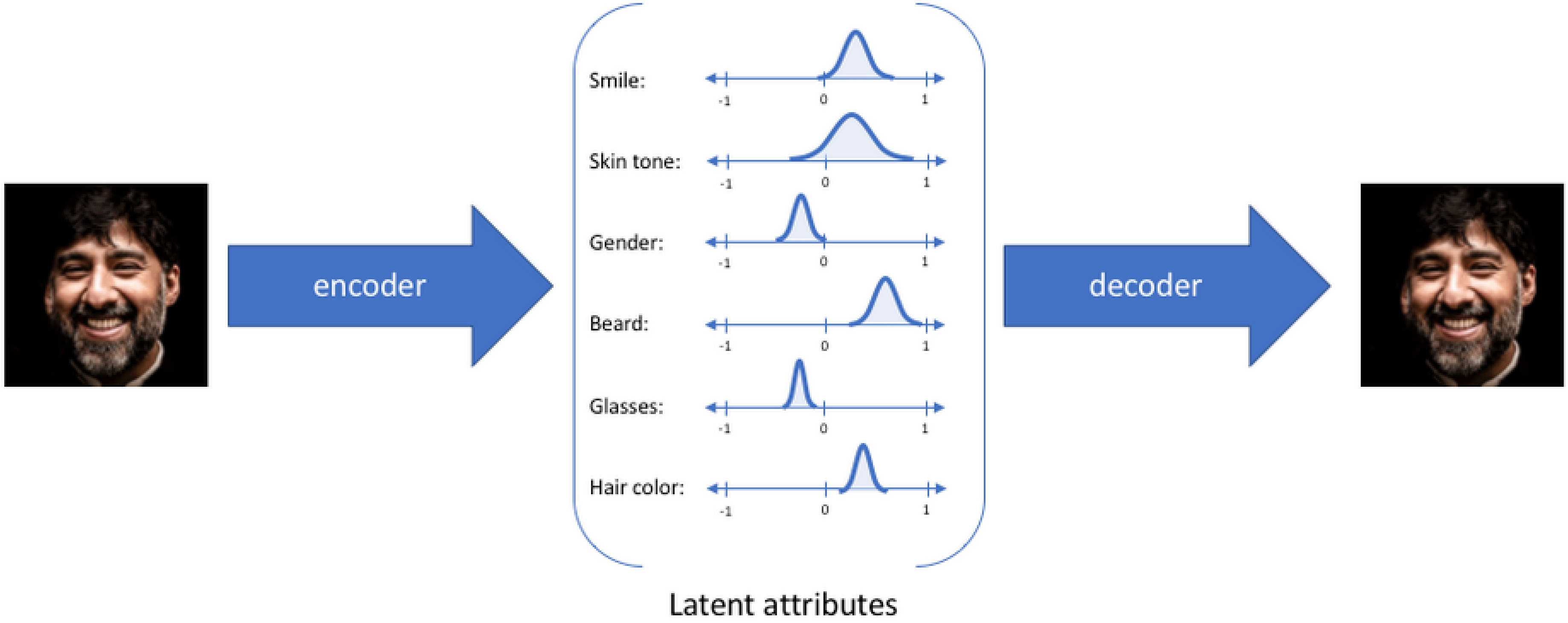


- With this approach, we'll now represent each latent attribute for a given input as a probability distribution. When decoding from the latent state, we'll randomly sample from each latent state distribution to generate a vector as input for our decoder model.



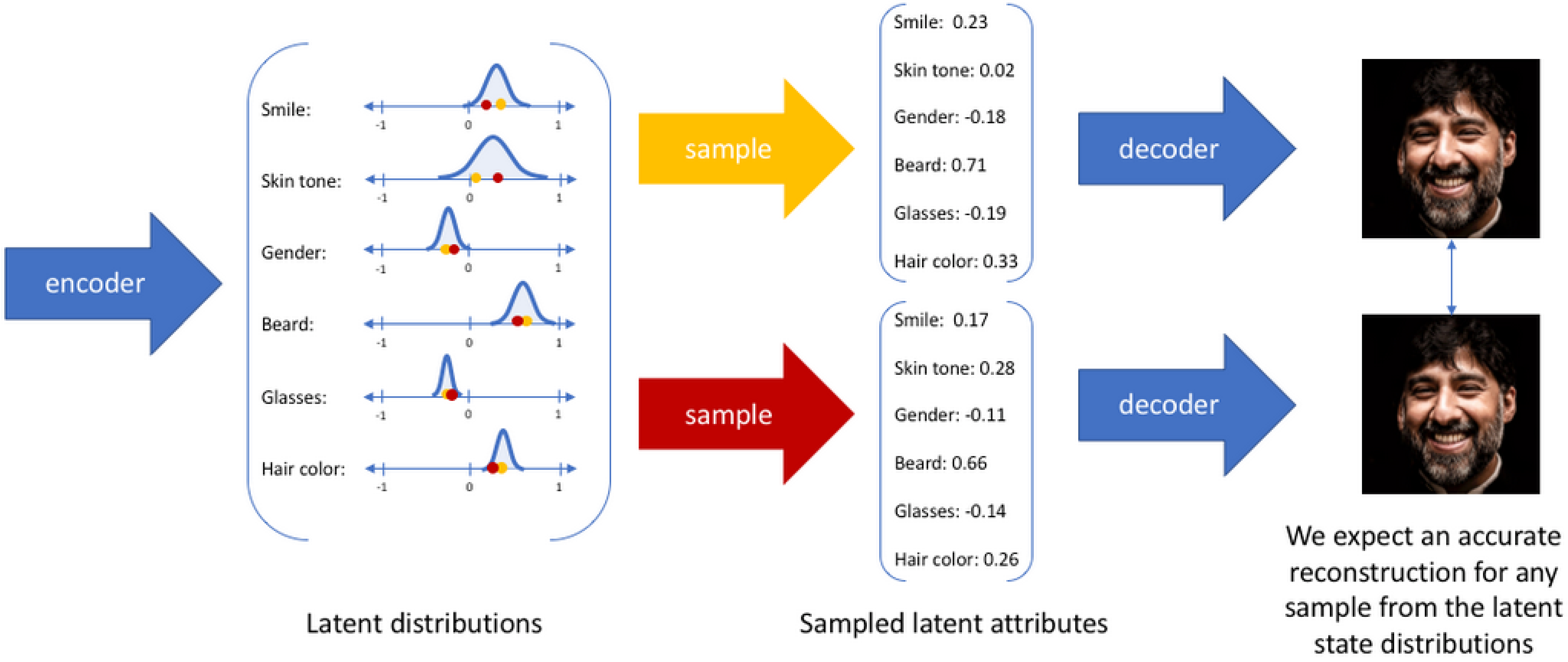
https://www.jeremyjordan.me/variational-autoencoders/

- By constructing our encoder model to output a range of possible values (a statistical distribution) from which we'll randomly sample to feed into our decoder model, we're essentially enforcing a continuous, smooth latent space representation.
- For any sampling of the latent distributions, we're expecting our decoder model to be able to accurately reconstruct the input. Thus, values which are nearby to one another in latent space should correspond with very similar reconstructions.



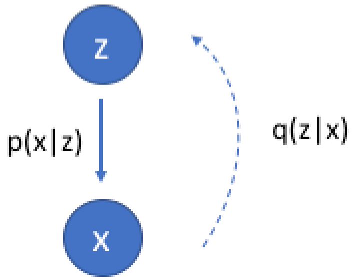
https://www.jeremyjordan.me/variational-autoencoders/

VARIATIONAL AUTOENCODERS

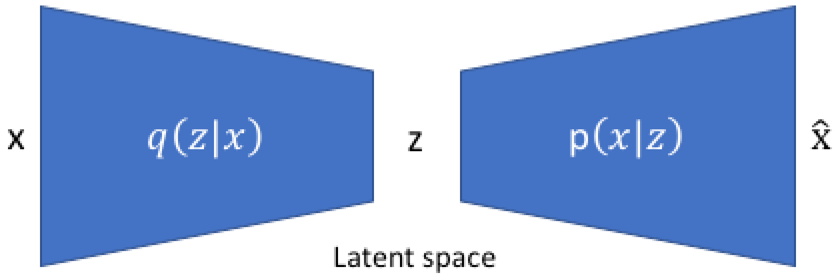


https://www.jeremyjordan.me/variational-autoencoders/

VARIATIONAL AUTOENCODERS



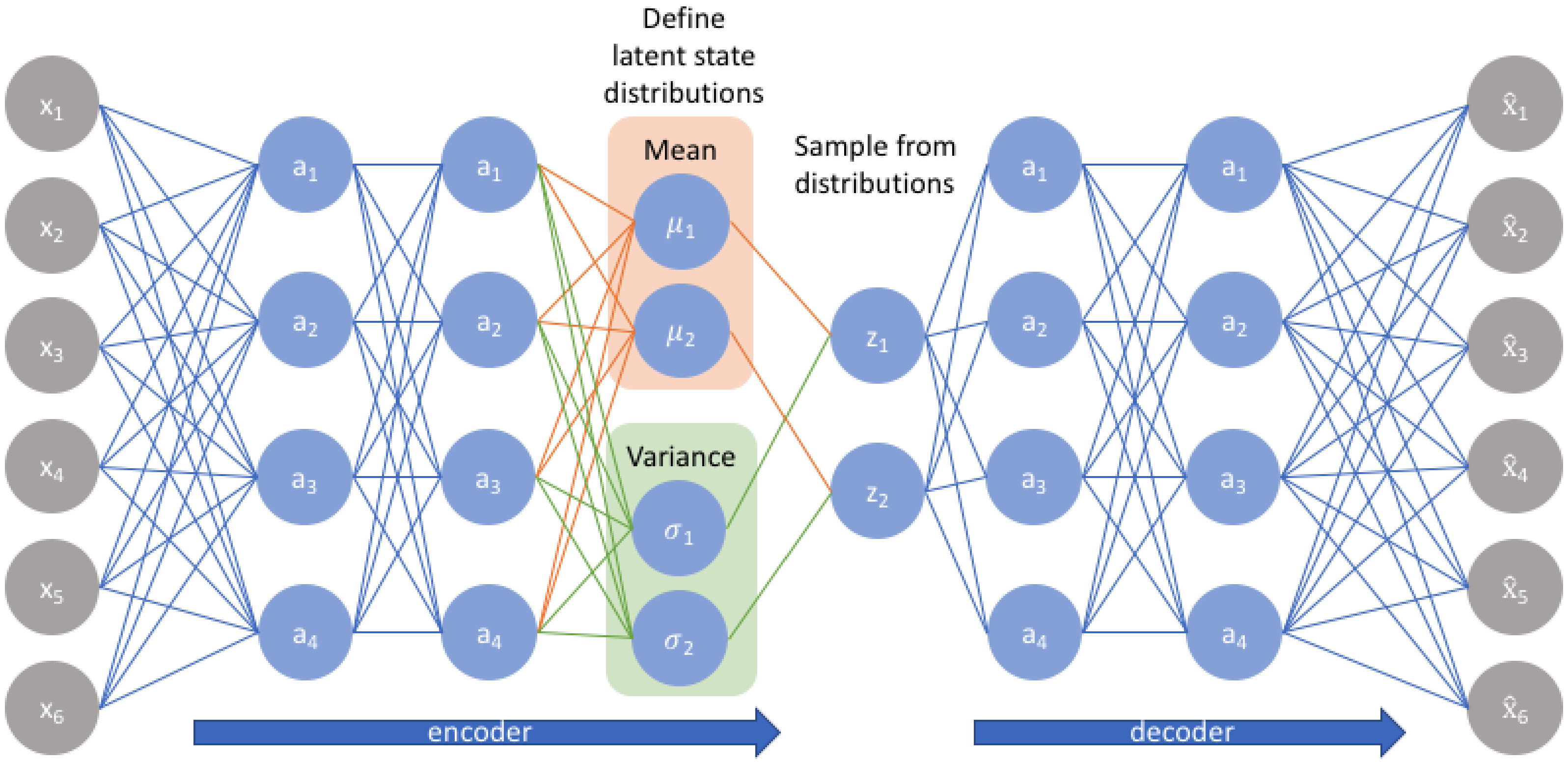
We'd like to use our observations to understand the hidden variable.



Latent space representation.

Neural network mapping x to z.

Neural network mapping z to x.



https://www.jeremyjordan.me/variational-autoencoders/

TRANSFER TASK

- Notebook AE > 03_Variational_AutoEncoder.ipynb

REVIEW STUDY GOALS

- After completing this unit, you will be able to ...
  … describe how the concept of attention is modeled in deep learning.
  … explain what bidirectional RNNs are and how they are used in language translation.
  … understand how feedback alignment addresses issues in the backpropagation algorithm.
  … use synthetic gradients and decoupled neural interfaces to parallelize computations and speed up training.
  … recognize the importance of transfer learning in real-world applications of deep learning.

SESSION 5.1

Further Network Architectures

1. How many layers are there in the Autoencoder?

A. 2

B. 3

C. 4

D. 5

2. Select the correct option.

Statement 1: Supervised learning methods include autoencoders.

Statement 2: The output and input of the autoencoder are identical.

A. Both the statements are TRUE.

B. Statement A is TRUE, but statement B is FALSE.

C. Statement A is FALSE, but statement B is TRUE.

D. Both the statements are FALSE.

3. Autoencoders are trained using______.

A. Feed forward.

B. Reconstruction.

C. Back propagation.

D. They do not require training.