Intro to Generative AI

Thanos Tagaris

What is Generative AI?

[Sudar Pichai (Google CEO) - Google I/O 2023]

Image Generation?

Teddy bears swimming at the Olympics 400m Butterfly event.

A blue jay standing on a large basket of rainbow macarons.

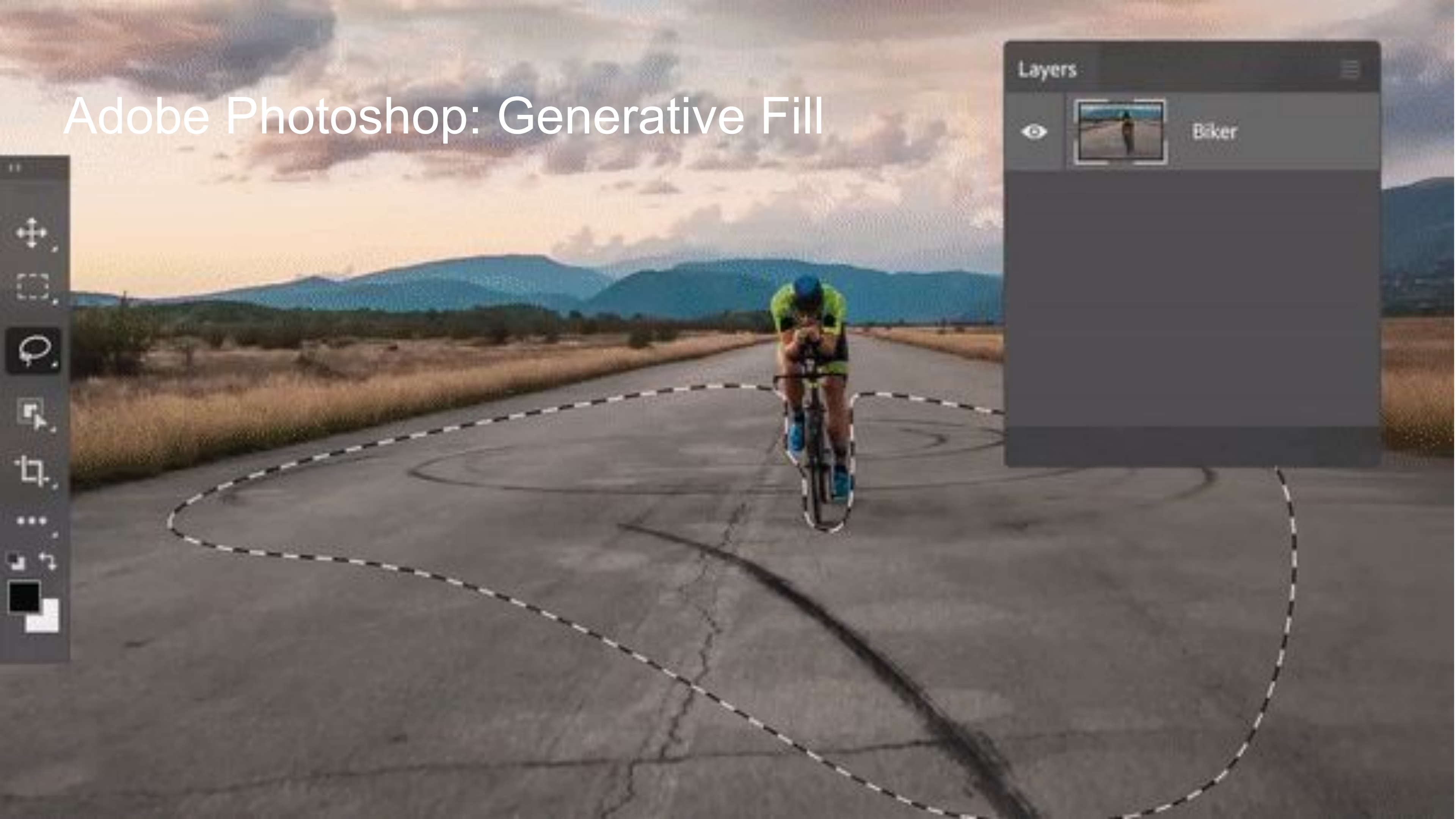
An art gallery displaying Monet paintings. The art gallery is flooded. Robots are going around the art gallery using paddle boards.

A transparent sculpture of a duck made out of glass. The sculpture is in front of a painting of a landscape.

Imagen


DALL-E

Adobe Photoshop: Generative Fill



Conversational Agents?


bard.google.com
Bard
Experiment
I’m Bard, your creative and helpful collaborator. I have limitations and won’t always get it right, but your feedback will help me improve.
Not sure

Scope

This session has 1 main goal: to go beyond the hype, the buzzwords and the success stories and understand 3 main things

- What do Generative ML models try to learn?
- What are Generative Neural Networks and how do they differ from NN for classification?
- What architectures and training schemes are used to produce Generative Neural Networks?

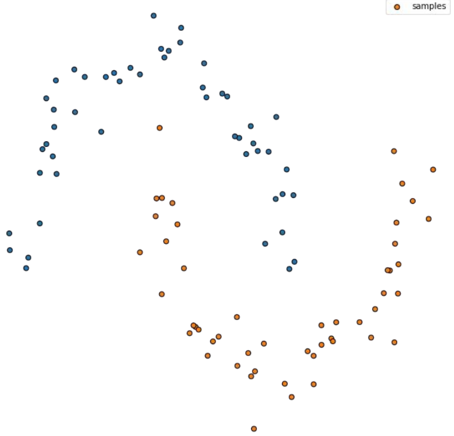
Contents

1. Theoretical background
    - Data generating distribution
    - Generative vs discriminative ML models
2. Neural Network Architectures & Training Schemes
    - Discriminative Neural Nets
    - Autoencoder
    - VAE
    - GAN
    - Diffusion models

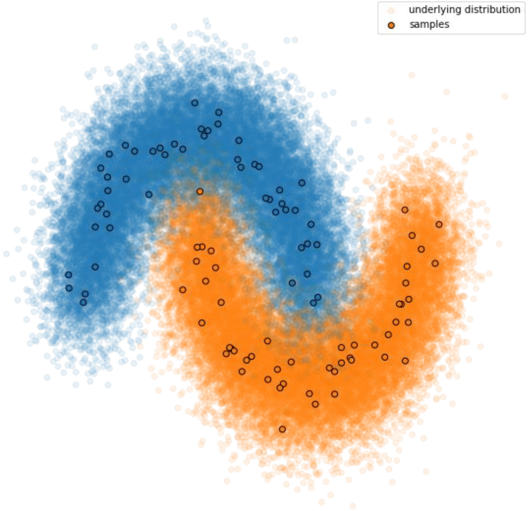
Theoretical Background

Data Generating Distribution

A very important concept in Machine Learning is that of the **data-generating distribution**.



Say we are given a dataset



We consider these instances to be samples drawn from a data generating distribution

Data Generating Distribution

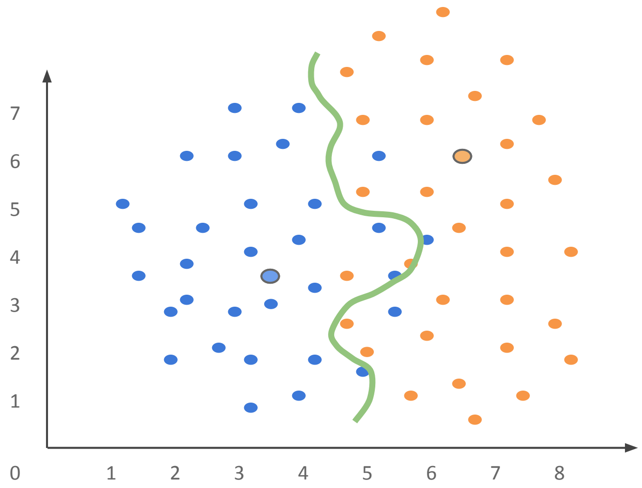
- This distribution is a theoretical notion. We have no concrete evidence on how the distribution actually looks like.
- E.g. “cat vs dog image classification”
- What is the underlying distribution of the cat class?
- It should consist of everything that makes a cat, a cat. Any conceivable cat image will be drawn from that distribution.
- Besides this, the samples are also accompanied by noise.
- For example the “cat images” class this could be a tree, a table or any object that is not related to the cat.

Discriminative vs Generative Models

Depending on how Machine Learning models approach the process of “learning” we can make two main distinctions:

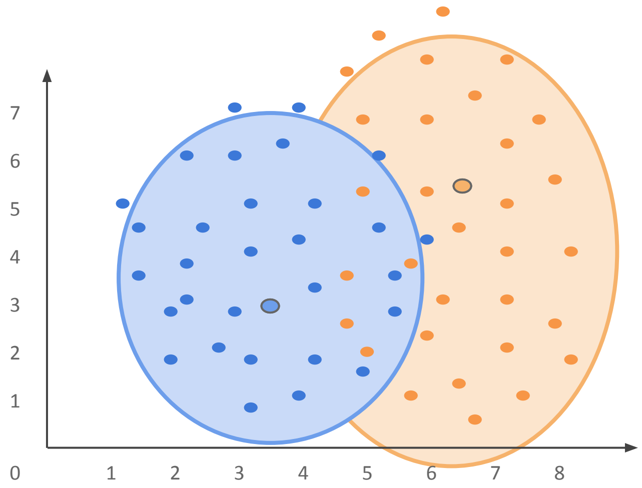
- **Generative** models
  - These attempt to predict the target y by learning the joint probability distribution p(x, y).
- **Discriminative** models
  - These attempt to predict the target y by learning conditional probability p(y | x) directly.

Discriminative models



- **Discriminative** models try to learn what features **distinguish** each class from the other.
- To predict a previously unseen example it looks at what the distinguishing features have to say about the class.

Generative models



- **Generative** models try to “learn” the **underlying distribution** behind each class.
- To predict a previously unseen example it looks at what distribution that example is more likely to have come from.

Question time

- Which of the two tasks do you think is easier?
- Which of the two do you think us humans learn?
- Challenge:

draw a 20€ bill

Discriminative vs Generative Neural Networks

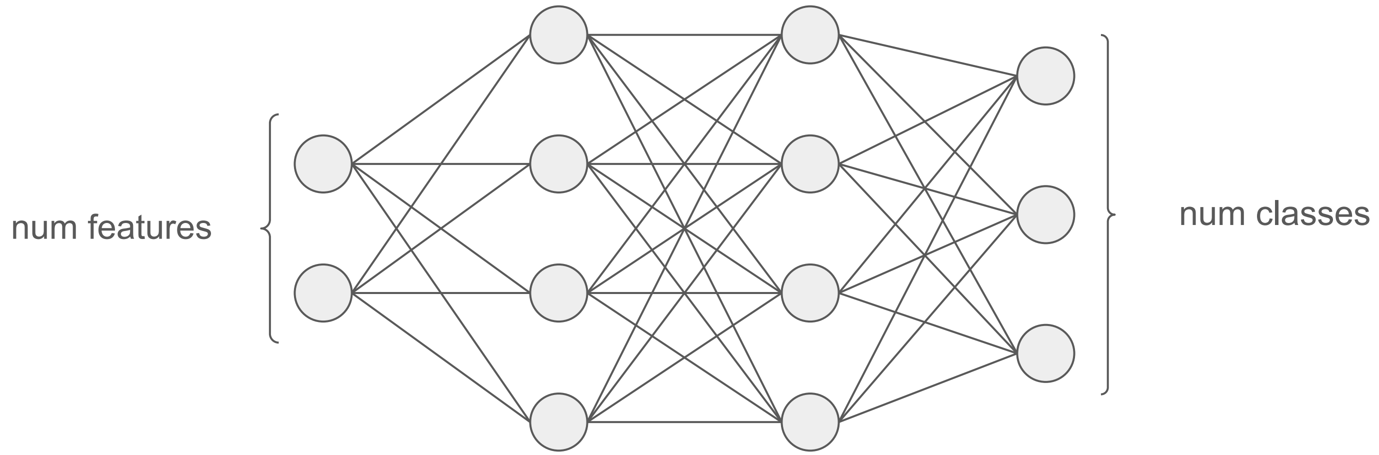
- **Discriminative** Neural Networks try to solve their task by learning what features **distinguish** each class from the other.
    - e.g. Neural Networks for classification
- **Generative** Neural Networks try to solve their task by learning the **underlying data generating distribution**
    - e.g. Autoencoders
- It all comes down to (a) the architecture of the Neural Network and (b) the training scheme to determine if it will be trained in a **generative** or **discriminative** way.

Neural Network Architectures and Training Schemes

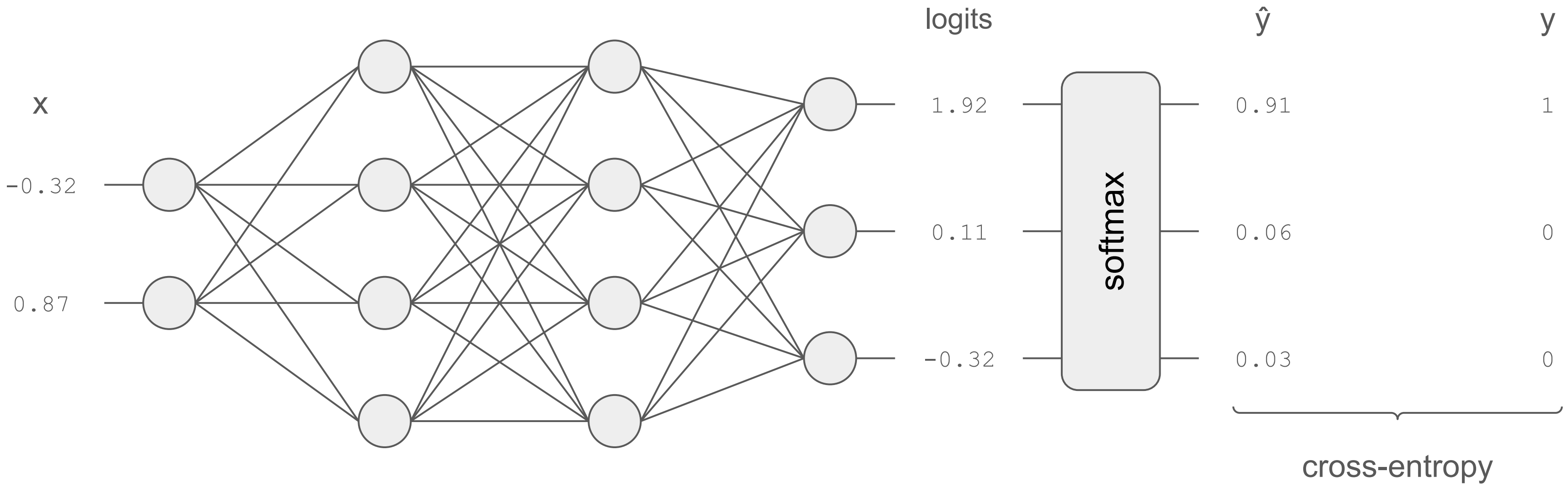
Discriminative Neural Networks

Neural Networks for classification

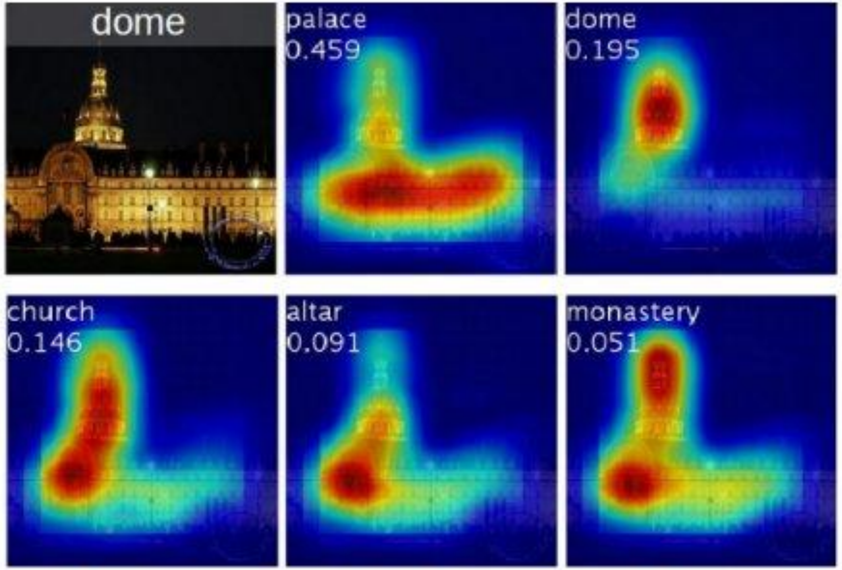
- Typical type of Neural Network we are familiar with.
- Are trained with Maximum Likelihood Estimation.
- This leads to a discriminative training



Neural Networks for classification



So what do these types of networks actually learn?



Class activation maps of top 5 predictions

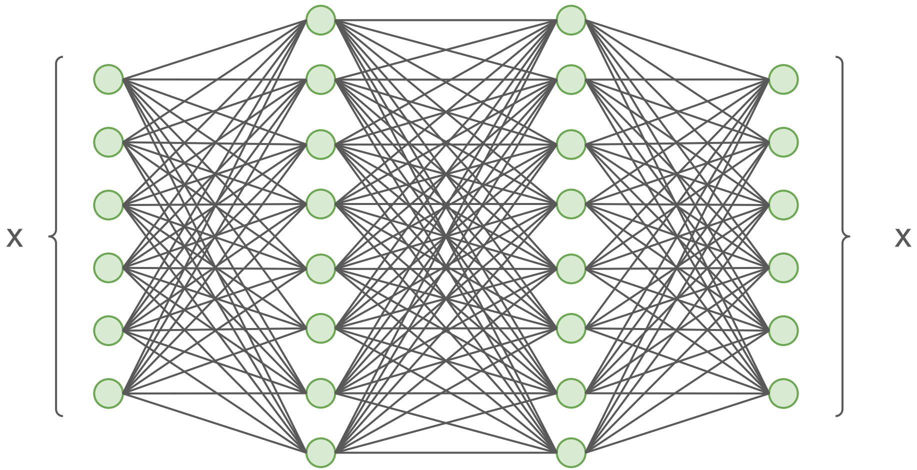
Class activation maps for one object class

discriminative features, i.e. how to distinguish from one class to another

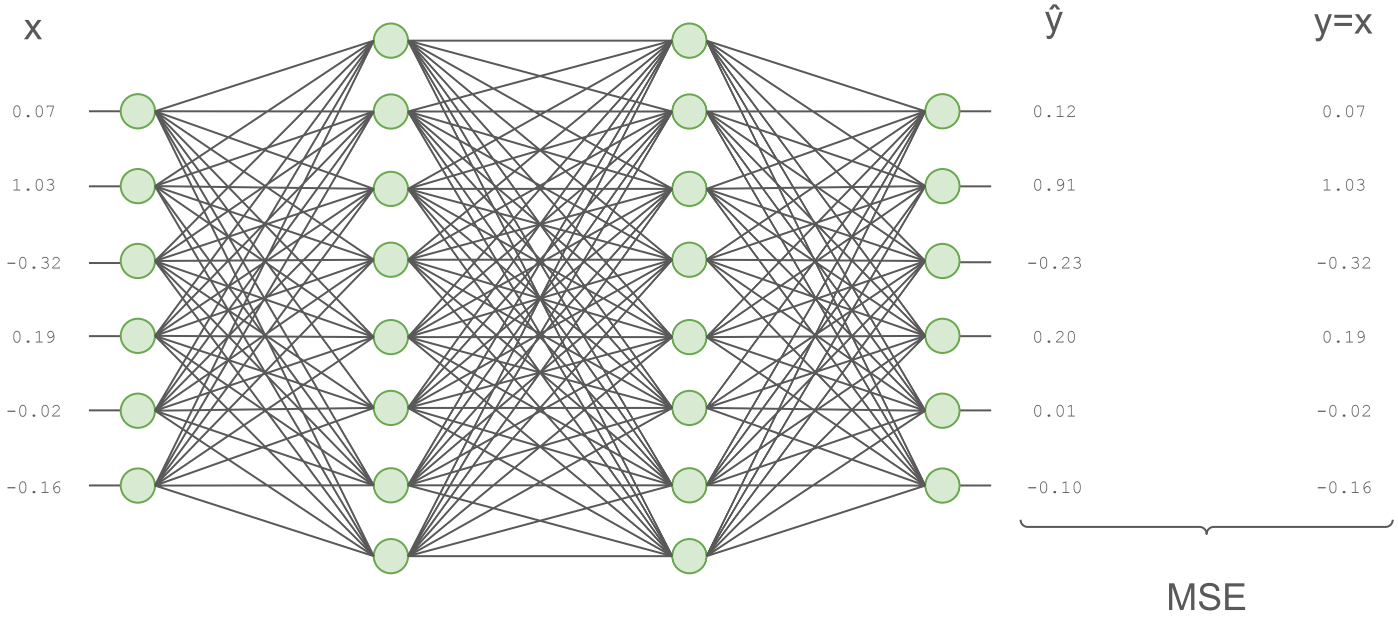
Generative Neural Networks: AutoEncoders

AutoEncoders (AE)

- A Neural Network architecture that has the same shape for its input and output
- Trained in an unsupervised manner
- This leads to a generative training

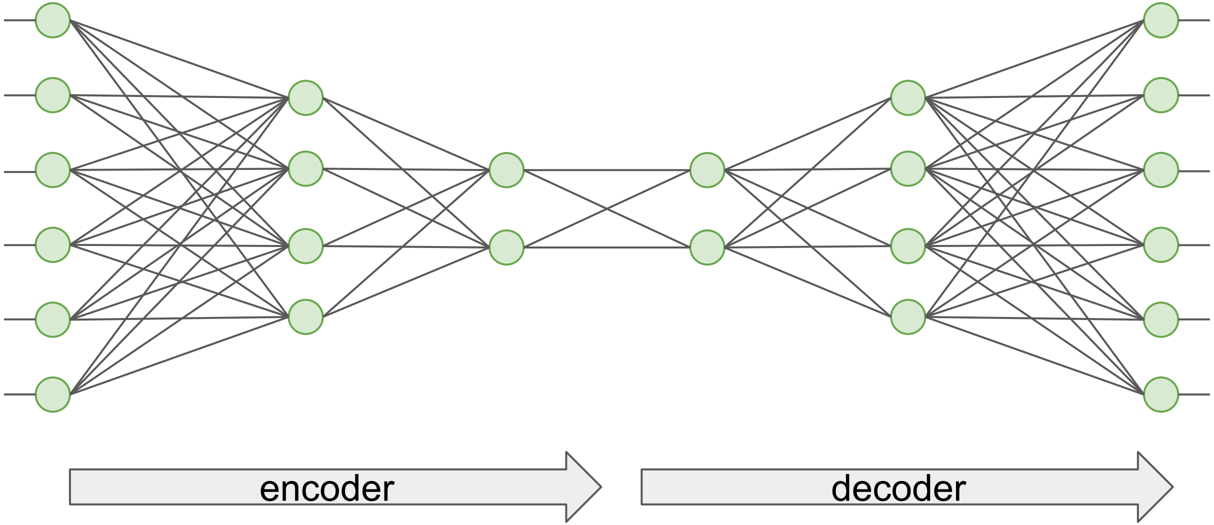


AutoEncoders (AE)



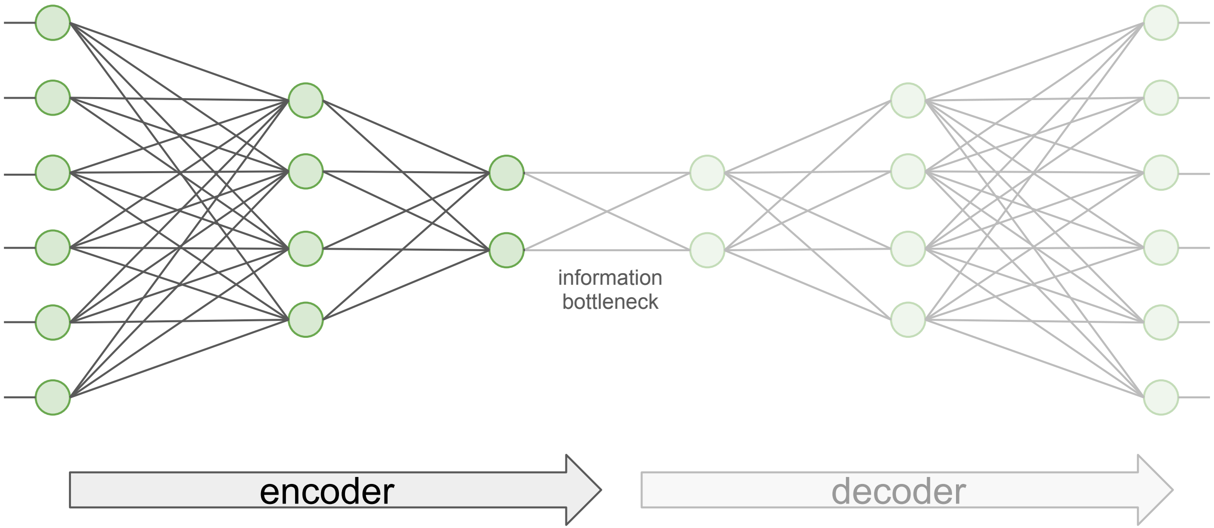
AutoEncoders (AE)

- One thing was inaccurate in the previous depictions
- Autoencoders need to *shrink* the input dimensions to *compress* the input information
- Else they could just learn to copy the information from input directly to output



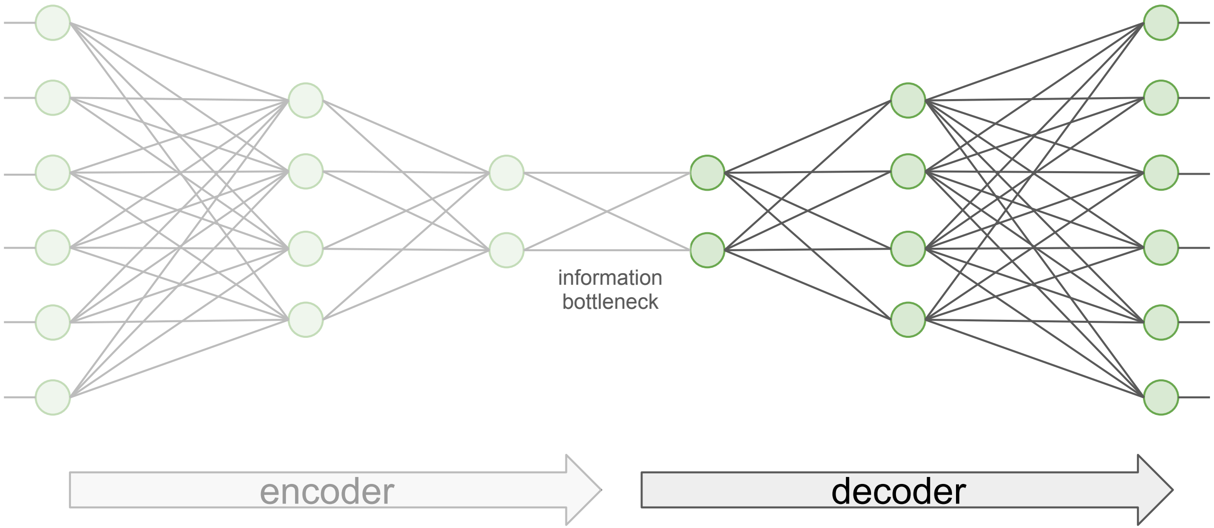
Encoding

- The task of the autoencoder is to essentially *reconstruct* the original input
- The encoder compresses the most **useful information** on how to do this
- Due to the information shrinkage, it can't capture all the details.



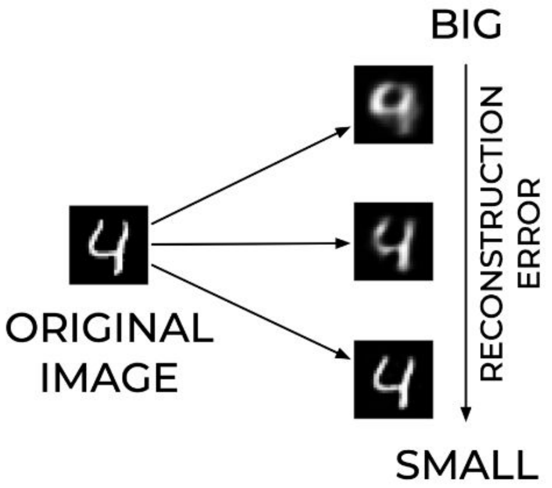
Decoding

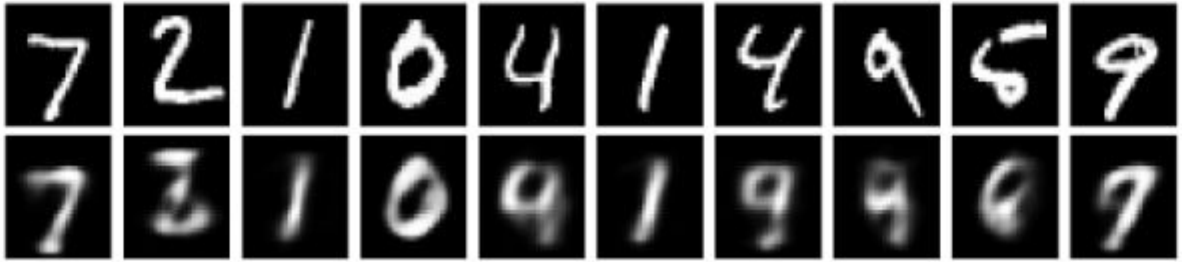
- The decoder takes the compressed vector as its input and needs to fill in all the missing details.
- To do this it needs to understand some fundamental properties about how the input data's underlying distribution.
- This is what makes it generative.



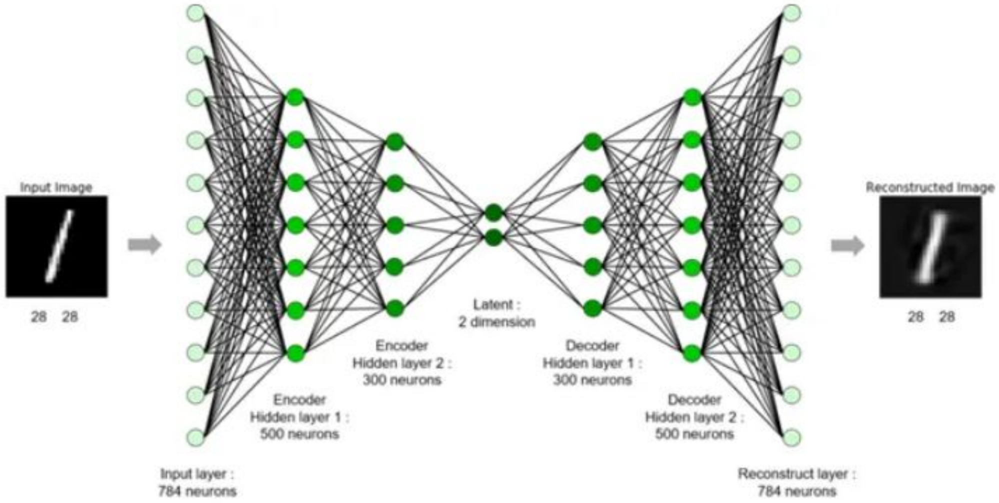
Input Reconstruction

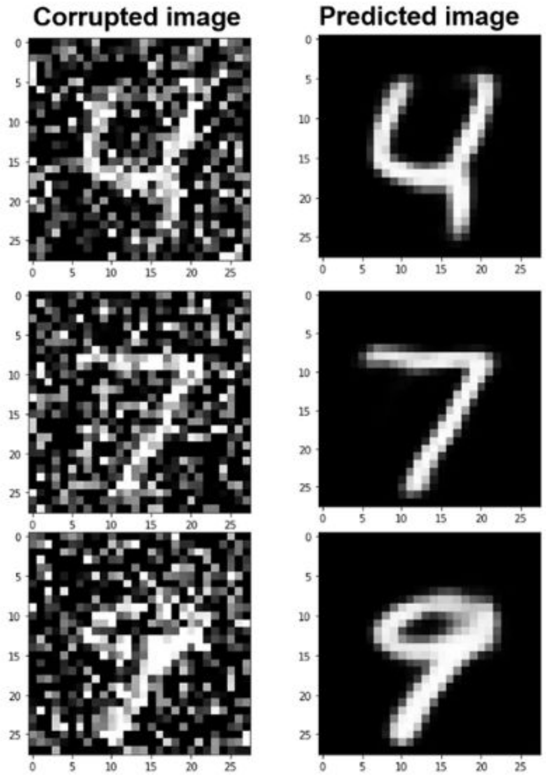
- An autoencoder can work on any modality (images, audio, text, etc.)
- Again, to be able to reconstruct its input, it needs to learn the distribution of its inputs
- … it needs to be generative





Denoising Autoencoders

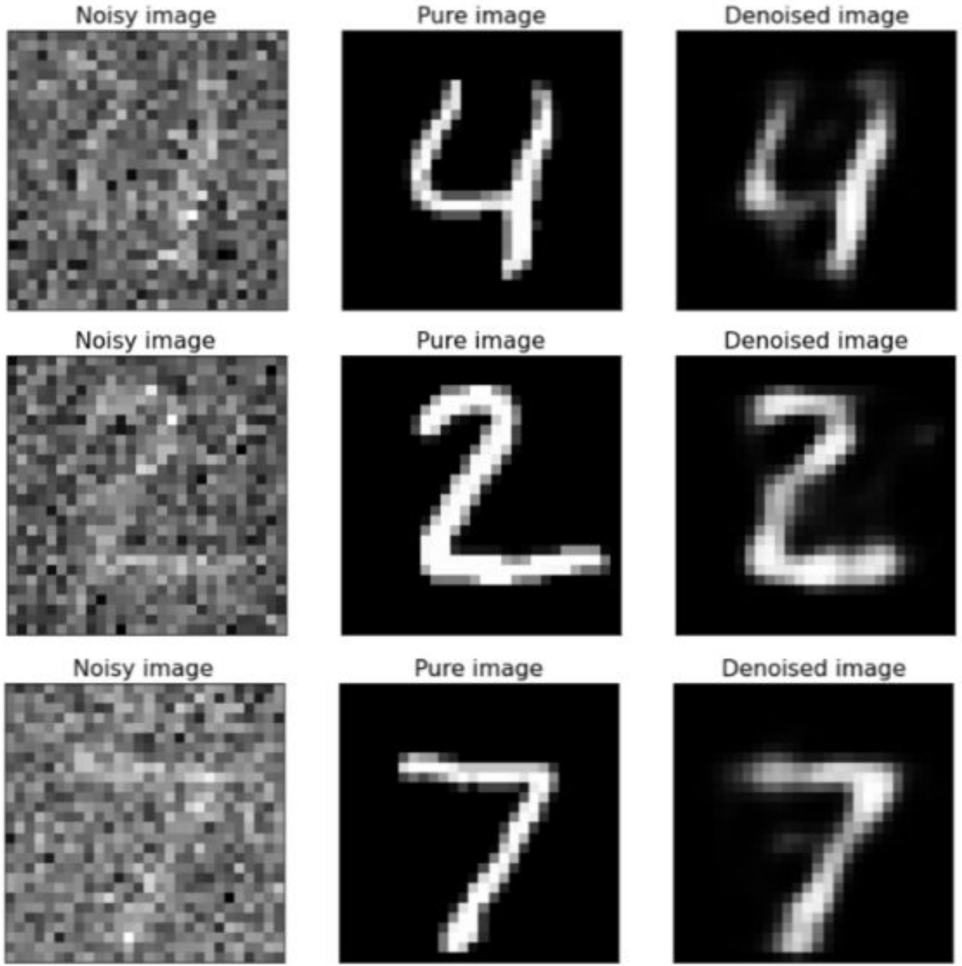




Denoising Autoencoders

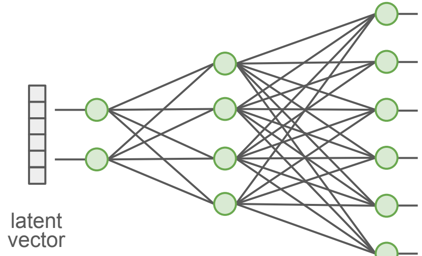
even harder examples…

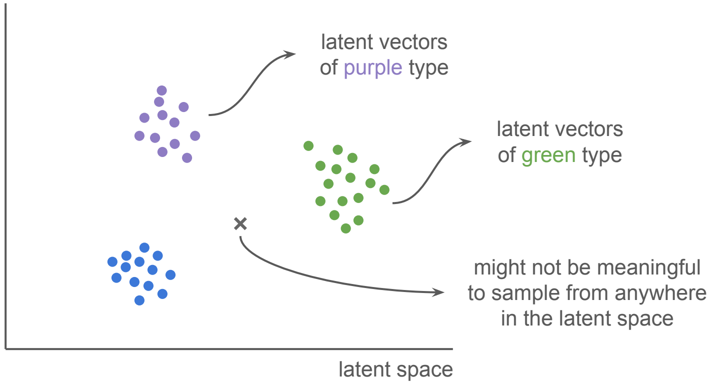
This process forces the autoencoder to look beyond the input noise and learn the true underlying distribution



Using Autoencoders for Generative tasks

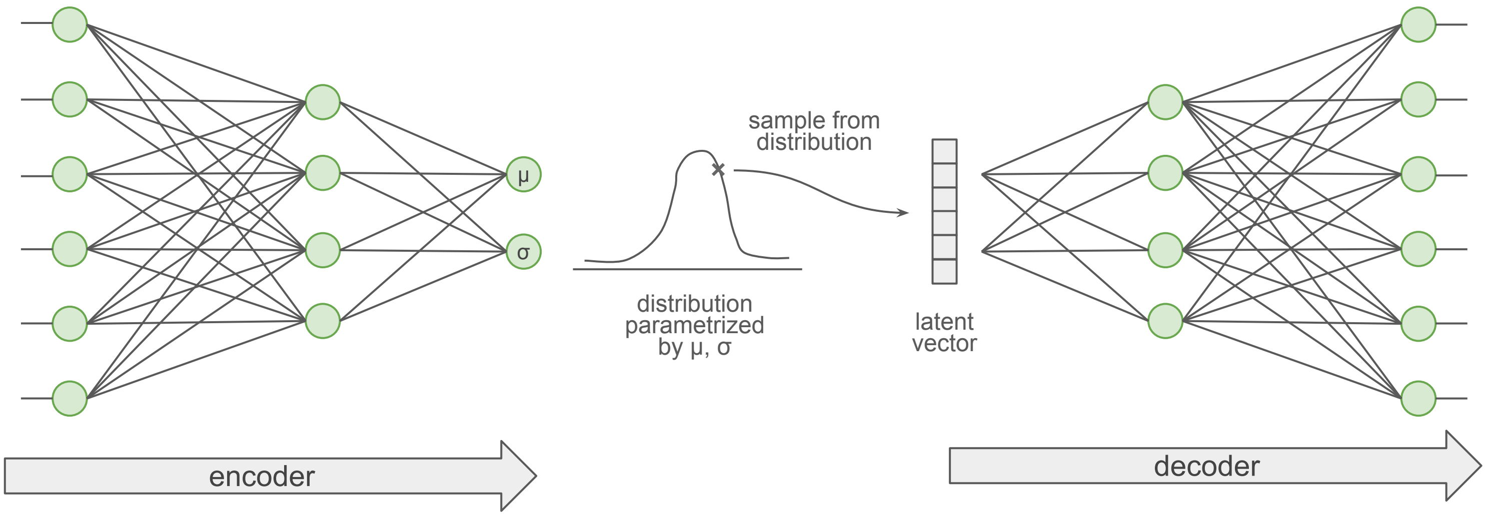
- So, we know that the decoding part of the AE is generative.
- What do we need to do to make it generate something?
  - ➢ supply it with a latent vector
- How do we know what values to choose in the latent vector?
  - ➢ we can't
  - ➢ can we randomly choose a vector?





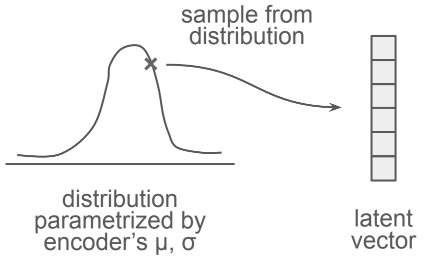
Variational AutoEncoders (VAE)

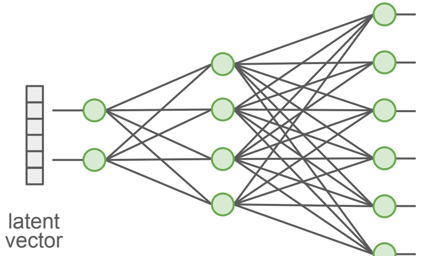
- Idea: instead of learning the latent vectors deterministically, learn the distribution from which they are sampled.
- Encoder will learn the parameters of this distribution (e.g. μ, σ in normal distribution)
- Decoder will reconstruct output from a latent vector, sampled from this distribution

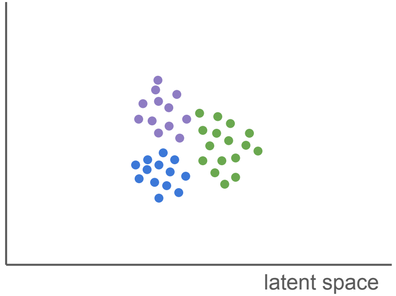


Using VAEs for Generative tasks

- So, how does a VAE help us with our generative tasks?
  - ➢ In deterministic AEs we didn't know how to select the latent vector
  - ➢ In VAEs, we know the distribution from which latent vectors can be sampled
- Generative process:
  - ➢ Step 1: sample a latent vector from the distribution
  - ➢ Step 2: supply latent vector to the decoder
- VAE's training objective leads them to have a more continuous latent space

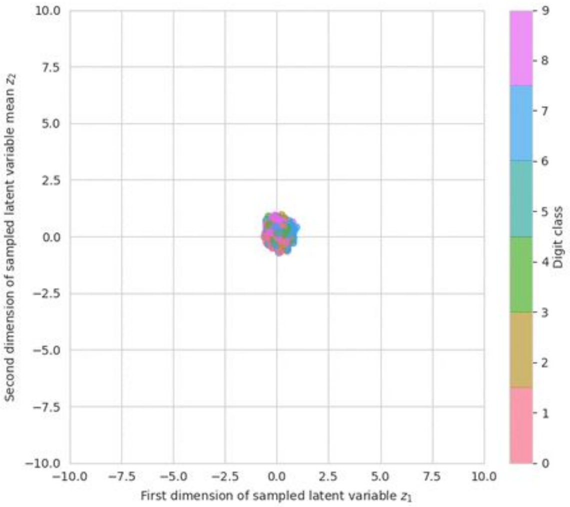


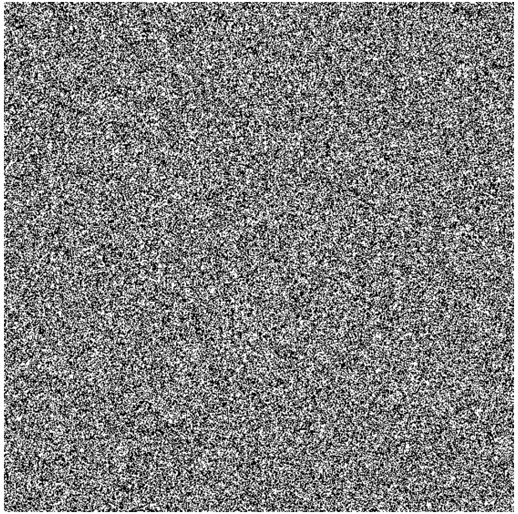




VAE: training details

- Trained with ELBO loss function. Two terms:
    - Reconstruction loss
    - Regularization term to constrain latent distribution to “follow” a prior (e.g. gaussian)





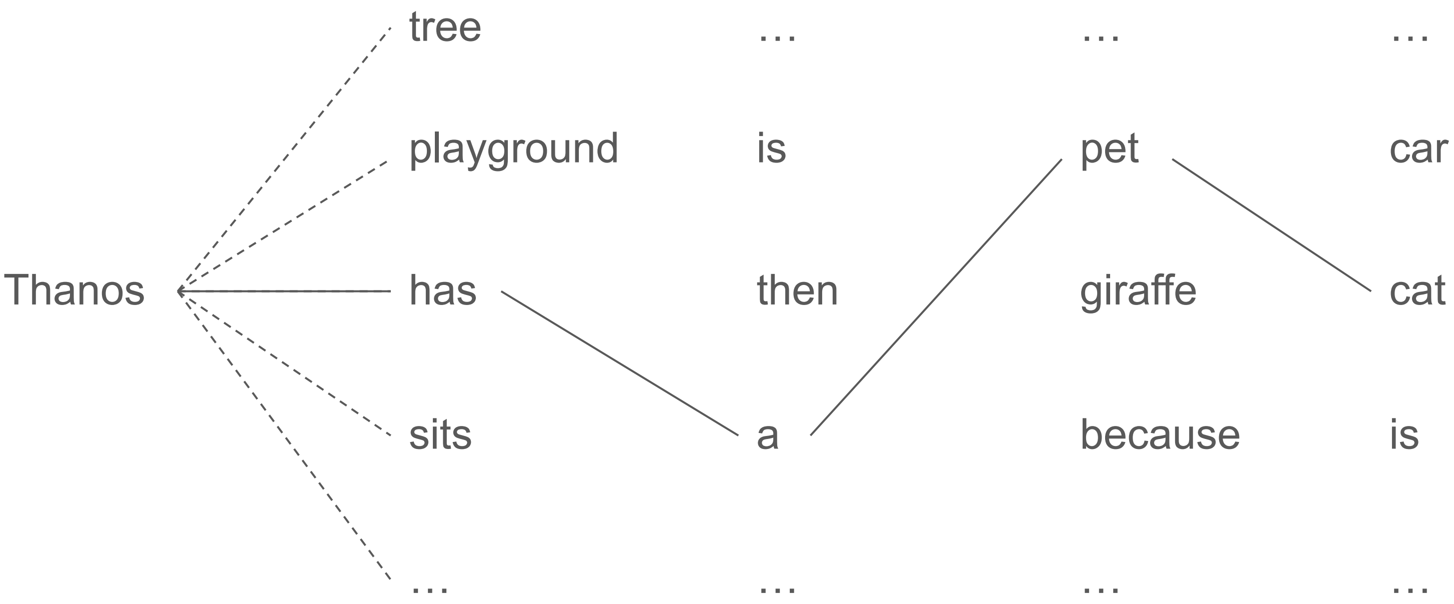
More Advanced Examples

Language Models

Are language models generative?

- Yes!
- They are trained in an unsupervised fashion to **predict the next word in a sequence**.
- This allows them to learn how words are conditioned upon one another in language.
- During inference they generate the output words, conditioned upon their inputs.

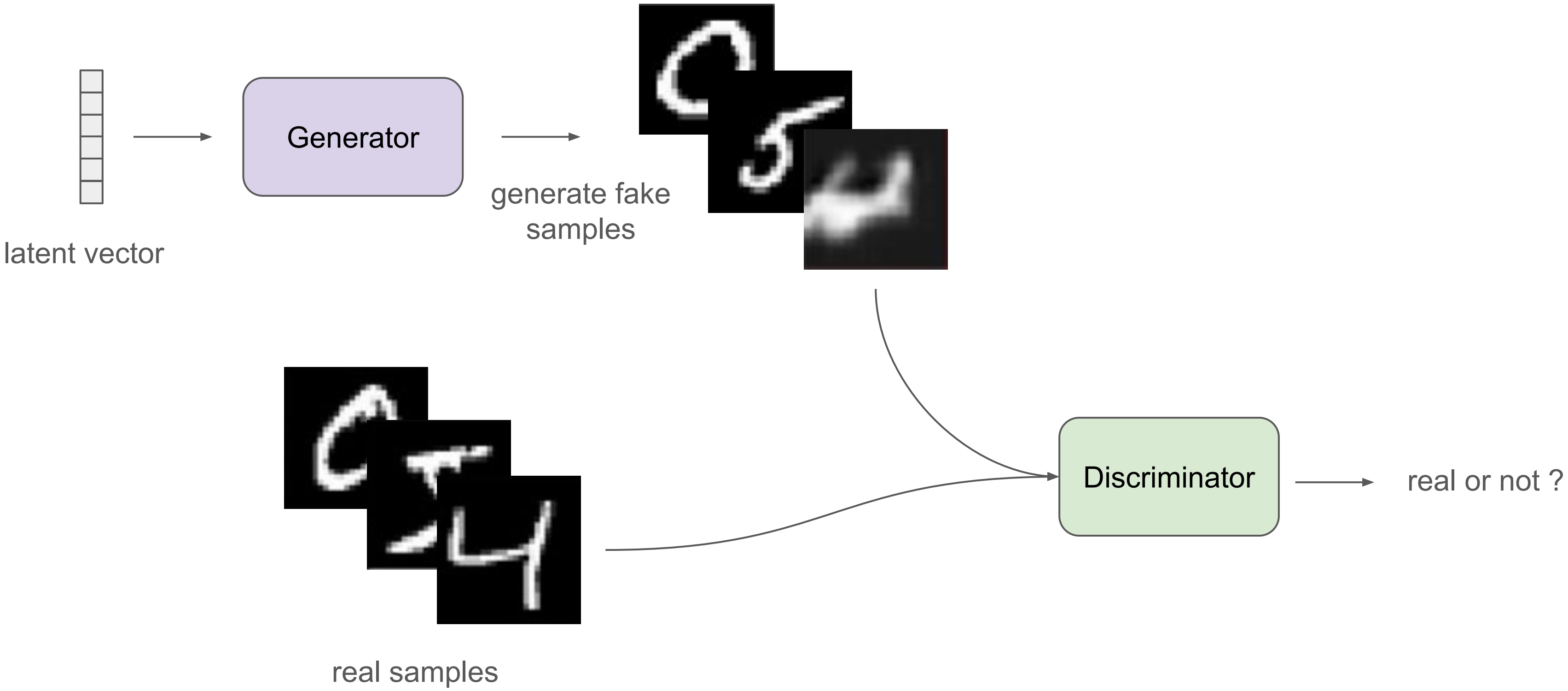
Language Models



Generative Adversarial Networks (GAN)

- Framework for training a NN in an unsupervised manner to achieve generative capabilities
- Consists of two NNs:
  - Generator: learns to generate realistic samples
  - Discriminator: learns to discriminate between real and fake samples
- These contest with each other in the form of a zero-sum game, where one agent's gain is another agent's loss

Generative Adversarial Networks



Diffusion Models

- Process for training generative models
- Consider a diffusion process iteratively adding gaussian noise to the input samples
- Model is trained to reverse this process
- After training model is capable of generating realistic sample completely from random noise