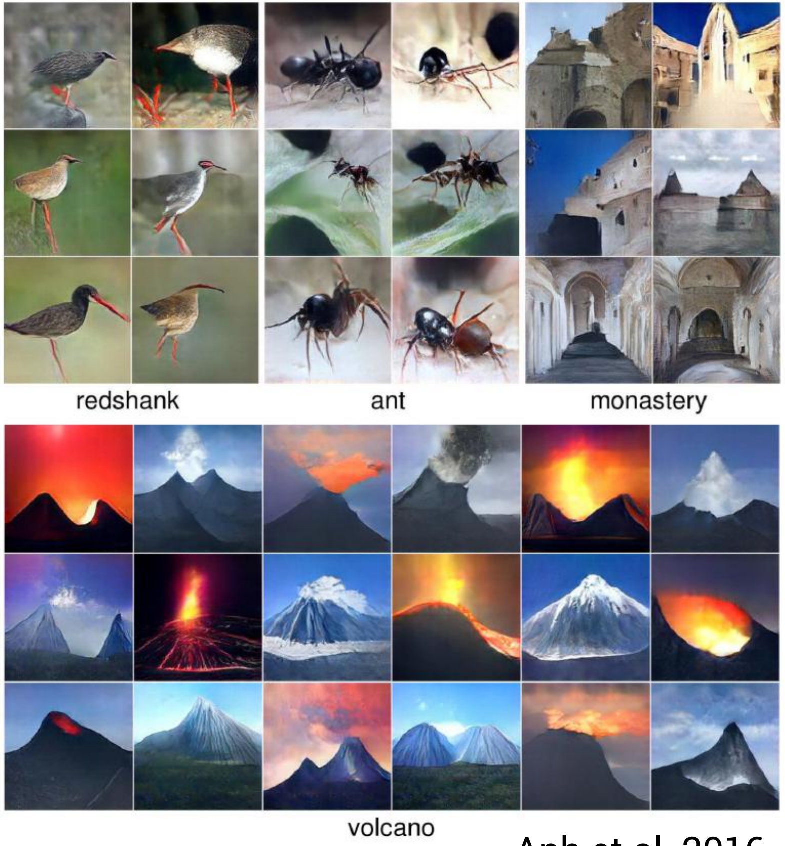
Deep Generative Models

Ishaan Gulrajani

Learning to generate

Images



Anh et al. 2016

Learning to generate

Images

Speech

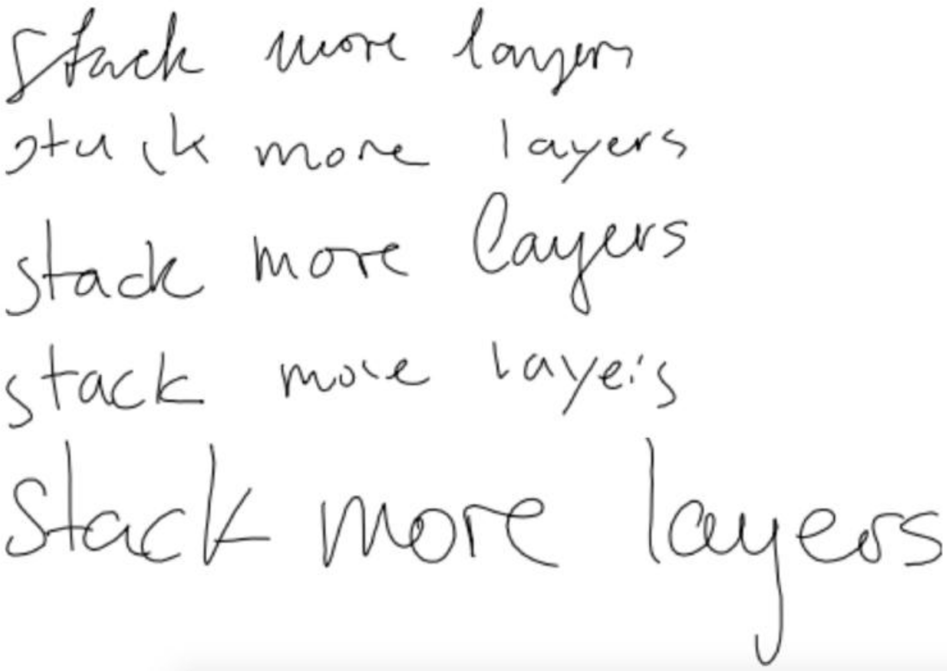
van der Oord et al. 2016

Learning to generate

Images

Speech

Handwriting



Graves 2013

Learning to generate

Images

Speech

Handwriting

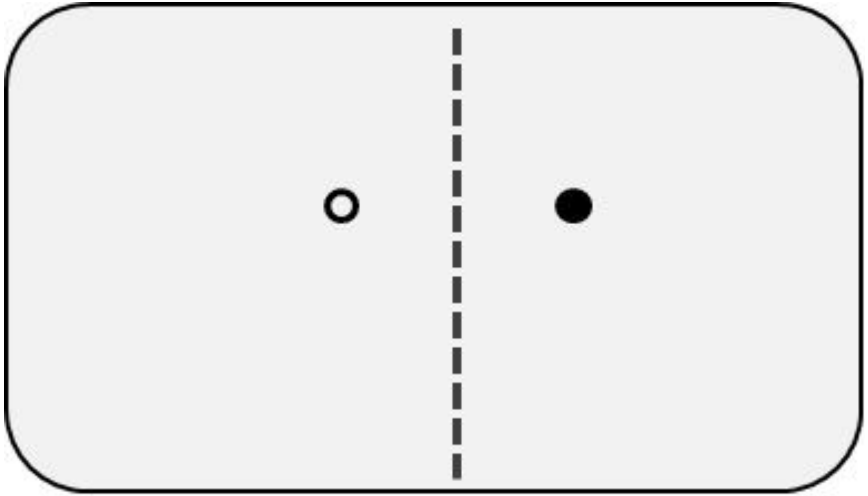
Language

Human: *i am seeing an error related to vpn*
Machine: *what is the error message that you are getting when connecting to vpn using network connect ?*
Human: *connection refused or something like that*
Machine: *may i know the version of network connect you connect ?*
Human: *i am not sure i know that*
Machine: *is the network connect prompting for any user certificate*
Machine: *?*
Human: *yes*
Machine: *are you at home or at work at the moment ?*
Machine: *office ?*
Human: *i am at home*
Machine: *try this*
Machine: *goto <URL> page and enter your name and password*
Human: *done*
Machine: *try logging in with <NAME> and check if its working fine now*
Human: *yes , now it works !*

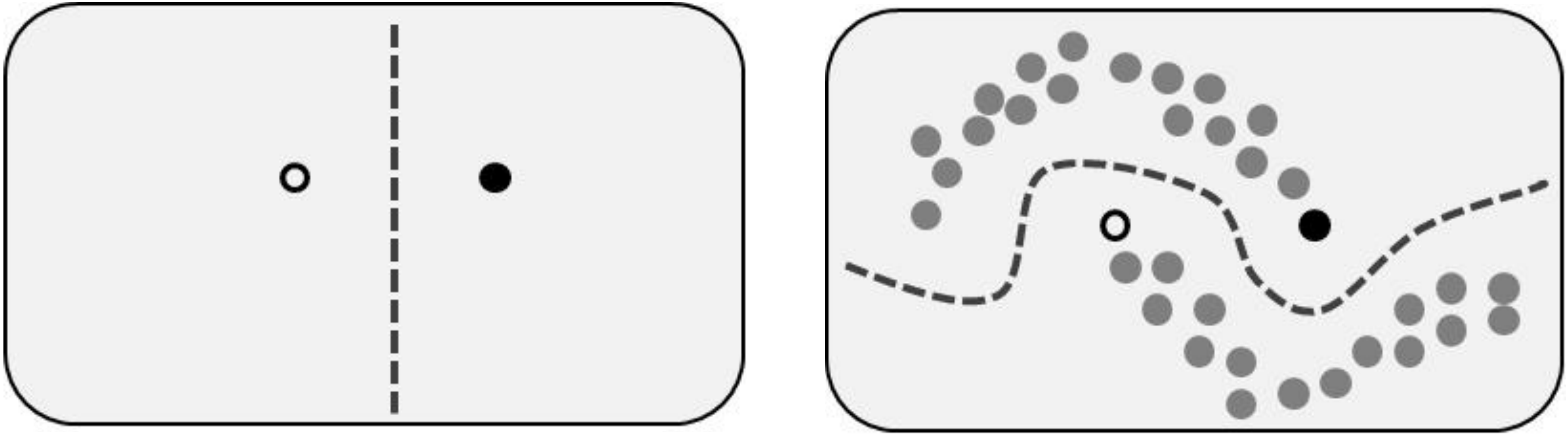
Vinyals et al. 2015

idea: learn to *understand* data through generation

Generative models for *classification*



Generative models for *classification*



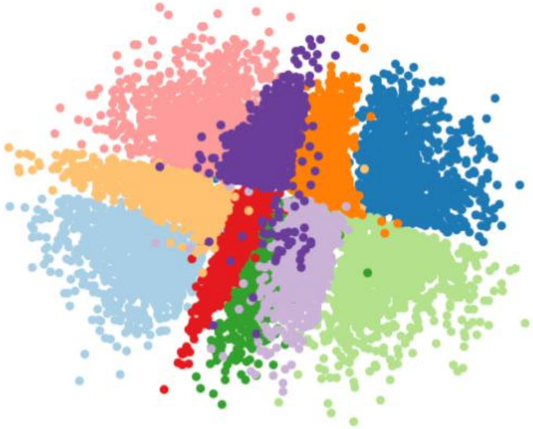
Generative models for *representation learning*

(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation)

(a) Varying c_1 on InfoGAN (Digit type)

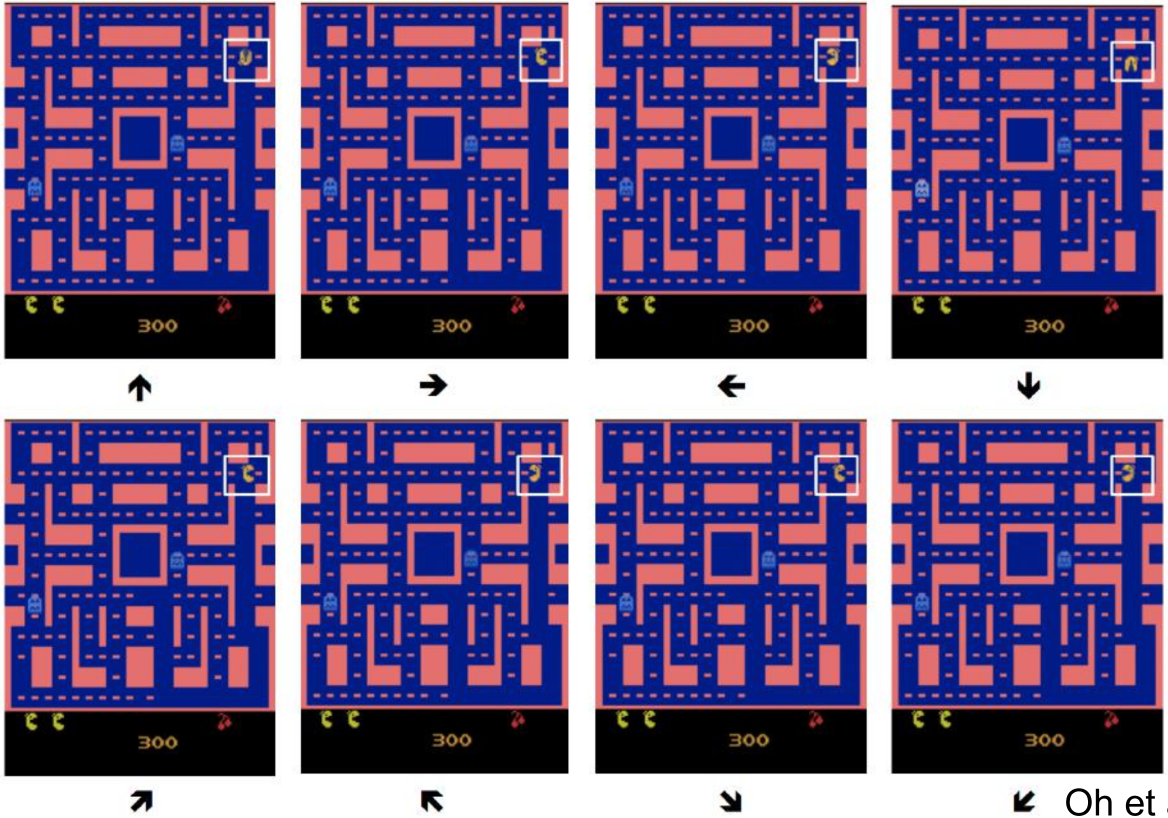
(d) Varying c_3 from -2 to 2 on InfoGAN (Width)

Chen et al. 2016



Sønderby et al. 2016

Generative models for *simulation, planning, reasoning*



Oh et al. 2015

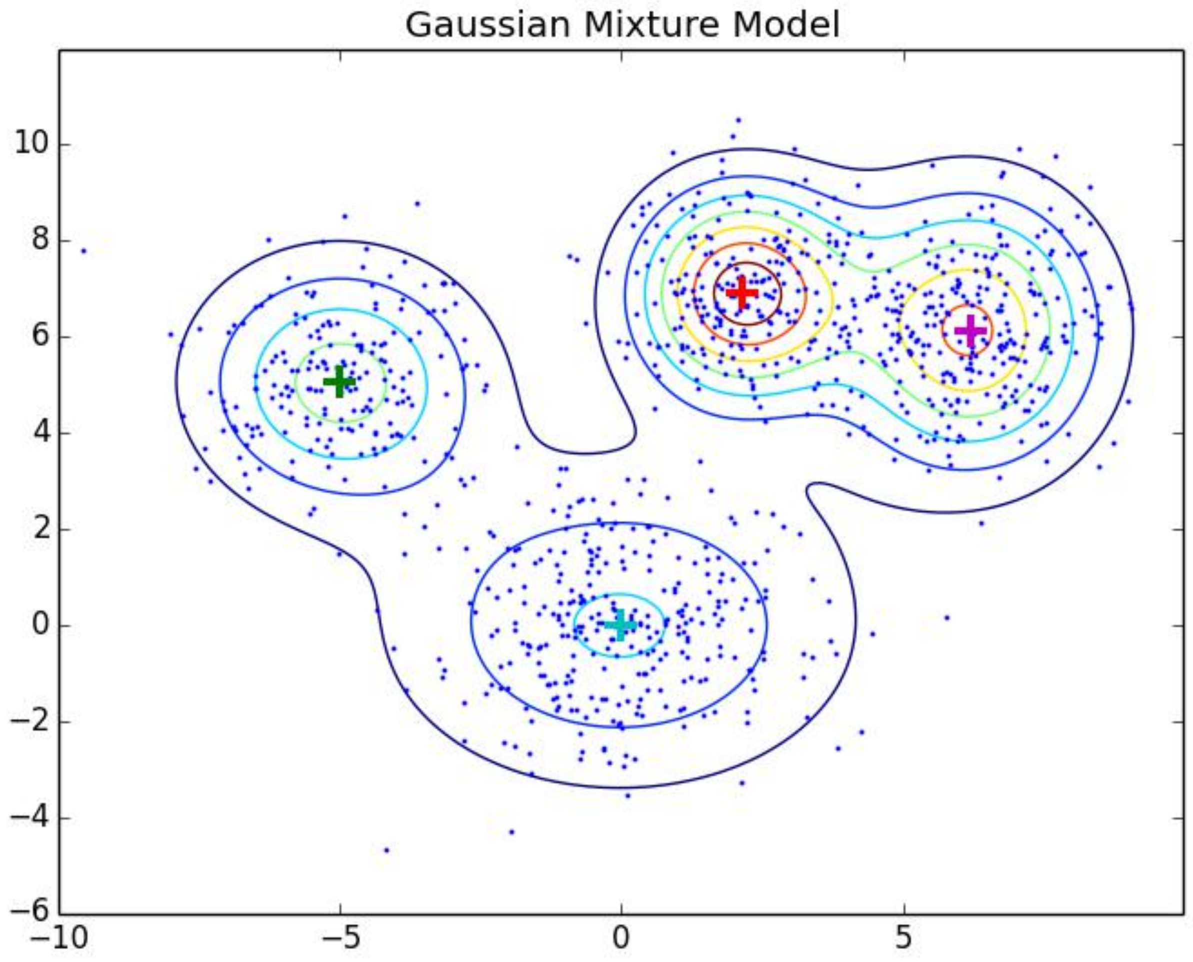
Setup

Discriminative model: given *n* examples $(x^{(i)}, y^{(i)})$
learn $h : X \to Y$

Generative model: given *n* examples $x^{(i)}$, recover $p(x)$

Maximum-likelihood objective: $\prod_i p_\theta(x) = \sum_i \log p_\theta(x)$

Generation: Sampling from $p_\theta(x)$


Gaussian Mixture Model
10
8
6
4
2
0
−2
−4
−6
−10
−5
0
5

Attempt 1: learn $p_\theta(x)$ directly

Attempt 1: learn $p_\theta(x)$ directly

Problem: We need to enforce that $\int_x p_\theta(x)dx = 1$

For most models (i.e. neural networks) this integral is intractable.

Autoregressive Models

Factorize dimension-wise:

$$p(x) = p(x_1)p(x_2|x_1)\dots p(x_n|x_1,\dots,x_{n-1})$$

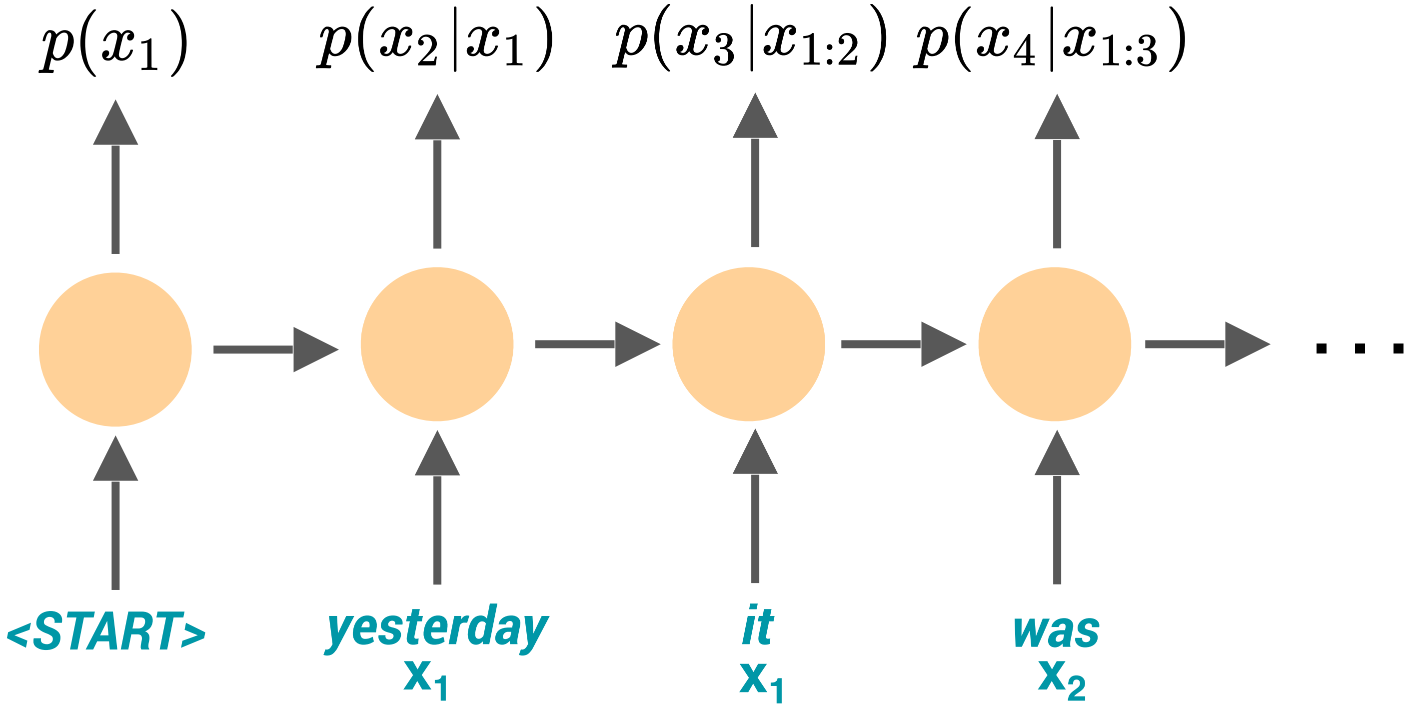
Build a “next-step prediction” model $p(x_n|x_1,\dots,x_{n-1})$

If x is **discrete**, network outputs a probability for each possible value

If x is **continuous**, network outputs parameters of a simple distribution (e.g. Gaussian mean and variance)… *or just discretize!*

Generation: sample one step at a time, conditioned on all previous steps

RNNs for Autoregressive Language Modeling



PixelRNN (van der Oord et al. 2016)

Autoregressive RNN over pixels in an image

Models pixels as discrete-valued (256-way softmax at each step)

Solution 1: Autoregressive models

Autoregressive models are powerful density estimators, *but:*

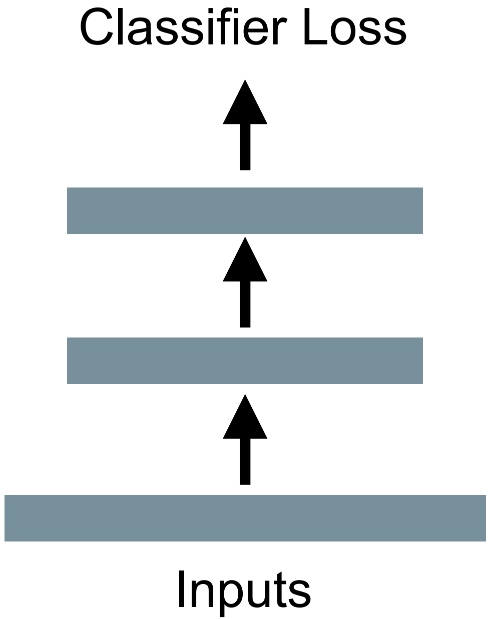
Sequential generation can be slow

Doesn’t closely reflect the “true” generating process

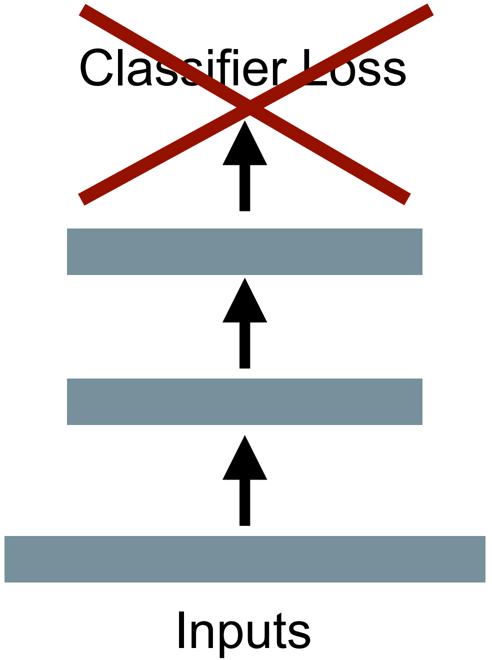
Tends to emphasize details over global data

Not very good for learning representations

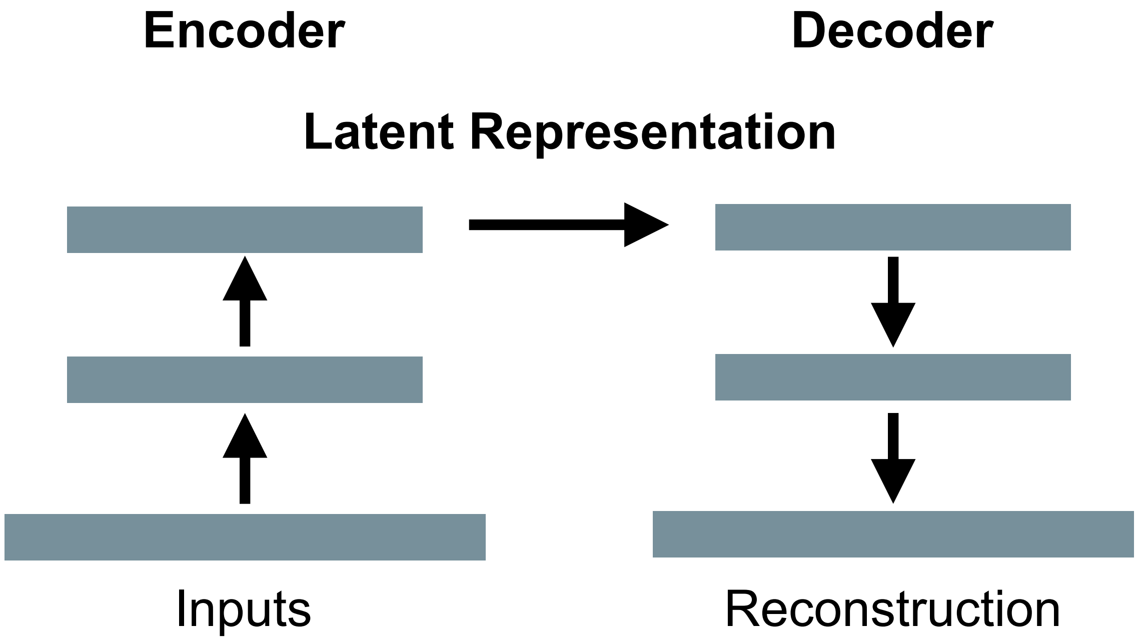
Autoencoders for Representation Learning



Autoencoders for Representation Learning

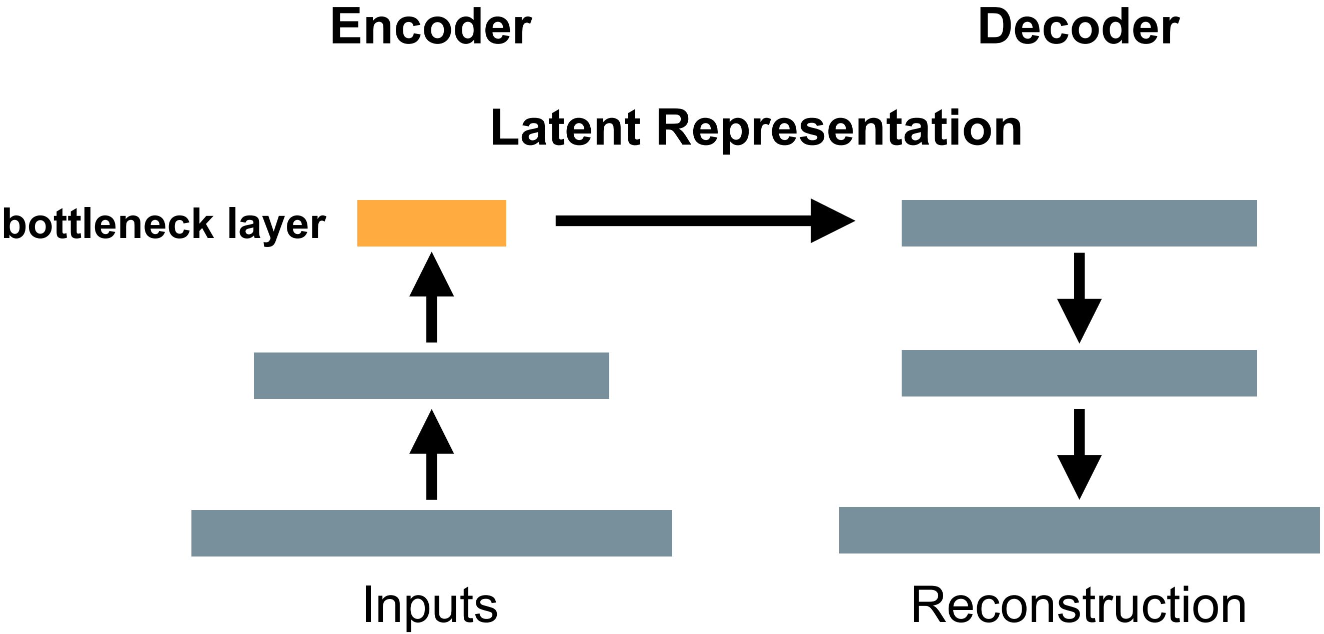


Autoencoders for Representation Learning



$$L = (x - \hat{x})^2$$

Autoencoders for Representation Learning



$$L = (x - \hat{x})^2$$

Autoencoders for Representation Learning

Reconstruction loss forces hidden layer to represent information about the input

Bottleneck hidden layer forces network to learn a compressed latent representation

idea: compression as implicit generative modeling

Variational Autoencoders (VAEs)

Generative extension of autoencoders which allow sampling and estimating probabilities

“Latent variables” with fixed prior distribution $p(z)$

Probabilistic encoder and decoder: $q(z|x), p(x|z)$

Trained to maximize a lower bound on log-probability:

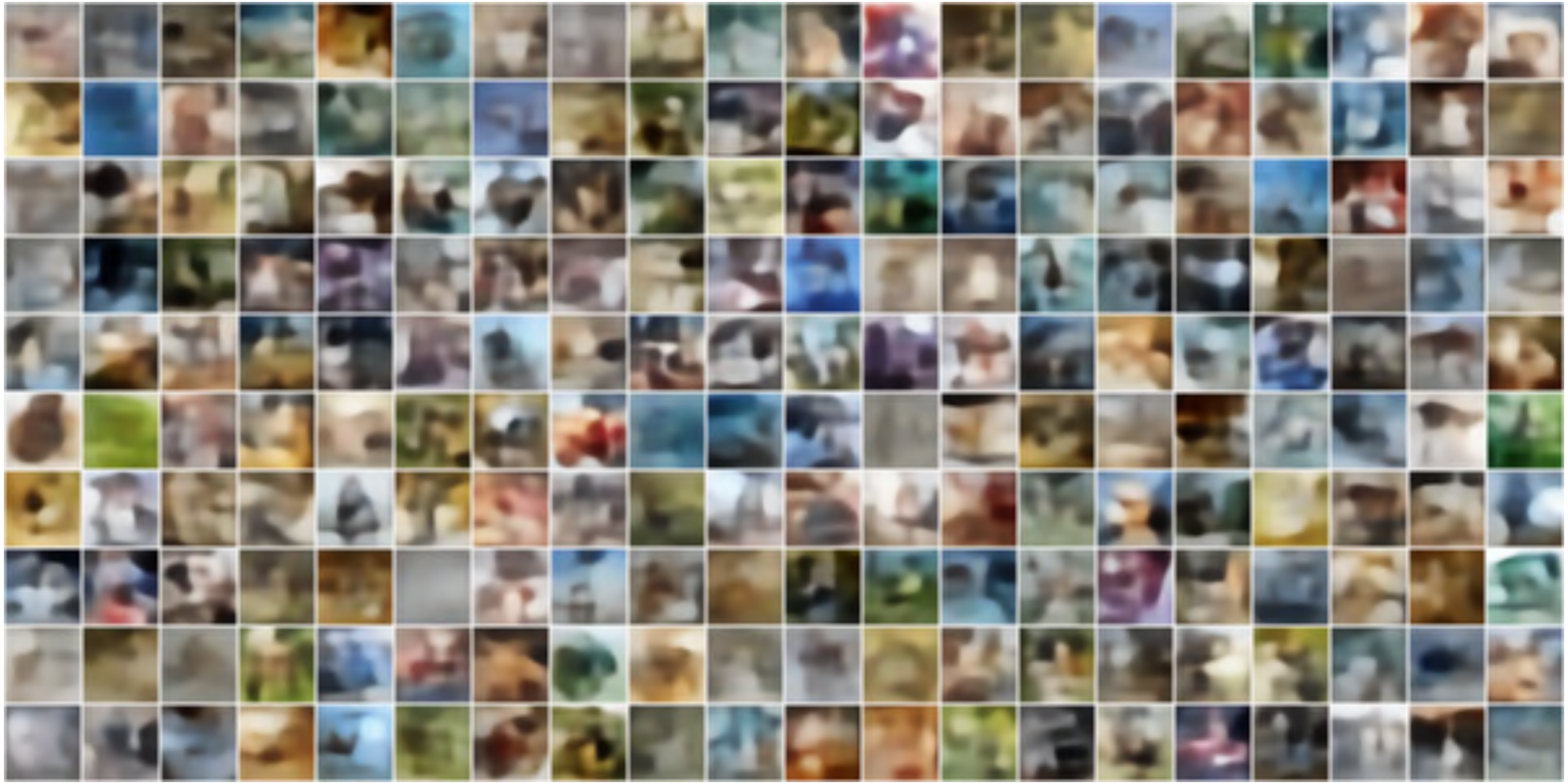
$$\log p(x) \geq \mathbb{E}_{z \sim q(z|x)} [\log p(x|z) + \log p(z) - \log q(z)]$$

Tom White 2016

Problems with VAEs

Encoder and decoder’s output distributions are typically limited (diagonal-covariance Gaussian or similar)

This prevents the model from capturing fine details and leads to blurry generations



Andrej Karpathy 2015

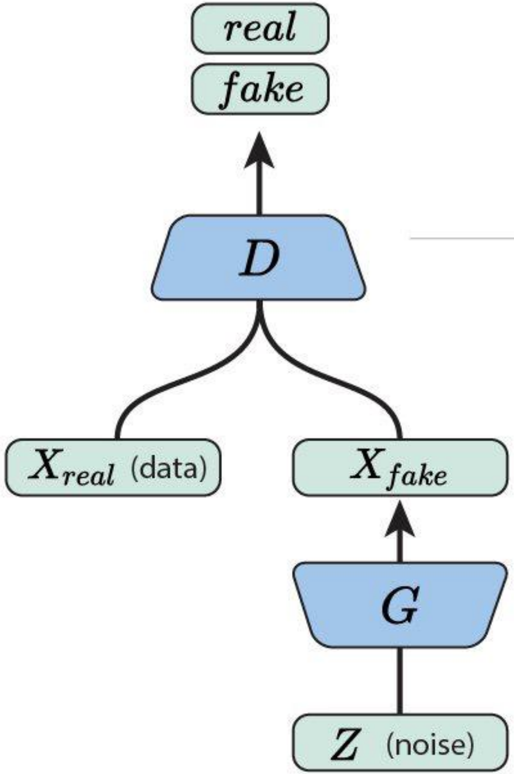
Problems with VAEs

Encoder and decoder’s output distributions are typically limited (diagonal-covariance Gaussian or similar)

This prevents the model from capturing fine details and leads to blurry generations

Solution: use autoregressive networks in encoder and decoder

Generative Adversarial Networks (GANs) are a way to make a generative model by having two neural networks compete with each other.



The **discriminator** tries to distinguish genuine data from forgeries created by the generator.

The **generator** turns random noise into immitations of the data, in an attempt to fool the discriminator.

Chris Olah 2016

redshank

ant

monastery

volcano

We have many orders of magnitude more data than labels; **unsupervised learning is important**.

Generating Implausible Scenes from Captions

A stop sign is flying in blue skies.

A herd of elephants flying in the blue skies.

A toilet seat sits open in the grass field.

A person skiing on sand clad vast desert.

Mansimov et al. 2015