10-423/10-623 Generative AI

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Diffusion Models

Matt Gormley
Lecture 7
Feb. 7, 2024

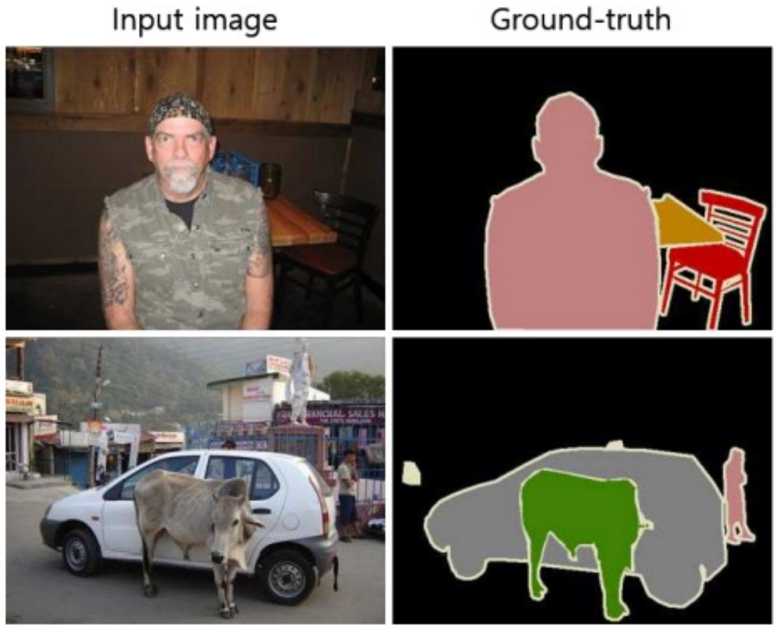
Reminders

- Homework 1: Generative Models of Text
  - Out: Thu, Jan 25
  - Due: Wed, Feb 7 at 11:59pm
- Homework 2: Generative Models of Images
  - Out: Thu, Feb 8
  - Due: Mon, Feb 19 at 11:59pm

U-NET

Semantic Segmentation

- Given an image, predict a label for every pixel in the image
- Not merely a classification problem, because there are strong correlations between pixel-specific labels



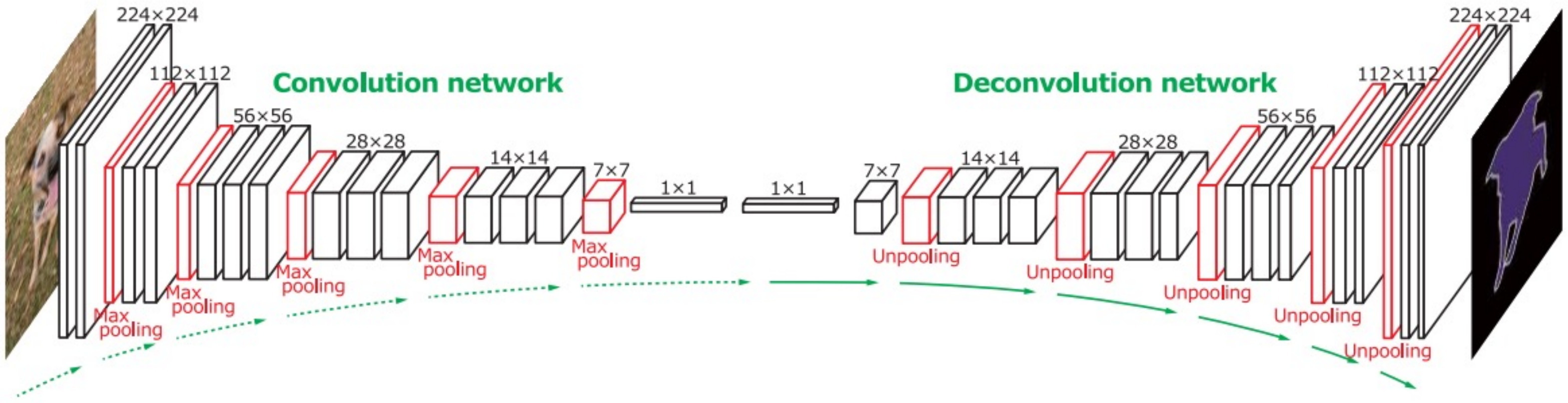


Figure from https://openaccess.thecvf.com/content_iccv_2015/papers/Noh_Learning_Deconvolution_Network_ICCV_2015_paper.pdf

Instance Segmentation

- Predict per-pixel labels as in semantic segmentation, but differentiate between different instances of the same label
- *Example*: if there are two people in the image, one person should be labeled **person-1** and one should be labeled **person-2**

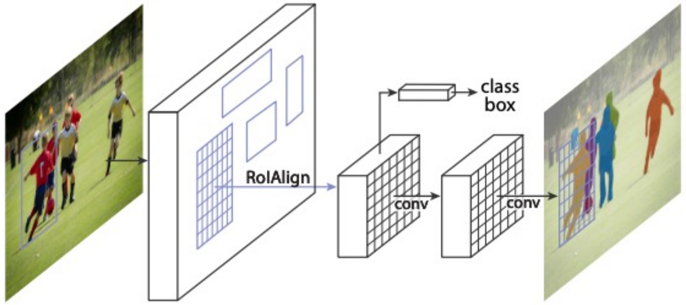
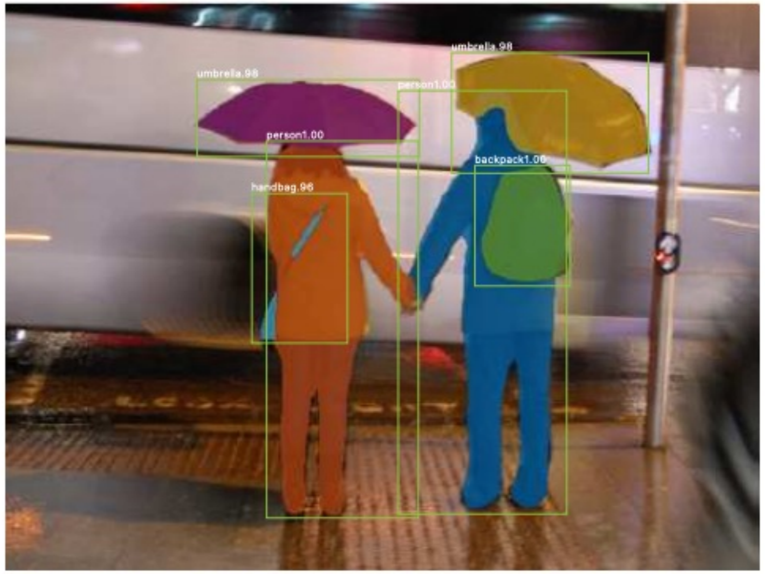


Figure 1. The **Mask R-CNN** framework for instance segmentation.



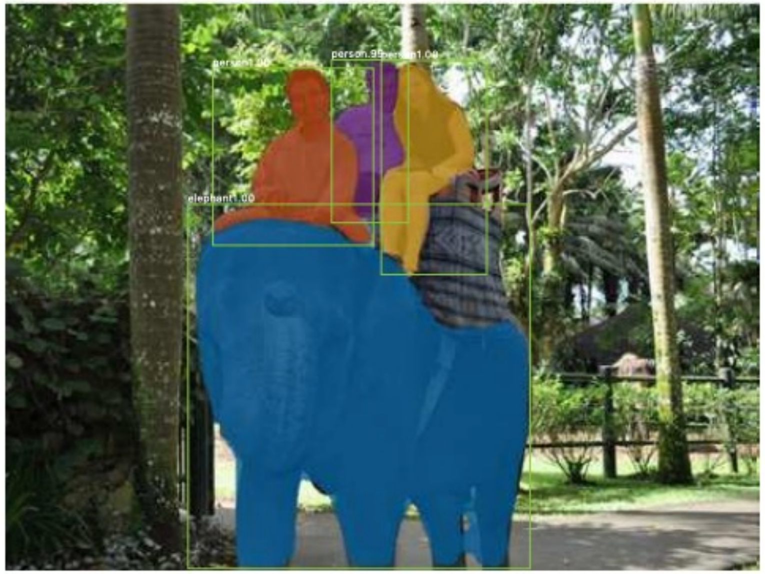


Figure from https://openaccess.thecvf.com/content_ICCV_2017/papers/He_Mask_R-CNN_ICCV_2017_paper.pdf

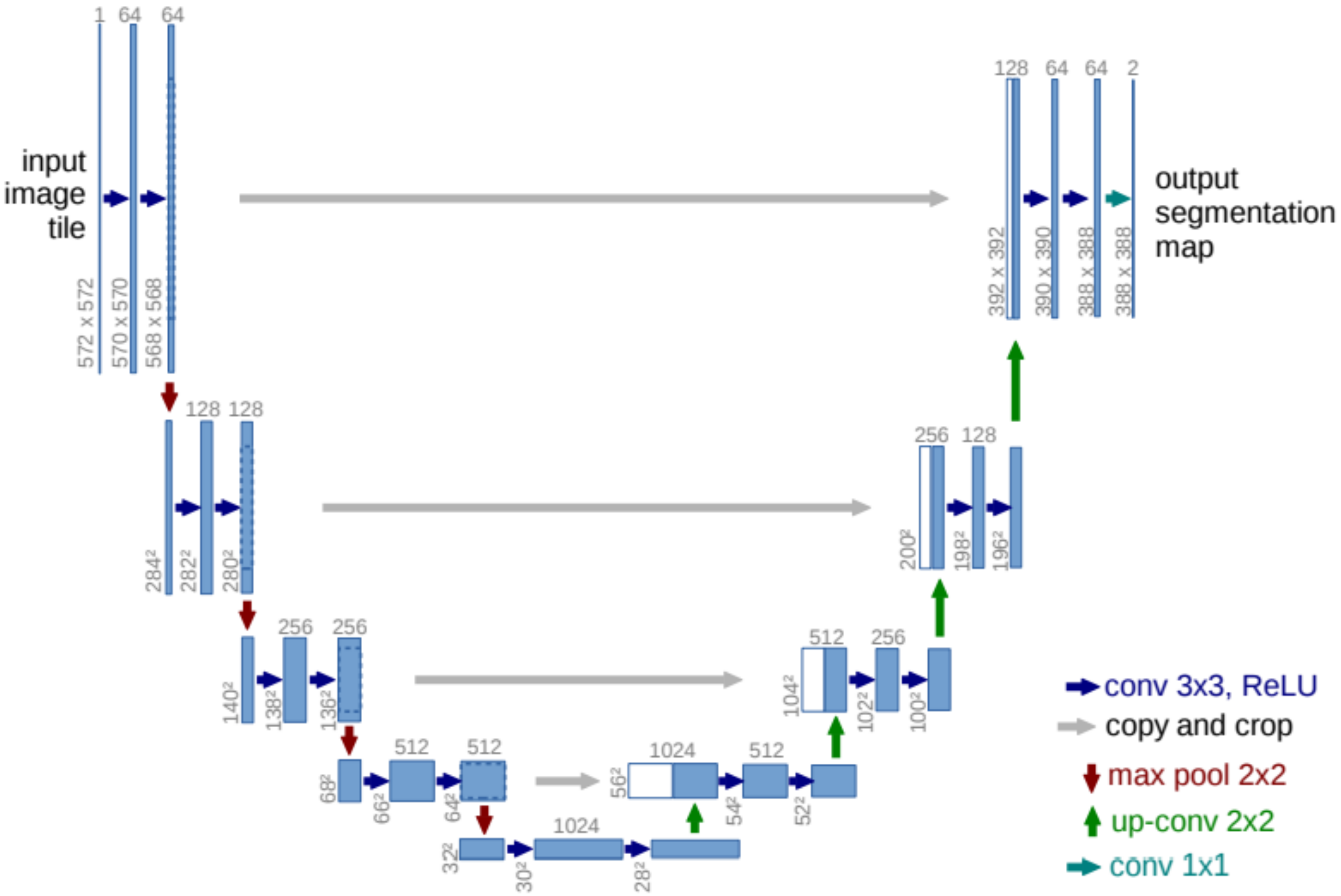
U-Net

Contracting path

- block consists of:
  - 3x3 convolution
  - 3x3 convolution
  - ReLU
  - max-pooling with stride of 2 (downsample)
- repeat the block N times, doubling number of channels

Expanding path

- block consists of:
  - 2x2 convolution (upsampling)
  - concatenation with contracting path features
  - 3x3 convolution
  - 3x3 convolution
  - ReLU
- repeat the block N times, halving the number of channels



U-Net

- Originally designed for applications to biomedical segmentation
- Key observation is that the output layer has the **same** dimensions as the input image (possibly with different number of channels)

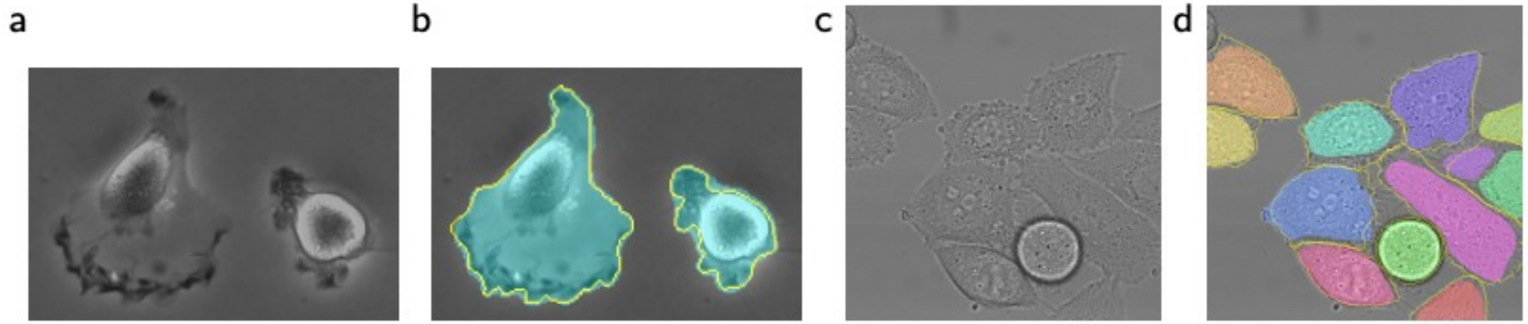


Fig. 4. Result on the ISBI cell tracking challenge. (**a**) part of an input image of the "PhC-U373" data set. (**b**) Segmentation result (cyan mask) with manual ground truth (yellow border) (**c**) input image of the "DIC-HeLa" data set. (**d**) Segmentation result (random colored masks) with manual ground truth (yellow border).

UNSUPERVISED LEARNING

Unsupervised Learning

Assumptions:

1. our data comes from some distribution $q(x_0)$
2. we choose a distribution $p_\theta(x_0)$ for which sampling $x_0 \sim p_\theta(x_0)$ is tractable

Goal: learn θ s.t. $p_\theta(x_0) \approx q(x_0)$

Unsupervised Learning

Assumptions:

1. our data comes from some distribution $q(x_0)$
2. we choose a distribution $p_\theta(x_0)$ for which sampling $x_0 \sim p_\theta(x_0)$ is tractable

Goal: learn θ s.t. $p_\theta(x_0) \approx q(x_0)$

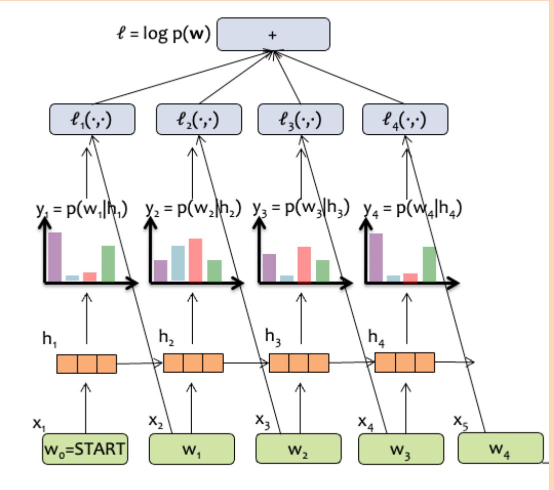
Example: autoregressive LMs

- true $q(x_0)$ is the (human) process that produced text on the web
- choose $p_\theta(x_0)$ to be an autoregressive language model
  - autoregressive structure means that $p(x_t \mid x_1, \ldots, x_{t-1}) \sim \text{Categorical}(.)$ and ancestral sampling is exact/efficient
- learn by finding

  $$\theta \approx \text{argmax}_\theta \log(p_\theta(x_0))$$

  using gradient based updates on

  $$\nabla_\theta \log(p_\theta(x_0))$$

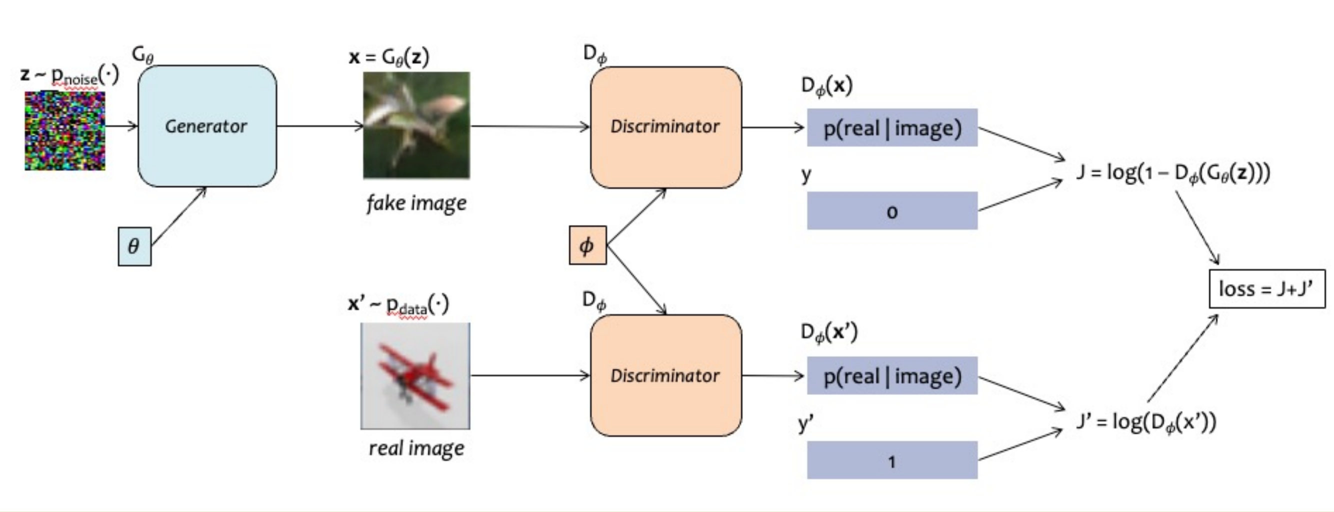


Unsupervised Learning

Assumptions:

1. our data comes from some distribution $q(x_0)$
2. we choose a distribution $p_\theta(x_0)$ for which sampling $x_0 \sim p_\theta(x_0)$ is tractable

Goal: learn θ s.t. $p_\theta(x_0) \approx q(x_0)$



so optimize a minimax loss instead

Example: GANs

- true $q(x_0)$ is distribution over photos taken and posted to Flikr
- choose $p_\theta(x_0)$ to be an expressive model (e.g. noise fed into inverted CNN) that can generate images
  - sampling is typically easy: $z \sim N(0, I)$ and $x_0 = f_\theta(z)$

learn by finding $\theta \approx \text{argmax}_\theta \log(p_\theta(x_0))$?

- No! Because we can't even compute $\log(p_\theta(x_0))$ or its gradient
- Why not? Because the integral is intractable even for a simple 1-hidden layer neural network with nonlinear activation

$$p(x_0) = \int_z p(x_0 \mid z) p(z) dz$$

Unsupervised Learning

Assumptions:

1. our data comes from some distribution $q(x_0)$
2. we choose a distribution $p_\theta(x_0)$ for which sampling
   $x_0 \sim p_\theta(x_0)$ is tractable

Goal: learn θ s.t. $p_\theta(x_0) \approx q(x_0)$

Example: Diffusion Models

- true $q(x_0)$ is distribution over photos taken and posted to Flikr
- choose $p_\theta(x_0)$ to be an expressive model (e.g. noise fed into inverted CNN) that can generate images
  - sampling is will be easy
- learn by finding $\theta \approx \text{argmax}_\theta \log(p_\theta(x_0))$?
  - Sort of! We can't compute the gradient $\nabla_\theta \log(p_\theta(x_0))$
  - So we instead optimize a variational lower bound (more on that later)

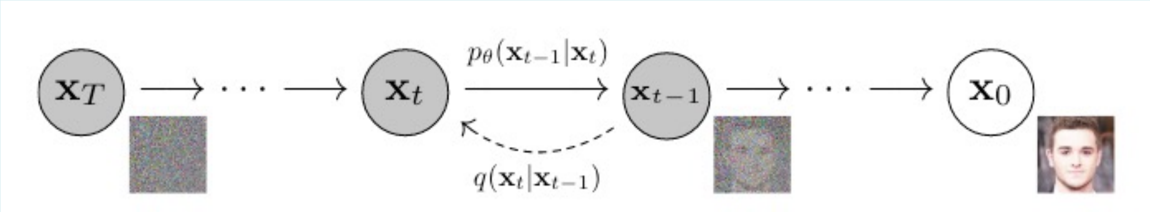


Figure from Ho et al. (2020)

Latent Variable Models

- For GANs, we assume that there are (unknown) **latent variables** which give rise to our observations
- The **noise vector z** are those latent variables
- After learning a GAN, we can **interpolate** between images in latent **z** space

Figure 4: Top rows: Interpolation between a series of 9 random points in Z show that the space learned has smooth transitions, with every image in the space plausibly looking like a bedroom. In the 6th row, you see a room without a window slowly transforming into a room with a giant window. In the 10th row, you see what appears to be a TV slowly being transformed into a window.

Figure from Radford et al. (2016)

DIFFUSION MODELS

Diffusion Models

- Next we will consider (1) **diffusion mo**
variational autoencoders (VAEs)
  - Although VAEs came first, we're going to
models since they will receive more of ou
- The steps in defining these models is
  - Define a probability distribution involvin
  - Use a variational lower bound as an obje
  - Learn the parameters of the probabi
the objective function
- So what is a variational lower bound?

The standard presentation of diffusion models requires an understanding of variational inference. (we'll do that next time)

Today, we'll do an alternate presentation without variational inference!

Diffusion Model

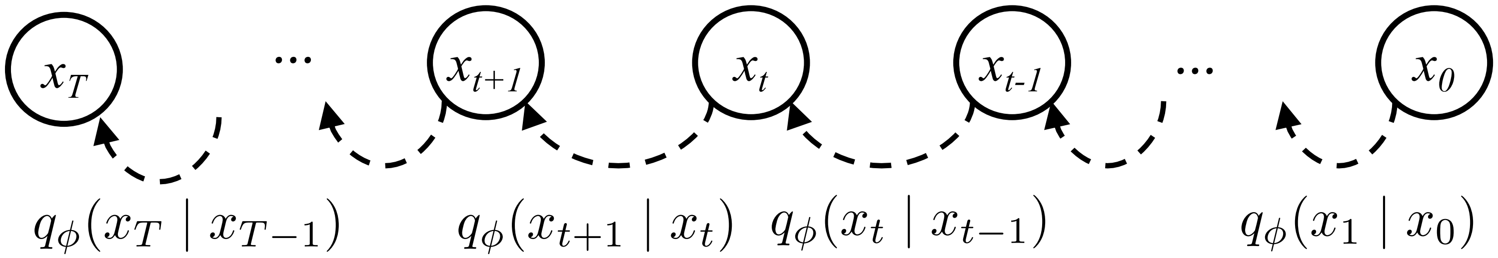
Question:
Which are the latent variables in a diffusion model?

Answer:

Define a very simple forward process for adding noise to data:

$$q_\phi(x_{1:T}) = q(x_0) \prod_{t=1}^{T} q_\phi(x_t \mid x_{t-1})$$

where $q(x_0)$ is the data distribution and $q_\phi(x_t \mid x_{t-1})$ is some simple/tractable distribution (e.g. Gaussian).



The *exact* reverse process requires inference:

$$q_\phi(x_{1:T}) = q_\phi(x_T) \prod_{t=1}^{T} q_\phi(x_{t-1} \mid x_t)$$

And, even though $q_\phi(x_t \mid x_{t-1})$ is simple, computing $q_\phi(x_{t-1} \mid x_t)$ is intractable! Why? Because $q(x_0)$ might be not-so-simple.

Diffusion Models

Whiteboard:

1. probabilistic definition of diffusion model
   (forward process and reverse process)
2. Gaussian conditionals for forward/reverse diffusion
3. analogy for learning diffusion model
4. marginals of the forward process
5. learning by matching marginals with the reverse process
6. training algorithms
7. sampling algorithms