Carnegie Mellon University

Lecture 1: Basics of Probabilistic & Generative Modeling

Yutong (Kelly) He

10-799 Diffusion & Flow Matching, Jan 13th, 2026

Modal aws

What is probabilistic modeling?

Probabilistic modeling is to model the world with uncertainty using probabilities (duh)

- E.g. There is a 70% chance that it will snow tomorrow

We usually do probabilistic modeling with

- **Random variables:** The things we are trying to describe (e.g. tomorrow's weather)
    - Can be continuous (image) or discrete (text)
    - Denoted as X, Y, Z, etc
- **Probability Distributions:** How things are and how often do they happen (e.g. weather in Pittsburgh is 50% chance snowy and 50% chance cloudy)
    - Denoted as $p(X)$, $p(x)$, $P(X)$ or $P(x)$

Carnegie Mellon University

Image from The Redmen TV

Key concepts in probability theory

Let's say we have two random variables X and Y

- **Joint distribution:** $p(X, Y)$ (reads "X and Y")
- **Marginal distribution:** $p(X)$ and $p(Y)$
- **Conditional distribution:** $p(X|Y)$and $p(Y|X)$ (reads "X given Y")
- **Bayes theorem:** $p(X|Y) = \frac{p(X,Y)}{p(Y)} = \frac{p(Y|X)p(X)}{p(Y)}$
    - **Prior:** $p(X)$ (what I originally believe about X)
    - **Posterior:** $p(X|Y)$ (what I believe about X now that I've seen Y)
- **Independence:** Knowing X tells you nothing about Y
    - $p(X|Y) = p(X)$ *and* $p(X, Y) = p(X)p(Y)$

Key concepts in probabilistic modeling

The goal of probabilistic modeling is to learn the probability distributions

We can usually describe the probability distributions through some **parameters** (θ)

- Gaussian: mean (μ) and variance (σ^2)
- Poisson: average rate (λ)
- Some complicated distribution: can be parametrized by neural networks (θ)

The goal of probabilistic modeling is to learn the probability distributions

=> The goal now is to learn those parameters given data

(Note: There's also nonparametric probabilistic modeling, but we won't cover them here)

Key concepts in probabilistic modeling

The goal of probabilistic modeling is to learn the probability distributions

We can usually describe the probability distributions through some **parameters** (θ)

- Gaussian: mean (μ) and variance (σ^2)
- Poisson: average rate (λ)
- Some complicated distribution: can be parametrized by neural networks (θ)

The goal now is to learn those parameters given data

We call the probability of data given model parameters as **likelihood** $p(x|\theta)$

i.e. “if my parameters are accurate, how likely is the data that I observe”

What is generative modeling?

Say we have some data (X) and some labels associated with the data (Y)

e.g. X are an image of a bedroom, Y is whether this bedroom is luxurious

Discriminative modeling: the goal is to learn **p(Y|X)** so that we can determine if a bedroom is luxurious given its image

- Image X is always given

Generative modeling: the goal is to learn **p(X,Y)** or **p(X)**, i.e. we are learning what a (luxurious) bedroom should look like

- Image X is not given => need to be able to "imagine" bedrooms

Carnegie Mellon University

Figure from Stanford CS 236 by Prof. Stefano Ermon

Generative modeling

Given a set of data {x} and some prior knowledge & assumptions

- **Data:** samples (e.g. images of bedrooms)
- **Prior knowledge & assumptions:** parametric form, loss function, optimization, etc

We want to learn a probability distribution $p_\theta(x)$ such that

- **Generation:** If we sample a new datapoint from $p_\theta(x)$, it'd look like a "real" sample (e.g. looks like a real image of bedroom)
- **Density estimation:** Given an existing datapoint x, we should be able to assign a probability to it (probability should be high if x looks "real")
- **Unsupervised learning:** We learn everything by just looking at the data

How to train your ~~dragon~~ generative models

Given a set of data {x} and some prior knowledge & assumptions

- **Data:** samples (e.g. images of bedrooms)
- **Prior knowledge & assumptions:** parametric form, loss function, optimization, etc

We want to learn a probability distribution $p_\theta(x)$ such that

- **Generation:** If we sample a new datapoint from $p_\theta(x)$, it'd look like a "real" sample (e.g. looks like a real image of bedroom)
- **Density estimation:** Given an existing datapoint x, we should be able to assign a probability to it (probability should be high if x looks "real")
- **Unsupervised learning:** We learn everything by just looking at the data

Attempt 1: Maximizing likelihood

We call the probability of data given model parameters as **likelihood** $p_\theta(x)$

i.e. “if my parameters are accurate, how likely is the data that I observe”

$\Rightarrow$ If my parameters are accurate, then a “real” datapoint should have high likelihood

$\Rightarrow$ We shall maximize the likelihood of existing data under the learned model

Given dataset $\{x^{(i)}\}$, we want to find

$$\text{argmax}_\theta \sum_i p_\theta(x^{(i)})$$

Attempt 1: Chain rule

Let's consider a single datapoint x for now, and let's say we know that the datapoint x is composed by a bunch of smaller elements (e.g. a sentence is composed by a bunch of tokens, an image is composed by a bunch of pixels)

Say x has K of these elements, we denote each element as x_k , then we can decompose the likelihood of x by chain rule:

$$p_\theta(x) = p_\theta(x_1)\, p_\theta(x_2|x_1) \ldots p_\theta(x_K|x_{K-1}, \ldots x_1)$$

$$\log p_\theta(x) = \sum_k \log p_\theta(x_k^{(i)}|x_{<k}^{(i)})$$

Attempt 1: Autoregressive modeling

Given dataset $\{x^{(i)}\}$,

$$L(\theta) = -\sum_i \sum_k \log p_\theta(x_k^{(i)} | x_{<k}^{(i)})$$

$$= -\sum_i \sum_k (\log p_\theta \left(x_k^{(i)} \middle| x_{<k}^{(i)} \right) \cdot 1 + \sum_{x' \neq x_k^{(i)}} p_\theta(x' | x_{<k}^{(i)}) \cdot 0)$$

which is cross entropy loss when ground truth labels are one-hot

This is LLM!

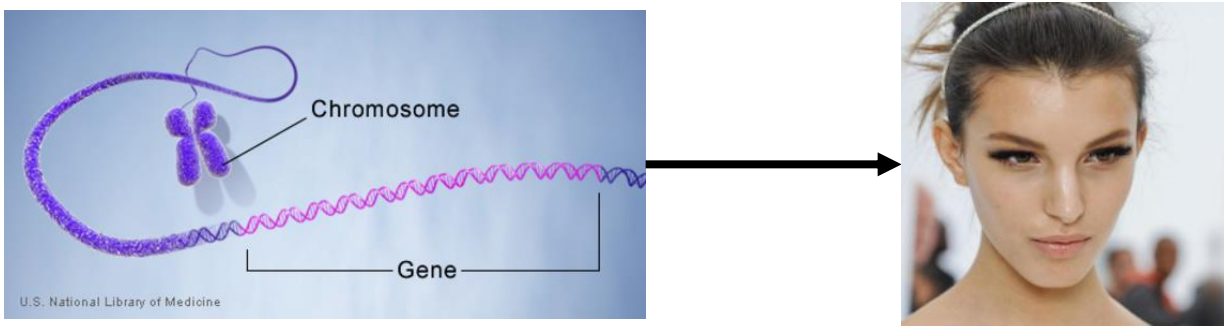
Attempt 2: Latent variables

How a person look is largely determined by their genes

There are a lot of variability of people's looks, but genes are just combinatorial

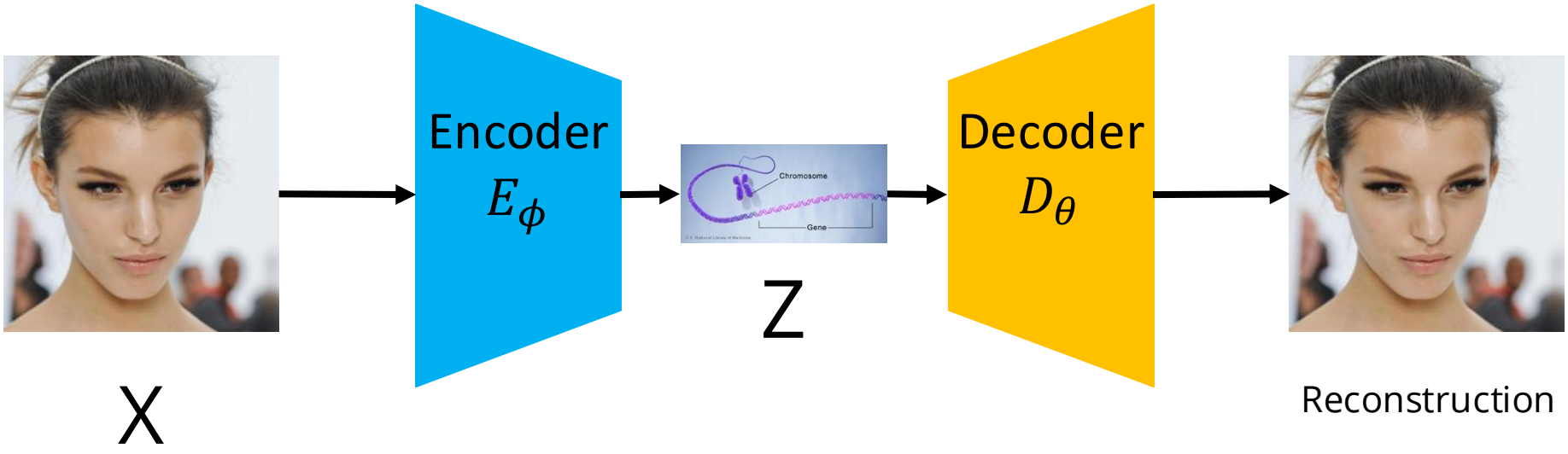
However, we cannot directly observe genes

But can we still take into account of the fact that there is a hidden variable that influence how a person look when modeling human face images?



Figures from U.S. National Library of Medicine and CelebA-HQ

Attempt 2: Variational autoencoder (VAE)



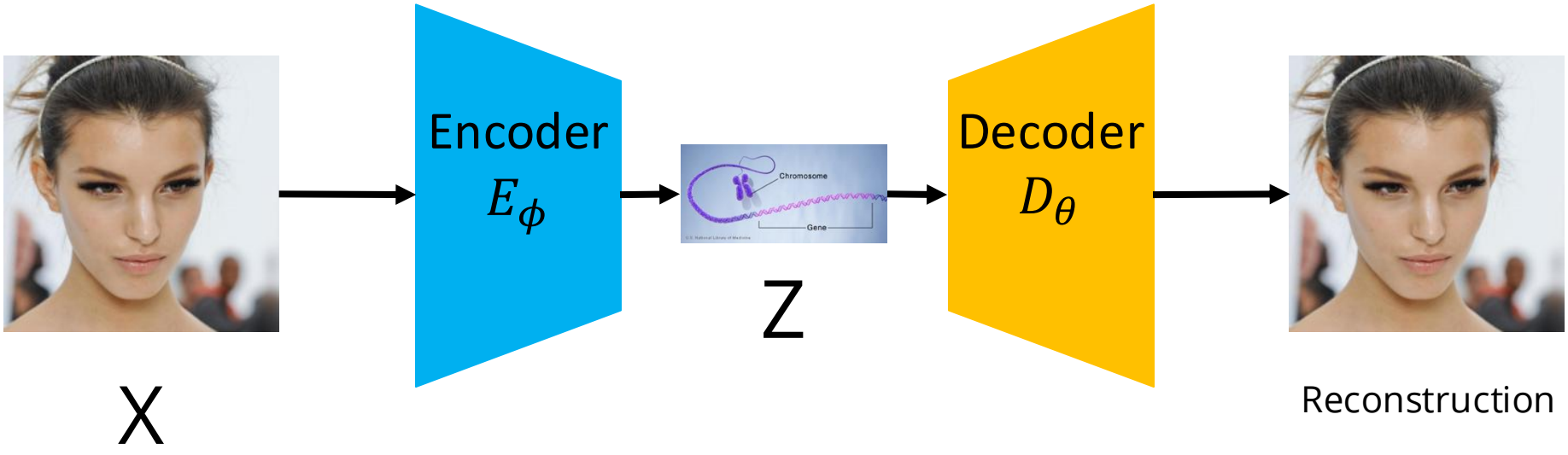
We need to do two things

- Maximize the likelihood of data X
- Make sure that the Z we get from encoding X can actually be decoded into the same X

Carnegie Mellon University

Kingma & Welling. "Auto-Encoding Variational Bayes". ICLR 2014. https://arxiv.org/pdf/1312.6114

Attempt 2: Variational autoencoder (VAE)



We need to do two things

- Maximize the likelihood of data X => maximize $\log p_\theta(x)$
- Make sure that the Z we get from encoding X can actually be decoded into the same X => minimize the "difference" between $q_\phi(z|x)$ and $p_\theta(z|x)$

Carnegie Mellon University

Kingma & Welling. "Auto-Encoding Variational Bayes". ICLR 2014. https://arxiv.org/pdf/1312.6114

How to measure the “difference” between distributions

We can’t directly use geometric distance, it doesn’t make sense for distributions!

Instead we use something called **probability divergence**

A function D need to satisfy the following two conditions in order to be a probability divergence:

1. D(p||q) >= 0
2. D(p||q) = 0 if and only if p=q

Note: divergence doesn’t need to be symmetric

How to measure the “difference” between distributions

We can’t directly use geometric distance, it doesn’t make sense for distributions!

Instead we use something called **probability divergence**

There are many different probability divergence out there

One of the most popular ones is called KL divergence

$$D_{KL}(p||q) = E_{x\sim p}[\log\frac{p(x)}{q(x)}]$$

Intuitively, it means “if the world distributes like p, how surprise we are going to be if we model it like q”

VAE's evidence lower bound (ELBO)

We are trying to

- Maximize the likelihood of data X => maximize $\log p_\theta(x)$
- Make sure that the Z we get from encoding X can actually be decoded into the same X => minimize the "difference" between $q_\phi(z|x)$ and $p_\theta(z|x)$

$$\Rightarrow \text{argmax}_{\phi,\theta} \log p_\theta(x) - D_{KL}(q_\phi(z|x)||p_\theta(z|x))$$

$$\Rightarrow \text{argmax}_{\phi,\theta} E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p_\theta(z))$$

KL regularization

Encode-Decode reconstruction loss

Evidence lower bound (ELBO)

Two ways to derive ELBO (1)

$$
\begin{aligned}
\log p_{\boldsymbol{\theta}}(\mathbf{x}) &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x})\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right] \\
&= \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\right]\right]}_{=\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})\ \text{(ELBO)}} + \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right]}_{=D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}))}
\end{aligned}
$$

Carnegie Mellon University

Two ways to derive ELBO (1)

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\right]\right]}_{=\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})\ (\text{ELBO})} + \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right]}_{=D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}))}$$

$$E_{z\sim q_\phi(z|x)}\left[\log\left(\frac{p_\theta(x,z)}{q_\phi(z|x)}\right)\right] = E_{z\sim q_\phi(z|x)}\left[\log p_\theta(x,z) - \log q_\phi(z|x)\right]$$

$$= E_{z\sim q_\phi(z|x)}\left[\log p_\theta(z) + \log p_\theta(x|z) - \log q_\phi(z|x)\right]$$

$$= E_{z\sim q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - E_{z\sim q_\phi(z|x)}\left[\log q_\phi(z|x) - \log p_\theta(z)\right]$$

$$= E_{z\sim q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - D_{KL}(q_\phi(z|x)||p_\theta(z))$$

Carnegie Mellon University

Two ways to derive ELBO (2)

$$\log p_\theta(x) = \log \int p_\theta(x,z)dz$$

$$= \log \int \frac{q_\phi(z|x)}{q_\phi(z|x)} p_\theta(x,z)dz = \log \mathrm{E}_{q_\phi(z|x)}[\frac{p_\theta(x,z)}{q_\phi(z|x)}]$$

$$\geq \mathrm{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right]$$

$$= E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] \; - D_{KL}(q_\phi(z|x)||p_\theta(z))$$

Attempt 3: Catch me if you can

Generator

Tries to make the fake samples more and more realistic so that it can fool the discriminator

Discriminator

Tries to be better and better at distinguishing fake samples from real ones

Carnegie Mellon University

Attempt 3: Generative adversarial networks (GAN)

Generator

Tries to make the fake samples more and more realistic so that it can fool the discriminator

A generator that directly transform a sample from the easy distribution to the complicated target distribution

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Discriminator

Tries to be better and better at distinguishing fake samples from real ones

A discriminator that predicts if the input sample if real or fake

Some distribution that is easy to sample from (e.g. Gaussian)

Carnegie Mellon University

Goodfellow et al. "Generative Adversarial Networks". NeurIPS 2014. https://arxiv.org/pdf/1406.2661

We will actually use all of these techniques that we learned today to understand diffusion!

Next class, we will

- Take a deeper dive into VAE and how to actually train & sample from one
- What's wrong with all of these models
- Intuition behind diffusion models and why it can be better
- How was diffusion developed from all these prior works (especially VAE) back in the day
- How people made it work by using tricks that they learned from these prior works (especially VAE)