

Machine learning and economics reading group
Diffusion models and variational autoencoders

Maximilian Kasy

Department of Economics, University of Oxford

Winter 2024

People in Oxford sitting in a room without windows, discussing AI



Generated using Stable Diffusion v2-1.

Outline

- Variational auto-encoders.
 - Self-prediction with a “bottleneck.”
 - Encoder and decoder models.
- Diffusion models.
 - Special case of hierarchical autoencoders.
 - Fix the encoder model: Just add normal noise.
 - Alternative ways of estimating the decoder model.
- Conditioning and guidance.
 - Same as before, but conditioning on prompts.
 - Can over-emphasize examples which fit a prompt.

Takeaways

- What transformers have achieved for language generation, diffusion models have achieved for image generation.
- The basic idea is simple:
 1. Add normal noise to images in a data-base.
 2. Predict the de-noised image from the noisy one.
 3. Do so in multiple rounds.
 4. Then generate images by starting with pure noise.
- Conditioning predictions on (encodings of) text labels yields image generation based on text prompts.

Variational autoencoders

Diffusion models

Conditioning and guidance

Setup

- i.i.d. observables: \mathbf{x} (e.g., images).
- Latent variables: \mathbf{z} .
- Goal: Model the distribution $p(\mathbf{x})$.
- Decoder model: $p_{\theta}(\mathbf{x}|\mathbf{z})$.
- Encoder model: $q_{\phi}(\mathbf{z}|\mathbf{x})$.
- Marginal (prior) for \mathbf{z} : $p(\mathbf{z})$.

The decoder as a generative model

- Given θ , it is easy to sample from $p(\mathbf{x})$:
 1. Obtain a draw of $\mathbf{z} \sim p(\mathbf{z})$.
 2. Then obtain a draw from $p_{\theta}(\mathbf{x}|\mathbf{z})$.

- Maximum likelihood estimation:
Given the sample of observed \mathbf{x}_i , find θ to maximize

$$\sum_i \log p_{\theta}(\mathbf{x}_i) = \sum_i \log \left(\int_{\mathbf{z}} p_{\theta}(\mathbf{x}_i|\mathbf{z}) p(\mathbf{z}) d\mathbf{z} \right).$$

- Problem: The integral is too hard to compute for interesting models (e.g., neural networks).

Decomposing the likelihood

- By definition of conditional probabilities, for arbitrary \mathbf{z} :

$$\begin{aligned}\log p_{\theta}(\mathbf{x}) &= \log \left(\frac{p_{\theta}(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p_{\theta}(\mathbf{z}|\mathbf{x})} \cdot \frac{q_{\phi}(\mathbf{z}|\mathbf{x})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right) \\ &= \log \left(\frac{p_{\theta}(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right) + \log \left(\frac{q_{\phi}(\mathbf{z}|\mathbf{x})}{p_{\theta}(\mathbf{z}|\mathbf{x})} \right).\end{aligned}$$

- Taking expectations of this over $q_{\phi}(\mathbf{z}|\mathbf{x})$, for arbitrary ϕ , gives:

$$\log p_{\theta}(\mathbf{x}) = \underbrace{E_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} \left[\log \left(\frac{p_{\theta}(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right) \right]}_{L(\phi, \theta; \mathbf{x}) \quad \text{(Evidence lower bound)}} + \underbrace{E_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} \left[\log \left(\frac{q_{\phi}(\mathbf{z}|\mathbf{x})}{p_{\theta}(\mathbf{z}|\mathbf{x})} \right) \right]}_{D_{\text{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x}) || p_{\theta}(\mathbf{z}|\mathbf{x})) \quad \text{(KL divergence)}}$$

Estimating the model by maximizing the ELBO

- Rearranging the likelihood decomposition:

$$L(\phi, \theta; \mathbf{x}) = \log p_{\theta}(\mathbf{x}) - D_{KL}(q_{\phi}(\mathbf{z}|\mathbf{x})||p_{\theta}(\mathbf{z}|\mathbf{x})).$$

- Maximizing the ELBO $L(\phi, \theta; \mathbf{x})$ wrt θ and ϕ is equivalent to simultaneously
 1. Maximizing $\log p_{\theta}(\mathbf{x})$.
 2. Minimizing $D_{KL}(q_{\phi}(\mathbf{z}|\mathbf{x})||p_{\theta}(\mathbf{z}|\mathbf{x}))$.

How to maximize the ELBO

- We can decompose the ELBO further:

$$\begin{aligned} L(\phi, \theta; \mathbf{x}) &= E_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \left(\frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right) \right] \\ &= \underbrace{E_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{(Reconstruction term)}} - \underbrace{E_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \left(\frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})} \right) \right]}_{D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \quad \text{(Prior matching term)}}. \end{aligned}$$

- The expectations can easily be approximated using simulation.
- Suppose $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\Sigma}_\phi(\mathbf{x}))$.
- A differentiable estimate of the expectations averages over draws of

$$\mathbf{z}_j = \boldsymbol{\mu}_\phi(\mathbf{x}) + \boldsymbol{\Sigma}_\phi(\mathbf{x})^{1/2} \cdot \boldsymbol{\varepsilon}_j,$$

for fixed draws $\boldsymbol{\varepsilon}_j \sim \mathcal{N}(\mathbf{0}, I)$.

Variational autoencoders

Diffusion models

Conditioning and guidance

Hierarchical autoencoders

- Straightforward generalization: Denote $\mathbf{x}^0 = \mathbf{x}$,
Hierarchy of multiple latent variables $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^T$.
- Encoder and decoder models for each layer:

$$q_\phi(\mathbf{x}^t | \mathbf{x}^{t-1})$$

$$p_\theta(\mathbf{x}^t | \mathbf{x}^{t+1}).$$

- ELBO for this hierarchical model:

$$L(\phi, \theta; \mathbf{x}) = E_{\mathbf{x}^{1:T} \sim q_\phi(\mathbf{x}^{1:T} | \mathbf{x}^0)} \left[\log \left(\frac{p_\theta(\mathbf{x}^{0:T})}{q_\phi(\mathbf{x}^{1:T} | \mathbf{x})} \right) \right]$$

Diffusion models

- Simplification: q_ϕ is a *known* distribution q .

- In particular:

$$x^t | x^{t-1} \sim N(\sqrt{\alpha_t} \cdot x^{t-1}, (1 - \alpha_t) \cdot I).$$

- For $\bar{\alpha}_T = \prod_{t=1}^T \alpha_t \approx 0$, we get

$$x^T | x^0 \sim N(\sqrt{\bar{\alpha}_T} \cdot x^0, (1 - \bar{\alpha}_T) \cdot I) \approx N(0, I).$$

- Furthermore

$$x^{t-1} | x^0, x^t \sim N(\mathbf{a}^t \cdot x^0 + \mathbf{b}^t \cdot x^t, \mathbf{c}^t \cdot I),$$

for constants $\mathbf{a}^t, \mathbf{b}^t, \mathbf{c}^t$ that are easy to calculate.

Estimating diffusion models

- Leading terms in ELBO for diffusion models are of the form

$$E_{x^t \sim q(x^t|x^0)} \left[D_{KL} \left(q(x^{t-1}|x^0, x^t) \parallel p_\theta(x^{t-1}|x^t) \right) \right]$$

- Recall $q(x^{t-1}|x^0, x^t)$ is a normal distribution.
- For such normal distributions with known variance, minimizing D_{KL} is equivalent to predicting the mean

$$E[x^{t-1}|x^0, x^t] = a^t \cdot x^0 + b^t \cdot x^t,$$

based on x^t .

Three equivalent prediction targets

- Goal: predict $E[x^{t-1}|x^0, x^t] = a^t \cdot x^0 + b^t \cdot x^t$, based on x^t .
- Three equivalent approaches:
 1. Predict x^0 based on x^t
Plug into $a^t \cdot x^0 + b^t \cdot x^t$.
 2. Predict ε_t based on x^t ,
where $x^t = \sqrt{\bar{\alpha}_t} \cdot x^0 + \sqrt{1 - \bar{\alpha}_t} \cdot \varepsilon_t$.
 3. Predict $\nabla \log p(x^t)$ based on x^t .
Recall Tweedie's formula:

$$E[x^0|x^t] = x^t + (1 - \bar{\alpha}_t) \cdot \nabla \log p(x^t).$$

- All three prediction targets can be predicted using neural networks.
- Approach 3 leads to an interpretation of denoising as gradient flow.

Variational autoencoders

Diffusion models

Conditioning and guidance

Conditioning

- Typically, in generative AI, the goal is not to learn $p(\mathbf{x})$, but instead $p(\mathbf{x}|\mathbf{y})$.
- Leading example: \mathbf{y} is a text prompt, or LLM encoding thereof.
- Immediate extension of our previous approach:
Learn conditional predictions of \mathbf{x}^{t-1} given \mathbf{x}^t and \mathbf{y} .
- Works, but leads to generated \mathbf{x} that might not be “clear-cut” representations of \mathbf{y} .

Classifier guidance

- By Bayes' rule,

$$\nabla \log p(\mathbf{x}^t | \mathbf{y}) = \nabla \log \left(\frac{p(\mathbf{x}^t) \cdot p(\mathbf{y} | \mathbf{x}^t)}{p(\mathbf{y})} \right) = \nabla \log p(\mathbf{x}^t) + \nabla \log p(\mathbf{y} | \mathbf{x}^t).$$

- Can learn the score of the conditional model by learning the score of the unconditional model, and a classifier.
- To generate more clear-cut examples, overweight the classifier in gradient flow:

$$\nabla \log p(\mathbf{x}^t) + \gamma \cdot \nabla \log p(\mathbf{y} | \mathbf{x}^t)$$

for $\gamma \geq 1$.

References

- https://en.wikipedia.org/wiki/Evidence_lower_bound
- *Luo, C. (2022). Understanding diffusion models: A unified perspective. arXiv preprint arXiv:2208.11970*
- *Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., and Yang, M.-H. (2023). Diffusion models: A comprehensive survey of methods and applications. ACM Computing Surveys, 56(4):1–39*