Foundations of machine learning

Diffusion models and variational autoencoders

Maximilian Kasy

Department of Economics, University of Oxford

Winter 2025

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 for this part of class

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

The decoder as a generative model

- Given θ, it is easy to sample from $p(x)$:
  1. Obtain a draw of $z \sim p(z)$.
  2. Then obtain a draw from $p_\theta(x|z)$.
- Maximum likelihood estimation:
  Given the sample of observed x_i, find θ to maximize

$$\sum_i \log p_\theta(x_i) = \sum_i \log \left(\int_z p_\theta(x_i|z) p(z) dz \right).$$

- 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 z:

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

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

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

Estimating the model by maximizing the ELBO

- Rearranging the likelihood decomposition:

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

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

How to maximize the ELBO

- We can decompose the ELBO further:

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

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

- The expectations can easily be approximated using simulation.
- Suppose $q_\phi(z|x) = N(\mu_\phi(x), \Sigma_\phi(x))$.
- A differentiable estimate of the expectations averages over draws of

$$z_j = \mu_\phi(x) + \Sigma_\phi(x)^{1/2}\cdot\varepsilon_j,$$

for fixed draws $\varepsilon_j \sim N(0,I)$.

Variational autoencoders

Diffusion models

Conditioning and guidance

Hierarchical autoencoders

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

$$q_\phi(x^t | x^{t-1}) \qquad\qquad p_\theta(x^t | x^{t+1}).$$

- ELBO for this hierarchical model:

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

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
$$x^T | x^0 \sim N(\sqrt{\bar{\alpha}_T} \cdot x^0, (1-\bar{\alpha}_T) \cdot I) \cdot \approx N(0, I).$$
- Furthermore
$$x^{t-1} | x^0, x^t \sim N(a^t \cdot x^0 + b^t \cdot x^t, c^t \cdot I),$$
for constants a^t, b^t, c^t that are easy to calculate.

Estimating diffusion models

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

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

- 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.

Conditioning

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

Classifier guidance

- By Bayes' rule,

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

- 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:

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

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*