Deep Generative Models

Chapter 5: Variational Inference and VAEs

Ali Bereyhi

ali.bereyhi@utoronto.ca

Department of Electrical and Computer Engineering

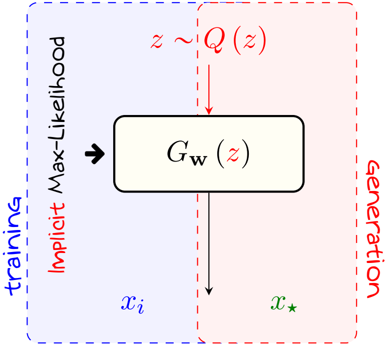
University of Toronto

Summer 2025

Modern Data Generation

Recall that we are looking into modern generative models

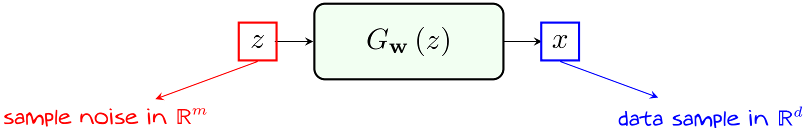
modern generative models mainly learn how to sample!



Recall: *Generic Model for Generator*

Generator Model

Generator model $G_{\mathbf{w}} : \mathbb{R}^m \mapsto \mathbb{R}^d$ is a mapping that maps a latent sample $z \sim P(z)$, typically Gaussian noise, into a data sample

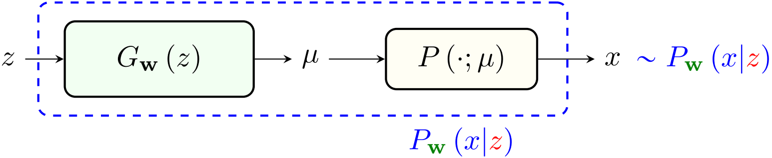


GANs consider deterministic generators

+ *Can we also consider a probabilistic generator?!*
– Sure! Why not?!
+ *But how does it look like?*
– Let's build one together!

Modeling Probabilistic Generator

Consider following setting: *a model $G_{\mathbf{w}}$ computes some statistics from* latent



We use the statistics to specify a distribution $P(x;\mu)$ and sample from it

↳ *Sample z is drawn as*

$$x \sim P(x;\mu) = P(x; G_{\mathbf{w}}(z))$$

↳ *The distribution depends on model parameters $\mathbf{w}$ and is conditioned on z*

$$x \sim P(x;\mu) \equiv P_{\mathbf{w}}(x|z)$$

Probabilistic Generator: *Gaussian Example I*

Let's look at an example: *say* $G_{\mathbf{w}} : \mathbb{R}^m \mapsto \mathbb{R}^d$

- *Sample a latent z and compute a mean value*

$$\mu = G_{\mathbf{w}}(z)$$

- *Set μ to be the mean of a Gaussian distribution and sample it*
  ↳ *with identity covariance it's very easy to sample this distribution*

$$x \sim P(x; \mu) \equiv \mathcal{N}(\mu, 1) = \mathcal{N}(G_{\mathbf{w}}(z), 1)$$

We can think of this process as sampling x conditioned to z from

$$P_{\mathbf{w}}(x|z) = \frac{1}{(2\pi)^{d/2}} \exp\left\{-\frac{\|x - G_{\mathbf{w}}(z)\|^2}{2}\right\}$$

Probabilistic Generator: *Gaussian Example II*

We can further extend this example: *say* $G_{\mathbf{w}} : \mathbb{R}^m \mapsto \mathbb{R}^d \times \mathbb{R}^{d \times d}$

- *Sample a latent z and compute mean and covariance*

$$\mu, \Sigma = G_{\mathbf{w}}(z)$$

- *Sample a Gaussian distribution with mean μ and covariance Σ*
  - ↳ *with general covariance sampling might need more computation*

$$x \sim P(x; \mu, \Sigma) \equiv \mathcal{N}(\mu, \Sigma) = \mathcal{N}(G_{\mathbf{w}}(z))$$

We can think of this process as sampling from

$$P_{\mathbf{w}}(x|z) = \frac{1}{(2\pi|\Sigma_{\mathbf{w}}(z)|)^{d/2}} \exp\left\{ -\frac{(x - \mu_{\mathbf{w}}(z))^{\mathsf{T}} \Sigma_{\mathbf{w}}^{-1}(z)(x - \mu_{\mathbf{w}}(z))}{2} \right\}$$

Probabilistic Generation: *General Form*

We can formulate a general *latent-space probabilistic generation as*

Probabilistic Latent-Space Generation

To build a probabilistic latent-space generator

1. *Sample latent space as* $z \sim P(z)$
   ↳ $P(z)$ *is an arbitrary distribution*
2. *Compute statistics as* $\mu = G_{\mathbf{w}}(z)$
   ↳ $G_{\mathbf{w}}$ *is a computational model, e.g., NN*
3. *Sample from* $P(x; \mu) \equiv P_{\mathbf{w}}(x|z)$
   ↳ $P(x; \mu)$ *is an arbitrary distribution parameterized by* μ

- *Is such model computationally tractable?*
- *Let's check!*

Probabilistic Generation: *Practical Setting*

Let's take a look at each step

1. *Sample latent space as* $z \sim P(z)$
   - ↳ $P(z)$ *is an arbitrary distribution*
   - ✔ *We set* $P(z)$ *simply to* $\mathcal{N}(0,1)$
2. *Compute statistics as* $\mu = G_{\mathbf{w}}(z)$
   - ✔ *This is a simple forward pass*
3. *Sample from* $P(x;\mu) \equiv P_{\mathbf{w}}(x|z)$
   - ↳ $P(x;\mu)$ *is an arbitrary distribution parameterized by* μ
   - ✔ *We set* $P(x;\mu)$ *simply to* $\mathcal{N}(\mu,1)$

Probabilistic Generation in Practice

In practice we work with easy-to-simple distributions $\equiv$ *Gaussian*

- *Wait a moment! Isn't output a Gaussian sample?! Isn't this too limited?!*
- *Careful! The output is only conditionally Gaussian!*

Model Distribution: *Example*

Consider a scalar example: *say* $z \sim \text{Unif}\,[0,1]$ *and* $G_\theta : [0,1] \mapsto \{0,1\}$ *is*

$$\mu_\theta(z) = G_\theta(z) = \begin{cases} 0 & z < \theta \\ 1 & z \geqslant \theta \end{cases}$$

We set $P(x;\mu)$ *to* $\mathcal{N}(\mu,1)$*: we thus have*

$$P_\theta(x|z) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu_\theta(z))^2}{2}\right\}$$

This is though the distribution of x *for a given* z*!*

Distribution of x *is determined by marginalization over latent*

$$P_\theta(x) = \int P_\theta(x|z)\, P(z) \mathrm{d}z$$

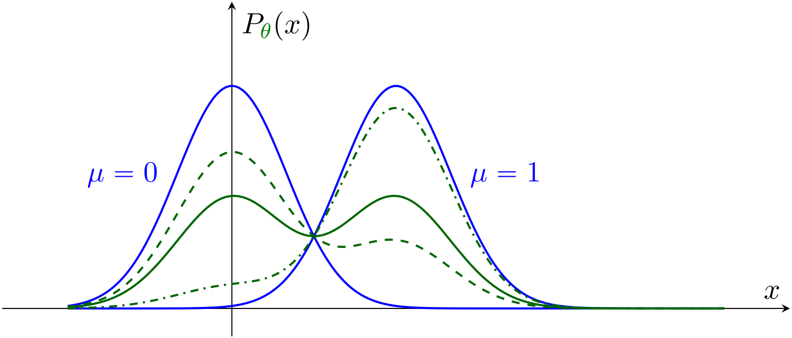
Model Distribution: *Example*

Let's use the marginalization to compute $P_\theta(x) \equiv$ model distribution

$$\begin{aligned}
P_\theta(x) &= \int_0^1 P_\theta(x|z)\, P(z) \mathrm{d}z \\
&= \int_0^1 \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{(x-\mu_\theta(z))^2}{2} \right\} \cdot 1 \mathrm{d}z \\
&= \int_0^\theta \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{(x-0)^2}{2} \right\} \mathrm{d}z + \int_\theta^1 \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{(x-1)^2}{2} \right\} \mathrm{d}z \\
&= \frac{\theta}{\sqrt{2\pi}} \exp\left\{ -\frac{x^2}{2} \right\} + \frac{1-\theta}{\sqrt{2\pi}} \exp\left\{ -\frac{(x-1)^2}{2} \right\}
\end{aligned}$$

This is a mixture of Gaussian distributions

Model Distribution: *Example*



Say $\theta = 0.5$: the model distribution is average of the two Gaussians

↳ *Setting $\theta = 0.7$ gives more weight to zero-mean Gaussian*

↳ *Setting $\theta = 0.1$ gives more weight to unit-mean Gaussian*

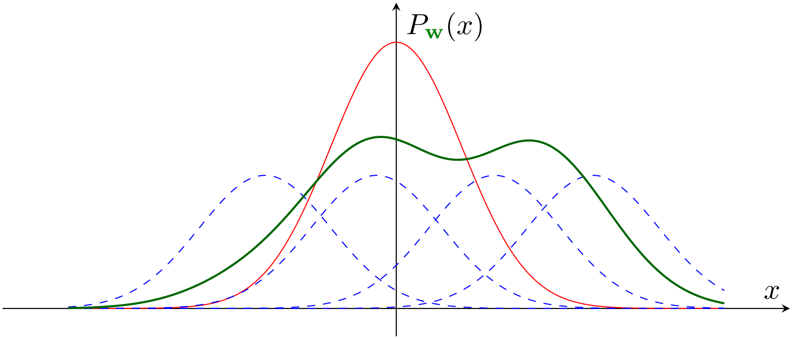
We can sketch a class of distributions by changing θ

Model Distribution: *Mixture of Gaussians*

In practice, *the model distribution is a mixture of infinite Gaussians*

$$P_{\mathbf{w}}(x) = \int P_{\mathbf{w}}(x|z)\, P(z)\, \mathrm{d}z$$

Here, $P(z)$ is Gaussian itself $\rightsquigarrow$ we can make a large set of dustributions



Universal Model Distribution

Universality of Gaussian Mixtures *(informal)*

Consider the distribution model on $\mathbb{X}$

$$P_{\mathbf{w}}(x) = \int P_{\mathbf{w}}(x|z)\, P(z)\, \mathrm{d}z$$

whose conditional distributions $P_{\mathbf{w}}(x|z)$ *are learnable* Gaussians: *this model describes a dense set of distributions on* $\mathbb{X}$ $\rightsquigarrow$ *for almost any smooth* $Q(x)$ *there exists a* $\mathbf{w}$ *that*

$$P_{\mathbf{w}}(x) \approx Q(x)$$

! Attention

We could not do this without conditional modeling $P_{\mathbf{w}}(x|z)$

Training via MLE

- *Sounds good! How can we then train these probabilistic models?!*
- Well, we need to use MLE

The likelihood of the model for a given sample x is

$$
\begin{aligned}
P_{\mathbf{w}}(x) &= \int P_{\mathbf{w}}(x|z)\, P(z)\, \mathrm{d}z \\
&= \boxed{C} \int \exp\left\{-\frac{\|x - G_{\mathbf{w}}(z)\|^2}{2}\right\} \exp\left\{-\frac{\|z\|^2}{2}\right\} \mathrm{d}z \\
&= C \int \exp\left\{-\frac{\|x - G_{\mathbf{w}}(z)\|^2 + \|z\|^2}{2}\right\} \mathrm{d}z
\end{aligned}
$$

some constant

$G_{\mathbf{w}}(z)$ *is a complex model, e.g., NN $\rightsquigarrow$ this is not a tractable task!*

MLE Training: *Formulating Challenge*

Moral of Story

Using a probabilistic generator, we can approximate data distribution very well

↳ *It's however not tractable to train the model by explicit MLE!*

Let's formulate the training challenge: *we have access to*

- *latent distribution $P(z)$, and*
- *conditional generative model $P(x|z)$*

We intend to compute the marginal generative model which gives us likelihood

$$P(x) = \int P(x|z) P(z) \, \mathrm{d}z$$

We can overcome this challenge using variational inference framework