Training Deep Generative Models: Variations on a Theme

Philip Bachman and Doina Precup
McGill University, School of Computer Science

McGill

Neural Information Processing Systems Foundation

Abstract

Recent techniques for training deep generative models are based on coaxing pairs of sample generating systems into agreement. Methods such as stochastic variational inference (as used in variational auto-encoders), denoising (as used in denoising auto-encoders), and contrastive divergence (as used to train Restricted Boltzmann Machines) all fit nicely under this interpretation. We formally develop this point of view, which provides a unified framework in which to compare and contrast many approaches to training deep generative models. We hope our effort might help other researchers compress their understanding of methods in this domain, and thus avoid getting overwhelmed as they continue to proliferate.

The Main Idea

- We step back slightly, and extend the standard variational free energy bound to a KL divergence between distributions $q(x,\tau)$ and $p(x,\tau)$.
  - x denotes an "observable" variable.
  - τ denotes one or more latent variables z_i, i.e. $\tau \equiv \{z_0, ..., z_n\}$.
- For this, we incorporate the distribution $\mathcal{D}$ over $x \in \mathcal{X}$ into the *inference* model $q(\tau|x)$.
  - This produces $q(x,\tau) \equiv \mathcal{D}(x)q(\tau|x)$.
- The *generation* model is given by $p(x,\tau) \equiv p(x|\tau)p(\tau)$.
  - This is the same as in the "standard" setting.
- By considering a bound on the joint data/inference/generation system, we can more easily assimilate diverse techniques for training deep generative models into a shared framework.
  - This contrasts with the more typical view, which considers bounding $\log p(x)$ separately for each $x \in \mathcal{X}$.

Using $\mathrm{KL}(q\,||\,p)$ to Bound $\mathbb{E}_{q(x)} \log p(x)$

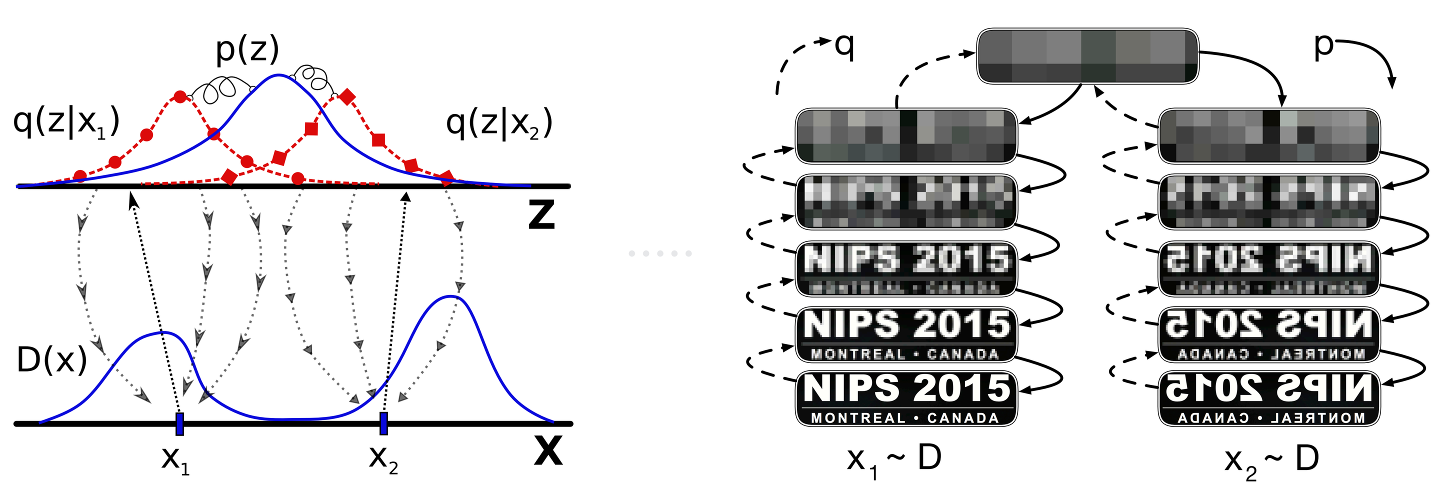
To begin, assume distributions $q(x, z_0..., z_n)$ and $p(x, z_0, ..., z_n)$ over variables $\{x, z_0, ..., x_n\}$. For these distributions, we can say:

$$\mathrm{KL}(q(x, z_0, ..., x_n)\,||\,p(x, z_0, ..., x_n)) = \mathop{\mathbb{E}}_{q(x,z_0,...,z_n)} \left[\log \frac{q(x, z_0, ..., z_n)}{p(x, z_0, ..., x_n)}\right]$$
$$= \mathop{\mathbb{E}}_{q(x)} \left[\log \frac{q(x)}{p(x)} + \mathop{\mathbb{E}}_{q(z_0,...,z_n|x)} \left[\log \frac{q(z_0, ..., z_n|x)}{p(z_0, ..., z_n|x)}\right]\right]$$
$$= \mathrm{KL}(q(x)\,||\,p(x)) + \mathop{\mathbb{E}}_{q(x)} \left[\mathrm{KL}(q(z_0, ..., z_n|x)\,||\,p(z_0, ..., z_n|x))\right]$$
$$\geq \mathrm{KL}(q(x)\,||\,p(x)), \quad \text{i.e.} \mathop{\mathbb{E}}_{q(x)} [\log q(x) - \log p(x)]$$

So, if we can compute $\mathrm{KL}(q\,||\,p) - \mathbb{E}_{q(x)} [\log q(x)]$, we can compute:

$$\mathrm{KL}(q(x, z_0, ..., z_n)\,||\,p(x, z_0, ..., z_n)) - \mathop{\mathbb{E}}_{q(x)} [\log q(x)] \geq \mathop{\mathbb{E}}_{q(x)} [-\log p(x)]$$

VAEs and Deep Non-Eq Thermodynamics



(a) (b)

Left: The standard variational autoencoder – an inference model $q(z|x)$ is used to approximate $p(z|x)$ to train the generator $p(x,z) = p(x|z)p(z)$.
Right: The method of Sohl-Dickstein et al. (ICML 2015) – a fixed reverse process q goes from $q(x_0) \equiv \mathcal{D}(x_0)$ to a prior $q(x_T)$ via diffusion steps.
Both: These methods train p to match q, by minimizing $\mathrm{KL}(q\,||\,p)$.

Denoising Autoencoders

TLDR: Define a prior $p(z) = \mathbb{E}_{\mathcal{D}(x)}\, q(z|x)$ by "convolving" the (noisy) encoder $q(z|x)$ with the data distribution $\mathcal{D}$. Then, use SGVB to train the directed generative model $p(x) = \mathbb{E}_{p(z)}\, p(x|z)$.
Details: Basic DAE training minimizes $\mathrm{KL}(q\,||\,p)$ by gradient descent on:

$$\nabla_q\, \mathrm{KL}(q(x,z)\,||\,p(x,z)) = \;...$$
$$= \mathop{\mathbb{E}}_{q(x,z)} \left[\nabla_q \log \frac{q(z|x)}{p(x|z)} - \nabla_q \log \left(\mathop{\mathbb{E}}_{q(\hat{x})} [q(z|\hat{x})]\right)\right]$$
$$= \mathop{\mathbb{E}}_{q(x,z)} \left[\nabla_q \log \frac{q(z|x)}{p(x|z)} - \mathop{\mathbb{E}}_{q(\hat{x}|z)} \left[\nabla_q \log q(z|\hat{x}) + \nabla_q \log q(\hat{x})\right]\right]$$
$$= \mathop{\mathbb{E}}_{q(z)} \left[\mathop{\mathbb{E}}_{q(x|z)} [-\nabla_q \log p(x|z)]\right] \qquad (\nabla_p \text{ is easy to get from this})$$

- DAEs minimize $KL(q\,||\,p)$ just by minimizing reconstruction error.

Importance-Weighted Autoencoders

With $q(x,z) \equiv \mathcal{D}(x)q(z|x)$ and $p(x,z) \equiv p(x|z)p(z)$, we define $q_p^k(z|x)$ by sampling $\{z_1, ..., z_k\}$ from $q(z|x)$, then resampling $\{z_1, ..., z_k\}$ using the NIS weights $\{w_1, ..., w_k\}$ for sampling from $p(z|x)$ via $q(z|x)$. I.e.:

$$w_i = \frac{\frac{p(z_i|x)}{q(z_i|x)}}{\sum_{j=1}^k \frac{p(z_j|x)}{q(z_j|x)}} = \frac{\frac{p(x|z_i)p(z_i)}{q(z_i|x)}}{\sum_{j=1}^k \frac{p(x|z_j)p(z_j)}{q(z_j|x)}}$$

This permits a variational bound on $\log p(x)$, using samples from $q_p^k(z|x)$:

$$\log p(x) \geq \mathop{\mathbb{E}}_{(z_i,w_i)\sim q_p^k(z_i|x)} \left[\log \frac{p(x|z_i)p(z_i)}{w_i q(z_i|x)}\right]$$

Sample k zs at a time and marginalize over resampling from $\{z_1, ..., z_k\}$:

$$\log p(x) \geq \mathop{\mathbb{E}}_{\{z_1,...,z_k\}\sim q(z|x)} \left[\sum_{i=1}^k w_i \log \frac{p(x|z_i)p(z_i)}{w_i q(z_i|x)}\right] \qquad (1)$$

For each x_i/w_i, the log-ratio in Eq. 1 simplifies to (see paper for algebra):

$$\log \frac{p(x|z_i)p(z_i)}{w_i q(z_i|x)} = \log \left(\sum_{j=1}^k \frac{p(x|z_j)p(z_j)}{q(z_j|x)}\right) \qquad (2)$$

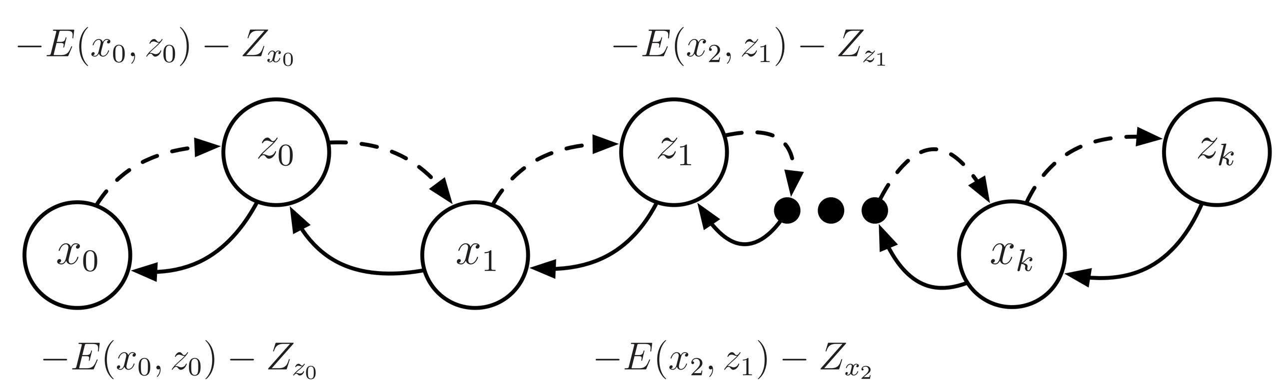
Using Eq. 2 and knowing $\sum_i w_i = 1$, we can rewrite the bound in Eq. 1:

$$\log p(x) \geq \mathop{\mathbb{E}}_{\{z_1,...,z_k\}\sim q(z|x)} \left[\log \left(\sum_{z_i} \frac{p(x|z_i)p(z_i)}{q(z_i|x)}\right)\right]$$

This variational bound based on the "meta distribution" q_p^k reproduces the bounds for Reweighted Wake-Sleep and Importance-Weighted Autoencoders (see paper for refs). Using an NIS correction towards $p(z|x)$, RWS and IWAEs put a tighter bound on $\log p(x)$ than $q(z|x)$ provides on its own.

Contrastive Divergence for RBMs

TLDR: Define a prior $p(z_k) = \mathbb{E}_{q(x_k,z_{k-1},...,z_0,x_0)}\, q(z_k|x_k, z_{k-1}, ..., z_0, x_0)$ by "convolving" q with $\mathcal{D}$. Then, use policy gradient to train the directed generative model $p(x) = \mathbb{E}_{p(z_k,x_k,...,x_1,z_0)}\, p(x|z)$. Use tied weights in q/p.



$$\log p(x|z) = -E(x,z) - Z_z$$
$$\log p(z|x) = -E(x,z) - Z_x$$
$$\log \frac{q(x_0, z_0, ..., x_k, z_k)}{p(x_0, z_0, ..., x_k, z_k)} = Z_{z_k} - Z_{x_0}$$