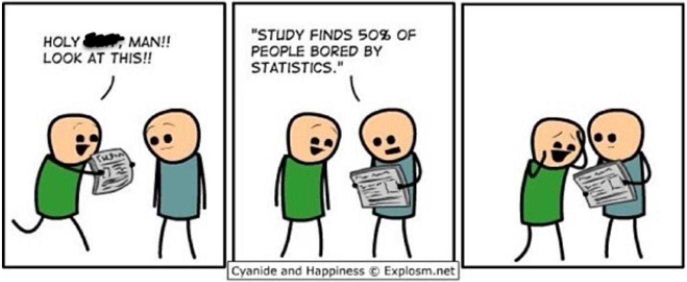
Tutorial on Variational Autoencoder and its Gradient Estimators

Raymond A. Yeh

University of Illinois at Urbana-Champaign

February 21, 2019

Motivation


HOLY [redacted] MAN!!
LOOK AT THIS!!
"STUDY FINDS 50% OF
PEOPLE BORED BY
STATISTICS."
Cyanide and Happiness © Explosm.net

Motivation

- Suppose we are interested in modeling the distribution of

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z}) \tag{1}$$

  where only $\mathbf{x}$ is observed and $\mathbf{z}$ is an unobserved variable.

Motivation

- Suppose we are interested in modeling the distribution of

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z}) \tag{1}$$

  where only $\mathbf{x}$ is observed and $\mathbf{z}$ is an unobserved variable.
- To apply maximum-likelihood,

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}, \mathbf{z})d\mathbf{z} \tag{2}$$

  - Is this integral tractable?
  - Can we approximate it? $p_\theta(\mathbf{x}) \approx \sum_{\mathbf{z}^{(i)}} p_\theta(\mathbf{x}|\mathbf{z}^{(i)})$,

    where $\mathbf{z}^{(i)} \sim p(\mathbf{z})$.

Variational inference

- Sampling problem $\rightarrow$ optimization problem.
- Evidence Lower Bound (ELBO)

$$
\begin{aligned}
\log p_\theta(\mathbf{x}) &= \int q_\phi(\mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x})\, d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left(p_\theta(\mathbf{x}) \frac{p_\theta(\mathbf{z}|\mathbf{x})\, q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})\, q_\phi(\mathbf{z}|\mathbf{x})} \right) d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left(\frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right) d\mathbf{z} - \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left(\frac{p_\theta(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} \right) d\mathbf{z} \\
&= \mathbb{E}_{q_\phi}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] + KL(q_\phi || p_\theta) \\
&\geq \mathbb{E}_{q_\phi}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] = \mathcal{L}(\phi, \theta)
\end{aligned}
$$

Evidence lower bound (ELBO)

- When is ELBO tight? $\mathbb{E}_{q_\phi}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] + KL(q_\phi||p_\theta) \geq \mathcal{L}(\phi, \theta)$
  - To get the tightest bound, find q_ϕ such that maximizes ELBO.

Evidence lower bound (ELBO)

- When is ELBO tight? $\mathbb{E}_{q_\phi}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] + KL(q_\phi||p_\theta) \geq \mathcal{L}(\phi, \theta)$
  - To get the tightest bound, find q_ϕ such that maximizes ELBO.
- Further decompose:
  $\mathbb{E}_{q_\phi}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] = \mathbb{E}_{q_\phi}[\log(p_\theta(\mathbf{x}|\mathbf{z}))] - KL(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$
  - Estimate the first term using Monte Carlo samples.
  - KL can be computed analytically, if q and p are "simple".

Evidence lower bound (ELBO)

- When is ELBO tight? $\mathbb{E}_{q_\phi}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] + KL(q_\phi||p_\theta) \geq \mathcal{L}(\phi,\theta)$
  - To get the tightest bound, find q_ϕ such that maximizes ELBO.
- Further decompose:
  $\mathbb{E}_{q_\phi}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] = \mathbb{E}_{q_\phi}[\log(p_\theta(\mathbf{x}|\mathbf{z}))] - KL(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$
  - Estimate the first term using Monte Carlo samples.
  - KL can be computed analytically, if q and p are "simple".
- **Side Note:** EM algorithm is choosing $q_\phi(\mathbf{z}|\mathbf{x})$ as $p_{\theta^{t-1}}(\mathbf{z}|\mathbf{x})$, *i.e.* assumes the computation of the posterior is tractable.

Evidence lower bound (ELBO)

- When is ELBO tight? $\mathbb{E}_{q_\phi}\Big[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\Big] + KL(q_\phi || p_\theta) \geq \mathcal{L}(\phi, \theta)$
  - To get the tightest bound, find q_ϕ such that maximizes ELBO.
- Further decompose:
  $\mathbb{E}_{q_\phi}\Big[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\Big] = \mathbb{E}_{q_\phi}[\log(p_\theta(\mathbf{x}|\mathbf{z}))] - KL(q_\phi(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}))$
  - Estimate the first term using Monte Carlo samples.
  - KL can be computed analytically, if q and p are "simple".
- **Side Note:** EM algorithm is choosing $q_\phi(\mathbf{z}|\mathbf{x})$ as $p_{\theta^{t-1}}(\mathbf{z}|\mathbf{x})$, *i.e.* assumes the computation of the posterior is tractable.
- Need to choose a "flexible" $q_\phi(\mathbf{z}|\mathbf{x})$ that is also easy to sample from. How? Deep nets!

Variational AutoEncoder (VAE)

- Variational AutoEncoder models both $p_\theta(\mathbf{x}|\mathbf{z})$ and $q_\phi(\mathbf{z}|\mathbf{x})$ with deep networks:
  - Encoder: $q_\phi(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}(\mu_\phi(\mathbf{x}), \sigma_\phi(\mathbf{x}) \cdot \mathbf{I})$
  - Decoder: $p_\theta(\mathbf{x}|\mathbf{z}) \sim \mathcal{N}(\mu_\theta(\mathbf{z}), c \cdot \mathbf{I})$

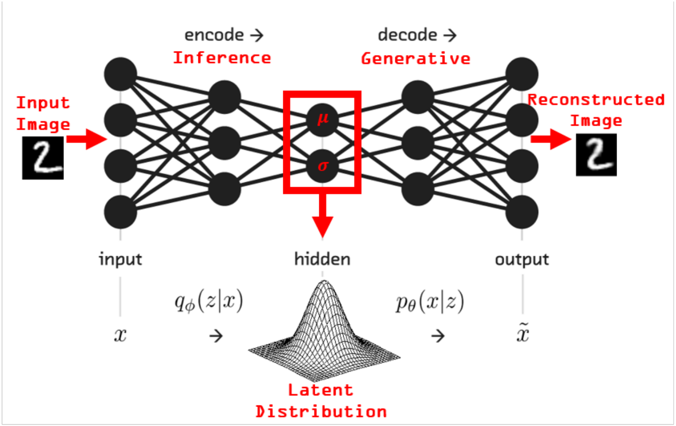
Variational AutoEncoder (VAE)

- Variational AutoEncoder models both $p_\theta(\mathbf{x}|\mathbf{z})$ and $q_\phi(\mathbf{z}|\mathbf{x})$ with deep networks:
  - Encoder: $q_\phi(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}(\mu_\phi(\mathbf{x}), \sigma_\phi(\mathbf{x}) \cdot \mathbf{I})$
  - Decoder: $p_\theta(\mathbf{x}|\mathbf{z}) \sim \mathcal{N}(\mu_\theta(\mathbf{z}), c \cdot \mathbf{I})$
- How to learn ϕ?:
  - Reparameterization Trick:
    $\mathbf{z} \sim \mathcal{N}(\mu, \sigma)$ is equivalent to $\mu + \sigma \cdot \epsilon$ where $\epsilon \sim \mathcal{N}(0, 1)$.
  - Sample $\mathbf{z}$ from q is a deterministic function of ϵ.
  - Use standard backpropgation for training

Variational AutoEncoder (VAE)

- Variational AutoEncoder models both $p_\theta(\mathbf{x}|\mathbf{z})$ and $q_\phi(\mathbf{z}|\mathbf{x})$ with deep networks:
  - Encoder: $q_\phi(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}(\mu_\phi(\mathbf{x}), \sigma_\phi(\mathbf{x}) \cdot \mathbf{I})$
  - Decoder: $p_\theta(\mathbf{x}|\mathbf{z}) \sim \mathcal{N}(\mu_\theta(\mathbf{z}), c \cdot \mathbf{I})$
- How to learn ϕ?:
  - Reparameterization Trick:
    $\mathbf{z} \sim \mathcal{N}(\mu, \sigma)$ is equivalent to $\mu + \sigma \cdot \epsilon$ where $\epsilon \sim \mathcal{N}(0, 1)$.
  - Sample $\mathbf{z}$ from q is a deterministic function of ϵ.
  - Use standard backpropgation for training
- How to learn θ? Standard backpropgation.

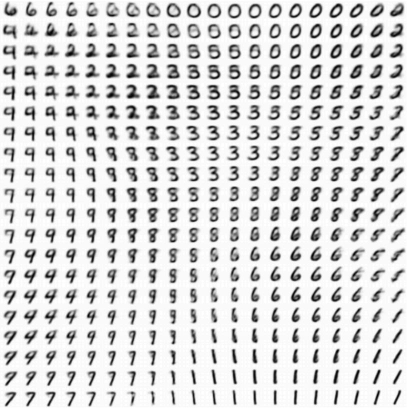
Overall pipeline



Credit: Visualizing MNIST using a Variational Autoencoder

Applications

(a) Learned Frey Face manifold



(b) Learned MNIST manifold

Credit: Kingma *et al.*, 2013

Since the original VAE paper...

- Extension to the family of $q_\phi(z)$
  - `Variational inference with normalizing flows`

Since the original VAE paper...

- Extension to the family of $q_\phi(z)$
  - `- Variational inference with normalizing flows`
- Reparametrization trick with discrete latent variable
  - `- Categorical Reparameterization with Gumbel-Softmax`

Since the original VAE paper...

- Extension to the family of $q_\phi(z)$
  - Variational inference with normalizing flows
- Reparametrization trick with discrete latent variable
  - Categorical Reparameterization with Gumbel-Softmax
- Tighter Variational Bounds
  - Importance Weighted Autoencoders

Since the original VAE paper...

- Extension to the family of $q_\phi(z)$
  `- Variational inference with normalizing flows`
- Reparametrization trick with discrete latent variable
  `- Categorical Reparameterization with Gumbel-Softmax`
- Tighter Variational Bounds
  `- Importance Weighted Autoencoders`
- Lower variance gradient estimator
  `- Sticking the Landing: Simple, Lower-Variance Gradient Estimators for Variational Inference`

Since the original VAE paper...

- Extension to the family of $q_\phi(z)$
  - Variational inference with normalizing flows
- Reparametrization trick with discrete latent variable
  - Categorical Reparameterization with Gumbel-Softmax
- Tighter Variational Bounds
  - Importance Weighted Autoencoders
- Lower variance gradient estimator
  - Sticking the Landing: Simple, Lower-Variance Gradient Estimators for Variational Inference
- Lots and lots of applications
  - Generative model with X using VAE
  - Semi-supervised learning

Since the original VAE paper...

- Extension to the family of $q_\phi(z)$
  - Variational inference with normalizing flows
- Reparametrization trick with discrete latent variable
  - Categorical Reparameterization with Gumbel-Softmax
- Tighter Variational Bounds
  - Importance Weighted Autoencoders
- **Lower variance gradient estimator**
  - Sticking the Landing: Simple, Lower-Variance Gradient Estimators for Variational Inference
- Lots and lots of applications
  - Generative model with X using VAE
  - Semi-supervised learning

Lower variance gradient estimator

- ELBO

$$\mathcal{L}(\phi) = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - KL(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \tag{3}$$

$$= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log(p(\mathbf{x}|\mathbf{z})) + \log(p(\mathbf{z})) - \log(q_\phi(\mathbf{z}|\mathbf{x}))] \tag{4}$$

Credit: Roeder *et al.*, 2017

Lower variance gradient estimator

- ELBO

$$\mathcal{L}(\phi) = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - KL(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \quad (3)$$

$$= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log(p(\mathbf{x}|\mathbf{z})) + \log(p(\mathbf{z})) - \log(q_\phi(\mathbf{z}|\mathbf{x}))] \quad (4)$$

- Gradient estimator, let $\mathbf{z} = t(\epsilon, \phi)$

$$\hat{\nabla}_{TD} = \nabla_\phi[\log p(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})]$$

$$= \underbrace{\nabla_{\mathbf{z}}[\log p(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})]\nabla_\phi t(\epsilon,\phi)}_{\text{path derivative}} - \underbrace{\nabla_\phi \log q_\phi(\mathbf{z}|\mathbf{x})}_{\text{score function}}$$

For any finite samples of $\mathbf{z}$ the score function is not necessarily zero, even when $q_\phi(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{z}|\mathbf{x})$.

Credit: Roeder *et al.*, 2017

Lower variance gradient estimator

- Remove the score function?

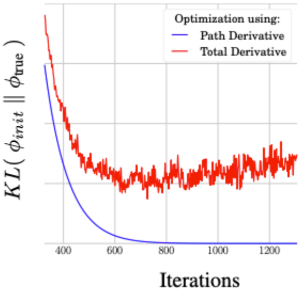
$$\hat{\nabla}_{PD} = \underbrace{\nabla_{\mathbf{z}}[\log p(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})]\nabla_\phi t(\epsilon,\phi)}_{\text{path derivative}} - \cancel{\underbrace{\nabla_\phi \log q_\phi(\mathbf{z}|\mathbf{x})}_{\text{score function}}}$$

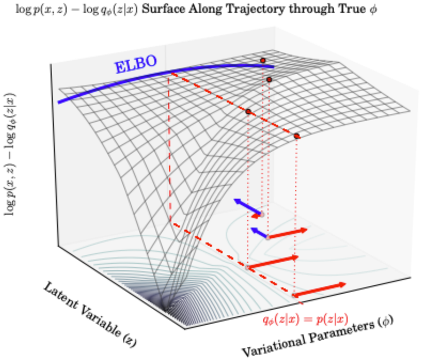
Lower variance gradient estimator

- Remove the score function?
  $\hat{\nabla}_{PD} = \underbrace{\nabla_{\mathbf{z}}[\log p(\mathbf{x}, \mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})]\nabla_\phi t(\epsilon, \phi)}_{\text{path derivative}} - \cancel{\underbrace{\nabla_\phi \log q_\phi(\mathbf{z}|\mathbf{x})}_{\text{score function}}}$
- The score function has expected value of zero, thus $\hat{\nabla}_{PD}$ is an unbaised estimator. Proof:

$$\begin{aligned}
\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\nabla_\phi \log q_\phi(\mathbf{z}|\mathbf{x})] &= \int \left(\nabla_\phi \log q_\phi(\mathbf{z}|\mathbf{x})\right) q(\mathbf{z}|\mathbf{x})\, d\mathbf{z} \\
&= \int \left(\nabla_\phi q_\phi(\mathbf{z}|\mathbf{x})\right) d\mathbf{z} \\
&= \nabla_\phi \int q_\phi(\mathbf{z}|\mathbf{x})\, d\mathbf{z} = 0
\end{aligned}$$

Lower variance gradient estimator





Credit: Roeder *et al.*, 2017