Variational Autoencoder and Extensions

Maha ELBAYAD

03/03/2017

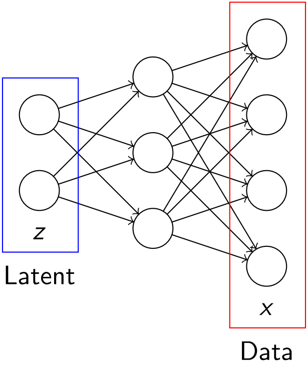
Outline

- Variational Autoencoder
- Incorporating normalizing flows
- Semi-supervised learning with VAE

Introduction

- Kingma and Welling, Auto-encoding Variational Bayes, ICLR 2014
- Rezende, Mohamed and Wierstra, Stochastic back-propagation and variational inference in deep latent Gaussian models, ICML 2014

A **latent** variable generative model using deep **directed** graphical models.



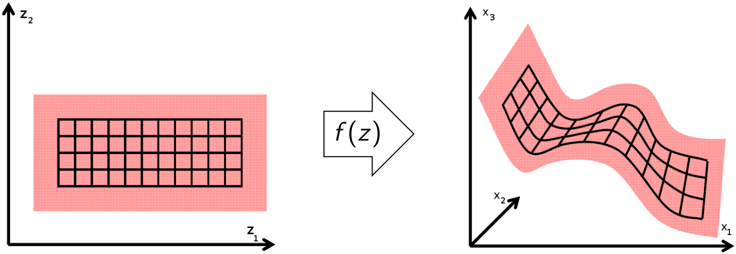
Latent variable generative model

Learn a mapping from some latent variable z to a complicated distribution on x.

$$p(x) = \int p(x,z)dz = \int p(x|z)p(z)dz$$

$$p(z) = \text{A simple distribution, usually } \mathcal{N}(z|0, I_D)$$

$$p(x|z) = f(z)$$



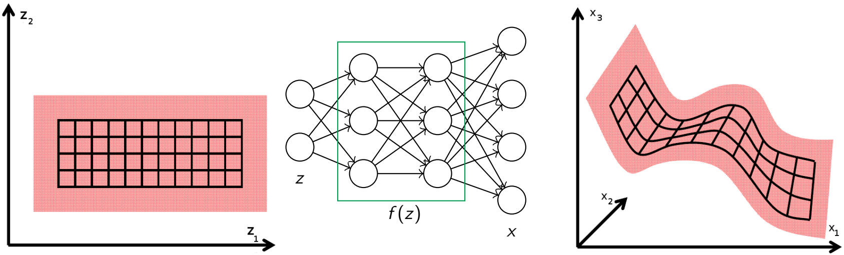
Variational Autoencoder approach

Leverage **neural networks** to learn a **continuous** latent variable model.

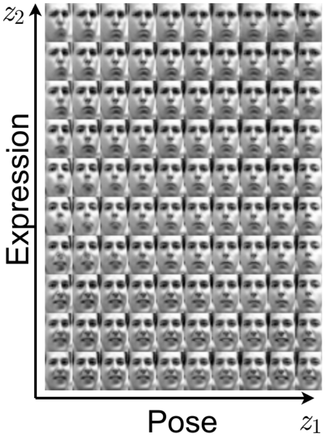
$$p(x) = \int p(x, z)dz = \int p(x|z)p(z)dz$$

$$p(z) = \text{A simple distribution, usually } \mathcal{N}(z|0, I_D)$$

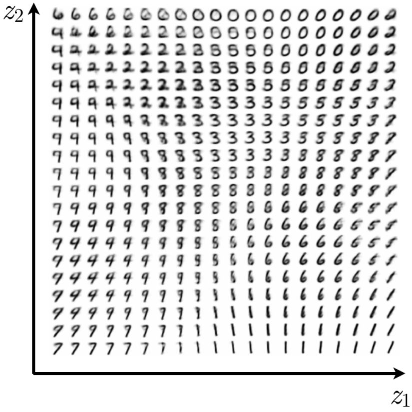
$$p(x|z) = f(z) = NN_\theta(z)$$



What VAE can do?



(a) Frey face dataset



(b) MNIST

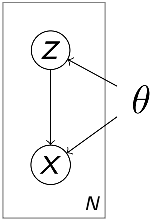
From Aaron Courville's slides (Deep Learning Summer School 2015)

VAE's approach

- How to infer z for a given sample x?
- How to compute/approximate the intractable posterior $p(z|x)$?
- How to train the directed model

VAE's approach

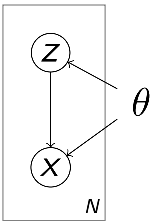
- How to infer z for a given sample x?
- How to compute/approximate the intractable posterior $p(z|x)$?
- How to train the directed model



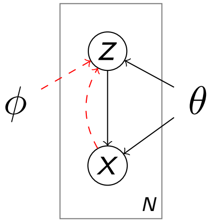
Generation

VAE's approach

- How to infer z for a given sample x?
- How to compute/approximate the intractable posterior $p(z|x)$?
- How to train the directed model



Generation

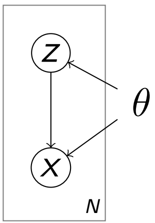


Fast approximate

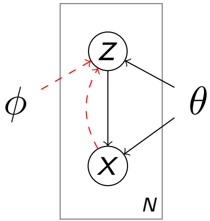
posterior inference

VAE's approach

- How to infer z for a given sample x?
- How to compute/approximate the intractable posterior $p(z|x)$?
- How to train the directed model



Generation



Fast approximate

posterior inference

Example:

$$q_\phi(z|x) = \mathcal{N}(z|\mu_z(x), \sigma_z(x)^2)$$
$$[\mu_z(x), \sigma_z(x)^2] = NN_\phi(x)$$

VAE's approach

- ~~How to infer z for a given sample x?~~
- ~~How to compute/approximate the intractable posterior $p(z|x)$?~~
- How to train the directed model

Variational lower bound (per sample)

$$\log p_\theta(x) = \mathcal{L}(x) + D_{KL}(q_\phi(z|x)||p_\theta(z|x))$$
$$\mathcal{L}(x) = E_{q_\phi(z|x)}\left[\log p_\theta(x,z) - \log q_\phi(z|x)\right]$$

The ELBO[1] ($\mathcal{L}$) is usually written as:

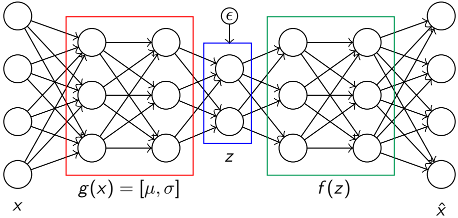
$$\mathcal{L}(x) = \underbrace{-D_{KL}(q_\phi(z|x)||p_\theta(z))}_{\text{Regularization term}} + \underbrace{E_{q_\phi(z|x)}[\log p_\theta(x|z)]}_{\text{Reconstruction term}}$$

[1] Evidence Lower BOund

Issue with backpropagation!

Reparemetrization trick: substitute a random variable by a deterministic transformation of a simpler random variable.[2]

$$z = \mu_z(x) + \sigma_z(x)\epsilon, \quad \text{where } \epsilon = \mathcal{N}(\epsilon|0, I_D)$$



Stochastic gradient variational bayes

$$E_{q_\phi(z|x)}[f(z)] \underset{MC}{\approx} \frac{1}{L}\sum_l f(\mu_z(x) + \sigma_z(x)\epsilon_l), \quad \epsilon_l \sim p(\epsilon)$$

[2](1) A tractable inverse CDF, (2) location-scale distributions (3) Composition

KL term collapse

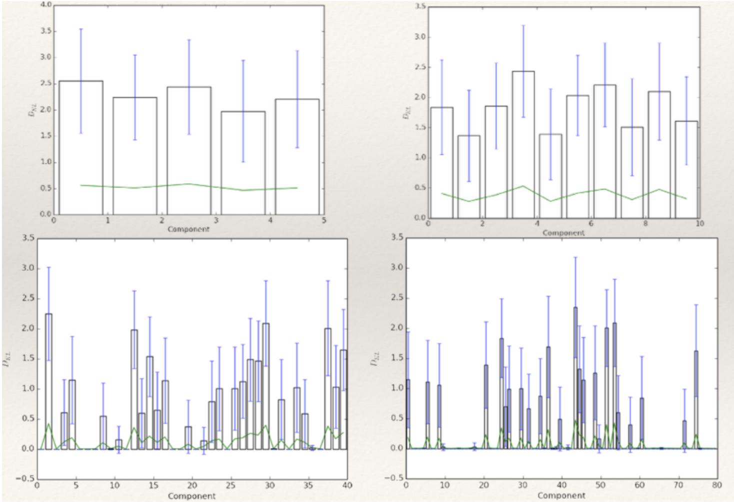


Figure from Laurent Dinh & Vincent Dumoulin

KL tempering (Sonderby et al. 2016 and Bowman et al. 2016)

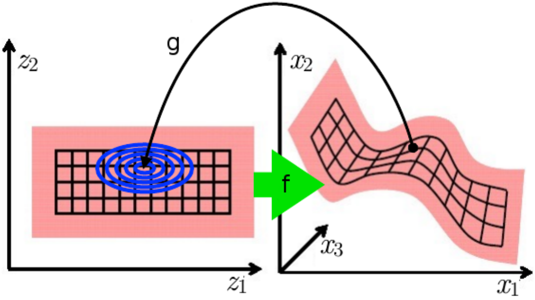
$$\tilde{\mathcal{L}}_\alpha \equiv -\alpha KL(q_\phi(z|x), p_\theta(z)) + E_q[\log p_\theta(x|z)], \quad \alpha : 0 \to 1$$

Inference in the VAE

The VAE factors the approximate posterior into:

$$q_\phi(z|x) = \prod_i q_\phi(z_i|x)$$

Unimodal gaussain distribution for $g(x) = q_\phi(z|x)$



Can we lessen this mean-field restriction and get closer to the true posterior $p_\theta(z|x)$?

Variational Inference with Normalizing Flows

Normalizing Flows: Improve the posterior's complexity

Rezende and Mohamed, ICML2015

Normalizing flow: The transformation of a pdf through a sequence of invertible mappings. The initial density $q_0(\mathbf{z}_0)$ *flows* through this sequence.
Scalability: A class of transformations for which the Jacobian determiant can be computed in linear time.

Approach

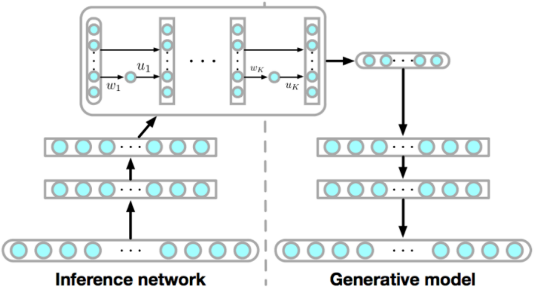
Start with $\mathbf{z}_0 \sim q_0$ and apply K normalizing flows $f_1, f_2, ..f_k$.

$$\ln q_K(\mathbf{z}_K) = \ln q_0(\mathbf{z}_K) - \prod_i^K \ln |J_i(z_{i-1})|^{-1}$$

Where J_i the Jacobian determinant of the i^{th} flow.

We can rewrite the variational lower bound as:

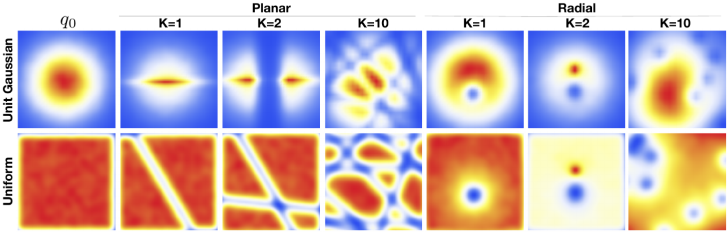
$$\begin{aligned}\tilde{\mathcal{L}}(x) &= E_{q_K}\left[\log p(x, \mathbf{z}_K) - \log q_K(\mathbf{z}_K)\right] \\ &= E_{q_0}[\log p(x, \mathbf{z}_K)] - E_{q_0}[\log q_0(\mathbf{z}_0)] + E_{q_0}\left[\sum_i^k \log |J_i(z_{i-1})|\right]\end{aligned}$$



Normalizing Flows: Improve the posterior's complexity

Rezende and Mohamed, ICML2015

Chaining these tranformations gives us a rich family of posteriors.



The effect of expansions and contractions on a uniform and Gaussian initial density using the proposed flows

- Planar: $f(z) = z + uh(w^T z + b)$, $(w, u \in \mathbb{R}^D, b \in \mathbb{R})$
- Radial: $f(z) = z + \frac{\beta}{\alpha + |z - z_0|}(z - z_0)$, $(z_0 \in \mathbb{R}^D, \alpha \in \mathbb{R}^+, \beta \in \mathbb{R})$

Semi-supervised learning with VAE

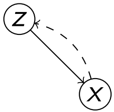
Semi-supervised learning with VAE

Kingma, Rezende, Mohamed and Welling (NIPS2014)

[M1] Standard unsupervised feature learning:

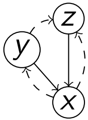
- Train z on unlabeled data.
- Train a classifier to map $z \rightarrow y$.

$$p(z) = \mathcal{N}(z|0, I_D),\ p_\theta(x|z) = f(x; z, \theta)$$



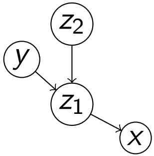
[M2] Generative semi-supervised model:

$$p(y) = cat(y|\pi),\ p(z) = \mathcal{N}(z|0, I_D)$$
$$p_\theta(x|z, y) = f(x; y, z, \theta)$$



[M1+M2] Combination semi-supervised model:

- Train on unsupervised features z_1
- Train M2 with z_1 as the modeled data.



Semi-supervised Learning with Deep Generative Models

Kingma, Rezende, Mohamed and Welling (NIPS2014)

The ELBO of the **M1+M2** model:

- With labeled data

$$\mathcal{L}_L(x, y) = E_{q_\phi(z|x,y)}\left[\log p_\theta(x|y, z) + \log p_\theta(y) + \log p(z) - \log q_\phi(z|x, y)\right]$$

- Without labels

$$\begin{aligned}\mathcal{L}_U(x, y) &= E_{q_\phi(z|x,y)}\left[\log p_\theta(x|y, z) + \log p_\theta(y) + \log p(z) - \log q_\phi(z|x, y)\right] \\ &= \sum_y q_\phi(y|x)\mathcal{L}_L(x, y) + \mathcal{H}(q_\phi(y|x))\end{aligned}$$

- The semi-supervised objective:

$$\mathcal{L}_\alpha = \sum_{(x,y)\sim\tilde{p}_l} \mathcal{L}_L(x, y) + \sum_{x\sim\tilde{p}_u} \mathcal{L}_U + \alpha\mathbb{E}_{\tilde{p}_l(x,y)}\left[\log q_\phi(y|x)\right]$$

  - Workaround for $q_\phi(y|x)$ contributing only to the unlabelled data term.
  - α controls the weight between generative and purely discriminative learning.

Semi-supervised Learning with Deep Generative Models

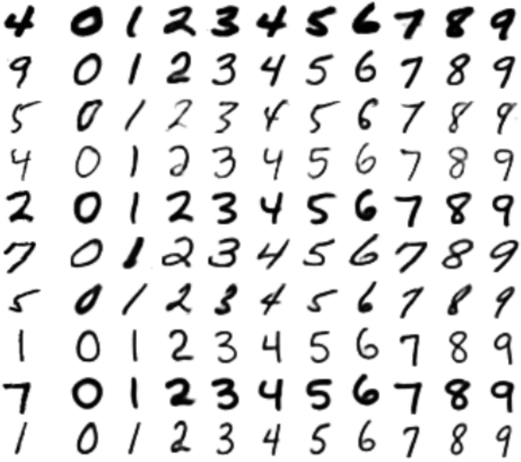
Kingma, Rezende, Mohamed and Welling (NIPS2014)

(a) Handwriting styles for MNIST obtained by fixing the class label and varying the 2D latent variable $\mathbf{z}$

Semi-supervised Learning with Deep Generative Models

Kingma, Rezende, Mohamed and Welling (NIPS2014)

Analogy making:



(b) MNIST analogies

Summary

- VAE applies to almost any directed model with continuous latent variables.
- Optimizes a lower bound of the marginal likelihood.
- Scales to very large datasets.
- Simple and fast.

For more on VAE:

- Salimans, Tim, Diederik P. Kingma, and Max Welling. "Markov Chain Monte Carlo and Variational Inference: Bridging the Gap." ICML2015.
- Chung, Junyoung, et al. "A recurrent latent variable model for sequential data." NIPS2015.
- Sønderby, Casper Kaae, et al. "Ladder variational autoencoders." NIPS2016.

Thanks!