An Introduction to Variational Inference

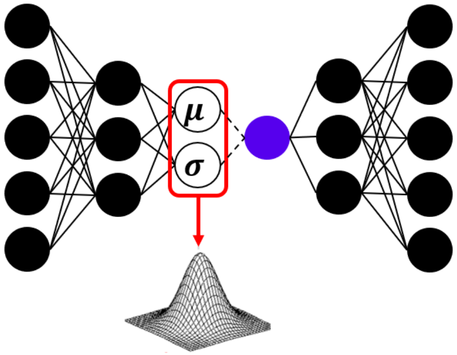
杨帆

2018年6月4日

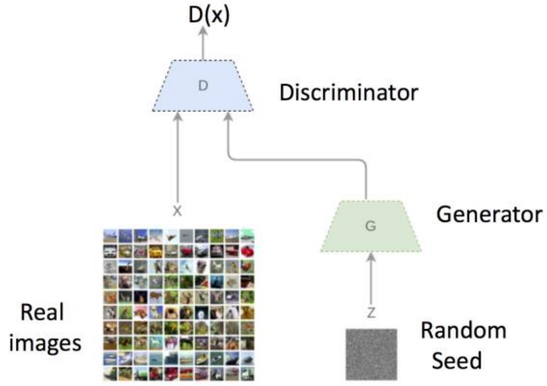
Motivations For This Talk

- Some trends in machine learning: make ML to be
  - Robust to uncertain and adversarial inputs
  - Unsupervised, semi-supervised or self-supervised
  - Interpretable
  - Nonparametric and automatic

- Probabilistic generative models are promising tools for these goals
  - E.g., two popular probabilistic generative models: VAE and GAN



Variational Autoencoders



Generative Adversarial Networks

This Talk

- Gives a high-level impression of how probabilistic model works
- Introduces the variational inference method
  - which is the basis of VAE
- Helps us understand the VAE

- NOTE: there will be some math and statistics, please interrupt me if you do not understand them.

Outline

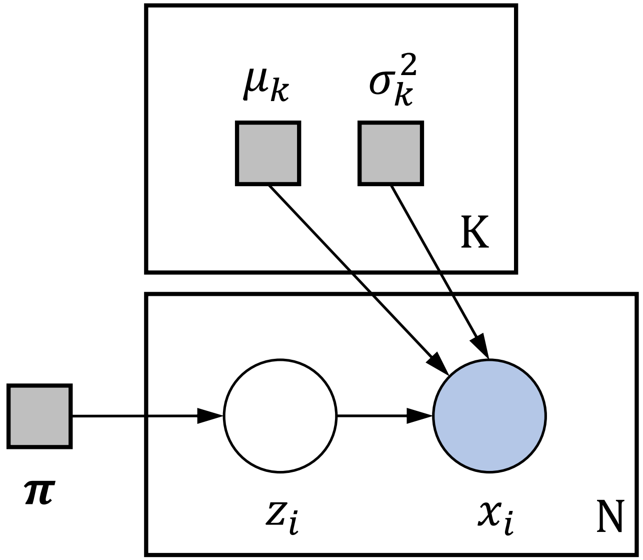
- Probabilistic Generative Models
- Variational Inference
- Variational Autoencoder

Outline

- **Probabilistic Generative Models**
- Variational Inference
- Variational Autoencoder

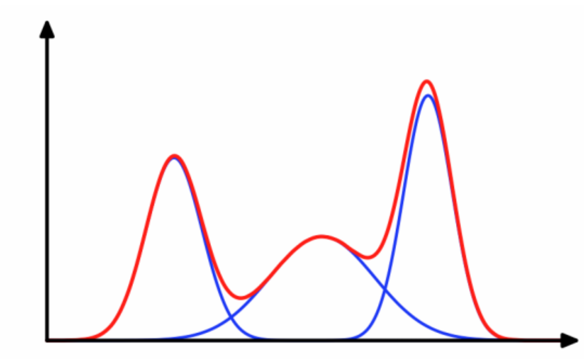
Probabilistic Generative Models

- Example: Gaussian Mixture Models



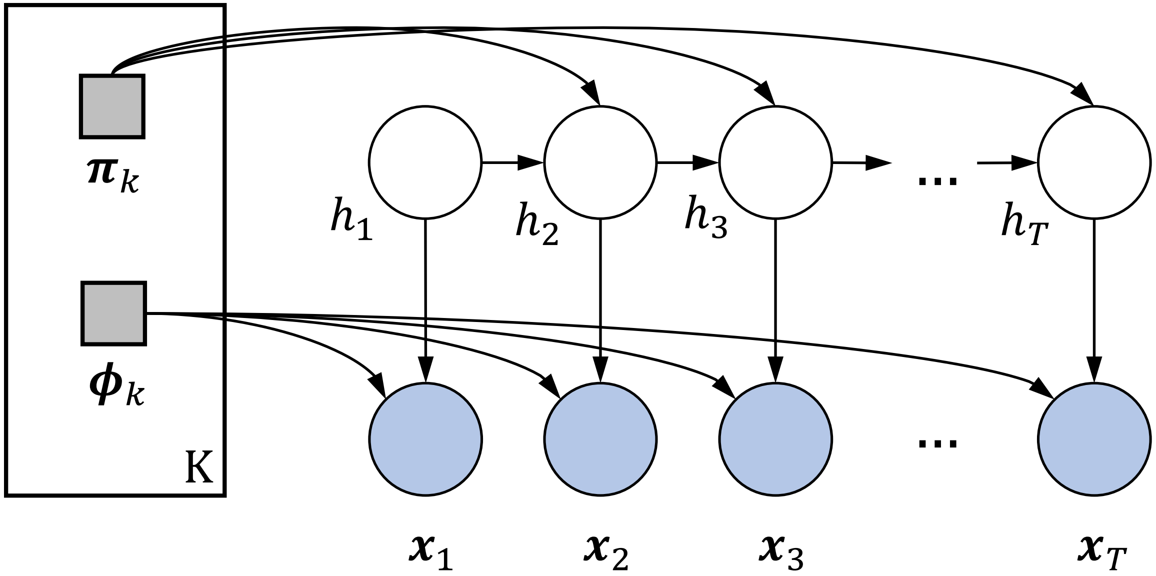
$$z_i \sim \text{categorical}(\pi_1, \dots, \pi_K), \quad i = 1, \dots, N,$$

$$x_i | z_i \sim \mathcal{N}\left(\mu_{z_i}, \sigma_{z_i}^2\right) \qquad i = 1, \dots, N.$$



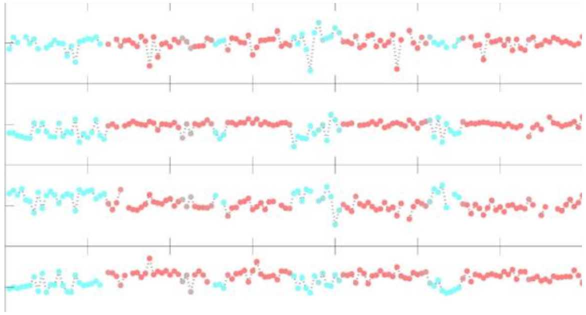
Probabilistic Generative Models

- Example: Hidden Markov Models



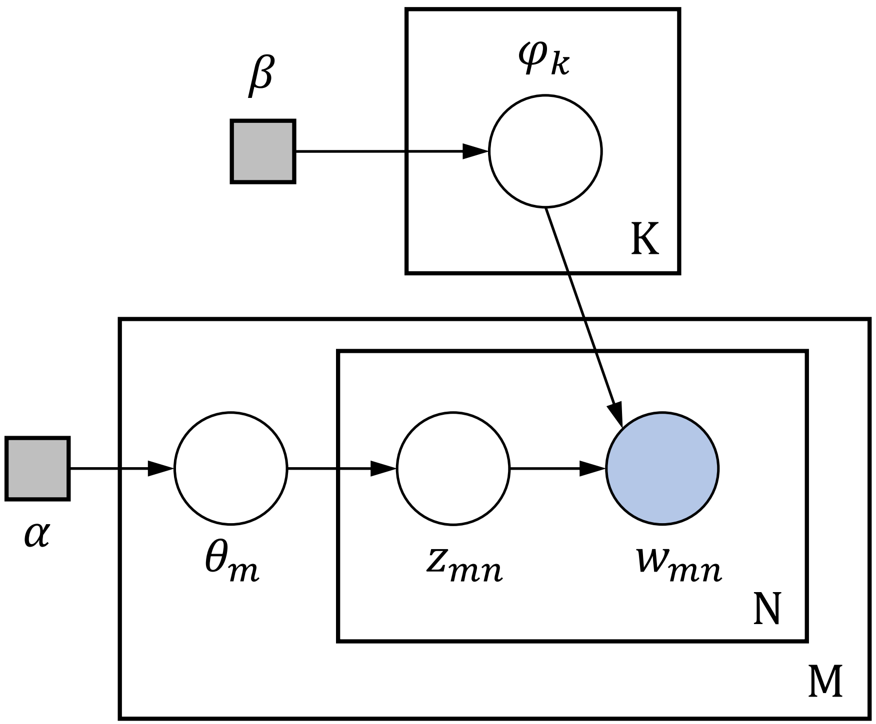
$$h_t \sim \text{categorical}\left(\boldsymbol{\pi}_{h_{t-1}}\right)$$

$$\boldsymbol{x}_t | h_t \sim F\left(\boldsymbol{\phi}_{h_t}\right)$$



Probabilistic Generative Models

- Example: Latent Dirichlet Allocation

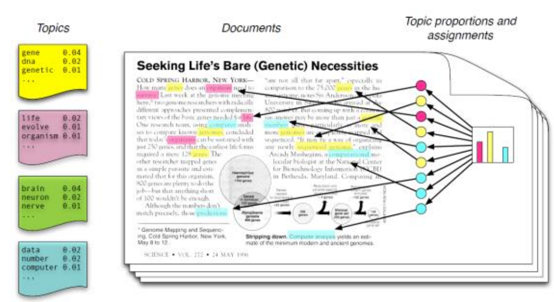


$$\varphi_k \sim Dir(\beta), \quad k = 1, \ldots, K,$$

$$\theta_m \sim Dir(\alpha), \quad m = 1, \ldots, \mathrm{M},$$

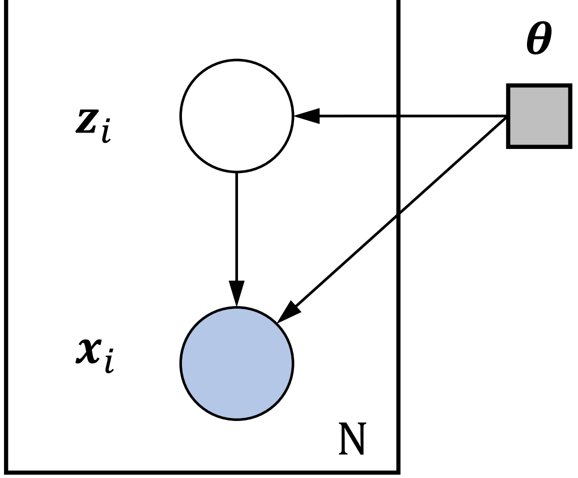
$$z_{mn} | \theta_m \sim \text{categorical}(\theta_m),$$

$$w_{mn} | \boldsymbol{\varphi}, z_{mn} \sim \text{categorical}(\varphi_{z_{mn}})$$



Probabilistic Generative Models

- Example: Deep Latent Gaussian Models



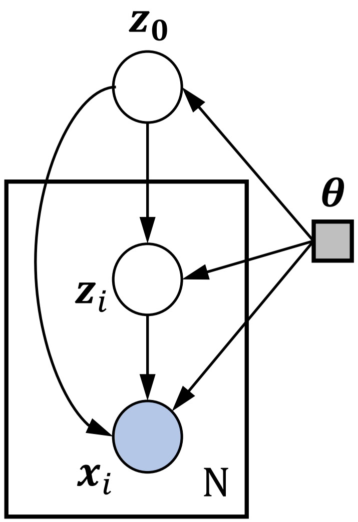
$$\boldsymbol{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbb{I}), \qquad i = 1, \dots, N,$$

$$\boldsymbol{x}_i | \boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{z}_i), \boldsymbol{\sigma}^2(\boldsymbol{z}_i)\mathbb{I}), \quad i = 1, \dots, N.$$

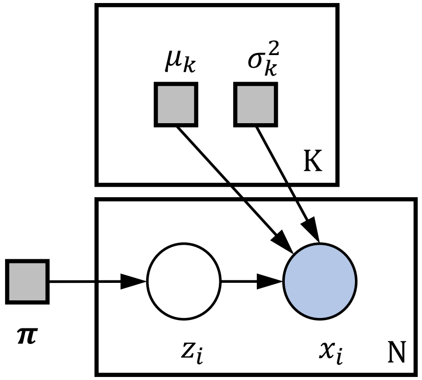
where $\boldsymbol{\mu}(\cdot)$ and $\boldsymbol{\sigma}(\cdot)$ are neural networks,
$\boldsymbol{\theta} = \{\text{parameters of } \boldsymbol{\mu} \text{ and } \boldsymbol{\sigma}\}$

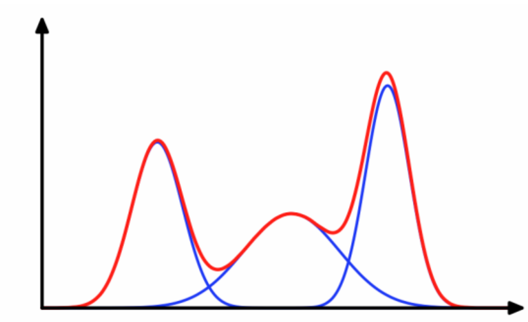
Probabilistic Generative Models: Basic Tasks

- Parameter Learning: fit the model to the dataset
  - Maximum likelihood estimation for the parameters $\boldsymbol{\theta}$
- Inference: compute unknown probability distributions
  - Posterior distribution of latent variable $\boldsymbol{z}$
  - Marginal distribution of observations $\boldsymbol{x}$



$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{x})}$$

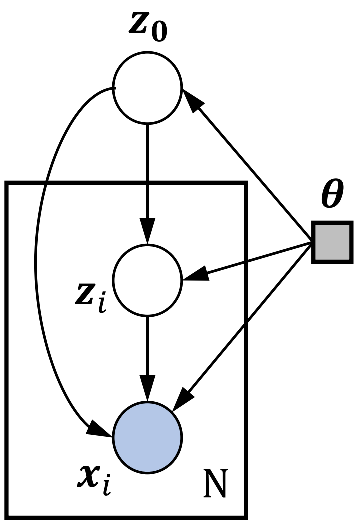




The Need for **Approximate** Inference

- Posterior distribution of latent variables $\boldsymbol{z} = \{\boldsymbol{z}_i\}, i = 1 \ldots \mathrm{M}$
- Marginal distribution of observations $\boldsymbol{x} = \{\boldsymbol{x}_i\}, i = 1 \ldots \mathrm{N}$

$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{x},\boldsymbol{z})}{p(\boldsymbol{x})} = \frac{\overbrace{p(\boldsymbol{x}|\boldsymbol{z})}^{\text{likelihood}}\,\overbrace{p(\boldsymbol{z})}^{\text{prior}}}{\underbrace{\int p(\boldsymbol{x},\boldsymbol{z})d\boldsymbol{z}}_{\text{marginal/evidence}}}$$



For many models, the integral (or sum) is **intractable**:

- Unavailable in closed form, or
- Requires exponential time to compute.

Approximate Inference of $p(\boldsymbol{z}|\boldsymbol{x})$

- Monte Carlo sampling: MCMC (Metropolis-Hasting or Gibbs sampling)
  - Approximate the posterior using samples
  - √ Converge to the posterior asymptotically
  - X Computationally intensive

- Variational Inference: turn inference into an optimization problem
  - Set up a ***family*** of approximate densities $\mathbb{Q}$ over the latent variables
  - Find the member q^* in the family $\mathbb{Q}$ that is closest to the exact posterior

$$q^*(\boldsymbol{z}) = \underset{q(\boldsymbol{z}) \in \mathbb{Q}}{\arg\min} \, \mathrm{KL}(q(\boldsymbol{z}) \| p(\boldsymbol{z}|\boldsymbol{x}))$$

  - √ Tends to be faster and easier to scale to large datasets

Outline

- Probabilistic Generative Models
- **Variational Inference**
- Variational Autoencoder

Variational Inference

- Optimization problem:

$$q^*(\boldsymbol{z}) = \underset{q(\boldsymbol{z}) \in \mathbb{Q}}{\arg\min}\, \mathrm{KL}(q(\boldsymbol{z}) \| p(\boldsymbol{z}|\boldsymbol{x}))$$

where

$$\mathrm{KL}(q(\boldsymbol{z}) \| p(\boldsymbol{z}|\boldsymbol{x})) = \int q(\boldsymbol{z}) \log \frac{q(\boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{x})} d\boldsymbol{z}$$

$$= \mathbb{E}_q[\log q(\boldsymbol{z})] - \mathbb{E}_q[\log p(\boldsymbol{z}|\boldsymbol{x})]$$

intractable

$$= \mathbb{E}_q[\log q(\boldsymbol{z})] - \mathbb{E}_q[\log p(\boldsymbol{z}, \boldsymbol{x})] + \log p(\boldsymbol{x})$$

constant, but intractable

- Because we cannot compute the KL, we optimize an alternative objective called **ELBO**:

$$\mathrm{ELBO}(q) = \mathbb{E}_q[\log p(\boldsymbol{z}, \boldsymbol{x})] - \mathbb{E}_q[\log q(\boldsymbol{z})]$$

  - Maximizing the ELBO is equivalent to minimizing the KL divergence

Interpretations of the ELBO

$$\text{ELBO}(q) = \mathbb{E}_q[\log p(\boldsymbol{z},\boldsymbol{x})] - \mathbb{E}_q[\log q(\boldsymbol{z})]$$

- **E**vidence **L**ower **Bo**und

$$\log p(\boldsymbol{x}) = \text{KL}(q(\boldsymbol{z}) \| p(\boldsymbol{z}|\boldsymbol{x})) + \text{ELBO}(q) \geq \text{ELBO}(q)$$

  - Maximizing the ELBO is equivalent to minimizing the KL divergence

---

- Another perspective

$$\text{ELBO}(q) = \mathbb{E}_q[\log p(\boldsymbol{z},\boldsymbol{x})] - \mathbb{E}_q[\log q(\boldsymbol{z})]$$

$$= \mathbb{E}_q[\log p(\boldsymbol{x}|\boldsymbol{z})] + \mathbb{E}_q[\log p(\boldsymbol{z})] - \mathbb{E}_q[\log q(\boldsymbol{z})]$$

$$= \mathbb{E}_q[\log p(\boldsymbol{x}|\boldsymbol{z})] - \text{KL}(q(\boldsymbol{z}) \| p(\boldsymbol{z}))$$

encourage q to place mass on configurations of $\boldsymbol{z}$ that explain the observed data $\boldsymbol{x}$

regularization term

Variational Inference: Maximizing the **ELBO**

$$\text{ELBO}(q) = \mathbb{E}_q[\log p(\boldsymbol{z}, \boldsymbol{x})] - \mathbb{E}_q[\log q(\boldsymbol{z})]$$

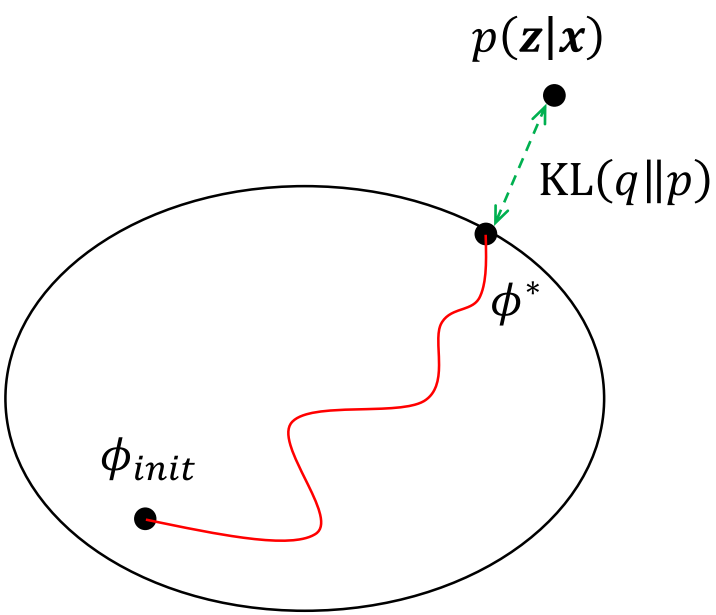
- Given a family of distributions $\mathbb{Q} = \{q_\phi\}$
  - ϕ are the parameters of these distributions
    - Called **variational parameters**
    - E.g., mean and variance of gaussians
- Our task: try to find

$$q^*(\boldsymbol{z}) = \underset{q_\phi(\boldsymbol{z}) \in \mathbb{Q}}{\arg\min} \, \text{KL}\big(q_\phi(\boldsymbol{z}) \| p(\boldsymbol{z}|\boldsymbol{x})\big)$$

equivalent to find:

$$\phi^* = \underset{\phi}{\arg\min} \, \text{KL}\big(q_\phi(\boldsymbol{z}) \| p(\boldsymbol{z}|\boldsymbol{x})\big)$$

$$= \underset{\boldsymbol{\phi}}{\textbf{arg max}} \, \textbf{ELBO}\big(\boldsymbol{q_\phi}\big)$$



Traditional Variational Inference

- Recall that $\boldsymbol{x} = \{\boldsymbol{x}_i\}, i = 1 \dots N$
- Expand the ELBO:

$$\mathrm{ELBO}(q) = \mathbb{E}_q[\log p(\boldsymbol{z}, \boldsymbol{x})] - \mathbb{E}_q[\log q(\boldsymbol{z})]$$

$$= \sum_{i=1}^{N} \mathbb{E}_q \left[\log \frac{p(\boldsymbol{z}, \boldsymbol{x}_i)}{q(\boldsymbol{z})} \right]$$

$$\int q(\boldsymbol{z})(\dots)d\boldsymbol{z}$$

- Traditional VI:

(1) Design a class of **tractable** densities $q_\phi(\boldsymbol{z}) \in \mathbb{Q}$

(2) Derive closed-form expression of the expectation

(3) Derive the gradient of the closed-form expectation

(4) Use coordinate ascent to update ϕ

Variational Inference: Modern Challenges

$$\mathrm{ELBO}(q) = \mathbb{E}_q[\log p(\boldsymbol{z}, \boldsymbol{x})] - \mathbb{E}_q[\log q(\boldsymbol{z})]$$

$$= \sum_{i=1}^{N} \mathbb{E}_q \left[\log \frac{p(\boldsymbol{z}, \boldsymbol{x}_i)}{q(\boldsymbol{z})} \right]$$

$$\int q(\boldsymbol{z})(\dots)d\boldsymbol{z}$$

- Traditional VI:

(1) Design a class of **tractable** densities $q_\phi(\boldsymbol{z}) \in \mathbb{Q}$

(2) Derive closed-form expression of $\mathbb{E}_q$

(3) Derive the gradient of the closed-form $\mathbb{E}_q$

(4) Use coordinate ascent to update ϕ

Modern challenges

- The ELBO involves the whole dataset, but dataset can be **large**
- We want a flexible family $\mathbb{Q}$ (e.g., neural networks), but for such $q \in \mathbb{Q}$, $\mathbb{E}_q$ is generally **intractable**
- We want to handle complex generative models

Variational Inference: Toward Modernization

- Using stochastic optimization to:
  - Scale up VI to massive data
  - Enable VI with flexible families of approximation densities
  - Enable VI on a wide class of complex/difficult models

$$\int q(\boldsymbol{z})(\dots)dz$$

$$\mathrm{ELBO}(q_\phi) = \sum_{i=1}^{N} \mathbb{E}_q \left[\log \frac{p(\boldsymbol{z}, \boldsymbol{x}_i)}{q_\phi(\boldsymbol{z})} \right]$$

Modern challenges

- ~~The ELBO involves the whole dataset, but dataset can be~~ **large** ← **mini-batch**
- We want a flexible family $\mathbb{Q}$ (e.g., neural networks), but for such $q \in \mathbb{Q}$, $\mathbb{E}_q$ is generally intractable
- We want to handle complex generative models

$$\nabla_\phi \,\mathrm{ELBO}(q_\phi) = \sum_{i=1}^{N} \nabla_\phi \mathbb{E}_q \left[\log \frac{p(\boldsymbol{z}, \boldsymbol{x}_i)}{q_\phi(\boldsymbol{z})} \right]$$

$$\approx \frac{N}{S} \sum_{i=1}^{S} \boxed{\nabla_\phi \mathbb{E}_q \left[\log \frac{p(\boldsymbol{z}, \boldsymbol{x}_i)}{q_\phi(\boldsymbol{z})} \right] \quad ???}$$

REINFORCE Gradients

$$\nabla_\phi \mathbb{E}_q \left[\log \frac{p(\boldsymbol{z}, \boldsymbol{x}_i)}{q_\phi(\boldsymbol{z})}\right]$$

- Remember that $\mathbb{E}_q$ is intractable: $\int q(\boldsymbol{z})(\dots)d\boldsymbol{z}$
- A similar problem in reinforcement learning: maximizing the expected reward f: $\mathbb{E}_p[f(\dots)]$
- REINFORCE gradients (also called *score function estimator*)

$$\nabla \mathbb{E}_p[f(\dots)] = \mathbb{E}_p[\nabla \log p(x) f(\dots)]$$

- REINFORCE gradient of the ELBO:

$$\nabla_\phi \mathbb{E}_q \left[\log \frac{p(\boldsymbol{z}, \boldsymbol{x}_i)}{q_\phi(\boldsymbol{z})}\right] = \mathbb{E}_q \left[\nabla_\phi \log q_\phi(\boldsymbol{z}) \log \frac{p(\boldsymbol{z}, \boldsymbol{x}_i)}{q_\phi(\boldsymbol{z})}\right]$$

$$\approx \frac{1}{M} \sum_{k=1}^{M} \nabla_\phi \log q_\phi(\boldsymbol{z}_k) \log \frac{p(\boldsymbol{z}_k, \boldsymbol{x}_i)}{q_\phi(\boldsymbol{z}_k)}, \qquad \boldsymbol{z}_k \sim q_\phi(\boldsymbol{z})$$

Monte Carlo Approximation of the Gradient

- We have a gradient estimator for each *single data point* $\boldsymbol{x}_i$:

$$\nabla_\phi \, \mathrm{ELBO}\left(q_\phi, \boldsymbol{x}_i\right) \approx \frac{1}{M} \sum_{k=1}^{M} \nabla_\phi \log q_\phi(\boldsymbol{z}_k) \log \frac{p(\boldsymbol{z}_k, \boldsymbol{x}_i)}{q_\phi(\boldsymbol{z}_k)}, \qquad \boldsymbol{z}_k \sim q_\phi(\boldsymbol{z})$$

- This enables scalable stochastic optimization
  - We can update the variational parameters ϕ using a single data point
- This also enables more flexible families of $q_\phi \in \mathbb{Q}$
  - Only require that we can sample from q, rather than requiring $\mathbb{E}_q$ is tractable

- Problem: the **variance** of this gradient estimator is high
  - There are some variance reduction techniques
- One key contribution of the VAE papers is that they proposed a new gradient estimator

Outline

- Probabilistic Generative Models
- Variational Inference
- **Variational Autoencoder**

Auto-Encoding Variational Bayes.
Kingma DP, Welling M. ICLR 2014

Stochastic Backpropagation and Approximate Inference in Deep Generative Models.
Rezende DJ, Mohamed S, Wierstra D. ICML 2014

Stronger Assumptions Enable A New Estimator

- Recall the REINFORCE gradient estimator:

$$\nabla_\phi \, \mathrm{ELBO}(q_\phi, \boldsymbol{x}_i) \approx \frac{1}{M}\sum_{k=1}^{M} \nabla_\phi \log q_\phi(\boldsymbol{z}_k) \log \frac{p(\boldsymbol{z}_k, \boldsymbol{x}_i)}{q_\phi(\boldsymbol{z}_k)}, \qquad \boldsymbol{z}_k \sim q_\phi(\boldsymbol{z})$$

- This estimator requires:
  - Sampling from q
  - Evaluation of $\nabla_\phi \log q_\phi(\boldsymbol{z})$ and $\log p(\boldsymbol{z}, \boldsymbol{x})$
- The VAE papers made two further assumptions:
  - Sampling from $q_\phi(\boldsymbol{z})$ can be reparametrized to sampling from a simple distribution (e.g., standard gaussian)

$$\boldsymbol{z} \sim q_\phi(\boldsymbol{z}) \Leftrightarrow$$
$$\boldsymbol{z} = \mathrm{transform}(\boldsymbol{\epsilon}, \phi), \qquad \boldsymbol{\epsilon} \sim \mathrm{simple}(\boldsymbol{\epsilon})$$

  - $\log p(\boldsymbol{z}, \boldsymbol{x})$ and $\log q_\phi(\boldsymbol{z})$ are differentiable with respect to $\boldsymbol{z}$

The Reparameterization Trick

- Now we assume the simple distribution is standard gaussian $\mathcal{N}$

$$\boldsymbol{z} \sim q_\phi(\boldsymbol{z}) \Leftrightarrow$$
$$\boldsymbol{z} = t(\boldsymbol{\epsilon}, \phi), \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon})$$

- Then

$$\nabla_\phi \mathrm{ELBO}(q_\phi, \boldsymbol{x}_i) = \nabla_\phi \mathbb{E}_q \left[\log \frac{p(\boldsymbol{z}, \boldsymbol{x}_i)}{q_\phi(\boldsymbol{z})}\right]$$

$$= \nabla_\phi \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon})} \left[\log \frac{p(t(\boldsymbol{\epsilon}, \phi), \boldsymbol{x}_i)}{q_\phi(t(\boldsymbol{\epsilon}, \phi))}\right] \quad \leftarrow \text{reparameterization}$$

$$= \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon})} \left[\nabla_\phi \log \frac{p(t(\boldsymbol{\epsilon}, \phi), \boldsymbol{x}_i)}{q_\phi(t(\boldsymbol{\epsilon}, \phi))}\right] \quad \leftarrow \text{property of gaussian}$$

$$= \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon})} \left[\nabla_{\boldsymbol{z}} \log \frac{p(\boldsymbol{z}, \boldsymbol{x}_i)}{q_\phi(\boldsymbol{z})} \nabla_\phi t(\boldsymbol{\epsilon}, \phi)\right] \quad \leftarrow \text{chain rule of derivative}$$

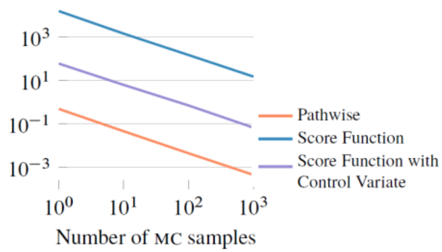
$$\boxed{\approx \frac{1}{M} \sum_{k=1}^{M} \nabla_{\boldsymbol{z}} \log \frac{p(\boldsymbol{z}, \boldsymbol{x}_i)}{q_\phi(\boldsymbol{z})} \nabla_\phi t(\boldsymbol{\epsilon}_k, \phi), \qquad \boldsymbol{\epsilon}_k \sim \mathcal{N}(\boldsymbol{\epsilon})}$$

Two Gradient Estimators

- REINFORCE gradient estimator (also call *score function estimator*)

$$\nabla_\phi \mathrm{ELBO}(q_\phi, \boldsymbol{x}_i) \approx \frac{1}{M}\sum_{k=1}^{M} \nabla_\phi \log q_\phi(\boldsymbol{z}_k) \log \frac{p(\boldsymbol{z}_k, \boldsymbol{x}_i)}{q_\phi(\boldsymbol{z}_k)}, \qquad \boldsymbol{z}_k \sim q_\phi(\boldsymbol{z})$$

  - Requires: (1) sampling from q. (2) evaluation of $\nabla_\phi \log q_\phi(\boldsymbol{z})$ and $\log p(\boldsymbol{z}, \boldsymbol{x})$
  - Variance can be a big problem



- Reparameterization trick (also called *path-wise gradient estimator*)

$$\nabla_\phi \mathrm{ELBO}(q_\phi, \boldsymbol{x}_i) \approx \frac{1}{M}\sum_{k=1}^{M} \nabla_{\boldsymbol{z}} \log \frac{p(\boldsymbol{z}, \boldsymbol{x}_i)}{q_\phi(\boldsymbol{z})} \nabla_\phi t(\boldsymbol{\epsilon}_k, \phi), \qquad \boldsymbol{\epsilon}_k \sim \mathcal{N}(\boldsymbol{\epsilon})$$

  - Requires: (1) $\boldsymbol{z}$ is parameterizable. (2) $\log p(\boldsymbol{z}, \boldsymbol{x})$ & $\log q_\phi(\boldsymbol{z})$ are differentiable
  - Variance is generally much smaller

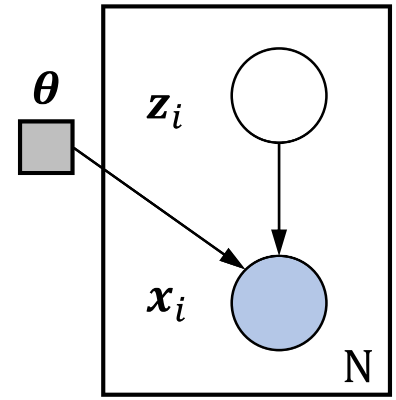
Variational Autoencoders

- Assume a deep latent gaussian generative model

$$\boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{0}, \mathbb{I}), \qquad i = 1, \dots, N,$$

$$\boldsymbol{x}_i | \boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{\mu_\theta}(\boldsymbol{z}_i), \boldsymbol{\sigma_\theta}^2(\boldsymbol{z}_i)\mathbb{I}), \quad i = 1, \dots, N.$$

  - where $\boldsymbol{\mu_\theta}(\cdot)$ and $\boldsymbol{\sigma_\theta}(\cdot)$ are neural networks



- Make q dependent on $\boldsymbol{x}$: $q_\phi(\boldsymbol{z}) \rightarrow q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \prod_{i=1}^{N} q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i)$
  - Model the dependence as a neural network (i.e., the *inference network*)

$$\boldsymbol{z}_i | \boldsymbol{x}_i \sim q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i) = \mathcal{N}\left(\boldsymbol{\mu_\phi}(\boldsymbol{x}_i), \boldsymbol{\sigma_\phi}^2(\boldsymbol{x}_i)\mathbb{I}\right)$$

where $\boldsymbol{\mu_\phi}(\cdot)$ and $\boldsymbol{\sigma_\phi}(\cdot)$ are neural networks

- Train the generative parameters $\boldsymbol{\theta}$ and the variational parameters ϕ together

Recall the Two Basic Tasks

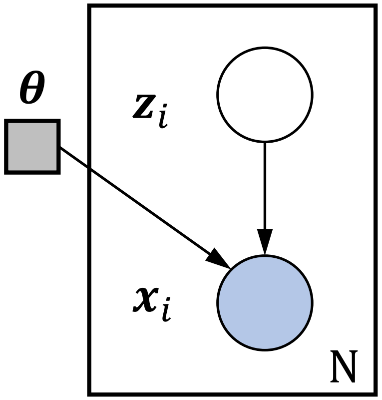
- Parameter Learning: fit the model to the dataset
  - Maximum likelihood estimation for the parameters $\boldsymbol{\theta}$
- Inference: compute unknown probability distributions
  - Posterior distribution of latent variable $\boldsymbol{z}$
  - Marginal distribution of observations $\boldsymbol{x}$

$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{x},\boldsymbol{z})}{p(\boldsymbol{x})}$$

- Generative parameters $\boldsymbol{\theta}$ enable us to generate new data
- Variational parameters $\boldsymbol{\phi}$ give an approximation of the posterior

$$p(\boldsymbol{z}_i|\boldsymbol{x}_i) \approx q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)$$

  - Useful for representation learning
    - $\boldsymbol{z}_i$ is the "code" of $\boldsymbol{x}_i$



Co-training of Generative and Variational Parameters

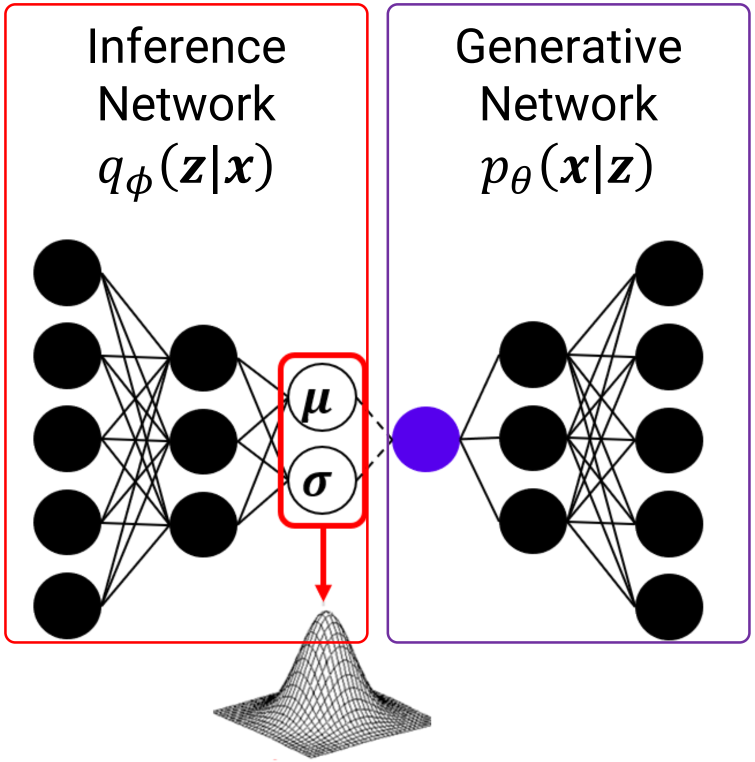
- Recall the ELBO is the Evidence Lower Bound

$$\mathrm{ELBO}(q_\phi) = \mathbb{E}_{q_\phi}[\log p_\theta(\boldsymbol{z}, \boldsymbol{x})] - \mathbb{E}_{q_\phi}[\log q_\phi(\boldsymbol{z}|\boldsymbol{x})]$$

$$\log p_\theta(\boldsymbol{x}) = \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \| p_\theta(\boldsymbol{z}|\boldsymbol{x})) + \mathrm{ELBO}(q_\phi) \geq \mathbf{ELBO}(q_\phi)$$

not a constant any more

optimizing ϕ makes q approximate the true posterior



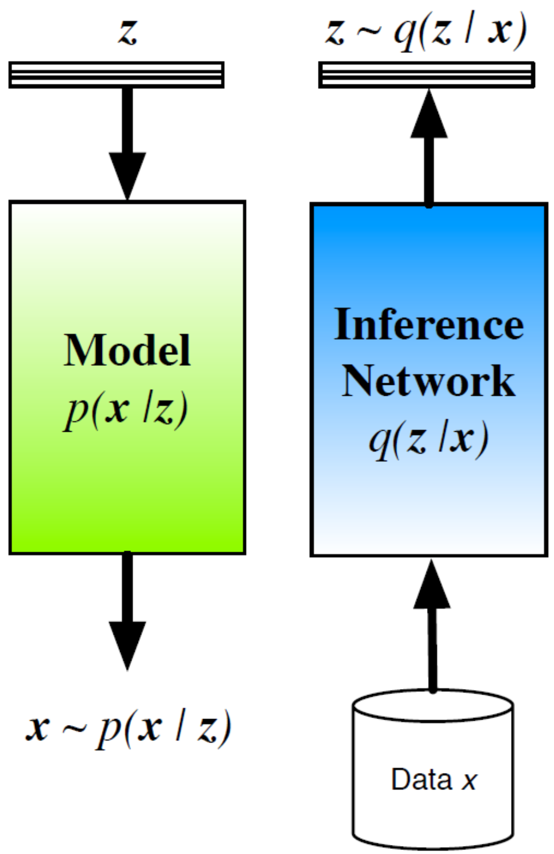
optimizing θ makes the generative model fit the data

Optimization of VAE

- Reformulation of the ELBO:

$$\mathbb{E}_{q_\phi}[\log p_\theta(\boldsymbol{x}_i|\boldsymbol{z}_i)] - \mathrm{KL}\left(q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i)\|p(\boldsymbol{z}_i)\right)$$

- First term: use the reparameterization trick to estimate the gradient
- Second term: solve the KL in closed-form
  - To reduce the variance of gradient estimator
- Stochastic gradient ascent on both $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$



Contributions of VAE

- Proposed the reparameterization tricks, which yield a low-variance gradient estimator
- Introduced the inference network
- Co-training of variational parameters and generative parameters

Recent Developments of VAE

- Divergences beyond KL, rethinking of ELBO
  - E.g., **W**asserstein **A**uto-**E**ncoders [ICLR 2018 Oral]
- More powerful and flexible families of approximation densities
  - E.g., Normalizing Flows
- Variance reduction of gradient estimators
  - E.g., Reducing Reparameterization Gradient Variance [NIPS 2017]
- Better ELBOs for structured models, such as sequential models
  - E.g., Auto-Encoding Sequential Monte Carlo [ICLR 2018]
- Combinations of VAE and GAN
- …

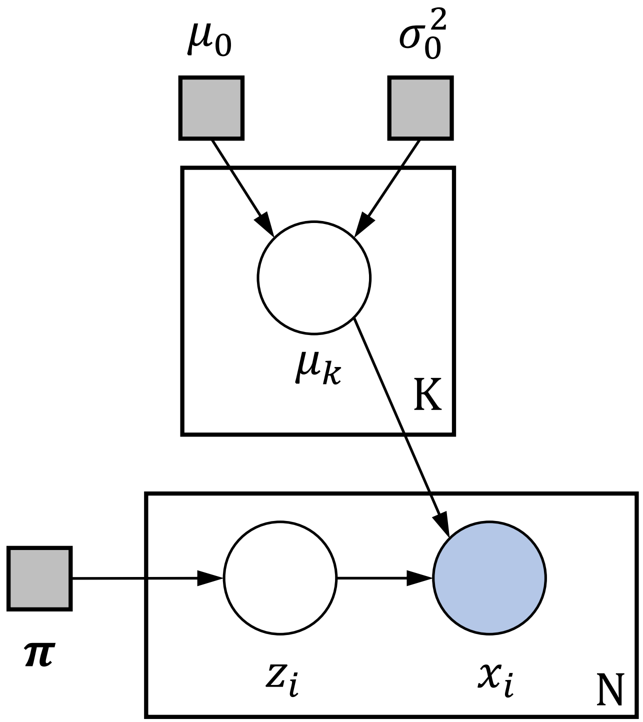
The End

Questions?

Backup Slides

Probabilistic Generative Models

- Example: Mixture of unit-variance univariate Gaussians



$$\mu_k \sim \mathcal{N}(\mu_0, \sigma_0^2), \qquad k = 1, \dots, K,$$

$$z_i \sim \text{categorical}(\pi_1, \dots, \pi_K), \qquad i = 1, \dots, N,$$

$$x_i | z_i, \boldsymbol{\mu} \sim \mathcal{N}\left(\mu_{z_i}, 1\right) \qquad i = 1, \dots, N.$$