DSC291: Advanced Statistical Natural Language Processing

Unsupervised Learning

Zhiting Hu

Lecture 8, April 21, 2022

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

Outline

- Variational Inference (VI)
- Stachastic VI; Black-box VI
- Variational Autoencoders (VAEs)

EM Algorithm

- Observed variables $\boldsymbol{x}$, latent variables $\boldsymbol{z}$
- To learn a model $p(\boldsymbol{x}, \boldsymbol{z}|\theta)$, we want to maximize the marginal log-likelihood

$$\ell(\theta; \boldsymbol{x}) = \log p(\boldsymbol{x}|\theta) = \log \sum_z p(\boldsymbol{x}, \boldsymbol{z}|\theta)$$

  - But it's too difficult
- EM algorithm:
  - maximize a lower bound of $\ell(\theta; \boldsymbol{x})$
  - Or equivalently, minimize an upper bound of $\ell(\theta; \boldsymbol{x})$
- Key equation:

$$\ell(\theta; \boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$

EM Algorithm

- Observed variables $\boldsymbol{x}$, latent variables $\boldsymbol{z}$
- To learn a model $p(\boldsymbol{x}, \boldsymbol{z}|\theta)$, we want to maximize the marginal log-likelihood

$$\ell(\theta; \boldsymbol{x}) = \log p(\boldsymbol{x}|\theta) = \log \sum_z p(\boldsymbol{x}, \boldsymbol{z}|\theta)$$

  - But it's too difficult
- EM algorithm:
  - maximize a lower bound of $\ell(\theta; \boldsymbol{x})$
  - Or equivalently, minimize an upper bound of $\ell(\theta; \boldsymbol{x})$
- Key equation:

Evidence Lower Bound (ELBO)

$$\ell(\theta; \boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$

$$= -F(q, \theta) + \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$

Variational free energy

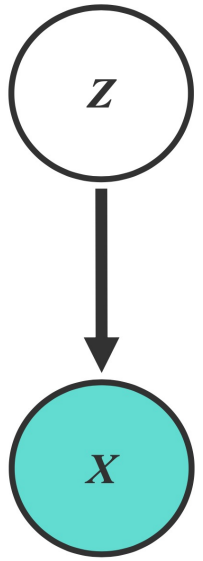
EM Algorithm

- The EM algorithm is coordinate-decent on $F(q, \theta)$
  - E-step: $q^{t+1} = \arg\min_q F\left(q, \theta^t\right) = p(\boldsymbol{z}|\boldsymbol{x}, \theta^t)$
    - the posterior distribution over the latent variables given the data and the current parameters
  - M-step: $\theta^{t+1} = \arg\min_\theta F\left(q^{t+1}, \theta^t\right) = \text{argmax}_\theta \sum_z q^{t+1}(\boldsymbol{z}|\boldsymbol{x}) \log p(\boldsymbol{x}, \boldsymbol{z}|\theta)$

$$\ell(\theta; \boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + \text{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$

$$= -F(q, \theta) + \text{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$

Example: Gaussian Mixture Models (GMMs)



- Consider a mixture of K Gaussian components:
  - Z is a latent class indicator vector:

$$p(z_n) = \text{multi}(z_n : \pi) = \prod_k (\pi_k)^{z_n^k}$$

  - X is a conditional Gaussian variable with a class-specific mean/covariance

$$p(x_n \mid z_n^k = 1, \mu, \Sigma) = \frac{1}{(2\pi)^{m/2}|\Sigma_k|^{1/2}} \exp\left\{-\tfrac{1}{2}(x_n - \mu_k)^T \Sigma_k^{-1}(x_n - \mu_k)\right\}$$

  - The likelihood of a sample:

$$p(x_n|\mu,\Sigma) = \sum_k p(z^k = 1 \mid \pi) p(x, \mid z^k = 1, \mu, \Sigma)$$
$$= \sum_{z_n} \prod_k \left((\pi_k)^{z_n^k} N(x_n : \mu_k, \Sigma_k)^{z_n^k}\right) = \sum_k \pi_k N(x, \mid \mu_k, \Sigma_k)$$

mixture component

mixture proportion

EM Algorithm for GMM

- Initialize the means μ_k , covariances Σ_k and mixing coefficients π_k
- Iterate until convergence:
  - E-step: Evaluate the posterior given current parameters

$$p(z^k = 1 \mid \boldsymbol{x}) = \frac{\pi_k \mathcal{N}\left(\boldsymbol{x} \mid \mu_k, \Sigma_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\boldsymbol{x} \mid \mu_j, \Sigma_j\right)} := \gamma_k$$

  - M-step: Re-estimate the parameters given current posterior

$$\begin{aligned}
&\mathbb{E}_{q^{t+1}}\left[\log\left(p\left(\boldsymbol{x}, z \mid \boldsymbol{\theta}\right)\right)\right] \\
=&\sum_k \gamma_k \left(\log p\left(z^k = 1 | \boldsymbol{\theta}\right) + \log P\left(\boldsymbol{x} \mid z^k = 1, \boldsymbol{\theta}\right)\right) \\
=&\sum_k \gamma_k \log \pi_k + \sum_k \gamma_k \log \mathcal{N}\left(\boldsymbol{x}; \mu_k, \Sigma_k\right)
\end{aligned}$$

EM Algorithm

- The EM algorithm is coordinate-decent on $F(q, \theta)$
  - E-step: $q^{t+1} = \arg\min_q F\left(q, \theta^t\right) = p(\boldsymbol{z}|\boldsymbol{x}, \theta^t)$
  - M-step: $\theta^{t+1} = \arg\min_\theta F\left(q^{t+1}, \theta^t\right) = \text{argmax}_\theta \sum_z q^{t+1}(\boldsymbol{z}|\boldsymbol{x}) \log p(\boldsymbol{x}, \boldsymbol{z}|\theta)$

$$\ell(\theta; \boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + \text{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$

$$= -F(q, \theta) + \text{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$

- Limitation: need to be able to compute $p(\boldsymbol{z}|\boldsymbol{x}, \theta)$, not possible for more complicated models --- solution: Variational inference

Variational Inference

Content adapted from CMU 10-708 Spring 2017

Inference

- Given a model, the goals of inference can include:
  - Computing the likelihood of observed data $p(\boldsymbol{x}^*)$
  - Computing the marginal distribution over a given subset of variables in the model $p(\boldsymbol{x}_A)$
  - Computing the conditional distribution over a subsets of nodes given a disjoint subset of nodes $p(\boldsymbol{x}_A|\boldsymbol{x}_B)$
  - Computing a mode of the density (for the above distributions) $\mathrm{argmax}_{\boldsymbol{x}}\, p(\boldsymbol{x})$
  - ….

Variational Inference

- Observed variables $\boldsymbol{x}$, latent variables $\boldsymbol{z}$
- Variational (Bayesian) inference, a.k.a. **variational Bayes**, is most often used to approximately infer the conditional distribution over the latent variables given the observations (and parameters)
  - i.e., the **posterior distribution** over the latent variables

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

Motivating Example

- Why do we often need to use an approximate inference methods (such as variational Bayes) to compute the posterior distribution?

- It's because we cannot directly compute the posterior distribution for many interesting models
  - I.e. the posterior density is in an intractable form (often involving integrals) which cannot be easily analytically solved.

- As a motivating example, we will try to compute the posterior for a (Bayesian) mixture of Gaussians.

Bayesian mixture of Gaussians

- The mean μ_k is treated as a (latent) random variable

$$\mu_k \sim \mathcal{N}(0, \tau^2) \text{ for } k = 1, \ldots, K$$

- For each data $i = 1, \ldots, n$

$$z_i \sim \text{Cat}(\pi).$$

$$x_i \sim \mathcal{N}(\mu_{z_i}, \sigma^2).$$

- We have
  - observed variables $x_{1:n}$
  - latent variables $\mu_{1:k}$ and $z_{1:n}$
  - parameters $\{\tau^2, \pi, \sigma^2\}$
- $p(x_{1:n}, z_{1:n}, \mu_{1:k} | \tau^2, \pi, \sigma^2) = \prod_{k=1}^{K} p(\mu_k) \prod_{i=1}^{n} p(z_i) p(x_i | z_i, \mu_{1:K})$

Bayesian mixture of Gaussians

- We can write the posterior distribution as

$$p(\mu_{1:K}, z_{1:n}|x_{1:n}) = \frac{\prod_{k=1}^{K} p(\mu_k) \prod_{i=1}^{n} p(z_i) p(x_i|z_i, \mu_{1:K})}{\int_{\mu_{1:K}} \sum_{z_{1:n}} \prod_{k=1}^{K} p(\mu_k) \prod_{i=1}^{n} p(z_i) p(x_i|z_i, \mu_{1:K})}$$

- The numerator can be computed for any choice of the latent variables
- The problem is the denominator (the marginal probability of the observations)
  - This integral cannot easily be computed analytically
- We need some approximation..

Variational Inference

The main idea behind variational inference:

- Choose a family of distributions over the latent variables $z_{1:m}$ with its own set of variational parameters ν , i.e.

$$q(z_{1:m}|\nu)$$

- Then, we find the setting of the parameters that makes our approximation q closest to the posterior distribution.
  - This is where optimization algorithms come in.

- Then we can use q with the fitted parameters in place of the posterior.
  - E.g. to form predictions about future data, or to investigate the posterior distribution over the hidden variables, find modes, etc.

Variational Inference

- We want to minimize the KL divergence between our approximation $q(\boldsymbol{z}|\boldsymbol{x})$ and our posterior $p(\boldsymbol{z}|\boldsymbol{x})$

$$\mathrm{KL}(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}))$$

  - But we can't actually minimize this quantity w.r.t q because $p(\boldsymbol{z}|\boldsymbol{x})$ is unknown

Evidence Lower Bound (ELBO)

$$\ell(\theta; \boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$

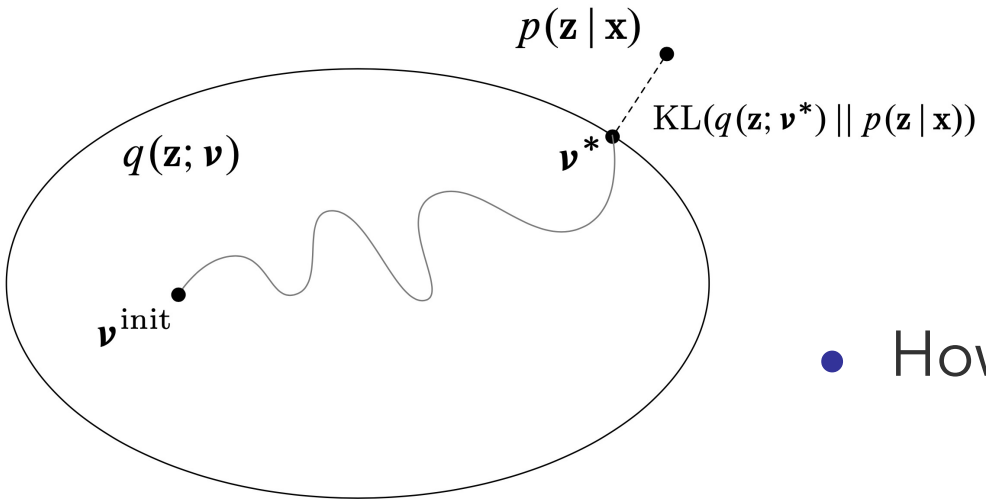
- The ELBO is equal to the negative KL divergence up to a constant $\ell(\theta; \boldsymbol{x})$
- We maximize the ELBO over q to find an "optimal approximation" to $p(\boldsymbol{z}|\boldsymbol{x})$

Variational Inference

- Choose a family of distributions over the latent variables $\boldsymbol{z}$ with its own set of variational parameters ν , i.e. $q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})$
- We maximize the ELBO over q to find an "optimal approximation" to $p(\boldsymbol{z}|\boldsymbol{x})$

$$\text{argmax}_{\nu}\ \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})}\right]$$

$$= \text{argmax}_{\nu}\ \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})}[\log p(\boldsymbol{x},\boldsymbol{z}|\theta)] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})}[\log q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})]$$



- How do we choose the variational family $q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})$?

Mean Field Variational Inference

- A popular family of variational approximations

- In this type of variational inference, we assume the variational distribution over the latent variables factorizes as

$$q(\mathbf{z}) = q(z_1, \ldots, z_m) = \prod_{j=1}^{m} q(z_j)$$

  - (where we omit variational parameters for ease of notation)
  - We refer to $q(z_j)$, the variational approximation for a single latent variable, as a "local variational approximation"

- In the above expression, the variational approximation $q(z_j)$ over each latent variable z_j is independent

Mean Field Variational Inference

- Note that this is a fairly general setup; we can also partition the latent variables $z_1, \ldots, z_m$ into R groups $z_{G_1}, \ldots, z_{G_R}$, and use the approximation:

$$q(z_1, \ldots, z_m) = q(z_{G_1}, \ldots, z_{G_R}) = \prod_{r=1}^{R} q(z_{G_r})$$

  - Often called “generalized mean field” versus (the above) “naïve mean field”.

Mean Field Variational Inference

- Note that this is a fairly general setup; we can also partition the latent variables $z_1, \ldots, z_m$ into R groups $z_{G_1}, \ldots, z_{G_R}$, and use the approximation:

$$q(z_1, \ldots, z_m) = q(z_{G_1}, \ldots, z_{G_R}) = \prod_{r=1}^{R} q(z_{G_r})$$

  - Often called "generalized mean field" versus (the above) "naïve mean field".

- Typically, this approximation does not contain the true posterior (because the latent variables are dependent).
  - E.g.: in the (Bayesian) mixture of Gaussians model, all of the cluster assignments z_i for $i = 1, \ldots, n$ are dependent on each other and on the cluster locations $\mu_{1:K}$ given data.

Optimizing the ELBO in Mean Field Variational Inference

How do we optimize the ELBO in mean field variational inference?

- Typically, we use coordinate ascent optimization.
- I.e. we optimize each latent variable's variational approximation $q(z_j)$ in turn while holding the others fixed.
  - At each iteration we get an updated "local" variational approximation.
  - And we iterate through each latent variable until convergence.

Optimizing the ELBO in Mean Field Variational Inference

- Recall that the ELBO is defined as:

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

- Note that we can decompose the entropy term of the ELBO (using the mean field variational approximation) as:

$$\mathbb{E}_q\left[\log q(z_{1:m})\right] = \sum_{j=1}^{m} \mathbb{E}_{q_j}\left[\log q(z_j)\right]$$

- Therefore, under the mean field approximation, the ELBO can be written:

$$\mathcal{L} = \mathbb{E}_{q_j}\mathbb{E}_{q_{-j}}[\log p(\boldsymbol{x}, \boldsymbol{z})] - \sum_{j=1}^{m} \mathbb{E}_{q_j}[\log q(z_j)]$$

Optimizing the ELBO in Mean Field Variational Inference

- Therefore, under the mean field approximation, the ELBO can be written:

$$\mathcal{L} = \mathbb{E}_{q_j}\mathbb{E}_{q_{-j}}[\log p(\boldsymbol{x}, \boldsymbol{z})] - \sum_{j=1}^{m} \mathbb{E}_{q_j}\big[\log q(z_j)\big]$$

- Next, we want to derive the coordinate ascent update for a latent variable z_j , keeping all other latent variables fixed.
  - i.e. we want the $\text{argmax}_{q_j}\mathcal{L}$.

- Removing the parts that do not depend on $q(z_j)$, we can write:

$$\mathcal{L} = \mathbb{E}_{q_j}\mathbb{E}_{q_{-j}}[\log p(\boldsymbol{x}, \boldsymbol{z})] - \mathbb{E}_{q_j}\big[\log q\big(z_j\big)\big] + \text{const.}$$

Optimizing the ELBO in Mean Field Variational Inference

- To find this argmax, we take the derivative of $\mathcal{L}$ w.r.t $q(z_j)$ and and set the derivative to zero :

$$\frac{d\mathcal{L}}{dq(z_j)} = \mathbb{E}_{q_j}\mathbb{E}_{q_{-j}}[\log p(\boldsymbol{x}, \boldsymbol{z})] - \log q(z_j) - 1 = 0$$

- From this, we arrive at the coordinate ascent update:

$$q^*(z_j) \propto \exp\left\{ \mathbb{E}_{q_{-j}}[\log p(\boldsymbol{x}, \boldsymbol{z})] \right\}$$

Optimizing the ELBO in Mean Field Variational Inference

- The coordinate ascent update:

$$q^*(z_j) \propto \exp\left\{ \mathbb{E}_{q_{-j}}[\log p(\boldsymbol{x}, \boldsymbol{z})] \right\}$$

  - The optimal solution for factor $q(z_j)$ is obtained simply by considering the log of the joint distribution over all observed and latent variables and then taking the expectation with respect to all of the other factors $q(z_k)$, $k \neq j$, then taking exponential and normalizing

- Note that the only assumption we made so far is the mean-field factorization:

$$q(\mathbf{z}) = q(z_1, \ldots, z_m) = \prod_{j=1}^{m} q(z_j)$$

  - We haven't yet made any assumptions on the form of $q(z_j)$

Simple example:

- Consider a univariate Gaussian distribution $p(x) = \mathcal{N}(x|\mu, \tau^{-2})$, given a dataset $\mathcal{D} = \{x_1, \ldots, x_N\}$:

$$p(\mathcal{D}|\mu, \tau) = \left(\frac{\tau}{2\pi}\right)^{N/2} \exp\left\{-\frac{\tau}{2}\sum_{n=1}^{N}(x_n - \mu)^2\right\}$$

$$p(\mu|\tau) = \mathcal{N}\left(\mu|\mu_0, (\lambda_0\tau)^{-1}\right)$$

$$p(\tau) = \mathrm{Gam}(\tau|a_0, b_0)$$

  - $\mathrm{Gam}(\tau|a_0, b_0) = \frac{1}{\Gamma(a)} b^a \lambda^{a-1} \exp(-b\lambda)$: gamma distribution

- For this simple problem the posterior distribution can be found exactly. But we use it as an example for tutorial anyway

$$q^*(z_j) \propto \exp\left\{ \mathbb{E}_{q_{-j}}[\log p(\boldsymbol{x}, \boldsymbol{z})] \right\}$$

Simple example:

$$p(\mathcal{D}|\mu,\tau) = \left(\frac{\tau}{2\pi}\right)^{N/2} \exp\left\{-\frac{\tau}{2}\sum_{n=1}^{N}(x_n-\mu)^2\right\} \qquad \begin{aligned} p(\mu|\tau) &= \mathcal{N}\left(\mu|\mu_0,(\lambda_0\tau)^{-1}\right) \\ p(\tau) &= \mathrm{Gam}(\tau|a_0,b_0) \end{aligned}$$

- Introduce the factorized variational approximation: $q(\mu,\tau) = q_\mu(\mu)q_\tau(\tau)$
- Solution to q_μ:

$$\begin{aligned} \ln q_\mu^\star(\mu) &= \mathbb{E}_\tau\left[\ln p(\mathcal{D}|\mu,\tau) + \ln p(\mu|\tau)\right] + \text{const} \\ &= -\frac{\mathbb{E}[\tau]}{2}\left\{\lambda_0(\mu-\mu_0)^2 + \sum_{n=1}^{N}(x_n-\mu)^2\right\} + \text{const.} \end{aligned}$$

  - We can see q_μ^* is a Gaussian $\mathcal{N}(x|\mu_N,\lambda_N^{-1})$:

$$\begin{aligned} \mu_N &= \frac{\lambda_0\mu_0 + N\overline{x}}{\lambda_0 + N} \\ \lambda_N &= (\lambda_0 + N)\mathbb{E}[\tau] \end{aligned}$$

$$q^*(z_j) \propto \exp\left\{ \mathbb{E}_{q_{-j}}[\log p(\boldsymbol{x}, \boldsymbol{z})] \right\}$$

Simple example:

$$p(\mathcal{D}|\mu,\tau) = \left(\frac{\tau}{2\pi}\right)^{N/2} \exp\left\{-\frac{\tau}{2}\sum_{n=1}^{N}(x_n-\mu)^2\right\} \qquad \begin{aligned} p(\mu|\tau) &= \mathcal{N}\left(\mu|\mu_0,(\lambda_0\tau)^{-1}\right) \\ p(\tau) &= \mathrm{Gam}(\tau|a_0,b_0) \end{aligned}$$

- Introduce the factorized variational approximation: $q(\mu,\tau) = q_\mu(\mu)q_\tau(\tau)$
- Solution to q_τ:

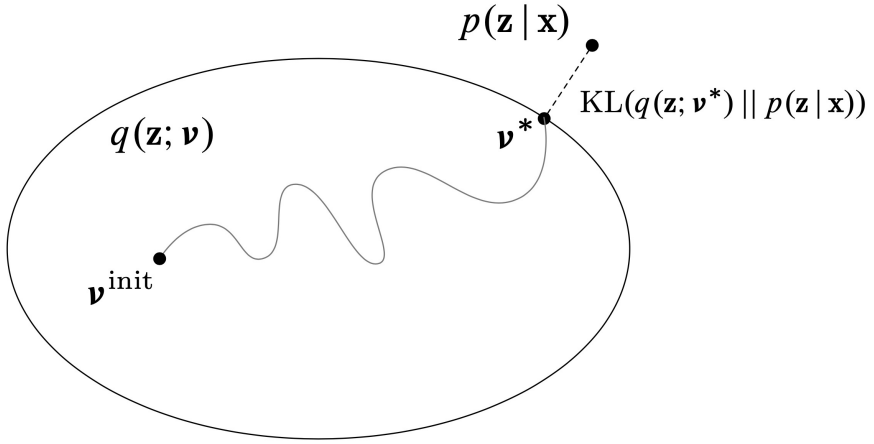
$$\begin{aligned} \ln q_\tau^\star(\tau) &= \mathbb{E}_\mu\left[\ln p(\mathcal{D}|\mu,\tau) + \ln p(\mu|\tau)\right] + \ln p(\tau) + \text{const} \\ &= (a_0-1)\ln\tau - b_0\tau + \frac{N}{2}\ln\tau \\ &\quad -\frac{\tau}{2}\mathbb{E}_\mu\left[\sum_{n=1}^{N}(x_n-\mu)^2 + \lambda_0(\mu-\mu_0)^2\right] + \text{const} \end{aligned}$$

  - We can see q_τ^* is a gamma distribution $\mathrm{Gam}(\tau|a_N,b_N)$:

$$\begin{aligned} a_N &= a_0 + \frac{N}{2} \\ b_N &= b_0 + \frac{1}{2}\mathbb{E}_\mu\left[\sum_{n=1}^{N}(x_n-\mu)^2 + \lambda_0(\mu-\mu_0)^2\right] \end{aligned}$$

Quick Recap

- We often cannot compute posteriors, and so we need to approximate them, using variational methods.
- In variational Bayes, we'd like to find an approximation within some family that minimizes the KL divergence to the posterior, but we can't directly minimize this
- Therefore, we defined the ELBO, which we can maximize, and this is equivalent to minimizing the KL divergence.



Evidence Lower Bound (ELBO)

$$\ell(\theta; \boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$

Quick Recap

- We defined a family of approximations called "mean field" approximations, in which there are no dependencies between latent variables

$$q(\boldsymbol{z}) = q(z_1, \ldots, z_m) = \prod_{j=1}^{m} q(z_j)$$

- We optimize the ELBO with coordinate ascent updates to iteratively optimize each local variational approximation under mean field assumptions

$$q^*(z_j) \propto \exp\left\{ \mathbb{E}_{q\_j}[\log p(\boldsymbol{x}, \boldsymbol{z})] \right\}$$

Key Takeaways

- KL Divergence $$\mathrm{KL}(q(\boldsymbol{x}) \,||\, p(\boldsymbol{x})) = \sum_{\boldsymbol{x}} q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p(\boldsymbol{x})}$$
- The key equation of EM and VI

Evidence Lower Bound (ELBO)

$$\ell(\theta; \boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$

  - Free energy $F(q, \theta)$

- EM: E-step and M-step optimizing ELBO w.r.t q and θ
- Mean-field VI: optimizing ELBO w.r.t factorized q components

Stochastic VI; Black-box VI

VI with coordinate ascent

Example: Bayesian mixture of Gaussians

- Treat the mean μ_k and cluster proportion π as latent variables

$$\mu_k \sim \mathcal{N}(0, \tau^2) \text{ for } k = 1, \ldots, K$$

$$\pi \sim Dirichlet(\boldsymbol{\alpha})$$

- For each data $i = 1, \ldots, n$

$$z_i \sim \mathrm{Cat}(\pi).$$

$$x_i \sim \mathcal{N}(\mu_{z_i}, \sigma^2).$$

- We have
  - observed variables $x_{1:n}$
  - latent variables $\mu_{1:k}$, π and $z_{1:n}$
  - Hyper-parameters $\{\tau^2, \sigma^2\}$

VI with coordinate ascent

Example: Bayesian mixture of Gaussians

Assume mean-field $q(\mu_{1:K}, \pi, z_{1:n}) = \prod_k q(\mu_k) q(\pi) \prod_i q(z_i)$

- Initialize the global variational distributions $q(\mu_k)$ and $q(\pi)$
- Repeat:
  - For each data example $i \in \{1,2, \ldots, D\}$
    - Update the local variational distribution $q(z_i)$
  - End for
  - Update the global variational distributions $q(\mu_k)$ and $q(\pi)$
- Until ELBO converges

- What if we have millions of data examples? This could be very slow.

Stochastic VI

Example: Bayesian mixture of Gaussians

Assume mean-field $q(\mu_{1:K}, \pi, z_{1:n}) = \prod_k q(\mu_k) q(\pi) \prod_i q(z_i)$

- Initialize the global variational distributions $q(\mu_k)$ and $q(\pi)$
- Repeat:
  - Sample a data example $i \in \{1,2, \dots, D\}$
  - Update the local variational distribution $q(z_i)$
  - Update the global variational distributions $q(\mu_k)$ and $q(\pi)$ with **natural gradient ascent**
- Until ELBO converges

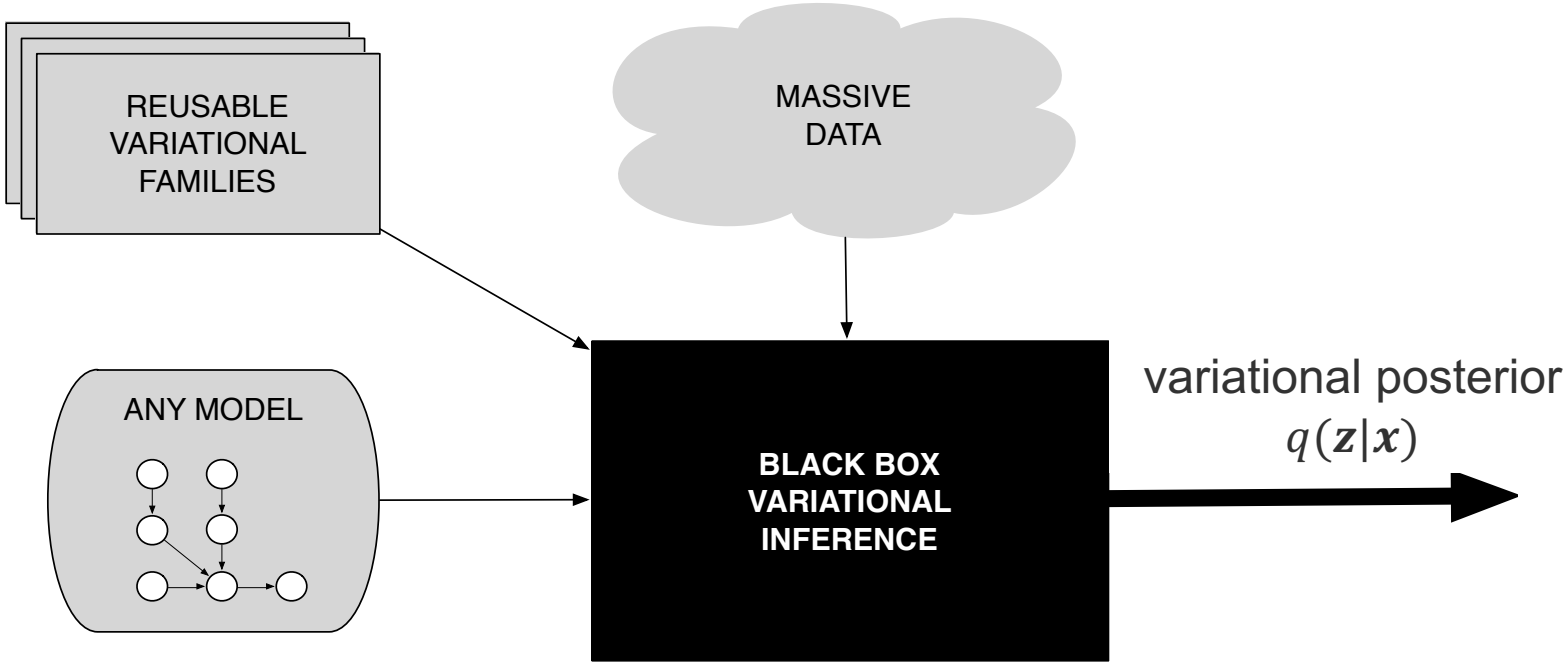
- (Setting natural gradient $= 0$ gives the traditional mean-field update)

[Hoffman et al., Stochastic Variational Inference, 2013]

Black-box Variational Inference (BBVI)

- We have derived variational inference specific for Bayesian Gaussian (mixture) models
- There are innumerable models
- Can we have a solution that does not entail model-specific work?

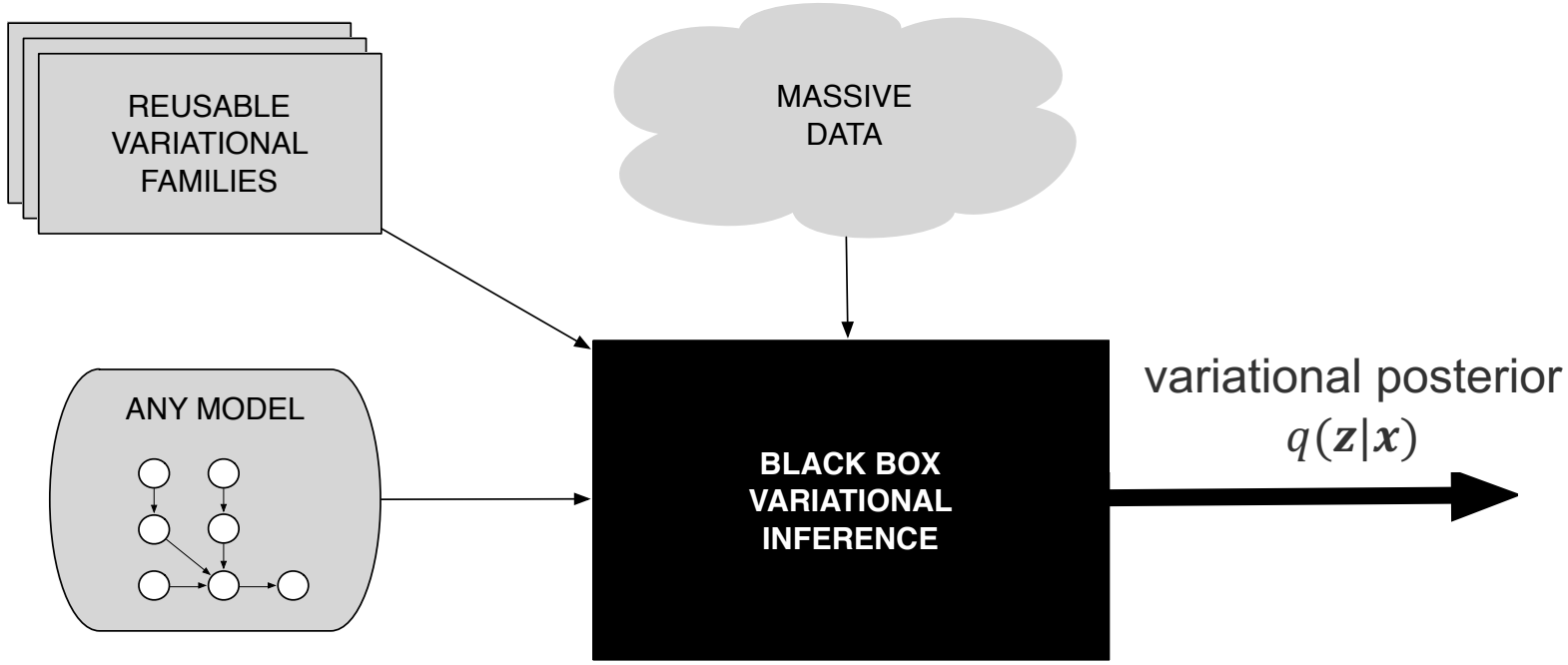
Black-box Variational Inference (BBVI)



- Easily use variational inference with **any model**
- Perform inference with **massive data**
- **No mathematical work** beyond specifying the model

(Courtesy: Blei et al., 2018)

Black-box Variational Inference (BBVI)



- Sample from $q(.)$
- Form noisy gradients (without model-specific computation)
- Use stochastic optimization

(Courtesy: Blei et al., 2018)

Black-box Variational Inference (BBVI)

- Probabilistic model: $\boldsymbol{x}$ -- observed variables, $\boldsymbol{z}$ -- latent variables
- Variational distribution $q_\lambda(\boldsymbol{z}|\boldsymbol{x})$ with parameters λ, e.g.,
  - Gaussian mixture distribution:
    - "A Gaussian mixture model is a universal approximator of densities, in the sense that any smooth density can be approximated with any specific nonzero amount of error by a Gaussian mixture model with enough components." (Deep Learning book, pp.65)
  - Deep neural networks

- ELBO:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log p(\boldsymbol{x}, \boldsymbol{z})] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log q(\boldsymbol{z}|\lambda)]$$

- Want to compute the gradient w.r.t variational parameters λ

[Ranganath et al.,14]

The General Problem: Computing Gradients of Expectations

- When the objective function $\mathcal{L}$ is defined as an expectation of a (differentiable) test function $f_\lambda(\boldsymbol{z})$ w.r.t. a probability distribution $q_\lambda(\boldsymbol{z})$

$$\mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})]$$

- Computing exact gradients w.r.t. the parameters λ is often unfeasible
- Need stochastic gradient estimates
  - The score function estimator (a.k.a log-derivative trick, REINFORCE)
  - The reparameterization trick (a.k.a the pathwise gradient estimator)

Computing Gradients of Expectations w/ score function

- Loss: $\mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})]$
- Log-derivative trick: $\nabla_\lambda q_\lambda = q_\lambda \nabla_\lambda \log q_\lambda$
- Gradient w.r.t. λ:

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z}) \underline{\nabla_\lambda \log q_\lambda(\boldsymbol{z})} + \nabla_\lambda f_\lambda(\boldsymbol{z})]$$

  - score function: the gradient of the log of a probability distribution
- Compute noisy unbiased gradients with Monte Carlo samples from q_λ

$$\nabla_\lambda \mathcal{L} \approx \frac{1}{S} \sum_{\boldsymbol{s=1}}^{\boldsymbol{S}} f_\lambda(\boldsymbol{z}_s) \nabla_\lambda \log q_\lambda(\boldsymbol{z_s}) + \nabla_\lambda f_\lambda(\boldsymbol{z_s}) \qquad \text{where } \boldsymbol{z}_s \sim q_\lambda(\boldsymbol{z})$$

- Pros: generally applicable to any distribution $q(z|\lambda)$
- Cons: empirically has high variance → slow convergence
  - To reduce variance: Rao-Blackwellization, control variates, importance sampling, ...

Computing Gradients of Expectations w/ reparametrization trick

- Loss: $\mathcal{L} = \mathbb{E}_{q_{\boldsymbol{\lambda}}(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})]$
- Assume that we can express the distribution $q_{\boldsymbol{\lambda}}(\boldsymbol{z})$ with a transformation

$$\begin{aligned} \epsilon &\sim s(\epsilon) \\ z &= t(\epsilon, \lambda) \end{aligned} \quad \Leftrightarrow \quad z \sim q(z|\lambda)$$

  - E.g.,

$$\begin{aligned} \epsilon &\sim Normal(0,1) \\ z &= \epsilon\sigma + \mu \end{aligned} \quad \Leftrightarrow \quad z \sim Normal(\mu, \sigma^2)$$

- Reparameterization gradient

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{\epsilon}\sim s(\boldsymbol{\epsilon})}[f_\lambda(\boldsymbol{z}(\boldsymbol{\epsilon}, \lambda))]$$

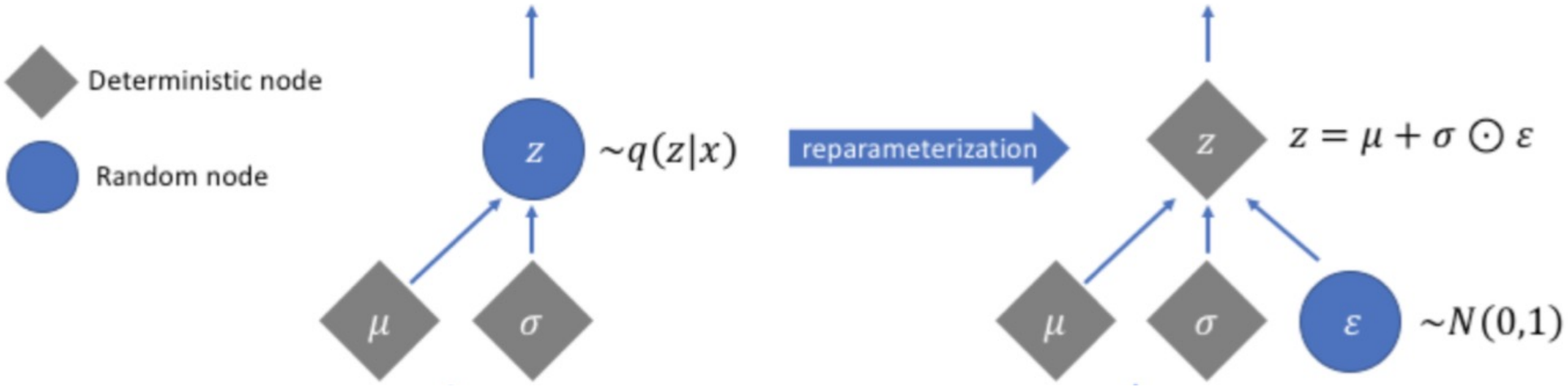
$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{\boldsymbol{\epsilon}\sim \boldsymbol{s}(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{z}} f_\lambda(\boldsymbol{z})\, \nabla_\lambda t(\epsilon, \lambda)]$$

- Pros: empirically, lower variance of the gradient estimate
- Cons: Not all distributions can be reparameterized

Reparameterization trick

- Reparametrizing Gaussian distribution

$$\epsilon \sim Normal(0,1) \\ z = \epsilon\sigma + \mu \Leftrightarrow \quad z \sim Normal(\mu, \sigma^2)$$



[Courtesy: Tansey, 2016]

Reparameterization trick

- Reparametrizing Gaussian distribution

$$\epsilon \sim Normal(0,1) \\ z = \epsilon\sigma + \mu \Leftrightarrow \quad z \sim Normal(\mu, \sigma^2)$$

- Other reparameterizable distributions: $\epsilon \sim Uniform(\epsilon) \\ z = F^{-1}(\epsilon) \Leftrightarrow \quad z \sim q(z)$
  - Tractable inverse CDF F^{-1}:
    - Exponential, Cauchy, Logistic, Rayleigh, Pareto, Weibull, Reciprocal, Gompertz, Gumbel, Erlang
  - Location-scale:
    - Laplace, Elliptical, Student's t, Logistic, Uniform, Triangular, Gaussian
  - Composition:
    - Log-Normal (exponentiated normal) Gamma (sum of exponentials) Dirichlet (sum of Gammas) Beta, Chi-Squared, F

[Courtesy: Tansey, 2016]

Computing Gradients of Expectations: Summary

- Loss: $\mathcal{L} = \mathbb{E}_{q_{\boldsymbol{\lambda}}(\boldsymbol{z})}[f_{\lambda}(\boldsymbol{z})]$
- Score gradient

$$\nabla_{\lambda}\mathcal{L} = \mathbb{E}_{q_{\boldsymbol{\lambda}}(\boldsymbol{z})}[f_{\lambda}(\boldsymbol{z})\nabla_{\lambda}\log q_{\lambda}(\boldsymbol{z}) + \nabla_{\lambda}f_{\lambda}(\boldsymbol{z})]$$

  - Pros: generally applicable to any distribution $q(z|\lambda)$
  - Cons: empirically has high variance → slow convergence

- Reparameterization gradient

$$\nabla_{\lambda}\mathcal{L} = \mathbb{E}_{\boldsymbol{\epsilon}\sim\boldsymbol{s}(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{z}}f_{\lambda}(\boldsymbol{z})\ \nabla_{\lambda}t(\epsilon,\lambda)]$$

  - Pros: empirically, lower variance of the gradient estimate
  - Cons: Not all distributions can be reparameterized

Recall: Black-box Variational Inference (BBVI)

- Probabilistic model: $\boldsymbol{x}$ -- observed variables, $\boldsymbol{z}$ -- latent variables
- Variational distribution $q_\lambda(\boldsymbol{z}|\boldsymbol{x})$ with parameters λ, e.g.,
  - Gaussian mixture distribution:
    - "A Gaussian mixture model is a universal approximator of densities, in the sense that any smooth density can be approximated with any specific nonzero amount of error by a Gaussian mixture model with enough components." (Deep Learning book, pp.65)
  - Deep neural networks

$$\mathcal{L}(\lambda) \triangleq \mathrm{E}_{q_\lambda(z)}[\log p(x,z) - \log q(z)]$$

- ELBO:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log p(\boldsymbol{x},\boldsymbol{z})] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log q(\boldsymbol{z}|\lambda)]$$

- Want to compute the gradient w.r.t variational parameters λ

[Ranganath et al.,14]

BBVI with the score gradient

- ELBO:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log p(\boldsymbol{x}, \boldsymbol{z})] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log q(\boldsymbol{z}|\lambda)]$$

- Gradient w.r.t. λ (using the log-derivative trick)

$$\nabla_\lambda \mathcal{L} = \mathrm{E}_q[\nabla_\lambda \log q(z|\lambda)(\log p(x, z) - \log q(z|\lambda))]$$

- Compute noisy unbiased gradients of the ELBO with Monte Carlo samples from the variational distribution

$$\nabla_\lambda \mathcal{L} \approx \frac{1}{S} \sum_{s=1}^{S} \nabla_\lambda \log q(z_s|\lambda)(\log p(x, z_s) - \log q(z_s|\lambda)),$$

$$\text{where } z_s \sim q(z|\lambda).$$

[Ranganath et al.,14]

BBVI with the reparameterization gradient

- ELBO:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log p(\boldsymbol{x}, \boldsymbol{z})] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log q(\boldsymbol{z}|\lambda)]$$

- Gradient w.r.t. λ

$$\begin{aligned} \epsilon &\sim s(\epsilon) \\ z &= t(\epsilon, \lambda) \end{aligned} \quad \Leftrightarrow \quad z \sim q(z|\lambda)$$

$$\nabla_\lambda \mathcal{L} = \mathrm{E}_{\epsilon \sim s(\epsilon)}[\nabla_z [\log p(x, z) - \log q(z)] \nabla_\lambda t(\epsilon, \lambda)]$$

Variational Autoencoders (VAEs)

Variational Auto-Encoders (VAEs)

VAEs are a combination of the following ideas:

- Variational Inference
  - ELBO
- Variational distribution parametrized as neural networks
- Reparameterization trick

[Courtesy: Dhruv, CS 4803]

Variational Auto-Encoders (VAEs)

- Model $p_\theta(\boldsymbol{x}, \boldsymbol{z}) = p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$
  - $p_\theta(\boldsymbol{x}|\boldsymbol{z})$: a.k.a., generative model, generator, (probabilistic) decoder, …
  - $p(\boldsymbol{z})$: prior, e.g., Gaussian
- Assume variational distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$
  - E.g., a Gaussian distribution parameterized as **deep neural networks**
  - a.k.a, recognition model, inference network, (probabilistic) encoder, …
- ELBO:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) = \mathrm{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}, \boldsymbol{z})] - \mathrm{H}(q_\phi(\boldsymbol{z}|\boldsymbol{x}))$$
$$= \mathrm{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}))$$

Reconstruction

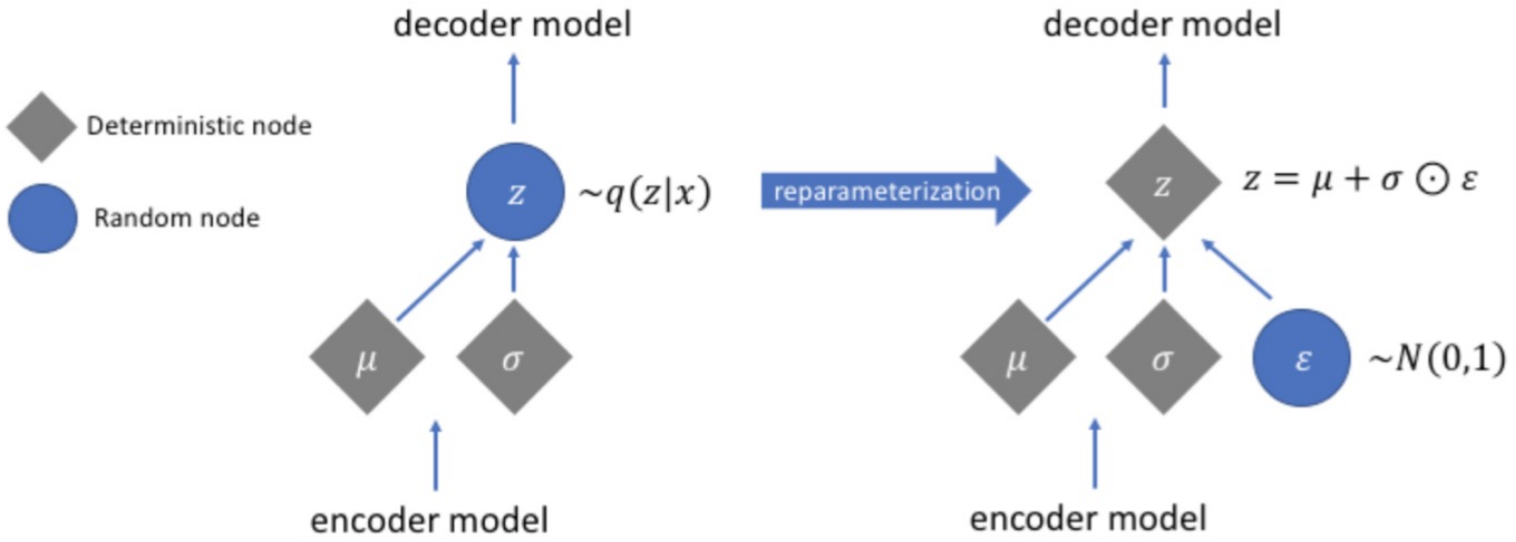
Divergence from prior
(KL divergence between two Guassians has an analytic form)

Variational Auto-Encoders (VAEs)

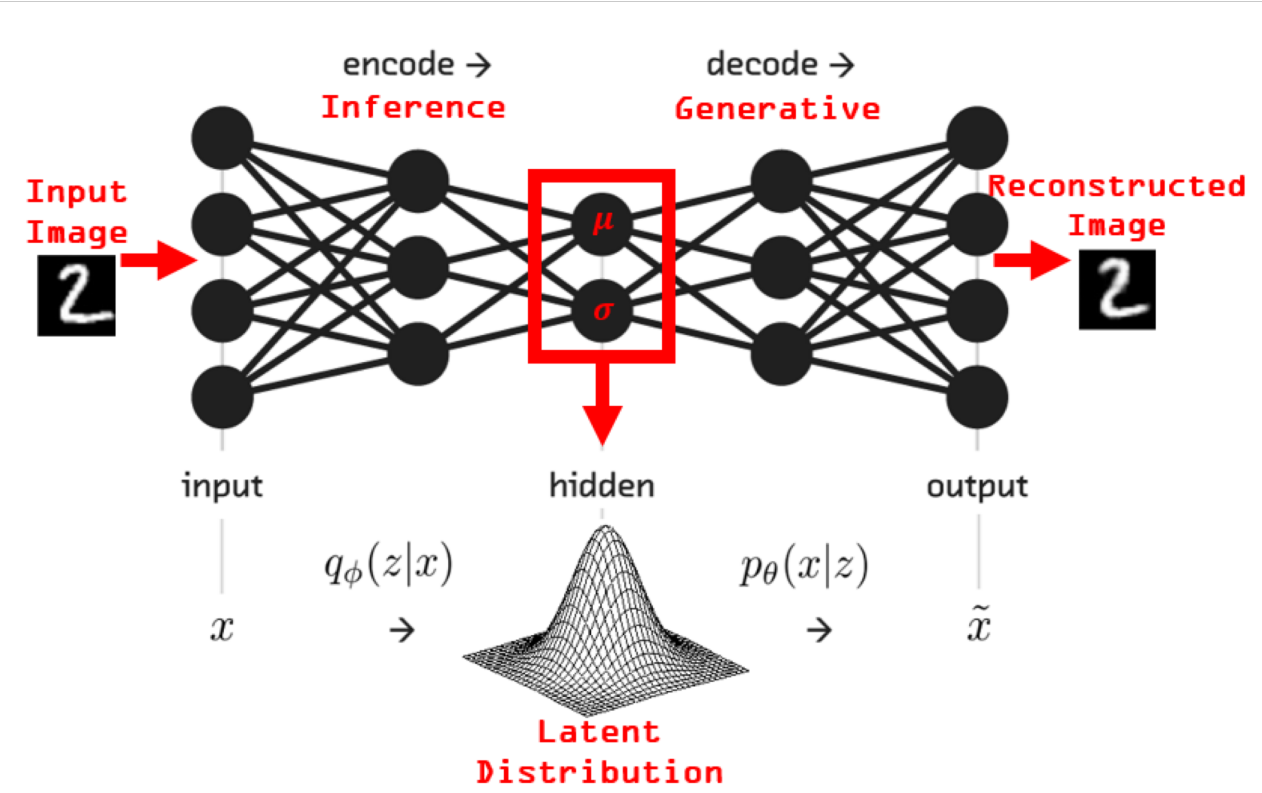
- ELBO:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) = \mathrm{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}, \boldsymbol{z})] - \mathrm{H}(q_\phi(\boldsymbol{z}|\boldsymbol{x}))$$
$$= \mathrm{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}))$$

- Reparameterization:
  - $[\boldsymbol{\mu}; \boldsymbol{\sigma}] = f_\phi(\boldsymbol{x})$ (a neural network)
  - $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{1})$

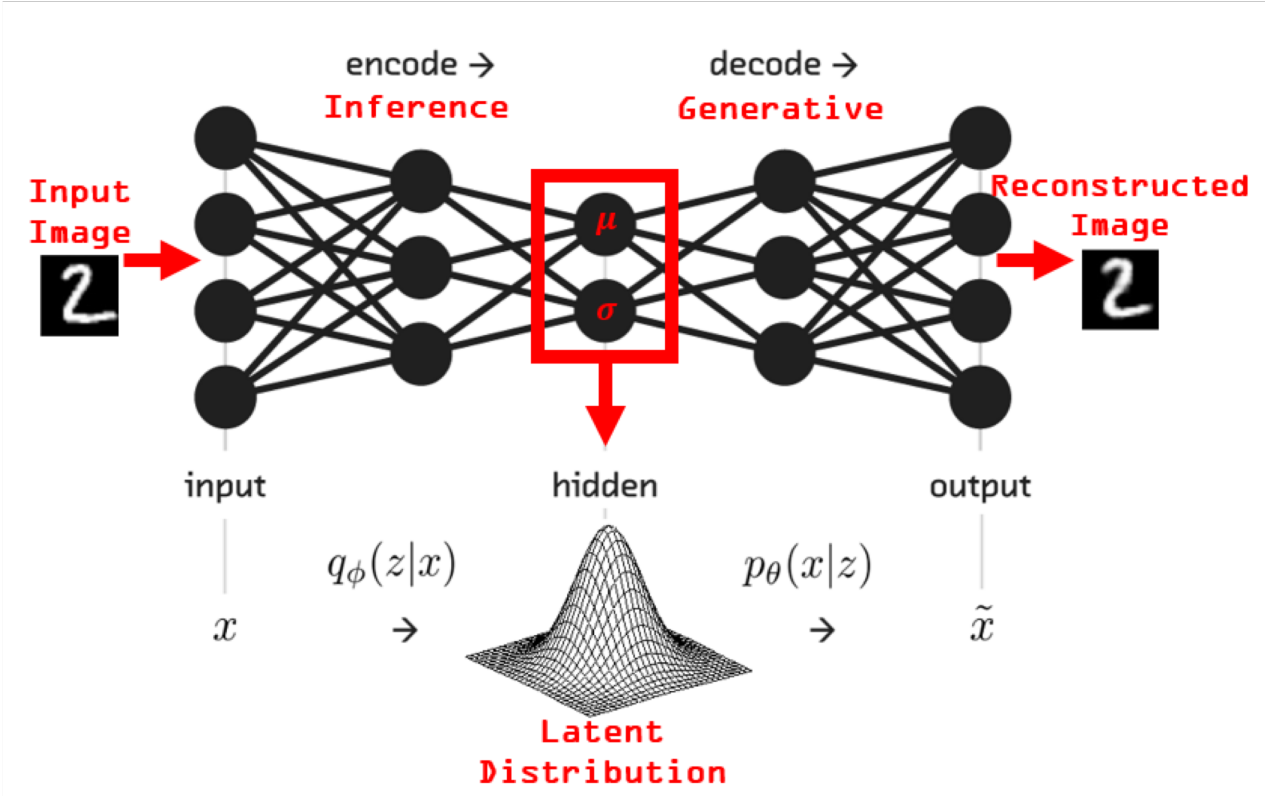


Example: VAEs for images



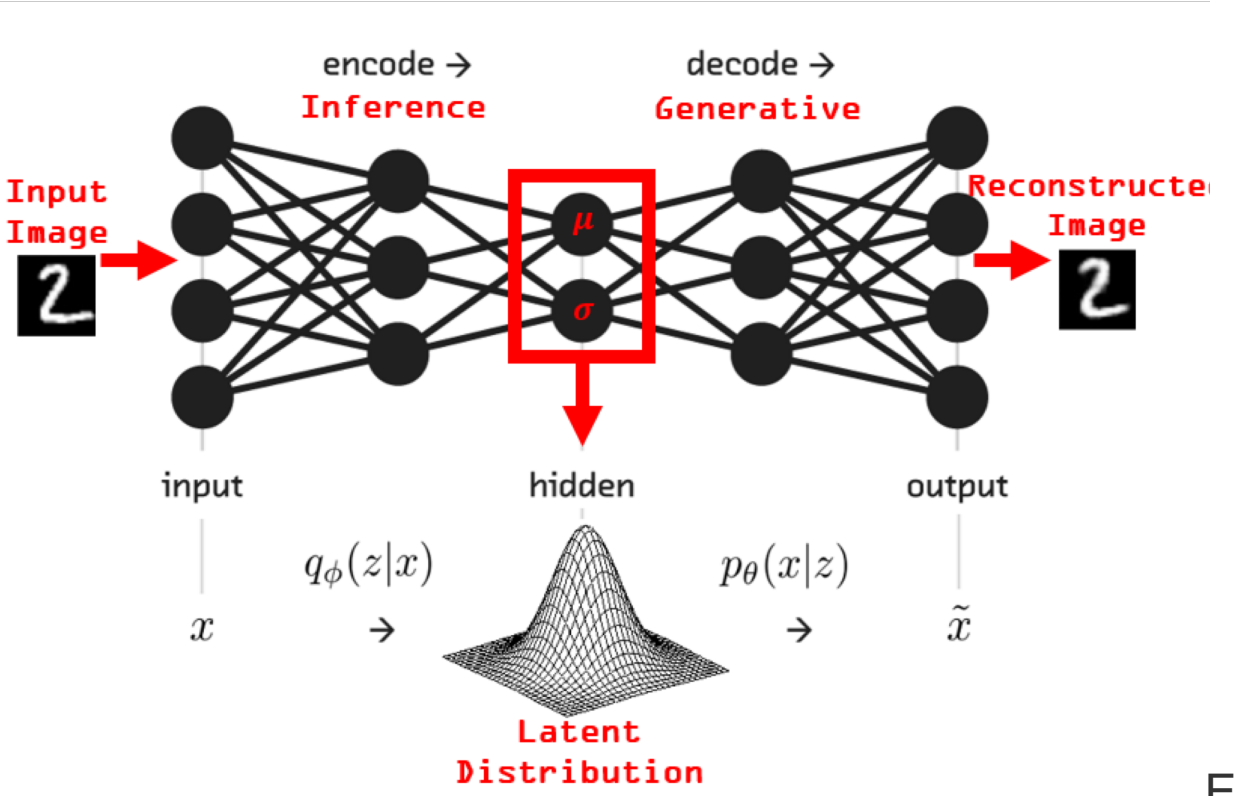
[https://www.kaggle.com/rvislaywade/visualizing-mnist-using-a-variational-autoencoder]

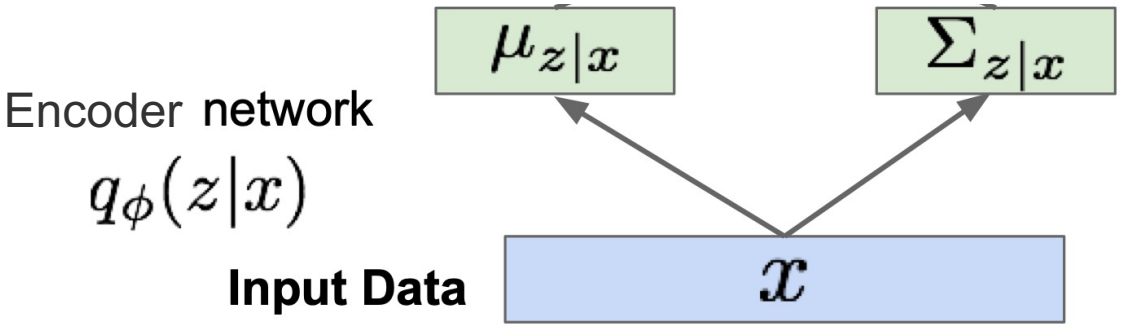
Example: VAEs for images



[Courtesy: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n]

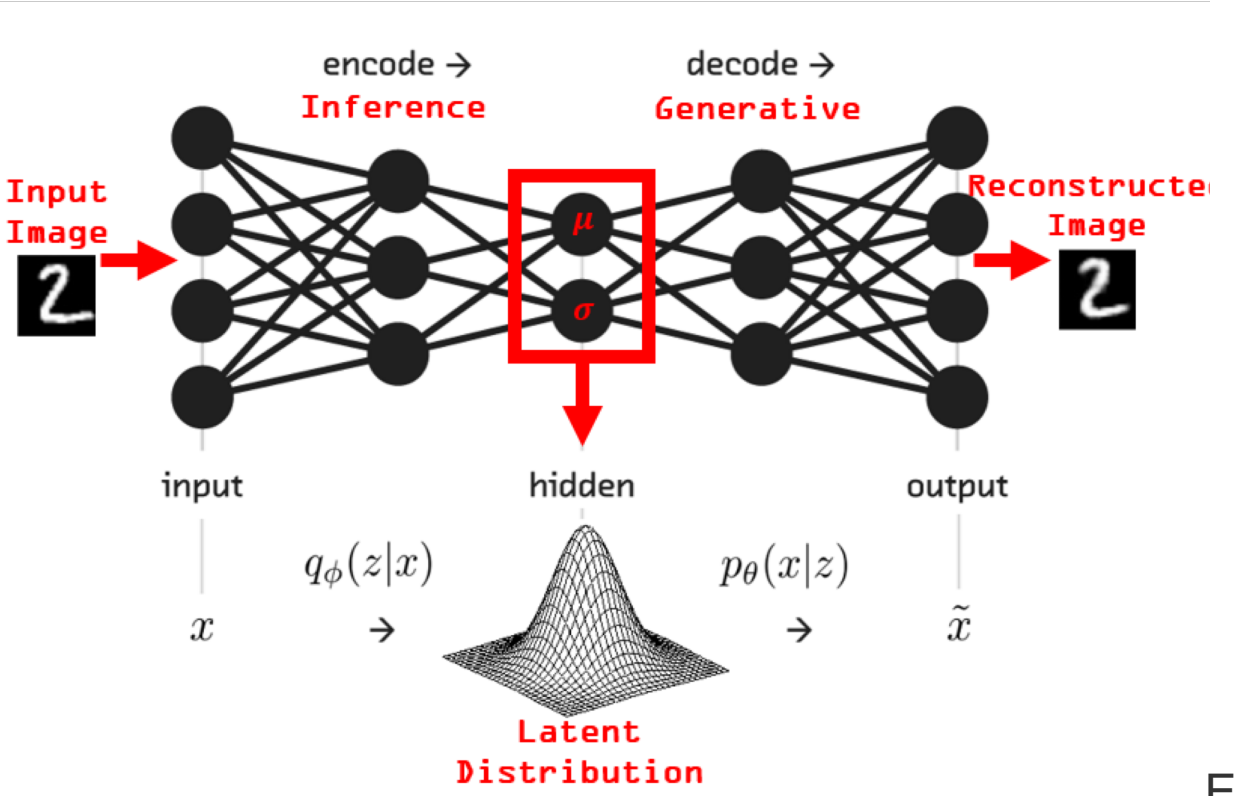
Example: VAEs for images

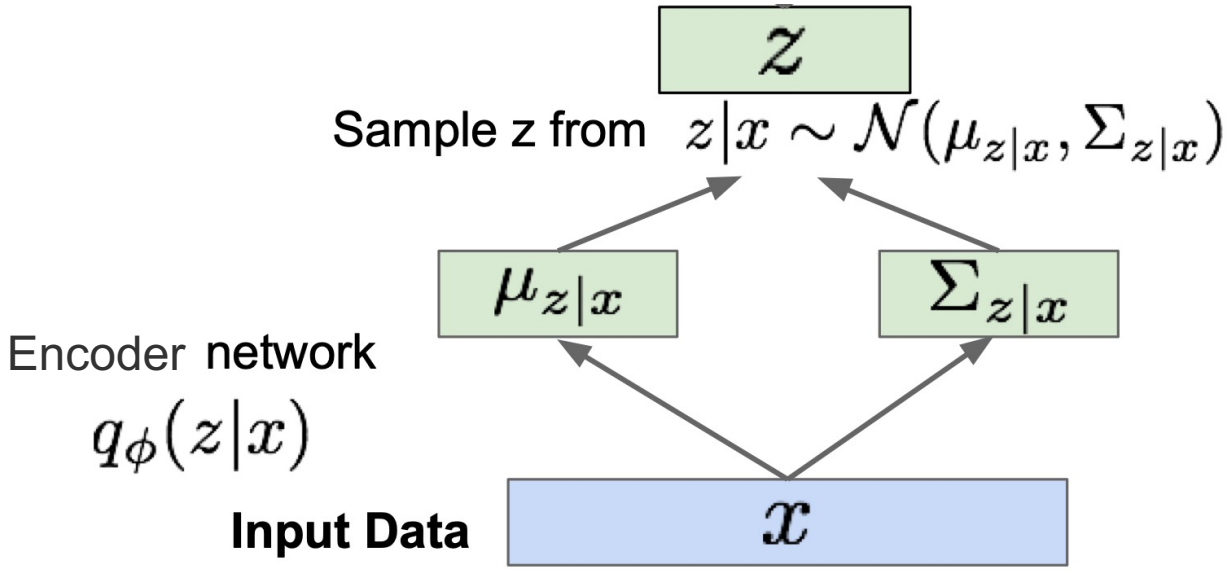




[Courtesy: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n]

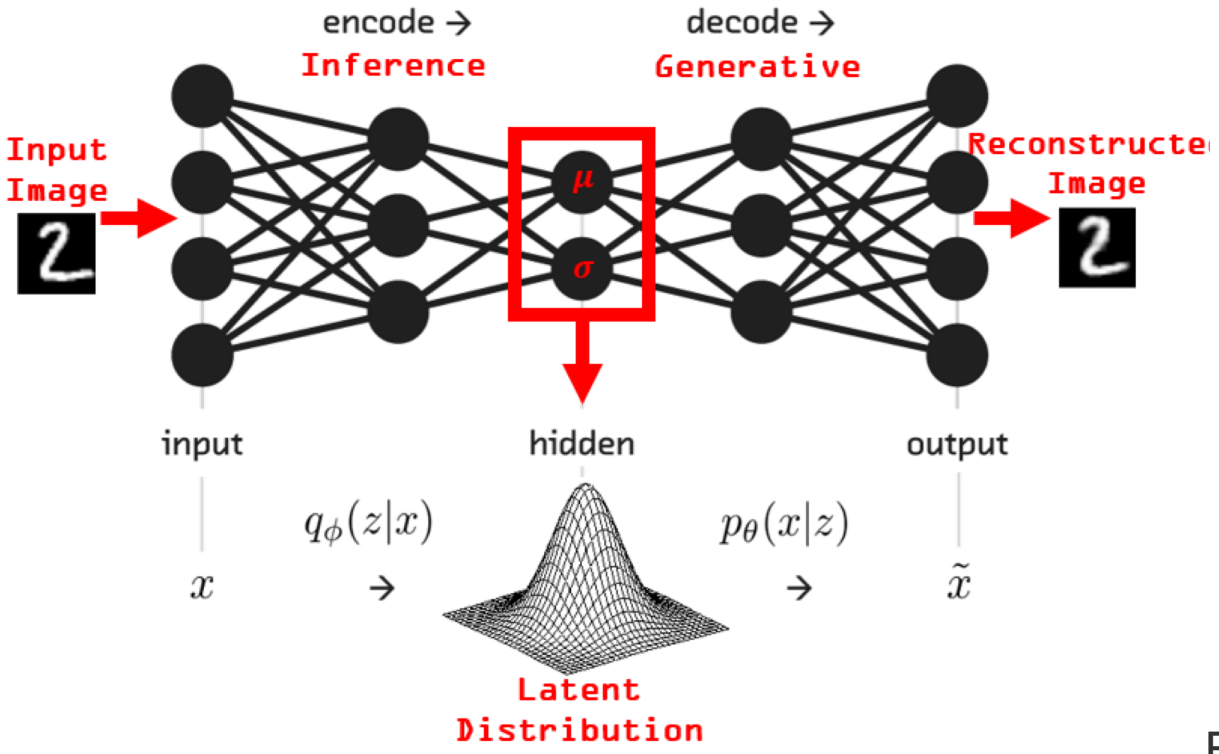
Example: VAEs for images

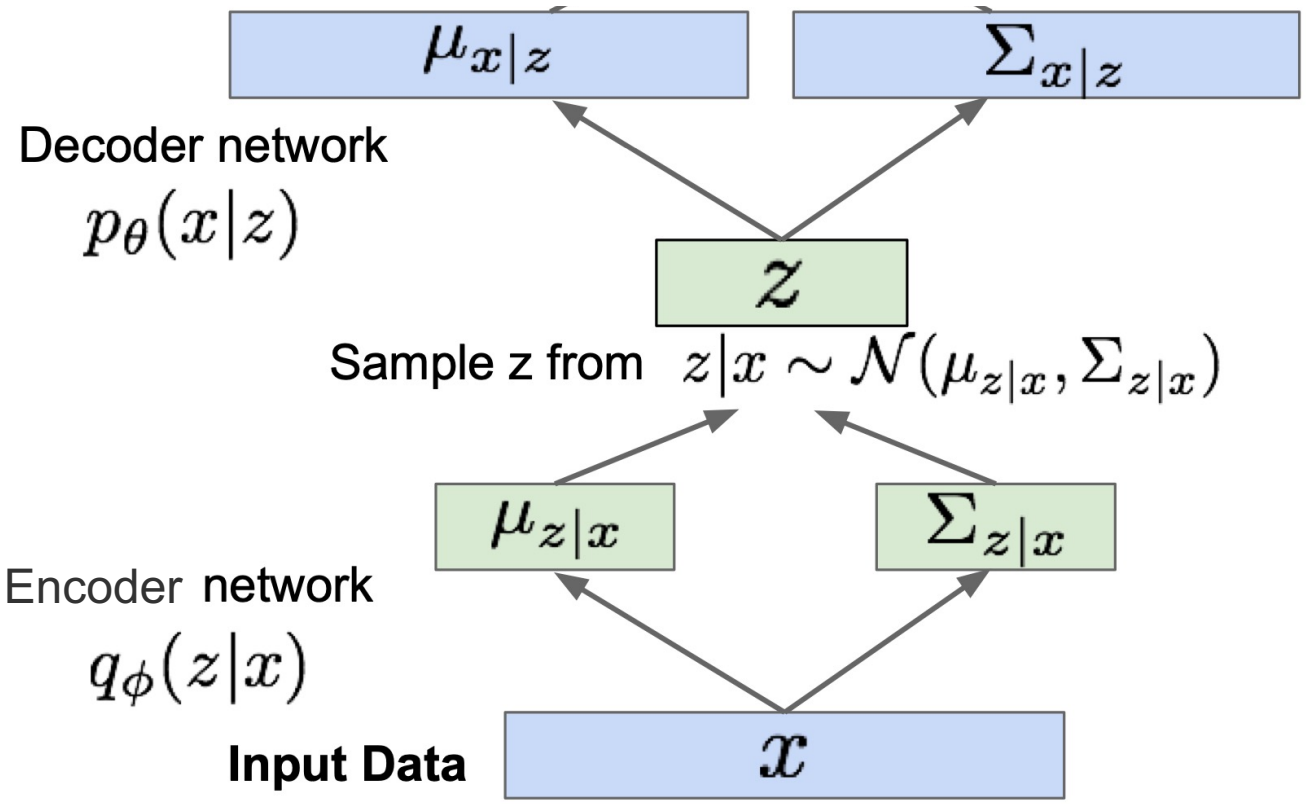




[Courtesy: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n]

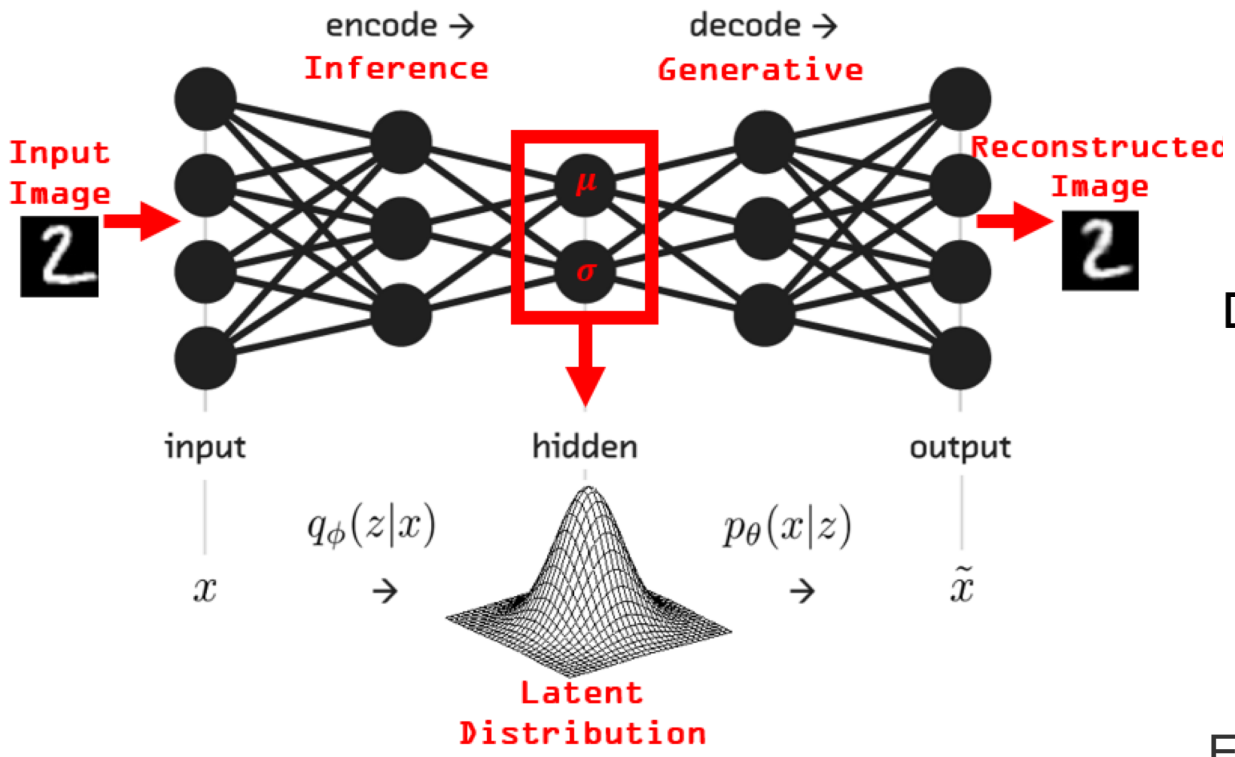
Example: VAEs for images

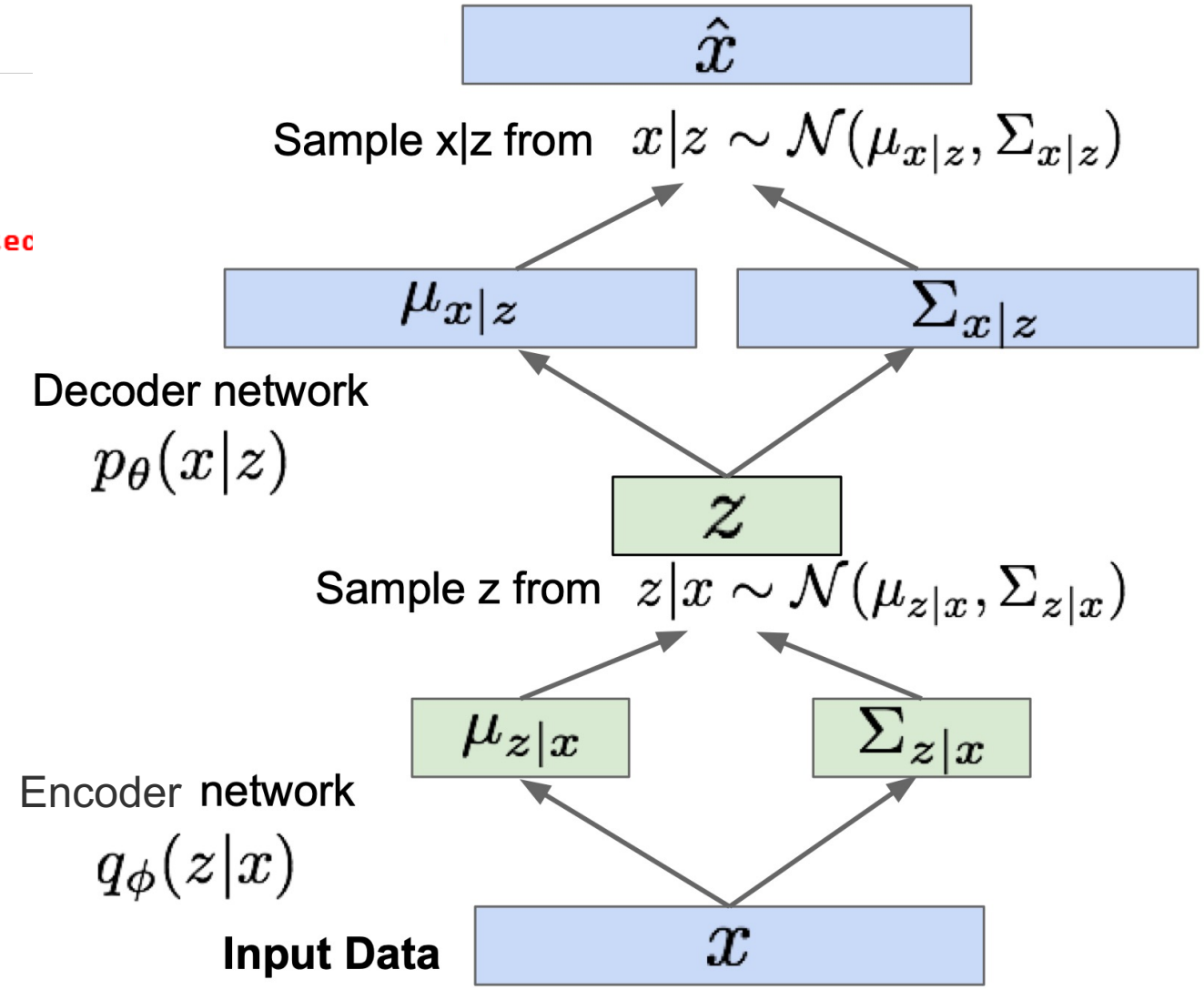




[Courtesy: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n]

Example: VAEs for images



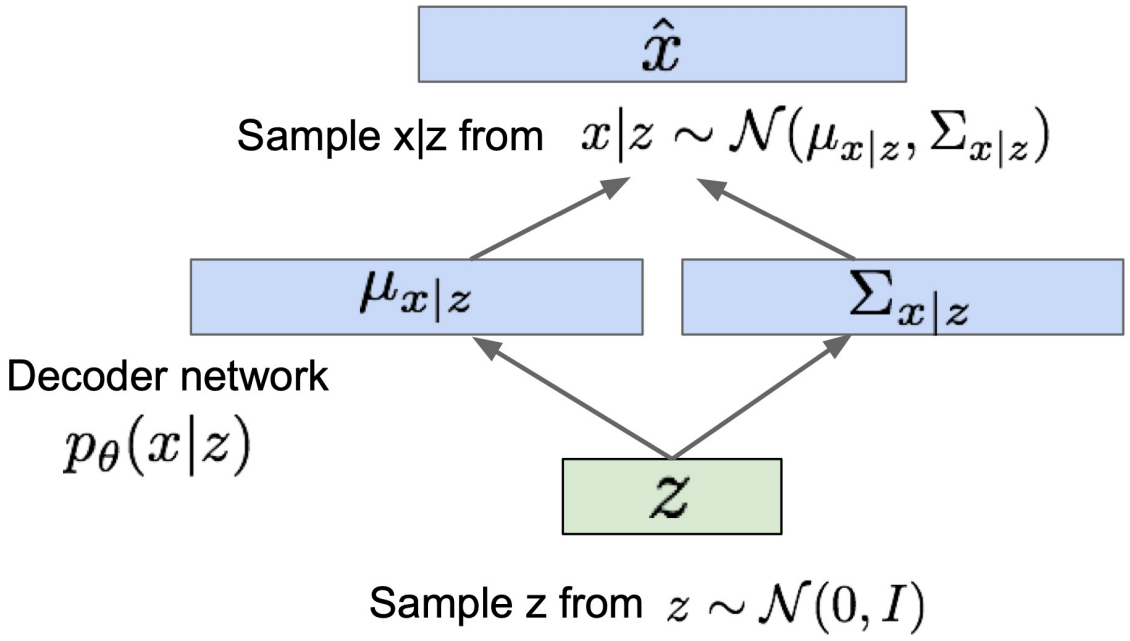


[Courtesy: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n]

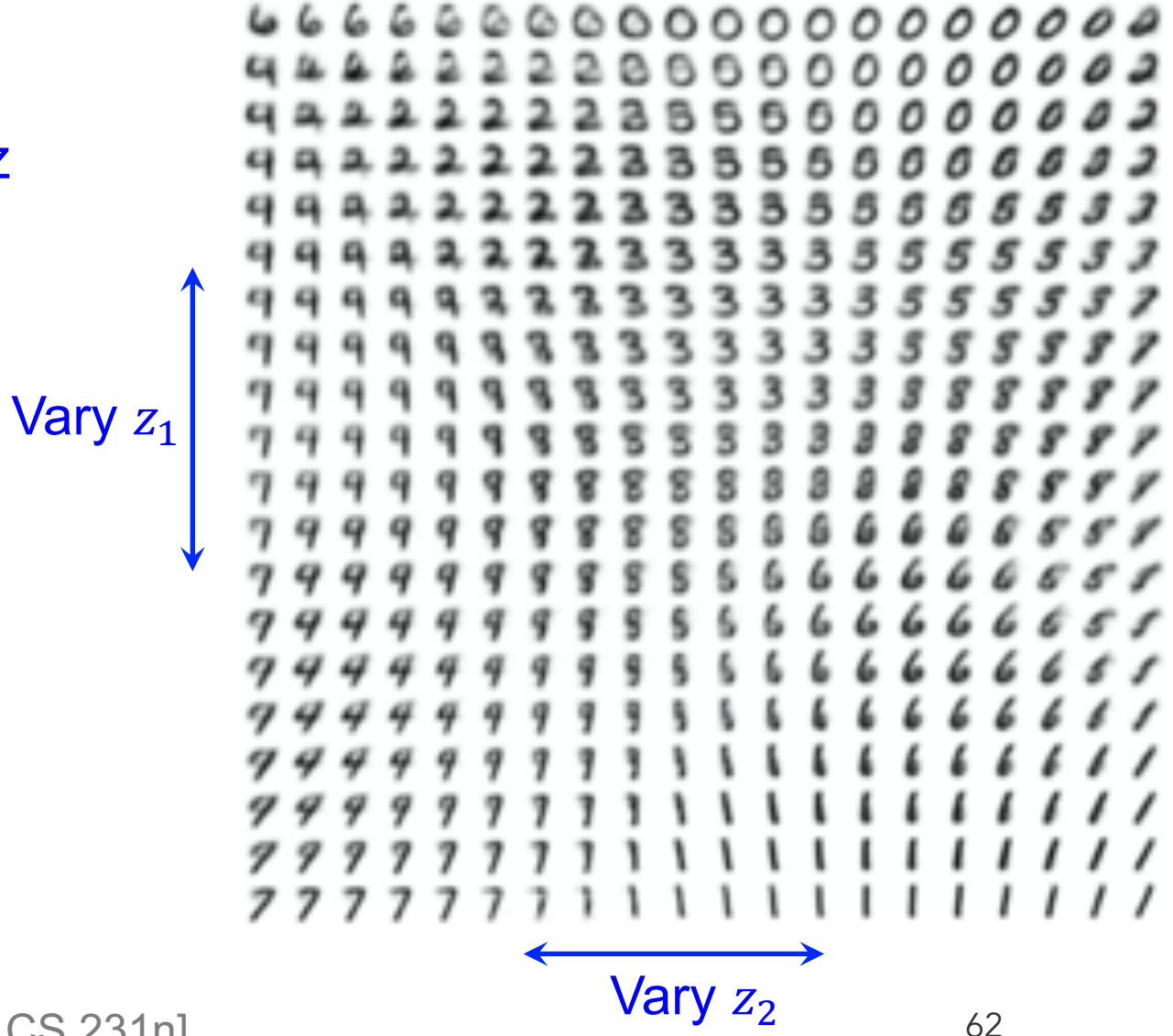
Example: VAEs for images

Generating samples:

- Use decoder network. Now sample z from prior!



Data manifold for 2-d **z**

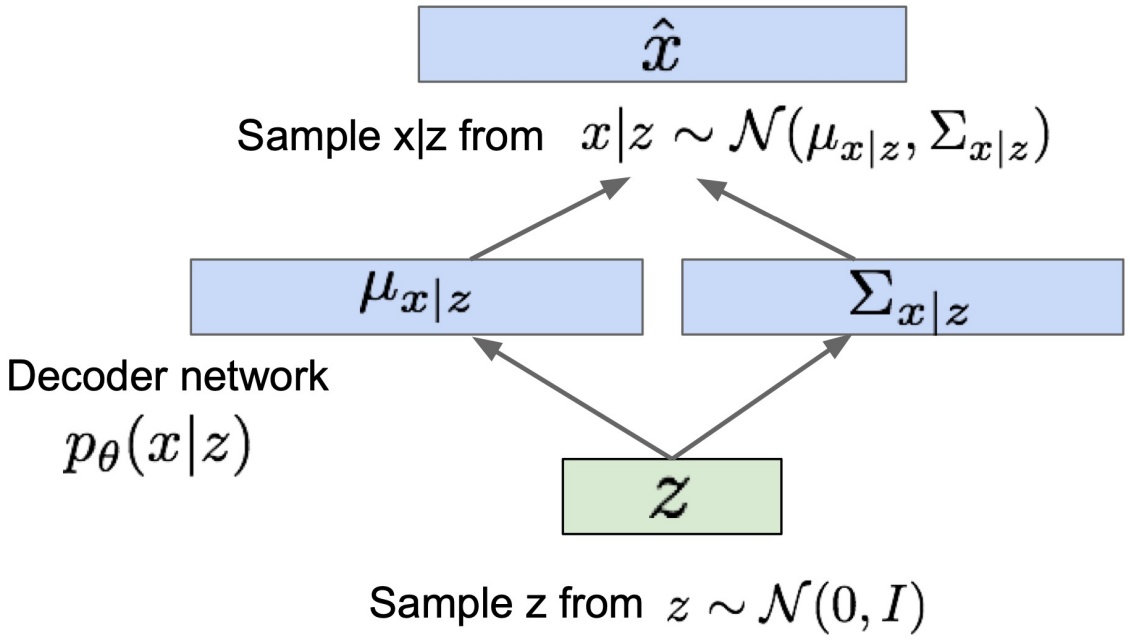


[Courtesy: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n]

Example: VAEs for images

Generating samples:

- Use decoder network. Now sample z from prior!



Data manifold for 2-d **z**

[Courtesy: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n]

Example: VAEs for text

- Latent code interpolation and sentences generation from VAEs [Bowman et al., 2015].

“ i want to talk to you . ”

“i want to be with you . ”

“i do n’t want to be with you . ”

i do n’t want to be with you .

she did n’t want to be with him .

Variational Auto-Encoders (VAEs)

Algorithm 1 Minibatch version of the Auto-Encoding VB (AEVB) algorithm. Either of the two SGVB estimators in section 2.3 can be used. We use settings $M = 100$ and $L = 1$ in experiments.

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialize parameters
repeat
  $\mathbf{X}^M \leftarrow$ Random minibatch of M datapoints (drawn from full dataset)
  $\boldsymbol{\epsilon} \leftarrow$ Random samples from noise distribution $p(\boldsymbol{\epsilon})$
  $\mathbf{g} \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \widetilde{\mathcal{L}}^M(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}^M, \boldsymbol{\epsilon})$ (Gradients of minibatch estimator (8))
  $\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Update parameters using gradients $\mathbf{g}$ (e.g. SGD or Adagrad [DHS10])
until convergence of parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$
return $\boldsymbol{\theta}, \boldsymbol{\phi}$

[Kingma & Welling, 2014]

Note: Amortized Variational Inference

- Variational distribution as an inference model $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ with parameters $\boldsymbol{\phi}$ (which was traditionally factored over samples)
- Amortize the cost of inference by learning a **single** data-dependent inference model
- The trained inference model can be used for quick inference on new data

Variational Auto-encoders: Summary

- A combination of the following ideas:
  - Variational Inference: ELBO
  - Variational distribution parametrized as neural networks
  - Reparameterization trick

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) = [\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}))$$

Reconstruction

Divergence from prior

(Razavi et al., 2019)

- Pros:
  - Principled approach to generative models
  - Allows inference of $q(z|x)$, can be useful feature representation for other tasks
- Cons:
  - Samples blurrier and lower quality compared to GANs
  - Tend to collapse on text data

Key Takeaways

- Stochastic VI
- Computing Gradients of Expectations $\mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})]$
  - Score gradient

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z}) \nabla_\lambda \log q_\theta(\boldsymbol{z}) + \nabla_\lambda f_\lambda(\boldsymbol{z})]$$

  - Reparameterization gradient

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{\boldsymbol{\epsilon} \sim \boldsymbol{s}(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{z}} f_\lambda(\boldsymbol{z})\, \nabla_\lambda t(\epsilon, \lambda) + \nabla_{\boldsymbol{\lambda}} f_\lambda(\boldsymbol{z})]$$

- Black-box VI
- Variational autoencoders (VAEs)

Questions?