EECE 571F: Deep Learning with Structures

Lecture 10: Discrete Latent Variable Models & Amortized Inference & Learning

Renjie Liao

University of British Columbia

Winter, Term 1, 2022

Course Scope

- Brief Intro to Deep Learning

- Geometric Deep Learning

  - Deep Learning Models for Sets and Sequences: Deep Sets & Transformers
  - Deep Learning Models for Graphs: Graph Convolution & Message Passing GNNs
  - Expressiveness & Generalizations of GNNs
  - Unsupervised/Self-supervised Graph Representation Learning

- Probabilistic Deep Learning

  - Deep Generative Models:
    Auto-regressive models, GANs, VAEs, Diffusion/Score based models
  - Discrete/Hybrid Latent Variable Models: RBMs, Latent Graph Models
  - Stochastic Gradient Estimation

Course Scope

- Brief Intro to Deep Learning

- Geometric Deep Learning

  - Deep Learning Models for Sets and Sequences: Deep Sets & Transformers
  - Deep Learning Models for Graphs: Graph Convolution & Message Passing GNNs
  - Expressiveness & Generalizations of GNNs
  - Unsupervised/Self-supervised Graph Representation Learning

- Probabilistic Deep Learning

  - Deep Generative Models:
    Auto-regressive models, GANs, VAEs, Diffusion/Score based models
  - **Discrete/Hybrid Latent Variable Models: RBMs, Latent Graph Models**
  - Stochastic Gradient Estimation

Discrete Latent Variable Models

They are fundamental models in deep learning, e.g., Restricted Boltzmann Machines (RBMs) [1,2]

Discrete Latent Variable Models

They are fundamental models in deep learning, e.g., Restricted Boltzmann Machines (RBMs) [1,2]

Binary visible units (observable variables) x, binary hidden units (latent variables) h

Discrete Latent Variable Models

They are fundamental models in deep learning, e.g., Restricted Boltzmann Machines (RBMs) [1,2]

Binary visible units (observable variables) x, binary hidden units (latent variables) h

Energy function $E_\theta(x, h) = -a^\top x - b^\top h - x^\top W h$

Discrete Latent Variable Models

They are fundamental models in deep learning, e.g., Restricted Boltzmann Machines (RBMs) [1,2]

Binary visible units (observable variables) x, binary hidden units (latent variables) h

Energy function $E_\theta(x, h) = -a^\top x - b^\top h - x^\top W h$

Probability distribution

$$p_\theta(x, h) = \frac{1}{Z} \exp\left(-E_\theta(x, h)\right) \qquad Z = \int \int \exp\left(-E_\theta(x, h)\right) dx dh$$

Partition function / Normalization constant

Discrete Latent Variable Models

They are fundamental models in deep learning, e.g., Restricted Boltzmann Machines (RBMs) [1,2]

Binary visible units (observable variables) x, binary hidden units (latent variables) h

Energy function $E_\theta(x,h) = -a^\top x - b^\top h - x^\top W h$

Probability distribution

$$p_\theta(x,h) = \frac{1}{Z}\exp\left(-E_\theta(x,h)\right) \qquad Z = \int\int \exp\left(-E_\theta(x,h)\right) dxdh$$

Energy-Based Models (EBMs)

Discrete Latent Variable Models

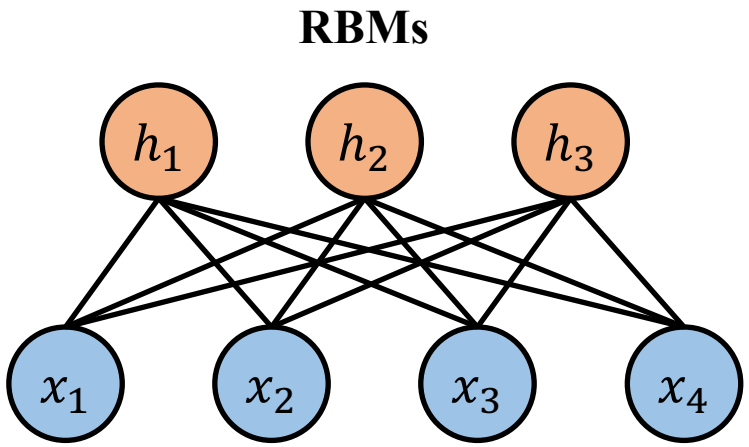
They are fundamental models in deep learning, e.g., Restricted Boltzmann Machines (RBMs) [1,2]

Binary visible units (observable variables) x, binary hidden units (latent variables) h

Energy function $E_\theta(x, h) = -a^\top x - b^\top h - x^\top W h$

Probability distribution

$$p_\theta(x, h) = \frac{1}{Z} \exp\left(-E_\theta(x, h)\right)$$



Discrete Latent Variable Models

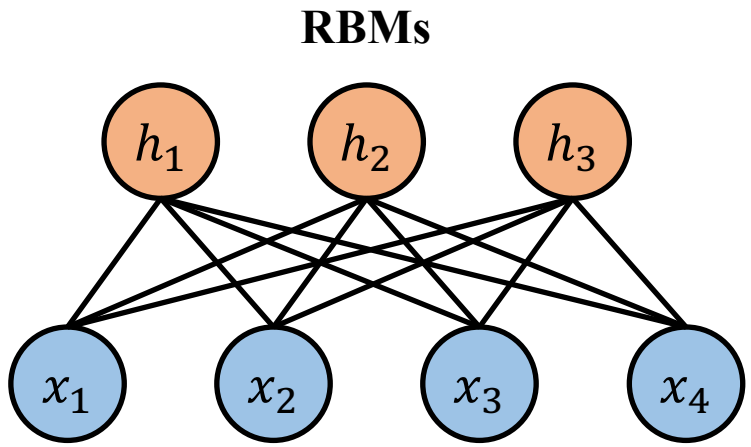
They are fundamental models in deep learning, e.g., Restricted Boltzmann Machines (RBMs) [1,2]

Binary visible units (observable variables) x, binary hidden units (latent variables) h

Energy function $E_\theta(x,h) = -a^\top x - b^\top h - x^\top W h$

Probability distribution

$$p_\theta(x,h) = \frac{1}{Z}\exp\left(-E_\theta(x,h)\right)$$



Bipartite graph structure implies conditional independence:

$$p(h|x) = \prod_j p(h_j|x)$$

$$p(x|h) = \prod_i p(x_i|h)$$

Discrete Latent Variable Models

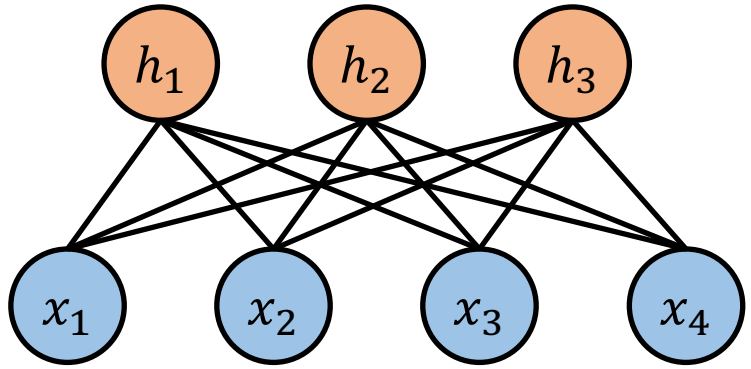
Bipartite graph structure implies conditional independence:

$$p(h|x) = \prod_j p(h_j|x)$$

$$p(x|h) = \prod_i p(x_i|h)$$

Why?

RBMs



Discrete Latent Variable Models

Bipartite graph structure implies conditional independence:

$$p(h|x) = \prod_j p(h_j|x)$$

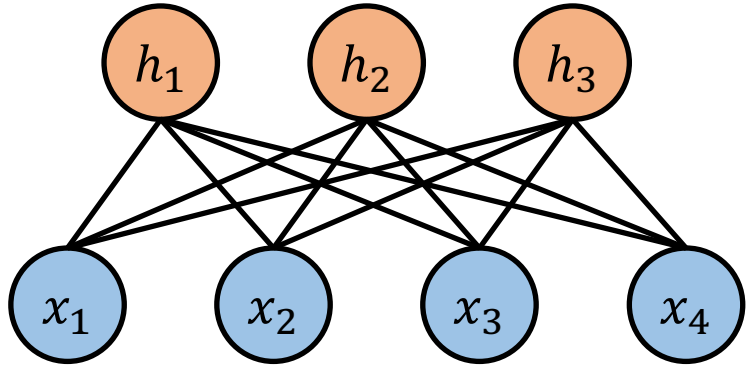
$$p(x|h) = \prod_i p(x_i|h)$$

Why?

Intuition:

- Observed visible units block the paths among hidden units
- Change of one hidden unit would not affect others

RBMs



Discrete Latent Variable Models

Bipartite graph structure implies conditional independence:

$$p(h|x) = \prod_j p(h_j|x)$$

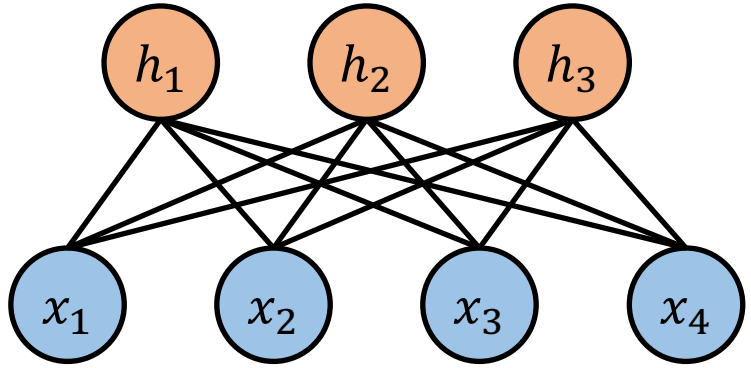
$$p(x|h) = \prod_i p(x_i|h)$$

Why?

Intuition:

- Observed visible units block the paths among hidden units
- Change of one hidden unit would not affect others

RBMs



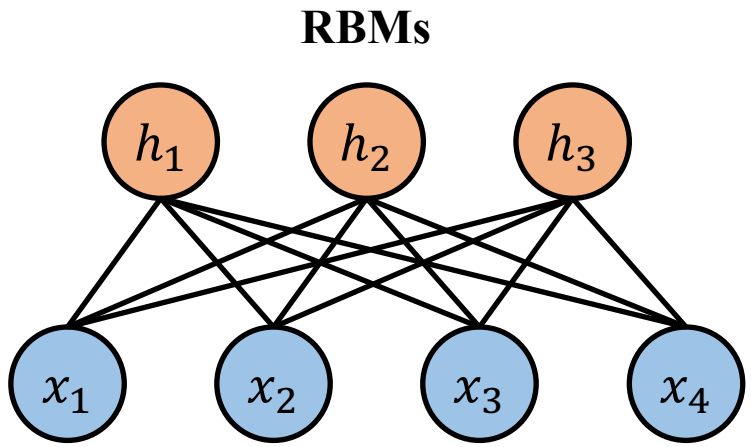
Formally:

$$E_\theta(x, h) = -a^\top x - b^\top h - x^\top W h$$

$$p(x|h = \tilde{h}) \propto \exp\left(-E_\theta(x, h = \tilde{h})\right) \propto \exp\left(-\tilde{a}^\top x\right) = \prod_i \exp\left(-\tilde{a}_i x_i\right)$$

Discrete Latent Variable Models

Inference: Computing Marginals $p(h)$ & Maximum A Posterior (MAP) $\arg\max_h p(h|x)$

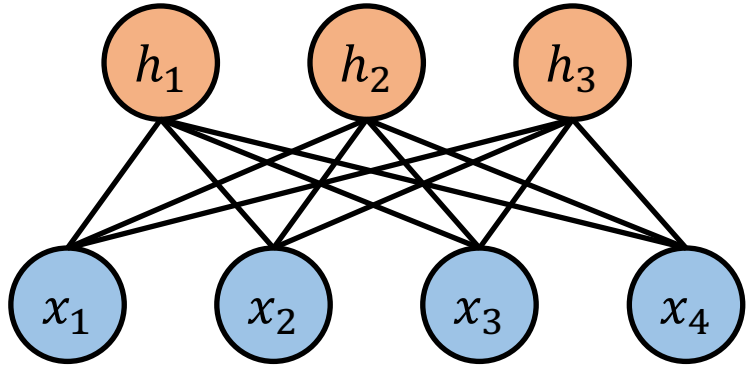


Discrete Latent Variable Models

Inference: Computing Marginals $p(h)$ & Maximum A Posterior (MAP) $\arg\max_h p(h|x)$

Due to conditional independence, inference is tractable!

RBMs



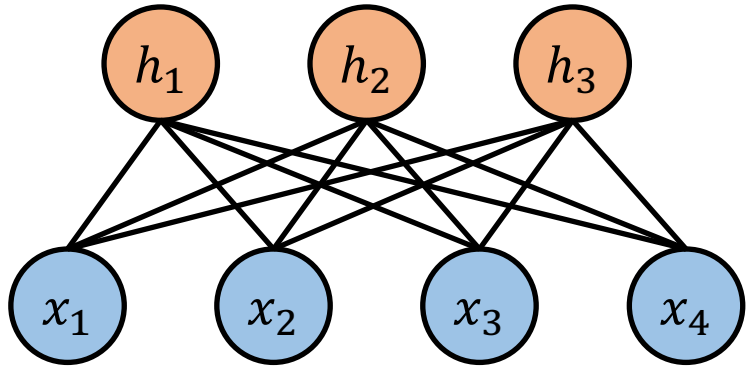
Discrete Latent Variable Models

Inference: Computing Marginals $p(h)$ & Maximum A Posterior (MAP) $\arg\max_h p(h|x)$

Due to conditional independence, inference is tractable!

Learning: Maximum Likelihood

RBMs



Discrete Latent Variable Models

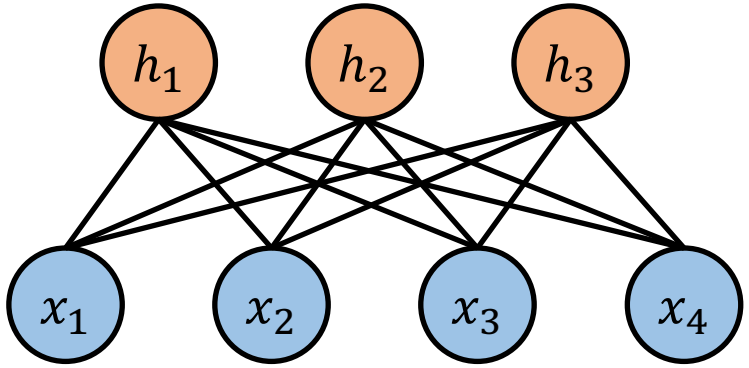
Inference: Computing Marginals $p(h)$ & Maximum A Posterior (MAP) $\arg\max_h p(h|x)$

Due to conditional independence, inference is tractable!

Learning: Maximum Likelihood

$$\max_\theta \quad \log p_\theta(x) = \log \int p_\theta(x,h)dh$$
$$= \log \int \exp \log p_\theta(x,h)dh$$
$$= \log \int \exp\left(-E_\theta(x,h) - \log Z\right) dh$$

RBMs



Discrete Latent Variable Models

Inference: Computing Marginals $p(h)$ & Maximum A Posterior (MAP) $\arg\max_h p(h|x)$

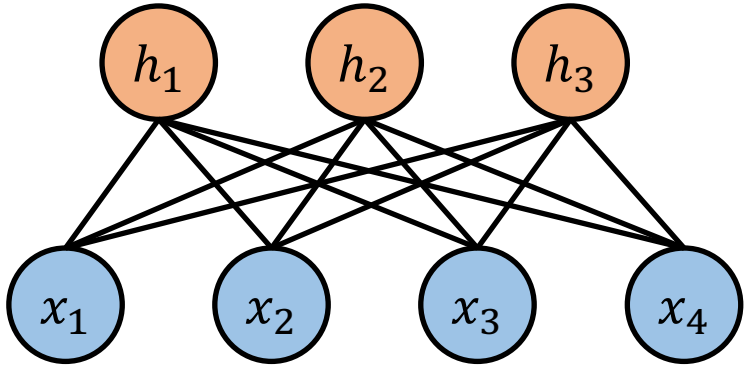
Due to conditional independence, inference is tractable!

Learning: Maximum Likelihood

$$\max_\theta \quad \log p_\theta(x) = \log \int p_\theta(x,h)dh$$
$$= \log \int \exp \log p_\theta(x,h)dh$$
$$= \log \int \exp\left(-E_\theta(x,h) - \log Z\right) dh$$

Intractable! Why?

RBMs



Discrete Latent Variable Models

Inference: Computing Marginals $p(h)$ & Maximum A Posterior (MAP) $\arg\max_h p(h|x)$

Due to conditional independence, inference is tractable!

Learning: Maximum Likelihood

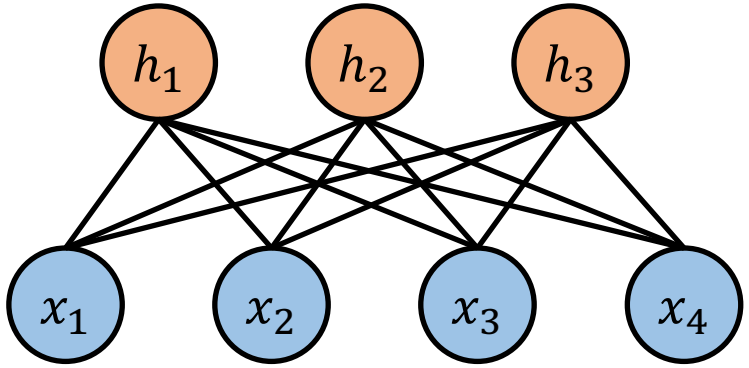
$$\max_\theta \quad \log p_\theta(x) = \log \int p_\theta(x,h)dh$$

$$= \log \int \exp \log p_\theta(x,h)dh$$

$$= \log \int \exp\left(-E_\theta(x,h) - \log Z\right) dh$$

Intractable! Why?

RBMs



$$Z = \int \int \exp\left(-E_\theta(x,h)\right) dxdh$$

Stochastic Approximated Gradient

Learning: Maximum Likelihood

$$
\begin{aligned}
\frac{\partial \log p_\theta(x)}{\partial \theta} &= \frac{1}{p_\theta(x)} \frac{\partial p_\theta(x)}{\partial \theta} \\
&= \frac{1}{p_\theta(x)} \frac{\partial \int p_\theta(x,h) dh}{\partial \theta} \\
&= \frac{1}{p_\theta(x)} \int \frac{\partial p_\theta(x,h)}{\partial \theta} dh \\
&= \frac{1}{p_\theta(x)} \int \frac{\partial \frac{1}{Z} \exp\left(-E_\theta(x,h)\right)}{\partial \theta} dh \\
&= \frac{1}{p_\theta(x)} \int \frac{\left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) \exp\left(-E_\theta(x,h)\right) Z - \frac{\partial Z}{\partial \theta} \exp\left(-E_\theta(x,h)\right)}{Z^2} dh
\end{aligned}
$$

Stochastic Approximated Gradient

Learning: Maximum Likelihood

$$
\begin{aligned}
\frac{\partial \log p_\theta(x)}{\partial \theta} &= \frac{1}{p_\theta(x)} \int \frac{\left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) \exp\left(-E_\theta(x,h)\right) Z - \frac{\partial Z}{\partial \theta} \exp\left(-E_\theta(x,h)\right)}{Z^2} dh \\
&= \frac{1}{p_\theta(x)} \int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(x,h) dh - \frac{1}{p_\theta(x)} \int \frac{1}{Z} \frac{\partial Z}{\partial \theta} p_\theta(x,h) dh \\
&= \int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(h|x) dh - \int \frac{1}{Z} \frac{\partial Z}{\partial \theta} p_\theta(h|x) dh \\
&= \int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(h|x) dh - \frac{1}{Z} \frac{\partial Z}{\partial \theta} \\
&= \int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(h|x) dh - \frac{1}{Z} \frac{\partial \int \int \exp\left(-E_\theta(x,h)\right) dx dh}{\partial \theta} \\
&= \int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(h|x) dh - \int \int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(x,h) dx dh
\end{aligned}
$$

Stochastic Approximated Gradient

Learning: Maximum Likelihood

$$\frac{\partial \log p_\theta(x)}{\partial \theta} = \int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(h|x) dh - \int\int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(x,h) dx dh$$
$$= \mathbb{E}_{p_\theta(h|x)}\left[-\frac{\partial E_\theta(x,h)}{\partial \theta}\right] - \mathbb{E}_{p_\theta(h,x)}\left[-\frac{\partial E_\theta(x,h)}{\partial \theta}\right]$$

Stochastic Approximated Gradient

Learning: Maximum Likelihood

$$\begin{aligned}\frac{\partial \log p_\theta(x)}{\partial \theta} &= \int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(h|x) dh - \int\int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(x,h) dx dh \\ &= \mathbb{E}_{p_\theta(h|x)}\left[-\frac{\partial E_\theta(x,h)}{\partial \theta}\right] - \mathbb{E}_{p_\theta(h,x)}\left[-\frac{\partial E_\theta(x,h)}{\partial \theta}\right]\end{aligned}$$

Recall we sample multiple training data and maximize the summed log likelihood of them, which in expectation amounts to:

Stochastic Approximated Gradient

Learning: Maximum Likelihood

$$\begin{aligned}\frac{\partial \log p_\theta(x)}{\partial \theta} &= \int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(h|x) dh - \int\int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(x,h) dx dh \\ &= \mathbb{E}_{p_\theta(h|x)}\left[-\frac{\partial E_\theta(x,h)}{\partial \theta}\right] - \mathbb{E}_{p_\theta(h,x)}\left[-\frac{\partial E_\theta(x,h)}{\partial \theta}\right]\end{aligned}$$

Recall we sample multiple training data and maximize the summed log likelihood of them, which in expectation amounts to:

$$\begin{aligned}\min_\theta \quad \mathrm{KL}\left(p_{\mathrm{data}}(x) \| p_\theta(x)\right) &= \int p_{\mathrm{data}}(x) \log p_{\mathrm{data}}(x) dx - \int p_{\mathrm{data}}(x) \log p_\theta(x) dx \\ &= -\mathcal{H}_{p_{\mathrm{data}}(x)} + \mathrm{CrossEntropy}(p_{\mathrm{data}}(x), p_\theta(x))\end{aligned}$$

Stochastic Approximated Gradient

Learning: Maximum Likelihood

$$\begin{aligned}\frac{\partial \log p_\theta(x)}{\partial \theta} &= \int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(h|x)dh - \int\int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(x,h)dxdh \\ &= \mathbb{E}_{p_\theta(h|x)}\left[-\frac{\partial E_\theta(x,h)}{\partial \theta}\right] - \mathbb{E}_{p_\theta(h,x)}\left[-\frac{\partial E_\theta(x,h)}{\partial \theta}\right]\end{aligned}$$

Recall we sample multiple training data and maximize the summed log likelihood of them, which in expectation amounts to:

$$\begin{aligned}\min_\theta \quad \mathrm{KL}\left(p_{\mathrm{data}}(x)\|p_\theta(x)\right) &= \int p_{\mathrm{data}}(x)\log p_{\mathrm{data}}(x)dx - \int p_{\mathrm{data}}(x)\log p_\theta(x)dx \\ &= -\mathcal{H}_{p_{\mathrm{data}}(x)} + \mathrm{CrossEntropy}(p_{\mathrm{data}}(x), p_\theta(x))\end{aligned}$$

$$\min_\theta \quad \mathrm{CrossEntropy}(p_{\mathrm{data}}(x), p_\theta(x)) \quad \Leftrightarrow \quad \max_\theta \int p_{\mathrm{data}}(x)\log p_\theta(x)dx$$

Maximum Likelihood

Stochastic Approximated Gradient

Learning: Maximum Likelihood

$$\begin{aligned}\frac{\partial \log p_\theta(x)}{\partial \theta} &= \int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(h|x) dh - \int\int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(x,h) dx dh \\ &= \mathbb{E}_{p_\theta(h|x)}\left[-\frac{\partial E_\theta(x,h)}{\partial \theta}\right] - \mathbb{E}_{p_\theta(h,x)}\left[-\frac{\partial E_\theta(x,h)}{\partial \theta}\right]\end{aligned}$$

Since we care about $\max_\theta \int p_{\text{data}}(x) \log p_\theta(x) dx$

Stochastic Approximated Gradient

Learning: Maximum Likelihood

$$\begin{aligned}\frac{\partial \log p_\theta(x)}{\partial \theta} &= \int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(h|x) dh - \int\int \left(-\frac{\partial E_\theta(x,h)}{\partial \theta}\right) p_\theta(x,h) dx dh \\ &= \mathbb{E}_{p_\theta(h|x)}\left[-\frac{\partial E_\theta(x,h)}{\partial \theta}\right] - \mathbb{E}_{p_\theta(h,x)}\left[-\frac{\partial E_\theta(x,h)}{\partial \theta}\right]\end{aligned}$$

Since we care about $\max_\theta \int p_{\text{data}}(x) \log p_\theta(x) dx$

we have the gradient

$$\int p_{\text{data}}(x) \frac{\partial \log p_\theta(x)}{\partial \theta} dx = \mathbb{E}_{p_\theta(h|x)p_{\text{data}}(x)}\left[-\frac{\partial E_\theta(x,h)}{\partial \theta}\right] - \mathbb{E}_{p_\theta(h,x)}\left[-\frac{\partial E_\theta(x,h)}{\partial \theta}\right]$$

Stochastic Approximated Gradient

Learning: Maximum Likelihood

Stochastic Approximated Gradient

$$\int p_{\text{data}}(x) \frac{\partial \log p_\theta(x)}{\partial \theta} dx = \mathbb{E}_{p_\theta(h|x)p_{\text{data}}(x)} \left[-\frac{\partial E_\theta(x,h)}{\partial \theta} \right] - \mathbb{E}_{p_\theta(h,x)} \left[-\frac{\partial E_\theta(x,h)}{\partial \theta} \right]$$

Monte Carlo Estimation!

Stochastic Approximated Gradient

Learning: Maximum Likelihood

Stochastic Approximated Gradient

$$\int p_{\text{data}}(x) \frac{\partial \log p_\theta(x)}{\partial \theta} dx = \mathbb{E}_{p_\theta(h|x)p_{\text{data}}(x)} \left[-\frac{\partial E_\theta(x,h)}{\partial \theta} \right] - \mathbb{E}_{p_\theta(h,x)} \left[-\frac{\partial E_\theta(x,h)}{\partial \theta} \right]$$

Monte Carlo Estimation!

Positive Gradient: sample from the data distribution $p_\theta(h|x)p_{\text{data}}(x)$

Stochastic Approximated Gradient

Learning: Maximum Likelihood

Stochastic Approximated Gradient

$$\int p_{\text{data}}(x) \frac{\partial \log p_\theta(x)}{\partial \theta} dx = \mathbb{E}_{p_\theta(h|x)p_{\text{data}}(x)} \left[-\frac{\partial E_\theta(x,h)}{\partial \theta} \right] - \mathbb{E}_{p_\theta(h,x)} \left[-\frac{\partial E_\theta(x,h)}{\partial \theta} \right]$$

Monte Carlo Estimation!

Positive Gradient: sample from the data distribution $p_\theta(h|x)p_{\text{data}}(x)$

Negative Gradient: sample from the model distribution $p_\theta(h,x)$

Stochastic Approximated Gradient

Learning: Maximum Likelihood

Stochastic Approximated Gradient

$$\int p_{\text{data}}(x) \frac{\partial \log p_\theta(x)}{\partial \theta} dx = \mathbb{E}_{p_\theta(h|x)p_{\text{data}}(x)} \left[-\frac{\partial E_\theta(x,h)}{\partial \theta} \right] - \mathbb{E}_{p_\theta(h,x)} \left[-\frac{\partial E_\theta(x,h)}{\partial \theta} \right]$$

Monte Carlo Estimation!

Positive Gradient: sample from the data distribution $p_\theta(h|x)p_{\text{data}}(x)$

Negative Gradient: sample from the model distribution $p_\theta(h,x)$

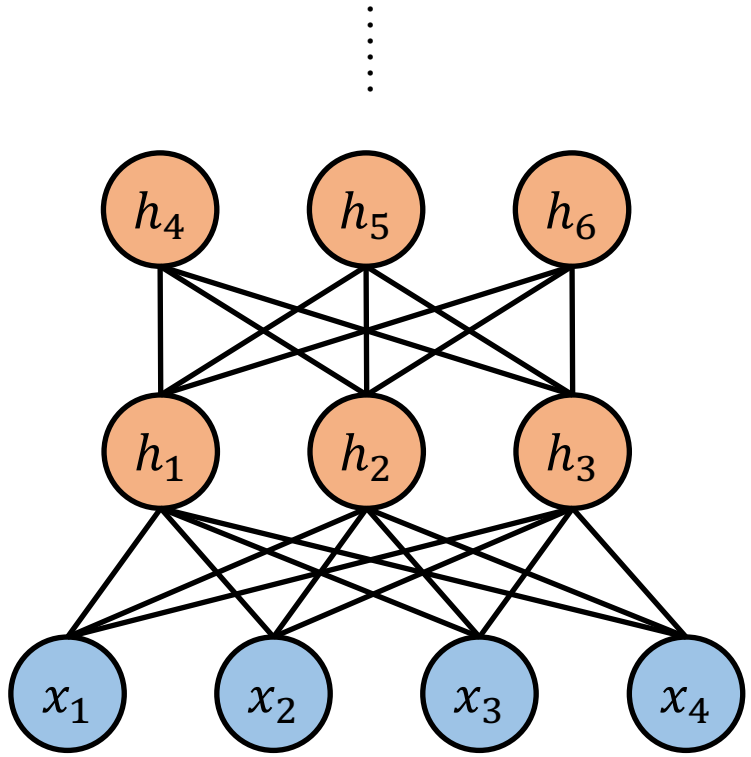
If we use finite-step Gibbs sampler, this method is called *Contrastive Divergence* (CD) [3]!

Contents

Discrete Latent Variable Models

What if we stack multiple RBMs?

Deep RBMs [4]

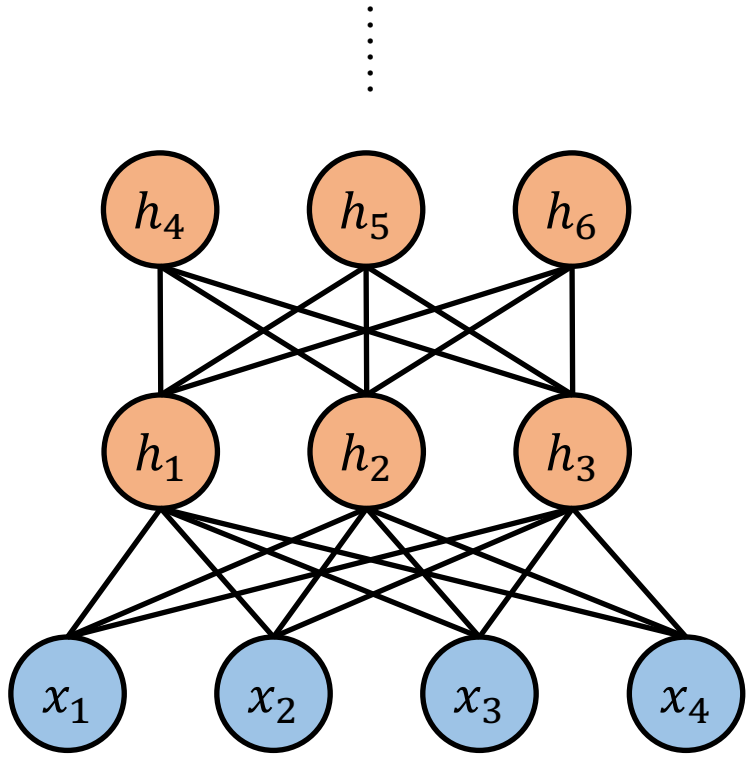


Discrete Latent Variable Models

What if we stack multiple RBMs?

- We could still use CD to learn
- But inference becomes intractable!

Deep RBMs [4]



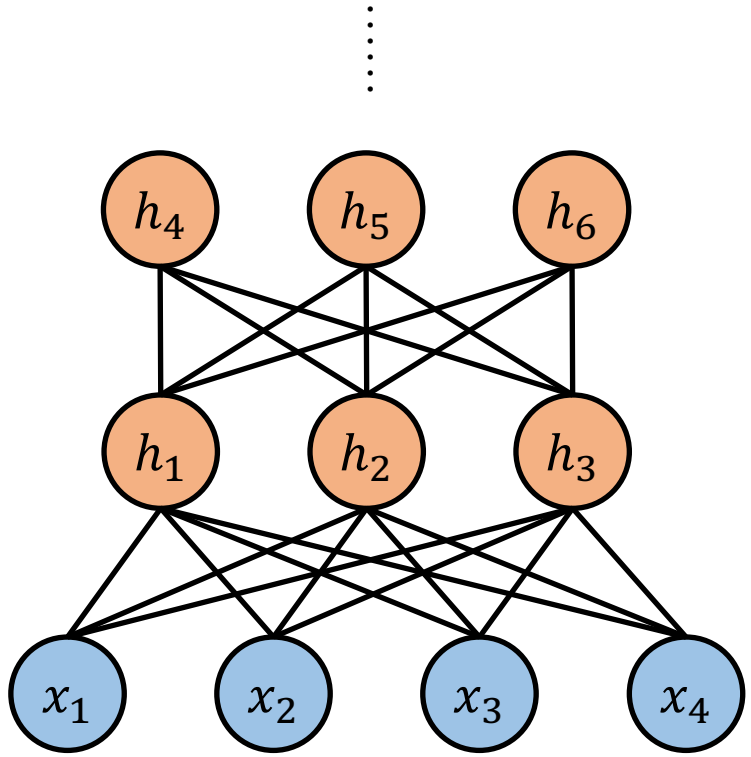
Discrete Latent Variable Models

What if we stack multiple RBMs?

- We could still use CD to learn
- But inference becomes intractable!

How to deal with the general case that hidden units (latent variables) are dependent?

Deep RBMs [4]



Discrete Latent Variable Models

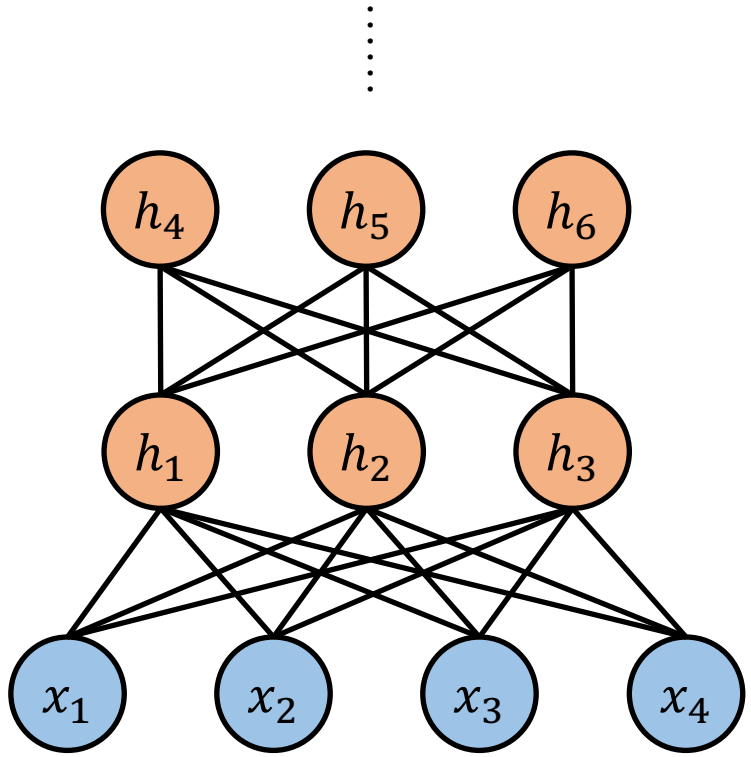
What if we stack multiple RBMs?

- We could still use CD to learn
- But inference becomes intractable!

How to deal with the general case that hidden units (latent variables) are dependent?

(Amortized/Neural) Variation Inference!

Deep RBMs [4]



Discrete Latent Variable Models

Given data $X \in \mathbb{R}^d$, Maximum Likelihood is:

$$\max_{\theta} \quad \log p_\theta(X)$$

We introduce latent variable $Z \in \{0,1\}^m$ or $Z \in \mathbb{R}^m$

$$p_\theta(X) = \int_Z p_\theta(X, Z) dZ$$
$$= \int_Z p_\theta(X|Z) p_\theta(Z) dZ$$

Intractable Integration!

Variational Inference

Variational Approximation

$$\log p_\theta(X) = \log\left(\frac{p_\theta(X,Z)}{p_\theta(Z|X)}\right)$$
$$= \log\left(\frac{p_\theta(X,Z)}{q_\phi(Z|X)}\frac{q_\phi(Z|X)}{p_\theta(Z|X)}\right)$$

Variational Inference

Variational Approximation

$$\begin{aligned}\log p_\theta(X) &= \log\left(\frac{p_\theta(X,Z)}{p_\theta(Z|X)}\right) \\ &= \log\left(\frac{p_\theta(X,Z)}{q_\phi(Z|X)}\frac{q_\phi(Z|X)}{p_\theta(Z|X)}\right)\end{aligned}$$

Integrating from both sides:

$$\begin{aligned}\log p_\theta(X) &= \int q_\phi(Z|X)\log p_\theta(X)dZ \\ &= \int q_\phi(Z|X)\log\left(\frac{p_\theta(X,Z)}{q_\phi(Z|X)}\frac{q_\phi(Z|X)}{p_\theta(Z|X)}\right)dZ \\ &= \int q_\phi(Z|X)\log\left(\frac{p_\theta(X,Z)}{q_\phi(Z|X)}\right)dZ + \int q_\phi(Z|X)\log\left(\frac{q_\phi(Z|X)}{p_\theta(Z|X)}\right)dZ \\ &= \mathbb{E}_{q_\phi(Z|X)}\left[\log\left(\frac{p_\theta(X,Z)}{q_\phi(Z|X)}\right)\right] + \mathrm{KL}\left(q_\phi(Z|X)\|p_\theta(Z|X)\right)\end{aligned}$$

Variational Inference

Variational Approximation

$$\log p_\theta(X) = \log\left(\frac{p_\theta(X,Z)}{p_\theta(Z|X)}\right)$$
$$= \log\left(\frac{p_\theta(X,Z)}{q_\phi(Z|X)}\frac{q_\phi(Z|X)}{p_\theta(Z|X)}\right)$$

Integrating from both sides:

$$\log p_\theta(X) = \int q_\phi(Z|X)\log p_\theta(X)dZ$$
$$= \int q_\phi(Z|X)\log\left(\frac{p_\theta(X,Z)}{q_\phi(Z|X)}\frac{q_\phi(Z|X)}{p_\theta(Z|X)}\right)dZ$$
$$= \int q_\phi(Z|X)\log\left(\frac{p_\theta(X,Z)}{q_\phi(Z|X)}\right)dZ + \int q_\phi(Z|X)\log\left(\frac{q_\phi(Z|X)}{p_\theta(Z|X)}\right)dZ$$
$$= \underbrace{\mathbb{E}_{q_\phi(Z|X)}\left[\log\left(\frac{p_\theta(X,Z)}{q_\phi(Z|X)}\right)\right]}_{\text{Evidence Lower Bound (ELBO)}} + \underbrace{\mathrm{KL}\left(q_\phi(Z|X)\|p_\theta(Z|X)\right)}_{\text{Kullback-Leibler Divergence}}$$

Variational Inference

Since true posterior $p_\theta(Z|X)$ is often unknown, KL term is intractable

ELBO:

$$\begin{aligned}
\mathbb{E}_{q_\phi(Z|X)}\left[\log\left(\frac{p_\theta(X,Z)}{q_\phi(Z|X)}\right)\right] &= \mathbb{E}_{q_\phi(Z|X)}\left[\log\left(\frac{p_\theta(X|Z)p_\theta(Z)}{q_\phi(Z|X)}\right)\right] \\
&= \mathbb{E}_{q_\phi(Z|X)}\left[\log\left(p_\theta(X|Z)\right)\right] + \mathbb{E}_{q_\phi(Z|X)}\left[\log\left(\frac{p_\theta(Z)}{q_\phi(Z|X)}\right)\right] \\
&= -\mathbb{E}_{q_\phi(Z|X)}\left[-\log\left(p_\theta(X|Z)\right)\right] - \mathrm{KL}\left(q_\phi(Z|X)\|p_\theta(Z)\right)
\end{aligned}$$

Reconstruction Error/Loss — Regularizer

Variational Inference

Since true posterior $p_\theta(Z|X)$ is often unknown, KL term is intractable

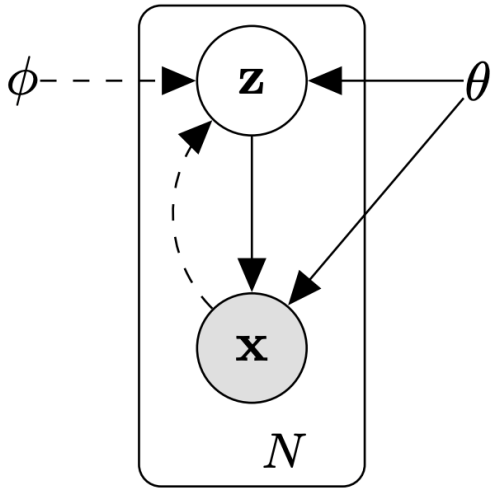
ELBO:

$$\mathbb{E}_{q_\phi(Z|X)}\left[\log\left(\frac{p_\theta(X,Z)}{q_\phi(Z|X)}\right)\right] = \mathbb{E}_{q_\phi(Z|X)}\left[\log\left(\frac{p_\theta(X|Z)p_\theta(Z)}{q_\phi(Z|X)}\right)\right]$$

$$= \mathbb{E}_{q_\phi(Z|X)}\left[\log\left(p_\theta(X|Z)\right)\right] + \mathbb{E}_{q_\phi(Z|X)}\left[\log\left(\frac{p_\theta(Z)}{q_\phi(Z|X)}\right)\right]$$

$$= -\mathbb{E}_{q_\phi(Z|X)}\left[-\log\left(p_\theta(X|Z)\right)\right] - \mathrm{KL}\left(q_\phi(Z|X)\|p_\theta(Z)\right)$$

Reconstruction Error/Loss Regularizer



Encoder: $q_\phi(Z|X)$

Decoder: $p_\theta(X|Z)$

Prior: $p_\theta(Z)$

Variational Inference

Since true posterior $p_\theta(Z|X)$ is often unknown, KL term is intractable

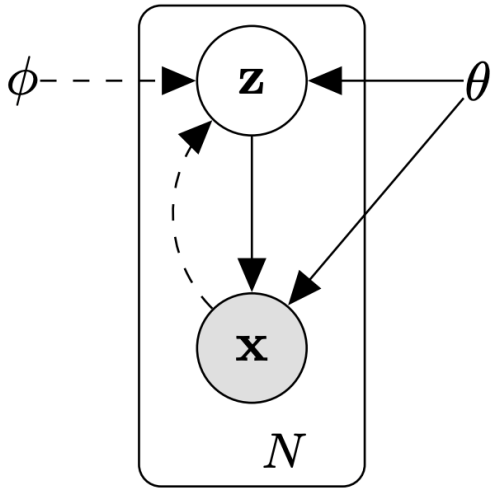
ELBO:

$$\mathbb{E}_{q_\phi(Z|X)}\left[\log\left(\frac{p_\theta(X,Z)}{q_\phi(Z|X)}\right)\right] = \mathbb{E}_{q_\phi(Z|X)}\left[\log\left(\frac{p_\theta(X|Z)p_\theta(Z)}{q_\phi(Z|X)}\right)\right]$$

$$= \mathbb{E}_{q_\phi(Z|X)}\left[\log\left(p_\theta(X|Z)\right)\right] + \mathbb{E}_{q_\phi(Z|X)}\left[\log\left(\frac{p_\theta(Z)}{q_\phi(Z|X)}\right)\right]$$

$$= -\mathbb{E}_{q_\phi(Z|X)}\left[-\log\left(p_\theta(X|Z)\right)\right] - \mathrm{KL}\left(q_\phi(Z|X)\|p_\theta(Z)\right)$$

Reconstruction Error/Loss Regularizer



Encoder: $q_\phi(Z|X)$ Encoder is local, i.e., for each X

Decoder: $p_\theta(X|Z)$ there is a separate set of ϕ

Prior: $p_\theta(Z)$

Amortized/Neural Variational Inference

Since true posterior $p_\theta(Z|X)$ is often unknown, KL term is intractable

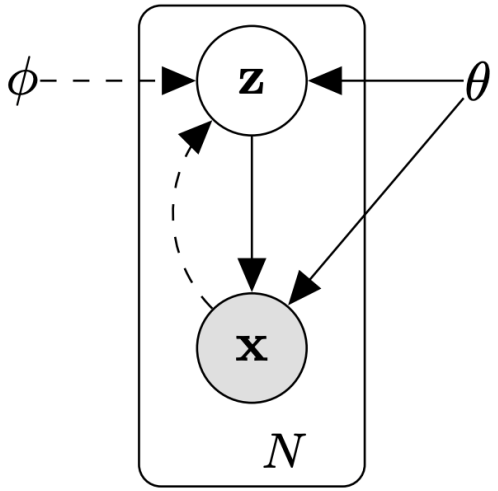
ELBO:

$$\mathbb{E}_{q_\phi(Z|X)}\left[\log\left(\frac{p_\theta(X,Z)}{q_\phi(Z|X)}\right)\right] = \mathbb{E}_{q_\phi(Z|X)}\left[\log\left(\frac{p_\theta(X|Z)p_\theta(Z)}{q_\phi(Z|X)}\right)\right]$$

$$= \mathbb{E}_{q_\phi(Z|X)}\left[\log\left(p_\theta(X|Z)\right)\right] + \mathbb{E}_{q_\phi(Z|X)}\left[\log\left(\frac{p_\theta(Z)}{q_\phi(Z|X)}\right)\right]$$

$$= -\mathbb{E}_{q_\phi(Z|X)}\left[-\log\left(p_\theta(X|Z)\right)\right] - \mathrm{KL}\left(q_\phi(Z|X)\|p_\theta(Z)\right)$$

Reconstruction Error/Loss Regularizer



Encoder: $q_\phi(Z|X)$ Encoder is global, i.e., for each X

Decoder: $p_\theta(X|Z)$ there is a shared set of ϕ [10]

Prior: $p_\theta(Z)$

Amortized/Neural Variational Inference

Encoder is global, i.e., for each X, there is a shared set of ϕ

Examples:

If Z is continuous,

$$q_\phi(Z|X) = \mathcal{N}(Z|\mu, \sigma^2 I)$$
$$\mu = \text{EncoderNetwork}_\phi(X)$$
$$\log \sigma^2 = \text{EncoderNetwork}_\phi(X)$$

Amortized/Neural Variational Inference

Encoder is global, i.e., for each X, there is a shared set of ϕ

Examples:

If Z is continuous,

$$q_\phi(Z|X) = \mathcal{N}(Z|\mu, \sigma^2 I)$$
$$\mu = \text{EncoderNetwork}_\phi(X)$$
$$\log \sigma^2 = \text{EncoderNetwork}_\phi(X)$$

If Z is binary,

$$q_\phi(Z|X) = \prod_i \text{Bernoulli}(Z_i|\eta_i)$$
$$\eta = \text{EncoderNetwork}_\phi(X)$$

Contents

- Discrete Latent Variable Models:
  - Restricted Boltzmann Machines (RBMs)
  - Contrastive Divergence
- Variational Inference & Amortized Inference
- **Learning**
  - Score based gradient estimator + Variance Reduction
  - Reparameterization based gradient estimator
  - Wake-sleep algorithm

Learning in Neural/Amortized Variational Inference

Negative ELBO:

$$\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$$

$$\approx -\frac{1}{N} \sum_{\substack{i=1 \\ Z_i \sim q_\phi(Z|X)}}^{N} \log \left(p_\theta(X|Z_i) \right) + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$$

Learning in Neural/Amortized Variational Inference

Negative ELBO:
$$\mathcal{L}(\phi,\theta) = \mathbb{E}_{q_\phi(Z|X)}\left[-\log\left(p_\theta(X|Z)\right)\right] + \mathrm{KL}\left(q_\phi(Z|X)\|p_\theta(Z)\right)$$
$$\approx -\frac{1}{N}\sum_{\substack{i=1 \\ Z_i\sim q_\phi(Z|X)}}^{N} \log\left(p_\theta(X|Z_i)\right) + \mathrm{KL}\left(q_\phi(Z|X)\|p_\theta(Z)\right)$$

Gradient of Decoder (assuming prior is not learnable for simplicity) [5]:

$$\frac{\partial\mathcal{L}(\phi,\theta)}{\partial\theta} = \mathbb{E}_{q_\phi(Z|X)}\left[-\frac{\partial\log\left(p_\theta(X|Z)\right)}{\partial\theta}\right]$$
$$\approx -\frac{1}{N}\sum_{\substack{i=1 \\ Z_i\sim q_\phi(Z|X)}}^{N} \frac{\partial\log\left(p_\theta(X|Z_i)\right)}{\partial\theta}$$

Learning in Neural/Amortized Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$

Gradient of Encoder [5]:

$$
\begin{aligned}
\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} &= -\int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log p_\theta(X|Z) dZ - \int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log p_\theta(Z) dZ \\
&\quad + \int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log q_\phi(Z|X) dZ + \int \frac{\partial q_\phi(Z|X)}{\partial \phi} dZ \\
&= -\int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log p_\theta(X, Z) dZ + \int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log q_\phi(Z|X) dZ + \frac{\partial}{\partial \phi} \int q_\phi(Z|X) dZ \\
&= -\int \frac{\partial q_\phi(Z|X)}{\partial \phi} \left(\log p_\theta(X, Z) - \log q_\phi(Z|X) \right) dZ
\end{aligned}
$$

Learning in Neural/Amortized Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$

Gradient of Encoder [5]:

$$\begin{aligned}
\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} &= -\int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log p_\theta(X|Z) dZ - \int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log p_\theta(Z) dZ \\
&\quad + \int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log q_\phi(Z|X) dZ + \int \frac{\partial q_\phi(Z|X)}{\partial \phi} dZ \\
&= -\int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log p_\theta(X, Z) dZ + \int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log q_\phi(Z|X) dZ + \frac{\partial}{\partial \phi} \int q_\phi(Z|X) dZ \\
&= -\int \frac{\partial q_\phi(Z|X)}{\partial \phi} \left(\log p_\theta(X, Z) - \log q_\phi(Z|X) \right) dZ
\end{aligned}$$

$$\frac{\partial q_\phi(Z|X)}{\partial \phi} = q_\phi(Z|X) \frac{\partial \log q_\phi(Z|X)}{\partial \phi}$$

Log Derivative Trick

Learning in Neural/Amortized Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$

Gradient of Encoder [5]:

$$\begin{aligned}
\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} &= -\int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log p_\theta(X|Z) dZ - \int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log p_\theta(Z) dZ \\
&\quad + \int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log q_\phi(Z|X) dZ + \int \frac{\partial q_\phi(Z|X)}{\partial \phi} dZ \\
&= -\int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log p_\theta(X, Z) dZ + \int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log q_\phi(Z|X) dZ + \frac{\partial}{\partial \phi} \int q_\phi(Z|X) dZ \\
&= -\int \frac{\partial q_\phi(Z|X)}{\partial \phi} \left(\log p_\theta(X, Z) - \log q_\phi(Z|X) \right) dZ
\end{aligned}$$

$$\frac{\partial q_\phi(Z|X)}{\partial \phi} = q_\phi(Z|X) \frac{\partial \log q_\phi(Z|X)}{\partial \phi}$$

Log Derivative Trick

Score Function

Learning in Neural/Amortized Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)}\left[-\log\left(p_\theta(X|Z)\right)\right] + \mathrm{KL}\left(q_\phi(Z|X) \| p_\theta(Z)\right)$

Gradient of Encoder [5]:

$$
\begin{aligned}
\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} &= -\int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log p_\theta(X|Z) dZ - \int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log p_\theta(Z) dZ \\
&\quad + \int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log q_\phi(Z|X) dZ + \int \frac{\partial q_\phi(Z|X)}{\partial \phi} dZ \\
&= -\int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log p_\theta(X, Z) dZ + \int \frac{\partial q_\phi(Z|X)}{\partial \phi} \log q_\phi(Z|X) dZ + \frac{\partial}{\partial \phi} \int q_\phi(Z|X) dZ
\end{aligned}
$$

$$\frac{\partial q_\phi(Z|X)}{\partial \phi} = q_\phi(Z|X) \frac{\partial \log q_\phi(Z|X)}{\partial \phi}$$

$$
\begin{aligned}
&= -\int \frac{\partial q_\phi(Z|X)}{\partial \phi} \left(\log p_\theta(X, Z) - \log q_\phi(Z|X)\right) dZ \\
&= -\int q_\phi(Z|X) \frac{\partial \log q_\phi(Z|X)}{\partial \phi} \left(\log p_\theta(X, Z) - \log q_\phi(Z|X)\right) dZ \\
&= -\mathbb{E}_{q_\phi(Z|X)}\left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi} \left(\log p_\theta(X, Z) - \log q_\phi(Z|X)\right)\right] \\
&\approx -\frac{1}{N} \sum_{\substack{i=1 \\ Z_i \sim q_\phi(Z|X)}}^{N} \frac{\partial \log q_\phi(Z_i|X)}{\partial \phi} \left(\log p_\theta(X, Z_i) - \log q_\phi(Z_i|X)\right)
\end{aligned}
$$

Learning in Neural/Amortized Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$

Gradient of Encoder [5] (Score-based Estimator or REINFORCE [7]):

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} \approx -\frac{1}{N} \sum_{\substack{i=1 \\ Z_i \sim q_\phi(Z|X)}}^{N} \frac{\partial \log q_\phi(Z_i|X)}{\partial \phi} \left(\log p_\theta(X, Z_i) - \log q_\phi(Z_i|X) \right)$$

Learning in Neural/Amortized Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$

Gradient of Encoder [5] (Score-based Estimator or REINFORCE [7]):

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} \approx -\frac{1}{N} \sum_{\substack{i=1 \\ Z_i \sim q_\phi(Z|X)}}^{N} \frac{\partial \log q_\phi(Z_i|X)}{\partial \phi} \left(\log p_\theta(X, Z_i) - \log q_\phi(Z_i|X) \right)$$

REINFORCE in Reinforcement Learning:

$$\Delta\theta \propto \mathbb{E}_{p_\theta(a|s)} \left[r(a, s) \nabla_\theta \log p_\theta(a|s) \right]$$

Learning in Neural/Amortized Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$

Gradient of Encoder [5] (Score-based Estimator or REINFORCE [7]):

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} \approx -\frac{1}{N} \sum_{\substack{i=1 \\ Z_i \sim q_\phi(Z|X)}}^{N} \frac{\partial \log q_\phi(Z_i|X)}{\partial \phi} \left(\log p_\theta(X, Z_i) - \log q_\phi(Z_i|X) \right)$$

REINFORCE in Reinforcement Learning:

$$\Delta\theta \propto \mathbb{E}_{p_\theta(a|s)} \left[r(a, s) \nabla_\theta \log p_\theta(a|s) \right]$$

Reward Policy

Learning in Neural/Amortized Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$

Gradient of Encoder [5] (Score-based Estimator or REINFORCE [7]):

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} \approx -\frac{1}{N} \sum_{\substack{i=1 \\ Z_i \sim q_\phi(Z|X)}}^{N} \frac{\partial \log q_\phi(Z_i|X)}{\partial \phi} \left(\log p_\theta(X, Z_i) - \log q_\phi(Z_i|X) \right)$$

Policy Reward

REINFORCE in Reinforcement Learning:

$$\Delta\theta \propto \mathbb{E}_{p_\theta(a|s)} \left[r(a, s) \nabla_\theta \log p_\theta(a|s) \right]$$

Reward Policy

Learning in Neural/Amortized Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$

Gradient of Encoder [5] (Score-based Estimator or REINFORCE [7]):

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} \approx -\frac{1}{N} \sum_{\substack{i=1 \\ Z_i \sim q_\phi(Z|X)}}^{N} \frac{\partial \log q_\phi(Z_i|X)}{\partial \phi} \left(\log p_\theta(X, Z_i) - \log q_\phi(Z_i|X) \right)$$

- This estimator works for both discrete and continuous latent variables!
- But it may come with high variances!

Learning in Neural/Amortized Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)}\left[-\log\left(p_\theta(X|Z)\right)\right] + \mathrm{KL}\left(q_\phi(Z|X) \| p_\theta(Z)\right)$

Gradient of Encoder [5] (Score-based Estimator or REINFORCE [7]):

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} \approx -\frac{1}{N} \sum_{\substack{i=1 \\ Z_i \sim q_\phi(Z|X)}}^{N} \frac{\partial \log q_\phi(Z_i|X)}{\partial \phi} \left(\log p_\theta(X, Z_i) - \log q_\phi(Z_i|X)\right)$$

- This estimator works for both discrete and continuous latent variables!
- But it may come with high variances!

How to reduce variances?

Variance Reduction

There are many variance reduction methods, e.g., see [13]:

- Antithetics
- Stratification
- Common Random Numbers
- Rao-Blackwellization
- Control Variate

……

Variance Reduction

There are many variance reduction methods, e.g., see [13]:

- Antithetics
- Stratification
- Common Random Numbers
- Rao-Blackwellization
- **Control Variate**

……

Control Variate

Suppose we want to compute $\mu = \mathbb{E}_{p(X)}\left[f(X)\right]$

Monte Carlo estimator is $\hat{\mu} = \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} f(X_i)$

Control Variate

Suppose we want to compute $\mu = \mathbb{E}_{p(X)}\left[f(X)\right]$

Monte Carlo estimator is $\hat{\mu} = \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} f(X_i)$

Suppose we know $\theta = \mathbb{E}_{p(X)}\left[h(X)\right]$

Monte Carlo estimator is $\hat{\theta} = \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} h(X_i)$

Control Variate

We can construct a *difference estimator*:

$$\hat{\mu}_{\text{diff}} = \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} (f(X_i) - h(X_i)) + \theta$$

$$= \hat{\mu} - \hat{\theta} + \theta$$

Control Variate

We can construct a *difference estimator*: $\hat{\mu}_{\text{diff}} = \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} (f(X_i) - h(X_i)) + \theta$

$$= \hat{\mu} - \hat{\theta} + \theta$$

It is unbiased since $\mathbb{E}_{p(X)}[\hat{\mu}_{\text{diff}}] = \mathbb{E}_{p(X)}[\hat{\mu}] - \mathbb{E}_{p(X)}[\hat{\theta}] + \theta = \mu$

Control Variate

We can construct a *difference estimator*: $\hat{\mu}_{\text{diff}} = \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} (f(X_i) - h(X_i)) + \theta$

$$= \hat{\mu} - \hat{\theta} + \theta$$

It is unbiased since $\mathbb{E}_{p(X)}[\hat{\mu}_{\text{diff}}] = \mathbb{E}_{p(X)}[\hat{\mu}] - \mathbb{E}_{p(X)}[\hat{\theta}] + \theta = \mu$

Its variance:

$$\begin{aligned}
\text{Var}[\hat{\mu}_{\text{diff}}] &= \text{Var}\left[\hat{\mu} - \hat{\theta} + \theta\right] \\
&= \text{Var}\left[\hat{\mu} - \hat{\theta}\right] \\
&= \text{Var}\left[\frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} f(X_i) - \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} h(X_i)\right] \\
&= \frac{1}{N^2} \sum_{i=1, X_i \sim p(X)}^{N} \text{Var}[f(X_i) - h(X_i)] \\
&= \frac{1}{N} \text{Var}[f(X) - h(X)]
\end{aligned}$$

Control Variate

We can construct a *difference estimator*: $\hat{\mu}_{\text{diff}} = \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} (f(X_i) - h(X_i)) + \theta$

$$= \hat{\mu} - \hat{\theta} + \theta$$

It is unbiased since $\mathbb{E}_{p(X)}[\hat{\mu}_{\text{diff}}] = \mathbb{E}_{p(X)}[\hat{\mu}] - \mathbb{E}_{p(X)}[\hat{\theta}] + \theta = \mu$

Its variance:

$$\begin{aligned}
\text{Var}[\hat{\mu}_{\text{diff}}] &= \text{Var}\left[\hat{\mu} - \hat{\theta} + \theta\right] \\
&= \text{Var}\left[\hat{\mu} - \hat{\theta}\right] \\
&= \text{Var}\left[\frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} f(X_i) - \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} h(X_i)\right] \\
&= \frac{1}{N^2} \sum_{i=1, X_i \sim p(X)}^{N} \text{Var}[f(X_i) - h(X_i)] \\
&= \frac{1}{N} \text{Var}[f(X) - h(X)]
\end{aligned}$$

Control Variate, a.k.a. baseline in RL

Control Variate

We introduce *regression estimator*:

$$\hat{\mu}_\beta = \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} (f(X_i) - \beta h(X_i)) + \beta\theta$$
$$= \hat{\mu} - \beta\hat{\theta} + \beta\theta$$

Control Variate

We introduce *regression estimator*:

$$\hat{\mu}_\beta = \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} (f(X_i) - \beta h(X_i)) + \beta\theta$$
$$= \hat{\mu} - \beta\hat{\theta} + \beta\theta$$

It is unbiased

$$\mathbb{E}_{p(X)}[\hat{\mu}_\beta] = \mathbb{E}_{p(X)}[\hat{\mu}] - \beta\mathbb{E}_{p(X)}[\hat{\theta}] + \beta\theta = \mu$$

Control Variate

We introduce *regression estimator*:

$$\hat{\mu}_\beta = \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} (f(X_i) - \beta h(X_i)) + \beta\theta$$
$$= \hat{\mu} - \beta\hat{\theta} + \beta\theta$$

It is unbiased $\mathbb{E}_{p(X)}[\hat{\mu}_\beta] = \mathbb{E}_{p(X)}[\hat{\mu}] - \beta\mathbb{E}_{p(X)}[\hat{\theta}] + \beta\theta = \mu$

If $\beta = 0$ we have the original Monte Carlo estimator

If $\beta = 1$ we have the difference estimator

Control Variate

We introduce *regression estimator*:

$$\hat{\mu}_\beta = \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} (f(X_i) - \beta h(X_i)) + \beta\theta$$
$$= \hat{\mu} - \beta\hat{\theta} + \beta\theta$$

Its variance is

$$\begin{aligned}
\mathrm{Var}\left[\hat{\mu}_\beta\right] &= \mathrm{Var}\left[\hat{\mu} - \beta\hat{\theta}\right] \\
&= \frac{1}{N}\mathrm{Var}\left[f(X) - \beta h(X)\right] \\
&= \frac{1}{N}\left(\mathrm{Var}\left[f(X)\right] - 2\beta\mathrm{Cov}\left[f(X), h(X)\right] + \beta^2\mathrm{Var}\left[h(X)\right]\right)
\end{aligned}$$

Control Variate

We introduce *regression estimator*:

$$\hat{\mu}_\beta = \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} (f(X_i) - \beta h(X_i)) + \beta\theta$$
$$= \hat{\mu} - \beta\hat{\theta} + \beta\theta$$

Its variance is

$$\begin{aligned}\mathrm{Var}\left[\hat{\mu}_\beta\right] &= \mathrm{Var}\left[\hat{\mu} - \beta\hat{\theta}\right] \\ &= \frac{1}{N}\mathrm{Var}\left[f(X) - \beta h(X)\right] \\ &= \frac{1}{N}\left(\mathrm{Var}\left[f(X)\right] - 2\beta\mathrm{Cov}\left[f(X), h(X)\right] + \beta^2\mathrm{Var}\left[h(X)\right]\right)\end{aligned}$$

Taking the gradient w.r.t. β and set it to zero, we have

$$\beta^* = \frac{\mathrm{Cov}\left[f(X), h(X)\right]}{\mathrm{Var}\left[h(X)\right]}$$

Control Variate

We introduce *regression estimator*:

$$\hat{\mu}_\beta = \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} (f(X_i) - \beta h(X_i)) + \beta\theta$$
$$= \hat{\mu} - \beta\hat{\theta} + \beta\theta$$

Given $\beta^* = \frac{\mathrm{Cov}\left[f(X), h(X)\right]}{\mathrm{Var}\left[h(X)\right]}$, the optimal variance is

$$\begin{aligned}
\mathrm{Var}\left[\hat{\mu}_{\beta^*}\right] &= \frac{1}{N}\left(\mathrm{Var}\left[f(X)\right] - 2\beta^*\mathrm{Cov}\left[f(X), h(X)\right] + {\beta^*}^2\mathrm{Var}\left[h(X)\right]\right) \\
&= \frac{1}{N}\left(\mathrm{Var}\left[f(X)\right] - \frac{\mathrm{Cov}\left[f(X), h(X)\right]^2}{\mathrm{Var}\left[h(X)\right]}\right) \\
&= \frac{\mathrm{Var}\left[f(X)\right]}{N}\left(1 - \frac{\mathrm{Cov}\left[f(X), h(X)\right]^2}{\mathrm{Var}\left[f(X)\right]\mathrm{Var}\left[h(X)\right]}\right) \\
&= \frac{\mathrm{Var}\left[f(X)\right]}{N}\left(1 - \rho^2\right)
\end{aligned}$$

Control Variate

We introduce *regression estimator*:

$$\hat{\mu}_\beta = \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} (f(X_i) - \beta h(X_i)) + \beta\theta$$
$$= \hat{\mu} - \beta\hat{\theta} + \beta\theta$$

Given $\beta^* = \frac{\mathrm{Cov}\left[f(X), h(X)\right]}{\mathrm{Var}\left[h(X)\right]}$, the optimal variance is

$$\mathrm{Var}\left[\hat{\mu}_{\beta^*}\right] = \frac{1}{N}\left(\mathrm{Var}\left[f(X)\right] - 2\beta^*\mathrm{Cov}\left[f(X), h(X)\right] + \beta^{*2}\mathrm{Var}\left[h(X)\right]\right)$$
$$= \frac{1}{N}\left(\mathrm{Var}\left[f(X)\right] - \frac{\mathrm{Cov}\left[f(X), h(X)\right]^2}{\mathrm{Var}\left[h(X)\right]}\right)$$
$$= \frac{\mathrm{Var}\left[f(X)\right]}{N}\left(1 - \frac{\mathrm{Cov}\left[f(X), h(X)\right]^2}{\mathrm{Var}\left[f(X)\right]\mathrm{Var}\left[h(X)\right]}\right)$$
$$= \frac{\mathrm{Var}\left[f(X)\right]}{N}\left(1 - \rho^2\right)$$

If *control variate* is more correlated with the function, the more variance we can reduce!

Control Variate in Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$

Gradient of Encoder [5]:

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} = -\mathbb{E}_{q_\phi(Z|X)} \left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi} \left(\log p_\theta(X, Z) - \log q_\phi(Z|X) \right) \right]$$

Control Variate in Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$

Gradient of Encoder [5]:

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} = -\mathbb{E}_{q_\phi(Z|X)} \left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi} \left(\log p_\theta(X, Z) - \log q_\phi(Z|X) \right) \right]$$

Denote $$\ell_\phi(X, Z) = \log p_\theta(X, Z_i) - \log q_\phi(Z_i|X)$$

We have $$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} = -\mathbb{E}_{q_\phi(Z|X)} \left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi} \ell_\phi(X, Z) \right]$$

Control Variate in Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$

Gradient of Encoder [5]:

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} = -\mathbb{E}_{q_\phi(Z|X)} \left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi} \ell_\phi(X, Z) \right]$$

Control Variate in Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)}\left[-\log\left(p_\theta(X|Z)\right)\right] + \mathrm{KL}\left(q_\phi(Z|X) \| p_\theta(Z)\right)$

Gradient of Encoder [5]:

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} = -\mathbb{E}_{q_\phi(Z|X)}\left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi} \ell_\phi(X, Z)\right]$$

For any c that does not depend on Z, we have

$$\begin{aligned}
\mathbb{E}_{q_\phi(Z|X)}\left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi} c\right] &= c\mathbb{E}_{q_\phi(Z|X)}\left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi}\right] \\
&= c \int q_\phi(Z|X) \frac{\partial \log q_\phi(Z|X)}{\partial \phi} dZ \\
&= c \int \frac{\partial q_\phi(Z|X)}{\partial \phi} dZ \\
&= c \frac{\partial}{\partial \phi} \int q_\phi(Z|X) dZ \\
&= c \frac{\partial 1}{\partial \phi} = 0
\end{aligned}$$

Control Variate in Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$

Gradient of Encoder [5]:

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} = -\mathbb{E}_{q_\phi(Z|X)} \left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi} (\ell_\phi(X, Z) - c) \right]$$

Recall control variate estimator

$$\begin{aligned} \hat{\mu}_{\text{diff}} &= \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} (f(X_i) - h(X_i)) + \theta \\ &= \hat{\mu} - \hat{\theta} + \theta \end{aligned}$$

Control Variate in Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$

Gradient of Encoder [5]:

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} = -\mathbb{E}_{q_\phi(Z|X)} \left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi} (\ell_\phi(X, Z) - c) \right]$$

Recall control variate estimator

$$\begin{aligned} \hat{\mu}_{\text{diff}} &= \frac{1}{N} \sum_{i=1, X_i \sim p(X)}^{N} (f(X_i) - h(X_i)) + \theta \\ &= \hat{\mu} - \hat{\theta} + \theta \end{aligned}$$

In our case, $h(X) = c \dfrac{\partial \log q_\phi(Z|X)}{\partial \phi}$ and $\theta = 0$

Control Variate in Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)}\left[-\log\left(p_\theta(X|Z)\right)\right] + \mathrm{KL}\left(q_\phi(Z|X) \| p_\theta(Z)\right)$

Gradient of Encoder [5]:

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} = -\mathbb{E}_{q_\phi(Z|X)}\left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi}(\ell_\phi(X, Z) - c)\right]$$

Recall control variate estimator

$$\begin{aligned}\hat{\mu}_{\text{diff}} &= \frac{1}{N}\sum_{i=1, X_i \sim p(X)}^{N}(f(X_i) - h(X_i)) + \theta \\ &= \hat{\mu} - \hat{\theta} + \theta\end{aligned}$$

In our case, $h(X) = c\dfrac{\partial \log q_\phi(Z|X)}{\partial \phi}$ and $\theta = 0$

How to choose c to increase correlation?

Control Variate in Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$

Gradient of Encoder [5]:

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} = -\mathbb{E}_{q_\phi(Z|X)} \left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi} (\ell_\phi(X, Z) - c) \right]$$

Since one source of stochasticity comes from X, making c depend on X could help improve correlation

Control Variate in Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)}\left[-\log\left(p_\theta(X|Z)\right)\right] + \mathrm{KL}\left(q_\phi(Z|X) \| p_\theta(Z)\right)$

Gradient of Encoder [5]:

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} = -\mathbb{E}_{q_\phi(Z|X)}\left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi}(\ell_\phi(X, Z) - c)\right]$$

Since one source of stochasticity comes from X, making c depend on X could help improve correlation

In [5], a neural net $C_\psi(X)$ is used and learned to minimize the following objective:

$$\min_\psi \quad \mathbb{E}_{q_\phi(Z|X)}\left[(\ell_\phi(X, Z) - C_\psi(X) - c)^2\right]$$

Control Variate in Variational Inference

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)}\left[-\log\left(p_\theta(X|Z)\right)\right] + \mathrm{KL}\left(q_\phi(Z|X) \| p_\theta(Z)\right)$

Gradient of Encoder [5]:

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} = -\mathbb{E}_{q_\phi(Z|X)}\left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi}(\ell_\phi(X, Z) - c)\right]$$

Since one source of stochasticity comes from X, making c depend on X could help improve correlation

In [5], a neural net $C_\psi(X)$ is used and learned to minimize the following objective:

$$\min_\psi \quad \mathbb{E}_{q_\phi(Z|X)}\left[(\ell_\phi(X, Z) - C_\psi(X) - c)^2\right]$$

There are follow-up papers on learning baselines to reduce variances [14,15]

Reparameterization Trick

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$

Gradient of Encoder [5]:

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} = -\mathbb{E}_{q_\phi(Z|X)} \left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi} \left(\log p_\theta(X, Z) - \log q_\phi(Z|X) \right) \right]$$

If Z is continuous, *reparameterization trick* is potentially another way to estimate the gradient!

Reparameterization Trick

Negative ELBO: $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[-\log \left(p_\theta(X|Z) \right) \right] + \mathrm{KL} \left(q_\phi(Z|X) \| p_\theta(Z) \right)$

Gradient of Encoder [5]:

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \phi} = -\mathbb{E}_{q_\phi(Z|X)} \left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi} \left(\log p_\theta(X, Z) - \log q_\phi(Z|X) \right) \right]$$

If Z is continuous, *reparameterization trick* is potentially another way to estimate the gradient!

For example, let us assume

$$\begin{aligned} q_\phi(Z|X) &= \mathcal{N}(Z|\mu, \sigma^2 I) \\ \mu &= \mathrm{EncoderNetwork}_\phi(X) \\ \log \sigma^2 &= \mathrm{EncoderNetwork}_\phi(X) \end{aligned}$$

Reparameterization Trick

For any function f, we have

$$
\begin{aligned}
\mathbb{E}_{\mathcal{N}(Z|\mu,\sigma^2 I)}\left[f(Z)\right] &= \int \frac{1}{\sqrt{(2\pi)^m \prod_i \sigma_i}} \exp\left(-\frac{1}{2}\left\|\frac{Z-\mu}{\sigma}\right\|^2\right) f(Z) dZ \\
&= \int \frac{1}{\sqrt{(2\pi)^m \prod_i \sigma_i}} \exp\left(-\frac{1}{2}\left\|\frac{\mu+\sigma\epsilon-\mu}{\sigma}\right\|^2\right) f(\mu+\sigma\epsilon) d\left(\mu+\sigma\epsilon\right) \\
&= \int \frac{1}{\sqrt{(2\pi)^m}} \exp\left(-\frac{1}{2}\left\|\epsilon\right\|^2\right) f(\mu+\sigma\epsilon) d\epsilon \\
&= \mathbb{E}_{\mathcal{N}(\epsilon|0,I)}\left[f(\mu+\sigma\epsilon)\right]
\end{aligned}
$$

Change of Variable

Reparameterization Trick

For any function f, we have

$$\begin{aligned}
\mathbb{E}_{\mathcal{N}(Z|\mu,\sigma^2 I)}\left[f(Z)\right] &= \int \frac{1}{\sqrt{(2\pi)^m}\prod_i \sigma_i} \exp(-\frac{1}{2}\left\|\frac{Z-\mu}{\sigma}\right\|^2) f(Z) dZ \\
&= \int \frac{1}{\sqrt{(2\pi)^m}\prod_i \sigma_i} \exp(-\frac{1}{2}\left\|\frac{\mu+\sigma\epsilon-\mu}{\sigma}\right\|^2) f(\mu+\sigma\epsilon) d\left(\mu+\sigma\epsilon\right) \\
&= \int \frac{1}{\sqrt{(2\pi)^m}} \exp(-\frac{1}{2}\left\|\epsilon\right\|^2) f(\mu+\sigma\epsilon) d\epsilon \\
&= \mathbb{E}_{\mathcal{N}(\epsilon|0,I)}\left[f(\mu+\sigma\epsilon)\right]
\end{aligned}$$

Change of Variable

Therefore,

$$\begin{aligned}
\mathcal{L}(\phi,\theta) &= \mathbb{E}_{q_\phi(Z|X)}\left[-\log\left(p_\theta(X|Z)\right)\right] + \mathrm{KL}\left(q_\phi(Z|X)\|p_\theta(Z)\right) \\
&= \mathbb{E}_{\mathcal{N}(\epsilon|0,I)}\left[-\log\left(p_\theta(X|\mu_\phi(X)+\sigma_\phi(X)\epsilon)\right)\right] + \mathrm{KL}\left(q_\phi(Z|X)\|p_\theta(Z)\right)
\end{aligned}$$

Reparameterization Trick

In original VAE,

$$q_\phi(Z|X) = \mathcal{N}(Z|\mu_\phi(X), \sigma_\phi(X)^2 I)$$
$$p_\theta(Z) = \mathcal{N}(X|0, I)$$

Reparameterization Trick

In original VAE,

$$q_\phi(Z|X) = \mathcal{N}(Z|\mu_\phi(X), \sigma_\phi(X)^2 I)$$
$$p_\theta(Z) = \mathcal{N}(X|0, I)$$

Using Gaussian integrals, we have

$$\mathrm{KL}\left(q_\phi(Z|X)\|p_\theta(Z)\right) = \frac{1}{2}\left(\mu_\phi(X)^\top \mu_\phi(X) + \sigma_\phi(X)^\top \sigma_\phi(X)\right) - \frac{1}{2}\sum_{i=1}^{m} \log \sigma_i^2 - \frac{m}{2}$$

where

$$\sigma_\phi(X) = [\sigma_1, \sigma_2, \cdots, \sigma_m]^\top$$

Reparameterization Trick

Therefore, in original VAE, we have

$$\begin{aligned}\mathcal{L}(\phi,\theta) = \ & \mathbb{E}_{\mathcal{N}(\epsilon|0,I)}\left[-\log\left(p_\theta(X|\mu_\phi(X)+\sigma_\phi(X)\epsilon)\right)\right] \\ & + \frac{1}{2}\left(\mu_\phi(X)^\top\mu_\phi(X)+\sigma_\phi(X)^\top\sigma_\phi(X)\right) - \frac{1}{2}\sum_{i=1}^{m}\log\sigma_i^2 - \frac{m}{2}\end{aligned}$$

Reparameterization Trick

Therefore, in original VAE, we have

$$\begin{aligned}\mathcal{L}(\phi,\theta) = \ & \mathbb{E}_{\mathcal{N}(\epsilon|0,I)}\left[-\log\left(p_\theta(X|\mu_\phi(X)+\sigma_\phi(X)\epsilon)\right)\right] \\ & + \frac{1}{2}\left(\mu_\phi(X)^\top\mu_\phi(X)+\sigma_\phi(X)^\top\sigma_\phi(X)\right) - \frac{1}{2}\sum_{i=1}^{m}\log\sigma_i^2 - \frac{m}{2}\end{aligned}$$

We only need *reparameterization trick* and *Monte Carlo estimation* in the first term

$$\begin{aligned}\mathcal{L}(\phi,\theta) \approx \ & -\sum_{i=1,\epsilon_i\sim\mathcal{N}(\epsilon|0,I)}^{N}\log\left(p_\theta(X|\mu_\phi(X)+\sigma_\phi(X)\epsilon_i)\right) \\ & + \frac{1}{2}\left(\mu_\phi(X)^\top\mu_\phi(X)+\sigma_\phi(X)^\top\sigma_\phi(X)\right) - \frac{1}{2}\sum_{i=1}^{m}\log\sigma_i^2 - \frac{m}{2}\end{aligned}$$

Reparameterization Trick

Therefore, in original VAE, we have

$$\begin{aligned}\mathcal{L}(\phi,\theta) = \ & \mathbb{E}_{\mathcal{N}(\epsilon|0,I)}\left[-\log\left(p_\theta(X|\mu_\phi(X)+\sigma_\phi(X)\epsilon)\right)\right] \\ & + \frac{1}{2}\left(\mu_\phi(X)^\top\mu_\phi(X)+\sigma_\phi(X)^\top\sigma_\phi(X)\right) - \frac{1}{2}\sum_{i=1}^{m}\log\sigma_i^2 - \frac{m}{2}\end{aligned}$$

We only need *reparameterization trick* and *Monte Carlo estimation* in the first term

$$\begin{aligned}\mathcal{L}(\phi,\theta) \approx \ & -\sum_{i=1,\epsilon_i\sim\mathcal{N}(\epsilon|0,I)}^{N}\log\left(p_\theta(X|\mu_\phi(X)+\sigma_\phi(X)\epsilon_i)\right) \\ & + \frac{1}{2}\left(\mu_\phi(X)^\top\mu_\phi(X)+\sigma_\phi(X)^\top\sigma_\phi(X)\right) - \frac{1}{2}\sum_{i=1}^{m}\log\sigma_i^2 - \frac{m}{2}\end{aligned}$$

Now we can get the gradient directly!

Reparameterization Trick

For continuous distributions other than Gaussian, can we still use *reparameterization trick*?

Reparameterization Trick

For continuous distributions other than Gaussian, can we still use *reparameterization trick*?

- Tractable inverse cumulative distribution function (CDF) [6]

  e.g., Exponential, Cauchy, Gumbel, Erlang, …

- Location-scale family [6]

  e.g., Laplace, Elliptical, Student's t, …

- Composition [6]

  e.g., Log-Normal, Beta, Chi-Squared, …

- Implicit differentiation [16]

  e.g., Dirichlet, Von-Mises, Mixture, …

Reparameterization Trick vs. REINFORCE

Let us simplify the context to better compare them

Suppose we want to optimize $\mathcal{L}(\phi) = \mathbb{E}_{p_\phi(X)}[f(X)]$ where $p_\phi(X) = \mathcal{N}(X|\phi, I)$

Reparameterization Trick vs. REINFORCE

Let us simplify the context to better compare them

Suppose we want to optimize $\mathcal{L}(\phi) = \mathbb{E}_{p_\phi(X)}[f(X)]$ where $p_\phi(X) = \mathcal{N}(X|\phi, I)$

REINFORCE gradient estimator:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \phi} &= \mathbb{E}_{p_\phi(X)}\left[\frac{\partial \log p_\phi(X)}{\partial \phi} f(X)\right] \\ &= \mathbb{E}_{\mathcal{N}(X|\phi,I)}\left[(X-\phi)f(X)\right]\end{aligned}$$

Reparameterization Trick vs. REINFORCE

Let us simplify the context to better compare them

Suppose we want to optimize $\mathcal{L}(\phi) = \mathbb{E}_{p_\phi(X)}[f(X)]$ where $p_\phi(X) = \mathcal{N}(X|\phi, I)$

REINFORCE gradient estimator:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \phi} &= \mathbb{E}_{p_\phi(X)}\left[\frac{\partial \log p_\phi(X)}{\partial \phi} f(X)\right] \\ &= \mathbb{E}_{\mathcal{N}(X|\phi,I)}\left[(X-\phi)f(X)\right]\end{aligned}$$

Reparameterization gradient estimator:

$$\frac{\partial \mathcal{L}}{\partial \phi} = \mathbb{E}_{\mathcal{N}(\epsilon|0,I)}\left[\frac{\partial f(\phi+\epsilon)}{\partial \phi}\right]$$

Reparameterization Trick vs. REINFORCE

Let us simplify the context to better compare them

Suppose we want to optimize $\mathcal{L}(\phi) = \mathbb{E}_{p_\phi(X)}[f(X)]$ where $p_\phi(X) = \mathcal{N}(X|\phi, I)$

REINFORCE gradient estimator:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \phi} &= \mathbb{E}_{p_\phi(X)}\left[\frac{\partial \log p_\phi(X)}{\partial \phi} f(X)\right] \\ &= \mathbb{E}_{\mathcal{N}(X|\phi,I)}\left[(X-\phi)f(X)\right]\end{aligned}$$

Reparameterization gradient estimator:

$$\frac{\partial \mathcal{L}}{\partial \phi} = \mathbb{E}_{\mathcal{N}(\epsilon|0,I)}\left[\frac{\partial f(\phi+\epsilon)}{\partial \phi}\right]$$

They are the same in expectation (unbiased)! Why?

Reparameterization Trick vs. REINFORCE

REINFORCE gradient estimator:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \phi} &= \mathbb{E}_{p_\phi(X)}\left[\frac{\partial \log p_\phi(X)}{\partial \phi} f(X)\right] \\ &= \mathbb{E}_{\mathcal{N}(X|\phi,I)}\left[(X-\phi) f(X)\right]\end{aligned}$$

Reparameterization gradient estimator:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \phi} &= \mathbb{E}_{\mathcal{N}(\epsilon|0,I)}\left[\frac{\partial f(\phi+\epsilon)}{\partial \phi}\right] \\ &= \mathbb{E}_{\mathcal{N}(X|\phi,I)}\left[\frac{\partial f(X)}{\partial X}\right]\end{aligned}$$

Reparameterization Trick vs. REINFORCE

REINFORCE gradient estimator:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \phi} &= \mathbb{E}_{p_\phi(X)}\left[\frac{\partial \log p_\phi(X)}{\partial \phi} f(X)\right] \\ &= \mathbb{E}_{\mathcal{N}(X|\phi,I)}\left[(X-\phi) f(X)\right]\end{aligned}$$

Reparameterization gradient estimator:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \phi} &= \mathbb{E}_{\mathcal{N}(\epsilon|0,I)}\left[\frac{\partial f(\phi+\epsilon)}{\partial \phi}\right] \\ &= \mathbb{E}_{\mathcal{N}(X|\phi,I)}\left[\frac{\partial f(X)}{\partial X}\right]\end{aligned}$$

We have $$\frac{\partial \mathcal{L}}{\partial X} = \frac{\partial}{\partial X}\mathbb{E}_{p_\phi(X)}\left[f(X)\right] = \mathbb{E}_{\frac{\partial}{\partial X}p_\phi(X)}\left[f(X)\right] + \mathbb{E}_{p_\phi(X)}\left[\frac{\partial f(X)}{\partial X}\right] = 0$$

Reparameterization Trick vs. REINFORCE

REINFORCE gradient estimator:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \phi} &= \mathbb{E}_{p_\phi(X)}\left[\frac{\partial \log p_\phi(X)}{\partial \phi} f(X)\right] \\ &= \mathbb{E}_{\mathcal{N}(X|\phi,I)}\left[(X-\phi)f(X)\right]\end{aligned}$$

Reparameterization gradient estimator:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \phi} &= \mathbb{E}_{\mathcal{N}(\epsilon|0,I)}\left[\frac{\partial f(\phi+\epsilon)}{\partial \phi}\right] \\ &= \mathbb{E}_{\mathcal{N}(X|\phi,I)}\left[\frac{\partial f(X)}{\partial X}\right]\end{aligned}$$

We have

$$\frac{\partial \mathcal{L}}{\partial X} = \frac{\partial}{\partial X}\mathbb{E}_{p_\phi(X)}\left[f(X)\right] = \mathbb{E}_{\frac{\partial}{\partial X}p_\phi(X)}\left[f(X)\right] + \mathbb{E}_{p_\phi(X)}\left[\frac{\partial f(X)}{\partial X}\right] = 0$$

Hence

$$\begin{aligned}\mathbb{E}_{p_\phi(X)}\left[\frac{\partial f(X)}{\partial X}\right] &= -\mathbb{E}_{p_\phi(X)}\left[\frac{\partial \log p_\phi(X)}{\partial X} f(X)\right] \\ &= \mathbb{E}_{\mathcal{N}(X|\phi,I)}\left[(X-\phi)f(X)\right]\end{aligned}$$

Reparameterization Trick vs. REINFORCE

Let us simplify the context to better compare them

Suppose we want to optimize $\mathcal{L}(\phi) = \mathbb{E}_{p_\phi(X)}[f(X)]$ where $p_\phi(X) = \mathcal{N}(X|\phi, I)$

REINFORCE gradient estimator:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \phi} &= \mathbb{E}_{p_\phi(X)}\left[\frac{\partial \log p_\phi(X)}{\partial \phi} f(X)\right] \\ &= \mathbb{E}_{\mathcal{N}(X|\phi,I)}\left[(X-\phi)f(X)\right]\end{aligned}$$

Reparameterization gradient estimator:

$$\frac{\partial \mathcal{L}}{\partial \phi} = \mathbb{E}_{\mathcal{N}(\epsilon|0,I)}\left[\frac{\partial f(\phi+\epsilon)}{\partial \phi}\right]$$

They are the same in expectation (unbiased)!

Reparameterization is often found to have lower variance than REINFORCE empirically

Reparameterization Trick vs. REINFORCE

Let us simplify the context to better compare them

Suppose we want to optimize $\mathcal{L}(\phi) = \mathbb{E}_{p_\phi(X)}[f(X)]$ where $p_\phi(X) = \mathcal{N}(X|\phi, I)$

REINFORCE gradient estimator:

$$\frac{\partial \mathcal{L}}{\partial \phi} = \mathbb{E}_{p_\phi(X)}\left[\frac{\partial \log p_\phi(X)}{\partial \phi} f(X)\right]$$
$$= \mathbb{E}_{\mathcal{N}(X|\phi,I)}[(X-\phi)f(X)]$$

Reparameterization gradient estimator:

$$\frac{\partial \mathcal{L}}{\partial \phi} = \mathbb{E}_{\mathcal{N}(\epsilon|0,I)}\left[\frac{\partial f(\phi+\epsilon)}{\partial \phi}\right]$$

They are the same in expectation!

Gradient of reward function is leveraged!

Reparameterization is often found to have lower variance than REINFORCE empirically

Yet Another “Gradient Estimator”

Wake-Sleep Algorithm [9]

Wake Phase: Update Decoder as before

$$\frac{\partial \mathcal{L}(\phi, \theta)}{\partial \theta} = \mathbb{E}_{q_\phi(Z|X)} \left[-\frac{\partial \log \left(p_\theta(X|Z) \right)}{\partial \theta} \right] \approx -\frac{1}{N} \sum_{\substack{i=1 \\ Z_i \sim q_\phi(Z|X)}}^{N} \frac{\partial \log \left(p_\theta(X|Z_i) \right)}{\partial \theta}$$

Yet Another “Gradient Estimator”

Wake-Sleep Algorithm [9]

Wake Phase: Update Decoder as before

$$\frac{\partial \mathcal{L}(\phi,\theta)}{\partial \theta} = \mathbb{E}_{q_\phi(Z|X)}\left[-\frac{\partial \log\left(p_\theta(X|Z)\right)}{\partial \theta}\right] \approx -\frac{1}{N}\sum_{\substack{i=1 \\ Z_i \sim q_\phi(Z|X)}}^{N} \frac{\partial \log\left(p_\theta(X|Z_i)\right)}{\partial \theta}$$

Sleep Phase: Update “Encoder Gradient” as

$$\frac{\partial \mathcal{L}(\phi,\theta)}{\partial \phi} = -\mathbb{E}_{p_\theta(X,Z)}\left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi}\right]$$

Yet Another “Gradient Estimator”

Wake-Sleep Algorithm [9]

Wake Phase: Update Decoder as before

$$\frac{\partial \mathcal{L}(\phi,\theta)}{\partial \theta} = \mathbb{E}_{q_\phi(Z|X)}\left[-\frac{\partial \log\left(p_\theta(X|Z)\right)}{\partial \theta}\right] \approx -\frac{1}{N}\sum_{\substack{i=1 \\ Z_i \sim q_\phi(Z|X)}}^{N} \frac{\partial \log\left(p_\theta(X|Z_i)\right)}{\partial \theta}$$

Sleep Phase: Update “Encoder Gradient” as

$$\frac{\partial \mathcal{L}(\phi,\theta)}{\partial \phi} = -\mathbb{E}_{p_\theta(X,Z)}\left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi}\right]$$

Pros: easier to implement

Cons: sleep phase does not necessarily maximize ELBO

Yet Another "Gradient Estimator"

Wake-Sleep Algorithm [9]

Wake Phase: Update Decoder as before

$$\frac{\partial \mathcal{L}(\phi,\theta)}{\partial \theta} = \mathbb{E}_{q_\phi(Z|X)}\left[-\frac{\partial \log\left(p_\theta(X|Z)\right)}{\partial \theta}\right] \approx -\frac{1}{N}\sum_{\substack{i=1 \\ Z_i \sim q_\phi(Z|X)}}^{N} \frac{\partial \log\left(p_\theta(X|Z_i)\right)}{\partial \theta}$$

Sleep Phase: Update "Encoder Gradient" as

$$\frac{\partial \mathcal{L}(\phi,\theta)}{\partial \phi} = -\mathbb{E}_{p_\theta(X,Z)}\left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi}\right]$$

Pros: easier to implement

Cons: sleep phase does not necessarily maximize ELBO

True Gradient of ELBO

$$\frac{\partial \mathcal{L}(\phi,\theta)}{\partial \phi} = -\mathbb{E}_{q_\phi(Z|X)}\left[\frac{\partial \log q_\phi(Z|X)}{\partial \phi}\left(\log p_\theta(X,Z) - \log q_\phi(Z|X)\right)\right]$$

References

[1] Smolensky, P., 1986. Information processing in dynamical systems: Foundations of harmony theory. Colorado Univ at Boulder Dept of Computer Science.

[2] Hinton, G.E. and Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. science, 313(5786), pp.504-507.

[3] Hinton, G.E., 2002. Training products of experts by minimizing contrastive divergence. Neural computation, 14(8), pp.1771-1800.

[4] Salakhutdinov, R. and Larochelle, H., 2010, March. Efficient learning of deep Boltzmann machines. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 693-700). JMLR Workshop and Conference Proceedings.

[5] Mnih, A. and Gregor, K., 2014, June. Neural variational inference and learning in belief networks. In International Conference on Machine Learning (pp. 1791-1799). PMLR.

[6] Kingma, D.P. and Welling, M., 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

[7] Williams, R.J., 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning, 8(3), pp.229-256.

[8] Greensmith, E., Bartlett, P.L. and Baxter, J., 2004. Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning. Journal of Machine Learning Research, 5(9).

[9] Hinton, G.E., Dayan, P., Frey, B.J. and Neal, R.M., 1995. The" wake-sleep" algorithm for unsupervised neural networks. Science, 268(5214), pp.1158-1161.

[10] Hinton, G.E. and Zemel, R., 1993. Autoencoders, minimum description length and Helmholtz free energy. Advances in neural information processing systems, 6.

References

[11] Dayan, P., Hinton, G.E., Neal, R.M. and Zemel, R.S., 1995. The helmholtz machine. Neural computation, 7(5), pp.889-904.

[12] Mohamed, S., Rosca, M., Figurnov, M. and Mnih, A., 2020. Monte Carlo Gradient Estimation in Machine Learning. J. Mach. Learn. Res., 21(132), pp.1-62.

[13] Owen, A.B., 2013. Monte Carlo theory, methods and examples.

[14] Tucker, G., Mnih, A., Maddison, C.J., Lawson, J. and Sohl-Dickstein, J., 2017. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. Advances in Neural Information Processing Systems, 30.

[15] Grathwohl, W., Choi, D., Wu, Y., Roeder, G. and Duvenaud, D., 2017. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. arXiv preprint arXiv:1711.00123.

[16] Figurnov, M., Mohamed, S. and Mnih, A., 2018. Implicit reparameterization gradients. Advances in Neural Information Processing Systems, 31.

Questions?