Importance Weighted Autoencoders (IWAE)

Presenter: Huanyu Zhang

Department of Statistics
University of Chicago

October 31, 2023

Overview

The Limitations of VAE[KW14]

The limitations of VAE:

- It typically makes strong assumptions about posterior inference, for instance that the posterior distribution is approximately factorial, and that its parameters can be approximated with nonlinear regression from the observations.
- The VAE objective can lead to overly simplified representations which fail to use the network's entire modeling capacity.

Exemplary Assumptions of Vanilla VAE

1. The prior $p_\theta(\mathbf{z})$ follows a centered isotropic multivariate Gaussian $\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$.
2. The likelihood $p_\theta(\mathbf{x}|\mathbf{z})$ is a multivariate Gaussian. Its distribution parameters are computed from $\mathbf{z}$ using an MLP.
3. The authors assumes that the true (but intractable) posterior approximates a Gaussian form with a nearly diagonal covariance, and so is the variational approximate posterior:

$$\log q_\phi\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right) = \log \mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}\mathbf{I}\right).$$

The Variance Lower Bound Revisited

The marginal likelihood is composed of a sum over the marginal likelihoods of individual datapoints $\log p_{\theta}\left(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)}\right) = \sum_{i=1}^{N} \log p_{\theta}\left(\mathbf{x}^{(i)}\right)$, which can each be rewritten as:

$$\log p_{\theta}\left(\mathbf{x}^{(i)}\right) = D_{KL}\left(q_{\phi}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right) \| p_{\theta}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right)\right) + \mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}\right)$$

The second RHS term $\mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}\right)$ is called the (variational) lower bound on the marginal likelihood of datapoint i, and can be written as:

$$\mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}\right) = -D_{KL}\left(q_{\phi}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right) \| p_{\theta}(\mathbf{z})\right) + \mathbb{E}_{q_{\phi}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right)}\left[\log p_{\theta}\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right)\right]$$

The Introduction of IWAE[BGS15]

The introduction of IWAE:

- A generative model with the same architecture as the VAE, but which uses a strictly tighter log-likelihood lower bound (compared to ELBO) derived from importance weighting.
- In the IWAE, the recognition network uses multiple samples to approximate the posterior, giving it increased flexibility to model complex posteriors which do not fit the VAE modeling assumptions.

The architecture of VAE(IWAE):

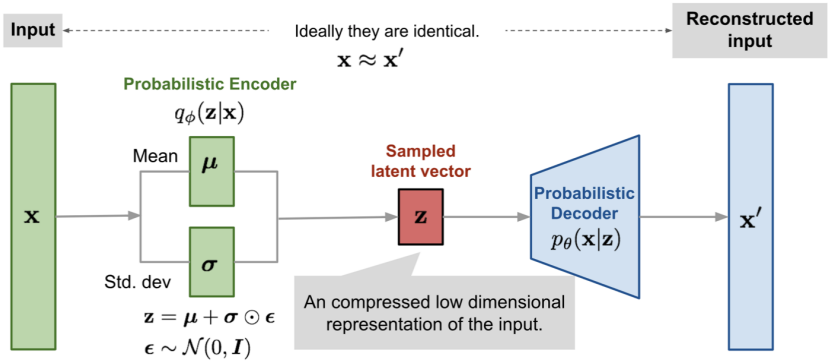

Figure: Illustration of VAE model architecture[**weng**]

Problem Scenario

Let us consider some dataset $\mathbf{X} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N}$ consisting of N i.i.d. samples of some continuous or discrete variable $\mathbf{x}$.

We assume:

- Data are generated by a process involving an unobserved variable $\mathbf{z}$.
- Two steps: (1) Generate $\mathbf{z}^{(i)}$ from $p_{\boldsymbol{\theta}^*}(\mathbf{z})$; (2) Generate $\mathbf{x}^{(i)}$ from $p_{\boldsymbol{\theta}^*}(\mathbf{x} \mid \mathbf{z})$.
- Distributions are from parametric families.
- PDFs are differentiable almost everywhere.
- Unknown: true parameters $\boldsymbol{\theta}^*$ and latent variables $\mathbf{z}^{(i)}$.

Intractability and Large Datasets

We do not make simplifying assumptions:

1. **Intractability:**
   - Marginal likelihood $p_{\theta}(\mathbf{x})$ is intractable.
   - True posterior density $p_{\theta}(\mathbf{z} \mid \mathbf{x})$ is intractable.
   - EM algorithm cannot be used.
   - Appears in cases like neural networks with nonlinear hidden layers.
2. **Large Dataset:**
   - Batch optimization costly.
   - Need updates with small minibatches or single points.
   - Monte Carlo EM too slow.

A Different (Strictly Tighter) Lower Bound(1)

The IWAE is trained to maximize a different lower bound on $\log p(\mathbf{x})$, which is

$$\mathcal{L}_k(\mathbf{x}) = \mathbb{E}_{\mathbf{z}_1,\ldots,\mathbf{z}_k \sim q(\mathbf{z}|\mathbf{x})} \left[\log \frac{1}{k} \sum_{i=1}^{k} \frac{p(\mathbf{x}, \mathbf{z}_i)}{q(\mathbf{z}_i \mid \mathbf{x})} \right].$$

Here, $\mathbf{z}_1, \ldots, \mathbf{z}_k$ are sampled independently from the recognition model. The wm inside the sum corresponds to the unnormalized importance weights for the joint distribution, which we will denote as $w_i = p(\mathbf{x}, \mathbf{z}_i) / q(\mathbf{z}_i \mid \mathbf{x})$.
This is a lower bound on the marginal log-likelihood, as follows from Jensen's Inequality:

$$\mathcal{L}_k = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \left[\log \frac{1}{k} \sum_{i=1}^{k} w_i \right] \leq \log \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \left[\frac{1}{k} \sum_{i=1}^{k} w_i \right] = \log p(\mathbf{x}),$$

.

A Different (Strictly Tighter) Lower Bound(2)

Theorem

For all k, the lower bounds satisfy

$$\log p(\mathbf{x}) \geq \mathcal{L}_{k+1} \geq \mathcal{L}_k.$$

Moreover, if $\frac{p(\mathbf{z},\mathbf{x})}{q(\mathbf{z}|\mathbf{x})}$ is bounded, then $\mathcal{L}_k$ approaches $\log p(\mathbf{x})$ as the number of samples k goes to infinity.

Estimation of $\mathcal{L}_k$

The bound $\mathcal{L}_k$ can be estimated using the straightforward Monte Carlo estimator, and so are its gradients.

Training Procedure(1)

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_k(\mathbf{x}) = \nabla_{\boldsymbol{\theta}}\mathbb{E}_{\mathbf{z}_1,\ldots,\mathbf{z}_k}\left[\log\frac{1}{k}\sum_{i=1}^{k} w_i\right] = \nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_k}\left[\log\frac{1}{k}\sum_{i=1}^{k} w\left(\mathbf{x},\mathbf{z}\left(\mathbf{x},\boldsymbol{\epsilon}_i,\boldsymbol{\theta}\right),\boldsymbol{\theta}\right)\right]$$

$$= \mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_k}\left[\nabla_{\boldsymbol{\theta}}\log\frac{1}{k}\sum_{i=1}^{k} w\left(\mathbf{x},\mathbf{z}\left(\mathbf{x},\boldsymbol{\epsilon}_i,\boldsymbol{\theta}\right),\boldsymbol{\theta}\right)\right]$$

$$= \mathbb{E}_{\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_k}\left[\sum_{i=1}^{k}\widetilde{w}_i\nabla_{\boldsymbol{\theta}}\log w\left(\mathbf{x},\mathbf{z}\left(\mathbf{x},\boldsymbol{\epsilon}_i,\boldsymbol{\theta}\right),\boldsymbol{\theta}\right)\right],$$

where $\epsilon_1,\ldots,\epsilon_k$ are the same auxiliary variables, $w_i = w\left(\mathbf{x},\mathbf{z}\left(\mathbf{x},\boldsymbol{\epsilon}_i,\boldsymbol{\theta}\right),\boldsymbol{\theta}\right)$ are the importance weights expressed as a deterministic function, and $\widetilde{w}_i = w_i/\sum_{i=1}^{k} w_i$ are the normalized importance weights.

Training Procedure(2)

In the context of a gradient-based learning algorithm, we draw k samples from the recognition network (or, equivalently, k sets of auxiliary variables), and use the Monte Carlo estimate of

$$\sum_{i=1}^{k} \widetilde{w}_i \nabla_{\boldsymbol{\theta}} \log w\left(\mathbf{x}, \mathbf{z}\left(\boldsymbol{\epsilon}_i, \mathbf{x}, \boldsymbol{\theta}\right), \boldsymbol{\theta}\right).$$

Training Procedure(3)

We unpack this update because it does not quite parallel that of the standard VAE. The gradient of the log weights decomposes as:

$$\nabla_{\boldsymbol{\theta}} \log w\left(\mathbf{x}, \mathbf{z}\left(\mathbf{x}, \boldsymbol{\epsilon}_i, \boldsymbol{\theta}\right), \boldsymbol{\theta}\right) = \nabla_{\boldsymbol{\theta}} \log p\left(\mathbf{x}, \mathbf{z}\left(\mathbf{x}, \boldsymbol{\epsilon}_i, \boldsymbol{\theta}\right) \mid \boldsymbol{\theta}\right) - \nabla_{\boldsymbol{\theta}} \log q\left(\mathbf{z}\left(\mathbf{x}, \boldsymbol{\epsilon}_i, \boldsymbol{\theta}\right) \mid \mathbf{x}, \boldsymbol{\theta}\right).$$

The first term (generative model gradient) encourages the generative model to assign high probability to the likelihood. The second term (recognition model gradient) also encourages to produce representations that can be easily sampled and provide a concise summary of the data.

IWAE Algorithm (SGD)

Algorithm 1 IWAE Training Procedure (SGD)

Input: Dataset $\mathcal{D}$, encoder $q_\phi(z|x)$, decoder $p_\theta(x|z)$, num of samples k
for each data point x_i in $\mathcal{D}$ **do**
    for each $1 \leq j \leq k$ **do**
        Sample: $z_{ij} \sim q_\phi(z|x_i)$ and compute: $w_{ij} \propto \frac{p_\theta(x_i|z_{ij})p(z_{ij})}{q_\phi(z_{ij}|x_i)}$
    end for
    Average weights: $w_i \leftarrow \frac{1}{k}\sum_{j=1}^{k} w_{ij}$ and compute the loss $= -\log w_i$
    Update parameters ϕ and θ using SGD on the computed loss
end for

Discussions on Computational Costs

The dominant computational cost in IWAE training is computing the activations and parameter gradients needed for $\nabla_{\boldsymbol{\theta}} \log w\left(\mathbf{x}, \mathbf{z}\left(\mathbf{x}, \boldsymbol{\epsilon}_i, \boldsymbol{\theta}\right), \boldsymbol{\theta}\right)$. This corresponds to the forward and backward passes in backpropagation. In the basic IWAE implementation, both passes must be done independently for each of the k samples. Therefore, the number of operations scales linearly with k.

Experiments

We conducted experiments for IWAE on the MNIST dataset. The metrics for evaluation include:

- The Frechet Inception Distance (FID).
- Negative Log Likelihood $-\log p_\theta(\mathbf{X})$

Reference

[KW14] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: (2014). Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1312.6114.

[BGS15] Yuri Burda, Roger Baker Grosse, and Ruslan Salakhutdinov. "Importance Weighted Autoencoders". In: *CoRR* abs/1509.00519 (2015). URL: https://api.semanticscholar.org/CorpusID:11383178.

Importance Weighted Monte Carlo: Concepts

- **Definition**:
  - A Monte Carlo technique where samples are drawn from a different distribution than the target, and then reweighted to account for this difference.
- **Why Use IWMC?**:
  - **Efficiency**: Targets regions of high importance to reduce variance.
  - **Flexibility**: Allows sampling from convenient or known distributions.
- **Formula**:
  - Weight: $w(x) = \frac{p(x)}{q(x)}$
    - $p(x)$: Target distribution
    - $q(x)$: Proposal distribution from which we sample
    - Note that the support of $q(x)$ should contain the support of $p(x)$.
  - Estimation: $\mathbb{E}_{x\sim p}[f(x)] = \mathbb{E}_{x\sim q}[w(x)f(x)] \approx \sum_{i=1}^{k} w_i f(x_i)$, where k instances of x_i are i.i.d. sampled from the distribution $q(x)$.

Importance Weighted Monte Carlo in Action

Estimating an Integral

- **Objective**:
  - Estimate the integral $I = \int_0^1 e^{-x^2} dx$
- **Standard Monte Carlo**:
  - Sample x_i from Uniform(0, 1).
  - Estimate: $\hat{I} = \frac{1}{N} \sum_{i=1}^{N} e^{-x_i^2}$
- **IWMC**:
  - **Proposal Distribution**: $q(x) = 3x^2$ (Biased towards the peak of the integrand)
  - Sample x_i from $q(x)$.
  - Compute weights: $w(x_i) = \frac{1}{3x_i^2}$
  - Estimate: $\hat{I}_{IWMC} = \frac{1}{N} \sum_{i=1}^{N} w(x_i) e^{-x_i^2}$