Learning Deep Generative Models with Annealed Importance Sampling

Xinqiang Ding [*]
Department of Neurobiology
The University of Chicago
Chicago, IL 60637 United States

David J. Freedman
Department of Neurobiology and
The Grossman Institute for Neuroscience, Quantitative Biology and Human Behavior
The University of Chicago
Chicago, IL 60637 United States

Abstract

Variational inference (VI) and Markov chain Monte Carlo (MCMC) are two main approximate approaches for learning deep generative models by maximizing marginal likelihood. In this paper, we propose using annealed importance sampling, which is equivalent to the Jarzynski equality from non-equilibrium thermodynamics, for learning deep generative models. Our proposed approach bridges VI with MCMC. It generalizes VI methods such as variational auto-encoders and importance weighted auto-encoders (IWAE) and the MCMC method proposed in [1]. It also provides insights into why running multiple short MCMC chains can help learning deep generative models. Through experiments, we show that our approach yields better density models than IWAE and can effectively trade computation for model accuracy without increasing memory cost.

1 Introduction

Deep generative models with latent variables are powerful probabilistic models for high dimensional data. One of the challenges for learning such models by maximizing marginal likelihood is sampling from posterior distributions of latent variables, because the posterior distributions are often intractable and have complex dependency structures. Two main approximate approaches for learning such models are variational inference (VI) [2, 3] and Markov chain Monte Carlo (MCMC) [4].

Here let us assume the generative model of interest is defined by a joint distribution of observed data variables x and continuous latent variables z: $p_\theta(x,z) = p_\theta(z)p_\theta(x|z)$, where θ represents parameters of the generative model. Given training data x, we are interested in learning the generative model $p_\theta(x,z)$ by maximizing its marginal likelihood, i.e., maximizing $\log p_\theta(x) = \log \int p_\theta(x,z)dz$. Because $\log p_\theta(x)$ is usually a high dimensional integration when z is high-dimensional and has complex dependence structure between its components, computing $\log p_\theta(x)$ is expensive. However, we note that maximizing $\log p_\theta(x)$ does not necessarily require computing the value of $\log p_\theta(x)$. If we use first order optimization methods for training, what is necessarily required is the gradient of $\log p_\theta(x)$ with respect to θ, i.e.,

$$\nabla_\theta \log p_\theta(x) = \mathop{\mathbb{E}}_{z\sim p_\theta(z|x)} \left[\nabla_\theta \log p_\theta(x,z)\right]. \tag{1}$$

[*] xqding@umich.edu

Third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020), Vancouver, Canada.

As shown in (1), computing $\nabla_\theta \log p_\theta(x)$ is equivalent to calculating the expectation of $\nabla_\theta \log p_\theta(x,z)$ with respect to the posterior distribution $p_\theta(z|x)$. Because the posterior distribution $p_\theta(z|x)$ is usually analytically intractable and it is often even difficult to draw samples from it, accurately computing $\nabla_\theta \log p_\theta(x)$ based on Eq. (1) is computationally expensive. Therefore, efficient approximation methods are required to estimate the expectation in (1).

Several VI methods, including variational auto-encoders (VAEs) [5, 6] and importance weighted auto-encoders (IWAEs) [7] can be viewed/understood as devising ways of approximating the expectation in (1). VAEs use an amortized inference model $q_\phi(z|x)$ to approximate the posterior distribution $p_\theta(z|x)$. Both the generative model and the inference model are learned by maximizing the evidence lower bound (ELBO):

$$\mathcal{L}(\theta,\phi) = \mathop{\mathbb{E}}_{z\sim q_\phi(z|x)}[\log p_\theta(x,z) - \log q_\phi(z|x)]. \tag{2}$$

In maximizing ELBO, θ follows the gradient

$$\nabla_\theta \mathcal{L}(\theta,\phi) = \mathop{\mathbb{E}}_{z\sim q_\phi(z|x)}\big[\nabla_\theta \log p_\theta(x,z)\big]. \tag{3}$$

In VAEs, the expectation in (3) is usually estimated by drawing one sample from $q_\phi(z|x)$. If we view VAEs as an approximate method for maximum likelihood learning, comparing (3) to (1), we note that VAEs can be viewed as using the expectation in (3) to approximate the expectation in (1) using one sample from $q_\phi(z|x)$. The rationale behind the approximation is that the inference model $q_\phi(z|x)$ is optimized to approximate the posterior distribution $p_\theta(z|x)$. However, when the inference model can not approximate the posterior distribution well due to either its limited expressibility or issues in optimizing the inference model, the estimator based on (3) will have large bias with respect to the target expectation in (1). A canonical approach to reduce the bias in (3) is using multiple samples with importance sampling [8]. Specifically, we can draw K independent samples, $z_1, ..., z_K$, from $q_\phi(z|x)$. Then the expectation in (1) is estimated using

$$\sum_{k=1}^{K} \widetilde{w_k} \nabla_\theta \log p_\theta(x, z_k) \tag{4}$$

where $\widetilde{w_k} = w_k / \sum_{i=1}^{K} w_i$ and $w_k = w(x, z_k, \theta, \phi) = p_\theta(x, z_k)/q_\phi(z_k|x)$. The same gradient estimator as (4) was used in IWAEs, but in IWAEs the estimator was devised as the gradient of a new ELBO function based on multiple (K) samples [7]:

$$\mathcal{L}_K(\theta,\phi) = \mathop{\mathbb{E}}_{z_1,\ldots,z_K\sim q_\phi(z|x)}\left[\log \frac{1}{K}\sum_{k=1}^{K}\frac{p_\theta(x,z_k)}{q_\phi(z_k|x)}\right]. \tag{5}$$

It is easy to verify that the estimator in (4) is an unbiased estimator of $\nabla_\theta \mathcal{L}_K(\theta,\phi)$. Again, if we view IWAEs as an approximate method for maximum likelihood learning, IWAEs can be viewed as using the estimator in (4) to approximate the expectation in (1).

MCMC approaches [4] aim to sample from posterior distributions by running a Markov chain with posterior distributions as its stationary distributions. Because it could take a large number of steps for a Markov chain to converge, MCMC approaches were known as much slower than VI and were not as widely used as VI approaches for learning deep generative models, especially on large datasets. Compared with rapid developments of VI approaches in recent years, relatively fewer studies investigate the use of MCMC approaches for learning deep generative models [1, 9].

2 Method

In this work, we propose using annealed importance sampling (AIS) [10] (equivalent to the Jarzynski equality [11] from non-equilibrium thermodynamics) to estimate the expectation in (1). As a generalization of importance sampling, AIS [11, 10] uses samples from a sequence of distributions that bridge an initial tractable distribution, which is $q_\phi(z|x)$ in our case, with a final target distribution, which is $p_\theta(z|x) \propto p_\theta(x,z)$. To bridge $q_\phi(z|x)$ with $p_\theta(z|x) \propto p_\theta(x,z)$, we construct a sequence of intermediate distributions whose probability densities are proportional to $f_1(z), ..., f_{T-1}(z)$ and

$$f_t(z) = f_0(z)^{1-\beta_t} f_T(z)^{\beta_t}, \tag{6}$$

where $f_0(z) = q_\phi(z|x), f_T(z) = p_\theta(x, z)$. β_t are inverse temperatures and satisfy the condition $0 = \beta_0 \le ... \le \beta_T = 1$. To estimate the expectation in (1), we can generate K samples $\{z_1, ..., z_K\}$ and compute their corresponding weights $\{w_1, ..., w_K\}$ as follows. To generate the kth sample z_k and calculate its weight w_k, a sequence of samples $\{z_k^0, ..., z_k^T\}$ is generated using the following procedure. Initially, z_k^0 is generated by sampling from the distribution $f_0(z) = q_\phi(z|x)$. Here $q_\phi(z|x)$ is learned by optimizing the ELBO (2) as in VAEs. For $1 \le t \le T$, z_k^t is generated using a reversible transition kernel $T_t(z|z_k^{t-1})$ that keeps f_t invariant. The transition kernel $T_t(z|z_k^{t-1})$ is constructed using the Hamiltonian Monte Carlo (HMC) sampling method [12] in which the potential energy function $U_t(z)$ is set to $U_t(z) = -\log f_t(z)$. After T steps the sample z_k is set to $z_k = z_k^T$ and its weight w_k is calculated as:

$$w_k = \frac{f_1(z^0)}{f_0(z^0)} \frac{f_2(z^1)}{f_1(z^1)} \cdots \frac{f_{T-1}(z^{T-2})}{f_{T-2}(z^{T-2})} \frac{f_T(z^{T-1})}{f_{T-1}(z^{T-1})}. \tag{7}$$

With the generated K samples $z_1, ..., z_K$ and their weights $w_1, ..., w_K$, the expectation in (1) is estimated using

$$\nabla_\theta \log p_\theta(x) = \mathop{\mathbb{E}}_{z \sim p_\theta(z|x)} \left[\nabla_\theta \log p_\theta(x, z) \right] \simeq \sum_{k=1}^{K} \widetilde{w_k} \nabla_\theta \log p_\theta(x, z_k), \tag{8}$$

where $\widetilde{w_k} = w_k / \sum_{i=1}^{K} w_i$ are normalized weights. In summary, the detailed procedures of our proposed method are described in Algorithm (1).

Algorithm 1: Learning Deep Generative Models with Annealed Importance Sampling

```
  Require:
  $x$: training data
  $K$: the number of annealed importance weighted samples
  $p_\theta(x, z)$: the generative model
  $q_\phi(z|x)$: the inference model
  $T$: the number of inverse temperatures
  $\{\beta_t : 0 = \beta_0 \le ... \le \beta_T = 1\}$: inverse temperatures for the generative model
  $\{\epsilon_t : t = 1, ..., T\}$: the step sizes used in leapfrog integration of HMC at each inverse temperature
  $L$: the number of integration steps in HMC
  Calculate Gradients and Optimize Parameters:
  while $\theta, \phi$ not converged do
      sample example(s) $x$ from the training data;
      update the generative model parameter $\theta$
      set $\log w_k = 0$ for $k = 1, ..., K$;
      sample $z^0 = [z_1^0, z_2^0, ..., z_K^0]$, where $z_k^0$ are i.i.d. samples from $q_\phi(z|x)$;
      $\log w_k \leftarrow (\beta_1 - \beta_0)[\log p_\theta(x, z_k^0) - \log q_\phi(z_k^0)]$ for $k = 1, ..., K$;
      for $t \leftarrow 1$ to $T - 1$ do
          $z^t$ = HMC$(z^{t-1}, \beta_t, \epsilon_t, L)$, where the potential energy function is: $U_t(z) = -\log f_t(z)$ and
          $f_t(z) = q_\phi(z|x)^{1-\beta_t} p_\theta(x, z)^{\beta_t}$;
          $\log w_k \leftarrow \log w_k + (\beta_t - \beta_{t-1})[\log p_\theta(x, z_k^{t-1}) - \log q_\phi(z_k^{t-1})]$ for $k = 1, ..., K$;
      end for
      $z^T$ = HMC$(z^{T-1}, \beta_T, \epsilon_T, L)$ or set $z^T = z^{T-1}$
      set $z = [z_1, ..., z_K] = [z_1^T, ..., z_K^T]$ and $\widetilde{w_k} = w_k / \sum_{i=1}^{K} w_i$;
      estimate the gradient $\nabla_\theta \log p_\theta(x)$ with $\delta_\theta = \sum_{k=1}^{K} \widetilde{w_k} \nabla_\theta \log p_\theta(x, z_k)$;
      apply gradient update to $\theta$ using $\delta_\theta$;
      update the inference model parameter $\phi$
      sample $\epsilon \sim \mathcal{N}(0, \mathbf{I})$;
      set $z = \mu(\phi, x) + \sigma(\phi, x) \odot \epsilon$ and calculate $\mathcal{L}(\theta, \phi)$;
      estimate the gradient $\delta_\phi = \nabla_\phi \mathcal{L}(\theta, \phi)$ with the reparameterization trick;
      apply gradient update to $\phi$ using $\delta_\phi$;
  end while
```

Our approach (Algorithm 1) is most closely related to the IWAE approach [7] and Matthew D. Hoffman's HMC approach (MH-HMC) [1] for learning deep generative models with MCMC. Both methods can be viewed as special cases of our proposed approach. The IWAE approach corresponds to setting $T = 1$ and $z^1 = z^0$ in Algorithm (1). The MH-HMC approach [1] is equivalent to setting $K = 1$ and $0 = \beta_0 < \beta_1 = ... = \beta_T = 1$. Previous studies showed that IWAEs can also be interpreted as optimizing the standard ELBO (2) using a more complex variational distribution that is implicitly defined by importance sampling (IS) [13, 14]. Similar interpretation can be used to understand Algorithm 1. Specifically, when using (4) or (8) to estimate the expectation in (1), IS or AIS implicitly defines a proposal distribution, $q_{\text{IS}}(z|x)$ or $q_{\text{AIS}}(z|x)$, using samples from $q_\phi(z|x)$ to approximate the posterior $p_\theta(z|x)$. We can sample from the implicitly defined proposal distributions, $q_{\text{IS}}(z|x)$ or $q_{\text{AIS}}(z|x)$, with Algorithm 2.

```
Algorithm 2: Sampling $q_{\text{IS}}(z|x)$ or $q_{\text{AIS}}(z|x)$
    K: number of importance samples
    L: number of integration steps in HMC
    T: num of intermediate distributions
    case 1: when importance sampling is used
      sample $z_1, ..., z_K \sim q_\phi(z|x)$
      set $w_k = \frac{p_\theta(x, z_k)}{q_\phi(z_k|x)}$ and $\widetilde{w_k} = w_k / \sum_{k=1}^{K} w_k$
    case 2: when AIS is used
      sample $z_1, ..., z_K$ and compute $\widetilde{w_1}, ..., \widetilde{w_K}$
      with Algorithm 1
    sample $j \sim Categorical(\widetilde{w_1}, ..., \widetilde{w_K})$
    return $z_j$
```

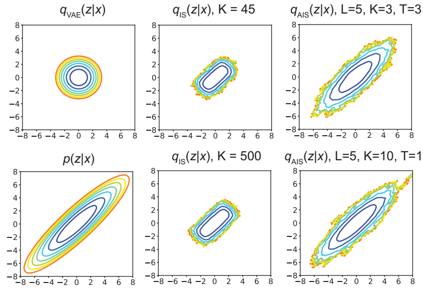

Figure 1: Comparison of implicit distributions $q_{\text{IS}}(z|x)$ and $q_{\text{AIS}}(z|x)$ for approximating the target distribution $p(z|x)$.

One way to compare IWAEs and Algorithm (1) is to compare the computational efficiency of $q_{\text{IS}}(z|x)$ and $q_{\text{AIS}}(z|x)$ for approximating the posterior $p_\theta(z|x)$. To do that, we apply them in the following simple example. The target distribution $p(z|x)$ is chosen to be a normal distribution of two correlated random variables. The proposal distribution $q(z|x)$ is set to the normal distribution of two independent random variables that minimizes the Kullback-Leibler (KL) divergence from $q(z|x)$ to $p(z|x)$. The computational cost of IS using $q_{\text{IS}}(z|x)$ increases linearly with K and the cost of AIS using $q_{\text{AIS}}(z|x)$ scales linearly with $L \times K \times T$. To make a fair comparison, we compare $q_{\text{IS}}(z|x)$ and $q_{\text{AIS}}(z|x)$ under the same computational cost, i.e., K in IS is equal to $L \times K \times T$ in AIS. The inverse temperatures β_t in AIS are set to changing linearly with t from 0 to T. The results are shown in Fig. (1). Both $q_{\text{IS}}(z|x)$ and $q_{\text{AIS}}(z|x)$ become better approximation of $p(z|x)$ when increasing K or $L \times K \times T$. With the same amount of computational cost, $q_{\text{AIS}}(z|x)$ approximates the target distribution better than $q_{\text{IS}}(z|x)$. The better performance of $q_{\text{AIS}}(z|x)$ for approximating $p(z|x)$ is expected to help Algorithm (1) learn better generative models than IWAEs.

3 Experiment Results

Table 1: Results of IWAE-DReG and our approach on the Omniglot and the MNIST dataset.

$\approx \log p(x)$	Omniglot			MNIST		
	K = 1	K=5	K=50	K = 1	K=5	K=50
IWAE-DReG	-109.41	-106.11	-103.91	-86.90	-85.52	-84.38
Ours (T = 5)	-103.22	-102.47	-102.03	-84.56	-84.25	-83.93
Ours (T = 11)	-102.45	-101.94	-101.64	-84.14	-83.78	-83.63

We conducted a series of experiments to evaluate the performance of our proposed algorithm (1) on learning deep generative models using the Omniglot [15] and the MNIST [16] datasets. We used same generative models and same inference models as that used in the IWAE study [7]. Same models are also learned using the two closely related approaches: IWAEs and MH-HMC [1]. Because previous study [17] showed that IWAEs with doubly reparameterized gradient estimators (IWAE-DReG) can improve its performance, we used IWAE-DReG in all computations involving IWAE. Following [18, 1, 19], we evaluate learned generative models using marginal likelihoods estimated with AIS

Table 2: Results of IWAE-DReG, MH-HMC and our approach on the Omniglot dataset with same computational cost.

	$\approx \log p(x)$
IWAE-DReG (K = 55)	-103.85
Ours (L = 5, K = 1, T = 11)	-102.45
IWAE-DReG (K = 275)	-103.13
Ours (L = 5, K = 5, T = 11)	-101.94
IWAE-DReG (K = 2750)	-102.40
Ours (L = 5, K = 50, T = 11)	-101.64

	$\approx \log p(x)$
MH-HMC (L = 5, K = 5, T = 5)	-102.57
Ours (L = 5, K = 5, T = 5)	-102.47
MH-HMC (L = 5, K = 5, T = 11)	-101.32
Ours (L = 5, K = 5, T = 11)	-101.94
MH-HMC (L = 5, K = 50, T = 5)	-102.32
Ours (L = 5, K = 50, T = 5)	-102.03
MH-HMC (L = 5, K = 50, T = 11)	-101.25
Ours (L = 5, K = 50, T = 11)	-101.64

[10] on test datasets. To be confident that the estimated likelihoods are accurate enough for comparing models, we follow [18] to empirically validate the estimates using Bidirectional Monte Carlo [20].

For models trained with IWAE-DReG, the marginal likelihood increases on both datasets when the value of $K \in \{1, 5, 50\}$ increases. This agrees with previous studies [7, 17]. For a fixed K, our approach with $T = 5$ or $T = 11$ produces better density models than IWAE-DReG. If we view IWAE-DReG as a special case of our approach with $T = 1$, then the performance of our approach improves when increasing the value of either K or T. This is because increasing K or T can make implicit distributions defined by IS and AIS better approximate the posterior distribution, which in turn improves estimators in (4) and (8) for estimating the expectation in (1). For a fixed K, the computational cost of our approach is about $L \times T$ times that of IWAE-DReG. Therefore results in Table 1 only show that our approach is an effective way of trading computation for model accuracy. To show that our approach is also computationally more efficient, we also compared learned models trained using IWAE-DReG and our approach with the same computational cost on the Omniglot dataset. The result is shown in Table 2 (left). It shows that, with the same computational cost, our approach leads to better density models than IWAE-DReG.

We also compared our approach with the MH-HMC method for learning deep generative models with same computational cost. The MH-HMC method [1] always sets β_t as $0 = \beta_0 < \beta_1 = ... = \beta_T = 1$. In our approach, β_t are free to choose as long as they satisfy the constraints that $0 = \beta_0 \leq \beta_1 \leq ... \leq \beta_T = 1$. In this study, we set β_t to change linearly between 0 and 1. When $K = 1$, the MH-HMC approach is a special case of our approach for setting β_t. When $K > 1$, the MH-HMC method is different from our approach in the way of weighting samples. The marginal likelihoods of models learned using the MH-HMC method and our approach are shown in Table 2 (right). Similar to models trained with our approach, models trained with the MH-HMC method also improves when increasing the value of K or T. For the cases where $T = 5$, our approach leads to better density models, whereas for the cases where $T = 11$, the MH-HMC method leads to better density models.

Broader Impact

The algorithm developed in this work can be applied to improve the training of various generative models for applications including image recognition/generation and nature language processing. It can effectively trade computational cost for model accuracy without increasing memory cost. With available computational resources rapidly increasing, the algorithm can greatly benefit researchers in various scientific fields that are in need of training generative models.

We note that the algorithm is not designed to be able to remove biases that may exist in the training data. Models trained with the algorithm on biased datasets may inherit those biases such as gender, racial, nationality or age biases in the datasets. Therefore, we encourage users of this algorithm to be careful of any bias that may exist in their datasets. If there is any bias found in the dataset, they should either remove the bias before applying the algorithm or combine the algorithm with other approaches that can eliminate the effect of bias in datasets on trained models.

Acknowledgments and Disclosure of Funding

Funding in direct support of this work: Vannevar Bush Faculty Fellowship from USA Department of Defense

References

[1] Matthew D Hoffman. Learning deep latent gaussian models with markov chain monte carlo. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1510–1519. JMLR. org, 2017.

[2] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

[3] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[4] Radford M Neal. *Probabilistic inference using Markov chain Monte Carlo methods*. Department of Computer Science, University of Toronto Toronto, ON, Canada, 1993.

[5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2013.

[6] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014.

[7] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.

[8] Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[9] Tian Han, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Alternating back-propagation for generator network. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[10] Radford M Neal. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.

[11] Christopher Jarzynski. Nonequilibrium equality for free energy differences. *Physical Review Letters*, 78(14):2690, 1997.

[12] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

[13] Chris Cremer, Quaid Morris, and David Duvenaud. Reinterpreting importance-weighted autoencoders. In *Workshop at International Conference on Learning Representations*, 2017.

[14] Andriy Mnih and Danilo J Rezende. Variational inference for monte carlo objectives. In *Proceedings of the 33rd International Conference on Machine Learning-Volume 48*, pages 2188–2196, 2016.

[15] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[16] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[17] George Tucker, Dieterich Lawson, Shixiang Gu, and Chris J. Maddison. Doubly reparameterized gradient estimators for monte carlo objectives. In *International Conference on Learning Representations*, 2019.

[18] Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. On the quantitative analysis of decoder-based generative models. In *International Conference on Learning Representations*, 2017.

[19] Chris Cremer, Xuechen Li, and David Duvenaud. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pages 1078–1086, 2018.

[20] Roger B Grosse, Zoubin Ghahramani, and Ryan P Adams. Sandwiching the marginal likelihood using bidirectional monte carlo. *arXiv preprint arXiv:1511.02543*, 2015.