Variational Inference with Normalizing Flows

Danilo Jimenez Rezende, Shakir Mohamed

Google DeepMind, London

February 16, 2024

Presented by Alexander Thomson from Duke B&B

Introduction

This paper presents the use of **normalizing flows** to develop rich posterior approximations for variational inference [3].

Some Background:

- Variational inference aims to approximate a complex posterior distribution, $p_\theta(\boldsymbol{z}|\boldsymbol{x})$, with a separate, simpler and tractable class of distributions, $q_\phi(\boldsymbol{z}|\boldsymbol{x})$.
- This approximate posterior distribution might not be able to fully capture the structure of the true posterior distribution.
- Richer posterior approximations often have the trade-off of being less computationally feasible.

Goals:

- This normalizing flow work therefore aims to provide a way to incorporate potentially very complex approximate posterior distributions into the framework of variational inference in a way that is not so limited by computational cost.

Related Work

- This work builds on research in amortized variational inference.
- Deep latent Gaussian models (DLGM) are studied in this work [4].
- When viewed using an encoder/decoder structure, these are the variational autoencoder [2].

Deep Latent Gaussian Models

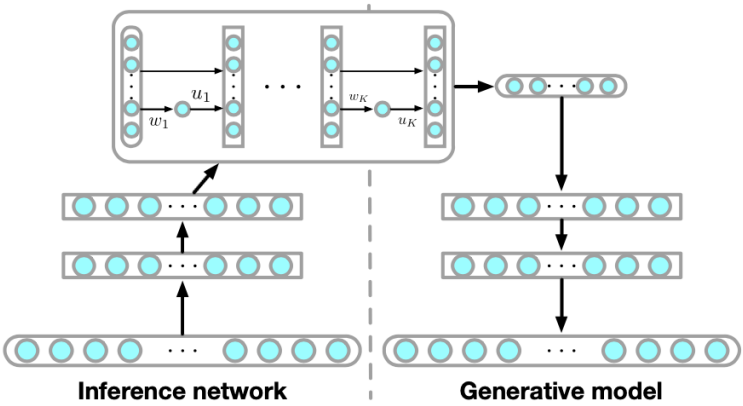


Figure: Model structure discussed in this work.

Amortized Variational Inference

- We consider a probabilistic model with observed evidence $\boldsymbol{x}$, latent variables $\boldsymbol{z}$, and parameters $\boldsymbol{\theta}$.
- To train this model, we wish to find $\boldsymbol{\theta}$ that maximizes the probability of the observed evidence $p_\theta(\boldsymbol{x})$.
- This then would involve integrating out the latent variables from our probabilistic graphical model $\boldsymbol{z}$, which tends to be intractable:

$$\log p_\theta(\boldsymbol{x}) = \log \int p_\theta(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}.$$

Amortized Variational Inference

Instead, an approximate posterior distribution, $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, is used:

$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z} \tag{1}$$

$$= \log \int \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z} \tag{2}$$

$$\geq -\mathbb{D}_{\mathrm{KL}}[q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})] + \mathbb{E}_q[\log p_\theta(\mathbf{x}|\mathbf{z})] = -\mathcal{F}(\mathbf{x}), \tag{3}$$

Figure: Derivation of the evidence lower bound (ELBO) $-\mathcal{F}$ (or negative free energy), where step (3) follows from Jensen's inequality. This can viewed as a reconstruction term, recreating the input from a sample of the latent variables, and a regularization term, the *encoder* section of the network should be similar to a prior $p(\boldsymbol{z})$ on the latent variables.

Note that the ELBO can also be viewed as $-\mathcal{F} = \log p_\theta(\boldsymbol{x}) - \mathbb{D}_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x}))$ i.e. it should maximize the probability of the observed data and minimize the KL-divergence between the true and approximate posteriors [2].

Stochastic Backpropogation

- A simple posterior distribution is chosen, for example: $q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_\phi(\boldsymbol{x}), \text{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{x}))$ where the mean and variance are parameterized by a neural network.
- Samples can then be easily drawn from this known distribution and Monte Carlo approximation can be used for the parameter updates (rather than a closed form expectation)
- An example of that structure using a Gaussian posterior:

$$z \sim \mathbb{N}(z|\mu, \sigma^2) \leftrightarrow z = \mu + \sigma\epsilon,$$

$$\epsilon \sim \mathbb{N}(0, 1)$$

$$\nabla_\phi \mathbb{E}_{q_\phi(z)}[f_\theta(z)] \leftrightarrow \mathbb{E}_{\mathbb{N}(\epsilon|0,1)}[\nabla_\phi f_\theta(\mu + \sigma\epsilon)],$$

$$\text{where } \phi = \{\mu, \sigma^2\}$$

- Note that when a random mini-batch is incorporated into this training, it may be referred to as doubly stochastic estimation.

Finite Normalizing Flows

- The true posterior distribution is often more complex than the chosen approximate posterior distribution can represent.
- Ideally, we would want there to exist some ϕ such that $q_\phi(\boldsymbol{z}|\boldsymbol{x}) = p_\theta(\boldsymbol{z}|\boldsymbol{x})$.
- Often variational methods cannot do this even asymptotically due to the restrictive choice of their approximate posterior family.
- Normalizing flows can be used to map a simple probability distribution to a much more complex distribution.

Finite Normalizing Flows

- Normalizing flows transform this simple distribution through a series of smooth, invertible mappings, $f : \mathbb{R}^d \to \mathbb{R}^d$, where $f^{-1} = g$, $g \circ f(\boldsymbol{z}) = \boldsymbol{z}$.
- If $\boldsymbol{z}$ has distribution $q(\boldsymbol{z})$, then $\boldsymbol{z}' = f(\boldsymbol{z})$ has distribution:

$$q(\boldsymbol{z}') = q(\boldsymbol{z}) |\det \frac{\partial f^{-1}}{\partial \boldsymbol{z}'}| = q(\boldsymbol{z}) |\det \frac{\partial f}{\partial \boldsymbol{z}}|^{-1}.$$

- A series of K transformations $\boldsymbol{z}_K = f_K \circ ... \circ f_2 \circ f_1(\boldsymbol{z}_0)$ then has log density:

$$\ln q_K(\boldsymbol{z}_K) = \ln q_0(\boldsymbol{z}_0) - \sum_{k=1}^{K} \ln |\det \frac{\partial f_k}{\partial \boldsymbol{z}_{k-1}}|$$

- By the law of the unconscious statistician (LOTUS):

$$\mathbb{E}_{q_K}[h(\boldsymbol{z})] = \mathbb{E}_{q_0}[h(f_K \circ f_{K-1} \circ ... \circ f_1(\boldsymbol{z}_0))],$$

where $h(\boldsymbol{z})$ is a function that does not depend on q_K. This relates the expectation over the transformed density to an expectation over the simple original density.

Computational Cost

- These transformations could be defined and trained using standard neural networks.
- However, the calculation of the Jacobian determinant and relevant gradient information in existing invertible neural network structures had computational complexity of $O(LD^3)$ where D is the dimension of the hidden layer and L is the number of hidden layers.
- The following methods are designed so that this calculation is much faster, $O(D)$.

Normalizing Flows

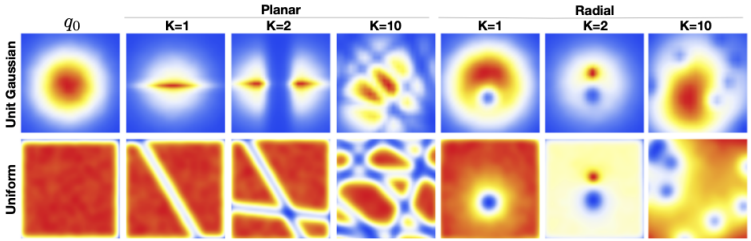


Figure: Normalizing flows applied to a uniform and Gaussian distribution.

Planar Flows

Consider the transformation of the form $f(\boldsymbol{z}) = \boldsymbol{z} + \boldsymbol{u}h(\boldsymbol{w}^\top \boldsymbol{z} + b)$, where $\lambda = \{\boldsymbol{w} \in \mathbb{R}^D, \boldsymbol{u} \in \mathbb{R}^D, b \in \mathbb{R}\}$ are parameters that can be adjusted during training and $h(\cdot)$ is smooth element-wise non-linearity.

- The log determinant of the Jacobian can then be calculated in $O(D)$ time.
- It is $|\det \frac{\partial f}{\partial \boldsymbol{z}}| = |\det(\boldsymbol{I} + \boldsymbol{u}\psi(\boldsymbol{z})^\top)| = |1 + \boldsymbol{u}^\top \psi(\boldsymbol{z})|$ where $\psi(\boldsymbol{z}) = h'(\boldsymbol{w}^\top \boldsymbol{z} + b)\boldsymbol{w}$.
- The name planar flow refers to how the expansions and contractions of the density defined by this transformation are perpendicular to a hyperplane $\boldsymbol{w}^\top \boldsymbol{z} + b = 0$.

Radial Flow

Consider the transformation of the form $f(\boldsymbol{z}) = \boldsymbol{z} + \beta h(\alpha, r)(\boldsymbol{z} - \boldsymbol{z}_0)$ where $r = |\boldsymbol{z} - \boldsymbol{z}_0|$, $h(\alpha, r) = 1/(\alpha + r)$.

- The radial flow in named as such since it modifies the density around a reference point $\boldsymbol{z}_0$.
- Its Jacobian is given by: $|\det \frac{\partial f}{\partial \boldsymbol{z}}| = [1 + \beta h(\alpha, r)]^{d-1}[1 + \beta h(\alpha, r) + \beta h'(\alpha, r)r]$

The Objective

Since $\ln q_K(\boldsymbol{z}_K) = \ln q_0(\boldsymbol{z}_0) - \sum_{k=1}^{K} \ln \left| \det \frac{\partial f_k}{\partial \boldsymbol{z}_{k-1}} \right|$, if we choose to express the posterior distribution using normalizing flows, $q_\phi(\boldsymbol{z}|\boldsymbol{x}) = q_K(\boldsymbol{z}_K)$, we can write the evidence lower bound (ELBO) as:

$$
\begin{aligned}
-\mathcal{F}(\boldsymbol{x}) &= -\mathbb{D}_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})] + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] \\
&= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[-\log q_\phi(\boldsymbol{z}|\boldsymbol{x}) + \log p(\boldsymbol{z}) + \log p_\theta(\boldsymbol{x}|\boldsymbol{z})] \\
&= \mathbb{E}_{q_0(\boldsymbol{z}_0)}[-\log q_K(\boldsymbol{z}_K|\boldsymbol{x}) + \log p(\boldsymbol{z}_K) + \log p_\theta(\boldsymbol{x}|\boldsymbol{z}_K)] \\
&= -\mathbb{E}_{q_0(\boldsymbol{z}_0)}[\log q_K(\boldsymbol{z}_K|\boldsymbol{x})] + \mathbb{E}_{q_0(\boldsymbol{z}_0)}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z}_K)p(\boldsymbol{z}_K)] \\
&= -\mathbb{E}_{q_0(\boldsymbol{z}_0)}[\log q_0(\boldsymbol{z}_0|\boldsymbol{x})] + \mathbb{E}_{q_0(\boldsymbol{z}_0)}[\sum_{k=1}^{K} \log |1 + \boldsymbol{u}_k^\top \psi_k(\boldsymbol{z}_{k-1})|] + \mathbb{E}_{q_0(\boldsymbol{z}_0)}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z}_K)p(\boldsymbol{z}_K)]
\end{aligned}
$$

The Algorithm

Algorithm 1 Variational Inf. with Normalizing Flows

Parameters: ϕ variational, θ generative
while not converged **do**
  $\mathbf{x} \leftarrow \{\text{Get mini-batch}\}$
  $\mathbf{z}_0 \sim q_0(\bullet|\mathbf{x})$
  $\mathbf{z}_K \leftarrow f_K \circ f_{K-1} \circ \ldots \circ f_1(\mathbf{z}_0)$
  $\mathcal{F}(\mathbf{x}) \approx \mathcal{F}(\mathbf{x}, \mathbf{z}_K)$
  $\Delta\boldsymbol{\theta} \propto -\nabla_\theta \mathcal{F}(\mathbf{x})$
  $\Delta\boldsymbol{\phi} \propto -\nabla_\phi \mathcal{F}(\mathbf{x})$
end while

Figure: The training loop of VI with normalizing flows. The algorithmic complexity is $O(LN^2) + O(KD)$ where L is the number of layers used to map data to the initial parameters of the latent space, N is the hidden layer size, K is flow-length, D is the latent variable dimension.

Simulated Results

Table 1. Test energy functions.

Potential $U(\mathbf{z})$
1: $\frac{1}{2}\left(\frac{\|\mathbf{z}\|-2}{0.4}\right)^2 - \ln\left(e^{-\frac{1}{2}\left[\frac{\mathbf{z}_1-2}{0.6}\right]^2} + e^{-\frac{1}{2}\left[\frac{\mathbf{z}_1+2}{0.6}\right]^2}\right)$
2: $\frac{1}{2}\left[\frac{\mathbf{z}_2-w_1(\mathbf{z})}{0.4}\right]^2$
3: $-\ln\left(e^{-\frac{1}{2}\left[\frac{\mathbf{z}_2-w_1(\mathbf{z})}{0.35}\right]^2} + e^{-\frac{1}{2}\left[\frac{\mathbf{z}_2-w_1(\mathbf{z})+w_2(\mathbf{z})}{0.35}\right]^2}\right)$
4: $-\ln\left(e^{-\frac{1}{2}\left[\frac{\mathbf{z}_2-w_1(\mathbf{z})}{0.4}\right]^2} + e^{-\frac{1}{2}\left[\frac{\mathbf{z}_2-w_1(\mathbf{z})+w_3(\mathbf{z})}{0.35}\right]^2}\right)$
with $w_1(\mathbf{z}) = \sin\left(\frac{2\pi\mathbf{z}_1}{4}\right)$, $w_2(\mathbf{z}) = 3e^{-\frac{1}{2}\left[\frac{(\mathbf{z}_1-1)}{0.6}\right]^2}$, $w_3(\mathbf{z}) = 3\sigma\left(\frac{\mathbf{z}_1-1}{0.3}\right)$ and $\sigma(x) = 1/(1+e^{-x})$.

Figure: Several unnormalized potential functions that define the distributions used to test the normalizing flow structure's representative power compared to a known truth.

Simulated Results

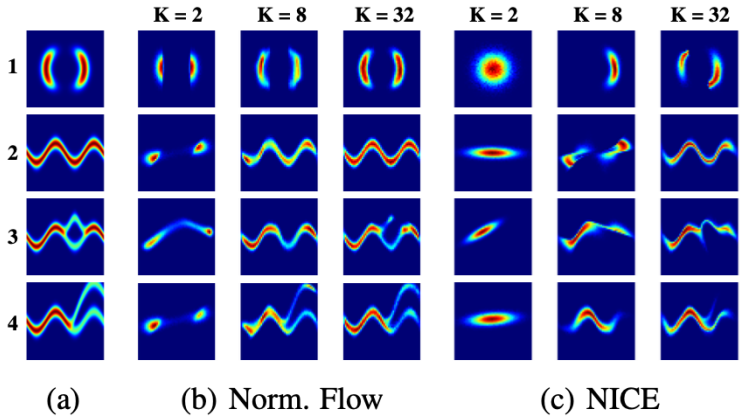


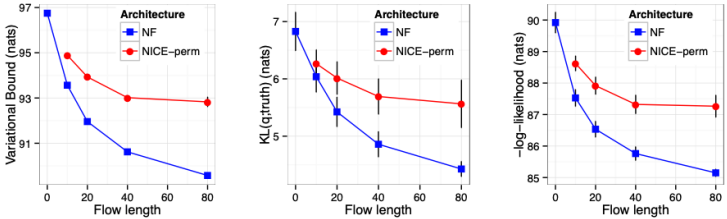
Figure: The results of those experiments. (a) shows the true distribution of $\boldsymbol{z}$ while (b) shows the proposed Norm Flow method and (c) the Non-linear Independent Component Estimation (NICE) method [1]. K is the number of transformations used. NICE can be viewed a 'volume-preserving' flow where the Jacobian equals 1.

MNIST and CIFAR

Model	$-\ln p(\mathbf{x})$
DLGM diagonal covariance	≤ 89.9
DLGM+NF (k = 10)	≤ 87.5
DLGM+NF (k = 20)	≤ 86.5
DLGM+NF (k = 40)	≤ 85.7
DLGM+NF (k = 80)	≤ 85.1
DLGM+NICE (k = 10)	≤ 88.6
DLGM+NICE (k = 20)	≤ 87.9
DLGM+NICE (k = 40)	≤ 87.3
DLGM+NICE (k = 80)	≤ 87.2
Results below from (Salimans et al., 2015)	
DLGM + HVI (1 leapfrog step)	88.08
DLGM + HVI (4 leapfrog steps)	86.40
DLGM + HVI (8 leapfrog steps)	85.51
Results below from (Gregor et al., 2014)	
DARN $n_h = 500$	84.71
DARN $n_h = 500$, adaNoise	84.13

Figure: A comparison with other variational approaches on binary MNIST data. (DARN) Deep AutoRegressive Networks. (HVI) Hamiltonian Variational Inference. HVI can be viewed as an infinitesimal volume preserving flow.

MNIST and CIFAR



(a) Bound $\mathcal{F}(\mathbf{x})$ (b) $\mathbb{D}_{\mathrm{KL}}(q; p(z|x))$ (c) $-\ln p(\mathbf{x})$

Figure: Performance of the normalizing flow method and NICE on MNIST data.

MNIST and CIFAR

	$K = 0$	$K = 2$	$K = 5$	$K = 10$
$-\ln p(\mathbf{x})$	-293.7	-308.6	-317.9	-320.7

Figure: Performance of DLGM + K flows on continuous CIFAR-10 image data. Note that the test log-likelihood improves as the nunmber of flow layers increases.

Conclusion and Recommendations

- I would recommend this paper to anyone generally interested in variational inference or with a specific interest in a type of deep generative model such as generative adversarial networks (GANs) or variational autoencoders (VAEs).
- Normalizing flows incorporate the transformations of densities directly into their model structure, which differs from a diffusion approach, for example, which learns a transition with added noise with a neural network.
- I also found their albeit brief discussion of the limited representational power of common approximate posterior distributions quite interesting (a motivation for the paper). Especially, the noted limitations of the assumptions made by sigmoid belief networks.

References

[1] L. Dinh, D. Krueger, and Y. Bengio. Nice: Non-linear independent components estimation, 2015.

[2] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2022.

[3] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015. PMLR.

[4] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models, 2014.