Demystifying Normalizing Flows for Variational Inference

Ashay Athalye [*1] **Cindy Zhang** [*2]

Abstract

Normalizing flows transform simple probability distributions into complex distributions that are efficient to sample from, efficient to evaluate, and are highly expressive. They are a useful tool for variational inference because they allow for arbitrarily complex approximate posterior distributions. In this work, we provide a mathematical and visual guide for understanding how normalizing flows work and how they are used for variational inference. We demonstrate the benefits of variational inference with normalizing flows relative to mean-field variational inference on a Gaussian mixture model.

1. Introduction

Variational inference lies at the core of many commonly used machine learning methods, from large-scale topic models to variational autoencoders. In order to enable efficient, tractable inference, simplifying assumptions about the posterior approximation have to be made. One commonly made assumption is that the posterior distribution is part of the mean-field family, which means that all of its latent variables are independent of one another.

Restricting the family of distributions used to approximate an intractable posterior distribution comes at a cost—the more restrictive the family is, the less likely any of the approximations will resemble the true posterior. As a result, we would like to have minimal restrictions on the complexity of our posterior approximation while still ensuring that variational inference is computationally tractable.

Normalizing flows is one tool that can be used to produce rich posterior approximations for variational inference (Rezende & Mohamed, 2016). It consists of a series of invertible and differentiable mappings that transform simple probability distributions into more complex ones. We can learn and adjust the parameters of these transforms so that the resulting distribution approximates our true posterior. These resulting distributions are efficient to sample from and evaluate—the key to this is the change-of-variables formula.

This paper is a tutorial on how normalizing flows work and how they are used for variational inference. In Section 2, we review recent work on normalizing flows and its applications. In Section 3, we describe the change-of-variables formula as it relates to normalizing flows and provide figures that illustrate that transformation of probability distributions. Section 4 goes through the implementation of two types of flows: planar flows and radial flows. In Section 5, we go through the procedure of performing variational inference with normalizing flows and illustrate the process of fitting normalizing flows to toy distributions. Empirical results for Gaussian mixture models are presented in Section 6, and our conclusions are presented in Section 7. Especially because this is meant to be an educational guide, we encourage the enthusiastic reader to look at all our code at `https://colab.research.google.com/drive/1KrovUf2mh-x8DWNqj3LWc-i48o5jpcFt?usp=sharing`, which provides some animations and interactivity.

2. Background and Related Work

Normalizing flows were first introduced to machine learning back in the context of density estimation. They were used to estimate densities and marginals on never-before-seen data, which allowed for detecting corruptions of images (Rippel & Adams, 2013) and image generation. The introduction of coupling flows (Dinh et al., 2015) led to competitive results for image generation for models trained on MNIST, TFD, SVHN, and CIFAR-10.

Further advances were made in the area of image generation by the Glow architecture (Kingma & Dhariwal, 2018), which produced compelling full-color images using normalizing flows composed of invertible convolutions. This work has been extended to produce high-dimensional images as well (Behrmann et al., 2019; Grathwohl et al., 2018).

In the context of inference, normalizing flows can be used for both sampling and variational inference. In importance

[*]Equal contribution [1]Department of EECS, MIT [2]Department of Mathematics, MIT. Correspondence to: Ashay Athalye <ashay@mit.edu>, Cindy Zhang <cindyyz@mit.edu>.

Template borrowed from *Proceedings of the 39^{th} International Conference on Machine Learning*, 2022.

sampling, the user-specified density function being sampled from can be implemented and optimized with normalizing flows (Müller et al., 2019). Flows can be applied to MCMC algorithms to model the Hamiltonian dynamics in HMC (Neal et al., 2011) or to reparameterize the target distribution (Papamakarios et al., 2021).

As we will discuss further, normalizing flows are very useful as posterior approximations in variational inference (Rezende & Mohamed, 2016; Kingma et al., 2016; Berg et al., 2018). They can be thought of as a reparameterization trick or a generalized application of the change-of-variables formula. A fixed distribution that is easy to sample from and evaluate is transformed into a complex distribution that is easily reparameterizable by design. In the following section, we will go into more detail on this transformation procedure.

3. Transforming Probability Distributions

To transform a probability distribution, we can perform a change of variable transformation, defined below.

Let $\mathbf{z} \in \mathcal{R}^d$ be a random variable with distribution $q(\mathbf{z})$ and $f : \mathcal{R}^d \rightarrow \mathcal{R}^d$ an invertible smooth mapping. We can use f to transform $\mathbf{z} \sim q(\mathbf{z})$. The change of variables formula tells us that the probability density $p_{z'}(\mathbf{z'})$ is the product of the probability density $p_z(f^{-1}(\mathbf{z'}))$ and a volume correction term $|\det J(f^{-1}(\mathbf{z'}))|$ (Kobyzev et al., 2021). The resulting random variable $\mathbf{z}' = f(\mathbf{z})$ has the following probability distribution

$$q(\mathbf{z}') = q(\mathbf{z}) \left| \det \frac{\delta f^{-1}}{\delta \mathbf{z}'} \right| = q(\mathbf{z}) \left| \det \frac{\delta f}{\delta \mathbf{z}} \right|^{-1} \quad (1)$$

where the last equality is obtained through the inverse function theorem.

How does volume correction work? We can think of a determinant as the local, linearized rate of volume change of a transformation. If we take **dz** to be the small neighborhood around **z** and **dz'** to be the small neighborhood around **z'** that **dz** maps to, then

$$|\det J(f^{-1}(\mathbf{z'}))| \approx \frac{\text{Volume}(\mathbf{dz})}{\text{Volume}(\mathbf{dz'})}$$

We see in equation (1) that if the volume of **dz'** is greater than the volume of **dz**, then the probability mass $p_{z'}(\mathbf{z'})$ is less than the probability mass $p_z(\mathbf{z})$. Similarly, if the volume of **dz'** is less than the volume of **dz**, the $p_{z'}(\mathbf{z'}) > p_z(\mathbf{z})$. Intuitively, this means $|\det J(f^{-1}(\mathbf{z'}))|$ balances the probability density function $p_{z'}(z')$ so that it integrates to 1. Additional intuition and visuals about how this works can be found in (Jang, 1970), (Jean, 2018), and (Cheng, 2013).

3.1. Chaining transformations

If we start with a random vector $\mathbf{z}_0$ with distribution $\mathbf{q}_0$, we can apply a series of invertible, differentiable mappings $f_i, i \in 1, \cdots, k$ with $k \in \mathbb{R}^+$ and obtain a normalizing flow:

$$\mathbf{z}_k = f_k \circ f_{k-1} \circ ... \circ f_1(\mathbf{z}_0)$$

The distribution of $\mathbf{z}_k \sim q_k(\mathbf{z}_k)$ will be given by

$$\begin{aligned} q_k(\mathbf{z}_k) &= q_0(f_1^{-1} \circ f_2^{-1} \circ ... \circ f_k^{-1}(\mathbf{z}_k)) \prod_{i=1}^{k} \left| \det \frac{\delta f_i^{-1}}{\delta \mathbf{z}_i} \right| \\ &= q_0(\mathbf{z_0}) \prod_{i=1}^{k} \left| \det \frac{\delta f_i}{\delta \mathbf{z}_{i-1}} \right|^{-1} \end{aligned}$$

Note that the Chain Rule tells us that the determinant of the Jacobian of f is the product of the determinants of the individual f_k.

This series of transformations can transform a simple probability distribution (e.g. Gaussian) into a complicated multi-modal one. We will often write this in terms of log-probabilities to simplify the computation and obtain

$$\log q_K(\mathbf{z}_k) = \log q_0(\mathbf{z}_0) - \sum_{i=1}^{k} \log \left| \det \frac{\delta f_i}{\delta \mathbf{z}_{i-1}} \right|$$

To be of practical use, normalizing flows should satisfy several conditions (Kobyzev et al., 2021):

- Be invertible because we will need f^{-1} to compute the likelihood, as we will see later
- Be sufficiently expressive to model the distribution of interest
- Be computationally efficient: f and the determinant must be efficient to calculate. The Jacobian determinant generally requires $O(LD^3)$ operations, where D is the dimension of the Jacobian and L is the number of chained mappings, but we will see that in the planar and radial flows this determinant can be computed in linear time, $O(LD)$.

There are two ways we can use normalizing flows: we can generate samples using $x = f(z)$ with $z \sim p_z(z)$, and we can evaluate the model's density at a given point using $p_x(x) = p_z(x)|\det J_f(x)|^{-1}$.

To draw samples, we need to sample the base distribution $p_z(z)$ and compute the forward transformation f. To evaluate the model's density, we need to perform the inverse

transformation f^{-1}, calculate its Jacobian determinant, and evaluate the density $p_z(x)$.

4. Types of flows

4.1. Planar flows

Planar flows are a family of transformations of the form

$$f(\mathbf{z}) = \mathbf{z} + \mathbf{u}h(w^T\mathbf{z} + b)$$

where $\mathbf{u} \in \mathbb{R}^D$ and $\mathbf{w} \in \mathbb{R}^D$ are vectors, $\mathbf{b} \in \mathbb{R}$ is a scalar bias, and h is a smooth non-linear activation function that we apply element-wise (equation 13 in (Rezende & Mohamed, 2016)). The second term can be interpreted as a MLP with a bottleneck hidden layer with a single unit. Since information goes through the single bottleneck, a long chain of transformations is required to capture high-dimensional dependencies (Rezende & Mohamed, 2016).

To compute $|\det(\frac{\partial f}{\partial z})|$, note that: $\frac{da^T x}{dx} = a$ (matrix cookbook, page 10), and that for an invertible matrix $\mathbf{A}$ and vectors $\mathbf{x}$ and $\mathbf{y}$, $\det(\mathbf{A} + \mathbf{x}\mathbf{y}^T) = (1 + \mathbf{y}^T\mathbf{A}^{-1}\mathbf{x})\det(\mathbf{A})$ (matrix determinant lemma).

Then we have:

$$\begin{aligned}
|\det(\frac{\partial f}{\partial z})| &= |\det(\frac{\partial z}{\partial z} + \frac{\partial h(w^T\mathbf{z}+b)}{\partial z}\mathbf{u}^T)| \\
&= |\det(\mathbf{I} + h'(\mathbf{w}^T\mathbf{z}+b)\frac{\partial(\mathbf{w}^T\mathbf{z}+b)}{\partial z}\mathbf{u}^T| \\
&= |\det(\mathbf{I} + h'(\mathbf{w}^T\mathbf{z}+b)\mathbf{w}\mathbf{u}^T| \\
&= |(1 + \mathbf{u}^T\psi(\mathbf{z}))\det(\mathbf{I}) \\
&= |1 + \mathbf{u}^T\psi(\mathbf{z})|
\end{aligned}$$

where $\psi(\mathbf{z}) = h'(\mathbf{w}^T\mathbf{z} + b)\mathbf{w}$ Inserting this expression into our equation for the final density at the end of the chain of mappings, it results that

$$log(q_k(\mathbf{z}_k)) = log(q_0(\mathbf{z}_0)) - \sum_{k=1}^{K} log(|1 + \mathbf{u}^T\psi(\mathbf{z})|)$$

It is important to note that not all parameters make f invertible. Invertibility conditions for planar flows are derived in (Appendix A.1. of (Rezende & Mohamed, 2016)). A sufficient condition for the invertibility of f is $\mathbf{w}^T\mathbf{u} \geq -1$, which is enforced by considering, instead of $\mathbf{u}$, taking $\hat{\mathbf{u}} = \mathbf{u} + (m(\mathbf{w}^T\mathbf{u}) - \mathbf{w}^T\mathbf{u})\frac{\mathbf{w}}{||\mathbf{w}||^2}$ where $m(x) = -1 + log(1 + e^x)$.

Note that this family allows for linear-time computation of the determinant (Papamakarios et al., 2019).

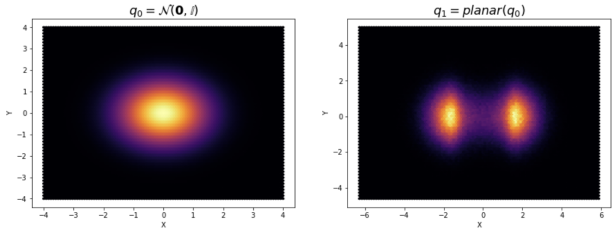

Figure 1. The effect of a single planar flow with parameters $w = (4, 0), u = (2, 0), b = 0$ on a multivariate normal distribution

4.2. Radial flows

Radial flows are a family of transformations that modify an initial density around a reference point z_r, that take the form

$$f(\mathbf{z}) = \mathbf{z} + \beta h(\alpha, r)(\mathbf{z} - \mathbf{z}_0)$$

where $r = \|\mathbf{z} - \mathbf{z}_r\|_2$, $\alpha \in \mathbb{R}^+$, and $\beta \in \mathbb{R}$, and $h(\alpha, r) = \frac{1}{\alpha + r}$ (equation 14 in (Rezende & Mohamed, 2016)).

$$\begin{aligned}
|\det(\frac{\partial f}{\partial z})| &= |\det(\frac{\partial \mathbf{z}}{\partial \mathbf{z}} + \beta\frac{\partial h(\alpha, r)}{\partial \mathbf{z}}(\mathbf{z} - \mathbf{z}_r)^T + \beta h(\alpha, r)\frac{\partial \mathbf{z}}{\partial \mathbf{z}})| \\
&= |\det((1 + \beta h(\alpha, r))\mathbf{I} + \beta\frac{\partial h(\alpha, r)}{\partial \mathbf{z}}(\mathbf{z} - \mathbf{z}_r)^T)|
\end{aligned}$$

because

$$\frac{\partial r}{\partial \mathbf{z}} = \frac{\partial ||\mathbf{z} - \mathbf{z}_r||_2}{\partial \mathbf{z}} = \frac{\mathbf{z} - \mathbf{z}_r}{||\mathbf{z} - \mathbf{z}_r||_2}$$

We apply chain rule to obtain

$$\frac{\partial h(\alpha, r)}{\partial \mathbf{z}} = h'(\alpha, r)\frac{\mathbf{z} - \mathbf{z}_r}{||\mathbf{z} - \mathbf{z}_r||_2}$$

We note that the matrix $\frac{\mathbf{z}-\mathbf{z}_r}{||\mathbf{z}-\mathbf{z}_r||_2}(\mathbf{z} - \mathbf{z}_r)^T$ has rank 1, and can therefore be diagonalized into

$$\frac{\mathbf{z} - \mathbf{z}_r}{||\mathbf{z} - \mathbf{z}_r||_2}(\mathbf{z} - \mathbf{z}_r)^T = \mathbf{P}\mathbf{A}\mathbf{P}^{-1}$$

where $\mathbf{A}$ is a matrix with all zeros except for the topleft element, which is r.

Finally, noting that $\det(\mathbf{P}\mathbf{A}\mathbf{P}^{-1}) = \det(\mathbf{A})$, we get that

$$|\det(\frac{\partial f}{\partial z})| = (1 + \beta h(\alpha, r))^{D-1}(1 + \beta h(\alpha, r) + \beta h'(\alpha, r)r)$$

Inserting this expression into our equation for the final density at the end of the chain of mappings, it results that

$$log(q_k(\mathbf{z}_k)) = log(q_0(\mathbf{z}_0)) - \sum_{k=1}^{K}[(D-1)log(1 + \beta h(\alpha, r)) + log(\beta h(\alpha, r) + \beta h'(\alpha, r)r)]$$

Again, the choice of parameter values will determine if f is invertible. A sufficient condition for the invertibility of f is $\beta \geq \alpha$, which is enforced by taking $\hat{\beta} = \alpha + m(\beta)$ where $m(x) = log(1 + e^x)$ (Rezende & Mohamed, 2016).

Note that this family allows for linear-time computation of the determinant (Papamakarios et al., 2019).

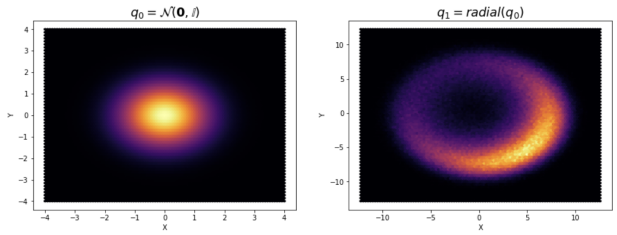

Figure 2. The effect of a single radial flow with parameters $\alpha = 0.5, \beta = 9, z_r = (-0.5, 0.5)$ on a multivariate normal distribution

4.3. Effect on transformation

The planar and radial flows get their names from their effect on distributions. Below we evaluate the inverse of the Jacobian for different sets of parameters. Depending on the parameters, the planar flow induces an expansion or contraction of the initial density along the hyperplane defined by $\mathbf{w}^T\mathbf{z} + b = 0$. Similarly, the radial flow induces an expansion or contraction of the initial density around the reference point $\mathbf{z}_r$.

Recall that when applying a single flow, the resulting distribution is expressed as

$$q(\mathbf{z}') = q(\mathbf{z}) \left| \det \frac{\delta f}{\delta \mathbf{z}} \right|^{-1}$$

We can plot the inverse of the absolute value of the determinant of the Jacobian to visualize the change in volume that is occurring to get an understanding and intuition about how the distribution is being transformed, which we see in figures 3 and 4.

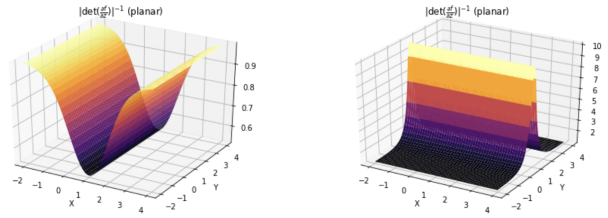

Figure 3. Evaluation of $\left| \det \frac{\delta f}{\delta \mathbf{z}} \right|^{-1}$ for an expansion away from the line $x = 1$, with parameters $w = (-1, 0), u = (-0.9, -0.9), b = 1$ (left), and for a contraction towards the line $y = 1$, with parameters $w = (0, -1), u = (0.9, 0.9), b = 1$ (right).

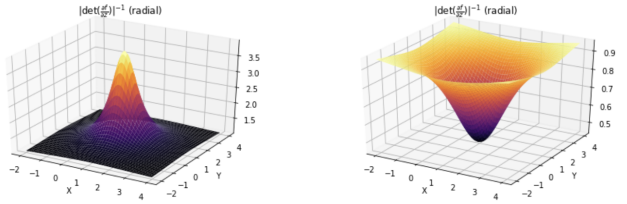

Figure 4. Evaluation of $\left| \det \frac{\delta f}{\delta \mathbf{z}} \right|^{-1}$ for a radial contraction towards the point $(1, 1)$, with parameters $\alpha = 2, \beta = -0.99, z_r = (1.0001, 1.00001)$ (left), and for a radial expansion away from the point $(1, 1)$, with parameters $\alpha = 2, \beta = 0.99, z_r = (1.00001, 1.00001)$ (right).

5. Fitting normalizing flows

So far, we have composed a series of invertible mappings to form a transformation f, which is then applied to some known base distribution $p_x(x)$, typically a multivariate normal distribution. This forms a new distribution $p_y(y)$ that can be evaluated and sampled from using the change of variables formula. Our flow-based model consists of the transformation of $p_x(x)$ into our more complex distribution $p_y(y)$ through the chain of functions that form f. Let our model be parameterized by $\theta = (\phi, \psi)$, where ϕ are the parameters of f and ψ are the parameters of $p_x(x)$.

To perform variational inference, we want our flow-based model $p_y(y; \theta)$ to be able to approximate a target distribution $p^*_x(x)$.We can do so by minimizing the KL divergence between $p_y(y; \theta)$ and $p^*_x(x)$. Depending on what we know about our target distribution, we can choose to either minimize the forward KL divergence or the reverse KL divergence.

5.1. Forward KL Divergence

We minimize the forward KL divergence when we have existing samples from the target distribution $p^*_x(x)$. The forward KL divergence between $p_y(y; \theta)$ and $p^*_x(x)$ can be written as

$$\begin{aligned}\mathcal{L}(\theta) &= D_{KL}(p^*_x(x) p_y(y; \theta)) \\ &= \mathbb{E}_{\log p^*_x(x)}[\log p^*_x(x) - \log p_y(y; \theta)] \\ &= -\mathbb{E}_{p_y(\log p^*_x(x)}[\log p_y(y; \theta)] + C\end{aligned}$$

where C is a constant not dependent on θ. From our change of variables formula, we know that

$$p_y(y; \theta) = p_x(f^{-1}(y, \phi); \psi) |\det J(f^{-1}(y; \phi))|. \quad (5)$$

Substituting this equation into our loss function, we get

$$\mathcal{L}(\theta) = -\mathbb{E}_{\log p^*_x(x)}[\log p_x(f^{-1}(y, \phi); \psi) + \log |\det J(f^{-1}(y; \phi))|] + C.$$

Using samples $\{\mathbf{x}_i\}_{i=1}^N$ from the target distribution $p^*_x(x)$, we can compute a Monte Carlo estimate of this expectation,

so the loss function we want to minimize becomes

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{n=1}^{N} \log p_x(f^{-1}(x_i, \phi); \psi) + \log|\det J(f^{-1}(x_i; \phi))| \tag{6}$$

In practice, θ is then typically optimized using stochastic gradient-based methods.

5.2. Reverse KL Divergence

Suppose we have a target distribution $p_x^*(x)$, and we are able to evaluate this distribution up to a normalizing constant, i.e. we can evaluate $\tilde{p}_x(x)$ where

$$p_x^*(x) = \frac{\tilde{p}_x(x)}{Z}.$$

In this case we want to minimize the reverse KL divergence, given by the following equation:

$$\mathcal{L}(\theta) = D + KL(p_y(y;\theta)p_x^*(x)) = \mathbb{E}_{p_y(y;\theta)}[\log p_y(y;\theta) - \log p_x^*(x)].$$

Because we cannot sample from the target distribution, we want to instead reparameterize this expectation to be with respect to the base distribution $p_x(x;\psi)$. Given $x = f^{-1}(y)$, the change of variables formula tells us that

$$p_x(x;\psi) = p_y(y;\theta)|\det J(f(x;\phi))| \tag{7}$$
$$\implies p_y(y;\theta) = \frac{p_x(x;\psi)}{|\det J(f(x;\phi))|}.$$

Using this equation, the expectation in our loss function can be rewritten as

$$\mathcal{L}(\theta) = \mathbb{E}_{p_x(x;\psi)}\left[\log \frac{p_x(x;\psi)}{|\det J(f(x;\phi))|} - \log p_x^*(x)\right]$$
$$= \mathbb{E}_{p_x(x;\psi)}\left[\log p_x(x;\psi) - \log|\det J(f(x;\phi))| - \log p_x^*(x)\right]$$

This loss function can also be minimized iteratively via stochastic gradient descent.

5.3. Representative Power of Normalizing Flows

We reproduce the experiments in section 6.1 of (Rezende & Mohamed, 2016) to show that our implementation matches the performance displayed in the paper. Each energy function represents interesting properties. The first is symmetric, with a large gap between the two regions of high density; the second has periodic properties, and the second has periodic properties with some variation. In figures 5 and 6 we display the samples generated by the trained normalizing flow with parameters $K = 32$ for 32 planar flow layers, and 4000 epochs during the training process alongside the true potential energy function. To see an animation of how the samples get transformed through each individual flow, see the accompanying jupyter notebook https://colab.research.google.com/drive/1KrovUf2mh-x8DWNqj3LWc-i48o5jpcFt?usp=sharing.

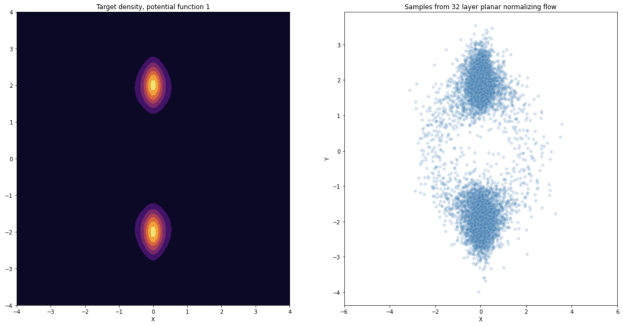

Figure 5. Samples (right) generated from the trained normalizing flow with 32 planar layers on the first potential energy function (left) from section 6.1 of (Rezende & Mohamed, 2016).

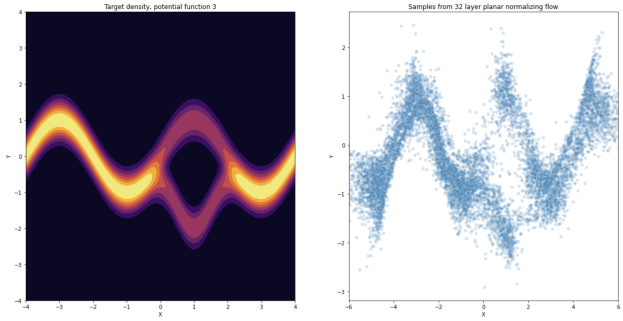

Figure 6. Samples (right) generated from the trained normalizing flow with 32 planar layers on the third potential energy function (left) from section 6.1 of (Rezende & Mohamed, 2016).

6. Experiments

We demonstrate the benefit of using normalizing flows for variational inference by comparing its performance on a Gaussian mixture model. A Gaussian mixture model describes a distribution of the form

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k), \tag{1}$$

where π_k represents the mixing coefficients and $\sum_{k=1}^{K} \pi_k = 1$. Each cluster is a normal distribution parameterized by mean μ_k and covariance Σ_k.

In our experiment setup, we use variational inference to approximate a Gaussian mixture model with 4 clusters. The density plot of the original distribution can be seen in Figure

7 (top-left). The top-right plot in Figure 7 shows the distribution converged upon by the No U-Turn Sample (Hoffman et al., 2014) after 2000 samples.

When performing mean-field variational inference, we use ADVI (Kucukelbir et al., 2016) on the Gaussian mixture model. The variables being estimated are the mixing components π_i, the means μ_i, and the covariances Σ_i. These variables depend on the number of components initially specified; the result of running ADVI on a 2-component Gaussian mixture model is displayed in Figure 7 (middle-left), and the result of running ADVi on a 4-component Gaussian mixture model in displayed in Figure 7 (middle-right). We see that in the 2 component model, the 3 smaller components of the original distribution are roughly averaged together. Because the 2 component model does not contain any distributions that can closely approximate the target distribution, the distribution that ADVI converges to is a poor approximation.

We can compare these distributions to the distributions produced by fitting 4 planar flows (Figure 7, bottom-left) and 32 planar flows (Figure 7, bottom-right) to the Gaussian mixture model. We see that 32 flows succeeds at capturing the 4 clusters, which shows that our normalizing flows model grows more expressive as we chain more transformations. Normalizing flows could produce an even better approximation of the target distribution given more layers. Through this example, we see the benefit of using normalizing flows, which does not require a specification of the number of clusters, in comparison to mean-field ADVI.

7. Conclusion

Normalizing flows transform simple probability distributions into complex ones through a series of invertible and differentiable mappings and can be used to produce rich posterior approximations for variational inference. We provide a mathematical and visual tutorial to help understand how normalizing flows work and how they can be used for variational inference by illustrating the transformation of probability distributions and working through the mathematics of planar and radial flows. For a more interactive experience, the interested reader than look at our code at https://colab.research.google.com/drive/1KrovUf2mh-x8DWNqj3LWc-i48o5jpcFt?usp=sharing. In future work, we aim to demonstrate variational inference with normalizing flows on a real dataset and provide the mathematical and visual explanations of more intricate, complicated flows.

7.1. Division of work

Ashay and Cindy worked equally on all aspects.

Acknowledgements

We thank Professor Tamara Broderick and Tan Zhi-Xuan for their helpful input and feedback on our project.

References

Behrmann, J., Grathwohl, W., Chen, R. T., Duvenaud, D., and Jacobsen, J.-H. Invertible residual networks. In *International Conference on Machine Learning*, pp. 573–582. PMLR, 2019.

Berg, R. v. d., Hasenclever, L., Tomczak, J. M., and Welling, M. Sylvester normalizing flows for variational inference. *arXiv preprint arXiv:1803.05649*, 2018.

Cheng, S. Change of variables in integral on n, 2013. URL https://planetmath.org/changeofvariablesinintegralonmathbbrn.

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

Grathwohl, W., Chen, R. T., Bettencourt, J., Sutskever, I., and Duvenaud, D. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

Hoffman, M. D., Gelman, A., et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

Jang, E. Normalizing flows tutorial, part 1: Distributions and determinants, Jan 1970. URL https://blog.evjang.com/2018/01/nf1.html.

Jean, N. Change of variables for normalizing flows, Oct 2018. URL https://nealjean.com/ml/change-of-variables/.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016.

Kobyzev, I., Prince, S. J., and Brubaker, M. A. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, nov 2021. doi: 10.1109/tpami.2020.2992934. URL https://doi.org/10.1109%2Ftpami.2020.2992934.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. Automatic differentiation variational inference. *arXiv preprint arXiv:1603.00788*, 2016.

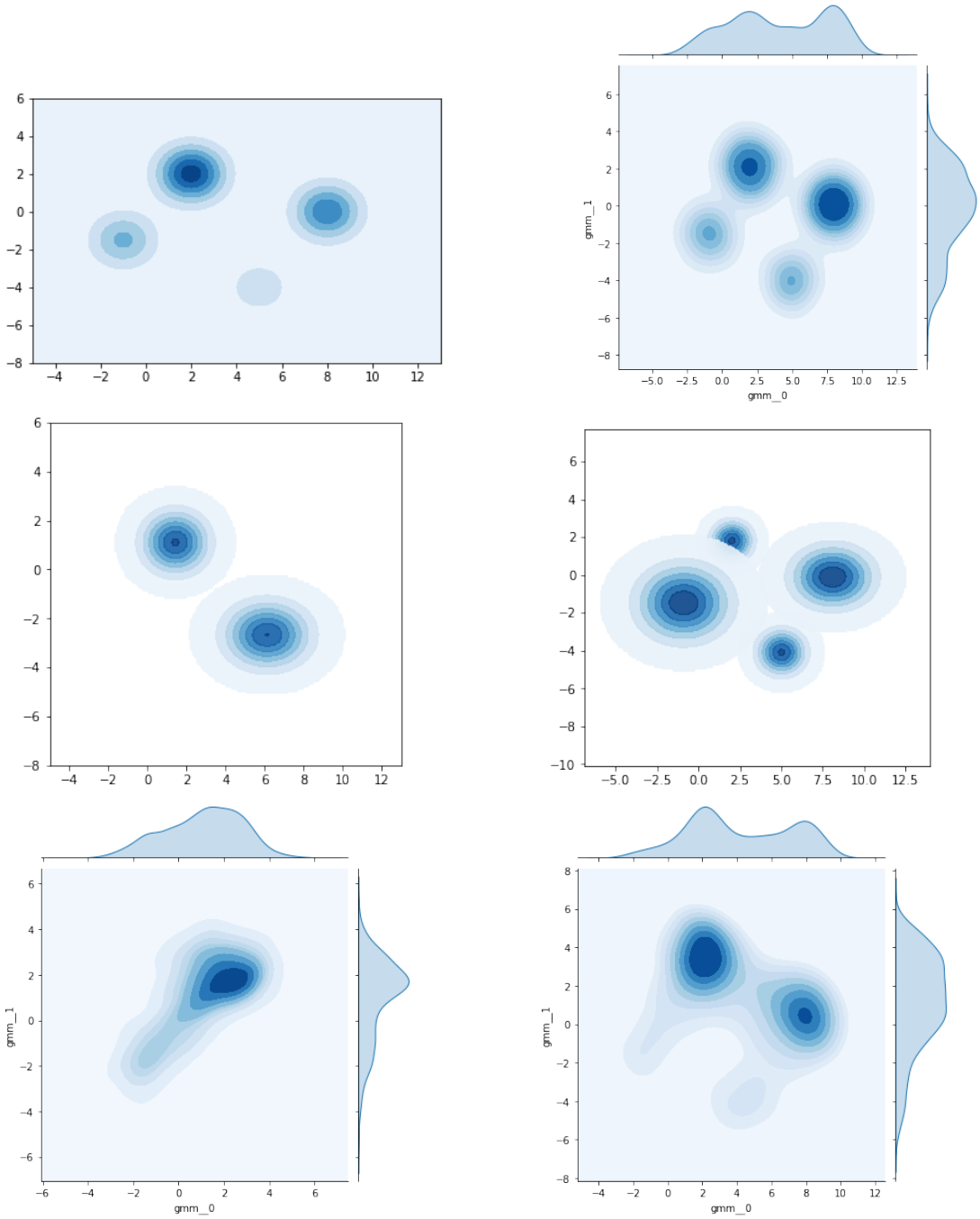

Figure 7. (top-left) Density plot of the original Gaussian mixture model. *(top-right)* Density plot produced by the NUTS sampler after 2000 samples. *(middle-left)* Density plot of the 2-cluster Gaussian mixture model estimated by ADVI. *(middle-right)* Density plot of the 4-cluster Gaussian mixture model estimated by ADVI. *(bottom-left)* Density plot produced by fitting 2 planar flows to the mixture model. *(bottom-right)* Density plot produced by fitting 32 planar flows to the mixture model.

Müller, T., McWilliams, B., Rousselle, F., Gross, M., and Novák, J. Neural importance sampling. *ACM Transactions on Graphics (TOG)*, 38(5):1–19, 2019.

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

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. 2019. doi: 10.48550/ARXIV.1912.02762. URL https://arxiv.org/abs/1912.02762.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows, 2016.

Rippel, O. and Adams, R. P. High-dimensional probability estimation with deep density models. *arXiv preprint arXiv:1302.5125*, 2013.