Normalizing Flows for Approximating Probability Densities

David Liu[*]
Department of Engineering
University of Cambridge
dl543@cam.ac.uk

Lukas Ryll
Department of Engineering
University of Cambridge
lr487@cam.ac.uk

Vincent Stimper
Department of Engineering
University of Cambridge
vs488@cam.ac.uk

Abstract

Rezende and Mohamed in [1] approximate complex distributions by transforming an initial simple distribution through invertible mappings called normalizing flows. They propose two particular types of flows, planar and radial, and show their expressivity in a variational inference setting. In this work, I reproduce their main results on approximating 2D toy probability densities, and compare their proposed flows with NICE [2], RealNVP [3] and Glow [4]. Normalizing flows are found to be able to approximate complex multimodal distributions in a 2D setting. The implementation is publicly available at `https://github.com/VincentStimper/normalizing-flows`.

1 Introduction

Normalizing flows are flexible one-to-one mappings between probability densities. Starting with a reparameterizable base distribution $q_0(\boldsymbol{z}_0)$, $\boldsymbol{z}_0$ is transformed by an invertible smooth mapping $\boldsymbol{z}_1 = f_1(\boldsymbol{z}_0)$, causing the density to 'flow' into a new shape. The basic rule of variable transforms for densities applied consecutively to $\boldsymbol{z}_k = f_K \circ f_{K-1} \circ ... \circ f_1(\boldsymbol{z}_0)$ yields

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

where $K \in \mathbb{N}$ is the length of the flow. Expectations w.r.t. $q_K(\boldsymbol{z})$ can now be taken using $q_0(\boldsymbol{z})$, known as the law of the unconscious statistician (LOTUS)

$$\mathbb{E}_{q_K}[h(\boldsymbol{z}_K)] = \mathbb{E}_{q_0}[h(f_K \circ \ldots \circ f_0(\boldsymbol{z}_0))] \tag{2}$$

The mappings are chosen such that they are expressive while having Jacobian determinants that are simple to compute. In general, computing the determinant of an $D \times D$ matrix is $O(D^3)$. For specific kinds of matrices, the computational complexity can be reduced [5].

Two types of flow mappings are introduced in [1]: the planar and radial flows

$$f_{\text{Planar}}(\boldsymbol{z}) = \boldsymbol{z} + \boldsymbol{u} \tanh(\boldsymbol{w}^\top \boldsymbol{z} + b) \qquad f_{\text{Radial}}(\boldsymbol{z}) = \boldsymbol{z} + \frac{\beta}{\alpha + r}(\boldsymbol{z} - \boldsymbol{v}) \tag{3}$$

with $\boldsymbol{u}, \boldsymbol{w} \in \mathbb{R}^D$ for $\boldsymbol{z} \in \mathbb{R}^D$ and $b \in \mathbb{R}$, $\alpha \in \mathbb{R}_+$, $\beta \in \mathbb{R}$, $v \in \mathbb{R}^D$ and $r = \|\boldsymbol{z} - \boldsymbol{v}\|$. Note that for these two flows, reversibility is guaranteed only for a range of parameters values. This constraint can be alleviated by reparameterizing the parameters so optimization is unconstrained, as shown in the appendix of [1]. The Jacobian determinant of the planar and radial flow can be computed in $O(D)$ thanks to the matrix determinant lemma used in the form $\det(I + \boldsymbol{u}\boldsymbol{v}^T) = 1 + \boldsymbol{v}^T \boldsymbol{u}$.

[*] This work contains my contributions to a joint paper reproduction project with the other authors.

Extended abstract (paper reproduction), Eastern European Machine Learning Summer School.

Dinh et al. introduced another type of flow mapping based on coupling layers as part of their models NICE [2] and RealNVP [3]. The affine coupling layer of RealNVP is given by the mapping

$$(z_1)_{1:d} = (z_0)_{1:d} \qquad (z_1)_{d+1:D} = (z_0)_{d+1:D} \odot \exp(s[(z_0)_{1:d}] + t[(z_0)_{1:d}]) \tag{4}$$

where s and t are the scaling and translation functions mapping $\mathbb{R}^d \to \mathbb{R}^{D-d}$. Here $d < D$ indicates the split of $\boldsymbol{z}$ into two vectors, and $\odot$ denotes the Hadamard product. The additive coupling layers in NICE are a special case with $s \equiv 0$. Due to the special structure, the inverse can be calculated easily and the Jacobian is triangular. There are no invertibility restrictions on s and t, so they can be parameterized by any neural network resulting in expressive mappings.

Glow [6] builds on RealNVP, extending it with invertible 1×1 convolution layers denoted by $\boldsymbol{z}' = W\boldsymbol{z}$ for some invertible matrix W acting on the channel dimension. Computing $\det(W)$ can be done efficiently when parameterized in its LU-decomposition $W = PL(U + \mathrm{diag}(\boldsymbol{s}))$ with lower and upper triangular matrices L and U respectively. W is initialized as a random orthogonal matrix, and the extracted permutation matrix P is kept fixed while the other parameters are subsequently learned.

2 Expressivity of Normalizing Flows

To visualize the flexibility of normalizing flows described in section 1 for representing complex distributions, I train a sequence of flows to transform a unit normal $q_0(\boldsymbol{z})$ to toy target distributions in two dimensions. Note the 1×1 convolutions in Glow act on the variable dimensions here and include random permutations. In [1] such permutations and orthogonal linear transformations were also applied to NICE to enhance its expressivity. NICE as implementated here only uses additive coupling layers (volume-preserving flows).

The loss function used was an annealed version of the Kullback-Leibler divergence between the flow distribution $q_\phi(\boldsymbol{z})$, with ϕ the flow parameters, and the target $p(\boldsymbol{z})$

$$L(\phi) = \mathbb{E}_{q_\phi(\boldsymbol{z})}\left[\log q_\phi(\boldsymbol{z}) - \beta_t \log p(\boldsymbol{z})\right] \tag{5}$$

with the expectation evaluated using MC samples from the flow. Using (1) and (2), this is done by sampling from the initial Gaussian and taking into account log determinant terms. Especially when dealing with multimodal targets, an unfortunate initialization can cause $q_\phi(\boldsymbol{z})$ to collapse onto one mode. The annealing parameter $\beta_t = \min(1, 10^{-2} + t/T_{ann})$ in (5) increases with the optimization iteration t until it reaches 1. Hence for small t, optimization is equivalent to maximizing the entropy of $q_\phi(\boldsymbol{z})$, making it expand in volume and helping samples from $q_\phi(\boldsymbol{z})$ cover all modes of $p(\boldsymbol{z})$ to avoid mode-collapse.

The multimodal target distributions are visualized in Figure 2. No annealing was used for the sinusoidal distributions (2-4 in Figure 2), as this was found to obtain more robust results in less training time. Presumably, this is due to the connectedness of the high-density region preventing mode-collapse. These target distributions were infinite in extent in [1], and this is no issue in (5) as only the unnormalized $\tilde{p}(\boldsymbol{z})$ is required. While training, the learned transformation would sometimes collapse to a sub-optimal solution of a straight line stretched out horizontally. By adding a big Gaussian envelope to these unnormalized distributions, this pathological behaviour was avoided.

Numerical underflow and overflow were encountered when evaluating the log probabilities of the target distributions, as samples initially may end up in very low density regions. To circumvent this problem, I applied the log sum exponential trick to stabilize logarithm terms. In addition, batch normalization layers (essentially affine coupling layers) were added every 2 flows for RealNVP and Glow (replacing ActNorm [4]) to stabilize training even further.

3 Conclusion

In this report, the basic concepts of normalizing flows were reviewed and the visualization of flows were replicated in the setting of [1]. Furthermore, other flow-based models based on coupling layers [2, 3, 4] were compared. These architectures proved to be more flexible than the planar and radial flows. In higher dimensions for planar flows, this is argued to be due to the single-unit bottleneck [7] as only one dimension is transformed each layer. Indeed, more recent work has proposed extensions to planar transformations to overcome this bottleneck [8]. The optimization procedure also proved to be nontrivial, as mode-collapse and numerical underflow arise easily when dealing with multimodal densities. The annealing procedure suggested in [1] indeed avoids sub-optimal mode-collapse.

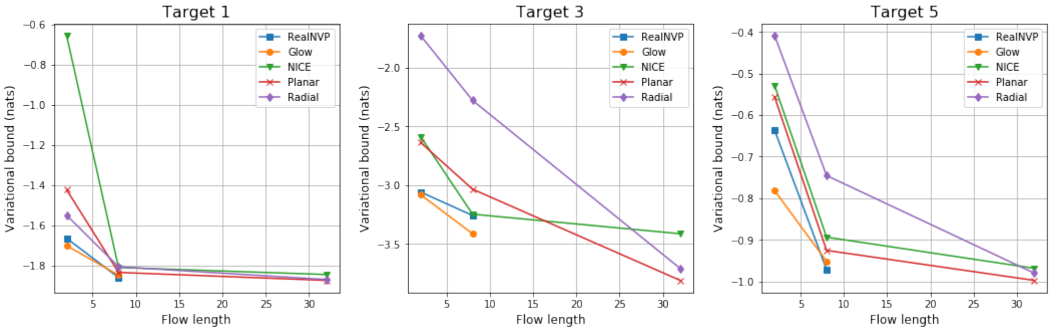

Figure 1: Comparison of the achieved variational bounds (free energy or negative ELBO) for various normalizing flow types from the literature. The planar type seems to fit the target distributions better compared to radial flows, but both improve their variational bounds with K. NICE achieves a better fit for smaller K but seems to be less expressive as the flows get deeper, which may be due to the volume-preserving nature of NICE. RealNVP and Glow show a better fit than the previous flow types in these experiments with $K = 2$ and $K = 8$, and form a class of highly expressive reversible transformations as demonstrated in the density estimation experiments performed in the original papers [3, 6].

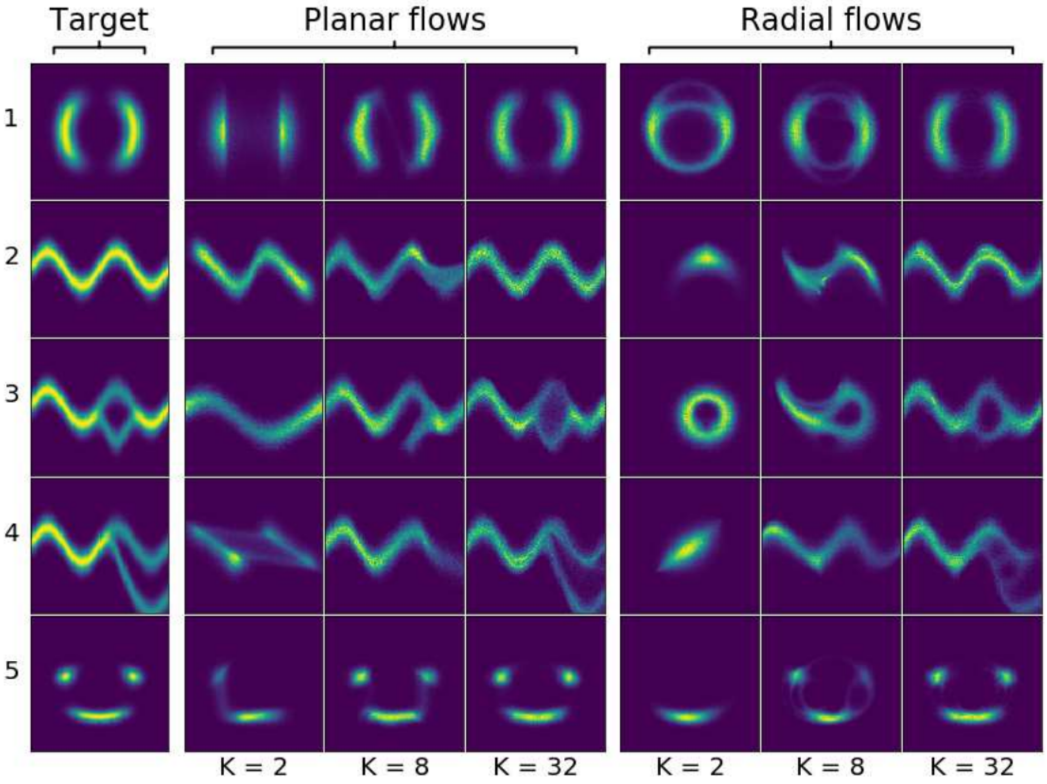

Figure 2: Planar and radial flows as introduced in [1], trained to fit given target distributions. As the number of flows K is increased, the overall flow becomes arbitrarily expressive.

References

[1] Danilo Jimenez Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. *arXiv:1505.05770 [cs, stat]*, June 2016.

[2] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear Independent Components Estimation. *arXiv:1410.8516 [cs]*, April 2015.

[3] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. *arXiv:1605.08803 [cs, stat]*, February 2017.

[4] Chin-Wei Huang, Shawn Tan, Alexandre Lacoste, and Aaron Courville. Improving Explorability in Variational Inference with Annealed Variational Objectives. *arXiv:1809.01818 [cs, stat]*, October 2018.

[5] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing Flows for Probabilistic Modeling and Inference. *arXiv:1912.02762 [cs, stat]*, December 2019.

[6] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.

[7] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751, 2016.

[8] Rianne van den Berg, Leonard Hasenclever, Jakub M Tomczak, and Max Welling. Sylvester normalizing flows for variational inference. *arXiv preprint arXiv:1803.05649*, 2018.