Networks for Variations: A Review of Normalizing Flows for Bayesian Variational Inference

Søren Winkel Holm

June 23, 2022

Introduction

In many technical fields, modelling complex systems is in recent years achieved using Deep Learning (DL) instead of setting up domain-suitable inferential statistical models [BAK18; Bre01]. The success of DL can be attributed to the potential for using similar algorithms to achieve high prediction accuracy across different big data problems [Par15]. However, a need for moving these high-accuracy, black box methods towards more robustness and explainability has been highlighted. Seeking this goal, methods have been developed for characterising complete distributions of model predictions, parametrizations or training data instead of only focusing specific realisations of these. This distributional view of DL is used in Deep Generative Modelling (DGM) and in the wider context Bayesian Machine Learning (BML). For both the problem of DGM specifically and the general task of approximating posterior distributions in BML, robust and general methods for constructing complex distributions are needed. Normalizing Flows (NF's) specify a scalable mechanism allowing for the representation of arbitrary distributions. This method is here reviewed with a focus on its' relevance for DL.

Fundamental Concepts

Let

$$\mathbf{Z} \in \mathbb{R}^D \sim p_{\mathbf{Z}}, \tag{1}$$

where $p_{\mathbf{Z}}$ is a known and analytically tractable distribution, e.g. $p_{\mathbf{Z}} = \mathcal{N}$. Now using a composition of N bijective functions $\mathbf{g} = \mathbf{g}_N \circ \cdots \circ \mathbf{g}_1$ with inverse $\mathbf{f} = \mathbf{f}_N \circ \cdots \circ \mathbf{f}_1$ and Jacobian $\mathbf{Dg}$, set $\mathbf{W} = \mathbf{g}(\mathbf{Z})$ giving the density of $\mathbf{W}$

$$p_{\mathbf{W}}(\mathbf{w}) = \mathbf{g}_\star p_{\mathbf{Z}}(\mathbf{w}) = \frac{p_{\mathbf{Z}}(\mathbf{f}(\mathbf{w}))}{|\det \mathbf{Dg}(\mathbf{f}(\mathbf{w}))|}. \tag{2}$$

In the context of NF's, $\mathbf{g}_\star p_{\mathbf{Z}}$ is named the *pushforward* of the base density $p_{\mathbf{Z}}$. $\mathbf{g}_\star p_{\mathbf{Z}}$ pushes the simple density $p_{\mathbf{Z}}$ to a possibly arbitrarily complex distribution which is called flow in the *generative direction* [KPB21] as

$$\mathbf{z} \sim p_{\mathbf{Z}} \wedge \mathbf{w} = g(\mathbf{z}) \Rightarrow \mathbf{w} \sim \mathbf{g}_{\star} p_{\mathbf{Z}}. \quad (3)$$

Inversely, $\mathbf{f}$ moves density towards the simple distribution, a process called flow in the *normalizing direction* [KPB21].

Using this construction, arbitrarily complex distributions $p_{\mathbf{W}}$ can provenly be represented [BKM07], but the functions are only considered NF's if $\mathbf{g}_i$, $\mathbf{f}_i$ and the Jacobian determinant are easy to compute [KPB21] e.g. using

$$\left|\det \mathbf{Dg}\left(\mathbf{f}(\mathbf{w})\right)\right|^{-1} = \left|\prod_{i}^{N} \det \mathbf{Df}_i\left(\mathbf{f}_{i+1} \circ \cdots \circ \mathbf{f}_N(\mathbf{w})\right)\right|. \quad (4)$$

$\mathbf{g}$ may have a parametrization ϕ, resulting in the pushforward being parameter dependant $\mathbf{g}_{\star} p_{\mathbf{Z}}(\mathbf{w}|\phi)$.

Using (2), NF's allows for density evaluation and using (3) for sampling. The first quality makes the method relevant for Variational Inference (VI) used in BML for approximating $p = p(\mathbf{w}|\mathcal{D})$ with approximate distribution

$$q^{\star} = \operatorname{argmin}_{q \in \mathcal{Q}} \mathbb{D}_{KL}[q||p] \quad (5)$$

where $\mathbb{D}_{KL}$ is the Kullback-Leibler divergence (KL) and $\mathcal{Q}$ is the variational family of possible approximations [BKM16]. Minimization of KL corresponds to maximization of the evidence lower bound (ELBO) which is, assuming this family is parametrized with ϕ,

$$\mathcal{L}(\phi) = \mathbb{E}_{q|\phi}[\ln p(\mathbf{w}, \mathcal{D})] - \mathbb{E}_{q|\phi}[\ln q(\mathbf{w}|\phi)]. \quad (6)$$

To optimize this without model-dependant derivations, gradient ascent on $\mathcal{L}$ is carried out resulting in Black-box Variational Inference (BBVI). Here, computing gradients of the form $\nabla_{\phi}\mathbb{E}_{q|\phi}[h(\mathbf{w})]$ is required. If NF's are used, such that $q(\mathbf{w}|\phi) = \mathbf{g}_{\star} p_{\mathbf{Z}}(\mathbf{w}|\phi)$, the gradients can be computed using the reparametrization trick [RM15]

$$\nabla_{\phi}\mathbb{E}_{q|\phi}[h(\mathbf{w})] = \nabla_{\phi}\mathbb{E}_{p_{\mathbf{Z}}}[h(\mathbf{g}(\mathbf{z}|\phi))]. \quad (7)$$

An alternative use for NF's is directly modelling data as a type of density estimation. Here, data likelihood is

$$\ln p(\mathcal{D}|\phi) = \sum_{i=1}^{M} \ln \mathbf{g}_{\star} p_{\mathbf{Z}}(\mathbf{y}_i|\phi) = \sum_{i=1}^{M} \left(\ln p_{\mathbf{Z}}(\mathbf{f}(\mathbf{y}_i|\phi)) + \ln\left|\det \mathbf{Df}(\mathbf{y}_i|\phi)\right|\right).$$

This model is generative using (3) and can be fitted using Maximum Likelihood Estimation (MLE).

State of the Art

NF's build on basic probablistic rules and have been used in the current form since 2010 [KPB21], but their application to BBVI was made popular in 2015 by Rezende and Mohamed [RM15]. Here, main NF's used were called planar flows and were of the form

$$\mathbf{g}(\mathbf{z}|(\mathbf{u}, \boldsymbol{\theta}, b)) = \mathbf{z} + \mathbf{u}h\left(\boldsymbol{\theta}^T\mathbf{z} + b\right) \quad (8)$$

where h is a smooth non-linearity. The term added to $\mathbf{z}$ can be considered as a single neural network unit motivating stacking this function N times to get more expressiveness in the composition. These flows have the strength of linear-time determinant computation but do not have closed form inverses [RM15, Chap. 4.1]. For running VI, however, the inverse is not needed and fast computation is key.

Empirical tests were performed, modelling the posterior distribution of deep latent Gaussian models fitted to MNIST and CIFAR-10 [RM15, Chap. 6.2]. As base density, an isotropic Gaussian was used [RM15, Chap. 6.1] and NF's show competetive performance on this task with KL and $-\ln p(\mathcal{D}_{test})$ falling systematically for higher N [RM15, Fig. 4, Tab. 2 and 3]. The choice of N governing complexity and possibility to set $\mathbf{g}$ to match distributional assumptions were highlighted as strengths compared to e.g. mean-field fixed-form BBVI.

As each flow of the form (8) has limited expressivity, a further version similar to a neural network layer with L hidden units has been proposed on the form

$$\mathbf{g}(\mathbf{z}|(\mathbf{U}, \boldsymbol{\Theta}, \mathbf{b})) = \mathbf{z} + \mathbf{U}h\left(\boldsymbol{\Theta}^T\mathbf{z} + \mathbf{b}\right) \quad (9)$$

where $\mathbf{U}, \boldsymbol{\Theta} \in \mathbb{R}^{D\times L}, \mathbf{b} \in \mathbb{R}^L$ [Ber+18, Chap. 3]. The flow was named Sylvester's flow after a determinant identity allowing the determinant computation to be efficient for low L [Ber+18, Theorem 1]. Empirical results show better approximations than the planar flows on most tasks including MNIST with parameters such as $L = 16, N = 4$ compared to planar $N = 16$ [Ber+18, Tab. 3]. NF's were also compared to plain Variational Autoencoders (VAE) with fully factorized Gaussians with NF's winning on all tasks [Ber+18, Tab. 1, Tab.2].

Same year as the renewed interest in NF's for VI, Dinh, Krueger, and Bengio [DKB15] presented Non-linear Independent Components Estimation (NICE), using NF's for DGM though not referring to the model as NF's. In this context, easy invertibility is required and expressivity has to be high, motivating the introduction of *coupling flows* defined as

$$\mathbf{g}(\mathbf{z}) = \mathbf{w}; \mathbf{w}_{\mathbb{I}} = \mathbf{h}(\mathbf{z}_{\mathbb{I}}|m(\mathbf{z}_{\mathbb{J}})), \mathbf{w}_{\mathbb{J}} = \mathbf{z}_{\mathbb{J}}, \quad (10)$$

where $\mathbf{w}$ is partitioned disjointly into $(\mathbf{w}_{\mathbb{I}}, \mathbf{w}_{\mathbb{J}})$, $\mathbf{h}(\cdot, \theta)$ is a bijection and m is any function, often a shallow neural network [DKB15, Chap. 3][KPB21, Chap.

3.4]. For different tasks, different partitionings can be used, possibly inducing structure such as pixel neighbourhoods [KPB21, Chap. 3.4]. Coupling flows along with autoregressive flows, introduced as Inverse Autoregressive Flows (IAF) by Kingma, Salimans, and Welling [KSW17], are State of The Art (SOTA) for NF's density estimation of many tabular datasets [KPB21, Tab. 2] and close to SOTA on image datasets [KPB21, Tab. 3].

Open Problems

- *How to choose the base density?* Much focus in the literature is on the choice of flows $\mathbf{g}$. The choice of base density $p_{\mathbf{Z}}$ is often not analysed, usually being a standard Gaussian. For tail behaviour, the choice of base density has been discovered to impact results [Jai+19] and $p_{\mathbf{Z}}$ should possibly seen as a form of prior on modelling behaviour [KPB21, Chap. 5.1.1] and could be adapted to the task at hand.

- *How to handle discrete distributions?* To expand the use of NF's to more tasks such as Natural Language Processing (NLP), discrete target distributions should be modelled. According to Kobyzev, Prince, and Brubaker [KPB21], this is currently an open problem. Some approximations have been succesful in specific cases such as transforming discrete variables to continous by using VAE's or adding continous noise [KPB21, Chap. 5.2.2].

- *How compute efficient are flows?* For VI and most BML, computational cost of producing a posterior distribution is the largest problem. While all NF's are presented with theoretical computational considerations for the Jacobian, comparative analysis of NF's often only focuses on final approximation accuracy. A possible lack of empirical performance analyses might stem from the software implementations being newly developed and lacking maturity or adequate hardware acceleration. However with the continued development of unifying frameworks such as Pyro Normalizing Flows [Bin+18], a timed comparison might be relevant for answering the question of what NF's to use when operating under a constrained compute budget.

References

[BAK18] Danilo Bzdok, Naomi S. Altman, and Martin Krzywinski. "Points of Significance: Statistics versus machine learning". In: *Nature Methods* 15 (2018), pp. 233–234.

[Ber+18] Rianne van den Berg, Leonard Hasenclever, Jakub M. Tomczak, and Max Welling. "Sylvester Normalizing Flows for Variational Inference". In: *UAI*. 2018.

[Bin+18] Eli Bingham et al. "Pyro: Deep Universal Probabilistic Programming". In: *Journal of Machine Learning Research* (2018).

[BKM07] Vladimir Bogachev, Alexander Kolesnikov, and Kirill Medvedev. "Triangular transformations of measures". In: *Sbornik: Mathematics* 196 (Oct. 2007), p. 309. DOI: `10.1070/SM2005v196n03ABEH000882`.

[BKM16] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. "Variational Inference: A Review for Statisticians". In: *Journal of the American Statistical Association* 112 (2016), pp. 859–877.

[Bre01] Leo Breiman. "Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author)". In: *Statistical Science* 16.3 (2001), pp. 199–231. DOI: `10.1214/ss/1009213726`. URL: `https://doi.org/10.1214/ss/1009213726`.

[DKB15] Laurent Dinh, David Krueger, and Yoshua Bengio. "NICE: Non-linear Independent Components Estimation". In: *CoRR* abs/1410.8516 (2015).

[Jai+19] Priyank Jaini, Ivan Kobyzev, Marcus A. Brubaker, and Yaoliang Yu. "Tails of Triangular Flows". In: *ArXiv* abs/1907.04481 (2019).

[KPB21] Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. "Normalizing Flows: An Introduction and Review of Current Methods". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.11 (2021), pp. 3964–3979. DOI: `10.1109/TPAMI.2020.2992934`.

[KSW17] Diederik P. Kingma, Tim Salimans, and Max Welling. "Improved Variational Inference with Inverse Autoregressive Flow". In: *ArXiv* abs/1606.04934 (2017).

[Par15] Roger Parloff. "Why Deep Learning Is Suddenly Changing Your Life". In: *Fortune* (Sept. 28, 2015). URL: `https://fortune.com/longform/ai-artificial-intelligence-deep-machine-learning/` (visited on 06/10/2022).

[RM15] Danilo Jimenez Rezende and Shakir Mohamed. "Variational Inference with Normalizing Flows". In: *ICML*. 2015.