Random Normalizing Flow
Research Internship - Master 2

Alexandre Vérine
alexandre.verine@dauphine.psl.eu

February 4, 2022

Abstract

A Normalizing Flow (NF) is a bijective deterministic mapping based on Neural Network. It is used to learn high dimensionnal probability distributions in order to perform data generation like Generative Adversarial Network (GAN). However the bijective structure often reduces the expressivity. The objective of the research internship is to study solutions to improve the expressivity of Normalizing Flows.

Normalizing Flow A *normalizing flow* [7, 4] is an invertible density model in which both density estimation and sampling can be done efficiently. In short, training a normalizing flow consists in learning an invertible mapping between a data space $\mathcal{X}$ and a latent space $\mathcal{Z}$. Typically, the forward direction $F : \mathcal{X} \to \mathcal{Z}$ (i.e. the *normalizing* direction) is tractable and exact and the inverse direction $F^{-1} : \mathcal{Z} \to \mathcal{X}$ (i.e. the *generative* direction) either has a closed form, or can be approximated using an iterative algorithm.

Suppose that P^* is the true data distribution over $\mathcal{X}$, and that P^* admits a density function denoted p^* that we wish to approximate. We then choose a simple latent d-dimensional distribution Q over $\mathcal{Z}$. For instance, we can choose a Gaussian distribution and its density function would be $q(\boldsymbol{z}) = \frac{1}{(\sqrt{2\pi})^d} e^{-\frac{1}{2}\|\boldsymbol{z}\|_2^2}$. Then, we can define $\hat{p}$, the approximation of p^*, based on q and the mapping $F : \mathcal{X} \to \mathcal{Z}$, using a simple change of variable formula:

$$\forall \boldsymbol{x} \in \mathcal{X}, \quad \hat{p}(\boldsymbol{x}) = |\det \mathrm{Jac}_F(\boldsymbol{x})| \; q(F(\boldsymbol{x})). \tag{1}$$

As seen in Equation 1, performing density estimation requires computing the determinant of the Jacobian matrix which can be large in practice, thus most normalizing flows have been specifically designed to make this computation efficient.

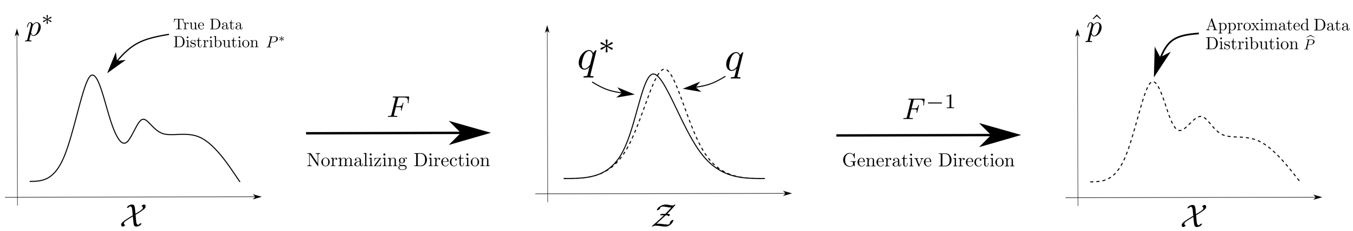

Figure 1: 1D example of a Normalizing Flow

Once the probiblity distribution is learned, it can perform density estimation by computing the estimated density of a datapoint $\hat{p}(\boldsymbol{x})$, but mostly we can generate new datapoints. Like GAN [2], a sample $\boldsymbol{z}$ in drawn from the latent distribution Q and then its image in the dataspace is computed. The main differents with GANs it the training process : Normalizing Flows are trained by maximizing the log-likelihood $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}} [\log \hat{p}(\boldsymbol{x})]$.

Objective : Improving the expressivity of Normalizing Flows Since the bijective and continuous properties of Normalizing Flows induces bi-Lipschitz constraint, the expressivity of the network can be limited in some particular pathological datasets [8]. Among all the potential remedies to tackle the limitations, some might change the latent distribution [3]. Another solution is remove the deterministic constraint on the mapping. Until now, one way is to introduce a random translation and scaling at every step of the mapping [1].

The objective of the internship would be to implement a new method to introduce randomness in the mapping : a random block based on the principle of local winner has been proven to be quite effective for other task [5, 6]. The expected work is the implementation on the LWTA block proposed in the blocks, propose a method to compute an inverse, then train the model for usual datasets such as toy datasets, MNIST or CIFAR10.

References

[1] Rob Cornish, Anthony L. Caterini, George Deligiannidis, and Arnaud Doucet. Relaxing Bijectivity Constraints with Continuously Indexed Normalising Flows. In *Proceedings of the 37th International Conference on Machine Learning*, April 2021. arXiv: 1909.13833.

[2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. In *27th Conference on Neural Information Processing Systems (NeurIPS 2014)*, June 2014. arXiv: 1406.2661.

[3] Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-Supervised Learning with Normalizing Flows. In *Proceedings of the 37th International Conference on Machine Learning*, December 2019. arXiv: 1912.13025.

[4] Ivan Kobyzev, Simon J. D. Prince, and Marcus A. Brubaker. Normalizing Flows: An Introduction and Review of Current Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. arXiv: 1908.09257.

[5] Konstantinos Panousis, Sotirios Chatzis, and Sergios Theodoridis. Nonparametric Bayesian Deep Networks with Local Competition. In *Proceedings of the 36th International Conference on Machine Learning*, pages 4980–4988. PMLR, May 2019.

[6] Konstantinos P. Panousis, Sotirios Chatzis, and Sergios Theodoridis. Stochastic Local Winner-Takes-All Networks Enable Profound Adversarial Robustness. In *arXiv:2112.02671 [cs, stat]*, December 2021. arXiv: 2112.02671.

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

[8] Alexandre Verine, Yann Chevaleyre, Fabrice Rossi, and Benjamin Negrevergne. On the expressivity of bi-Lipschitz normalizing flows. In *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, June 2021.