# Flow models

Lesson No. 3

João Victor da Silva Guerra [*]

## 1 Introduction

Flow-based generative models are powerful exact likelihood models with efficient inference and sampling steps. The exact likelihood models, such as autoregressive and non-autoregressive models, uses maximum likelihood to define the probability density function, in which later can be used to generate new data points from this distribution.

The autoregressive models, described in the previous lecture, are designed to deal with discrete data in a sequential architecture, where each output depends on a previous data data, but not on the future. These architectures has a fast evaluation of the probability density function (i.e. inference step), great compression performance and good sample quality with careful design dependence structure; however, these models has a slow sampling (i.e. sampling step) and only deals with discrete data [1], [2].

The non-autogressive flow-based models, named flow models, are designed to handle continuous data in a structure composed by a sequence of invertible functions $f_i(x)$. These architectures have been lagged behind autoregressive models in terms of probability density function $p(x)$ (i.e. inference step); however, they have an efficient sampling procedures (i.e. sampling step) and a meaningful latent representation $z$ [2], [3].

## 2 Normalizing Flow

A normalizing flow transforms a simple distribution $(x)$ into a complex distribution by applying a chain of invertible and stable transformations $(x \rightarrow f_1 \rightarrow ... \rightarrow f_k \rightarrow z)$, as shown in Fig. 1. Through this chain of transformations, a set of change of variable occur to eventually obtain a probability distribution of the latent variable $(z)$ [1], [3].

From Fig. 1, given a set of observations $(x)$ and a the target distribution $p_k(z_k)$ on a latent space, we have the

[*] RA: 117410 - j117410@g.unicamp.br

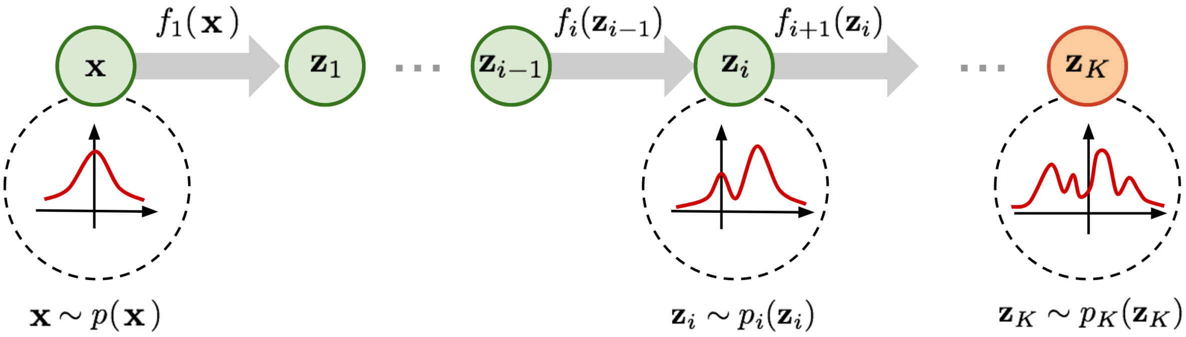

**Figure 1.** Representation of a normalizing flow. **Adapted from:** [1]

following generic change of variable:

$$
\begin{aligned}
p_i(z_i) &= p_{i-1}(f_i^{-1}(z_i)) \left| \det \frac{\delta f_i^{-1}}{\delta z_i} \right| \\
&= p_{i-1}(f_i^{-1}(z_i)) \left| \det \left( \frac{\delta f_i}{\delta z_{i-1}} \right)^{-1} \right| \\
&= p_{i-1}(f_i^{-1}(z_i)) \left| \det \frac{\delta f_i}{\delta z_{i-1}} \right|^{-1}
\end{aligned}
\tag{1}
$$

Taken this chain of probability density functions, the relationship between each pair of consecutive variables are clear. Then, we expand the equation of the latent variable $(z)$ step by step until the initial distribution $(x)$ as the following:

$$
\begin{aligned}
\log p(x) &= \log p(z) + \sum_{i=1}^{k} \log \left| \det \frac{\delta f_i}{\delta f_{i-1}} \right| \\
&= \log \mathcal{N}(f(x)|0, I) + \sum_{i=1}^{k} \log \left| \det \frac{\delta f_i}{\delta f_{i-1}} \right|
\end{aligned}
\tag{2}
$$

As core requirements of the transformation functions $f_i$, those must be easily invertible and its Jacobian determinant easy to compute in order to the normalizing flow be applicable.

## 2.1 Training procedure

With normalizing flows, the exact log-likelihood of the input data $\log p(x)$ is tractable. Such that, the training criterion of a flow-based generative model is the negative log-likelihood (NLL) over the training set $\mathcal{D}$.

$$
\text{NLL} = -\frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \log p(x) \tag{3}
$$

The metodology consists of minimizing the NLL by correcting the paramaters of the model through an stochastic optimization procedure (e.g. Stochastic Gradient Descent).

## 2.2 Sampling procedure

Since the normalizing flows are invertible, the sampling procedure is just the calculation of the invert of the composed flow functions.

$$
x = f^{-1}(z) = g(z) \tag{4}
$$

where $z \sim U(0, 1)$.

# 3 Examples

Flows models are trending today due to its fast and efficient inference and sampling setps. Here is a quick overview of today's most important implementations.

## 3.1 NICE

The NICE (Non-linear Independent Component Estimator; [4]) model implement a normalizing flow by stacking invertible bijective functions ($f : \mathbf{x} \mapsto \mathbf{z}$), known as *additive coupling layer*. The input dimensions are split into two parts: the first $d$ dimensions does not change, the second part, $d+1$ to $D$ dimensions, undergo affine transformation and the shift parameters are function fo the first $d$ dimensions.

$$
\begin{cases}
\mathbf{z}_{1:d} = \mathbf{x}_{1:d} \\
\mathbf{z}_{d+1:D} = \mathbf{x}_{d+1:D} + m(\mathbf{x}_{1:d})
\end{cases}
\iff
\begin{cases}
\mathbf{x}_{1:d} = \mathbf{z}_{1:d} \\
\mathbf{x}_{d+1:D} = \mathbf{z}_{d+1:D} - m(\mathbf{z}_{1:d}))
\end{cases}
$$

### 3.2 RealNVP

The RealNVP (Real-valued Non-Volume Preserving; [3]) model is the successor of NICE. Instead of just a shift tranformation, the RealNVP introduces an *affine coupling layer* that scales and shifts the second part of the dimensions with parameters dependent on the first part.

$$\begin{cases} \mathbf{z}_{i:d} = \mathbf{x}_{i:d} \\ \mathbf{z}_{d+1:D} = \mathbf{x}_{d+1:D} \odot \exp(s(\mathbf{x}_{1:d})) + t(\mathbf{x}_{1:d}) \end{cases} \Longleftrightarrow \begin{cases} \mathbf{x}_{1:d} = \mathbf{z}_{1:d} \\ \mathbf{x}_{d+1:D} = (\mathbf{z}_{d+1:D} - t(\mathbf{z}_{1:d})) \odot \exp(-s(\mathbf{z}_{1:d})) \end{cases}$$

In addition, this structure work in a multi-scale architecture to build a more efficient model for large datasets, which applies different operations to normal affine layers, including spatial checkerboard pattern masking, squeezing operation, and channel-wise masking.

### 3.3 Glow

The Glow ([5]) model further improves flow models, such as NICE and RealNVP, by introducing an activation normalization step, named *actnorm*, and replacing the reverse permutation on the channel ordering with invertible 1x1 convolutions.

The *actnorm* step performs an affine transformation (scale and shift) per channel for mini-batches of size 1. These scale and shift parameters are trainable, but they are initializated in a manner that the first mini-batch has mean 0 and standard deviation 1 after actnorm.

The invertible 1x1 convolution is a generalization of any permutation of the channel ordering.

### 3.4 Flow++

Currently, the Flow++ ([2]) is the state-of-the-art non-autoregressive model for unconditional density estimation. Their implementation improves upon three limiting design choices of previous models:

- uniform dequantization;
- inexpressive affine flows;
- purely convolutional coupling layers.

The design choices that closed the performance gap between autoregressive models and flow-based models are:

- variational dequantization: find an "expression" to define the dequantization;
- mixture of logistics CDF in coupling flows: more complex and non-linear transformation;
- self-attention in the conditioning networks of coupling layers: convolution + self-attention.

### 3.5 Conclusion

In recent years, flow models have begun to successfully model high dimensional raw observations from complex real-world datasets, being applicable from natural images to natural language. These models are essential part of our understanding of the deep unsupervised leaning research area, which is rapidly developing and bringing efficient solutions to previously unsolved (or even "unsolvable") problems.

## References

[1] L. Weng, “Flow-based deep generative models,” *lilianweng.github.io/lil-log*, 2018. [Online]. Available: `http://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html`.

[2] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel, *Flow++: Improving flow-based generative models with variational dequantization and architecture design*, 2019. arXiv: `1902.00275 [cs.LG]`.

[3] L. Dinh, J. Sohl-Dickstein, and S. Bengio, *Density estimation using real nvp*, 2016. arXiv: `1605.08803 [cs.LG]`.

[4] L. Dinh, D. Krueger, and Y. Bengio, “NICE: non-linear independent components estimation,” in *3rd International Conference on Learning Representations, ICLR 2015,San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[5] D. P. Kingma and P. Dhariwal, *Glow: Generative flow with invertible 1x1 convolutions*, 2018. arXiv: `1807.03039 [stat.ML]`.