Flow models

Lesson No. 3

Patrick de Carvalho Tavares Rezende Ferreira
175480

1 Introduction

Flow models are graphical representations of how information is mapped between the systems used (the original system and the compressed representation). When we want to compress information without loss, different methods can be applied, such that flow is a way to compress information without loss by applying unsupervised machine learning to find a compressed representation for the original information, so that it is obviously reversible.

Different flow models have been proposed in the literature, and the compression process is called inference and the decompression process is called sampling. There are flow models with fast sampling and fast inference (RealNVP), models with a better compression rate with a loss of speed in one of these processes (such as autoregressive models), and models that have state of the art performance in terms of compression with high-speed performance, as proposed in Flow++. These models will be described below.

2 Flow models

Flows are efficient representation techniques since they can obtain high inference sampling rates to be used in compression applications. Given an original data function $x \sim p(x)$, we can represent this function in a noise distribution $z \sim p(z)$ with a more compact representation (figure 1. We then define a flow function as a transfer function from x to z, according to the equation 1,

$$z = f(x), \tag{1}$$

$$x = f^{-1}(z) \tag{2}$$

where "f" is the flow function. Likewise, a flow function must be invertible such that the equation 2 maps the noise function back to the original space. Since the probability density function ("p") and "f" are unknown functions to us, we try to approximate it by p_θ and f_θ, according to some set of parameters θ that will be optimized during training.

$$x = f^{-1}(z) \leftrightarrow p_\theta(x)dx = p(z)dz \leftrightarrow p_\theta = p(f_\theta(x)) * \left|\frac{\partial f_\theta(x))}{\partial x}\right| \tag{3}$$

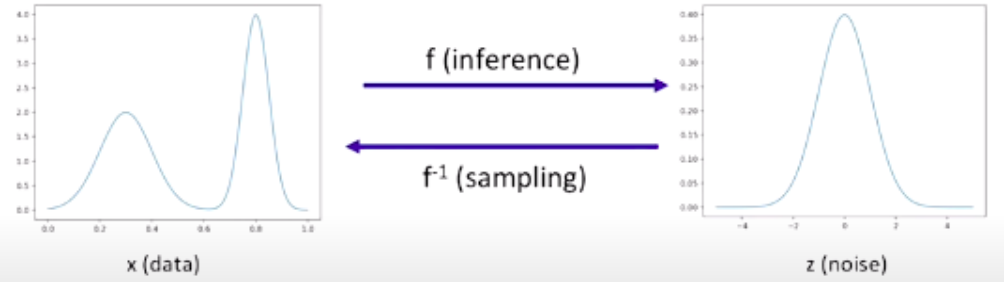

Figure 1. Sampling and inference operations[1].

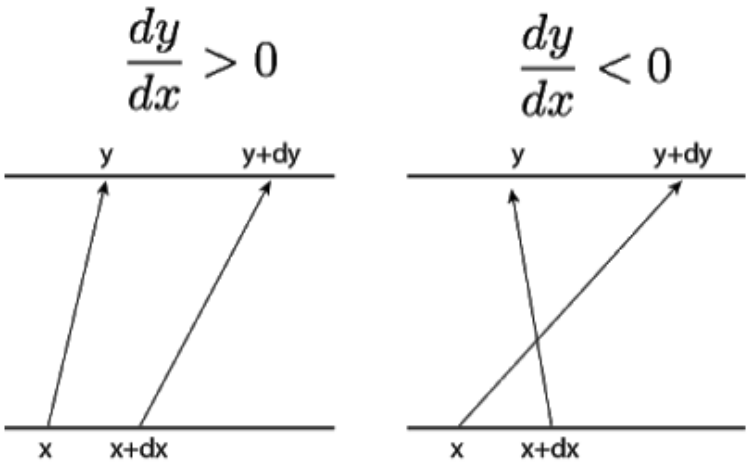

Figure 2. Mapping volume into new distribution[2].

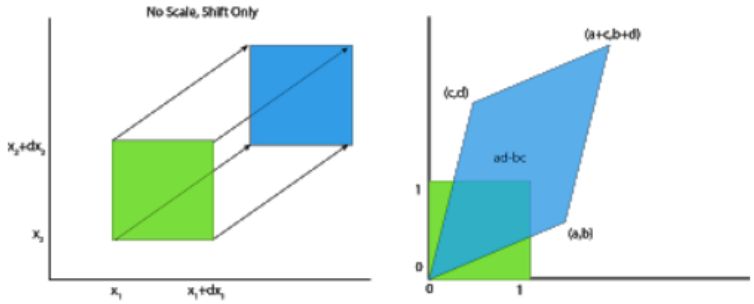

Figure 3. The use of determinant to adjust distributions volume[2].

We can then manipulate the 1 equation to show that the 3 equation is valid. We can also show that changing variables promotes the acquisition of a matrix representation in terms of the determinant, as well as in the equation 4.

$$p(x) = p(x)\frac{vol(dz)}{vol(dx)} = p(z)\left|det\frac{dz}{dx}\right| \tag{4}$$

The use of absolute in equation 4 is because we are only trying to scale the "volume" of the transformation, and the signal of this only changes the axis of the feature space, as shown in figure 2. The use of determinant also in this equation happens beacuase the determinant gives us an idea of expansion in different representations, as shown in figure 4.

$$\underset{\theta}{arg\,min}\, E_x(-log(p_\theta(x))) \tag{5}$$

This leads us to the objective of minimizing the negative likelihood of the data for the last term of the equation 3, as shown in the equation 5. We can compose the flows successively, to obtain a higher compression rate, which leads us to the equation 6.

$$log(p_\theta(x)) = log(p_\theta(z)) + \sum_{i=1}^{k} log\left|det\frac{f_i}{f_{i-1}}\right| \tag{6}$$

As mentioned earlier, there are different flow models with different characteristics and we will cover the most common ones in the following sections.

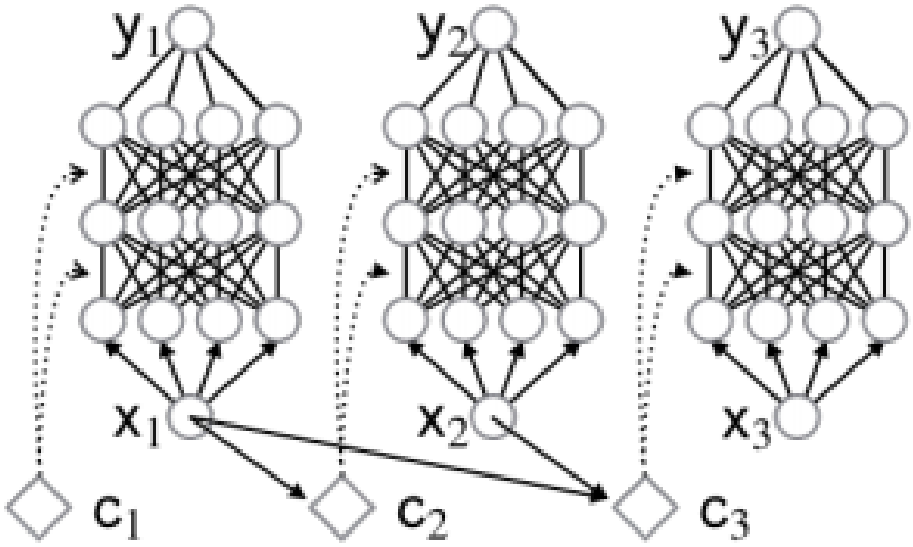

Figure 4. Autoregressive models structure example[3].

2.1 RealNVP

In order to obtain a well-succeded flow implementation, we need to construct a model capable of mapping input data into the noise distribution (inference) and get it back (sampling). A model proposed to this is RealNVP, which uses a matrix construction to data and trainable deep neural models that maps the outputs variables as a function of input ones. If we use equations 7, you can calculate Jacobian determinant more easily and optimize "s" and "t" neural networks as your parameters with some iterable method, such as SGD, minimizing the log-likelihood target.

$$\begin{cases} y_{1:d} = x_{1:d} \\ y_{d:D} = x_{d:D} \odot \exp(s_\theta(x_{1:d})) + t_\theta(x_{1:d}) \end{cases}$$

Equation 7 uses the fact that a triangular matrix has an easier to calculate the determinant, which helps when applying equation 6. This process is easy to invert, which promotes fast sampling and inference for RealNVP, although this model doesn't achieve such a high compression as autoregressive models.

2.2 Autoregressive Flow

Autoregressive models improve higher performance to flow models in comparison to RealNVP ou NICE, for example. That happens because of the bayesian array that makes the network to calculate based on previously extracted information, as shown in figure 4.

When using this approach, it is needed that the autoregressive model produces a sampling or inference process, but one of them will necessarily be slow. This happens because the final output needs to compute all the previous ones and that takes a lot of time, although the reverse process can be computed independently (either sampling or inference, respectively to direct or inverse autoregressive flow).

2.3 Flow++

Flow++ is created to solve the problem of slow sampling or inference when using autoregressive flow models, keeping their high performance as the state of the art.

The principle design choices features of Flow++ are:

- Uniform noise for dequantization;

- The use of inexpressive affine flows;
- The use of purely convolutional conditioning networks in coupling layers.

References

[1] *Week 2 CS294-158 Deep Unsupervised Learning (2/6/19)*, Feb. 2019. [Online]. Available: https://www.youtube.com/watch?v=mYCLVPRy2nc&feature=youtu.be&t=8186 (visited on 07/03/2020).

[2] Eric, *Normalizing Flows Tutorial, Part 1: Distributions and Determinants*, Library Catalog: blog.evjang.com. [Online]. Available: https://blog.evjang.com/2018/01/nf1.html (visited on 07/03/2020).

[3] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville, "Neural Autoregressive Flows," *arXiv:1804.00779 [cs, stat]*, Apr. 2018, arXiv: 1804.00779. [Online]. Available: http://arxiv.org/abs/1804.00779 (visited on 07/03/2020).