Flow Models

Lesson No. 03

Thiago Gomes Marçal Pereira [*]

1 Generative Models

Generative Model is a special kind of Unsupervised Learning, that aims to learn the true data distribution of a training set. After that, it is capable of generating new data points from this distribution.

2 Normalizing Flows

A normalizing flow transforms a simple distribution into a complex one by applying a sequence of invertible transformation functions. Flowing through a chain of transformations, we repeatedly substitute the variable for the new one according to the change of variables theorem and eventually obtain a probability distribution of the final target variable.

3 Flow Models

Flow Model is one kind of Generative Model that learns the probability density function of the real Data, and does so with the help of normalizing flows.
A Flow Model is constructed by a sequence of transformations $z = f_\theta(x)$. But we must ensure that the final flow will have a proper distribution of probabilities. So,

$$\begin{aligned} z &= f_\theta(x) \\ p_\theta(x)dx &= p(z)dz \\ p_\theta(x) &= p(f_\theta(x))\left|\frac{\delta f_\theta(x)}{\delta x}\right| \end{aligned} \tag{1}$$

This requires that f_θ is both invertible and differentiable, so that data can be correctly sampled.

4 Training

In the inference part of the model, we must maximize the logarithm probability of each flow:

$$\underset{\theta}{max} \sum_i log p_\theta(x^{(i)}) \tag{2}$$

So,

$$= \underset{\theta}{max} \sum_i log p_z(f_\theta(x^{(i)})) + log\left|\frac{\delta f_\theta}{\delta x}(x^{(i)})\right| \tag{3}$$

The change of variables principle, lets us compute the density over x

$$p_\theta(x) = p(f_\theta(x))\left|det\frac{\delta f_\theta(x)}{\delta x}\right| \tag{4}$$

[*]RA: 189691 - t189691@g.unicamp.br

Training with maximum likelihood, we get:

$$\underset{\theta}{argmin}\mathbb{E}_x[-log p_\theta(x)] = \mathbb{E}_x\left[-log p(f_\theta(x)) - log det\left|\frac{\delta f_\theta(x)}{\delta x}\right|\right] \tag{5}$$

And this adds a new requirement, that the Jacobian determinant must be easy to calculate and differentiate.

5 Sampling

As the flows are invertible, sampling is only a matter of calculating the invert of the flow functions.

$$x = f_\theta^{-1}(z) \tag{6}$$

where z is a Uniform distribution:

$$z \sim Uniform([0,1]) \tag{7}$$

6 Composition

As each flow is invertible, and composed of invertible transformations, a flow can be composed of multiple flows. This way, expressiveness can be increased and inference can have better results. Producing also better sampling.

7 Flow Model Implementations

Flow models are quite new and are increasing in usage. This results in some implementations, trying to reach the state of the art. We can list a few that have good results here:

7.1 Real NVP

Real NVP model implements a normalizing flow by stacking a sequence of invertible bijective transformation functions.

7.2 NICE

NICE model is a predecessor of RealNVP. The transformation in NICE is the affine coupling layer without the scale term, known as additive coupling layer.

7.3 Glow

This model extends the previous reversible generative models, NICE and RealNVP, and simplifies the architecture by replacing the reverse permutation operation on the channel ordering with invertible 1x1 convolutions.

7.4 Flow++

Flow++ can be currently considered the state of the art, as it provides a few improvements over other Flow models in density estimation performance, by applying:

- Variational dequantization
- Logistic mixture CDF coupling flows
- Self-attention on coupling layers

8 Conclusion

Flow Models are a powerful and evolving tool in Generative Models, being specially used in Audio and Image generation. It's an area that is also growing in popularity and need of solutions. So, knowing about these models is very helpful for our learning and future works.