[Paper review 34]

Maksed Autoregressive Flow for Density Estimation

(Papamakarios, et al. 2017)

[Contents]

1. Abstract

Autoregressive models :

- best performing neural density estimators

introduce MAF (Masked Autoregressive Flow)

- by stacking autoregressive models

  (like a Normalizing flow)

- closely related to IAF & generalization of Real NVP

2. Introduction

Neural density estimators

- readily provide exact density evaluations
- more suitable in applications when the focus is on "explicitly evaluating densities", rather than generating synthetic data

Challenges in Neural density estimators is to construct....

- 1) flexible
- 2) tractable density functions

2 families of neural density estimators, that are both flexible & tractable

- 1) autoregressive models
  - decompose joint pdf as a product of conditionals
  - model each conditional
- 2) normalizing flows
  - transform a base density into target density
  - with an "invertible" transformation with "tractable" Jacobian

View autoregressive models as a normalize flow!

- to increase its flexibility, by "stacking multiple models"
- still remains tractable

introduce MAF (Masked Autoregressive Flow)

- normalizing flow + MADE
- with MADE : enables density evaluations without sequential loop (unlike other autoregressive models)

  $\rightarrow$ makes MAF fast!

3. Background

3.1 Autoregressive density estimation

Introduction

- decompose into product of 1D conditional

  $p(\mathbf{x})=\prod_{i} p\left(x_{i} \mid \mathbf{x}_{1: i-1}\right)$.
- model each conditional $p\left(x_{i} \mid \mathbf{x}_{1: i-1}\right)$, which is a function of hidden state h_i

Drawback of autoregressive models

- sensitive to order of variables
- our approach) use a different order in each layer (random order)

Update hidden state sequentially?

- (original) required D sequential computations to compute $p(x)$
- enable parallel with drop out connections! (ex. MADE)

  $\rightarrow$ satisfies autoregressive property

$\rightarrow$ enable parallel computing on GPU

3.2 Normalizing flows

$$p(\mathbf{x})=\pi_u\left(f^{-1}(\mathbf{x})\right)\left|\operatorname{det}\left(\frac{\partial f^{-1}}{\partial \mathbf{x}}\right)\right|$$

4. Masked Autoregressive Flow

4.1 Autoregressive models as NF

Autoregressive model with conditional as a single Gaussian

$$p\left(x_i \mid \mathrm{x}_{1:i-1}\right)=\mathcal{N}\left(x_i \mid \mu_i,\left(\exp \alpha_i\right)^2\right)$$

- $\mu_i=f_{\mu_i}\left(\mathrm{x}_{1:i-1}\right)$
- $\alpha_i=f_{\alpha_i}\left(\mathrm{x}_{1:i-1}\right)$

WE can generate data, using "recursion" (express $\mathbf{x}=f(\mathbf{u})$ where $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$)

$$x_i=u_i \exp \alpha_i+\mu_i$$

- $\mu_i=f_{\mu_i}\left(\mathrm{x}_{1:i-1}\right)$
- $\alpha_i=f_{\alpha_i}\left(\mathrm{x}_{1:i-1}\right)$
- $u_i \sim \mathcal{N}(0,1)$

IAF (Inverse Autoregressive Flow)

$$u_i=\left(x_i-\mu_i\right) \exp \left(-\alpha_i\right)$$

- $\mu_i=f_{\mu_i}\left(\mathrm{x}_{1:i-1}\right)$
- $\alpha_i=f_{\alpha_i}\left(\mathrm{x}_{1:i-1}\right)$

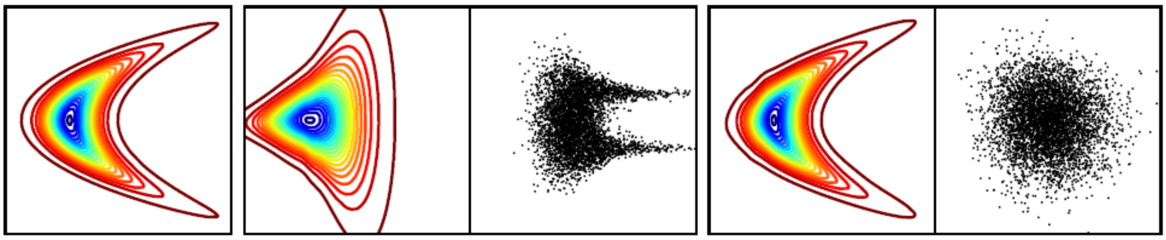
Due to autoregressive structure, the Jacobian of f^{-1} is traingular

hence, determinant can be easily obtained!

$$\left|\operatorname{det}\left(\frac{\partial f^{-1}}{\partial \mathbf{x}}\right)\right|=\exp \left(-\sum_i \alpha_i\right) \quad \text{where} \quad \alpha_i=f_{\alpha_i}\left(\mathbf{x}_{1:i-1}\right)$$

Useful diagnostic :

- step 1) transform the train data x_n into corresponding random numbers u_n
- step 2) asses whether u_n comes from independent standard normal



(a) Target density (b) MADE with Gaussian conditionals (c) MAF with 5 layers

Figure 1: **(a)** The density to be learnt, defined as $p(x_1, x_2) = \mathcal{N}(x_2 \mid 0, 4)\mathcal{N}\left(x_1 \mid \frac{1}{4}x_2^2, 1\right)$. **(b)** The density learnt by a MADE with order (x_1, x_2) and Gaussian conditionals. Scatter plot shows the train data transformed into random numbers $\mathbf{u}$; the non-Gaussian distribution indicates that the model is a poor fit. **(c)** Learnt density and transformed train data of a 5 layer MAF with the same order (x_1, x_2).

MAF (Masked Autoregressive Flow)

- implementation of stacking MADEs into a flow
- this stacking adds flexibility

4.2 Relationship with IAF

Difference

- [MAF]
  - μ_i and α_i are directly computed from previous "data variables $x_{1:i-1}$"
  - capable of calculating the density $p(x)$ of any data point in one pass

    but sampling requires D sequential passes
- [IAF]
  - μ_i and α_i are directly computed from previous "random numbers $u_{1:i-1}$"
  - sampling requires only one pass

    but calculating the density $p(x)$ of any data point requires D passes

Theoretical equivalence

- training MAF with maximum likelihood = fitting an implicit IAF to the base density
- $\pi_x(\mathrm{x})$: data density we wish to learn

  $\pi_u(\mathbf{u})$: base density

  f : transformation from u to x
- density defined by MAF

  $p_x(\mathrm{x}) = \pi_u\left(f^{-1}(\mathrm{x})\right)\left|\det\left(\frac{\partial f^{-1}}{\partial \mathrm{x}}\right)\right|$
- implicit density over u space

  $p_u(\mathbf{u}) = \pi_x(f(\mathbf{u}))\left|\det\left(\frac{\partial f}{\partial \mathbf{u}}\right)\right|$

4.3 Relationship with Real NVP

Real NVP : NF obtained by stacking coupling layers

$$\begin{aligned} \mathrm{x}_{1:d} &= \mathbf{u}_{1:d} \\ \mathrm{x}_{d+1:D} &= \mathbf{u}_{d+1:D} \odot \exp \alpha + \mu \end{aligned}$$

- $\mu = f_{\mu}\left(\mathbf{u}_{1:d}\right)$
- $\alpha = f_{\alpha}\left(\mathbf{u}_{1:d}\right)$

NICE = special case of coupling layer when $\alpha = 0$

(coupling layer : special case of both MAF and IAF)

MAF vs IAF vs Real NVP

- MAF & IAF : more flexible generalization of Real NVP
- Real NVP : can both generate data & estimate densities with only one forward pass

  (MAF : D passes to generate data(=sampling))

  (IAF : D passes to estimate densities)

4.4 Conditional MAF

conditional density estimation = task of estimating $p(x \mid y)$

- decompose as $p(\mathrm{x} \mid \mathrm{y}) = \prod_i p\left(x_i \mid \mathrm{x}_{1:i-1}, \mathrm{y}\right)$
- can turn any unconditional autoregressive model into a conditional one by augmenting its set of input variables with y
- vector y becomes an additional input for every layer
- conditional MAF significantly outperforms unconditional MAF when conditional information (such as data labels) is available

5. Summary

(from coursera)

Masked Autoregressive Flow (MAF)

Use a masked autoencoder for distribution estimation (MADE) to implement the functions f_{μ_i} and f_{σ_i}. \
For clarity, let's see how $\mathbf{x}$ is sampled. This is done as follows:

1. $x_1 = f_{\mu_1} + \exp(f_{\sigma_1})z_1$ for $z_1 \sim N(0,1)$
2. $x_2 = f_{\mu_2}(x_1) + \exp(f_{\sigma_2}(x_1))z_2$ for $z_2 \sim N(0,1)$
3. $x_3 = f_{\mu_3}(x_1, x_2) + \exp(f_{\sigma_3}(x_1, x_2))z_3$ for $z_3 \sim N(0,1)$

and so on. For the f_{μ_i} and f_{σ_i}, they use the same MADE network across the i, but mask the weights so that x_i depends on x_j for all $j < i$ but not any others. By re-using the same network, weights can be shared and the total number of parameters is significantly lower.

A note on computational complexity: determining $\mathbf{x}$ from $\mathbf{z}$ is relatively slow, since this must be done sequentially: first x_1, then x_2, and so on up to x_D. However, determining $\mathbf{z}$ from $\mathbf{x}$ is fast: each of the above equations can be solved for z_i at the same time:

$$z_i = \frac{x_i - f_{\mu_i}}{\exp(f_{\sigma_i})} \qquad i = 0, \ldots, D-1$$

Hence, the *forward* pass through the bijector (sampling $\mathbf{x}$) is relatively slow, but the *inverse* pass (determining $\mathbf{z}$), which is used in the likelihood calculations used to train the model, is fast.