Key Ideas

- Neural autoregressive flow (NAF) by Huang et al. (2018)
- PRO: universal approximator of density functions
- CONS: hyper-network \( \mapsto \) parameter num. grows quadratically
- We propose Block Neural Autoregressive Flow (B-NAF)
  - a more compact universal approximator of density functions, directly modelled as a single feed-forward network
  - comparable in performance while using orders of magnitude fewer parameters

Introduction

A normalising flows (NFs) maps two density functions via a differentiable bijection \( f \):

\[ p(y) = p(x) \left| \det J_{xf} \right|^{-1} \]

NFs are useful for learning densities: wide in density estimation and variational inference

Usually, a density is decomposed in an autoregressive way:

\[ p(y) = p(x_1) \prod_{i=2}^{d} p(x_i \mid x_1, \ldots, x_{i-1}) \]

The NF is decomposed in:

- Invertible transformer
- Conditioner

Invertibility depends on the transformers

Trivially invertible transformations may not be expressive enough

Neural autoregressive flow (NAF) by Huang et al. (2018): replaces hand-crafted transformers with invertible neural networks!

The Jacobian is computed with backpropagation:

\[ J_{xf} = \begin{bmatrix} \nabla_{x_1} \gamma & \nabla_{x_2} \gamma & \cdots & \nabla_{x_i} \gamma \end{bmatrix} \]

Method

ADVANTAGES:

- NAFs are universal approximators of density functions

DRAWBACKS:

- NAFs are hyper-networks and therefore the number of parameters scale quadratically!

SOLUTION:

- our model a universal approximator of density functions with single feed-forward network!
- we model each \( i \) directly as an NN without a conditioner
- we employ affine transformations with positive weights to process \( x \), ensuring strict monotonicity and thus invertibility

For each affine layer, the weight matrix \( W \) is a lower-triangular block matrix with strictly positive diagonal blocks:

\[ W = \begin{bmatrix} \exp(B_{11}) & 0 & \cdots & 0 \\ B_{11} & \exp(B_{22}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ B_{1l} & B_{2l} & \cdots & \exp(B_{ll}) \end{bmatrix} \]

- Universal approximator of densities: we can arbitrarily increase the hidden layer dimension
- Stable: the det-Jacobian can be computed in the log-domain
- Efficient: fewer parameters than NAF and easy-to-compute Jacobian

Fig. 1. Main differences between NAF (Huang et al., 2018) and our B-NAF.

Results

Comparison with Glow (Kingma and Dhariwal, 2018) on density estimation:

- discontinuities and low-density regions are better modelled by B-NAF

Comparison with Planar Flows (Rezende and Mohamed, 2015) on density matching:

- 2 layers of B-NAF work better than 32 layers of planar flows
- More shallow!
- Faster training!

Code available at https://github.com/nicola-decao/BNAF

Contact Information

Nicola De Cao
Ph. D. Candidate at University of Amsterdam
nicola.decao@gmail.com
https://github.com/nicola-decao
https://twitter.com/nicola_decao

References

We would like to thank George Papamakarios and Luca Falorsi for insightful discussions. This project is supported by SAP Innovation Center Network, ERC Starting Grant BrainSem (678154) and the Dutch Organization for Scientific Research (NWO) Vidi 639.022.518. Wilker Aziz is supported by the European Union’s Horizon 2020 research and innovation programme under grant agreement No 852199 (Gourmet).

Acknowledgements