Inverse Autoregressive Flow (IAF)

Yichen Ji

University of Chicago
jiyichen@uchicago.edu

October 26, 2023

Presentation Overview

Structure in Q

VAE consensus: use NNs for both encoder and decoder

Simple case: $q_\phi(\mathrm{z} \mid \mathrm{x}) = \mathcal{N}\left(\mathrm{z} \mid \mu_\phi(\mathrm{x}), \operatorname{diag}\left(\sigma_\phi^2(\mathrm{x})\right)\right)$
whose distribution parameters are outputs of the encoding MLP

More flexible specification of approximate posterior distributions?

- Structured mean-field
- Mixture model
  - require evaluation of log-likelihood and its gradients for each mixture component per parameter update = expensive
- **Normalizing Flows**
  - integrates with VAE: Encoder outputs not just the parameters of the base distribution $q_0(z)$, but also the parameters of the NF (later)

Great NF review paper as reference: Papamakarios et al. (2019) & Kobyzev et al. (2020)

Core - Change of Variables

Idea: A mechanism to construct new families of distributions by choosing an initial density and then chaining together (in)finite number of parameterized, invertible, differentiable transformations:

$$\mathrm{z}_0 \sim q\left(\mathrm{z}_0 \mid \mathbf{x}\right), \quad \mathbf{z}_i = \mathrm{f}_i\left(\mathrm{z}_{i-1}, \mathbf{x}\right) \quad \forall i = 1 \ldots D \tag{1}$$

Since invertible and differentiable transformations are composable:

$$\log q\left(\mathbf{z}_D \mid \mathbf{x}\right) = \log q\left(\mathbf{z}_0 \mid \mathbf{x}\right) - \sum_{i=1}^{D} \log \det \left| \frac{d\mathbf{z}_i}{d\mathbf{z}_{i-1}} \right| \tag{2}$$

Note that $\left| \frac{d\mathbf{z}_i}{d\mathbf{z}_{i-1}} \right|$ quantifies the relative change of volume (how much multiplication by the matrix expands or contracts space)

Nice conditions to have

in order to be practically useful

- transformation f must be invertible;
  - why crucial? Related to sampling & density estimation (come back later)
- both f and $g = f^{-1}$ must be differentiable;
  - diffeomorphism!
- easy to compute density
  - easy to compute the determinant of Jacobian
- easy to sample from
  - possible to have both?
  - typically only achieves one of them

Flow & Normalizing Flow

Where are the names coming from?

- Flow: Trajectory that a collection of samples from the base distribution $q_0(z)$ being gradually transformed by the sequence of transformations $f_1, ..., f_K$
  - Sampling operation $x = z_K = f(z_0), \quad f = f_K \circ \cdots \circ f_1$
- Normalizing Flow: The inverse flow through $f_K^{-1}, f_{K-1}^{-1}, ..., f_1^{-1}$ takes a collection of samples from $q_K(z)$ and 'normalizes' them into a collection of samples from a prescribed base $q_0(z)$ e.g. Multivariate Normal
  - Density evaluation operation $p_{\mathrm{x}}(\mathrm{x}) = p_{\mathrm{z}}\left(f^{-1}(\mathrm{x})\right)\left|\det J_{f^{-1}}(\mathrm{x})\right|$

(board)

Illustration

and what's not told in figure

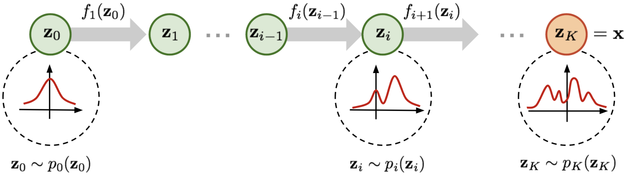


Figure: Generative direction - f pushes forward the base density $p_0(z_0)$ (noise)

The density of a sample can be evaluated by transforming it back to the base distribution and then computing the product of
i) the density of the inverse-transformed sample under the base;
ii) the associated change in volume induced by the sequence of inverse transformations.

Interplay

NF $\bigcap$ VI

What VI looks like with NF? How NF interacts with VI for modeling and inference?

This point seems like a remaining question on which we haven't meditated enough.

I found Papamakarios et al. (2019) answered this question nicely, so I stole their ideas (and notations, sorry!).

Spoil alert: Reverse K-L divergence

NF $\bigcap$ VI

with new notations...

Similarly to fitting any probabilistic model, fitting a flow-based model $p_{\mathrm{x}}(\mathrm{x}; \boldsymbol{\theta})$ to a target distribution $p_{\mathrm{x}}^*(\mathrm{x})$ can be done by minimizing some divergence or discrepancy between them.

This minimization is performed with respect to the model's parameters $\theta = \{\phi, \psi\}$, where ϕ are the parameters of invertible transformation T and ψ are the parameters of base distribution $p_{\mathrm{u}}(\mathbf{u})$.

think about VAE

In a standard Variational Autoencoder (VAE), the encoder neural network takes observed data x as input and outputs the parameters of a base distribution $p_u(u)$, commonly a Gaussian distribution. These parameters often include the mean and variance of the Gaussian.

After incorporating Normalizing Flows (NF) into a VAE, the encoder not only has to output the parameters of the base distribution $p_u(u)$, but also the parameters that define the transformations in the NF, i.e. $\theta = \{\phi, \psi\}$.

Forward K-L Divergence

Fancy name, but essentially 2 sides of K-L by asymmetry

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &= D_{\mathrm{KL}}\left[p_{\mathrm{x}}^{*}(\mathbf{x}) \| p_{\mathrm{x}}(\mathbf{x};\boldsymbol{\theta})\right] \\
&= -\mathbb{E}_{p_{\mathrm{x}}^{*}(\mathbf{x})}\left[\log p_{\mathrm{x}}(\mathbf{x};\boldsymbol{\theta})\right] + \text{ const.} \\
&= -\mathbb{E}_{p_{\mathrm{x}}^{*}(\mathbf{x})}\left[\log p_{\mathrm{u}}\left(T^{-1}(\mathbf{x};\phi);\psi\right) + \log\left|\det J_{T^{-1}}(\mathbf{x};\phi)\right|\right] + \text{ const.} \\
&\approx -\frac{1}{N}\sum_{n=1}^{N} \log p_{\mathrm{u}}\left(T^{-1}\left(\mathrm{x}_n;\phi\right);\psi\right) + \log\left|\det J_{T^{-1}}\left(\mathrm{x}_n;\phi\right)\right| + \text{ const.}
\end{aligned}
$$

Useful when: have samples from the target distribution (or the ability to generate them), but we cannot necessarily evaluate the target density $p_x^*(x)$

$$
\nabla_{\phi}\mathcal{L}(\boldsymbol{\theta}) \approx -\frac{1}{N}\sum_{n=1}^{N} \nabla_{\phi} \log p_{\mathrm{u}}\left(T^{-1}\left(\mathrm{x}_n;\phi\right);\psi\right) + \nabla_{\phi}\log\left|\det J_{T^{-1}}\left(\mathrm{x}_n;\phi\right)\right|
$$

$$
\nabla_{\psi}\mathcal{L}(\boldsymbol{\theta}) \approx -\frac{1}{N}\sum_{n=1}^{N} \nabla_{\psi} \log p_{\mathrm{u}}\left(T^{-1}\left(\mathrm{x}_n;\phi\right);\psi\right)
$$

Reverse K-L

VI: reverse KL is all you need

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &= D_{\mathrm{KL}}\left[p_{\mathrm{x}}(\mathbf{x};\boldsymbol{\theta}) \| p_{\mathrm{x}}^{*}(\mathbf{x})\right] \\
&= \mathbb{E}_{p_{\mathrm{x}}(\mathbf{x};\boldsymbol{\theta})}\left[\log p_{\mathrm{x}}(\mathbf{x};\boldsymbol{\theta}) - \log p_{\mathrm{x}}^{*}(\mathbf{x})\right] \\
&= \mathbb{E}_{p_{\mathrm{u}}(\mathbf{u};\psi)}\left[\log p_{\mathbf{u}}(\mathbf{u};\psi) - \log\left|\det J_{T}(\mathbf{u};\phi)\right| - \log p_{\mathrm{x}}^{*}(T(\mathbf{u};\phi))\right] \\
&= \mathbb{E}_{p_{\mathbf{u}}(\mathbf{u};\psi)}\left[\log p_{\mathbf{u}}(\mathbf{u};\psi) - \log\left|\det J_{T}(\mathbf{u};\phi)\right| - \log \widetilde{p}_{\mathrm{x}}(T(\mathbf{u};\phi))\right] + \text{ const.}
\end{aligned}
$$

Useful when: have the ability to evaluate the target density $p_{x}^{*}(x) = \frac{\widetilde{p}_{\mathrm{x}}(x)}{C}$ up to a normalizing constant, but not necessarily sample from it.

$$
\nabla_{\phi}\mathcal{L}(\boldsymbol{\theta}) \approx -\frac{1}{N}\sum_{n=1}^{N} \nabla_{\phi} \log\left|\det J_{T}\left(\mathbf{u}_{n};\phi\right)\right| + \nabla_{\phi} \log \widetilde{p}_{\mathrm{x}}\left(T\left(\mathbf{u}_{n};\phi\right)\right)
$$

Gradient estimate ∇_{ψ} by reparameterizing u as
$\mathbf{u} = T'\left(\mathbf{u}';\psi\right)$ where $\mathbf{u}' \sim p_{\mathbf{u}'}\left(\mathbf{u}'\right)$, like we did in VAE.
(Board)

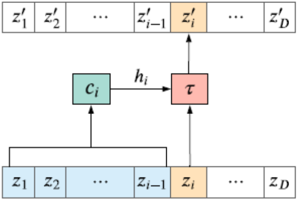
Inverse Autoregressive Flow

one example of flow construction

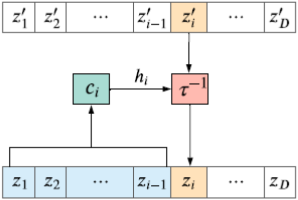
Autoregressive flow:

$$z_i' = \tau\left(z_i; \boldsymbol{h}_i\right) \quad \text{where} \quad \boldsymbol{h}_i = c_i\left(z_{<i}\right)$$

$$z_i = \tau^{-1}\left(z_i'; \boldsymbol{h}_i\right) \quad \text{where} \quad \boldsymbol{h}_i = c_i\left(z_{<i}\right) \left(IAF\right).$$



(a) Forward



(b) Inverse

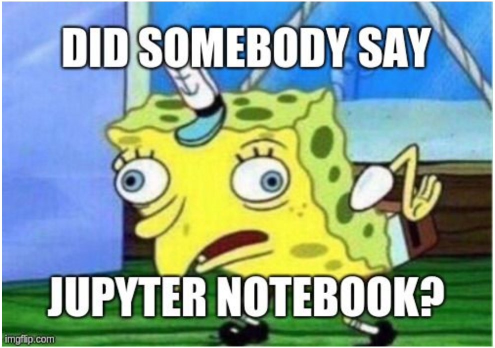
- Autoregressive Flow
  - Easy: Each z_i' can be computed in parallel (forward)
  - Hard: To compute z_i, all $z_{<i}$ need be have been computed (inverse)
- Inverse Autoregressive Flow
  - Easy: hard part of AF
  - Hard: easy part of AF

Puzzling inverse(to me, it's like inverse of inverse)!
(board to match notation in IAF paper)

Cheatcode

complexity in sampling and density evaluation

See Jupyter Notebook.



The End

Questions? Comments?