Introduction to normalizing flows and their estimation

Medha Agarwal and Garrett Mulcahy

October 18, 2022

Contents

Motivation

Motivation

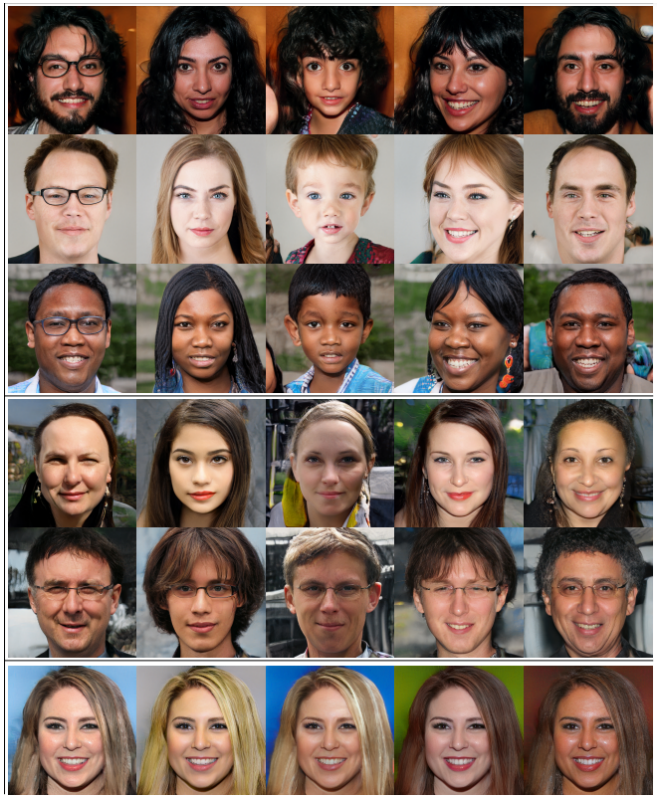

Figure: Samples from distribution of human faces (from Karras et al. (2019))

Motivation (Sampling)

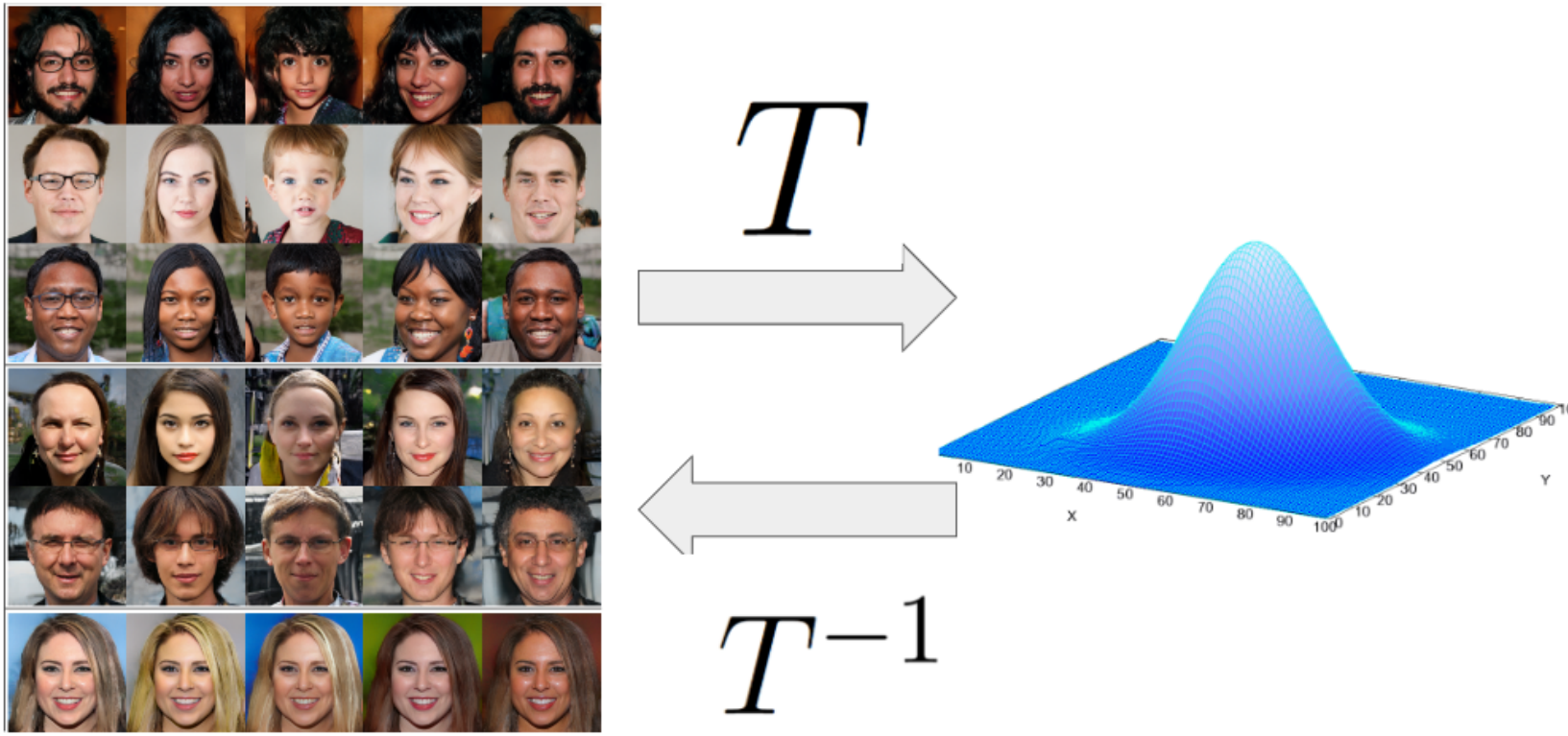

Figure: Schematic of Normalizing Flows

Motivation (Likelihood computation)

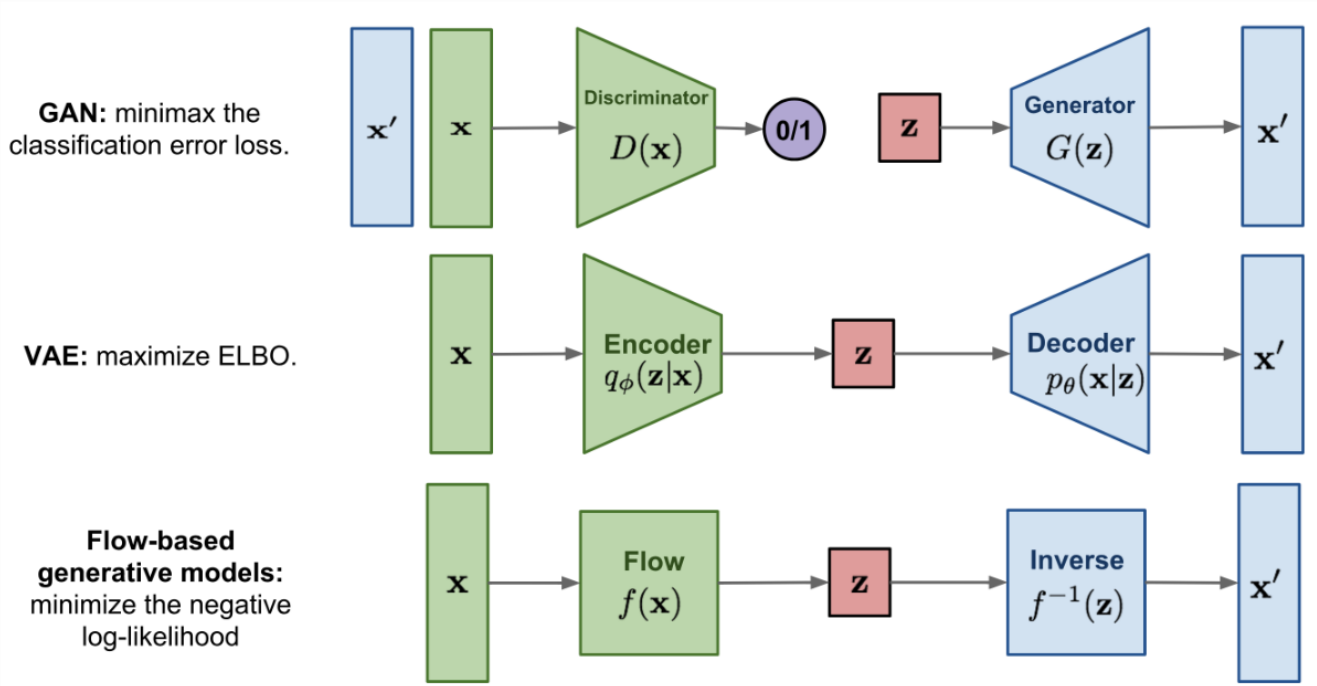

Figure: Application of Flows to Likelihood Estimation and Sampling

Preliminaries

Normalizing Flows

Conventions

- Let μ, ν be probability measures on $\mathbb{R}^d$ with Lebesgue densities p and q, respectively.
- We call μ and p the **target measure/density**, ν and q that **reference measure/density**

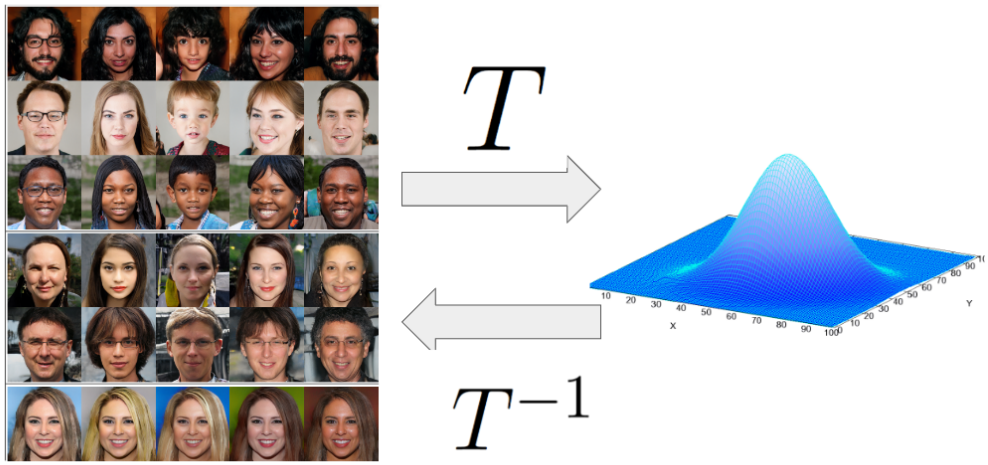

Figure: μ is the left, ν is the right

Normalizing Flows

A **normalizing flow** from p to q is a map $T : \mathbb{R}^d \to \mathbb{R}^d$ such that

(i) T is differentiable a.e. and $\det J_T(z) \neq 0$ for a.e. z

(ii) $T_{\#}\mu = \nu$, or as stated with densities,

$$q(y) = \frac{p(T^{-1}(y))}{\left|\det J_T(T^{-1}(y))\right|} \Leftrightarrow p(x) = q(T(x))\left|\det J_T(x)\right| \tag{1}$$

where J_T is the Jacobian of T. We may also write this as $q = T_{\#}p$.
We often consider the log of the quantity in (1):

$$\log p(x) = \log q \circ T(x) + \log\left|\det J_T(x)\right|$$

Composing Flows

Let $T_1, T_2, \ldots, T_n$ be normalizing flows with $T = T_n \circ T_{n-1} \circ \cdots \circ T_1$ a normalizing flow from q to p, then

$$\begin{aligned}\log p(x) &= \log q \circ T(x) + \log\left|\det J_T(x)\right| \\ &= \log q \circ T(x) + \sum_{j=1}^{n} \log\left|\det J_{T_j}(z_{j-1})\right|\end{aligned}$$

where $z_0 = x, z_1 = T_1(z_0),\ z_2 = T_2(z_1),\ \ldots,\ z_{n-1} = T_{n-1}(z_{n-2})$.

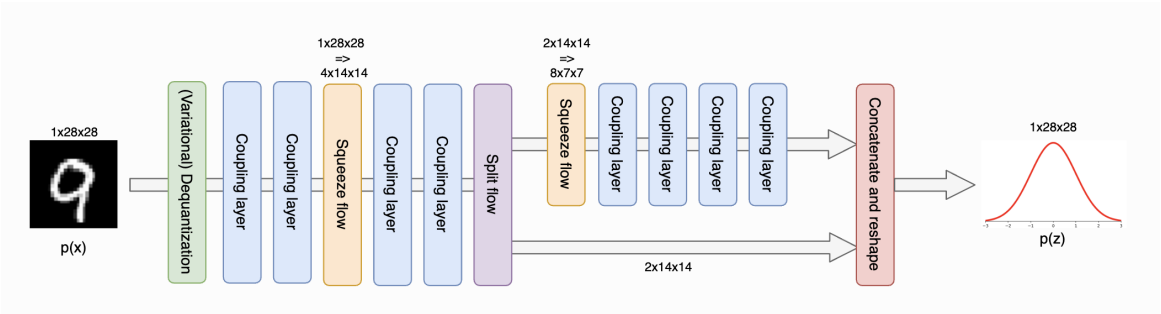

Figure: Composition of Flows

Total Variation

We define **total variation** to be

$$TV(\mu,\nu) := \frac{1}{2}|\mu-\nu|\left(\mathbb{R}^d\right) = \frac{1}{2}\|p-q\|_1.$$

Approximation results will be stated in terms of

$$TV(T_{\#}p-q) = \left\| \frac{p(T^{-1}(y))}{\left|\det J_T(T^{-1}(y))\right|} - q(y) \right\|_1.$$

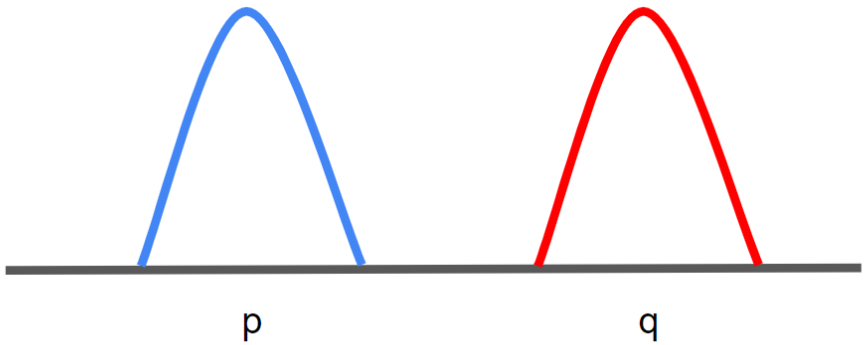

Figure: Example with $TV(\mu,\nu) = 1$

Wasserstein p-distance

We define **Wasserstein p-distance** for $p \geq 1$ and $p, q \in L^p$ to be

$$W_p(\mu, \nu) := \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} |x - y|^p \, d\pi(x, y) \right)^{1/p}.$$

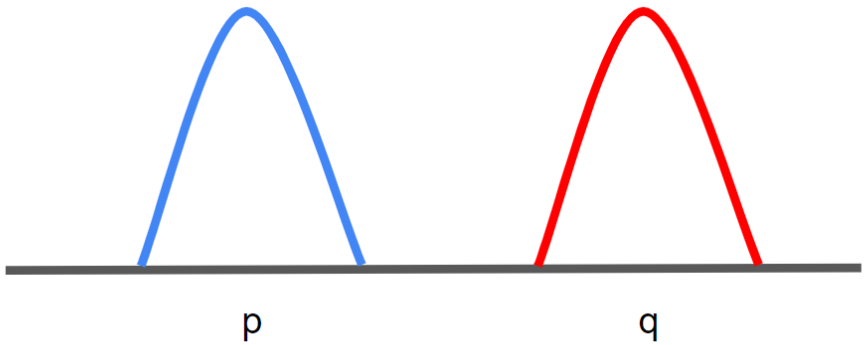

Figure: Example with $TV(\mu, \nu) = 1$

KL-divergence

Definition

Given two probability measures with densities π_1 and π_2, Marzouk et al. (2016) defines the KL-divergence from π_1 with respect to π_2 to be

$$\mathcal{D}_{\mathrm{KL}}(\pi_1|\pi_2) := \mathbb{E}_{\pi_1}\left(\log\frac{\pi_1}{\pi_2}\right) = \mathbb{E}_{\pi_1}\left(-\log\frac{\pi_2}{\pi_1}\right).$$

Properties include

(1) $\mathcal{D}_{\mathrm{KL}}(\pi_1|\pi_2) \geq 0$

(2) $\mathcal{D}_{\mathrm{KL}}(\pi_1|\pi_2) = 0$ if and only if $\pi_1 = \pi_2$

KL-divergence

Properties of KL-divergence result from Jensen's inequality:

$$
\begin{aligned}
-\mathcal{D}_{\mathrm{KL}}(\pi_1|\pi_2) &= -\mathbb{E}_{\pi_1}\left(\log\frac{\pi_1}{\pi_2}\right) \\
&= \mathbb{E}_{\pi_1}\left(\log\frac{\pi_2}{\pi_1}\right) \\
&\leq \log\left(\mathbb{E}_{\pi_1}\left(\frac{\pi_2}{\pi_1}\right)\right) && \text{since log is concave} \\
&= \log\left(\int_{\{\pi_1(x)>0\}} \pi_1(x)\frac{\pi_2(x)}{\pi_1(x)}dx\right) \\
&\leq \log(1) && \text{as log increasing} \\
&= 0.
\end{aligned}
$$

KL-divergence

From Marzouk et al. (2016), we have

Learning T via Optimization

$$\min \mathcal{D}_{\mathrm{KL}}(T_{\#}p|q)$$
$$\text{s.t. } \det \nabla T > 0 \text{ and } T \in \mathcal{F}\,.$$

Since $\mathcal{D}_{\mathrm{KL}}(T_{\#}p|q) = \mathcal{D}_{\mathrm{KL}}(p|T_{\#}^{-1}q)$, the sample-average approximation with $\{\mathbf{x}_i\}_{i=1}^n$ i.i.d. observations from μ is

$$\min -\frac{1}{n}\sum_{i=1}^{n}[\log q \circ T(\mathbf{x}_i) + \log|\det \nabla T(\mathbf{x}_i)|]$$
$$\text{s.t. } \det \nabla T > 0 \text{ and } T \in \mathcal{F}\,. \tag{2}$$

Note: Computing the objective requires calculating the determinant of the Jacobian matrix which is an $\mathcal{O}(d^3)$ task for dense matrices.

Examples of Normalizing Flows

Examples

Normalizing flows we consider include (Kong and Chaudhuri (2020),Kobyzev et al. (2020))

- Planar flows*
- Radial flows*
- Sylvester flows*
- Householder flows*
- Autoregressive Flows
- Neural Networks

Normalizing Flow Basics

- Nonlinearity function $h : \mathbb{R} \to \mathbb{R}$
- Jacobian computations require "matrix determinant lemma"

$$\det(A + uv^T) = (1 + v^T A^{-1} u) \det(A)$$

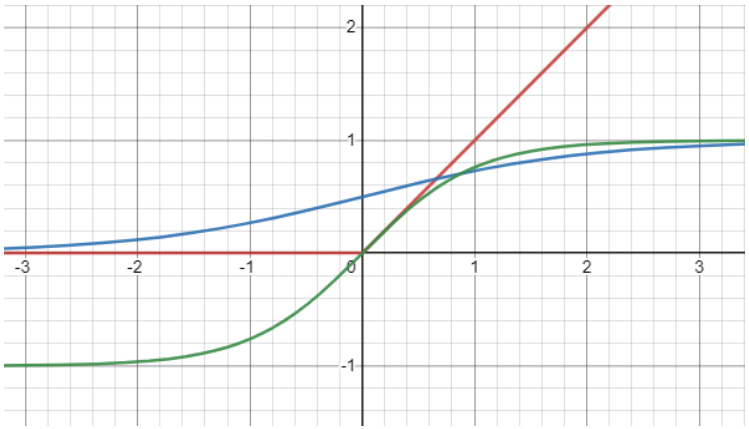

Figure: Examples of h

Planar Flows

Given $u, w \in \mathbb{R}^d$, $b \in \mathbb{R}$, and nonlinearity $h : \mathbb{R} \to \mathbb{R}$, define **planar flow** $T_{pf} : \mathbb{R}^d \to \mathbb{R}^d$ as

$$T_{pf}(z) = z + uh(w^T z + b).$$

We compute

$$J_{T_{pf}}(z) = \mathrm{Id}_d + (uw^T)h'(w^T z + b) \implies \det J_{T_{pf}}(z) = 1 + w^T u h'(w^T z + b).$$

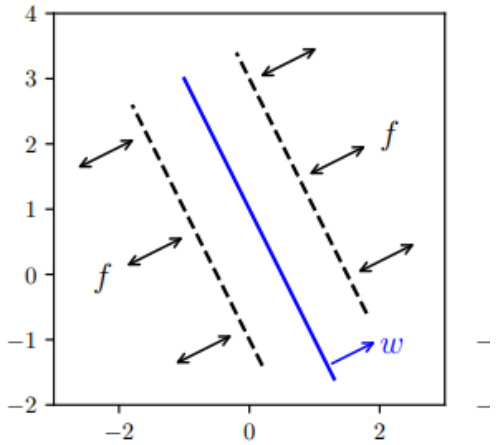

Figure: Geometric intuition from Kong and Chaudhuri (2020)

Radial Flows

Given $a \in \mathbb{R}_{>0}$, $b \in \mathbb{R}$, and $z_0 \in \mathbb{R}^d$ we define radial flow $T_{rf} : \mathbb{R}^d \to \mathbb{R}^d$ by

$$T_{rf}(z) = z + \frac{b}{a + \|z - z_0\|_2}(z - z_0)$$

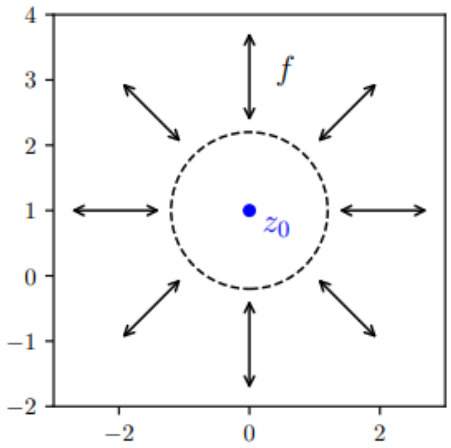

Figure: Geometric intuition from Kong and Chaudhuri (2020)

Sylvester Flows

Given a positive integer $m < d$, $A \in \mathbb{R}^{d \times m}$, $B \in \mathbb{R}^{d \times m}$, $b \in \mathbb{R}^m$, $h : \mathbb{R} \to \mathbb{R}$, define **Sylvester flow** $T_{syl} : \mathbb{R}^d \to \mathbb{R}^d$ by

$$T_{syl}(z) = z + Ah(B^T z + b).$$

We compute

$$J_{T_{syl}}(z) = \mathrm{Id}_d + A\mathrm{diag}(h'(B^T z + b))B^T,$$

By Sylvester's determinant identity (Kobyzev et al. (2020)), we get that

$$\begin{aligned} \det J_{T_{syl}}(z) &= \det(\mathrm{Id}_d + A\mathrm{diag}(h'(B^T z + b))B^T) \\ &= \det(\mathrm{Id}_m + \mathrm{diag}(h'(B^T z + b))B^T A). \end{aligned}$$

Householder Flows

Given a unit vector $v \in \mathbb{R}^d$, we define the householder flow $T_{hh} : \mathbb{R}^d \to \mathbb{R}^d$ by

$$T_{hh}(z) = z - 2vv^T z.$$

We compute

$$J_{T_{hh}}(z) = \mathrm{Id} - 2vv^T = \mathrm{Id} + (-\sqrt{2}v)(\sqrt{2}v)^T$$

and thus

$$\det J_{T_{hh}}(z) = 1 + (-\sqrt{2}v)^T(\sqrt{2}v) = 1 - 2v^T v = -1.$$

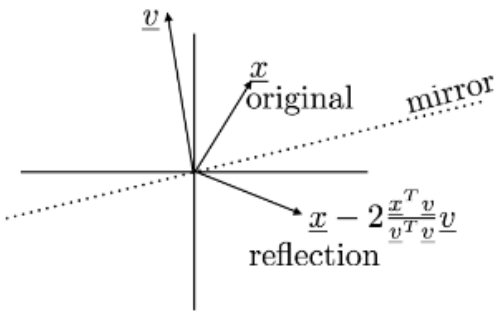

Figure: Geometric intuition

Neural Networks

- The coupling layers of normalizing flows can be modeled using neural networks.
- Typically, neural networks are not invertible. However, invertibility is often ensured by showing that the network is bijective.

Lemma

If $NN() : \mathbb{R} \to \mathbb{R}$ is a multilayer percepton, such that all weights are positive and all activation functions are strictly monotone, then $NN(\cdot)$ is a strictly monotone function.

Other options for forcing invertibility include

- Force $\text{NN}(\cdot) : \mathbb{R} \to \mathbb{R}_{>0}$ and **integrate**
- Force $\text{NN}(\cdot) : \mathbb{R} \to \mathbb{R}$ to be convex and **differentiate** (input convex neural networks)

Autoregressive Flows

Autoregressive flows are essentially triangular flows. Calculating the determinant of the Jacobian is an $\mathcal{O}(d)$ operation.

Lemma

If μ and ν are absolutely continuous Borel probability measures on $\mathbb{R}^d$, then there exists an increasing triangular transformation $T : \mathbb{R}^d \to \mathbb{R}^d$, such that $\nu = T_{\#}\mu$. This transformation is unique up to null sets of μ. A similar result holds for measures on $[0, 1]^d$.

Proof of Universality

- First show that the function class considered is dense in the set of all monotone triangular functions in the pointwise convergence topology.
- Then by Lemma 2, $\exists\, T^* \in \mathcal{F}$ such that $T * (X) \sim \nu$.
- By denseness of $\mathcal{F}$, there exists a sequence of functions $\{T_n\} \subset \mathcal{F}$ such that $T_n \to T^*$ pointwise as $n \to \infty$.
- Using dominated convergence theorem, followed by Pormanteau's theorem, we have that $T_n(X) \xrightarrow{d} T^*(X)$.

Challenges

- Universality results are not as impressive as they appear!
- Existence of a solution does not give any idea about how the expressiveness of the function class is related to its complexity.
- It is possible that even to estimate simple transformation in the function class, the model needs depth that is beyond computational reason.
- Kong and Chaudhuri (2020) describe the expressive power of a simple class of normalizing flows.

Statement of Approximation Results

$d = 1$

When $d = 1$, planar flows are universal approximators under moderate assumptions on p and q.

Theorem 3.1, Kong and Chaudhuri (2020)

Let p and q be densities on $\mathbb{R}$ such that $\operatorname{supp} p$ is contained in a finite union of intervals and $\operatorname{supp} q = \mathbb{R}$. Then for all $\epsilon > 0$ there exists a planar flow T_{pf} such that $\|(T_{pf})_{\#}q - p\|_1 \leq \epsilon$.

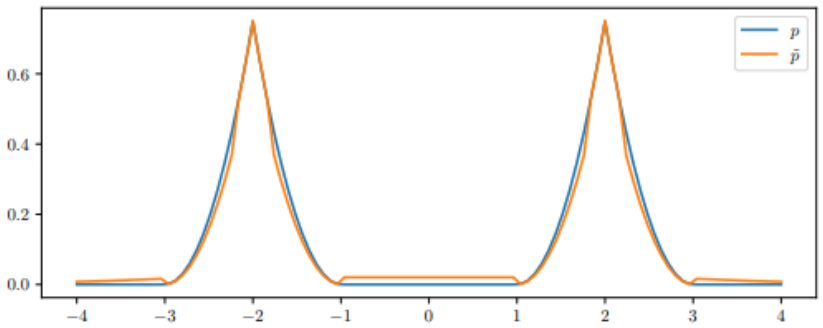

Figure: Illustrative example with $\tilde{p} = T_{\#}q$

$d > 1$

When $d > 1$, Kong and Chaudhuri (2020) prove negative results about various flows based on a topology matching condition. A sampling of these results include

Corollary 4.1.1, Kong and Chaudhuri (2020)

If p and q are two mixture of Gaussian distributions, there generally does not exist a finite composition of Sylvester flows f such that $f_{\#}q = p$.

Corollary 4.2.1, Kong and Chaudhuri (2020)

If $p \sim N(0, \Sigma_p)$ and $q \sim N(0, \Sigma_q)$ are such that $\Sigma_q^{-1} - \Sigma_p^{-1}$ has high rank, then it is impossible with a limited number of either planar flows or Sylvester flows to transport q to p.

Estimation of KR maps with ICNNs

Setup

- The goal is to describe a function class $\mathcal{F}$ such that for each $T \in \mathcal{F}$, T^j is a function of $x_{1:j}$ and is a monotonically increasing function of x_j.
- Let each $T \in \mathcal{F}$ be indexed by its model parameters θ s.t. $\forall\, \theta \in \Theta$, $T(\theta) \in \mathcal{F}$.
- Then the optimization objective as a function of model parameters is

$$KL(\theta) = -\frac{1}{n} \sum_{j=1}^{n} \left(\log q \circ T(\theta)(\mathbf{x}_j) + \sum_{i=1}^{d} \log(\nabla T(\theta))_i(\mathbf{x}_j) \right). \tag{3}$$

- Now, since $T(\theta)$ is a triangular map, it can be expressed as

$$S(\theta)(\mathbf{x}) = \left(T^1(x_1), T^2(x_{1:2}), \ldots, T^j(x_{1:j}), \ldots, T^d(\mathbf{x}) \right)^T.$$

Input Convex Neural Network

- Input convex neural networks (ICNN) proposed by Amos et al. (2017) are scalar-valued neural networks $f(x, y; \theta)$ with inputs x and y, and are defined by the model parameters θ.
- $f(x, y; \theta)$ is a convex function of y.
- Now each component T^j can be modelled as the partial derivative (in jth input) of a partial ICNN which takes as input $x_{1:j}$ and is convex in the input x_j.
- That is, if $f^j(x_{1:j-1}, x_j; \theta_j)$ is the PICNN convex in x_j, then $T^j(x_{1:j}) = \partial f^j(x_j, x_{1:j-1}; \theta^j)$.
- This implies that $\theta = (\theta_1, \ldots, \theta_d)$ fully parameterizes the KR map, though some constraints need to be imposed on Θ that we will discuss in next slide.

Input Convex Neural Network

- PICNN $f(x, y; \theta)$ is described by two set of hidden layers $\{u_i\}_{i=1}^k$ and $\{z_i\}_{i=1}^k$ that correspond to the x-path and y-path respectively.
- The architecture of a K-layer PICNN is given by the following following recurring hidden units

$$u_{i+1} = \tilde{g}_i(\tilde{W}_i u_i + \tilde{b}_i)$$

$$z_{i+1} = g_i \left(W_i^{(z)} \left(z_i \circ [W_i^{(zu)} u_i + b_i^{(zu)}] \right) + W_i^{(y)} \left(y \circ [W_i^{(yu)} u_i + b_i^{(yu)}] \right) + W_i^{(u)} u_i + b_i \right),$$

where $\tilde{g}_i$ and g_i are non-linear activation functions. The final scalar-valued output of the PICNN is $f(x, y; \theta) = z_K$.

Training

Proposition 1, Amos et al. (2017)

The function f is convex in y provided that all $\{W_i^{(z)}\}_{i=1}^{k-1}$ are non-negative, and all functions $\tilde{g}_i$ are convex and non-decreasing.

- The proof follows the simple idea that the convex combinations and compositions of convex functions is also convex.
- To ensure that f^j is a convex function of x_j, we need to constrain all its entries of $W^{(z)}$ to be non-negative.
- The learning process optimizes the parameters θ such that $KL(\theta)$ from (3) is minimized.
- A regularization term is added to ensure that the $W^{(z)}$ weight matrix entries for all layers of all d PICNN is positive giving

$$-\frac{1}{n}\sum_{i=1}^{n}\left(\log g \circ T(\theta)(\mathbf{x}_i) + \sum_{j=1}^{d}\log(\nabla T(\theta))_j(\mathbf{x}_i)\right) + \lambda\sum_{j=1}^{d}\sum_{k=1}^{K}\| \max(-(W_j)_k^{(z)}, 0)\|_F^2 , \tag{4}$$

where λ is the regularization tuning parameter.

References

Amos, B., Xu, L., and Kolter, J. Z. (2017). Input convex neural networks. In *International Conference on Machine Learning*, pages 146–155. PMLR.

Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405.

Kobyzev, I., Prince, S. J., and Brubaker, M. A. (2020). Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979.

Kong, Z. and Chaudhuri, K. (2020). The expressive power of a class of normalizing flow models. *International Conference on Artificial Intelligence and Statistics*, 108:3599–3609.

Marzouk, Y., Moselhy, T., Parno, M., and Spantini, A. (2016). An introduction to sampling via measure transport. *Handbook of Uncertainty Quantification*, 1.