Introduction to Normalizing Flows

Medha Agarwal & Garrett Mulcahy

October 18 2022

Motivation

Two major goals that drive research in normalizing flows are sampling new data and likelihood estimation based on given finite samples from a complex target distribution. However, in many real-world applications, the data generating distribution is either complicated to evaluate or completely unavailable. While there is a volume of literature on sampling algorithms, like Markov chain Monte Carlo (MCMC) and sequential Monte Carlo, these methods usually rely on density evaluation for each iteration. This is computationally expensive in case of complex target densities and impossible when the target measure is unknown. An efficient and exact way to characterize such complex target measures is to construct a transport map between the target distribution and a simple reference distribution.

Since in many practical applications it is impossible to evaluate the target density and only a finite number of samples are available from the target distribution, the transport maps are a pushforward from the target to the reference measure. This is called inverse transport and its approximate estimation can be cast as an optimization problem. The structure of the flow depends on both the target and reference measure as well as the cost function used to express the optimization objective. Recall from previous meeting, that the existence of the optimal transport in case of quadratic cost function is guaranteed by Brenier's theorem. Further, Carlier et al. (2010) show that the sequence of Brenier maps minimizing the weighted quadratic cost function converges to the triangular flow, called Knothe-Rosenblatt rearrangement.

In deep learning paradigm, the class of generative models that strive to estimate these transport maps are dubbed as normalizing flows. They are usually modeled as a sequence of simple invertible transformations from the target to normal distribution, hence the name normalizing flows. At this junction, it is important to point the deviation of normalizing flows from the optimal transport theory. Let μ be the target measure and ν be the reference measure on $\mathbb{R}^d$. The subspace of measures on $\mathbb{R}^d \times \mathbb{R}^d$ with marginal measures μ and ν is denoted by $\Pi(\mu, \nu)$. Unlike optimal transport, normalizing flows do not aim to minimize a cost function over $\pi \in \Pi(\mu, \nu)$. In fact, implementing such an optimization problem is not straightforward in practice. Therefore, normalizing flows estimate the transport map T by minimizing a divergence metric between $T_{\#}\mu$ and ν, over all transport maps T belonging to a function class $\mathcal{F}$.

A variety of normalizing flow models have been proposed in literature (see Papamakarios et al. (2021), Kobyzev et al. (2020) for a detailed overview). Kong and Chaudhuri (2020) provide an exposition on the expressive power of simple normalizing flows and how it varies with their complexity. Note that, usually high expression power (for example, by using neural networks) comes at the cost of ease of invertibility, i.e. finding the inverse T^{-1} becomes more difficult as we increase the complexity of T. Kong and Chaudhuri (2020) conduct a thorough analysis of expressive power of planar flows and their multi-dimensional generalizations - Sylvester and Householder flows. The

authors also point out the dearth of knowledge on universal approximation properties of the subset of $\mathcal{F}$ corresponding to all invertible functions in $\mathcal{F}$. These results will be discussed in next meeting; for now, we will present an overview on normalizing flows.

Preliminaries

To start, we will define some terminology and conventions that we will use throughout this exposition. We let μ and ν denote probability measures on $\mathbb{R}^d$ that are absolutely continuous with respect to Lebesgue measure. We let $p(x)$ denote the density corresponding to μ and $q(x)$ denote the density corresponding to ν. We will call p and μ the target density/measure and q and ν the reference density/measure. This means that q is a "nice" distribution such as a Gaussian or mixture of Gaussians, and p is a complicated distribution that we would like to learn more about.

Following the definition in Kong and Chaudhuri (2020), 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,

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

where J_T is the Jacobian of T. We may also write this as $q = T_{\#}p$. Moreover, since T is a diffeomorphism, T^{-1} exists and we have that $\det J_{T^{-1}}(y) = \frac{1}{\det J_T(T^{-1}(y))}$ giving

$$q(y) = p(T^{-1}(y))\left|\det J_{T^{-1}}(y)\right|$$

Equivalently, we can write $\mu = T_{\#}^{-1}\nu$ which allows us to write

$$\begin{aligned} p(x) &= q(T(x))\left|\det J_T(x)\right| \qquad \text{or} \\ \log p(x) &= \log q \circ T(x) + \log\left|\det J_T(x)\right|. \end{aligned} \tag{2}$$

Later on we will want to consider the composition of normalizing flows. That is, if we have $T = T_n \circ T_{n-1} \circ \cdots \circ T_1$ is a normalizing flow from p to q with each T_i satisfying (i), then by the chain rule we get

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

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

Distances in $\mathcal{P}(\mathbb{R}^n)$

The primary goal is to find a transport map T such that $T_{\#}\mu = \nu$. However, this exact computation is rarely possible and the problem of estimating T is cast as optimization where we try to minimize some measure of distance between $T_{\#}\mu$ and ν. There are several natural ways to quantify distance. We define the **total variation** between μ and ν to be

$$TV(\mu, \nu) := \frac{1}{2}|\mu - \nu|\,(\mathbb{R}^n).$$

However, since μ and ν have Lebesgue densities, we can obtain

$$TV(\mu, \nu) = \frac{1}{2}\|p - q\|_1,$$

where $\|.\|_1$ is the L^1-norm. We will drop the scaling factor of $\frac{1}{2}$ and consider the problem of minimizing the total variation as minimizing the L^1 distance between two densities. This notion is used in Kong and Chaudhuri (2020) to formulate approximation results. Using 1, given a normalizing flow f we measure how "close" $f_{\#}p$ is to q (in a sense, how "good" of a flow f is) via

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

The approximation results of Kong and Chaudhuri (2020) are then given in terms of the above quantity.

Another population notion of distance between two probability measures is Wasserstein distances. For any $p \geq 1$ we define the **Wasserstein p-distance** between two probability measures μ, ν with finite pth moment to be

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

The W_p metric metrizes weak convergence in the space of probability measures with finite pth moment (Villani (2003)). It is immediate that W_p distance is closely connected with optimal transport. There are many situations where W_p may be preferred to total variation. For example, suppose μ is a probability measure with bounded support, and let μ' be a translation of μ such that the support of μ' has empty intersection with the support of μ. Then $TV(\mu, \mu') = 1$, suggesting that the two distributions are very dissimilar. However, since there is a clear transport map between these two measures, the Wasserstein distance may be more reasonable and thus reflect the similarities between μ and μ'. However, for the purposes of this exposition we will not consider Wasserstein distance any further.

Another distance popularly used by Marzouk et al. (2016) is the **Kullback-Leibler divergence** due to its connections with likelihood function. 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

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

In a sense, KL-divergence measures the extent to which π_1 is "different" from π_2. We observe that in our application of learning normalized flows, we will want to compute the KL-divergence of $\pi_1 = f_{\#}p$ with respect to our reference measure $\pi_2 = q$. We have choice over q and often choose it to be a Gaussian. Thus, $\pi_2(x) > 0$ for all $x \in \mathbb{R}^d$ and the quantity $\mathcal{D}_{\mathrm{KL}}(\pi_1|\pi_2)$ is well-defined as we can compute the expectation by integrating over $\{x : \pi_1(x) > 0\}$.

It is important to note that KL-divergence is not a metric. However, it does satisfy some important properties. Namely, we have $\mathcal{D}_{\mathrm{KL}}(\pi_1|\pi_2) \geq 0$ and $\mathcal{D}_{\mathrm{KL}}(\pi_1|\pi_2) = 0$ if and only if $\pi_1 = \pi_2$.

This is a consequence of Jensen's inequality, as we have

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

Moreover, Jensen's inequality gives that we have equality if and only if $-\log$ is affine (which is false) or $\frac{\pi_2}{\pi_1}$ constant. Thus, $\frac{\pi_2}{\pi_1}$ is constant, but since these are both probability measures this forces that $\pi_1 = \pi_2$ a.e.

Returning to the problem of finding a normalizing flow T from target measure p to reference measure q, the KL-divergence arises in the following optimization problem stated by Marzouk et al. (2016):

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

The global minimizer of the above optimization problem is such that $\mathcal{D}_{\mathrm{KL}}(S_{\#}p|g) = 0$, i.e. $S_{\#}p = q$. Since $\mathcal{D}_{\mathrm{KL}}(T_{\#}p|q) = \mathcal{D}_{\mathrm{KL}}(p|T^{-1}_{\#}q)$, the population objective can also be written as

$$\begin{aligned}
&\min \mathbb{E}_p[-\log q \circ T(\mathbf{x}) - \log|\det \nabla T(\mathbf{x})|] \\
&\text{s.t. } \det \nabla T > 0 \text{ and } T \in \mathcal{F}\,.
\end{aligned}$$

Since the objective involves an expected with respect to the target distribution, we can use its Monte Carlo estimator to approximate the optimization objection. If $\{\mathbf{x}_i\}_{i=1}^n$ are i.i.d. observations from measure space $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d), \mu)$, then the sample-average approximation to the objective is

$$\begin{aligned}
&\min -\frac{1}{n}\sum_{i=1}^{n}[\log q \circ T(\mathbf{x}_i) + \log|\det \nabla T(\mathbf{x}_i)|] \\
&\text{s.t. } \det \nabla T > 0 \text{ and } T \in \mathcal{F}\,.
\end{aligned} \tag{3}$$

Notice that the above objective is the negative log-likelihood of data following the change of variable formula in (1). Moreover, we observe that the above quantity is given entirely in terms of known quantities: namely the samples $\{x_i\}$ from μ, the reference density q, and the normalizing flow T. Importantly, it does not require any knowledge of the target distribution p. Hence, we can learn T by solving the above optimization problem.

To summarize this section, we have introduced three common notions of distance in the space of probability measures: total variation, Wasserstein p-distance, and KL-divergence. Both total variation and Wasserstein p-normalized are metrics, whereas KL-divergence is not. KL-divergence is used in learning normalizing flows, whereas universal approximation will be stated in terms of total variation.

Normalizing Flows

Before diving into specific normalizing flow models, let us consider some key applications and what kind of properties drive research in these applications. Considering that the data generating target distribution is difficult to evaluate, the two main applications are - density estimation (for likelihood based inference) and sampling. Note that calculating the density at each data point $\mathbf{x} \in \mathbb{R}^d$, requires us to evaluate T as well as the determinant of its Jacobian at $\mathbf{x}$ using (2). Therefore, it is important that both operations are efficient. On the other hand, we can sample from μ by applying T^{-1} on a sample from ν. As a consequence, when sampling is the main goal, normalizing flows are often modelled in the generative direction (reference to target).

Recall that evaluating data likelihood (or KL divergence) to solve the optimization problem (3) requires calculating the determinant of the Jacobian matrix, which is a $\mathcal{O}(d^3)$ operation. This can be computationalluy prohobitive in high-dimensional problems. To ameliorate this problem, a triangular structure can be imposed on T so that the determinant is the product of the diagonal elements making it an $\mathcal{O}(d)$ operation. We know that Knothe-Rosenblatt map is a specific triangular map which is also monotone in lexicographic order. In fact, KR map is the *unique* global minimizer of (3) if $\mathcal{F}$ is restricted to the vector space of smooth triangular maps. Due to the desirable properties of KR map, we will later focus on its estimation via input convex neural network.

However, one must keep in mind that while being excellent to work with, the expressiveness of triangular flows is sensitive to coordinate reordering. In addition, except for universality result (Bogachev et al., 2005), not enough literature exists on relation between depth of triangular flow models and their expressiveness. Therefore, we will first study Kong and Chaudhuri (2020) to build an understanding on proof techniques one can use to study the expressiveness of simple flows. The approximation results proven in Kong and Chaudhuri (2020) not only consider certain pairs (p,q) but also specific normalizing flows. These normalizing flows will be the first four that we discuss. For completeness, we will also discuss neural networks and splines. In this section we will define the structure of each normalizing flow, discuss how we can enforce invertibility, and (when tractable) compute its Jacobian.

Planar Flows

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

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

Thus, we consider w as specifying a tangent direction, u as a scaling factor (how must to weigh the nonlinearity), and b as a bias term.

For $j \in \{1, \dots, d\}$ we compute

$$\frac{\partial (T_{pf}(z))_j}{\partial z_i}(z) = \frac{\partial}{\partial z_i}\left(z_j + u_j h(w^T z + b)\right) = \delta_{ij} + u_j w_i h'(w^T z + b)$$

and thus

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

By the matrix determinant lemma (Kobyzev et al. (2020)) we have

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

In particular, Kong and Chaudhuri (2020) will use this computation with h equal to ReLU, in which case we get

$$\det J_{T_{pf}}(z) = 1 + w^T u \cdot \mathbf{1}_{\{w^T z + b > 0\}}.$$

Under certain bounds on w and u, T_{pf} with $h = \text{ReLU}$ will be invertible as it will be Lipshitz (come back to later).

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

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

The occurrence of $\|z - z_0\|_2$ in the denominator makes computing $\det J_{rf}(z)$ unpleasant to put in a compact form.

Sylvester Flows

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

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

where we understand h as mapping coordinate-wise.

For $j \in \{1, \ldots, d\}$ we compute

$$\begin{aligned}\frac{\partial (T_{syl}(z))_j}{\partial z_i}(z) &= \frac{\partial}{\partial z_i}\left(z_j + \sum_{k=1}^{m} A_{jk} h'\left(\sum_{\ell=1}^{d} B^T_{k\ell} z_\ell + b_j\right)\right) \\ &= \delta_{ij} + \sum_{k=1}^{m} A_{jk} h'\left(\sum_{\ell=1}^{n} B_{\ell j} z_\ell + b_j\right) B^T_{ki}.\end{aligned}$$

This gives that

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

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

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

Since we have chosen $m < d$, the second formula for $\det J_{T_{syl}}(z)$ is computationally advantageous.

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

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

For $j \in \{1, \ldots, d\}$ we compute

$$\begin{aligned}\frac{\partial (T_{hh}(z))_j}{\partial z_i}(z) &= \frac{\partial}{\partial z_i}\left(z_j - \sum_{k=1}^{d} 2v_j v_k z_k\right) \\ &= \delta_{ij} - 2v_j v_i\end{aligned}$$

It follows that

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

so again by the matrix determinant lemma we have

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

Neural Networks

As reflected in the survey provided by Kobyzev et al. (2020), there has been success using neural networks to learn normalizing flows. As mentioned in the motivation, the key obstacle in using neural networks for this task is guaranteeing invertibility of the functions in the class modeled by the neural network. In 1-dimension, we have the following result: a multilayer perception $\mathrm{NN}(\cdot) : \mathbb{R} \to \mathbb{R}$ with positive weights and strictly monotone activation function is a strictly monotone function. However, forcing the weights to be positive makes training more difficult. There are two workarounds: one with integration and one with differentiation. Namely, if instead we train a neural network $\mathrm{NN}(\cdot) : \mathbb{R} \to \mathbb{R}_{>0}$ and integrate it, we obtain a strictly monotone function. Conversely, if we learn $\mathrm{NN}(\cdot) : \mathbb{R} \to \mathbb{R}$ such that NN is convex, then since we are dealing with one dimensional transport, its derivative will be an increasing function (see later section on input convex neural networks).

Splines

Splines provide another class of normalizing flows. A spline is a piecewise polynomial function which passes through a specified number of points $(x_i, y_i)_{i=0}^K$. We call these points knots. In order for a spline to be strictly increasing, it is necessary that $x_i < x_{i+1}$ and $y_i < y_{i+1}$. Kobyzev et al. (2020) reports several successful methods of using splines to learn one-dimensional normalizing flows.

Autoregressive Flows

Autoregressive flows are essentially triangular flows that are implemented by using autoregressive models in form of flows. As discussed before, the key property of autoregressive flows that make then computationally lucrative normalizing flows is the $\mathcal{O}(d)$ complexity of determinant calculation. The existing universality proofs for autoregressive flows are based on the following result by Bogachev et al. (2005)

Lemma 1. *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 $[0,1]^d$.*

For specific models, the proof of universality proceeds by showing that the function class considered is dense in the set of all monotone triangular functions in the pointwise convergence topology. Then by using Lemma 1, we know that there exists a $T^* \in \mathcal{F}$ such that $T*(X) \sim \nu$ and by denseness of $\mathcal{F}$, we have that there exists a sequence of functions $\{T_n\} \subset \mathcal{F}$ such that $T_n \to T^*$ pointwise as $n \to \infty$. Then using dominated convergence theorem, followed by Pormanteau's theorem, we have that $T_n(X) \xrightarrow{d} T^*(X)$. Huang et al. (2018) showed this for autoregressive flows constructed using monotone neural networks and Jaini et al. (2019) showed this for monotone polynomials.

However, one should keep in mind that 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, the model needs depth that is beyond computational reason. At this juncture, one can appreciate the work of Kong and Chaudhuri (2020) in describing the expressive power of a class of normalizing flows.

Approximation Results

At this moment, we now cite from Kong and Chaudhuri (2020) several approximation results. **N.B.** In these results, we are reversing the roles of the target and reference measure. Namely, a normalizing flow T will satisfy $T_{\#}q = p$.

Case $d = 1$

In short, Kong and Chaudhuri (2020) show that when $d = 1$ planar flows are universal approximators under moderate assumptions on p and q.

Theorem 1. *(Theorem 3.1) 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 $\|(T_{pf})_{\#}q - p\|_1 \leq \epsilon$.*

We note that we do not even have to compose planar flows to get universal approximation. An example of such a permissible q would be a Gaussian distribution.

Case $d > 1$

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

Theorem 2. *(Corollary 4.1.1) If p and q are two mixture of Gaussian distributions, there generally does not exist a finite composition of Sylvester flows f such that $f_{\#}q = p$.*

Theorem 3. *(Corollary 4.2.1) 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.*

Estimating KR maps using input convex neural networks

Recall that for the random variable $X \in \mathbb{R}^d$ distributed according to the absolutely continuous measure μ with Lebesgue density p, we want to find the KR map T^* such that $T^*_{\#}p = q$ for some reference density p. Estimating this KR map can be cast as an optimization problem given by (3). The goal is to describe a function class $\mathcal{F}$ with high expressive power such that for each $T \in \mathcal{F}$, the jth component of T is a monotonically increasing function of x_j and only depends on $x_{1:j}$. Let each $T \in \mathcal{F}$ be indexed by its model parameters θ such that for each $\theta \in \Theta$, $T(\theta) \in \mathcal{F}$. This allows us to define the optimization objective as a function of model parameters as

$$KL(\theta) = -\frac{1}{n}\sum_{j=1}^{n}\left(\log q \circ T(\theta)(\mathbf{x}_j) + \sum_{i=1}^{d}\log(\nabla T(\theta))_i(\mathbf{x}_j)\right). \tag{4}$$

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 networks 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. They are characterized by the unique property that $f(x, y; \theta)$ is a convex function of y. Now each component T^j can be modelled using a partial input convex neural network (PICNN) which takes as input $x_{1:j}$ and is convex in the input x_j. The model parameters of this PICNN are θ_j. That is, we can introduce the notation $f^j(x_{1:j-1}, x_j; \theta_j)$ for the PICNN used to model the jth component of T. This implies that $\theta = (\theta_1, \ldots, \theta_d)$ fully parameterizes the KR map. PICNN 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 $f(x, y; \theta)$ is given by the following following recurring hidden units

$$\begin{aligned}
u_{i+1} &= \tilde{g}_i(\tilde{W}_i u_i + \tilde{b}_i) \\
z_{i+1} &= g_i\left(W_i^{(z)}\left(z_i \circ [W_i^{(zu)}u_i + b_i^{(zu)}]\right) + W_i^{(y)}\left(y \circ [W_i^{(yu)}u_i + b_i^{(yu)}]\right) + W_i^{(u)}u_i + b_i\right),
\end{aligned}$$

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

Proposition 1. *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 each component of T defined by f^j is a convex function of x_j, we need to constrain all its entries of $W^{(z)}$ to be non-negative. To index the weights of d PICNN corresponding to each component of S, we use the notation $\{(W_j)_i^{(z)}\}_{i=1}^k$ for the K $W^{(z)}$ weights matrices of f^j.

Now we know that a PICNN $f^j(x_{1:j-1}, x_j; \theta^j)$ is a convex function of x_j, which implies that the derivative $\partial f^j(x_{1:j-1}, x_j; \theta^j)/\partial x_j$ is a monotonically non-decreasing function of x_j. Therefore, we can model the triangular flow as

$$T^j(x_{1:j}) := \frac{\partial f^j(x_{1:j-1}, x_j; \theta^j)}{\partial x_j}.$$

Model Learning

The learning process optimizes the parameters θ such that $KL(\theta)$ from (4) is minimized. This optimization problem can be modified with adequate regularization to ensure that the $W^{(z)}$ weight matrix entries for all layers of all d PICNN is positive. We propose using the following objective

$$\tilde{KL}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\left(\log g \circ T(\theta)(\mathbf{x}_i) + \sum_{j=1}^{d}\log(\nabla T(\theta))_j(\mathbf{x}_i)\right) + \lambda\sum_{j=1}^{d}\sum_{k=1}^{K}\|\max(-(W_j)_k^{(z)}, 0)\|_F^2, \quad (5)$$

where λ is the regularization tuning parameter. In the above equation, $\max(-(W_j)_k^{(z)}, 0)$ represents the element-wise maximum between entries of $-(W_j)_k^{(z)}$ and a zero matrix of same size. The regularized objective $\tilde{KL}(\theta)$ can be optimized using the Adam optimizer. In the next subsection, we will see a bivariate example that use input convex neural networks to approximate the KR map.

References

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

Bogachev, V. I., Kolesnikov, A. V., and Medvedev, K. V. (2005). Triangular transformations of measures. *Sbornik: Mathematics*, 196(3):309–335.

Carlier, G., Galichon, A., and Santambrogio, F. (2010). From knothe's transport to brenier's map and a continuation method for optimal transport. *SIAM Journal on Mathematical Analysis*, 41(6):2554–2576.

Huang, C.-W., Krueger, D., Lacoste, A., and Courville, A. (2018). Neural autoregressive flows. In *International Conference on Machine Learning*, pages 2078–2087. PMLR.

Jaini, P., Selby, K. A., and Yu, Y. (2019). Sum-of-squares polynomial flow. In *International Conference on Machine Learning*, pages 3009–3018. PMLR.

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.

Papamakarios, G., Nalisnick, E. T., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22(57):1–64.

Villani, C. (2003). *Topics in optimal transportation*, volume 58. American Mathematical Soc.