A short introduction on normalizing flow

Yen-Chi Chen
University of Washington
July 26, 2024

This note is a simplification of the following papers:

1. Kobyzev, I., Prince, S. J., & 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.

2. Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. Journal of Machine Learning Research, 22(57), 1-64.

1 Multivariate inverse CDF

The normalizing flow is based on the inverse CDF formula of a continuous random variable. To start with, we first discuss an interesting property of the inverse CDF. Suppose $Z \in \mathbb{R}$ is a continuous random variable with a CDF F_Z that has a well-defined inverse F_Z^{-1}. Then it is well-known that you can generate Z by sampling from a uniform random variable $U \sim \mathsf{Uni}[0,1]$ and then use the following property

$$Z \overset{d}{=} F_Z^{-1}(U).$$

In other words, $F_Z(Z) \overset{d}{=} U$.

This phenomenon also occurs in multivariate case but with a small modifications.

1.1 Multivariate case

Suppose $Z \in \mathbb{R}^d$ is a multivariate continuous random variable with a CDF F_Z and PDF p_Z. For any vector $z \in \mathbb{R}^d$, we denote

$$z_{<j} = (z_1, \cdots, z_{j-1}), \qquad z_{\leq j} = (z_1, \cdots, z_j).$$

We cannot directly apply the inverse CDF in the multivariate case because the inverse F_Z^{-1} will be a set, not a point ($d-1$ dimensional manifold under suitable conditions).

Now we consider the *partial CDF*:

$$G_j(z) = F_j(z_j|z_{<j}) = P(Z_j \leq z_j | Z_{<j} = z_{<j}). \tag{1}$$

Namely, $G_j(z)$ only depends on the variables $z_1, \cdots, z_j$ and is the conditional CDF of z_j given the 'past' $z_1, \cdots, z_{j-1}$. Clearly, the distributions F_j and G_j are well defined.

Given a vector $z \in \mathbb{R}^d$ and the functions $G_1, \cdots, G_d$, we then define

$$w_j = F_j(z_j | z_{<j}). \tag{2}$$

Since F_j is a CDF, we have the following iterative relation

$$z_j = F_j^{-1}(w_j | z_{<j}). \tag{3}$$

By iteratively applying equation (2), we are able to construct $w = (w_1, \cdots, w_d)^T$.

Here is an interesting question:

> Suppose we apply equation (2) to the random vector Z and let the resulting vector be W, what is the distribution of W?

Using equations (1) and (2), $W = (G_1(Z), \cdots, G_d(Z))^T$ so we can apply the Jacobian method to investigate the PDF of W. Note that each function G_j has the following interesting differential:

$$\frac{\partial G_j(z)}{\partial z_j} = \frac{\partial P(Z_j \leq z_j | Z_{<j} = z_{<j})}{\partial z_j} = p(z_j | z_{<j}), \qquad \frac{\partial G_j(z)}{\partial z_i} = 0 \text{ for } i > j.$$

As a result, the Jacobian matrix

$$J_G(z) = \left\{ \frac{\partial G_j(z)}{\partial z_j} \right\}$$

is a lower-triangular matrix with diagonal being $p(z_j | z_{<j})$. Thus, the determinant

$$\mathsf{det}[J_G(z)] = \prod_{j=1}^{d} p(z_j | z_{<j}) = p_Z(z).$$

Because $W = (G_1(Z), \cdots, G_d(Z))^T$, the change of variable formula shows that the PDF of w is

$$p_W(w) = p_Z(G^{-1}(w))(\mathsf{det}[J_G(G^{-1}(w))])^{-1} = 1$$

for $w \in [0,1]^d$. Therefore,

$$W \overset{d}{=} \mathsf{Uni}[0,1]^d$$

is a uniform random variable in the unit cube.

1.2 The flow

With the above result, given any continuous random variable $Z \in \mathbb{R}^d$, we can create a mapping $G : R^d \to \mathbb{R}^d$ such that $G(Z)$ follows a uniform distribution. Moreover, we can apply this procedure inversely that maps a uniform distribution over $[0,1]^d$ to any continuous random variable by iteratively applying equation (3):

$$Z_1 = F_1^{-1}(U_1) = Q_1(U), Z_2 = F_2^{-1}(U_2 | Z_1) = Q_2(U), \cdots, Z_d = F_d^{-1}(U_d | Z_{<d}) = Q_d(U) \tag{4}$$

and $U_1, \cdots, U_d$ are IID $\mathsf{Uni}[0,1]$. Namely, any random vector can be generated by

$$Z = Q(U), \tag{5}$$

where $U \sim \mathsf{Uni}[0,1]^d$ and $Q = (Q_1, \cdots, Q_d)^T$.

Moreover, we can morph one continuous random variable to another. Suppose Y has a CDF F_Y and we define G_Y to be the function created by the conditional CDF of Y in equation (1). Let Q_Z be the function Q in equation (5) constructed from a CDF F_Z. Then the function

$$\phi_{Y \to Z} = Q_Z \circ G_Y$$

will morph the random variable Y into the random variable Z, i.e.,

$$Z \overset{d}{=} \phi_{Y \to Z}(Y) = Q_Z(G_Y(Y)).$$

2 Normalizing flow: one-layer

The normalizing flow is based on the idea in the previous section–it starts with a well-known distribution such as uniform or multivariate Gaussian and then follow a sequence of *parametric* transformations in the above form to eventually match the data's distribution. For simplicity, we start with a uniform random variable $U \in \mathbb{R}^d$ and construction a sequence of mappings

$$\phi_1(u;\theta_1), \cdots, \phi_K(u;\theta_K)$$

such that $\phi_i(\cdot;\theta_i) : \mathbb{R}^d \to \mathbb{R}^d$ and

$$X = \phi_1(\cdot\theta_1) \circ \phi_2(\cdot;\theta_2) \circ \cdots \phi_K(\cdot;\theta_K)(U) = \phi_1(\phi_2(\cdots \phi_K(U;\theta_K);\theta_2)\theta_1) \tag{6}$$

matches the distribution of the observed data. Note that $\theta_1, \cdots, \theta_K$ are the parameters for each transformation.

In theory, we only need one transformation that is the inverse CDF. However, it is infeasible to estimate the inverse CDF when the dimension d is large, so the normalizing flow attempts to approximate the inverse CDF via a sequence of parametric transformations.

Examples of flows. Here are some common flows:

1. *Linear flow.* $\phi(y;\theta) = Ay + b$, where $A \in \mathbb{R}^{d \times d}, b \in \mathbb{R}^d$ are the parameters.
2. *Planar flow.* $\phi(y;\theta) = y + \eta \cdot h(\omega^T y + b)$, where $\eta, \omega \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are the parameters and h is a given function.

We will discuss more examples in Section 4.

For simplicity, we first consider the scenario that our flow only has one layer, i.e., $K = 1$.

2.1 Forward modeling

Let $X_1, \cdots, X_n$ be the observed data. Under equation (6), we model the distribution of X_i as

$$X \sim \phi_1(U; \theta_1).$$

This is called *forward modeling* since we are moving from U toward X like moving forward.

Suppose U has a PDF p_U. The resulting PDF of X is

$$p_X(x; \theta_1) = p_U(\phi_1^{-1}(x; \theta_1)) \cdot \left| \det[J_{\phi_1^{-1}(;\theta_1)}(x)] \right| = L_1(\theta_1 | x), \tag{7}$$

where

$$\left[J_{\phi_1^{-1}(;\theta_1)}(z) \right]_{ij} = \frac{\partial \phi_{1,j}^{-1}(z; \theta_1)}{\partial z_i}, \qquad i, j = 1, \cdots, d$$

is the Jacobian matrix. Note that the mapping $\phi_{1,j}^{-1}(\cdot; \theta_1)$ is the j-th component of the mapping $\phi_1^{-1}(\cdot; \theta_1) : \mathbb{R}^d \to \mathbb{R}^d$.

Equation (7) describes the likelihood function $L(\theta_1 | x)$. With this, we then estimate θ_1 by

$$\widehat{\theta}_1 = \mathsf{argmax}_{\theta_1} \frac{1}{n} \sum_{i=1}^{n} \ell_1(\theta_1 | X_i), \qquad \ell_1(\theta_1 | X_i) = \log L_1(\theta_1 | X_i).$$

Note that when we choose U to be from uniform distribution, $p_U(u) = 1$ so the log-likelihood reduces to

$$\ell_1(\theta_1 | x) = \log \left| \det[J_{\phi_1^{-1}(;\theta_1)}(x)] \right|.$$

Forward trick. Using property of inverse Jacobian, we can rewrite

$$\det[J_{\phi_1^{-1}(;\theta_1)}(x)] = \left\{ \det[J_{\phi_1(;\theta_1)}(u)] \right\}^{-1}, \qquad u = \phi_1^{-1}(x; \theta_1). \tag{8}$$

In the construction of forward model, $\phi_1(u; \theta_1)$ generally has a simple form so we often have a closed-form for its derivative. Thus, computing the Jacobian

$$\left\{ \det[J_{\phi_1(;\theta)}(W_i)] \right\}^{-1}, \qquad W_i = \phi_1^{-1}(X_i; \theta_1)$$

can be done easily.

Forward trick: linear flow. As an example, consider a linear flow $\phi_1(u; \theta) = Au + b$, where $\theta = (A, b)$. The Jacobian $J_{\phi_1(;\theta)}(u) = A$, so

$$\left\{ \det[J_{\phi_1(;\theta_1)}(u)] \right\}^{-1} = \det(A)^{-1}.$$

Forward trick: planar flow. Suppose we use the planar flow

$$\phi_1(u; \theta) = u + \rho \cdot h(\omega^T u + b), \theta = (\rho, \omega, b)$$

and h isa given function. Then the Jacobian matrix

$$[J_{\phi_1(\cdot;\theta)}(u)]_{ij} = 1 + \rho_i \cdot h'(\omega^T u + b)\omega_j$$

or equivalently,

$$J_{\phi_1(\cdot;\theta)}(u) = \mathbf{I}_d + h'(\omega^T u + b) \cdot \rho\omega^T.$$

Thus,

$$\det[J_{\phi_1(\cdot;\theta)}(u)] = 1 + h'(\omega^T u + b) \cdot \rho^T \omega.$$

2.2 Backward modeling

In equation (7), we see that the key to evaluate the likelihood function is the Jacobian, which involves the inverse of the model ϕ_1^{-1}. To simplify this, one may consider putting a parametric model directly on the inverse, i.e.,

$$\phi_1^{-1}(\cdot;\theta) = \Psi_1(\cdot;\lambda_1).$$

In this case, we finding the model

$$\Psi_1(X;\lambda_1) \overset{d}{=} U$$

that maps the data's distribution back to the generating random variable U. Since this goes backward from observed data to the generating distribution, it is called the *backward modeling*.

Under the backward modeling, the Jacobian

$$J_{\phi_1^{-1}(;\theta_1)}(x) = J_{\Psi_1(;\lambda_1)}(x)$$

is generally easy to compute.

The only downside of this idea is that the backward model $\Psi : \mathbb{R}^d \to [0,1]^d$, so the choice of models is restricted.

An interesting fact is that if the PDF of X follows from a parametric model $p(x;\rho)$ and and we construct our backward modeling via the corresponding conditional PDF

$$p_j(x_j|x_{<j};\rho), \qquad F_j(x_j|x_{<j};\rho) = \int_{-\infty}^{x_j} p_j(t|x_{<j};\rho)dt, \qquad \Psi_{1,j}(x;\rho) = F_j(x_j|x_{<j};\rho).$$

Then the backward model Ψ_1 leads to a Jacobian

$$J_{\Psi_1(;\lambda_1)}(x) = \prod_{j=1}^{d} p_j(x_j|x_{<j};\rho) = p(x;\rho),$$

which is the usual likelihood model. Thus, the usual MLE approach can be viewed as a backward modeling normalizing flow.

3 Normalizing flow: K-layers

When we have K-layers, the normalizing flow in equation (6) can be expressed as an iterative sampling procedure:

1. $Y_0 = U \sim p_U$.
2. For $k = 1, \cdots, K$, compute $Y_k = \phi_k(Y_{k-1}; \theta_k)$
3. $X = Y_K$.

The above sampling procedure also allows us to generate X if we have learned/estimated the parameters $\phi_1, \cdots, \phi_K$.

In this case, we have a set of parameters $\theta = (\theta_1, \cdots, \theta_K)$. To estimate the parameters, we use the likelihood approach. However, the usual likelihood function in equation (7) will be very complicated and not easy to work with, so people often use the forward the forward trick in equation (8) so that we obtain the following elegant form

$$\begin{aligned} \ell(\theta|X) &= \log p_U(Y_0) - \sum_{k=1}^{K} \log \left| \det \left[J_{\phi_k(\cdot;\theta_k)}(Y_{k-1}) \right] \right| \\ &= \log \Omega_0(\theta|X) - \sum_{k=1}^{K} \log \Omega_k(\theta_{k:K}|X), \\ Y_k &= \phi_k^{-1}(\cdot;\theta_k) \circ \phi_{k+1}^{-1}(\cdot;\theta_{k+1}) \circ \phi_K^{-1}(X;\theta_K). \end{aligned} \tag{9}$$

The first term Ω_0 can be eliminated if we pick p_U to be the uniform distribution so it is generally not a big problem. Each $\Omega_k(\theta_{k:K}|X)$ is the k-th Jacobian evaluated at Y_{k-1} so it only depends on the parameters 'forward' $\theta_{k:K} = (\theta_k, \theta_{k+1}, \cdots, \theta_K)$.

Maximizing $\theta(\theta|X)$ can be done by numerical methods. Note that in our forward modeling the Jacobian

$$J_{\phi_k(\cdot;\theta_k)}(y_{k-1}) = \frac{\partial \phi_k(y_{k-1};\theta_k)}{\partial y_{k-1}}$$

often has a closed-form (see, e.g., linear and planar flows in Section 2.1). So the computational cost is mostly in the evaluation of Y_{k-1} since it involves all the forward models $\phi_k, \cdots, \phi_K$.

4 Autoregressive flows

The design of flow $\phi(\cdot;\theta) : \mathbb{R}^d \to \mathbb{R}$ plays a key role in Normalizing Flows. While we want to choose a family of flows that is rich, we also have to be cautious about the computational costs on its derivative. In particular, the determinant of Jacobian

$$\det[J_{\phi(;\theta)}(u)], \qquad [J_{\phi(;\theta)}(u)]_{ij} = \frac{\partial \phi_j(u;\theta)}{\partial u_i}$$

is crucial in the inference.

The *autoregressive flows* refer to the family where $\phi_\ell(u;\theta)$, the ℓ-th component of the function $\phi(u;\theta)$, only depends on $u_1,\cdots,u_\ell$. Namely,

$$\phi_\ell(u;\theta) = \kappa_\ell(u_{\le \ell};\theta) = \kappa_\ell(u_1,\cdots,u_\ell;\theta) \tag{10}$$

for some function κ_ℓ.

The power of autoregressive flow is the fact that for any $i > j$,

$$\frac{\partial \phi_j(u;\theta)}{\partial u_i} = 0.$$

Thus, the Jacobian $J_{\phi(;\theta)}(u)$ is upper-triangular, making the determinant a product form:

$$\mathsf{det}[J_{\phi(;\theta)}(u)] = \prod_{j=1}^{d} \frac{\partial \phi_j(u;\theta)}{\partial u_j}.$$

The log-likelihood function involves the logarithm of the determinant, which further leads to an elegant expression:

$$\log \mathsf{det}[J_{\phi(;\theta)}(u)] = \sum_{j=1}^{d} \log\left(\frac{\partial \phi_j(u;\theta)}{\partial u_j}\right).$$

4.1 Transformer flow

The transformer flow is a special class of autoregressive flow. In the autoregressive flow, we have

$$\phi_\ell(u;\theta) = \kappa_\ell(u_1,\cdots,u_\ell;\theta).$$

The transformer flow further require κ_ℓ to be

$$\kappa_\ell(u_1,\cdots,u_\ell;\theta) = \zeta(u_\ell;\xi_\ell), \qquad \xi_\ell = \xi_\ell(u_1,\cdots,u_{\ell-1};\theta),$$

where ζ is a given function called transformer and ξ_ℓ is also a given function called the conditioner. Note that in the transformer function ζ, the output from the conditioner ξ_ℓ behaves like the parameter of ζ.

A feature of the transformer is that the function ζ is a univariate function and we may construct it by an integral

$$\zeta(t;\xi) = \int_{-\infty}^{t} g(s;\xi)ds,$$

where g is a non-negtive function. In this choice,

$$\frac{\partial \phi_j(u;\theta)}{\partial u_j} = \frac{\partial \zeta(u_j;\xi_j)}{\partial u_j} = \frac{\partial \int_{-\infty}^{u_j} g(s;\xi_j)ds,}{\partial u_j} = g(u_j;\xi_j) = g(u_j;\xi_j(u_1,\cdots,u_{j-1};\theta)).$$

Thus, the log-determinant is

$$\log \mathsf{det}[J_{\phi(;\theta)}(u)] = \sum_{j=1}^{d} \log\left(\frac{\partial \phi_j(u;\theta_j)}{\partial u_j}\right) = \sum_{j=1}^{d} g(u_j;\xi_j(u_1,\cdots,u_{j-1};\theta_j)),$$

a very elegant form.