HMC with Normalizing Flows

Sam Foreman[1]

[1]Argonne National Laboratory

Normalizing Flows

For a random variable z with a given distribution $z \sim r(z)$, and an invertible function $x = f(z)$ with $z = f^{-1}(x)$, we can use the change of variables formula to write

$$p(x) = r(z)\left|\det\frac{\partial z}{\partial x}\right| = r(f^{-1}(x))\left|\det\frac{\partial f^{-1}}{\partial x}\right| \tag{1}$$

Where $r(z)$ is the (simple) prior density, and our goal is to generate independent samples from the (difficult) target distribution $p(x)$. This can be done using *normalizing flows* to construct a model density $q(x)$ that approximates the target distribution, i.e. $q(\cdot) \approx p(\cdot)$ for a suitably-chosen flow f.

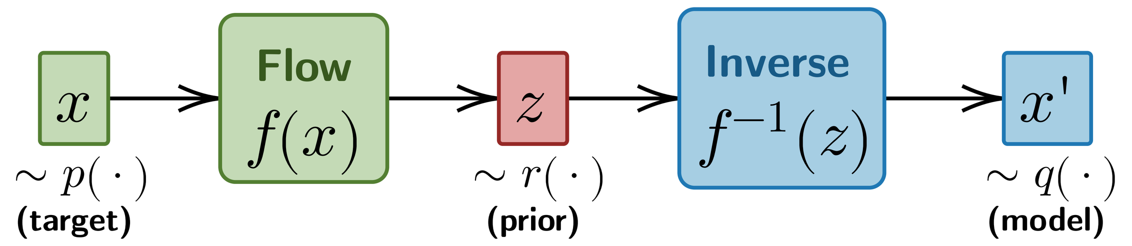

Figure 1. Using a Flow to generate data x'. Image adapted from [5]

We can construct a normalizing flow by composing multiple invertible functions f_i so that $x \equiv [f_1 \circ f_2 \circ \cdots \circ f_K](z)$. In practice, the functions f_i are usually implemented as *coupling layers*, which update an "active" subset of the variables, conditioned on the complimentary "frozen" variables.

Affine Coupling Layers

A particularly useful template function for constructing our normalizing flow is the affine coupling layer which is defined as

$$f(x_1, x_2) = \left(e^{s(x_2)}x_1 + t(x_2),\, x_2\right), \quad \text{with} \quad \log J(x) = \sum_k [s(x_2)]_k \tag{2}$$

$$f^{-1}(x_1', x_2') = \left((x_1' - t(x_2'))e^{-s(x_2')},\, x_2'\right) \quad \text{with} \quad \log J(x') = \sum_k -[s(x_2')]_k \tag{3}$$

where $s(x_2)$ and $t(x_2)$ are of the same dimensionality as x_1 and the functions act elementwise on the inputs.

In order to effectively draw samples from the correct target distribution $p(\cdot)$, our goal is to minimize the error introduced by approximating $q(\cdot) \approx p(\cdot)$. To do so, we use the (reverse) Kullback-Leibler (KL) divergence from Eq. 4, which is minimized when $p = q$.

$$D_{\mathrm{KL}}(q||p) \equiv \int dy\, q(y)[\log q(y) - \log p(y)] \approx \frac{1}{N}\sum_{i=1}^{N}[\log q(y_i) - \log p(y_i)] \quad \text{where} \quad y_i \sim q. \tag{4}$$

Hamiltonian Monte Carlo (HMC)

1. Introduce $v \sim \mathcal{N}(0, \mathbb{I}_n) \in \mathbb{R}^n$ and write the joint distribution:
$$p(x, v) = p(x)p(v) \propto e^{-S(x)}e^{-\frac{1}{2}v^T v} = e^{-H(x,v)}. \tag{5}$$
2. Evolve the joint system $\xi \equiv (\dot{x}, \dot{v})$ using Hamilton's equations along $H = \text{const}$.
3. Accept or reject the proposal configuration using the Metropolis-Hastings test.

Leapfrog Integrator:

1. $\tilde{v} \leftarrow v - \frac{\varepsilon}{2}\partial_x S(x)$
2. $x' \leftarrow x + \varepsilon\tilde{v}$
3. $v' \leftarrow \tilde{v} - \frac{\varepsilon}{2}\partial_x S(x')$

Metropolis-Hastings:

$$x_{i+1} = \begin{cases} x' & \text{w/ prob.} \quad A(\xi'|\xi) \equiv \min\left\{1, \frac{p(\xi')}{p(\xi)}\left|\frac{\partial\xi'}{\partial\xi^T}\right|\right\} \\ x & \text{w/ prob.} \quad 1 - A(\xi'|\xi). \end{cases}$$

Trivializing Map

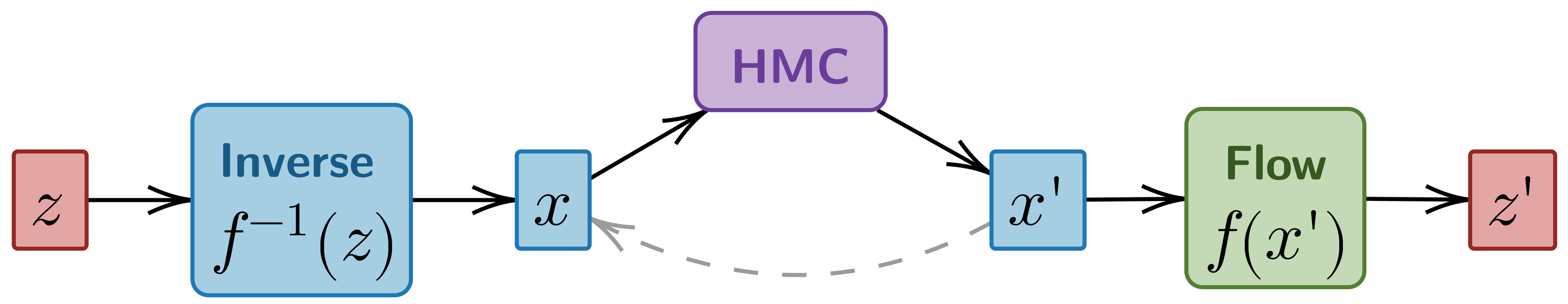

Figure 2. Normalizing Flow with inner HMC block.

Our goal is to evaluate expectation values of the form

$$\langle\mathcal{O}\rangle = \frac{1}{\mathcal{Z}}\int dx\, \mathcal{O}(x)e^{-S(x)}. \tag{6}$$

. Using a normalizing flow, we can perform a change of variables $x = f(z)$ so Eq. 6 becomes

$$\langle\mathcal{O}\rangle = \frac{1}{\mathcal{Z}}\int dz\, |\det[J(z)]|\, \mathcal{O}(f(z))\, e^{-S(f(z))}, \quad \text{where} \quad J(z) = \frac{\partial f(z)}{\partial z} \tag{7}$$

$$= \frac{1}{\mathcal{Z}}\int dz \mathcal{O}(f(z))e^{-S(f(z))+\log|\det[J(z)]|}. \tag{8}$$

The Jacobian matrix $J(z)$ must satisfy

1. Injective (1-to-1) between domains of integration
2. Continuously differentiable (*or*, differentiable with continuous inverse).

The function f is a *trivializing map* when $S(f(z)) - \log|\det J(z)| = \text{const.}$, and our expectation value simplifies to

$$\langle\mathcal{O}\rangle = \frac{1}{\mathcal{Z}}\int dz\, \mathcal{O}(f(z)). \tag{9}$$

HMC with Normalizing Flows

We can implement the trivializing map defined above using a normalizing flow model. For conjugate momenta π, we can write the Hamiltonian

$$H(z, \pi) = \frac{1}{2}\pi^2 + S(f(z)) - \log|\det J(f(z))| \tag{10}$$

and the associated equations of motion as

$$\dot{z} = \frac{\partial H}{\partial z} = \pi, \quad \text{and} \quad \dot{\pi} = -\frac{\partial H}{\partial z} = -J(z)S'(f(z)) - \log|\det J(z)|. \tag{11}$$

If we introduce a change of variables $\pi = J(z)\rho = J(f^{-1}(x))\rho$ and $z = f^{-1}(x)$, the determinant of the Jacobian matrix reduces to 1 and we obtain the modified Hamiltonian

$$\tilde{H}(x, \rho) = \frac{1}{2}\rho^\dagger\rho + S(x) - \log|\det J|. \tag{12}$$

As shown in Fig. 2, we can use $f^{-1}: z \to x$ to perform HMC updates on the transformed variables x, and $f: x \to z$ to recover the physical target distribution.

2D U(1) Gauge Theory

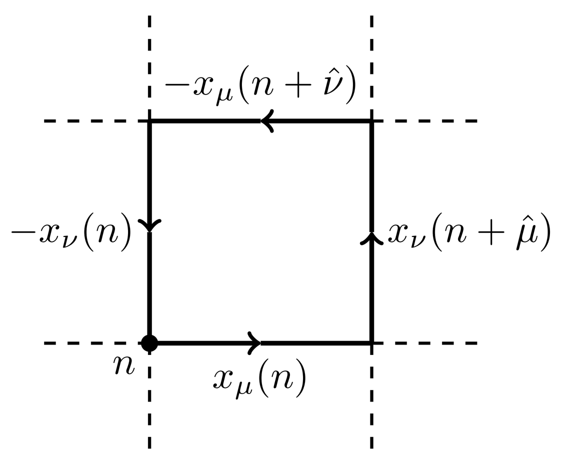

Figure 3. Plaquette

Let $U_\mu(n) = e^{ix_\mu(n)} \in U(1)$, with $x_\mu(n) \in [-\pi, \pi]$ denote the *link variables*, where $x_\mu(n)$ is a link at the site n oriented in the direction $\hat{\mu}$.

We can write our target distribution, $p(x)$, in terms of the Wilson action $S(x)$ as

$$p(x) \propto e^{-S(x)}, \quad \text{where} \quad S(x) \equiv \sum_P 1 - \cos x_P \quad \text{and} \tag{13}$$

$$x_P = x_\mu(n) + x_\nu(n+\hat{\mu}) - x_\mu(n+\hat{\nu}) - x_\nu(n) \tag{14}$$

as shown in Figure. 3. For a given lattice configuration, we can define the *topological charge* $Q \in \mathbb{Z}$ as

$$Q = \frac{1}{2\pi}\sum_P \arg(x_P), \quad \text{where} \quad \arg(x_P) \in [-\pi, \pi] \tag{15}$$

References

[1] Michael S. Albergo, Denis Boyda, Daniel C. Hackett, Gurtej Kanwar, Kyle Cranmer, Sébastien Racanière, Danilo Jimenez Rezende, and Phiala E. Shanahan. Introduction to normalizing flows for lattice field theory, 2021.

[2] Luigi Del Debbio, Joe Marsh Rossney, and Michael Wilson. Efficient modelling of trivializing maps for lattice ϕ^4 theory using normalizing flows: A first look at scalability, 2021.

[3] Martin Lüscher. Trivializing maps, the wilson flow and the hmc algorithm. *Communications in Mathematical Physics*, 293(3):899–919, Nov 2009.

[4] Martin Lüscher. Trivializing maps, the wilson flow and the hmc algorithm. *Communications in Mathematical Physics*, 293(3):899–919, Nov 2009.

[5] Lilian Weng. Flow-based deep generative models. *lilianweng.github.io/lil-log*, 2018.

foremans@anl.gov