Generative Flow

Instructor: John Thickstun

Discussion Board: Available on Ed

Zoom Link: Available on Canvas

Instructor Contact: thickstn@cs.washington.edu

Course Webpage: https://courses.cs.washington.edu/courses/cse599i/20au/

Maximum Likelihood Estimation

- Generative latent variable model:

$$1. \quad z \sim r,$$
$$2. \quad x = g_\theta(z) \sim p_\theta(x).$$

- The maximum likelihood estimator:

$$\hat{\theta}_{\text{mle}} \equiv \arg\max_\theta \mathop{\mathbb{E}}_{x \sim p} \log p_\theta(x) \approx \arg\max_\theta \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i), \text{ where } x_i \sim p.$$

- The log-likelihood of a sample:

$$\log p_\theta(x) = \log r(g_\theta^{-1}(x)) + \text{logdet} \left(\nabla_x g_\theta^{-1}(x) \right).$$

Normalizing Flows

- Build a series of transformations of our initial proposal $\mathbf{z}_0 \sim q_\phi(\cdot|x)$.
- Let $\mathcal{Z} = \mathcal{X}$, $g_s : \mathcal{Z} \to \mathcal{Z}$, and define $\mathbf{z}_t = g_t \circ \cdots \circ g_1(\mathbf{z}_0)$.
- The log-density of the pushforward distribution on $\mathbf{z}_t$ is given by

$$\log q_t(\mathbf{z}_t) = \log q_0(\mathbf{z}_0) - \sum_{s=1}^{t} \operatorname{logdet} \left(\frac{\partial g_s(\mathbf{z}_{s-1})}{\partial \mathbf{z}_{s-1}} \right).$$

- Choose functions g_s so that $\operatorname{logdet} \left(\frac{\partial g_s(\mathbf{z}_{s-1})}{\partial \mathbf{z}_{s-1}} \right)$ is easy to calculate.

Inverse Autoregressive Flow

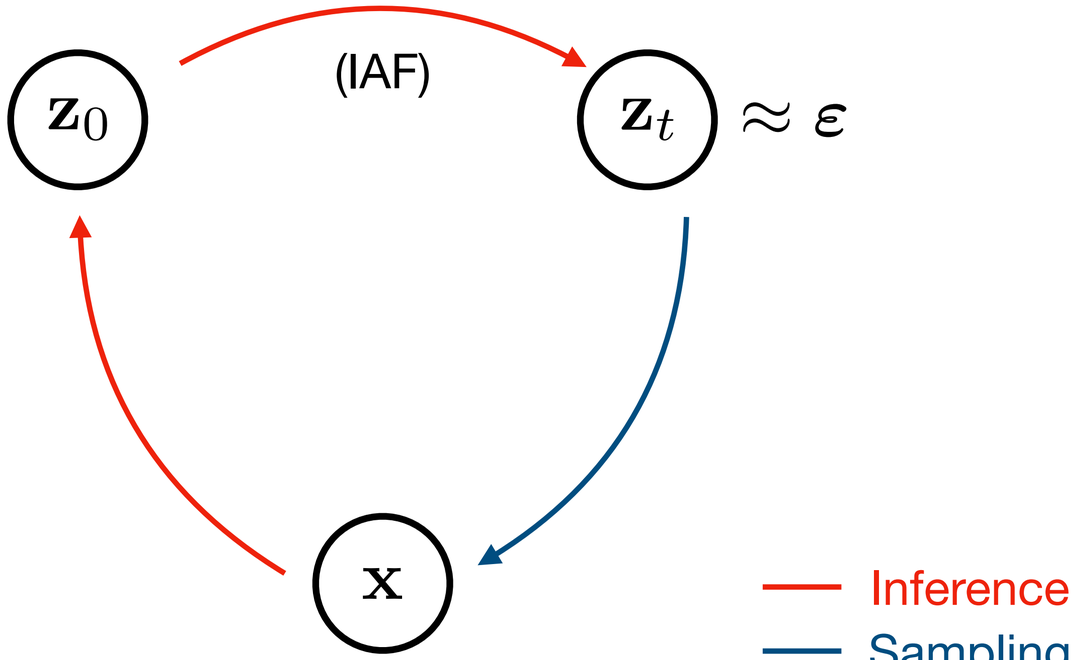
- Inverse autoregressive transformation: $\mathbf{z}_t = \dfrac{\mathbf{z}_{t-1} - \boldsymbol{\mu}_t(\mathbf{z}_{t-1})}{\boldsymbol{\sigma}_t(\mathbf{z}_{t-1})}$.

- Claim: $\operatorname{logdet}\left(\dfrac{\partial \mathbf{z}_t}{\partial \mathbf{z}_{t-1}}\right) = -\sum_{k=1}^{p} \log \boldsymbol{\sigma}_{t,k}(\mathbf{z}_{t-1,<k})$. Proof:

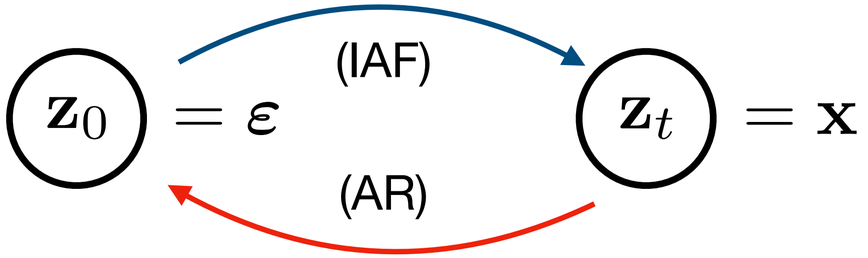
$$\frac{\partial \mathbf{z_t}}{\partial \mathbf{z_{t-1}}} = \begin{bmatrix} \frac{\partial z_{t,0}}{\partial z_{t-1,0}} & 0 & \cdots & 0 \\ \frac{\partial z_{t,0}}{\partial z_{t-1,1}} & \frac{\partial z_{t,1}}{\partial z_{t-1,1}} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \frac{\partial z_{t,0}}{\partial z_{t-1,p}} & \cdots & \frac{\partial z_{t,p-1}}{\partial z_{t-1,p}} & \frac{\partial z_{t,p}}{\partial z_{t-1,p}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\boldsymbol{\sigma}_{t,0}} & 0 & \cdots & 0 \\ \frac{\partial z_{t,0}}{\partial z_{t-1,1}} & \frac{1}{\boldsymbol{\sigma}_{t,1}} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \frac{\partial z_{t,0}}{\partial z_{t-1,p}} & \cdots & \frac{\partial z_{t,p-1}}{\partial z_{t-1,p}} & \frac{1}{\boldsymbol{\sigma}_{t,p}} \end{bmatrix}.$$

IAF for Pushforward Inference?

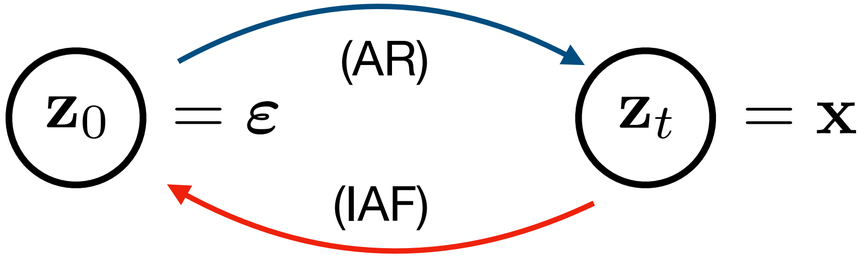
IAF for VAE



IAF for Pushforward



IAF for Inverse of Pushforward



Additive Coupling Flow

- Partition $\mathcal{Z} = \mathcal{X} = \mathbb{R}^d$ into two feature sets: $x = (x_{1,\ldots,d/2}, x_{d/2+1,\ldots,d})$.
- Parameterize $h_\theta : \mathbb{R}^{d/2} \to \mathbb{R}^{d/2}$.
- Define an **additive coupling**

$$x_{1,\ldots,d/2} = z_{1,\ldots,d/2},$$
$$x_{d/2+1,\ldots,d} = z_{d/2+1,\ldots,d} + h_\theta(z_{1,\ldots,d/2}).$$

- Claim: additive coupling is invertible for *any* function $h_\theta : \mathbb{R}^{d/2} \to \mathbb{R}^{d/2}$.
- Stack multiple flows to construct $x = g_\theta(z) = g_\theta^{(L)} \circ \cdots \circ g_\theta^{(1)}(z)$.

Inverses for Additive Flow

- One step of additive coupling flow: $z \mapsto x$ where

$$x_{1,\ldots,d/2} = z_{1,\ldots,d/2},$$
$$x_{d/2+1,\ldots,d} = z_{d/2+1,\ldots,d} + h_\theta(z_{1,\ldots,d/2}).$$

- Given x, recover the inverse z by computing

$$z_{1,\ldots,d/2} = x_{1,\ldots,d/2},$$
$$z_{d/2+1,\ldots,d} = x_{d/2+1,\ldots,d} - h_\theta(x_{1,\ldots,d/2}).$$

- This is the same construction as a Feistel cipher (DES, Blowfish, Twofish, etc.)

Jacobians for Additive Flow

- The inverse of an additive flow is

$$z_{1,\ldots,d/2} = x_{1,\ldots,d/2},$$
$$z_{d/2+1,\ldots,d} = x_{d/2+1,\ldots,d} - h_\theta(x_{1,\ldots,d/2}).$$

- The Jacobian of this inverse flow is

$$\frac{\partial z(x)}{\partial x} = \begin{bmatrix} \mathrm{Id}_{d/2} & 0 \\ -\frac{\partial h_\theta(x_{1,\ldots,d/2})}{\partial x_{1,\ldots,d/2}} & \mathrm{Id}_{d/2} \end{bmatrix}.$$

- Therefore $\mathrm{logdet}(\nabla_x z(x)) = 0$ (volume preserving transformation).

Affine Coupling Flow

- One step of **affine coupling** (Real Non-Volume Preserving) flow:

$$x_{1,\dots,d/2} = z_{1,\dots,d/2},$$
$$x_{d/2+1,\dots,d} = z_{d/2+1,\dots,d} \odot \exp(s_\theta(z_{1,\dots,d/2})) + t_\theta(z_{1,\dots,d/2}).$$

- Inverses are similar to additive flow. Jacobians of the inverse are

$$\frac{\partial z(x)}{\partial x} = \begin{bmatrix} \mathrm{Id}_{d/2} & 0 \\ \dots & \mathrm{diag}\left(\exp(-s_\theta(x_{1,\dots,d/2}))\right) \end{bmatrix}.$$

- The change of variables correction is $\mathrm{logdet}(\nabla_x z(x)) = -\sum_{i=1}^{d/2} s_\theta(x_{1,\dots,d/2})_i.$

Glow

- Scales up the RealNVP.
- A few additional architectural tricks (invertible 1x1 convolutions).
- Lots of parameters, expensive to train.
- Not competitive with other methods (yet).

Kingma and Dhariwal, Neurips 2018

5-Minute Break

Neural Differential Equations

- A step of additive coupling flow: $\mathbf{z} \mapsto \mathbf{x} = \mathbf{z} + h_\theta(\mathbf{z})$.

$$x_{1,\ldots,d/2} = z_{1,\ldots,d/2},$$
$$x_{d/2+1,\ldots,d} = z_{d/2+1,\ldots,d} + h_\theta(z_{1,\ldots,d/2}).$$

- Think of $h_\theta(\mathbf{z})$ as the dynamics of a differential equation: $\frac{\partial \mathbf{z}}{\partial t} = h_\theta(\mathbf{z}, t)$.

- A sequence of flows $\mathbf{x} = g_\theta(\mathbf{z}) = h_\theta^{(t)} \circ \cdots \circ h_\theta^{(1)}(\mathbf{z})$.

- Think of $g_\theta(\mathbf{z})$ as an Euler discretization of an integral

$$\mathbf{x} = \int_0^t h_\theta(\mathbf{z}_s, s)\, ds \approx \mathbf{z}_0 + \sum_{s=1}^{t} h_\theta(\mathbf{z}_{s-1}, s).$$

Inference for Continuous Flows

- One step of discrete flow: if $\mathbf{z} \sim q$ and $\mathbf{x} = h_\theta(\mathbf{z})$, then $\mathbf{x} \sim p_\theta$ where

$$\log p_\theta(\mathbf{x}) - \log q(\mathbf{z}) = \operatorname{logdet}\left(\nabla_x h_\theta^{-1}(\mathbf{x})\right).$$

- Instantaneous change in the log-likelihood for dynamics $\frac{\partial \mathbf{z}}{\partial t} = h_\theta(\mathbf{z}, t)$:

$$\frac{\partial \log q_s(\mathbf{z}(s))}{\partial s} = -\operatorname{tr}\left(\nabla_z h_\theta(\mathbf{z}_s, s)\right).$$

- Relatively cheap trace operation replaces expensive determinant.

- Invertibility replaced with more mild existence and uniqueness conditions.

Cumulative Inference

- Cumulative inference for discrete flows:

$$\log q_t(\mathbf{z}_t) = \log q_0(\mathbf{z}_0) - \sum_{s=1}^{t} \operatorname{logdet} \left(\frac{\partial g_s(\mathbf{z}_{s-1})}{\partial \mathbf{z}_{s-1}} \right).$$

- Analogous cumulative inference for continuous flows:

$$\log q_t(\mathbf{z}_t) = \log q_0(\mathbf{z}(0)) - \int_0^t \operatorname{tr} \left(\frac{\partial h_\theta(\mathbf{z}_s, s)}{\partial \mathbf{z}} \right) ds.$$

- For density estimation application: $\mathbf{x} = \mathbf{z}_t = g_\theta(\mathbf{z}_0)$, where $\mathbf{z}_0 \sim q_0$.

- For MLE, need to compute $\nabla_\theta \log p_\theta(\mathbf{z}) = \nabla_\theta \log q_t(\mathbf{z}_t)$.

Pontryagin Adjoints

- For maximum likelihood estimation, compute

$$\nabla_\theta \log p_\theta(\mathbf{z}) = \nabla_\theta \log q_t(\mathbf{z}_t) = -\nabla_\theta \int_0^t \operatorname{tr}\left(\frac{\partial h_\theta(\mathbf{z}_s, s)}{\partial \mathbf{z}}\right) ds.$$

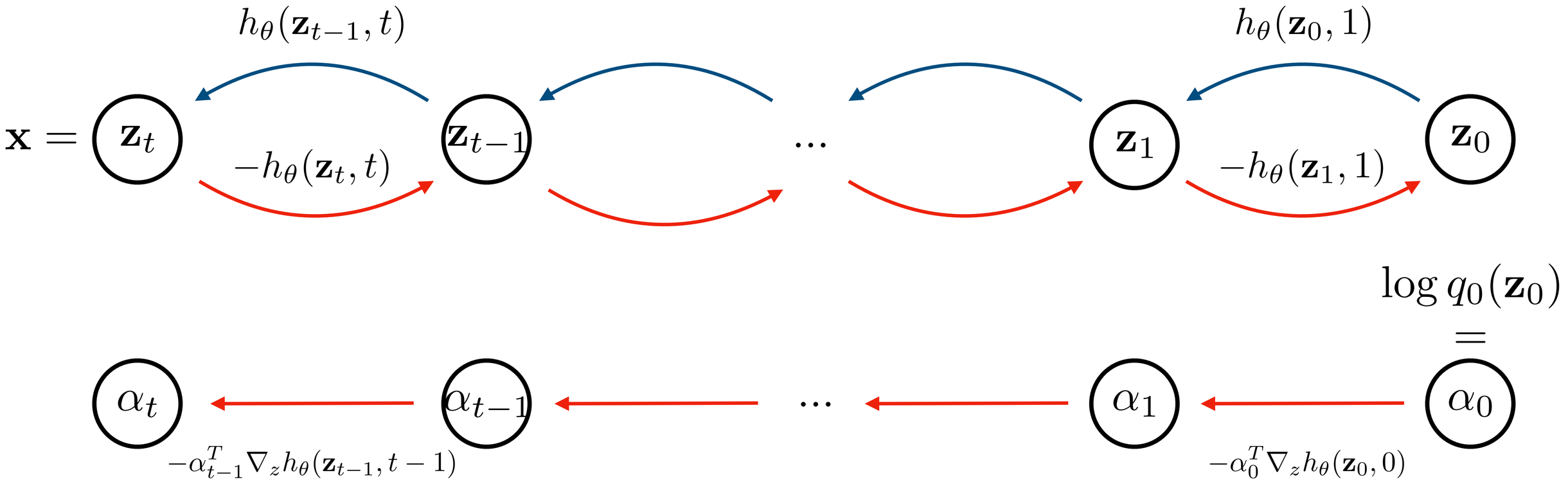
- Decompose the problem. Define an adjoint process with dynamics

$$\frac{d\alpha_s}{ds} = -\alpha_s^T \frac{\partial h_\theta(\mathbf{z}_s, s)}{\partial \mathbf{z}}, \text{ where } \alpha_0 = \frac{\partial \log q_0(\mathbf{z}_0)}{\partial \mathbf{z}_t}.$$

- The gradients of the log-likelihood are given in terms of the adjoint:

$$\nabla_\theta \log q_t(\mathbf{z}_t) = \int_0^t \alpha_s^T \frac{\partial h_\theta(\mathbf{z}_s, s)}{\partial \theta} ds.$$

Continuous Back-Prop



Inference

Sampling

$$\nabla_\theta \log q_t(\mathbf{z}_t) = \int_0^t \alpha_s^T \frac{\partial h_\theta(\mathbf{z}_s, s)}{\partial \theta} \, ds.$$

Ffjord

- Neural ODE for density estimation.
- Some additional tricks beyond what we discussed today.
- Not competitive with other methods (yet).
- More theory needed? Better parameterization?

CIFAR-10 Modeling

Grathwohl et. al., ICLR 2019