Groupe de lecture :

OT-Flow :

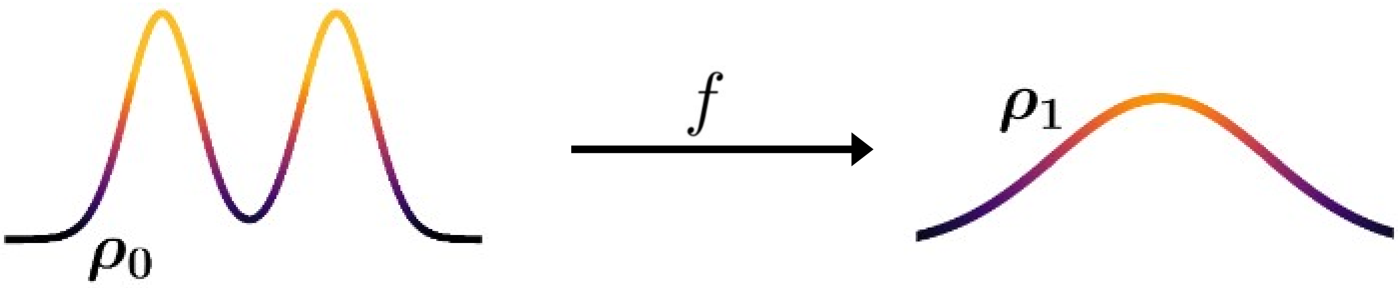
Fast and Accurate Continuous Normalizing Flows via Optimal Transport

Derek Onken, Samy Wu Fung, Xingjian Li, Lars Ruthotto 2021

Buonomo Camille,
14/05/2024,

1. Normalizing flows

Invertible mapping : $f : \mathbb{R}^d \to \mathbb{R}^d$



Condition on f : $X \sim \rho_1, f \text{ s.t. } f(X) \sim \rho_0$

$$log(\rho_0(x)) = log(\rho_1(f(x))) + log(det(\mathcal{J}_f(x)))$$

1. Normalizing flows

In practice : Composition of simple invertible map

$$f = f_k \circ f_{k-1} \circ \ldots \circ f_1 \circ f_0$$

$$f^{-1} = f_1^{-1} \circ \ldots \circ f_k^{-1}$$

$$det(\mathcal{J}_f)(f(x)) = \prod det(\mathcal{J}_{f_k})(x_k)$$

1. Normalizing flows

In practice : Composition of simple invertible map

$$f = f_k \circ f_{k-1} \circ \ldots \circ f_1 \circ f_0$$

Lots and lots of possibility for f_k

- Expressivity
- Inference
- Inverse
- Jacobian
- Determinant of Jacobian
- ...

[Kobyzev et Al. 2021]

Noteworthy exemple : $f_k(x) = x + hg(x;\theta)$

2. Continuous Normalizing flows

$$f_k(x) = x + hg(x;\theta)$$

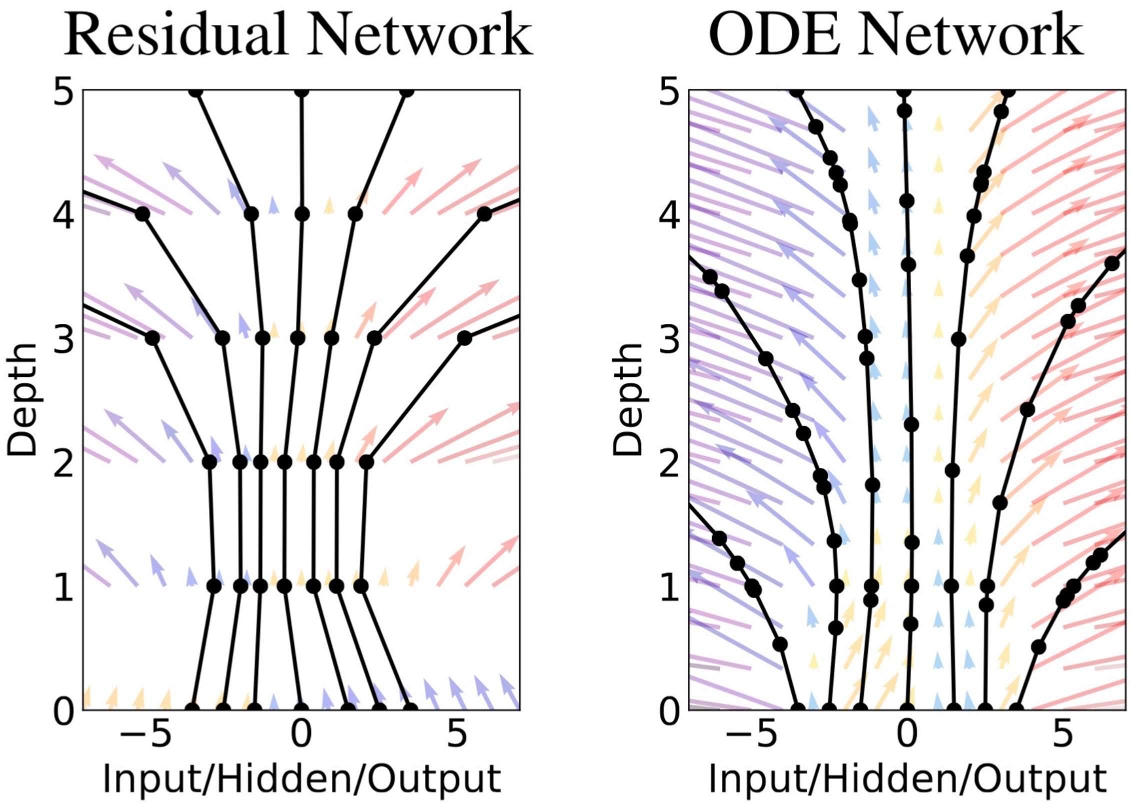
Forward Euler step : $y^k = y^{k-1} + \Delta t V(y^{k-1}, t_{k-1})$

Corresponding EDO : $\partial_t z = g(z, t; \theta)$
$z(0) = x$

→ Integration of a learned dynamic : $f(x) = z_x(T) = x + \int_0^T g(z(t), t; \theta) dt$

2. Continuous Normalizing flows

"Infinitely" deep neural network



[Chen et Al. 2019]

2. Continuous Normalizing flows

Instantaneous change of variable :

$$\partial_t l(x,t) = Tr(\mathcal{J}_{g(\cdot,t;\theta)}(z(x,t))) = div(g)(z(x,t),t;\theta)$$

Such that : $l(x,T) = log(det(\mathcal{J}_f(x)))$

Continous backpropagation :

$$\partial_\theta L = \int_0^T \nabla_{z(t)} L \, \partial_\theta g(z(t),t;\theta)dt$$

[Chen et al. 2019]

2. Continuous Normalizing flows

Training : Kullback-Leibler divergence

$$
\begin{aligned}
&\mathbb{D}_{\mathrm{KL}}\left[\,\rho(\boldsymbol{z}(\boldsymbol{x},T))\,||\,\rho_1(\boldsymbol{z}(\boldsymbol{x},T))\,\right] \\
&= \int_{\mathbb{R}^d} \log\left(\frac{\rho(\boldsymbol{z}(\boldsymbol{x},T))}{\rho_1(\boldsymbol{z}(\boldsymbol{x},T))}\right)\rho(\boldsymbol{z}(\boldsymbol{x},T))\det\left(\nabla \boldsymbol{z}(\boldsymbol{x},T)\right)\mathrm{d}\boldsymbol{x}, \\
&= \int_{\mathbb{R}^d} \log\left(\frac{\rho_0(\boldsymbol{x})}{\rho_1\left(\boldsymbol{z}(\boldsymbol{x},T)\right)\ \det\left(\nabla \boldsymbol{z}(\boldsymbol{x},T)\right)}\right)\rho_0(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \\
&= \int_{\mathbb{R}^d} \left[\log\left(\rho_0(\boldsymbol{x})\right) - \log\left(\rho_1(\boldsymbol{z}(\boldsymbol{x},T))\right) - \log\det\left(\nabla \boldsymbol{z}(\boldsymbol{x},T)\right)\right]\rho_0(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.
\end{aligned}
$$

- ρ0 is unknown
- ρ1 is chosen as a normal distribution

2. Continuous Normalizing flows

Training : log-likehood maximisation

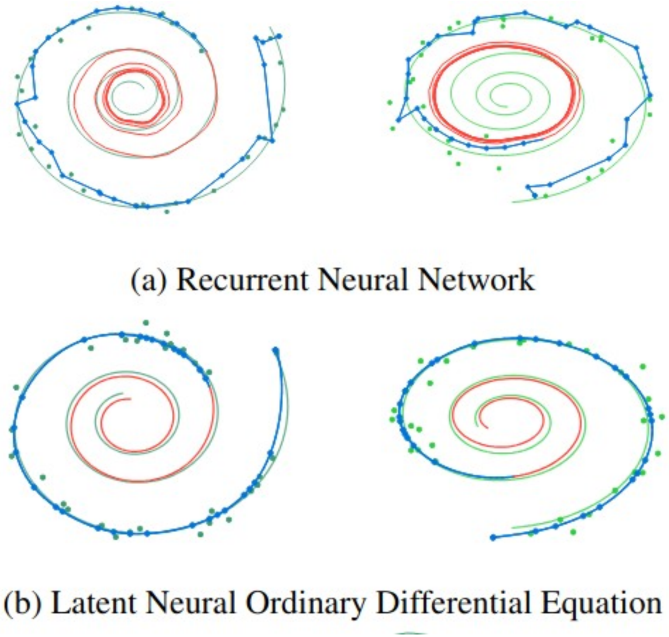
$$\begin{aligned}
\mathbb{D}_{\mathrm{KL}} &= \int_{\mathbb{R}^d} \left[\log\left(\rho_0(\boldsymbol{x})\right) - \log\det\left(\nabla \boldsymbol{z}(\boldsymbol{x},T)\right) + \frac{1}{2}\|\boldsymbol{z}(\boldsymbol{x},T)\|^2 + \frac{d}{2}\log(2\pi) \right] \rho_0(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \\
&= \int_{\mathbb{R}^d} \left[\log\left(\rho_0(\boldsymbol{x})\right) + C(\boldsymbol{x},T) \right] \rho_0(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \qquad C(x,T) = \frac{1}{2}\|z(x,T)\|^2 + l(x,T) - \frac{d}{2}log(2\pi) \\
&= \mathbb{E}_{\rho_0(\boldsymbol{x})} \left\{ \log\left(\rho_0(\boldsymbol{x})\right) + C(\boldsymbol{x},T) \right\},
\end{aligned}$$

$$\rightarrow \quad \min_{\theta} E_{\rho_0}[C(x,T)]$$

2. Continuous Normalizing flows

Drawbacks :

- Slow integration
- “High” computational cost of the score



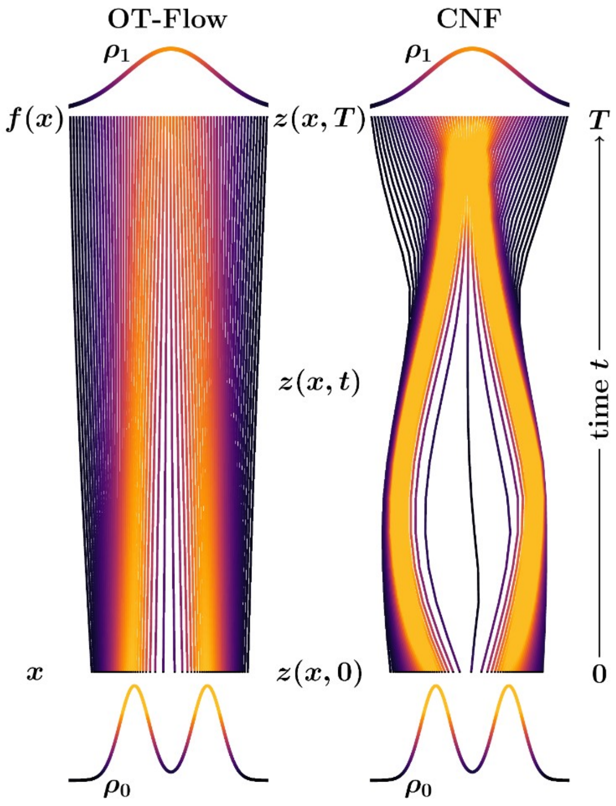
(a) Recurrent Neural Network

(b) Latent Neural Ordinary Differential Equation

[Chen et Al. 2019]

3. OT-Flow

Main idea : Regularization of the trajectories for a faster integration



3. OT-Flow

Main idea : Regularization of the trajectories for a faster integration

OT penalization : $L(x,T) = \int_0^T \frac{1}{2} \|g(z_x(t), t; \theta)\|^2 dt$

→ Relaxed Benamou-Brenier formulation

$$\min_\theta E_{\rho_0}[C(x,T) + L(x,T)]$$

- “Convex optimization” problem
- Straight and non-intersecting trajectories

3. OT-Flow

Pontryagin Maximum Principle : $\exists \Phi \text{ s.t. } g(x,t;\theta) = -\nabla\Phi(x,t;\theta)$

→ Learning of the potential instead

Hamilton-Jacobi-Bellman equation (optimal controle theory):

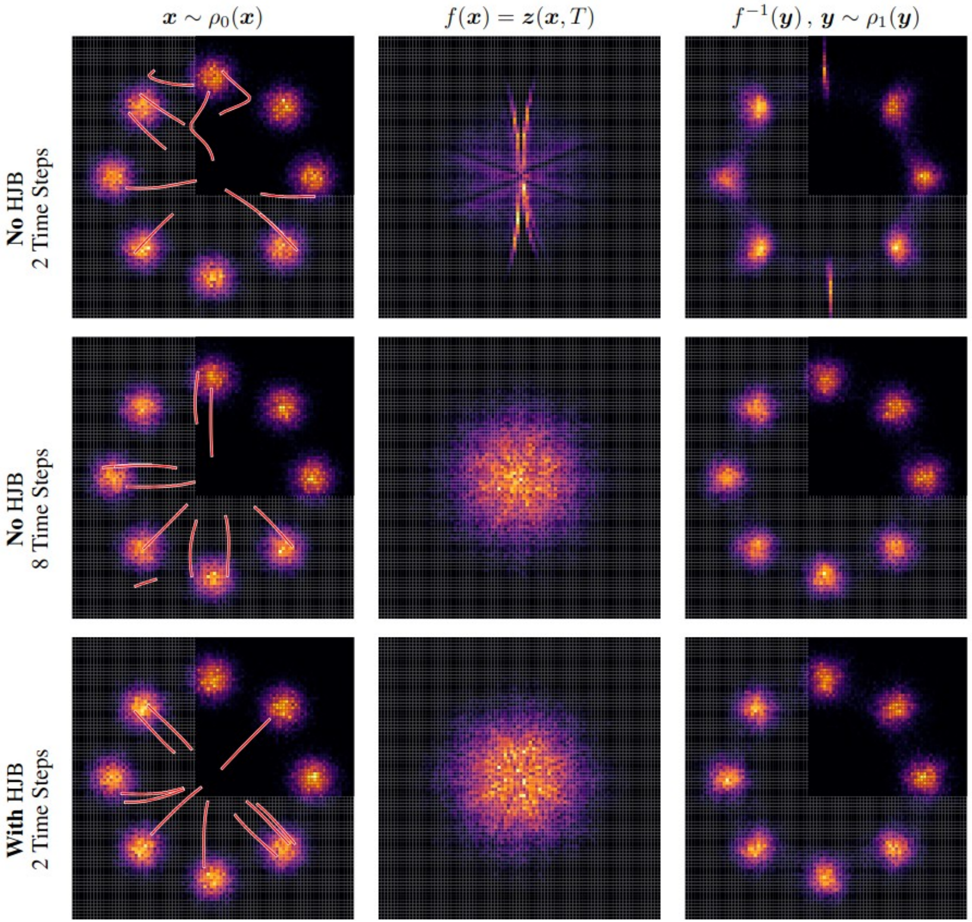
$$-\partial_t \Phi(\boldsymbol{x},t) + \frac{1}{2}\|\nabla\Phi(\boldsymbol{z}(\boldsymbol{x},t),t)\|^2 = 0$$

$$\begin{aligned}\Phi(\boldsymbol{x},T) = 1 + \log\left(\rho_0(\boldsymbol{x})\right) - \log\left(\rho_1(\boldsymbol{z}(\boldsymbol{x},T))\right) \\ - \ell(\boldsymbol{z}(\boldsymbol{x},T),T).\end{aligned}$$

Additional constraint :

$$R(\boldsymbol{x},T) = \int_0^T \left| \partial_t \Phi\big(\boldsymbol{z}(\boldsymbol{x},t),t\big) - \frac{1}{2}\|\nabla\Phi\big(\boldsymbol{z}(\boldsymbol{x},t),t\big)\|^2 \right| \, \mathrm{d}t.$$

3. OT-Flow



$\boldsymbol{x} \sim \rho_0(\boldsymbol{x})$
$f(\boldsymbol{x}) = \boldsymbol{z}(\boldsymbol{x}, T)$
$f^{-1}(\boldsymbol{y})\,,\ \boldsymbol{y} \sim \rho_1(\boldsymbol{y})$
No HJB
2 Time Steps
No HJB
8 Time Steps
With HJB
2 Time Steps

3. Implementation

Network architecture :

$$\Phi(\boldsymbol{s};\boldsymbol{\theta}) = \boldsymbol{w}^\top N(\boldsymbol{s};\boldsymbol{\theta}_N) + \frac{1}{2}\boldsymbol{s}^\top(\boldsymbol{A}^\top\boldsymbol{A})\boldsymbol{s} + \boldsymbol{b}^\top\boldsymbol{s} + c$$

ResNet :

$$\boldsymbol{u}_0 = \sigma(\boldsymbol{K}_0\boldsymbol{s} + \boldsymbol{b}_0)$$
$$N(\boldsymbol{s};\boldsymbol{\theta}_N) = \boldsymbol{u}_1 = \boldsymbol{u}_0 + h\,\sigma(\boldsymbol{K}_1\boldsymbol{u}_0 + \boldsymbol{b}_1)$$

4. Score Computation

Exact trace computation : $\operatorname{tr}\left(\nabla^2\Phi(\boldsymbol{s};\boldsymbol{\theta})\right) = \operatorname{tr}\left(\boldsymbol{E}^\top \nabla^2_{\boldsymbol{s}}(N(\boldsymbol{s};\boldsymbol{\theta}_N)\boldsymbol{w})\,\boldsymbol{E}\right) + \operatorname{tr}\left(\boldsymbol{E}^\top(\boldsymbol{A}^\top\boldsymbol{A})\,\boldsymbol{E}\right),$

$$\operatorname{tr}\left(\boldsymbol{E}^\top \nabla^2_{\boldsymbol{s}}(N(\boldsymbol{s};\boldsymbol{\theta}_N)\boldsymbol{w})\,\boldsymbol{E}\right) = t_0 + h\,t_1, \quad \text{where}$$

$$t_0 = \left(\sigma''(\boldsymbol{K}_0\boldsymbol{s}+\boldsymbol{b}_0)\odot\boldsymbol{z}_1\right)^\top((\boldsymbol{K}_0\boldsymbol{E})\odot(\boldsymbol{K}_0\boldsymbol{E}))\boldsymbol{1}$$

$$t_1 = \left(\sigma''(\boldsymbol{K}_1\boldsymbol{u}_0+\boldsymbol{b}_1)\odot\boldsymbol{w}\right)^\top((\boldsymbol{K}_1\boldsymbol{J})\odot(\boldsymbol{K}_1\boldsymbol{J}))\boldsymbol{1}$$