$$\min_{u} \int_0^T L(u(t), \dot{u}(t))\, dt \qquad L(u,v) = \frac{1}{2}|v - b(u)|^2$$

$u(0) = u_0$

$u(T) = u_1$

Standard approach: solve E.L eq.

↳ MAM, GMAM, etc.

Machine learning:

- represent solution by neural network
- view objective fct as loss for network parameters
- optimize parameters by stochastic gradient descent

Here : solution = $u(t)$ or $u(t,x)$

u fct of time u fct of space.

ex 1) $\int_0^T dt \int_\Omega | \partial_t u - \Delta u - \lambda \underbrace{|\nabla u|^2}_{\text{non gradient}} - f(u) |^2 dx\, dt$

2) $\int_0^T dt \int_0^1 dx \; | \partial_t u - \partial_x^2 u - \underbrace{a\, \partial_x u}_{\text{shear}} - f(u) |^2 dx\, dt$

shear   non gradient

...

$u(t,x) \longrightarrow u^\theta(t,x)$ = neural network

θ = parameters.

$$u^\theta(t,x) = \sigma_L \left(\cdots \sigma_2 \left(W_2 \left(\sigma_1 \left(W_1 \begin{pmatrix} t \\ x \end{pmatrix} + b_1 \right) + b_2 \right) \right) \cdots \right)$$

where: $\theta = (W_1 \; b_1 \; W_2 \; b_2 \; \ldots)$

with: W_j = weight matrices   eg $W_1 \in \mathbb{R}^{p \times (1+d)}$

b_j = bias vectors

σ_j = non linearities $\sigma_1 : \mathbb{R}^p \to \mathbb{R}^q$ eg ReLU

$$\Longrightarrow \quad \int_0^T L(u^\theta(t), \dot{u}^\theta(t))\,dt = L(\theta)$$ Loss

$$\approx \frac{1}{N}\sum_{j=1}^{N} L(u^\theta(t_j), \dot{u}^\theta(t_j)) = L^N(\theta)$$ empirical loss

t_j = iid $U(0,T)$.

also x_j = iid $U(\Omega)$ if space needed

PINN loss.

physics informed neural network.

optimization SGD.

repeat: $\theta^{n+1} = \theta^n - \eta^n \partial_\theta L(\theta^n)$ $n=0,1\dots$

learning rate.

until convergence.

! $u^\theta(t,x)$ = analytical fn of (t,x,θ)

all derivatives computed exactly

approx? # of parameters θ is finite.

More ambitious calculations with ?

SDE: $dX_t = b(X_t)\,dt + \sigma\, dW_t$

FPE: $\partial_t \rho_t = -\nabla\cdot(b\rho_t) + \frac{\sigma^2}{2}\Delta\rho_t$.

let $\rho_{-t} = \rho^R_t$ time reversed

$$\partial_t \rho^R_t = +\nabla\cdot(b\rho^R_t) - \frac{\sigma^2}{2}\Delta\rho^R_t$$

$$= \nabla\cdot\Big[\underbrace{\left(b - \sigma^2\nabla\log\rho^R_t\right)}_{= -b^R(x)}\rho^R_t\Big] + \frac{\sigma^2}{2}\Delta\rho^R_t$$

time reversed drift

time reversed SDE:

$$dX^R_t = b^R(X^R_t)\,dt + \sigma\, dW_t.$$

$$\boxed{\text{time reversal symmetry} : \quad b^R(x) = b(x)}$$

e.g. if $\rho_t = \rho$, ρ_t^R (NE) SS

$$\Rightarrow \quad b^R(x) = -b(x) + \sigma^2 \nabla \log \rho(x)$$

$$\boxed{\begin{aligned} &= b(x) \quad ? \\ \Leftrightarrow \quad & b(x) = \tfrac{1}{2}\sigma^2 \nabla \log \rho(x) \end{aligned}}$$

ex: $b(x) = -\nabla U(x)$ $\qquad \frac{\sigma^2}{2} = k_B T = \beta^{-1}$

$$\Rightarrow \quad \rho(x) = Z^{-1} e^{-\beta U(x)}$$

$$\frac{\sigma^2}{2} \nabla \log \rho(x) = -\beta^{-1} \beta \nabla U(x) = -\nabla U(x)$$

In general $b(x) \neq b^R(x)$ breakup of TRS.

Can we check TRS ?

Can we estimate $\nabla \log \rho(x)$?

useful also in Stochastic Thermodynamics

on NESS.

$$\nabla \log \rho(x) = \arg\min_{s} \int_{\Omega} | s(x) - \nabla \log \rho(x) |^2 \rho(x)\, dx$$

Fisher divergence

$$\int_{\Omega} | s(x) - \nabla \log \rho(x) |^2 \rho(x)\, dx$$

$$= \int_{\Omega} |s(x)|^2 \rho(x) dx \left(- 2 \int_{\Omega} s(x) \cdot \underbrace{\nabla \log \rho(x)\, \rho(x)}_{= \nabla \rho(x).} dx \right. + \left(\int |\nabla \log \rho(x)|^2 \rho(x) dx \right.$$

$= \text{cst} \; (\text{no } s(x))$

$$= +2 \int_{\Omega} \nabla \cdot s(x)\, \rho(x)\, dx$$

$$\Rightarrow \quad \boxed{\nabla \log p(x) = \underset{s}{\operatorname{argmin}} \left(\int_{\Omega} \left(|s(x)|^2 + 2 \nabla \cdot s(x) \right) p(x)\, dx \right.}$$

$$\approx \frac{1}{N} \sum_{j=1}^{N} \left[|s(x_j)|^2 + 2 \nabla \cdot s(x_j) \right]$$

x_j drawn from $p(x)$

$\Downarrow$

Empirical loss for $s(x)$

! here $s: \mathbb{R}^d \to \mathbb{R}^d$ high dim. fct.

use power of approx of neural networks