Optimal time sampling in PINNs (Physics-Informed Neural Networks)

Gabriel Turinici

CEREMADE Université Paris Dauphine - PSL, Paris, France

27^{th} ICPR conference, Kolkata, India, Dec. 1-5, 2024

What are PINNs (Physics-Informed Neural Networks) ?

• one of the most impactful applications of Neural Networks (cf; physics Nobel price 2024) in … physics; huge literature, original 2019 paper [1] $\geq$2k citations/y !

• **PINNs** combine deep learning techniques with physical laws described by partial differential equations (PDEs).

• They allow neural networks to learn solutions to PDEs, leveraging both data and known physical constraints.

• Unlike traditional neural networks, PINNs directly encode the physical knowledge into the loss function.

Applications of PINNs

- **Computational Fluid Dynamics**: Modeling airflow or fluid flow using Navier-Stokes equations.
- **Heat Transfer**: Solving heat equations with real-world constraints.
- **Wave Propagation**: Modeling sound, seismic waves, and other phenomena described by hyperbolic PDEs.
- **Quantum Mechanics**: Schrödinger's equation in high dimensions or for complex quantum systems.
- **...**

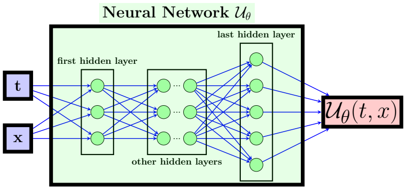
How do PINNs work? I

- consider some evolution equation:

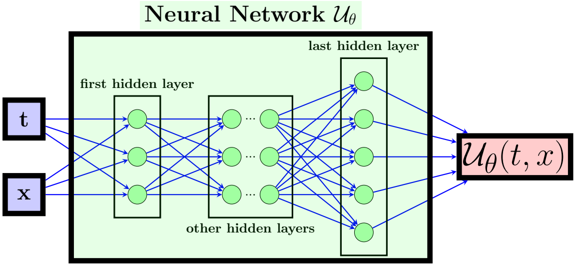
$$\partial_t u = \mathcal{G}_t u, \tag{1}$$
$$u(0,x) = u_0(x), \ \forall x \in \Omega \tag{2}$$
$$u(t,x) = u_{bc}(t,x), \ \forall t \in [0,T], x \in \partial\Omega. \tag{3}$$

$u(\cdot,\cdot)$ = solution, t=time, x=space, $\mathcal{G}_t$=operator (e.g., $\mathcal{G}_t = \partial_{xx}$ for the heat equation), $u_0(\cdot)$, $u_{bc}(\cdot,\cdot)$ = initial & boundary conditions.

- NN constructs an approximation $\mathcal{U}_\theta(t,x)$ of the solution $u(t,x)$



How do PINNs work? II



to solve $\partial_t u = \mathcal{G}_t u$ + IC + BC

- construct **automatically** the error $\boxed{w(t,x) = \partial_t \mathcal{U}_\theta(t,x) - \mathcal{G}_t \mathcal{U}_\theta(t,x)}$

- The neural network is trained to minimize loss : $\|w\|^2$ + fit to IC + fit to BC + fit any available data /measurements.

- Technical remark: constructing the loss and optimizing with gradient descent is a second/ third order auto-diff process (!) Special activations are necessary (RELU3, tanh, etc.).

Sampling time in PINN

- The NN is trained to minimize a loss functional; original [1] is $\int_0^T \int_\Omega |w(t,x)|^2 dxdt$
- in fact any weighted ($\rho > 0$) loss has the same optimum $w = 0$:

$$\mathcal{L}(\theta) = \int_0^T \int_\Omega \rho(t) |w(t,x)|^2 dxdt, \tag{4}$$

- **weight $\rho(t) \geq 0$ are main object here; $\rho(t)$ is chosen to speed up the convergence and improve the quality of the output.**
- Computational cost assumption obtaining error w has numerical costs. If max computational cost is B then :

$$\int_0^T \frac{1}{\|w(t,\cdot)\|^2} dt \leq B. \tag{5}$$

- Research question : given max cost B what is the best weighting scheme ρ that makes final error $\|w(T,\cdot)\|$ minimal ? Similar for $\int_0^T \|w(t,x)\|^2 dt$.

Sampling time in PINN: intuition

$$\text{loss}: \quad \boxed{\mathcal{L}(\theta) = \int_0^T \int_\Omega \rho(t)|w(t,x)|^2 dxdt} \tag{6}$$

$$\text{error}: \quad \boxed{w(t,x) = \partial_t \mathcal{U}_\theta(t,x) - \mathcal{G}_t \mathcal{U}_\theta(t,x)} \tag{7}$$

• **Intuition**: errors will propagate in time, any initial error will amplify; this is the "causal" view [4]
• **Causal recommendation**: solve more precisely the initial points in time.

• **Question**: is this optimal, what "more precisely" means quantitatively ?

• **more general question**: are we going to oversample past, present or future ? Which one is more important ?

Sampling time in PINN: formulation min/max, control

- loss : $\boxed{\mathcal{L}(\theta) = \int_0^T \int_\Omega \rho(t) |w(t,x)|^2 dx dt}$

- error : $\boxed{w(t,x) = \partial_t \mathcal{U}_\theta(t,x) - \mathcal{G}_t \mathcal{U}_\theta(t,x)}$

- **Intuition**: assuming NN is expressive, problem can be formalized as a random control problem or a min-max problem

$$\min_{\rho} \max_{\substack{\partial_t \mathcal{U}_\theta(t,x) = \mathcal{G}_t \mathcal{U}_\theta(t,x) + w(t,x), \\ \int_0^T 1/\|w(t,x)\|^2 dt \le B}} \|\mathcal{U}_\theta(T,\cdot) - u(T,\cdot)\| \tag{8}$$

- **Note**: other formalizations [2]: $\min_\rho \max_{\substack{\partial_t \mathcal{U}_\theta(t,x) = \mathcal{G}_t \mathcal{U}_\theta(t,x) + w(t,x), \\ \int_0^T \rho(t) \|w(t,x)\|^2 dt \le C_p}} \|\mathcal{U}_\theta(T,\cdot) - u(T,\cdot)\|$

Sampling time in PINN

Proposition ([3])

Consider $\mathcal{G}(u) = \lambda u$ and assume that (5) (computational cost hypothesis) holds true. Then under hypothesis ... the error $|\mathcal{U}_\theta(T) - u(T)|$ is optimal when $w(t)$ is proportional to $e^{-\lambda(T-t)/3}$ i.e., $d\rho(t)$ of the form $e^{-rt}dt$ for some $r = r(\lambda)$ (explicit).

• When $\lambda < 0$ future is oversampled no the past ! We follow regimes: chaotic for $Re(\lambda) > 0$, periodic $Re(\lambda) = 0$ or stable $Re(\lambda) < 0$.

• compare with [4] that uses a discrete form of $\rho(t) = e^{-\epsilon \int_0^t \|w(t,x)\|^2_{L^2_x}}$
Futher results : [2].

Numerical results I

All tests reproducible using codes on Github

`https://github.com/gabriel-turinici/pinn_exponential_sampling` version August 31st 2024.

- situation: toy example $\mathcal{G}(u) = \lambda u$, equation $u' = \lambda u$

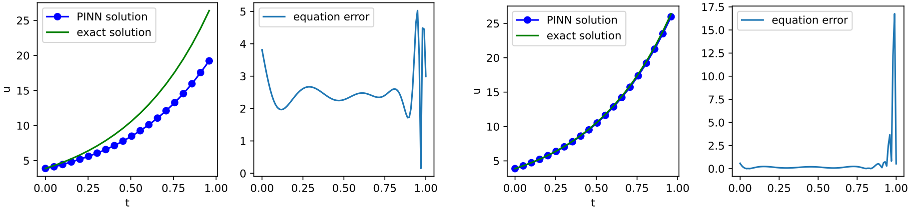


Figure: Results for $\lambda = 2.$ and sampling parameter $r = 2.0$. First two plots: the results for 500 epochs. Last two plots: results for 1500 epochs.

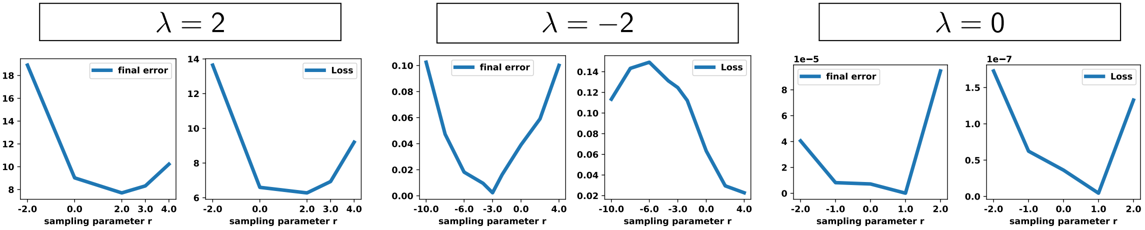


Figure: Sampling law influence for $\lambda = 2$ (from left to right, plots 1 and 2), $\lambda = -2$ (plots 3 and 4) and $\lambda = 0$ (plots 5 and 6). All sampling are done with law $\mathcal{E}^{0,T,r}$. Plots 1, 3 and 5 : the final error as a function of r. Plots 2,4 and 6 : the loss.

Numerical results III

- Burgers' equation ($u(x,t)$=velocity, $x \in [-1,1]$, $\nu = 0.01/\pi$=viscosity)

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \;\; u(0,x) = -\sin(\pi x), \;\; u(\pm 1, t) = 0 \;\forall t \in [0,1], \tag{9}$$

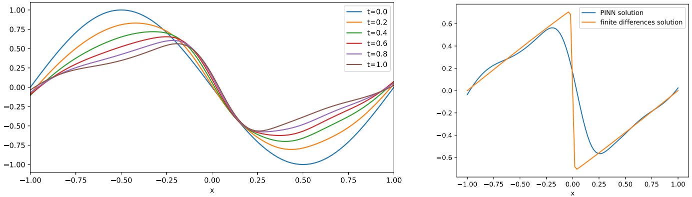


Figure: Burgers' equation sampling parameter $r = 0$ i.e., uniform law $\mathcal{E}^{0,T,0}$. Left plot: the solution at different times. Right plot: the comparison with a finite difference solution considered exact.

Numerical results IV

- Burgers' equation ($u(x,t)$=velocity, $x \in [-1,1]$, $\nu = 0.01/\pi$=viscosity)

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \; u(0,x) = -\sin(\pi x), \; u(\pm 1, t) = 0 \; \forall t \in [0,1], \tag{10}$$

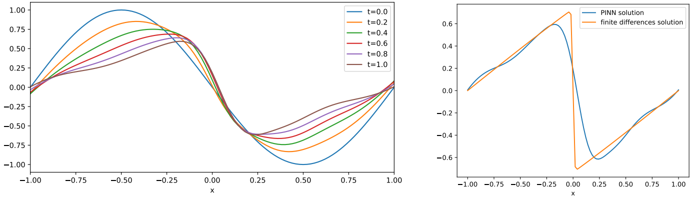


Figure: Burgers' equation sampling parameter $r = 1$ and law $\mathcal{E}^{0,T,r}$. Left plot: the solution at different times. Right plot: the comparison with a finite difference solution considered exact.

Numerical results V

- situation: Lorenz system (chaotic) $x'(t) = \sigma(y - x)$, $y'(t) = x(\rho - z) - y$, $z'(t) = xy - \beta z$. Here $x, y, z=$ state variables, $\sigma = 10$, $\rho = 28$, $\beta = 8/3$=parameters , initial state $(1, 1, 1)$.

$r = 0$	$r = -10$

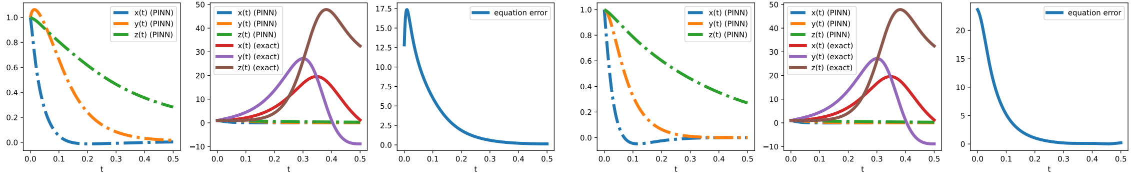


Figure: Lorenz system with weight parameters $r = 0$ and $r = -10$. The solution quality is not good.

Numerical results VI

• situation: Lorenz system (chaotic) $x'(t) = \sigma(y - x)$, $y'(t) = x(\rho - z) - y$, $z'(t) = xy - \beta z$. Here $x, y, z=$ state variables, $\sigma = 10$, $\rho = 28$, $\beta = 8/3$=parameters , initial state $(1, 1, 1)$.

$r = 10$

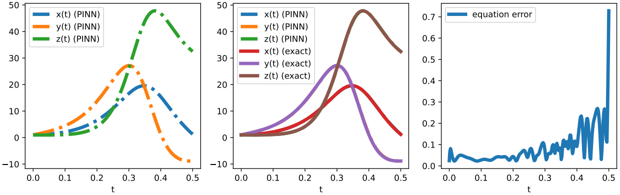


Figure: Lorenz system with weight parameter $r = 10$; the numerical and exact solutions are superposed and indistinguishable graphically.

Conclusions

- adapted sampling in time is important in many time evolution PINNs
- the optimal sampling is of exponential type
- numerical results are positive
- the optimal parameter is not explicit but alternatives exist cf. [2]. Further work required.

References I

M. Raissi, P. Perdikaris, and G.E. Karniadakis.
Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations.
Journal of Computational Physics, 378:686–707, 2019.

Gabriel Turinici.
Lyapunov weights to convey the meaning of time in physics-informed neural networks, 2024.
arxiv:2407.21642.

Gabriel Turinici.
Optimal time sampling in physics-informed neural networks, 2024.
arxiv:2404.18780, ICPR 2024 accepted paper.

Sifan Wang, Shyam Sankaran, and Paris Perdikaris.
Respecting causality for training physics-informed neural networks.
Computer Methods in Applied Mechanics and Engineering, 421:116813, 2024.
arXiv preprint arXiv:2203.07404.