Tomáš Oberhuber

Faculty of Nuclear Sciences and Physical Engineering
Czech Technical University in Prague

Physics-informed neural networks

M. Raissi, P. Perdikaris, 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, vol. 378, pp. 686-707, 2019.

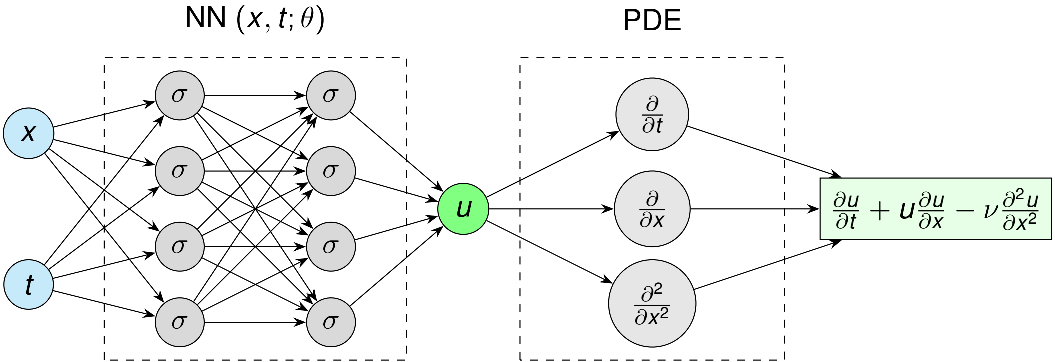
G.E.Karniadakis, I.G.Kevrekidis, L. Lu, P.Perdikaris, S. Wang, L. Yang, *Physics-informed machine learning*, Nature Reviews Physics, 3, 422-440, 2021.

Mějme následující PDR

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = 0$$

Řešení této rovnice budeme hledat pomocí plně propojené neuronové sítě.

Physics-informed neural networks



Obecně, mějme rovnici tvaru

$$
\begin{aligned}
\frac{\partial u}{\partial t} + \mathcal{N}(u, \nabla u, \nabla^2 u, \ldots) &= 0 \text{ na } \Omega \times (0, T), \\
u\,|_{\Omega} &= g \text{ na } (0, T), \\
u\,|_{t=0} &= u_0 \text{ na } \Omega.
\end{aligned}
$$

Ztrátovou funkci pak definujeme jako:

$$\mathcal{L} = \mathcal{L}_{PDE} + \lambda_1 \mathcal{L}_{ini} + \lambda_2 \mathcal{L}_{bnd},$$

kde

$$\begin{aligned}
\mathcal{L}_{PDE} &= \frac{1}{N_{PDE}} \sum_{i=1}^{N_{PDE}} \left(\frac{\partial u(\vec{x}_i, t_i)}{\partial t} + \mathcal{N}(u(\vec{x}_i, t_i), \nabla u(\vec{x}_i, t), \nabla^2 u(\vec{x}_i, t), \ldots) \right)^2 \\
&\quad \text{pro } (\vec{x}_i, t_i) \in \Omega \times (0, T), \\
\mathcal{L}_{ini} &= \frac{1}{N_{ini}} \sum_{i=1}^{N_{ini}} \left(u(\vec{x}_i, 0) - u_0(\vec{x}_i) \right)^2 \text{ pro } \vec{x}_i \in \Omega, \\
\mathcal{L}_{bnd} &= \frac{1}{N_{bnd}} \sum_{i=1}^{N_{bnd}} \left(u(\vec{x}_i, t_i) - g(\vec{x}_i, t_i) \right)^2 \text{ pro } (\vec{x}_i, t_i) \in \Omega \times (0, T).
\end{aligned}$$

Physics-informed neural networks

- Body $(\vec{x}_i, t_i) \in \Omega \times (0, T)$, $(\vec{x}_i, t_i) \in \partial\Omega \times (0, T)$ a $\vec{x}_i \in \Omega$ volíme náhodně.
- Velkou výhodou PINNů je to, že jsou tzv. bez síťové. Nutnost vytvářet numerickou síť je přitom jednou z největších komplikací při řešení PDR.
- Do ztrátové funkce lze přitom také přidat člen pro fitování na experimentálně naměřená data

$$\mathcal{L}_{data} = \left\| u(\vec{x}_i, t_i) - d(\vec{x}_i, t_i) \right\|_2^2 .$$

M. F. Fathi, I. Perez-Raya, A. Baghaie, P. Berg, G. Janiga, A. Arzani, R. M. D'Souza, *Super-resolution and denoising of 4D-Flow MRI using physics-Informed deep neural nets*, Computer Methods and Programs in Biomedicine, Volume 197, 2020.

Physics-informed neural networks

- PINNy lze také snadno použít pro optimalizace s vazbami danými pomocí PDR
- Mějme úlohu

  $$\min_{\vec{w}} J(u(\vec{w})),$$

  kde $u(\vec{w})$ je řešení PDR

  $$\begin{aligned} u_t + \mathcal{A}(u, \vec{w}_{pde}) &= 0 \text{ na } \Omega \times (0, T), \\ u\,|_{t=0} &= u_0(\vec{w}_{ini}) \text{ na } \Omega, \\ u\,|_{\partial\Omega} &= g(\vec{w}_{bnd}) \text{ na } (0, T) \end{aligned}$$

  a $\vec{w} \equiv (\vec{w}_{pde}^T, \vec{w}_{ini}^T, \vec{w}_{bnd}^T)^T$.
- Je-li $u(\vec{w})$ reprezentováno pomocí NN, lze počítat $\nabla_{\vec{w}} u$ snadno pomocí AD bez nutnosti řešit adjungované PDR.

PINN - slabé řešení

- Hlavní výhoda PINNů není v samotném hledání řešení PDR - na to jsou klasické metody stále efektivnější.
- Výhody jsou:
  - možnost snadno fitovat řešení PDR s daty,
  - řešení optimalizačních úloh s vazbami danými pomocí PDR,
  - možnost kombinovat ostatními typy NN.

PINN - slabé řešení

- Základní PINNy hledají klasické řešení.
- Při numerickém řešení PDR ale preferujeme spíše *slabé řešení*.

Uvažujme Laplaceovu úlohu:

$$\begin{aligned} -\Delta u &= 0 \text{ na } \Omega, \\ u\,\big|_{\partial\Omega} &= g. \end{aligned}$$

PINN - slabé řešení

První rovnici vynásobíme tzv. testovací funkcí $\varphi \in C_0^\infty(\Omega)$ nulovou na hranici $\partial\Omega$ a integrujeme:

$$\int_\Omega -\Delta u \varphi dx = 0$$

Použijeme Greenovu větu a dostaneme:

$$\int_\Omega -\Delta u \varphi dx = \int_{\partial\Omega} \varphi \nabla u \vec{n} dS - \int_\Omega \nabla u \cdot \nabla \varphi dx = -\int_\Omega \nabla u \cdot \nabla \varphi dx.$$

PINN - slabé řešení

- Lze ukázat, že pokud je $u \in C^2(\Omega)$ a splňuje

$$-\int_\Omega \nabla u \cdot \nabla \varphi dx = 0, \tag{1}$$

pro libovolné $\varphi \in C_0^\infty(\Omega)$, pak u je klasické řešení původní rovnice.

- Pokud je $u \in H_2^1(\Omega)$ a splňuje (1) pro libovolné $\varphi \in C_0^\infty(\Omega)$, pak jde o slabé řešení.
- Výhoda slabého řešení je v tom, že vyžaduje jen derivace do prvního řádu a to jen skoro všude.
- Ukážeme si, jak hledat slabé řešení pomocí PINNů.

G.Bao, X. Ye, Y. Zang, H. Zhou, *Numerical solution of inverse problems by weak adversarial networks*, vol.36, no.11, 115003, 2020.

PINN - slabé řešení

Zavedeme značení

$$(\mathcal{A}[u], \varphi) = \int_\Omega \nabla u \nabla \varphi dx$$

a definujeme indukovanou normu

$$\|\mathcal{A}[u]\| := \sup_{\varphi \neq 0} \frac{(\mathcal{A}[u], \varphi)}{\|\varphi\|_{H^1}},$$

kde H^1-normu definujeme jako

$$\|\varphi\|^2_{H^1(\Omega)} = \int_\Omega \varphi^2 + |\nabla \varphi|^2 dx.$$

PINN - slabé řešení

Jelikož řešení úlohy má splňovat

$$(\mathcal{A}[u], \varphi)$$

lze ho hledat i jako

$$\min_u \|\mathcal{A}[u]\| = \min_u \sup_{\varphi \neq 0} \frac{(\mathcal{A}[u], \varphi)^2}{\|\varphi\|_{H^1}}$$

Pro okrajové podmínky pak hledáme

$$\min_u \|\mathcal{B}[u]\|^2_{L^2(\partial\Omega)}$$

PINN - slabé řešení

- Funkce u a φ aproximujeme pomocí dvou neuronových sítí funkcemi $u_{\vec{\theta}_1}$ a $\varphi_{\vec{\theta}_2}$.
- Celkem definujeme ztrátovou funkci

$$L(\vec{\theta}_1, \vec{\theta}_2) = \min_{\vec{\theta}_1} \max_{|\vec{\theta}_2|^2 \leq B} \frac{(\mathcal{A}[u_{\vec{\theta}_1}], \vec{\varphi}_{\theta_2})^2}{\left\|\varphi_{\vec{\theta}_2}\right\|_{H^1}} + \beta \min_{\vec{\theta}_1} \left\|\mathcal{B}[u_{\vec{\theta}_1}]\right\|^2_{L^2(\partial\Omega)}$$

- Aproximace integrálu se provádí pomocí Monte-Carlo integrací.
- Právě řešení min-max sedlové úlohy se vyskytuje i u tzv. *generative adversarial networks*.