

REDES NEURAIS INFORMADAS PELA FÍSICA COMO ARCABOUÇO UNIFICADO PARA PROBLEMAS DIRETOS E INVERSOS: ANÁLISE DE DESEMPENHO EM IDENTIFICAÇÃO DE PARÂMETROS

Abdeladhim Tahimi¹

¹Campus de Engenharias e Ciências Agrárias, Rio Largo, abdeladhim.tahimi@ceca.ufal.br

Resumo

As Redes Neurais Informadas pela Física (Physics-Informed Neural Networks, PINNs) representam um paradigma emergente no Aprendizado de Máquina Científico (Scientific Machine Learning, SciML), integrando leis físicas em redes neurais para solução de equações diferenciais. Este trabalho apresenta uma análise comparativa entre PINNs e métodos clássicos, utilizando problemas canônicos para revelar seus domínios de aplicabilidade. Primeiramente, demonstramos que resolvidores tradicionais como Runge-Kutta superam PINNs em problemas diretos bem-postos por ordens de magnitude em acurácia e eficiência. Em seguida, aplicamos PINNs a um problema inverso de identificação de parâmetros com dados ruidosos, onde a física atua como regularizador, filtrando ruído e reconstruindo a dinâmica do sistema. Os resultados revelam que a força das PINNs não reside na competição com métodos consolidados, mas em cenários complexos — problemas mal-postos, dados esparsos ou heterogêneos — onde abordagens tradicionais falham. Este estudo posiciona PINNs como ferramenta indispensável para problemas inversos e aplicações avançadas em engenharia.

Palavras-chave: *Redes Neurais Informadas pela Física, Problemas Inversos*

Introdução

A integração de conhecimento físico com aprendizado de máquina está revolucionando a ciência computacional, consolidando o campo do Aprendizado de Máquina Científico (*Scientific Machine Learning*, SciML). As Redes Neurais Informadas pela Física (*Physics-Informed Neural Networks*, PINNs), propostas por RAISSI et al. (2019), emergiram como abordagem promissora que difere fundamentalmente de resolvidores numéricos clássicos: em vez de discretizar domínios em malhas, PINNs empregam redes neurais para aprender representações contínuas de soluções, injetando leis físicas diretamente na função de perda através de diferenciação automática.

Apesar da crescente popularidade, comparações diretas com métodos tradicionais em problemas simples revelam desempenho inferior. Este resultado, embora verdadeiro, alimenta interpretações equivocadas. Avaliar PINNs em domínios onde métodos clássicos foram aperfeiçoados por décadas é inadequado — como testar veículos *off-road* em pistas de Fórmula

Este trabalho oferece análise aprofundada do arcabouço PINN através de dois estudos: (1) solução de problema de valor inicial, estabelecendo linha de base (*baseline*) de comparação com resolvidores clássicos; (2) problema inverso de identificação de parâmetros com dados ruidosos, demonstrando a força das PINNs como regularizador físico. Nosso objetivo é revelar onde reside o verdadeiro valor das PINNs: problemas mal-postos, dados esparsos e cenários onde métodos tradicionais são inviáveis.

Metodologia

Formulação da PINN

Consideramos o problema de valor inicial: $y'(t) = -\theta y(t)$, $y(0) = 1$, com solução analítica $y(t) = \exp(-\theta t)$ em $t \in [0, 1]$.

A PINN emprega rede neural *feed-forward* (propagação direta) $\hat{y}(t; w)$ com 3 camadas ocultas de 20 neurônios cada, função de ativação tanh e 901 parâmetros treináveis w . A função de perda combina o resíduo da EDO e a condição inicial:

$$L(w) = \frac{1}{N_c} \sum_{i=1}^{N_c} [R(t_i)]^2 + \lambda \cdot (\hat{y}(0) - 1)^2 \quad (1)$$

onde $R(t) = \hat{y}'(t) + \theta \hat{y}(t)$ é o resíduo obtido por diferenciação automática, $N_c = 100$ são pontos de colocação e λ pondera a condição inicial. A otimização utiliza *Adam* (KINGMA e BA, 2014) por 1000 épocas.

Resolvidores clássicos e problema inverso

Implementamos Euler explícito/implícito, Runge-Kutta de 2ª ordem (Heun) e RK4. Para o problema inverso, estendemos a PINN para identificar θ a partir de 500 pontos sintéticos ruidosos: $y_{\text{data},i} = \exp(-3,0 \cdot t_i) + \varepsilon_i$, onde $\varepsilon_i \sim N(0, 0,02^2)$. O parâmetro θ torna-se variável treinável (inicializado em 1,0), com função de perda aumentada. Comparamos com regressão log-linear e mínimos quadrados não-lineares (*curve_fit* do SciPy). Experimentos em *Google Colaboratory* (CPU 2 núcleos) com *Python 3.12*, *Tensor Flow 2.19*.

Resultados e discussão

Problema direto: PINN versus resolvidores clássicos

A Tab. 1 apresenta as cinco melhores configurações da PINN após varredura de hiperparâmetros. Notavelmente, todas utilizaram $\lambda = 1$, indicando que balanço equitativo entre resíduo da EDO e condição inicial foi ótimo.

A Tab. 2 revela resultado categórico: métodos clássicos superaram PINNs por ordens de magnitude. RK4 atingiu erro $3,33 \times 10^{-7}$ com apenas 10 passos em $< 0,001$ s, versus ~ 60 s da melhor PINN. Para problemas bem-postos, resolvidores tradicionais são ferramentas inigualáveis em eficiência.

Tabela 1: Cinco melhores configurações da PINN (largura, profundidade, taxa aprendizado, λ)

Posição	Config. (w, d, lr, λ)	Erro L_2	Tempo (s)
1	(50, 2, 10^{-3} , 1)	$8,29 \times 10^{-4}$	46,1
2	(50, 3, 10^{-3} , 1)	$1,09 \times 10^{-3}$	59,9
3	(20, 3, 10^{-3} , 1)	$1,34 \times 10^{-3}$	61,4

Tabela 2: Desempenho de resolvidores clássicos (erro absoluto $< 10^{-4}$)

Método	Ordem	Passos	Erro Absoluto
Euler Explícito	$O(h)$	2560	$7,19 \times 10^{-5}$
RK 2ª Ordem	$O(h^2)$	40	$3,90 \times 10^{-5}$
RK 4ª Ordem	$O(h^4)$	10	$3,33 \times 10^{-7}$

Problema inverso: identificação de parâmetros

No problema inverso ($\theta_{\text{verdadeiro}} = 3,0$), a PINN estimou $\theta = 2,886$ (erro: 0,114), resultado comparável ao método log-linear ($\theta = 3,115$, erro: 0,115), enquanto `curve_fit` obteve maior precisão ($\theta = 2,990$, erro: 0,010). Mais importante que a precisão pontual é a capacidade da PINN de reconstruir a dinâmica completa. A Fig. 1 demonstra que a solução aprendida $\hat{y}(t)$ filtra eficazmente o ruído, convergindo para a solução verdadeira (não utilizada no treinamento). Este resultado visualiza a regularização física: a EDO na função de perda impede sobreajuste (*overfitting*), forçando a rede a encontrar função que obedeça à lei física subjacente.

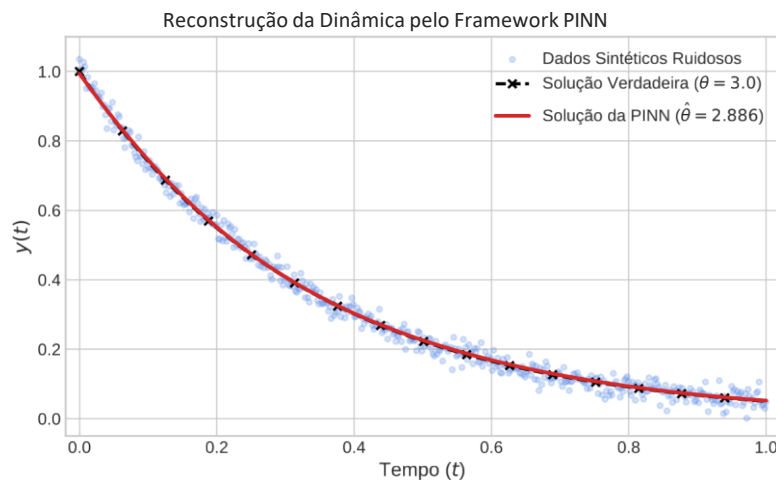


Figura 1: Reconstrução da dinâmica pela PINN. A rede aprende solução suave que filtra ruído e converge para a solução verdadeira (tracejada)

Aplicações avançadas das PINNs

O valor do arcabouço PINN se revela em cenários complexos:

- Dados esparsos: em regimes com poucos dados, a regularização física previne sobreajuste, mantendo soluções fisicamente consistentes.
- Fusão de dados heterogêneos: PINNs integram trivialmente medições de diferentes grandezas (e.g., $y(t)$ e $y'(t)$) adicionando termos à função de perda.
- Parâmetros variáveis: quando parâmetros variam no domínio, $\theta(t)$, PINNs representamnos com redes neurais auxiliares, otimizadas simultaneamente.
- Problemas mal-postos: reconstrução 3D de escoamentos a partir de dados 2D é viável com PINNs assimilando medições enquanto equações de Navier-Stokes guiam a solução (CAI et al., 2021).
- Descoberta de equações: PINNs podem aprender formas funcionais desconhecidas de termos em EDOs a partir de dados

Referências

- RAISSI, M., Perdikaris, P., Karniadakis, G. E., 2019. 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.
- KARNIADAKIS, G. E., et al., 2021. Physics-informed machine learning, *Nature Reviews Physics*, 3(6), 322-343.
- KINGMA, D. P., Ba, J., 2014. Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.
- CAI, S., et al., 2021. Physics-informed neural networks for fluid mechanics: A review, *Acta Mechanica Sinica*, 37(12), 1727-1738.
- VIRTANEN, P., et al., 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, *Nature Methods*, 17(3), 261-272.