

# Dados de proveniência para redes neurais guiadas pela Física: o caso da equação eikonal

Lyncoln S. de Oliveira<sup>1</sup>, Rômulo M. Silva<sup>2</sup>, Liliane Kunstmann<sup>1</sup>, Débora Pina<sup>1</sup>, Daniel de Oliveira<sup>3</sup>, Alvaro L. G. A Coutinho<sup>2</sup>, Marta Mattoso<sup>1</sup>

<sup>1</sup>PESC/COPPE- Universidade Federal do Rio de Janeiro (UFRJ)

{oliveiral, lneves, dbpina, marta}@cos.ufrj.br

<sup>2</sup>PEC/COPPE- Universidade Federal do Rio de Janeiro (UFRJ)

{romulo.silva, alvaro}@coc.ufrj.br

<sup>3</sup>Instituto de Computação - Universidade Federal Fluminense (UFF)

danielcmo@ic.uff.br

**Abstract.** *Physics Informed Neural Networks (PINNs) have been showing a great impact on numerical methods applications. Despite the complexity of configuration and model generation, once trained, it presents a significant improvement in calculation time compared to numerical methods. The Physics constraining in training takes place through the modeling of new components added to the neural network's loss function. Such components augment the hyperparameter settings. In this work, we show how collecting provenance data can help the scientist to evaluate hyperparameters in the training of PINNs. We present experiments of PINNs with the partial differential equation given by the Eikonal equation.*

**Resumo.** *As redes neurais guiadas pela Física (PINNs) vêm revolucionando a aplicação de métodos numéricos. Apesar da complexidade de configuração e geração do modelo, uma vez treinado, o mesmo mostra um ganho significativo em relação ao tempo de cálculo dos métodos numéricos. A incorporação da Física no treinamento se dá por meio da modelagem de novos componentes na função de perda da rede neural. Tais componentes aumentam as configurações de hiperparâmetros. Mostramos como a coleta de dados de proveniência pode ajudar na avaliação de hiperparâmetros no treinamento de PINNs. Apresentamos experimentos de PINNs com a equação diferencial parcial dada pela equação Eikonal.*

## 1. Introdução

O Aprendizado de Máquina aplicado às Redes Neurais Profundas (DNNs) tem mostrado eficiência em uma variedade de problemas [Kidger and Lyons 2020]. Uma das variações recentes de DNNs são as redes neurais guiadas pela Física (*Physics-Informed Neural Networks*-PINNs) que vêm revolucionando a solução de problemas governados por equações diferenciais parciais na ciência e na engenharia [Raissi et al. 2019]. A incorporação da Física no treinamento de DNNs se dá por meio de novos componentes na função de perda, refletindo o resíduo da equação diferencial e suas condições de contorno. Tais componentes aumentam as configurações de hiperparâmetros dificultando a escolha do modelo da PINN.

As PINNs podem ser definidas e treinadas usando os ambientes clássicos de DNN, como TensorFlow e PyTorch. No entanto, são incipientes os recursos de AutoML [Karmaker et al. 2021] para PINNs e o cientista precisa gerenciar a definição de configurações e desenvolver recursos próprios para a análise dos modelos considerando os hiperparâmetros e novos componentes das PINNs. Na prática, o cientista especialista em PINNs e na Física em questão precisa configurar, por ex., o TensorBoard para exibir os valores dos componentes da função de perda por meio de rotinas de sumarização de dados do TensorFlow. Essa estratégia requer a programação de um *script* específico para relacionar informações em diferentes arquivos e persistí-los em arquivos tipo CSV. Além disso, em problemas de análise de múltiplos modelos é necessário planejar a organização desses arquivos/diretórios de diversas configurações de modelos.

Uma forma de minimizar esta dificuldade de análise de configurações é usar um SGBD para fazer a persistência dos dados e consultas. No entanto, é preciso uma modelagem dos dados da DNN e os específicos da PINN, o que não é trivial, nem a combinação de instalação/uso do SGBD junto aos ambientes de DNN. Além disso, cada cientista faria sua própria modelagem gerando retrabalho e heterogeneidade. Para facilitar essa modelagem e uso do SGBD, foi proposta a DNNProv, uma biblioteca de serviços de proveniência para DNNs em geral, sendo uma evolução da Keras-Prov [Pina et al. 2021], favorecendo a estensibilidade. O objetivo é aproveitar a captura de dados de proveniência durante o treinamento para servir tanto à reprodutibilidade, quanto à análise de dados, metadados e o caminho de derivação dos dados. Com base nessa análise qualitativa, adaptações (sintonia fina) são realizadas e representadas junto à proveniência.

Ao adotar o padrão W3C PROV, a DNNProv reduz os problemas de heterogeneidade de representação e evita a redundância de dados em CSVs e de proveniência. Os dados de proveniência têm uma representação e propósito natural para apoiar essa análise de inúmeras configurações de modelos em DNNs. O W3C PROV define proveniência como *"informação sobre entidades, atividades e pessoas envolvidas na produção de um dado ou coisa, que pode ser usada para formar avaliações sobre sua qualidade, validade ou confiabilidade"* [Moreau and Groth 2013]. Esse uso da proveniência está alinhado com a análise de modelos. Além disso, a estensibilidade do modelo de dados PROV-DM favorece a modelagem de novos dados, como é o caso da função de perda das PINNs.

Apoiar o humano no processo iterativo de definição e escolha de modelos de DNN em larga escala, se faz necessário conforme [Kumar et al. 2021, Vartak et al. 2016, Wang et al. 2019]. No caso de PINNs esse apoio é ainda mais importante. O objetivo deste trabalho é abordar o problema de escolha de modelos em PINNs explorando a geração de dados de proveniência e seu uso para monitorar a evolução do treinamento da PINN. Este monitoramento visa a complementar ferramentas de visualização. Neste trabalho, avaliamos a biblioteca DNNProv quanto à generalidade do modelo de dados de proveniência na representação de métricas de PINNs, explorando um novo problema direto e inverso com escala maior, treinando PINNs com diferentes arquiteturas e variando sua complexidade.

A PINN avaliada neste trabalho tem o objetivo de resolver a equação diferencial parcial (EDP) dada pela equação eikonal fatorada (EEF), que descreve fenômenos como propagação de ondas para meios acústicos e elásticos, bem como meios eletromagnéticos [Debnath 2012]. Esta PINN retrata um tipo de problema de tomografia sísmica. Os experimentos apresentados são uma evolução de trabalhos anteriores. Em

[Silva and Coutinho 2020] PINNs são usadas para a resolução do problema direto e inverso associado à EEF. Em [Silva et al. 2021], esta PINN foi treinada e refinada com base em dados de proveniência capturada pela DNNProv, mas ainda em pequena escala, usando a plataforma Grid’5000. No entanto, em problemas reais de maior proporção, o treinamento, as adaptações e geração do modelo da PINN torna a escolha de configuração de hiperparâmetros mais complexa e difícil de gerenciar.

Os experimentos com PINNs deste trabalho mostraram como os dados de proveniência podem complementar as visualizações gráficas e aumentar a confiabilidade no processo de experimentar configurações das PINNs em larga escala. O trabalho foi estruturado da seguinte maneira. A Seção 2 apresenta resumidamente a teoria que envolve PINNs em conjunto com a nossa arquitetura de rede, a Seção 3 detalha o estudo de caso e a Seção 4 conclui o trabalho.

## 2. Physics-Informed Neural Networks (PINNs)

Como apresentado em [Raissi et al. 2019, Silva et al. 2021], uma maneira eficiente de resolver problemas diretos e inversos associados a EDPs é por meio de PINNs. Para isso, é necessário informar as leis físicas regidas pelo problema para a PINN por meio da função de perda da rede, que é mostrada na equação (1), onde  $\theta$  representa os parâmetros da PINN e  $\mathcal{T}_i, i = \mathcal{R}, \mathcal{BC}, \mathcal{D}$  os conjuntos de pontos de avaliação do resíduo, condições de contorno, e dos dados da EEF. O objetivo é minimizar essa função durante o treinamento da PINN.

$$\mathcal{L}(\theta; \mathcal{T}) = w_{\mathcal{R}}\mathcal{L}_{\mathcal{R}}(\theta; \mathcal{T}_{\mathcal{R}}) + w_{\mathcal{BC}}\mathcal{L}_{\mathcal{BC}}(\theta; \mathcal{T}_{\mathcal{BC}}) + w_{\mathcal{D}}\mathcal{L}_{\mathcal{D}}(\theta; \mathcal{T}_{\mathcal{D}}) \tag{1}$$

A função de perda da rede  $\mathcal{L}(\theta; \mathcal{T})$  possui três componentes. O componente  $\mathcal{L}_{\mathcal{R}}$  quantifica a qualidade de aproximação do resíduo da EDP ( $\mathcal{R}$ , na Figura 1), o  $\mathcal{L}_{\mathcal{BC}}$  é responsável pela qualidade da aproximação das condições de contorno ( $\mathcal{BC}$ , na Figura 1), por fim, o  $\mathcal{L}_{\mathcal{D}}$  incorpora os dados ( $\mathcal{D}$ , na Figura 1). Cada componente pode estar associado a um peso  $w$ . O esquema de PINN utilizado neste trabalho é apresentado na Figura 1, onde uma rede auxiliar é também treinada para resolver o problema inverso. Note que os termos associados às condições de contorno e ao resíduo estão definidos para o caso teste deste trabalho em [Silva et al. 2021].

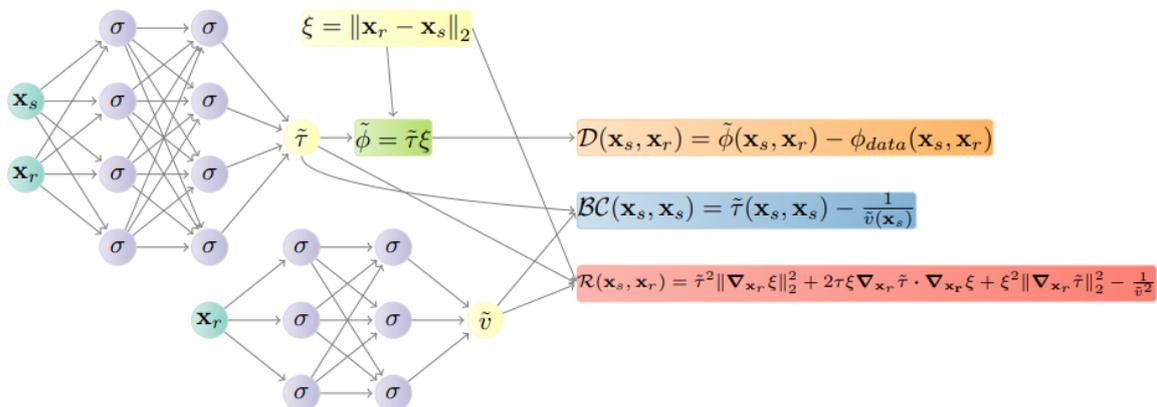


Figura 1. Esquema PINN para resolver o problema inverso EEF.  $\tilde{\phi}(\mathbf{x}_s, \mathbf{x}_r)$  e  $\tilde{v}(\mathbf{x}_r)$  são respectivamente aproximações dos tempos de transito e velocidade de propagação da onda no meio acústico [Silva et al. 2021].

### 3. Experimentos com Proveniência para PINNs

O ambiente computacional utilizado foi um computador pessoal com 32 GB de memória RAM, processador Intel I5-8400, placa de vídeo GeForce GTX 1050Ti e sistema operacional Ubuntu-20.04.3. O sistema de banco de dados é o MonetDB, que visa a representação de dados científicos, como vetores que foram usados para representar os componentes da função de perda, o que facilitou a análise e consultas para PINNs.

O conjunto de dados de entrada utilizado para aplicação das PINNs foi o mesmo de [Silva et al. 2021], sendo referente a dados sísmicos e de radar de penetração no solo. Este trabalho explorou configurações em escala maior. Conforme a Figura 1, foram utilizadas duas redes neurais, uma para estimar o tempo de transito (TT), relacionado ao problema direto, e outra para estimar a velocidade de propagação da onda (Vel), relacionado ao problema inverso. Essas redes estão indiretamente conectadas para o cálculo da função de perda.

Os pesos  $w$  da função de perda apresentados na Equação 1 foram de  $w_{\mathcal{R}} = w_{BC} = 1$ , e  $w_{\mathcal{D}} = 25$ . A determinação dos valores dos pesos é um problema em aberto. Sua definição depende de uma análise envolvendo otimização multi-objetivo. Neste experimento, foi usada a intuição do cientista, baseada em experimentos prévios e adaptações ao longo das ações de sintonia fina dos hiperparâmetros. As configurações utilizadas são os parâmetros *batch size* = 3000, número de épocas = 400000 e taxa de aprendizado com a função de decaimento exponencial, esta atribui um valor inicial para a taxa e a diminui ao decorrer das épocas.

Em relação à proveniência, as atividades capturadas durante a execução são treinamento e adaptação. A atividade de treinamento usa entidades como o conjunto de dados de treino e os hiperparâmetros, que são por exemplo: número de épocas, tamanho do *batch*, nome do otimizador, taxa de aprendizado, função de ativação e arquitetura da rede. A entidade de saída conta com as métricas que avaliam o modelo, como o coeficiente de determinação do modelo  $R^2$ , a função de perda e seus componentes, o tempo decorrido e a data e hora do término da execução de cada intervalo de épocas. A atividade de adaptação usa o conjunto de dados gerados pelo treinamento e os parâmetros que serão adaptados. A entidade gerada pela atividade de adaptação pode ser o identificador da adaptação, o novo valor para a taxa de aprendizado, a época, a data e hora da adaptação.

Desta forma, a cadeia de derivação entre treinamento e adaptação das PINNs fica representada por meio dos relacionamentos definidos no PROV. Nove configurações de PINNs são apresentadas por meio de consultas à base de proveniência, como mostra a Tabela 1. A consulta da Tabela 2 agrupa as entidades usadas e geradas no treinamento para analisar em quais épocas foram obtidos os melhores valores da métrica  $R^2$  para cada configuração, em conjunto com o seu tempo de obtenção e a média do tempo de processamento de cada intervalo de 100 épocas. Por meio desta consulta o especialista verificou que aumentar a complexidade das PINNs não melhorou a métrica  $R^2$ , interrompendo essa linha de configurações. Foi possível também analisar a função de perda, em conjunto com seus componentes na Tabela 3. Assim, o especialista percebeu que ao aumentar a complexidade da PINN a função de perda tende a ser menor, porém, essa redução não acompanha a melhora da capacidade preditiva do modelo dada pela métrica  $R^2$ , e isso é um bom indicativo de realizar o experimento com uma nova abordagem para a função de perda. Esta análise de dados pode ser complementada com visualizações gráficas utilizando bibliotecas auxiliares.

**Tabela 1. Quais são as configurações das PINNs treinadas?**

ID	Otimizador	Ativação TT	Ativação Vel	N.º de Neurônios TT	N.º de Neurônios Vel	N.º de Camadas intermediárias TT	N.º de Camadas intermediárias Vel
1	Adam	tanh	tanh	20	32	8	4
2	Adam	relu	relu	20	32	8	4
3	Adam	sigmoid	sigmoid	20	32	8	4
4	Adam	relu	tanh	20	32	8	4
5	RMSProp	tanh	tanh	20	32	8	4
6	Adam	tanh	tanh	40	64	16	8
7	RMSProp	tanh	tanh	40	64	16	8
8	Adam	tanh	tanh	20	32	16	8
9	RMSProp	tanh	tanh	20	32	16	8

**Tabela 2. Quais são os maiores valores de  $R^2$ , qual é a época e tempo gasto para obtenção?**

ID	Máximo $R^2$	Época de obtenção	Tempo até obtenção(m)	Média 100 épocas(s)
5	0.415	399600	104.029	1.562
1	0.374	391500	99.833	1.530
9	0.273	399600	182.018	2.733
6	0.246	67200	43.288	3.865
2	0.245	41600	10.546	1.521
7	0.222	46100	29.896	3.891
8	0.185	66300	30.255	2.738
3	0.150	399000	102.477	1.541
4	0.045	19200	4.861	1.519

**Tabela 3. Quais são os menores valores da função perda e seus componentes, qual é a época e tempo gasto para obtenção?**

ID	$\mathcal{L}$	$\mathcal{L}_{\mathcal{R}}$	$\mathcal{L}_{BC}$	$\mathcal{L}_{\mathcal{D}}$	Época de obtenção	Tempo até obtenção(m)
6	$3.166 \cdot 10^{-6}$	$3.136 \cdot 10^{-6}$	$1.910 \cdot 10^{-10}$	$1.211 \cdot 10^{-9}$	390500	251.547
8	$5.801 \cdot 10^{-6}$	$5.744 \cdot 10^{-6}$	$3.608 \cdot 10^{-10}$	$2.263 \cdot 10^{-9}$	398800	181.986
7	$6.940 \cdot 10^{-6}$	$5.787 \cdot 10^{-6}$	$1.266 \cdot 10^{-9}$	$4.608 \cdot 10^{-8}$	396600	257.195
9	$2.570 \cdot 10^{-5}$	$2.068 \cdot 10^{-5}$	$2.079 \cdot 10^{-9}$	$2.009 \cdot 10^{-7}$	394600	179.740
5	$2.709 \cdot 10^{-5}$	$2.498 \cdot 10^{-5}$	$3.976 \cdot 10^{-10}$	$8.418 \cdot 10^{-8}$	399600	104.029
1	$6.830 \cdot 10^{-5}$	$4.026 \cdot 10^{-5}$	$6.278 \cdot 10^{-10}$	$1.122 \cdot 10^{-6}$	392400	100.062
2	$2.500 \cdot 10^{-4}$	$1.957 \cdot 10^{-4}$	$3.416 \cdot 10^{-9}$	$2.172 \cdot 10^{-6}$	397000	100.640
4	$2.579 \cdot 10^{-4}$	$1.906 \cdot 10^{-4}$	$7.775 \cdot 10^{-9}$	$2.694 \cdot 10^{-6}$	391000	98.988
3	$3.742 \cdot 10^{-3}$	$6.204 \cdot 10^{-4}$	$4.821 \cdot 10^{-8}$	$1.249 \cdot 10^{-4}$	241000	61.897

A análise de proveniência de adaptações permite observar o seu impacto sobre o modelo durante o treinamento. O especialista pode consultar as entidades usadas e geradas pelas atividades de treinamento e adaptação, podendo associar dados de treinamento anteriores e posteriores à adaptação do modelo. Por exemplo, o especialista pode associar os dados de entidades geradas no treinamento com dados usados em uma adaptação específica, utilizando o identificador os une. Assim, o especialista pode buscar quais foram os valores para as métricas  $R^2$ , função de perda e seus componentes para uma taxa de aprendizado adaptada. A Tabela 4 mostra os resultados dessas consultas buscando as épocas 300000 e 300100.

**Tabela 4. Quais os valores de adaptação de taxa de aprendizado na época 30000 e o impacto para o treinamento de id 5?**

Época	$R^2$	$\mathcal{L}$	$\mathcal{L}_{\mathcal{R}}$	$\mathcal{L}_{BC}$	$\mathcal{L}_{\mathcal{D}}$	Nova taxa de aprendizado
300000	0.204	$3.010 \cdot 10^{-2}$	$4.940 \cdot 10^{-4}$	$6.389 \cdot 10^{-6}$	$1.185 \cdot 10^{-3}$	$7.397 \cdot 10^{-4}$
300100	0.209	$2.383 \cdot 10^{-2}$	$4.336 \cdot 10^{-4}$	$2.106 \cdot 10^{-6}$	$9.359 \cdot 10^{-4}$	$7.390 \cdot 10^{-4}$

## 4. Conclusão

Apresentamos configurações de arquiteturas de PINNs para a resolução do problema proposto em [Silva et al. 2021] em maior escala. A análise e adaptações na busca de melhores configurações de modelos contaram com apoio de uma base de dados de proveniência gerada pela DNNProv. Verificamos como os dados de proveniência ajudaram o especialista, desenvolvedor de PINNs, em análises dos resultados e adaptações como a mudança da taxa de aprendizado após consultas. Como trabalhos futuros, pretendemos investir na calibração dos pesos das métricas e associar esse apoio às escolhas automáticas das soluções de AutoML que venham a surgir para PINNs, mantendo o humano na tomada de decisão.

## Agradecimentos

Trabalho realizado com apoio das agências de fomento CNPq, FAPERJ e CAPES.

## Referências

- Debnath, L. (2012). *First-Order Nonlinear Equations and Their Applications*, pages 227–256. Birkhäuser Boston, Boston.
- Karmaker, S. K., Hassan, M. M., Smith, M. J., Xu, L., Zhai, C., and Veeramachaneni, K. (2021). Automl to date and beyond: Challenges and opportunities. *ACM Comput. Surv.*, 54(8):1–36.
- Kidger, P. and Lyons, T. (2020). Universal approximation with deep narrow networks. In *Conference on learning theory*, pages 2306–2327.
- Kumar, A., Nakandala, S., Zhang, Y., Li, S., Gemawat, A., and Nagrecha, K. (2021). Cerebro: A layered data platform for scalable deep learning. In *11th Annual Conference on Innovative Data Systems Research (CIDR’21)*.
- Moreau, L. and Groth, P. (2013). *Provenance: An Introduction to PROV*. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool Publishers.
- Pina, D., Neves, L., de Oliveira, D., and Mattoso, M. (2021). Captura automática de dados de proveniência de experimentos de aprendizado de máquina com keras-prov. In *Anais Estendidos do XXXVI SBBD*, pages 69–74. SBC.
- Raissi, M., Perdikaris, P., and 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.
- Silva, R., Pina, D., Kunstmann, L., de Oliveira, D., Valduriez, P., Coutinho, A., and Mattoso, M. (2021). Capturing provenance to improve the model training of pinns: first hand-on experiences with grid5000. In *42nd CILAMCE*, pages 1–7.
- Silva, R. M. and Coutinho, A. L. (2020). Physics-informed neural networks for the factored eikonal equation. In *41st CILAMCE*.
- Vartak, M., Subramanyam, H., Lee, W.-E., Viswanathan, S., Husnoo, S., Madden, S., and Zaharia, M. (2016). Modeldb: a system for machine learning model management. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, pages 1–3.
- Wang, D., Weisz, J. D., Muller, M., Ram, P., Geyer, W., Dugan, C., Tausczik, Y., Samulowitz, H., and Gray, A. (2019). Human-ai collaboration in data science: Exploring data scientists’ perceptions of automated ai. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–24.