

# PositPR.jl: Uma Biblioteca para Aritmética Posit e Integração com Frameworks de Machine Learning

**Mateus R. Ussui<sup>1</sup>**

**Thiago O. Quinelato<sup>2</sup>**

<sup>1</sup>Bacharelado em Matemática Industrial – Universidade Federal do Paraná

<sup>2</sup>Departamento de Matemática – Universidade Federal do Paraná

E-mail: <sup>1</sup>mateusussui@ufpr.br, <sup>2</sup>thiago.quinelato@ufpr.br

## RESUMO

O sistema numérico Posit, proposto por Gustafson e Yonemoto [2], foi concebido como alternativa ao padrão IEEE-754, buscando maior eficiência na representação de números reais. Em um Posit( $ps, es$ ), o número é descrito por quatro campos: sinal, regime, expoente e fração, compondo o formato

$$x = (-1)^s \text{useed}^k 2^e (1 + f),$$

onde  $s$  é o bit de sinal,  $\text{useed} = 2^{2^{es}}$ ,  $k$  é o valor do regime (codificado por *run-length encoding*),  $e$  é o expoente armazenado em até  $es$  bits, e  $f \in [0, 1)$  é a fração. Diferentemente do ponto flutuante tradicional, a precisão dos Posits é adaptativa: valores próximos da origem recebem mais bits de fração, enquanto números de grande magnitude expressam faixa dinâmica ampliada. Essa redistribuição interna de bits elimina redundâncias (como  $-0$  e NaNs múltiplos), reduz casos de overflow/underflow e torna o formato especialmente atrativo para computação científica, simulações iterativas e aprendizado de máquina em baixa precisão.

Embora o formato Posit tenha recebido atenção crescente na literatura, não existe atualmente nenhuma biblioteca que implemente o sistema de forma completa e funcional sem recorrer a operações intermediárias em ponto flutuante. Implementações existentes, quando disponíveis, utilizam conversões internas para Float32 ou Float64 para realizar funções matemáticas ou operações aritméticas, o que impede avaliar o comportamento numérico do Posit em sua totalidade.

Este trabalho apresenta a **PositPR.jl**, uma biblioteca desenvolvida em Julia que implementa o sistema Posit de modo completo e modular. Essa abordagem responde diretamente a uma limitação destacada na literatura: embora estudos anteriores tenham demonstrado o potencial do Posit para treinamento e inferência em baixa precisão ([1, 3, 4], p. ex.), todos eles dependem, em algum estágio crítico, de conversões para a aritmética IEEE-754, seja para funções matemáticas, operações intermediárias ou implementação de kernels. Assim, não avaliam o comportamento do Posit em sua forma plena. Na biblioteca PositPR.jl, por outro lado, todas as operações são implementadas diretamente a partir da lógica binária definida pelo formato Posit, permitindo investigar o sistema de forma isolada.

A biblioteca foi projetada para interoperar com o ecossistema SciML e com *frameworks* modernos de aprendizado de máquina, como Lux.jl. Para utilizar Posits em modelos treináveis, a infraestrutura necessária é surpreendentemente pequena: basta garantir que os inicializadores de parâmetros do modelo (como **rand**, **randn**, **zeros**, **ones** e métodos equivalentes) produzam tensores no tipo Posit{ps,es}. Uma vez que pesos e vieses são criados diretamente como Posits, toda a *pipeline* subsequente – operações matriciais, propagação direta, retropropagação e cálculo de gradientes – é automaticamente tratada pelo sistema de despacho múltiplo do Julia e pelos mecanismos de autodiferenciação disponíveis no SciML. Assim, integrar Posits a modelos de aprendizado de máquina reduz-se essencialmente à disponibilização de inicializadores

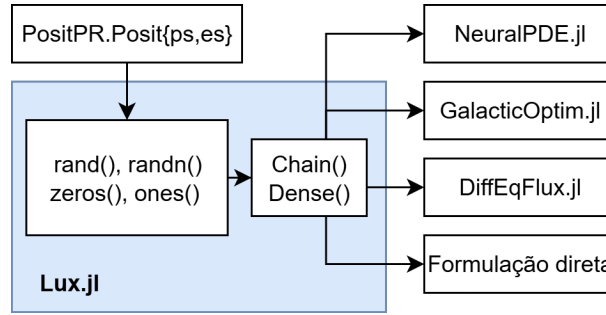


Figura 1: Fluxo de integração da PositPR.jl ao ecossistema SciML: uma vez que a inicialização em Lux.jl aceita Posits, toda a cadeia de ferramentas — NeuralPDE.jl, GalacticOptim.jl, DiffEqFlux.jl e formulações diretas — passa a operar nativamente em aritmética Posit.

compatíveis, como ilustrado na Figura 1.

Os experimentos realizados a partir de PositPR.jl incluíram análises de precisão, erros de arredondamento e comportamento do formato em diferentes escalas, bem como aplicações em redes neurais e PINNs para resolução de EDOs e EDPs simples. Os resultados demonstram que Posit{16,2} oferece boa estabilidade numérica, enquanto Posit{8,2} apresenta desempenho coerente com seu regime de baixa precisão, mantendo elevada resolução na vizinhança de zero. Redes pequenas treinadas inteiramente com Posits foram capazes de aproximar soluções analíticas, evidenciando o potencial do formato para aplicações de aprendizado de máquina e simulação numérica sensível a recursos.

PositPR.jl fornece, portanto, uma infraestrutura aberta para pesquisa em sistemas numéricos alternativos, computação de baixa precisão e integração direta entre Posits e modelos de aprendizado de máquina. Os resultados reforçam a viabilidade do Posit como substituto parcial do IEEE-754 em tarefas de computação científica, otimização e simulação orientada por eficiência. **Palavras-chave:** *Ponto flutuante; Posit; aritmética de precisão variável; Julia; SciML; métodos numéricos; aprendizado de máquina.*

## Referências

- [1] Z. Carmichael, H. F. Langroudi, C. Khazanov, J. Lillie, J. L. Gustafson e D. Kudithipudi, Deep Positron: A Deep Neural Network Using the Posit Number System, em “Proceedings of DATE 2019 – Design, Automation & Test in Europe Conference & Exhibition”, (2019) pp. 1421–1426.
- [2] J. L. Gustafson e I. T. Yonemoto, Beating Floating Point at its Own Game: Posit Arithmetic, *Supercomputing Frontiers and Innovations*, 4(2) (2017) 71–86.
- [3] J. Lu, S. Lu, Z. Wang, C. Fang, J. Lin, Z. Wang e L. Du, Training Deep Neural Networks Using Posit Number System, em “2019 32nd IEEE International System-on-Chip Conference (SOCC)”, (2019) pp. 62–67.
- [4] G. Raposo, P. Tomás e N. Roma, PositNN: Training Deep Neural Networks with Mixed Low-Precision Posit, em “Proceedings of ICASSP 2021 – IEEE International Conference on Acoustics, Speech and Signal Processing”, (2021) pp. 7908–7912.