Transformando Caixas-Pretas em Caixas de Vidro: Aumentando a Explicabilidade de Redes Neurais com Ferramentas de Visualização 3D

Karoline A F Barbosa
FEELT - UFU
Uberlândia, Brasil
karolinebarbosa2204@ufu.br

Rafael Campos Teixeira
FEELT - UFU
Uberlândia, Brasil
rafael.teixeira@ufu.br

Igor Santos Peretta
FEELT - UFU
Uberlândia, Brasil
iperetta@ufu.br

Alexandre Cardoso
FEELT - UFU
Uberlândia, Brasil
alexandre@ufu.br

***Abstract*—This article presents an ongoing project aimed at transforming the "black box" nature of neural networks into "glass box" models using 3D visualization tools. As artificial intelligence (AI) becomes increasingly integral to various fields, the need for explainability in deep learning models grows paramount. We employ the Godot Engine, an open-source platform, to create interactive visualizations that elucidate the inner workings of neural networks, facilitating user understanding of decision-making processes. Our methodology converts trained neural network data from Python into JSON format for real-time visualization, allowing users to intuitively explore neural architecture, activation pathways, and decision routes through a visually rich interface. Preliminary findings affirm the project's potential to improve AI transparency, with future work focused on expanding visualizations to encompass more complex neural networks.**

***Keywords*—Artificial Intelligence, Neural Networks, 3D Visualization, Godot Engine, Explainability, Interactive Tools.**

***Resumo*—Este artigo apresenta um projeto em andamento que visa transformar a natureza de "caixa preta" das redes neurais em modelos de "caixa de vidro" usando ferramentas de visualização 3D. À medida que a inteligência artificial (IA) se torna cada vez mais essencial em várias áreas, cresce a necessidade de explicabilidade nos modelos de deep learning. Utilizamos o Godot Engine, uma plataforma de código aberto, para criar visualizações interativas que elucidam o funcionamento interno das redes neurais, facilitando a compreensão dos processos de tomada de decisão. Nossa metodologia converte dados de redes neurais treinadas do Python para o formato JSON para visualização em tempo real, permitindo que os usuários explorem intuitivamente a arquitetura neural, os caminhos de ativação e as rotas de decisão por meio de uma interface visualmente rica. Resultados preliminares confirmam o potencial do projeto para melhorar a transparência da IA, com trabalho futuro focado em expandir as visualizações para abranger redes neurais mais complexas.**

***Palavras-chave*—Inteligência Artificial, Redes Neurais, Visualização 3D, Godot Engine, Explicabilidade, Ferramentas Interativas.**

I. Introdução

A inteligência artificial (IA) tem revolucionado áreas como ciência de dados e robótica, oferecendo capacidades inéditas de processamento e aprendizado a partir de grandes volumes de dados [1]. Desde os primeiros sistemas de IA, como o "Logic Theorist", que simulava o pensamento humano [2], até as redes neurais profundas que hoje dominam as pesquisas mais avançadas [3], o campo tem evoluído rapidamente. No entanto, a lógica por trás das decisões dessas redes muitas vezes é vista como uma "caixa preta", criando desafios significativos para explicar suas operações internas [4].

As redes neurais profundas, com suas múltiplas camadas de processamento não linear, transformam as entradas em saídas de maneiras complexas e difíceis de rastrear, o que dificulta a compreensão de como certas decisões são tomadas [5]. Essa opacidade é especialmente preocupante em áreas sensíveis, como a saúde [6], onde decisões críticas dependem da precisão e confiabilidade dos sistemas de IA. Desde os primeiros estudos sobre visualização computacional e displays tridimensionais [7], destaca-se a importância de ferramentas que permitem uma interpretação visual intuitiva, aumentando a necessidade de explicar redes neurais complexas de maneira acessível.

Neste contexto, a explicabilidade tem se tornado um foco essencial de pesquisa na IA [8]. Estudos voltados à análise de redes neurais e técnicas de aprendizado profundo [9] buscam demonstrar como essas redes processam dados, facilitando a construção de modelos mentais claros sobre suas decisões [10].

Este artigo propõe a utilização da Godot Engine [11], uma plataforma de código aberto, para desenvolver visualizações interativas que tornem as operações internas de redes neurais mais acessíveis e compreensíveis [12]. A escolha pela Godot, em vez de engines tradicionais como *Unity* ou *Unreal* [13], é motivada pela flexibilidade e facilidade de integração com

JSON, permitindo visualizações dinâmicas e personalizáveis de camadas intermediárias [14].

Nosso objetivo é promover a transparência em sistemas de IA, alinhando-se a pesquisas sobre tecnologias imersivas e 3D, como os primeiros displays de realidade virtual [15]. A proposta inicial inclui uma representação gráfica que ilustra o caminho percorrido pela rede neural até a obtenção de uma saída, permitindo que usuários explorem intuitivamente o fluxo computacional em redes neurais simples. A transparência resultante visa garantir que a IA avance de forma ética e acessível.

II. Metodologia

A metodologia deste estudo envolve o uso da engine open source Godot para criar uma visualização gráfica interativa, permitindo a exploração detalhada do processo de decisão em redes neurais artificiais já treinadas em Python. Primeiramente, extraímos a matriz principal da rede neural, que contém informações essenciais, como as camadas, os pesos, os biases e as funções de ativação. Em seguida, utilizamos um script em Python para converter essas informações em um arquivo JSON estruturado de acordo com um padrão pré-definido. Esse JSON serve como base para configurar a visualização 3D na Godot, facilitando a compreensão da estrutura e do comportamento da rede neural ao disponibilizar dados detalhados sobre cada componente da sua arquitetura.

A ferramenta proposta permite que os usuários inspecionem a topologia da rede e explorem, de forma intuitiva, como os dados fluem através das camadas, como os pesos são aplicados e como as funções de ativação influenciam os resultados. Além disso, a visualização ajuda a traçar toda a trajetória até a tomada de decisão final, esclarecendo o impacto de cada escolha ao longo do processamento e proporcionando uma compreensão mais profunda da dinâmica da rede neural.

A. Sobre Implementação em Godot

Sendo uma engine open source, o Godot Engine permitiu uma integração eficiente com outras ferramentas, como o Python, e facilitou a manipulação do formato JSON, essencial para a visualização das redes neurais. A liberdade de customização permitiu o desenvolvimento da aplicação de acordo com os requisitos do projeto, sem as limitações das plataformas fechadas, tornando o processo mais ágil e acessível. Além disso, a documentação do Godot e o suporte ativo da comunidade em fóruns forneceram soluções valiosas, enquanto as bibliotecas prontas da própria engine aceleraram o desenvolvimento e a adaptação da equipe.

Para a visualização da rede neural, foram desenvolvidos scripts que instanciam dinamicamente cenas representando componentes fundamentais, como neurônios, conexões e camadas de entrada. Essas cenas organizam os neurônios e suas conexões no espaço 3D, proporcionando uma visualização interativa e clara da estrutura da rede. A engine também facilita o carregamento e salvamento de dados em formato JSON, permitindo que as informações da rede neural, como camadas, pesos e ativações, sejam manipuladas em tempo real para análise.

A interação entre neurônios foi aprimorada com o uso de *signals*, que permitem uma comunicação dinâmica entre os elementos, reagindo a eventos como a ativação de neurônios durante o processamento. Isso tornou a simulação mais responsiva e representativa do funcionamento das redes neurais. A aplicação também salva o estado da rede neural em arquivos JSON, registrando dados como a posição dos neurônios e valores de entrada, permitindo que simulações sejam retomadas de forma eficiente.

A escolha do Godot simplificou o desenvolvimento e ampliou as possibilidades de interação e visualização do comportamento das redes neurais. Sua flexibilidade e robustez abrem caminho para futuras expansões no projeto, tornando a plataforma ideal para essa aplicação.

III. Resultados e Discussão

A implementação em Godot obtida traz uma visualização interativa da rede neural que se torna significativamente mais informativa e fácil de usar. O *hover* do mouse sobre neurônios e conexões permite que os usuários obtenham detalhes importantes sobre cada neurônio sem precisar navegar por dados brutos, proporcionando uma experiência mais rica e visual. Além disso, a interface oferece uma maneira organizada e atraente de apresentar os dados, mantendo a clareza e o foco na exploração da estrutura e do comportamento da rede neural.

A. Formato do JSON de Entrada

A Figura 1 ilustra o arquivo JSON gerado após o processamento de uma rede neural MLP[1] pelo script em Python mencionado anteriormente. Esse script extrai as informações necessárias da rede neural a partir da matriz principal da rede e gera um JSON com um formato padronizado, ideal para ser utilizado como entrada no sistema de visualização. Esse formato genérico contém os elementos essenciais para representar a arquitetura e o funcionamento da rede neural, facilitando sua leitura e manipulação na Godot. O JSON é composto pelos seguintes elementos principais:

- **layers**: Um array de objetos, onde cada objeto representa uma camada da rede neural e contém as seguintes informações:

[1]Uma rede do tipo *Multi-Layer Perceptron* com 4 entradas, 3 neurônios na camada oculta e 2 neurônios da camada de saída, ou $4 \times 3 \times 2$.

```json
{
  "layers": [
    {
      "weights": [
        [0.367544247, 0.399710701, 0.562482995, 1.21584223],
        [0.733857775, 1.63110209, -1.62498659, -1.09241287],
        [-0.490230015, 0.277883485, 0.540120978, 0.34298549]
      ],
      "biases": [-0.19923328, 2.29299105, -0.3737735],
      "activation": "ReLU"
    },
    {
      "weights": [
        [-1.44311522, 1.60677038, 0.431788186],
        [-0.302050927, 0.320649974, -0.0542225471]
      ],
      "biases": [0.16481194, 1.43615104],
      "activation": "ReLU"
    }
  ]
}
```

Fig. 1. Exemplo de JSON de entrada gerado pelo *script* desenvolvido referente a uma MLP $4 \times 3 \times 2$ treinada previamente

- **weights**: Matriz de pesos da camada.
- **biases**: Array de biases aplicados aos neurônios da camada.
- **activation**: A função de ativação utilizada na camada (por exemplo, "ReLU").

*Uma rede do tipo *Multi-Layer Perceptron* com 4 entradas, 3 neurônios na camada oculta e 2 neurônios da camada de saída.

B. Representação dos Neurônios e Camadas no Godot

Cada neurônio na rede neural é representado visualmente como uma esfera no Godot, organizadas em camadas conforme o número de camadas especificado no arquivo JSON. O caminho percorrido pela rede é determinado pela ativação dos neurônios a partir de entradas arbitrárias, e cada neurônio ativado ao longo desse processo é destacado visualmente (Figura 2). As esferas que representam os neurônios ativados no caminho de decisão são pintadas de verde, enquanto os neurônios que permanecem inativos são pintados de vermelho, facilitando a compreensão das conexões relevantes na tomada de decisão.

C. Cálculo do Caminho de Decisão

A aplicação em Godot carrega o arquivo JSON contendo os dados de uma rede neural artificial previamente treinada, utilizando essa estrutura para construir uma visualização 3D da rede. Com base em uma entrada arbitrária (Figura 3), o script em *GDScript* processa os pesos, *biases* e funções de ativação fornecidos no JSON, realizando os cálculos necessários para determinar quando um neurônio é ativado. À medida que os neurônios se ativam, o script desenha uma linha conectando cada neurônio ativado ao próximo neurônio correspondente na camada subsequente, criando um caminho visual que ilustra o processo de decisão da IA. A precisão da visualização

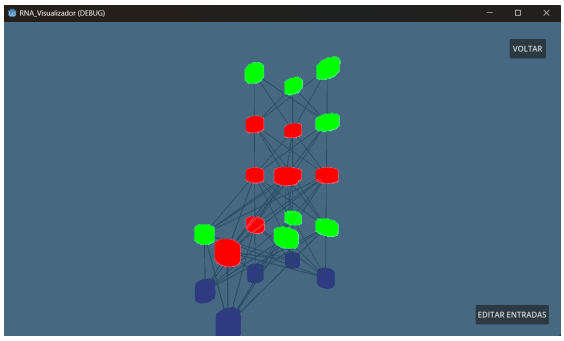

Fig. 2. Rede neural (MLP $5 \times 6 \times 4 \times 3 \times 3$ treinada previamente) e suas ativações com base nas entradas fornecidas

é garantida por meio de testes de unidade e validação dos cálculos teóricos da rede neural, assegurando que as conexões e ativações dos neurônios correspondam ao comportamento esperado e que a saída gerada pela rede seja consistente com resultados previamente conhecidos.

Fig. 3. Mecanismo de edição para os valores de entradas

D. Interação e Visualização de Dados do Neurônio e das Conexões da Rede Neural

Para aprimorar a interatividade e fornecer informações detalhadas sobre cada neurônio na visualização 3D, implementamos uma funcionalidade de *hover* que permite aos usuários visualizar dados relevantes de cada neurônio e das conexões neurais simplesmente passando o mouse sobre eles (Figura 4). Quando o cursor do mouse passa sobre um neurônio, ele muda de cor para amarelo, destacando-o na interface. Essa funcionalidade foi implementada utilizando o *MeshInstance3D* em conjunto com *CollisionShape3D* para detectar colisões, além de um nó *Control* para exibir as informações de forma clara e dinâmica.

O painel de informações é dinamicamente atualizado com os dados do neurônio sendo apontado. A estrutura JSON carregada previamente no projeto serve como a fonte para

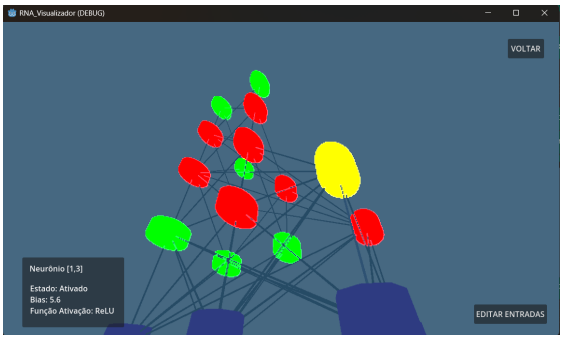

Fig. 4. Menu suspenso de informações relevantes do neurônio selecionado

esses valores, e cada neurônio possui seu próprio conjunto de dados armazenado para rápida recuperação quando o *hover* é ativado. O painel de informações é ocultado assim que o mouse sai da área de colisão do neurônio, evitando o acúmulo de informações na tela.

IV. Conclusão

A implementação da visualização 3D para redes neurais artificiais apresentada continua em desenvolvimento, com o objetivo de oferecer uma ferramenta interativa e acessível para a exploração dos processos de decisão dessas redes. Embora ainda enfrentemos alguns desafios técnicos, especialmente na área de ferramentas e nas decisões relativas à *UI/UX* para proporcionar uma melhor experiência ao usuário, os resultados preliminares demonstram a viabilidade da abordagem proposta e a precisão da lógica subjacente.

Comprometidos com os princípios de transparência e colaboração, planejamos disponibilizar a aplicação como código aberto. Acreditamos que, ao compartilhar nosso trabalho com a comunidade, não apenas incentivamos a contribuição e a inovação, mas também promovemos uma maior compreensão sobre o funcionamento das redes neurais. A participação da comunidade será essencial para aprimorar a ferramenta, permitindo que desenvolvedores e pesquisadores explorem novas possibilidades e ampliem os fundamentos que estamos estabelecendo. Com esse espírito colaborativo, seguimos em frente, empenhados em tornar a visualização de redes neurais mais acessível e compreensível para todos.

Além disso, pretendemos prosseguir com trabalhos futuros que incluam visualizações de redes neurais mais complexas, com ênfase nas arquiteturas utilizadas em *Deep Learning*. Ao abordar essas redes mais sofisticadas, esperamos enriquecer ainda mais a experiência de visualização e compreensão dessas redes para os potenciais usuários, contribuindo para o avanço do conhecimento em uma área em constante evolução.

Agradecimentos

Este projeto de iniciação científica contou com o apoio financeiro do CNPq e da Universidade Federal de Uberlândia, fundamentais para sua realização. Agradecemos aos professores, colegas e colaboradores pelas contribuições essenciais, que enriqueceram nossa pesquisa e aprimoraram a visualização das redes neurais. Também reconhecemos o uso de ferramentas de IA, especialmente o *ChatGPT*, que auxiliou na organização e redação de partes do texto deste artigo, demonstrando o potencial da IA como uma aliada no processo de pesquisa e produção acadêmica.

Referências

[1] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," 2017. [Online]. Available: https://arxiv.org/abs/1708.08296

[2] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, 2017, https://distill.pub/2017/feature-visualization.

[3] R. Salama and M. Elsayed, "A live comparison between unity and unreal game engines," *Global Journal of Information Technology: Emerging Technologies*, vol. 11, no. 1, pp. 01–07, 2021. [Online]. Available: https://doi.org/10.18844/gjit.v11i1.5288

[4] I. E. Sutherland, "A head-mounted three dimensional display," in *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, ser. AFIPS '68 (Fall, part I). New York, NY, USA: Association for Computing Machinery, 1968, p. 757–764. [Online]. Available: https://doi.org/10.1145/1476589.1476686

[5] T. Munzner, *Visualization Analysis and Design*, 1st ed. A K Peters/CRC Press, 2014. [Online]. Available: https://doi.org/10.1201/b17511

[6] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017. [Online]. Available: https://arxiv.org/abs/1702.08608

[7] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144. [Online]. Available: https://doi.org/10.1145/2939672.2939778

[8] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[9] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau, "Visual analytics in deep learning: An interrogative survey for the next frontiers," 2018. [Online]. Available: https://arxiv.org/abs/1801.06889

[10] D. Smilkov, S. Carter, D. Sculley, F. B. Viégas, and M. Wattenberg, "Direct-manipulation visualization of deep networks," 2017. [Online]. Available: https://arxiv.org/abs/1708.03788

[11] Godot Engine, "Godot Engine - Free and Open Source Game Engine," 2024, acesso em: 22 set. 2024. [Online]. Available: https://godotengine.org/

[12] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," 2015. [Online]. Available: https://arxiv.org/abs/1506.06579

[13] Q. Zhang and S.-C. Zhu, "Visual interpretability for deep learning: a survey," 2018. [Online]. Available: https://arxiv.org/abs/1802.00614

[14] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. Chau, "Activis: Visual exploration of industry-scale deep neural network models," 2017. [Online]. Available: https://arxiv.org/abs/1704.01942

[15] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," 2013. [Online]. Available: https://arxiv.org/abs/1311.2901