Comparações entre a rede Kolmogorov-Arnold e a rede perceptron multicamada

Pedro H. S. Soares, Renato Candido e Magno T. M. Silva

***Resumo*— Neste trabalho, as clássicas redes perceptron multicamada (MLP) são comparadas com as recém-propostas redes Kolmogorov-Arnold (KAN) em um problema de regressão e outro de classificação binária. A KAN, baseada em *splines* ajustáveis, apresenta uma maior interpretabilidade às custas de uma complexidade computacional elevada e uma convergência mais lenta.**

***Palavras-Chave*— aprendizado de máquina, redes neurais, teorema de Kolmogorov-Arnold, *splines*, interpretabilidade.**

***Abstract*— In this work, classical multilayer perceptron (MLP) networks are compared with the newly proposed Kolmogorov-Arnold networks (KAN) in a regression problem and a binary classification problem. KAN, based on adjustable splines, offers greater interpretability at the cost of higher computational expense and slower convergence.**

***Keywords*— machine learning, neural networks, Kolmogorov-Arnold theorem, splines, interpretability.**

I. Introdução

Nos últimos anos, o aprendizado de máquina vem ganhando destaque no desenvolvimento tecnológico, influenciando como hardwares e softwares são projetados e mudando radicalmente a interação homem-máquina [1]–[3]. Em aprendizado profundo, a rede perceptron multicamada (*multilayer perceptron* – MLP) é protagonista na solução de problemas não lineares, respaldada pelo teorema da aproximação universal [1]. No entanto, a profundidade dessas redes dificulta o entendimento de como o problema é resolvido [1]–[3]. Em outras palavras, as redes MLP conseguem de fato resolver problemas complexos, mas as soluções em si são uma "caixa-preta", dificultando ou mesmo impedindo sua interpretabilidade.

A fim de possibilitar a interpretabilidade dos modelos, foi proposta recentemente a rede Kolmogorov-Arnold (*Kolmogorov-Arnold network* – KAN), inspirada pelo teorema de representação homônimo [4]. Enquanto as redes MLP possuem funções de ativação fixas nos neurônios, as KANs consideram as funções de ativação como parâmetros ajustáveis. Os pesos são substituídos por funções univariadas parametrizadas como *splines*. Essa mudança faz com que as KANs possam superar as MLPs em termos de precisão e interpretabilidade em alguns casos [4].

Neste trabalho, a KAN é comparada com a MLP em um problema de regressão e em outro de classificação binária. O artigo está organizado da seguinte forma. Na Seção II, a KAN é revisitada. Na Seção III, são mostrados os resultados da comparação com a MLP e na Seção IV, as conclusões.

Pedro H. S. Soares, Renato Candido e Magno T. M. Silva, Depto. de Engenharia de Sistemas Eletrônicos, Escola Politécnica, Universidade de São Paulo, São Paulo, SP, Brasil, emails: soares.pedro@usp.br; renatocan@lps.usp.br; magno.silva@usp.br. Este trabalho foi financiado pela CAPES (código de financiamento 001) e pelo CNPq (122941/2024-1, 303826/2022-3 e 404081/2023-1).

II. Revisitando a KAN

O teorema de representação de Kolmogorov-Arnold estabelece que dada uma função contínua multivariada $f : [0,1]^n \to \mathbb{R}$, então existe um conjunto de funções contínuas univariadas $\{\phi_{q,p}\}$ e $\{\Phi_q\}$ tais que f pode ser representada como [4], [5]

$$f(x_1, x_2, \ldots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right). \quad (1)$$

A motivação desse teorema é utilizar funções univariadas para reduzir a complexidade no cálculo de funções multivariadas.

A KAN se baseia nesse teorema, considerando funções *B-splines*. Essas funções, definidas em intervalos específicos denominados intervalos de nó, são construídas a partir da combinação linear de polinômios de grau baixo e utilizadas para aproximar curvas contínuas de maneira suave [6]. Assim, a combinação linear com coeficientes c_i das funções de base $B_{i,k}$ levam à *B-spline* de grau k, ou seja,

$$\phi_{q,p}(x_p) \approx \sum_{i=0}^{m} c_i B_{i,k}(x_p), \quad (2)$$

em que as funções de base

$$B_{i,k}(x_p) = \frac{x_p - t_i}{t_{i+k} - t_i} B_{i,k-1}(x_p) + \frac{t_{i+k+1} - x_p}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x_p),$$

são definidas com a fórmula de Cox-deBoor, t_i é o i-ésimo nó e $B_{\ell,0} = 1$ para $t_\ell \leq x \leq t_{\ell+1}$ e $B_{\ell,0} = 0$, caso contrário, $\ell = 0, \cdots, m$. Para $\ell > m$, assume-se $B_{\ell,k}(x_p) = 0$ [4], [6].

Na KAN, cada função de ativação $\phi_{q,p}(x_p)$ é aproximada por uma *B-spline* como em (2) e portanto, representada por um conjunto de $m + 1$ coeficientes c_i. No ajuste desses coeficientes, empregam-se algoritmos de treinamento similares aos da MLP, como o algoritmo *backpropagation* [1]. É importante garantir que as funções base sejam iguais para todas funções de ativação. Para isso, é necessário definir o número de intervalos de nós, denominado *grid*, e o grau das funções base como hiperparâmetros. Também é necessário verificar durante o treinamento o domínio das *B-splines* e atualizar os limites dos nós para que comporte os dados de treinamento [4].

A KAN se assemelha à MLP, pois ambas possuem uma estrutura de camadas conectadas, em que todos os neurônios de uma camada estão ligados aos neurônios das camadas subsequentes. A principal diferença está na forma como a ativação de cada neurônio é calculada: na MLP, a entrada é ponderada por pesos ajustáveis e a saída passa por uma função de ativação fixa, enquanto na KAN, cada componente da entrada é processado por uma função de ativação ajustável baseada em *B-splines*, e a saída é obtida por uma soma simples dessas transformações, conforme ilustrado na Figura 1.

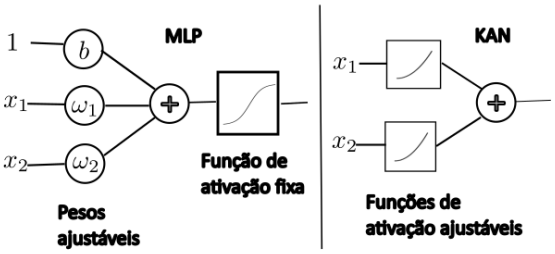

Fig. 1. Comparação entre as ativações de uma MLP e de uma KAN.

III. Resultados

Considerou-se inicialmente um problema de regressão com o objetivo de aproximar a função $f(x,y) = \sin^2(x+y)$, utilizando 4000 pontos (x,y), gerados aleatoriamente no domínio $[-\pi,\pi]\times[-\pi,\pi]$. A KAN e a MLP foram configuradas com três camadas, contendo 3, 2, 1 e 24, 25, 1 neurônios, respectivamente. Ambas foram treinadas por 3000 épocas com o otimizador Adam ($\beta_1 = 0.9$ e $\beta_2 = 0.999$) [7]. Na KAN, consideraram-se ainda $k = 3$ e o valor do *grid* iniciando com 3 e dobrando a cada 1000 épocas. Esse aumento, conhecido como extensão de *grid* [4], resultou no aumento do número de parâmetros treináveis a cada 1000 épocas. Na MLP, foram usadas ReLU nos neurônios das camadas ocultas e tangente hiperbólica para o neurônio de saída. Essas configurações permitem que o números de parâmetros treináveis de ambas as redes estejam próximos. Para comparação, também considerou-se um modelo da KAN com grid fixo igual a 12.

Em termos de custo computacional, na KAN com *grid* variável foram utilizados 462 parâmetros ao final de 130 s de treinamento, enquanto na MLP, foram treinados 463 parâmetros em 134 s. Apesar do mesmo custo neste caso, a raiz do erro quadrático médio (*root mean-square error* – RMSE) atingida pela KAN é de 0,0529 e o coeficiente de determinação (R^2) é de 0,9776, enquanto para a MLP essas métricas foram de 0,0228 e 0,9958, respectivamente. Na Figura 2-(a), é possível observar que a KAN apresentou uma convergência mais lenta que a MLP. Apesar do número de parâmetros finais da KAN com *grid* variável ser o mesmo da KAN com *grid* fixo, a extensão de *grid* ao longo do treinamento é essencial para se obter um melhor desempenho. No entanto, isso causa um aumento da função custo (pico da curva verde em torno da época 2000), mas que logo volta ao patamar de antes da alteração do *grid*. Esses resultados sugerem que a KAN apresenta resultados promissores em termos de desempenho para um custo computacional semelhante ao da MLP, embora a MLP ainda consiga um desempenho um pouco superior.

No problema de classificação binária, considerou-se o *German Credit Dataset* [8], que contém 1000 instâncias compostas por 20 variáveis. A variável-alvo assume valor $d = 1$ para clientes confiáveis e $d = -1$ para aqueles considerados de risco. Para se obter o mesmo custo computacional, a KAN e a MLP foram configuradas com 2 e 3 camadas, contendo $2,1$ e $20,13,1$ neurônios, respectivamente. Consideraram-se para a KAN $k = 3$ e *grid* = 5, enquanto para a MLP, ReLU nos neurônios das camadas ocultas e tangente hiperbólica para o de saída. Ambas foram treinadas com otimizador Adam ($\beta_1 = 0.9$ e $\beta_2 = 0.999$) ao longo de 1000 épocas.

A KAN atingiu uma acurácia de 68% e a MLP de 77%. Novamente, a MLP convergiu mais rápido que a KAN, como pode ser observado na Figura 2-(b). Em termos de custo computacional, a MLP treinou 707 parâmetros por 17,3 s, enquanto a KAN treinou 714 parâmetros por 33,2 s. Apesar do desempenho inferior que o da MLP, simplificando a KAN com a técnica de poda ($\theta = 0{,}1$) como em [4] e utilizando o método dos mínimos quadrados não linear [9], foi possível extrair uma expressão de saída da KAN. As funções de ativação treinadas foram ajustadas considerando três funções candidatas: afim, cosseno e tangente hiperbólica. Para selecionar o melhor ajuste de cada função de ativação, foi utilizado o maior valor de coeficiente de determinação. Ao final do processo, a composição das funções resultou em

$$\begin{aligned}\hat{d} = &-0{,}025\cos\left(-0{,}032x_2 + 1{,}842\cdot 10^{-4}x_5 + 6{,}374\right)\\ &+ 0{,}210\tanh(2{,}151x_{13} + 1{,}235x_2 + 2{,}198x_4\\ &- 0{,}009x_5 + 45{,}244) - 0{,}116.\end{aligned}$$

Essa expressão é relativamente compacta e leva à mesma acurácia da KAN com os dados de teste. Ela possibilita verificar a relevância de cada componente da entrada (x_k, $k = 1,\cdots,20$) na decisão de crédito, algo que não se consegue facilmente com a MLP. Isso faz da KAN um modelo promissor quando se deseja buscar interpretabilidade.

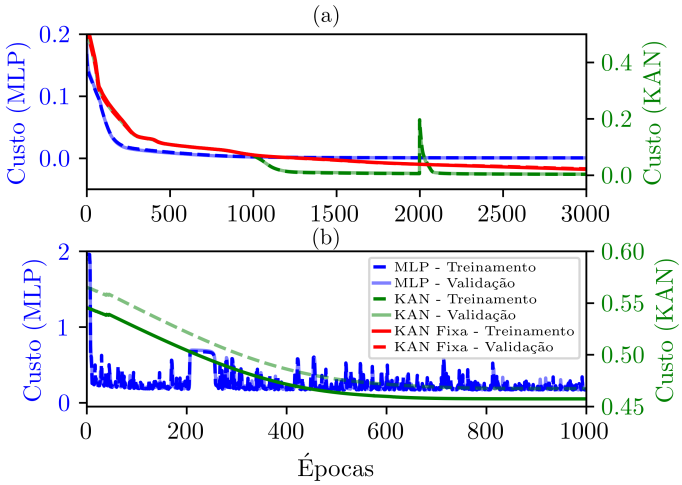

Fig. 2. Evolução do custo durante o treinamento para os problemas de (a) regressão e (b) classificação binária.

IV. Conclusões

Quando se busca interpretabilidade, a KAN é uma boa alternativa à rede MLP, apesar da convergência mais lenta no treinamento e do desempenho inferior em alguns casos.

Referências

[1] S. Haykin, *Neural networks and learning machines*, Prentice Hall, 2009.
[2] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2016.
[3] C. M. Bishop, *Deep Learning*, Springer, 2024.
[4] Z. Liu *et al.*, "KAN: Kolmogorov-Arnold networks," *arXiv*, https://arxiv.org/abs/2404.19756, 2024.
[5] A. N. Kolmogorov, "On the representation of continuous functions of several variables as superpositions of continuous functions of a smaller number of variables," *Dokl. Akad. Nauk*, 1956.
[6] L. Schumaker, *Spline Functions: Computational Methods*, SIAM, 2015.
[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv*, https://arxiv.org/abs/1412.6980, 2014.
[8] H. Hofmann, "Statlog (German Credit Data)," UCI Machine Learning Repository, 1994, DOI: https://doi.org/10.24432/C5NC77.
[9] D. M. Bates and D. G. Watts, *Nonlinear Regression Analysis and Its Applications*, John Wiley & Sons, 1988.