Towards solving Kolmogorov-Arnold Theorem using Variational Optimization

Francesco Alesiani
NEC Laboratories Europe
francesco.alesiani@neclab.eu

Federico Errica
NEC Laboratories Europe
federico.errica@neclab.eu

Henrik Christiansen
NEC Laboratories Europe
henrik.christiansen@neclab.eu

***Abstract*—Kolmogorov Arnold Networks (KANs) are an emerging architecture for building machine learning models. KANs are based on the theoretical foundation of the Kolmogorov-Arnold Theorem and its expansions, which provide an exact representation of a multi-variate continuous bounded function as the composition of a limited number of uni-variate continuous functions. While such theoretical results are powerful, its use as a representation learning alternative to multi-layer perceptron (MLP) hinges on the choice of the number of bases modeling each of the univariate functions. In this work, we show how to address this problem by adaptively learning a potentially infinite number of bases for each univariate function during training. We do so by means of a variational inference optimization problem. Our proposal, called INFINITYKAN, extends the potential applicability of KANs by treating an important hyper-parameter as part of the learning process.**

***Index Terms*—Kolmogorov-Arnold Theorem, KAN, MLP, Machine Learning, Variational Optimization**

I. Introduction

Kolmogorov-Arnold Networks (KANs) [1] have recently gained attention in the machine learning community as a potential alternative to the widely-used Multi-Layer Perceptrons (MLPs) [2]. MLPs have been instrumental in transforming machine learning due to their ability to approximate any continuous function, a capability supported by the universal approximation theorem [2]. The Kolmogorov-Arnold Theorem (KAT), originally developed to address Hilbert's 13th problem, is a fundamental mathematical result with numerous implications [3]. While the universal approximation theorem suggests that any continuous function can be approximated using an MLP of bounded width, KAT represents any multivariate function exactly using a finite and known number of univariate functions. KAT's influence extends beyond pure mathematics, finding applications in diverse fields such as fuzzy logic, pattern recognition, and neural networks [4]–[8]. This versatility has contributed to its growing importance in the machine-learning community. KAT-based results have been applied in several ways, including the development of machine learning models, called Kolmogorov-Arnold Networks (KANs) that stand as a potential alternatives to MLPs in solving arbitrary tasks [9], [10].

However, while the KAT argues for the existence of a univariate functions that represent the target function exactly, the choice of the basis functions that model each univariate function remains an open problem. It is of no surprise that KANs' effectiveness in addressing complex, high-dimensional problems heavily relies on the choice, construction, and training of appropriate basis functions.

Various proposals for the basis functions have been made, such as orthogonal polynomials, spline, sinusoidal, wavelets, or adaptive basis selection methods, which may depend on the specific problem at hand [11]–[14]. Not only does the choice of family for basis functions remain a problem, but also the number of basis function to use is not known in advance, an a wrong selection of this number can greatly affect the representational ability of KANs for a given problem.

We therefore present INFINITYKAN, which models the univariate functions using an adaptive and potentially infinite number of bases. INFINITYKAN handles the unbounded number of bases by means of a truncated window function, in a way that provides gradient information for the window to be updated. The model's design stems from a variational treatment of the learning problem, which is in line with the topic of the workshop.

Summarizing, our contributions are: *i)* a variational treatment of the learning problem (Section III) that tractably models an unbounded number of basis for the univariate functions; *ii)* an experimental validation (Section IV) of the performance of the proposed variational approach on common regression and classification tasks.

II. Related Works

Recent research [15] has expanded on KAT foundations, exploring the capabilities of KAN-based models in high-dimensional spaces and their potential to mitigate the curse of dimensionality [16]. Various KAN architectures have been proposed: KAN has been combined with Convolutional Neural Networks (CNNs) [17], or with transformer models [18], leading to improved efficiency in sequence modeling tasks. Furthermore, EKAN incorporates matrix group equivariance [19] into KANs, while GKSN [20] explores the extension to invariant and equivariant functions to model physical and geometrical symmetries.

KANs have demonstrated their versatility across a wide spectrum of machine learning applications [21], particularly in scenarios demanding efficient (i.e. small number of parameters) function approximation with a limited parameter budget. Their effectiveness in high-dimensional regression problems, where traditional neural networks often face scalability issues, was notably demonstrated by Kůrková in 1991 [22].

Adaptive architectures have been proposed for MLP models. For example, [23] extends the network with an additional hidden units as the end of a training phase, while firefly network descent [24] grows the width and depth of a neural network during training. In continual learning [25], network models are updated based on new tasks, or neurons are duplicated or removed according to heuristics to create more capacity [26], [27]. The unbounded depth network of [28], recently applied to graphs [29], and adaptive width neural network [30] also use a variational approach to learn the number of layers of a residual neural network or the number of neurons in a neural network, but these approaches are not directly applicable, since the output is not additive in KAN models.

III. Infinity Kolmogorov-Arnold Network

We first recap the definition of a KAN layer before introducing our extension to learn an unbounded number of basis functions.

A. KAN Layer and basis functions

While the KAT theorem provides a way to represent a generic continuous multivariate function $f(x_1, \dots, x_n)$ as

$$f(x_1, \dots, x_d) = \sum_{q=1}^{2d+1} \psi_q \left(\sum_{p=1}^{d} \phi_{qp}(x_p) \right)$$

with $x_p \in [0,1]$ and ϕ_{qp}, ψ_q continuous univariate functions, we consider the KAN composed of L layers (see Figure 1), where each layer $\ell \in \{1, \dots, L\}$ implements the mapping from $[0,1]^{d_{\ell-1}} \to [0,1]^{d_\ell}$ using the univariate functions $\{\phi^\ell_{qp}\}$, which are used to compute the hidden variables $\boldsymbol{x}^\ell = \boldsymbol{\phi}^\ell(\boldsymbol{x}^{\ell-1}) = \{x^\ell_q \mid x^\ell_q = h^\ell_q(x^{\ell-1}_1, \dots, x^{\ell-1}_{d_{\ell-1}}) = \sum_{p=1}^{d_{\ell-1}} \phi^\ell_{qp}(x^{\ell-1}_p)\}, \forall q \in [d_\ell]$, from previous layer variables $\boldsymbol{x}^{\ell-1} = \{x^{\ell-1}_p\}, \forall p \in [d_{\ell-1}]$. The KAT does not tell us how to find the univariate functions, but it is possible to build a convergent series for any uniformly continuous function $\phi(x)$ as a linear combination of either step or ReLU functions: $\phi(x) = \lim_{n\to\infty} \phi^n(x)$ with $\phi^n(x) = \sum_{k=1}^n \phi^n_k(x) = \sum_{k=1}^n \theta^n_k \varphi^n_k(x)$ where $\varphi^n_k(x)$ can either be a step function parametrized by δ_k or a ReLU function. Therefore, in the following, we refer to $\varphi^n_k(x)$ as the generative functions of the basis $\phi^n_k(x)$. Therefore, w.l.o.g. we represent each univariate function in a KAN layer ℓ as the limit of the linear combination of the basis functions $\varphi^n_k(x)$

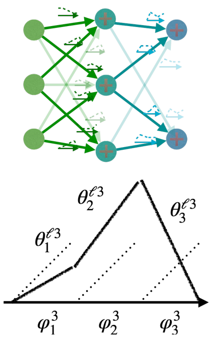

Fig. 1: (Upper) KAN composed of two layers; (Bottom) the basis functions $\varphi^n_k(x)$ (ReLU) used to build $\phi^{\ell n}_{qp}(x)$.

$$\phi^\ell_{qp}(x) = \lim_{n\to\infty} \sum_{k=1}^{n} \theta^{\ell n}_{qpk} \varphi^n_k(x) \tag{1}$$

Given this model, we can pick a **finite** n and can train the parameters $\{\theta^{\ell n}_{qpk}\}_{k\in[n]}$, where $[n] = \{1, \dots, n\}$, for each layer ℓ, using, for example, back-propagation.

B. Variational training objective

We consider a regression or a classification problem and the corresponding dataset $\mathcal{D}$ composed of i.i.d. samples $(\boldsymbol{X}, \boldsymbol{Y}) = \{(x_i, y_i)\}_{i=1}^D$, with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}^{d'}$. If we build a probabilistic model implementing the distribution $p(\boldsymbol{Y}|\boldsymbol{X})$ the objective is to maximize the dataset log-likelihood

$$\mathcal{L}\{\mathcal{D}\} = \ln p(\boldsymbol{Y}|\boldsymbol{X}) \tag{2}$$

If we modeled the probability distribution with a multi-layer KAN network, we would need to optimize Equation (2) with respect to the functions ϕ^ℓ_{qp}. However, based on Equation (1), we first introduce an infinite-dimensional family of KANs. To this end, we introduce two latent variables that parameterize such a family. Each layer has a set of parameters $\boldsymbol{\theta}^\ell = \{\theta^{\ell n}_{qpk}, k \in [n]\}_{n=1}^\infty$ (see Equation (1)), with $\theta^{\ell n}_{qpk}$ is a multivariate variable over the learnable weights of the k-th basis function at layer ℓ and for the qp univariate function.

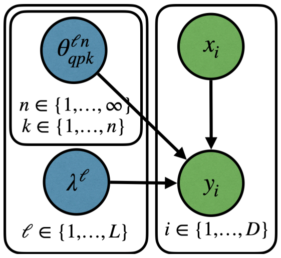

Fig. 2: The graphical model of InfinityKAN, with the observable variables (in green) x_i, y_i and latent variables (in blue) $\theta^{\ell n}_{qpk}, \lambda^\ell$.

We further introduce a latent variable λ^ℓ that defines the number of basis functions n used at layer ℓ. As we sample $n \sim p(n|\lambda^\ell)p(\lambda^\ell)$, in effect we are defining a finite learning objective and we can perform inference. For a KAN of L layers, we define $\boldsymbol{\theta} = \{\boldsymbol{\theta}^\ell\}_{\ell\in[L]}$ and $\boldsymbol{\lambda} = \{\lambda^\ell\}_{\ell\in[L]}$ and we assume independence across all layers, which allows us to write $p(\boldsymbol{Y}|\boldsymbol{X}) = \int d\boldsymbol{\theta} d\boldsymbol{\lambda} p(\boldsymbol{Y}, \boldsymbol{\theta}, \boldsymbol{\lambda}|\boldsymbol{X})$. Similar to [28]), we now assume that $\boldsymbol{\theta}, \boldsymbol{\lambda}$ are indepenendent, i.e. $p(\boldsymbol{\theta}, \boldsymbol{\lambda}) = p(\boldsymbol{\theta})p(\boldsymbol{\lambda})$ and, based on the graphical model of Figure 2, we write the following distributions

$$p(\boldsymbol{Y}, \boldsymbol{\theta}, \boldsymbol{\lambda}|\boldsymbol{X}) = p(\boldsymbol{Y}|\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{X})p(\boldsymbol{\theta})p(\boldsymbol{\lambda}) \tag{3}$$

$$p(\boldsymbol{\lambda}) = \prod_{\ell=1}^{L} p(\lambda_\ell) = \prod_{\ell=1}^{L} \mathcal{P}(\lambda_\ell; \eta_\ell) \tag{4}$$

$$p(\boldsymbol{\theta}) = \prod_{\substack{\ell\in[L],\\ n=1,\dots,\infty,k\in[n],\\ q\in[d_\ell],p\in[d_{\ell-1}]}} p(\theta^{\ell n}_{qpk}) \tag{5}$$

$$p(\theta^{\ell n}_{qpk}) = \mathcal{N}(\theta^{\ell n}_{qpk}; \boldsymbol{0}, \operatorname{diag}(\sigma^\ell)) \tag{6}$$

with $\mathcal{P}(\lambda; \eta)$ the Poisson distribution, while $\mathcal{N}(\theta; \mu, \sigma)$ is the Gaussian distribution. The predictive model $p(\boldsymbol{Y}|\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{X})$ is based on the KAN architecture and is described later. The distributions depend on the prior's hyper-parameters $\boldsymbol{\eta} = \{\eta_\ell\}$ and $\boldsymbol{\sigma} = \{\sigma_\ell\}$, while the KAN is parametrized by $\boldsymbol{\theta}$, and $\boldsymbol{\lambda}$. Maximizing directly Equation (2) would require computing an intractable integral, therefore we apply the mean-field variational inference approach [31], which entails maximizing the expected lower bound (ELBO), by introducing a learnable variational distribution $q(\boldsymbol{\theta}, \boldsymbol{\lambda})$ and using the concavity logarithmic function, write the objective as

$$\ln p(\boldsymbol{Y}|\boldsymbol{X}) \geq \mathbb{E}_{q(\boldsymbol{\lambda},\boldsymbol{\theta})}\left[\ln \frac{p(\boldsymbol{Y}, \boldsymbol{\lambda}, \boldsymbol{\theta}|\boldsymbol{X})}{q(\boldsymbol{\lambda}, \boldsymbol{\theta})}\right] \tag{7}$$

Using the same intuition from [28], we then assume that the variational distribution can be written as

$$q(\boldsymbol{\theta}, \boldsymbol{\lambda}) = q(\boldsymbol{\theta}|\boldsymbol{\lambda})q(\boldsymbol{\lambda}) \tag{8}$$

$$q(\boldsymbol{\lambda}) = \prod_{\ell=1}^{L} q(\lambda_\ell) = \prod_{\ell=1}^{L} \mathcal{P}(\lambda_\ell; \bar{\lambda}_\ell) \tag{9}$$

$$q(\boldsymbol{\theta}|\boldsymbol{\lambda}) = \prod_{\substack{\ell\in[L],\\ n=K_\ell,\\ k\in[K_\ell],\\ q\in[d_\ell],p\in[d_{\ell-1}]}} q(\theta^{\ell n}_{qpk}) \prod_{\substack{\ell\in[L],\\ n=1,\dots,\infty,n\neq K_\ell\\ k\in[n]\\ q\in[d_\ell],p\in[d_{\ell-1}]}} p(\theta^{\ell n}_{qpk}) \tag{10}$$

$$q(\theta^{\ell n}_{qpk}) = \mathcal{N}(\theta^{\ell n}_{qpk}; \bar{\theta}^{\ell n}_{qpk}, \boldsymbol{I}), \tag{11}$$

with $K_\ell = 2\lambda_\ell + 1$.

By modeling the distribution of the parameters belonging to a different function in the infinite series with the same a priori distribution p, its influence on the maximization problem is removed. While we could model the variance of the basis's coefficients with additional trainable parameters, in the following, we see how the variance is ignored. We have selected K_ℓ to be even, to simplify the construction of a symmetric basis. We can now write the final objective by using the previous assumptions and the first-order approximation of the expectation, i.e. $\mathbb{E}_{q(\boldsymbol{\lambda};\bar{\boldsymbol{\lambda}})}[f(\boldsymbol{\lambda})] = f(\bar{\boldsymbol{\lambda}})$, and $\mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\lambda};\bar{\boldsymbol{\theta}})}[f(\boldsymbol{\theta})] = f(\bar{\boldsymbol{\theta}})$, in Equation (7),

$$\sum_{i=1}^{D} \ln p(y_i|\boldsymbol{\lambda} = \bar{\boldsymbol{\lambda}}, \boldsymbol{\theta} = \bar{\boldsymbol{\theta}}, x_i) + \sum_{\ell=1}^{L} \ln \frac{p(\bar{\lambda}_\ell; \eta_\ell)}{q(\bar{\lambda}_\ell; \bar{\lambda}_\ell)} + \sum_{\substack{\ell\in[L],\\ k\in[K_\ell],\\ q\in[d_\ell],p\in[d_{\ell-1}]}} \ln p(\bar{\theta}^{\ell K_\ell}_{qpk}; \boldsymbol{0}, \operatorname{diag}(\sigma^\ell)), \tag{12}$$

where we remove the constant term arising from the evaluation of q distribution at its mean value, i.e. $q(\bar{\theta}^{\ell K_\ell}_{qpk}) = \mathcal{N}(\bar{\theta}^{\ell K_\ell}_{qpk}; \bar{\theta}^{\ell K_\ell}_{qpk}, \boldsymbol{I}) = \text{const}$ and σ_ℓ, η_ℓ are the prior' parameters. Equipped with Equation (12), we can now train the basis parameters $\bar{\boldsymbol{\theta}}$ and the bases' sizes $\bar{\boldsymbol{\lambda}}$ using standard stochastic gradient descent algorithms.

C. Symmetric basis

We now introduce the KAN-based model that implements the prediction model $p(\boldsymbol{Y}|\boldsymbol{\lambda} = \bar{\boldsymbol{\lambda}}, \boldsymbol{\theta} = \bar{\boldsymbol{\theta}}, \boldsymbol{X})$, given the data samples $\boldsymbol{X}$ and the variational parameters $\{\boldsymbol{\lambda}, \boldsymbol{\theta}\}$. When sampling the number of basis functions, $\lambda_\ell \sim q(\lambda_\ell)$, we need to train a different set of parameters $\theta^{\ell K_\ell}_{qpk}$ of Equation (11), where we dropped the bar from the variable to ease the notation. By changing the number of basis functions, we also change their locations. This makes the training difficult. Further, to estimate the impact of the change in the number of bases on the loss, we need a continuous relationship between their size $\boldsymbol{\lambda}$ and the weights of the bases $\boldsymbol{\theta}$. We therefore introduce an additional weighting function $\boldsymbol{w} = \{w_k^{K_\ell(\lambda_\ell)}\}_{\ell=1}^{L}$ parametrized by $\boldsymbol{\lambda}$ that multiplies the basis weights $\boldsymbol{\theta}$, and write the KAN Layer as

$$h^\ell_q(x^{\ell-1}_1, \dots, x^{\ell-1}_{d_{\ell-1}}) = \sum_{\substack{k\in[K_\ell],\\ p\in[d_{\ell-1}]}} \theta^{\ell K_\ell}_{qpk} w^{K_\ell}_k \varphi^{K_\ell}_k(x^{\ell-1}_p) \tag{13}$$

with $w^{K_\ell}_k$ mimic a symmetric distribution over the basis functions over the interval $[-1, 1]$. As a symmetric positive function, we select

$$w_\lambda(x) = \left(1 + e^{-2\lambda+2|x|}\right)^{-1} \tag{14}$$

evaluated for $x_k = -\lambda + 2(k-1), k = 1, \dots, 2\lambda + 1$, so that $w^{K_\ell}_k = w_{(K_\ell-1)/2}(x_k)$.

D. Interpolation of the weights

Whenever a new λ_ℓ is sampled, the number of bases could change. When the number of bases changes from n to n', we use simple linear interpolations of the weights $w^{\ell,n'}_k = \mathcal{I}[w^{\ell,n}_k]$, where $w^{\ell,n'}_k = \mathcal{I}[w^{\ell,n}_j] = (1 - k\frac{n}{n'} + j)w^{\ell,n}_j + (k\frac{n}{n'} - j)w^{\ell,n}_{j+1}$, where $j = \arg\max_j\{\frac{j}{n} \leq \frac{k}{n'}\}$,

IV. Experimental validation

InfinityKAN overcomes the limitation of selecting the number of basis functions for each of the layers of a KAN, we therefore would like to validate if 1) the training procedure is stable and 2) if the performances are at least competitive with the a KAN with fixed number of bases. We focus on regression and classification tasks. We selected 6 datasets, two synthetic regression tasks on the class of the spiral dataset (with $k = 2, 3$ the number of spirals), and four image classification tasks: MNIST [32], CIFAR10, CIFAR100 [33], and the RGB version of the EUROSAT [34]. We use the datasets' standard split, except for the last, where we randomly split 80/10/10 training, validation and test, and we apply normalization on pixel values. We compare the standard KAN with AdamW [35], with weight decay of 10^{-5}, learning rate of 10^{-2}, and a reduce-on-plateau scheduler. The KAN generative basis function is Relu [20], with a Batch Normalization 1d layer [36] to center the input distribution. We use 3 layer KAN with the initial number of hidden units equal to 16 and the number of bases equal to 8. The MLP has also 3 layers of 128 neurons each. We train and test for $5'000$ epochs. All models have a three-layer structure and were selected on the spiral dataset to have a similar number of parameters. We report the classification test accuracy associated with the best validation

TABLE I: We compare the accuracy of KAN with a fixed number of bases, an MLP, and INFINITYKAN on the classification tasks: CIFAR10, CIFAR100, MNIST, and EUROSAT. The number of bases per layer (L0,L1,L2) is reported in the last column.

Model	INFINITYKAN	KAN	MLP	L0	L1	L2
MNIST	96.97	96.23	**97.87**	5.3	10.3	17.0
	(std) 0.09	0.12	0.04	0.6	0.6	1.0
CIFAR10	49.88	46.36	**51.21**	5.7	12.0	12.0
	(std) 0.38	0.89	0.70	0.6	0.0	1.0
CIFAR100	**21.69**	18.57	19.21	5.7	12.0	13.0
	(std) 0.41	0.92	0.32	0.6	0.0	0.0
EUROSAT	**71.09**	69.56	62.59	5.7	12.3	15.7
	(std) 0.78	0.78	0.92	0.6	0.6	1.2

TABLE II: We compare the accuracy of KAN with a fixed number of bases, an MLP, and INFINITYKAN on the regression tasks: Spiral $k = 2$, and Spiral $k = 3$.

Dataset	INFINITYKAN	KAN	MLP	L0	L1	l2
Spiral $k = 2$	5.55	**6.59**	6.11	12.3	6.0	5.3
	(std) 1.11	0.26	0.19	2.1	2.0	1.2
Spiral $k = 3$	5.05	**5.37**	5.23	12.3	5.3	6.3
	(std) 0.39	0.16	0.61	2.1	0.6	0.6

accuracy. While for the regression, we report the negative log loss at the end of the training epochs. We use the Hubert loss for the regression task, while we use the cross-entropy loss for the classification task.

V. RESULTS

In Table I, we show the accuracy of MLP, KAN, and INFINITYKAN, for the classification tasks and different datasets. We notice that INFINITYKAN generally outperforms KAN, while it improved over the MLP for CIFAR10, CIFAR100, and EUROSAT datasets. In the last three columns, we report the number of bases per layer. The number of bases increases for the last layers, while it decreases for the first layers. INFINITYKAN exhibits relatively high variation in the test accuracy, while MLP and KAN reduce test accuracy during training, possibly induced by overfitting to the training dataset. In Table II, we show the performance of the regression tasks. In this case, the KAN with a fixed number of bases performs better overall. The last three columns of the table, report the number of bases per layer, where we find the opposite behavior we observed compared to Table I. Therefore, the number of bases per layer heavily depends on the task.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

We proposed INFINITYKAN, a variational inference method for training KAN model with a potentially infinite number of bases for each of the layers. Our experiments show the impact in terms of accuracy or regression error for both classification and regression tasks, where INFINITYKAN performs generally well on image classification tasks and displays a non-trivial number of learned bases per layer. We hope that INFINITYKAN will broaden the scope of applicability of KANs.

REFERENCES

[1] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "Kan: Kolmogorov-arnold networks," no. arXiv:2404.19756, Jun. 2024, arXiv:2404.19756 [cs]. [Online]. Available: http://arxiv.org/abs/2404.19756

[2] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[3] A. N. Kolmogorov, *On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables*. American Mathematical Society, 1961.

[4] M. Laczkovich, "A superposition theorem of Kolmogorov type for bounded continuous functions," *Journal of Approximation Theory*, vol. 269, p. 105609, 2021.

[5] V. Kreinovich, H. T. Nguyen, and D. A. Sprecher, "Normal Forms For Fuzzy Logic — An Application Of Kolmogorov'S Theorem," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 04, no. 04, pp. 331–349, Aug. 1996.

[6] M. Köppen, "On the training of a kolmogorov network," in *Artificial Neural Networks—ICANN 2002: International Conference Madrid, Spain, August 28–30, 2002 Proceedings 12*. Springer, 2002, pp. 474–479.

[7] V. Kŭrková, "Kolmogorov's theorem and multilayer neural networks," *Neural networks*, vol. 5, no. 3, pp. 501–506, 1992.

[8] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "Kan: Kolmogorov-arnold networks," 2024.

[9] K. Xu, L. Chen, and S. Wang, "Are kan effective for identifying and tracking concept drift in time series?" 2024. [Online]. Available: https://arxiv.org/abs/2410.10041

[10] G. D. Carlo, A. Mastropietro, and A. Anagnostopoulos, "Kolmogorov-arnold graph neural networks," 2024. [Online]. Available: https://arxiv.org/abs/2406.18354

[11] S. SS, K. AR, G. R, and A. KP, "Chebyshev polynomial-based kolmogorov-arnold networks: An efficient architecture for nonlinear function approximation," 2024. [Online]. Available: https://arxiv.org/abs/2405.07200

[12] F. Mostajeran and S. A. Faroughi, "Epi-ckans: Elasto-plasticity informed kolmogorov-arnold networks using chebyshev polynomials," 2024. [Online]. Available: https://arxiv.org/abs/2410.10897

[13] Z. Bozorgasl and H. Chen, "Wav-kan: Wavelet kolmogorov-arnold networks," 2024. [Online]. Available: https://arxiv.org/abs/2405.12832

[14] J. Xu, Z. Chen, J. Li, S. Yang, W. Wang, X. Hu, and E. C. H. Ngai, "Fourierkan-gcf: Fourier kolmogorov-arnold network – an effective and efficient feature transformation for graph collaborative filtering," 2024. [Online]. Available: https://arxiv.org/abs/2406.01034

[15] M.-J. Lai and Z. Shen, "The kolmogorov superposition theorem can break the curse of dimensionality when approximating high dimensional functions," *arXiv preprint arXiv:2112.09963*, 2021.

[16] T. Poggio, "How deep sparse networks avoid the curse of dimensionality: Efficiently computable functions are compositionally sparse," *CBMM Memo*, vol. 10, p. 2022, 2022.

[17] M. M. Ferdaus, M. Abdelguerfi, E. Ioup, D. Dobson, K. N. Niles, K. Pathak, and S. Sloan, "KANICE: Kolmogorov-Arnold Networks with Interactive Convolutional Elements," Oct. 2024.

[18] X. Yang and X. Wang, "Kolmogorov-Arnold Transformer," Sep. 2024.

[19] L. Hu, Y. Wang, and Z. Lin, "EKAN: Equivariant Kolmogorov-Arnold Networks," Oct. 2024.

[20] F. Alesiani, T. Maruyama, H. Christiansen, and V. Zaverkin. Geometric Kolmogorov-Arnold Superposition Theorem. [Online]. Available: http://arxiv.org/abs/2502.16664

[21] S. Somvanshi, S. A. Javed, M. M. Islam, D. Pandit, and S. Das, "A Survey on Kolmogorov-Arnold Network," Nov. 2024.

[22] V. Kůrková, "Kolmogorov's theorem and multilayer neural networks," *Neural Networks*, vol. 5, no. 3, pp. 501–506, 1991.

[23] S. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Proceedings of the 3rd Conference on Neural Information Processing Systems (NIPS)*, 1989.

[24] L. Wu, B. Liu, P. Stone, and Q. Liu, "Firefly neural architecture descent: a general approach for growing neural networks," in *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.

[25] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," in *6th International Conference on Learning Representations (ICLR)*, 2018.

[26] L. Wu, D. Wang, and Q. Liu, "Splitting steepest descent for growing neural architectures," in *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[27] R. Mitchell, M. Mundt, and K. Kersting, "Self expanding neural networks," *arXiv preprint*, 2023.

[28] A. Nazaret and D. Blei, "Variational inference for infinitely deep neural networks," in *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

[29] F. Errica, H. Christiansen, V. Zaverkin, T. Maruyama, M. Niepert, and F. Alesiani, "Adaptive message passing: A general framework to mitigate oversmoothing, oversquashing, and underreaching," *arXiv preprint*, 2024.

[30] F. Errica, H. Christiansen, V. Zaverkin, M. Niepert, and F. Alesiani, "Adaptive width neural networks," 2025. [Online]. Available: https://arxiv.org/abs/2501.15889

[31] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, pp. 183–233, 1999.

[32] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[33] A. Krizhevsky, "Learning multiple layers of features from tiny images," *Master's thesis, University of Toronto*, 2009.

[34] P. Helber, B. Bischke, A. Dengel, and D. Borth, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

[35] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. pmlr, 2015, pp. 448–456.