Adaptive Basis Functions for Enhanced Kolmogorov–Arnold Neural Networks: A Dynamic Approach to Universal Function Approximation

Aditya Chakraborty, Dr. Nirmala Ramakrishnan

**Abstract**

Kolmogorov–Arnold Neural Networks (KANs) are powerful tools for approximating complex functions based on the Kolmogorov–Arnold theorem. However, they face several challenges, such as being inflexible due to fixed basis functions, struggling with high-dimensional data, and requiring significant computational resources. In this paper, we propose a new approach that introduces **adaptive basis functions** to the KAN framework. These basis functions can adjust dynamically during training, allowing the network to better fit complex patterns while reducing computational demands. Our experiments show that this method improves accuracy, handles noisy data more effectively, and generalizes better to new inputs. By addressing these key limitations, our work makes KANs more practical and versatile for solving real-world problems.

**Introduction**

The ability of neural networks to approximate functions has long been an endeavor in the AI field of computer science. In the realm of mathematical modeling, there exists the universal function approximation theorem for replicating patterns and complex systems and creating predictions based on these models. MLPs utilize the Universal Approximation Theorem. Kolmogorov Arnold Neural Networks emerge powerful in the industry by utilizing the Kolmogorov-Arnold Representation theorem [[1],](#Citationone) which decomposes multivariate functions into the sum of singular univariate functions.

Where and

This approach effectively tackles the curse of dimensionality problem for high-dimension datasets, as seen in [[1]](#Citationone).

Approximating real-valued functions that are too complicated or unmanageable for direct evaluation is a frequent problem in computational economics and numerical analysis. Applications where precise solutions are frequently impossible, such as asset pricing and dynamic optimization, frequently call for function approximation. The process of approximating a function  involves selecting a computationally feasible approximant and two common problems emerge: interpolation and functional equation solving.

Interpolation refers to the process of constructing an approximant that matches the known values or derivatives of the function  at specific data points. Modern interpolation techniques extend beyond simple table-based approximations, focusing on optimal data extraction and computational efficiency. In contrast, the functional equation problem entails finding a function  that satisfies a functional equation , where  is an operator mapping functions to functions and  is known. This type of problem is frequently encountered in dynamic economic models, such as Bellman and Euler equations.

Function approximation methods like polynomial and spline interpolation, as well as more advanced techniques like collocation methods, are essential tools in addressing both of these problems. This discussion explores the principles of interpolation, the benefits of different polynomial bases, and the advantages of Chebyshev nodes, which have been shown to significantly improve the accuracy and stability of approximations in comparison to traditional methods of *Liu et al.* [[1]](#Citationone) and *Boston College* [[4]](#Citationfour).

Traditional implementations of KANs, however, rely on static-based functions, which hampers their ability to capture non-linear relationships in real-world data. To address these limitations, this paper proposes integrating adaptive basis functions into KANs. By dynamically changing basis functions in the training process, KANs can recognize complex, non-linear relationships between the feature and target.

This paper introduces a dynamic methodology for universal function approximation by incorporating adaptive basis functions into KANs. The proposed approach not only improves approximation accuracy but also offers insights into the interplay between adaptability and efficiency in neural network design. This approach leads to faster convergence and improved generalization in the model.

**Related Works**

**Kolmogorov-Arnold Superposition and Representation Theorem**

**(KAST & KART)**

Hilbert’s 13th problem proposed that any continuous function on any finite interval could be represented as a finite sum of continuous functions or other forms, such as polynomials. The challenge was finding a **universal approximation** of any continuous functions, as seen in Liu et al. [[1]](#Citationone) and [[3]](#Citationthree). Kolmogorov defined that on a compact interval, such as [0,1], continuous functions could be expressed on the intervals of two-variable functions.

The original version of the KA representation theorem states that for any continuous function , there exists univariate functions such that

This means that the Univariate functions and are enough for an exact representation of a -variate function. Kolmogorov published the result in 1957, disproving the statement of Hilbert’s 13th problem that is concerned with the solution of algebraic equations. The earliest proposals in the literature introducing multiple layers in neural networks date back to the sixties, and the link between KA representation and multilayer neural networks occurred much later[[2]](#Citationtwo). Kolmogorov and Arnold show that every continuous multivariate function can be represented as a superposition of continuous univariate functions and addition in a universal form and thus solve the problem positively. In Kolmogorov’s representation, only one univariate function (the outer function) depends on it, and all the other univariate functions (inner functions) are independent of the multivariate function to be represented in [[3]](#Citationthree).

**KAN Architecture**

KANs are specifically designed to follow the Kolmogorov-Arnold Representation Theorem (KART), which decomposes multivariate functions into the sum of univariate functions. This setup provides a way to where each layer has a matrix of 1D function parametrized as B-spline curves with trainable coefficients. The layers connect input and output through summation operations on nodes and edges, as seen in [[1]](#Citationone).

Deeper KANs are generated by stacking layers of such, each as such that p and q are input and output dimensions, respectively. These layers process inputs through differentiable transformation, making them suitable for backpropagation. The shape of a KAN is represented by where representing the number of nodes in the i-th layer, as seen in [[1]](#Citationone). The output, , of a KAN model is, therefore, the composition of all layers inputting x.

Such that represents the whole function ‘set’ of the -th layer as per the KART, where the function set is :

KAN introduces several optimization techniques to improve training. Residual activation functions combine B-spline approximations with a smooth residual component, enabling better learning dynamics. During training, spline grids dynamically adjust to maintain stability as activations change. The trainability of the network is further improved via parameter initialization, which includes techniques like Xavier initialization. Compared to MLPs, KANs may appear to require more parameters, but this expense is mitigated by their capacity to achieve high generalization with fewer nodes per layer. KANs are excellent in representing and interpreting high-dimensional functions; in some applications, they outperform traditional structures thanks to their special structure [[1]](#Citationone).

KANs share completely connected structures with MLPs. However, KANs place learnable activation functions on edges (also known as "weights"), whereas MLPs place fixed activation functions on nodes (also known as "neurons") [[1]](#Citationone). Activation functions are treated as weights such that they are learnable, making the functional relationship between nodes more adaptive and stable, as shown by Liu et al. [[1]](#Citationone) and Schmidt-Hieber [[2]](#Citationtwo).

As a result, KANs have no linear weight matrices at all; instead, each weight parameter is replaced by a learnable 1D function parametrized as a spline. KANs’ nodes simply sum incoming signals without applying any non-linearities. One might worry that KANs are hopelessly expensive since each MLP’s weight parameter becomes KAN’s spline function. Fortunately, KANs usually allow much smaller computation graphs than MLPs, as seen in [[1]](#Citationone).

**Adaptive Basis Function**

As per Adams in [[5]](#Citationfive), in the context of machine learning, basis functions are mathematical transformations applied to input data, enabling linear models to approximate more complex relationships. These transformations map input features into a higher-dimensional space, allowing linear regression and classification models to capture nonlinear patterns. Basis functions can include simple polynomials, Fourier series, radial basis functions (RBFs), or piecewise linear transformations. They play a crucial role in making data linearly separable or better suited for predictive modeling by effectively altering the representation of input features, as shown by Liu et al. [[1]](#Citationone).

Adaptive Basis Functions (ABFs) are functions used in mathematical modeling to adapt to data. Contrasting to fixed basis functions, such as splines, ABFs change based on the problem at hand.

Examples of basis functions include simple polynomials, Fourier series, radial basis functions (RBFs), or piecewise linear transformations. Each type of basis function introduces a unique way to represent input features, offering flexibility in addressing a variety of modeling challenges. For instance, polynomial basis functions are often used to capture trends in data with curved patterns, while RBFs excel in localized transformations that emphasize certain regions of the input space.

The role of basis functions is particularly critical in enhancing the separability of data. By altering the representation of input features, basis functions enable machine learning algorithms to model relationships that would otherwise be difficult or impossible to capture using linear transformations alone. This principle underpins the effectiveness of kernel-based methods like Support Vector Machines (SVMs) and Gaussian Process Regression, where basis functions are implicitly defined through kernel functions 1.

ABFs use differentiable methods to process inputs, which can be set up for training through backpropagation. ABFs are represented as a set of parameters like the b-spline, which sets parameters as control points, slopes, and degrees of polynomials. ABFs can have location and shape parameters similar to Gaussian Functions with standard deviation ().

**Methodology**

This experiment aims to investigate the performance of two types of KANS-one with a fixed basis function and another with an adaptive basis function. Both will be targeted for effectiveness in approximation and accuracy using mean-squared error over a sine function in the range from [0,10] and will use the Adam optimizer.

Both KANs have a control architecture with an input layer, a basis function, and a linear output layer.

**Control Experiment**

The control experiment will use the Gaussian Radial Basis Function (GRBF). The GRBF is represented as such:

Where is the parameter which controls the width of the curve. The GRBF is a smooth bell shape and is commonly used for interpolation and kernel methods. This function is a default basis function for a KAN. This basis function is a universal function approximator; a linear combination of Gaussian functions can arbitrarily approximate any continuous function given parameters. GRBFs are smooth and differentiable, making them suitable for convergence as a default for many KANs.

The GBRF has fixed mean and standard deviation parameters, which are not refined during the training process. The input is passed through the GBRF and outputs the weighted sum of the basis function outputs.

**Adaptive KAN**

In the Adaptive KAN, the centers and width are dynamically adjusted in backpropagation. The Adaptive Gaussian Radial Basis Functions (aGRBFs). These parameters are refined to adapt over time, which enhances function approximation and detection of non-linear relationships.

**Dataset**

The dataset comprises a sine wave data using Gaussian noising. The input values range from , and the corresponding y-values are outputted from the function such that is Gaussian noising to simulate real-world datasets. The data is divided into training, validation, and test sets.

**Training**

The model is trained using Mean Squared Error over the Adam optimizer. Both experiments are trained over 100 epochs on a batch size of 32. For the control experiment, the GRBFs’ centers and standard deviations are randomly initialized, and the second experiment’s aGRBFs’ are randomly initialized and then optimized during the training process.

**Results**

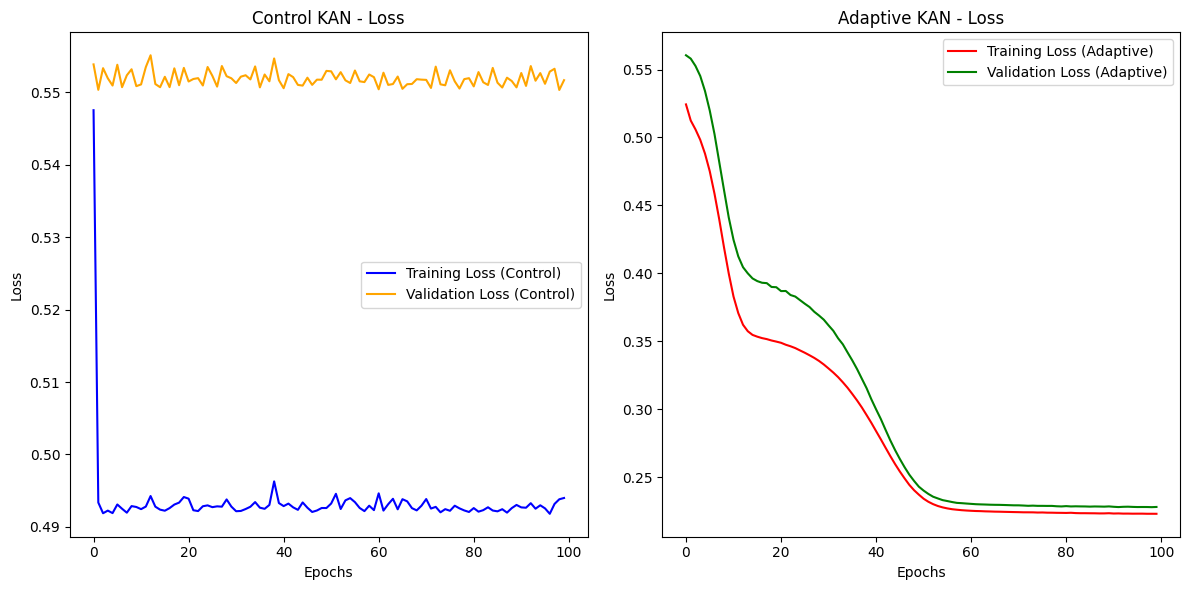


Figure 1

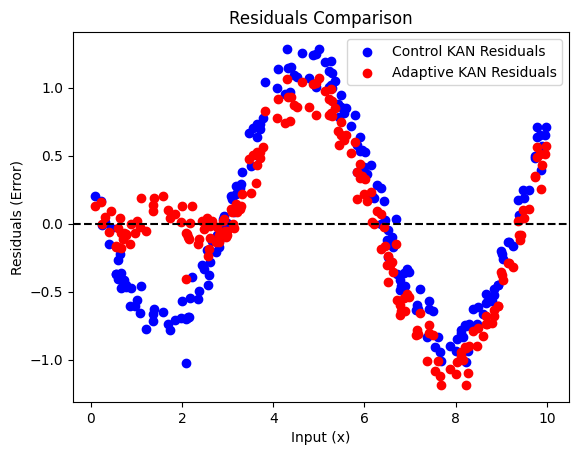


Figure 2

**Results**

In **Figure 1**, it is evident that the Adaptive KAN outperformed the regular KAN, and there is a significant difference in convergence and training. As shown in **Figure 1**, the KAN with adaptive basis functions outperformed the Control KAN with fixed basis functions, achieving a Mean Squared Error (MSE) of approximately 0.23 after around 50 epochs. **In contrast, the Control KAN maintained a consistent error of approximately 0.493 from the very first epoch onward.**

Additionally, **Figure 1** illustrates that the Control KAN exhibited significantly higher overfitting compared to the KAN with adaptive basis functions. Throughout the training, the Control KAN showed a consistent error difference of 6% between the training and validation errors, whereas the model with adaptive basis functions demonstrated near-similar error values between training and validation by the 50th epoch. **This highlights that adaptive basis functions not only enable KANs to achieve higher accuracy during training but also improve generalization and reduce the risk of overfitting.**

Moreover, the error reduction in the training process is noticeably smoother for the model with adaptive basis functions. As seen in **Figure 1**, the Control KAN experienced abrupt fluctuations in error, while the adaptive model showed a smoother decline. This suggests that the use of adaptive basis functions **promotes faster generalization and may contribute to improving issues related to vanishing and exploding gradients.**

As seen in Figure 2, this graph presents a comparative analysis of residual errors generated by two distinct neural network architectures: Control Kolmogorov-Arnold Network (KAN) and Adaptive KAN. Residuals, defined as the difference between predicted and actual values, serve as a metric for evaluating predictive accuracy. The data, depicted as a scatter plot, reveals that the Adaptive KAN exhibits a reduced magnitude of residuals compared to the Control KAN, as evidenced by the closer proximity of red data points to the zero-error line. This observation indicates a higher degree of predictive accuracy for the Adaptive KAN. Both networks demonstrate a sinusoidal pattern in their residual distributions, suggesting a systematic component in the prediction errors. However, the Adaptive KAN's ability to dynamically adjust its parameters results in a significant reduction in the magnitude of these errors. Therefore, it can be concluded that the Adaptive KAN, which incorporates adaptive basis functions, demonstrates superior predictive performance compared to the standard Control KAN. This enhanced accuracy is attributed to the network's capacity to optimize its parameters during the learning process, effectively minimizing prediction errors.

**Future Works**

The investigation into adaptive basis functions within Kolmogorov–Arnold Neural Networks (KANs) presents several promising directions for future research and innovation. While this study highlights the potential of adaptive basis functions to enhance approximation accuracy and generalization, significant opportunities remain to deepen and expand this work.

Future research can focus on designing and testing novel KAN architectures that integrate adaptive basis functions more seamlessly. Such designs might include multi-resolution or hierarchical basis functions, enabling the networks to capture intricate and highly nonlinear relationships more effectively.

A critical area of exploration involves improving the computational efficiency of adaptive basis functions. Efforts could target the development of optimized learning algorithms that minimize the added computational overhead, ensuring that adaptability does not come at the expense of scalability or training speed.

Another direction involves applying the proposed adaptive KANs to a wider range of real-world problems. These include, but are not limited to, robotics, financial modeling, climate science, and bioinformatics. Such applications would help validate the versatility and robustness of the proposed framework in diverse scenarios.

By pursuing these directions, the research community can continue to build upon the foundational contributions of this study, advancing the capabilities of Kolmogorov–Arnold Neural Networks in both theory and application.

**Conclusion**

Kolmogorov–Arnold Neural Networks (KANs) enhanced with Adaptive Basis Functions demonstrate superior performance compared to regular KANs. The incorporation of adaptive basis functions leads to smoother and faster learning processes, which in turn improves generalization. These improvements highlight the potential of adaptive techniques in optimizing KANs for more efficient and effective applications. Future studies could explore further refinements to the adaptive basis function approach, as well as its applicability across different domains, to fully realize its potential in enhancing neural network performance.

Citations

[1] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T.Y Hou, M. Tegmark, "*KAN: Kolmogorov–Arnold Networks* "  vol. arXiv:2404.19756, 2024

[2] J. Schmidt-Hieber, “*The Kolmogorov-Arnold representation theorem revisited*” vol. arXiv: 2007.15884, 2020

[3] Liu et al., “*Kolmogorov superposition theorem and its applications”* 2015

[4] Boston College, “*Function Approximation”*

[5] R.P. Adams, “*Features and Basis Functions”* Princeton University, Dept. of Computer Science, 2018