

Title: Kolmogorov-Arnold Networks

URL: https://en.wikipedia.org/wiki/Kolmogorov-Arnold_Networks

PageID: 81016886

Categories: Category:Artificial neural networks, Category:Deep learning, Category:Machine learning

Source: Wikipedia (CC BY-SA 4.0).

Kolmogorov–Arnold Networks (KANs) are a type of artificial neural network architecture inspired by the Kolmogorov–Arnold representation theorem, also known as the superposition theorem. Unlike traditional multilayer perceptrons (MLPs), which rely on fixed activation functions and linear weights, KANs replace each weight with a learnable univariate function, often represented using splines. [1] [2] [3]

Architecture

KANs are based on the Kolmogorov–Arnold representation theorem, which was linked to the 13th Hilbert problem. [4] [5]

Given $x = (x_1, x_2, \dots, x_n)$ consisting of n variables, a multivariate continuous function $f(x)$ can be represented as:

This formulation contains two nested summations: an outer and an inner sum. The outer sum $\sum_{q=1}^{2n+1}$ aggregates $2n+1$ terms, each involving a function $\Phi_q: \mathbb{R} \rightarrow \mathbb{R}$.

The inner sum $\sum_{p=1}^n$ computes n terms for each q , where each term $\phi_{q,p}: [0, 1] \rightarrow \mathbb{R}$ is a continuous function of the single variable x_p .

Liu et al. [1] proposed the name KAN. A general KAN network consisting of L layers takes x to generate the output as:

Here, Φ^l is the function matrix of the l -th KAN layer or a set of pre-activations.

Let i denote the neuron of the l -th layer and j the neuron of the $(l+1)$ -th layer. The activation function $\phi_{j,i}^l$ connects (l, i) to $(l+1, j)$:

where n_l is the number of nodes of the l -th layer.

Thus, the function matrix Φ^l can be represented as an $(n_l + 1) \times n_l$ matrix of activations:

Functions used in KAN

The choice of functional basis strongly influences the performance of KANs. Common function families include:

B-splines : Provide locality, smoothness, and interpretability; most widely used in current implementations. [3]

RBFs : Capture localized features in data and are effective in approximating functions with non-linear or clustered structures. [3] [6]

Chebyshev polynomials : Offer efficient approximation with minimized error in the maximum norm, making them useful for stable function representation. [3] [7]

Rational functions: Useful for approximating functions with singularities or sharp variations, as they can model asymptotic behavior better than polynomials. [3] [8]

Fourier series : Capture periodic patterns effectively and are particularly useful in domains such as physics-informed machine learning . [3] [9]

Wavelet functions [3] [10]

Usage

KANs are usually employed as drop-in replacements for MLP layers in modern neural architectures such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and Transformers . Researchers have applied them in a variety of tasks:

Function fitting: KANs outperform MLPs of similar parameter size in tasks like fitting symbolic formulas or special functions. [1]

Solving partial differential equations (PDEs): A two-layer, 10-width KAN can outperform a four-layer, 100-width MLP by two orders of magnitude in both accuracy and parameter efficiency. [1] [11] [12]

Continual learning: KANs better preserve previously learned information during incremental updates, avoiding catastrophic forgetting —owing to the locality of spline adjustments. [2] [13]

Scientific discovery: Due to the interpretability of learned functions, KANs have been used as a tool for rediscovering physical or mathematical laws. [2]

Graph neural networks: Extensions such as Kolmogorov–Arnold Graph Neural Networks (KA-GNNs) integrate KAN modules into message-passing architectures, showing improvements in molecular property prediction tasks. [3] [14] [15]

See also

Kolmogorov–Arnold representation theorem

Universal approximation theorem

References