

KAN: Kolmogorov–Arnold Networks

Ziming Liu^{1,4*} Yixuan Wang² Sachin Vaidya¹ Fabian Ruehle^{3,4}
James Halverson^{3,4} Marin Soljačić^{1,4} Thomas Y. Hou² Max Tegmark^{1,4}

¹ Massachusetts Institute of Technology

² California Institute of Technology

³ Northeastern University

⁴ The NSF Institute for Artificial Intelligence and Fundamental Interactions

Abstract

Inspired by the Kolmogorov-Arnold representation theorem, we propose Kolmogorov-Arnold Networks (KANs) as promising alternatives to Multi-Layer Perceptrons (MLPs). While MLPs have *fixed* activation functions on *nodes* (“neurons”), KANs have *learnable* activation functions on *edges* (“weights”). KANs have no linear weights at all – every weight parameter is replaced by a univariate function parametrized as a spline. We show that this seemingly simple change makes KANs outperform MLPs in terms of accuracy and interpretability. For accuracy, much smaller KANs can achieve comparable or better accuracy than much larger MLPs in data fitting and PDE solving. Theoretically and empirically, KANs possess faster neural scaling laws than MLPs. For interpretability, KANs can be intuitively visualized and can easily interact with human users. Through two examples in mathematics and physics, KANs are shown to be useful “collaborators” helping scientists (re)discover mathematical and physical laws. In summary, KANs are promising alternatives for MLPs, opening opportunities for further improving today’s deep learning models which rely heavily on MLPs.

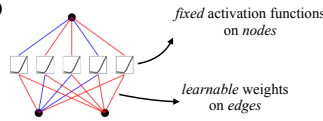
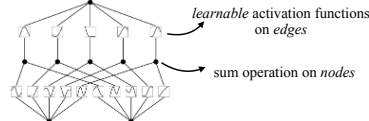
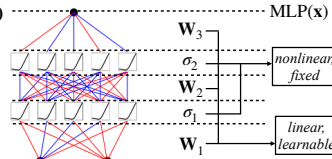
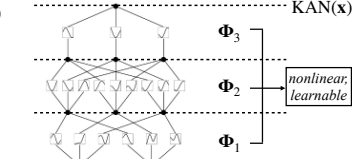
Model	Multi-Layer Perceptron (MLP)	Kolmogorov-Arnold Network (KAN)
Theorem	Universal Approximation Theorem	Kolmogorov-Arnold Representation Theorem
Formula	$f(\mathbf{x}) \approx \sum_{i=1}^{N(e)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$	$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^n \phi_{q,p}(x_p) \right)$
Model (Shallow)	(a)  fixed activation functions on nodes learnable weights on edges	(b)  learnable activation functions on edges sum operation on nodes
Formula (Deep)	$\text{MLP}(\mathbf{x}) = (\mathbf{W}_3 \circ \sigma_2 \circ \mathbf{W}_2 \circ \sigma_1 \circ \mathbf{W}_1)(\mathbf{x})$	$\text{KAN}(\mathbf{x}) = (\Phi_3 \circ \Phi_2 \circ \Phi_1)(\mathbf{x})$
Model (Deep)	(c)  MLP(x) \mathbf{W}_3 σ_2 \mathbf{W}_2 σ_1 \mathbf{W}_1 nonlinear, fixed linear, learnable x	(d)  KAN(x) Φ_3 Φ_2 Φ_1 nonlinear, learnable x

Figure 0.1: Multi-Layer Perceptrons (MLPs) vs. Kolmogorov-Arnold Networks (KANs)

*zmliu@mit.edu

1 Introduction

Multi-layer perceptrons (MLPs) [1, 2, 3], also known as fully-connected feedforward neural networks, are foundational building blocks of today’s deep learning models. The importance of MLPs can never be overstated, since they are the default models in machine learning for approximating nonlinear functions, due to their expressive power guaranteed by the universal approximation theorem [3]. However, are MLPs the best nonlinear regressors we can build? Despite the prevalent use of MLPs, they have significant drawbacks. In transformers [4] for example, MLPs consume almost all non-embedding parameters and are typically less interpretable (relative to attention layers) without post-analysis tools [5].

We propose a promising alternative to MLPs, called Kolmogorov-Arnold Networks (KANs). Whereas MLPs are inspired by the universal approximation theorem, KANs are inspired by the Kolmogorov-Arnold representation theorem [6, 7]. Like MLPs, KANs have fully-connected structures. However, while MLPs place fixed activation functions on *nodes* (“neurons”), KANs place learnable activation functions on *edges* (“weights”), as illustrated in Figure 0.1. 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. For example, we show that for PDE solving, a 2-Layer width-10 KAN is **100 times more accurate** than a 4-Layer width-100 MLP (10^{-7} vs 10^{-5} MSE) and **100 times more parameter efficient** (10^2 vs 10^4 parameters).

Unsurprisingly, the possibility of using Kolmogorov-Arnold representation theorem to build neural networks has been studied [8, 9, 10, 11, 12, 13]. However, most work has stuck with the original depth-2 width- $(2n + 1)$ representation, and did not have the chance to leverage more modern techniques (e.g., back propagation) to train the networks. Our contribution lies in generalizing the original Kolmogorov-Arnold representation to arbitrary widths and depths, revitalizing and contextualizing it in today’s deep learning world, as well as using extensive empirical experiments to highlight its potential role as a foundation model for AI + Science due to its accuracy and interpretability.

Despite their elegant mathematical interpretation, KANs are nothing more than combinations of splines and MLPs, leveraging their respective strengths and avoiding their respective weaknesses. Splines are accurate for low-dimensional functions, easy to adjust locally, and able to switch between different resolutions. However, splines have a serious curse of dimensionality (COD) problem, because of their inability to exploit compositional structures. MLPs, On the other hand, suffer less from COD thanks to their feature learning, but are less accurate than splines in low dimensions, because of their inability to optimize univariate functions. To learn a function accurately, a model should not only learn the compositional structure (*external* degrees of freedom), but should also approximate well the univariate functions (*internal* degrees of freedom). KANs are such models since they have MLPs on the outside and splines on the inside. As a result, KANs can not only learn features (thanks to their external similarity to MLPs), but can also optimize these learned features to great accuracy (thanks to their internal similarity to splines). For example, given a high dimensional function

$$f(x_1, \dots, x_N) = \exp \left(\frac{1}{N} \sum_{i=1}^N \sin^2(x_i) \right), \quad (1.1)$$

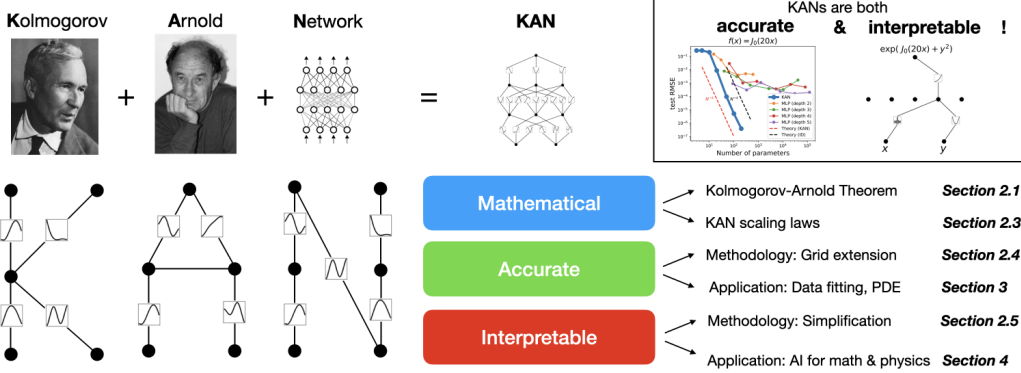


Figure 2.1: Our proposed Kolmogorov-Arnold networks are in honor of two great late mathematicians, Andrey Kolmogorov and Vladimir Arnold. KANs are mathematically sound, accurate and interpretable.

splines would fail for large N due to COD; MLPs can potentially learn the the generalized additive structure, but they are very inefficient for approximating the exponential and sine functions with say, ReLU activations. In contrast, KANs can learn both the compositional structure and the univariate functions quite well, hence outperforming MLPs by a large margin (see Figure 3.1).

Throughout this paper, we will use extensive numerical experiments to show that KANs can lead to remarkable accuracy and interpretability improvement over MLPs. The organization of the paper is illustrated in Figure 2.1. In Section 2, we introduce the KAN architecture and its mathematical foundation, introduce network simplification techniques to make KANs interpretable, and introduce a grid extension technique to make KANs increasingly more accurate. In Section 3, we show that KANs are more accurate than MLPs for data fitting and PDE solving: KANs can beat the curse of dimensionality when there is a compositional structure in data, achieving much better scaling laws than MLPs. In Section 4, we show that KANs are interpretable and can be used for scientific discoveries. We use two examples from mathematics (knot theory) and physics (Anderson localization) to demonstrate that KANs can be helpful “collaborators” for scientists to (re)discover math and physical laws. Section 5 summarizes related works. In Section 6, we conclude by discussing broad impacts and future directions. Codes are available at <https://github.com/KindXiaoming/pykan> and can also be installed via `pip install pykan`.

2 Kolmogorov–Arnold Networks (KAN)

Multi-Layer Perceptrons (MLPs) are inspired by the universal approximation theorem. We instead focus on the Kolmogorov-Arnold representation theorem, which can be realized by a new type of neural network called Kolmogorov-Arnold networks (KAN). We review the Kolmogorov-Arnold theorem in Section 2.1, to inspire the design of Kolmogorov-Arnold Networks in Section 2.2. In Section 2.3, we provide theoretical guarantees for the expressive power of KANs and their neural scaling laws. In Section 2.4, we propose a grid extension technique to make KANs increasingly more accurate. In Section 2.5, we propose simplification techniques to make KANs interpretable.

2.1 Kolmogorov-Arnold Representation theorem

Vladimir Arnold and Andrey Kolmogorov established that if f is a multivariate continuous function on a bounded domain, then f can be written as a finite composition of continuous functions of a

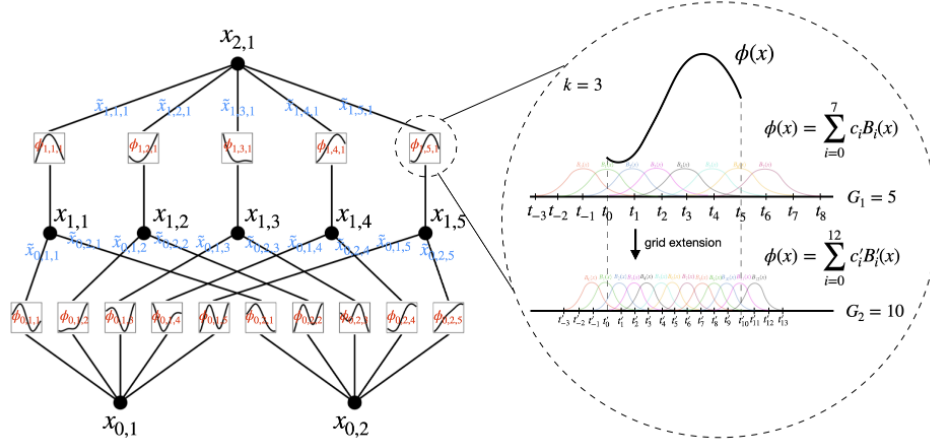


Figure 2.2: Left: Notations of activations that flow through the network. Right: an activation function is parameterized as a B-spline, which allows switching between coarse-grained and fine-grained grids.

single variable and the binary operation of addition. More specifically, for a smooth $f : [0, 1]^n \rightarrow \mathbb{R}$,

$$f(\mathbf{x}) = f(x_1, \dots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^n \phi_{q,p}(x_p) \right), \quad (2.1)$$

where $\phi_{q,p} : [0, 1] \rightarrow \mathbb{R}$ and $\Phi_q : \mathbb{R} \rightarrow \mathbb{R}$. In a sense, they showed that the only true multivariate function is addition, since every other function can be written using univariate functions and sum. One might naively consider this great news for machine learning: learning a high-dimensional function boils down to learning a polynomial number of 1D functions. However, these 1D functions can be non-smooth and even fractal, so they may not be learnable in practice [14]. Because of this pathological behavior, the Kolmogorov-Arnold representation theorem was basically sentenced to death in machine learning, regarded as theoretically sound but practically useless [14].

However, we are more optimistic about the usefulness of the Kolmogorov-Arnold theorem for machine learning. First of all, we need not stick to the original Eq. (2.1) which has only two-layer nonlinearities and a small number of terms ($2n + 1$) in the hidden layer: we will generalize the network to arbitrary widths and depths. Secondly, most functions in science and daily life are often smooth and have sparse compositional structures, potentially facilitating smooth Kolmogorov-Arnold representations. The philosophy here is close to the mindset of physicists, who often care more about typical cases rather than worst cases. After all, our physical world and machine learning tasks must have structures to make physics and machine learning useful or generalizable at all [15].

2.2 KAN architecture

Suppose we have a supervised learning task consisting of input-output pairs $\{\mathbf{x}_i, y_i\}$, where we want to find f such that $y_i \approx f(\mathbf{x}_i)$ for all data points. Eq. (2.1) implies that we are done if we can find appropriate univariate functions $\phi_{q,p}$ and Φ_q . This inspires us to design a neural network which explicitly parametrizes Eq. (2.1). Since all functions to be learned are univariate functions, we can parametrize each 1D function as a B-spline curve, with learnable coefficients of local B-spline basis functions (see Figure 2.2 right). Now we have a prototype of KAN, whose computation graph is exactly specified by Eq. (2.1) and illustrated in Figure 0.1 (b) (with the input dimension $n = 2$), appearing as a two-layer neural network with activation functions placed on edges instead of nodes (simple summation is performed on nodes), and with width $2n + 1$ in the middle layer.