Super-KAN (Sparse Superposition KAN)

Gary Nan Tie, Oct 26, 2024

Abstract
We introduce a sparse Kolmogorov-Arnold network with superposition layers, each direct summand a version of the Kolmogorov-Arnold representation theorem. Super-KAN are interpretable, enjoy compositional sparsity, and have linear runtime.

Consider a Kolmogorov-Arnold network [Liu et al, 2024]

of shape $[n_0, n_1, \ldots, n_L]$

$$x_{l+1} = \Phi_l \, x_l$$

$$= \underset{n_{l+1} \times n_l}{[\varphi_{l,j,i}]} \; \underset{n_l \times 1}{[x_{l,i}]}$$

$$\Phi_l : \mathbb{R}^{n_l} \to \mathbb{R}^{n_{l+1}}, \quad \Phi_l = (f_1, \ldots, f_{n_{l+1}}), \quad f_k : \mathbb{R}^{n_l} \to \mathbb{R}$$

For now, suppress the subscript k as being understood.

Let $d = n_l$, $d \geqslant 2$ and $x = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$.

For <u>Super-KAN</u>, define $f : \mathbb{R}^d \to \mathbb{R}$

as a superposition of shape $[d, 2d+1, 1]$,

$$f(x) \triangleq \underset{1 \times (2d+1)}{\Gamma(g)} \circ \underset{(2d+1) \times d}{\Psi(h)} \; \underset{d \times 1}{x}$$

where $\Psi(h) = [\Psi_{q,p}]$, $\Psi_{q,p}(x_p) = b_p h(x_p + qa) + c_q$

For $p = 1, \ldots, d$ and $q = 0, \ldots, 2d\cancel{+1}$

and $\Gamma(g) = \underset{1 \times (2d+1)}{[g, \ldots, g]}$.

For each layer l, and subscript k, we learn:

① univariate functions $g, h : \mathbb{R} \to \mathbb{R}$,

like wavelets with three parameters (shift, scale, normalization)

② parameters a, b_p, c_q

in $$f\left(\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}\right) = \sum_{q=0}^{2d\cancel{+1}} g\left(\sum_{p=1}^{d} b_p h(x_p + qa) + c_q\right),$$

a [Braun, 2009] superposition.

Note that f has 2 univariate functions *

and $3d+2$ parameters to learn.

So Super-KAN layer Φ_l has $2n_{l+1}$ univariate

functions and $(3n_l + 2)n_{l+1}$ parameters to learn.

In summary, a Super-KAN is a Kolmogorov-Arnold network of shape $[n_0, n_1, \ldots, n_L]$, where

L-th layer $\Phi_L : \mathbb{R}^{n_L} \longrightarrow \mathbb{R}^{n_{L+1}}$,

$$\Phi_L = (f_1^L, \ldots, f_{n_{L+1}}^L), \quad f_k^L : \mathbb{R}^{n_L} \longrightarrow \mathbb{R}$$

$f_k^L(x) \triangleq \Gamma(g_k^L) \circ \Psi(h_k^L)\, x$, a Braun (2009) superposition

with $\Psi(h_k^L) = [\Psi_{q,p}]$, $\Psi_{q,p}(x_p) = b_p\, h_k^L(x_p + q a) + c_q$

and a, b_p, c_q depend on (L,k), $h_k^L : \mathbb{R} \longrightarrow \mathbb{R}$,

and $\underset{1 \times (2n_L+1)}{\Gamma(g_k^L) = [g_k^L, \ldots, g_k^L]}$, $\quad g_k^L : \mathbb{R} \longrightarrow \mathbb{R}$.

Super-KAN are interpretable being a Kolmogorov-Arnold network, compositionally sparse by design, and have linear runtime in terms of univariate functions to compute.

$\square$

Appendix: each superposition is a KAN of shape [d, 2d+1, 1]

Kolmogorov-Arnold representation theorem (KART) 1957

For any continuous function $f : [0,1]^d \to \mathbb{R}$

there exist univariate continuous functions

$g_q : \mathbb{R} \to \mathbb{R}$ and $\psi_{p,q} : [0,1] \to \mathbb{R}$ such that

$$f(x_1, \dots, x_d) = \sum_{q=0}^{2d} g_q \left(\sum_{p=1}^{d} \psi_{p,q}(x_p) \right).$$

Theorem (Braun, 2009)

Fix $d \geq 2$. There are real numbers a, b_p, c_q

and a continuous and monotone $\psi : \mathbb{R} \to \mathbb{R}$

such that for any continuous function $f : [0,1]^d \to \mathbb{R}$

there exists a continuous $g : \mathbb{R} \to \mathbb{R}$ with

$$f(x_1, \dots, x_d) = \sum_{q=0}^{2d} g \left(\sum_{p=1}^{d} b_p \psi(x_p + qa) + c_q \right).$$

References

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle,
James Halverson, Marin Soljaˇci'c, Thomas Y. Hou,
Max Tegmark. (2024)
KAN: Kolmogorov–Arnold Networks
arXiv:2404.19756v4 [cs.LG]

Kolmogorov, A. N. (1957).
On the representation of continuous functions of many variables
by superposition of continuous functions of one variable and addition.
Doklady Akademii Nauk SSSR, 114, 953–956.

Braun, J. (2009).
An application of Kolmogorov's superposition theorem
to function reconstruction in higher dimensions.
(Ph.D. thesis), Universität Bonn.

Footnote:

* In $f(g,h)$ note that inner h is independent of target f, ~~other~~ only outer g depends on f.

So we can further sparsify, to reduce the time to train and run, by choosing a common inner function h across superpositions f in a layer, at the expense of expressivity.