

如何看待KAN论文被NeurIPS 2024拒稿?

目录

1. 理论基础

1. 这个定理如何与机器学习结合?
2. KAN 如何摆脱维度诅咒?

2. 什么是 Kolmogorov-Arnold Networks (KAN)?

1. KAN 架构
2. B-样条: KAN 的核心
3. 结合两者的优势

3. 网络简化

4. KAN 的创新性

5. 四个引人入胜的例子

1. 拟合符号公式
2. 特殊函数
3. 持续学习
4. 偏微分方程求解

6. 最后的思考

1. KAN 的热议是否值得？
2. KAN + 大语言模型

7. 结论

了解过KAN的人自然知道它是什么，至于被NeurIPS2024被拒，大家心里都有自己的判断。

对于不知道KAN的人，可能好奇KAN到底是什么，我把CSDN看到的一篇《深入理解 Kolmogorov-Arnold Networks (KAN)》的文章搬过来，希望不了解的人看过之后知道KAN是什么。以下是原文：

最近，一篇名为 KAN: Kolmogorov-Arnold Network 的论文在机器学习领域引起了广泛关注。这篇论文提出了一种全新的神经网络视角，并提出了一种可以替代现有多层感知器

(MLP) 的新方案。要知道，多层感知器是当前机器学习技术的基石，如果 KAN 方案证明有效，将极大地推动深度学习的发展。

Kolmogorov



+

Arnold



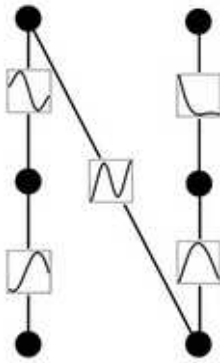
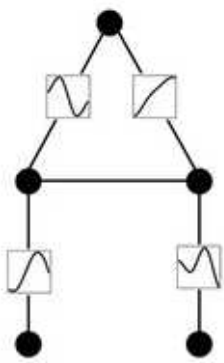
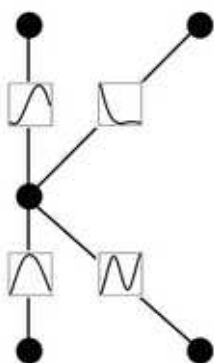
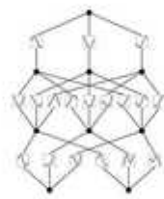
+

Network



=

KAN



Mathematical

Accurate

Interpretable

知乎 @安安

CSDE@Bloomberg

KAN 的设计灵感来源于 Kolmogorov-Arnold 表示定理，与传统的多层感知器（MLP）不同，它们通过使用可学习的函数替代固定的激活函数，从根本上消除了对线性权重矩阵的依赖。

本文将带你快速理解 KAN 的核心概念。

理论基础

KAN 的理论基础，源自两位苏联数学家 Vladimir Arnold 和 Andrey Kolmogorov 的研究成果。

他们的理论研究基于多变量连续函数的概念，根据这一理论，任何多变量连续函数 f 都可以表示为有限个单变量连续函数的组合。

$$f(\mathbf{x}) = f(x_1, \dots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^n \phi_{qp}(x_p) \right)$$

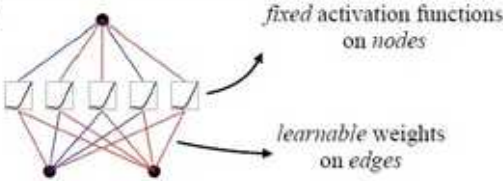
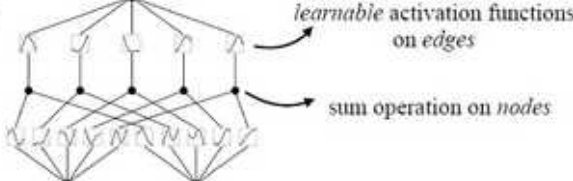
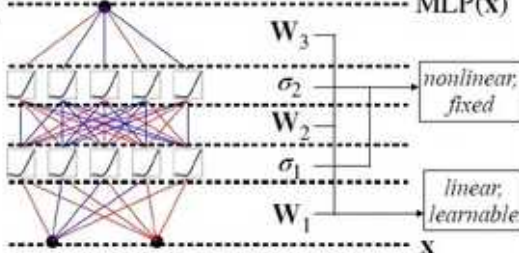
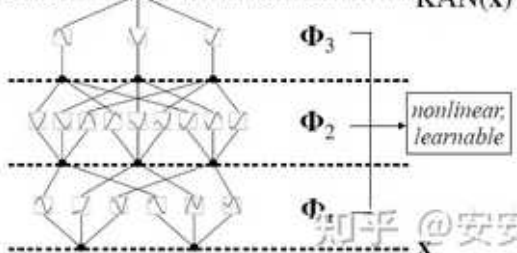
其中 $\phi_{qp} : [0, 1] \rightarrow \mathbb{R}$, $\Phi_q : \mathbb{R} \rightarrow \mathbb{R}$.

知乎 @安安

这个定理如何与机器学习结合？

在机器学习中，随着数据维度的增加，高效又精确地近似复杂函数显得尤为重要。目前的主流模型，如多层感知器（MLP），常常难以应对高维数据，这种现象被称为维度诅咒。

然而，Kolmogorov-Arnold 定理为构建能够克服这一挑战的网络（如 KAN）提供了理论支撑。

Model	Multi-Layer Perceptron (MLP)	Kolmogorov-Arnold Network (KAN)
Theorem	Universal Approximation Theorem	Kolmogorov-Arnold Representation Theorem
Formula (Shallow)	$f(\mathbf{x}) \approx \sum_{i=1}^{N(c)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$	$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^n \phi_{q,p}(x_p) \right)$
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 x nonlinear, fixed linear, learnable	(d)  KAN(x) Φ_3 Φ_2 Φ_1 x nonlinear, learnable

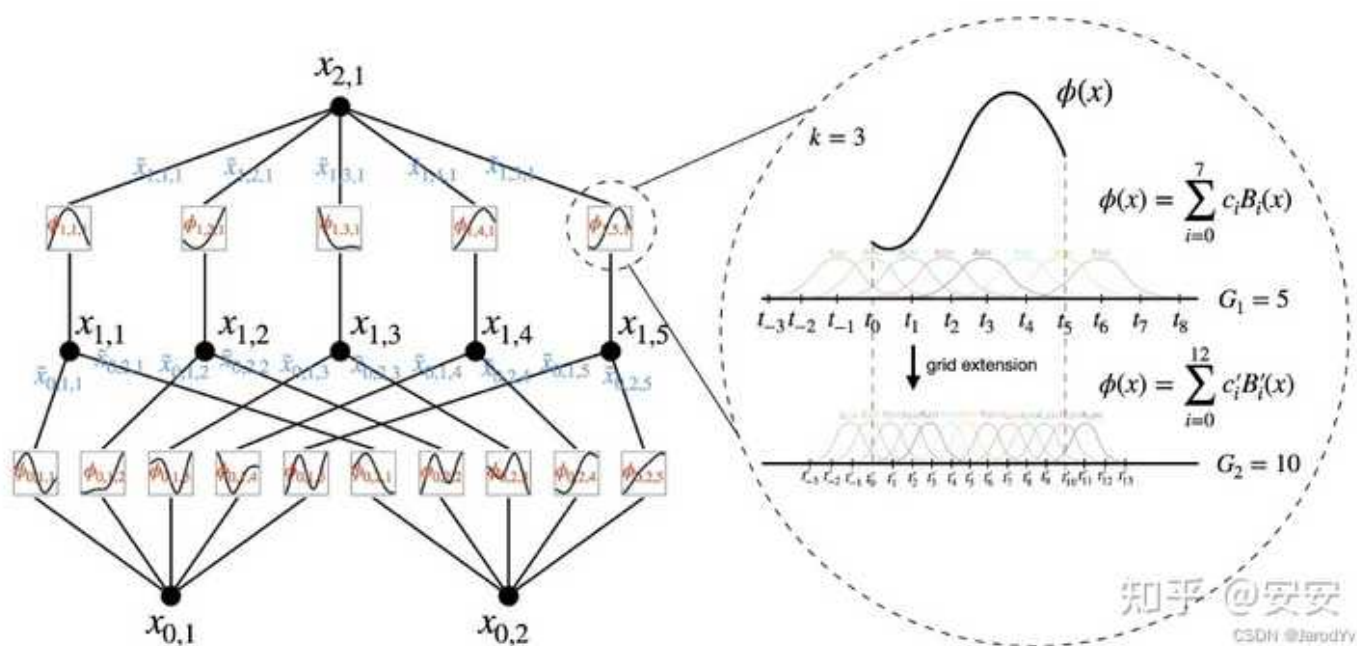
KAN 与 MLP 的对比

KAN 如何摆脱维度诅咒？

该定理允许将复杂的高维函数分解为简单的一维函数组合。通过集中优化这些一维函数而非整个多变量空间，KAN 大幅降低了实现精确建模所需的复杂性和参数数量。此外，由于这些函数较为简单，KAN 也因此成为了模型简单且易于解释的代表。

什么是 Kolmogorov-Arnold Networks (KAN)?

Kolmogorov-Arnold Networks（简称 KAN）是一种新型神经网络架构，灵感来源于 Kolmogorov-Arnold 表示定理。与传统使用固定激活函数的神经网络不同，KAN 在网络的边缘采用可学习的激活函数。这种设计使得 KAN 中每个权重参数都可以被一个单变量函数替换，这些函数通常以样条函数形式参数化，从而提供了极高的灵活性，并能够用更少的参数来模拟复杂的函数，增强了模型的可解释性。



KAN 兼具 MLP 结构和样条的优势

KAN 架构

Kolmogorov-Arnold Networks (KAN) 的架构围绕一个创新的概念：传统的权重参数在网络的边缘被单变量函数参数所取代。在 KAN 中，每个节点汇总这些函数的输出时不进行任何非线性变换，这与 MLP 中的做法（线性变换后跟非线性激活函数）形成了鲜明对比。

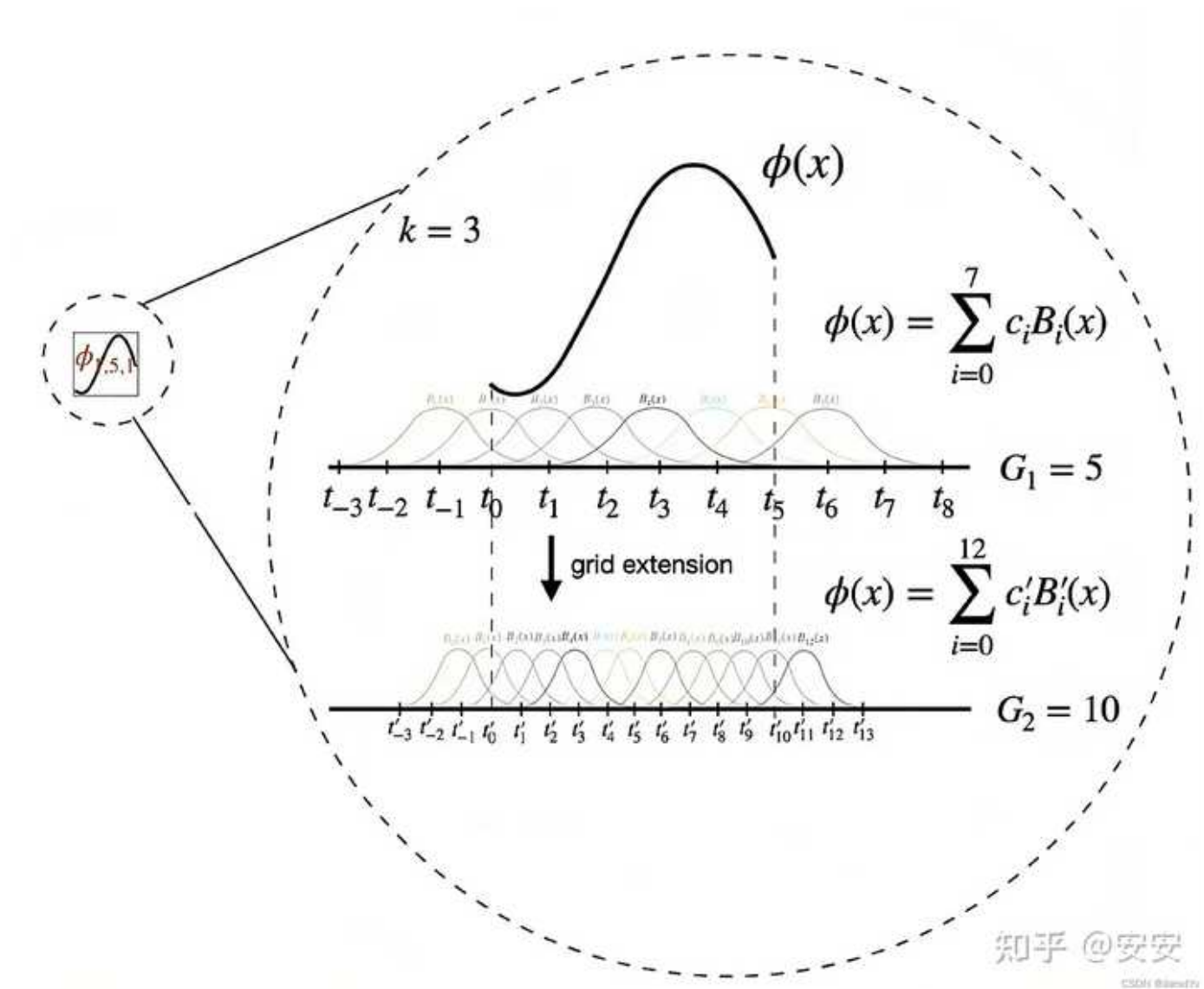
$$\begin{aligned}\text{KAN}(\mathbf{x}) &= (\Phi_{L-1} \circ \Phi_{L-1} \circ \dots \circ \Phi_1 \circ \Phi_0) \\ \text{MLP}(\mathbf{x}) &= (W_{L-1} \circ \sigma \circ W_{L-1} \circ \sigma \dots \circ W_1 \circ \sigma \circ W_0 \circ \sigma)\end{aligned}$$

KAN vs. MLP 公式

其中 MLP 中，W 表示线性权重参数， σ 表示非线性激活函数。

B-样条：KAN 的核心

论文中一个容易被忽略的重要部分是对样条的描述。样条是 KAN 学习机制的核心，它们取代了神经网络中通常使用的传统权重参数。



样条结构细节

样条的灵活性使其能够通过调整其形状来适应性地建模数据中的复杂关系，从而最小化近似误差，增强了网络从高维数据集中学习细微模式的能力。

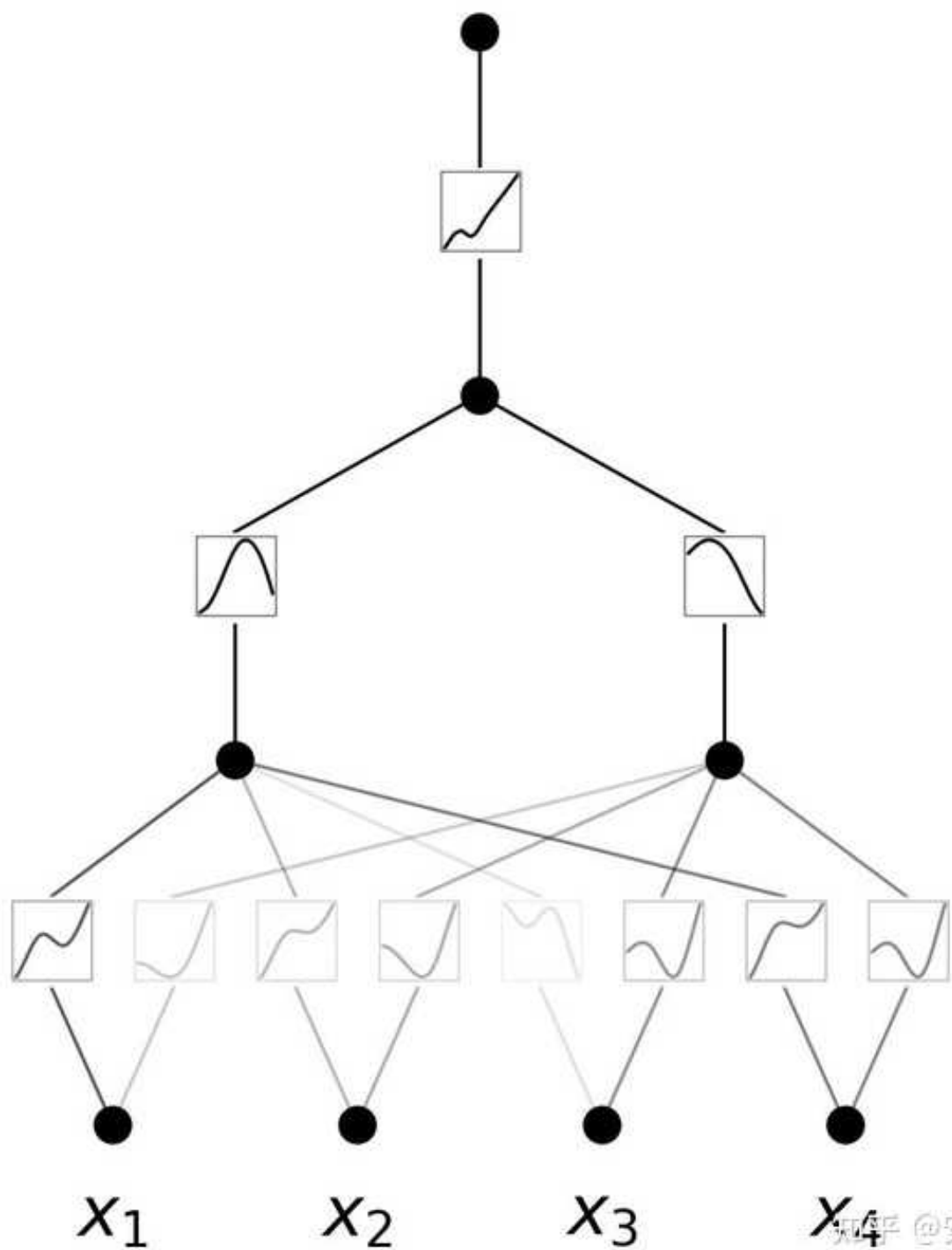
KAN 中样条的通用公式可以用 B-样条来表示：

$$spline(x) = \sum_i c_i B_i(x)$$

这里， $spline(x)$ 表示样条函数。 c_i 是训练期间优化的系数，而 $B_i(x)$ 是定义在网格上的 B-样条基函数。网格点定义了每个基函数 B_i 活跃并显著影响形状和平滑度的区间。你可以将它们视为影响网络准确性的超参数。更多的网格意味着更多的控制和更高的精度，同时也意味着需要学习更多的参数。

Step 0

$$\exp(\sin(x_1^2 + x_2^2) + \sin(x_3^2 + x_4^2))$$



通过多个步骤训练 KAN (来源: GitHub)

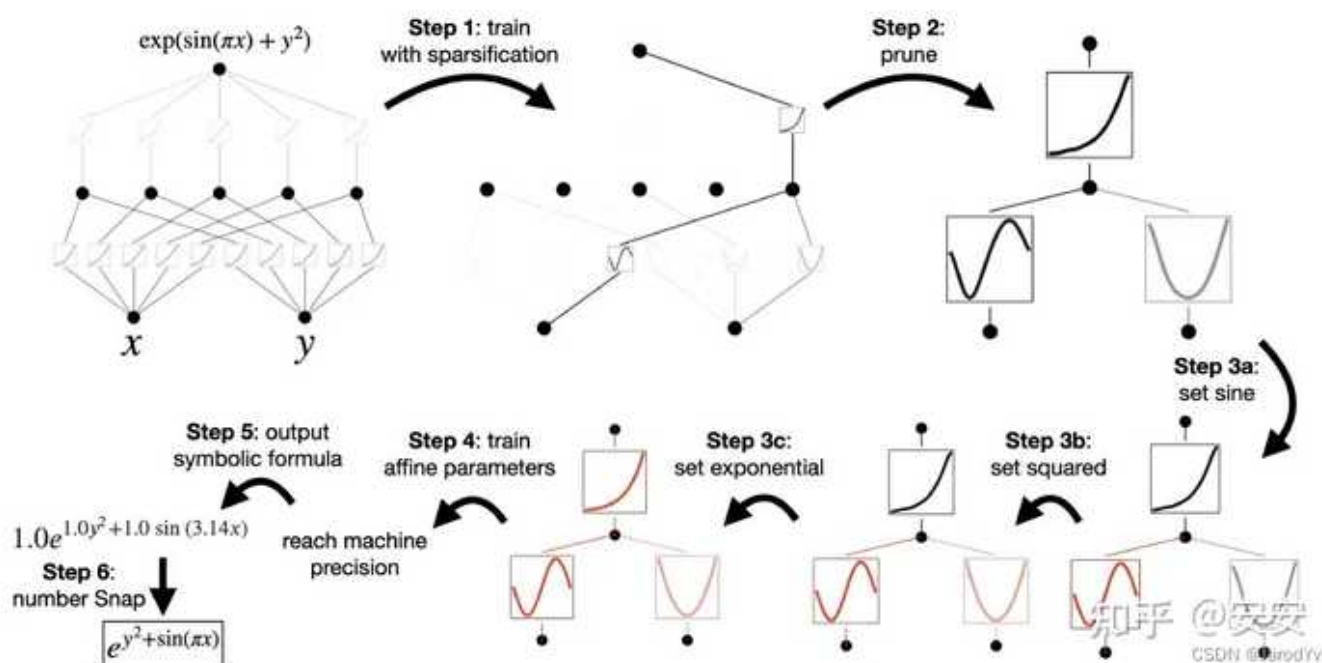
在训练期间，这些样条的 c_i 参数（基函数 $B_i(x)$ 的系数）通过优化以最小化损失函数，从而调整样条的形状以最佳地适应训练数据。这种优化通常涉及梯度下降等技术，每次迭代都会更新样条参数以减少预测误差。

结合两者的优势

虽然 KAN 基于 Kolmogorov-Arnold 表示定理，但它也从 MLP 中汲取灵感，结合了各自的优势，避免了各自的缺点。KAN 在外部结构上借鉴了 MLP，而在内部则采用了样条。

因此，KAN 不仅能学习特征（得益于与 MLP 的外部相似性），还能通过优化这些学到的特征达到更高的精度（得益于与样条的内部相似性）。

网络简化



网络符号概述

论文进一步解释了一些网络简化的方法。我将只介绍其中两种我认为非常有趣的方法。

符号化： KAN 通过使用组合的更简单、通常可解释的函数来近似函数，从而能够提供可解释的数学公式，如上图所示。

修剪： 论文中还讨论了通过在网络训练后移除不太重要的节点或连接来优化网络架构的另一个方面。这个过程有助于降低复杂性和规模。修剪侧重于识别并消除对输出贡献最小的网络部分，使网络更加轻便，也可能更易于解释。

KAN 的创新性

Kolmogorov-Arnold 表示定理本身并不是新的，那么为什么之前没有在机器学习中广泛使用它呢？正如论文所解释的：

虽然已经进行了多次尝试.....然而，大多数研究都坚持使用原始的深度 -2 ，宽度 $-(2n + 1)$ 表示，并没有机会利用更现代的技术（例如，反向传播）来训练网络。

这篇论文的创新之处在于，它调整了这一理念并将其应用于当前的机器学习环境。通过使用任意的网络架构（深度、宽度）并采用反向传播和修剪等技术，KAN 比以往的研究更接近实际应用。

在机器学习中，Kolmogorov-Arnold 表示定理基本上被视为理论上可行但实际上无用。

因此，尽管之前已经有尝试在机器学习中使用 Kolmogorov-Arnold 表示定理，但可以说 KAN 提供了一种全新的方法，因为它考虑到了机器学习目前的状况。这是对之前有限探索的一个重要更新。

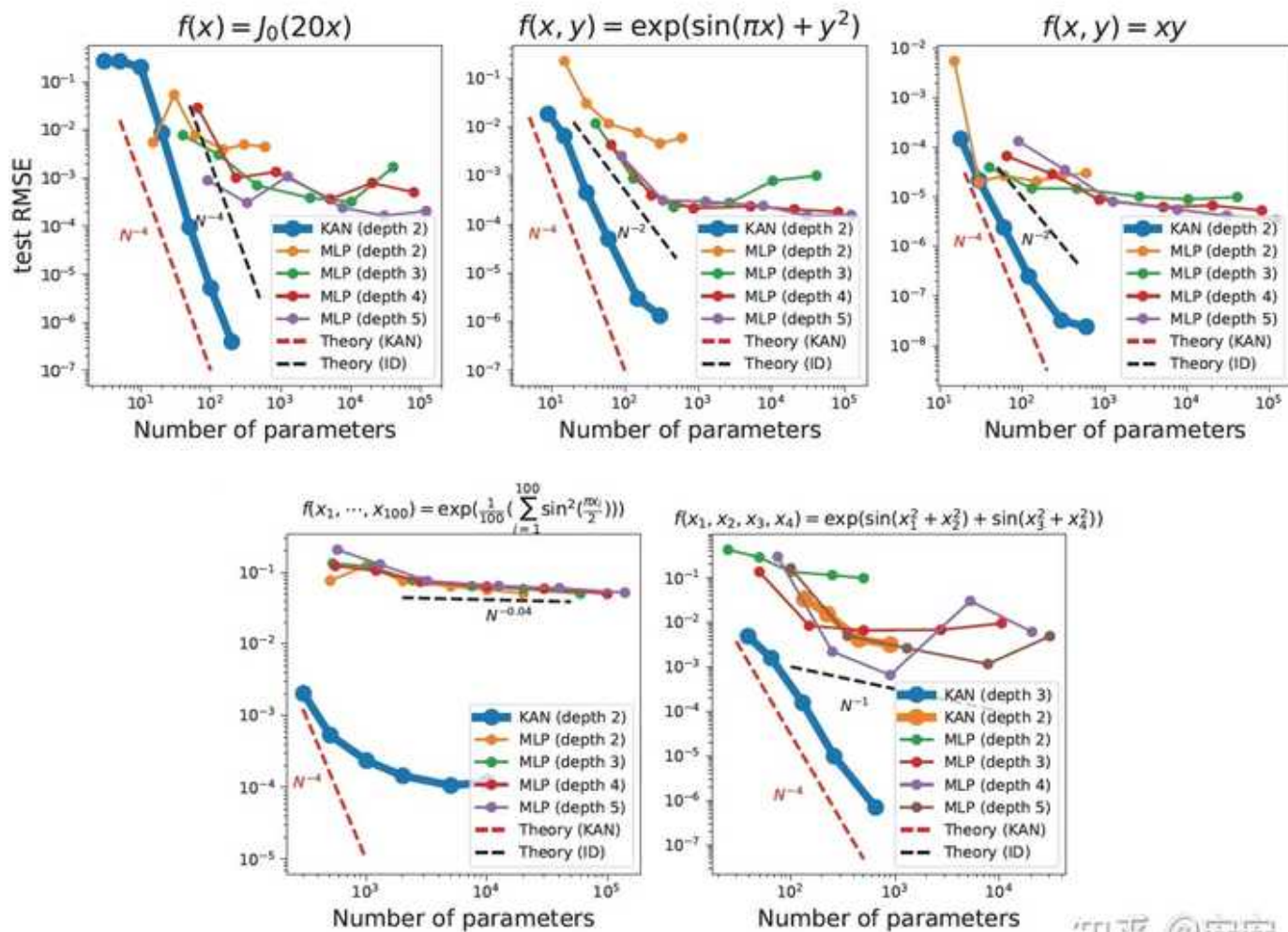
四个引人入胜的例子

论文比较了 KAN 和 MLP 在多个方面的性能，其中大多数都非常引人注目。在这一部分，我将列举一些特别有趣的例子。这些例子的详细内容和更多信息可以在论文中找到。

拟合符号公式

这是一个例子，展示了训练不同的 MLP 和一个 KAN 来拟合不同输入维度的函数。从下面的描述可以看出，与 MLP 相比，KAN 在这个参数数量范围内具有更好的扩展性。

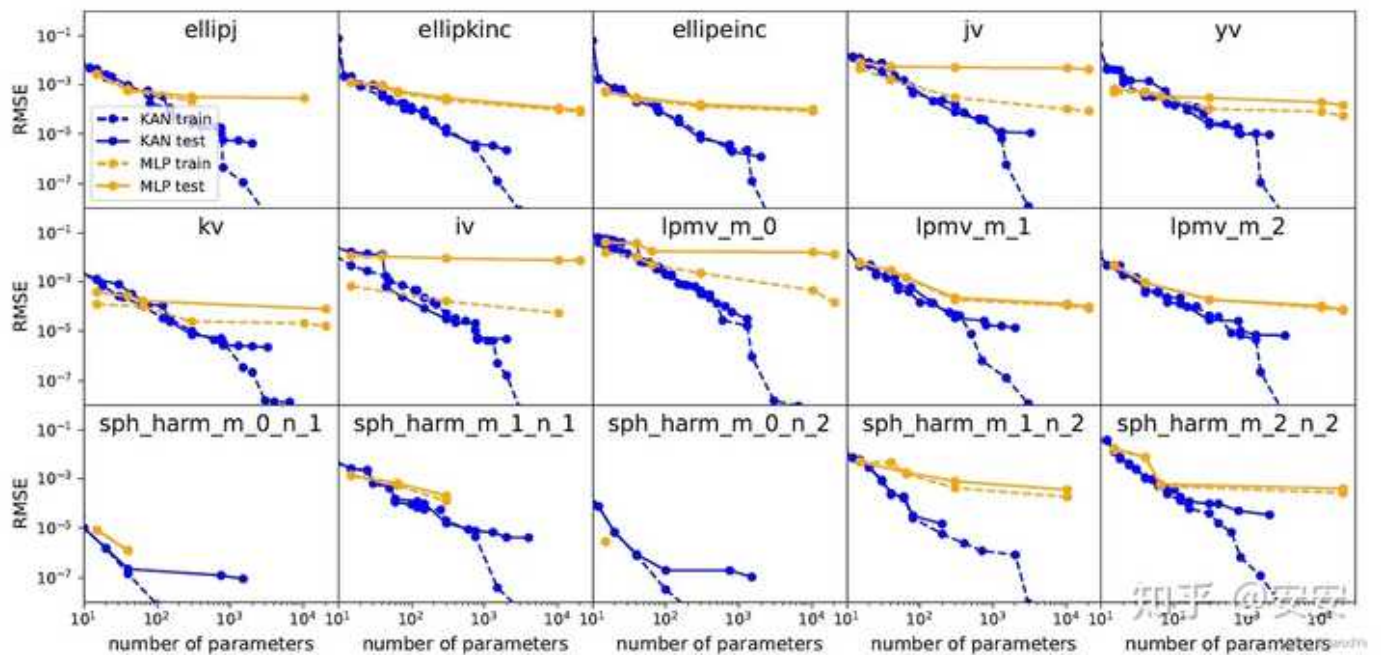
这突出了更深层次的 KAN 拥有更大的表达能力，这一点对于 MLP 也是如此：更深的 MLP 比较浅的具有更强的表达能力。



知乎 @安安

特殊函数

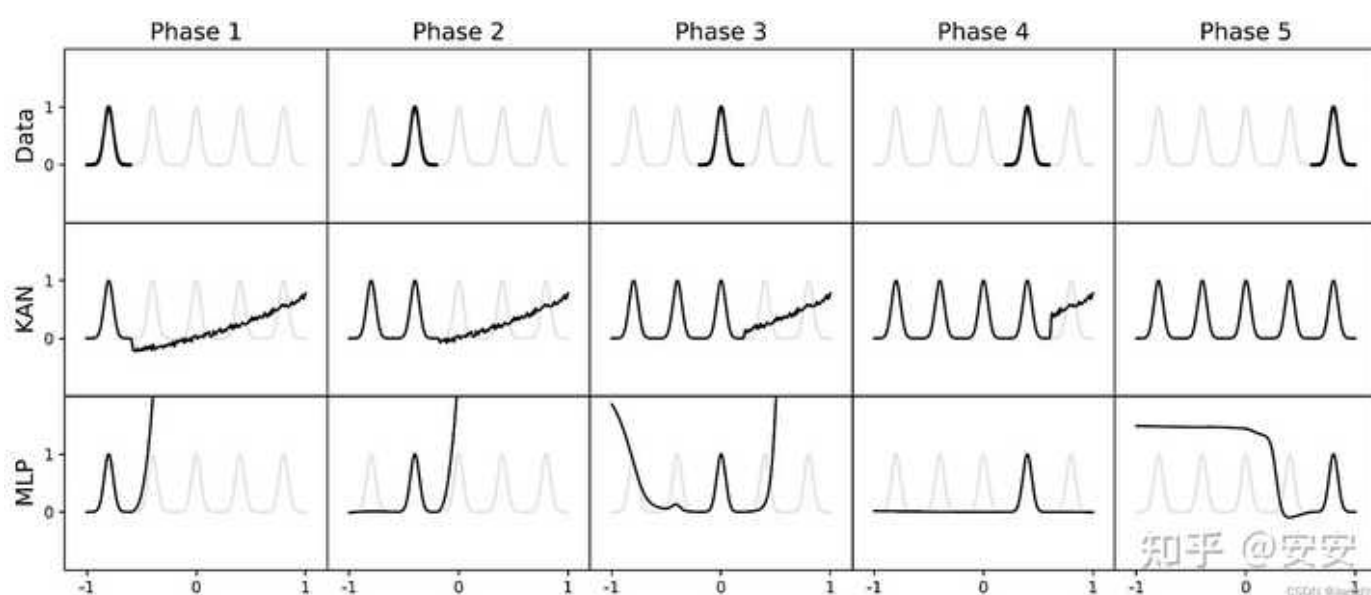
论文中的另一个例子是比较 KAN 和 MLP 在拟合 15 个数学和物理中常用的特殊函数上的表现。结果显示，在几乎所有这些函数中，KAN 在具有相同数量的参数的情况下，实现了更低的训练/测试损失。



持续学习

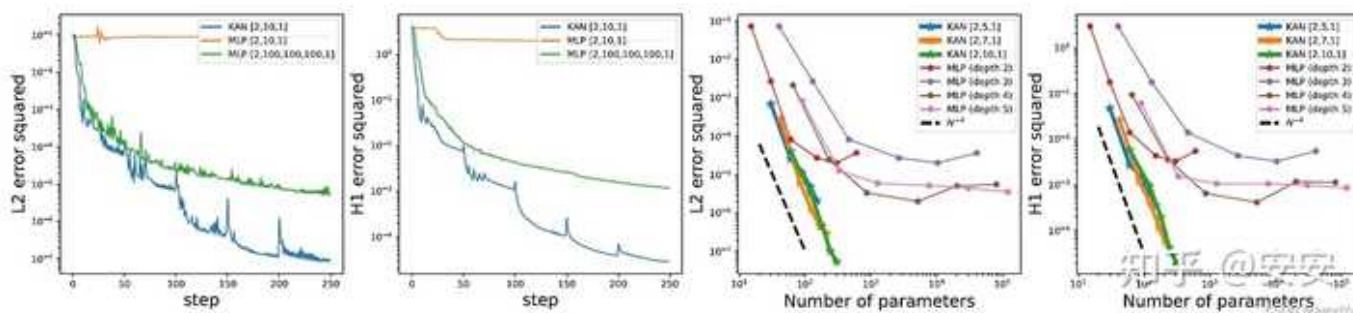
持续学习是指网络如何随时间适应新信息而不忘记已经学到的知识。这在神经网络训练中是一个重大挑战，特别是在避免所谓的灾难性遗忘问题上，后者指的是新知识的获取导致之前的信息迅速丧失。

得益于样条函数的局部性，KAN 展示了保持已学习信息并适应新数据的能力，而不会发生灾难性遗忘。与依赖全局激活的 MLP 不同，后者可能无意中影响模型远处的部分，KAN 仅修改每个新样本附近的有限组样条系数。这种集中调整保持了样条其他部分中存储的先前信息。



偏微分方程求解

在偏微分方程求解的应用中，一个 2 层宽度 ~ 10 的 KAN 的准确度比层宽度 ~ 100100100 的 MLP 高 100 倍 (10^{-7} vs 10^{-5} MSE)，并且在参数效率上也高出 100 倍 (10^2 vs 10^4 参数)。



论文还展示了更多的实验。其中一个实验涉及将 KAN 应用于几何扭结不变量 (geometric knot invariant) 问题，实现了 81.6% 的测试准确率，而 Google Deepmind 开发的一个 MLP 模型实现了 78% 的准确率，拥有约 3×10^5 参数。

最后的思考

KAN 的热议是否值得？

这取决于你的视角。之所以广泛讨论 KAN，是因为它被视为机器学习领域的一线希望。我在之前的文章中讨论了我们需要新的创新来帮助我们克服机器学习未来的障碍，特别是在数据和计算方面。尽管 KAN 并非刻意为此设计，但它可能成为一种解决方案。

KAN 主要针对 AI 在科学应用中的使用而设计，但现在已有人将其用于混合各种机器学习技术，包括多头注意力机制。

KAN + 大语言模型

由于 KAN 能够有效地建模和发现复杂的科学规律和模式，因此特别适用于 AI + 科学应用。KAN 特别适合那些需要理解和解释基础原理的任务，因为它们的结构允许将函数分解为符号数学表达式。这使得它们非常适合科学研究，这些研究需要发现这类关系，而不像大语言模型（LLM）那样，主要任务通常涉及处理庞大的数据集以进行自然语言理解和生成。

结论

从我的角度来看，最好的做法是根据 KAN 本身的特性来评价它，而不是基于我们希望它成为什么。这并不意味着将 KAN 整合入大语言模型是不可能的，事实上，已经有了一个高效的 PyTorch 实现的 KAN。但必须注意，现在称 KAN 为革命性或改变游戏规则还为时尚早。KAN 还需要更多由社区专家进行的实验。

尽管 KAN 在特定环境下提供了显著的优势，但它们也带来了一些限制和需要谨慎考虑的因素：

1. 复杂性和过拟合：KAN 可能会过拟合，特别是在数据有限的情况下。它们形成复杂模型的能力可能会将噪声误认为是重要模式，导致泛化能力差。
2. 计算需求：由于其专门化的性质，KAN 可能在 GPU 优化方面面临挑战，这可能会影响并行处理的效率。这种架构可能导致 GPU 上的操作速度较慢，需要进行序列化处理，从而导致内存利用效率低下，可能使 CPU 成为这些网络的更合适的平台。
3. 适用性：KAN 主要设计用于科学和工程任务，其中理解底层函数至关重要。它们可能不那么适用于需要大规模模式识别或分类的领域，如图像识别或自然语言处理，这些领域可能更适合使用更简单或更抽象的模型。

原文链接：

[深入理解 Kolmogorov-Arnold Networks \(KAN\)](#)