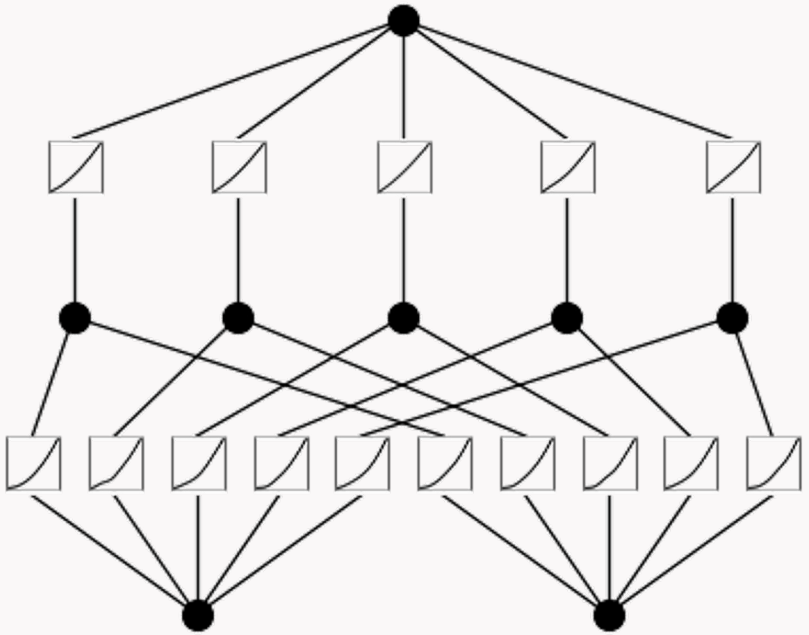
Kolmogorov-Arnold Network (KAN)

关键词
取代MLP?
精度更高，参数更少
可解释性
可训练的激活函数
有望解决灾难性遗忘问题
训练时间更长，训练难度更高



Universal Approximation Theorem（通用近似定理）

使$C(X, \mathbb{R}^m)$表示一个从 $X \in \mathbb{R}^n$ 到 $\mathbb{R}^m$ 连续函数集合,令$\sigma \in C(\mathbb{R}, \mathbb{R})$，且$(\sigma \circ x)_i = \sigma(x_i)$,即$\sigma \circ x$表示将$\sigma$应用于 x的每个分量。

那么当σ非多项式时,当且仅当对于所有$n \in \mathbb{N}, m \in \mathbb{N}$, K是在$\mathbb{R}^n$ 上的紧子集, $\mathrm{f} \in C(K, \mathbb{R}^m)$,$\varepsilon > 0$,存在 $\mathrm{k} \in \mathbb{N}, W \in \mathbb{R}^{k \times n}$, $b \in \mathbb{R}^k$, $C \in \mathbb{R}^{k \times n}$,使得:

$$\sup_{x \in K} \|f(x) - g(x)\| < \varepsilon$$

其中 $g(x) = C \cdot (\sigma \circ (W \cdot x + b))$

Universal Approximation Theorem（通用近似定理）

使$C(X, \mathbb{R}^m)$表示一个从 $X \in \mathbb{R}^n$ 到 $\mathbb{R}^m$ 连续函数集合,令$\sigma \in C(\mathbb{R}, \mathbb{R})$，且$(\sigma \circ x)_i = \sigma(x_i)$,即$\sigma \circ x$表示将$\sigma$应用于 x的每个分量。

那么当σ非多项式时,当且仅当对于所有$n \in \mathbb{N}, m \in \mathbb{N}$, K是在$\mathbb{R}^n$ 上的紧子集, $\mathrm{f} \in C(K, \mathbb{R}^m)$,$\varepsilon > 0$,存在 $\mathrm{k} \in \mathbb{N}, W \in \mathbb{R}^{k \times n}$, $b \in \mathbb{R}^k$, $C \in \mathbb{R}^{k \times n}$,使得:

$$\sup_{x \in K} \|f(x) - g(x)\| < \varepsilon$$

其中 $g(x) = C \cdot (\sigma \circ (W \cdot x + b))$

缺点：可解释性差，参数量大

Kolmogorov-Arnold表示定理

如果 f 是在有界域上的多元连续函数，那么 f 可以写成有限数量的单变量连续函数和二元加法运算的组合。进一步来说，对于一个平滑的函数$f: [0,1]^n \to \mathbb{R}$,

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

其中$\phi_{q,p}: [0,1] \to \mathbb{R}$，以及$\Phi_q: \mathbb{R} \to \mathbb{R}$。

e.g.

$$f(x,y) = xy = \underbrace{\exp(\log(x+1) + \log(y+1))}_{\text{自变量为 } \log(x+1) + \log(y+1) \text{ 的一元函数}} - \underbrace{(x+0.5)}_{\text{一元函数}} - \underbrace{(y+0.5)}_{\text{一元函数}}$$

$\log(x+1)$：一元函数　$\log(y+1)$：一元函数

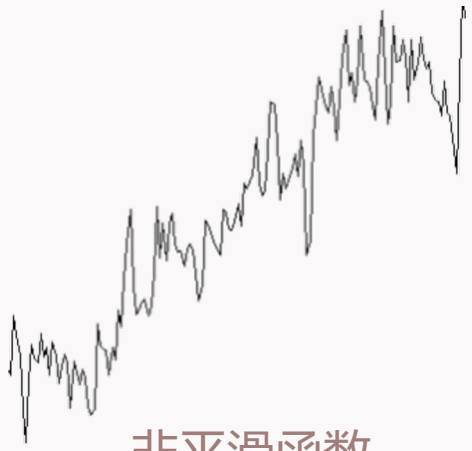
Kolmogorov-Arnold表示定理

如果 f 是在有界域上的多元连续函数，那么 f 可以写成有限数量的单变量连续函数和二元加法运算的组合。进一步来说，对于一个平滑的函数$f: [0,1]^n \to \mathbb{R}$,

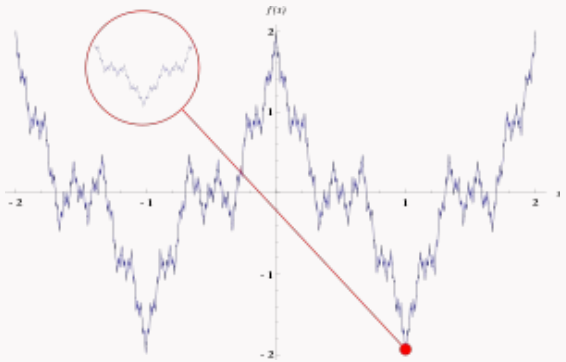
$$f(x) = f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}\left(x_p\right) \right)$$

其中内部函数$\phi_{q,p}: [0,1] \to \mathbb{R}$，以及外部函数$\Phi_q: \mathbb{R} \to \mathbb{R}$。

可能是



非平滑函数



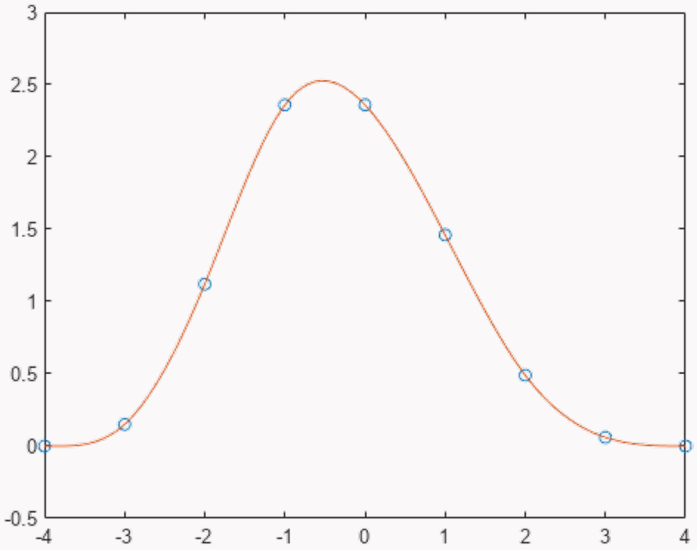
分形函数

不可学习的!

样条函数(Spline)

样条函数是由一些具有连续性条件的子空间上的分段多项式构成，给定$n+1$个点$t_0, \dots, t_n$并且满足$\mathrm{a} = t_0 < t_1 < \cdots < t_n = b$，这些点被称为结点(knot)，如果满足下列条件，参数曲线$S\colon [a,b] \to \mathbb{R}$被称为$k$次样条：

1. 在每个分段区间$[t_i, t_{i+1}]$上，S是一个次数小于等于k的多项式。
2. 在$[t_0, t_n]$上S有$k-1$阶连续导数。



优点：在低维空间中的数据插值和函数逼近中表现出色

缺点：维数灾难

样条函数(Spline)

样条函数是由一些具有连续性条件的子空间上的分段多项式构成，给定$n+1$个点$t_0, \dots, t_n$并且满足a $= t_0 < t_1 < \cdots < t_n = b$，这些点被称为结点(knot)，如果满足下列条件，参数曲线$S\colon [a,b] \to \mathbb{R}$被称为$k$次样条：

1. 在每个分段区间$[t_i, t_{i+1}]$上，S是一个次数小于等于k的多项式。
2. 在$[t_0, t_n]$上S有$k-1$阶连续导数。

B样条(B-Spline)可以用Cox-de Boor递推公式表达：

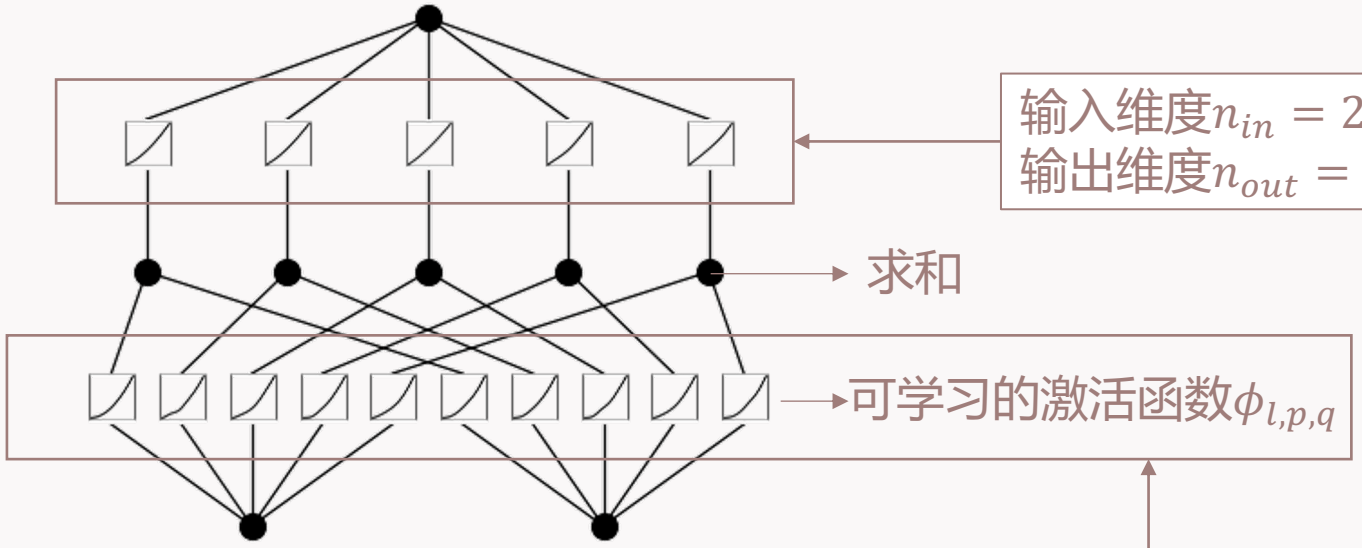
$$B_{i,0}(x) := \begin{cases} 1 & if\ t_i \le x < t_{i+1}, \\ 0 & otherwise. \end{cases}$$

$$B_{i,k}(x) := \frac{x - t_i}{t_{i+k} - t_i} B_{i,k-1}(x) + \frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x).$$

KAN架构

l	层数的索引
i	x的索引
p	输入维度n_in的索引
q	输出维度n_out的索引
n	输入x的维度

输出: $KAN(x) = (\Phi_{L-1} \circ \Phi_{L-2} \dots \circ \Phi_1 \circ \Phi_0)x$



输入维度$n_{in} = 2\mathrm{n} + 1$
输出维度$n_{out} = 1$

求和

可学习的激活函数$\phi_{l,p,q}$

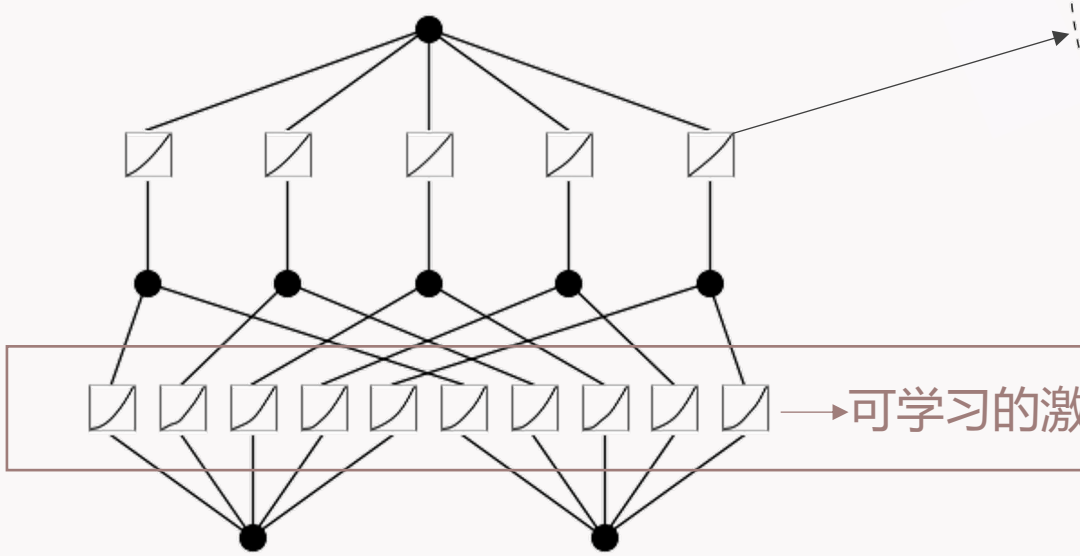
输入$x_{l,i}$

内部函数层$\Phi_l = \{\phi_{p,q}\} = \begin{bmatrix} \phi_{l,1,1}(\cdot) & \phi_{l,1,2}(\cdot) & \cdots & \phi_{l,1,n_l}(\cdot) \\ \phi_{l,2,1}(\cdot) & \phi_{l,2,2}(\cdot) & \cdots & \phi_{l,2,n_l}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{l,n_{l+1},1}(\cdot) & \phi_{l,n_{l+1},2}(\cdot) & \cdots & \phi_{l,n_{l+1},n}(\cdot) \end{bmatrix}$

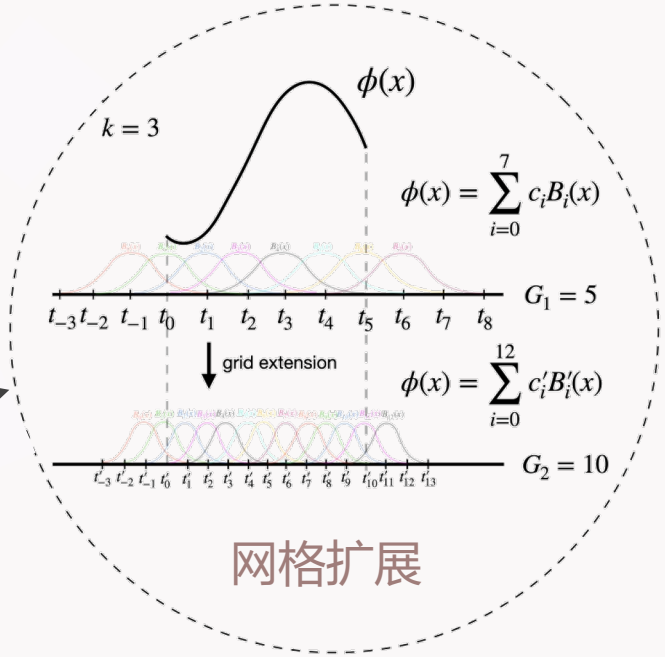
输入维度$n_{in} = \mathrm{n}$
输出维度$n_{out} = 2\mathrm{n} + 1$

KAN架构

输出: $KAN(x) = (\Phi_{L-1} \circ \Phi_{L-2} \dots \circ \Phi_1 \circ \Phi_0)x$



输入$x_{l,i}$



网格扩展

可学习的激活函数$\phi_{l,p,q}$

w使用Xavier初始化

残差激活函数:

$$\phi(x) = w(b(x) + spline(x))$$

$$b(x) = silu(x) = \frac{x}{1 + e^{-x}}$$

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

$spline(x)$被初始化为 $\approx 0^2$

KAN的可解释性

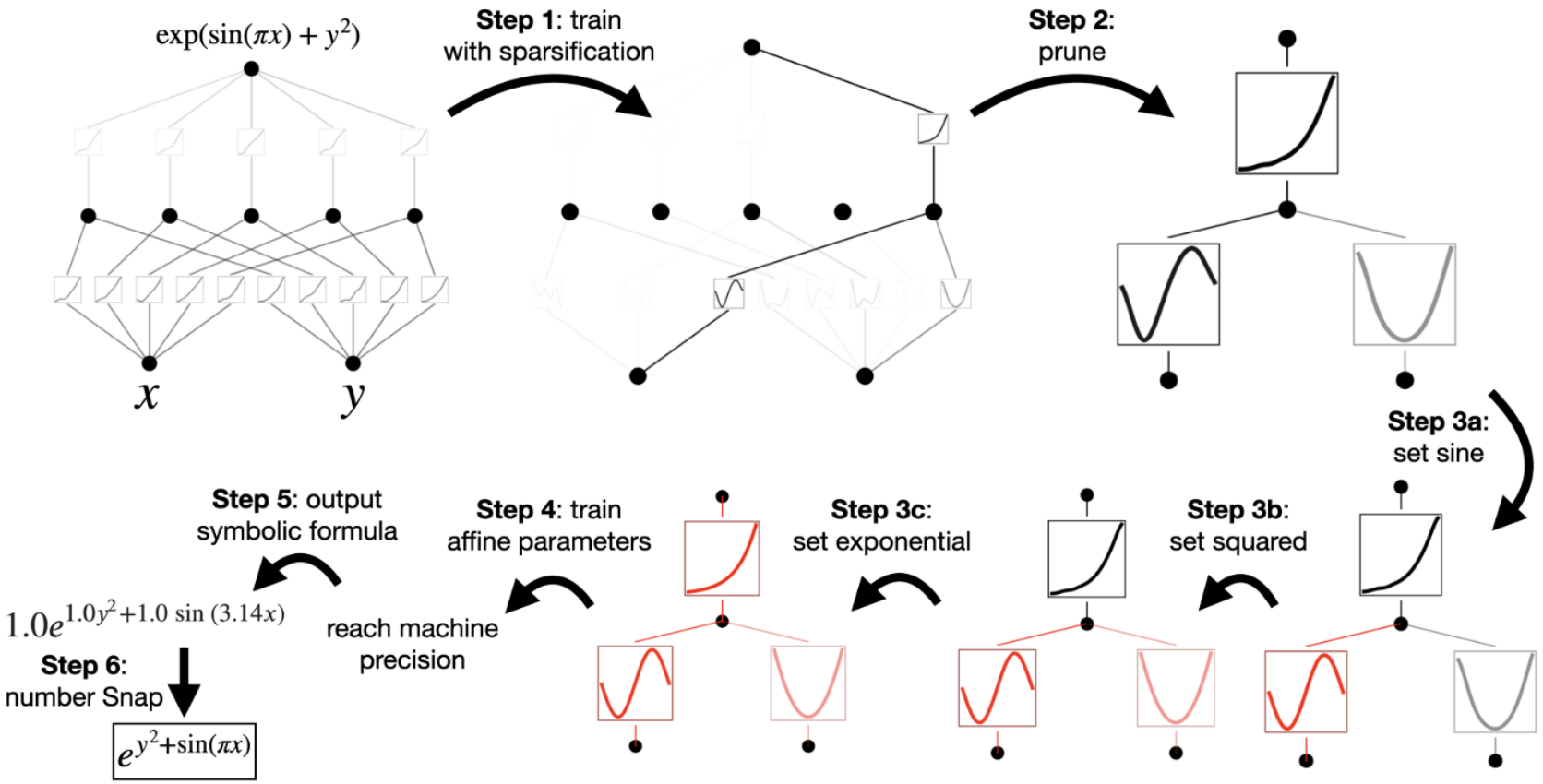


Figure 2.4: An example of how to do symbolic regression with KAN.

拟合特殊函数

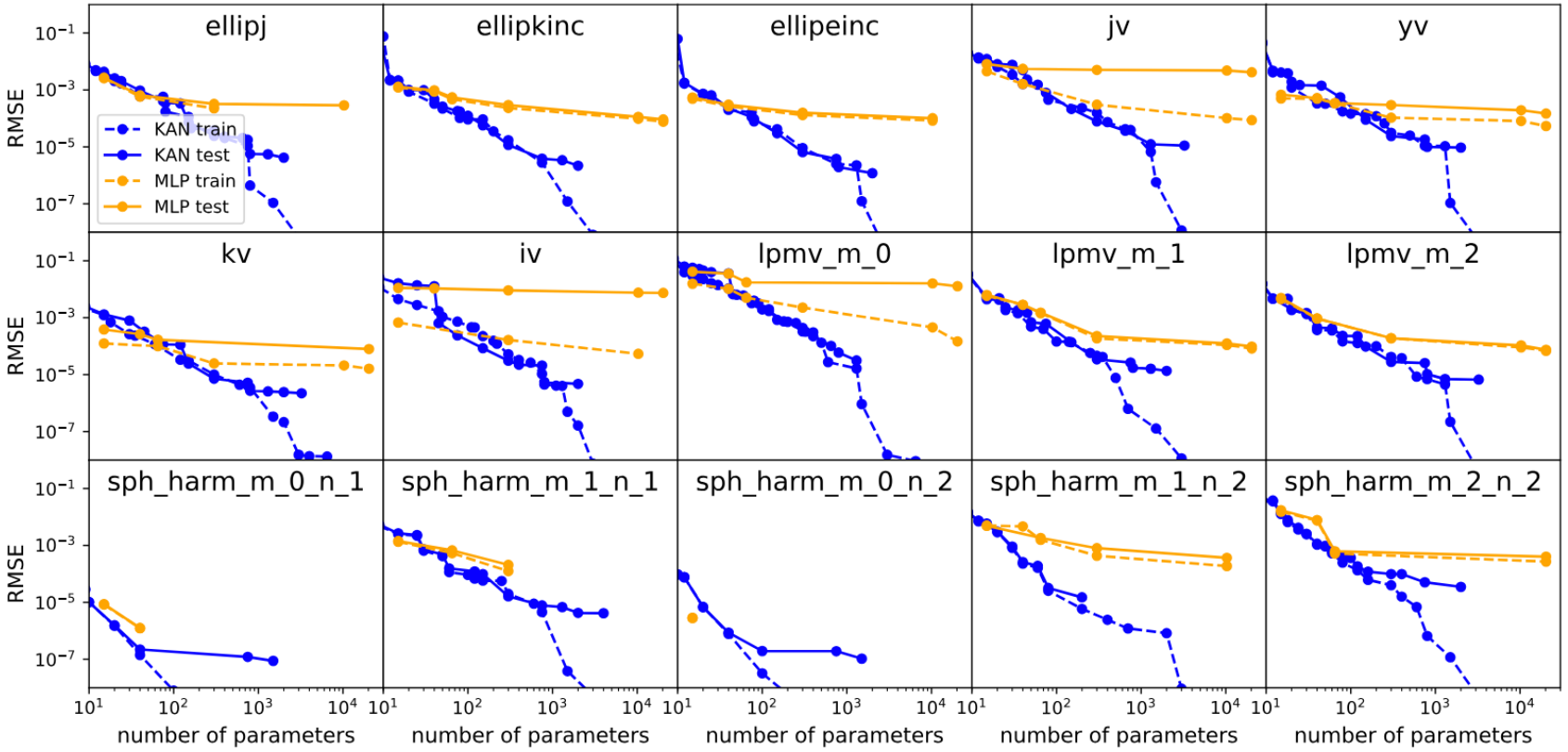


Figure 3.2: Fitting special functions. We show the Pareto Frontier of KANs and MLPs in the plane spanned by the number of model parameters and RMSE loss. Consistently accross all special functions, KANs have better Pareto Frontiers than MLPs. The definitions of these special functions are in Table 2.

解偏微分方程

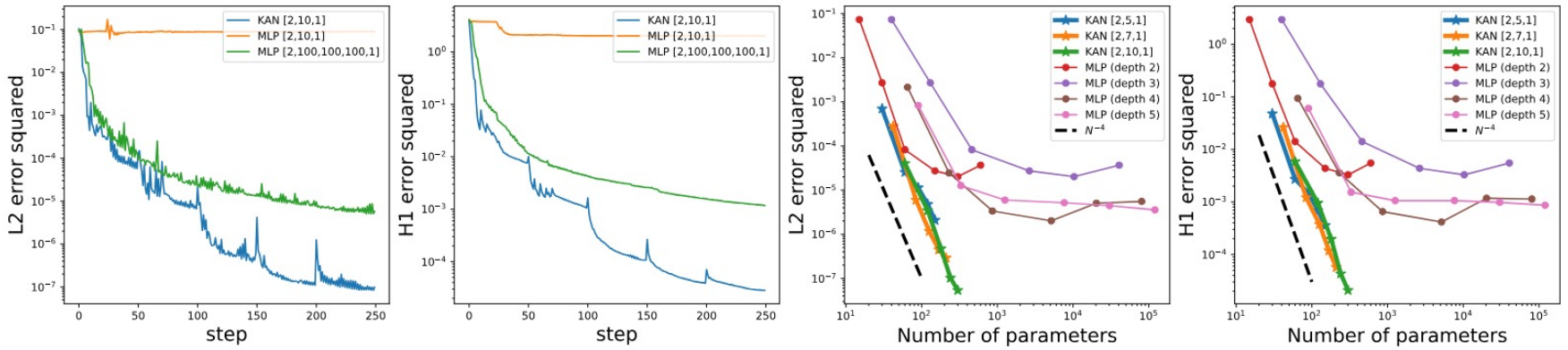


Figure 3.3: The PDE example. We plot L2 squared and H1 squared losses between the predicted solution and ground truth solution. First and second: training dynamics of losses. Third and fourth: scaling laws of losses against the number of parameters. KANs converge faster, achieve lower losses, and have steeper scaling laws than MLPs.

解决灾难性遗忘

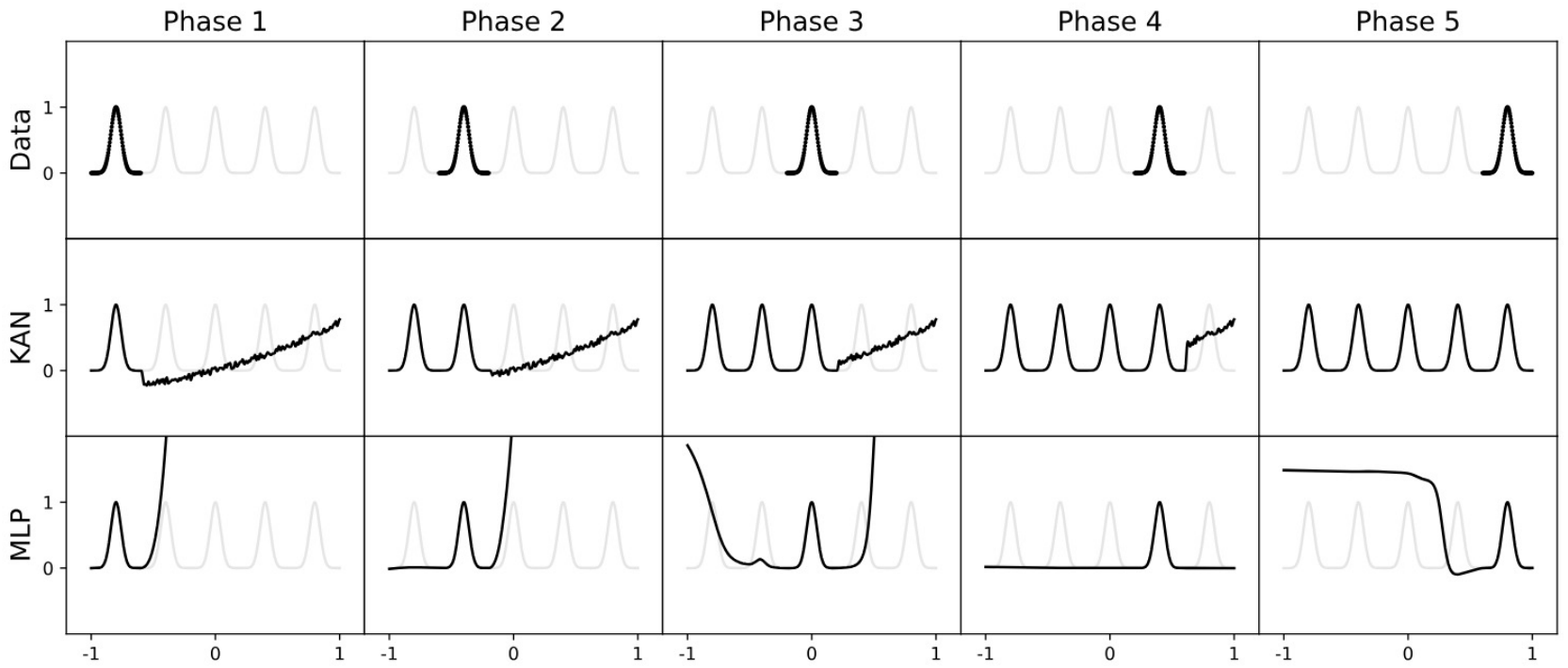


Figure 3.4: A toy continual learning problem. The dataset is a 1D regression task with 5 Gaussian peaks (top row). Data around each peak is presented sequentially (instead of all at once) to KANs and MLPs. KANs (middle row) can perfectly avoid catastrophic forgetting, while MLPs (bottom row) display severe catastrophic forgetting.

MLPs or KANs?

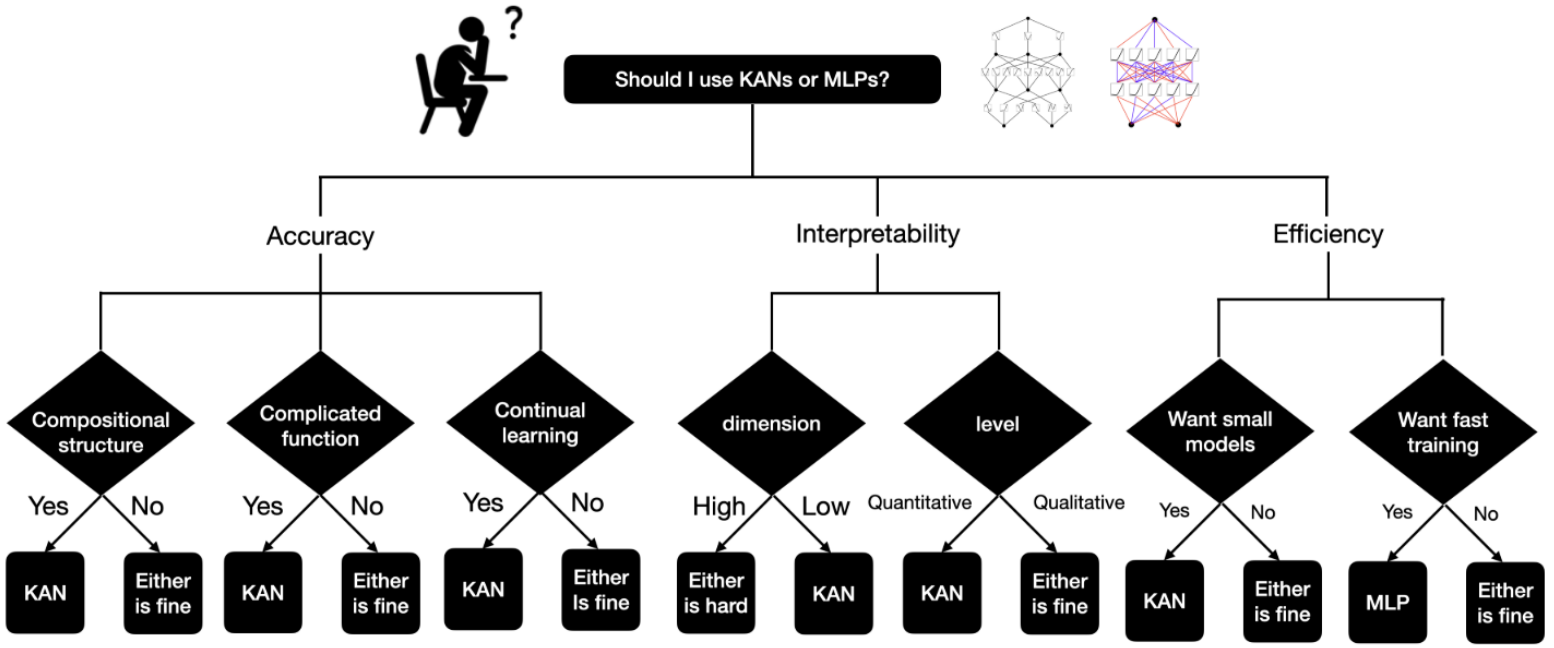


Figure 6.1: Should I use KANs or MLPs?