中国人民大學 RENMIN UNIVERSITY OF CHINA

高瓴人工智能学院 Gaoling School of Artificial Intelligence

Paper Sharing

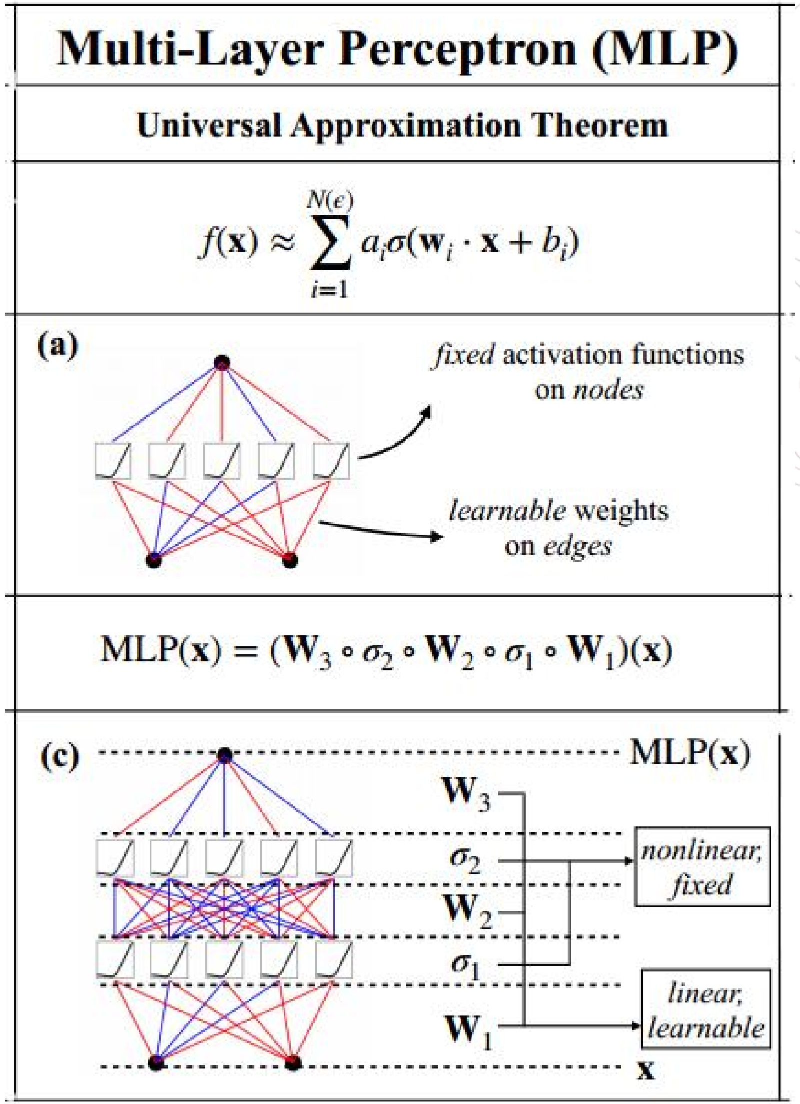
KAN：Kolmogorov–Arnold Networks

Lecturer：Yuxin Wu

2024.5.30

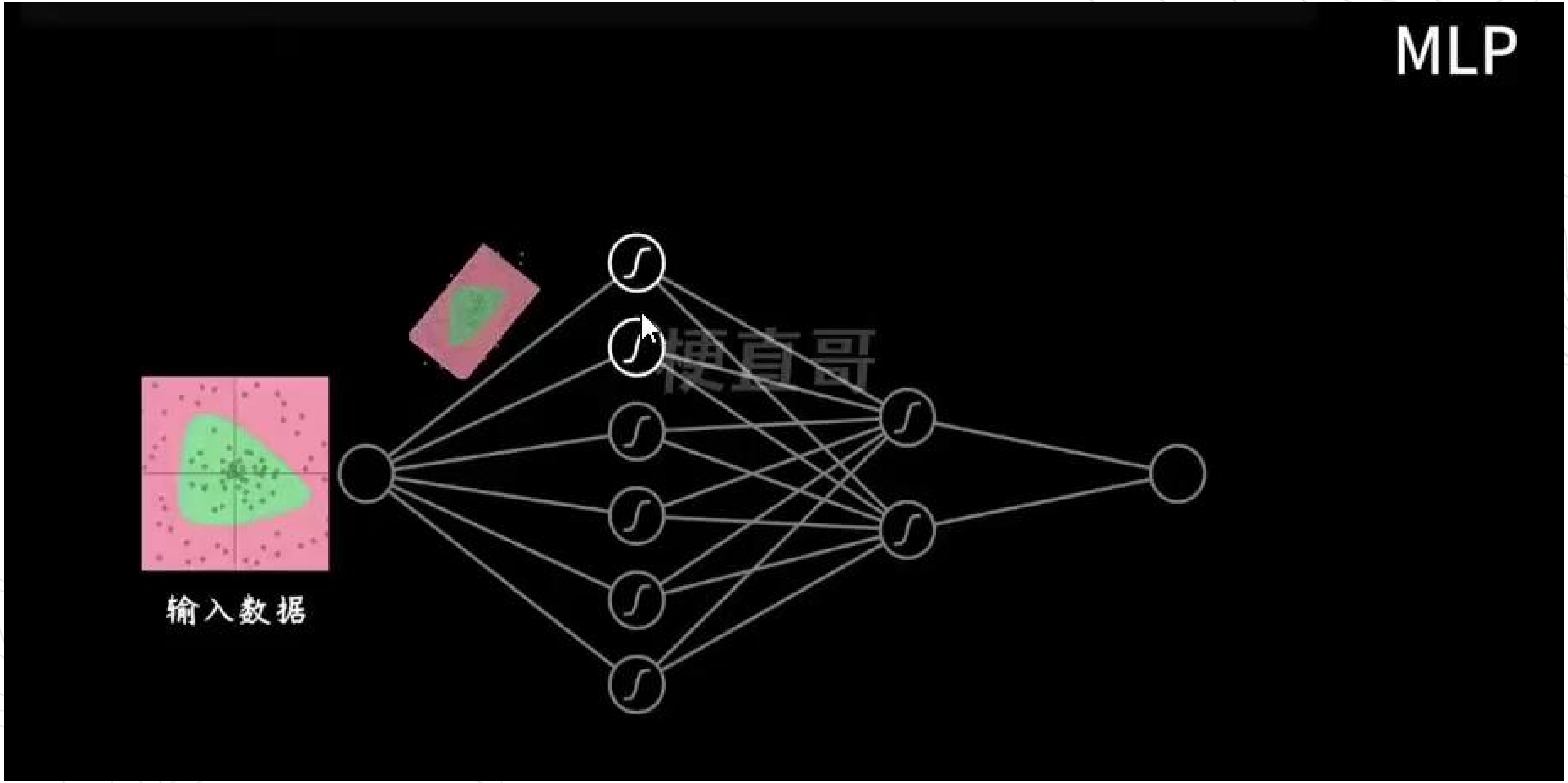
Review: MLP (Multilayer Perceptron)

- CNN
- Transformer
- LLM



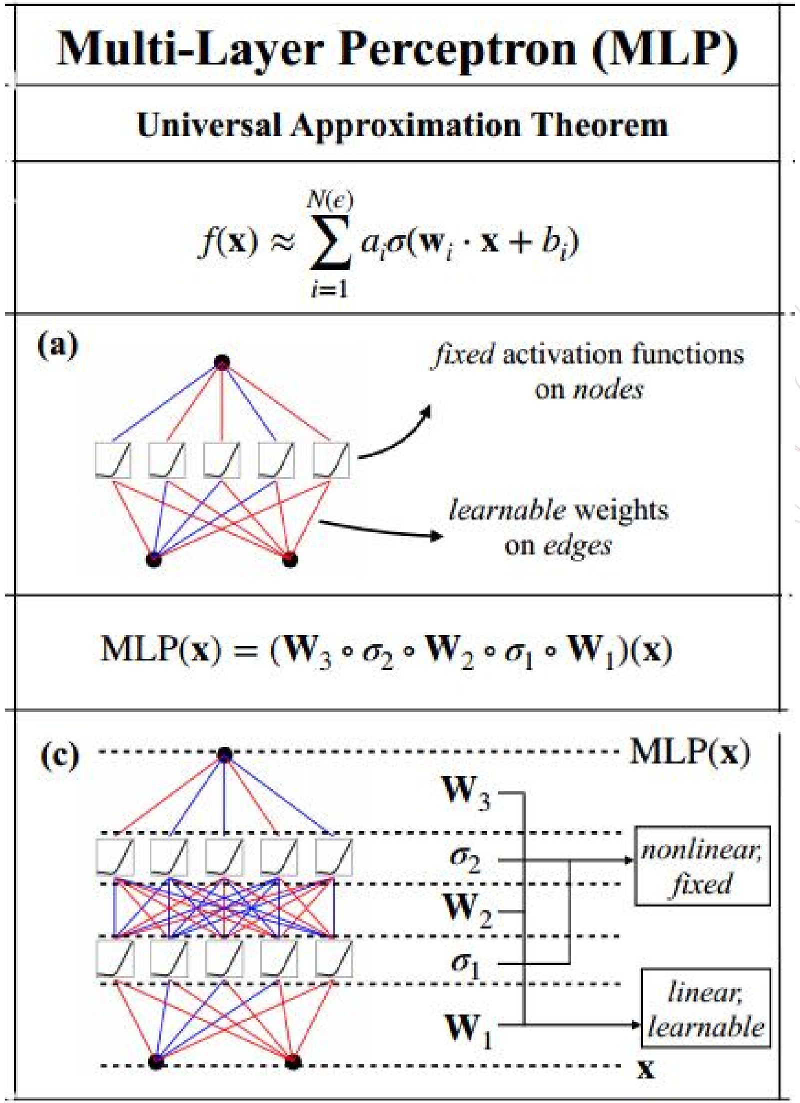
Review: MLP (Multilayer Perceptron)

Review: MLP (Multilayer Perceptron)



Review: MLP (Multilayer Perceptron)

- Gradient Vanishing/Exploding
- Low Parameter Efficiency
- Limited Ability to Process High-dimensional Data
- Unable to solve long-term dependencies



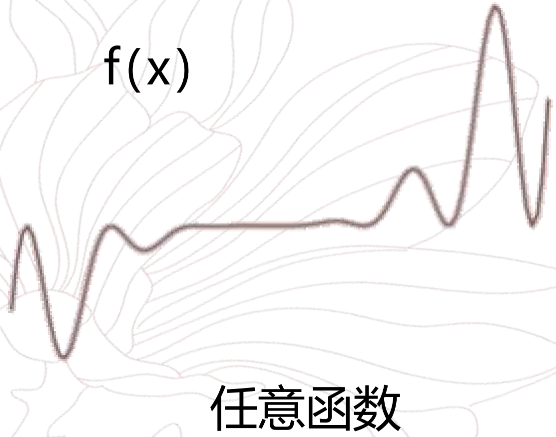
Review：Universal Approximation Theorem（通用近似定理）

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, $f\in C(K,\mathbb{R}^m),\varepsilon>0$,存在 $k\in\mathbb{N},W\in\mathbb{R}^{k\times n}$, $b\in\mathbb{R}^k$, $C\in\mathbb{R}^{k\times n}$,使得:

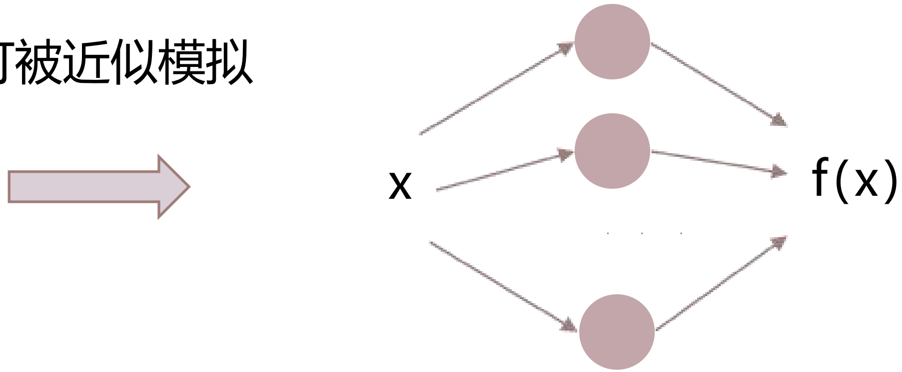
$$\sup_{x\in K}\|f(x)-g(x)\|<\varepsilon$$

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

通用近似定理，也称为万能近似定理，是人工神经网络领域的一个重要数学理论，它指出神经网络具备近似任意函数的能力。这个定理对于神经网络的设计和应用具有重要的指导意义。具体而言，通用近似定理表明，对于任意一个连续函数$f(x)$和任意一个正数ε，存在一个具有至少一个隐藏层的神经网络$g(x)$，使得对于所有的输入x，满足$|f(x)-g(x)|<\varepsilon$。换句话说，神经网络可以用来逼近任意连续函数，并且可以达到任意给定的精度要求。[1] [2]



可被近似模拟



Review: Kolmogorov-Arnold representation theorem (1957)

If f is a multivariate continuous function on a bounded field, then f can be written as a combination of a finite number of univariate continuous functions and binary addition operations

Kolmogorov-Arnold表示定理指出每个光滑的**多元**函数 $f:[0,1]^n \to \mathbb{R}$ 都可以被如下方式**逼近**:

$$f(x) = f(x_1, x_2, \ldots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \varphi_{q,p}(x_p) \right)$$

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

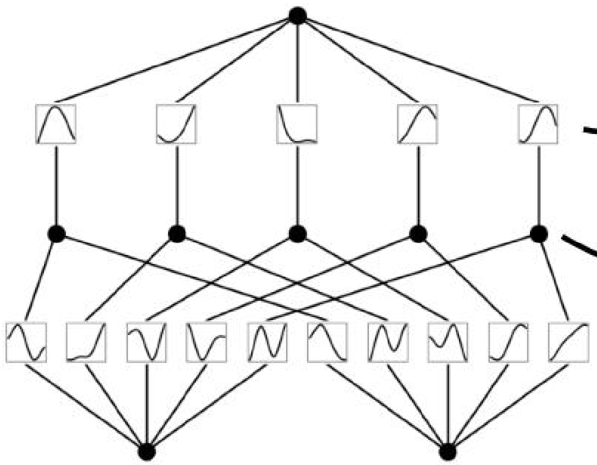
e.g.

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

Review: Kolmogorov-Arnold representation theorem (1957)

Kolmogorov-Arnold Representation Theorem

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



learnable activation functions on *edges*

sum operation on *nodes*

Formula (Deep)	$\mathrm{MLP}(\mathbf{x}) = (\mathbf{W}_3 \circ \sigma_2 \circ \mathbf{W}_2 \circ \sigma_1 \circ \mathbf{W}_1)(\mathbf{x})$	$\mathrm{KAN}(\mathbf{x}) = (\mathbf{\Phi}_3 \circ \mathbf{\Phi}_2 \circ \mathbf{\Phi}_1)(\mathbf{x})$

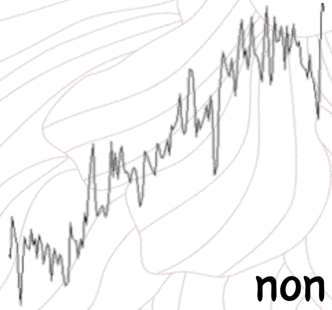
Review: Kolmogorov-Arnold representation theorem (1957)

If f is a multivariate continuous function on a bounded field, then f can be written as a combination of a finite number of univariate continuous functions and binary addition operations

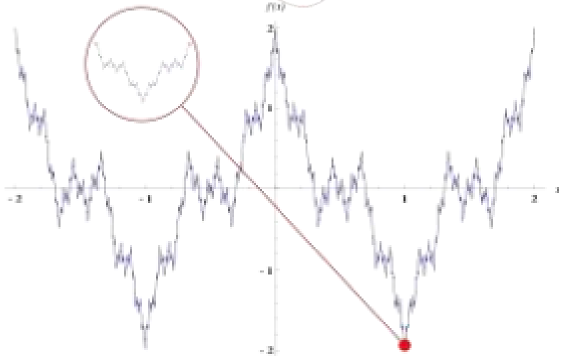
Kolmogorov-Arnold表示定理指出每个光滑的**多元**函数 $f:[0,1]^n \to \mathbb{R}$ 都可以被如下方式逼近:

$$f(x) = f(x_1, x_2, \ldots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \varphi_{q,p}(x_p) \right)$$

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



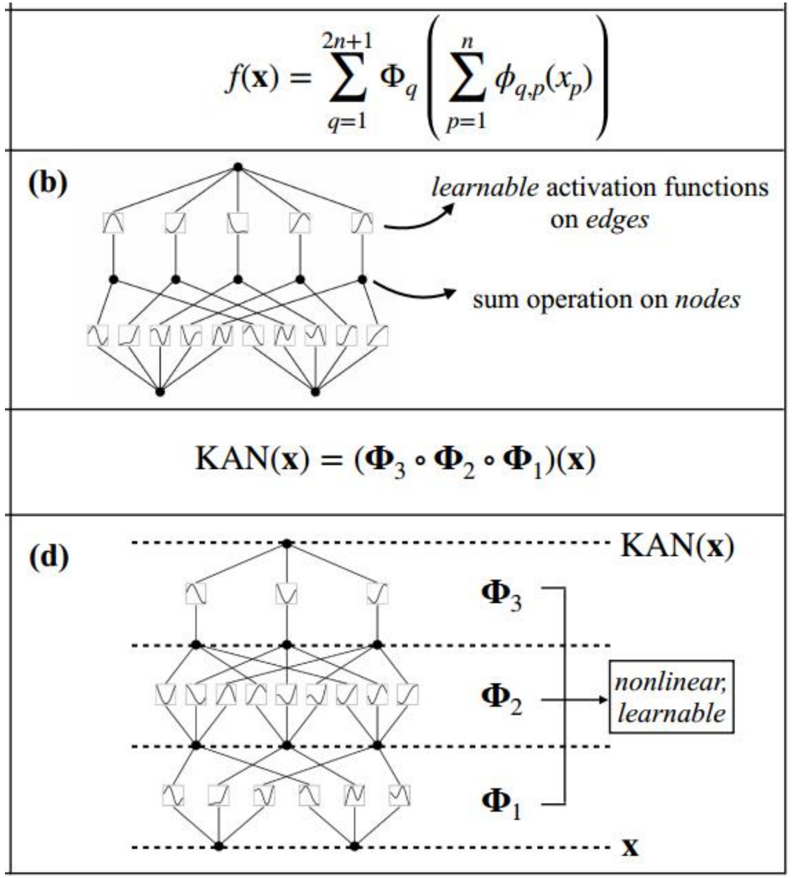
non-smooth



fractal

- 2 is sufficient
- what about 2+ ?

“Deeper” : Kolmogorov-Arnold Network



- "Activate、activate、and activate"
- Non-linear Representation Ability ↑

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).

B-Spline Function

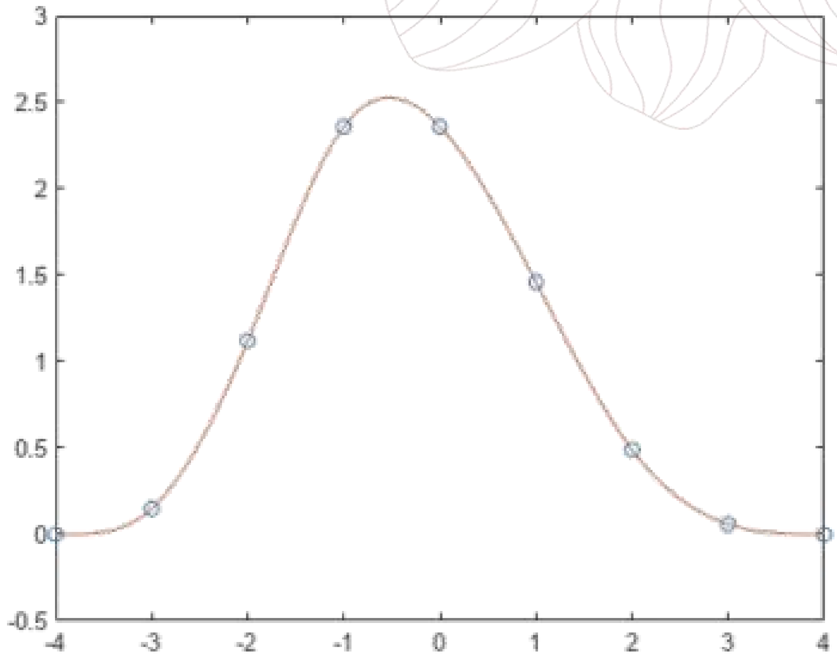
样条函数是由一些具有连续性条件的子空间上的分段多项式构成，给定$n+1$个点$t_0,\dots,t_n$并且满足$a=t_0<t_1<\cdots<t_n=b$，这些点被称为结点(knot)，如果满足下列条件，参数曲线$S:[a,b]\to\mathbb{R}$被称为k次样条：

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：

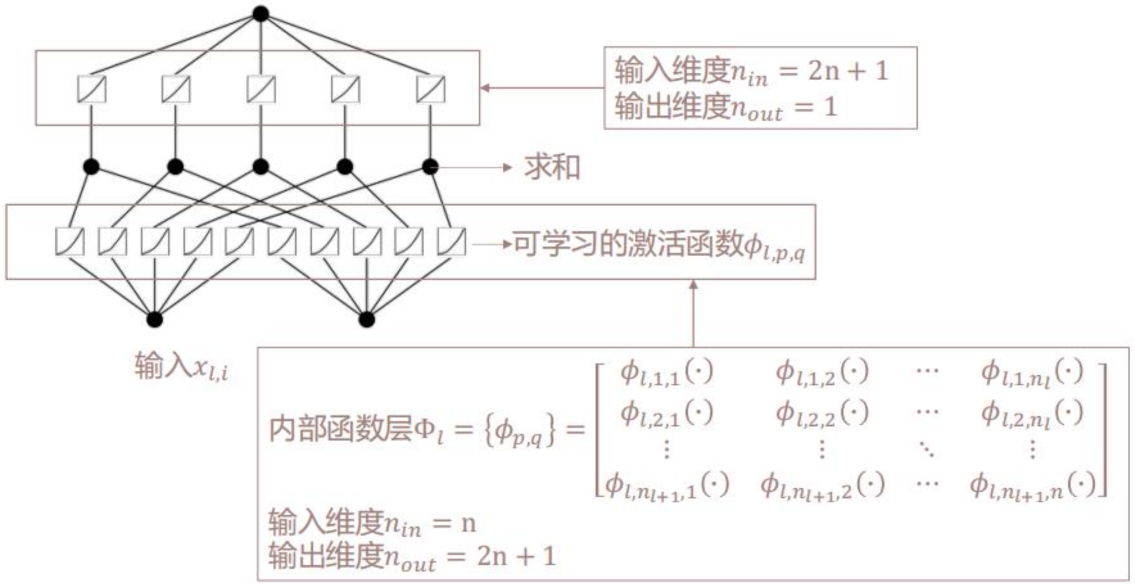
$$B_{i,0}(x) := \begin{cases} 1 & if\ t_i \le x < t_{i+1}, \\ 0 & otherwise. \end{cases}$$

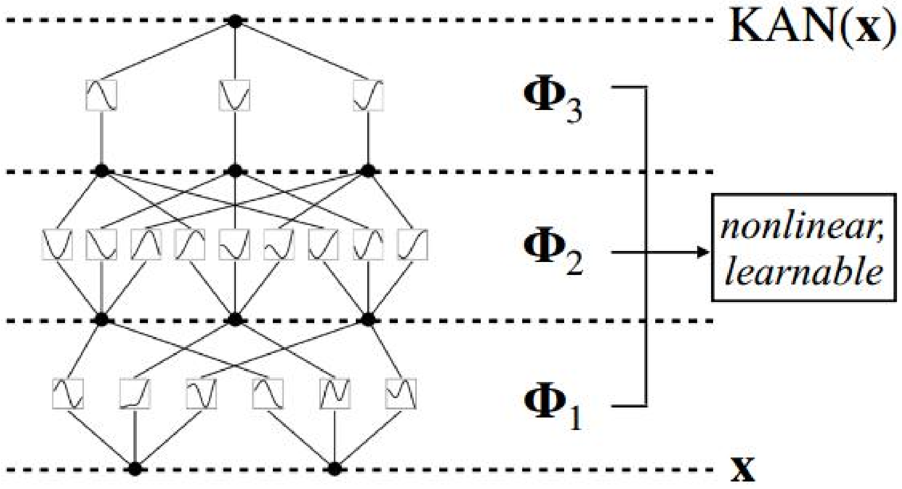
$$B_{i,k}(x) := \frac{x-t_i}{t_{i+k}-t_i}B_{i,k-1}(x) + \frac{t_{i+k+1}-x}{t_{i+k+1}-t_{i+1}}B_{i+1,k-1}(x).$$



Complete KAN structure

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





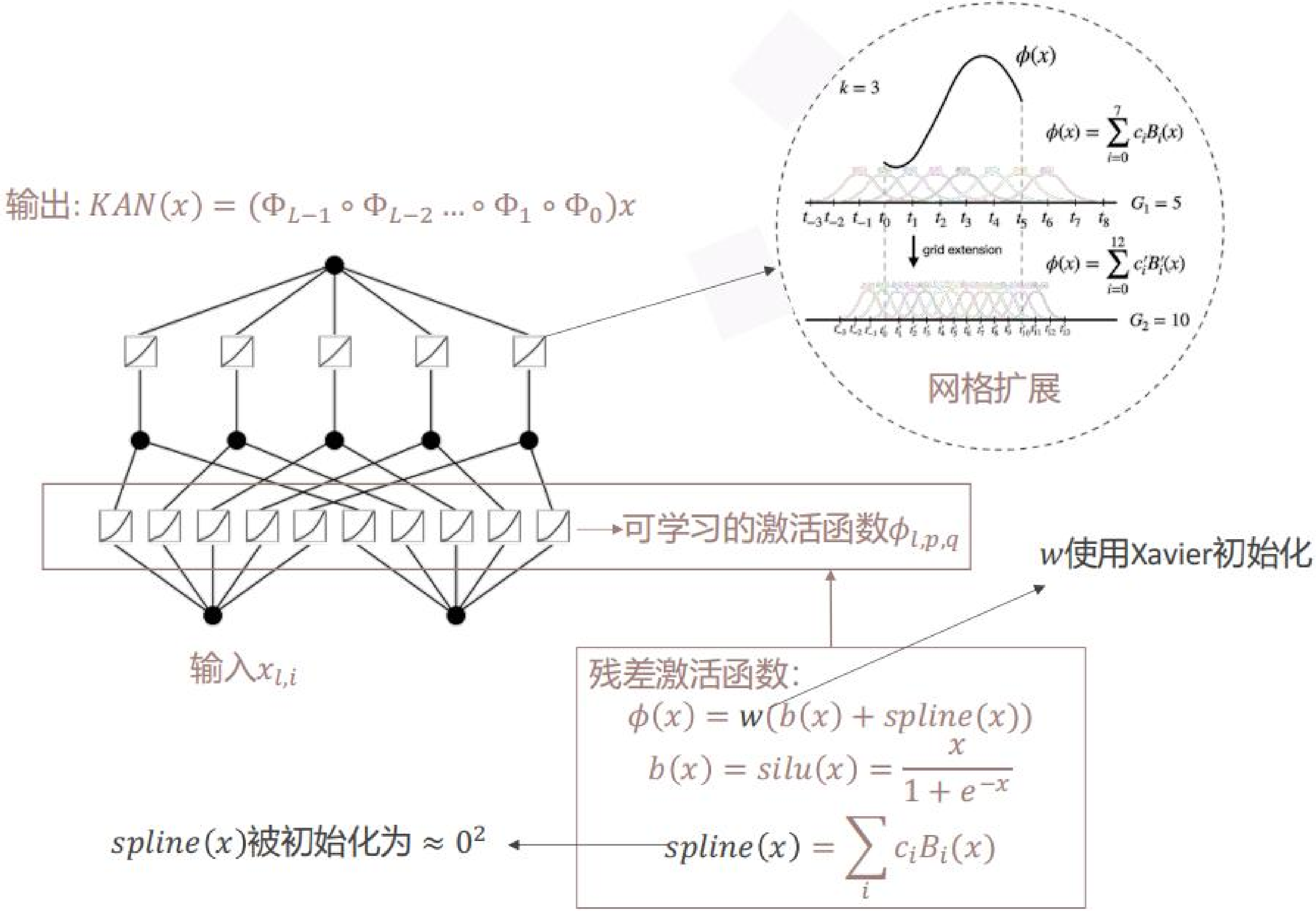
where $\mathbf{\Phi}_l$ is the function matrix corresponding to the l^{th} KAN layer. A general KAN network is a composition of L layers: given an input vector $\mathbf{x}_0 \in \mathbb{R}^{n_0}$, the output of KAN is

$$\text{KAN}(\mathbf{x}) = (\mathbf{\Phi}_{L-1} \circ \mathbf{\Phi}_{L-2} \circ \cdots \circ \mathbf{\Phi}_1 \circ \mathbf{\Phi}_0)\mathbf{x}. \tag{2.7}$$

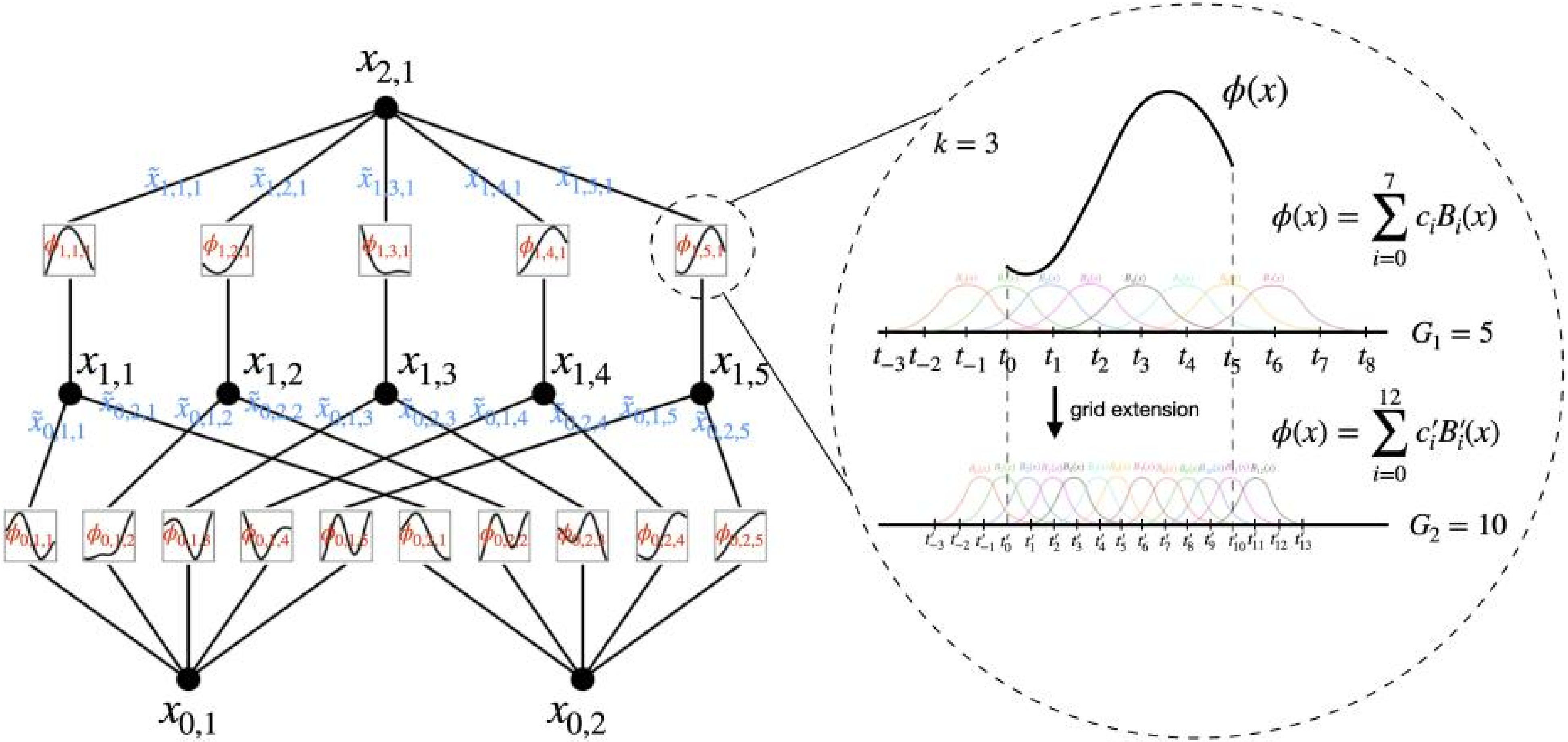
We can also rewrite the above equation to make it more analogous to Eq. (2.1), assuming output dimension $n_L = 1$, and define $f(\mathbf{x}) \equiv \text{KAN}(\mathbf{x})$:

$$f(\mathbf{x}) = \sum_{i_{L-1}=1}^{n_{L-1}} \phi_{L-1,i_L,i_{L-1}} \left(\sum_{i_{L-2}=1}^{n_{L-2}} \cdots \left(\sum_{i_2=1}^{n_2} \phi_{2,i_3,i_2} \left(\sum_{i_1=1}^{n_1} \phi_{1,i_2,i_1} \left(\sum_{i_0=1}^{n_0} \phi_{0,i_1,i_0}(x_{i_0}) \right) \right) \right) \cdots \right), \tag{2.8}$$

Optimization of KAN (1)



Optimization of KAN (2)



Optimization of KAN (2) -improve the accurancy

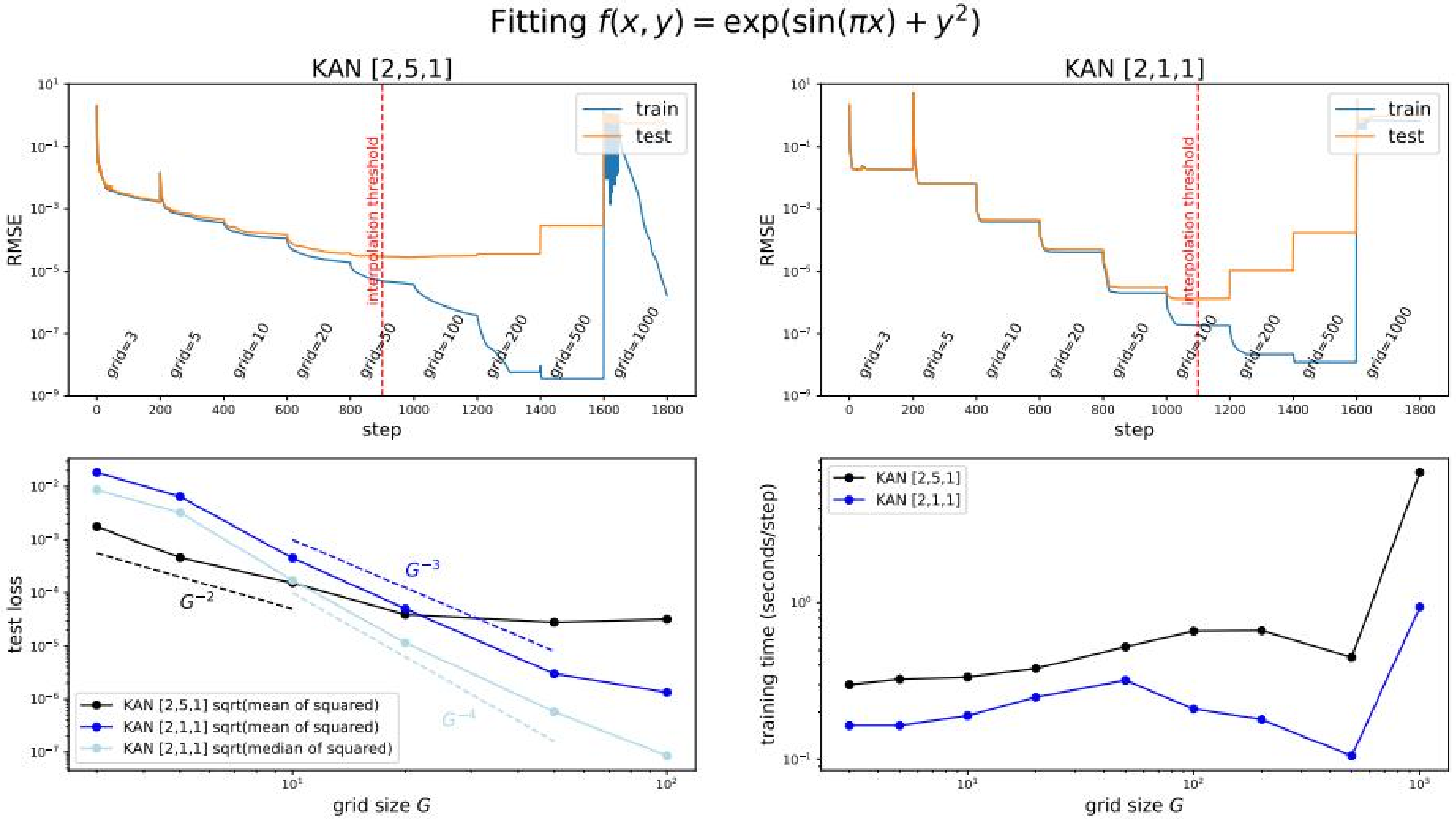
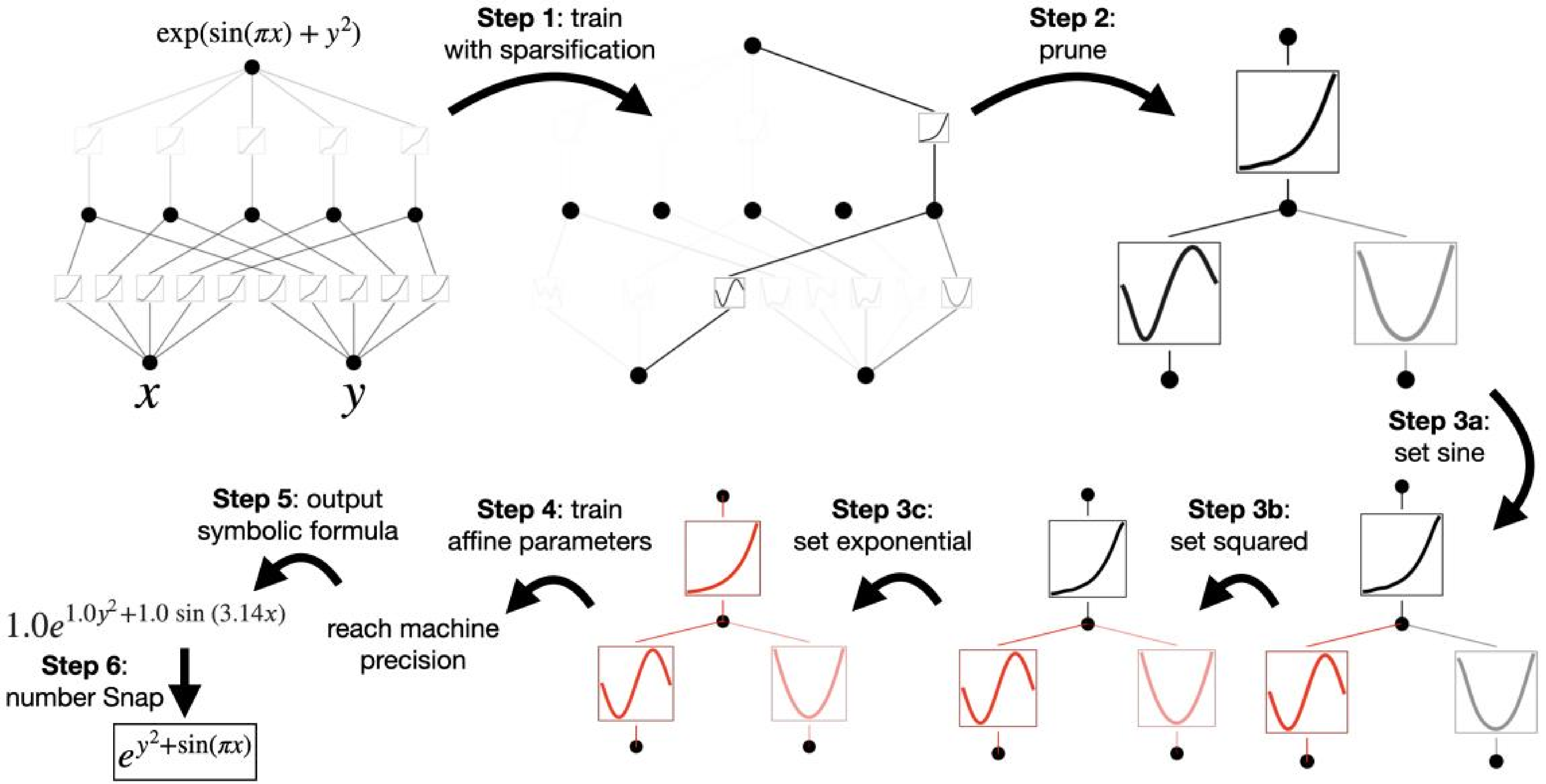


Figure 2.3: We can make KANs more accurate by grid extension (fine-graining spline grids). Top left (right): training dynamics of a [2, 5, 1] ([2, 1, 1]) KAN. Both models display staircases in their loss curves, i.e., loss suddently drops then plateaus after grid extension. Bottom left: test RMSE follows scaling laws against grid size G. Bottom right: training time scales favorably with grid size G.

The Interpretability of KAN



The Interpretability of KAN

KAN用于拟合特殊函数

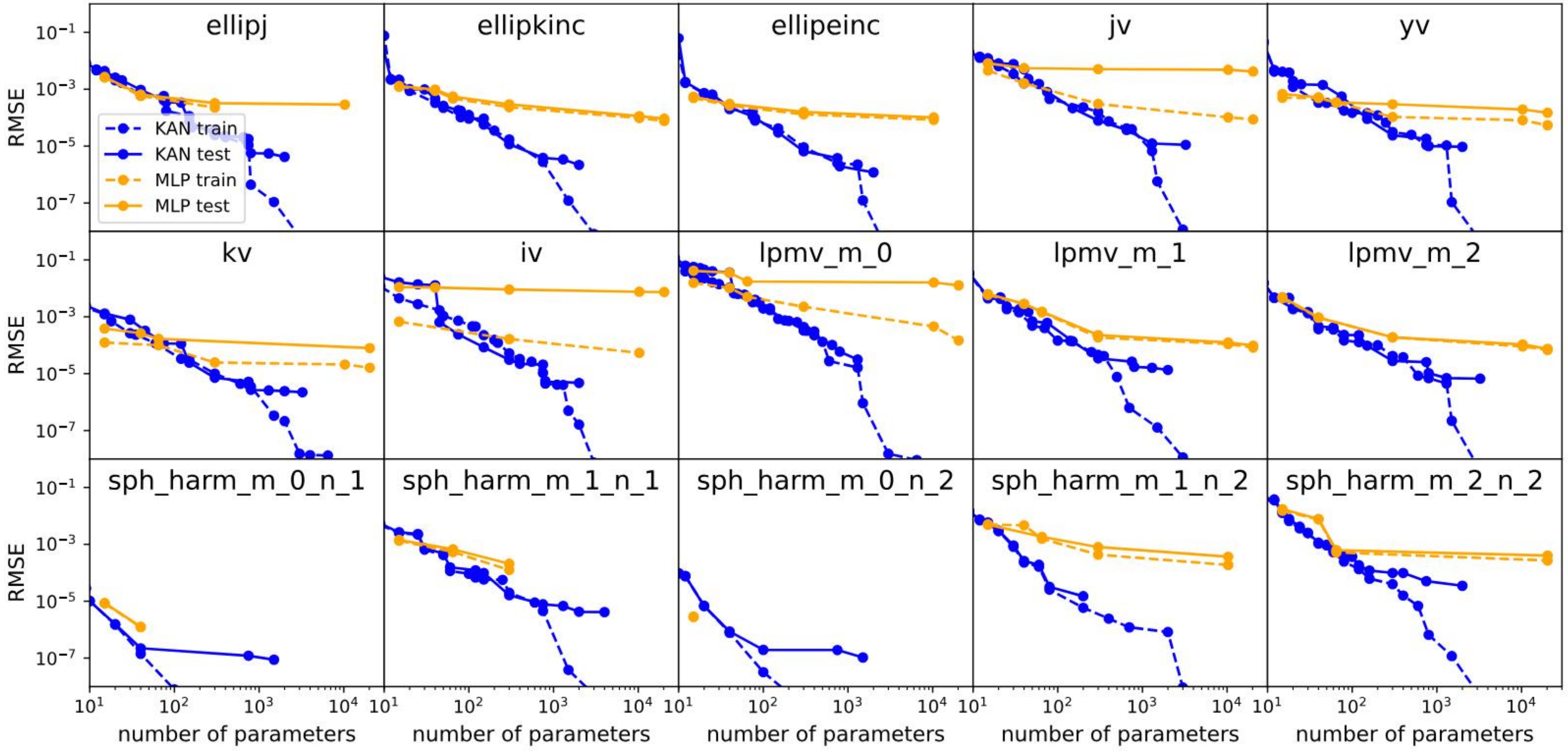


Figure 3.2: Fitting special functions. We show the Pareto Frontier of KANs and MLPs in the plane spanned by the number of model parameters and RMSE loss. Consistently accross all special functions, KANs have better Pareto Frontiers than MLPs. The definitions of these special functions are in Table 2.

KAN用于解偏微分方程

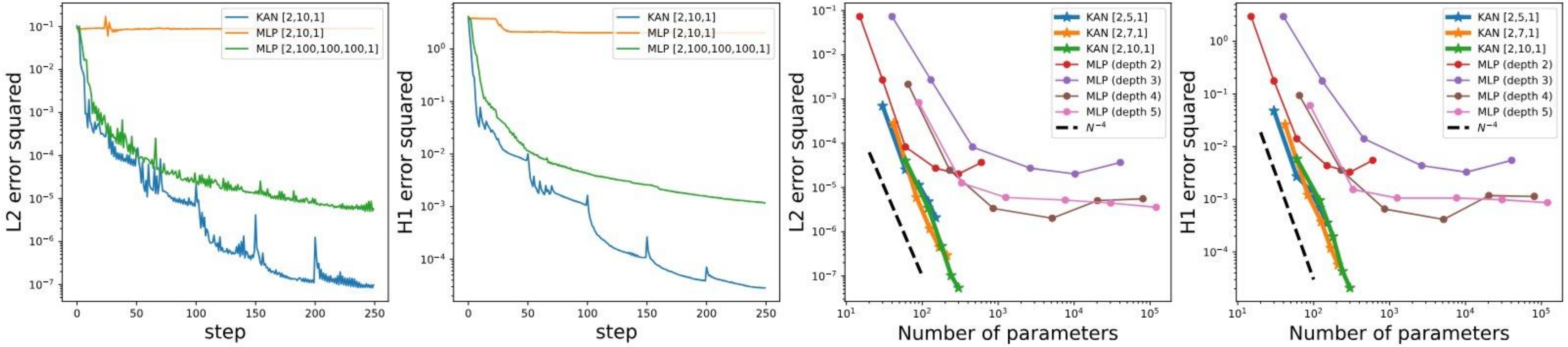


Figure 3.3: The PDE example. We plot L2 squared and H1 squared losses between the predicted solution and ground truth solution. First and second: training dynamics of losses. Third and fourth: scaling laws of losses against the number of parameters. KANs converge faster, achieve lower losses, and have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

KAN可以解决灾难性遗忘

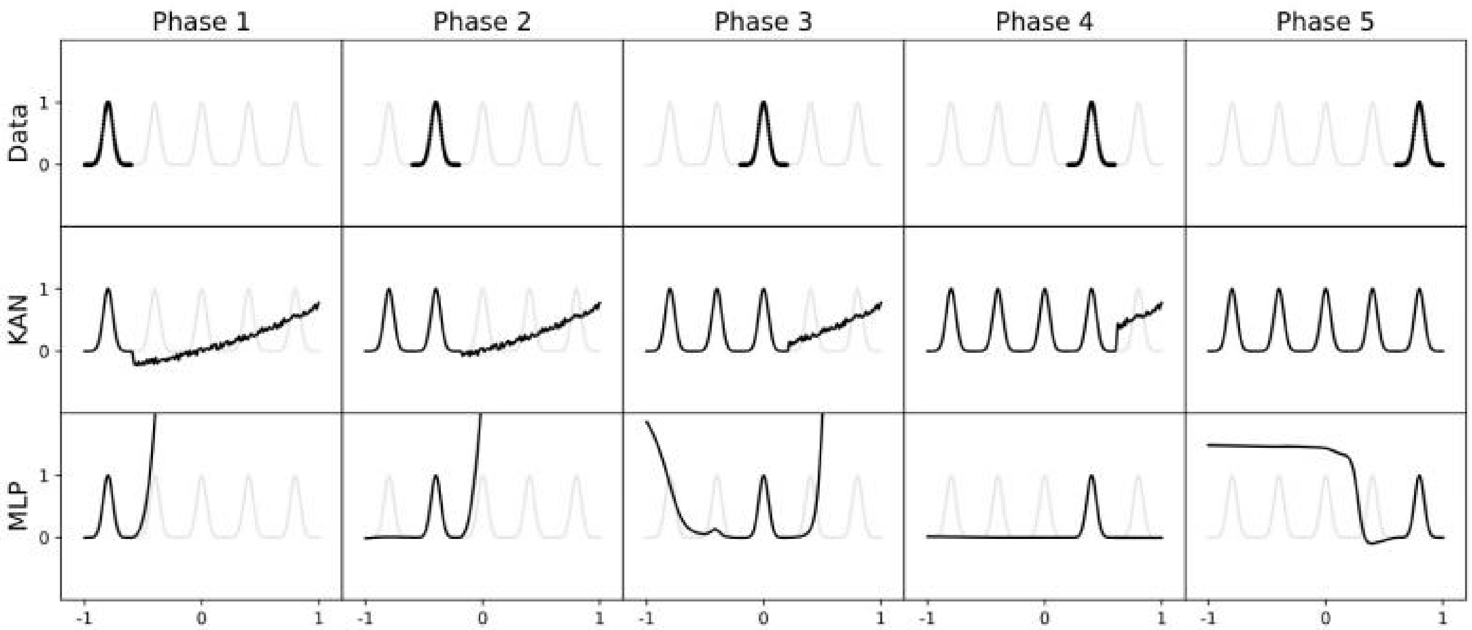


Figure 3.4: A toy continual learning problem. The dataset is a 1D regression task with 5 Gaussian peaks (top row). Data around each peak is presented sequentially (instead of all at once) to KANs and MLPs. KANs (middle row) can perfectly avoid catastrophic forgetting, while MLPs (bottom row) display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

使用KAN的场景讨论

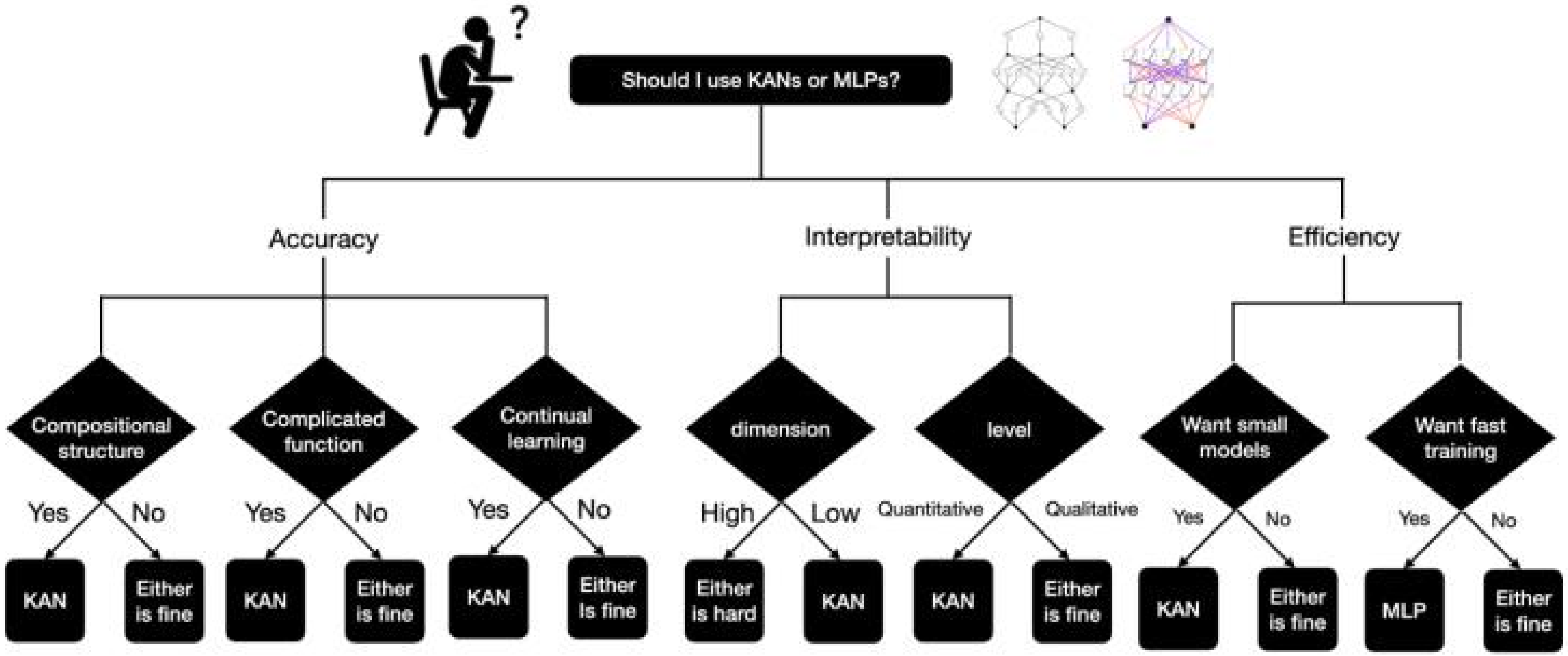


Figure 6.1: Should I use KANs or MLPs?

Thank you for listening

主講人：吴雨欣

2024.5.30