

# AI For Math

簡偉恆、盧詠涵

輔仁大學 數學系資訊數學組 一年級

2025 年 6 月 8 日

- ① 社群活動簡介
- ② Kolmogorov–Arnold Network 簡介
- ③ Kolmogorov–Arnold Network 實作
- ④ 記憶模型是否能降低 KAN 的誤差

## 113 年學年度第二學期 AI for Math 系列演講

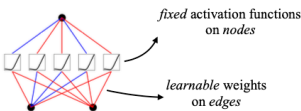
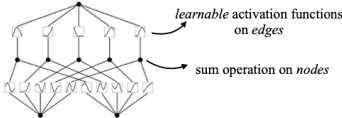
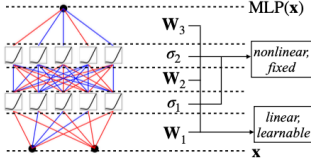
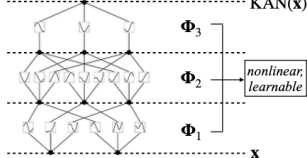
日期	講者	講題
114/02/27	潘老師	感知機 (The Perceptron)
114/03/06	潘老師	淺談 Adaptive Linear Neuron 和 Widrow-Hoff Learning
114/03/20	潘老師	The Basics of Multilayer Perceptron and Back-propagation
114/03/27	俞讚城老師	Introduction to Shannon Entropy and Cross Entropy
114/04/17	俞讚城老師	Introduction to Universal Approximation Theorems and Application in AI
114/05/08	嚴健彰老師	KAN: Kolmogorov-Arnold Networks

## 論文簡介

- 發表時間：2023 年
- 標題：KAN: Kolmogorov–Arnold Networks
- 主要特點：整合領域先驗知識與深度神經網路
- 應用領域：數學建模與科學計算

# Kolmogorov-Arnold Network 簡介

## MLP vs. KAN

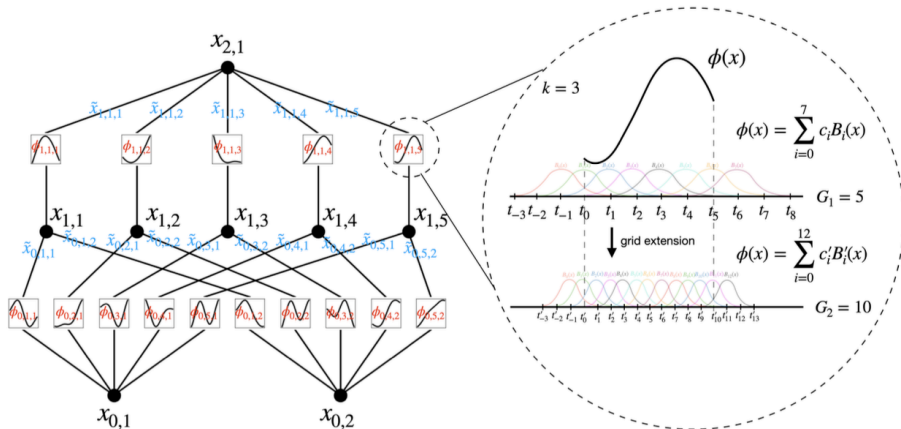
Model	Multi-Layer Perceptron (MLP)	Kolmogorov-Arnold Network (KAN)
Theorem	Universal Approximation Theorem	Kolmogorov-Arnold Representation Theorem
Formula (Shallow)	$f(\mathbf{x}) \approx \sum_{i=1}^{N(e)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$	$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(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

## Kolmogorov–Arnold Networks 架構與特點

- Kolmogorov–Arnold Network(KAN) 受 Kolmogorov–Arnold Representation theorem(KAT) 啟發
- 創新架構：
  - 可學習的一維激活函數位於邊上，取代傳統線性權重
  - 使用樣條函數 (spline) 參數化
  - 每個節點僅執行線性加總，不附加任何非線性激活函數

# Kolmogorov–Arnold Network 簡介

## Kolmogorov–Arnold Networks 架構圖



# Kolmogorov–Arnold Network 簡介

## Kolmogorov–Arnold Representation Theorem 簡介

- 提出者：Andrey Kolmogorov, Vladimir Arnold
- 內容：針對任意有界區間  $[0, 1]^n$  上的連續函數  $f(x_1, \dots, x_n)$ ，該定理保證：

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

- 其中：
  - $\varphi_{q,p} : [0, 1] \rightarrow \mathbb{R}$  為一維連續函數，作用於單一變量  $x_p$
  - $\Phi_q : \mathbb{R} \rightarrow \mathbb{R}$  為一維連續函數，作用於所有  $\varphi_{q,p}(x_p)$  的加總結果



## Kolmogorov–Arnold Networks 優勢

- 效能優勢：
  - 小規模 AI 任務中，參數量更少
  - 比 MLP 擁有更高精度
  - 更快的泛化縮放律
- 可解釋性：
  - 激活函數可視化
  - 可逐層稀疏修剪
  - 適用於（準）符號回歸與科學發現
- 應用優勢：
  - 可用於 PDE 求解（PINN 框架）
  - 連續學習中能有效避免遺忘現象
  - 結合樣條高精度與 MLP 組合結構

# Kolmogorov–Arnold Networks 實作

[2,1,1] Test RMSE: 1.265029

=== [2,1,1] 測試點比較 ===

x=0.10, y=0.10	真實 : 1.375775	預測 : 3.680561	誤差 : 2.304786
x=0.50, y=0.50	真實 : 3.490343	預測 : 2.694129	誤差 : 0.796214
x=0.75, y=0.25	真實 : 2.158917	預測 : 2.694129	誤差 : 0.535212

▲ [2,1,1] 測試點比較

[2,5,1] Test RMSE: 1.445008

=== [2,5,1] 測試點比較 ===

x=0.10, y=0.10	真實 : 1.375775	預測 : 2.344764	誤差 : 0.968989
x=0.50, y=0.50	真實 : 3.490343	預測 : 3.075765	誤差 : 0.414578
x=0.75, y=0.25	真實 : 2.158917	預測 : 2.608403	誤差 : 0.449486

▲ [2,5,1] 測試點比較

# Kolmogorov–Arnold Networks 實作

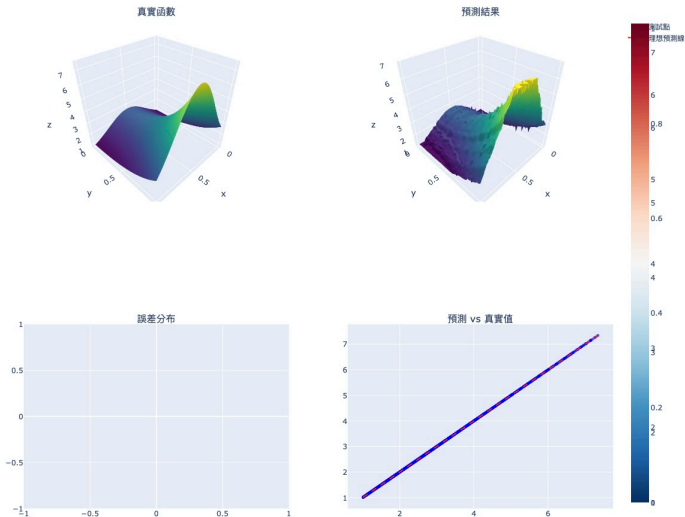
```
[2,1,1] Test RMSE: 0.002997
```

```
=== [2,1,1] 測試點比較 ===
```

x=0.10, y=0.10	真實 : 1.375775	預測 : 1.376149	誤差 : 0.000375
x=0.50, y=0.50	真實 : 3.490343	預測 : 3.486988	誤差 : 0.003355
x=0.75, y=0.25	真實 : 2.158917	預測 : 2.158463	誤差 : <u>0.000454</u>

▲ [2,1,1] 測試點比較 2

# Kolmogorov–Arnold Networks 實作



## ▲ [2,1,1] 測試點比較 2-視覺化

# Kolmogorov–Arnold Networks 實作

[2,1,1] Test RMSE: 0.003584

=== [2,1,1] 測試點比較 ===

x=0.10, y=0.10	真實 : 1.375775	預測 : 1.376706	誤差 : 0.000932
x=0.50, y=0.50	真實 : 3.490343	預測 : 3.489892	誤差 : 0.000452
x=0.75, y=0.25	真實 : 2.158917	預測 : 2.158352	誤差 : 0.000565

[2,1,1] Test RMSE: 1.112584

=== [2,1,1] 測試點比較 ===

x=0.10, y=0.10	真實 : 1.375775	預測 : 2.449288	誤差 : 1.073513
x=0.50, y=0.50	真實 : 3.490343	預測 : 2.449288	誤差 : 1.041055
x=0.75, y=0.25	真實 : 2.158917	預測 : 2.449288	誤差 : 0.290371

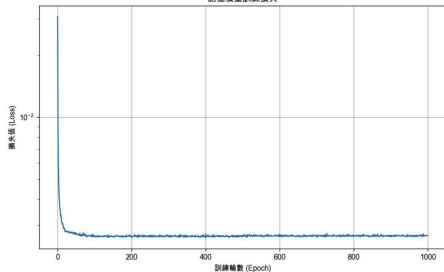
[2,1,1] Test RMSE: 0.002997

=== [2,1,1] 測試點比較 ===

x=0.10, y=0.10	真實 : 1.375775	預測 : 1.376149	誤差 : 0.000375
x=0.50, y=0.50	真實 : 3.490343	預測 : 3.486988	誤差 : 0.003355
x=0.75, y=0.25	真實 : 2.158917	預測 : 2.158463	誤差 : 0.000454

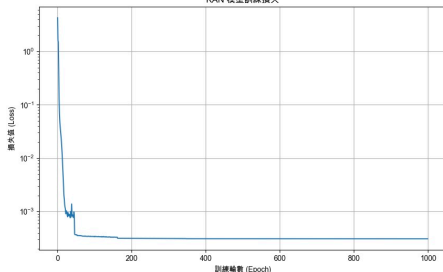
# 記憶模型是否能降低 KAN 的誤差

記憶模型訓練損失



▲ 記憶功能訓練結果

KAN 模型訓練損失



▲ KAN 訓練結果

# 記憶模型能不能降低 KAN 的誤差

測試結果：

輸入： $x = 0.5, y = 0.5$

預測值：3.489805

真實值：3.490343

相對誤差：0.02%

▲ 增加記憶功能後的結果數據

# 謝謝大家

## 報告結束