SoICT 2024 | Session 8B | AI Foundations and Big Data | December, 13th

BSRBF-KAN: A combination of B-splines and Radial Basis Functions in Kolmogorov-Arnold Networks

Author: Hoang-Thang Ta
Affiliate: Dalat University, Vietnam

Index

1. Introduction
2. Related works
3. Methodology: Kolmogorov-Arnold Representation Theorem, Kolmogorov-Arnold Networks, and Several KANs
4. Experiments: MNIST + Fashion-MNIST, Misclassification Analysis, Ablation Study
5. Limitation
6. Conclusion
7. Questions + Answers

1. Introduction

- Kolmogorov-Arnold Networks (KANs) use learnable **activation functions** as "edges" to fit training data, as opposed to the fixed activation functions used as "nodes" in multi-layer perceptrons (MLPs).
- KANs are based on the **Kolmogorov-Arnold Representation Theorem**. Multilayer Perceptrons (MLPs) are based on **Universal Approximation Theory.**

→ New research trends: novel KANs and integrating KANs with CNNs, RNNs, etc.

1. Introduction

- Propose **BSRBF-KAN** combines **B-splines** and **radial basis functions (RBFs)** within Kolmogorov Arnold Networks (KAN) for improved data fitting.
- BSRBF-KAN shows stability and **better convergence** compared to MLPs and other networks.

→ Inspire new mathematical function combinations for KAN design.

2. Related works

- **Hilbert's 13th problem**
  - concerns the solvability of the general **7th-degree polynomial function** using continuous functions of only two variables
- **Kolmogorov-Arnold Representation Theorem (Vladimir Arnold, 1957)**
  - demonstrating that a multivariate continuous function can be represented as a combination of single-variable functions and additions
- Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T.Y., Tegmark, M.: **Kan: Kolmogorov-arnold networks**. arXiv preprint arXiv:2404.19756 (2024)
  - Original KAN (LiuKAN, Spl-KAN), EfficientKAN, FastKAN, FasterKAN, Chebyshev KAN, etc.

3. Methodology

Kolmogorov-Arnold Representation Theorem (KART -- Vladimir Arnold, 1957)

- Any multivariate continuous function **f** defined on a bounded domain can be expressed using a finite number of continuous single-variable functions and additions
- Given **x** = x_1 , x_2 , . . . , x_n consisting of **n** variables, a multivariate continuous function **f(x)** is represented by:

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

3. Methodology

Kolmogorov-Arnold Networks

- An MLP characterized by affine transformations and non-linear functions.
- From an input **x**, the network performs the composition of weight matrices by layers (from **layer 0 to layer L-1** and the non-linearity (activation function) **σ**.

$$\begin{aligned} \mathrm{MLP}(\mathbf{x}) &= (W_{L-1} \circ \sigma \circ W_{L-2} \circ \sigma \circ \cdots \circ W_1 \circ \sigma \circ W_0)\mathbf{x} \\ &= \sigma(W_{L-1}\sigma(W_{L-2}\sigma(\cdots\sigma(W_1\sigma(W_0\mathbf{x}))))) \end{aligned} \tag{2}$$

3. Methodology

Kolmogorov-Arnold Networks (based on KART)

- In Equation 1, we must search proper $\boldsymbol{\Phi_q}$ and $\boldsymbol{\varphi_{q,p}}$ to solve the problem. A general KAN network consisting of $\mathbf{L}$ layers takes $\mathbf{x}$ to generate the output as:

$$\mathrm{KAN}(\mathbf{x}) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \cdots \circ \Phi_1 \circ \Phi_0)\mathbf{x} \quad (3)$$

- which $\boldsymbol{\Phi_l}$ is the function matrix of the $\mathbf{l^{th}}$ KAN layer or a set of pre-activations.

3. Methodology

Kolmogorov-Arnold Networks

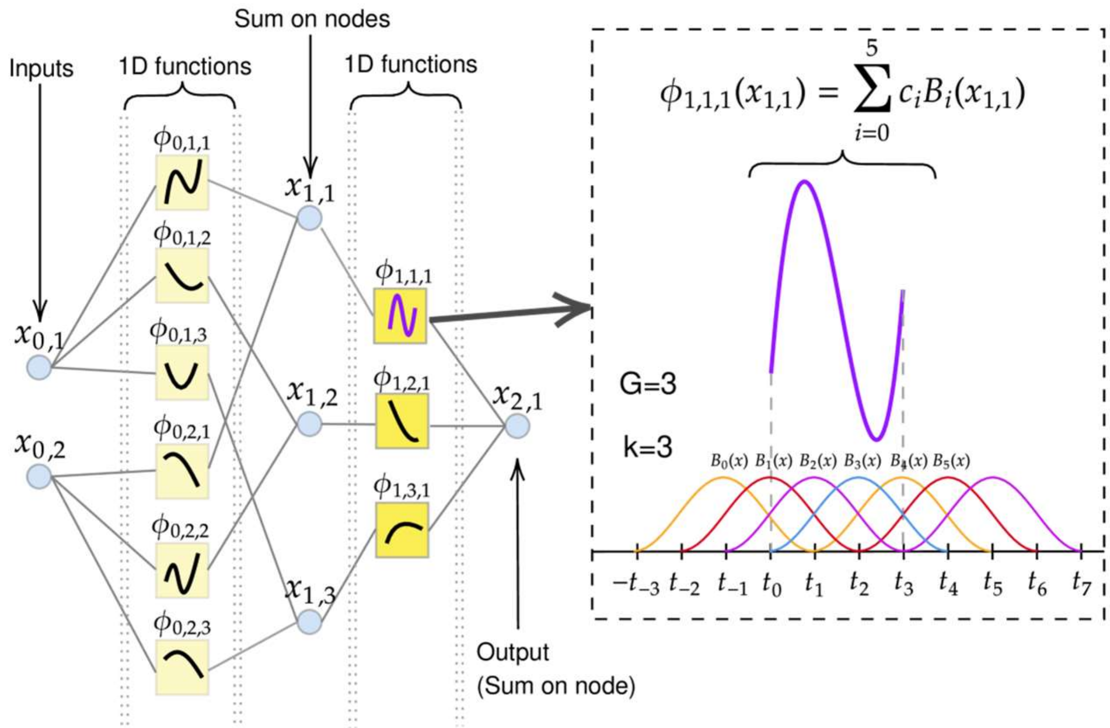
- The function matrix $\mathbf{\Phi_l}$ can be represented as a matrix $\mathbf{n_{l+1} \times n_l}$ of activations as:

$$\mathbf{x}_{l+1} = \underbrace{\begin{pmatrix} \phi_{l,1,1}(\cdot) & \phi_{l,1,2}(\cdot) & \cdots & \phi_{l,1,n_l}(\cdot) \\ \phi_{l,2,1}(\cdot) & \phi_{l,2,2}(\cdot) & \cdots & \phi_{l,2,n_l}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{l,n_{l+1},1}(\cdot) & \phi_{l,n_{l+1},2}(\cdot) & \cdots & \phi_{l,n_{l+1},n_l}(\cdot) \end{pmatrix}}_{\Phi_l} \mathbf{x}_l \qquad (5)$$

3. Methodology

Kolmogorov-Arnold Networks

- Example: The structure of KAN(2,3,1)



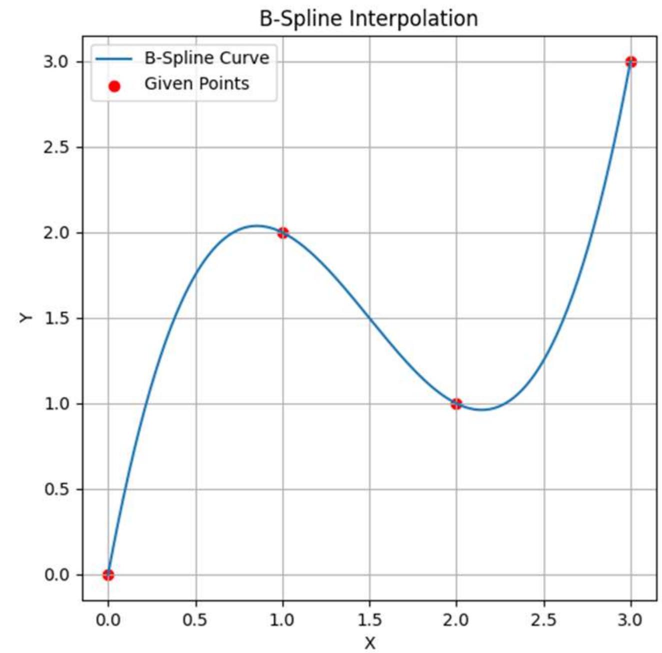
3. Methodology

Implementation of the current KANs

- **Original KAN (Spl-KAN, LiuKAN)**
  - Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T.Y., Tegmark, M.: Kan: Kolmogorov-arnold networks. arXiv preprint arXiv:2404.19756 (2024)
  - The residual activation function **φ(x)** as the sum of the base function and the spline function with their corresponding weight matrices $\mathbf{w_b}$ and $\mathbf{w_s}$:

$$\phi(x) = w_b b(x) + w_s spline(x) \quad (6)$$

    - *b(x) = silu(x)* (activation functions)
    - *spline(x)* = linear combination of B-Splines



3. Methodology

Implementation of the current KANs

- **EfficientKAN**
  - using B-splines followed by linear combination, reducing memory cost and simplifying computation.
  - https://github.com/Blealtan/efficient-kan
- **FastKAN**
  - using GRBFs to approximate the 3-order B-spline and employing layer normalization to keep inputs within the RBFs' domain. The RBF has the formula:

$$\phi(r) = e^{-\epsilon r^2} \qquad (9)$$

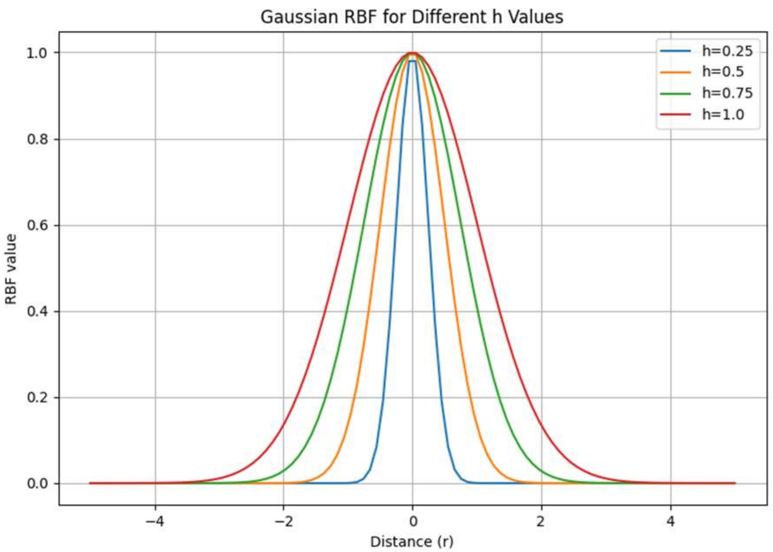
  - where **$r=||x-c||$** is the distance between an input vector **x** and a center **c**, and **$\epsilon > 0$** controls the width of the Gaussian function.

3. Methodology

Implementation of the current KANs

- **FastKAN**
  - Li, Z.: Kolmogorov-arnold networks are radial basis function networks. arXiv preprint arXiv:2405.06721 (2024)
  - FastKAN uses a special form of RBFs, GRBFs where **ϵ=0.5** and **h** controls the width of the Gaussian function:

$$\phi_{RBF}(r) = \exp\left(-\frac{r^2}{2h^2}\right) \qquad (8)$$



  - The RBF network with **N** centers can be shown as:

$$RBF(x) = \sum_{i=1}^{N} w_i \phi_{RBF}(r_i) = \sum_{i=1}^{N} w_i \exp\left(-\frac{\|x - c_i\|}{2h^2}\right) \quad (9)$$

3. Methodology

Implementation of the current KANs

- **FasterKAN**
  - Li, Z.: Kolmogorov-arnold networks are radial basis function networks. arXiv preprint arXiv:2405.06721 (2024)
  - Use Reflectional Switch Activation Functions (RSWAFs), which are variants of RBFs:

$$RSWAF(x) = \sum_{i=1}^{N} w_i \phi_{RSWAF}(r_i) = \sum_{i=1}^{N} w_i \left(1 - \left(\tanh\left(\frac{\|x - c_i\|}{h}\right)\right)^2\right) \quad (11)$$

- **GottliebKAN**, which is based on **Gottlieb polynomials**
  - Teymoor Seydi, S.: Exploring the potential of polynomial basis functions in kolmogorov-arnold networks: A comparative study of different groups of polynomials. arXiv e-prints pp. arXiv–2406 (2024)

3. Methodology

Implementation of the current KANs

- **BSBRF-KAN**
  - Combine B-splines and Gaussian RBFs for interpolation and approximation, ensures smoothness, continuity, and proper derivative handling.
  - From an input *x*, the BSRBF function is represented as:

```python
def forward(self, x):

 # layer normalization
 x = self.layernorm(x)
 # base
 base_output = F.linear(self.base_activation(x), self.base_weight)
 # b_splines
 bs_output = self.b_splines(x).view(x.size(0), -1)
 # rbfs
 rbf_output = self.rbf(x).view(x.size(0), -1)
 # combine
 bsrbf_output = bs_output + rbf_output
 bsrbf_output = F.linear(bsrbf_output, self.spline_weight)
 return base_output + bsrbf_output
```

$$BSRBF(x) = w_b b(x) + w_s(BS(x) + RBF(x)) \quad (13)$$

4. Experiments

Configuration + Model Losses

- Each network was trained with **5 independent runs** on MNIST (15 epochs) and Fashion-MNIST (25 epochs) using models structured as **(784, 64, 10)**, and calculate the metric averages.
- For GottliebKAN, we used (784, 64, 64, 10) to be similar to the original design.
- Parameters: **batch_size=64, learning_rate=1e-3, weight_decay=1e-4, gamma=0.8, optimize=AdamW, and loss=CrossEntropy**.

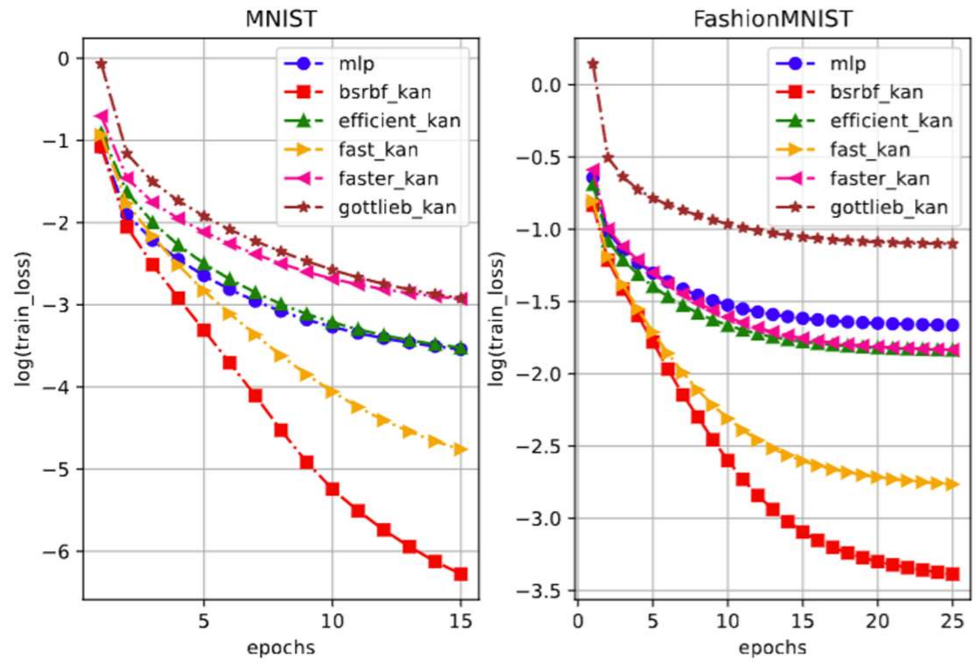


Fig. 1. The logarithmic values of training losses during a training run over 15 epochs on MNIST and 25 epochs on Fashion-MNIST.

4. Experiments

The best metric values in 5 training runs

Table 1. The best metric values in 5 training runs on MNIST and Fashion-MNIST.

Dataset	Model	Train. Acc.	Val. Acc.	F1	Time (seconds)	#Params
MNIST	BSRBF-KAN	**100.0**	97.63	97.6	222	459040
	FastKAN	99.94	97.38	97.34	102	459114
	FasterKAN	98.52	97.38	97.36	**93**	408224
	EfficientKAN	99.34	97.54	97.5	122	508160
	GottliebKAN	99.66	**97.78**	**97.74**	269	219927
	MLP	99.42	97.69	97.66	273	**52512**
Fashion-MNIST	BSRBF-KAN	**99.3**	89.59	89.54	219	459040
	FastKAN	98.27	**89.62**	**89.6**	160	459114
	FasterKAN	94.4	89.39	89.3	157	408224
	EfficientKAN	94.83	89.11	89.04	182	508160
	GottliebKAN	93.79	87.69	87.61	241	219927
	MLP	93.58	88.51	88.44	**147**	**52512**
Average of MNIST + Fashion-MNIST	BSRBF-KAN	**99.65**	**93.61**	**93.57**	220.5	459040
	FastKAN	99.11	93.50	93.47	131	459114
	FasterKAN	96.46	93.39	93.33	**125**	408224
	EfficientKAN	97.09	93.33	93.27	152	508160
	GottliebKAN	96.73	92.74	92.68	255	219927
	MLP	96.50	93.10	93.05	210	**52512**

Train. Acc = Training Accuracy, Val. Acc. = Validation Accuracy
#Params = Parameters

4. Experiments

The average metric values in 5 training runs

Dataset	Model	Train. Acc.	Val. Acc.	F1	Time (secs)
MNIST	BSRBF-KAN	**100.00 ± 0.00**	97.55 ± 0.03	97.51 ± 0.03	231
	FastKAN	99.94 ± 0.01	97.25 ± 0.03	97.21 ± 0.03	101
	FasterKAN	98.48 ± 0.01	97.28 ± 0.06	97.25 ± 0.06	**93**
	EfficientKAN	99.37 ± 0.04	97.37 ± 0.07	97.33 ± 0.07	120
	GottliebKAN	98.44 ± 0.61	97.19 ± 0.22	97.14 ± 0.23	221
	MLP	99.44 ± 0.01	**97.62 ± 0.03**	**97.59 ± 0.03**	181
Fashion-MNIST	BSRBF-KAN	**99.19 ± 0.03**	89.33 ± 0.07	89.29 ± 0.07	211
	FastKAN	98.19 ± 0.04	**89.42 ± 0.07**	**89.38 ± 0.07**	162
	FasterKAN	94.40 ± 0.01	89.26 ± 0.06	89.17 ± 0.07	154
	EfficientKAN	94.76 ± 0.06	88.92 ± 0.08	88.85 ± 0.09	183
	GottliebKAN	90.66 ± 1.08	87.16 ± 0.24	87.07 ± 0.25	238
	MLP	93.56 ± 0.05	88.39 ± 0.06	88.36 ± 0.05	**148**
Average of MNIST + Fashion-MNIST	BSRBF-KAN	**99.60**	**93.44**	**93.40**	221
	FastKAN	99.07	93.34	93.30	131.5
	FasterKAN	96.44	93.27	93.21	123.5
	EfficientKAN	97.07	93.15	93.09	151.5
	GottliebKAN	94.55	92.18	92.11	229.5
	MLP	96.50	93.01	92.98	164.5

Train. Acc = Training Accuracy, Val. Acc. = Validation Accuracy

4. Experiments

Misclassification Analysis

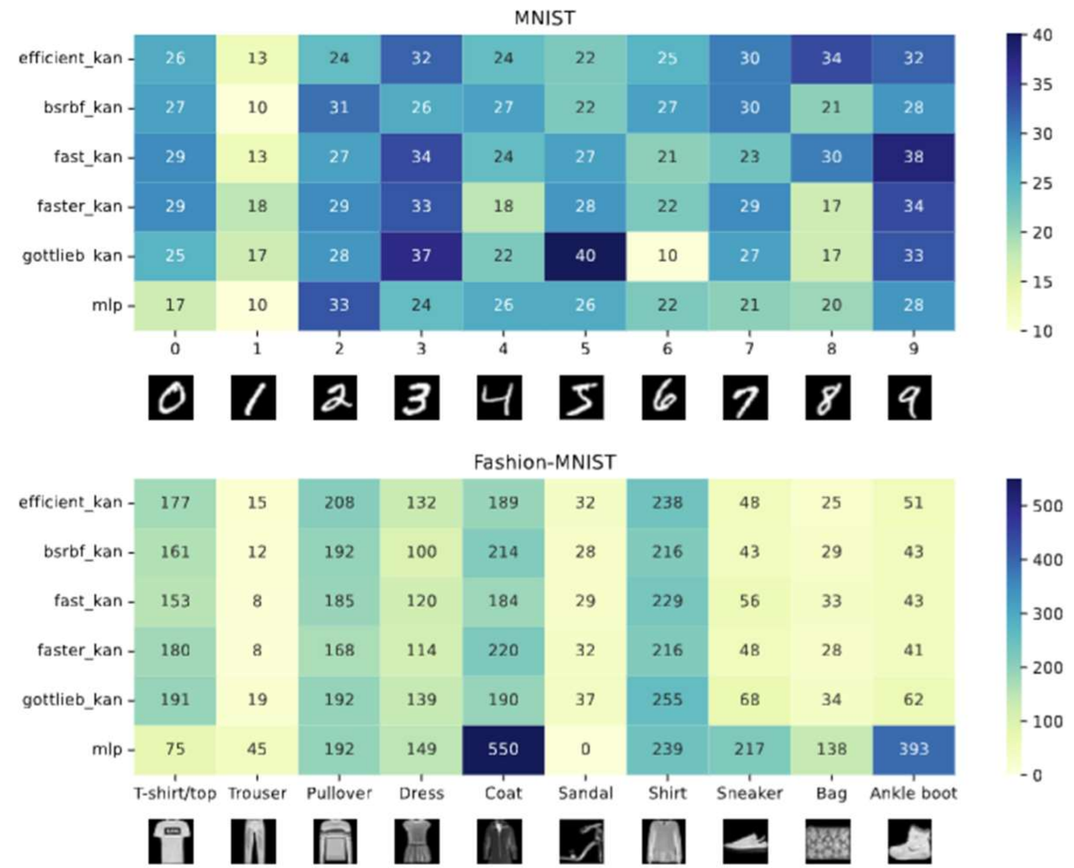


Fig. 2. Heatmap of the misclassified images in the test set by models over MNIST and Fashion-MNIST.

4. Experiments

Ablation Study

Table 3. The performance of BSRBF-KAN by different components on MNIST and Fashion-MNIST.

Dataset	Components	Train. Acc.	Val. Acc.	F1
MNIST	Full	100.0	97.53	97.49
	No BS	- 0.01	- 0.21	- 0.21
	No RBF	- 0.003	- 0.14	- 0.14
	No BS + No RBF = MLP	- 0.55	+ 0.05	+ 0.06
	No BO	- 0.01	- 0.41	- 0.41
	No LN	- 1.7	- 1.5	- 1.52
	No BO + No LN	**- 5.16**	**- 5.97**	**- 6.08**
Fashion-MNIST	Full	99.39	89.43	89.4
	No BS	- 0.16	- 0.39	- 0.37
	No RBF	- 0.34	- 0.14	- 0.15
	No BS + No RBF = MLP	-5.86	-0.81	-0.78
	No BO	- 0.04	- 0.26	- 0.29
	No LN	- 6.12	- 1.1	- 1.13
	No BO + No LN	**-6.23**	**-3.29**	**-3.38**

Train. Acc = Training Accuracy, Val. Acc. = Validation Accuracy
No BS = No B-spline, No RBF = No Radial Basis Function
No BO = No Base Output, No LN = No Layer Normalization

5. Limitations

- Experiments were conducted **simple datasets** (MNIST and Fashion-MNIST)
- **The fairness of using the number of parameters** (KANs vs. MLP)
- Combining B-splines and Radial Basis Functions was chosen based on observed performance **without a formalized mathematical justification or deep analyses**.

→Requirements:

→ Test more datasets and tasks

→Design models with equal parameter counts for balanced comparisons

→ Explore the theoretical basis for function combinations through mathematical and qualitative analyses.

6. Conclusion + Future works

- The paper introduces BSRBF-KAN, a new Kolmogorov Arnold Network combining B-splines and RBFs for training.
  - The focus was on understanding KAN combinations rather than optimizing hyperparameters.
- BSRBF-KAN showed **competitive performance and rapid convergence** on MNIST and Fashion-MNIST,
  - high convergence could lead to overfitting, which can be mitigated.
- Misclassification analysis and an ablation study were conducted to identify critical components.
- Future work involves testing on more datasets and exploring KAN combinations to improve model performance.

7. Questions + Answers

- Thank you for listening.
- GitHub: https://github.com/hoangthangta/BSRBF_KAN