Kolmogorov Arnold Networks for Class Incremental Learning

Shashwat Roy (B21CS075), Sukriti Goyal (B21CS075)

Kolmogorov-Arnold Representation Theorem

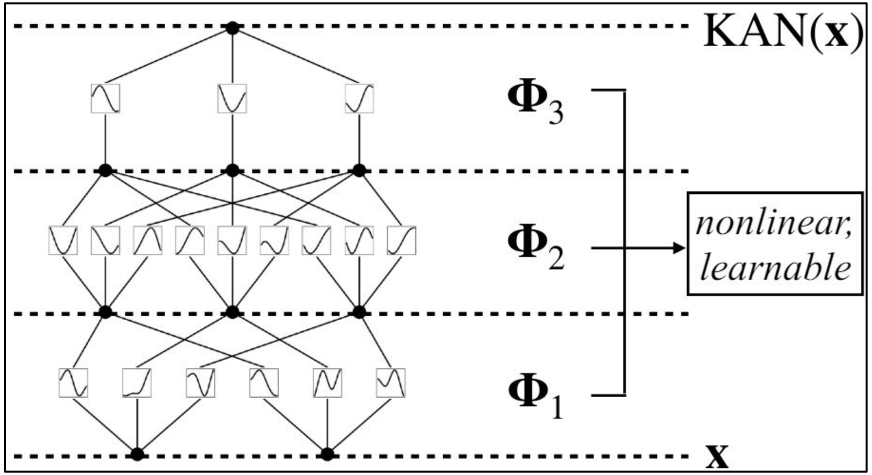
$$f(\mathbf{x}) = f(x_1, \ldots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right).$$

where $\phi_{q,p} \colon [0,1] \to \mathbb{R}$ and $\Phi_q \colon \mathbb{R} \to \mathbb{R}$.

- If f is a multivariate continuous function on a bounded domain, then f can be written as a finite composition of continuous functions of a single variable and the binary operation of addition.
- $f(x_1, \ldots, x_n)$ is a multivariate function.
- $\phi_{q,p}$ are univariate functions.
- Φ_q takes the univariate functions and combines them.

Architecture

- While MLPs have fixed activation functions on nodes (or “neurons”), KANs have learnable activation functions on edges (or “weights”).
- In a KAN, each weight parameter is replaced by a univariate function, typically parameterized as a spline. As a result, KANs have no linear weights at all.
- Since all functions to be learned are univariate functions, we can parametrize each 1D function as a B-spline curve, with learnable coefficients of local B-spline basis functions



Learnable Functions

- A KAN layer comprising an input of dimension n_{in} and output of dimension n_{out} can be defined as a matrix of 1-D functions.
- The activation function ϕ(x) is the sum of the basis function b(x) and the spline function.
- spline(x) is parametrized as a linear combination of B-splines.
- c_i is trainable.

$$\boxed{\mathbf{\Phi} = \{\phi_{q,p}\}, \qquad p = 1, 2, \cdots, n_{\text{in}}, \qquad q = 1, 2 \cdots, n_{\text{out}}}$$

$$\boxed{\phi(x) = w_b b(x) + w_s \text{spline}(x)}$$

$$\boxed{b(x) = \text{silu}(x) = x/(1 + e^{-x})}$$

$$\boxed{\text{spline}(x) = \sum_i c_i B_i(x)}$$

Why B-spline?

- Splines are accurate for low-dimensional functions, easy to adjust locally, and can switch between different resolutions
- The flexibility of splines allows them to adaptively model complex relationships in the data by adjusting their shape using coefficients.

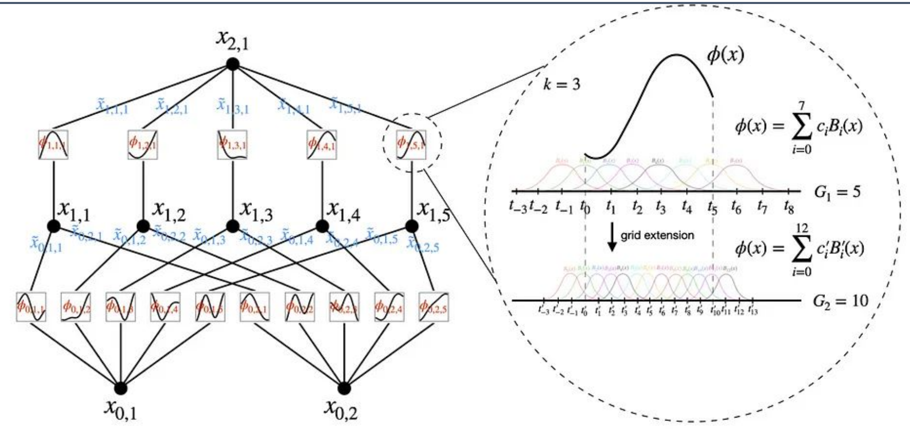
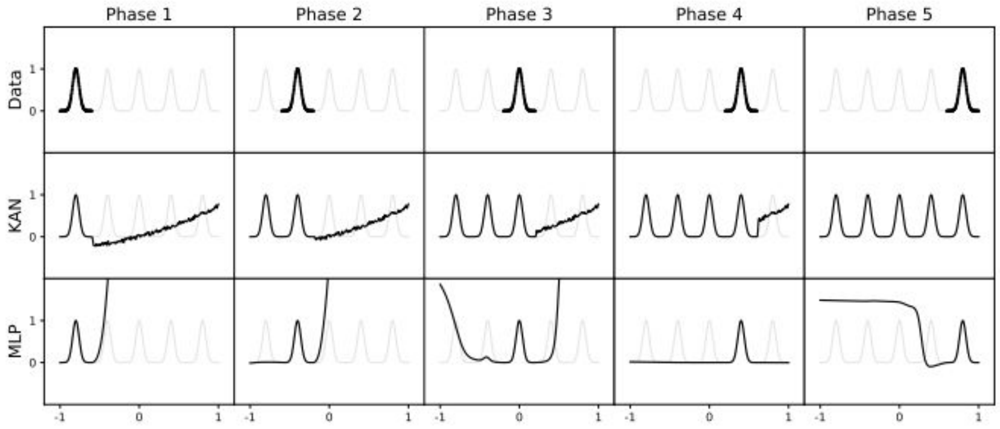


Figure 2.2: Left: Notations of activations that flow through the network. Right: an activation function is parameterized as a B-spline, which allows switching between coarse-grained and fine-grained grids.

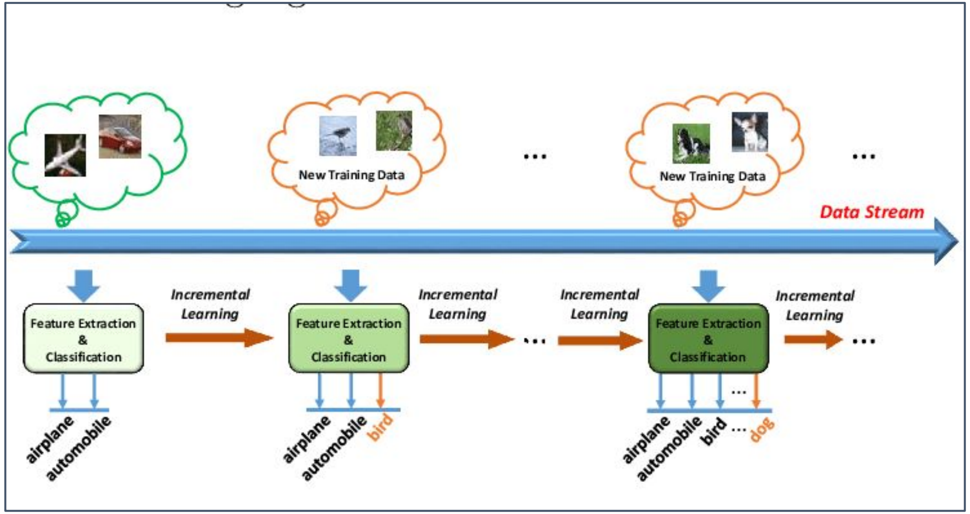
$$\text{spline}(x) = \sum_i c_i B_i(x)$$

Advantages over MLP

- KANs have local plasticity and can avoid catastrophic forgetting by leveraging the locality of splines
- The spline bases have local control , i.e. a sample will only affect the nearby local spline coefficients.
- This is not the case with MLPs which use global activation functions like ReLU and Tanh.

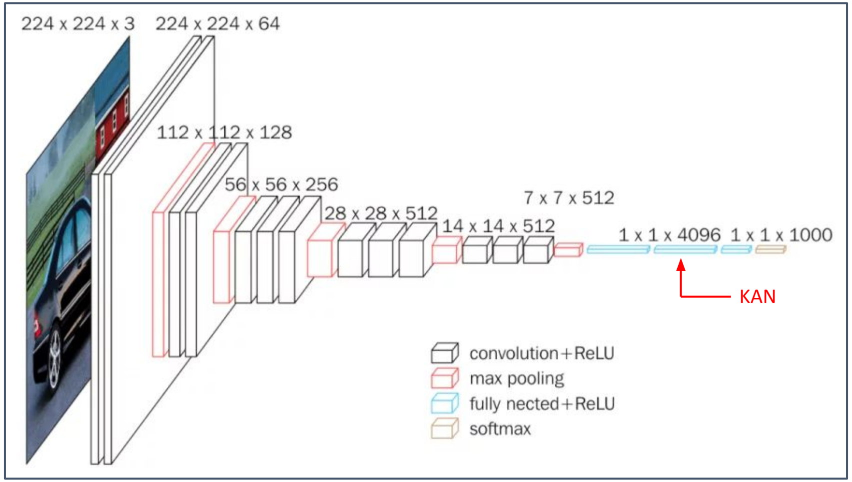


Class Incremental Learning

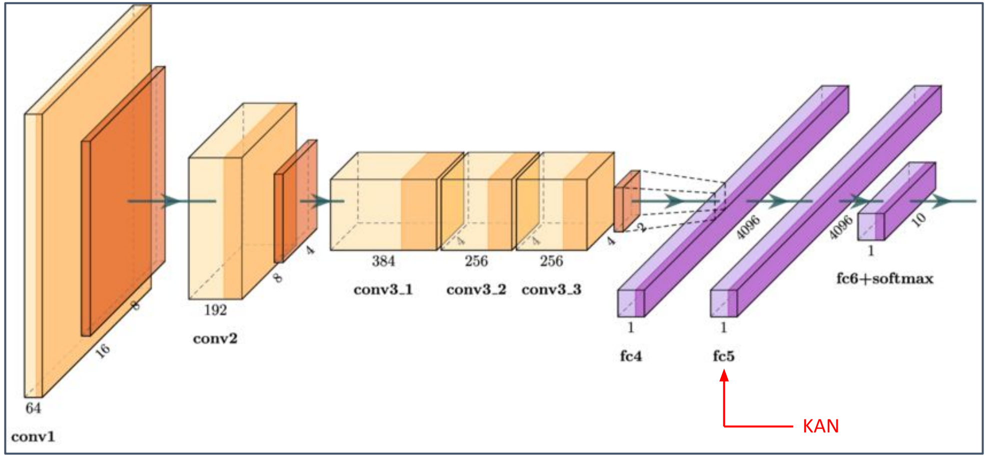


- Class-Incremental Learning (CIL) is a machine learning paradigm designed to enable models to learn and adapt continually in scenarios where new classes of data emerge over time.
- The model must be able to learn the features of the incoming newer classes without affecting its ability to discriminate the earlier classes.

Model Architectures



Vgg16



AlexNet

Experiments

Learning Rate	1-Layer KAN	MLP
0.001	31.927	32.884
0.01	25.766	26.613

Table 1. Results of LwF for experiments with AlexNet architecture on CIFAR100 dataset

Epochs	MLP		KAN	
	CNN	NME	CNN	NME
50	63.31	66.88	63.49	69.25
50	65.28	69.25	64.824	69.74
100	-	-	62.5	65.7

Table 2. Results of iCarl with AlexNet architecture on CIFAR100 with learning rate = 0.001 and weight decay = 0.001

Experiments

Model	Epoch	Learning Rate	Decay	1-Layer KAN	2-Layer KAN	MLP
AlexNet	400	0.001	0.1	24.24	23.128	23.378
AlexNet	400	0.01	0.005	22.57	23.27	Not Available
AlexNet	400	0.05	0.01	23.699	24.58	23.34
AlexNet	200	0.0005	0.1	22.16	Not Needed	21.8
AlexNet	200	0.0001	0.1	14.895	Not Needed	17.49
AlexNet	200	0.01	0.5	24.044	20.029	23.838

Table 3. Results of LwF for experiments with AlexNet on CUB200 dataset (100 classes).

Model	Epoch	Learning Rate	Decay	1-Layer KAN	MLP
AlexNet	50/75	0.001	0.1	32.763	34.57
AlexNet	100/150	0.001	0.1	35.81	34.7665
AlexNet	100/150	0.01	0.001	30.483	29.89
VGG16	75/100	0.001	0.1	38.78	37.87
VGG16	75/100	0.001	0.1	12.63	12.47
VGG16	75/100	0.001	0.1	41.21	38.98

Table 4. Results of LwF for experiments on CUB200 dataset (100 classes) with AlexNet and VGG16 pre-trained on ImageNet

Model	Epoch	Learning Rate	Decay	1-Layer KAN	MLP
AlexNet	150	0.001	0.1	15.46	21.49
AlexNet	250	0.005	0.01	38.2	41.6
VGG16	50/75	0.005	0.01	48.55	50.285
VGG16	50/75	0.001	0.1	64.77	66.33
VGG16	100/150	0.001	0.1	69.79	70.1
VGG16	250	0.001	0.1	68.18	56.76

Table 5. Results of LwF with AlexNet and VGG16 architectures pre-trained on ImageNet for experiments with the Scenes dataset

Future Work

- Prompt-based CIL : Learn to Prompt , APG
- Incorporating convolutional KAN layers : Resnet , Densenet etc.
- Further hyperparameter tuning

References

- **KAN: Kolmogorov-Arnold Networks** : Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, Max Tegmark
- **PyCIL: a Python toolbox for class-incremental learning** : Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye De-Chuan Zhan