Evidential Kolmogorov-Arnold Networks

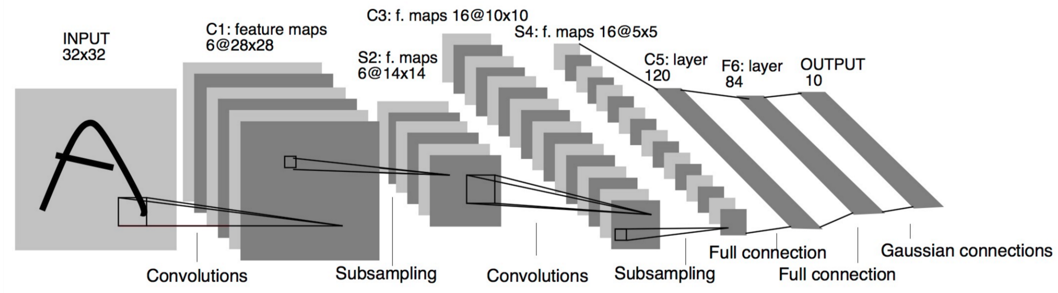
Transfer Learning Using Dempster–Shafer Layers

Alejandro Veloz

Rodrigo Pizarro

ConvNets for image classification

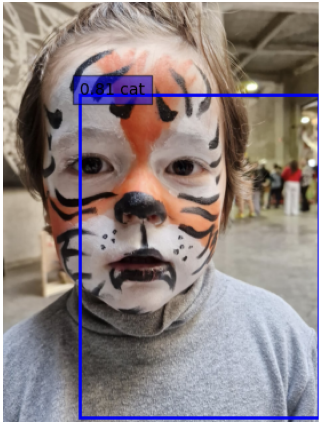
CNN = Convolutional Neural Networks = ConvNet



LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition.

ConvNets for image classification

The model always outputs the class it was trained on (cat, with confidence 0.81).



Outline

1. Dempster-Shafer theory
2. Kolmogorov-Arnold Networks (KAN)
3. Evidential KAN
4. Preliminary results

Uncertainty

Lack of knowledge about a system or process.

Random uncertainty:

- Represents intrinsic variation.
- Can be reduced adding more data.
- Can be addressed by Bayesian learning.

Epistemic uncertainty:

- Represents the lack of knowledge.
- Can be reduced by acquiring more knowledge or training better models.
- Can be addressed by **Dempster-Shafer theory.**

Mass functions

Let $\Omega = \{\omega_1, \ldots, \omega_M\}$ be a finite set of states called the **frame of discernment.**

Let 2^Ω be the set of all subsets of Ω, that is, $2^\Omega = \{A : A \subseteq \Omega\}$.

A mass function is a function $m : 2^\Omega \to [0, 1]$ such that

$$m(\emptyset) = 0, \qquad \sum_{A \subseteq \Omega} m(A) = 1.$$

Mass functions - example

Let X be the type of object in a region of an image and $\Omega = \{$G, R, T, O, S$\}$ the possible classes corresponding to grass, road, tree, obstacle, and sky.

Mass functions - example

Suppose a radar provides the information that $X \in \{\text{T}, \text{O}\}$, but there is a probability $p = 0.1$ that the information is unreliable.

Mass functions - example

Note that the probability p does not provide information about X, but rather about the sensor.

Let $S = \{\text{working}, \text{faulty}\}$ denote the possible states of the sensor.

- If the sensor is working, then $X \in \{\mathsf{T}, \mathsf{O}\}$.
- If the sensor is faulty, then $X \in \Omega$ and nothing else can be determined.

Mass functions - example

This uncertainty in the information can be represented by the following mass function m over Ω:

$$m(\{\mathsf{T}, \mathsf{O}\}) = 0.9, \quad m(\Omega) = 0.1$$

We can conclude that:

- $m(\{\mathsf{T}, \mathsf{O}\})$ is the probability of only knowing that $X \in \{\mathsf{T}, \mathsf{O}\}$ and nothing more.
- $m(\Omega)$ is the probability of knowing nothing at all.

Belief and Plausibility Functions

Given a mass function m and a subset $A \subseteq \Omega$.

The total belief of A, Bel : $2^\Omega \to [0,1]$, is:

$$\text{Bel}(A) = \sum_{E \subseteq A, E \neq \emptyset} m(E).$$

The plausibility of A, Pl : $2^\Omega \to [0,1]$, is:

$$\text{Pl}(A) = \sum_{E \cap A \neq \emptyset} m(E) = 1 - \text{Bel}(\bar{A}).$$

Belief and Plausibility Functions - example

Based on the previous example, it follows that:

$$\Omega = \{\text{G}, \text{R}, \text{T}, \text{O}, \text{S}\}, \quad m(\{\text{T}, \text{O}\}) = 0.9, \quad m(\Omega) = 0.1$$

Belief and plausibility values of some subsets of Ω:

A	$\emptyset$	$\{\text{T}\}$	$\{\text{O}\}$	$\{\text{T}, \text{O}\}$	$\{\text{T}, \text{O}, \text{R}\}$	$\{\text{T}, \text{R}\}$	$\{\text{R}, \text{S}\}$	Ω
Bel(A)	0	0	0	0.9	0.9	0	0	1
Pl(A)	0	1	1	1	1	1	0.1	1

Dempster's Combination Rule

Suppose that m_1 and m_2 are mass functions over Ω.
The combined function $m : 2^\Omega \to [0,1]$ defined by and

$$m(A) = (m_1 \oplus m_2)(A) = \frac{1}{1-\kappa} \sum_{B \cap C = A} m_1(B) m_2(C),$$

where

$$\kappa := \sum_{B \cap C = \emptyset} m_1(B) m_2(C) < 1,$$

is a valid mass function.

Kolmogorov–Arnold representation theorem (1957)

Kolmogorov–Arnold representation theorem (1957)

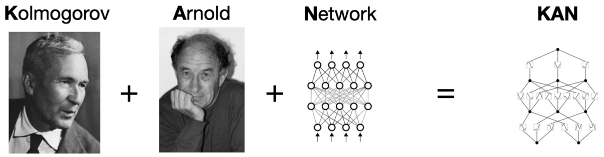
Any multivariate continuous function can be represented (or decomposed) as a finite superposition of continuous univariate functions.

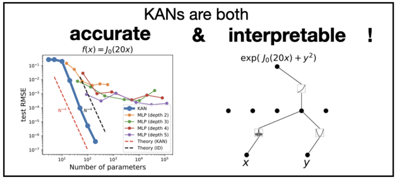
For any smooth function $f : \mathbb{R}^n \to \mathbb{R}$,

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

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

Kolmogorov-Arnold Networks

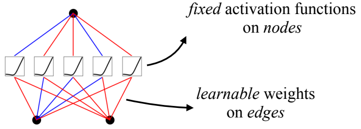
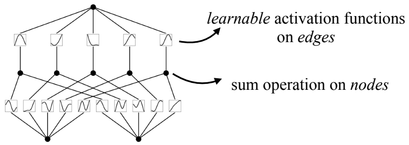
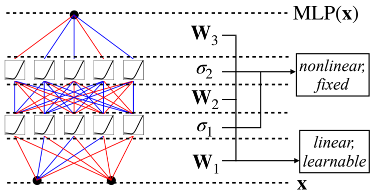
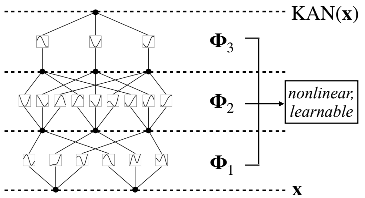




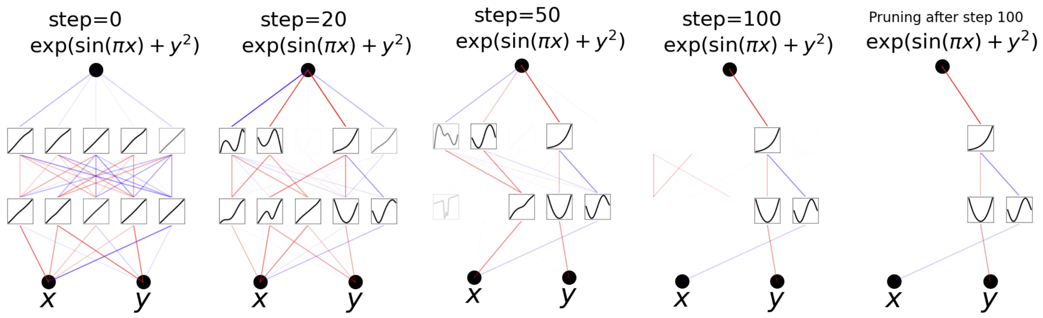
- MLPs have fixed activation functions on nodes ("neurons").
- KANs have **learnable activation functions on edges** ("weights").

For interpretability, KANs can be intuitively visualized and can easily interact with human users.

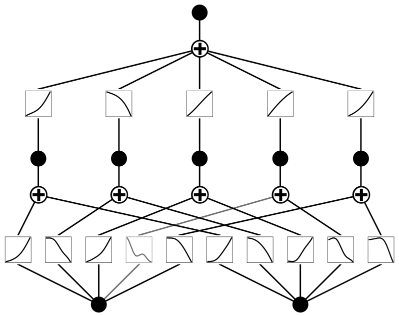
Liu et al. (2024). KAN: Kolmogorov–Arnold Networks.

Model	Multi-Layer Perceptron (MLP)	Kolmogorov-Arnold Network (KAN)
Theorem	**Universal Approximation Theorem**	**Kolmogorov-Arnold Representation Theorem**
Formula (Shallow)	$f(\mathbf{x}) \approx \sum_{i=1}^{N(\epsilon)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$	$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$
Model (Shallow)	**(a)**  *fixed* activation functions on *nodes* *learnable* weights on *edges*	**(b)**  *learnable* activation functions on *edges* sum operation on *nodes*
Formula (Deep)	$\text{MLP}(\mathbf{x}) = (\mathbf{W}_3 \circ \sigma_2 \circ \mathbf{W}_2 \circ \sigma_1 \circ \mathbf{W}_1)(\mathbf{x})$	$\text{KAN}(\mathbf{x}) = (\mathbf{\Phi}_3 \circ \mathbf{\Phi}_2 \circ \mathbf{\Phi}_1)(\mathbf{x})$
Model (Deep)	**(c)**  MLP(**x**) $\mathbf{W}_3$ σ_2 $\mathbf{W}_2$ σ_1 $\mathbf{W}_1$ **x** *nonlinear, fixed* *linear, learnable*	**(d)**  KAN(**x**) $\mathbf{\Phi}_3$ $\mathbf{\Phi}_2$ $\mathbf{\Phi}_1$ **x** *nonlinear, learnable*

Kolmogorov-Arnold Networks



Kolmogorov-Arnold Networks



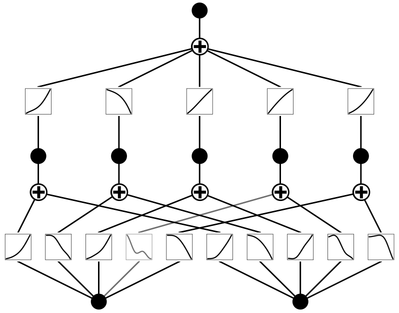
KAN network dimensions: $[n_0, n_1, \ldots, n_L]$.

The activation value of neuron (ℓ, i) in layer ℓ is $x_{\ell,i}$.

Between layers ℓ and $\ell+1$, each pair of neurons (ℓ, i) and $(\ell+1, j)$ is connected by a univariate function $\phi_{\ell,j,i}$.

$$\ell = 0, \ldots, L-1,$$
$$i = 1, \ldots, n_\ell,$$
$$j = 1, \ldots, n_{\ell+1}.$$

Kolmogorov-Arnold Networks



Forward pass

Each connection applies its own function:

$$\tilde{x}_{\ell,j,i} = \phi_{\ell,j,i}(x_{\ell,i}).$$

Each neuron in the next layer sums incoming activations:

$$x_{\ell+1,j} = \sum_{i=1}^{n_\ell} \phi_{\ell,j,i}(x_{\ell,i}).$$

Kolmogorov-Arnold Networks

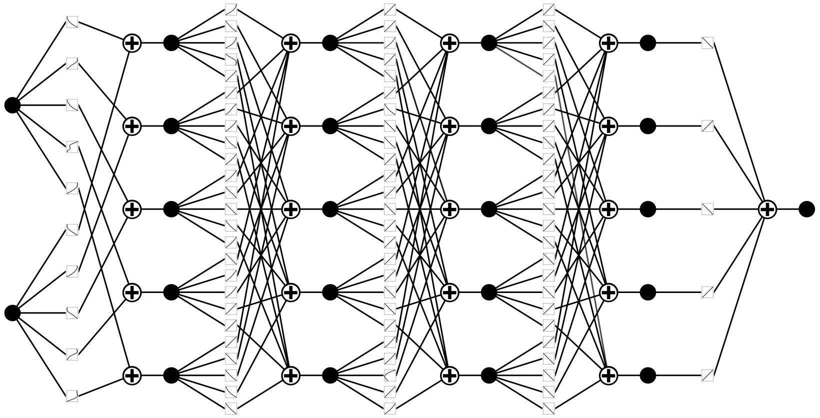
In matrix form:

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

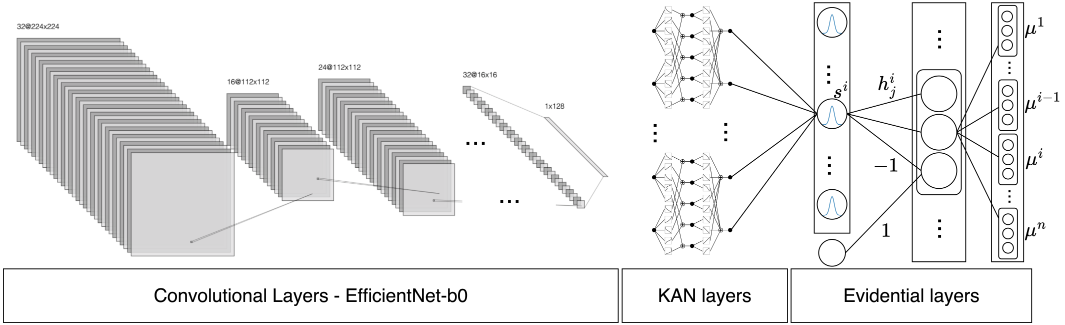
where $\mathbf{\Phi}_l$ is the matrix of the l-th KAN layer.

Multilayer KAN

$$\text{KAN}(\mathbf{x}) = (\mathbf{\Phi}_{L-1} \circ \cdots \circ \mathbf{\Phi}_1 \circ \mathbf{\Phi}_0)(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^{n_0}.$$



E-KAN: Evidential KAN Classifier



	Operation	$\hat{H}_i \times \hat{W}_i$	$\hat{C}_i$	#Layers
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3×3	112×112	24	2
4	MBConv6, k5×5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5×5	14×14	112	3
7	MBConv6, k5×5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1 $\times 1$ & Pooling & FC	7×7	1280	1

Evidential Layers

Let us consider the input feature vector $\mathbf{x} \subseteq \mathbb{R}^P$.

An evidential classifier consists of n prototypes, $\{\mathbf{p}^1, \ldots, \mathbf{p}^n\}$, in $\mathbb{R}^P$.

An evidential layer constructs mass functions to quantify uncertainty about classes $\omega \in \Omega = \{\omega_1, \ldots, \omega_M\}$ following a three-step scheme.

Evidential Layers

1. The support between $\mathbf{x}$ and each prototype $\mathbf{p}^i$, $i \in \{1, \cdots, n\}$, is:

$$s^i = \tau^i \exp\left(-\left(\eta^i \left\|\mathbf{x} - \mathbf{p}^i\right\|\right)^2\right), \quad \tau^i \in (0,1),\ \eta^i \in \mathbb{R}$$

2. The mass function m^i associated to $\mathbf{p}^i$ is:

$$\begin{aligned} m^i\left(\{\omega_j\}\right) &= h^i_j s^i, \quad j \in \{1, \cdots, M\} \\ m^i(\Omega) &= 1 - s^i \end{aligned}$$

where h^i_j is the degree of membership of $\mathbf{p}^i$ to class ω_j, with $\sum_{j=1}^{M} h^i_j = 1$. This yields:

$$\mathbf{m}^i = \left(m^i\left(\{\omega_1\}\right), \cdots, m^i\left(\{\omega_M\}\right), m^i(\Omega)\right)^{\mathsf{T}}.$$

3. The n mass functions $\mathbf{m}^i$ are combined using Dempster's rule.

$$\begin{array}{llll}
m^1(\{\omega_1\}) & m^2(\{\omega_1\}) & \dots & m^n(\{\omega_1\}) \\
m^1(\{\omega_2\}) & m^2(\{\omega_2\}) & \dots & m^n(\{\omega_2\}) \\
\vdots & \vdots & \ddots & \vdots \\
m^1(\{\omega_M\}) & m^2(\{\omega_M\}) & \dots & m^n(\{\omega_M\}) \\
m^1(\Omega) & m^2(\Omega) & \dots & m^n(\Omega)
\end{array}$$

[Mass functions combination]

$$\mu^i\left(\{\omega_j\}\right) = \begin{cases} m^1\left(\{\omega_j\}\right), & \text{for } i = 1 \\ \mu^{i-1}\left(\{\omega_j\}\right) \oplus m^i\left(\{\omega_j\}\right), & \text{for } i > 1 \end{cases}$$

where

$$\begin{aligned}\mu^{i-1}\left(\{\omega_j\}\right) \oplus m^i\left(\{\omega_j\}\right) = {} & \mu^{i-1}\left(\{\omega_j\}\right) m^i\left(\{\omega_j\}\right) \\ & + \mu^{i-1}\left(\{\omega_j\}\right) m^i(\Omega) + \mu^{i-1}(\Omega) m^i\left(\{\omega_j\}\right)\end{aligned}$$

The output vector of the evidential layers is given by:

$$\mathbf{m} = \left(m\left(\{\omega_1\}\right), \ldots, m\left(\{\omega_M\}\right), m(\Omega)\right)^{\mathsf{T}},$$

and is obtained through the following relations:

$$m\left(\{\omega_j\}\right) = \frac{\mu^n\left(\{\omega_j\}\right)}{\sum_{k=1}^{M} \mu^n\left(\{\omega_k\}\right) + \mu^n(\Omega)}$$

and

$$m(\Omega) = \frac{\mu^n(\Omega)}{\sum_{k=1}^{M} \mu^n\left(\{\omega_k\}\right) + \mu^n(\Omega)}.$$

Decision-Making Criteria

- **Decision problem:** An entity must choose a course of action (act) from a set $\mathcal{F} = \{f_1, \cdots, f_M\}$.
- Each of these decisions has a consequence drawn from $\mathcal{C} = \{c_1, \cdots, c_M\}$.
- These decisions are taken from the states $\Omega = \{\omega_1, \cdots, \omega_M\}$.

In particular, an act is a function $f : \Omega \to \mathcal{C}$.

Utility Function

The function $u : \mathcal{C} \to \mathbb{R}$ assigns a real-valued number to every consequence.

- A higher value of u indicates a better decision.
- $c_{ij} = f_i(\omega_j)$ represents the consequence of choosing act f_i when state ω_j occurs.
- $u_{ij} = u(c_{ij})$ denotes the corresponding utility.

Decision-Making Criteria

Consider the following sets:

- $\Omega = \{\omega_1, \cdots, \omega_M\}$ [classes]
- $\mathcal{F} = \{f_{\omega_1}, \cdots, f_{\omega_M}\}$ [acts]

Each act f_{ω_i}, which represents assigning class ω_i, defines the lower and upper expected values of the utility function as:

$$\underline{\mathbb{E}}_m\left(f_{\omega_i}\right) = \sum_{B \subseteq \Omega} m(B) \min_{\omega_j \in B} u_{ij}$$

$$\overline{\mathbb{E}}_m\left(f_{\omega_i}\right) = \sum_{B \subseteq \Omega} m(B) \max_{\omega_j \in B} u_{ij}$$

Decision-Making Criteria

Pessimistic preference-based decision rule:

$$f_{\omega_i} \succcurlyeq_* f_{\omega_j} \iff \underline{\mathbb{E}}_m\left(f_{\omega_i}\right) \geq \underline{\mathbb{E}}_m\left(f_{\omega_j}\right),$$

Optimistic preference-based decision rule:

$$f_{\omega_i} \succcurlyeq^* f_{\omega_j} \iff \overline{\mathbb{E}}_m\left(f_{\omega_i}\right) \geq \overline{\mathbb{E}}_m\left(f_{\omega_j}\right),$$

Decision-Making Criteria

Pignistic transformation, that distributes the mass equally among all elements of $\mathcal{C}$ (Smets 1990):

$$\mathrm{BetP}_m\left(\omega_j\right) = \sum_{A \subseteq \Omega, \omega_j \in A} \frac{m(A)}{|A|}, \forall \omega_j \in \Omega.$$

Thus, the criterion is defined by maximizing the quantity $\mathbb{E}_p(f_{\omega_i}) = \sum_{j=1}^{M} \mathrm{Bet}\, P_m(\omega_j)\, u_{ij}$.

Experimental evaluation

Datasets:

- MNIST
- CIFAR-10

Metric:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\left(y_{\text{pred}_i} = y_{\text{true}_i}\right),$$

where $\mathbb{1}$ is the indicator function.

Results

[MNIST]

Model	Accuracy (%)	# parameters (trainable)
MLP	98.13	101770
CNN	99.50	824458
MLPKAN	98.53	298176
CNNKAN	99.35	1162368
EfficientNetKan	93.89	258400
Evidential EfficientNet	92.13	3302400
E-KAN	94.35	4692320

Results

[CIFAR-10]

Model	Accuracy (%)	# parameters (trainable)
MLP	45.29	394634
CNN	72.82	1070794
MLPKAN	49.55	591040
CNNKAN	72.01	1408704
EfficientNetKan	82.27	3302400
Evidential EfficientNet	83.32	258400
E-KAN	84.43	14584160

Results

- For MNIST, implementing the three proposed models did not lead to any performance improvement.
- For CIFAR-10, there was a significant improvement compared to the baseline models.

This difference likely arises because MNIST consists of grayscale digit images, while EfficientNet-b0 was pretrained on ImageNet, a large dataset of high-resolution color images. Therefore, its learned features do not transfer effectively to MNIST.
In contrast, CIFAR-10 contains RGB images of diverse objects, making it more similar to ImageNet and allowing the pretrained features to generalize better.