Kolmogorov Arnold Networks

Hou, Y., Zhang, D., (2024). **A Comprehensive Survey on Kolmogorov Arnold Networks (KAN)**

Overview

- Kolmogorov (RUS) & Arnold (UKR) have posited that any continuous multivariate function (*one in which a small variation in function implies a small variation in result*), can be represented by a finite number of univariate functions (*only one variable*), and those univariates can replace the weight parameters of a neural network.
- The theorem, proposed in 1957, is based on several mathematical theorems involving numerical analysis and partial differential equations.
- The advantages are improvements in data fitting and complex learning tasks, as well as performance with high dimensional data in their pursuit of capturing complex temporal dependencies.

Key Differences: Standard MLP v. KAN

MLP

- Goal is to optimize based on learned parameters w & b
- Predefined activation function

KAN

- Goal is to optimize based on learned coefficients of spline, with unfixed functions
- Flexible activation function (spline)

Main Goal of KAN: Kolmogorov-Arnold Networks (KAN) enhance their ability to handle nonlinear relationships by using spline functions to implement complex nonlinear transformations on each edge.

Mathematical Expression of Theorem

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{2n+1} g_i \left(\sum_{j=1}^{n} h_{ij}(x_j) \right)$$

Where:
x = sample data
n = dimensions
g = univariate function
h = univariate function

$$S_i(x) = a_i + b_i \cdot (x - x_i) + c_i \cdot (x - x_i)^2 + d_i \cdot (x - x_i)^3$$

Where:
x = any value along spline interval
xi = knot point
a-d = coeffs

Reference

Mathematical Expression In Practice

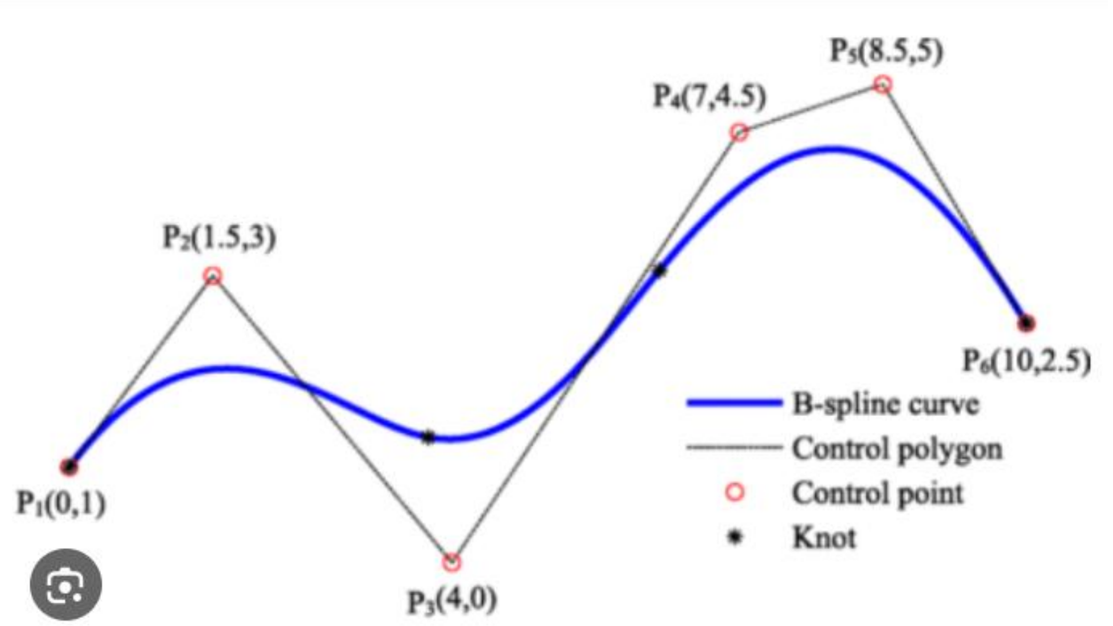
$$B_{i,0}(x) = 1, \text{if } t_i \leq x < t_{i+1}, \text{otherwise } 0,$$

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

where,

x	=	data point
t	=	knots
k	=	exponential degree

Reference



Reference

Basic MLP Example

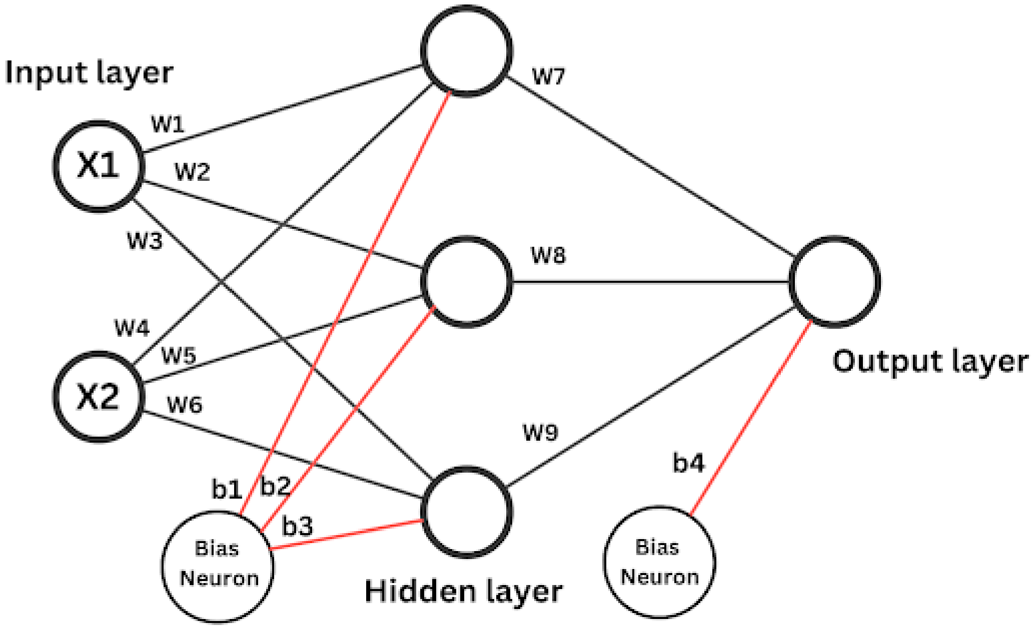


Fig 4 (MLP Weights and Baises)

[Reference]()

Model Parameter Comparison

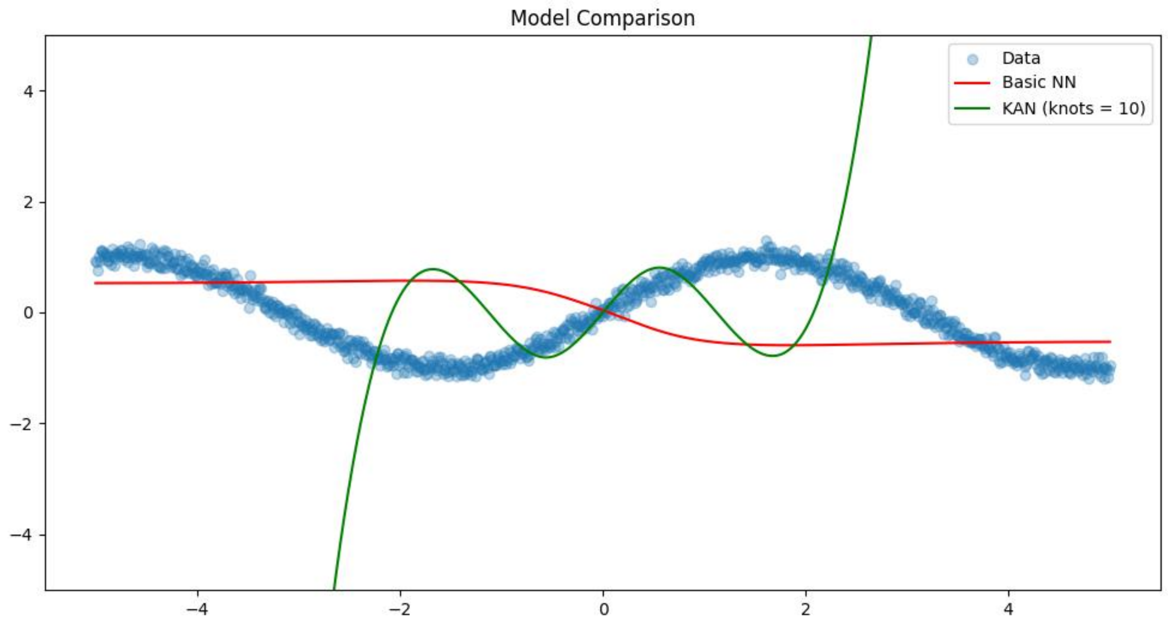
Basic Neural Network

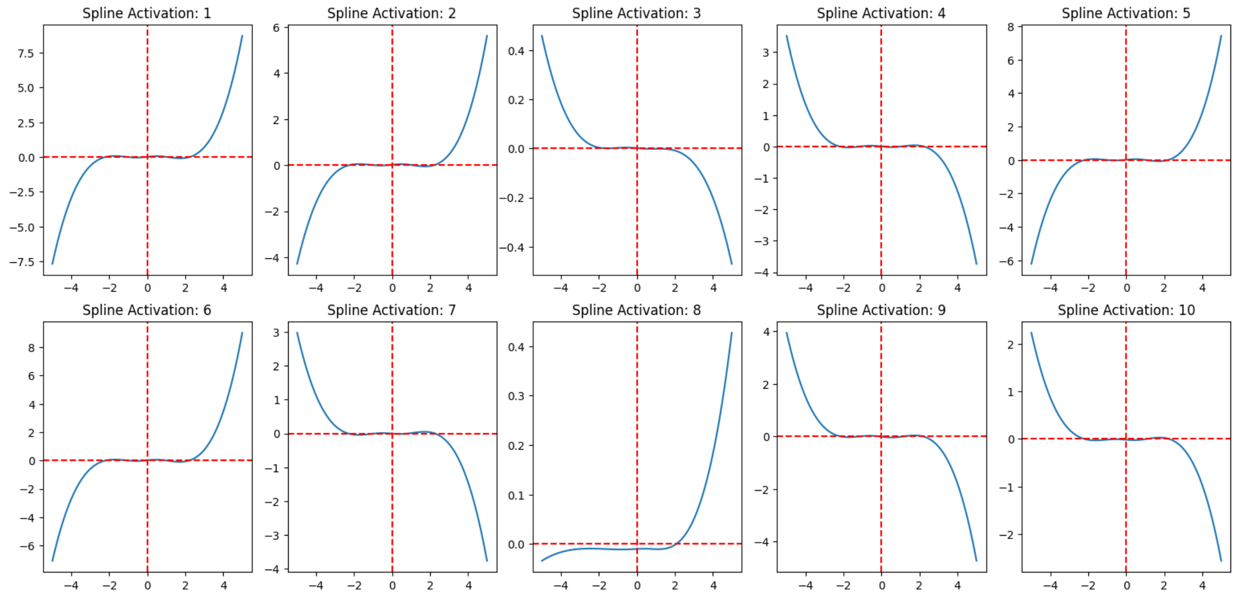
```
['W1', 'b1', 'W2', 'b2']
[[ 0.60856746  0.94646912  1.36256079 -0.80811515  1.68428053  1.45438326
   0.25701588  0.10033642  1.27488093  1.49932301]]
[[-0.00685603  0.00436755 -0.00049083 -0.03636161  0.00264737 -0.00041073
   0.00596033  0.09783393  0.00030786  0.00086073]]
[[ 0.17837218]
 [-0.18783807]
 [ 0.26433984]
 [ 0.38592123]
 [ 0.4928461 ]
 [-0.32040121]
 [-0.4879605 ]
 [ 0.69860776]
 [-0.17076239]
 [ 0.38719292]]
[[-0.05554404]]
```

KAN

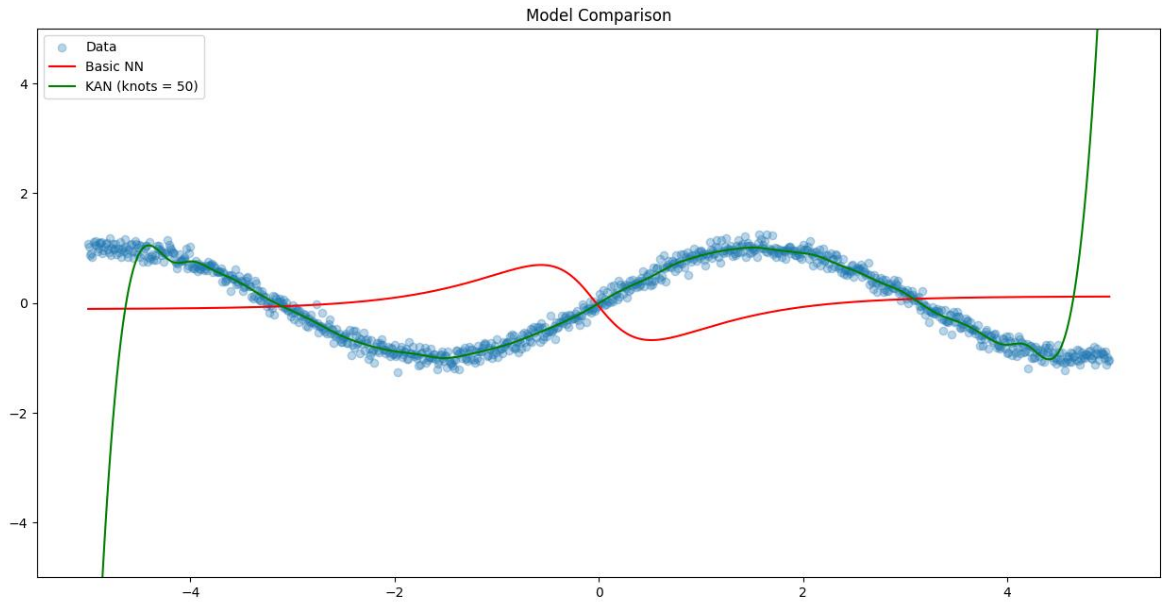
```
['knots', 'coeffs', 'W', 'b']
[-5.         -3.88888889 -2.77777778 -1.66666667 -0.55555556  0.55555556
  1.66666667  2.77777778  3.88888889  5.        ]
[[ 1.00000000e+00 -1.00000000e+00  1.00000000e+00 -1.00000000e+00
   1.00000000e+00 -1.00000000e+00  8.21284089e-03  2.03309747e-03
  -1.52311071e-02 -1.51004496e-04]
 [ 9.48938837e-01 -1.00000000e+00  1.00000000e+00 -1.00000000e+00
   1.00000000e+00 -9.48908774e-01  1.14880059e-03 -8.31089032e-03
  -3.74943916e-03  4.69016505e-03]
 [ 2.16978030e-01 -2.14890994e-01 -2.14332426e-01  2.14545746e-01
   2.14893720e-01 -1.71872616e-01  4.53034001e-03 -1.14413120e-02
   6.18545797e-03  7.43075008e-03]
 [-5.97341235e-02 -1.86845850e-01 -1.74881709e-01  1.63154871e-01
   2.12948328e-01  9.64298628e-02 -8.49678504e-04 -6.23140917e-03
   2.57250080e-02 -1.82942266e-03]
 [ 1.00000000e+00 -1.00000000e+00  1.00000000e+00 -1.00000000e+00
   1.00000000e+00 -1.00000000e+00  2.22294817e-03 -9.96355659e-03
  -1.87966794e-03  1.78938473e-02]
 [ 1.33624523e-02  1.42450332e-02 -9.94085883e-03 -2.07696589e-02
   8.77440278e-03  6.82837422e-03 -3.74824117e-03 -1.70008327e-02
  -7.39738265e-03 -1.82690510e-02]
 [-1.00000000e+00  1.00000000e+00 -1.00000000e+00  1.00000000e+00
  -1.00000000e+00  1.00000000e+00  7.03913329e-03 -6.88053452e-03
  -1.60209847e-02  4.53066129e-03]
 [-1.00000000e+00  1.00000000e+00 -1.00000000e+00  1.00000000e+00
  -1.00000000e+00  1.00000000e+00  3.25165409e-03 -1.14205974e-03
  -1.24350893e-02  1.43648394e-02]
 [-2.63353093e-03 -7.15177796e-03 -9.84193566e-03 -8.93985862e-03
   2.21769810e-03  5.57305592e-04 -1.13821713e-02  2.63695837e-03
   7.93932149e-03  2.91782974e-03]
 [-1.00000000e+00  1.00000000e+00 -1.00000000e+00  1.00000000e+00
  -1.00000000e+00  1.00000000e+00  8.76327808e-03  1.33265177e-02
  -4.01853884e-03 -1.71115533e-03]]
[[-6.39517724e-01]
 [-1.68538399e-01]
 [ 3.62556624e-02]
 [ 2.21791714e-02]
 [-7.12217681e-01]
 [-3.15505350e-06]
 [ 5.80088044e-01]
 [ 7.19485974e-01]
 [ 5.49807206e-06]
 [ 6.53063463e-01]]
[[-0.00307258]]
```

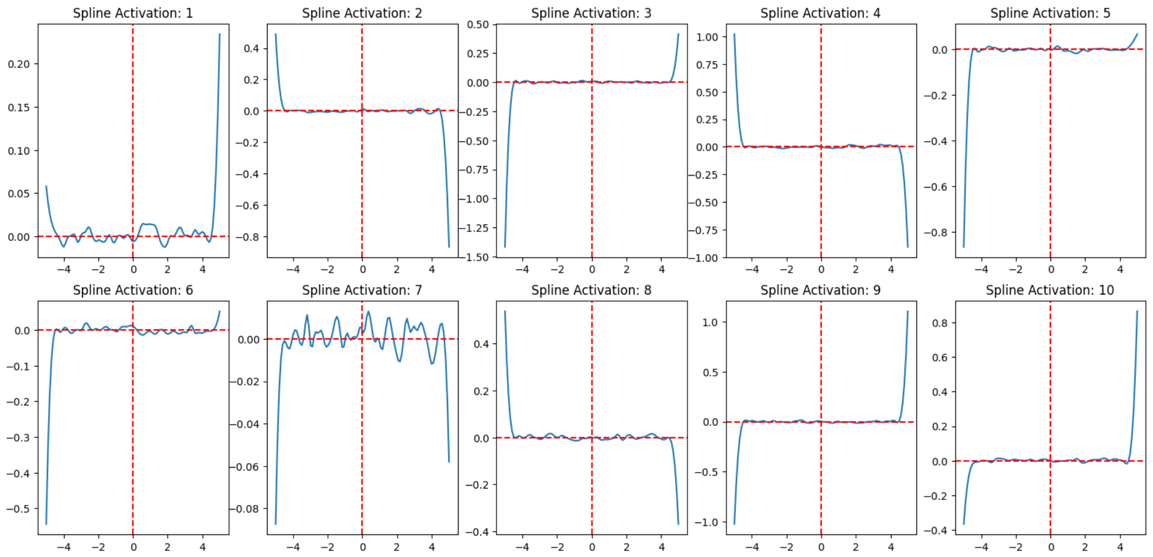
Graphical Display @ 10 knots



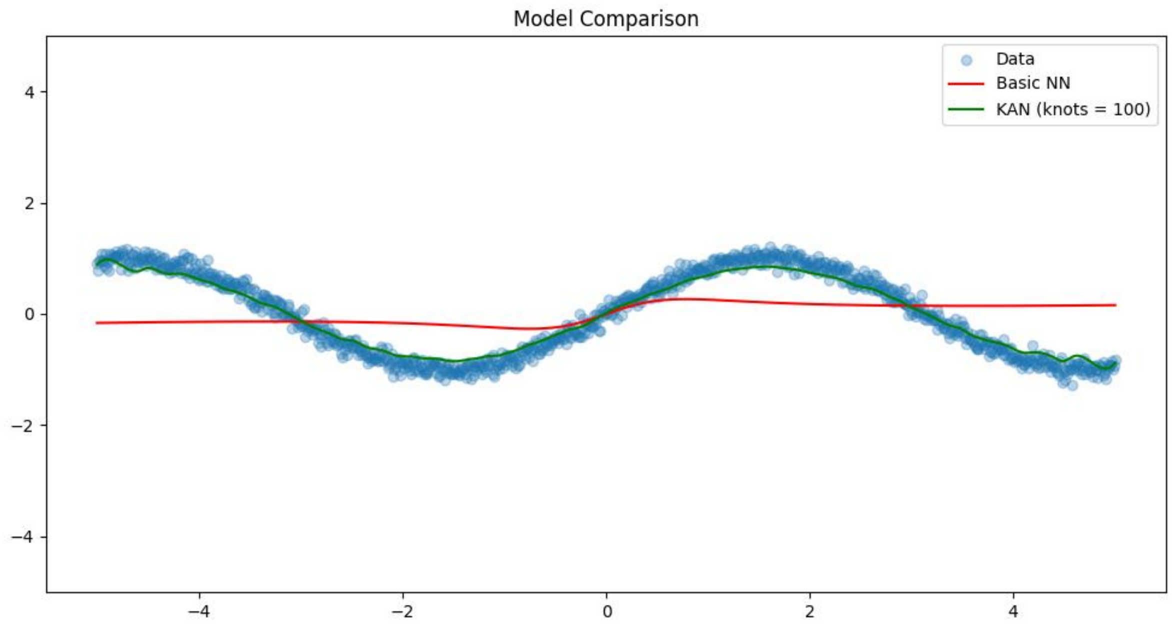


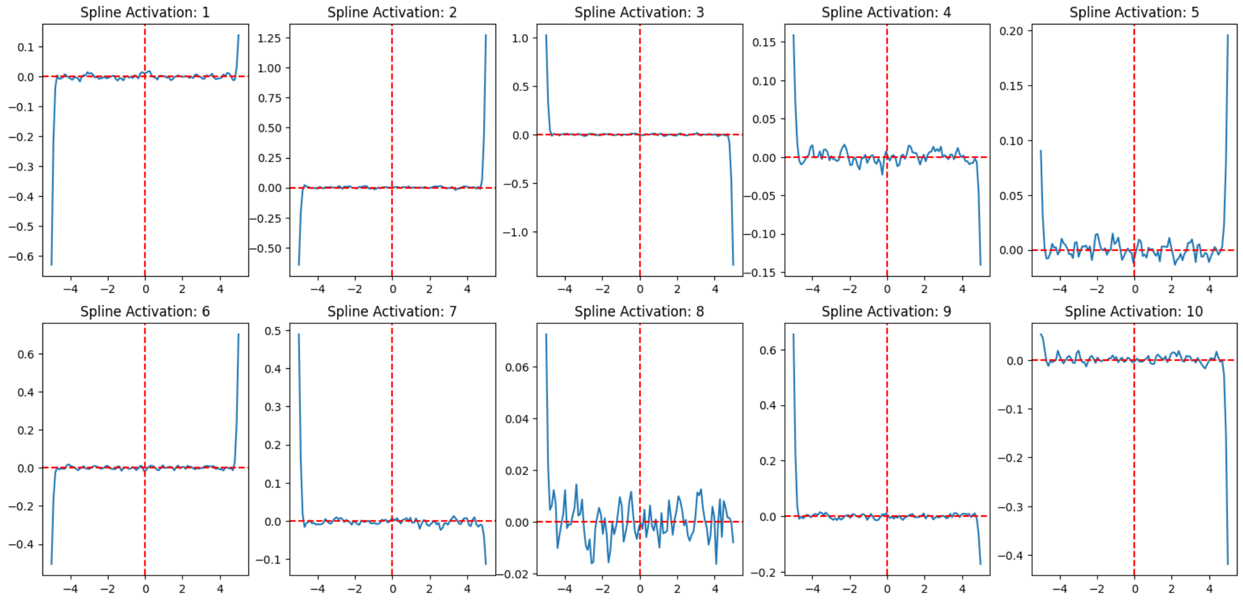
Graphical Display @ 50 knots





Graphical Display @ 100 knots





Advantages & Uses for KAN

Advantages

- Fewer parameters than NN, in theory?
- Can adapt better to high dimensional data
- Better accuracy and interpretability

Uses

- Symbolic Regression (solving for formula)
- Time Series Prediction
- Graph-Structured Data Processing

Source Code

- [Google Drive]
- Questions?