Piecewise linear approximation

Nadezda Sukhorukova, in collaboration with J. Ugon, R. Diaz Millan and V. Peiris

Swinburne University of Technology

nsukhorukova@swin.edu.au

May 30, 2026

Kolmogorov-Arnold theorem

Kolmogorov-Arnold theorem states that every multivariate continuous function can be written as a finite composition of univariate functions and binary operation of addition:

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

The degree of the monomials does not exceed one.
This theorem solved a more constrained, yet more general form of Hilbert's thirteenth problem (continuous part). The algebraic part is still unresolved.

Kolmogorov-Arnold theorem: variants

1. 1962, George Lorentz: the outer functions Φ_q can be replaced by a single function Φ:
2. 1965, David Sprecher replaced the inner functions $\phi_{q,p}$ and, with some restrictions,

$$f(\mathrm{x}) = \sum_{q=0}^{2n} \Phi \left(\sum_{p=1}^{n} \lambda_p \phi(x_p + \eta q) + q \right).$$

There are still some limitations:

1. The theorem does not hold in general for complex multi-variate functions.
2. The non-smoothness of the inner functions has limited the practical use of the representation (we will come back to it).

Neural Network and universal approximators: Universal Approximation Theorem by Cybenko, 1989.

Theorem

Fix a continuous function $\sigma : \mathbb{R} \to \mathbb{R}$ *(activation function, sigmoildal) and positive integers* d, D*, then for every continuous function* $f : \mathbb{R}^d \to \mathbb{R}^D$ *and every* $\epsilon > 0$ *there exists a continuous function* $f_\epsilon : \mathbb{R}^d \to \mathbb{R}^D$ *(the layer output) with representation*
$f_\epsilon = W_2 \circ \sigma \circ W_1$*, where* W_2, W_1 *are affine maps and* $\circ$ *denotes component-wise composition, such that the approximation bound*

$$\sup_{x \in K} \|f(x) - f_\epsilon(x)\| < \varepsilon$$

arbitrarily small (distance from f *to* f_ϵ *can be infinitely small).*

Universal Approximation Theorem: activation function choice

1. Hornik also showed in 1991 that it is not the specific choice of the activation function but the multilayer architecture itself (number of hidden layers, number of nodes in each layer) that gives neural networks the potential of being universal approximators.
2. Allan Pinkus in 1999 showed that the universal approximation property is equivalent to having a nonpolynomial activation function.

Discussion around Kolmogorov-Arnold theorem and its relevance to deep learning

1. F. Girosi and T. Poggio, "Representation Properties of Networks: Kolmogorov's Theorem Is Irrelevant," in Neural Computation, vol. 1, no. 4, pp. 465-469, Dec. 1989, doi: 10.1162/neco.1989.1.4.465.
2. Věra Kůrková . "Kolmogorov's Theorem Is Relevant", https://doi.org/10.1162/neco.1991.3.4.617

Universal Approximation Theorem: Cybenko-Hornik-Pinkus

Theorem

Fix a continuous function $\sigma : \mathbb{R} \to \mathbb{R}$ (activation function) and positive integers d, D. The function σ is not a polynomial if and only if, for every continuous function $f : \mathbb{R}^d \to \mathbb{R}^D$ and every $\epsilon > 0$ there exists a continuous function $f_\epsilon : \mathbb{R}^d \to \mathbb{R}^D$ (the layer output) with representation $f_\epsilon = W_2 \circ \sigma \circ W_1$, where W_2, W_1 are affine maps and $\circ$ denotes component-wise composition, such that the approximation bound

$$\sup_{x \in K} \|f(x) - f_\epsilon(x)\| < \varepsilon$$

arbitrarily small (distance from f to f_ϵ can be infinitely small).

If σ is monotone then $\sigma \circ W_1$ is quasiaffine and f_ϵ the sum of quasiaffine (which may or may not be quasiconvex or quasiaffine itself).

Neural Network: activation function and weights

$$\varphi(W,x)=\sigma\left(\sum_{j=1}^{n}w_jx_j^i+w_0\right), \tag{1}$$

- σ is called *activation function*. This is a chosen function, not subject to optimisation.
- W are weights (subject to optimisation).
- One hidden layer

$$\varphi(\alpha,W,x)=\sum_{i=1}^{N}\alpha_i\sigma\left(\sum_{j=1}^{n}w_jx_j^i+w_0\right), \tag{2}$$

Approximation of univariate functions by neural networks with ReLU and Leaky ReLU activation functions

Artificial Neural Networks (ANNs) can be also seen as an approximation tool.

ANN approximations with ReLU activation functions and a single hidden layer are piecewise linear approximation and **in the case of univariate approximation ANN approximations are equivalent to the piecewise linear approximations**.

Approximation by piecewise linear functions are hard if the location of knots (switches from one linear/affine piece to another) are unknown.

Two methods

- Direct (hard, only possible for a small number of linear pieces). We use MILP-based approach.
- Convex optimisation-based approach. We allow a large number of linear pieces, fix the location of knots and apply convex optimisation tools.

In the case of one knot, the problems can be solved exactly (optimal solution) using MILPP approach.
Our main approach is different (later).

MILP-based approach: Maximum Problem

First, for each discretisation point t_i, $i = 1, \ldots, N$ we introduce a new variable $c_i = \max\{a_1 t_i + b_1, a_2 t_i + b_2\}$, $i = 1, \ldots, N$, where N is the number of discretisation points. The objective is to minimise the absolute deviation z, subject to the following constraints:

$$f(t_i) - c_i \leq z, \; i = 1, \ldots, N, \tag{3}$$

$$c_i - f(t_i) \leq z, \; i = 1, \ldots, N. \tag{4}$$

Then

$$a_1 t_i + b_1 \leq c_i, \; i = 1, \ldots, N, \tag{5}$$

$$a_2 t_i + b_2 \leq c_i, \; i = 1, \ldots N, \tag{6}$$

and for every i, at least one of the inequalities has to be satisfied as equality. This is where we have to introduce a binary variable.

Maximum problem: continue

This can be achieved by requiring that for each group i, at least one of the following reverse inequalities is satisfied:

$$a_1 t_i + b_1 \geq c_i,\ \ i = 1, \ldots, N,$$
$$a_2 t_i + b_2 \geq c_i,\ \ i = 1, \ldots N.$$

For each group, introduce a binary variable z_i. Also consider a larger positive parameter M (big-M, fixed value). Then the requirement for at least one of the inequalities in each group holds can be expressed as follows:

$$c_i - (a_1 t_i + b_1) \leq M z_i,\ \ i = 1, \ldots, N, \tag{7}$$
$$c_i - (a_2 t_i + b_2) \leq M(1 - z_i),\ \ i = 1, \ldots N, \tag{8}$$
$$z_i \in \{0, 1\},\ \ i = 1, \ldots, N. \tag{9}$$

Finally, the goal is to minimise z subject to (3)-(9).

MILPP-based approach: Minimum Problem

Similar to the maximum problem, introduce a new variable $d_i = \min\{a_1 t_i + b_1, a_2 t_i + b_2\}, \; i = 1, \dots, N$, where N is the number of discretisation points. The objective is to minimise the absolute deviation y, subject to

$$f(t_i) - d_i \leq y, \; i = 1, \dots, N, \quad (10)$$

$$d_i - f(t_i) \leq y, \; i = 1, \dots, N. \quad (11)$$

We have the following equations:

$$a_1 t_i + b_1 \geq d_i, \; i = 1, \dots, N, \quad (12)$$

$$a_2 t_i + b_2 \geq d_i, \; i = 1, \dots N. \quad (13)$$

and for every i, at least one of the inequalities has to be satisfied as equality. For each group i, at least one of the following reverse inequalities is satisfied:

$$a_1 t_i + b_1 - d_i \leq 0, \; i = 1, \dots, N,$$

$$a_2 t_i + b_2 - d_i \leq 0, \; i = 1, \dots N.$$

Minimum problem: continue

For each group, introduce a binary variable y_i and a large positive number M.
Then the final block of inequalities is as follows:

$$a_1 t_i + b_1 - d_i \leq M(1 - y_i),\ \ i = 1, \dots, N, \tag{14}$$

$$a_2 t_i + b_2 - d_i \leq M y_i,\ \ i = 1, \dots N, \tag{15}$$

$$y_i \in \{0, 1\}. \tag{16}$$

Finally, the problem is to minimise y subject to (10)-(16).

Main approach (pairs)

The optimisation problem is constructed as follows. Assume that a_1 and b_1 are the slope and the intercept of the affine piece in the first ReLU, and a_2 and b_2 are the slope and the intercept of the affine piece in the second ReLU.
For each discretisation point t_i, $i = 1, \dots, N$ we introduce two new variables

$$c_i = \max\{0, a_1 t_i + b_1\}, \ i = 1, \dots, N, \tag{17}$$

$$d_i = \max\{0, a_2 t_i + b_2\}, \ i = 1, \dots, N, \tag{18}$$

where N is the number of discretisation points. The objective is to minimise the absolute deviation z, subject to the following constraints:

$$f(t_i) - (c_i - d_i) \le z, \ i = 1, \dots, N, \tag{19}$$

$$(c_i - d_i) - f(t_i) \le z, \ i = 1, \dots, N. \tag{20}$$

Main approach: continue

Due to (17)-(18), we have the following equations:

$$0 \leq c_i, \tag{21}$$

$$a_1 t_i + b_1 \leq c_i, \;\; i = 1, \ldots, N, \tag{22}$$

$$0 \leq d_i \tag{23}$$

$$a_2 t_i + b_2 \leq d_i, \;\; i = 1, \ldots N, \tag{24}$$

and for every i, at least one of the inequalities has to be satisfied as equality. Now, we have to introduce a binary variable.

Main approach: continue

1. For each group i, at least one of the following reverse inequalities is satisfied (for c_i):

$$a_1 t_i + b_1 \geq c_i, \ \ i = 1, \dots, N,$$
$$0 \geq c_i, \ \ i = 1, \dots N.$$

   This can be achieved using binary variables z_i, $i = 1, \dots, N$.

2. For each group i, at least one of the following reverse inequalities is satisfied (for d_i):

$$a_2 t_i + b_2 \geq d_i, \ \ i = 1, \dots, N,$$
$$0 \geq d_i, \ \ i = 1, \dots N.$$

   This can be achieved using binary variables $\tilde{z}_i$, $i = 1, \dots, N$.

Main approach: continue

Consider a large positive parameter M (big-M, fixed value). Then the requirement for at least one of the inequalities in each group (case 1 for c_i and case 2 for d_i) holds can be expressed as follows:

$$c_i - (a_1 t_i + b_1) \leq M z_i, \ i = 1, \dots, N, \quad (25)$$

$$c_i \leq M(1 - z_i), \ i = 1, \dots N, \quad (26)$$

$$d_i - (a_2 t_i + b_2) \leq M \tilde{z}_i, \ i = 1, \dots, N, \quad (27)$$

$$d_i \leq M(1 - \tilde{z}_i), \ i = 1, \dots N, \quad (28)$$

$$z_i, \tilde{z}_i \in \{0, 1\}, \ i = 1, \dots, N. \quad (29)$$

Finally, the goal is to minimise z subject to (19)-(29). Therefore, every additional pair leads to a significant increase in the number of the decision variables, including binary variables.

Experiments with univariate functions

All the numerical experiments are performed on the interval $[-1, 1]$, the discretisation step is $h = 10^{-3}$ and the number of pairs is 1. We test our method on five different functions:

1. $f_1(t) = \sqrt{|t|}$; this function is nonsmooth and non-Lipschitz.
2. $f_2(t) = \sqrt{|t - 0.75|}$; this function is similar to $f_1(t)$, but it is non-symmetric.
3. $f_3(t) = \sin(2\pi t)$; this function is periodic and oscillating.
4. $f_4(t) = t^3 - 3t^2 + 2$; this is a cubic function. The experiments with this function are interesting, since this is an example, where neural network was especially inaccurate, despite the fact that this is a smooth function without any abrupt changes.
5. $f_5 = 1/(t^{25} + 0.5)$; this is a very complex function for approximating by a continuous piecewise linear function with only two linear pieces. The structure of the approximation drastically changes when the the discretisation step is changing.

Comments and graphs

Function f_5 is changing abruptly near the point $t = -1$. This property makes it very hard to approximate f_5 by any continuous function piecewise linear function even when the number of linear pieces is large.

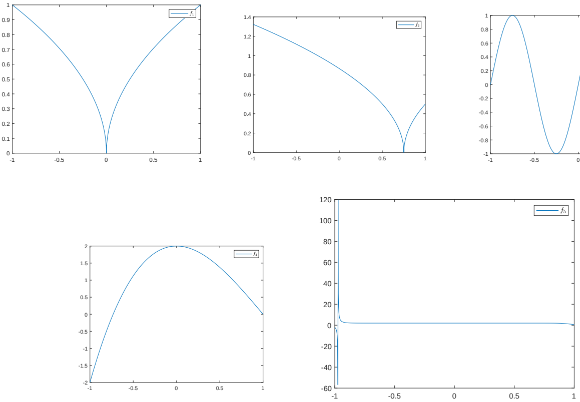


Figure: Univariate functions for approximation

Results

The results of the numerical experiments are presented in Table 1.

Fun	Knot(s)	Max. abs. dev.	Time (sec.)
f_1	N/A	0.4997	5333
f_2	N/A	0.3002	3044
f_3	N/A	0.9936	8
f_4	$\theta_1 = 0.2309$	0.3388	959
f_5	$\theta_1 = -0.8742$, $\theta_2 = 0.9730$	171.8496	364

Table: Computational results: one pair, $h = 10^{-3}$.

The second column of Table 1 contains the location of knots: we only keep the knots from the interval $[-1, 1]$. With only one pair, the maximal possible number of knots is two (function f_5). In the case where all the knots are outside $[-1, 1]$, the column contains "N/A". In the case of functions f_1, f_2 and f_3, no knot was found in the interval $[-1, 1]$, while the computational time is high (for functions f_1 and f_2).

Two pairs

To be able to handle the large scaled problems, we increase the discretisation step-size from $h = 10^{-3}$ to $h = 0.04$. Even with the increased step-size, the computational time increased drastically without any significant improvement of the objective function value. In some cases (function f_4) it became even worse, since the corresponding MILPP is very large. In the case of function f_5, the value of the objective function significantly improved, but this is probably due to the discretisation effect: the discretisation step is much larger and it "smoothed" the function that we need to approximate. Overall conclusion: the direct approach is not efficient for most cases.

Fun	Knot(s)	Max.Abs.Dev.	Time (s)
f_1	$\theta_1 = 0$, $\theta_2 = 0.0766$, $\theta_3 = 0.7871$	0.1417	6579
f_2	$\theta_1 = 0$	0.1264	4321
f_3	$\theta_1 = 0.2223$, $\theta_2 = 0.72$	0.1933	285
f_4	$\theta_1 = 0$	1.7619	3033
f_5	$\theta_1 = -0.9157$, $\theta_2 = 0.0802$, $\theta_3 = 0.8944$	0.0802	1925

f_5

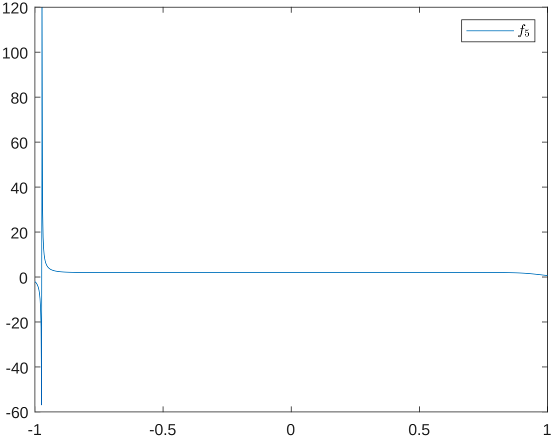
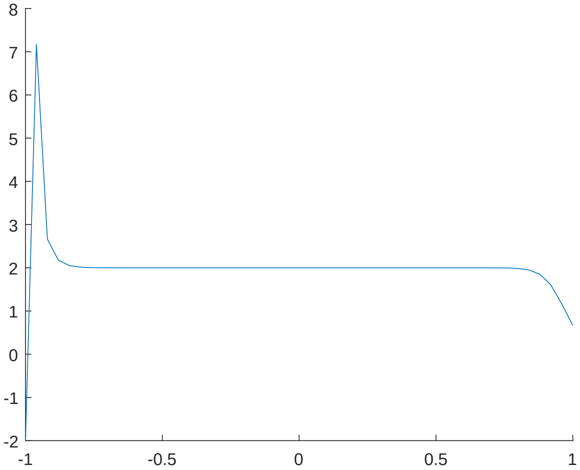


Figure: f_5

Reduced f_5



Fixed knots, convex optimisation

Now we assume that the knots are known and therefore they are not part of the decision variables. Since the optimal location of the knots is not known, we assume that the knots are equidistant.

The maximal number of linear pieces is three and in the case of two pairs it is at most five. Therefore, in our experiments, for each function, we consider the cases when the number of pieces is 3, 4 or 5. The discretisation step is again $h = 0.001$ (main settings).

Table: convex and univariate

Fun	number of linear pieces	Max. abs. dev.	Time (sec.)
f_1	3	0.2885	0.3054
	4	0.0884	0.1607
	5	0.2236	0.1538
f_2	3	0.2772	0.1356
	4	0.25	0.1591
	5	0.2148	0.1324
f_3	3	0.7267	0.2913
	4	0.8311	0.1405
	5	0.3576	0.1261
f_4	3	0.2776	0.1367
	4	0.1642	0.1408
	5	0.1080	0.1367
f_5	3	88.2412	0.1241
	4	88.2090	0.1290
	5	88.1758	0.1249

Comments

Comparing the results, one can see that the computational time is significantly lower in the case of the fixed knots model. It is more efficient to run the convex model several times with different number of (equidistant) knots and choose the best approximation.

Remark

One approach is to assign the knots to the location, obtained through piecewise polynomial approximation (discontinuous case). Note that this location is NOT optimal.

Multivariate case

In the case of multivariate functions, the dimension is increasing and this makes it difficult to solve the corresponding MILPPs. In our study, we consider two functions:

- $\Phi_1(x, y) = \sqrt{|x - 0.5| + 3|y|}$ (function with a deep minimum);
- $\Phi_2(x, y) = \sin(5x - 0.5) - \sqrt{|\cos(7y)|}$ (function with several shallow local minima).

Graphs

The functions are approximated on the hypercube $[-1, 1] \times [-1, 1]$, the discretisation step-size along each direction is $h = 0.05$, this leads to 41 discretisation points along each direction.

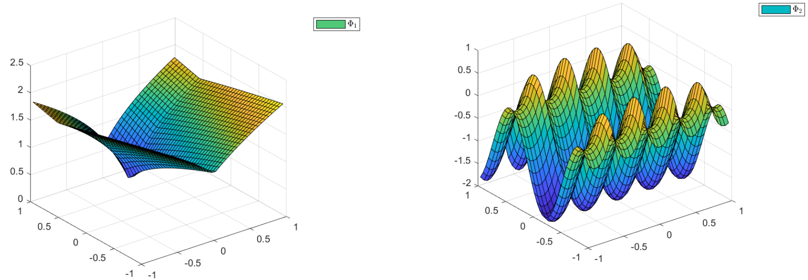


Figure: Multivariate functions for approximation

Multivariate: direct

Fun	Number of pairs	Max. abs. dev.	Time (sec.)
Φ_1	1	0.9493	623
	2	1.9365	7206
Φ_2	1	1.2697	765
	2	1.9974	7203

Table: Computational results: multivariate approximation, $h = 0.05$ along each dimension

Comments

The computational time is very high. Moreover, in the case of two pairs, the program terminated prematurely, because it exceeded the time limit. Moreover, due to this premature stopping, the value of the objective function is better for the case of only one pair (for both functions). The overall conclusion is that this method is not efficient. In the next section we present the convex optimisation-based approach, which can be seen as a generalisation of fixed knot approximation.

Convex approach

In this section, we propose an approach which can be seen as an extension of fixed knot linear splines for multivariate approximation. We use the following knot-extension coverage:

$$\pm\max\{0,x+c_1\},\pm\max\{0,y+c_2\},\pm\max\{0,x+y+c_3\},\pm\max\{0,x-y+c_4\},$$

where c_1, c_2, c_3 and c_4 are grid nodes, defined on the interval $[-1,1]$, the step-size is $h=10^{-3}$ (we can even affort a much finer discretisation grid). Our approach simply splits the full 2D sphere (2π) into sectors of $\pi/4$. For higher dimensions the problem is much harder. Potentially, it can be based on the geometrical study of the so called kissing number, which is still an open problem.

Results: multivariate and convex

Function	Maximal absolute deviation	Time (sec.)
Φ_1	0.0687	220
Φ_2	0	213

Table: Computational results: multivariate approximation, fixed knots, $h = 10^{-3}$.

Conclusions

one can see that the second approach (convex optimisation-based approach) is fast and accurate. In this study, we use a straightforward approach for constructing the grid for multivariate fixed knot coverage. This approach relies on the fact that ReLU is positive homogeneous. The main reason for the efficiency of this method is that it is based on convex optimisation, which is efficient even for large problems. In our experiments, we reformulate the convex problems as equivalent linear programming problems, the dimension is over 16,000. Nonetheless, the computational time is much lower for this approach than it is for the direct approach. One of the most important future research directions is to establish an efficient approach for constructing multivariate fixed knot coverage.

Future research

Similar to univariate approximation, the convex optimisation-based approach can only reach an optimal solution if the optimal location of the knots (coverage) is known. Theoretically, with the same number of linear pieces, one can reach more accurate approximation, but in practice this is a very challenging task, since the problems are non-convex.
It is also important to note that in the case of function Φ_2 the maximal deviation is zero (interpolation). It may be possible that there is more than one way to interpolate the grid points of this function and then one needs to decide which approximation is best. In many practical problems, this can be done by applying a regularisation. For example, to choose a piecewise interpolation with the smallest number of linear pieces involved.
Finally, any breakthrough in approximation by linear splines, will advance KAN (Kolmogorov-Arnold Networks). This field was declared as "very promising", but it can't progress without obtaining solid results on piecewise linear approximation with free knots.