Two approaches to piecewise affine approximation

Nadezda Sukhorukova, e-mail: nsukhorukova@swin.edu.au
Swinburne University of Technology

Abstract

The problem of approximation by piecewise affine functions has been studied for several decades (least squares and uniform approximation). If the location of switches from one affine piece to another (knots for univariate approximation) is known the problem is convex and there are several approaches to solve this problem. If the location of such switches is unknown, the problem is complex even in the case of univariate approximation. In its classical formulation, the number of affine pieces is restricted, since it is proportional to the dimension of the corresponding optimisation problems. On the other hand, the recent development of the theory of neural networks, demonstrate that the functions can be efficiently approximated by overparametrised neural networks (least squares-based loss function). In this paper, we use Chebyshev (uniform) optimisation criteria and compare the classical approximation approach (direct) and a convex optimisation-based approach, where the number of affine pieces is large, but smaller than it is in the case of overparametrised networks. This can be seen as a step towards understanding of the optimisation background behind overparametrised networks with uniform loss function.

Keywords: piecewise affine approximation, neural network approximation, optimality conditions, convex optimisation

MSC: 90C30, 90C11, 90C59, 90C90

1 Introduction

The problem of function approximation by a piecewise affine function was studied for decades. In the case of univariate function approximation, the continuous piecewise affine functions are also called linear splines. This problem was extensively studied [18,24–26]. The coefficients of the affine pieces are the decision variables. When the points of joining the affine pieces (called knots) are also variables (free knot approximation), the problem of best Chebyshev approximation remains open [18]. Moreover, this problem was identified as **one the of the most important problems of modern approximation** [5].

There are several theoretical results on free knot linear spline approximation (most of them have been extended to higher degree polynomial splines as well): necessary optimality conditions [8,18,19,27], sufficient optimality conditions [18, 19] (just to name a few). Since the problem of free knot spline approximation is non-convex, necessary and sufficient optimality conditions do not coincide and it is not known if the conditions can be improved.

In the case of multivariate function approximation, the situation is similar: when the location of switches from one affine piece to another is known, the problem is convex, while in the case when they are subject to optimisation, the problems are non-complex and remain open. The role of knots are assigned to the intersection of affine pieces and therefore they are not points, but affine spaces of lower dimensions. In this paper, for simplicity, we call these switches knots (similar to univariate case).

Remark 1. It is common to call these affine pieces as linear pieces.

In [4], the authors study piecewise affine approximations of bilinear terms. The authors provide the structural comparison of univariate and bivariate problems and reformulate them as mixed-integer programming problems. At first glance, it is easy to underestimate the importance of this study: the authors approximate very specific types of functions. At the same time, this problem is a difficult optimisation problem and has many practical applications (see [4] for details).

In its classical formulation, the objective is to mimimise the maximal deviation of the approximation from the function, given the maximal number of affine pieces. With the increase of the number of possible pieces, the approximations become more accurate, but the dimension of the corresponding optimisation problems is increasing.

On the other hand, artificial neural networks (ANNs) are very powerful and popular approximation techniques, which lead to piecewise affine approximations when the activation functions are ReLU or leaky ReLU. ANNs have many practical applications, including image and sound recognition, partial differential equation, fluid dynamics and many others, see [13] and references therein. ANNs can handle models with a large number of variables by decomposing complex optimisation problems into several simpler problems that can be solved efficiently using modern optimisation. This approach is so efficient that in practice one uses overparametrised ANNs and therefore, in the case of discretised function approximation, it leads to piecewise affine interpolation, that is, the deviation at the discretisation points is zero. One has to remember, however, about the danger of overfitting, which is quite common for overparametrised networks. Modern ANNs use a number of regularisers to reduce this possibility.

The theory of ANNs is based on solid mathematical modelling established in [9,14,16,21], but the origins come to the work of A. Kolmogorov and his student V. Arnold. The celebrated Kolmogorov-Arnold Theorem [2,15] is an attempt to solve the 13th problem of Hilbert. An excellent overview of optimisation techniques for ANNs can be found in [28]. In [30], the authors provide a comprehensive study of the activation functions from the point of view of optimisation. Another interesting publication [1] demonstrates the existence of target functions (functions to be approximated) that are as difficult to approximate using neural networks with pre-selected activation functions. This does not contradict the classical results (for example [9]), just the architecture of these networks may be extremely complex. In this paper, we develop a convex optimisation-based approach, which can be seen as an extension of ANNs, where the activation functions within the same layer may differ. Therefore, there are still many open problems in this field.

In this paper, we are working with uniform (Chebyshev) approximation, where the objective is to minimise the maximal absolute deviation of the approximation from the original function. In our study, we approximate discretised functions, defined on fine grids. The extension of the results to continuous functions (multivariate case) is not straightforward even for classical polynomial approximation, where the corresponding optimisation problems are convex [22] and therefore it is out of scope of this paper. There are two “polar” approaches to piecewise affine approximation. The first (classical) approach requires the minimisation of the maximal absolute deviation when the total number of the affine pieces is fixed. The second (overparametrised) approach allows for a large number of possible affine pieces and therefore leads to interpolation and the maximal absolute deviation is zero. It is also possible to design an approach, which combines the classical and the overparametrised one.

It is clear that in the case of overparametrised approximations the interpolation is not always unique and therefore the choice of possible regulariser is essential. In most practical problems, the objective of the regulariser is the reduction of overfitting.

Most ANNs methods are relying on Gradient Descents (GD) and Stochastic Gradient Descent (SGD) methods. There are several modifications of these methods, for more information, refer to [6] and also the references within [28].

As it has been pointed out, ANNs approximation problems can be formulated as optimisation problems. At the same time, some optimisation models rely on ANNs. For example, in [11] ANNs are used to approximate the weakly efficient frontier of convex vector optimisation problems satisfying Slater's condition. Such a fruitful cross-pollination between these ares leads to the construction of efficient methods.

In this paper, we develop and compare two approaches for piecewise affine approximation. The first one (direct approach) is based on mixed-integer linear programming and remotely related to those proposed in [20], but extended to more than two affine pieces. The second one is a convex optimisation-based approach which requires a significant increase in the number of affine pieces, while the approximations remain underparameterised. It was discovered that the second approach is more efficient than the direct one.

In this paper, all the approximations are in Chebyshev (uniform or infinity) norm. This extends the existing studies [29], where the authors deal with l_p, $p < \infty$.

The paper is organised as follows. In section 2 we provide the background of the problem. Then in sections 3 and 4 we present our main results: section 3 covers univariate approximation, while section 4 refers to multivariate approximation. Section 5 contains conclusions and further research directions.

2 Preliminaries

2.1 Piecewise affine approximation and its connections with Artificial Neural Networks

ANNs are a popular tool of Machine learning and Artificial Intelligence (AI). Deep learning is a special type of ANNs that has many practical applications [13,17,28]. Deep learning and ANNs can be used as a purely approximation tool for univariate and multivariate functions. The objective is to optimise weights in the network. This approximation problem can be solved using modern optimisation tools. In the corresponding optimisation models, the objective function is the deviation of the approximation from the original function (which may be a continuous function or a function whose values are known only at some discretisation points). The deviation function (called "loss function", essentially, it represents the approximation error) can be chosen in a number of ways. For example, it can be the sum of the squares of the deviations (least squares based models). Other possible options include, but not limited to the maximum of the absolute values (uniform or Chebyshev approximation based models), the sum of the absolute values (Manhattan-distance based models), etc.

In deep learning, ANNs have a very specific structure: input layer, one or more hidden layers and the output layer. Theoretically, one hidden layer (with sufficiently large number of nodes in it) is enough to achieve high accuracy [9, 14,16,21]. In practice, however, it is recommended to train neural networks with several hidden layers [13,28]. There is no specific mathematical explanation for this, just experimental results. In this paper, we study models with a single hidden layer. In this case, ANNs build approximations in the form

$$S(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \sigma\left(\mathbf{w}^i \mathbf{x} + w_0^i\right), \tag{2.1}$$

where N is the number nodes in the hidden layer, $\mathbf{x} \in \mathbb{R}^n$, function $\sigma : \mathbb{R} \to \mathbb{R}$ is called the activation function. Activation functions are predefined in advance and not subject to optimisation. The decision variables (weights) are α_i, $i = 1, \ldots, N$ and w_j^i, $i = 1, \ldots, N$, $j = 1, \ldots, n$. In this study, we use $\sigma(t) = \max\{0, t\}$ as the activation function, also known as ReLU. There are two main reasons for choosing this activation function: this choice is quite common in ANNs and also $S(\mathbf{x})$ in (2.1) is a continuous piecewise affine function.

2.2 Univariate functions

The history of uniform approximation starts with P. Chebyshev [7]. The conditions are based on the number of alternating points, the point of the highest absolute deviation, while the deviation signs are alternating.

Theorem 2.1. *(Chebyshev) A necessary and sufficient optimality condition for a polynomial of degree n is the existence of $n + 2$ alternating points.*

In particular, if the degree of the polynomial is one (single affine piece), the necessary and sufficient optimality condition is the existence of three alternating points.

In most practical problems, one needs to discretise the domain rather than working with continuous functions. The problem is as follows:

$$\text{minimise} \max_{t_j, j=1,\ldots,N} \left| a_0 + \sum_{i=1}^{n} a_i \max\{0, t_j - \theta_{i-1}\} - f(t) \right|, \text{ subject to } X \in \mathbb{R}^{2n}, \tag{2.2}$$

where N is the number of discretisation points, $t_j \in [c, d]$ for $j = 1, \ldots, N$, n is the number of subintervals, $\theta_0 = c$, $\theta_n = d$, $X = (a_0, a_1, \ldots, a_n, \theta_1, \ldots, \theta_{n-1})$ are the decision variables.

If the knots are part of the decision variables, the problem is still open. In [20] the authors consider this problem with two affine pieces. The corresponding optimisation problem was reformulated as a mixed-integer linear programming problem (MILPP) and then was efficiently solved. When the number of affine pieces is increasing the problem becomes more complex. One of the objectives of this paper is to extend this approach for more than two affine pieces.

2.3 Multivariate functions

In the case of multivariate approximation, the problems are more complex. In particular, the corresponding basis functions do not form a Chebyshev system and the notion of alternating sequence is not defined, since the points are not totally ordered. At the same time, when the location of knots is known or when there is a single affine piece, the optimisation problems are convex and therefore can be efficiently solved.

The discretised version of the problem is as follows:

$$\text{minimise} \max_{\mathbf{T}_j, j=1,\ldots,N} \left| \sum_{i=1}^{n} a_i \max\{0, \mathbf{w}^i \mathbf{T}_j + w_0^i\} - f(\mathbf{T}_j) \right|, \text{ subject to } X \in \mathbb{R}^{n+nd+n}, \tag{2.3}$$

where N is the number of discretisation points, $bfT_j \in Q$ are d-dimensional vectors, Q is a d-dimensional hypercube, $j = 1, \ldots, N$, n is the maximal number of affine pieces, w_k^i, $i = 1, \ldots, n$, $k = 1, \ldots, d$ and a_i, $i = 1, \ldots, n$ are the decision variables. $X = (a_1, \ldots, a_n, w_0^1, \ldots, w_d^n$ are the decision variables.

2.4 Overparameterised ANNs

It is worth mentioning one result [12], where the authors prove that the sub-level set is connected for single hidden layer overparametrised (ultra-wide) networks with ReLU activation. This does not imply every local minimum is global, but implies there is no spurious valley (and no strict local minimum). In particular, it is possible to have “flat” areas and there is a need to develop numerical methods to escape from these areas. It maybe possible to escape from such areas using a suitable regualiser.

The development of an optimisation approach for overparameterised ANNs is out of scope of this paper. Note that for overparameterised ANNs the optimal loss function value is zero. Moreover, since for overparameterised network optimal solutions correspond to interpolation, any optimal solution for least squares (smooth function) is also optimal for uniform (Chebyshev) approximation.

In sections 3 and 4 we present our main contribution. We consider two main approaches to piecewise affine approximation. In the first case, we reformulate non-convex problems as MILPPs, which can be efficiently solved when the size of the problems is not large. In the second case, we increase the number of possible affine pieces, fix the location of switches from one affine piece to another and then optimise the remaining variables. This can be done using convex optimisation techniques, which are fast and efficient. We are not claiming that our solution is optimal for a given number of affine pieces, since there is no efficient way to find optimal location of knots.

In the case of univariate approximation (section 3) we obtained the results that satisfy sufficient optimality conditions. In the case of multivariate approximation, many problems remain open.

3 Univariate function approximation

In the case of univariate approximation, the problems are simpler and therefore there are efficient methods to construct accurate approximations.

3.1 Direct method

In [20] the authors reformulated univariate linear spline approximation with a single internal knot problems as MILPPs. This approach can be extended to more than two subintervals, but the number of binary variables increases drastically, since one needs to consider all possible partitions for the so called max min representation (see, for example [10,23]).

In our experiments, we use a formulation, which is different from (2.2), but related to ANNs. The main reason for choosing this formulation is its straightforward extension to multivariate settings (see section 4).

Consider the approximation in the form of (2.1). We can reformulate it as follows:

$$S(\tilde{\mathbf{x}}) = \sum_{i=1}^{N} \alpha_i \sigma\left(\tilde{\mathbf{w}}^i(\tilde{\mathbf{x}})\right), \tag{3.1}$$

where $\tilde{\mathbf{w}}^{\mathbf{i}} = ((\mathbf{w}^i)^T, w_0^i)$ and $\tilde{\mathbf{x}}^{\mathbf{i}} = ((\mathbf{x}^i)^T, 1)$. In our experiments, we use ReLU as the activation function and therefore $\sigma(t) = \max\{0, t\}$, which is a positive homogeneous function. Therefore, if we know which α_i-s are positive and which are negative in the formulation (3.1), we can omit optimisation of α_i, $i = 1, \dots, N$ and only optimise the weights. In practice, one can implement it by splitting the sum in (3.1) into a difference of two sums. Strictly speaking, the number of components in each sum is not known. In our experiments, we assume that the number of components in each sum is the same and refer to this number as the number of pairs. This approach is not equivalent to the original problem, where the function is approximated by a linear spline with at most n linear pieces, but this is a way to construct piecewise affine approximation with only a few linear pieces. This approach has several advantages. First of all, it connects the classical linear spline approximation theory with ANNs. Second, it further explores the direction of reformulating the problem as a difference of convex (DC) problem [3] and as an MILPP (similar to [4] or [20], although our methods are novel). Finally, it has a straightforward generalisation to multivariate settings (see section 4 for more details).

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.

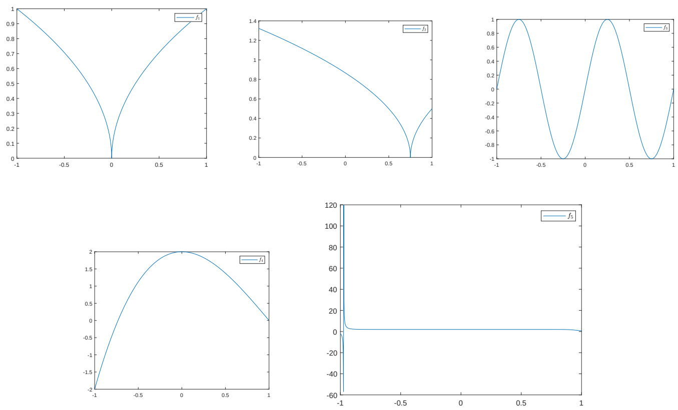

Figure 1: Univariate functions for approximation

The graphs of the functions are presented in Figure 1. 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.

The experimental settings and the functions are the same as in [20], but in the current paper we are not restricting ourselves to just one knot. For completeness, we summarise the approach from [20] in the Appendix A.

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

$$c_i = \max\{0, a_1 t_i + b_1\}, \; i = 1, \dots, N, \tag{3.2}$$

$$d_i = \max\{0, a_2 t_i + b_2\}, \; i = 1, \dots, N, \tag{3.3}$$

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

$$f(t_i) - (c_i - d_i) \leq z, \; i = 1, \dots, N, \tag{3.4}$$

$$(c_i - d_i) - f(t_i) \leq z, \; i = 1, \dots, N. \tag{3.5}$$

Due to (3.2)-(3.3), we have the following equations:

$$0 \leq c_i, \tag{3.6}$$

$$a_1 t_i + b_1 \leq c_i, \; i = 1, \dots, N, \tag{3.7}$$

$$0 \leq d_i \tag{3.8}$$

$$a_2 t_i + b_2 \leq d_i, \; i = 1, \dots N, \tag{3.9}$$

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

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):

$$\begin{aligned} a_2 t_i + b_2 &\geq d_i, \ i = 1, \dots, N, \\ 0 &\geq d_i, \ i = 1, \dots N. \end{aligned}$$

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

Consider a larger 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:

$$c_i - (a_1 t_i + b_1) \leq M z_i, \ i = 1, \dots, N, \tag{3.10}$$

$$c_i \leq M(1 - z_i), \ i = 1, \dots N, \tag{3.11}$$

$$d_i - (a_2 t_i + b_2) \leq M \tilde{z}_i, \ i = 1, \dots, N, \tag{3.12}$$

$$d_i \leq M(1 - \tilde{z}_i), \ i = 1, \dots N, \tag{3.13}$$

$$z_i, \tilde{z}_i \in \{0, 1\}, \ i = 1, \dots, N. \tag{3.14}$$

Finally, the goal is to minimise z subject to (3.4)-(3.14). Therefore, every additional pair leads to a significant increase in the number of the decision variables, including binary variables.

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

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

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 f_4 and f_5 the approximation results are similar to those found in [20]. 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).

Our next step is to improve the results by using two pairs. To be able to handle the large scaled problems, we increase the discretisation step-size to $h = 0.04$. The computational time increased drastically without any significant improvement of the objective function value. Moreover, 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. The results are in Table 2.

Fun	Knot(s)	Max. abs. dev.	Time (sec.)
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

Table 2: Computational results: two pairs, $h = 0.04$

Overall conclusion: the direct approach is not efficient for most cases. The only exception is function f_3. In the next section, we suggest another approach, which is not optimal if the optimal location of the knots is not known. At the same time, this approach relies on convex optimisation.

Remark 2. The results in [20] are not directly comparable with the results in the current paper, since the structure of approximations is very different.

3.2 Convex approximation method

In this section, we assume that the knots are known. We use the classical formulation (2.2), but the location of the knots is fixed 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.

In the case of one pair (section 3.1) 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 $h = 0.001$. The results are in Table 3. Comparing

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

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

the results in Table 1 and Table 3, one can see that the computational time is significantly lower in the case of the fixed knots model. It is more efficient to

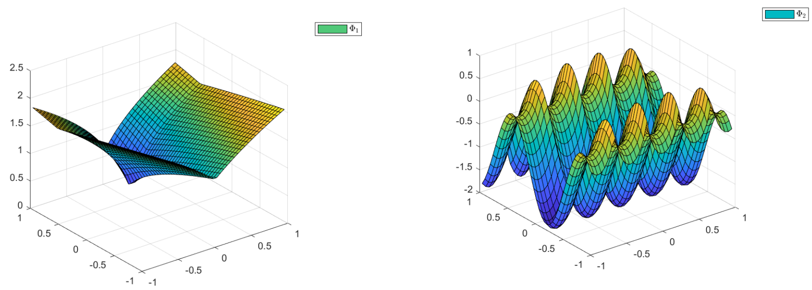

Figure 2: Multivariate functions for approximation

run the convex model several times with different number of (equidistant) knots and choose the best approximation.

Remark 3. 1. The increase in the number of knots does not always lead to the improvement in the objective function value. This is due to the fact that the knots are fixed.

2. In the case when the location of the segments of the abrupt changes of the function are known, it may be beneficial assign more knots in this segments. We use equidistant knots, since we assume that there is no prior information where to locate the knots.

3. The results in Table 2 and Table 3 are not comparable, since the discretisation step-size is very different, in particular, function f_5 is significantly smoothed when the step-size is increasing.

In the next section, we implement similar models for multivariate functions.

4 Multivariate function approximation

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).

The graphs of the functions are presented in Figure 2. 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.

4.1 Direct method

In our experiments, we use formulation (2.2), where T_j, $j = 1,\dots,N$ are two-dimensional vectors. The results are in Table 4. The computational time is very

Fun	Number of pairs	Max. abs. dev.	Time (sec.)
Φ_1	1	0.9493	623
	2	1.9365	7206
Φ_3	1	1.2697	765
	2	1.9974	7203

Table 4: Computational results: multivariate approximation, $h = 0.05$ along each dimension

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.

4.2 Convex approximation method

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:

$$\pm\max\{0, x + c_1\}, \pm\max\{0, y + c_2\}, \pm\max\{0, x + y + c_3\}, \pm\max\{0, x - y + c_4\}, \tag{4.1}$$

where c_1, c_2, c_3 and c_4 are grid nodes, defined on the interval $[-1, 1]$, the step-size is $h = 10^{-3}$. There are many ways how the knot-coverage can be defined. This study is out of scope of this paper. The experimental results are presented in Table 5. Comparing the results in Tables 4-5, one can see that the second

Function	Maximal absolute deviation	Time (sec.)
Φ_1	0.0687	220
Φ_3	0	213

Table 5: Computational results: multivariate approximation, fixed knots, $h = 10^{-3}$.

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 (4.1). 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.

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 [13]. For example, to choose a piecewise interpolation with the smallest number of linear pieces involved.

5 Conclusions and further research directions

In this paper, we study two approaches for univariate and multivariate function approximation by continuous piecewise affine function. The first approach (direct) is relying on solving low dimensional non-convex problems, while the second one is a larger-scaled convex optimisation-based approach. For both univariate and multivariate approximation, the results are better for the convex optimisation-based approach.

For our future research directions, we highlight the following:

1. the development of an efficient method for finding the location of knots for univariate approximation;

2. the development of an efficient approach constructing multivariate knot coverage;

3. the development of efficient and computationally inexpensive regularisation techniques for interpolation problems that can also solve a number of practical needs, including the reduction in potential overfitting (data approximation).

Acknowledgments

We are grateful to the Australian Research Council for supporting this work via Discovery Project DP180100602.

Data sharing is not applicable to this article as no data sets were generated or analysed during the current study.

6 Funding

The project is supported by ARC (Australian Research Council), Discovery Project DP180100602.

7 Data sharing

Data sharing is not applicable to this article as no data sets were generated or analysed during the current study.

8 Competing Interest Declaration

No Competing Interests.

9 Author Contribution

The authors contributed equally to this work.

10 Conflict of Interest Declaration

The authors declared that they have no conflict of interest.

A Mixed Integer Linear Programming Approach for one free knot

In the case of univariate approximation with one free knot, the solution can be found by solving two mixed-integer linear programming problems [20]. In the first problem, the optimal linear spline is the maximum of two affine pieces, while in the second problem the optimal linear spline is the minimum of two affine pieces.

Assume that a_1 and b_1 are the slope and the intercept of the affine piece in the first subinterval, respectively and a_2 and b_2 are the slope and the intercept of the affine piece in the second subinterval, respectively.

A.1 Maximum Problem

First, for each discretisation point t_i, $i = 1, \dots, N$ we introduce a new variable

$$c_i = \max\{a_1 t_i + b_1, a_2 t_i + b_2\},\ i = 1, \dots, N, \tag{A.1}$$

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

$$f(t_i) - c_i \leq z,\ i = 1, \dots, N, \tag{A.2}$$
$$c_i - f(t_i) \leq z,\ i = 1, \dots, N. \tag{A.3}$$

Due to (A.1), we have the following equations:

$$a_1 t_i + b_1 \leq c_i,\ i = 1, \dots, N, \tag{A.4}$$
$$a_2 t_i + b_2 \leq c_i,\ i = 1, \dots N, \tag{A.5}$$

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

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

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:

$$c_i - (a_1 t_i + b_1) \leq M z_i,\ i = 1, \dots, N, \tag{A.6}$$
$$c_i - (a_2 t_i + b_2) \leq M(1 - z_i),\ i = 1, \dots N, \tag{A.7}$$
$$z_i \in \{0, 1\},\ i = 1, \dots, N. \tag{A.8}$$

Finally, the goal is to minimise z subject to (A.2)-(A.8).

A.2 Minimum Problem

Similar to the maximum problem, introduce a new variable

$$d_i = \min\{a_1 t_i + b_1, a_2 t_i + b_2\},\ i = 1, \dots, N, \tag{A.9}$$

where N is the number of discretisation points. The objective is to minimise the absolute deviation y, subject to

$$f(t_i) - d_i \leq y,\ i = 1, \dots, N, \tag{A.10}$$
$$d_i - f(t_i) \leq y,\ i = 1, \dots, N. \tag{A.11}$$

Due to (A.9), we have the following equations:

$$a_1 t_i + b_1 \geq d_i,\ i = 1, \dots, N, \tag{A.12}$$
$$a_2 t_i + b_2 \geq d_i,\ i = 1, \dots N. \tag{A.13}$$

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:

$$a_1 t_i + b_1 - d_i \leq 0,\ i = 1, \dots, N,$$
$$a_2 t_i + b_2 - d_i \leq 0,\ i = 1, \dots N.$$

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

$$a_1 t_i + b_1 - d_i \leq M(1 - y_i),\ i = 1, \dots, N, \tag{A.14}$$
$$a_2 t_i + b_2 - d_i \leq M y_i,\ i = 1, \dots N, \tag{A.15}$$
$$y_i \in \{0, 1\}. \tag{A.16}$$

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

References

[1] J.M. Almira, P.E. Lopez de Teruel, D.J. Romero-López, and F. Voigtlaender. Negative results for approximation using single layer and multilayer feedforward neural networks. *Journal of Mathematical Analysis and Applications*, 494(1):124584, 2021.

[2] V. Arnold. On functions of three variables. *Dokl. Akad. Nauk SSSR*, 114:679–681, 1957. English translation: Amer. Math. Soc. Transl., 28 (1963), pp. 51–54.

[3] Pranjal Awasthi, Anqi Mao, Mehryar Mohri, and Yutao Zhong. Dc-programming for neural network optimizations. *Journal of Global Optimization*, 2024.

[4] Andreas Barmann, Robert Burlacu, Lukas Hager, and Thomas Kleinert. On piecewise linear approximations of bilinear terms: structural comparison of univariate and bivariate mixed-integer programming formulations. *Journal of Global Optimization*, 85:789–819, 2023.

[5] P. Borwein, I. Daubechies, V. Totik, and G. Nürnberger. Bivariate segment approximation and free knot splines: Research problems 96-4. *Constructive Approximation*, 12(4):555–558, 1996.

[6] Léon Bottou. *On-line Learning and Stochastic Approximations*, pages 9–42. Publications of the Newton Institute. Cambridge University Press, 1999.

[7] P. Chebyshev. Théorie des mécanismes connus sous le nom de parallélogrammes. *Mémoires des Savants étrangers présentés à l'Académie de Saint-Pétersbourg*, 7:539–586, 1854.

[8] J.P. Crouzeix, N. Sukhorukova, and J. Ugon. Finite alternation theorems and a constructive approach to piecewise polynomial approximation in chebyshev norm. *Set-Valued and Variational Analysis*, pages 1–25, 2020.

[9] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[10] M. V. Dolgopolik. Nonlocal error bounds for piecewise affine functions. *Set-Valued and Variational Analysis*, 31:32, 2023.

[11] Zachary Feinstein and Birgit Rudloff. Deep learning the efficient frontier of convex vector optimization problems. *Journal of Global Optimization*, 2024.

[12] C. Daniel Freeman and Joan Bruna. Topology and geometry of half-rectified network optimization. *ArXiv*, abs/1611.01540, 2016.

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[14] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[15] A. N. Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Dokl. Akad. Nauk SSSR*, 114:953–956, 1957.

[16] M. Leshno, Ya. L. Vladimir, A Pinkus, and S Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.

[17] Ariel Neufeld and Tuan Anh Nguyen. Rectified deep neural networks overcome the curse of dimensionality when approximating solutions of mckean–vlasov stochastic differential equations. *Journal of Mathematical Analysis and Applications*, 541(1):128661, 2025.

[18] G. Nürnberger. *Approximation by Spline functions.* Springer-Verlag, Berlin, 1989.

[19] G. Nürnberger, L. Schumaker, M. Sommer, and H. Strauss. Approximation by generalized splines. *Journal of Mathematical Analysis and Applications*, 108:466–494, 1985.

[20] Vinesha Peiris, Duy Khoa Pham, and Nadezda Sukhorukova. Best free knot linear spline approximation and its application to neural networks. *ArXiv*, 2404.00008, 2024.

[21] A. Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8:143–195, 1999.

[22] Vera Roshchina, Nadia Sukhorukova, and Julien Ugon. Uniqueness of solutions in multivariate chebyshev approximation problems. *Optim. Lett.*, 18:33–55, 2019.

[23] Stefan Scholtes. *Piecewise Affine Functions*, pages 13–63. Springer New York, New York, NY, 2012.

[24] L. Schumaker. Uniform approximation by chebyshev spline functions. II: free knots. *SIAM Journal of Numerical Analysis*, 5:647–656, 1968.

[25] N. Sukhorukova. Uniform approximation by the highest defect continuous polynomial splines: necessary and sufficient optimality conditions and their generalisations. *Journal of Optimization Theory and Applications*, 147(2):378–394, 2010.

[26] N. Sukhorukova and J. Ugon. Chebyshev approximation by linear combinations of fixed knot polynomial splines with weighting functions. *Journal of Optimization Theory and Applications*, 171(2):536–549, 2016.

[27] N. Sukhorukova and J. Ugon. Characterisation theorem for best polynomial spline approximation with free knots. *Trans. Amer. Math. Soc.*, pages 6389–6405, 2017.

[28] Ruo-Yu Sun. Optimization for deep learning: An overview. *Journal of the Operations Research Society of China*, 8:249–294, 2020.

[29] John Alasdair Warwicker and Steffen Rebennack. A comparison of two mixed-integer linear programs for piecewise linear function fitting. *INFORMS Journal on Computing*, 34(2):1042–1047, 2022.

[30] Matthew E. Wilhelm, Chenyu Wang, and Matthew D. Stuber. Convex and concave envelopes of artificial neural network activation functions for deterministic global optimization. *Journal of Global Optimization*, 85:569–594, 2023.