Journal of Global Optimization
https://doi.org/10.1007/s10898-026-01588-8

Optimal Piecewise Linear Approximations for Sigmoid, Tanh, Probability Density and Cumulative Distribution Functions: Tabular Form and R Package *pwlapprox2d*

John Alasdair Warwicker[1] · **Steffen Rebennack**[1]

Received: 5 December 2024 / Accepted: 16 January 2026

Abstract
We present information in tabular form about optimal piecewise linear (PWL) approximations of the sigmoid and tanh activation functions as used in neural networks, as well as the probability density and cumulative distribution functions of the Normal and log-Normal distributions. The presented approximations minimise the maximum absolute difference between the PWL function and the continuous function being modelled; we also provide information on optimal over- and underestimators. We provide information on the optimal breakpoint locations for different numbers of breakpoints in a series of tables, as well as the accuracy provided by these approximations. This allows practitioners to utilise the PWL approximations whenever needed; for example, to enable the use of mixed-integer linear programming techniques to solve problems where these functions may appear, or to approximate integrals. The provided optimal breakpoint locations lead to an improvement over a uniform breakpoint location of the maximum absolute difference of up to 95% and an average of 84% among the six functions. This information is available online at https://doi.org/10.5281/zenodo.16362215 and can be recreated using the R package *pwlapprox2d* [1].

Keywords Function approximation · Piecewise linear function · Neural networks · Probability density function · Cumulative distribution function · Statistics · Mixed-integer optimisation

1 Introduction

As opposed to approximating continuous functions with polynomial regression functions, piecewise linear (PWL) approximations can be advantageous due to the removal of the complicating non-linearity aspect. Further, PWL approximations can retain important properties

✉ John Alasdair Warwicker
john.warwicker@kit.edu

Steffen Rebennack
steffen.rebennack@kit.edu

[1] Institute of Operations Research, Karlsruhe Institute of Technology, Karlsruhe 76185, Baden-Württemberg, Germany

Published online: 30 January 2026

of the original function, such as shape-based features and convexity. PWL approximations of continuous functions also allow the use of mixed-integer linear programming (MILP) techniques to solve complex mixed-integer non-linear programming (MINLP) problems [2]. Further applications of PWL functions can be found in data envelopment analysis [3] and energy systems [4, 5].

Optimal PWL functions have recently been found using MILP techniques [6–8], including robust decomposition approaches to overcome effects from outliers [9]. However, approaches from Computer Science have used the inherent nature of the PWL regression problem to find optimal PWL functions efficiently [10, 11]. Warwicker and Rebennack [12] recently presented an algorithm to efficiently fit PWL regression functions to continuous functions, utilising both approaches. Their approach was shown to be over 100,000 times faster than the state-of-the-art approaches [6, 13].

In this paper, we use the approach presented by Warwicker and Rebennack [12] to present optimal PWL approximations (alongside over- and underestimators) to common non-linear functions at different approximation levels. For the given number of breakpoints, we present the best accuracy possible (up to three significant figures). Note that we present the results as lower bounds, since there may exist PWL functions with the same number of breakpoints but with worse accuracy. For some specific functions, optimal breakpoint locations are known analytically. In particular, the quadratic function x^2 yields optimal PWL overestimators with equidistant breakpoints [14], while optimal approximations to the logarithmic function $\log(x)$ can be analytically calculated.

In many applications, uniformly distributed breakpoint locations are chosen to approximate non-linear functions. Such a uniform distribution is very natural and easy to implement. However, this comes at the cost of a (potentially) larger maximum absolute difference between the PWL function and the non-linear function for any given number of breakpoints. This difference can indeed be large - among the six functions tested in this paper, the lowest improvement gain is 64.70% while the largest is 95.33%. These are very significant improvements in the approximation quality without the need to increase the number of breakpoints or segments. Unfortunately, it is not straightforward to compute such optimal breakpoint locations. One would need to use some methods from the literature, run some foreign code or even re-implement the algorithms; this is a large obstacle. Therefore, we provide easy access to these locations in this paper in tabular form. Thus, the improvement gain comes now at no extra "cost".

In some cases, it is important to have valid under- or overestimating PWL functions for a given non-linear function. This is even cumbersome for non-linear functions, especially if they are not given in a standard functional form, like the CDF of the standard Normal or standard log-Normal distributions. In this paper, we also provide valid under- and overestimators.

For the purpose of embedding PWL approximations within mixed-integer programming frameworks, we present (in Section 4) the breakpoint locations for the presented PWL approximations; this allows different formulations to be used, see e.g., [14] or [15]. Equations for the segments of the resulting PWL can be easily calculated from the breakpoint locations. Further, we present the accuracy found using a uniform breakpoint selection approach for means of comparison.

The rest of the paper is structured as follows. In the remainder of the introduction, we introduce PWL functions and discuss the approach for approximating them. In Sections 2-3, we introduce the neural network activation functions and probabilistic functions, presenting full tables regarding optimal breakpoint placements in Section 4. We finish with conclusions.

1.1 Piecewise Linear Functions

Throughout this paper we use the following notation: $[n]$ to denote the set $\{1, \ldots, n\}$.

Definition 1 *(*[6]*)* A continuous univariate function

$$p(x) : [\underline{X}, \overline{X}] \to \mathbb{R}$$

with compact interval $[\underline{X}, \overline{X}]$ is called a continuous **piecewise linear (PWL) function**, if there exists a finite number B with

$$\underline{X} = r_1 < \cdots < r_b < r_{b+1} < \cdots < r_B = \overline{X},$$

such that $p(x)$ is an affine function on $[r_b, r_{b+1}]$ for all $b \in [B-1]$. The values $r_1, \ldots, r_B$ are called **breakpoints**, with B the number of breakpoints. For each $b \in [B-1]$, the function $p(x) : [r_b, r_{b+1}] \to \mathbb{R}$ is called a **linear segment**.

For the problem of PWL function fitting of bivariate data, a set of I ordered tuples $(X_i, Y_i) \in \mathbb{R}^2$, $i \in [I]$, is given where

$$-\infty < \underline{X} = X_1 < \cdots < X_i < X_{i+1} < \cdots < X_I = \overline{X} < \infty.$$

The data tuples (X_i, Y_i) are the sorted values of a discrete function mapping $[\underline{X}, \overline{X}]$ to $\mathbb{R}$. We seek to model these data points by a PWL function $p(x) : [\underline{X}, \overline{X}] \to \mathbb{R}$ with a given number of breakpoints $B < I$. Moreover, we seek an optimal PWL function which minimises some distance metric $d(\cdot, \cdot)$ between the data points and the given PWL function.

When fitting a PWL function to a continuous function f, we typically seek a PWL function p within a given error tolerance $\xi > 0$ such that the number of breakpoints of p is minimised. Typically, a discretised set of points from the function f are taken, and approaches for fitting PWL functions to data points are used. For the construction of an optimal PWL function, the breakpoints must be free within the given range (i.e., not constrained to lie on the function itself) and the PWL function should attain the maximum deviation (i.e., the desired error tolerance ξ) to the modelled data/function at least once per segment (otherwise it can be improved) [13]. Further, it is known in both cases that a non-uniform selection of breakpoints often leads to a better PWL approximation (compared to breakpoints that are uniformly spaced along the original function, or those that are placed randomly) [16].

We further consider optimal over- and underestimators of a function f.

Definition 2 For a given constant $\delta > 0$, a function $p(x) := [\underline{X}, \overline{X}] \to \mathbb{R}$ with compact interval $[\underline{X}, \overline{X}]$ is called a δ-underestimator of $f(x) := [\underline{X}, \overline{X}] \to \mathbb{R}$ if $p(x) \leq f(x)$ for all $x \in [\underline{X}, \overline{X}]$ and

$$\max_{x \in [\underline{X}, \overline{X}]} |p(x) - f(x)| \leq \delta$$

is satisfied. p is a δ-overestimator of f iff $-p$ is a δ-underestimator of $-f$.

1.2 Methodology for Finding Optimal PWL Approximations

In this section, we describe the methodology presented by Warwicker and Rebennack [12], which fits an optimal PWL function (say, f') to a continuous function f with maximum error $\varepsilon > 0$. Their approach embeds the linear time algorithm for PWL fitting within an iterative discretisation framework, first considered in [6].

After an initial (uniform) discretisation of the continuous function f into discrete data points (denoted as the set $\mathcal{I}$), the data points $\mathcal{I}$ are modelled with an approximating PWL function f' that has a maximum error (i.e., ℓ_0 loss) of $\xi = \varepsilon - \alpha$ (for some $0 < \alpha < \varepsilon$) using the algorithm from [10, 11]; we use $\alpha = \varepsilon/5$ as suggested. Then, the maximum error between the PWL function f' and the continuous function f is calculated to within $\alpha/2$-optimality by a global optimisation solver. If this maximum error exceeds ξ by more than $\alpha/2$, then the discretisation of the continuous function (i.e., $\mathcal{I}$) is refined (that is, more data points sampled from the continuous function f are added to the discretisation $\mathcal{I}$); otherwise, an optimal PWL function f' with maximum error ε has been found. This process repeats until the optimality conditions are met. We refer the reader to [12] for the full details of this method.

We implemented the presented approach within C++, and used the locally biased variant of the DIRECT algorithm [17] within NLopt [18] as the global optimisation solver.

This method is guaranteed to find a minimal breakpoint PWL approximation f' for the given continuous function f [12]. Therefore, the results presented in this paper are verifiable.

In this work, we provide optimal PWL approximations at different error bounds ε for six commonly used functions. We note that for the given error bounds, the presented breakpoint locations are not necessarily unique in that there may exist others that lead to an approximating PWL function with the same error bound. Proposition 1 provides the correctness of function over- and underestimators formed by translating the optimal PWL functions; we refer to [19] for a proof.

Proposition 1 ([19]) *Let $p(x)$ denote an ε-optimal PWL approximation of a continuous function $f(x)$ on a compact interval $[\underline{X}, \overline{X}]$. Then, for $\delta = 2\varepsilon$, δ-optimal over-and underestimators for $f(x)$ are given by $p(x) + \varepsilon$ and $p(x) - \varepsilon$ respectively.*

We further extend the generality of the results we present in Proposition 2.

Proposition 2 ([19]) *Let p be an ε-optimal PWL approximation of a function f on a compact interval $[a, b]$ with $a < b$, and breakpoints $a = r_1, \ldots, r_B = b$. Let $\hat{a} < a$, and define $\hat{p}$ where the first breakpoint of p is changed such that it now has $r_1 = (\hat{a}, f(\hat{a}))$. Then, $\hat{p}$ is a PWL approximation of f with maximum error ε over $[\hat{a}, b]$ if*

$$\max_{x \in [\hat{a}, r_2]} |\hat{p}(x) - f(x)| \leq \varepsilon.$$

Proposition 2 allows the general extension of the presented bounds and holds symmetrically (i.e., also for some $\hat{b} > b$). In particular, we present breakpoint information on the updated first (respectively last) breakpoints such that the presented bounds still hold for a general domain. Since many of the functions we present approximations for have asymptotic bounds, we are able to use Proposition 2 for general bounds of the form $\hat{a} \leq a$. Where this occurs, we explicitly point this out. Otherwise, we note that generally extending the bounds would increase the number of breakpoints of the required PWL function by 1 (on each side); the new breakpoint would occur with x-value as the new chosen bound and y-value either coincident with the continuous function or at the asymptotic bound (depending on the function).

2 Neural Network Activation Functions

This section is motivated by the increase in mathematical programming approaches used in tandem with neural network architectures (see e.g., [20–24]). For a given solution to a neural

network (i.e., a collection of trained weights and biases), this solution can be transformed into a MILP model if the activation functions are (piecewise) linear (e.g., ReLU functions) and the output functions are linear (e.g., max-pooling, convolutions). This allows interesting features of the architecture to be discovered, such as feature relevance and the generation of adversarial inputs. Recent work has built the approach within two-stage stochastic programming models, leading to a singular monolithic MILP model [25, 26].

We therefore present PWL approximations of the two most common non-linear activation functions (i.e., sigmoid and tanh) to allow the use of these within the aforementioned MILP models (using PWL formulations as in e.g., [14, 27]). Sildir and Aydin [28] showed how PWL approximations of these activation functions can lead to efficient formulations. PWL approximations of the sigmoid function can also be useful in classification tasks, as it forms the basis of the standard model for logistic regression [29].

2.1 Sigmoid Function

The sigmoid function is defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$

We present approximations over the range of inputs $x \in [-7, 7]$ as this suffices for almost all outputs in the range of $[-1, 1]$ (up to an accuracy of less than one thousandth). For negative values outside this range, the function can be overapproximated by $\sigma(-7) \approx 0.000912$ or underapproximated by 0; each giving a maximum error of $\sigma(-7)$. Likewise, for positive values outside this range, the function can be overapproximated by 1 or underapproximated by $\sigma(7)$; again, the maximum error would be the same as the negative case. Otherwise, taking the approximated PWL value at the extreme values in the range suffices for the presented error bounds. In general, we seek approximations with an even number of breakpoints to take advantage of the symmetry of the function. We present summarised results in Table 1a, with an example in Figure 1a.

2.2 Tanh Function

The hyperbolic tangent function $\tanh(x)$ is defined as:

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}.$$

We again present approximations over the range of inputs $x \in [-4, 4]$ as this suffices for almost all outputs in the range of $[-1, 1]$ (up to an accuracy of less than one thousandth). For negative values outside this range, the function can be overapproximated by $\tanh(-4) \approx 0.000671 - 1$ and underapproximated by -1. Likewise, for positive values outside this range, the function can be overapproximated by 1 or underapproximated by $\tanh(4) = -\tanh(-4)$. In each case, the maximum error would be $1 - \tanh(4)$. Otherwise, taking the approximated PWL value at the extreme values in the range suffices for the presented error bounds. We once more seek approximations with an even number of breakpoints to take advantage of the symmetry of the function. We present summarised results in Table 1b, with an example in Figure 1b.

Table 1 Accuracies of optimal breakpoint locations vs. uniform breakpoint locations

#Breakpoints	Accuracy		Relative improvement
	Optimal	Uniform	
(a) Sigmoid function			
4	≥ 0.0274	0.0568	51.76%
6	≥ 0.0108	0.0502	78.52%
8	≥ 0.00575	0.0392	85.33%
10	≥ 0.00357	0.0273	86.92%
12	≥ 0.00244	0.0188	87.02%
14	≥ 0.00177	0.0131	86.49%
16	≥ 0.00135	0.00938	85.61%
18	≥ 0.00106	0.00771	86.25%
20	≥ 0.000850	0.00640	86.72%
(b) Tanh function			
4	≥ 0.0584	0.148	60.54%
6	≥ 0.0228	0.102	77.65%
8	≥ 0.0121	0.0896	86.49%
10	≥ 0.00750	0.0673	88.86%
12	≥ 0.00510	0.0486	89.51%
14	≥ 0.00370	0.0351	89.46%
16	≥ 0.00282	0.0257	89.03%
18	≥ 0.00222	0.0191	88.38%
20	≥ 0.00178	0.0159	88.81%

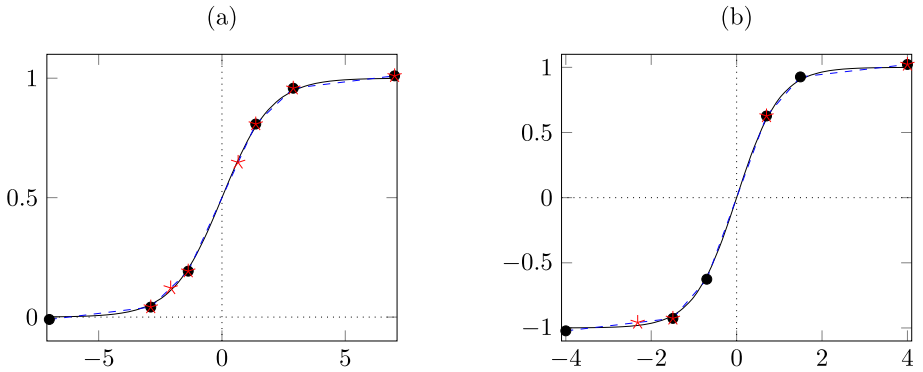

Fig. 1 Optimal PWL approximations for the neural network activation functions: (a) sigmoid with 6 breakpoints; (b) $\tanh(x)$ with 6 breakpoints. Dashed lines are used to indicate PWL functions with breakpoints at black dots. Red stars indicate where the PWL function achieves its maximum deviation (at least once per linear segment)

3 Probabilistic Functions

This section is motivated by the use of approximating probabilistic functions within the context of mathematical programming models. For example, Kaya et al. [30] used the approach presented by Warwicker and Rebennack [12] to approximate the probability distribution of a

price bid for different market stages in the context of improving the reliability of renewable energy producers.

3.1 Probability Density Functions

The probability density function (PDF) gives the relative likelihood that a real-valued variable X takes a given value in the sample space; it is mostly used to specify the probability that the value falls within a certain range.

3.1.1 Standard Normal Distribution

The PDF of the standard Normal distribution which has mean $\mu = 0$ and standard deviation $\sigma = 1$ (not to be confused with the sigmoid function discussed in the previous section) is given as

$$f(X) = \frac{1}{\sqrt{2\pi}} e^{-X^2/2}.$$

We present approximations over the range $X \in [-3.5, 3.5]$ to approximate all values up to an error of one thousandth. Over- and underestimators outside this range can be found by $f(-3.5) = f(3.5) \approx 0.000873$ and 0 respectively.

We present summarised results in Table 2a, with an example in Figure 2a.

3.1.2 Standard Log-Normal Distribution

The standard log-Normal distribution satisfies

$$f(X) = \frac{1}{X\sqrt{2\pi}} e^{-(\ln X)^2/2}.$$

We present approximations over the range $X \in (0, 13.5]$ to approximate all values up to an error of one thousandth. At $X = 0$, the log-Normal distribution is undefined but we note $\lim_{x\to 0^+} f(X) = 0$. The extreme values can be overestimated and underestimated by $f(13.5) \approx 0.000999$ and 0 respectively. In the results in the appendix, we consider the lower bound as 10^{-6} since the PWL approximation method requires the function to be evaluated at the endpoints. We present summarised results in Table 2b, with an example in Figure 2b.

3.2 Cumulative Distribution Functions

Alongside the PDFs, we present PWL approximations of the cumulative distribution functions (CDFs) of the standard Normal and log-Normal distribution. The CDF of a real-valued variable X, evaluated as x, provides the probability that X will take a value less than or equal to x. Formally, we denote a CDF of X evaluated at x by $F_X(x) : \mathbb{R} \to [0, 1]$, where

$$F_X(x) = \mathbb{P}(X \leq x).$$

For every distribution X over $\mathbb{R}$, F_X is unique. Further,

$$\lim_{x\to-\infty} F_X(x) = 0 \quad \text{and} \quad \lim_{x\to+\infty} F_X(x) = 1$$

hold.

Table 2 Accuracies of optimal breakpoint locations vs. uniform breakpoint locations of PDFs

#Breakpoints	Accuracy		Relative improvement
	Optimal	Uniform	
(a) Standard Normal distribution			
5	≥ 0.0216	0.0425	49.18%
6	≥ 0.0108	0.0867	87.55%
8	≥ 0.00807	0.0469	82.80%
9	≥ 0.00432	0.0273	84.18%
11	≥ 0.00420	0.0197	78.68%
12	≥ 0.00258	0.0197	86.91%
13	≥ 0.00237	0.0146	83.77%
14	≥ 0.00233	0.0142	83.59%
15	≥ 0.00179	0.0111	83.88%
(b) Standard log-Normal distribution			
4	≥ 0.0471	0.655	92.81%
5	≥ 0.0442	0.651	93.21%
6	≥ 0.0311	0.646	95.18%
7	≥ 0.0180	0.637	97.17%
8	≥ 0.0179	0.627	97.15%
9	≥ 0.0097	0.614	98.42%
10	≥ 0.00970	0.600	98.38%
11	≥ 0.00675	0.585	98.85%
12	≥ 0.00608	0.568	98.93%
13	≥ 0.00535	0.551	99.03%
14	≥ 0.00464	0.536	99.13%

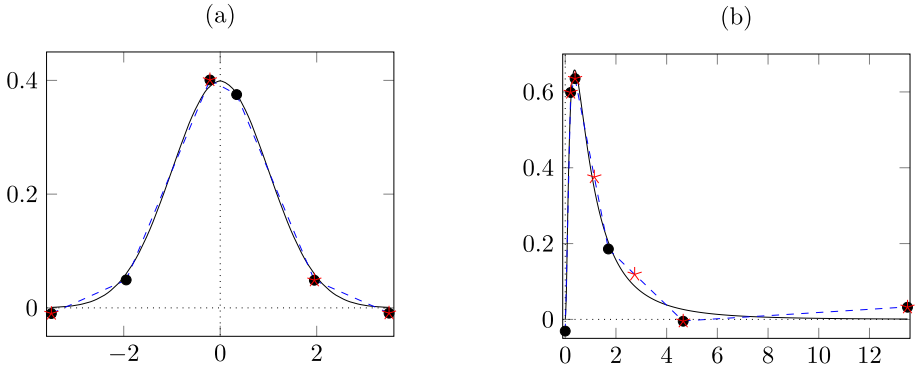

Fig. 2 Optimal PWL approximations for PDFs: (a) Normal distribution with 6 breakpoints; (b) log-Normal distribution with 6 breakpoints. Dashed lines are used to indicate PWL functions with breakpoints at black dots. Red stars indicate where the PWL function achieves its maximum deviation (at least once per linear segment)

Table 3 Accuracies of optimal breakpoint locations vs. uniform breakpoint locations of CDFs

#Breakpoints	Accuracy		Relative improvement
	Optimal	Uniform	
(a) Standard Normal distribution			
4	≥ 0.0244	0.0557	56.17%
6	≥ 0.00954	0.0494	80.69%
8	≥ 0.00508	0.0290	82.48%
10	≥ 0.00315	0.0168	81.25%
12	≥ 0.00215	0.0113	80.97%
14	≥ 0.00156	0.00862	81.90%
16	≥ 0.00118	0.00649	81.82%
18	≥ 0.000928	0.00492	81.14%
20	≥ 0.000748	0.00404	81.48%
(b) Standard log-Normal distribution			
4	≥ 0.0304	0.492	93.82%
5	≥ 0.0249	0.409	93.91%
6	≥ 0.0164	0.343	95.22%
7	≥ 0.0103	0.288	96.42%
8	≥ 0.00698	0.243	97.13%
9	≥ 0.00508	0.205	97.52%
10	≥ 0.00469	0.173	97.29%
11	≥ 0.00435	0.146	97.02%
12	≥ 0.00390	0.123	96.83%
13	≥ 0.00311	0.103	96.98%
14	≥ 0.00257	0.0863	97.02%

3.2.1 Standard Normal Distribution

We firstly analyse the CDF of the standard Normal distribution, which is given as:

$$\Phi(x) := \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right)$$
$$= \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right),$$

where $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \mathrm{d}t$ is the error function.

We present approximations over the range of $x \in [-3.5, 3.5]$ to approximate all values up to an error of one thousandth. Over- and underestimators can be found as before (i.e., with 0 on the negative side, 1 on the positive side, and $\Phi(-3.5) \approx 0.000233$; respectively $\Phi(3.5)$). We present summarised results in Table 3a with an example in Figure 3a.

3.2.2 Standard Log-Normal Distribution

The standard log-Normal distribution assumes the logarithm of the random variable follows the standard Normal distribution. The CDF of the log-Normal distribution satisfies:

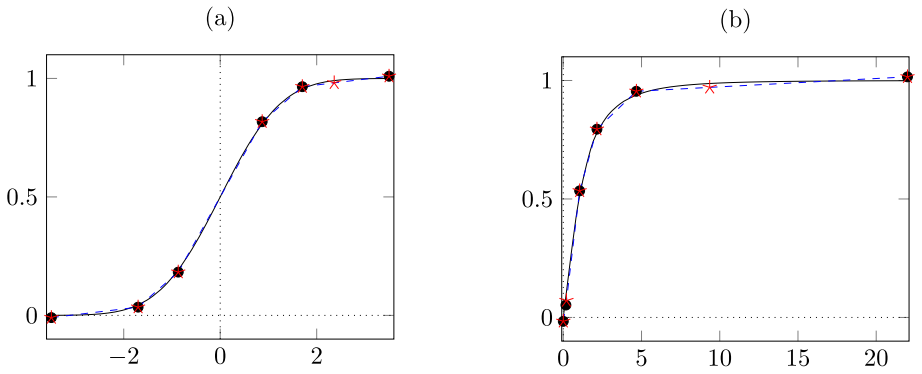

Fig. 3 Optimal PWL approximations for CDFs: (a) Normal distribution with 6 breakpoints; (b) log-Normal distribution with 6 breakpoints. Dashed lines are used to indicate PWL functions with breakpoints at black dots. Red stars indicate where the PWL function achieves its maximum deviation (at least once per linear segment)

$$F_X(x) = \Phi(\ln(x)) = \frac{1}{2}\left(1 + \text{erf}\left(\frac{\ln(x)}{\sqrt{2}}\right)\right).$$

In order to ensure accuracy within one thousandth for values outside the considered range, we model for $x \in (0, 22]$ (since $x = 0$ is undefined, but can be approximated by 0). For $x > 22$, the function can be overapproximated by 1, or underapproximated by $F_X(22)$ with maximum error $1 - F_X(22) \approx 0.000997$ in both cases. In the results in the appendix, we consider the lower bound as 10^{-6} since the PWL approximation method requires the function to be evaluated at the endpoints. There is no apparent symmetry in this function. We present summarised results in Table 3b with an example in Figure 3b.

4 Full Results

The full information about the optimal PWL approximations $p(x)$ to the six functions in tabular form is available online [31]. Specifically, we present optimal breakpoint locations (of the form $(x_b, p(x_b))$, for $b \in [B]$) for PWL approximations across 5 different numbers of breakpoints B, alongside optimal breakpoint locations for PWL over- and underestimators. For each setting, we present the accuracy achieved by the given approximation as a lower bound; note that PWL approximations exist with the same number of presented breakpoints, but with a worse (i.e., higher) accuracy.

The breakpoint locations can also be found using the R package *pwlapprox2d* [1]. As an example, the code below produces breakpoint coordinates of the *sigmoid* function with an accuracy of 0.0274 (i.e., four sets of coordinates) - note that different parameter choices may lead to different (optimal) solutions.

```
library(pwlapprox2d)

# Run optimization (example: Sigmoid)
res <- optimize_main(
  choice = 1,
  accuracy = 0.0274,
  init_points = 200,
  max_iter = 20,
```

```
  verbose = FALSE
)

# Number of breakpoints
bp <- res$breakpoints_coords

#Print breakpoint coordinates
cat("x,y\n")
for(i in 1:nrow(bp)) {
  cat(paste(bp[i, "x"], bp[i, "y"], sep = ","), "\n")
}
```

In order to utilise the information presented in these tables within MILP formulations, the breakpoint locations are used. Different formulations (see e.g., [14, Section 4] or [15]) will utilise different constraints to calculate the argument and function approximations from the breakpoint locations. Therefore, practitioners must decide on their desired accuracy before implementing the PWL approximation within their program.

As discussed in Sections 2-3, the interval bounds are chosen such that all outputs in the desired range can be approximated, up to an error of less than one thousandth. Should practitioners require alternative interval bounds within a given accuracy, the end-breakpoints can be calculated based on interpolation of the presented segments or function values.

We next discuss the cases where the desired intervals do not match with the ones provided in the table. The discussion is for approximations only but also holds for over- and underestimators. In these cases, the function value needs to be carefully rounded up or down to obtain a valid over- or underestimator, respectively.

4.1 Internal Intervals

Assume that you want to construct a PWL function in the interval $[a, b]$. By selecting the desired accuracy, one can find the number of breakpoints required from Tables 1-3 or Tables 4-9. Now assume that the approximation in the table is for interval $[\tilde{a}, \tilde{b}]$ with $\tilde{a} < a$. Then, the PWL function can be constructed as follows.

Assume b_1 is the smallest breakpoint with $b_1 \geq \tilde{a}$. If $b_1 = \tilde{a}$, then you can directly select this breakpoint with the associated function value. If not, then you can use interpolation. Therefore, let f_1 be the PWL function value associated with b_1. Further, let b_{-1} be the largest breakpoint with $b_{-1} < \tilde{a}$ and associated PWL function value f_{-1}. Then, the PWL function value at the new breakpoint location $\tilde{a}$ is given by

$$f_1 - \frac{f_1 - f_{-1}}{b_1 - b_{-1}} \cdot (b_1 - \tilde{a}). \tag{1}$$

If $b < \tilde{b}$, then one can use the same construction as above.

Note that the maximal absolute difference between the PWL function and the non-linear function does not worsen. However, the newly obtained breakpoint location might no longer be optimal in the sense that there might now exist an alternative breakpoint distribution leading to a better accuracy.

Example 1 *(Internal Intervals)* A practitioner would like to approximate the sigmoid activation function to an accuracy of 0.02 over the interval $[-3, 7]$. The results in [31] suggest a PWL function with 6 breakpoints. The four internal breakpoints can be taken directly from the table. The external breakpoints can be interpolated from the existing endpoint segments.

Using the formula (1), we first identify $b_{-1} = -7$, $f_{-1} = -0.00988895$, $b_1 = -2.88703$, $f_1 = 0.0419985$, and obtain

$$-0.00988895 - \frac{0.0419985 + 0.00988895}{-2.88703 + 7} \cdot (-2.88703 + 3) \approx 0.040573319$$

as the function value for breakpoint -3. We obtain for the breakpoints and PWL function values:

$$(-3, 0.040573319)\ (-2.88703, 0.0419985)\ (-1.36777, 0.192332)$$
$$(1.37017, 0.808208)\ (2.89103, 0.958052)\ (7, 1.00989)$$

with a maximal absolute difference of ≥ 0.0108.

4.2 External Intervals

We now consider the case where the goal is to approximate over the interval $[a, b]$ and the approximation in the table is for $[\tilde{a}, \tilde{b}]$ with $a < \tilde{a}$. Otherwise, a new breakpoint can be added at the desired breakpoint a.

Example 2 *(External Intervals)* A practitioner would like to approximate the sigmoid activation function to an accuracy of 0.02 over the interval $[-10, 10]$. The results in [31] suggest a PWL function with 6 breakpoints can be calculated over the interval $[-7, 7]$. Therefore, to extend the function to the desired interval, one extra breakpoint can be included on each side. These can be calculated by simply taking the function value at these new endpoints; i.e., $(-10, \sigma(-10)) = (-10, 0.00004539786871)$ and $(10, \sigma(10)) = (10, 0.9999546021)$.

5 Conclusions

In this paper, we have presented information regarding optimal breakpoint locations for piecewise linear approximations of common non-linear continuous functions that are used within neural networks and probabilistic analyses. This information allows practitioners to easily approximate the given functions to their desired accuracy where necessary. The PWL functions were found using the approach presented by Warwicker and Rebennack [12]; other continuous functions can be approximated using the same strategy. Future work would allow the incorporation of the presented breakpoint locations within packages for commonly used optimisation software, as well as providing analytical solutions to the optimal breakpoint placement problem for a wide variety of functions.

Acknowledgements This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) [Grant 445857709].

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Warwicker, J.A., Rebennack, S.: Pwlapprox2d: approximates univariate continuous functions through piecewise linear regression. (2025). R package version 0.1.0. https://CRAN.R-project.org/package=pwlapprox2d
2. Geißler, B., Martin, A., Morsi, A., Schewe, L.: Using piecewise linear functions for solving MINLPs. In: Mixed Integer Nonlinear Programming, pp. 287–314. Springer, New York, NY (2011)
3. Charnes, A., Cooper, W.W., Rhodes, E.: Measuring the efficiency of decision making units. Eur. J. Oper. Res. **2**(6), 429–444 (1978)
4. Hwangbo, H., Johnson, A.L., Ding, Y.: Spline model for wake effect analysis: characteristics of a single wake and its impacts on wind turbine power generation. IISE Trans. **50**(2), 112–125 (2018)
5. Ding, Y.: Data Science for Wind Energy. Chapman and Hall/CRC, London (2019)
6. Rebennack, S., Krasko, V.: Piecewise linear function fitting via mixed-integer linear programming. INFORMS J. Comput. **32**(2), 507–530 (2020)
7. Kong, L., Maravelias, C.T.: On the derivation of continuous piecewise linear approximating functions. INFORMS J. Comput. **32**(3), 531–546 (2020)
8. Warwicker, J.A., Rebennack, S.: A comparison of two mixed-integer linear programs for piecewise linear function fitting. INFORMS J. Comput. **34**(2), 1042–1047 (2022)
9. Warwicker, J.A., Rebennack, S.: Generating optimal robust continuous piecewise linear regression with outliers through combinatorial benders decomposition. IISE Trans. **55**(8), 755–767 (2023). https://doi.org/10.1080/24725854.2022.2107249
10. Imai, H., Iri, M.: An optimal algorithm for approximating a piecewise linear function. J. Inf. Process. **9**(3), 159–162 (1986)
11. Hakimi, S.L., Schmeichel, E.F.: Fitting polygonal functions to a set of points in the plane. CVGIP: Graph. Models Image Process. **53**(2), 132–136 (1991)
12. Warwicker, J.A., Rebennack, S.: Efficient continuous piecewise linear regression for linearising univariate non-linear functions. IISE Trans. **0**(ja), 1–25 (2024). https://doi.org/10.1080/24725854.2023.2299809
13. Toriello, A., Vielma, J.P.: Fitting piecewise linear continuous functions. Eur. J. Oper. Res. **219**(1), 86–95 (2012)
14. Rebennack, S.: Computing tight bounds via piecewise linear functions through the example of circle cutting problems. Math. Methods Oper. Res. **84**, 3–57 (2016)
15. Warwicker, J.A., Rebennack, S.: In: Pardalos, P.M., Prokopyev, O.A. (eds.) Mixed-Integer Programming Formulations for Piecewise Linear Functions, pp. 1–7. Springer, Cham (2023). https://doi.org/10.1007/978-3-030-54621-2_791-1
16. Frenzen, C.L., Sasao, T., Butler, J.T.: On the number of segments needed in a piecewise linear approximation. J. Comput. Appl. Math. **234**(2), 437–446 (2010)
17. Gablonsky, J.M., Kelley, C.T.: A locally-biased form of the direct algorithm. J. Global Optim. **21**(1), 27–37 (2001)
18. Johnson, S.G.: The NLopt Nonlinear-optimization Package. http://github.com/stevengj/nlopt
19. Rebennack, S., Kallrath, J.: Continuous piecewise linear delta-approximations for univariate functions: computing minimal breakpoint systems. J. Optim. Theory Appl. **167**(2), 617–643 (2015)
20. Anderson, R., Huchette, J., Ma, W., Tjandraatmadja, C., Vielma, J.P.: Strong mixed-integer programming formulations for trained neural networks. Math. Program. **183**(1–2), 3–39 (2020)
21. Fischetti, M., Jo, J.: Deep neural networks and mixed integer linear optimization. Constraints **23**(3), 296–309 (2018)
22. Serra, T., Tjandraatmadja, C., Ramalingam, S.: Bounding and counting linear regions of deep neural networks. In: International Conference on Machine Learning, pp. 4558–4566 (2018). PMLR
23. Tjeng, V., Xiao, K., Tedrake, R.: Evaluating robustness of neural networks with mixed integer programming. arXiv preprint arXiv:1711.07356 (2017)
24. Cheng, C.-H., Nührenberg, G., Ruess, H.: Maximum resilience of artificial neural networks. In: Automated Technology for Verification and Analysis: 15th International Symposium, ATVA 2017, Pune, India, October 3–6, 2017, Proceedings 15, pp. 251–268 (2017). Springer
25. Dumouchelle, J., Patel, R., Khalil, E.B., Bodur, M.: Neur2SP: neural two-stage stochastic programming. arXiv preprint arXiv:2205.12006 (2022)
26. Kronqvist, J., Li, B., Rolfes, J., Zhao, S.: Alternating mixed-integer programming and neural network training for approximating stochastic two-stage problems. arXiv preprint arXiv:2305.06785 (2023)
27. Huchette, J., Vielma, J.P.: Nonconvex piecewise linear functions: advanced formulations and simple modeling tools. Oper. Res. **71**(5), 1835–1856 (2023)
28. Sildir, H., Aydin, E.: A mixed-integer linear programming based training and feature selection method for artificial neural networks using piece-wise linear approximations. Chem. Eng. Sci. **249**, 117273 (2022)

29. Zhang, J., Yang, Y.: Probabilistic score estimation with piecewise logistic regression. In: Proceedings of the Twenty-first International Conference on Machine Learning, p. 115 (2004)
30. Kaya, A., Ghiassi-Farrokhfal, Y., Rebennack, S., Hesamzadeh, M.R., Pan, K., Bunn, D.W.: Delivering reliable reserve commitments from intermittent electricity resources (2024). https://doi.org/10.2139/ssrn.4843369
31. Warwicker, J.A., Rebennack, S.: Optimal piecewise linear approximations for sigmoid, tanh, probability density and cumulative distribution functions in tabular form. https://doi.org/10.5281/zenodo.16362215