Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen

doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P3432-cd

Subinterval Sensitivity for High Dimensional Models

Dawid Ochnio

Civil and Environmental Engineering, University of Strathclyde, United Kingdom.
E-mail: dawid.ochnio.2020@uni.strath.ac.uk

Marco de Angelis

Civil and Environmental Engineering, University of Strathclyde, United Kingdom.
E-mail: marco.de-angelis@strath.ac.uk

This paper introduces an interval-based non-probabilistic sensitivity analysis method, named subinterval sensitivity. A powerful, reliable and rigorous sensitivity analysis method, which is best suited to quantify the importance of inputs purely with respect to their mathematical model. The method has only recently and partially appeared in the literature, while its scalability to high-dimensional models is claimed here for the first time. We apply subinterval sensitivity to quantify and rank the importance of the parameters of a trained neural network model while drawing comparisons with the established Sobol' sensitivity analysis method. Sensitivities on the parameters of a trained neural network can shed light on overparametrization and explainability of the neural network surrogate model.

Keywords: Interval computation, Subinterval reconstitution, Sensitivity analysis, High-dimensional models.

1. Introduction

In recent years, sensitivity analysis has primarily been studied in a probabilistic sense to supersede the more local perturbation-based methods (Saltelli, 2002). Sensitivity analysis is the study of the determination of contributions by the uncertainty of each input variable to the uncertainty of the output within a function or model (Helton et al., 2006). The sensitivities help define the relative importance of each input variable with respect to their given mathematical model. Determining the importance of variables becomes increasingly important when dealing with higher numbers of variables, as when more variables are introduced into a model, the increase of dimensionality means that the number of possible combinations increases exponentially, commonly referred to as the "curse of dimensionality". Knowing which variables to keep or ignore can drastically decrease the number of value combinations in further analysis, e.g. through fixing unimportant variables to predetermined values.

This paper introduces the use of subinterval sensitivity analysis as a way to break the curse of dimensionality without sacrificing rigour, efficiency and scalability. The parameters of a trained neural network model for functional approximation are chosen as the inputs of the sensitivity study, because of the large and easy-to-increase number of inputs. The sensitivities are computed against the neural network output.

Interval analysis methods are often avoided by analysts because of the so called black-box models which impede propagation of intervals as opposed to floating-point numbers (Gray et al., 2023). Interval computations are also avoided in non-black-box models as their conservatism can lead to output intervals that are wider than ought be, e.g. because of repeated variables in the computation graph (Rump, 2010).

There is no reason however, for not using interval computation when the mathematical model is known, even more so when there are no repeated variables. It can be shown with subinterval relations that trained fully-connected multilayer perceptron neural networks implement expressions with no repeated variables, thus are naturally suited to be evaluated with interval arithmetic. Because these models typically contain a large number of parameters, they represent an excellent application for benchmarking interval-based methods on high-dimensional models.

1.1. *Related Work*

The literature on probabilistic sensitivity analysis methods is vast and it includes variance-, entropy-, density- and derivative-based methods. Interval-based sensitivity analysis, on the other hand, is still niche and relatively recent. Perhaps the most popular interval-based sensitivity analysis method is *pinching* due to its simplicity. Pinching operates by reducing the uncertainty of a variable to determine its impact on the total uncertainty of a function output by fixing an uncertain variable to the values where it is most sensitive (Ferson and Tucker, 2006). In Ferson's paper the sensitivity is introduced in general terms with

$$1-\frac{\mathrm{unc}(T)}{\mathrm{unc}(B)},$$

where B is the base value of the risk quantity, T is the risk value computed with an input pinched, and unc() is a measure of the uncertainty of the output. Even though pinching can be understood as a very general method somewhat independent of the uncertainty model, it ties well with interval uncertainty for its simplicity. Whilst Ferson and Tucker (2006) formalized pinching to work with p-boxes, Alvarez (2009) extended the work to more general uncertainty objects like Dempster-Shafer structures using non-specificity measures. Subinterval sensitivity utilizes the formalization introduced by Ferson and Tucker (2006), with the difference that unc() is now an xy-graph area, volume or hyper-volume and T is a sub-intervalised input as opposed to a pinched one. In this sense, subinterval sensitivity can be understood as a generalization of pinching, in fact pinching can be retrieved "for free" from subinterval sensitivity.

The concept of subinterval sensitivity is not new and it has surfaced the literature in different forms, even though it has never been introduced in a manner that the authors deem satisfactory, not least in a reproducibility sense. For example, Miralles-Dolz et al. (2022) were the first who compared subinterval sensitivity against Sobol' while highlighting the advantage of the former being distribution free. Nevertheless, the paper places no emphasis on the scalability of the method to high dimensional models, which has motivated this paper. Other appearances of the method in its non-intrusive optimization- or sampling-based flavor have been sighted in Chang et al. (2022) who used it for black-box engineering models.

2. Subinterval Sensitivity Analysis

2.1. *Interval Computation*

An interval is a compact set of $\mathbb{R}$ denoted $[x]=[\underline{x},\overline{x}]=\{x\in\mathbb{R}:\underline{x}\leq x\leq\overline{x}\}\in\mathbb{IR}$, where $\underline{x}$ is the left and $\overline{x}$ is the right endpoint. $\mathbb{IR}$ denotes the space of intervals (Neumaier, 1990). Interval arithmetic operations must be defined between two intervals $[\underline{x},\overline{x}]$ and $[\underline{y},\overline{y}]$, for example for $(+,-,*,/)$ these are

$$\begin{aligned}
&[\underline{x},\overline{x}]+[\underline{y},\overline{y}]=[\underline{x}+\underline{y},\overline{x}+\overline{y}],\\
&[\underline{x},\overline{x}]-[\underline{y},\overline{y}]=[\underline{x}-\overline{y},\overline{x}-\underline{y}],\\
&[\underline{x},\overline{x}]\cdot[\underline{y},\overline{y}]=[\min A,\max A],\\
&A=\{\underline{x}\underline{y},\overline{x}\underline{y},\underline{x}\overline{y},\overline{x}\overline{y}\},\\
&[\underline{x},\overline{x}]/[\underline{y},\overline{y}]=[\underline{x},\overline{x}]\cdot 1/[\underline{y},\overline{y}]\\
&\quad=[\underline{x},\overline{x}]\cdot[1/\overline{y},1/\underline{y}]\ \text{ if } 0\notin[\underline{y},\overline{y}].
\end{aligned}$$

While these operations are guaranteed to enclose the true answer in the respective intervals, their use in expressions with repeated variables yields inflated intervals due to the dependence problem.

A real-valued function $f:\mathbb{R}^n\to\mathbb{R}^m$ whose expression f is evaluated with interval operations satisfies the fundamental theorem of interval analysis, which ensures the rigor of interval computation. Such an interval extension $\mathsf{f}:\mathbb{IR}^n\to\mathbb{IR}^m$ is said to be inclusion monotonic $\mathsf{f}([\mathrm{x}])\subseteq\mathsf{f}([\mathrm{y}])$, whenever $[\mathrm{x}]\subseteq[\mathrm{y}]$.

2.2. *Subinterval Reconstitution*

In one dimension, this technique partitions an interval into N subintervals typically of equal size. The interval extension is evaluated on each subinterval separately, and finally the output is reconciled by taking the *hull* of all the subinterval outputs (reconstitution). Subinterval reconstitution can be very effective to reduce the inflation of interval computation whenever there are only a few repeated variables to partition. Subinterval sensitivity, on the other hand, utilizes subinterval reconstitution in only one or two dimensions, regardless of the dimensionality of the model, which makes it well suited for high-dimensional models.

2.3. *Subinterval Sensitivity Indices*

Let $f : \mathbb{R}^d \to \mathbb{R}$ be the model, $x = x_1, ..., x_d$ be a bounded vector of inputs, $[\underline{x}, \overline{x}]$ some fixed bounds of interest such that $x \in [\underline{x}, \overline{x}]$, and f an interval extension of f such that $[\underline{y}, \overline{y}] = \mathsf{f}([\underline{x}, \overline{x}])$. After partitioning the i-th input into N tiling subintervals such that $[\underline{x}_i, \overline{x}_i] = \cup_n^N [\underline{x}_i, \overline{x}_i]_n$ and evaluating the interval extension on each of them $[\underline{y}, \overline{y}]_{i,n} = \mathsf{f}([\underline{x}_i, \overline{x}_i]_n)$, the subinterval sensitivity index for the i-th input is

$$S_i = 1 - \frac{\sum_n^N (\overline{x}_i - \underline{x}_i)_n \left(\overline{y} - \underline{y}\right)_{i,n}}{\left(\overline{y} - \underline{y}\right)(\overline{x}_i - \underline{x}_i)}. \quad (1)$$

The numerator in (1) is the sum of all subinterval

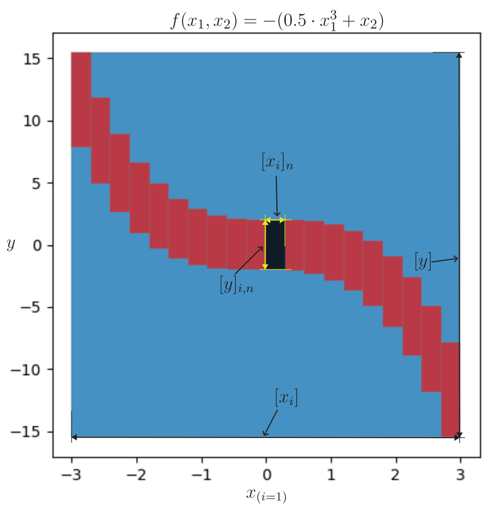

Fig. 1. Visual representation of Eq.(1) using an example where $[y] = \mathsf{f}([-3, 3], [-2, 2])$, $N = 20$, and the highlighted subinterval is $n = 11$ where $[x_{i=1}]_{n=11} = [0, 0.3]$ and $[y]_{i=1,n=11} = [-2.0135, 2]$. The sensitivities for this example are $S_1 = 0.827$ and $S_2 = 0.123$.

areas (see overall area of sub boxes in Fig. 1) and the denominator is the area of the $x_i y$ graph enclosing box (see background box in Fig. 1). The sensitivity index S_i, ranges from 0 to 1 $\forall i \in \{1, ..., d\}$. When $S_i = 0$, the partitioning has no effect, the numerator is equal to the denominator and so y has no functional dependence on x_i. When $S_i = 1$, the subinterval areas are zero and so y has full functional dependence on x_i. It is worth repeating that these sensitivity indices are immune to the curse of dimensionality because the partitioning takes place in one dimension.

3. Experiments

3.1. *The High-Dimensional Model*

Multilayer perceptron neural networks are often used as a surrogate model or functional approximant of engineering models. In this study, the forward sweep of a trained network is considered as the high-dimensional model whilst the inputs of the sensitivity analysis are the networks parameters. Let $f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$ be the forward sweep that is a function of the network input t and parameters x, and let f be the its interval extension. A two-layer neural network is trained to approximate the cubic function $y = t^3 - 3t^2 + 2t + 5$ with five Rectified Linear Units (ReLU), see Fig. 2. Sensitivity indices are computed for each parameter in the trained neural network, also known as weights and biases $W^{(1)} \in \mathbb{R}^{1\times 5}$, $b^{(1)} \in \mathbb{R}^5$, $W^{(2)} \in \mathbb{R}^{5\times 1}$, $b^{(2)} \in \mathbb{R}$. The input parameters for the sensitivity study $x =$

$$W_{11}^{(1)}, W_{21}^{(1)}, ..., b_1^{(1)}, b_2^{(1)}, ..., W_{11}^{(2)}, W_{12}^{(2)}, ..., b^{(2)},$$

are all arranged in a one single vector. The trained network settled on the following values $W^{(1)} = ((-1, -1, 0, 1, 1))$, $b^{(1)} = (-1, 0, 0, -2, -3)$, $W^{(2)} = ((-13, -5, 0, 5, 13))^T$, and $b^{(2)} = 5$.

3.2. *Repeated Parameters in Multilayer Neural Networks*

Even though neural networks are regarded as complex and interconnected functional models, we found that once trained their forward sweep mathematical expression contains no repeated parameters. The absence of repeated parameters and inputs can be visually appreciated in the computation graph in Fig. 3. One can also appreciate that the absence of repeated parameters scales up with the number of layers.

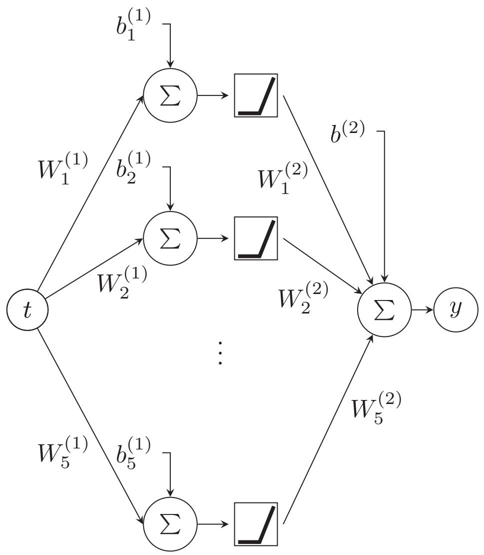

Fig. 2. Visual representation of the neural network used within this paper.

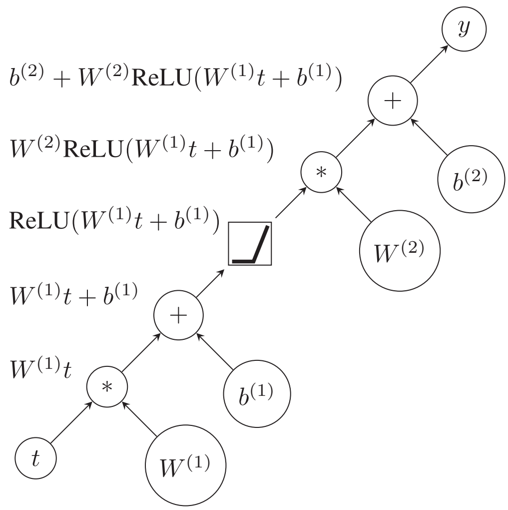

Fig. 3. Computation graph of the neural network's forward sweep being used as a model.

3.3. *Subinterval Sensitivity Analysis*

Subinterval sensitivity is calculated utilizing an input partition of $N = 50$ subintervals. The sensitivity is investigated within a sufficiently large box to appreciate wide deviation from the optimal values after training. This is achieved adding uncertainty to the optimal values as follows $[\underline{x}, \overline{x}] = \hat{x} + [-1, 1]$, equivalent to a radius of ± 1 for each parameter. The forward sweep is evaluated across 31 equally spaced t values $[-2, -1.8, \cdots, 3.8, 4]$. The inputs are organized in the single vector $x = (W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)})$ of size $d = 16$. Fig. 4 shows that parameters $b_2^{(1)}$, $W_5^{(1)}$, and $W_1^{(1)}$ are the most important when looking at the max sensitivity of all parameters. This is mostly expected for $W_5^{(1)}$ and $W_1^{(1)}$ as these parameters have the greatest influence on the units which have to mimic the cubic growth of the function; the $W^{(1)}$ layer is also the first in the computation graph, so this layer has an indirect impact on all other layers, increasing its importance. Table 1 reinforces this information as the t values where max sensitivity for $W_1^{(1)}$ and $W_5^{(1)}$ is achieved are at their respective ends $[-2, 4]$ of the function range where the highest cubic growth occurs. The high max. sensitivity of $b_2^{(1)}$ occurs right as all $W^{(1)}$ parameters and units 1, 4, and 5 have no effect at $t = 0$, removing many typically impactful parameters; as $W_2^{(2)}$ has a much bigger absolute coefficient than $W_3^{(2)}$, it makes sense that $b_2^{(1)}$ is a major contributor to the model output uncertainty.

3.4. *Black-box Subinterval Sensitivity*

It may be argued that black-box models are incompatible with interval computation. This is true e.g. if the model's source code is unavailable, lost, obfuscated, locked, encrypted etc. (Gray et al., 2023). Black-box subinterval sensitivity is not a new concept, it has appeared in the literature a couple of times already (Chang et al., 2022; Miralles-Dolz et al., 2022). The idea is that to keep the same theoretical foundations of the method whilst replacing interval computation with non-intrusive interval propagation. While this idea is legitimate, care must be taken with non-intrusive propagation, especially for non linear and high dimensional models. Non-intrusive interval propagation is implemented using coverage samples for each subinterval aiming to obtain an inner approximation of the endpoints. These results are shown alongside the subinterval sensitivity analysis results in Fig. 4. Black-box subintervals display inflated indices compared to subinterval sensitivity, particularly where the parameters are supposed to have zero impact. Furthermore, for parameters with low impact on the model (such as the $W^{(2)}$ layer), the black-box version is unable to correctly identify the t values of highest importance, displaying a seemingly chaotic pattern. However, for the important parameters, black-box indices seem satisfactory albeit with more fluctuations. It is interesting to note in Fig. 4 that black-box indices seem always greater than the actual indices.

3.5. *Sobol' Total Effect Indices*

Subinterval sensitivity indices are compared against Sobol' total effect indices computed with the Python Sensitivity Analysis Library (SALib) (Herman and Usher, 2017). It seems acceptable to posit that subinterval sensitivity captures interaction effects when a variable is being partitioned while all other variables are left at their full uncertainty. So, the total effects index is a more applicable measure of sensitivity for comparison as it includes the interaction effects. Therefore total indices are chosen over the first-order indices in this study. The analysis is again carried out across 31 equally spaced t values $[-2, -1.8, \cdots, 3.8, 4]$, enabling direct comparison with subinterval sensitivity. $N = 2^{13} = 8192$ samples are used for each parameter in the neural network, with the sensitivity values from Sobol' indices having a statistical interval of ± 0.001 at this number of samples through multiple iterations, showing that results from Sobol' indices are reproducible and reliable for comparisons while also striking a balance with the computation time required. The results in Fig. 5 suggest that the $W^{(2)}$ layer is very insensitive to changes in its value compared to other layers, most likely because the relative uncertainty in that layer, except for unit 3, is far lower compared to the other layers as most other parameters have coefficients in the range $[-1, 1]$ whereas the $W^{(2)}$ layer has coefficients $(-13, -5, 5, 13)$ which are far less sensitive when the uncertainty is the same for all parameters.

3.6. *Introducing a 'Noisy' Unit into the Neural Network*

An additional test using the neural network is the addition of a 'noisy' unit, $f_6(t) = 20 \cdot \mathrm{ReLU}(t - 9)$ where $W^{(1)}_{(6,1)} = (1)$, $b^{(1)}_{(6)} = (-9)$, $W^{(2)}_{(1,6)} = (20)$, designed to have no effect on the model output within the domain $t \in [-2, 4]$. The adjusted network now has shape: $W^{(1)} \in \mathbb{R}^{1\times 6}$, $b^{(1)} \in \mathbb{R}^6$, $W^{(2)} \in \mathbb{R}^{6\times 1}$, $b^{(2)} \in \mathbb{R}$, and values: $W^{(1)} = ((-1, -1, 0, 1, 1, 1))$, $b^{(1)} = (-1, 0, 0, -2, -3, -9)$, $W^{(2)} = ((-13, -5, 0, 5, 13, 20))$, and $b^{(2)} = 5$. The intuition is that this unit will have a sensitivity of zero across all values of t and all other units have the same sensitivities as before, showing that subinterval sensitivity can identify non-important parameters to fix their values. Sobol' indices are calculated as a control test as they can find unnecessary variables when their total effects index $S_{Ti} = 0$ (Saltelli et al. (2008)). Table 1 shows the noisy unit has absolutely no impact on the network, as max senstivity is zero across all three parameters. Also as predicted, the maximum sensitivity stays the same for all other parameters, as seen in Fig. 4 and 5.

4. Discussion

Subinterval sensitivity analysis and Sobol' indices show general agreement in determining the most important parameters of a function, as shown in Fig. 4 and 5. However, looking at the importance of each parameter at a given t like $t = -2$ in Fig. 6 shows that some parameters slightly change ranking across both methods, such as $W_2^{(1)}$ being more important than $b_1^{(1)}$ for subinterval sensitivity and vice versa for Sobol'. Another difference between the methods noticeable in Fig. 6 is the substantially different allocation of sensitivity index values for the most and least non-zero important parameters. For $t = -2$, the most important parameter $W_1^{(1)}$ has twice the relative importance in Sobol' total indices when compared to subinterval sensitivity analysis; the opposite happens with parameters ranked 4th to 10th. However, both methods seem to agree on the relative importance magnitude of somewhat important parameters, such as $W_2^{(1)}$ and $b_1^{(1)}$ at $t = -2$, which suggests that these extreme changes in sensitivity indices between both methods have a point where they switch in over- or underestimating the impor-

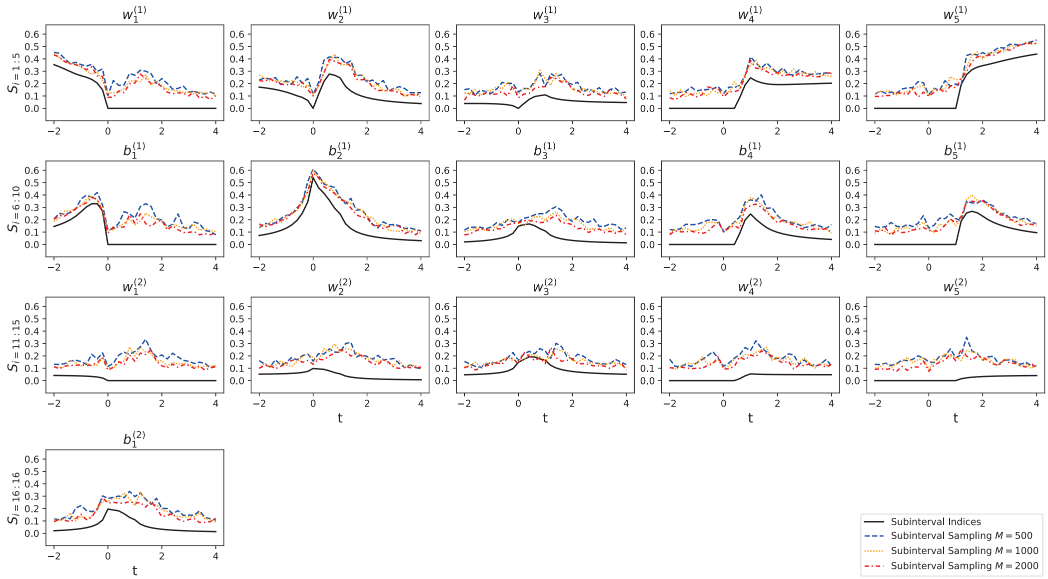

Fig. 4. Subinterval sensitivity indices (continuous line) across $t \in [-2, 4]$ for $x = (W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)})$ plotted against black-box subinterval indices (discontinuous lines) using $M = 200, 400, 800$ samples per subinterval.

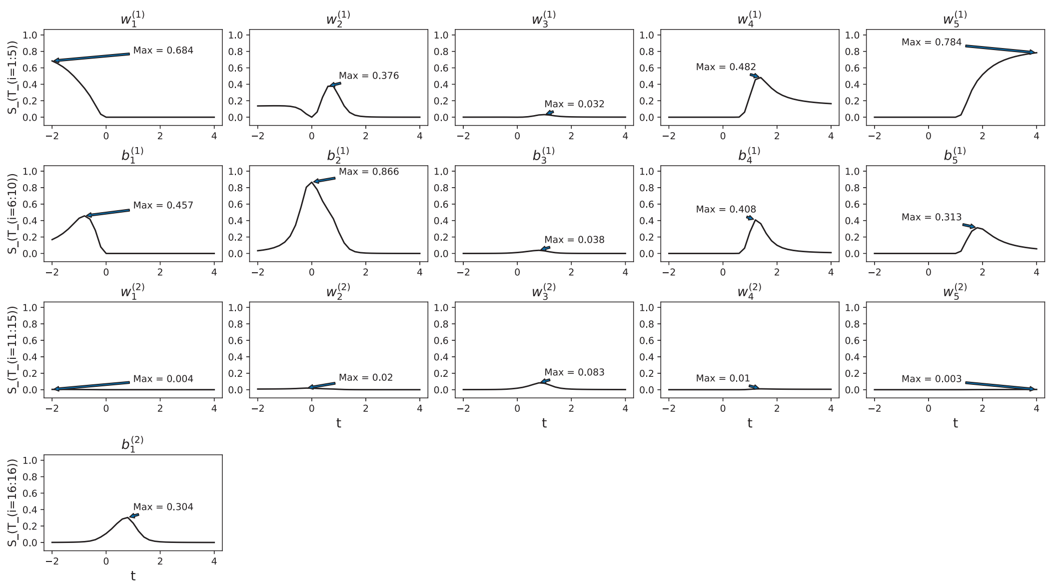

Fig. 5. Sobol' total effect sensitivity indices $2^{13} = 8192$, across $t \in [-2, 4]$ for $x = (W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)})$.

tance of a parameter; this however, is observed to change depending on the relative importance of parameters within a specific value and as such cannot be generalized. These differences show the usefulness of comparing multiple methods, since Sobol' indices better highlight which parameter seems to be most important (factor prioritization), while subinterval sensitivity analysis more clearly

Table 1. Max sensitivity values for Sobol and Subintervals when noisy unit is added into the network. As noisy unit has no impact on output, this table also describes the max sensitivity values for Sobol and Subinterval method when no noisy unit is present as the results are identical in both tests.

$\max_t$	Subinterval Sensitivity											
$S_i(t)$	Unit 1		Unit 2		Unit 3		Unit 4		Unit 5		Unit 6	
$W^{(1)}$	0.35	$t=-2.0$	0.28	$t=0.6$	0.11	$t=1.0$	0.25	$t=1.0$	0.44	$t=4.0$	0.0	$\forall t$
$b^{(1)}$	0.33	$t=-0.4$	0.54	$t=0$	0.17	$t=0.4$	0.25	$t=1.0$	0.27	$t=1.6$	0.0	$\forall t$
$W^{(2)}$	0.04	$t=-2.0$	0.10	$t=0$	0.19	$t=0.4$	0.05	$t=1.0$	0.04	$t=4.0$	0.0	$\forall t$
$b^{(2)}$	0.20	$t=0$										
$\max_t$	Sobol' Indices											
$S_i(t)$	Unit 1		Unit 2		Unit 3		Unit 4		Unit 5		Unit 6	
$W^{(1)}$	0.68	$t=-2.0$	0.38	$t=0.6$	0.03	$t=1.0$	0.48	$t=1.4$	0.79	$t=4.0$	0.0	$\forall t$
$b^{(1)}$	0.46	$t=-0.8$	0.87	$t=0$	0.038	$t=0.8$	0.41	$t=1.2$	0.31	$t=1.8$	0.0	$\forall t$
$W^{(2)}$	0.00	$t=-2.0$	0.02	$t=-0.2$	0.08	$t=0.8$	0.01	$t=1.4$	0.00	$t=4.0$	0.0	$\forall t$
$b^{(2)}$	0.30	$t=0.8$										

shows which parameters hold no importance in the sensitivity of a model or function (factor fixing). A comparison of the execution time for both methods found that subinterval sensitivity analysis ($\sim$ 300 sec per run) is twice as fast compared to Sobol' indices ($\sim$ 600 sec per run). As such, subinterval sensitivity may be a preferable method for high-dimensional models due to its faster analysis with comparable accuracy in results.

5. Conclusion

This paper has introduced subinterval sensitivity, a non-probabilistic, interval-based and rigorous sensitivity analysis method, with application to the parameters of a trained fully-connected neural network model. Aside from formally introducing and naming the method, the aim is to show that subinterval sensitivity can be scaled to high-dimensional models while obtaining results that are comparable to Sobol' total-effect indices, a renowned sensitivity gold standard. Subinterval sensitivity has demonstrated prominent numerical correctness on the high-dimensional example, surpassing Sobol' total effects in efficiency and certainty. The efficiency is easily explained by the number of model evaluations required by the two methods. On the 16-dimensional model, Sobol' total indices require at least 8192 evaluations, while subinterval sensitivity only just about $50 \times 16 = 800$ evaluations in interval arithmetic. The certainty of the method stems from the mathematical rigour of interval computations, which makes the method reliable against spiky nonlinearities and safe against numerical instability.

The black-box version of subinterval sensitivity, popular among engineers, has also been investigated. Such a version replaces interval computation, arguably unavailable in black boxes, with coverage samples for endpoints inner approximation. Our study on the 16-dimensional model shows that such black-box method fails to correctly identify the unimportant parameters. More precisely, the parameters that have no effect whatsoever on the model appear to have significant sensitivities (false positives). This behaviour is to be attributed to the non-intrusive propagation of intervals via sampling, which is unsuitable in high-dimensional models. We would discourage the use of black-box subinterval sensitivity in models with input dimension higher than five.

The introduction of a 'noisy' unit within the neural network model, that is known to have no effect on the output has enabled subinterval sensitivity to be tested further against false positives. Sobol' total indices are already known to work well on these examples to confirm the lack of importance of the noisy parameters. Subinterval sensitivity has worked as expected proving that the 'noisy' unit has no effect on the model output.

Subinterval sensitivity is still largely unexplored and further research is needed to truly understand its mathematical implications in multi-

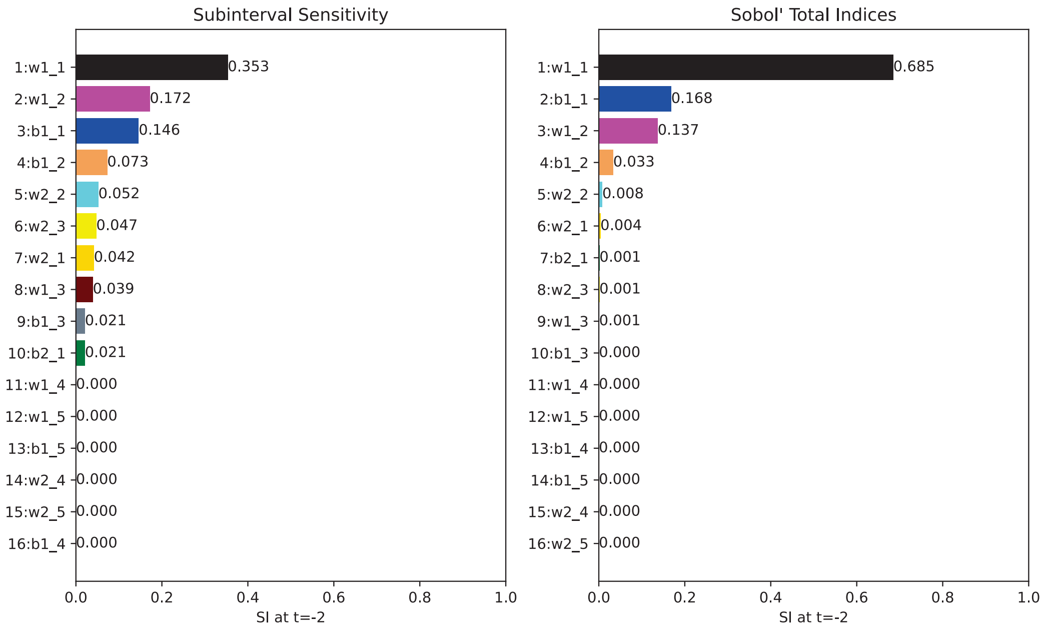

Fig. 6. Ranking of sensitivity indices at $t = -2$ for Sobol' Total Indices and Subinterval sensitivity analysis.

variate functional analysis.

Acknowledgement

This research is funded by the University of Strathclyde's StrathDRUMS centre for doctoral training.

References

Alvarez, D. A. (2009). Reduction of uncertainty using sensitivity analysis methods for infinite random sets of indexable type. *International Journal of Approximate Reasoning 50*(5), 750–762.

Chang, Q., C. Zhou, M. A. Valdebenito, H. Liu, and Z. Yue (2022). A novel sensitivity index for analyzing the response of numerical models with interval inputs. *Computer Methods in Applied Mechanics and Engineering 400*, 115509.

Ferson, S. and W. T. Tucker (2006). Sensitivity analysis using probability bounding. *91*(10), 1435–1442.

Gray, N., M. De Angelis, and S. Ferson (2023). Towards an automatic uncertainty compiler. *International Journal of Approximate Reasoning 160*, 108951.

Helton, J., J. Johnson, C. Sallaberry, and C. Storlie (2006). Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability engineering & system safety 91*(10), 1175–1209.

Herman, J. and W. Usher (2017, Jan). SALib: An open-source python library for sensitivity analysis. *The Journal of Open Source Software 2*(9).

Miralles-Dolz, E., A. Gray, M. de Angelis, and E. Patelli (2022). Interval-based global sensitivity analysis for epistemic uncertainty. Research Publishing.

Neumaier, A. (1990). *Interval methods for systems of equations*. Number 37. Cambridge university press.

Rump, S. M. (2010). Verification methods: Rigorous results using floating-point arithmetic. *Acta numerica 19*, 287–449.

Saltelli, A. (2002). Sensitivity analysis for importance assessment. *Risk analysis 22*(3), 579–590.

Saltelli, A., F. Campolongo, S. Tarantola, J. Cariboni, M. Saisana, D. Gatelli, M. Ratto, and T. Andres (2008). Global sensitivity analysis. the primer. In *Global Sensitivity Analysis*. John Wiley and Sons, Incorporated.