Second Glance: A Novel Explainable AI to Understand Feature Interactions in Neural Networks using Higher-Order Partial Derivatives

Zohaib Shahid[1,*], Yogachandran Rahulamathavan,[1] and Safak Dogan[1]

[1]*Institute for Digital Technologies, Loughborough University London, United Kingdom*

Abstract

Neural networks often operate as "black boxes," making understanding how they arrive at their decisions difficult. To build trust and improve neural networks, it is essential to identify the most salient inputs and how they interact within the network. We present "Second Glance," a novel approach for performing second-order sensitivity analysis on neural networks with Rectified Linear Unit (ReLU) activations to address this. The first-order sensitivity analysis quantifies the individual influence of the input features on the model output. However, it fails to capture how features interact, potentially leading to misleading conclusions. Second-order sensitivity analysis, using second-order partial derivatives, can reveal these interactions, providing a more comprehensive understanding of the model's inner workings. Unfortunately, ReLU activation, a popular choice because of its efficiency, introduces zero second-order partial derivatives. To overcome this limitation, Second Glance employs a two-stage strategy. First, it trains a primary neural network with ReLU activations. Then, it trains a separate "surrogate" model using the concerned features as the input and the first-order partial derivatives obtained from the primary model as its output. In this paper, we show that the subtle second-order sensitivity analysis of the original neural network with ReLU activation function can be effectively obtained by analyzing the first-order partial derivatives of the surrogate model. We further validate the proposed method by experimenting with popular UCI bank marketing and UCI adult income datasets.

Keywords

Feature Interactions, Higher-Order Sensitivity Analysis, Interpretable AI

1. Introduction

In the context of explainable AI (XAI), sensitivity analysis is the quantification and evaluation of the sensitivity of the output of a machine learning model to changes in its input features. Concerning sensitivity analysis, the focus of this research is on neural networks. Sensitivity analysis in neural networks involves assessing the impact of input variations on the neural network's predictions. First-order sensitivity analysis is the technique whereby the impact of a single input on the output is measured. One can also think of it as measuring a linear change in the output concerning an input. Second-order sensitivity analysis is done to understand how different inputs affect or interact. This type of analysis is concerned with measuring the nonlinear changes in an output concerning a number of inputs.

Late-breaking work, Demos and Doctoral Consortium, colocated with The 2nd World Conference on eXplainable Artificial Intelligence: July 17–19, 2024, Valletta, Malta

*Corresponding author.

z.shahid@lboro.ac.uk (Z. Shahid)

It is understood that a deeper understanding of the behaviour of the neural networks can be achieved by quantifying how *features interact to affect predictions* [1]. There are many ways to measure the interaction of features like Shap-iq (Computation of Shapley interactions for arbitrary cardinal interaction indices by using a sampling-based approximator) [2] and analyzing the directed graph made by bivariate methods [3]. This research focuses on feature interactions in neural networks based on partial derivatives like the usage of rule ensembles [4], analyzing interactions in non-linear models [5] and Integrated Hessians [6]. Concerning a function and a point, the Interaction Effect between a concerned set of features denotes the partial derivative of the function output with respect to the features. The partial derivatives show the small changes in the function caused by the change in each chosen feature. Our research is around *pair-wise interactions* or second-order partial derivatives, which constitute the elements of the Hessian matrix.

Neural networks, based on ReLU activation functions, have valuable properties like mitigating the vanishing gradient problem [7]. Concerning feature interactions, the issue is that these ReLU networks are piece-wise linear. Therefore, they generate a zero Hessian almost everywhere, and studying the feature interactions in such networks is impossible. The proposed approach, Second Glance, as shown in Figure 1, mitigates this issue by taking the first-order partial derivatives of the concerned ReLU-based neural network (primary model **M1**) and training a *surrogate* model (**M2**). Table 1 explains what pair-wise feature interactions mean according to their sign (direction) using one input as an example and vice versa.

Table 1
Explaining the meaning of feature interactions (1st order and 2nd order partial derivatives) in neural networks

$\frac{\partial y}{\partial x_1}$	$\frac{\partial}{\partial x_1}\left(\frac{\partial y}{\partial x_2}\right)$	**Explanation**
$+ve$	$+ve$	As the 1st order partial derivative of x_1 is positive, this means that when x_1 increases, the output of the neural network increases. As the 2nd order partial derivative is also positive, the rate at which x_2 changes, increases. In short, the impact of x_2 on the output is amplified by x_1.
$+ve$	$-ve$	The output will increase, as the 1st order partial derivative is positive. As the 2nd order partial derivative is negative, the rate of change of x_2 decreases with the increase in x_1. In short, the influence of x_2 on the output is dampened by x_1.
$-ve$	$+ve$	The output will decrease due to the negative value of the 1st order partial derivative of x_1 (or when x_1 is increased). As the rate of change of x_2 due to the rate of change of x_1 is positive, this means that the rate of change of x_2 increases. In summary, the absence of x_1 magnifies the influence of x_2 on the output.
$-ve$	$-ve$	The negative sign of the 1st order partial derivative of x_1 indicates an inversely proportional relationship between itself and the output. The 2nd order partial derivative being negative shows that the effect of x_2 on the output decreases as x_1 becomes more negative.

Section 2 gives a brief literature review of gradient-based sensitivity analysis. Section 3 explains the functioning of Second Glance. Section 4 will show some experiments on 2 popular UCI datasets using Second Glance and how it can lead to another way of estimating feature interactions where zero Hessians are an issue. Overall, Second Glance aims to provide a more granular analysis of feature interactions.

2. Literature Review

The gradient-based sensitivity analysis methods will be focused on as they are more relevant to this research. Given a sample, the gradient-based methods use the natural interpretation of the gradient as the infinitesimally local importance. A well-known approach is the saliency map [8], which is simply the gradient of model output with respect to the input. SmoothGrad [9] mitigated the noise in saliency maps by averaging them and came up with sample complexity guarantees. Research related to Grad-CAM [10] is gradient-based with the main distinction that the importance is calculated over hidden (internal) layers. The calculation of the Jacobian matrix or the matrix of first-order partial derivatives has been thoroughly discussed by [11].

Higher-order interactions are estimated using gradient-based approaches like Gradient-NID [12], which estimates the corresponding Hessian element squared as the strength of feature interaction. By extending Integrated Gradients to utilize a path-integrated Hessian, [6] came up with Integrated Hessian. SmoothHess by [1] convolves the Hessian Matrix of a ReLU network with a Gaussian to mitigate the issue of zero Hessians.

Though these methods handle ReLU networks in their way, like the replacement of ReLU function with SoftPlus post-hoc before applying Integrated Hessians [6], similar usage of SoftPlus activation by [12] and the usage of Stein's Lemma by [1], there are not many methods that use surrogate models for second-order sensitivity analysis, specifically. [13] uses AI-surrogate models to estimate the relationships between input features and ventricular parameters for medical applications; it does not focus specifically on second-order sensitivity analysis. The commendable work by [14] uses surrogate models for point cloud deep neural networks based on LIME (Local Interpretable Model-Agnostic Explanations). The use of generalized additive models (surrogate models) with pairwise interactions (GA2M) has been explored to understand the trade-off between accuracy and interpretability in machine learning techniques applied to clinical data [15] but it does not focus on using partial derivatives. In contrast, Second Glance targets global explainability by generating second-order partial derivatives of the primary model using the surrogate model.

3. Proposed Algorithm

In the two-stage process of **Second Glance** (Figure 1), the primary neural network or model (M1) is trained and its first-order partial derivatives are obtained. These are put together with the inputs as a dataset to train the surrogate model or neural network (M2). The surrogate model (M2) is the main contribution, where the inputs are the features of the primary model, and the outputs are the first-order partial derivatives from the primary model. The second-order partial

derivatives or the Hessian of the primary model can be obtained by calculating the first-order partial derivatives of the surrogate model.

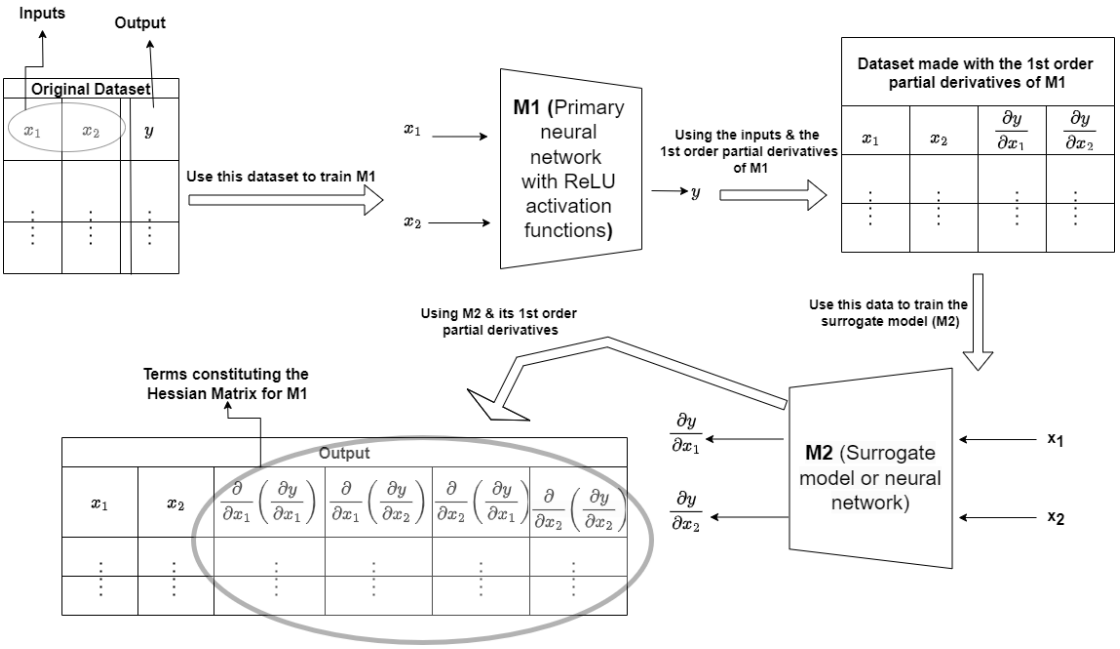

Figure 1: High-level view of the proposed Second Glance algorithm

If M_1 takes an input, x and produces an output, y, its first-order partial derivative will be $M_1'(x)$, as shown in (1). The first-order partial derivative of (M_1) will be used as the output for the surrogate model, M_2, which takes an input of x. As shown in (2), the first-order partial derivative of M_2 will indeed be equal to the second-order partial derivative of M_1 (represented by $\frac{\partial^2 y}{\partial x^2}$). In other words, this happened because the first-order partial derivatives (from M1) are backpropagated to the inputs, in M2 to get the second-order partial derivatives.

$$y = M_1(x)\ ;\ \frac{\partial y}{\partial x} = M_1'(x) \qquad (1) \qquad\qquad \frac{\partial y}{\partial x} = M_2(x)\ ;\ \frac{\partial^2 y}{\partial x^2} = M_2'(x) \qquad (2)$$

Following this approach, we can obtain higher-order partial derivatives i.e., to obtain 3rd-order partial derivatives, we can train a third model (M3) using x as inputs but using the first-order partial derivatives of the $M2$ as outputs. The first-order partial derivatives of M3 would be the 3rd order partial derivatives of M1. The 3rd order of partial derivatives identifies how a change in two features impacts the change in the third feature on the prediction. In this preliminary study, we focus only on the 2nd order sensitivity analysis.

4. Experiments with Second Glance

To test Second Glance, the UCI bank marketing [16] and UCI adult income [17] datasets were used, which are for classification problems. They were selected as they are well-known bench-

mark tabular datasets used for testing neural networks. The most influential 5 features from each dataset were selected using SHAP to make it easy to understand and present the functioning of Second Glance. However, the proposed approach can support an arbitrary number of features. The UCI bank marketing dataset contains data for marketing campaigns based on phone calls, and the target was to assess whether a client would subscribe to a term deposit (*yes* or $y = 1$) or not (*no* or $y = 0$). This dataset has a total of 41,188 instances and 19 multivariate features. The UCI adult income dataset, which aims to predict whether a person will make over $50K per year or not, is a multivariate dataset with 30,162 instances (cleaned dataset) and 14 features.

For simplicity, we kept the same architecture for $M1$ for both of these datasets as follows: 5 inputs, 3 hidden layers with 4 neurons each (ReLU activation is used in the hidden nodes), and 1 output neuron (Sigmoid activation) to ensure uniformity. Binary crossentropy was used as the loss. The hidden layers carry ReLU activation because the surrogate model (from Second Glance) will be created to analyze and mitigate the effect of zero Hessians due to ReLU activations. The selected 5 features and performance metrics of $M1$ and $M2$ are in Table 2.

It can be seen that the primary neural network trained on the UCI bank marketing gives high values of accuracy, recall, and F1 score. The performance metrics of M1 for the UCI adult income dataset are decent. The explainable AI model, made from any model, only gives accurate explanations as long as the performance of the original model is high, so it is essential to ensure that. As M2 had continuous values (first-order partial derivatives from M1) as the output, the R-squared score was used as the performance metric.

Table 2
Table showing the performance metrics for both M1 and M2

Dataset	Selected 5 features	Accuracy of M1	Recall of M1	F1 score of M1	R-squared score of M2
UCI bank marketing dataset	`emp.var.rate`, `euribor3m`, `cons.price.idx`, `contact`, `previous`	88.9%	98%	0.94	0.914
UCI adult income dataset	`age`, `workclass`, `education_num`, `marital_status`, `hours_per_week`	73.1%	77.5%	0.59	0.826

Table 3 shows some of the first-order partial derivatives obtained from the primary neural network trained on the adult income dataset. As there were 5 inputs, the number of partial derivatives per row is also 5. The range of the partial derivatives is between -1 and 1. As discussed in Table 1, the positive values mean that the output of the model increased with the change in the feature. Meanwhile, the negative values depict an inverse relationship between the input and the output.

The surrogate models (M2) were trained for both datasets with different architectures. It is emphasized that the surrogate model can have any architecture. The given architecture was

Table 3
Table showing some of the first-order partial derivatives of M1 for the UCI adult income dataset

age	workclass	education_num	marital_status	hours_per_week
0.939	-0.482	1.00	-1.00	0.998
1.00	-1.00	-0.476	-0.949	-0.300
0.195	-0.496	1.00	-1.00	0.387

picked to get the best possible performance. Each M2 had 5 input features and the relevant first-order partial derivatives of M1 as the outputs (5 output neurons with $tanh$ as the activation function to place the continuous values within a suitable range). The number of instances of input features (along with the choice of inputs) was the same as M1 for M2 in each case.

Concerning M2 for the bank marketing dataset, there were 3 hidden layers, with ReLU activation used in the first 2 layers and sigmoid activation in the last hidden layer. Concerning the adult income dataset, the surrogate model had 5 hidden layers. ReLU was used in the first 2 layers. The 3rd and 4th layers had GeLU (Gaussian error Linear Unit) activation. Sigmoid was used in the last hidden layer. The loss used in both cases was Mean Squared Error. The R-squared score for the trained surrogate (M2) models in Table 2 shows that the models performed modestly. The first-order partial derivatives of all M2 models were obtained by using (2) and accordingly, the first-order partial derivatives (or the Jacobian matrix) of M2 indeed represent the Hessian matrices of $M1$ in each case.

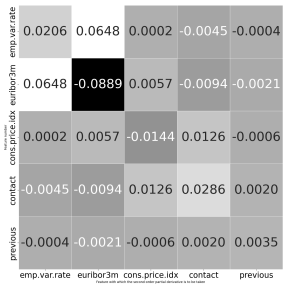

Figure 2: Feature Interaction Based on SHAP.

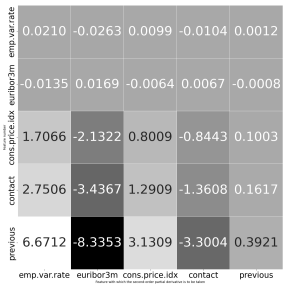

Figure 3: Feature Interaction based on Second Glance.

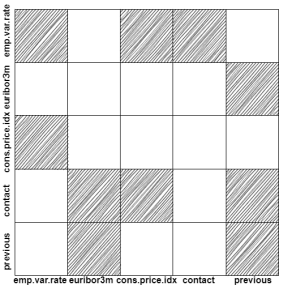

Figure 4: Matching Polarities.

SHAP interactions were calculated from XGBoost Classifiers trained for both datasets to validate the Hessian matrices generated by Second Glance. XGBoost Classifiers were used because the present version of the SHAP Python library can only calculate interactions for XGBoost models [18]. Although SHAP interactions and Second Glance are very different in implementation, feature interactions exist regardless of the model used, as long as the relationship between the features influences the outcome [19]. In Figure 3, each value represents the measure of interaction (second-order partial derivative) between the features. The black squares represent highly negative values, while the white represent highly positive values. The grey represent the rest of the values that lie between them. The negative and positive values (polarity) play a significant role in interpreting the results. The SHAP interactions and the Hessian matrices were calculated for all the instances of both datasets for a better understanding.

Upon comparing the polarities, it was found that for the bank marketing dataset, 45.4% of the polarities were the same in the SHAP interactions and the Hessian matrices. This amount was 50.4% in the case of the adult income dataset. This validates the correctness of the proposed approach. The reason is that if the proposed approach's outputs are random then the probability of matching 50% of the polarities between both the approaches would be around $\frac{1}{2^{12.5}} \approx 0.01\%$.

For the selected features for a single datapoint of the bank marketing dataset, the Hessian matrix (Figure 3) has been compared with the SHAP interactions (Figure 2). As shown in Figure 4, nearly 40% of the total feature pairs have similar polarities. SHAP interactions show the absolute impact on the output due to the interactions, while Second Glance shows the increase or decrease in the rate of change of an output with respect to the interaction between features (as explained generally in Table 1). For example, in terms of the interaction of the `emp.var.rate` with itself, the effect on the output is positive. The relevant SHAP interaction (Figure 2) shows that the probability of a client subscribing to a term deposit increased by 2.06% while Figure 3 shows that this interaction amplifies the influence of `emp.var.rate` on the output. In terms of the interaction between `previous` and `euribor3m`, both heatmaps carry a negative value near zero. This confirms that there is no or less effect on the output of this interaction. The value of -3.4367 corresponding to `contact` and `euribor3m` (Figure 3) means that the influence of `contact` is lessened or dampened by `euribor3m` or vice versa. As the influence of one of the features is being dampened, the corresponding SHAP interaction shows that there is indeed a negative impact on the output, but not a lot (overall output not much affected). In short, the probability of a subscription by a client decreased but not significantly.

5. Conclusions and Future Works

The proposed method, Second Glance, provides a unique post-hoc way to generate Hessians for ReLU-based neural networks. It opens up another research direction where surrogate models and more granularity can be considered while aiming to generate non-zero Hessians from ReLU-based neural networks. We have done some preliminary experiments with the tabular UCI bank marketing and UCI adult income datasets and interpreted what the result (Hessian), produced by Second Glance, shows and validated the results with the SHAP feature interactions. Our research aims to expand Second Glance's capabilities to encompass image datasets. As a future research direction, we will conduct a rigorous comparison against contemporary gradient-based second-order sensitivity analysis algorithms, scrutinizing metrics such as the frequency of zeros in the Hessian and symmetry while prioritizing enhancements in efficiency.

References

[1] M. Torop, A. Masoomi, D. Hill, K. Kose, S. Ioannidis, J. Dy, Smoothhess: Relu network feature interactions via stein's lemma, Advances in Neural Information Processing Systems 36 (2024).

[2] F. Fumagalli, M. Muschalik, P. Kolpaczki, E. Hüllermeier, B. Hammer, Shap-iq: Unified approximation of any-order shapley interactions, Advances in Neural Information Processing Systems 36 (2024).

[3] A. Masoomi, D. Hill, Z. Xu, C. P. Hersh, E. K. Silverman, P. J. Castaldi, S. Ioannidis, J. Dy, Explanations of black-box models based on directional feature interactions, arXiv preprint arXiv:2304.07670 (2023).
[4] J. H. Friedman, B. E. Popescu, Predictive learning via rule ensembles (2008).
[5] C. Ai, E. C. Norton, Interaction terms in logit and probit models, Economics letters 80 (2003) 123–129.
[6] J. D. Janizek, P. Sturmfels, S.-I. Lee, Explaining explanations: Axiomatic feature interactions for deep networks, Journal of Machine Learning Research 22 (2021) 1–54.
[7] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: Proceedings of the fourteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.
[8] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
[9] D. Smilkov, N. Thorat, B. Kim, F. Viégas, M. Wattenberg, Smoothgrad: removing noise by adding noise, arXiv preprint arXiv:1706.03825 (2017).
[10] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, 2017 IEEE International Conference on Computer Vision (ICCV) (2017). doi:10.1109/iccv.2017.74.
[11] J. Pizarroso, J. Portela, A. Muñoz, Neuralsens: sensitivity analysis of neural networks, arXiv preprint arXiv:2002.11423 (2020).
[12] M. Tsang, D. Cheng, H. Liu, X. Feng, E. Zhou, Y. Liu, Feature interaction interpretability: A case for explaining ad-recommendation systems via neural interaction detection, arXiv preprint arXiv:2006.10966 (2020).
[13] G. D. M. Talou, T. P. B. Gamage, M. P. Nash, Efficient ventricular parameter estimation using ai-surrogate models, Frontiers in Physiology 12 (2021). doi:10.3389/fphys.2021.732351.
[14] H. Tan, H. Kotthaus, Surrogate model-based explainability methods for point cloud nns (2021). doi:10.48550/arxiv.2107.13459.
[15] T. Karatekin, S. Sancak, G. Celik, S. Topcuoglu, G. Karatekin, P. Kirci, A. Okatan, Interpretable machine learning in healthcare through generalized additive model with pairwise interactions (ga2m): Predicting severe retinopathy of prematurity, in: 2019 international conference on deep learning and machine learning in emerging applications (deep-ML), IEEE, 2019, pp. 61–66.
[16] R. P. Moro, S., P. Cortez, Bank Marketing, UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C5K306.
[17] B. Becker, R. Kohavi, Adult, UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.
[18] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, S.-I. Lee, From local explanations to global understanding with explainable ai for trees, Nature Machine Intelligence 2 (2020) 2522–5839.
[19] A. Jakulin, I. Bratko, Analyzing attribute dependencies, Knowledge Discovery in Databases: PKDD 2003 (2003) 229–240. doi:10.1007/978-3-540-39804-2_22.