Visualizing Pairwise Feature Interactions in Neural Additive Models

C. Steinparz, A. Hinterreiter, and M. Streit

Johannes Kepler University Linz, Austria

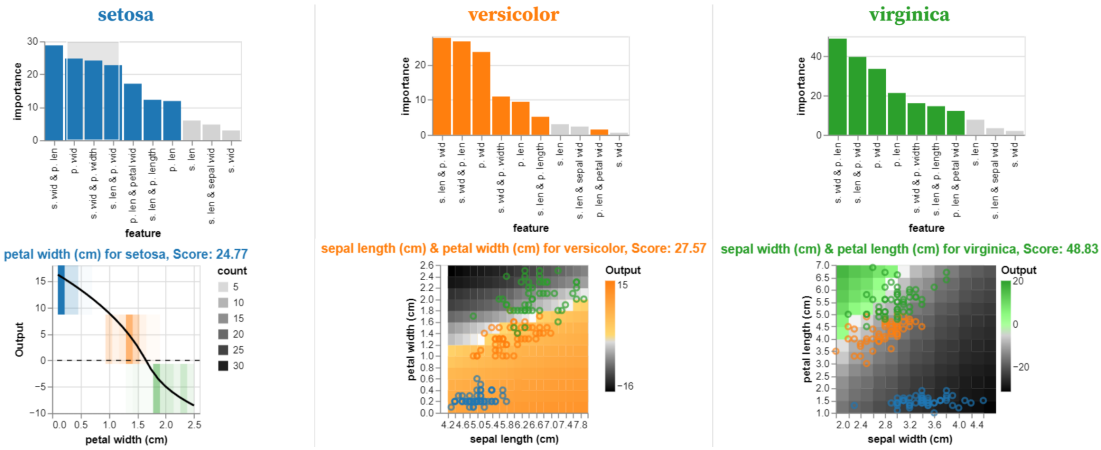

Figure 1: *Sorted 1-D and 2-D feature maps of a NAM trained to classify the Iris dataset are presented. For 1-D feature maps, the x-axis represents the input feature for the corresponding submodule, while the line illustrates the submodule's output. The background displays the data distribution for each class along the specific feature. For 2-D feature maps, the axes indicate the input features used for the submodule. The color map visualizes the submodule's output, with black areas denoting negative output and class-colored areas signifying positive output. A scatterplot demonstrates the class distribution based on the two input features. In this case, the feature maps are scored by their output range, and then sorted and filtered to show the highest scoring per class. Note that the tool can display multiple rows of feature maps, which is not depicted here. The tool can be found at* `https://observablehq.com/@cursedseraphim/nams-vis`.

Abstract

We present an approach for incorporating feature interactions into Neural Additive Models (NAMs), building upon existing work in this area, to enhance their predictive capabilities while maintaining interpretability. Our contribution focuses on the visual exploration and management of the increased number of feature maps resulting from the addition of pairwise feature combinations to NAMs. This method allows for effectively visualizing individual and pairwise feature interactions using line plots and heatmaps, respectively. To address the potential explosion in the number of feature maps, we apply different scoring functions to compute the importance of a feature map and then filter and sort them based on their importance. The proposed interactive dashboard effectively manages large sets of feature maps, while preserving the white-box properties of NAMs.

CCS Concepts

• ***Computing methodologies*** → ***Neural networks;*** • ***Human-centered computing*** → ***Visual analytics;***

1. Introduction

Neural Additive Models (NAMs) [AMF*21] blend the expressiveness of deep neural networks with the interpretability of generalized additive models. NAMs learn a linear combination of neural network submodules, each focusing on a single input feature. As a result, their main advantage is the ability to generate 1-D feature maps, as the submodules are trained on a single feature each.

A major drawback of NAMs is their inability to account for feature interactions. Consequently, they struggle to model intricate relationships between features that may be crucial for specific tasks. To mitigate this shortcoming, incorporating 2-D feature interactions into NAMs has been suggested [YZS21] [LCGH13] [CCG21]. By doing so, NAMs can represent more complex relationships between features, potentially enhancing their perfor-

mance in certain tasks. Furthermore, this extension still allows for the interpretability of submodules that learn from two features, as they can be effectively visualized using heatmaps.

Various machine learning solutions have been proposed that incorporate feature interactions, including pairwise and higher-order interactions. However, the need for a visualization approach arises when dealing with an overwhelming number of feature maps, as incorporating all possible features and pairwise feature interactions results in $n + n(n-1)/2$ features per class, where n denotes the number of features. This calls for an efficient visualization strategy to manage and interpret the increased complexity introduced by the feature interactions.

2. Related Work

Methodologically, NAMs are part of the Generalized Additive Models (GAMs) [Has17] family, which do not originally include feature interactions. While our work specifically extends NAMs to include pairwise interactions, the visualization contribution can be applied to any set of feature maps, regardless of the specific implementation. This includes NAMs or other similar models, such as Explainable Boosting Machines [CLG*15].

Existing machine learning research on feature interactions in GAMs focuses on determining relevant interactions for model training and pruning the model accordingly. However, these studies often overlook visualization techniques to support understanding a potentially large final set of feature maps. GAMI-Net [YZS21] uses an interaction ranking algorithm to select the top-K pairwise interactions. Lou et al. propose tree-based GA^{2}M [LCGH13] and address weaknesses of existing interaction measures, by introducing a fast interaction detection algorithm to rank feature pairs as candidates for model inclusion. Chang et al. present NODE-GAM [CCG21] based on NODE [PMB19] and GA^{2}M and design gating mechanisms that gradually reduce higher-order feature interactions, allowing the architecture to automatically perform feature selection for both marginal and pairwise features.

From a visualization perspective, the most relevant work is GAM-Changer [WKN*21], an open-source interactive system that assists data scientists and domain experts in easily and responsibly editing their GAMs. GAM-Changer employs an importance score based on the weighted average of a feature's absolute contribution, displaying the single highest-scoring feature upon starting the tool.

In contrast to these approaches, our work emphasizes the use of visualization techniques to comprehend large numbers of feature maps. To this end, we employ an interactive dashboard that enables users to simultaneously explore multiple feature maps based on their scores.

3. Methodology

Our approach is demonstrated with the Iris dataset (Fig. 1) and applied to various other datasets, including Penguins [GWF14], California Housing [PVG*11], and the XOR problem.

We extend the NAM architecture by incorporating all possible feature combinations through pairwise feature submodules. Each submodule takes two features as input and produces a single output representing the combined positive or negative contribution of these features to a specific class.

For visualizing the feature maps, we employ line plots to represent main effects and heatmaps for pairwise feature interactions. We enhance the line plots by displaying class-wise data distributions. This is achieved by dividing the y-axis into l equal sections for l classes and representing histograms as heatmaps along the x-axis. The histograms use categorical colors to encode the classes and opacity for the counts.

Various scoring methods can be computed to filter and sort feature maps. One approach is to use permutation feature importance, but instead of applying it directly to the dataset, we apply it to individual submodules with permuted data. This prevents the permutation of feature i from affecting all interaction modules that use i as input. Another scoring function considers the effect of submodule outputs on the final sum of the NAM, which can be achieved by comparing the output ranges. Submodules with a range from -1 to $+1$ are likely to be less important for classification compared to those with outputs ranging from -30 to $+30$. Similarly, the variance in feature maps could be used to determine the scores.

We utilize one interactive bar chart per class to display the importance score of each feature. This allows users to select which features should be included in the final grid of visualizations. By brushing the bar plot or using a text input field, users can filter the sorted features to be displayed. The final visualization after brushing consists of a grid where each column represents one class. Within the columns, the feature maps are sorted by the score and potentially filtered based on the user's selection. This enables users to interactively explore the importance and relationships of features in a class-wise manner, enhancing interpretability and understanding of the model. It also allows them to discover any unwanted effects in these crucial feature maps, such as the model's inability to learn steep changes or capture complex relationships between features.

4. Limitations and Future Work

Our current implementation supports classification tasks only. It could be extended to handle regression problems where the same principles can be applied.

To improve the interpretability and utility of the visualizations, various plot types could be incorporated depending on categorical or continuous features (similar to GAM-Changer [WKN*21]).

Furthermore, our current implementation does not display the variance among multiple models trained with random initializations. This could be addressed by plotting multiple lines in the 1-D plot to visualize the variance of trained models as in the original NAM paper [AMF*21]. A related challenge is to develop methods of displaying variance among multiple models for the 2-D plots.

Acknowledgements

We gratefully acknowledge the financial support provided by the Boehringer Ingelheim Regional Center Vienna, and by the Austrian Science Fund through grant number FWF DFH 23–N.

References

[AMF*21] AGARWAL R., MELNICK L., FROSST N., ZHANG X., LENGERICH B., CARUANA R., HINTON G. E.: Neural additive models: Interpretable machine learning with neural nets. *Advances in Neural Information Processing Systems 34* (2021), 4699–4711. 1, 2

[CCG21] CHANG C.-H., CARUANA R., GOLDENBERG A.: Node-gam: Neural generalized additive model for interpretable deep learning. *arXiv preprint arXiv:2106.01613* (2021). 1, 2

[CLG*15] CARUANA R., LOU Y., GEHRKE J., KOCH P., STURM M., ELHADAD N.: Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining* (2015), pp. 1721–1730. 2

[GWF14] GORMAN K. B., WILLIAMS T. D., FRASER W. R.: Ecological sexual dimorphism and environmental variability within a community of antarctic penguins (genus pygoscelis). *PloS one 9*, 3 (2014), e90081. 2

[Has17] HASTIE T. J.: Generalized additive models. In *Statistical models in S*. Routledge, 2017, pp. 249–307. 2

[LCGH13] LOU Y., CARUANA R., GEHRKE J., HOOKER G.: Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (2013), pp. 623–631. 1, 2

[PMB19] POPOV S., MOROZOV S., BABENKO A.: Neural oblivious decision ensembles for deep learning on tabular data. *arXiv preprint arXiv:1909.06312* (2019). 2

[PVG*11] PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V., VANDERPLAS J., PASSOS A., COURNAPEAU D., BRUCHER M., PERROT M., DUCHESNAY E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830. 2

[WKN*21] WANG Z. J., KALE A., NORI H., STELLA P., NUNNALLY M., CHAU D. H., VORVOREANU M., VAUGHAN J. W., CARUANA R.: Gam changer: Editing generalized additive models with interactive visualization. *arXiv preprint arXiv:2112.03245* (2021). 2

[YZS21] YANG Z., ZHANG A., SUDJIANTO A.: Gami-net: An explainable neural network based on generalized additive models with structured interactions. *Pattern Recognition 120* (2021), 108192. 1, 2