INAM: Image-Scale Neural Additive Models

Jana Hüls, Jan-Ole Perschewski and Sebastian Stober [*]

Otto von Guericke University - Department of Computer Science
Universitätsplatz 2, 39106 Magdeburg - Germany

Abstract. Neural Additive Models are inherently interpretable models that can be applied to tabular data. However, when applying these models to images the value of a given pixel is not a meaningful feature for understanding the model. For that reason, we propose INAM - Image scale Neural Additive Model - a combination of trainable feature extractors and NAMs. We show INAMs can be successfully applied to image data sets with low variability while allowing global explanations of the models and data point-specific explanations.

1 Introduction

Generalized Additive Models are a well-established method to solve supervised machine learning tasks [1]. Here, we fit a separate function for each variable to predict the target. For that reason, the influence of each variable is easily accessible for interpreting the overall model. Recent developments led to the introduction of Neural Additive Models where each function of a Generalized Additive Model is a neural network [2]. Further approaches introduce feature interaction [3] or improve the overall performance [4]. However, a common limitation is that these models are only applicable to tabular data with meaningful variables. This is contrary to the current trend of preferring data-driven feature extraction.

In this paper, we propose to improve upon the Neural Additive Models by using a deep projection pursuit approach [5] to extract interpretable features that are then combined with Neural Additive Models. We call these models Image-scale Neural Additive Models (INAMs). We show how INAMs can be visualized to get insights into the overall model or how the decision for a given data sample is made. Moreover, we show the current limitations concerning the variability of the data set.

2 Method

INAMs extend Neural Additive Models to address image classification tasks (depicted in Figure 1). First, for an input image x we use a convolutional layer with N interpretable kernels $K^{(l)} \in \mathbb{R}^{n \times n \times c}$, with $l \in \{1, ..., N\}$, kernel size n and channels c, to extract feature map $F^{(l)}(x) = x * K^{(l)}$. The kernels in this layer correspond directly to detected features. Next, we apply global max-pooling $X_l = max(F^{(l)}(x))$ to get the maximal detection. Additionally,

[*] This research was funded by German Federal Ministry of Education and Research, grant number 16DHBKI008.

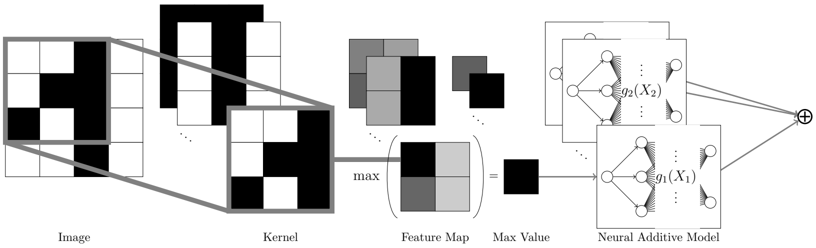

Fig. 1: Overview of INAM. We extract features with a single convolutional layer and global max pooling. The resulting feature activations are processed by a neural additive model.

this gives us a location corresponding to the feature, making data-dependent explanations easier. Afterwards, we apply a NAM to solve the classification task whilst allowing us to interpret the influence of the detected features. After training, we adapt the output of the NAM as follows.

$$\tilde{\mathbf{g}}_l(X_l) = \mathbf{g}_l(X_l) - \bar{\mathbf{g}}_l + \frac{1}{N}\sum_{i=1}^{N} \bar{\mathbf{g}}_i .$$

Here, $\mathbf{g}_l(X_l) : \mathbb{R} \to \mathbb{R}^K$ is the output after the training, and the calculated mean over the training samples D is given by $\bar{\mathbf{g}}_l = \frac{1}{|D|}\sum_{x \in D} \mathbf{g}_l(X_l(x))$. By adding the terms $-\bar{\mathbf{g}}_l + \frac{1}{N}\sum_{i=1}^{N} \bar{\mathbf{g}}_i$ we center the function and equalize the bias for all functions. This reparameterization simplifies the comparison of function values.

As mentioned, the convolutional layers should employ interpretable kernels , which does not need to be the case without adding constraints. Hence, we propose to employ Total Variation (TV) [6] to reduce noise within the kernels, which increases the visual interpretability. According to [7], the TV is given by the sum of the total variation per kernel. This means the TV of our model is

$$\Omega(\mathbf{K})_{TV} = \sum_{l=1}^{N} \sum_{m=1}^{c} \sum_{i,j=1}^{n-1} \left(\left(K^{(l)}_{i+1,j,m} - K^{(l)}_{i,j,m} \right)^2 + \left(K^{(l)}_{i,j+1,m} - K^{(l)}_{i,j,m} \right)^2 \right) . \tag{1}$$

Minimizing this term suppresses high-frequency information and makes coarse structures more apparent by making neighboring positions more similar.

The second important property of the kernels is their independence to avoid interaction between different features. For that, we propose regularizing the kernels to be orthonormal since differentiable orthonormalization would add a significant cost to the model. This means we minimize the distance to the unit matrix as:

$$\Omega(\mathbf{K})_{ortho} = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\sum_{h,w,c} K^{(i)}_{h,w,c} K^{(j)}_{h,w,c} - \delta_{i,j})^2 \tag{2}$$

where $\delta_{i,j} = 1$ if $i = j$ and $\delta_{i,j} = 0$ else. With this the overall loss is $L_{total} = L_{task}(INAM(x), y) + \alpha_{TV}\Omega(\mathbf{K})_{TV} + \alpha_{ortho}\Omega(\mathbf{K})_{ortho}$ With this definition of our model, we can show possible visualizations to understand what the model has learned.

2.1 Visualization

Global Interpretation Given a trained model, we want to understand what the model has learned. INAMs have the advantage of being inherently interpretable. One part of this is the choice only to use a single convolutional layer because we directly see what kind of features lead to higher activation as visible in Figure 2. In contrast, approaches such as classification-by-component[8] rely on a feature extractor that is an injection which is not guaranteed. Moreover, we can directly benefit from NAMs, which give us a visualization of the function depending on the strength of the feature. This means we can plot feature strength against influence on the class as visible on the right in Figure 2. Additionally, we include the density of the data distribution by shading the background, which shows how the corresponding feature is distributed. For example, Figure 2 presents a kernel detecting an arch shape. As indicated by the corresponding function, a high max pooling value for this kernel suggests a prediction of the digit "2" or "3", aligning with the discernible pattern evident in the kernel image.

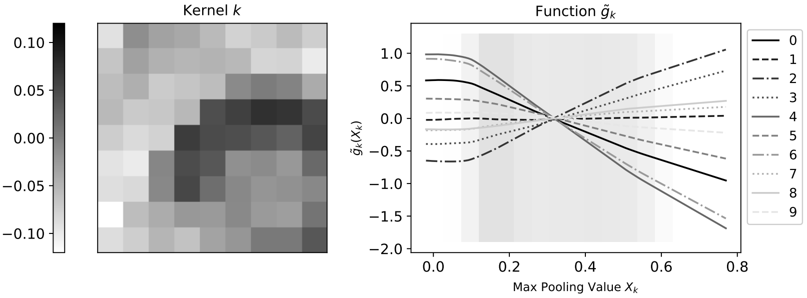

Fig. 2: Kernel with Corresponding Function on MNIST.

Local Interpretation In addition to making the model overall interpretable, we aim to explain the decision for a certain class for the input image using a decision image. Here, the idea is to determine all kernels $K^{(l)}$ for a class t such that $\tilde{g}_l(X_l)_t > \tilde{g}_l(X_l)_k \forall t \neq k$. These kernels are then weighted by their function value and added to the position corresponding to X_l.

Figure 3(center) illustrates the decision image of the correctly classified input image with the label "two" (left). Notably, the lower-left part of the digit and the arch at the top are highlighted as crucial components. Analogously, we can analyze which parts in the image are taken as evidence for another class in

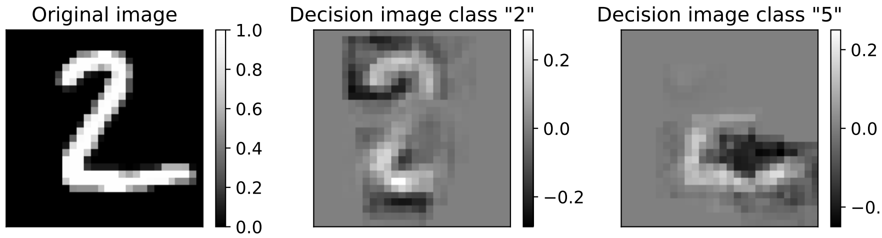

Fig. 3: Visualization of Decision Maps with respect to an image (left) for INAM predicting the class "two" correctly with respect to the class "two"(center) and the class "five"(right).

Figure 3(right). Here, we plot the decision image with respect to the class label "five". Even though the model gives the class "two" a probability of 99.99%, there is still evidence for other classes due to common patterns between classes. In the figure, we see the model could have identified the upper part of the class "five" near the bottom of the image.

Further, we want to evaluate whether this visualization represents the predicted class. For that, we only keep the positive parts of the visualization and scale to $[0, 1]$. Focusing on the positive part can introduce some errors since it ignores negative reasoning. However, in the other case, we leave the data distribution since the original input data is also in the range $[0, 1]$. Afterwards, we check if the model predicts the same class for the decision image. Analyzing the complete validation data set shows that 81.98% are assigned to the same class. This means that a positive explanation alone does not imply the same class. However, in many cases, we can assume that the explanation is characteristic of the predicted class.

3 Performance

In this section, we evaluate INAMs on three image data sets: MNIST [9], CIFAR [10] and a histological colorectal cancer data set (CRC) [11]. MNIST helps to show how the interpretation works since it is easy to understand. CIFAR and CRC show the applicability to different color data sets. CIFAR has a large variety within and between classes, whereas the CRC data set is more simplistic due to the similar appearance of tumors in tissue slides.

The general setup of INAM follows the NAM architecture [12] with three linear layers with 64, 64, and 32 units utilizing the ReLU activation and an output layer with as many units as classes in the task. We train each model using the Adam optimizer with default parameters [13] for 10 epochs on MNIST and 20 epochs on CIFAR and CRC. Furthermore, we search for good hyperparameters employing a grid search over the number of kernels N, the kernel size n, the strength of the orthogonal regularization (α_{ortho}) and the TV regularization

(α_{TV}).

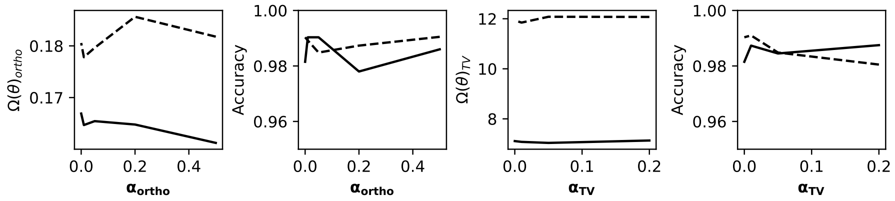

Fig. 4: Influence and training accuracy with respect to regularization terms for kernel size 9 (-) and kernel size 13 (- -) for models trained on MNIST. We assume the weights to be zero unless otherwise suggested by the figure.

First, we investigate the influence of the choice of our weighting terms on the model performance with different kernel sizes. The results of this can be seen in Figure 4. Here, we see that the orthogonality loss only slightly increases the orthogonality of the kernel. Furthermore, the loss leads to a slight decrease in accuracy, which is less if the kernel size is larger. The TV loss barely changes with the weighting term. Interestingly, a small value for α_{TV} leads to a better performance than $\alpha_{TV} = 0$.

In addition to the evaluation of the weights, we also need to consider the performance in comparison to similar models shown in Table 1. Here, we can make two observations. First, if the patterns in the data set have a low variability, INAMs perform on par with comparable models, as seen in the performance on MNIST and CRC, where all models are simplistic and interpretable. Second, when the variability of patterns in the data set increases, the performance is significantly worse than in performance-focused models, as evident by the performance on CIFAR-10.

Data Set	Method	$(\mathbf{N}, \mathbf{n}, \boldsymbol{\alpha}_{\mathbf{ortho}}, \boldsymbol{\alpha}_{\mathbf{TV}})$	Accuracy
MNIST	INAM	$(256, 13, 0.05, 0.05)$	**0.9920**
	Linear Model	-	0.9385
CIFAR-10	ResNet110 [14]	-	**0.9357**
	INAM	$(256, 15, 0.5, 0.01)$	0.6018
CRC[11]	RBF [11]	-	**0.8740**
	INAM	$(256, 15, 0.5, 0.01)$	0.8725

Table 1: Performance of INAM in comparison to existing approaches. INAMs do not scale to data sets with a large variability of patterns.

4 Conclusion

In this paper, we propose INAMs that scale NAMs to image-scale while inheriting their interpretability. We focus on image classification and elaborate a methodology that transforms the input images using convolution and max pooling. Furthermore, we introduce visualization for explaining INAMs as a whole or specifically for a single sample. Finally, we show that INAMs perform well as long as the variability of patterns is not too large. This means INAMs are a promising approach for explainable image classification.

References

[1] Trevor Hastie and Robert Tibshirani. Generalized Additive Models. *Statistical Science*, 1(3):297 – 310, 1986.

[2] Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. In *NeurIPS*, volume 34, pages 4699–4711. Curran Associates, Inc., 2021.

[3] Zebin Yang, Aijun Zhang, and Agus Sudjianto. Gami-net: An explainable neural network based on generalized additive models with structured interactions. *Pattern Recognition*, 120:108192, 2021.

[4] Filip Radenovic, Abhimanyu Dubey, and Dhruv Mahajan. Neural basis models for interpretability. In *NeurIPS*, volume 35, pages 8414–8426. Curran Associates, Inc., 2022.

[5] Jan-Ole Perschewski, Johann Schmidt, and Sebastian Stober. Pursuing the perfect projection: A projection pursuit framework for deep learning. In *Advances in Self-Organizing Maps, Learning Vector Quantization, Interpretable Machine Learning, and Beyond*, pages 43–52, Cham, 2024. Springer Nature Switzerland.

[6] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, November 1992.

[7] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[8] Sascha Saralajew, Lars Holdijk, Maike Rees, Ebubekir Asan, and Thomas Villmann. Classification-by-components: Probabilistic modeling of reasoning over a set of components. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[9] Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *ATT Labs [Online]*, 2, 2010.

[10] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[11] Jakob Nikolas Kather, Cleo-Aron Weis, Francesco Bianconi, Susanne M. Melchers, Lothar R. Schad, Timo Gaiser, Alexander Marx, and Frank Gerrit Zöllner. Multi-class texture analysis in colorectal cancer histology. *Scientific Reports*, 6(1):27988, June 2016.

[12] Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. In *Advances in Neural Information Processing Systems*, volume 34, pages 4699–4711. Curran Associates, Inc., 2021.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980v9 [cs.LG]*, January 2017.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.