

HIERARCHICAL VISUALIZATION OF DECISIONS USING NEURAL-BACKED DECISION TREES

Siddhant Agarwal, Aadarsh Sahoo, Adarsh Patnaik, Rajat Kumar Jenamani

Indian Institute of Technology Kharagpur

1. ABSTRACT

Deep Learning models involving Convolutional Neural Networks enjoy stellar success in solving the classic problem of image classification. Despite being highly accurate in predicting class labels, many models fail to explain the underlying decisions taken to reach the final prediction. Most methods based on saliency maps explain only the final decision made by the model. In this work, we analyse the intermediate decisions involved in the task of classification by using saliency maps. This analysis work is based on the prior work of NBDT[1]. We report our analysis on two publicly available basic datasets of CIFAR-10 and TinyImageNet and use RISE[2] for saliency map generation.

2. INTRODUCTION

The field of artificial intelligence has seen staggering progress recently with thousands of publications each year that guarantee newer and more astonishing results. Deep learning has provided state of the art results in various fields like image classification, object detection, image segmentation and domain adaptation. With increase in computation power and heavy use of parallel computing, these models have become excessively complex. Current research on deep learning considers such models as a black box and tries to improve their performances even further. These works are relevant to some fields, but in safety critical domains like autonomous driving and medical applications, the model not only needs to be highly accurate, but also explainable for its mistakes. This has brought the field of explainable AI into limelight.

Explainable AI has seen considerable progress in the recent years. The role of explainable AI can be divided into three stages. First, to understand the data in order to explain what a model should learn from the data. Second, to develop models that are explainable and performing. Third, to be able to explain previously developed black box models. The current focus in this field is highly inclined towards the third stage due to the excessively complex and accurate black box models available. The major methods used for this purpose include saliency maps and decision tree based methods. Saliency map based methods [3] [2] are input based methods that can find the visual attention of the model in the input im-

age. Thus, this method can explain the exact features that a model focuses on. However, it cannot explain the decision process for the model. Decision Tree based methods take intermediate decisions to reach the final result which helps us understand how a model reaches the final decision. However, such methods lag excessively in terms of model accuracy in comparison to more complex neural networks.

Recent works [4] [5] combine the robustness of neural networks and the interpretability of decision trees to create highly accurate explainable models. One such method, Neural Backed Decision Trees (NBDT) [6], can convert any neural network to a NBDT without any architectural changes to the network backbone. NBDTs take a set of intermediate decisions to narrow down on the prediction class. These intermediate decisions can reveal perceptually informative hierarchical structure in the underlying target classes. Fine tuned on a custom loss, NBDTs also achieve neural network accuracy. However, the work does not explore the regions of the image the model focuses on while taking intermediate decisions and how this focus shifts as we progress down the decision tree. For example, a model might focus on the wheels to take an intermediate decision of "vehicle" and then focus on the overall structure of the vehicle to decide between "car" or "truck".

In this work we propose a method to incorporate the advantages of decision tree based explanations with saliency map methods in order to make complex models more explainable and remove bias from such models. We use state of the art methods for generating saliency maps in combination with neural network based decision trees in order to find hierarchical differences in the visual attention of a neural network model. This helps us to visualize the model attention at different levels of a decision tree. The use of such an approach is to learn how a model behaves with unseen data. The decision trees have embedded decisions that explain the final result under a hierarchical process. Under unseen data, the final result can be wrong, but intermediate decisions can be accurate and studying the attention of the model at these intermediate steps can help to remove the problems of bias in highly complex models.

Our contributions in this work are summarized below:

1. We propose to incorporate NBDTs and saliency map based explanations to monitor the subsequent vari-

ations in the visual attention of the model between different layers of the decision tree.

2. We generate a combined saliency map by combining all decisions taken by the model and compare it with direct maps generated by RISE[2].

3. RELATED WORKS

Explainable AI has seen a recent explosion in the interest of researchers and multiple techniques that try to explain various models have been developed. Methods based on information theory [7] [8] minimize the mutual information between the model decision and the generated explanations. Activation based techniques [9] [10] make white-box models interpretable by using activation values across convolution layers. However, these techniques cannot be generalised over all model architectures. On the other hand, LIME [11] is able to explain black box models by drawing random samples around each instance to be explained and fitting an approximate local linear decision model in the vicinity of the input. However, LIME's dependence on superpixels leads to inferior saliency maps and thus it fails on complex non-linear classifiers. Back-propagation based methods [12] [13] generate the important measure of a pixel by backpropagating the output of a deep neural network back to the input space using gradients or their variants. Perturbation based methods [14] perturb the inputs (blur some pixels, or add some noise etc) and measure the response of the model to these. RISE [2] is one such method which passes an image covered with random masks through a model and generates a saliency map for the same by taking a weighted sum of these masks where the weights are the class scores predicted by the model after the mask is applied.

Decision Trees [15] use a tree like model where each node is a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. The predicted class depends on outcome of the intermediate tests thus making this model interpretable. Therefore, recent works are combining neural networks with decision trees, to create highly accurate explainable models. One such work, Deep Neural Decision Forests [1] is able to match neural networks on ImageNet by unifying decision trees with the representation learning functionality known from deep convolutional networks and training them in an end-to-end manner. However, interpretability is forgone due to the use of impure leaves and a forest. Deep Decision Network (DDN) [5] substitutes each node of a decision tree by a neural network. It is not only able to match neural network performance, but also provides some insight into the data it is trained on by identifying the most confusing classes and their performances.

Our work is based on Neural Backed Decision Trees [6] which also incorporates the accuracy of neural networks with interpretability of decision trees. It uses only the weights of the neural network's fully connected layer to find a hierarchy

that can be seen as nodes of a decision tree. We propose to increase the explainability of NBDT by applying state of the art approaches for generating saliency maps for black box models [2] to visualize the model attention at different levels of a decision tree.

4. APPROACH

In this section we discuss our approach to generate hierarchical visualization of deep neural network predictions. Given a model, we first create a neural backed decision tree using the model. Then we generate explanations for each of the decision taken. We first discuss the method of developing induced hierarchy of a decision tree from the neural network. Then we explain the saliency map generation principles used. Finally we propose a hybrid method to incorporate both of these together.

4.1. Developing the Induced Hierarchy

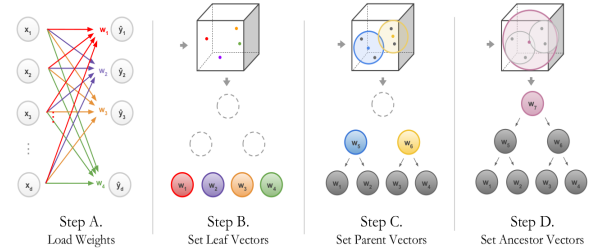


Fig. 1. Developing Induced Hierarchy from a Neural Network. (Image taken from [6])

We use the recent work on Neural Backed Decision Trees[6] that are able to convert any Neural Network architecture into a decision layer based model using the pretrained final fully connected layer weights. The output of any Neural Network are based on finding the class with the highest inner product value with the final layer. If we consider the final fully connected layer weights as the leaf nodes of a decision tree, then the decision making procedure becomes similar to a neural network. We use this to develop the nodes in the decision tree with each node associated with a weight vector. As shown in Fig. 1 starting from the leaf node, each leaf node represents a row of the weights of the fully connected layer. To get the parent nodes, we perform agglomerative clustering among the child weights and take the mean of the closest weight vectors and assign that to their parent node. In this way we induce a decision tree hierarchy for the neural network.

4.2. Training the model using the decision tree

After generating the induced hierarchy, we can train the neural network with a modified tree supervision loss which takes into consideration the intermediate decisions in the decision tree.

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N \text{CrossEntropy}(\mathcal{D}(i)_{pred}, \mathcal{D}(i)_{label}) \quad (1)$$

Using the Tree Supervision Loss in 1, we fine tune the neural network to consider the intermediate decisions as well by taking the cross entropy loss over all child nodes for a node in the decision tree. Given a node i having c children, this is a c way cross entropy loss between predicted probabilities $\mathcal{D}(i)_{pred}$ and labels $\mathcal{D}(i)_{labels}$.

The work in [6] discusses another variation of this loss called the Soft Tree Supervision Loss which takes into account soft decisions based on probability distribution over each leaf and takes the argmax of the product of probabilities along each subtree to find the best decision. Soft decision tree hence provides a distribution over the leaves \mathcal{D}_{pred} . This loss can be obtained by adding a soft supervision loss term to the original cross entropy loss term such that $\mathcal{L} = \mathcal{L}_{Original} + \mathcal{L}_{Soft}$ where

$$\mathcal{L}_{soft} = \text{CrossEntropy}(\mathcal{D}_{pred}, \mathcal{D}_{label}) \quad (2)$$

4.3. Saliency Map Generation

We follow the techniques used by [2] to generate Saliency Maps for each decision. Each node n in the graph can have k children representing k decisions that can be taken at that node. Each decision is in fact conditioned by the event that the previous decision was taken. Hence if at from a node n , we choose to go to node m , which is the child of node n , we can do so with a probability $p(m|n, I)$. We represent this decision as d_{mn} and $p(m|n, I)$ as $p_{d_{mn}}$.

We generate Saliency Maps for each possible decision. Hence, if the tree consists of $(n + c + 1)$ nodes where, n is the number of internal nodes(except the root) and c are the number of leaf nodes i.e. classes, we generate $(n+c)$ saliency maps.

4.3.1. Saliency Map Generation for a Decision

We follow the same technique followed by [2]. The first step is the creation of N random masks of size $H \times W$, where H and W are height and width of the image respectively. To do that, we sample N random binary masks of size $h \times w$ where h and w are smaller than H and W respectively and each element can be 1 with a probability p . We then upsample these masks using bilinear interpolation to get them to size $H \times W$. We then shift these masks by a random number of

pixels in both the spacial directions. The generated masks are hence not binary and do not create any edges when applied on the Image.

We next apply these masks on the Image I to get $I'_i = I \odot M_i$ where M_i is the i^{th} mask and I'_i is the i^{th} masked image generated. Let the probability of a decision d_{mn} , for an image I be $p_d(I)$. Next we compute the saliency map as follows,

$$S' = \sum_{i \in N} p_{d_{mn}}(I'_i) \cdot M_i \quad (3)$$

We then normalise S' with the expected value of a pixel of the image remaining unmasked which is $N \cdot p$. So, $S = \frac{1}{N \cdot p} \cdot S'$.

Each pixel in the saliency map, i.e $S(x, y)$ corresponds to an importance score, defined as the expected value of the probability of taking the decision given that the corresponding pixel is unmasked over all the masks.

4.3.2. Hierarchical Saliency Map Generation

As already mentioned, we generate the saliency maps for every decision. The final saliency map is generated by combining the saliency maps for all the relevant decisions. We generate a path that will be followed by the root to the leaf node that predicts the class. We combine only those saliency maps that fall in this path. For example, for class cat, the path followed is "whole \rightarrow animal \rightarrow chordate \rightarrow carnivore \rightarrow cat". Each \rightarrow represents a decision and has a corresponding saliency map. As already mentioned, each decision to go from node n to m can be characterised by a conditional probability. So the decision made will be characterised by probabilities $p(\text{animal} | I)$, $p(\text{chordate} | \text{animal}, I)$, $p(\text{carnivore} | \text{chordate}, I)$ and $p(\text{cat} | \text{carnivore}, I)$

Final saliency map is given by,

$$S = \frac{1}{|P|} \cdot \sum_{d_i \in P} w_i \cdot S_{d_i} \quad (4)$$

where, P is the Path, d_i is a decision made in the path at the i^{th} , w_i is the weight of the i^{th} step in the path and $|P|$ is the length of the path or number of decisions made.

5. RESULTS

We perform experiments on two publicly available basic datasets: CIFAR-10[16] and Tiny-ImageNet[17]. CIFAR10 consists of images of dimension 32×32 belonging to 10 different classes. Tiny-ImageNet has images of dimension 64×64 belonging to 200 classes. We use Wide-ResNet28x10[18] as the back bone model for CIFAR-10 and ResNet18[19] for Tiny-ImageNet.

The induced hierarchy generated for CIFAR-10 is shown in Figure 2. For a sample prediction of "deer", the path of

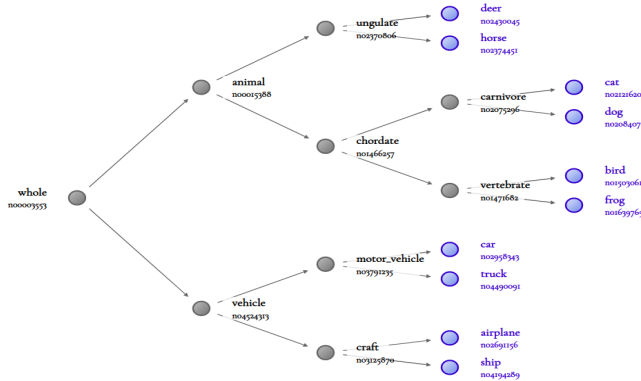


Fig. 2. The underlying decision graph(tree) using induced hierarchy for WideResNet28x10 architecture involved in the classification of images from CIFAR-10 dataset. There are in total 8 intermediate nodes and 10 leaf nodes which correspond to the number of classes.

correct classification will be "whole" \rightarrow "animal" \rightarrow "ungulate" \rightarrow "deer" and the conditional probabilities involved will be $p(\text{animal} \mid I)$, $p(\text{ungulate} \mid \text{animal}, I)$ and $p(\text{deer} \mid \text{ungulate}, I)$. It must also be noted that since there exists unique path from the state "whole" to any other node, the probability of a node conditioned on the entire chain is same as the probability conditioned on the parent. That means, $p(\text{carnivore} \mid \text{chordate}, I) = p(\text{carnivore} \mid \text{chordate}, \text{animal}, I)$. Also, $p(\text{cat} \mid I) = p(\text{cat} \mid \text{carnivore}, I) \cdot p(\text{carnivore} \mid \text{chordate}, I) \cdot p(\text{chordate} \mid \text{animal}, I) \cdot p(\text{animal} \mid I)$.

We obtain the saliency maps for all the decisions i.e. for all the conditional probabilities involved. In table 1 we show the explanations for decisions in the prediction path for images sampled from the CIFAR-10 dataset.

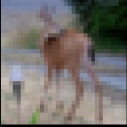
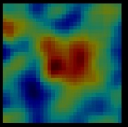
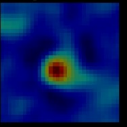
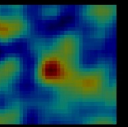

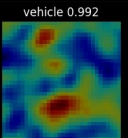
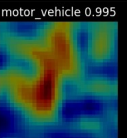
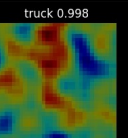
Image	Intermediate Explanations		
	animal 0.993 	ungulate 0.995 	deer 0.992 
	vehicle 0.992 	motor_vehicle 0.995 	truck 0.998 

Table 1. Explanations obtained from two sample images of the CIFAR-10 dataset. The images were predicted in a 3-step path i.e. 3 decisions were taken to predict the final label. The left-most column (col-1) shows the original image, col-2-3 show the RISE heatmaps for each of the intermediate decision nodes. The right-most column (col-4) shows the RISE heatmap for the leaf node.

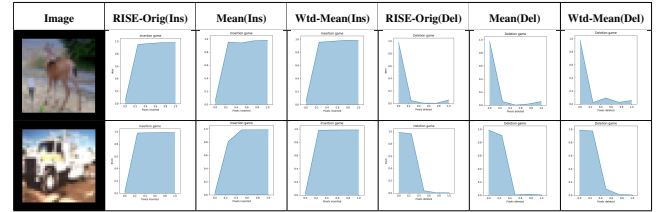


Table 2. Insertion and Deletion plots for an image from the CIFAR-10 dataset. The col-2-4 show the plots corresponding to RISE-Original, Mean heatmap and Weighted-Mean heatmap respectively.

Dataset	Saliency Map	Avg-Ins-Score	Avg-Del-Score
CIFAR-10	Rise-Orig	0.72098	0.33828
	Mean-Map	0.64969	0.36107
	Wt-Mean-Map	0.63565	0.34126
TinyImageNet	Rise-Orig	0.42426	0.11706
	Mean-Map	0.30726	0.17007
	Wt-Mean-Map	0.33740	0.12356

Table 3. The average insertion and deletion scores for the three saliency maps. The average results are computed using 100 different images for each of the datasets.

We perform our experiments with two setups. One with $w_i = 1, \forall i$ and another with $w_i = p_{d_i}$ i.e. the weight is equal to the probability of taking that decision. In the second setup, we give an intermediate-map importance based on the confidence of taking that decision. Table 4 compares the saliency maps generated by the two setups as well as with the saliency maps generated from using direct forwarding the images through the SoftNBDT Model.

We evaluate all the Saliency maps generated using the Causal Metrics proposed by RISE[2]. It consists of two metrics, namely insertion score and deletion score. While calculating the insertion score, we keep inserting the regions on an initially gaussian blurred image from the corresponding original image in the decreasing order of importance predicted by heatmap. After each insertion we predict the class score using the SoftNBDT model and plot it with respect to the amount of image uncovered. The Area under the curve gives the insertion score. Deletion score works in the similar fashion but instead of inserting, we remove the regions from the original image in decreasing order of importance. Again we plot the class scores and the area under the curve provides the deletion score.

In Table 3 we present the average insertion and deletion scores of the saliency maps generated from both of our experiments as well as for the saliency maps generated by applying RISE directly to the SoftNBDT models.

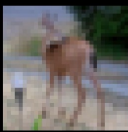
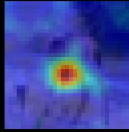
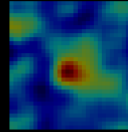
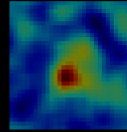

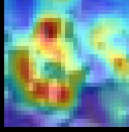
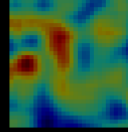
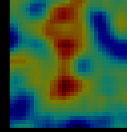
Image	RISE-Orig	Mean-Map	Wtd-Mean-Map
			
			

Table 4. Explanations obtained from two sample images of the CIFAR-10 dataset. The left-most column (col-1) shows the original image, col-2 shows the RISE heatmap for the original image when forwarded through the SoftNBDT model for direct probabilities. col-3 and col-4 show the mean and weighted-mean heatmaps of the intermediate explanations shown in table 1.

6. REFERENCES

- [1] P. Kotschieder, M. Fiterau, A. Criminisi, and S. R. Bulò, “Deep neural decision forests,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1467–1475.
- [2] Vitali Petsiuk, Abir Das, and Kate Saenko, “Rise: Randomized input sampling for explanation of black-box models,” in *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.
- [3] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra, “Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization,” *CoRR*, vol. abs/1610.02391, 2016.
- [4] Peter Kotschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulò, “Deep neural decision forests,” in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, USA, 2015, ICCV ’15, p. 1467–1475, IEEE Computer Society.
- [5] V. N. Murthy, V. Singh, T. Chen, R. Manmatha, and D. Comaniciu, “Deep decision network for multi-class image classification,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2240–2248.
- [6] Alvin Wan, Lisa Dunlap, Daniel Ho, Jihan Yin, Scott Lee, Henry Jin, Suzanne Petryk, Sarah Adel Bargal, and Joseph E. Gonzalez, “Nbd: Neural-backed decision trees,” 2020.
- [7] Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan, “Learning to explain: An information-theoretic perspective on model interpretation,” 2018.
- [8] Seojin Bang, Pengtao Xie, Heewook Lee, Wei Wu, and Eric Xing, “Explaining a black-box using deep variational information bottleneck approach,” 2019.
- [9] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba, “Learning deep features for discriminative localization,” 2015.
- [10] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, Oct 2019.
- [11] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin, ““why should I trust you?”: Explaining the predictions of any classifier,” *CoRR*, vol. abs/1602.04938, 2016.
- [12] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” 2013.
- [13] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller, “How to explain individual classification decisions,” *J. Mach. Learn. Res.*, vol. 11, pp. 1803–1831, Aug. 2010.
- [14] Jörg Wagner, Jan Mathias Köhler, Tobias Gindele, Leon Hetzel, Jakob Thaddäus Wiedemer, and Sven Behnke, “Interpretable and fine-grained visual explanations for convolutional neural networks,” 2019.
- [15] JR QUINLAN, “Induction of decision trees,” *Machine Learning*, vol. 1, pp. 81–106, 1986.
- [16] Alex Krizhevsky, “Learning multiple layers of features from tiny images,” 2009.
- [17] Ya Le and Xuan Yang, “Tiny imagenet visual recognition challenge,” 2015.
- [18] Sergey Zagoruyko and Nikos Komodakis, “Wide residual networks,” 2016.
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” 2015.