УДК 004.8:004.93
DOI: 10.18523/2617-3808.2025.8.57-67

M. Mokryi, N. Shvai

ROBUSTNESS OF NEURAL DECISION TREES TO NOISE IN INPUT DATA FOR IMAGE CLASSIFICATION TASKS

Neural networks, particularly convolutional neural networks (CNNs), have demonstrated high effectiveness in image classification tasks. However, they are known to be vulnerable to input data perturbations and have weak interpretability due to their black-box nature. In contrast, traditional decision trees (DTs) provide transparent decision-making processes, but are limited to low-dimensional or tabular data, restricting their field of application in computer vision tasks such as image classification. To address this gap, a hybrid architecture known as Neural Decision Trees (NDTs) has emerged, combining strong generalization and learning capabilities of neural networks, with transparent hierarchical inference and interpretability of DTs.

The article investigates the robustness of NDTs to noise in input data for image classification tasks. Despite the extensive studies covering the robustness of both CNNs and traditional DTs against various forms of input perturbations, the robustness of NDT models remains a largely underexplored area. This study provides two robust training methods to improve robustness: constant noise learning and incremental noise learning, originally developed for CNNs, but which can be effectively applied to NDT-based architectures and significantly improve the robustness to noisy images for models. These methods involve adding perturbed samples via a Gaussian blur during the training stage. The noisy test set consists of images perturbed by a Gaussian blur and is used to evaluate the robustness performance.

A series of experiments were conducted on the CIFAR-10 dataset using the original training baseline and robust training methods. The results demonstrate that constant and incremental noise learning significantly improve the robustness of all tested NDT models to noisy images compared to their original training performance. While the ResNet18 baseline model demonstrates higher overall performance, the NDT models show comparable robustness improvements using the proposed robust training strategies. Constant noise learning offered an adjustable trade-off between performance on clean and noisy images, while incremental noise learning provided a more stable training process. The first method is considered preferable due to the simplicity of implementation.

This study empirically confirms that NDT models can effectively use methods adapted from CNNs to improve their robustness against perturbations in input data. An NDT framework was developed to conduct training and validation using a standardized shared pipeline. It is available via the link: github.com/MikhailoMokryy/NDTFramework.

Keywords: Neural Decision Trees, machine learning, robustness, image perturbations, image classification, computer vision, convolutional neural networks.

Introduction

The field of computer vision, particularly image classification tasks, has advanced significantly with the introduction of convolutional neural networks (CNNs), which demonstrate remarkable generalization abilities for high-dimensional input data. However, they are not ideal, suffer from a lack of interpretability due to their black-box architecture, and are vulnerable to perturbations in input data [3, 7, 9, 14]. Such images can create misclassifications by a neural network (NN), which leads to poor performance. The robustness of traditional decision trees (DTs) to noise has been studied only for low-dimensional or tabular data due to their limited field of application. To extend this field, a hybrid architecture referred to as Neural Decision Trees (NDTs) has been proposed [1, 4, 6, 8, 11, 13]. The NDT model combines two distinct architectures – CNN and DTs, resulting in a tree-based architecture capable of solving image classification tasks by combining a strong generalization ability acquired from CNNs with the interpretability of a DT hierarchy. While the robustness of CNNs and DTs to noise has been extensively studied, NDT models remain largely under-

explored. This study aims to investigate methods that can be incorporated to improve the robustness of NDTs.

Previous efforts to solve the challenge of improving the robustness of NNs, including defensive strategies, have drawbacks and do not cover all possible input perturbation forms. Robust image classification, focused on real-world alterations such as hardware defects or environmental distortions, uses different training strategies, including constant noise learning, incremental noise learning, and architecture modification [14]. Robust training methods for traditional DT models are also proposed, but they primarily target low-dimensional or tabular data and do not cover the field of image classification tasks [2, 12, 16]. The robustness of existing NDT models to input data noise is largely unexplored. Given the limited understanding of NDT robustness to noisy images, this study investigates whether learning strategies developed to improve the robustness of CNN models can be effectively applied to NDT-based architectures and demonstrate comparable robustness improvement.

To validate the proposed approach, a series of experiments were conducted on the CIFAR-10 dataset, which is a widely used benchmark for image classification tasks. Image perturbations are created by applying a Gaussian blur to input samples. These perturbed images are used both to evaluate model robustness in experiments and as a part of the robust training process. The experiments involve evaluating the robustness of various models using the original training baseline, constant noise learning, and incremental noise learning strategies. The constant noise learning method incorporates a fixed proportion of perturbed images during the training phase, with noise applied at varying probabilities: 0.05, 0.1, 0.2, 0.4, 0.6, 0.8. In contrast, the incremental noise learning approach gradually increases the proportion of noisy images during the training phase, starting from a low initial level, eventually almost reaching the size of the original training set in the final stage of training. In total, 40 models were trained under various conditions to evaluate their performance on clean and noisy test sets. The trained models include NDT models: Deep Neural Decision Tree, Deep Neural Decision Forest, and Neural Backed Decision Trees variations with hard and soft inferences, as well as the baseline ResNet18 model for comparison.

Main contributions of this study:

1. A comprehensive investigation of the robustness of NDT models for image classification tasks, focusing on the impact of perturbed images on these models' performance.
2. Demonstration that methods to improve robustness, originally developed for CNNs, specifically constant and incremental noise learning, can be successfully applied and significantly improve the robustness of NDT models against input perturbations.
3. Empirically determined these findings through a series of experiments under various conditions on the CIFAR-10 dataset, providing detailed performance metrics including model accuracy on clean and noisy test sets, accuracy drop, and robustness gain.

The rest of the paper is organized as follows: the Related Work section provides a general literature overview of NDT models and studies about improving robustness to noisy input samples for CNNs, traditional DTs, and the current state of NDT research. The Methodology section describes the NDT architectures and strategies to improve robustness. The Experiments section outlines the CIFAR-10 dataset, the used models, including their hyperparameters, and the overall training workflow. In the Results and Discussion section, results of experiments are presented in tabular and graphical form, with a detailed comparison between models and methods to improve robustness. Finally, the Conclusion section provides a comprehensive summary of this study.

Related Work

Early attempts to combine neural networks and decision trees, known as Hierarchical Mixtures of Experts (HMoE), were introduced by Jordan and Jacobs [10]. The HMoE architecture has a routing function: a linear classifier in each tree node, which decides where to send an input sample: to the left or right branch, passing it down a fixed tree structure. A more advanced, Soft Decision Trees (SDT), the base part of many NDT architectures, which is a fuzzy DT used for classification and regression tasks, appeared in Suarez and Lutsko [15] work. It is built with consideration of data partial membership in the tree nodes that form the tree structure. It was a key part in the future development of NDTs, allowing the use of the back-propagation method for training models. The next significant improvement of NDT was made by Kontschieder et al. [4] by enhancing soft DTs with an updated routing function in each node that contains a neural linear layer and sigmoid activation function. Their neural decision forest model (dNDF) is an ensemble of DTs in which the

whole CNN architecture, excluding the final linear layer, is used as the root transformer, which extracts features and passes them to tree-structured classifiers. Another vision of NDTs development presented tree-like structures of NNs with a routing mechanism. Ioannou et al. [6] introduced a Conditional Network model that reduces the computational load and the number of CNN architecture parameters by distributing computations through a hierarchical structure: a directed acyclic graph. This model uses an MLP-based route and achieves the same level of accuracy on image classification tasks with lower computational cost. Frosst et al. [8] utilized a NN to extract knowledge from it and use it in the SDT model training process. Thus, a NN provides a more informative soft target for training. The soft DT has learning filters to make hierarchical decisions, based on input targets in every internal node and a static probability distribution over the output classes for every leaf node. Tanno et al. [1] proposed adding adaptive architecture growth support to NDT, a feature of DTs. The Adaptive Neural Trees (ANT) model is constructed using a greedy algorithm that selects the best option between increasing the tree's depth and partitioning the input space before the model training phase. Unlike previous works, the Neural-Backed Decision Trees (NBDT) model introduced by Wan et al. [11] addresses the limitation of NDT models: the trade-off between accuracy and interpretability. It employs a WordNet lexical database to assign a selected concept to each tree node to improve model interpretability by labeling. NBDTs replace the final linear layer of the NN with a differentiable sequence of decisions and uses a hierarchical tree loss for model training. A novel approach to learning trees from deep NN (DNN) architecture, called Self-born Wiring (SeBoW), was proposed by Chen et al. [13]. It extended the ANT architecture growth ideas by using self-born neural trees that evolve themselves from a user-designed mother DNN architecture, instead of growing trees progressively or by using greedy algorithms. This self-born learning procedure allows for global-level tree-architecture parameter optimization over the neural tree search.

The generation of perturbed images, designed to cause a misclassification in a NN, has been extensively studied. These images are commonly referred to as adversarial examples. The early foundational work on adversarial example misclassification was covered by Goodfellow et al. [9]. Their work revealed that the linear nature of NNs is one of the main reasons why NNs are prone to adversarial examples. Input data is made by adding a selected adversarial perturbation to an original sample. Subsequent research by Papernot et al. [7] introduced a defense mechanism called defensive distillation to reduce the negative impact of adversarial examples on image classification, highlighting the fundamental nature of NN vulnerabilities. Carlini and Wagner [3] demonstrated that defensive distillation has some drawbacks and does not cover all adversarial examples, and created a set of attacks that can be used to improve the robustness of NNs. Stock et al. [14] proposed comprehensive strategies for robust image classification, examining defensive training techniques and architecture modifications that improve robustness to noisy input data for NN models. Their study focuses on real-world alterations that occur when an image is captured and can be caused by hardware defects or environmental distortions. Different techniques were used for altering input data to create digital augmentations, such as single-pixel modification, noise, and blur.

Research on adversarial examples and model robustness has been extensively conducted for linear models and NNs. However, the impact of adversarial examples on tree-based model robustness remains poorly studied. Unlike NNs, tree-based models are not differentiable, have a hierarchical structure, and are interpretable due to their nature, which can lead to the assumption that they are more robust than CNNs. However, Chen et al. [12] show that tree-based models can also struggle against adversarial examples. They investigate the robustness of tree-based models and the impact of adversarial examples on both classical DTs and more advanced ensemble boosting methods. A novel robust DT training framework based on a robust splitting function was proposed to improve robustness. Further ideas were presented by Andriushchenko and Hein [2]. This paper identified a drawback of the previously proposed method: a lack of robustness guarantee. The authors proposed robust training methods that achieve a provable robustness for boosted trees, and the results are comparable with CNN-based methods. Vos and Verwer [16] proposed a novel method for training robust DTs called growing robust trees, or GROOT for short, which adds a parameter to control a trade-off between model accuracy and robustness against adversarial examples.

All existing tree-based methods that improve robustness focus primarily on low-dimensional and tabular data. NDT robustness against adversarial examples remains underexplored. To address this gap, this study aims to investigate the impact of images perturbed by noise on the robustness of NDT models and determine whether robust training methods originally developed for CNN models can be effectively applied to NDTs, demonstrating improvements in robustness.

Methodology

DTs are a tree-structured machine learning method with internal decision nodes and prediction nodes, known as leaf nodes, used for low-dimensional or tabular data for classification or regression tasks. In contrast, NNs, including CNNs, are powerful models with strong generalization capabilities, widely used for computer vision tasks involving image classification. NDT models aim to combine these two distinct architectures to provide high performance on high-dimensional data, strong generalization and learning features, abilities obtained from NNs, with interpretability and transparent inference of a DT hierarchy. To achieve this, NNs should be integrated into a tree structure by implementing a differentiable routing function that controls how samples traverse down the tree. It unlocks the usage of gradient descent-based optimization methods with back-propagation during the training stage.

Generic NDT can be divided into three main modules:

1. *Router*. This decision module is responsible for sending input data to the child nodes. Each internal (decision) tree node contains a router module, which performs a routing function and partitions the input space. This mandatory module of NDTs is closely related to the reasoning mechanism of the model.
2. *Solver*. This prediction module is essential in NDT architecture. Each leaf (prediction) node contains a solver module that produces the outcomes. Depending on the implementation, it can provide a final output of both the class hierarchy and the static probability distribution over these classes.
3. *Learner*. A transformer module is assigned to every edge of the tree. The learner module transforms input samples from the parent node and passes them down to the child nodes. However, it is an optional module in NDTs. While some architectures, like ANT, incorporate it for representation learning, most NDTs use an identity function, passing features down the tree without data transformation.

Different NDT models are used in this paper to evaluate robustness against noise in input data, alongside a baseline ResNet18 model [5] for comparison.

Deep Neural Decision Forest (dNDF) is the first model considered for evaluating robustness to noise in input data. The model structure is quite straightforward. It uses a CNN architecture without the last fully connected linear layer to extract features, while a decision forest produces final predictions. The representations obtained from the trained CNN are passed to an ensemble of DTs with soft inference. These representations provide routing functions for all nodes in the forest of trees.

Unlike a standard decision forest with binary and deterministic routing, the dNDF model uses probabilistic routing. Routing functions, which determine the routing direction decisions for each internal node, are an output of Bernoulli random variables and are defined by a sigmoid activation function. When a sample traverses down a tree to a leaf node, the corresponding tree predictor gives the distribution over output classes. With this stochastic routing, each leaf node predictor provides a result averaged according to the probability of a sample reaching a leaf. The final prediction from a single tree is computed as a sum over all leaves of the learned class distribution multiplied by its routing probability. Then, the final dNDF model prediction for a sample is obtained by averaging the predictions of each tree in the ensemble. A learning procedure requires estimating both internal node parameters, responsible for decisions, and the leaf node parameters, responsible for the output predictions. The dNDF model uses a log-loss function and a two-step optimization strategy. Internal node parameters are randomly initialized and iterated during the learning procedure for a predefined number of training epochs. During each training epoch, the prediction parameters of all leaf nodes are updated independently for each tree by retrieving the current parameters from internal nodes, using a specific iterative scheme that solves a convex optimization problem.

Next, the training set is split into random mini-batches. The Stochastic Gradient Descent (SGD) optimization algorithm updates internal node parameters for each mini-batch. The original model structure is designed as an ensemble of DTs with soft inference, but it can also be converted into a single DT. This model is referred to as Deep Neural Decision Tree (dNDT). It has an identical structure but uses one tree instead of an ensemble of DTs.

At first glance, the next model structure looks similar to the previous model, utilizing a ResNet18 architecture without the final layer, which is replaced by a DT. In addition, much attention is given to model interpretability.

NBDT model implementation employs a differentiable oblique DT and incorporates several key design choices:

1. *Path probabilities for inference.* They help the model to tolerate highly uncertain intermediate decisions.
2. *Induced hierarchies*. Hierarchies are built from pre-trained NN weights to decrease the impact of overfitting.
3. *Tree Supervision Loss*. Training with surrogate hierarchy loss helps the model to make significantly better high-level decisions, leading to improved model generalization and performance.
4. *Pure leaves*. Each leaf corresponds to one pure class, allowing the model to select one path from the root to the leaf.

Similar to the dNDF implementation, the learning procedure shares many similarities. Input samples are transformed into feature representations from the ResNet18 NN backbone. After the feature extraction process, a final fully-connected layer is replaced by an oblique DT with soft inference. For each input sample and tree node, the probability of traversing to each child is computed by applying a softmax function of the inner product between the extracted features and the child node weight. The path probability for a leaf node representing a selected class is computed as a product of the traversal probabilities of each node along the unique path from the root to that leaf.

The next step is to build an induced hierarchy, which is essential for the NBDT model. Using data-based hierarchies like information gain or existing hierarchies like WordNet has notable drawbacks. The former is prone to input data overfitting, and the latter emphasizes conceptual rather than visual similarities.

NBDT is constructed using the hierarchy derived from pre-trained model weights to address this limitation. The process of the induced hierarchy creation starts with the final fully-connected layer weights from the ResNet18 backbone, viewing each row vector as a class representation. Next, the hierarchical agglomerative clustering is performed on normalized class representatives, iteratively pairing tree nodes and groups of nodes. For leaf nodes, the weights are normalized row vectors, while for internal nodes, the weights are the average of the weights of all leaf nodes within the corresponding subtree.

After the induced hierarchy generation, the decision nodes are labeled using the WordNet lexical database hierarchy of nouns. The earliest common ancestor for all leaf nodes in a subtree is identified to assign a corresponding WordNet noun to an internal node.

A hierarchical loss named Tree Supervision Loss is proposed to improve the NBDT training process. Being a modified cross-entropy loss, it is calculated over the class distribution of path probabilities. The total model loss is a weighted sum of the original cross-entropy loss and the Tree Supervision Loss. Tree Supervision Loss includes two variants: the Hard Tree Supervision Loss, which applies a cross-entropy at each node, and the Soft Tree Supervision Loss, which computes a cross-entropy loss over the distribution of leaf probabilities. Accordingly, based on the Tree Supervision Loss function variant, the NBDT inference can be either soft or hard. Despite hard inference being more intuitive and improving model interpretability, starting at the root node, each sample is sent to the child node with the most similar representative and traverses down the tree until a leaf node is reached. In the original paper, an NBDT model with hard inference, referred to as Hard NBDT, underperforms the NBDT model with soft inference. It was decided to evaluate both models' robustness to input perturbations.

The Gaussian blur method is used to create image perturbations. It is based on 2D Gaussian filtering, achieved by shifting a kernel filter over the image and performing a convolution based on the kernel size for each pixel in the image, without changing its dimensions. The kernel size depends on the value of σ, the standard deviation of the distribution. As σ increases, the kernel size increases accordingly, resulting in a more blurry image. The Gaussian function with (x, y) coordinates relative to the kernel center is shown below:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Constant noise learning is a straightforward method for training models that uses a predefined part of the perturbed data during training. The perturbed data remains fixed during training, generalizing from the original data with a certain proportion of modified samples during representation learning. Images perturbed by a Gaussian blur are used, with a certain part of the noisy images selected for each experiment. This method aims to achieve both high performance on clean images and noisy images, based on a proportion of perturbed samples in a training set. With an adjustable noise application probability, the constant noise learning method can significantly enhance the robustness of NDTs and the ResNet18 model to input noise, while remaining simple to integrate into the model training pipeline. More details are provided in Algorithm 1.

Algorithm 1 Constant noise learning

Require: Training set T, probability of applying noise p, $nEpochs$

for all $i \in \{1, \ldots, nEpochs\}$ **do**
  Shuffle T and divide into mini-batches $\{B_j\}$
  for each mini-batch B_j **do**
    for each image x in B_j **do**
      $r \sim \mathcal{U}(0, 1)$
      if $r < p$ **then**
        $x \leftarrow \text{GaussianBlur}(x)$
      end if
    end for
    Compute loss using current batch B_j
    Update model parameters
  end for
end for

Algorithm 1. Constant noise learning algorithm

The incremental noise learning method is a more complex method for improving the model's robustness. This strategy gradually adds perturbed data to the training set, progressively increasing the proportion with each training epoch. Training begins with a small proportion of perturbed data, and in the final stage of training, the proportion of perturbed data can reach almost the same size as the original training set. The proportion of noisy images to be added is computed based on the current epoch number and batch size, while the set of noisy images from previous epochs remain unaltered. It is assumed that this method is more effective in improving the robustness of NDT models, as it gradually introduces perturbed images based on the current stage of training, which can help the model learn sample features more naturally and generalize them better. A detailed implementation is provided in Algorithm 2.

Algorithm 2 Incremental noise learning

Require: Training set T, indices set S, initial noise percentage p_0, max noise percentage p_{max}, $nEpochs$

Initialize noisy indices set $S \leftarrow \emptyset$
Compute noise increment $\delta = \frac{p_{max} - p_0}{nEpochs - 1}$
for all $i \in \{1, \ldots, nEpochs\}$ **do**
  Compute target noise percentage $p_i = \min(p_0 + (i-1)\delta,\ p_{max})$
  Get number of new noisy indices $k = \lfloor p_i \cdot |T| \rfloor$
  Select random indices from non-noisy samples $M \leftarrow k - |S|$
  Update $S \leftarrow S \cup M$
  Shuffle T and divide into mini-batches $\{B_j\}$
  for each mini-batch B_j **do**
    for each image x with index idx in B_j **do**
      if $idx \in S$ **then**
        $x \leftarrow \text{GaussianBlur}(x)$
      end if
    end for
    Compute loss using current batch B_j
    Update model parameters
  end for
end for

Algorithm 2. Incremental noise learning algorithm

Experiments

The experiments are conducted on the CIFAR-10 dataset. It contains 60,000 images, including a training set of 50,000 images and a validation set of 1,000 images. The dataset is labeled with 10 different classes, with an equal number of images per class. Each sample is a 3-channel RGB image with square dimensions of 32 pixels. The CIFAR-10 dataset is widely used to validate computer vision tasks, particularly image

classification. It is well-suited for the needs of this study and provides sufficient input data complexity, making it relevant for robustness experiments.

Common hyperparameters are used for training and shared across all models. The number of training epochs is set to 40. A fixed random seed of 1 is applied to make all experiments more consistent and to provide fair results. A Gaussian blur σ value is set to 0.8, with the corresponding kernel filter size of 5 for image perturbation. An example of a perturbed image with Gaussian blur and clean images is demonstrated in Fig. 1.

Figure 1. Clean and noisy images from CIFAR-10

Training hyperparameters, including learning rate, input batch size, and others, vary according to the model.

Both dNDF and dNDT models use a batch size of 64. The tree structure has a depth of 8. In the case of the dNDF model, the ensemble contains 20 trees. This implementation employs the Adam optimization algorithm instead of the original SGD optimization used in the paper, with weight decay of 1^{-5} and a learning rate of 1^{-4}. For training models, a negative log-likelihood loss is applied. As a NN module, a 3-block CNN is used, each composed of 2 convolutional layers and batch normalization layers, the second layer in each block is followed by max pooling and dropout.

A batch size of 128 is used for training NBDTs and ResNet18 models. Two types of NBDT models are used in experiments: one with soft inference and another with hard inference.

Although NBDTs and ResNet18 models share the same ResNet18 backbone architecture and have similar training pipelines, their loss functions are different.

A NBDT model (with soft inference) uses a Soft Tree Supervision Loss during training, Hard NBDT uses a Hard Tree Supervision Loss, and ResNet18 uses a cross-entropy loss widely used for classification tasks. The NBDT, Hard NBDT, and ResNet18 models are trained with the SGD optimization algorithm with 0.9 momentum, 5^{-4} weight decay, and starting learning rate of 0.1 with a cosine annealing schedule where a maximum number of iterations is set to total epochs.

The constant and incremental noise learning training pipeline includes several new hyperparameters. Constant noise learning experiments were conducted by applying perturbed images to the original training set with varying probabilities denoted as ρ. The values of p are set to 0.05, 0.1, 0.2, 0.4, 0.6, and 0.8. For incremental noise learning experiments, the training process is configured to start with an initial noise level of 5 %, which gradually increases to a maximum of 95 % on the final training epoch.

Overall, 5 selected models proceeded through 8 learning procedures, resulting in 40 models trained under varying conditions on the CIFAR-10 dataset. Each model was trained using a regular training method with the original training set, a constant noise learning method with 6 different noise application probabilities, and an incremental noise learning method.

Experimental results are presented in a subsequent section.

Results and Discussion

The results of constant noise learning and the original training baseline are presented in Table 1, Table 2, and Table 3, evaluating the effect of the robust training strategy on model performance on the image classification tasks with validation on both clean and noisy images.

In Table 1, the performance of the model trained using the original baseline is compared against robustness to Gaussian blur. The performance is evaluated on both the original CIFAR-10 test set, which consists of clean images, as well as a noisy test set, containing images perturbed by Gaussian blur. The evaluation metrics are referred to as clean accuracy and noise accuracy, respectively. Additionally, the accuracy drop evaluation metric is used to demonstrate the performance drop on images with noise. Each model shows a significant performance degradation on the Gaussian blur test set, with an accuracy drop from 27.81% on the dNDT model to 36.86% on NBDT. The difference between NBDT and Hard NBDT models is also notable at 5.34%, indicating that NBDT with hard inference is less affected by noisy images than NBDT with soft inference. The original training baseline performance results are compared to robust training methods that should improve robustness across all models.

Table 1. **Original training baseline performance**

Model	Clean accuracy	Noise accuracy	Accuracy drop
dNDT	85.92%	58.11%	-27.81%
dNDF	86.21%	58.09%	-28.12%
ResNet18	93.75%	60.63%	-33.12%
NBDT	93.48%	56.62%	-36.86%
Hard NBDT	93.94	62.42%	-31.52%

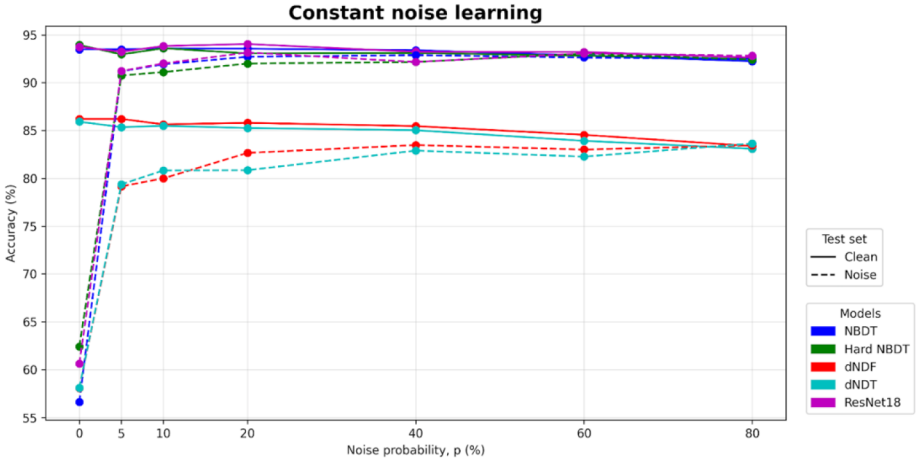

Figure 2. Constant noise learning performance on models

The results of the constant noise learning method using various application probabilities of perturbed images ρ to improve robustness, are presented in Table 2 and Table 3. Table 2 demonstrates the performance of each model on both clean and noisy images. Based on these results, Fig. 2 visualizes a performance change of each model depending on the noise probability ρ.

Table 2. **Constant noise learning performance**

Model	Acc.	p=0.05	p=0.1	p=0.2	p=0.4	p=0.6	p=0.8
dNDT	Clean	85.35%	85.49%	85.26%	85.03%	83.92%	83.09%
dNDT	Noise	79.37%	80.83%	80.85%	82.90%	82.27%	83.64%
dNDF	Clean	86.21%	85.64%	85.80%	85.46%	84.55%	83.37%
dNDF	Noise	79.16%	80.00%	82.66%	83.48%	83.01%	83.44%
ResNet18	Clean	93.23%	93.83%	94.04%	93.23%	93.21%	92.62%
ResNet18	Noise	91.22%	92.01%	93.15%	92.51%	93.07%	92.81%
NBDT	Clean	93.49%	93.58%	93.55%	93.39%	92.85%	92.24%
NBDT	Noise	91.19%	91.92%	92.71%	92.88%	92.63%	92.47%
Hard NBDT	Clean	93.60%	92.96%	93.06%	93.10%	92.82%	92.46%
Hard NBDT	Noise	90.75%	91.11%	92.00%	92.16%	93.05%	92.41%

Table 3 presents the accuracy drop for each model at different ρ values, along with a robustness gain metric. This metric indicates the model accuracy change from baseline to robust training on a noisy test set.

Table 3. **Constant noise learning vs Original baseline performance**

Model	Metric	p=0.05	p=0.1	p=0.2	p=0.4	p=0.6	p=0.8
dNDT	Acc. Drop	-5.98%	-4.66%	-4.41%	-2.13%	-1.65%	+0.55%
dNDT	R. Gain	+21.83%	+23.15%	+23.40%	+25.68%	+26.16%	+28.36%
dNDF	Acc. Drop	-7.05%	-5.64%	-3.14%	-1.98%	-1.54%	+0.07%
dNDF	R. Gain	+21.07%	+22.48%	+24.98%	+26.14%	+26.58%	+28.19%
ResNet18	Acc. Drop	-2.01%	-1.82%	-0.89%	-0.72%	-0.14%	+0.19%
ResNet18	R. Gain	+31.11%	+31.30%	+32.23%	+32.40%	+32.98%	+33.31%
NBDT	Acc. Drop	-2.30%	-1.66%	-0.84%	-0.51%	-0.22%	+0.23%
NBDT	R. Gain	+34.56%	+35.20%	+36.02%	+36.35%	+36.64%	+37.09%
Hard NBDT	Acc. Drop	-2.85%	-1.85%	-1.06%	-0.94%	+0.23%	-0.05%
Hard NBDT	R. Gain	+28.67%	+29.67%	+30.46%	+30.58%	+31.75%	+31.47%

The dNDT and dNDF models show higher volatility on the noisy test set at lower ρ values. However, from a ρ value of 0.2, the performance drop becomes closer to NBDTs and ResNet18 models. Based on the performance of models trained with constant noise learning, NBDTs and ResNet18 models are less dependent on varying probabilities of noisy images. The optimal performance, showing high accuracy on both clean and noisy test sets, is achieved at ρ values of 0.2 and 0.4. The best robustness results are obtained with ρ values of 0.6 and 0.8, causing a slight decrease in performance on clean images. The robustness gain compared to the original baseline performance for each model is significant, even the smallest ρ of 0.05 provides a robustness gain from 21.83% on NDT to 34.56% on NBDT.

Incremental noise learning provides excellent robustness results with minimal performance drop on the clean test set for dNDT, ResNet18, and Hard NBDT models, and a small performance improvement for dNDF and NBDT models. A trade-off between accuracy on the clean and noisy test sets is minimal for every model. Robustness gain performance ranges from 27.40% on NDT to 37.0% on the NBDT model. Results are presented in Table 4.

Table 4. **Incremental noise learning vs Original baseline performance**

Model	Clean acc.	Noise acc.	Acc. Drop	Robust. Gain
dNDT	83.00%	82.59%	-0.41%	+27.40%
dNDF	83.86%	83.92%	+0.06%	+28.18%
ResNet18	93.25%	93.09%	-0.16%	+32.96%
NBDT	92.71%	92.84%	+0.13%	+37.00%
Hard NBDT	92.28%	92.23%	-0.05%	+31.47%

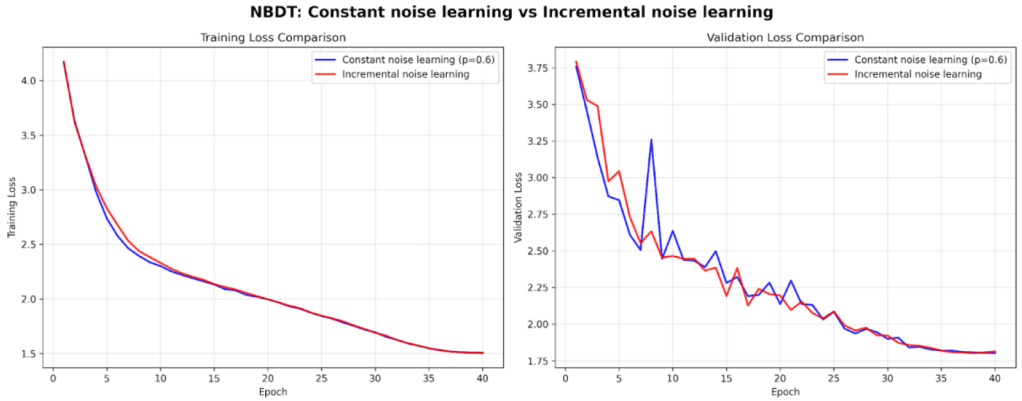

Figure 3. Loss changes of NBDT for constant and incremental noise learning

The constant learning method offers an adjustable trade-off between clean performance and robustness on noisy images, while incremental noise learning provides more stable results without relying on fine-tuning the application probability. Both methods can improve robustness to image perturbations without compromising performance on clean images. Although the dNDT and dNDF models have lower baseline performance, they achieve comparable robustness improvements using constant and incremental noise learning training strategies. For a detailed comparison of constant noise learning with a ρ value of 0.6 and

incremental noise learning, an NBDT model is selected. The training and validation loss for each training epoch for both approaches is demonstrated in Fig. 3.

Training losses are almost identical across 40 epochs, starting with very similar initial loss values, then steadily decreasing, showing a healthy training curve without overfitting. Validation losses have minor changes, indicating that the two models generalize differently. A constant noise learning method achieves slightly better final performance due to a lower loss with a notable spike at epoch 8, while incremental noise learning demonstrates a more stable curve with fewer sharp spikes. Thus, the NBDT model trained with incremental noise learning has a more stable validation loss progression. Although incremental noise learning demonstrates better stability during training and notable improvements, constant noise learning provides a simpler implementation, showing identical results. It is determined to be the preferred method for improving the robustness of the models to noise in input data.

Overall, a ResNet18 model shows slightly better results across all experiments. Moreover, it significantly outperforms dNDT and dNDF models in both training strategies, alongside the original training baseline. This confirms the assumption that, while the NDT models offer better interpretability, they involve some performance trade-offs. However, NDT models demonstrate comparable robustness improvements with robust training methods, demonstrating that they can benefit from the CNN-based method to improve robustness.

Conclusion

This work investigates the robustness of various NDT models, including dNDT, dNDF, NBDT, and Hard NBDT, to perturbations applied by a Gaussian blur in input data. The CIFAR-10 dataset is used for training models and validating experimental results. A ResNet18 model is used as a baseline for comparison with NDT models. Using the original training baseline, all models experienced a significant performance drop on images perturbed by a Gaussian blur. Two methods, originally developed to improve the robustness of CNN models, are applied to enhance robustness: constant noise learning and incremental noise learning. Experimental results demonstrate that both methods significantly improve robustness to noise for all models compared to the original model training baseline. Outcomes of the two robust training methods are nearly identical, showing no noticeable differences. Constant noise learning method offered an adjustable trade-off between performance on clean and perturbed images, while incremental noise learning provided more stable training results. However, constant noise learning is preferable for improving model robustness to noise in input data due to the simplicity of implementation. The experiments show that even a small part of noisy images significantly improves the robustness of the model.

Overall, NDT models demonstrate improvements in robustness to noise in input data comparable to the baseline ResNet18 model using both constant and incremental noise learning methods, indicating that NDTs can effectively benefit from adapted methods originally developed for CNNs.

Список літератури

1. Adaptive neural trees / Ryutaro Tanno [et al.] // International conference on machine learning. — 2019. — Pp. 6166–6175.
2. Andriushchenko M. Provably robust boosted decision stumps and trees against adversarial attacks / Maksym Andriushchenko, Matthias Hein // Advances in neural information processing systems. — 2019. — Vol. 32.
3. Carlini N. Towards evaluating the robustness of neural networks / Nicholas Carlini, David Wagner // 2017 IEEE symposium on security and privacy (SP). — 2017. — Pp. 39–57.
4. Deep neural decision forests / Peter Kontschieder [et al.] // IEEE international conference on computer vision. — 2015. — Pp. 1467–1475.
5. Deep residual learning for image recognition / Kaiming He [et al.] // IEEE conference on computer vision and pattern recognition. — 2016. — Pp. 770–778.
6. Decision forests, convolutional networks and the models in-between [Electronic resource] / Yani Ioannou [et al.] // ArXiv preprint arxiv:1603.01250. — 2016. — Mode of access: https://doi.org/10.48550/arXiv.1603.01250.
7. Distillation as a defense to adversarial perturbations against deep neural networks / Nicolas Papernot [et al.] // 2016 IEEE symposium on security and privacy (SP). — 2016. — Pp. 582–597.
8. Frosst N. Distilling a neural network into a soft decision tree [Electronic resource] / Nicholas Frosst, Geoffrey Hinton // ArXiv preprint arxiv:1711.09784. — 2017. — Mode of access: https://doi.org/10.48550/arXiv.1711.09784.
9. Goodfellow I. J. Explaining and harnessing adversarial examples [Electronic resource] / Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy // ArXiv preprint arxiv:1412.6572. — 2014. — Mode of access: https://doi.org/10.48550/arXiv.1412.6572.
10. Jordan M. I. Hierarchical mixtures of experts and the EM algorithm / Michael I. Jordan, Robert A. Jacobs // Neural computation. — 1994. — Vol. 6, no. 2. — Pp. 181–214.
11. NBDT: Neural-backed decision trees [Electronic resource] / Alvin Wan [et al.] // ArXiv preprint arxiv:2004.00221. — 2020. — Mode of access: https://doi.org/10.48550/arXiv.2004.00221.

12. Robust decision trees against adversarial examples / Hongge Chen [et al.] // International conference on machine learning. — 2019. — Pp. 1122–1131.
13. Self-born wiring for neural trees / Ying Chen [et al.] // Proceedings of the IEEE/CVF international conference on computer vision. — 2021. — Pp. 5047–5056.
14. Stock J. Strategies for Robust Image Classification [Electronic resource] / Jason Stock, Andy Dolan, Tom Cavey // ArXiv preprint arxiv:2004.03452. — 2020. — Mode of access: https://doi.org/10.48550/arXiv.2004.03452.
15. Suárez A. Globally optimal fuzzy decision trees for classification and regression / Alberto Suárez, James F. Lutsko // IEEE transactions on pattern analysis and machine intelligence. — 1999. — Vol. 21, no. 12. — Pp. 1297–1311.
16. Vos D. Efficient training of robust decision trees against adversarial examples / Daniel Vos, Sicco Verwer // International conference on machine learning. — 2021. — Pp. 10586–10595.

References

Andriushchenko, M., & Hein, M. (2019). Provably robust boosted decision stumps and trees against adversarial attacks. *Advances in Neural Information Processing Systems*, *32*.

Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 IEEE symposium on security and privacy (SP)* (pp. 39–57). IEEE.

Chen, H., Zhang, H., Boning, D., & Hsieh, C. J. (2019). Robust decision trees against adversarial examples. In *International Conference on Machine Learning* (pp. 1122–1131). PMLR.

Chen, Y., Mao, F., Song, J., Wang, X., Wang, H., & Song, M. (2021). Self-born wiring for neural trees. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 5047–5056).

Frosst, N., & Hinton, G. (2017). Distilling a neural network into a soft decision tree. *arXiv preprint arXiv*:1711.09784.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv*:1412.6572.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Ioannou, Y., Robertson, D., Zikic, D., Kontschieder, P., Shotton, J., Brown, M., & Criminisi, A. (2016). Decision forests, convolutional networks and the models in-between. *arXiv preprint arXiv*:1603.01250.

Jordan, M. I., & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural computation*, *6* (2), 181–214.

Kontschieder, P., Fiterau, M., Criminisi, A., & Bulo, S. R. (2015). Deep neural decision forests. In *Proceedings of the IEEE international conference on computer vision* (pp. 1467–1475).

Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016, May). Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)* (pp. 582–597). IEEE.

Stock, J., Dolan, A., & Cavey, T. (2020). Strategies for Robust Image Classification. *arXiv preprint arXiv*:2004.03452.

Suárez, A., & Lutsko, J. F. (1999). Globally optimal fuzzy decision trees for classification and regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21* (12), 1297–1311.

Tanno, R., Arulkumaran, K., Alexander, D., Criminisi, A., & Nori, A. (2019). Adaptive neural trees. In *International Conference on Machine Learning* (pp. 6166–6175). PMLR.

Vos, D., & Verwer, S. (2021). Efficient training of robust decision trees against adversarial examples. In *International Conference on Machine Learning* (pp. 10586–10595). PMLR.

Wan, A., Dunlap, L., Ho, D., Yin, J., Lee, S., Jin, H., ... & Gonzalez, J. E. (2020). NBDT: Neural-backed decision trees. *arXiv preprint arXiv*:2004.00221.

Мокрий М. В., Швай Н. О.

СТІЙКІСТЬ НЕЙРОННИХ ДЕРЕВ РІШЕНЬ ДО ШУМУ У ВХІДНИХ ДАНИХ ДЛЯ ЗАДАЧІ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

У роботі досліджується стійкість моделей нейронних дерев рішень, які об'єднують архітектуру нейронних мереж і дерев рішень, до шуму у вхідних даних для класифікації зображень. Було запропоновано використати два методи навчання для підвищення стійкості моделей, які початково використовувалися в згорткових нейронних мережах. Зашумлення зображень з набору даних CIFAR-10 відбувається за допомогою методу гаусівського розмиття. Було розглянуто вплив методів підвищення стійкості на моделей нейронних дерев рішень і показано, що стійкість моделей до шуму у вхідних даних значно покращується.

Ключові слова: нейронні дерева рішень, машинне навчання, стійкість, збурення зображень, класифікація зображень, комп'ютерний зір, згорткові нейронні мережі.

Матеріал надійшов 21.06.2025