From Soft Trees to Hard Trees: Gains and Losses

Xin Zeng [1] Jiayu Yao [1] Finale Doshi-Velez [1] Weiwei Pan [1]

[1]SEAS, Harvard University

Introduction

- **Soft & Hard Tree:** Soft tree is a tree in which each split is probabilistic and thus each input is assigned to each decision region with certain probability and the model prediction is a weighted sum of the prediction of each region. The biggest difference between soft trees and hard trees is the splitting network — soft trees have probabilistic splitting network while hard trees have deterministic one.
- **Motivation:** Interpretability is an important property of models that are deployed for high stakes decision-making tasks. Although trees are human-interpretable, their optimization is challenging. First, trees are often trained greedily. Second, trees cannot be easily incorporated into end-to-end training pipelines with other models since traditional tree training is not differentiable. To address these issues, a body of works proposes to first train a soft tree and then *harden* the soft tree into a hard one. In practice, the hardening process works well for trees in classification settings due to the discretization nature of classification tasks. Unfortunately, for regression tasks, these promises have not been realized: there is often a performance gap when obtaining hard trees via soft tree training.
- **Our Work:** We systematically study two types of soft trees. We summarize two key factors contributing to the performance gap of trees on regression tasks. (1) **Soft trees training is highly non-convex (with many local optima)**; thus, the training process is very sensitive to initialization and learning rate; thus, moving from optimizing a non-differentiable loss function to a continuous but highly non-convex may provide limited practical benefit. (2) **The hardening process does not preserve the relative orderings of the loss**: a soft tree with a low loss might harden to a tree with high loss, whereas a soft tree with slightly worse performance might harden to a much better hard tree.

Background

We formalize soft trees using Hierarchical Mixture of Experts (HMEs). We consider a regression task consisting of N observations, $\mathcal{D} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^N$ with $\mathbf{x}_n \in \mathbb{R}^l$, $\mathbf{y}_n \in \mathbb{R}^k$. We consider a binary HME of depth D with $2^D - 1$ *gating networks* at the non-terminal nodes and 2^D *expert networks* at the leaf nodes. The gating networks divide the input space into a set of regions with expert networks determining the predicted values of each region.

We are interested in the performance of hard trees. To obtain a hard tree, instead of marginalizing over leaf nodes, we assign the input to leaf nodes following the path with the greatest probability,

$$p_{\text{hard}}(\mathbf{y}_n|\mathbf{x}_n, \mathbf{v}, \tau) = p(\mathbf{t}|h_{j^*}(\mathbf{x}_n), \tau_{j^*})$$

where $j^* = \arg\max_j p(\xi_j|\mathbf{x}_n, \mathbf{v})$. We define the above process as *hardening*. By hardening, we hope to gain an interpretable model while retaining the predictive performance of the soft tree.

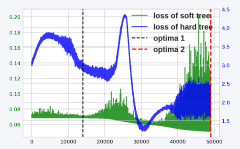
In this work, we investigate expert networks defined by two different link functions: (1) constant experts $h_j(\mathbf{x}_n) = \mathbf{c}$ where $\mathbf{c} \in \mathbb{R}^k$, (2) linear experts with $h_j(\mathbf{x}_n) = \mathbf{W}_j\mathbf{x}_n$ where $\mathbf{W}_j \in \mathbb{R}^{k\times l}$.

Experimental Setup

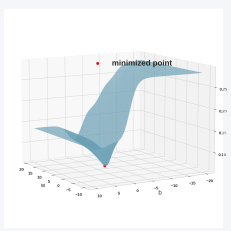
- **Overview:** We define the difference in terms of the predictive performance between the soft and the corresponding hardened model as the *performance gap due to hardening*. We investigate the performance gap with two types of soft trees: HMEs with constant experts for easy analyses, and HMEs with linear experts for more complicated regression tasks.
- **Datasets:** We designed two toy datasets: (1) a step function with 4 pieces, which matches inductive biases of HMEs with constant experts; (2) a cubic function $y = 3x^3$ with a sparse data region, which is a common benchmark for uncertainty quantification.
- **Evaluation Metrics:** For the step function, we compare the MSEs of soft and hard trees. For the cubic funcion, we investigate the likelihood.

Experimental Results

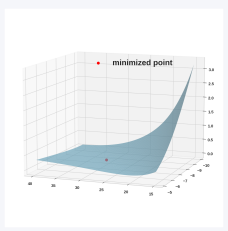
- **Issue 1: Ill-behaved Loss Landscape. The loss function of soft trees has local optima with high curvature and large plateaus, which are hard to escape.** Figure 1a shows the trace of the soft tree loss of the step function during training. We see that SGD reaches the first local optima around $15k$ iteration. Figure 1b plots the loss landscape near the first optima, which is deep, and thus hard to escape.



(a) Loss during Training



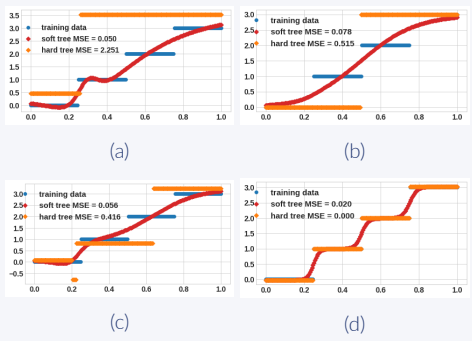
(b) Loss near Local Optima 1



(c) Loss near Local Optima 2

Figure 1. The landscape of the loss function of the step function: (a) trace of the loss function of $50k$ training iterations. (b)(c) The surface plots of the loss function near the two local optima (black line and red line in (a), respectively).

Soft tree training is highly sensitivity to hyperparameters. Because of the high curvature and the plateau of the loss landscape, learning is highly sensitive to the choice of initialization and learning rate, and thus the tree performance has a large variance.

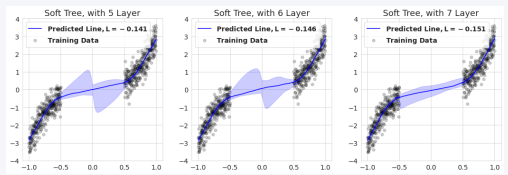


(a) (b) (c) (d)

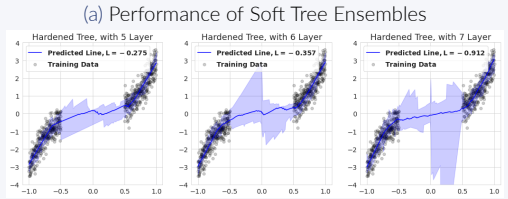
Figure 2. Plots of the best soft tree during training and the corresponding hard tree performance given different initialization strategy.

Experimental Results-Continued

- **Issue 2: Inconsistency with the Hard Tree Loss. Soft tree losses do not preserve the relative ordering of hard tree losses. A better soft tree does not necessarily harden to a better hard tree.** In Figure 1a, we see that during training, the trend of the hard tree loss (blue curve) is not consistent with the one of the soft tree (green curve). During training, the loss of the hard tree increases around the $20k$-th iteration and then decreases drastically around the $30k$-th iteration. Contrarily, the loss of the soft tree decreases in general. Comparing Figure 2a to 2b, we see that a better soft tree can result in a much worse hard tree.
  Similar inconsistency can be observed in terms of log-likelihood. Figure 3 shows the posterior predictive distribution of tree ensembles with increasing tree depth. We see that soft trees with higher log-likelihood harden to trees with lower log-likelihood.



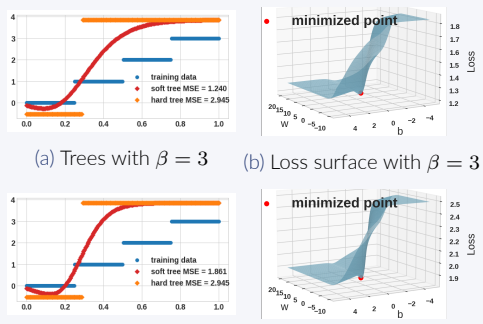
(a) Performance of Soft Tree Ensembles



(b) Performance of Corresponding Hard Tree Ensembles

Figure 3. Plots of posterior predictive distribution with different depths on the cubic function task (a) the soft tree ensembles (b) the corresponding hard tree ensembles.

- **The Trade-off Between Soft Optimization and Performance Gap Due to Hardening.** When β increases, the performance gap due to hardening decreases. However, the loss function becomes harder to optimize.



(a) Trees with $\beta = 3$ (b) Loss surface with $\beta = 3$

(c) Trees with $\beta = 5$ (d) Loss surface with $\beta = 5$

Figure 4. Plots of soft and hard tree performance and loss surface with $\beta = 3, 5$.

Conclusion

This paper systematically studies factors contributing to the performance gap between soft trees and their hardened counterparts. We also show that simple methods for closing the performance gap do not necessarily yield hard trees with better predictive performance - as they trade-off between difficult soft optimization and performance gap due to hardening. Although existing works aim to obtain predictive and interpretable models by globally optimizing soft trees and then hardening the solution, we show that this way of training hard trees does not get around fundamental issues on how fundamentally difficult it is to train a hard tree.

The 39th International Conference on Machine Learning (ICML 2022), Baltimore, Maryland USA — Responsible Decision Making in Dynamic Environments Workshop