LEARNING A TREE OF NEURAL NETS

Arman Zharmagambetov and **Miguel Á. Carreira-Perpiñán**
Dept. Computer Science & Engineering, UC Merced

Poster Number: 1823

1 Abstract

Much of the success of deep learning is due to choosing good neural net architectures and being able to train them effectively. A type of architecture that has been long sought is one that combines decision trees and neural nets. This is straightforward if the tree makes soft decisions (i.e., an input instance follows all paths in the tree with different probabilities), because the model is differentiable. However, the optimization is much harder if the tree makes hard decisions, but this produces an architecture that is much faster at inference, since an instance follows a single path in the tree. We show that it is possible to train such architectures, with guaranteed monotonic decrease of the loss, and demonstrate it by learning trees with linear decision nodes and deep nets at the leaves. The resulting architecture improves state-of-the-art deep nets, by achieving comparable or lower classification error but with fewer parameters and faster inference time. In particular, we show that, rather than improving a ResNet by making it deeper, it is better to construct a tree of small ResNets. The resulting tree-net hybrid is also more interpretable.

Work supported by NSF award IIS–2007147

2 Motivation: Decision Trees + Neural Nets

Deep Neural Nets

+ representation learning: can learn and extract good features
+ scalable and efficient optimization (e.g. using SGD)
+ etc...
– relatively long inference time
– interpretability is non-trivial

Decision Trees

+ interpretability: thanks to the hierarchical structure
+ fast inference time: instance follows unique root-leaf path
+ etc...
– difficult to train (non-differentiable, non-convex)
– do not extract/learn features
– simple models at each node (e.g. axis-aligned) → limited feature utilization

These advantages and limitations of the decision trees and neural nets motivate us for combining them to obtain a better model:

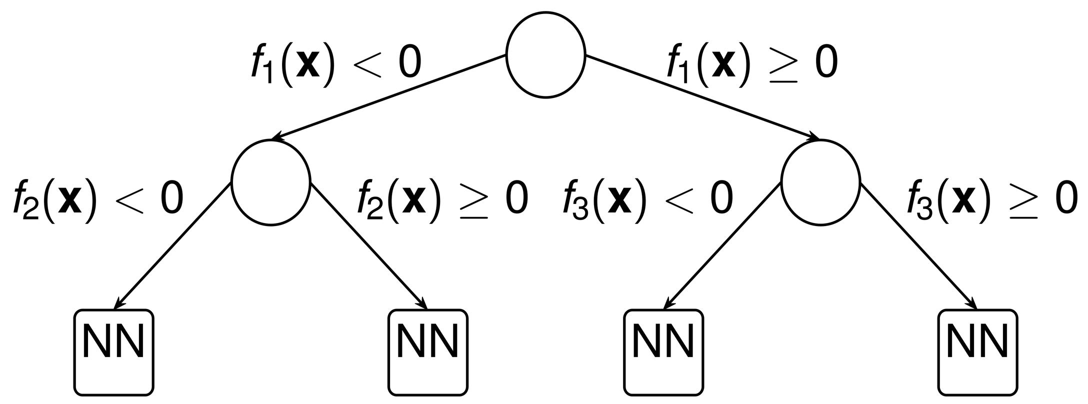
- Employ neural nets inside tree nodes. Nodes are now have feature extraction capability.
- The inference time is still "relatively" fast due to conditional computation.
- New model would be more interpretable compared to regular neural nets.

3 Training hybrids of trees and neural nets

Optimizing such models is difficult because the whole architecture is discrete. Majority of the existing works rely on:

- soft relaxation (a.k.a probabilistic trees) where each instance follows all root-to-leaf paths with certain probability: easy to optimize but have slow inference time (not a tree anymore).
- greedy top-down tree induction based on "purity" criteria: generate highly suboptimal trees.

Our proposal:



Consider above neural tree architecture which has:

- Neural nets in the leaves: each leaf specializes on some semantically similar group (e.g. subset of classes).
- Sparse linear decision nodes (i.e. $f_i(x) = \mathbf{w}_i^T\mathbf{x} + b_i$ in the above figure). **Motivation**: decision nodes are weak classifiers which are responsible to send an instance to the corresponding leaf. They are responsible for doing a very high level classification and the actual classification is done by NNs at the leaves.

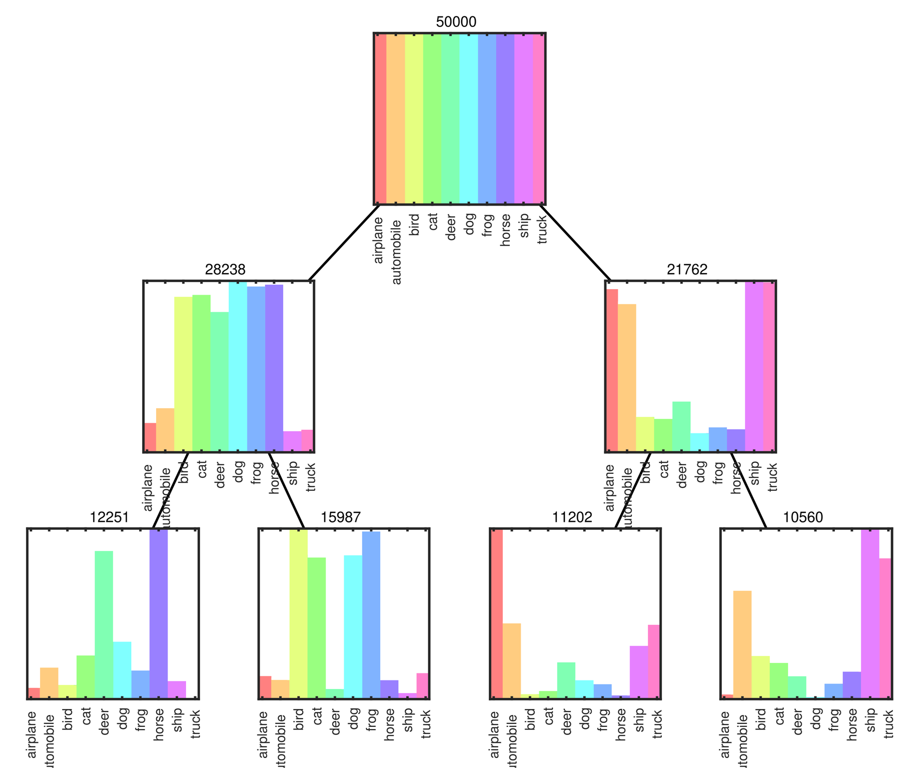
How to train this model? Use TAO–non-greedy tree learning algorithm: trains a decision tree with hard splits (i.e. input follow one root-to-leaf path); can handle tree nodes of arbitrary complexity (e.g. axis-aligned, oblique and beyond); shows promising results in training a single tree as well as tree-based ensembles. TAO repeatedly alternates between optimizing over a subset of nodes and fixing the remaining ones. The optimization itself is done by training a binary classifier in the decision nodes and a neural net in the leaves.

input training set $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$; initial tree $\mathbf{T}(\cdot;\boldsymbol{\Theta})$ of depth Δ
and with parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_i\}$, where $\boldsymbol{\theta}_i$ each node parameters
$\mathcal{N}_0, \dots, \mathcal{N}_\Delta \leftarrow$ nodes at depth $0, \dots, \Delta$, respectively
repeat
  for $d = 0$ **to** Δ
    parfor $i \in \mathcal{N}_d$
      if i is a leaf **then**
        $\boldsymbol{\theta}_i \leftarrow$ train a neural net on the training point that reach leaf i
      else
        compute the "best" child for each training points that reach node i
          and set it as a pseudolabel (call this modified training set $\mathcal{R}_i$)
        $\boldsymbol{\theta}_i \leftarrow$ train a linear binary classifier on $\mathcal{R}_i$
until stop
return T

4 Experiments

We experimentally evaluate our proposed method against tree-based or neural net based models or combinations of those. Our produced trees of neural nets have comparable performance w.r.t. deep nets but have more compact model sizes and fast inference time.

	Method	E_{test} (%)	Number of params	Inference (FLOPS)
MNIST	CART axis-aligned	12.50	(4k)	(12)
	CART oblique	11.00	(3.2M)	(9k)
	Linear Classifier	7.81	8k	16k
	tao-mnist-lin	4.11	0.1M	19k
	Random Forests	3.21	(3.6M)	(2.5k)
	Shallow NDF (sNDF)	2.80	(18M)	(18M)
	Alternating Decision Forests	2.71	(3.6M)	(2.5k)
	Neural Decision Tree (NDT)	2.10	(2M)	(0.5M)
	tao-mnist-cnn2	0.91	24k	0.3M
	Deep NDF (dNDF)	0.70	(0.5M)	(4.3M)
	Adaptive Neural Trees (ANT)	0.69	0.1M	–
	LeNet5	0.67	0.4M	4.2M
	tao-mnist-cnn3	0.67	21k	0.5M
CIFAR-10	ResNet20	8.51	0.27M	(58.42M)
	tao-cifar-resnet20	7.81	1.07M	(58.42M)
	ResNet56	6.73	0.85M	(183.11M)
	Adaptive Neural Trees (ANT)	6.72	1.30M	–
	tao-cifar-resnet56	6.51	1.70M	(183.11M)
	ResNet110	6.43	1.70M	(370.15M)
	DenseNet-BC(k=24)	3.74	27.2M	–



- Hierarchical structure allows interpretability in some sense.
- Above figure shows the class distributions of the points that reach the corresponding node. Each leaf focuses only on subset of classes rather than classifying all of them.