Learning a Tree of Neural Nets

Arman Zharmagambetov and Miguel Á. Carreira-Perpiñán

Dept. of Computer Science & Engineering
University of California, Merced

IEEE ICASSP 2021

Introduction

Deep Neural Nets

+ representation learning: can learn and extract good features
+ scalable and efficient optimization (e.g. using SGD)
+ etc...

Introduction

Deep Neural Nets

+ representation learning: can learn and extract good features
+ scalable and efficient optimization (e.g. using SGD)
+ etc...

Decision Trees

+ interpretability: thanks to the hierarchical structure
+ fast inference time: instance follows unique root-leaf path
+ etc...

Introduction

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

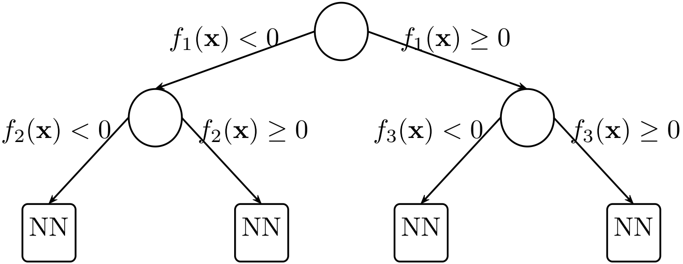
Our proposal: Decision Trees + Neural Nets

- Take the best from two paradigms:
  - Powerful hybrid model: trees with NN inside nodes
  - Hierarchical structure brings benefits: fast inference, interpretability

Our proposal: Decision Trees + Neural Nets

- Take the best from two paradigms:
  - Powerful hybrid model: trees with NN inside nodes
  - Hierarchical structure brings benefits: fast inference, interpretability
- Have been extensively studied:
  - Guo and Gelfand, 1992: Classification trees with neural network feature extraction.
  - Jordan and Jacobs, 1994: Hierarchical mixtures of experts.
  - Buló and Kontschieder, 2014: Neural decision forests.
  - Kontschieder et al., 2015: Deep neural decision forests.
  - Tanno et al., 2019: Adaptive neural trees.
  - and many others...

Our proposal: Decision Trees + Neural Nets



- 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 primitive classification and the main job of actual classification is done by NNs at the leaves.

Tree Alternating Optimization (TAO): non-greedy tree learning algorithm

Fundamental problem: hard to train (still non-differentiable and non-convex). Majority of the existing works rely on: soft relaxation (a.k.a probabilistic trees) and/or greedy top-down induction based on “purity” criteria.

Tree Alternating Optimization (TAO): non-greedy tree learning algorithm

Fundamental problem: hard to train (still non-differentiable and non-convex). Majority of the existing works rely on: soft relaxation (a.k.a probabilistic trees) and/or greedy top-down induction based on "purity" criteria.

We use TAO as the basis of our algorithm which has the following advantageous:

- 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 (Zharmagambetov et al., 2020; Zharmagambetov and Carreira-Perpiñán, 2020)

TAO for learning neural trees

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.

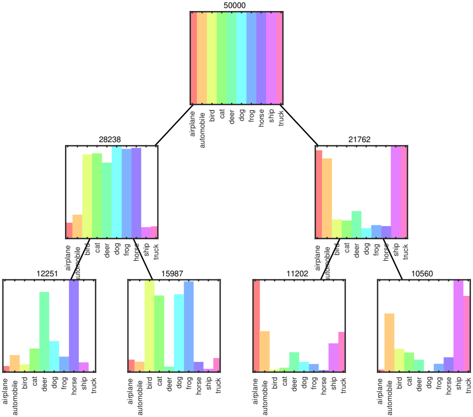
```
input training set $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$; initial tree $\mathbf{T}(\cdot;\boldsymbol{\Theta})$ of depth $\Delta$
and with parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_i\}$, where $\boldsymbol{\theta}_i$ each node parameters
$\mathcal{N}_0, \dots, \mathcal{N}_\Delta \leftarrow$ nodes at depth $0, \dots, \Delta$, respectively
repeat
  for $d = 0$ to $\Delta$
    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 $\mathbf{T}$
```

Experiments: model performance

	Method	E_{test} (%)	Number of params	Inference (FLOPS)
MNIST	tao-mnist-lin	4.11	0.1M	19k
	Random Forests	3.21	(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	–

The TAO neural trees are more compact (in terms of no. of params and inference time) yet accurate.

Experiments: Interpretability



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

Conclusion

- We address an issue of optimizing a tree of neural nets. Such hybrids show a good balance between error, model size and interpretability.
- Not easy to train. We have presented our approach, based on TAO, to train such hybrids with complex structures.
- The proposed method is efficient to train and can be scaled to large data.
- Future works: other types of neural tree architecture, other ways of interpreting neural trees.

- Work supported by NSF award IIS–2007147