Neurosymbolic models based on hybrids of convolutional neural networks and decision trees

Rasul Kairgeldin Miguel Á. Carreira-Perpiñán

Dept. Computer Science & Engineering
University of California, Merced

October 10, 2025

Motivation

Symbolic AI:

- Includes rule-based and logic-based models
- Transparent and interpretable reasoning
- Enables trust and correctness guarantees

Neural AI:

- Dominated by deep neural networks (NNs)
- Excels at perceptual tasks (e.g. image, text)
- Requires high compute, memory, and energy
- Acts as a black box—hard to interpret or debug

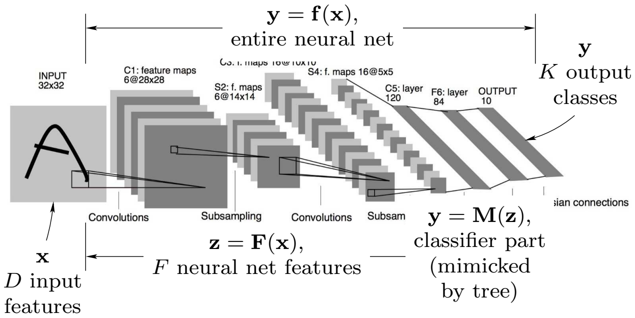
Neurosymbolic AI:

- Combines strengths of neural and symbolic AI
- Active research area [3, 4, 2]
- E.g. combination of CNNs and decision trees (DTs):
  - CNN extracts complex features from input (e.g. images)
  - DT performs classification in an interpretable way

Summary of Our Contribution

- Introduce a ”type-3 or Neuro—Symbolic” model (per Kautz [4]) to the neurosymbolic community
- Modify the TAO algorithm to learn sparse oblique decision trees with controllable sparsity for better interpretability and high accuracy
- Propose a way to interpret the results:
  - Visualizing receptive fields of neurons critical for class discrimination
  - Using density maps to show where neurons ”look” in the image
  - Demonstrating neuron-level detection of key features (e.g., gap vs. corner in Fashion MNIST)

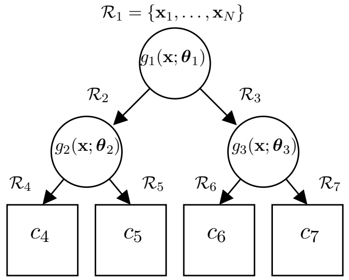
The neural net / tree hybrid model



- CNN: $\mathbf{y} = \mathbf{M}(\mathbf{F}(\mathbf{x}))$, where $\mathbf{x}$ is the image, $\mathbf{F}$ the convolutional layers, and $\mathbf{M}$ the fully-connected layers.
- Replace $\mathbf{M}$ with a sparse oblique tree $\mathbf{T}$ trained to map $\mathbf{F}(\mathbf{x}_n)$ to $\mathbf{M}(\mathbf{F}(\mathbf{x}_n))$ in teacher-student manner. The hybrid $\mathbf{T}(\mathbf{F}(\mathbf{x}))$ is interpretable and approximately functionally equivalent to $\mathbf{M} \circ \mathbf{F}$.

Sparse Oblique Decision Trees

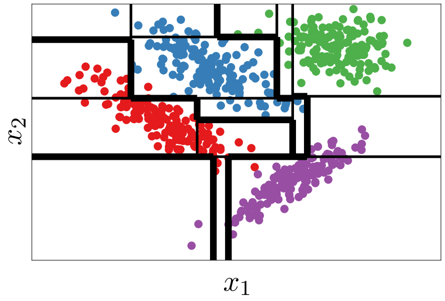
- Binary tree of depth Δ
- Decision nodes $\mathcal{D}$:
  - Each contains a routing function $g_i(\mathbf{x};\boldsymbol{\theta}_i)\colon \mathbb{R}^D \to \{\texttt{left}_i, \texttt{right}_i\} \subset \{\mathcal{D} \cup \mathcal{L}\}$
  - $g_i(\mathbf{x};\boldsymbol{\theta}_i) = \texttt{left}_i$ if $\mathbf{w}_i^T\mathbf{x} + w_{0i} < 0$, otherwise $\texttt{right}_i$
- Leaf nodes $\mathcal{L}$:
  - Each contains a constant label classifier that outputs a single class $c_j \in \{1, \dots, K\}$
- Tree's learnable parameters: $\boldsymbol{\Theta} = \{(\mathbf{w}_i, w_{0i})\}_{i\in\mathcal{D}} \cup \{c_j\}_{j\in\mathcal{L}}$
- Tree routing function $\mathbf{T}(\mathbf{x}_n;\boldsymbol{\Theta})$ directs a patch $\mathbf{x}_n$ from a root to a single leaf and predicts a corresponding class c_j



$\mathcal{R}_1 = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
$g_1(\mathbf{x};\boldsymbol{\theta}_1)$
$\mathcal{R}_2$
$\mathcal{R}_3$
$g_2(\mathbf{x};\boldsymbol{\theta}_2)$
$g_3(\mathbf{x};\boldsymbol{\theta}_3)$
$\mathcal{R}_4$
$\mathcal{R}_5$
$\mathcal{R}_6$
$\mathcal{R}_7$
c_4
c_5
c_6
c_7

Sparse Oblique Decision Trees

Univ.: Δ=8, 15 leaves, 4% error



sparse oblique: Δ=3, 5 leaves, 2% error

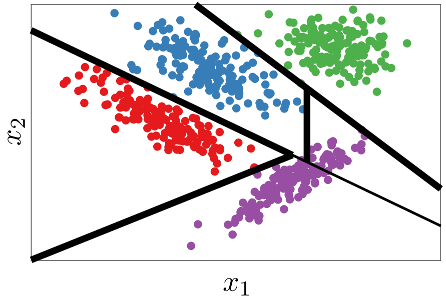


Figure: Partitioning by a univariate (left) and sparse oblique tree (right). By allowing feature correlations, oblique trees achieve better performance with much smaller trees.

Formulation of the tree learning problem

- Given a oblique tree $\mathbf{T}(\mathbf{x}, \boldsymbol{\Theta})$ of a fixed structure (e.g. a complete tree of depth Δ) and initial parameters (e.g. random), we use Tree Alternating Optimization (**TAO** [1]) to minimize the following objective:

$$
\begin{aligned}
E(\boldsymbol{\Theta}) &= \sum_{n=1}^{N} L(\mathbf{y}_n, \mathbf{T}(\mathbf{x}_n; \boldsymbol{\Theta})) + \lambda \sum_{i \in \mathcal{D}} h_\alpha(|\mathcal{R}_i|)\, \|\mathbf{w}_i\|_1 \\
h_\alpha(t) &= \begin{cases} 1, & t = 0 \\ t^\alpha, & t > 0 \end{cases}
\end{aligned}
\tag{1}
$$

- where $L(\cdot,\cdot)$ is the loss, $\boldsymbol{\Theta} = \{(\mathbf{w}_i, w_{i0})\}_{i \in \mathcal{D}} \cup \{c_j\}_{j \in \mathcal{L}}$ are the set of all learnable model parameters, ℓ_1 is penalty over the weight vectors to promote sparsity via hyperparameters $\lambda \geq 0$, $\mathcal{R}_i$ is the reduced set of node i and $|\mathcal{R}_i|$ its cardinality.

Intuitive explanation of the TAO algorithm

- TAO optimizes iteratively an arbitrary objective function over the parameters of the tree model
- Updates each node's parameters in sequence by solving a reduced problem
- Iterates over nodes in reverse BFS order
- For non-descendant nodes can update parameters in parallel
- Decreases objective funciton monotonically

The role of the regularization

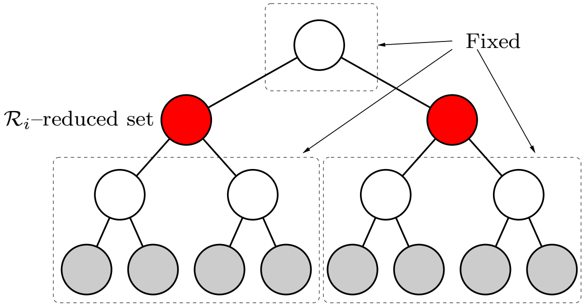
- Hyperparameter λ controls overall sparsity in the tree
- Large λ also induces pruning $\rightarrow$ automatic tree structure learning
- Shallow nodes (e.g., root) become denser than deeper nodes (e.g., leaf parents)
- To address it, introduce hyperparameter α to control node-level sparsity relative to the number of instances handled

Tree Alternating Optimization (TAO): Separability Condition

- TAO is used to optimize objective function (eq (1))
- TAO iteratively updates each node's parameters so as to decrease the objective function
- TAO algorithm is based on 2 theorem: separability condition and reduced problem over nodes.

TAO: Separability Condition

Assume the parameters are not shared across nodes ($i \neq j \Rightarrow \boldsymbol{\theta}_i \cap \boldsymbol{\theta}_j = \emptyset$). Assume nodes i and j are non-descendant of each other, and all other parameters ($\boldsymbol{\Theta}_{rest} = \boldsymbol{\Theta} \backslash \{\boldsymbol{\theta}_i, \boldsymbol{\theta}_j\}$) are fixed.



TAO: Separability Condition

In general, if $\mathcal{S} \subset \mathcal{N}$ is a nonempty set of non-descendant nodes in the tree and $\{\boldsymbol{\theta}_i : i \in \mathcal{S}\}$ is the set of their parameters, then $E(\boldsymbol{\Theta})$ can be rewritten as:

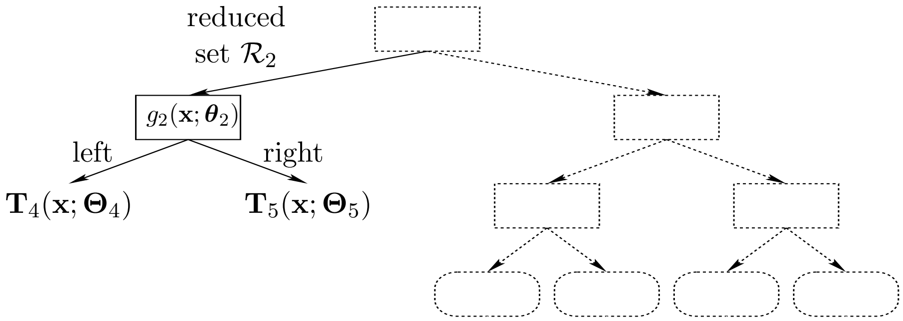
$$E(\boldsymbol{\Theta}) = \sum_{i \in \mathcal{S}} E_i(\boldsymbol{\theta}_i) + E_{\text{rest}}(\boldsymbol{\Theta}_{\text{rest}})$$

Optimization of $i \in \mathcal{S}$ can be done in parallel, which drastically facilitates the process.

TAO: Reduced Problem Over Leaves

Reduced problem over leaves states that for a leaf $j \in \mathcal{L}$ the problem (1) is reduced to a form involving the original loss but only over the parameters of the leaf predictor function. It can be solved by finding the majority class (or mean value of the samples in the reduced set for regression)

TAO: Reduced Problem Over Decision Nodes


reduced set $\mathcal{R}_2$
$g_2(\mathbf{x};\boldsymbol{\theta}_2)$
left
right
$\mathbf{T}_4(\mathbf{x};\boldsymbol{\Theta}_4)$
$\mathbf{T}_5(\mathbf{x};\boldsymbol{\Theta}_5)$

- $\mathcal{R}_i$ is the reduced set of decision node i.
- Function $l_{in} : \mathcal{C}_i \rightarrow \mathbb{R}$ as $l_{in} = L(\mathbf{y}_n, \mathbf{T}_z(\mathbf{x}_n;\boldsymbol{\Theta}_z))$, for any $z \in \mathcal{C}_i$ (child of i).

TAO: Reduced Problem Over Decision Nodes

This is NP-hard to optimize, but it can be approximated by a convex surrogate (we use a weighted ℓ_1-regularized logistic regression classifier):

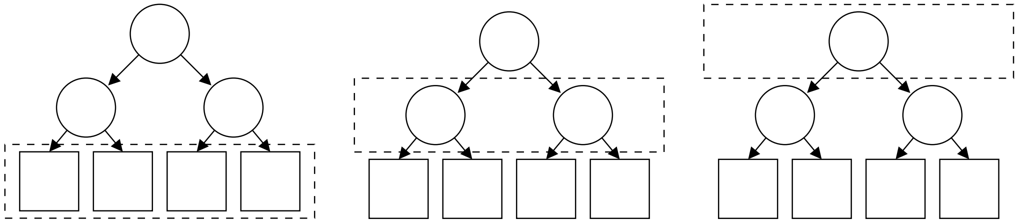
$$\min_{\boldsymbol{\theta}_i} \sum_{n \in \mathcal{R}_i} \overline{L}(\overline{y}_n, g_i(\mathbf{x}_n; \boldsymbol{\theta}_i)) + \lambda\, h_\alpha(|\mathcal{R}_i|)\, \|\mathbf{w}_i\|_1 \tag{2}$$

where a weighted 0-1 loss $\overline{L}_{in}(\overline{y}_{in}, \cdot)$ of each sample $n \in \mathcal{R}_i$ is defined as $\overline{L}_{in}(\overline{y}_{in}, y) = l_{in}(y) - l_{in}(\overline{y}_{in})\ \forall y \in \{\texttt{left}, \texttt{right}\}$, where $\overline{y}_{in} = \arg\min_y l_{in}(y)$ is a "pseudolabel" indicating the optimal child to route to.

TAO: Reduced Problem Over Decision Nodes

- We can rewrite RP objective as "avg-loss + λ' reg", with $\lambda' = \lambda N_i^{\alpha-1}$, avg-loss is the loss per instance in node i; reg $= \|\mathbf{w}_i\|_1$ (an *effective sparsity* hyperparameter)
- $\alpha < 1$: large RS penalized less $\rightarrow$ the root is denser
- $\alpha = 1$: all nodes penalized equally
- $\alpha > 1$: large RS penalized more $\rightarrow$ the root is sparser
- Can produce trees that are sparser and more accurate than regular TAO (i.e., $\alpha = 0$)

TAO: Optimization



At each TAO teration, nodes of the same depth are trained in parallel and the algorithm proceeds in the reverse BFS order.

TAO: Optimization

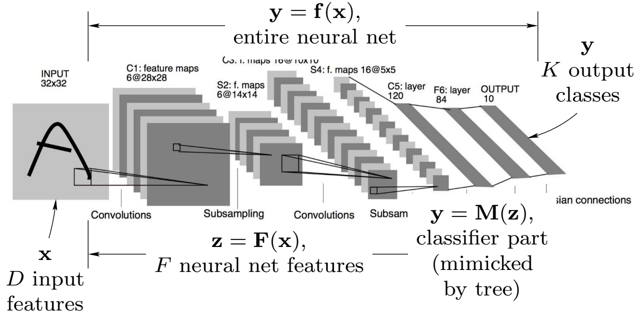
```
input training set; initial tree $\mathbf{T}(\cdot;\boldsymbol{\Theta})$ of depth $\Delta$
$\mathcal{N}_0,\dots,\mathcal{N}_\Delta \leftarrow$ nodes at depth $0,\dots,\Delta$, respectively
generate $\mathcal{R}_1 \leftarrow \{1,\dots,N\}$ using initial tree
repeat
   for $d = \Delta$ down to $0$
      parfor $i \in \mathcal{N}_d$
         if $i$ is a leaf then
            $\boldsymbol{\theta}_i \leftarrow$ fit a leaf predictor
               $\mathbf{g}_i$ on reduced set $\mathcal{R}_i$
         else
            generate pseudolabels $\overline{y}_n$ for each point $n \in \mathcal{R}_i$
            $\boldsymbol{\theta}_i \leftarrow$ minimizer of the reduced problem:
            $\sum_{n\in\mathcal{R}_i} \overline{L}(\overline{y}_n, g_i(\mathbf{x}_n;\boldsymbol{\theta}_i)) + \lambda\, h_\alpha(|\mathcal{R}_i|)\, \|\mathbf{w}_i\|_1$
   update $\mathcal{R}_i$ for each node
until stop
prune dead subtrees of $\mathbf{T}$
return $\mathbf{T}$
```

Computational Complexity: Training and Inference

Training complexity for a complete tree of depth Δ and K leaves on a dataset $\{\mathbf{x}_n\}_{n=1}^N \subset \mathbb{R}^D$:

- Root node: $\mathcal{O}(\Delta DN) + \mathcal{O}(cDN)$(pseudolabel assignment and binary classifier fitting with c iterations)
- Decision nodes at level Δ_i can be optimized in parallel, the complexity is: $\mathcal{O}(\Delta_i DN) + \mathcal{O}(cDN)$
- Total complexity at most $\mathcal{O}(\Delta^2 DN) + \mathcal{O}(c\Delta DN) \approx \mathcal{O}(DN \log^2 K)$
- Inference time is $\mathcal{O}(D \log K)$

The neural net / tree hybrid model



- Our neurosymbolic model consists of the convolutional layers of a LeNet NN followed by a tree of depth 4
- The tree is a very close mimic to the NN classifier layers, so it can replace them and we can trust to explain the NN features and classification

Experiments: Sparse Oblique Decision Tree Visualization

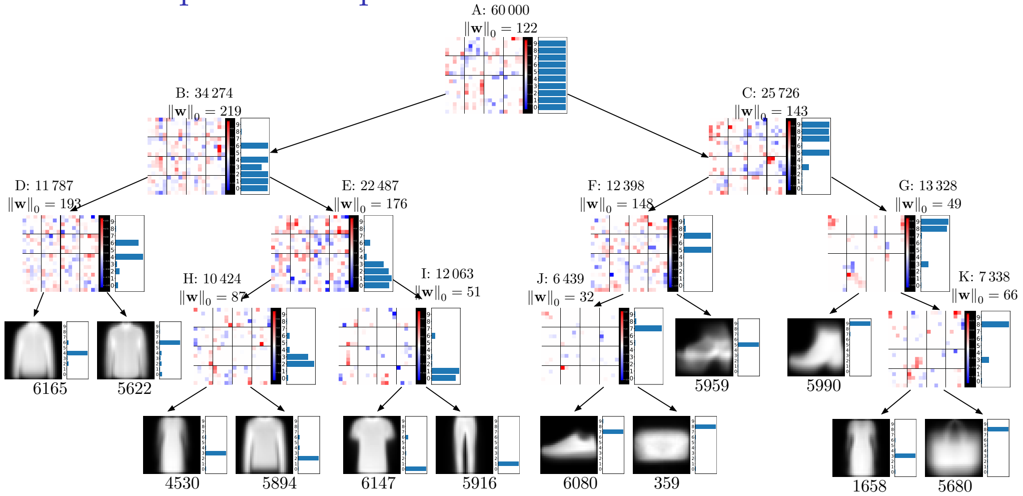


Figure: Tree trained on LeNet embeddings on Fashion MNIST dataset $\lambda = 0.001, \alpha = 1$. $E_{\text{train}} = 5.4\%$, $E_{\text{test}} = 11.7\%$ and # non-zero params is 1298

Experiments: Sparse Oblique Decision Tree Class Separation

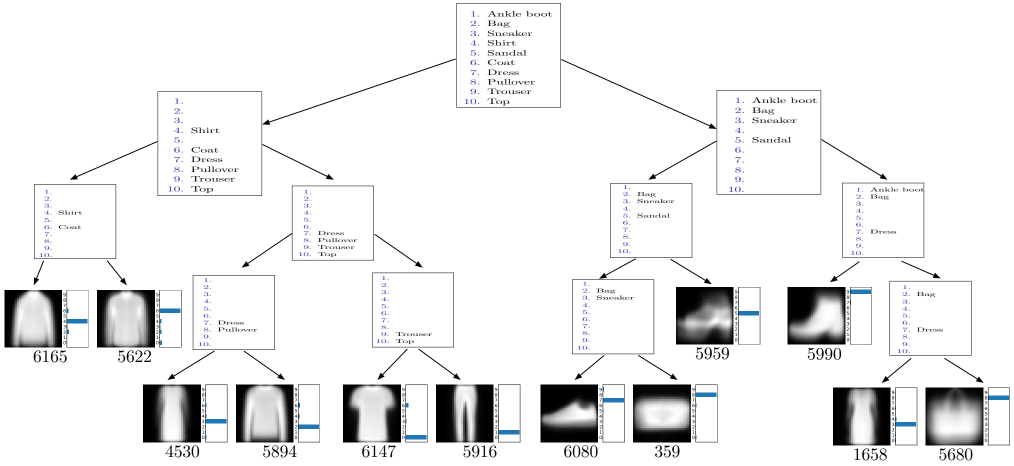


Figure: Class separation for each decision node. Tree trained on LeNet embeddings on Fashion MNIST dataset $\lambda = 0.001, \alpha = 1$

Experiments: Density Map

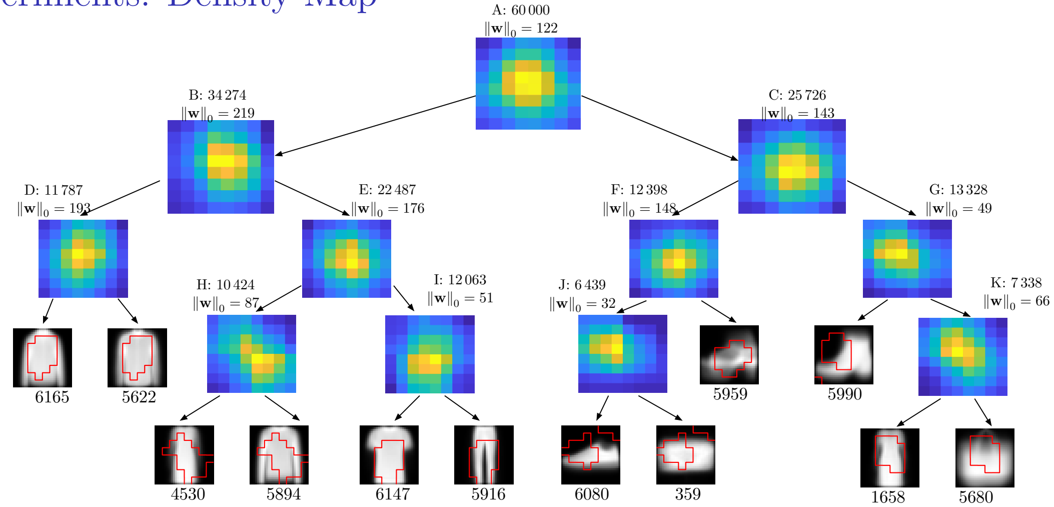
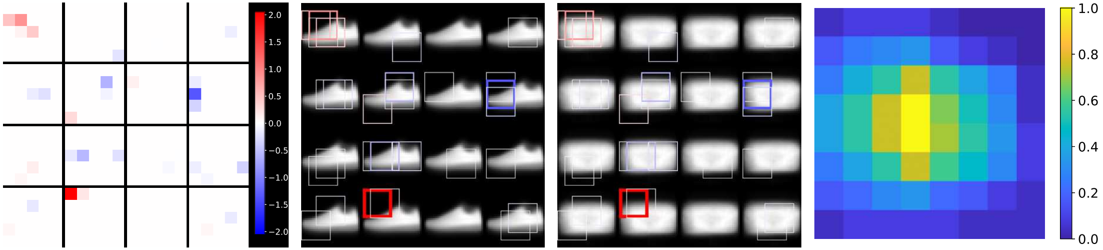
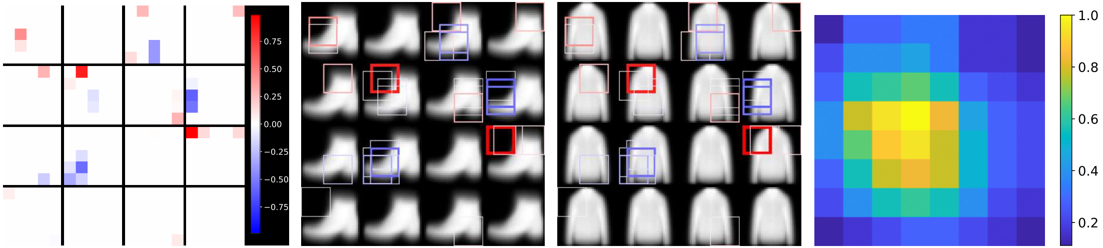


Figure: Tree trained on LeNet embeddings on Fashion MNIST dataset $\lambda = 0.001, \alpha = 1$. Similar to fig. 24 each decision node contains "density" map of the receptive fields. Each leaf node contains contour produced by the parent decision node weights and density map.

Experiments: Where is decision node looking in the image?





Experiments: Changing Classification

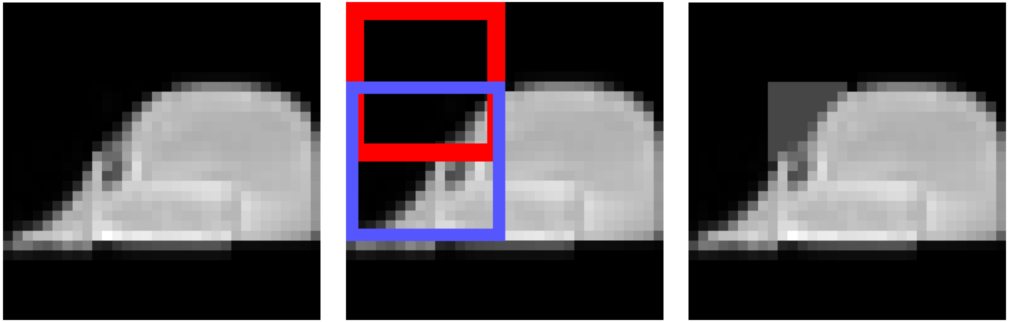


Figure: Sample of class 8 misclassified as 7 (Left). Receptive field of neurons from the last convolutional layer of Lenet with largest positive (red) and negative (blue) weights in oblique decision node J (Middle). Small changes in the intersection of two regions fixed the misclassification error (Right).

Conclusion

- Added new regularization and modified the TAO algorithm to control feature sparsity across the tree
- Two-stage training:
  - Train CNN with SGD
  - Train a tree with TAO to replace CNN's fully connected layers using a teacher-student approach
- Combines CNN's representational power with tree's interpretability
- Enables:
  - Tracing which neurons affect which classes
  - Visualizing where neurons focus in the image.
  - Explaining (mis)classifications
  - Editing images to change classification

Future Work

- LLM probing using sparse oblique decision trees
- Joint optimization of NN and a tree

References

[1] M. Á. Carreira-Perpiñán and P. Tavallali. Alternating optimization of decision trees, with application to learning sparse oblique trees. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pages 1211–1221. MIT Press, Cambridge, MA, 2018.

[2] A. d'Avila Garcez and L. C. Lamb. Neurosymbolic AI: The 3rd wave. *Artificial Intelligence Review*, 56:12387–12406, 2023.

[3] P. Hitzler and M. K. Sarker, editors. *Neuro-Symbolic Artificial Intelligence: The State of the Art*. Number 342 in Frontiers in Artificial Intelligence and Applications. IOS Press, 2021.

[4] H. Kautz. The third AI summer. *AI Magazine*, 43(1):105–125, Spring 2022.