A Simple, Effective Way to Improve Neural Net Classification: Ensembling Unit Activations with a Sparse Oblique Decision Tree

Arman Zharmagambetov and Miguel Á. Carreira-Perpiñán

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

IEEE ICIP 2021

Overview and Motivation

- Deep learning has become highly successful in many applications involving complex inputs such as images, audio or text. They compute features that capture important properties of the input and these features can be invariant to certain transformations (translation, rotation, etc.).
- As a result, we have now a proliferation of deep net architectures of ever increasing complexity, containing millions of parameters.
- At the same time, the cost of training such models (computing time, memory size, energy consumption and human expertise) has escalated dramatically. This also leads to diminishing returns: large increases in size within a family quickly translate into tiny reductions in error.

Overview and Motivation

- Diminishing return:

ResNet	size	error	VGG	size	error	DenseNet-BC	size	error
20	0.27M	8.75%	11	9.23M	8.01%	100,12	0.80M	22.27%
32	0.46M	7.51%	13	9.41M	6.48%	94,12	0.97M	22.16%
56	0.85M	6.97%	16	14.72M	6.41%	100,14	1.08M	22.01%
110	1.70M	6.43%	19	20.03M	6.35%	250,24	15.30M	17.60%
1202	19.40M	7.93%				190,40	25.60M	17.18%

Overview and Motivation

- Deep learning has become highly successful in many applications involving complex inputs such as images, audio or text. They compute features that capture important properties of the input and these features can be invariant to certain transformations (translation, rotation, etc.).
- As a result, we have now a proliferation of deep net architectures of ever increasing complexity, containing millions of parameters.
- At the same time, the cost of training such models (computing time, memory size, energy consumption and human expertise) has escalated dramatically. This also leads to diminishing returns: large increases in size within a family quickly translate into tiny reductions in error.
- A different, proven way to construct accurate classifiers is via ensemble learning and we propose a new simple ensembling mechanism that is specially designed for neural nets.

Proposed Approach

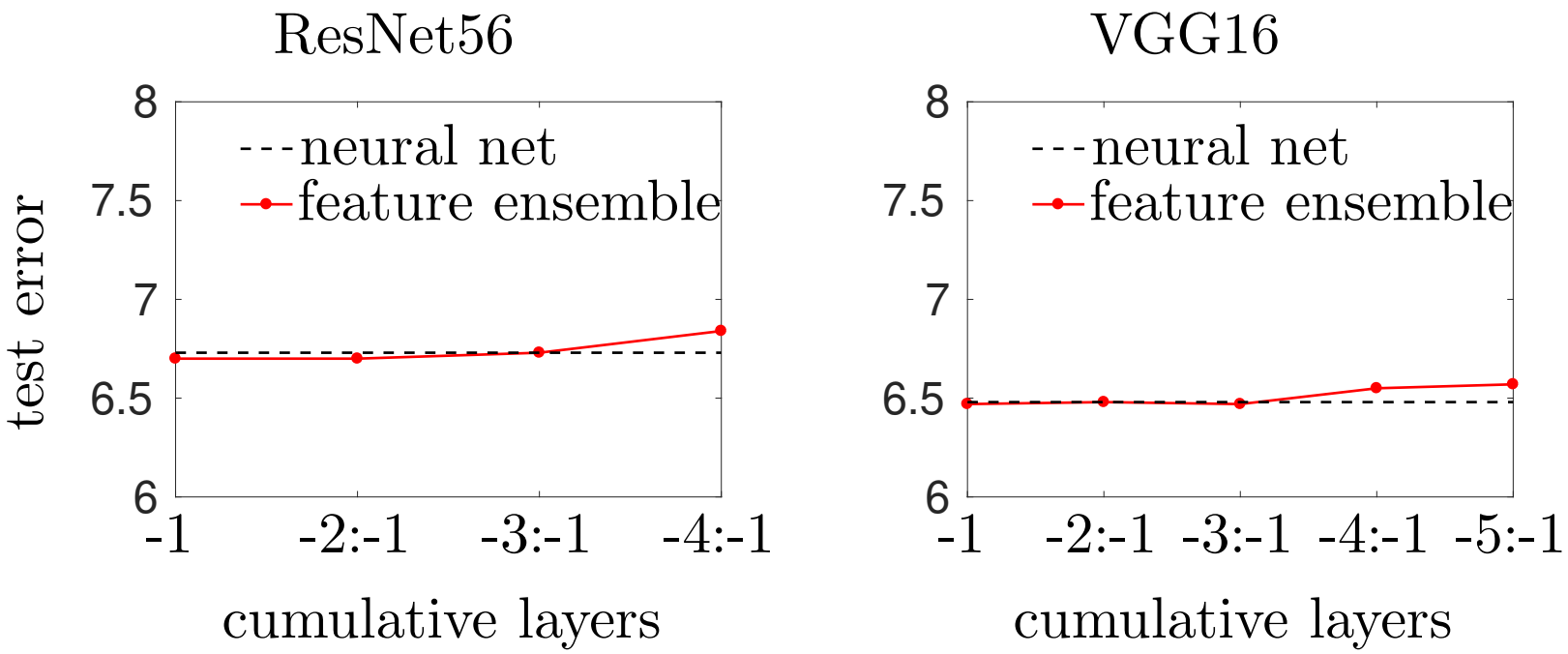
- Assume we have a dataset of input instances and their labels (in K classes); and several **deep nets** trained, somehow, on that dataset.
- Then first construct an **ensemble feature vector** by picking a subset of features (output of intermediate layers) from each net and concatenating them;
- Then we train a **sparse oblique tree classifier** with TAO on a dataset where the inputs are the ensemble feature vectors.
- This procedure is generic and admits multiple variations:
  - Features: within net and across nets;
  - Deep net training: train the individual nets ourselves or download existing neural nets.

Training with TAO

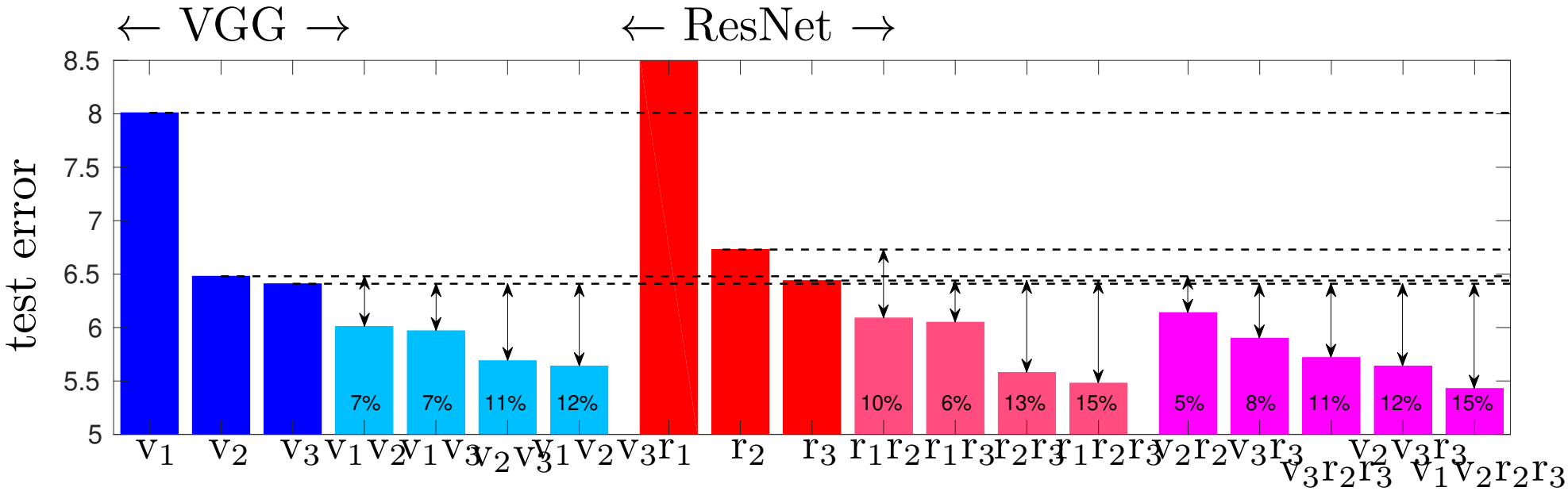
Why TAO? they produce trees and forests with high accuracy; they are very fast (at training and inference); and they do feature selection (useful when input dimension is high). Pseudocode:

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 classifier 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

Experiments: within net



Experiments: across net



Conclusion

- We have proposed a new form of ensembles specifically tailored for deep nets based on training sparse oblique DT on neural net features.
- While this barely improves within a single net, it significantly and consistently improves if ensembling features across different nets.
- The decision tree, trained with the TAO algorithm, allows us to handle efficiently and accurately the resulting high-dimensional feature vector.
- Future works: jointly training the entire architecture, using a forest classifier (rather than a tree), and neural net compression using sparse oblique trees.
- Work supported by NSF award IIS–2007147