Deep Neural Decision Forests

Peter Kontschieder[1] Madalina Fiterau[2] Antonio Criminisi[1] Samuel Rota Bulò[1,3]

Microsoft Research[1] Cambridge, UK
Carnegie Mellon University[2] Pittsburgh, PA
Fondazione Bruno Kessler[3] Trento, Italy

Supplementary Material

This document provides the following supplementary contributions:

- Additional details regarding the ImageNet experiment (see Section 1).
- Detailed derivations for the gradient term depending on the decision tree splits (see Section 2).
- The proof that our update rule for the leaf predictions $\boldsymbol{\pi}$ in Equ. (11) of our main ICCV paper monotonically decreases the risk R until a fixed point is reached (see Section 3).

1. ImageNet experiment: GoogLeNet vs. dNDF.NET architectures

This section provides additional description for the ImageNet experiment [3]. In particular, we describe and illustrate the changes we made to GoogLeNet [4] to obtain our proposed dNDF**.**NET architecture, using deep neural decision forests (dNDFs) as classifiers. The GoogLeNet architecture we have used as basis for our experiments (see left illustration in Fig. 1, taken from [4]) has a reported Top5-Error of 10.07%, when used in a single-model, single-crop setting (see first row in Tab. 3 of [4]).

Fig. 1 (right illustration) shows that we have introduced two different modifications with respect to the original GoogLeNet architecture. First, we have connected the outputs of the Concat layers to the inputs of the AveragePool layers (as described in the main paper), visualized by red arrows in the plot. The resulting, modified baseline network is dubbed GoogLeNet$\star$ and achieves a Top5-Error of 10.02% when using conventional SoftMax layers as in the original network. The implementation yielding this score was realized in the Distributed (Deep) Machine Learning Common (DMLC) library [2, 1][1], using resized images with dimensionality 100x100 as described in [2]. The training used the standard settings for GoogLeNet, stochastic gradient descent with 0.9 momentum, fixed learning rate schedule, decreasing the learning rate by 4% every 8 epochs. We trained GoogLeNet$\star$ with mini-batches composed of 50 images.

In order to obtain a *Deep Neural Decision Forest* architecture coined dNDF**.**NET, we have replaced each Softmax layer from GoogLeNet$\star$ with a forest consisting of 10 trees (each fixed to depth 15), resulting in a total number of 30 trees. For our architecture, which we implemented in DMLC as well, we trained the network for 1000 epochs using mini-batches composed of 100.000 images. This is feasible due to distribution of the computational load to a cluster of 52 CPUs and 12 hosts, where each host is equipped with a NVIDIA Tesla K40 GPU.

We refer to the individual forests as dNDF_0, dNDF_1 and dNDF_2, where dNDF_0 is closest to the input layer and dNDF_2 is the final (last) layer in the architecture. Each tree is a balanced and fixed depth 15 tree, which means that the total number of per-tree split nodes is $2^{15} - 1 = 32.767$ and the number of leaf nodes is $2^{15} = 32.768$.

[1] https://github.com/dmlc/cxxnet.git

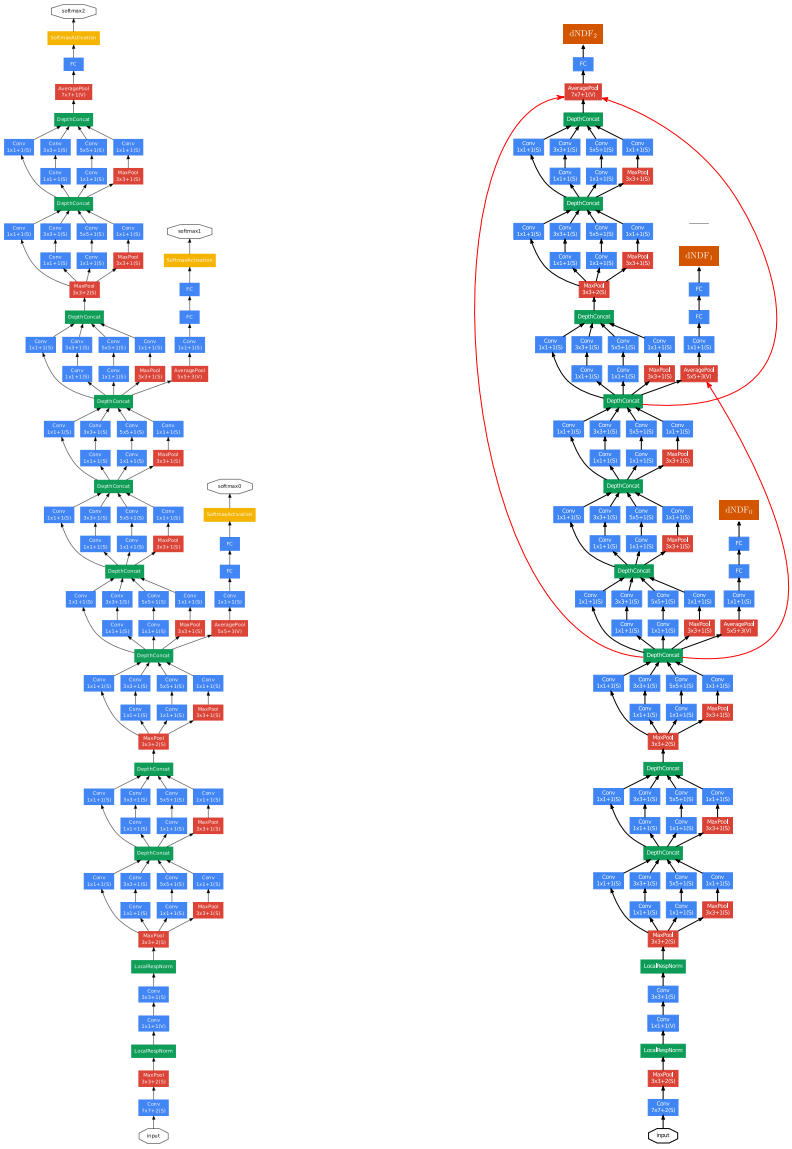

Figure 1. Left: Original GoogLeNet architecture proposed in [4]. Right: The modifications we brought to the GoogLeNet architecture resulting in dNDF.NET – our proposed model using dNDFs as final classifiers. Best viewed with digital zoom.

2. Split function gradient term derivations

Given the definitions for our split decision functions $d_n(\boldsymbol{x};\Theta)$ and the log-loss $L(\Theta,\boldsymbol{\pi};\boldsymbol{x},y)$ (see Equ. (3) and Equ. (6) in the main paper, respectively), we can derive the gradient term in Equ. (9) of the main paper as follows:

$$\begin{aligned}\frac{\partial L(\Theta,\boldsymbol{\pi};\boldsymbol{x},y)}{\partial f_n(\boldsymbol{x};\Theta)} &= \sum_{\ell\in\mathcal{L}} \frac{\partial L(\Theta,\boldsymbol{\pi};\boldsymbol{x},y)}{\partial \mu_\ell(\boldsymbol{x};\Theta)} \frac{\partial \mu_\ell(\boldsymbol{x};\Theta)}{\partial f_n(\boldsymbol{x};\Theta)} \\ &= -\sum_{\ell\in\mathcal{L}} \frac{\pi_{\ell y}}{\mathbb{P}_T[y|\boldsymbol{x},\Theta,\boldsymbol{\pi}]} \frac{\partial \mu_\ell(\boldsymbol{x};\Theta)}{\partial f_n(\boldsymbol{x};\Theta)} \\ &= -\sum_{\ell\in\mathcal{L}} \frac{\pi_{\ell y}\mu_\ell(\boldsymbol{x};\Theta)}{\mathbb{P}_T[y|\boldsymbol{x},\Theta,\boldsymbol{\pi}]} \frac{\partial \log \mu_\ell(\boldsymbol{x};\Theta)}{\partial f_n(\boldsymbol{x};\Theta)}\,,\end{aligned}$$

where

$$\begin{aligned}\frac{\partial \log \mu_\ell(\boldsymbol{x};\Theta)}{\partial f_n(\boldsymbol{x};\Theta)} &= \mathbb{1}_{\ell \swarrow n} \frac{\partial \log d_n(\boldsymbol{x};\Theta)}{\partial f_n(\boldsymbol{x};\Theta)} \\ &\quad + \mathbb{1}_{n \searrow \ell} \frac{\partial \log \bar{d}_n(\boldsymbol{x};\Theta)}{\partial f_n(\boldsymbol{x};\Theta)} \\ &= \mathbb{1}_{\ell \swarrow n} \bar{d}_n(\boldsymbol{x};\Theta) - \mathbb{1}_{n \searrow \ell} d_n(\boldsymbol{x};\Theta)\,.\end{aligned}$$

By substituting the latter in the previous formula we get

$$\begin{aligned}\frac{\partial L(\Theta,\boldsymbol{\pi};\boldsymbol{x},y)}{\partial f_n(\boldsymbol{x};\Theta)} &= -\sum_{\ell\in\mathcal{L}} \mathbb{1}_{\ell \swarrow n} \frac{\pi_{\ell y}\mu_\ell(\boldsymbol{x};\Theta)}{\mathbb{P}_T[y|\boldsymbol{x},\Theta,\boldsymbol{\pi}]} \bar{d}_n(\boldsymbol{x};\Theta) \\ &\quad + \sum_{\ell\in\mathcal{L}} \mathbb{1}_{n \searrow \ell} \frac{\pi_{\ell y}\mu_\ell(\boldsymbol{x};\Theta)}{\mathbb{P}_T[y|\boldsymbol{x},\Theta,\boldsymbol{\pi}]} d_n(\boldsymbol{x};\Theta) \\ &= -\sum_{\ell\in\mathcal{L}_{n_l}} \frac{\pi_{\ell y}\mu_\ell(\boldsymbol{x};\Theta)}{\mathbb{P}_T[y|\boldsymbol{x},\Theta,\boldsymbol{\pi}]} \bar{d}_n(\boldsymbol{x};\Theta) \\ &\quad + \sum_{\ell\in\mathcal{L}_{n_r}} \frac{\pi_{\ell y}\mu_\ell(\boldsymbol{x};\Theta)}{\mathbb{P}_T[y|\boldsymbol{x},\Theta,\boldsymbol{\pi}]} d_n(\boldsymbol{x};\Theta) \\ &= d_n(\boldsymbol{x};\Theta) A_{n_r} - \bar{d}_n(\boldsymbol{x};\Theta) A_{n_l}\,.\end{aligned}$$

3. Proof of update rule for $\boldsymbol{\pi}$

Theorem 1. *Consider a tree with parameters Θ and $\boldsymbol{\pi}$ and let*

$$\hat{\pi}_{\ell y} = \frac{1}{Z_\ell} \sum_{(\boldsymbol{x},y')\in\mathcal{T}} \mathbb{1}_{y=y'} \frac{\pi_{\ell y}\mu_\ell(\boldsymbol{x};\Theta)}{\mathbb{P}_T[y|\boldsymbol{x};\Theta,\boldsymbol{\pi}]}\,, \qquad \textit{for all } (\ell,y)\in\mathcal{L}\times\mathcal{Y}\,, \tag{12}$$

where Z_ℓ is the normalizing factor ensuring that $\hat{\boldsymbol{\pi}}_\ell = (\hat{\pi}_{\ell y})_{y\in\mathcal{Y}}$ is a probability distribution. In other terms, we assume $\hat{\pi}_{\ell y}$ to be the result of an update step as per (11) of our ICCV contribution. The following holds:

$$R(\Theta,\boldsymbol{\pi};\mathcal{T}) \geq R(\Theta,\hat{\boldsymbol{\pi}};\mathcal{T})$$

with equality if and only if $\hat{\boldsymbol{\pi}} = \boldsymbol{\pi}$, where R is the risk defined in (5) of our ICCV contribution, and $\boldsymbol{\pi} = (\boldsymbol{\pi}_\ell)_{\ell\in\mathcal{L}}$.

Proof. Consider the following auxiliary function:

$$\phi(\boldsymbol{\pi},\overline{\boldsymbol{\pi}}) = R(\Theta,\overline{\boldsymbol{\pi}};\mathcal{T}) - \frac{1}{|\mathcal{T}|} \sum_{(\boldsymbol{x},y)\in\mathcal{T}} \sum_{\ell\in\mathcal{L}} \xi_\ell(\overline{\boldsymbol{\pi}};\boldsymbol{x},y) \log\left(\frac{\pi_{\ell y}}{\overline{\pi}_{\ell y}}\right)\,,$$

where

$$\xi_\ell(\boldsymbol{\pi};\boldsymbol{x},y) = \frac{\pi_{\ell y}\mu_\ell(\boldsymbol{x};\Theta)}{\mathbb{P}_T[y|\boldsymbol{x};\Theta,\boldsymbol{\pi}]}\,.$$

and $\mathbb{P}_T[y|\boldsymbol{x},\Theta,\boldsymbol{\pi}]$ is defined as per (1) of our ICCV contribution. Note that $\phi(\boldsymbol{\pi},\boldsymbol{\pi}) = R(\Theta,\boldsymbol{\pi};\mathcal{T})$ holds for any $\boldsymbol{\pi}$, for the logarithm term in ϕ nullifies. Moreover, $\phi(\boldsymbol{\pi},\overline{\boldsymbol{\pi}}) \geq R(\Theta,\boldsymbol{\pi};\mathcal{T})$ holds for any $\boldsymbol{\pi}$ and $\overline{\boldsymbol{\pi}}$. This can be seen by applying

Jensen's inequality and with few algebraic manipulations:

$$
\begin{aligned}
R(\Theta, \boldsymbol{\pi}; \mathcal{T}) &= -\frac{1}{|\mathcal{T}|} \sum_{(\boldsymbol{x},y)\in\mathcal{T}} \log\left(\sum_{\ell\in\mathcal{L}} \pi_{\ell y}\mu_\ell(\boldsymbol{x};\Theta)\right) \\
&\leq -\frac{1}{|\mathcal{T}|} \sum_{(\boldsymbol{x},y)\in\mathcal{T}} \sum_{\ell\in\mathcal{L}} \xi_\ell(\overline{\boldsymbol{\pi}};\boldsymbol{x},y) \log\left(\frac{\pi_{\ell y}\mu_\ell(\boldsymbol{x};\Theta)}{\xi_\ell(\overline{\boldsymbol{\pi}};\boldsymbol{x},y)}\right) \qquad \text{(by Jensen's inequality)} \\
&= -\frac{1}{|\mathcal{T}|} \sum_{(\boldsymbol{x},y)\in\mathcal{T}} \sum_{\ell\in\mathcal{L}} \xi_\ell(\overline{\boldsymbol{\pi}};\boldsymbol{x},y) \left[\log\left(\frac{\pi_{\ell y}}{\overline{\pi}_{\ell y}}\right) + \log \mathbb{P}_T[y|\boldsymbol{x},\Theta,\overline{\boldsymbol{\pi}}]\right] \\
&= R(\Theta,\overline{\boldsymbol{\pi}};\mathcal{T}) - \frac{1}{|\mathcal{T}|} \sum_{(\boldsymbol{x},y)\in\mathcal{T}} \sum_{\ell\in\mathcal{L}} \xi_\ell(\overline{\boldsymbol{\pi}};\boldsymbol{x},y) \log\left(\frac{\pi_{\ell y}}{\overline{\pi}_{\ell y}}\right) = \phi(\boldsymbol{\pi},\overline{\boldsymbol{\pi}})\,.
\end{aligned}
$$

We can now show that $\hat{\boldsymbol{\pi}}$ is a global minimizer of $\phi(\cdot,\boldsymbol{\pi})$ for any value of $\boldsymbol{\pi}$. We start rewriting $\hat{\boldsymbol{\pi}}$ in terms of ξ_ℓ as follows:

$$
\hat{\pi}_{\ell y} = \frac{1}{Z_\ell} \sum_{(\boldsymbol{x},y')\in\mathcal{T}} \mathbb{1}_{y=y'} \xi_\ell(\boldsymbol{\pi};\boldsymbol{x},y)\,,
$$

where Z_ℓ is the normalizing factor. Then, we have that

$$
\begin{aligned}
&\phi(\hat{\boldsymbol{\pi}},\boldsymbol{\pi}) - \phi(\overline{\boldsymbol{\pi}},\boldsymbol{\pi}) \\
&\quad= -\frac{1}{|\mathcal{T}|} \sum_{(\boldsymbol{x},y)\in\mathcal{T}} \sum_{\ell\in\mathcal{L}} \xi_\ell(\boldsymbol{\pi};\boldsymbol{x},y) \log\left(\frac{\hat{\pi}_{\ell y}}{\overline{\pi}_{\ell y}}\right) \\
&\quad= -\frac{1}{|\mathcal{T}|} \sum_{\ell\in\mathcal{L}} \sum_{y\in\mathcal{Y}} \sum_{(\boldsymbol{x},y')\in\mathcal{T}} \mathbb{1}_{y=y'} \xi_\ell(\boldsymbol{\pi};\boldsymbol{x},y) \log\left(\frac{\hat{\pi}_{\ell y}}{\overline{\pi}_{\ell y}}\right) \\
&\quad= -\frac{1}{|\mathcal{T}|} \sum_{\ell\in\mathcal{L}} \sum_{y\in\mathcal{Y}} Z_\ell \hat{\pi}_{\ell y} \log\left(\frac{\hat{\pi}_{\ell y}}{\overline{\pi}_{\ell y}}\right) \\
&\quad= -\frac{1}{|\mathcal{T}|} \sum_{\ell\in\mathcal{L}} Z_\ell D_{KL}(\hat{\boldsymbol{\pi}}_\ell \| \overline{\boldsymbol{\pi}}_\ell) \leq 0\,,
\end{aligned}
$$

holds for all values of $\overline{\boldsymbol{\pi}}$, where $D_{KL}(\cdot\|\cdot)$ is the Kullback-Leibler divergence. Note that the last inequality yields equality if and only if $\hat{\boldsymbol{\pi}} = \overline{\boldsymbol{\pi}}$, for the Kullback-Leibler divergence yields zero if and only if the two distributions in input coincide. Accordingly, $\hat{\boldsymbol{\pi}}$ is a *strict* global minimizer of $\phi(\cdot,\boldsymbol{\pi})$ for any $\boldsymbol{\pi}$.

As a consequence of the previous derivations we have

$$
R(\Theta,\boldsymbol{\pi};\mathcal{T}) = \phi(\boldsymbol{\pi},\boldsymbol{\pi}) > \phi(\hat{\boldsymbol{\pi}},\boldsymbol{\pi}) \geq R(\Theta,\hat{\boldsymbol{\pi}};\mathcal{T})\,,
$$

where equality holds if and only if we have a fixed point of the update rule (12), *i.e.* if $\hat{\boldsymbol{\pi}} = \boldsymbol{\pi}$. □

References

[1] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*. USENIX Association, 2014. 1

[2] M. Li, L. Zhou, Z. Yang, A. Li, F. Xia, D. G. Andersen, and A. Smola. Parameter server for distributed machine learning. In *Big Learning NIPS Workshop*, 2013. 1

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2014. 1

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. 1, 2