2020 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)

Differentiable branching in deep networks for fast inference

Authors: S. Scardapane, D. Comminiello, M. Scarpiniti, E. Baccarelli, A. Uncini

SAPIENZA
UNIVERSITÀ DI ROMA

Introduction

Early exits in deep networks

Early exits

An early exit is an intermediate classification step inside a classical neural network (NN) architecture.

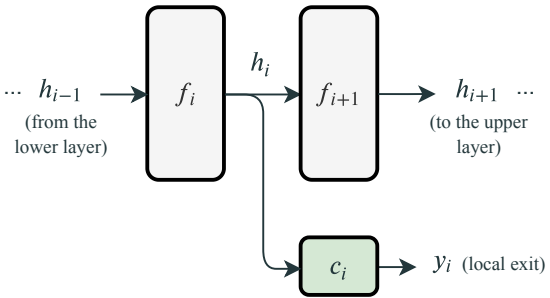


Figure 1: Graphical depiction of a generic early exit in neural network architectures. In green, we show the auxiliary predictor.

Why do we need early exits?

Early exits can be used for a variety of reasons:

1. They simplify optimization and gradient propagation (e.g., early Inception architectures);
2. They can perform faster inference if an input exits early on;
3. Possibility of distributing computation on multiple tiers of computation.[1]

We are especially interested in points 2 and 3.

[1] Zhang, C., Patras, P. and Haddadi, H., 2019. **Deep learning in mobile and wireless networking: A survey**. *IEEE Communications Surveys & Tutorials*, 21(3), pp. 2224-2287.

An example in a distributed system

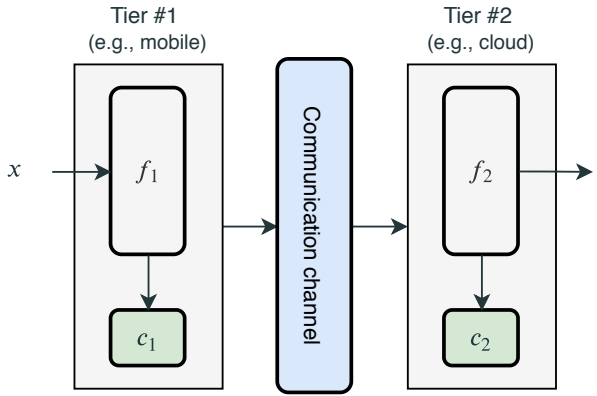


Figure 2: Distributed implementation of a multi-exit neural network on separate tiers of an underlying distributed computing platform.

Introduction

Contribution of the paper

Contribution of the paper

With many early exits, deciding **whether or not to exit** at a certain point is a **hard problem**.

Simple strategies (e.g., checking confidence score) requires careful fine-tuning *layer-by-layer* and are not scalable.

Contribution:
We propose a differentiable formulation to train the early exit strategy *together* with the main network.

Differentiable branching in deep networks

Description of the algorithm

Schema of the algorithm

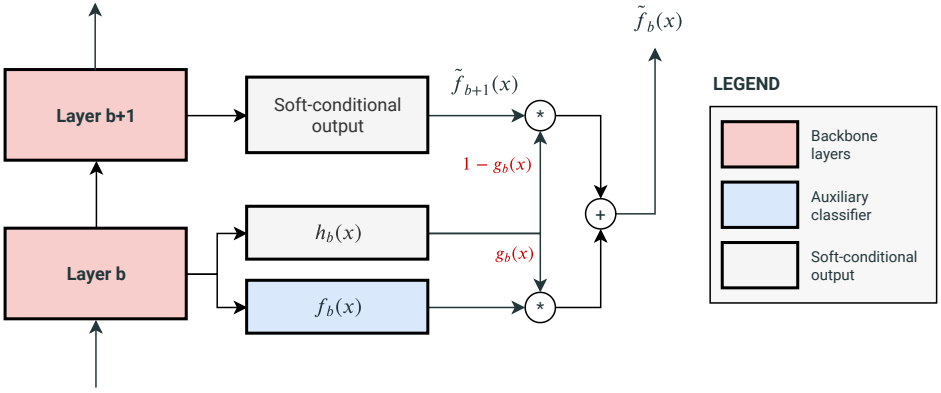


Figure 3: Schema of the proposed soft-conditional output. Red blocks are the main network, blue block is the additional auxiliary classifier. Note: the definition is recursive.

Mathematical formulation

Consider a neural network $f(x)$, endowed with early exits $f_1(x), \ldots, f_B(x)$.

Each auxiliary classifier is equipped with an additional early-stopping decision mechanism $g_b(x)$.

We define a *soft-conditional output* for the branch:

$$\widetilde{f}_b(x) = g_b(x) f_b(x) + (1 - g_b(x)) \widetilde{f}_{b+1}(x) \,. \tag{1}$$

The base case of the recursion is the final auxiliary classifier, for which $f_{B+1}(x) = f(x)$ is the standard output of the network and the confidence is always set to one: $h_{B+1}(x) = 1.0$.

Training and inference

Training the network:
We train the network with a cross-entropy loss on the last soft-conditional output.

Inference phase:

- Option 1: use the soft-conditional output as an ensemble strategy;
- Option 2: replace the soft-conditional early stopping strategy with a hard binary classifier.

Differentiable branching in deep networks

Regularizing for energy efficiency

Enhancing energy efficiency

Suppose that exiting at branch b has computational complexity γ_b (e.g., measured in number of elementary operations).

We penalize the average computational cost of the network as:

$$C = \frac{1}{N} \sum_{i=1}^{N} \gamma_{B+1}(x_i) \,, \tag{2}$$

where γ_{B+1} is recursively defined similar to (1) as:

$$\gamma_b(x_i) = g_b(x_i)\gamma_b + (1 - g_b(x_i))\gamma_{b+1}(x_i) \,.$$

Experimental results

Evaluating the algorithm

Results

Table 1: Test accuracy on the three datasets for the different architectures being compared. The best result for each scenario is highlighted in bold.

Dataset	AlexNet			VGG-13			ResNet-18		
	Baseline	B-NET	Prop.	Baseline	B-NET	Prop.	Baseline	B-NET	Prop.
CIFAR-10	73.21%	80.59%	**82.54%**	75.44%	82.88%	**87.11%**	81.61%	83.94%	**85.78%**
CIFAR-100	48.36%	51.18%	**56.11%**	47.18%	51.97%	**57.31%**	51.13%	54.78%	**58.54%**
CINIC-10	62.44%	67.99%	**70.12%**	63.41%	72.63%	**74.02%**	66.31%	70.02%	**74.37%**

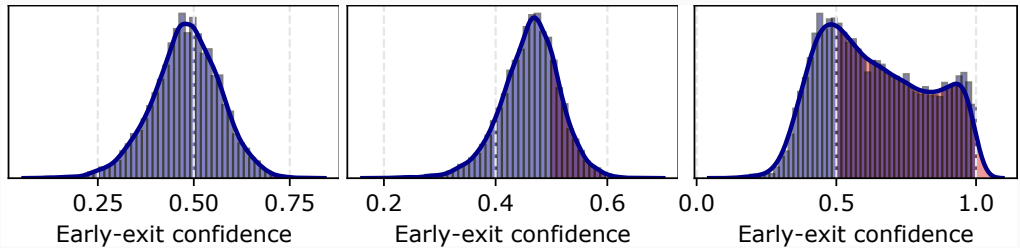
Teerapittayanon, S., McDanel, B. and Kung, H.T., 2016. **Branchynet: Fast inference via early exiting from deep neural networks**. In 2016 23rd ICPR (pp. 2464-2469). IEEE.

Results for the early-exit strategy

Table 2: Results of the proposed algorithm when enabling the early-exit strategy. Speed-up is relative to the baseline version. In the last column, $\#b$ is the percentage of inputs exiting at branch b.

Architecture	Test acc.	Inf. time	Speed-up	Exits
AlexNet	79.96%	2.74 ms	12%	#1: 1.3%, #4: 82.1%, #5: 12.3%
VGG-13	86.70%	1.59 ms	44%	#1: 5.1%, #6: 68.8%, #8: 20.3%, #9: 9.8%
ResNet-18	84.92%	2.18 ms	58%	#1: 0.4%, #2: 22.4%, #6: 1.6%, #7: 52.2%, #8: 13.5%

Distribution of the exits



(a) Branch #1 **(b)** Branch #2 **(c)** Branch #7

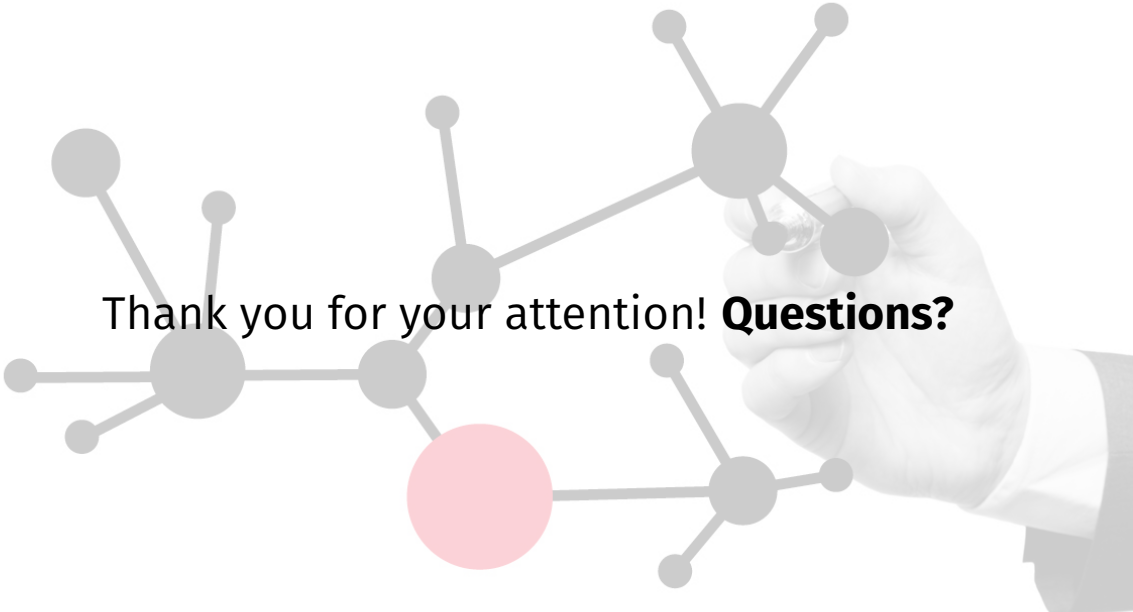
Figure 4: Early-exit confidence for 3 branches from the trained ResNet-18 architecture on CIFAR-10.

Conclusions

Conclusions and future work

Conclusions

- We proposed a new formulation for networks having multiple auxiliary classifiers that can be trained end-to-end with the main network.
- We also proposed a novel regularization term to trade-off accuracy and computational cost of inference.
- In future work we plan on further exploiting the differentiability of our formulation, by designing additional regularization terms and loss functions.
- We also plan on investigating more complex networks and different benchmarks beyond image classification.



Thank you for your attention! **Questions?**