N° 1671

Extracting Effective Subnetworks with Gumbel-Softmax

Robin Dupont[1,2], Mohammed Amine Alaoui[2], Hichem Sahbi[1], Alice Lebois[2]
[1]Sorbonne Université - LIP6, [2]Netatmo

Summary : We proposed a method to prune large untrained networks that yields lightweight networks with compelling performances, without tuning their weigths.

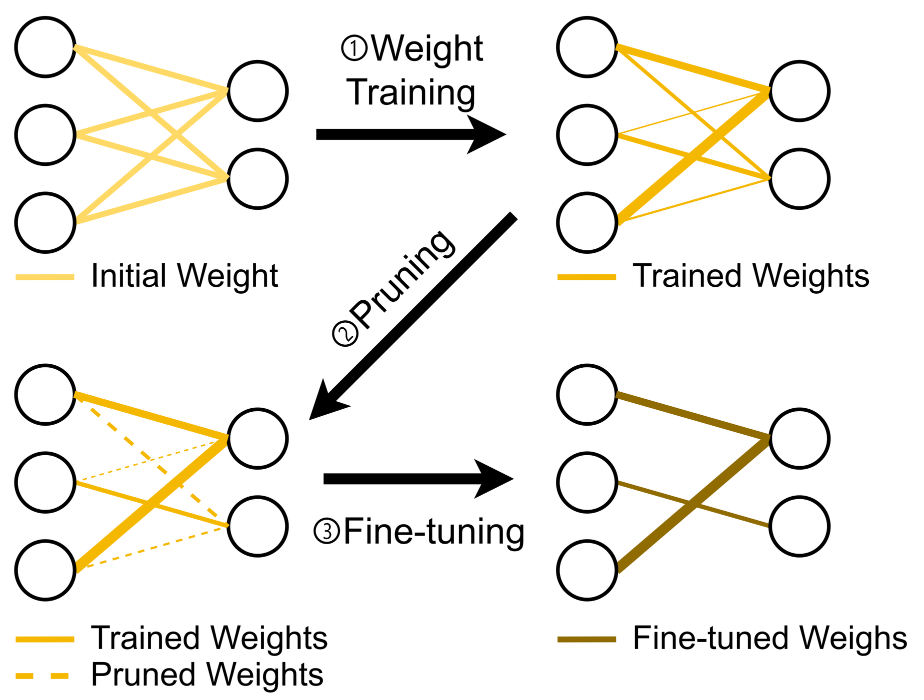
MOTIVATIONS

- Artificial neural networks are more and more present in **embedded applications and devices** such as mobile phones, autonomous cars, satellites and smart cameras for examples. These devices are **low on resources** compared to the powerful servers the neural networks are trained on.
- Developing **lightweight** neural networks is crucial to allows for the use of such algorithms on a wide variety of embedded applications and devices.

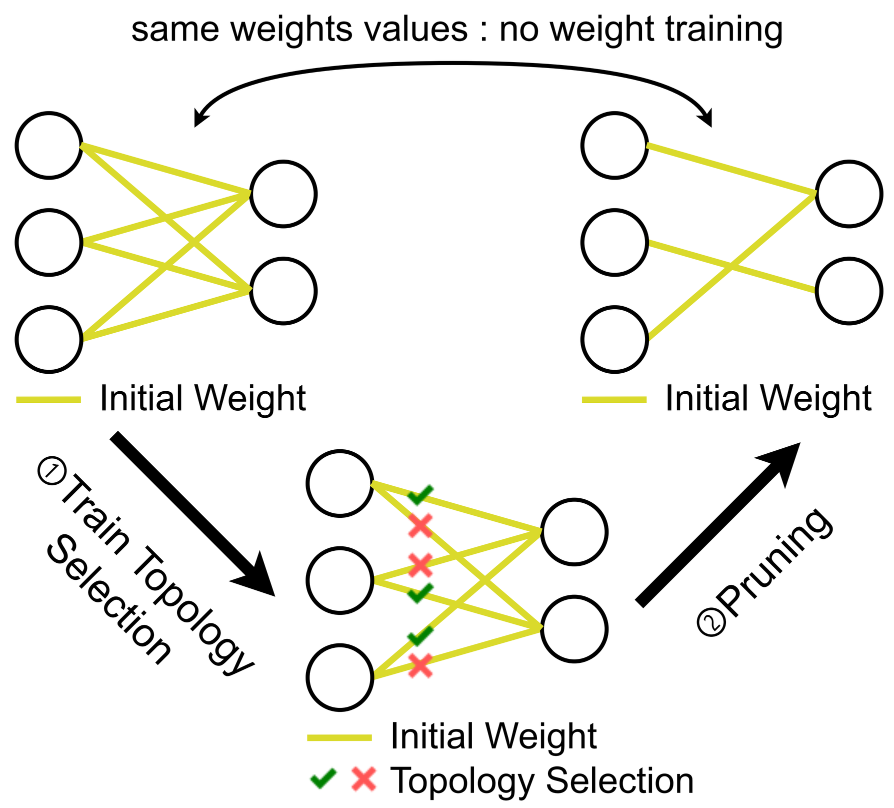
OVERVIEW

Our work aims to extract **lightweight yet efficient subnetworks** from large networks. It is achieved by **pruning** a large **untrained network**, **without any weight tuning.**

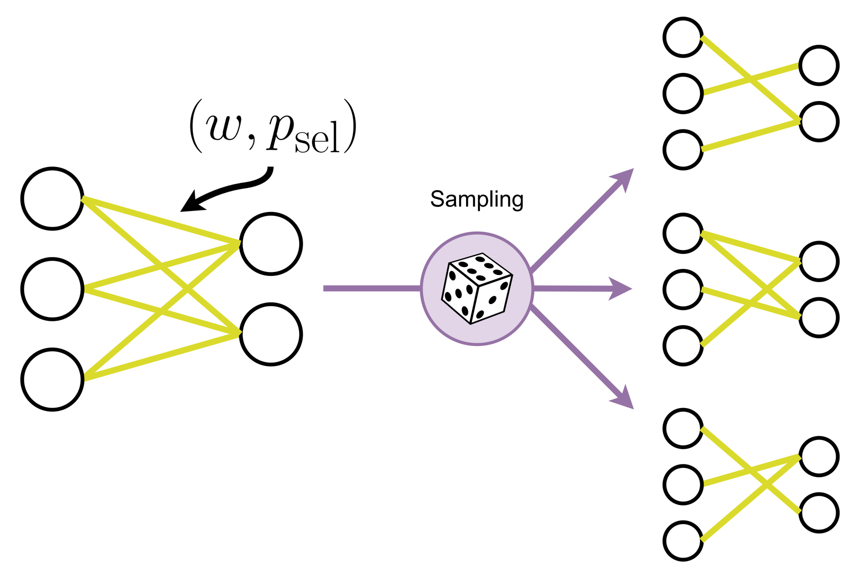
Standard pruning pipelines rely on a **train – prune – fine-tune** procedure. Weights are trained, then relevant weights are identified and selected and finally the remaining weights are fine-tuned.



Our pipeline does not require to train the network weights. We only perform **topology selection**. The pruned network exhibits compelling performances with **no weight training**.



OUR METHOD



ARBITRARILY SHIFTED LOG PARAMETRIZATION

Topology selection is achieved by **pruning** the original network. In practice the weight tensor $\boldsymbol{w}$ is multiplied by a binary mask tensor $\boldsymbol{m}$. Thus, a layer equation can be written :

$$\mathbf{z}_\ell = g_\ell((\boldsymbol{w}_\ell \odot \boldsymbol{m}_\ell) \otimes \mathbf{z}_{\ell-1})$$

Topology selection is stochastic. Each coefficient *m of the binary mask* **m** follows a Bernoulli distribution :

$$m \sim \mathcal{B}(p_{\text{sel}})$$

where p_{sel} is the probability for a weight of **being selected** in a **sampled topology**. Each weight has its corresponding p_{sel} and m.

Sampling is **not differentiable**, so we use Straight Through Gumbel Softmax[4] (STGS) and reparameterize p_{sel} as the sigmoid of $\hat{m}$ The sigmoid ensures that $0 \leq p_{\text{sel}} \leq 1$ and $\hat{m}$ is the learnt variable.

$$p_{\text{sel}} = \sigma(\hat{m})$$

STGS takes for argument the **log probabilities** of the two outcomes (*w* is selected or not).

Naive formulation :

$$m = \text{STGS}\left(\begin{bmatrix} \log(\sigma(\hat{m})) \\ \log(1-\sigma(\hat{m})) \end{bmatrix}\right)$$

computationally intensive   numerical instabilities

Our formulation ASLP :

$$m = \text{STGS}\left(\begin{bmatrix} \hat{m} \\ 0 \end{bmatrix}\right)$$

not computationally intensive   numerically stable

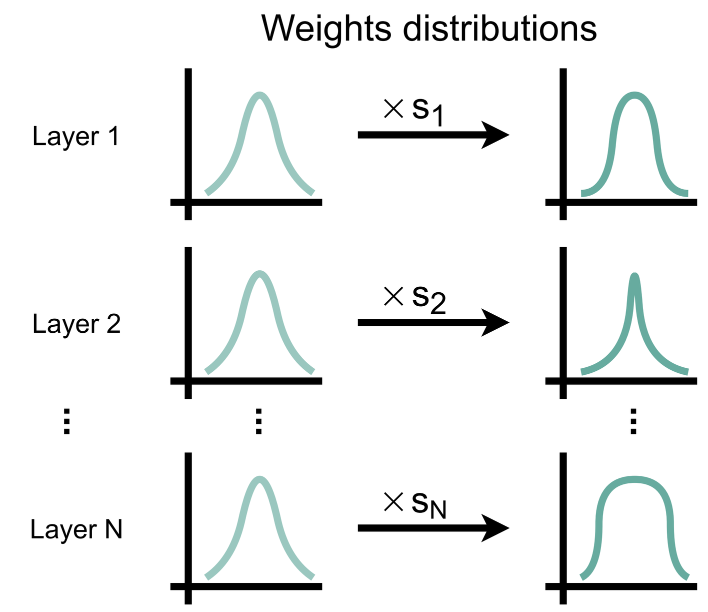
Interpreting our formulation in terms of probabilities lead to the same reparameterization.

arbitrary unknown constant that shift log-probabilities

$$\begin{bmatrix} \hat{m} \\ 0 \end{bmatrix} = \begin{bmatrix} \log(p_{\text{sel}}) + c \\ \log(1 - p_{\text{sel}}) + c \end{bmatrix} \implies p_{\text{sel}} = \sigma(\hat{m})$$

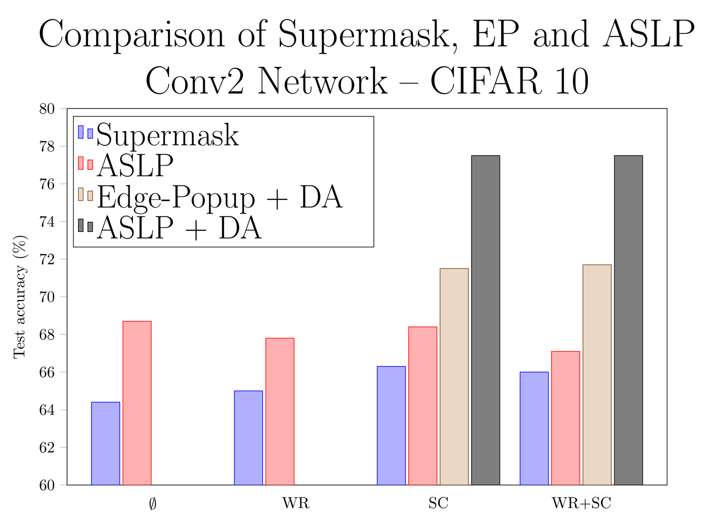
SMART RESCALE

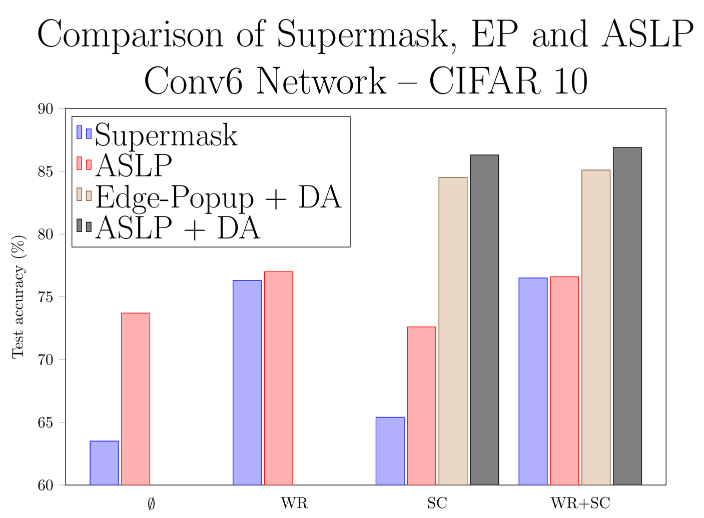
Smart Rescale is a learnt dynamic scaling factor that **rescales** weight distribution on a **per-layer** basis. This compensates for the **shift in variance** in the distribution due to **pruning**.



RESULTS

Our method is compared against **Edge-Popup**[1] and **Supermask**[2] methods on **CIFAR10** and **CIFAR100** datasets. We used **Conv2, Conv4 and Conv6**[3] networks. For CIFAR10, results are shown with and without Data Augmentation (DA).





WR = Weight Rescale, SC = Signed Constant[1]

	Conv2	Conv4	Conv6
EP	40.9	51.1	**53.2**
ASLP	**43.4**	**51.7**	52.8

Table 1: Edge Popup and ASLP on CIFAR100

CONCLUSION

Our method extracts networks from large untrained ones by identifying and removing the least significant weights. We introduced **Arbitrarily Shifted Log Parametrization** (ASLP) which **reduces computational** cost and **prevents numerical instabilities** of Gumbel-Softmax, as well as **Smart Rescale** which **improves performances.**

REFERENCES

[1] V. Ramanujan, *et al.* "What's hidden in a randomly weighted neural network?," in CVPR, 2020
[2] H. Zhou, *et al.* "Deconstructing lottery tickets: Zeros, signs, and the supermask," in NeurIPS, 2019
[3] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in ICLR, 2019
[4] E. Jang, *et al.* "Categorical reparameterization with gumbel-softmax," in ICLR, 2017