Knowledge Evolution FAQ

Ahmed Taha Abhinav Shrivastava Larry Davis
University of Maryland, College Park

How is Knowledge Evolution (KE) different from Dropout?
After training, dropout delivers a dense network, while KE delivers a slim network with a smaller inference cost.

How is KE different from pruning approaches?
Pruning approaches either introduce loss terms or require certain layers [4, 3]. In contrast, KE supports standard network architectures and leverages vanilla loss functions, *e.g.*, cross-entropy.

What happens if we train a Sub-ResNet18 equivalent in size to the fit-hypothesis $H^{\triangle}$?
Fig. 1 presents the performance of a ResNet18 on Flower-102 (1020-samples dataset). In this experiment, We train a ResNet18 for one generation, a Sub-ResNet18 for one generation, and a ResNet18 for 100 generations. The Sub-ResNet18 is equivalent to the fit-hypothesis for a given split-rate. While ResNet18 has approximately 11 million parameters, Sub-ResNet18 has $\approx 11 * 0.8^2 \approx 7$ million parameters with split-rate $s_r = 0.8$ and $\approx 11 * 0.5^2 \approx 3$ million parameters with $s_r = 0.5$. Both ResNet18 and Sub-ResNet18 degenerate on Flower-102 which indicates overfitting. Even with $s_r = 0.5$, Sub-ResNet18 overfits on Flower-102 because 3 million parameters are huge for a 1020-samples dataset. In contrast, KE mitigates overfitting and achieves an absolute 21% improvement margin.

How does KE compare with Meta-Learning?
KE can be regarded as a simple meta-learning approach without bells and whistles. As the number of generations increases, the fit-hypothesis $H^{\triangle}$ becomes better initialized and closer to convergence. Accordingly, KE is a better initialization method compared to dataset-oblivious initialization methods, *e.g.*, Kaiming uniform [2].

What happens if the split-mask M changes across generations?
We evaluate this scenario in the paper appendix; please check WELS vs. WELS-Rand experiment (Fig. 16). The overhead of changing the split-mask – across generations – is never justified.

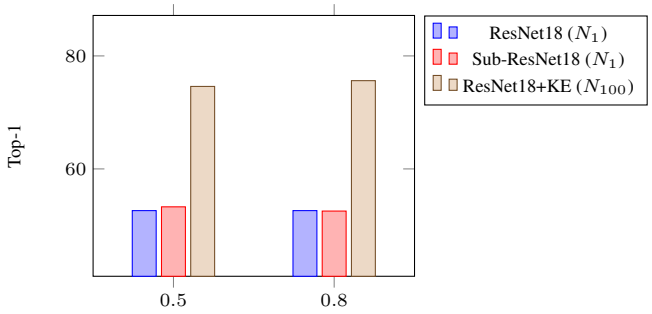

Figure 1. Quantitative evaluation using Flower102 and a randomly initialized ResNet18. The x-axis denotes the split-rate s_r while the y-axis denotes the Top-1 accuracy. Sub-ResNet18 denotes a sub-network of ResNet18 equivalent in size to the fit-hypothesis $H^{\triangle}$. ResNet18+KE is superior to both ResNet18 and Sub-ResNet18.

How do you pick the split-rate s_r?
In our experiments, we evaluate KE against DSD [1] and RePr [5]. These baselines have a prune-rate hyperparameter with a default value of $p = 0.3$. Accordingly, we set $s_r = 0.8$ such that our prune-rate $p \approx 1 - 0.8^2 = 0.36$ is close to the aforementioned baselines. In the ablation studies, Fig. 9 highlights the trade-offs of small and large split-rates.

References

[1] Song Han, Jeff Pool, Sharan Narang, Huizi Mao, Enhao Gong, Shijian Tang, Erich Elsen, Peter Vajda, Manohar Paluri, John Tran, et al. Dsd: Dense-sparse-dense training for deep neural networks. *arXiv preprint arXiv:1607.04381*, 2016.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *ICCV*, 2015.

[3] Gao Huang, Shichen Liu, Laurens Van der Maaten, and Kilian Q Weinberger. Condensenet: An efficient densenet using learned group convolutions. In *CVPR*, 2018.

[4] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *ICCV*, 2017.

[5] Aaditya Prakash, James Storer, Dinei Florencio, and Cha Zhang. Repr: Improved training of convolutional filters. In *CVPR*, 2019.