Supplementary Material of
ShiftAddAug: Augment Multiplication-Free Tiny Neural Network with Hybrid Computation

Yipin Guo, Zihao Li, Yilin Lang, Qinyuan Ren
College of Control Science and Engineering, Zhejiang University
{guoyipin, lzh_jeong, langyilin, renqinyuan}@zju.edu.cn

1. About knowledge distillation

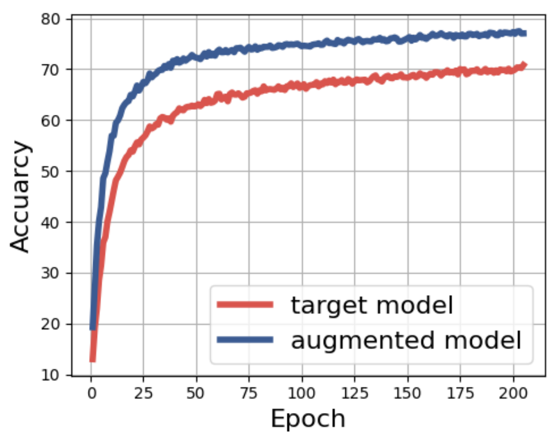

Figure 1. Training curve of MobileNetV2-w0.35 with ShiftAddAug.

As can be seen in Fig. 1, throughout the training process, the augmented model will have higher accuracy due to the larger capacity. It is a natural idea to use knowledge distillation to further improve the performance of the target model. Inplace Distillation[6] looks perfect for our situation. But in fact, it don't work very well.

Table 1. ShiftAddAug results on MobileNetV2 with knowledge distillation.

model	criterion				origin
	KLLoss		CELoss		
	$\alpha = 0.9$	$\alpha = 0.3$	$\alpha = 0.9$	$\alpha = 0.3$	
MobileNetV2 - w0.35	65.7	68.93	64.89	69.02	71.83

Inplace Distillation expects small models to gain more supervision from the soft labels of large models. It learns correct information while also learning biases in large models. Due to weight sharing, The large model and the small model in the same training step may exhibit similar biases. This problem was not obvious in previous work. But multiplication-free operators are more unstable during training, making this problem serious in our case.

2. Discussion about AddConv

In order to obtain a smaller model, using depthwise separable convolution with InvertedBlock[4] is a must. But AdderNet's [2] implementation only works with ordinary convolutions. It will be slow and unstable in DWConv. So we keep DWConv as multiplication and convert the other parts to AddConv. But this still causes a loss of stability because the original AdderNet retains some multiplicative convolutions in the input and classification heads for higher accuracy. As you can see in Tab. 2, even though our method can boost accuracy compared with direct training, the result is still not ideal. This is a problem with AddConv itself.

However, in the experiment of neural architecture search, keeping AddConv in the first few layers of the model helps improve accuracy. We keep the first 3 convolutions as AddConv instead of the original Conv, obtaining **0.43%** accuracy increase and some energy savings.

3. Training cost

The purpose of ShiftAddAug is to improve the accuracy of the multiplication-free model without generating any inference overhead. However, since we use additional multiplication structures to assist training, this will consume more resources during training. This is consistent with NetAug[1].

In order to present the training overhead in detail, we compare the training resource consumption of our method with directly trained multiplicative, shift[3], and add[2] models, as well as NetAug[1] and ShiftAddNet[5].

We use MobileNetV2-w0.35 as the basic model structure, input resolution=160, batch size=32. Results were evaluated on NVIDIA GTX 3090. The training speed and memory usage are shown in Tab. 3

Table 2. Accuracy of MobileNetV2-w0.35 / MCUNet with AddConv.

Model	Methods	CIFAR10	CIFAR100	Food101	Flower102
MobileNetV2 - w0.35	Shift	88.59	69.45	72.99	92.25
	Add	85.76	67.85	67.89	75.78
	AugAdd	87.21	69.38	68.62	78.92
MCUNet	Shift	90.61	70.87	78.46	95.59
	Add	89.38	70.25	70.6	78.63
	AugAdd	91.02	72.72	72.04	84.33

Table 3. The training cost comparison in the term of speed and memory usage.

Method	Speed(it/s)	Memory(MB)
Base	21.06	2325
Shift	15.21	2349
Add	11.70	3215
ShiftAddNet	1.09	3697
NetAug	8.69	6945
AugShift	4.39	7195
AugAdd	3.14	7225

References

[1] Han Cai, Chuang Gan, Ji Lin, and song han. Network augmentation for tiny deep learning. In *International Conference on Learning Representations*, 2022. 1

[2] Hanting Chen, Yunhe Wang, Chunjing Xu, Boxin Shi, Chao Xu, Qi Tian, and Chang Xu. Addernet: Do we really need multiplications in deep learning? In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1465–1474, 2020. 1

[3] Mostafa Elhoushi, Zihao Chen, Farhan Shafiq, Ye Henry Tian, and Joey Yiwei Li. Deepshift: Towards multiplication-less neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2359–2368, 2021. 1

[4] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 4510–4520, 2018. 1

[5] Haoran You, Xiaohan Chen, Yongan Zhang, Chaojian Li, Sicheng Li, Zihao Liu, Zhangyang Wang, and Yingyan Lin. Shiftaddnet: A hardware-inspired deep network. In *Advances in Neural Information Processing Systems*, pages 2771–2783. Curran Associates, Inc., 2020. 1

[6] Jiahui Yu and Thomas Huang. Universally slimmable networks and improved training techniques. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1803–1811, 2019. 1