BlockDrop: Dynamic Inference Paths in Residual Networks (Supplemental Material)

Zuxuan Wu[1*], Tushar Nagarajan[2*], Abhishek Kumar[3], Steven Rennie[4]
Larry S. Davis[1], Kristen Grauman[2], Rogerio Feris[3]
[1] UMD, [2] UT Austin, [3] IBM Research, [4] Fusemachines Inc.

Details of BlockDrop-seq (Ours-seq)

We construct a sequential version of BlockDrop for dropping blocks, where the decision $\mathbf{a}_i \in \{0, 1\}$ to drop or keep the i-th block is conditioned on the activations of its previous block, y_{i-1}. Unlike BlockDrop, where all the actions are predicted in one shot, this model predicts one action at a time, which is a typical reinforcement learning setting. We follow the procedure to generate the *halting scores* in [1], and arrive at an equivalent per-block *skipping score* according to:

$$\mathbf{p}_i = \texttt{softmax}(\widetilde{W}^i \texttt{pool}(y_{i-1}) + b^i),$$

where `pool` is a global average pooling operation. For fair comparisons, Ours-seq is compared to a BlockDrop model, which attains equivalent accuracy, with the same number of blocks.

Implementation Details

- On CIFAR, we train the model for 5000 epochs during curriculum learning with a batch size of 2048 and a learning rate of $1e-4$. We further jointly finetune the model for 1600 epochs with a batch size of 256 and a learning rate of $1e-4$, which is annealed to $1e-5$ for 400 epochs.
- On ImageNet, the policy network is trained for 45 epochs for curriculum learning with a batch size of 2048 and a learning rate of $1e-4$. We then use a batch size of 320 during joint finetuning for 10 epochs.

Detailed Results on CIFAR-10 and ImageNet

We present detailed results of our method on CIFAR-10 (Table 1) and ImageNet (Table 2). We highlight the accuracy, block usage and speed up for variants of our model compared to full ResNets.

Network	FLOPs	Block Usage	Accuracy	Speed-up
ResNet-32	1.38E+08 $\pm$ 0.00E+00	15.0 $\pm$ 0.0	92.3	–
ResNet-110	5.06E+08 $\pm$ 0.00E+00	54.0 $\pm$ 0.0	93.2	–
BlockDrop-32 ($\gamma = 5$)	8.66E+07 $\pm$ 1.40E+07	6.9 $\pm$ 1.6	91.3	37.2%
BlockDrop-110 ($\gamma = 2$)	1.18E+08 $\pm$ 2.46E+07	10.3 $\pm$ 2.7	91.9	76.7%
BlockDrop-110 ($\gamma = 5$)	1.51E+08 $\pm$ 3.24E+07	13.8 $\pm$ 3.5	93.0	70.1%
BlockDrop-110 ($\gamma = 10$)	1.81E+08 $\pm$ 3.43E+07	16.9 $\pm$ 3.7	93.6	64.3%

Table 1: Results of different architectures on CIFAR-10. Depending on the base ResNet architecture, speedups ranging from 37% to 76% are observed with little to no degradation in performance.

[*] Authors contributed equally

Network	FLOPs	Block Usage	Accuracy	Speed-up
ResNet-72	1.17E+10 $\pm$ 0.00E+00	24.0 $\pm$ 0.0	75.8	–
ResNet-75	1.21E+10 $\pm$ 0.00E+00	25.0 $\pm$ 0.0	75.9	–
ResNet-84	1.34E+10 $\pm$ 0.00E+00	28.0 $\pm$ 0.0	76.1	–
ResNet-101	1.56E+10 $\pm$ 0.00E+00	33.0 $\pm$ 0.0	76.4	–
BlockDrop ($\gamma = 2$)	9.85E+09 $\pm$ 3.34E+08	18.8 $\pm$ 0.8	75.2	36.9%
BlockDrop ($\gamma = 5$)	1.25E+10 $\pm$ 4.26E+08	24.8 $\pm$ 1.0	76.4	19.9%
BlockDrop ($\gamma = 10$)	1.47E+10 $\pm$ 4.02E+08	29.7 $\pm$ 0.9	76.8	5.7%

Table 2: Results of different architectures on ImageNet. BlockDrop is built upon ResNet-101, and can achieve around 20% speedup on average with $\gamma = 5$.

References

[1] M. Figurnov, M. D. Collins, Y. Zhu, L. Zhang, J. Huang, D. Vetrov, and R. Salakhutdinov. Spatially adaptive computation time for residual networks. In *CVPR*, 2017. 1