Rethinking Online Knowledge Distillation with Multi-Exits

Hojung Lee[0000−0001−9312−3904] and Jong-Seok Lee[0000−0002−8038−1119]

School of Integrated Technology, Yonsei University, Incheon 21983, South Korea
{hjlee92,jong-seok.lee}@yonsei.ac.kr

A Supplementary Material

We provide the implementation details including the structures of the networks and the training hyperparameters used for the experiments. In addition, we provide additional experimental results that are not included in the paper due to lack of space. Code is available at https://github.com/hjdw2/BEED.

A.1 Network

The network architectures for CIFAR-100 are described in Figure 1. For ResNet [1], we modify the first convolution layer: kernel size ($7\times7 \rightarrow 3\times3$), stride ($2 \rightarrow 1$), and padding ($3 \rightarrow 1$). In addtion, we remove the max-pooling operation as in [3]. For WideResNet (WRN) [8], we do not use the dropout operation. For ResNet and WRN, the numbers of channels in the convolutional layers in the exits is the same as the numbers of channels of the corresponding residual blocks in the main network. For MobileNet-V2 [6] and EfficientNetB0 [7], the numbers of channels in the convolutional layers in the exits is the numbers of channels of the input feature multiplied by the channel expansion. However, the numbers of channels in the last convolutional layers in the exits is the same as the numbers of channels of the last convolution layer in the main network in order to match the feature dimension for feature distillation. For MSDNet [2], we set the number of total exits to four and other implementation details are the same as in the original work. For ImageNet, there is no modification in the main network and we use the same exits used for CIFAR-100.

A.2 Implementation detatils

For CIFAR-100, the batch size is set to 128 and the maximum training epoch is set to 200. We use the stochastic gradient descent (SGD) with a momentum of 0.9 and an initial learning rate of 0.1, and the learning rate decreases by an order of magnitude after 75, 130, and 180 epochs. The L2 regularization is used with a fixed constant of 5×10^{-4}.

For ImageNet, the batch size is set to 256 and the maximum training epoch is set to 90. We use the SGD with a momentum of 0.9 and an initial learning rate of 0.1, and the learning rate decreases by an order of magnitude after 30 and 60 epochs. The L2 regularization is used with a fixed constant of 1×10^{-4}.

The temperature (T) is set to 1 for classification losses and 3 for distillation losses.

All experiments are performed using Pytorch with NVIDIA RTX 8000 graphics processing units (GPUs).

A.3 Ablation study

Figure 2 shows how the performance changes with respect to the values of the hyper-parameters. Since α and β are set according to the typical setting used in other works [4, 5, 9], we vary the importance coefficient (λ) and balancing constant (γ). The red circle indicates the case reported in our paper. The performance remains similarly satisfactory.

A.4 Experiments

We show additional experimental results for Section 3 and Section 6.

Table 1 shows the test accuracy of each main network for various combinations of exit structures and training methods. When our BEED is used, the proposed bottleneck exit structure outperforms the other exit structures even for MobileNet-V2 and EfficientNetB0, as mentioned in Section 3.

Table 2 shows the test accuracy of all classifiers (i.e., all exits, main network, and ensemble classifier) for different training methods. When the ensemble classification performance is compared, our method outperforms both CE and KD, as mentioned in Section 6.2. In addition, our method shows improved performance not only with the main network but also with the exits.

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778. Las Vegas, Nevada (2016)
2. Huang, G., Chen, D., Li, T., Wu, F., v. d. Maaten, L., Weinberger, K.Q.: Multi-scale dense networks for resource efficient image classification. In: Proceedings of the International Conference on Learning Representations (ICLR) (2018)
3. Ji, M., Shin, S., Hwang, S., Park, G., Moon, I.C.: Refine myself by teaching myself: Feature refinement via self-knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10659–10668 (2021)
4. Li, H., Zhang, H., Qi, X., Ruigang, Y., Huang, G.: Improved techniques for training adaptive deep networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1891–1900 (2019)
5. Phuong, M., Lampert, C.: Distillation-based training for multi-exit architectures. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1355–1364 (2019)

6. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4510–4520 (2018)
7. Tan, M., Le, Q.: EfficientNet: Rethinking model scaling for convolutional neural networks. In: Proceedings of the International Conference on Machine Learning (ICML). vol. 97, pp. 6105–6114 (2019)
8. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: Proceedings of the British Machine Vision Conference (BMVC) (2016)
9. Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., Ma, K.: Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3712–3721 (2019)

ResNet18

Operator	Channel	Repeat	Stride
Conv2d	64	1	1
ResNet Block	64	2	1
ResNet Block	128	2	2
ResNet Block	256	2	2
ResNet Block	512	2	2
Avgpool 4x4 FC layer	-	-	-

Exit1

Operator	Channel	Repeat	Stride
Bottleneck Block	128	1	2
Bottleneck Block	256	1	2
Bottleneck Block	512	1	2
Avgpool 4x4 FC layer	-	-	-

Exit2

Operator	Channel	Repeat	Stride
Bottleneck Block	256	1	2
Bottleneck Block	512	1	2
Avgpool 4x4 FC layer	-	-	-

Exit3

Operator	Channel	Repeat	Stride
Bottleneck Block	512	1	2
Avgpool 4x4 FC layer	-	-	-

ResNet34

Operator	Channel	Repeat	Stride
Conv2d	64	1	1
ResNet Block	64	3	1
ResNet Block	128	4	2
ResNet Block	256	6	2
ResNet Block	512	3	2
Avgpool 4x4 FC layer	-	-	-

Exit1

Operator	Channel	Repeat	Stride
Bottleneck Block	128	1	2
Bottleneck Block	256	1	2
Bottleneck Block	512	1	2
Avgpool 4x4 FC layer	-	-	-

Exit2

Operator	Channel	Repeat	Stride
Bottleneck Block	256	1	2
Bottleneck Block	512	1	2
Avgpool 4x4 FC layer	-	-	-

Exit3

Operator	Channel	Repeat	Stride
Bottleneck Block	512	1	2
Avgpool 4x4 FC layer	-	-	-

WRN16-4

Operator	Channel	Repeat	Stride	Expansion
Conv2d	16	1	1	-
WRN Block	16	2	1	4
WRN Block	32	2	2	4
WRN Block	64	2	2	4
Avgpool 8x8 FC layer	-	-	-	

Exit1

Operator	Channel	Repeat	Stride
Bottleneck Block	128	1	2
Bottleneck Block	256	1	2
Avgpool 8x8 FC layer	-	-	-

Exit2

Operator	Channel	Repeat	Stride
Bottleneck Block	256	1	2
Avgpool 8x8 FC layer	-	-	-

WRN28-4

Operator	Channel	Repeat	Stride	Expansion
Conv2d	16	1	1	-
WRN Block	16	4	1	4
WRN Block	32	4	2	4
WRN Block	64	4	2	4
Avgpool 8x8 FC layer	-	-	-	

Exit1

Operator	Channel	Repeat	Stride
Bottleneck Block	128	1	2
Bottleneck Block	256	1	2
Avgpool 8x8 FC layer	-	-	-

Exit2

Operator	Channel	Repeat	Stride
Bottleneck Block	256	1	2
Avgpool 8x8 FC layer	-	-	-

MobileNet-V2

Operator	Channel	Repeat	Stride	Expansion
Conv2d	32	1	1	-
MobileNet Block	16	1	1	1
MobileNet Block	24	2	1	6
MobileNet Block	32	3	2	6
MobileNet Block	64	4	2	6
MobileNet Block	96	3	1	6
MobileNet Block	160	3	2	6
MobileNet Block	320	1	1	6
Conv2d 1x1	1280	1	1	-
Avgpool 4x4 FC layer	-	-	-	

Exit1

Operator	Channel	Repeat	Stride
Bottleneck Block	144	1	2
Bottleneck Block	192	1	2
Bottleneck Block	320	1	2
Avgpool 4x4 FC layer	-	-	-

Exit2

Operator	Channel	Repeat	Stride
Bottleneck Block	192	1	2
Bottleneck Block	320	1	2
Avgpool 4x4 FC layer	-	-	-

Exit3

Operator	Channel	Repeat	Stride
Bottleneck Block	320	1	2
Avgpool 4x4 FC layer	-	-	-

EfficientNetB0

Operator	Channel	Repeat	Stride	Expansion
Conv2d	32	1	1	-
EfficientNet Block	16	1	1	1
EfficientNet Block	24	2	1	6
EfficientNet Block	60	2	2	6
EfficientNet Block	80	3	2	6
EfficientNet Block	112	3	1	6
EfficientNet Block	192	4	2	6
EfficientNet Block	320	1	1	6
Avgpool 4x4 FC layer	-	-	-	

Exit1

Operator	Channel	Repeat	Stride
Bottleneck Block	144	1	2
Bottleneck Block	240	1	2
Bottleneck Block	320	1	2
Avgpool 4x4 FC layer	-	-	-

Exit2

Operator	Channel	Repeat	Stride
Bottleneck Block	240	1	2
Bottleneck Block	320	1	2
Avgpool 4x4 FC layer	-	-	-

Exit3

Operator	Channel	Repeat	Stride
Bottleneck Block	320	1	2
Avgpool 4x4 FC layer	-	-	-

Fig. 1. Description of the main networks and exits for CIFAR-100. The lines in the main network structure correspond to the locations where the exits are attached.

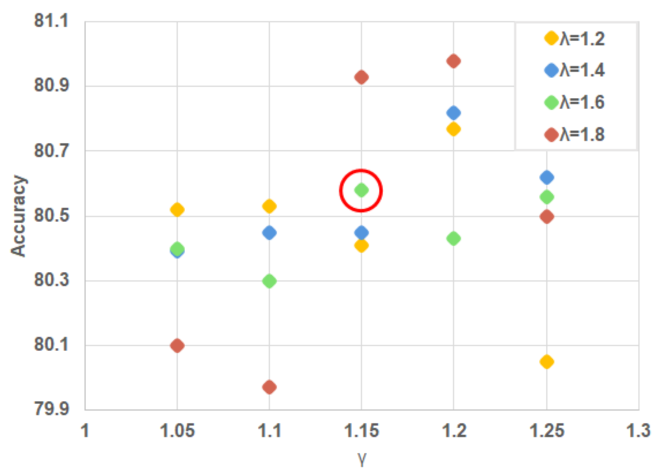

Fig. 2. Test accuracy (%) of ResNet18 trained for CIFAR-100 with varying hyper-parameters.

Table 1. Test accuracy (%) of the main network trained by different training methods with different exit structures for CIFAR-100. The best results are marked in bold. The column 'CE' corresponds to Table 1 of the main paper.

Network	Exit structure	CE	KD	EED	BEED
ResNet18	LCT	78.22	78.17	78.81	78.95
	BYOT	78.51	78.47	78.38	79.23
	TOFD	79.01	78.97	79.61	80.43
	Proposed bottleneck	**79.39**	**79.21**	**80.03**	**80.58**
ResNet34	LCT	79.80	79.40	80.65	81.07
	BYOT	79.41	79.40	79.98	80.60
	TOFD	79.92	79.65	81.16	81.17
	Proposed bottleneck	**80.22**	**80.17**	**81.61**	**81.62**
WRN16-4	LCT	75.71	75.92	75.66	75.62
	BYOT	76.61	76.30	75.58	75.75
	TOFD	76.81	77.11	77.06	76.96
	Proposed bottleneck	**77.49**	**77.75**	**78.26**	**78.51**
WRN28-4	LCT	78.19	77.81	77.84	77.51
	BYOT	78.56	78.58	77.77	78.48
	TOFD	79.02	78.89	78.63	79.25
	Proposed bottleneck	**80.05**	**80.06**	**80.55**	**80.93**
MobileNet-V2	LCT	**74.31**	73.98	76.49	76.23
	BYOT	74.10	**74.04**	75.32	75.91
	TOFD	73.51	73.61	76.61	76.47
	Proposed bottleneck	73.73	73.68	**76.63**	**76.74**
EfficientNetB0	LCT	**75.01**	74.90	77.21	77.22
	BYOT	74.51	**75.02**	75.39	76.63
	TOFD	74.38	74.77	77.28	77.07
	Proposed bottleneck	74.21	74.44	**77.69**	**77.62**

Table 2. Test accuracy (%) of the main network, exits, and ensemble for CIFAR-100. The best results are marked in bold.

Network	Method	Exit1	Exit2	Exit3	Main	Ens.
ResNet18	Baseline	-	-	-	77.60	-
	CE	74.44	76.68	78.48	79.39	80.68
	KD	75.16	77.45	78.81	79.21	80.93
	EED	77.38	79.00	79.93	80.03	81.00
	BEED	**77.55**	**79.42**	**80.33**	**80.58**	**81.45**
ResNet34	Baseline	-	-	-	77.96	-
	CE	75.04	78.25	79.72	80.22	81.70
	KD	75.49	79.15	80.05	80.17	81.89
	EED	77.67	80.14	81.38	81.61	82.45
	BEED	**77.93**	**80.19**	**81.43**	**81.62**	**82.50**
WRN16-4	Baseline	-	-	-	76.38	-
	CE	70.23	74.71	-	77.49	78.47
	KD	71.28	75.67	-	77.75	78.41
	EED	74.24	76.92	-	78.26	78.36
	BEED	**74.71**	**77.00**	-	**78.51**	**78.69**
WRN28-4	Baseline	-	-	-	78.64	-
	CE	73.33	77.64	-	80.05	81.17
	KD	74.28	78.58	-	80.06	81.13
	EED	76.29	79.25	-	80.55	80.49
	BEED	**76.77**	**79.55**	-	**80.93**	**81.18**
MobileNet-V2	Baseline	-	-	-	71.87	-
	CE	72.65	74.37	76.20	73.73	78.21
	KD	73.82	75.09	76.29	73.68	78.36
	EED	75.71	76.58	77.43	76.63	78.69
	BEED	**75.76**	**76.87**	**77.47**	**76.74**	**78.86**
EfficientNetB0	Baseline	-	-	-	71.79	-
	CE	73.88	75.35	77.51	74.21	79.30
	KD	73.89	75.41	76.62	74.44	78.96
	EED	**76.13**	76.72	77.94	77.47	**79.38**
	BEED	76.12	**77.23**	**78.01**	**77.62**	79.36