1 Diving models into blocks

All models in our experiment were built on top of PyTorch, which is a popular deep learning library in Python. Note that we only present how to divide the model into blocks. Please see the corresponding papers for more details.

1.1 ResNet

The structure of ResNet-18 and ResNet-34 [1] are shown in Table 1. Note that the input size is 32×32 because the image size of the CIFAR-10 and CIFAR-100 are 32×32. And the output dimension of the model depends on which dataset is used.

Table 1 Model Structure of ResNet-18 and ResNet-34

Block	Output size	ResNet-18	ResNet-34
conv1	32×32	7×7, 64, stride 2	
conv2	56×56	3×3 max pool, stride 2	
		$\begin{bmatrix} 3\times3\times64 \\ 3\times3\times64 \end{bmatrix} \times 2$ conv	$\begin{bmatrix} 3\times3\times64 \\ 3\times3\times64 \end{bmatrix} \times 3$ conv
conv3	28×28	$\begin{bmatrix} 3\times3\times128 \\ 3\times3\times128 \end{bmatrix} \times 2$ conv	$\begin{bmatrix} 3\times3\times128 \\ 3\times3\times128 \end{bmatrix} \times 4$ conv
conv4	14×14	$\begin{bmatrix} 3\times3\times256 \\ 3\times3\times256 \end{bmatrix} \times 2$ conv	$\begin{bmatrix} 3\times3\times256 \\ 3\times3\times256 \end{bmatrix} \times 6$ conv
conv5	7×7	$\begin{bmatrix} 3\times3\times512 \\ 3\times3\times512 \end{bmatrix} \times 2$ conv	$\begin{bmatrix} 3\times3\times512 \\ 3\times3\times512 \end{bmatrix} \times 3$ conv
fc	1×10 / 1×100	average pool, 10/100-d fc, softmax	

1.2 MobileNet

The model structure of MobileNet [2] is shown in Table 2. Note that *dw* is the depthwise separable convolutional layer.

Table 2 Model Structure of MobileNet

Block	Output size	MobileNet
conv1	112×112	3×3×3×32 conv 3×3×32 dw 1×1×32×64 conv
conv2	56×56	3×3×64 dw 1×1×128×128 conv 3×3×128 dw
conv3	28×28	1×1×128×256 conv 3×3×256 dw 1×1×256×256 conv 3×3×256 dw
conv4	7×7	1×1×256×512 conv $\begin{bmatrix} 3\times3\times512 \text{ dw} \\ 1\times1\times512\times512 \text{ conv} \end{bmatrix} \times 5$ 3×3×512 dw
conv5	7×7	1×1×512×1024 conv

		3×3×1024 dw 1×1×1024×1024 conv 3×3×1024 dw 1×1×1024×1024 conv
avg	1×1	7×7 average pool
fc	1×10 / 1×100	10/100-d fc, softmax

1.3 MobileNetV2

The model structure of MobileNetV2 [3] is shown in Table 3. Note that *bottleneck* is the bottleneck layer.

Table 3 Model Structure of MobileNetV2

Block	**Output size**	**MobileNetV2**
conv2d1	112×112	3×3×2×24 conv 3×3 max pool
bottleneck1	112×112	bottleneck×1
bottleneck2	56×56	bottleneck×2
bottleneck3	28×28	bottleneck×3
bottleneck4	14×14	bottleneck×4
bottleneck5	14×14	bottleneck×3
bottleneck6	7×7	bottleneck×3
bottleneck7	7×7	bottleneck×1
avg	1×1	7×7 average pool
conv2d2	1×10 / 1×100	1×1 conv

1.4 ShuffleNetV2

The model structure of ShuffleNetV2 [4] is shown in Table 4.

Table 4 Model Structure of ShuffleNetV2

Block	**Output size**	**MobileNetV2**
conv1	56×56	3×3×3×32 conv
stage2	28×28	ShuffleUnit×4
stage3	14×14	ShuffleUnit×8
stage4	7×7	ShuffleUnit×4
conv5	7×7	1×1 conv
global pool	1×1	7×7 global pool
fc	1×10 / 1×100	10/100-d fc, softmax

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. *In Proceedings of The. IEEE conference on computer vision and pattern recognition (CVPR'16)*. 770-778.

[2] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, et al. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv: 1704.04861. Retrieved from https://arxiv.org/abs/1704.04861.

[3] M. Sandler, A, Howard, M. Zhu, A. Zhmoginov, and L. Chen. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *In Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*. 4510-4520.

[4] N. Ma, X. Zhang, H. Zheng, and J. Sun. 2018. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In *Proceedings of The European Conference on Computer Vision (ECCV'18)*. 116-131.