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-20 and ResNet-32 [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 is 32×32, and the output dimension of the model depends on which dataset is used.

Table 1 Model Structure of ResNet-20 and ResNet-32

Block	ResNet-20	ResNet-32
conv1	3×3 conv BatchNorm ReLU	
layer1	$\begin{bmatrix} 3\times3\text{ conv} \\ \text{BatchNorm} \\ \text{ReLU} \\ 3\times3\text{ conv} \\ \text{BatchNorm} \end{bmatrix}\times 3$	$\begin{bmatrix} 3\times3\text{ conv} \\ \text{BatchNorm} \\ \text{ReLU} \\ 3\times3\text{ conv} \\ \text{BatchNorm} \end{bmatrix}\times 5$
layer2	$\begin{bmatrix} 3\times3\text{ conv} \\ \text{BatchNorm} \\ \text{ReLU} \\ 3\times3\text{ conv} \\ \text{BatchNorm} \end{bmatrix}\times 3$	$\begin{bmatrix} 3\times3\text{ conv} \\ \text{BatchNorm} \\ \text{ReLU} \\ 3\times3\text{ conv} \\ \text{BatchNorm} \end{bmatrix}\times 5$
layer3	$\begin{bmatrix} 3\times3\text{ conv} \\ \text{BatchNorm} \\ \text{ReLU} \\ 3\times3\text{ conv} \\ \text{BatchNorm} \end{bmatrix}\times 3$	$\begin{bmatrix} 3\times3\text{ conv} \\ \text{BatchNorm} \\ \text{ReLU} \\ 3\times3\text{ conv} \\ \text{BatchNorm} \end{bmatrix}\times 5$
fc	average pool, 10/100-d fc, softmax	

1.2 MobileNetV2

The model structure of MobileNetV2 [2] is shown in Table 2. Note that the *bottleneck* is the bottleneck layer.

Table 2 Model Structure of MobileNetV2

Block	MobileNetV2
conv2d1	3×3×2×24 conv 3×3 max pool
bottleneck1	bottleneck×1 bottleneck×2
Bottleneck2	bottleneck×3 bottleneck×4
Bottleneck3	bottleneck×3 bottleneck×3 bottleneck×1
avg	7×7 average pool

conv2d2	1×1 conv

1.3 ShuffleNetV2

The model structure of ShuffleNetV2 [3] is shown in Table 3.

Table 3 Model Structure of ShuffleNetV2

Block	**MobileNetV2**
conv1	3×3×3×32 conv
stage2	ShuffleUnit×4
stage3	ShuffleUnit×8
stage4	ShuffleUnit×4
conv5	1×1 conv 7×7 global pool 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] 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.

[3] 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.