Supplementary Material for ”Towards Efficient Tensor Decomposition-Based DNN Model Compression with Optimization Framework”

Miao Yin[1], Yang Sui[1], Siyu Liao[2†] and Bo Yuan[1]
[1]Department of ECE, Rutgers University, [2]Amazon
{miao.yin, yang.sui}@rutgers.edu, liasiyu@amazon.com, bo.yuan@soe.rutgers.edu

Implementation Details

All TT operations are implemented by reshape, transpose and matrix multiplication offered by PyTorch.

For ResNet-18 on ImageNet, the original weight tensor of size $O \times I \times K \times K$ is reshaped and transposed to size $O \times K^2 \times I$. The TT-ranks for FLOPs reduction 2.47× are summarized in Table 1.

Layer	TT-ranks
layer1.0.conv1	$[1, 64, 64, 1]$
layer1.0.conv2	$[1, 64, 64, 1]$
layer1.1.conv1	$[1, 64, 64, 1]$
layer1.1.conv2	$[1, 64, 64, 1]$
layer2.0.conv1	$[1, 120, 60, 1]$
layer2.0.conv2	$[1, 100, 100, 1]$
layer2.1.conv1	$[1, 100, 100, 1]$
layer2.1.conv2	$[1, 100, 100, 1]$
layer3.0.conv1	$[1, 200, 150, 1]$
layer3.0.conv2	$[1, 135, 135, 1]$
layer3.1.conv1	$[1, 135, 135, 1]$
layer3.1.conv2	$[1, 135, 135, 1]$
layer4.0.conv1	$[1, 320, 200, 1]$
layer4.0.conv2	$[1, 170, 170, 1]$
layer4.1.conv1	$[1, 170, 170, 1]$
layer4.1.conv2	$[1, 170, 170, 1]$

Table 1: TT-ranks for ResNet-18.

For other DNN models, the settings keep similar to prior works.

[†] This work was done when the author was with Rutgers University.