A. Appendix

A.1. Layer Pruning

Theoretical Issues. Assume a network $\mathcal{F}$ of L layers as a set of L transformations $f_i(.)$. For the sake of simplicity, f_i consists of a series of convolution, batch normalization, and activation operations. In this definition, we obtain the network output (y) by forwarding the input data through the sequential layers f, where the input of layer i is the output of the previous layer $i-1$; therefore, $y = f(x)\ f_L(...f_2(f_1(.)))$. This composes the idea behind plain networks (i.e., VGG).

In residual-like networks, the output of layer i, y_i, consists of the transformation f_i plus the input it receives y_{i-1} (see Figure 4). Formally, we can define the output of the i-th layer as

$$y_i = f_i(y_{i-1}) + y_{i-1}. \tag{2}$$

Equation 2 composes a residual module, where the rightmost part is named *identity-mapping shortcut* (or identity for short). It is important to observe in Equation 2 that if we disable $f(i)$ (a layer) then $y_i = y_{i-1}$.

Veit et al. [39] showed that the identity enables the information to take different paths in the network, in the sense that, we can disable some f_i without degrading (or with negligible damage) the expected representation of the subsequent layers (i.e., f_{i+1}). In other words, some layers f_i do not depend strongly on each other; hence, we can eliminate them. For example, in Figure 4, we could remove layer i with no loss in the predictive ability of the network. On the other hand, due to the absence of identity, plain networks meet collapse in the representation if we remove only one of their layers. We refer interested readers to Figure 3 of the study by Veit et al. [39] for a comparison of accuracy drop between residual and plain networks.

Technical Issues. We can disable layer i by setting its weighs to zero (the widely employed zeroed-out scheme). This way, the output of layer $i-1$ is directly connected to layer $i+1$ (see Figure 4 middle). However, such a process does not achieve performance gains without specialized frameworks or hardware for sparse computation. Instead of zeroing weights, we can perform the following process. After identifying which layers remove (i.e., a victim), we create a new network without layer i and transfer the weights of the kept layers to the new network. For example, if we have a network with L layers and want to remove k layers, then, we create a novel network with $L-k$ layers. In summary, the pruned network (bottom in Figure 2) inherits the weights of the kept layers of the original network (top in Figure 2).

We highlight that the pruning cannot remove some layers due to incompatible dimensions of (input/output) tensors. Such an incompatibility comes from the spatial resolution layer (downsampling layers). More specifically, we cannot remove layers before and after the downsampling layers. Importantly, filter pruning also suffers from this issue.

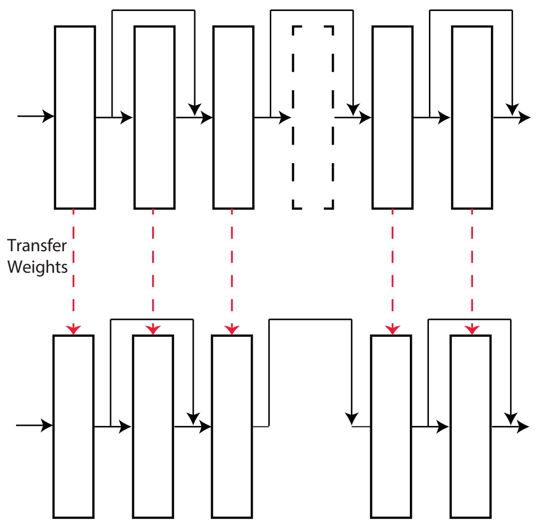

Figure 2. Overall process to remove layers (residual models) from a residual network. After identifying a victim layer (dashed rectangle), we create a novel network (bottom) without it. Finally, we transfer the weights (red arrows) of the kept layers from the original unpruned network (top) to the new network.

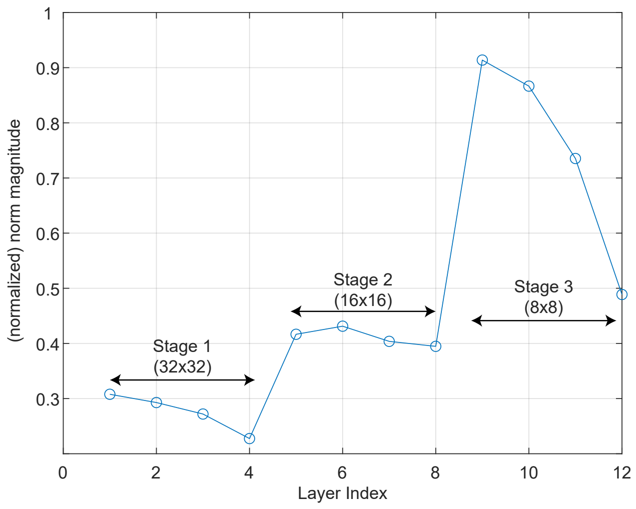

Figure 3. ℓ_1-norm score of layers of ResNet32. Layers within a stage operate on the same input/output spatial resolution (i.e., the size of the feature map – values in parentheses).

A.2. ℓ_1-norm

Figure 3 illustrates the ℓ_1-norm scores of layers (the ones that the pruning could remove) of ResNet32.

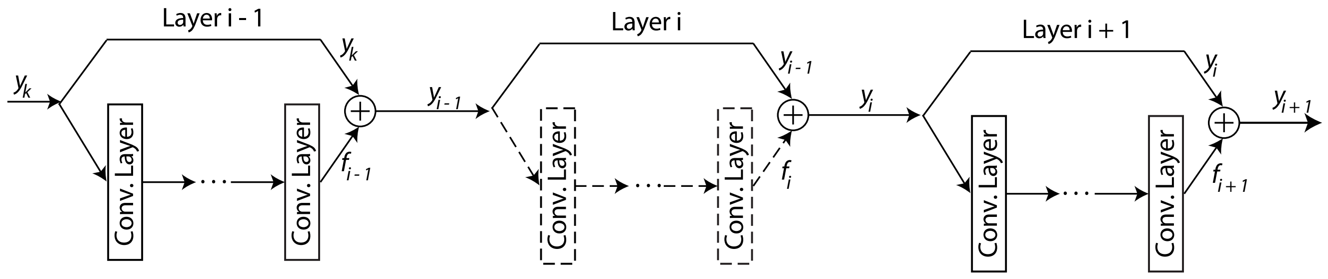

Figure 4. Architecture of a residual-like network. The rationale behind this architecture is that the output of a layer takes into account the transformation performed by it (f) plus ($\oplus$) the input (y) it receives. Due to this essence, when we disable layer i (its transformation – dashed lines), the output (representation) of layer $i-1$ is propagated to layer $i+1$, which means that the output y_i belongs y_{i-1}. For the sake of simplicity, we omit the batch normalization and activation layers.

From this figure, we see that the magnitude of scores of layers correlates with the stage (groups of layers operating on the same resolution of feature maps) to which they belong. Since our pruning strategy takes into account all layers (i.e., all scores) at once, this criterion is infeasible. Specifically, there is a bias to layers of early stages (i.e., they will always be selected as victims).