Layer Folding: Neural Network Depth Reduction using Activation Linearization

Amir Ben Dror, Niv Zehngut, Avraham Raviv, Evgeny Artyomov and Ran Vitek

SAMSUNG

SIRC – Samsung Israel R&D Center, Tel Aviv, Israel

BMVC 2022

MOTIVATION

- As deep neural networks become more prevalent, their applicability to **resource-constrained devices** is limited.

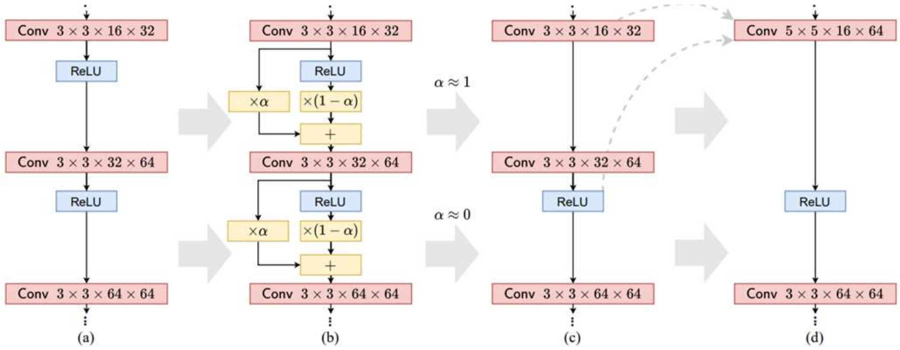
- While modern devices exhibit a high level of parallelism, **real-time latency** is still highly dependent on networks' **depth**.

- Recent works show the width of shallower networks must grow exponentially below a certain depth. However, we presume that neural networks **usually exceed this minimum depth** to accelerate convergence and incrementally increase accuracy

- This motivates us to **transform** pre-trained deep networks that already exploit such advantages **into shallower forms**.

METHOD

Removing activations (non-linearities) allows us to merge consecutive linear layers into a single layer. Thus, we focus on removing activations as a method to reduce depth.



We replace each activation σ with is learnable parametric counterpart:

$$\sigma_\alpha(x) = \alpha x + (1-\alpha)\sigma(x)$$

When $\alpha = 0$ we get the original activation, when $\alpha = 1$ we get the identity (essentially removing the activation).

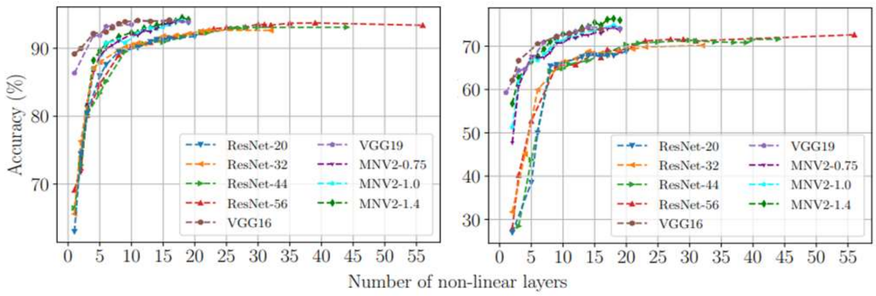
We use an auxiliary loss to encourage each α to become 1:

$$\mathcal{L}_c = \sum_{l \in L} (1 - \alpha_l^p)$$

OBJECTIVES

- **Reduce the depth of a pre-trained network with minimal impact on accuracy.**
- **Provide more efficient alternatives to MobileNet and EfficientNet architectures on the classification task.**
- **Explore the accuracy-depth and accuracy-latency trade-offs.**

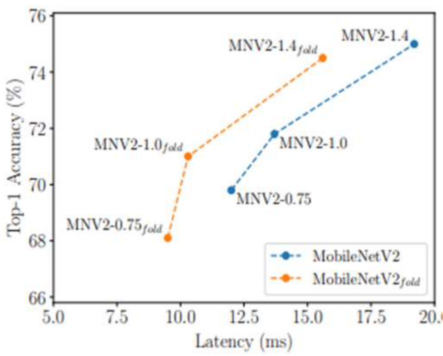
ACCURACY ~ DEPTH



Layer Folding applied on ResNet, VGG, and MobileNetV2 (MNV2) architectures on CIFAR-10 (left) and CIFAR-100 (right). For each network, we gradually remove nonlinear layers.

Visit Us:

ACCURACY ~ LATENCY



Model	Acc. (%) / Acc. Drop (%)	Latency Reduction	FLOPs Reduction
MNV2-0.75	68.1 / 1.7	21%	4%
MNV2-1.0	71.0 / 0.8	25%	7%
MNV2-1.4	75.5 / 0.5	19%	3%
EffNet-lite0	74.6 / 0.5	15%	3%
EffNet-lite1	75.8 / 1.0	13%	0%

Latency and FLOPs reduction obtained by applying Layer Folding on MobileNetV2 (MNV2) and EfficientNet (EffNet) on ImageNet.