# Thanks for Nothing: Predicting Zero-Valued Activations with Lightweight Convolutional Neural Networks

## Supplementary Material

Gil Shomron[1], Ron Banner[2], Moran Shkolnik[1,2], and Uri Weiser[1]

[1] Faculty of Electrical Engineering — Technion, Haifa, Israel
{gilsho@campus, uri.weiser@ee}.technion.ac.il
[2] Habana Labs — An Intel Company, Caesarea, Israel
{rbanner, mshkolnik}@habana.ai

**Abstract.** In this supplementary material, we visualize the curve fitting results used to optimize the threshold values and provide the parameters and commands used to achieve the paper results.

## 1 Curve Fit

To solve the optimization problem presented in the main text, each measured curve was fitted with a sigmoid curve $d + \frac{a}{1+e^{-bx+c}}$. Results for AlexNet, VGG-16, and ResNet-18 with masks B and D are presented in Figures 1, 2, and 3, respectively. Layer-wise ZAP analysis, together with curve fitting and threshold optimization, is achieved by executing the following command:

```
python ./main.py -a [ARCH]-imagenet --phase EVAL_ZAP_ISOL
                                    --mask_list [MASK]
                                    --opt_th
```

Note that the indexes for masks A, B, C, and D are 6, 5, 4, and 3, respectively. `STATS_ERR_TO_TH_MIN` and `STATS_ERR_TO_TH_MAX` were set in the `Config.py` file to -1.3 and 1.3, respectively. `STATS_MASK_VAL_HIST_MIN` and `STATS_MASK_VAL_HIST_MAX` were set, as well, to -1.3 and 1.3, respectively.

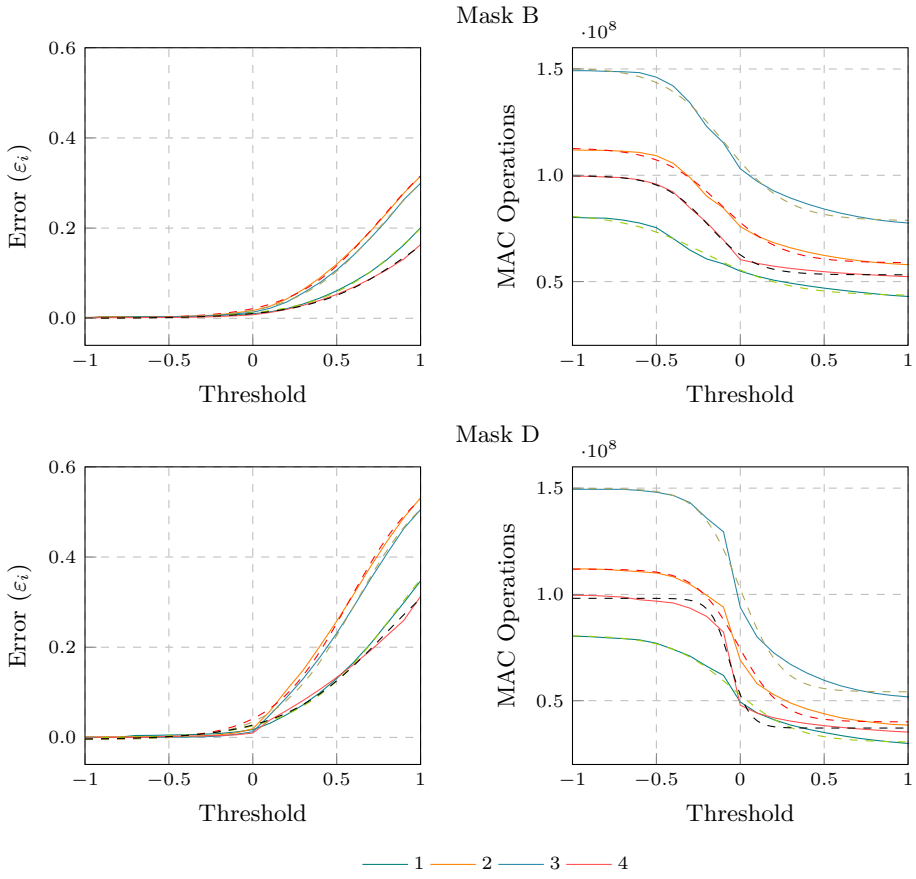


**Fig. 1. AlexNet + ImageNet.** ZAP error contribution and MAC savings per layer. Solid lines represent the measured values. Dashed lines represent the fitted curve.

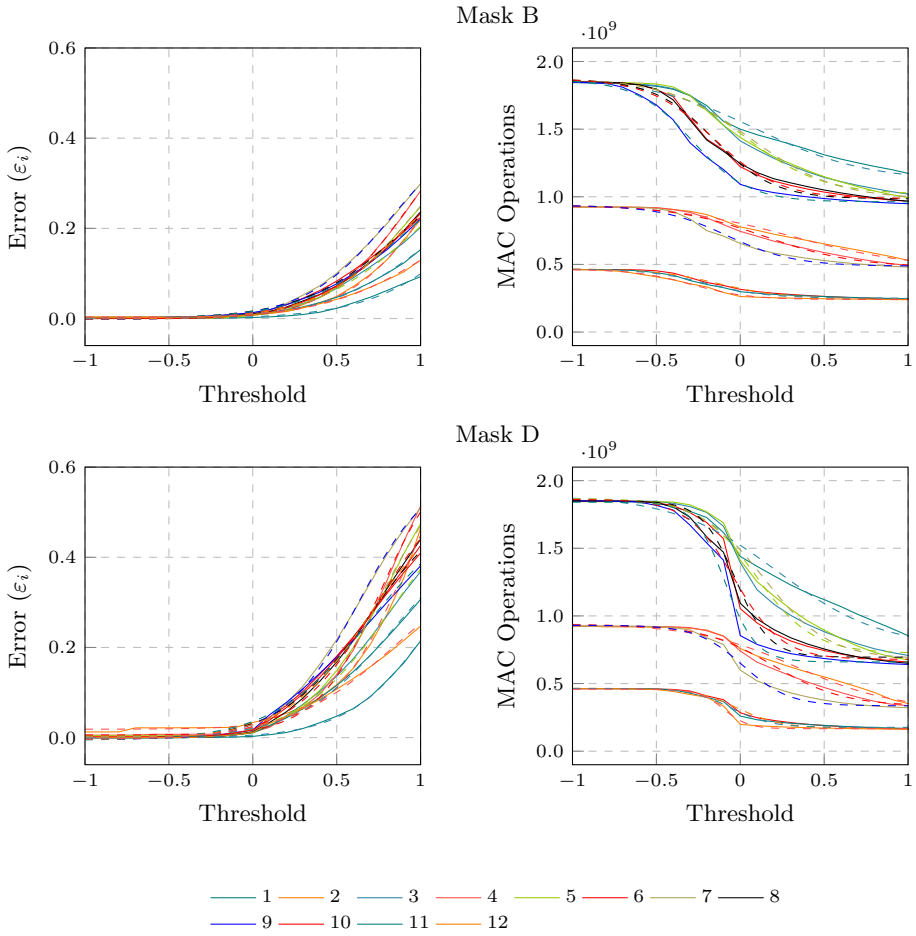


**Fig. 2. VGG-16 + ImageNet.** ZAP error contribution and MAC savings per layer. Solid lines represent the measured values. Dashed lines represent the fitted curve.

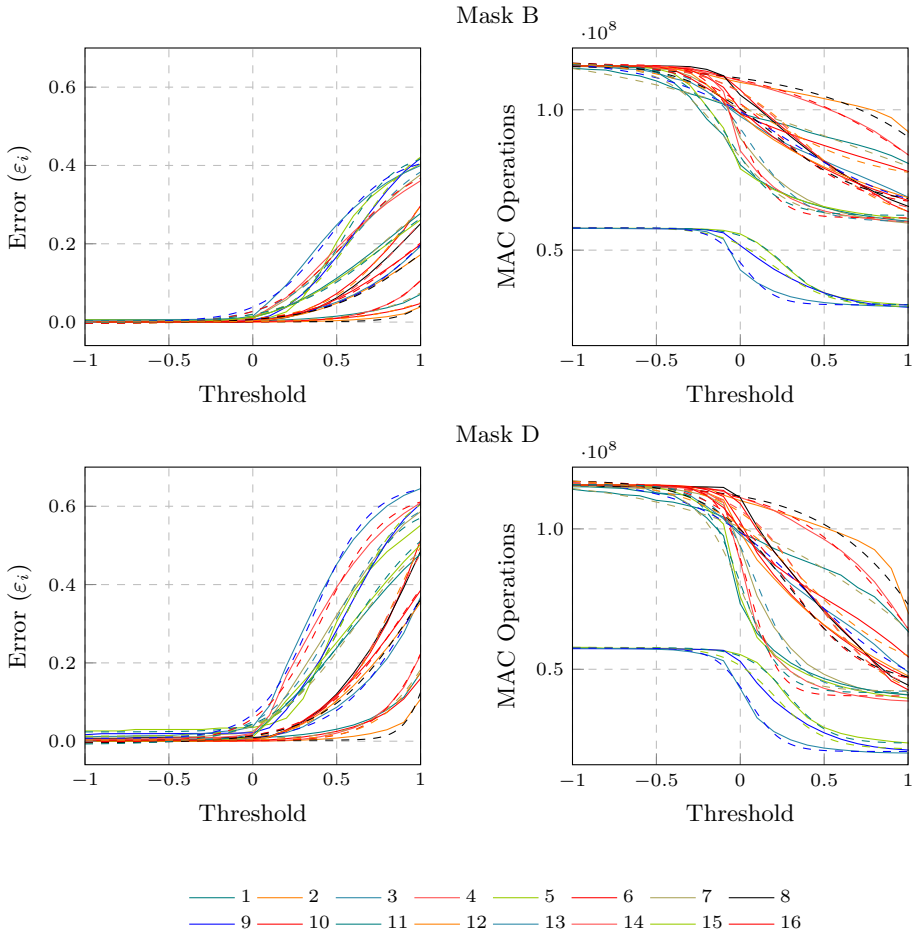


**Fig. 3. ResNet-18 + ImageNet.** ZAP error contribution and MAC savings per layer. Solid lines represent the measured values. Dashed lines represent the fitted curve.

## 2 Experimental Settings

Our code was tested with Python 3.5.2 and 3.6.8. Code dependencies can be found in the `requirements.txt` file. We provide the pretrained ZAP checkpoint files for all described models in the main text and the ImageNet threshold dictionaries (which correspond to the optimal thresholds, given an error constraint).

### 2.1 Training ZAP

To train the model ZAPs from scratch, the following command should be executed:

```
python ./main.py -a [ARCH]-imagenet --phase TRAIN_ZAP
                                    --mask_list [MASK]
```

As before, indexes for masks A, B, C, and D are 6, 5, 4, and 3, respectively. The command outputs one ZAP checkpoint per layer, which can later be used by updating the checkpoint paths in the `Config.py` file.

### 2.2 Results Comparison Table

In Table 1 we report the parameters used to achieve the results presented in the comparison table in the main text. The results in Table 1 were achieved by executing the following command:

```
python ./main.py -a [ARCH]-imagenet
    --phase EVAL_MODEL
    --mask_list [MASK]
    --th_dict ./data/results/curve_fit/[ARCH]-imagenet/
                              [ARCH]-imagenet_th_dict-[MASK].p
    --th_list [EPSILON] --lr [LR] --wd [WD] --batch_size [BATCH]
    --epochs 1
```

**Table 1. Chosen operating points and their execution parameters:** mask, $\epsilon$, learning rate (LR), weight decay (WD), and batch size.

| Network | Top1 | Top1-ft | Top5 | Top5-ft | MAC | Mask | $\epsilon$ | LR | WD | Batch |
|---|---|---|---|---|---|---|---|---|---|---|
| AlexNet | 4.63 | 1.97 | 3.04 | 1.17 | 51.1% | D | 0.38 | 0.00001 | 0.0005 | 128 |
| | 0.34 | 0.33 | 0.24 | 0.14 | 32.4% | B | 0.09 | | | |
| | 1.28 | 0.78 | 0.84 | 0.42 | 38.0% | B | 0.17 | | | |
| VGG-16 | 11.02 | 1.27 | 7.02 | 0.71 | 44.5% | D | 0.80 | 0.00001 | 0.0005 | 32 |
| | 0.54 | 0.24 | 0.33 | 0.11 | 26.2% | B | 0.17 | | | |
| | 1.71 | 0.31 | 0.87 | 0.19 | 31.3% | B | 0.28 | | | |
| ResNet-18 | 12.35 | 7.22 | 8.37 | 4.66 | 34.0% | D | 1.80 | 0.0001 | 0.0001 | 256 |
| | 2.96 | 1.86 | 1.70 | 1.13 | 23.8% | B | 0.80 | | | |