Supplementary Material of Dynamic ReLU

Anonymous ECCV submission

Paper ID 3223

In this supplementary material, we report additional analysis and experimental results for our dynamic ReLU (DY-ReLU) method.

1 Is DY-ReLU Dynamic?

In this section, we check if DY-ReLU is dynamic. Therefore, we inspect the input and output of DY-ReLU, and expect different activation values (y) across different images for a fixed input value (e.g. $x = 0.5$). In contrast, for a given

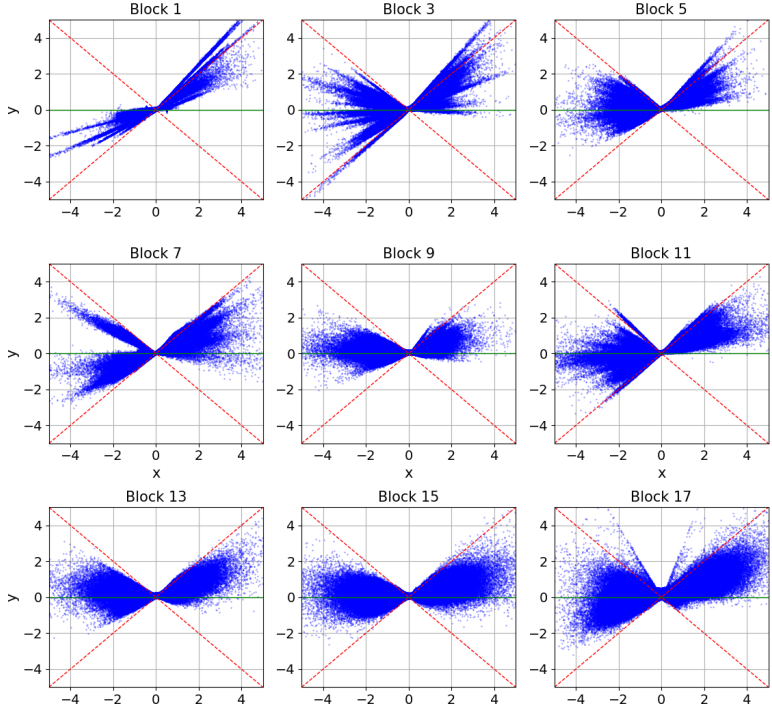

Fig. 1. Plots of input and output values of DY-ReLU in a well trained model (using MobileNetV2 $\times 0.35$) over 50,000 validation images in ImageNet [2]. We choose the dynamic ReLU after the depthwise convolution in every other mobile block. Block 1 (the top-left plot) corresponds to the lowest block, and Block 17 (the bottom-right plot) corresponds to the highest block. The two red lines correspond to $y = x$ and $y = -x$, respectively. Best viewed in color.

input (e.g. $x = 0.5$), the output of ReLU is fixed ($y = 0.5$) regardless of channel or input image. Thus, the input-output pairs of ReLU fall into two lines ($y = x$ if $x > 0$, $y = 0$ otherwise).

Fig. 1 plots the input and output values of DY-ReLU at different blocks (from low level to high level) for 50,000 validation images in ImageNet [2]. We confirm that the activation values (y) vary in a range (that blue dots in Fig. 1 cover) for a fixed input x. This demonstrates that the learnt DY-ReLU is dynamic to features. Furthermore, we observe that the distribution of input-output pairs varies across different blocks, indicating different dynamic functions learnt across levels.

2 Comparison between DY-ReLU with Prior Works

Table 1 shows the comparison between DY-ReLU and prior works, using MobileNetV2 $\times 1.0$. This is additional to the results (Table 6) shown in the submitted paper, which are generated by using MobileNetV2 $\times 0.35$. The same conclusion holds for these two experiments: *our method outperforms all prior works*. Compared to Maxout which has significantly more computational cost, DY-ReLU gains 1.1% and 0.4% top-1 accuracy for $K = 2$ and $K = 3$, respectively. This demonstrates that DY-ReLU not only has more representation capability, but also is computationally efficient.

3 Ablations of DY-ReLU Variations on Pose Estimation

In this section, we report additional results of comparing three DY-ReLU variations on COCO keypoint detection [6] (or pose estimation). The three variations are listed as follows:

Activation	K	#Param	MAdds	Top-1	Top-5
ReLU	2	3.5M	300.0M	72.0	91.0
RReLU [8]	2	3.5M	300.0M	$72.5_{(+0.5)}$	$90.8_{(-0.2)}$
LeakyReLU [7]	2	3.5M	300.0M	$72.7_{(+0.7)}$	$90.8_{(-0.2)}$
PReLU (channel-wise) [4]	2	3.5M	300.0M	$72.9_{(+0.9)}$	$91.0_{(+0.0)}$
PReLU (channel-shared) [4]	2	3.5M	300.0M	$73.3_{(+1.3)}$	$91.2_{(+0.2)}$
SE[5]+ReLU	2	5.1M	304.8M	$74.2_{(+2.2)}$	$91.9_{(+0.9)}$
Maxout [3]	2	5.7M	579.1M	$75.1_{(+3.1)}$	$92.3_{(+1.3)}$
Maxout [3]	3	7.8M	866.4M	$75.8_{(+3.8)}$	$92.7_{(+1.7)}$
DY-ReLU-B	2	7.5M	315.5M	$\mathbf{76.2}_{(+4.2)}$	$93.1_{(+2.1)}$
DY-ReLU-B	3	9.2M	322.8M	$\mathbf{76.2}_{(+4.2)}$	$\mathbf{93.2}_{(+2.2)}$

Table 1. Comparing DY-ReLU with related activation functions on ImageNet [2] classification. MobileNetV2 with width multiplier $\times 1.0$ is used. We use spatial-shared and channel-wise DY-ReLU-B with $K = 2, 3$ linear functions. The numbers in brackets denote the performance improvement over the baseline. DY-ReLU outperforms all prior works including Maxout, which has significantly more computations.

		Head			
		ReLU	DY-ReLU-A	DY-ReLU-B	DY-ReLU-C
Backbone	ReLU	59.2	57.0	58.4	61.0
	DY-ReLU-A	58.8	51.5	56.5	62.4
	DY-ReLU-B	61.5	54.3	58.6	63.2
	DY-ReLU-C	61.9	53.5	58.8	63.3

Table 2. Comparing three DY-ReLU variations on COCO keypoint detection [6]. The numbers in the table are average precision (AP) over 10 object key point similarity (OKS) thresholds. We use MobileNetV2 with width multiplier $\times 0.5$ as backbone and use up-sampling and inverted residual bottleneck blocks [1] as head. In the backbone, channel-wise variations (DY-ReLU-B and DY-ReLU-C) are more effective than channel-shared (DY-ReLU-A). In the head, spatial-wise variation (DY-ReLU-C) is more effective than spatial-shared (DY-ReLU-A and DY-ReLU-B).

DY-ReLU-A: the dynamic activation is *spatial and channel-shared*.

DY-ReLU-B: the dynamic activation is *spatial-shared and channel-wise*.

DY-ReLU-C: the dynamic activation is *spatial and channel-wise*.

Table 2 shows average precisions (AP) for all 16 combinations of using ReLU or DY-ReLU variations in both backbone and head. This is additional to the results (Table 4) in the submitted paper. The original conclusions hold: (a) spatial-wise (DY-ReLU-C) is critical in the head network as the last column in Table 2 has higher AP than the previous three columns, and (b) the optimal solution is to use channel-wise variation (variation B or C) in backbone and use spatial-wise DY-ReLU-C in head (see the last two rows in the last column of Table 2). Compared to the baseline that uses ReLU in both backbone and head, using DY-ReLU-C achieves 4.1 AP improvement.

The spatial-wise variation (DY-ReLU-C) is a better fit for keypoint detection, which is spatially sensitive (distinguishing body joints in pixel level). This is because the spatial attention allows different magnitudes of activation at different locations. Thus, it encourages better learning of DY-ReLU especially in the head network with larger resolutions.

4 Implementation Details of MobileNetV2

We now show the implementation details of MobileNetV2. Basically, we use larger weight decay, dropout and more data augmentation for higher width multipliers (e.g. $\times 1.0$) to prevent overfitting. We use weight decay 2e-5 and dropout 0.1 for width $\times 0.35$ and increase weight decay (3e-5) and dropout (0.2) for width $\times 0.5$, $\times 0.75$, $\times 1.0$. Random cropping/flipping and color jitter are used for all width multipliers. Mixup [9] is used for width $\times 1.0$. Without using Mixup, the top-1 accuracy of DY-ReLU drops from 76.2% to 75.7%, which still outperforms the static counterpart (72.0%) by a clear margin.

References

1. Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., Liu, Z.: Dynamic convolution: Attention over convolution kernels. ArXiv **abs/1912.03458** (2019)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
3. Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. arXiv preprint arXiv:1302.4389 (2013)
4. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: ICCV (2015)
5. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
6. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
7. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: in ICML Workshop on Deep Learning for Audio, Speech and Language Processing (2013)
8. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. CoRR (2015)
9. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=r1Ddp1-Rb