Supplementary Information for: Convolution with even-sized kernels and symmetric padding

Anonymous Author(s)
Affiliation
Address
email

S1 Compare with NAS models

Table S1: Test error rates (%) on CIFAR10 dataset. c/o and *mixup* denotes cutout [1] and mixup [14] data augmentation.

Model	Error (%)	Params (M)
NASNet-A [15]	3.41	3.3
PNASNet-5 [7]	3.41	3.2
AmoebaNet-A [11]	**3.34**	3.2
Wide-DenseNet C3	3.81	3.4
Wide-DenseNet C2sp	3.54	3.2
NASNet-A + c/o [15]	2.65	3.3
Wide-DenseNet C2sp + c/o + *mixup*	**2.44**	3.2

In Table S1, we compare C2sp with NAS models: NASNet [15], PNASNet [7], and AmoebaNet [11]. We apply Wide-DenseNet [3] and adjust the width and depth ($\mathrm{K} = 48, \mathrm{L} = 50$) to have approximately 3.3M parameters. C2sp suffers less than 0.2% accuracy loss compared with state-of-the-art auto-generated models, and achieves better accuracy ($+0.21\%$) when the augmentation is enhanced. Although NAS models leverage fragmented operators [9], e.g., pooling, group convolution, DWConv to improve accuracy with similar numbers of parameters, the regular-structured Wide-DenseNet has better memory and computational efficiency in runtime. In our reproduction, the training speeds on TitanXP for NASNet-A and Wide-DesNet are about 200 and 400 SPS, respectively.

S2 Implementation details

Results reported as mean$\pm$std in tables or error bars in figures are trained for 5 times with different random seeds. The default settings for CIFAR classifications are as follows: We train models for 300 epochs with mini-batch size 64 except for the results in Table S1, which run 600 epochs as in [15]. We use a cosine learning rate decay [8] starting from 0.1 except for DenseNet tests, where the piecewise constant decay performs better. The weight decay factor is 1e-4 except for parameters in depthwise convolutions. The standard augmentation [6] is applied and the α equals to 1 in mixup augmentation.

For ImageNet classifications, all the models are trained for 100 epochs with mini-batch size 256. The learning rate is set to 0.1 initially and annealed according to the cosine decay schedule. We follow the data augmentation in [13]. Weight decay is 1e-4 in ResNet-50 and DenseNet-121 models, and decreases to 4e-5 in the other compact models. Some results are worse than reported in the original

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

papers. It is likely due to the inconsistency of mini-batch size, learning rate decay, or total training epochs, e.g., about 420 epochs in [12].

In generation tasks with GANs, we follow models and hypermeters recommended in [5]. The learning rate is 0.2, β_1 is 0.5 and β_2 is 0.999 for Adam optimizer [4]. The mini-batch size is 64, the ratio of discriminator to generator updates is 5:1 ($n_{\text{critic}} = 5$). The results in Table 3 and Figure 4 are trained for 200k and 500k discriminator update steps, respectively. We use the non-saturation loss [2] without gradient norm penalty. The spectral normalization [10] is applied in discriminators, no normalization is applied in generators.

References

[1] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[3] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[5] Karol Kurach, Mario Lucic, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. The gan landscape: Losses, architectures, regularization, and normalization. *arXiv preprint arXiv:1807.04720*, 2018.

[6] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial Intelligence and Statistics*, pages 562–570, 2015.

[7] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018.

[8] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

[9] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 116–131, 2018.

[10] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[11] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. *arXiv preprint arXiv:1802.01548*, 2018.

[12] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[13] Nathan Silberman and Sergio Guadarrama. Tensorflowslim image classification model library, 2017.

[14] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[15] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.