DEEP PYRAMIDAL RESIDUAL NETWORKS WITH STOCHASTIC DEPTH

Yoshihiro Yamada, Masakazu Iwamura, and Koichi Kise
Graduate School of Engineering Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku Sakai-shi, Osaka
yamada@m.cs.osakafu-u.ac.jp,{masa,kise}@cs.osakafu-u.ac.jp

ABSTRACT

In generic object recognition tasks, ResNet and its improvements have broken the lowest error rate records. ResNet enables us to make a network deeper by introducing residual learning. Some ResNet improvements achieve higher accuracy by focusing on channels. Thus, the network depth and channels are thought to be important for high accuracy. In this paper, in addition to them, we pay attention to use of multiple models in data-parallel learning. We refer to it as data-parallel multi-model learning. We observed that the accuracy increased as models concurrently used increased on some methods, particularly on the combination of PyramidNet and the stochastic depth proposed in the paper. As a result, we confirmed that the methods outperformed the conventional methods; on CIFAR-100, the proposed methods achieved error rates of 16.13% and 16.18% in contrast to PiramidNet achieving that of 18.29% and the current state-of-the-art DenseNet-BC 17.18%.

1 INTRODUCTION

It is expected that a deeper network can have a higher discriminant ability (Simonyan & Zisserman (2014)). However, realizing it is difficult because of nuisances such as the vanishing gradient problem (Glorot & Bengio (2010)). To avoid them and facilitates learning of deeper networks, ResNet (He et al. (2016)) introduces residual learning. The residual learning is realized by a processing block, called residual block; it has the ability to realize the identity mapping that directly outputs the input. ResNet is known well because ResNet and its improvements, some of which are shown in Table 1, have broken the lowest error rate records.

ResDrop (Huang et al. (2016b)) is a ResNet improvement which further avoids the nuisances. In deep convolutional neural networks (even in ResNet), as a network becomes deeper, gradients of processing layers tend to be smaller. As a result, learning does not progress well. To avoid the problem, ResDrop makes the network apparently shallow in learning by introducing a regularizer called Stochastic Depth; it treats some of residual blocks stochastically selected as the identity mapping.

In addition to making a network deeper for which ResNet and ResDrop aim, increase of channels is also thought to be important to increase accuracy. On this line, some ResNet improvements such as Wide ResNet (Zagoruyko & Komodakis (2016)), PyramidNet (Han et al. (2016)), ResNeXt (Xie et al. (2016)) and DenseNet-BC (Huang et al. (2016a)) are proposed. Among them, we focus on PyramidNet because it was the state of the art when we began this research. Compared with ResNet where the number of channels does not increase except a few special residual blocks, on PyramidNet, the number of channels increases step by step on each residual block.

As ResDrop and PyramidNet are complementary improvements of ResNet, merging them is relatively easily conceivable but has a potential to increase accuracy. Indeed, the authors of PyramidNet mention use of stochastic regularizers such as Dropout (Srivastava et al. (2014)) and the stochastic depth (Huang et al. (2016b)) could improve the performance of PyramidNet, without reporting any result. Thus, in this paper, we investigate the effect on combining PyramidNet and the stochastic depth by "proposing" two methods; one is a simple combination of them, named PyramidDrop, that might be suggested by the authors of PyramidNet, and the other is its extended version named PyramidSepDrop. In the investigation, in addition to network depth, we pay attention to use of multiple models in data-parallel learning. We refer to it as data-parallel multi-model learning. To the best of

Table 1: Comparison of conventional and two proposed methods. Random Drop means the stochastic depth is introduced if checked. Gentle Channel the number of channels increases step by step on each residual block if checked. The error rates of conventional methods are from their papers.

Method	Random Drop	Gentle Channel	CIFAR-10	CIFAR-100
ResNet (He et al. (2016))	×	×	6.43%	25.16%
ResDrop(Huang et al. (2016b))	✓	×	5.23%	24.58%
PyramidNet (Han et al. (2016))	×	✓	3.77%	18.29%
ResNeXt (Xie et al. (2016))	×	×	3.58%	17.31%
DenseNet-BC (Huang et al. (2016a))	×	✓	3.46%	17.18%
PyramidDrop (Han et al. (2016) and this paper)	✓	✓	-	**16.13%**
PyramidSepDrop (this paper)	✓	✓	**3.31%**	**16.18%**

Table 2: Result of preliminary experiment with 110 layers and 4 models. The error rates of PyramidNet are from their papers.

Method	CIFAR-10	CIFAR-100
PyramidNet (Han et al. (2016))	3.77%	18.29%
PyramidDrop (Han et al. (2016) and this paper)	3.99%	18.30%
PyramidSepDrop (this paper)	**3.66%**	**18.01%**

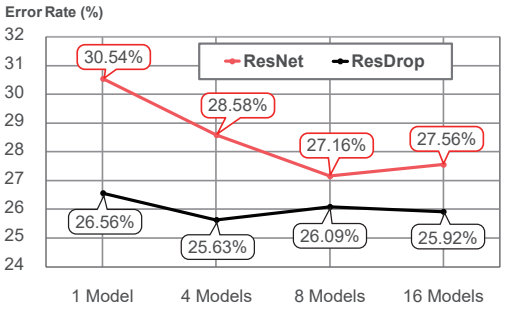
our knowledge, it has not been intentionally introduced for the purpose of increasing the accuracy because it can be thought to even decrease the accuracy. However, surprisingly, we observed that the accuracy increased as models concurrently used increased on some methods, particularly on the proposed methods. As a result, we confirmed that the proposed methods outperformed the conventional methods; on CIFAR-100, the introduced methods achieved error rates of 16.13% and 16.18% in contrast to existing methods shown in Table 1.

2 Proposed Methods

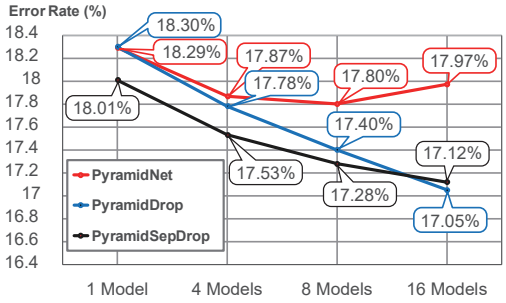
We introduce two methods obtained by combining PyramidNet with the stochastic depth. One is named Deep Pyramidal Residual Networks with Stochastic Depth (PyramidDrop) that is a simple combination of PyramidNet and the stochastic depth; the random drop mechanism of the stochastic depth is introduced to each residual block. In our preliminary experiment shown in Table 2, PyramidDrop did not gain the accuracy as expected. Therefore, we propose another method named Deep Pyramidal Residual Networks with Separated Stochastic Depth (PyramidSepDrop). To explain this method, let us remind the readers of the following; the number of channels increases in each residual block in PyramidNet, and the same number of channels as the dimensionality of the input vector are convoluted with the input vector while zero is padded in the rest. Thus, in PyramidSepDrop, two independent random drop mechanisms of the stochastic depth are introduced to the two parts of channels of each residual block.

3 Experiments

In addition to the preliminary experiment shown in Table 2, we conducted three experiments to investigate the effects on 1) the number of models, 2) network depth and 3) their combination. For conventional methods, we used existing implementations based on the Facebook ResNet implementation on Torch available at `https://github.com/facebook/fb.resnet.torch`. The proposed methods are also based on it. In the data-parallel multi-model learning, we turned on the `shareGradInput` flag. Regarding the preprocessing of images and learning conditions, we followed the experiments of ResNet and PyramidNet papers. On the other hand, the initial learning rate of ResNet and ResDrop was set to 0.1, decayed by a factor of 0.1 at 81 and 122 training epochs for 163 epochs. Parameters α and Death Rate were set according to PyramidNet and ResDrop papers; parameter α was adjusted in order to increase 5 channels per residual block.



(a) ResNet and ResDrop



(b) PyramidNet and proposed methods

Figure 1: Effect on increasing models in the data-parallel multi-model learning with 110 layers.

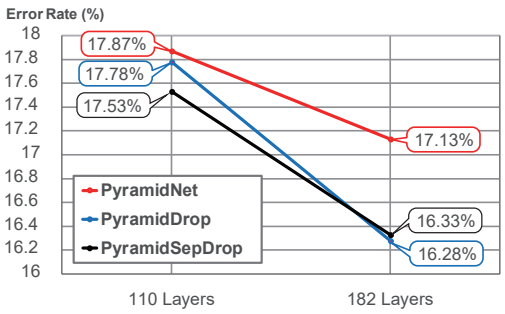


Figure 2: Effect on increasing layers with 4 models.

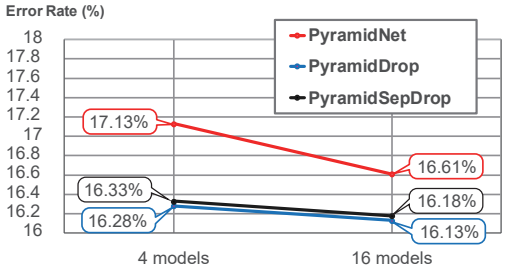


Figure 3: Effect on increasing models with 182 layers.

1) Effects on the number of models
On the experiment, the input to the network was a mini-batch of 128 samples. They were divided into the sub-batch size ($\equiv 128/\#\text{model}$). The samples in each sub-batch were used in each model for training, and network parameters learned in each model were communicated across the models in the process. Figure 1 shows the effect on increasing models in the data-parallel multi-model learning with 110 layers in five methods. Hereafter, error rates shown in the figure are on the last epoch on CIFAR-100. The figure shows that as the number of models increased, the error rates of two proposed methods kept decreasing until 16 models we could run, although the others were not. The result suggest that two proposed methods have different mechanism from other compared methods. Only in ResDrop and two proposed methods, models concurrently processed in the data-parallel multi-model learning are not identical due to the random drop mechanism. Thus, we expected that the results of these three methods have the same tendency. However, unexpectedly it was not true.

2) Effects on network depth
Figure 2 shows the effect on increasing layers with 4 models. Error rates of the proposed methods, especially PyramidDrop, dropped much more than that of PyramidNet.

3) Effect on the number of models on a deep network
Figure 3 shows the effect on increasing models in the data-parallel multi-model learning with 182 layers. As a result, the error rates tended to decrease as the number of layers increased. In addition, the proposed methods achieved error rates of 16.13% and 16.18% with 16 models. Thus, we confirmed that the proposed methods outperformed the conventional methods.

4 Acknowledgement

This work is partly supported by JSPS KAKENHI #25240028, JST CREST and AWS Cloud Credits for Research program.

REFERENCES

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS' 10). Society for Artificial Intelligence and Statistics*, 2010.

Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. arXiv:1610.02915 [cs.CV], 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. doi: 10.1109/CVPR.2016.90.

Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. arXiv:1608.06993 [cs.CV], 2016a.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth. arXiv:1603.09382 [cs.LG], 2016b.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 [cs.CV], abs/1409.1556, 2014. URL http://arxiv.org/abs/1409.1556.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. arXiv:1611.05431 [cs.CV], abs/1611.05431, 2016. URL http://arxiv.org/abs/1611.05431.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. arXiv:1605.07146 [cs.CV], abs/1605.07146, 2016. URL http://arxiv.org/abs/1605.07146.