Weighted Channel Dropout for Regularization of Deep Convolutional Neural Network

Saihui Hou, Zilei Wang
Department of Automation, University of Science and Technology of China
saihui@mail.ustc.edu.cn, zlwang@ustc.edu.cn

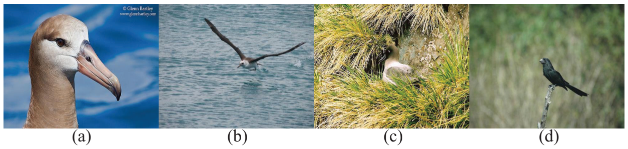

Figure 1: Randomly selected images from CUB-200-2011 for channel activation analysis.

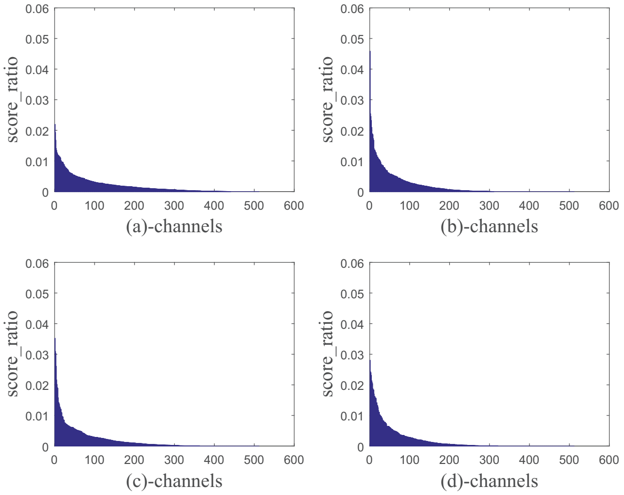

Figure 2: Channel activation analysis in *pool5* of VGGNet-16 pretrained on ImageNet. (a)(b)(c)(d) are corresponding to the images in Figure 1.

Appendix

Channel Activation Analysis

Our WCD is based on the observation that in high convolutional layers of CNN (pretrained on ImageNet or finetuned on the specific dataset), for an input image, only a few channels are activated with relatively high values while the neuron responses in the other channels are close to zero. Here we perform some visualization analysis to validate it. Specifically, VGGNet-16 and CUB-200-2011 are adopted to illustrate the proof. A Global Average Pooling (GAP) layer is added after *pool5* to acquire a global view of activation status in each channel, *i.e.*, assigning a $score_i$ for channel x_i using the notations in the manuscript. For the convenience of visualization, we further compute a $score\_ratio_i$ for channel x_i as

$$score\_ratio_i = \frac{score_i}{\sum_{j=1}^{N} score_j} \tag{1}$$

where N is the total number of channels ($N = 512$ here), $score\_ratio_i \in [0, 1]$. Without loss of generality, four images of different classes are randomly selected from CUB-200-2011 for the analysis as shown in Figure 1. The corresponding $score\_ratio$ in each channel of *pool5* is visualized in the descending order, which is illustrated in Figure 2 (pretrained on ImageNet) and Figure 3 (finetuned on CUB-200-2011). Note that, the maximum value of vertical axis in each figure is set by the maximum $score\_ratio$ of four images. It can be seen that, nearly half of channels hold the responses that are zero or very close to zero.

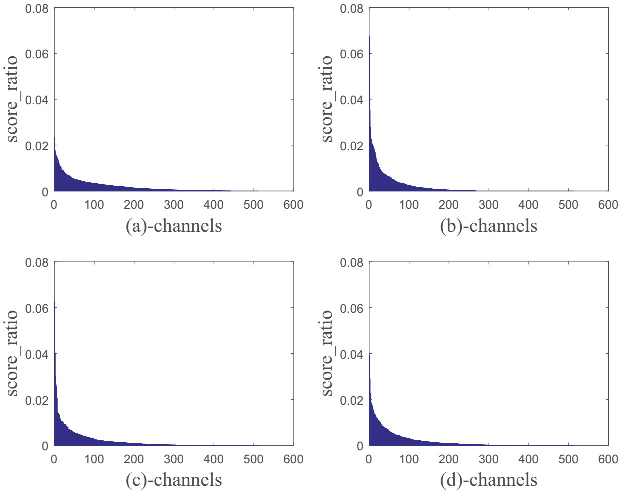

Figure 3: Channel activation analysis in *pool5* of VGGNet-16 finetuned on CUB-200-2011. (a)(b)(c)(d) are corresponding to the images in Figure 1.

Besides, we also provide the channel activation analysis of VGGNet-16 with the proposed WCD in Figure 4. The

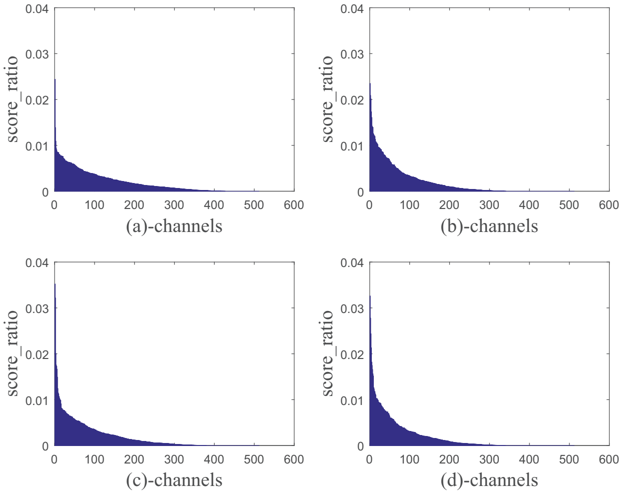

Figure 4: Channel activation analysis in *pool5* of VGGNet-16 with the proposed WCD finetuned on CUB-200-2011. (a)(b)(c)(d) are corresponding to the images in Figure 1.

Table 1: The mean number of channels holding $score\_ratio_i > 0$.

Model	$positive\_cnt$
VGGNet-16 pretrained on ImageNet (Figure 2)	359
VGGNet-16 finetuned on CUB-200-2011 (Figure 3)	346
WCD-VGGNet-16 finetuned on CUB-200-2011 (Figure 4)	372

distribution of $score\_ratio$ in Figure 4 is relatively flatten compared to those in Figure 2 and Figure 3. To make it more clear, for three cases, we count the number of channels holding $score\_ratio_i > 0$ for each image and then compute the mean number, which is denoted as $positive\_cnt$ shown in Table 1. The results indicate that, for a given image, there are more channels activated after the deployment of WCD.

More Ablation Study

In this section, we provide more ablation study to analyze the behaviors of WCD.

More comparison with SE-Block. Compared to SE-Block (Hu, Shen, and Sun 2017), the way to combine WCD with ResNet is a little different (please refer to the paper for details). Actually we have tried to deploy WCD at the same place as SE-Block, however, the performance is a little worse (*e.g.*, 76.60% with ResNet-101 on CUB-200-2011, worse than 77.22% in our settings), which is probably caused by the differences between these two modules.

Deploying WCD at more places. In our experiments, we mainly deploy WCD after the high layers in the stack of convolutional layers, such as *pool4* and *pool5* in VGGNet-16. We have tried to insert WCD at more places such as *pool3*, which, however, cannot bring additional performance gain. The probable reasons are that the low layers consist of less channels (*e.g.*, 256 channels in *pool3 vs.* 512 channels in *pool4* and *pool5*), and the channels in the early layers are more related with each other

Performance on ImageNet. The intuition of WCD is to mitigate the severe overfitting when training CNN on small datasets. For the large-scale ImageNet, we have tried the deployment of WCD and find that the improvement is less significant (*e.g.* 0.28% with VGGNet-16 as the base model). On one hand, ImageNet consists of much more training images (about 1280 images per class) compared to the small datasets such as CUB-200-2011 (about 30 images per class). On the other hand, the modern CNNs are already equipped with multiple regularization methods (*e.g.* Dropout and Weight Decay in VGGNet-16[1]). While in practice, for a specific task, it is usually infeasible to collect such a large-scale dataset, especially in the scenario where the data annotation needs the expertise of specific domain, such as fine-grained visual categorization.

Performance on CIFAR. In addition, we evaluate WCD on another two popular datasets: CIFAR10 and CIFAR100. CIFAR10 is composed of 60k 32×32 images of 10 classes while CIFAR100 collects the same number of images of 100 classes. The images in both datasets are split into 50k for training and 10k for evaluation. We conduct the experiments on CIFAR10/100 with VGGNet-16 as the base model and find that WCD can also bring consistent improvements (*i.e.*, +0.17% on CIFAR10 and +2.22% on CIFAR100). It is worth noting that, the number of training images per class in CIFAR10 is ten times of that in CIFAR100.

Comparison with other regularization methods. As mentioned in the manuscript, our work falls into the scope to add regularization to the neural networks and the previous works in this filed mainly consist of Dropout (Srivastava et al. 2014), DropConnect (Wan et al. 2013), Batch Normalization (Ioffe and Szegedy 2015), DisturbLabel (Xie et al. 2016), Stochastic Depth (Huang et al. 2016). The experiments are mainly performed compared to Dropout and SE-Block[2]. Actually, Dropout can treated as a special case of DropConnect while the evaluated ResNet and Inception are already equipped with BN. DisturbLabel is used to regularize the loss layer, while Stochastic Depth suits for the residual networks. The proposed WCD is fairly generic and complementary to these works. It unifies the original Dropout and Channel-Wise Dropout, and can be combined with BN in modern CNNs to provide more regularization to the convolutional layers.

References

Hu, J.; Shen, L.; and Sun, G. 2017. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*.

Huang, G.; Sun, Y.; Liu, Z.; Sedra, D.; and Weinberger, K. Q. 2016. Deep networks with stochastic depth. In *ECCV*.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerat-

[1] WCD is applied to VGGNet-16 retaining the Dropout in the fully connected layers

[2] WCD shares the similar structures with SE-Block.

ing deep network training by reducing internal covariate shift. In *ICML*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1):1929–1958.

Wan, L.; Zeiler, M.; Zhang, S.; Cun, Y. L.; and Fergus, R. 2013. Regularization of neural networks using dropconnect. In *ICML*.

Xie, L.; Wang, J.; Wei, Z.; Wang, M.; and Tian, Q. 2016. Disturblabel: Regularizing cnn on the loss layer. In *CVPR*.