Is Each Layer Non-trivial in CNN? (Student Abstract)

Wei Wang, Yanjie Zhu, Zhuoxu Cui, Dong Liang,
Center for Medical AI,Institute of Biomedical and Health Engineering, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
Shen Zhen 518055, China
wei.wang1@siat.ac.cn, dong.liang@siat.ac.cn

Abstract

Convolutional neural network (CNN) models have achieved great success in many fields. With the advent of ResNet, networks used in practice are getting deeper and wider. However, is each layer non-trivial in networks? To answer this question, we trained a network on the training set, then we replace the network convolution kernels with zeros and test the result models on the test set. We compared experimental results with baseline and showed that we can reach similar or even the same performances. Although convolution kernels are the cores of networks, we demonstrate that some of them are trivial and regular in ResNet.

Introduction

The structures of neural networks are getting more and more complex. There are two basis forms: short-connection and no-connection. Short-connection:ResNet (He et al. 2015). No-connection: VGG (Simonyan and Zisserman 2014). Particularly, long-connection can be seen as a special no-connection in the local area. Long-connection: UNet (Ronneberger, Fischer, and Brox 2015), SegNet (Badrinarayanan, Kendall, and Cipolla 2017). We say the layers are non-trivial if the performances change slightly after replacing the convolution kernel with zeros, vice versa. It is obvious that each layer in no-connection form is important. But, many layers in short-connection (ResNet) are trivial.

The main contributions of this paper can be summarized as : We donate the non-trivial layers of ResNet are mainly concentrated on feature decomposition layers, which refers to the layers changing the number of channel dimension, when the model is over-parameterized.

Analyze the Convolution Kernels of ResNet Replaced by 0

ResNet residual unit can be formulated as:

$$
\begin{aligned}
\mathbf{x}_{l+1} &= \sigma(\mathbf{x}_l + BN(\sigma(BN(\mathbf{x}_l * \mathbf{w}_l')) * \mathbf{w}_l''))\\
\mathbf{x}_{l+1} &= \sigma(BN(\mathbf{x}_l * \mathbf{w}_l^{1*1}) + BN(\sigma(BN(\mathbf{x}_l * \mathbf{w}_l')) * \mathbf{w}_l''))
\end{aligned}
$$

Replacing one of the convolution kernels in residual unit with $\mathbf{0}$ can be written as:

$$
\begin{aligned}
\mathbf{x}_{l+1}' &= \sigma(\mathbf{x}_l + BN(\sigma(BN(\mathbf{x}_l * \mathbf{0})) * \mathbf{w}_l''))\\
&= \sigma(\mathbf{x}_l + BN(\sigma(\beta') * \mathbf{w}_l''))\\
\mathbf{x}_{l+1}'' &= \sigma(\mathbf{x}_l + BN(\sigma(BN(\mathbf{x}_l * \mathbf{w}_l')) * \mathbf{0}))\\
&= \sigma(\mathbf{x}_l + \beta'')
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{x}_{l+1}' &= \sigma(BN(\mathbf{x}_l * \mathbf{w}_l^{1*1}) + BN(\sigma(BN(\mathbf{x}_l * \mathbf{0})) * \mathbf{w}_l''))\\
&= \sigma(BN(\mathbf{x}_l * \mathbf{w}_l^{1*1}) + BN(\sigma(\beta') * \mathbf{w}_l''))\\
\mathbf{x}_{l+1}'' &= \sigma\left(BN\left(\mathbf{x}_1 * \mathbf{w}_l^{1*1}\right) + BN\left(\sigma\left(BN\left(\mathbf{x}_l * \mathbf{w}_l'\right)\right) * \mathbf{0}\right)\right)\\
&= \sigma\left(BN\left(\mathbf{x}_l * \mathbf{w}_l^{1*1}\right) + \beta''\right)\\
\mathbf{x}_{l+1}''' &= \sigma(BN(\mathbf{x}_l * \mathbf{0}) + BN(\sigma(BN(\mathbf{x}_l * \mathbf{w}_l')) * \mathbf{w}_l''))\\
&= \sigma(\beta''') + BN(\sigma(BN(\mathbf{x}_l * \mathbf{w}_l')) * \mathbf{w}_l''))
\end{aligned}
$$

$\mathbf{x}_l, \mathbf{x}_{l+1}$:The input and output feature maps of the lth residual unit. $\mathbf{x}_{l+1}', \mathbf{x}_{l+1}'', \mathbf{x}_{l+1}'''$: The output feature maps of the lth residual unit. $\mathbf{w}_l', \mathbf{w}_l''$: The first convolution kernel and the second convolution kernel of the lth residual unit. $*$: Convolution operation. BN : Batch normalization operation. $\beta', \beta'', \beta'''$: The bias in BN layers.

Experiment

We chose ResNet34 and PSPNet-ResNet34 (Zhao et al. 2016) to conduct a classification task and image segmentation task on Cifar-10 (Krizhevsky 2012) and T1 (Fahmy et al. 2019), respectively. The baselines are 84% and 87%. We conducted three groups of experiments. Firstly, we replaced each layer's convolution kernel with 0, respectively (see Figure 1 in supplementary material). Secondly, except for the feature decomposition layers and adjacent layers, we replaced all the other convolution kernels with 0 in the one layer block which refers to a continuous layer with the same channel number (see Figure 2 in supplementary material). Thirdly, we replaced feature decomposition layers of short-connection with 0 (see Figure 3 in supplementary material).

Results

The classification results of Cifar-10 are shown in Figure 1 and Table 1,2. The segmentation results of T1 are shown in Figure 2 and Table 3,4.

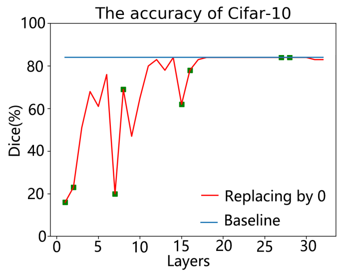

Figure 1: The first experimental results for Cifar-10.

Layer block	ACC(%)
Layer block 1	0.51
Layer block 2	0.61
Layer block 3	0.83
Layer block 4	0.84

Table 1: The third experimental results for Cifar-10.

Layer block	ACC(%)
Layer block 2	0.28
Layer block 3	0.33
Layer block 4	0.16

Table 2: The second experimental results for Cifar-10.

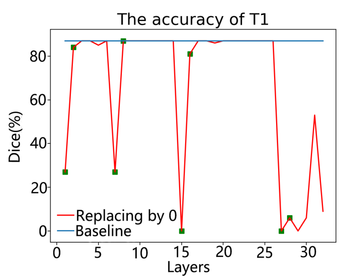

Figure 2: The first experimental results for T1.

Layer block	Dice
Layer block 1	0.82
Layer block 2	0.86
Layer block 3	0.82
Layer block 4	0.00

Table 3: The second experimental results for T1.

Layer block	Dice
Layer block 2	0.00
Layer block 3	0.00
Layer block 4	0.00

Table 4: The third experimental results for T1.

According to the structure of ResNet and the result of Figure 1 and Figure 2, it is obvious that the feature decomposition layers' convolution kernels are non-trivial, while the rests are trivial. Table 1 and Table 3 also confirm our conjecture. Table 2 and Table 4 also demonstrate the feature decomposition layers of short-connection are non-trivial.

Discussion

We argue that ResNet is a continuous process of feature decomposition and information storage. ResNet shows different changes in non-trivialness at the front and back of the network for different tasks. Generally, the classification task needs to learn enough information about the global abstract feature. Since enough information has learned in the front, the back layers are no longer non-trivial. Segmentation requires information for each pixel, so the back layers are non-trivial.

Conclusion

When there are redundant parameters in ResNet, not all layers of the network are non-trivial, or some layers may not be needed when the network parameters have learned enough information. The feature decomposition layers and identity mappings are important. Particularly, the feature decomposition layers are responsible for the feature decomposition, the identity mapping is responsible for the information storage and the residual layers are responsible for the adjustment of the feature to make it fit the final target. According to the above conclusions and experiments, when the model is over-parameterized, we can eliminate unnecessary layers in the ResNet and improve the training efficiency on the premise of ensuring performance.

References

Badrinarayanan, V.; Kendall, A.; and Cipolla, R. 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(12): 2481–2495.

Fahmy, A. S.; El-Rewaidy, H.; Nezafat, M.; Nakamori, S.; and Nezafat, R. 2019. Automated analysis of cardiovascular magnetic resonance myocardial native T1 mapping images using fully convolutional neural networks. *Journal of Cardiovascular Magnetic Resonance* 21(1).

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. *arXiv e-prints* arXiv:1512.03385.

Krizhevsky, A. 2012. Learning Multiple Layers of Features from Tiny Images. *University of Toronto* .

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv e-prints* arXiv:1505.04597.

Simonyan, K.; and Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv e-prints* arXiv:1409.1556.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2016. Pyramid Scene Parsing Network. *arXiv e-prints* arXiv:1612.01105.