Scene classification of remote sensing image based on compound pruning

Fengbing Jiang[1,*], *Fang* Li[2], and *Guoliang* Yang[2]

[1]Science and Technology College of Gannan Normal University, GanZhou, JiangXi, China 341000
[2]School of Electrical Engineering and Automation, JiangXi University of Science and Technology, GanZhou, JiangXi, China 341000

Abstract. Convolution neural network for remote sensing image scene classification consumes a lot of time and storage space to train, test and save the model. In this paper, firstly, elastic variables are defined for convolution layer filter, and combined with filter elasticity and batch normalization scaling factor, a compound pruning method of convolution neural network is proposed. Only the superparameter of pruning rate needs to be adjusted during training. in the process of training, the performance of the model can be improved by means of transfer learning. In this paper, algorithm tests are carried out on NWPU-RESISC45 remote sensing image data to verify the effectiveness of the proposed method. According to the experimental results, the proposed method can not only effectively reduce the number of model parameters and computation, but also ensure the accuracy of the algorithm in remote sensing image classification.

1 Introduction

The task of remote sensing image scene classification is related to many applications. Compared with traditional methods, the accuracy of remote sensing image scene classification based on convolution neural network has been greatly improved. But at the same time, convolution neural network has complex structure, many parameters, high computational cost, and may bring over-fitting to the model due to the lack of data. Based on this, a coarse-grained convolution layer filter pruning method is proposed to overcome the above shortcomings. Han. Song et al. [1] proposed a famous deep model pruning method. At present, the commonly used coarse-grained pruning methods are channel pruning [2] and filter pruning[3]. The channel pruning method makes the scaling factor of the corresponding batch specification layer of the secondary channel close to zero through sparse training strategy, and reduces the model parameters by pruning, and its performance is verified by experiments. The deficiency of this method is that the algorithm is sensitive to hyperparameters, and because of the introduction of the channel selection layer, the network must be transformed into ONNX format in order to deploy effectively. Filter pruning adopts the way of filter soft pruning, model training is carried out after setting pruning interval,

[*] Corresponding author: 258408012@qq.com

pruning is carried out after training, and then pruning interval training is re-set, which is repeated until the network converges.

In this paper, by defining the elastic variable for the convolution layer, a new compound pruning method is proposed by using the filter elastic variable and the batch normalized layer scaling factor. Considering that in the process of pruning the filter and its channel, it is necessary to evaluate the importance of the filter and its channel, so that the proposed coarse-grained filter pruning method has only the superparameter of pruning rate, and the network obtained after pruning is simple and practical. In this paper, no new network layer will be added after pruning.

2 Convolution neural network compound pruning

The compound pruning of convolution neural network mainly includes judging the importance of convolution filter and corresponding channel, and judging whether to delete the filter or not. There is a concept of elasticity in economics. Inspired by this concept, elasticity can be used to describe the interaction between all causal variables. It is assumed that the variable as the cause is an independent variable and the variable affected by it is a dependent variable. Then economic elasticity can be expressed by a formula: elasticity = the proportion of changes in dependent variables / the proportion of changes in independent variables, with the help of this concept. The filter elasticity of each convolution layer in the network can be defined. First, assuming that the loss value of the convolution neural network before the filter pruning is L(W), and the network loss value of a filter parameter w_j in the network after being pruned is$\mathrm{L}(\mathbf{W}|w_j = 0, w_j \in \mathbf{F}_\mathbf{i}^\mathbf{k})$, then the elasticity of the loss value with respect to the parameter wj can be defined as:

$$\Delta \mathrm{L}/\Delta \mathrm{w}_\mathrm{j} = \frac{\left|\mathrm{L}(\mathbf{W})-\mathrm{L}\left(\mathbf{W}|\mathrm{w}_\mathrm{j}=0,\mathrm{w}_\mathrm{j}\in\mathbf{F}_\mathbf{i}^\mathbf{k}\right)\right|}{\mathrm{L}(\mathbf{W})}\bigg/\frac{\mathrm{w}_\mathrm{j}-0}{\mathrm{w}_\mathrm{j}} \tag{1}$$

$\mathbf{F}_\mathbf{i}^\mathbf{k}$in formula (1) is the parameter set of the ith filter in the k convolution layer of the convolution neural network. The change of the loss value is based on the first order Taylor formula approximation.

$$\mathrm{L}(\mathbf{W}) = \mathrm{L}\left(\mathbf{W}|\mathrm{w}_\mathrm{j} = 0, \mathrm{w}_\mathrm{j} \in \mathbf{F}_\mathbf{i}^\mathbf{k}\right) + \frac{\partial \mathrm{L}}{\partial \mathrm{w}_\mathrm{j}}\left(\mathrm{w}_\mathrm{j} - 0\right) \tag{2}$$

As a result, the elastic calculation formula of the parameter w_j is obtained as follows:

$$\left|\Delta \mathrm{L}/\Delta \mathrm{w}_\mathrm{j}\right| = \left|\frac{\partial \mathrm{L}}{\partial \mathrm{w}_\mathrm{j}}\frac{\mathrm{w}_\mathrm{j}}{\mathrm{L}(\mathbf{W})}\right| \tag{3}$$

The ith filter in the kth convolution layer of the convolution neural network is defined as:

$$\mathrm{Elasticity}_\mathrm{i}^\mathrm{k} = \frac{1}{\mathrm{N}_{\mathbf{F}_\mathbf{i}^\mathbf{k}}}\sum\nolimits_{\omega_\mathrm{j}\in\mathbf{F}_\mathbf{i}^\mathbf{k}}\left|\frac{\partial \mathrm{L}}{\partial \mathrm{w}_\mathrm{j}}\frac{\mathrm{w}_\mathrm{j}}{\mathrm{L}(\mathbf{W})}\right| \tag{4}$$

$\mathrm{N}_{\mathbf{F}_\mathbf{i}^\mathbf{k}}$ is the number of parameters of the **i** filter in the **k** convolution layer of the network. Because the number of filter parameters in different convolution layers is different, in order to avoid specifying the clipping rate of each convolution layer in layers, in this paper, in a certain training period, the elasticity of the **i** filter of the **k** convolution layer of the network is defined as:

$$\mathrm{Elasticity}_{i\ \mathrm{epoch}}^{k}=\sum_{1}^{N_{\mathrm{step}}}\frac{1}{N_{\mathbf{F}_i^k}}\sum_{\omega_j\in\mathbf{F}_i^k}\left|\frac{\partial L}{\partial w_j}\frac{w_j}{L(\mathbf{W})_{\mathrm{step}}}\right| \tag{5}$$

Formula (5) is the basis for evaluating the importance of the filter, and $\mathbf{N}_{\mathbf{step}}$ represents the total number of **(epoch)** steps in a training period. It can be seen from equation (5) that the elasticity of the filter can be calculated online in the process of network training, and it does not involve the solution of each order partial derivative, and the greater the elasticity value is, the greater the change of the loss value after deleting the filter is, and the more important the corresponding filter is.

In order to evaluate the importance of the feature graph of each layer of convolution neural network, the scaling factor γ of batch normalization layer can be introduced to judge. Assuming that the input and output of the batch normalization layer are represented by the current batch of $\mathbf{Z}_{\mathbf{in}}$， $\mathbf{Z}_{\mathbf{out}}$ respectively, the output of the batch normalization layer is:

$$\hat{\mathbf{z}}=(\mathbf{Z}_{\mathbf{in}}-\mu_B)/\sqrt{\sigma_B^2+\epsilon}\qquad \mathbf{Z}_{out}=\gamma\hat{\mathbf{z}}+\beta \tag{6}$$

Among them, $\boldsymbol{\mu}_{\mathbf{B}}$ and $\boldsymbol{\sigma}_{\mathbf{B}}$ are the mean and standard deviation of the current batch, γ is the scaling factor, and $\boldsymbol{\beta}$ is the translation factor. These two parameters are obtained by network training. It can be seen from equation (6) that the smaller the scaling factor γ is, the less the output $\mathbf{Z}_{\mathbf{out}}$ of the batch normalization layer is affected by the input $\mathbf{Z}_{\mathbf{in}}$**,** thus it can be determined that the corresponding feature graph of the input channel is less important to the overall performance of the network.

In the comprehensive judgment part, assuming that the pruning rate in the compound pruning method is $\boldsymbol{\rho}$, and the convolution neural network has a total of $\mathbf{N}_{\mathbf{F}}$ filters. All filter elasticity values are calculated and sorted from large to small, and the filter elasticity threshold is set to the elasticity value of the $\mathbf{N}_{\mathbf{F}}\mathbf{(1\text{-}\rho)}$ filter. If the filter elasticity value is greater than the threshold value, the filter is set to 1 (for important), otherwise it is set to zero (for unimportant). At the same time, a similar operation is performed on the scaling factor of the batch normalization layer. When both the convolution filter and the output feature graph of the convolution filter are determined to be unimportant, the algorithm is to cut out the convolution filter and the output feature graph, otherwise the convolution filter and the output characteristic diagram are preserved, and the pruning diagram is shown in figure 1. In figure 1, the elastic threshold of the convolution filter is set to **E**, and the scaling factor threshold of the batch normalization layer is represented by γ.

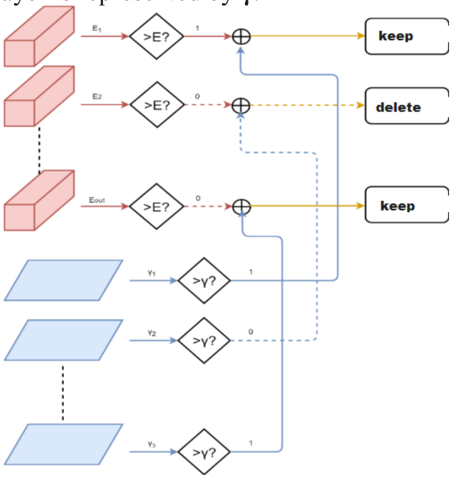

Fig. 1. Schematic diagram of pruning process.

3 Experimental results and analysis

In order to verify the effectiveness of the compound pruning method proposed in this paper, the algorithm is tested on the remote sensing data set NWPU-RESISC45, and compared with other pruning methods. In the experimental environment, GPU is used for Nvidia GTX1080, to program software on the pytorch platform, which is an open source deep learning framework of Facebook.

3.1 Data set

NWPU-RESISC45 dataset is a scene classification dataset of remote sensing images created by Northwestern Polytechnic University. the dataset contains 31500 remote sensing images with a resolution of 256x256. Random clipping and horizontal flipping are also used to enhance the data set. The above data sets are used to train ResNet50[4] and VGG16NB[5] networks respectively, and the compound pruning method proposed in this paper is used to prune them in the training process. The batch sizes of the two kinds of networks are set to 64 and 32 respectively, and the test samples are tailored to the center and the size is 224x224. The division of training set and test set is as follows: (1) 10% training set + 90% test set; (2) 20% training set + 80% test set.

3.2 Experimental results

The ResNet50 and VGG16NB models are trained by the compound pruning method in this paper, and the trained models are tested for remote sensing image scene classification. Table 1 shows the test results of the algorithm on the data set NWPU-RESISC45. It can be seen from Table 1 that the compound pruning in this paper can reduce the network overfitting phenomenon, and the accuracy of the model after pruning is higher than that of the original model. In order to verify whether the pruned model can retain the ability of the original model to locate the key regions of the remote sensing image, this paper also gives the (CAM) comparison of the class activation mapping of the ResNet50 model after layer4. As shown in figure 2, the first is the original image in figure 2, the second is the CAM diagram of ResNet50, the third is the ResNet50CAM diagram obtained by 0.5 pruning rate, and the fourth is the CAM map of ResNet50 obtained by 0.6 pruning rate. Columns 5 to 8 are similar. The above experimental results show that deleting some unimportant filters and corresponding feature graphs have little influence on the classification.

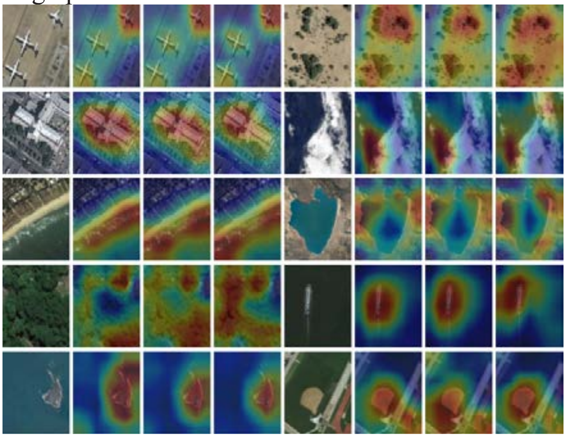

Fig. 2. CAM diagram.

Table 1. Test results of this method on remote sensing data set NWPU-RESISC45.

Resnet50　NWPU-RESISC45 (M: Million，G: Billion)					
Baseline accuracy (20%)	Number of baseline parameters (20%)	Baseline calculation (20%)	0.5 Pruning	0.5 Number of pruning ginseng	0.5 Pruning calculation
92.99%	23.60M	4.13G	92.81%(-0.18%)	18.32M(-22.4%)	3.33G(-19.37%)
			0.6 Pruning	0.6 Number of pruning ginseng	0.6 Pruning calculation
			91.94%(-1.05%)	11.4M(-51.8%)	2.63G(-36.32%)
Baseline accuracy (10%)	Number of baseline parameters (10%)	Baseline calculation (10%)	0.5 Pruning	0.5 Number of pruning ginseng	0.5 Pruning calculation
89.57%	23.60M	4.13G	89.49%(-0.08%)	18.71M(-20.7%)	3.42G(-17.19%)
			0.6 Pruning	0.6 Number of pruning ginseng	0.6 Pruning calculation
			88.72%(-0.85%)	11.42M(-51.6%)	2.67G(-35.35%)
VGG16BN　NWPU-RESISC45 (M: Million，G: Billion)					
Baseline accuracy (20%)	Number of baseline parameters (20%)	Baseline calculation (20%)	0.5 Pruning （Accuracy rate）	0.5 Number of pruning ginseng	0.5 Pruning calculation
92.14%	134.45M	15.65G	91.66%(-0.48%)	126.81M(-5.7%)	10.12G(-35.34%)
			0.7 Pruning	0.7 Number of pruning ginseng	0.7 Pruning calculation
			90.62%(-1.52%)	122.25M(-9%)	6.88G(-56.04%)
Baseline accuracy (10%)	Number of baseline parameters (10%)	Baseline calculation (10%)	0.5 Pruning	0.5 Number of pruning ginseng	0.5 Pruning calculation
88.79%	134.45M	15.65G	88.05%(-0.74%)	126.78M(-5.7%)	10.28G(-34.31%)
			0.7 Pruning	0.7 Number of pruning ginseng	0.7 Pruning calculation
			86.46%(-2.33%)	122.33M(-9%)	7.04G(-55.02%)

4 Conclusion

In this paper, aiming at the problems of convolution neural network in remote sensing image scene classification, such as complex model, large amount of calculation and easy over-fitting, and the number of parameters, calculation, over-fitting and other problems, a compound pruning method is proposed. This method is simple to calculate and easy to use. This method can be used to prune the network to obtain a more effective new network. When this method is applied to remote sensing image scene classification, a better classification accuracy is obtained.

References

1. HAN S, POOL J, TRAN J, et al. Learning both Weights and Connections for Efficient Neural Networks[C]. Neural Information Processing Systems.2015:1135-1143.
2. LIU Z, LI J, SHEN Z, et al. Learning Efficient Convolutional Networks through Network Slimming[C]. The IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2736-2744.
3. YANG HE, GUOLIANG KANG, XUANYI DONG, et al. Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks[C]. The Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), 2018: 2234-2240.
4. LI H, KADAV A, DURDANOVIC I, et al. Pruning Filters for Efficient ConvNets [C].in International Conference on Learning Representations, 2017.
5. He K M，Zhang X Y，Ren S Q，et al. Deep Residual Learning for Image Recognition[C]. Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition Workshops（CVPRW），Las Vegas，NV，F Jun 26-Jul 01，2016. IEEE: New York，2016.
6. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* 2014, arXiv:1409.1556.