ATLANTIS PRESS

Advances in Computer Science Research, volume 87

3rd International Conference on Mechatronics Engineering and Information Technology (ICMEIT 2019)

A Review of Network Compression based on Deep Network Pruning

Jie Yu[1, a], Sheng Tian[2, b]

[1]Naval University of Engineering, PLA Wuhan, China

[2]Hangzhou Dianzi University Hangzhou, China

[a]KG107@163.COM, [b]TIANS_HDU@163.COM

Abstract. In recent years, the deep network has made considerable achievements in the field of computer vision and gradually becomes a hot research topic. The performance of the deep network is very good, however, due to its large size of parameters, high storage, and computational cost, it is hard to deploy the deep network on limited hardware platforms (such as mobile devices). The parameters of the model can express its complexity to some extent, but related studies have shown that not all parameters work in the model. Some parameters are useless, redundancy, and even degrade the performance of the model. Firstly, this paper sorts the results achieved by the scholars domestic and overseas in the field of deep network pruning, and sums up the pruning methods based on single weight granularity, kernel weight granularity and channel granularity; Then, summarizes the effect of the relevant pruning methods on a variety of public deep network models; Finally, it combs the achievements of the current researches and thoughts of network pruning, summarizes the important progress and discusses the future directions.

Keywords: Network pruning, Deep learning, Convolutional neural network.

1. Introduction

Deep network pruning refers to a simplified operation for trained deep network model using pruning method to get a lightweight and equal-accuracy network. After pruning, its structure will be smaller with much fewer parameters, which can reduce the cost of calculation and the storage space. Thus, it can be deployed in the limited-storage hardware environment (such as mobile devices). In this paper, the research about deep network model pruning has been summed up very well, and the effectiveness of pruning has been evaluated systematically. Section 2 introduces the background and rationality of deep network pruning; Section 3 sums up the evolution of the related research methods and the main idea of network pruning methods of single weight granularity, kernel weight granularity and channel granularity separately, and discusses the core steps of each method; Section 4 compares a variety of performance indicators of all kinds of pruning methods based on the public deep network models; Section 5 discusses the direction of the further development of the deep network pruning; Section 6 makes a conclusion about the whole paper.

2. Background

With the development of deep learning[1], it has become one of the hottest subdomains of machine learning, and has been applied successfully in many areas, such as target classification, image recognition, target detection, target tracking and intelligent question answering(Q&A) system[2~6]. In these areas, deep network model has made much more achievements than the traditional methods. There are two main reasons for the success of deep network model. The first one is that the parameters of the model become more and the size of the model becomes larger and deeper. The other one is that the amount of data, including the marked and unmarked, becomes larger and larger. In other words, large model enhances the ability of nonlinear fitting, and big data enhances the ability of the model's generalization.

Deep network model performs very well in many experiments, but it is still restricted by computation time and storage space in practical applications. Because the computation costs too much time so that it still cannot meet the demand of real-time requirements in many applications, in spite of the help of GPU[7]. In addition, the parameters of the large-scale model also take up a lot of

memory space, so it is not applicable to mobile phones or other mobile devices. Therefore, it is a very important research issue to compress the network model without impacting its performance.

The traditional deep network model is typically composed of convolution layers, nonlinear activation layers, down-pooling layers as well as fully connected layers. The characteristics of convolution layers are local-connection and shared weights. So, it takes a lot of time to do feedforward calculation, though there are only a few parameters needed to be trained. In contrast, there are a great number of parameters in the fully connected layer, which reaches more than 80% of all parameters of the network. However, it only takes a little time to forward calculate. Some traditional deep network models perform very well in image classification, such as AlexNet [8] and VGGNet [9]. Usually, these modules can be roughly divided into two categories. The first kind of the modules contains convolution layers [10] and fully connected layers, in which there are a lot of parameters to be trained. The other kind contains nonlinear activation layers and down-pooling layers, in which there is no parameter to be trained. To some extent, the size of the parameters represents the complexity of the model and determine the memory space occupied by the model. Normally, the number of parameters is set after repeated experiments in the laboratory. This kind of local optimum parameter does not represent the real need of the network. And the parameters without pruning are heavily redundant without any thoughts about balancing the cost of computing and effect of recognition. Therefore, to reduce the complexity of the model by pruning the parameters of the model is one of the main directions for the development of the network compression. [11]

3. Network Pruning

In early days, network pruning refers to delete redundant parameters of weights to improve the network's ability of generalization. Reference[12] put forward a method to dynamically select the number of hidden nodes in backpropagation network. Reference[13] combined pruning method with the genetic algorithm to optimize the neural network weight links and to discover new connection modes and structures. After CNN's appearance, the researchers began to reduce the redundant Floating Point Operations(FLOP)[14] to improve the network's efficiency. Network Pruning can be divided into three categories as shown in Fig. 1, Fig. 2 and Fig. 3 according to the pruning granularities, which can be classified as single weight granularity, kernel weight granularity and channel granularity from fine to coarse.

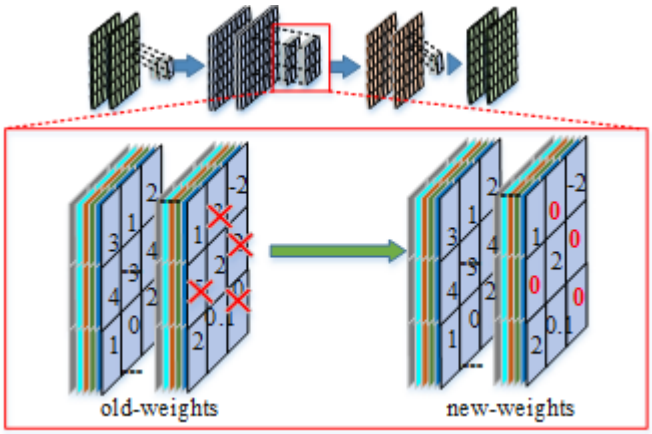

Fig 1. single weight granularity pruning

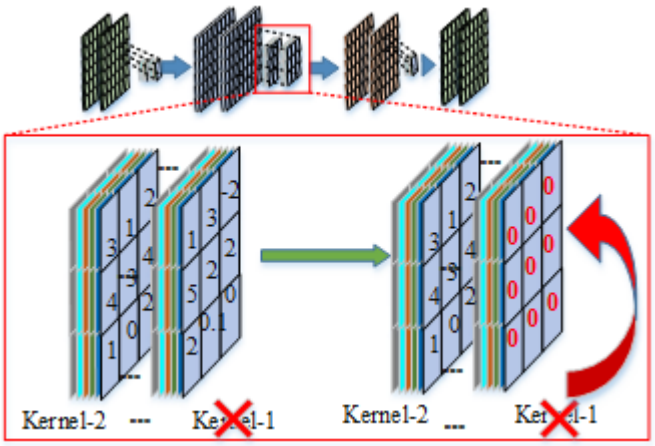

Fig 2. kernel granularity pruning method

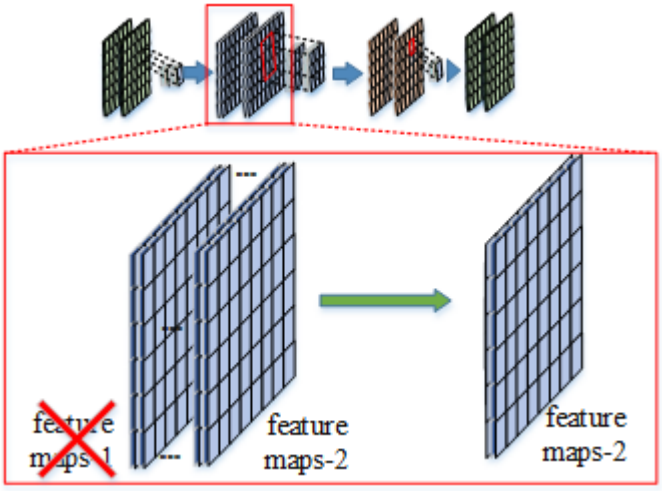

Fig 3. channel/feature map granularity pruning method

What's more, the key points are different from method to method. Some pruning methods focus on the parameters of the network, which don't need to fine-tuning to improve the accuracy after pruning. And some pruning methods are only applicable to pruning the fully connected layer. The following sections are listed according to the classification of pruning granularity order from fine to coarse.

3.1 Single Weight Granularity

In the early, LuCun proposed the OBD (Optimal Brain Damage) method[16], which regards any weight parameter as a single parameter. It can effectively improve the accuracy of prediction, but can't reduce the running time. At the same time, it will take too much time to prune so that it only suits the small network. We set the objective function is $E = f_{NN}(X,U)$, X is the input data, U is the parameter of neural network. Through Taylor expansion of vector function[17], the diagonal hypothesis, the hypothesis of extremum and the second hypothesis[18], we can get the result:

$$\delta E = \frac{1}{2}\sum_{i} h_{ii}\delta u_i^2 \tag{1}$$

Therefore, LuCun put forward a method that fits the significance of the parameters by using the second derivative method.

Then, Hassibi questioned LuCun's method about the usage of diagonal hypothesis and put forward the Optimal Brain Surgeon method, which added a step to update the weights based on the surgery

recovery[17,19]. Optimal Brain Surgeon method acquired a great improvement in accuracy and generalization ability. But both of them need to update the significance of all parameters during iterative computation. In reference[14], hessian matrix and its inverse matrix are also needed to be computed, so that this kind of pruning method cannot be used in large networks.

Be similar to Optimal Brain Surgeon method, Srinivas et al focused on pruning the parameters of the fully connected layer containing most of the parameters[20]. This kind of method can greatly reduce the computational complexity by a lot of deducing approximate methods. Through the exploration of the redundancy and the similarity between the neurons, this method is only dependent on the weights of the network instead of any training data when pruning. So that the accuracy can be recovered without any training data to fine tune. The main idea of this method is to regard the process of pruning as an operation to reset small weights to zero. Surgery recovery method is equivalent to find the similar weight to compensate activation value loss caused by the operation of zero-weight reset. The definition of similar degree between two weights is as follows:

$$s_{i,j} = \left\langle a_j^2 \right\rangle \left\| \varepsilon_{i,j} \right\|^2 \tag{2}$$

Here, $s_{i,j} = W_i - W_j$,which is used to measure the similar degree of weight vector between input node i and j. $\left\langle a_j^2 \right\rangle$ is the average value of node j, representing the degree of proximity between node j and 0. The steps for pruning are as follows: Firstly, calculate the matrix $s_{i,j}$ for all of the possible combinations of weight vectors during the initialization, and then find the minimum of matrix $(i^{'}, j^{'})$ and delete the neuron $j^{'}$. After that, update the weight $a_{j^{'}} \leftarrow a_{i^{'}} + a_{j^{'}}$ and update matrix S through simple operations like deletion and superposition. At last, by experimental observation, Srinivas proposed the automatic pruning method solely depends on the weights[20]. As shown in Fig. 5, The position $s_{i,j}$ in the statistical histogram (the second peak) is the critical point of Fig. 4 in which the error rate will rise sharply. So we only need to delete the nodes whose value is smaller than the critical value.

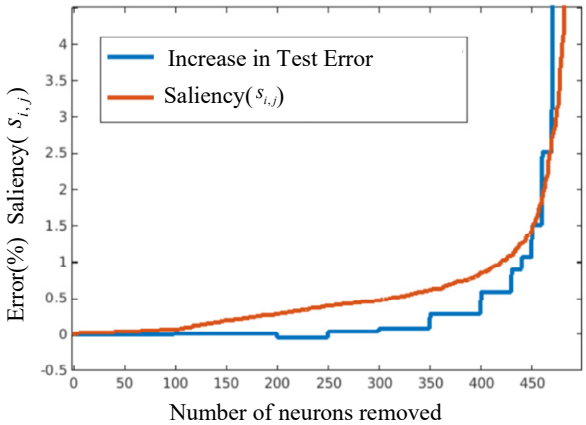

Fig 4. the relation among the saliency, test error rate and Number of neurons removed [20]

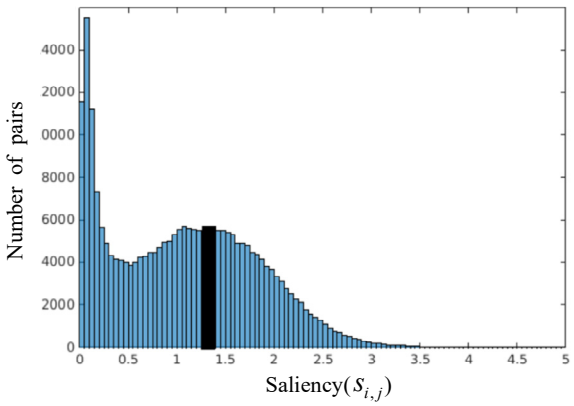

Fig 5. The histogram of saliency values. The black bar indicates the mode of the Gaussian-like curve[20]

3.2 Kernel Granularity

In previous section, any network weight is regarded as a single parameter and is pruned randomly. This kind of pruning method will lead to a problem that network connection will be not neat, so the computer needs to reduce the memory footprint by sparse representation. So that, it takes lots of memory space to mark the locations of the zero or non-zero parameters, during the forward propagation prediction. In ref[21], Han obtained a very good result in compressing the VGG-16 model, but it heavily depended on the special software computing libraries and specialized hardware[22].

Anwar put forward the concept of structured pruning, which makes full use of the sparse regular after pruning to speed up the network's efficiency[23]. It is very easy for the matrix multiplication operation with the existing hardware and software libraries such as BLAS. The innovation of this method is that Anwar puts forward the concept of IntraKernelStrided Sparsity, turning the fine-grained pruning into structured pruning. The steps of this kind of pruning are as follows: First, randomly initialize the step length as m and the offset as n, and set the subscript of the initial item as (i, j) , $i = j = n$, because the convolution kernels are $k \times k$ squares. Then, traverse all of the positions of the figure, such as (n, n) , $(n+m, n)$, $(n, n+m)$ .The key idea of IntraKernelStrided Sparsity pruning method is shown below: The kernel applied on the same input feature map must use the same step length and offset. When convolution layer is not dense connection, kernel's step length and offset in different feature maps can be different; However, if the convolution layer is connected with fully connected layer, all kernels must use the same step length and offset to generated the size matching Lowering KenrelMatrix during the Lowering operation(in cuDNN), so that the size of the kernel matrix and feature map matrix can be reduced down. So a large amount of computing resources can be saved.

What's more, Anwar also put forward another kind of method which uses evolutionary particle filter decided the importance of the network connection[23]. This method doesn't only use the definition of significant measurement but also use the greed-pruning method. x_k is the state vector used to determine whether to cut off a certain connections' weights; Z_k is the observation, which determines the weights of the particles; select $h(\cdot)$ as the observation function of the trained network. The process of particle filter can be described by the following equations. The process of observation is as follows: we can get the h(x), through a Misclassification Rate (MCR), $h(x) = 1 - MCR$, and then get Z_k , under the interference of noise V_k .

$$x_k = f(x_{k-1}) + u_k \tag{3}$$

$$z_k = f(x_k) + V_k \tag{4}$$

The remaining steps still use the traditional methods such as Sequential Importance Resampling (SIR). Ref[15] proved that the Monte Carlo method is better than the method of human-defined significant measurement combined with the greed-pruning method. Particle filter method can get a better result with a lower error rate at the same degree of pruning.

3.3 Channel and Feature Map Granularity

Channel granularity pruning methods don't rely on any sparse convolution calculation library and dedicated hardware. At the same time, They can obtain high compression rate, and greatly reduce the computing time of testing.

Assuming the kernel matrix of convolution layer i is $F_i \in R^{n_i \times n_{i+1} \times k \times k}$, the kernel is $F_{i,\alpha,\beta} \in R^{k \times k}$, Filter is $F_{i,\beta} \in R^{n_i \times k \times k}$, the input matrix of the layer i is $x_i \in R^{n_i \times h_i \times w_i}$, so feature map is $x_{i,\alpha} \in R^{h_i \times w_i}$. The visual figure of dimension reduction can be described by the former signs, as shown in Fig. 6. Convolution kernel matrix of i th convolution layer can be seen as composed of $n_i \times n_{i+1}$ convolution kernels of $k \times k$. Filter effects on the n_i input feature maps and generates new feature maps, so n filters will generate n new feature maps. Thus, the filter can convert 3_d input Y_i to 3_d output Y_{i+1}.

According to the method of Fig. 6 for dimension reduction visualization, we can clearly know the steps of the filter granularity pruning method: firstly, minus the filter of the i th layer; then, minus the feature map generated by i th layer and the kernel generated by $i+1$ th layer.

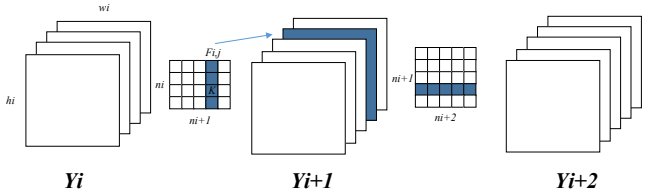

Fig 6. The effect on the i-th and (i+1)-th layers of reducing some filters of layer i[26]

Kernel granularity significance measurement can be simply judged by the sum of kernel weight. Besides, the Inbound Pruning method proposed by Polyak[24] can also realize the measurement. The output feature map is the sum of n_i convolution kernels' results effecting on the corresponding input feature maps:

$$x_{i+1,\beta} = \sum_{\alpha=1}^{n_i} x_{i,\alpha} * F_{i,\alpha,\beta} \tag{5}$$

Use the channel's contribution variance $\sigma_{i,\alpha}$ to measure the significance of the a th convolution kernel, which can be defined as follows:

$$\sigma_{i,\alpha} = \mathrm{var}(\left\| x_{i,\alpha} * F_{i,\alpha,\beta} \right\|_F) \tag{6}$$

$\|\cdot\|_F$ is the European Paradigm of the matrix, the final variance is the variance of feature map paradigm produced by the random input data.

We can choose the sum of the filters' weights as the significance of feature map granularity. The key lies in how to determine the number of pruning and how to realize overall network pruning[25]. Li et al. proposed the HolisticGlobalPruning[26] method, which selects the sum of filter weights as significance, and sorts the filter of every layer according to the significance, and then draws the curve of the weights and the sorted subscripts. If the curve is steep, minus the more filters in this layer; if the curve is flat, minus the fewer filters. It provides the empirical guidance for how to determine the number of Filters needed to prune. The specific number of pruning is the super parameter needed to

optimize. After the number of each layer needed to prune is determined, we can start to apply the Global Greedy Pruning method. “Global” refers to recovering the accuracy by training once again after completing all pruning steps; “Greedy” refers to updating the kernel weights of next layer, after reducing last layer’s filter, so that the pruned kernel will not contribute any significance when pruning the next layer.

Ref[27] found that most of the output neurons are zero after activation function, it means that part of the basic network structure is redundant and not affected by the input data at all. Furthermore, the author deleted those units whose outputs are always zero for different inputs based on the statistical method, and alternate retraining. Ref[28] combined the greed pruning method and fine-tuning method of backpropagation to ensure the generalization of the network after pruning. In particular, the author put forward a method to approximately calculate the change of the loss function after removing some parameters based on the Taylor expansion. Ref[29] also put forward a kind of strategy for pruning the total neurons, the author put forward a kind of maximizing output (Maxout) unit incorporating multiple neurons for more complex express-ion of convex function, and choose them based on the each neuron’s response of partial correlation on the training dataset.

4. Evaluation and Comparison of Pruning Methods

4.1 Public Deep Network Model

Since AlexNet won the championship in ImageNet image classification contest in 2012, there have been a lot of more deeper network models with higher accuracy, such as VGG-16, GoogLeNet, ResNet-101. AlexNet made an outstanding contribution in summarizing a lot of experience to design deep network. Such as the use of replacement for ReLU nonlinear unit, the use of Dropout to avoid overfitting, the use of MaxPooling. VGG’s contribution is the use of the smaller receptive field, such as 3x3 convolution kernels. But its forward propagation is very time-consuming because VGG adopts more channels to increase the width of the network in the former convolution layers’ kernels. GoogLeNet uses inception structure to increase network’s width and improve the accuracy, which reduces the computational cost at the same time. ResNet promotes the flow of information with simple “Bypass” and reduces the possibility of gradient disappearance so that the depth of the network even reaches 152 layers and the accuracy gets greatly improved. Performance and number of parameters in typical deep networks on Keras basing on the ImageNet’s dataset are shown in the following table.

Some results using multiple models polymerization[30] can get a lower error rate. For the convenience of comparison, we only list the statistical information for the single model about the indicator for Top-5 in table 1. The numbers of the parameters are from statistical results basing on Caffe, an open source frame model.

Table 1. Performance and Numbers of Parameters in Typical Deep Networks on Caffe

Deep network models	Top-5 accuracy	The number of parameters can be trained
AlexNet	74.46%	60.9M
VGG-19	92.5%	138.4M
GoogLeNet	92.33%	23.6M
ResNet-50	93.29%	23.7M
ResNet-101	93.95%	42.7M
ResNet-152	94.29%	58.5M

4.2 The Evaluation Indicators of Compression Effectiveness

Network compression evaluation indicators include effici-ency, the compression ratio of parameters and accuracy. When comparing with the benchmark model to measure the performance improvement, we can use ascend multiples (Speedup) or ascending scale (Ratio), and they can transform into each other. In this paper, we use the Ratio to measure. At present, most of the research

works choose Top-1 accuracy as a measurement. Because Top-5 accuracy is only used on a few of occasions, such as ImageNet classification dataset. For the convenience of comparison, in table 2, we choose the Top-1 accuracy as the measurement in this paper.

The evaluation indicators for the ratio of parameter compression are relatively unified, we usually convert all units into Byte and keep two significant figures for the approximate estimate. In the network operation efficiency, we can evaluate it from three aspects: the number of floating point arithmetic times(FLOP), the number of frequency multiplication times(MULTS) and the time of average forward propagation. The results are shown in table 2 and table 3.

Table 2. Models' Top-1 Accuracy and Parameters' Size

ref	Model (O: original) (P: pruning)	Top-1 Accuracy	Parameters (Byte)
[26]	VGG-16(O)	93.25%	1.50E+07
	VGG-16(P)	93.40%	5.40E+06
	ResNet-56(O)	93.04%	8.60E+05
	ResNet-56(P)	93.06%	7.30E+05
	ResNet-56(O)	93.53%	1.72E+06
	ResNet-56(P)	93.30%	1.16E+06
	ResNet-101(O)	73.23%	2.16E+07
	ResNet-101(P)	72.56%	1.99E+07
[20]	LeNet(O)	99.06%	1.96E+07
	Amplitude (P)	96.50%	1.65E+07
	Random (P)	91.37%	1.65E+07
	Data-free (P)	98.35%	1.65E+07
	AlexNet(O)	57.84%	6.09E+07
	Data-free on FC6(P)	56.08%	4.23E+07
	Data-free on FC7(P)	56.00%	5.37E+07
	Data-free on FC6&7(P)	55.60%	3.97E+07

Table 3. The Performance of Pruning Method

ref	Model (O: original) (P: pruning)	Compres-sion ratio	Promotion of Efficiency
[26]	VGG-16(O) VGG-16(P)	\ 64.00%	(FLOP) 13.70%
	ResNet-56(O) ResNet-56(P)	\ 15.12%	(FLOP) 38.60%
	ResNet-56(O) ResNet-56(P)	\ 32.56%	(FLOP) 34.20%
	ResNet-101(O) ResNet-101(P)	\ 7.87%	(FLOP) 13.70%
[20]	LeNet(O) Amplitude (P) Random (P) Data-free (P)	\ 16.01% 16.01% 16.01%	These pruning methods can't significantly increase efficiency, due to FCs hardly affect the running time
	AlexNet(O) Data-free on FC6(P) Data-free on FC7(P) Data-free on FC6&7(P)	\ 30.57% 11.80% 34.89%	

5. The Directions of Future Research

Network pruning is an effective method to realize the compression of the network. The purpose of the deep network compression is to extract the useful information in the network. Here are some directions those are worth to study and explore:

5.1 How to Measure the Influence Degree of the Weights' Parameters on the Results

The result of the deep network is formed by all of the parameters together, and we still use the simple method to measure the importance of a single convolution kernel weight. Ref[14] gives a more detailed method to analysis, but it isn't practical at all due to its heavy difficulty in calculating. Therefore, it is very significant to figure out how to find a more efficient way to approximately measure the effect of every individual parameter on the model.

5.2 The Compression for Specific Task or Usage Scenario

Large networks are usually trained on large datasets. For example, the model trained on ImageNet is for classification of 1000 kinds of objects. But in some specific scenarios of applications, we may only need a small model that can identify several classes. Therefore, it is very significant to figure out how to compress a fully functional network to get sub-networks which can meet the needs of some specific application scenarios.

5.3 Evaluation of Network Compression Utility

Now, the evaluations of compression algorithm for all kinds of deep network is not unified. We usually focus on the comparison of the number of parameters and the running time of network. Besides, we need to focus on some more generalized compression evaluation standards. On the one hand, we can consider the running speed and the influence of model size under different scenarios. On the other hand, we can evaluate the model after the compression in the structure of the model.

6. Conclusion

Network weight pruning method focus on reducing the model's parameters, the storage space, and the operation cost. It plays a more and more important role in the practical applications of the deep network. This paper summarizes the network pruning methods and provides the corresponding indicators on the evaluation of compression performance. For the method of single weight granularity, it can effectively improve the prediction accuracy, but it cannot reduce the running time. At the same time, it will take too much time to prune so that it only suits the small network; For the method of kernel weight granularity, it takes lots of memory space to mark the locations of the zero or non-zero parameters during the forward propagation prediction. So, it is not suitable for parallel computing; For the method of channel granularity, it doesn't rely on any sparse convolution calculation library and dedicated hardware. And it can obtain high compression rate, at the same time, greatly reduce the computing time of testing. I sincerely hope that readers could obtain a comprehensive understanding of deep network pruning through the above introduction, and make full use of these methods in the practical applications.

References

[1]. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521(7553): 436-444.

[2]. Carvalho S R, Cordeiro Filho I, De Resende D O, et al. A Deep Learning Approach for Classification of Reaching Targets from EEG Images[C]. Graphics, Patterns and Images (SIBGRAPI), 2017 30th SIBGRAPI Conference on. IEEE, 2017: 178-184.

[3]. Wan J, Wang D, Hoi S C H, et al. Deep learning for content-based image retrieval: A comprehensive study. In: Proceedings of the22nd ACM international conference on Multimedia (MM). Orlando: ACM, 2014. 157-166.

[4]. Girshick R. Fast r-cnn.In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). Santiago: IEEE, 2015.1440-1448.

[5]. Wang N, Yeung D Y. Learning a deep compact image representation for visual tracking.In: Advances in neural information processing systems (NIPS). Tahoe: IEEE, 2013. 809-817.

[6]. Severyn A, Moschitti A. Learning to rank short text pairs with convolutional deep neural networks. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. Santiago:ACM, 2015. 373-382.

[7]. Ngiam J, Coates A, Lahiri A, et al. On optimization methods for deep learning. In: Proceedings of the 28th international conference on machine learning (ICML). Bellevue: ACM, 2011. 265-272.

[8]. Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems (NIPS). Tahoe:IEEE, 2012. 1097-1105.

[9]. Sercu T, Puhrsch C, Kingsbury B, et al. Very deep multilingual convolutional neural networks for LVCSR. In: Acoustics, Speech and Signal Processing (ICASSP). Shanghai: IEEE, 2016. 4955-4959.

[10]. Abdel-Hamid O, Deng L, Yu D. Exploring convolutional neural network structures and optimization techniques for speech recognition[C]. Interspeech. 2013: 3366-3370.

[11]. Kim Y D, Park E, Yoo S, et al. Compression of deep convolutional neural networks for fast and low power mobile applications[J]. arXiv preprint arXiv:1511.06530, 2015.

[12]. Hanson S J, Pratt L Y. Comparing biases for minimal network construction with back-propagation. In: Advances in neural information processing systems (NIPS). Denver: IEEE, 1989. 177-185.

[13]. Whitley D, Starkweather T, Bogart C. Genetic algorithms and neural networks: Optimizing connections and connectivity. Parallel computing, 1990, 14(3): 347-361.

[14]. Oberman S F, Flynn M J. Design issues in division and other floating-point operations. IEEE Transactions on computers, 1997, 46(2): 154-161.

[15]. Sajid Anwar, Wonyong Sung. Coarse Pruning of Convolutional Neural Networks with Random Masks. In: International conference on learning and representation(ICLR). France: IEEE, 2017. 134-145.

[16]. LeCun Y, Denker J S, Solla S A, et al. Optimal brain damage.In: Advances in neural information processing systems (NIPS). Denver: IEEE, 1989. 598-605.

[17]. Rosenblueth E. Point estimates for probability moments. Proceedings of the National Academy of Sciences, 1975, 72(10): 3812-3814.

[18]. Walter B, Horender S, Voegeli C, et al. Experimental assessment of Owen's second hypothesis on surface shear stress induced by a fluid during sediment saltation[J]. Geophysical Research Letters, 2014, 41(17): 6298-6305.

[19]. Hassibi B, Stork D G. Second order derivatives for network pruning: Optimal brain surgeon. In: Advances in neural information processing systems (NIPS). Denver:IEEE, 1993. 164-171.

[20]. Srinivas S, Babu R V. Data-free parameter pruning for deep neural networks.In: 26th British machine vision conference (BMVC). Swansea: IEEE, 2015. 120-129.

[21]. Han S, Mao H, Dally W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In: International conference on learning and representation(ICLR).San Juan: IEEE,2016. 233-242.

[22]. Han S, Pool J, Tran J, et al. Learning both weights and connections for efficient neural network. In: Advances in Neural Information Processing Systems. Montreal: IEEE,2015. 1135-1143.

[23]. Anwar S, Hwang K, Sung W. Structured pruning of deep convolutional neural networks. ACM Journal on Emerging Technologies in Computing Systems (JETC), 2017, 13(3): 32.

[24]. Polyak A, Wolf L. Channel-Level Acceleration of Deep Face Representations. IEEE Access, 2015, 3: 2163-2175.

[25]. Figurnov M, Ibraimova A, Vetrov D P, et al. PerforatedCNNs: Acceleration through elimination of redundant convolutions. In: Advances in Neural Information Processing Systems (NIPS). Barcelona: IEEE, 2016. 947-955.

[26]. Li H, Kadav A, Durdanovic I, et al. Pruning Filters for Efficient ConvNets.In: International conference on learning and representation(ICLR). France: IEEE, 2017. 34-42.

[27]. Hu H, Peng R, Tai Y W, et al. Network Trimming: A data-driven neuron pruning approach towards efficient deep architectures. In: International conference on learning and representation (ICLR). France: IEEE, 2017. 214-222.

[28]. Molchanov P, Tyree S, Karras T, et al. Pruning convolutional neural networks for resource efficient transfer learning. In: International conference on learning and representation (ICLR). France: IEEE, 2017. 324-332.

[29]. Rueda F M, Grzeszick R, Fink G A. Neuron Pruning for Compressing Deep Networks using Maxout Architectures.In: German conference on pattern recognition (GCPR). Saarbrücken: Springer, 2017.110-120.

[30]. Antipov G, Berrani S A, Dugelay J L. Minimalistic CNN-based ensemble model for gender prediction from face images. Pattern recognition letters, 2016, 70: 59-65.