Research and Analysis on the Application of Lightweight Algorithm Optimization in Complex Scenarios

Puyuan Li[1*]

[1]School of Computer Science, Zhuhai College of Science and Technology, 519000, Zhuhai, China

Abstract. With the rise of deep convolutional neural network and continuous development of artificial intelligence, remarkable achievements have been made in the field of target detection model. However, the traditional target detection algorithm based on deep learning usually needs huge computing resources and long-time training, which is difficult to deploy in edge computing devices, embedded devices, real-time monitoring and other environments. Therefore, the research and design of lightweight algorithm is one of the current research hotspots. This paper summarizes the operation effect of various lightweight algorithm optimization in complex scenes, aiming to further explore new ideas and new directions of lightweight network design. Firstly, the lightweight optimization strategies are divided into three categories: model pruning, knowledge distillation and low-rank Factorization. The performance, advantages and limitations of each kind of target detection task in complex environment are analyzed and introduced in detail. Finally, the research contents are summarized, and the challenges and potential directions of future work are discussed.

1 Introduction

Nowadays, the demand for target detection algorithm is increasing because of the continuous development of artificial intelligence. Target detection has become more and more significant in the field of computer vision. To identify and locate specific targets in images or videos is its purpose. Target detection algorithm is widely used in intelligent transportation system, video surveillance system, medical image analysis and other fields. In 2014, Shick et al. Integrated Convolutional Neural Networks (CNN) into target detection for the first time, and proposed the Region-CNN algorithm, which significantly improved the mean average precision of target detection Mean Average Precision (mAP) [1]. Since then, the target detection algorithm based on CNN framework has sprung up.

However, traditional target detection methods based on manually designed features and classifiers have many limitations. Although the new target detection algorithm significantly improves the detection accuracy, it often has high computational cost and memory occupation, which limits its application in edge computing devices and embedded devices. Therefore, the lightweight of the algorithm is very important. There are several ideas for

* Corresponding author's email: Li102720@outlook.com

designing lightweight target detection network; Design artificial lightweight network, including innovative convolution layer design, efficient activation function and new network connection mode; Design lightweight network using neural architecture search Neural Architecture Search (NAS); Use method like model pruning, low-rank factorization and knowledge distillation to compressed the network.

This paper focuses on the methods of model pruning, knowledge distillation and low-rank Factorization. Model pruning is a key technology to optimize the deep learning model, which decreases the number of parameters and calculation of the model by eliminating the "unimportant" weights or neurons in the model, and realizes the sparseness of the model; The core of knowledge distillation is the "teacher student" architecture, in order to improve the performance of student network, it uses a large network (called teacher network) to guide the small network (called student network). This guidance comes from the probability vector output by the teacher network. The student network takes it as the learning goal to obtain the relationship between classes learned by the teacher network from the data set, which is the definition of "knowledge" in knowledge distillation. The "distillation" comes from the temperature parameter, which acts on the softmax layer that generates the probability vector, making the relationship between classes easier to learn. In the training process of knowledge distillation, the temperature parameter is set to a larger value, but when the student network is actually applied, the temperature parameter is not used, which is the meaning of knowledge distillation; Low-rank factorization initially involves identifying an approximate representation of a lower rank, followed by decomposing the original weight matrix into the product of several lower rank matrices. Large convolution kernels and higher dimensional weight matrices often cause model parameters to occupy a lot of storage space and computing resources. Based on the information redundancy and low rank characteristics of the weight matrix, the model can be compressed by low-rank factorization. It can effectively decrease the storage requirements and computational complexity. This paper compares the advantages and limitations of compression methods in complex scenes, and gives the outlook for the future.

2 Model pruning

As shown in Fig. 1, model pruning is to remove redundant parameters that contribute little to performance in the model according to the evaluation criteria of the importance of network parameters. Pruning is designed to remove unimportant parts in the neural network to improve the reasoning efficiency of the model and decrease the space occupation of the model.

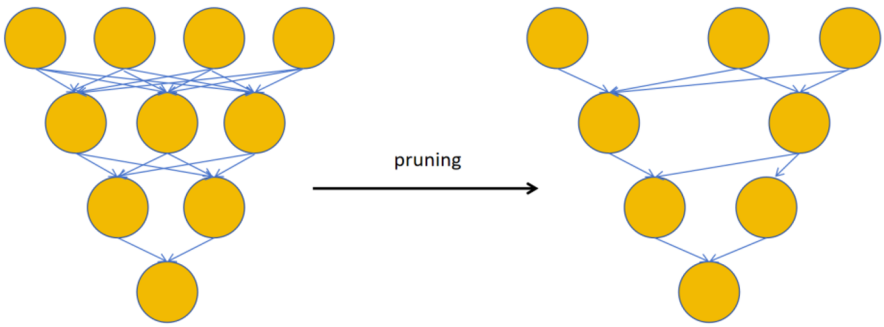

Fig. 1. Schematic diagram of pruning process (Picture credit: Original)

2.1 Unstructured pruning

Unstructured pruning simplifies the network structure by setting the weight value or its mask that contributes less to the final output of the model to 0.

$$M_i = \begin{cases} 0, M_i < \theta \\ M_i, M_i > \theta \end{cases} \quad (1)$$

Where, M_i represents a weight or mask of weight in the network, and θ represents the threshold calculated by some algorithm. Because unstructured pruning is the modification of the most basic elements in the network, its advantage is that it is easy to implement and will hardly affect the performance of the network. For example, Guo et al. Proposed to delete redundant connections through continuous network maintenance[2]. Although unstructured pruning can get better compression effect, the network compressed by unstructured pruning is difficult to be directly deployed on the cutting-edge lightweight equipment, mainly because the cutting-edge lightweight equipment is inefficient in processing sparse matrix. That is, the arrangement of models after unstructured pruning is irregular, and the processor still takes time to deal with the pruned places.

2.1.1 Unstructured pruning method based on amplitude

The amplitude-based pruning approach eliminates neurons or weights that make relatively minor contributions to the model, thereby reducing its overall complexity. However, in the pruning process of the large language model (LLM), the complex internal structure of the LLM is ignored, resulting in the irregular internal structure of the model after pruning, resulting in serious performance damage of the model after pruning, which needs to be retrained to regain accuracy. Gordon et al. Conducted an amplitude based pruning study on the Bert model [3]. The results show that about 30% -40% of the weights in the model are redundant, and pruning the redundant weights does not affect the performance of the model.

2.1.2 Unstructured pruning method based on loss

The pruning method based on loss function is a kind of technology to realize model compression by optimizing loss function in the process of neural network training. The core idea is to determine which weights can be removed by evaluating the impact of model weights on model performance (usually loss function) without affecting model performance, so as to decrease the complexity and computational cost of the model and improve operation efficiency. SparseGPT introduces a one-time pruning strategy without retraining [4]. This method regards the pruning process as a generalized sparse regression problem, and uses an approximate sparse regression solver to solve this problem, so as to decrease the amount of calculation of training and reasoning, and improve the efficiency and speed of model pruning. The complex weight update process, including the use of synchronous second-order Hessian update, bypasses the traditional retraining and avoids the overhead and difficulty of retraining, making it more suitable for pruning LLM with large scale.

2.2 Structured pruning

Unstructured pruning corresponds to structured pruning. Compared with fine-grained elements such as pruning weights, structured pruning takes coarse-grained elements such as filters and network layers as pruning objects. This method is easy to be accelerated by hardware, decrease memory consumption and accelerate reasoning. The research focus of structured pruning is how to evaluate the importance of different structures. For instance, Li et al. Evaluated the L1 norm of the convolution kernel in a single layer[5], that is, added the

absolute values of each parameter in the convolution kernel to obtain the L1 norm of the convolution kernel, and finally sorted according to the L1 norm in the same convolution layer.

2.2.1 Structured pruning method based on amplitude

Structural pruning can maintain the integrity of the overall structure of the LLM by pruning the weight of the entire row or column in the model, decrease the complexity of the model and memory usage, and provide an effective solution for the lightweight deployment of the LLM. Saleh et al. Proposed a pruning method after training SLICEGPT [6], which decreases the embedding dimension of the network by replacing each weight matrix with a smaller (dense) matrix to achieve the purpose of reducing model parameters. The specific implementation method is to set the entire row (column) of the weight matrix in the LLM to zero, and compensate the sparse part by updating the surrounding elements of the matrix. After the weight sparse update process, a smaller weight matrix is obtained to decrease the dimension of the weight matrix, and then decrease the embedding dimension of the neural network, so as to decrease the calculation requirements of the model and maintain the performance of the model to the greatest extent.

2.2.2 Structured pruning method based on loss

In the process of structural pruning of LLM, gradient information plays an important role in evaluating the importance, determining the pruning unit and amplitude, and fine-tuning the recovery. However, different algorithms use gradient information with different emphasis. LLM pruner and Lo-RAshear use gradient information by defining pruning structure and ShortLLaMA by selecting pruning target. LLM pruner accurately locates the coupling structure in the model and constructs the coupling structure group by identifying the coupling relationship between the model structures. The importance of a single weight is quantified by the error caused by removing the weight parameter. Taylor expansion of error function caused by weight removal, and the importance of the weight is estimated by using the importance estimation method of approximate second-order Hessian information. Finally, the importance of the coupling group is summarized by summing, multiplying or other methods to determine the overall importance of the coupling group. After evaluating the importance of each group, the non critical coupling structure is pruned first according to the predefined pruning ratio to decrease the amount of model parameters [7].

2.3 Experimental data

The model pruning algorithm should decrease the number of parameters and floating-point operations of the model as much as possible without loss of model precision or small loss of model precision. The higher the compression ratio, the better the performance of pruning method. From the experimental results of classical algorithms, this paper finds the results of different pruning algorithms for ResNet-50 network on ILSVRC-2012 dataset, as shown in Table 1.

Table 1. Performance of different algorithms on ResNet-50 network and ILSVRC-2012 dataset

Model pruning method	**Model parameter compression ratio %**	**FLOPs compression ratio%**	**Top-1 loss of accuracy%**
SFP30% [8]	70	58．2	-1．54
GBN-50 [9]	53．4	55．06	-0．67
Filtersketch-0.6 [10]	43．0	45．5	-1．63

3 Knowledge distillation

Knowledge distillation means letting a "student model" with poor performance on the dataset learn the pseudo tags generated by a "teacher model" with good performance on the dataset to obtain the excellent performance of the "teacher model" on the dataset. As shown in Fig. 2, the teacher model has a complex structure and powerful performance, and can make accurate predictions, while the student model is smaller, with lower storage requirements and computational complexity. Knowledge distillation can decrease model size and maintain or improve model performance by transfers knowledge from teacher model to student model.

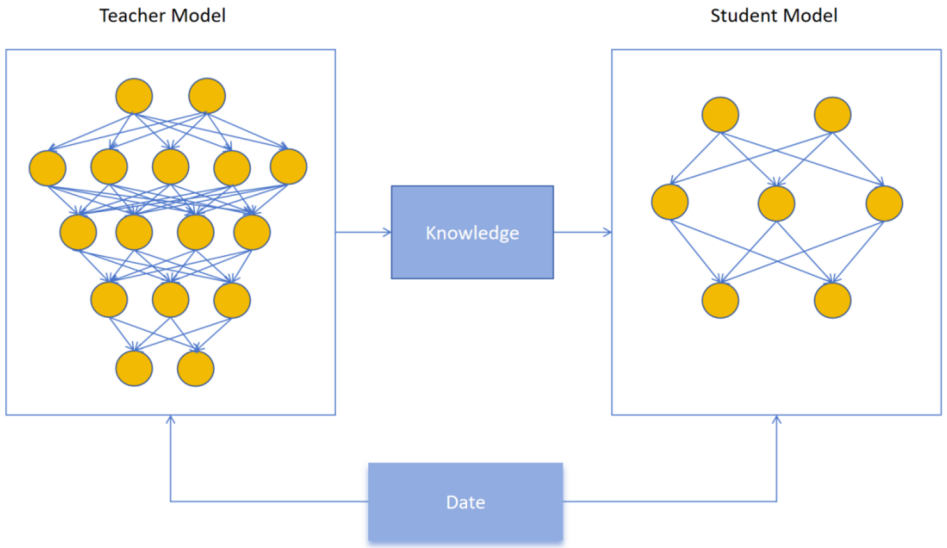

Fig. 2. Schematic diagram of knowledge distillation (Picture credit: Original)

3.1 Online distillation

Online distillation is also called deep mutual learning. In this structure, there is no need for a pre trained teacher network, but multiple student networks are used to learn at the same time, and the knowledge distillation method is used to learn from each other. The loss function of online distillation can be expressed as

$$L_{DML} = \sum_{k} \mathrm{L_t}\,(\mathrm{l_i}, \mathrm{t}) + \sum_{\mathrm{i<j}} \mathrm{L_{KD}}(\mathrm{l_i}, \mathrm{l_j}) \tag{2}$$

l is the output of each sub network and t is the task target. Similar to traditional knowledge distillation, the improvement of online distillation mainly focuses on adding supervision information to the multi-student structure. Lan et al. Used the learning method to integrate the output of the student network in the training process to form an online "teacher network"[11], which can provide additional supervision for all student networks while learning the real value.

Online distillation has achieved two goals: the teacher network independent of pre training and the improvement of network performance. However, the new research found that a composite teacher network is still needed in this framework to further improve the network performance, which will lead to the problem that it is difficult to train multiple networks at the same time.

3.2 Multi teacher distillation

In order to make up for the limited supervision that a single teacher network can provide, some work uses multiple teacher networks to increase more supervision information, that is, multi teacher distillation. The advantage of multi teacher distillation is that it can provide more diversified information in addition to the softening label in traditional knowledge distillation, but it has the disadvantage of high difficulty in obtaining teachers' network. Polytheism.The loss function of distillation can be expressed as

$$L_{MTD} = L_t(l_s, t) + L_{KD}\left(l_S, E(l_{t1}, l_{t2}, \cdots)\right) \quad (3)$$

E (·) refers to the output integration function of the teacher network. The research focus of multi teacher distillation is mainly on how to select the teacher network. Liu et al. Calculate the inner product of teacher network characteristics and student network characteristics to get the matching weight by the attention mechanism, and added the hint loss and the angle loss based on structural knowledge [12].

The model of multi teacher distillation can provide more abundant and diverse supervision information for student networks, but it also increases the cost of using knowledge distillation methods. How to choose the appropriate teacher network and how to train the teacher network for diversified knowledge need further study.

3.3 Experimental data

As an auxiliary training method, knowledge distillation uses the same data set and training method as the corresponding task. All experiments were conducted on the CIFAR10 (Canadian Institute for Advanced Research) dataset commonly used in classification tasks. The images of this dataset are 32 × 32 pixels, with a total of 50000 training set data and 10000 test set data. For fair comparison, ResNet56 is used as the teacher network, ResNet20 is used as the student network, and the same super parameters (including learning rate, learning rate attenuation, temperature parameters, etc.) are used. The classification accuracy of three experiments was used as the final performance evaluation standard. Table 2 shown the experimental results.

Table 2. Performance comparison of knowledge distillation method on CIFAR10 dataset

Types	Method	Classification accuracy%	Notes
Direct training	ResNet56 [13]	94.01	
	ResNet20 [13]	92.36	
Online distillation	ONE [11]	93.08	3 student network
Multi teacher distillation	ALTKD [12]	92.53	3 teather network

4 Low-rank factorization

The model can be compressed by low-rank factorization based on the information redundancy and low rank characteristics of the weight matrix. The larger convolution kernel and the higher dimension weight matrix often cause the model parameters to occupy a lot of storage space and computing resources. As shown in Fig. 3, Low-rank factorization involves decomposing the original weight matrix into the product of several low-rank matrices by seeking an approximate lower-rank representation, thereby effectively reducing computational complexity and storage demands.

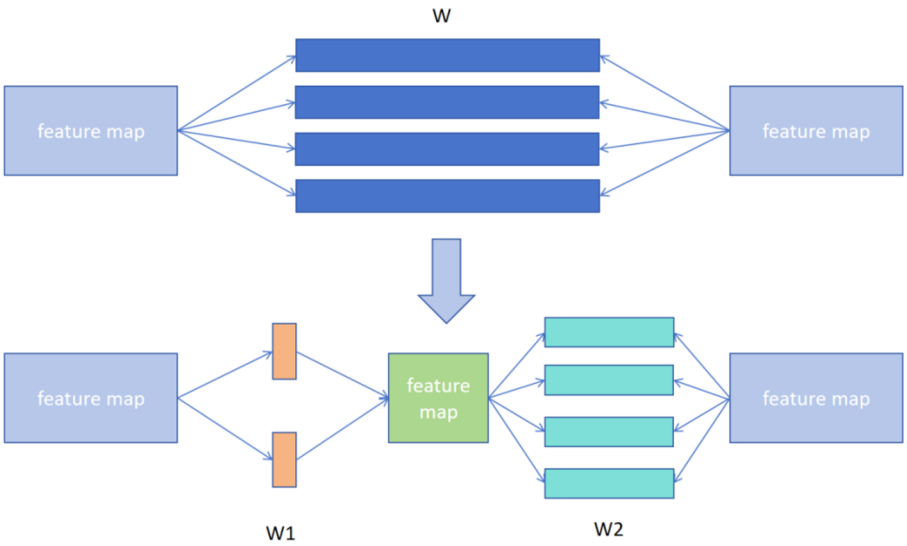

Fig. 3. Schematic diagram of low-rank Factorization (Picture credit: Original)

4.1 Singular value decompose

The main idea of singular value decompose is to factorization a matrix into the product of three matrices: u, Σ, vt. Where Σ is a diagonal matrix, u and V are orthogonal matrices,the elements on the diagonal are called singular values. These singular values represent the stretching degree of the matrix in each principal direction, while u and V represent the projection of the data set in the new coordinate system after dimension reduction and the coordinates of each data point in the new coordinate system. Model compression based on singular value decomposition usually refers to approximating a large matrix by preserving its main singular values and corresponding singular vectors, so as to decrease the storage and computational complexity of the model. Yang et al. Who make Deep Neural Network (DNN) reach low rank by used SVD training method[14], so as to decrease memory and computational burden, and proposed orthogonal regularization and sparsity induced regularization techniques to achieve model compression without significantly reducing model performance, providing a new method for DNN deployment on resource constrained platforms.

4.2 Tensor decompose

Tensor decomposition can be regarded as a generalization of matrix decomposition. For instance, a two-dimensional matrix can be regarded as a second-order tensor, and the weight of convolution layer is represented by 4-D tensor. In theory, employing more flexible algorithms can lead to a higher compression ratio as the tensor order increases. Yin et al. introduced a systematic tensor decomposition framework grounded in the Alternating Direction Method of Multipliers (ADMM) [15]. In this framework, the TT decomposition model compression problem is transformed into a constrained nonconvex optimization problem, which is solved by ADMM iteration. In this process, the whole DNN model gradually shows the low tensor rank characteristics, and then it is decomposed into TT format

and fine tuned, finally a high-precision TT format DNN model is obtained, but the model training time is long.

5 Conclusion

Model pruning is to use some algorithm to judge and prune the weight or structure in the model, and then fine tune it. Its advantage is that for some models with redundant parameters, its generalization may be improved. But the relative disadvantage is that after pruning, the model may need specific hardware support, and it is prone to the problem of large decline in accuracy, and how to judge the objects in the model that need to be pruned is also difficult. In the future, researchers can study the automation of model pruning, and design different solutions according to different network layers to solve the difficulties in the process of pruning.

In knowledge distillation, it is necessary to select the knowledge transferred from the teacher model to constitute a portion of the loss function for the student model, followed by fine-tuning. Its advantage is that the student model can use the pseudo tags which output by the teacher model better, so as to decrease the dependence on a large number of labeled data. But the relative disadvantage is that more complex models and training processes are needed to establish and optimize the knowledge transfer from teacher model to student model, but that may lead to the need for more computing resources and time, and how to determine which parts to select as "knowledge" is also its difficulty. Therefore, it is necessary to explore more efficient knowledge transfer methods in the future, including selecting better knowledge for model architecture design.

Low-rank factorization involves decomposing the original large weight matrix into several smaller matrices and employing the low-rank matrix to approximate the original one, thereby reducing the model's weight. Its advantage is that it can induce a more sparse representation by introducing sparsity constraints, which is particularly useful for processing large-scale data sets. But the disadvantages are also obvious. The model may not capture all the details of the data. In the case of high dimension of the data itself, low-rank factorization is difficult to capture the real structure of the data, so how to design a robust algorithm for the data set itself is its difficulty. In the future, it is necessary to explore a more efficient, robust and interpretable low-rank factorization method to capture the main characteristics of the data, decrease the computational complexity and improve the accuracy of decomposition.

This paper introduces three kinds of commonly used lightweight methods and compares their experimental data in different cases, analyzes the advantages and limitations of these methods, and gives the outlook for the future.

References

1. R. Girshick, J. Donahue, T. Darrell, et al. Rich feature hierarchies for accurate object detection and semantic segmentatio//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 580-587, (2014).
2. Y. W. Guo, A. B. Yao, Y. R. Chen, et al. Dynamic network surgery for efficient DNNs//Proceedings of the 30th International Conference on Neural Information Processing. Systems. New York: ACM: 1387-1395, (2016)
3. M. A. Gordon, K. Duh, N. Andrews. Compressing bert: Studying theeffects of weight pruning on transfer learning. arXiv preprint arXiv:2002.08307, (2020)
4. E. Frantar, D. Alistarh. Sparsegpt: Massive language models can beaccurately pruned in one-shot//International Conference on Machine Learning. PMLR: 10323-10337, (2023)

5. H. Li, A. Kadav, I. Durdanovic, et al. Pruning filters for efficient ConvNets[EB/OL]. https://arxiv.org/abs/1608.08710. (2023)
6. S. Ashkboos, M. L. Croci, M. G. Nascimento, et al. Slicegpt: Compresslarge language models by deleting rows and columns. arXiv preprintarXiv:2401.15024, (2024)
7. X. Ma, G. Fang, X. Wang. Llm-pruner: On the structural pruning of large language models. Advances in neural information processing sys-tems,36: 21702-21720, (2023)
8. Y. He, G. L. Kang, X. Y. Dong, et al. Soft filter pruning for accelerating deep convolutional neural networks//Proceedings of International Joint Conference on Artificial Intelligence: 2234-2240, (2018)
9. Z. H. You, K. Yan, J. M. Ye, et al. Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks//Proceedings of Advances in Neural Information Processing Systems: 2133-2144, (2019)
10. M. B. Lin, R. R. Ji, S. J. Li, et al. Filter sketch for network pruning. arXiv preprint arXiv: 2001,08514, (2020)
11. X. Lan, X. T. Zhu, S. G. Gong. Knowledge distillation by on-thefly native ensemble//Proceedings of the 32nd International Confer-ence on Neural Information Processing Systems. Montréal，Canada: Curran Associates Inc. 7528-7538 (2018)
12. L. Yuang, Z. Wei, W. Jun. Adaptive multi-teacher multi-level knowledge distillation. Neurocomputing, 415: 106-113, (2020)
13. K. He, X. Zhang, S. Ren, J. Sun. Deep residual learning for image recognition//Proceedings of 2016 IEEE Conference on Com-puter Vision and Pattern Recognition （CVPR）. Las Vegas， USA：IEEE： 770-778 (2016)
14. H. R. Yang, M. X. Tang, W. Wen, et al. Learning low-rankdeep neural networks via singular vector orthogonality regularization and singular value sparsification//Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. Piscataway: IEEE:2899-2908, (2020)
15. M. Yin, Y. Sui, S. Y. Liao, et al. Towards efficient tensor decomposition-based DNN model compression with optimization framework//Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition.Piscataway: IEEE: 10669-10678, (2021)