No-reference image quality assessment based on automatic machine learning

Qi Qian*, and *Qingbing* Sang

School of Artificial Intelligence and Computer Science, Wuxi, Jiangsu 214122, China

Abstract. In different applications in deep learning, due to different required features, it is necessary to design specialized Neural Network structure. However, the design of the structure largely depends on the relevant subject knowledge of researchers and lots of experiments, resulting in huge waste of manpower. Therefore, in the field of Image Quality Assessment (IQA), the authors propose a method to apply Neural Architecture Search (NAS) to IQA. Mainly through the Differentiable Architecture Search algorithm, the structure of the modular Neural Network unit is searched by the stochastic gradient descent algorithm with better training performance by relaxing the operation features into a continuous space. Also, the idea of weight sharing is used to further save. The authors use the mainstream IQA database LIVE to search for Neural Network structures, and retrain and validate the searched structures in four datasets. A large number of experiments show that the model obtained by the search experiment achieves the effect of the best algorithm at this stage, and has a certain quality. The main contributions of this paper are: Transform the DARTS algorithm to adapt the regression problem, and introduce the Neural Architecture Search algorithm into the IQA field and conduct experimental verification.

1 Introdction

Image Quality Assessment has strong applications in image systems and some other fields. It is necessary to judge the performance of a camera's hardware framing and image rendering algorithms through certain evaluation rules. In recent years, researchers have proposed some evaluation methods in scenarios where deep learning is used for non-reference Image Quality Assessment.

Neural Network Structure Search Algorithm (NAS)[1] is a two-step optimization problem. Generally, a controller is used to sample to generate the structure of a model, and the trainer trains the model. Due to the model searched by the NAS algorithm have a certain performance improvement when compared with the artificially designed models, and the model search generally consumes a lot of computing power, the research in the industry has been focusing on improve the time and space optimization of NAS.

* Corresponding author: alanchien@yeah.net

2 Hypermodel design and algorithm design

To training a NAS model, if you use reinforcement learning [1], each round of training often requires thousands or even tens thousands of GPU DAY (Geforce 1080Ti). This article will introduce a time-optimized NAS algorithm.

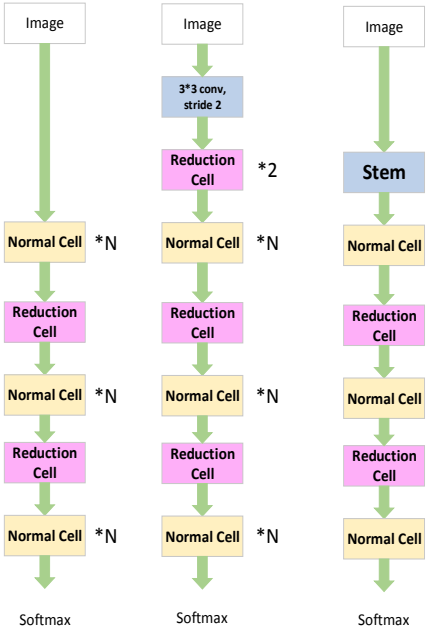

Fig 1. Architecture Design of hyper Model (a) CIFAR10 Architecture (b) ImageNet Architecture (c) LIVE Architecture in This Paper.

2.1 The main flow of hypermodel design and search algorithm

We call the neural network model used for architecture adjustment Hypermodel. According to NASnet[2], the author designed a hypermodel composed of cells, which narrowed the neural network space from the entire model structure set to the search of the internal structure of the cell. Cell can be understood as a sub-network structure. We use two different types of cells. The same type of cell will share the same neural network structure and different weights. The overall Hypermodel, stacked by cells, can be understood as specific stack of cells. Literature [2] designed two different supermodel structures in Figure 1(a) and Figure 1(b) to adapt to different sizes of data sets CIFAR-10 and ImageNet. In this paper, considering that the IQA data set has the characteristics of a small number of pictures and high training data resolution requirements, the entire model structure is designed as Figure 1(c). Next, we run the entire model search algorithm. During train search, the entire model architecture is alternately trained, as well as the hyperparameters which represent the model structure and operation. The specific principles are shown in the following sections 2.2 and 2.3. In one batch iteration, the current training batch is used as training data, and the next training batch is used as valid data to adjust the parameters of the model. See section 2.3 for specific adjustment methods and modeling methods.

The above-mentioned relaxation and approximate gradient ideas are mainly derived from a differentiable search framework [3]. The author has made some changes to its code to adapt

to the experimental scenario: in the first five epochs of training, no model adjustment is performed, because the first three epochs, The model may even not have a small convergence trend, and the evaluation effect is bound to be low. It is unreasonable to use it as a basis for structural adjustment.

The algorithm has another optimization point that needs to be described, which is the idea of parameter sharing [4]. This idea can be considered as the basis for the perfect time performance of the algorithm. Every time the cell is searched, the weight data of each layer in the cell will not be cleared, but is retained and used. While the structure is adjusted, there is no need for retraining, and use former parameter for further training directly, which greatly improves the training performance.

2.2 Transform from black box optimization to continuous optimization

Literature [3] introduces a relaxation algorithm, so that search space of the parameter becomes continuous. It can relax the search space and expandly relax the choice of specific operations of all candidate operationss. The specific formula corresponding to the softmax layer is as follows:

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \Phi} \frac{\exp\left(\alpha_o^{(x,y)}\right)}{\sum_{o' \in \Phi} \exp\left(\alpha_{o'}^{(x,y)}\right)} o(x) \tag{1}$$

Among them, $|\Phi|$ represents to the dimension of α(i,j), and the operation between a pair of nodes (i, j) is parameterized by the vector α(i,j). The task of architecture search will therefore be simplified to learning a set of continuous tensors α = {α(i,j)}, as shown in Figure 2.

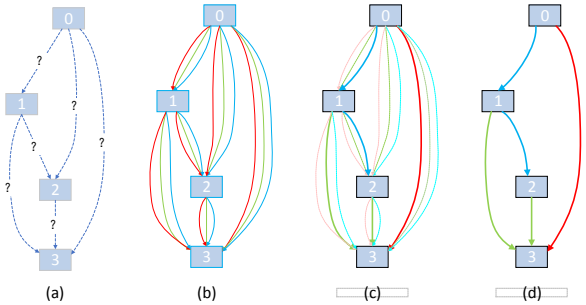

Fig. 2. Searching steps in cell (a) Initially, operations between edges are unknown. (b) The search space is continuously relaxed by placing a mixture of selected operations on each edge (c) By solving the bilevel optimization problem, the hybrid probability and network weight are jointly optimized. (d) The final architecture is derived from the learned mixed probability.

2.3 Approximate structural gradient

Literature [3] proposed an approximate gradient calculation scheme to minimize the time consumption:

$$\nabla_\alpha L_{\text{val}}(\omega^*(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \approx \nabla_\alpha L_{\text{val}}(\boldsymbol{\omega} - \xi \nabla_\omega L_{train}(\boldsymbol{\omega}, \boldsymbol{\alpha}), \boldsymbol{\alpha}) \tag{2}$$

Where ω means the current weight of the model, and ξ represents the learning rate of the optimization step. This idea can be achieved by adjusting in a single training step to approximate $*(\alpha)$, instead of training from the beginning each time until convergence.

The above method cannot ensure the convergence of the optimization algorithm. When ξ is selected appropriately, it can actually reach a suitable fixed point. Applying the chain rule to Equation 2, we get:

$$\nabla_{\alpha} \mathrm{L}_{\mathrm{val}}\left(\boldsymbol{\omega}^{'}, \boldsymbol{\alpha}\right) - \xi \nabla^{2}_{\alpha,\omega} \mathrm{L}_{\mathrm{train}}(\omega, \boldsymbol{\alpha}) \nabla_{\omega} \mathrm{L}_{\mathrm{val}}\left(\boldsymbol{\omega}^{'}, \boldsymbol{\alpha}\right) \tag{3}$$

$\boldsymbol{\omega}' = \boldsymbol{\omega} - \xi \nabla_{\omega} \mathrm{L}_{train}(\boldsymbol{\omega}, \boldsymbol{\alpha})$, represents the weight of the model in the one-step forward propagation algorithm. Since the expression contains the vector product of a matrix in the second term, the calculation is too complicated. The use of finite difference approximation can significantly reduce the computational complexity. Let a small scalar ϵ, and we have $\boldsymbol{\omega}^{\pm} = \boldsymbol{\omega} \pm \epsilon \nabla_{\omega'} \mathrm{L}_{\mathrm{val}}(\boldsymbol{\omega}', \boldsymbol{\alpha})$. Then you can get:

$$\nabla^{2}_{\alpha,\omega} \mathrm{L}_{\mathrm{train}}(\boldsymbol{\omega}, \boldsymbol{\alpha}) \nabla_{\omega'} \mathrm{Lval}(\boldsymbol{\omega}', \boldsymbol{\alpha}) \approx \frac{\nabla_{\alpha} \mathrm{L}_{\mathrm{train}}(\boldsymbol{\omega}^{+}, \boldsymbol{\alpha}) - \nabla_{\alpha} \mathrm{L}_{\mathrm{train}}(\boldsymbol{\omega}^{-}, \boldsymbol{\alpha})}{2\epsilon} \tag{4}$$

The method of evaluating finite difference is to perform two forward propagation algorithms and two back propagation algorithms on α. The current method can reduce the complexity from O (|α||ω|) to O(|α| + |ω|).

3 Experimental design and result analysis

3.1 Picture database and performance indicators

This article selects LIVE, the most common used dataset in the IQA field, to search the model. Then we use TID2013 and CSIQ dataset to retrained. The way to evaluate the quality of a quality evaluation algorithm is to compare the difference between the objective quality evaluation score of the current algorithm and the subjective quality evaluation score labelled by human beings. The indicators commonly used in the industry are Spearman's Correlation Coefficient (SROCC) and Pearson's Linear Correlation Coefficient (PLCC). The value range of SPROCC and PLCC are both -1~1. If the absolute value is closer to 1, the effect will be better.

3.2 Experimental design

The experimental hardware environment uses intel Xeon E5-2600V processor with eight cores and sixteen threads; the graphics card uses NVIDIA GeForce RTX 2080Ti. The software environment is implemented using pytorch 0.3.1 based on ubuntu16.04. For the processing of the image database, consider the following:

First of all, the data set in the IQA field often has the problem of a small amount of data. At the same time, due to the limitation of computing power, the size of the picture in the data set is often not too large.

Secondly, the picture characteristics of the data set in the IQA field are actually its distortion information. If the picture is compressed, it will increase the distortion information and also lose some of the original distortion information, which is undesirable. While using image segmentation, if the segmentation is too small, it will result in learning too few distortion features. The author cuts images to different sizes for experimentation, and appropriately reduces the batch size and the number of model layers to meet the limit of video memory capacity. This experiment will crop the image To the size of 112*112. In this paper, the structure search and evaluation training of the experimental classification model are two parts of the experiments. The structure of the cell can be obtained by the structure search. It is found through experiments that the evaluation training is retrained on the empty model

composed of the searched cells. For details, see 2.3. After using the LIVE dataset for model structure search, if you use training sets with other different distortion types the Live dataset do not contain, The retraining of the model, the performance of the obtained model structure in other data sets cannot reach the performance of the recent mainstream algorithms, and it can only be close to it. For this reason, we choose to filter the dataset used in the training part of the evaluation and only keep the distortion types covered during the architecture search.

Finally, for regression tasks, we need to control the original output neuron to one, and simply control the output category to one. The calculation method of the loss function of the classification task and the regression task is different. In this experiment, we use MSE to calculate the current loss value of the entire model.

3.3 Experimental results and algorithm versatility

For experiments on the live_release2 data set, the Normal cell and Reduction cell we searched out are as follows:

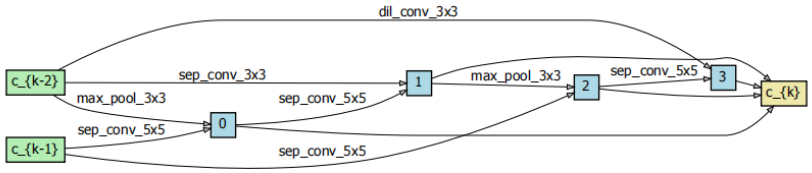

Fig. 3. Search result of reduction cell.

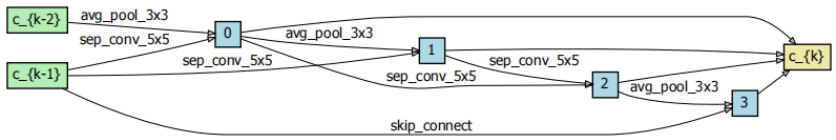

Fig. 4. search result of normal cell.

After retraining and testing, we query the classic non-reference Image Quality Assessment methods, as well as the evaluation effects of some new methods in recent years. Table 2 shows the results of our model and BLIINDS-II[5], DLIQA[6], SRNSS[7] , DipIQ[8], CORNIA[9], BLISS[10], BRISQUE[11], LINIQE[12], DIIVINE[13], BIQI[14] and other algorithms after experiments in different data sets SROCC and PLCC results. The experimental results show that in the dimensions of PLCC and SROCC, our neural network structure algorithm can efficiently and high-quality complete the learning task of ordinary distortion without reference image.

Table 1. Comparison of different Algorithm's experimental results of different dataset.

Algorithm	LIVE		CSIQ		TID2013	
	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC
BLIINDS-II	0.940	0.942	0.943	0.943	0.943	0.943
DLIQA	0.956	0.953	0.941	0.941	0.941	0.941
SRNSS	0.964	0.962	0.94	0.94	0.94	0.94
dipIQ	0.942	0.935	0.929	0.955	0.921	0.939
CORNIA	0.845	0.867	0.921	0.909	0.888	0.91
BLISS	0.927	0.927	0.92	0.942	0.882	0.9
BRISQUE	0.916	0.916	0.915	0.945	0.894	0.907
LINIQE	0.940	0.938	0.891	0.92	0.873	0.897
DIIVINE	0.947	0.943	0.857	0.882	0.819	0.845
BIQI	0.950	0.944	0.845	0.845	0.845	0.835
Ours	0.959	0.972	0.925	0.945	0.946	0.959

4 Conclusion

This paper proposes a neural network structure search algorithm for non-reference Image Quality Assessment. It relaxes the operation of the cell's internal convolutional layer to a continuous space, and then uses the SGD algorithm to search for the structure of each cell in the entire model. This algorithmic idea keeps the training time within two GPU DAYs while ensuring that an excellent network structure can be learned. Experiments have proved that the model composed of search results of different Cells performs well in the mainstream quality evaluation data set. Our follow-up research contains two directions. One is to focus on other search ideas of the search algorithm which is based on based on stochastic gradient descent, to find some method which is different from differentiable and parameter sharing; the other is to focus on the optimization of this method. The method still have large places to improvement in terms of accuracy, stability, and computing power consumption.

References

1. B. Zoph and Q.V. Le. Neural architecture search with reinforcement learning. ICLR, (2017).
2. B. Zoph, V. Vasudevan, J. Shlens, and Q.V. Le. Learning transferable architectures for scalable image recognition. CVPR, (2018).
3. H.X. Liu, K. Simonyan, Y.M. Yang.Differentiable neural network architecture search.ICLR (2019)
4. H. Pham, Melody Y Guan, Barret Zoph, Quoc V Le, and Jeff Dean. Efficient neural architecture search via parameter sharing ICML, (2018) b.
5. A. K. Moorthy and A. C. Bovik, Blind image quality assessment: From natural scene statistics to perceptual quality. IEEE Trans Image Process,vol. 20, no. 12, pp. 3350–3364, Dec. (2011).
6. W. Hou,X. Gao,D. Tao,et al. Blind image quality assessment via deep learning. IEEE Trans Neural Netw Learn Syst, (2015), 26(6):1275-1286.
7. L. He,D. Tao,X. Li,et al. Sparse representation for blind image quality assessment. Proceedings of 2012 IEEE CVPR, (2012):1146-1153.
8. K. Ma,W. Liu,T. Liu,et al. dipIQ:Blind image quality assessment by learning- to- rank discriminable image pairs[J]. IEEE Trans Image Process,(2017), **26**(8):3951-3964
9. P. Ye,J. Kumar,L. Kang,et al. Unsupervised feature learning framework for no-reference image quality assessment[C]. IEEE CVPR (2012): 1098-1105.
10. P. Ye,J. Kumar,D. Doermann. Beyond human opinion scores: Blind image quality assessment based on synthetic scores[C].IEEE CVPR,(2014): 4241-4248.
11. A. Mittal,A.K. Moorthy,A.C. Bovik. No-reference image quality assessment in the spatial domain[J] IEEE Trans Image Process,(2012),**21**(12):4695-4708.
12. L. Zhang, L. Zhang,A.C. Bovik. A feature-enriched completely blind image quality evaluator[J]. IEEE Trans Image Process, (2015), **24**(8):2579-2591.
13. A.K. Moorthy, A.C. Bovik. Blind image quality assessment: From natural scene statistics to perceptual quality [J] IEEE Trans Image Process, (2011), **20**(12): 3350-3364.
14. A.K. Moorthy, A.C. Bovik. A two-step framework for constructing blind image quality indices [J]. IEEE Signal Process Lett,(2010), **17**(5):513-516