Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 5, No. 3, 2023

A Zero-Cost Darts Base on Multi-Step Optimization

Minghui Zhang *

Department of software, Chengdu University of Information Technology, Chengdu, China
*** Corresponding author Email:** 1204616897@qq.com

Abstract: DARTS has achieved great result in Image classification field, the accuracy predictor and computation costs are the key of DNAS algorithm. Searching for a high-performance architecture always costs Large amount of computation. With a gradient-based bi-level optimization, DARTS using one-step optimization which makes the process available within a few GPU day, because of the one-step optimization , there exists a great gap between the architectures in search and evaluation. In this paper, we propose a zero-cost DARTS method which using multi-step optimization to address the above issues. To further reduce the computational requirements, we use the zen-score to estimate architectures in evaluation stage. Experiments on CIFAR-10 and our private data sets show that our algorithm play a certain role in solving the above problems.

Keywords: Zero-Cost Darts; Multi-Step; Zen-NAS.

1. Introduction

With the development of the Deep Learning over the past few years, the focus of machine learning has shifted from feature design to architecture design. The quality of architecture has become a crucial part of machine learning. Generally, a high-quality architecture can not only obtain high-precision prediction results, but also save a lot of computing resources[1]. To address different computer vision problems, A large number of architecture has been designed manually[2][3][4]. Designing a man-made architecture requires lots of expert knowledge and take much time. Neural architecture search (NAS) is an automated and efficient algorithm which requires no specialized knowledge. The core idea of NAS is to search the large architecture space for an architecture that meets the needs of the target task[1], and existing NAS methods can be divided to three categories, including one-shot methods, reinforcement-learning (RL) and evolution (EA) methods[5].

RL and EA methods requires a lot of computation overheads, to improve the search cost, weight-sharing-base methods are proposed which named one-shot methods. The key of the one-shot methods is training a supernet which covering all the candidate network[6]. Recently, DARTS[7] was proposed recently, which is a differentiable NAS algorithm based on one-shot method, it narrowed the NAS search time to a few GPU days, and greatly saves computing resources. In some previous work, it was mentioned that DARTS has a performance collapse in the architecture search process[8][9], because of due to its one-step optimization and the transitional use of skip. In our engineering projects, DARTS 'search time was too long and not very efficient.

In this work, we propose a novel multi-step optimized DARTS, which use the zero-cost proxy, dubbed Zen-Score. Zero-Score only required a few forward propagation, which using random Gaussian inputs on randomly initialized network[10]. In our experiment, the improved algorithm limits the abuse of skip, thus reducing the performance collapse, and the accuracy is comparable to SOTA.

2. Related Works

Neural architecture search[1] is a great way to automate the search for neural networks. Although [3][4][6][17][20] use weight sharing to reduce the computational cost, we still need to spend a large number of computational cost on a deep learning task. DARTS was the first algorithm to make the discrete search space continuous, this differentiable search space can be optimized using gradient descent, ultimately reducing the search time to a few GPU days. DARTS uses a one-step optimization strategy, which has two drawbacks. DARTS uses a one-step optimization strategy, which has two drawbacks. The first is that the skip operation can be abused during the search process and lead to accuracy collapse[8][9][22]. The second is that one-step optimization is easy to overfit and fall into local optimal solutions.

In the NAS algorithm, the largest time cost comes from the training of the neural network, and during the search process, the number of candidate networks to be trained is too large, even with DARTS and its derivatives, it is very computationally expensive. The Zen score metric enables you to test the performance of a neural network without using samples for supervised learning, so using Zen scores allows you to avoid spending computational resources and time on training candidate network architectures. The most common metric is the Neural Tangent Kernel[11], and its derivatives NCN[12], F Norm[13].

Zen-NAS[14] uses the LGA metric to evaluate the performance of candidate architectures in NAS, which is a derived metric of NTK and uses both label information and NTK metric. Our approach is to use two hyperparameters to balance NTK and label information.

3. Method

3.1. Preliminaries of DARTS

DARTS [9] uses a stack of normal cells and reduced cells to form the final network. There exist n nodes in the Cell, and each node represents a potential expression. A directed link edge $e_{i,j}$ between every two nodes represents an information transfer from node i to node j. In other words, a Cell is represented as a DAG graph. Each edge (i,j) contains some candidate operations (e.g., convolution, pooling).

DARTS applies continuous relaxation to synthesize the outputs of different operations. Let x_i be an input of node i and let $o_{i,j}$ be an operation of edge (i, j). For any intermediate

node j, its input is the weighted sum of the outputs of all predecessor nodes:

$$x_j = \sum_{i<j} o_{i,j}(x_i) \quad (1)$$

Let O be a set of all candidate operations, each candidate operation o is a function whose role is to feature map the input x_i, $O_{i,j}$ is the set of candidate operations between edge (i, j):

$$O_{i,j} = \{o^1_{i,j}, o^2_{i,j}, \ldots, o^n_{i,j}\} \quad (2)$$

Let $\alpha_{o_{i,j}}$ denote the architectural weights vector. To optimize with gradient descent, DARTS uses softmax over all candidate operations to make the search space continuous:

$$\bar{o}_{i,j}(x) = \sum_{o \in O} \frac{exp(\alpha_{o_{i,j}})}{\sum_{o' \in O} exp(\alpha_{o'_{i,j}})} o(x) \quad (3)$$

DARTS is a bilevel optimization problem, α is the upper-level variable and w is the weight of the neural network as the lower-level variable:

$$\min_{\alpha} L_{val}(w^*(\alpha), \alpha) \quad (4)$$

$$\text{s.t. } w^*(\alpha) = \arg\min_{w} L_{\text{train}}(w, \alpha) \quad (5)$$

DARTS applies one-step optimization to update the architecture parameter a. Specifically, the lower-level variable w is-first updated using one-step gradient descent based on $\nabla_w L_{train}(w, \alpha)$. Then the w parameter is kept constant and the upper-level variable a is updated using one-step gradient descent based on $\nabla_\alpha L_{val}(w - \varepsilon \nabla_w L_{train}(w, \alpha), \alpha)$. Where α is learning rate.

Many previous works [14][15][18][21] have pointed out that the double optimization of DARTS causes the final model accuracy to collapse, and this is also true in our project. In the next section we describe the causes and our improvements.

3.2. Multi-step Optimization

Our target task is to identify the age of a person, and DARTS does not perform well in our task. Our target task is age recognition of people, and DARTS does not perform well in our task. We analyze the main reasons, because the single-step optimization of DARTS is easy to fall into local optima, and the data of our task are all faces, which contain too many identical features, DARTS is not efficient to find a high accuracy model.

We have made adjustments to the single-step optimization strategy of DARTS. We have made adjustments to the single-step optimization strategy of DARTS. First let DARTS optimize the network parameters w using Equation (6) n times, where n is a hyperparameter. Then a one-step optimization strategy is used to update the architecture parameter α as in Equation (7). Using this strategy, the architecture parameter α of DARTS are a slow learning process, while the shared network parameters w of the DARTS is a fast learning process. The faster convergence of w ensures better learning of image features during the search process, thus improving the accuracy of the final architecture searched by DARTS. We refer to this approach as multi-DARTS.

$$w' = w - \varepsilon \nabla_w L_{train}(w, \alpha), \alpha) \quad (6)$$

$$\alpha = \alpha - \epsilon \nabla_\alpha L_{val}(w', \alpha) \quad (7)$$

However, we found in our experiments that the time cost of DARTS with this strategy is greatly increased. Suppose that in one-step optimized DARTS, the time cost of training w is t1, while the time cost of training a architecture parameter α is t2, then the total time cost is t1+t2. However, in our method, w is trained using multi-step optimization, the total time cost is n*t1+t2. In other words, the time cost of w training is improved to n times.

3.3. Zen-score

Zen-score allows us to train neural networks without using labeled data. These methods use information inside the architecture to improve the accuracy of the architecture. Just like the Neural Tangent Kernel (NTK), it uses the number of linear regions inside the network as a metric. The higher the number of linear regions, the better the accuracy of the network[11].

Specifically, the network weights w are first randomly initialized. Input x is randomly sampled from a Gaussian distribution. Then the NTK of this network is calculated based on the input x, and the loss function is designed based on the NTK and the gradient descent algorithm is used to optimize the network weights w.

Let f be f mapping function, θ a trainable set of weight parameters, and θ_0 a randomly initialized weight. NTK can be expressed as equation (9).

$$f_\theta(x) = f_{\theta_0}(x) + (\theta - \theta_0)\nabla_\theta f_{\theta_0}(x) \quad (8)$$

$$\text{NTK}(x, x') = < \nabla_\theta f_{\theta_0}(x), \nabla_\theta f_{\theta_0}(x')^T > \quad (9)$$

$\nabla_\theta f_{\theta_0}(x)$ is a Jacobian matrix. It can be seen that the essence of NTK is the dot product of two gradient vectors, which represents the condensed expression of the gradient value and gradient correlation [14].

In our project, NTK did not perform well. Because NTK and network accuracy are positively correlated on the premise that NTK cannot change too much during training, and the learning rate should be small enough [16]. In our target task, the variation of NTK is large, so we make improvements to the traditional NTK. The output of the BN layer can similarly be used as zen-score. The output of BN layer does not need too many assumptions to express the classification ability of the network [16].

We find through experiments that BN as Zen-score is not as effective as NTK in our target task, and when they are used together, the effect is higher than that of using one alone. Let β_1, β_2 be two hyperparameters, and the final Zen-score used is as in (10).

$$\text{Zen}(F) = \beta_1 \log(\text{NTK}) + \beta_2 BN \quad (10)$$

To summarize, our algorithm first trains the searched subarchitectures using Zen-score and updates the shared weights w. After m epochs (m is a hyperparameter), the labeled data will be used to train the searched subarchitectures and the shared weights w will be updated. With this method, the time overhead of training the searched candidate network is negligible during the first m epochs.

4. Experiments

4.1. Datasets

In this section, we conduct several experiments to verify the effectiveness of the proposed method. The datasets we have used include CIFAR-10 and the datasets in our project.

4.2. Search Space

The parameter Settings are described next. The search space is the same as DARTS[9], using normal cells and reduction cells, and the alternative operations for each connected edge are: 3 × 3 and 5 × 5 separable convolutions, 3 × 3 and 5 × 5, dilated separable convolutions, 3 × 3 max pooling, 3 × 3 average pooling, identity, and zero. Each cell consists of 7 nodes, and the output node is concatenated by all intermediate nodes.

4.3. Searching on CIFAR-10

We tested our algorithm on CIFAR-10 and achieved an error rate of 2.56%, a 0.44% improvement over DARTS, and a time cost of 2GPU days, 0.25GPU days longer than DARTS.

4.4. Performance on the Project

Our target task is to judge the age of a person based on an image, divided into four classes, child, young adult, old adult, and non-human. The original DARTS is for binary classification, so we need to make some improvements in the output layer. Our algorithm uses 4 fully connected layers for the output layer.

In our project, DARTS has a mAP of only 78%, while using our algorithm it achieves 85%. Although the time cost of searching increased by 5-6 GPU hours, it greatly improved the accuracy of the final model.

5. Conclusion

In this paper, we use a multi-step optimization strategy to reduce the overfitting of DARTS, and then use Zen-score to reduce the time overhead of Multi-DARTS. Finally, the DARTS algorithm has been greatly improved in our project.

References

[1] Zoph B, Le Q V. Neural architecture search with reinforcement learning[J]. arXiv preprint arXiv:1611.01578, 2016.

[2] He K, Zhang X , Ren S ,et al.Deep Residual Learning for Image Recognition[J].IEEE, 2016.DOI:10.1109/CVPR.2016.90.

[3] Simonyan K, Zisserman A .Very Deep Convolutional Networks for Large-Scale Image Recognition[J].Computer Science, 2014.DOI:10.48550/arXiv.1409.1556.

[4] Lecun Y, Bottou L .Gradient-based learning applied to document recognition[J].Proceedings of the IEEE, 1998, 86(11):2278-2324.DOI:10.1109/5.726791.

[5] Pham H, Guan M, Zoph B, et al. Efficient neural architecture search via parameters sharing[C]//International conference on machine learning. PMLR, 2018: 4095-4104.

[6] Brock A, Lim T, Ritchie J M, et al. Smash: one-shot model architecture search through hypernetworks[J]. arXiv preprint arXiv:1708.05344, 2017.

[7] Liu H, Simonyan K, Yang Y. Darts: Differentiable architecture search[J]. arXiv preprint arXiv:1806.09055, 2018.

[8] Chu X, Zhou T , Zhang B ,et al.Fair DARTS: Eliminating Unfair Advantages in Differentiable Architecture Search [C]// European Conference on Computer Vision.Springer, Cham, 2020.DOI:10.1007/978-3-030-58555-6_28.

[9] Hong W, Li G , Zhang W ,et al.DropNAS: Grouped Operation Dropout for Differentiable Architecture Search[C]//Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence {IJCAI-PRICAI-20.2020. DOI:10. 24963/ ijcai.2020/318.

[10] Abdelfattah M S, Mehrotra A , Dudziak U ,et al.Zero-Cost Proxies for Lightweight NAS[J]. 2021.DOI:10. 48550/ arXiv. 2101.08134.

[11] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: convergence and generalization in neural networks. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, pages8580–8589, 2018.

[12] Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on imagenet in four gpu hours: A theoretically inspired perspective. In International Conference on Learning Representations, 2020.

[13] Jingjing Xu, Liang Zhao, Junyang Lin, Rundong Gao, Xu Sun, and Hongxia Yang. Knas: Green neural architecture search. In International Conference on Machine Learning, pages 11613–11625. PMLR, 2021.

[14] Lin M, Wang P , Sun Z ,et al.Zen-NAS: A Zero-Shot NAS for High-Performance Deep Image Recognition[J]. 2021.DOI: 10. 13140/ RG.2.2.33579.98084.

[15] Wang H, Yang R , Huang D ,et al.iDARTS: Improving DARTS by Node Normalization and Decorrelation Discretization [J].IEEE transactions on neural networks and learning systems, 2023.

[16] Mok J , Na B , Kim J H ,et al.Demystifying the Neural Tangent Kernel from a Practical Perspective: Can it be trusted for Neural Architecture Search without training?[J]. 2022.DOI: 10. 48550/arXiv.2203.14577.

[17] Xie S, Zheng H, Liu C, et al. SNAS: stochastic neural architecture search[J]. arXiv preprint arXiv:1812.09926, 2018.

[18] Wang Y, Yang Y, Chen Y, et al. Textnas: A neural architecture search space tailored for text representation[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(05): 9242-9249.

[19] Heuillet A, Tabia H , Arioui H ,et al.D-DARTS: Distributed Differentiable Architecture Search[J]. 2021.DOI:10. 48550/ arXiv. 2108.09306.

[20] Liang H, Zhang S , Sun J ,et al.DARTS+: Improved Differentiable Architecture Search with Early Stopping[J]. 2019. DOI:10.48550/arXiv.1909.06035.

[21] Xu Y, Xie L , Zhang X ,et al.PC-DARTS: Partial Channel Connections for Memory-Efficient Architecture Search[C]// International Conference on Learning Representations.2020.

[22] Chen J T Q. Progressive DARTS: Bridging the Optimization Gap for NAS in the Wild[J].International Journal of Computer Vision, 2021, 129(3).