Differentiable algorithm for searching the model architecture with control of its complexity[*]

K. D. Yakovlev[1][0000−1111−2222−3333], O. S. Grebenkova[1][1111−2222−3333−4444],
and O. Y. Bakhteev[1][2222−−3333−4444−5555]

MIPT, Russia {iakovlev.kd, grebenkova.os, bakhteev}@phystech.edu
http://www.springer.com/gp/computer-science/lncs

Abstract. The paper investigates the problem of deep learning model optimization. The authors propose a method for finding the architecture of a model that allows you to control its complexity with a small computational cost. The complexity of the model refers to the minimum length of the description, the minimum amount of information required to transmit information about the model and the dataset. The method is based on a differentiable architecture search algorithm (DARTS). It is proposed to use the hypernet as a relaxation function. A hypernet is a model that generates the parameters of an optimal model. The proposed method allows you to control the complexity of the model in the process of searching for an architecture. To assess the quality of the proposed algorithm, experiments are conducted on a sample of MNIST.

Keywords: differentiable architecture search · deep learning · hypernetwork · neural networks · model complexity control.

1 Introduction

In this paper, we consider the problem of searching the architecture of a deep learning model with the control of its complexity. A model is a superposition of functions that solves a classification or regression problem [1]. The search for the model architecture is understood as the search for optimal structural parameters. Relaxation refers to the translation of a set of acceptable structural parameters from discrete to continuous. The basic algorithm is differentiable architecture search algorithm [9]. It solves the problem of searching the model architecture by translating the search space of structural parameters from a discrete to a continuous representation. It is proposed to use gradient optimization methods. They use less computational resources than methods that operate on a discrete set of structural parameters. This algorithm works with both convolutional and recurrent neural networks.

In [2] it is stated that DARTS is unstable. This is due to the fact that the parameters of the model architecture converge in a narrow region. Therefore, small perturbations of the architecture lead to a significant decrease in quality.

[*] Supported by organization x.

In [4], it was noticed that the softmax function has a significant drawback. Some components of the architecture parameter vector increase faster than others. This leads to the fact that the architecture is oversimplified and, as a result, the quality deteriorates [4]. In this regard, it is proposed to use the sigmoid relaxation function and abandon the normalization. Thus, all the architecture parameters change at the same rate. In this paper, we propose to use the hypernet [5] as a relaxation function. The approach is to use a small network to generate the architecture parameters of the desired network. The hypernet is also used to control the complexity.

Alternative DARTS approaches to solving the problem of searching the model architecture are proposed. In [3], the problem of distribution learning is formulated. The architecture parameters are subject to the Dirichlet distribution, since they are defined on the probabilistic simplex. Thus, the task of searching the architecture is reduced to finding the parameters of the Dirichlet distribution.

The paper [6] builds a method for searching for a neural architecture with a limited resource (RC-DARTS). Restrictions are added to the basic DARTS algorithm, such as the number of model parameters. To solve the problem of conditional optimization, an iterative projection algorithm is introduced, which consists in the fact that after a certain number of iterations of gradient descent, the projection occurs on the set specified by the constraints.

The computational experiment is performed on a dataset MNIST [8].

2 Problem statement

2.1 Differentiable cell architecture search algorithm

Let's set the task of finding the cell architecture. A cell is a N of numbered nodes represented as a directed acyclic graph. Each edge of (i, j) is assigned a mapping (operation) $o^{(i,j)} \in \mathcal{O}$, where $\mathcal{O}$ is a family of maps (a set of operations). Values in each of the intermediate nodes $x^{(j)}$ are defined through the values in the nodes with the lower number:

$$x^{(j)} = \sum_{i<j} o^{(i,j)}(x^{(i)}). \tag{1}$$

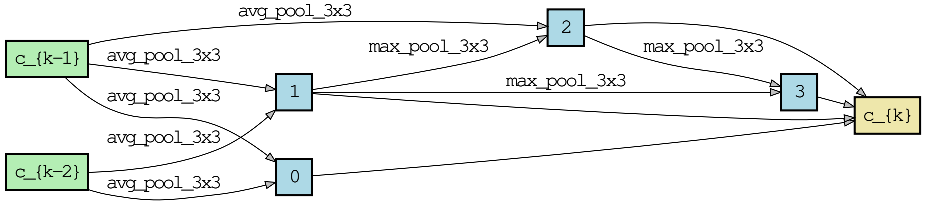

Fig. 1. Example of the found architecture

The task of architecture search is to choose the mapping between the cell nodes. In order to reduce the discrete optimization problem to the continuous optimization problem, we introduce a mixed operation for each edge (i, j):

$$\hat{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x), \tag{2}$$

where $\alpha_o^{(i,j)}$ denotes the corresponding weight of the operation o on the edge of (i, j). Thus, each edge (i, j) is assigned a vector $\alpha^{(i,j)}$ of dimension $|\mathcal{O}|$. Let $\boldsymbol{\alpha} = [\alpha^{(i,j)}]$. We formulate a two-level optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \mathcal{L}_{\text{val}}(\mathbf{w}^*, \boldsymbol{\alpha}), \\ \text{s.t.} \quad \boldsymbol{w}^* = \arg &\min_{\mathbf{w}} \mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\alpha}) \end{aligned} \tag{3}$$

Here $\mathcal{L}_{\text{val}}$ and $\mathcal{L}_{\text{train}}$ are the loss functions of the model on validation and on training, respectively.

2.2 Linear hypernet

Let Λ be a set of parameters that control the complexity of the model. A hypernet is a mapping:

$$\mathbf{G} : \Lambda \times U \to A, \tag{4}$$

where A is the architecture parameter space $\{\alpha_o^{(i,j)}\}$, and U is the set of hypernet parameters. In this paper, a linear hypernet is used to obtain the weights of operations:

$$\boldsymbol{\alpha} = \mathbf{G}_{\text{linear}}(\lambda) = \lambda \mathbf{b}_1 + \mathbf{b}_2, \quad [\mathbf{b}_1, \mathbf{b}_2]^\top \in U \tag{5}$$

where $\mathbf{b}_1,\ \mathbf{b}_2$ are configured according to the optimization problem 3.

3 Computational experiment

References

1. Bakhteev, O.Y., Strijov, V.V.: Deep learning model selection of suboptimal complexity. Autom. Remote. Control **79**(8), 1474–1488 (2018)
2. Chen, X., Hsieh, C.J.: Stabilizing differentiable architecture search via perturbation-based regularization. CoRR **abs/2002.05283** (2020)
3. Chen, X., Wang, R., Cheng, M., Tang, X., Hsieh, C.J.: Drnas: Dirichlet neural architecture search. CoRR **abs/2006.10355** (2020)
4. Chu, X., Zhou, T., 0046, B.Z., Li, J.: Fair darts: Eliminating unfair advantages in differentiable architecture search. CoRR **abs/1911.12126** (2019), http://arxiv.org/abs/1911.12126
5. Ha, D., Dai, A.M., Le, Q.V.: Hypernetworks. CoRR **abs/1609.09106** (2016), http://arxiv.org/abs/1609.09106
6. Jin, X., Wang, J., Slocum, J., 0001, M.H.Y., Dai, S., Yan, S., Feng, J.: Rc-darts: Resource constrained differentiable architecture search. CoRR **abs/1912.12814** (2019), http://arxiv.org/abs/1912.12814
7. Krizhevsky, A., Nair, V., Hinton, G.: Cifar-10 (canadian institute for advanced research) http://www.cs.toronto.edu/ kriz/cifar.html
8. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010), http://yann.lecun.com/exdb/mnist/
9. Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search. CoRR **abs/1806.09055** (2018), http://arxiv.org/abs/1806.09055
10. Yakovlev, K.: https://github.com/Intelligent-Systems-Phystech/2021-Project85