Selection of concordant models with complexity control

K. D. Yakovlev[1],, and O. Y. Bakhteev[1,2]

[1] MIPT, Russia
[2] Dorodnicyn Computing Center RAS, Russia
{iakovlev.kd, bakhteev}@phystech.edu

Abstract. The paper investigates the problem of choosing the architecture of a deep learning model. A model is understood as a superposition of functions differentiable by parameters. The problem of choosing a model of a special type in which the structural parameters of the model are a function of the input object to be analyzed. To increase the interpretability of the final architecture, it also depends on the parameter of the required complexity, which allows us to find a compromise between the complexity of the model and its predictive characteristics during operation. The complexity of the model is understood as the minimum length of the description, the minimum amount of information required to transmit information about the model and the dataset.

Keywords: differential architecture search · deep learning · hypernetwork · neural networks · model complexity control · mixture of experts

1 Introduction

In this paper, the problem of searching for the architecture of a deep learning model with the control of its complexity is considered. The model is understood as a superposition of functions that solves the problem of classification or regression [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 permissible structural parameters from discrete to continuous. The differentiable algorithm for searching for the DARTS [9] architecture is used as the basic algorithm. It solves the problem of searching for the architecture of the model 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 operating on a discrete set of structural parameters. This algorithm works with both convolutional and recurrent neural networks.

In this paper, it is proposed 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 complexity.

There are several approaches that search for architecture with complexity control. In the work [6], a method for searching for a neural architecture with

a limited resource (RC-DARTS) is constructed. 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, projection onto a set set by constraints occurs.

In [13], an approach is proposed that allows you to control the complexity of the resulting architecture. The relaxation described in [9] is subject to restrictions. Exactly one mapping is involved in the mix. To control the complexity, it was proposed to use an L2 regularizer for structural parameters. The optimization algorithm is constructed in such a way that a discrete architecture is obtained at each iteration. In addition, an acceleration in architecture optimization is achieved due to the fact that only the selected mapping for each edge is used during the backward pass.

There are approaches that use experts to search for architecture. In [11], the search space is the same as in [4]. The importance of the mappings in the mix is regulated by experts. During the optimization process, the number of experts involved in the mix gradually decreases. This approach has a number of advantages. Firstly, the resulting discrete architecture will not be much mixed in quality relative to the optimal continuous one. Secondly, the effect of retraining decreases. Thirdly, the optimization process is accelerated due to the fact that the number of experts is reduced.

In this paper, the architecture is subject to the requirement of consistency. There are approaches that formalize this concept. In [14], the problem of predicting a complex target variable is investigated. A matching function is introduced, which is responsible for the proximity of latent representations of the independent and target variable. Then the prediction problem is solved in the hidden space.

2 Problem statement

Given a dataset $\mathfrak{D} = (\mathbf{X}, \mathbf{y})$. Each object $\mathbf{x} \in \mathbf{X}$ is assigned a target variable $y \in \mathbf{y}$.

In the approach described in DARTS, the architecture is considered, which is a sequence of *cells*. All cells have the same structure, but different parameter values. A cell is an oriented acyclic graph. Formally, there is a set of vertices $V = \{1, \ldots, N\}$ and a set of edges $E = \{(i, j)V \times V : i < j\}$, where N is some natural number. Each edge of (i, j) corresponds to a nonlinear mapping $\mathbf{g}^{(i,j)}$, which is a component of the vector $\vec{\mathbf{g}}^{(i,j)}$.

The values that are calculated in the jth node are calculated through values with smaller numbers:

$$\mathbf{x}^{(j)} = \sum_{(i,j)\in E} \mathbf{g}^{(i,j)}(\mathbf{x}^{(i)}),$$

where $\mathbf{x}^{(0)} = \mathbf{x}$.

The task of architecture search is to find nonlinear mappings $\mathbf{g}^{(i,j)}$ for each edge of the cell $(i, j) \in E$. This discrete optimization problem is reduced to a

continuous one. To do this, the concept of *mixed operation* is introduced:

$$\hat{\mathbf{g}}^{(i,j)} = \langle \mathbf{softmax}(\boldsymbol{\alpha}^{(i,j)}), \vec{\mathbf{g}}^{(i,j)}(\mathbf{x}^{(i)}) \rangle,$$

where $\boldsymbol{\alpha}^{(i,j)}$ are *structural parameters* that determine the importance of each mapping $\mathbf{g}^{(i,j)}$. Thus, each edge $(i,j) \in E$ corresponds to the vector $\boldsymbol{\alpha}^{(i,j)} \in \mathbb{R}^{\dim \vec{\mathbf{g}}^{(i,j)}}$. Let $\boldsymbol{\alpha}$ be the concatenation of vectors $\boldsymbol{\alpha}^{(i,j)}$ over all edges $(i,j) \in E$.

Let the sample $\mathfrak{D}$ consist of a disjunct union of training and validation samples: $\mathfrak{D} = \mathfrak{D}_{\text{train}} \sqcup \mathfrak{D}_{\text{valid}}$. The structural parameters of α are found from the following two-level optimization problem:

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \mathcal{L}_{\text{valid}}(\mathbf{w}^*, \boldsymbol{\alpha}), \tag{1}$$

$$s.t., \quad \mathbf{w}^* = \arg\min_{\mathbf{w}} \mathcal{L}_{\text{train}}(\mathbf{w}, \boldsymbol{\alpha}). \tag{2}$$

Here $\mathcal{L}_{\text{train}}$ and $\mathcal{L}_{\text{valid}}$ are the cross-entropy loss function on samples $\mathfrak{D}_{\text{train}}$ and $\mathfrak{D}_{\text{valid}}$ respectively, $\mathbf{w} \in \mathbb{R}^n$ is the vector of model parameters.

2.1 Complexity control

Our first modification of DARTS is that instead of using **softmax** operation in in mixed operation, we employ Gumbel-Softmax distribution [10]. Second, we split $\vec{\mathbf{g}}^{(i,j)}$ on dishoint sets. Each set consists of mappings assigned to a particular expert, but having different complexity. Formally, let be $\boldsymbol{\gamma}^{(i,j)}$ belonging to a $|\vec{\mathbf{g}}^{(i,j)}| - 1$ dimentional simplex. The mixed operation takes the following form:

$$\hat{\mathbf{g}}^{(i,j)}(\mathbf{x}^{(i)}) = \langle \boldsymbol{\gamma}^{(i,j)}, \vec{\mathbf{g}}^{(i,j)}(\mathbf{x}^{(i)}) \rangle. \tag{3}$$

Now split $\vec{\mathbf{g}}^{(i,j)}$ into N disjoint sets of experts:

$$\vec{\mathbf{g}}^{(i,j)} = [\vec{\mathbf{g}}_k^{(i,j)}]_{k=1}^N. \tag{4}$$

For simplicity, Define a vector $\boldsymbol{\gamma}_{\text{exp}}^{(i,j)}$ from $N-1$ dimensional simplex. Also define $\boldsymbol{\gamma}_{\text{comp}}^{(i,j),k}$ from $\dim \vec{\mathbf{g}}_k^{(i,j)} - 1$ dimensional simplex for each k. Rewite (3) in new notation:

$$\hat{\mathbf{g}}^{(i,j)}(\mathbf{x}^{(i)}) = \sum_{k=1}^{N} \gamma_{\text{exp},k}^{(i,j)} \langle \boldsymbol{\gamma}_{\text{comp}}^{(i,j),k}, \vec{\mathbf{g}}_k^{(i,j)}(\mathbf{x}^{(i)}) \rangle. \tag{5}$$

In this paper we use a hypernetwork to control complexity of the architecture. Let Λ a set of the regularization parameters λ values. A hypernetwork is a parametric mapping from the set λ to the set of model's structural parameters. We also assume that the hypernetwork depends on object $\mathbf{x}$:

$$\mathbf{h}^{(i,j)} = \mathbf{h}^{(i,j)}(\mathbf{x}, \mathbf{a}^{(i,j)}, \lambda), \quad \lambda \in \lambda, \ \mathbf{x} \in \mathbf{X}.$$

Define a probalistic model:

$$p(\boldsymbol{\gamma}_{\text{exp}}^{(i,j)}, [\boldsymbol{\gamma}_{\text{comp}}^{(i,j),k}], \lambda, \mathbf{a}_{\text{exp}}^{(i,j)}, [\mathbf{a}_k^{(i,j)}]|\mathbf{x}) =$$
$$p(\boldsymbol{\gamma}_{\text{exp}}^{(i,j)}|\mathbf{x}, \mathbf{a}_{\text{exp}}^{(i,j)}, \lambda) \prod_{k=1}^{N} p(\boldsymbol{\gamma}_{\text{comp}}^{(i,j),k}|\mathbf{x}, \mathbf{a}_k^{(i,j)}, \lambda) p(\lambda) =$$
$$\mathcal{GS}(\boldsymbol{\gamma}_{\text{exp}}^{(i,j)}|\mathbf{exp}(\mathbf{h}_{\text{exp}}^{(i,j)}(\mathbf{x}, \mathbf{a}_{\text{exp}}^{(i,j)}, \lambda)), t) \prod_{k=1}^{N} \mathcal{GS}(\boldsymbol{\gamma}_{\text{comp}}^{(i,j),k}|\mathbf{exp}(\mathbf{h}_k^{(i,j)}(\mathbf{x}, \mathbf{a}_k^{(i,j)}, \lambda)), t) \mathcal{U}(\lambda|a, b), \tag{6}$$

where $\mathbf{h}_{\text{exp}}^{(i,j)}, \mathbf{h}_k^{(i,j)}$ are hypernetworks, vectors $\mathbf{a}_{\text{exp}}^{(i,j)}$ and $\mathbf{a}_k^{(i,j)}$ are parameters of hypernetworks.

3 Computational experiment

3.1 Basic experiment

In this experiment, it is required to establish a relationship between modality and architecture. The first modality is a regular MNIST dataset. The second modality is a MNIST dataset in which each picture is transposed being a matrix. RBF is used as a hypernet. Two main components (PCA) are the network input. Note that we don't use λ coefficient. The network consists of a single cell. After training, we took 5 batches from each modality and perforemed average pooling after PCA. Thus, we have obtained 5 average representations of each modality. The result is that the architectures for each batch are identical.

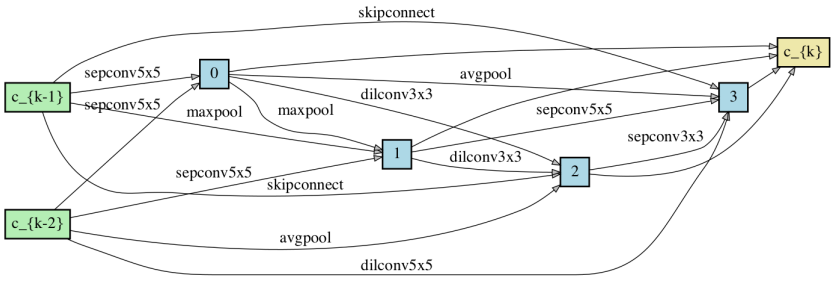

Fig. 1: Obtained architecture

4 Notation

– $\mathfrak{D} = \mathfrak{D}_{\text{train}} \sqcup \mathfrak{D}_{\text{val}}$ – dataset

- $\mathbf{x} \in \mathbf{X}$ – object from object space
- $y \in \mathbf{Y}$ – target from target space
- V – set of vertices
- E – set of edges
- $\mathbf{x}^{(i)}$ – intermediate representation of the object $\mathbf{x}$ corresponding to the node i
- $\mathbf{g}^{(i,j)}$ – nonlinear mapping corresponding the edge (i, j)
- $\vec{\mathbf{g}}^{(i,j)}$ – vector of all available mappings for the edge (i, j)
- $\boldsymbol{\alpha}^{(i,j)}$ – structural parameters for the edge (i, j)
- $\hat{\mathbf{g}}^{(i,j)}$ – mixed operation
- $\mathcal{L}_{\text{train}}$ and $\mathcal{L}_{\text{val}}$ – loss functions on $\mathfrak{D}_{\text{train}}$ and $\mathfrak{D}_{\text{val}}$ respectively
- $\mathbf{w} \in \mathbb{R}^n$ – vector of model parameters
- $\vec{\mathbf{g}}_k^{(i,j)}$ – vector of experts having the same type (ex. dilated convolution), but having different complexity
- $\boldsymbol{\gamma}^{(i,j)}$ – structural parameters
- $\boldsymbol{\gamma}_{\text{exp}}^{(i,j)}$ – structural parameters from N - 1 dimensional simplex. They control importance of each of N experts
- $\boldsymbol{\gamma}_{\text{comp}}^{(i,j),k}$ – structural parameters from $\dim \vec{\mathbf{g}}_k^{(i,j)}$ - 1 dimensional simplex. They control importance of each mapping among $\vec{\mathbf{g}}_k^{(i,j)}$
- $\Lambda \subset \mathbb{R}$ – a set of regularization parameters λ
- $\mathbf{h}_{\text{exp}}^{(i,j)}(\mathbf{x}, \mathbf{a}^{(i,j)}, \lambda)$ – hypernetwork, where $\mathbf{a}_{\text{exp}}^{(i,j)}$ is a vector of its parameters
- $\mathbf{h}_k^{(i,j)}(\mathbf{x}, \mathbf{a}_k^{(i,j)}, \lambda)$ – hypernetwork, where k is a number of expert, $\mathbf{a}_k^{(i,j)}$ is a vector of the hypernetwork parameters
- $\mathcal{GS}(\boldsymbol{\alpha}, t)$ – gumbel-softmax distribution

5 Conclusion

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. Maddison, C.J., Mnih, A., Teh, Y.W.: The concrete distribution: A continuous relaxation of discrete random variables. arXiv preprint arXiv:1611.00712 (2016)
11. Nayman, N., Noy, A., Ridnik, T., Friedman, I., Jin, R., Zelnik-Manor, L.: Xnas: Neural architecture search with expert advice. arXiv preprint arXiv:1906.08031 (2019)
12. Yakovlev, K.: https://github.com/Intelligent-Systems-Phystech/2021-Project85
13. Yao, Q., Xu, J., Tu, W.W., Zhu, Z.: Efficient neural architecture search via proximal iterations. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 6664–6671 (2020)
14. Yaushev, F.Y., Isachenko, R.V., Strijov, V.: Concordant models for latent space projections in forecasting. Sistemy i Sredstva Informatiki [Systems and Means of Informatics] **31**(1), 4–16 (2021)