
Differential neural ensemble search with diversity control

A Preprint

P. Babkin, K. Yakovlev, K. Petrushina, O. Bakhteev,

Abstract

In our research, we investigate a novel method for sampling deep learning models using a hypernetwork. The hypernetwork is a neural network that controls the diversity of the models by translating a real number representing ensemble diversity into a sampled neural network architecture. Architectures are obtained in a one-shot manner by perturbing from the base architecture in terms of the Jensen-Shannon divergence (JSd). We evaluate the performance of the proposed algorithm by conducting experiments on the Fashion-MNIST and CIFAR-10 datasets, comparing the resulting ensembles with those sampled by other searching algorithms.

Keywords differential search · neural ensembles · hypernetwork · diversity control

1 Introduction

Nowadays methods of neural architecture search (NAS) are well-explored and proved to be an effective way of creating more effective and efficient neural networks [2, 11, 4]. Some of these methods use different ways to smooth out the architecture so optimum for it can be found by wide range of methods for smooth optimization problems. On the other hand, neural ensemble search (NES) is modern and not as well investigated problem as NAS, although it is known that ensembles of deep learning models show better results in different applied problems [3]. Our paper investigates an method of sampling deep learning models in a new way that gives compatible results and has its own sphere of implementations.

Our method takes the result of NAS as a base architecture than it samples architectures which are close to the optimal one in terms of Jensen-Shannon divergence (JSd). Architectures differ in terms of λ , idea of method is shown in Fig. 1. Basic architecture is gained with $\lambda = \lambda_1$. Blue ellipses are equidistant surfaces of architectures. Starting with diverse parameter $\lambda = \lambda_2$ resulting architecture performs unacceptable accuracy, so architectures beyond the surface are not included into ensemble. Method can control whether sampled architectures are close enough to the optimal one so they perform good accuracy on the original dataset and are diverse enough so every architecture makes its own contribution to the final answer.

To sample architectures we use hypernetwork [1]. This network generates parameters for another network, which is called target network. Previously hypernetworks were intended to control different characteristics such as complexity of architecture [9] or parameters of the target model [7] in several modern papers. In our paper it controls diversity of the target models, so every sampled model differs from previously sampled ones in terms of JSd.

The hypernetwork uses JSd to measure difference between two architectures which is symmetric and finite in contrast to more popular Kullback–Leibler divergence. Our main idea of sampling different model is to use a regularizer, based on JSd as a source of diversity.

This way we sample deep learning models in one-shot unlike NES for Uncertainty Estimation [10], where every sampled models must be tested and least appropriate architectures are excluded from ensemble. In this way our method is similar to NES via Bayesian Sampling [6], but we sample architectures using different approach, posterior distribution for architectures is gained based on the optimal one. To sum up the scheme of our method: find a base architecture using DARTS, sample architectures in one-shot via differentiable algorithm. Inference answer is ensemble of the sampled deep learning models.

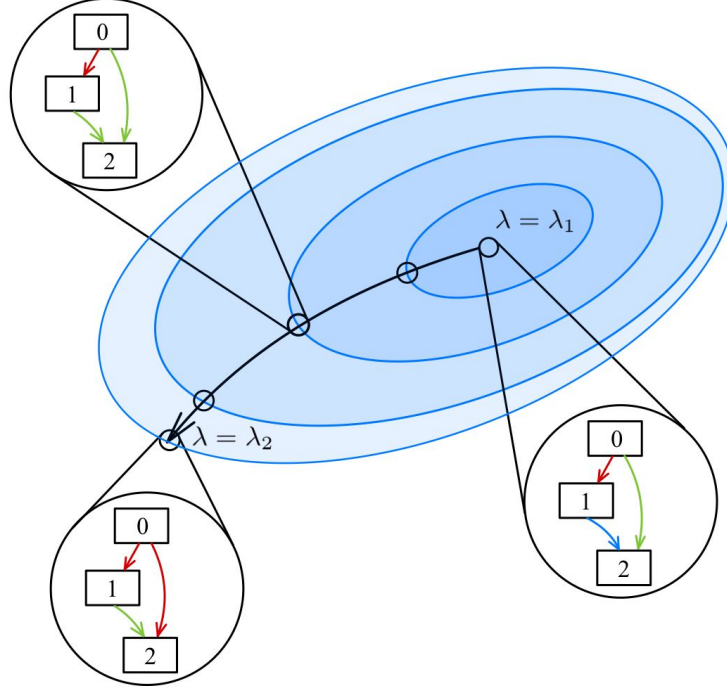


Рис. 1: Architecture space with JSd metrics. Blue ellipses are equidistant surfaces. The space is continuous but has varying preferred operations across different architectural levels. Architectures with different operations are pictured separately on the picture.

We conduct experiments on CIFAR and MNIST datasets to evaluate performance of the proposed method in terms of accuracy and time. Also we compare the performance with state-of-art NES algorithms [10, 6].

2 Problem statement

In our paper, we address the problem of classification. We assume that the dataset $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ is given, where \mathbf{X} represents the feature matrix and \mathbf{y} represents the target vector. The dataset is divided into training and validation subsets, for which the loss functions \mathcal{L}_{train} and \mathcal{L}_{val} are specified, respectively. In general, these functions depend on the feature matrix, target vector, and the predicted outputs of the training model. However, in our paper, we will not explicitly state their dependence on the dataset, assuming it implicitly. Therefore, we will state that the loss functions depend only on the predicted outputs of the model.

Let $\mathcal{V} = \{1, \dots, N\}$ be a set of vertices, where N is a number of vertices, and $\mathcal{E} = \{(i, j) \in \mathcal{V} \times \mathcal{V} \mid i < j\}$ a set of edges between them. A set of possible operations \mathcal{O} usually contains pooling, convolutions, etc. For each edge there is an operation $o \in \mathcal{O}$ that transits information from one node to another.

Contrary to the selection of one single architecture in conventional NAS algorithm [2, 5, 12], this paper focuses on the problem of selecting a well-performing neural network ensemble with diverse architectures from the NAS search space, i.e., neural ensemble search (NES).

In order to facilitate our study, several key terms are defined. The architecture of a model, i.e. a set of node operations, is denoted by α . For a fixed architecture, optimal parameters are denoted by \mathbf{w}_α^* . The optimal architecture resulting from neural architecture search (NAS) is denoted as α^* . To quantify architectural diversity, we define λ , a real number ranging from 0 to Λ . The corresponding architecture for a given λ is denoted by $\alpha(\lambda)$, which can be obtained by solving a specific problem. The output of an architecture α , given its corresponding model parameters \mathbf{w}_α , is denoted by $f(\mathbf{w}_\alpha, \alpha)$. Finally, the set of architectures included in the ensemble is denoted by \mathcal{S} .

Given the ensemble scheme, NES can be formally framed as

$$\min_S \mathcal{L}_{val} \left(\frac{1}{|S|} \sum_{\alpha \in S} f(\mathbf{w}_\alpha^*, \alpha) \right)$$

$$s.t. \forall \alpha \in S \mathbf{w}_\alpha^* = \arg \min_{\mathbf{w}} \mathcal{L}_{train}(f(\mathbf{w}_\alpha^*, \alpha))$$

3 Method

Below we briefly described our method for solving problem of classification with dataset $\mathcal{D} = (\mathbf{X}, \mathbf{y})$. For each object $\mathbf{x} \in \mathbf{X}$ there is a label $y \in \mathbf{y}$. We solve the problem via NES, sampling architectures according to novel methodology which is formally described below.

3.1 Architecture

NAS algorithms search for optimal architecture. As it was mentioned below, architecture of neural network is a set of operations between nodes. In NAS methods architecture is a vector constructed by following rules. For each edge $(i, j) \in \mathcal{E}$, $\alpha^{(i,j)}$ is a vector, which assigns impact of each operation. α is a concatenation of all structural parameters vectors $\alpha^{(i,j)}$.

3.2 Regularizer and heuristic

Source of diversity in our method is a regularizer with diversity parameter λ . By subtracting it from a loss function we promote method for finding an architecture which differs from the optimal one.

We use Jensen-Shannon divergence to measure diversity of two architectures. NES algorithms give discrete architecture as a result of their work, i.e. initially optimal architecture α_{init}^* is discrete, so JSd cannot be calculated with α_{init}^* as an argument. We smooth the architecture using smooth parameter τ to solve this problem.

$$\alpha^* = (1 - \tau)\alpha_{init}^* + \tau \frac{1}{|\mathcal{O}|}$$

Assuming τ close to one we obtain a smoothed architecture that contains the same information about architecture as a initial one, but we also can use it in JSd.

3.3 Diversity control via hypernetwork

In order to control diversity we employ the concept of hypernetwork. A hypernetwork is a parametric mapping from $[0, \Lambda]$ to the set of model structural parameters [9].

$$\alpha : [0, \Lambda] \times \mathbb{R}^u \rightarrow \mathbb{R}^s$$

Where \mathbb{R}^u is a hypernetwork parametric space and \mathbb{R}^s is space of model structural parameters. In this terms α can be redefined using hypernetwork

$$\alpha = \alpha(\lambda, \mathbf{a}) \text{ or } \alpha^{(i,j)} = \alpha^{(i,j)}(\lambda, \mathbf{a}^{(i,j)})$$

In this paper each function $\alpha^{(i,j)}$ is a piecewise linear function

$$\alpha^{(i,j)}(\lambda, \mathbf{a}^{(i,j)}) = \sum_{k=0}^{N-1} \left(\frac{\lambda - r_k}{r_{k+1} - r_k} \mathbf{a}_k^{(i,j)} + \left(1 - \frac{\lambda - r_k}{r_{k+1} - r_k} \right) \mathbf{a}_{k+1}^{(i,j)} \right) I[\lambda \in [r_k, r_{k+1}]]$$

In our method λ is sampled from predefined distribution $p(\lambda) = U(0, \Lambda)$ and new architecture is obtained from hypernetwork.

3.4 Final statement

Upon concluding the method description, a reformulated problem can be postulated: specifically, architectures are obtained via hypernetwork, and the regularizer is subtracted from the loss function. A varied parameter λ , ranging from 0 to Λ , is distributed according to a uniform distribution.

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \mathbb{E}_{\lambda \sim U(0, \Lambda)} [\mathcal{L}_{val}(w^*, \boldsymbol{\alpha}(\lambda)) - \lambda JS(\boldsymbol{\alpha}^*, \boldsymbol{\alpha}(\lambda))] \\ s.t. \ w^* = \arg \min_{\mathbf{w}} \mathbb{E}_{\lambda \sim U(0, \Lambda)} [\mathcal{L}_{train}(\mathbf{w}, \boldsymbol{\alpha}(\lambda))] \end{aligned}$$

Following problem resolution, a new architecture is acquired and subsequently included into the ensemble. It is assured that said architecture is diverse enough from its counterparts and performs an acceptable level of accuracy. Formal algorithm is written down as Algorithm 1.

Algorithm 1

```

Initialize:  $N \in \mathbb{N}$ ,  $\mathcal{S} = \emptyset$ , hypernetwork
 $S \leftarrow \{\boldsymbol{\alpha}^*\}$  ▷ Result os NAS
for  $i = 1, \dots, N$  do
    Sample  $\lambda \sim U(0, \Lambda)$ 
     $\mathcal{S} \leftarrow \mathcal{S} \cup \{\alpha(\lambda, \boldsymbol{\alpha}^*)\}$  ▷  $\alpha$  gaind from hypernetwork
end for
Return:  $\mathcal{S}$  as a resulting ensemble

```

4 Computational experiment

In this study, we conducted a series of experiments on the fashionMNIST dataset [8] using a convolutional neural network (CNN) model. Specifically, we utilized a one-layer architecture with six nodes and seven possible operations for each edge, as outlined in the DARTS methodology [2]. Our optimizer was stochastic gradient descent (SGD), and we employed a cosine annealing learning rate scheduler with a minimum learning rate of 0.001. The loss criterion utilized in this study was CrossEntropyLoss.

First goal of computational experiment is to investigate dependence of resulting architecture’s performance and their diversity on λ . Second goal is to investigate effectiveness of ensembling. Main question is how to choose architectures that on one hand are diverse enough and on other hand perform acceptable accuracy.

4.1 Comparison of architectures

We employed the Differentiable Architecture Search (DARTS) algorithm to sample a base architecture with $\lambda = 0$. Subsequently, we sampled architectures for values of λ ranging from 1 to 40, recording the resulting architectural structures and their corresponding classification performance. Specifically, we sought to examine the impact of varying levels of architectural diversity on model accuracy, with the aim of identifying optimal architectures for the given task.

The results of the investigation into various neural network architectures are presented in Tab. 1. The table displays the maximum accuracy achieved by the sampled architectures, along with the number of matched operations between nodes. Last one indicates the number of edges between nodes that contain the same operation in the obtained architecture and the basic architecture, gained from NAS. The graphical visualization of the architectures can be found in Fig. 2.

Although the instability of the graph presented in Fig.2(a) is noticeable, it is evident that the accuracy decreases as the architecture diversity parameter λ increases. However, contrary to our expectations, the amount of intersecting edges depicted in Fig.2(b) does not align with our anticipated results. The reason for this can be attributed to the non-convexity of the optimized function, which causes the algorithm to converge to different minima. However, we can obtain a higher number of intersecting edges by using a negative λ .

4.2 Ensembling effectiveness investigation

To investigate the effectiveness of ensembling, we employed the following methodology. We randomly sampled three architectures corresponding to different diverse parameters λ chosen randomly and uniformly from a range of 0 to Λ . The obtained architectures were then included in an ensemble, and the accuracy of the resulting ensemble was measured. By changing the diversity limit Λ , we were able to control the diversity of our deep learning models. The graphical results of our experiment are depicted in Fig.3.

λ	accuracy, %	matched operations
0 (optimum)	92.10	18
5	89.66	2
10	90.59	2
15	90.09	1
20	88.52	1
25	88.54	1
30	87.42	2
35	86.41	1
40	88.67	2

Таблица 1: Table to test captions and labels.

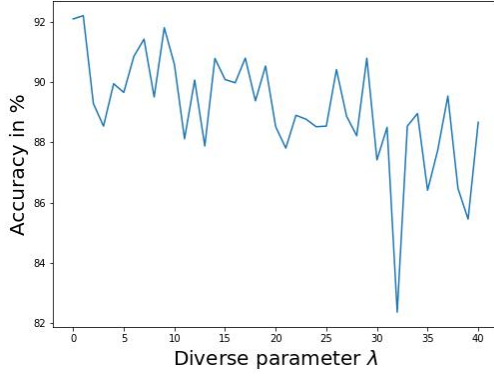
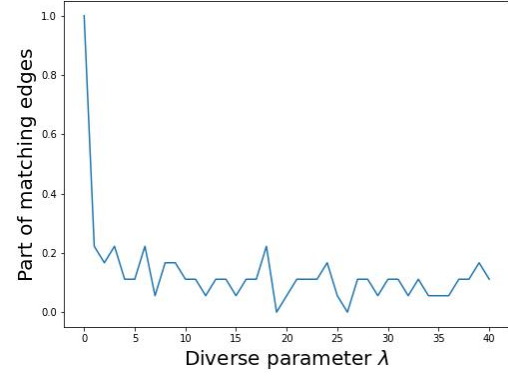
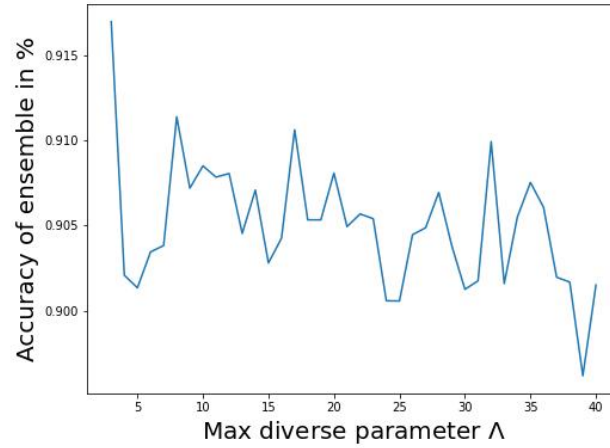
(a) Dependence of performed accuracy on diversity parameter λ (b) Dependence of matching edges number on diversity parameter λ

Рис. 2: Graphical results of computational experiment

Рис. 3: Dependence of effectiveness of deep learning models ensemble on diversity limit Λ

5 Conclusion

In this paper, we proposed a novel method for sampling ensembles of deep learning models with diversity control. Our method utilizes a hypernetwork to generate diverse architectures by perturbing a base architecture in terms of Jensen-Shannon divergence. The diversity of the ensemble is controlled by a penalty term added to the loss function, which encourages the ensemble members to be diverse. We conducted experiments on the Fashion-MNIST dataset and demonstrated that our method performs compatible results in terms of accuracy and architectural diversity. Overall, our proposed method shows potential for practical applications in deep learning ensembling.

Список литературы

- [1] D. Ha, A. Dai, and Q. V. Le. Hypernetworks. 2016.
- [2] H. Liu, K. Simonyan, and Y. Yang. DARTS: Differentiable architecture search. In Proc. ICLR, 2019.
- [3] A. R. Narayanan, A. Zela, T. Saikia, T. Brox, and F. Hutter. Multi-headed neural ensemble search. 2021.
- [4] N. Nayman, A. Noy, T. Ridnik, I. Friedman, R. Jin, and L. Zelnik-Manor. XNAS: Neural architecture search with expert advice. In Proc. NeurIPS, pages 1975–1985, 2019.
- [5] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean. Efficient neural architecture search via parameter sharing. In Proc. ICML, pages 4092–4101, 2018.
- [6] Y. Shu, Y. Chen, Z. Dai, B. Kian, and H. Low. Neural ensemble search via bayesian sampling. 2022.
- [7] J. von Oswald, C. Henning, B. F. Grewe, and J. Sacramento. Continual learning with hypernetworks. 2022.
- [8] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017.
- [9] K. D. Yakovlev, O. S. Grebenkova, O. Y. Bakhteev, and V. V. Strijov. Neural architecture search with structure complexity control. 2022.
- [10] S. Zaidi, A. Zela, T. Elsken, C. C. Holmes, F. Hutter, and Y. W. Teh. Neural ensemble search for uncertainty estimation and dataset shift. In Proc. NeurIPS, 2022.
- [11] A. Zela, T. Elsken, T. Saikia, Y. Marrakchi, T. Brox, and F. Hutter. Understanding and robustifying differentiable architecture search. In Proc. ICLR, 2020.
- [12] B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. In Proc. ICLR, 2017.