
Differential neural ensemble search with diversity control

A Preprint

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

Abstract

This paper introduces a new method of deep learning ensemble construction. In our research we investigate an algorithm of sampling the deep learning models using hypernetwork. Hypernetwork is a neural network which controls diversity of the models. It translates a real number into an architecture of a sampled neural network. This method samples deep learning models in one-shot regime, without any additional calculations losses. To evaluate the performance of the proposed algorithm, we conduct experiments on the Fashion-MNIST and CIFAR-10 datasets and compare the resulting ensembles with ones 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 [1, 6, 3]. 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 [2]. 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). Method can control whether sampled architectures are close enough to the optimal one so they give good results 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. 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 [5] or parameters of the target model [4] in several modern investigations. 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 are able to sample deep learning models in one-shot, without and additional computational losses. 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.

We conducted 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 and NAS algorithms [1?].

2 Method

2.1 Problem statement

Contrary to the selection of one single architecture in conventional NAS algorithm, 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). We use following terms

- α – an architecture of a model, i.e. a set of operations between nodes
- α^* – a NAS resulting architecture, i.e. an optimal architecture
- w_α^* – optimal parameters for architecture α
- λ – a measure of diversity, a real number from 0 to Λ
- $\alpha(\lambda)$ – an architecture that corresponds λ
- $f(w_\alpha, \alpha)$ – the output of an architecture α and model parameter w_α
- S – a set of architectures included into ensemble
- $\mathcal{L}_{train}, \mathcal{L}_{val}$ – the training and validation losses, respectively. We imply their dependence on preset dataset and do not denote it explicitly, because we don't deal with it

Given the ensemble scheme, NES can be formally framed as

$$\begin{aligned} \min_S \mathcal{L}_{val} \left(\frac{1}{|S|} \sum_{\alpha \in S} f(w_\alpha^*, \alpha) \right) \\ s.t. \forall \alpha \in S \ w_\alpha^* = \arg \min_w \mathcal{L}_{train}(f(w_\alpha^*, \alpha)) \end{aligned}$$

We rearranged the problem: general for all architectures. Also architectures differ in terms of λ so resulting functions can be calculated in terms of expected values.

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

2.2 Computational experiment

In our model λ is a random value, distributed according to uniform distribution from 0 to Λ ($\sim U(0, \Lambda)$). Main goal of basic experiment was to estimate Λ and also to investigate dependence of resulting architecture's performance on λ .

2.2.1 Experiment planning

In our basic experiment we run the algorithm on fashionMNIST dataset for several different λ and looked at resulting architecture and accuracy. Further we will use λ distributed randomly, but in basic experiment it is fixed.

2.2.2 Preliminary report

Obtained results are depicted in the table below. Maximal accuracy and amount of matching edges between resulting and optimal architecture are written down.

This information allows us to choose $\Lambda = 32$. However amount of matching edges does not really coincide with our expectations. Architectures almost do not intersect. It can be explained by the fact that the optimized function is not convex, so algorithm fall into different minimums, however we can get more matching edges by using negative λ .

Таблица 1: preliminary results

λ	accuracy, %	matched edges
0 (optimum)	91.24	-
1/2	91.24	4
2	91.26	3
32	89.21	2
64	87.09	1

Список литературы

- [1] H. Liu, K. Simonyan, and Y. Yang. DARTS: Differentiable architecture search. In Proc. ICLR, 2019.
- [2] A. R. Narayanan, A. Zela, T. Saikia, T. Brox, and F. Hutter. Multi-headed neural ensemble search. 2021.
- [3] 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.
- [4] J. von Oswald, C. Henning, B. F. Grewe, and J. Sacramento. Continual learning with hypernetworks. 2022.
- [5] K. D. Yakovlev, O. S. Grebenkova, O. Y. Bakhteev, and V. V. Strijov. Neural architecture search with structure complexity control. 2022.
- [6] A. Zela, T. Elsken, T. Saikia, Y. Marrakchi, T. Brox, and F. Hutter. Understanding and robustifying differentiable architecture search. In Proc. ICLR, 2020.