



HAL
open science

Bayesian optimization for NAS with pretrained deep ensembles

Housseem Ouertatani, Cristian Maxim, Smail Niar, El-Ghazali Talbi

► **To cite this version:**

Housseem Ouertatani, Cristian Maxim, Smail Niar, El-Ghazali Talbi. Bayesian optimization for NAS with pretrained deep ensembles. International Conference in Optimization and Learning (OLA), May 2023, Malaga, Spain. hal-04076075

HAL Id: hal-04076075

<https://hal.science/hal-04076075v1>

Submitted on 29 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Bayesian optimization for NAS with pretrained deep ensembles

H. Ouertatani¹, C. Maxim², S. Niar³ and E-G. Talbi⁴

¹ IRT SystemX & INRIA Lille - France

`housem.ouertatani@irt-systemx.fr`

² IRT SystemX - France

`cristian.maxim@irt-systemx.fr`

³ UPHF & LAMIH UMR CNRS, Valenciennes - France

`smaill.niar@uphf.fr`

⁴ University of Lille & INRIA - France

`el-ghazali.talbi@univ-lille.fr`

1 Introduction

Neural architecture search (NAS) is the use of a search algorithm to find, according to one or many criteria, the most suitable neural architecture for a given task, within a search space of neural network architectures. Among prominent search methods used in NAS literature, we can find reinforcement learning (RL)[1], evolutionary algorithms (EAs)[2, 3], and bayesian optimization (BO)[4, 5].

In this paper, we present a BO-based method, where instead of using Gaussian Processes which are usually associated with BO, we use deep ensembles as performance predictors for candidate neural networks. Specifically, we explore the potential of pretraining the ensemble networks as a way to mitigate their need for more data compared to GPs

The idea is to use pretraining to accelerate the training of the deep ensemble, in order to obtain a good prediction performance early on in the optimization process. In the case of NAS, we have access to zero-cost metrics, which can be calculated quickly and without the need to train the candidate networks, by far the most costly part of evaluating them.

By using existing and widely used NAS benchmarks, we show the improvement pretraining brings to the deep-ensemble based method, as a result accelerating the search process.

2 Bayesian optimization for NAS with deep ensembles

At a glance, the description of the procedure is as follows. At each iteration, the best points to evaluate are selected, according to a criteria calculated using the prediction of the ensemble, and which balances exploration and exploitation, known as the acquisition function. In our work, we used *probability of improvement*, ie. the probability of improving over the current best observed value. Subsequently, these selected candidates are observed or evaluated, ie. fully trained and tested on the task. Their evaluation data is then added to the training data for the deep ensemble, which is updated for the next iteration. Figure 1 describes the overall procedure of using a deep ensemble for BO.

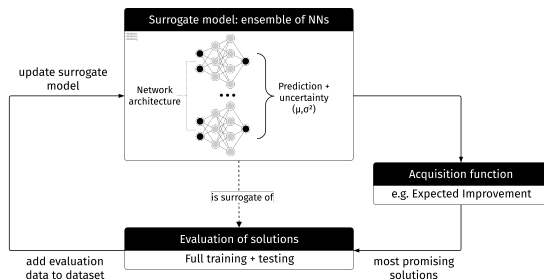


Fig. 1. Bayesian optimization for NAS with deep ensembles

Before starting the search, the deep ensemble is first pretrained on zero-cost metrics. In the following section, we describe the pretraining procedure for the deep ensemble.

Pretraining step

In order to increase the sample efficiency obtained with the deep ensemble, we pretrain them on zero-cost metrics, ie. metrics whose measurement doesn't require the prior training of the candidate networks. Among such metrics we mention FLOPs, number of trainable parameters, or the average time of a forward pass (or latency).

The idea is to force the last layer of the deep ensemble networks to have meaningful representations of the architectures in the search space. The embeddings generated by these last layers for input architectures need to hold relevant information that helps in predicting the eventual performance of these architectures once the BO iterations begin.

In order to achieve this, we use a separate prediction head for each of the metrics we are using as targets. Then, the weights of the deep ensemble networks are updated using backpropagation. The forward pass and optimization steps are performed for all the different metrics at the same time, as we found this achieved better performance. Our intuition in explaining this is it ensures the last layer doesn't specialize on one of the target metrics, instead being as general as possible.

Figure 2 describes the pretraining procedure.

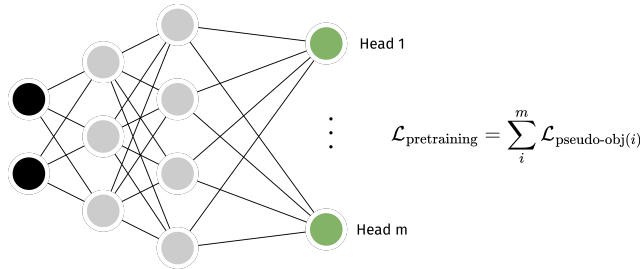


Fig. 2. Pretraining procedure for one network of the deep ensemble networks

First results

Our tests were performed on NASBench101, NASBench201, and NATS-Bench-size benchmarks. These benchmarks can be queried to obtain the accuracies, as well as other measurements, for the networks described in their search spaces.

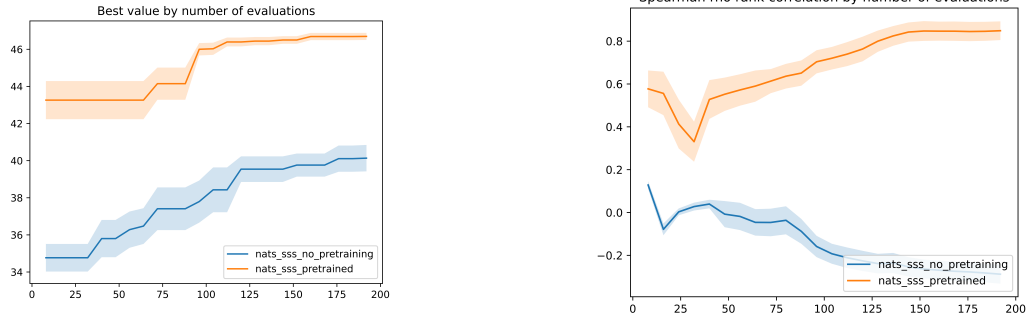
One of the most used ways to compare predictor-based NAS methods is the correlation or ranking correlations between the predictions, and the real ranking of the networks. Table 1 shows the difference in ranking correlations measured by the Kendall-Tau and Spearman-Rho ranking correlations, and the Pearson-R correlation of the accuracies. The first two columns also report the best found accuracy on CIFAR10 and CIFAR100 within the evaluations budget.

Table 1. Comparison of results on NATS-Bench-size with and without pretraining

	C-10	C-100	τ	ρ	R
No pretraining	92.99	69.92	0.46	0.65	0.61
With pretraining	93.65	70.72	0.73	0.90	0.938

To situate the correlations with regards to other methods in the literature, we can mention the paper describing BossNAS, which reports a Kendall-tau measure on the same benchmark of $\tau = 0.59$, a Spearman-Rho of $\rho = 0.76$, and a Pearson-R of $R = 0.79$.

We can also plot the evolution of the best observed value, and of the Spearman-Rho correlation, as the number of evaluations performed increases.



Results obtained on NAS-Bench-101, NAS-Bench-201 and NATS-Bench-Size are competitive with state-of-the-art predictor-based methods, getting close to or better results in most datasets offered by the benchmarks (CIFAR10, CIFAR100 and ImageNet). Further improvement of the architecture of the ensemble networks or their training hyperparameters can potentially offer further performance gains.

3 Conclusion and further perspectives

In this work, we have presented a method to use Bayesian optimization powered by deep ensembles for the NAS problem. One of the advantages of using a deep ensemble is the ability to easily pretrain it and keep the weights for use in the Bayesian optimization procedure. Pretraining allows us to gain in sample efficiency, and enables the deep ensemble to produce good quality predictions from fewer datapoints.

This work is a first step which will be built upon for multi-objective NAS, to produce energy-efficient and low-latency models. The deep ensemble can be used in a similar fashion as it was used in the pretraining step described previously: the weights of the initial parts of the network could be shared and used to produce predictions for the different objectives, potentially unlocking further sample efficiency by the inclusion of additional information. Our future work will focus on multi-objective aspects, both in terms of improving the search method for multi-objective optimization, and in terms of the objectives that will be optimized.

References

1. Hsu, Chi-Hung and Chang, Shu-Huan and Liang, Jhao-Hong and Chou, Hsin-Ping and Liu, Chun-Hao and Chang, Shih-Chieh and Pan, Jia-Yu and Chen, Yu-Ting and Wei, Wei and Juan, Da-Cheng: MONAS: Multi-Objective Neural Architecture Search using Reinforcement Learning.
2. Thomas Elsken and Jan Hendrik Metzen and Frank Hutter: Efficient Multi-Objective Neural Architecture Search via Lamarckian Evolution. International Conference on Learning Representations (2019)
3. Lu, Zhichao and Whalen, Ian and Boddeti, Vishnu and Dhebar, Yashesh and Deb, Kalyanmoy and Goodman, Erik and Banzhaf, Wolfgang: NSGA-Net: neural architecture search using multi-objective genetic algorithm. Proceedings of the Genetic and Evolutionary Computation Conference (2019)
4. White, Colin and Neiswanger, Willie and Savani, Yash: BANANAS: Bayesian Optimization with Neural Architectures for Neural Architecture Search. AAAI Conference on Artificial Intelligence (2021)
5. Cai, Lile and Barneche, Anne-Maëlle and Herbout, Arthur and Foo, Chuan Sheng and Lin, Jie and Chandrasekhar, Vijay Ramaseshan and Sabry, Mohamed M.: TEA-DNN: the Quest for Time-Energy-Accuracy Co-optimized Deep Neural Networks.