Exercise 4 deep learning lab 2018

Max Fuchs
Matriculation number: 4340529
maxfuchs@gmx.de

1. Introduction

The task in exercise 4 was to use Bayesian optimization (BO) and Hyperband (HB) to optimize the hyperparameters of an surrogate benchmark. In a final step, BO and HB were combined. The resulting optimizer Bohb was as well tested on the surrogate.

2. Implementation

As already mentioned in the introduction, a surrogate benchmark was used to test the different optimization algorithms. This surrogate enables to test hyperparameter configurations in a very short time, by using a regression model of the hyperparameter space . The reg. Model is based on a prior training with a large set of different hyperparameter settings. In the following the hyperparameters of a CNN, namely the learning rate, batch size and the filter sizes for tree layers, are optimized.

3. Optimizers

3.1 Bayesian optimization

The basic idea of Bayesian optimization is to keep track of past evaluation results, and use them to predict the next hyperparameter setting. Therefore BO chooses the next setting in an informed manner, in contrast to grid- or random search [1]. The Bayesian estimator was implemented using the emukit library.

The results of an optimization for 20 iterations can be seen in figure 1. It can be seen that a local validation error minimum can be found already after 5 iterations. Interestingly the found minimum, is in this case not further decreased with increasing runtime.

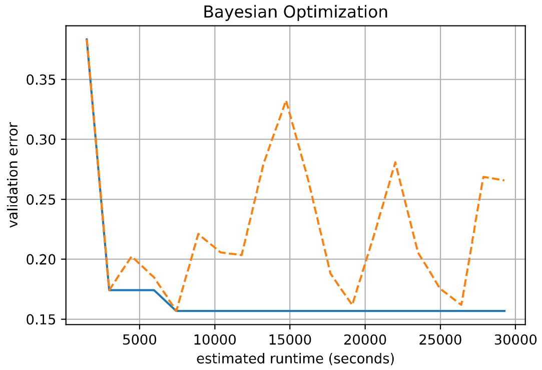

figure 1:validation error over cumulated time, for bayesian optimization on surrogate for 20 epochs

3.1. Hyperband

This optimization algorithm underlays the principle, that if an parameter combination performs best in long terms, it is likely that it will perform in the top half of a random set of configurations after a small number of iterations. To achieve this, a set of random parameter configurations of size n is drawn, with r being the initial number of iterations. The algorithm executes successive halving in several combinations of n and r, to trade off breadth versus depth[2]. These combinations can be seen in figure 2, with s denoting the different combinations, and representing the tradeoff. The blue dots, represent sets that are trained. These get trained for r epochs. After that only the best performing half of the set keeps on training for $2 \cdot r$ epochs (orange dots). This is done until the stopping criterion is met. In each training step with half the amount of sets and double the epoch budget from the last iteration. The last configurations s = 0, is therefor trained on a very small amount of parameter sets with the maximum budget on epochs.

The best validation error for each run of s can be seen in table 1. Configuration s = 1, outperforms the other configurations. This shows that a tradeoff between breadth and depth is necessary.

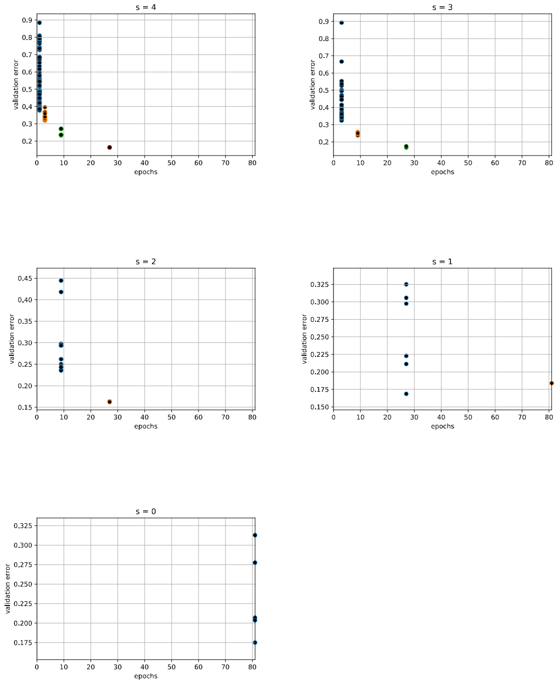

figure 2:Hyperband varying tradoffs between breadth and width

s	0	1	2	3	4
Val. error	0.175	0.158	0.162	0.163	0.163

table 1: incumbent validation error on surrogate for successive halving steps of Hyperband

3.2 Combination Bayesian optimization and Hyperband (Bohb)

This approach tries to reduce the effect that Hyperband might take exponentially log to approach the global optimum, due to its random start configurations. To achieve this, only a fraction of the Hyperband input configurations is drawn randomly. The other fraction is generated using Bayesian optimization. figure 3 shows the performance, for random and model based hyperparameter configuration sets. It can be seen that in the first time of the training, only random configurations are drawn. This is intended and is necessary to provide enough start data for the Bayesian optimization. After about 150000 seconds, the first model drawn configurations are feed to Hyperband and get evaluated on the surrogate. For that a set of random samples is sampled from a multivariate Normal. These configurations are evaluated by the acquisition function. Finally the expected improvement of the acquisition function is minimized and the related configuration is used for Hyperband. It can also be seen that the model configurations are very close together. The amount these values are spread, represents the trade of between exploration in uncertain model areas and exploitation in the good regions of the configuration space. It can also be seen that after 150000 seconds still a part of the configurations are drawn randomly. This is also intended and makes sure the algorithm still performs as good as random search.

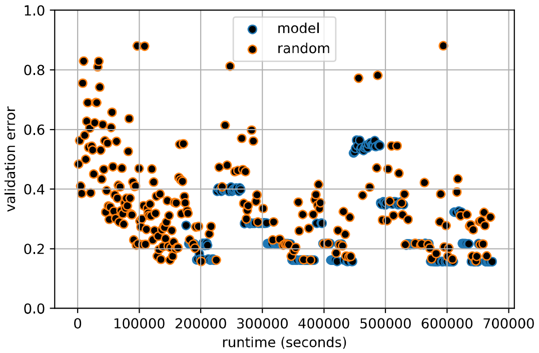

figure 3: validation error over cumulated time for Bohb estimated configurations, either random or model based

4. Conclusion

Bayesian optimization and Hyperband offer alternatives for hyperparameter optimization adverse random - and grid search. Both methods follow different approaches and bring their own dis- and advantages. Therefor the approach of Bohb is introduced, which tries to combine the best of the two algorithms.

5. References

[1] A Conceptual Explanation of Bayesian Hyperparameter Optimization for Machine Learning, Will Koehrsen, website: https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f access: 30.12.2018, 12:18

[2] Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization,Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, Ameet Talwalkar,18 Jun 2018