HyperUCB: Hyperparameter Optimization using Contextual Bandits

Maryam Tavakol[1], Sebastian Mair[2], and Katharina Morik[1]

[1] Technical University of Dortmund, Dortmund, Germany
{maryam.tavakol,katharina.morik}@tu-dortmund.de
[2] Leuphana University of Lüneburg, Lüneburg, Germany
mair@leuphana.de

Abstract. Setting the optimal hyperparameters of a learning algorithm is a crucial task. Common approaches such as a grid search over the hyperparameter space or randomly sampling hyperparameters require many configurations to be evaluated in order to perform well. Hence, they either yield suboptimal hyperparameter configurations or are expensive in terms of computational resources. As a remedy, Hyperband, an exploratory bandit-based algorithm, introduces an early-stopping strategy to quickly provide competitive configurations given a resource budget which often outperforms Bayesian optimization approaches. However, Hyperband keeps sampling iid configurations for assessment without taking previous evaluations into account. We propose HyperUCB, a UCB extension of Hyperband which assesses the sampled configurations and only evaluates promising samples. We compare our approach on MNIST data against Hyperband and show that we perform better in most cases.

Keywords: hyperparameter optimization · multi-armed bandits

1 Introduction

The performance of machine learning models highly depends on the choice of the hyperparameters. For many years, grid search was the standard approach for tuning the underlying models. However, with the emergence of more sophisticated models such as in deep learning, grid search is no longer practical due to the large hyperparameter space, and thus simpler approaches such as random search became more desirable and showed to be more effective [2].

Over the last few years, the problem of hyperparameter optimization has been successfully presented as metalearning using Bayesian optimization methods [3, 5, 10]. Nevertheless, bandit-based approaches exhibit superb performance in many scenarios [8, 9]. Li et al. [8] propose a method, called Hyperband (HB), for hyperparameter selection which, in their settings, outperforms Bayesian methods while providing a significant speed-up compared to those competitors. Hyperband is based on the successive halving approach [11] for improving random search by an adaptive allocation of available resources to different configurations.

However, Hyperband is an amended version of random search in which there is no learning to guide the search. In addition, despite the fact that Hyperband

Algorithm 1: Hyperband

input : R, η

initialization: $s_{max} = \lfloor \log_\eta R \rfloor$ and $B = (s_{max} + 1)R$;

for $s \in \{s_{max}, s_{max} - 1, \ldots, 0\}$ **do**

  $n = \lceil \frac{B}{R} \frac{\eta^s}{s+1} \rceil, \quad r = R\eta^{-s}$;

  Λ_s =get_hyperparameter_configuration(n);

  for $i \in \{0, \ldots, s\}$ **do**

    $n_i = \lfloor n\eta^{-i} \rfloor \ , \quad r_i = r\eta^i$;

    $\mathcal{L}(\Lambda_s) = \{\text{run\_then\_return\_val\_loss}(\lambda, r_i) \mid \lambda \in \Lambda_s\}$;

    $\Lambda_s = \text{top\_k}(\Lambda_s, \mathcal{L}(\Lambda_s), \lfloor \frac{n_i}{\eta} \rfloor)$;

  end

end

output: configuration λ with lowest validation loss seen so far

is highly efficient for finding a good configuration, it does not find an optimum fast enough. Hence, modeling the hyperparameter optimization as a learning problem is more reliable than a search algorithm. Therefore, instead of only sampling iid configurations of hyperparameters as Hyperband does, we propose to leverage the information of previous *batches* in order to pre-evaluate sampled configurations and to discard unpromising ones. This is done by a UCB bandit strategy in a contextual setting.

In this paper, we introduce *HyperUCB*, a model-based bandit framework, to accommodate exploitation into the purely exploratory algorithm of Hyperband. In HyperUCB, the arm selection is carried out by incorporating an Upper Confidence Bound (UCB) strategy [1] to guide the search within the iterations in order to balance exploration vs. exploitation. We further model the arms in a contextual setting which generalizes the model for unseen arms (i.e., configurations). Therefore, we employ a modified version of LinUCB [7] in our approach to achieve a model-based Hyperband for the task of hyperparameter optimization. Empirically, we show that our proposed approach either outperforms Hyperband or performs on par on optimizing the hyperparameters of a deep learning model.

2 Background

2.1 Problem Setting

Let $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$ be a data set and M be a learning algorithm. The data is usually split into a training set for optimizing the parameters of the model, a validation set for optimizing the hyperparameters and a test set for evaluating the overall performance of the model. Assume that $\mathcal{H}$ is the set of all possible hyperparameter configurations, we denote by $\mathcal{L}(\lambda)$ the loss of M using $\lambda \in \mathcal{H}$ on the validation set. The goal is to find the best hyperparameter configuration $\lambda^\star = \arg\min_\lambda \mathcal{L}(\lambda)$, which minimizes the validation loss for a given budget.

2.2 Hyperband

Hyperband (HB) is an anytime search algorithm based on multi-armed bandits to find the best configuration for a machine learning approach given limited resources. The method performs several iterations based on the available resources, and in each iteration repeatedly calls the SuccessiveHalving method [6] for choosing the best ones. Let R be the maximum budget available for training various instances of a model, then Hyperband conducts $s_{max} = \lfloor \log_\eta R \rfloor$ iterations for exploration, where η is the ratio of sampling the best arms.

Hyperband is outlined in Algorithm 1. Note that the evaluation of the hyperparameters $\lambda \in \Lambda_s$ in line 7 can be done in parallel. Within the algorithm, three methods are used. The method `get_hyperparameter_configuration(`n`)` returns a set Λ_s of $n \in \mathbb{N}$ hyperparameter configurations $\{\lambda_1, \ldots, \lambda_n\}$ sampled iid from a given hyperparameter space $\mathcal{H}$ of feasible configurations. Furthermore, by calling `run_then_return_val_loss(`λ, r`)`, we obtain the validation loss $\mathcal{L}(\lambda)$ of configuration λ and resource allocation r. Finally, `top_k(`$\Lambda_s, \mathcal{L}(\Lambda_s), k$`)` returns a subset of Λ_s of size k with the k lowest validation losses given in $\mathcal{L}(\Lambda_s)$.

2.3 Contextual Bandits

The multi-armed bandits in contextual settings benefit from the available information (context) to make a better decision at the time of action (arm) selection. That means, before making a decision, some context is shown to the bandits, and depending on the situation the decision might be different. The context could include the information about the current state, the attributes of the arms, or any other available data. A contextual bandit aims at finding a mapping between the contexts and their corresponding outcomes in order to minimize the total regret. Li et al. [7] propose LinUCB in which the outcome of every arm is modeled as a linear function of the context. In the next section, we present a modified form of LinUCB to design contextual Hyperband.

3 Contextual HyperUCB

In this section, we present our approach to upgrade Hyperband to a contextual bandit method using a UCB strategy. Let $\mathcal{H}$ be the space of all possible hyperparameter configurations for a machine learning approach. We are interested in finding $\lambda^\star \in \mathcal{H}$ that gives the best performance $y^\star$ in terms of the validation loss $\mathcal{L}$ of the model

$$\lambda^\star = \arg\min_\lambda \mathcal{L}(\lambda). \tag{1}$$

We assume that a hyperparameter configuration can be represented by a d-dimensional vector λ and model the contextual bandit as a linear function of the configurations. After learning the parameters $\boldsymbol{\theta}$ of the linear model, a new configuration λ can be evaluated as $\hat{y} = \boldsymbol{\theta}^\top \lambda$. The optimization problem in Equation (1) suggests a lower confidence bound strategy since we aim to minimize $\mathcal{L}$. However, by considering negative loss values $-y$, we can retain the usual upper

Algorithm 2: HyperUCB

input : R, η, α, γ

1 **initialization** (HB): $s_{max} = \lfloor \log_\eta R \rfloor$ and $B = (s_{max} + 1)R$;
2 **initialization** (UCB): $\boldsymbol{\theta} = \mathbf{0}_{d\times 1}$, $\boldsymbol{X} \leftarrow \emptyset_{0\times d}$, $A \leftarrow \gamma \boldsymbol{I}_{d\times d}$, $n_0 = \eta^{s_{max}}$;
3 **for** $s \in \{s_{max}, s_{max} - 1, \ldots, 0\}$ **do**
4 compute n and r as in HB;
5 Λ_s = top_ucb(get_hyperparameter_configuration(n_0), $\boldsymbol{\theta}, A, n$);
6 append λ to $\boldsymbol{X}$ $\forall \lambda \in \Lambda_s$ and initialize $y_\lambda = 0$;
7 **for** $i \in \{0, \ldots, s\}$ **do**
8 compute n_i and r_i as in HB;
9 **for** $\lambda \in \Lambda_s$ **do**
10 $A = A + \lambda\lambda^\top$;
11 $y_\lambda = -$run_then_return_val_loss(λ, r_i);
12 **end**
13 $\boldsymbol{\theta} = (\boldsymbol{X}^\top \boldsymbol{X} + \gamma \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$;
14 Λ_s = top_ucb($\Lambda_s, \boldsymbol{\theta}, A, \lfloor \frac{n_i}{\eta} \rfloor$);
15 **end**
16 **end**
17 **top_ucb** *($\Lambda_s, \boldsymbol{\theta}, A, n$)*:
18 $p_\lambda = \boldsymbol{\theta}^\top \lambda + \alpha\sqrt{\lambda^\top A^{-1} \lambda}$ $\forall \lambda \in \Lambda_s$;
19 **return** top_k($\Lambda_s, \boldsymbol{p}, n$)

confidence bound (UCB) strategy since maximizing the negative validation loss $-\mathcal{L}$ is equivalent to minimizing $\mathcal{L}$. The UCB approach trades off exploration and exploitation as it also considers the uncertainty for a specific hyperparameter configuration. The score p_λ is thus obtained from $\boldsymbol{\theta}^\top \lambda + \alpha\sqrt{\lambda^\top A^{-1}\lambda}$, where $A = \boldsymbol{X}^\top \boldsymbol{X} + \gamma \mathbf{I}$ is the regularized design matrix of the configurations with $\gamma \geq 0$ which have been evaluated so far and $\alpha > 0$ is a trade-off parameter.

Algorithm 2 summarizes our approach for HyperUCB. In this algorithm, the bandit model is learned in line 13, and together with the covariance matrix it computes the upper confidence values in two sampling steps. At every iteration, a number of n_0 configurations are randomly sampled as in HB, and from those, the bandit model selects the n most promising ones. The next sampling step is at line 14, where `top_ucb` is performed on the values of p_λ rather than y_λ. Note that the matrix A is updated every time a configuration is chosen, even within an iteration, which leads to a tighter confidence interval for those configurations.

4 Empirical Study

In this section, we evaluate the performance of the HyperUCB strategy compared to Hyperband [8]. The experiments are conducted on the MNIST data which consists of 60,000 training and 10,000 test instances. As a model, we use a simple multi-layer perceptron (MLP) which learns to classify images of handwritten

Hyperparameter	Range	Type
learning rate	$[0.0001, 1]$	float
# hidden layers	$\{1, 2, 3, 4, 5\}$	integer
# neurons	$\{16, 32, \ldots, 512\}$	integer
activation	{relu, tanh, sigmoid}	categorical

Table 1. Hyperparameters of the multi-layer perceptron.

digits. We use the categorical cross entropy as a loss function and the RMSprop optimizer. The validation loss is computed on the hold-out data. Within the MLP we use four hyperparameters which are outlined in Table 1. We determine a minimum budget of one unit of resource which corresponds to 100 mini-batches of size 100. The maximum budget consists of R units of resources, hence $100R$ mini-batches. We use the default value of $\eta = 3$ as specified in Hyperband. Our approach contains two additional parameters: the exploration-exploitation trade-off α and a regularization-weight γ in ridge regression. We select the values of $\alpha = 0.4$ as it gives best performance in [7] and the regularization is set to $\gamma = 0.1$.

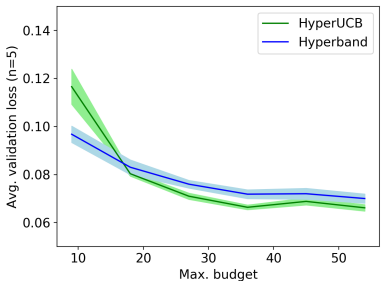

Fig. 1. Performance w.r.t. the budget.

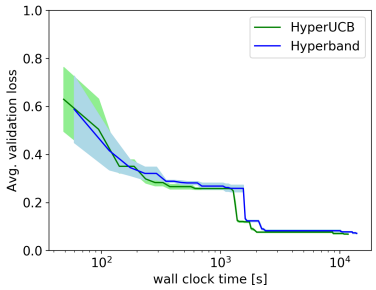

Fig. 2. Performance w.r.t. the time.

Figure 1 shows the validation loss averaged over five independent runs for various maximum budgets including standard errors. With a max. budget higher than 19, HyperUCB outperforms Hyperband as it consistently yields lower validation errors. We credit this finding to the fact that using a higher budget, more rounds are conducted on which the bandit model can learn to discriminate promising from unpromising hyperparameter configurations. This can be hardly done with lower max. budgets due to the lack of training data.

Figure 2 depicts the average validation loss in dependence of computational time, measured in seconds, for a budget of 45. It can be seen that HyperUCB performs on par with Hyperband, meaning it is as fast or faster than Hyperband.

5 Conclusion and Future Work

In this paper, we presented HyperUCB, a contextual extension using a UCB strategy for Hyperband, which is a bandit-based method for hyperparameter optimization. The idea was as follows: Instead of sampling n iid hyperparameter configurations in each round for evaluation, we sampled more configurations, assessed them using a multi-armed bandit with a UCB strategy and only evaluated the n best configurations. This way, we guided the sampling procedure towards more promising configurations and avoided evaluating hyperparameters which are already assumed to yield a high validation error. An experiment on the MNIST data showed that it outperforms the Hyperband baseline for moderate budgets at optimizing several hyperparameters of a multi-layer perceptron.

Further work will utilize the ideas from Tavakol & Brefeld [12], in which the parameters of the bandit model can be learned using kernel methods in the dual space to capture non-linearity. We also plan on extending the experimental setup by adding more baselines, e.g., BO-HB [4] as well as considering multiple hyperparameter optimization scenarios on various data sets and models.

References

1. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Machine learning **47**(2-3) (2002)
2. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of Machine Learning Research **13**(Feb) (2012)
3. Bergstra, J.S., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: Advances in neural information processing systems (2011)
4. Falkner, S., Klein, A., Hutter, F.: BOHB: Robust and efficient hyperparameter optimization at scale. In: International Conference on Machine Learning (2018)
5. Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: International Conference on Learning and Intelligent Optimization (2011)
6. Jamieson, K., Talwalkar, A.: Non-stochastic best arm identification and hyperparameter optimization. In: Artificial Intelligence and Statistics (2016)
7. Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th international conference on World wide web (2010)
8. Li, L., Jamieson, K.: Hyperband: A novel bandit-based approach to hyperparameter optimization. Journal of Machine Learning Research **18** (2018)
9. Shang, X., Kaufmann, E., Valko, M.: A simple dynamic bandit algorithm for hyperparameter tuning. In: Workshop on Automated Machine Learning at International Conference on Machine Learning (2019)
10. Snoek, J., Larochelle, H., Adams, R.P.: Practical bayesian optimization of machine learning algorithms. In: Advances in neural information processing systems (2012)
11. Sparks, E.R., Talwalkar, A., Haas, D., Franklin, M.J., Jordan, M.I., Kraska, T.: Automating model search for large scale machine learning. In: Proceedings of the Sixth ACM Symposium on Cloud Computing (2015)
12. Tavakol, M., Brefeld, U.: A unified contextual bandit framework for long-and short-term recommendations. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases (2017)