MULTI-TASK MULTICRITERIA HYPERPARAMETER OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a new method for searching optimal hyperparameters among several tasks and several criteria. Multi-Task Multi Criteria method (MTMC) provides several Pareto-optimal solutions, among which one solution is selected with given criteria significance coefficients. The article begins with a mathematical formulation of the problem of choosing optimal hyperparameters. Then, the steps of the MTMC method that solves this problem are described. The proposed method is evaluated on the image classification problem using a convolutional neural network. The article presents optimal hyperparameters for various criteria significance coefficients.

1 INTRODUCTION

Hyperparameter optimization (Hutter et al., 2009) is an important component in the implementation of machine learning models (for example, logistic regression, neural networks, SVM, gradient boosting, etc.) in solving various tasks, such as classification, regression, ranking, etc. The problem is how to choose the optimal parameters when a trained model is evaluated using several sets and several criteria.

This article describes a method to solving the above problem. We will present the results of experiments on the selection of hyperparameters obtained using the proposed approach (MTMC) with various criteria significance coefficients.

The article is organized as follows. First, we discuss related work in Section 2. Section 3 describes the proposed method. Section 4 presents the results of experiments on the selection of optimal hyperparameters. Section 5 contains the conclusion and future work.

2 RELATED WORK

The problem of choosing optimal hyperparameters has long been known. Existing methods for solving this problem give both the single optimal solution, and several ones.

In (Sener & Koltun, 2018), a Pareto optimization method is proposed, in which the optimal solution is given for several problems simultaneously. This method consists in minimizing the weighted sum of loss functions for each task. (Fliege & Svaiter, 2000) describes the Pareto optimization method, which gives an optimal solution according to several criteria based on gradient descent, and this optimization is also carried out in the learning process. In (Igel, 2005), the search for a Pareto-optimal solution is carried out according to several criteria. The method in (Bengio, 2000) gives optimal hyperparameters using back propagation through the Cholesky decomposition. In (Bergstra et al., 2011), optimization is performed using a random choice of hyperparameters based on the expected improvement criterion. (Bergstra & Bengio, 2012) proposes method of hyperparametric optimization based on random search in the space of hyperparameters. In (Snoek et al., 2012), search for optimal hyperparameters is carried out using Bayesian optimization.

The novelty of MTMC method:

1. Optimization is carried out simultaneously according to several criteria and several tasks with setting the significance of the criteria.

2. The choice of optimal hyperparameters is provided *after* training and evaluation, which eliminates the need to re-train the model.

3. The proposed method does not need to be trained.

3 THE PROPOSED METHOD

First, we describe the mathematical problem that MTMC solves, then we present the steps performed in MTMC.

3.1 FORMALIZATION OF THE PROBLEM

In the proposed method, the model is evaluated on several test sets (tasks) T. The problem of finding a minimum for tasks T is known as minimizing the expected value of empirical risk (Vapnik, 1992).

The choosing optimal hyperparameters is formalized as follows:

$$\theta = \underset{\theta \in \Theta}{\operatorname{argmin}} \mathbb{E}_{\tau}[\mathcal{L}(\theta, \phi)] \tag{1}$$

where Θ is the set of all hyperparameters, θ is the selected optimal hyperparameters, ϕ is the vector of significance coefficients of the criteria, $\mathcal{L}(\cdot)$ is the estimation function of the model with the given hyperparameters θ and the coefficients ϕ, τ is the task for which optimization is performed.

The developed method gives a solution to the problem (1).

3.2 DESCRIPTION OF MTMC

According to (1), the developed method should fulfill the following requirements:

1) the method should solve the minimization problem;

2) the significance of each criterion is determined by the vector of coefficients ϕ (the higher the coefficient, the more important the corresponding criterion).

We denote the test sample of the task τ:

$$x^i \sim \mathcal{D}, i = 1 \dots N_{\text{task}} \tag{2}$$

where x^i is the ith test set has the distribution $\mathcal{D}$, N_{task} is the number of tasks.

Before choosing hyperparameters, for model $\mathcal{M}$ we obtain an evaluation matrix for the test set x^i and the given evaluation criteria:

$$V = \mathcal{M}\left(x^i; \Theta\right) \tag{3}$$

$$\mathcal{M}\left(x^i; \Theta\right) :: \left(\mathbb{R}^{x_{\text{sise}}}, \mathbb{R}^{N_{\text{parameter}} \times N_{\text{combination}}}\right) \rightarrow \mathbb{R}^{N_{\text{combination}} \times N_{\text{criteria}}} \tag{4}$$

where $\mathcal{M}(\cdot)$ is the model function that transforms the given set x^i and with the given hyperparameters Θ into the evaluation matrix V, N_{criteria} is the number of criteria, x_{size} is the dimension of the test set, $N_{\text{parameter}}$ is the number of hyperparameters, $N_{\text{combination}}$ is the number of hyperparameter combinations.

Then, the function $\mathcal{L}$ is calculated for each set x^i, which is formally described as follows:

$$\mathcal{L}(\cdot; \mathbf{\Theta}, \phi) = \mathcal{E}(V; \phi) \tag{5}$$

$$\mathcal{E}(V; \phi) :: \left(\mathbb{R}^{N_{\text{criteria}}}, \mathbb{R}^{N_{\text{criteria}}}\right) \rightarrow \mathbb{R}^1 \tag{6}$$

MTMC method gives Pareto optimal solutions in which the following steps are performed:

1. The vectors from the evaluation V (the number of such vectors is N_{criteria}) is in the space of given criteria.

2. Then we get Pareto optimal solutions $\tilde{V} \subseteq V$ the nearest Pareto front to the origin of the criteria space:

$$\tilde{V} = \text{ ParetoFront }(V), \quad \tilde{V} \in \mathbb{R}^{N_{\text{opt}} \times N_{\text{criteria}}} \tag{7}$$

where N_{opt} is the number of Pareto optimal solutions.

3. The optimal solutions $\tilde{v} \in \tilde{V}$ are scaled according to each criterion to the interval $[0; 1]$:

$$\tilde{V}_{\text{scaled}} = \frac{\tilde{V}_i - \tilde{v}_{\text{min}}}{\tilde{v}_{\text{max}} - \tilde{v}_{\text{min}}}, \tilde{v}_{\text{min}} \in \mathbb{R}^{N_{\text{criteria}}}, \tilde{v}_{\text{min}} \in \mathbb{R}^{N_{\text{criteria}}}, i = 1 \dots N_{opt} \tag{8}$$

where $\tilde{v}_{max}$ is the vector of maximum values of $\tilde{V}$ for each criterion, $\tilde{v}_{min}$ is the vector of minimum values of V for each criterion.

Thus, the optimal solution is the solution closest to the origin, and if any solution $\tilde{v} \in \tilde{V}$ is the origin, then it is optimal for any ϕ.

4. The vector ϕ in the space of criteria is defined.

We introduce the vector of the optimal solution, which is the middle of the segment $[0; 1]$ in the axes of the criteria space:

$$\phi_{opt} = \left(\forall i : \phi_0 = \dots = \phi_i = \dots = \phi_{N_{criteria}} = \frac{1}{2} \right). \tag{9}$$

Conditions for ϕ are:

$$\phi = \begin{cases} \phi_{\text{opt}}, \text{ if } \forall i : \phi_i = 0, \\ \phi \in [0; 1], \text{ otherwise.} \end{cases} \tag{10}$$

5. Project the vectors from the matrix $\tilde{V}_{scaled}$ onto the vector ϕ:

$$\tilde{V}_{proj} = \frac{\tilde{V}^T_{scaled} \cdot \phi}{\|\phi\|}, \quad \tilde{V} \in \mathbb{R}^{N_{opt}}. \tag{11}$$

From (9) and (11) it follows that if the vectors ϕ and ϕ_{opt} are collinear, then:

$$\exists \lambda : \phi = \lambda \cdot \phi_{opt} \Rightarrow \tilde{V}_{proj} = \sum_i \left[\frac{1}{\phi_i} \cdot \frac{\tilde{V}_{scaled_i}}{\|\phi\|} \right] \propto \sum_i \tilde{V}_{scaled_i} = \left\| \tilde{V}_{scaled} \right\|_1. \tag{12}$$

That is, in the case of equality of all elements of ϕ, the minimization problem reduces to finding the minimum $L1$-norm $\tilde{V}_{scaled}$.

From (11) it also follows that if some component of the vector ϕ is equal to zero, then the corresponding criterion will not affect the choice of the optimal hyperparameter. If all criteria are equal to zero, except for one, then only the criterion with a nonzero component of the vector ϕ will affect the choice of optimal hyperparameters.

6. We find hyperparameters θ at which the minimum of the vector $\tilde{V}_{proj}$ is reached:

$$\theta = \underset{\theta}{\text{argmin}} \tilde{V}_{\text{proj}}. \tag{13}$$

Figure 1 shows an example solution using MTMC for random numbers in the three-dimensional space.

4 Conducting Experiments

First, the evaluation matrix V for the selected model $\mathcal{M}$ is obtained. Then, for various combinations of components of ϕ, optimal hyperparameters are selected using MTMC.

4.1 Obtaining The Evaluation Matrix

The developed MTMC method is applied to solve the problem of image classification. The problem we are solving is described in the article (Akhmetzyanov & Yuzhakov, 2019).

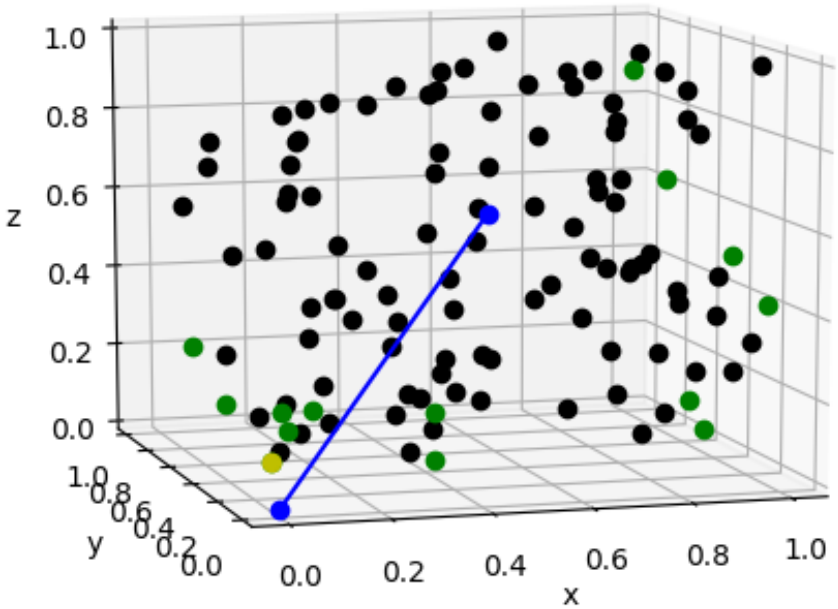

Figure 1: Example of a solution given by MTMC, green points denote Pareto optimal solutions, blue vector is the vector ϕ, yellow point denotes the optimal solution given by MTMC for a given ϕ.

In (Akhmetzyanov & Yuzhakov, 2018), we selected the MobileNet neural network architecture (Howard et al., 2017) as a mathematical model for image processing.

The search for optimal hyperparameters was carried out among two popular training methods: changing the learning speed based on the epoch $lr = base_lr \cdot lr_decay^{epoch}$ (where $base_lr$ is the initial learning rate, lr_decay is the coefficient of decreasing learning rate, $epoch$ is the number of epochs) and cyclical learning (Smith, 2017). In cyclic learning, there are three ways to change the learning rate:

1. *triangular* is fixed initial learning rate (*base_lr*), maximum fixed learning rate (*max_lr*), learning rate increases from *base_lr* to *max_lr* and decreases from *max_lr* to *base_lr* linearly.

2. *triangular2* is fixed initial learning rate (*base_lr*), maximum learning rate (*max_lr*), learning rate, as in triangular, varies linearly, but *max_lr* in the learning process is halved.

3. *exp_range* is fixed initial learning rate (*base_lr*), maximum learning rate (*max_lr*), learning rate also changes linearly, but *max_lr* in the learning process decreases exponentially.

In the first learning method, the hyperparameters are the value of the initial learning rate (*base_lr*) and the coefficient of decreasing learning rate (*lr_decay*). In the second method, hyperparameters a way to change the learning rate (*cyclic_mode*), the value of the initial learning rate (*base_lr*) and maximum learning rate (*max_lr*).

For each hyperparameter, a range of change and a constant step of change within the range were selected. For training, Grid search was used among $N_{combination} = 100$ combinations of hyperparameters.

For each combination of hyperparameters, training was carried out using cross-validation k-fold (Stone, 1974) with 10 folds. For training, Keras (Chollet et al., 2015) and TensorFlow (Abadi et al., 2015) were used. The training lasted 15 epochs; the test was carried out on $N_{task} = 5$ different test sets. That is, $100 \cdot 10 = 1000$ is number of different neural networks, $15 \cdot 1000 = 15000$ neural networks evaluations are conducted, $15000 \cdot 5 = 75000$ evaluation results are obtained. Neural networks trained on ten TPUs v2, which took several days.

Among all epochs, for each fold and for each test set, the maximum accuracy is selected, as well as the number of the epoch at which the maximum accuracy is achieved. The following values are

calculated for each test set among the folds: the expected value and the variance of the classification error, the expected value and the variance of the epoch number at which convergence on the test set is achieved. We have obtained an evaluation matrix among all neural networks with their hyperparameters and among all test samples.

4.2 Processing The Evaluation Matrix

Based on (1), for each criterion, among all the samples, the expected value is considered. That is, for all test sets, the criteria: (i) the sample mean of the classification error, (ii) the sample variance of the classification error, (iii) the sample mean and (iv) sample variance of the epoch number at which convergence is achieved in the test sample. These values are the criteria for evaluating hyperparameters for a certain test set (matrix V from (3)) with the number of criteria $N_{criteria} = 4$.

$\tilde{V}$ is calculated from (7), the number of Pareto optimal solutions obtained is $N_{opt} = 25$. Optimal hyperparameters, i.e., $\tilde{V}$, are presented in Appendix A.

The vector of the optimal solution according to (9) for $N_{criteria} = 4$ is $\phi_{opt} = \{0.5; 0.5; 0.5; 0.5\}$. Next, calculations are carried out according to (8) and (11), and for various ϕ optimal solutions are chosen according to (5). These optimal solutions are presented in Appendix B.

5 Conclusion

In this work, we proposed a new method for hyperparameter optimization among several tasks and several criteria. We trained several neural networks with various hyperparameters to solve the image classification problem. Then, for these neural networks, evaluation matrices were obtained on several tasks. We applied MTMC to these matrices and got optimal solutions with different significance coefficients. In the future, we will work to create a meta-learning method that solves the same problem as the method described in this article, but optimization will be performed among various models.

Acknowledgments

The reported study was partially supported by the Government of Perm Krai, research project No. C-26/174.6.

References

Martín Abadi, Ashish Agarwal, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `http://tensorflow.org/`. Software available from tensorflow.org.

Kirill Akhmetzyanov and Alexander Yuzhakov. Convolutional neural networks comparison for waste sorting tasks. *Izvestiya SPbGETU "LETI"*, (6):27, 2018.

Kirill Akhmetzyanov and Alexander Yuzhakov. Waste sorting neural network architecture optimization. In *2019 International Russian Automation Conference (RusAutoCon)*, pp. 1–5. IEEE, 2019.

Yoshua Bengio. Gradient-based optimization of hyperparameters. *Neural computation*, 12(8):1889–1900, 2000.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, pp. 2546–2554, 2011.

François Chollet et al. Keras. `https://keras.io`, 2015.

Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

Table 1: Pareto optimal solutions for the first learning method

base_lr	**lr_decay**
0.001	0.75
0.001	0.8
0.005	0.75
0.01	0.9
0.01	0.95

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Frank Hutter, Holger H Hoos, Kevin Leyton-Brown, and Thomas Stützle. Paramils: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36:267–306, 2009.

Christian Igel. Multi-objective model selection for support vector machines. In *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 534–546. Springer, 2005.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, pp. 527–538, 2018.

Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 464–472. IEEE, 2017.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pp. 2951–2959, 2012.

Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133, 1974.

Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pp. 831–838, 1992.

A Pareto Optimal Hyperparameters

Table 1 and Table 2 show the Pareto optimal hyperparameters for the two learning methods.

Table 2: Pareto optimal solutions for the second learning method

base_lr	**max_lr**	**cyclic_mode**
0.0001	0.005	exp_range
0.0001	0.005	triangular2
0.0005	0.001	exp_range
0.0005	0.005	triangular2
0.001	0.0001	triangular2
0.001	0.0005	triangular
0.001	0.001	exp_range
0.001	0.005	triangular2
0.005	0.0001	triangular
0.005	0.005	triangular
0.01	0.0001	triangular
0.01	0.0001	triangular2
0.01	0.005	triangular
0.01	0.005	triangular2
0.01	0.01	triangular
0.0001	0.0001	triangular
0.0005	0.001	triangular
0.0005	0.01	triangular2
0.001	0.005	triangular
0.005	0.01	triangular

B MTMC Optimal Hyperparameters

Table 3 shows the optimal hyperparameters θ obtained using MTMC method for given significance coefficients of the criteria ϕ.

Table 3: Optimal hyperparameters for relevant criteria significance coefficients

ϕ_0	ϕ_1	ϕ_2	ϕ_3	θ
0.5	0.5	0.5	0.5	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
0.0	0.5	0.5	0.5	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
1.0	0.5	0.5	0.5	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
0.5	0.0	0.5	0.5	base_lr=0.005, max_lr=0.0001, cyclic_mode=triangular
0.5	1.0	0.5	0.5	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
0.5	0.5	0.0	0.5	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
0.5	0.5	1.0	0.5	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
0.5	0.5	0.5	0.0	base_lr=0.0001, max_lr=0.005, cyclic_mode=triangular2
0.5	0.5	0.5	1.0	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
0.0	0.0	0.5	0.5	max_lr=0.005, lr_decay=0.75
1.0	1.0	0.5	0.5	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
0.5	0.5	0.0	0.0	base_lr=0.0001, max_lr=0.005, cyclic_mode=triangular2
0.5	0.5	1.0	1.0	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
1.0	0.0	0.0	0.0	base_lr=0.0001, max_lr=0.005, cyclic_mode=triangular2
0.0	1.0	0.0	0.0	base_lr=0.0001, max_lr=0.005, cyclic_mode=triangular2
0.0	0.0	1.0	0.0	max_lr=0.005, lr_decay=0.75
0.0	0.0	0.0	1.0	base_lr=0.0005, max_lr=0.001, cyclic_mode=exp_range