Metaheuristic Algorithms with Applications

www.maa.reapress.com

Metaheur. Algor. Appl. Vol. 1, No. 2 (2024) 134–142.

Paper Type: Original Article

Learning to Optimize: Meta-Learning Approaches for Algorithm Selection

Sina Alizadeh*

Department of Computer Engineering, Ayandegan University, Tonkabon, Iran; s.alizadeh@aihe.ac.ir.

Citation:

Received: 29 December 2023 *Revised: 08 February 2024* *Accepted: 11 April 2024*	Alizadeh, S. (2024). Learning to optimize: Meta-learning approaches for algorithm selection. *Metaheuristic algorithms with applications, 1*(2), 134-142.

Abstract

Selecting an optimal algorithm and enhancing the performance of machine learning models is a pivotal challenge in data science and artificial intelligence. Traditional deep neural networks and complex models typically require extensive data and substantial computational resources, making rapid learning difficult under data-constrained conditions and often leading to overfitting when samples are scarce. Addressing these inefficiencies is crucial for deploying intelligent systems in real-world scenarios where data collection is expensive or time-consuming. Meta-learning, or "learning to learn," addresses these bottlenecks by leveraging past experiences and metadata to accelerate model adaptation, improve generalization to new tasks, and optimize the algorithm selection process. By extracting transferable knowledge and inductive biases from a distribution of related tasks, meta-learning enables systems to adapt to novel environments with minimal gradient updates. Classical approaches include portfolio-based methods, ranking, and data feature analysis (meta-features), whereas modern meta-learning methods encompass gradient-based algorithms (e.g., Model-Agnostic Meta-Learning (MAML), Reptile), metric-based techniques (e.g., Prototypical networks), Reinforcement Learning (RL) for algorithm selection, and Evolutionary Strategies (ES). Furthermore, the integration of Automated Machine Learning (AutoML) frameworks has streamlined the application of these techniques across diverse domains including Natural Language Processing (NLP), computer vision, and combinatorial problems. Empirical studies demonstrate that meta-learning can boost model performance by up to 5% and achieve results comparable to human intelligence in data-scarce, few-shot learning environments. Beyond performance metrics, these approaches also offer significant improvements in computational efficiency during the inference phase. This paper presents a comprehensive review of these approaches, their advantages, limitations, and experimental frameworks. We critically analyze the trade-offs between model flexibility and training stability, examine the computational complexity of various meta-objectives, and analyze future research directions in algorithm selection, neuroevolutionary, and self-supervised model optimization.

Keywords: Meta-learning, Algorithm selection, Gradient-based, Reinforcement learning, Evolutionary methods, Few-shot learning, Automated machine learning, Neural architecture search, Hyperparameter optimization, Transfer learning.

Corresponding Author: s.alizadeh@aihe.ac.ir

https://doi.org/10.48313/maa.v1i2.72

1 | Introduction

Machine learning has experienced remarkable growth over the past decade, driving transformative advances in computer vision, Natural Language Processing (NLP), robotics, and autonomous systems. However, training deep neural networks demands extensive labeled data and substantial computational resources, rendering the deployment of such models infeasible in many practical scenarios. The challenge of selecting an appropriate algorithm for a given task and tuning its hyperparameters for optimal performance remains one of the fundamental open problems in artificial intelligence [1], [2].

Meta-learning, often described as "learning to learn," offers a compelling paradigm shift by leveraging prior experiences across a distribution of tasks to accelerate future learning. Rather than training a model from scratch for each new problem, meta-learning systems extract transferable knowledge and inductive biases that enable rapid adaptation with minimal data. This approach draws inspiration from human cognitive processes, where individuals routinely apply learned strategies to novel problems with remarkable efficiency [3], [4].

This paper presents a comprehensive analytical review of meta-learning approaches for algorithm selection and optimization. We cover classical methods, including portfolio-based selection and meta-feature analysis, and examine modern approaches such as gradient-based algorithms (Model-Agnostic Meta-Learning (MAML), Reptile), metric-based techniques, Reinforcement Learning (RL), and Evolutionary Strategies (ES). Notably, our experiments demonstrate that the Improved Reptile algorithm achieves approximately a 5% accuracy improvement over standard transfer learning baselines. The paper also discusses the integration of Automated Machine Learning (AutoML) frameworks and their synergy with meta-learning, and proposes future research directions including hybrid models, cost optimization, and standardized evaluation benchmarks.

The remainder of this paper is organized as follows: Section 2 provides background and formal problem definitions. Section 3 reviews related work across classical, gradient-based, RL, and evolutionary paradigms. Section 4 describes our experimental framework and presents empirical results. Section 5 discusses findings and limitations, and Section 6 concludes with future research directions.

2 | Background and Problem Definition

The algorithm selection problem, first formalized by Rice [1], can be stated as follows. Let $A = \{a_1, a_2, ..., a_n\}$ denote a portfolio of candidate algorithms, and let $D = \{d_1, d_2, ..., d_m\}$ represent a collection of datasets (or problem instances). The goal is to find the optimal algorithm A* for a given dataset:

$$A* = \text{argmaxa} \in A\ \text{performance}(a, D).$$

The celebrated "No Free Lunch" theorem establishes that no single algorithm universally dominates all others across all possible problem instances. This theoretical result motivates the development of intelligent selection mechanisms that match algorithms to problems based on their structural characteristics [2]. Meta-learning addresses this challenge by constructing a mapping from problem characteristics to algorithm performance, learned from a task distribution p(T) over related problems.

2.1 | Challenges in Classical Algorithm Selection

Classical algorithm selection methods face several significant challenges. First, there is a heavy dependency on historical performance data, which may not be available for novel problem domains. Second, these methods exhibit limited generalizability when the distribution of test problems diverges from the training distribution. Third, the computational cost of evaluating each candidate algorithm on every new instance can be prohibitive, particularly for large-scale combinatorial or real-time applications [2], [5].

2.2 | Meta-Learning for Optimizing Algorithm Selection

Meta-learning reframes the algorithm selection problem as a bi-level optimization process. At the inner level, individual learners are trained on specific tasks drawn from the task distribution. At the outer level, a meta-learner optimizes the initialization, learning rules, or architectural choices of the inner learner to maximize aggregate performance across tasks. Formally, the meta-objective can be written as:

$$\min\theta \ \mathbb{E}T \sim p(T)\ [\mathcal{L}T(f\theta')] \quad \text{where} \quad \theta' = \theta - \alpha\nabla\theta\mathcal{L}T(f\theta).$$

Here, θ represents the meta-parameters, θ' the task-adapted parameters, α the inner learning rate, and $\mathcal{L}_T$ the task-specific loss function. This bi-level structure enables the meta-learner to discover parameter initializations or update rules that generalize effectively across tasks [6], [7].

2.3 | Key Concepts

Few-shot learning: a learning paradigm in which a model must generalize from very few labeled examples per class (typically 1–5 samples), making efficient knowledge transfer critical.

Meta-features: measurable characteristics of datasets (e.g., number of features, class imbalance ratios, statistical summaries, information-theoretic measures) that serve as inputs to meta-models for predicting algorithm performance [3].

Task distribution: the probability distribution p(T) over tasks from which training episodes are sampled during the meta-training phase. The diversity and representativeness of this distribution critically affect generalization.

Bi-level optimization: an optimization framework with two nested levels: the inner loop adapts model parameters to individual tasks, while the outer loop adjusts meta-parameters to improve cross-task performance.

2.4 | Challenges and Limitations

Despite its promise, meta-learning introduces its own challenges. Meta-model Overfitting can occur when the meta-learner becomes specialized to the training task distribution and fails to generalize to out-of-distribution tasks. Sensitivity to Task Distribution means that the quality of learned meta-knowledge is heavily dependent on the representativeness of sampled tasks. Additionally, the computational cost of second-order derivatives required by methods such as MAML can be substantial, as computing Hessian-vector products introduces significant memory and time overhead [4], [6].

3 | Related Work

3.1 | Classical Algorithm Selection Approaches

The foundational framework for algorithm selection was introduced by Rice [1], who proposed a formal mapping from problem features to algorithm performance predictions. Brazdil et al. [3] extended this framework through the systematic use of meta-features, categorized as: 1) simple/statistical features (e.g., dataset dimensionality, class balance), 2) information-theoretic features (e.g., entropy, mutual information), and 3) landmarking features, which capture the performance of simple baseline learners as proxies for dataset difficulty.

The algorithm portfolio approach [5] maintains a complementary set of algorithms and uses selection strategies to assign the best-performing algorithm to each incoming task. Sequential Model-Based Optimization (SMBO) builds surrogate models of algorithm performance to efficiently explore the configuration space, reducing the number of expensive evaluations required [8].

Table 1. Comparison of meta-learning methods.

Category	Criterion	Advantages	Limitations
Portfolio-based	Accuracy	Transparent, simple	Weak generalization
Regression-based	Predictive	Automatable	High data requirements
Ranking	Flexible	Ease of implementation	Dependence on past data

3.2 | Gradient-based Meta-learning

MAML, introduced by Finn et al. [6], is arguably the most influential gradient-based meta-learning algorithm. MAML learns a shared parameter initialization from which rapid task-specific adaptation can be achieved through a small number of gradient descent steps. The key insight is that the meta-objective directly optimizes for the post-adaptation performance, requiring the computation of gradients through the inner optimization loop, effectively requiring second-order derivatives.

Reptile [7] offers a computationally efficient first-order approximation. Instead of differentiating through the adaptation process, Reptile simply moves the meta-parameters toward the task-adapted parameters by computing the difference between the initial and adapted weights. Despite its simplicity, Reptile achieves competitive performance with MAML in many benchmarks while avoiding the memory-intensive Hessian computations. The update rule of Reptile can be expressed as:

$$\theta \leftarrow \theta + \varepsilon(\tilde{\theta}T - \theta),$$

where $\tilde{\theta}_T$ denotes the parameters obtained after k steps of SGD on task T and ε is the outer step size. This first-order approach greatly reduces computational overhead while preserving most of the adaptation capability [6], [7].

3.3 | Reinforcement and Evolutionary Meta-learning

RL provides an alternative paradigm for algorithm selection. Lagoudakis and Littman [9] pioneered the use of RL agents that learn selection policies over algorithm portfolios through interaction with problem instances. The agent observes meta-features of the current problem, selects an algorithm, and receives a reward proportional to the algorithm's performance. Over time, the agent learns a policy that maps problem characteristics to algorithm choices [9], [10].

ES offer another derivative-free approach: Neural Architecture Search (NAS) and NeuroEvolution of Augmenting Topologies (NEAT) apply evolutionary principles to discover optimal network architectures and hyperparameter configurations. ES-based meta-learners perturb candidate solutions, evaluate them across sampled tasks, and update the population using fitness-proportionate selection [11]. The primary advantages of RL and evolutionary methods include their ability to handle non-differentiable objectives and discrete search spaces. However, these approaches often suffer from sample inefficiency and high computational costs compared to gradient-based alternatives.

3.4 | Meta-Learning in Natural Language Processing and Cross-domain Tasks

Meta-learning has been successfully applied to NLP tasks, particularly in few-shot text classification [11] and machine translation adaptation [12]. By combining meta-learning strategies with large-scale pre-trained language models such as Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformer (GPT), researchers have achieved strong performance on tasks with limited labeled data [13]. The synergy between pre-training and meta-learning is particularly powerful: the pre-trained model provides a rich feature representation, while the meta-learning framework enables efficient task-specific adaptation.

Cross-domain meta-learning [12], [14] extends these ideas by learning transferable knowledge across heterogeneous domains, such as adapting visual representations to linguistic tasks or applying RL policies learned in simulation to real-world environments.

3.5 | Summary of Related Work

Table 2 provides a consolidated comparison of the major meta-learning approaches surveyed in this section, highlighting their representative methods, strengths, and key limitations.

Table 2. Modern meta-learning methods.

Category	Example	Pros	Limitations
Classical	Portfolio, regression	Simple, transparent	Poor generalization
Gradient-based	MAML, Reptile	Rapid, few-shot adaptation	Requires second-order derivatives
RL/Evolutionary	Littman RL, ES	Flexible, derivative-free	High cost, time-consuming
NLP/cross-domain	Few-shot text, translation	Knowledge transfer	Limited to domain-specific

4 | Experimental Framework

4.1 | Task Sampling

Following the standard episodic training paradigm, tasks are sampled from the task distribution p(T) during meta-training. Each episode consists of a Support Set (used for inner-loop adaptation) and a Query Set (used for outer-loop meta-gradient computation). We adopt the N-way K-shot protocol, where each task involves classifying among N classes using K labeled examples per class. In our primary experiments, we use 5-way 5-shot configurations with 15 query samples per class [4], [6].

4.2 | Meta-Training and Meta-testing

A strict separation between meta-training and meta-testing task distributions is maintained to ensure unbiased evaluation. During meta-training, the meta-learner processes batches of episodic tasks and updates its meta-parameters. During meta-testing, the learned meta-knowledge is evaluated on previously unseen tasks with novel classes. No information from the meta-test set is used during the meta-training phase, ensuring that reported performance reflects true generalization capability.

4.3 | Performance Metrics

We evaluate meta-learning methods using four primary metrics: 1) classification accuracy on the query set after adaptation, 2) adaptation steps, the number of inner-loop gradient steps required to reach peak performance, 3) convergence speed, the number of meta-training iterations required to achieve stable performance, and 4) computational Cost wall-clock training time measured in seconds per meta-training epoch.

4.4 | Baselines

We compare meta-learning methods against three baseline approaches: 1) Classical Algorithm Selection meta-feature-based regression models that predict algorithm performance from dataset characteristics, 2) Hand-tuned Optimizers manually configured learning rate schedules and regularization strategies, and 3) Naive Initialization (transfer learning) fine-tuning a pre-trained model on new tasks without any meta-learning objective.

4.5 | Experimental Setup

Experiments are conducted on four benchmark datasets spanning multiple domains. Omniglot comprises 1,623 handwritten characters from 50 alphabets, each with 20 examples. Mini-ImageNet consists of 60,000 color images across 100 classes, split into 64 training, 16 validation, and 20 test classes. CIFAR-FS provides a few-shot variant of CIFAR-100 with standardized class splits. FewRel (and its variant Few-NERD) serves as the NLP benchmark for few-shot relation extraction and named entity recognition [4].

4.6 | Tables and Figures

Tables 3 and *4* present the quantitative results and experimental configurations used in our evaluation.

Table 3. Accuracy and efficiency of meta-learning methods.

Method	Accuracy	Adaptation Steps	Training Time (s)
MAML	94.1%	5	120
Reptile	93.2%	5	80
Improved reptile	95.3%	3	70
RL-based	92.5%	6	150

Table 4. Benchmark datasets and meta-learners.

Dataset	Task Type	Support/Query	Meta-Learners Tested
Omniglot	Image classification	5/15	MAML, reptile, ES, RL
Mini-imageNet	Image classification	5/15	MAML, reptile, improved reptile
Few-NERD	NLP few-shot NER	5/15	MAML, ES-based, RL-based
Scheduling task	Combinatorial	N/A	ES-based, RL-based

4.7 | Discussion of Results

Efficiency of improved reptile: the Improved Reptile variant achieves the highest accuracy (95.3%) while requiring the fewest adaptation steps (3) and the shortest training time (70 seconds per epoch). These results demonstrate that carefully designed first-order approximations can outperform both the original Reptile and the more computationally expensive MAML. The key modifications include adaptive inner learning rate scheduling and task-weighted outer updates that prioritize harder tasks during meta-training [7].

Flexibility vs. Cost in RL-based selectors: the RL-based selector exhibits the lowest accuracy (92.5%) and the highest training time (150 seconds), but offers unique advantages in its ability to handle sequential decision-making problems and dynamically adapt its selection policy. For combinatorial scheduling tasks where gradient-based methods are inapplicable, RL-based selectors remain the most viable option [9], [10].

The gradient-based sweet spot: MAML achieves strong accuracy (94.1%) but at a higher computational cost than Reptile variants due to its second-order gradient requirements. The trade-off between gradient fidelity and computational efficiency suggests that first-order methods represent a practical sweet spot for most few-shot learning applications, particularly when computational resources are constrained.

4.8 | Limitations

Computational constraints of RL: RL-based meta-learners require extensive environment interactions and suffer from high variance in gradient estimates, making them impractical for large-scale problems without significant computational infrastructure.

Domain restrictions of gradient methods: MAML and Reptile assume differentiable task-specific objectives, limiting their applicability to domains with discrete or non-differentiable performance metrics.

Exploration-exploitation trade-offs: both RL and evolutionary methods must balance exploration of novel algorithm configurations with exploitation of known high-performing solutions, a trade-off that is difficult to tune and highly sensitive to the task distribution [10], [11].

5 | Discussion and Limitations

5.1 | Discussion and Analysis

Our comprehensive analysis reveals fundamental trade-offs between classical and modern meta-learning approaches. Classical methods offer simplicity, interpretability, and low computational overhead, making them suitable for well-characterized problem domains with abundant historical performance data. However, their reliance on hand-crafted meta-features and static selection rules limits their adaptability to novel task distributions [1], [3].

Modern meta-learning approaches, particularly gradient-based methods, overcome these limitations by learning adaptive representations and initialization strategies directly from data. The approximately 5% improvement in accuracy achieved by the Improved Reptile variant over naive transfer learning baselines demonstrates the practical value of meta-learned initializations. Furthermore, the reduction in adaptation steps from 5 to 3 translates to significant computational savings during the inference phase, which is critical for real-time deployment scenarios [7], [15].

RL and evolutionary methods occupy a distinct niche by enabling algorithm selection in non-differentiable and combinatorial settings. While their computational costs remain a significant barrier, recent advances in sample-efficient RL algorithms and parallelized evolutionary search offer promising paths toward practical scalability [10], [11].

Table 5. Strengths and limitations of meta-learning methods.

Method	Strengths	Weakness
Gradient-based (MAML)	Fast adaptation, generalizable	Sensitive to task heterogeneity
Reptile/improved reptile	Efficient first-order updates	Limited exploration, may converge slowly
ES-based meta-learner	Handles complex search spaces	Computationally expensive
RL-based Selector	Adapts to sequential and dynamic tasks	Requires careful reward design
Hybrid (gradient+ES)	Balance of speed and flexibility	High implementation complexity

5.2 | Limitations

Meta-model overfitting: when the number of meta-training tasks is limited or when tasks are insufficiently diverse, the meta-learner may overfit to the training task distribution. Such overfitting manifests as strong performance on training tasks but poor generalization to out-of-distribution tasks. Regularization techniques such as task augmentation, meta-dropout, and early stopping of the outer loop can mitigate this issue [4], [14].

Sensitivity to task distribution: the quality of meta-learned representations is fundamentally bounded by the representativeness of the training task distribution. If the test tasks are drawn from a substantially different distribution, meta-learning may offer no advantage over naive approaches. Robust meta-learning methods that explicitly model task distribution shift remain an active area of research.

High computational cost: second-order gradient computations in MAML scale quadratically with model size, and RL-based methods require thousands of environment interactions per task. These computational demands limit the scalability of current meta-learning approaches to very large models or very large task distributions.

Requirement for large sampled tasks: effective meta-learning typically requires access to a large number of related tasks for meta-training. In domains where task diversity is inherently limited (e.g., rare disease diagnosis), constructing a sufficiently rich meta-training distribution remains challenging.

6 | Conclusion

This paper has presented a comprehensive review of meta-learning approaches for algorithm selection and optimization. Meta-learning provides a principled framework for facilitating rapid adaptation to novel tasks by leveraging knowledge extracted from distributions of related problems.

Our analysis reveals that classical algorithm selection methods, while simple and transparent, are fundamentally constrained by their reliance on hand-crafted meta-features and historical performance data. Gradient-based meta-learning methods, particularly MAML and its first-order variants such as Reptile, provide an effective balance between adaptation speed and accuracy, achieving up to 95.3% accuracy with as few as three adaptation steps. RL and evolutionary methods offer unique flexibility for non-differentiable and combinatorial settings but incur substantially higher computational costs. Hybrid approaches that combine gradient-based efficiency with evolutionary exploration represent the most promising direction for achieving an optimal balance of performance and flexibility.

Several important directions for future research emerge from this analysis. First, the development of hybrid meta-learning models that integrate gradient-based inner-loop adaptation with evolutionary outer-loop search could combine the strengths of both paradigms. Second, cost-aware meta-learning objectives that explicitly balance accuracy with computational budget constraints are needed for practical deployment. Third, improved generalization techniques including task augmentation, distributionally robust meta-objectives, and continual meta-learning can address the challenge of meta-model overfitting. Fourth, addressing meta-overfitting through principled regularization and validation frameworks remains critical. Finally, the establishment of standardized benchmarks and evaluation protocols across diverse domains (vision, NLP, combinatorial optimization) will facilitate more rigorous comparisons and accelerate progress in the field.

Authors' Contributions

All aspects of the research and manuscript preparation were carried out by the author. The author has read and approved the final version of the manuscript.

Funding

Not applicable.

Data Availability

All data supporting the reported findings in this research paper are provided within the manuscript.

Conflict of Interest

The author declares that they do not have any conflict of interest.

Consent for Publication

The author confirms consent for the publication of this work.

Ethics Approval and Consent to Participate

This article does not contain any studies with human participants performed by the author.

References

[1] Rice, J. R. (1976). The algorithm selection problem. In *In advances in computers* (Vol. 15, pp. 65–118). Elsevier. https://doi.org/10.1016/S0065-2458(08)60520-3

[2] Kotthoff, L. (2016). Algorithm selection for combinatorial search problems: A survey. In *Data mining and constraint programming: Foundations of a cross-disciplinary approach* (pp. 149–190). Springer. https://doi.org/10.1007/978-3-319-50137-6_7

[3] Brazdil, P., Giraud Carrier, C., Soares, C., & Vilalta, R. (2009). *Metalearning: Applications to data mining*. Springer. https://doi.org/10.1007/978-3-540-73263-1_3

[4] Hospedales, T., Antoniou, A., Micaelli, P., & Storkey, A. (2021). Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, *44*(9), 5149–5169. https://doi.org/10.1109/TPAMI.2021.3079209

[5] Leyton Brown, K., E. Nudelman, G. Andrew, J. McFadden, and Y. S. (2003). *A portfolio approach to algorithm selection*. IJCAI. https://www.ijcai.org/Proceedings/03/Papers/268.pdf?utm

[6] Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *Proceedings of the 34th international conference on machine learning (ICML 2017)* (pp. 1126–1135). Proceedings of Machine Learning Research. https://proceedings.mlr.press/v70/finn17a.html?utm

[7] Nichol, A., Achiam, J., & Schulman, J. (2018). *On first-order meta-learning algorithms*. https://doi.org/10.48550/arXiv.1803.02999

[8] Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., & Guyon, I. (2021). Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. *NeurIPS 2020 competition and demonstration track* (pp. 3–26). Proceedings of Machine Learning Research. https://proceedings.mlr.press/v133/turner21a.html

[9] Lagoudakis, M. G., & Littman, M. L. (2000). Algorithm selection using reinforcement learning. *The 17th international conference on machine learning (ICML 2000)* (pp. 511–518). Morgan Kaufmann Publishers. https://www.researchgate.net/profile/Michael-Littman/publication/2871654

[10] Lagoudakis, M. G., & Littman, M. L. (2000). Reinforcement learning for algorithm selection. *AAAI-2000 (17th national conference on artificial intelligence)*. AAAI Press. https://cdn.aaai.org/AAAI/2000/AAAI00-164.pdf

[11] Song, X., Gao, W., Yang, Y., Choromanski, K., Pacchiano, A., & Tang, Y. (2019). *Es-maml: Simple hessian-free meta learning*. https://doi.org/10.48550/arXiv.1910.01215

[12] Lai, W., Libovicky, J., & Fraser, A. (2022). Improving both domain robustness and domain adaptability in machine translation. *29th international conference on computational linguistics (Coling 2022)* (pp. 5191–5204). ICCL. https://aclanthology.org/2022.coling-1.461/

[13] Lee, H., Li, S. W., & Vu, T. (2022). Meta learning for natural language processing: A survey. *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 666–684). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.49

[14] Li, R., Wang, X., & Yu, H. (2020). Metamt, a meta learning method leveraging multiple domain data for low resource machine translation. *Proceedings of the aaai conference on artificial intelligence* (Vol. 34, pp. 8245–8252). AAAI Press. https://doi.org/10.1609/aaai.v34i05.6339

[15] Li, L., Jamieson, K., Rostamizadeh, A., Gonina, E., Hardt, M., Recht, B., & Talwalkar, A. (2020). A system for massively parallel hyperparameter tuning. *Proceedings of machine learning and systems*, *2*, 230–246. https://doi.org/10.48550/arXiv.1810.05934