Automated Model Selection and Hyperparameter Optimization Using Bayesian Optimization

Date: May 21, 2025

# Abstract

Bayesian Optimization (BO) has become a cornerstone technique for automating the joint tasks of model selection and hyperparameter tuning. By building a probabilistic surrogate of the objective function and using an acquisition strategy to balance exploration and exploitation, BO can discover high‑performing model–hyperparameter configurations using orders of magnitude fewer evaluations than exhaustive or random search.

# 1. Introduction

Selecting the best model architecture and tuning its hyperparameters are critical but computationally expensive steps in the machine‑learning pipeline—particularly for deep‑learning and edge‑deployed applications such as real‑time facial recognition on smart glasses. Traditional searches (grid or random) scale poorly with dimensionality and compute budget, motivating adaptive, sample‑efficient methods like Bayesian Optimization.

# 2. Bayesian Optimization Fundamentals

BO iteratively (i) updates a surrogate model (commonly a Gaussian Process or Tree‑Structured Parzen Estimator) from observed evaluations and (ii) maximizes an acquisition function (e.g., Expected Improvement, Upper Confidence Bound) to propose the next configuration. This loop continues until a budget (trials or time) is reached. Extensions include multi‑fidelity BO (e.g., combining with Successive Halving/HyperBand) and constrained or multi‑objective BO.

# 3. Algorithms and Frameworks

Popular open‑source frameworks—Optuna, Hyperopt, Ray Tune, BoTorch, and scikit‑optimize—provide production‑ready BO implementations. Optuna, for example, offers samplers based on TPE, CMA‑ES, and since v3.6 a light‑weight Gaussian‑Process sampler, as well as OptunaHub for sharing optimization recipes.

# 4. Advances in 2024‑2025

• ASHA‑BO: Roy et al. (2024) coupled BO with ASHA to reduce wall‑clock time by terminating poorly performing trials early.

• OptunaHub & Distributed Storage Proxy: Optuna v4.2 (March 2025) introduced gRPC‑based storage and constraint‑aware GP samplers for large‑scale, distributed BO.

• Systematic Reviews: Surveys in 2024 synthesize empirical comparisons of BO against evolutionary and gradient‑based HPO, confirming BO’s superior sample efficiency.

# 5. Edge‑Device Considerations

When tuning models destined for microcontrollers or single‑board computers (e.g., Arduino Portenta H7), objectives should penalize latency, memory, and energy alongside accuracy. Multi‑objective BO or scalarized cost functions are recommended. Hardware‑in‑the‑loop evaluation—with models compiled to ONNX/TFLite and measured on the target board—avoids simulation‑reality gaps.

# 6. Experimental Design Guidelines

• Define a realistic search space: Start with log‑uniform priors for learning rates and integer ranges for depth/width.

• Warm‑start from expert defaults or previous runs.

• Use multi‑fidelity schedules (epochs, image resolution) to prune cheap proxies before full training.

• Track and cache trial metadata for reproducibility and surrogate re‑use across related tasks.

# 7. Future Directions

Research frontiers include federated BO (privacy‑preserving HPO across devices), neural acquisition functions, BO for neural architecture search (NAS) with large language model priors, and integrating BO with reinforcement‑learning‑based controllers for continual adaptation on‑device.

# 8. Conclusion

Bayesian Optimization delivers principled, data‑efficient automation of model selection and hyperparameter tuning. Recent algorithmic and software advances make BO practical for real‑world, resource‑constrained deployments such as facial‑recognition smart glasses, unlocking higher accuracy within tight compute and power envelopes.

# References

[1] Roy D.K. et al. (2024). Automated Model Selection Using Bayesian Optimization and the Asynchronous Successive Halving Algorithm. Agriculture, 14(2):278.

[2] Xia Y. et al. (2024). A Systematic Review of Hyperparameter Optimization Techniques in Machine Learning. Data‑Centric AI Journal, 3(1):1‑27.

[3] Optuna Team (2024). Announcing Optuna 3.6 (blog).

[4] Optuna Team (2025). Gaussian Process‑Based Sampler Can Now Handle Inequality Constraints (blog).

[5] Hasan T. et al. (2025). Bayesian Optimization Techniques for Automated Model Selection. QITP International Journal of Data Science, 5(1):1‑15.