Multi-Objective Bayesian Optimization with Reinforcement Learning for Edge Deployment of DNNs on Microcontrollers

Deutel, Mark, et al. "Combining Multi-Objective Bayesian Optimization with Reinforcement Learning for TinyML." ACM Transactions on Evolutionary Learning (2025).

Mark Deutel[1], Georgios Kontes[2], Christopher Mutschler[2], Jürgen Teich[1]

1. *Friedrich-Alexander-Universität Erlangen-Nürnberg*
2. *Fraunhofer Institute for Integrated Circuits IIS*

Motivation

Efficient AI – Machine Learning on the Edge

Efficiency

- Processing of data close to the sensor
- Re-usage of (hardware) resources

Reliability

- No communication via error prone network required
- Short, predictable “round-trip time"

Efficient AI

Cost

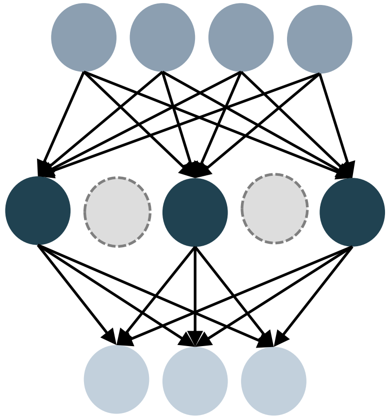
- Exploitation of already available cheap consumer-grade hardware
- Low energy footprint

Privacy

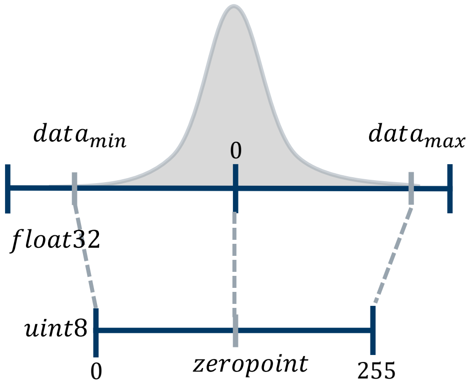
- Possibly confidential data is processed on the sensor node
- No connection to external cloud or server required

Deep Compression Techniques[1]:

Neural Network Pruning



Quantization



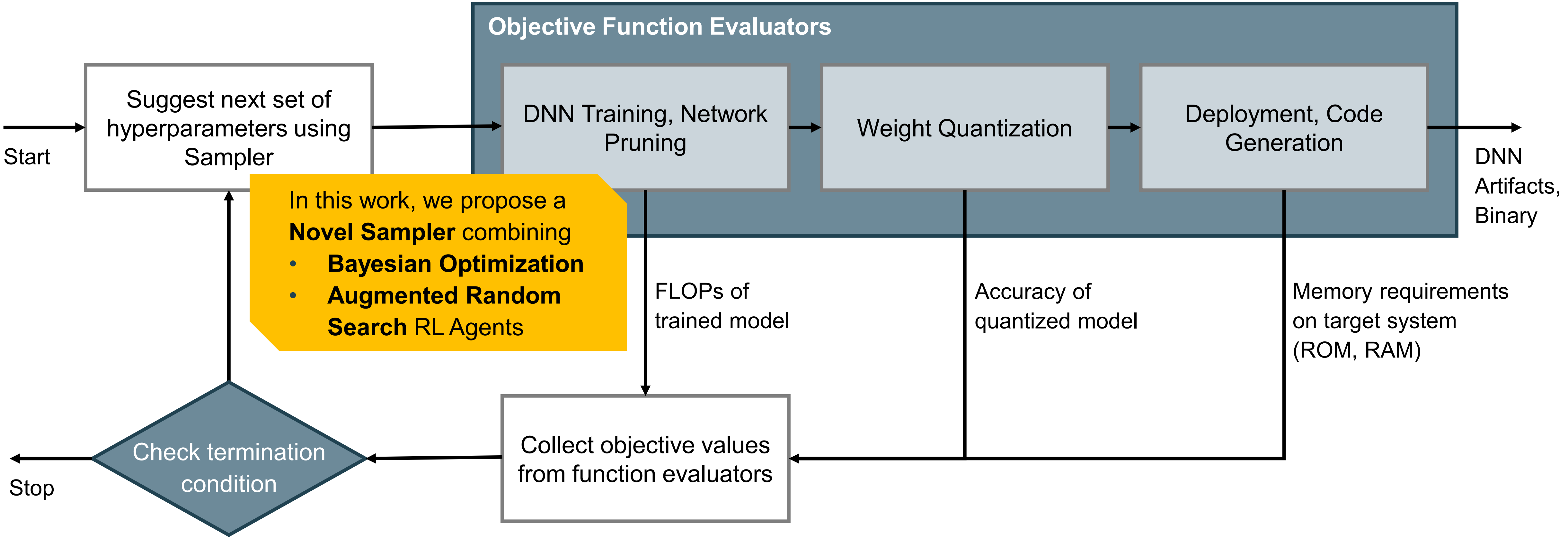
- Deep compression techniques are controlled by hyperparameters
- Selecting hyperparameters is highly dataset dependent but strongly influences the accuracy resource trade-off
- Hyperparameter exploration can be done with multi-objective optimization, i.e., Neural Architecture Search (NAS)

1. Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." *arXiv preprint arXiv:1510.00149* (2015).

Optimizing DNNs with Multi-Objective Optimization

Exploration Loop

Querying the objective function evaluators, especially DNN training is resource and time consuming



In this work, we propose a **Novel Sampler** combining

- **Bayesian Optimization**
- **Augmented Random Search** RL Agents

Optimizing DNNs with Multi-Objective Optimization

Augmented Random Search[1] (1)

Minimize the objective $f(x) = [1 - Acc(x), ROM(x), RAM(x), FLOPs(x)]$ until the search budget is exhausted.

For each sample, re-fit the Bayesian surrogate model, Gaussian Process GP, on objectives using prior samples.

For each local MLP RL policy $\pi_l, l \in \{0,1,\dots,L-1\}$, train it on the GP using the centers of L-means clusters of the current Pareto-front as starting points $x_l \in X$ with X being the search-space.

For each training step $i \in \{0,1,\dots,I-1\}$

Sample directions $\varphi_1\ \varphi_2, \dots, \varphi_N \in R^{n\times m}$ with i.i.d. standard normal entries.

Perform j $\in \{1,2,\dots,N\}$ rollouts over the horizon H from the GP using

$$\pi_{l,j,+}(x_l) = (\Theta_{l,i} + \nu\varphi_j)x_l$$
$$\pi_{l,j,-}(x_l) = (\Theta_{l,i} - \nu\varphi_j)x_l$$

Collect rewards $r(\pi_{l,j,+}(x_l))$ and $r(\pi_{l,j,-}(x_l))$ by scaling objectives and evaluating the Expected Improvement (EI) empirically using Monte Carlo Sampling.

Sort the N direction pairs by $\max\{r(\pi_{l,j,+}(x_l)), r(\pi_{l,j,-}(x_l))\}$ and then update the MLP weights $\Theta_{l,i}$ using a top-K selection.

$$\Theta_{l,i+1} = \Theta_{l,i} + \frac{\alpha}{K\sigma_R}\sum_{k=1}^{K}\left(r(\pi_{l,j,+}(x_l)) - r(\pi_{l,j,-}(x_l))\right)\varphi_K$$

Evaluate trained policy π_l by performing a rollout using

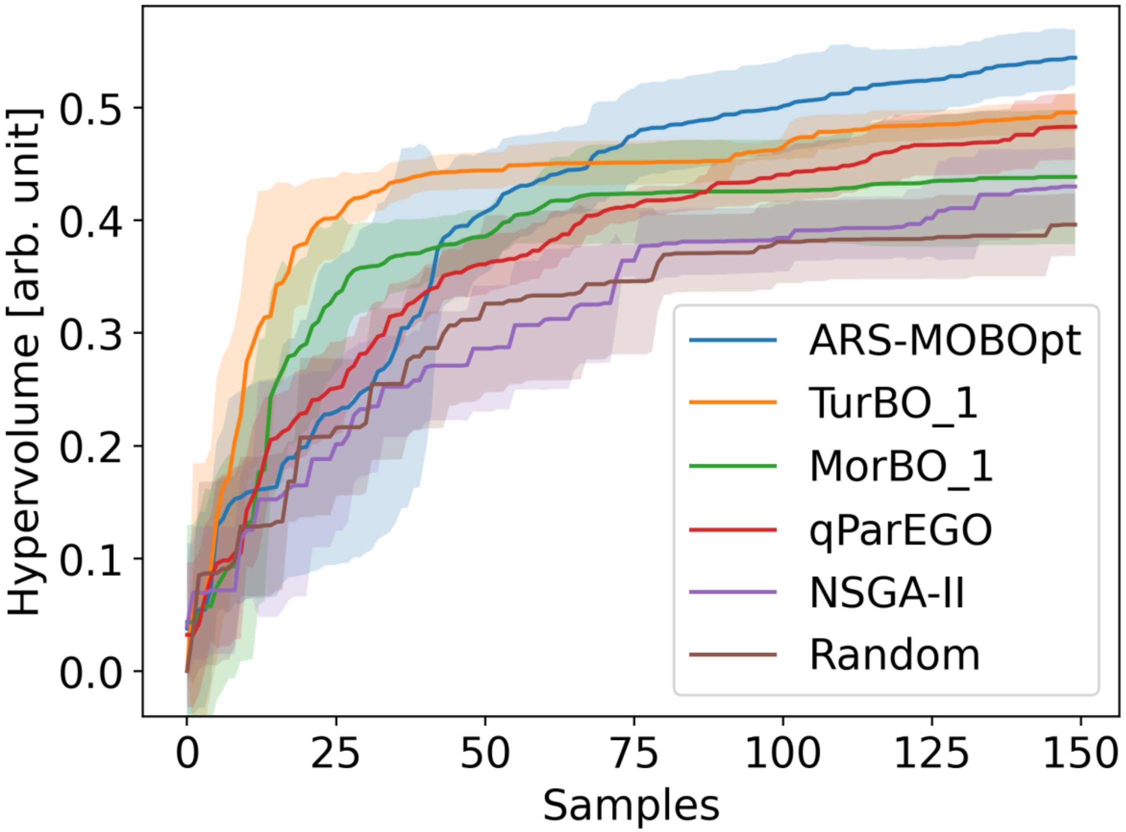
$$\pi_{l,j}(x_l) = (\Theta_l)x_l$$

Let hyperparameters proposed by the best performing RL policy be evaluated next by the function evaluators; Update prior with the result from the function evaluators

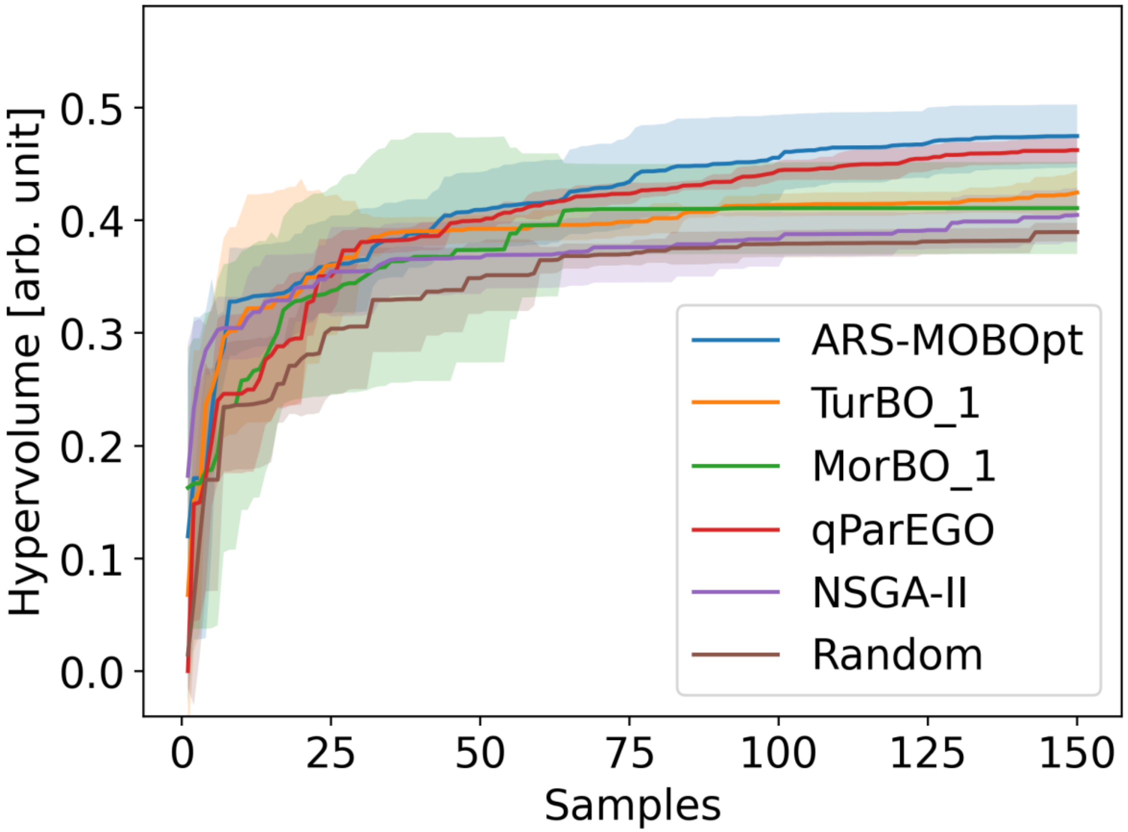
1. Mania, Horia, Aurelia Guy, and Benjamin Recht. "Simple random search provides a competitive approach to reinforcement learning." *arXiv preprint arXiv:1803.07055* (2018).

Evaluation

Hypervolume Improvement



Hypervolume Improvement for ResNet18, CIFAR10,
150 samples, 5 seeds each



Hypervolume Improvement for MobileNetV3, DaLiAc,
150 samples, 5 seeds each

Thank you for your attention!

Contact: mark.deutel@fau.de

Optimizing DNNs with Multi-Objective Optimization

Bayesian Optimization[1]

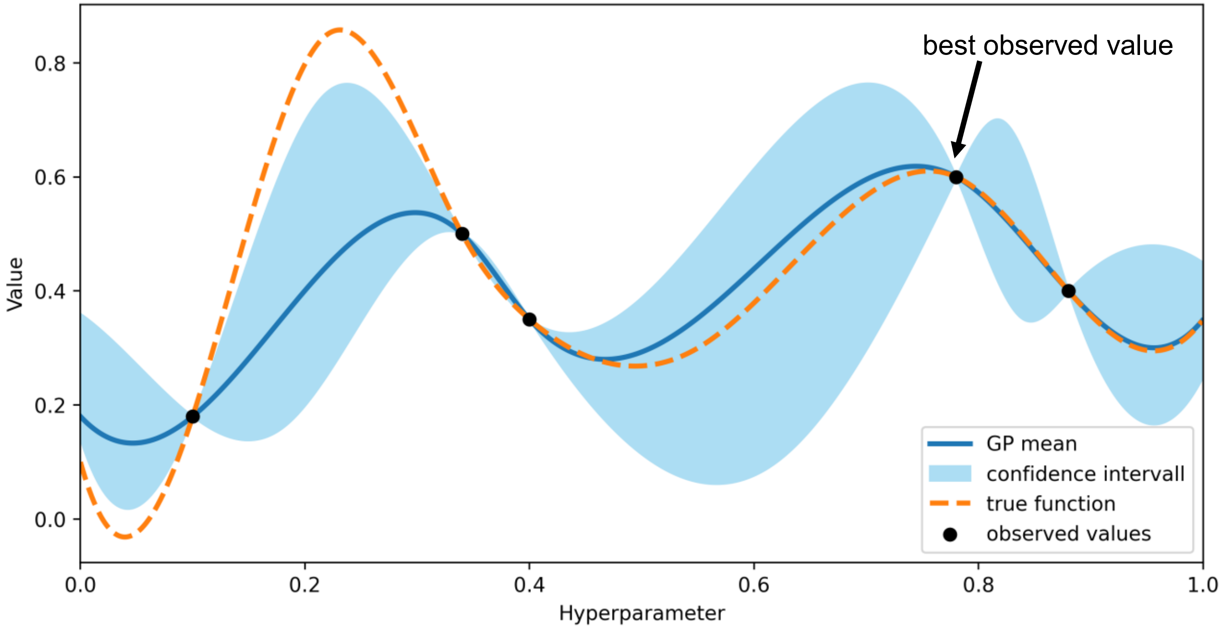
- Improve sample efficiency of black-box optimization by optimizing a surrogate model (Gaussian process)
  - Fit Gaussian process using previous samples (prior)
  - Solve optimization problem on surrogate model using an acquisition function

Expected Improvement[1]

$$\mathrm{EI}(X) = \mathbb{E}\max(f(x) - f^*, 0)$$

  - Suggest next parametrization to be evaluated
- Evaluate posterior of surrogate either analytical or by using Monte-Carlo (MC) sampling[2]

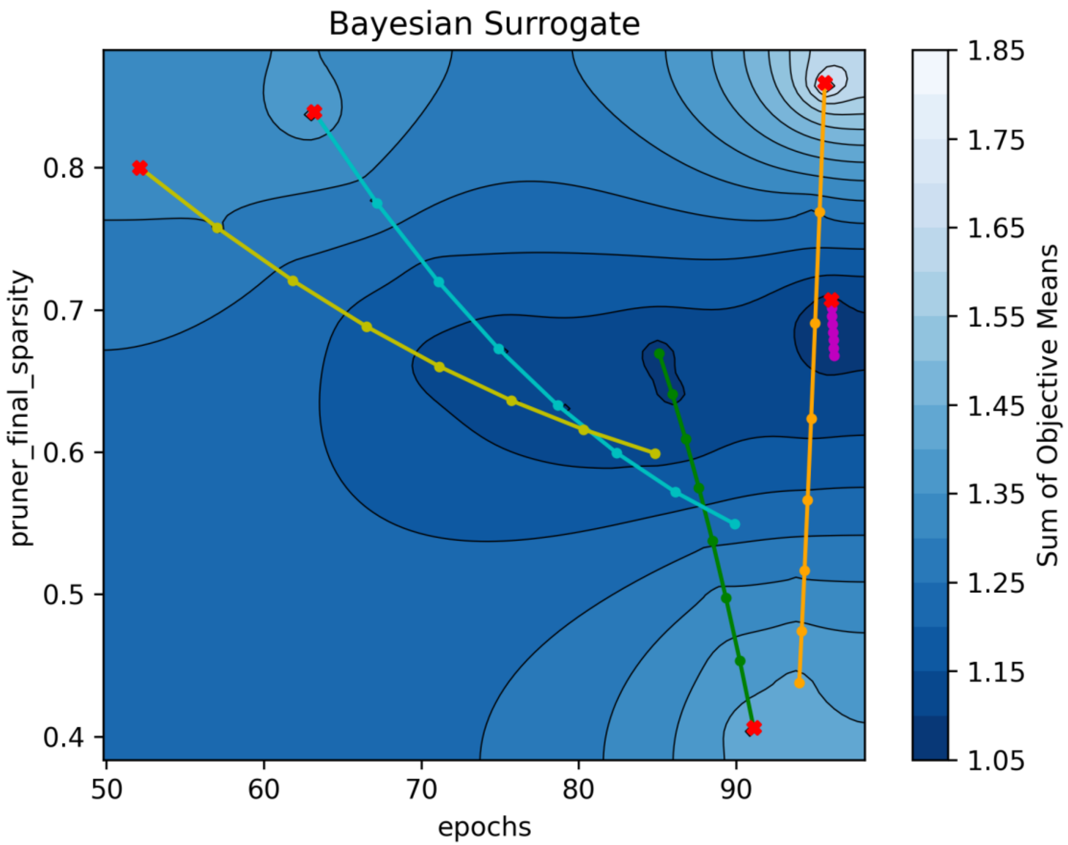
GP estimate of a function



1. Mŏckus, J. On Bayesian methods for seeking the extremum. In Optimization Techniques IFIP Technical Conference, pp. 400–404, 1975.
2. Wilson, J. T., Moriconi, R., Hutter, F., and Deisenroth, M. P. The reparameterization trick for acquisition functions. NIPS 2017 Workshop on Bayesian Optimization (BayesOpt 2017), 2017.

Optimizing DNNs with Multi-Objective Optimization

Augmented Random Search[1] (2)



1. Mania, Horia, Aurelia Guy, and Benjamin Recht. "Simple random search provides a competitive approach to reinforcement learning." *arXiv preprint arXiv:1803.07055* (2018).