RADICAL-Learning

Radical.hpo

Shantenu Jha: Project Manager

Matteo Turilli: Technical Supervisor

What is HPO?

- Any machine learning model has some values (hyperparameters) that need to be specified a priori before the training process.
- They help adapt the model to the data and they influence the quality of the prediction.

➢ Hyperparameter optimization deals with the search of the best combination of values for the given model, and there are already many methods that help us find them.

HPO Approaches

1. Grid Search: try all combinations (brute force)
2. Random Search: try as many combinations as possible (better)
3. Informed Search (Bayesian Optimization): try the most promising combinations by reevaluating where to look next (even better)

4. ...more

Background

- There already exists HyperSpace, a parallel Bayesian Model-Based Optimization (parallel Bayesian SMBO) library with one of the main goals of optimizing model performance with respect to hyperparameters. It supports Scikit-Optimize, RoBo and Hyperband

    - i.e, we can run a bag-of-tasks of optimizations where each task runs the Gaussian process that explores a search space (hyperspace)

Goal

- Since our HPO module wants to integrate HyperSpace with EnTK, the logical path to follow is to treat each hyperspace optimization as an independent task, and set up a bag (stage) of tasks inside a single pipeline

    - We take full advantage of EnTK and achieve concurrency at EnTK level, while still making use of HyperSpace’s spaces creation and optimization through its supported HPO engines

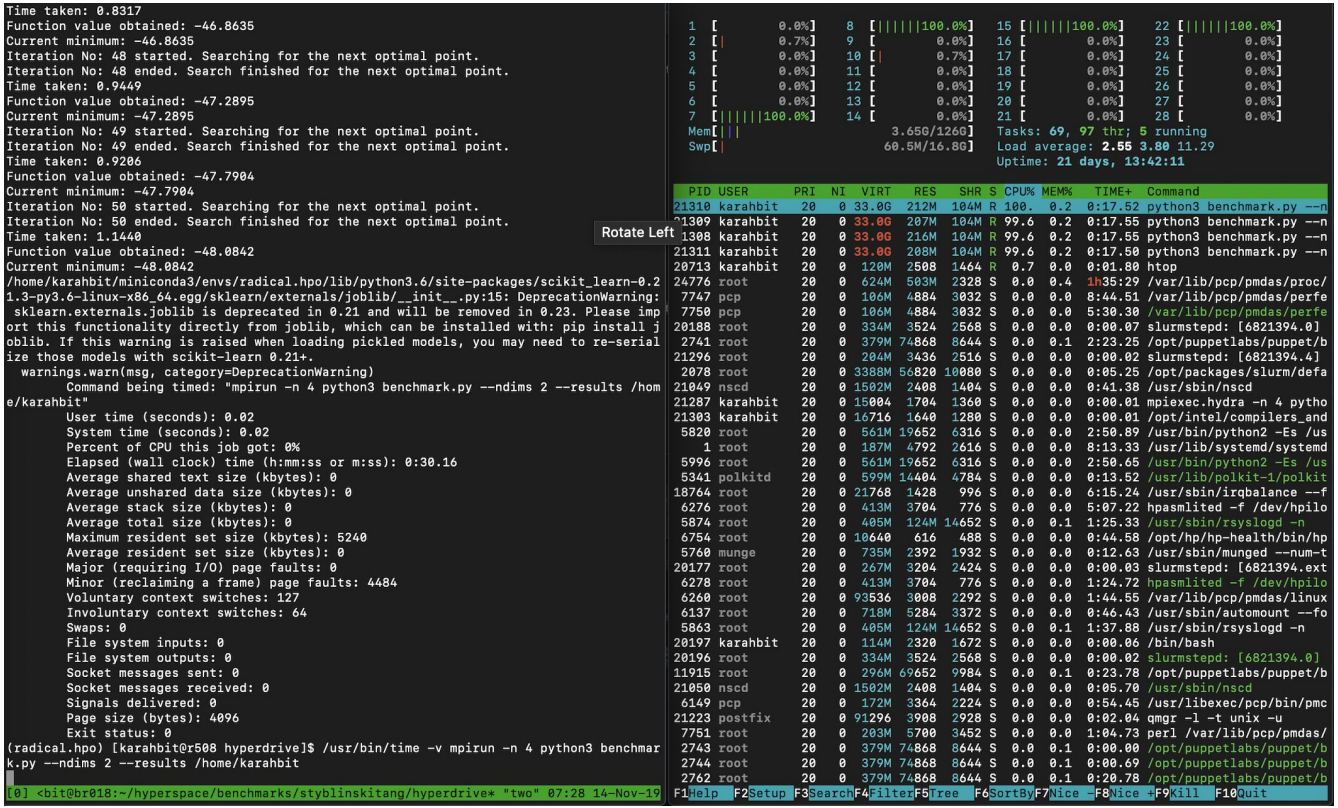
Requirements

- Functional
  - It must satisfy Bayesian Model-Based Optimization (SMBO) requirements:
    - Define a machine learning model (or objective function)
    - Provide with train/test datasets (if ML model selected)
    - Define a validation protocol: cross-validation (if ML model selected)
    - Define the parameter search space: upper and lower bounds for each hyperparameter
    - Define the optimization function: Gaussian process with guided sampling
  - Must use RADICAL-EnTK
- Non-Functional
  - The code must be simple
  - The code must be easy to maintain
- Non-Functional
  - It must use the maximum number of cores available on the largest XSEDE machine, without significant overhead

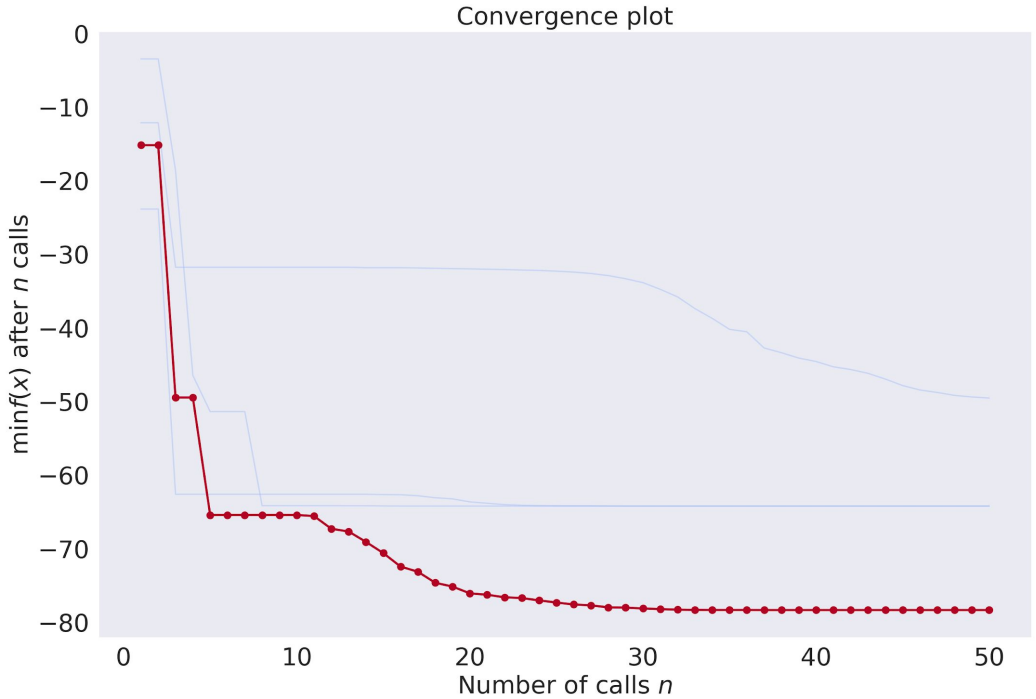
Execution of HyperSpace

- HyperSpaces use Scikit-Optimize
- Creates combinations of hyperparameters (hyperspaces) using overlapping boundaries between hyperparameters
- Bag-of-tasks are executed with mpi4py
- Number of tasks depends on the number of hyperparameters for the model:
  - HyperSpaces = 2^H where H is the number of hyperparameters
  - Avg. num of hyperparameters ~ 7-8 but depending on model can go up to 12
  - Each optimization runs for N-iterations, where N is ~100
- Tasks (Bayesian optimization step) are independent on HyperSpace, requiring each 1 MPI Rank at minimum and using 1 core each

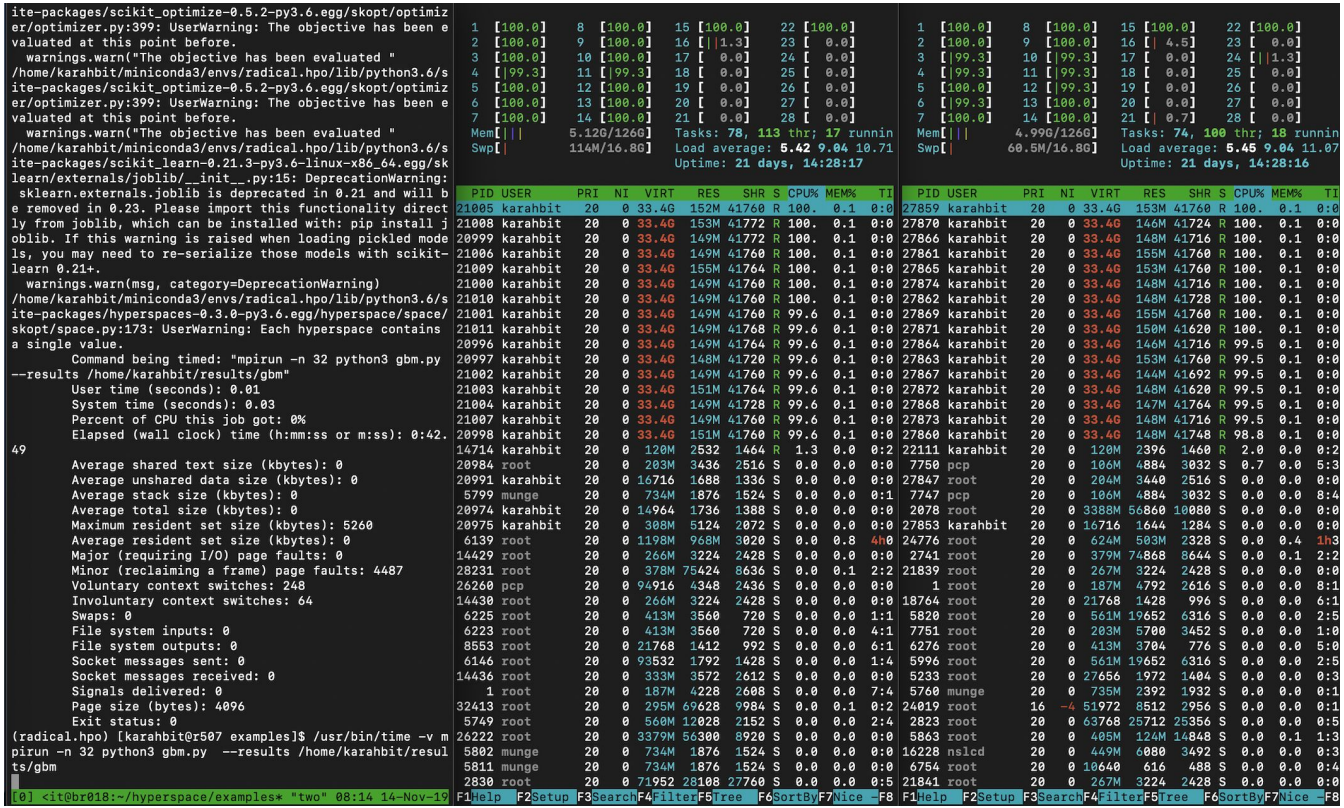
Minimal Example: Styblinski-Tang



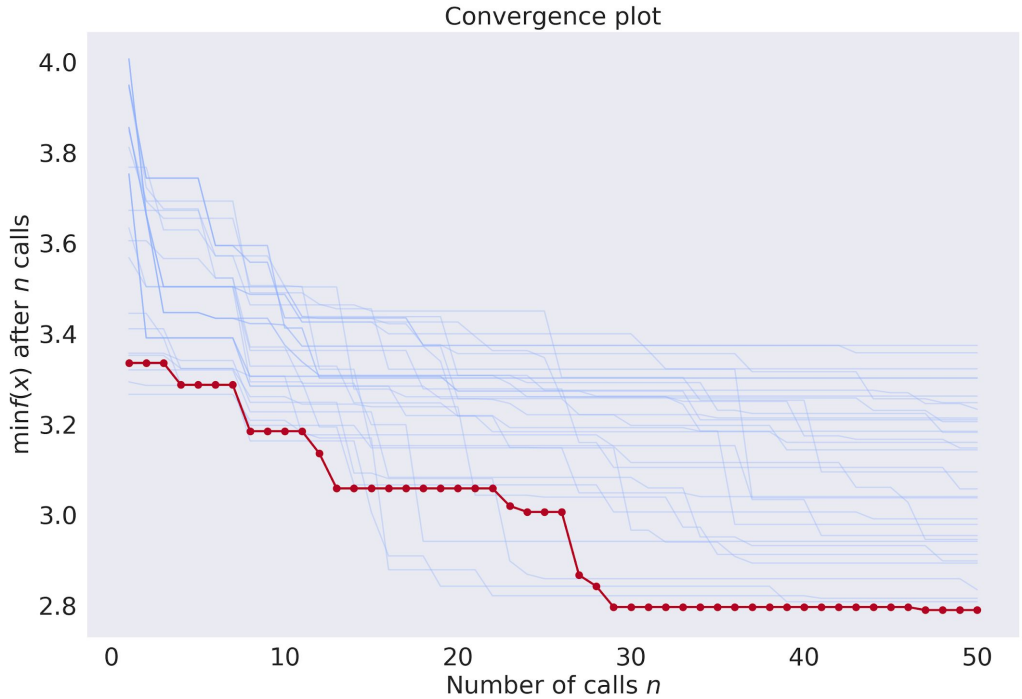
Minimal Example: Styblinski-Tang



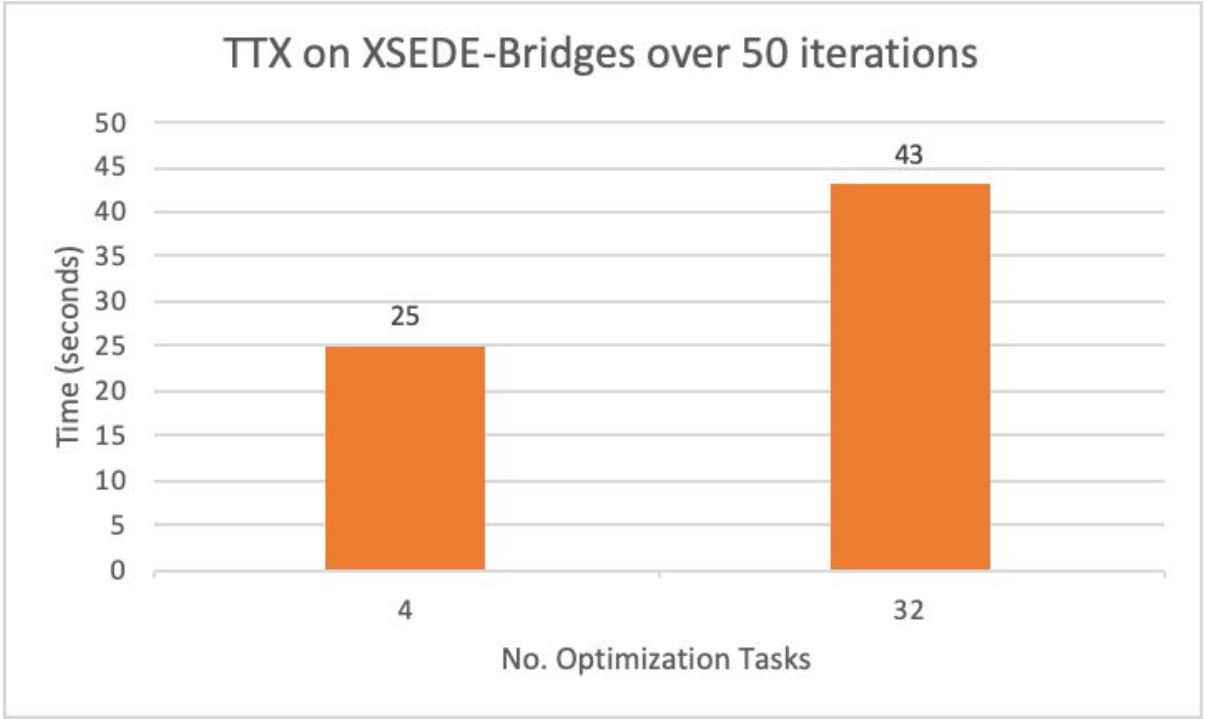
ML Example: Gradient Boosted Trees



ML Example: Gradient Boosted Trees



TTX on XSEDE-Bridges over 50 iterations



References

https://scikit-optimize.github.io

https://examples.dask.org/machine-learning/hyperparam-opt.html

https://towardsdatascience.com/hyperparameter-optimization-in-python-part-0-introduction-c4b66791614b

https://github.com/jdakka/hyperspace-RCT

https://github.com/yngtodd/hyperspace

https://buildmedia.readthedocs.org/media/pdf/hyperspace/latest/hyperspace.pdf

https://towardsdatascience.com/an-introductory-example-of-bayesian-optimization-in-python-with-hyperopt-aae40fff4ff0

https://medium.com/criteo-labs/hyper-parameter-optimization-algorithms-2fe447525903