Hyperparameter Tuning in Python: a Complete Guide 2021

19th, 2021

for machine learning or deep learning models is one of the best ways to extract
article, I will show you some of the best ways to do hyperparameter tuning that

nce between parameter and

between a hyperparameter and a parameter in machine learning.

arameters that are estimated by the model from the given data. For example the weights of a deep neural network.
- **Model hyperparameters:** These are the parameters that cannot be estimated by the model from the given data. These parameters are used to estimate the model parameters. For example, the learning rate in deep neural networks.

What is hyperparameter tuning and why it is important?

Hyperparameter tuning is the process of **determining the right combination of hyperparameters** that allows the model to maximize model performance. Setting the correct combination of hyperparameters is the only way to extract the maximum performance out of models.

How do I choose good hyperparameters?

Choosing the right combination of hyperparameters is not an easy task. There are two ways to set them.

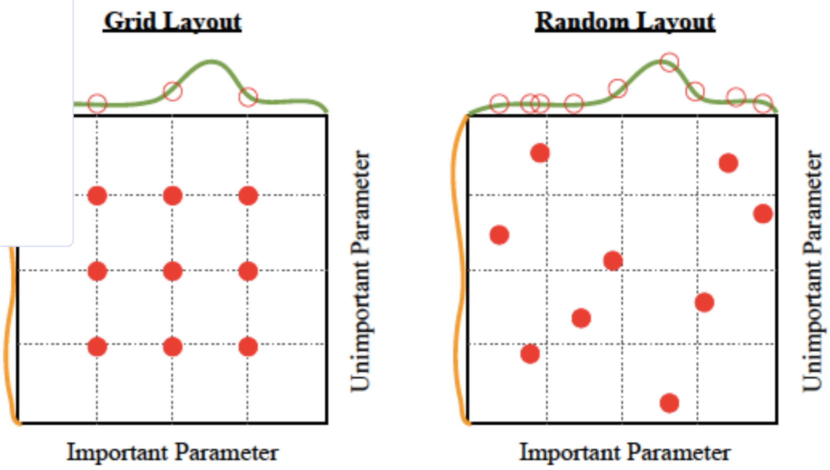
- **Manual hyperparameter tuning:** In this method, different combinations of hyperparameters are set (and experimented with) manually. This is a tedious process and cannot be practical in cases where there are many hyperparameters to try.
- **Automated hyperparameter tuning:** In this method, optimal hyperparameters are found using an algorithm that automates and optimizes the process.

Hyperparameter tuning methods

In this section, I will introduce all of the hyperparameter tuning methods that are popular today.

Random Search

ate a grid of possible values for hyperparameters. Each iteration tries a combination of er. It fits the model on each and **every combination of hyperparameter** possible and inally, it returns the best model with the best hyperparameters.



Source

Bayesian Optimization

Tuning and finding the right hyperparameters for your model is an **optimization problem.** We want to **minimize the loss function** of our model **by changing model parameters**. Bayesian optimization helps us find the minimal point in the minimum number of steps. Bayesian optimization also uses an *acquisition function* that directs sampling to areas where an improvement over the current best observation is likely.

Tree-structured Parzen estimators (TPE)

The idea of Tree-based Parzen optimization is similar to Bayesian optimization. Instead of finding the values of p(y|x) where y is the function to be minimized (e.g., validation loss) and x is the value of hyperparameter the TPE models P(x|y) and P(y). One of the great drawbacks of tree-structured Parzen estimators is that they do not model interactions between the hyper-parameters. That said TPE works extremely well in practice and was battle-tested across most domains.

Hyperparameter tuning algorithms

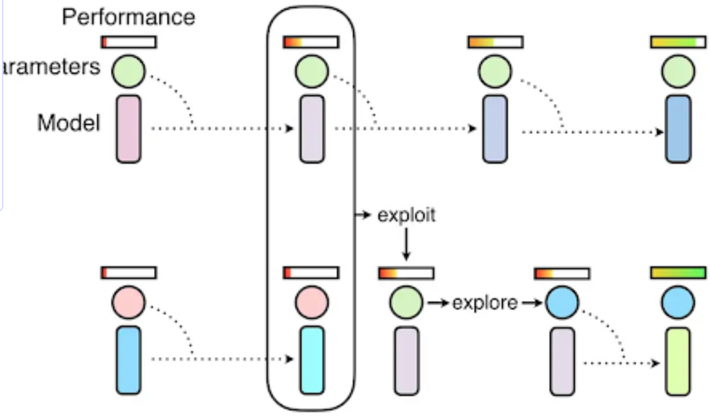
These are the algorithms developed specifically for doing hyperparameter tuning.

Hyperband

Hyperband is a variation of random search, but with some explore-exploit theory to find the best time allocation for each of the configurations. You can check this research paper for further references.

ɑl networks in parallel with ran̶dom hyperparameters. But these networks aren't fully

ɔf the population to refine the hyperparameters and determine the value of
ɔeck this article for more information on PBT.



Source

BOHB

BOHB (Bayesian Optimization and HyperBand) mixes the Hyperband algorithm and Bayesian optimization. You can check this article for further reference.

LEARN MORE

HyperBand and BOHB: Understanding State of the Art Hyperparameter Optimization Algorithms

Tools for hyperparameter optimization

Now that you know what are the methods and algorithms let's talk about tools, and there are a lot of those out there.

Some of the best Hyperparameter Optimization libraries are:

- Scikit-learn (grid search, random search)
- Hyperopt
- Scikit-Optimize
- Optuna
- Ray.tune

Scikit learn

Scikit-learn has implementations for grid search and random search and is a good place to start if you are building

ar chcv runs the search over some number of random parameter combinations
the search over all parameter sets in the grid

a good start but there are better options out there and they often have random search

lar hyperparameter tuning packages available. Hyperopt allows the user to describe a
pects the best results allowing the algorithms in hyperopt to search more efficiently.

Currently, three algorithms are implemented in hyperopt.

- Random Search
- Tree of Parzen Estimators (TPE)
- Adaptive TPE

To use hyperopt, you should first describe:

- the objective function to minimize
- space over which to search
- the database in which to store all the point evaluations of the search
- the search algorithm to use

This tutorial will walk you through how to structure the code and use the hyperopt package to get the best hyperparameters.

RELATED ARTICLES

Automate Hyperparameter Tuning for your models
Optuna vs Hyperopt: Which Hyperparameter Optimization Library Should You Choose?

Scikit-optimize

Scikit-optimize uses Sequential model-based optimization algorithm to find optimal solutions for hyperparameter search problems in less time.

Scikit-optimize provides many features other than hyperparameter optimization such as:

- store and load optimization results,
- convergence plots,
- comparing surrogate models

Optuna

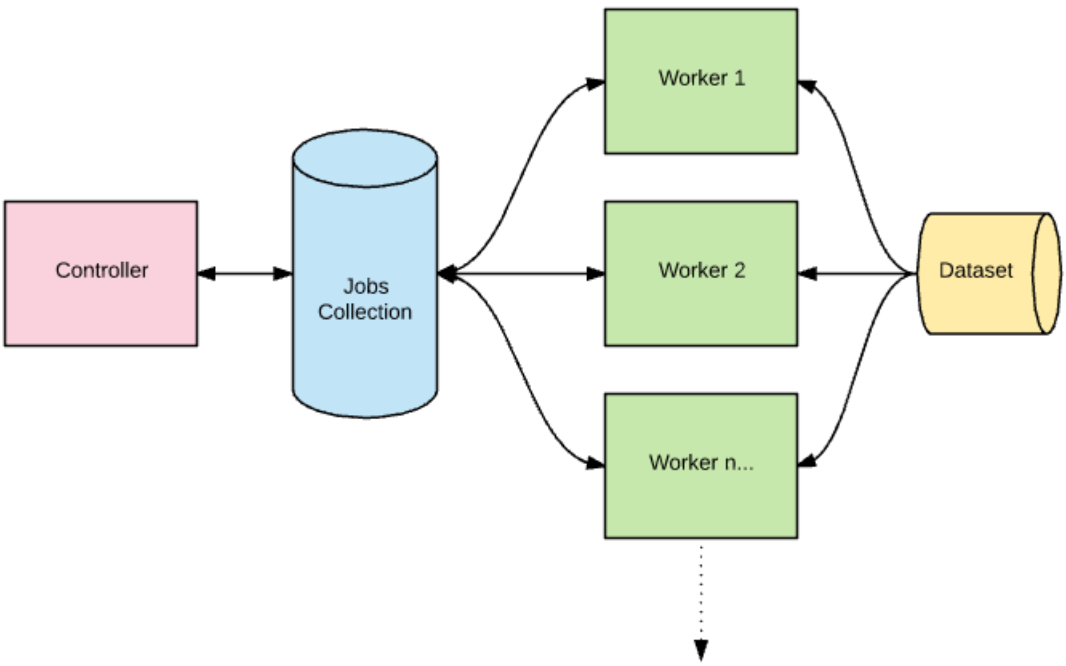
Optuna uses a historical record of trails details to determine the promising area to search for optimizing the hyperparameter and hence finds the optimal hyperparameter in a minimum amount of time.

nentation for tutorials on how to start using optuna.

mentation and hyperparameter tuning at any scale. Ray uses the power of distributed
parameter optimization and has an implementation for several states of the art

Some of the core features provided by ray tune are:

- distributed asynchronous optimization out of the box by leveraging Ray.
- Easily scalable.
- Provided SOTA algorithms such as ASHA, BOHB, and Population-Based Training.
- Supports Tensorboard and MLflow.
- Supports a variety of frameworks such sklearn, xgboost, Tensorflow, pytorch, etc.



Source

You can refer to this tutorial to learn how to implement ray tune for your problem.

Keras Tuner

The Keras Tuner is a library that helps you pick the optimal set of hyperparameters for your TensorFlow program. When you build a model for hyperparameter tuning, you also define the hyperparameter search space in addition to the model architecture. The model you set up for hyperparameter tuning is called a *hypermodel*.

You can define a hypermodel through two approaches:

track of your hyperparameters search when working with:

Hyperparameter tuning resources and examples

In this section, I will share some hyperparameter tuning examples implemented for different ML and DL frameworks.

Random forest

- Understanding Random forest hyperparameters
- Bayesian hyperparameter tuning for random forest
- Random forest tuning using grid search

XGBoost

- XGBoost hyperparameters tuning python
- XGBoost hyperparameters tuning in R
- XGBoost hyperparameter using hyperopt
- Optuna hyperparameter tuning example

LightGBM

- Understanding LightGBM parameters
- LightGBM hyperparameter tuning example
- Optuna for LIghtGBM hyperparameter tuning

Catboost

- Catboost hyperparamters overview
- Grid search for catboost hyperparameter tuning

Keras

orch
ter tuning

fferent hyperparameter tuning algorithms and tools which are currently widely used.
ul in your projects.

Shahul ES

Freelance Data Scientist | Kaggle Master
Data science professional with a strong end to end data science/machine learning and deep learning (NLP) skills. Experienced working in a Data Science/ML Engineer role in multiple startups. Kaggle Kernels Master ranked the top 20 among 100,000+ users.

Follow me on

READ NEXT

How to Track Hyperparameters of Machine Learning Models?

Kamil Kaczmarek | Posted July 1, 2020

Machine learning algorithms are tunable by multiple gauges called hyperparameters. Recent deep learning models are tunable by tens of hyperparameters, that together with data augmentation parameters and training procedure parameters create quite complex space. In the reinforcement learning domain, you should also count environment params.

Data scientists should **control hyperparameter space** well in order to **make progress.**

Here, we will show you **recent practices**, **tips & tricks,** and **tools** to track hyperparameters efficiently and with minimal overhead. You will find yourself in control of most complex deep learning experiments!

Why should I track my hyperparameters? a.k.a. Why is that important?

Almost every deep learning experimentation guideline, like this deep learning book, advises you on how to tune hyperparameters to make models work as expected. In the **experiment-analyze-learn loop**, data scientists must control what changes are being made, so that the "learn" part of the loop is working.

practices for managing hyperparameters. We focus on how to build, keep and pass
…ts.

Best Tools for Model Tuning and Hyperparameter Optimization

by Bunmi Akinremi, May 15th, 2021

Read more

HyperBand and BOHB: Understanding State of the Art Hyperparameter Optimization Algorithms

by MJ Bahmani, November 10th, 2020

Read more

How to Track Hyperparameters of Machine Learning Models?

by Kamil Kaczmarek, July 1st, 2020

Read more