

Grid Search vs. Random Search vs. Bayesian Optimization

Better methods for hyperparameter tuning.



AVI CHAWLA

AUG 22, 2024



35



1



2

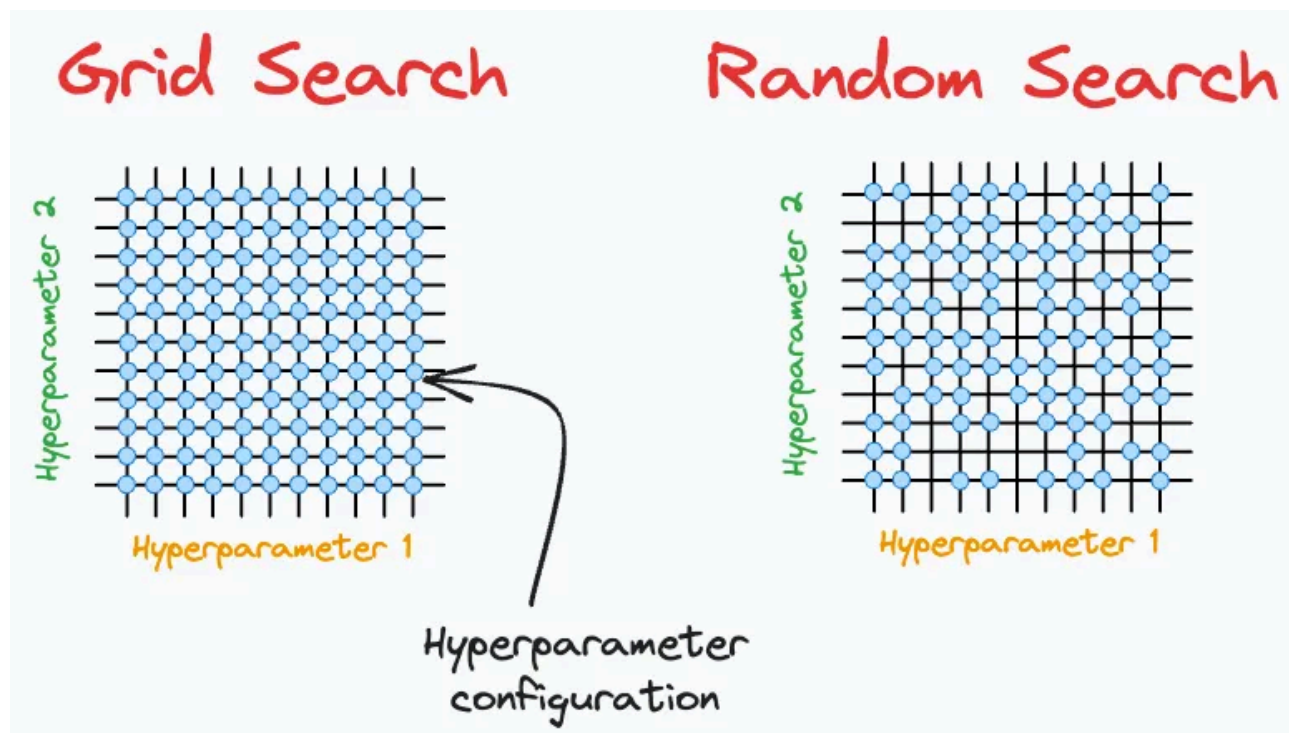
Share



The two most common approaches for hyperparameter tuning are:

- Grid search
- Random search

The visual below depicts how they work:

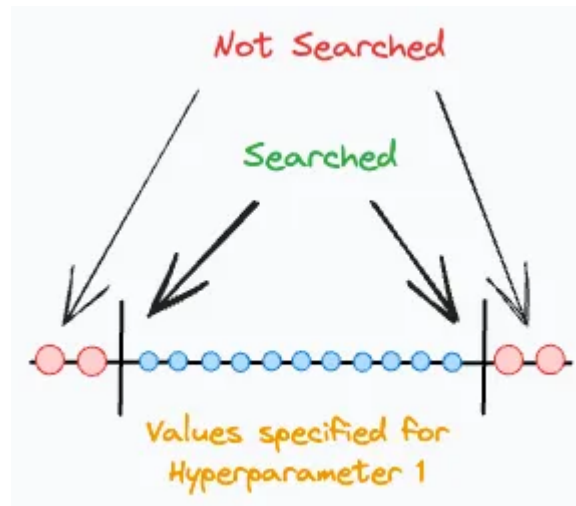


Grid search and random search

But they have many limitations.

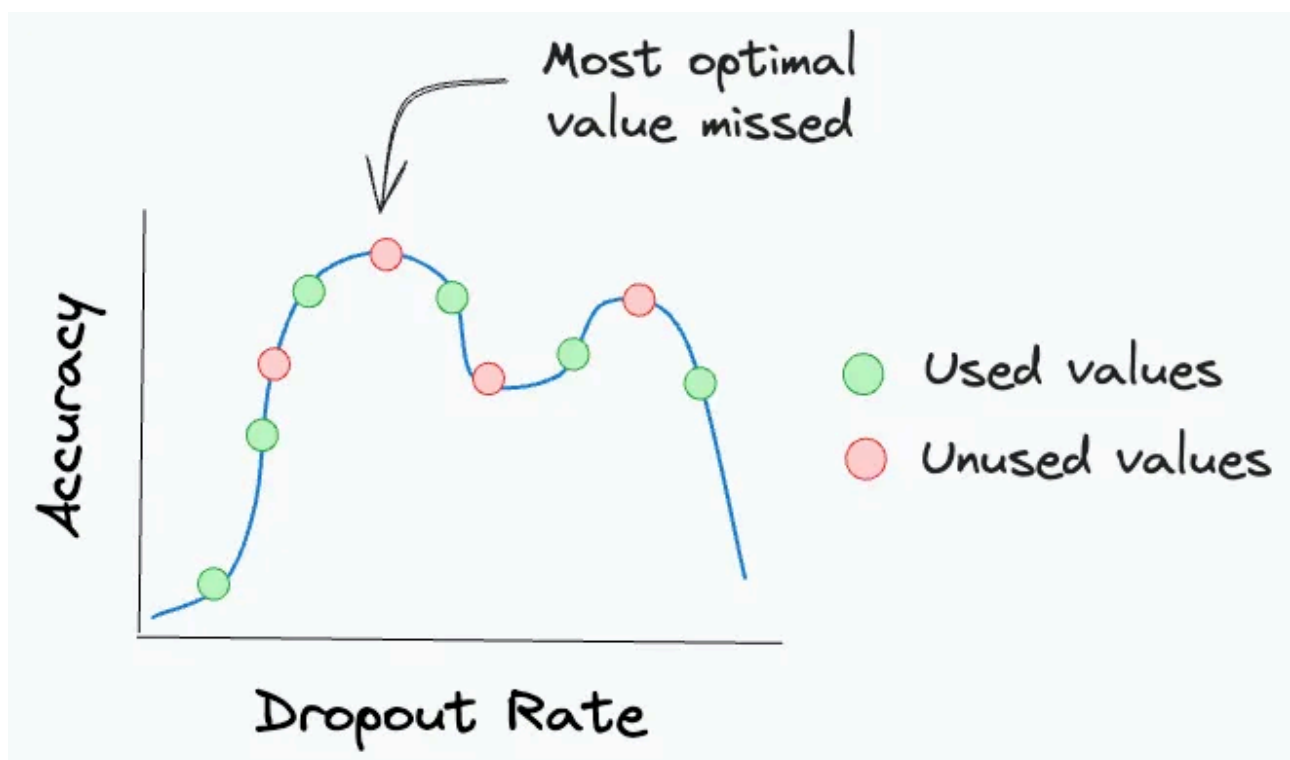
For instance:

- Grid search performs an exhaustive search over all combinations. This is computationally expensive.
- Grid search and random search are restricted to the specified hyperparameter range. Yet, the ideal hyperparameter may exist outside that range.



Limited search space with grid search and random search

- They can ONLY perform discrete searches, even if the hyperparameter is continuous.



Grid search and random search can only try discrete values for continuous hyperparameters

To this end, **Bayesian Optimization** is a highly underappreciated yet immensely powerful approach for tuning hyperparameters.

We covered this in detail with implementation here: [*Bayesian Optimization for Hyperparameter Tuning*](#).

It uses **Bayesian statistics** to estimate the distribution of the best hyperparameters.

Here's how it differs from Grid search and Random Search:

Both Grid search and Random Search evaluate every hyperparameter configuration independently. Thus, they iteratively explore all hyperparameter configurations to find the most optimal one.

However, Bayesian Optimization takes informed steps based on the results of the previous hyperparameter configurations.

This lets it confidently discard non-optimal configurations. Consequently, the model converges to an optimal set of hyperparameters much faster.

The efficacy of Bayesian Optimization is evident from the image below.

Limitation of Grid Search and Random Search



blog.DailyDoseofDS.com

	Grid Search	Random Search	Bayesian Optimization	
Total Iterations	720	360	100	7x less iterations ✓
Total Run-time	242s	113s	48s	5x less run-time ✓
Best Trial Index	667	221	83	Optimal config. found earlier ✓
Best F1 score	0.93	0.93	0.93	Same score ✓

Bayesian optimization leads the model to the same F1 score but:

- it takes 7x fewer iterations
- it executes 5x faster
- it reaches the optimal configuration earlier

Typically, Bayesian Optimization is only advised for intermediate/large models since small models can be trained in quick time with grid search and random search. In other words, the gain in speed-up isn't significant for small models.

If you are curious about Bayesian Optimization, I once published a full deep dive on Bayesian optimization, which you can read here: [Bayesian Optimization for Hyperparameter Tuning](https://blog.dailydoseofds.com/p/grid-search-vs-random-search-vs-bayesian?utm_source=substack&publication_id=1119889&post_id=148017120...).

Why care?

The idea behind Bayesian optimization appeared to be extremely compelling to me when I first learned it.

Learning about this optimized hyperparameter tuning and utilizing it has been extremely helpful to me in building large ML models quickly.

Thus, learning about Bayesian optimization will be immensely valuable if you envision doing the same.

Assuming you have never had any experience with Bayesian optimization before, the article covers:

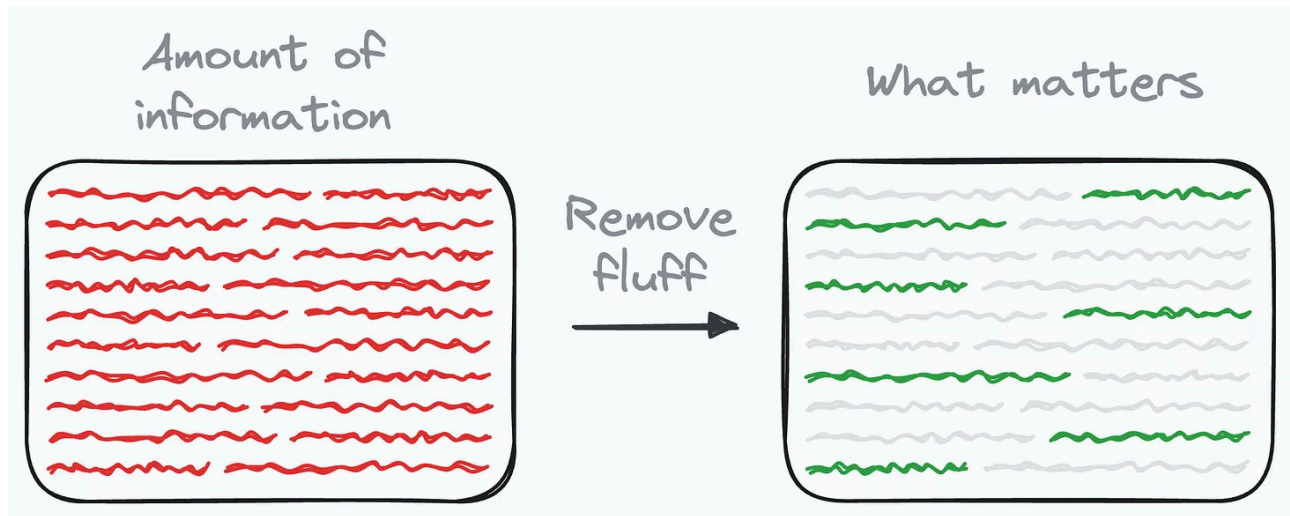
- What is the motivation for Bayesian optimization?
- How does Bayesian optimization work and its intuition?
- Results from the research paper that proposed Bayesian optimization for hyperparameter tuning.
- **A hands-on Bayesian optimization experiment.**
- Comparing Bayesian optimization with grid search and random search.
- Best practices for using Bayesian optimization.

👉 Interested folks can read it here: [Bayesian Optimization for Hyperparameter Tuning](#).

Thanks for reading Daily Dose of Data Science!
Subscribe for free to learn something new and insightful about Python and Data Science every day. Also, get a Free Data Science PDF (550+ pages) with 320+ tips.

Are you overwhelmed with the amount of information in ML/DS?

Every week, I publish no-fluff deep dives on topics that truly matter to your skills for ML/DS roles.



For instance:

- [A Crash Course on Graph Neural Networks \(Implementation Included\)](#)
- [Formulating and Implementing XGBoost From Scratch](#)
- [A Crash Course of Model Calibration – Part 1](#)
- [A Crash Course of Model Calibration – Part 2](#)
- [Conformal Predictions: Build Confidence in Your ML Model's Predictions](#)
- [Quantization: Optimize ML Models to Run Them on Tiny Hardware](#)
- [A Crash Course on Causality – Part 1](#)
- [A Crash Course on Causality – Part 2](#)
- [5 Must-Know Ways to Test ML Models in Production \(Implementation Included\)](#)
- [A Beginner-Friendly Guide to Multi-GPU Model Training](#)
- And many many more.

Join below to unlock all full articles:

SPONSOR US

Get your product in front of 85,000 data scientists and other tech professionals.

Our newsletter puts your products and services directly in front of an audience that matters — thousands of leaders, senior data scientists, machine learning engineers, data analysts, etc., who have influence over significant tech decisions and big purchases.

To ensure your product reaches this influential audience, reserve your space [here](#) or reply to this email to ensure your product reaches this influential audience.



35 Likes · 2 Restacks

← Previous

Next →

1 Comment



Write a comment...



Blessing Iwerumor · Aug 23 · Liked by [Avi Chawla](#)

Thank you for this, if I had known this earlier I would have used it for my msc data science project I'm doing right now and it would have saved me lots of time.

LIKE (1) REPLY SHARE

...