Shahul ES | 7 min | 30th August, 2023

ML Model Development

Choosing the correct hyperparameters for machine learning or deep learning models is one of the best ways to extract the last juice out of your models. In this article, I will show you some of the best ways to do hyperparameter tuning that are available today.

What is the difference between parameter and hyperparameter?

First, let's understand the differences between a hyperparameter and a parameter in machine learning.

- **Model parameters**: These are the parameters that are estimated by the model from the given data. For

- **Model parameters**: These are the parameters 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.

PARAMETERS	HYPERPARAMETERS
They are required for making predictions	They are required for estimating the model parameters
algorithms(Gradient Descent, Adam, Adagrad)	They are estimated by hyperparameter tuning
They are not set manually	They are set manually
The final parameters found after training will decide how the model will perform on unseen data	The choice of hyperparameters decide how efficient the training is. In gradient descent the learning rate decide how efficient and accurate the optimization process is in estimating the parameters

Model parameters vs model hyperparameters | Source: GeeksforGeeks

What is hyperparameter tuning and why it is important?

hyperparameters and see all types of data results like images, metrics, etc. Head over to the docs to see how you can log different metadata to Neptune.

Alternative solutions include W&B, Comet, or MLflow. Check more tools for experiment tracking & management here.

Advantages of manual hyperparameter optimization:

- Tuning hyperparameters manually means more control over the process.
- If you're researching or studying tuning and how it affects the network weights then doing it manually would make sense.

Disadvantages of manual hyperparameter optimization:

- Manual tuning is a tedious process since there can be many trials and keeping track can prove costly and time-consuming.
- This isn't a very practical approach when there are a lot of hyperparameters to consider.

Read about how to manually optimize Machine Learning model hyperparameters here.

- Tuning hyperparameters manually means more control over the process.
- If you're researching or studying tuning and how it affects the network weights then doing it manually would make sense.

Disadvantages of manual hyperparameter optimization:

- Manual tuning is a tedious process since there can be many trials and keeping track can prove costly and time-consuming.
- This isn't a very practical approach when there are a lot of hyperparameters to consider.

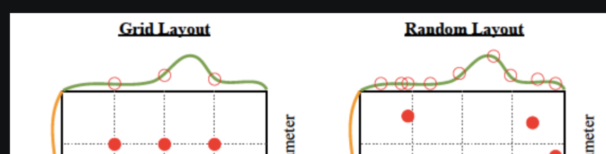
Read about how to manually optimize Machine Learning model hyperparameters here.

Random Search

In the random search method, we create a grid of possible values for hyperparameters. Each iteration tries a random combination of hyperparameters from this grid, records the performance, and lastly returns the combination of hyperparameters that provided the best performance.

Grid Search

In the grid search method, we create a grid of possible values for hyperparameters. Each iteration tries a combination of hyperparameters in a specific order. It fits the model on each and every combination of hyperparameters possible and records the model performance. Finally, it returns the best model with the best hyperparameters.



combination of hyperparameters that provided the best performance.

Grid Search

In the grid search method, we create a grid of possible values for hyperparameters. Each iteration tries a combination of hyperparameters in a specific order. It fits the model on each and every combination of hyperparameters possible and records the model performance. Finally, it returns the best model with the best hyperparameters.

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

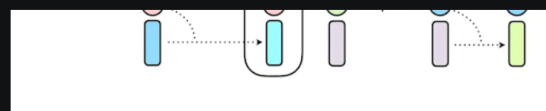
Hyperband is a variation of random search, but with some explore-exploit theory to find the best time allocation for each of the configurations. You can check this research paper for further references.

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.



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

Source

BOHB

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

Learn more

11. SigOpt
12. Fabolas

1. Scikit-learn

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

For both of those methods, scikit-learn trains and evaluates a model in a k fold cross-validation setting over various parameter choices and returns the best model.

Specifically:

- **Random search:** with `randomsearchcv` runs the search over some number of random parameter combinations
- **Grid search:** `gridsearchcv` runs the search over all parameter sets in the grid

Tuning models with scikit-learn is a good start but there are better options out there and they often have random search strategies anyway.

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

For both of those methods, scikit-learn trains and evaluates a model in a k fold cross-validation setting over various parameter choices and returns the best model.

Specifically:

- **Random search:** with `randomsearchcv` runs the search over some number of random parameter combinations
- **Grid search:** `gridsearchcv` runs the search over all parameter sets in the grid

Tuning models with scikit-learn is a good start but there are better options out there and they often have random

3. 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.

It has the pruning feature which automatically **stops the unpromising trails in the early stages of training**. Some of the key features provided by optuna are:

- Lightweight, versatile, and platform-agnostic architecture
- Pythonic search spaces
- Efficient optimization algorithms
- Easy parallelization
- Quick visualization

You can refer to the official documentation for tutorials on how to start using optuna.

May be useful

It has the pruning feature which automatically **stops the unpromising trails in the early stages of training**. Some of the key features provided by optuna are:

- Lightweight, versatile, and platform-agnostic architecture
- Pythonic search spaces
- Efficient optimization algorithms
- Easy parallelization
- Quick visualization

You can refer to the official documentation for tutorials on how to start using optuna.

Optuna hyperparameter optimization metadata logged to neptune.ai | **See the documentation**

4. Hyperopt

Hyperopt is one of the most popular hyperparameter tuning packages available. Hyperopt allows the user to describe a search space in which the user expects 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

4. Hyperopt

Hyperopt is one of the most popular hyperparameter tuning packages available. Hyperopt allows the user to describe a search space in which the user expects 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

the art optimization algorithms at scale.

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.

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

6. Keras Tuner

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:

- 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.

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

6. Keras Tuner

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*.

Visit the GitHub repo here to see it in action.

8. Metric Optimization Engine

MOE (Metric Optimization Engine) is an efficient way to optimize a system's parameters when evaluating parameters is time-consuming or expensive.

It is ideal for problems in which

- the optimization problem's objective function is a black box, not necessarily convex or concave,
- derivatives are unavailable,
- and we seek a global optimum, rather than just a local one.

This ability to handle black-box objective functions allows us to use MOE to optimize nearly any system, without

This ability to handle black-box objective functions allows us to use MOE to optimize nearly any system, without requiring any internal knowledge or access.

Visit the GitHub repo to read more about it.

9. Spearmint

It is ideal for problems in which

- the optimization problem's objective function is a black box, not necessarily convex or concave,
- derivatives are unavailable,
- and we seek a global optimum, rather than just a local one.

This ability to handle black-box objective functions allows us to use MOE to optimize nearly any system, without requiring any internal knowledge or access.

Visit the GitHub repo to read more about it.

Head over to its GitHub repo here.

11. SigOpt

SigOpt fully integrates automated hyperparameter tuning with training runs tracking to give you a sense of the bigger picture and the path to reach your best model.

With features like highly customizable search spaces and multimetric optimization, SigOpt can advance your model with a simple API for sophisticated hyperparameter tuning before taking it into production.

Visit the documentation here to learn more about SigOpt's hyperparameter tuning.

12. Fabolas

While traditional Bayesian hyperparameter optimizers model the loss of machine learning algorithms on a given dataset as a black box function to be minimized, FAst Bayesian Optimization on LArge data Sets (FABOLAS) models loss and computational cost across dataset size and uses these models to carry out Bayesian optimization with an extra degree of freedom.

bigger picture and the path to reach your best model.

With features like highly customizable search spaces and multimetric optimization, SigOpt can advance your model with a simple API for sophisticated hyperparameter tuning before taking it into production.

Visit the documentation here to learn more about SigOpt's hyperparameter tuning.

12. Fabolas

While traditional Bayesian hyperparameter optimizers model the loss of machine learning algorithms on a given dataset as a black box function to be minimized, FAst Bayesian Optimization on LArge data Sets (FABOLAS) models loss and computational cost across dataset size and uses these models to carry out Bayesian optimization with an extra degree of freedom.

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

XGBoost hyperparameter tuning

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

LightGBM hyperparameter tuning

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

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

LightGBM hyperparameter tuning

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

Final thoughts

Congratulations, you've made it to the end! Hyperparameter tuning represents an integral part of any Machine Learning project, so it's always worth digging into this topic. In this blog, we talked about different hyperparameter tuning algorithms and tools which are widely used and studied. But even though, we covered a good chunk of techniques and tools, as a wise man once said, there's no end to knowledge.

Here are some of the latest research happening in the area that might interest you:

- Improving Hyperparameter Optimization By Planning Ahead
- Hyperparameter Optimization: Foundations, Algorithms, Best Practices and Open Challenges
- Experimental Investigation And Evaluation Of Model-Based Hyperparameter Optimization

That's it for now, stay tuned for more, adios!

Congratulations, you've made it to the end! Hyperparameter tuning represents an integral part of any Machine Learning project, so it's always worth digging into this topic. In this blog, we talked about different hyperparameter tuning algorithms and tools which are widely used and studied. But even though, we covered a good chunk of techniques and tools, as a wise man once said, there's no end to knowledge.

Here are some of the latest research happening in the area that might interest you:

- Improving Hyperparameter Optimization By Planning Ahead
- Hyperparameter Optimization: Foundations, Algorithms, Best Practices and Open Challenges
- Experimental Investigation And Evaluation Of Model-Based Hyperparameter Optimization

That's it for now, stay tuned for more, adios!

Natural Language Processing | Reinforcement Learning | Tabular Data | Time Series

Manage your model metadata in a single place

Join 30,000+ ML Engineers & Data Scientists using Neptune to easily log, compare, register, and share ML metadata.

Computer Vision | General | Machine Learning Tools | ML Model Development | MLOps

Natural Language Processing | Reinforcement Learning | Tabular Data | Time Series

Manage your model metadata in a single place

Join 30,000+ ML Engineers & Data Scientists using Neptune to easily log, compare, register, and share ML metadata.

Try Neptune for free | Check out the Docs

Take an interactive product tour →

Top MLOps articles, case studies, events (and more) in your inbox every month.

Your e-mail

Get Newsletter

Overview
Resources
Pricing
Deployment options
Roadmap
Service status

SOLUTIONS

ML Platform Engineer
Data Scientist
ML Engineer
ML Team Lead
Enterprise

Quickstart
Neptune Docs
Neptune Integrations

Neptune vs Weights & Biases
Neptune vs MLflow
Neptune vs TensorBoard
Other Comparisons
ML Experiment Tracking Tools
ML Metadata Stores
ML Model Registries

MLOps Blog
ML Platform Podcast
MLOps Newsletter

About us
Customers
Careers
Security portal and SOC 2

Comparisons

SOLUTIONS

ML Platform Engineer
Data Scientist
ML Engineer
ML Team Lead
Enterprise

ML Experiment Tracking Tools
ML Metadata Stores
ML Model Registries

The Best MLOps Tools • MLOps at a Reasonable Scale • ML Metadata Store • MLOps: What, Why, and How • Experiment Tracking in Machine Learning

Terms of Service | Privacy Policy | SLA