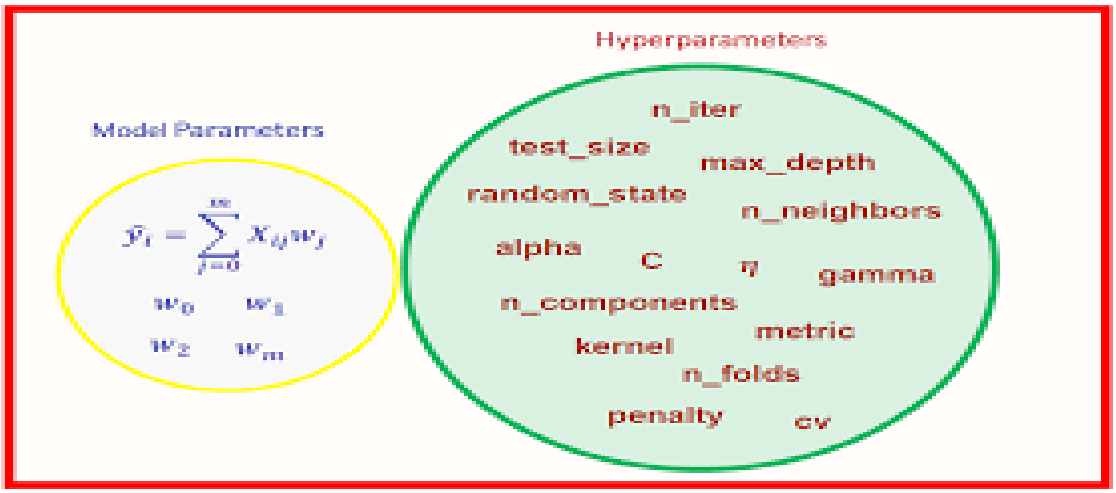
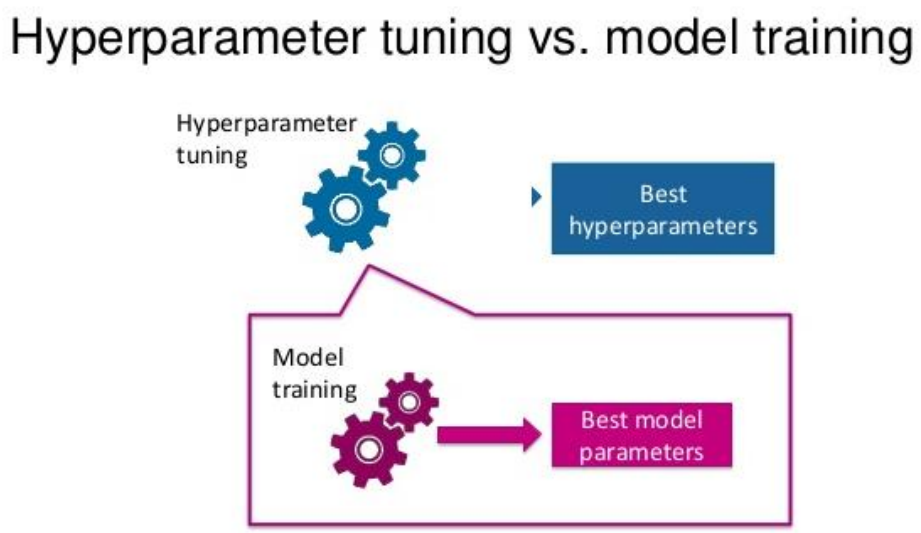
Top 8 Approaches For Tuning Hyperparameters Of Machine Learning Models

Hyperparameter Tuning is one of the fundamental steps in the machine learning routine. Also known as hyperparameter optimisation, the method entails searching for the best configuration of hyperparameters to enable optimal performance.



Machine learning algorithms require user-defined inputs to achieve a balance between accuracy and generalisability. This process is known as hyperparameter tuning. There are various tools and approaches available to tune hyperparameters.



1| Bayesian Optimisation

Bayesian Optimisation has emerged as an efficient tool for hyperparameter tuning of machine learning algorithms, more specifically, for complex models like deep neural networks. It offers an efficient framework for optimising the highly expensive black-box functions without knowing its form. It has been applied in several fields including learning optimal robot mechanics, sequential experimental design, and synthetic gene design.

2| Evolutionary Algorithms

Evolutionary algorithms (EA) are optimisation algorithms that work by modifying a set of candidate solutions (population) according to certain rules called Operators. One of the main advantages of the EA is their generality: Meaning EA can be used in a broad range of conditions due to their simplicity and independence from the underlying problem. In hyperparameter tuning problems, evolutionary algorithms have proved to perform better than grid search techniques based on an accuracy-speed ratio.

3| Gradient-Based Optimisation

Gradient-based optimisation is a methodology to optimise several hyperparameters, based on the computation of the gradient of a machine learning model selection criterion with respect to the hyperparameters. This hyperparameter tuning

methodology can be applied when some differentiability and continuity conditions of the training criterion are satisfied.

4| Grid Search

Grid search is a basic method for hyperparameter tuning. It performs an exhaustive search on the hyperparameter set specified by users. This approach is the most straightforward leading to the most accurate predictions. Using this tuning method, users can find the optimal combination. Grid search is applicable for several hyper-parameters, however, with limited search space.

5| Keras' Tuner

Keras tuning is a library that allows users to find optimal hyperparameters for machine learning or deep learning models. The library helps to find kernel sizes, learning rate for optimisation, and different hyper-parameters. Keras tuner can be used for getting the best parameters for various deep learning models for the highest accuracy.

6| Population-based Optimisation

Population-based methods are essentially a series of random search methods based on

genetic algorithms, such as evolutionary algorithms, particle swarm optimisation, among others. One of the most widely used population-based methods is population-based training (PBT), proposed by DeepMind. PBT is a unique method in two aspects:

- It allows for adaptive hyper-parameters during training
- It combines parallel search and sequential optimisation

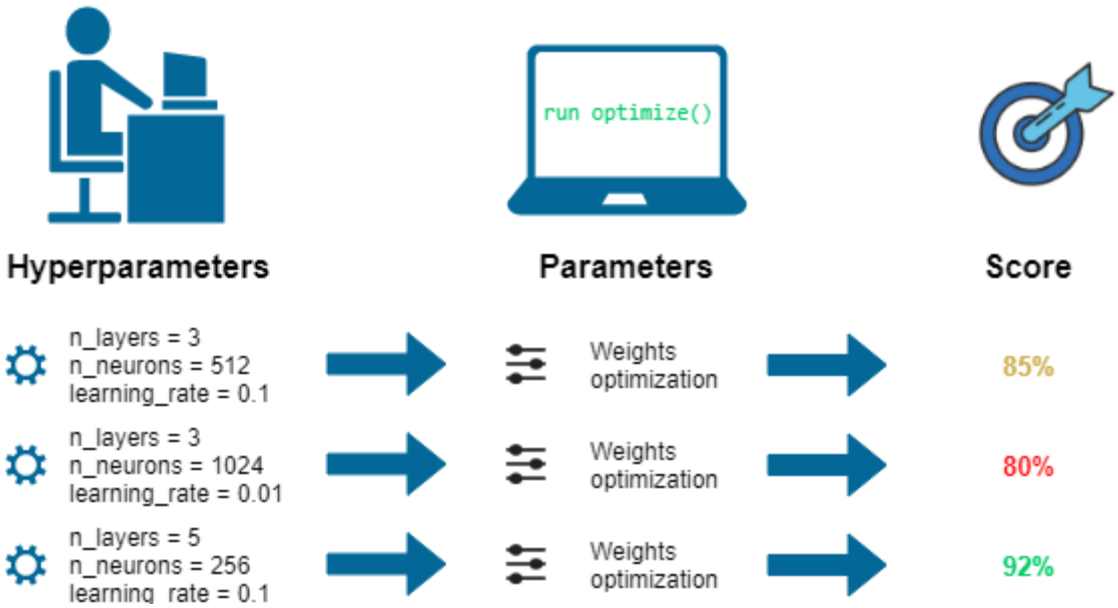
7| ParamILS

ParamILS (Iterated Local Search in Parameter Configuration Space) is a versatile stochastic local search approach for automated algorithm configuration. ParamILS is an automated algorithm configuration method that helps dvelop high-performance algorithms and their applications.

ParamILS uses default and random settings for initialisation and employs iterative first improvement as a subsidiary local search procedure. It also uses a fixed number of random moves for perturbation and always accepts better or equally-good parameter configurations, but re-initialises the search at random with probability.

8| Random Search

Random search can be said as a basic improvement on grid search. The method refers to a randomised search over hyper-parameters from certain distributions over possible parameter values. The searching process continues till the desired accuracy is reached. Random search is similar to grid search but has proven to create better results than the latter. The approach is often applied as the baseline of HPO to measure the efficiency of newly designed algorithms. Though random search is more effective than grid search, it is still a computationally intensive method.



Hyperparameter types:

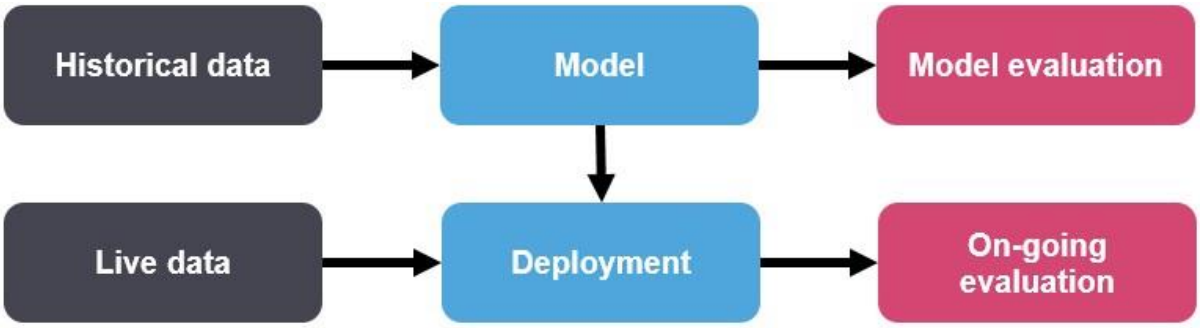
- K in K-NN
- Regularization constant, kernel type, and constants in SVMs
- Number of layers, number of units per layer, regularization in neural network

Generalization (test) error of learning algorithms has two main components:

- Bias: error due to simplifying model assumptions
- Variance: error due to randomness of the training set

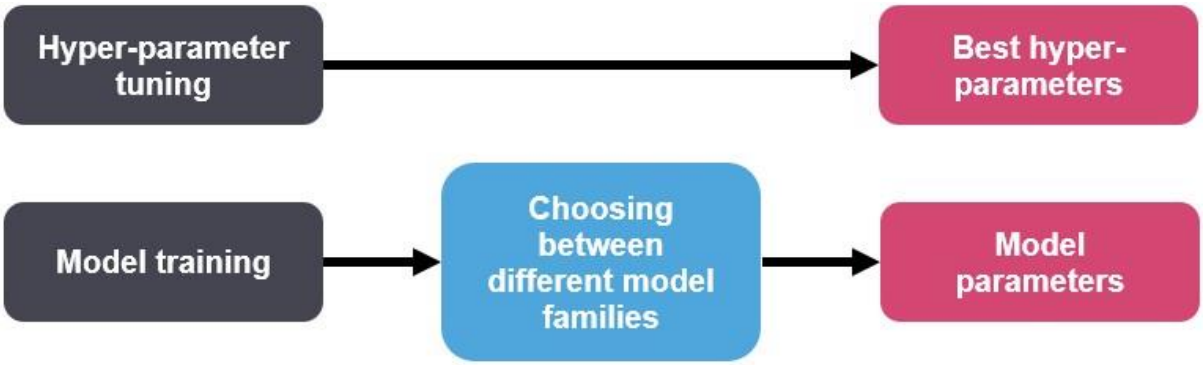
The trade-off between these components is determined by the complexity of the model and the amount of training data. The optimal hyperparameters help to avoid under-fitting (training and test error are both high) and over-fitting (Training error is low but test error is high)

Workflow: One of the core tasks of developing an ML model is to evaluate its performance. There are multiple stages in developing an ML model for use in software applications.



Evaluation: Model evaluation and ongoing evaluation may have different matrices. For example, model evaluation may include Accuracy or AUROC and ongoing evaluation may include customer lifetime value. Also, the distribution of the data might change between the historical data and live data. One way to detect distribution drift is through continuous model monitoring.

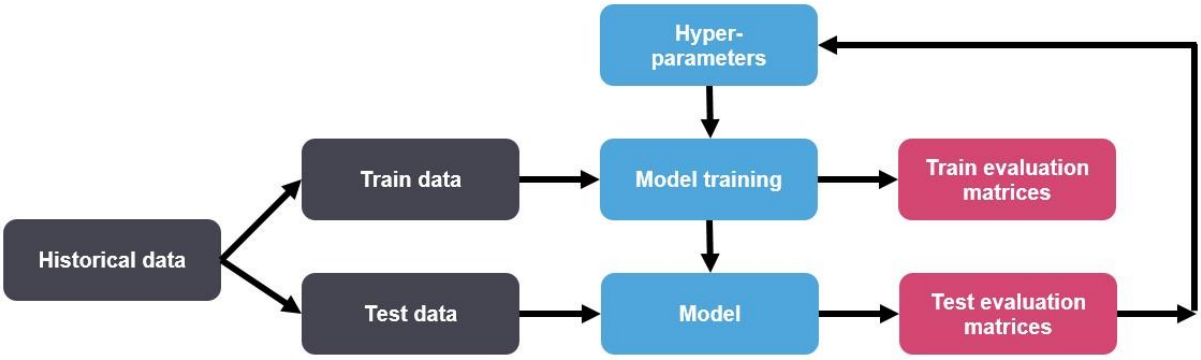
Hyper-parameters: Model parameters are learned from data and hyper-parameters are tuned to get the best fit. Searching for the best hyper-parameter can be tedious, hence search algorithms like grid search and random search are used.



Model Evaluation

Evaluation Matrices: These are tied to ML tasks. There are different matrices for supervised algorithms (classification and regression) and unsupervised algorithms. For example, the performance of classification of the binary class is measured using Accuracy, AUROC, Log-loss, and KS.

Evaluation Mechanism: Model selection refers to the process of selecting the right model that fits the data. This is done using test evaluation matrices. The results from the test data are passed back to the hyper-parameter tuner to get the most optimal hyperparameters.



Hyperparameter Tuning

Hyperparameters: Vanilla linear regression does not have any hyperparameters. Variants of linear regression (ridge and lasso) have regularization as a hyperparameter. The decision tree has max depth and min number of observations in leaf as hyperparameters.

Optimal Hyperparameters: Hyperparameters control the over-fitting and under-fitting of the model. Optimal hyperparameters

often differ for different datasets. To get the best hyperparameters the following steps are followed:

1. For each proposed hyperparameter setting the model is evaluated

2. The hyperparameters that give the best model are selected.

Hyperparameters Search: Grid search picks out a grid of hyperparameter values and evaluates all of them. Guesswork is necessary to specify the min and max values for each hyperparameter. Random search randomly values a random sample of points on the grid. It is more efficient than grid search. Smart hyperparameter tuning picks a few hyperparameter settings, evaluates the validation matrices, adjusts the hyperparameters, and re-evaluates the validation matrices. Examples of smart hyper-parameter are Spearmint (hyperparameter optimization using Gaussian processes) and Hyperopt (hyperparameter optimization using Tree-based estimators).

ML Algorithms	Hyperparameters	Definition	Defined Parameters
SVM (support vector machines)	Kernel type	the kernel function	RBF
	C	the penalty parameter	0.01–100
	σ	the bandwidth parameter	0.01–100
Cubist (regression tree)	committees	the number of model trees	1–100
	neighbors	the number of nearest neighbors	0–9
XGBoost (extreme gradient boosting)	booster	the type of model	gbtree
	max_depth	the depth of tree	3–10
	min_child_weight	the minimum sum of weights of all observations	0–5
	colsample_bytree	the number of variables supplied to a tree	0.5–1
	subsample	the number of samples supplied to a tree	0.5–1
	eta	learning rate	0.01–0.5
RF (random forest)	Mtry	the number of input variables	1–30
	Ntree	the number of trees	100–3000
ANN (artificial neural networks)	decay	learning rate	0.001–0.05
	size	the number of neurons in the hidden layer	1–10
DNN	hidden	the number of hidden layers	2–10