**1. What is Hyperparameter optimization and why do we need it?**

* Hyperparameter optimization finds a combination of hyperparameters that returns an optimal model which reduces a predefined loss function and in turn increases the accuracy on given independent data.
* To tune α to its most efficient value, and this process of optimization is called hyperparameter tuning.
* α is called the learning rate in the gradient descent algorithm. It controls the rate by which loss reaches its minima.

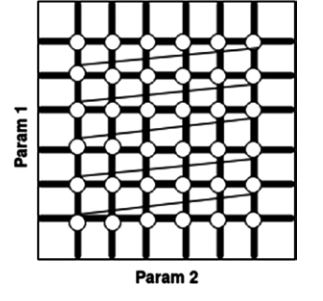
**Why do we need it?**

* hyperparameters are set before any Machine learning algorithm is run, hence, it becomes very essential to set an optimal value of hyperparameters as it effects the convergence of any algorithm to a large extent.
* Parameters are constant or variable terms in any function that facilitates the specific form of the function. These parameters are a part of the Machine Learning algorithm and its optimal values are found within the algorithm whereas hyperparameters are constants or variables that are set before the algorithm is run.
* **For example**, learning rate, penalty, C in Logistic regression, number of estimators, min samples split, etc. in Random Forest and Decision Trees, etc. are all examples of hyperparameters while values of coefficients of x’s in linear and logistic regression are examples of parameters.

**2. List out other types of Hyperparameter Optimization**

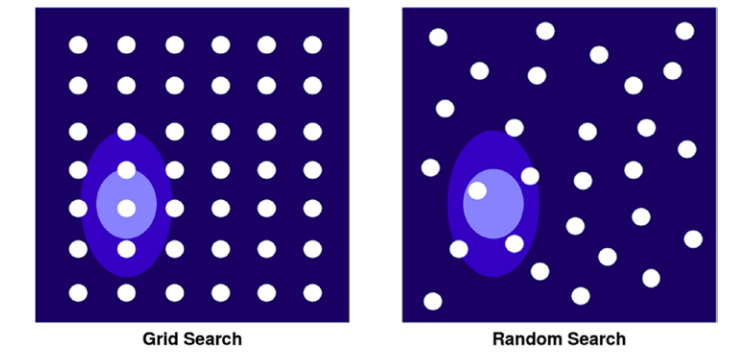
**Grid Search**

* Perhaps the most brute-force approach for finding the most optimized set of hyperparameters is to train the dataset on each possible set. This approach, called grid search, is the most certain way of finding the best set of hyperparameters, but it also has its disadvantages
* Figure depicts a grid going through all possible combinations of parameters 1 and 2



**Random Search**

* Grid search eventually finds the near optimal set of hyperparameters, but its time and resource consumption is high. Another method, random search, consumes less time and resources. It randomly picks hyperparameters, makes a set, and trains the model on it. This method may not find the best set, but there are higher chances of finding a near best set saving a huge amount of time.
* Unlike grid search, instead of spending a large amount of time on unpromising candidates, random search jumps to random hyperparameters, and even though it does not learn from its past results, it usually delivers satisfactory results. In random search, we define the number of trials, which is the number of sets of hyperparameters to be tried. Let’s see how random search can be better than grid search by exploring the example shown in Figure.



**Manual Search**

* It is an ad-hoc approach to find the best values of hyperparameters for any machine learning algorithm. The idea is to first take big jumps in values and then small jumps to focus around a specific value which performed better.
* For example, in the Random Forest algorithm, the **n\_estimators** is the number of trees to grow. We can find the best value of this parameter by starting with big values like 100, 200, 500, 1000. and then once you know which one of them gave the best accuracy outcome you can choose to try values around that. e.g. if the accuracy was best around 500, then keep trying the values around it like 480, 490, 500, 510, 520, etc. Then choose whichever value gives the highest accuracy amongst it.

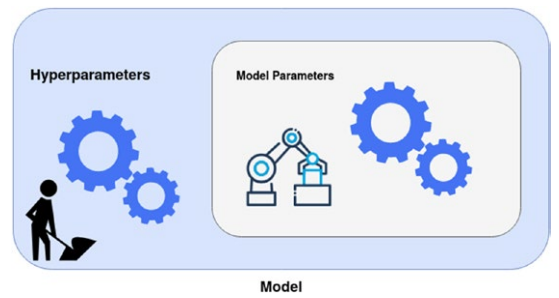
**3.What are random search and their drawbacks?**

* [Random search](https://en.wikipedia.org/wiki/Random_search) is also referred to as random optimization or random sampling.
* Random search involves generating and evaluating random inputs to the objective function. It’s effective because it does not assume anything about the structure of the objective function. This can be beneficial for problems where there is a lot of domain expertise that may influence or bias the optimization strategy, allowing non-intuitive solutions to be discovered.
* Random search may also be the best strategy for highly complex problems with noisy or non-smooth (discontinuous) areas of the search space that can cause algorithms that depend on reliable gradients.
* We can generate a random sample from a domain using a [pseudorandom number generator](https://machinelearningmastery.com/how-to-generate-random-numbers-in-python/). Each variable requires a well-defined bound or range and a uniformly random value can be sampled from the range, then evaluated.
* Generating random samples is computationally trivial and does not take up much memory, therefore, it may be efficient to generate a large sample of inputs, then evaluate them. Each sample is independent, so samples can be evaluated in parallel if needed to accelerate the process.

**Drawbacks?**

* The drawback of random search is that it yields high variance during computing.
* Since the selection of parameters is completely random; and since no intelligence is used to sample these combinations, luck plays its part.

**4. What is the difference between parameter and hyperparameter?**

****

* The box inside represents model parameters, where the machine learning algorithm is at work. The outer box represents the hyperparameters, which we have to set before algorithm starts training.

**Parameter**

* A parameter is a configuration variable that is internal to the model and whose value can be estimated from data.
* They are required by the model when making predictions.
* They values define the skill of the model on your problem.
* They are estimated or learned from data.
* They are often not set manually by the practitioner.
* They are often saved as part of the learned model.

**Some examples of parameters include:**

* The weights in an artificial neural network.
* The support vectors in a [support vector machine](https://machinelearningmastery.com/support-vector-machines-for-machine-learning/).
* The coefficients in a linear regression or logistic regression.

**Hyperparameter**

hyperparameter is a configuration that is external to the model and whose value cannot be estimated from data.

* They are often used in processes to help estimate model parameters.
* They are often specified by the practitioner.
* They can often be set using heuristics.
* They are often tuned for a given predictive modeling problem.

Some examples of model hyperparameters include:

* The learning rate for training a neural network.
* The C and sigma hyperparameters for support vector machines.
* The k in k-nearest neighbours.

**5.What are the examples of Hyperparameter tuning?**

Examples of model hyperparameters include:

* The penalty in Logistic Regression Classifier i.e. L1 or L2 regularization
* The learning rate for training a neural network.
* The C and sigma hyperparameters for support vector machines.
* The k in kNN or K-Nearest Neighbour algorithm
* Learning rate for training a neural network
* Train-test split ratio
* Batch Size
* Number of Epochs
* Branches in Decision Tree
* Number of clusters in Clustering Algorithm