WELCOME TO

TECH I.S.

Hyperparameter Optimization

Parameters Types

<u>Model parameters</u>: Parameters learned by fitting model on training set

- Example: Parameters m and x in a Linear Regression model etc.

<u>Model hyperparameters</u>: Parameters set by the user before training, not changed when fitting to data.

- Example: Learning rate in gradient descent, L1 (Lasso) or L2 (Ridge) penalty term in a loss function etc.

<u>Model performance strongly depends on hyperparameters.</u>

Hyperparameter Optimization: Motivation

Parameters are learned by applying Gradient Descent to an optimization problem - so why not the same for hyperparameters?

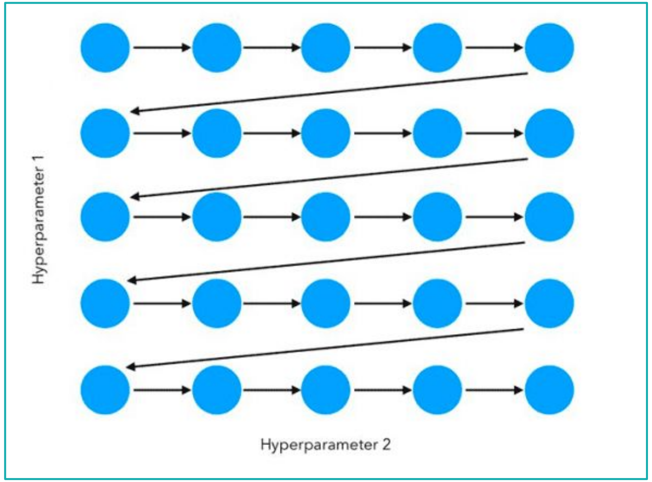
- Because hyperparameters are not differentiable:
  - If Hyperparameter is an integer - eg: number of neuron in NN.
  - If Hyperparameter is a categorical variable - eg: type of kernel function in SVM.
  - If Hyperparameter is continuous, but differentiation is impossible - eg: dropout rate.

Our goal: Build automatic techniques to Search Hyperparameters.

Exhaustive Search - Grid

<u>Brute-Force approach:</u> Try all possible combinations of hyperparameters.

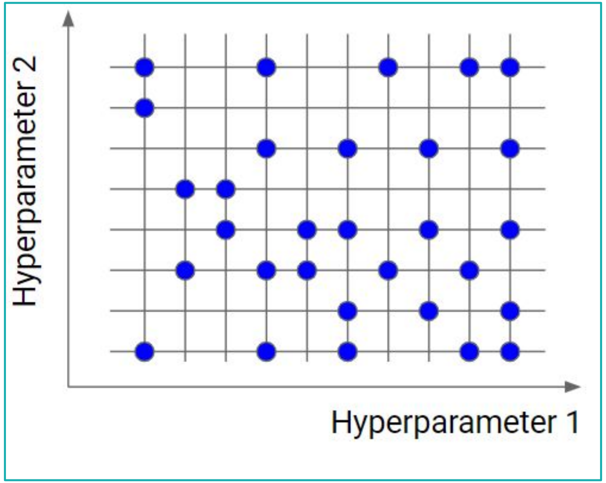
- <u>Advantage:</u> Easy to implement.
- <u>Disadvantage</u>: As the number of parameters increases, the cost of grid search increases exponentially!



Randomized Search

Monte Carlo Search Approach: Randomly chose points instead of all points on a grid.

- **Advantage:** Solves the curse of dimensionality.
- **Disadvantage**: Because of random search we are not guaranteed to obtain optimal parameters.



How to evaluate the hyperparameters?

Unlike the model parameters, we're not given a loss function so we can use the following approaches:

- Multiple rounds of cross-validation can be performed to get a very good sense of how good the hyperparameters are but at a significant computational cost!
- Evaluating the System Cost: In practice we don't just care about the statistics but also - Time and Energy Cost.

Much obliged.

TECH I.S.