# Multi-fidelity optimization

1. Search strategies

* Random search – computation intensive
* Bayesian optimization – time intensive
* Both strategies aim to find the best hyperparams, x, that minimize f(x)
* To find the ‘global’ minimum we require a good coverage of f(x)
* As x dimension increases, the number of evaluations needed to cover f(x) increases exponentially
* More computing resource or time
* Not practical

1. Time budget

* In practice, one constraint for optimization techniques is the **time budget**
* The aim of the optimization algorithm is to select the hyperparam that can achieve **(near)-optimal** performance **within a defined time budget (or cost**)

1. Black-box strategies

* Black-box optimization
* Grid search
* Random search
* Bayesian optimization
* Simulated annealing
* Genetic algorithms
* We can only know f(x) after we have **fully evaluated** it at certain values of the hyperparams
* F(x) is often expensive to obtain
* Often, no getting around the fact that a thorough search of f(x) will require sampling a large number of xs
* What if, we could use a cheaper version of f(x)?

1. Multi-fidelity

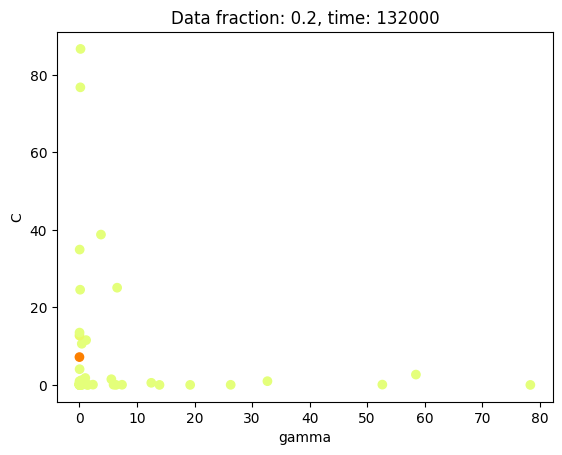
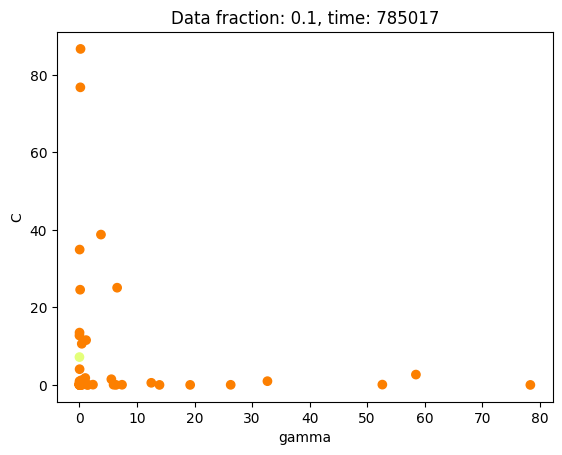
* Multi-fidelity optimization focuses on decreasing the evaluation cost by combining **large number** of **cheap low-fidelity evaluations** and **a small number** of **expensive high-fidelity evaluations**
* The **high-fidelity** evaluation outputs **precise performance** at a higher cost
* The **low-fidelity evaluations** output a **(good) approximation** of the best f(x) at a much cheaper cost

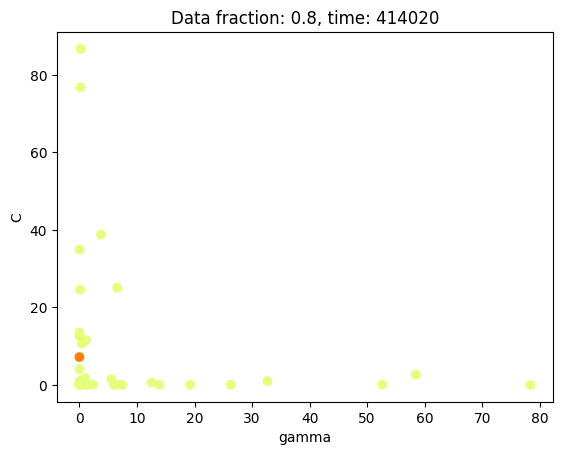
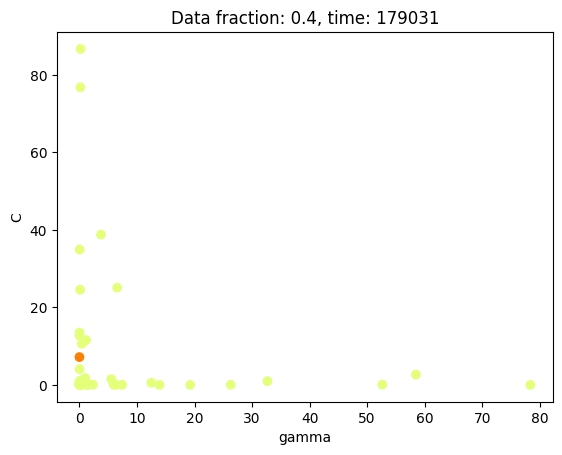
1. Multi-fidelity optimization

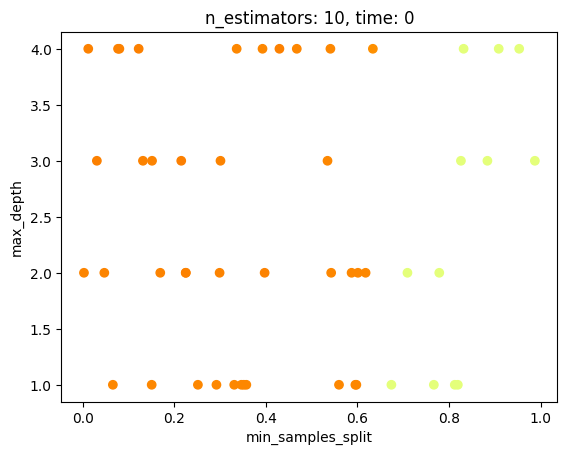
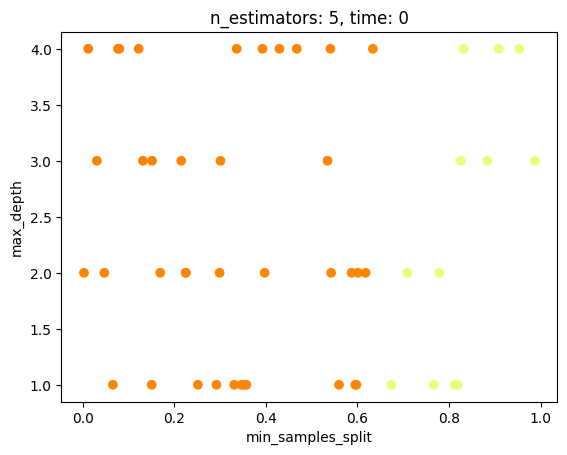
* Uses many low-fidelity models to **reduce the total evaluation cost**, find a subset of potentially good solutions, and then, examine the best solutions with high-fidelity model(s)
* Many low-fidelity models + Few high-fidelity models = Optimal hyperparam at a reduced cost

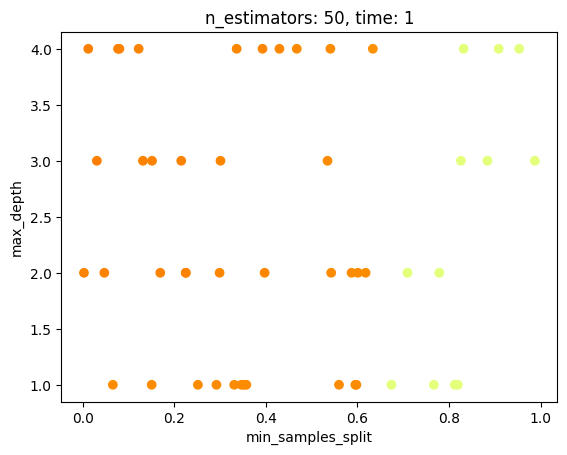
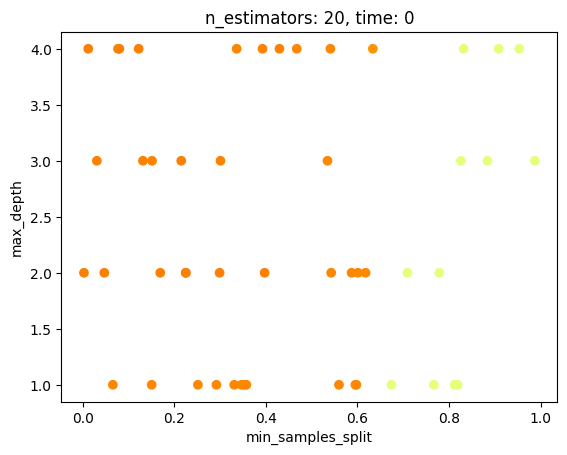
1. Low fidelity models

* Low fidelity models are cheaper representations of f(x)
* Reduced dataset
* Reduced number of features
* Early stopping
* Cheaper value of a hyperparam
* Created to reduce computational cost when a large number of expensive simulations
* Low fidelity models approximate f(x):









* With low-fidelity models we can already detect where the hyperparams are at a fraction of the time

1. Multi-fidelity optimization methods

* Successive halving – sklearn
* Hyperband – Keras tuner
* BOHB – Ray tune