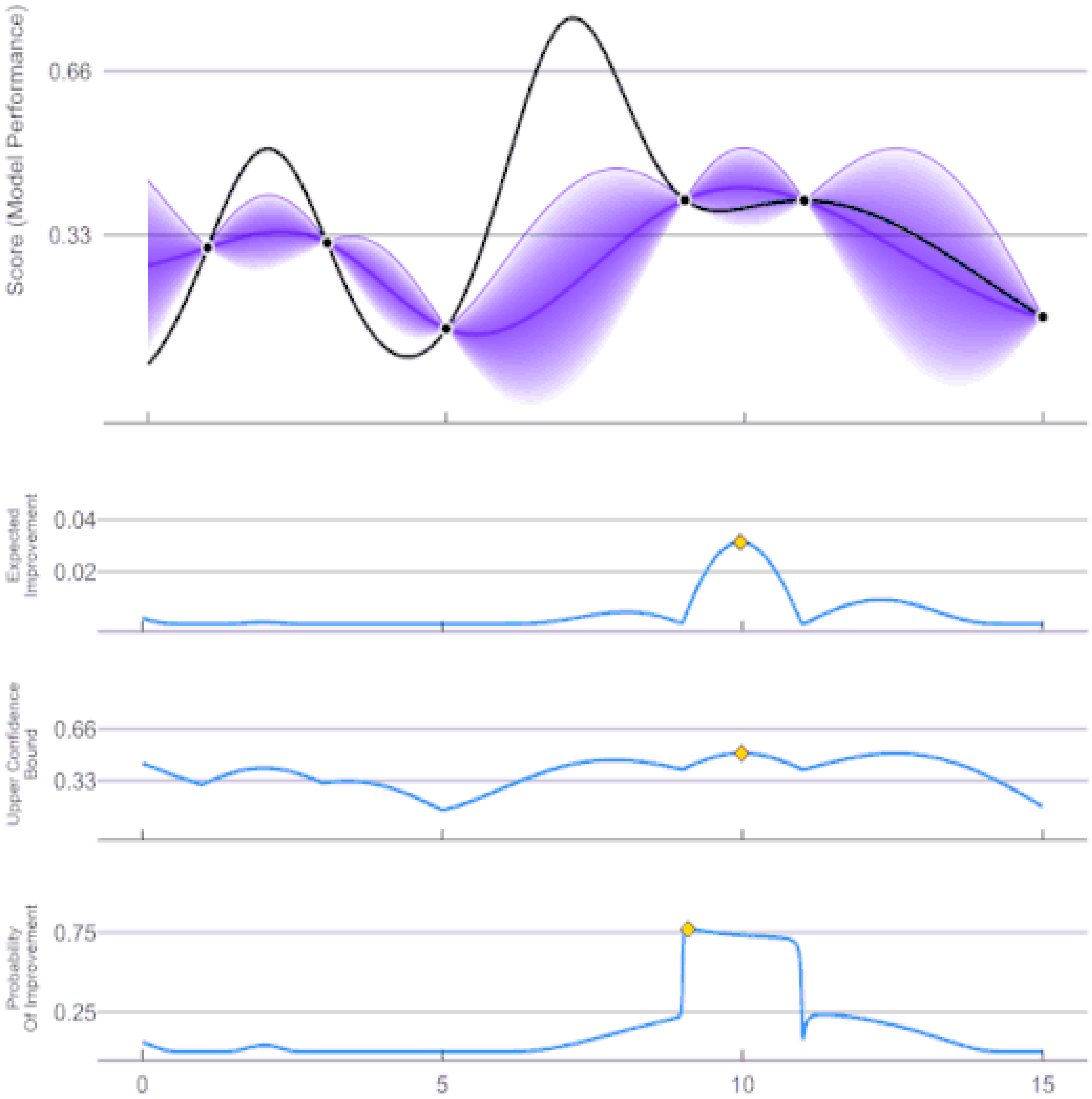


Bayesian Optimization

Feb.19.2020

Team . A

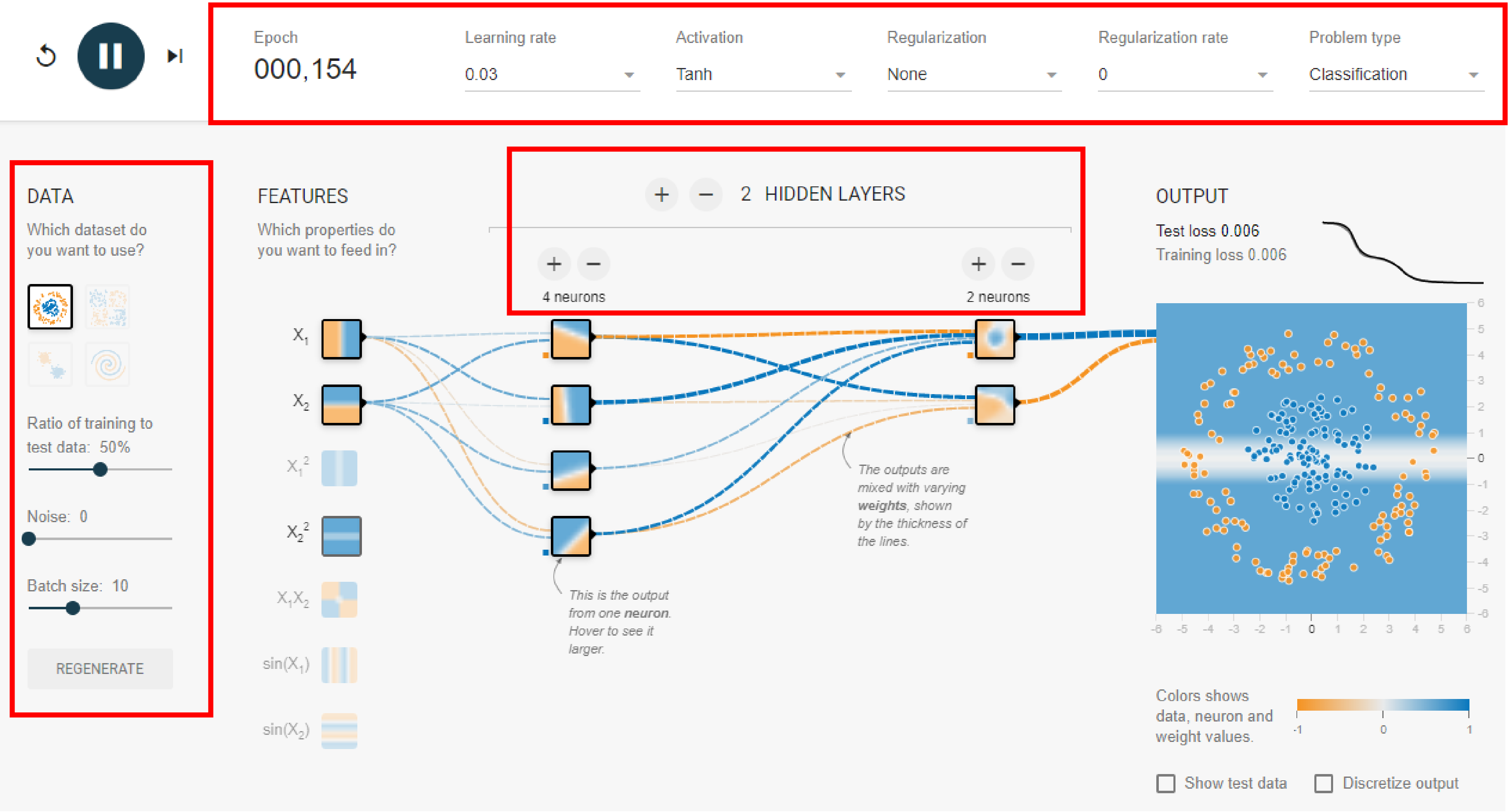
Kwangwoon University ML Study

Week #11

Machine Learning is
Extremely powerful

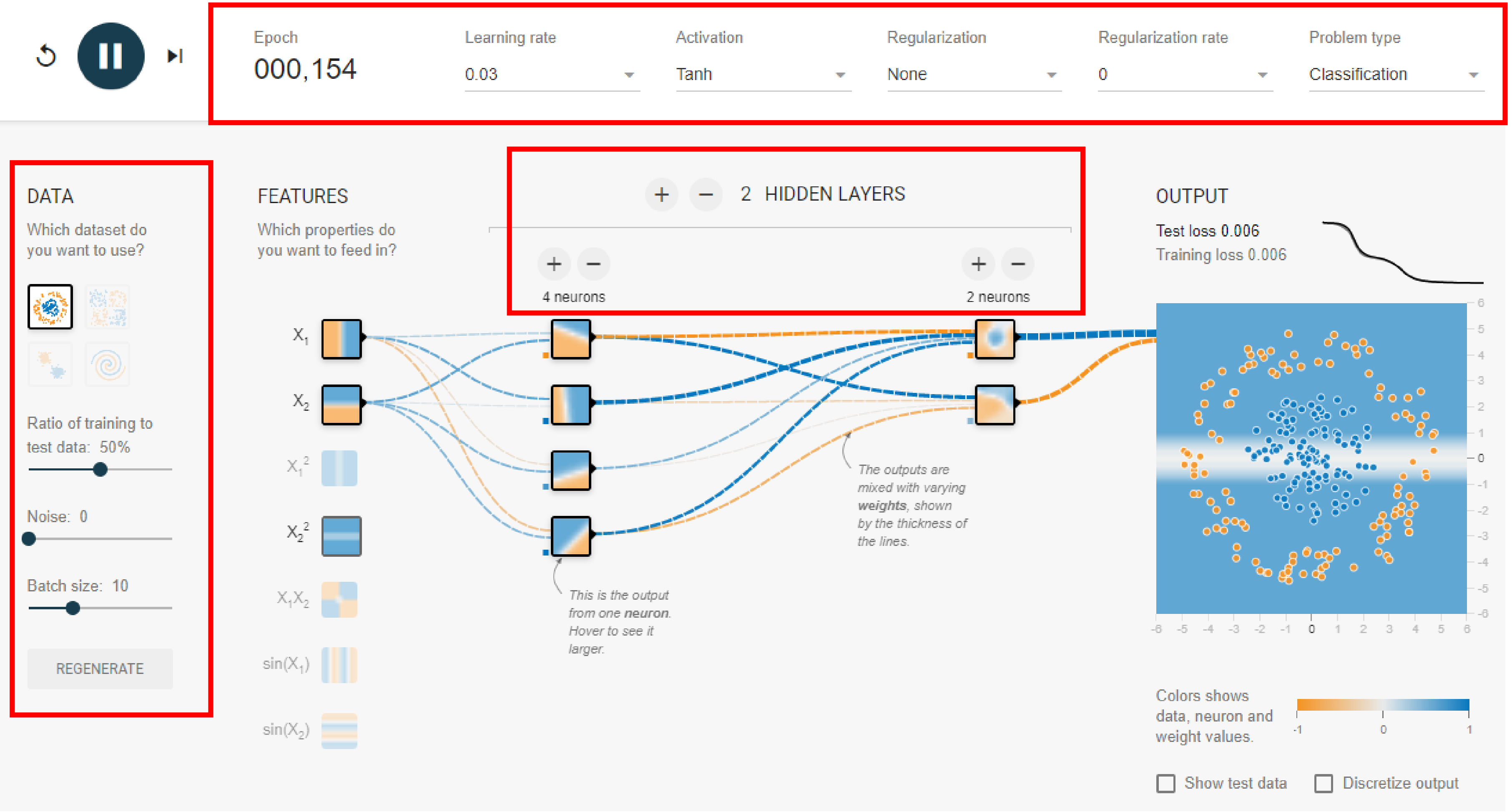
But Tuning Machine Learning system is
Extremely non-intuitive

Bayesian Optimization



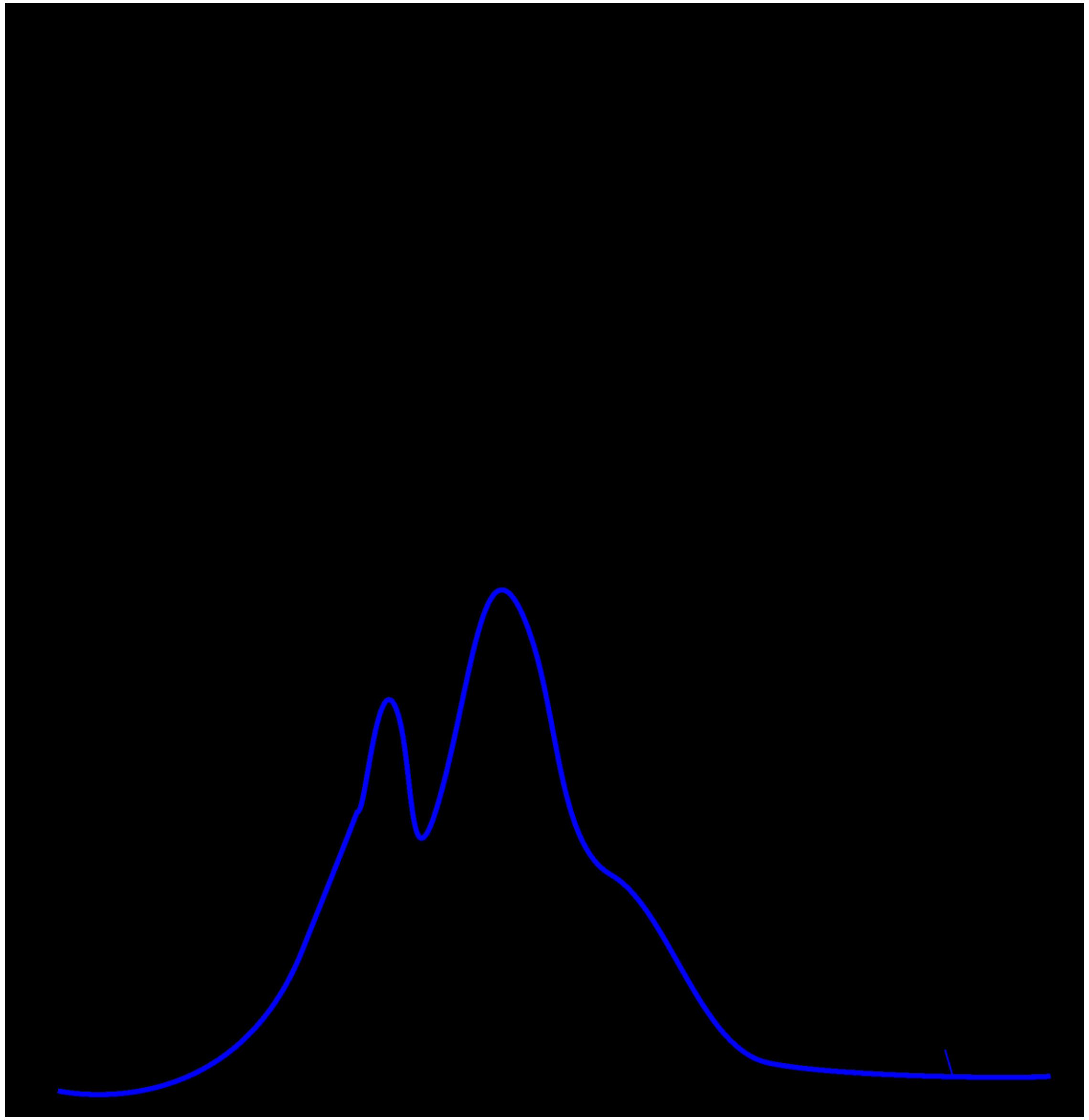
https://playground.tensorflow.org/

Bayesian Optimization



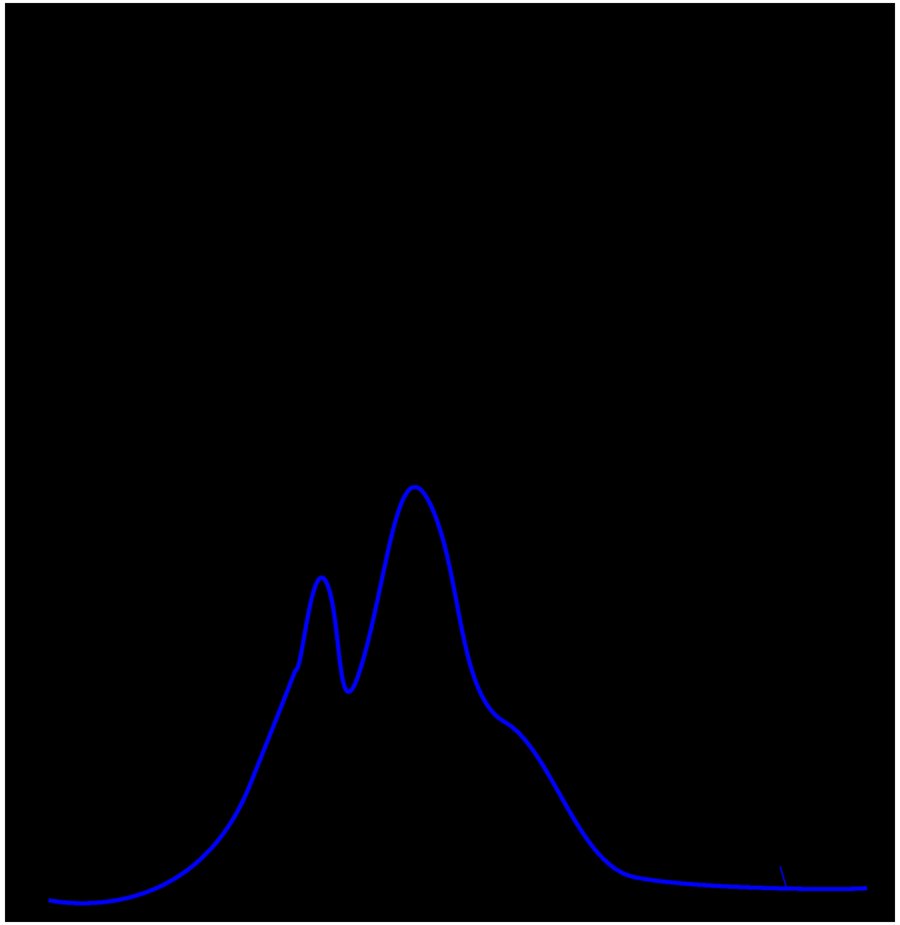
Many Tunable parameters in Machine learning

Bayesian Optimization



Manual Search

Select hyperparameter manually
with using prior knowledge
and intuition

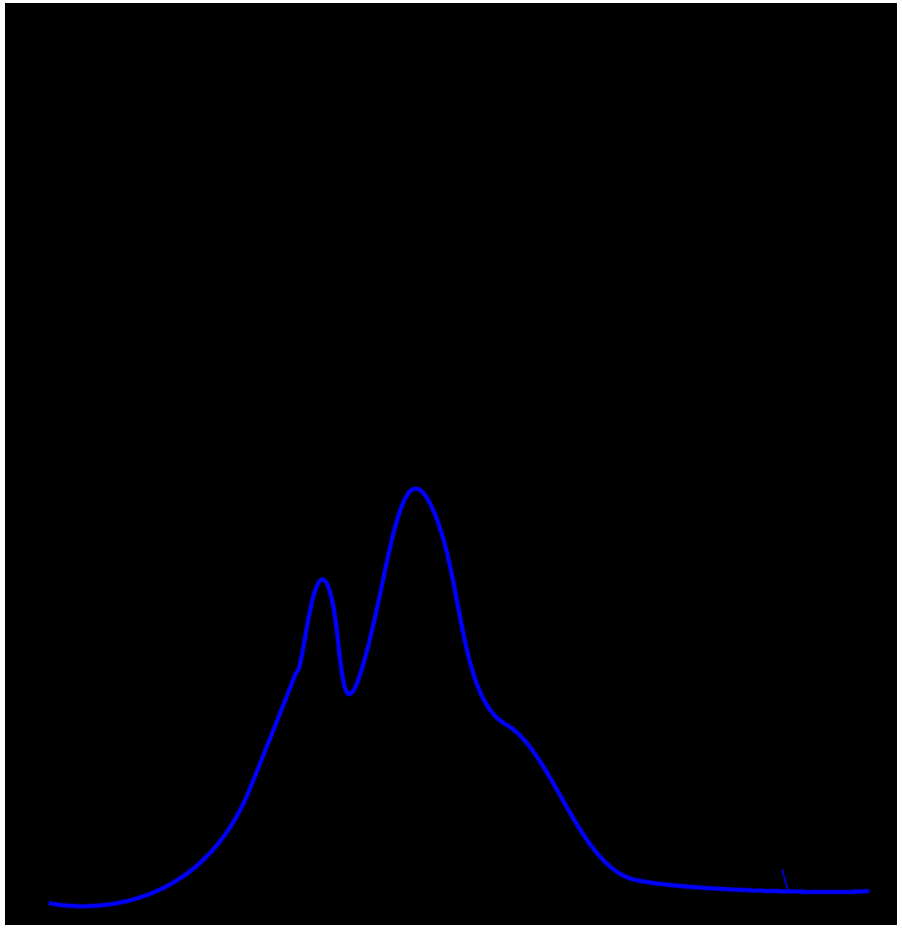


Manual Search

Problem

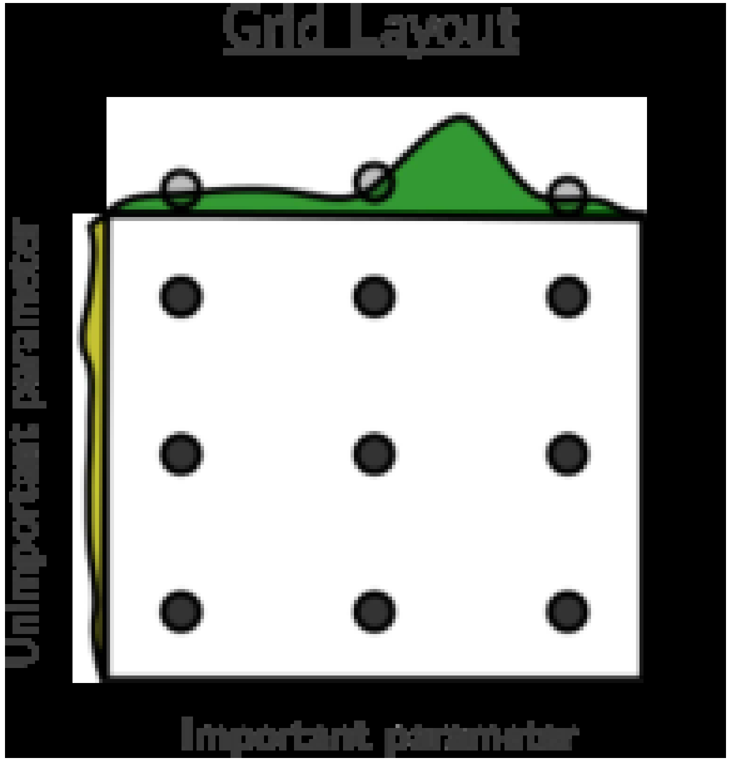
1. It doesn't guarantee that the solution is truly optimal.

2. As the number of hyperparameters increases, the problem becomes more complex



Grid Search

Select all hyperparameter within the certain section at regular intervals

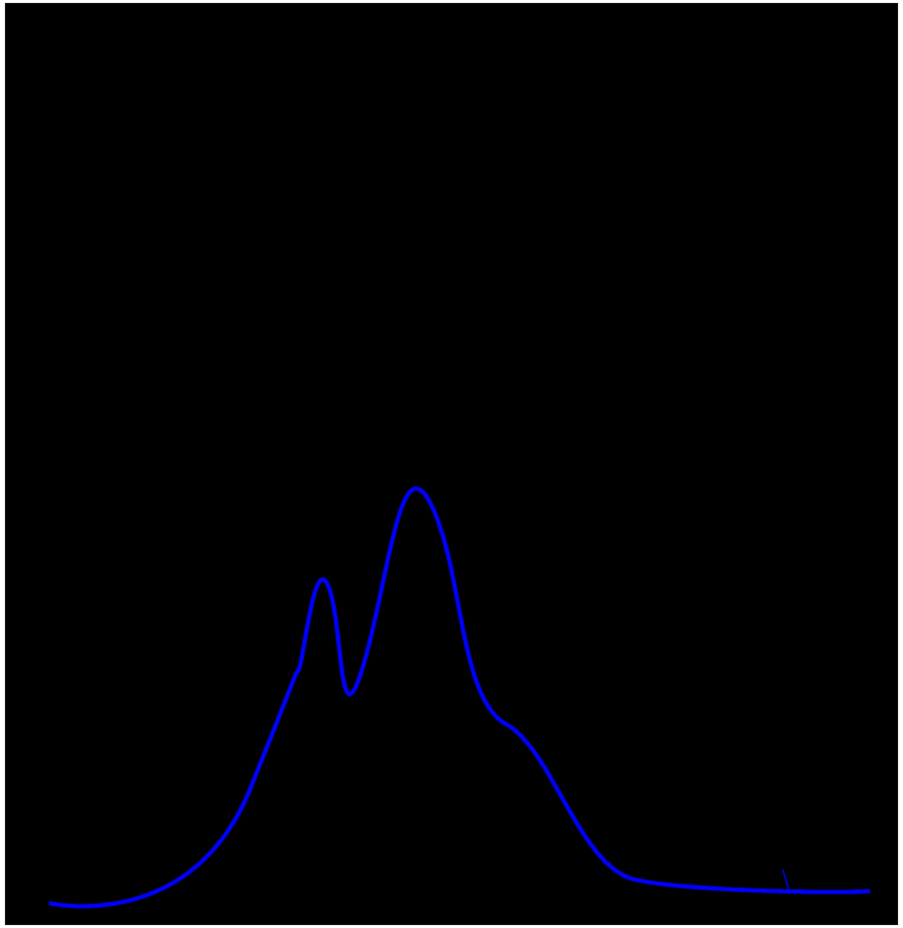


Grid Search

Problem

1. As the number of hyperparameters increases, the problem becomes more complex

2. There is no exploitation



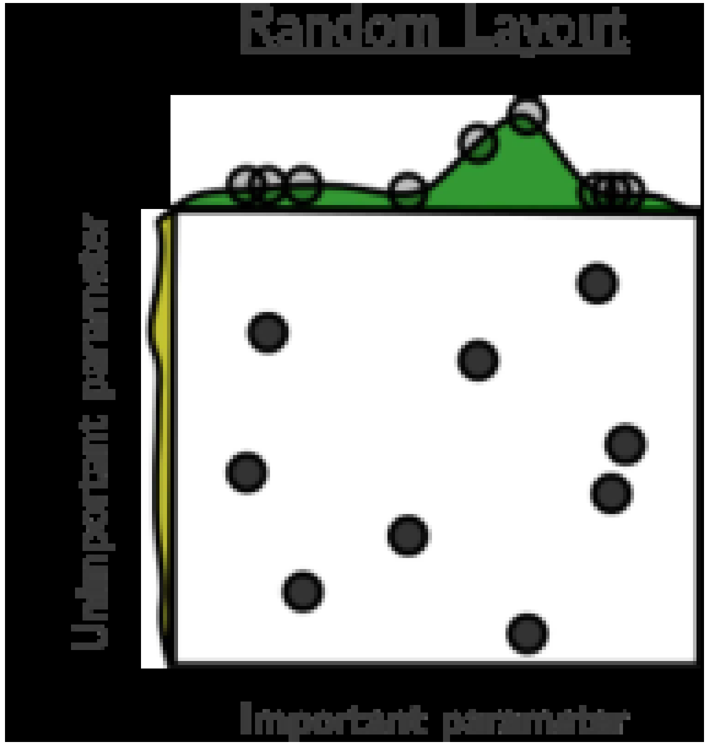
Random Search

Select hyperparameter
by random sampling

Advantage

1. Reduce the number of iteration

2. The values located between grids can also be stochastically searched

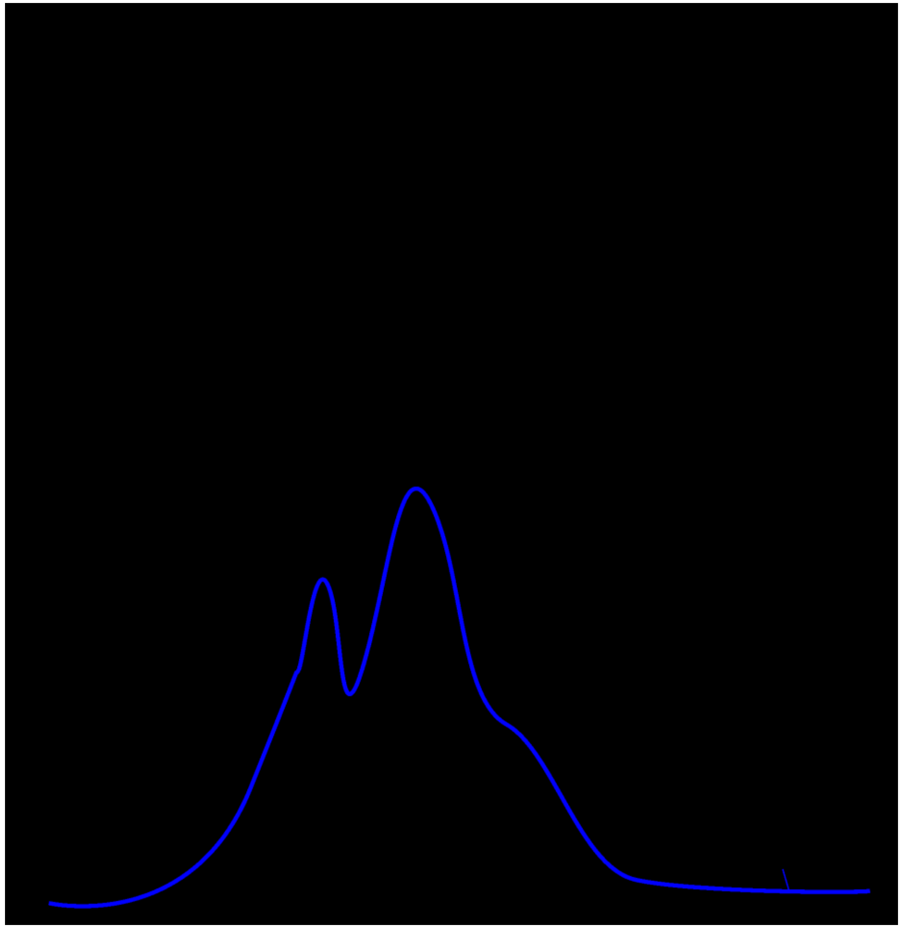


Random Search

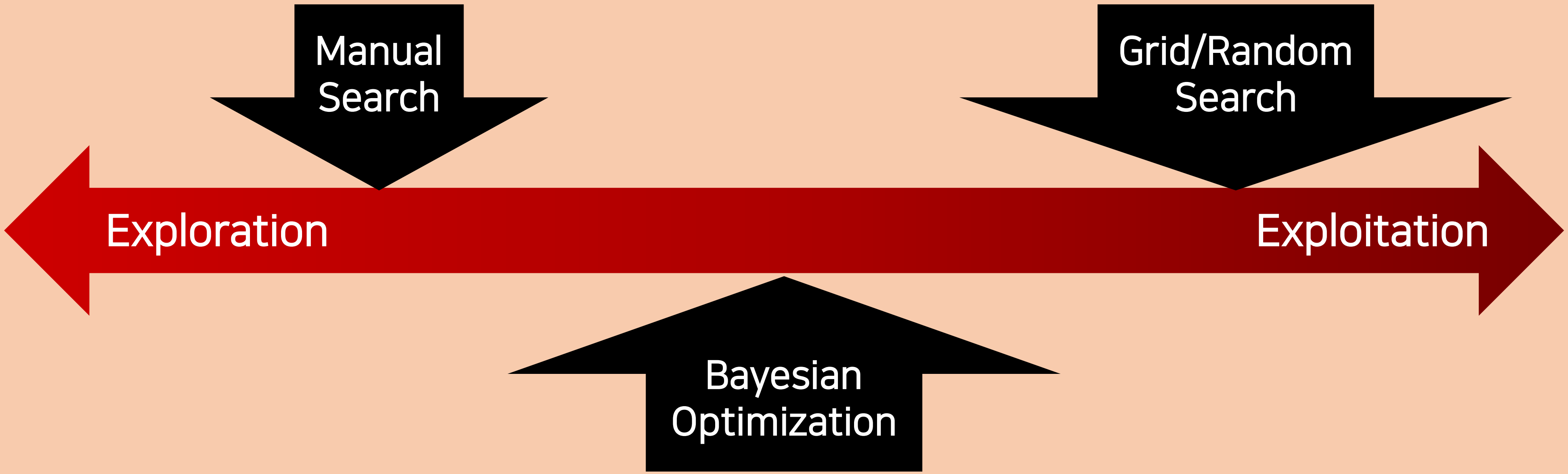
Problem

1. As the number of hyperparameters increases, the problem becomes more complex

2. There is no exploitation



Exploration-Exploitation Dilemma



Common Approach

1. Random search or grid search
2. Expert defined grid search near “good” points
3. Refine domain and repeat steps - “grad student descent”

Problem

- Expert intensive
- Computationally intensive
- Finds potentially local optima
- Does not fully exploit useful information

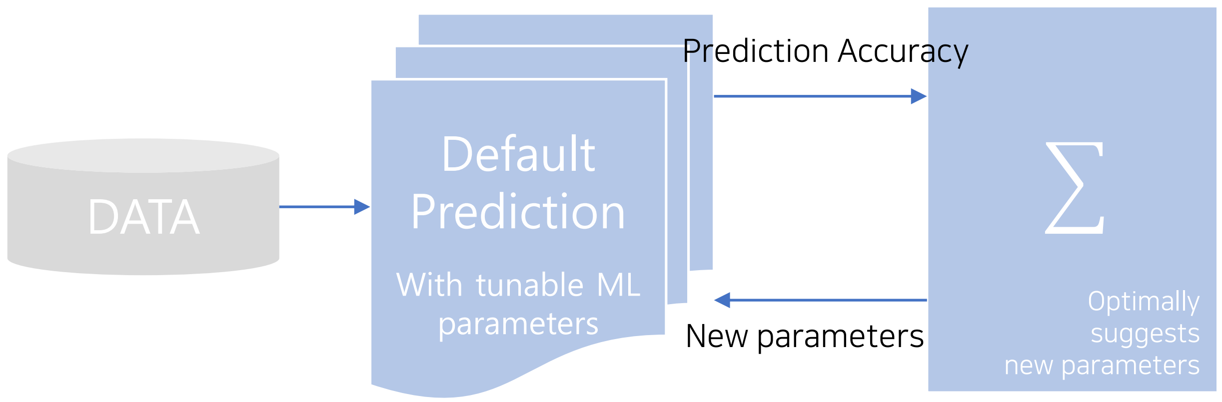
Common Approach

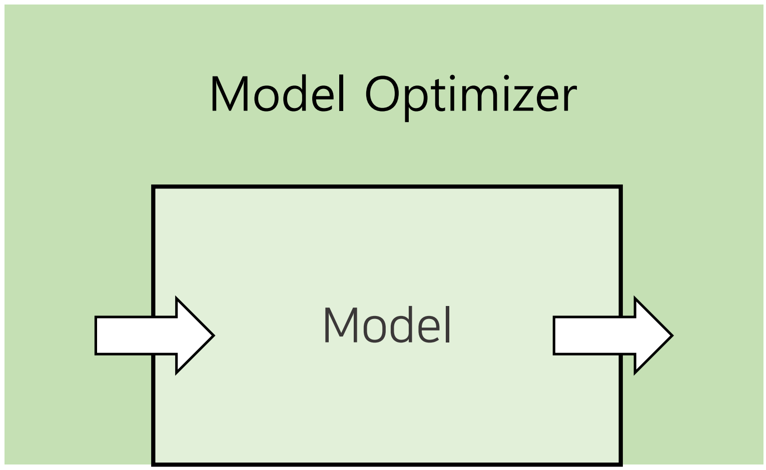
1. Optimize some Overall Evaluation Criterion (OEC)
   - Loss, Accuracy, Likelihood, Revenue

2. Given tunable parameters
   - Hyperparameters, feature parameters

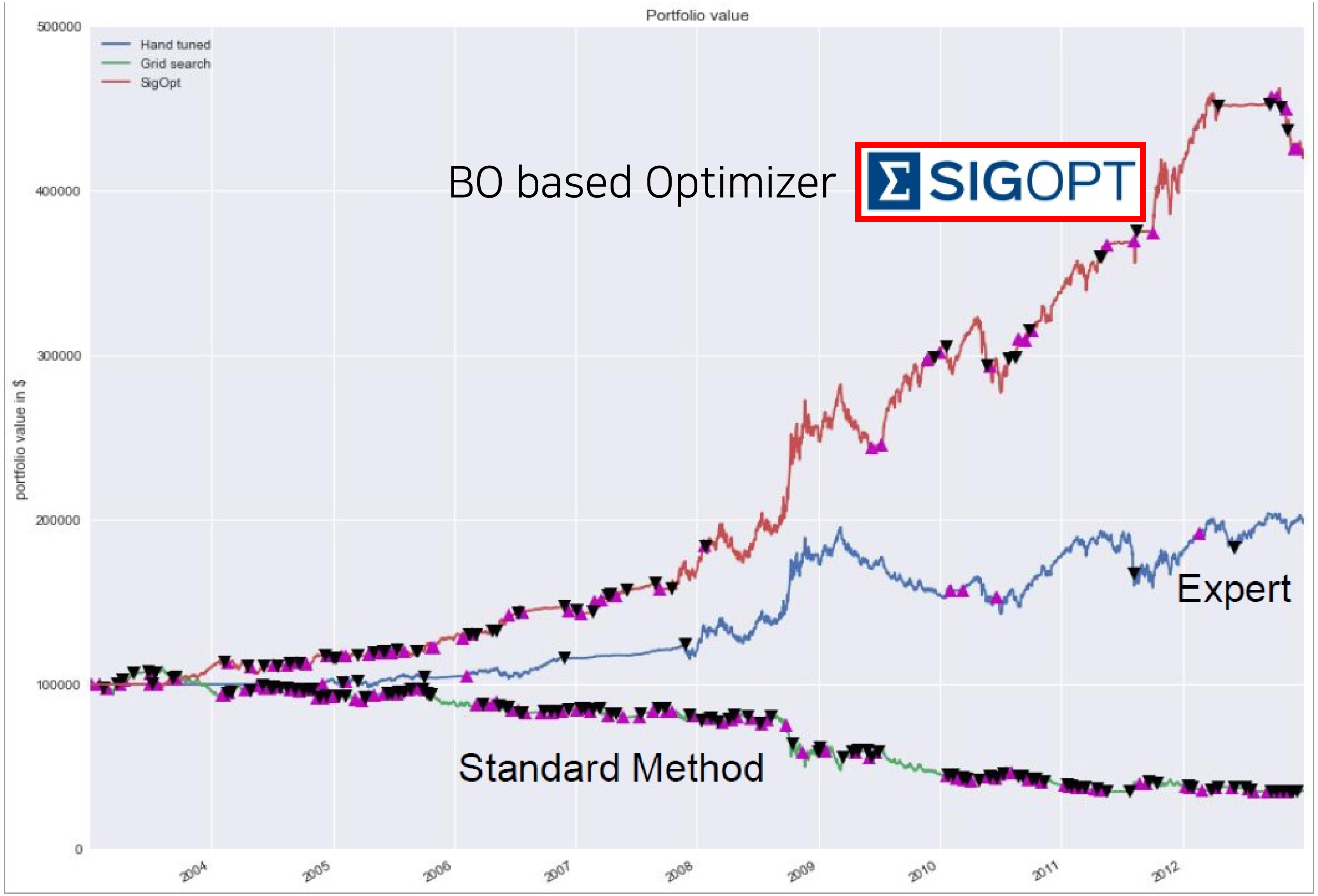
3. In an efficient way
   - Sample function as few times as possible
   - Training on big data is expensive

Common Approach





Common Approach



Optimization

Find optimal solution x^* that maximize $f(x)$

Suggest Two essential element

Surrogate Model

$$\left(x_1, f(x_1)\right), \ldots, (x_t, f(x_t))$$

Probabilistic Estimation of
Unknown Objective Function

Acquisition Function

$$Recommand\ x_{t+1}$$

Recommend candidate values to help
you find the optimal input value

Optimization

Find optimal solution x^* that maximize $f(x)$

Suggest Two essential element

알고리즘 Bayesian Optimization

for $t = 1, 2, \dots$ **do**

기존 입력값-함숫값 점들의 모음 $(x_1, f(x_1)), (x_2, f(x_2)), \dots, (x_t, f(x_t))$ 에 대한 Surrogate Model의 확률적 추정 결과를 바탕으로, Acquisition Function을 최대화하는 다음 입력값 후보 x_{t+1}을 선정한다.

입력값 후보 x_{t+1} 에 대한 함숫값 $f(x_{t+1})$ 을 계산한다.

기존 입력값-함숫값 점들의 모음에 $(x_{t+1}, f(x_{t+1}))$ 를 추가하고, Surrogate Model로 확률적 추정을 다시 수행한다.

end for

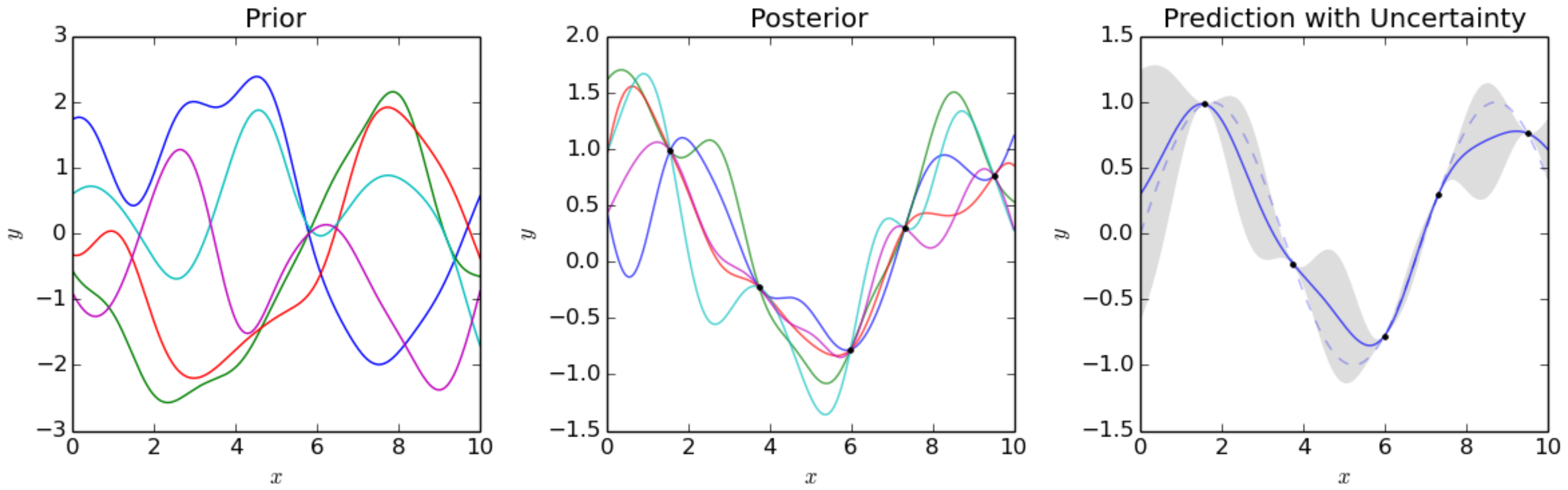
Pseudo-code for BO algorithm

Bayesian Optimization

Surrogate Model

Gaussian Processes(GP)

$$f(x) \sim GP(\mu(x), k(x, x'))$$



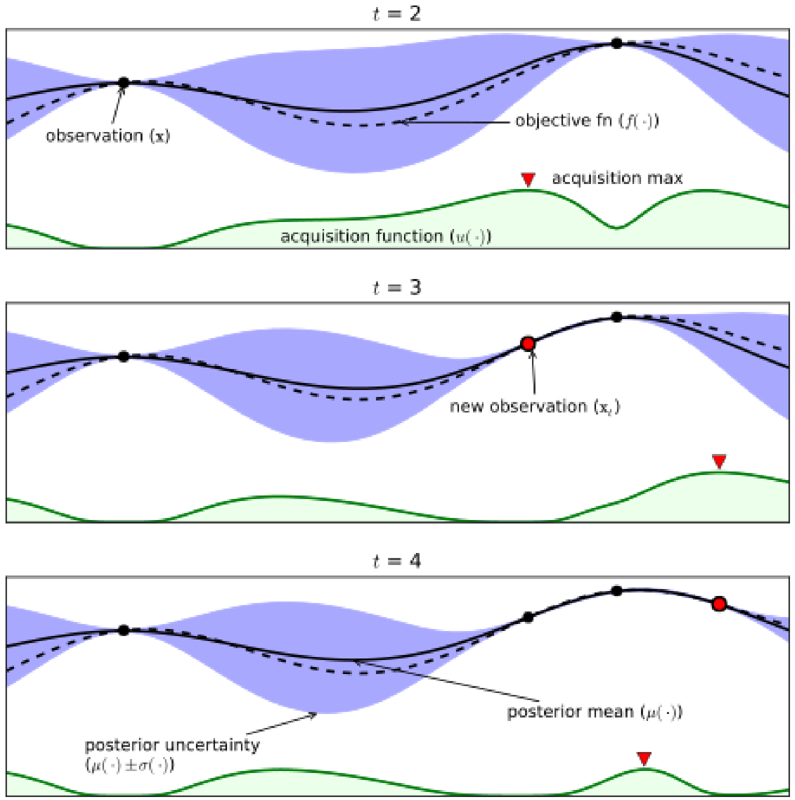
Gaussian process uses lazy learning and a measure of the similarity between points (the kernel function) to predict the value for an unseen point from training data

Gaussian processes can be seen as an infinite-dimensional generalization of multivariate normal distributions

Bayesian Optimization

Surrogate Model

Gaussian Processes(GP)



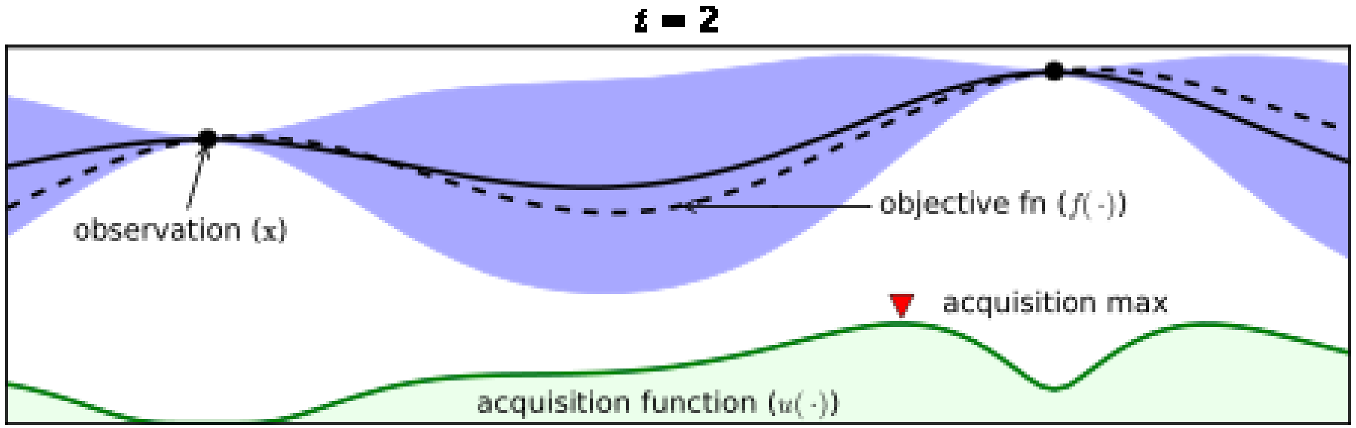
black dotted line : Actual object function

Black line : Estimated avg function

Blue space : Estimated Standard deviation

Black dots : searched input-function output point

Acquisition Function

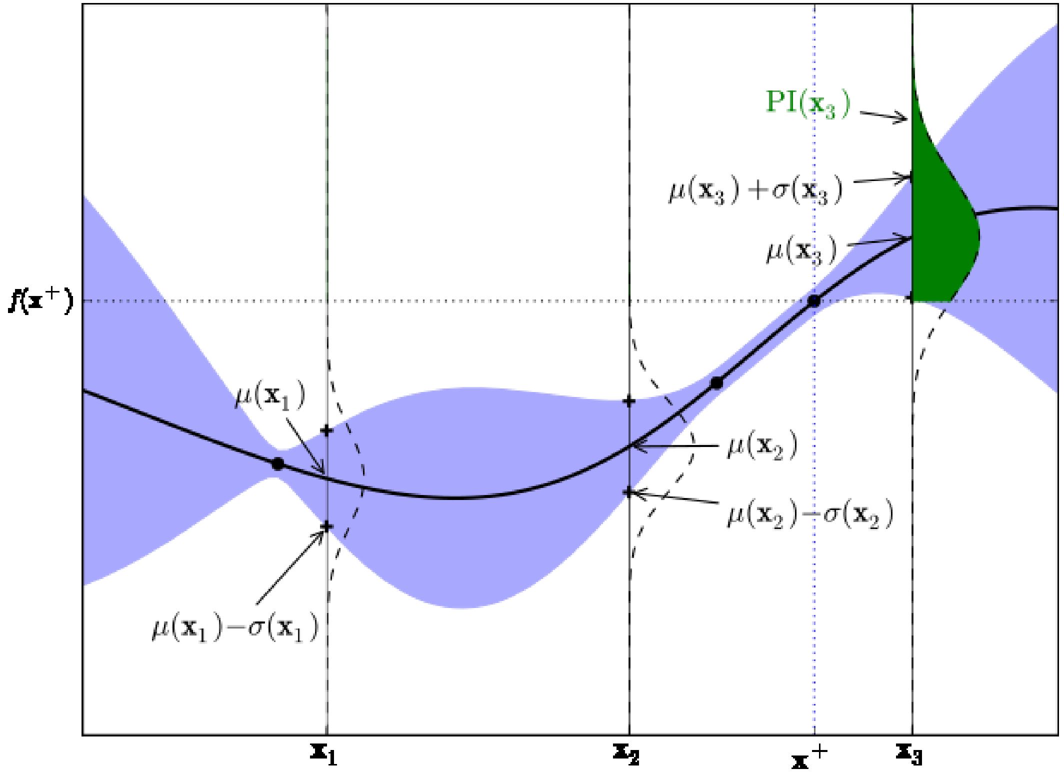


Trade-off

Exploitation vs Exploration

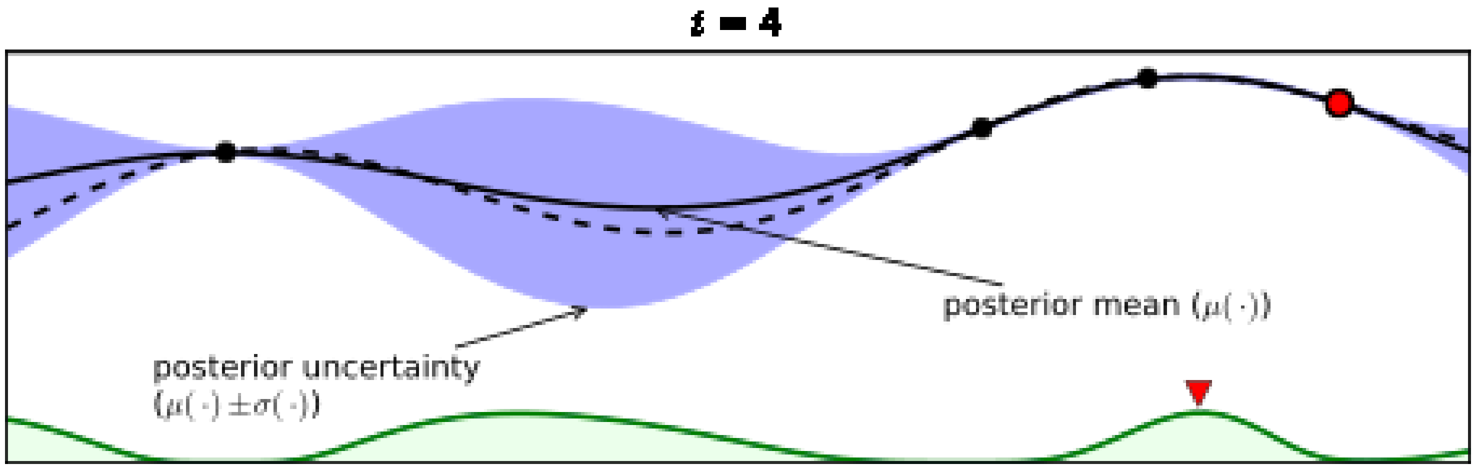
Acquisition Function

Expected Improvement (EI)



Acquisition Function

Expected Improvement (EI)



Define input, object function f and other setting values.

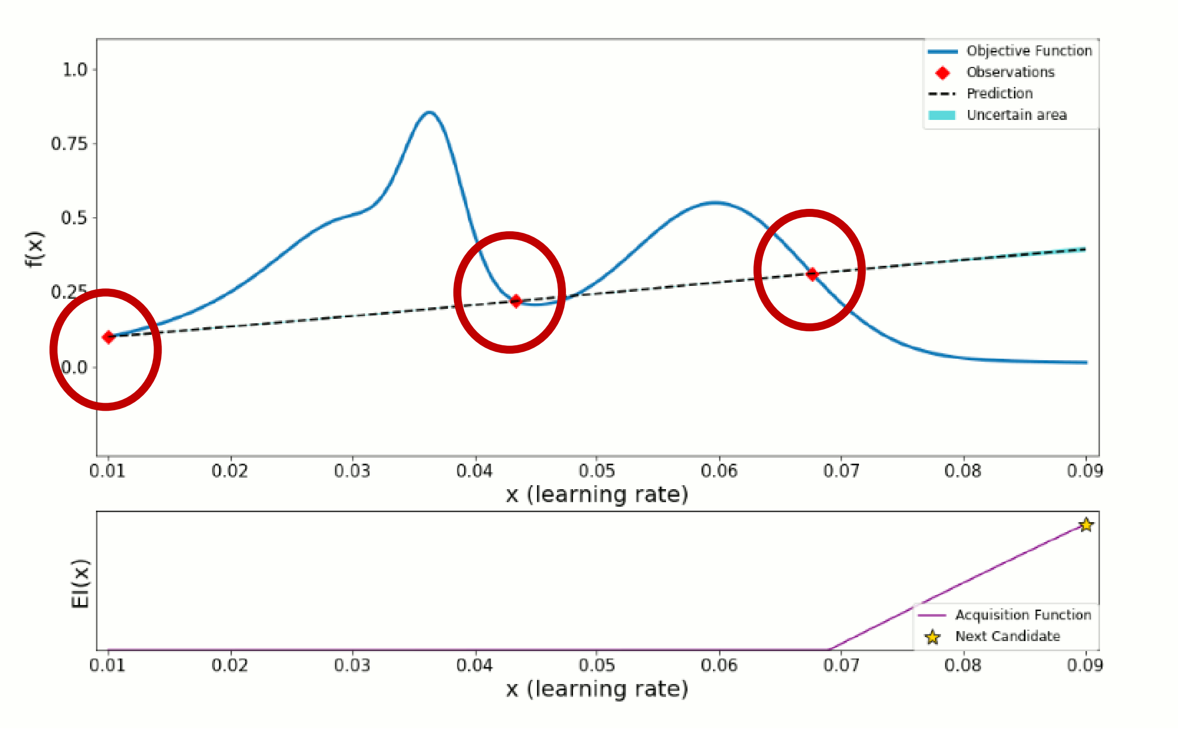
- Input: hyperparameters
- Objective function $f(x)$
- Others
  - Set lower/upper bound of x (a,b)
  - Determine N_{warmup} and N

    N_{warmup}: the number of based points $(x_i, f(x_i))$

    N : the number of training iteration

Bayesian Optimization

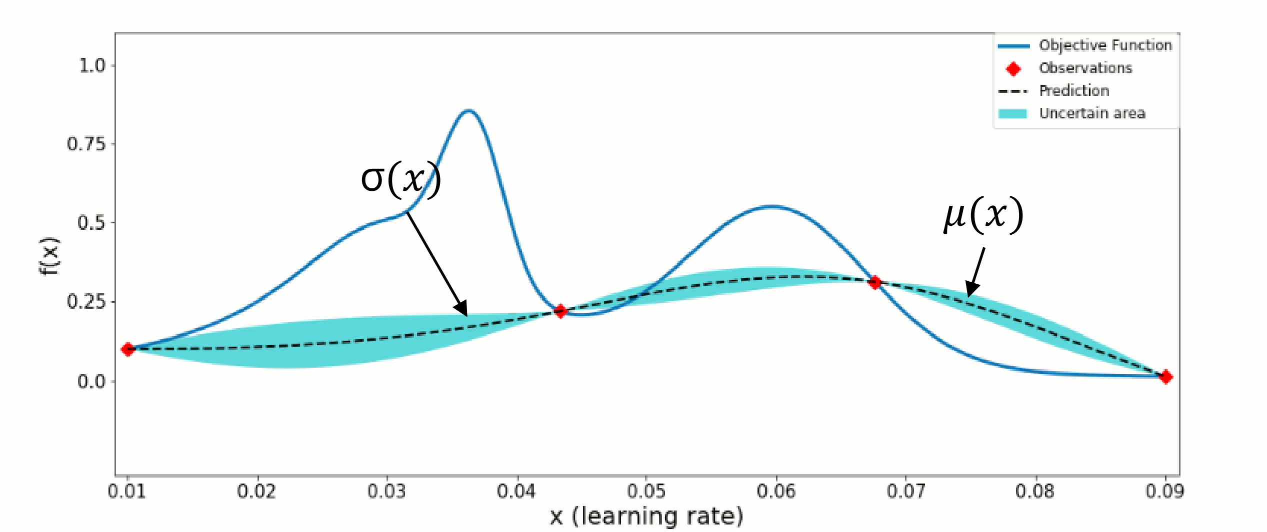
Warmup Phase



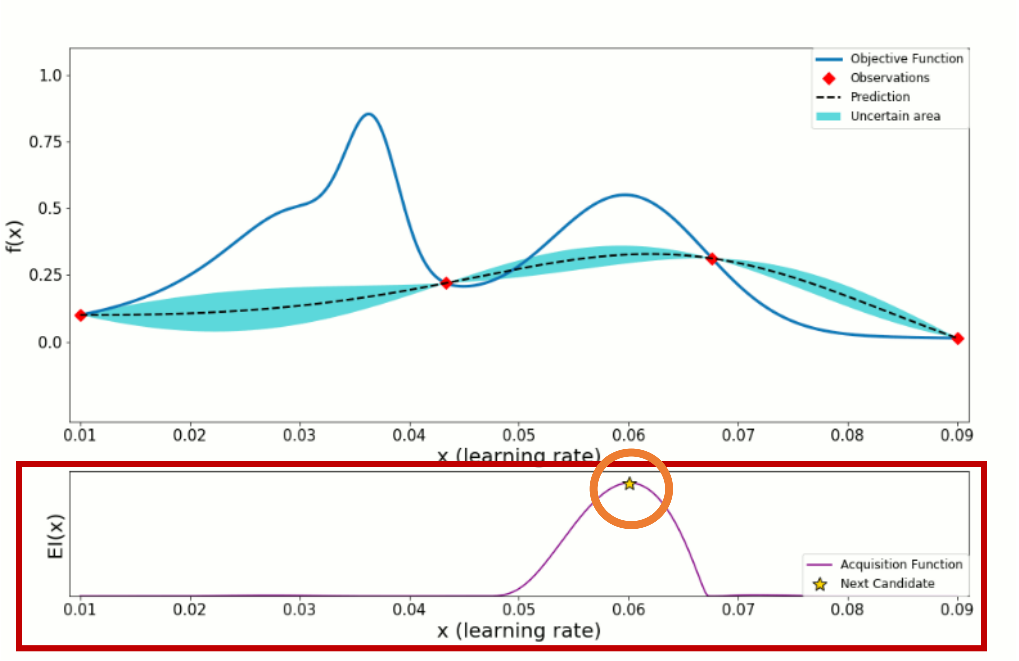
select x_i with random sampling and calculate $f(x_i)$ (1<= i < =N_{warmup})

$$(\boldsymbol{x}_1,\ \boldsymbol{f}(\boldsymbol{x}_1)), \ldots, (\boldsymbol{x}_i,\ \boldsymbol{f}(\boldsymbol{x}_i))$$

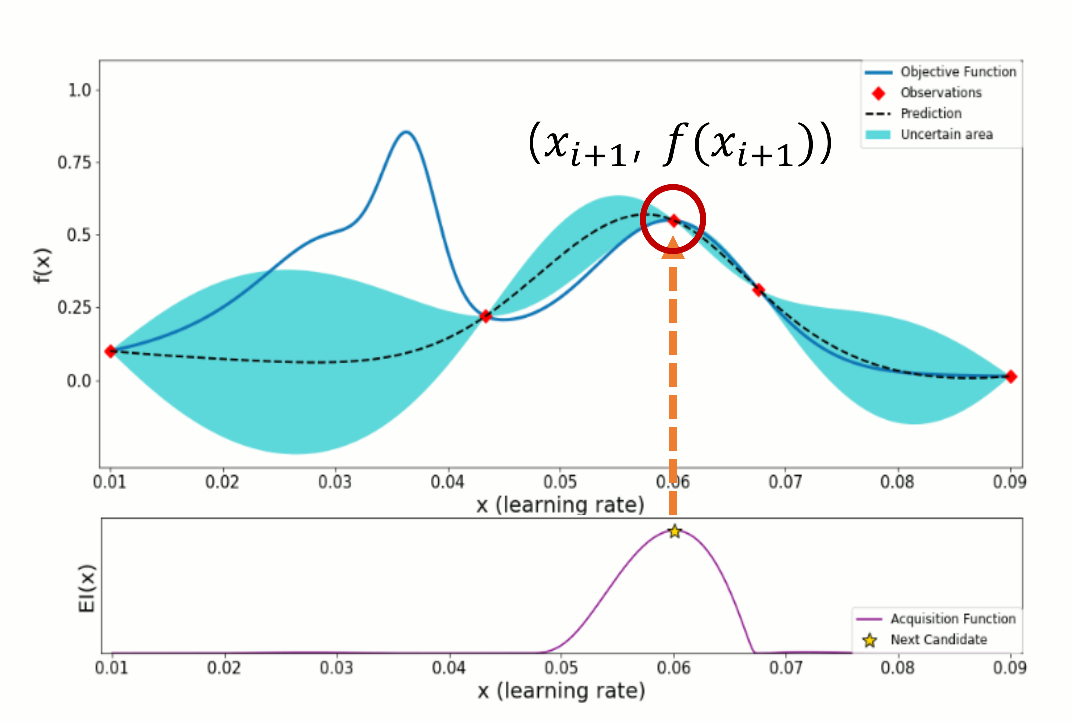
$$\boldsymbol{\sigma}(\boldsymbol{x}),\ \boldsymbol{\mu}(\boldsymbol{x})$$



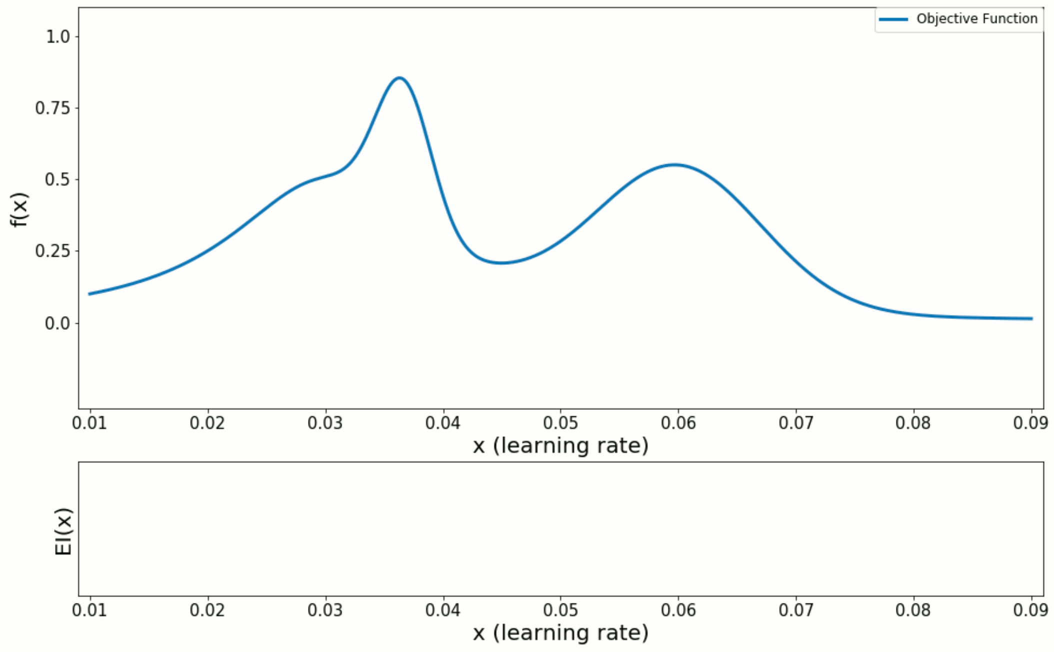
Execute probabilistic modeling and estimation under the gaussian process model, conditioned on all the previous observations of $(x_i,\ f(x_i))$



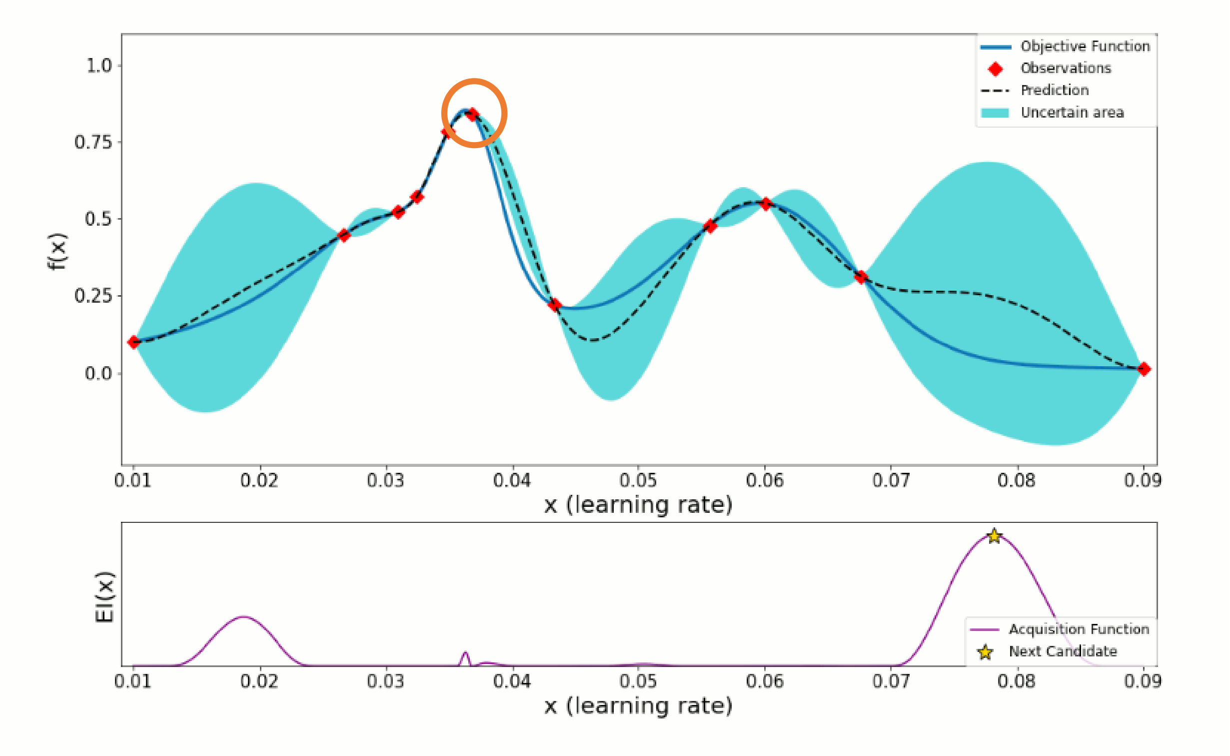
Calculate Expected Improvement function based on σ($\boldsymbol{x}$), $\boldsymbol{\mu}(\boldsymbol{x})$, and select $x_{i+1} = argEI(x)$



Train DL model with x_{i+1}, and
the performance of the result is considered $f(x_{i+1})$

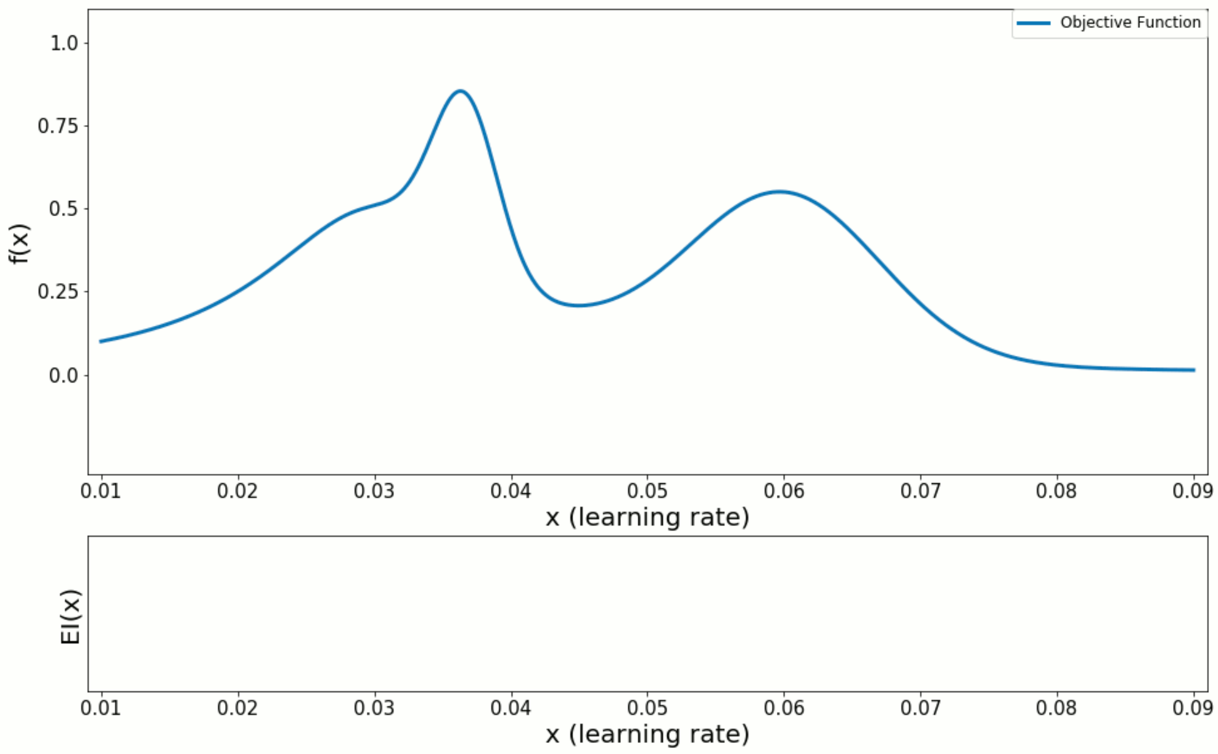


Repeat Upper process until the number of iteration is equal to N



Select x^* that maximize $\mu(x)$ based on objective function

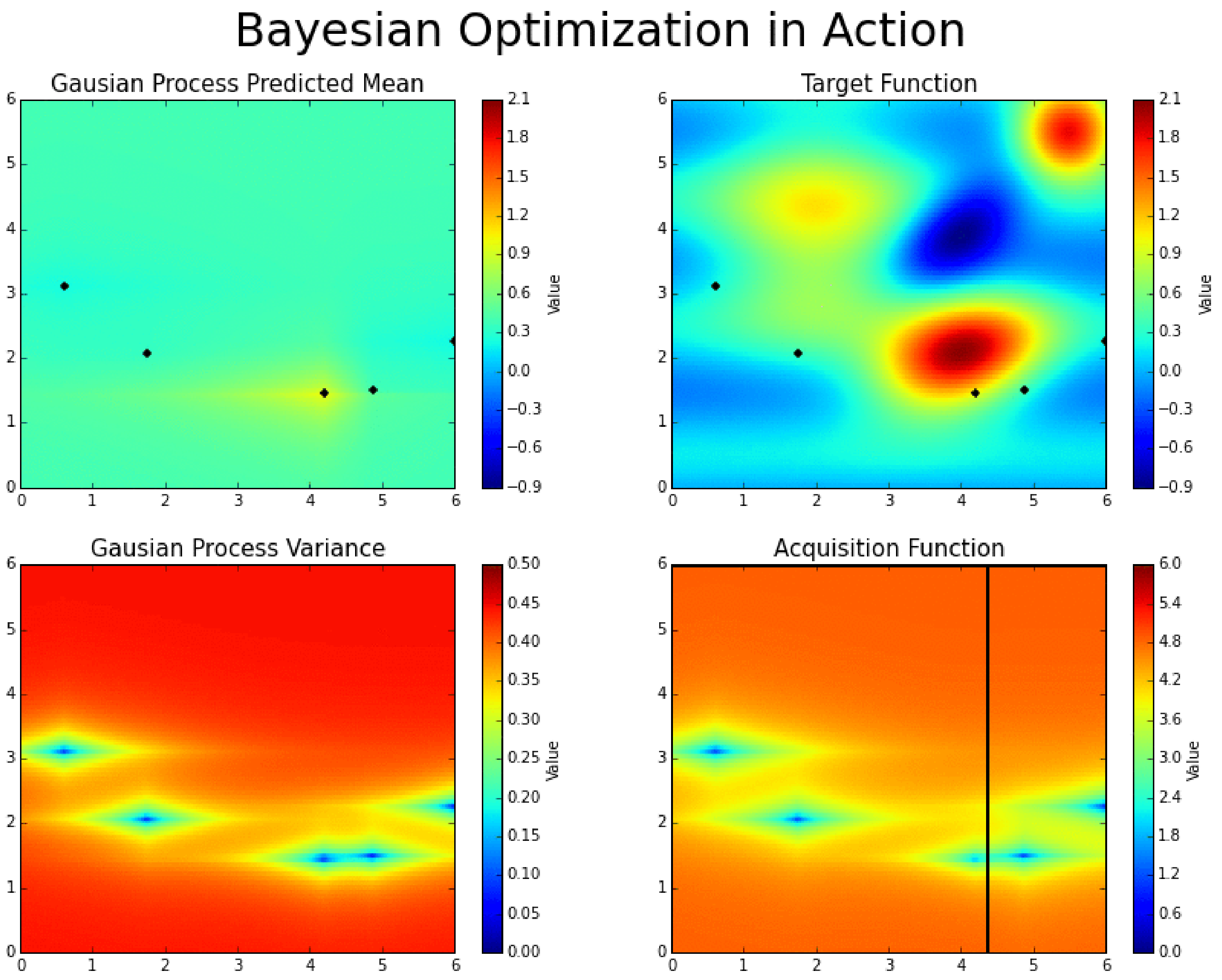
Bayesian Optimization



	EGO	RBF	DIRECT	GPGO 1-Step Non-Periodic	GPGO 1-Step Periodic	GPGO 2-Step Non-Periodic
Br	0.943	0.960	0.958	**0.980**	—	—
C6	0.962	0.962	0.940	0.890	—	**0.967**
G–P	0.783	0.815	**0.989**	0.804	—	**0.989**
H3	0.970	0.867	0.868	**0.980**	—	—
H6	0.837	0.701	0.689	**0.999**	—	—
Sh5	0.218	0.092	0.090	**0.485**	—	—
Sh7	0.159	0.102	0.099	**0.650**	—	—
Sh10	0.135	0.100	0.100	**0.591**	—	—
GK2	0.571	0.567	0.538	**0.643**	—	—
GK3	0.519	0.207	0.368	**0.532**	—	—
Shu	**0.492**	0.383	0.396	0.437	0.348	0.348
G2	0.979	**1.000**	0.981	**1.000**	**1.000**	—
G5	**1.000**	0.998	0.908	0.925	0.957	—
A2	0.347	0.703	0.675	0.606	0.612	**0.781**
A5	0.192	**0.381**	0.295	0.089	0.161	—
R	0.652	0.647	0.776	0.675	**0.933**	—
mean	0.610	0.593	0.604	**0.705**	—	—

BO works better than others in practice

Bayesian Optimization



SMBO

Sequential Model-Based Optimization

Common Approach

1. Build Gaussian Process (GP) with points
2. Optimize the fit of the GP (covariance
3. Find the point(s) of highest Expected Improvement within parameter domain
4. Return optimal next best point(s) to sample