The Intro

13 September 2024 13:39

The Why

13 September 2024 13:39

Hyperparameter

trial and error

max_depth → 5 / 6 / 7

max_features → 0.5

cgpa	iq	placement
7	70	1
8	80	1
6	60	0

tabular

settings

Decision Tree classifier

train

X_train, y_train

Xtest → pred → y_test → accuracy

[max_depth = 5] [max_fe 0.5] → 0.68 →

max_depth = 6 [max_f - 0.5] → |0.74|

= 7 0.78

8 Costly → time ↑ 0.76

compute ↑

Deep learning

face recog

Methods

1) Manual search → Scalable

2) Grid search

3) random search

gini entro

criteria → 2

max → [1 → 5] 1, 2, 3, 4, 5

max_f → 0 – 1 → 0.1 0.2 0.3 . . . 1

$\boxed{5 \times 10 \times 2 = 100}$

max depth (5)

max_features (10)

	1	2	3	4	5
0.1	0.68	0.99			
0.2	0.38				
.					
.					
1					

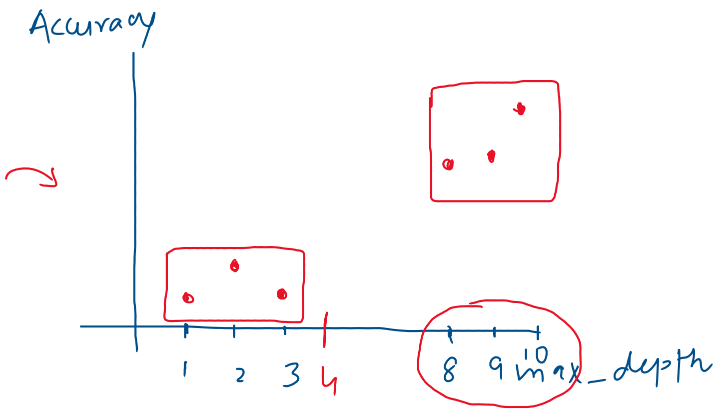
→ 50 times

simple

time taking

fast

intelligence



max_depth →

Bayesian search / optimizal

1
2
3
8
9
0

The What

13 September 2024 14:28

function

Bayesian Optimization is a strategy for optimization of black-box functions that are expensive to evaluate. It is particularly useful when dealing with functions that lack an analytical expression, are noisy, or are costly in terms of computational resources to evaluate.

math formulation → HP

accuracy

black box

$y = f(X)$ → formula X

$x^2 + 2$

$x^2 + \sin x$

max_depth = 10

input → processing → output

max depth

training

test

accuracy

D

train

test → accuracy value →

ml model → max depth = 3 4 5

black box

$y = f(x)$

metrics

hyperpara

max depth

max_depth

max_feature

$y = f(X, Z)$ → analytic graph → black box

max_feature

accuracy

Objective function

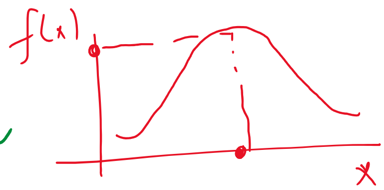
mse → min

$y = f(x)$ → maxima, minima

formula graph

accuracy

max_depth



The How

13 September 2024 16:39

$$y = \underset{x \in X}{\operatorname{argmin}} f(x)$$

max/min

f(x) → blackbox: formula, graph

cgpa	iq	placed
7	70	1
8	80	0
6	60	1
⋮	⋮	⋮

Decision Tree → HP → max depth

Steps

1, 2, 3, 4, 5 . . . 20 $\boxed{1 - 20}$

$\boxed{0.1 - 1.0}$

Step 1 - Define Objective Function f(x)

Step 2 - Define Search space -> Define a Search space for each of your hyperparameters

- Continuous Parameters: Specify bounds (e.g., max_features between 0.1 and 1.0).
- Discrete Parameters: List possible values (e.g., max_depth: 1, 2, 3, 4).
- Categorical Parameters: Enumerate categories (e.g., criteria: entropy, gini).

[1, 2, 3 . . . 20] → 20 items

Step 3 - Sample initial data points -> Select 5-10 initial points {x1, x2, x3 … xn} using technique like random sampling

3 initial points

5 → 0.68

6 → 0.74

Step 4 - For each initial data point train the model and compute the output for the objective function f(xi)

Step 5 - Prepare a dataset D

$$D = \{(x_i, f(x_i))\}_{i=1}^{n}$$

2, 9, 19

x_1 x_2 x_3

```
def objective(max_depth):
    clf = DecisionTreeClassifier(max_depth=max_depth)
    # Train the classifier
    clf.fit(X_train, y_train)
    # Predict on the validation set
    y_pred = clf.predict(X_val)
    # Compute the accuracy
    accuracy = accuracy_score(y_val, y_pred)
    # Minimize negative accuracy
    return -accuracy
```

= 2, 9, 19

input max_depth → output accuracy

accuracy

$y = f(x)$

max depth

Step 6 - Choose a Surrogate model -> **surrogate function** (also known as a **response surface**) is a probabilistic model used to approximate the expensive-to-evaluate objective function.

- Gaussian Process (GP): Common for continuous spaces; provides uncertainty estimates.
- Random Forests: Suitable for mixed parameter types; robust to noisy observations.
- Tree-structured Parzen Estimator (TPE): Efficient for high-dimensional spaces.

Step 7 - Fit the Surrogate model on the dataset D

Step 8 - Approximate the objective prediction by doing predictions on random points.

GPR ← D

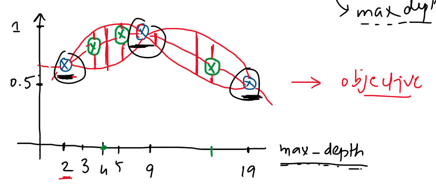
train

regression problem

9, 8, 16

surrogate function

input	output
2	0.62
9	0.81
19	0.59



→ objective

GPR

train

5

accuracy

ml model (regressor)

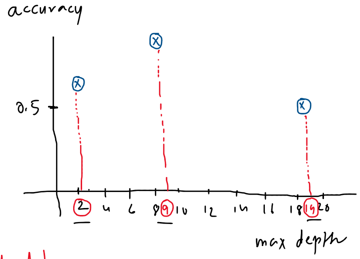
↳ data → new data → predn

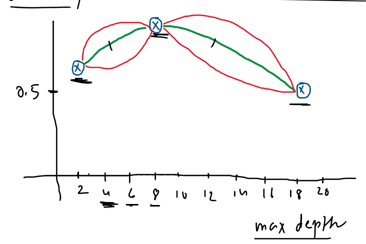
GPR → 0.73 → 0.3

9

actual relation

uncertainty





input	output
2	0.62
9	0.81
19	0.59

with the help of objective function

with the help of surrogate function

$\boxed{2, 9, 19}$

1 - 20 → 3 values 3 8, 15

6.52

0.67

0.91

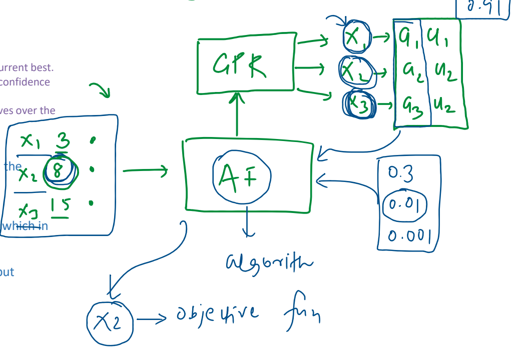
Step 9 - Select a Acquisition function

- Expected Improvement (EI): Measures the expected improvement over the current best.
- Upper Confidence Bound (UCB): Balances exploration and exploitation using confidence intervals.
- Probability of Improvement (PI): Estimates the probability that a point improves over the best observed value.

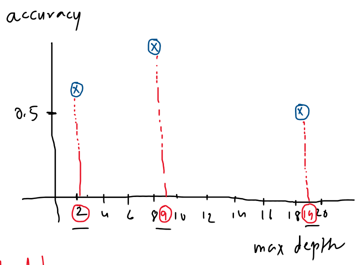
Step 10 - Randomly select some candidate points (3) and send them to the surrogate function to get the accuracy and uncertainty scores

Step 11 - Send these accuracy scores and uncertainty scores to the AF, which in turn will decide the best possible candidate point

Step 12 - Send the best candidate point to the objective function as input

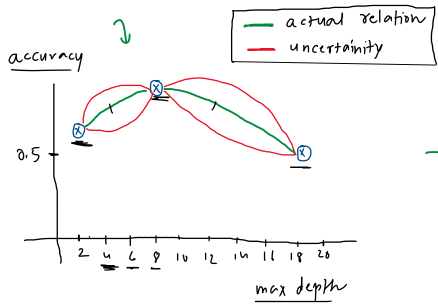


md → 8 → of → 0.80

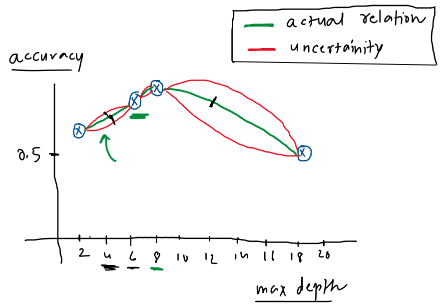


input	output
2	0.62
9	0.81
19	0.59
8	0.80

with the help of objective function



with the help of surrogate function



with the help of surrogate function

Step 13 - Append the next {xi, f(xi)} to your dataset D

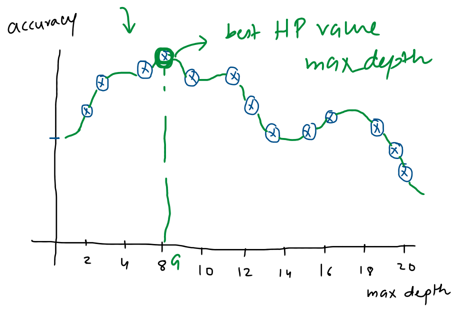
Step 14 - Retrain the Surrogate function on the updated dataset.

GPR

3 blue

13 blue

13 points



for i in range(10)

- -> find the next best candidate point using AF (with the help of SF)
- -> Send the next best candidate point to OF, get the new blue point
- -> Append the new blue to your dataset D
- -> retrain your SF on the updated dataset D

Step 15 - Find the overall best max_depth value