Stanford University

CS 329P : Practical Machine Learning (2021 Fall)

9.2 HPO algorithms

Qingqing Huang, Mu Li, Alex Smola

https://c.d2l.ai/stanford-cs329p

Search Space

- Specify range for each hyperparameter

Hyper-Parameter	Range	Distribution
model(backbone)	[mobilenetv2_0.25, mobilenetv3_small, mobilenetv3_large, resnet18_v1b, resnet34_v1b, resnet50_v1b, resnet101_v1b, vgg16_bn, se_resnext50_32x4d, resnest50, resnest200]	categorical
learning rate *	[1e-6, 1e-1]	log-uniform
batch size *	[8, 16, 32, 64, 128, 256, 512]	categorical
momentum **	[0.85, 0.95]	uniform
weight decay **	[1e-6, 1e-2]	log-uniform
detector	[faster-rcnn, ssd, yolo-v3, center-net]	categorical

- The search space can be exponentially large
  - Need to carefully design the space to improve efficiency

HPO algorithms: Black-box or Multi-fidelity

- Black-box: treats a training job as a black-box in HPO:
  - Completes the training process for each trial
- Multi-fidelity: modifies the training job to speed up the search
  - Train on subsampled datasets
  - Reduce model size (e.g less #layers, #channels)
  - Stop bad configuration earlier

HPO algorithms

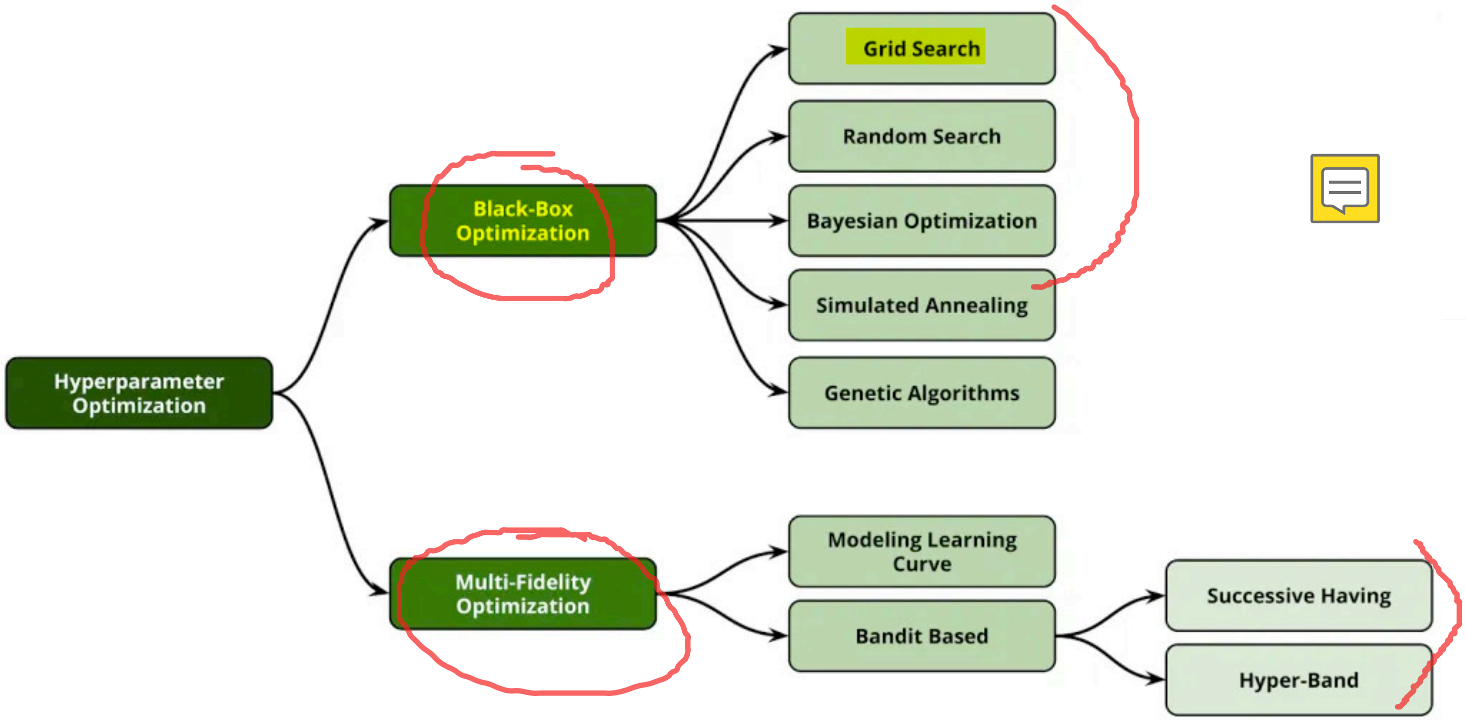


Image credit: [Automated Machine Learning: State-of-The-Art and Open Challenges]()

Two most common HPO strategies

- Grid search

```
for config in search_space:
    train_and_eval(config)
return best_result
```

  - All combinations are evaluated
  - Guarantees the best results
  - Curse of dimensionality

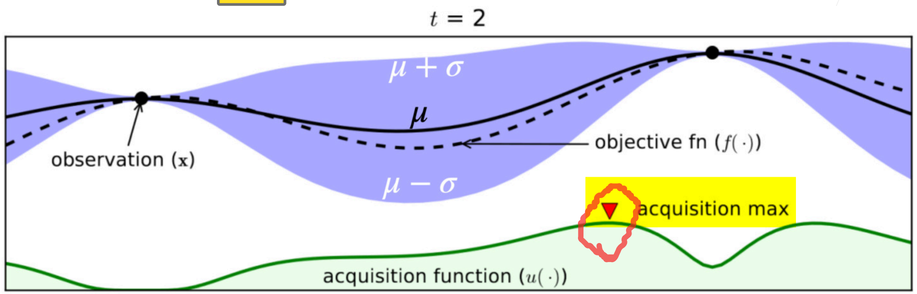
- Random search

```
for _ in range(n):
    config = random_select(search_space)
    train_and_eval(config)
return best_result
```

  - Random combinations are tried
  - More efficient than grid search (empirically and in theory, shown in [Random Search for Hyper-Parameter Optimization](#))

Bayesian Optimization (BO)

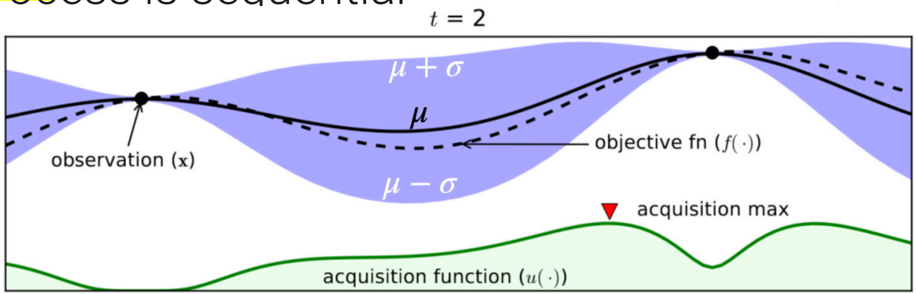
- BO: Iteratively learn a mapping from HP to objective function. Based on previous trials. Select the next trial based on the current estimation.
- Surrogate model
  - Estimate how the objective function depends on HP
  - Probabilistic regression models: Random forest, Gaussian process, …



Stanford CS 329P (2021 Fall) - htt

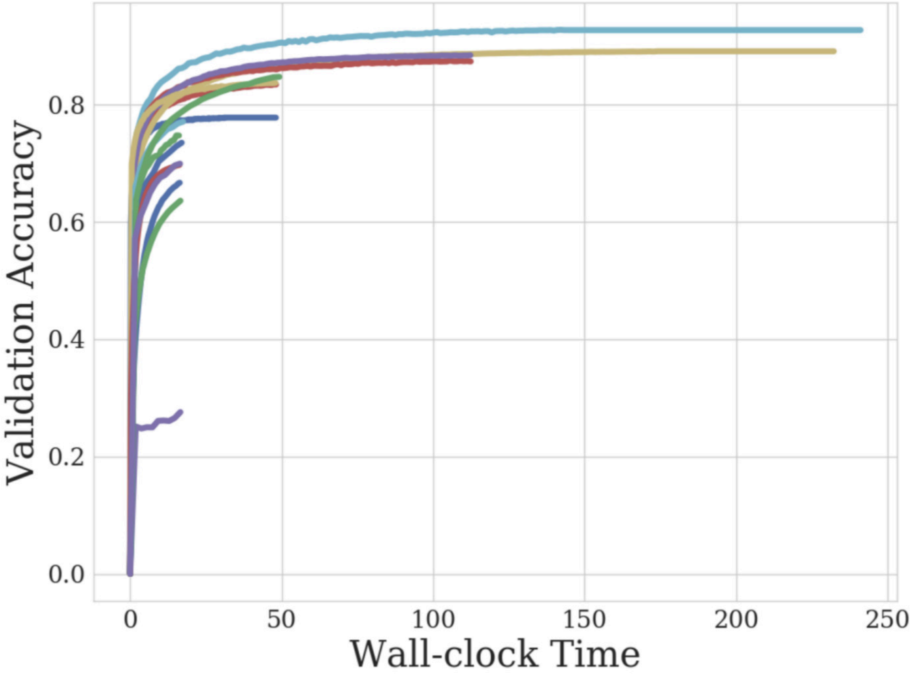
Bayesian Optimization (BO)

- Acquisition function
  - Acquisition max means uncertainty and predicted objective are high.
  - Sample the next trial according to the acquisition function
  - Trade off exploration and exploitation
- Limitation of BO:
  - In the initial stages, similar to random search
  - Optimization process is sequential



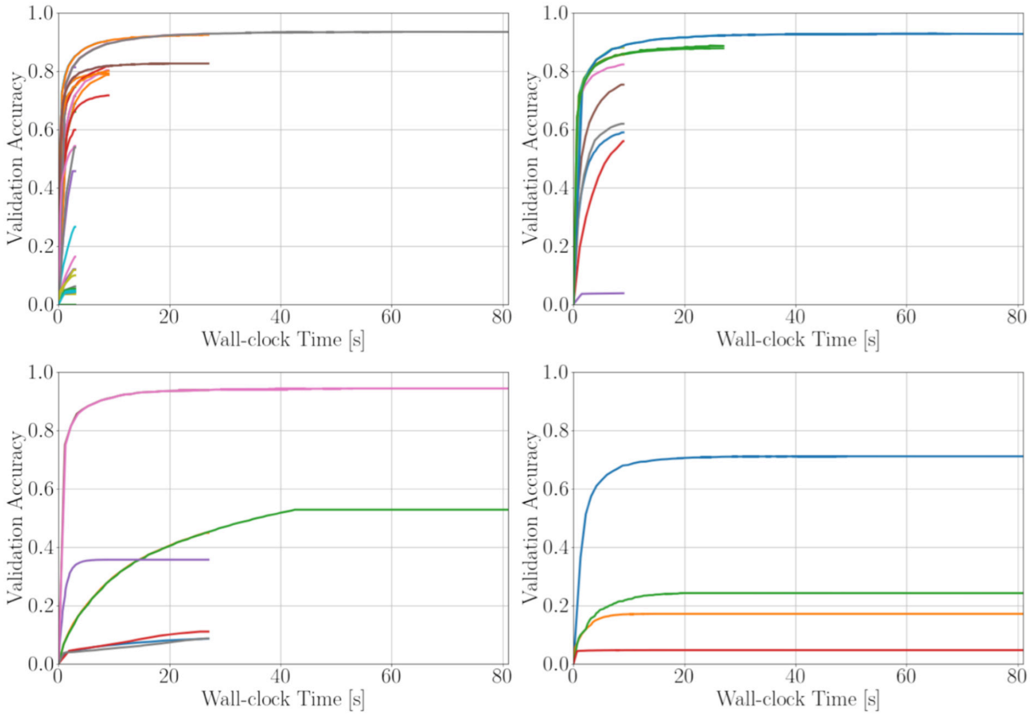
Successive Halving

- Save the budget for most promising config
- Randomly pick n configurations to train m epochs
- Repeat until one configuration left:
  - Keep the **best** $n/2$ configuration to train another m epochs
  - Keep the **best** $n/4$ configuration to train another $2m$ epochs
  - ……
- Select n and m based on training budget and #epoch needed for a full training



Hyperband

- In Successive Halving
  - n: exploration
  - m: exploitation
- Hyperband runs multiple Successive Halving, each time decreases n and increases m
  - More exploration first, then do more exploit



Summary

- Black-box HPO: grid/random search, bayesian optimization
- Multi-fidelity HPO: Successive Halving, Hyperband
- In practice, start with random search
- Beware there are top performers
  - You can find them by mining your training logs, or what common configurations used in paper/code