Hyperband

Yuxuan Zhao

March 14, 2022

Quiz

Which is not a type of resource to be allocated considered in this paper?

1. Number of training iterations
2. Number of random features
3. Number of validation points

Which statement is true about Hyperband?

1. Each hyperband call executes successive halving n times with $1, 2, ..., n$ configurations respectively.
2. Each outer loop of hyperband executes successive halving.
3. Each inner loop of hyperband executes successive halving.

Motivation: why not Bayesian optimization (BO)?

- BO outperforms random search only by a small margin
    - in experiments using over 100 datasets

Motivation: why not Bayesian optimization (BO)?

- BO outperforms random search only by a small margin
  - in experiments using over 100 datasets
- BO's assumption may not always be valid
  - BO has continuity assumption w.r.t. hyperparameters
- Random search has minimal assumptions

Motivation: why not Bayesian optimization (BO)?

- BO outperforms random search only by a small margin
  - in experiments using over 100 datasets
- BO's assumption may not always be valid
  - BO has continuity assumption w.r.t. hyperparameters
- Random search has minimal assumptions

Speeding up random search!

Configuration evaluation

- Adaptively allocate resources to a set of configurations
    - Resources: training iterations, training data size, etc.
- Find bad configurations quickly and discard them
- Examine more configurations than random search

Successive halving

For simplicity, suppose n is power of 2

```
Require: Number of configurations $n$, total budget $B = (\log_2(n) + 1)\bar{B}$
  Sample $n$ configurations as $S_0$
  for $t = 0, \ldots, \log_2(n)$ do
    Run each configuration in $S_t$ using $\frac{\bar{B}}{n/2^t}$ resource
    Let $S_{t+1}$ be the set of $|S_t|/2$ best performed configurations
  end for
```

- Equal round budget: $\bar{B}$
  - At round t: $n/2^t$ configurations left
- Equal configuration budget in each round
  - 1st round: $\bar{B}/n$ budget for each configuration
  - last round: $\bar{B}$ budget for each configuration
  - A configuration gets at least $\bar{B}/n$ and at most $\bar{B}(2 - 1/n)$ resource

Successive halving

For simplicity, suppose n is power of 2

```
Require: Initial $n$ configurations in $S_0$, total budget $B = (\log_2(n) + 1)\bar{B}$
  for $t = 0, \ldots, \log_2(n)$ do
    Run each configuration in $S_t$ using $\frac{\bar{B}}{n/2^t}$ resource
    Let $S_{t+1}$ be the set of $|S_t|/2$ best performed configurations
  end for
```

Demo: `https://i0.wp.com/neptune.ai/wp-content/uploads/Successive-halving.gif`

Tradeoff in n versus B/n

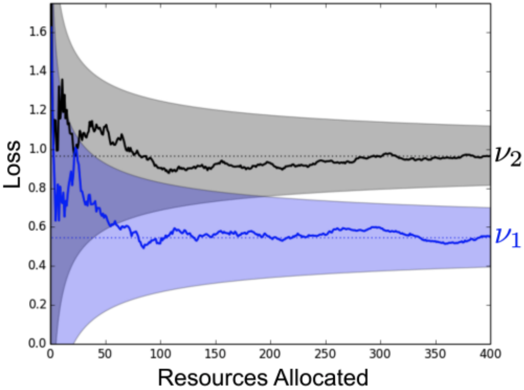


Figure 1: The validation loss as a function of total resources allocated for two configurations is shown.

Motivation of hyperband

Do grid search over (n_i, B) for different n_i but the same total resource

- Determine $\{n_i\}$ by the number of rounds $\lfloor \log_\eta(n_i) \rfloor + 1$
    - In successive halving, $\eta = 2$
    - Number of eliminations: $s_i = \lfloor \log_\eta(n_i) \rfloor$
    - For a given $s_{\max}$, use $s_i = s_{\max}, \ldots, 1, 0.$
        - $n_i = \eta^{s_i}$
        - Call successive halving $s_{\max} + 1$ times
- Determine $s_{\max}$ by the resource constraint
    - Max R and min r for a configuration in a single round

Max R and min r for a configuration in a single round

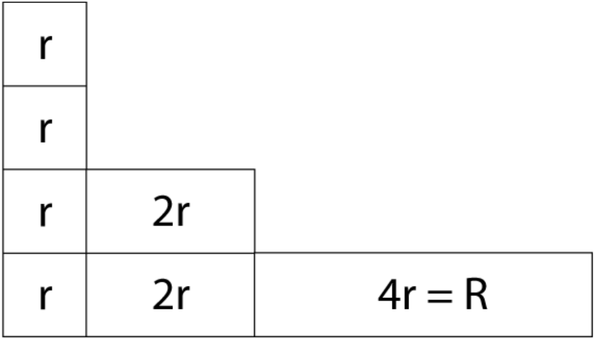


Figure 2: Successive halving with $n = 4, \eta = 2$. Match the total resource at the first round and the last round

Determine the maximum number of eliminations

In the most aggressive setting, $s = s_{\max}$,

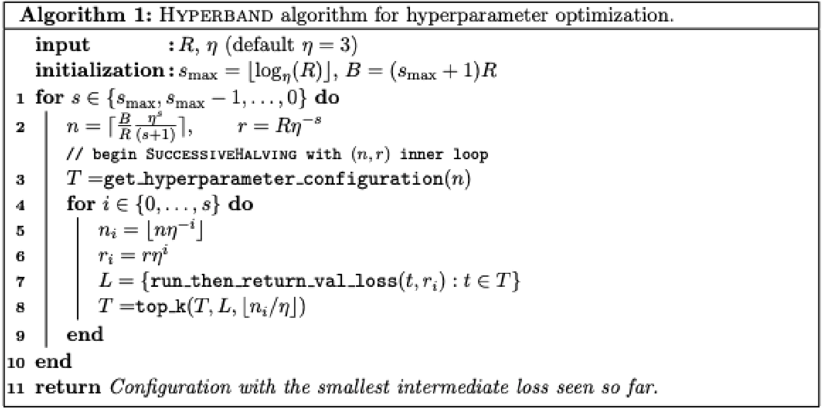
- Resource at the $1st$ round: $\geq r\eta^{s_{\max}}$
- Resource at the last round if only one configuration: $\leq R$
- By equal resource across rounds, we know $s_{\max} = \lfloor \log_\eta(\frac{R}{r}) \rfloor$.
  - if the minimum and the maximum resource are realized
- **The paper seems to assume** $r = 1$
- Fix the total resource $B = (s_{\max} + 1)R$ for all successive halving calls

For each successive halving call

If there are $s+1$ rounds (s eliminations),

- Total resource: $(s_{\text{max}}+1)R$
- Per round resource: $\frac{s_{\text{max}}+1}{s+1}$
- Minimal resource for a configuration: $\frac{R}{\eta^s}$
    - $r\eta^{(s_{\text{max}}-s)}$ for $r=1$
- Number of configurations: $\lceil \eta^s \frac{s_{\text{max}}+1}{s+1} \rceil$

Hyperband



Algorithm 1: HYPERBAND algorithm for hyperparameter optimization.

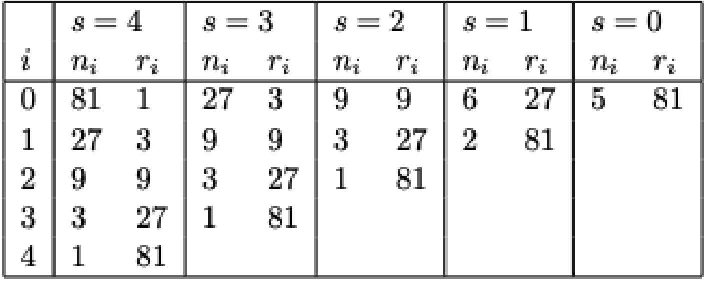
```
input          : $R$, $\eta$ (default $\eta = 3$)
initialization : $s_{\max} = \lfloor \log_\eta(R) \rfloor$, $B = (s_{\max} + 1)R$
for $s \in \{s_{\max}, s_{\max} - 1, \ldots, 0\}$ do
    $n = \lceil \frac{B}{R} \frac{\eta^s}{(s+1)} \rceil$,    $r = R\eta^{-s}$
    // begin SUCCESSIVEHALVING with $(n, r)$ inner loop
    $T$ =get_hyperparameter_configuration($n$)
    for $i \in \{0, \ldots, s\}$ do
        $n_i = \lfloor n\eta^{-i} \rfloor$
        $r_i = r\eta^i$
        $L$ = {run_then_return_val_loss($t, r_i$) : $t \in T$}
        $T$ =top_k($T, L, \lfloor n_i/\eta \rfloor$)
    end
end
return Configuration with the smallest intermediate loss seen so far.
```

Figure 3: One may set the maximum number of configurations $n_{\max}$ to explore in the most aggressive setting so that $s_{\max} = \lfloor \log_\eta(n_{\max}) \rfloor$

Example: LeNet on MNIST

- Hyperparameters: learning rate, batch size, number of kernels for two layers
- Resources: number of iterations of SGD
  - One unit of resource for one epoch

Example: LeNet on MNIST



	$s=4$		$s=3$		$s=2$		$s=1$		$s=0$	
i	n_i	r_i	n_i	r_i	n_i	r_i	n_i	r_i	n_i	r_i
0	81	1	27	3	9	9	6	27	5	81
1	27	3	9	9	3	27	2	81		
2	9	9	3	27	1	81				
3	3	27	1	81						
4	1	81								

Figure 4: The values of n_i and r_i for the brackets of Hyperband corresponding to various values of s, when $R = 81$ and $\eta = 3$.

Example: LeNet on MNIST

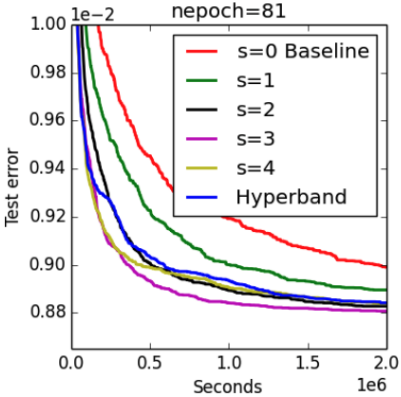


Figure 5: Performance of individual brackets s and Hyperband. $s = 3$ is the best and hyperband is close to the best.

Setting η

- Larger values lead to more aggressive elimination schedules
- "For most problems, 5 is a reasonable number of n versus B/n tradeoffs to explore"

Setting R

- Natural upper bound often exists
- Use infinite R (came first)
  1. Double the budget B over time
  2. For each $n \in \{\eta^k : k = 1, ..., \log_\eta(B)\}$, run successive halving with $(n, R = B/(\log_\eta(n) + 1))$.

Theoretical guarantee

$\nu_i(j)$: the error of configuration i with j units of resource.

Assumption:

- ν_i: the limit of $\nu_i(j)$ as the resource j approaches R (could be ∞)
- Each ν_i is a bounded i.i.d. R.V. with CDF F

Let $\nu_* = \inf_i \nu_i$ denote the optimal error

Results

- The convergence rate to stable performance is uniformly bounded
  - $\sup_i |\nu_i(j) - \nu_i| \leq \gamma(j)$ for all j

Theoretical guarantee

Theorem

Fix $\delta \in (0,1)$. For any total resource $T \in \mathbb{N}$, let τ_T be the empirically best-performing arm output from SuccessiveHalving from the last round of Hyperband after exhausting a total budget of T from all rounds, then

$$\nu_{\tau_T} - \nu_* \leq c \left(\frac{\overline{\log}(T)^3 \overline{\log}(log(T)/\delta)}{T} \right)^{1/\max(\alpha,\beta)}$$

for some constant $c = \exp(O(\max(\alpha, \beta)))$, with probability at least $1 - \delta$, where $\overline{\log} x = \log(x) \log(\log(x))$.

- $F(x)$ has rate $(x - \nu_*)^\beta$ when $x \geq \nu_*$
- $\gamma(j)$ has rate $\left(\frac{1}{j}\right)^{1/\alpha}$

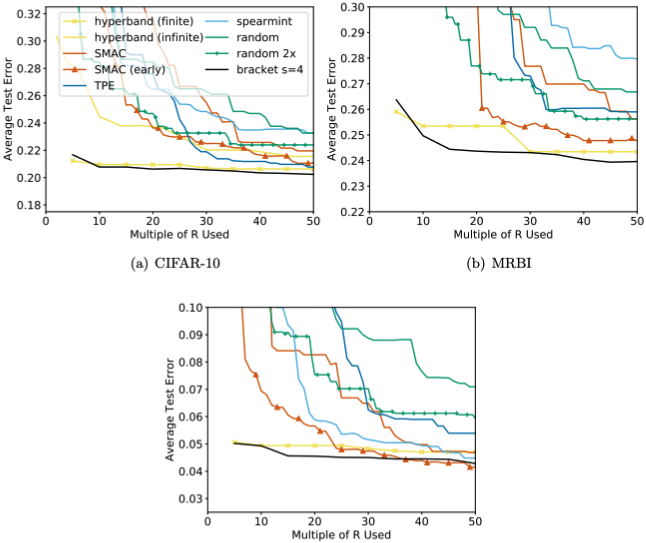
Competitors

- Random search and random search 2×
- Most aggressive successive halving (bracket $s = 4$)
- Bayesian optimization (SMAC, SMAC early, TPE and spearmint)

Early-stopping iterative algorithms

- Resource: number of iterations
- Configuration: 6 hyperparameters for SGD and 2 hyperparameters for the response normalization layers
- The most aggressive successive halving has 4 brackets (5 rounds).

Early-stopping iterative algorithms



(a) CIFAR-10

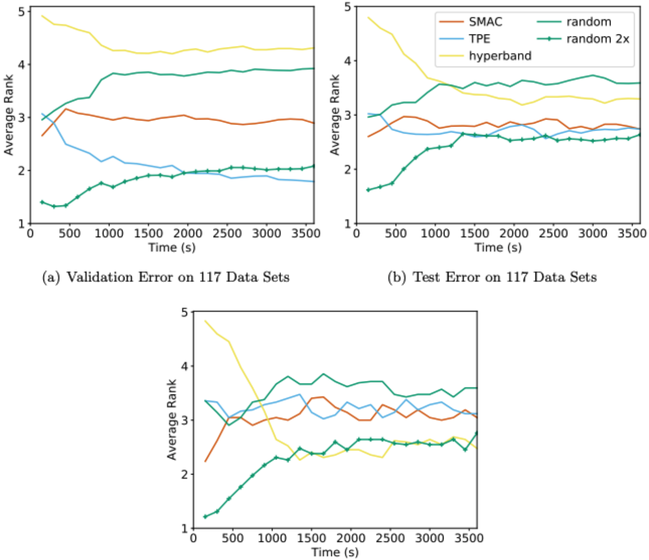
(b) MRBI

(c) SVHN

Dataset subsampling

- 140 binary and multiclass classification datasets from openML
  - Divided into two groups: 117 datasets and 23 datasets
- Configuration: 110 hyperparameters
  - 15 classifiers, 14 feature preprocessing methods, 4 data preprocessing methods
- Evaluation: rank among methods (smaller is better)
  - averaged over datasets

Dataset subsampling: averaged rank



(a) Validation Error on 117 Data Sets

(b) Test Error on 117 Data Sets

(c) Test Error on 21 Data Sets

Dataset subsampling: test error difference

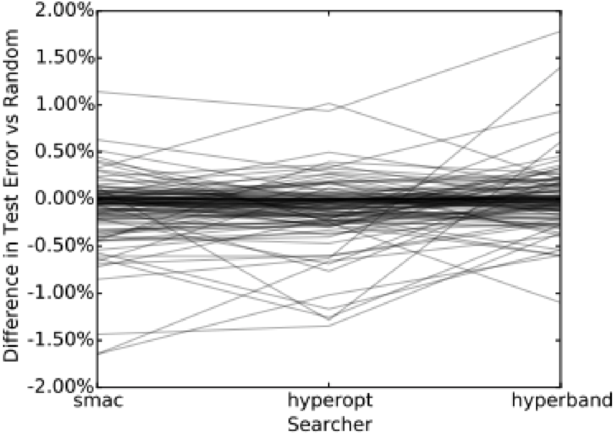


Figure 6: Each line plots the difference in test error versus random search for a single dataset. Lower is better.

Kernel based classifier for CIFAR-10

- The number of training samples as resource, 6 hyperparameters
- The number of random features as resource, 4 hyperparameters

Kernel based classifier for CIFAR-10

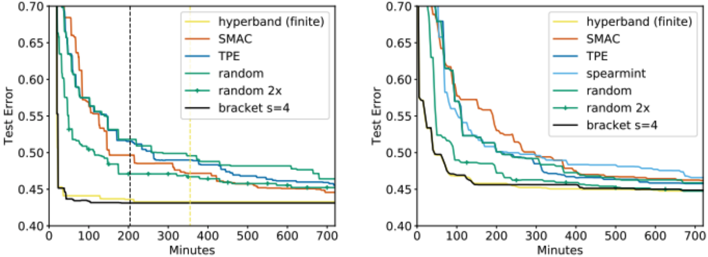


Figure 7: Left: number of training samples as resource. Right: number of random features as resource.

Discussion

- Successive halving seems to outperform hyperband
- Hyperband leverages downsampling to boost the number of configurations that are evaluated, and thus is better suited for hard problems

FLAML: Fast and lightweight AutoML library

Optimized for low computational resource

1. Chooses a learner/model l with probability $p_l = 1/ECI(l)$
   - $ECI(l)$ is the expected cost for improvement for l
2. For the selected learner, select hyperparameter and training sample size by perform cost-effective optimization.
3. Update $ECI(l)$ for learner l

FLAML versus HpBandSter in the same search space

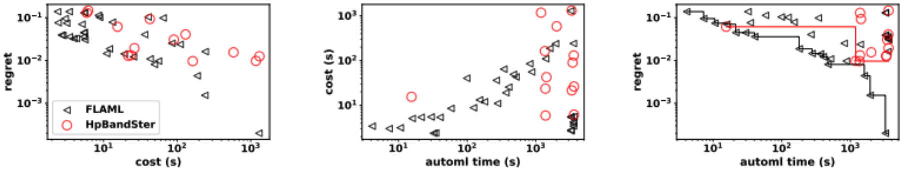


Figure 8: Each marker corresponds to one trial of configuration evaluation in a particular method. (a) suggests that FLAML makes fewer expensive trials with high error. (b) further displays that the expense of trials made by FLAML grows gradually with total time spent. (c) shows that FLAML outperforms in both early and late stages.