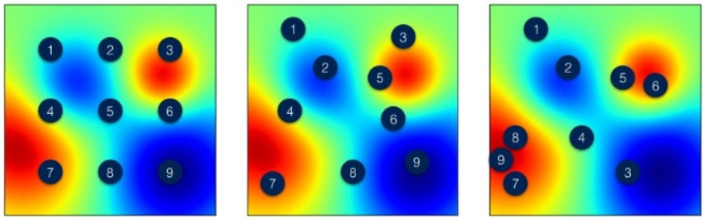
Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization

Tian Liu

Li, Lisha, et al. "Hyperband: A novel bandit-based approach to hyperparameter optimization." *The Journal of Machine Learning Research* 18.1 (2017): 6765-6816.

Background of Hyperparameter Optimization

- ***"What is the best hyperparameter configuration?"***
- Configuration selection
  - Brute-force: grid search, random search
  - Adaptive selection: Bayesian Optimization methods (SMAC,TPE, etc.)
  - Computational cost increases drastically with more hyperparameters



Grid Search Random Search Adaptive Selection

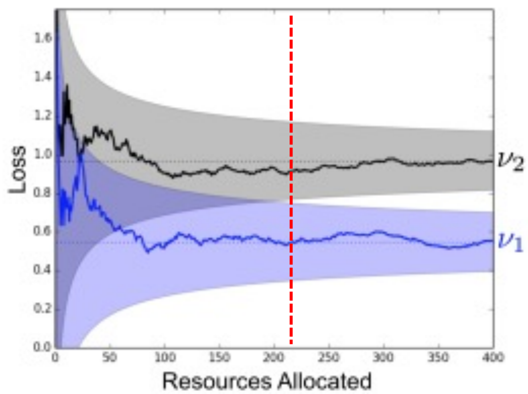
(https://blog.ml.cmu.edu/2018/12/12/massively-parallel-hyperparameter-optimization/)

- Configuration evaluation
  - Adaptive resource allocation: Successive Halving Algorithm
  - Aiming to examine more configurations with limited budget

- Hyperband Algorithm
  - Based on randomly sampled hyperparameter configurations
  - Principled early-stopping strategy to allocate resources based on Successive Halving
  - Evaluate order-of-magnitude more configurations than black-box methods
  - General-purpose technique for various machine learning models

Motivations: n vs. B/n Trade-off

- Successive Halving Algorithm requires input of n: $number\ of\ configs$
- Limited resource budget B
  - large n gives small averaging training time $\frac{B}{n}$



- Problem
  - Amount of resources required to differentiate two configs are unknown
  - Large n and small $\frac{B}{n}$, or small n and large $\frac{B}{n}$?
- Solution via Hyperband
  - Considering different trade-offs between n and $\frac{B}{n}$
  - Perform grid search over feasible n

Hyperband Algorithm

Algorithm 1: Hyperband algorithm for hyperparameter optimization.

input : R, η (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**
  $n = \lceil \frac{B}{R} \frac{\eta^s}{(s+1)} \rceil, \quad r = R\eta^{-s}$
  // 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.*

Outer loop: try different n

Inner loop: SH with fixed n

- R: max resource can be allocated to a single configuration
  - Time, Data Set Subsampling, Feature Subsampling
  - overhead cost < R ≤ Natural upper bound
- η: discard proportion
  - Results not sensitive to η
  - Choose 3 or 4 for practical use
- Each inner loop use around B resources, causing total cost of $(s_{max} + 1)B$

- **Example: R = 81 iterations, $\eta = 3$**

Most aggressive / exploratory strategy → Least aggressive strategy (Random Search)

	$s=4$		$s=3$		$s=2$		$s=1$		$s=0$	
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								

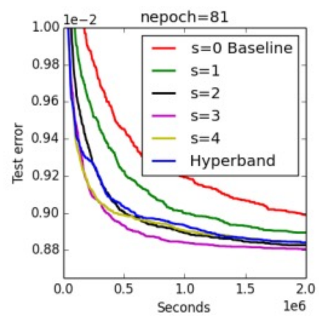
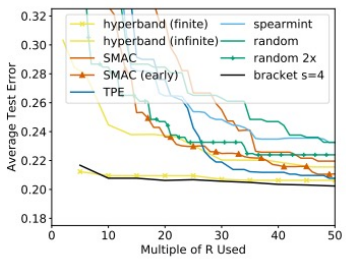


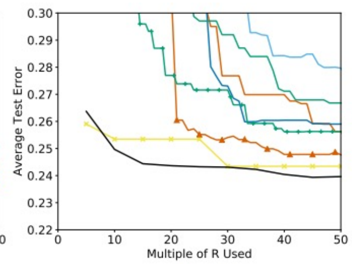
Figure 3: Performance of individual brackets s and Hyperband.

Experiments with Various Resource Types

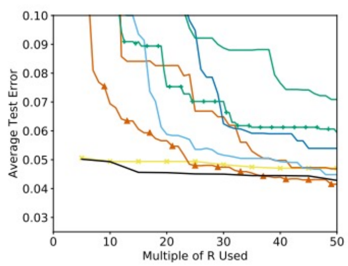
- *Early stop with iterations for DNN*



(a) CIFAR-10



(b) MRBI



(c) SVHN

- *Data set subsamples*

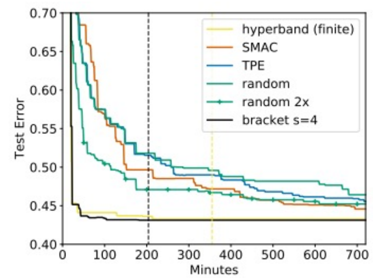


Figure 7: Average test error of the best kernel regularized least square classification model found by each searcher on CIFAR-10. The color coded dashed lines indicate when the last trial of a given searcher finished.

- *Feature samples*

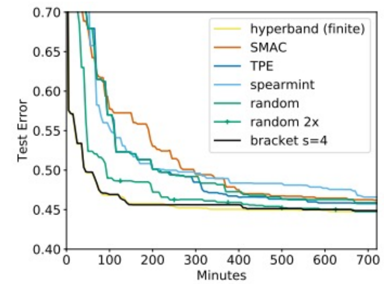


Figure 8: Average test error of the best random features model found by each searcher on CIFAR-10. The test error for HYPERBAND and bracket $s = 4$ are calculated in every evaluation instead of at the end of a bracket.

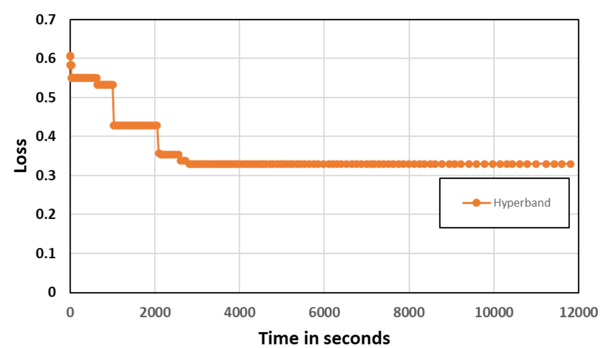
Implementation of Hyperband

- Optimizing an ANN for predicting the exit of a bank customer w/ 10,000 customer records (Churn Modelling Data)
- 8,000 for training, 2,000 for testing, $R = 81$ iterations, $\eta = 3$
- Hyperparameters (HP) to optimize: # of hidden layers ([1,5]), # of nodes for each layer ([2,200])
- Fixed HPs: init='uniform', batch_size='256', optimizer='adam', activation='relu'

RowNumt	Customer	Surname	CreditScoı	Geograph	Gender	Age	Tenure	Balance	NumOfPrı	HasCrCard	IsActiveM	Estimated	Exited
1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101348.9	1
2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.6	0
3	15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.6	1
4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.63	0
5	15737888	Mitchell	850	Spain	Female	43	2	125510.8	1	1	1	79084.1	0
6	15574012	Chu	645	Spain	Male	44	8	113755.8	2	1	0	149756.7	1
7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8	0
8	15656148	Obinna	376	Germany	Female	29	4	115046.7	4	1	0	119346.9	1
9	15792365	He	501	France	Male	44	4	142051.1	2	0	1	74940.5	0
10	15592389	H?	684	France	Male	27	2	134603.9	1	1	1	71725.73	0

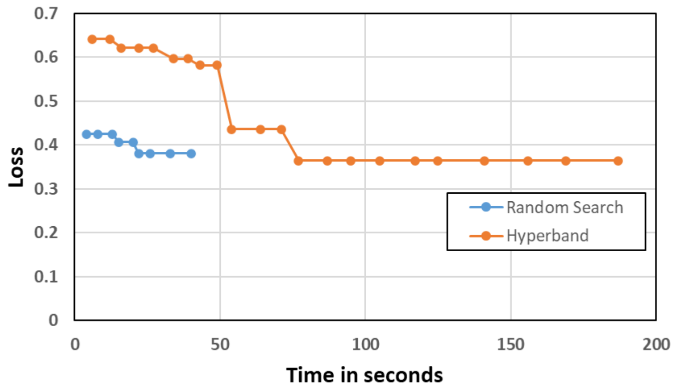
- Results: best 10 configs

Loss	auc	# of layers	# of nodes for layer 1	# of nodes for layer 2	Total # of Nodes
32.62%	87.45%	2	65	9	74
33.44%	86.88%	2	35	175	210
33.61%	86.47%	1	183		183
33.67%	86.41%	1	200		200
33.70%	86.39%	1	128		128
33.71%	86.36%	1	159		159
33.74%	86.39%	1	121		121
33.75%	86.72%	2	191	51	242
33.75%	86.30%	1	191		191
33.82%	86.25%	1	100		100

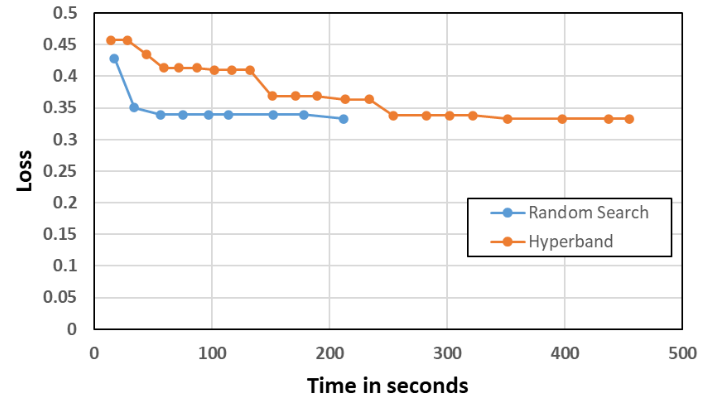


Hyperband vs. Random Search

- $R = 9$ iterations, $\eta = 3$, 2 hyperparameters



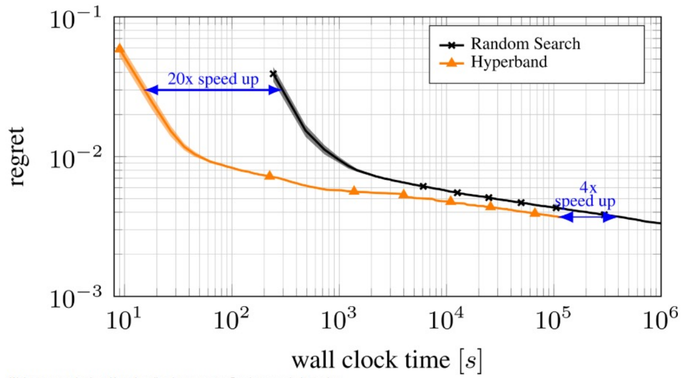
- $R = 9$ iterations, $\eta = 3$, n hyperparameters



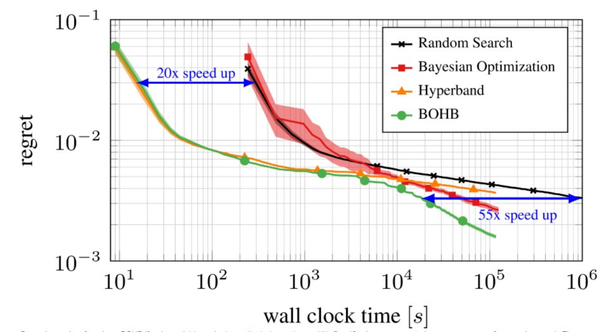
- Problem too simple, performance dominated by iteration number rather than configuration selection
- When searching hyperparameter number increases, the advantage of Hyperband starts to show up

Current and Future Work

- In small to medium budget, HB outperforms RS and BO. But in large budget, the advantages of HB over RS typically diminishes. BO may converge to global optimum faster than HB
- BOHB = Bayesian Optimization + Hyperband
- Future work: automatic adaptation of budget to alleviate misspecification by user



With increasing budget, Hyperband's advantage over Random search diminishes.



Example results of applying BOHB (freely available under https://github.com/automl/HpBandSter) to optimize six hyperparameters of a neural network. The curves show the immediate regret of the best configuration found by the methods as a function of time. BOHB combines the best of BO and Hyperband: it yields good performance (20x) faster than BO in the beginning, and converges much faster than Hyperband in the end.

(https://www.automl.org/blog_bohb/)

- *<u>Personal thought</u>: Genetic Algorithm + Hyperband =GAHB ?*

Falkner, Stefan, Aaron Klein, and Frank Hutter. "BOHB: Robust and efficient hyperparameter optimization at scale." arXiv preprint arXiv:1807.01774 (2018).