Towards Automated Deep Learning: Efficient Joint Neural Architecture & Hyperparameter Search

(arXiv:1807.06906v1 [cs.LG] 18 Jul 2018)

Toronto Deep Learning Series (TDLS)

Discussion Lead: Mark Donaldson
Discussion Facilitator: Masoud Hashemi

Ryerson University

TED ROGERS SCHOOL OF MANAGEMENT

December 10, 2018

Table of Contents

Ryerson University

TED ROGERS SCHOOL OF MANAGEMENT

Discussion Paper Abstract: Introduction to Problems

1. Neural Architecture search (NAS) tunes hyperparameters in a separate post-processing step, rendering this method suboptimal

2. Use of very few epochs during the main NAS and much larger numbers of epochs during a post-processing step is inefficient due to little correlation in relative rankings

Discussion Paper Abstract: Introduction to Solutions

1. Combination of Bayesian optimization and Hyperband (BOHB) for efficient joint neural architecture and hyperparameter search

Ryerson University TED ROGERS SCHOOL OF MANAGEMENT

Network Architecture Search (NAS): Problems

1. Early machine learning workflows had manual feature engineering which was time consuming and tedious to configure

2. Recent work with NAS provided automation of the choice of network architecture which lead to improved performance at extreme computational costs (up to 800 GPUs for two weeks!)

3. NAS did not promote an anytime approach in automated machine learning (AutoML) systems that make predictions after a given time budget

4. Jump from small budget of 20 to large budget of 600 epochs lead to little correlation between small & large training budgets

Network Architecture Search (NAS): Solutions

1. Combine Bayesian optimization (BO) and Hyperband (HB) to perform efficient joint neural architecture and hyperparameter search [best of both worlds solution] (BOHB)

2. Overcome weak correlation between performance after long training budgets (up to 3 hours / 10800 seconds) by incrementally increasing the training budget during the optimization process

3. Great results on CIFAR-10 after training budget of 3 hours / 10800 seconds by optimizing the hyperparameters and architecture jointly

Efficient Joint Hyperparameter Optimization and Architecture Search: ResNet Blocks & Wide Residual Networks (WRN)

WRN is important to increase representational power of residual blocks by…..

1. Adding more convolutional layers per block
2. Widening the convolutional layers by adding more feature planes
3. Increasing filter sizes in convolutional layers

Efficient Joint Hyperparameter Optimization and Architecture Search: ResNet Blocks & Wide Residual Networks (WRN)

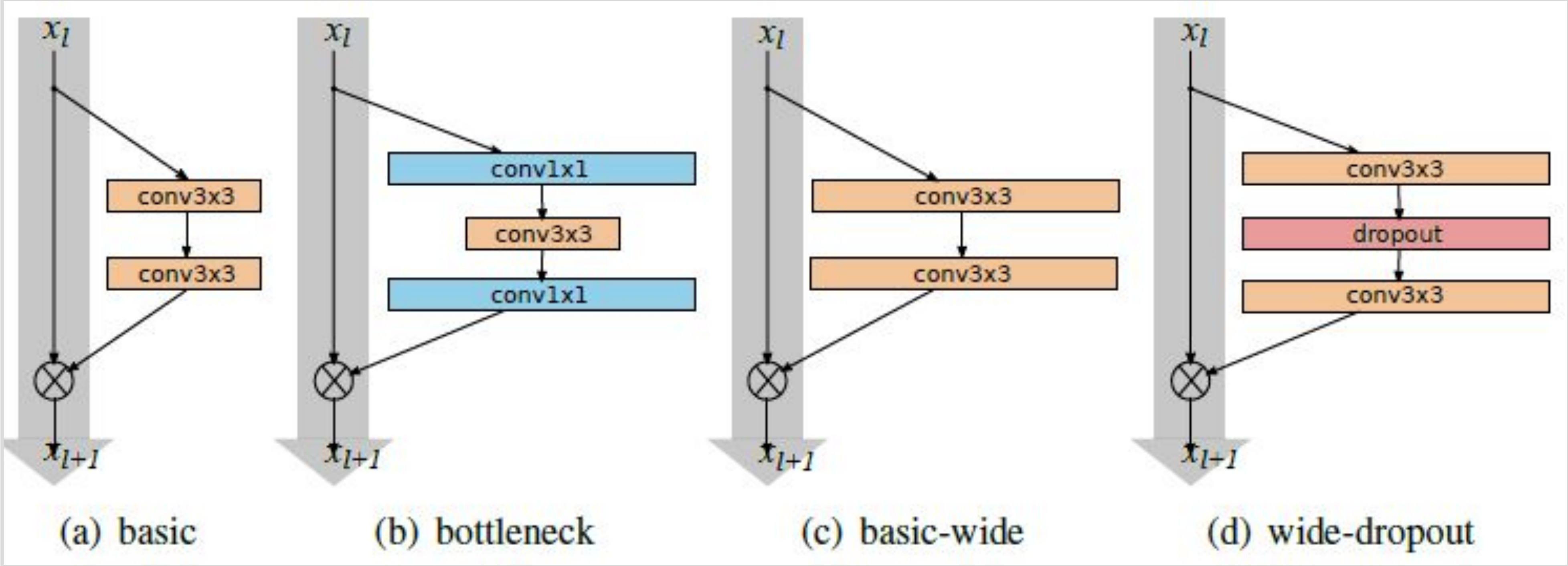


Fig. 1

Efficient Joint Hyperparameter Optimization and Architecture Search: ResNet Blocks & Wide Residual Networks (WRN)

Basic ResNet - with two consecutive 3 X 3 convolutions with batch normalization and ReLU preceding convolution: conv 3 X 3 - conv 3, shown in Fig 1, (a) basic

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l)$$

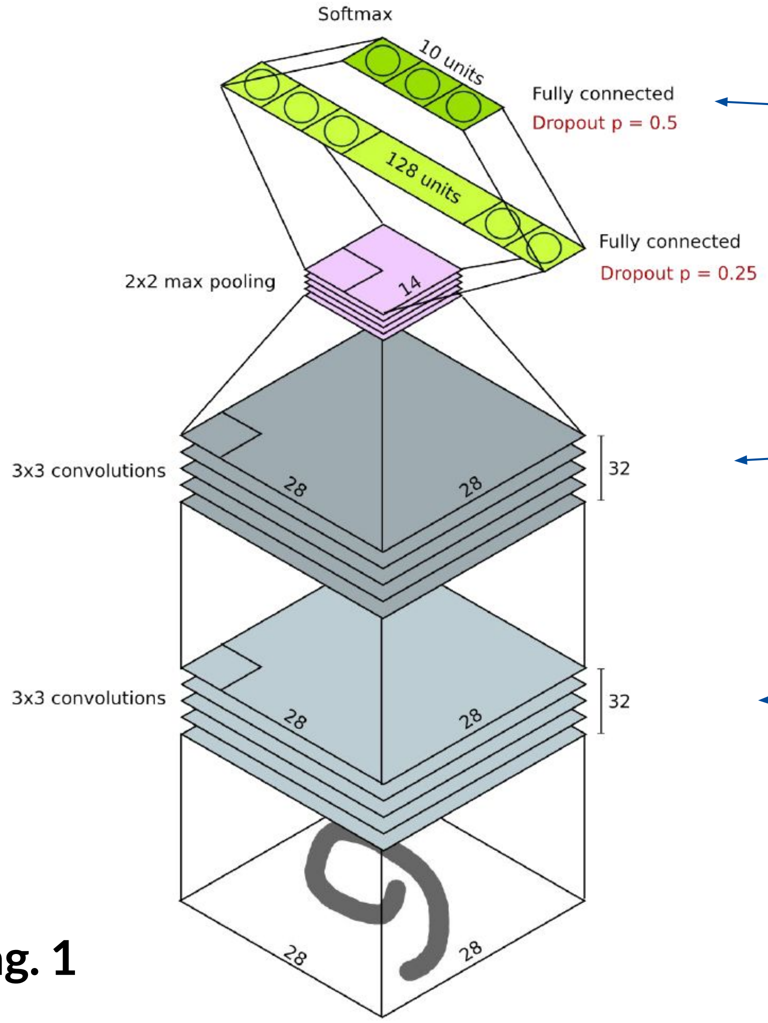
Eq. 1

group name	output size	block type = $B(3,3)$
conv1	32×32	$[3\times3, 16]$
conv2	32×32	$\begin{bmatrix} 3\times3, 16\times k \\ 3\times3, 16\times k \end{bmatrix} \times N$
conv3	16×16	$\begin{bmatrix} 3\times3, 32\times k \\ 3\times3, 32\times k \end{bmatrix} \times N$
conv4	8×8	$\begin{bmatrix} 3\times3, 64\times k \\ 3\times3, 64\times k \end{bmatrix} \times N$
avg-pool	1×1	$[8 \times 8]$

Table 1

Efficient Joint Hyperparameter Optimization and Architecture Search: ResNet Blocks & Wide Residual Networks (WRN)

Pictorial representation of WRN layers and structure



Diag. 1

WRN Layers:

classification

avg-pool

conv4

conv3

conv1 & 2 (not shown)

Efficient Joint Hyperparameter Optimization and Architecture Search: ResNet Blocks & Wide Residual Networks (WRN)

Performance of Block structure B(3 x 3) [Original basic block]

block type	depth	# params	time,s	CIFAR-10
$B(1,3,1)$	40	1.4M	85.8	6.06
$B(3,1)$	40	1.2M	67.5	5.78
$B(1,3)$	40	1.3M	72.2	6.42
$B(3,1,1)$	40	1.3M	82.2	5.86
$B(3,3)$	28	1.5M	67.5	5.73
$B(3,1,3)$	22	1.1M	59.9	5.78

Table 2

l	CIFAR-10
1	6.69
2	5.43
3	5.65
4	5.93

Table 3

Efficient Joint Hyperparameter Optimization and Architecture Search: ResNet Blocks & Wide Residual Networks (WRN)

Test error (%) results of various ‘k’ widening factors on CIFAR-10

depth	k	# params	CIFAR-10
40	1	0.6M	6.85
40	2	2.2M	5.33
40	4	8.9M	4.97
40	8	35.7M	4.66
28	10	36.5M	**4.17**
28	12	52.5M	4.33
22	8	17.2M	4.38
22	10	26.8M	4.44
16	8	11.0M	4.81
16	10	17.1M	4.56

Table 4

Efficient Joint Hyperparameter Optimization and Architecture Search: ResNet Blocks & Wide Residual Networks (WRN)

Training curve of CIFAR-10

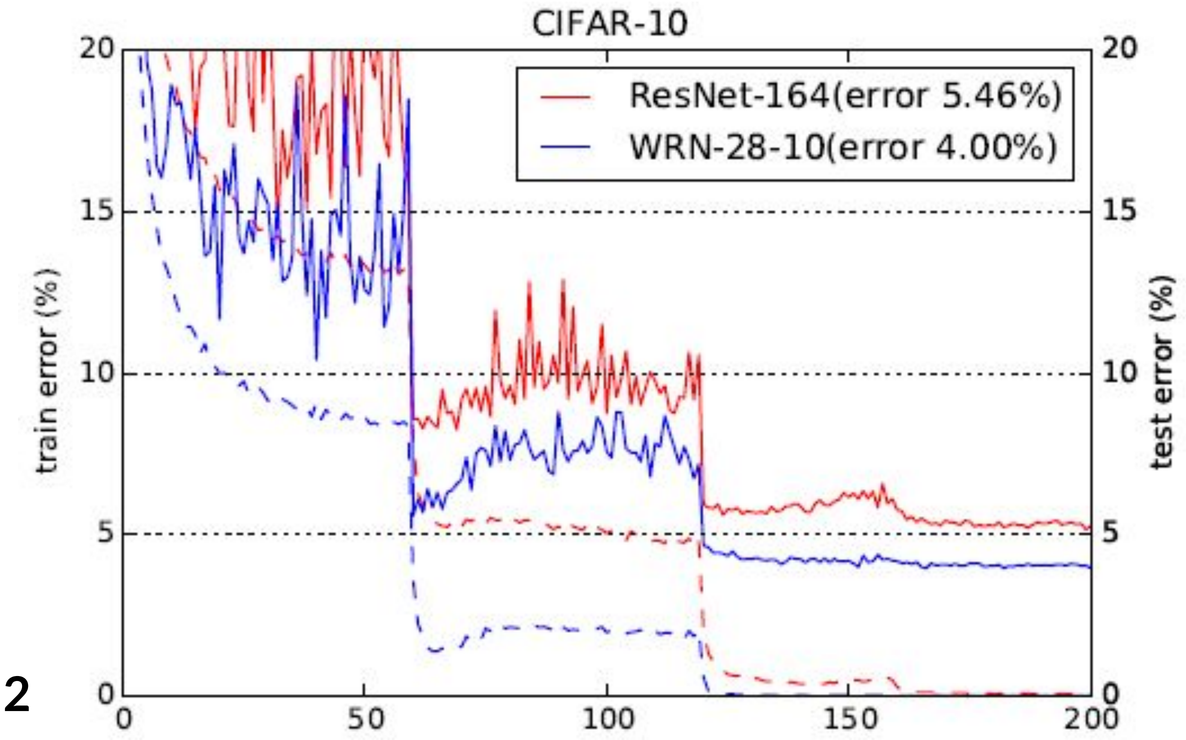


Fig. 2

Efficient Joint Hyperparameter Optimization and Architecture Search: ResNet Blocks & Wide Residual Networks (WRN)

Resulting benefits of using WRN architecture:

1. Widening consistently improves performance across residual networks of different depths

2. Increasing both depth and width helps until the number of parameters becomes too high and stronger regularization is needed

3. Regularization effect from very high depth RNs as WRNs with same number of parameters as thin WRNs can learn same or better representations

Efficient Joint Hyperparameter Optimization and Architecture Search: ResNet Blocks & Wide Residual Networks (WRN)

Resulting benefits of using WRN architecture:

4. WRNs can successfully learn with a 2 or more times larger number of parameters as thin RNs, besting thin RNs which would require doubling thin RN depth, making them unfeasibly expensive to train

Efficient Joint Hyperparameter Optimization and Architecture Search: Bayesian Optimization and Hyperband (BOHB)

Goals of using BOHB:

1. Strong anytime performance
2. Strong final performance
3. Effective use of parallel resources
4. Scalability
5. Robustness and flexibility
6. Simplicity
7. Computational efficiency

Efficient Joint Hyperparameter Optimization and Architecture Search: Bayesian Optimization and Hyperband (BOHB)

Bayesian Optimization (BO)

- Used to build a model that can be updated and queried to drive optimization decisions (training)

Real World Applications of BO

1. A/B Testing
2. Recommender Systems
3. Robotics and Reinforcement Learning
4. Environmental Monitoring and Sensor Networks
5. Preference Learning and Interactive Interfaces
6. Automatic Machine Learning and Hyperparameter Tuning
7. Combinatorial Optimization
8. Natural Language Processing and Text

Efficient Joint Hyperparameter Optimization and Architecture Search: Bayesian Optimization and Hyperband (BOHB)

Bayesian Optimization (BO)

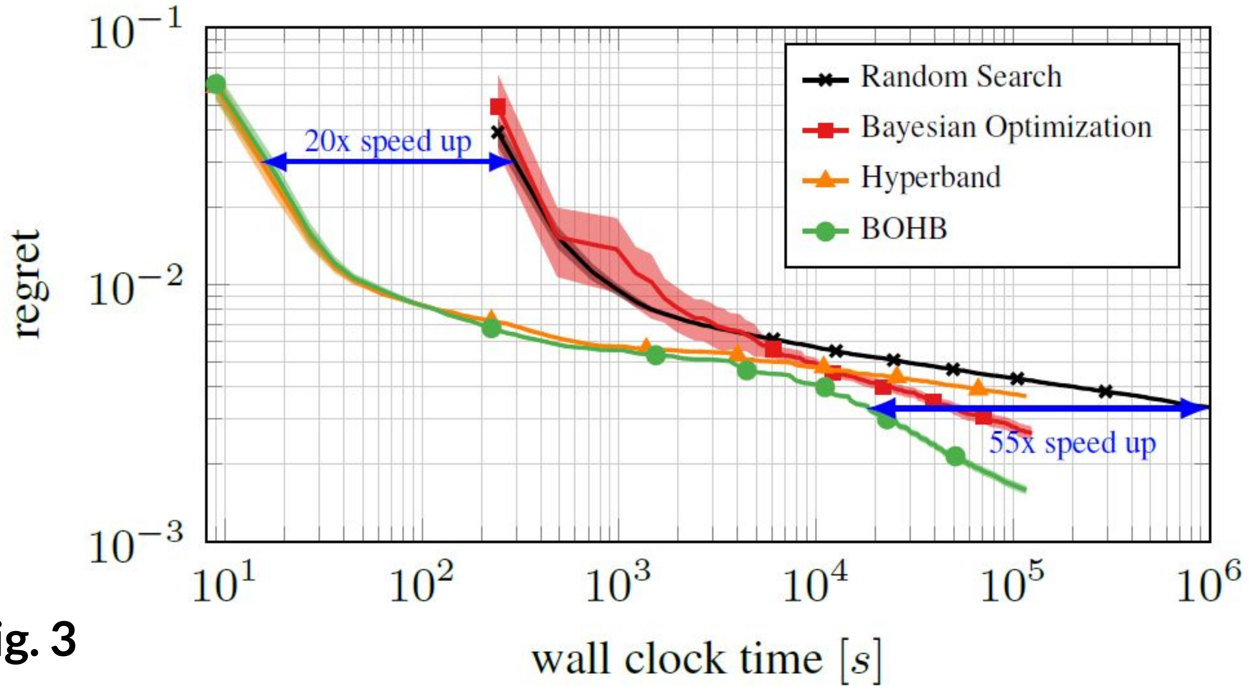


Fig. 3

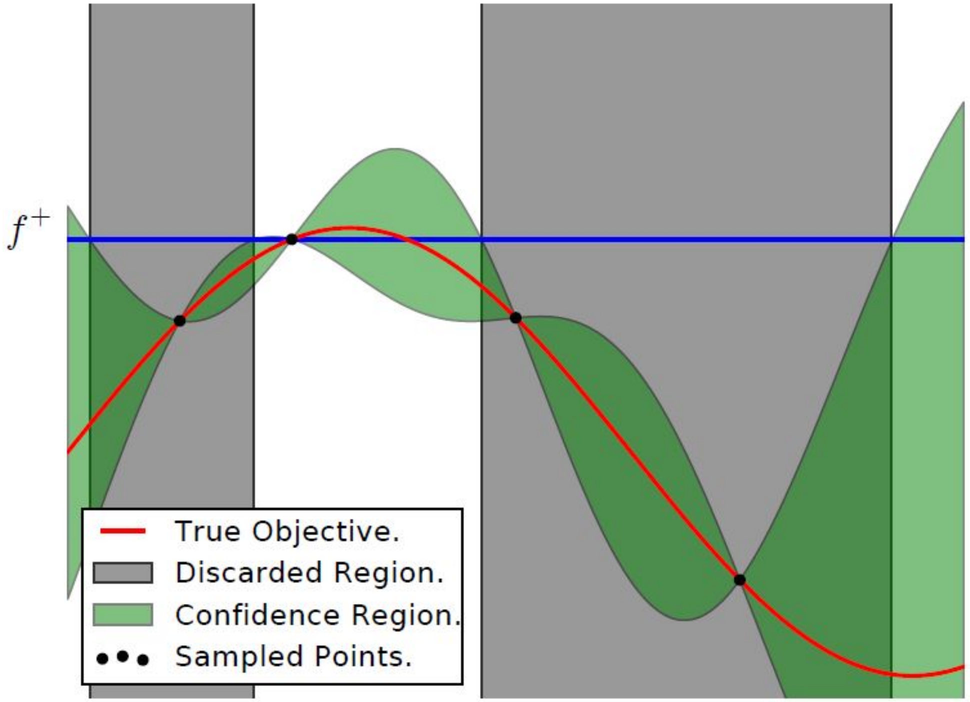


Fig. 4

Efficient Joint Hyperparameter Optimization and Architecture Search: Bayesian Optimization and Hyperband (BOHB)

Hyperband (HB)

- Uses ‘Successive Halving’ while performing random search using ‘shake-shake’ regularization method
- Finds the local minima very quickly using ‘Successive Halving’ method to reduce the WRN or ResNet training budget

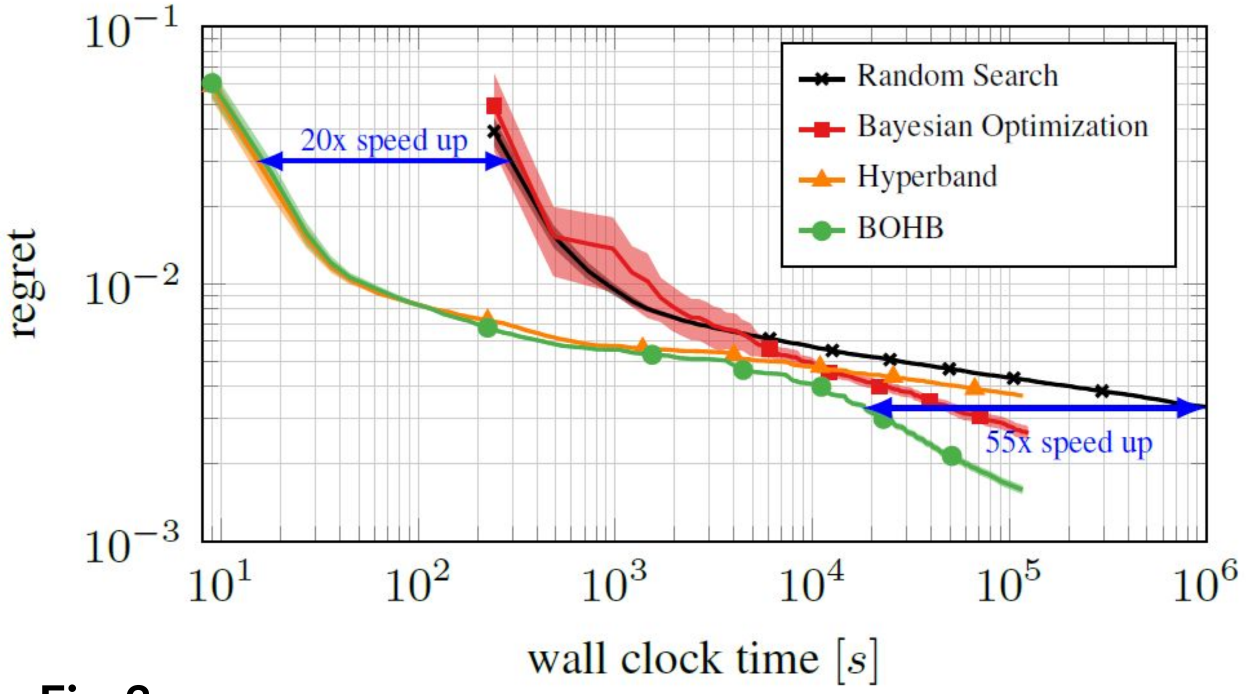


Fig. 3

Ryerson University

TED ROGERS SCHOOL OF MANAGEMENT

Efficient Joint Hyperparameter Optimization and Architecture Search: Bayesian Optimization and Hyperband (BOHB)

Hyperband (HB)

Requires 2 key inputs:

1. R, maximum amount of resource that can be allocated to a single config
2. η, an input (tuning) that controls the proportion of configs discarded in each round of 'Successive Halving'

Algorithm 1: HYPERBAND algorithm for hyperparameter optimization.

input : R, η (default $\eta = 3$)
initialization : $\boxed{s_{\max} = \lfloor \log_\eta(R) \rfloor}$, $B = (s_{\max} + 1)R$

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.*

Fig. 5

$$B = (\lfloor \log_\eta(R) \rfloor + 1)R$$

Eq. 2

Ryerson University

TED ROGERS SCHOOL OF MANAGEMENT

Efficient Joint Hyperparameter Optimization and Architecture Search: Bayesian Optimization and Hyperband (BOHB)

Hyperband (HB)

Table 5

i	$s=4$ n_i	$s=4$ r_i	$s=3$ n_i	$s=3$ r_i	$s=2$ n_i	$s=2$ r_i	$s=1$ n_i	$s=1$ r_i	$s=0$ n_i	$s=0$ 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								

Values of n_i and r_i for the brackets of Hyperband when R = 81 and η =3

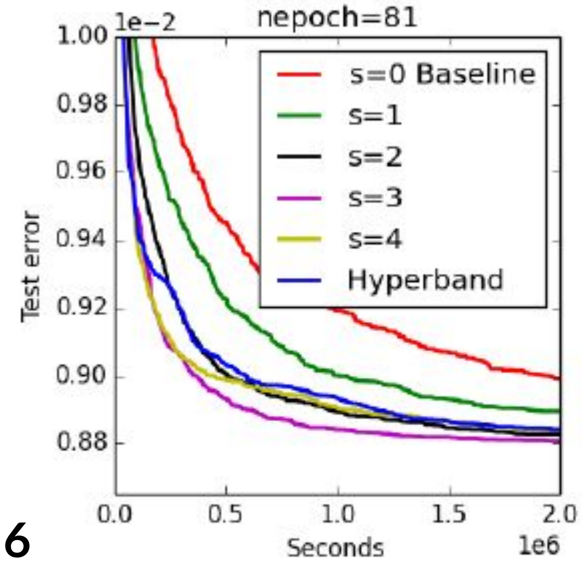


Fig. 6

Performance of individual brackets S and Hyperband

Efficient Joint Hyperparameter Optimization and Architecture Search: Conclusion

What did we learn….?

- Neural Architecture (WRNs for example) is an important factor in parameter search performance using BOHB
- Combining 2 or more optimization methods will yield better search results, faster performance with minimal training error
- Hyperparameter optimization during training using WRN and BOHB yields optimal results at a lesser cost expense of performance, time and low training error (%)

Efficient Joint Hyperparameter Optimization and Architecture Search: Paper Discussion Topics

Open for further discussion:

1. Shake-shake WRN Regularization method of various ResNet dimensions [R3X3] error (%)
2. How to generalize the NAS BOHB parameter settings to start at the optimum settings for a given dataset to classify
3. Neural Architecture Search (NAS) and BOHB classifier performance applied to various data types and data sets (Images [CIFAR-10], NLP [Text data or text to speech and vice versa], Voice [IVR systems], Medical Images [Pathology classification of X-Ray images])

Efficient Joint Hyperparameter Optimization and Architecture Search: References used

S. Falkner, A. Klein, and F. Hutter. Practical hyperparameter optimization for deep learning. ICLR 2018 Workshop, 2018.

S. Falkner, A. Klein, and F. Hutter. BOHB: Robust and Efficient Hyperparameter Optimization at Scale. ICML 2018 Stockholm, Sweden, PMLR 80, 2018.

L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: Bandit-based configuration evaluation for hyperparameter optimization. 2017.

B. Shahriari, K. Swersky, Z. Wang, R. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. Proceedings of the IEEE, 104(1):148{175, 2016.

Xavier Gastaldi. Shake-shake regularization. ICLR 2017 Workshop, 2017.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. CoRR, abs/1605.07146, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In CVPR, 2016.

L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: Bandit-based configuration evaluation for hyperparameter optimization. 2017.

Efficient Joint Hyperparameter Optimization and Architecture Search: References used

Giro-i-Nieto, X., Sayrol, E., Salvador, A., Torres, J., Mohedano, E., & McGuinness, K. (2016). *Deep Learning for Computer Vision, Summer Seminar UPC TelecomBCN*(Tech.). Barcelona: Universitat Politecnica de Catalunya Barcelona Tech. doi:http://imatge-upc.github.io/telecombcn-2016-dlcv/slides/D2L1-memory.pdf; Neural Net Model image used.

Contact me at:

Email: mark.donaldson@ryerson.ca

LinkedIn: https://www.linkedin.com/in/markdonaldson888/

Twitter: @markdheilong

Ryerson University

TED ROGERS SCHOOL OF MANAGEMENT

Efficient Joint Hyperparameter Optimization and Architecture Search: Bayesian Optimization and Hyperband (BOHB)

Bayesian Optimization (BO)

$$\mathcal{D}$$

$$p(\mathbf{w})$$

$$p(\mathcal{D})$$

$$p(\mathcal{D} \mid \mathbf{w})$$

$$p(\mathbf{w} \mid \mathcal{D})$$

$$p(\mathbf{w} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}.$$

Ryerson University

TED ROGERS SCHOOL OF MANAGEMENT