Google JOHNS HOPKINS UNIVERSITY Stanford University

Progressive Neural Architecture Search

Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, Kevin Murphy

09/10/2018 @ECCV

Outline

- Introduction and Background
- Architecture Search Space
- Progressive Neural Architecture Search Algorithm
- Experiments and Results

Introduction and Background

AutoML

- Hit Enter, sit back and relax, come back the next day for a **high-quality machine learning solution ready to be delivered**

What Is Preventing Us?

Machine Learning solution	Parameter	Hyperparameter
Neural Network		

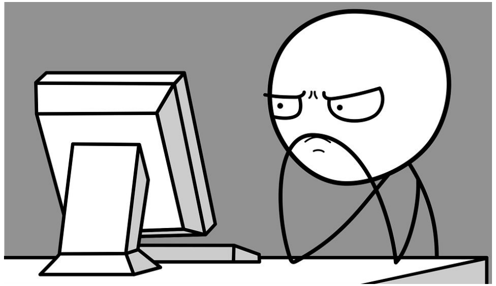
What Is Preventing Us?

Machine Learning solution	Parameter	Hyperparameter
Neural Network	Automated :)	

What Is Preventing Us?

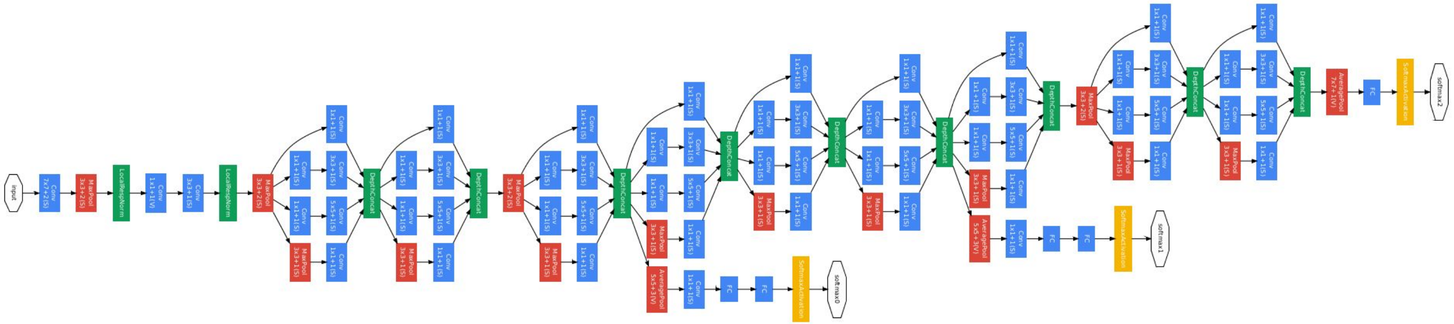
Key of AutoML

Machine Learning solution	Parameter	Hyperparameter
Neural Network	Automated :)	Not quite automated :(



Where Are Hyperparameters?

- We usually think of those related to learning rate scheduling
- But for a neural network, many more lie in its architecture:



Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In CVPR. 2015.

Neural Architecture Search (NAS)

- Can we design network architectures automatically, instead of relying on expert experience and knowledge?

- Broadly, existing NAS literatures fall into two main categories:
    - Evolutionary Algorithms (EA)
    - Reinforcement Learning (RL)

Evolutionary Algorithms for NAS

Best candidates

(0, 1, 0, 1): 0.85
(2, 0, 3, 1): 0.84
(5, 1, 3, 3): 0.91
(0, 2, 0, 6): 0.92
...
(0, 7, 3, 5): 0.82

String that defines network architecture

Accuracy on validation set

Evolutionary Algorithms for NAS

Best candidates

(0, 1, 0, 1): 0.85
(2, 0, 3, 1): 0.84
(5, 1, 3, 3): 0.91
(0, 2, 0, 6): 0.92
...
(0, 7, 3, 5): 0.82

→ mutate →

New candidates

(0, 1, 0, 2): ????
(2, 0, 4, 1): ????
(5, 5, 3, 3): ????
(0, 2, 1, 6): ????
...
(0, 6, 3, 5): ????

Evolutionary Algorithms for NAS

Best candidates

(0, 1, 0, 1): 0.85
(2, 0, 3, 1): 0.84
(5, 1, 3, 3): 0.91
(0, 2, 0, 6): 0.92
...
(0, 7, 3, 5): 0.82

New candidates

(0, 1, 0, 2): 0.86
(2, 0, 4, 1): 0.83
(5, 5, 3, 3): 0.90
(0, 2, 1, 6): 0.91
...
(0, 6, 3, 5): 0.80

Evolutionary Algorithms for NAS

Best candidates

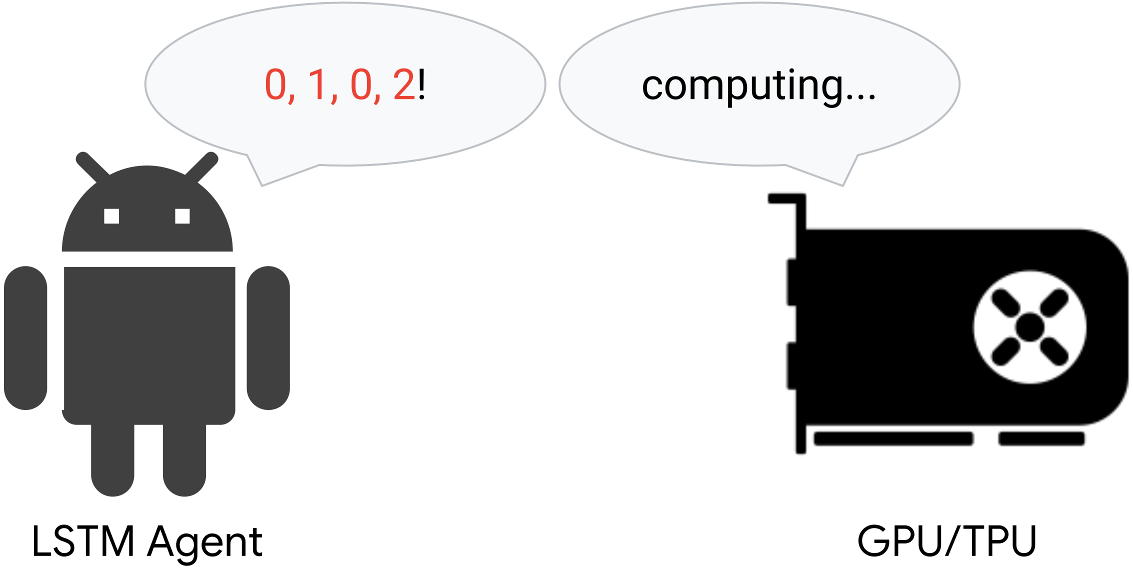
(5, 5, 3, 3): 0.90
(0, 2, 1, 6): 0.91
(5, 1, 3, 3): 0.91
(0, 2, 0, 6): 0.92
…
(0, 1, 0, 2): 0.86

merge

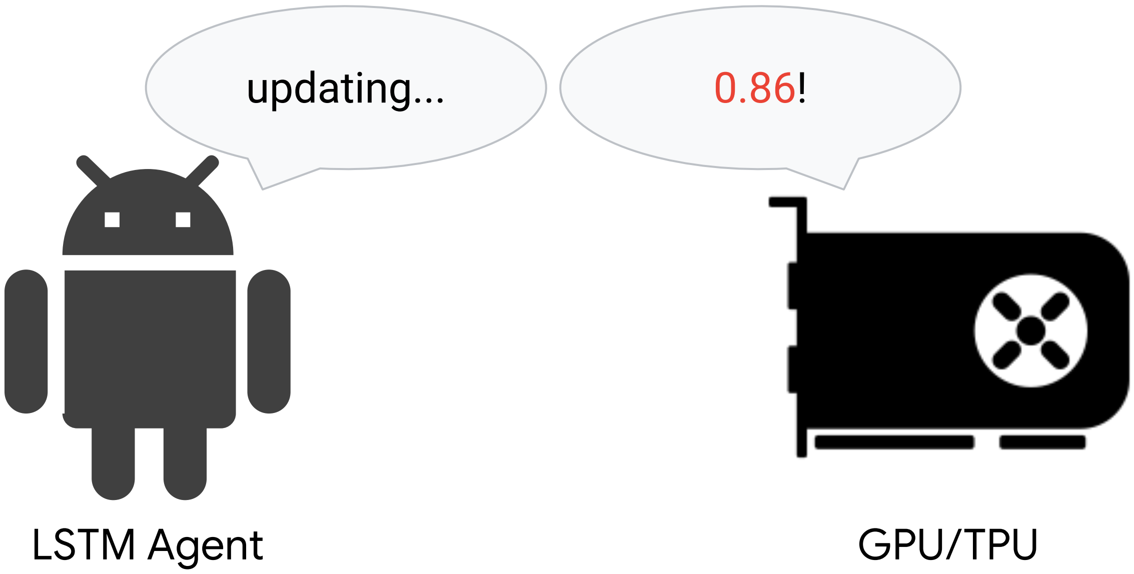
New candidates

(0, 1, 0, 2): 0.86
(2, 0, 4, 1): 0.83
(5, 5, 3, 3): 0.90
(0, 2, 1, 6): 0.91
…
(0, 6, 3, 5): 0.80

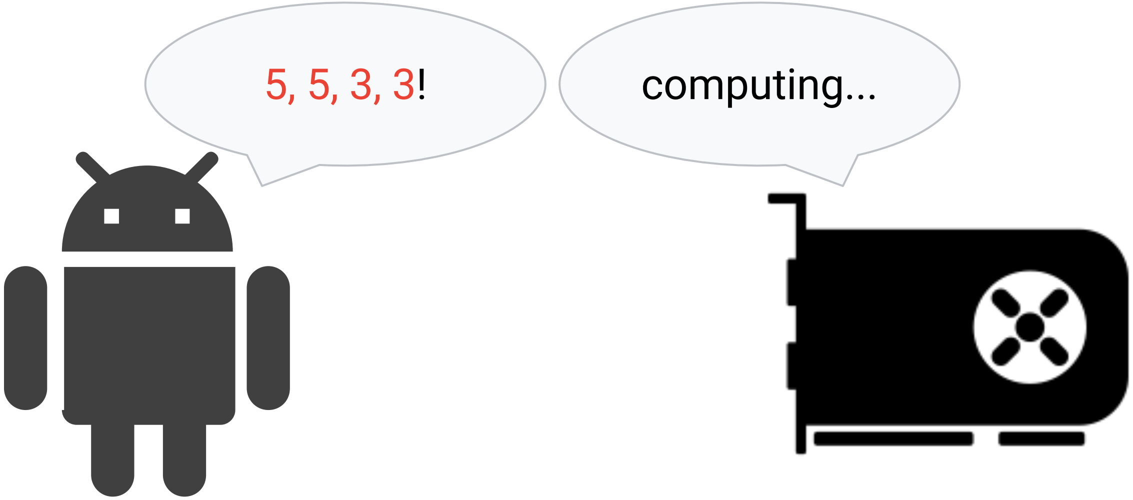
Reinforcement Learning for NAS



Reinforcement Learning for NAS



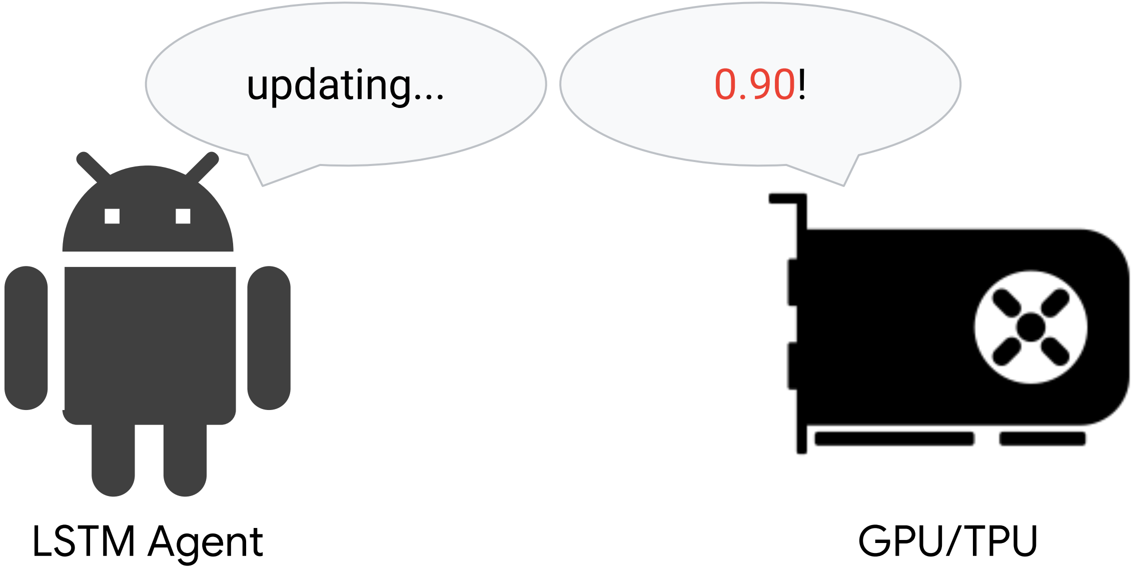
Reinforcement Learning for NAS



LSTM Agent

GPU/TPU

Reinforcement Learning for NAS



Success and Limitation

- NASNet from Zoph et al. (2018) already surpassed human designs on ImageNet under the same # Mult-Add or # Params
- But very computationally intensive:
  - Zoph & Le (2017): 800 K40 for 28 days
  - Zoph et al. (2018): 500 P100 for 5 days

Zoph, Barret, and Quoc V. Le. "Neural architecture search with reinforcement learning." In ICLR. 2017.
Zoph, Barret, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. "Learning transferable architectures for scalable image recognition." In CVPR. 2018.

Our Goal

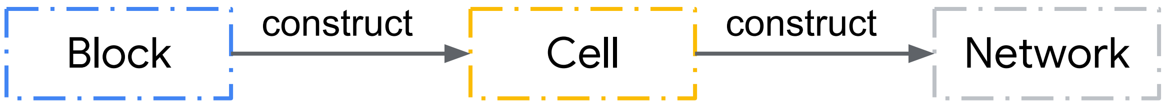
- NASNet from Zoph et al. (2018) already surpassed human designs on ImageNet under the same # Mult-Add or # Params
- But very computationally intensive:
    - Zoph & Le (2017): 800 K40 for 28 days
    - Zoph et al. (2018): 500 P100 for 5 days
- **Our goal: Speed up NAS by proposing an alternative algorithm**

Zoph, Barret, and Quoc V. Le. "Neural architecture search with reinforcement learning." In ICLR. 2017.
Zoph, Barret, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. "Learning transferable architectures for scalable image recognition." In CVPR. 2018.

Architecture Search Space

Taxonomy

- Similar to Zoph et al. (2018)



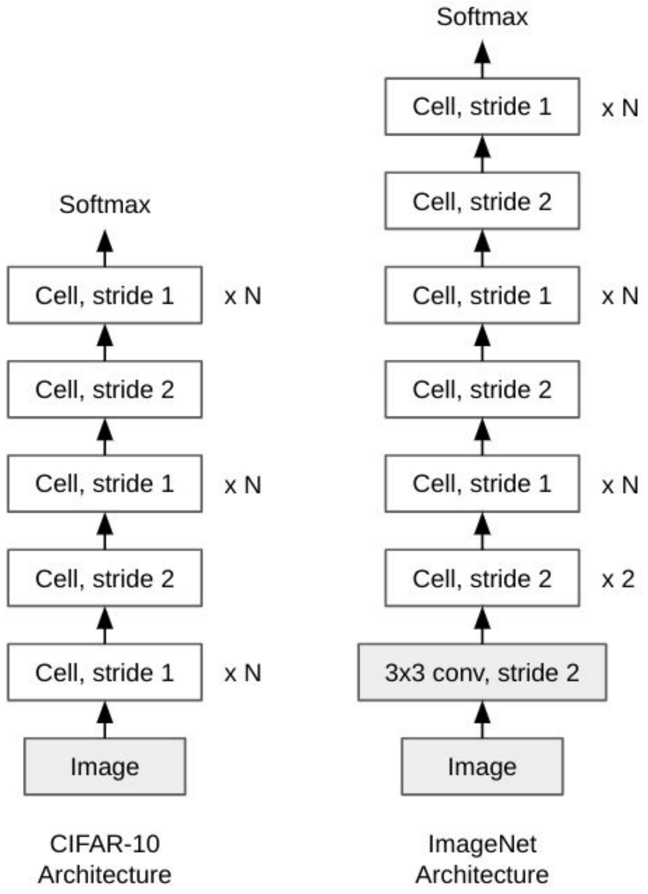
Zoph, Barret, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. "Learning transferable architectures for scalable image recognition." In CVPR. 2018.

Cell -> Network

- Once we have a cell structure, we stack it up using a predefined pattern

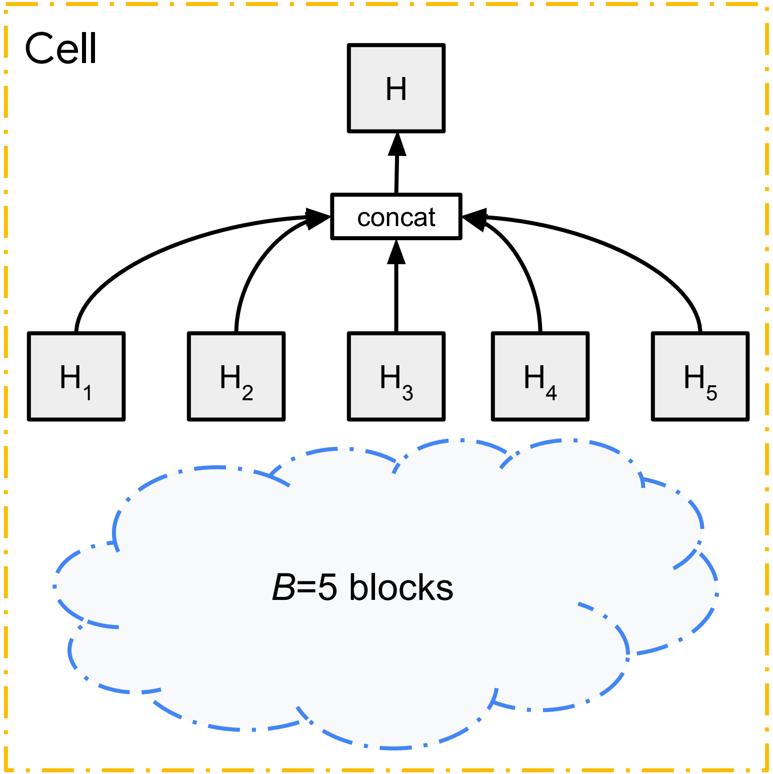
- A network is fully specified with:
  - Cell structure
  - *N* (number of cell repetition)
  - *F* (number of filters in the first cell)

- *N* and *F* are selected by hand to control network complexity



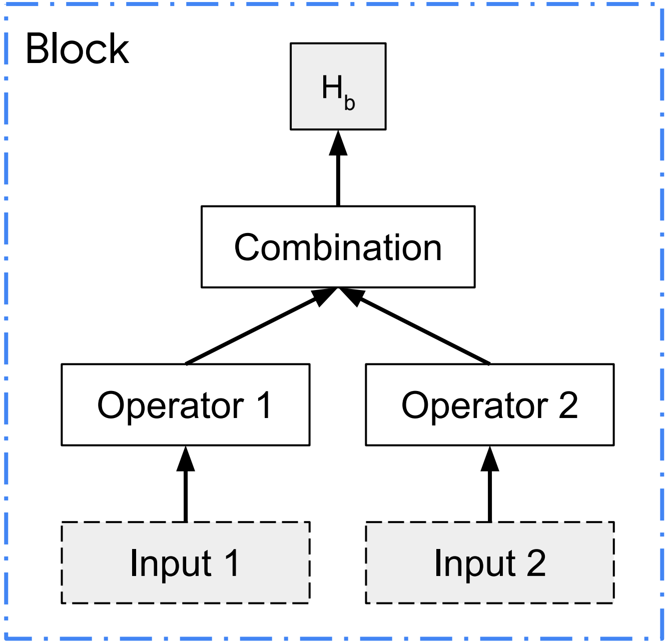
Block -> Cell

- Each cell consists of *B*=5 blocks
- The cell's output is the concatenation of the 5 blocks' outputs



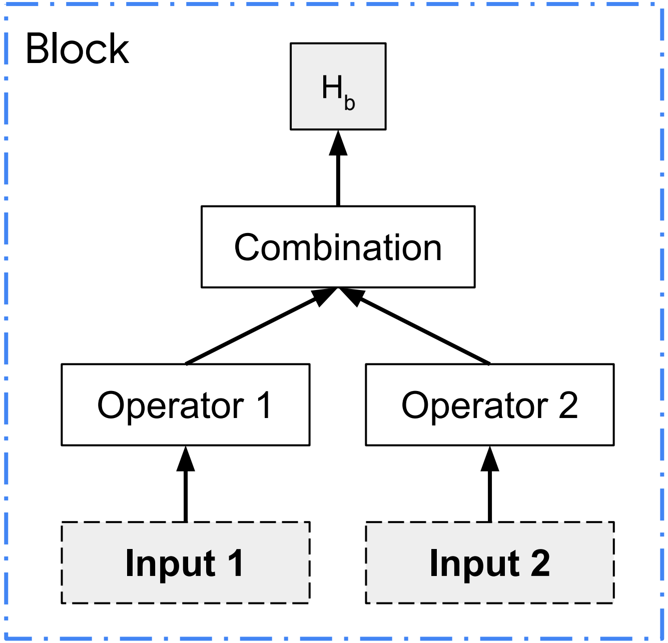
Within a Block

- Input 1 is transformed by Operator 1
- Input 2 is transformed by Operator 2
- Combine to give block's output



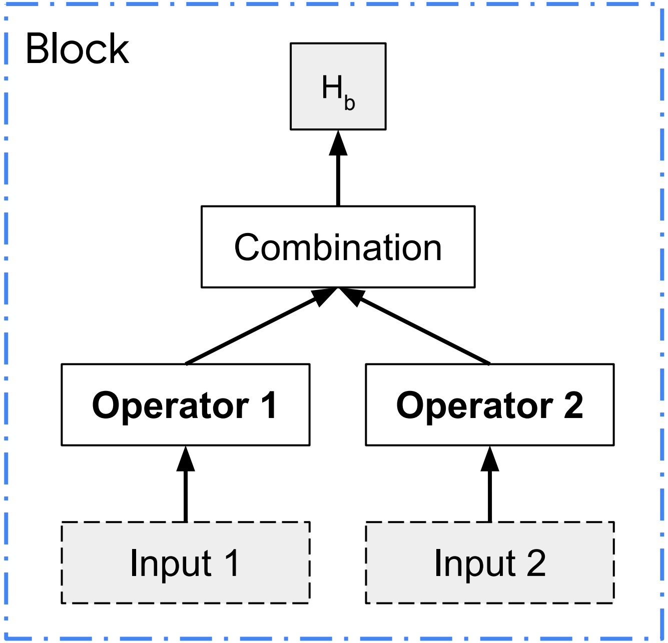
Within a Block

- **Input 1** and **Input 2** may select from:
    - Previous cell’s output
    - Previous-previous cell’s output
    - Previous blocks’ output in current cell



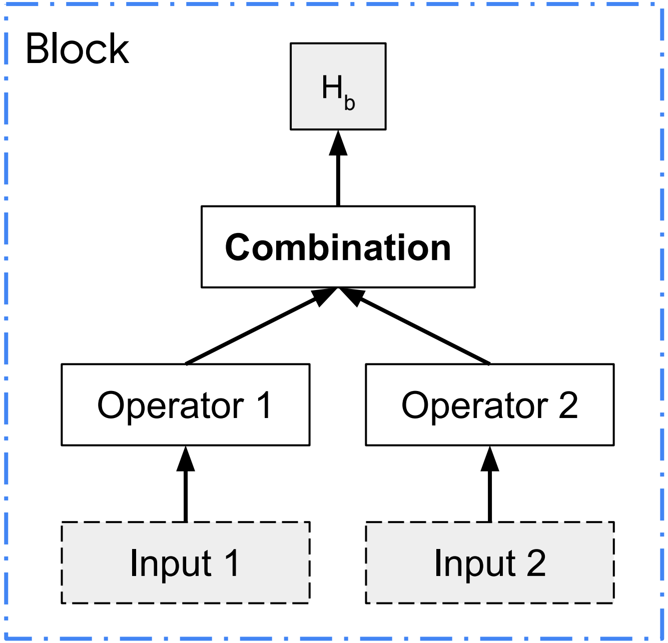
Within a Block

- **Operator 1** and **Operator 2** may select from:
  - 3x3 depth-separable convolution
  - 5x5 depth-separable convolution
  - 7x7 depth-separable convolution
  - 1x7 followed by 7x1 convolution
  - Identity
  - 3x3 average pooling
  - 3x3 max pooling
  - 3x3 dilated convolution



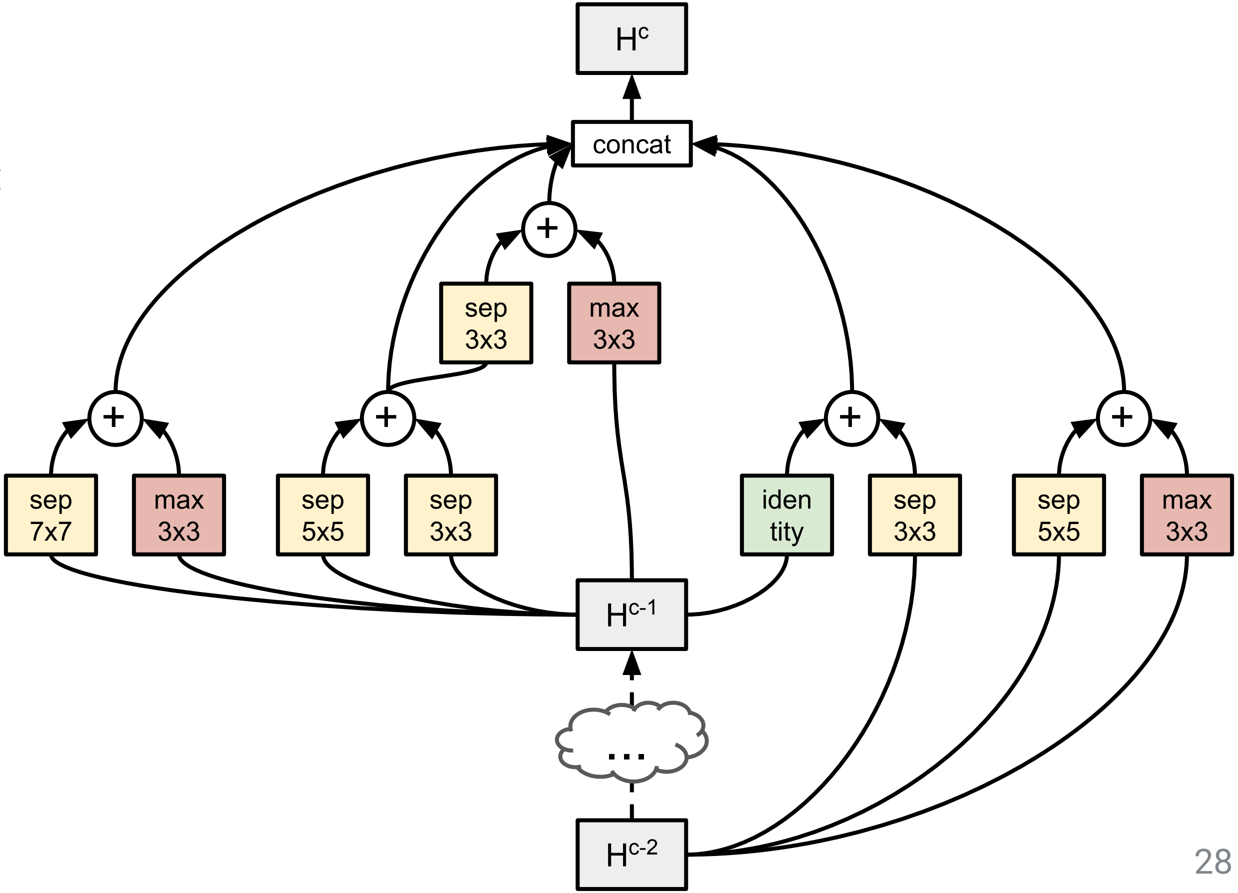
Within a Block

- **Combination** is element-wise addition



Architecture Search Space Summary

- One cell may look like:
- $2^2 * 8^2 * 1 *$
  $3^2 * 8^2 * 1 *$
  $4^2 * 8^2 * 1 *$
  $5^2 * 8^2 * 1 *$
  $6^2 * 8^2 * 1 =$
  10^{14} possible combinations!



Progressive Neural Architecture Search Algorithm

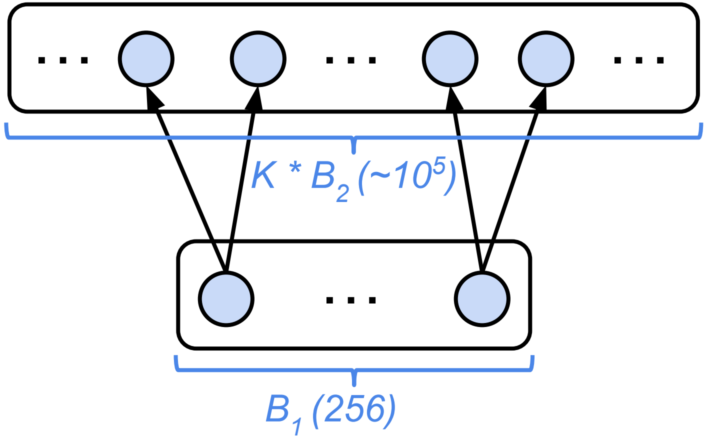
Main Idea: Simple-to-Complex Curriculum

- Previous approaches directly work with the 10^{14} search space

- Instead, what if we progressively work our way in:
    - Begin by training all 1-block cells. There are only 256 of them!
    - Their scores are going to be low, because of they have fewer blocks...
    - But maybe their relative performances are enough to show which cells are promising and which are not.
    - Let the *K* most promising cells expand into 2-block cells, and iterate!

Progressive Neural Architecture Search: First Try

- **Problem:** for a reasonable *K*, too many 2-block candidates to train
  - It is “expensive” to obtain the performance of a cell/string
  - Each one takes hours of training and evaluating
  - Maybe can afford 10^2, but definitely cannot afford 10^5



train these 2-block cells ✗

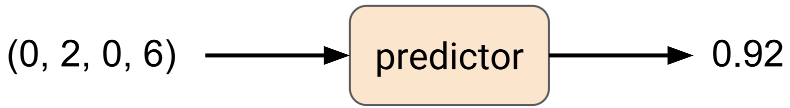
expand promising 2-block cells ✓

enumerate, train, select top *K* ✓

Performance Prediction with Surrogate Model

- **Solution**: train a “cheap” surrogate model that predicts the final performance simply by reading the string
    - The data points collected in the “expensive” way are exactly training data for this “cheap” surrogate model

- The two assessments are in fact used in an alternate fashion:
    - Use “cheap” assessment when candidate pool is large ($\sim 10^5$)
    - Use “expensive” assessment when it is small ($\sim 10^2$)



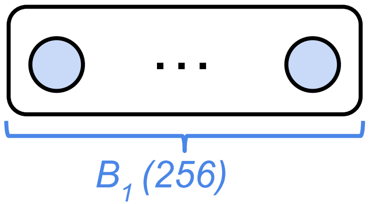
Performance Prediction with Surrogate Model

- Desired properties of this surrogate model/predictor:
    - Handle variable-size input strings
    - Correlate with true performance
    - Sample efficient

- We try both a MLP-ensemble and a RNN-ensemble as predictor
    - MLP-ensemble handles variable-size by mean pooling
    - RNN-ensemble handles variable-size by unrolling a different number of times

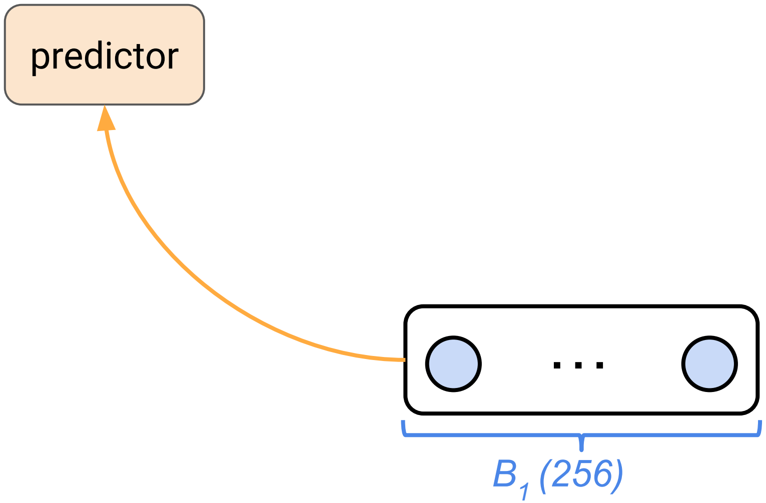
Progressive Neural Architecture Search

predictor



enumerate and train all 1-block cells

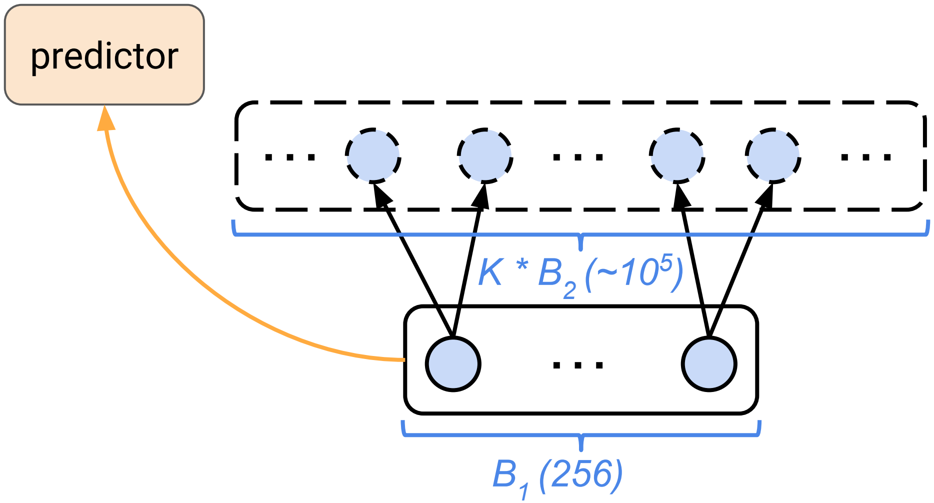
Progressive Neural Architecture Search



train predictor

enumerate and train all 1-block cells

Progressive Neural Architecture Search

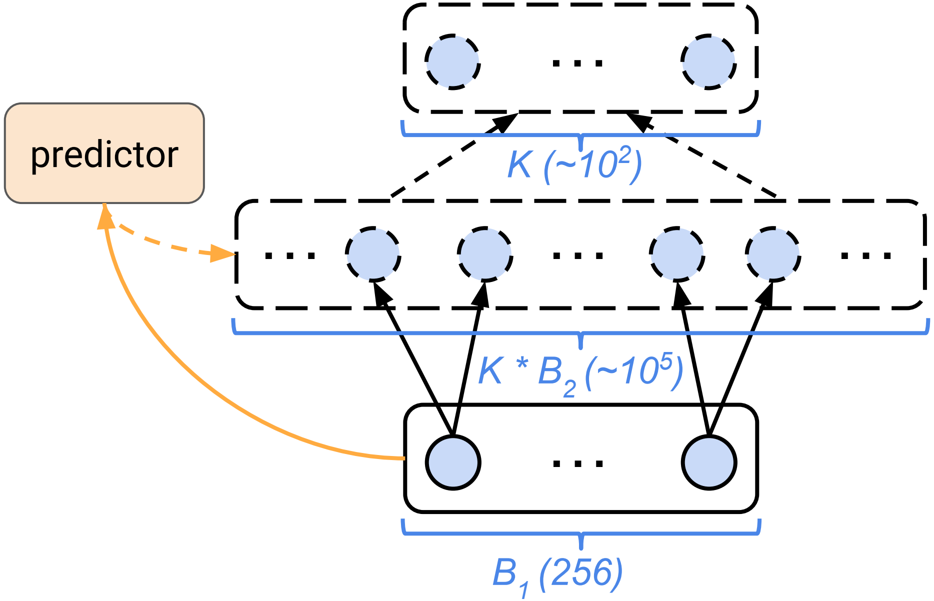


expand promising 2-block cells

train predictor

enumerate and train all 1-block cells

Progressive Neural Architecture Search



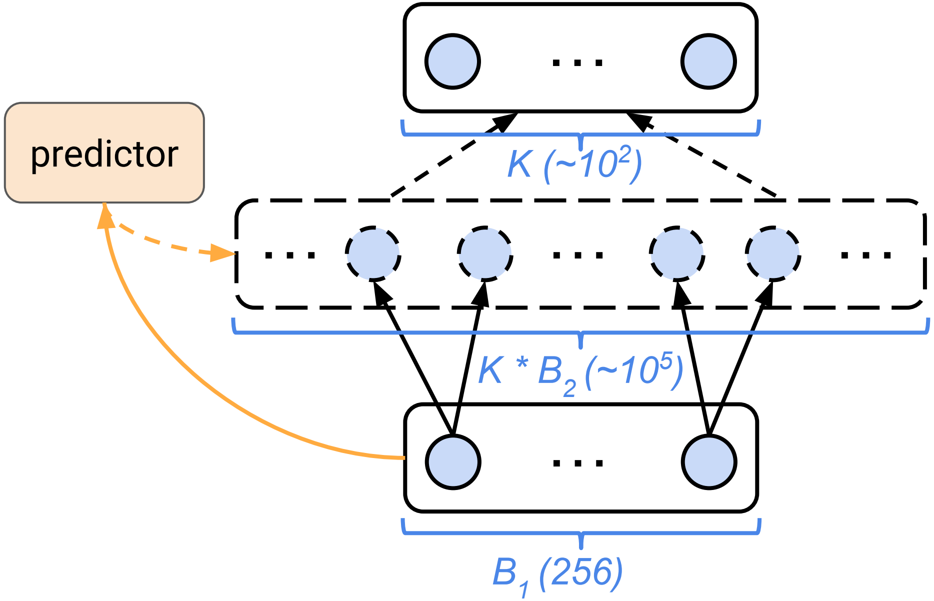
apply predictor to select top *K*

expand promising 2-block cells

train predictor

enumerate and train all 1-block cells

Progressive Neural Architecture Search



train the selected 2-block cells

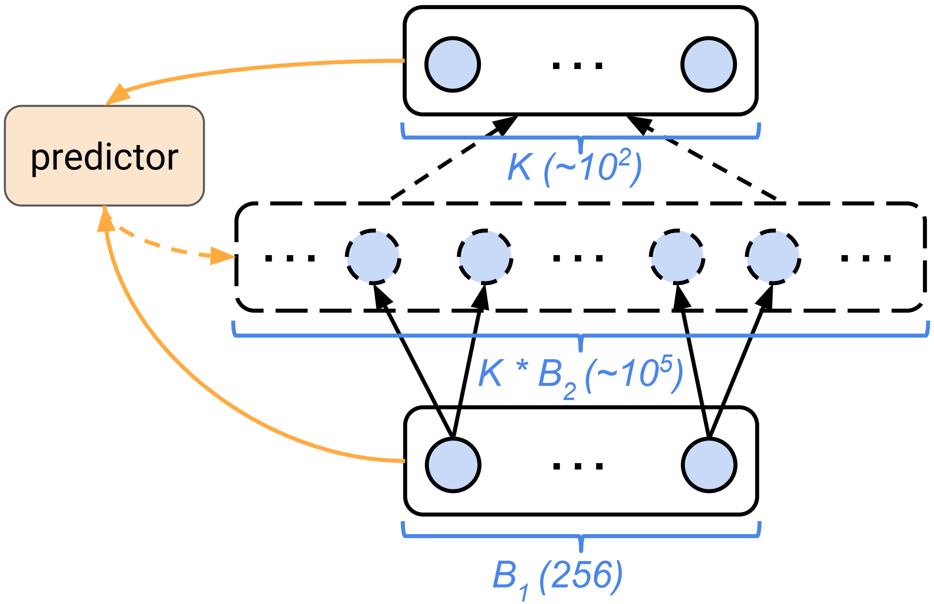
apply predictor to select top *K*

expand promising 2-block cells

train predictor

enumerate and train all 1-block cells

Progressive Neural Architecture Search



finetune predictor

train the selected 2-block cells

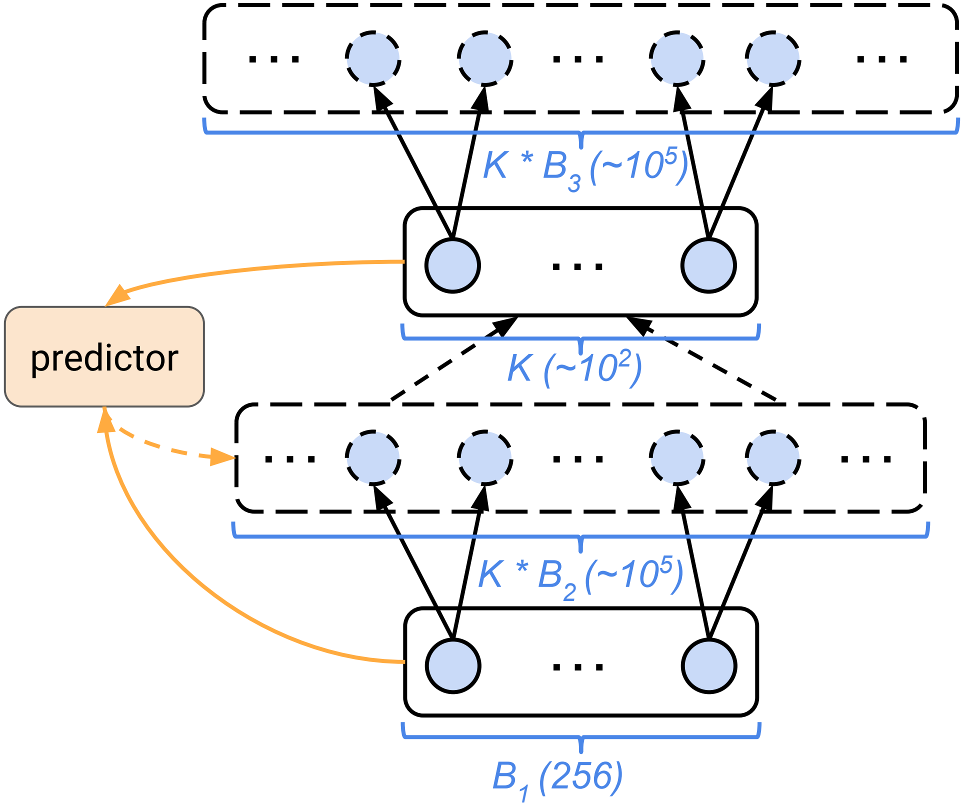
apply predictor to select top K

expand promising 2-block cells

train predictor

enumerate and train all 1-block cells

Progressive Neural Architecture Search



expand promising 3-block cells

finetune predictor

train the selected 2-block cells

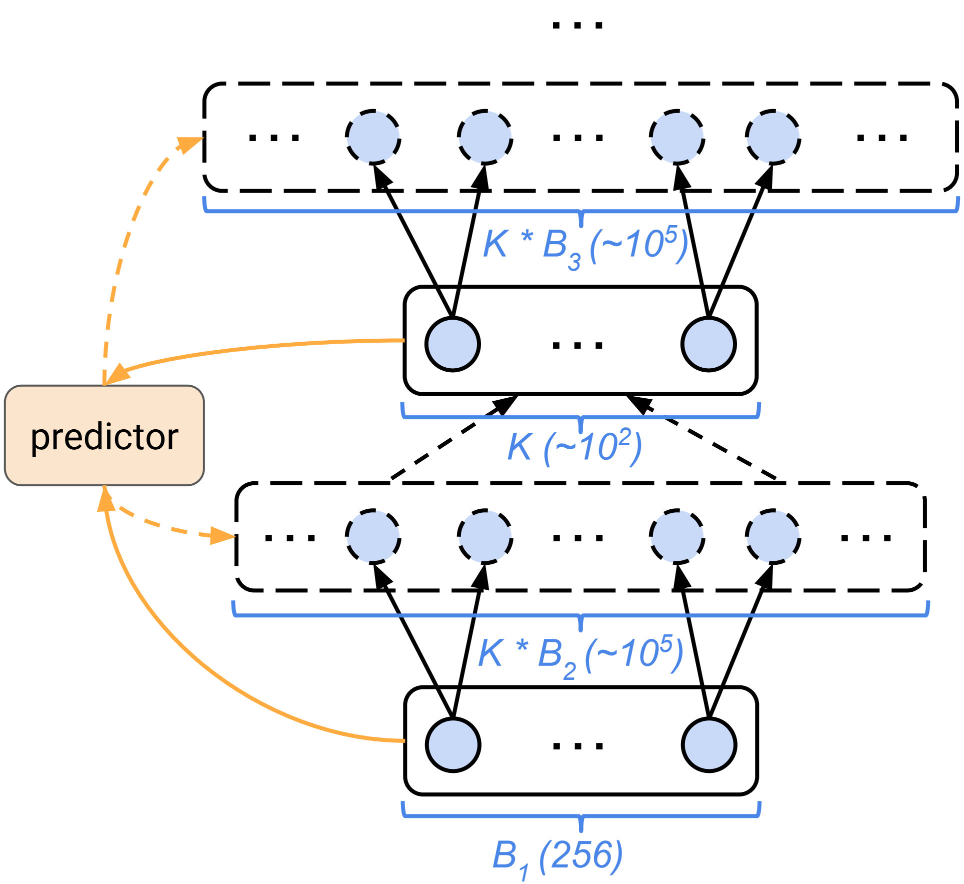
apply predictor to select top *K*

expand promising 2-block cells

train predictor

enumerate and train all 1-block cells

Progressive Neural Architecture Search



apply predictor to select top K

expand promising 3-block cells

finetune predictor

train the selected 2-block cells

apply predictor to select top K

expand promising 2-block cells

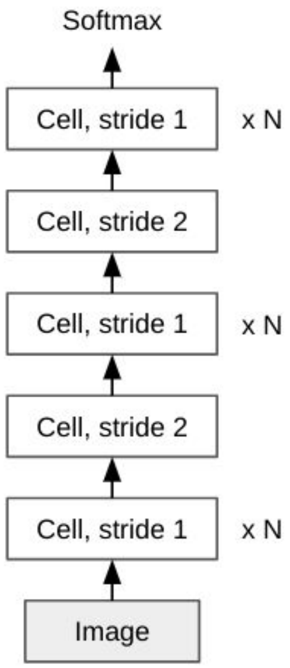
train predictor

enumerate and train all 1-block cells

Experiments and Results

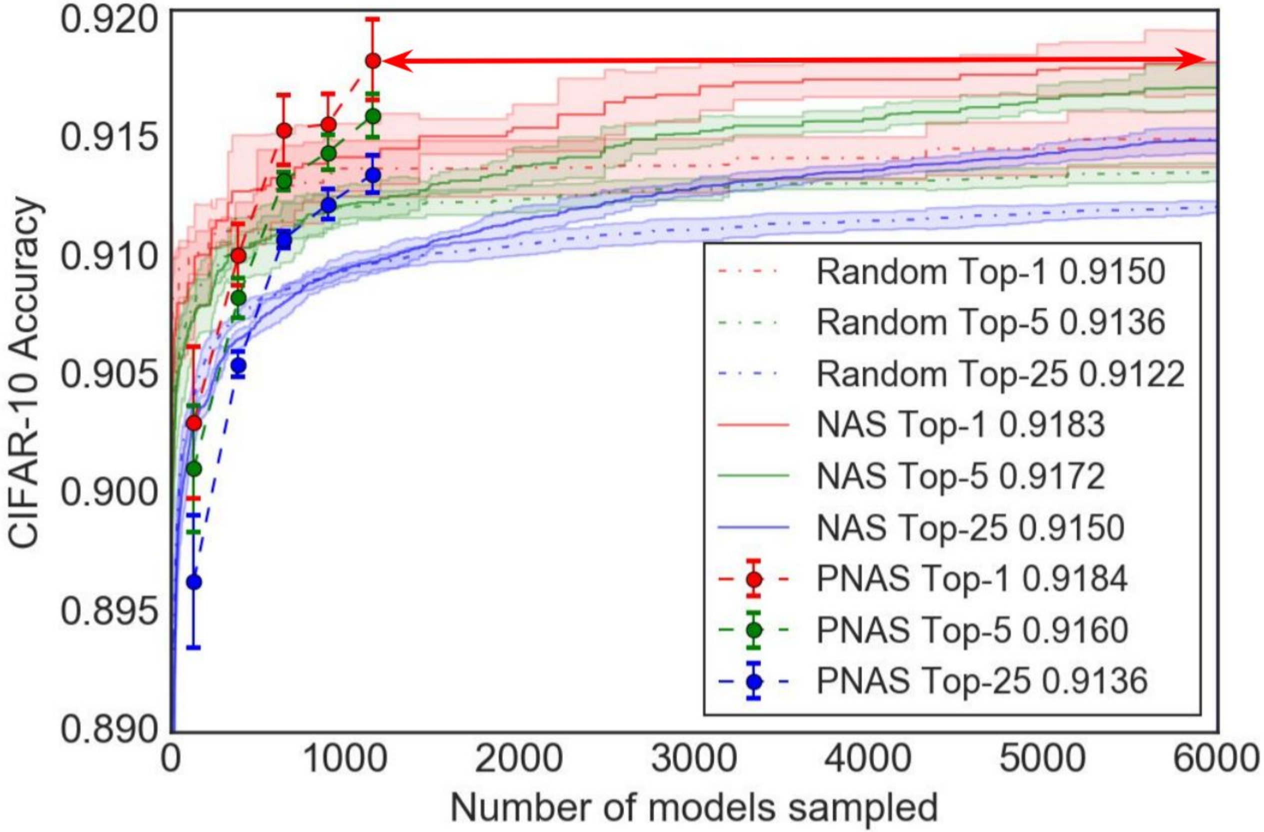
The Search Process

- We performed Progressive Neural Architecture Search (K = 256) on CIFAR-10
- Each model (N = 2, F = 24) was trained for 20 epochs with cosine learning rate
- **First big question: Is our search more efficient?**

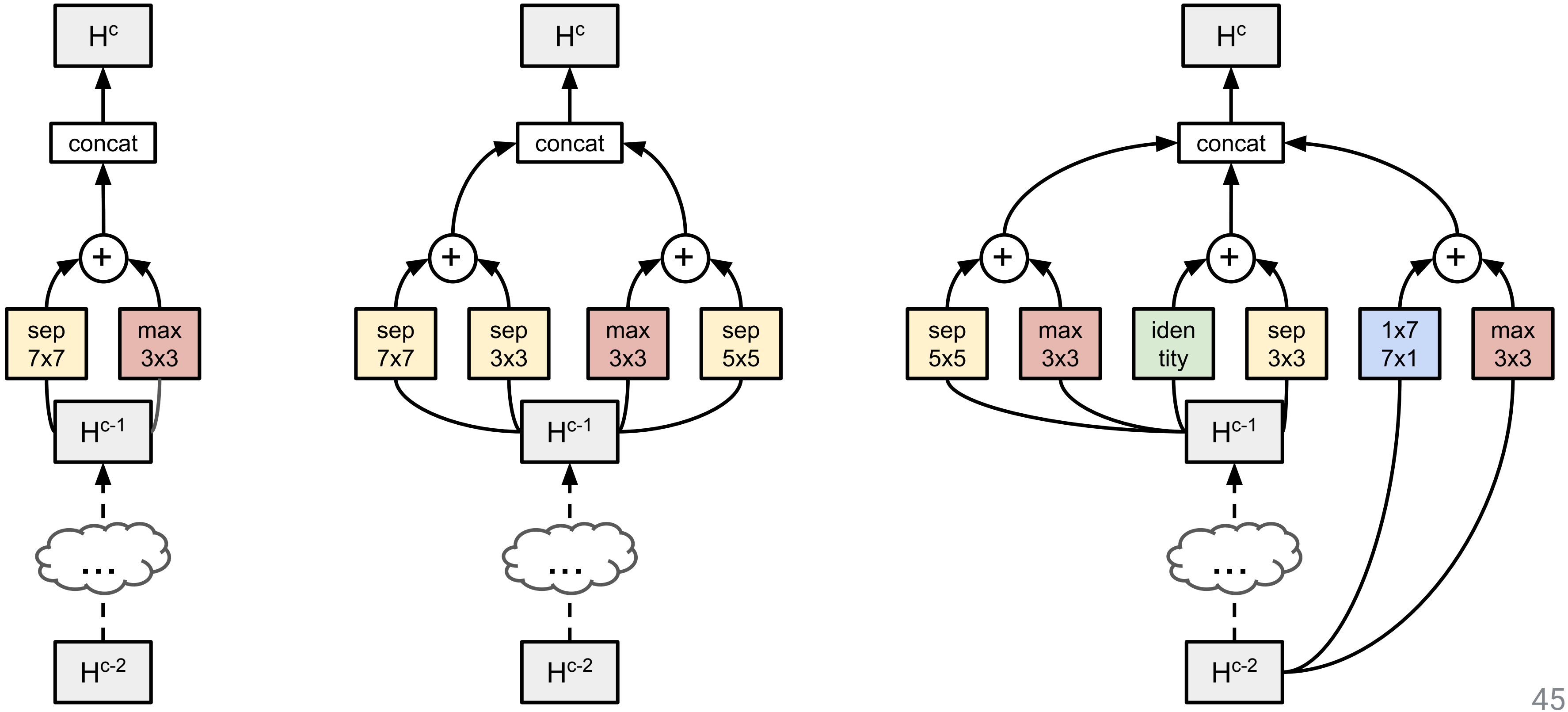


CIFAR-10
Architecture

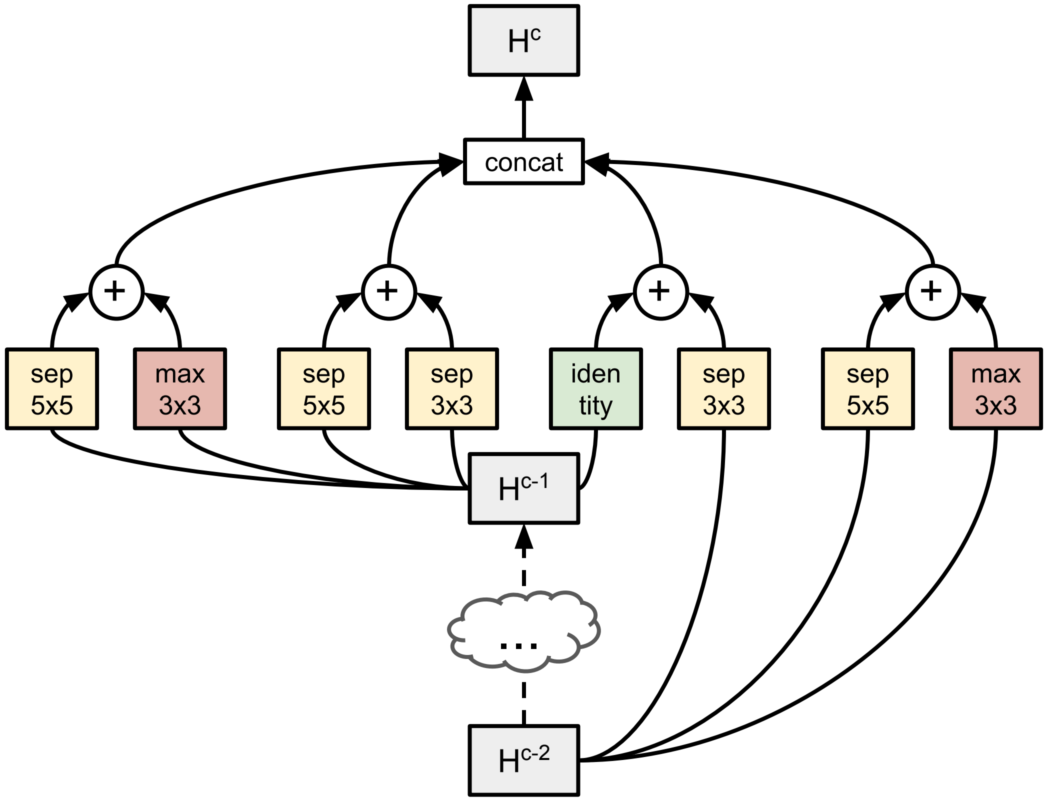
The Search Process: 5x Speedup



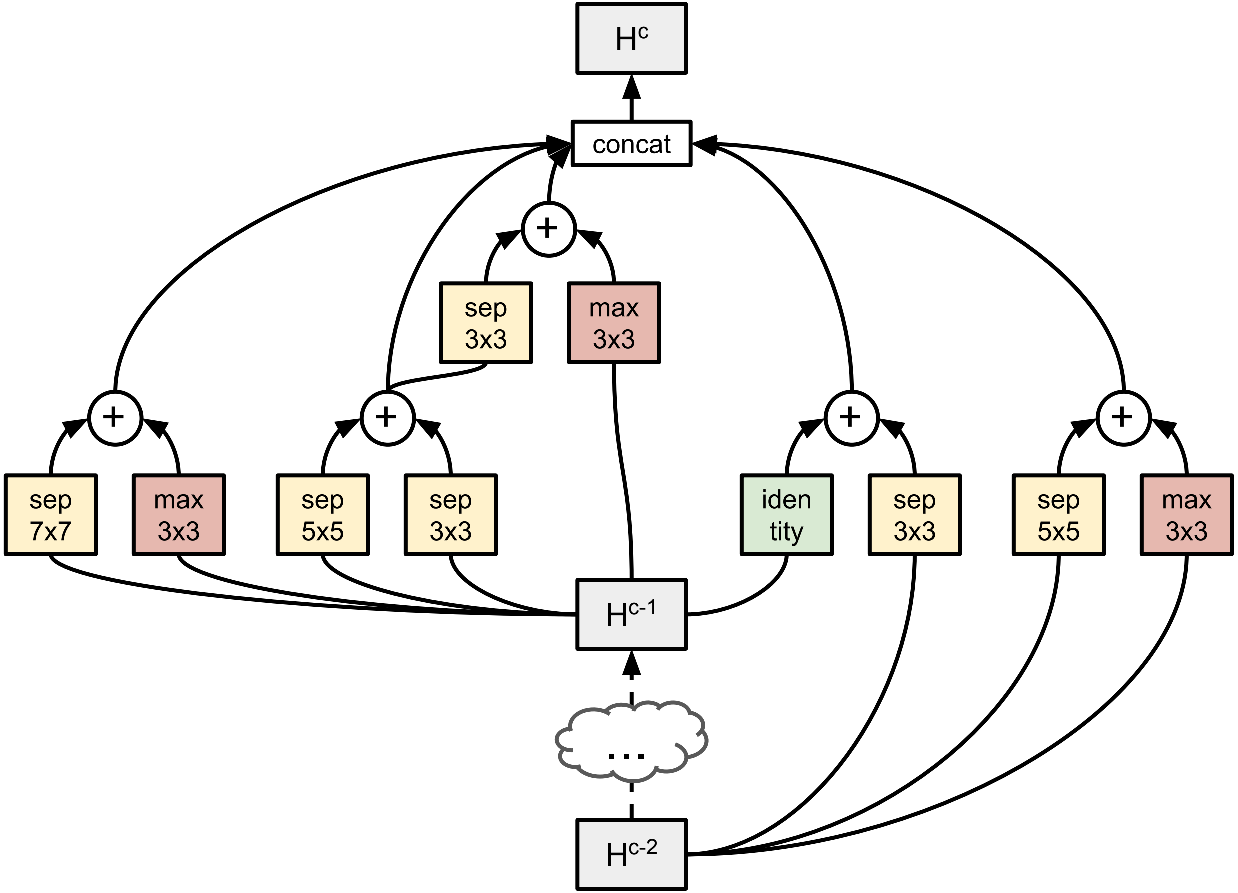
The Search Process: PNASNet-1, 2, 3



The Search Process: PNASNet-4



The Search Process: PNASNet-5



After The Search

- Select the best 5-block cell structure; increase *N* and *F*
- Train and evaluate on both CIFAR-10 and ImageNet
- **Second big question: How competitive is the found cell structure on benchmark datasets?**

After The Search: CIFAR-10

Model	# Params	Error Rate	Method	Search Cost
NASNet-A [1]	3.3M	3.41	RL	21.4 - 29.3B
NASNet-B [1]	2.6M	3.73	RL	21.4 - 29.3B
NASNet-C [1]	3.1M	3.59	RL	21.4 - 29.3B
Hier-EA [2]	15.7M	3.75 ± 0.12	EA	35.8B
AmoebaNet-B [3]	2.8M	3.37 ± 0.04	EA	63.5B
AmoebaNet-A [3]	3.2M	3.34 ± 0.06	EA	25.2B
PNASNet-5	3.2M	3.41 ± 0.09	SMBO	1.0B

[1] Zoph, Barret, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. "Learning transferable architectures for scalable image recognition." In CVPR. 2018.
[2] Liu, Hanxiao, et al. "Hierarchical representations for efficient architecture search." In ICLR. 2018.
[3] Real, Esteban, et al. "Regularized evolution for image classifier architecture search." arXiv preprint arXiv:1802.01548 (2018).

After The Search: ImageNet (Mobile)

Model	# Params	# Mult-Add	Top 1	Top 5
MobileNet [1]	4.2M	569M	70.6	89.5
ShuffleNet [2]	5M	524M	70.9	89.8
NASNet-A [3]	5.3M	564M	74.0	91.6
AmoebaNet-B [4]	5.3M	555M	74.0	91.5
AmoebaNet-A [4]	5.1M	555M	74.5	92.0
AmoebaNet-C [4]	6.4M	570M	75.7	92.4
PNASNet-5	5.1M	588M	74.2	91.9

[1] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).
[2] Zhang, ,Xiangyu, et al. "Shufflenet: An extremely efficient convolutional neural network for mobile devices." arXiv preprint arXiv:1707.01083 (2017).
[3] Zoph, Barret, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. "Learning transferable architectures for scalable image recognition." In CVPR. 2018.
[4] Real, Esteban, et al. "Regularized evolution for image classifier architecture search." arXiv preprint arXiv:1802.01548 (2018).

After The Search: ImageNet (Large)

Model	# Params	# Mult-Add	Top 1	Top 5
ResNeXt-101 [1]	83.6M	31.5B	80.9	95.6
Squeeze-Excite [2]	145.8M	42.3B	82.7	96.2
NASNet-A [3]	88.9M	23.8B	82.7	96.2
AmoebaNet-B [4]	84.0M	22.3B	82.3	96.1
AmoebaNet-A [4]	86.7M	23.1B	82.8	96.1
AmoebaNet-C [4]	155.3M	41.1B	83.1	96.3
PNASNet-5	86.1M	25.0B	82.9	96.2

[1] Xie, Saining, et al. "Aggregated residual transformations for deep neural networks." In CVPR. 2017.
[2] Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." In CVPR. 2018.
[3] Zoph, Barret, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. "Learning transferable architectures for scalable image recognition." In CVPR. 2018.
[4] Real, Esteban, et al. "Regularized evolution for image classifier architecture search." arXiv preprint arXiv:1802.01548 (2018).

Conclusion

- We propose to search neural network architectures in order of increasing complexity, while simultaneously learning a surrogate function to guide the search.

- PNASNet-5 achieves state-of-the-art level accuracies on CIFAR-10 and ImageNet, while being 5 to 8 times more efficient than leading RL and EA approaches during the search process.

Code and Model Release

- We have released PNASNet-5 trained on ImageNet
    - Both *Mobile* and *Large*
    - Both *TensorFlow* and *PyTorch*
    - **SOTA on ImageNet amongst all publicly available models**

https://github.com/tensorflow/models/tree/master/research/slim
https://github.com/chenxi116/PNASNet.TF
https://github.com/chenxi116/PNASNet.pytorch

Extensions

- Our PNAS algorithm has been applied on related tasks:
  - PPP-Net [1] and DPP-Net [2]: Pareto-optimal architectures
  - Auto-Meta [3]: Meta-learning
- PNAS did not address sharing among child models:
  - ENAS [4] and DARTS [5] showed its importance to speedup
  - EPNAS [6] combined ENAS and PNAS for further speedup

[1] Dong, Jin-Dong, et al. "PPP-Net: Platform-aware Progressive Search for Pareto-optimal Neural Architectures." ICLR 2018 Workshop.
[2] Dong, Jin-Dong, et al. "DPP-Net: Device-aware Progressive Search for Pareto-optimal Neural Architectures." ECCV 2018.
[3] Kim, Jaehong, et al. "Auto-Meta: Automated Gradient Based Meta Learner Search." arXiv preprint arXiv:1806.06927 (2018).
[4] Pham, Hieu, et al. "Efficient Neural Architecture Search via Parameter Sharing." ICML 2018.
[5] Liu, Hanxiao, Karen Simonyan, and Yiming Yang. "DARTS: Differentiable Architecture Search." arXiv preprint arXiv:1806.09055 (2018).
[6] Perez-Rua, Juan-Manuel, Moez Baccouche, and Stephane Pateux. "Efficient Progressive Neural Architecture Search." BMVC 2018.

Thank You

Poster session 3B (Wednesday, September 12, 2:30pm - 4:00pm)

@chenxi116 https://cs.jhu.edu/~cxliu/