NAS

Neural Architecture Search

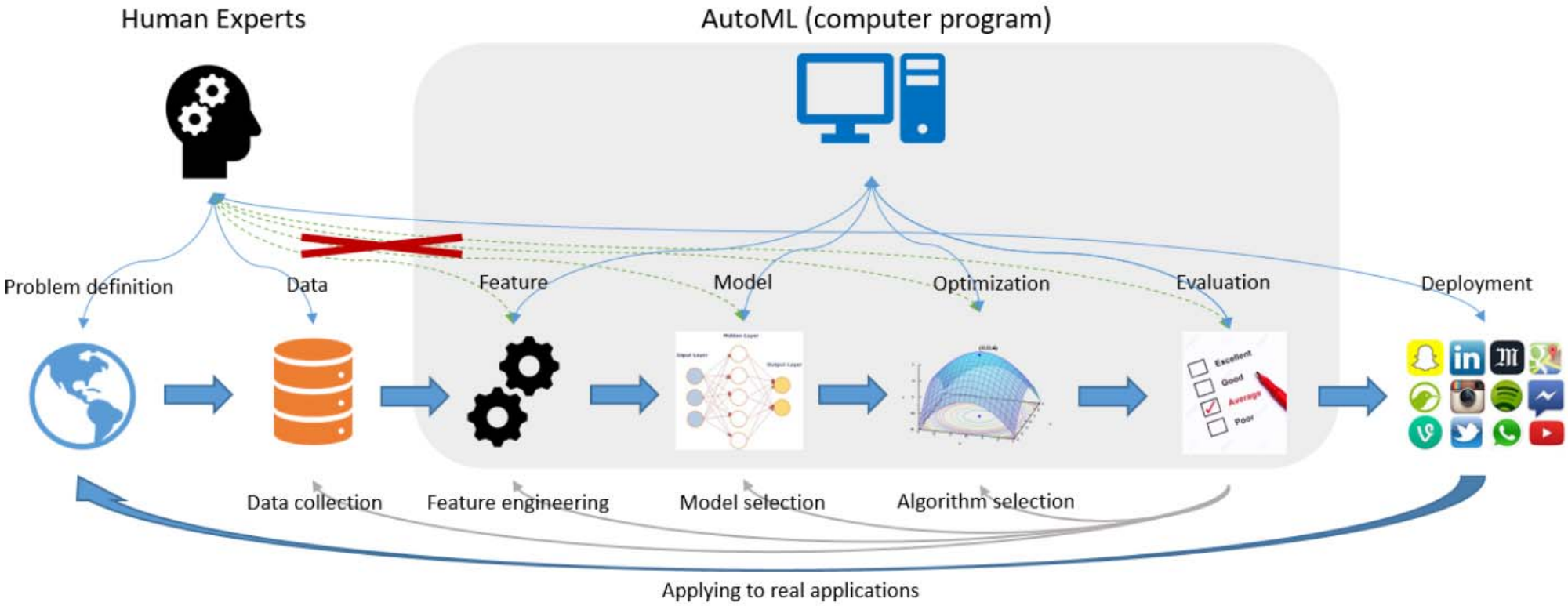
Introduction

- NAS是AutoML的一个分支

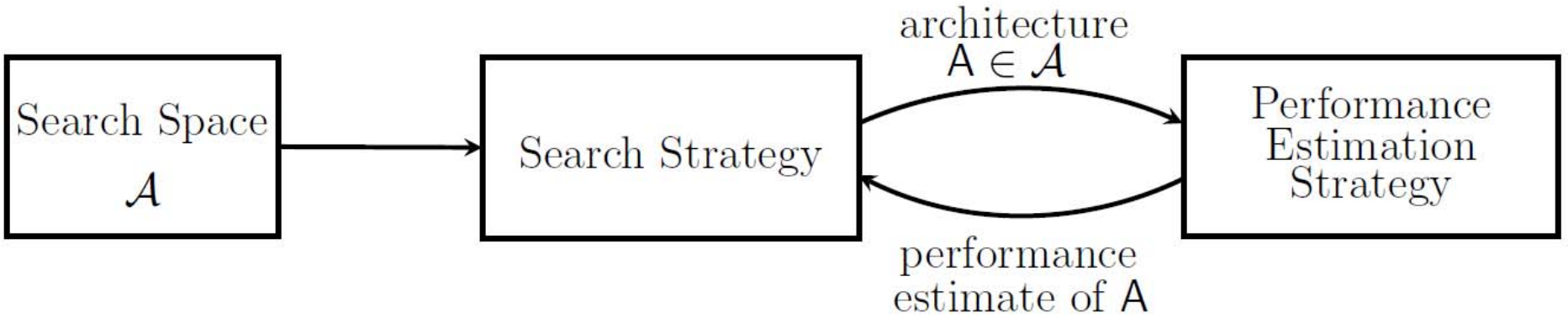
- AutoML是自动化和人工智能领域的交叉学科。

- Hyper-parameter Optimization
- Meta Learning
- Neural Architecture Search

AutoML


Human Experts
AutoML (computer program)
Problem definition
Data
Feature
Model
Optimization
Evaluation
Deployment
Data collection
Feature engineering
Model selection
Algorithm selection
Applying to real applications

NAS



- Search Strategy
- Search Space
- Approximation

Search Space

- Plane
- Cell-based
- Module based

Search Strategy

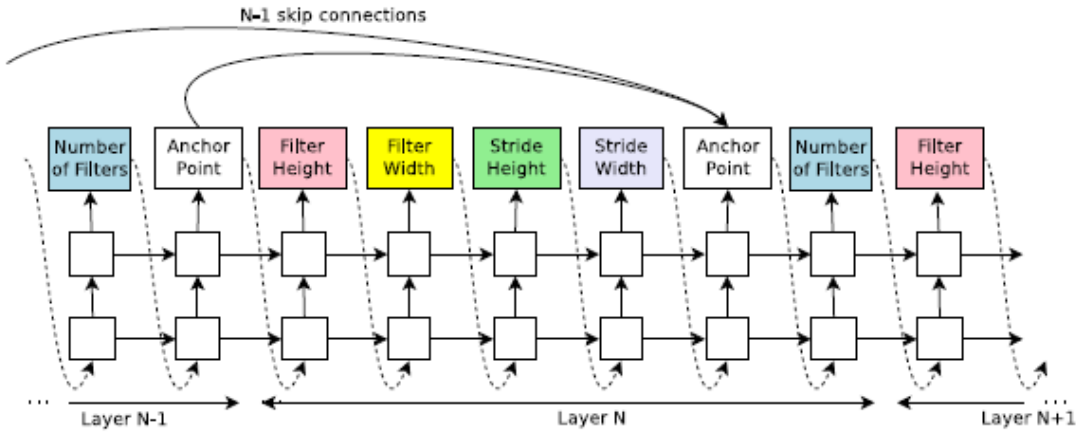
- Grid Search
- Random Search
- Bayesian Optimization
- Reinforcement Learning
- Evolution
- Continues Space

Approximation

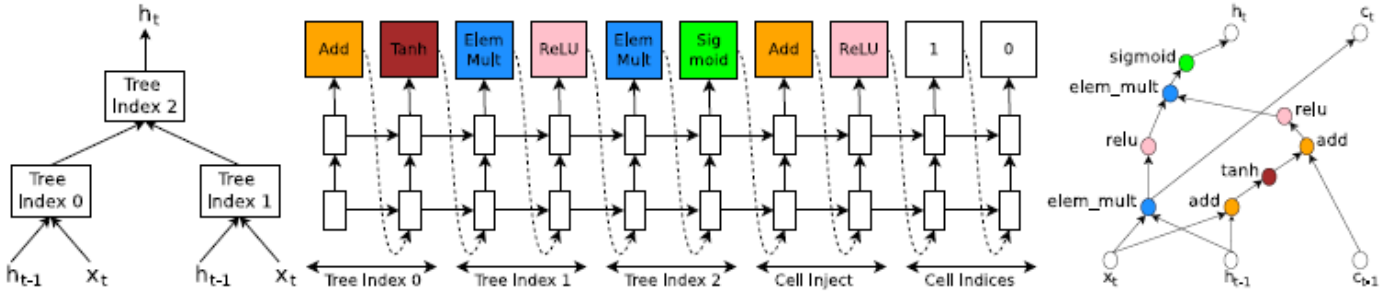
- Down-Sample
- Early Stop
- Reuse weight
- One shot graph
- 序列模型外推

NAS

- 同时还有预测learning rate、Pooling、Batchnorm等，没有明确说具体做法。



- 图为2base，在实验过程中用的是8base



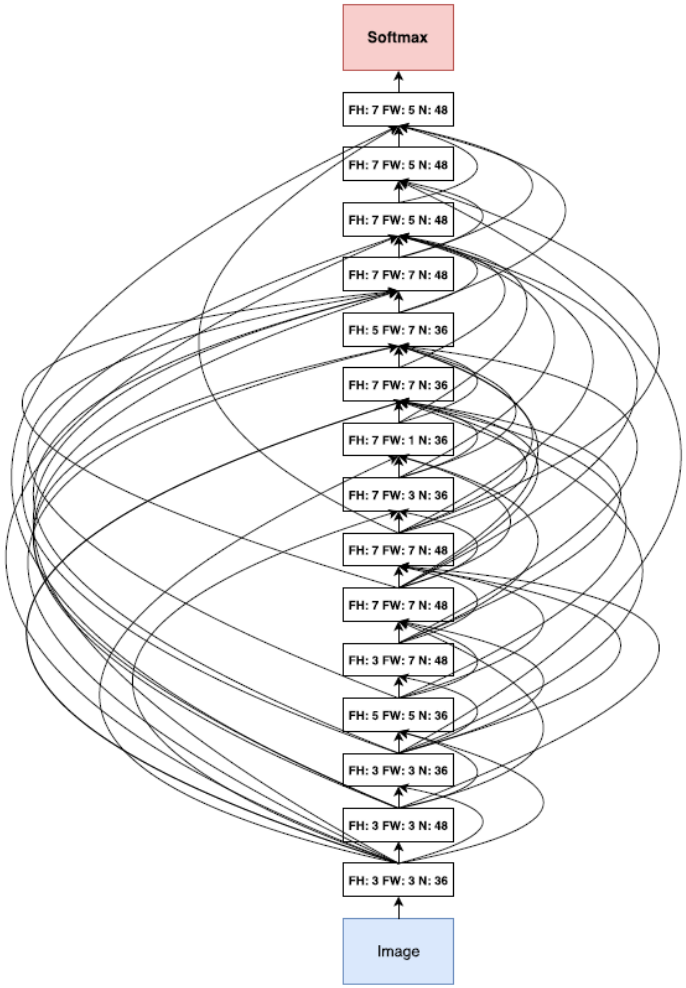
NAS

- CIFAR-10
- H,W [1,3,5,7] Num[24,36,48,64] Stride[1,2,3]

- 22400GPU-hours, P100 * 800

DenseNet ($L = 40, k = 12$) Huang et al. (2016a)	40	1.0M	5.24
DenseNet($L = 100, k = 12$) Huang et al. (2016a)	100	7.0M	4.10
DenseNet ($L = 100, k = 24$) Huang et al. (2016a)	100	27.2M	3.74
DenseNet-BC ($L = 100, k = 40$) Huang et al. (2016b)	190	25.6M	3.46
Neural Architecture Search v1 no stride or pooling	15	4.2M	5.50
Neural Architecture Search v2 predicting strides	20	2.5M	6.01
Neural Architecture Search v3 max pooling	39	7.1M	4.47
Neural Architecture Search v3 max pooling + more filters	39	37.4M	3.65

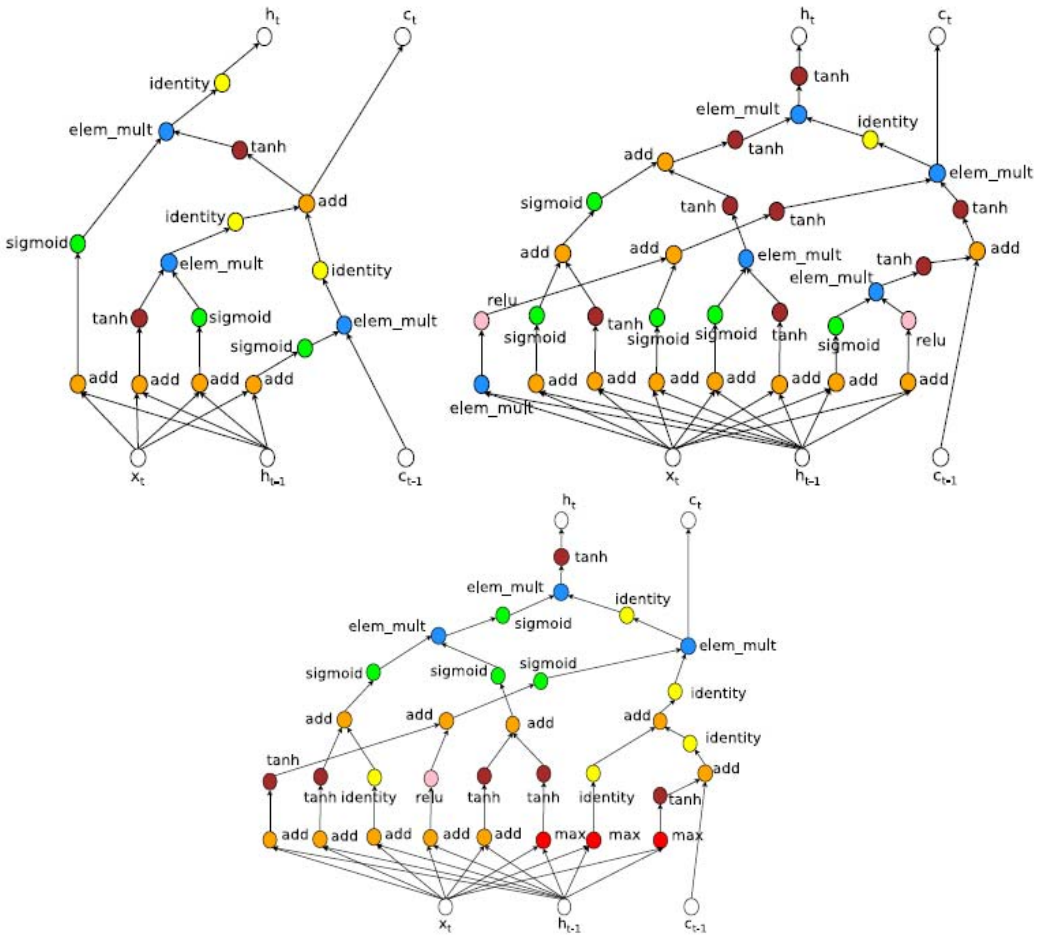
Table 1: Performance of Neural Architecture Search and other state-of-the-art models on CIFAR-10.



NAS

- PTB

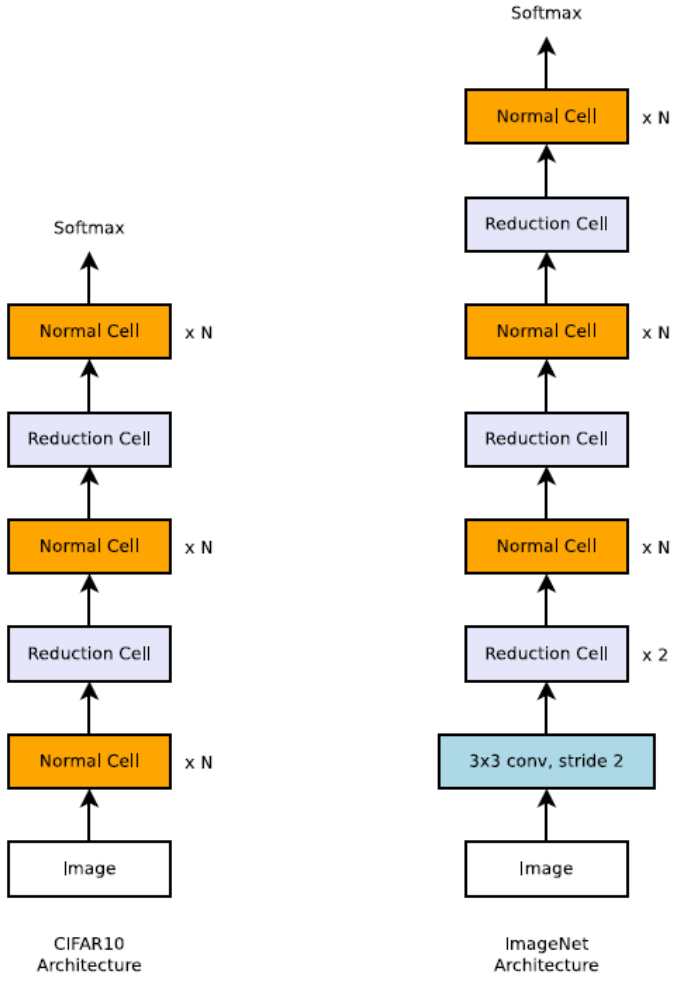
Model	Parameters	Test Perplexity
Mikolov & Zweig (2012) - KN-5	2M‡	141.2
Mikolov & Zweig (2012) - KN5 + cache	2M‡	125.7
Mikolov & Zweig (2012) - RNN	6M‡	124.7
Mikolov & Zweig (2012) - RNN-LDA	7M‡	113.7
Mikolov & Zweig (2012) - RNN-LDA + KN-5 + cache	9M‡	92.0
Pascanu et al. (2013) - Deep RNN	6M	107.5
Cheng et al. (2014) - Sum-Prod Net	5M‡	100.0
Zaremba et al. (2014) - LSTM (medium)	20M	82.7
Zaremba et al. (2014) - LSTM (large)	66M	78.4
Gal (2015) - Variational LSTM (medium, untied)	20M	79.7
Gal (2015) - Variational LSTM (medium, untied, MC)	20M	78.6
Gal (2015) - Variational LSTM (large, untied)	66M	75.2
Gal (2015) - Variational LSTM (large, untied, MC)	66M	73.4
Kim et al. (2015) - CharCNN	19M	78.9
Press & Wolf (2016) - Variational LSTM, shared embeddings	51M	73.2
Merity et al. (2016) - Zoneout + Variational LSTM (medium)	20M	80.6
Merity et al. (2016) - Pointer Sentinel-LSTM (medium)	21M	70.9
Inan et al. (2016) - VD-LSTM + REAL (large)	51M	68.5
Zilly et al. (2016) - Variational RHN, shared embeddings	24M	66.0
Neural Architecture Search with base 8	32M	67.9
Neural Architecture Search with base 8 and shared embeddings	25M	64.0
Neural Architecture Search with base 8 and shared embeddings	54M	62.4



Cell-based NAS

- Normal Cell
- Reduction Cell

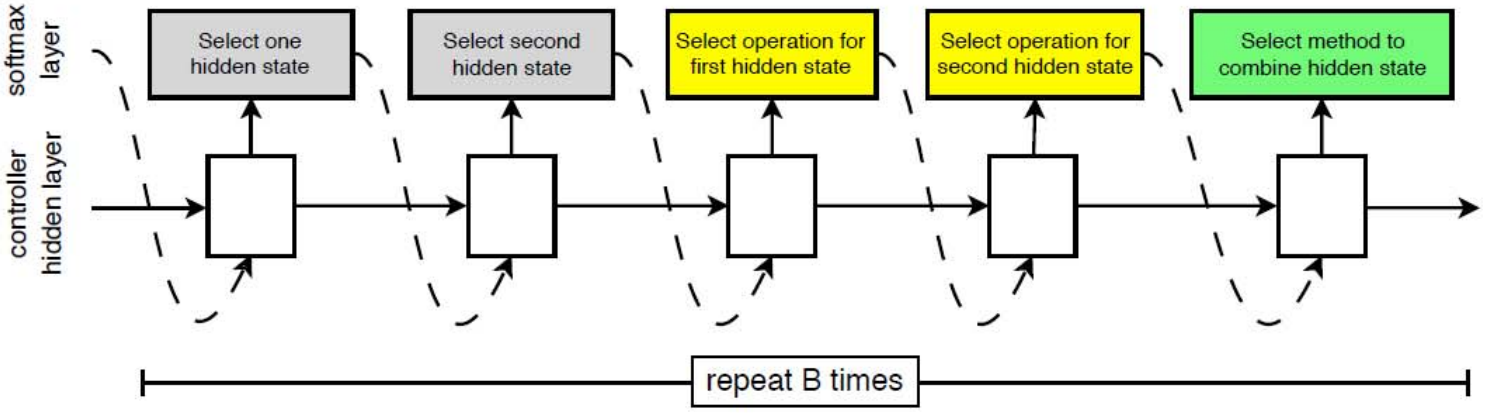
- 不同的任务之间输入的尺寸不同，来自其他模型的启发

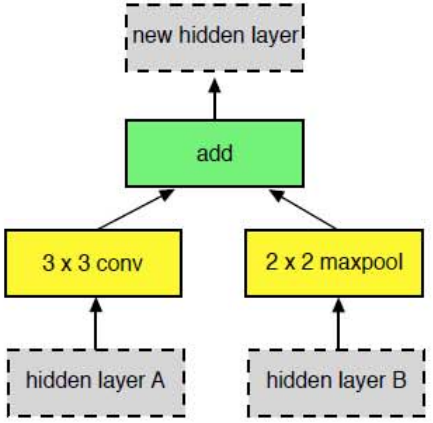


Cell-based NAS

- 搜索空间:
- 搜索策略
- Proximal Policy Optimization

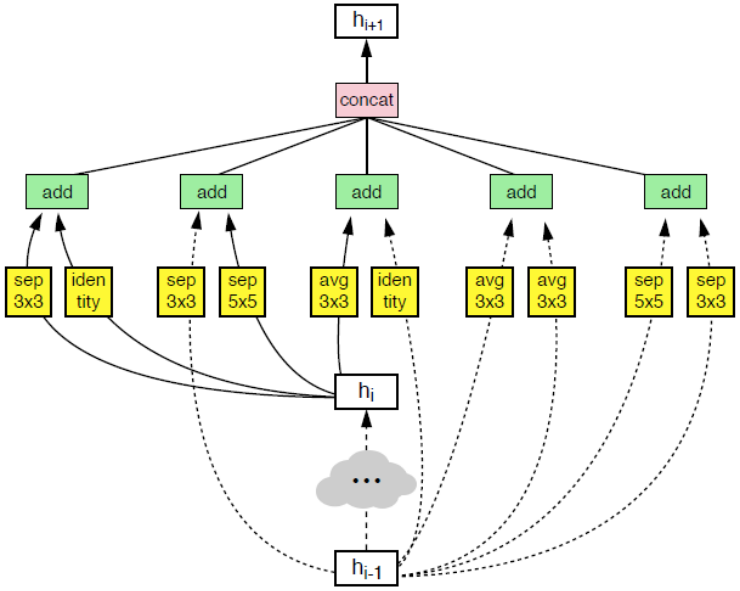
- identity
- 1x7 then 7x1 convolution
- 3x3 average pooling
- 5x5 max pooling
- 1x1 convolution
- 3x3 depthwise-separable conv
- 7x7 depthwise-separable conv
- 1x3 then 3x1 convolution
- 3x3 dilated convolution
- 3x3 max pooling
- 7x7 max pooling
- 3x3 convolution
- 5x5 depthwise-seperable conv



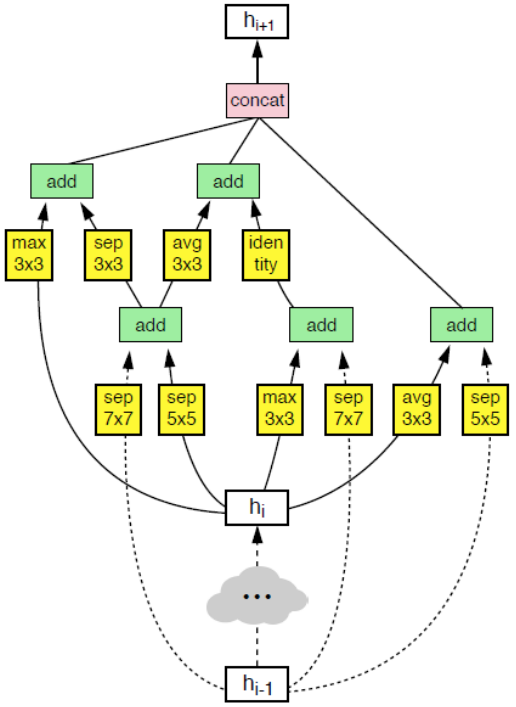


Cell-based NAS

- 2000GPU-hours, K40 * 500
- 在CIFAR10上训练得到
- 在CIFAR10上5次运行平均性能达到2.4%，最好的一次达到2.19%



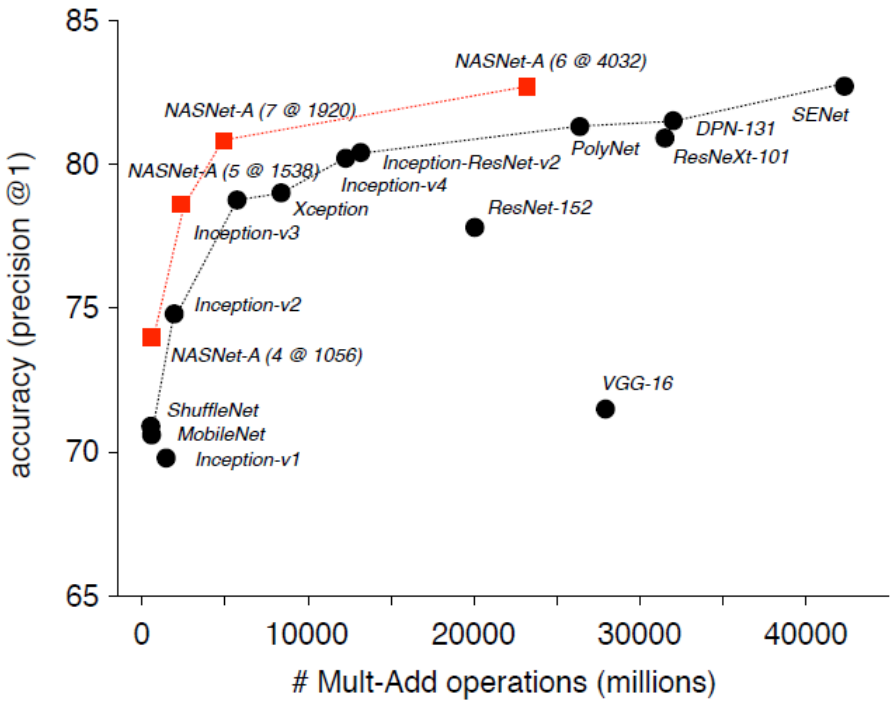
Normal Cell

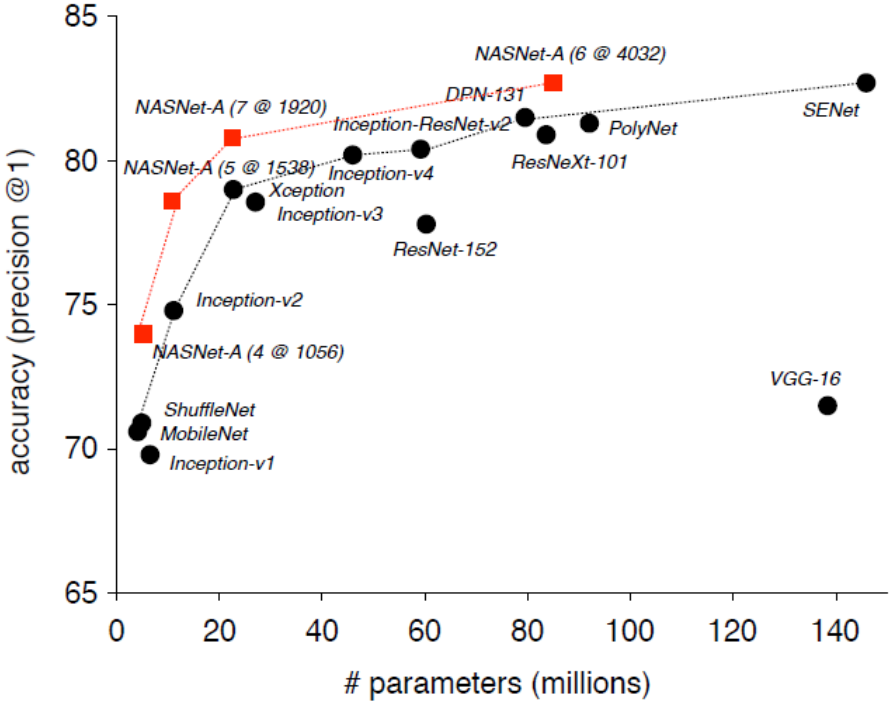


Reduction Cell

Cell-based NAS

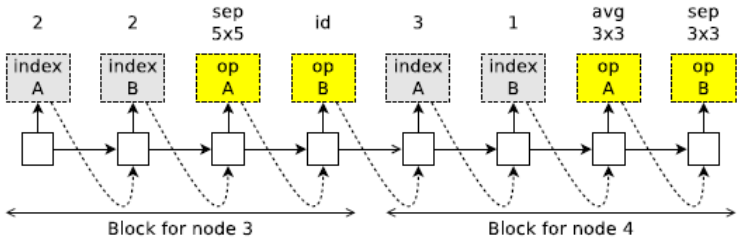
- ILSVRC2012

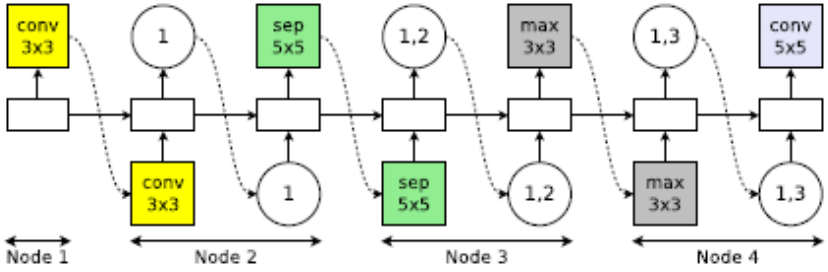


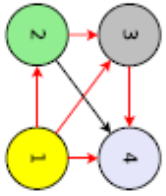


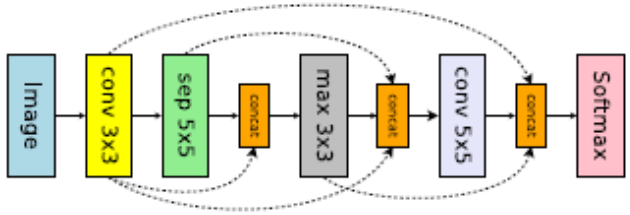
ENAS

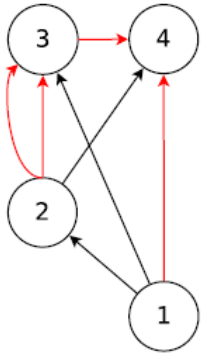
- DAG
- 搜索Block
- 搜索边并在过程中复用边的权值

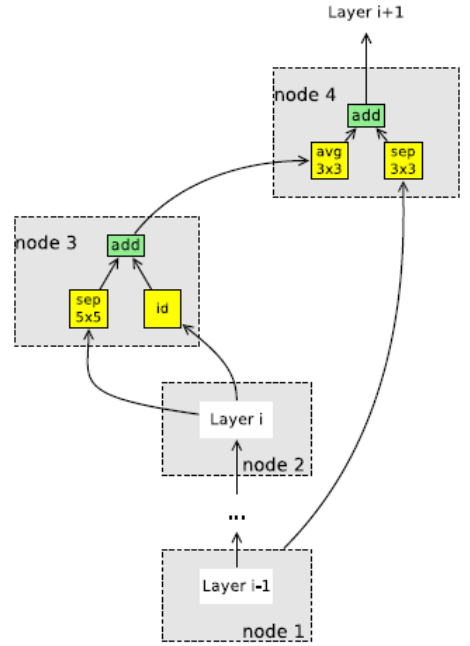






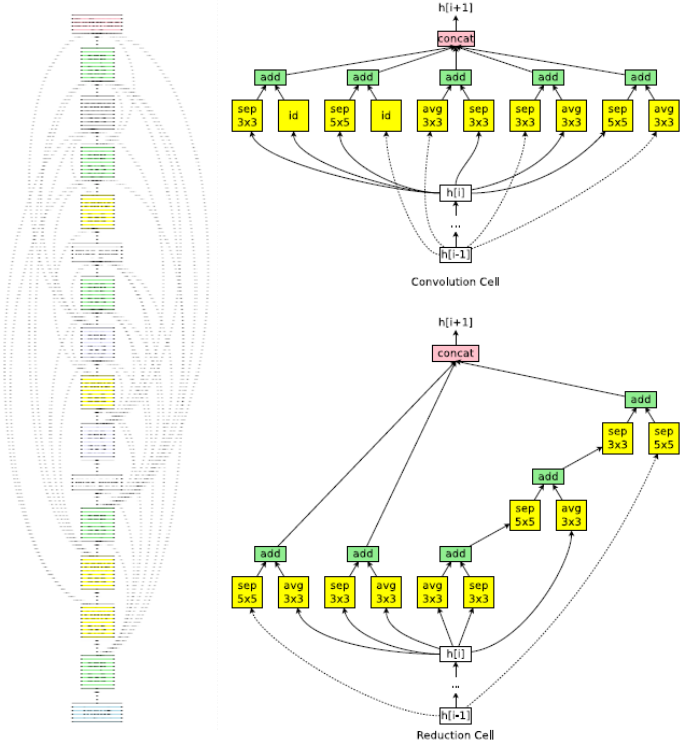






ENAS

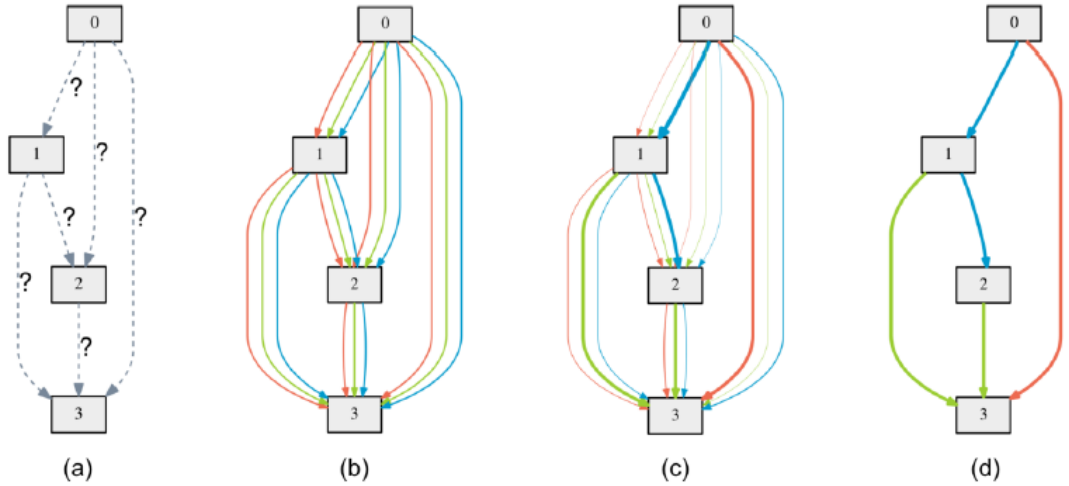
- 10 GPU-hours, GTX 1080Ti * 1



Method	GPUs	Times (days)	Params (million)	Error (%)
DenseNet-BC (Huang et al., 2016)	–	–	25.6	3.46
DenseNet + Shake-Shake (Gastaldi, 2016)	–	–	26.2	2.86
DenseNet + CutOut (DeVries & Taylor, 2017)	–	–	26.2	**2.56**
Budgeted Super Nets (Veniat & Denoyer, 2017)	–	–	–	9.21
ConvFabrics (Saxena & Verbeek, 2016)	–	–	21.2	7.43
Macro NAS + Q-Learning (Baker et al., 2017a)	10	8-10	11.2	6.92
Net Transformation (Cai et al., 2018)	5	2	19.7	5.70
FractalNet (Larsson et al., 2017)	–	–	38.6	4.60
SMASH (Brock et al., 2018)	1	1.5	16.0	4.03
NAS (Zoph & Le, 2017)	800	21-28	7.1	4.47
NAS + more filters (Zoph & Le, 2017)	800	21-28	37.4	**3.65**
ENAS + macro search space	1	0.32	21.3	4.23
ENAS + macro search space + more channels	1	0.32	38.0	**3.87**
Hierarchical NAS (Liu et al., 2018)	200	1.5	61.3	3.63
Micro NAS + Q-Learning (Zhong et al., 2018)	32	3	–	3.60
Progressive NAS (Liu et al., 2017)	100	1.5	3.2	3.63
NASNet-A (Zoph et al., 2018)	450	3-4	3.3	3.41
NASNet-A + CutOut (Zoph et al., 2018)	450	3-4	3.3	**2.65**
ENAS + micro search space	1	0.45	4.6	3.54
ENAS + micro search space + CutOut	1	0.45	4.6	**2.89**

DARTS

- 直接给边赋权而不采用搜索
- 将边设置为soft-max的加权和
- 多次迭代后砍去权重低的边
保留权重高的边



DARTS

- CIFR10
- PTB

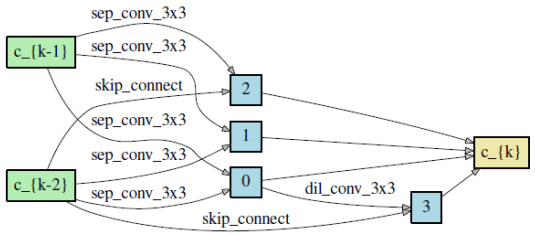
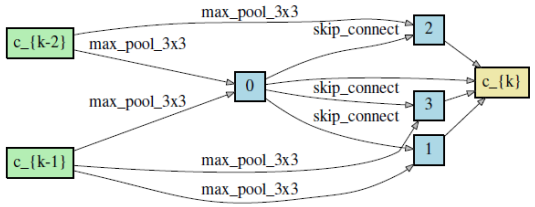
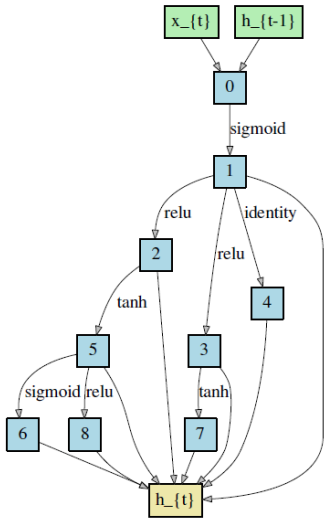


Figure 4: Normal cell learned on CIFAR-10.





Architecture	Test Error (%)	Params (M)	Search Cost (GPU days)	#ops	Search Method
DenseNet-BC (Huang et al., 2017)	3.46	25.6	–	–	manual
NASNet-A + cutout (Zoph et al., 2018)	2.65	3.3	2000	13	RL
NASNet-A + cutout (Zoph et al., 2018)†	2.83	3.1	2000	13	RL
BlockQNN (Zhong et al., 2018)	3.54	39.8	96	8	RL
AmoebaNet-A (Real et al., 2018)	3.34 ± 0.06	3.2	3150	19	evolution
AmoebaNet-A + cutout (Real et al., 2018)†	3.12	3.1	3150	19	evolution
AmoebaNet-B + cutout (Real et al., 2018)	2.55 ± 0.05	2.8	3150	19	evolution
Hierarchical evolution (Liu et al., 2018b)	3.75 ± 0.12	15.7	300	6	evolution
PNAS (Liu et al., 2018a)	3.41 ± 0.09	3.2	225	8	SMBO
ENAS + cutout (Pham et al., 2018b)	2.89	4.6	0.5	6	RL
ENAS + cutout (Pham et al., 2018b)*	2.91	4.2	4	6	RL
Random search baseline‡ + cutout	3.29 ± 0.15	3.2	4	7	random
DARTS (first order) + cutout	3.00 ± 0.14	3.3	1.5	7	gradient-based
DARTS (second order) + cutout	2.76 ± 0.09	3.3	4	7	gradient-based

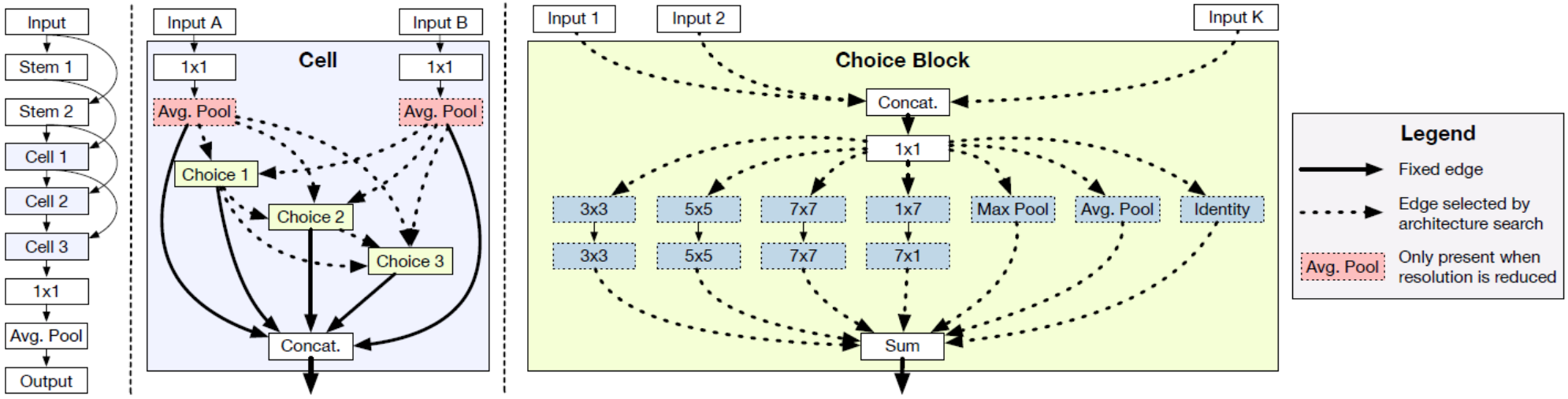
Architecture	Perplexity valid	Perplexity test	Params (M)	Search Cost (GPU days)	#ops	Search Method
Variational RHN (Zilly et al., 2016)	67.9	65.4	23	–	–	manual
LSTM (Merity et al., 2018)	60.7	58.8	24	–	–	manual
LSTM + skip connections (Melis et al., 2018)	60.9	58.3	24	–	–	manual
LSTM + 15 softmax experts (Yang et al., 2018)	58.1	56.0	22	–	–	manual
NAS (Zoph & Le, 2017)	–	64.0	25	1e4 CPU days	4	RL
ENAS (Pham et al., 2018b)*	68.3	63.1	24	0.5	4	RL
ENAS (Pham et al., 2018b)†	60.8	58.6	24	0.5	4	RL
Random search baseline‡	61.8	59.4	23	2	4	random
DARTS (first order)	60.2	57.6	23	0.5	4	gradient-based
DARTS (second order)	58.1	55.7	23	1	4	gradient-based

One-Shot

- 前面还有一篇NIPS2017的SMASH（不是来自谷歌的）：
- 使用一个HyperNet根据网络架构自动生成网络内部参数。
- 训练的最后采样一批网络结构，生成权重并排序，将最好的从零开始训练得到最终结果。

One-Shot

- 暴力为每个可能性都建边，在评估阶段时只取用其中一条边。



One-Shot

- 设计一个能够表示足够大的搜索空间的One-shot模型
- One-shot模型要足够精巧以便于在有限的资源上训练
- 整个模型中的卷积层只有可分离卷积和非对称卷积
- 非线性激活层全部使用BN-ReLU-Conv

- 防止直接训练导致的模块耦合
- 使用了线性递增的path dropout

One-Shot

- 保证训练模型的稳定性
- BN-ReLU-Conv，Ghost batch

- 防止过度正则化
- 训练期间，L2只应用于未被drop out的层

One-Shot

- 保证训练模型的稳定性
- BN-ReLU-Conv，Ghost batch

- 防止过度正则化
- 训练期间，L2只应用于未被drop out的层

One-Shot

- 在ImageNet上并没有和其他模型比较。
- 简化了SMASH。

Method	Param $\times 10^6$	Accuracy
ENAS General	34.9	95.8
ENAS masks	12.6	95.7
ENAS skip	14.1	95.0
ENAS skip large	38.0	96.1
ENAS Cell search	4.6	96.5
NASNet-A	3.3	96.6
SMASHv2	16.0	96.0
One-Shot Top ($F = 16$)	0.7 ± 0.1	94.6 ± 0.2
One-Shot Top ($F = 32$)	2.7 ± 0.3	95.5 ± 0.1
One-Shot Top ($F = 64$)	10.4 ± 1.0	95.9 ± 0.2
One-Shot Top ($F = 128$)	41.3 ± 4.0	96.1 ± 0.2
One-Shot Small ($F = 16$)	0.4 ± 0.01	94.6 ± 0.2
One-Shot Small ($F = 32$)	1.3 ± 0.04	95.6 ± 0.2
One-Shot Small ($F = 64$)	5.0 ± 0.2	96.0 ± 0.1
One-Shot Small ($F = 128$)	19.3 ± 0.6	96.1 ± 0.2
All On ($F = 16$)	1.3	95.0
All On ($F = 32$)	4.8	95.6
All On ($F = 64$)	18.5	96.0
All On ($F = 128$)	72.7	96.2
Random ($F = 16$)	0.5 ± 0.2	94.1 ± 0.5
Random ($F = 32$)	1.7 ± 0.7	95.0 ± 0.5
Random ($F = 64$)	6.7 ± 2.6	95.6 ± 0.2
Random ($F = 128$)	26.4 ± 10.5	95.8 ± 0.2

Single Path One-Shot

- 问题:
- SuperNet内部的权重深耦合，目前不知道为什么从中采样并继承权重有效。
- 结构参数和SuperNet的权重深耦合

- 单路径采样训练
- 使用演化方法搜索
- 增加了关于Channel数与Quantization程度的搜索

Single Path One-Shot

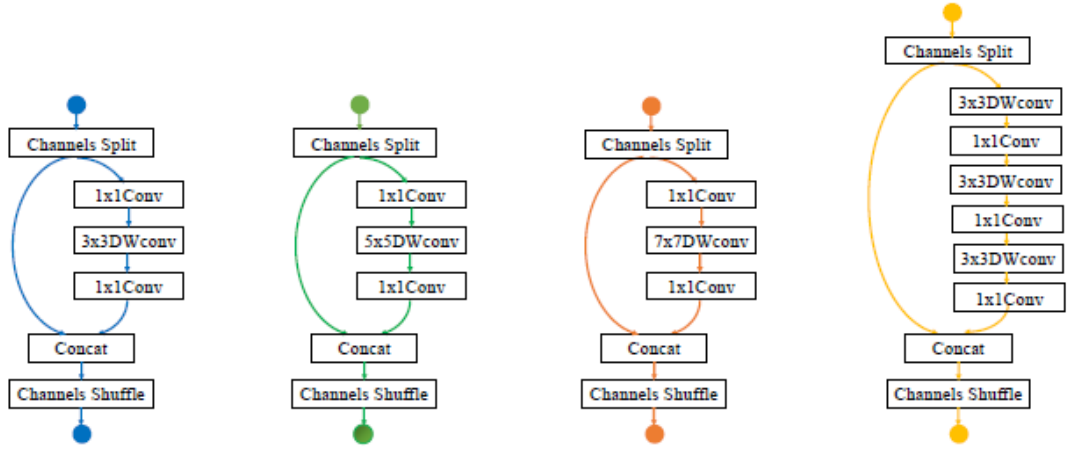
- The batch size is 1024.
- Supernet is trained for 120 epochs (150000 iterations)
- the best architecture for 240 epochs (300000 iterations).
- 1080Ti * 8

Single Path One-Shot

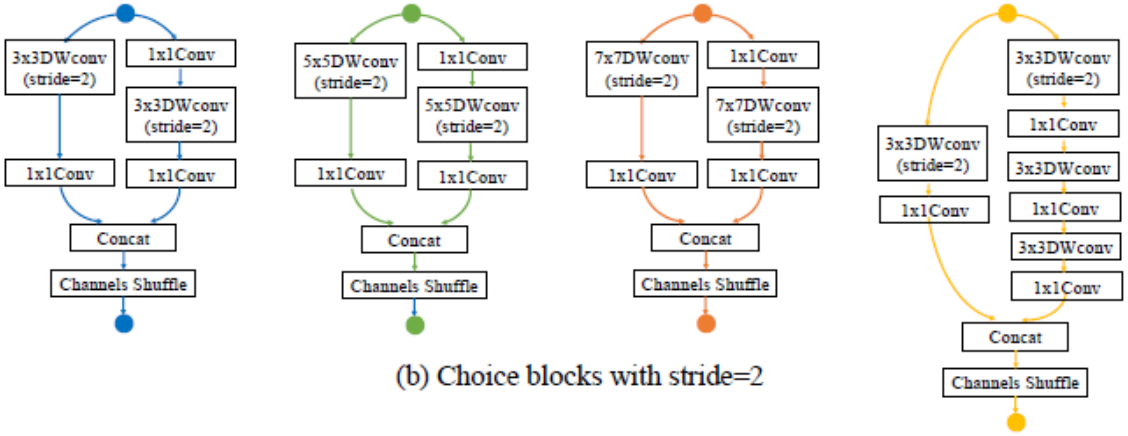
- Imagenet

- 使用FLOPs约束模型

input shape	block	channels	repeat	stride
$224^2 \times 3$	3×3 conv	16	1	2
$112^2 \times 16$	CB	64	4	2
$56^2 \times 64$	CB	160	4	2
$28^2 \times 160$	CB	320	8	2
$14^2 \times 320$	CB	640	4	2
$7^2 \times 640$	1×1 conv	1024	1	1
$7^2 \times 1024$	GAP	-	1	-
1024	fc	1000	1	-



(a) Choice blocks with stride=1



(b) Choice blocks with stride=2

Single Path One-Shot

Method	Proxyless	FBNet	Ours
GPU memory cost (8 GPUs in total)	37G	63G	24G
Training time	15 Gds	20 Gds	12 Gds
Search time	0	0	<1 Gds
Retrain time	16 Gds	16 Gds	16 Gds
Total time	31 Gds	36 Gds	29 Gds

Model	FLOPs	Top-1 acc(%)
all choice_3	324M	73.4
rand sel. channels (5 times)	~ 323M	~ 73.1
choice_3 + channel search	329M	**73.9**
rand sel. blocks + channels	~ 325M	~ 73.4
block search	319M	74.3
block search + channel search	328M	**74.7**
MobileNet V1 (0.75x) [10]	325M	68.4
MobileNet V2 (1.0x) [23]	300M	72.0
ShuffleNet V2 (1.5x) [17]	299M	72.6
NASNET-A [38]	564M	74.0
PNASNET [13]	588M	74.2
MnasNet [24]	317M	74.0
DARTS [15]	595M	73.1
Proxyless-R (mobile)* [4]	320M	74.2 (74.6)
FBNet-B* [26]	295M	74.1 (74.1)

Thanks