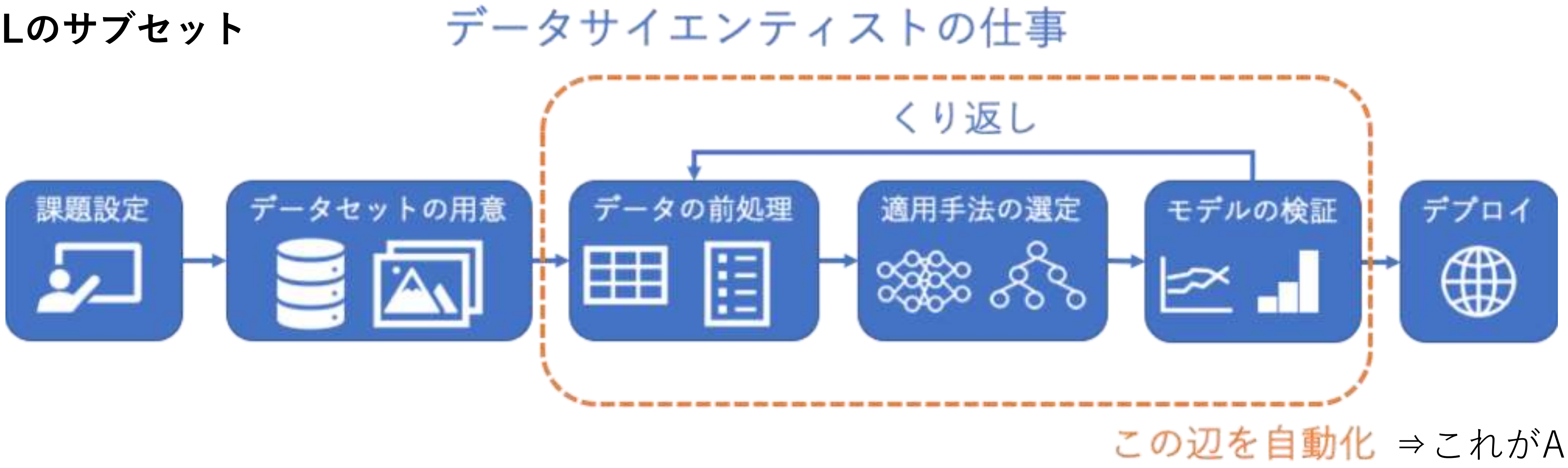
NAS Survey

Coyote#009

NAS(Neural Architecture Search)とは

AutoMLのサブセット



⇒これがAutoML

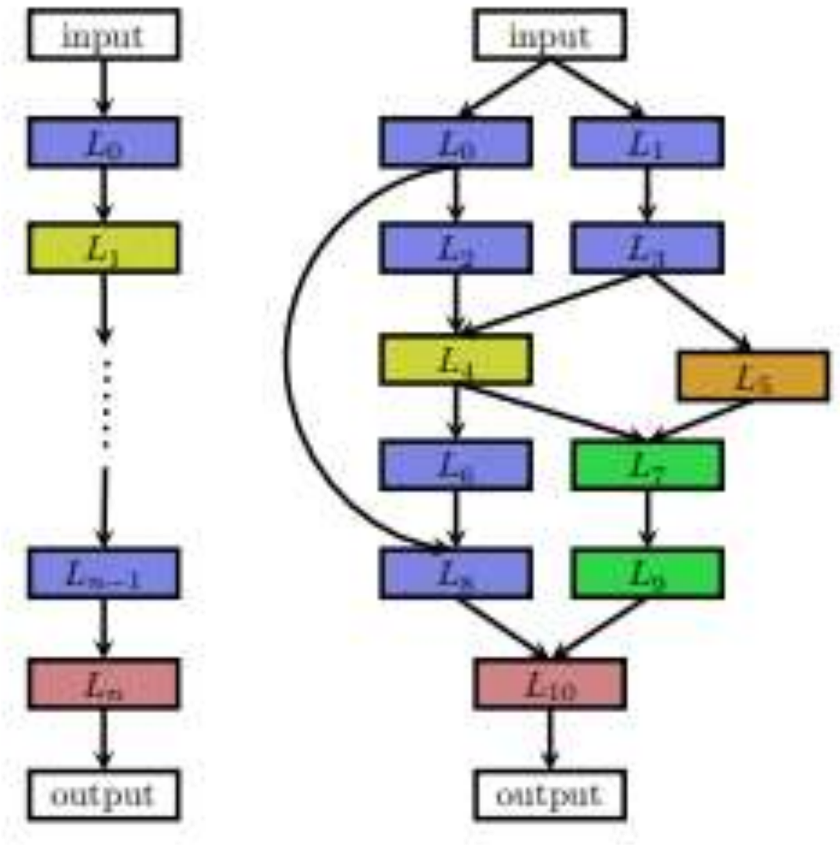
Hyper parameter optimizationやMeta learningとかぶる

Learning rate, Optimizer choice,
Model design related parameters

ターゲットドメインに応じた学習器のバイアス決定
学習の学習という意味で似ている

NAS(Neural Architecture Search)とは

- 最適なネットワーク構造を見つけるための探索



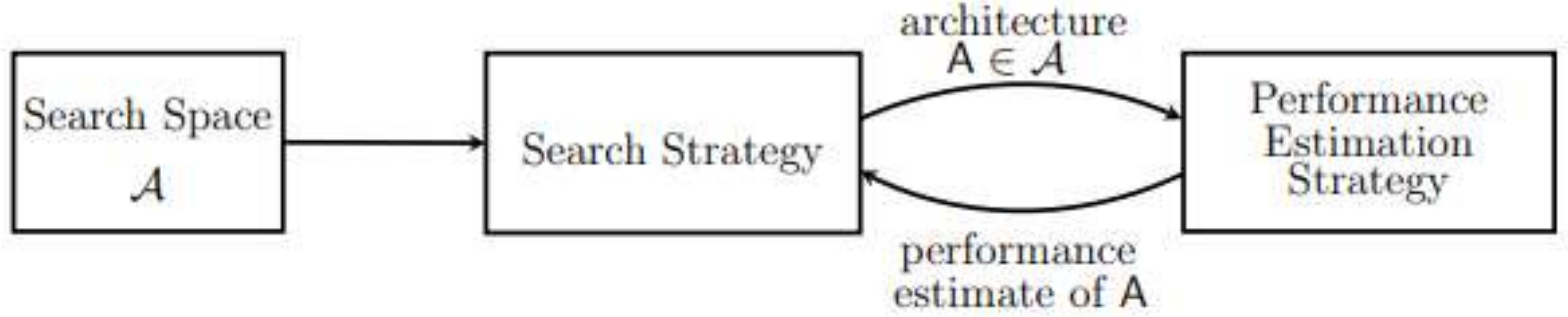
Survey papers

- Elsken, Thomas, Jan Hendrik Metzen, and Frank Hutter. "Neural architecture search: A survey." *arXiv preprint arXiv:1808.05377* (2018). pdf
- Wistuba, Martin, Ambrish Rawat, and Tejaswini Pedapati. "A survey on neural architecture search." *arXiv preprint arXiv:1905.01392* (2019). pdf
- Ren, Pengzhen, et al. "A Comprehensive Survey of Neural Architecture Search: Challenges and Solutions." *arXiv preprint arXiv:2006.02903* (2020). pdf

これを中心に解説

NASの構成要素

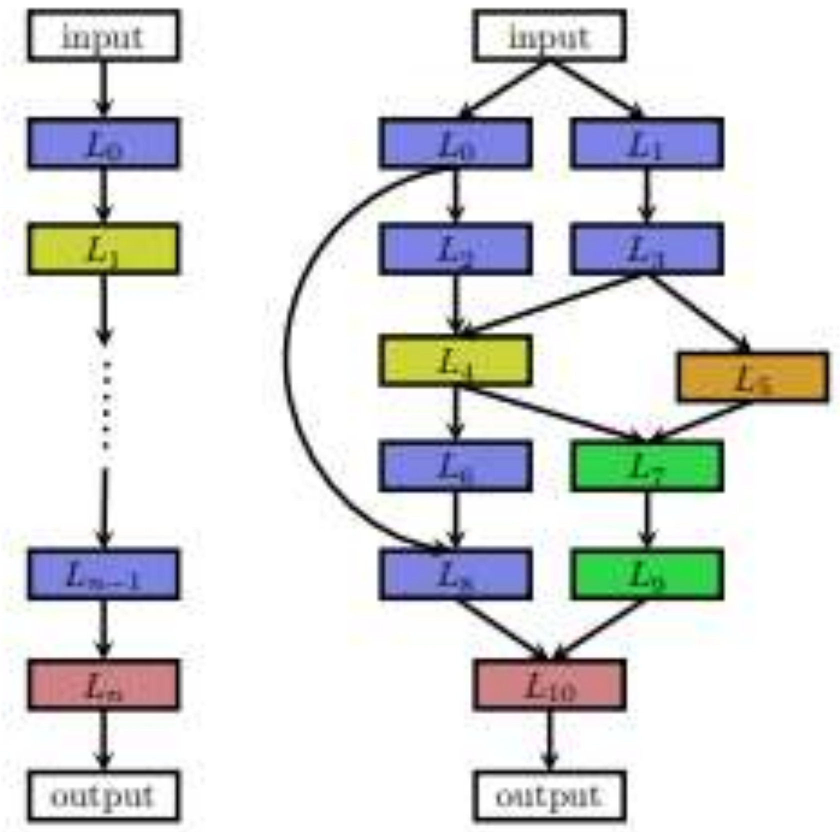
1. Search space
   - どの構造までをサーチ範囲とするか、Search spaceの簡単化
2. Search strategy
   - どういう基準（順番）でサーチするか
3. Performance evaluation strategy
   - サーチ対象の選択基準



1. Search space

- Chain structure
  - 最大Layer数
  - Layerの種類
  - Layer毎のHyper parameter

- Complex structure
  - 複数のLayer出力がLayer入力に
    - ResNet[He2016]タイプ: 加算
    - DenseNet[Huang2017]タイプ: Concatenation
  - ⇒Search spaceが爆発

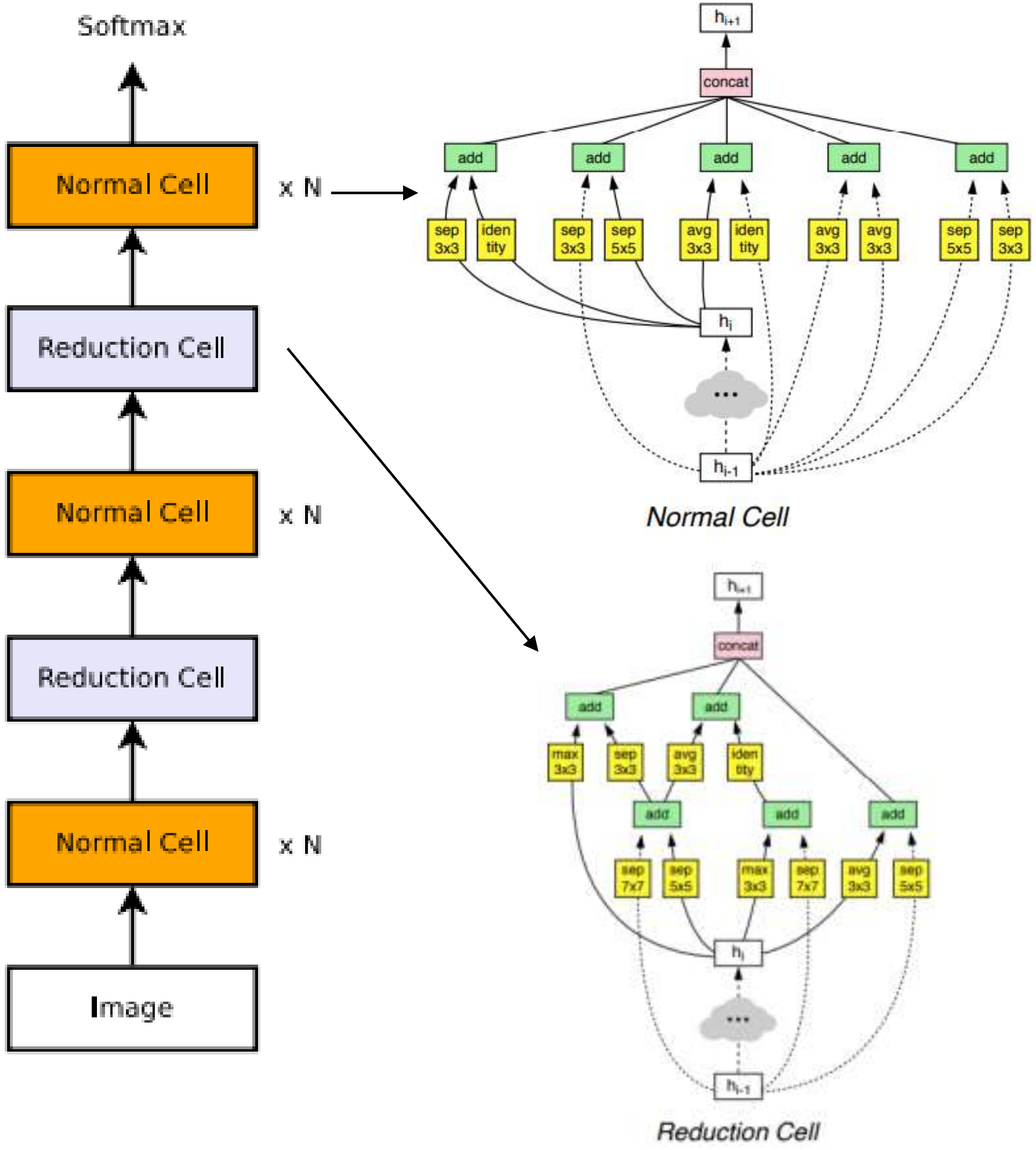


1. Search space

NASNet[Zoph2018]

- 処理量の削減が課題
    - [Zoph2017] ⇒ 800GPU×3～4weeks
    - Search spaceを如何に狭めるか

- Cell-based architecture
    - NASNet[Zoph2018]
    - マクロアーキテクチャは手動で決めて、Cell内だけ最適化



1. Search space

- マクロアーキテクチャはどうやって決めるの？
  - Hierachical search space[Liu2018b]

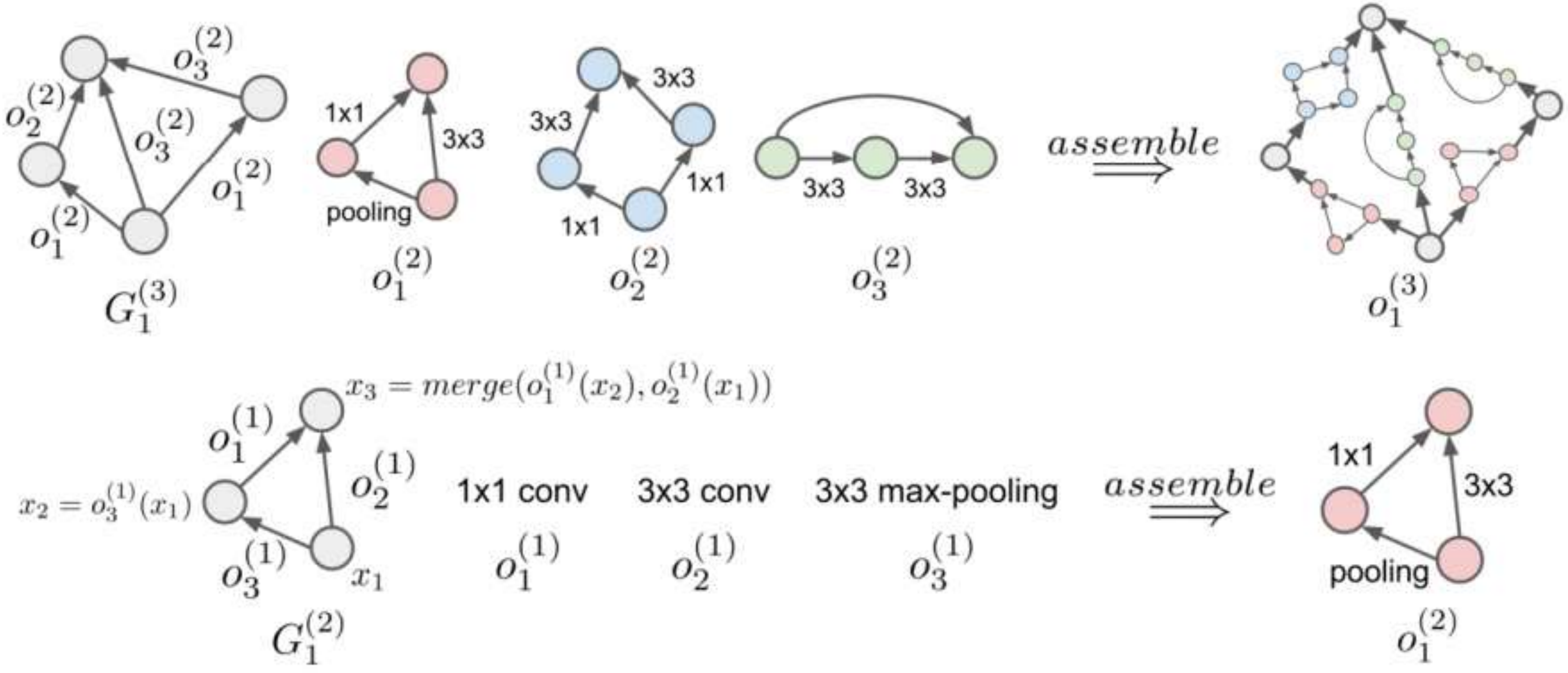


Figure 1: An example of a three-level hierarchical architecture representation. The bottom row shows how level-1 primitive operations $o_1^{(1)}, o_2^{(1)}, o_3^{(1)}$ are assembled into a level-2 motif $o_1^{(2)}$. The top row shows how level-2 motifs $o_1^{(2)}, o_2^{(2)}, o_3^{(2)}$ are then assembled into a level-3 motif $o_1^{(3)}$.

2. Search strategy

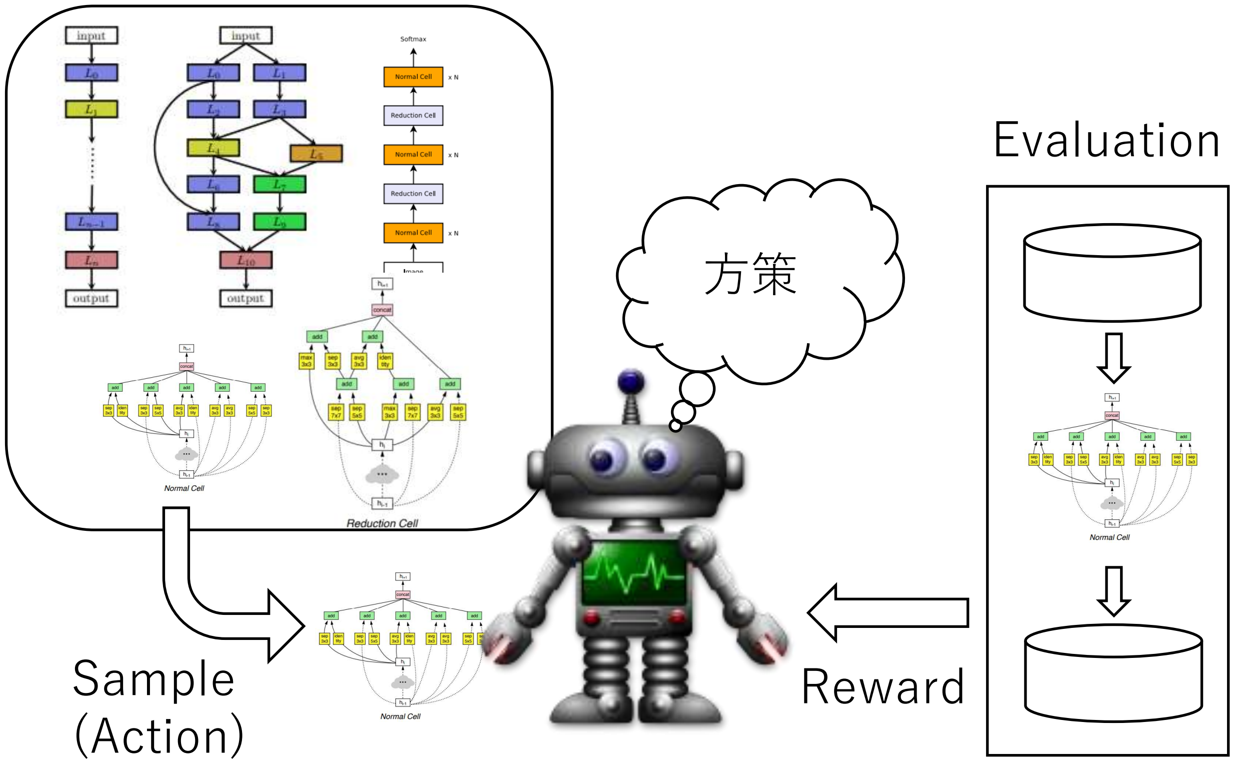
- どういう順番で（離散的な）Searchをするか、もしくは連続的なSearch

- 様々なSearch strategy
  - Random search, grid search, bayesian optimization, evolutionary method, reinforcement learning
  - ↑Hyper parameter optimizationとかぶる

- 最近主流の手法
  - Reinforcement learning
  - Evolutionary method
  - Bayesian optimization
  - Direct gradient based optimization（連続的なSearch）

2. Search strategy

- Reinforcement learning
  - RNN+REINFORCE[Zoph2017]
  - Proximal policy optimization[Zoph2018]
  - Q-learning[Baker2017a]

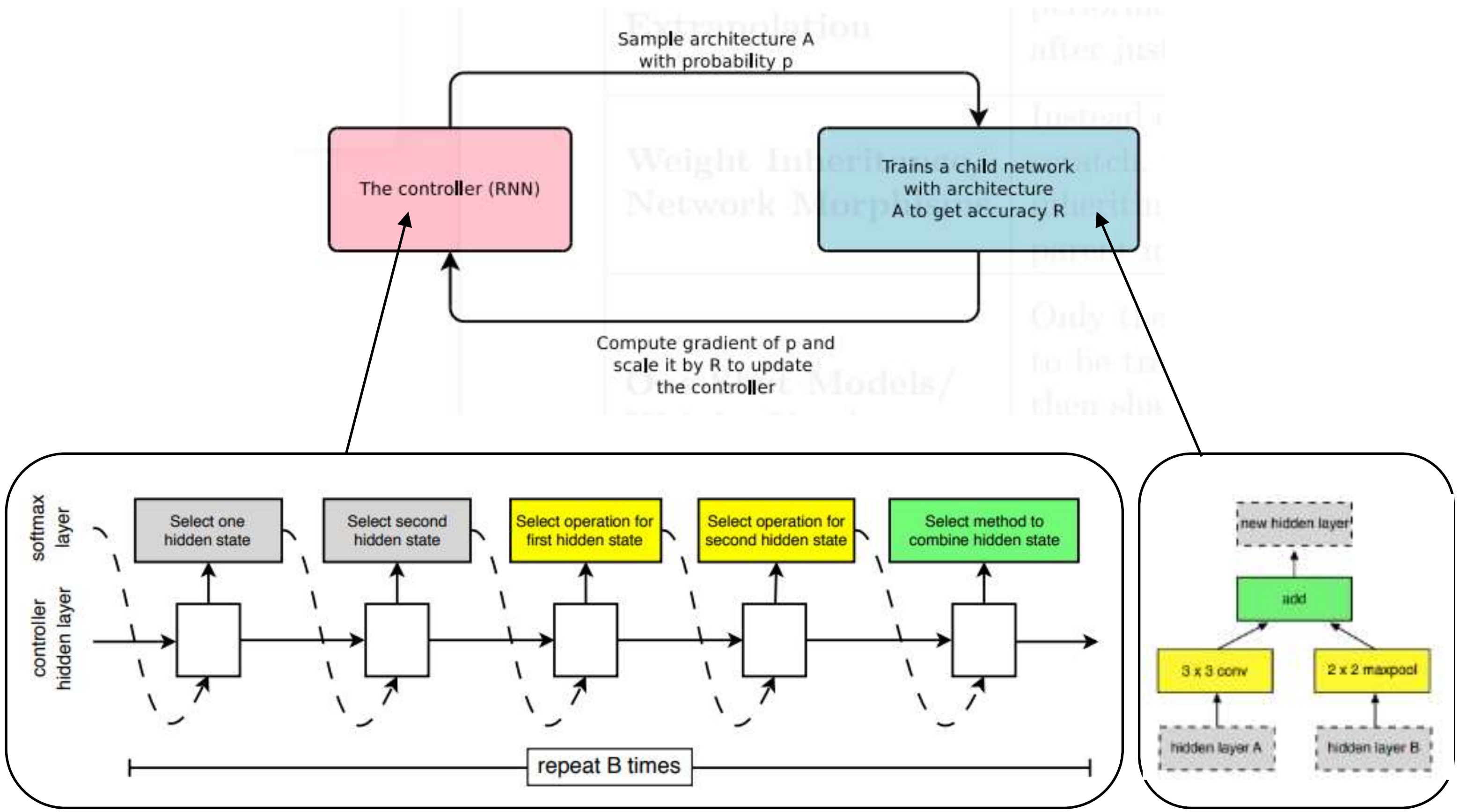


多腕バンディット問題

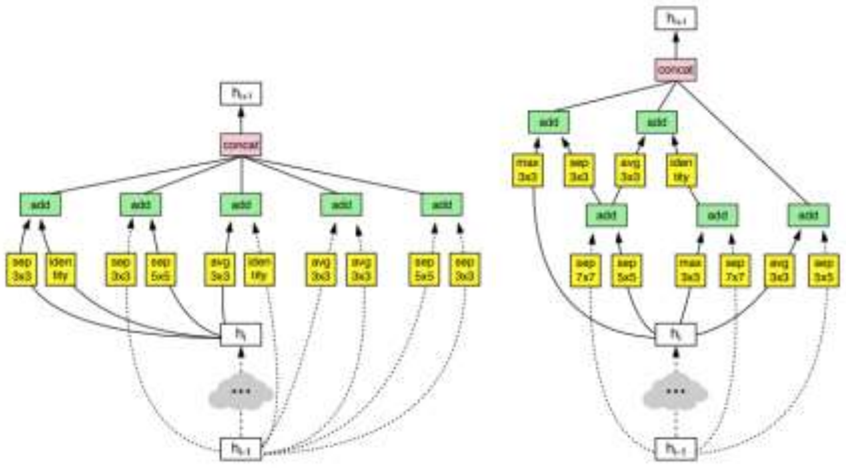
状態が変化しない単純なモデル

2. Search strategy

- Reinforcement learning
  - NASNet[Zoph2018] by Google Brain

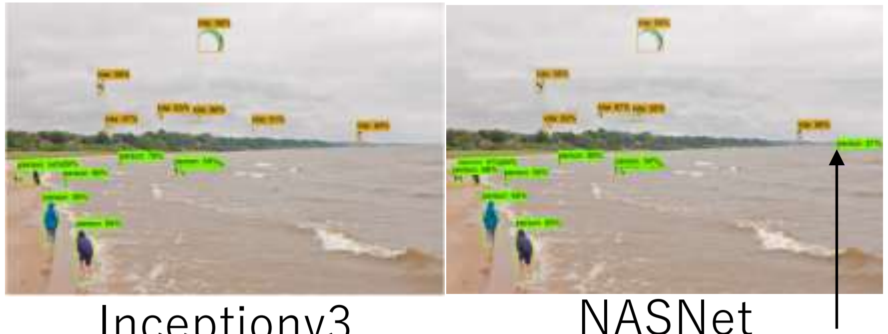


NASNetのContribution
2種類のCellを学習するだけでいい



Normal Cell　　Reduction Cell

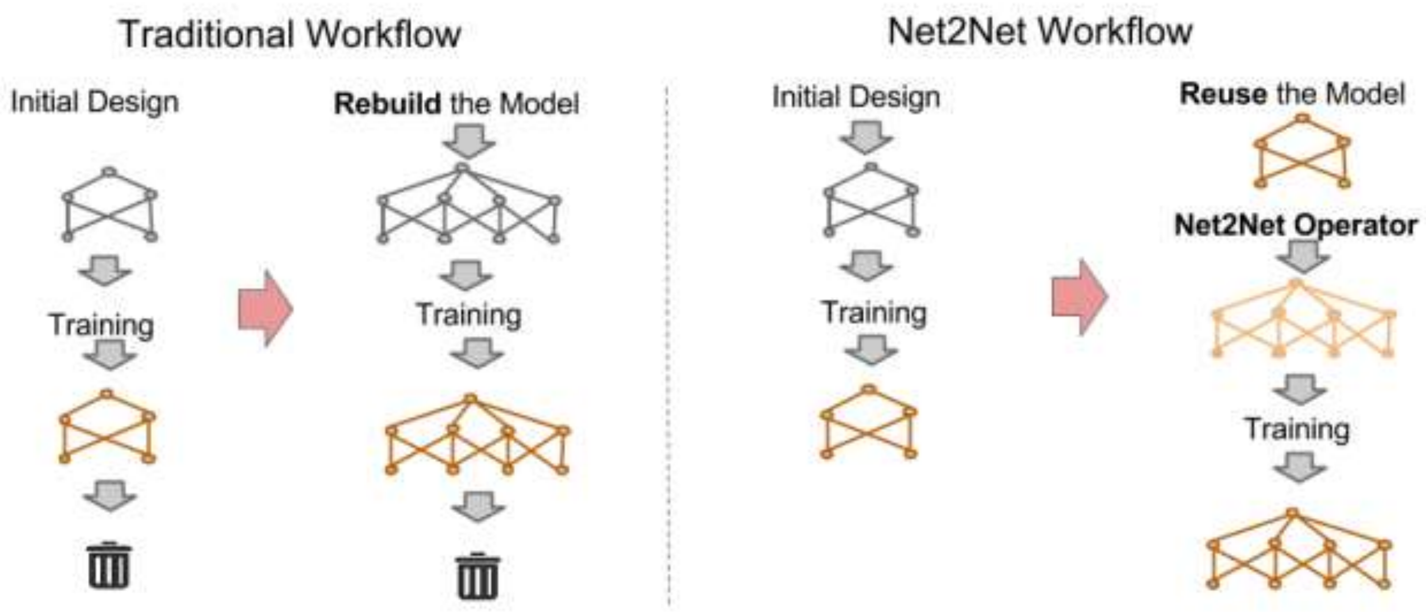
様々なDatasetで優れた結果



Inceptionv3　　NASNet

2. Search strategy

- Reinforcement learning
  - 状態が変化するモデル
  - dubbed network morphisms: Net2Net[Chen2016], [Wei2017]

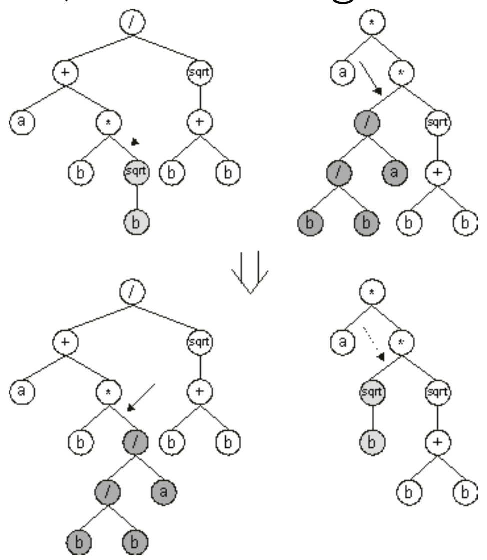


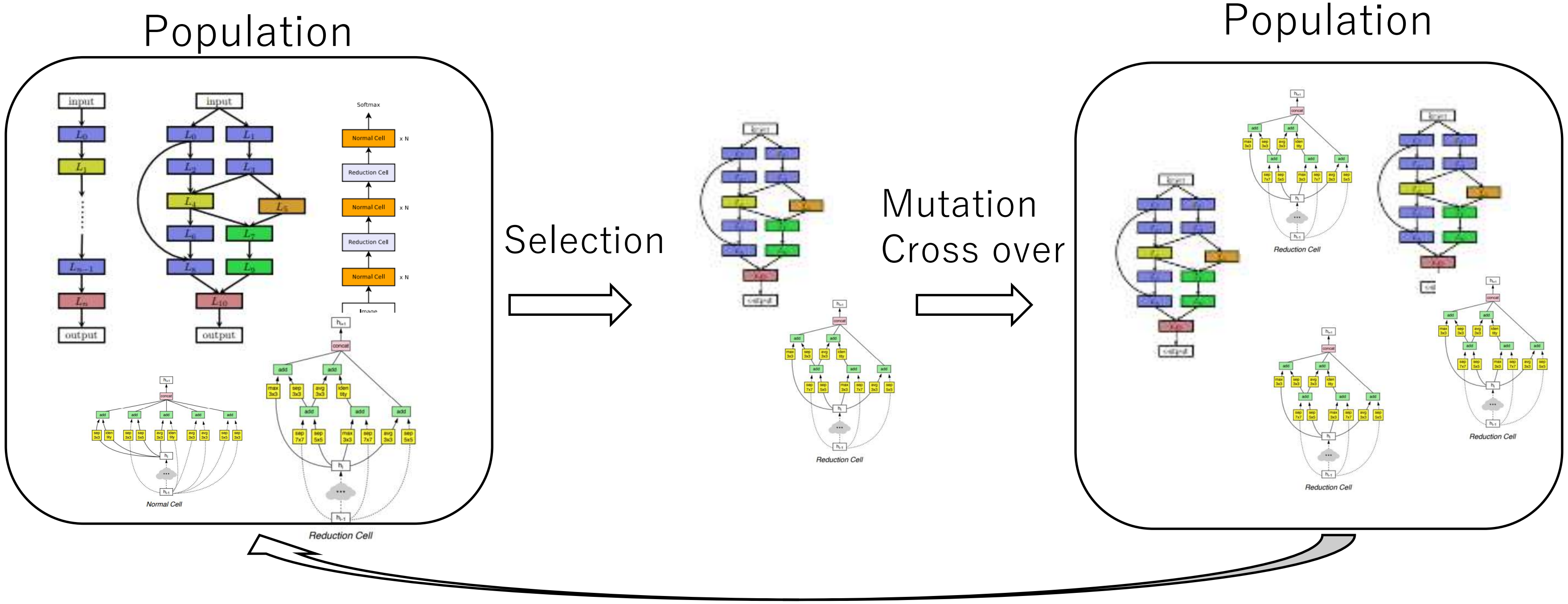
Function preserving transformation
ネットワーク構造を少しずつ変化させながら最適化

2. Search strategy

- Evolutionary method
  - 昔は・・GPで構造探索＋重み探索
  - 今は・・GPで構造探索＋BPで重み最適化
    - Selection・Offspring generation**手法は様々**
      - [Real2017][Suganuma2017][Liu2018b][Real2019][Miikkulainen2017][Xie2017][Elsken2019]

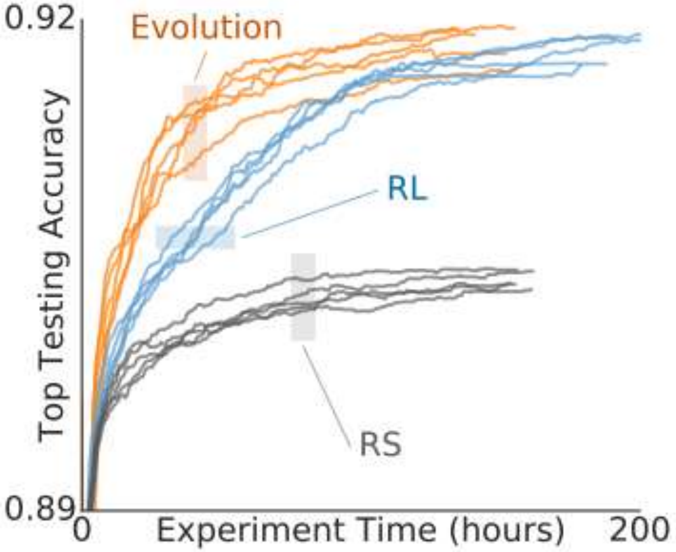
GP (Genetic Programming)





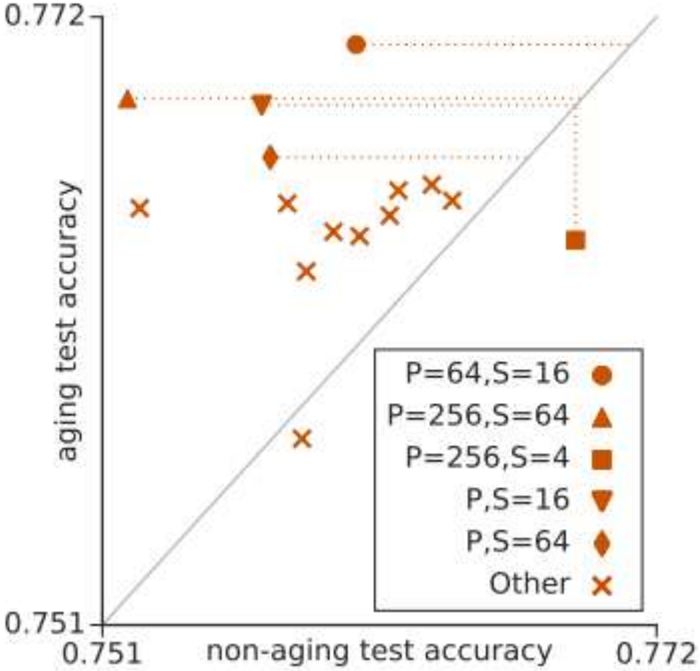
2. Search strategy

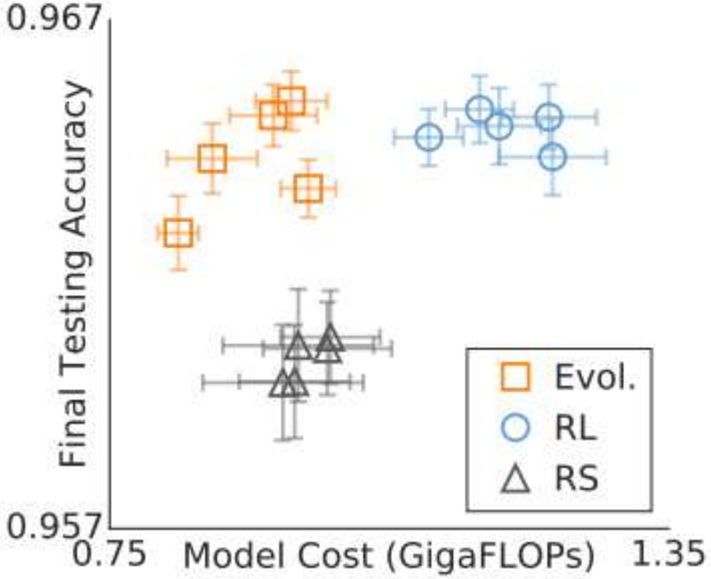
RL・RSより勝っている



- Evolutionary method
  - AmoebaNet[Real2019]
    - Selection時に、Age（年齢）考慮し若い個体を優遇
    - Search spaceはNASNetと同様

Agingの効果
- **“たまたま”**よい**Model**が生成されることがある
- 通常のSelectionではそのような**Architecture**でも長く生き残ってしまう
- Agingをすることで安定的によい**Model**を生み出す**Architecture**のみが生き残る

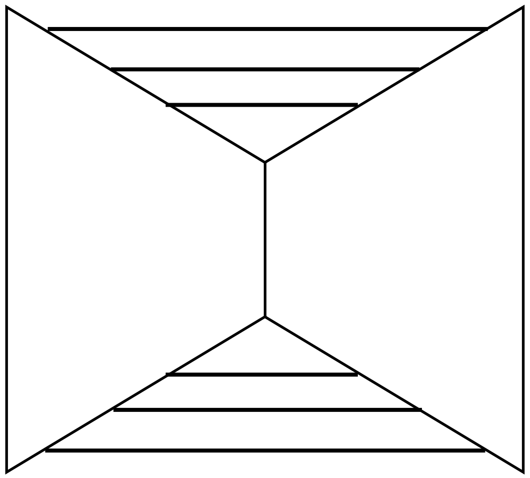




2. Search strategy

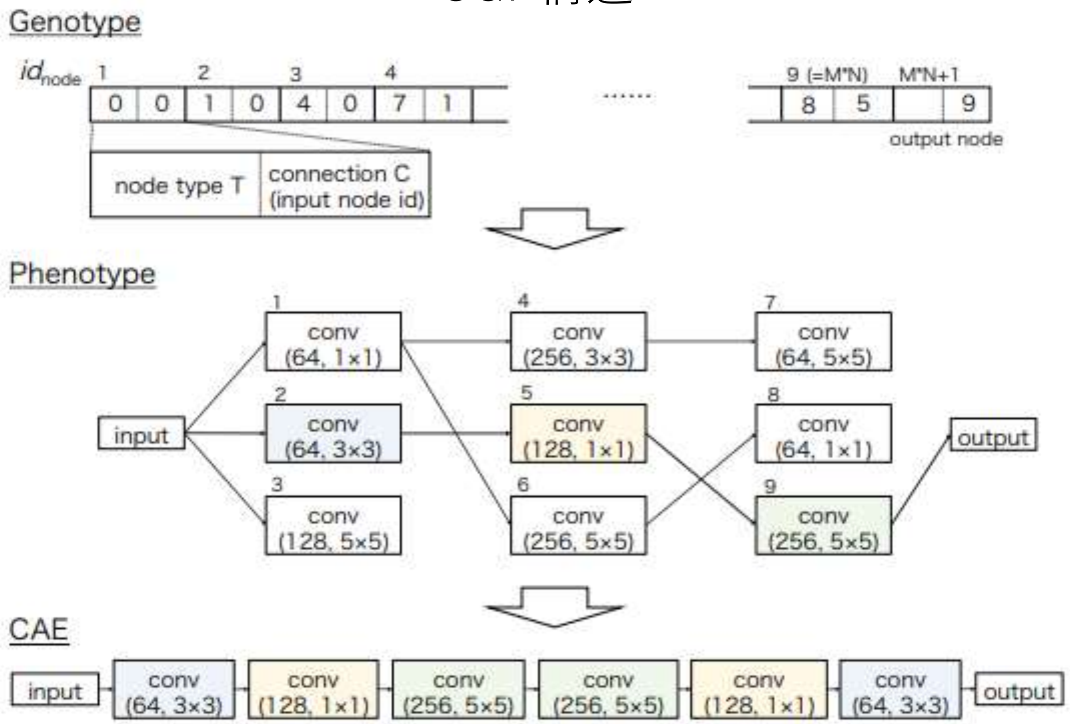
- Evolutionary method
  - Image restoration[Suganuma2017]
    - “SimpleなCAE(Convolutional Auto-Encoder)でGANに勝てる”

対象なAE構造とSkip Connection
=UNet構造



Lossは単純なMSE

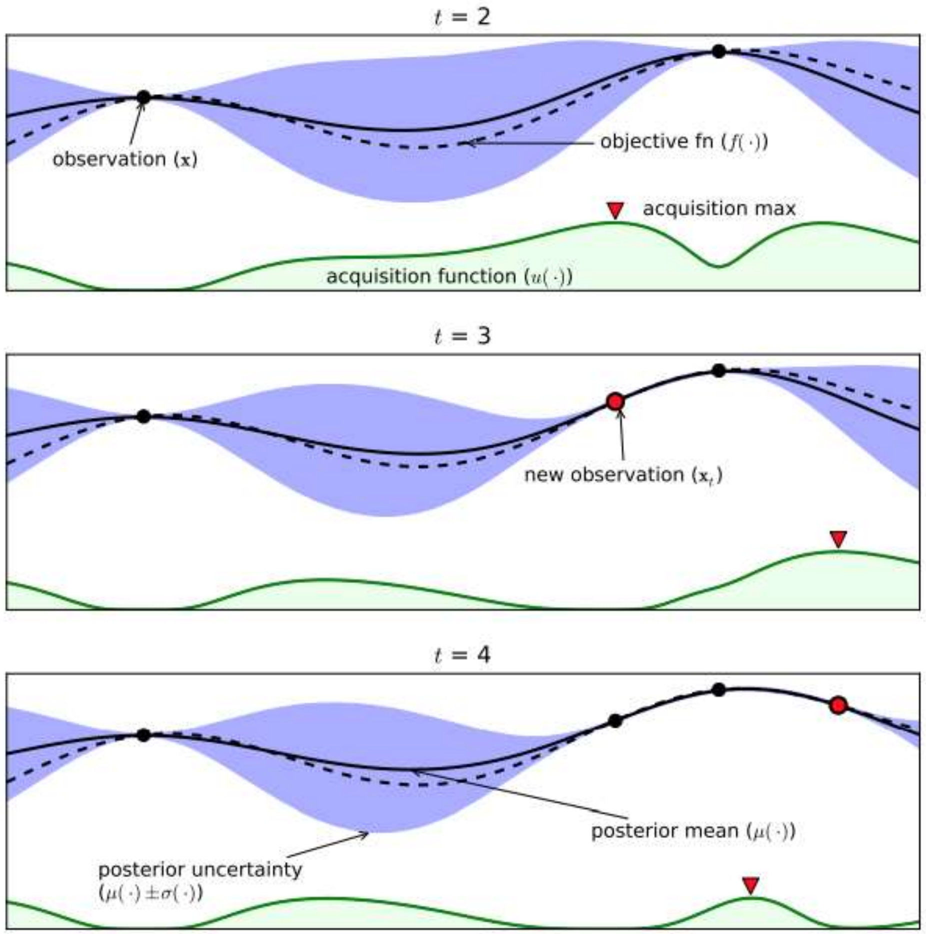
CGP構造



2. Search strategy

- Bayesian optimization
  - GP（ガウス過程）ベース、ツリーベースの手法
  - ArchitectureとHyper parameterを同時に最適化
  - Evolutionary methodよりもいいという報告も [Klein2018]

  - BOは、基本的にはRegression問題
    - 各Architectureの評価関数をRegressionしたい
    - 逐次的に{Architecture,評価値}のセットが与えられ、少しずつ評価関数の確度が上がる
    - これまでに与えられたデータから求める分布を事前分布として、今回与えられたデータからベイズの定理で事後分布を求める
    - RLに似ているところがある（Exploration & Exploitation）

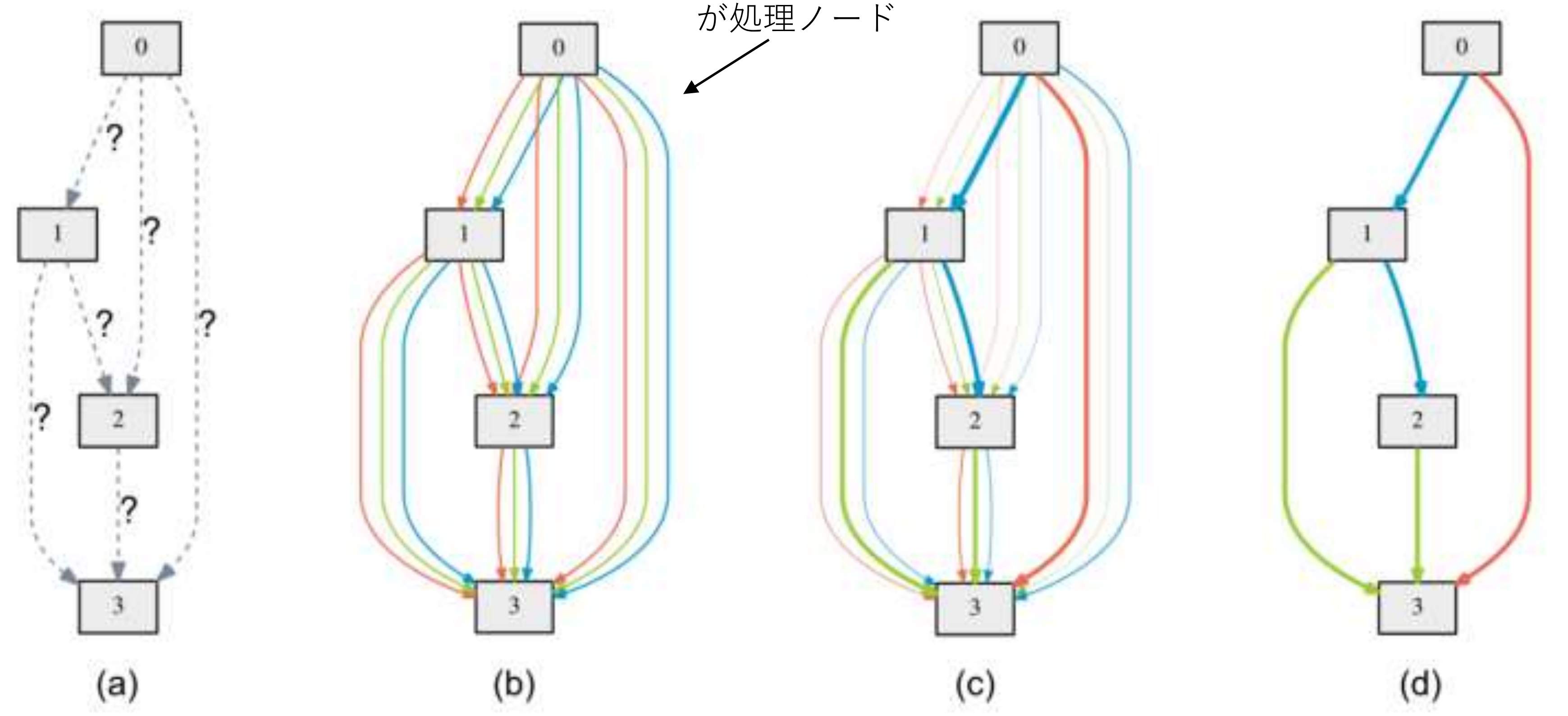


Brochu, Cora & de Freitas (2010)

2. Search strategy

- Direct gradient based optimization
  - 離散的な構造探索を連続的に
  - Continuous relaxation: DARTS[Liu2019b]

それぞれの辺
が処理ノード



複数の処理を並列してつないでおき、まとめて学習、最後にそのうちのどれかを選ぶ

3. Performance estimation

Speed-up method	How are speed-ups achieved?	References
Lower fidelity estimates	Training time reduced by training for fewer epochs, on subset of data, downscaled models, downscaled data, ...	Li et al. (2017), Zoph et al. (2018), Zela et al. (2018), Falkner et al. (2018), Real et al. (2019), Runge et al. (2019)
Learning Curve Extrapolation	Training time reduced as performance can be extrapolated after just a few epochs of training.	Swersky et al. (2014), Domhan et al. (2015), Klein et al. (2017a), Baker et al. (2017b)
Weight Inheritance/ Network Morphisms	Instead of training models from scratch, they are warm-started by inheriting weights of, e.g., a parent model.	Real et al. (2017), Elsken et al. (2017), Elsken et al. (2019), Cai et al. (2018a,b)
One-Shot Models/ Weight Sharing	Only the one-shot model needs to be trained; its weights are then shared across different architectures that are just subgraphs of the one-shot model.	Saxena and Verbeek (2016), Pham et al. (2018), Bender et al. (2018), Liu et al. (2019b), Cai et al. (2019), Xie et al. (2019)

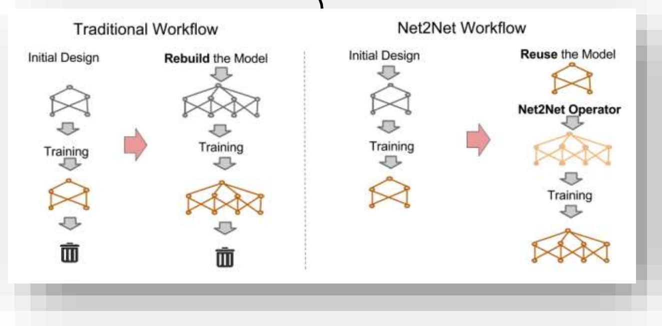
[Elsken2018]

3. Performance estimation

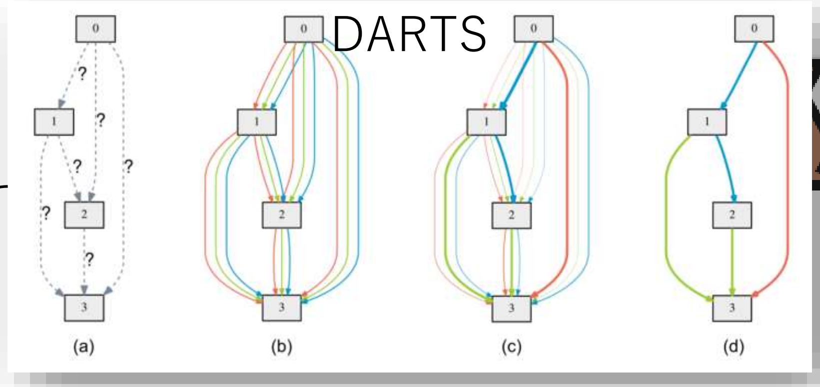
- Lower fidelity estimates
  - 精度は悪いほうにでるが、相対精度が変わらなければOKという考え
    - が、相対精度も結構変わる[Zela2018]
  - 少しずつFidelityを上げていくのがよい[Li2017][Falkner2018]

- Learning curve extrapolation
  - 代理モデルを用いて新たなArchitectureの学習を予測する[Liu2018a]

- Weight inheritance / Network morphism
  - Dubbed network morphism[Wei2016] ←既に説明済み
  - 課題はネットワークを大きくすることしかできないこと
  - ⇒Shrinkを考慮した手法[Elsken2019]



3. Performance estimation



- One shot models / Weight sharing
  - 既に説明済み
  - 学習は一度だけ、その後各ArchitectureのEvaluationのみ行う
  - 精度は悪いほうにでるが、相対精度が変わらなければOKという考え
    - が、相対精度も結構変わる[Bender2018][Sciuto2019]
  - ENAS[Pham2018]
  - DARTS[Liu2019b]
  - SNAS[Xie2019]

どれがいいか？

比較表[Ren2020]

ImageNet

Search mothed	Reference	Venue	Top 1/Top 5 Acc(%)	Params (Millions)	Image Size (squarred)	GPU Days
Human	Mobilenets [6]	CoRR17	70.6/89.5	4.2	224	-
	ResNeXt [138]	CVPR17	80.9/95.6	83.6	320	-
	Polynet [139]	CVPR17	81.3/95.8	92.0	331	-
	DPN [140]	NIPS17	81.5/95.8	79.5	320	-
	Shufflenet [137]	CVPR18	70.9/89.8	5.0	224	-
RL	NASNet [31]	CVPR18	82.7/96.2	88.9	331	2,000
	NASNet-A [31]	CVPR18	74.0/91.6	5.3	224	2,000
	Block-QNN [32]	CVPR18	77.4/93.5	N/A	224	96
	Path-level EAS [56]	ICML18	74.6/91.9	594	224	200
	FPNAS [38]	ICCV19	73.3/N/A	3.41	224	0.8
EA	GeNet [16]	ICCV17	72.1/90.4	156	224	17
	Hierarchical-EAS [33]	ICLR18	79.7/94.8	64.0	299	300
	AmoebaNet [42]	AAAI19	82.8/96.1	86.7	331	3,150
	AmoebaNet [42]	AAAI19	83.9/96.6	469	331	3,150
GO	Understanding One-Shot Models [22]	ICML18	75.2/N/A	11.9	224	N/A
	SMASH [23]	ICLR18	61.4/83.7	16.2	32	3
	PARSEC [130]	CoRR19	74.0/91.6	5.6	N/A	1
	DARTS [17]	ICLR19	73.3/91.3	4.7	224	4
	SNAS [45]	ICLR19	72.7/90.8	4.3	224	1.5
	ProxylessNAS [103]	ICLR19	75.1/92.5	N/A	224	8.33
	GHN [90]	ICLR19	73.0/91.3	6.1	224	0.84
	SGAS [151]	CVPR20	75.62/92.6	5.4	224	0.25
	PC-DARTS (CIFAR10) [83]	ICLR20	74.9/92.2	5.3	224	0.1
	PC-DARTS (ImageNet) [83]	ICLR20	75.8/92.7	5.3	224	3.8
RS	Hierarchical-EAS Random [33]	ICLR18	79.0/94.8	N/A	299	300
SMBO	PNAS [36]	ECCV18	74.2/91.9	5.1	224	225
	PNAS [36]	ECCV18	82.9/96.2	86.1	331	225

Cifar-10 →

Search method	Reference	Venue	Optimization Strategy: Modular search strategy	Optimization Strategy: Continuous search space	Optimization Strategy: Architecture recycle	Optimization Strategy: Incomplete training	Error Acc (%)	Params (Millions)	GPU Days
Human	WRN [131]	CVPR16					3.87	36.2	-
	Shark [132]	CoRR17					3.55	2.9	-
	PyramidSepDrop + ShakeDrop[133]	CoRR16					2.67	26.2	-
	ResNet [134]	ECCV16					6.41	1.7	-
	Fractalnet [135]	ICLR17					5.22	38.6	-
	DenseNet-BC [35]	CVPR17					3.46	25.6	-
RL	NAS-RL [11]	ICLR17					3.65	37.4	22,400
	MetaQNN [12]	ICLR17					6.92	11.18	100
	EAS [50]	AAAI18			✓	✓	4.23	23.4	10
	NASNet-A [31]	CVPR18	✓				3.41	3.3	2,000
	NASNet-A + Cutout [31]	CVPR18	✓				2.65	3.3	2,000
	Block-QNN [32]	CVPR18	✓				3.54	39.8	96
	Path-level EAS [56]	ICML18			✓	✓	2.99	5.7	200
	Path-level EAS + Cutout [56]	ICML18			✓	✓	2.49	5.7	200
	N2N learning [51]	ICLR18			✓	✓	6.46	3.87	2.1
	ProxylessNAS-R + Cutout [103]	ICLR19				✓	2.30	5.8	N/A
	FPNAS + Cutout [38]	ICCV19	✓			✓	3.01	5.76	0.8
EA	Large-scale Evolution [15]	ICML17				✓	5.40	5.4	2,600
	GeNet [16]	ICCV17					5.39	N/A	17
	Genetic Programming CNN [5]	GECCO17					5.98	1.7	14.9
	Hierarchical-EAS [33]	ICLR18	✓				3.75	15.7	300
	NASH-Net [84]	ICLR18			✓	✓	5.20	19.7	1
	Neuro-Cell-based Evolution + Cutout [128]	ECML PKDD18	✓		✓	✓	3.57	5.8	0.5
	AmoebaNet [42]	AAAI19	✓				3.34	3.2	3,150
GO	ENAS + micro [19]	ICML18	✓	✓		✓	3.54	4.6	0.5
	ENAS + micro + Cutout [19]	ICML18	✓	✓		✓	3.54	4.6	0.5
	ENAS + macro [19]	ICML18		✓		✓	4.23	21.3	0.32
	SMASH [23]	ICLR18		✓		✓	4.03	16	1.5
	Understanding One-Shot Models [22]	ICML18	✓	✓		✓	4.00	5.0	N/A
	DARTS (1^{st} order) + Cutout [17]	ICLR19	✓	✓		✓	3.0	3.3	1.5
	DARTS (2^{nd} order) + Cutout [17]	ICLR19	✓	✓		✓	2.76	3.3	4
	SNAS + Cutout [45]	ICLR19	✓	✓		✓	2.85	2.8	1.5
	PARSEC + Cutout [130]	CoRR19	✓	✓		✓	2.81	3.7	1
	GHN [90]	ICLR19	✓	✓		✓	2.84	5.7	0.84
	ProxylessNAS-G + Cutout [103]	ICLR19		✓		✓	2.08	5.7	N/A
	BayesNAS [136]	ICML19	✓	✓		✓	2.81	3.4	0.2
	P-DARTS + Cutout [43]	ICCV19	✓	✓		✓	2.50	3.4	0.3
	DATA + Cutout [74]	NeurIPS19	✓	✓		✓	2.59	3.4	1
	SGAS [151]	CVPR20	✓	✓		✓	2.66	3.7	0.25
	GDAS-NSAS [25]	CVPR20	✓	✓		✓	2.73	3.54	0.4
	PC-DARTS + Cutout [83]	CVPR20	✓	✓		✓	2.57	3.6	0.1
RS	Hierarchical-EAS Random [33]	ICLR18	✓				3.91	N/A	300
	NAO Random-WS [73]	NeurIPS18	✓			✓	3.92	3.9	0.3
	NASH-Net Random [84]	ICLR18					6.5	4.4	0.2
	DARTS Random [17]	ICLR19	✓				3.29	3.2	4
	RandomNAS + Cutout [104]	UAI19	✓			✓	2.85	4.3	2.7
	RandomNAS-NSAS [25]	CVPR20	✓				2.64	3.08	0.7
SMBO	NASBOT [61]	NeurIPS18					8.69	N/A	1.7
	PNAS [36]	ECCV18	✓			✓	3.41	3.2	225
	NAO [73]	NeurIPS18	✓	✓		✓	2.98	28.6	200
	NAO-WS [73]	NeurIPS18	✓	✓		✓	3.53	2.5	0.3
	NAO-Cutout [73]	NeurIPS18	✓	✓		✓	2.11	128	200

GO: Gradient Optimization
RS: Random Search
SMBO: Sequential Model Based Optimization

今後のResearch direction

- 共通のベンチマークが必要
  - 様々な要因で精度が大きく変わる、Learning rate, data augmentation, etc.
  - 純粋にArchitecture要因での性能を見たい
  - [Ying2019], [Klein2018]
- 画像分類以外の応用
  - Image restoration[Suganuma2018], Semantic segmentation[Chen2018], Transfer learning, RNN for language or music modeling, GAN, sensor fusion
- Multi-objective
  - Compression by NAS[Han2015], [Liu2017], [Gordon2018], [Liu2019c], [Cao2019]

Reference

- 資料中のReferenceは、基本的には下記Survey論文のReferenceから

    - Elsken, Thomas, Jan Hendrik Metzen, and Frank Hutter. "Neural architecture search: A survey." arXiv preprint arXiv:1808.05377 (2018).

- 性能比較表のみ下記Survey論文のReferenceから

    - Ren, Pengzhen, et al. "A Comprehensive Survey of Neural Architecture Search: Challenges and Solutions." arXiv preprint arXiv:2006.02903 (2020).