

“Weight Agnostic Neural Networks“

by Adam Gaier & David Ha, in 2019; <https://arxiv.org/abs/1906.04358>

&

“Consciousness and Subconsciousness as a Means of AGI’s and Narrow AI’s Integration”

by AA. Sukhobokov; <https://link.springer.com/bookseries/11156>

Adam Gaier ; Bonn-Rhein-Sieg 応用科学大学

2021年末は、サンフランシスコの Autodesk社のAI Lab勤務

David Ha : Google Brain in Tokyo, Japan

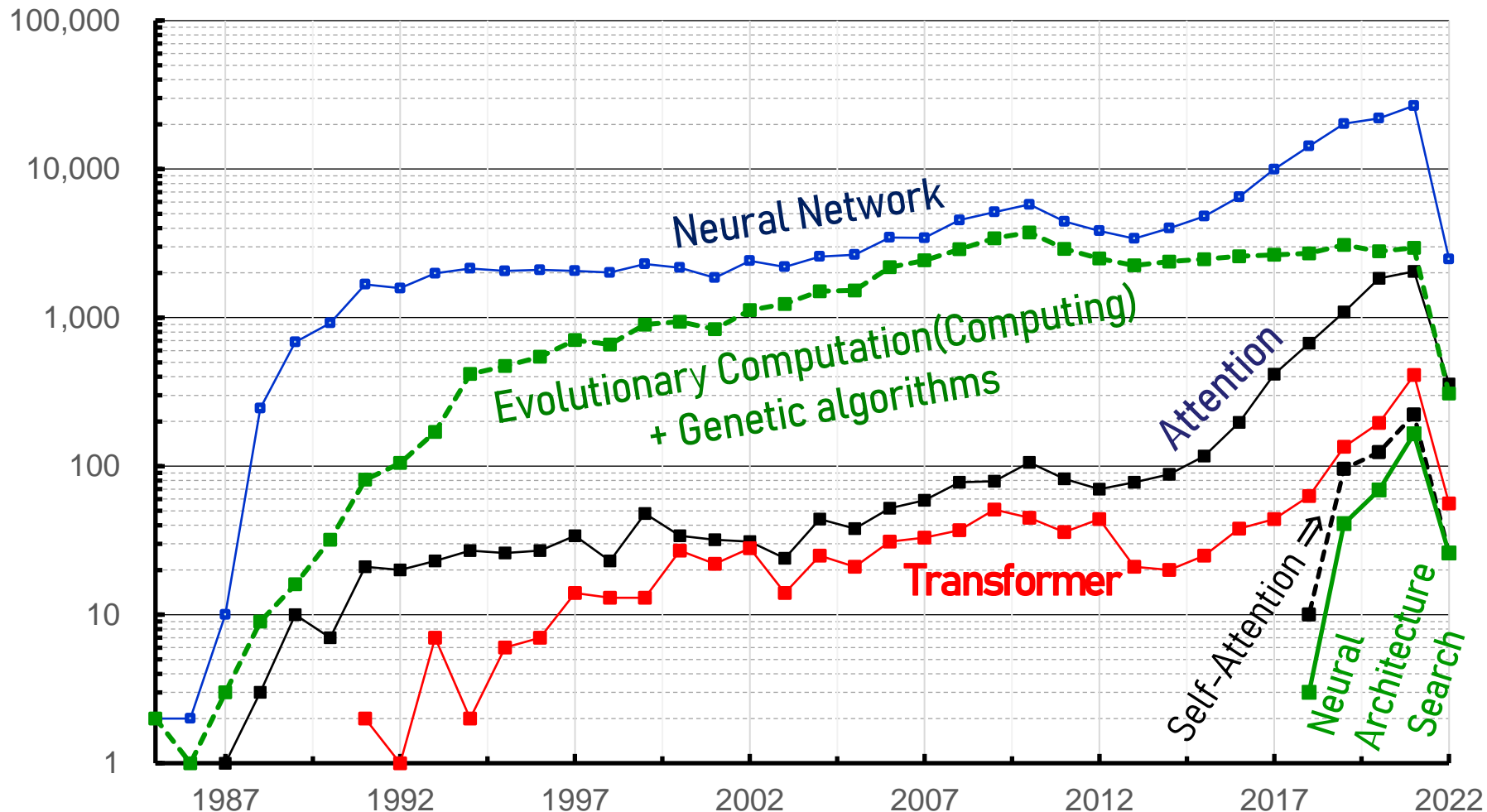
A. Sukhobokov ; SAPプリンシパルコンサルタント兼BMSTU講師

2022, 03, 28

EC/GA、NASへの関心状況

Abstractにそれらの単語を含む論文数

- NNは、2015年以降急増 (DL、Attention、Self-Attention、etc)
- EC (進化的計算, Evolutionary Computing(Computing)+ Genetic Algorithms)は増えていないが、NNと融合中。 NSA (Neural Architecture Search)は急増中



Feedback機構 (Loop構造) と Consciousnessへの注目する研究者

Norbert Wiener (1894-1964) ; MIT

サイバネティックス、ロボティクスやオートメーションの先駆者
独自の人間機械論 (Feed-back Loopを持つ情報機構)



Marvin Minsky (1927-2016) ; MIT

ダートマス会議の発起人の一人 (John McCarthyとともに)
Mindの働きに注目



Jiri Wiedermann ; ユトレヒト大学(オランダ)

[2018] Finite State Machines with Feedback :
An Architecture Supporting Minimal Machine Consciousness

[2020] Towards Minimally Conscious Finite-State Controlled
Cyber-Physical Systems : A Manifesto



A. A. Sukhobokov ; SAPプリンシパルコンサルタント兼BMSTU講師

[2018] Business analytics and AGI in corporate management systems

[2019] Consciousness and Subconsciousness as a Means of AGI's and
Narrow AI's Integration

[2020] AGI components for enterprise management systems



なぜ、それらに注目するか？

1) ネットワークの(再)構築方法に注目するため（複数ネットワークの統合含む）

- **進化的計算(進化に基づく選択圧力)**は、何でも学べる汎用学習アルゴリズム (Neural Architecture Search、Evolutional Computing、Genetic algorithms)

[106] K. O. Stanley and R. Miikkulainen (2001);

“Evolving neural networks through augmenting topologies.”

- 既に、ニューラルネットワーク技術と進化的計算の融合は進んで来ている。

ただし、現在の進化的計算アルゴリズムは畳み込み層を生成しないとされる。Loop構造やFeedback構造も生成しない。

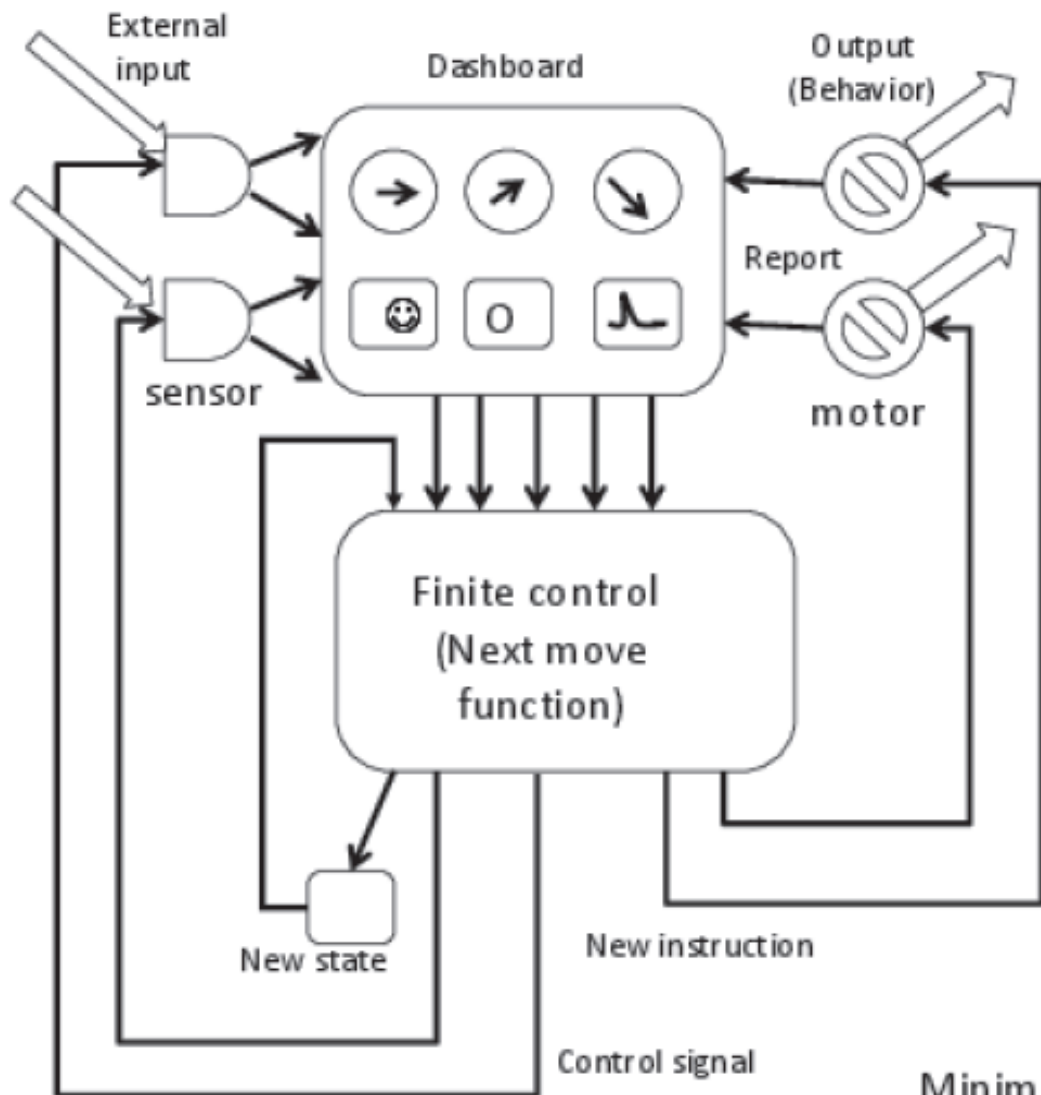
2) Loop構造やFeedback構造に基づくConsciousnessも、知的情報生成に重要

- Jiri Wiedermann(ユトレヒト大学)や、AA. Sukhobokov(SAP)
- AA. Sukhobokovは、Enterprise-IT向けのAGIニーズが大きいとする。

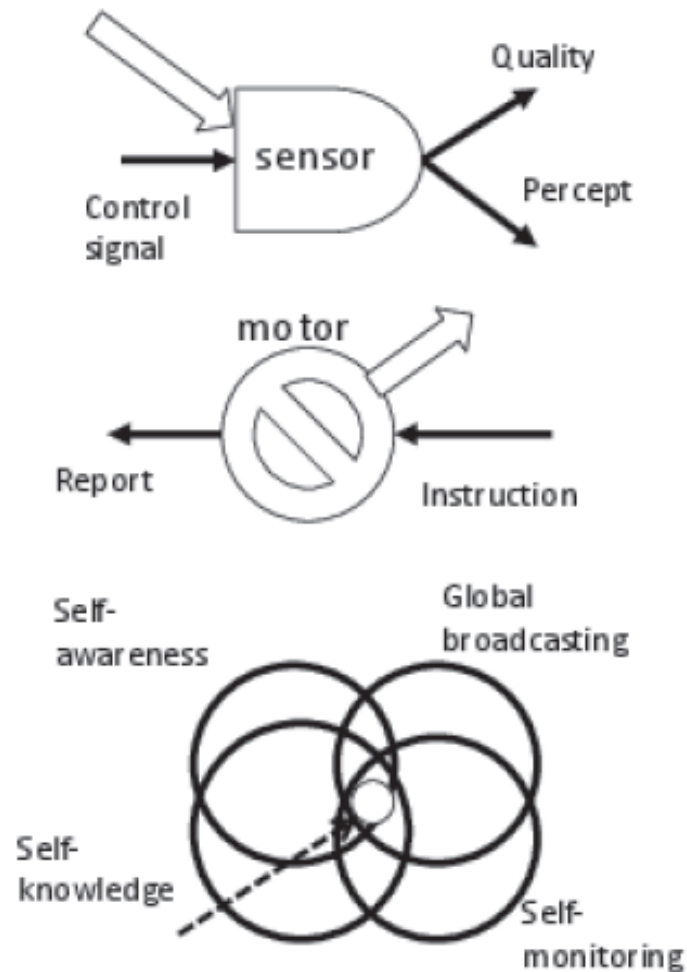
“Cognitive Cyber-Physical Systems”

proposed by Jiri Wiedermann and Jan van Leeuwen (2020).

Jiri Wiedermann and Jan van Leeuwen (2020); “Towards Minimally Conscious Finite-State Controlled Cyber-Physical Systems: A Manifesto?”



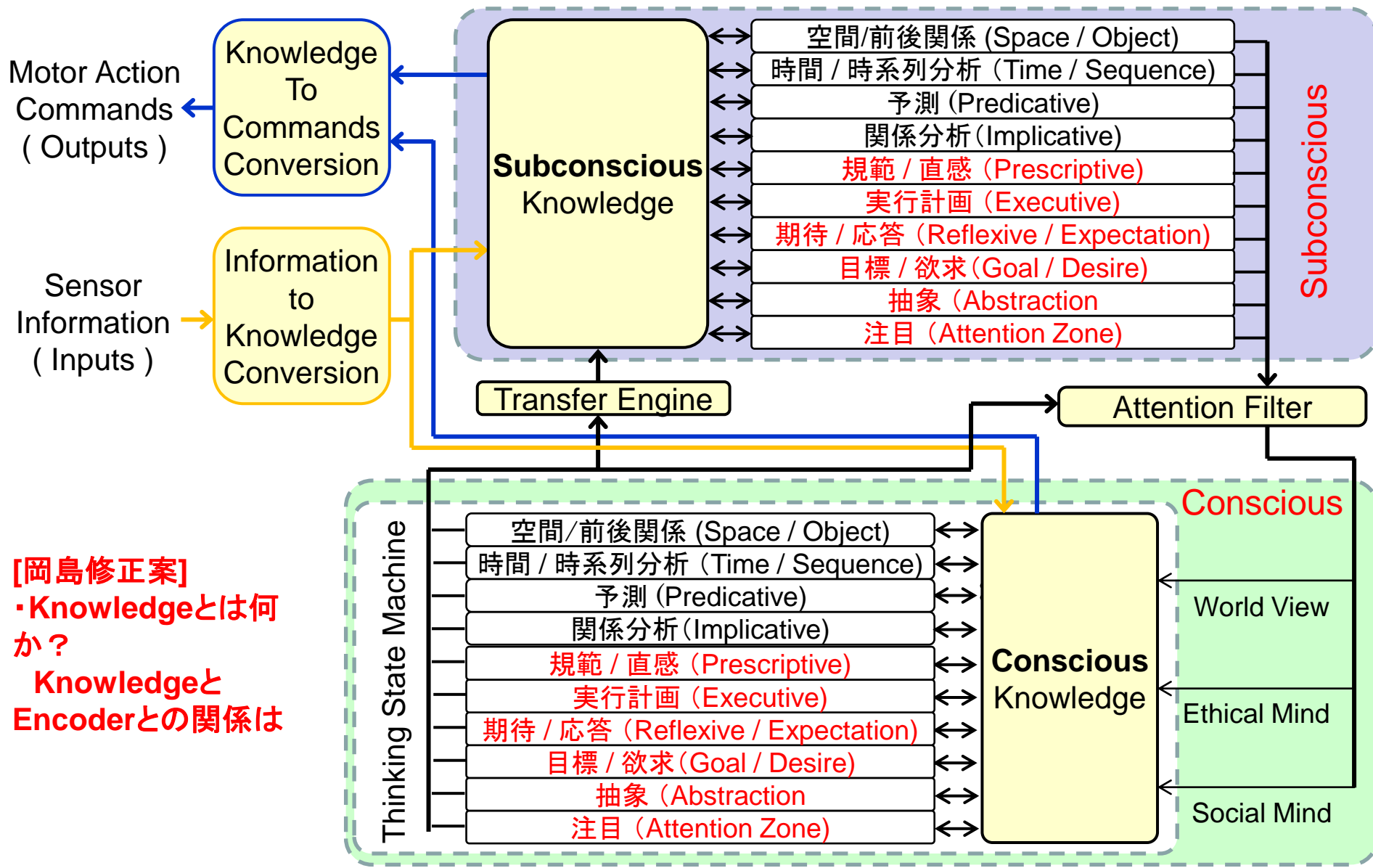
A detail of the feedback between S/M units and finite state control



Minimal machine consciousness

The proposed architecture ; An agent to form the consciousness.

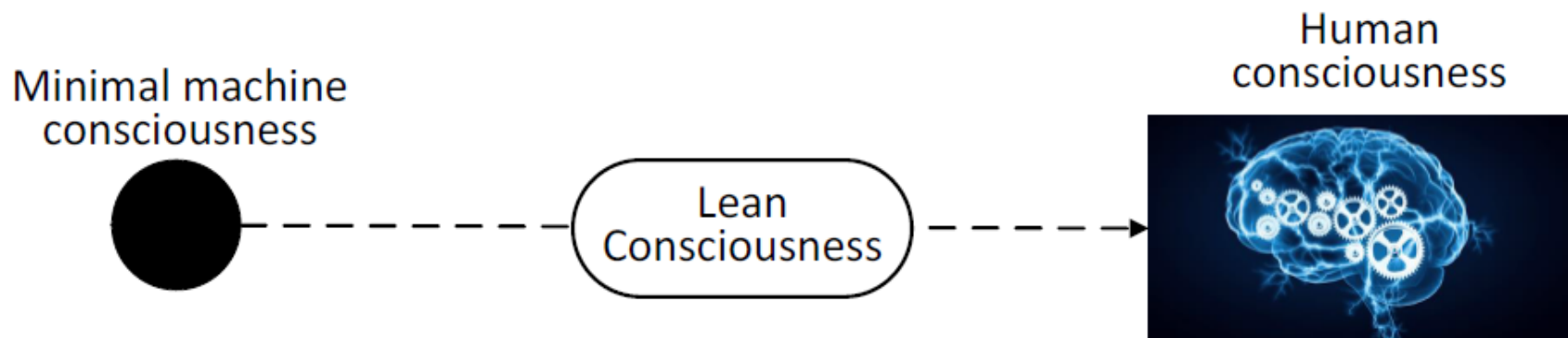
AA. Sukhobokov (2019) "Consciousness and Subconsciousness as a Means of AGI's and Narrow AI's Integration"



[岡島修正案]
 ・Knowledgeとは何か？
 KnowledgeとEncoderとの関係は

Lean Consciousness

Jiří Wiedermann(2019);“Finite State Machines with Feedback: ”An Architecture Supporting Minimal Machine Consciousness”



Lean Consciousnessは、対象分野の必要な情報に基づく意識。
(不必要で余分な知識には依拠しない。価値のある脳内状態変化を進行させる情報と機能
“facilitate a meaningful behavior“). Jiří Wiedermann

The Lean Consciousness must be limited by specific knowledge from the subject area, necessary for solving specific problems and not include unnecessary extraneous knowledge.

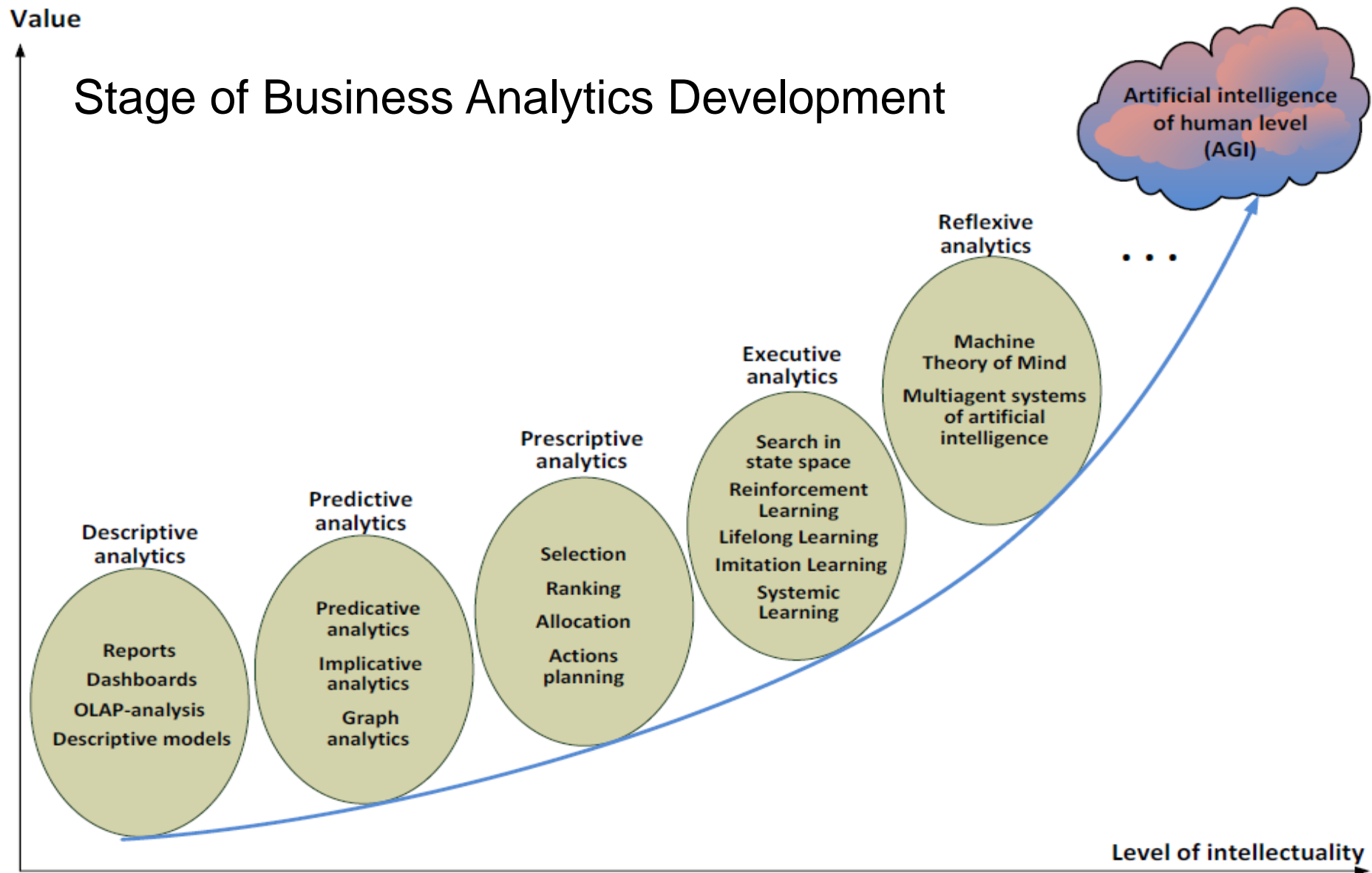
Business analytics tools for various forms of knowledge representation.

Artem A. Sukhobokov (2018); "Business analytics and AGI in corporate management systems"

Knowledge form	Complexity level 1	Complexity level 2	Complexity level 3	Complexity level 4	...	Complexity level N
Images	Using of fixed images: <ul style="list-style-type: none"> • Infographic formation • Working with dashboards • Working with diagrams, pictures and photos while reports and presentations forming 	Image Analytics: <ul style="list-style-type: none"> • Formation of image's text description • Pattern recognition (medicine, safety, quality control) • Computer vision 	Video analytics: <ul style="list-style-type: none"> • Image recognition on video • Textual description of what is happening on video • Video Matching 	Tools of virtual and augmented reality: <ul style="list-style-type: none"> • Enhancement for people working with large information content or in a rapidly changing environment 		Image processing in AGI
Descriptions	Structured Data Analytics: <ul style="list-style-type: none"> • Reports forming • Working with data in dashboards and applications • OLAP-analysis 	Text Analytics: <ul style="list-style-type: none"> • Key entities highlighting • Text subject definition • Tonality definition • Text abstracts composition 	Speech analytics: <ul style="list-style-type: none"> • Voice control of devices • Converting of speech to text • Converting of text to speech • Processing of speech streams to collect 	Conversational analytics: <ul style="list-style-type: none"> • Personal digital assistants and virtual assistants • Talking Bots 		Description processing in AGI
Numbers	Analytics of structured numerical data: <ul style="list-style-type: none"> • Working with tables • Working with numeral arrays • Reports • OLAP-analysis 	Methods of numerical coding: <ul style="list-style-type: none"> • Hash functions for steady data allocation in storage systems • Coding to improve of security and error correction • Data protection coding 	Numeral algorithms: <ul style="list-style-type: none"> • Optimization • Solving of ordinary and differential equations systems • Interpolation • Integrating • Regression analysis 	Probability analytics: <ul style="list-style-type: none"> • Application of fuzzy sets and fuzzy logic • Risk Analysis • Forecasting using probabilities • Simulation • Probability Machine Learning 		Numbers processing in AGI
Algorithms	Working with simple actions sequences: <ul style="list-style-type: none"> • Steps planning based on simple and nested action sequences • Working with chains of added quality 	Working with branching and cyclic algorithms: <ul style="list-style-type: none"> • Work with block diagrams • Working with BPMN and BPMN2 diagrams • Working with EPC diagrams • Working with information flow diagrams 	Development of actions sequence based on available samples: <ul style="list-style-type: none"> • Imitation Learning • Algorithmic metaheuristics application • Actions planning based on previously solved problems • Expert systems • Inference algorithms 	Independent actions planning based on previous experience: <ul style="list-style-type: none"> • Search in state space • Reinforcement Learning • Lifelong Learning • Systemic Learning 		Working with algorithms in AGI
Graphs	Working with hierarchies: <ul style="list-style-type: none"> • Working with organizational structures • Working with ownership structures without crossholding • Working with hierarchical assets 	Working with directed acyclic graphs: <ul style="list-style-type: none"> • Working with diagrams of electrical and gas distribution networks, water supply networks • Working with graphs of logistic networks that do not provide product return 	Working with complete graphs: <ul style="list-style-type: none"> • Working with graphs of logistics networks providing product return • Working with graphs of communication networks • Working with transport route graphs 	Working with hypergraphs and metagraphs: <ul style="list-style-type: none"> • Working with graphs of complex networks in which there are several levels of decision making 		Working with graphs in AGI

マイルストーン

Artem A. Sukhobokov.(2018); “Business Analytics and AGI in corporate management systems”.



(参)「ConnectivityとWeightsを組み合わせた統合化スキームはこの分野の聖杯といえる」

N.J. ラドクリフ(英エディンバラ大学 数学科教授)

Nicholas J. Radcliffe (1993);

”Genetic set recombination and its application to neural network topology optimization”.

- **Deep Learning と Evolutionary Computing 融合**
(後天的学習) (進化で獲得する先天的コネクトーム)

	DL Method	Evolutionary Computing
ネットワークトポロジー	人間が設定	自動化 (Neural Architecture Search)
Hyperパラメータ		
活性化関数		
学習データ	(人間が用意)	同左
重みパラメータ	自動	
	後天的学習	進化の再現

“Weight Agnostic Neural Networks”

by Adam Gaier & David Ha, in 2019;

<https://arxiv.org/abs/1906.04358>

(参) Two best paper awards for Adam Gaier

Multidisciplinary Analysis and Optimization (MDO) Student Paper Competitionで設計空間の探索と最適化のための新しい手法、Surrogate-Assisted Illumination (SAIL)を紹介し1位を獲得

受賞論文「Aerodynamic Design Exploration through Surrogate-Assisted Illumination」

その数週間後、ベルリンで開催された Genetic and Evolutionary Computation Conference (GECCO) において、複雑系部門で最優秀論文賞を受賞しています。

「サロゲート・アシスト・イルミネーションによる多様なデザインのデータ効率的な探索、最適化、モデリング」



Hochschule
Bonn-Rhein-Sieg

Departments & Institutes

Study

Application

Guidance

Computer Science »

<https://www.h-brs.de/en/inf/news/two-best-paper-awards-adam-gaier>

TWO BEST PAPER AWARDS FOR ADAM GAIER

Monday 21 August 2017

Monday 21 August 2017

Our PhD candidate Adam Gaier has had a successful summer winning two best paper awards for his current research on surrogate-assisted illumination in design space exploration.



Adam Gaier (2nd right) at the AIAA awards ceremony

目次

1. Introduction
2. Related Work
 - Architecture Search
 - Bayesian Neural Networks
 - Algorithmic Information Theory
 - Network Pruning
 - Neuroscience
3. Weight Agnostic Neural Network Search
 - Topology Search
 - Performance and Complexity
4. Experimental Results
 - Continuous Control (reinforcement learning task)
 - BipedalWalker-v2
 - CartPoleSwingUp(CarRacing-v0)
 - Classification
5. Discussion and Future Work

Abstract

- 重みパラメータとネットワーク・アーキテクチャは、どちらがどれほど重要か？
- 本研究では、ネットワークアーキテクチャを、重みパラメータを学習せずに、問題とされる強化学習タスク、教師あり学習タスクに対して最適化(最小化)する手法を考える。
- ウェイト・トレーニングを実施せずに、タスクに最善/最適/最小なニューラルネットワークアーキテクチャを見出す方法を考える。

(その生成方法を評価するために、「均一なランダム分布からサンプリングされた重みパラメータ」と「各ケースで予想されるパフォーマンス」のデータを集積し解析する)

- そのネットワークアーキテクチャにて強化学習を行った時の精度は、ランダムな重みを前提とした場合(Chance Accuracy)よりも、はるかに高い精度で画像認識するという実例を示す。

(参) Interactive version of this paper
<https://weightagnostic.github.io/>

1-1. Introduction ; Weight Agnostic Neural Networksとは？

- 人工ニューラルネットワークは、設計者が、トポロジーを選択し、ハイパーパラメータを調整し、学習アルゴリズムを使用して、重みパラメータを設定するまで何もできない。
- 生体(の幼体)は、生まれながらに、ある程度のことができる。
(両者は、「学習以前の状態」が大きく異なる)
 - トカゲ(Lizard)やヘビ(Snake)の幼体(hatchling) : 捕食者からの逃避
 - アヒルや七面鳥のヒヨコ : 泳ぐ、捕食者の認識
- 本研究では、「学習以前の状態」でも、ある程度のことのできるニューラルネットワークを開発する。(WeightにAgnostic(不可知な)のNeural Networks : WANN)
 - 重みのパラメータを最適化する以前の機能獲得
 - ネットワーク・トポロジーの問題 (各シナプスには同じ重みパラメータ値を設定)

1-2. Introduction ; Strong inductive bias for a task domain.

- ウェイト・パラメータ学習前(randomly-initialized networks)の段階から特定のタスクに対する潜在的な適合性を持つネットワーク・トポロジーの探索 (WANNs : Weight Agnostic Neural Networks)
 - CNNs : 画像認識
 - LSTMs : 時系列の認識や予測
 - Self-Attention and Capsule Networks

} Building Blocks
(強いドメインを持つ基本構成要素群)
- NAS (Neural Architecture Search) : Search ⇄ Selection
 - **ボールドウィン効果**
 - 進化的計算/進化的コンピューティング(Evolutionary Computing、EC)
 - **変異のオペレーション** (Crossover、Mutation、etc)
 - **Instatance(試行解)** を集め、**Population(試行解の集合)** を生成
 - **環境に対するFitness(適合度)の評価**
 - **Speciation(種化)** と **Building Blockの生成**

(参) ボールドウィン効果 (Baldwin effect)

<https://scrapbox.io/nobuoka-pub/ボールドウィン効果>

ジェームズ・マーク・ボールドウィン(1861-1934、米国の心理学者)の進化理論。

(参) Baldwin, Mark J. (Jun., 1896); "A New Factor in Evolution."

- 最初は、「後天的学習」によって獲得されていた形質が、次第に遺伝的に獲得され、先天的能力に転化する。一般には、次のような二つの段階からなる。

第1段階 : 学習によって、生存に適した形質を獲得した個体が次世代に多くの子孫を残す。

(その集団に、「その形質」を学習するネットワークポロジを生得的に持つよう「選択圧」が働く。)

第2段階 : 生存上有利なネットワークポロジを先天的に獲得している個体の子孫が、その集団内に優勢となる。(進化と学習の相互作用)

- 1987年に、Hinton&Nowlanが、計算機実験によってボールドウィン効果を確認

[44] G. E. Hinton and S. J. Nowlan(1987); "How learning can guide evolution."

(参) 野場康徳、他(2009)ニッチ構築と学習の相互作用に関する個体ベース進化シミュレーション
<http://www.alife.cs.is.nagoya-u.ac.jp/~ari/stuff/papers/sice08-niche.pdf>

2-1. Related Work : Architecture Search

- 1) NEAT[106]を元とするEvolutionary Computing アルゴリズムを用いる。
(NeuroEvolution of Augmenting Topologies)
[106] K. O. Stanley and R. Miikkulainen (2002);
“Evolving neural networks through augmenting topologies.”
- 2) 探索空間(Search Space)を、Task-Domainに適した基本構造群(CNN、RNN、Self-Attention、等)のBuilding Block群に狭めることで、計算時間を削減
- 3) Building Block群を探索空間とすることで、State Of The Arts(SOTA)のネットワークアーキテクチャを自動探索し生成する [65, 90, 100]
[65] L. Li et al. (2019); “Random search and reproducibility for neural architecture search”
[90] E. Real, et al. (2019); “Regularized evolution for image classifier architecture search”
[100] C. Sciuto, et al. (2019); “Evaluating the search phase of neural architecture search”
- 4) [44, 102]のように、ウエイト最適化を行わずにアーキテクチャの良否を評価
[44] G. E. Hinton and S. J. Nowlan.(1996); “How learning can guide evolution.”
[102] J. M. Smith (1987); “When learning guides evolution.

引用1

- [2] P. J. Angeline, et al.(1994);
“An evolutionary algorithm that constructs recurrent neural networks”
- [3] J. M. Baldwin (1896);
“A new factor in evolution.”
- [49] R. Jozefowicz, et al. (2015);
“An empirical exploration of recurrent network architectures”
- [65] L. Li et al. (2019);
“Random search and reproducibility for neural architecture search”
- [66] H. Liu, et al. (2018);
“Hierarchical representations for efficient architecture search”
- [74] R. Miikkulainen, et al. (2019);
“Evolving deep neural networks”
- [90] E. Real, et al. (2019);
“Regularized evolution for image classifier architecture search”
- [91] E. Real, et al. (2017);
“Largescale evolution of image classifiers”
- [100] C. Sciuto, et al. (2019);
“Evaluating the search phase of neural architecture search”
- [103] D. R. So, C. Liang, and Q. V. Le (2019);
“The evolved transformer”
- [106] K. O. Stanley and R. Miikkulainen (2002);
“Evolving neural networks through augmenting topologies.”
- [123] B. Zoph and Q. V. Le (2017);
“Neural architecture search with reinforcement learning”

(参) 進化的計算/進化的コンピューティング (Evolutionary Computing、EC) (含) 遺伝的アルゴリズム (Genetic Algorithm) & 遺伝的プログラミング (Genetic Programming)

遺伝的アルゴリズム (Genetic Algorithm、GA): 遺伝/進化のモデルを利用して探索・最適化を行う計算手法。1975年のJ.H.Holland^[1]が最初に提案。

[1] John H. Holland (1975), “Adaptation in Natural and Artificial Systems : An introductory analysis with applications to biology, control, and artificial intelligence”.

[106] K. O. Stanley and R. Miikkulainen (2002); “Evolving neural networks through augmenting topologies.”
(参考) 安藤晋(2008年9月); “遺伝的アルゴリズム”, 電子情報通信学会『知識の森』S3群4編3章

- 個体 (Instance) : 問題に対する一つの解 (試行解)
- 種の形成 (Speciation) : 保全すべき「種」として扱う (NEAT論で最も重要な概念)
- 集団 (Population) : 解 (試行解) の集合
- 個体の形質 : 決定変数 (遺伝子によって発現する個体の属性)
- 遺伝子型 (GenoType) : 決定変数を1次元に配列したデータ
- 適合度 (Fitness) : 「個体の形質」と「環境」との関係で、 $\times\times\times$ が評価する試行解の評価値
- 遺伝オペレータ (Operator) : 遺伝子型に依存する操作。評価値が既知の解を改変する。
 - 交叉オペレータ (Crossover): 複数の個体を引数として新しい個体を生成する。
 - 変異オペレータ (Mutation): 単一の個体を引数)、等。
- 複製 (Reproduction) : 集団から抽出した親個体から新個体を生成するオペレータ操作
- 淘汰 (Selection) : オペレータによって個体をより適合度の高い子個体に置き換える操作
- 進化プロセス : 複製と淘汰の反復としてモデル化される。
- 検索 (Search) : 淘汰 (Selection) による適合度の高い試行解を抽出する操作
- ビルディングブロック (Building Block) : 適合度の増大に寄与する遺伝子型の部分列
- リンケージ (linkage) : BB に含まれ、適合度への寄与において強い相関を持つ遺伝子

(参)[106-3.1] NEATの変異は、重みとネットワーク構造の両方を変えるEncodingによって表現する。

1) 接続追加の表現

ゲノム = Σ (接続遺伝子)

接続制御に関するゲノム

1	3	4	5	6
1->4	2->4	2->5 DIS	3->5	4->5

1	3	4	5	6	7
1->4	2->4	2->5 DIS	3->5	4->5	3->4

≡ 変異番号
innovation number
(historical markers)

(注) 「ゲノム」とは、遺伝情報の事であり、「遺伝子+染色体」を意味する。



新しい接続遺伝子(変異番号付き)が遺伝子の末尾に追加される。

≡ 新しい接続(重み値は“1”)

(注) “DIS”は、「無効化」の意味

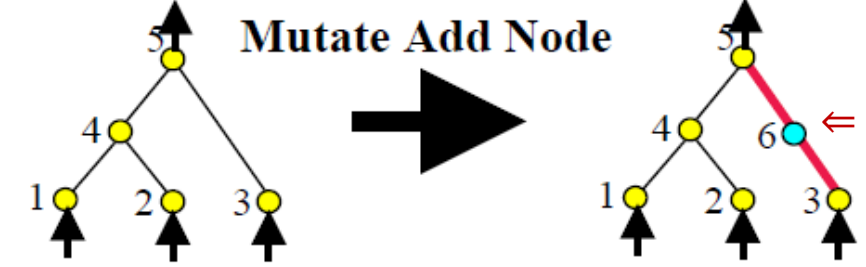
2) ノード追加の表現

接続制御に関するゲノム

1	3	4	5	6
1->4	2->4	2->5 DIS	3->5	4->5

1	3	4	5	6	8	9
1->4	2->4	2->5 DIS	3->5 DIS	4->5	3->6	6->5

≡ 変異番号



新しいノードと新しい接続(重み値は、従来と同じ値)

分割される接続遺伝子が無効化(DIS)され、2つの新しい接続遺伝子(#8と#9)がゲノムの末尾に追加される

[106-3.2] ゲノムの交差を正しく行う為に、遺伝子の起源の記録を実装

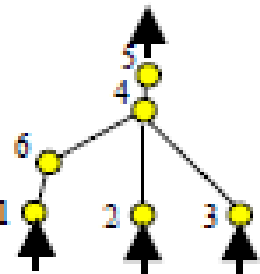
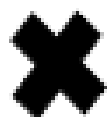
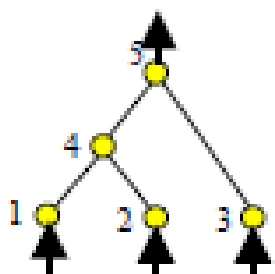
Parent1

Parent2

“DISAB”=“Disabled”

1	2	4	5	6
1->4	2->4	2->5 DISAB	3->5	4->5

1	2	3	4	6	7	8
1->4 DISAB	2->4	3->4	2->5 DISAB	4->5	1->6	6->4



どの遺伝子が双対相手かを理解するために、
遺伝子の歴史的起源が記録してある。

マッチング遺伝子は、1, 2, 4, 6。
(両ゲノムにある同じイノベーション番号を持つ遺伝子)

disjoint

Parent1

1	2
1->4	2->4

4	5	6
2->5 DISAB	3->5	4->5

Parent2

1	2	3	4	6	7	8
1->4 DISAB	2->4	3->4	2->5 DISAB	4->5	1->6	6->4

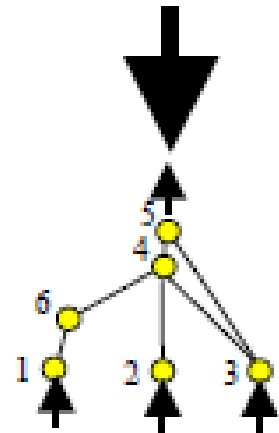
disjoint

excess excess

“disjoint” や “excess” な遺伝子は追加される。

Offspring

1	2	3	4	5	6	7	8
1->4 DISAB	2->4	3->4	2->5 DISAB	3->5	4->5	1->6	6->4



[106-3.2] Minimizing Dimensionality through Incremental Growth from Minimal Structure

- Searchの意味 : 環境への適合度が最も高いGenome組み合わせ(=ネットワーク構造)を調べる。
- NEATは隠れノードがゼロ(すなわち、すべての入力が出力に直接接続する)のネットワークの均一な母集団から始める。

=> 最小次元の空間に向かって探索を偏らせる。
- 新しい構造は、構造変異が起こるたびに段階的に導入され、適合度評価によって有用と判断された構造のみが生き残る。

[106-4.2] Verification: Evolving XORs

■ XORはNEATの構造進化能力をテストするのに適している。

・ XORは線形分離ができないので、ニューラルネットワークはこれを解くために隠れユニットを必要とするが、入力を適切なクラスに分離できるような入力の線形結合に対する関数は存在しないので、2つの入力は出力ノードだけでなく、ある隠れユニットで結合されなければならない。

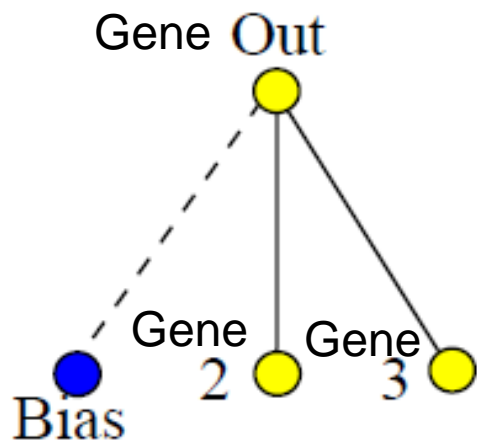
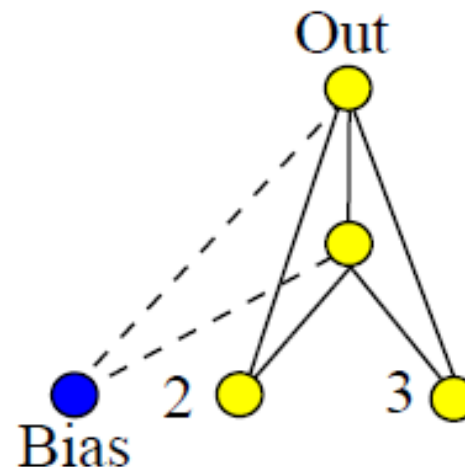


図5(a) 初期集団全体に与えられた表現型
(隠れノードがない)



(b)最適解にて、隠れノードが1つだけある場合
(隠れノードがないネットワークはXORを計算できない。解によってはバイアスの接続は必ずしも必要ではなく、他の接続はすべて必要である。)

- ・ 最適解(隠れノード1個)は100回実行のうち22回で見つかった。平均的な解は2.35個の隠れノードからなり、標準偏差は1.11個であった。
- ・ Each connection gene received a random connection weight.

2-2. Related Work : Bayesian Neural Networks

1) BNN

- BNNの重みパラメータは分布からサンプリングされ、固定値ではない。
- 分布を表現するパラメータは学習により抽出可能だが、そのパラメータ個数は、ネットワークのウエイト・パラメータ数よりも多くなってしまうかもしれない。

- [4] D. Barber and C. Bishop (1998); "Ensemble learning in bayesian neural networks. NATO ASI series",
- [5] C. M. Bishop (2006); "Pattern recognition and machine learning".
- [25] Y. Gal (2016); "Uncertainty in deep learning".
- [43] G. Hinton and D. Van Camp (1993); "Keeping neural networks simple by minimizing the description length of the weights".
- [70] D. J. MacKay (1992); "Bayesian interpolation".
- [80] R. M. Neal (2012); "Bayesian learning for neural networks, volume 118".
- [28] A. Graves (2011); "Practical variational inference for neural networks".
- [37] S. J. Hanson (1990); "Meiosis networks".
- [38] S. J. Hanson (1990); "A stochastic version of the delta rule".
- [55] D. Krueger, et al. (2017); "Bayesian hypernetworks".

2) Variance Networks ; K. Neklyudov [81]

- Variance networksは、平均値がゼロの学習後の分散変数となるパラメータを持つ分布から重みパラメータをサンプリングする。
- そのアンサンブル評価後の画像認識性能は、DL手法のものよりも向上する。
- 本研究でも、同様のアプローチを採用し、アンサンブルの性能の評価も行う。

- [81] K. Neklyudov , et al. (2019); "Variance networks: When expectation does not meet your expectations".

(参)データ量とネットワーク規模(パラメータ量)の関係に関する問題

[43] G. Hinton, et al. (1993);

“Keeping neural networks simple by minimizing the description length of the weights”

[関連1] LeCun (1989); “Back-Propagation Applied to Handwritten Zip-code Recognition”.

[関連2] Lang, Waibel and Hinton (1990); “A Time-Delay Neural Network Architecture for Isolated Word Recognition”.

- 多くの実用的な学習課題では、利用可能な学習データが少ないため、適度に複雑なモデルはデータに過剰適合し、新しいデータへの汎化がうまくいかない傾向がある。
- 学習事例の出力ベクトルよりも重みの情報を少なくする必要がある。
具体的には、「ネットワークの接続数を制限(各重みの情報が多くなりすぎないように)する。
- もしくは、接続を部分集合に分割し、その部分集合内の重みを強制的に同一にする。
- この「重みの共有」は、タスクの自然な対称性の分析にて限定されている場合、ECの手法は非常に効果的に作用する。

2-3. Related Work : Algorithmic Information Theory (AIT)

- 良いモデルは、そのデータを圧縮するモデル。
良いモデルを用いると、モデルの記述コスト(計算)コストは最小となる。
(これは、アルゴリズム情報理論でのMinimal Description Length (MDL)と同じ)
 - (注1) オッカムの剃刀：「必要なしに多くのものを定立してはならない」という原則。
 - (注2) MDL：「必要なしに多くのものを定立してはならない」という原則。
オブジェクトのコルモゴロフ複雑さは、そのオブジェクトを計算するプログラムの最小長。
- 従って、学習データ量に合わせて、ネットワーク・サイズを増減させなくてはならない。
- ネットワークの単純化法としては、以下が知られる。
 - ① ノイズを付加することによる情報量の削減[43]
 - ② ソフト・ウェイト・シェアリング[82]
 - ③ ネットワークの重みパラメータが作る空間の単純化[98]
 - ④ 深層化ネットワークでの手法[7]
 - ⑤ アプリケーション[63,111]
- 本論では、MDLに基づき、ネットワーク全体にウエイト値を同じとし、かつそのウエイト値は、固定分布からサンプリングされた(選ばれた)ランダム変数(適当な値)とする。
(例えば、-2から+2までの間で、0.5刻みで。)

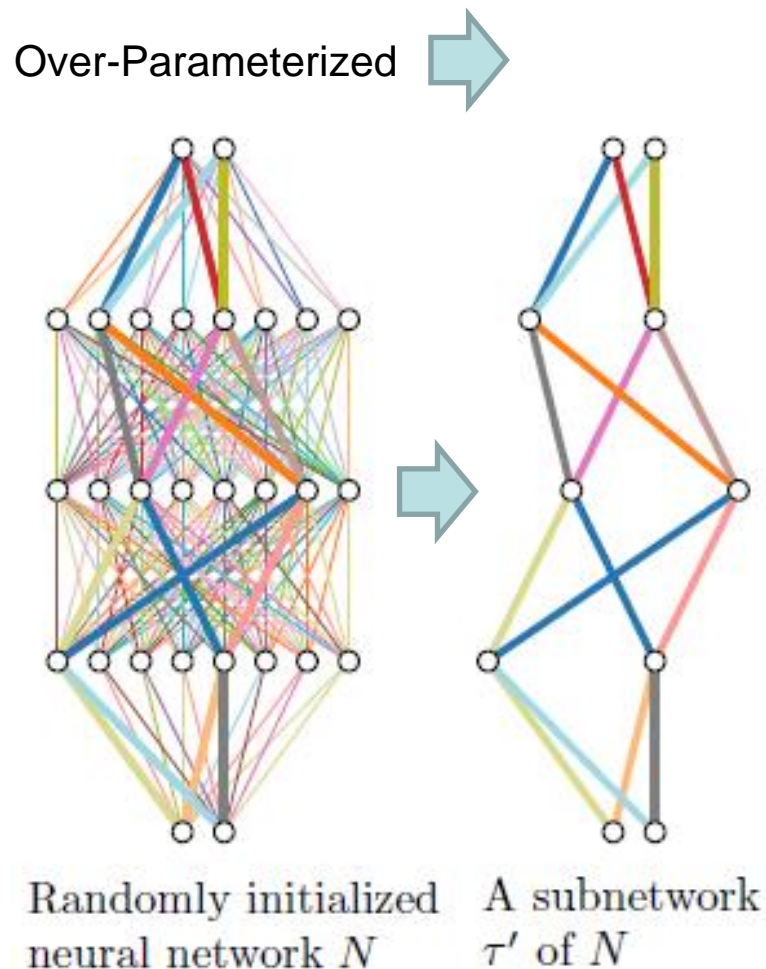
Algorithmic Information Theory (AIT)に関する引例

- [7] L. Blier and Y. Ollivier.(2018); “The description length of deep learning models”.
- [32] P. D. Grünwald (2007); “The minimum description length principle”.
- [43] G. Hinton, et al. (1993); “Keeping neural networks simple by minimizing the description length of the weights”
- [52] A. N. Kolmogorov (1965); “Three approaches to the quantitative definition of information”.
- [63] C. Li, et al. (2018); “Measuring the intrinsic dimension of objective landscapes”.
- [82] S. J. Nowlan and G. E. Hinton.(1992); “Simplifying neural networks by soft weight-sharing”.
- [93] J. Rissanen (1978); “Modeling by shortest data description”.
- [94] J. Rissanen (2014); “Information and complexity in statistical modeling”.
- [98] J. Schmidhuber (1997);
“Discovering neural nets with low kolmogorov complexity and high generalization capability”.
- [104] R. J. Solomonoff (1964); “A formal theory of inductive inference. Part I”.
- [111] A. Trask, et al. (2018); “Neural arithmetic logic units”.

2-4. Related Work : Network-Pruning

- ネットワークプルーニング(ネットワーク剪定)は、本論の逆のアプローチ(本論では、最小ネットワークからBuilding-Blockするが、Pruningでは、Full-Networkから刈り込む。プルーニングでは、Full-Networkをトレーニングするため計算時間が長い。)
- プルーニング後に、ウェイト値を初期値に戻し、スクラッチから訓練し直すと元のネットワークよりも高いテスト精度を達成することがある[22,61]。
- プルーニング後のネットワークのウェイト値を、ランダムに初期化しても、偶然よりもはるかに良い画像分類精度を達成できることがある[122]。

- イノベーションの保護が無いと、大きな構造的付加を行ったネットワークは再現されずらい。
対策は、母集団を種化(Spaciation)すること。
種化されると、最小限のスタートが可能になる。
- 最小限の状態から始めて、何世代にもわたって解を複雑化していくシステムは、自然進化とのアナロジーがある。



Network-Pruningの引例

- [22] J. Frankle and M. Carbin (2019); “The lottery ticket hypothesis: Finding sparse, trainable neural networks”.
- [33] Y. Guo, A. Yao, and Y. Chen (2016); “Dynamic network surgery for efficient dnns”.
- [36] S. Han, et al. (2015); “Learning both weights and connections for efficient neural network”.
- [40] B. Hassibi and D. G. Stork (1993); “Second order derivatives for network pruning: Optimal brain surgeon”.
- [59] Y. LeCun, J. S. Denker, and S. A. Solla (1990); “Optimal brain damage”.
- [61] N. Lee, et al. (2019); “Single-shot network pruning based on connection sensitivity”.
- [64] H. Li, et al. (2017); “Pruning filters for efficient convnets”.
- [68] Z. Liu, et al. (2019); “Rethinking the value of network pruning”.
- [69] J.-H. Luo, et al. (2017); “A filter level pruning method for deep neural network compression”.
- [71] A. Mallya, et al. (2018); “Piggyback: Adapting a single network to multiple tasks by learning to mask weights.”
- [77] P. Molchanov, et al. (2017); “Pruning convolutional neural networks for resource efficient inference”.
- [122] H. Zhou, , et al.(2019); “Deconstructing lottery tickets: Zeros, signs, and the supermask.”

2-4. Related Work : Neuroscience

- コネクトームは、グラフ[12,42,114]と見なすことができ、グラフ理論、ネットワーク科学およびコンピュータシミュレーションにて分析できる。
- [6] J. E. Black, et al. (1990); “Learning causes synaptogenesis, whereas motor activity causes angiogenesis, in cerebellar cortex of adult rats”.
- [11] J. T. Bruer (1999); “Neural connections: Some you use, some you lose”.
- [12] E. Bullmore and O. Sporns (2009); “Complex brain networks: graph theoretical analysis of structural and functional systems”.
- [18] E. Dayan and L. G. Cohen (2011); “Neuroplasticity subserving motor skill learning”.
- [20] K. Eichler , et al. (2017); “The complete connectome of a learning and memory centre in an insect brain”.
- [42] Y. He and A. Evans (2010); “Graph theoretical modeling of brain connectivity”.
- [46] P. R. Huttenlocher (1990); “Morphometric study of human cerebral cortex development”.
- [51] J. A. Kleim , et al. (2002); “Motor learning-dependent synaptogenesis is localized to functionally reorganized motor cortex”.
- [101] S. Seung (2012).; “Connectome: How the brain’s wiring makes us who we are”.
- [105] O. Sporns , et al. (2005); “The human connectome: a structural description of the human brain”.
- [108] S. Takemura , et al. (2017); “A connectome of a learning and memory center in the adult drosophila brain”.
- [110] A. L. Tierney, et al. (2009); “III. Brain development and the role of experience in the early years”.
- [114] M. P. Van Den Heuvel and O. Sporns (2011); “Rich-club organization of the human connectome”.
- [115] L. R. Varshney , et al. (2011); “Structural properties of the caenorhabditis elegans neuronal network”.
- [117] J. G. White , et al. (1986); “The structure of the nervous system of the nematode caenorhabditis elegans”.

3. Weight Agnostic Neural Network Search (WANNs)

- 探索空間 : 大きさは、 $C > B > A$ 。

- A) ネットワークトポロジが作る空間 …… ウェイト値を特定値としては探索
- B) 重みパラメータ値が作る空間 …… DLの手法 (WANNsの2nd-Step)
- C) A+B

- WANNsでは、以下の手順で、まず A を探索する。

- 1) 全体ネットワークを構成する「小トポロジの初期ネットワーク母集団」を作成

↓
ウェイト値を特定値に設定

← ウェイト値更新

- 2) 各ネットワークの性能を評価(入力データセット)

- 3) 各ネットワークを累積報酬にてランク分け[76]

- 4) ネットワークトポロジの母集団を再構成 /

↓ ランクが最高となるよう全体ネットワークを更新
進化完了後の全体ネットワークを出力

NASの手法で、
ネットワークを進化
させる。
(トポロジ探索)

[76] B. L. Miller and D. E. Goldberg (1995); “Genetic algorithms, tournament selection, and the effects of noise”.

3-1. トポロジ探索 (Topology Search); NEAT

左:小トポロジの初期ネットワーク母集団のネットワーク例
(入力と出力を部分的に接続しているだけです。)

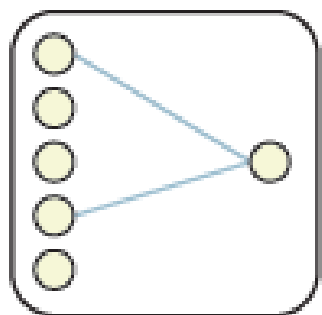
中央:ネットワークは3つの方法のうちの一つで変更される。

- 挿入ノード :既存の接続を分割して新しいノードを挿入
- 接続追加 :2つの以前に接続されていないノード間に接続を追加
- アクティベーション変更 :隠しノードの活性化関数の変更

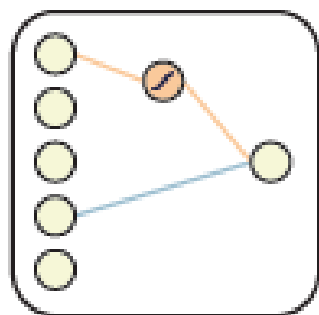
右:活性化関数のオプション

線形、ステップ、SIN、コサイン、ガウス、TANH、SIGMOID、絶対値、
反転(負の線形)、RELU

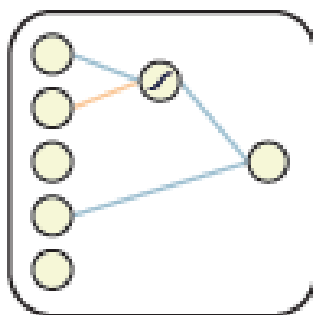
Minimal Network



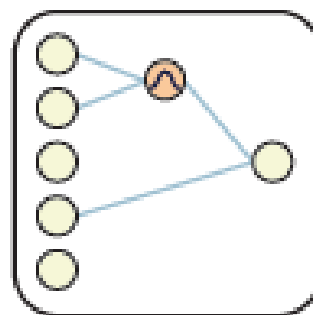
Insert Node



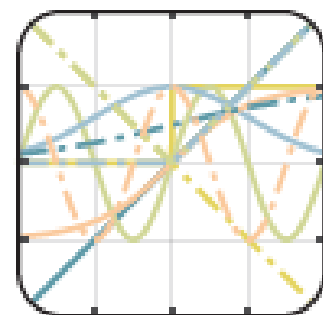
Add Connection



Change Activation



Node Activations



[106] K. O. Stanley and R. Miikkulainen (2002); "Evolving neural networks through augmenting topologies."

3-2. Performance and Complexity

- -2 から +2 の範囲のウエイト値で、性能は最も変動した。
- 性能値の算出時には、全てのロールアウトでの累積報酬を平均化して評価した。
- 最小限の説明長で説明できるネットワークを重視した[32,93,94]。
(性能が同等の時には、以下の基準での最もシンプルなものをモジュール化した)
 - ① すべての重み値
 - ② 単一の最良の重み付け値の最大パフォーマンス
 - ③ ネットワーク内の接続数
- 探索を多目的最適化問題として定式化した[53,79]。
(ある目的関数の値を改善する時には、他の目的関数の値を改悪し、バランスさせる)
- Connection Cost Technique[15]を採用

[15] J. Clune, J.-B. Mouret, and H. Lipson (2013); The evolutionary origins of modularity.

[19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan (2002); “A fast and elitist multiobjective genetic algorithm: Nsga-ii”.

[32] P. D. Grünwald (2007); “The minimum description length principle”.

[53] A. Konak, D. W. Coit, and A. E. Smith (2006); “Multi-objective optimization using genetic algorithms: A tutorial”.

[79] J.-B. Mouret (2011); “Novelty-based multi objectivization”.

[93] J. Rissanen (1978) ; “Modeling by shortest data description”.

[94] J. Rissanen (2007); “Information and complexity in statistical modeling”.

[104] R. J. Solomonoff (1964); “A formal theory of inductive inference. part I”.

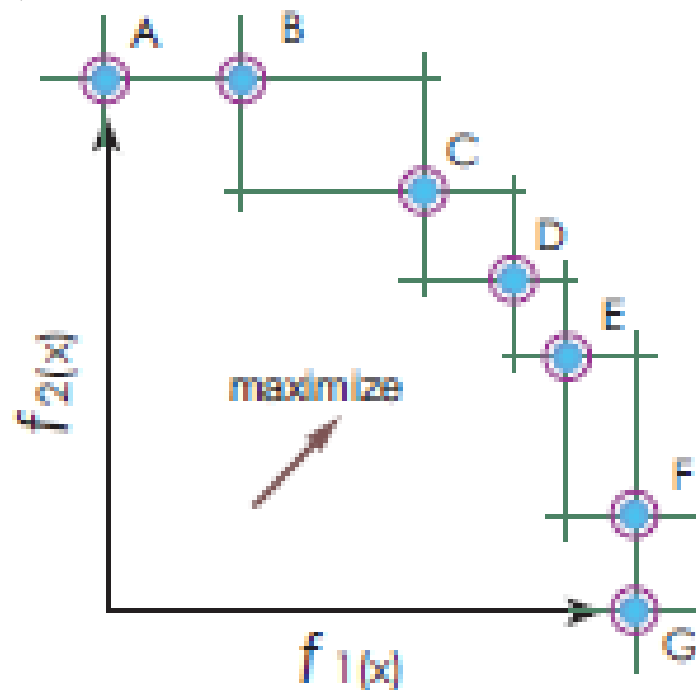
[106] K. O. Stanley and R. Miikkulainen (2001); “Evolving neural networks through augmenting topologies.”

(参)多目的最適化問題：Connection Cost Technique[15]

多目的最適化とは「複数個の互いに競合する目的関数を、与えられた制約条件の中で最大化(あるいは最小化)する問題」と定義される。

目的関数が互いに競合しあっているため、与えられた複数の目的関数に対して完全最適解を求めることはできない。そのため、多目的最適化では「ある目的関数の値を改善するためには、少なくとも他の目的関数の値を改悪せざるを得ないような解」を求めていく。

多目的最適化問題では、このような解の集合をパレート最適解(Pareto optimal solution)と呼ぶ²⁾。



Pareto Optimal Solutions

A, C, D, E, G : Pareto Optimal Solutions

B, F : Weak Pareto Optimal Solutions

weight parameters

A : $\omega_{f_1} = 0.0, \omega_{f_2} = 1.0$

G : $\omega_{f_1} = 1.0, \omega_{f_2} = 0.0$

4-1. Results ; Continuous Control Tasks #1 ; 強化学習 (WANN を3つの連続制御タスクで評価)

- 評価タスクは、① SwingUp(カートとポールのシステム)、② Biped(2本足のエージェント)、③ CarRacing (カーレース) の連続制御タスク
- 累積報酬値比較** ; 「本論のWANNの最良版」 vs 「典型的なネットワーク設計^[34, 35]版」
 - ランダムウェイト : {-2.0 to +2.0}の範囲で、個々のウェイト値をランダムに決めた場合
 - ランダムな共有重み : {-2.0 to +2.0}の範囲で、ウェイト値を共通値とした場合
 - 調整済み共有重み : {-2.0 to +2.0}の範囲でのウェイト共通値で、最も性能が高い場合
 - 調整済みの重み : 母集団を使った強化学習にて調整した重みの場合^[118]

WANNの場合は、ウェイト調整なくても報酬値は高い

両者は同等

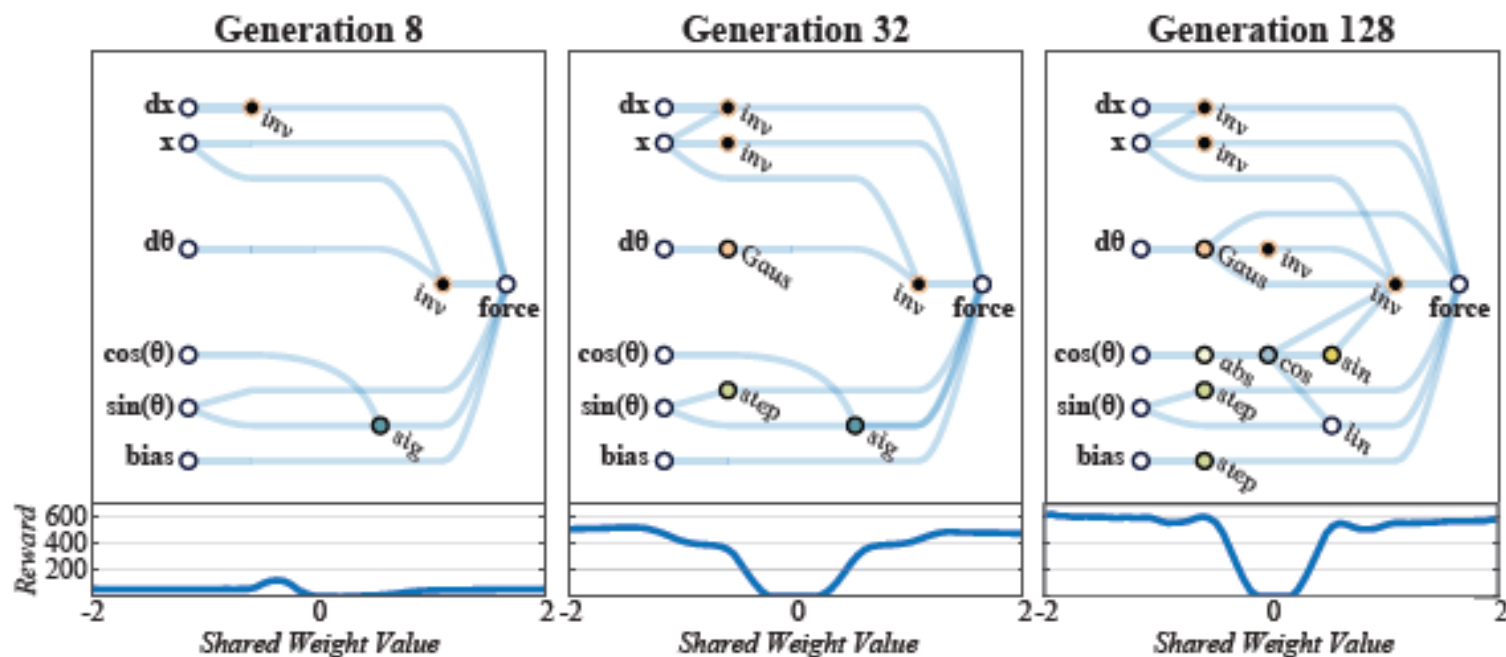
Task	Random Weights	Random Shared Weight	Tuned Shared Weight	Tuned Weights
Swing Up				
WANN	57 ± 121	515 ± 58	723 ± 16	932 ± 6
Fixed Topology	21 ± 43	7 ± 2	8 ± 1	918 ± 7
Biped				
WANN	-46 ± 54	51 ± 108	261 ± 58	332 ± 1
Fixed Topology	-129 ± 28	-107 ± 12	-35 ± 23	347 ± 1 [34]
CarRacing				
WANN	-69 ± 31	375 ± 177	608 ± 161	893 ± 74
Fixed Topology	-82 ± 13	-85 ± 27	-37 ± 36	906 ± 21 [35]

4-1. Results ; Continuous Control Tasks #2 ; 強化学習 (Development of WANNs Over Time)

- ・ 学習させた動作は、ネットワークのアーキテクチャの中に符号化される。
WANNは訓練しなくてもある程度の性能を発揮するが、そのように潜在能力があるからといって、重みを訓練した時の到達レベルが下がる訳ではない。

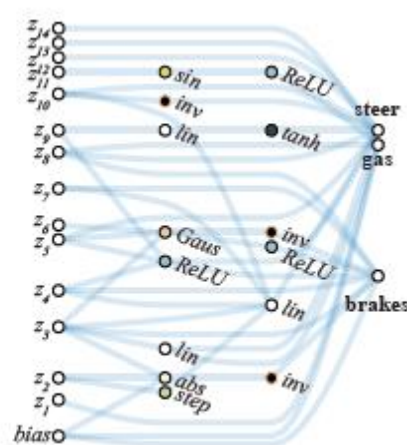
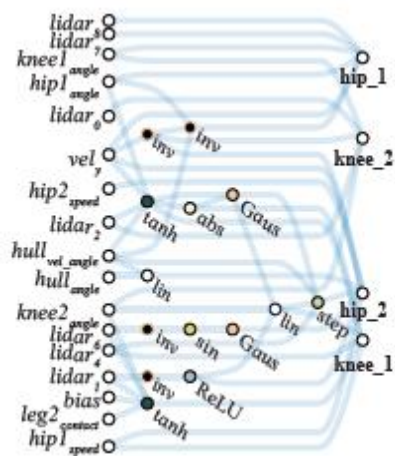
「① SwingUp(カートとポールのシステム)」の進化

- > 第8世代目 : ほぼすべての重みでパフォーマンスが低下する段階
- > 第32世代目 : カートの位置とポールの速度の関係が確立される。
(センタリングとスイングアップ間に両立できない問題が残った)
- > 第128世代目 : 複雑さが追加して、昇降ポールのバランス挙動を洗練させた。



4-2. Results ; Continuous Control Tasks #2 BipedalとCarRacingのWANNコントローラ

- WANNコントローラのシンプルさとモジュール性に注目すべきだ。
- WANNアーキテクチャは、個々の重みを学習することなくタスクを解決するだけでなく、一般的に使用されているトポロジー(SOTAベースライン[34]では、2804の接続がある)よりも1桁少ない210接続となっている。
(DLにおいて、わずかな性能向上のために、多くの複雑さが付加されることが多い)
- 入力間の関係を符号化するWANNは、RLタスクによく適している。
低次元の入力と内部状態や環境との相互作用を組み合わせることで、反動的で適応的なコントローラの発見が可能になる。



Results ; Continuous Control Tasks の引例

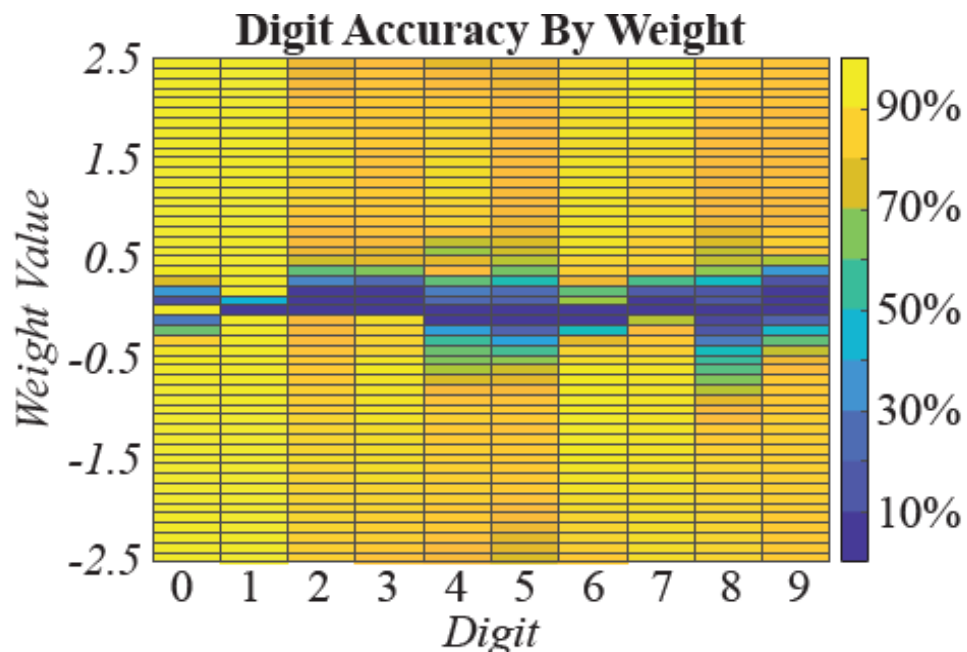
- [10] G. Brockman, et al. (2016); “Openai gym”.
- [34] D. Ha (2018); “Reinforcement learning for improving agent design”.
- [35] D. Ha and J. Schmidhuber (2018); “Recurrent world models facilitate policy evolution”.
- [50] D. P. Kingma, et al. (2014); “Auto-encoding variational bayes. In International Conference on Learning”.
- [59] Y. LeCun, J. S. Denker, and S. A. Solla (1990); “Optimal brain damage”.
- [92] D. J. Rezende, et al. (2014); “Stochastic backpropagation and approximate inference in deep generative models”.
- [118] R. J. Williams (1992); “Simple statistical gradient-following algorithms for connectionist reinforcement learning”.

4-3. Results ; Classification #1

- この高次元分類課題において、重みの値を全て同じとしたWANNは、MNISTの数字をある程度分類することができるが、人間が手で最適化したCNNを勾配降下法で学習させた単層ニューラルネットワークに劣る。(下表)
- 全ての「数字」において、単一の重み値で精度が向上することはない。(右図)

WANN	Test Accuracy
Random Weight	82.0% ± 18.7%
Ensemble Weights	91.6%
Tuned Weight	91.9%
Trained Weights	94.2%

ANN	Test Accuracy
Linear Regression	91.6% [62]
Two-Layer CNN	99.3% [15]



[14] F. Chollet (2015); “Keras”.

[56] Y. LeCun (1998); “The mnist database of handwritten digits”.

[58] Y. LeCun, et al.(1998); “Gradient-based learning applied to document recognition”.

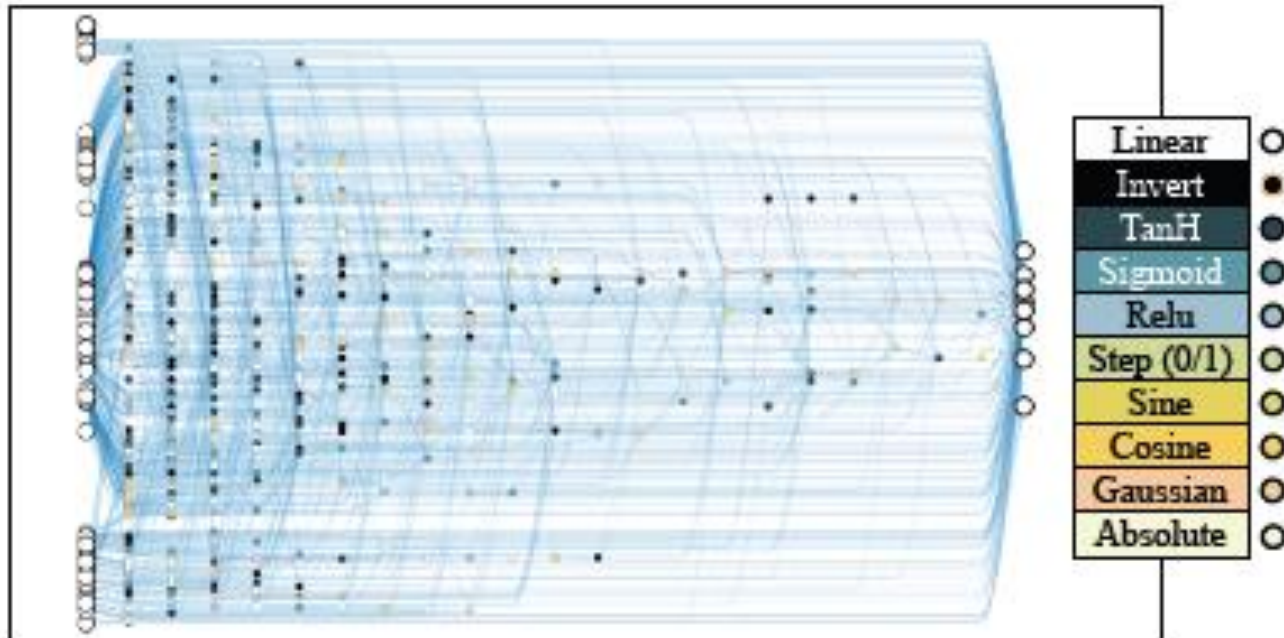
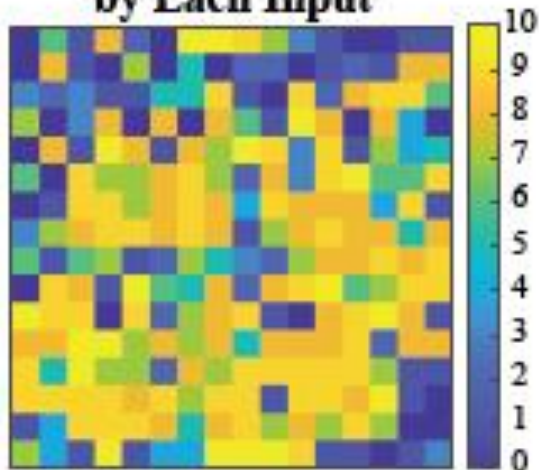
[96] S. Sabour, N. Frosst, and G. E. Hinton(2017); “Dynamic routing between capsules”.

4-4. Results ; Classification #2

- WANNを用いたアンサンブル手法に、可能性がある。

WANNに様々な重み値を与えてインスタンス化し、ネットワークの集合を作成する。

of Classes Used
by Each Input



5 Discussion and Future Work

- 本研究では、特定の推論動作に適正を持つ (strong inductive biases) ニューラルネットワーク・トポロジーの探索方法を提示した。「適正を持つ」といのは、「”intrinsic motivation“に基づいて、”Intrinsic bias”を持つ [84, 86, 97]ビルディングブロックである」ことを意味する。
これは、遺伝子の働きによる先天的な段階のネットワーク形成と対応させうる。
- 探索されたネットワークは、重みパラメータの変動に対して鈍感 (weight tolerant) であるが、更に重みパラメータを調節すると、更に精度向上 (fine-tune) させることができるので、Few-shot learning [21] や continual learning [85] を期待できる。これは、生まれた後の終わりのない環境 (open-ended environment [62]) 下で、報酬 (curiosity reward signal) を受けて後天的学習と対応させうる。
- 進化によるネットワークトポロジーの先天的最適化と、学習による精度向上 (fine-tune) は、Baldwin効果 [3] と対応しうる [1, 44, 102, 120]。

[29] A. Graves (2016); “Adaptive computation time for recurrent neural networks.”

[30] A. Graves, G. Wayne, and I. Danihelka (2014); “Neural Turing machines.”

[48] E. Jang, S. Gu, and B. Poole (2017); “Categorical reparameterization with gumbel-softmax.”

[67] H. Liu, et al. (2019); “Differentiable architecture search. In International Conference”

[71] A. Mallya, et al. (2018); “Piggyback: Adapting a single network to multiple tasks by learning to mask weights.”

[120] A. M. Zador (2019); “A critique of pure learning and what artificial neural networks can learn from animal brains.”

[122] H. Zhou, et al. (2019); Deconstructing lottery tickets: Zeros, signs, and the supermask.”

5 Discussion and Future Work : まとめ

- **Baldwin効果に注目し、「AIの学習」を「進化論」と結び付けて考えている。**
(進化によりネットワークアーキテクチャを探索。生後の学習でウェイトを調整)
[1] D. Ackley and M. Littman.(1991); “Interactions between learning and evolution.”
[3] J. M. Baldwin.(1896); “A new factor in evolution.”
[44] G. E. Hinton and S. J. Nowlan.(1996); “How learning can guide evolution.”
[102] J. M. Smith.(1987); “When learning guides evolution.”
- **この研究は、ノード間のコネクションを追加することで、WANNアーキテクチャを抽出したが、逆に、Network-Pruningという手法も有力。**
(マスク情報を抽出するという手法にて、プルーニング・ネットワークを生成した場合の画像認識精度は、本研究と同程度に良い。)
[71] A. Mallya, et al.(2018); “Piggyback: Adapting a single network to multiple tasks by learning to mask weights.”
[122] H. Zhou, et al.(2019); “Deconstructing lottery tickets: Zeros, signs, and the supermask.”
- **特定の種類の論理を実装するのに適正を持つネットワークアーキテクチャの生成**
(ウェイトを均一とした状態で、最も性能/精度が高くなるアーキテクチャを探索)
- **抽出された後のネットワークは、Weight Tolerantだが、ウェイトのFine-Tune後は、Few-Shot Learningや、Continual Learningと対応させうる。**
[21] C. Finn, et al. (2017); “Model-agnostic meta-learning for fast adaptation of deep networks.”
[85] G. I. Parisi, et al.(2018);“Continual lifelong learning with neural networks: A review. “

5 Discussion and Future Work : 今後の課題

- 「進化に基づく選択圧力」は、何でも学べる汎用学習アルゴリズムだ。[120]。
「進化による先天的な獲得」と「(専門化)学習」の両方に跨る理論が必要だ。
[120] A. M. Zador.(2019);
“A critique of pure learning and what artificial neural networks can learn from animal brains.”
- アーキテクチャ検索をより効率的にするために、[67]のような微分可能な定式化を導入する必要がある。
[67] H. Liu, et al.(2019) ; “Differentiable architecture search. In International Conference on Learning Representations (ICLR)”
- 勾配降下法は、ある意味ではネックとなっている。
 - ・ 離散的成分に基づいたトレーニング方法、
 - ・ 勾配降下法と両立する新たな計算手法、
 - ・ 勾配計算を必要としない新しいアーキテクチャ、等を研究する必要がある。
[30] A. Graves, et al.(2014); “Neural turing machines”
[48] E. Jang, et al.(2017); “Categorical reparameterization with gumbel-softmax.”
[29] A. Graves. (2016) ; “Adaptive computation time for recurrent neural networks.”
- 本研究の手法は、画像認識精度の良いCNNを抽出しない。
(CNNも抽出するの進化アルゴリズムがあるのかもしれない)