報酬最大化原理にもとづく脳型ＡＧＩアーキテクチャの構想

Research plan of brain-based AGI architecture based on the reward maximization principle

一杉裕志 [1*]
Yuuji Ichisugi[1]

[1] 産業技術総合研究所 人工知能研究センター
[1] National Institute of Advanced Industrial Science and Technology (AIST), AIRC

Abstract: We describe our vision of a demonstration of a brain-based AGI architecture that we aim to realize. This architecture is based on the reward maximization principle as one of the key design guidelines, and has three core technologies: recursive reinforcement learning, program synthesis, and generative model. In order to make the early realization of a convincing demonstration realistic, we focus on simplifying the environment in which the agents operate as much as possible while retaining the essential features of the real world.

1 はじめに

汎用人工知能 (AGI) の実現を加速する 1 つの方法は、AGI の実現可能性を多くの人々が一目で確信できるようなデモを、小規模なものでもよいから実現することだろう。

実世界で動作する十分に性能の高い強化学習エージェントは AGI であると見なせる。ヒトの大脳もおそらく強化学習を用いて報酬を最大化する器官である。本稿では脳を階層型強化学習を用いたプログラム合成システムと見なし、その機構を計算機上で再現することを目指す我々のこれまでの取り組みと、それを拡張して実現を目指す脳型ＡＧＩアーキテクチャのデモの構想について述べる。このアーキテクチャは報酬最大化原理を重要な設計指針の 1 つとし、再帰的強化学習、プログラム合成、生成モデルの 3 つを中核技術としている。説得力のあるデモの早期実現を現実的なものにするために、エージェントが動く環境を、実世界の本質的特徴を残しつつ極力単純化することを重視している。

本稿は以下のような構成になっている。まず 2 節で報酬最大化原理について述べる。3 節では開発中の脳型ＡＧＩアーキテクチャの技術的特徴、4 節で今後解決すべき重要な課題、5 節では個々の構成要素の将来の性能向上の可能性について述べる。最後に 6 節でまとめについて述べる。

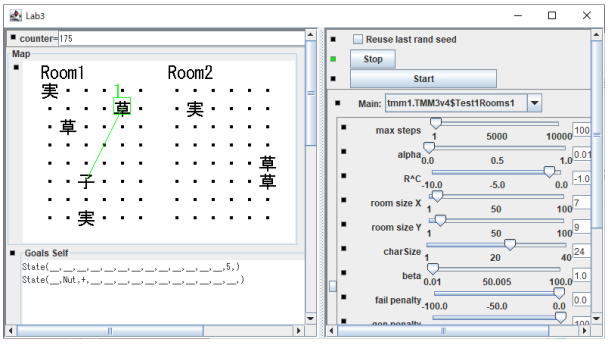

図 1: 現在開発中の脳型ＡＧＩアーキテクチャ開発フレームワークの画面。実世界の特徴を残しつつ極力単純化された環境の中で、エージェントは他者と会話をしながら動き回る。詳細は本文の 3.5 節および 4.2 節参照。

*連絡先：産業技術総合研究所
茨城県つくば市梅園 1 − 1 − 1 中央第 1
E-mail: y-ichisugi@aist.go.jp

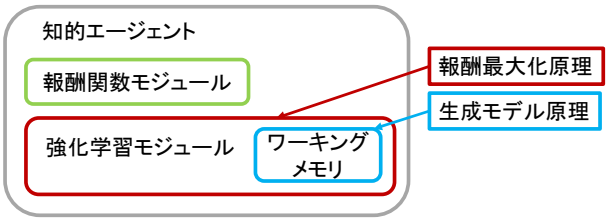

図 2: 我々が開発を目指す脳型ＡＧＩアーキテクチャの概念図とその設計原理。アーキテクチャは報酬関数モジュールと強化学習モジュールから構成される。強化学習モジュールは報酬最大化原理（2 節）にしたがって設計され、その内部には生成モデル原理（3.4 節）に従うワーキングメモリがある。報酬関数モジュールの設計目標は、生物の場合は生存して子孫を残すことであり、AGI の場合は目的に合わせてエンジニアが自由に設定する。

2　報酬最大化原理

Silver ら [1] は知識、学習、知覚、社会的知性、言語、汎化、模倣、一般知能などの知性に関する能力はいずれも報酬最大化に役立つものとして理解できるという仮説を主張している。また同時に、十分に強力な強化学習エージェントから知能が生まれると予想し、それが AGI の理解・実現の直接的な道筋を提供すると述べている。

我々も似た考えを持っており、実際に階層型強化学習 RGoal を中核技術の 1 つとした脳型 AGI アーキテクチャの実現を目指した研究を進めている。AGI エージェントの目的を報酬最大化であると仮定した設計指針を**報酬最大化原理**と呼ぶことにする[1]。我々のこれまでの経験では、報酬最大化原理はアーキテクチャ全体の見通しをよくする上、個々のパーツの設計の妥当性の議論もしやすくするという利点がある。

我々は脳の主要な動作目的もまた報酬最大化であると考えている。生物にとっての報酬最大化原理の利点の 1 つは、外界と身体の状態の価値を評価する報酬関数モジュールと、報酬を最大化する強化学習モジュールを分割することで、それぞれを独立に進化させることが可能であるという点であろう。例えば栄養のある果物が豊富な土地にたどり着いた個体はその果物の味覚的特徴が高い報酬を与えるように報酬関数モジュールを進化させるだけで、その果物を他の個体よりも摂取しやすくなり、子孫を残しやすくなる。強化学習モジュールの方は汎用性が高いので、報酬関数が変化してもそのまま動作することができる。この利点は脳型 AGI 技術を実社会に応用する際にも役に立つ。機械の動作目的をカスタマイズするには報酬関数モジュールだけを設計しなおせばよく、強化学習モジュールの方は修正しなくても機能する。

報酬関数の設計は報酬最大原理の対象の外にある点には注意が必要である（図 2）。生物の場合は、子孫を確実に残すという目的を達成する方向に報酬関数が進化する。ＡＧＩの場合はその応用目的に応じてエンジニアが報酬関数を設計する必要がある。

なお、強化学習には簡単な問題に対しては「目的さえ与えれば試行錯誤により自動的に解が得られる」という特徴があるが、**ＡＧＩに関してはこの特徴はほとんど役立たない**と筆者は考えている。実世界では状態行動空間があまりに広く、素朴な試行錯誤では手に負えないのである。したがって「実世界で動作する十分に強力な強化学習アージェントは AGI である」という主張が正しいとしても、どうやれば「十分に強力」にできるかが問題となる。これに関しては上記の Silver らの論文では具体的に書かれていない。この問題に対するこれまでの我々の取り組みと、今後取り組むべき重要な課題について、以下の節で述べていく。

3　開発中の脳型ＡＧＩアーキテクチャの技術的特徴

3.1　アーキテクチャの概要

我々は報酬最大化原理に基づき、神経科学的知見・認知科学的知見を参考にして脳型ＡＧＩアーキテクチャの実現を目指している。

その出発点として、我々はこれまでヒトの前頭前野周辺の情報処理機構のモデルの構築に取り組んだ。前頭前野はヒトの思考や創造性を担う脳の最高中枢であると考えられている[2]。我々は、前頭前野は強化学習を用いてプログラム合成するシステムであるととらえている [2][3][4][5]。前頭前野モデルの中核には、サブルーチンの再帰呼び出しが可能な強化学習アーキテクチャ RGoal [4][5] を用いる。プログラムは、ワーキングメモリの参照・更新を行いながら、環境の隠れた状態に関する推論や環境への働きかけを行う。ワーキングメモリは生成モデル（センサー入力の確率分布を表現するモデル）の形をしており、将来は筆者が提案する大脳皮質モデル BESOM[6] で実装することを想定している。

以下の節ではこのアーキテクチャの中核技術である再帰的強化学習、プログラム合成、生成モデルの 3 つについてより詳しく述べていく。

[1] より正確には累積報酬期待値最大化もしくは単位時間あたりの報酬期待値最大化が AGI エージェントの目的である。

[2] 参考：脳科学辞典 https://bsd.neuroinf.jp/wiki/前頭前野

3.2 再帰的強化学習

RGoal は再帰的なサブルーチン呼び出しを可能とする階層型強化学習（これを再帰的強化学習と呼ぶことにする）のアーキテクチャである。

先行研究の 1 つ MAXQ [7] は、多層（ただし深さは固定）の階層型強化学習アーキテクチャであり、以下の 3 つの特徴により学習を効率化している。

1. Subtask sharing: マルチタスク環境での学習において、サブルーチンをタスク間で共有することにより、学習を速くする。
2. Temporal abstraction: 複雑なタスクの学習において、学習済みのサブルーチンの組み合わせのみに試行錯誤の範囲を限定することで、学習を速くする。
3. State abstraction: サブルーチンの内部の学習において、サブタスクの実行に無関係な情報を無視し状態を抽象化することで、学習を速くする。

RGoal では 1. は価値関数の分解と共有、 2. は思考モードと呼ぶ機構により実現する [4][5]。 3. はパターンマッチの機構を利用したテーブル圧縮手法 [2][8] と注意の機構 [9] で実現しようとしている。

階層型強化学習では適切にサブゴールが与えられないと、学習の効率はよくならない。経験のみから適切なサブゴールを学習することは可能だが、サブゴールの学習まで含めた全体の学習効率がよくなるとは限らない。この問題は、ヒトの場合は言語や模倣による他者からの知識獲得と、身体に作り込まれたヒューリスティックスにより解決されていると仮定し、それを 4.4 節と 4.6 節の機構で解決を図ろうとしている。

3.3 プログラム合成

プログラム合成の目的は、少ない入出力例のみから未知の入力に対してもできるだけ正しい出力を行えるプログラムを見つけることである。AGI エージェントは、センサーからの入力を受け取り報酬を最大化する運動を出力するプログラムを合成するシステムと見なすこともできる。プログラム合成を用いた AGI の先行研究としては AIXI[10], MagicHaskeller[11], DCN[12], DreamCoder[13] などがある。

我々が RGoal を用いて構築中のプログラム合成システムでは、強化学習における状態行動対をパターンを用いて圧縮・抽象化した行動ルールの集合としてプログラムを表現する [2][8]。行動ルールは強化学習によって学習した価値に応じた確率で選択され、ワーキングメモリの参照・更新や環境に対する行動出力を行う。このアーキテクチャはプロダクションシステムの基本構造を踏襲している。プロダクションシステムは Soar[14] や ACT-R[15] などの認知アーキテクチャで用いられている。これらの認知アーキテクチャはヒトの心の振る舞いに関する様々なモデル構築に使われきており、プロダクションシステムは脳の妥当なモデルの 1 つと言える。

プログラム合成による AGI へのアプローチの大きな利点は、様々な個別の問題に対して個別の解決手段を開発する必要がなく、解法プログラムの発見をプログラム合成機構によってエージェント自身に行わせる（もしくは他者から教わった解法プログラムの正しさをエージェント自身に検証させる）ことができるという点である。

プログラム合成の性能を上げるためには合成の対象とする「プログラミング言語」の性質が重要である。プログラミング言語はプログラマーの生産性を上げるためにソースコードの再利用性を上げる機構を進化させてきたが、その機構の一部はプログラム合成の性能を上げるためにも役立つ。サブルーチンはその一例で、強化学習でサブルーチンを扱えるようにすることで、サブルーチンが再利用され、学習が効率化する。他には、例外処理、モジュール機構、マクロ、高階関数、型システムなどに関する知見が役立つ可能性がある。

合成したプログラムの「正しさ」とそのプログラムの実行効率の間にはトレードオフがある。入出力仕様を完全に満たさなくてよければプログラムはいくらでも速くなる。プログラムの正しさを価値で表現 [3] すれば、正しさと効率の間のトレードオフの問題は報酬最大化原理によって自然に解決する。

しかしながら、プログラム合成は一般に容易ではない。プログラムの探索空間は広く、複雑な環境下では、素朴な試行錯誤だけでは有用なプログラムを合成することは不可能である。この問題を解決する 1 つが生成モデル原理によるワーキングメモリの状態の制約であり 3.4 節で述べる。また、4.3 節以降の各節で述べる工夫も必要不可欠であろうと考えている。

3.4 ワーキングメモリと生成モデル原理

我々が実現を目指す脳型ＡＧＩアーキテクチャのもう 1 つの大きな特徴は、ワーキングメモリの状態と環境の状態がベイジアンネットを使って生成モデルの形で結び付けられている点にある。このことは今後アーキテクチャを拡張していく際にも重要な設計指針になる。この指針を**生成モデル原理**と呼ぶことにする。ワーキングメモリは報酬を最大化するための機構の一部にすぎないため、生成モデル原理は報酬最大化原理よりも下位に位置する設計原理である（図 2）。

生成モデル原理にもとづくアーキテクチャはいくつかの利点を持つ。

- ワーキングメモリの状態は環境の状態を抽象化したものと意味付けされる。このため、ワーキングメモリを単なる情報処理の中間的生産物の置き場所ではなく、行動選択のための特徴ベクトルであると見なすことができるようになる。生成モデルであればフィードフォーワードネットワークに比べて、文脈に応じたロバストな特徴抽出が可能である [16]。報酬最大化に適したワーキングメモリの構造は教師なし学習（自己教師あり学習）で経験から獲得することができる。その際には神経科学的知見などを参考にした事前知識を作り込むことで学習を加速できる。また、学習の目的は予測誤差最小化や尤度最大化ではなくあくまでも報酬最大化と考えることで、報酬に無関係な環境の情報（道端の小石の位置など）を無視するような特徴抽出器に理論的妥当性を与え得る。

- 生成モデルは環境のモデルでもある。強化学習エージェントが環境のモデルを持てば、モデルベース強化学習に利用することができ、学習のサンプル効率を上げることができる。また、環境の隠れた状態を観測可能な情報から推定することもできるようになる [3]。

- ワーキングメモリが取りうる状態が環境の状態と矛盾しないように強く制約されるので、プログラム合成における解の探索空間が狭まり、合成の効率が上がる。

- ワーキングメモリの時刻 t と時刻 t+1 の状態の間の関係を生成モデルで学習することにより、記憶の保持と忘却というワーキングメモリの基本的な機能を環境の状態と結び付けて解釈・設計できるようになる [9]。

- ワーキングメモリ内のある場所の値と他の場所の値の関係を生成モデルで学習することにより、目的に応じたデータ表現の変換を自動的に（エージェントにとって無意識に）行えるようにできる。例えば我々が提案する言語野のモデル [17][18] は、文の意味表現と単語列表現の間の双方向変換を行う生成モデルがベイジアンネットで実現可能なことを示している。脳内にある複数の情報表現の他の例として、物体の位置の座標系の表現がある。脳内では領野ごとに網膜中心座標、身体中心座標、環境中心座標など様々な座標系が用いられている[3]。異なる座標系表現の間の相互変換は、感覚連合野の領野間の回路が無意識のうちに行っているものと思われる。このようにデータ構造の変換が自動的に行われるならばプログラムの処理の負担が軽くなるため、プログラム合成の効率が上がることが期待される。

- 生成モデル原理はワーキングメモリの設計だけでなく、強化学習に必要な状態行動対テーブルの圧縮手法の設計にも役に立つ [8]。

なお、生成モデル原理は「大脳皮質がベイジアンネットである」という仮説 [6] からも妥当性が支持されるが、逆に言えば、大脳皮質以外の器官、例えば海馬や小脳の動作に関しては、必ずしも生成モデル原理が成り立たない点には注意が必要である。

また、実世界で動作するＡＧＩアーキテクチャを実現するには大規模化しても現実的な計算コストで動作するベイジアンネットの推論・学習アルゴリズムの開発が必要不可欠である。この点について 5.1 節で議論する。

3.5 開発フレームワーク

複雑な AGI 開発を極力効率化するために、研究開発を容易にする開発フレームワークを構築することが重要になる。

筆者はまず Java 上で動作する lab.Lab [4] と呼ぶ Immediate mode GUI [5]とコンポーネントウエアの機能をあわせもつフレームワークを実装した。これまでこれを用いて RGoal に関する実験を行ってきており、今後さらにシステムを拡張することで脳型 AGI 開発フレームワークを構築しようとしている（図 1）。

4 解決すべき重要な課題

4.1 研究目標の設定

AGI 研究の進展が遅い理由の 1 つに、その最終的な達成目標が曖昧な上に、合意の得られたわかりやすい中間目標すらないという点があるだろう。

「AGI はどのような未知の問題に対してもうまく対処できる機械である」と考える人もいるかもしれないが、我々は**そのようなものは実現不可能である**と考える。ヒトの脳は地球上の環境という極めて特殊な環境のもとで、食料を調達し、敵からは逃げて、種の存続の確率を高めるという極めて特殊な目的のために特化した情報処理装置であり、我々はその特徴を人工的に再現したシステム（脳型ＡＧＩと呼ぶもの）を作ろうとしている。脳型ＡＧＩは極めて特殊な特化型ＡＩであ

[3]参考： https://bsd.neuroinf.jp/wiki/座標系

[4]https://staff.aist.go.jp/y-ichisugi/besom/download.html

[5]参考： https://en.wikipedia.org/wiki/Immediate_mode_GUI

る一方で、人間が行っている様々な仕事をこなせる程度には十分に汎用的で役に立つと思われるからである。

また、ノーフリーランチ定理 [19] が示唆するように、解こうとする問題に特殊化するほどアルゴリズムの性能は向上する可能性がある。いまだ AGI が実現できていない理由は、アルゴリズムの汎用性が足りないのではなく、**むしろヒトが解くべきタスクへの特殊化が足りないために実用レベルの性能が出ていない**と考えるべきである。そして、「ヒトが解くべきタスク」が何かを注意深く分析する必要がある。

筆者は現在のところ、幼児や動物の日常をイメージしていくつかの行動の具体的シナリオを作り、その振る舞いを再現することを目指している。ヒトは大人になると経験によって獲得したプログラムによって複雑なタスクをこなせるようになるが、幼児や動物であれば獲得したプログラムが少ないはずであり、それでも容易にこなせる行動シナリオこそが脳が生まれながらにして持っている機構の推定に役立つと考えるからである。具体的なシナリオの 1 つとして、木の実の入った殻に石をぶつけて中の実を取り出す、といったシナリオのプロトタイプ実装を行った [9]。現在、この枠組みを拡張し、「タスクを解くために必要な道具がなければ道具の作り方を他者に聞く」「他者から道具がある場所を聞かれたら最近見かけた場所を教える」といった簡単な対話を行える知的エージェントのプロトタイプの実装に取り組んでいる。これらのシナリオは、 AGI 実現の具体的な中間目標として役に立つ。

最初のプロトタイプ実装では、エージェントの行動ルールを学習で獲得することは目指さず、最初からエージェントに作り込む。プロトタイプ実装の目的は、作り込んだプログラムの表現の特徴を分析し、それが原理的に経験や対話などから知識獲得可能かどうかを見極める点にある。その次のステップは、知識を持っていないエージェントが、対話や模倣などを通じて他者から知識獲得できることを示すことが目標となる。さらに次のステップは、他者から獲得した知識を、知識を持っていない別の他者に伝達できることを示すことが目標になる。ここまで実現できれば、誰かが偶然発見した有益な知識を集団内で共有するという、ヒトが持つ重要な特性が再現できることになる。また、 AGI エージェントと人間とがお互いに必要な知識を伝達する見通しができるとも言えるだろう。

人類がまず最初に必要とする AGI は、人間に解けない問題を解く機械ではなく、**人間が仕事のやり方を教えればそれを人間程度の上手さで実行してくれる機械**であろう。エージェント間の知識の伝達は、その実現に向けた妥当な中間目標であると考える。

4.2 エージェントの環境の設計

Silver ら [1] は、豊かな環境では報酬を最大化するために様々な能力が要求されるため、豊かな知的能力が生み出されると指摘している。AGI の目標は複雑な実世界の環境で動くことである。しかし AGI アーキテクチャ開発の初期段階からいきなり実世界で動作するように開発を進めることは、計算コストの問題や、ノイズなどの錯乱要因やそれによる実験の再現性の問題などにより困難である。そこで AGI の実現を最短で目指すためには、**実世界の本質的性質を保ったまま極力単純化**した人工的な実験環境を構築することが重要になる。

エージェントの環境を設計するにあたって筆者が参考にしたのはローグライクゲーム[6] [7]である。そこでは時間と空間が離散化されており、マップ上の物体は記号化されている。これらの特徴はチェスや将棋と似ているが、チェスや将棋との大きな違いは、マップのサイズ、アイテムの数、物体同士の相互作用の種類の数にほぼ制限がなく、AGI エージェントはそれらについて経験から学習しなければならないという点である。また、他者（敵など）の振る舞いも非常に複雑である。これらの点が、実世界で動作する AGI に要求される本質的特徴を備えていると思われる。

筆者が設計・実装を進めている実験環境（図 1）においては、高次の知能に関する実験を行いやすくするため、部屋内の地点間の距離の概念をなくすなど、ローグライクゲームよりもさらに大胆な抽象化を行っている。その代わり環境内にいる他者は自分自身と同じアーキテクチャで動くため、エージェントから見た社会的環境は非常に複雑なものになり得る。

4.3 データ構造の設計

ソフトウエアはアルゴリズムとデータ構造という 2 つの要素からなる。豊かなデータ構造が扱えるプログラミング言語ほど、再利用性が高く実行効率の高い複雑なプログラムが容易に書けるようになる。また、ヒトが解くべきタスクに特化したデータ構造を生成モデルのネットワークの形に作り込むことで、プログラム合成の性能を上げられるはずである。

ヒトが解くことが多いタスクの 1 つに、道具作りや料理など物体を操作するタスクがある。また、ヒトは過去や未来の様々な場所での出来事（イベント）の内容について推論することも、日常的に行っている。これらをタスクを得意とする AGI の実現のためは、その

[6] 参考：https://ja.wikipedia.org/wiki/ローグライクゲーム
[7] ローグライクゲーム は NeurIPS 2021 におけるコンペティションの題材にもなっている。 https://ai.facebook.com/blog/launching-the-nethack-challenge-at-neurips-2021

問題の処理に適したデータ構造を表現する生成モデルを設計することが重要であり、我々はその 1 つの案としてオブジェクトレジスタ・イベントレジスタという構造を提案している [9]。また、ヒトの海馬は過去に経験したイベントをエピソード記憶として保持する器官であるが、その具体的な機構を推定する上でも、ヒトが解くべきタスクとそのためのデータ構造に関する分析が重要になるだろう。

複雑なデータ構造を生成モデルの形で表現する際に、物体やイベントのグローバルＩＤをどう実現するかが問題となる。計算機上ではメモリのアドレスを一意性の保証されたＩＤとして使うことができるが、固定したネットワーク上での実現方法は自明ではない。脳では歯状回におけるニューロン新生[8]という現象が知られており、それに関する知見が実現のヒントになるかもしれない。

4.4 社会性の設計

ヒトの重要な特徴の 1 つに社会性がある。ヒトは集団で生活することでそうでない種よりも有利に生き延びてきた。

現在開発中の脳型 AGI 開発フレームワークでは、エージェントどうしが対話する機能の実装も進めている。実装を極力単純化するために、文の内部表現と単語列との間の符号化・複合化は行わず、内部表現をエージェントどうしで直接やり取りする。対話のシナリオの策定とそれを実現するための機構の設計には、対話システム [20] 研究に関する知見を参考にする。

また、他者が視線を向けている物体に自分も注意を向けるといった、共同注視に関する機構をエージェントの身体に作り込むことで、模倣や教示による知識伝達を促進しようとしている。

筆者はヒトの情動もまた報酬最大化原理で理解可能であると考えている[9]。エージェントに社会性を持たせるためには、神経科学的知見などを参考にして、適切な快情動・不快情動を設計する必要があるだろう。例えば他者に知識を伝達する促進する情動と他者から知識を受け取ることを促進する情動の設計が必要になるだろう。

語用論の原理にしたがう会話は、おそらく報酬最大化原理から自然に生まれるだろう考えている。

4.5 メタ強化学習の機構の設計

我々のこれまでの研究では、再帰的強化学習 RGoal に対しあらかじめゴールやハイパパラメタを与えてエピソード的タスクを解かせていたが、ＡＧＩアーキテクチャとしては明らかに不十分である。状況に応じてゴールやハイパパラメタを変更する機構が必要になるだろう。

つまり「学習し行動する」という通常の強化学習の枠組みの外側にある機構に対する学習が必要である。これを**メタ強化学習**と呼ぶことにする。

メタ強化学習の機構を設計する際には、４つの並行した大脳皮質－基底核ループ [21] に関連した神経科学的知見が重要なヒントになるだろう。前部帯状回を含むループは情動・動機に関与すると考えられており、筆者はこれがメタ強化学習を実行する機構ではないかと考えている。

4.6 ハードウエア支援の設計

コンピュータを設計する際、よく使われる機能をハードウエアで支援することによってソフトウエア側の負担を減らすことができる。（グラフィックスアクセラレーターはその一例である。）同じことはプログラム合成システムにも言えるはずで適切なハードウエア支援があれば合成すべきプログラムが簡潔になり、合成の性能が上がるだろう。ここで「ハードウエア」とは合成すべきプログラムの実行を支援する機構、という意味であり、物理的なハードウエアである必要はない。脳においては、大脳以外の器官による支援を想定している。

例えば我々が行った予備的研究 [9] では、眼球運動に関するハードウエア支援の存在を仮定することで物体を操作するプログラムが簡潔になることが確認されている。もし自由な眼球運動による試行錯誤しかなければ、タスク遂行に必要な視覚情報の収集能力の獲得は難しいだろう。

ハードエウア支援のもう 1 つの目的として、強化学習に必要な試行錯誤の範囲を制限することにより、学習を加速することが考えられる。例えば視線の移動には、顕著な視覚的特徴を持つ場所に自動的に注意がひきつけられるというボトムアップの注意と、現在のタスクに関連した場所や特徴に能動的に注意を向けるというトップダウンの注意がある。ボトムアップの注意は報酬最大化の学習を加速するためのハードウエア支援であると解釈できるだろう。同様に、試行錯誤のみでは到底獲得不可能な様々な能動的行為の学習が、ハードウエア支援によるボトムアップの自動的行為によって加速するのではないだろうか。例えば以下の機構が考えられる。

[8] 参考： https://bsd.neuroinf.jp/wiki/ニューロン新生

[9] 情動に関する筆者の考えについては下記文書を参照されたい。一杉裕志 感情や欲求の正体 https://staff.aist.go.jp/y-ichisugi/rapid-memo/emotion.html

- 不快な触覚刺激（熱や痛み）から自動的に体を離す機構があれば、不快な刺激を能動的に避ける行動の獲得が加速するだろう。
- 耳から入った発話内容を自動的にオウム返しする機構があれば、現在の知覚内容を能動的に発話する行動の獲得が加速するだろう。
- 現在の状況に似たエピソードを自動的に想起する機構があれば、現在のタスクを解くために役立つエピソードを能動的に想起する行動の獲得が加速するだろう。

5 個々の構成要素の性能向上

すでに述べたように AGI の本質的機能を極力簡潔な実装によるデモで実現することが AGI 研究の最優先課題と考えているが、実世界での応用に向けた性能向上ももちろん重要である。この節では我々が開発を目指す脳型 AGI アーキテクチャの個々の構成要素に関して、性能向上の余地が非常に大きいことを述べておく。

5.1 大規模化可能なベイジアンネット

筆者はヒトの大脳皮質はベイジアンネットであると考えているが、実世界で動く脳型ＡＧＩを実現するためにはベイジアンネットの大規模化可能な推論・学習アルゴリズムの開発が必要である。

我々はすでに条件付確率の表現に制限を入れることで大規模化への道筋をつけている [22][23][8]。今後は loopy belief propagation のメッセージ計算をニューラルネットワークによって関数近似することで、推論･学習の高速化と精度向上を図る予定である。素朴な loopy belief propagation の実装では、ネットワーク内のすべてのノード、あらゆる結合の重みパターン、すべての反復のイテレーションに対して均一のメッセージ計算･メッセージ伝搬戦略が使われるが、この制約を取り払うことで、以下に述べるように性能を向上させる大きな余地が生まれる。

- 脳内においてはコラムごとに他のコラムとの間の情報伝達の神経回路を持っている。これにならって、ベイジアンネットのノードごとにその結合の重みに最適化されたメッセージ近似計算を行うニューラルネットを持たせるようにすれば、計算コストを減らせる可能性がある。
- 反復計算の序盤と終盤で異なるメッセージ計算を行えば loopy belief propagation の性能を向上させられる可能性がある。反復計算の計算グラフを展開して誤差逆伝搬でパラメタ学習する手法 [24] を使ってメッセージ計算式も学習するようにすれば、反復計算の進行に応じた最適なメッセージ計算が自然に学習される可能性がある。
- Residual belief propagation[25] はネットワークの状態を大きく変化させるメッセージを優先的に伝搬させることで収束性を高める手法である。この手法を生成モデルのネットワークの特徴に合わせて改良することで、より性能を上げられる可能性がある。

5.2 記憶の複数のタイムスケールの機構

Dyna-2 アーキテクチャ[26] は行動価値関数を永続メモリと一時メモリという 2 つの異なるタイムスケールを持つメモリを使って学習することで、コンピューター囲碁において当時としては高い性能を出している。（AlphaGo[27] も似た構造を持っている。）Dyna-2 では囲碁において「あらゆるエピソード（対局）において共通の知識」と「現在のエピソードにおいてのみ使われる知識」をそれぞれ永続メモリと一時メモリで学習している。一時メモリは環境のモデルを用いたシミュレーションにより学習される。同様の機構はおそらく生物にとっても有用であり、普段の平均的状況において役立つ知識と現在たまたま生じた特殊な状況においてのみ役立つ知識を異なる時定数を持ったメモリで保持することは合理的である。臨床神経学では記憶は即時記憶、近時記憶、遠隔記憶に区分されており[10]、それらに関する知見がアーキテクチャ設計のよい指針を与えるだろう。

5.3 効率的な思考の機構

我々はヒトが行う思考のモデルを 2 つ提案してる。1 つは時間を抽象化した脳内シミュレーションを繰り返すことで未経験の行動の価値を推論する思考モード [4][5] と呼ぶ機構であり、もう 1 つはワーキングメモリへの読み書きを繰り返すことにより環境の隠れた状態を推論する機構 [28] である。どちらも現在は RGoal を用いた簡単なモンテカルロ探索で実装されている。Dyna-2 や AlphaGo で用いられているモンテカルロ木探索 (MCTS, Monte Carlo Tree Search) を応用することで、これらの機構はより効率化するだろう。

思考モードの機構のもう 1 つの拡張の方向として、脳内シミュレーションと外界の観測のオーバーラップがある。思考モードの最初の素朴な実装では、脳内シミュレーション中の環境の状態を脳内に完全に保持しているが、実世界で動作する必要がある AGI ではそれ

[10] 参考： https://bsd.neuroinf.jp/wiki/記憶の分類

は不可能である。ヒトが将棋の次の手を考える時にも常に盤面を眺めて駒の状況に関する情報を観測しているように、ヒトは脳内シミュレーション中に必要に応じて外界を観測することで、有限のワーキングメモリの容量で複雑な実世界についてのシミュレーションを行っているものと考えられる。その具体的な機構を明らかにする上でも、報酬最大化原理と生成モデル原理はおそらく役に立つだろう。

6 まとめ

我々が実現を目指す脳型ＡＧＩアーキテクチャのデモの構想について述べた。

AGI の実現に真剣に取り組む研究者の数はその重要さにも関わらず多くない。この現状を打破するには、AGI の実現可能性を多くの人々が一目で確信できるようなデモを、一日でも早く実現することが必要である。

謝辞

本研究は JSPS 科研費 JP18K11488, JP18K18117 の助成を受けたものです。

参考文献

[1] David Silver, Satinder Singh, Doina Precup, and Richard S. Sutton. Reward is enough. *Artificial Intelligence*, Vol. 299, p. 103535, 2021.

[2] 一杉裕志, 高橋直人, 中田秀基, 佐野崇. 単一化の機構を利用した階層型強化学習のテーブル圧縮手法の検討. 第 10 回 人工知能学会 汎用人工知能研究会 (SIG-AGI), 2018.

[3] 一杉裕志, 中田秀基, 高橋直人, 佐野崇. 推論規則の価値を階層型強化学習 RGoal を用いて学習する手法の提案. 第 14 回 人工知能学会 汎用人工知能研究会 (SIG-AGI), 2020.

[4] 一杉裕志, 高橋直人, 中田秀基, 佐野崇. RGoal Architecture:再帰的にサブゴールを設定できる階層型強化学習アーキテクチャ. 第 9 回 人工知能学会 汎用人工知能研究会 (SIG-AGI), 2018.

[5] Yuuji Ichisugi, Naoto Takahashi, Hidemoto Nakada, and Takashi Sano. Hierarchical reinforcement learning with unlimited recursive subroutine calls. In *Artificial Neural Networks and Machine Learning – ICANN 2019: Deep Learning*, pp. 103–114, Cham, 2019.

[6] Y. Ichisugi. A cerebral cortex model that self-organizes conditional probability tables and executes belief propagation. In *2007 International Joint Conference on Neural Networks*, pp. 178–183, 2007.

[7] Thomas G Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of artificial intelligence research*, Vol. 13, pp. 227–303, 2000.

[8] 一杉裕志, 中田秀基, 高橋直人, 佐野崇. 脳の自律的プログラム合成機構のモデルに向けて：２層ベイジアンネットによる記号処理命令の獲得・実行機構. 第 15 回 人工知能学会 汎用人工知能研究会 (SIG-AGI), 2020.

[9] 一杉裕志, 中田秀基, 高橋直人, 佐野崇. 物体操作に適したワーキングメモリを持つ汎用人工知能アーキテクチャの検討. 第 16 回 人工知能学会 汎用人工知能研究会 (SIG-AGI), 2020.

[10] Marcus Hutter. A theory of universal artificial intelligence based on algorithmic complexity. *arXiv preprint cs/0004001*, 2000.

[11] Susumu Katayama. Efficient exhaustive generation of functional programs using monte-carlo search with iterative deepening. In *Pacific Rim International Conference on Artificial Intelligence*, pp. 199–210. Springer, 2008.

[12] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, Vol. 538, No. 7626, pp. 471–476, 2016.

[13] Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sable-Meyer, Luc Cary, Lucas Morales, Luke Hewitt, Armando Solar-Lezama, and Joshua B Tenenbaum. Dreamcoder: Growing generalizable, interpretable knowledge with wake-sleep bayesian program learning. *arXiv preprint arXiv:2006.08381*, 2020.

[14] Soar Tutorial 9.6.0. `https://soar.eecs.umich.edu/articles/downloads/soar-suite/228-soar-tutorial-9-6-0`, 2017.

[15] ACT-R 7 Tutorial Units. `http://act-r.psy.cmu.edu/software/`, 2020.

[16] Hidemoto Nakada and Yuuji Ichisugi. Context-dependent robust text recognition using large-scale restricted bayesian network. *Procedia computer science*, Vol. 123, pp. 314–320, 2018.

[17] 一杉裕志, 高橋直人. 脳における文の意味解析機構のモデル. 第 8 回 人工知能学会 汎用人工知能研究会 (SIG-AGI), 2018.

[18] Yuuji Ichisugi and Naoto Takahashi. A formal model of the mechanism of semantic analysis in the brain. In Alexei V. Samsonovich, editor, *Biologically Inspired Cognitive Architectures 2018*, pp. 128–137, Cham, 2019. Springer International Publishing.

[19] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, Vol. 1, No. 1, pp. 67–82, 1997.

[20] 中野幹生, 駒谷和範, 船越孝太郎, 中野有紀子, 奥村学 (監修). 対話システム. コロナ社, 2015.

[21] Garrett E Alexander, Mahlon R DeLong, and Peter L Strick. Parallel organization of functionally segregated circuits linking basal ganglia and cortex. *Annual review of neuroscience*, Vol. 9, No. 1, pp. 357–381, 1986.

[22] 一杉裕志. 疑似ベイジアンネットを用いた認知モデルのプロトタイピング手法の提案. 第 4 回 人工知能学会 汎用人工知能研究会 (SIG-AGI), 2016.

[23] Naoto Takahashi and Yuuji Ichisugi. Restricted quasi bayesian networks as a prototyping tool for computational models of individual cortical areas. Vol. 73 of *Proceedings of Machine Learning Research*, pp. 188–199. PMLR, 20–22 Sep 2017.

[24] Matthew R Gormley, Mark Dredze, and Jason Eisner. Approximation-aware dependency parsing by belief propagation. *Transactions of the Association for Computational Linguistics*, Vol. 3, pp. 489–501, 2015.

[25] Gal Elidan, Ian McGraw, and Daphne Koller. Residual belief propagation: Informed scheduling for asynchronous message passing. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'06, p. 165173, Arlington, Virginia, USA, 2006. AUAI Press.

[26] David Silver, Richard S Sutton, and Martin Müller. Sample-based learning and search with permanent and transient memories. In *Proceedings of the 25th international conference on Machine learning*, pp. 968–975, 2008.

[27] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, Vol. 529, No. 7587, pp. 484–489, 2016.

[28] 一杉裕志, 中田秀基, 高橋直人, 佐野崇. 階層型強化学習 RGoal を用いた記号推論の実現手法の検討. 第 12 回 人工知能学会 汎用人工知能研究会 (SIG-AGI), 2019.