

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

Preliminary Study of AGI Architecture with Working Memory Suitable for Object Manipulation

一杉裕志^{1*} 中田秀基¹ 高橋直人¹ 佐野崇²
Yuuji Ichisugi¹ Hidemoto Nakada¹ Naoto Takahashi¹ Takashi Sano²

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

² 東洋大学 情報連携学部情報連携学科

² Faculty of Information Networking for Innovation And Design, Toyo University

Abstract: We propose a model of working memory based on the object file theory in cognitive science. This model is expressed as a generative model using a Bayesian network. We implemented a prototype system of cognitive architecture with this working memory, and as a result, we expected that the description of behavior rules would be concise and versatile.

概要

認知科学におけるオブジェクトファイル理論を参考にしたワーキングメモリのモデルを提案する。このモデルはベイジアンネットワークを用いた生成モデルとして表現される。我々はこのワーキングメモリを持った認知アーキテクチャのプロトタイプシステムを実装し、行動ルールの記述が簡潔で汎用性の高いものになる見込みを得た。

1 はじめに

ヒトが住む環境は主に物体から構成されており、ヒトが行う知的作業の多くは、物体の操作である。例えば道具を作成し、それを使用して食料を収穫し調理するには複雑な物体操作の手順が必要である。ヒトに似た汎用人工知能の実現には、このような物体操作タスクを得意とする脳のアーキテクチャの解明が役立つはずである。

本稿では認知科学におけるオブジェクトファイル理論 [1] の特徴を取り入れた脳のワーキングメモリのモデルを提案する (図 1、図 2)。オブジェクトファイルは物体の属性を組にしたものを数個記憶する機構である。提案モデルは、オブジェクトファイルに似た機能を持つワーキングメモリを、ベイジアンネットワークを用い

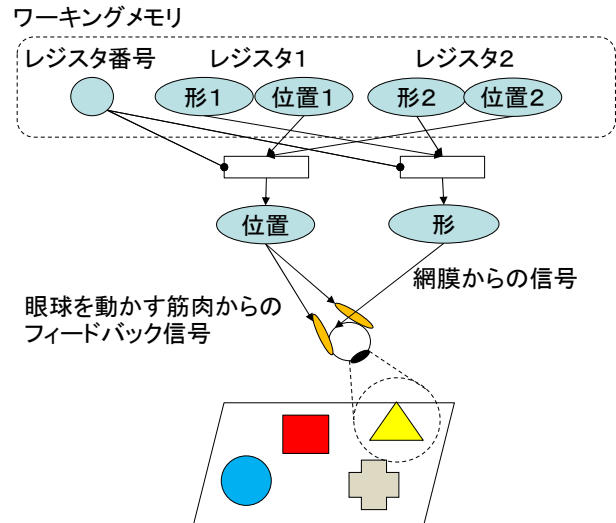


図 1: ワーキングメモリの状態からセンサー入力の値を生成する過程をベイジアンネットワークによって生成モデルとして表現したもの。ワーキングメモリは複数の物体の属性の組を保持する。レジスタ番号で指定された番号の物体の属性がセンサーの値を生成する。詳細は 3.2 節参照。

*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

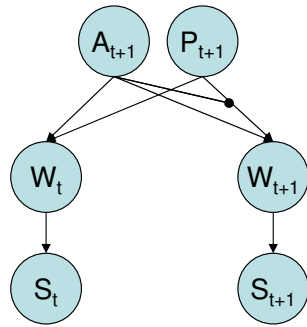


図 2: ワーキングメモリの機能を実現するための生成モデルをベイジアンネットで表現したもの。

て生成モデルとして表現する。このモデルはオートエンコーダーを積み重ねた形をしており、ワーキングメモリを読み書きする行動ルールを、原理的には経験のみから獲得可能になるはずである。

本研究は我々がやっているヒトの前頭前野周辺の情報処理機構のモデルに関する研究の一環である。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してとらえている [2][3]。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。

提案するワーキングメモリのモデルはまだ実装はされておらず、まずその前にプロトタイプシステムを実装した。そして、エージェントが自律的に獲得すると想定されるプログラムを、このプロトタイプシステム上で記述してみることで、提案アーキテクチャの妥当性の予備的検討を行った。

本稿は以下のような構成になっている。まず 2 節でワーキングメモリの生成モデルによる表現について述べた後、3 節で物体操作に適したワーキングメモリの構造について述べる。4 節でプロトタイプシステムとその上で動作するプログラム例について説明し、提案アーキテクチャの特徴について考察する。5 節ではモデルの様々な拡張の可能性について述べる。6 節で関連研究について述べ、最後に 7 節でまとめについて述べる。

2 ワーキングメモリの生成モデルによる表現

この節ではワーキングメモリをベイジアンネットを用いた生成モデルの形で表現する方法について述べる(図 2)。ヒトではワーキングメモリの実体の多くの部

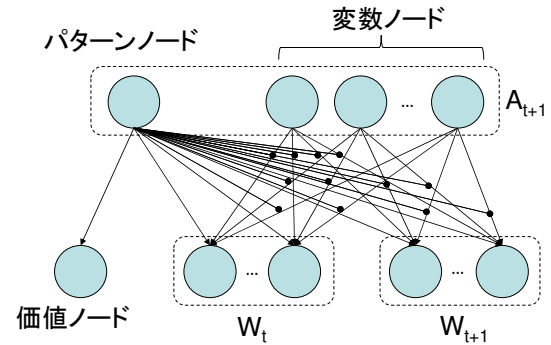


図 3: ワーキングメモリを読み書きする行動ルールとその価値を学習する BESOM のネットワーク。下の層に訓練データを与えることで、 W_t を W_{t+1} に書き換えるルールを抽象化したものが、このネットワークの結合の重みとして学習される。また、 W_t の値が与えられれば、報酬期待値を最大化する W_{t+1} の値が決まる。(各ノードの役割については [3] を参照。)

分はおそらく大脳皮質上にある¹。例えばワーキングメモリに複数の形式の情報を統合する際に、前頭前野が活性化することが報告されている [6]。我々は大脳皮質は一種のベイジアンネットであると考えている [7]。もしそれが正しいならば、大脳皮質上のワーキングメモリの機能もまた、ベイジアンネットを用いて実現されているはずである。なお、モデルを設計するにあたって、脳内に物理的に存在するのはすべて現在の時刻 t の情報を表現するノードだけであり、各ノードからはせいぜい時刻 $t-1$ の情報のみが参照できるにすぎないと仮定した。

モデルの表現には、我々が大脳皮質の計算論的モデルとして提案している BESOM [7] を用いる。BESOM は一種のベイジアンネットワークであり、条件付確率表に制約を入れることでパラメタ数の爆発を抑えている。最新版の BESOM は、ゲートによりノード間の接続を制御する機能を持っている [8][9][10]。我々は以前にワーキングメモリの生成モデルによる表現方法を示した [3] が、本節ではそれをより詳細化して説明する。

図 2 のノード W はワーキングメモリの状態、 S はセンサー入力、 A は強化学習の機構により選択される行動ルールを表す。 P はメモリの状態のデフォルトの振る舞い(値の保持・忘却)を規定する。 A_{t+1} は、メモリの値を更新するときは P_{t+1} の影響を抑制する。そのことを A_{t+1} から P_{t+1} と W_{t+1} の間を制御するゲート(黒丸)で表現している。

ワーキングメモリの更新規則はこのモデル上の推論として定式化されるが、脳内では loopy belief propagation のメッセージを時間をさかのぼって送ることは物理的に不可能なため、脳はなんらかの近似推論を行っ

¹一部はおそらく海馬にもある。

ていると推測される。現時点では、以下に説明するように2段階のMAP推定が行われると考えている。まず時刻 t におけるワーキングメモリ W_t の値が決定されているとする。その値から A_{t+1} と P_{t+1} の値をMAP推定し決定する。この推論の際には子ノード W_{t+1} は存在しないものとして扱う。次に、 A_{t+1} と P_{t+1} と S_{t+1} の値から W_{t+1} の値をMAP推定する。この推論の際には W_{t+1} の時刻 $t+1$ における親ノード A_{t+2}, P_{t+2} は存在しないものとして扱う。

図2はオートエンコーダーが2層積み重なった形をしており、値 S_t と S_{t+1} のペアが与えられれば、原理的には、パラメタを教師なし学習することが可能である。この性質を用いて、ワーキングメモリの読み書きを含む行動ルールを経験から自律的に獲得する汎用人工知能アーキテクチャがおそらく実現可能であろう。我々は特殊な2層ベイジアンネットと数値ベクトルを抽象化したパターンを学習できることを示した[10]。その機構をワーキングメモリの読み書きルールの獲得に応用する場合のネットワークの構造を図3に示す。

図2のモデルにはセンサー入力は書かれているが環境に対する行動出力については書かれていない。しかし、ワーキングメモリのある場所への値の書き込みが行動出力を引き起こすようにエージェントのハードウェアが作られているならば、ワーキングメモリの読み書きだけで入出力が実現できることになる。これは計算機アーキテクチャにおけるMemory-mapped I/Oと同じである。

ノード P は値の保持だけでなく忘却のルールについても規定する。例えばワーキングメモリの個々の要素を一定の確率で0にするというルールがあり得る。BESOMにおいて値0は“unknown”を表す。環境の状態は時々刻々と変化するので、認識した物体の状態も、時間とともに未知の状態に変化する可能性があるため、ワーキングメモリの値が“unknown”に変化することは意味的に妥当である。エージェントはタスクの遂行に必要な情報が“unknown”であるときは、その情報を推論するか環境を観測することで得ればよい。自動的な忘却には、プログラムによる明示的なメモリ消去の手間を減らすという利点がある。また、忘却によりメモリの大部分が値0になっていれば、行動ルールの学習に使う訓練データがスパースになり、汎化性能の向上に役立つと期待できる。

ノード P と W の間の結合の重みは、生得的に決まった値を初期値または固定値として与えることで、エージェントの汎化性能を上げられる可能性がある。

この節ではワーキングメモリを1つのノード W で表現したが、実際にはワーキングメモリは生物が環境で生き残るために必要なタスクを効率的にこなせるような構造を持っているはずである。次の節では、認知科学におけるオブジェクトファイル理論の特徴を取り

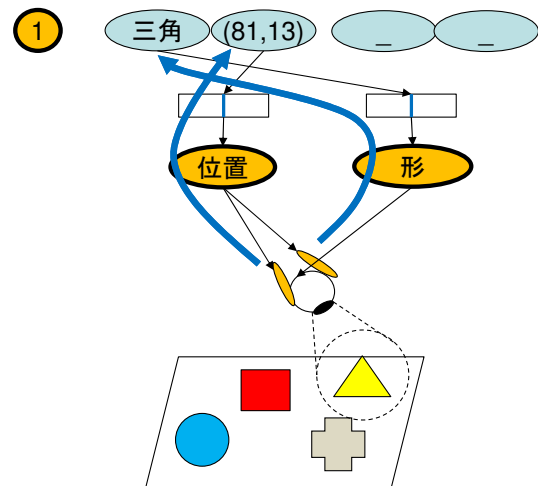


図4: レジスタ1に物体の属性をセットする時の情報の流れ。

入れたワーキングメモリの構造を提案する。

3 物体操作に適したワーキングメモリの構造

3.1 特徴統合理論とオブジェクトファイル理論

特徴統合理論とオブジェクトファイル理論は、視覚バインディング問題に関連する認知科学的理論である。

Treismanらは、視覚情報処理の初期段階で特徴(輝度、色、傾きなど)ごとに空間的に並列に処理された情報に対し、逐次的に空間的注意を向けることで物体ごとに特徴を結びつけるとする特徴統合理論を提唱した[11]。Kahnemanらは結びつけられた物体の属性がオブジェクトファイルとして短期記憶で保持されるとするオブジェクトファイル理論を提唱し、実験によりオブジェクトファイルの数は4個程度であると主張した[1]。

特徴統合理論とオブジェクトファイル理論はどちらも最初に提唱された形が完全に正しいわけではないかもしれないが、ヒトの脳の視覚情報処理のおおまかな特徴をとらえていると考えられる。そこで本稿では、ワーキングメモリのアーキテクチャを設計するにあたり、これらの理論を参考にした。

3.2 提案アーキテクチャ

設計したワーキングメモリのアーキテクチャを図1に示す。本稿では、オブジェクトの属性を組にしたものを、マイクロプロセッサのレジスタになぞらえてオ

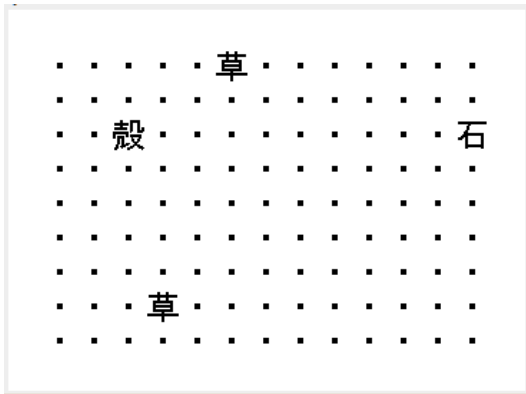


図 5: プロトタイプシステムにおける環境の初期状態の例。

ブジェクトレジスタと呼ぶ。物体の属性には色・動き・大きさなど少なくとも数個以上あると思われるが、この図では形と位置の2つだけを示している。

図1は、オブジェクトレジスタの状態からセンサー入力の値を生成する過程をベイジアンネットで表現した生成モデルである。図の白い四角は我々がゲート行列 [8] と呼ぶもので、ゲートを制御するノード（この図ではレジスタ番号ノード）がゲート行列の上流にあるノードと下流にあるノードの間の接続を制御できることを表現している。レジスタ番号の値が1ならばレジスタ1（形1と位置1の組）、2ならばレジスタ2（形2と位置2の組）が下流のノード（形と位置）に接続される。

図4はエージェントが眼球を動かして物体を探索している時の情報の流れを表している。眼球は目の前にある複数の物体の中からランダムに1つ選んだ物体の方向を向くものとする。そのときの網膜の情報は物体の形を表現するノードへ、眼球の向きを表す情報は物体の位置を表現するノードに向かう。レジスタ番号の値が1であれば、これらの情報は形1と位置1のノードに送られることになる。

エージェントはレジスタ番号の値を能動的に設定することで、レジスタ1とレジスタ2のどちらかに物体の情報をしまうことができる。ゲートが遮断されている方のレジスタの値は、図2の機構により変化せずに保持される。

4 プロトタイプシステム

4.1 プロトタイプシステムの目的

提案するアーキテクチャのもとでエージェントに行動ルールを自律獲得させる際に、原理的な困難がないかどうかを予備的に検証するために、物体操作の簡単

なタスクを解く認知アーキテクチャのプロトタイプシステムを実装した。ワーキングメモリに関しては、提案モデルを忠実に実装するのではなく、タスクの実行に最低限必要な振る舞いを直接作り込む形で実装した。

エージェントは、あらかじめ与えられた行動ルールにしたがって行動する。与える行動ルールは、エージェントが理想的に学習を完了した時に獲得されるべきと想定される手続き的知識である。また、このプロトタイプシステムではエージェントはプランニング（脳内シミュレーションによる行動系列の探索）は行わない。エージェントはプランニングをしなくても視覚刺激に誘導されて最適な行動を実行できるものとする。

4.2 タスクの説明

まずエージェントが置かれる環境について説明する。図5は環境の初期状態の一例である。格子状の空間に「殻」「石」「草」といった物体がランダムに配置されている。殻の中には「実」が入っており、石を殻の場所に移動してぶつけることで中の実が取り出せる（マップ上の「殻」が「実」に変化する）。取り出した実は食べることができるが、殻、石、草は食べられない。実を食べると後には「空」という物体が残る。

環境の初期状態が与えられた時、殻から実を取り出して食べて空にすることが、エージェントに与えられるタスクである。

エージェントはプリミティブな行動として「物体2を物体1がある場所に移動する (MoveO2toO1)」と「物体1を食べる (EatO1)」の2つを持つ²。物体1と物体2はそれぞれワーキングメモリ中のレジスタ1とレジスタ2で表現されているものを指す。

4.3 RGoal によるプログラムの実行方法

図6は、殻に石をぶつけて実を取り出して食べるプログラムを、RGoalの行動ルールの集合で表現したものである。

ここで、RGoalにおいてプログラムがどのように実行されるかを簡単に説明する。（詳細は [3] を参照。）各行動ルールは $rule(S,G,A)$ という形をしている。現在のワーキングメモリの値 s と設定されたサブゴール g の値を各行動ルールに書かれている S,G とパターンマッチさせ、マッチする最も特殊なパターンを持つ行動ルールを1つ選択し、その行動 A を実行する。行動 A が $call(g')$ ならばサブルーチン g' を呼び出し、 $set(s')$ ならばワーキングメモリの値を s' に更新する。

²脳ではこれらプリミティブな行動は前運動野で実現され、前頭前野は前運動野に実行を指令する立場にあると想定している。本稿では前頭前野の振る舞いにだけ焦点を当てる。

```
// 実を食べるサブルーチン。
g1 = (_, 空,+,+, _,-,-)
// 実を手に入れる。
1: rule((_, _,-,-, _,-,-), g1, call(_, 実,+,+, _,-,-))
// 実を食べる。
2: rule((_, 実,+,+, _,-,-), g1, Eat01)

// 実を手に入れるサブルーチン。
g2 = (_, 実,+,+, _,-,-)
// 殻と石を探す。
3: rule((_, _,-,-, _,-,-), g2, call(_, 殻,+,+, 石,+,+))
// 石を殻にぶつける。
4: rule((_, 殻,+,+, 石,+,+), g2, Move02to01)

// 殻と石を探すサブルーチン。
g3 = (_, 殻,+,+, 石,+,+)
// 石を探す。
5: rule((_, _,-,-, _,-,-), g3, set(2, _,-,-, _,-,-))
// 殻を探す。
6: rule((_, _,-,-, 石,+,+), g3, set(1, _,-,-, _,-,-))
```

図 6: 実を食べるタスクを解くプログラムを RGoal の行動ルール集合で表現したもの。Java 言語で書かれた実装を読みやすく整形した。行動ルールは rule(状態, サブゴールの状態, 行動) という形をしている。状態は (レジスタ番号, 形 1, x1, y1, 形 2, x2, y2) という 7 次元のベクトルである。状態に対するパターンマッチの際に、“_” は任意の値に、“+” は 0 以外の任意の値にそれぞれマッチする。

今回はワーキングメモリの忘却は実装していない。設定された値は set 命令で明示的に書き換えられない限り、保持される。

4.4 プログラムの実行過程

図6のプログラムは、6つの行動ルール（ルール1からルール6まで番号を付けた）から構成される。プログラムの実行は以下のように進む。

まずプログラムの実行開始前に、最初のゴールが「目の前に「空」がある状態」（図6の g1）に設定される³。

まず最初のゴールは g1 なので行動ルール1と2が選択の候補である。ワーキングメモリの初期状態にはすべて値0が入れているのでマッチするのはルール1であり、サブルーチン g2 が呼び出される。次にルール3が選択されサブルーチン g3 が呼び出される。

サブルーチン g3 のゴールはオブジェクトレジスタ1に殻の形と位置、レジスタ2に石の形と位置をセットすることである。その最初のステップとしてルール5の set 命令が実行される。この命令は、ランダムに物体に視線を向けて、その時に見えた物体の形と位置をレジスタ2にセットする。もしそれが石であれば次はルール6が選択されるが、そうでなければ石が見えるまでルール5が選択され続ける。ルール6も同様に、ランダムに見えた物体をレジスタ1にセットする。それが殻であればゴール g3 が達成されたのでサブルーチン g2 に戻るが、そうでなければルール6が選択され続ける。

戻ってきたサブルーチン g2 ではルール4が選択されるので石を殻にぶつける行動 MoveO2toO1 が実行される。その結果、物体1は「実」に変化することが観察されるのでゴール g2 が達成され、サブルーチン g1 に戻る。

戻ってきたサブルーチン g1 ではルール2が選択され、EatO1 が実行され、その結果、物体1が「空」に変化し、ゴール g1 が達成され、タスクが終了する。

4.5 考察

図6のプログラムは非常に簡潔かつ抽象的であり、タスクの実行に本質的に必要な情報のみが記述されている。プログラムの簡潔さは、獲得した知識の汎化能力の高さと記憶コストの小ささにつながる。このことは、提案アーキテクチャが脳のモデルとしても汎用人

³実際の脳では、最初のゴールは前頭前野以外のどこかの部位（おそらく前部帯状回など）が生理的欲求にしたがって設定すると想定しているが、本稿では扱わない。本稿で扱う前頭前野のモデルは、他の部位が設定したゴールを達成するための行動計画・実行を担当する。

工知能アーキテクチャとしても有望であることを示している。

プログラムが簡潔になっている理由は主に2つある。1つは、物体操作が「操作対象の物体を探す」と「物体を動かす」の2つの段階から構成される点であり、これにより、物体の具体的な座標に依存しない行動ルール表現になっている。もう1つはタスクに無関係な「草」などの存在がプログラム中に現れない表現になっている点である。すべての物体はデフォルトで無視されており、タスクを解くために必要な物体の情報だけを、サブゴールにあわせて能動的に取得するプログラムになっている。

ルール6の set 命令は物体2の情報の保持を明示的には指示していないが、図2のノード P の働きにより保持される。このようにワーキングメモリのデフォルトの挙動を分離したことで、行動ルールが学習すべき情報が少なくなり、学習がより容易になることが期待できる。

このプロトタイプシステムでは、エージェントが持つ眼球という「ハードウェア」もプログラムの簡素化に不可欠な役割を果たしている。ランダムな視線移動はプログラム中には一切表現されておらず、いわば眼球が自律的に行うと仮定している⁴。一般に、進化の未獲得されたハードウェアによるサポートがあれば、プログラムは簡素化され、自律的プログラム獲得が容易になるだろう。また、本稿では扱っていないが、ハードウェアのサポートがプログラム獲得を直接的に促すこともあるだろう。例えば生物にとって重要な物体の視覚的特徴（熟した果物の色や危険な昆虫の模様）に視線が向きやすくなるようなハードウェアを持つ個体は、生存に必要な知識をより速やかに獲得できるだろう。

提案アーキテクチャでは、ワーキングメモリはオブジェクト指向言語のヒープのような自由度の高い記憶機構ではなく、はるかに制限の強いものになっている。オブジェクトは環境中に存在する物体そのものであり、ワーキングメモリが保持するのは定数個の物体の属性の組にすぎない。エージェントが環境モデルを使って脳内シミュレーションを行う場合でも、エージェントの視点からはあくまでもオブジェクトは（脳内でシミュレーションされた）環境中に存在する。提案アーキテクチャはこの強い制限のおかげで、行動ルールの自律的獲得を現実的なものにするだろうと考えている。

一般に物体がどのような属性を持つかは、物体の種類による。本稿のプロトタイプシステムでは形と位置のみとしたが、一般には物体が持つ様々な特徴を教師なし学習することにより、物体のクラスとそのクラスが持つ属性が自己組織化されるものと思われる。例えばレストランに対しては「洋食か和食か中華か」や「高

⁴実際の脳では視線移動は前頭眼野や上丘の役割である。

いか安いか」「開店中か閉店中か」といった属性が獲得されるかもしれない。クラスとその属性は、プログラムが扱う「データ構造」である。認知アーキテクチャの性能を上げるためには、様々なタスクに適したデータ構造を自律的に獲得する機構の実現が必要である。

提案アーキテクチャは物体の状態を固定個の属性の組で表現するので、その表現力には限界がある。例えば他者の心の状態は表現しきれない。そのような複雑な属性を表現するためには、別の機構が必要になる。1つの案として、特定の人物の心の状態に関するエピソードや推論規則の集合によって、実質的にその人物の心の状態を表現する方法が考えられる。その記憶場所が海馬や大脳皮質上の長期記憶であるならば、短期記憶の容量に制限されず、十分に複雑な情報を保持することができる。そのような機構の詳細について検討することは将来の課題である。

図6のような行動ルール集合をエージェントが経験のみから本当に獲得できるのかどうかはまだ明らかではないが、現時点では特に本質的な障害はないと考えている。ただし、学習を促進するための仕組みの作り込みや、モデルのパラメタによい事前分布を設定することは必要不可欠であろう。実際に自律的な行動ルール獲得ができることを示すことは、将来の重要な課題である。

なお、ヒトに似た知能を実現するためには、ワーキングメモリの構造にさらにいくつかの拡張が必要である。現時点で必要だと考えている拡張について、次の節で述べる。

5 将来の拡張

5.1 物体間の関係

本稿の提案アーキテクチャでは、複数の物体間の関係についてはまだ扱えない。例えば「物体1は部品として物体2を持っている」「物体1の上に物体2がある」「物体1と物体2は似ている」などのような関係を表現できるようにモデルを拡張する必要がある。そのためには、ワーキングメモリ上に物体間の関係の種類を表現するノードを導入し、感覚連合野の回路や前頭前野による推論により関係を認識するようなアーキテクチャが考えられる。このような機構があれば「電子レンジのスタートボタンを押す」「机の上のコップを手取る」「着ている服と似た服を探す」といった行動ルールが表現可能になる。例えば電子レンジのスタートボタンを押すためには、まず電子レンジを探し、つぎにその部品であるスタートボタンを探すことで「電子レンジが部品としてスタートボタンを持っている」という状況を認識し、それからスタートボタンを押す、という手順をとればよい。

5.2 イベントレジスタ

本稿で提案したモデルは空間内に存在する物体のうちいくつかに注意を向ける機構であるが、それを時間軸に対しても拡張することができる。図7はそのようなモデルの1つの案である。イベントレジスタは、現在もしくは過去に起きたイベント（あるいは未来に起きうるイベント）の情報を保持する。1つのイベントは「いつどこで誰が誰に何をどうした」というような単文で表現可能な情報で構成される。現在のイベントであればセンサー入力から得た情報、過去の情報であればエピソード記憶から想起した情報が、イベントレジスタ番号で指定されたイベントレジスタにしまわれる。意味役割ノードは、イベントレジスタ内のノードの1つを意味役割の種類（例えば動作主や被動作主など）で指定し、ゲートを制御して感覚情報と結びつける。個々のイベントレジスタの中のノードがオブジェクトレジスタ、意味役割はオブジェクトレジスタ番号だと思えば、図7のモデルの中にはイベントレジスタごとに図1のオブジェクトレジスタのモデルが含まれていることになる。

ワーキングメモリにイベントレジスタの構造を持たせることにより、エージェントは複数のイベントに関わる推論ができるようになる。例えばエピソード記憶から「きのう戸棚にチョコレートがあったのを見た。」というイベント、センサー入力から「いま戸棚にチョコレートがない。」というイベントをワーキングメモリにセットすることで「誰かがチョコレートを食べた。」という別のイベントの存在を推論することができる。

5.3 文の意味表現

ヒトが行う知的作業には、物体の操作以外にも、言語を使った対話や思考がある。我々はヒトの脳のウェルニッケ野に、自然言語の文の意味を表現する場所があると考えている [12][13]。有限長の単語列はベイジアンネットを用いて固定長の数値ベクトルの形をした意味表現に変換でき [12][13]、さらに RGoal のパターンマッチを使った推論機構 [14] を用いてさまざまな推論に用いることができるはずである。文の意味表現は、if, not, every, some, may, must などの単語を用いた文も表現できる。これは 5.2 節で述べたイベントレジスタでは表現できない情報である。

ここで、単語の意味を経験から学習することができるかどうか問題となる。多くの名詞・形容詞・動詞については、イベントレジスタを経由してセンサー入力と接続することで環境に接地させることができる。直接センサー入力に結び付けられない単語であっても、それを扱う推論規則の価値を [3] で述べた方法で学習する

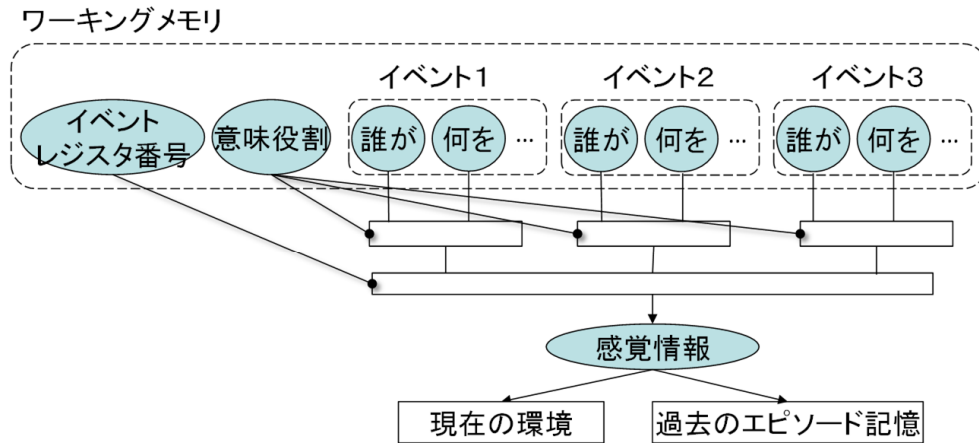


図 7: 複数のイベント間の関係を扱うためのワーキングメモリの構造の案。個々のイベントレジスタは「いつどこで誰が誰に何をどうした」という情報を保持する。

ことで実質的に意味を学習できると考えているが、今後実験による検証が必要である。

6 関連研究

ワーキングメモリを持つ認知アーキテクチャは多いが、提案アーキテクチャはワーキングメモリが生成モデルの形で定式化されている点に独自性がある。ワーキングメモリの値はセンサー入力を通して環境の状態に接地しており、これにより、[3]で示したように、ワーキングメモリを更新する行動ルールの価値を経験から学習可能にしている。

いくつかの関連するシステムについて以下に説明する。

Soar[15], ACT-R[16]は、我々のプロトタイプシステムと同様に、プロダクションシステムをベースにした認知アーキテクチャであり、いずれもワーキングメモリ上にオブジェクト（属性値の組）を保持できる。しかし、提案アーキテクチャのようなオブジェクトレジスタやイベントレジスタに相当する構造は仮定されおらず、ワーキングメモリは汎用的な記憶装置になっている。

Soarのワーキングメモリはオブジェクトのグラフ構造を保持できるため、表現できるデータ構造の自由度が高いが、容量に制限のある脳の短期記憶と対応が付くものではない。

ACT-RはSoarと比較して脳との対応をより重視した設計になっている。ACT-Rではワーキングメモリはバッファ、オブジェクトはチャンクと呼ばれる。プロダクションルールは、ゴールモジュール、宣言的記憶モジュール、視覚モジュール、運動モジュールが持つバッファに対してパターンマッチや更新を行う。それ

ぞれのバッファは一度に1つのチャンクしか保持できないという強い制約が持たせてある。

DNC(Differentiable Neural Computers)[17]は、ニューラルネットワークで表現されたプログラムを、教師あり学習または強化学習により獲得するシステムである。外部メモリへの読み書きの機能を持っており、複雑なタスクを学習できるようにメモリの構造に工夫がされている。DNCが想定するタスクは汎用の情報処理である。一方本稿の提案アーキテクチャは物体の操作にタスクを絞っており、ワーキングメモリの構造もそれに特化させることで汎化能力の向上をねらっている。

深層学習を用いた表現学習の文脈でも物体をうまく扱えるようにする試みがいくつかある。

GQN(Generative Query Network)[18]は複数の物体から構成される環境において、与えられた2次元の画像から、別視点から見た動画を生成するシステムである。教師なし学習の結果、個々の物体の色や形などの属性が分離表現される。なお本稿の提案アーキテクチャは生成モデルの形はしているが、物体を操作する行動ルールの自律的獲得を容易にすることが目的であるため、複数の物体からなる画像全体を再構成する機構は持っていない。

MONet(Multi-Object Network) [19]も教師なし学習を用いて、複数の物体から構成される画像を意味のあるコンポーネントに分解するシステムである。GQNは認識の際に画像全体を一度に処理するが、MONetは逐次的に注意を向けて1つ1つ物体を切り出していく。ネットワークの容量が限られているとき、1つの物体を処理するネットワークを繰り返し再利用した方が、シーン全体を一度に処理するよりも効率的だろうという仮定が、設計の動機として挙げられている。また、訓練時の物体の個数よりも多くの物体がある場合も、反復回数を増やすことで処理できる。これらの利点は本稿

の提案アーキテクチャにもあてはまり、おそらく脳にもあてはまるだろう。

7 まとめ

本稿ではワーキングメモリの更新・保持・忘却の機構をベイジアンネットを用いて生成モデルとして定式化した。また、認知科学におけるオブジェクトファイル理論を参考にした機構を生成モデルで表現した。生成モデルを用いることで、ワーキングメモリを扱うタスクを処理するプログラムを、経験のみから獲得する機構に関して理論的見通しが立った。

提案したワーキングメモリのモデルは実装はまだ行っていないが、このモデルからワーキングメモリの挙動をある程度推定することができる。その推定される挙動に基づいて物体操作のプロトタイプシステムを作成し、物体操作の手順が極めて汎用性の高い形で簡潔に書けることを確認した。

今後の拡張の方向として、よりヒトに似た知能を実現するために、複数の物体間の関係、複数のイベント間の関係、言語の意味を扱うためのワーキングメモリの構造を検討した。

プログラムはアルゴリズムとデータ構造から表現される。脳の前頭前野がプログラム合成システムだとすれば、その性能を決めるのはアルゴリズムの表現方法とデータ構造の表現方法である。前者についてはこれまでの RGoal の研究でかなり具体化されたが、後者については本研究により具体化の方向性が見えたと考えている。

本研究のように、神経科学的知見や認知科学的知見との対応を強く意識し、ヒトが行うタスクに特化した汎用人工知能アーキテクチャの実現を目指した研究が増えることを期待する。

謝辞

本研究に関して議論をしていただいた竹内泉氏に感謝いたします。

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] Daniel Kahneman, Anne Treisman, and Brian J Gibbs. The reviewing of object files: Object-specific integration of information. *Cognitive Psychology*, Vol. 24, No. 2, pp. 175 – 219, 1992.
- [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 Igor V. Tetko, Věra Kůrková, Pavel Karpov, and Fabian Theis, editors, *Artificial Neural Networks and Machine Learning – ICANN 2019: Deep Learning*, pp. 103–114, Cham, 2019. Springer International Publishing.
- [6] Vivek Prabhakaran, K Narayanan, Z Zhao, and JDE Gabrieli. Integration of diverse information in working memory within the frontal lobe. *Nature neuroscience*, Vol. 3, No. 1, pp. 85–90, 2000.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] Anne M. Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, Vol. 12, No. 1, pp. 97 – 136, 1980.

- [12] 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.
- [13] 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.
- [14] 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.
- [15] Soar Tutorial 9.6.0. <https://soar.eecs.umich.edu/articles/downloads/soar-suite/228-soar-tutorial-9-6-0>, 2017.
- [16] ACT-R 7 Tutorial Units. <http://act-r.psy.cmu.edu/software/>, 2020.
- [17] 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.
- [18] S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural scene representation and rendering. *Science*, Vol. 360, No. 6394, pp. 1204–1210, 2018.
- [19] Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. MONet: Unsupervised Scene Decomposition and Representation. *arXiv e-prints*, p. arXiv:1901.11390, January 2019.