

上位概念の内部情報処理モジュール制御へのグラウンディング

Grounding Superordinate Concepts on the Control of Inner Information Processing Modules

神山 薫^{*1} 深田 智^{*1} 尾関 基行^{*1} 岡 夏樹^{*1}
 Kohyama Kaoru Fukada Chie Ozeki Motoyuki Oka Natsuki

^{*1}京都工芸繊維大学 大学院工芸科学研究科
 Graduate School of Science and Technology, Kyoto Institute and Technology

A robot receives questions such as “What color is this?” or “What shape is this?” The robot then replies, “It is red” or “It is a square” etc. and receives rewards from the interlocutor. Through this kind of interaction, the robot learns that it should reply when it receives a question, and that it should reply using Color Module when asked color, and Shape Module when asked shape. At the same time, the robot learns the correspondence between colors and their names, and shapes and their names.

1. はじめに

我々は、モジュール自体とその組合せ方の両方を学習可能とし、モジュールの組合せにより複雑な問題を解決することを目指している [1]。複雑な問題を扱えるようにするため、モジュールの組合せ方の自由度を上げたいが、自由度を上げるとモジュール数の増加が組合せ爆発を招くため、大規模化を可能とするためには、何らかの対策が必要となる。外部からモジュールの切替え方を教示できれば有効な対策となりえるが、モジュール自体の機能も学習により変化するため、外部から教示するためには、まず、学習により獲得されたモジュールやその組み合わせ方に対して、外部から参照できる名前をつける必要がある。

そこで我々は、学習進行中に、モジュールやモジュールの組合せ方に名前を付けて外部から参照できるようにし、モジュールの切替え方の学習時に外部から助言できるようにすることを目指している。

言葉の意味を計算機やロボットに獲得させようとする研究は多くの蓄積があり、引き続き多くの研究者が取り組もうとしているが、具体物を参照する内容語を獲得対象とする研究がその多くを占める (review paper として例えば [2] を参照)。これに対して、機能語や抽象的な概念 (上位概念) を表す語の意味を計算機やロボットに獲得させようとする研究はまだ手薄である。

本論文では、抽象的な概念を表す語のうち属性を示す「色」「形」「大きさ」などの語を、内部情報処理モジュールにグラウンディングする方法を提案し、計算機シミュレーションによりそれが可能であることを示す。これは、モジュールやモジュールの組合せ方に名前を付けて外部から参照できるようにする試みの第一歩と位置付けられる。

2. 関連研究

本節では抽象語の意味獲得に関する先行研究、および、モジュールの組換えにより知的処理を実現しようとする先行研究を概観する。

2.1 抽象語の意味獲得

抽象語は、広い意味では様々な種類の語を含みうるが、人工知能分野においては概念階層を仮定し、一般化/特殊化操作によって抽象概念を獲得する方法が古くから提案されてきた。

これに対して本論文では、属性を示す「色」「形」「大きさ」などの語の意味を獲得対象とし、それをロボットの内部情報処理モジュールにグラウンディングすることを目指す。

2.2 モジュールの組換えモデル

モジュールを組み合わせることで複雑な問題を解決しようとする研究が多数行われてきた。モジュールの切替え方を限定した場合にはモジュール自体の学習と切替え方の学習を同時に行うことができた [3] が、モジュールの切替え方の自由度を上げた場合は、モジュール数に対して組合せ爆発的に切替え方が増え、学習が難しくなる。そこでモジュール自体はあらかじめ作りこんでおき、切替え方だけを学習する方法 [4] が提案されたり、逆にモジュール自体は学習するが、切替え方は与えておく方法 [5] が提案されてきた。こうした中で、岡はモジュールの組み合わせ方を限定しないことを特徴とするモデル [6] を提案し、坂本らはこのモデルを使い、モジュール機能の学習とモジュールの組換え系列の学習を同時に行うことができることを示した [1]。

坂本らは単純な仮想空間上の迷路探索という比較的単純なタスクを用いて評価実験を行ったが、本研究の目的は、言語獲得というより複雑なタスクにおいてもモジュール組換えモデルが有効であることを示すことである。ただし、研究の第一段階として、本研究ではタスクを単純化し、単純なモジュール組換えモデルを用いて有効性を確認することから始める。

3. 計算モデル

赤い三角、青い丸などの図形を見せられ、それについての質問「何色ですか」「何という形ですか」に対して「赤」や「丸」などと試行錯誤的に答え、その答えに対する評価を利用して正しく答えられるように学習するシステムを構築する。

学習システムの構成図を図 1 に示す。切替モジュール以外の 5 個のモジュールと作業記憶がゲートを通じて接続可能となっている。ゲートは切替モジュールにより制御される。切替モジュールは報酬 (評価) を受け取り強化学習によりゲートの開閉の仕方を学習する。色名モジュールと形名モジュールも報酬を受け取り、それぞれ色と色名の対応、形と形名の対応を学

連絡先: 神山 薫, 京都工芸繊維大学, 京都市左京区松ヶ崎橋上町, kohyama@ii.is.kit.ac.jp

習する。これら以外のモジュールは作りこみで機能は固定とする。図1中の色を付けたモジュールが学習するモジュールである。

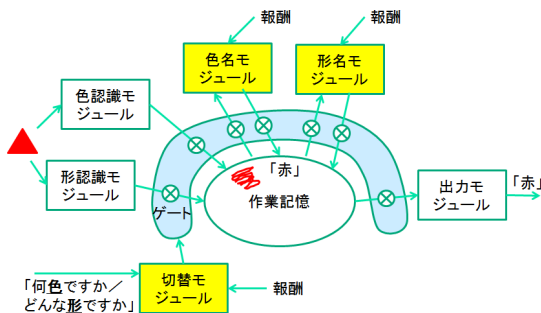


図1: 提案するモジュール型学習モデル

赤い三角形を見せられて「何色ですか」と尋ねられた場合は、以下の手順でゲートを開閉することができ、かつ、色名モジュールが正しい色名を出力することができれば、正しく回答することができる：

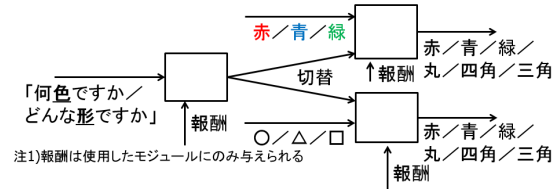
1. 色認識モジュールから作業記憶へのゲートを開け、色認識結果 (赤) を作業記憶上に置く。
2. 作業記憶から色名モジュールへのゲートを開け、色名モジュールにて色認識結果 (赤) を色名「赤」に変換する。
3. 色名モジュールから作業記憶へのゲートを開き、色名「赤」を作業記憶上に置く。
4. 作業記憶から出力モジュールへのゲートを開け、答である「赤」を出力する。

3.1 仕様の単純化

以上のような切替モジュールの動作と色名モジュール/形名モジュールの動作をすべて並行して強化学習により獲得することは、容易ではないと考えられるため、本研究では、手始めに、次のように単純化した仕様 (図2) でシミュレーションを試みることにした。

なお、色名モジュールでは色認識結果 (赤, 青, 緑の3種類) を入力として、「赤」「青」「緑」「丸」「四角」「三角」の6種類の語の中のどれが認識結果と対応するかを学習するものとする。同様に、形名モジュールでは形認識結果 (丸, 四角, 三角の3種類) を入力として、「赤」「青」「緑」「丸」「四角」「三角」の6種類の語の中のどれが認識結果と対応するかを学習する。本計算モデルの特徴は次のとおりである。

1. 「色」とその名称の対応, 「形」とその名称の対応を学習すると同時に、質問に応じて色/形いずれのモジュールを使用するかを学習する。
2. 「色」「形」という上位カテゴリの名前も学習する。



切替モジュール(状態数:2, 行動数:2)

色名モジュール(状態数:3, 行動数:6)

形名モジュール(状態数:3, 行動数:6)

図2: 単純化したモジュール型学習モデル

3. 「色」「形」というカテゴリ名と、内部モジュールの制御 (内部モジュール名) を対応付けて学習する。

4. 評価実験

本研究では、提案モデル (モジュール型モデル) と単一モジュールでの学習との比較実験を行った。単一モジュールで学習する場合の入出力を図3に示す。

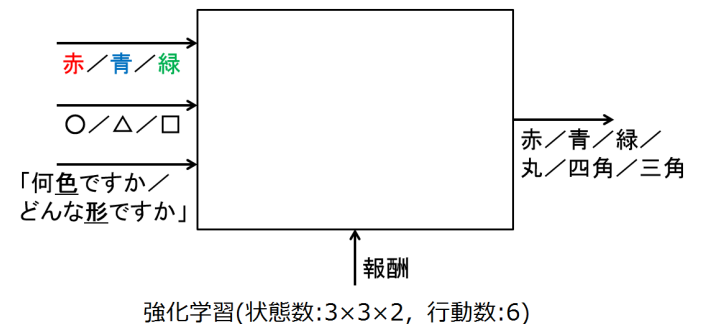


図3: 比較実験で使った単一モジュールからなる学習モデル

モジュール型モデルと単一モジュールモデルとの違いを以下に示す。

1. 状態数の差

単一モジュール (図2) の場合、強化学習の状態数は各入力属性の値の種類の掛け算となる。一方、モジュール型モデル (図3) では、強化学習の状態数は各入力属性の値の種類そのものである。強化学習の速度は状態数に少なくとも比例して遅くなるため、この点ではモジュール型モデルの方が有利だと言える。ここまでは、入力属性の種類は色と形の2種類だけであるとして説明をしてきたが、これが3種類以上に増えると単一モジュールでは

掛け算で効いてくるので、モジュール型モデルとの差が広がることになる。本論文では、入力属性の種類が2種類の場合と3種類の場合のシミュレーション結果を示す。

2. 偽正解の出現

モジュール型モデルにおいては、モジュールの切替えを間違えた状態で偶然正解を出す場合がありうる。これを偽正解と呼ぶ。偽正解は、切替モジュールとそれ以外のモジュールの両方で誤った学習を引き起こすため、学習が遅れる、または、学習に失敗する可能性が生じる。単一モジュールでは偽正解は生じない。

4.1 シミュレーションの設定

システムの解答に対する評価は毎回正しいものが与えられるとして、即時報酬のみのQ学習を用いてシミュレーションを行った。3つの学習モジュールは、同時に並行して学習を進める。

$$Q(s,a) = Q(s,a) + \alpha(r - Q(s,a))$$

学習率 $\alpha = 0.1$ 、正答時の報酬は+1、誤答時の報酬は-1とし、行動選択にはソフトマックス法を用いた。

$$\frac{e^{Q(s,a)/\tau}}{\sum_{b=1}^n e^{Q(s,b)/\tau}}$$

τ の値は、0.05, 0.10, 0.20, 0.40の4種類でシミュレーションを実施した。

4.2 実験結果

入力属性が2種類（色と形）の場合は、均等な割合でランダムに9種類（色3種類×形3種類）の図形を提示し、2種類の質問（色または形）も均等な割合でランダムに選択されとした。入力属性が3種類（色と形と大きさ）の場合も同様に、均等な割合でランダムに図形提示と質問選択が行われるとした。

以下に示すグラフは上記の各条件におけるシミュレーション結果（10試行分の学習曲線）である。グラフの横軸は学習回数（1個の図形を提示し1回質問することを学習1回と数える）、縦軸は解答の正答率である。破線はベースライン（ランダムに解答した場合の正答率）である。

これらのシミュレーションでは、10試行とも同一の訓練例集合（同一の順序で提示される図形と質問）を用いているため、試行ごとの学習曲線の違いは、ソフトマックス行動選択が持つ確率的な性質に起因する。

図4は入力属性が2種類の場合の提案手法及び比較手法のシミュレーション結果である。上の2つが提案手法、下の2つが比較手法の学習曲線である。また、各グラフの右下に記載されているのが τ の値である。

図5は入力属性が3種類の場合の提案手法及び比較手法のシミュレーション結果である。上の2つが提案手法、下の2つが比較手法の学習曲線である。また、各グラフの右下に記載されているのが τ の値である。

また、質問及び入力属性それぞれに2

図6は入力属性が2種類の場合の提案手法及び比較手法のシミュレーション結果である。上の2つが提案手法、下の2つが比較手法の学習曲線である。また、各グラフの右下に記載されているのが τ の値である。

図7は入力属性が3種類の場合の提案手法及び比較手法のシミュレーション結果である。上の2つが提案手法、下の2つが比較手法の学習曲線である。また、各グラフの右下に記載されているのが τ の値である。

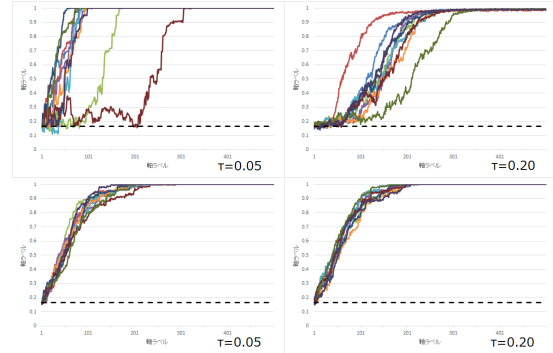


図4: 入力属性が2種類の場合の学習曲線（上段：モジュール型、下段：単一モジュール）

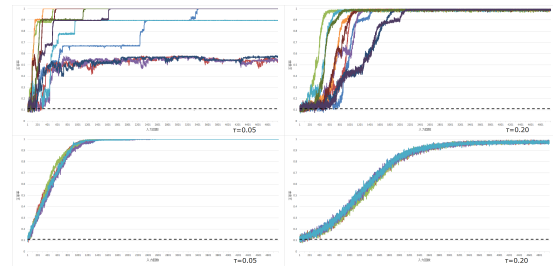


図5: 入力属性が3種類の場合の学習曲線（上段：モジュール型、下段：単一モジュール）

4.3 考察

以上で示した提案モデルの学習結果は、(単純化した仕様の学習システムを用いた場合は)「色」や「形」や「大きさ」という抽象語を、内部情報処理(モジュールの切替え方)にグラウンディングできたことを意味する。ただし、このように相当単純化した場合であっても、モジュール自体とモジュールの切替え方の同時学習は非常に簡単だという訳ではなく、学習が進みにくい場合も確率的にあることが分かる。

提案モデルでは、試行により、グラフの立ち上がり時期が早いものと遅いものがあることが分かる。先述のように、これらのシミュレーションでは、10試行とも同一の訓練例集合（同一の順序で提示される図形と質問）を用いているため、試行ごとの学習曲線の違いは、ソフトマックス行動選択が持つ確率的な性質に起因している。行動選択の確率的な変動により、たまたま偽正解が初期に多く出現した試行では、グラフの立ち上がり時期が遅れたのではないかと考えているが、原因の詳細な検討は今後の課題である。

用するという方法により、「常に提案モデルの方が学習が早い」というまとめとすることもできよう。

5. まとめ

本研究では、色とその名称の対応、形とその名称の対応、大きさとその名称の対応を個々のモジュールで学習すると同時に、それぞれのモジュールをいつ使えばよいかを学習できることを実験的に示した。この結果は、各モジュールが「色」「形」「大きさ」という名称で呼ばれることを学習できた（すなわち、内部処理に名前を付けることができた）ことも示している。

今後は、ノイズの大きさや、モジュール数等を増やしてさらにシミュレーションを行い、提案モデルが持つ性質を明らかにしていく計画である。

図1に示す（単純化していない）構成の学習システム、すなわち、モジュールの組合せ方の自由度が大きいシステムで同様の学習が可能であるか検討することは、今後の大きな課題である。終助詞「か」をきっかけとして、解答を出力するために出力モジュールへのゲートを開けることも含めて学習したい。さらに、「学習したモジュールの名前を利用した、モジュールの組合せ方への助言」の可能性について研究を進めていきたい。

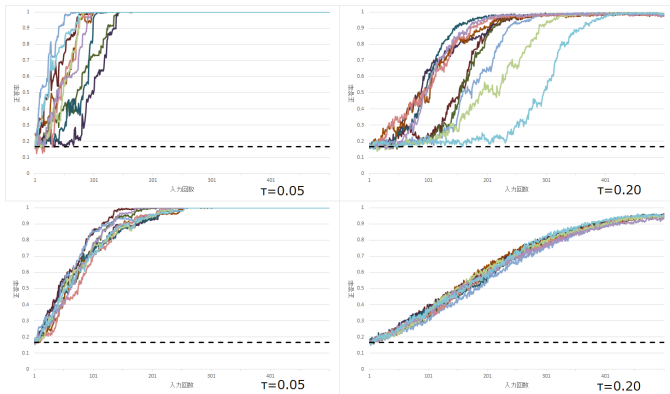


図 6: 入力属性が 2 種類の場合の学習曲線 (上段: モジュール型, 下段: 単一モジュール)

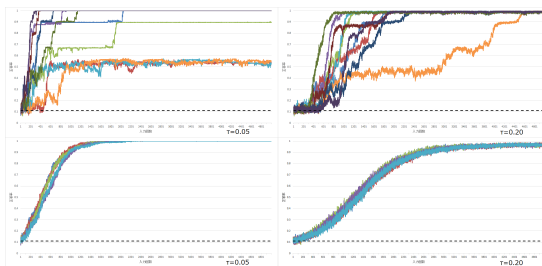


図 7: 入力属性が 3 種類の場合の学習曲線 (上段: モジュール型, 下段: 単一モジュール)

一方、単一モジュールの場合は、このような偽正解による試行ごとの大きな変動は見られず、ソフトマックス行動選択による確率的なゆらぎの直接的な影響だけが試行ごとのわずかな変動となって現れていると考えられる。

モジュール型モデルと単一モジュールを比較すると、グラフの立ち上がりの傾きが異なっている（モジュール型の方が急な傾きで立ち上がっている）ことが読み取れる。これは、先に記した強化学習における状態数の差を反映していると考えられる。したがって、入力属性が 3 種類であるシミュレーションにおける両者の差の方が、2 種類であるシミュレーションにおける差よりも大きくなっている。

以上をまとめると、「立ち上がりの傾きは提案モデルの方が急だが、偽正解の出現により立ち上がりが遅れることがある」ということになる。これは、一見、一長一短の関係にあるように見えるが、たとえば今回のシミュレーションで行った 10 試行はハードウェア資源さえあれば並列に実施可能であることを考慮すると、その 10 試行の中で最速で立ち上がった試行を採

参考文献

- [1] 坂本裕太, 坂戸達陽, 尾関基行, 岡夏樹: モジュール組換え型モデルにおけるモジュールの学習とモジュール組換え系列の学習, 2012 年度人工知能学会全国大会 (第 26 回) 論文集, 3B2-R-2-6, 2012.
- [2] D. Roy: “Grounding words in perception and action: computational insights. TRENDS in Cognitive Sciences,” vol.9, no.8, pp.389-396, 2005.
- [3] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton: “Adaptive mixture of local experts,” Neural Computation, vol.3, pp.79-87, 1991.
- [4] 小川昭利, 大森隆司: “機能部品組み合わせモデルによるナビゲーション行動学習処理の獲得方式の提案,” 電子情報通信学会論文誌, vol.J87-D-II, no.4, pp.987-998, 2004.
- [5] J.C. Sylvester, J.A. Reggia, S.A. Weems, and M.F. Bunting: “Controlling working memory with learned instructions,” Neural Networks, vol. 41, pp.23-38, 2013.
- [6] N. Oka: ““ Apparent free will ” caused by representation of module control,” No matter, Never mind: Proceedings of Toward a Science of Consciousness: Fundamental Approaches, pp.243-249, 1999.