

認知コンピューティングのための汎用ソフトウェアプラットフォームの設計と開発

Design and Development of a Generic Software Platform for Cognitive Computing

高橋恒一^{*123}
Koichi Takahashi

板谷琴音^{*12}
Kotone Itaya

中村政義^{*3}
Masayoshi Nakamura

小泉守義^{*4}
Moriyoshi Koizumi

荒川直哉^{*3}
Naoya Arakawa

富田勝^{*2}
Masaru Tomita

山川宏^{*3}
Hiroshi Yamakawa

^{*1} 理化学研究所生命システム研究センター
RIKEN QBiC

^{*2} 慶應義塾大学大学院政策・メディア研究科
Keio University Graduate School of Media and Governance

^{*3} ドワンゴ人工知能研究所
Dwango AI Laboratory

^{*4} 株式会社オープンコレクター
Open Collector, Inc.

We have been developing a generic software platform BriCA (Brain-inspired Computing Architecture) that can combine an arbitrary number of machine learning (ML) modules to construct higher structures such as cognitive architectures, having in mind its primary utility in the whole-brain architecture project. We discuss requirements analysis and design principles of this cognitive computing platform, report the development of the prototype version 0, and describe plans for further development.

1. 背景

1.1 脳に学んで脳に到達できるか

近年の計算論的神経科学の発展に触発され、脳の神経科学的特性に注目した機械学習手法の研究が盛んである。特に、異なったパラダイムで動作する異種の機械学習手法を複数組み合わせ、各々の機械学習手法単体では到達が難しかった性能あるいは機能を実現できることが示されるなど[Vinyals 14; Kalpathy 14; Minh 15]、脳を真似て複数の機械学習器を構成する研究が盛んになってきた。ディープラーニング[Le 12]が分散知識表現の獲得技術における突破口を開いたことにより、AIの基本問題(フレーム問題, 記号接地問題, 知識獲得ボトルネック)が本質的には解決され、人間並みの汎用人工知能の実現に向けて最も大きな障害は取り除かれたと考えられる。我々が関与する全脳アーキテクチャプロジェクトでは、日本発でいち早く汎用人工知能を完成させることを目的として活動している。

1.2 全脳アーキテクチャ中心仮説

全脳アーキテクチャプロジェクト¹では、次の全脳アーキテクチャ中心仮説(WBA 中心仮説)を設定し、これを検証することを中心的なミッションに据えている。

「脳はそれぞれよく定義された機能を持つ機械学習器が一定のやり方で組み合わせられることで機能を実現しており、それを真似て人工的に構成された機械学習器を組み合わせることで人間並みかそれ以上の能力を持つ汎用の知能機械を構築可能である。」

WBA 中心仮説は、下記の3点の重要な、しかし必ずしも網羅的でない論点を含む。

(1) 脳のモジュール性

大脳皮質のマイクロカラム約 2×10^8 の間の接続率は 6×10^4 程度とされ[Johansson 06]、ランダム結合であった場合2ステップほどで大多数のカラム間の組み合わせが接続される。その一方で、脳は新皮質、小脳、海馬、基底核といった器官レベルから、新皮質の中でも一次視覚野、二次視覚野、連合野、体性感覚野といった領野レベル、また海馬の中の CA1, CA3 といった器官内の部位レベルまで、様々なレベルで解剖学的に独立性を持つ単位で構成される。脳が計算論的に独立したモジュールの集合として構成されるという仮説をモジュール性仮説と呼ぶ。

(2) 脳器官は機械学習器か

仮に(1)の脳のモジュール性仮説が成立するとしても、それら一つ一つが「よく定義しうる(アルゴリズムに書き下しうる)機能を持つ機械学習器(プログラム)」として構成可能かどうか、つまり、全てのニューロンとそれらの間の結合を一々書き下すのではなく、計算論的に一貫性のある単一の原理で部品の動作を定義できるかどうかは自明ではない。脳全体の認知機能を、既知の機械学習アルゴリズムで動作が記述されるモジュールの集合で構築できることを脳器官の機械学習仮説と呼ぶ。用いられる機械学習アルゴリズムは全脳アーキテクチャの完成までに既知となっている必要がある。

(3) 機械学習器の非加算性

個々の機械学習器の機能とヒトの脳全体が発揮する認知機能は質的に異なるように見える。機械学習器を複数積み上げて行った先に脳と同等の機能を持つ学習機械があるかどうかは自明ではない。ただし、神経回路のレベルでは、たとえばニューロン A の発火とニューロン B の発火が同時に起きた時のみ発火するニューロン C が存在し、A や B が直接引き起こすことがない神経活動を発生させた場合には非加算性があると考えられることもできる。また、前述のように2つの機械学習法を組み合わせると新たな機能を実現した例もある[Vinyals 14; Kalpathy 14;

¹ <http://www.sig-agi.org/wba>

連絡先: 高橋恒一, 理化学研究所生命システム研究センター,
565-087 大阪府吹田市古江台 6-2-3, ktakahashi@riken.jp

Minh 15].

上記の全脳アーキテクチャ中心仮説の検証は、未解明の神経科学的な知見や今後の機械学習技術、また認知アーキテクチャ分野での進展を前提とする部分もあるため、10年単位の安定した研究開発活動に支えられた構成論的アプローチで行うのが最も現実的である。本論文では、全脳アーキテクチャ中心仮説を「作ってみることで検証すること」を、できる限り実現性高く、また効率的に支援することを目的に我々が開発を行っているソフトウェアプラットフォーム、BriCA (Brain-inspired Computing Architecture) Version 0 の設計と開発状況を報告する(図1)。

2. 目的

2.1 ソフトウェアプラットフォームの必要性

全脳アーキテクチャ中心仮説およびそのサブ仮説群の構成論的な検証には、多数かつ異種の機械学習器を同時に動作させ、またそれらの間のスケジューリング、同期、通信を適切かつ効率よく仲介できる実行機構が必要である(図1実行機構層)。また、全脳アーキテクチャの構築に向けては、脳アーキテクチャの解明とそれに基づいた認知アーキテクチャの設計、機械学習アルゴリズムの研究開発やライブラリ化、認知アーキテクチャの学習、育成、さらに応用製品への実装など様々な工程が存在する。コミュニティーも巻き込んだ多数の参加者による協力体制の構築が必要となると想定され、これらの工程をシームレスに結合するソフトウェア基盤も重要である。

このような目的に利用できる可能性がある既存のソフトウェアとして、ROS¹や MIRA²などのロボットミドルウェア、Weka³や Garuda⁴、Jubatus⁵などのデータ解析プラットフォーム、また Simulink⁶や LabVIEW⁷、E-Cell[Takahashi 04]などのモジュラーシミュレーション環境などがある。これらの汎用ソフトウェアの発展形が将来的に認知アーキテクチャの実行にも用いられるようになる可能性は否定できないが、現時点ではロボット OS は一般にモジュールの階層性などの複雑なモデルや学習カリキュラム(後述)への対応、データ解析環境やモジュラーシミュレーション環境ではリアルタイム性への対応がそれぞれ弱い。要求仕様の吟味も重要な目的である現段階では、独自ソフトウェアプラットフォーム BriCA V0 を開発することとした。

2.2. プラットフォームの要求仕様

プラットフォームとして有効に機能するため、BriCA は少なくとも下記の要求を満たす必要がある。

1. 任意の新規あるいは既存の機械学習器の実装をプラットフォームのプラグインモジュールとして利用できるよう、モジュールライブラリ化できること。
2. 任意の数 N 個の機械学習器モジュールを階層的に結合した認知アーキテクチャを構成可能であること。
3. 機械学習器モジュール間の統一した通信方式を持ち、任意の機械学習器間の結合が可能なこと。
4. 構成した認知アーキテクチャはセンサー・アクチュエーターインターフェースを介して環境と相互作用できること。ロボット

¹ <http://www.ros.org/>

² <http://www.mira-project.org/joomla-mira/>

³ <http://www.cs.waikato.ac.nz/ml/weka/>

⁴ <http://www.garuda-alliance.org/>

⁵ <http://jubat.us/ja/>

⁶ <http://jp.mathworks.com/products/simulink/>

⁷ <http://www.ni.com/labview/ja/>

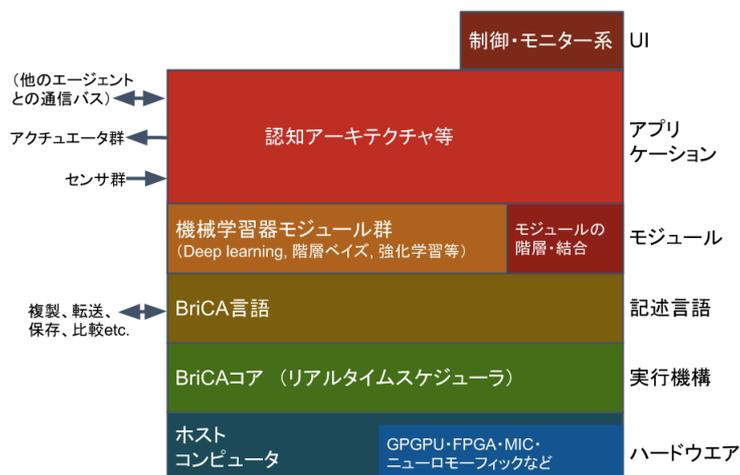


図1. BriCA プラットフォームの概念設計

最もハードウェアに近い層(最下段)から、実行機構層、記述言語層、モジュール層、アプリケーション層(汎用人工知能構築に用いる場合は認知アーキテクチャ)、さらに最もユーザーに近い UI(ユーザーインターフェース)層に至る6層で構成される。言語層でのシリアライズ機構で認知アーキテクチャの複製、転送、保存、比較等を行う。アプリケーション層ではセンサー・アクチュエータ系や他のエージェントとの通信バスと接続する。

ト OS やゲームエンジンなどの他のプラットフォームと結合した動作が必要である。

5. 機械学習器の非同期呼出しを調停するスケジューラーを持つこと。環境が実世界とリンクしている場合にはスケジューラーはリアルタイム性を持つ必要がある。
6. 機械学習器の数 N に対して、ソフトウェア的、実行速度的にスケーラブルであること。
7. 複数の機械学習器で構成されるアーキテクチャを育成させる際に発生する複合学習問題(後述)に対処するため、学習カリキュラムを構築、実行する統一的な機構を持つこと。
8. コミュニティーによる分散開発に対応できること。

3. 手法・実装

BriCA V0 を Java 言語で開発した。V0 開発の目的は、前章で述べた要求仕様を満たすソフトウェアを構築することはもちろんであるが、それよりもむしろ開発を通じて要求仕様の過不足を明らかにし、要求仕様リストそのものとその実現方法を模索することにあつた。なお、前章の要求仕様リストは今回の開発を踏まえた上でのものである。

BriCA V0 では、機械学習器を抽象化し、入力ポート(in)と出力ポート(out)および内部状態を持つ Module クラスを定義し、これをサブクラス化することで任意の機械学習モジュールの実装を可能にした。Module はそれぞれ0個以上の入力ポート in と出力ポート out, 更新間隔 Δt , および内部状態 states を持つ。任意の Module の出力ポートは別の Module の入力ポートと結合可能である(Connection クラス)。更新間隔に応じて、Scheduler が Module を呼び出すことにより時間を進行する。1つの Scheduler と1個以上の Module と Connection が CognitiveArchitecture を構成する。入出力ポートの値は現在のところ符号付き16ビット整数を値とする任意の長さのベクトルであるが、この仕様は仮のものであり今後変更の可能性はある。

各々の Module は下記の計算を実行する fire()メソッドを持つ。

```
out, states <- fire(in, states)
```

つまり、モジュールはユーザーが定義する `fire()`メソッドの呼び出しにより入力ポート群と現在の内部状態の値にのみ依存して出力ポート群の値と内部状態を更新する。この操作は、実際には下記の3段階のフェーズで行われる。

1. `input`: 結合先出力ポートの値で入力ポートを更新
2. `fire`: `results, states <- fire(in, states)`
3. `output`: `out <- results`

主な計算を行う `fire` フェーズでは結果を一時バッファ `results` に蓄え、`input` フェーズと `output` フェーズの2つのクリティカルセクションでのみ入出力バッファを操作することで、モジュール発火の並行性を確保し、モジュール数 N に対するスケラビリティの確保を狙っている。Module の発火は離散事象であり、Module の出力ポートの値は時間区分的に一定 (piecewise-constant) に定義される[Zeigler 00]。Module の発火のスケジューリングは離散事象スケジューラ (実時間および仮想時間スケジューリングをサポート) が行う[Takahashi 04]。スケラビリティの観点から、Module 間の通信には共有メモリモデルではなくメッセージパッシングモデルを採用した。

4. 結果

BriCA のデモアプリケーションとして、過去に山川が開発した認知アーキテクチャ CITTA (Cognition based Intelligent Transaction Architecture) [山川 98] を BriCA 上に実装した。CITTA は GSA (Goal Stack Architecture) の実装でありエージェントの学習処理として認知距離学習を採用している。特徴として同一の意味を持つ情報をスタックに積み上げて保持するためセマンティックスラベルが存在しないことや、センサ入力と行動出力を統一的に扱うという点がある。CITTA では環境から現在の状態を取り出し、それを外部から与えられた目標に到達するための意図をエージェントを通して生成して意図スタックにプッシュする。この意図スタック内の最上部の情報を環境に対して出力し一つのイテレーションが完了する。BriCA ベースの CITTA では GSA 処理部分を BriCA のモジュールとして分離し環境からの入力を受け取り意図スタックを生成して出力バッファに置く。これを環境が受け取り状態を更新することで CITTA 同様のサイクルを実現している。エージェントが探索する環境としてローグライクゲームをベースにした迷路を実装した。現在は環境の時間は同期的に設計されており意図生成が完了する度に環境が更新される。

5. 議論

5.1. モジュール構造の階層性

BriCA V0 ではモジュール間に階層を設定しないフラットなアーキテクチャのみをサポートしているが、脳器官はたとえば新皮質が領野に分けられるなどの階層性を持っている。全脳アーキテクチャ構築の過程においては、あるモジュールが複数のサブモジュールで構成され、複数の開発者が同等な機能を持つモジュールを置き換え、また再利用しつつ試行錯誤を行うなどの状況も頻繁に起きると考えられ、階層的なモジュール構造をサポートする必要がある。

5.2. 分散表象とベクトル表現

BriCA がメッセージパッシングでやりとりするデータは数値ベクトル (分散表象) である。もし脳のモジュール性仮説が成り立つならば、仮に脳を構成する器官や部位がニューロンレベルよ

りも抽象度の高い機械学習器モジュールとして実装されていたとしても、原理的にはモジュール間の情報のやりとりはニューロンの発火頻度の数値ベクトルとして表現出来るはずである。

一方、一般に認知アーキテクチャでは、プランニングや言語表現のパーズングなどにおいてスタックや木構造をはじめとするグラフ構造を扱うことが求められる場合がある。実際に、今回 BriCA への移植を試みた認知アーキテクチャ CITTA は内部に木構造などの内部構造を持っている。グラフ的な構造を分散表象で扱う方法としては、グラフをトラバースして分解し、グラフ上の位置を分散表象化するということが考えられるが、今回の試行ではグラフの分散表象化までは行わなかった。BriCA のような分散表象ベースの認知アーキテクチャでは分散表象によるグラフ構造の取り扱いのデザインパターンを明確にしておくことが必要となる [Eliasmith 13]。

5.3. BriCA 言語

現在、BriCA 上の開発は、モジュールの実装もモジュール間接続の設定も Java コードを書くことにより行っている。今後の課題として、プラットフォーム独立なアーキテクチャ記述言語の開発を企画している (図1記述言語層)。特に、WBA では複数の研究者が互いのアーキテクチャを参照、再利用しつつ、脳の器官に相当する学習器を換装したり、器官内の領域の設計をサブモジュールなどを用いて詳細化したりすることを想定しているため、アーキテクチャ記述言語および関連する開発環境には、そうした開発プロセスに対応できる柔軟な設計が求められる。

5.4. 複合学習問題・カリキュラム

複数の機械学習器で構成される系では、あるモジュールの学習の結果入出力関数が変化して他のモジュールの動作に影響する問題が指摘されている[Sculley 14]。脳の発達過程においては例えば視覚野の場合等で下位の領野から固定してゆく現象が観察されるなど、学習カリキュラムが重要であると考えられる。プラットフォームは、ユーザーが様々な学習カリキュラムを試行錯誤し、一般化、実装、共有できる仕組みを提供する必要がある (図1UI層等が中心に対応)。

5.5. 育成・応用環境

BriCA はロボット、ゲーム、そしてデータ解析などの広範な分野、環境での応用が想定されるため、それぞれの用途で一般的に用いられるプラットフォームとそれぞれセンサー・アクチュエーターインターフェースを介して結合できる必要がある。環境は、認知アーキテクチャの育成環境としても重要である。ロボットについては例えば前述の ROS や MIRA が広く用いられているほか、Gazebo¹などのロボットシミュレーション環境も整備されている。ゲームへの応用に関して、現在普及しているフリーのゲームエンジンとしては Unity², Irrlicht Engine³, cocos2D⁴などが挙げられる。大規模なデータ解析のプラットフォームとしては近年 Jubatus が注目を浴びておりプラグイン開発もサポートしている。また、バイオ分野でのデータ解析では Garuda などが存在する。

5.6. 機械学習ライブラリの充実

BriCA プラットフォームの有用性は、ユーザーが用いることができる機械学習器ライブラリの充実度に多分に依存する (図1モジュール層)。オープンソースで利用できる機械学習ライブラリ

¹ <http://gazebosim.org/>

² <http://unity3d.com/5>

³ <http://irrlicht.sourceforge.net/>

⁴ <http://jp.cocos2d-x.org/>

は言語ごとに様々あるが、中でも代表的な物を以下に挙げる。汎用的なライブラリとしては Scikit-learn, PyBrain, Theano, Pylearn2, Apache Spark, Weka, MALLET, Dlib-ML, 将軍 (shogun) などがある。特化した機能の実装としては SLAM (Simultaneous Localization and Mapping) のライブラリである RatSLAM, RT-SLAM, LSD-SLAM や大脳新皮質のモデル BESOM [一杉 14], 小脳のフィードバック制御モデル MOSAIC [Haruno 01] などが存在する。これらの既存実装を BriCA で利用できるようにするほかに、新たな機械学習手法の実装、評価、共有をコミュニティベースで促進することも重要であろう。

5.7. 性能に関する要求

脳内にはニューロンの発火頻度の上限である概ね 1kHz よりも速い処理は存在しないと考えても大きな問題はない。従って、ニューロンが集合して形成される器官や領野を抽象化した BriCA モジュール間の通信機構に求められる要求は、認知アーキテクチャが保持する全てのモジュールの数 N に依存せずに遅延が 1ms を基準として無視できる程度小さいことである。仮に、「無視できる程度のオーバーヘッド」を 1ms の 1% とすると、 $10\mu s$ となる。BriCA V0 の現状の実装では、同一プロセス内で同期的にデータの受け渡しを行っており、ペイロード無し時の片道通信遅延は 100 ns 程度である (データ示さず)。プロセス間通信を行う分散系 (ROS や MIRA 等) では一般に数十から数百 μs の遅延があるため、これらのような分散系を用いる場合にはプロセス内通信 (ROS の nodelet や MIRA の intra-process など) に限定して利用しなければいけない状況がありうる。

ヒトの脳の計算量は 1 PFLOPS 程度と言われる。現状のコアあたりの FLOPS 値を控えめに 10 GFLOPS と考えると、約 10 万コア程度を集積する必要があるが、スレッド数で 10 万を同時に実行しながら、任意のスレッド間通信のペイロード込みでの遅延を $10\mu s$ 以下に保証するのは、現在の HPC 分野でのスーパーコンピュータを用いた大規模計算の水準を考えれば突拍子のないものではないものの、現在一般に入手可能なハードウェアへの要求としては多少厳しい。今後の GPGPU や MIC, FPGA, ニューロチップなど、ハードウェア面や実行機構のソフトウェア面での進展も見据えながら、今後の課題として考えてゆかなければならないであろう (図 1 ハードウェア層に関連する議論)。

5.8. まとめと今後

本稿では、脳を真似て機械学習器を複数結合することで、より高性能、高機能な学習機械を構成する事を目的としたソフトウェアプラットフォーム BriCA のプロトタイプ V0 の開発と、全脳アーキテクチャ実現へ向けた試みの現状を報告した。

前述の通り、複数の機械学習器を組み合わせることで高度なタスクを達成している例としては Convolutional Neural Network の出力を自然言語生成のための Recurrent Neural Network に与えるもの [Vinyals 14; Karpathy 14] があるが、これらは 2 つの機械学習器の組み合わせに留まっている。Deep Q-Network [Mnih 15] の場合も、ニューラルネットワークの出力として Q 学習の行動価値関数を埋め込んでいるため、モジュラーな設計ではない。

全脳アーキテクチャの実現性を説得力のある形で示すためにしなければならないことは多数ある。そのうちのひとつは、2 つ以上、出来れば 3 つ以上の機械学習器を脳を模した構成で結合する事で、単体では実現困難だった認知機能を実現出来る実例をいくつも示す事である。この機械学習器の非加算性をよ

り一般的に示す事は次期 Version 1 の開発の焦点となると思われる。今後、このようなデモンストレーションを一般からの公募も含めて実現する全脳アーキテクチャハッカソンの開催を予定している。このほか、様々な方法で全脳アーキテクチャ実現に向けたソフトウェア開発、認知アーキテクチャー研究、人材育成、コミュニティ形成などを図ってゆきたい。

謝辞

本研究を行う上で、一杉裕志、市瀬龍太郎、大森隆司、柏岡秀紀、栗原聡、櫻田剛史、佐藤健、泰地真弘人、西本伸志、松尾豊の各氏には有用なご意見、ご指導を頂いた。また、全脳アーキテクチャ若手の会一同との議論を通じて多数のヒントを得た。感謝したい。

参考文献

- [Johansson 06] Johansson, C. Lansner A.: Towards cortex sized artificial neural systems, Neural Networks, Science Direct (2006)
- [Vinyals 14] Vinyals, O. Toshev, A. Bengio, S. and Erhan, D.: Show and Tell: A Neural Image Caption Generator, arXiv, 1411.4555 (2014)
- [Karpathy 14] Karpathy, A. Fei-fei, L.: Deep Visual-Semantic Alignments for Generating Image Descriptions, arXiv, 1412.2306 (2014)
- [Mnih 15] Mnih, V. Kavukcuoglu, K. Silver, D. Rusu, AA. Veness, J. Bellmare, MG. Graves, A. Riedmiller, M. FidjeLand, AK. Ostrovski, G. Petersen, S. Beattie, C. Sadik, A. Antonoglou, I. King, H. Kumaran, D. Wierstra, D. Legg, S. Hassabis, D.: Human-level control through deep reinforcement learning, Nature, Nature Publishing Group (2015)
- [Le 12] Le, Q. Ranzato, M. Monga, R. Devin, M. Chen, K. Corrado, G. Dean, J. and Ng, A.: Building high-level features using large scale unsupervised learning, International Conference in Machine Learning, In ICML (2012)
- [Zeigler 00] Theory of Modeling and Simulation, Second Edition, B.P. Zeigler et al (2000)
- [Takahashi 04] Takahashi, Kaizu, K. Hu, B. and Tomita, M.: A multi-algorithm, multi-timescale method for cell simulation, Bioinformatics, Oxford (2004)
- [山川 98] 山川 宏, 岡田浩之, 渡部信雄, 松尾 和洋.: 実世界で自律学習を行うための分散知能アーキテクチャ -CITTA-, 1998 年度人工知能学会全国大会 (JSAI-98). (1998)
- [Eliasmith 13] Eliasmith, C.: How to Build a Brain: A Neural Architecture for Biological Cognition, Oxford (2013)
- [Sculley 14] Sculley, D. Holt, G. Golovin, D. Davydov, E. Phillips, T. Ebner, D. Chaudhary, V. and Young, M.: Machine Learning: The High Interest Credit Card of Technical Debt, SE4ML: Software Engineering for Machine Learning (NIPS 2014 Workshop), In NIPS, Google (2014)
- [一杉 14] 一杉裕志, 高橋直人: 制限付きベイジアンネット BESOM における認識アルゴリズム OOBP 2014 年度 人工知能学会全国大会 (JSAI-14). (2014)
- [Haruno 01] Haruno, M., Wolpert, D. M., & Kawato, M.: Mosaic model for sensorimotor learning and control, Neural Computation, 13. (2001).