人工知能学会研究会資料
SIG-AGI-001-08

Brain-inspired Computing Architecture (BriCA):

- 全脳アーキテクチャ開発のためのオープンソフトウェアプラットフォーム -

Brain-inspired Computing Architecture (BriCA): an Open Source Platform for WBA Development

板谷 琴音[12*]
Kotone Itaya

髙橋 恒一[123]
Koichi Takahashi

中村 政義[3]
Masayoshi Nakamura

小泉 守義[4]
Moriyoshi Koizumi

荒川 直哉[3]
Naoya Arakawa

冨田 勝[1]
Masaru Tomita

山川 宏[3]
Hiroshi Yamakawa

[1]慶應義塾大学政策・メディア研究科
Keio University Graduate Schol of Media and Governance

[2]理化学研究所生命システム研究センター
RIKEN QBiC

[3]ドワンゴ人工知能研究所
Dwango Artificial Intelligence Laboratory

[4]株式会社オープンコレクター
Open Collector, Inc.

Abstract: The Whole Brain Architecture (WBA) project aims to create an artificial general intelligence with near human capabilities by implementing brain regions using machine learning algorithms and connecting them according to the architecture of the brain. To encourage the community based development of WBA, we are implementing BriCA (Brain-inspired Computing Architecture) which can connect and execute an arbitrary number of machine learning components. In this presentation I will introduce the requirements analysis, concepts, implementation status, and future work for the core features of BriCA.

1.背景

計算論的神経科学の発展に伴い脳の神経科学的特性に基づいた機械学習アルゴリズム開発が盛んに行われるようになってきた．特に異なるパラダイムで動作する機械学習アルゴリズムを複数組み合わせることによってそれぞれ単一のアルゴリズムでは達成できなかった性能や機能が実現出来ることが示されるようになってきており [1-3] 注目を集めつつある．

全脳アーキテクチャ(WBA)プロジェクト[1]では以下の全脳アーキテクチャ中心仮説を設定しこの妥当性を構成論的に検証することを目的としている．

> 「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さ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「脳のモジュール性」，「脳器官は機械学習器か」，「機械学習器の非加算性」という少なくとも3点の必ずしも網羅的ではないが重要な論点を含んでいる．

[1] http://www.sig-agi.org/wba

i) 脳のモジュール性

脳は解剖学的に新皮質，海馬，小脳，基底核，視床などの器官に分かれており器官内でも新皮質内の視覚野や聴覚野，海馬内のCA1・CA3・歯状回などの領野・回路レベルで独立した構造を持っている．WBAは脳器官がモジュラーになっていることを前提にしているが，大脳皮質内のマイクロカラムの数約$2e^8$に対して接続率が$6e^4$程度であり [4]，全ての接続がランダムであれば大多数のマイクロカラムが2ステップで接続される．

ii) 脳器官は機械学習器か

例え脳がモジュラーになっていたとしても，それぞれのパーツのを一つ一つのニューロンの結合を書き下すのではなく機械学習アルゴリズムとして抽象化可能かどうかは自明ではない．また，それぞれの脳器官を表現する機械学習アルゴリズムは全脳アーキテクチャ完成時までに既知になっている必要がある．

iii) 機械学習器の非加算性

また脳全体が発揮する認知機能と同等の機能が機械学習器が構成するネットワークによって再現可能かどうかは自明ではない．二つの異なる機械学習アルゴリ

*連絡先: 慶應義塾大学 政策・メディア研究科
〒252-0882 神奈川県藤沢市遠藤5322
E-mail: kotone@sfc.keio.ac.jp

ズムの組み合わせによって従来解くことのできなかったタスクをこなすことができるという報告は既に数例あり [1-3]，更に多くの機械学習器を組み合わせることによってより高度な機能を獲得できると期待される．

上述のWBA中心仮説を検証するには脳の神経科学的な知見の蓄積，機械学習アルゴリズムの開発，認知アーキテクチャの設計，ソフトウェアの実装など非常に広い範囲での研究活動が欠かせず，そのためオープンな共同開発を促進する環境整備が重要である．より多くの協力者を募るため全脳アーキテクチャのベースとなる接続モデルを提供し，それを自由に編集・共有・実行できるソフトウェアフレームワークを用意する必要がある．そこで，私たちは機械学習アルゴリズムを実装し自由に階層的に組み合わせることのできるプログラミングライブラリ、アーキテクチャ記述のための言語、認知アーキティクチャの学習環境や育成カリキュラムの構成などを含む統合ソフトウエア環境であるBriCA (Brain-inspired Computing Architecture)の開発に着手した．

2.目的

WBA実装のためには複数の異種の機械学習器を同時に動作させ，それらを適切かつ効率的にスケジューリング，同期，通信させる実行機構が必要となる．このような目的に応用できる可能性のある既存のソフトウェアとしてロボットミドルウェアのROS[1]やMIRA[2]，データ解析プラットフォームのWeka[3]，Garuda[4]，Jubatus[5]，シミュレーションソフトウェアのSimulink[6]，LabVIEW[7]，E-Cell [5]，数値演算ライブラリであるTensorFlow[8]などが挙げられる．ロボットミドルウェアは記述言語を提供しておらず，比較的レイテンシが大きい通信を許容する設計になっているためWBAの実装には不向きである．データ解析プラットフォーム，シミュレーションソフトウェア，数値演算ライブラリらは実時間と同期した使用を想定しておらず機能も限定的であるケースが多い．これらのソフトウェアはWBAを構成する要素技術の開発には有用であるものの、全体の統合を行う基盤ソフトウエアが必要である．現在我々は実行機構の必要最低限の要求仕様として以下の八項目を定めている．

a) 任意の新規あるいは既存の機械学習器のモジュールライブラリ化が可能
b) 任意の数の機械学習器を階層的に結合した認知アーキテクチャを構成可能
c) 統一した通信方式で任意の機械学習器間の結合が可能
d) 統一のセンサー・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可能（ロボットOSやゲームエンジンなど他のプラットフォームと結合し動作）
e) 機械学習器の非同期呼び出しを調停するリアルタイムスケジューラを持つ
f) 機械学習器の数Nに対してソフトウェア的，実行性能的にスケーラブル
g) 育成・学習カリキュラムを構築，実行する統一的な機構を持つ
h) コミュニティーによる分散開発に対応可能

BriCAでは個々のニューロンレベルで実装するのではなく，脳器官のレベルでアルゴリズムを選択し利用するため実行時のロードにばらつきが出ることが予想される．そのため同期的な実行では機械学習器が実時間で設定されたインターバルで発火できない可能性があるため非同期に実行できる必要がある．通信についてはメッセージパッシングの採用を検討しているが，この理由付けとして実装上のスケーラビリティと脳の器官間の情報のやり取りは恐らく共有メモリ的になっていないという二点がある．ただし今後黒板モデルのような実装の要求が出てきた場合，それに対応するモジュールを実装することで対応するなどの措置を検討している．

3.実装

BriCA V1はPythonで書かれており，他クラスのベースとなるUnitに加えComponent，Module，Agent，Schedulerを含めた5つのクラスによって主な機能を実現している．

UnitはBriCA V1における脳器官や機械学習器などの不可分な単位を取り扱うベースクラスとして定義されている．入力と出力を制御するためにデータを保持するバッファを内包する入出力ポート群を持っており，任意のUnitの出力ポートへ入力ポートを接続することができる．ポートは接続先のポートのデータを自分の持つバッファにコピーするPull型の同期モデルを採用しており，Unitのポートが持つデータの型は制限されていない．またポートは別のポートに対してエイリアスを形成可能なため，複数のUnitのポートを同一とみなすことも可能である．後述のComponentとModuleはUnitを継承して実装されておりUnitの機能を全て使用できる．

BriCA V1では機械学習アルゴリズムの実装を持つクラスとしてComponentが定義されている．Componentクラスはfire()という抽象メソッドを持ったクラスであり，特殊化の際には本クラスを継承しオ

[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/
[8] http://www.tensorflow.org/

ーバーライドしたfire()メソッド内に機械学習アルゴリズムの動作を記述する．具体的にfire()メソッドは以下の計算を実行する．

```
outputs, states <- fire(inputs, states)
```

実際には複数のComponentが並列に発火するため，fire()メソッドが直接入出力ポート群のバッファの内容を更新することはできないようになっており，スケジューラによって以下の三段階を経て実行される．

```
1. inputs <- In Ports
2. outputs, states <- fire(inputs, states)
3. Out Ports <- outputs
```

上述の1，3のそれぞれの操作はComponentのinput()及びoutput()メンバ関数として定義されており，input()内では先ず全ての入力ポートの値が更新されてからバッファの中身がコピーされる．Componentの設計思想はBriCA V0のModule [6] を踏襲しており，ポートクラスが定義されバッファ自体を持たなくなった点及びポートの型制限が撤廃された点を除いて共通の機構を実装している．

以前にBriCA V0を実装した際は全てを階層性を持たないアーキテクチャとして記述していたが，実際にはそれぞれの脳器官もさらに下位の部位や器官から構成されるなど (大脳皮質の領野等) 階層的に構成されている．そのため複数のComponentをまとめ，階層化するための仕組みとしてModuleをBriCA V1で再定義している．Moduleは内部に任意の数のComponentとModuleを持つことができるがそれ自体は実装を持たない．また，Moduleは脳器官の抽象化を実現するため，ポートが保持できるデータは固定長の符号付き整数(現在の実装では16bit長)を要素とするベクトルに限定される．階層のトップレベルのModuleとしてサブクラスであるAgentが定義されており，内部にスケジューラを持ちstep()メンバ関数を通してスケジューラの更新を行うことができる．

スケジューラはComponentの発火を調停しており，Component毎に設定されたインターバルを元にfire()メソッドを呼び出す．BriCA V1では三種類のスケジューラが実装されており，仮想時間の同期・非同期スケジューラに加え実時間の同期スケジューラが利用できる．

4.結果・考察

BriCA V1ライブラリのソースコード (https://github.com/wbap/BriCA1) とドキュメント (https://wbap.github.com/BriCA1) はApacheライセンスの元で自由に利用できる．最も単純な複合学習器の構成例としてニューラルネットワークのフレームワークであるChainerをBriCAコード内から用いて，手書き数字の分類を行う Stacked Denoising Autoencoder (SDA) を実装した．三つの独立したAutoencoderをそれぞれComponentとしてインスタンス化して結合することで数字認識を行う一つの複合学習器を構成した．

表1にBriCAを用いた実装とChainerのみを用いた実装をそれぞれ100度実行した際の学習結果の検証データに対する精度と実行時間の平均を示した．テストにはMNISTデータセット[1]を用いてDeep Learning TutorialsのSDAの項[2]で紹介されている構成のSDAをIntel(R) Core(TM) i7-5930K CPU @ 3.50 GHz 12コア，メモリ64 GB，GeForce GTX TITAN X搭載のマシン上で20 Epoch学習させた．

表1. 通常の実装とBriCA実装の比較

	Chainer	BriCA V1
精度 (検証データ)	0.965987	0.966094
実行時間 (μsec)	194,354,796	201,845,639

精度はMNISTデータセットのうち学習セットを用いて学習したSDAを用いて検証データのラベル予測を行いその正答率を算出している．実行時間はマイクロ秒単位で計測した．

いずれの場合も検証データに対する予測精度は96.6%程度であり，F検定の結果両者が等分散であることがわかったため(P>>0.05)，StudentのT検定を行ったところ平均値に有意な差は認められなかった(P>>0.05)．実行時間においては分散に有意な差が見られたため (P=0.004593<0.05) WelchのT検定を用いたところ平均値に有意な差が認められた(P=2.2e-16<0.05)．BriCAを用いて複合学習器とした場合の実行時間とChainer単体での実行時間を比較すると、BriCAの通信機構を用いて学習器を分解したことによるオーバーヘッドは3.85%であった．

5.議論

BriCA V1ではComponent間の通信は任意のデータ型を受け渡すことが許されているが，Moduleは固定長の16 bit符号付き整数のベクトルに限定されている．これはModuleが脳の器官の抽象化であるため，Module間の情報のやりとりは本質的には分散表象を用いるべきであり，ニューロンの発火頻度として表現可能であるという前提に立っている．一方で脳器官より抽象度の高いComponentは一般に認知アーキテクチャなどで用いられる文字列や木構造など，アルゴリズムの都合上ベクトルとして表現できない記号表現のやりとりが発生しうるため制限を設けていない．しかし実行機構レベルではデータ型が自由だったとしても

[1] http://yann.lecun.com/exdb/mnist/
[2] http://deeplearning.net/tutorial/SdA.html

DSLなどのより上位の実装で束縛することも可能となるためV2ではModuleの型制限も撤廃することを視野に入れている．

複合学習器の実装に際して，BriCA V1を用いた場合には独自に機械学習アルゴリズムをComponentクラスにラップする必要があるためコードを余分に記述する必要があるが，既に実装済みのComponentを使い回す場合機械学習アルゴリズム間の接続を定義するだけで複雑な複合学習器を構成できる．コード量の厳密な比較はできないがDSLなどの開発に伴いより簡便に記述できるようになると期待される．現在JSONをベースとしたBriCA言語をWBAの最小セット記述と併せて開発中である．

BriCA V2以降に期待される機能として複数の機械学習器の学習タイミングをスケジューリングするためのカリキュラム定義及びその実行がある．今回実装したSDAは非常に単純なモデルだったため全てのComponentを同時に学習させたとしても正しく収束することが既知だったが，全く異なるパラダイムの学習器を接続した場合，接続先の学習結果が接続元の学習結果の影響を受けて変化してしまうことが知られている [7]．脳の発達過程においても視覚野が下位の領野から固定されていくなどの現象が観察されているため，同様の機構をBriCAに作りこんでおく必要がある．

またBriCA V1では外部環境との連携のプロトタイプとしてROS及びGazebo[1]との接続インターフェースを実装したが，今後は他にもUnity[2]，Irrlicht Engine[3]，cocos2D[4]などに代表されるゲームエンジン，データ解析プラットフォームであるJubatas，Garudaなどとの連携も実装する必要がある．

BriCAを用いた様々な認知アーキテクチャ実装を促進するために既存のアルゴリズムや機械学習フレームワークを用いたComponentのライブラリをある程度実装しておく必要がある．代表的な汎用ソフトウェアライブラリとしてはChainer，TensorFlow，scikit-learn，PyBrain，Theano，PyLearn2，Apache Spark，Weka，MALLET，Dlib-ML，shogunなどがある．特化した機能を提供するライブラリにはRatSLAM，RT-SLAM，LSD-SLAMなどの海馬のSLAM (Simultaneous Localization and Mapping) 機能を提供するものや，大脳新皮質のモデルBESOM [8]，小脳モデルであるMOSAIC [9]などが存在する．BriCA上でのこれらの実装は当然，新規の機械学習アルゴリズムの実装，評価，共有をコミュニティーベースで促進する必要がある．

実証コードとしての利用を主眼としてPythonで実装されているBriCA V1だが，現在は通信のオーバーヘッドが大きいため最終的なWBAの実装に向けたライブラリとしては不適切である．Javaで実装されたBriCA V0ではペイロードなしで通信遅延が100 nsec程度であったのに対しV1では10 µsec程度まで膨らんでいるが，これは動的型付けを行うPythonインタプリタの影響によるものと考えられる．ニューロンの発火頻度の上限1 kHzに対して許容できる遅延を1%の10 µsecとするならば，現在のPythonによる実装はペイロードを乗せた場合に許容値を下回ることができない．そのためV2以降ではC++などの言語の採用によって性能の改善を図る．ただし機械学習ライブラリは様々な言語で実装されているため，核となる機能群を提供するカーネルをC++で実装し，Componentなどの実装のためには複数の言語インターフェースを提供する事を予定している．

[1] http://gazebosim.org/

[2] http://unity3d.com/

[3] http://irrlicht.sourceforge.net/

[4] http://jp.cocos2d-x.org

参考文献

[1] Vinyals, O. Toshev, A. Bengio, S. and Erhan, D.: Show and Tell: A Neural Image Caption Generator, arXiv, 1411.4555 (2014)

[2] Karpathy, A. Fei-fei, L.: Deep Visual-Semantic Alignments for Generating Image Descriptions, arXiv, 1412.2306 (2014)

[3] Mnih, V. Kavukcuoglu, K. Sliver, D. Rusu, AA. Veness, J. Bellmare, MG. Graves, A. Riedmiller, M. FidjeLand, AK. Ostrovski, G. Petersen, S. Beattle, 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)

[4] 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)

[5] Takahashi, Kaizu, K. Hu, B. and Tomita, M.: A multi-algorithm, multi-timescale method for cell simulation, Bioinformatics, Oxford (2004)

[6] 髙橋恒一, 板谷琴音, 中村政義, 小泉守義, 荒川直哉, 冨田勝, 山川宏.: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, 2015年度人工知能学会全国大会 (JSAI-15). (2015)

[7] 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)

[8] 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)

[9] Haruno, M., Wolpert, D. M., & Kawato, M.: Mosaic model for sensorimotor learning and control, Neural Computation, 13. (2001).