

全脳アーキテクチャを支える プラットフォーム

理化学研究所 生命システム研究センター チームリーダー

高橋恒一

twitter @ktakahashi74

慶應義塾大学 政策・メディア研究科
ドワンゴ人工知能研究所
全脳アーキテクチャ・イニシアチブ

特任准教授
客員研究員
理事・副代表

全脳アーキテクチャまで

15年

>

3-5年

企業、公的機関のプロジェクトの寿命

< 25年

研究者の1世代

NPO全脳アーキテクチャ・イニシアチブ (WBAI)

認証申請中

●目的

- 今後15年以上に渡るWBA研究開発を公益的な視点から支援する。
- 「全脳アーキテクチャ研究開発の伴走者」

●活動の基本方針

- 長期的発展： 15年以上先の汎用人工知能の完成まで
- 人材育成： シニアから若手中堅への伝承
次の勝負どころに備える
- 基礎研究： 標準的な基盤となるプラットフォーム構築
- 公益性： 研究成果、ソフトウェアなどをオープン化

●主な公益事業

- 勉強会開催、ハッカソン運営、若手支援
- 基盤技術開発、社会への影響検討（検討中）、技術認定制度（検討中）等

NPO全脳アーキテクチャ・イニシアチブ (WBAI)

認証申請中

運営体制

(全脳アーキ主要メンバーで分担)

総会

理事会

運営委員会

人材育成委員会

研究戦略委員会

広報委員会

管理委員会

若手の会

理事

代表： 山川宏

副代表： 松尾豊

副代表： 高橋恒一

幹事： 佐藤健

入会のご案内

会員：

個人が対象

入会金無料, 年会費1000円

後日募集開始

賛助会員：

一口8万円/年より

特別賛助会員も設定

※受付中

事務局：ドワンゴ人工知能研究所内に設置

連絡先：info@wbai.jp

ウェブページを準備中(<http://wbai.jp>)

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.

to appear, JSAI15.

脳型計算基盤ソフトウェア
Brain-inspired Computing Architecture
BriCA

全脳アーキテクチャ(WBA)の中心仮説

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事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事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。

全脳アーキテクチャ中心仮説の構成論的な検証

サブ仮説群（必ずしも網羅的でない）

1. 脳のモジュール性
2. 脳器官は機械学習器か
3. 機械学習器の非加算性

1. 脳のモジュール性

- 大脳皮質マイクロカラムはランダム結合なら2ステップで大多数カラム間が接続。
- その一方で、脳には様々な解剖学的独立単位が観察される。

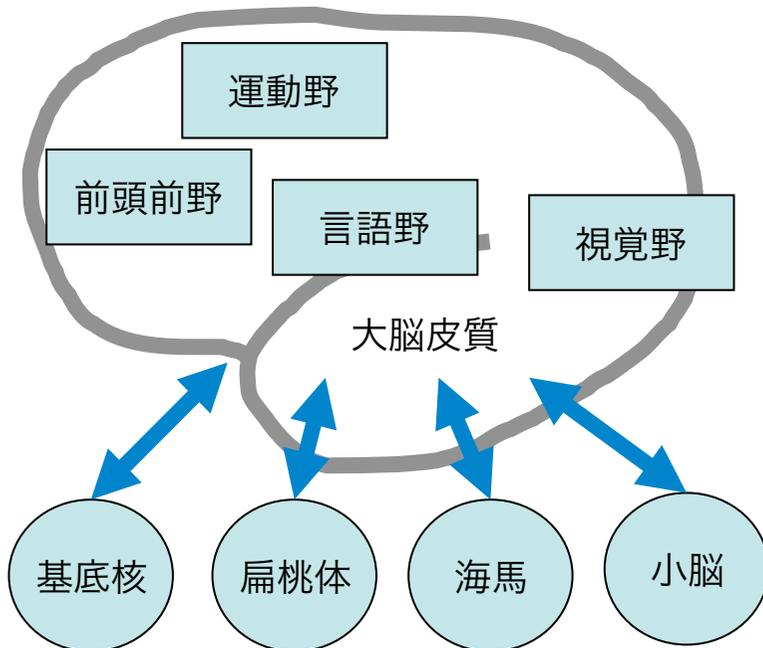


Table 2
Number of minicolumns and hypercolumns in cortex

	Minicolumns	Hypercolumns
Human	2×10^8	2×10^6
Macaque	2×10^7	2×10^5
Cat	6×10^6	6×10^4
Rat	5×10^5	5×10^3
Mouse	1.6×10^5	1.6×10^3

The figures are computed from the data in Table 1.

Table 3
Number of connections and connectivity in cortex

	Corticocortical connections	Connectivity
Human	2.4×10^{13}	6×10^{-4}
Macaque	3.6×10^{12}	4×10^{-3}
Cat	7.2×10^{11}	2×10^{-2}
Rat	6.0×10^{10}	0.24
Mouse	2.4×10^{10}	0.94

The connections are between minicolumns and the level of connectivity is computed as the fraction of full connectivity between the minicolumns. The majority of all synapses in a minicolumn are devoted to corticocortical connections, and each connection is on average supported by five synapses.

Johanson and Lansner, *Neural Networks*, 2007.

2. 脳器官は機械学習器か

- 脳のモジュール一つひとつが「よく定義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は自明ではない。
- 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
VS
計算論的に一貫性のある単一の原理で動作を定義
- 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。

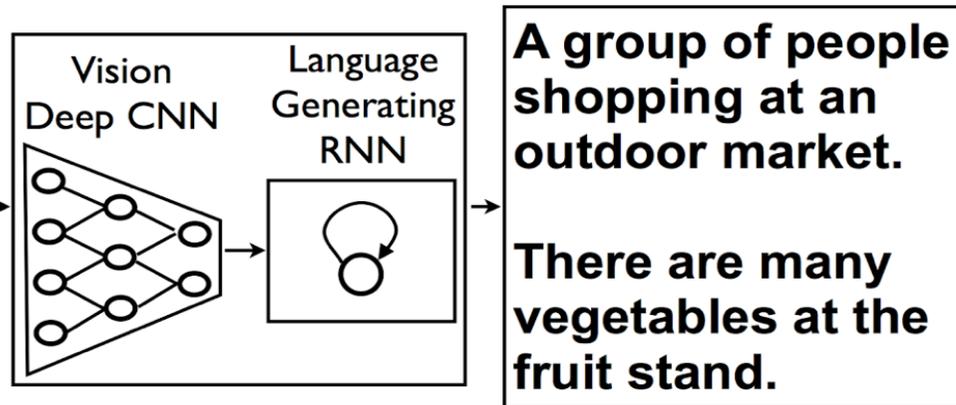
全脳アーキテクチャ俯瞰図（たたき台）

統合 ↑ ↓	すべての機能を統合するモジュール (全脳アーキテクチャモデル)				
	滑らかな運動 モジュール (皮質・小脳連 携モデル)	言語理解・発話モジュール (言語野モデル)			情動モジ ュール (扁桃体モ デル)
		思考・ナビゲーションモジュール (皮質・基底核・海馬連携モデル)			
		階層型強化学習モジュール (皮質・基底核連携モデル)			
	教師あり学習 モジュール (小脳モデル)	強化学習モジ ュール (大脳基底核 モデル)	教師なし学習・認 識モジュール (大脳皮質 モデル)	エピソード記憶 モジュール (海馬モデル)	
パーセプトロン、 リキッドステー トマシン	強化学習	Deep learning, SOM, ベイジアン ネット	自己連想ネット ワーク、直交符 号化	連合学習	

部品

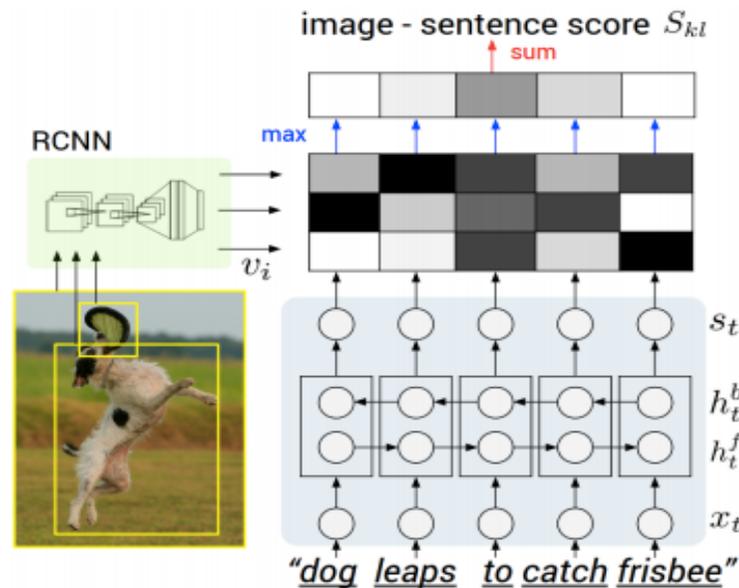
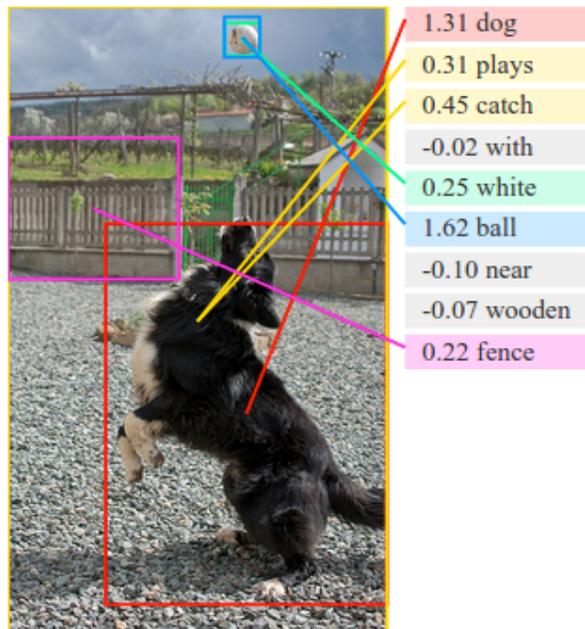
3. 機械学習器の非加算性

- 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。
- 神経回路のレベルでは、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に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も考えることもできる。
- $1+1 > 2$ (非可算的であっても $1+1 < 2$ ではだめ)
- -> 複合学習問題



GoogLeNet team
google.com

キャプション生成：
どちらもほぼ同じ機械学習アーキテクチャー；
convolutional neural net + long short-term memory neural net.



Neural Talk
cs.stanford.edu

Deep Q-Network

Human-level control through deep reinforcement learning

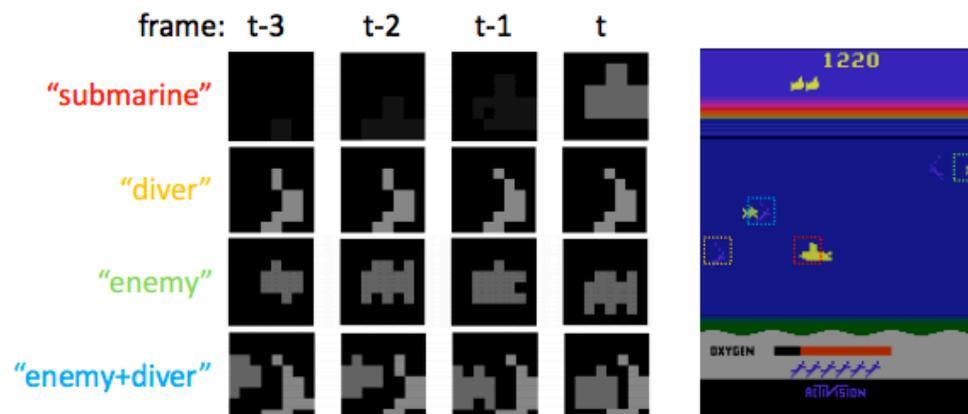
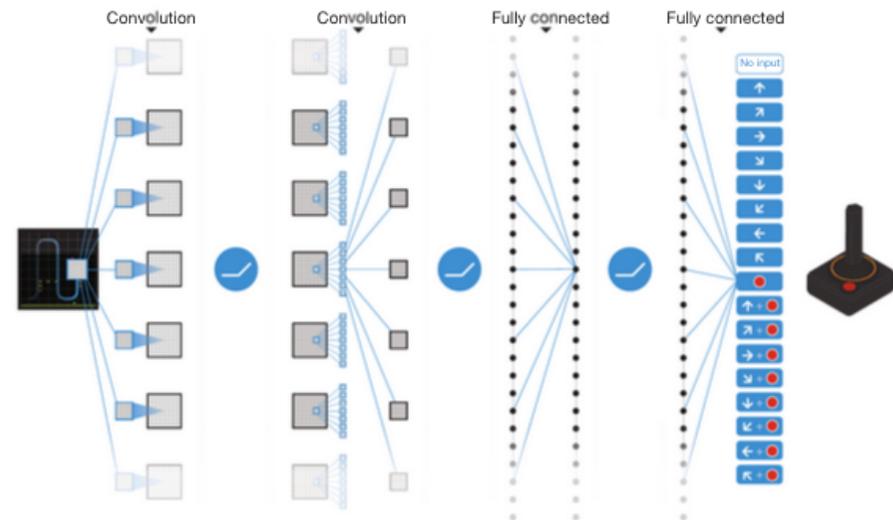
Volodymyr Mnih^{1*}, Koray Kavukcuoglu^{1*}, David Silver^{1*}, Andrei A. Rusu¹, Joel Veness¹, Marc G. Bellemare¹, Alex Graves¹, Martin Riedmiller¹, Andreas K. Fiedjeland¹, Georg Ostrovski¹, Stig Petersen¹, Charles Beattie¹, Amir Sadik¹, Ioannis Antonoglou¹, Helen King¹, Dharshan Kumaran¹, Daan Wierstra¹, Shane Legg¹ & Demis Hassabis¹

Deep belief network上に強化学習を載せている。

Q-function: $Q(s, a)$ (model free)

optimizing action a by predicting future system state s' .

$T(s, a, s')$ (model-based)



複合学習器の構成方法

1 Composition

互いに独立な機械学習器を2つ以上結合し、全体としてより大きな（より高性能、高機能な）機械学習器を構成する。

例：NeuralTalk

→ プラットフォームへの要求：モジュール階層性

2 Hybrid

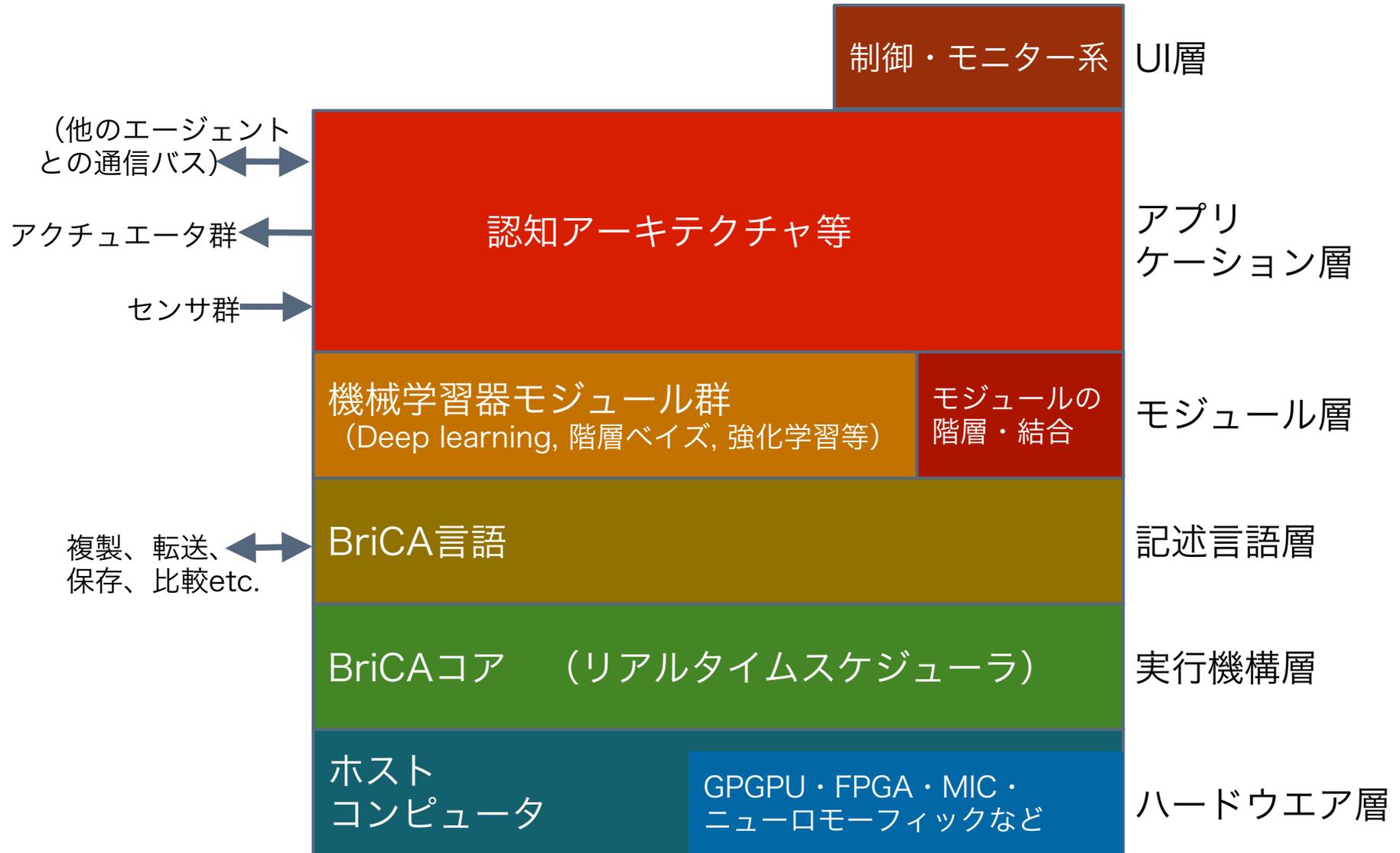
2つ以上の機械学習アルゴリズムを組み合わせて新たな機械学習アルゴリズムを構成する。

例： Deep Q-Network

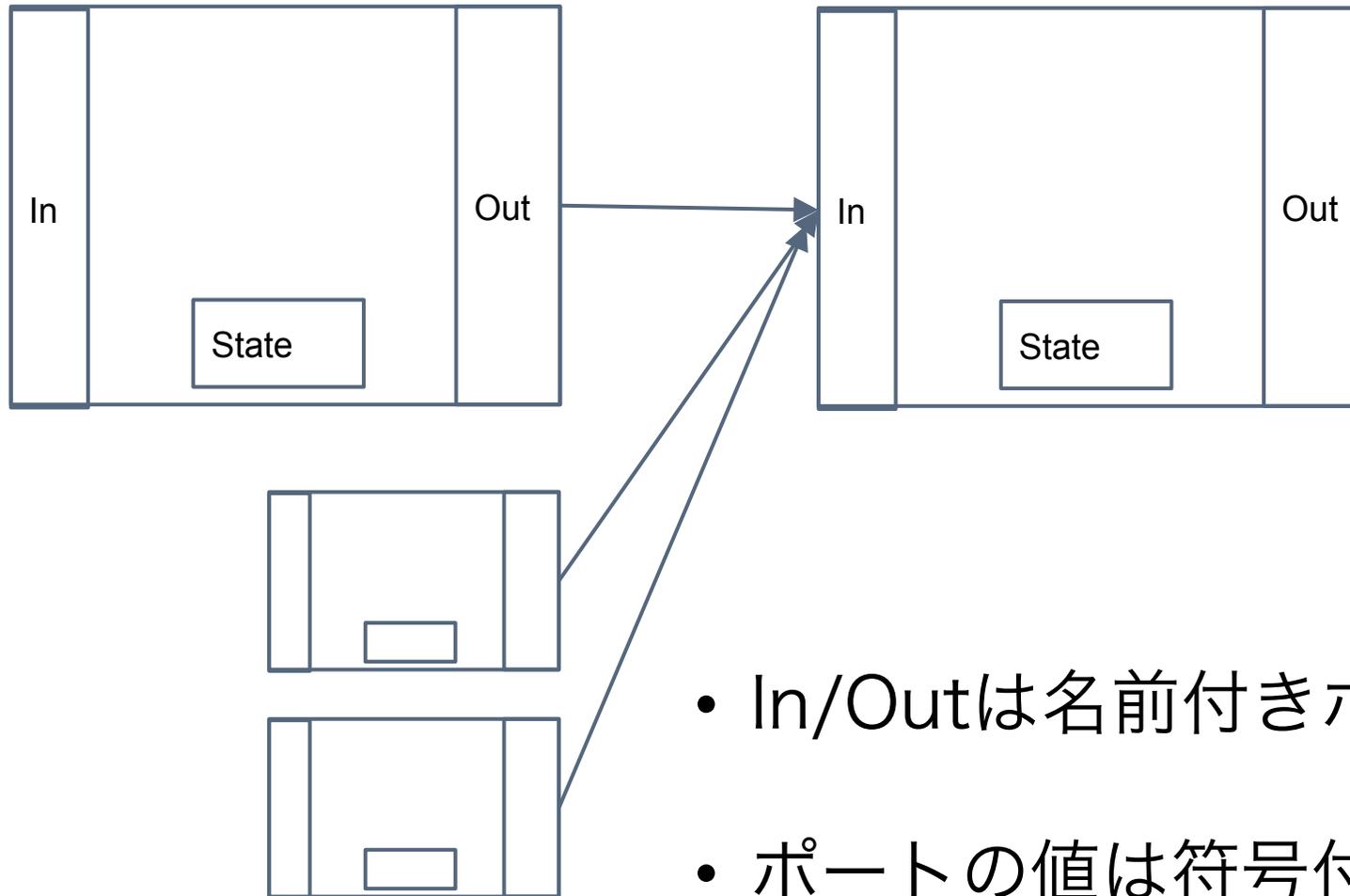
BriCA 要求仕様の概要

1. 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
2. 任意の数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
3. 統一した通信方式で任意の機械学習器間の結合が可能
4. 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)
5. 機械学習器の非同期呼出しを調停するリアルタイムスケジューラーを持つ。
6. 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。
7. 複合学習問題（後述）に対処するため、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。
8. コミュニティによる分散開発に対応可能

BriCAの6層構造



BriCAモジュール



- In/Outは名前付きポート。
- ポートの値は符号付き16bit整数のベクトル。

モジュールの発火

モジュールの計算（発火）は以下の操作を実行する。

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

実際には、モジュールの並行性のため、3段階のフェーズで行われる。

1. input 入力バッファinの値を更新
2. fire results, states <- fire(in, states)
3. output out <- results

モジュール発火の並行性

モジュール発火の並列実行を安全に行うためにはモジュールのI/Oポートの同期において下記2つの要求を満たす必要がある。

1. ある時点でモジュールAが自分の入力として取得したモジュールBの出力値を、モジュールAが計算を行っている間、仮にモジュールBが出力値を更新した後も安全に保持したい。

→ ポートの値の非同期転送で対処

2. モジュールの計算過程が外に見えないようにしたい。（出力ポートの値を計算が終わった後のある時点でinstantaneousに更新したい）。

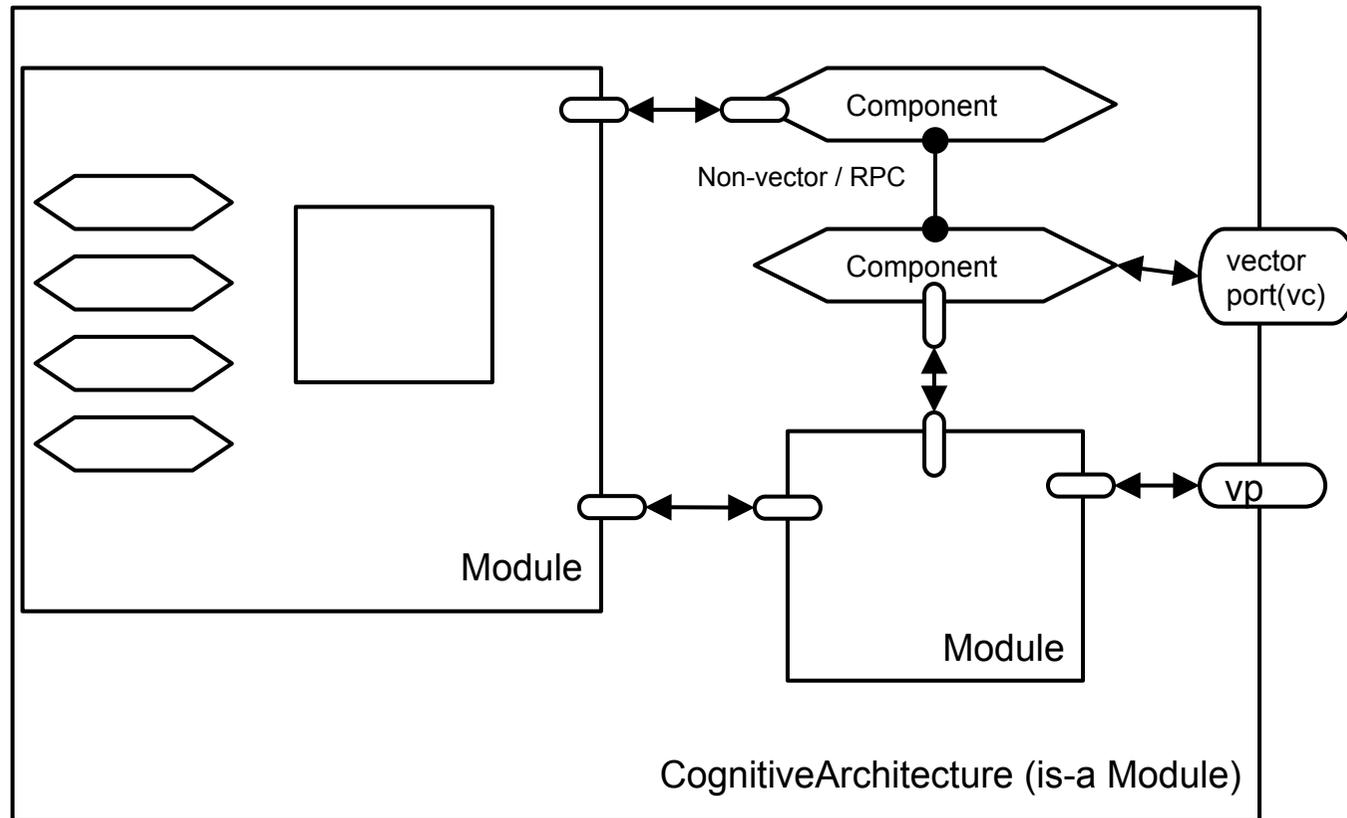
→ ダブルバッファリング (DB) やcopy-on-write等のテクニックで対処
(現在の実装ではDB)

分散表象とベクトル表現

- もし強い意味での脳のモジュール性仮説が成り立つならば、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。
- 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でスタックや木構造などのグラフ構造を扱うことが求められる。

→ 詳しくは後半の市瀬さんの講演で議論

モジュール階層



2つの部品単位で全ての構造を構築。

1. Module: ベクトルポートのみ。RPCなし。階層性あり。
2. Component: 非ベクトルポートあり。RPCあり。階層性なし。

BriCA: 性能に関する要求

(遅延)

- ニューロンの発火頻度の上限は1kHz
- 通信機構はモジュール数Nに依存せず1msを基準に無視できる程度を要求 (仮に10 μ s)
- プロトタイプではペイロード無し時の片道通信遅延は100 ns程度をマーク。
(プロセス間通信を行う場合、分散系 (ロボットOS等) では一般に数十から数百 μ 秒)

(スケーラビリティ)

- ヒトの脳の計算量は1 PFLOPS程度
- コアあたりのFLOPS値を10 GFLOPSとすると10万コア程度の集積が必要
- 10万並列で任意のスレッド間通信での遅延を10 μ s以下に保証：
 - HPC分野の大規模計算の水準。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た設計開発が必要。

(バンド幅)

検討中。

バージョン3の法則

人類は3つの種類のソフトウェアしか作れない。
というソフトウェア工学の経験則。

バージョン1：

プロトタイプ。最小限の機能。大抵短期間で完成。それなりに成功。

バージョン2：

「正しく」設計し「汎用」「拡張可能」なものを目指す。
何かは出来るが、開発期間長期化、肥大化、使いにくい。

バージョン3：

バージョン2の経験で、どこが肝でどこで手を抜いていいかわかる。
軽くて使いやすく長持ちするものが完成！

バージョン4：

そんなものはない。（あったとしたらそれは別の何か）

バージョン進行の予定

- V0: 2014/H2 - 2015/3
 - 目的： 要求仕様の模索
 - モジュール設計検証、スケジューラ設計検証
 - Java
- V1: 2015/4 -
 - 目的： プロトタイプ、非加算性仮説検証（3つ以上）。
 - Python
 - ハッカソン等での迅速なプロトタイピングが重要要求。
 - 性能はそこそこ
- V2: 2016- ?
 - 目的： きちんと作り込む。
 - 言語等は未定（C++、Python、Julia、Java/Scala, Google Go等の多言語対応にしたい）
- V3: 2019以降？
 - AGIの完成を目指す

第1回 全脳アーキテク チャキックオフ ハッカソン



2015年9月19日午前から
23日夕方（夜打ち上げあり）まで5日間
（遠方の場合18日からの前泊も応相談）

予定場所：慶應義塾大学日吉キャンパス 協生館

アイディアソン枠を既にエントリー募集中。

参加方法等詳細はWBA若手の会facebookグループで