

海馬・扁桃体・前頭前野の機能を統合した組み込み指向型脳型人工知能モデルの開発

著者	田中 悠一郎
その他のタイトル	Development of brain-inspired artificial intelligence model with functions of hippocampus, amygdala, and prefrontal cortex for embedded systems
学位授与年度	令和2年度
学位授与番号	17104甲生工第402号
URL	http://hdl.handle.net/10228/00008356

2020年度 博士学位論文

海馬・扁桃体・前頭前野の機能を統合した
組込み指向型脳型人工知能モデルの開発

九州工業大学大学院生命体工学研究科

学籍番号 18899011

田中悠一郎

目次

第 1 章	はじめに	1
1.1	研究背景	1
1.1.1	現行の AI 技術	1
1.1.2	現行の AI 技術とヒトの知能	2
1.1.3	サービスロボットと AI 技術	2
1.2	研究目的	4
1.3	論文構成	5
第 2 章	関連研究	7
2.1	ホームサービスロボット	7
2.2	ニューロンモデル	7
2.3	ハードウェア	8
2.4	本研究のアプローチ	10
第 3 章	個人の経験に基づいた知識の獲得	13
3.1	本章の概要	13
3.2	扁桃体	13
3.3	扁桃体モデルの先行研究	15
3.4	提案モデル	17
3.4.1	モデル構成	17
3.4.2	アルゴリズム	19
3.4.3	ハードウェア	20
3.5	実験	27
3.5.1	アルゴリズム	27

3.5.2	ハードウェア	31
3.6	本章のまとめ	38
第 4 章	個人の経験に基づいた記憶の獲得及び記憶に基づいた予測	39
4.1	本章の概要	39
4.2	海馬・前頭前野	39
4.2.1	海馬	39
4.2.2	前頭前野	40
4.3	海馬・前頭前野モデルの先行研究	40
4.3.1	海馬モデル	40
4.3.2	前頭前野モデル	41
4.4	提案モデル	42
4.4.1	海馬モデル	42
4.4.2	前頭前野モデル	44
4.4.3	三部位モデルの統合	44
4.5	実験	46
4.5.1	他者の移動経路を学習・予測するモデル	46
4.5.2	自身の移動経路と価値を学習・予測し行動決定するモデル	49
4.6	本章のまとめ	51
第 5 章	マルチタスク学習への適用	55
5.1	本章の概要	55
5.2	マルチタスク学習	55
5.3	マルチタスク学習の先行研究	56
5.4	提案モデル	56
5.4.1	モデル構成	57
5.4.2	アルゴリズム	57
5.4.3	海馬モデルとの統合	59
5.5	実験	60
5.5.1	RC-SOM	60
5.5.2	RC-SOM と海馬モデルの統合	64
5.6	本章のまとめ	67

第 6 章	おわりに	69
6.1	まとめ	69
6.2	議論と今後の展望	70
6.2.1	個人の経験に基づいた知識の獲得	70
6.2.2	個人の経験に基づいた記憶の獲得及び記憶に基づいた予測	72
6.2.3	マルチタスク学習への適用	74
	参考文献	77

第 1 章

はじめに

1.1 研究背景

1.1.1 現行の AI 技術

2006 年に Hinton らが提案した Deep Brief Network[1] の登場を皮切りに，現行の人工知能（Artificial Intelligence : AI）技術の中心は深層学習（Deep Learning : DL）となっている．特に，2012 年に開催された画像認識コンペティション ImageNet Large Scale Visual Recognition Challenge[2] において，ディープニューラルネットワーク（Deep Neural Network : DNN）による手法 [3] が，他の手法に圧倒的な差をつけて優勝したことが大きなブレイクスルーとなった．DL は画像認識のみならず，画像生成 [4, 5] や，音声認識 [6, 7, 8]，強化学習 [9, 10] など様々なタスクで高い成果をあげている．

かつて，1950 年代頃に起こった第 1 次 AI ブームや，1980 年代頃に起こった第 2 次 AI ブームが程なくして衰退していった一方で，近年の DL による第 3 次 AI ブームが成功を収めたのは，それを取り巻く環境の変化によるものが大きい．ひとつは，インターネットのデータの転送量の増大によりビッグデータが手に入りやすくなったことである．これにより DNN は，大量の学習データの中から，タスクを解くために必要な特徴を自動的に取り出せるようになった．もうひとつの理由は，計算機の能力の向上により，膨大な構造を持つニューラルネットワークの学習を現実的な時間で実行できるようになったことである．特に，Graphics Processing Unit（GPU）による演算の並列化は，DL の実行を大きく加速することができる．

1.1.2 現行の AI 技術とヒトの知能

DL は様々なタスクにおいて高い性能を発揮し、特に画像認識の性能は、ヒトをも超えると言われる（画像分類のタスクにおいて、DNN によるエラー率が、ヒトによるエラー率 5.1% を下回る）[11]。しかしながら、DL を中心とする現行の AI 技術のみで、ヒトのように考える知能を実現するのは困難である。例えば、我々ヒトは家族や友人から「アレを持ってきて」と言われれば、「アレ」が指すものを推定して持ってくることができる。これは、過去に自分自身が経験した数回のエピソードから、「アレ」とは何かという知識を獲得しているからである。こうした個人の経験に基づいた知識の獲得は、少量のデータの学習によって行われるもので、大量のデータを必要とする DL によるアプローチでは困難である。

ヒトの脳の学習には大きく分けて 2 つのパラダイムがあると考えられる。ひとつは、膨大な経験から得られる一般的な知識の獲得である。これは例えば「これは机か」や「これは椅子か」など、子供の頃から長い時間をかけて獲得する、あらゆる環境で用いる知識である。このような知識の獲得は、DL によるビッグデータを用いた学習に相当すると考えられる。もうひとつのパラダイムは、数少ない経験から得られる個人固有の知識の獲得である。ここでは、個人が経験したエピソード記憶など、特定の環境で用いられる知識を獲得する。前述の「アレとは何か」という知識の獲得は、後者のパラダイムに分類される。DL の性能はそのデータ量に依存するため、後者の学習パラダイムに DL を適用し、性能を発揮させることは困難である。

ここで重要なことは、DL が後者のパラダイムの学習に不向きであるのは、DL の技術に発展の余地があるためというよりも、そもそも学習する対象が異なるためということである。すなわち、現行の AI 技術とヒトの知能の間には、実現されていない学習パラダイムが存在するため、ヒトのような知能の実現には、この両方のパラダイムを考慮する必要がある。

1.1.3 サービスロボットと AI 技術

近年、少子高齢化による労働力人口の減少を背景に、サービスロボットの需要が高まっている。経済産業省、新エネルギー・産業技術総合開発機構（NEDO）のロボットの将来市場予測 [12] によると、サービス分野のロボットの市場規模は 2015 年の約 3700 億円から、2035 年の約 4 兆 9000 億円と、約 13 倍の成長が予測されている。

サービスロボットは家庭やレストランなどの店舗において、人と共存・協働するロボットである。将来的には、ロボットが家族の一員として、あるいは同僚として生活することが期待される。こうしたロボットの実現には、ヒトのように考える知能の実現が必要である。

サービスロボットに求められる機能は多岐にわたり、様々なタスクが連続して与えられることが想定される。したがって、ただひとつのタスクをこなす AI ではなく、様々なタスクを単一のモデルでこなす汎用的な AI が求められる。これはマルチタスク学習 [13] と呼ばれる課題である。マルチタスク学習では、過去に与えられたタスクによって獲得したパラメータが、新たに与えられたタスクによって上書きされることで、複数のタスクを両立して達成できなくなる、破滅的忘却 (Catastrophic Forgetting) の問題が生じることがある。マルチタスク学習を行うためには、破滅的忘却を回避する AI モデルが必要となる。また、連続して与えられる複数のタスクをこなすにあたり、全てのタスクを 1 から学習するのではなく、過去に学習したタスクによって得られた知識を活用することで、効率の良い学習を行うことができる。以上をまとめると、サービスロボットに搭載される AI モデルは、複数のタスクを破滅的忘却を避けながら学習し、学習済みのパラメータを新たなタスクをこなすために活用する機能が求められる。

サービスロボットへの応用を考える場合、AI のモデルだけでなく、それを実行するハードウェアも考慮する必要がある。サービスロボットのシステムでは、産業用ロボットとは異なり、計算資源や消費電力に限られるが、画像認識・音声認識・音声合成・マニピュレーション・ナビゲーションなど、多くのアプリケーションを同時に実行する必要がある。このため、ロボットのシステムをリアルタイムに動作させるためには、演算の高速化が求められる。高速化のひとつの方法は、ロボットに GPU を搭載し、Central Processing Unit (CPU) に集中する演算をオフロードすることである。GPU による並列演算は DL などの高負荷な演算を高速化するが、消費電力が高く、電力に限られるサービスロボットに適用するのは難しい。高速化の別の方法として、高負荷な演算をエッジではなくクラウドで実行する方法がある。この場合、インターネットの接続の問題や、データのプライバシーの問題がある。特に、サービスロボットが獲得したローカルな知識はプライバシー性が高く、クラウドで取り扱う場合、プライバシーの侵害が懸念される。

上記の問題を解決するために、演算のハードウェア化が効果的である。現行のノイマン型コンピュータでは、ハードウェアはプロセッサとメモリに分かれており、その

振る舞いはソフトウェアによって定義される。構成は汎用的ではあるが、プロセッサ・メモリ間のデータの頻繁なやりとりが必要であり、効率が低い。演算のハードウェア化では、特定の演算の専用アーキテクチャを構成し、その振る舞いをハードウェアによって直接定義する。これにより、ノイマン型コンピュータよりも高効率な演算を実現することができる。この高効率な演算を行うハードウェアを、CPUよりも動作周波数が低いデバイスに実装することで、ソフトウェアよりも高速かつ低消費電力なシステムを設計することが可能である。例えば、中原ら [14] は、畳み込みニューラルネットワーク [15] の専用ハードウェアを設計し、Field Programmable Gate Array (FPGA) に搭載することで、組込み GPU よりも高速で低消費電力であることを示した。また、ムーアの法則の限界のため、現行のコンピュータの性能の向上はいずれ頭打ちとなるとされており、現行の計算機の性能の限界という面からも、ハードウェア化による高速化は求められている。

1.2 研究目的

本研究は、サービスロボットの实用化のため、ヒトのように考える AI モデルの実現を目指す。1.1 で述べたように、上記 AI モデルの実現は、現行の DL 中心の AI 技術の単なる延長では困難である。それは、DL が脳の機能の一部を模倣しているにすぎず、実現されていない学習パラダイムがあるためである。本研究は、生物の脳全体の機能を模倣することで、ヒトのように考える AI モデルを設計するアプローチをとる。特に、現行の AI では実現されていない学習パラダイムに着目し、個人の経験に基づいた記憶の獲得、および記憶に基づいた予測・行動生成を行うモデルを考案する。具体的には、エピソード記憶に関わるとされる海馬・扁桃体、予測や行動生成に関わるとされる前頭前野の機能に着目し、これらの機能を備えた脳部位モデルを実装する。ここでは、提案する AI モデルが数回の経験により、個人の経験に基づいた知識を獲得できるようになることを目指す。

提案する AI モデルは人工ニューラルネットワークによる実装を行い、厳密な神経細胞の模倣（すなわちスパイクニューラルネットワークによる実装）は目的としない。これは、ロボットアプリケーションとして機能することを最優先とするためであり、ロボットのシステムに親和性の高い AI モデルを実装する。

提案する AI モデルは単一のタスクだけでなく、複数のタスクが連続して与えられるマルチタスク学習に対応できることを目指す。このため、前述の AI モデルに破滅

的忘却を回避する仕組みを導入し、さらに、過去に学習したタスクによって得られた知識を、別のタスクで活用する仕組みを導入する。

システムの高速度化・省電力化のため、提案する AI モデルの専用ハードウェアを設計することを目指す。ハードウェアはデジタル方式とし、FPGA をターゲットデバイスとする。ここでは、FPGA に実装されたハードウェアが、ソフトウェア (CPU や GPU) よりも高速に動作することを目指す。

1.3 論文構成

本研究は、大きく以下の 3 つのステップに分けられる。

1. 個人の経験に基づいた知識の獲得を行う AI モデルの考案
2. 上記の AI モデルを拡張し、個人の経験に基づいた記憶の獲得、及び記憶に基づいた予測を行う手法の考案
3. 上記の AI モデルを拡張し、破滅的忘却を回避しながら複数のタスクを学習する手法の考案

ステップ 1 では、扁桃体の機能に着目し、個人の経験に基づいた知識の獲得を行う。ステップ 2 では、扁桃体に加えて、海馬・前頭前野を統合し、個人の経験に基づいた記憶の獲得及び記憶に基づいた予測を行う。ステップ 3 では、前頭前野モデルを拡張し、破滅的忘却を回避しながら複数のタスクを学習する。さらに、海馬モデルとの統合も行う。

以上を踏まえて、本論文は図 1.1 に示す 6 章から構成する。第 1 章では、研究背景と研究目的を述べる。第 2 章では、本研究に関連する理論・技術を述べ、本研究の立ち位置を明らかにする。第 3 章では、個人の経験に基づいた知識の獲得を行う AI モデルについて述べる。第 4 章では、個人の経験に基づいた記憶の獲得・記憶に基づいた予測を行う AI モデルについて述べる。第 5 章では、提案する AI モデルが複数のタスクを学習するための手法について述べる。第 6 章では、提案する AI モデルを議論し、本論文をまとめる。特に、第 3 章、4 章、5 章では、その目標に関する先行研究、提案手法、実験について述べる。

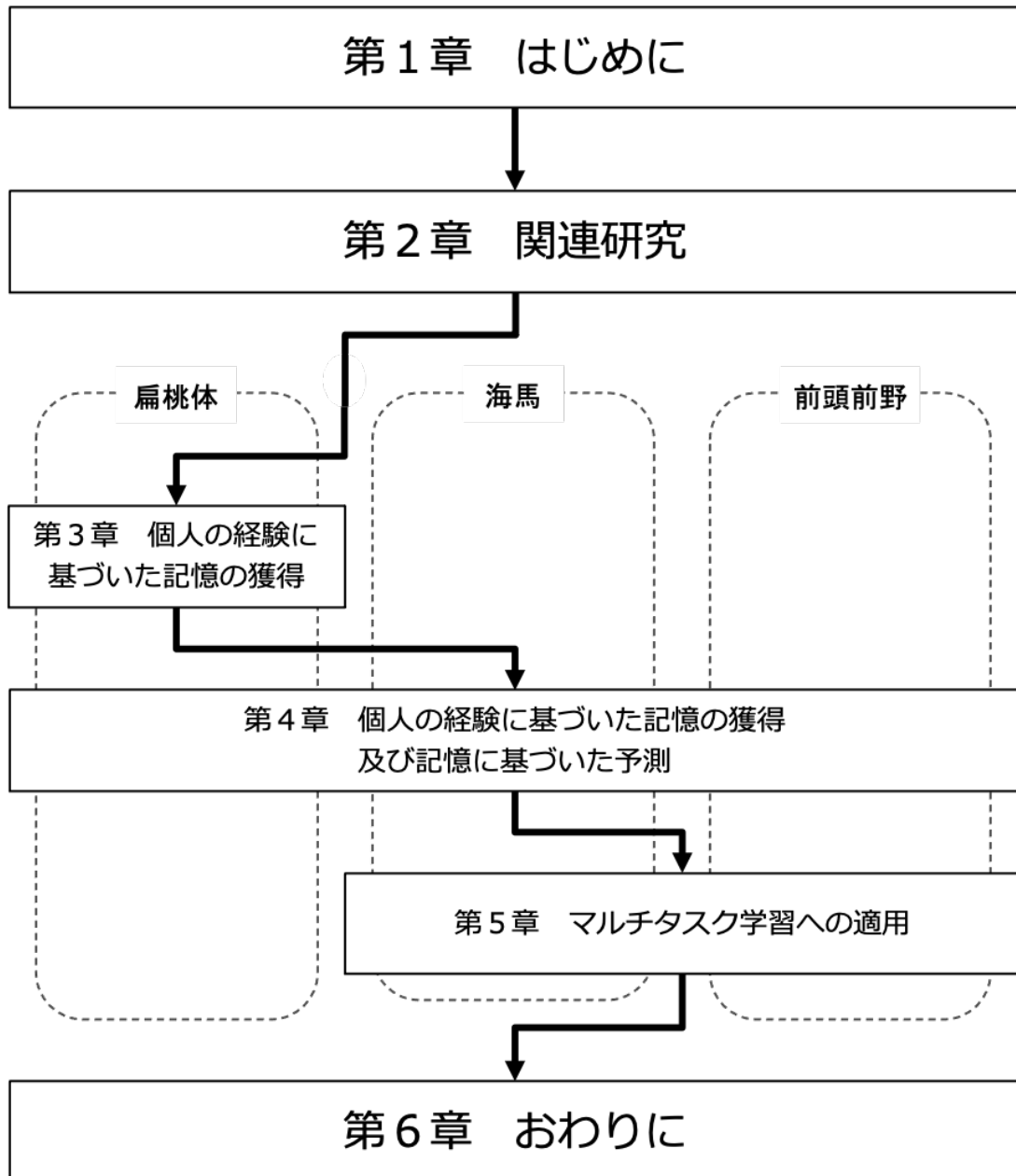


図 1.1: 論文構成

第 2 章

関連研究

本章では、本研究に関連する理論・技術について述べ、アプローチの立ち位置について明らかにする。1 章で述べたように、本研究は提案する AI モデルをサービスロボットに応用するために、高速かつ低消費電力なハードウェアに実装することを目指す。このためには、どのような実装方法をとるべきか議論する。

2.1 ホームサービスロボット

ホームサービスロボットは家庭環境において、人と共存・協働するロボットである。将来的には、ロボットが家族の一員として生活することが期待される。

図 2.1 に、ホームサービスロボット TOYOTA Human Support Robot (HSR) [16] を示す。ホームサービスロボットは、センサとして物体を認識するためのカメラ、Laser Range Finder、音声を認識するためのマイクロフォンなどを備え、アクチュエータとして、物体を操作するためのマニピュレータ、移動するための台車などを備える。ホームサービスロボットは多くの産業用ロボットとは異なり、バッテリーで駆動するため電力に限られる。また、搭載される計算資源も限られる。

2.2 ニューロンモデル

ニューラルネットワークの実装は大きく、スパイクニューラルネットワークと人工ニューラルネットワークの 2 種類に分けられる。

生物の脳の神経細胞は、スパイクを用いて情報を表現することが知られている。ス



図 2.1: ホームサービスロボット : TOYOTA Human Support Robot[16]

パイキングニューラルネットワークは、これを直接模倣したモデルである。積分発火ニューロンモデルはその一例である。このモデルでは、内部状態がスパイク入力とその結合荷重に応じて変化し、内部状態がある閾値を超えたタイミングでスパイクを発生させる。すなわち、このモデルでは、スパイクのタイミングが情報を表現している。

一方で、人工ニューラルネットワークは、神経細胞の機能を抽象化したモデルで、情報をスパイクのタイミングではなく、発火頻度を示す値で表現する。基本的な人工ニューラルネットワークモデルは、入力ベクトルと結合荷重ベクトルの内積に非線形関数を適用したものが出力となる。

2.3 ハードウェア

CPU に代わる演算のアクセラレータとして、GPU や FPGA などが挙げられる。本節では、GPU と FPGA の特性を比較する。また、Application Specific Integrated Circuit (ASIC) についても言及する。

表 2.1 に、速度・消費電力・開発工数に関する、CPU・GPU・FPGA の比較を示す。この表の数値は、Smith-Waterman アルゴリズムを各プラットフォームにおい

表 2.1: CPU・GPU・FPGA の比較 [17]

	CPU	GPU	FPGA
速度	×1	×14	×228
電力あたりの性能	×1	×31	×584
開発工数	×1	×45	×300

て実装，評価したものに基づいている [17]。表に示すように，速度・電力あたりの性能については，CPU・GPU と比較して FPGA が圧倒的な性能を誇る。中原らも，畳み込みニューラルネットワークを FPGA に実装し，その処理速度は組込み CPU や GPU よりも高く，消費電力は低いことを言及している [14]。特に FPGA は電力あたりの性能が高く，低消費電力であることが注目すべき点である。但し，FPGA に実装する場合，その性能を引き出すための専用ハードウェアの設計に，多くの開発工数が必要となる。

専用ハードウェアがソフトウェアよりも高速な処理を実現できるのは，そのデータパスによるものが大きい。CPU による逐次的な処理よりも，GPU や FPGA などが高速なのはその並列処理によるものだが，GPU と FPGA に性能の差が生じるのは，そのデータパスの深さの違いによるものである。GPU は汎用的な計算のために，データパスは固定であるが，FPGA で専用ハードウェアを構成する場合，データパスは自由であり，計算資源が許す限りの深いデータパスを構成することが可能である。したがって，ノイマン型アーキテクチャにおける，プロセッサとメモリの頻繁なやりとりによるボトルネックが生じにくい。

ASIC は，特定のアプリケーションを動作させるための IC チップである。ASIC は，FPGA と比較すると，実装する際に最適化されるため演算速度や消費電力において利点がある。また，ASIC は大量少品種の IC を製造する場合は，大量生産によるコストの削減が可能となる。その一方で，FPGA は回路を再構成可能であるため，柔軟性という点において利点がある。したがって，適宜回路の書き換えが必要な場合や，少量多品種の製造においては FPGA が適する。

近年，AI ハードウェアが盛んに開発されている。IBM は，2014 年にニューロモルフィックチップ TrueNorth[18] を発表した。TrueNorth は，100 万個のスパイクングニューロンと，約 2 億 5000 万個のシナプスを構成し，低電力な画像検出・画像認

識 (400×240 ピクセル・30fps の動画に対して 63mW) を実現している。また Intel は、1 億 3000 万個のシナプスを有した 13 万個のスパイクニューロンを構成する AI チップ Loihi[19] を発表した。Loihi は、MNIST の手書き数字文字認識のタスクにおいて、一般的な汎用コンピュータと比較して 1000 倍のエネルギー効率があるとされている。さらに、768 基の Loihi を搭載したニューロモルフィック・リサーチ・システム Pohoiki Springs によって、リアルタイムなジェスチャー認識、人工皮膚を使用した点字認識、新しい匂いの学習のタスクを、数 10mW の電力消費で実現している [20]。

2.4 本研究のアプローチ

本研究は、生物の脳全体の機能を模倣することで、ヒトのように考える AI モデルを設計するアプローチをとる。特に、現行の AI では実現されていない学習パラダイムに着目し、個人の経験に基づいた記憶の獲得、および記憶に基づいた予測・行動生成を行う AI モデルを考案する。

本研究は提案する AI モデルを人工ニューラルネットワークモデルで実装し、FPGA に実装することを目指す。提案モデルを人工ニューラルネットワークとして実装する理由は、1.2 節で述べたように、本研究の目的は、生物の神経細胞の動作を完全に模倣することではなく、サービスロボットに個人の経験に基づいた記憶の獲得の機能を実装することであり、ロボットシステムとの親和性を重視するためである。ロボットシステムとの親和性という観点から見ると、ソフトウェアで制御されるロボットのメインシステムは、AI モデルのハードウェアに信号を送り、その出力結果を受け取る。この時に、ロボットのシステムで取り扱う情報は、スパイクではなく連続値で表現される。したがって、スパイクニューラルネットワークを用いる場合は、情報のエンコードとデコードが必要になる。また、Izhikevich ニューロンモデル [21] など、ニューロンモデルの実装方法によっては高コストな演算を必要とし、ハードウェア実装が困難となる。

提案する AI モデルを FPGA に実装するのは、演算の高速化のためのプラットフォームとして、消費電力と柔軟さのバランスが取れているためである。GPU は演算を高速化できるが、消費電力が高いため、サービスロボットの応用には適さない。また、ソフトウェアによる制御のため、専用ハードウェアよりも高効率な演算を実装することは困難である。したがって、ハードウェアによる高速かつ低消費電力な実装

を目指すが、ここでは ASIC ではなく FPGA を選択する。FPGA は回路を再構成可能であるため、与えられたタスクに対して適切なサイズの AI モデルを書き込むことができる。

TrueNorth や Loihi などの既存の AI チップは、低消費電力な AI のアクセラレータという点では本研究が目指すものと共通するが、学習のパラダイムと、脳機能の実現に対するアプローチが異なる。前述の TrueNorth や Loihi を評価したタスクは、いずれも現行の AI が取り組むような学習パラダイムであり、本研究が取り組む、個人の経験に基づいた記憶の獲得のパラダイムとは異なる。さらに、これらの AI チップは汎用的な AI プラットフォームを構築しているのに対して、本研究は脳の機能を直接模倣したハードウェアを設計することで、ヒトのように考える AI モデルの実現を目指すものである。

第3章

個人の経験に基づいた知識の獲得

3.1 本章の概要

本章では、ヒトのように考える AI モデル実現の最初のステップとして、個人の経験に基づいた知識を獲得する AI モデルを提案する。ここでは、扁桃体 (Amygdala) に着目し、扁桃体の機能を模倣した古典的条件付けモデル (扁桃体モデル) を提案する。本章は始めに、扁桃体の古典的条件付けに関わる生理学的知見をまとめ、その知見に基づいた扁桃体モデルの先行研究をまとめる。次に、扁桃体モデルの構造とアルゴリズム、そしてその専用ハードウェア設計法を述べる。終わりに、提案する扁桃体モデルのアルゴリズム、およびハードウェアを評価する。

3.2 扁桃体

扁桃体は大脳辺縁系 (Limbic System) の一部で、情動を司り、記憶のメカニズムに関わることが知られている。扁桃体は、特に恐怖条件付けに関わることが古くから知られている [22, 23, 24, 25, 26]。恐怖条件付けは、パブロフの犬の実験 [27] で有名な古典的条件づけの一形態である。この実験は、犬が餌を与えられると同時に、ある音を聞かされるという事象を繰り返し経験することによって、餌とその音との関連づけが起き、次第に犬は音のみで反応を示すようになるというものである。ここで、犬は個人の経験に基づいた知識を獲得していることになる。すなわち、古典的条件づけは個人の経験に基づいた知識の獲得に効果的な方法である。

図 3.1 に、扁桃体の恐怖条件づけの神経回路を示す。恐怖条件付けでは、まず感覚刺

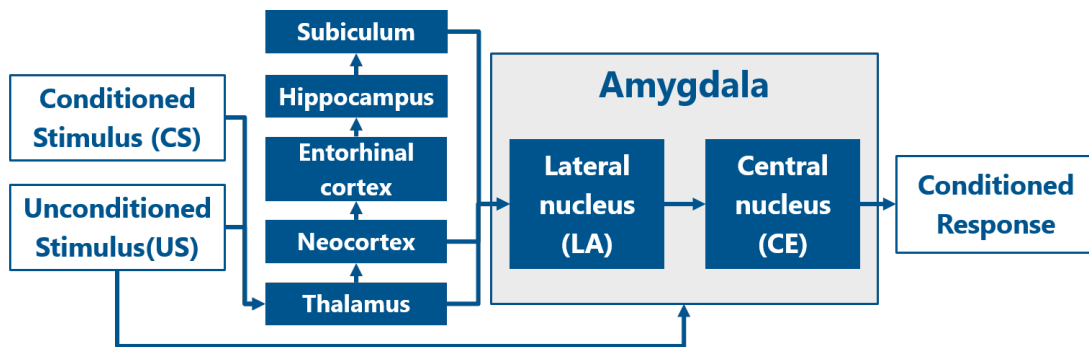


図 3.1: 扁桃体の恐怖条件付けの神経回路

激が無条件刺激 (Unconditioned Stimulus: US) と条件刺激 (Conditioned Stimulus: CS) に分類される。US は無条件反射 (Unconditioned Response: UR) を引き起こす刺激であり、パブロフの犬の実験では餌に相当する。CS は元来反射を引き起こさない中立的な刺激であり、条件づけにより条件反射 (Conditioned Response: CR) を引き起こす刺激となる。パブロフの犬の実験では音に相当する。

感覚刺激は扁桃体に至るまでに様々な脳器官: 視床 (Thalamus), 大脳新皮質 (Neocortex), 海馬 (Hippocampus) などを経由し、扁桃体外側核 (Lateral nucleus: LA) に投射される [24]。感覚刺激の LA への投射は大きく分けて 2 種類の経路: 直接的な経路, 間接的な経路がある。直接的な経路では, 視床から感覚刺激の特徴が投射され, 比較的速く情報が伝わる。間接的な経路では, 視床から大脳新皮質や海馬体を経由して, 物体情報, コンセプト, 文脈などの情報が比較的遅く伝わる。すなわち, LA は様々な脳器官から投射される情報を統合する部位である。LA は US と CS の条件づけを行う [28]。また, LA の神経細胞は特定の刺激に反応するものが局在化する特徴があることが知られている [24, 29]。扁桃体では LA から扁桃体中心核 (Central nucleus: CE) に投射がある。CE は体中に情動の反応を出力する部位である。US と CS の条件づけが完了している場合, 扁桃体は US が与えられずとも CS のみで CR を引き起こし, CE から出力される。

以上をまとめると, 扁桃体の恐怖条件付けは大きく分けて感覚刺激の認識と, US と CS の条件づけの 2 つの過程から成る。感覚刺激の認識では, 刺激の特徴に基づいて教師信号なしで CS を分類する。扁桃体は, その自己組織化的な認識機能を用いて, 未知の感覚刺激も分類することができる。条件づけの過程では, 扁桃体は US と CS の関係を学習する。この過程により扁桃体は, 元来中立的な刺激だった感覚刺激

の情動価値を推定できるようになる。

3.3 扁桃体モデルの先行研究

扁桃体の生理学的知見に基づいた、古典的恐怖条件付けモデルが提案されている [30, 31, 32, 33]. Arnomy ら [30] は、扁桃体・聴覚視床・聴覚野の接続を考慮した恐怖条件付けモデルを提案した。このモデルでは、聴覚刺激と無条件刺激の恐怖条件付けの検証を行っている。

Morén と Balkenius[31] は扁桃体と、消去 (Extinction) [34] を制御することが知られる前頭眼窩野 (Orbitofrontal Cortex) の接続を考慮したモデルを提案した。このモデルは、前頭眼窩野モデルと扁桃体モデルに分かれており、前頭眼窩野モデルが扁桃体モデルを抑制することにより、恐怖条件付けにおける消去を実現している。

Izhikevich[32] は古典的条件づけ、及びオペラント条件付けを行うスパイクニューラルネットワークモデルを提案した。このモデルは Izhikevich ニューロンモデル [21] により実装され、ドーパミンによって制御されるスパイクタイミング依存可塑性 (Spike-Timing-Dependent Plasticity : STDP) によって学習する。この研究では、1000 ニューロンから成るネットワークを用いたシミュレーションにより、報酬が行動の数秒後に与えられる Distal Reward Problem[35] を解決できることを説明している。

Li ら [33] は、恐怖の記憶の獲得と消去を行うスパイクニューラルネットワークモデルを提案した。このモデルは、扁桃体と内側前頭前野 (Medial Prefrontal Cortex) における「恐怖」「維持」「消去」の3種のニューロンによって構成されており、これらの組み合わせにより Partial Reinforcement Extinction Effect[36, 37, 38] (無条件刺激と条件刺激の部分的なペアリングによって獲得された恐怖の記憶は、完全なペアリングによって獲得された記憶よりも維持される効果) を再現できることを説明している。

園尾ら [39] は、自己組織化マップ (Self-Organizing Map : SOM) [40] と全結合型ニューラルネットワーク (Fully-Connected Neural Network : FCNN) によって構成される、情動発現モデル (Emotion expression Model of the Amygdala : EMA) を提案した。SOM は LA のモデルであり、パーセプトロンは CE のモデルである。SOM では、類似する入力に応答するニューロンが局在化しており、これは LA の特徴と一致する。EMA は小型ロボットの FPGA に実装され、色彩センサが取得した

色（CS とする）と触覚センサが取得した情報（US とする）の条件づけ学習が可能であることを示している。例えば、色彩センサから「赤」、触覚センサから「叩かれた」（「恐怖」を引き起こす）という刺激が同時に与えられると、EMA はこれらの刺激を条件づける。条件づけ学習が完了すると、EMA は「赤」の刺激を受け取るのみで「恐怖」を引き起こす。このモデルは、LA が感覚刺激を分類し、CS と US の関係性を学習するという扁桃体における恐怖条件づけの2つの過程が含まれる。また、LA として SOM を導入することで、このモデルは学習したことがない未知の感覚刺激に対しても情動評価を推測することができる。しかしながら、ここで用いられるタスクはホームサービスロボットに求められるものと比較すると単純である。

Rizzi ら [41, 42] は、Situation-Aware Fear Learning (SAFEL) モデルを提案した。このモデルでは、危険な状況を予測するため、状況が危険か否かを学習する。このモデルは、扁桃体モジュール・海馬モジュール・ワーキングメモリモジュールによって構成される。扁桃体モジュールは、状況の学習のトリガーとして用いられる。このモジュールは環境から感覚刺激を受け取り、各時刻における刺激が危険か否かを判定する。ここで、モジュールが刺激が危険であると判定した場合、刺激の危険度を示すアドレナリン信号を海馬モジュールに出力する。海馬モジュールは、環境から感覚刺激を蓄積し、蓄積した刺激の時系列（状況と定義する）が危険であるかどうかを定義する。扁桃体モジュールからアドレナリン信号を受け取ったとき、海馬モジュールは信号を受け取る直前の状況を危険な状況と判断する。海馬モジュールによって危険と判断された状況は、ワーキングメモリモジュールに保存される。以上の学習を行うことにより、このモデルはワーキングメモリモジュールを参照することで、危険な状況を予め推測することができる。SAFEL モデルにおける海馬モジュールは、海馬が経験を一時的に蓄積する機能を模倣したものと考えられるが、場所細胞や時間細胞など海馬における内部モデルの具体的な機能は考慮されていない。SAFEL モデルにおける扁桃体モジュールはパーセプトロン的一种によって実装される。刺激の危険度の推定は、予め定義された危険な刺激（US）と他の中立な刺激（CS）の古典的条件づけ学習に基づいている。

大林ら [43] は、情動を取り入れた Q 学習のシステムを提案した。このシステムでは、Morén と Balkenius が提案した扁桃体と前頭眼窩野からなる情動モデルを用いて、エージェントの情動の状態を定義する。ここでは、2つの情動モデルのインスタンス（快・不快の度合いを判断するもの、活動の度合いを判断するもの）の出力の組み合わせにより、情動の状態を決定する。このシステムは、状態と行動に加えて、情

動モデルによって決定した情動の状態を考慮した Q 学習を行う。大林らはこのシステムによって、既存の Q 学習では達成できない迷路探索課題を解決できることを示している。

3.4 提案モデル

本研究は、数回のエピソードから個人の経験に基づいた知識の獲得を行い、サービスロボットに応用することを目指す。3.3 で述べた扁桃体モデルの先行研究のうち、文献 [30, 31, 32, 33] は、生理学的妥当性を重視したものであり、本研究とは目的が異なる。また、Izhikevich ニューロンモデル [21] などのスパイクングニューラルネットワークは、計算コストが高くハードウェア実装に向かない。一方で、文献 [39, 41, 42] はロボットへの応用を考慮しており、本研究と立場に近い。但し、EMA[39] が対応できるタスクは、サービスロボットに求められるものと比較すると単純であるため、これを拡張する。また、SAFEL[41, 42] は、EMA で導入されている刺激の分類機能がないため、未知の感覚刺激の対応ができない。文献 [43] は、扁桃体モデルの応用を考慮している点では本研究と立場に近いが、用いられる情動モデルは Morén と Balkenius が提案した生理学的妥当性を重視したものである。このモデルもまた EMA で導入されている刺激の分類機能が考慮されていない。

3.4.1 モデル構成

図 3.2 に、提案する扁桃体を模倣した古典的条件付けモデル（扁桃体モデル）[44, 45, 46] の構造を示す。このモデルは、LA のモデルとしての複数の SOM と、CE のモデルとしての FCNN で構成される（図 3.2 右部）。これは、EMA[39] の SOM を Deep SOM Network[47] のレイヤーと入れ替えたものと捉えることができる。

扁桃体モデルは単一の US と複数の CS を同時に受け取る。CS は各 SOM に入力され分類される（CS の個数と SOM の個数が一致する）。各 SOM は分類結果に応じたベクトルを出力する。SOM が出力したベクトルは一つのベクトルとして結合される。結合されたベクトルは FCNN に入力される。扁桃体モデルには学習モードと推論モードがある。学習モードでは、FCNN は US と LA の出力ベクトルの関連付けを行う。推論モードでは、モデルは CS を受け取るのみで、CR を出力する。

このモデルはレストランで働くウェイターロボットに搭載し、客の好みを学習する

タスクに応用できる。このタスクに適用するため、ヒューマンロボットインタラクション (Human-Robot Interaction : HRI) 用インターフェースとして感覚系をモデルに追加する (図 3.2 左部)。ロボットが注文を取る時、感覚系 (音声認識) を通じて、注文したものの情報を取得する。ここで取得した情報を US とする。同時に、ロボットは感覚系を通じて、客の顔と場所の情報を CS として取得する。学習モードでは、扁桃体モデルはこの US と CS の関係性を学習する。学習モード完了後、モデルは客の顔と場所の情報のみを用いて、客が何を欲するかを推測する。

ここでは、図に示すような2つの SOM と、3つの感覚系で構成されるモデルを用いているが、SOM と感覚系の種類は拡張可能である。例えば、客の顔と場所に加えて、客が注文した時間も CS として考慮に入れることが可能である (SOM の個数は3つになる)。モデルが顔・場所・時間を CS として受け取る場合、客の注文場所と時間に応じた好みを推定することができるようになる。EMA[39] と比較すると、複数の要素の組み合わせを考慮できる点に提案モデルの利点がある。複数の要素を考慮に入れられるため、US のトリガーとなる CS の種類が未知の場合でも、このモデルを適用することができる。

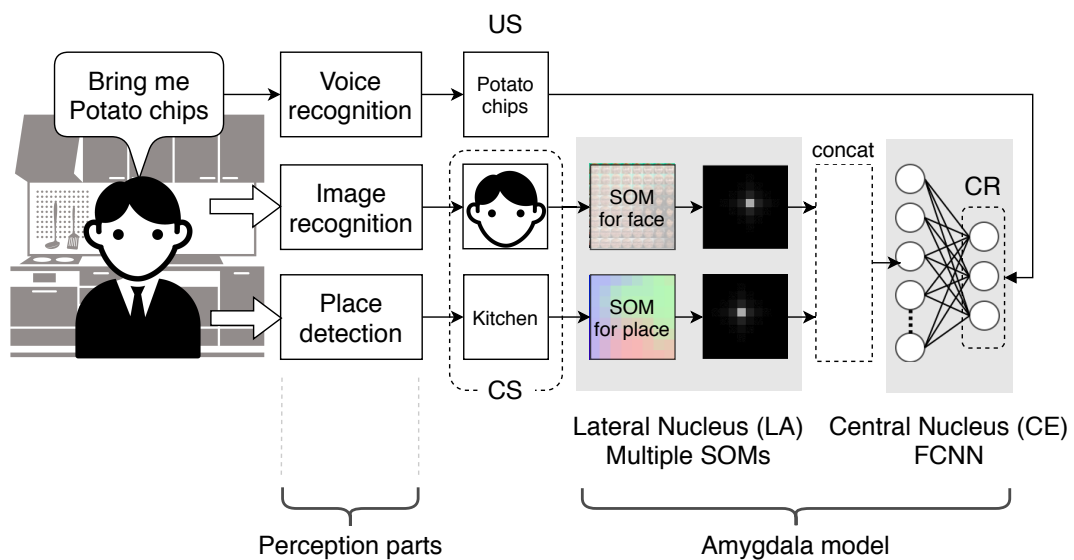


図 3.2: 扁桃体モデルの構成

3.4.2 アルゴリズム

扁桃体モデルの推論モードは、以下の手順で行う。CS を示す N 次元ベクトル $\mathbf{x}_k \in \mathbb{R}^N (k = 1, \dots, K)$ が与えられたとき (K は CS の個数を示す), 扁桃体モデルの LA が各ベクトルを受け付ける。LA は K 個の SOM を含み, ベクトル \mathbf{x}_k を k 番目の SOM に割り当て入力する。各 SOM は, 最も入力ベクトルに一致する参照ベクトルをもつ勝者ニューロンを, 以下の式により検索する,

$$(i_c, j_c) = \arg \min_{i,j} \|\mathbf{x}_k - \mathbf{w}_{i,j}\|, \quad (3.1)$$

ここで, $\mathbf{w}_{i,j} \in \mathbb{R}^N$ は SOM の参照ベクトルを示す。SOM のニューロンは 2 次元平面の $I \times J$ のグリッドに配置される。各ニューロンのインデックスを $(i, j) (i = 1, \dots, I, j = 1, \dots, J)$ で示す。式 (3.1) の (i_c, j_c) は, 勝者ニューロンのインデックスを示す。勝者ニューロンの決定後, SOM は勝者位置に応じた出力ベクトルを生成する。出力ベクトル $\mathbf{h}_k \in \mathbb{R}^{I \times J}$ は, 以下の式により算出する,

$$h_{i,j} = \exp(-d_{i,j}^2 / 2\sigma_{\text{prop}}^2), \quad (3.2)$$

$$d_{i,j} = \sqrt{(i - i_c)^2 + (j - j_c)^2}, \quad (3.3)$$

ここで, $h_{i,j}$ は出力ベクトル \mathbf{h}_k の要素である。すなわち, SOM のニューロン (i, j) がスカラー値 $h_{i,j}$ を出力すると捉えることもできる。変数 $d_{i,j}$ は, 勝者ニューロン (i_c, j_c) とニューロン (i, j) との 2 次元グリッド上の距離を示す。変数 σ_{prop} は, ガウス関数の分散を示す。全ての SOM の出力ベクトルは以下のように連結される,

$$\mathbf{H} = \{\mathbf{h}_1, \dots, \mathbf{h}_K\}. \quad (3.4)$$

ベクトル $\mathbf{H} \in \mathbb{R}^M (M = I \times J \times K)$ は CE に入力される。CE は条件反射ベクトル $\mathbf{y} \in \mathbb{R}^L (L$ は FCNN のニューロン数) を, 以下の式により算出する。

$$y_l = \phi(\mathbf{H} \cdot \mathbf{v}_l^\top), \quad (3.5)$$

ここで, $y_l (l = 1, \dots, L)$ は条件反射ベクトルの要素, $\mathbf{v}_l \in \mathbb{R}^M$ は CE の結合荷重ベクトルである。 ϕ は活性化関数を示し, 恒等関数, シグモイド関数, ソフトマックス関数などを適用できる。

扁桃体モデルの学習モードは、上記推論モードに加えて、以下の手順で行う。LA の SOM の参照ベクトルの更新は、以下に示す教師なし競合学習によって行う。

$$\Delta \mathbf{w}_{i,j} = \alpha(\mathbf{x} - \mathbf{w}_{i,j}) \exp(-d_{i,j}^2/2\sigma_{\text{learn}}^2), \quad (3.6)$$

$$\mathbf{w}_{i,j}^{\text{new}} = \mathbf{w}_{i,j}^{\text{old}} + \Delta \mathbf{w}_{i,j}, \quad (3.7)$$

ここで、 α は学習係数、 σ_{learn} は、ガウス関数の分散を示す。CE の結合荷重ベクトルの更新は、以下に示すデルタ則によって行う。

$$\Delta \mathbf{v}_l = -\beta(y_l - t_l)\phi'(\mathbf{H} \cdot \mathbf{v}_l^\top)\mathbf{H}, \quad (3.8)$$

$$\mathbf{v}_l^{\text{new}} = \mathbf{v}_l^{\text{old}} + \Delta \mathbf{v}_l, \quad (3.9)$$

ここで、 t_l は教師信号、すなわち US ベクトル $\mathbf{t} \in \mathbb{R}^L$ の要素である。 β は学習係数を示す。

3.4.3 ハードウェア

提案する扁桃体モデルを FPGA に搭載するためには、FPGA の回路資源を考慮する必要がある。例えば、乗算を実装する場合、Digital Signal Processor (DSP) slice が消費されるが、今回ターゲットデバイスとして使用する XCZU9EG FPGA[48] は DSP Slice が 2,520 個と他の資源よりも限られる (Look Up Table は 274,080 個、Flip Flop は 548,160 個)。ここでは、省回路資源で扁桃体モデルを実装するためのハードウェア指向アルゴリズムを考案する。このアルゴリズムでは、上記の乗算に加えて、多くの回路資源を必要とする指数関数の演算の回数を減らす。また、浮動小数点演算は多くの回路資源を必要とするため、演算に用いる変数を固定小数とする。

式 (3.1) では、入力ベクトルと参照ベクトルの距離 $\|\mathbf{x}_k - \mathbf{w}_{i,j}\|$ を算出する。通常の SOM アルゴリズムでは、L2 ノルム $\|\mathbf{x}_k - \mathbf{w}_{i,j}\|_2 = \sum_{n=1}^N (x_{k,n} - w_{i,j,n})^2$ がしばしば用いられるが、L2 ノルムはハードウェア実装で使用するべき乗算が含まれる。本手法では、L2 ノルムの代わりに乗算を必要としない L1 ノルム $\|\mathbf{x}_k - \mathbf{w}_{i,j}\|_1 = \sum_{n=1}^N |x_{k,n} - w_{i,j,n}|$ を用いる。ここでは、変数 \mathbf{x}_k , $\mathbf{w}_{i,j}$ には 8 ビット整数を用いる。

式 (3.2) (3.3) (3.6) は、乗算と指数関数の演算が含まれる。本手法では、これらの演算をビットシフト演算に置き換えて近似する。

$$h_{i,j} = 2^{-D_{i,j}}, \quad (3.10)$$

$$D_{i,j} = |i - i_c| + |j - j_c|, \quad (3.11)$$

$$\Delta \mathbf{w}_{i,j} = 2^{-(A+D_{i,j})} (\mathbf{x}_k - \mathbf{w}_{i,j}), \quad (3.12)$$

ここで、 A は正の整数であり、 2^{-A} は式 (3.6) における学習係数 α に相当する。式 (3.10) (3.12) では、指数関数を 2 の冪乗に置き換えている。2 の冪乗は算術ビットシフト演算で表現することができる。式 (3.10) では、8 ビットの固定小数（整数部 2 ビット、小数部 6 ビット）の変数を用いる。式 (3.3) では、ユークリッド距離 $d_{i,j}$ が用いられるが、これをマンハッタン距離に置き換える。このアルゴリズムで重要なのは、乗算を一切必要としないことである。

前述のように LA のアルゴリズムは改変するが、CE のアルゴリズムは改変しない。但し、浮動小数の変数を 8 ビットの固定小数（整数部 2 ビット、小数部 6 ビット）に置き換える。また、指数関数の使用を避けるため、活性化関数は恒等関数とする。

以上のハードウェア指向アルゴリズムを用いて、扁桃体モデルのハードウェアを設計する。図 3.3 に扁桃体モデル回路全体の構成図を示す。ハードウェアは K 個の LA モジュールと 1 個の CE モジュールで構成する。このハードウェアに対し、CS を示すベクトル \mathbf{x}_k が LA モジュールに入力される。LA モジュールの演算は並列に行われる。各 LA モジュールは出力ベクトル \mathbf{h}_k を計算し、これらのベクトルは \mathbf{H} として結合される。結合ベクトル \mathbf{H} は CE モジュールに入力される。CE モジュールは CR を示すベクトル \mathbf{y} を出力する。学習モードにおいては、US を示すベクトル \mathbf{t} が CE モジュールに与えられ、CR と US の誤差を最小化するように結合荷重を更新する。また、LA モジュールも自身の参照ベクトルを更新する。

図 3.4 に LA モジュールの構成図を示す。LA モジュールは $I \times J$ ニューロンを含んでおり、全てのニューロンの計算が並列に実行される。LA モジュールは参照ベクトル $\mathbf{w}_{i,j}$ を保持するための Random Access Memory (RAM) といくつかの演算ユニット：ノルム計算ユニット (Norm)、勝者決定ユニット (Winner)、距離計算ユニット (Dist)、出力ユニット (Output)、更新ユニット (Update) によって構成される。ノルム計算ユニットは入力ポートから \mathbf{x}_k 、RAM から $\mathbf{w}_{i,j}$ を受け取り、L1 ノルム $\|\mathbf{x} - \mathbf{w}_{i,j}\|_1$ を計算する。計算結果は勝者決定ユニットに入力される。勝者決定ユニットは最も小さい $\|\mathbf{x} - \mathbf{w}_{i,j}\|_1$ を持つニューロンを検索し、そのインデックス (i_c, j_c) を出力する。勝者ニューロンのインデックスは距離計算ユニットに入力される。距離計算ユニットは全てのニューロンと勝者ニューロンの距離 $D_{i,j} = D_i + D_j$ を計算する。距離は同時に出力ユニットと更新ユニットに入力される。出力ユニット

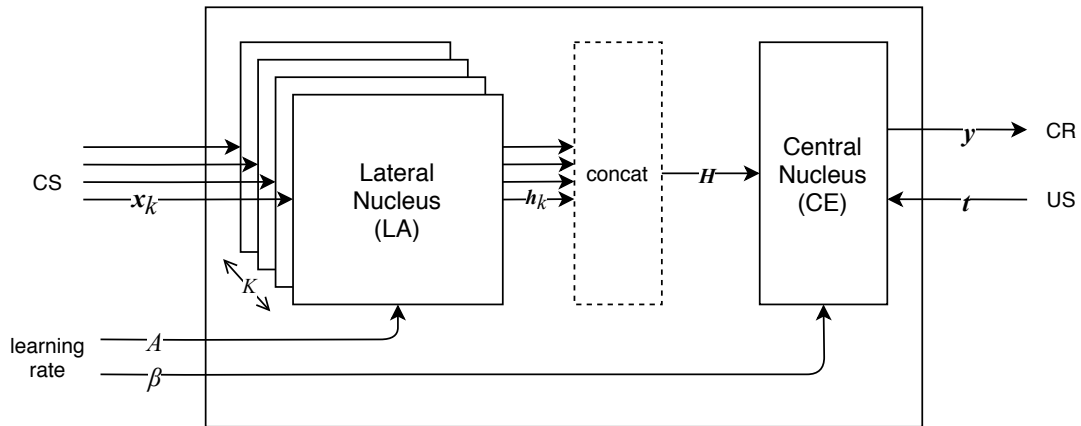


図 3.3: 扁桃体モデル回路

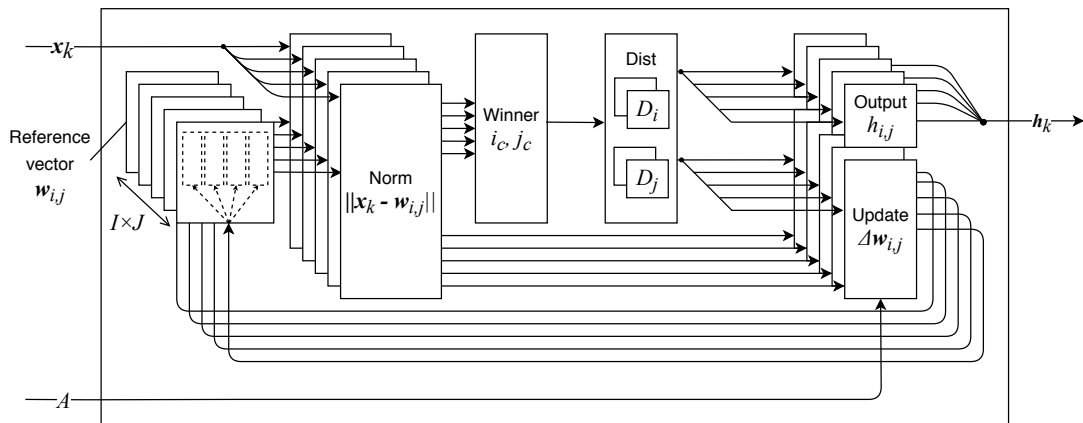


図 3.4: LA モジュール

は距離 $D_{i,j}$ を用いて出力ベクトル h_k を計算する。学習モードでは、更新ユニットが参照ベクトルの更新量 $\Delta w_{i,j}$ を計算する。このユニットは複数のベクトルの要素を同時並列に計算することができる。更新ユニットの出力は RAM に格納される。

図 3.5 にノルム計算ユニットを示す。図中の Sub は減算の演算素子、Abs は絶対値を算出する演算素子、Add は加算の演算素子を示す。図中の正方形はレジスタを示す。ノルム計算ユニットは入力ベクトルと参照ベクトルの要素を逐次的に受け取る。最初に、両要素の減算 $(x_{k,n} - w_{i,j,n})$ を計算する。次に、減算の絶対値 $|x_{k,n} - w_{i,j,n}|$ を計算し積算する。これにより L1 ノルムを計算することができる。この演算と同時に、入力ベクトルと参照ベクトルの減算の結果 $(x_{k,n} - w_{i,j,n})$ を別のレジスタに保持しておく。ここで保持した値は参照ベクトルの更新の際に再利用される。

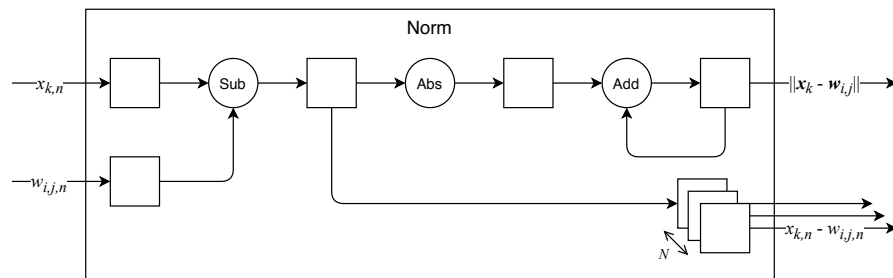


図 3.5: ノルム計算ユニット

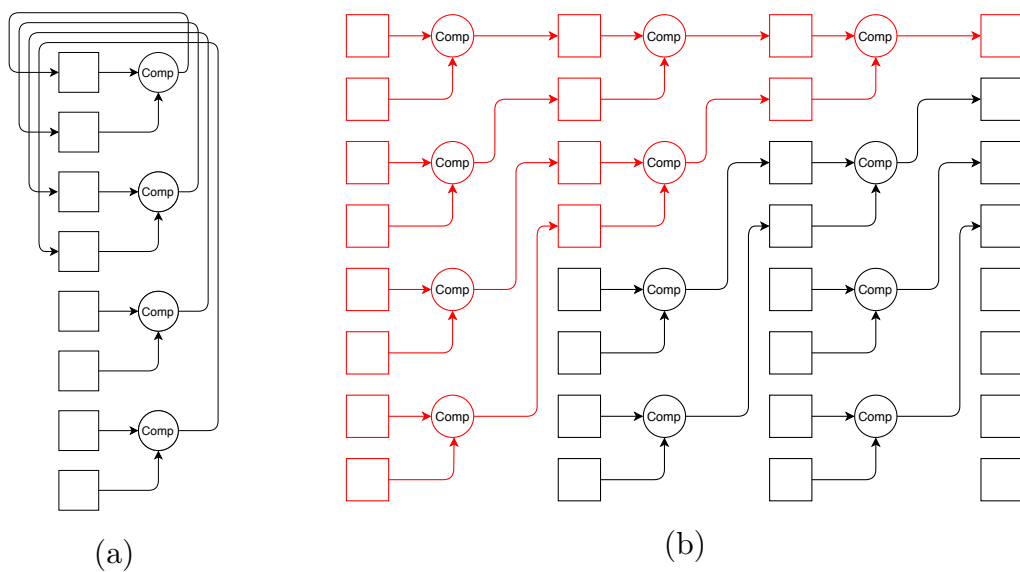


図 3.6: 勝者決定ユニット

図 3.6-(a) に勝者決定ユニットを示す．図中の Comp は比較器を示す．比較器は 2 つのノルム計算ユニットから L1 ノルムとそのインデックスを受け取り，小さい方の L1 ノルムとそのインデックスを出力する．勝者決定ユニットは $I \times J$ 個のレジスタと $I \times J \div 2$ 個の比較器によって構成される．比較器の出力は前半半分のレジスタに入力される．勝者ニューロンを検索するためには，各レジスタをノルム計算ユニットの出力値で初期化し，比較器による演算を $\log_2(I \times J)$ 回繰り返す．但し，ニューロン数 $I \times J$ は 2 の冪乗である必要がある．そうでなければダミーのレジスタを用意し，変数を取りうる最大値で初期化したものを追加する．図 3.6-(b) は勝者決定ユニットの演算を時間方向に展開したものである．図の赤線で描かれているように，最小の L1 ノルムとそのインデックスは，1 番目のレジスタに保持されることになる．

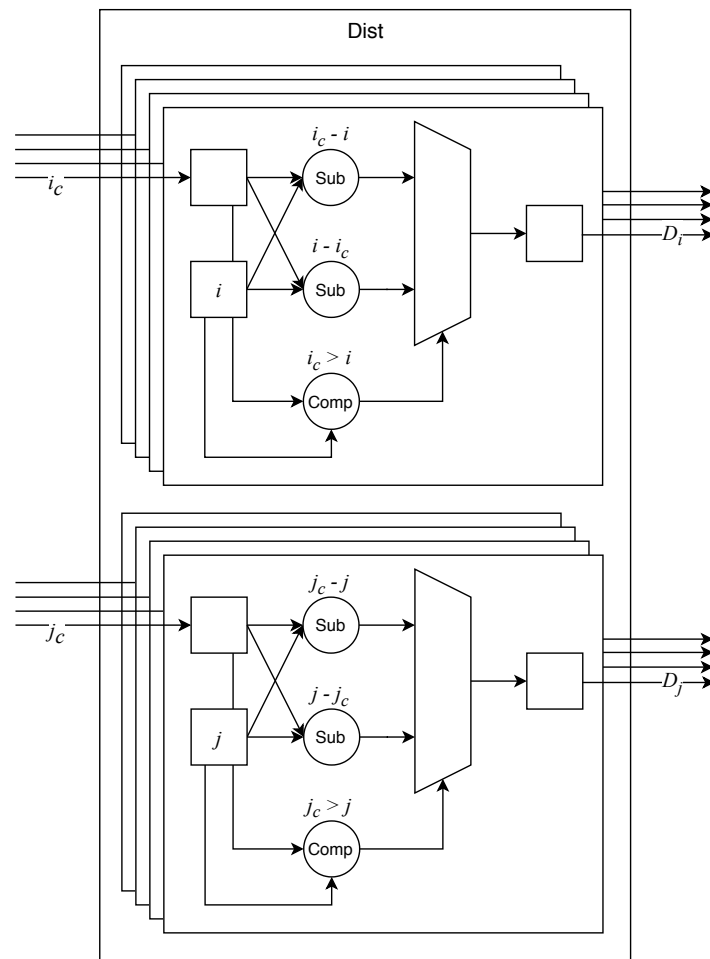


図 3.7: 距離計算ユニット

図 3.7 に距離計算ユニットを示す。距離計算ユニットは勝者ニューロンのインデックス (i_c, j_c) を受け取る。最初に、勝者ニューロンと自身のインデックスの全ての減算 ($i_c - i$), ($i - i_c$), ($j_c - j$), ($j - j_c$) を同時に計算する。勝者ニューロンと各インデックスの減算の絶対値を計算するため比較器を用いて、($i_c - i$) または ($i - i_c$), ($j_c - j$) または ($j - j_c$) を選択し出力する。

図 3.8 に出力計算ユニットを示す。図中の Sat は飽和処理の演算素子、Shift は算術ビットシフト演算の演算素子を示す。出力計算ユニットは勝者ニューロンからの距離 D_i, D_j を受け取り、それらの加算を行う。ここで求めた値は、0 から BIT_WIDTH_H の値になるように飽和処理される。ここで BIT_WIDTH_H は変数 h_k のビット幅であり、 $BIT_WIDTH_H = 8$ とする。ここで求めた値を用い

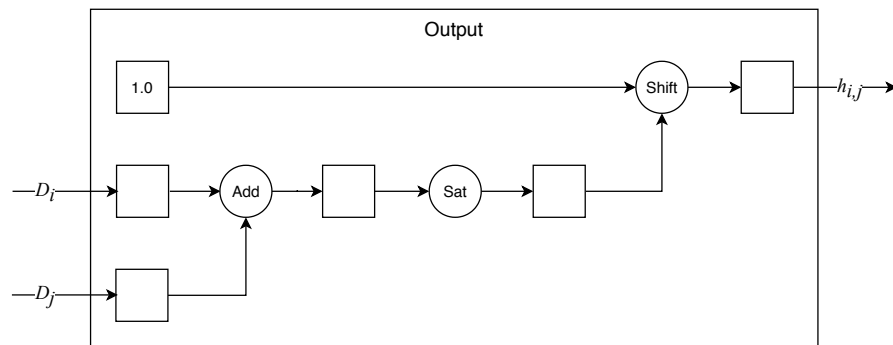


図 3.8: 出力計算ユニット

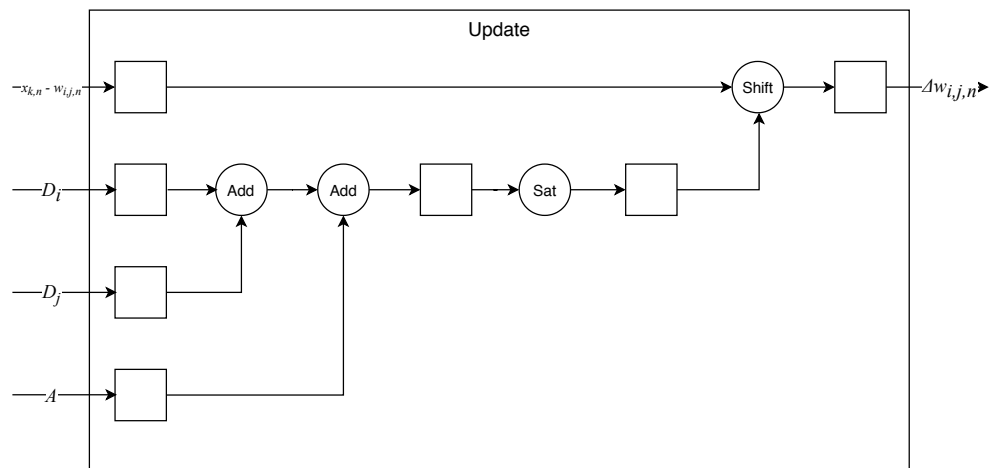


図 3.9: 更新ユニット (LA モジュール)

て、1.0 を算術ビットシフト演算する。

図 3.9 に LA モジュールの更新ユニットを示す。更新ユニットは距離ユニットの出力 D_i , D_j と学習係数 A を受け取り、それらの加算を行う。ここで求めた値は、0 から BIT_WIDTH_W の値になるように飽和処理される。ここで BIT_WIDTH_W は変数 $w_{i,j}$ のビット幅であり、 $BIT_WIDTH_W = 8$ とする。更新ユニットは $(x_{k,n} - w_{i,j,n})$ をノルム計算ユニットから受け取り、この値を算術ビットシフト演算する。更新量 $\Delta w_{i,j,n}$ は RAM に入力され、 $w_{i,j,n}$ に加算される。

図 3.10 に CE モジュールの構成図を示す。CE モジュールは L 個のニューロン (L は US の次元数) で構成され、全てのニューロンの演算は並列に行われる。CE モジュールは結合荷重ベクトルを保持する RAM といくつかの演算ユニット：乗算ユニット (Mult), 加算ツリーユニット (Adder), 差分計算ユニット (Diff), 更新ユ

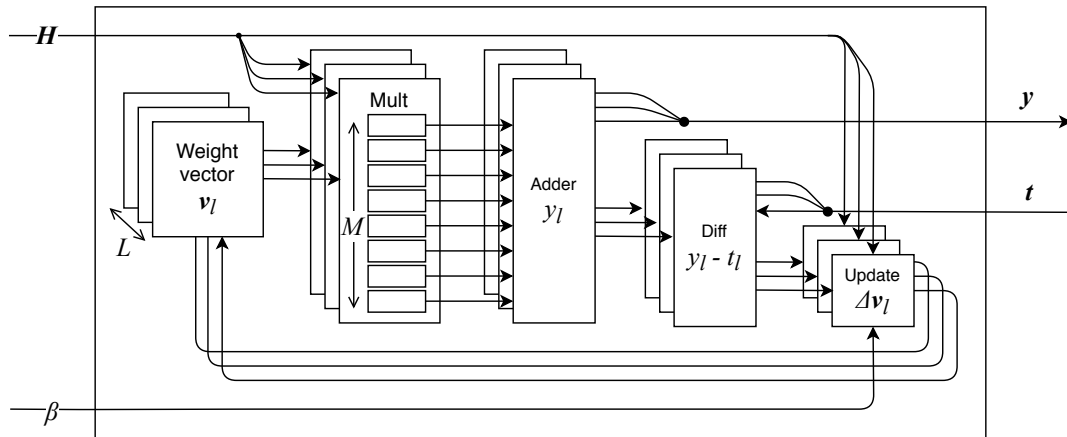


図 3.10: CE モジュール

ユニット (Update) によって構成される。乗算ユニットはベクトル \mathbf{H} と \mathbf{v}_l の要素積を計算する。したがって、計算結果は $M (= I \times J \times K)$ 要素となる。計算結果は加算ツリーユニットに入力され、内積 $\mathbf{H} \cdot \mathbf{v}_l$ を計算するために全ての要素の加算を行う。学習モードでは、加算ツリーユニットの出力 \mathbf{y} と、教師信号 (US) t が差分計算ユニットに入力され、 $y_l - t_l$ が計算される。その差分は更新ユニットに入力され、結合荷重の更新値 $\Delta \mathbf{v}_l$ が RAM に入力され格納される。

図 3.11-(a) に加算ツリーユニットを示す。加算ツリーユニットは $M (= I \times J \times K)$ 個のレジスタと $M \div 2$ 個の加算器から構成される。加算器は 2 つの入力を受け、その加算結果を出力する。加算器の出力は前半半分のレジスタに入力される。レジスタに入力された全ての値の加算を行うためには、加算器による演算を $\log_2 M$ 回繰り返す。但し、乗算ユニットの出力数 M は 2 の冪乗である必要がある。そうでなければダミーのレジスタを用意し、0 で初期化したものを追加する。図 3.11-(b) に加算ツリーユニットの演算を時間展開したものを示す。図中の赤線で描かれるように、加算の計算結果は 1 番目のレジスタに保持される。

図 3.12 に CE モジュールの更新ユニットを示す。図中の Mult は乗算器を示す。乗算器は 2 つの値を受け取り、その乗算結果を出力する。更新ユニットは学習係数 β と、 $y_l - t_l$ を差分計算ユニットから受け取り、乗算器に入力する。また更新ユニットはベクトル \mathbf{H} を受け取り、 \mathbf{H} と $\beta(y_l - t_l)$ の乗算を行う。更新ユニットの出力 $\Delta v_{l,m}$ は RAM に入力され、結合荷重に加算される。

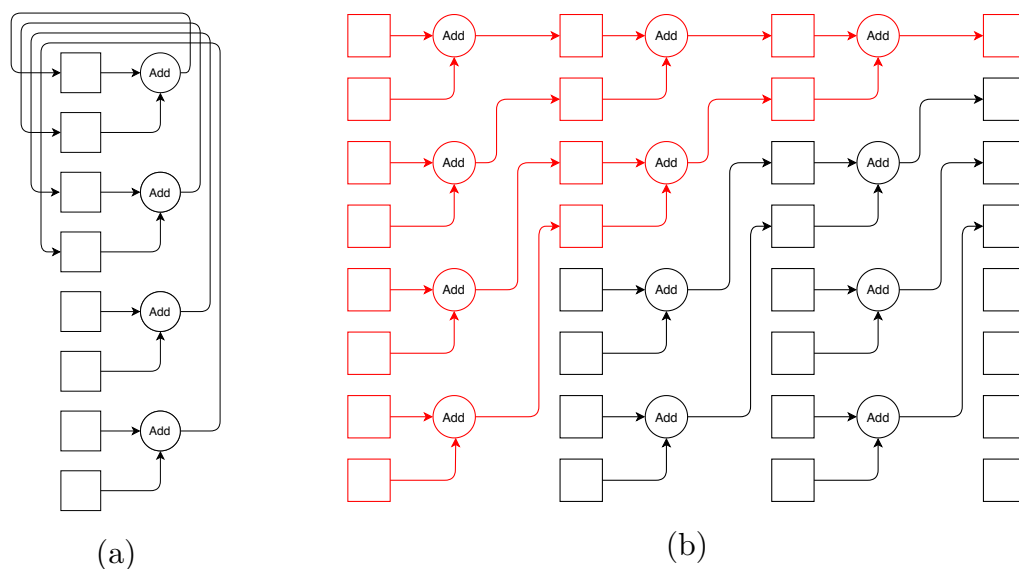


図 3.11: 加算ツリーユニット

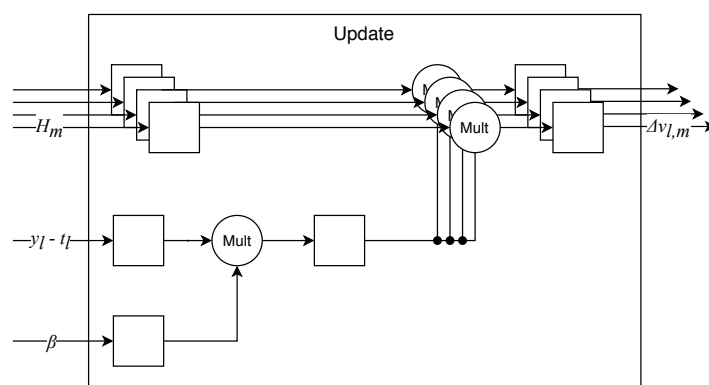


図 3.12: 更新ユニット (CE モジュール)

3.5 実験

3.5.1 アルゴリズム

提案する扁桃体モデルが、数回の HRI を通じて客の好みを学習できるかどうかを確認した。ここでは、提案モデルをサービスロボット (TOYOTA Human Support Robot : HSR) [16] に搭載して実験を行った。HRI に用いる感覚系モデルとして、音声認識のため Web Speech API[49] を、物体画像認識のため YOLOv2[50] を、場所

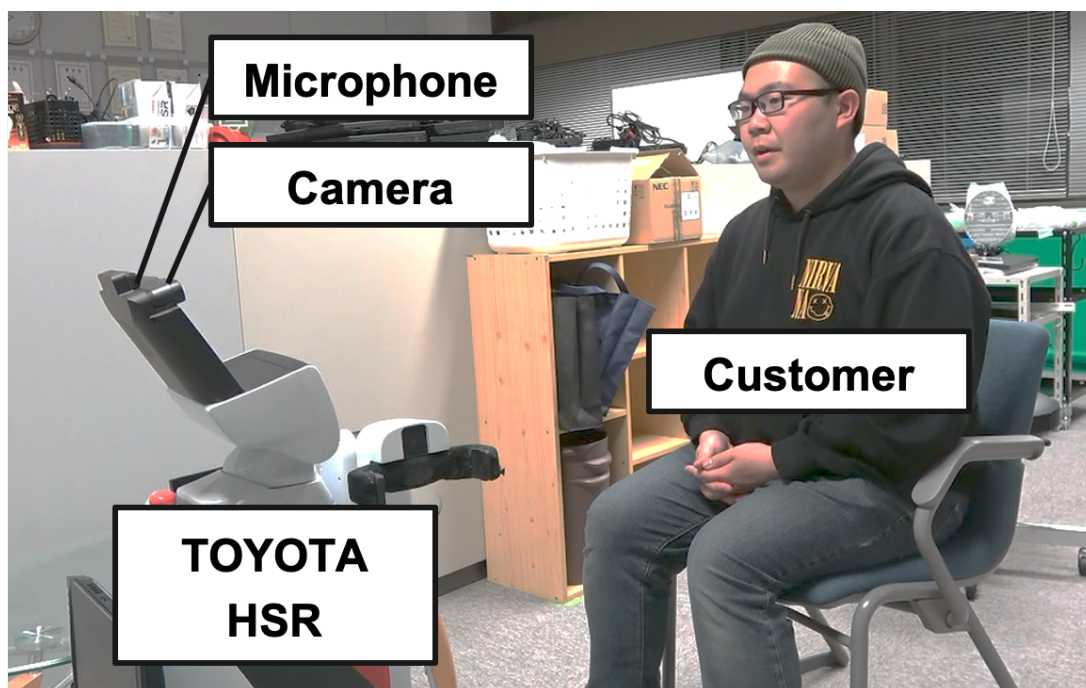


図 3.13: 扁桃体モデルの実験における HRI

の認識のため自己位置推定 (Simultaneous localization and mapping : SLAM) を用いた。ロボットに搭載されているカメラとマイクを用いて、US と CS を取得した (図 3.13)。ここで、CS として以下に示す 2 種類のベクトルを用意した：客 A・B を示す顔画像 (64×64 ピクセル・RGB チャンネル)、場所 A・B を示す場所ベクトル (3 次元の One-Hot ベクトル)。これらの CS を用いて、表 3.1 に示すシチュエーションを定義した。例えばシチュエーション A の場合、客 A が場所 A にいるときはいつもオブジェクト A を注文するように設定した。提案モデルは、表 4 に定義したシチュエーションの通りの US と CS を受け取り、その関係性を学習した。最初に、シチュエーション A の US と CS を 10 回入力し、次にシチュエーション B の US と CS を 10 回入力した。シチュエーション C, D に関しても、同様に続けて入力した。

本実験はソフトウェアによって実装された扁桃体モデルで検証した。また、このソフトウェアにはハードウェア指向アルゴリズムは適用しておらず、全ての変数は 32 ビットの浮動小数として実装した。式 (3.5) の活性化関数はソフトマックス関数とし、CE の出力は客が欲するオブジェクトの確率とみなした。

図 3.14 に、HRI 中に扁桃体モデルが推定したオブジェクトの確率を示す。それを

表 3.1: 定義したシチュエーション

Situation	Face (CS)	Place (CS)	Ordered object (US)
A	Customer A	Place A	Object A
B	Customer A	Place B	Object B
C	Customer B	Place A	Object B
D	Customer B	Place B	Object C

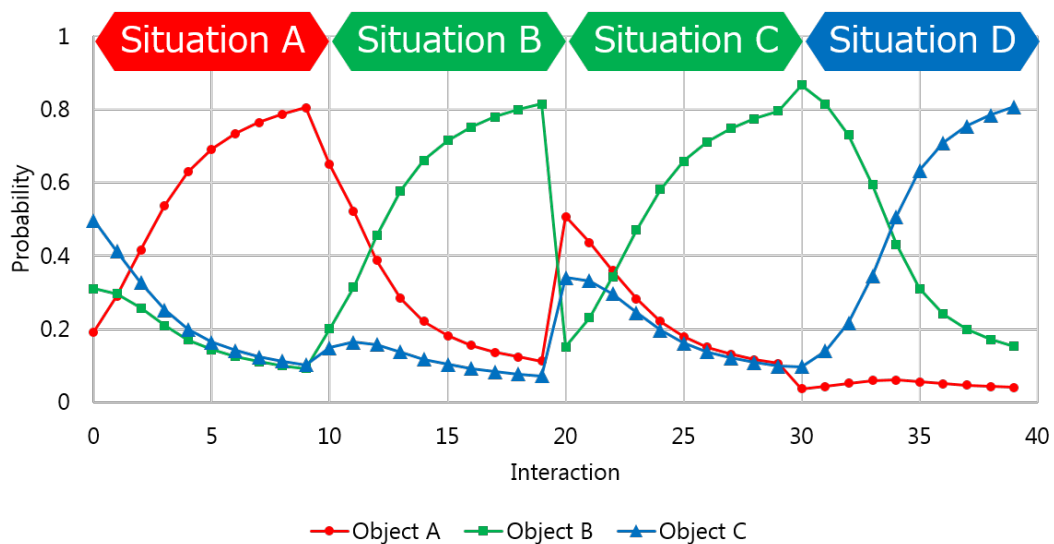


図 3.14: HRI 中に扁桃体モデルが推定したオブジェクトの確率

れのシチュエーションにおいて、3回から4回のHRIを通じて、モデルは正しいオブジェクトを推定できるようになった。この結果より、モデルは数回のHRIを通じて新しいシチュエーションに適応したことがわかる。

さらに、扁桃体モデルの汎化性能を確認するため、1人の客に対して1000枚の顔画像（様々な方向から撮影した顔を含む）を用意し、学習用750枚とテスト用250枚に分割した。学習モードのHRIでは、750枚の学習用画像からランダムに1枚を選択し、場所を示すOne-Hotベクトルと合わせてモデルに入力した。各HRIの後、250枚全てのテスト用画像と場所を示すOne-Hotベクトルをモデルに入力し、その

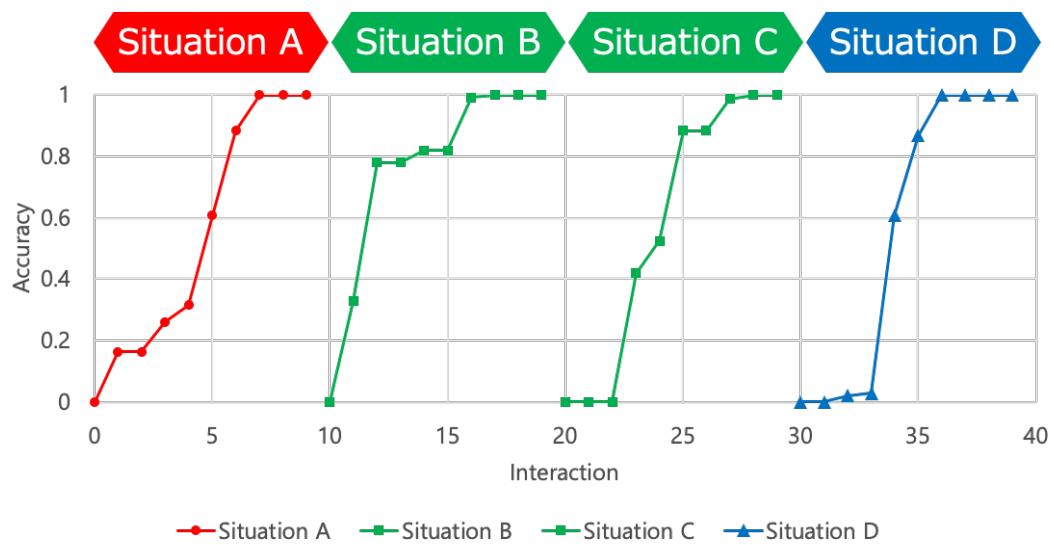


図 3.15: テストデータに対する扁桃体モデルの正答率

時の正答率を求めた。図 3.15 に、HRI 中の扁桃体モデルの正答率を示す。この結果より、モデルは 10 回の HRI の中で、各シチュエーションに適応したことがわかる。

表 3.2: 扁桃体モデルの回路使用率

	Used	Available	Utilization [%]
LUT	67,198	274,080	24.52
LUTRAM	1,472	144,000	1.02
FF	48,398	548,160	8.83
BRAM	672	912	73.68
DSP	675	2,520	26.79

3.5.2 ハードウェア

提案する扁桃体モデルのハードウェアを FPGA に搭載した。ターゲットデバイスは XCZU9EG[48] とし、開発環境は Xilinx SDSoC 2018.3[51] を用いた。

扁桃体モデルの構造のハイパーパラメータは以下の通り設定した：LA における SOM の個数 $K = 2$ ，SOM の参照ベクトルの次元数 $N = 12,288 (= 64 \times 64 \times 3)$ ，SOM のノード数 $I \times J = 8 \times 8$ ，CE のニューロン数 $L = 3$ 。

ハードウェアの設計は Vivado HLS[52] を用い、扁桃体モデルの演算をハードウェア関数として記述し論理合成した。表 3.2 に、ターゲット動作周波数を 150MHz とした時の扁桃体モデルの回路使用率を示す。表に示す通り、全ての回路資源の使用率が 100% を下回っていることから、この回路はターゲットデバイスに搭載可能であることがわかる。回路資源のうち、BRAM の使用率が最も高かった。Worst Negative Slack (WNS) 及び、Worst Hold Slack (WHS) はそれぞれ、0.568ns と 0.01ns であった。

提案するハードウェア指向アルゴリズムの効果を確認するため、改変前のオリジナルのアルゴリズムによって実装した扁桃体モデルのハードウェア関数を記述し論理合成した。ここでは、逐次型 (Sequential) と並列型 (Parallel) の 2 種類のハードウェア実装を行った。逐次型では演算が逐次的に行われ、並列型では演算が並列に行われるように実装した。両方の実装において、ターゲット動作周波数を 150MHz に設定した。表 3.3 に、回路使用率と学習モードのレイテンシを示す。但し、並列型の回路使用率は 100% を超えており、タイミング制約を満たさなかったためターゲットデバイスへの実装が不可であり、表に示す回路使用率とレイテンシは Xilinx Vivado と

表 3.3: 扁桃体モデルの回路使用率・動作速度の比較

	Sequential (floating-point)	Parallel (floating-point)	Proposed (fixed-point)
LUT	36,174 (13.20%)	1,057,222 (385.73%)	67,198 (24.52%)
LUTRAM	2,047 (1.42%)	484,387 (336.38%)	1,472 (1.02%)
FF	46,517 (8.49%)	401,409 (73.23%)	48,398 (8.83%)
BRAM	784.50 (86.02%)	912 (100.00%)	672 (73.68%)
DSP	329 (13.06%)	4,866 (193.10%)	675 (26.79%)
Latency	62,159,985 [CLK]	95,426 [CLK]	15,395 [CLK]

SDSoC による推定値である。逐次型では、LUTRAM と BRAM を除いて、回路使用率は提案ハードウェアよりも低かった。これは、単一の演算ユニットがハードウェアに搭載され、ベクトルの要素を逐次的に演算するように実装されたためである。しかしながら、SDSoC の推定によると、逐次型のレイテンシは 62,159,985CLK であり、提案ハードウェア (Proposed) の 15,395CLK を大幅に超えた。したがって、逐次型実装は非効率的である。並列型ではレイテンシが 95,426CLK と逐次型から大きく改善されたが、依然として提案ハードウェアよりも高いレイテンシとなった。

扁桃体モデルのハードウェアを FPGA に実装し、その動作を確認した。まず、事前学習を実行した。事前学習は CS を SOM に与え、自己組織化した SOM の参照ベクトルを獲得するために行う。顔用の SOM の場合、前述の顔画像データセットの学習用 750 枚からランダムに選択した画像を、場所用の SOM の場合、ランダムに生成した 3 次元ベクトル (8 ビット整数) を CS として与えた。ここで、式 (3.12) の A は 1 とした。図 3.16 に、500 回の事前学習前後の SOM の参照ベクトルを示す。LA には顔用の SOM ($k = 1$) と場所用の SOM ($k = 2$) があるため、それぞれの事前学習前後の結果を示す。図 3.16-(c), 3.16-(d) において、各参照ベクトルは 8 ビット整数の 3 次元ベクトルであるため、RGB カラーの正方形で表現した。図 3.16-(a), 3.16-(c) に示す通り、参照ベクトルの初期値は 0 であるが、図 3.16-(b), 3.16-(d) に示す通り、事前学習によって SOM は自己組織化されたことがわかる。

図 3.17 に、事前学習における勝者ニューロンの参照ベクトルと入力ベクトルの誤差を示す。顔用の SOM の場合、前述の顔画像データセットのうち評価用 250 枚を誤差の計算に用いた。場所用の SOM の場合、場所 A・B を示す One-Hot ベクトルを誤差の計算に用いた。 k 番目の SOM の誤差 L_k は以下の式によって算出した。

$$L_k = \frac{1}{P} \sum_{p=1}^P \frac{1}{N} \|\mathbf{x}_k(p) - \mathbf{w}_{i_c, j_c}\|_1, \quad (3.13)$$

ここで、 $\mathbf{x}_k(p)$ は p 番目のテストデータで、 P はテストデータの総数である。事前学習によって、誤差は減少し収束したことを確認できる。参照ベクトルと入力ベクトルは 8 ビットの整数であるため、誤差の最大値は 255 である。

事前学習完了後、提案ハードウェアによる古典的条件付け学習の動作確認を行った。ここでは、表 3.1 に示すシチュエーションに基づいて US と CS を同時に与えた。最初に、シチュエーション A の US と CS を 10 回入力し、次にシチュエーション B の US と CS を 10 回入力した。シチュエーション C, D に関しても、同様に続けて入力した。式 (3.12) の A は 4、式 (3.8) の β は 0.1 とした。図 3.18 に、古典的条件付け学習における CE の出力変化を示す。学習の初期では、シチュエーション A が入力されるのに対してオブジェクト A の出力が単調増加し、その他の出力は単調減少した。その後、10 ステップ目でシチュエーションが切り替わり、シチュエーション B の入力に対しオブジェクト B の出力が増加した。30 ステップ目でシチュエーション C に切り替わった後も同様に、オブジェクト C の出力が増加した。以上の結果より、ハードウェア実装のため簡略化されたアルゴリズムでも、US と CS の組み合わ

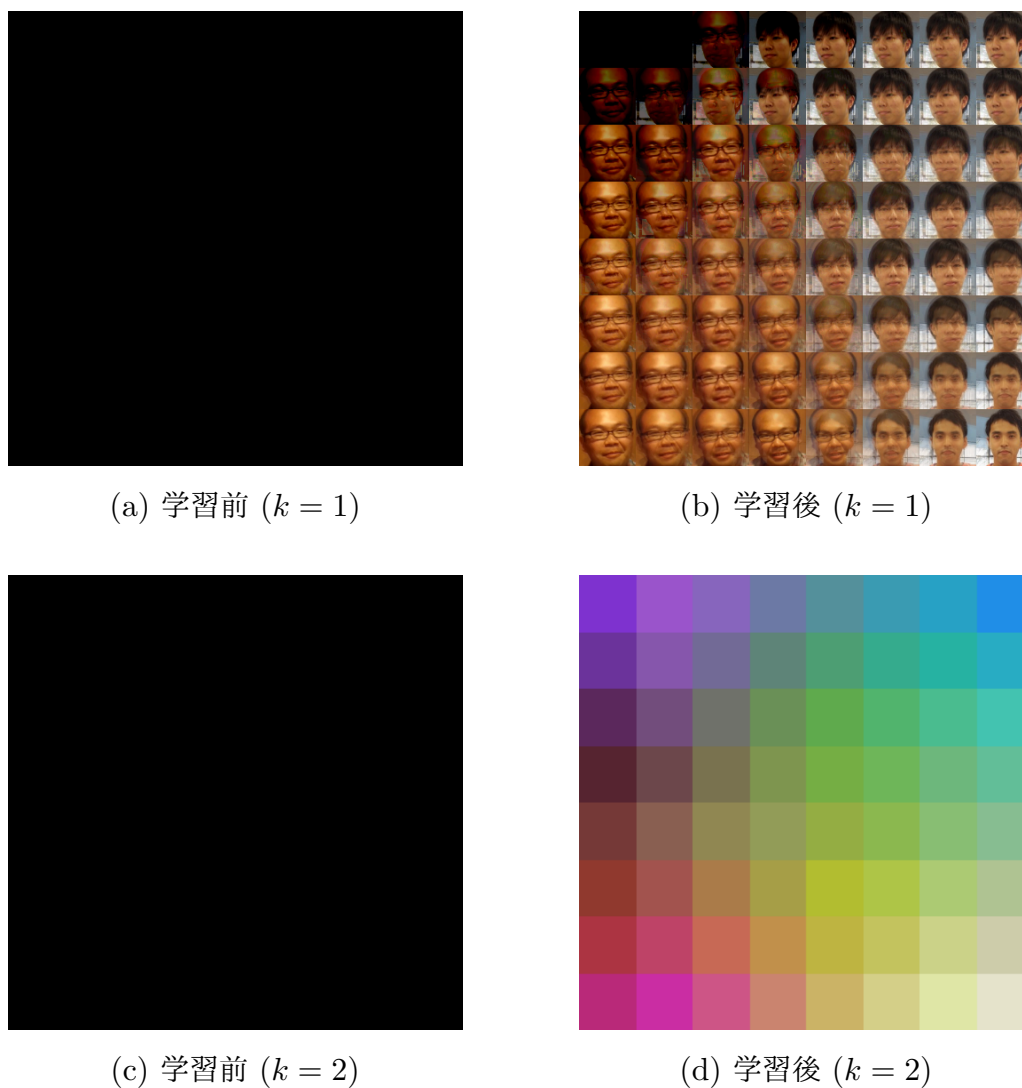


図 3.16: FPGA の計測結果：事前学習前後の SOM の参照ベクトル

せ（すなわち客の好み）の知識を数回の学習で獲得できたことがわかる。

さらに、扁桃体モデル回路の汎化性能を確認するため、前述の顔画像データセットのうちテスト用 250 枚全てを用いて各 HRI 後の正答率を計算した。図 3.19 に、HRI 中の扁桃体モデル回路の正答率を示す。以上の結果より、モデルは 10 回の HRI の中で、各シチュエーションに適応したことを確認した。

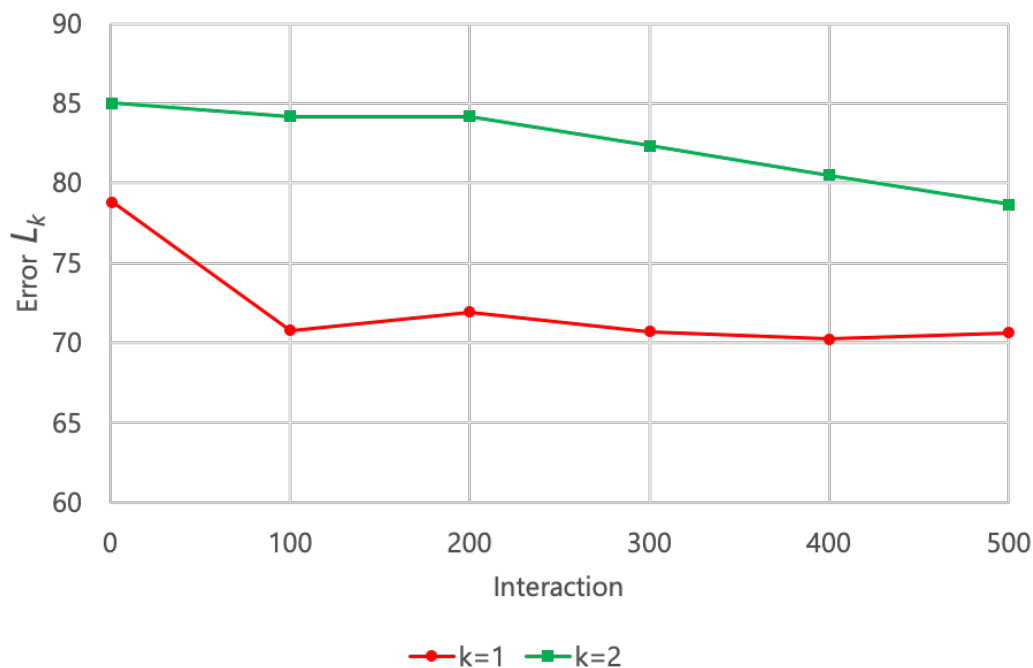


図 3.17: FPGA の計測結果 : SOM の誤差

提案ハードウェアの処理速度を計測した。本実験で用いた XCZU9EG は、Processing System (PS) と Programmable Logic (PL) が統合された System-on-Chip (SoC) FPGA である。ここでは、扁桃体モデル回路は PL に実装され、PS 上のソフトウェアが PL にデータを送り、PL の計算結果を受け取るようにした。学習モードを 500 回繰り返し、学習開始から終了までの実行時間を計測し、その平均値を求めた。また、比較のため、XCZU9EG の PS (Arm Cortex-A53), Intel Core i5 4670K, NVIDIA Jetson TX2 の CPU (Nvidia Denver + Arm Cortex-A57) に扁桃体モデルのソフトウェアを実装した。Cortex-A53 と Core i5 については C++ で記述したソフトウェアを、Jetson TX2 については Python と NumPy[53] によって記述したソフトウェアを実行した。これに加えて、NumPy の演算を CuPy[54] に置き換えたソフトウェアを Jetson TX2 の GPU で実行した。Jetson TX2 は、全ての演算コアが有効となる Max-N モードで実行した。表 3.4 にソフトウェアと提案ハードウェアの平均実行時間を示す。表に示す通り、提案ハードウェアは GPU を含めた全てソフトウェアよりも高速に動作した。

ハードウェアの実行時間には PS から PL にデータを送る転送時間が含まれる。扁

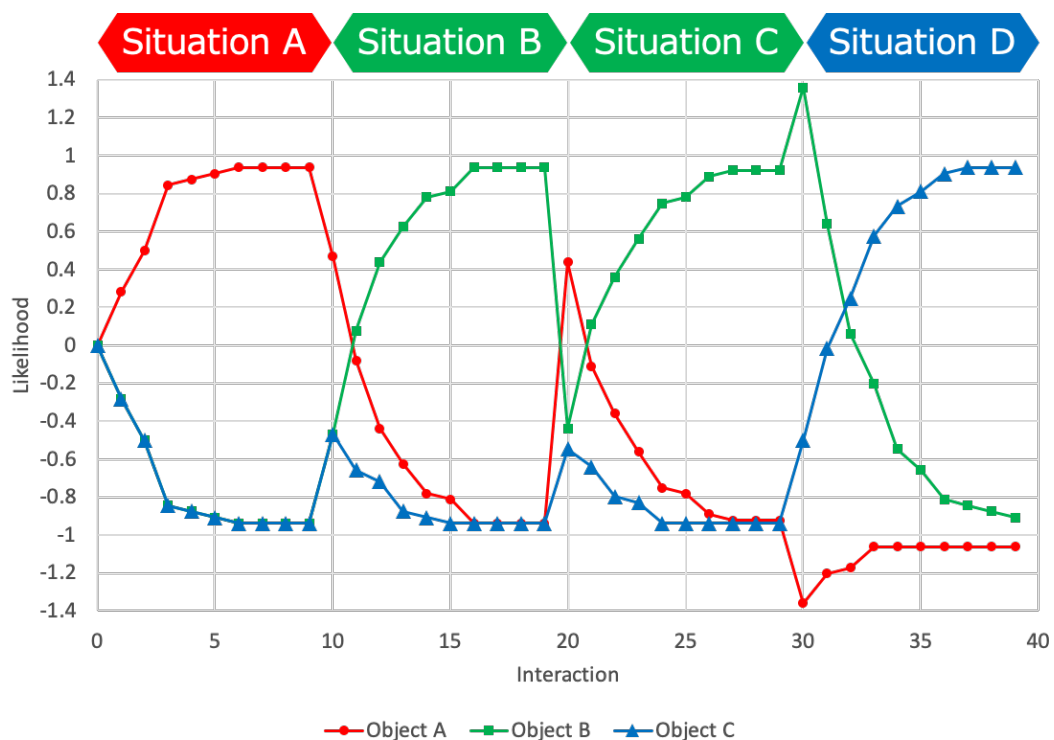


図 3.18: FPGA の計測結果 : HRI 中に扁桃体モデルが推定したオブジェクトの確率

扁桃体モデル回路本体の実行時間を計測するため、ダミーの CS を受け取りダミーの CR を出力するのみの空のハードウェア関数 (Empty) を実装した。ターゲット動作周波数を 150MHz に設定し、空のハードウェア関数を論理合成し XCZU9EG で実行した。表 3.5 に、提案ハードウェア (Entire) と空のハードウェアの実行時間を示す。

表 3.4: 扁桃体モデルの実行時間の比較

Implementation @ device	Execution time [s]
C++ @ A53	251.3 m
C++ @ i5	22.0 m
Python + NumPy @ TX2	42.3 m
Python + CuPy @ TX2	18.7 m
Hardware @ XCZU9EG	197.4 μ

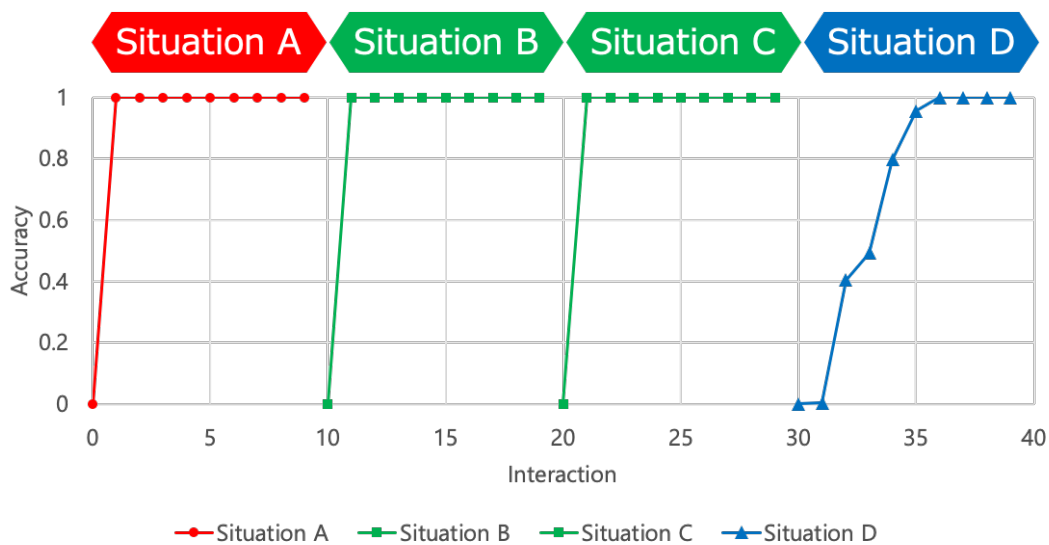


図 3.19: FPGA の計測結果：テストデータに対する扁桃体モデルの正答率

表 3.5: 扁桃体モデル回路の実行時間

	Execution time [s]
Entire hardware	197.4 μ
Empty hardware	90.7 μ

表 3.6: 扁桃体モデル回路の推定消費電力

	Power [W]
Entire hardware	5.009
Empty hardware	3.748

提案ハードウェアと空のハードウェアの消費電力を計測した。表 3.6 に、Xilinx Vivado が推定した消費電力を示す。表に示す通り、提案ハードウェアの消費電力は 5.009W と推定された。

3.6 本章のまとめ

本章では、個人の経験に基づいた知識を獲得する、扁桃体の機能を模倣した古典的条件付けモデルを提案した。ここでは、提案するモデルが、数回の HRI を通じて人の好み、すなわち個人の経験に基づいた知識を獲得できることを示した。さらに、提案モデルの専用ハードウェアを設計するため、ハードウェア指向アルゴリズムを提案し、LA モジュールのアルゴリズムから乗算と指数関数演算を排除した。仮に、同じサイズの提案モデルをハードウェア指向アルゴリズムなしで実装した場合、LA モジュールは 6,291,840 回の乗算を必要とする。ここで、ハードウェア化のためにアルゴリズムを単純化したにも関わらず、提案モデルのハードウェアは LA モジュールの参照ベクトルを自己組織化し、数回の HRI を通じて人の好みを学習できる。提案するハードウェア指向アルゴリズムを用いて設計した専用ハードウェアを FPGA に搭載した。150MHz で動作する提案ハードウェアは、Arm Cortex-A53 の 1,273 倍、Intel Core i5 4670K の 111.4 倍、Jetson TX2 の CPU の 214.3 倍、GPU の 94.73 倍で動作した。

本章では、提案モデルを単体で動作させるタスクで評価したが、生物の扁桃体がそうであるように、提案する扁桃体モデルの出力は、他の脳部位モデルに対して影響を及ぼして、学習を制御することにも応用可能である。特に、扁桃体は海馬との密接な関わりが知られており、記憶のメカニズムに関与することが知られている。次章では、扁桃体モデルと他の脳部位を組み合わせた人工知能モデルを提案し、個人の経験に基づいた記憶の獲得に応用する。

第4章

個人の経験に基づいた記憶の獲得 及び記憶に基づいた予測

4.1 本章の概要

本章では、個人の経験に基づいた記憶の獲得、及び記憶に基づいた予測を行う AI モデルを提案する。ここでは、3章で着目した扁桃体に加えて、海馬 (Hippocampus) と前頭前野 (Prefrontal Cortex) に着目し、これらの3部位の機能を統合した AI モデルを考案する。本章は始めに、海馬・前頭前野の記憶の獲得、予測に関する生理学的知見をまとめ、その知見に基づいた AI モデルの先行研究を述べる。次に、提案する AI モデルの構成例とアルゴリズムを述べる。終わりに、提案する AI モデルを評価する。

4.2 海馬・前頭前野

4.2.1 海馬

海馬は脳辺縁系の一部で、エピソード記憶に関わる脳部位である。エピソード記憶とは、「いつ」「どこ」で「何」が起きたかという、個人の経験に基づいた記憶を意識的に想起し、その記憶を追体験できる認知的システムである [55]。海馬では、特定の場所に対して反応する神経細胞 [56] や、特定の遅延時間を表現する神経細胞 [57] が発見されている。海馬では、これらの細胞の活動を組み合わせることで、現在の経験が内部表象されると考えられている [58, 59]。その経験が短期的な記憶として一時

的に海馬に蓄積され、それが後に再生されることで、大脳新皮質の長期記憶に置き換えられる [60, 61].

O'Keefe と Dostrovsky は、ラットの海馬内の特定の細胞が、環境中の特定の場所を通過する時に活動することを発見した [56]. これは場所細胞 (Place Cell) と呼ばれる海馬の神経細胞である. また、自身の場所だけでなく、他者の位置に応じて活動する場所細胞 (Social Place Cell) もラットやコウモリの海馬から発見されている [62, 63]. 場所細胞が活動する特定の場所は、場所受容野 (Place Field) と呼ばれる.

Fyhn らは、海馬の入出力の機能を担う内側嗅内皮質 (Medial Entorhinal Cortex) から、場所受容野が格子状に存在する格子細胞 (Grid Cell) を発見した [64]. 上記の場所細胞は点を、格子細胞は座標を表現していると考えられる. これらの位置情報を表象する細胞に加えて、方向を符号化する神経細胞が存在することが知られている. Ranck はラットの前海馬支脚 (Presubiculum) から、自身の頭部が向いている方向に受容野を持つ頭部方向細胞 (Head Direction Cell) を発見した [65, 66].

MacDonald らは、特定の遅延時間を表彰する時間細胞 (Time Cell) を発見した [57]. 時間細胞を用いて、距離を符号化する神経細胞の存在が示唆されている [67].

4.2.2 前頭前野

前頭前野は、大脳の前頭葉 (Frontal Lobe) に位置する連合野 (Association Area) である. 前頭前野は、海馬や扁桃体などの大脳辺縁系を含む様々な脳部位から入力が集まる. また、その接続は一方向ではなく双方向であることが知られている. 前頭前野の機能は多岐にわたり、認知・実行機能、意思決定、社会的行動などに関係する.

前頭前野の神経細胞の活動の計測の実験から、前頭前野では、これから行おうとする行動を先まで思い浮かべ、行動決定するメンタルシミュレーションの機能があることが知られている [68].

4.3 海馬・前頭前野モデルの先行研究

4.3.1 海馬モデル

Graves らは、ニューラルネットワークに動的で微分可能な外部メモリを追加することで、時系列を処理するタスクにおいて性能を向上する手法を提案した [69]. この提案されたモデルは、Differentiable Neural Computer (DNC) と呼ばれる. コン

コンピュータはプロセッサとメモリに分かれているため、必要に応じて動的にメモリを確保しながら処理することができる。一方で、人工ニューラルネットワークはプロセッサとメモリが一体化しており、動的なメモリの確保ができない。DNC は、ニューラルネットワークの外部に読み書き可能なメモリを追加することで、上記の問題を解決する。さらに、外部メモリの読み書きを微分可能にすることで、誤差逆伝播法と確率的勾配降下法 (Stochastic Gradient Descent : SGD) によりニューラルネットワークと外部メモリを同時に学習できる。Graves らは、この DNC のメモリ機構と海馬の機能の関連性を言及している。DNC のメモリの修正が早く、ワンショット可能である点が、海馬の CA3 及び CA1 のシナプスの長期増強 (Long-Term Potentiation) と類似すると主張している。

川内ら [70] は、エピソード記憶のため、経験の内部表象を行う海馬・嗅内皮質モデルを提案した。このモデルは、自身の速度と頭方向を入力として受け取り、速度細胞 (Speed Cell) と頭部方向細胞がそれぞれを表象する。速度と頭方向の情報は、空間の周期的な場所に対応して発火する格子細胞に与えられ、複数周期の格子細胞の出力の組み合わせにより自己位置が表象される。一方、頭方向の情報は時間細胞にも与えられ、どちらを向いていたかという事象からの経過時間が表象される。こうして処理され内部表象された時空間情報は、海馬の経路依存場所細胞 [71] に与えられ、どこ経路を通ってきたかが表象される。これらの神経細胞モデルは、スパイクニューロンモデルによって実装されている。また下留ら [72] は、このモデルをハードウェア実装のために改変し、場所細胞を抵抗ネットワークモデル [73] で実装している。

4.3.2 前頭前野モデル

前頭前野のモデルとして、レザバーコンピューティング (Reservoir Computing) [74] がしばしば用いられる。香取ら [75, 76] は、レザバー計算を利用した前頭前野モデルを提案し、予測符号化と行動計画に適用した。ここでは、レザバーコンピューティングの一種である、Echo State Network (ESN) [77] をベースとした、階層型の予測符号化モデルを構成している。

Kashyap ら [78] は、霊長類に見られる追跡眼球運動を模倣した前頭眼野 (Frontal Eye Field) モデルを提案した。このモデルは、逐次的な学習則 [79] を取り入れたレザバーコンピューティングを用いて実装される。このモデルは、たとえターゲットが隠されても、速度変化が非線形であっても、ターゲットを追跡するための眼球の速度

を即座に学習する。

4.4 提案モデル

本研究は、数回のエピソードから個人の経験に基づいた記憶の獲得を行い、記憶に基づいた予測を行うことでサービスロボットの行動計画に応用することを目指す。4.3 で述べた先行研究のうち、DNC[69] はが行うのは、本研究が目指す個人の経験に基づいた記憶の獲得とは異なる学習パラダイムと考えられる。DNC のメモリの読み書き自体はワンショットではあるが、モデル全体の最適化は SGD による膨大な回数の繰り返しが前提となっている。

一方で、川内ら、下留らの手法 [70, 72] は、エピソード記憶のための内部表象を行うモデルを提案しており、本研究が目指す個人の経験に基づく記憶の獲得に必要な機能を実装していると考えられる。本研究は、この手法を参考にしつつ、サービスロボットのシステムに搭載するために、よりシンプルな海馬モデルを実装する（川内ら、下留らのモデルでは速度と頭方向から自己位置を推定するが、サービスロボットには既に自己位置推定の機能が備わっており、本研究ではこの自己位置推定を海馬の部分的な機能として活用する）。また本研究は、前頭前野モデルの先行研究 [75, 76, 78] を参考にして、レザバーコンピューティングによる前頭前野モデルを実装する。

4.4.1 海馬モデル

川内・下留らが提案する生理学的知見に基づいた海馬・嗅内皮質のスパイクニューロンモデルを参考にした、場所細胞と時間細胞のモデルを提案する。図 4.1 に、場所細胞モデルを示す。ここでは 2 次元の長方形の空間を考え、場所細胞モデルはこの空間内を移動する自身の場所、あるいは他者の場所を表象する。細胞モデルは 2 次元平面にグリッド状に配置され、各細胞は空間を等間隔に分割してできる各領域に割り当てられる。このモデルには、ロボットの自己位置推定、あるいは物体検出によって獲得された 2 次元空間における自身／他者の座標が与えられ、各細胞は割り当てられた領域に自身／他者が存在する尤度を出力する。式 4.1 に、場所細胞モデルの i 番目の出力 p_i を示す。

$$p_i = \exp(-d_{i,j}^2/2\sigma^2), \quad (4.1)$$

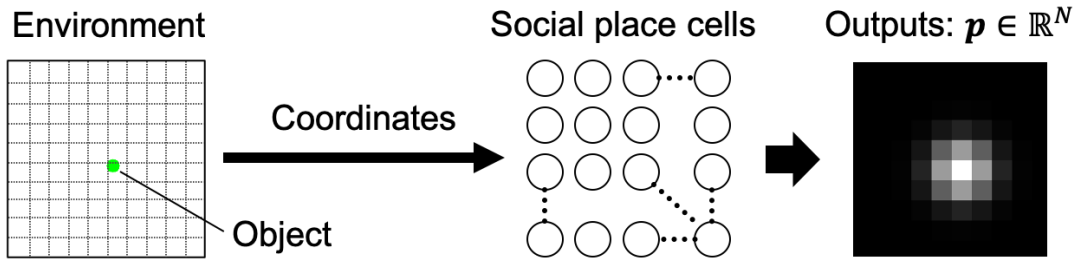


図 4.1: 海馬：場所細胞モデルの構成

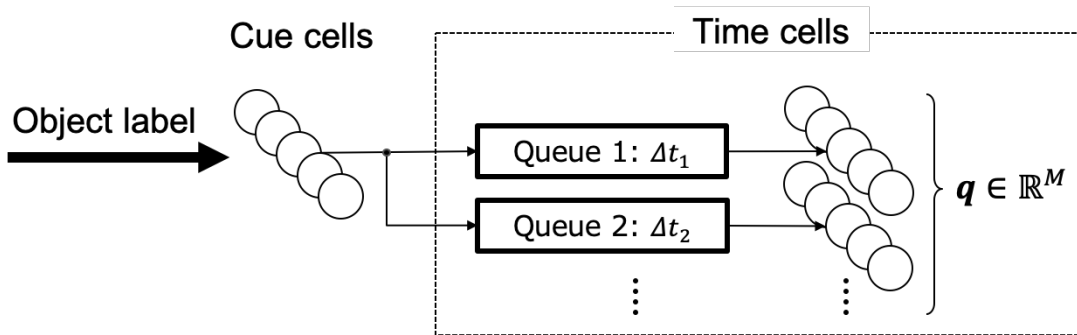


図 4.2: 海馬：時間細胞モデルの構成

ここで、 $d_{i,j}$ は i 番目のグリッドと、物体が検出されたグリッド j との距離である。 σ はガウス関数の分散である。式に示すように、場所細胞の出力はガウス関数で表現されており、物体が検出されたグリッドに対応する細胞だけでなく、その周辺の細胞も値を出力する。これは、例えばグリッド (X, Y) で物体が検出されたエピソードと、 $(X + 1, Y)$ で同じ物体が検出されたエピソードを、類似するエピソードとして統合するためである。

図 4.2 に、キュー（Cue）細胞モデルと時間細胞モデルを示す。キュー細胞モデルは、特定のイベントに反応する。イベントは例えば、自身の頭方向や検出したオブジェクトの種類などを用いる。時間細胞モデルは、キュー細胞の出力を受け取り、一定時間経過後同じ値を出力する。時間細胞モデルには複数の Queue があり、全ての Queue が入力信号を受け取る。各 Queue によって信号の保持時間 Δt_k が異なる。式 4.2 に、時間細胞の k 番目の Queue の出力 $\mathbf{q}_k(t)$ ($\mathbf{q}(t) = \{\mathbf{q}_1(t), \dots, \mathbf{q}_k(t), \dots\} \in \mathbb{R}^M$) を示す。

$$\mathbf{q}_k(t) = \mathbf{c}(t - \Delta t_k), \quad (4.2)$$

ここで、 c はキュー細胞の出力である。但し、 $c(t) = 0(t < 0)$ とする。

4.4.2 前頭前野モデル

4.3の前頭前野モデルの先行研究に倣って、レザバーコンピューティングを用いて前頭前野モデルを実装する。本研究ではレザバーコンピューティングとして、特にEcho State Network (ESN) [77]を用いる。前頭前野モデルは時系列データを学習し、学習後には外部からの入力なしで時系列データを生成する機能を有する。

図4.3に前頭前野モデルを示す。前頭前野モデルの実行モードには、学習モードと推論モードがある。学習モードでは、時刻 t のイベント入力に対して、時刻 $t+1$ のイベントを推測できるように教師信号を与えて学習する。学習はリッジ回帰 (Ridge Regression) によって行う。推論モードでは、時刻 t の自身の出力を時刻 $t+1$ の入力とするループを形成する。このループを繰り返すことにより、時系列イベントを連続的に生成することができる。但し、レザバーの初期状態を定めるため、イベントの最初の数時刻は外部からの入力を受け付け、その後には推論モードを実行する。

4.4.3 三部位モデルの統合

3章で提案した扁桃体モデルに加えて、本章で新たに提案した海馬モデル・前頭前野モデルを統合することで、個人の経験に基づいた記憶の獲得、及び記憶に基づいた予測を行うモデルを提案する。これらのモデルは組み合わせ方によって、様々なタスクに応用することができる。本研究では、他者の移動経路を学習・予測するモデル、及び自身の移動経路と価値を学習・予測し行動決定するモデルについて提案する。

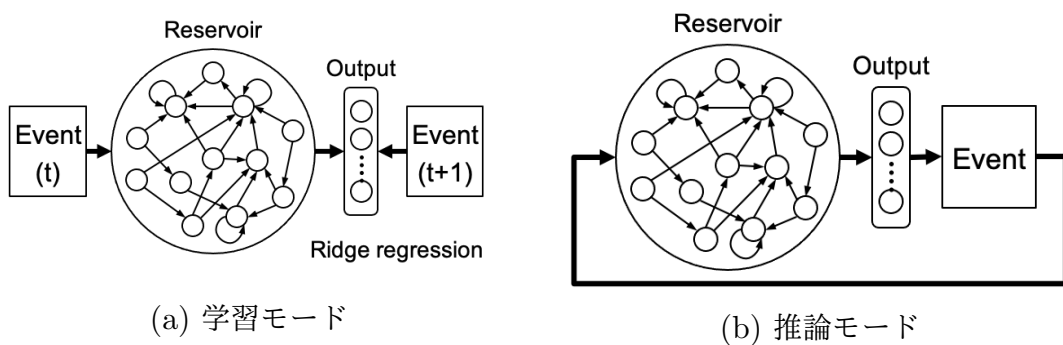


図 4.3: 前頭前野モデルの実行モード

図 4.4 に、海馬・扁桃体・前頭前野を統合した、他者の移動経路を学習・予測するモデルを示す。このモデルは、感覚野が検出したオブジェクトの種類の情報（オブジェクトのラベル）と場所の情報（オブジェクトの座標値）を受け取る。最初に、オブジェクトの種類情報が扁桃体モデルに入力される。扁桃体モデルはオブジェクトの価値を事前に古典的条件付けにより学習しており、入力されたオブジェクトの価値を推定する。ここで、重要と判断されたオブジェクトの種類情報、及び場所の情報のみが海馬モデルへ入力される。種類情報は海馬のキュー細胞モデルへ、場所情報は海馬の場所細胞モデルへ入力される。この場所細胞モデルは、他者の場所を表象するもので、Social Place Cell の役割を持つ。時間細胞及び場所細胞モデルは、これらの情報を統合し、イベントベクトルとして出力する。ここではイベントベクトルを、場所細胞の出力 $\mathbf{p} \in \mathbb{R}^N$ と時間細胞の出力 $\mathbf{q} \in \mathbb{R}^M$ を用いて、ベクトルの全ての組み合わせの要素積 $\mathbf{p}^T \mathbf{q}$ で定義する。イベントベクトルは前頭前野モデルへ入力される。前頭前野モデルは、時刻 t のイベントが与えられた時、時刻 $t+1$ のイベントを出力するように時系列イベントを学習する。この学習を経て、前頭前野モデルは推論モードによって、連続的なイベントの時系列を予測することができるようになる。

図 4.5 に、海馬・扁桃体・前頭前野を統合した、自身の移動経路と価値を学習・予測し行動決定するモデルを示す。ここでは、自身がある環境をナビゲーションする際に、複数の経路の選択枝の中から最適な経路を選択する問題を仮定する。図の黒線は学習モードの処理フロー、赤線は推論モードの処理フローを示す。学習モードでは、自己位置推定によって獲得した自身の位置座標を受け取り、海馬の場所細胞モデルに入力する。海馬モデルが出力するイベントベクトルは前頭前野モデルと扁桃体モデルに入力される。前頭前野モデルは時刻 t のイベントが与えられた時、時刻 $t+1$ のイ

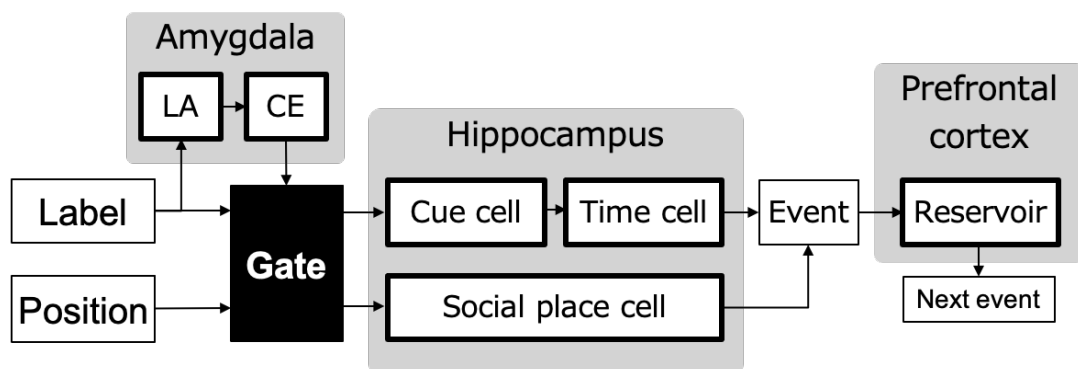


図 4.4: 他者の移動経路を学習・予測するモデルの構成

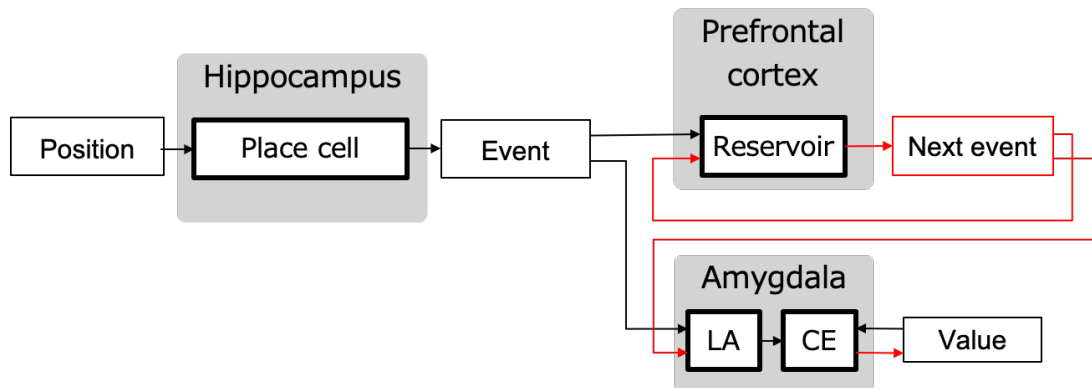


図 4.5: 自身の移動経路と価値を学習・予測し行動決定するモデルの構成
黒線は学習モード・赤線は推論モードの処理フローを示す

イベントを出力するように時系列イベントを学習する。扁桃体モデルはイベントベクトルが与えられると同時に、今自身が置かれている状況の報酬が与えられるものとし、イベントベクトル（ここでは場所）の価値を推定できるように古典的条件付け学習する。推論モードでは、前頭前野モデルがイベントベクトルの時系列を生成する。ここで生成されたイベントベクトルは扁桃体モデルに入力され、扁桃体モデルはイベントの価値を推定する。これにより、前頭前野モデルが複数のイベントの系列を生成し、それぞれのイベントの系列の価値を扁桃体モデルが推定することで、複数の行動の選択肢の中から最適の（最大の価値を持つ）行動選択をすることができる。

4.5 実験

4.5.1 他者の移動経路を学習・予測するモデル

提案モデルを評価するために、以下のようなタスクを設定した。正方形の部屋の中で、いくつかのオブジェクトが出現し移動する。但し、オブジェクトの移動先はその種類によって変わる、とした。このとき提案モデルが、出現したオブジェクトの経路を推論モードにより正しく予測できるかどうかを評価した。

オブジェクトが部屋の中で移動する状況をロボットシミュレータによって再現し、シミュレータの情報を提案モデルに入力した。出現するオブジェクトは、人 A、人 B、その他のオブジェクトの 3 種類とした。人 A・B は重要なオブジェクトとし、扁桃体モデルに重要度を事前学習させた（重要なオブジェクトの重要度は 1.0、それ以

外は 0.0). 人 A は位置 O に出現し, 位置 α に向かって移動, 人 B は位置 O に出現し, 位置 β に向かって移動するように設定した. その他のオブジェクトは, 上記以外の場所に配置した. 人 A が出現し移動するエピソード (エピソード A) と, 人 B が出現し移動するエピソード (エピソード B) を繰り返して, モデルに入力した. 各エピソードでオブジェクトの出現場所は少しずつ変動するようにした. 位置 O の座標を (X_O, Y_O) とし, 正方形の部屋の辺の長さを L とした時, オブジェクトの出現場所は $(X_O - 0.1L, Y_O - 0.1L)$ から $(X_O + 0.1L, Y_O + 0.1L)$ の範囲で変動するようにした.

図 4.6, 4.7 に, あるエピソードを実行したときのロボットシミュレータと, 学習済みの提案モデル (50 エピソードを学習) の出力を示す. 図 4.6 の 1 行目は, 人 A が位置 O から α に移動するエピソードをロボットシミュレータが実行している様子を示す. 図 4.7 の 1 行目は, 人 B が位置 O から β に移動するエピソードをロボットシミュレータが実行している様子を示す. 両図の 2 行目は場所細胞モデルの出力, 3 行

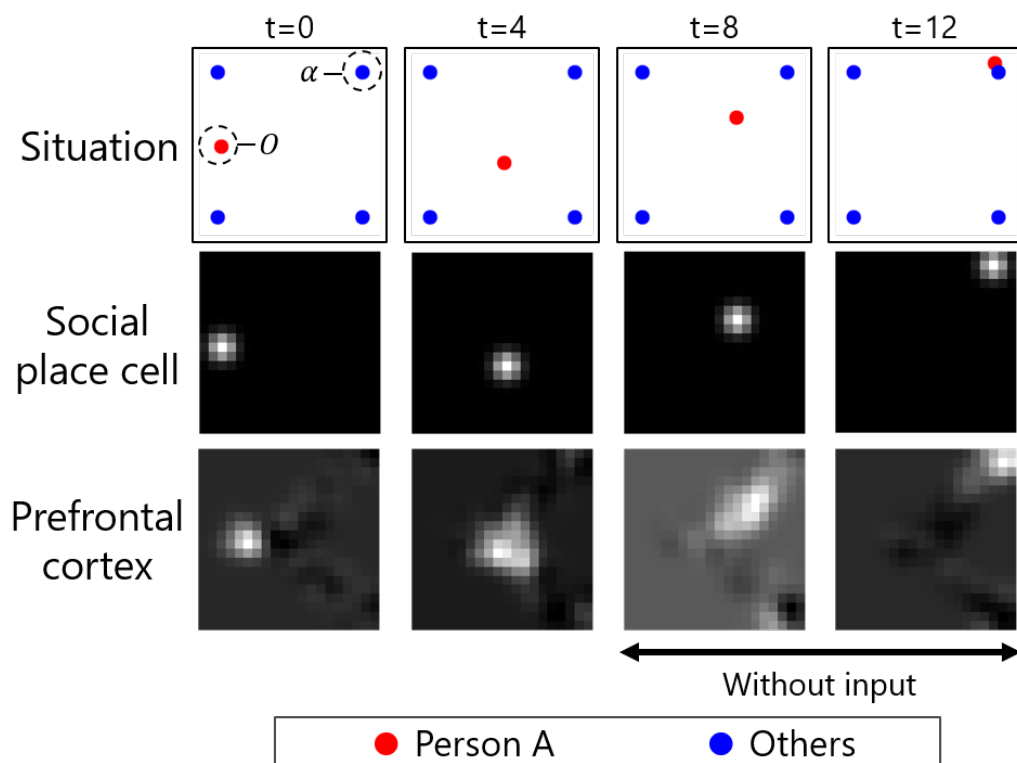


図 4.6: ロボットシミュレータと提案モデルの出力: エピソード A

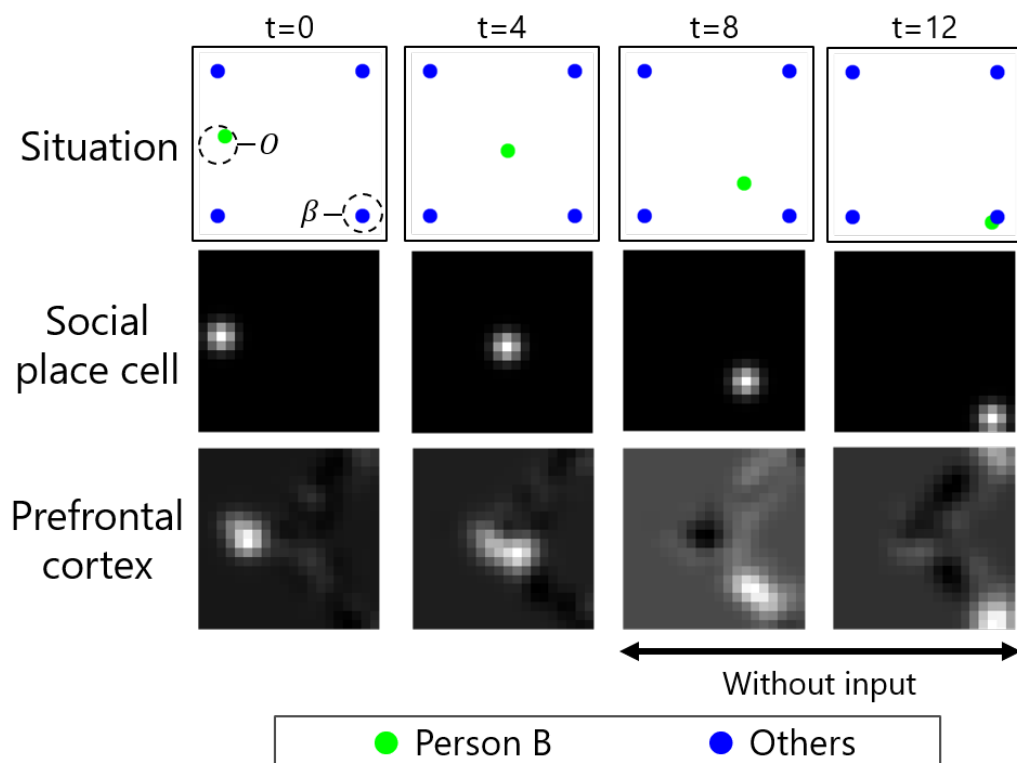


図 4.7: ロボットシミュレータと提案モデルの出力：エピソード B

目は前頭前野モデルの出力を示す。各エピソードの長さは 12 ステップとし、最初の 4 ステップまでは海馬モデルからの出力を受け付け、残りのステップは推論モードを実行した。場所細胞の出力（白いピクセルは高い値を示す）が人の位置を追従していることから、扁桃体モデルが重要なオブジェクトである人 A・B のみを抽出し、海馬モデルに入力したことがわかる。さらに、前頭前野モデルは推論モードにおいても人の位置を正しく推定できたことがわかる。

図 4.8 に、提案モデルの予測誤差（5 回の平均）を示す。ここで予測誤差は、 20×20 のグリッドにおける、人の予測位置と実際の位置の距離とした。図より、予測誤差の平均値が 1.41 よりも小さいことから、提案モデルの予測位置は、例えば推論モードにおいても実際の位置の 8 近傍内のグリッドにあったことがわかる。

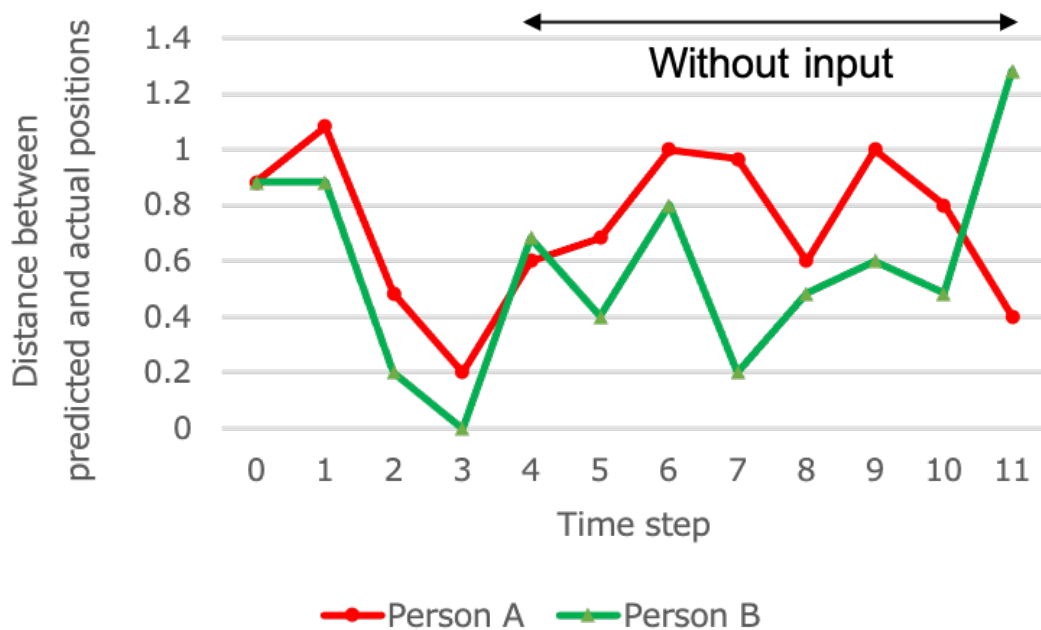
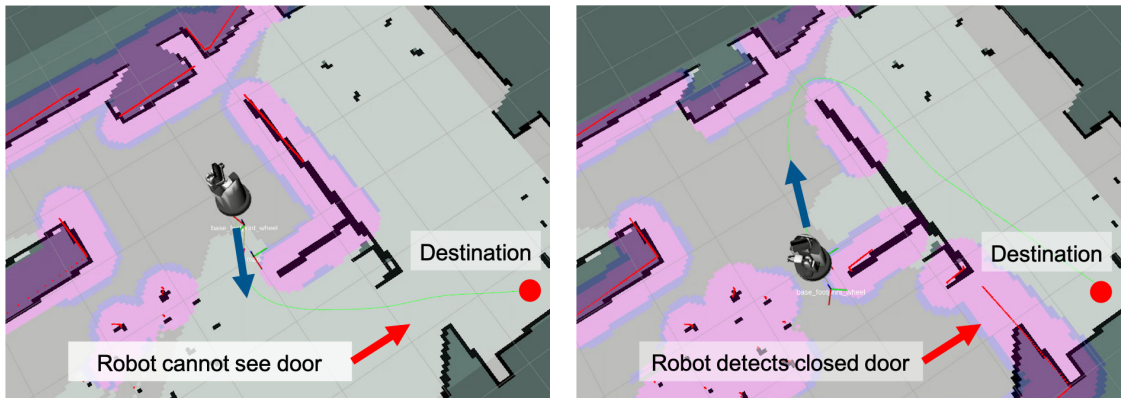


図 4.8: 提案モデルの予測誤差

4.5.2 自身の移動経路と価値を学習・予測し行動決定するモデル

提案モデルを評価するために、以下のようなタスクを設定した。このタスクは、既存のロボットのナビゲーションシステムを特定の環境で動作させると実際に生じる問題を元としている。

図 4.9 に、ロボットがナビゲーションを行う際に用いる地図を示す。図中の黒い線は、部屋の壁や家具などを示している。この地図情報は事前にロボットが Laser Range Finder (LRF) などを使って取得するもので、黒い領域は静的な障害物（タスク中一貫して不変）として取り扱われる。一方、図中のピンクで描かれる領域は、ロボットが LRF などを使って動的に検出した障害物を示す。例えば、人が検出されるとその周辺がピンクで描かれる。人はその場に居続けるわけではないので、人が検出されなくなるとしばらく時間が経過すると、ピンクの領域は消える。ロボットがナビゲーションを行う際には、このピンクの領域を避けながら移動することになる。この図は、ロボットが図右下にある目的地 (Destination) に向かってナビゲーションを試みている様子を示している。図 4.9-(a) は、ロボットの現在地 (図中央付近) から目



(a) ドアが死角にあり
閉じられたドアを検出できない

(b) 閉じられたドアを検出したため
別の経路を選択する

図 4.9: ナビゲーションの問題

的にナビゲーションをする場合、図の下側の通路が最短経路であるため、ロボットは図の青色の矢印で示す方向に移動を始める様子を示す。その後、図 4.9-(b) に示すように、ロボットが下側の通路に近づくと、ロボットは動的な障害物を検知する（ロボットが事前に地図情報を取得した際には下側の通路のドアが開いていたが、これが閉められたため、ドアを動的な障害物として判断したという状況を示す）。ロボットは下側の通路が塞がれているため、図の青色の矢印で示す方向に移動を始める。しばらく進むと、ロボットは下側の通路を塞ぐ動的な障害物が見えなくなるので、下側の経路を最短経路を判断し、下側に向かって移動を始める。以下、ロボットはこの行動を繰り返して目的地にたどり着けない状況に陥る。ここで、ロボットにドアが閉まっていたという個人の経験に基づいた記憶を保持し、記憶に基づいた行動決定をすることができれば、この問題を解決することができると思われる。

上記の問題を解決するために、提案モデルを用いた。始めに、ロボットに上側の通路と下側の通路を通過して目的地にたどり着くナビゲーションを行わせた。このとき、ナビゲーションに失敗した時（すなわち下側の経路を選択するも、ドアがしまっていたため行動を変更した時）に扁桃体モデルに報酬（危険度）を与えるようにした（図 4.5 の Value から伸びる黒矢印）。ここで与える報酬の大きさは 1 とした。扁桃体モデルはこの時、現在の場所と報酬を関連づけるように学習する。同時に、前頭前野モデルに移動経路を予測できるように学習を行わせた。この学習が完了した後、ロボットに初期位置からメンタルシミュレーションを実行させ、下側の経路（経路 1）と上

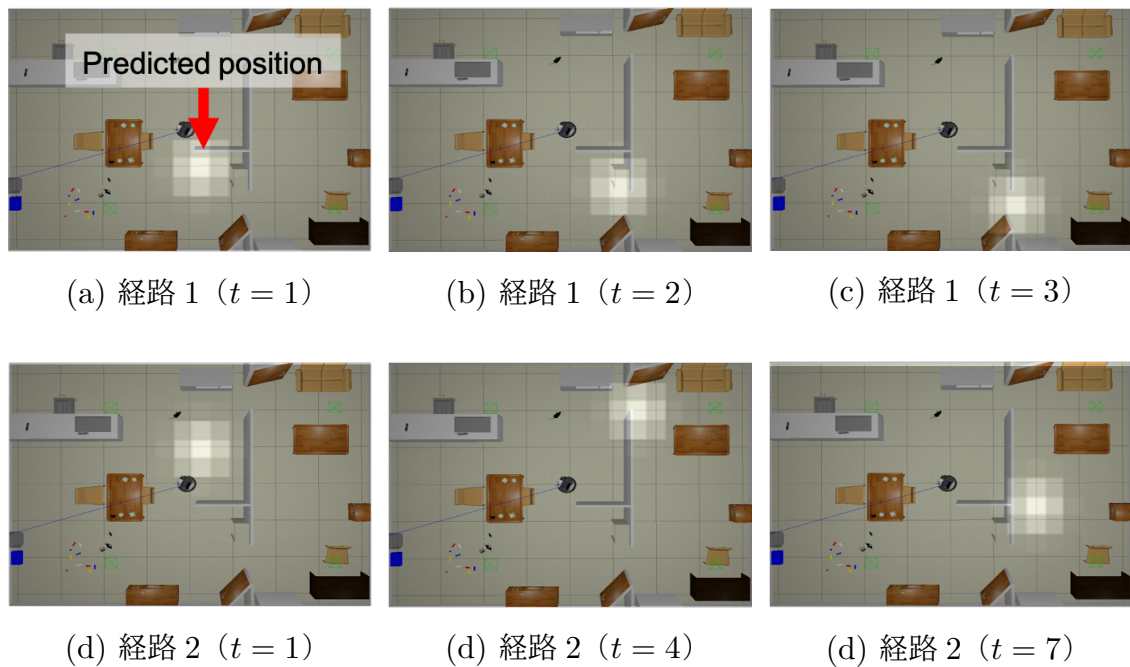


図 4.10: 提案モデルのメンタルシミュレーションにおける経路生成（図中の白いグリッドは予測位置を示す）

側の経路（経路 2）を生成させた．図 4.10 に，メンタルシミュレーションの結果を示す．生成した経路を扁桃体モデルが価値判断することで，経路の価値を計算することができる（図 4.5 の Value へ伸びる赤矢印）．図 4.11, 4.12 に，扁桃体モデルが出力した各経路の価値を示す．尚，扁桃体モデルの出力の活性化関数はソフトマックス関数とした（値域：0-1）．ここで求めた価値の平均に基づいて，どちらの行動が最適かを判断することができる．今回は扁桃体モデルの出力は危険度を示すものとし，危険度が低い経路 2 が選択された．最適な行動を求めた後，ロボットはメンタルシミュレーションによって取得した座標値を経由地としてナビゲーションすることで，目的地にたどり着くことができた（参考動画：<https://youtu.be/VBhbwxbpExk>）．

4.6 本章のまとめ

本章では，3 章で着目した扁桃体に加えて，海馬，前頭前野の機能を統合した AI モデルを提案し，個人の経験に基づいた記憶の獲得，及び記憶に基づいた予測を行った．特に，他者の移動経路を学習・予測するタスクと，自身の移動経路と価値を学

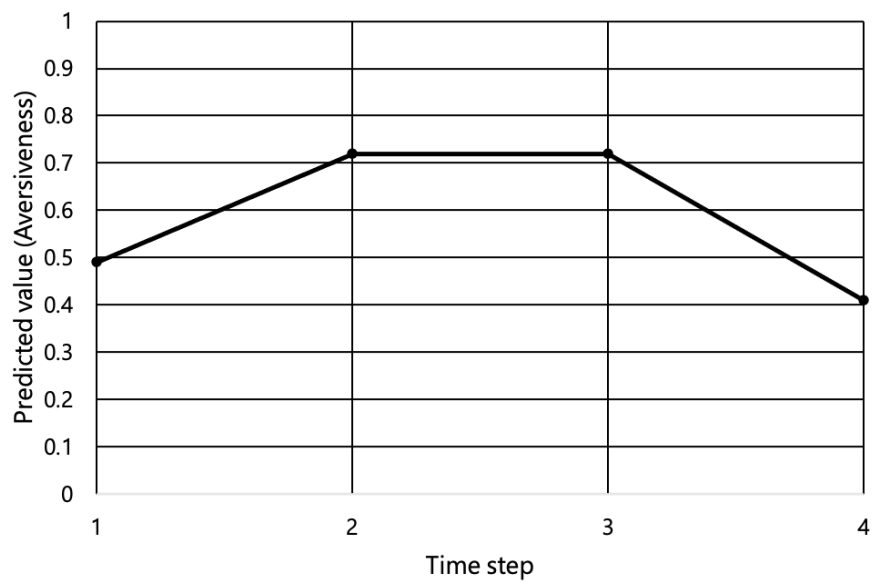


図 4.11: 経路 1 の価値（危険度）の推移（平均：0.64）

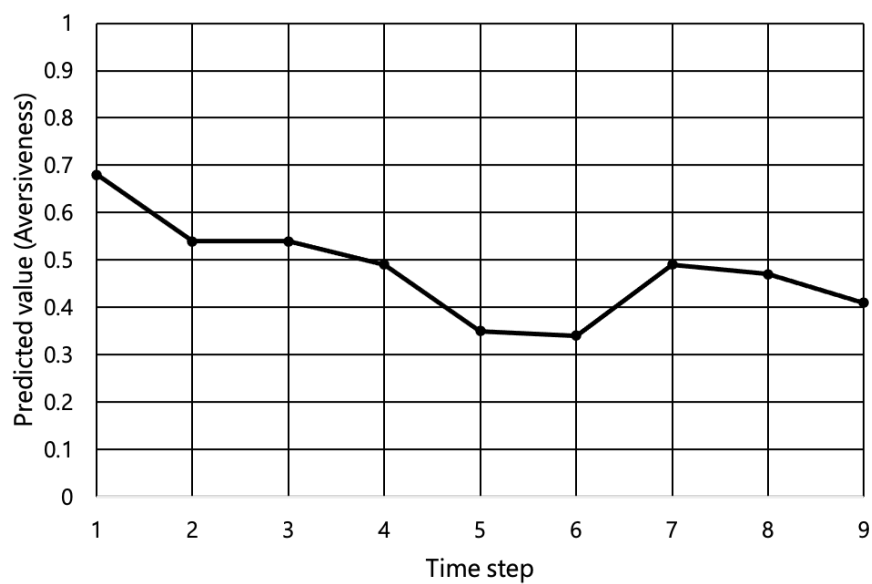


図 4.12: 経路 2 の価値（危険度）の推移（平均：0.48）

習・予測するタスクで、提案モデルを評価し、いずれの場合も個人の経験に基づいた記憶の獲得、及び記憶に基づいた予測を行うことができた。

しかしながら、ここで提案したモデルは、他の多くの AI モデルと同様に、特定の

タスクのみに対応する。ヒトのように考える AI の実現のためには、特定のタスクのみならず、様々なタスクに対応できる汎用的な AI モデルの実現が求められる。次章では、本章で提案した AI モデルを拡張し、複数のタスクを単一のモデルで学習できる手法を提案する。

第 5 章

マルチタスク学習への適用

5.1 本章の概要

本章では、4 章までに提案した AI モデルを、マルチタスク学習に適用する手法を提案する。本研究が提案する海馬・扁桃体・前頭前野の機能を統合した AI モデルにおいて、学習の中心となるのは前頭前野モデル（レザバーコンピューティング）である。したがって、本提案モデルをマルチタスク学習に適用させた時に、破滅的忘却の影響を強く受けるのはレザバーコンピューティングのリードアウトである。本章では、最初のステップとして破滅的忘却を回避するレザバーコンピューティングモデルを提案し、次にそのモデルを前頭前野モデルとして他の脳部位モデルと統合する方法を提案する。

5.2 マルチタスク学習

本研究はサービスロボットの実用化のため、単一のモデルで複数のタスクを達成できる汎用 AI モデルの実現を目指す。しかしながら、様々なタスクを単一のモデルで学習する場合、破滅的忘却が問題となる。この問題を回避する一つの方法として、モデルをいくつかのモジュールに分割し、タスクによってモジュールを使い分ける方法がある。モデルをモジュール化することは、破滅的忘却の問題を回避するだけでなく、他の未学習タスクを解決することに活用可能である。類似するタスクを解決するモジュールは互いに類似するパラメータを持つと仮定すると、類似するタスクを学習したモジュールを別の類似するタスクに転用することで学習の効率が上がると考えら

れる。このためには、タスクの類似性を求める手法、さらにモジュールがタスクの類似度に応じて自己組織化する仕組みが必要となる。

5.3 マルチタスク学習の先行研究

Fernando ら [80] は、複数のモジュール化された DNN を組み合わせて構成する PathNet を提案した。PathNet では、タスクに応じて、DNN モジュールを接続する Pathway を切り替えることでマルチタスク学習に対応している。この Pathway の選択は遺伝的アルゴリズムによって最適化される。

河合ら [81] は、タスクの誤差の最小化に加えて、ネットワークの結合コストを最小化するように制約をかけることで、ネットワーク内部にモジュール構造が現れるようにし、タスクに応じてモジュールを切り替えてマルチタスク学習に対応する手法を提案した。ここで提案するネットワークは Echo State Network がベースとなっている。また、結合コストの最小化の最適化手法には遺伝的アルゴリズムが導入されている。

5.4 提案モデル

本研究は、数回のエピソードから個人の経験に基づいた記憶の獲得を行い、記憶に基づいた予測を行う AI モデルの実現を目指す。5.3 で述べた先行研究のうち、PathNet[80] は DNN による実装であり、本研究が着目する学習とパラダイムが異なる。河合らの手法 [81] は、本研究が前頭前野モデルとして採用しているレザバーコンピューティングがベースとなっているが、遺伝的アルゴリズムによるモジュールの最適化のために、多くの繰り返し学習が前提となっており、本研究が着目する学習とパラダイムが異なる。

本研究は、少ない学習回数でタスクの学習を完了するために、遺伝的アルゴリズム以外の最適化手法でマルチタスク学習に対応する。5.3 で述べた先行研究のようにネットワークをモジュール化するが、ここではレザバーコンピューティングに複数のリードアウトを設けることで対応する。さらに、タスクの類似度という指標を導入し、タスクの類似度に応じてリードアウトを選択・学習する手法を提案する。

5.4.1 モデル構成

図 5.1 にマルチタスク学習のためのレザバーコンピューティング：Reservoir Computing with Self-Organizing Multi-readout (RC-SOM) を示す。このモデルはレザバー (Reservoir)・マルチリードアウト (Multi-readout)・類似度マップ (Task Similarity Map) によって構成される。一般的なレザバーコンピューティングと異なる点は、第一に複数のリードアウトを備えている点である。これにより、単一のネットワークでありながら、リードアウトを切り替えて動作することで、複数のタスクが与えられた際にも破滅的忘却を回避して学習することができる。また類似度マップは、与えられたタスクに対してどのリードアウトを用いるのかを決定するために用いる。類似度マップは SOM によって実装され、与えられたタスクの類似度を表現する地図を形成する。SOM のノード数はリードアウトの個数と一致しており、SOM の各ノードとリードアウトが対応付けられる。タスクが与えられると、SOM の勝者決定アルゴリズムにより勝者ノードを決定し、勝者ノードに対応するリードアウト (勝者リードアウト) をタスクに用いる。リードアウトの結合荷重を更新する際には、単に勝者リードアウトに接続する荷重値だけではなく、その近傍のリードアウトの荷重値も更新する。この結果、リードアウトにおいても SOM の類似度マップに現れるような空間的な連続性が生まれる。この性質を活用することで、未知のタスクの対応や、少量データによる微調整で学習を完了する転移学習に利用できる。既存の ESN は、ここで提案した RC-SOM の特殊形であるといえる (SOM のタスクの分類によるリードアウトが切り替えがない)。

5.4.2 アルゴリズム

以下に RC-SOM のアルゴリズムを示す。ここではタスクとして、系列長 T の時系列データ $\mathbf{u}(1), \dots, \mathbf{u}(t), \dots, \mathbf{u}(T)$ が与えられるものとする。SOM の参照ベクトル $\mathbf{m}_{i,j}$ を用いて、このタスクに対する勝者リードアウトを決定する。

$$(i_c, j_c) = \arg \min_{i,j} \|f(U) - \mathbf{m}_{i,j}\|, \quad (5.1)$$

ここで、 $f(U)$ はタスクの特徴量を示す静的な値である。 $f(U)$ は例えば、時系列データを時間方向に結合したベクトルや、時系列データのスペクトラムなどを用いることができる。また、与えられた時系列データを直接変換するのではなく、時系列デー

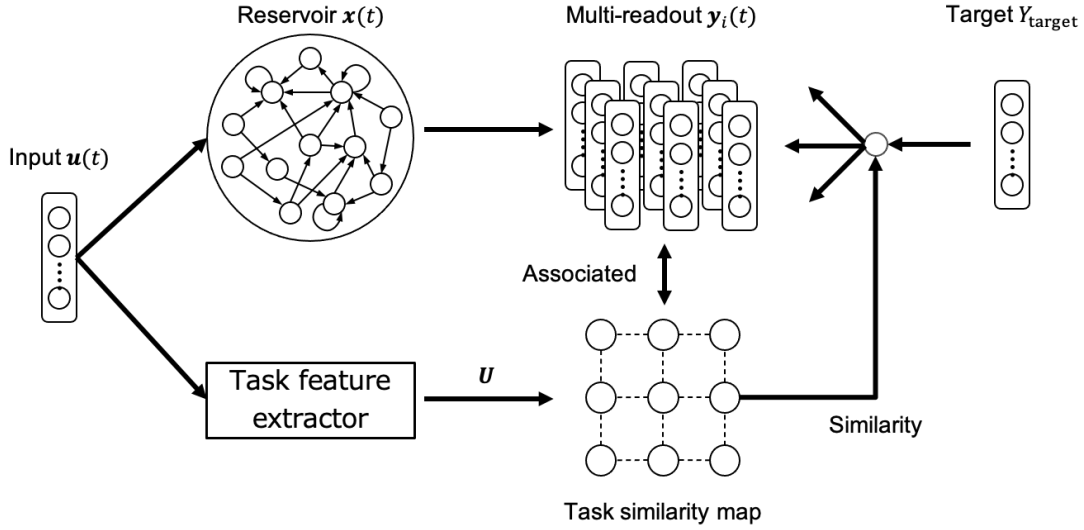


図 5.1: RC-SOM の構成

タと同時に得られた別の特徴量を用いることも有効である（移動経路の時系列データをレザバーが学習し、タスクの特徴量として経路で検出したオブジェクトの情報を用いる、など）。SOM の参照ベクトルは、以下の計算式によって更新される。

$$\Delta \mathbf{m}_{i,j} = \alpha (f(U) - \mathbf{m}_{i,j}) \exp(-d_{i,j}^2 / 2\sigma^2), \quad (5.2)$$

$$d_{i,j} = \sqrt{(i - i_c)^2 + (j - j_c)^2}, \quad (5.3)$$

$$\mathbf{m}_{i,j}^{\text{new}} = \mathbf{m}_{i,j}^{\text{old}} + \Delta \mathbf{m}_{i,j}, \quad (5.4)$$

ここで、 α は学習係数、 σ はガウス関数の分散である。 $d_{i,j}$ は、SOM の 2 次元グリッド上の距離を示す。

レザバーには、時系列データ $\mathbf{u}(t)$ を入力する。レザバーの内部状態 $\mathbf{x}(t)$ は、以下の計算式によって算出される。

$$\mathbf{x}(t+1) = \phi[(1 - \delta)\mathbf{x}(t) + \delta\{\mathbf{u}(t) \cdot W_{\text{in}} + \mathbf{x}(t) \cdot W_{\text{rec}}\}], \quad (5.5)$$

ここで、 δ は漏れ率、 $\phi()$ は非線形の活性化関数を示す。 W_{in} 、 W_{rec} はそれぞれ入力層とレザバー層の結合荷重行列、レザバー層の再帰結合荷重行列を示す。これらの結合荷重行列はランダムに設定する。特に、 W_{rec} は Echo State Property[82] を満たすように、行列のスペクトル半径が 1 未満になるように設定する。

リードアウト $\mathbf{y}_{i,j}$ は、以下の計算式によって算出される。但し、タスクに用いるのは勝者リードアウト \mathbf{y}_{i_c,j_c} である。

$$\mathbf{y}_{i,j} = \mathbf{x} \cdot W_{\text{out},i,j}, \quad (5.6)$$

ここで、 $W_{\text{out},i,j}$ は、リードアウト (i,j) の結合荷重行列である。

リードアウトの結合荷重行列は、目標値 Y_{target} を用いて、以下の計算式によって更新される。

$$W_{\text{opt}} = (X^T X + \lambda E)^{-1} X^T Y_{\text{target}}, \quad (5.7)$$

$$\Delta W_{\text{out},i,j} = \beta (W_{\text{opt}} - W_{\text{out},i,j}) \exp(-d_{i,j}^2 / 2\sigma^2), \quad (5.8)$$

$$W_{\text{out},i,j}^{\text{new}} = W_{\text{out},i,j}^{\text{old}} + \Delta W_{\text{out},i,j}, \quad (5.9)$$

ここで、 X はレザバーの内部状態の時系列を示す行列 $[\mathbf{x}(1)^T \dots \mathbf{x}(T)^T]^T$ である。式 (5.7) はリッジ回帰による、目標値 Y_{target} に対する最適な結合荷重行列 W_{opt} の算出式である。 λ はリッジ回帰の正則化項の係数を示し、 E は単位行列を示す。式 (5.8) は、各リードアウトの更新量を定める式で、勝者リードアウトを中心としたガウス関数によってその更新量を決定する。 β は学習係数を示す。

5.4.3 海馬モデルとの統合

図 5.2 に、RC-SOM（前頭前野モデルとして）と 4 章で提案した海馬モデルを統合したモデルを示す。このモデルを用いることで、自身の複数の移動経路を学習し想起することができる。ここでは、自身がある環境の中を移動することを想定する。このモデルは、自己位置推定によって獲得した自身の位置座標と頭方向を受け取り、それぞれ場所細胞モデルと時間細胞モデルに入力する。場所細胞の出力はレザバーに入力され、レザバーは場所細胞の出力系列を予測できるように学習する。時間細胞は頭方向の変化の順番、すなわちどのような順序で方向転換をしたかという文脈を表象する。この文脈の情報が SOM に入力され分類される。この SOM の分類結果に応じて、タスクに用いるリードアウトを選択する。これは、単一のリードアウトを用いて複数の移動経路を学習する場合、破滅的忘却が起こるため、文脈に合わせて学習するリードアウトを切り替えることが目的である。

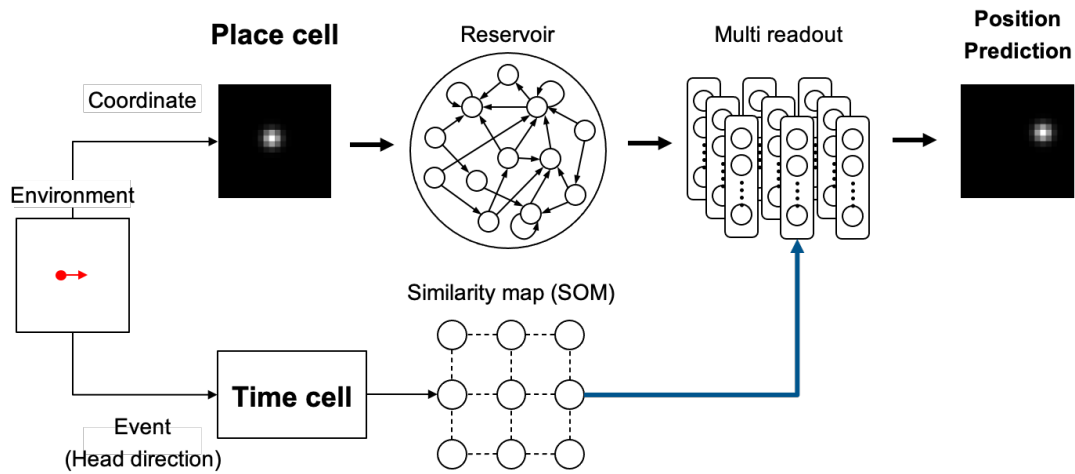


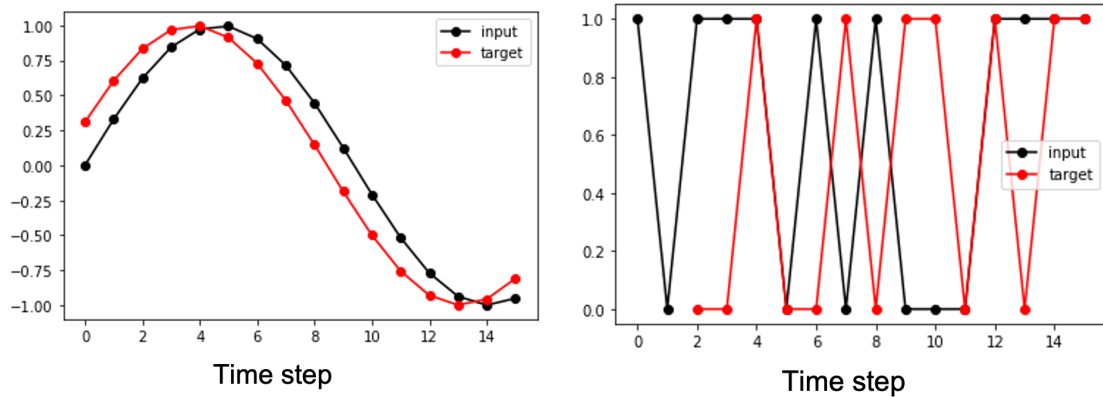
図 5.2: RC-SOM と海馬モデルの統合

5.5 実験

5.5.1 RC-SOM

ここでは提案するモデルがマルチタスク学習において破滅的忘却を回避できるかどうかを確認するため、2種類の時系列タスク：正弦波予測タスク，パリティチェックタスクを用意し，マルチタスク学習を行った．尚，このタスクは先行研究 [81] で用いられた方法を参考にした．正弦波予測タスクでは， $u_{sw}(t) = \sin(\pi t/10)$ を入力とし， $y_{sw_target}(t) = u_{sw}(t+1)$ をターゲット信号とした．パリティチェックタスクでは，バイナリのランダムシーケンス $u_{pc}(t)$ を入力とし， $y_{pc_target}(t) = \{\sum_{b=1}^B u_{pc}(t-b)\} \bmod 2$ (B はパリティチェックのビット幅) をターゲットとした．各タスクの時系列長は 16 ステップとし，これらの両タスクを交互に RC-SOM モデルに与えた．タスクの入力は 10 回繰り返した．パリティチェックタスクは，ビット幅を $B = 3$ とした．図 5.3 に，2 種の時系列タスクの入力信号（黒線）とターゲット信号（赤線）を示す．

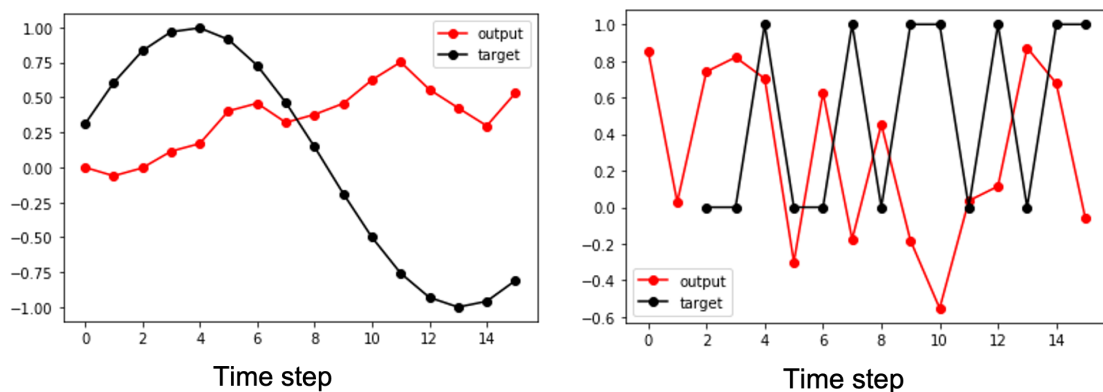
ここでは，単一のリードアウトを持つ ESN と，提案する RC-SOM モデルの比較を行った．いずれもレザバーのノード数は 100 とし，RC-SOM モデルのリードアウトは 3×3 とした．タスクの類似度として，時系列データを時間方向に結合したベクトルを SOM に入力し，勝者リードアウトを選択した．



(a) 正弦波予測タスク

(b) 3ビットパリティチェックタスク

図 5.3: 2 種の時系列予測タスク

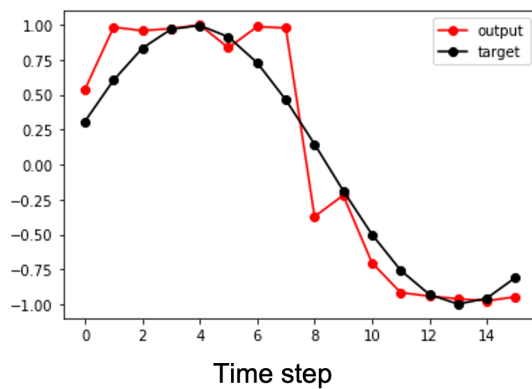


(a) 正弦波予測タスク

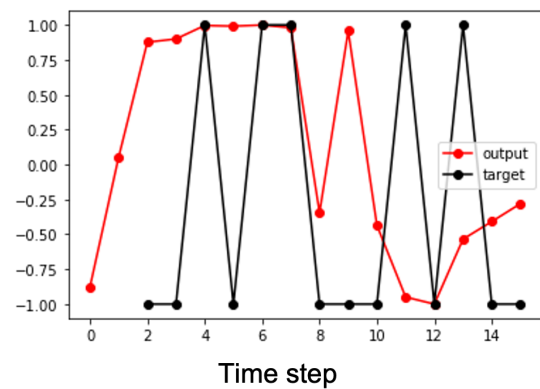
(b) 3ビットパリティチェックタスク

図 5.4: 単一リードアウトを持つ ESN のタスクの出力

図 5.4 に、単一のリードアウトを持つ ESN の各タスクの出力を示す。ここでは、各タスクの入力が完了する毎にリッジ回帰を行い学習した。その結果、破滅的忘却が生じ、2 種類の時系列タスクの両立に失敗した。次に、タスクの入力順序による学習パラメータの上書きが生じないように 2 種類のタスクの入力を先に行い、両方のタスクの誤差を最小化するように、まとめてリッジ回帰を実行した（バッチ学習）。図 5.5 に、単一のリードアウトを持つ ESN の各タスクの出力を示す。図 5.4 と比較して、誤差は小さくなっているが、2 種類のタスクは類似しないため、単一のリードアウトでこれらのタスクを両立することが困難であることがわかった。

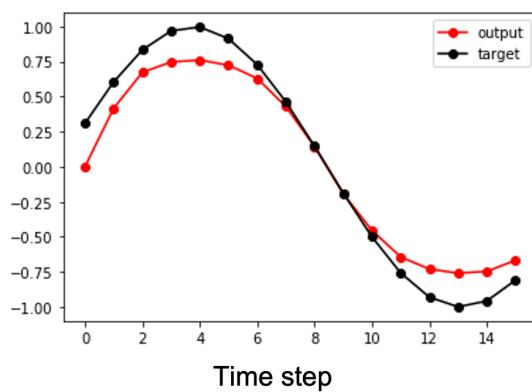


(a) 正弦波予測タスク

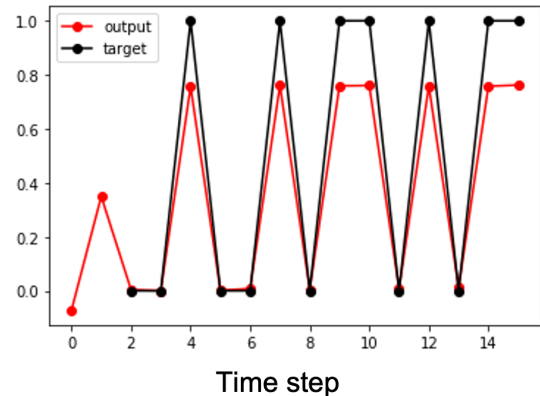


(b) 3ビットパリティチェックタスク

図 5.5: 単一リードアウトを持つ ESN のタスクの出力 (バッチ学習)



(a) 正弦波予測タスク

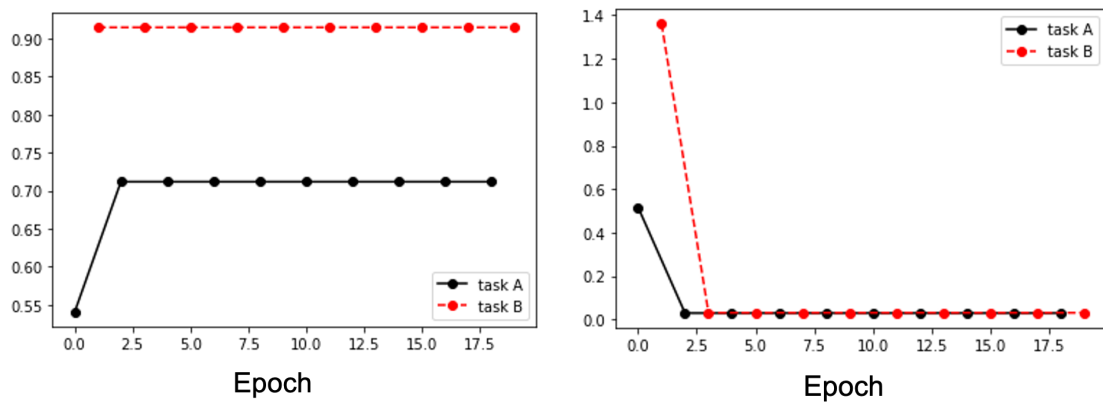


(b) 3ビットパリティチェックタスク

図 5.6: 提案モデルのタスクの出力

図 5.6 に、RC-SOM モデルの各タスクの出力を示す。ここではタスクに応じて異なるリードアウトが選択されたため、破滅的忘却を回避し両タスクとも予測することが可能となった。

図 5.7 に、2つのタスクを学習中のモデルの出力とターゲット信号との誤差（平均二乗誤差）の推移を示す。単一リードアウトの ESN の場合、破滅的忘却が生じているため学習は収束しなかった。一方で、提案手法では破滅的忘却を回避して、学習が収束したことがわかる。

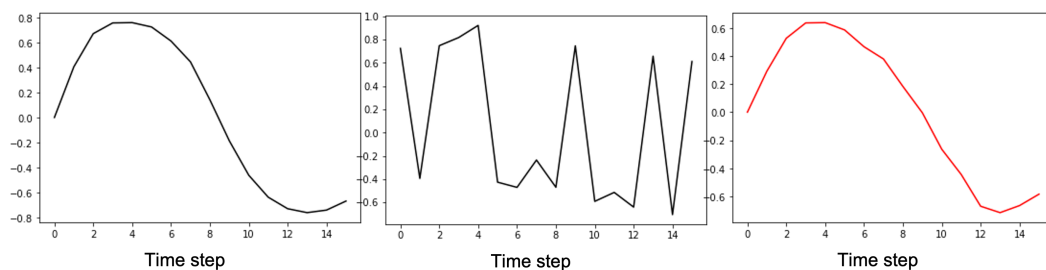


(a) 単一リードアウト ESN

(b) 提案手法

図 5.7: 誤差の推移

ここまでは、提案モデルがマルチタスク学習において、破滅的忘却を回避できるかどうかを確認した。上記に加え、学習済みのモジュールが他のタスクに転用可能かどうかを確認するための実験を行った。図 5.8 に、タスク A を入力したときの各リードアウトの応答を示す。ここでは、(a) タスク A を与えた時勝者になったリードアウト、(b) タスク B を与えた時勝者になったリードアウト、(c) 両リードアウトの中間に位置するリードアウトの応答を確認した。図に示すとおり、タスク A の勝者リードアウトはタスク A の波形を再現し、タスク B の勝者リードアウトはタスク A の波形を再現できなかった。興味深いのは、直接的に選択されなかった両リードアウトの中間にあったリードアウトがタスク A の波形をおおよそ追従できたことである。図 5.9 に、タスク B を入力したときの各リードアウトの応答を示す。図に示すとおり、



(a) A の勝者

(b) B の勝者

(c) A・B の中間

図 5.8: 各リードアウトのタスク A に対する応答

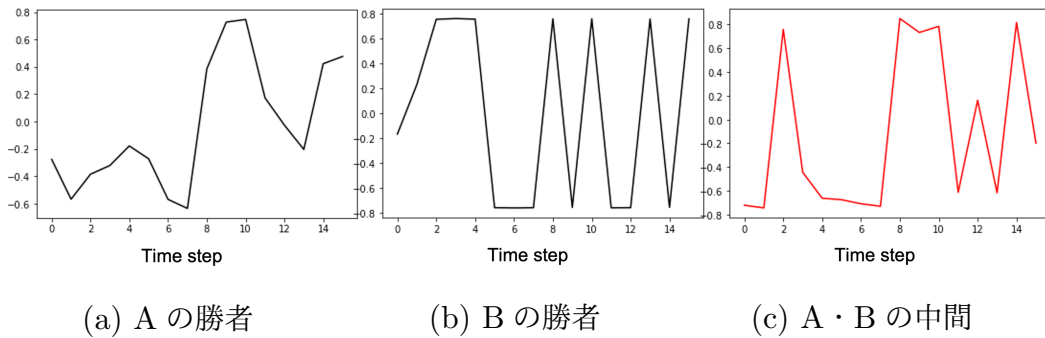


図 5.9: 各リードアウトのタスク B に対する応答

タスク B の勝者リードアウトはタスク B の波形を再現し、タスク A の勝者リードアウトはタスク B の波形を再現できなかった。先ほどと同様に、両リードアウトの中間にあったリードアウトがおおよそタスク B の波形を追従した。以上のことから、提案モデルのリードアウトは自己組織化機能を持っており、近くに配置されたリードアウトは似た応答を、遠くに配置されたリードアウトは異なる応答を示す。この性質を活用することで、未知のタスクを効率よく学習することができると考えられる。

5.5.2 RC-SOM と海馬モデルの統合

提案モデルを評価するために以下のようなタスクを設定した。図 5.10 に示す 8 の字迷路において、自身が移動してきた経路を想起できるかどうかを評価した。経路のパターンは図に示す 4 パターン：(a) 左側時計回り、(b) 左側反時計回り、(c) 右側時計回り、(d) 右側反時計回りとした。タスクでは、上記の 4 つの経路のパターンの中から 1 つが選択され、選択された経路を 1 週する。学習モードでは、同じ経路をもう 1 周し経路の時系列を学習する。推論モードでは、通った経路を想起する。両モードにおいて、モード実行直前（すなわち経路 1 周完了後）に時間細胞が算出した文脈情報を SOM に入力し、勝者リードアウトを選択した。図 5.11 に、本タスクのターゲット信号を示す。1 行目から左側時計回り、左側反時計回り、右側時計回り、右側反時計回りのターゲット信号である。尚、図では黄色に近いほど大きい値、紫に近いほど小さい値を示しており、黄色で示される部分が自己位置である。

図 5.12 に、単一リードアウトの ESN の出力を示す。単一リードアウトの ESN の場合、経路パターンに応じてリードアウトを切り替えないため、パターン毎にリードアウトの学習を行うと破滅的忘却が起こる。代わりに、ここでは 4 つのパターンを一

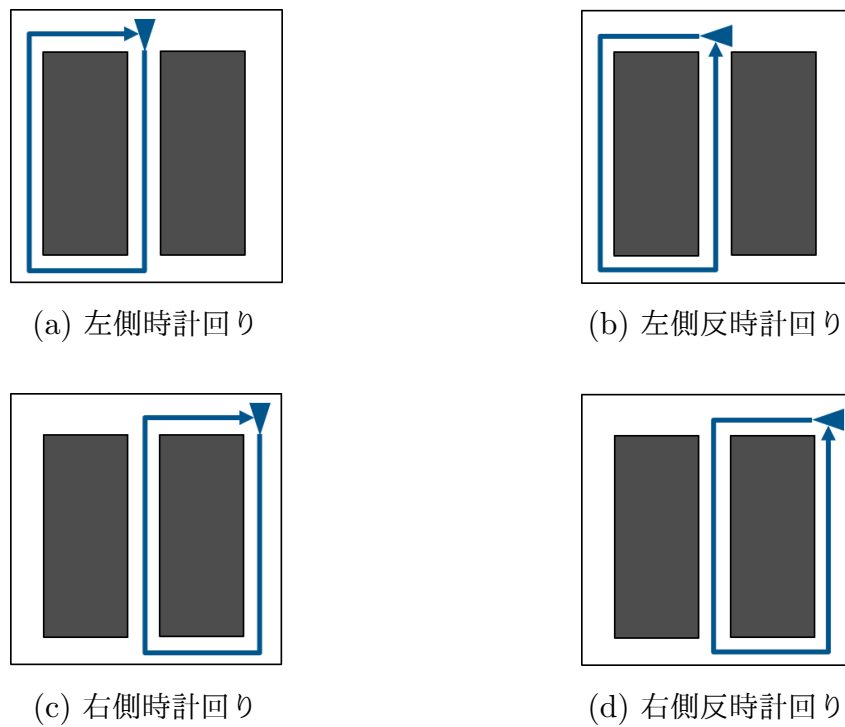


図 5.10: 8 の字迷路のタスク

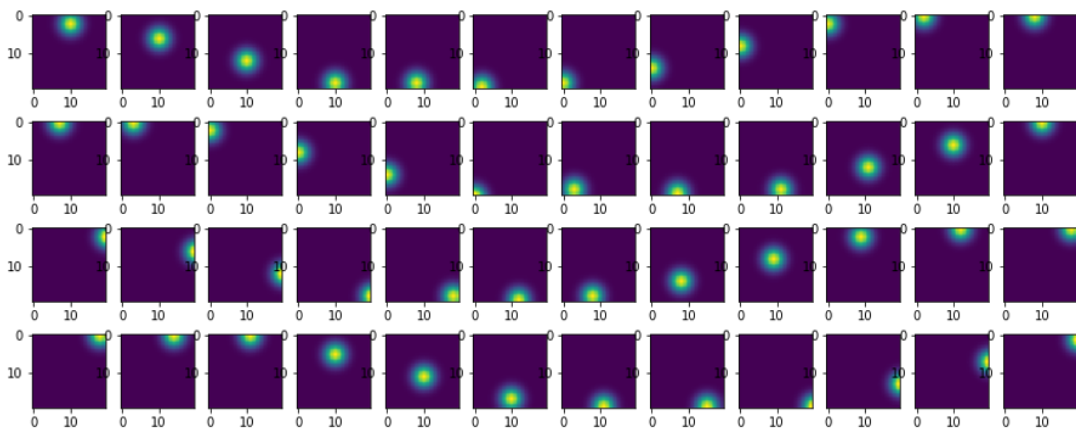


図 5.11: ターゲット信号

度に学習するバッチ学習を行った。図の1行目から3行目すなわちパターン (a) から (c) については、経路を想起できているが、4行目に示すパターン (d) については、途中から左側時計回りの経路が想起された。ここで、単一のリードアウトのESNでこのタスクを達成できなかった理由は以下のように考えられる。本実験で用いた8の字

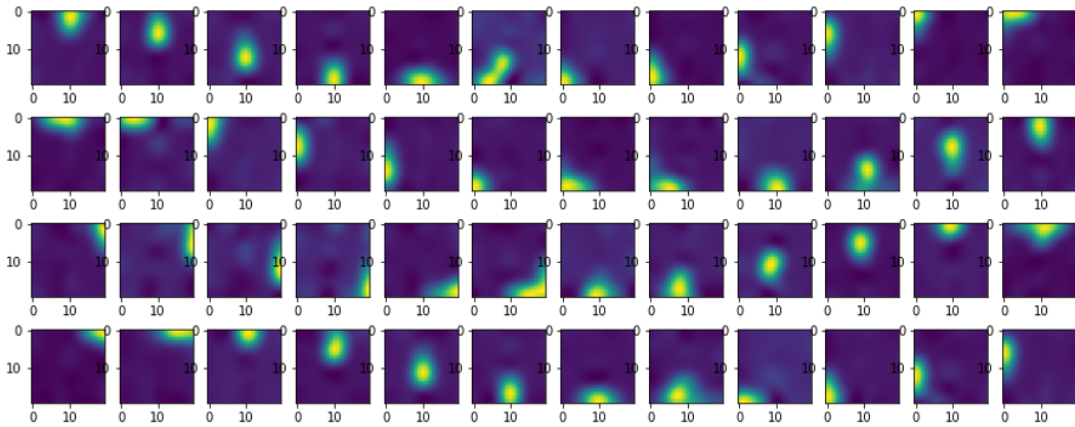


図 5.12: 単一リードアウトの ESN の出力

迷路において，図 5.13 の赤い点で示す三叉路に至るには，3 種類の経路がある．図の左から至る経路のパターン，及び右から至る経路のパターンはそれぞれ，左側を反時計回りする経路パターン (b) と，右側を時計回りする経路パターン (c) しかない．その一方，上から至る経路のパターンは，左側を時計回りする経路パターン (a) と，右側を反時計回りする経路パターン (d) の 2 つがある．したがって，ESN がどちらの経路から至ったのかを判定できなければ，三叉路に至った後どちらの方向へ進むべきか判定できない．今回の実験で用いた ESN は，この三叉路に至るまで，どちらの経路を通過してきたかという内部状態を保持できるほどの記憶容量がなく，三叉路に至った時点ではどちらの経路を通過してきたも同じ内部状態になっていたと考えられる．その状況において，4 つの経路パターンの予測誤差を最小化する場合，左側を時計回りする経路パターン (a) を正しく予測できた（三叉路に至った後の経路が長い）方が，全体的な予測誤差が小さくなるため，右側を反時計回りする経路パターン (d) の予測に失敗したと考えられる．したがって，単一のリードアウトを持つ ESN でこのタスクを解決する場合，ESN の記憶容量が上がるようにレザバーのノード数を十分設ける必要がある．但し，ESN のノード数を増やせば，計算コストが増える．

図 5.14 に，学習済みの提案モデルの各タスクの出力を示す．ここでは，経路パターンに応じて学習するリードアウトを切り替えるため，パターン毎にリードアウトの学習を行った．図に示すように，(a) から (d) の全てのパターンにおいて経路を正しく想起できた．

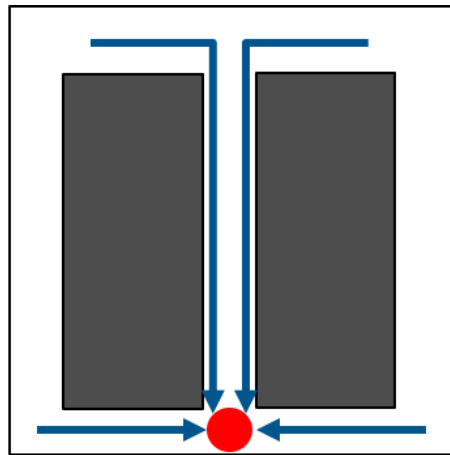


図 5.13: 単一のリードアウトの ESN でタスク達成が困難な理由

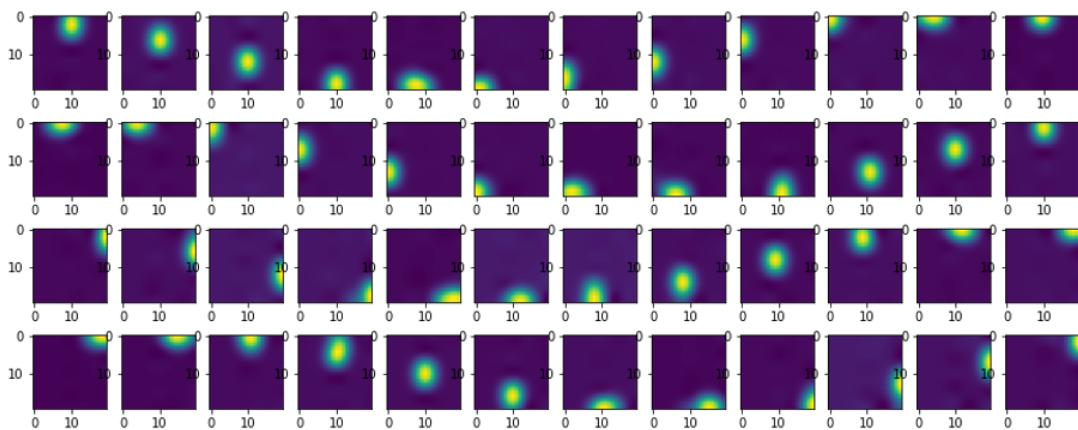


図 5.14: 提案モデルの出力

5.6 本章のまとめ

本章では、3, 4 章で提案した AI モデルを、マルチタスク学習に適用する手法を提案した。このため、本研究が提案する AI モデルの学習の中心となる、前頭前野モデル（レザバーコンピューティング）の拡張を行った。ここでは、複数のリードアウトをもち、タスクの類似度に合わせてリードアウトを切り替える、Reservoir Computing with Self-Organizing Multi-readout (RC-SOM) モデルを考案した。提案する RC-SOM モデルは、連続して与えられる複数のタスクを、破滅的忘却を回避

しながら学習することができた。さらに，RC-SOM モデルを前頭前野モデルとして，海馬モデルと統合し，自身が移動した複数の経路を想起するタスクを学習することができた。

第 6 章

おわりに

6.1 まとめ

本研究は、サービスロボットの实用化のため、ヒトのように考える AI モデルの実現を目指し、海馬・扁桃体・前頭前野の機能を統合した脳型人工知能モデルを提案した。このモデルを用いて、現行の AI では実現されていない学習パラダイム：個人の経験に基づいた記憶の獲得、および記憶に基づいた予測・行動生成を行った。ここでは、提案する AI モデルが数回の経験により、個人の経験に基づいた知識を獲得できたことを確認した。

さらに、単一のタスクだけでなく、複数のタスクが連続して与えられるマルチタスク学習に対応するため、Reservoir Computing with Self-Organizing Multi-readout (RC-SOM) モデルを考案し、上記 AI モデルに前頭前野モデルとして統合した。このモデルが、単一のリードアウトを持つレザバーコンピューティングが解けないマルチタスク学習を達成したことを確認した。

システムの高速度化・省電力化のため、提案する AI モデルの専用ハードウェアを設計した。本研究は、扁桃体モデルの専用ハードウェアを設計し FPGA に搭載した。ハードウェアの回路資源削減のために提案したハードウェア指向アルゴリズムは、6,000,000 回以上の乗算を削減した。XCZU9EG FPGA に実装された扁桃体モデルハードウェアは 150MHz で動作し、組込み CPU に搭載したソフトウェアの約 1,200 倍高速、組込み GPU に搭載したソフトウェアの約 90 倍高速となった。

表 6.1 に、ロボットアプリケーションとして、提案する AI モデルが達成したタスクをまとめる。表に示すように、提案する AI モデルによって、ロボットは個人の経

表 6.1: 提案する AI モデルが達成したタスク

タスク	概要
人の好みの学習	扁桃体の機能を模倣した古典的条件付けモデルが、数回の HRI を通じて客の好みを学習し、顔と注文場所から好みのオブジェクトを推定できる。
他者の移動経路の学習	海馬・扁桃体・前頭前野を統合したモデルが、人の移動経路を学習し移動経路を予測できる。
自分の移動経路と価値の学習	海馬・扁桃体・前頭前野を統合したモデルが、自分の移動経路とその価値をエピソード記憶として獲得し、価値に基づいた行動決定ができる。
自分の移動経路の想起	海馬・前頭前野 (RC-SOM) を統合したモデルが、自分の複数の移動経路をエピソード記憶として獲得し、移動経路の想起ができる。

験に基づいた記憶を獲得し、その記憶を活用してタスクを解決できるようになっている。本研究が発展することで、サービスロボットは現行の掃除ロボットのような単に家事をこなす機械から、人のパートナーとしての役割を担うようになることが期待される。

6.2 議論と今後の展望

6.2.1 個人の経験に基づいた知識の獲得

表 6.2 に提案する扁桃体モデルと、ロボット応用のための扁桃体モデルの既存手法との比較表を示す。EMA[39]と比較すると、提案モデルはより複雑なタスクに対応できる。EMA は CS として環境から単一の情報を取得するため、環境の複数の要素を考慮に入れることができない。さらに、US のトリガーとなる要素が未知である時、EMA を適用することはできない。一方で、提案モデルは環境からより多くの情報を CS として取得することができる。したがって、提案モデルは US のトリガーとなる要素が未知である時でも適用することができる。

SAFEL[41, 42] の扁桃体モデルと比較すると、提案モデルは LA・CE・感覚形の機能を含んでいるためロボットに応用しやすい。特に LA の導入により、提案モデルは未知の入力でも受け付けて推論することが可能である。例えば、若い男性の客が出現する状況と年老いた女性の客が出現する状況を提案モデルが学習したと仮定する。この時、若い女性の客が出現する状況が与えられると、若い男性の客に反応するニューロンと、年老いた女性の客に反応するニューロンの間のニューロンが反応するようになる。人の好みが年齢や性別に関係がある場合、提案モデルは未知の客の好みを推定することができる。

本来扁桃体は情動を司る脳部位で、恐怖条件づけに関わることが知られている。一方、本研究で提案した扁桃体モデルはその機能を拡張し、汎用的な古典的条件づけモデルとしている。古典的条件づけモデルは、恐怖条件づけを内包するため、提案するモデルは元来の扁桃体の機能としても活用可能である。但し、今回の実験で行ったような人の好みを学習するネットワークと、情動を司るネットワークは異なるインスタンスである必要がある。

Xilinx Vitis AI[83] を用いることで、Keras, TensorFlow, Caffe によって実装された DL モデルの変数を浮動小数から固定小数に変換して FPGA に実装することができるが、提案するアルゴリズムは扁桃体モデルを FPGA 実装により適している。提案手法により、乗算や指数関数はビットシフト演算に置き換えられ、劇的に回路使用率を減少することができる。Vitis AI では本モデルに特化した変換はサポートしていない。

表 6.2: ロボット応用のための扁桃体モデルとの比較

	モデルの構造	ハードウェア実装
EMA[39]	LA と CE を含む (単一の SOM と FCNN で構成)	FPGA
SAFEL[41, 42]	LA と CE を考慮していない (FCNN で構成)	ハードウェア実装なし
本研究	LA, CE, 感覚形を含む (複数の SOM と FCNN で構成)	FPGA

本研究では扁桃体モデルの変数として8ビットの固定小数を使用した³が、 8×8 ニューロンのSOMを用いたため、変数のビット幅としてはタスクの解決に十分であった。仮に、より多くのニューロンが必要となるタスクが与えられたとしても、ハードウェアはパラメータ化されているため、ビット幅の変更は用意である。

SDSoCは提案モデルの学習モードのレイテンシを15,395 CLKと推定した。したがって、ハードウェアのデータ転送を除いた理想の実行時間は、 $6.67 \text{ ns/CLK} \times 15,395 \text{ CLK} = 102.6 \mu\text{s}$ となる。これは、表3.5に示す提案ハードウェアの実行時間の実測値 $197.4 \mu\text{s} - 90.7 \mu\text{s} = 106.7 \mu\text{s}$ とほぼ一致する。

6.2.2 個人の経験に基づいた記憶の獲得及び記憶に基づいた予測

海馬の機能を模倣した関連研究のうちDNC[69]は、ニューラルネットワークに微分可能な外部メモリを追加することによってその性能を向上させるが、外部メモリを含めたネットワークの最適化はSGDによって行われるため膨大な学習回数を必要とする。したがって、本研究が着目する個人の経験に基づいた知識の獲得とは学習パラダイムが異なることになる。そこで本研究は川内ら[70]、下留[72]らが提案する、海馬・嗅内皮質モデルを参考にした場所細胞、時間細胞モデルを提案し、個人の経験に基づいた知識の獲得に活用した。

本論文で提案する海馬モデルは、場所と時間の内部表象を行うが、実際の海馬にはエピソードの統合や想起、リプレイなど、本研究で実装していない多くの機能がある[84]。今後は、これらの機能を追加し、より高度なAIモデルの実現を目指す。

本研究では、海馬・扁桃体・前頭前野を統合したモデルを提案し、その組み合わせに応じて、他者の移動経路を学習・予測するタスクと、自身の移動経路と価値を学習・予測し行動決定するタスクを行った。他者の移動経路を学習・予測するモデルは、数十回の学習で移動経路の予測ができるようになった。これはタスク学習中に更新するパラメータが、前頭前野モデル(レザバーコンピューティング)の出力層の結合荷重のみであったためである。DLベースの手法では、膨大な数のパラメータを最適化するために、大量の学習が必要となるため、数十回の学習では学習を完了することは困難と考えられる。

しかしながら、このモデルにおいては、扁桃体モデルの学習が独立しており、事前に学習を行う必要がある。今後は、End-to-endの学習を目指すため、前頭前野モデルから扁桃体モデルへのフィードバックの導入を検討している。尚、恐怖条件づけに

において、前頭前野から扁桃体への投射があることが解剖学的に知られている [85].

また、今回の実験では、検出したオブジェクトの情報として、One-Hot ベクトルで表現したラベル情報を用いた。今回は出現するオブジェクトの種類は3種類としたが、実際の問題ではこの数は大きくなることが予想される。その場合、現行の方法を用いると、海馬モデルが出力するイベントベクトルは巨大になり、メモリが不足することが予想される。これを解決するために、視覚野モデルを導入し、視覚野が抽出したオブジェクトの高次特徴量を活用することを検討している。具体的にはCNNにオブジェクトの画像を入力したときの、CNNの中間層の出力を取り出して用いる。CNNの中間層の出力のサイズは、オブジェクトの種類に依存せず一定であるため、イベントベクトルのサイズの爆発を防ぐことが可能と考えられる。

今回の実験では、単一のオブジェクトの経路を予測したが、実際の問題では、複数のオブジェクトが同時に出現し、それらの経路を同時に予測しなければならない状況も考えられる。これを対処する方法として、一つずつオブジェクトを注目し、それぞれ経路を予測する方法が考えられる。この場合、特定のオブジェクトを注視し続ける必要があるが、前頭前野モデルが出力した予測結果を視覚野、あるいは場所情報を取り扱う場所細胞にフィードバックすることで、実現可能である。または、場所細胞モデルを現状の1チャンネルではなく、検出したオブジェクトの数だけチャンネルを用意することで、複数の移動経路を同時に予測することができると考えられる。

自身の移動経路と価値を学習・予測するモデルでは、海馬モデルの場所細胞のみを活用しており、扁桃体モデルにその場所の価値を推定させた。提案する扁桃体モデルは、環境における複数の要素を考慮した価値判断をすることが可能である。例えば、海馬モデルの時間細胞が表象する文脈情報を考慮に入れることで、あるイベントが起こったあとはその場所の価値は高いが、他のイベントが起こったあとはその場所の価値は低いなど、状況に応じた価値を推定することができ、より複雑なタスクへの応用が期待できる。

ここで用いたナビゲーションのタスクは、強化学習 [9] を用いて解くことも可能と考えられる。強化学習の場合、トライアル・アンド・エラーを何度も繰り返すことによって、価値が最大となるような行動を探索する。一方、本研究の立場は、この強化学習による手法とわずかに異なる。ここでは、あるタスクを達成する過程でナビゲーションを行った結果、著しく価値が高い（もしくは低い）状況になり、それに対して扁桃体モデルがトリガーとなり、自身が経験したエピソードをたった1回、あるいは数回の繰り返して直接的に記憶することを想定している。すなわち、強化学習とは学

習のパラダイムが異なる。

また、今回はこのモデルをナビゲーションシステムに活用したが、このモデルはナビゲーションのみならず、予測した価値に基づいた行動決定を行うあらゆるタスクに応用可能である。例えば、特定の環境に置かれた物体を把持する際のロボットアームの軌道計画などに活用できると考えられる。

本研究は、提案する AI モデルのハードウェア化を目指すものだが、扁桃体モデル以外のハードウェア化は完了していない。今後は海馬モデル、前頭前野モデルのハードウェア化に取り組む。特に、前頭前野モデルのハードウェア化、すなわちレザバーコンピューティングのハードウェア化については、物理系のダイナミクスを演算に活用する、新しいレザバーコンピューティングデバイスに着目している。例えば、琴岡ら [86] は、セレン化銀ナノワイヤを用いたレザバーコンピューティングデバイスを提案している。このデバイスは、1V の電圧に対して、nA オーダーの電流で動作する。したがって、既存のデジタル方式（例えば本研究の扁桃体モデル回路の消費電力は約 5W）に比べて、超低消費電力である。これは電力の限られるサービスロボットへの応用を考えたとき有効である。

6.2.3 マルチタスク学習への適用

マルチタスク学習の関連研究 [80, 81] においては、本研究と同様にネットワークをモジュール化し、タスクによってモジュールを切り替える手法を取る。しかしながら、モジュールの選択、もしくはモジュール構造を形成するための最適化手法として遺伝的アルゴリズムが用いられる。遺伝的アルゴリズムは膨大な学習回数を必要とする。さらに、PathNet[80] は DNN ベースで SGD によって最適化されるもので、こちらも膨大な学習回数を必要とする要因となる。本研究は、レザバーコンピューティングのリッジ回帰による学習と SOM の近傍係数を組み合わせた学習法を提案し、少ない回数で学習を完了することを可能にした。これは、本研究が着目する個人の経験に基づいた知識の獲得及び記憶に基づく予測に有効である。

本論文では RC-SOM モデルがマルチタスク学習において破滅的忘却を回避することを確認した一方で、未知のタスクに対して、既知のタスクの情報を活用しながら効率よく学習する知識転移の効果については、その可能性は見出しているが、直接的な検証はしておらず、今後の課題となっている。知識転移が効果的なアプリケーションとしては強化学習のタスクが考えられる。例えば、あるビデオゲームのスコアを最大

化するという問題に直面したとき、過去に学習したゲームの中で類似する画面や操作方法を持つものから知識を転用することで、1 から学習するよりも効率よく学習ができると考えている。稲田ら [87] が、レザバーコンピューティングを用いた Q 学習を提案しており、RC-SOM モデルと組み合わせて強化学習を行うことが可能と考えられる。

参考文献

- [1] G.E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol.18, no.7, p.1527–1554, 2006.
- [2] “Imagenet large scale visual recognition challenge,” <http://www.image-net.org/challenges/LSVRC/> (2020年12月8日アクセス) .
- [3] A. Krizhevsky, I. Sutskever, and G.E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, p.1097–1105, 2012.
- [4] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *International Conference on Learning Representations 2016*, 2016.
- [5] J. Zhu, T. Park, P. Isola, and A.A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” *2017 IEEE International Conference on Computer Vision (ICCV)*, pp.2242–2251, 2017.
- [6] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine*, vol.29, no.6, pp.82–97, 2012.
- [7] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” *31st International Conference on Machine Learning*, pp.1764–1772, 2014.
- [8] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” *Proceedings of the 28th Interna-*

- tional Conference on Neural Information Processing Systems - Volume 1, p.577–585, 2015.
- [9] R.S. Sutton and A.G. Barto, Reinforcement Learning: An Introduction, A Bradford Book, 2018.
- [10] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” NIPS Deep Learning Workshop 2013, 2013.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.770–778, 2016.
- [12] “国立研究開発法人新エネルギー・産業技術総合開発機構, ロボットの将来市場予測を公表,” https://www.nedo.go.jp/news/press/AA5_0095A.html (2020年12月8日アクセス).
- [13] R. Caruana, “Multitask learning,” Machine Learning, vol.28, no.1, pp.41–75, 1997.
- [14] H. Nakahara, H. Yonekawa, T. Fujii, M. Shimoda, and S. Sato, “GUINNESS: A GUI based binarized deep neural network framework for software programmers,” IEICE Transactions on Information and Systems, vol.E102.D, no.5, pp.1003–1011, 2019.
- [15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol.86, no.11, pp.2278–2324, 1998.
- [16] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase, “Development of human support robot as the research platform of a domestic mobile manipulator,” ROBOMECH Journal, vol.6, 2019.
- [17] K. Benkrid, A. Akoglu, C. Ling, Y. Song, Y. Liu, and X. Tian, “High performance biological pairwise sequence alignment: FPGA versus GPU versus Cell BE versus GPP,” International Journal of Reconfigurable Computing, vol.2012, 2012.
- [18] P.A. Merolla, J.V. Arthur, R. Alvarez-Icaza, A.S. Cassidy, J. Sawada, F. Akopyan, B.L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S.K. Esser, R. Appuswamy, B. Taba, A. Amir, M.D. Flickner, W.P. Risk, R.

-
- Manohar, and D.S. Modha, “A million spiking-neuron integrated circuit with a scalable communication network and interface,” *Science*, vol.345, no.6197, pp.668–673, 2014.
- [19] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, P. Joshi, A. Lines, A. Wild, and H. Wang, “Loihi: A neuromorphic manycore processor with on-chip learning,” *IEEE Micro*, vol.38, no.1, pp.82–99, 2018.
- [20] “インテル, 1 億ニューロンが組み込まれたニューロモーフィック・リサーチ・システムを発表,” <https://newsroom.intel.co.jp/news/intel-scales-neuromorphic-research-system-100-million-neurons/> (2020 年 12 月 8 日アクセス).
- [21] E.M. Izhikevich, “Simple model of spiking neurons,” *IEEE Transactions on Neural Networks*, vol.14, no.6, pp.1569–1572, 2003.
- [22] L. Weiskrantz, “Behavioral changes associated with ablation of the amygdaloid complex in monkeys,” *Journal of Comparative and Physiological Psychology*, vol.49, no.4, pp.381–391, 1956.
- [23] J.E. LeDoux, *The emotional brain: The mysterious underpinnings of emotional life*, Simon and Schuster, 1996.
- [24] J.P. Aggleton, *The Amygdala: A Functional Analysis*, Oxford University Press, 2000.
- [25] E.T. Rolls, *Emotion Explained*, Oxford University Press, 2005.
- [26] E.A. Phelps and J.E. LeDoux, “Contributions of the amygdala to emotion processing: From animal models to human behavior,” *Neuron*, vol.48, no.2, pp.175–187, 2005.
- [27] I.P. Pavlov, *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex*, Oxford University Press, 1927.
- [28] E. Tsvetkov, W.A. Carlezon, F.M. Benes, E.R. Kandel, and V.Y. Bolshakov, “Fear conditioning occludes ltp-induced presynaptic enhancement of synaptic transmission in the cortical pathway to the lateral amygdala,” *Neuron*, vol.34, no.2, pp.289–300, 2002.
- [29] J.E. LeDoux, C. Farb, and D.A. Ruggiero, “Topographic organization of neurons in the acoustic thalamus that project to the amygdala,” *The Journal of Neuroscience*, vol.10, no.4, pp.1043–1054, 1990.

- [30] J. Armony, D. Servan-Schreiber, J. Cohen, and J. LeDoux, “An anatomically constrained neural network model of fear conditioning,” *Behavioral Neuroscience*, vol.109, no.2, pp.246–257, 1995.
- [31] J. Morén and C. Balkenius, “A computational model of emotional learning in the amygdala,” *From Animals to Animats 6: Proceedings of the 6th International Conference on the Simulation of Adaptive Behaviour*, 2000.
- [32] E.M. Izhikevich, “Solving the distal reward problem through linkage of stdp and dopamine signaling,” *Cerebral cortex*, vol.17, pp.2443–2452, 2007.
- [33] Y. Li, K. Nakae, S. Ishii, and H. Naoki, “Uncertainty-dependent extinction of fear memory in an amygdala-mpfc neural circuit model,” *PLOS Computational Biology*, vol.12, no.9, 2016.
- [34] E.T. Rolls, “A theory of emotion and consciousness, and its application to understanding the neural basis of emotion,” *The Cognitive Neurosciences*, p.1091–1106, 1995.
- [35] C.L. Hull, *Principles of Behavior: An Introduction to Behavior Theory*, New York: D. Appleton-Century, 1943.
- [36] D.W. Leonard, “Partial reinforcement effects in classical aversive conditioning in rabbits and human beings,” *Journal of Comparative and Physiological Psychology*, vol.88, no.2, pp.596–608, 1975.
- [37] R.A. Rescorla, “Partial reinforcement reduces the associative change produced by nonreinforcement,” *Journal of Experimental Psychology: Animal Behavior Processes*, vol.25, no.4, p.403–414, 1999.
- [38] N.J. Mackintosh, *The psychology of animal learning*, Academic Press, 1974.
- [39] S. Sonoh, K. Horio, S. Aou, and T. Yamakawa, “An emotional expression model inspired by the amygdala,” *International Journal of Innovative Computing, Information and Control*, vol.5, pp.1147–1160, 2009.
- [40] T. Kohonen, “Self-organized formation of topologically correct feature maps,” *Biological Cybernetics*, vol.43, no.1, pp.59–69, 1982.
- [41] C. Rizzi Raymundo and C.G. Johnson, “An artificial synaptic plasticity mechanism for classical conditioning with neural networks,” *Advances in Neural Networks – ISNN 2014*, vol.8866, pp.213–221, 2014.
- [42] C. Rizzi Raymundo, C.G. Johnson, and P.A. Vargas, “Fear learning for flex-

-
- ible decision making in robocup: A discussion,” RoboCup 2017, 2017.
- [43] M. Obayashi, S. Uto, T. Kuremoto, S. Mabu and K. Kobayashi, “An extended Q learning system with emotion state to make up an agent with individuality,” 2015 7th International Joint Conference on Computational Intelligence (IJCCI), pp.70–78, 2015.
- [44] Y. Tanaka and H. Tamukoh, “Hardware implementation of brain-inspired amygdala model,” 2019 IEEE International Symposium on Circuits and Systems (ISCAS), 2019.
- [45] Y. Tanaka and H. Tamukoh, “Live demonstration: Hardware implementation of brain-inspired amygdala model,” 2019 IEEE International Symposium on Circuits and Systems (ISCAS), 2019.
- [46] Y. Tanaka, T. Morie, and H. Tamukoh, “An amygdala-inspired classical conditioning model of an fpga for home service robots,” IEEE Access, vol.8, pp.212066–212078, 2020.
- [47] T. Shinozaki, “Biologically inspired feedforward supervised learning for deep self-organizing map networks,” NIPS Workshop on Representation Learning in Artificial and Biological Neural Networks (MLINI), 2016.
- [48] “Zynq UltraScale+ MPSoC,” <https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html> (2020 年 12 月 8 日アクセス) .
- [49] G. Shires and H. Wennborg, “Web Speech API Specification,” W3c community group final report, 2012.
- [50] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.6517–6525, 2017.
- [51] “SDSoC Development Environment,” <https://www.xilinx.com/products/design-tools/software-zone/sdsoc.html> (2020 年 12 月 8 日アクセス) .
- [52] “Vitis High-Level Synthesis,” <https://www.xilinx.com/products/design-tools/vivado/integration/esl-design.html> (2020 年 12 月 8 日アクセス) .
- [53] T.E. Oliphant, A guide to NumPy, vol.1, Trelgol Publishing USA, 2006.

- [54] R. Okuta, Y. Unno, D. Nishino, S. Hido, and C. Loomis, “Cupy: A numpy-compatible library for nvidia gpu calculations,” ML Systems Workshop in NIPS 2017, 2017.
- [55] N. Matsumoto, T. Sakaguchi, and Y. Ikegaya, “Hippocampal circuit computation: basic and advanced research topics,” *Japanese Psychological Review*, vol.56, no.2, pp.157–185, 2013.
- [56] J. O’Keefe and J. Dostrovsky, “The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat,” *Brain Research*, vol.34, no.1, pp.171–175, 1971.
- [57] Christopher J. MacDonald, Kyle Q. Lepage, Uri T. Eden, and H. Eichenbaum, “Hippocampal “time cells” bridge the gap in memory for discontinuous events,” *Neuron*, vol.71, no.4, pp.737–749, 2011.
- [58] G. Buzsáki, “Neural syntax: Cell assemblies, synapsembles, and readers,” *Neuron*, vol.68, pp.362–385, 2010.
- [59] K.D. Harris, “Neural signatures of cell assembly organization,” *Nature Reviews Neuroscience*, vol.6, pp.399–407, 2005.
- [60] H. Eichenbaum, “A cortical–hippocampal system for declarative memory,” *Nature Reviews Neuroscience*, vol.1, pp.41–50, 2000.
- [61] F. Haist, J.B. Gore, and H. Mao, “Consolidation of human memory over decades revealed by functional magnetic resonance imaging,” *Nature Neuroscience*, vol.4, pp.1139–1145, 2001.
- [62] T. Danjo, T. Toyozumi, and S. Fujisawa, “Spatial representations of self and other in the hippocampus,” *Science (New York, N.Y.)*, vol.359, pp.213–218, 2018.
- [63] D. Omer, S. Maimon, L. Las, and N. Ulanovsky, “Social place-cells in the bat hippocampus,” *Science (New York, N.Y.)*, vol.359, pp.218–224, 2018.
- [64] M. Fyhn, S. Molden, M.P. Witter, E.I. Moser, and M.-B. Moser, “Spatial representation in the entorhinal cortex,” *Science*, vol.305, no.5688, pp.1258–1264, 2004.
- [65] J.S. Taube, R.U. Muller, and J.J. Ranck, “Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis,” *The Journal of Neuroscience*, vol.10, no.2, pp.420–435, 1990.

-
- [66] J.S. Taube, R.U. Muller, and J.J. Ranck, “Head-direction cells recorded from the postsubiculum in freely moving rats. ii. effects of environmental manipulations,” *The Journal of Neuroscience*, vol.10, no.2, pp.436–447, 1990.
- [67] Benjamin J. Kraus, Robert J. Robinson, John A. White, H. Eichenbaum, and Michael E. Hasselmo, “Hippocampal “time cells” : Time versus path integration,” *Neuron*, vol.78, no.6, pp.1090–1101, 2013.
- [68] 坂本一寛, 虫明 元, “行動の計画と前頭前野ダイナミクス,” *日本神経回路学会誌*, vol.20, no.1, pp.37–41, 2013.
- [69] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S.G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, A.P. Badia, K.M. Hermann, Y. Zwols, G. Ostrovski, A. Cain, H. King, C. Summerfield, P. Blunsom, K. Kavukcuoglu, and D. Hassabis, “Hybrid computing using a neural network with dynamic external memory,” *Nature*, vol.538, pp.471–476, 2016.
- [70] M. Kawauchi, K. Takada, K. Tateno, and T. Morie, “A hippocampal spiking neural network model for path-dependent place cells,” 第 28 回日本神経回路学会全国大会 (JNNS2018) , pp.23–24, 2018.
- [71] L.M. Frank, E.N. Brown, and M. Wilson, “Trajectory encoding in the hippocampus and entorhinal cortex,” *Neuron*, vol.27, no.1, pp.169–178, 2000.
- [72] 下留 諒, 川内聖士, 高田健介, 立野勝巳, 田向 権, 森江 隆, “家庭用サービスロボットのための海馬-嗅内皮質の集積回路モデル,” *電子情報通信学会 NC 研究会*, 第 118 巻, pp.5–10, 2019.
- [73] T. Morie, J. Umezawa, and A. Iwata, “Gabor-type filtering using transient states of cellular neural networks,” *Intelligent Automation and Soft Computing*, vol.10, no.2, pp.95–104, 2004.
- [74] M. Lukoševičius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training,” *Computer Science Review*, vol.3, no.3, pp.127–149, 2009.
- [75] Y. Katori, “Network model for dynamics of perception with reservoir computing and predictive coding,” *Advances in Cognitive Neurodynamics (VI)*, pp.89–95, 2018.
- [76] K. Ara and Y. Katori, “Hierarchical network model of auditory information

- processing using dynamical predictive coding and non-negative matrix factorization,” 23rd International Conference on Artificial Life and Robotics, p.41–46, 2018.
- [77] H. Jaeger, “The” echo state” approach to analysing and training recurrent neural networks-with an erratum note’,” Bonn, Germany: German National Research Center for Information Technology GMD Technical Report, vol.148, 2001.
- [78] H.J. Kashyap, G. Detorakis, N. Dutt, J.L. Krichmar, and E. Neftci, “A recurrent neural network based model of predictive smooth pursuit eye movement in primates,” 2018 International Joint Conference on Neural Networks (IJCNN), pp.1–8, 2018.
- [79] D. Sussillo and L. Abbott, “Generating coherent patterns of activity from chaotic neural networks,” *Neuron*, vol.63, pp.544–57, 2009.
- [80] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A.A. Rusu, A. Pritzel, and D. Wierstra, “Pathnet: Evolution channels gradient descent in super neural networks,” arXiv:1701.08734, 2017.
- [81] 河合祐司, 小笹悠歩, 朴 志勲, 浅田 稔, “結合コスト最小化によるエコー状態ネットワークの破滅的忘却の回避,” 人工知能学会全国大会論文集, 2019.
- [82] I.B. Yildiz, H. Jaeger, and S.J. Kiebel, “Re-visiting the echo state property,” *Neural Networks*, vol.35, 2012.
- [83] “Vitis AI,” <https://www.xilinx.com/products/design-tools/vitis/vitis-ai.html> (2020年12月8日アクセス) .
- [84] H. Nakagawa, K. Tateno, K. Takada, and T. Morie, “A hippocampus model for episodic memory processing using associative memory,” 2020 International Symposium on Nonlinear Theory and Its Applications (NOLTA2020), p.136, 2020.
- [85] A.A. Moustafa, M.W. Gilbertson, S.P. Orr, M.M. Herzallah, R.J. Servatius, and C.E. Myers, “A model of amygdala–hippocampal–prefrontal interaction in fear conditioning and extinction in animals,” *Brain and Cognition*, vol.81, no.1, pp.29–43, 2013.
- [86] 琴岡 匠, S. Lilak, A.Z. Stieg, J.K. Gimzewski, 田中啓文, “セレン化銀ナノワイヤを用いたリザーバーコンピューティングデバイスの作製,” 第81回応用物

理学会秋季学術講演会, 2020.

- [87] M. Inada, Y. Tanaka, H. Tamukoh, K. Tatenno, T. Morie, and Y. Katori, “A reservoir based q-learning model for autonomous mobile robots,” 2020 International Symposium on Nonlinear Theory and Its Applications (NOLTA2020), pp.213–216, 2020.

謝辞

本研究を行うにあたり，研究者としての“いろは”を一からご教授いただき，世界大会出場や，有名国際学会の参加などの数多くの貴重な経験を与えていただいた，九州工業大学 大学院生命体工学研究科 人間知能システム工学専攻の田向 権 准教授に，心より感謝申し上げます。

九州工業大学 大学院生命体工学研究科 人間知能システム工学専攻の森江 隆 教授，田中 啓文 教授には，学部 4 年次の卒業研究から始まり，修士論文，そして現在に至るまでの 6 年間，数多くのご指導・アドバイスをいただき，さまざまな議論を交わしました。心より感謝申し上げます。はこだて未来大学 システム情報科学部 複雑系知能学科の香取 勇一 准教授，九州工業大学 大学院生命体工学研究科 人間知能システム工学専攻の立野 勝巳 准教授には，脳型人工知能モデルの研究において，さまざまな知見をご教授いただきました。心より感謝申し上げます。

最後に，公私に渡り良き相談相手として支えていただいた田向研究室の歴代メンバー，そして共に世界の強豪と戦った Hibikino-Musashi@Home の歴代メンバーに心より感謝申し上げます。

業績

Journal paper

- Yuichiro Tanaka, Takashi Morie, and Hakaru Tamukoh, “An amygdala-inspired classical conditioning model implemented on an FPGA for home service robots,” *IEEE Access*, Vol. 8, pp. 212066-212078, 2020.

International Conference

- Shoshi Tokuno, Yuichiro Tanaka, Takumi Kawasetsu, Koh Hosoda, and Hakaru Tamkouh, “Object Recognition by a Robot Hand Mounting Flexible Tactile Sensor,” 8th International Symposium on Applied Engineering and Science (SAES2020), C000275, 2020.
- Issei Uchino, Shoshi Tokuno, Yuichiro Tanaka, and Hakaru Tamkouh, “Analysis of Tactile Information Acquired by a Flexible Sensor,” 8th International Symposium on Applied Engineering and Science (SAES2020), C000302, 2020.
- Masafumi Inada, Yuichiro Tanaka, Hakaru Tamukoh, Katsumi Tateno, Takashi Morie, and Yuichi Katori, “A Reservoir Based Q-learning Model for Autonomous Mobile Robots,” 2020 International Symposium on Nonlinear Theory and Its Applications (NOLTA2020), pp. 213-216, 2020.
- Yuichiro Tanaka, Hakaru Tamukoh, Katsumi Tateno, Yuichi Katori, and Takashi Morie, “A Brain-inspired Artificial Intelligence Model of Hippocampus, Amygdala, and Prefrontal Cortex on Home Service Robots,” 2020 International Symposium on Nonlinear Theory and Its Applications

- (NOLTA2020), pp. 138-141, 2020.
- Shoshi Tokuno, Yuichiro Tanaka, Takumi Kawasetsu, Koh Hosoda, and Hakaru Tamukoh, “Object Recognition Using Flexible Tactile Sensor,” Asia Pacific Conference on Robot IoT System Development and Platform 2020 (APRIS2020), 2020
 - Masafumi Inada, Yuichiro Tanaka, Hakaru Tamukoh, Katsumi Tatenno, Takashi Morie, and Yuichi Katori, “Prediction of Sensory Information and Generation of Motor Commands for Autonomous Mobile Robots using Reservoir Computing,” 2019 International Symposium on Nonlinear Theory and Its Applications (NOLTA2019), pp. 333-336, 2019.
 - Yuichiro Tanaka and Hakaru Tamukoh, “Hardware implementation of brain-inspired amygdala model,” IEEE International Symposium on Circuit and Systems (ISCAS2019), Paper ID 2254, 2019.
 - Masatoshi Yamaguchi, Gouki Iwamoto, Yushi Abe, Yuichiro Tanaka, Yutaro Ishida, Hakaru Tamukoh and Takashi Morie, “Live Demonstration: A VLSI implementation of time-domain analog weighted-sum calculation model for intelligent processing on robots,” IEEE International Symposium on Circuit and Systems (ISCAS2019), Paper ID 2353, 2019.
 - Yuichiro Tanaka and Hakaru Tamukoh, “Live Demonstration: Hardware implementation of brain-inspired amygdala model,” IEEE International Symposium on Circuit and Systems (ISCAS2019), Paper ID 2351, 2019.
 - Yuichiro Tanaka and Hakaru Tamukoh, “Application of digital hardware of deep self-organizing map network,” Asia Pacific Conference on Robot IoT System Development and Platform 2018 (APRIS2018), 7, 2018.
 - Yuichiro Tanaka and Hakaru Tamukoh, “Hardware implementation of Deep Self-Organizing Map Networks,” Artificial Neural Networks and Machine Learning – ICANN2017, Vol. 10613, pp. 439-441, 2017.
 - Yutaro Ishida, Yuichiro Tanaka, Sansei Hori, Yuta Kiyama, Yuki Kuroda, Masataka Hisano, Hiroto Fujita, Yuma Yoshimoto, Yoshiya Aratani, Gouki Iwamoto, Kouhei Hashimoto, Dinda Pramanta, Yushi Abe, Takashi Morie and Hakaru Tamukoh, “Approach to accelerate the development of practical home service robots –RoboCup@Home DSPL-,” 26th International

Symposium on Robot and Human Interactive Communication, 2017.

- Yuichiro Tanaka and Hakaru Tamukoh, “Hardware-Oriented Algorithm for Deep Neural Networks Composed of Self-Organizing Maps,” Proceedings of International Workshop on Smart Info-Media Systems in Asia, pp. 116-121, 2016.

国内学会

- 徳野 将士, 田中 悠一郎, 川節 拓実, 細田 耕, 田向 権, “柔軟触覚センサを搭載したロボットハンドによる触覚情報からの物体認識,” 第 38 回日本ロボット学会学術講演会, 3E1-04, 2020.
- 田中 悠一郎, 田向 権, 立野 勝巳, 香取 勇一, 森江 隆, “海馬・扁桃体・前頭前野の機能を統合した脳型人工知能モデル,” 第 29 回日本神経回路学会全国大会 (JNNS2019), O3-41, pp. 28-29, 2019.
- 田中 悠一郎, 田向 権, “ヒューマンロボットインタラクションを通じて人の好みを学習する扁桃体モデルの実装,” 第 36 回日本ロボット学会学術講演会, 3D2-06, 2018.
- 田中 悠一郎, 鈴木 章央, 磯貝 浩久, 岩崎 正明, 田向 権, “スマートフォンのカメラで計測した脈波を用いた心拍変動パラメータの推定,” 電子情報通信学会スマートインフォメディアシステム研究会, pp. 35-38, 2018.
- 田中 悠一郎, 田向 権, “Deep Self-Organizing Map Networks の FPGA 実装,” 第 27 回日本神経回路学会全国大会, p.76, 2017.
- 田中 悠一郎, 田向 権, “自己組織化マップによって構成されるディープニューラルネットワークのハードウェア化,” 電子情報通信学会技術研究報告スマートインフォメディアシステム研究会, pp. 101-106, 2017.
- 石田 裕太郎, 田中 悠一郎, 森江 隆, 田向 権, “ホームロボット向け物体認識・把持システムの構築,” 第 34 回日本ロボット学会学術講演会, 3G1-03, 2016.
- 田中 悠一郎, 田向 権, “自己組織化マップを応用した Deep Neural Network のハードウェア指向アルゴリズム,” 電子情報通信学会 2016 年総合大会 ISS 特別企画「学生ポスターセッション」, ISS-SP-167, 2016.
- 田中 悠一郎, 田向 権, “hw/sw 複合体による高速画像認識システム,” 日本知

能情報ファジィ学会九州支部春季ワークショップ 2015, 講演番号 25, 2015.

投稿中

- Yuichiro Tanaka, Takashi Shinozaki, and Hakaru Tamukoh, “Hardware implementation of conformational self-organizing maps trained by advance propagation method,” PLOS ONE (Submitted).

その他

- Tomohiro Ono, Yuichiro Tanaka, Yutaro Ishida, Yushi Abe, Kazuki Kanamaru, Daichi Kamimura, Kentaro Nakamura, Yuta Nishimura, Shoshi Tokuno, Yuya Mii, Morio Yamauchi, Yuichiro Uemura, Takunori Hashimoto, Yugo Nakamura, Issei Uchino, Daiju Kanaoka, Takeru Hanyu, Kenta Tsukamoto, Takashi Morie, and Hakaru Tamukoh, “Hibikino-Musashi@Home 2020 Team Description Paper,” arXiv:2005.14451, 2020.
- Yuichiro Tanaka, Yutaro Ishida, Yushi Abe, Tomohiro Ono, Kohei Kabashima, Takuma Sakata, Masashi Fukuyado, Fuyuki Muto, Takumi Yoshii, Kazuki Kanamaru, Daichi Kamimura, Kentaro Nakamura, Yuta Nishimura, Takashi Morie, and Hakaru Tamukoh, “Hibikino-Musashi@Home 2019 Team Description Paper,” arXiv:2006.01233, 2019.
- Yutaro Ishida, Sansei Hori, Yuichiro Tanaka, Yuma Yoshimoto, Kouhei Hashimoto, Gouki Iwamoto, Yoshiya Aratani, Kenya Yamashita, Shinya Ishimoto, Kyosuke Hitaka, Fumiaki Yamaguchi, Ryuhei Miyoshi, Kentaro Honda, Yushi Abe, Yoshitaka Kato, Takashi Morie, and Hakaru Tamukoh, “Hibikino-Musashi@Home 2018 Team Description Paper,” 2018.
- Sansei Hori, Yutaro Ishida, Yuta Kiyama, Yuichiro Tanaka, Yuki Kuroda, Masataka Hisano, Yuto Imamura, Tomotaka Himaki, Yuma Yoshimoto, Yoshiya Aratani, Kouhei Hashimoto, Gouki Iwamoto, Takashi Morie, and Hakaru Tamukoh, “Hibikino-Musashi@Home 2017 Team Description Paper,” arXiv:1711.05457, 2017.

受賞

- ロボカップジャパンオープン 2020 @ホームシミュレーションリーグ 準優勝 (Technical Challenge)
- ロボカップジャパンオープン 2019 ながおか @ホーム・ドメスティックスタンダードプラットフォームリーグ優勝
- RoboCup2019 Sydney @Home League Domestic Standard Platform 3rd Place
- World Robot Summit 2018 Tokyo RSJ Special Award
- World Robot Summit 2018 Tokyo Service Robotics Category, Partner Robot Challenge / Real Space 1st Place, METI Minister's Award for Excellence in WRS
- RoboCup@Home P&G Challenge Winner
- RoboCup@Home League Procter & Gamble Dishwasher Challenge Award
- RoboCup2018 Montreal @Home League Domestic Standard Platform 1st Place
- RoboCup2017 Nagoya @Home League Domestic Standard Platform 1st Place
- RoboCup Japan Open 2017 @Home リーグ Domestic Standard Platform 2位・Open Platform 3位
- 第5回インテリジェントホームロボティクス研究会 iHR チャレンジ 総合優勝
- RoboCup JAPAN OPEN 2016 AICHI @Home Open Platform 準優勝
- NOLTA2020 Student Paper Award
- IEEE International Symposium on Circuit and Systems (ISCAS2019) Best Live Demonstration Award
- LSI デザインコンテスト 2019 準優勝 (日本シノプシス合同会社賞)
- Fighting-spirit award, APRIS Robot Challenge 2018
- 電子情報通信学会 九州支部 平成 27 年度 成績優秀賞 (学部)