パルスニューラルネットワークによる
時系列情報処理に関する研究

2004 年度

瀧田 航一朗

目 次

論文要旨

本論文では、工学的な応用を目的とした、パルスニューラルネットワークによる時系列情報処理に関する研究について述べる。パルスニューラルネットワークは従来の人工ニューラルネットワークに比べ、より生体の神経回路網に近いモデル化を行ったものである。そのため、パルスニューラルネットワークを用いる事で、従来型のニューラルネットワークでは扱えなかったような高度な知的情報処理の実現が期待されている。パルスニューラルネットワークの主な特長としては、ハードウェア実装の容易さ・生理学的知見の導入の容易さ・時系列情報の処理能力の高さなどがあるが、本論文は特に時系列情報処理に着目し、工学的な有用性の高いパルスニューラルネットワークモデルの確立を目指している。

本論文は、二本の柱から構成されている。第一の柱は、パルスニューラルネットワークを用いた強化学習則の研究である。強化学習は他の学習方式に比べ応用範囲の広い学習方式であり、生物の脳内においても一部に強化学習が用いられていることを示す知見が得られている。ここでは、ニューロン素子の動的な追加によるネットワークの拡張を特徴とする、二種類のパルスニューラルネットワークモデルを提案した。

第二の柱は、パルスニューラルネットワークへの新たな生理学的知見の導入と、その工学的応用の研究である。この観点からは、近年生理学の分野で研究が進んでいる、短期抑圧現象とよばれる生理現象をパルスニューラルネットワークに実装し、この特徴を動画像の注視制御に利用するモデルを提案した。

いずれのネットワークモデルにおいても、複数のコンピュータシミュレーションにより、その有効性を確認している。

第 1章

序論

本研究は、コンピュータによる高度な知的情報処理を目的とする、パルスニューラルネットワークによる時系列情報処理に関する研究である。本章では、本研究に関連する研究の歴史と背景について概説する。

1.1 人工ニューラルネットワーク研究の歴史

人工ニューラルネットワーク (artificial neural network) とは、生体の脳神経細胞組織をモデル化し、コンピュータによってその動作をシミュレートするものであると定義できる。その直接の起源は、1943 年に W.McCulloch と W.Pitts が発表した脳神経細胞の数理モデル [1] である。このモデルは、生体のニューロン (neuron) 素子の相互作用の動作を単純化したものであり、ニューロン素子が複数連結したネットワークによって脳機能の模倣を図るという、現在の人工ニューラルネットワーク研究の礎であると言える。このモデルは、ニューロンに与えられる入力を、単位時間にニューロン間で伝達される多数の電気パルスの積分として模式化したことから、一般に積分器型ニューロン (integrator-type neuron) と呼ばれる。これは、ニューロンの挙動は発火の平均量に基づいているという生理学上の学説、平均発火率コーディング理論 (Mean firing rate coding theory) に沿ったものであり、その起源は 1920 年代の E.D.Adrian の一連の研究に遡ることが出来る [2]〜[5]。

1949 年には、生体のニューロン間シナプス (synapse) 構造の可塑性に対する仮説が、D.O.Hebb によって提唱された [6]。その内容は、二つのニューロンが発火した場合、これらを結ぶシナプスの伝達効率が強化されるというものである。ニューロンの発火と学習、そして学習とシナプス伝達効率を結びつけて考えるこの仮説は、以降の研究に大きな影響を与えた。

積分器型ニューロン素子を用いた人工ニューラルネットワーク研究が加速するきっかけとなったのは、F.Rosenblatt が 1958 年に発表したパーセプトロン (perceptron [7], [8]) である。これは、シナプス伝達効率の変化によって学習が行われるという Hebb の仮説の流れを汲むものであり、学習が可能な識別器として多くの研究者の注目を集めた。個々のニューロンは、出力が連続値であるという違いがあるものの、積分器的に動作するという点で McCulloch-Pitts のモデルの流れを汲むものである。

しかし、この流れは、1969 年に M.Minsky らが自著 “Perceptrons” [9] の中でパーセプトロンの線形分離不可能性に言及したことで一転した。これによって、人工ニューラルネットワーク研究の波は大きく後退し、A.M.Turing, A.Newell，J.C.Shaw，H.Simon らの研究の流れを汲む人工知能 (AI) へと研究者の関心が移っていった [10], [11]。

しかしながら、パーセプトロンによってもたらされた人工ニューラルネットワークのブームが去ったこの時期においても、以降に影響を与える重要な研究がなされている。J.J.Hopfield らによる相互結合型ネットワーク [12]、T.Kohonen らによる自己組織化特徴マップ [13]、G.E.Hinton らによるボルツマンマシン [14], [15] などである。

人工ニューラルネットワークの研究は、1986 年に D.Rumelhart らによって誤差逆伝搬法 (back propagation algorithm [16]) が提案されると、再び脚光を浴びることとなっ

た。誤差逆伝搬法による学習は、パーセプトロンにおいて Minsky が指摘した線形分離不可能性を克服するものであり、これによって学習する識別器としての人工ニューラルネットワークが工学的に魅力あるものとなった。1987 年には T.J.Sejnowski らによって誤差逆伝搬法の実用例として NETtalk [17] が発表され、誤差逆伝搬法の名声を不動のものとした。

実際には、誤差逆伝搬法の考え方は 1967 年に既に甘利によって確率的降下法として発表されていた [18] が、最適解への収束が保証されない点などが問題視されたこともあり、当時は注目を集めなかった。Rumelhart の誤差逆伝搬法も最適解が保証されないことにおいては同様であるが、計算機能力の向上によって Sejnowski が行ったような実用的な応用が可能になったために、最適解への収束が保証されなくとも実用上は問題ではないとする考え方が受け入れられるようになったのである。

以後、Rumelhart のモデルは、積分器型ニューロンを用いた人工ニューラルネットワーク研究の中心的な存在となり、多くの応用研究が行われることとなる。また一方で、パルスニューロン素子という、積分器型ニューロンとは異なる立場でモデル化されたニューロン素子を用いた人工ニューラルネットワークも研究されていく。これが、次節で解説するパルスニューラルネットワークである。

1.2 パルスニューラルネットワークの諸研究

ニューロンの入力を単位時間あたりの電気パルスの総量として模式化した積分器型ニューロンモデルに対し、時系列的に入力される個々のパルスのそれぞれを入力として考えるのがパルスニューロン (pulsed neuron, spiking neuron) モデル [21], [22] である。現在研究されているパルスニューロン素子の多くは、1952 年に A.L.Hodgkin と A.F.Huxley が提案したモデル [20] の流れを汲むものである。これは、パルスの入力による神経細胞の内部電位の変化を生体のそれに近い形で模式化するものであり、一般的な積分器型のモデルよりも詳細なモデルであると言える。

パルスニューロンモデルの特長としては、ニューロン間で伝達される情報が 2 値であるために電気回路上での実装に適しているという点、内部電位の時間的な変化を模式化しているために時系列情報の処理に適しているという点、積分器型ニューロン素子では表現不可能な生理学的知見をモデルに導入することが可能である点、などが挙げられる。

にもかかわらず、工学的な利用を目的としたパルスニューロンと、それによって構築されるパルスニューラルネットワークの研究が加速するのは 1990 年代に入ってからであった。これは、以下のような要因による。第一に、コンピュータ上でのシミュレーションの困難さである。そもそも、積分器型のニューラルネットワークであっても、多数のニューロンの同時処理と繰り返し計算が必要となるために、そのソフトウェアシミュレーションの負荷は非常に大きく、多くの研究者を悩ませてきた。パルスニューロン素子の場合、内部電位の精密な計算を行うために、さらに莫大な計算機資源が必要とされる。これは、安価で高性能な計算機が普及する以前には、極めて深刻な問題であった。

第二の問題点は、パルス入力の時系列的な処理を行う必要があるパルスニューラルネットワークでは、工学的に有効かつ効率的な学習を行うことは容易ではないという点である。一方、積分器型ニューラルネットワークでは有望な学習則が次々と提案されていき、研究者の関心をこちらに集中させることとなった。

しかし、1990 年代に入ると、一連の生理学的研究 [23]〜[27] により、パルスの総量だけでなくそのタイミングもニューロンの挙動に大きな影響を与えているというテンポラルコーディング理論 (Temporal coding theory) の妥当性が証明されることとなった。これにより、人工ニューラルネットワークにおいても、個々のパルスのタイミングまで詳細に模式化したモデルでなければ表現できない現象があるのではないかと推測されることとなり、パルスニューラルネットワークへの関心は大きく高まった。また、積分器型ニューラルネットワークの処理能力の限界が示唆されるようになってきたのもこの時期であり、結果としてパルスニューラルネットワークの研究者の増大に

つながったのである。

1996年には、黒柳らが、生体の知覚現象をニューラルネットワークで模倣するという立場から、聴覚神経系を模倣したパルスニューラルネットワークモデル[28]を提案している。また、1997年には、海馬の記憶回路を模倣する塚田のモデル[29]が発表されている。

また、ハードウェア実装によってパルスニューラルネットワークを高速実行するという立場からは、1995年に発表された関根らのモデル[30]や、1998年の花形らの非同期パルスニューラルネットワークモデル[31]などが提案されている。特にFPGAへの実装を前提としたものでは、肥川らのモデル[32]などが知られている。

一方、パルスニューラルネットワーク内部において生じるカオス的現象を解析するという立場から、一般にカオスニューラルネットワーク(chaos neural network)と呼ばれるモデルの研究も行われている[33]〜[36]。

次節では、これらパルスニューラルネットワークにおいて研究されてきた学習則について解説する。

1.3 パルスニューラルネットワークの学習則

そもそも、人工ニューラルネットワークの学習則は三種類に分類できる。教師なし学習 (unsupervised learning)、教師あり学習 (supervised learning)、そして強化学習 (reinforcement learning) である。

教師なし学習則とは、その名の通り、外部からの教示なしに学習を行う手法である。前述の Hebb の学習則や、Kohonen の自己組織化特徴マップなどがこれに分類される。

生物の脳がどのようにして学習を行っているのかという疑問は、古くから人々の関心を集めてきた。世界的に積極的な研究がなされている一方で、様々な点において諸説入り乱れ、いまだに我々にとって最も大きな謎の一つである。このような現状において、Hebb の学習則は、数学的に簡潔であるだけでなく、脳神経科学的にも合理性があり、パルスニューラルネットワークの学習則の研究は、Hebb の学習則を中心に進められてきた。

Hebb の学習則は、Kohonen の自己組織化特徴マップなどと共に教師なし学習に分類され、工学的には、クラスタリング問題や、すでに問題が定式化されている組み合わせ最適化問題などを解くのに適している。一方、新しい入出力関係を学習するような類の問題、例えば、プラントの制御問題などに対しては適用が困難である。

パルスニューラルネットワークにおける教師なし学習を行うという研究は、前述の通り広く行われてきたが、特に工学的な有用性の高いモデルとしては、前節で述べた黒柳らの聴覚神経系モデル [28]、塚田の連想記憶モデル [29]、元木らの提案した改良型ヘブ学習則モデル [37] などが挙げられる。また、パルスニューロンを用いた自己組織化特徴マップとして、B.Ruf らのモデル [38] や雨森らのモデル [39]、C.Panchev らのモデル [40] などがある。

一方、教師あり学習とは、正しい出力が何であるかを外部から教えることにより学習を行う手法である。パーセプトロンや誤差逆伝搬法などがこれに該当し、複数の模範出力を補間・演繹することによって汎化能力を学習することができる。しかし、正しい出力を人間が用意してやる必要があるために、全く未知な環境では適用が難しく、既知の環境であっても、人間が想定していなかったような斬新な解法が得られる可能性が極めて低いという欠点がある。

パルスニューラルネットワークにおける教師あり学習の研究は、その実現の難しさに加え、生理学的合理性に疑いを持たれていたことから、教師なし学習と比べ立ち後れてきたと言わざるを得ない。このような中、R.C.O'Reilly が 1996 年に発表したモデル [41] と、B.Ruf らが 1997 年に発表したモデル [42] は、パルスニューラルネットワークにおいて実用的な教師あり学習が可能であるということを示し、研究者の注目を集めた。なお、O'Reilly のモデルが、誤差逆伝搬法をパルスニューラルネットワークに

適用したものと位置づけられる一方、B.Ruf らの時間パターン学習モデルは、Hebb の学習則を元にした形で教師あり学習を行うものである。

上記二種類の学習則に対し、強化学習では、得られた出力がどれだけ望ましかったかを外部から教える。Supervised Learning が日本では伝統的に教師あり学習と訳されているために混同されがちであるが、教師あり学習では Supervisor(指示者) が望ましい出力そのものを教え、強化学習では Critic(批評者) が出力の望ましさの度合だけを教えるという点に違いがある。この、望ましさの度合を示す信号は強化信号 (reinforcement signal) と呼ばれ、正のそれは特に報酬 (reward)、負のそれは罰 (penalty) と呼ばれる。強化学習は、教師あり学習に比べて多くの試行を必要とするという欠点があるものの、遥かに幅広い問題に対して適用が可能であることから、強化学習の研究は機械学習の中でも大きなテーマの一つとなっており [43]〜[46]、強化学習をニューラルネットワークに適用する研究 [47], [48] も進められている。

また、生理学的研究から、生体の脳においても強化学習的な学習がなされているという示唆 [49], [50] が多くなされており、パルスニューラルネットワークにおける強化学習は、生理学的合理性の点からも注目されている。工学的に有効なモデルとしては、D.Gorse らが 1997 年に発表したモデル [51] が挙げられる。これは、従来の強化学習においては難しいとされていた連続値関数の近似を、パルスニューロン素子を用いて行うというものである。

1.4 本研究の目的と位置付け

既に述べた通り、ハードウェア実装時の優位性・時系列情報の処理能力・生理学的知見導入の容易さなど、パルスニューラルネットワークには多くの長所があり、従来の人工ニューラルネットワークモデルでは扱うことのできなかった高度な知的情報処理の実現に大きな期待が寄せられている。特に工学的な観点からは、パルスニューロンに備わった時系列情報の処理能力をいかにして活用するかが重要となってきている。

このような背景のもと、本研究は、特にパルスニューラルネットワークの時系列情報処理能力に着目し、工学的有用性の高いネットワークモデルを確立することを目的としている。本論文は、二本の柱から構成される。第一の柱は、パルスニューラルネットワークを用いた強化学習則の研究である。パルスニューラルネットワークにおける学習則の研究そのものが量としては未だに少ないのが現状であるが、特に、工学的な利用を目的として、強化学習に基づいてパルスニューラルネットワークの学習を行うモデルは、非常に少なく、本研究は強化学習に基づくパルスニューラルネットワークの新しい流れを切り開くものである [86], [87]。第二の柱は、パルスニューラルネットワークへの新たな生理学的知見の導入と、その工学的応用の研究である。これは、近年生理学の分野で研究が進んでいる、短期抑圧現象とよばれる生理現象をパルスニューラルネットワークに実装し、この特徴を動画像の注視制御に利用するものである [88]。

図 1.1 に、本研究と、これまでのニューラルネットワーク研究との関係を示す。

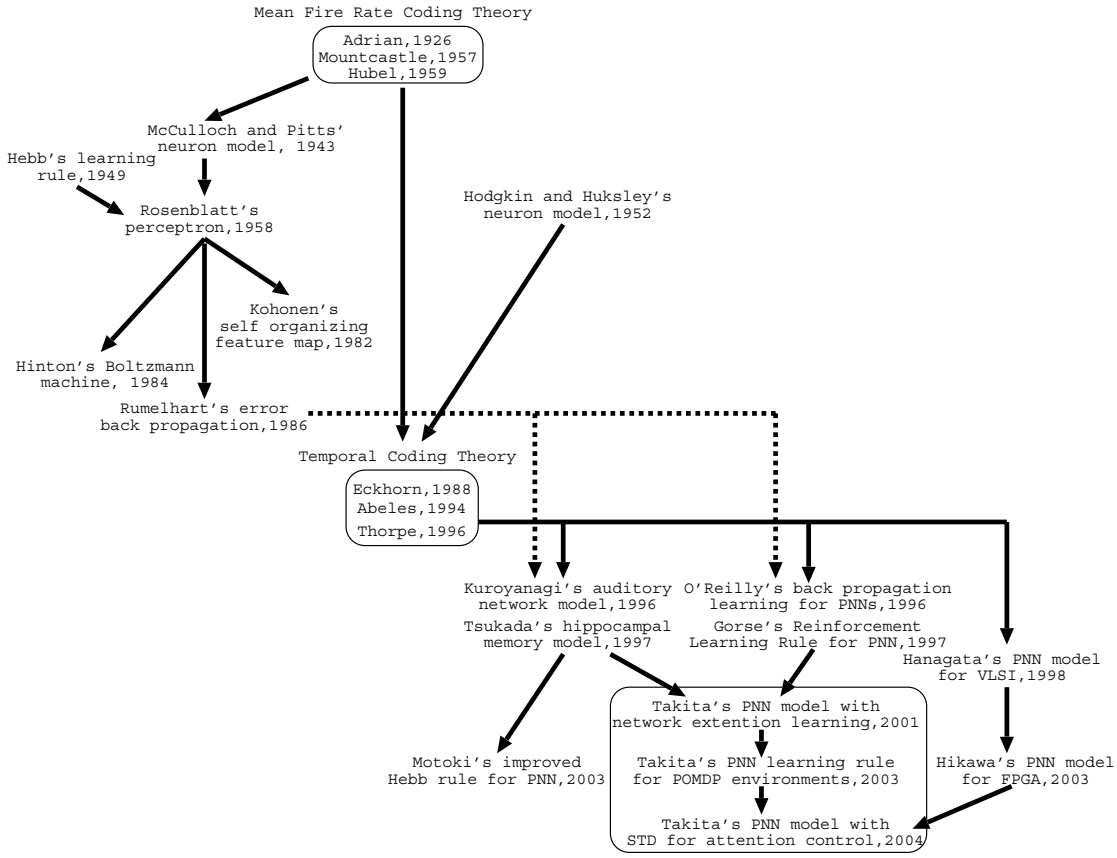


図 1.1 本研究と従来研究の関係

Fig. 1.1 The history of related researches.

1.5 本論文の構成

本論文は全 5 章から構成される。

第 2 章では、パルスニューラルネットワークにおいて、強化信号に基づいてネットワークの拡張と学習を行う研究について述べる。この研究では、パルスニューロン素子を用いることで、時系列情報を活用した上で強化学習が行えることを示す。

第 3 章では、第 2 章と同様に、強化学習とネットワークの拡張を取り入れたパルスニューラルネットワークにおいて、特徴の違うニューロンによって構成される複数の隠れ層を扱う研究について述べる。この研究では、複数の隠れ層の組み合わせにより、部分観測性が強い環境における学習精度が大きく向上することを示す。

第 4 章では、生体の神経細胞で見られる現象である、短期抑圧現象を導入したパルスニューロン素子を扱う研究について述べる。この研究では、短期抑圧現象の性質を応用することで、高度な注視制御を単純な構造のネットワークによって達成できることを示す。

第 5 章では、本論文のまとめを行う。

第 2章

パルスニューラルネットワークにおけるネットワーク拡張型強化学習則

本章では、生体の神経細胞を模したパルス駆動型ニューロンによる新しい階層型ネットワークと、そのための強化学習アルゴリズムを提案する。提案モデルでは、摂動的なパルスを各ニューロンに加えることで、偶発性を利用して時系列的な入出力空間の探索が行われる。学習は、外部から与えられる強化信号に基づいて行われる。結合荷重の修正に加え、入出力関係に対応した隠れ層ニューロンを動的に追加し、ネットワークの拡張を行いながら望ましい出力を学習していく。ネットワークは入力層、隠れ層、出力層の三層からなり、すべてのニューロンはパルス駆動型の素子である。計算機シミュレーションにより、提案するアルゴリズムの学習性能とその優れた特徴を分析している。

2.1 はじめに

生物の脳は高い情報処理能力を有するが、脳の中でどのような形で情報がコーディングされ、処理されているかは未だに明らかになっていない。しかし近年、細胞の平均発火率が情報を表現しているとする単一細胞仮説 [52] や Hebb アセンブリ仮説 [6] に対し、細胞の発火のタイミングも重要な意味をもつとするテンポラルコーディング [53], [54] が提唱されてきている。またこのような見地から、時空間的な依存性を持った細胞集団が情報コーディングの基本単位であるとするダイナミカルセルアセンブリ仮説 [55] に基づいた研究も進められている。また、生理学的な実験においても、テンポラルコーディングやダイナミカルセルアセンブリ仮説を支持するような結果が報告されている [56]。

人工ニューラルネットワークの分野においても、生体の神経細胞における平均発火率の概念を元にした積分器型のニューロン素子だけでなく、近年ではパルス（スパイク）に基づいた入出力をモデル化したパルスニューロン素子が考案されている [22], [35]。パルスニューロン素子は生体の神経細胞をより詳細にモデル化したものであると言え、その導入により生体の神経細胞に見られるような高次の情報コーディングへの道が開けると期待される。

パルスニューロン素子によって可能となる高次の情報処理のひとつに、時系列処理を挙げることができる。従来、TDNN [57] やリカレントニューラルネットワーク [58] などのように、ネットワークの構造を工夫することにより時系列処理を達成する手法が考案されてきた。しかし、生体の神経細胞は過去の入力履歴を局所膜電位の形で保持することが可能であり、細胞自身が基本的な時系列処理能力を持っていると推測されている [59]。このような観点から、人工ニューラルネットワークにおいてパルス駆動型ニューロン素子を用いることには、三つの大きな意義があると言える。第一に、時系列処理に関する新しい手法を開拓することができる。第二に、生体の神経組織をより精緻に模倣することで、より高次な処理能力が実現できると期待される。第三に、生理学的知見をより直接的に応用することが可能になる。

武田らは、パルス駆動型ニューロンの階層構造における学習則を提案し、時系列符号化を達成している [60]。また、塚田らは、海馬神経細胞における実験に基づき、高いパターン分離機能を持つ時空間学習則を提案している [29]。これらの手法は Hebb が提案した学習則 [6] を時間軸について拡張したものと位置づけられ、符号化問題については有効であるものの、その応用範囲は限定されている。雨森らの連想記憶モデル [39]、黒柳らの音源定位モデル [28] なども提案されているが、パルス駆動型ニューロンモデルにおいて汎用的に利用可能な学習則はいまだに確立されていないというのが現状である。

そもそもニューラルネットワークの学習は 3 種類に分類できる。Hebb 学習のような

教師無し学習、誤差逆伝播法 [16] のような教師あり学習、そして強化学習 [61], [62] である。教師無し学習は外部からの一切の教示無しに行われるため、一般に極めて限定された場合でなければ利用が難しい。教師あり学習では、外部から望ましい出力が提示されるため、学習の効率という点では申し分ない。しかしながら多くの問題においては、適切かつ充分な量の学習データを用意することが困難であり、強化学習こそが適切な手法となる。強化学習において必要とされる外部からの教示は、報奨と罰というスカラー量であり、これらは大抵の場合容易に設定できるからである。

以上のような観点から、本章ではパルス駆動型ニューロン素子を用いた新しいネットワーク構造と、そのための強化学習アルゴリズムを提案する。このモデルは偶発性を利用して入出力空間の探索を行い、強化信号に基づいた学習を実現するものである。また、時間的な相関を有すると推測される入出力に対しこれらを結ぶ隠れ層ニューロンを追加することによって、学習を達成する。

このモデルは、過去の入力をニューロンの内部状態として部分的に保持することにより、Barto らの Associative Search Network [47] を始めとする従来の強化学習則の多くと異なり、時系列的な入力を処理して望ましい出力を学習することができる。強化信号としては出力に対する時間遅れのあるものを扱い、直近の報奨を最大化するように学習を行う。

2.2 パルスニューラルネットワーク

ここでは、本研究で用いるパルスニューロンモデルとパルスニューラルネットワークの構造について説明する。

2.2.1 パルス駆動型ニューロン

提案モデルで用いたパルス駆動型ニューロン素子を図 2.1 に示す。このモデルでは、実際の神経細胞に見られる不応性や信号の時間的な加算などを考慮し、入出力としてパルス列を扱うことができる。このため、従来の積分器型のニューロンモデルに比べ、より実際の神経細胞に近いモデルになっている。また、過去の入力が内部状態として部分的に保持されるため、ニューロン素子単体で時系列入力を扱えるという特徴をもつ。提案モデルではこの点を活かし、帰還回路を用いることなく時系列処理を行っている。

このパルス駆動型ニューロンモデルでは、ある層のニューロン i に前階層のニューロン j からの入力パルスが到達すると、ニューロン i の内部電位 V_i は結合荷重 W_{ji} の分だけ上昇し、時間の経過とともに徐々に静止電位まで減衰していく。内部電位が閾値を越えると同時にニューロンは発火し、出力パルスが時間遅れののちに次階層に到達する。発火したニューロンの内部電位は静止電位にリセットされるとともに、不応性の影響を受け一時的にさらに電位が低下する。この不応性の影響も、時定数に則り徐々に減衰していく。またこのモデルでは、偶発的なパルス（ランダムパルス）の影響も受ける。これは個々のニューロンにおいてフラストレーション値と呼ばれるパラメータに依存して与えられるパルスであり、学習に利用される。なお、フラストレーション値については **2.3.2** で説明する。

ニューロン i の時刻 t における内部電位 $V_i(t)$ は、他のニューロンからの入力パルスによる影響 $P_i(t)$、不応性による影響 $R_i(t)$、フラストレーション値に依存したランダムパルスによる影響 $\lambda_i(t)$ によって、式 (2.1)〜(2.4) のように定義される。

$$V_i(t) = P_i(t) + R_i(t) + \lambda_i(t) \tag{2.1}$$

$$P_i(t) = \begin{cases} d_v \cdot P_i(t-1) + \sum_j W_{ji}(t-k_d) \cdot O_j(t-k_d), & O_i(t-1) = 0 \\ 0, & O_i(t-1) = 1 \end{cases} \tag{2.2}$$

$$R_i(t) = \begin{cases} d_r \cdot R_i(t-1) - k_r, & O_i(t-1) = 1 \\ d_r \cdot R_i(t-1), & O_i(t-1) = 0 \end{cases} \tag{2.3}$$

$$\lambda_i(t) = \begin{cases} d_v \cdot \lambda_i(t-1) + r(F_i(t)), & O_i(t-1) = 0 \\ 0, & O_i(t-1) = 1 \end{cases} \tag{2.4}$$

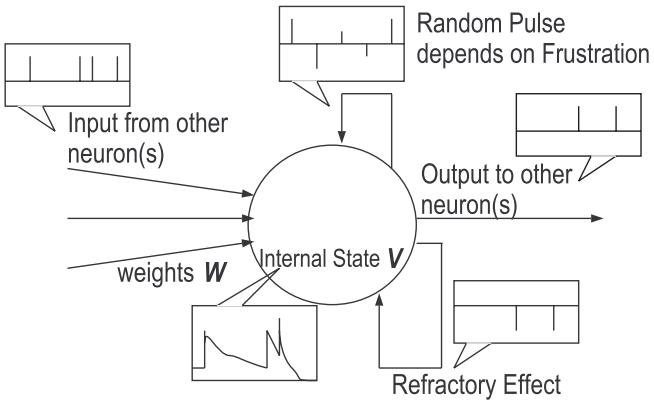


図 2.1 パルス駆動型ニューロン素子
Fig. 2.1 Pulsed neuron model.

ここで、d_v は内部電位の減衰率であり、k_d はパルス伝搬のディレイ、$W_{ji}(t-k_d)$ はニューロン j からニューロン i への時刻 $t-k_d$ における結合荷重、$O_j(t-k_d)$ はニューロン j の出力をそれぞれ示す。d_r は不応性の影響の減衰率を、k_r は一回の発火がニューロンに与える不応性の影響の大きさを示す。また、$r(F_i(t))$ は、$-F_i(t) \sim F_i(t)$ の範囲の一様乱数で、ランダムパルスの大きさを表す。なお、$F_i(t)$ はニューロンのフラストレーション値を示すもので、**2.3.2** で説明する。

式 (2.3) におけるパラメータ k_r および d_r の設定により、不応性の性質を大きく変えることができる。例えば k_r を高く d_r を低くした場合にはニューロンの発火直後の再発火が完全に抑止され、k_r を低く d_r を高くした場合には長期に渡って実質的に発火の閾値を上昇させることができる。また、不応性を適切に設定することにより発火の頻度に上限を設ける事ができ、特定の入力から極めて高頻度のパルスが与えられる場合などに、一つのニューロンの発火がネットワーク全体の挙動を支配してしまうような現象を防ぐ事が出来る。

ニューロン i の時刻 t における出力 $O_i(t)$ は、次式で定義される。

$$O_i(t) = \begin{cases} 1, & V_i(t) \geq \theta_v \\ 0, & V_i(t) < \theta_v \end{cases} \tag{2.5}$$

ここで、θ_v はニューロンの発火の閾値を表す。

2.2.2 ネットワーク構造

図 2.2 に、本研究で用いるパルスニューラルネットワークの構造を示す。提案モデルは入力層、隠れ層、出力層の三層からなる階層構造のネットワークで、各層は **2.2.1**

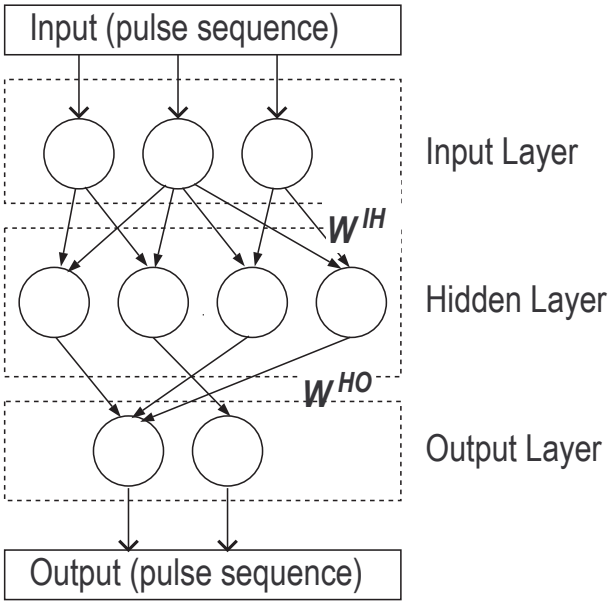


図 2.2 提案するパルスニューラルネットワークの構造
Fig. 2.2 The structure of proposed network.

で述べたパルス駆動型ニューロン素子によって構成されている。このネットワークにおいて、ニューロンは一つ上の層のいくつかのニューロンとのみ結合しており、層間の結合は全結合ではない。また、同じ層内のニューロン間の結合は存在しない。

2.3 パルスニューラルネットワークにおけるネットワーク拡張型強化学習アルゴリズム

ここでは、提案するパルスニューラルネットワークにおけるネットワーク拡張型強化学習アルゴリズムについて説明する。

2.3.1 概要

図 2.3 に、提案するネットワーク拡張型強化学習アルゴリズムの流れを示す。提案アルゴリズムは、(1) ネットワーク拡張処理、(2) 結合荷重修正処理、(3) 動作安定化処理、(4) 再不安定化処理の４つの処理から構成されている。ネットワークに対して外部から与えられる強化信号には正と負の２種類があり、正の信号を特に報奨信号と呼び、負の信号を罰信号と呼ぶこととする。

提案モデルは、**2.2.2** でも述べたように、入力層、隠れ層、出力層の三層によって構成される階層型のネットワークであるが、初期状態においては隠れ層ニューロンは存在せず、学習の進行に応じて追加されていく（図 2.4）。提案アルゴリズムでは、ネットワークの出力に対して報奨信号が与えられない場合には、各ニューロンのフラストレーション値が増大していく。各ニューロンはフラストレーション値に依存して生じるランダムパルスの影響を受けて、次第に不安定な出力を出すようになる。報奨信号が与えられた場合には、フラストレーション値が大幅に減少するとともに、ネットワーク拡張処理・結合荷重修正処理・動作安定化処理の三種類の処理のいずれかが適用され、学習が進められる。また、罰信号が与えられた場合には再不安定化処理が適用される。なお、学習の開始時には一切の隠れ層ニューロンが存在しないため、新しいニューロンが追加されるまでは出力層ニューロンはランダムパルスの影響のみを受けることになる。

生体の脳における学習では、シナプスの伸長によって新しい結合関係が生じ、使われていなかったニューロンが新しく使われるようになるという現象が、学習において大きな役割を果たしていると考えられている。提案モデルでは、単純な結合荷重の修正に加え、この現象がニューロンと結合の追加として導入されている。また、工学的な有用性を考えた場合にも、あらかじめ大きなネットワークを用意しておいて枝刈りを行っていく手法と違い、未知の環境や変化している環境への適用が容易である。

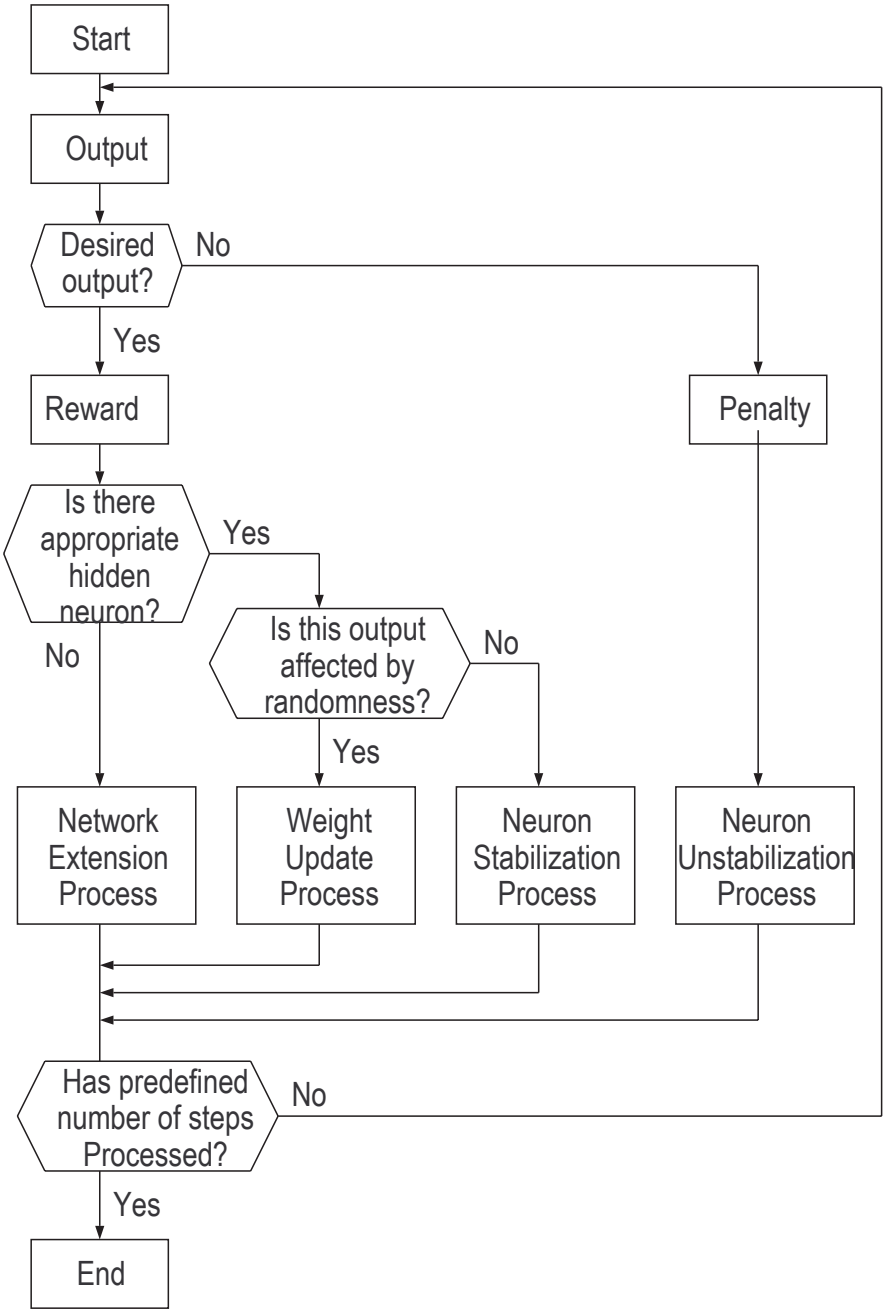
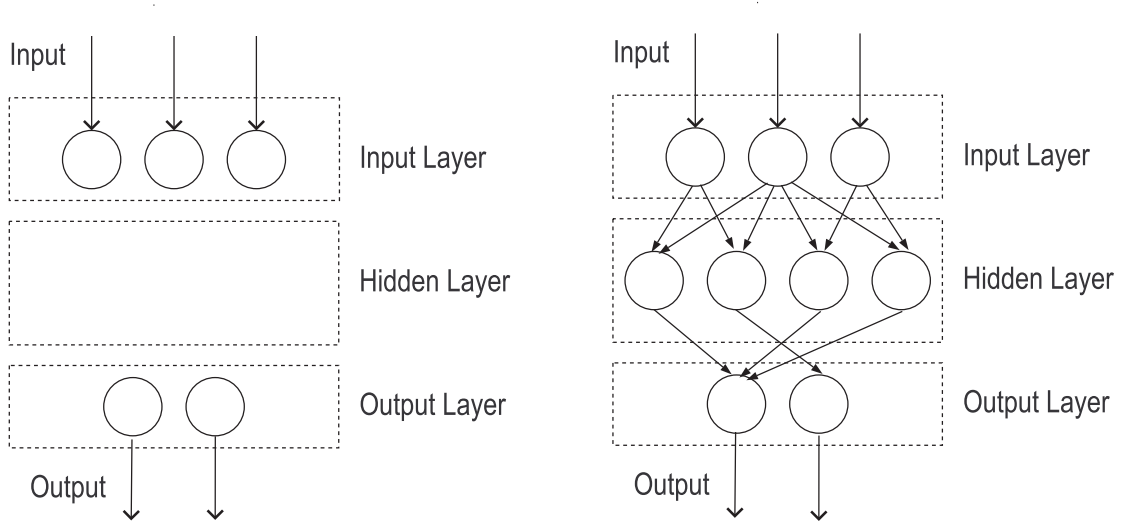


図 2.3 学習と動作の流れ

Fig. 2.3 The flow of the proposed model.



(a) 初期状態　　(b) 学習後状態の例

図 2.4 ネットワーク構造の変化
Fig. 2.4　An example of learning of network structure.

2.3.2　フラストレーション値

提案アルゴリズムでは、フラストレーション値に依存したランダムパルスによる偶発性を利用して学習を行う。

時刻 t におけるニューロン i のフラストレーション値 $F_i(t)$ を、次式のように定義する。

$$F_i(t) = \begin{cases} F_i(t-1) + f_i(t), & S(t) \leq 0 \text{ and } F_i(t-1) < \theta_f \\ 0, & F_i(t-1) \geq \theta_f \\ 0, & S(t) > 0 \text{ and } |R_i(t)| \geq |\theta_r| \\ D(t) \cdot F_i(t-1), & S(t) > 0 \text{ and } |R_i(t)| < |\theta_r| \end{cases} \tag{2.6}$$

ここで、$f_i(t)$ は時刻 t におけるフラストレーション値の増加量を表す。隠れ層ニューロンおよび出力層ニューロンについては $f_i(t)$ の初期値は正の微少量 k_f であり、入力層ニューロンについては $f_i(t)$ の初期値は 0 である。また $D(t)$ は、時刻 t における、フラストレーション値の解消を調整する変数であり、初期値を D_{init} とし、時間と共に増大していく。$D(t)$ の増大は学習の進行によって探索範囲が狭くなるのを防ぐ働きを持つ。また $S(t)$ は時刻 t における強化信号、θ_f はフラストレーション値に関する閾値、θ_r は不応性に関する閾値を示す。$R_i(t)$ は時刻 t におけるニューロン i の不応性の影響で、式 (2.3) で与えられる。

式 (2.6) から分かるように、報奨信号が与えられない場合には、フラストレーション

値は閾値 θ_f を越えない限り徐々に増大していく。また、報奨信号が与えられた場合には、フラストレーション値は大幅に減少する。この時、不応性の影響 $R_i(t)$ が閾値 θ_r を越えて残存していたならば、このニューロンの最近の発火が、報奨を得るに至った出力に寄与している蓋然性が高いとして、特にフラストレーション値を 0 にまで下げることとする。

2.3.3 ネットワーク拡張処理

ネットワーク拡張処理は、報奨信号が与えられた際に、その報奨信号と因果関係があると推定される入力層ニューロン全てと出力層ニューロンとを繋げるように、隠れ層ニューロンを追加する処理である。このような隠れ層ニューロンが既に存在している場合には、この処理は行われない。

強化信号とニューロンの因果関係

強化信号（報奨ないし罰信号）とニューロンとの間に因果関係があるかどうかを判別する基準として、提案アルゴリズムでは、ニューロンに残存する不応性 $R_i(t)$ に着目する。強化信号が与えられた際に不応性の影響が閾値を越えて残っている、つまり

$$|R_i(t)| \geq |\theta_r| \tag{2.7}$$

であるようなニューロンは最近発火したと考えられ、強化信号に何らかの関係があると推測される。

実行条件

ネットワーク拡張処理は、出力層ニューロン k について以下の条件が成り立つ時に実行される。

1. 時刻 t において報奨信号が与えられている。すなわち、

$$S(t) > 0 \tag{2.8}$$

が成り立つ。

2. 出力層ニューロン k に残っている不応性の影響 $R_k^O(t)$ が閾値 θ_r よりも大きい、すなわち、

$$|R_k^O(t)| \geq |\theta_r| \tag{2.9}$$

が成り立つ。この式が成り立つということは、時刻 t において与えられた報奨信号と、出力層ニューロン k との間に何らかの関係があると推測されることを意味する。

3. 入力層ニューロンのいずれかについて

$$|R_i^I(t)| \geq |\theta_r| \tag{2.10}$$

が成り立つ。すなわち、時刻 t において与えられた報奨信号と何らかの関係があると推測される入力層ニューロンが存在する。

4. 出力層ニューロン k と結合する隠れ層ニューロン j の中に、以下の二つの条件を同時に満たすものが存在していない。1) ニューロン j の不応性の残量について、式

$$|R_j^H(t)| \geq |\theta_r| \tag{2.11}$$

が成り立つ。2) ニューロン j と結合する全ての入力層ニューロンについて式 (2.10) が成り立つ。この二つの条件を同時に満たす隠れ層ニューロン j がもし存在する場合には、このニューロンは、これから作成しようとするニューロンと同じ働きをするものであるから、ネットワーク拡張処理を行う必要はない。

隠れ層ニューロンの追加

2.3.3 で述べた条件が全て満たされた場合には、報奨信号と因果関係があると推定される入力層ニューロンと出力層ニューロンとを繋げるような隠れ層ニューロンは存在しないと判断され、新たに隠れ層ニューロンが追加される（図 2.5。新たに追加する隠れ層ニューロンは、式 (2.10) の成り立つ入力層ニューロン全てと、式 (2.9) が成り立つ出力層ニューロンとの間に結合を持つ。

新たに追加する隠れ層ニューロンを m とすると、入力層ニューロン i から隠れ層ニューロン m への結合 W_{im}^{IH} は

$$W_{im}^{IH} = W_{\text{init}}^{IH} \tag{2.12}$$

と設定される。ここで、W_{init}^{IH} は正の微小量である。また、隠れ層ニューロン m から出力層ニューロン k への結合は

$$W_{mk}^{HO} = \theta_v \tag{2.13}$$

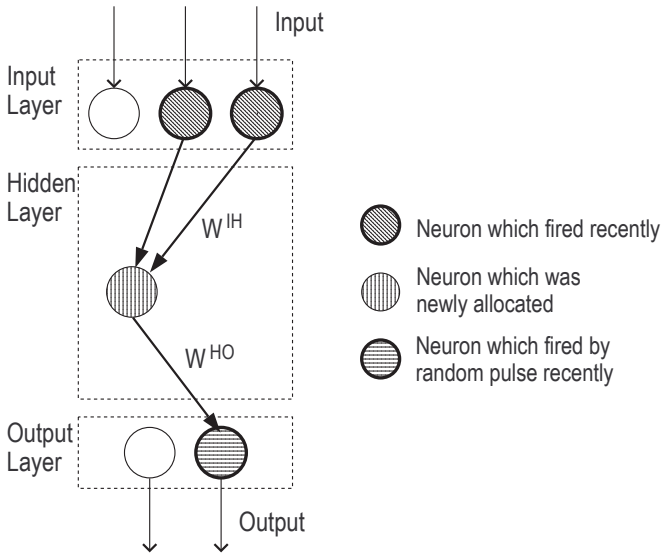


図 2.5 ネットワーク拡張処理
Fig. 2.5 Network extention process.

とする。ここで θ_v はニューロンの発火の閾値である。提案モデルにおいては隠れ層・出力層間の結合荷重は修正されず、隠れ層ニューロンが発火した際には不応性ないしランダムネスの影響がない限り出力層ニューロンが必ず発火するよう、結合荷重を発火閾値に設定している。

2.3.4 結合荷重修正処理

結合荷重修正処理は、報奨信号を利用して、結合荷重を強化する処理である。報奨信号が与えられる前に発火したニューロンは、報奨信号に対し何らかの寄与があると推測される。そのとき、もし発火がランダムパルスの影響によって起きたのであれば、今後も同じ状況で安定して発火するためには、そのニューロンに対する入力を増やす必要がある。そのために、同時期に発火していた他のニューロンからの結合荷重を増大させる。

実行条件

結合荷重修正処理は、隠れ層ニューロン j について以下の条件が成り立つ時に実行される。

1. 時刻 t において報奨信号が与えられている。すなわち、式 (2.8) が成り立つ。

2. 隠れ層ニューロン j に残っている不応性の影響 $R_j^H(t)$ が閾値 θ_r よりも大きい。すなわち、式 (2.11) が成り立つ。この式が成り立つということは、時刻 t において与えられた報奨信号と、隠れ層ニューロン j との間に、何らかの関係があると推測されることを意味する。

3. 隠れ層ニューロン j と結合した出力層ニューロンを k としたとき、式 (2.9) が成り立つ。すなわち、この隠れ層ニューロンの発火の影響を受けて出力層ニューロン k が発火している。

4. 隠れ層ニューロン j について

$$B_j(t) = 1 \tag{2.14}$$

が成り立つ。ここで、$B_j(t)$ は、ニューロン j の発火原因を表す変数であり、

$$B_j(t) = \begin{cases} 1, & V_j(t-1) \geq \theta_v \text{ and } V_j(t-1) - \lambda_j(t-1) < \theta_v \\ 0, & V_j(t-1) \geq \theta_v \text{ and } V_j(t-1) - \lambda_j(t-1) \geq \theta_v \\ B_j(t-1), & \text{otherwise} \end{cases} \tag{2.15}$$

で与えられる。式 (2.15) を見ても分かるように、ニューロン j が最後に発火したのがランダムパルスの影響によるものであれば $B_j(t) = 1$ となり、そうでなければ $B_j(t) = 0$ となる。ランダムパルスの影響によらずに隠れ層ニューロン j が発火したのであれば結合荷重を修正する必要はないため、式 (2.14) によってこの処理の必要性を判断している。

結合荷重の修正

2.3.4 で述べた条件が全て満たされた場合には、結合荷重の修正が行われる。隠れ層ニューロン j と結合した入力層ニューロンを i からの結合荷重 W_{ij} は以下のように更新される。

$$W_{ij}(t) = \begin{cases} W_{ij}(t-1) + k_l \cdot S(t) \cdot (W_{\max} - W_{ij}(t-1))^2, & |R_j(t)| \geq |\theta_r| \\ W_{ij}(t-1), & |R_j(t)| < |\theta_r| \end{cases} \tag{2.16}$$

ここで、k_l は学習係数を、$W_{\max}$ は結合荷重の上限値を示す。

報奨信号が与えられた時刻を基準に考えて、その直前に発火していた出力層ニューロンは報奨信号に寄与していると推測できる。それらの出力層ニューロンに結合して

いる隠れ層ニューロンのうち、ランダムパルスの影響によって発火していたニューロンは、今後類似の入力列が与えられたときにはランダムパルスによらず発火することが望ましい。そこで、それらの隠れ層ニューロンに結合している入力層ニューロンのうち、直前に発火していたものとの結合荷重を式 (2.16) のように増してやるわけである。

2.3.5 動作安定化処理

動作安定化処理は、報奨信号を利用して、結合荷重が不要に増加することを防ぐための処理である。報奨信号が与えられた際に、すでに適切な結合荷重に達していると推定されるニューロン、および結合荷重を修正しても報奨信号に寄与しないと推定されるニューロンに対して適用され、以後のフラストレーション値の上昇を抑える。これにより、結合荷重と出力の両方を安定させる。すでに適切な結合荷重に達しているニューロンとは、報奨信号の到達時に不応性の影響が残っているニューロンの中で、ランダムパルスの影響によらず発火したニューロンのことである。結合荷重を修正しても報奨信号に寄与しないニューロンとは、報奨信号が到達する前に充分な数のニューロンから入力パルスを受け取っているにもかかわらず、発火していないニューロンである。

実行条件

動作安定化処理は、隠れ層ニューロン j について以下の条件が成り立つ時に実行される。

1. 時刻 t において報奨信号が与えられている。すなわち、式 (2.8) が成り立つ。

2. 隠れ層ニューロン j に残っている不応性の影響 $R_j^H(t)$ が閾値 θ_r よりも大きい、すなわち、式 (2.11) が成り立つ。この式が成り立つということは、時刻 t において与えられた報奨信号と、隠れ層ニューロン j との間に何らかの関係があると推測されることを意味する。

3. 隠れ層ニューロン j が最後に発火したのはランダムパルスの影響によらない、すなわち

$$B_j(t) = 0 \tag{2.17}$$

が成り立つ。

適切な結合荷重に達していると推定されるニューロンの動作安定化

2.3.5 で述べた条件を全て満たすような隠れ層ニューロン j が存在する場合には、そのニューロンは適切な結合荷重に達していると推定され、フラストレーション値の増加量 $f_j(t)$ を次式に従って更新する。

$$f_j(t) = k^-_{f_1} \cdot f_j(t-1) \tag{2.18}$$

ここで、$k^-_{f_1}(0 < k^-_{f_1} < 1)$ はフラストレーション値の増加量の減衰率を示す。

また、ある隠れ層ニューロン j についてこの処理を適用した場合には、その隠れ層ニューロンに結合している出力層ニューロンに対しても、式 (2.18) を適用する。

これにより、適切な結合荷重に達していると推定されるニューロンのフラストレーション値は増加しにくくなり、それに伴ってランダムパルスの影響も弱くなる。結果として、出力と結合荷重の両方が安定化することとなる。

報奨に寄与しないと推定されるニューロンの動作安定化

2.3.5 で述べた条件を全て満たすような隠れ層ニューロン j が存在し、かつ $j \neq m$ なる隠れ層ニューロン m について以下の条件が全て成り立つならば、隠れ層ニューロン m は以降どのように結合荷重を修正していっても報奨に寄与しないと推定される。

1. 隠れ層ニューロン m に残っている不応性の影響 $R^H_m(t)$ が閾値 θ_r よりも小さい。

   $$|R^H_m(t)| < |\theta_r| \tag{2.19}$$

   すなわち、隠れ層ニューロン m は時刻 t においては報奨信号に寄与していない。
2. 隠れ層ニューロン m と結合している全ての入力層ニューロン i について、

   $$|R^I_i(t)| \geq |\theta_r| \tag{2.20}$$

   が成り立つ。結合している全ての入力層ニューロンが発火しているということは、隠れ層ニューロン m はまさにこの入力パルス列に対して発火することが期待されていたということである。しかしながら、実際にはこの入力パルス列に対しては別の隠れ層ニューロン j が発火しており、それによって報奨信号が得られたわけであるから、隠れ層ニューロン m は今後も発火する必要がないと推定されるわけである。

この場合には、隠れ層ニューロン m のフラストレーション値の増加量 $f_m(t)$ を次式に従って更新する。

$$f_m(t) = k^-_{f_2} \cdot f_m(t-1) \tag{2.21}$$

ここで、$k_{f_2}^-(0 < k_{f_2}^- < 1)$ はフラストレーション値の増加量の減衰率を示す。

これにより、報奨に寄与しないと推定されるニューロンのフラストレーション値は増加しにくくなる。

フラストレーションの解消を調整するパラメータ $D(t)$ の更新

θ_s 回の報奨信号に対して連続して動作安定化処理が実行された場合、すなわちいずれの場合にも **2.3.5** で述べた条件を全て満たすような隠れ層ニューロン j が存在した場合には、フラストレーションの解消を調整するパラメータ $D(t)$ を次式に従って更新する。

$$D(t) = D(t-1) + k_d \tag{2.22}$$

ここで、k_d は正の定数値であり、$D(t)$ の増加量を示す。

報奨信号が安定して与えられるようになると、式 (2.6) に従い、各ニューロンのフラストレーション値は低く抑えられる。これによってランダムパルスの影響は微弱なものとなるが、このとき学習の完了していない隠れ層ニューロンが残っていると、それ以降の学習が難しくなる。そこで、安定した報奨信号が連続して与えられている場合には $D(t)$ を増加させ、学習が完了していない隠れ層ニューロンの学習を促す。

2.3.6 再不安定化処理

再不安定化処理は、罰信号を利用して、まだ完全には学習が済んでいないニューロンに学習を促す処理である。これは動作安定化処理と対になるものであり、一度動作が安定化したニューロンに不安定な動作をさせるようにする。大部分の状況には正しく反応するが、それ以外のいくつかの状況に対応できていないニューロンの動作が安定化してしまった場合、それ以上の学習が行われなくなってしまう。そこで、ニューロンが罰信号に寄与している状況を検出し、そのようなニューロンの動作を不安定化させるのである。なお、罰信号に寄与しているニューロンとは、罰信号が与えられる前に発火していた隠れ層ニューロンである。

実行条件

再不安定化処理は、隠れ層ニューロン j について以下の条件が成り立つ時に実行される。

1. 時刻 t において罰信号が与えられている。すなわち、

$$S(t) < 0 \tag{2.23}$$

が成り立つ。

2. 隠れ層ニューロン j に残っている不応性の影響 $R_j^H(t)$ が閾値 θ_r よりも大きい。すなわち、式 (2.11) が成り立つ。この式が成り立つということは、時刻 t において与えられた罰信号と、隠れ層ニューロン j との間に何らかの関係があると推測されることを意味する。

3. 隠れ層ニューロン j が最後に発火したのはランダムパルスの影響によらない。すなわち式 (2.17) が成り立つ。

再不安定化

2.3.6 で述べた条件が全て満たされた場合には、隠れ層ニューロン j の学習はまだ完了していないものと推定され、フラストレーション値の増加量 $f_j(t)$ を次式に従って更新する。

$$f_j(t) = k_f^+ \cdot f_j(t-1) + (1 - k_f^+) \cdot k_f \tag{2.24}$$

ここで k_f は $f_j(t)$ の初期値であり、k_f^+ は $f_j(t)$ の増加を制御する定数値である。フラストレーション値の増加量 $f_j(t)$ が低下していた場合には、この処理によって $f_j(t)$ が初期値に近づく。それゆえ、まだ完全には学習が完了していなかった隠れ層ニューロン j は、再びランダムパルスの影響を受けて学習することができるようになる。

2.4 計算機実験

提案モデルの動作を確認し有効性を示すためにテニスゲームとシューティングゲームの２つの例題に関して計算機実験を行った。

2.4.1 実験環境１（テニスゲーム）

概要

この実験では、提案モデルを使って、ごく簡単なテニスゲームを実行した。これは、図 2.6 に示すように、横 4 マス縦 6 マスの領域のなかで、ラケットを左右に移動させながらボールを打ち返すゲームで、提案モデルを用いてラケットを操作し、ボールを落とさないようにラリーを続けるのが目的である。

ボールは、側面の壁かラケットに当たると跳ね返る。領域上部には相手のプレーヤーがいるものと考え、ここにボールが到達しても必ず跳ね返るものとする。もしボールが領域から下へ出てしまった場合には、ランダムな時間の後に画面上部のランダムな位置から、$(-1,1)$ ないし $(1,1)$ の運動ベクトルを持った新たなボールが投じられる。ただし、これではラケットを僅かに動かすだけで安定したラリーが保たれてしまうので、ボールが領域上部で跳ね返る場合にはその x 座標をランダムに変更するものとした。

提案モデルへの入出力は、次のようにした。4×6 の領域のうち、最下段を除く 20 のマスに対応して 20 の入力層ニューロンを用意し、一定時間ごとに、ボールが存在する位置に対応したニューロンに、発火閾値と等しい大きさの入力パルスを与える。この時間幅は、ネットワークの単位時間 (以下、ステップと呼ぶ) にして 12 ステップであり、これを 1 サイクルと呼ぶ。なお、ボールの移動も 1 サイクルごとに（斜め方向に）1 マスである。

また、出力層ニューロンは、4 マスの横幅に対応させて 4 つ用意した。これらはラケットの目標位置を示すもので、いずれかが発火すると、ラケットは対応する位置に移動する。ただし、1 サイクルに移動できるのは 1 マスのみである。移動している途中で別の出力層ニューロンが発火すると、新しい目標位置に向かって移動し始める。1 サイクルの間に複数の出力層ニューロンが発火した場合には、移動しないものとする。

ネットワークに対する報奨信号は、ラケットがボールを打ち返した瞬間に与えられるものとし、その値は 1.0 とした。また、罰信号はボールを打ち返し損なった場合に与えられるものとし、その値は -1.0 とした。

具体的なパラメータは、表 2.1 の通りである。

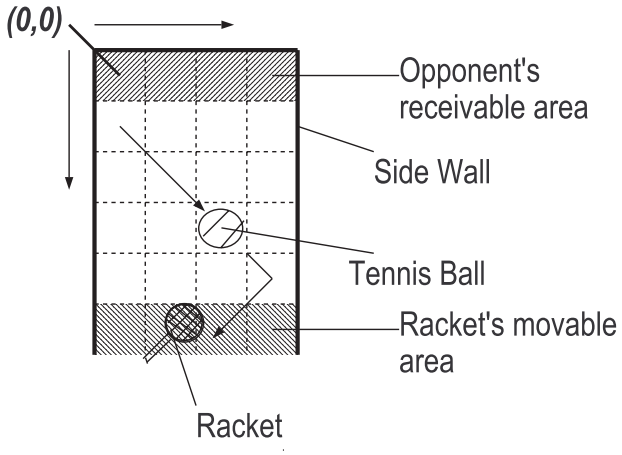


図 2.6 テニスゲーム環境
Fig. 2.6 Tennis game environment.

表 2.1 パラメータ設定
Table 2.1 Simulation parameters on tennis game environment.

発火閾値	θ_v	1.0
フラストレーション値上限	θ_f	0.2
不応性閾値	θ_r	-0.01
連続報奨回数閾値	θ_s	100
パルス減衰率	d_v	0.94
不応性減衰率	d_r	0.94
不応性強度	k_r	1.0
ニューロン間ディレイ	k_d	3
フラストレーション値増加量初期値	k_f	0.01
$D(t)$ 初期値	D_{init}	0.5
学習係数	k_l	0.01
結合荷重上限値	$W_{\max}$	1.1
f_i 減衰率A	$k_{f_1}^-$	0.95
f_i 減衰率B	$k_{f_2}^-$	0.5
$D(t)$ 増加量	k_d	0.01
f_i 増加率	k_f^+	0.5

学習成功率

図 2.7 は、**2.4.1** において 100 回の試行を行い、6,000,000 ステップまでに学習できたパターンの数の分布を表したものである。ここで、パターン数とは、ボールの軌道の種類と、そのボールが投じられた時にラケットが取りうる座標の種類とを掛けたもので、このシミュレーションではボールの軌道の種類が 6 でラケットの座標の種類が 4 であるから、パターン数は 24 である。図 2.7 より、全ての状況に対して正しく応答できるように学習できなかった場合でも、多くのパターンに対して正しく学習できていることが分かる。

また、学習の成功率を以下のように定義すると、図 2.7 の場合には、93.2%となる。

$$\text{学習成功率} = \frac{\sum \text{学習できたパターン数}}{\text{学習すべきパターン数} \times \text{試行回数}} \tag{2.25}$$

学習完了に要するステップ数

2.4.1 の試行において、24 種類のパターン全てについて正しく学習できたものに関して、学習が完了するまでに要したステップ数の分布を図 2.8 に示す。図 2.8 より、学習が完了するまでに要するステップ数は試行によりかなりばらつきがあることが分かる。図 2.8 において、学習が完了するまでに要したステップ数の平均は 2,945,357 であった。

打ち返し成功率

図 2.9 に、学習が成功した典型的な場合における、ステップ数と打ち返し成功率の変化の様子を示す。図 2.9 より、時間が経過するにつれて学習が進んでいることが分かる。学習が完了するステップ数に多少の差異は生じたが、学習が成功した試行の全てについて、これとほぼ同様の結果が得られた。ここで、打ち返し成功率とは、30,000 ステップの間にボールが領域の最下部に達した回数に対する、それを打ち返すことができた回数の割合である。なお、この 30,000 ステップの間に、ボールはおよそ 250 回領域の最下部に到達する。

また、失敗した典型的な事例について、図 2.10 に示す。このような学習の失敗は、ラケットの目標位置が間違っていても報奨信号が与えられることがあるという点に原因がある。たとえば、ラケットが領域の左端にある場合に、右端を目標位置とするような出力が発生すると、移動している途中で領域の中央の列に到達したボールを偶然に打ち返すことがある。このようなことが多発した場合には、間違った目標位置が学習されてしまう。そのため、ある程度までは学習が進むものの、全ての状況に正しく応答できるようにはならず、図 2.10 のような結果になってしまった。

出力のタイミング

図 2.9 の試行では学習開始からおよそ 2,000,000 ステップの時点で学習が完了している。図 2.11, 2.12 に、学習開始からそれぞれ 840,000 ステップと 2,000,000 ステップの時点で、ボールの軌道に対しどのようなタイミングで出力が行われるかを示す。

学習開始から 840,000 ステップの時点では、図 2.11 を見ても分かるように、軌道 (a)(f) 以外に対しては全く反応していない。また、軌道 (a) に対しては出力が発生するのが遅く、ラケットが左端にいる場合には打ち返すのが間に合わないことが分かる。しかし、学習が進行するにつれて、ネットワークは、全てボールの軌道に対して正しく応答できるようになっていく。学習が完了した 2,000,000 ステップの時点では、図 2.12 のように、全ての軌道に対して正しく応答できるようになっている。

図 2.12 を見ると、(c)(d) の軌道に対しては、他の場合に比べ 1 サイクル早く出力が発生していることが分かる。これは、ボールが中央 2 列の下端に到達する場合には、ラケットがどの位置にあろうと最高 2 サイクルの移動で打ち返せるのに対して、隅に到達するような軌道の場合には、逆の端にラケットがいた場合に 3 サイクルの移動が必要となるからである。

学習によって生成されたネットワーク構造

図 2.9 の試行での学習開始から 2,000,000 ステップの時点における、ネットワーク構造を図 2.13 に示す。上段から順に入力層、隠れ層、出力層であり、隠れ層は生成された順に左から並んでいる。なお、ニューロンを結ぶ線の太さは結合荷重の大きさを示している。

図 2.13 において、右から二列目への移動を表す出力層ニューロン 3 に着目すると、このニューロンは隠れ層ニューロン 2, 6, 7, 11, 21 と結合していることが分かる。しかしこれらの隠れ層ニューロンのうち、2, 11 以外のものは入力層ニューロンとの結合が弱く通常は発火しないことが分かる。ここで、隠れ層ニューロン 2 は軌道 (b) に対して、隠れ層ニューロン 11 は軌道 (f) に対して、それぞれ発火するものである。これらの軌道に対応してラケットを右から二列目の位置へ移動させることを、ネットワークが正しく学習していることが分かる。

学習済みパターン数の変化

図 2.14 は、図 2.9 の試行における、学習済みパターン数の変化の様子を示したものである。また図 2.15 は、図 2.9 の試行において、軌道 (a)(e)(f) について、軌道ごとの学習済みパターン数とステップ数との関係を示したものである。この場合パターン数と

はラケットの座標の種類である。学習の開始時においても、最初からラケットがボールの到達地点にいる場合には打ち返すことができるので、学習済みパターン数は１となる。図 2.15 の結果から、いずれの軌道でも学習の初期の段階ではラケットが遠い場合に打ち返しが間に合っていないが、すぐに充分に早いタイミングで移動を開始するように学習が行われていることが分かる。

行われる処理の変化

図 2.16 は、図 2.9 の試行において、30,000 ステップの間に結合荷重修正処理と動作安定化処理が行われた回数が、時間の経過と共にどのように変化していくかを示したものである。これを見ると、学習の初期では結合荷重修正処理が多く行われているが、学習が進んでいくにつれてその回数は減り、かわりに動作安定化処理が多く行われるようになってくることが分かる。図 2.16 では、学習開始からおよそ 1,000,000 ステップ付近で、結合荷重修正処理が行われた回数が激減していくのと同時に動作安定化処理が行われる回数が増えている。またその後、結合荷重修正処理がほとんど行われなくなるが、2,000,000 ステップ付近で再び結合荷重修正処理が多数実行され、この時点で学習が完全に収束したのが分かる。

パラメータを変更した場合の結果

各パラメータを変更してシミュレーションを行い、ステップ数と打ち返し成功率の関係を調べた。図 2.17 はフラストレーション値増加量の初期値 k_f を 0.02 とした場合において、学習が成功した試行と失敗した試行の典型的な結果である。この場合、k_f を 0.01 とした場合に比べてフラストレーション値が高速に増大していく。そのため、$k_f = 0.02$ として成功した場合には図 2.9 の試行に比べて高速に学習が収束する。しかしながら、$k_f = 0.02$ として失敗した場合には、学習の初期では図 2.10 の場合に比べて高速に成功率が向上するものの、途中から成功率が減退している。これは、高いフラストレーション値増加量の影響により、学習が完了したニューロンについても学習が行われてしまうことがあるためと考えられる。

図 2.18 は、学習係数 k_l を 0.02 とした場合の典型的な結果である。学習の初期の段階では、やはり図 2.9 の場合 ($k_l = 0.01$) よりも早く成功率が向上しているが、失敗した試行においては、途中から成功率が減退している。これは、学習が完了したニューロンについて間違った学習が行われてしまったときに、学習係数が大きいぶん結合荷重が大きく修正され、望ましくない出力を生じる可能性が高まるためと考えられる。また、成功した試行についても、最終的に学習が収束するまでに必要なステップ数は短

くなってはいない。図 2.17 の場合には学習の収束が劇的に早くなっていることから、このシミュレーション環境において学習に必要な時間を支配している要因を以下のように推測できる。このシミュレーション環境においては、学習済みのパターン数が多くなり、報奨信号が頻繁に与えられるようになると、フラストレーション値の増加が抑えられ、新しい学習を誘発するほどに大きなランダムパルスが生成されるのは非常にまれになる。図 2.14 からも分かるように、まさにこの段階で長い学習時間を要するので、学習係数を増やしても学習時間は短縮されないが、フラストレーション値の増加量を増やせば学習時間を短縮できる可能性があるのである。

図 2.19 はフラストレーション値の増加量の減衰率 $k_{f_1}^-$ を 0.90 とした場合の典型的な結果である。この場合には、成功例、失敗例ともに、図 2.9 の試行 ($k_{f_1}^- = 0.95$) の場合と特段の違いは見られなかった。ある程度まで学習が進むと動作安定化処理は極めて多く実行されるため、パラメータ $k_{f_1}^-$ の多少の違いは吸収されてしまうものと推測される。

動作安定化処理を省略した場合の結果

図 2.20 は、提案アルゴリズムにおいて動作安定化処理を実行しなかった場合の学習済みパターン数の変遷を示したものである。一度学習はかなり良い段階まで進むが、やがて結合荷重が過度に増大し、発火すべきでないニューロンが発火するようになってしまった。結果として、最終的な学習結果は非常に悪いものとなった。

この結果から、提案アルゴリズムにおいて動作安定化処理が有効に働いていることが分かる。

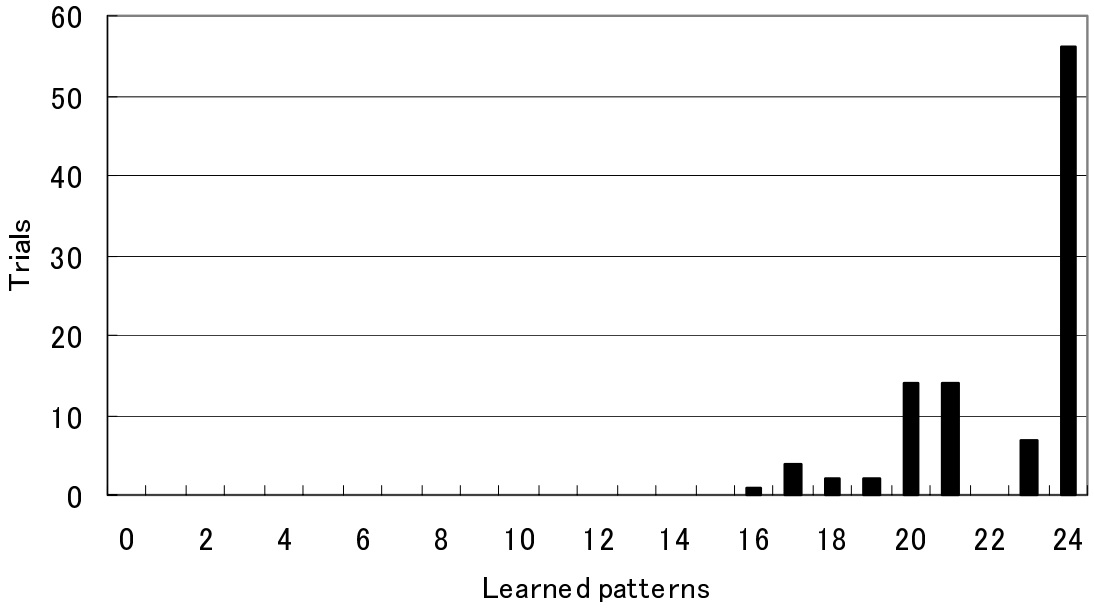


図 2.7 学習パターン数の分布
Fig. 2.7 Distribution of learned patterns.

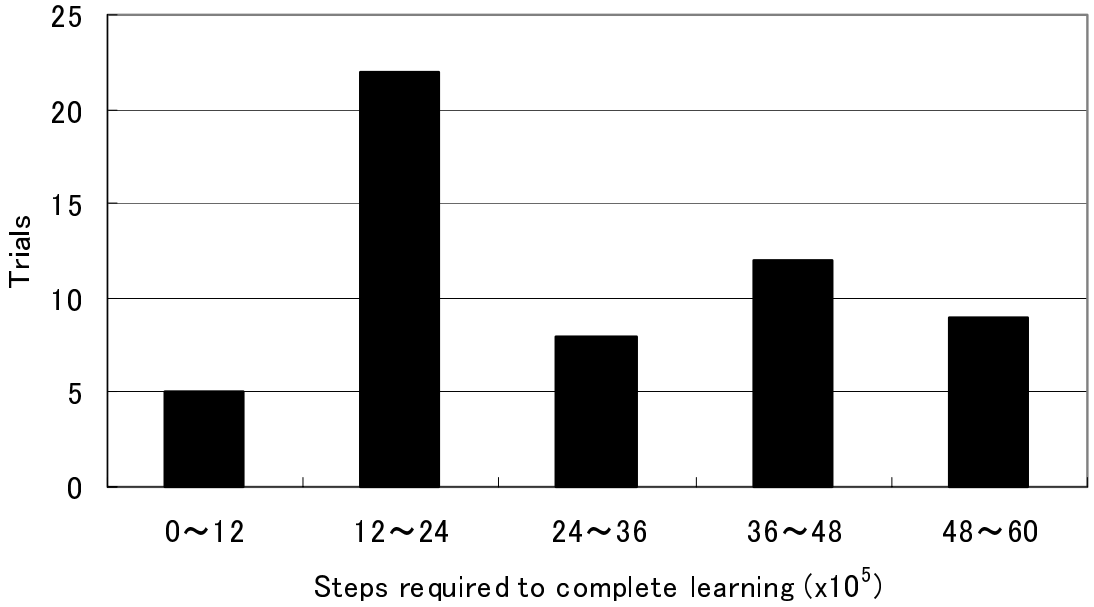


図 2.8 学習完了までのステップ数の分布
Fig. 2.8 Distribution of required steps to complete learning.

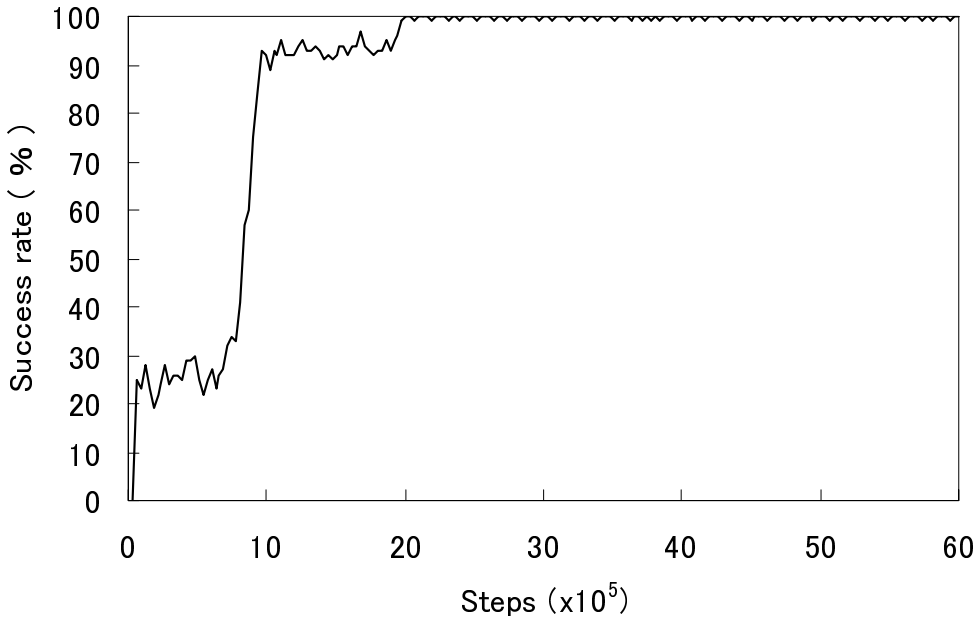


図 2.9 打ち返し成功率の変遷（成功例）

Fig. 2.9 Transition of success rate of rallies (an example of succeeded learning).

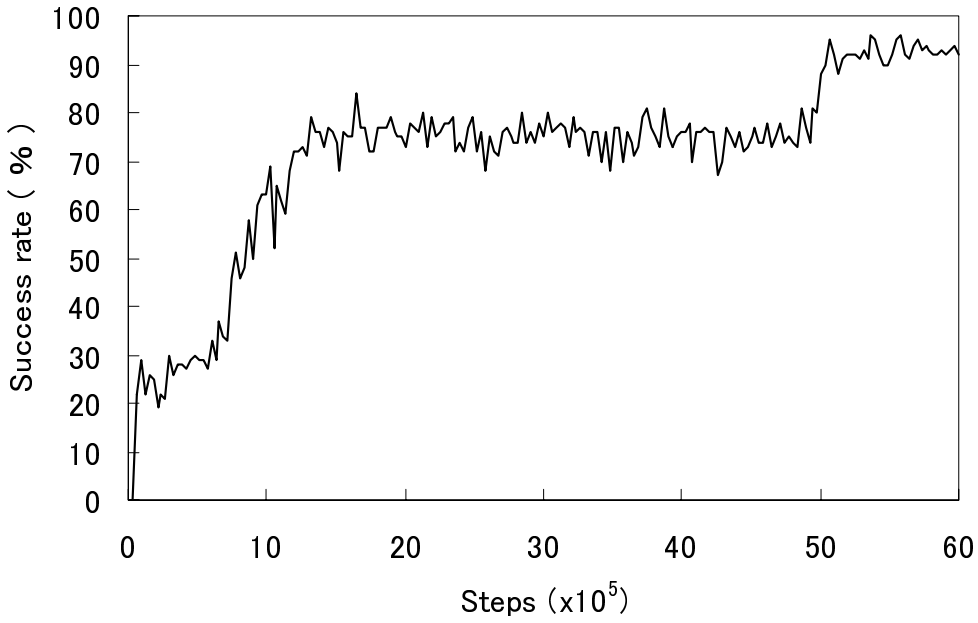


図 2.10 打ち返し成功率の変遷（失敗例）

Fig. 2.10 Transition of success rate of rallies (an example of failed learning).

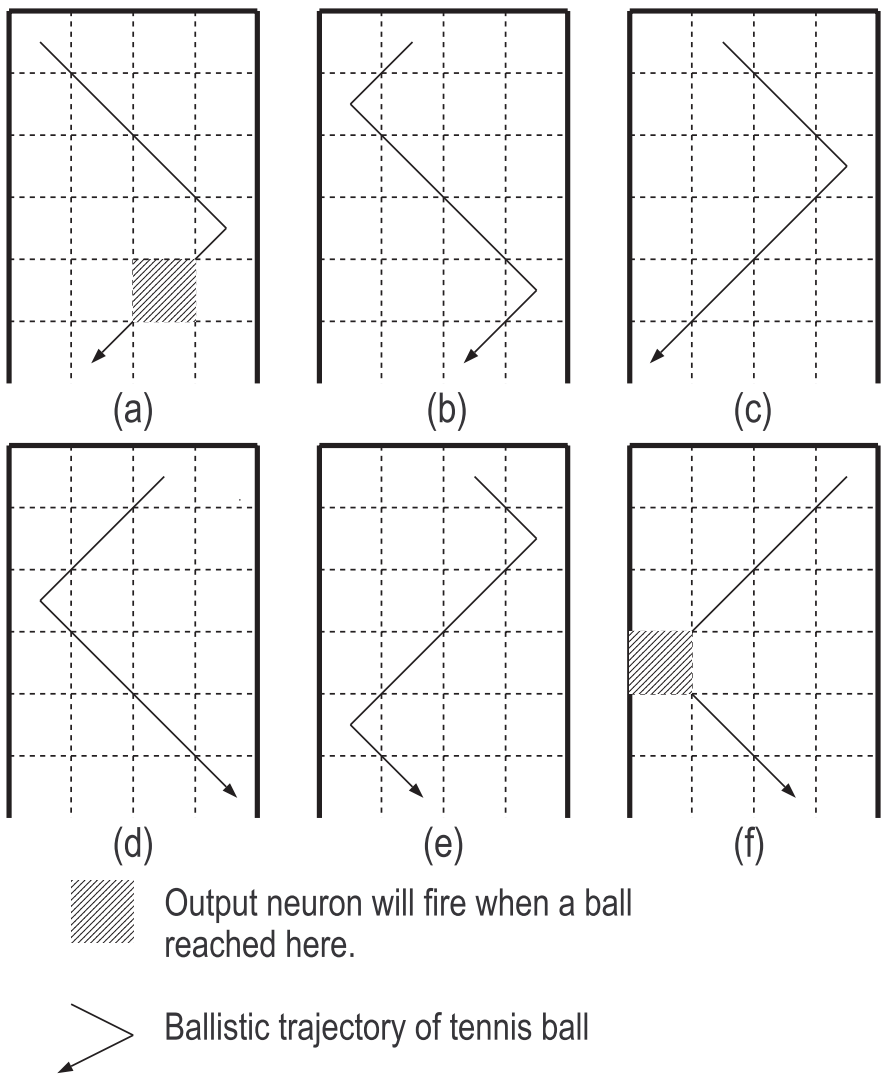


図 2.11 学習途中 (840,000 ステップ経過後) での出力タイミング

Fig. 2.11 Output timing in the middle of learning (an example of succeeded learning).

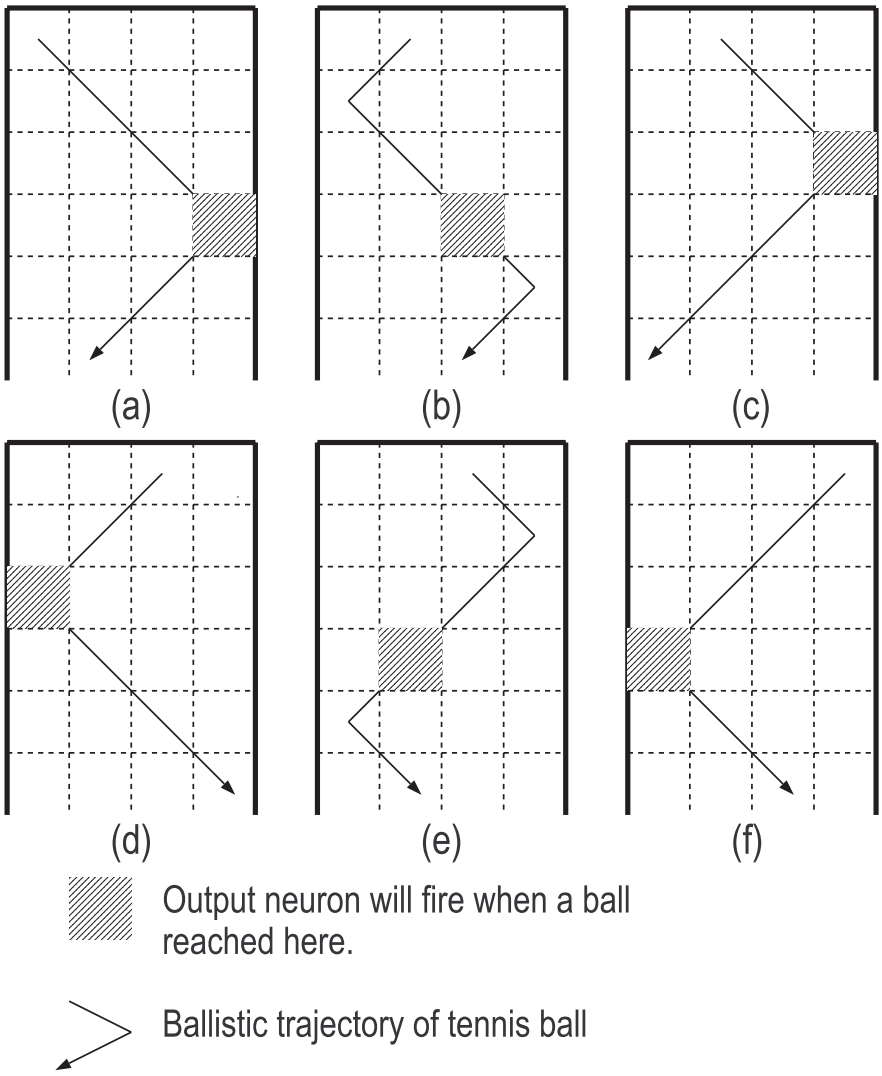
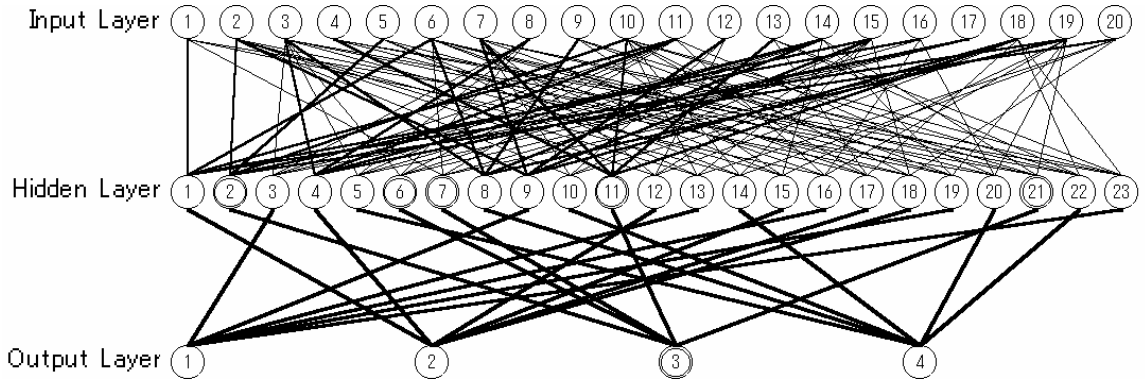


図 2.12 学習完了後の出力タイミング

Fig. 2.12 Output timing after the learning (an example of succeeded learning).



(a) ネットワーク

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20

(b) 入力層ニューロン
の番号と領域の対応

図 2.13 学習後のネットワーク構造

Fig. 2.13 Network structure after the learning (an example of succeeded learning).

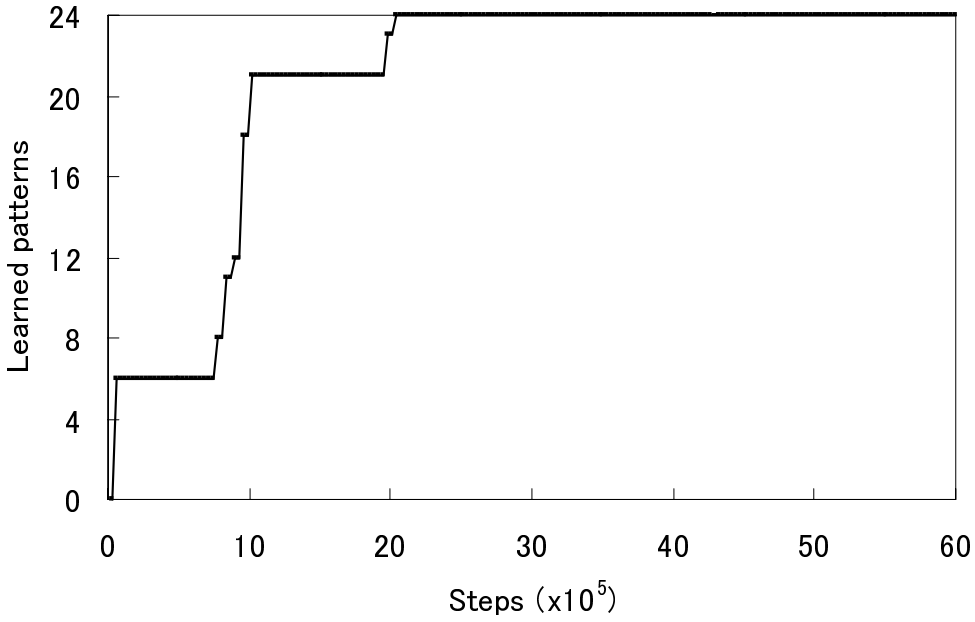


図 2.14 学習済みパターン数の変遷
Fig. 2.14 Transition of learned patterns.

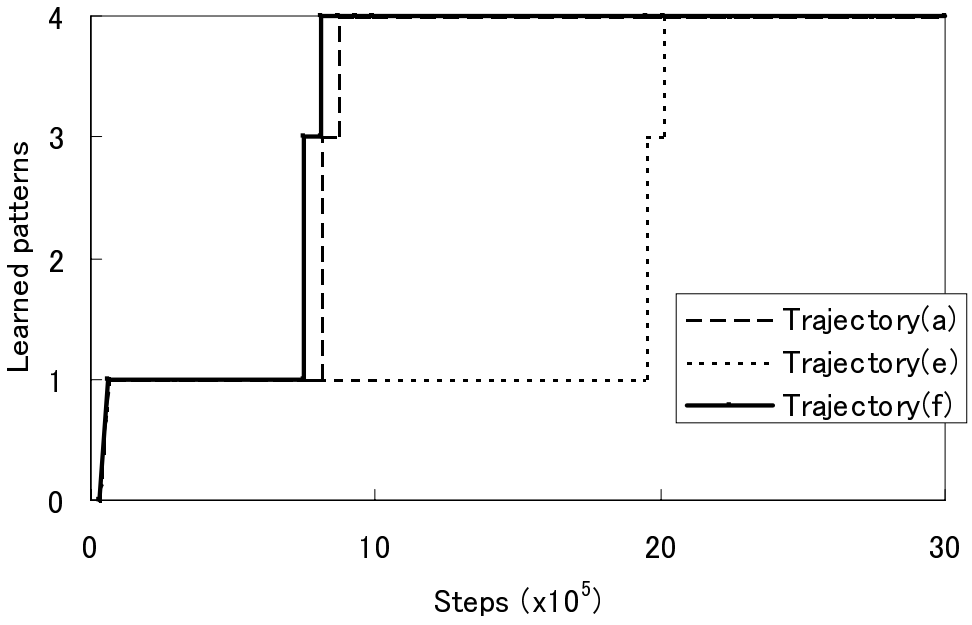


図 2.15 軌道ごとの学習済みパターン数の変遷
Fig. 2.15 Transition of learned patterns (trajectory based).

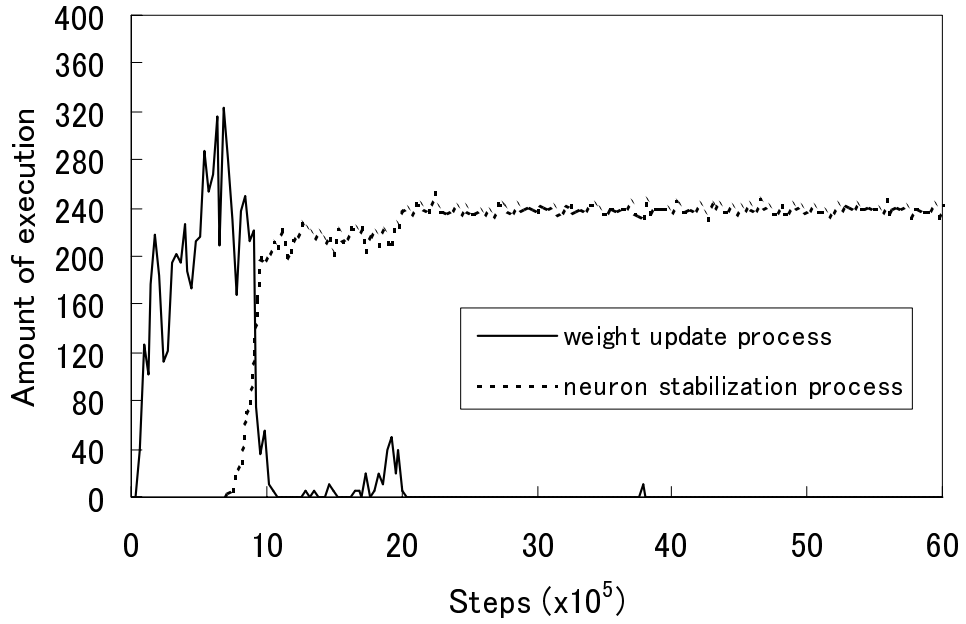


図 2.16 各処理の実行回数の変遷

Fig. 2.16 Transition of the amount of executions on each process.

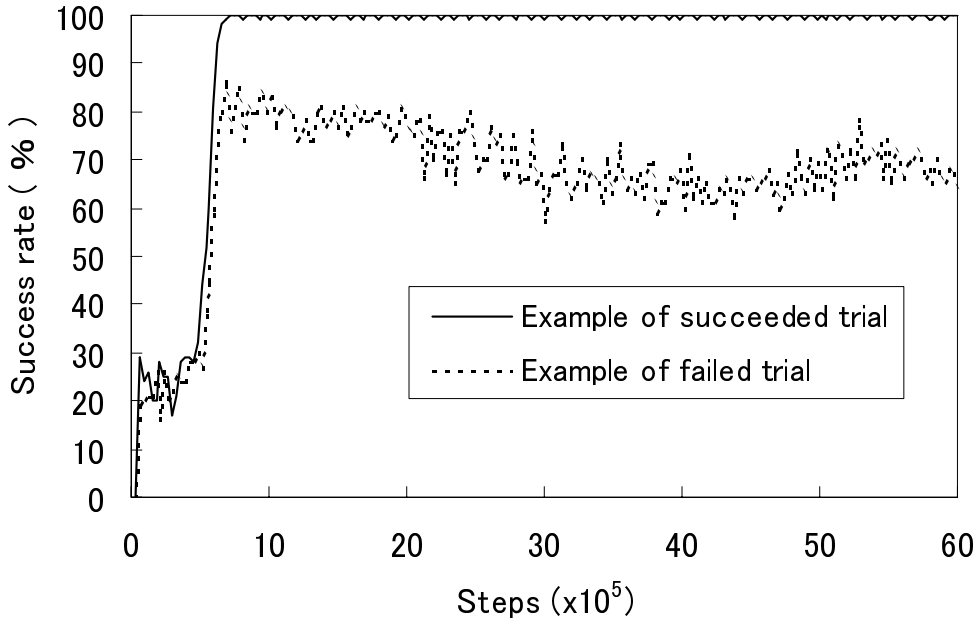


図 2.17 $k_f = 0.02$ とした場合の打ち返し成功率の変遷

Fig. 2.17 Transition of success rate of rallies ($k_f = 0.02$).

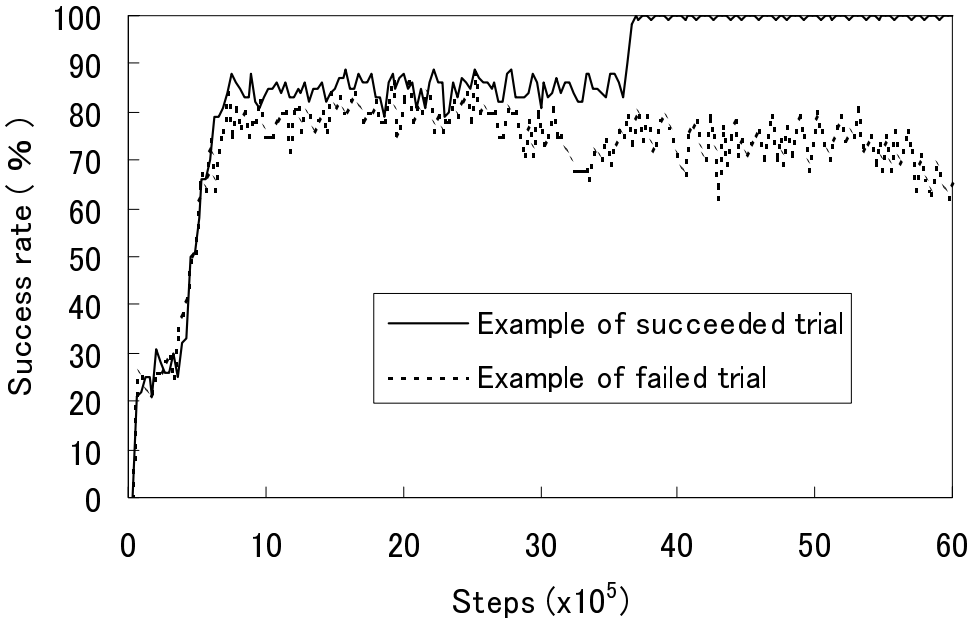


図 2.18 $k_l = 0.02$ とした場合の打ち返し成功率の変遷
Fig. 2.18 Transition of success rate of rallies ($k_l = 0.02$).

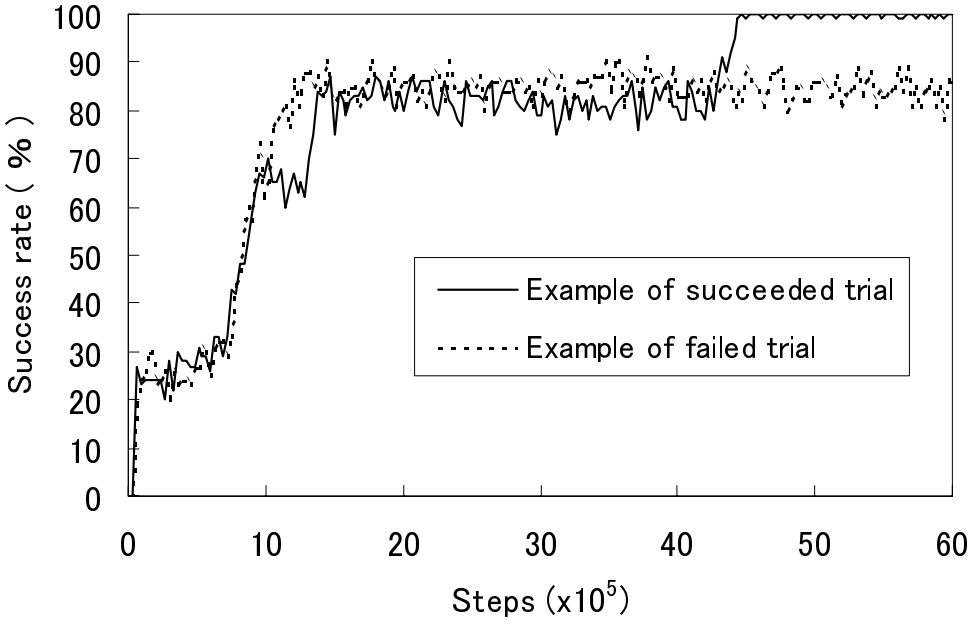


図 2.19 $k_{f_1}^- = 0.90$ とした場合の打ち返し成功率の変遷
Fig. 2.19 Transition of success rate of rallies ($k_{f_1}^- = 0.90$).

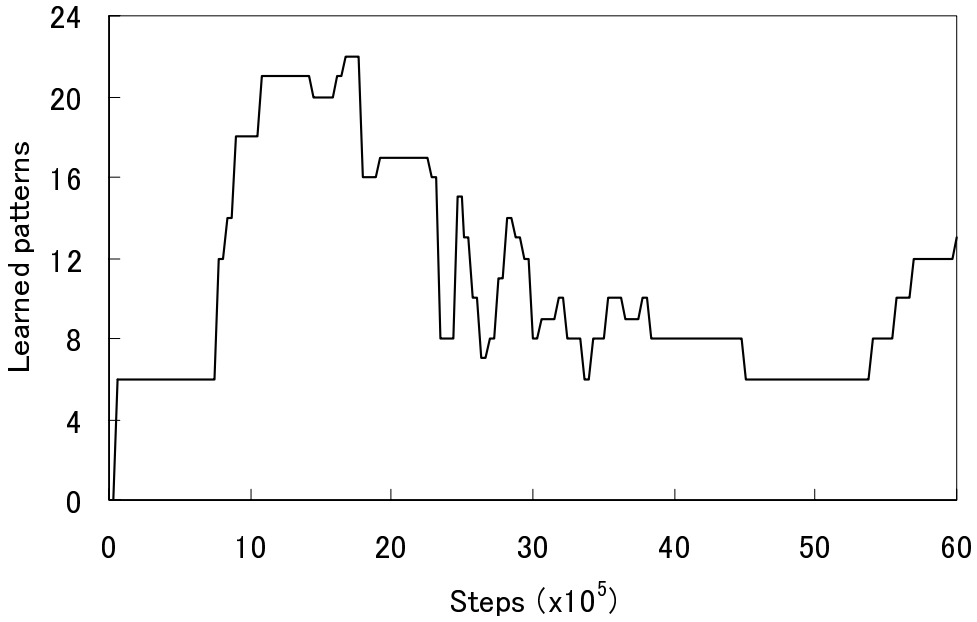


図 2.20 動作安定化処理を導入しない場合の学習済みパターン数の変遷

Fig. 2.20 Transition of learned patterns (without Neuron Stabilization Process).

2.4.2 実験環境２（シューティングゲーム）

概要

この実験では、提案モデルを使ってシューティングゲームを実行した。これは、図 2.21 に示すように、横４マス縦６マスの領域のなかで、並べられた砲台から弾丸を打ち出し、接近する敵を撃墜するというものである。弾丸を撃つことのできる砲台は一度にひとつだけなので、適切な砲台からタイミングよく弾を撃つ必要がある。構成としてはシミュレーション環境１に類似しているが、望ましい出力が複数存在するという点、また、出力タイミングが完全に正確でないと報奨が得られないという点において、より複雑な処理となっている。

敵は、画面上部のランダムな位置から $(-1, 1)$ ないし $(1, 1)$ の運動ベクトルを持って降下してくる。敵は、画面の側面に接触すると運動ベクトルを x 軸方向に反転し、画面の下部に到達すると消滅する。砲台から射出された弾丸は $(0, -1)$ の運動ベクトルを持ち、敵と接触するか画面上部に到達すると消滅する。弾丸が領域内に存在するかぎり、どの砲台も次の弾を撃つことはできない。また、敵が領域内に存在しない場合には、一定の確率で新しい敵が画面上部に出現する。

提案モデルへの入出力は、次のようにした。4×6 の領域のうち、最下段を除く 20 のマスに対応して 20 の入力層ニューロンを用意し、一定時間ごとに、敵が存在する位置に対応したニューロンに、発火閾値と等しい大きさの入力パルスを与える。この時間幅は、ネットワークの単位時間 (以下、ステップと呼ぶ) にして 12 ステップであり、これを 1 サイクルと呼ぶ。敵や弾の運動も 1 サイクルあたり 1 マスである。

また、出力層ニューロンを、4 マスの横幅に対応させて 4 つ用意した。これらはそれぞれ砲台に対する発射の合図を示すもので、いずれかが発火すると、対応する砲台から弾が射出される。1 サイクルの間に複数の出力層ニューロンが発火した場合には、いずれの砲台も発射することはできない。

ネットワークに対する報奨信号は、弾が敵と接触して撃墜した瞬間に与えられるとし、その値は 1.0 とした。また、罰信号は敵が領域の下端に達した瞬間に与えられるとし、その値は –1.0 とした。

このシミュレーションは、入力が与えられる時間間隔が未知であり、一回のボールの軌道に対応する一連の入力エピソードの長さも与えられていないという前提条件の下で行った。このような条件下では時系列入力を空間的にマッピングすることは容易ではなく、ASN を始めとする時系列処理能力を持っていない学習モデルの適用は困難である。なお、以下のシミュレーション結果は、前述の通り 1 サイクルが 12 ステップで形成される場合のものであるが、1 サイクルを 10 ないし 8 ステップとした場合にも同様の結果が確認されている。

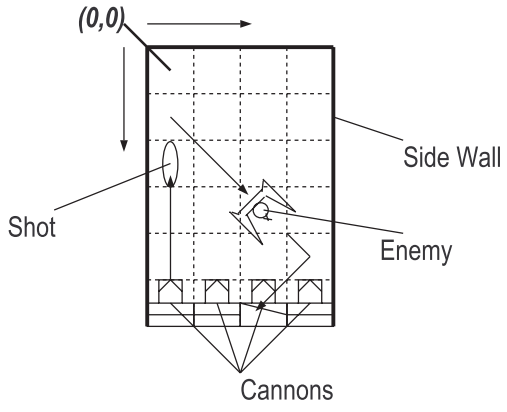


図 2.21 シューティングゲーム環境
Fig. 2.21 Shooting game environment.

具体的なパラメータは、表 2.2 の通りである。

学習成功率

図 2.22 は、**2.4.2** において 100 回の試行を行い、6,000,000 ステップまでに学習できたパターンの数の分布を表したものである。ここで、パターン数とは、敵の軌道の種類であり、このシミュレーションではパターン数は 6 である。図 2.22 より、全ての状況に対して正しく応答できるように学習できなかった場合でも、多くの場合、多数のパターンに対して正しく学習できていることが分かる。

また、式 2.25 で定義した学習の成功率は、図 2.22 の場合には、75.2%となる。

学習完了に要するステップ数

2.4.2 の試行において、6 種類のパターン全てについて正しく学習できたものに関して、学習が完了するまでに要したステップ数の分布を図 2.23 に示す。図 2.23 より、学習が完了するまでに要するステップ数は試行によりかなりばらつきがあることが分かる。図 2.23 において、学習が完了するまでに要したステップ数の平均は 3,420,000 であった。

撃墜成功率

図 2.24 に、学習が成功した典型的な場合における、ステップ数と撃墜成功率の関係を示す。学習完了までのステップ数には開きがあるものの、成功した試行の全てにつ

表 2.2 シューティングゲームにおける定数値の設定
Table 2.2 Simulation parameters on shooting game environment.

発火閾値	θ_v	1.0
フラストレーション値上限	θ_f	0.2
不応性閾値	θ_r	-0.01
連続報奨回数閾値	θ_s	100
パルス減衰率	d_v	0.94
不応性減衰率	d_r	0.94
不応性強度	k_r	1.0
ニューロン間ディレイ	k_d	3
フラストレーション値増加量初期値	k_f	0.01
$D(t)$ 初期値	D_{init}	0.5
学習係数	k_l	0.01
結合荷重上限値	$W_{\max}$	1.1
f_i 減衰率A	$k^-_{f_1}$	0.95
f_i 減衰率B	$k^-_{f_2}$	0.5
$D(t)$ 増加量	k_d	0.01
f_i 増加率	k^+_f	0.5

いて、これとほぼ同様の結果が得られた。この場合には、およそ 3,000,000 ステップで学習が収束し、あらゆる状況に対し適切に対応できているのがわかる。なお、撃墜成功率とは、30,000 ステップの間に敵が現れた回数に対する、敵を撃墜することができた回数の割合である。

また、失敗した典型的な事例について、図 2.25 に示した。この場合、複数の出力層ニューロンが発火するような結合荷重が学習されてしまい、成功率は低い水準でとどまっている。この原因は、ある軌道の敵に対して複数の砲台が均等に発射して撃墜することが続くと、同じ軌道に対して複数の出力層ニューロンが発火するようになってしまうことと考えられる。

学習済みパターン数の変化

図 2.26 に、図 2.24 の場合における、ステップ数と学習済みパターン数の関係を示した。また、図 2.25 の事例について、図 2.27 に示した。

出力のタイミング

図 2.28 は、図 2.24 の場合において、学習の完了後に、敵に対しどの砲台がどのようなタイミングで弾丸を射出するかの一例を示したものである。どの軌道に対しても一定の距離で撃墜するように学習されているのが判る。

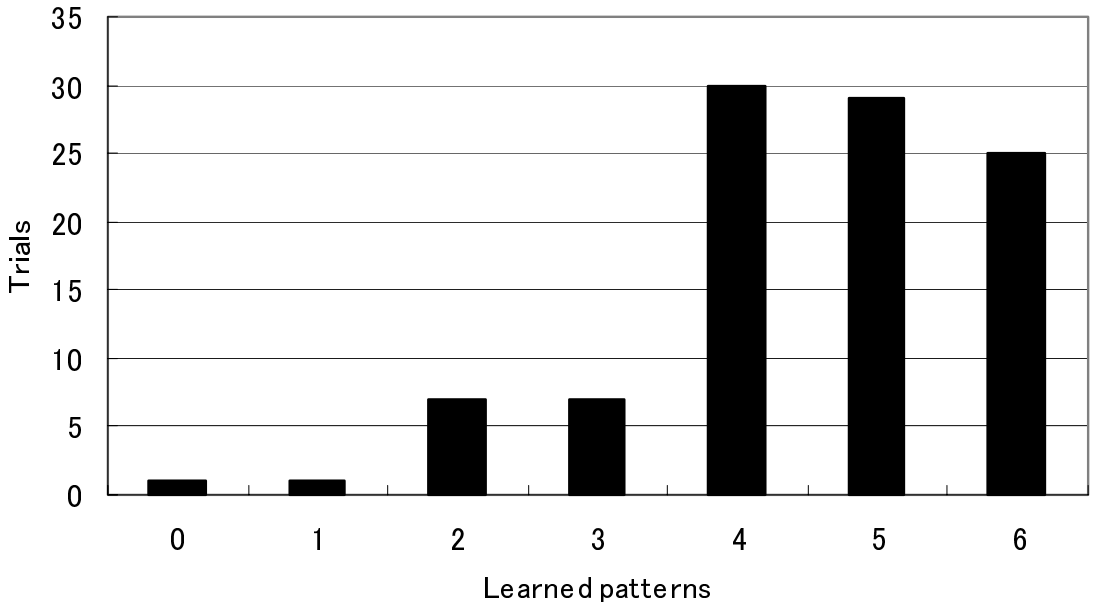


図 2.22 学習パターン数の分布

Fig. 2.22 Distribution of learned patterns.

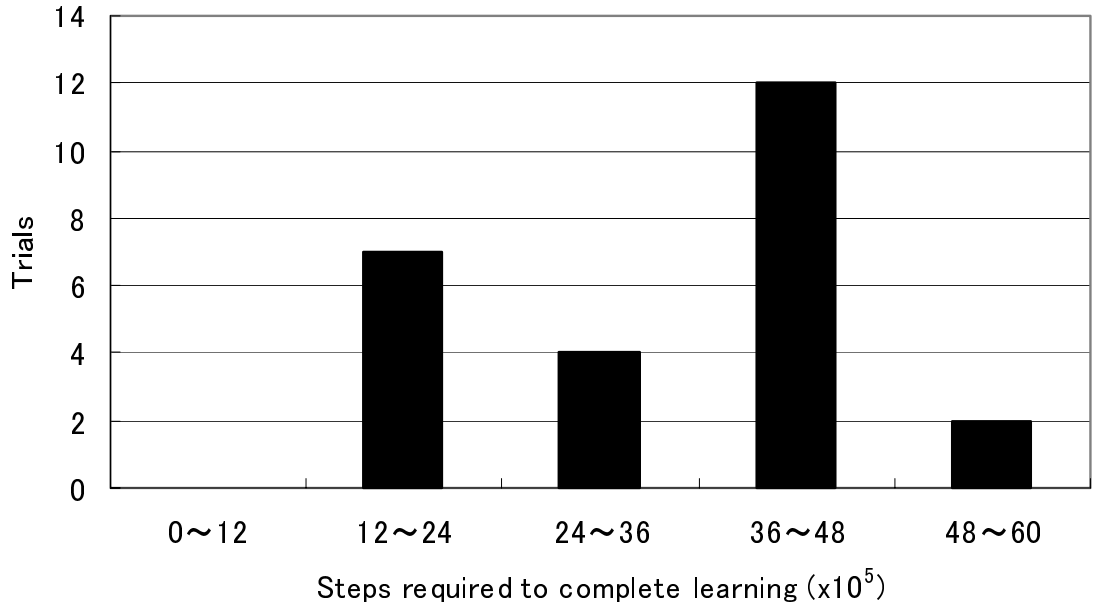


図 2.23 学習完了までのステップ数の分布

Fig. 2.23 Distribution of required steps to complete learning.

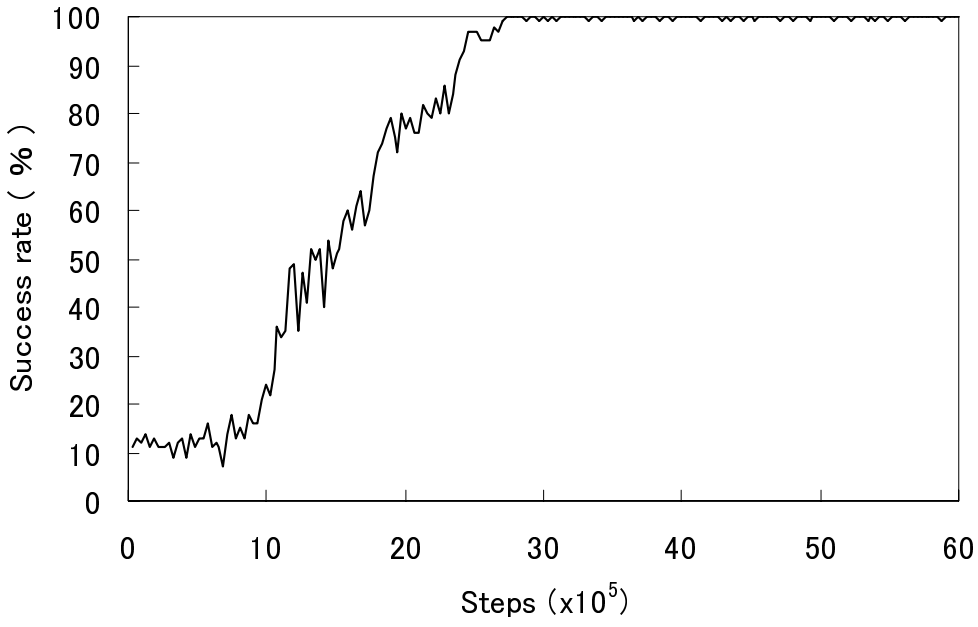


図 2.24 撃墜成功率の変遷（成功例）

Fig. 2.24 Transition of success rate (an example of succeeded learning).

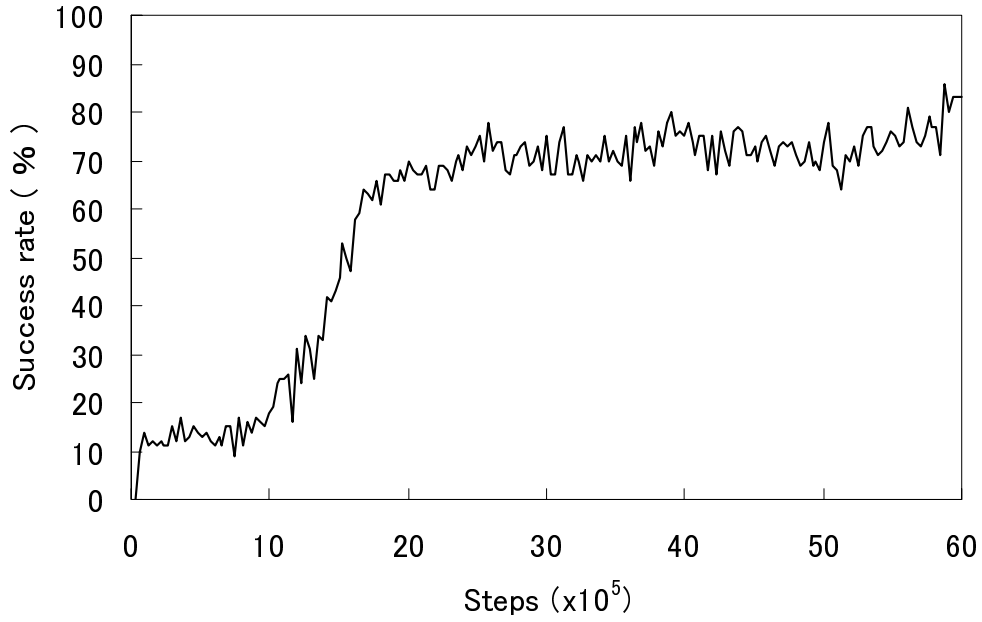


図 2.25 撃墜成功率の変遷（失敗例）

Fig. 2.25 Transition of success rate (an example of failed learning).

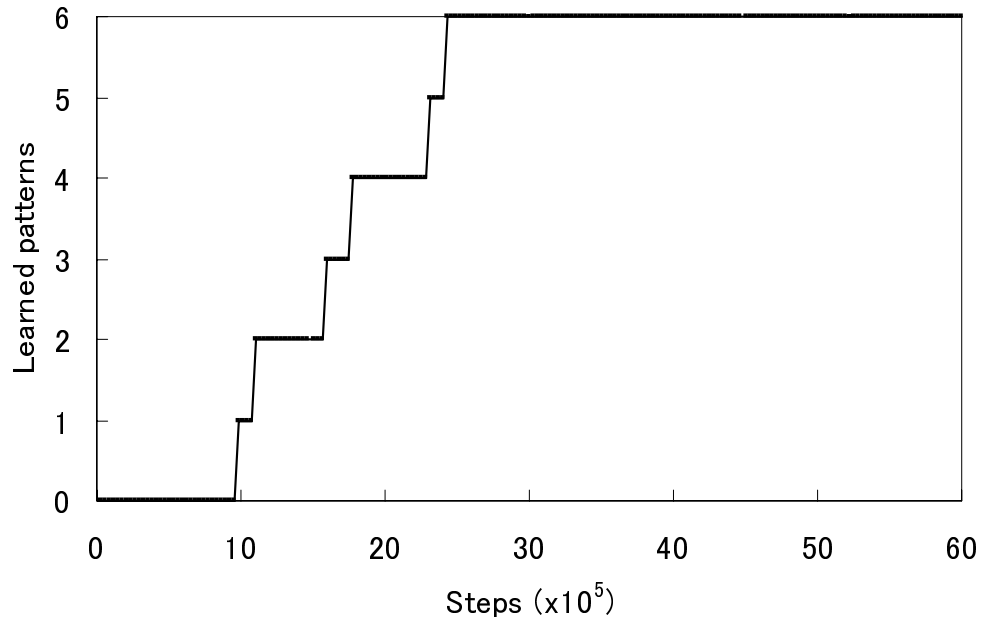


図 2.26 学習済みパターン数の変遷（成功例）

Fig. 2.26 Transition of learned patterns (an example of succeeded learning).

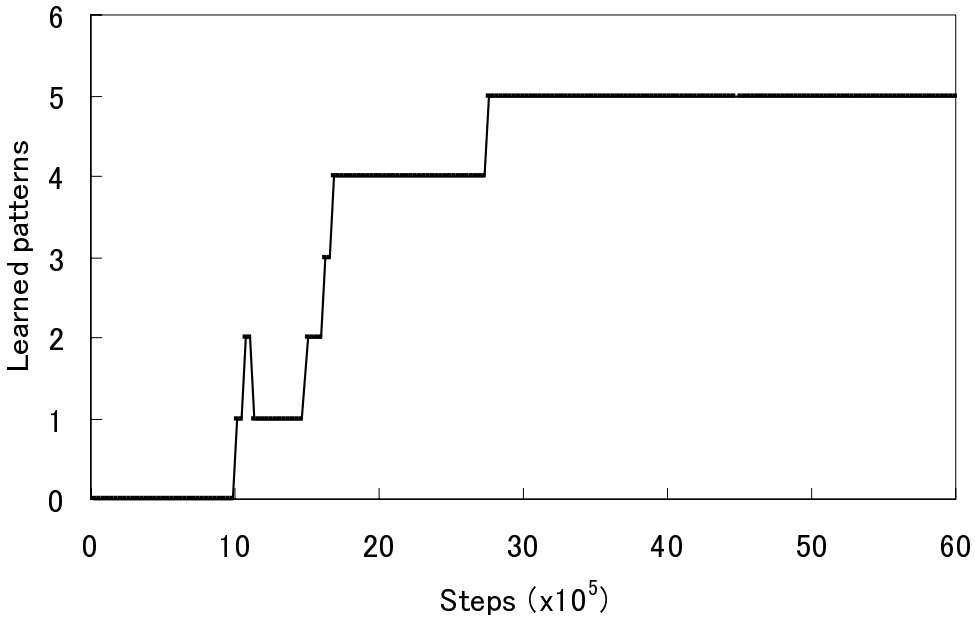


図 2.27 学習済みパターン数の変遷（失敗例）

Fig. 2.27 Transition of learned patterns (an example of failed learning).

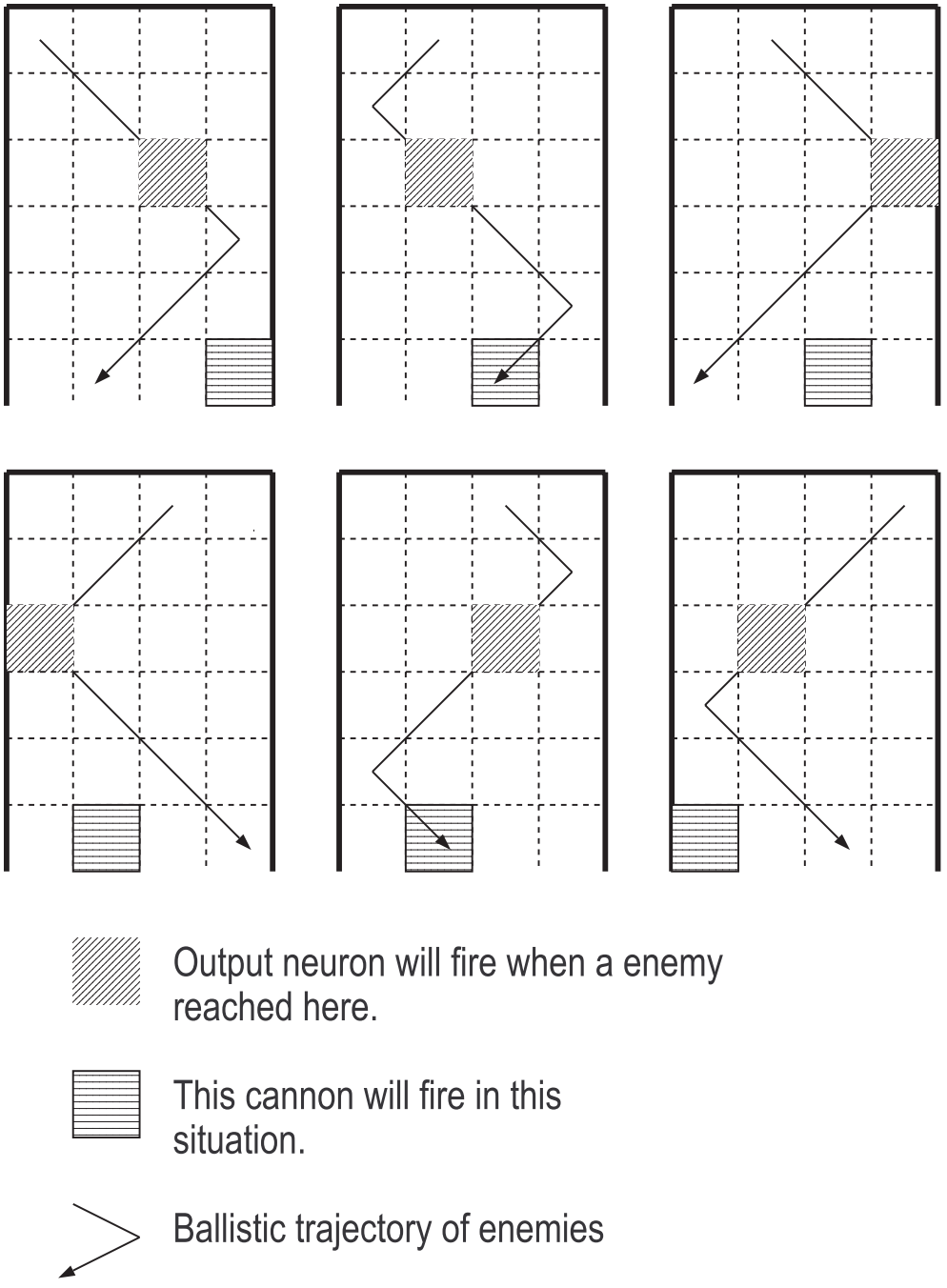


図 2.28 学習後の出力タイミング

Fig. 2.28 Output timing after the learning (an example of succeeded learning).

2.5 むすび

本章では、パルス駆動型ニューロン素子を用いた新しい階層型のネットワークの構造と、そのための強化学習アルゴリズムを提案した。このモデルでは、偶発性を利用することによって時系列的な入出力空間を探索し、外部からの強化信号に基づいて望ましい出力を出すように学習を行う。計算機シミュレーションを行い、時系列的な入力に対し適切なタイミングでの出力を学習できることを確認した。

提案モデルの今後の課題としては、学習精度の向上が挙げられる。特に、強化信号は常に正しい出力に対してのみ与えられるとは限らない。そのため、学習の過程で望ましくない出力が学習されてしまった場合にも、これを修正するような学習を行うことが必要になってくると考えられる。本章の計算機実験においても、学習が失敗した事例では、間違った出力が偶然に良い結果を生み報酬信号を導くという状況が連続したために、間違った出力を与えるようなネットワーク構造が確立されてしまうという場合が多く見られた。この問題の解決のために考えられる手法としては、不必要な結合の忘却処理などが挙げられる。

第 3章

部分観測マルコフ決定過程下の強化学習のためのパルスニューラルネットワーク学習則

本章では、パルスニューラルネットワークのための学習則として、パルスニューロン素子の時系列処理能力に着目した新しい強化学習則を提案する。減衰率の高いパルスニューロンは特にコインシデンスデテクタとして働くことが知られているが、提案モデルでは、減衰率の異なるパルスニューロン素子を組み合わせることで、時系列的な入力情報を処理し、部分観測マルコフ決定過程における曖昧な状態の識別を行う。提案するネットワークは四層構造をもつ非全結合のフィードフォワード型ニューラルネットワークであり、二層の隠れ層を構成するパルスニューロン素子が擬似的に環境中の状態を表現する。これらの素子は二次的な強化信号を生成することで、状態の評価関数に基づく従来の強化学習方式と類似した学習が可能となっている。二種類の計算機実験を行い、完全観測マルコフ決定過程および部分観測マルコフ決定過程の両方において、提案モデルが有効に働くことを確認している。

3.1 はじめに

コンピュータに知的な情報処理を達成させるための方法論として、人工ニューラルネットワークが着目され、誤差逆伝播法を始めとする学習則が広く研究されてきた [8], [63]。そのためのニューロン素子としては、生体の神経細胞の発火を単位時間あたりの平均発火量としてモデル化した、積分器型ニューロン素子が一般的である [1]。このような素子は、細胞の平均発火率が情報を表現しているとする単一細胞仮説 [2], [52] や Hebb アセンブリ仮説 [6] に基づくものであると言える。

しかしながら、近年、生体の神経細胞に見られる、時系列的なパルス (スパイク) に基づく入出力 [64], [65] を実装したパルスニューロンモデルの研究が進み、注目を集めている [22], [66]。パルスニューロンモデルを用いたパルスニューラルネットワークは、既に音源定位問題などにおいてその有用性が示されており [28]、工学的にその時系列処理能力が期待されている [40], [67]。また、より生体の神経細胞に近いことから、脳神経科学へのフィードバックや脳機能の補完デバイスへの応用を前提とした研究も進められ、成果を上げている [68]。

人工ニューラルネットワークの学習方式としては、特に工学的な応用性の面から強化学習 [43], [69] が着目されており、盛んに研究が行われている [47], [70]。そもそも強化学習は、系に対する望ましい出力を明示的に外部から与える必要のある教師あり学習とは異なり、結果が望ましいか否かだけを評価してやれば良い。そのため、多くの問題に適用可能であるだけでなく、人間の思いつかぬような解法が発見されやすい。また、生理学的な見地からも強化学習の妥当性を示す仮説が提唱されている [49]。

パルスニューラルネットワークにおける強化学習の研究としては、D.Gorse らが、従来の強化学習においては扱うことが難しいとされていた連続関数の学習を行うモデル [51] を提案している。しかし、パルスニューロン素子の時系列処理能力に着目した強化学習の研究は殆ど行われていないのが現状である。

工学的には近年において部分観測マルコフ決定過程 (Partially Observable Markov Decision Process, POMDP) 下での強化学習が注目を集めている [71], [72]。POMDP はセンサー入力などの限界により曖昧な状態 (不完全知覚状態) が存在する系であり、過去の入出力履歴により不完全知覚状態の識別を図る手法が主に研究されてきた [73], [74]。しかしながら、このような手法は一般に状態空間の複雑化に伴いメモリの使用量が爆発的に増加してしまい、学習にも膨大な試行錯誤を必要とする。

以上のような背景から、本章では、時系列入力の処理を目的としたパルスニューラルネットワークによる強化学習アルゴリズムを提案する。提案モデルでは、減衰率の異なる二種類のパルスニューロン素子を併用し、時系列情報の処理を行う。

3.2 で説明するように、個々のパルスニューロンは、入力された情報を内部状態と

いう形で抽象化して保持するものとなっている。そのため、時間履歴を使ったモデルで問題となる、メモリ使用量の爆発的増加が起こらない。一方で、抽象化による情報の欠落から、環境によっては学習精度の低下を招くことがあり得る。このような特徴から、パルスニューロン素子を用いた時系列処理は、以下のような環境で極めて有望な手法となりうる。すなわち、扱う必要のある入力列の長さなどの、環境に関する前提知識が与えられておらず、また、限られた状況における高精度の学習よりも、より幅広い状況に対しての総合的な学習が求められるような場合である。

提案モデルの学習能力の検証としては、二種類の計算機実験を行った。一方は cart-pole balancing problem と呼ばれるもので、状態空間が離散化されているために部分観測性が生じているものの、マルコフ決定過程として記述できるものである。そのため、時系列処理は必要とされないが、提案モデルの基礎的な学習能力の検証のために実験を行った。もう一方の実験は、筆者らが独自に考案した、ロボットエージェント等の行動制御を目標とした部分観測性の強い環境におけるものである。この問題では、長大な時系列の処理が必須であり、しかも、どれだけの長さの時系列を扱う必要があるのかを先験的に知ることができないため、履歴情報を元にした学習は非常に困難である。これらの実験により、提案するモデルの時系列処理能力が部分観測問題において有効に働くことを確認した。

以下、**3.2** で提案するネットワークの形状と使用するパルスニューロン素子の動作を述べ、**3.3** で学習アルゴリズムの詳細を述べる。**3.4** では提案モデルを用いた計算機実験により、その有効性を検証する。

3.2　ネットワークモデル

3.2.1　パルス駆動型ニューロン

提案モデルで用いたパルス駆動型ニューロン素子を図 3.1 に示す。このモデルでは、実際の神経細胞に見られる不応性や信号の時間的な加算などを考慮している。また、入出力としてパルス列を扱い、従来の積分器型のニューロンモデルに比べ、より実際の神経細胞に近いモデルとなっている。

このパルス駆動型ニューロンモデルでは、ある層のニューロン n_n に前階層のニューロン n_m からの入力パルスが到達すると、ニューロン n_n の内部状態 (内部電位)V_n が結合荷重 w_{mn} の分だけ上昇する。内部電位は時間の経過とともに徐々に静止電位まで減衰していく。もし内部電位が閾値を越えるとニューロンは発火し、出力パルスが時間遅れののちに次階層に到達する。

発火したニューロンの内部電位は静止電位にリセットされるとともに不応性の影響を受け、一時的にさらに電位が低下する。この不応性の影響も、時定数に則り徐々に減衰していく。

ニューロン n_n の時刻 t における内部状態 $V_n(t)$ は、他のニューロンからの入力残量 $I_n(t)$、不応性残量 $R_n(t)$ によって、式 (3.1)〜(3.3) のように定義される。

$$V_n(t+1) = I_n(t) - R_n(t) \tag{3.1}$$

$$I_n(t+1) = \begin{cases} 0, & O_n(t) = 1 \\ \sum_m \{w_{mn}(t-k_\mathrm{d}) \cdot O_m(t-k_\mathrm{d})\} & \\ \quad + (1-d_n) \cdot I_n(t), & O_n(t) = 0 \end{cases} \tag{3.2}$$

$$R_n(t+1) = \begin{cases} k_\mathrm{ref}, & O_n(t) = 1 \\ (1-d_n) \cdot R_n(t), & O_n(t) = 0 \end{cases} \tag{3.3}$$

ここで、d_n は内部状態の減衰率であり、この値はニューロン n_n がどの層に属するかによって決まる。提案学習則においては、H1 層についてのみ d_n を大きく設定し、残りの層では小さく設定する。k_d はパルス伝搬のディレイ、$w_{mn}(t)$ はニューロン n_m から n_n への時刻 t における結合荷重、$O_m(t)$ は n_m の出力を、k_ref は一回の発火がニューロンに与える不応性の影響の大きさを示す。k_d は通常は 1 であるとして差し支えないが、式 (3.2) ではこれを一般化して記述した。

また、ニューロン n_n の時刻 t における出力 $O_n(t)$ は、次式で定義される。

$$O_n(t) = \begin{cases} 1, & V_n(t) \geq \phi \\ 0, & V_n(t) < \phi \end{cases} \tag{3.4}$$

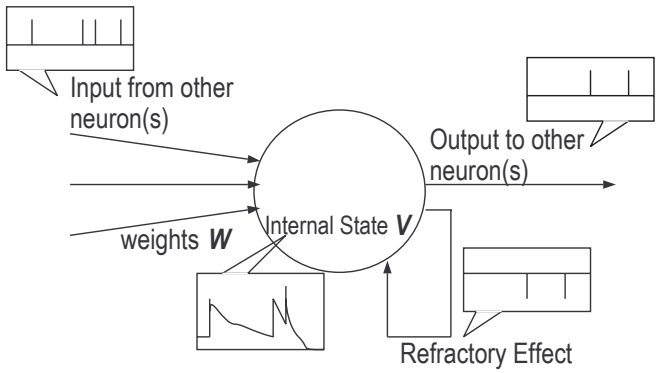


図 3.1 パルスニューロンモデル
Fig. 3.1　Pulse-based neuron model.

ここで、ϕ はニューロンの発火の閾値を表す。

ニューロン n_n の不応性残量 $R_n(t)$ は発火直後が最も大きくそこから徐々に減衰していくため、

$$R_n(t) \geq \theta_n^A \tag{3.5}$$

のような式が成り立つかどうかにより、ニューロンが過去一定時間内に発火したかどうかを判別することができる。この閾値の値は、内部状態の減衰率に応じて層ごとに異なる。また一つの層につき二種類の閾値、θ_n^A と θ_n^B（$\theta_n^A > \theta_n^B$）を用意している。

3.2.2　ネットワーク構造

提案モデルは図 3.2 に示すように、入力層、第一隠れ層 (H1 層)、第二隠れ層 (H2 層)、出力層の四層からなる階層構造のネットワークである。各層は **3.2.1** で述べるパルス駆動型ニューロン素子によって構成されている。このネットワークにおいて、ニューロンは一つ前の層のいくつかのニューロンとのみ結合しており、層間の結合は全結合ではない。また、同じ層内のニューロン間の結合は存在しない。

以下、i を入力層のニューロンに対するインデックスとして用い、入力層のニューロンを n_i^{IN} として示すこととする。同様に、H1 層のニューロンは j を用いて n_j^{H1} とし、H2 層では k を用いて n_k^{H2}、出力層では l を用いて n_l^O と表記することとする。任意の層のニューロンは m あるいは n を用いて n_m などと表現することとする。

また、入力層のニューロンの全体集合を N^{IN}、H1 層のそれを N^{H1}、H2 層のそれを N^{H2}、出力層のそれを N^O とする。

ここで、ニューロンの集合を返す関数 $U(n_n)$ を定義する。これは前階層のニューロ

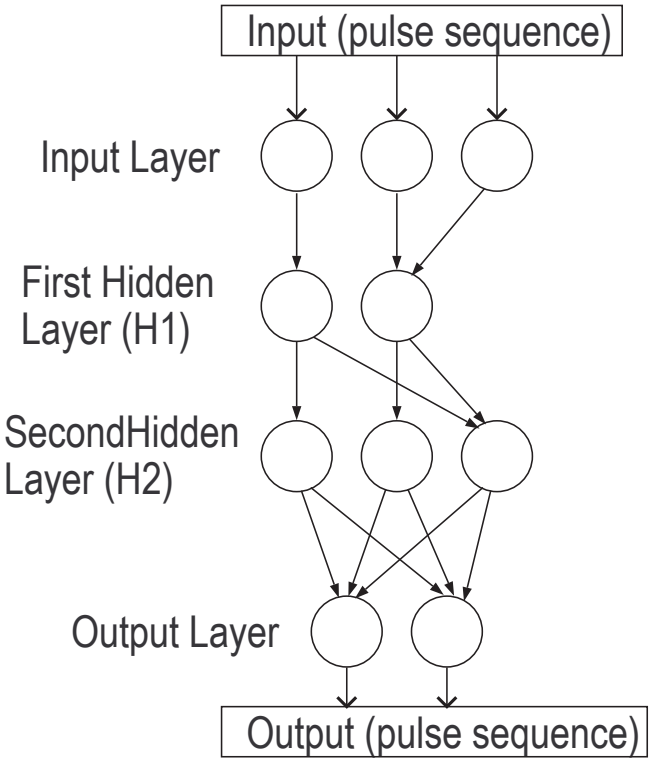


図 3.2 ネットワーク構成
Fig. 3.2　Network structure.

ンのうち、ニューロン n_n に対して結合を持つものの集合を返すものとする。

H1 層は **3.2.1** で説明する内部状態の減衰率が大きいニューロンで構成され、複数の入力層ニューロンからほぼ同時に入力を受けたときにのみ発火する。これに対し、H2 層は減衰率が小さいニューロンで構成され、時間差のある入力を受け取った場合にも発火するように設定される。

このような構造を用いた目的は、同時に与えられた入力と、時間差をのある入力とを区別することである。一般に、ある入力の組が時間差をもって与えられた時に発火するようなニューロンは、同じ入力の組が時間差なしに同時に与えられた場合にも発火してしまう。本モデルでは、まず H1 層によって入力の同時性を検出し、その後に H2 層で時間差のある入力を処理することにより、時間差の有無の区別を行うことができる。

ネットワークに対して与えられる強化信号は、スカラーで表現される。正のそれは外部の状態が望ましい場合に、負のそれは望ましくない場合に与えられる。しかしながら、ネットワークの出力と強化信号との時間的・確率的な相関は未知であるものとする。

3.3　学習アルゴリズム

3.3.1　概要

提案モデルにおける学習は、次の５つの処理によって構成される。すなわち、(1) 単純ネットワーク形成処理、(2) 複合ネットワーク形成処理、(3) 出力確率修正処理、(4) 抑制結合修正処理、(5) 内部強化信号修正処理、である。

図 3.3 に示すように、提案モデルは入力層・第一隠れ層 (H1 層)・第二隠れ層 (H2 層)・出力層の 4 層で構成される。このうち、H1 層においては内部状態の減衰率 d_n が大きく、それゆえに直近の入力にのみ反応して発火する。これに対し、H2 層においては d_n が比較的小さく、過去の入力情報を部分的に保持したうえで新しい入力に反応する。

提案モデルは非全結合のフィードフォワード型のネットワークであるが、初期段階においては H1 層および H2 層のニューロンは存在せず、入力層と出力層を結ぶ結合は存在しない。

単純ネットワーク形成処理は、H1 層および H2 層にニューロンを追加し、入力と出力を結ぶ回路を生成する。この処理では、H2 層の一つのニューロンに対して入力を与える結合は一つだけであるため、H2 層のニューロンは特段の機能を果たすことはなく、単なる中継点として働く。単純ネットワーク形成処理の目的は、ある瞬間の入力 (群) に対し、何らかの出力を発生するような回路を形成することである。

これに対し複合ネットワーク形成処理は、ある瞬間の入力群だけでなく、それ以前の入力群の影響も加味した上で出力を発生するような回路を形成する。この処理では H2 層のニューロンのみが追加される。この H2 層ニューロンに対しては複数の H1 層ニューロンからの結合が生成されるため、時系列的な入力情報の処理能力を有することとなる。

これらの処理で生成された H2 層のニューロンは、発火時には何れかの出力層ニューロンに対して出力を送る。この時、信号が送られる出力層ニューロンは確率的に決定される。この確率は出力確率修正処理において学習される。

提案モデルにおいて、同時あるいは連続的な入力を受けた H2 層のニューロンの発火は、それぞれ状態空間中の一つの状態を表現するものである。そのため、これらのニューロンが複数同時に発火することは望ましくない。抑制結合修正処理で、このような発火を抑えるように抑制結合を作成・強化する。

強化学習においては、エピソード (試行単位) についての前提知識がなく、しかも強化信号が長い入出力の系列の後で与えられるような場合には、その途中の状態が強化信号にどれだけ近いかを学習することが必要である。提案モデルでは、H2 層ニューロンに内部的な強化信号を与える機能を付加することにより、これを解決している。こ

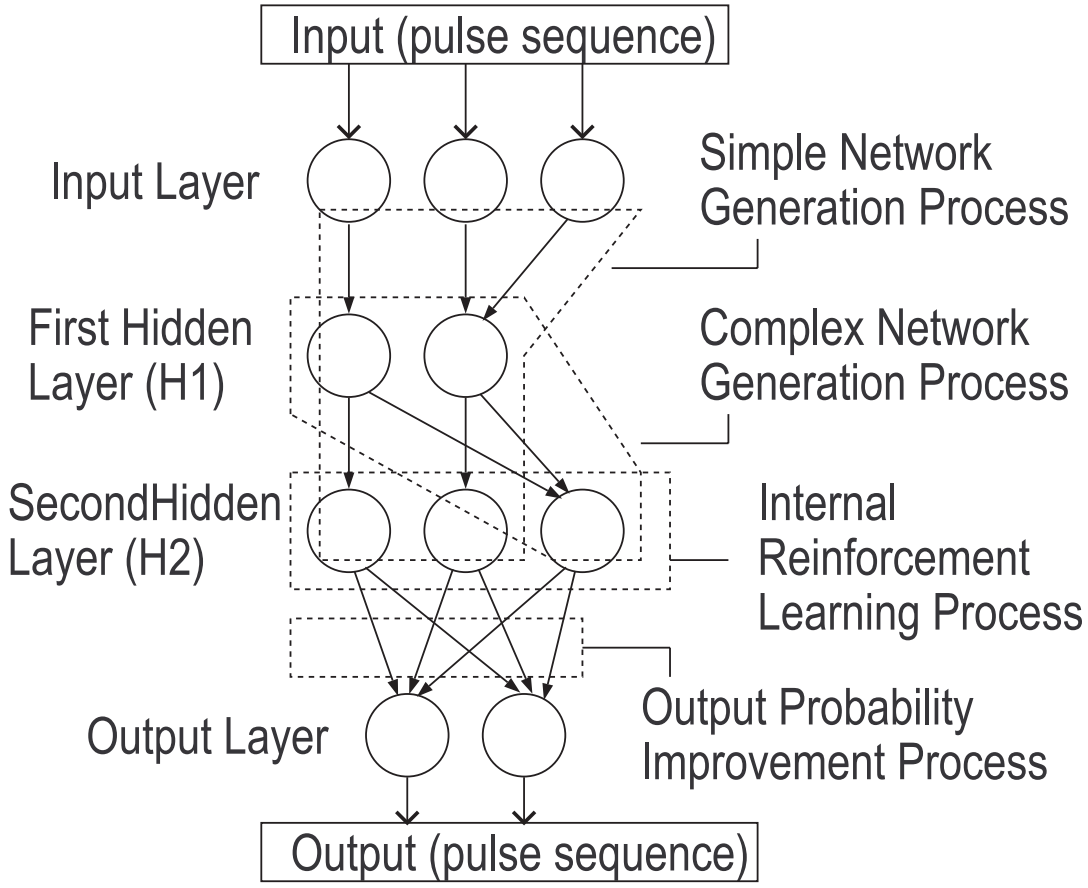


図 3.3 ネットワーク構造と学習処理の適用範囲
Fig. 3.3　Network structure and learning processes.

の信号の強さを学習するのが内部強化信号修正処理である。

以下、各学習処理について詳しく説明する。

3.3.2　単純ネットワーク形成処理

単純ネットワーク形成処理においては、図 3.4 に示すように、ある瞬間 (実際には多少の時間的ずれを許容する) に発火した入力層ニューロンの集合から、出力層ニューロンへと繋がるように H1 層・H2 層ニューロンを追加する。この処理により、再び同じ入力層ニューロンの集合がほぼ同時に発火した場合には、それに応じて H1 層・H2 層のニューロンが順次一つずつ発火し、出力が発生するようになる。

この処理では、動作の単位時間毎に、しかるべき H1 層ニューロンが存在するかどうかを調べる。もし存在しないのであれば、そのような H1 層ニューロンを作成し、あ

わせて H2 層ニューロンも作成する。ここで求められる H1 層ニューロンとは、すなわち、その瞬間に発火していたニューロンの集合に対して発火し、発火を伝搬していくようなニューロンである。

まず、次式を満たすような集合 X を求める。

$$X = \{n_i^{IN} | \quad n_i^{IN} \in N^{IN}, R_i(t) \geq \theta_i^A\} \tag{3.6}$$

ここで θ_i^A は、**3.2.1** で述べた、不応性判定のための閾値である。そして、

$$U(n_j^{H1}) = X \tag{3.7}$$

なるような H1 層ニューロン n_j^{H1} が存在するかどうかを調べる。このようなニューロンが存在しなかった場合に限り、以下の処理を実行する。

式 (3.7) を満たす n_j^{H1} が存在しなかった場合、これを満たすような H1 層ニューロン n_j^{H1} を任意に一個作成し、それから繋がる H2 層ニューロン n_k^{H2} を作成する。n_j^{H1} は、X の全要素から結合を受け、n_k^{H2} に対して結合を与える。また、n_k^{H2} は、N^O の全ニューロンに対して結合を与える。

ここで、

$$w_{ij} = w_0/S(X), \quad n_i^{IN} \in X \tag{3.8}$$

$$w_{jk} = w_0 \tag{3.9}$$

$$w_0 = \phi + \gamma \tag{3.10}$$

である。ただし $\S(X)$ は X の要素数であるとし、γ は発火のためのマージンである。なお、H2 層から出力層への結合については **3.3.4** において説明する。

3.3.3　複合ネットワーク形成処理

複合ネットワーク形成処理においては、ある時間幅のあいだに発火した H1 層のニューロンの集合から、出力層ニューロンへと繋がるように H2 層のニューロンを追加する。これにより、次にこれらの H1 層ニューロン群が同じような時間幅で発火した場合、それに応じて H2 層のニューロンが発火し、この時の入力列に対応した出力が発生する。

単純ネットワーク形成処理で作成された H1 層のニューロンは、ある瞬間の入力集合を一つの状態として認識し、出力を与えるものである。これに対し複合ネットワーク形成処理で生成される H2 層ニューロンは、時系列的な入力集合を一つの状態として認識し、出力を与えるものとなる。

この処理ではまず、正の強化信号に対する寄与がもっとも少ない H2 層ニューロン n_k^{H2} を一定時間ごとに選び出す。そして、図 3.5 に示すように、ニューロン n_k^{H2} が発火

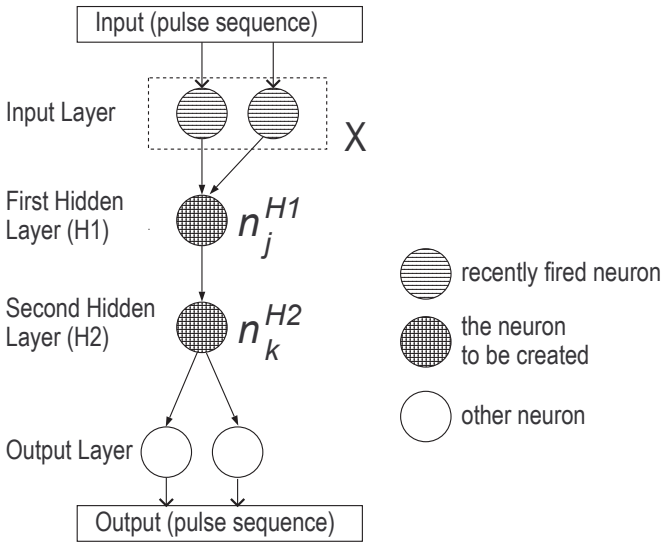


図 3.4 単純ネットワーク形成処理
Fig. 3.4　Simple network generation process.

するのに必要なそれよりもわずかに多い H1 層ニューロンからのパルスを受けて発火するような、新しい H2 層ニューロン n_n^{H2} を作成する。正の強化信号に対する寄与が最も少ない、すなわち学習がうまくいっていないニューロンは、部分観測問題における曖昧な (不完全知覚) 状態を表現している可能性が高い。そこで、より過去の入力情報を加味して出力を伝搬するような新しい素子を作る。

具体的には、全ての H2 層ニューロンについて予め定めた時間間隔 ρ 毎に、任意の評価式、たとえば

$$A_k(t) = P_k(t) \tag{3.11}$$

を適用する。ここで、$P_k(t)$ は **3.3.6** にて述べる内部強化信号生成量である。そして、

$$n_k^{H2} = \min_{n_m^{H2} \in N^{H2}} A_m(t) \tag{3.12}$$

なるようなニューロン n_k^{H2} にマーキングを行う。

マーキングされたニューロン n_k^{H2} が発火した際には、n_k^{H2} に対して結合を持たない H1 層ニューロンの中で最も遅く発火したニューロン、すなわち

$$n_j^{H1} = \max_{n_m^{H1} \in N^{H1} - U(n_k^{H2})} R_m(t) \tag{3.13}$$

なるニューロン n_j^{H1} を求める。ここで、

$$R_j(t) \geq \theta_j^B \tag{3.14}$$

でない場合には以下の処理は無視し、次の n_k^{H2} の発火を待つ。

次に、

$$U(n_n^{H2}) = U(n_k^{H2}) + n_j^{H1} \tag{3.15}$$

を満たすような H2 層ニューロン n_n^{H2} が存在するかどうかを調べる。このようなニューロンが存在しなかった場合に限り、以下の処理を実行し、n_k^{H2} のマーキングを消去する。

式 (3.15) を満たす n_n^{H2} が存在しなかった場合、これを満たすような H2 層ニューロン n_n^{H2} を作成する。n_n^{H2} は、$U(n_k^{H2}) + n_j^{H1}$ の全要素から結合を受け、N^O の全要素に対して結合を与える。ここで、H1 層ニューロン集合 $\{U(n_k^{H2}) + n_j^{H1}\}$ に対するインデックス m が、$R_m(t)$ が昇順に並ぶようにソートされているものとすると、

$$\begin{aligned} w_{mn} &= w_0 \cdot m / S(U(n_k^{H2}) + n_j^{H1}) \\ &\quad - \sum_{p=1}^{p<m} w_{pn} \cdot (1 - d_n)^{(\log_{1-d_p}(R_p(t)) - \log_{1-d_p}(R_m(t)))} \end{aligned} \tag{3.16}$$

として初期結合荷重を設定する。

これにより、次に同じタイミングで入力層ニューロン群が発火した場合、あるいは類似したタイミングで発火した場合に、それに応じて n_n^{H2} が発火し、出力が与えられることとなる。

例として、この学習処理の後で全く同じタイミングでニューロン群 $\{U(n_k^{H2}) + n_j^{H1}\}$ が発火した場合を考える。m 番目のニューロンからの出力パルスが n_n^{H2} に届く時刻を t_m とすると、t_m における n_n^{H2} の内部状態 $I_n(t_m)$ は、$m-1$ 番目のニューロンからの出力の到達時刻 t_{m-1} を用いて、

$$I_n(t_m) = I_n(t_{m-1}) \cdot (1 - d_n)^{t_m - t_{m-1}} + w_{mn} \tag{3.17}$$

と表せる。ここで、$m > 1$ の範囲において、

$$t_m - t_{m-1} = \log_{1-d_{m-1}}(R_{m-1}(t)) - \log_{1-d_m}(R_m(t)) \tag{3.18}$$

であるので、式 (3.16) を用いると、

$$I_n(t_m) = w_0 \cdot m / S(U(n_k^{H2}) + n_j^{H1}) \tag{3.19}$$

となる。t_1 の直前における n_n^{H2} の内部状態を 0 と仮定すると、式 (3.19) は $m = 1$ についても成り立つ。式 (3.19) より、時刻 $t_{S(U(n_k^{H2}) + n_j^{H1})}$ より前の時刻において n_n^{H2} が発火することはなく、逆に、時刻 $t_{S(U(n_k^{H2}) + n_j^{H1})}$ においては必ず発火する。つまり、n_n^{H2} は学習時と同じ入力列が完結した時に限って発火することとなる。

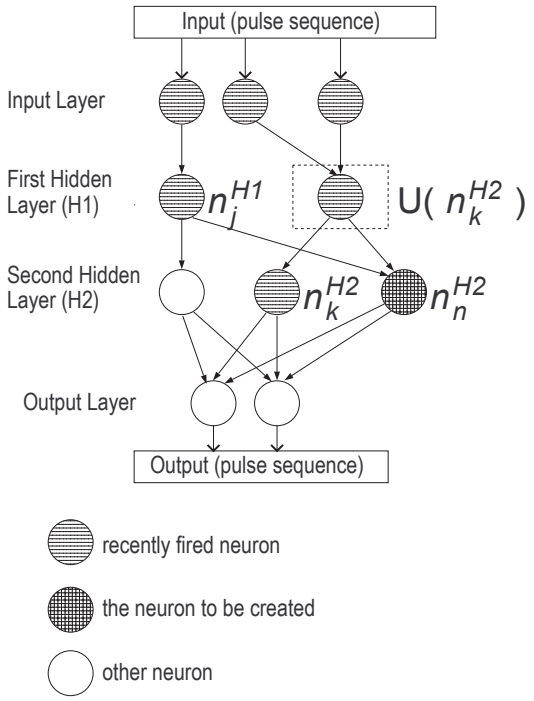


図 3.5 複合ネットワーク形成処理
Fig. 3.5　Complex network generation process.

3.3.4　出力確率修正処理

H2 層ニューロンのそれぞれからは出力層全体に対して結合構造が作成されるが、一つの H2 層ニューロンが発火した時にパルスが送られるのは出力層ニューロンのうちの一つだけである。H2 層ニューロン n_k^{H2} と出力層ニューロン n_l^O との結合荷重 w_{kl} は、発火時に送られる信号の強さではなく、その経路を通って信号が送られる確率を左右する。出力確率修正処理では、以下に述べるように w_{kl} を修正する。この模式図を図 3.6 に示す。

H2 層ニューロン n_k^{H2} が発火した際に、どの出力層ニューロンに対して出力が送られるかについては、ϵ-greedy 法を用いる [43]。すなわち、確率 ϵ で全ての出力層ニューロンの中から均等に選択を行い、それ以外の場合では最大の w_{kl} をもつ出力層ニューロンを選択し、このニューロンに対してパルスを送る。このとき n_l^O が受け取る入力の強さは w_0 となる。

この処理では、$n_k^H 2$ から n_l^O へパルスが送られた場合、まず w_{kl} を減衰させる。その後一定期間内に強化信号が与えられた場合には、それが正なら結合を強化し、負なら結合を減退させる。これにより、発火後に正の強化信号を受けとりやすい出力が促進され、報酬に寄与しない出力や、罰を与えられるような出力は行われないようになっ

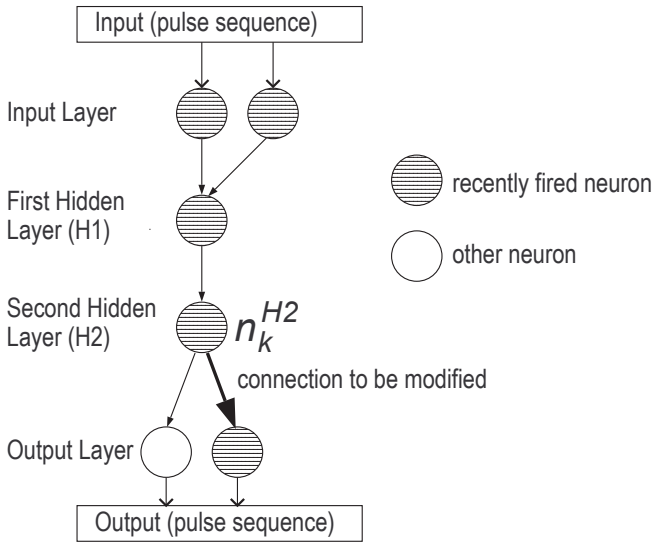


図 3.6 出力確率修正処理
Fig. 3.6　Output probability improvement process.

ていく。なお w_{kl} の初期値は 0 とする。

H2 層ニューロン n_k^{H2} が発火し、ニューロン n_l^O に出力が送られると、

$$w_{kl}(t+1) = (1 - d_{\text{prob}}) \cdot w_{kl}(t) \tag{3.20}$$

に従い、n_l^O への結合を減衰させる。ここで、d_{prob} は出力確率減衰率である。もし

$$R_k(t) \geq \theta_k^B \tag{3.21}$$

である間に強化信号 $s(t)$ が与えられた場合には、

$$w_{kl}(t+1) = w_{kl}(t) + s(t) \tag{3.22}$$

として修正を行う。

3.3.5　抑制結合修正処理

前述のように、H1 層のニューロンは同時に発火した入力層ニューロン群を一つの状態として認識するためのものであり、H2 層のニューロンは、時系列的に発火した入力層ニューロン群を一つの状態として認識するためのものである。そのため、同時に複数の H1 層ニューロンが発火したり、あるいは同時に複数の H2 層のニューロンが発火することは望ましくない。抑制結合修正処理は、このような発火を検出し、図 3.7 に

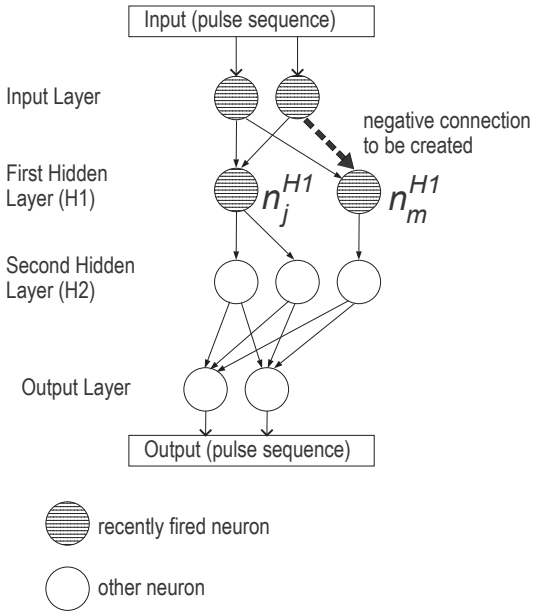


図 3.7 抑制結合修正処理

Fig. 3.7　Negative link modification process.

示すように、これを抑止するような抑制結合を作成する。以下、H1 層に対する抑制結合修正処理について述べるが、全く同様の処理を H2 層に対しても適用する。

まず、H1 層ニューロン n_j^{H1} が発火した場合、

$$R_m(t) \geq \theta_m^A, U(n_m^{H1}) \subset U(n_j^{H1}) \tag{3.23}$$

を満たす H1 層ニューロン n_m^{H1} を調べる。そして、

$$w_{im}(t+1) = w_{im}(t) - k_{\mathrm{neg}}, \quad n_i^{IN} \in U(n_j^{H1}) - U(n_m^{H1}) \tag{3.24}$$

として結合荷重を減退させる。ここで、k_{neg} は抑制結合の強化量を示す定数である。

n_m^{H1} は、n_j^{H1} に対して結合を与えている入力層ニューロンの一部からのみ結合を受けている。しかし式 (3.8) より、それぞれの結合は n_j^{H1} に対するそれよりも強い。つまり、n_m^{H1} は発火しやすいニューロンであり、n_j^{H1} はより限定された状況でのみ発火するニューロンであるといえる。そのため、n_j^{H1} を発火させるのに必要な入力層ニューロンが発火した場合には、n_m^{H1} でなく n_j^{H1} が発火する事が望ましい。この処理は、n_j^{H1} を発火せしめるのに必要な入力層ニューロンから、n_m^{H1} に対して負の結合を生成することで、上記の状況での n_m^{H1} の発火を抑制する。

3.3.6　内部強化信号修正処理

H2 層のニューロンのそれぞれは、発火時に内部的な強化信号を生成する。内部強化信号修正処理では、図 3.8 に示すように、この信号の量を学習する。外部から強化信号が与えると、一定時間以内に発火していた H2 層ニューロンが影響を受け、発する内部強化信号が増大する。再びこのニューロンが発火すると、さらにそれ以前に発火していた H2 層ニューロンが影響を受けるという様に、強化信号は H2 層ニューロンを伝搬していく事となる。

H2 層ニューロン n_k^{H2} が発火した場合、その内部強化信号生成量 P_k は

$$P_k(t+1) = (1 - d_{\text{pseudo}}) \cdot P_k(t) \tag{3.25}$$

に従い減衰し、$P_k(t)$ に等しいだけの内部強化信号がネットワークに与えられる。内部強化信号は、あらゆる面について外部からの強化信号と同等に扱われる。

ネットワークに強化信号 $s(t)$ が与えられると、

$$\begin{aligned} P_k(t+1) = (1 - d_{\text{pseudo}}) \cdot (P_k(t) + s(t) \cdot R_k(t)/R_{\text{total}}(t)), \\ n_k^{H2} \in N^{H2}, R_k(t) \geq \theta_k^B \end{aligned} \tag{3.26}$$

として、一定時間内に発火していたニューロンの間で強化信号の影響が割り振られ、その内部強化信号生成量が増減する。ここで、

$$R_{\text{total}}(t) = \sum_{n_m^{H2} \in N^{H2}, R_m(t) \geq \theta_m^B} R_m(t) \tag{3.27}$$

である。

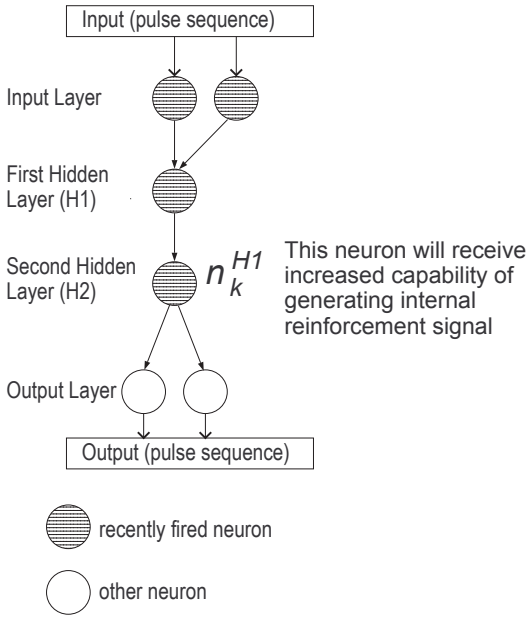


図 3.8 内部強化信号修正処理
Fig. 3.8　Internal reinforcement learning process.

3.4　計算機実験

提案モデルの能力を検証するために、二種類の計算機実験を行った。一方は cart-pole balancing problem と呼ばれるもので、状態空間が離散化されているために部分観測性が生じているものの、マルコフ決定過程として記述できるものである。この環境においては、複合ネットワーク形成処理による POMDP への対応が不要であるため、この学習処理を省略し、単純ネットワーク形成処理、出力確率修正処理、抑制結合修正処理、内部強化信号修正処理の４つのみを使用して実験を行った。

もう一方の実験環境は強い部分観測性をもつ環境であり、これを対戦エージェント環境と呼ぶことにする。この対戦エージェント環境を用い、複合ネットワーク形成処理の有無による POMDP での性能の変化を検証した。

実験で用いたパラメータを表 3.1 に示す。パラメータ ϕ、γ、k_{ref}、k_{neg}、θ_n^A、θ_n^B については、これらの比のみがネットワークの動作に影響を与える。γ は ϕ よりも充分に低く設定する必要があり、かつ極端に微小であってはならないことから、表 3.1 のように設定した。一方、k_{ref} は ϕ よりも充分に大きくする必要があり、かつ極端に大きい値であってはならない。k_{neg} については、充分に微小な値である限り、学習の結果にはほとんど影響を及ぼさない。θ_n^A および θ_n^B の指数部については、実験環境の入力間隔に対応する値として経験的に定めた。なお、これらのパラメータについては、$\theta_n^A > \theta_n^B$ である必要がある。また、H1 層以外に対する d_n は微小な正値である必要があり、H2

表 3.1 実験パラメータ

Table 3.1　Simulation parameters.

内部状態減衰率 (H1 層)	d_n	0.20
内部状態減衰率 (H1 層以外)	d_n	0.02
発火閾値	ϕ	0.5
結合荷重マージン	γ	0.05
不応性強度	k_{ref}	1.5
層間ディレイ	k_{d}	2
抑制結合強化量	k_{neg}	0.01
発火時期判定閾値 (1)	θ_n^A	$k_{\mathrm{ref}} \cdot (1-d_n)^{12}$
発火時期判定閾値 (2)	θ_n^B	$k_{\mathrm{ref}} \cdot (1-d_n)^{96}$
出力確率減衰率	d_{prob}	0.05
内部強化信号減衰率	d_{pseudo}	0.05
複合ネットワーク形成処理実行間隔	ρ	10000(ステップ)

表 3.2 ϵ と発火回数

Table 3.2　ϵ and neuron's age.

ニューロンの発火回数	ϵ
≤ 5	1.00
≤ 10	0.50
≤ 15	0.35
≤ 20	0.25
≤ 33	0.15
≤ 50	0.10
≤ 100	0.05
それ以外	0.01

層に対する d_n は H1 層に対するそれよりも充分に大きい必要がある。

H2 層からの出力選択パラメータ ϵ は、表 3.2 に従い、H2 層ニューロンの過去の発火回数に応じて変更した。これにより、学習が進みニューロンが発火を重ねるにつれて、出力は決定論的に定まるようになっていく。出力が極端に多い問題に対しては ϵ をより緩やかに下降させることが望ましいが、一般には、ϵ を環境に特化して設定する必要はない。

3.4.1 Cart-pole balancing problem

Cart-pole balancing problem の模式図を図 3.9 に示す。提案モデルの出力は、台車に加える力 F の方向であり、台車の移動方向に対して押す・引く・力を加えない、の三種類の出力を持つ。台車にはポールが据えつけられており、このポールは台車の移動方向と平行な向きにのみ傾く。うまく力を加えることで、台車が一定の範囲を越えないままで、ポールが傾きすぎないように維持し続けるのがこの問題での目的である。

提案モデルに対する入力は、ポールの角度 θ・ポールの角速度 $\dot{\theta}$・台車の位置 x・台車の速度 $\dot{x}$ であり、それぞれが離散化された形で入力される。もし θ か x が一定の閾値を越えた場合には、提案モデルに対して負の強化信号 -1 が与えられ、台車とポールは初期位置にリセットされる。

初期位置は、$\dot{\theta} = 0$、$x = 0$、$\dot{x} = 0$ であるが、θ については、位置がリセットされるたびにゼロでない微小な値がランダムに設定される。この値は、絶対値にして 4.0×10^{-7} 〜8.0×10^{-7} の値が均等な確率で選択され、等確率で正負が決定される。計算上の時間幅は 0.02(ms) であり、この 0.02(ms) は提案モデルの 10 ステップに相当する。

θ を離散化する際の閾値は ± 1 および ± 12(度) であり、これにより角度情報は 5 状態に分割される。同様に、$\dot{\theta}$ は閾値 ± 50(度/sec) により 3 状態に、x は ± 0.8(m) により 3 状態に、$\dot{x}$ は ± 0.5(m/s) により 3 状態に分割される。従って、提案モデルに対する入力は 14 であり、組み合わせによる状態数は 135 となる。

このような設定下で、複合ネットワーク形成処理を省いた状態で 20 回の試行を行った。試行はそれぞれ初期状態のネットワークで開始し、別個に一定回数の学習を行って、最終的に全試行の平均値を求める。この時、初期位置に置かれた台車が倒れるまでを一回の学習と数えることとし、学習の途中で継続時間が 240,000 ステップに達した場合には、そこでその回の学習を打ち切るものとした。

図 3.10 は、試行ごとに連続した 10 回の学習での平均継続時間をとり、それを 20 回の試行について平均したものをプロットした結果である。比較対象として、Q-Learning [45] による学習結果もプロットした。Q-Learning での出力の決定方法としては以下の式を用いた。

$$P(a|x) = \frac{\exp(Q(x,a)/T)}{\sum_{b \in \text{actions}} \exp(Q(x,b)/T)} \tag{3.28}$$

$P(a|x)$ は状態 x において出力 a を選択する確率であり、$Q(x,a)$ は状態 x における出力 a の Q 値である。温度パラメータ T は 0.005 とした。

同様に、20 回の試行についての中央値をプロットしたのが図 3.11 である。Q-Learning において、平均値をプロットしてものに比べ学習の立ち上がりが非常に遅いのは、試行ごとに学習の進みが大きく異なっているからである。

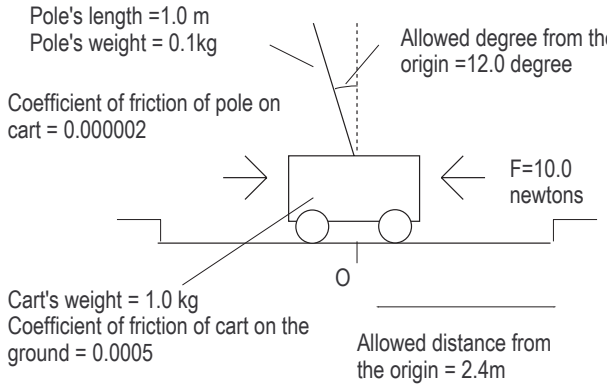


図 3.9 Cart-pole balancing problem

Fig. 3.9　Cart-pole balancing problem.

さらに、出力のたびに台車に加わる力に±５％の誤差が加わるように設定して実験を行った。この結果が図 3.12 である。図より明らかなように、このような環境では Q-Learning においては学習がある程度以上には進行しないのに対し、提案法では、誤差のない環境に比べれば結果は劣るものの学習が可能である。

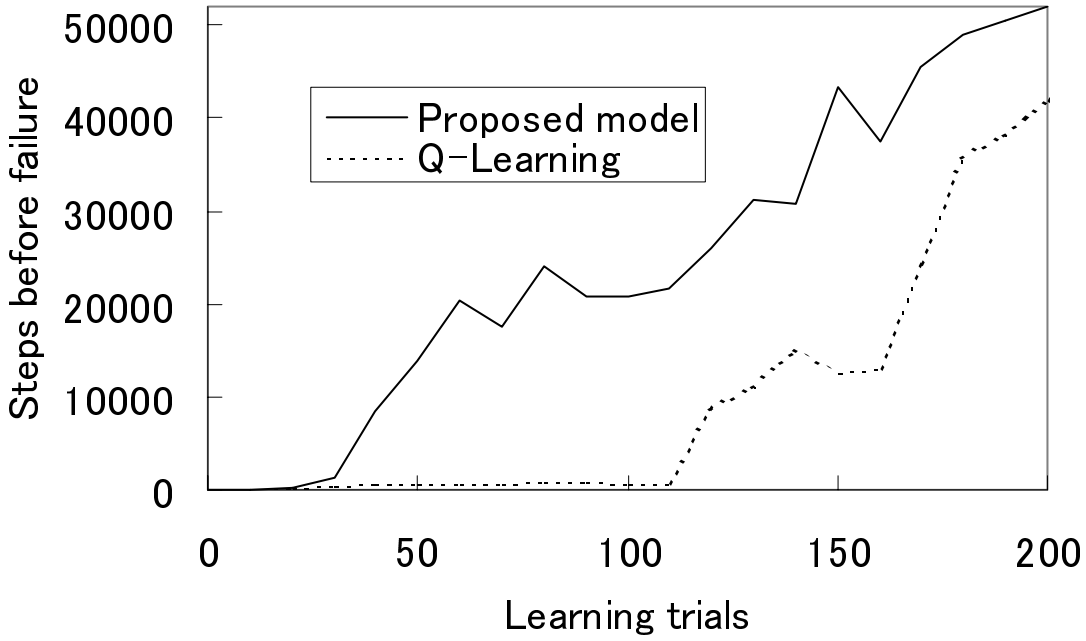


図 3.10 Cart-pole balancing problem の実験結果 (平均値)
Fig. 3.10　Results of cart-pole balancing problem (average).

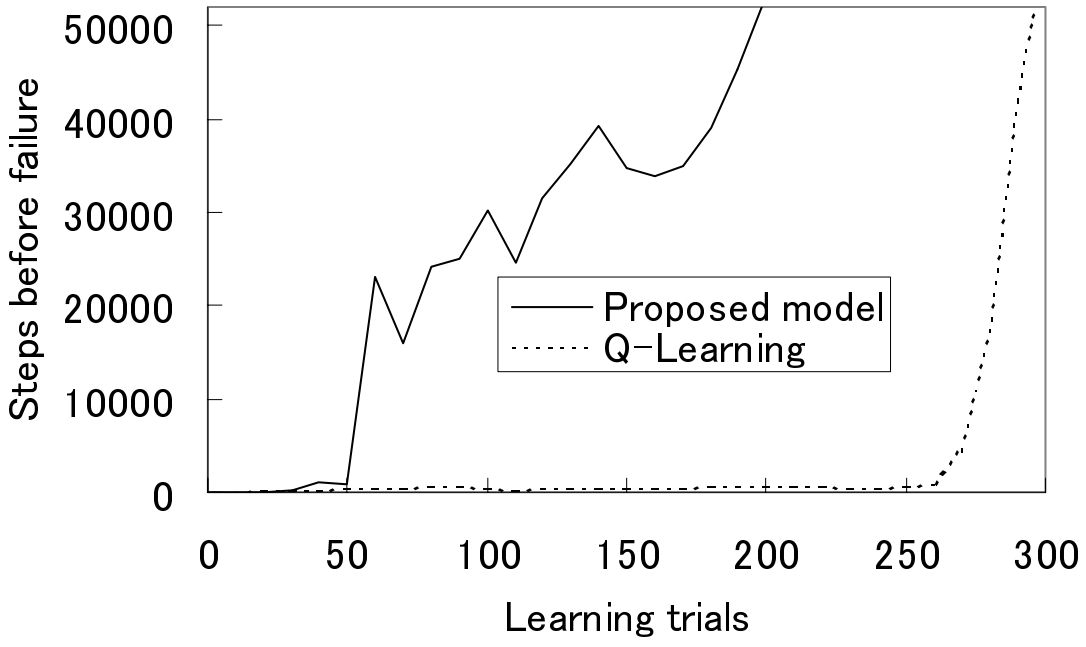


図 3.11 Cart-pole balancing problem の実験結果 (中央値)
Fig. 3.11　Results of cart-pole balancing problem (median).

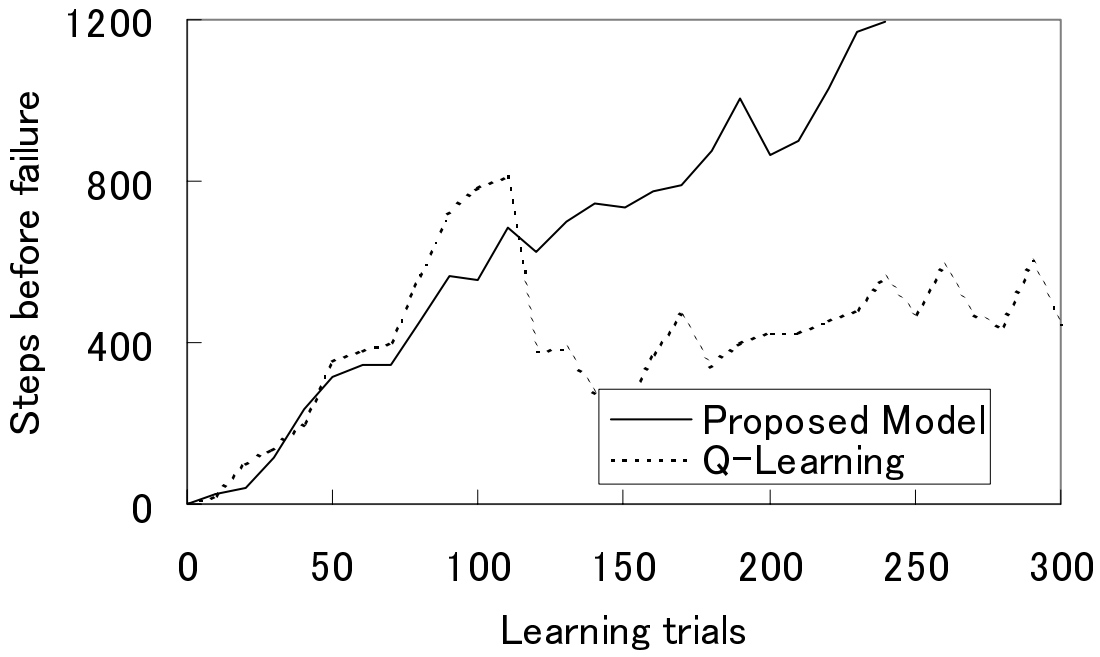


図 3.12 出力に誤差のある環境での結果 (中央値)

Fig. 3.12　Results on the environment with noisy output (median).

3.4.2　対戦エージェント環境

対戦エージェント環境は、図 3.13 に示されるように、ロボットエージェントによる射撃戦を模したものである。片方のエージェント (自エージェント) は提案モデルによる制御を行い、もう一方のエージェント (敵エージェント) はあらかじめプログラムされた制御を行う。座標や方位は連続値としてシミュレートされるが、センサー精度の限界などを想定し、自エージェントに対しては後述のような離散化された情報が入力されるものとした。

出力も右旋回・左旋回・前進・射撃・右移動・左移動という離散化されたものである。たとえば射撃という出力が発生した場合には、移動を停止して 12 単位時間にわたり武器を準備し、4 単位時間のあいだ銃撃を行い、その後 24 単位時間は静止する。この一連の行動の間、他の出力は一切無視される。前進という出力の場合には、6 単位時間にわたり一定速度で前進するが、他の出力が与えられれば即座に前進を停止する。

エージェントの銃撃中に、相手のエージェントが攻撃範囲内に存在する場合には、そのエージェントは破壊されたものとし、双方を初期位置にリセットした。このとき、自エージェントが破壊された場合には負の強化信号 -1 が、逆の場合には正の強化信号 $+5$ が与えられる。エージェントの初期位置は、破壊されるたびに均等な確率でランダムに再設定されるが、その際には次の 3 つの制約に縛られる。まず、エージェント間の距離は一定の範囲内であり、また両方のエージェントがお互いを視界に納めており、かつお互いに相手のエージェントの攻撃範囲に入ってはいない。

自エージェントが受け取る入力情報は以下のような、それぞれ離散化されたものである。敵エージェントの存在する方向 (正面・右前面・左前面・右側面・左側面・視界外の 6 種類)、敵エージェントとの距離 (自分の攻撃射程内・射程外の 2 種類)、敵エージェントの自分に対する向き (正面・右面・左面・それ以外の 4 種類) である。これに対し、敵エージェントは双方の正確な座標を入力として受け取る。

敵エージェントは以下のようなアルゴリズムに基づいて行動する。まず自エージェントが攻撃角度内にいなければ、旋回行動により正面におさめる。そして、自エージェントが攻撃射程内にいなければ、前進行動により射程におさめる。攻撃角度と攻撃射程の両方を満たしていれば、射撃行動により攻撃を行う。

自エージェントは、敵エージェントと同じ攻撃角度を持つものの、射程においては劣っているため、単純な行動では勝利が非常に困難な設定となっている。

この環境の特徴には二つが挙げられる。第一は、入力の部分観測性である。連続値である座標や方位は、自分と敵との相対関係をもって離散化され入力されている。また敵の現在の行動 (射撃準備中・旋回中など) は与えられておらず、非常に部分観測性の強い環境であるといえる。第二は、出力と状態変化の間の時間遅れである。ある観

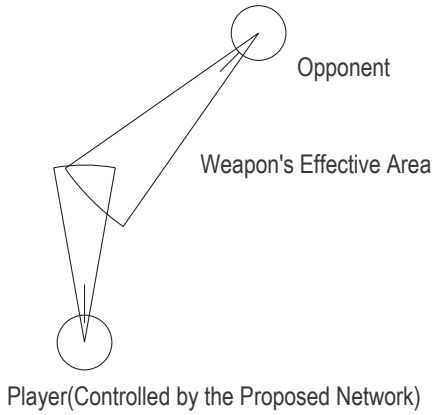


図 3.13 対戦エージェント環境
Fig. 3.13　Competitive Agent Environment.

測状態において出力された行動が観測状態の変化をもたらすまでには時間遅れがあり、さらに敵の (観測不能な) 行動による影響も受けるため状態変化の可能性は極めて多様である。

このような前提の下で、提案モデルの学習則のうち、複合ネットワーク形成処理を用いた場合と、用いない場合とで比較シミュレーションを行った。連続した 500 回の勝敗の平均をとり、その 20 試行での平均勝率をプロットしたのが図 3.14 である。同様に、全試行での最悪値を図 3.15、最良値を図 3.16、中央値を図 3.17 に示した。

複合ネットワーク形成処理を省いた場合には、隠れ層は減衰率の高いニューロンのみで構成されることになるため、直近の入力のみに基づいて動作することになる。もちろんこの場合にも H2 層のニューロンは生成されるが、常に一つの H1 層ニューロンからのみ入力を受け取るため、実用上リレーとしての働きしか持たない。

これに対し、複合ネットワーク形成処理を導入した場合には、H2 層ニューロンが複数の入力を受け取り、過去の入力情報も加味した上で出力が決定されることになる。図から判るように、これにより部分観測マルコフ過程での学習性能が大きく向上している。

また、複合ネットワーク形成処理を用いる設定の下で、H2 層の内部状態減衰率 d_n を変更して比較を行った場合の平均勝率を図 3.18 に、H1 層の内部状態減衰率を変更して比較を行った場合の平均勝率を図 3.19 に示した。図 3.18 より、H2 層の減衰率が 0.02、0.05、0.01 の場合については、減衰率の変化量に対する学習性能の変化量は過大なものではなく、減衰率の最適化を行わなくとも提案システムの運用は可能であるといえる。減衰率の大小により、提案ネットワークが入力情報を処理することのできる時間幅や、関係のない入力列から受けるノイズが異なるため、学習性能に若干の差異が生じているものと思われる。

一方、H2 層の減衰率が 0.1 および 0.2 の場合については学習結果が大きく悪化しており、特に減衰率 0.2 では、複合ネットワーク形成処理を導入しなかった場合と同程度の結果となった。これは、H2 層のニューロンが内部状態の形で蓄えるはずの過去の時系列情報が、減衰率の変更により、極めて短い時間で失われるようになってしまったためと推測される。つまり、複合ネットワーク形成処理を省いた場合と同様に、過去の入力情報を利用できなくなったために、部分観測状態の識別ができなくなり、それが学習性能の極端な低下を招いたと考えられる。

また、図 3.19 より、H1 層の減衰率を H2 層と同一にした場合にも、学習率が大きく悪化している。これは、減衰率が小さいニューロンだけでは、同時に受けた入力と時間差のある入力とを区別できないため、状態の混同が発生したものと考えられる。

これらの結果から、減衰率の異なるニューロンを多層に用いることによって、部分観測マルコフ過程での学習性能が大きく向上していると言える。

さらに、双方の射程をそのままに、自エージェントの攻撃角度を 2.5 倍にした場合の結果を図 3.20〜図 3.23 に示した。依然として自エージェントは射程において敵エージェントに劣っているため困難な問題ではあるが、学習の難易度は大きく低下している。そのため、図より明らかなように、複合ネットワーク形成処理の有無による性能差が大幅に減少していると考えられる。

自エージェントと敵エージェントの攻撃角度・攻撃射程を共に同じ値に設定した環境でもシミュレーションを行い、その結果を図 3.24 に示した。この設定においては、敵エージェントの前進に合わせて自エージェントが静止して射撃を行うという行動が最善のものとなる。このような行動をとる場合には、不完全知覚状態への適応は本質的に必要ではない。そのため、複合ネットワーク形成処理の有無に関らず、ほぼ同程度の勝率が得られているものと推測される。

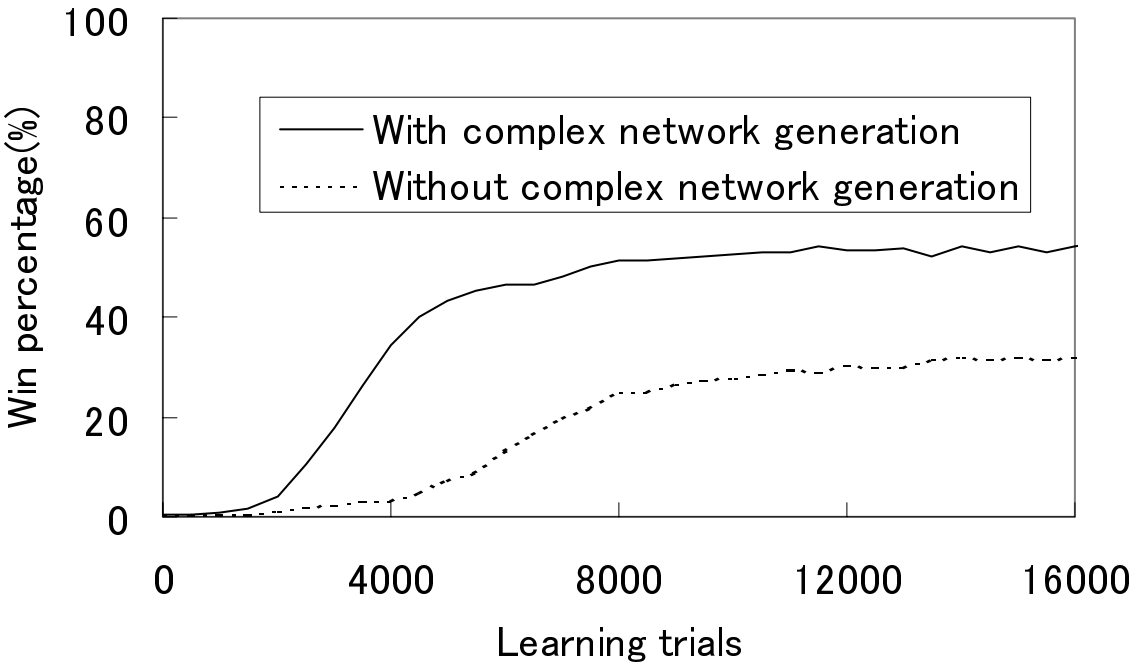


図 3.14 対戦エージェント環境の実験結果 (平均値)
Fig. 3.14　Results of competitive learning environment (average).

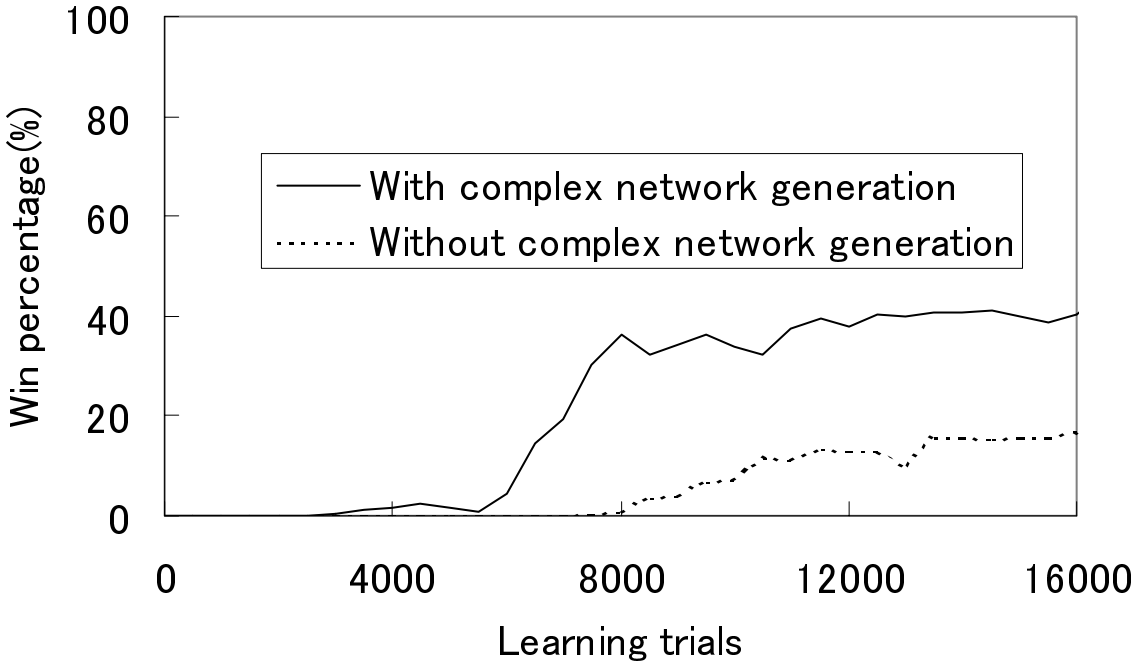


図 3.15 対戦エージェント環境の実験結果 (最悪値)
Fig. 3.15　Results of competitive learning environment (worst case).

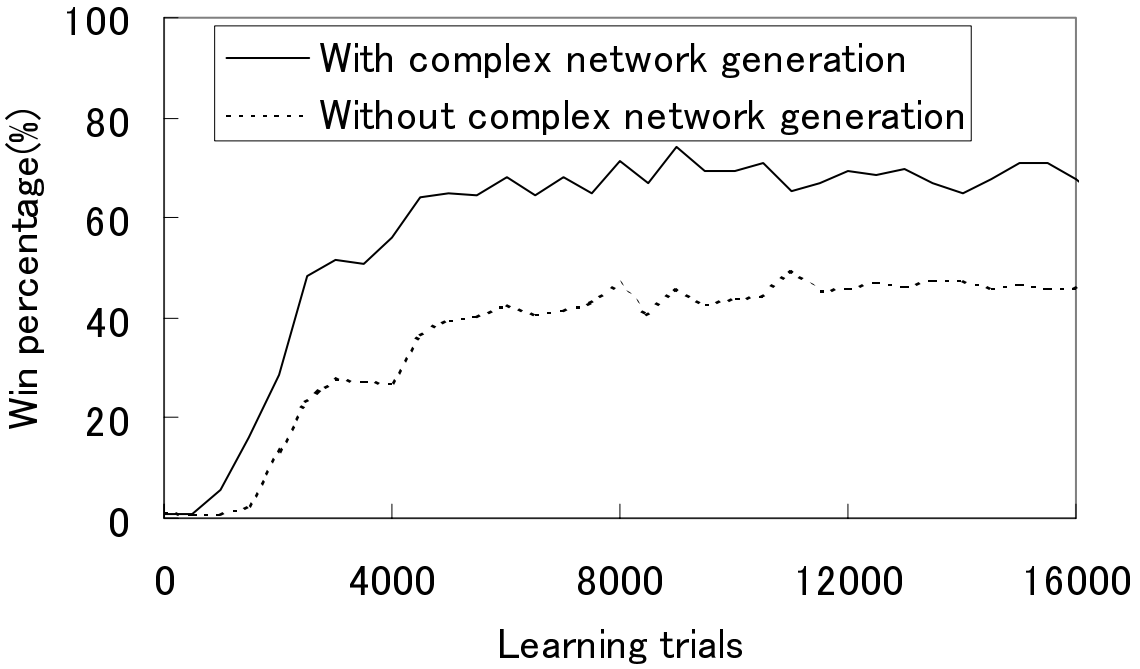


図 3.16 対戦エージェント環境の実験結果 (最良値)
Fig. 3.16　Results of competitive learning environment (best case).

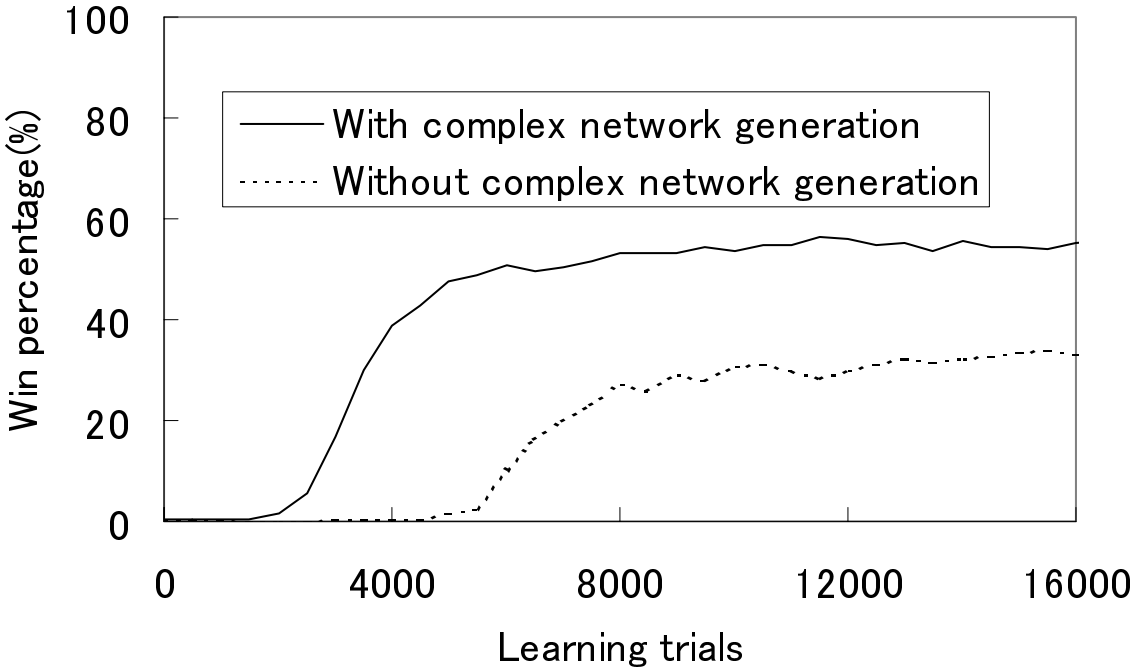


図 3.17 対戦エージェント環境の実験結果 (中央値)
Fig. 3.17　Results of competitive learning environment (median).

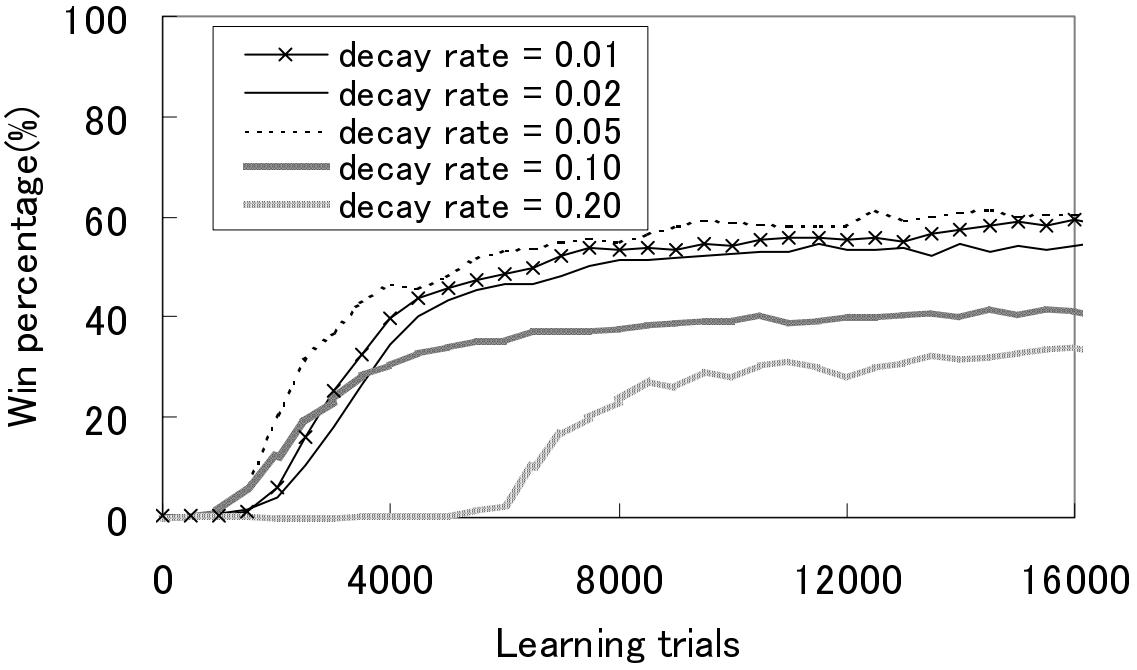


図 3.18 H2 層の減衰率を変えた場合の学習結果 (平均値)

Fig. 3.18　Results with different decay rates on H2 layer (average).

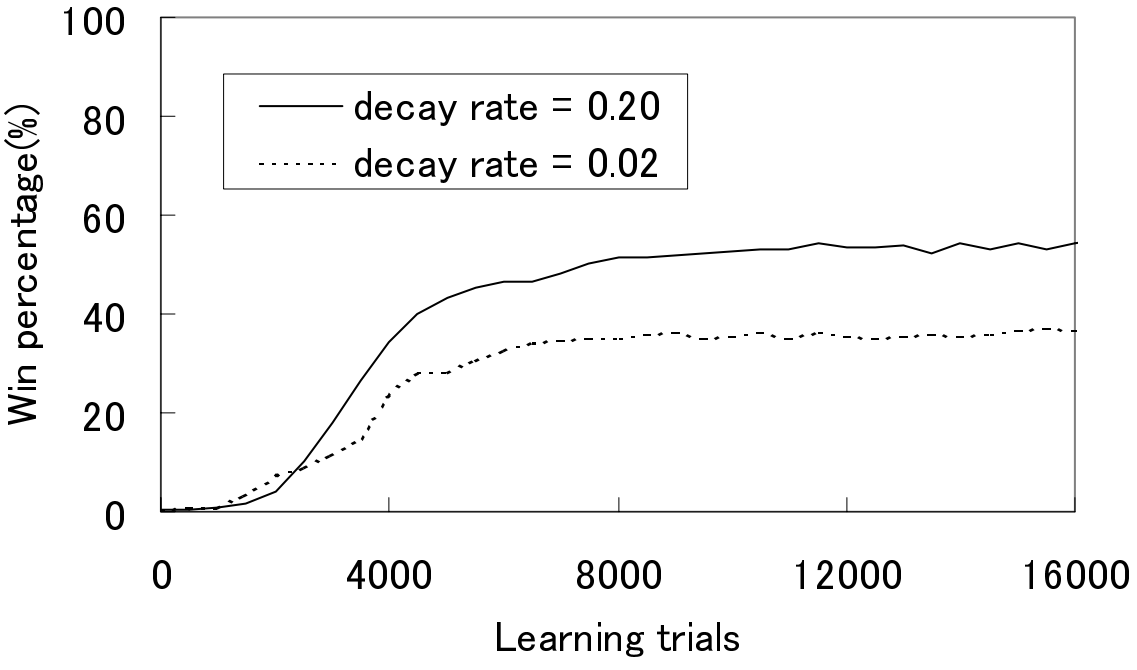


図 3.19 H1 層の減衰率を変えた場合の学習結果 (平均値)

Fig. 3.19　Results with different decay rates on H1 layer (average).

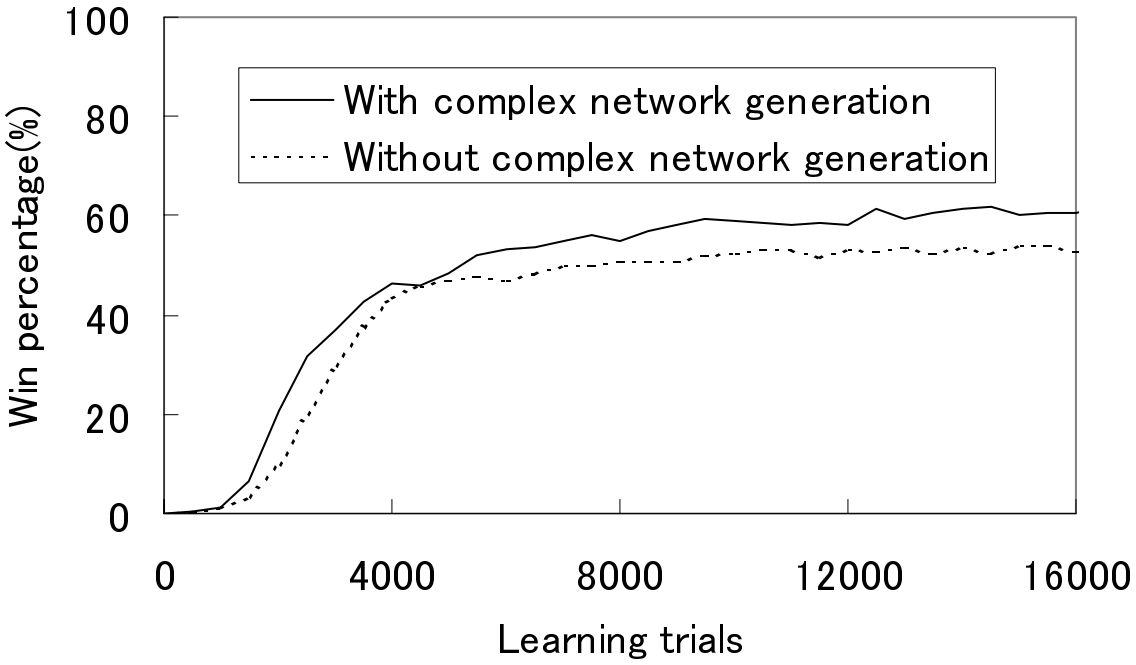


図 3.20 自エージェントの攻撃角度を広げた場合の結果 (平均値)

Fig. 3.20　Results with wider attack arc (average).

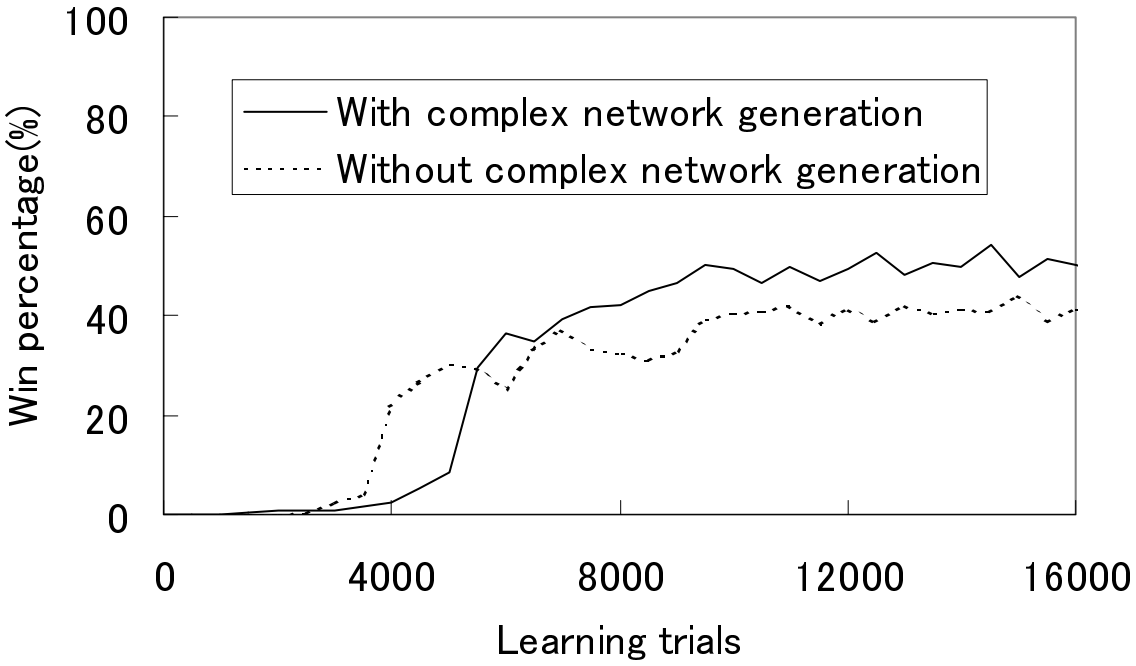


図 3.21 自エージェントの攻撃角度を広げた場合の結果 (最悪値)

Fig. 3.21　Results with wider attack arc (worst).

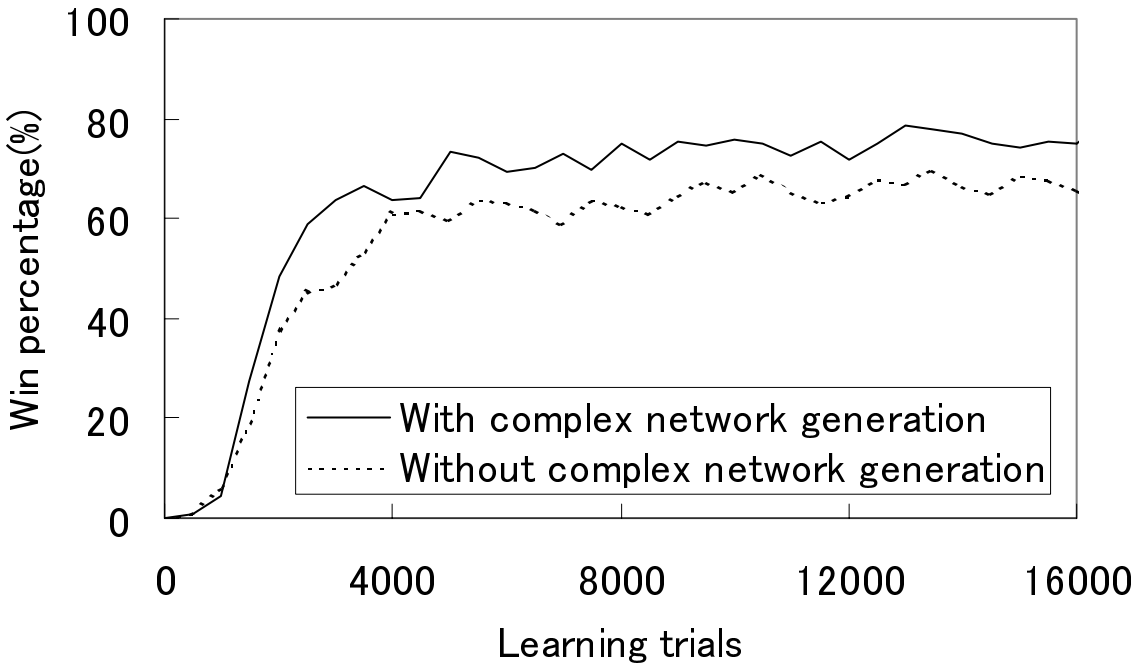


図 3.22 自エージェントの攻撃角度を広げた場合の結果 (最良値)

Fig. 3.22　Results with wider attack arc (best).

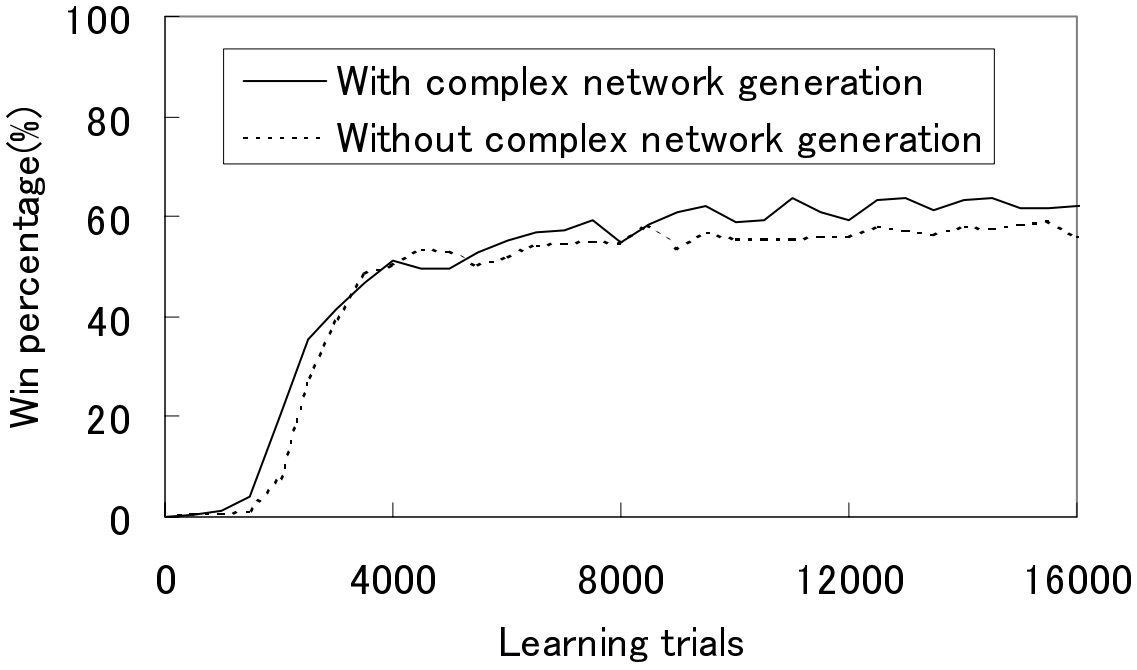


図 3.23 自エージェントの攻撃角度を広げた場合の結果 (中央値)

Fig. 3.23　Results with wider attack arc (median).

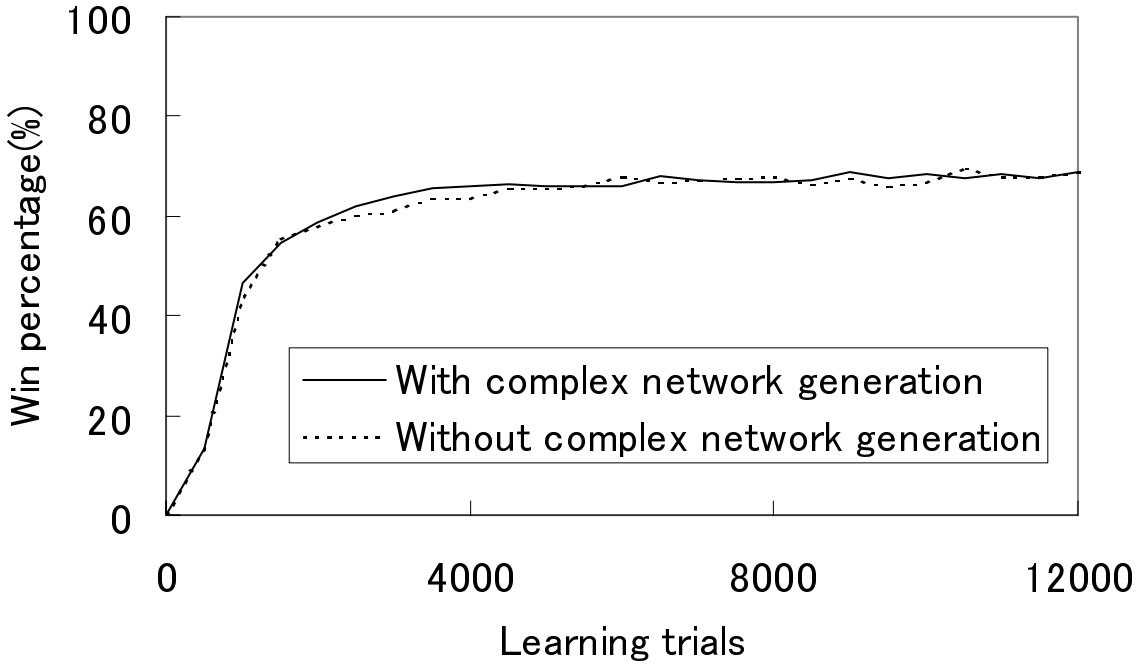


図 3.24 両エージェントの攻撃範囲を同じに設定した場合の結果 (平均値)

Fig. 3.24　Results with the same attack range on both agents (average).

3.5　むすび

本章では、パルスニューロン素子の時系列情報の処理能力に着目し、これを用いた新しい強化学習則を提案した。提案モデルは、内部状態の減衰率の異なる二種類のパルスニューロン素子を用い、部分観測マルコフ決定過程での学習を行うものである。計算機実験を行い、マルコフ決定過程および部分観測マルコフ決定過程の二種類の問題に対し、提案モデルが有効に働くことを検証した。今後の課題としては、現在経験的に定めているパラメータについて定量的な解析を行う事などが挙げられる。パラメータ敏感性は限定的であり、個々の環境に依存したパラメータの微調整が必要とされることはないと考えているが、特にH1層とH2層の内部状態の減衰率と、発火時期の判定閾値が与える影響については更なる解析が求められる。

第 4章

短期抑圧現象を取り入れたパルスニューラルネットワークによる注視制御

本章では、短期抑圧 (STSD) 現象を実装した人工パルスニューラルネットワークを用い、注視制御を行うモデルを提案する。パルスニューロン素子を用いた人工パルスニューラルネットワークは、従来の積分器型ニューロン素子を用いたものに比べ、時間情報を取り扱う能力に優れており、また、生理学的知見を取り入れることが容易であるという特長を持っている。本研究は、近年の生理学的知見の中でも特に STSD の現象に着目し、その工学的な利用を目的とするものである。STSD は生体の神経細胞のシナプスに見られる短期的な可塑性であり、ゲインコントロールを始めとする興味深い性質を持っているが、人工ニューラルネットワークへの応用は端緒についたばかりである。本章では、STSD が「慣れ」にあたる特徴を持っていることに着眼し、この現象を動画像における注視制御に応用した。計算機実験により、移動物体追跡・移動速度に基づく優先的注視・点滅物体の除外などの基本的な注視制御能力を確認している。

4.1 はじめに

生体の脳機能を電気回路ないしコンピュータで模倣し、その汎化能力や可塑性といった高度な情報処理能力を工学的に利用しようという考えは、広く研究者の関心を集めてきた。

特に工学的な応用が進んでいる人工神経細胞 (ニューロン) モデルとしては、個々の神経細胞を積分器としてモデル化した積分器型ニューロン素子 [1] が挙げられる。積分器型ニューロン素子を用いた人工ニューラルネットワークは、誤差逆伝搬法 [16] や自己組織化マップ [13] を始めとする優れたモデルが確立されるに伴い、様々な分野で用いられてきた。

一方、より実際の神経細胞に忠実に、integrate-and-fire 動作をするようにモデル化したパルスニューロン素子 [22], [28], [30], [32], [75] は、その挙動の解析が徐々に進んでいるものの、応用研究は端緒についたばかりであり、広く工学的に利用されるには至っていない。パルスニューロン素子の特徴としては、ネットワークをハードウェア的に実装することが容易であることから高速並列処理に適している点、また、積分器型のニューロンでは扱えない微小な時間単位での処理を行うことができ、時空間的な入出力の処理に適している点などが挙げられる。さらに、より生体の素子に近いモデルであることから、生理学的知見によって得られた動作を実装することが非常に容易である点などが挙げられる。

しかし、従来の研究におけるパルスニューロン素子への生理学的知見の実装は、多くがヘブの学習則 [6] のような基礎的なものにとどまっている。既存の積分器型ニューロン素子では実現することができなかったような優れた情報処理の実現のためには、多様な生理学的現象の実装と、その工学的な応用の模索が必要とされている。

生理学的な見地からは、生体の神経細胞に見られる様々な現象の中で、近年特に大脳皮質における短期抑圧 (short term synaptic depression, STSD) の現象 [76], [77] が注目を集めている。これは、ヘブの学習則に基づいて、人工ニューラルネットワークの研究において伝統的に用いられてきた長期増強 (long term potentiation, LTP) の現象とは異なり、ごく短い期間のみ持続する可塑性である。具体的には、あるシナプスを通じて継続的に入力パルスが送られた場合に、その周波数に反比例するように、一時的に結合荷重が減少 (興奮性の結合の場合) ないし増加 (抑制性の結合の場合) するというものである。結果として、ニューロンがあるシナプスから受ける入力の総和は、長期的には入力の周波数によらず一定になる。

STSD 現象が神経細胞において果たしている役割は多数考えられるが、その中の一つとして複数の入力間のゲインコントロールをしているという説が注目を集めており [78]、主に生理学的な見地からの研究が進んでいる。これに対し我々は、この現象

を人工パルスニューラルネットワークに実装し、その工学的な応用を模索するという観点から研究を行っている。特に、STSD 現象の特徴のうち、周波数が急激に上昇した直後には入力は抑圧されず、時間が経つに従って抑圧が進んでいくという点は、新規に与えられた刺激と古い刺激とを識別する「慣れ」に相当するものであり、選択的注意を行うためのシステムへの応用が考えられる。

以上のような背景から、本章では、STSD 現象を利用して注視制御 (visual attention control) を行う手法を提案する。注視制御という言葉は様々な分野で用いられるが、ここでは、画像の中の重要な部分のみに時間をかけて高度な画像処理を行うための前処理として、画像全体の粗い情報から重要な部分を高速にピックアップするというタスクを扱う [79]〜[81]。このような観点からの注視制御は、動的な環境において動作するロボットエージェントを設計する上で、膨大な入力情報の中から必要な情報を選定するために極めて重要なものであり、注目を集めている。また一方で、仮想環境におけるエージェントでの注視制御も近年研究が進んでいる。この場合には、効率的な入力情報の処理と意志決定だけでなく、リアリティのあるエージェントの描写という観点からも重要である [82], [83]。

提案するモデルは、入力として実時間に更新される低解像度・多階調の白黒画像を受け取り、STSD 現象を導入した 4 層のパルスニューラルネットワークによって、画像中の注視するべき領域を、簡易ながら高速に選定し出力するというものである。注視すべき領域としては、新しく視界に入ってきた物体や、高速に移動している物体などが選択される。この処理は基本的に輝度とその変化量に基づいて行われるが、低速に移動してる物体や、輝度がゆっくりと変化している領域、あるいは単純に点滅しているだけの領域などは、例え輝度やその変化量が大きかったとしても、注視対象としては非常に選択されにくい。この動作は、パルスニューロン素子における STSD 現象の特徴と、ネットワーク構造の組み合わせとによって実現されている。また、このモデルは並列処理が容易な構造をしており、将来的には、個々のパルスニューロンをハードウェア実装することによって、極めて高速な処理が期待できる。

急に視界に現れた物体や、高速に移動している物体は、すぐに環境やエージェント自身に影響を及ぼす可能性があり、一般に、物体認識や軌道の推測などを行うべき優先度が高い。このような注視制御は、動的な環境におけるロボットエージェントへの要求と合致するだけでなく、実際の生物においても見られる現象である [84], [85]。一方、輝度が高かったり色が奇抜だったりしても、変化していない物体、あるいは単純に点滅しているだけの物体などは、注視対象としての優先度は低いといえる。

計算機シミュレーションにより、提案モデルが、簡易ではあるがロボットビジョンへの応用に一定の有効性をもつ注視制御を行えることを確認した。これは、パルスニュー

ロン素子における STSD 現象の応用としては全く新規のものである。

以下、**4.2** では STSD 現象を導入したパルスニューロン素子のモデルについて述べ、**4.3** では、この素子を用いて動画像処理を行うパルスニューラルネットワークを提案する。**4.4** では、提案モデルの有効性を検証するために行った計算機シミュレーションについて述べる。最後に、**4.5** において結論と考察を述べる。

4.2 短期抑圧を取り入れたパルスニューロン素子

本節では、STSD 現象を導入したパルス駆動型ニューロン素子について述べる。この素子についてはソフトウェアシミュレーションの効率を優先して設計を行ったが、同じような挙動を示すものであれば、ハードウェア実装の効率を優先した素子を提案ネットワークに用いることも可能である。

このニューロン素子は、実際の神経細胞に見られる主な現象、具体的には、信号の時間的加算と減衰、不応性、結合の短期抑圧などを取り入れたものであり、入出力としてパルス列を扱う。次節で説明する提案ネットワークは、全てこのニューロン素子で構成されている。

以下、このモデルのニューロン単位での挙動と、ニューロン間結合の挙動について、分けて解説する。

4.2.1 ニューロンの挙動

このパルス駆動型ニューロンモデルでは，あるニューロン N_i が別のニューロン N_j にパルスを出力すると、ニューロン N_j の内部状態 (内部電位)v_j は、時間遅れ T_{ij} の後に上昇を始め、ピークに達した後に弧を描くように徐々に静止電位まで減衰していく。このとき、入力パルスが内部状態 v_j に与える影響はニューロン間の結合荷重 w_{ij} に比例する。なお、一般的には $w_{ij} \neq w_{ji}$ である。

もし内部電位 v_j が発火閾値 θ_f を越えるとニューロンは発火し，結合しているニューロン全てに対して出力パルスが送られる。発火したニューロンの内部電位は負の定数値である不応期電位 R にセットされ、時間を経ることで徐々に静止電位まで戻っていく。ここで、内部電位が絶対不応期閾値 θ_a よりも低い間は、このニューロンは入力パルスの影響を一切受けないものとする。この、v_j が θ_a よりも低い期間は絶対不応期に、v_j が θ_a よりは高いが静止電位よりも低い期間は相対不応期に相当する。

離散時刻を $t = 0, 1, 2, \ldots$ としたとき、ニューロン N_j の時刻 t における出力 $O_j(t)$ を，次式で定義する。

$$O_j(t) = \begin{cases} 1, & v_j(t) \geq \theta_f \\ 0, & v_j(t) < \theta_f \end{cases} \tag{4.1}$$

次に、時刻 t_0 にニューロン N_i から入力されたパルスの、ニューロン N_j の内部電位に対する影響 E_{ij} を以下のように設定する。

$$E_{ij}(t) = \beta_p^{t-t_0} \cdot \gamma_p \cdot w_{ij}(t_0) - \beta_n^{t-t_0} \cdot \gamma_p \cdot w_{ij}(t_0) \tag{4.2}$$

ここで、β_p および β_n は減衰を規定する定数値であり、$0 < \beta_n < \beta_p < 1$ である。また、γ_p は入力パルスのスケーリングのための定数値であり、$\gamma_p > 0$ である。E_{ij} は、上で述べたような弧状の曲線を描く。

式 (4.2) を元に、ニューロン N_j の時刻 $t+1$ における内部状態 $v_j(t+1)$ を、以下のように定義する。式 (4.2) の右辺第一項が $A_j(t)$ に、第二項が $B_j(t)$ に相当する。

$$v_j(t+1) = A_j(t) + B_j(t) + C_j(t) \tag{4.3}$$

$$I_j(t+1) = \sum_i \gamma_p \cdot w_{ij}(t) \cdot O_i(t - T_{ij}) \tag{4.4}$$

$$A_j(t+1) = \begin{cases} \beta_p \cdot A_j(t), & v_j(t) < \theta_a \\ \beta_p \cdot A_j(t) + I_j(t), & v_j(t) \geq \theta_a \end{cases} \tag{4.5}$$

$$B_j(t+1) = \begin{cases} \beta_n \cdot B_j(t), & v_j(t) < \theta_a \\ \beta_n \cdot B_j(t) - I_j(t), & v_j(t) \geq \theta_a \end{cases} \tag{4.6}$$

$$C_j(t+1) = \begin{cases} R, & O_j(t) = 1 \\ \beta_r \cdot C_j(t), & O_j(t) = 0 \end{cases} \tag{4.7}$$

ここで、$A_j(t)$ および $B_j(t)$ は内部電位に対する入力パルスからの寄与を、$C_j(t)$ は不応性からの寄与を示し、$A_j(0) = B_j(0) = C_j(0) = 0$ である。T_{ij} は、ニューロン N_i とニューロン N_j 間のパルス伝達の時間遅れである。$I_j(t)$ は、時刻 t におけるニューロン N_j に対する入力の総和を示す。R は、不応期電位を示す定数値であり、$R < 0$ である。また、β_r は不応性の減衰を規定する定数値であり、$0 < \beta_r < 1$ である。

4.2.2 ニューロン間結合の挙動

既に述べたように、このモデルでは結合荷重に対する短期抑圧を取り入れている。短期抑圧は、ある結合を通して継続的にパルスが送られた場合に、その結合の荷重を一時的に減少させるものである。結果として、結合荷重は、近似的にはパルスの頻度に反比例する。

このモデルでは、時刻 t における、ニューロン N_i から N_j への結合荷重 $w_{ij}(t)$ を、

$$w_{ij}(t) = \frac{W_{ij}}{\gamma_s \cdot D_{ij}(t) + 1} \tag{4.8}$$

$$D_{ij}(t+1) = \begin{cases} \beta_s \cdot D_{ij}(t) + 1, & O_i(t) = 1 \\ \beta_s \cdot D_{ij}(t), & O_i(t) = 0 \end{cases} \tag{4.9}$$

と定義する。ここで、W_{ij} は短期抑圧の影響を受けていない元の結合荷重である。γ_s は短期抑圧の影響のスケーリングを規定する定数であり、$\gamma_s > 0$ である。また、$D_{ij}(t)$ は N_i から N_j への結合に残存している短期抑圧の影響の大きさを示し、β_s がその減衰を規定する定数となっている。ここで、$0 < \beta_s < 1$ である。

4.3 短期抑圧を取り入れたパルスニューラルネットワークによる注視制御

本節では、本章の提案であるパルスニューラルネットワークについて述べる。まずネットワーク構造の概略について述べ、次にネットワークの各部位が注視制御に果たしている役割について説明する。

4.3.1 ネットワークの概略

図 4.1 に、提案ネットワークの入出力の流れを示す。このネットワークは四層構造をしており、それぞれの層を入力層、第一隠れ層、第二隠れ層、出力層と呼ぶこととする。基本的には入力層から出力層までのフィードフォワード構造となっているが、出力層内部には相互の結合が存在する。入力としては、ネットワーク動作の単位時間 (以下ステップ) ごとに、256 階調の白黒動画像を受け取る。出力は注視領域であり、ステップごとに、格子状に分割された画像の領域のうちの一つが選択されるか、あるいは無出力となる。無出力の場合には、注視すべき領域は最近に選択されたものから変化していないと考える。

入力層では、入力動画像のピクセル数と同じ個数のニューロンが平面上に配置されている。出力層においてもニューロンは平面上に配置されているが、出力層ニューロンの個数は、注視領域として選択されうる領域の個数に等しい。例えば、出力として

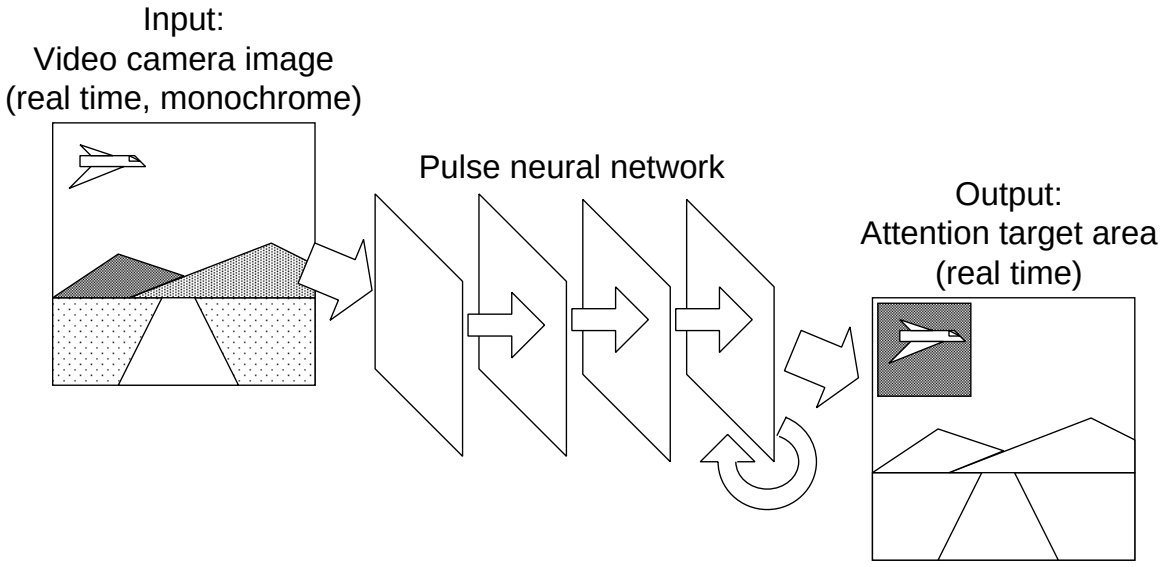
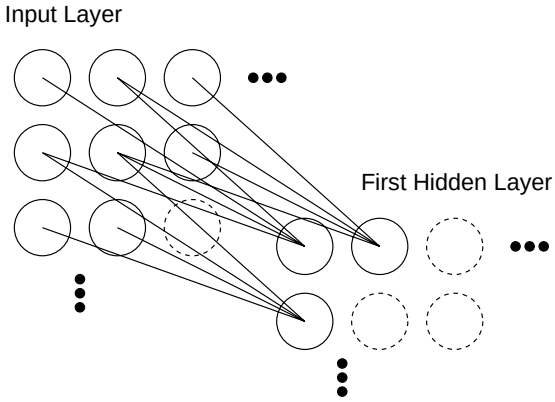
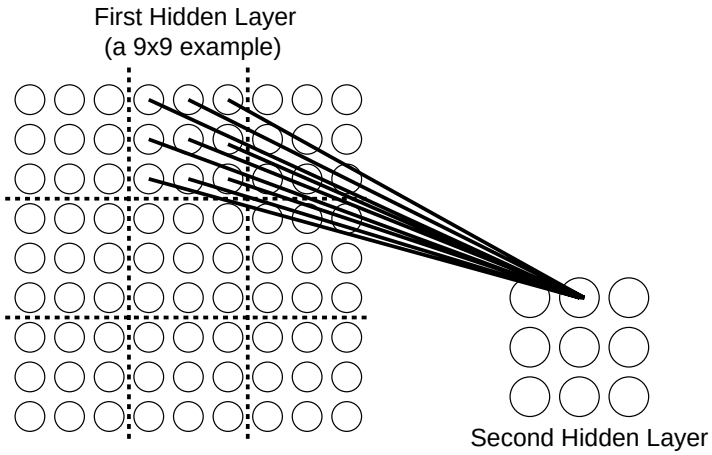


図 4.1 入出力の流れ
Fig. 4.1 I/O flow of the proposed model.



(a) Connections to the first hidden layer



(b) Connections to the second hidden layer

図 4.2 層間結合の様子

Fig. 4.2 Inter-layer connections.

3×3 の格子状に分割された画像の領域のうちの一つが選択されるように設定した場合、出力層ニューロンも 3×3 の格子状に配置される。出力層ニューロンが発火すると、対応する位置にある領域が注視領域として選択される。

入力層においては、それぞれのニューロンは、入力動画像中の対応するピクセルの輝度に応じた頻度で発火するものとする。ある入力層ニューロン N_i がステップあたりに発火する回数 $h_i(t)$ は、時刻 t において、対応するピクセル (x, y) の輝度を $b_{xy}(t)$ とすると、

$$h_i(t) = \gamma_b \cdot b_{xy}(t) + \theta_b \tag{4.10}$$

で定義される。ここで、γ_b および θ_b は、$\gamma_b > 0$、$\theta_b \geq 0$ を満たす定数値である。

第一隠れ層でも、同じようにニューロンが平面上に配置されている。ここで、それぞれの第一隠れ層ニューロンは、入力層のニューロンのうち、2×2 の計 4 ニューロンから正の荷重で結合されている。隣り合った第一隠れ層ニューロン同士は、図 4.2(a) に示すように、半分ずつ重なりあった領域から入力を受ける。

なお、入力層の任意のニューロン N_i から第一隠れ層の任意のニューロン N_j への結合荷重 W_{ij} は、$W_{ij} = \alpha_{i-h} > 0$ で定義され、一定である。ここで、α_{i-h} は入力層 (Input layer)-隠れ層 (Hidden layer) 間の結合荷重を示す定数値である。以下、同様に、α_{h-h} は隠れ層-隠れ層間の結合荷重を、α_{h-o} は隠れ層-出力層 (Output layer) 間の結合荷重を示す定数値である。

第二隠れ層では、出力層と全く同じ構成でパルスニューロンが平面上に配置されている。例えば、出力層ニューロンが 3×3 で配置されている場合、第二隠れ層でもニューロンは 3×3 で配置される。

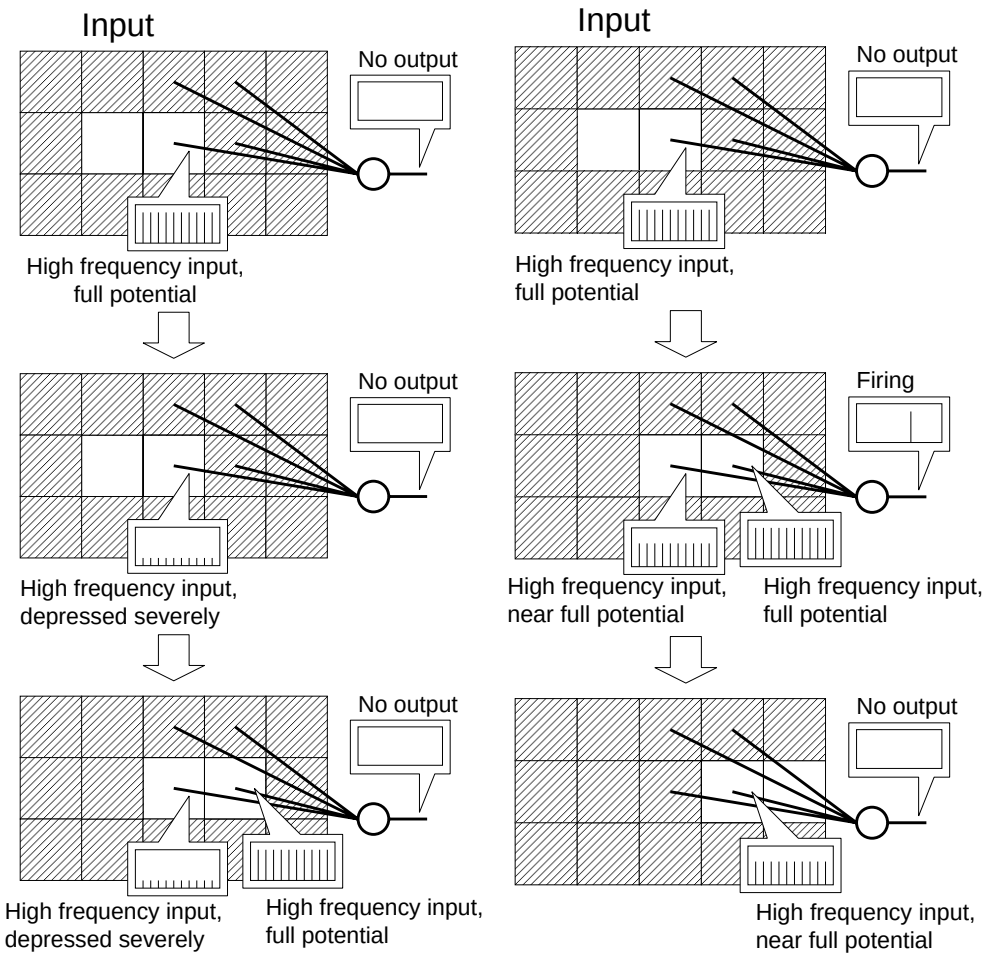
それぞれの第二隠れ層ニューロンは、図 4.2(b) に示されているように、対応する位置にある第一隠れ層ニューロン群から正の荷重で結合されている。仮に、第二隠れ層ニューロンが 3×3 に配置されている場合には、一つの第二隠れ層ニューロンは、第一隠れ層ニューロン全体の約 9 分の 1 から結合を受けることとなる。なお、複数の第二隠れ層ニューロンが同じ第一隠れ層ニューロンから入力を受け取ることはない。また、第一隠れ層の幅や高さが割り切れない場合には、第二隠れ層ニューロンのうち、中央にあるものが余計に結合をもつ。なお、第一隠れ層のニューロン N_j から第二隠れ層のニューロン N_k への結合荷重 W_{jk} は、N_j と N_k が対応する位置にある場合には $W_{jk} = \alpha_{h-h}$、それ以外の場合には $W_{jk} = 0$ で定義され、$\alpha_{h-h} > \theta_f$ である。

出力層においては、それぞれのニューロンは、対応する位置にあるただ一つの第二隠れ層ニューロンからのみ結合を受ける。出力層のニューロンが発火すると、それに対応する入力画像中の領域が注視領域として出力される。なお、第二隠れ層のニューロン N_k から出力層のニューロン N_l への結合荷重 W_{kl} は、$W_{kl} = \alpha_{h-o}$ で定義され、$\alpha_{h-o} > \theta_f$ である。

また、出力層ニューロンからは、出力層の他のニューロン全てに対して負の荷重で結合が伸びている。ここで、出力層ニューロン N_l から別の出力層ニューロン N_m への結合荷重 W_{lm} は、$l \neq m$ のとき $W_{lm} = \alpha_{o-o}$、$l = m$ のとき $W_{lm} = 0$ で定義され、$\alpha_{o-o} < 0$ である。

4.3.2 ネットワークの動作例

提案ネットワークにおいて、入力層-第一隠れ層間の結合は、STSD 現象を利用することによって、注視するべきではない領域を排除する働きをしている。具体的には、静



(a) A slow object (b) A fast object

図 4.3 移動物体検出の例

Fig. 4.3 An example of moving object detection.

止領域・ごくゆっくりと変化している領域・点滅している領域などに対応する第一隠れ層ニューロンは、STSD 現象によって結合荷重が減衰していくために、ある程度時間が経つと発火が抑制される。

また、物体が低速に移動している場合にも、図 4.3(a) に示すように、対応する位置にある第一隠れ層ニューロンに対する結合が STSD 現象により次々と減衰していくため、発火が抑制される。一方、物体が高速に移動している場合には減衰が進む前に多くの結合を通じて入力を受け取るため、図 4.3(b) に示すようにして発火が起こる。

第一隠れ層-第二隠れ層間の結合は、第一隠れ層ニューロンの発火をもとに、注視領域として出力する候補となる領域を決定する働きをしている。

出力層内部の相互結合は、複数の領域が注視領域の候補となっている場合に、その内

の刺激の弱い領域が選択されることを抑止する働きをしている。出力層の複数のニューロンが第二隠れ層から入力を受け取るような場合には、出力層内部での負の相互結合のために、より高頻度に入力を受け取るニューロンが優先的に発火することとなる。これによって、刺激の最も強い領域、例えば、より高速に移動する物体の存在する領域が、注視領域として選択される。

第二隠れ層-出力層間の結合そのものは、有意な働きをしていない。しかしながら、出力層をなくして第二隠れ層の内部に負の相互結合を作ってしまうと、次のような問題が発生する。複数の第一隠れ層ニューロンから同時に入力を受け取っている第二隠れ層ニューロンは、入力の頻度は少なくとも一度に受け取る入力の絶対量が大きいために、負の相互結合に抑止されないで発火してしまう。結果として、単に大きいだけの低速な物体が、高速な移動物体よりも優先して注視されることとなる。このような挙動は、ゆっくりと動く遠くの雲を注視する一方で飛んでくるナイフを無視するといったような結果を生む危険性が高く、提案法の目的とする注視制御には適さない。

このような問題を防ぐために、第二隠れ層の次の層としての出力層が必要であり、そこに負の相互結合が設けられている。出力層ニューロンに同時に入ってくる入力の絶対量は一定の範囲内にあるため、負の相互結合が有効に働き、入力画像中の物体の大きさではなく速度のほうが重要となる。

4.4 計算機実験

本節では、提案モデルの性能検証のために行った、計算機実験について述べる。まず全体に共通する実験環境の概要を説明した後、個々の実験の結果を順に示す。

4.4.1 実験環境

表 4.1 に、各実験で用いたパラメータ設定を示す。まず、パルスニューロンの個々の動作に関るパラメータについて説明する。パルス減衰係数・パルススケーリング係数については、単一の入力パルスの影響が頂点に達するまでの時間を 5 ステップ (単位時間) とし、この時の高さを 1.0、1000 ステップ後の高さを 0.01 とするように選んだ。なお、これらの設定値から減衰係数やスケーリング係数を導出する数式については割愛する。

発火閾値・不応期電位については、入力パルスの高さと結合荷重に対する相対的な値であり、ここではそれぞれ 1.0、-1.0 とした。絶対不応期閾値については、絶対不応期が 3 ステップ続くように設定した。また、不応性減衰係数はパルス減衰係数 (負値成分) と同じ値とした。

短期可塑性減衰係数・短期可塑性スケーリング係数については、入力パルスの頻度が数十 Hz の場合において、定常的には実効結合荷重がパルス頻度にほぼ反比例するように、また、入力パルスが停止すると、1000 ステップで D_{ij} が約十分の一に減衰するように設定した。

これらの設定下においては、仮に 1 ステップが約 1ms であると考えれば、個々のパルスニューロン素子の動作は時間スケール的には霊長類の脳神経細胞のそれに近いものとなる。

次に、本章の提案である、注視制御のためのネットワークに関わるパラメータについて説明する。層間時間遅れについては、本モデルでは特に重要ではないので最小限の値として 1 に設定してある。輝度-発火頻度変換のためのパラメータ γ_b および θ_b は、入力画像の輝度が 0 から 255 の範囲において、入力層ニューロンの発火の周波数が数十 Hz の範囲におさまるように設定した。

入力層-第一隠れ層間の結合荷重である α_{i-h} は、短期抑圧が充分に進んだ状況において、入力画像のうち変化していない領域に対応する第一隠れ層ニューロンが発火することがないように、また一方で、その内部状態が常に高く保たれるように、経験的に設定した。

他の層間の結合荷重 α_{h-h} および α_{h-o} は、発火閾値よりも僅かに大きい値として 1.005 に設定した。また、出力層-出力層間の結合荷重 α_{o-o} は経験的に設定した。

表 4.1 実験パラメータ
Table 4.1 Simulation parameters.

Intra-layer pulse delay	T_{ij}	1
Firing threshold	θ_f	1.000
Refractoriness	R	−1.000
Refractory period threshold	θ_a	−0.03707
Pulse decay (positive)	β_p	0.9954
Pulse decay (negative)	β_n	0.3334
Pulse scaling constant	γ_p	1.0280
Refractoriness decay	β_r	0.3334
STSD decay	β_s	0.9977
STSD scaling constant	γ_s	0.1800
Input frequency coefficient	γ_b	0.0002
Input frequency intercept	θ_b	0.02
I-H weight	α_{i-h}	0.0900
H-H weight	α_{h-h}	1.005
H-O weight	α_{h-o}	1.005
O-O weight	α_{o-o}	−0.5000

これらのパラメータのうち、環境に応じて調整を必要とするものは、主に短期可塑性スケーリング係数 (γ_s) と短期可塑性減衰係数 (β_s) である。一般に、γ_s を低く、β_s を若干高く設定した場合、短期抑圧の影響が現れるのが遅くなるため、低速な移動物体であっても注視の対象となる。逆に、γ_s を高く、β_s を若干低く設定した場合、高速な移動物体のみが注視されることとなる。

入力としては、白黒 256 階調、96 ピクセル× 96 ピクセルの低解像度の動画像を用いた。この動画像は毎ステップ更新される。出力としては、3 × 3 に区切られた画像領域のうちの一つが注視領域として選ばれるものとした。なお、より高解像度の動画像に対して本システムを適用する場合には、元画像に単純な縮小処理を施したものを入力とすることで、同様に注視領域の選定を行うことができる。

4.4.2 単一の移動物体が存在する環境における実験

単一の移動物体が存在する環境において行った実験について述べる。動画像の背景は変化することがなく、その輝度の分布は正規分布に基づくものとする。物体の輝度は一様であり、画像のＹ軸方向の中央を、画像の左端から右端へと、一定の速度で移動する。なお、物体のサイズは縦 16 ピクセル×横 16 ピクセルである。

物体の一部が画面に入ってから、物体が完全に画面から出るまでを一回のランと考え、20 回のランの間に物体を含む領域が注視されていた時間の割合を測定した。なお、定常状態からシミュレーションを始めるために、各ランの前にはネットワークを 2000

単位時間動作させた。

まず、物体と背景の輝度を固定して、物体の移動速度を変化させて実験を行った結果を表 4.2 に示す。物体の輝度は 255 とし、背景の輝度は各ランについて独立してピクセルごとに平均 128、分散 12.8^2 の正規分布に従って確率的に選択した。また、物体の速度はランごとに正規分布に従って確率的に選択した。表中では、1000 ステップあたりに移動するピクセル数で表記してある。

表 4.2 より明らかなように、広い速度範囲において移動物体を注視しつづけている一方で、物体が非常に低速、ないし静止している場合には無視するという、望ましい注視制御が提案モデルによって行われている。

また、物体の速度を一定として、背景と物体の輝度を変化させて実験を行った結果を表 4.3 に示す。物体の輝度はランごとに、背景の輝度は各ランについて独立してピクセルごとに、正規分布に従って確率的に選択した。ただし、輝度の下限および上限はそれぞれ 0、255 である。また、物体の移動速度は 1000 ステップにつき 10 ピクセルとした。この結果から、背景と物体の輝度の差がある程度大きければ、広い輝度範囲において正しく注視が行えることがわかった。

これらの結果から、提案モデルが基本的な移動物体への注視・追跡能力を持っていることが確認できた。

4.4.3 移動物体と点滅物体が存在する環境における実験

単一の移動物体に加え、点滅している物体が存在する環境において行った実験について述べる。動画像の背景は変化することがなく、その輝度は 0 とした。移動物体の輝度は 255 で一様であり、画像の Y 軸方向の中央を、画像の左端から右端へと一定の

表 4.2 移動物体が注視されていた時間割合 (変数：速度)

Table 4.2 Percentage of attention time on the moving object (single object, variable: velocity).

Velocity	Attention time (%)
$N(100, 10^2)$	86.63
$N(50, 5^2)$	94.48
$N(20, 2^2)$	97.96
$N(10, 1^2)$	97.78
$N(5, 0.5^2)$	98.84
$N(1, 0.1^2)$	90.57
$N(0.5, 0.05^2)$	20.00
$N(0.1, 0.01^2)$	5.00
0	0.00

速度で移動する。点滅している物体は静止しており、光っている状態 (輝度 255) と暗くなっている状態 (輝度 0) を交互に繰り返す。

移動物体および点滅物体のサイズは縦 16 ピクセル×横 16 ピクセルであり、点滅物体の位置は、画像の右上領域のちょうど中央に設定した。

各ランの開始時には、画面内には点滅物体のみが存在している。ランの開始から 1,000 ステップ後に移動物体が画面内に入り、それから 10,000 ステップかけて画面を横断する。移動物体が画面から出てから、9,000 ステップ後にランを終了するものとした。

このように、各ランの最初の 1,000 ステップおよび最後の 9,000 ステップには、画面内には点滅物体が存在しており、間の 10,000 ステップには、画面内には点滅物体と移動物体の両方が存在している。この 20,000 ステップの間に、移動物体と点滅物体が注視されていた時間をそれぞれ測定した。

なお、定常状態からシミュレーションを始めるために、各ランの前にはネットワークを 2000 単位時間動作させた。

点滅物体の点滅間隔を変化させて実験を行ったところ、点滅物体は実験の開始時に注視されるのみで、ほぼラン全体に渡って移動物体が注視され続けるという結果になった。移動物体が画面から出た後も、点滅物体が注視されることはなかった。表 4.4 に示すように、点滅間隔に関わらず、同じ傾向が見られた。なお、点滅間隔が 0 というのは、常に光ったままになっているということを示している。

また、同様の実験を、背景の輝度が一様でないように設定して行った。表 4.5 に、20 回のランにおける平均値を示す。**4.4.2** の実験と同様に、背景の輝度は各ランについて独立してピクセルごとに、正規分布に従って確率的に選択した。その平均値は 127、分散は 12.7^2 である。表から明らかなように、このような設定下においても点滅物体はほぼ無視され、移動物体のみが注視され続けるという結果となった。

前節の結果と合わせて、提案モデルにより、移動物体の注視・追跡を行う一方で点滅しているだけの物体を無視できることがわかった。

表 4.3 移動物体が注視されていた時間割合 (変数：輝度)

Table 4.3 Percentage of attention time on the moving object (single object, variable: brightness).

	Background brightness		
Object brightness	$N(31, 3.1^2)$	$N(63, 6.3^2)$	$N(127, 12.7^2)$
$N(127, 12.7^2)$	75.50	4.19	0.00
$N(191, 19.1^2)$	97.76	93.63	58.16
$N(255, 25.5^2)$	98.05	97.74	92.78

(%)

表 4.4 各物体が注視されていた時間割合 (点滅物体あり)

Table 4.4 Percentage of attention time on each object (with a flickering object).

Flickering interval (steps)	Attention time on moving object(%)	Attention time on flickering object(%)
0	48.90	5.32
1	48.90	5.32
2	48.90	3.30
5	48.90	4.69
10	48.90	4.20
20	48.90	5.10
50	48.90	4.97
100	48.90	5.32

表 4.5 各物体が注視されていた時間割合 (点滅物体あり・背景の輝度が 0 でない場合)

Table 4.5 Percentage of attention time on each object (with a flickering object, bright background).

Flickering interval (steps)	Attention time on moving object(%)	Attention time on flickering object(%)
1	48.65	5.69
10	48.72	3.51
100	48.79	3.56

4.4.4 二つの移動物体が存在する環境における実験

複数の移動物体が存在する環境において行った実験について述べる。動画像の背景は変化することがなく、その輝度は 0 とした。それぞれの移動物体の輝度は 255、サイズは縦 16 ピクセル×横 16 ピクセルで共通であり、どちらも画像の左端から右端へと一定の速度で移動する。

一方の移動物体 (これを物体Aと呼ぶ) の一部が画面に入ってから 100 単位時間の後に、もう一方の物体 (物体B) が画面に入るものとした。なお、物体Aは画像の左上領域から右上領域へと、物体Bは画像の左下領域から右下領域へと移動する。物体Aの一部が画面に入ってから、両方の物体が完全に画面から出るまでを一回のランと考える。なお、物体の速度はランごとに正規分布に従って確率的に選択した。表中では、1000 ステップあたりに移動するピクセル数で表記してある。

表 4.6 の結果は、物体A・Bの速度をほぼ同じに設定した場合の結果である。20 回のランの間に、物体Aを含む領域が注視されていた時間の割合と、物体Bを含む領域が注視されていた時間の割合とを、それぞれ示してある。この結果から、二つの移動

表 4.6 それぞれの移動物体が注視されていた時間割合

Table 4.6 Percentage of attention time on each object.

Object velocity (steps)	Attention time on object A (%)	Attention time on object B(%)
$N(20, 2.0^2)$	44.35	53.78
$N(15, 1.5^2)$	45.08	53.52
$N(10, 1.0^2)$	43.09	55.86
$N(5, 0.5^2)$	51.88	47.01

表 4.7 より高速な移動物体が注視されていた時間割合

Table 4.7 Percentage of attention time on the faster moving object.

Velocity of object A	Velocity of object B			
	$N(20, 2^2)$	$N(15, 1.5^2)$	$N(10, 1.0^2)$	$N(5, 0.5^2)$
$N(20, 2.0^2)$	–	62.59	74.60	91.22
$N(15, 1.5^2)$	60.77	–	64.55	87.13
$N(10, 1.0^2)$	72.09	65.05	–	78.36
$N(5, 0.5^2)$	91.41	87.14	76.33	–

(%)

物体の速度が同等の場合にはそれぞれの物体がほぼ均等に注視されていることがわかる。実際に注視の軌跡を確認したところ、ほぼ交互に注視されていた。

また、表 4.7 の結果は、物体Aと Bの速度が異なるように設定した場合の結果である。20 回のランの間に、より高速に移動している物体を含む領域が注視されていた時間の割合を示してある。表から明らかなように、一方の物体がもう一方よりも速く移動する場合には、速度の差が広がるに従って、より高速な物体の方に注視が集中するという結果となった。

同様に、物体Aと Bの速度が異なるという設定において、背景の輝度が一様でない場合の実験を行った。この結果を、表 4.8 に示す。**4.4.2** の実験と同様に、背景の輝度は各ランについて独立してピクセルごとに、正規分布に従って確率的に選択した。そ

表 4.8 より高速な移動物体が注視されていた時間割合 (背景の輝度が 0 でない場合)

Table 4.8 Percentage of attention time on the faster moving object (with bright background).

Velocity of object A	Velocity of object B			
	$N(20, 2^2)$	$N(15, 1.5^2)$	$N(10, 1.0^2)$	$N(5, 0.5^2)$
$N(20, 2.0^2)$	–	59.03	75.72	89.00
$N(15, 1.5^2)$	58.53	–	66.56	81.55
$N(10, 1.0^2)$	73.30	65.11	–	68.97
$N(5, 0.5^2)$	88.40	81.97	70.88	–

(%)

表 4.9 より高速な移動物体が注視されていた時間割合 (交差あり)

Table 4.9 Percentage of attention time on the faster moving object (with crossover).

Velocity of object A	Velocity of object B			
	$N(20, 2^2)$	$N(15, 1.5^2)$	$N(10, 1.0^2)$	$N(5, 0.5^2)$
$N(20, 2.0^2)$	–	60.56	74.91	79.93
$N(15, 1.5^2)$	59.40	–	68.05	76.53
$N(10, 1.0^2)$	73.64	67.06	–	69.70
$N(5, 0.5^2)$	90.16	83.42	72.32	–

(%)

の平均値は127、分散は12.7^2である。この設定においても、表4.7の結果と同様に、速度の差が広がるに従って、より高速な物体の方に注視が集中するという結果となった。

表4.9は、類似の実験を、二つの物体が各ランで必ず一回交差するように設定して行った場合の結果である。物体Aは今までの実験と同様に画像の左上領域から右上領域へと移動するが、物体Bは、画像の右上領域から左上領域へと移動する。この際、物体BのY座標は、物体Aのそれに対して-15ピクセルから+15ピクセルの範囲で、ランごとに独立して一様な確率で決定されるものとした。物体の高さは16ピクセルであるので、各ランで必ず一回交差することとなる。表から明らかなように、このように物体が交差する場合においても、より高速な物体を注視し続けることが可能であることが確認された。

これらの結果から、他と比べて非常に高速な移動物体が存在する場合にはそれを注視し続ける一方で、速度の近い物体が存在する場合にはほぼ均等に注視を行うという、物体の移動速度に応じた望ましい注視制御が行えることが確認できた。

4.5 むすび

本章では、パルスニューラルネットワークに短期抑圧 (STSD) 現象を導入することによって生じる「慣れ」の効果を利用して、注視制御を行うネットワークモデルを提案した。計算機実験により、本モデルによって、基本的な移動物体の注視・追跡に加え、点滅しているだけの物体を無視することができ、さらに、高速に移動している物体を優先して注視できるということを確認した。極めて単純な構造のニューラルネットワークにおいてこれらの性質を同時に兼ね備えるというのは、STSD 現象を導入することによって初めて可能になったことである。このモデルはまだ基礎的な段階にあるが、将来的には、個々のパルスニューロン素子をハードウェア化することによって高速な超並列処理が期待され、画像の走査を必要とする一般的な画像処理アルゴリズムに比べて優位性が見込まれる。

今後の課題としては、パラメータの自動化、ないし汎用的なパラメータ選定手法を確立することが挙げられる。特に、実アプリケーションへの応用を行うためには、輝度の差異に対する敏感性を調整する手法の確立が必要である。本章では、点滅しているだけの物体は注視対象からは除外すべきであるという前提の下にモデルの構築を行った。しかしながら、逆に点滅物体を注視することが必要とされる問題に対しても、将来的にはパラメータの適切な調整によって対応できるのではないかと考えている。具体的には、短期抑圧の減衰を早くすることで、点滅物体も高速の動物体と同様に注視できるものと思われる。また、カラー画像への拡張も検討課題である。現行のシステムでもカラー画像を多階調白黒画像に変換して入力とすることで処理は可能であるが、入力画像中に同一の輝度であるにも関わらず全く異なる色が複数登場するような場合には、精度の劣化は避けられない。そのような場合のために、色座標軸に対応するネットワークを別々に用意し、その出力結果を統合するようなシステムが必要となる。

第 5 章

結論

本論文では、従来の積分器型ニューロン素子に比べより生体の神経細胞に近いモデルであるパルスニューロン素子の時系列情報処理能力に着目し、工学的な利用を前提とした三種類の新しい階層型ネットワークモデルを提案した。

強化学習に基づく学習アルゴリズムをパルスニューラルネットワークに導入することで広範な応用範囲を持つシステムを構築するというアプローチと、新しい生理学的知見をパルスニューロン素子に導入してその工学的な利用価値を模索するというアプローチという、二つの側面から研究を行った。

前者のアプローチに基づいて考案した二つのネットワークモデルでは、パルスニューラルネットワークへの強化学習アルゴリズムの導入と、ネットワークの動的な拡張による学習処理を達成した。

第一のネットワークは、摂動的なパルスを各ニューロンに加えることで、偶発性を利用して時系列的な入出力空間の探索を行うモデルであり、入力層、隠れ層、出力層の三層からなる。学習としては、外部から与えられる強化信号に基づいて行われる結合荷重の修正に加え、入出力関係に対応した隠れ層ニューロンを動的に追加し、ネットワークの拡張を行いながら望ましい出力を学習していく。計算機実験により、時系列的な入力を処理して適切な出力を学習することができるということが確認された。

第二のネットワークは四層構造の階層型ネットワークであり、二層の隠れ層をそれぞれ減衰率の異なるパルスニューロン素子で構成することを特徴とする。一番目の隠れ層が同時に入力される情報を処理し、二番目の隠れ層が連続的に入力される情報を処理することで、部分観測マルコフ決定過程における曖昧な状態の識別を行う。このモデルでは、摂動を用いる代わりに出力層への結合で確率的な処理を行う。また、各ニューロンが二次的な強化信号を発生することにより、従来の強化学習における価値関数の時間的な伝搬と類似した学習が可能となっている。計算機実験により、完全観測マルコフ決定過程および部分観測マルコフ決定過程の両方において、このモデルが有効に働くことが確認された。

後者のアプローチに基づいて考案した第三のネットワークは、生体のシナプスにおける短期抑圧現象を導入した階層型ニューラルネットワークであり、短期抑圧現象の

特徴を利用することで並列処理を前提とした注視制御を行うものである。このモデルでは学習は行わず、設定されたパラメータに応じて結合荷重が予め固定される。入力としては実時間に更新される低解像度・多階調の白黒画像を受け取り、画像中の注視するべき領域を、簡易ながら高速に選定し出力する。この動作は、パルスニューロン素子における STSD 現象の特徴と、ネットワーク構造の組み合わせとによって実現されている。計算機実験により、短期抑圧現象の導入によって注視領域の選定が適切に行われているということが確認された。

パルスニューラルネットワークは、従来型のニューラルネットワークでは扱えなかったような高度な知的情報処理を実現できるのではないかと期待されている。しかし、その挙動にはまだ未解明の要素が多く残されており、工学的な利用は殆んど行われていないのが現状である。このような現状に対し、本論文では、工学的な利用を目的とした三種類の異なるパルスニューラルネットワークモデルを提案し、それぞれ良好な情報処理能力を確認した。

謝辞

萩原将文先生に。本研究を進めるにあたり常に適切なアドバイスを下さっただけでなく、大学生活全般について親身になって相談に乗って下さったことに心から感謝致します。先生の今後ますますのご活躍を祈願致します。

査読をして頂いた先生方、小沢慎治先生、櫻井彰人先生、岡浩太郎先生に。論文を完成させるにあたり、先生方には様々な角度からの多数の貴重な御指摘を頂きました。心より御礼申し上げます。

萩原研究室の皆様に。特に、長名優子先生には多くのご助言を頂きました。皆様にディスカッション・発表練習・論文添削といったことで助けて頂いただけでなく、家族のような暖かい雰囲気の中で研究をさせて頂いたことがこの成果につながりました。どうもありがとうございました。

父と母に。今の私があるのも二人のお蔭です。私を生み育ててくれたことに尽きぬ感謝を捧げます。

参考文献

[1] W.S.McCulloch and W.H.Pitts, “A logical calculus of the ideas immanent in nervous activity,” Bulletin of Mathematical Biophysics, vol.5, pp.115-133, 1943.

[2] E.D.Adrian, “The impulses produced by sensory nerve endings,” Journal of Physiology, vol.61, pp.49-72, 1926.

[3] E.D.Adrian, “The Basis of Sensation,” W.W.Norton, New York, 1928.

[4] V.B.Mountcastle, “Modality and topographic properties of single neurons of cat’s somatosensory cortex,” Journal of Neurophysiology, vol.20, pp.408-434, 1957.

[5] D.H.Hubel and T.N.Wiesel, “Receptive fields of single neurons in the cat’s striate cortex,” Journal of Physiology, vol.148, pp.574-591, 1959.

[6] D.O.Hebb, “The Organization of Behavior,” New York: John Wiley, 1949.

[7] F.Rosenblatt, “The perceptron: A probablistic model for information storage and organization in the brain”, Psychological Review, vol.65, no.6, pp.386-408, 1958.

[8] F.Rosenblatt, “Principles of Neurodynamics,” Spartan Books, 1962.

[9] M.Minsky and S.Papert, “Perceptrons”, MIT PRess, Cambridge, 1969.

[10] A.M.Turing, “Computing machinery and intelligence,” Mind, 59, no.236, pp.433-460, 1950.

[11] A.Newell, J.C.Shaw and H.A.Simon, “Chess-playing programs, and the problem of complexity”, IBM Journal of Research and Development, vol.2, no.4, pp.320-335, 1958.

[12] J.J.Hopfield, “Neural networks and physical systems with emergent collective computational abilities”, Proceedings of National Academic Science, USA, vol.79, no.8, pp.2554-2558, 1982.

[13] T.Kohonen, "Self-organized formation of topologically correct feature maps," Biological Cybernetics, vol.43, no.1, pp.59-69, 1982.

[14] G.E.Hinton, T.J.Sejnowski and D.H.Ackley, "Boltzmann machines: Constraint satisfaction networks that learn," Technical Report CMU-CS-84-119, Carnegie-Mellon University, 1984.

[15] D.H.Ackley, G.E.Hinton, T.J.Sejnowski, "A learning algorithm for Boltzmann machines," Cognitive Science, vol.9, pp.147-169, 1985.

[16] D.Rumelhart, D.Hinton and G.Williams, "Learning internal representations by error propagation," in D.Rumelhart and F.McClelland, eds., Parallel Distributed Processing, vol.1, MIT Press, 1986.

[17] T.J.Sejnowski, C.R.Rosenberg, "NETtalk: A parallel network that learns to pronounce English text", Complex Systems, 1, pp.145-168, 1987.

[18] S.Amari, "Theory of adaptive pattern classifiers," IEEE Transactions on Electronic Computers, EC-16, no.3, pp.299-307,1967.

[19] G.Carpenter and S.Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," Computer Vision, Graphics, and Image Processing, vol.37, no.1, pp.54-115, 1987.

[20] A.L.Hodgkin and A.F.Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," Journal of Physiology, vol.117, pp.500-544, 1952.

[21] W.Maass, "Networks of spiking neurons: the third generation of neural network models," Neural Networks, vol.10, no.9, pp.1659-1671, 1997

[22] W.Maass and C.M.Bishop, "Pulsed Neural Networks," Cambridge, MA: M.I.T. Press, 1999.

[23] R.Eckhorn, R.Bauer, W.Jordan, M.Brosch, W.Kruse, M.Munk and H.J.Reitboeck, "Coherent oscillations: A mechanism of feature linking in the visual cortex?" Biological Cybernetics, vol.60, no.2, pp.121-130, 1988.

[24] W.Bialek, F.Rieke, R.R. de Ruyter van Stevenick and D.Warland, "Reading a neural code," Science, vol.252, pp.1854-1857, 1991.

[25] M.Abeles, “Firing rates and well-timed events,” in E.Domany, K.Schulten and J.L.van Hemmen, eds., Models of Neural Networks 2, Springer, New York, pp.121-140, 1994.

[26] R.Lestienne, “Determination of the precision of spike timing in the visual cortex of anaesthetised cats,” Biological Cybernetics, vol.74, no.1, pp.55-61, 1996.

[27] S.Thorpe, D.Fize and C.Marlot, “Speed of processing in the human visual system,” Nature, vol.381, pp.520-522, 1996.

[28] 黒柳奨, 岩田彰, “音源方向定位聴覚神経系モデルによる ITD,ILD の脳内マッピングの実現,” 電子情報通信学会論文誌 DII, vol.J79-DII, no.2, pp.267-276, 1996.

[29] 塚田稔, “海馬記憶神経回路と学習則,” 日本神経回路学会誌, Vol.4, No.3, pp.126-135, 1997.

[30] 関根好文, 山崎昭広, 黒澤開輝, 佐藤典子, “Λ形トランジスタを用いた汎用形ハードウェアニューロンモデル,” 電子情報通信学会論文誌, J78-DII, no.1, pp.131-139, 1995.

[31] M.Hanagata, Y.Horio and K.Aihara, “Asynchronous pulse neural network model for VLSI implementation,” IEICE Transactions on Fundamentals, E81-A, no.9, pp.1853-1859, 1998.

[32] H.Hikawa, “A digital hardware pulse-mode neuron with piecewise linear activation function,” IEEE Transactions on Neural Networks, vol.14, no.5, pp.1028-1037, 2003.

[33] K.Aihara, T.Takabe and M.Toyoda, “Chaotic neural networks,” Physics Letters A, vol.144, no.6/7, pp.333-340, 1990.

[34] N.Ichinose and K.Aihara, “Asynchronous Chaotic Neural Netoworks”, ‘Towards the Harnessing of Chaos’ editted by M. Yamaguchi, Elsevier, pp.353-355, 1994

[35] 市瀬夏洋, 合原一幸, “非同期カオスニューラルネットワークにおけるパルス伝搬ダイナミックスの解析,” 電子情報通信学会誌 A, J78-A, no.3, pp.373-380, 1995.

[36] H.Kitajima, T.Yoshinaga, K.Aihara, H.Kawakami, “Chaotic bursts and bifurcation in chaotic neural networks with ring structure,” International Journal of Bifurcation and Chaos, vol.11, no.6, pp.1631-1643, 2001.

[37] 元木誠, 濱上知樹, 小圷成一, 平田廣則, “パルスニューラルネットワークにおける破局的な忘却の抑制を考慮したヘブ型学習則,” 電気学会論文誌C, vol.123-C, no.6, pp.1124-1133, 2003.

[38] B.Ruf and M.Schmitt, “Self-organization of spiking neurons using action potential timing”, IEEE Transactions on Neural Networks, vol.9, no.3, pp.575-578, 1998.

[39] 雨森 賢一, 石井 信, “精緻な時空間スパイク列の自己組織化学習と想起,” システム制御情報学会論文誌, vol.13, no.7, pp.308-317, 2000.

[40] C.Panchev and S.Wermter, “Sequential processing in neuroscience inspired models,” Proceedings of Third International Workshop on Current Computational Architectures Integrating Neural Networks and Neuroscience, pp.84-88, 2000.

[41] R.C.O'Reilly, “Biologically plausible error-driven learning using local activation differences: the generalized recirculation algorithm,” Neural Computation, vol.8, no.5, pp.895-938, 1996.

[42] B.Ruf and M.Schmitt, “Learning temporally encoded patterns in networks of spiking neurons,” Neural Processing Letters, vol.5, no.1, pp.9-18, 1997.

[43] R.S.Sutton and A.G.Barto, “Reinforcement Learning: An Introduction,” MIT Press, 1998.

[44] R.S.Sutton, “Learning to predict by the methods of temporal differences,” Machine Learning, vol.3, no.1, pp.9-44, 1988.

[45] C.J.Watkins and P.Dayan, “Technical Note: Q-Learning,” Machine Learning, vol.8, no.3/4, pp.279-292, 1992.

[46] R.J.Williams, “Simple statistical gradient following algorithms for connectionist reinforcement learning,” Machine Learning, vol.8, no.3, pp.229-256, 1992.

[47] A.G.Barto, R.S.Sutton and P.S.Brouwer, “Associative search network: a reinforcement learning associative memory,” Biological Cybernetics, vol.40, no.3, pp.201-211, 1981.

[48] A.G.Barto, R.S.Sutton and C.W.Anderson, “Neuronlike adaptive elements that can solve difficult learning control problems,” IEEE Transactions on Systems, Man, and Cybernetics, Vol.13, No.5, pp.834-846, 1983.

[49] W.Schultz, P.Dayan and P.R.Montague, “A neural substrate of prediction and reward,” Science, vol.275, pp.1593-1599, 1997.

[50] K.Doya, “Metalearning and neuromodulation,” Neural Networks, vol.15, no.4-6, pp.495-506, 2002.

[51] D.Gorse, D.A.Romano-Critchley and J.G.Taylor, “A pulse-based reinforcement algorithm for learning continuous functions,” Neurocomputing, vol.14, no.4, pp.319-344, 1997.

[52] H.B.Barlow, “Single units and sensatioperceptual psychology?,” Perception, Vol.1, pp.371-394, 1972.

[53] A.K.Engel, P.Konig, A.K.Kreiter, T.B.Schillen and W.Singer, “Temporal coding in the visual cortex: new vistas on integration in the nervous system,” Trends In Neuroscience, Vol.15, No.6, pp.218-226, 1992.

[54] W.Singer, C.Gray, A.Engel, P.Konig, A.Artola and S.Brocher, “Formation of cortical cell assemblies,” Cold Spring Harbor Symposium on Quantitative Biology, Vol. LV., Cold Springer Harbor Laboratory Press, pp.939-952, 1990

[55] H.Fujii, H.Ito, K.Aihara, N.Ichinose and M.Tsukada, “Dynamical Cell Assembly Hypothesis-Theoretical Possibility of Spatio-temporal Codeing in the Cortex,” Neural Networks, Vol.9, No.8, pp.1303-1350, 1996.

[56] 櫻井芳雄, “ニューロン集団の相関活動をみる,” 科学, Vol.66, no.11, pp.784-792, 1996.

[57] A.Waibel, “Modular construction of time delay neural networks for speech recognition,” Neural Computation, Vol.1, no.1, pp.39-46, 1989.

[58] A.Cleeremans, D.Servan-Schreiber and J.McClelland, “Finite state automata and simple recurrent networks,” Neural Computation, Vol.1, no.3, pp.372-381, 1989.

[59] 重松征史, 松本元, “時系列連想記憶の自己組織化 -時系列学習の生理学的な可能性について-,” 信学技報, NC94-131, pp.131-138, 1995.

[60] 武田治, 黒柳奨, 岩田彰, “階層構造を持つパルス駆動型ニューロンモデルを用いた時系列符号化,” 信学技報, NC97-117, pp.125-132, 1998.

[61] A.G.Barto and P.Anandan, "Pattern recognizing stochastic learning automata," IEEE Transactions on Systems, Man, and Cybernetics, Vol.15, No.3, pp.360-375, 1985.

[62] A.G.Barto, R.S.Sutton and C.W.Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," IEEE Transactions on Systems, Man, and Cybernetics, Vol.13, No.5, pp.834-846, 1983.

[63] P.J.Werbos, "Backpropagation through time: what it does and how to do it," Proceedings of the IEEE, vol.78, no.10, pp.1550-1560, 1990.

[64] M.Abeles, "Role of the cortical neuron: integrator or coincidence detector?," Israel Journal of Medical Sciences, vol.18, no.1, pp.83-92, 1982.

[65] F.C.Hoppensteadt and E.M.Izhikevich, "Thalamo-cortical interactions modeled by weakly connected oscillators: could the brain use FM radio principles?," BioSystems, vol.48, no.1-3, pp.85-94, 1998.

[66] M.Schmitt, "On the implications of delay adaptability for learning in pulsed neural networks," NeuroCOLT Technical Report, NC-TR-00-069, 2000.

[67] M.Samuelides, S.Thorpe and E.Veneau, "Implementing hebbian learning in a rank-based neural network," Proceedings of 7th International Conference on Artificial Neural Networks, pp.145-150, 1997.

[68] T.W.Berger, M.Baudry, R.D.Brinton, J-S.Liaw, V.Z.Marmarelis, Y.Park, B.J.Sheu and A.R.Tanguay Jr., "Brain-implantable biomimetic electoronics as the next era in neural prosthetics," Proceedings of the IEEE, vol.89, no.7, pp.993-1012, 2001.

[69] A.Ueno, K.Hori and S.Nakasuka, "Simultaneous learning of situation classification based on rewards and behavior selection based on the situation," Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol.3, pp.1510-1517, 1996.

[70] C.Gaskett, D.Wettergreen and A.Zelinsky, "Q-Learning in continuous state and action spaces," Proceedings of the 12th Australian Joint Conference on Artificial Intelligence, pp.417-428, 1999.

[71] S.D.Whitehead, “A complexity analysis of cooperative mechanisms in reinforcement learning,” Proceedings of 9th National Conference on Artificial Intelligence, vol.2, pp.607-613, 1991.

[72] A.Onat, H.Kita, Y.Nishikawa, “Recurrent neural networks for reinforcement learning: architecture, learning algorithms and internal representation,” Proceedings of the 1998 IEEE International Joint Conference on Neural Networks, 3, pp.2010-2015, 1998.

[73] L.Chrisman, “Reinforcement learning with perceptual aliasing: The perceptual distinctions approach,” Proceedings of the 10th International Conference on Artificial Intelligence, pp.183-188, 1992.

[74] R.A.McCallum, “Instance-based utile distinctions for reinforcement learning with hidden state,” Proceedings of the 12th International Conference on Machine Learning, pp.387-395, 1995.

[75] 佐伯勝敏, 関根好文, 合原一幸, “エンハンスメント型MOSFETを用いたパルス形バーストニューロンモデル,” 電子情報通信学会論文誌, J85-C, no.3, pp.174-180, 2002.

[76] R.A.Deisz and D.A.Prince, “Frequency-dependent depression of inhibition in guinea-pig neocortex in vitro by GABAB receptor feed-back on GABA release,” Journal of Physiology, vol.412, vol.1, pp.513-541, 1989.

[77] J.Varela, K.Sen, J.Gibson, J.Fost, L.F.Abbott, S.B.Nelson, “A quantitative description of short-term plasticity at excitatory synapses in layer 2/3 of rat primary visual cortex,” Journal of Neuroscience, vol.17, no.20, pp.7926-7940, 1997.

[78] L.F.Abbott, J.A.Varela, K.Sen, et al, “Synaptic depression and cortical gain control,” Science, vol.275, pp.220-224, 1997.

[79] E.Kowler, “Eye movements and visual attention,” in Encyclopedia of Cognitive Science, MIT Press, 1999.

[80] D.H.Ballard, “Reference frames for animate vision,” Proceedings of IJCAI, vol.41, pp.1635-1641, 1989.

[81] Y.Sun and R.Fisher, “Object-based visual attention for computer vision”, Artificial Intelligence, vol.146, no.1, pp.77-123, 2003.

[82] S.Chopra-Khullar and N.I.Badler, “Where to look? Automating attending behaviors of virtual human characters,” Proceedings of the third annual conference on autonomous agents, vol.4, no.1-2, pp.16-23, 1999.

[83] C.Peters and C.O’Sullivan, “Bottom-up visual attention for virtual human animation,” Proceedings of the Computer Animation and Social Agents 2003, pp 111-117, 2003.

[84] S.Yantis and A.P.Hillstrom, “Stimulus-driven attentional capture: Evidence from equiluminant visual objects,” Journal of Experimental Psychology: Human Perception and Performance, vol.20, no.1, pp.95-107, 1994.

[85] S.L.Franconeri and D.J.Simons, “Moving and looming stimuli capture attention,” Perception and Psychophysics, vol.65, no.7, pp.999-1010, 2003.

[86] 瀧田 航一朗, 長名 優子, 萩原 将文, “パルスニューラルネットワークにおけるネットワーク拡張型強化学習アルゴリズム,” 電気学会論文誌 C, vol.121-C, no.10, pp.1634-1640, 2001.

[87] 瀧田 航一朗, 萩原 将文, “部分観測マルコフ決定過程下の強化学習のためのパルスニューラルネットワーク学習則,” 電子情報通信学会論文誌, vol.J86-DII, no.7, pp.1067-1077, 2003.

[88] 瀧田 航一朗, 萩原 将文, “短期抑圧現象を取り入れたパルスニューラルネットワークによる注視制御,” 電気学会論文誌 C, (採録決定.)