

修士論文

リカレント シンプルスパイキング
ニューラルネットワークを用いた
状態機械の実現

指導教官 富田 眞治 教授

京都大学大学院情報学研究科
修士課程通信情報システム専攻専攻

三輪 忍

平成11年2月8日

リカレント シンプルスパイク ニューラル ネットワーク を用いた 状態機械の実現

三輪 忍

内容梗概

従来のニューラルネット研究では、認識や学習といったように脳の特定の機能に問題を絞って、それを実現することを目指してきた。つまり、ニューラルネットは神経系をモデル化したものであるが、多様な機能をもつ神経系とは異なり、パターン認識などの特定の機能を実現するために用いられることが多かった。また、特定の機能が実現できたとして、それらをどう組み合わせることで、脳と同じように振る舞うシステムが実現できるかということは、ほとんど議論されてこなかった。

こうした点が従来のニューラルネット研究に行き詰まりを与えていると我々は考える。そこで我々は、ニューラルネットを特定の機能を実現するための道具としてではなく神経系のモデルとして捉え直すことで、脳と同じように振る舞うシステムの構築を目指す。

システムの糸口を得るために、現実のニューロンに近いニューロン・モデルである Pulse-based Model を用いてヒューリスティックにニューラルネットを構成する。ニューラルネットは、回帰結合をもつリカレントネットを用いる。これに、解くのに記憶が必要な問題である迷路問題を与える。それは、記憶が認識や学習といった他の機能の基盤となっていると我々は考えるからである。

Pulse-based Model を用いたリカレントネットには、誤差逆伝播法のような一般的なネットワーク構成法が存在しない。そこで、Pulse-based Model でまず論理素子を作り、それを用いて状態機械としてニューラルネットを構成した後、ヒューリスティックな変換を施すという方法をとることにする。

本論では、この構成法に従い、Pulse-based Model で状態機械を構成した。Pulse-based Model にはパルス駆動型モデルとシンプルスパイクモデルが存在することから、両者を比較し、シンプルスパイクモデルを用いることにした。シンプルスパイクモデルを用いて、基本素子である AND, OR, そして NOT の代わりに否定包含ニューロンを作成した。そして、これらを用いて、迷路を解く状態機械を実現した。

Implementation of State Machine in Recurrent Simple Spiking Neural Network

Shinobu MIWA

Abstract

In the previous studies of Neural net, the aims are to extract the problems of recognition and learning, i.e. specific functions in a brain, and to implement such functions. This is that Neural net has been used for implementing specific functions such as pattern recognition, although Neural net has been modeled on a nervous system with various functions. And it has not been argued how to implement a brain-like system with specific functions when they are implemented.

We suppose that Neural net is in deadlock for such points. So we aim at the construction of a brain-like system by regarding Neural net as the model of a nervous system, not as the implementational tool for specific functions.

Looking for the clue of the system, we construct Neural net heuristically with Pulse-based Model which is a close neuron model to an actual neuron. Neural net is Recurrent net which has the feedback loops. We make this to solve maze problems which need memory to solve, because we suppose that memory is based on other functions, such as recognition and learning.

There isn't a general network construction such as Back Propagation on Recurrent net with Pulse-based Model. So we suggested the construction which is implementing neurons to behave as logic gates and which is implementing a state machine with them, then, which is converting it heuristically.

In this paper, we implemented a state machine by Pulse-based Model. There are two models in Pulse-based Model, one is Pulsed Model and another is Simple-Spiking Model. So we compared them and adopted later. We constructed basic logic gates with later which are AND, OR, and which are non-implication gates replaced for NOT. And we implemented a state machine solving maze problems with them.

リカレントシンプルスパイキングニューラルネットワーク を用いた状態機械の実現

目次

第1章	はじめに	1
第2章	研究の背景	2
2.1	ニューロン・モデル	2
2.1.1	ニューロン	3
2.1.2	ニューロン・モデルの抽象度	6
2.2	記憶	7
2.2.1	脳における記憶のしくみ	7
2.2.2	短期記憶とループ	8
2.3	リカレントネット	9
2.3.1	リカレントネット	9
2.3.2	リカレントネットの能力	11
2.4	研究の方針	13
2.5	研究の方法	13
2.5.1	ネットワークの構成法	14
2.5.2	Pulse-based に基づくネットワークの構成法	15
2.6	関連研究	15
第3章	リカレントシンプルスパイキングニューラルネットワークを用いた状態機械の実現	19
3.1	パルス駆動型モデル vs. シンプルスパイキングモデル	19
3.1.1	パルス駆動型モデル	19
3.1.2	シンプルスパイキングモデル	23
3.1.3	モデルの比較	26
3.2	論理素子の構成	27
3.3	順序回路の構成	33
3.3.1	迷路問題	33
3.3.2	アルゴリズム	35
3.3.3	迷路を解く順序回路の構成	36

第4章 課題	43
4.1 パルスと同期	43
4.2 今後の課題	45
謝辞	47
参考文献	48

第1章 はじめに

リカレントニューラルネットワーク(以下リカレントネットとする)とは、フィードバック・ループ(以下ループとする)をもつニューラルネットのことである。リカレントネットのループは、状態機械におけるループと同様、入力履歴をその状態として代表させることができるため、オートマトンと同様の働きができる。このためリカレントネットは、思考や言語、記憶といった、脳の高次情報処理を実現、解明する上での大きな糸口を与えるものと期待されている。

そこで、リカレントネットを用いて、脳のようなものを作ることにする。脳における基本的な処理の1つとして記憶が挙げられることから、我々はまず記憶について考える。

本稿では、そのようなリカレントネットの実現を目指す過程として、シンプルスパイクニューロンモデル(以下シンプルスパイクモデルとする)を用いたリカレントネットにより状態機械を実現したので、その構成について述べる。

次章では、まず、ニューラルネットの構成要素であるニューロン・モデルについて述べる。モデルには、現実のニューロンにおいて情報の最小表現と考えられている、パルスのレベルでモデル化された **Pulse-based Model** が存在する。続いて、脳における記憶のしくみについて説明し、そこで得られた知見からループ構造の重要性を述べる。そこで、リカレントネットについて説明し、これを用いることを述べる。続いて、ネットワークの構成法としては色々考えられるが、1.) **Pulse-based Model** を用いて論理素子を構成し、2.) そうしてできたネットワークに何らかの変換を施す、という方法をとることについて述べる。

3章では、前述の構成法に基づき、1.) までの構成を行ったので、それを説明する。まず、**Pulse-based Model** にはパルス駆動型ニューロンモデルとシンプルスパイクモデルが存在するが、後者を選択した理由について述べる。そして、そのモデルで論理素子を実現し、これにより迷路を解く状態機械を実現したので、その構成について説明する。

最後に、構成の過程で生じてきた、今後の課題について述べる。

第2章 研究の背景

ループをもつニューラルネットであるリカレントネットはオートマトン的な働きができることから、これによる人間の hoch 情報処理の実現、解明に期待が寄せられている。そこで我々は、これを用いて脳のようなものを作ることにする。脳では様々な処理が行われているが、まずは記憶について考えることにする。

そこで以下では、次節において、ニューラルネットを構成する要素であるニューロン・モデルについて、現実のニューロンのどのような特徴が抽象化されているか説明する。モデルは、大きく分けて2つのモデルが存在する。

続いて、2.2において、脳における記憶のしくみについて説明する。記憶には短期記憶と中長期記憶が存在すること、そして、それらが何に担われているかを説明する。そこから、記憶、特に短期記憶のためには、ループが重要な役割を果たしている、とする我々の仮定を導く。

そして2.3において、ループをもつニューラルネットであるリカレントネットについて説明する。2.1で述べた2つのモデルの違いにより、それを用いたリカレントネットがどのように異なっているか説明する。また、参考までに、リカレントネットの計算能力について述べておく。

以上の知見を基に、研究の方針、及び、方法について、続く2節で述べる。リカレントネットを用いて記憶の研究を行うにあたって、リカレントネットに記憶が必要な問題を解かせることにする。このようなリカレントネットを構成する方法としては、2.3で述べた2つのリカレントネットそれぞれについて考えられる。そこで、どちらの構成法を用いるか検討し、それに基づき、次章からネットワークを構成していく。その前に、我々と同種の問題をリカレントネットに解かせる研究を谷([10][11])が行っているので、彼の構成法について述べ、その違いを明らかにしておく。

2.1 ニューロン・モデル

ニューロンのモデルには、McCulloch & Pittsの形式ニューロン・モデルから、現実のニューロンの振る舞い[5]をほぼ完全に記述できるモデルまで、様々な抽象度を持つものが提案されている。これらのモデルは現実のニューロンの一部を抽象化したものである。そこで以下、まず次節において、現実のニューロンの振る舞いについてまとめた後、いくつかのモデルがそのどのような特徴

を抽象化しているのかについて述べる。

2.1.1 ニューロン

シナプス ニューロンとニューロンはシナプスと呼ばれる構造によって接続されており，ニューロン間の信号の伝達はシナプスを狭んで行われる．信号の送信側のニューロンはシナプス前細胞，受信側の細胞はシナプス後細胞と呼ばれる．シナプスにおける信号伝達は，多くの場合，化学伝達と呼ばれる方法で行われる．化学伝達では，シナプス前細胞が放出した神経伝達物質をシナプス後細胞が受け取ることによって信号が伝達される．一般にシナプスにおける信号の伝達し易さは異なっており，これを結合重み，あるいは単に重みと呼ぶ．

膜電位とチャネル 細胞膜の外側を基準とした内側の電位を膜電位と呼ぶ．静止状態では，細胞の内側の電位は外側より低く，膜電位は約 -70mV となっている．この電位を静止電位と呼ぶ．

膜電位は，細胞膜上に存在するチャネルを通して，イオンが細胞の内/外に流入/流出することで変化する．膜電位を決定するイオンには， Na^+ ， K^+ ，そして， Cl^- イオンなどがあり，それぞれに対応するチャネルが存在する．膜電位は， Na^+ イオンの流入により上昇し， K^+ イオンの流出，あるいは， Cl^- イオンの流入によって低下する．

信号伝達において，神経伝達物質を受け取ることによって開くチャネルは，リガンド感受性チャネルと呼ばれる．チャネルには，また，膜電位によって制御されるものもあり，これらは電位感受性チャネルと呼ばれる [2]．リガンド感受性チャネルには，主に， Na ， K ， Cl チャネルがあり，神経インパルスの発生タイミングに関与する．電位感受性チャネルには，主に， Na ， K チャネルがあり，神経インパルスの持続時間に関与する．

図 1 に， Na^+ ， K^+ の各イオンに対する細胞膜のコンダクタンスと，膜電位の変化の様子を示す．膜電位は， Na^+ コンダクタンス g_{Na} と K^+ コンダクタンス g_{K} の差 $g_{\text{Na}} - g_{\text{K}}$ にほぼ比例する．

興奮性伝達 興奮性の信号伝達は，神経伝達物質によってシナプス後細胞のリガンド感受性の Na チャネルが開くことによって始まる．リガンド感受性 Na チャネルが開くと，そこを通して Na^+ イオンが細胞内に流入することによって，膜電位は上昇する．上昇量，すなわち開くチャネルの数は，結合重みに依存する．開いたチャネルは，短い時間の間に再び閉じてしまう．

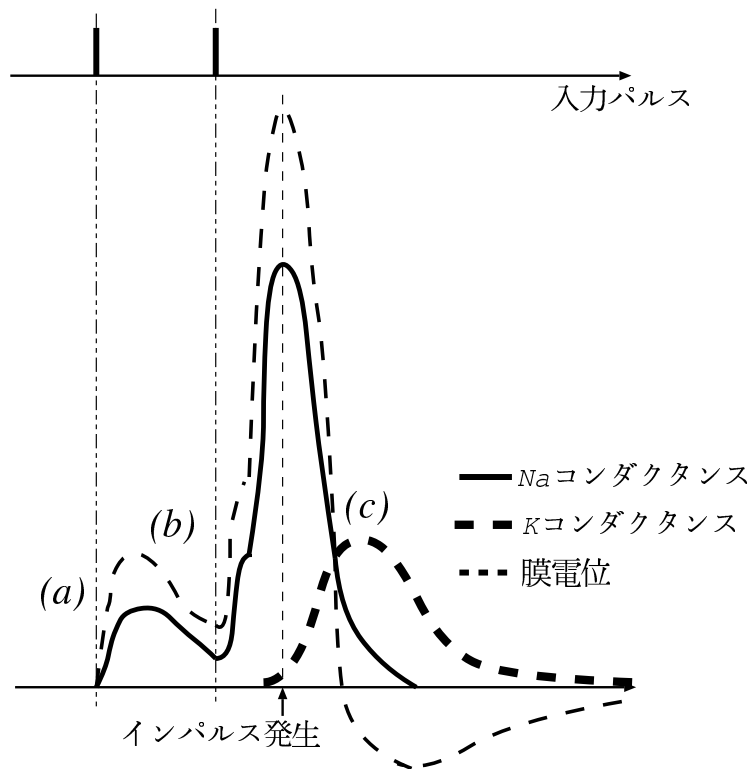


図1: コンダクタンスと膜電位の変化

開いたリガンド感受性 Na チャネルが少ない場合には (図 1 (a)), 流入した Na^+ イオンは拡散し, 電位は速やかに静止電位に戻ってしまう。

しかし, 多くのリガンド感受性 Na チャネルが開いた場合には, 拡散するより早くチャネルの付近に蓄積した Na^+ イオンによって膜電位が大きく上昇する。膜電位がある一定の値, 閾値電位を越えると (図 1 (b)), 電位感受性の Na チャネルが開き始める。すると, 電位感受性 Na チャネルを通して流入した Na^+ イオンが更に別の電位感受性 Na チャネルを開く。この正のフィードバックにより, 膜電位は急激に上昇する。これをニューロンの発火と呼ぶ。

電位感受性 Na チャネルには, 不活性化と呼ばれる機能があり, 一度開いたチャネルはすばやく自ら閉じ, 以降しばらくの間は再び開くことがない。

電位感受性 K チャネルは, 図 1 に示されているように, 電位感受性 Na チャネルに比べ, 電位の変化に対してゆっくりと開き, ゆっくりと閉じる。この電位感受性 K チャネルの働きによって, K^+ イオンが流出し, 一度上昇した膜電位は再び下降する。また, 電位感受性 K チャネルが閉じるのが遅れるため, 膜電位は一時的に静止電位を越えて下降する。

このような、電位感受性チャンネルの働きによる膜電位の急激な上昇と下降によって、1つの神経インパルスが形成される。神経インパルスが発生した場合、次の細胞に対して神経伝達物質の放出が行われ、信号がより下流へと伝わっていくことになる。

抑制性伝達 抑制性の伝達では、神経伝達物質によって、リガンド感受性Kチャンネル、または、リガンド感受性Clチャンネルが開く。リガンド感受性Naチャンネルが電位を上昇させるのとは異なり、これらのチャンネルの開放は必ずしも膜電位を下降させない。これらのチャンネルはNa⁺イオンの流入による内向の電流を打ち消すことにより、膜電位の上昇をまさに抑制する働きを持つ。従って、膜電位が高い状態で抑制性のチャンネルが開くと膜電位は降下するが、膜電位が十分に低い場合には、膜電位の更なる降下を引き起こすのではなく、それ以降の興奮性の伝達による膜電位の上昇を妨げる働きを持つ。

不応期と閾値電位 前述のように、一度開いた電位感受性Naチャンネルは不活性化により、一旦開くと以降しばらくの間は開くことができない。この間は、入力刺激の強弱に関わらず、ニューロンは発火することができない。この期間を絶対不応期と呼ぶ。絶対不応期の間は、閾値電位が無限大となっていると考えることもできる。

また、発火後、絶対不応期以降も、膜電位が上昇している間は、電位感受性Kチャンネルの働きによって内向の電流が打ち消されるため、電位感受性Naチャンネルが開きにくい状態が続く。この期間を相対不応期と呼ぶ。相対不応期の間は、絶対不応期の場合と同様に、静止状態に比べて閾値電位が高くなったように見える。

しかし、これらの閾値電位の上昇は見かけのものであることに注意する必要がある。相対不応期においても、発火が起こらないのは、電位感受性Naチャンネルの近傍の膜電位がチャンネルの持つ閾値電位に達していないためである。実際には、そのようなマイクロな領域の電位は観測することができず、マクロな膜電位が静止状態の閾値電位より高くなったように観測されるのである。

なお、ここでは述べていないが、細胞膜を流れるNa⁺やK⁺のイオン電流についても詳しいことがわかっている、**Hodgkin-Huxleyモデル**[9]はこのレベルでモデル化されており、4つの連立微分方程式によって表される。

しかし、電位感受性Naチャンネルの閉じる速さは、リガンド感受性チャンネル

の閉じる速さに比べて無視できる程速いことから、パルスは無時刻的に発生するものとしても構わないと考えられる。従って、次節ではニューロンのモデルについて述べるが、パルスレベル以降の抽象度をもつモデルのみ触れることにする。

モデル化における留意点 以上、現実のニューロンの振る舞いについて述べてきた。これをモデル化するには、以下の点に注意する必要がある。

発火とチャンネル 発火によって変化するチャンネルは電位感受性の2つである。

チャンネルの速さ 電位感受性チャンネルはリガンド感受性チャンネルに比べて速く閉じる。

抑制性伝達 膜電位の上昇を抑制するもの。膜電位を下降させるのではない。

2.1.2 ニューロン・モデルの抽象度

前節で述べた現実のニューロンに対するモデルは、その抽象度により、おおよそ2種類のモデルに分類できる。1つは、前節で述べたパルスの発生過程レベルでモデル化したものであり、もう1つは、パルスの頻度レベルでモデル化したものである。前者を総称して **Pulse-based Neuron Model**(以下 **Pulse-based Model** とする)、後者を総称して **Rate-based Neuron Model**(以下 **Rate-based Model** とする) と呼ぶ。

Pulse-based Model 前節で述べたように、現実のニューロンの動作は、パルスをその基礎とする。すなわち、パルスの時系列を入力とし、パルスの時系列を出力する。パルスを入出力とするため、前節で述べたパルスの発生過程も含めてモデル化する必要がある。このパルスの発生過程の計算のため、その計算量は、後に述べる **Rate-based Model** よりも多い。

Pulse-based Model ではパルスの発生過程も考慮することから、当然、発火の定常状態から定常状態への変化は、時間をかけて起こることになる。この発火状態の変化が時間をかけて起こることが、**Pulse-based Model** の特徴である。

Rate-based Model **Rate-based Model** は、発火頻度、すなわち、平均発火率が何らかの情報を表現しているとする考え方に基づいてモデル化されたものである。この考え方は、Adrian[1]によって初めて示された。Adrianは、筋肉における伸縮受容器のニューロンの発火頻度が筋肉に注がれる力に関係することを示している。

Rate-based Modelに基づく計算は、現実のニューロンに対してある一定頻度の入力パルスが無限に与えられた場合の、出力パルスの頻度の極限值を直接に求めることに相当する。すなわち、Rate-based Model モデルは、個々のニューロンの時間発展過程を無視し、その定常状態のみに着目したものとすることができる。

すなわち、Rate-based Model の入出力は定常状態における頻度であり、非定常状態を考慮しないことから、定常状態から定常状態への変化が無時刻的に起こるモデルである。このように、個々のニューロンの時間発展過程を無視し、計算を行わないことから、Rate-based Model の計算量は Pulse-based Model よりも少ない。

Rate-based Model の代表例として、出力関数にステップ関数を用いた、McCulloch & Pitts の形式ニューロンモデルが挙げられる。最近では、誤差逆伝播法を用いて結合重みを調整する際に微分を容易にするため、出力関数にシグモイド関数のような、単調非減少関数が用いられることが多い。

以上まとめると、ニューロンのモデルには、Pulse-based Model と Rate-based Model が存在し、それぞれの特徴は以下に示す通りである。

Pulse-based Model 定常状態から定常状態への変化が時間をかけて起こる

Rate-based Model 定常状態から定常状態への変化が無時刻的に起こる

2.2 記憶

大脳生理学の進展により、脳における記憶のしくみは、ある程度わかってきている。そこで以下では、大脳生理学研究によって明らかになった、脳における記憶のしくみについて説明する。また、記憶には短期記憶と中長期記憶が存在することが知られているが、その内、我々は短期記憶とループの関係について考察したので、それについて説明する。

2.2.1 脳における記憶のしくみ

記憶は大きく分けて2種類存在する。1つは中長期記憶であり、もう1つは短期記憶である。中長期記憶は、幼い頃に見た情景として我々に残る残像や、人や物の名前といった知識に関する記憶である。一方、短期記憶は、電話を掛ける時に番号を覚えておくような、数字の羅列といった、我々がそれ自体に意味を見出せない情報に関する記憶である。このような心理学レベルの状態を、マ

クロな状態と呼ぶことにする。

一方、脳におけるマイクロな状態には、電気的狀態と結合重みの2つが存在する。電気的狀態、すなわち、発火状態は、2.1.1 で述べたメカニズムによって変化する。結合重みは、そのシナプスに同じ刺激(パルスの時系列)が続くことにより、次に同じ刺激に来た時には伝達し易いように変化することが知られており、この重みの変化、保持が中長期記憶を担っている [15]。

中長期記憶が変化しない、すなわち、結合重みが変化しない十分短い時間を考える。その時間においても、短期記憶や意識は変化することから、短期記憶や意識は結合重みに担われていない、と言える。

前述のように、脳に存在するマイクロな状態は結合重みと発火状態である。従って、短期記憶や意識が結合重みに担われていないのであれば、それらを担っているものは発火状態である。

以上まとめると、記憶には中長期記憶と短期記憶の2つが存在し、これらを担うものは、それぞれ結合重みと発火状態である。次節では、この内、短期記憶について、ループとの関係を考察する。

2.2.2 短期記憶とループ

短期記憶が行えるだけの短期間のみ刺激が与えられ、以降は刺激がない場合を考える。このような場合でも、我々が望むのであれば、短期記憶や意識を任意の時間保つことができる。このような場合にループのないニューラルネットが発火状態を保つのであれば、その保つ時間分の遅延が必要となる。とすると、脳には任意の長さの遅延が存在していることになり、これは不合理である。それよりも、ループが存在して、それにより発火状態を保っていると考える方が合理的である。

そこで、ループの発火状態が短期記憶や意識に対応すると我々は考える。これらの対応関係について考える。まず、重みが変化しない十分短い時間において、ループの発火状態が同じで、短期記憶と意識が異なるというケースについて考える。この場合、マイクロな状態は全て変わっていないので、マクロな状態も同じ筈である。よってこのようなケースは有り得ない。一方、ループの発火状態は異なるが、短期記憶と意識が同じ、というケースは考えられる。従って、ループの発火状態と、短期記憶や意識というマクロな状態が1対1対応するかどうかは不明である。

以上のように、ループの発火状態と短期記憶や意識には密接な関係があると考えられる。次節からは、ループをもつニューラルネットであるリカレントネットについて説明する。

2.3 リカレント ネット

ニューラルネットは、ループの有無によってリカレント ネット と階層型ニューラルネットに分けられる。また、2.1.2で述べたように、ニューロン・モデルが大きく分けて2つ(Pulse-based Model と Rate-based Model) 存在することから、それぞれを用いたニューラルネットが考えられる。

また、リカレントネットにおいては、ループが状態機械におけるループと同様、入力履歴をその状態として代表させることができるため、オートマトンと同様の働きができる。しかも、Pulse-based Modelを用いたリカレントネットのループを流れる信号は発火頻度という連続値で表現されるため、理論的には離散的な状態をもつ Turing 機械よりも能力が高いことが知られている ([6][8])。

以下では、まず次節において、各ニューロン・モデルを用いた場合に、リカレントネットにどのような違いが現れるかについて述べる。続いて、Pulse-based Modelを用いたリカレントネットの能力が証明されているが、この証明は複雑であることから、我々はより簡単な証明を行ったので、その証明を示す。

2.3.1 リカレント ネット

リカレントネットの構成要素であるニューロン・モデルは、2.1.2で述べたように、Pulse-based Model と Rate-based Model に大別される。Pulse-based Model は定常状態から定常状態への変化が時間をかけて起こるモデルであり、Rate-based Model はその変化が無時刻的に起こるモデルである。

従って、Rate-based Modelに基づくニューラルネットでは、各層の定常状態を入力層から順に求めて最終的な出力を得ていることになる。現実のニューラルネットの時間発展をシミュレートするという観点からは、この計算過程は以下のようにとらえることができる、すなわち、各層において十分な時間が経過してその出力が安定した後、はじめてその出力を次の層に与えるのである。

この計算過程は、一見不自然に写るが、階層型ニューラルネットの極限值を求める上では、実際には効率のよい方法である。ニューラルネットがループを持たない場合には、上記の計算方法で求めた結果は、現実のニューラルネットの時間発展を正確にシミュレートして得られた極限值と一致するからである。

しかし、リカレントネットに対しては、この計算過程は全く問題がないというわけにはいかない。リカレントネットの時間発展を正確にシミュレートすると、入力パルスの時間変化が終わらないうちに、それによるループからのパルスの時間変化が始まり得るからである。そのため、Rate-based Modelを用いた計算は、その極限状態においてさえ、現実のニューラルネットとは異なる結果を与える。Rate-based Modelに基づく極限値の計算が正確なシミュレーションと同一の結果を与えるのは、現実のニューラルネットのループに十分な長さの遅延を挿入した場合に限られる。

このことは、単に不自然であるというだけに留まらず、同期の難しさという違いを生じさせる。脳において、例えば右脳で行われた計算結果と左脳で行われた計算結果を統合して出力することが考えられ、その場合、左脳の全ての計算結果と右脳の全ての結果を同時に統合しなければならないだろう。しかし、脳においてどのように同期がとられているかはわかっていない。

Rate-based Modelを用いたリカレントネットでは、Rate-based Modelの計算のため、その動作は離散化された時間の上で実現される。すなわち、ある時刻においてネットワークに入力が与えられ、それに対して、出力の計算が行われ、計算が終了した後に、次の時刻に進む。次の時刻では、1つ前の出力が入力の組にフィードバックされることになる。Rate-based Modelを用いたリカレントネットでは、各層で極限値を求めて次の層に渡すことから、フィードバックされてくるのは極限値である。このためRate-based Modelのリカレントネットでは、層状にするという簡単な方法で同期はとられる。これは、Rate-based Modelを用いたリカレントネットでは、脳における同期の難しさを重視せず、同期がとれたものとして、その先にある問題の解明に焦点を当てているためである。

一方、Pulse-based Modelを用いたリカレントネットでは、パルスレベルでシミュレートが行われることから、現実のニューラルネット同様、入力パルスの時間変化が終わらないうちに、それによるループからのパルスの時間変化が始まることも有り得る。そのため、Rate-based Modelのように単に層状にニューロンを並べただけで同期がとれる訳ではない。同期をとるためには、脳が同期をとっているのであればその同期方法のような、何らかの工夫が必要となる。

以上まとめると、Rate-based Modelを用いたリカレントネットでは、層状にニューロンを並べるという簡単な方法によって同期をとることができ、Pulse-

based Model を用いたリカレントネットでは、脳における同期の難しさと同様と思われる、同期の難しさが存在する。

2.3.2 リカレントネットの能力

Turing 機械には認識できない言語があることが知られている。一方、Pulse-based Model を用いたリカレントネットは、理論的には万能、すなわち、あらゆる言語を認識できることが、Hava [8] らによって証明されている。理論的には、無限精度の実数が扱えるのであれば、ということであり、実際は扱えないためである。

オートマトンにおける言語の認識は、大きく2つのフェーズに分けられる。1つは、入力と現状態から次状態を決定するフェーズであり、もう1つは、その状態が受理状態かどうかを判定するフェーズである。

Hava は、この2つのフェーズがリカレントネットで実現可能であることを証明した。一方、我々は、1つ目のフェーズがリカレントネットで実現可能であることをより簡単に証明した。2つ目のフェーズに関しては、オートマトン研究においても何らかの判定する機構の存在を仮定していることから、そのような仮定を置いた。

我々の証明では、無限状態機械という概念を用いる。これは、状態数が非可算無限個の状態機械である。これが万能であることをまず証明し、続いてそのような機械が必ずリカレントネットで実現できることを示す。

⟨⟨無限状態機械: M の定義⟩⟩ $M = (K, \Sigma, \delta, s_0, F)$

- K 状態記号の \aleph_0 の濃度をもつ無限集合
- Σ 入力記号の有限集合
- δ 状態遷移関数の \aleph_0 の濃度をもつ無限集合: $K \times \Sigma \rightarrow K$
- s_0 初期状態
- F 受理状態の \aleph_0 の濃度をもつ無限集合 ($F \subset K$)

⟨⟨定理⟩⟩ あらゆる言語を認識する無限状態機械が存在する。

⟨⟨証明⟩⟩ 非可算無限個の状態による状態表現とその上の detector を仮定する。ここで言う detector とは、現状態が受理状態集合に含まれるかどうかを判定する機械である。従って、リカレントネットの万能性を示すためには、全ての入力系列が異なる状態に遷移するようにすればよい。異なる状態に遷移すれば、そ

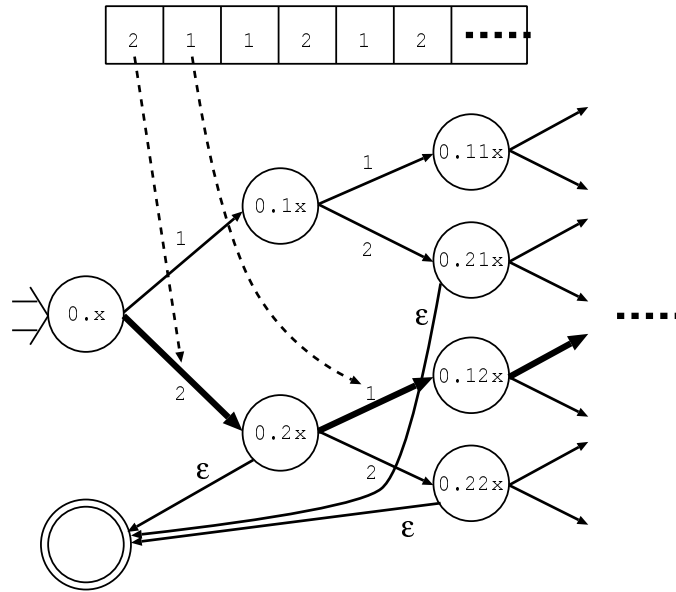


図2: $\{1, 2\}$ 上のあらゆる言語を認識する無限状態機械

れぞれが受理状態であるかどうかを, detector が判定するだけである.

入力記号を $\{1, 2\}$ とし, 図2のような2進木の状態遷移図をもつ無限状態機械を考える. 各状態は, 親の状態に入力記号を右シフトさせて得られるものとする. この無限状態機械は, 明らかに全ての入力系列を異なる状態に遷移させる. 入力が n 記号の時は, 同様に n 進木の状態遷移図をもつ無限状態機械によって認識できる. (証明終わり)

《(定理)》 あらゆる言語を認識するリカレントネットが存在する.

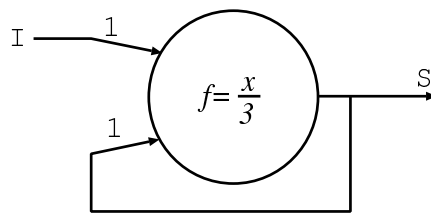


図3: $\{1, 2\}$ 上のあらゆる言語を認識する RNN

このような状態遷移図をもつリカレントネットは図3のようにして構成できる. このリカレントネットは, 明らかに上の右シフトを実現している. 入力が n 記号の時は評価関数を $x/(n+1)$ とすればよい. (証明終わり)

次節からは, このリカレントネットにおいて, 記憶について研究していくた

めの方針を述べる。

2.4 研究の方針

2.2.2で述べたように、記憶、特に短期記憶について考えるのであれば、階層型ニューラルネットを用いるよりもリカレントネットを用いた方がよさそうである。そこで、リカレントネットを用いて研究を行う。

リカレントネットを用いて記憶について考えるために、中長期記憶や短期記憶に当たるものを獲得したリカレントネットを、とにかく作ってみることにする。そのために、リカレントネットに記憶が必要な問題を解かせることにする。

リカレントネットに解かせる、記憶が必要な問題として、迷路問題を用いることにする。迷路問題でも、俯瞰視点によって解くのではなく、迷路内を動くエージェントの視点によって解くものである。つまり、このエージェントをリカレントネットで構成する。

迷路問題を解くのに必要な情報としては、1.)現在の周囲の状況、2.)通って来た道筋などの迷路の形、3.)迷路の解法などのアルゴリズムが挙げられる。この内、記憶に当たるものは2.)と3.)であり、前者が短期記憶に相当し、後者が中長期記憶に相当すると考えられる。

我々は、まず短期記憶について考える。中長期記憶を担う重みは、Rate-based Modelの階層型ニューラルネットにおける誤差逆伝播法のような、何らかの重み調整法により、アルゴリズムを記憶させることができるだろう。一方、そのように構成した場合でも、発火状態が担う短期記憶については考える余地がある。

以降は短期記憶をもつリカレントネットの構成法、すなわち、ニューロン・モデルの選択と重みの調整法について考えていく。

2.5 研究の方法

2.3.1で述べたように、リカレントネットとしてはPulse-based Modelを用いること、Rate-based Modelを用いること、の2つが考えられる。そこで以下では、各リカレントネットについて重みを調整する方法について概観し、Pulse-based Modelを用いる構成法を選択する。続いて、Pulse-based Modelを用いる構成法を詳しく説明する。

2.5.1 ネットワークの構成法

2.3.1で述べたように、リカレントネットとしてはPulse-based Modelを用いること、Rate-based Modelを用いること、の2つが考えられる。そして、そのそれぞれに対し、異なる重み調整法が考えられる。

Rate-based に基づく構成法 Rate-based Model と通時的誤差逆伝播法による構成法

Pulse-based に基づく構成法 Pulse-based Model と何らかの重み調整法による構成法

Rate-based に基づく構成法 Rate-based に基づく構成法は、重みの調整法として通時的誤差逆伝播法を用いる構成法である。通時的誤差逆伝播法は、Rate-based Model の階層型ニューラルネットに対する重み調整法である誤差逆伝播法を、Rate-based Model のリカレントネットに応用したものである。リカレントネットにはループが存在することから誤差逆伝播法を用いることはできない。

そこで、リカレントネットを空間展開する。すなわち、物理的に同じニューロンでも、時刻が異なるものは仮想的に異なるニューロンと見なす。このように見なしてできる仮想的なネットワークにはループが存在しないため、誤差逆伝播法を適用できる。こうして仮想的な重みに対する変化量を求め、その平均をとって物理的な重みの変化量とする。

このように仮想的なネットワークに対し誤差逆伝播法を行うため、全ての仮想的なニューロンの出力が必要となる。つまり、通時的誤差逆伝播法を行うためには、全てのニューロンの過去の出力を覚えておく必要がある。

2.3.1で述べたように、Rate-based Model のリカレントネットでは、ニューロンを層状に並べるといった簡単な方法で、同期をとることができる。これは、脳における同期の難しさを問題視していないためである。

Pulse-based に基づく構成法 Pulse-based Model のリカレントネットには、通時的誤差逆伝播法のような、一般的な重み調整法が存在しない。よって、Pulse-based に基づく構成法をとる場合には、何らかの重み調整法を考える必要がある。

2.3.1で述べたように、Pulse-based Model のリカレントネットでは、ニューロンを層状に並べるといった簡単な方法では、同期をとることができない。これは、脳における同期の難しさを問題視しているためである。

脳においては同期が難しく、また、本当に同期がとられているかどうか、実

際にはわかっていない。本論ではこれを考慮し、Pulse-based に基づく構成法を選択する。そこで次節では、Pulse-based に基づく構成法における重みの調整法について説明する。

2.5.2 Pulse-based に基づくネットワークの構成法

重みを調整する方法の1つとして、

- 1.) Pulse-based Modelで論理素子を構成し、論理素子を用いて順序回路を作る
- 2.) 順序回路であるリカレントネットに何らかの変換を施す

という手順が考えられる。以下では、この手順に従うものとする。

まず、Pulse-based Modelで論理素子を構成する。最低頻度/最高頻度を論理素子のLOW/HIGHに対応させて、デジタルな論理素子を構成する。そして、この素子を用いて順序回路を構成する。

こうしてできたリカレントネットは、順序回路であるため、ニューロンの発火状態は、多くの時間、デジタルな発火状態である。このようなリカレントネットは脳の構造には程遠いことから、リカレントネットに対し何らかの変換を施すようにする。

本論では、1.)までの構成を行い、迷路を解く順序回路をPulse-based Modelを用いたリカレントネット上に実現した。Pulse-based Modelで論理素子を構成し、迷路を解くアルゴリズムを考え、それを順序回路で実現した。

前節でも述べたように、Pulse-based Modelを用いたリカレントネットでは同期が難しい。脳において同期をとっているかどうか自体がわからないが、非同期式順序回路を構成することは難しいので、同期式順序回路を構成することにした。次章からは、その構成を述べていく。その構成を説明する前に、我々と同じく、迷路を解くりカレントネットを構成する研究を谷([10][11])が行っている。彼の構成法を述べ、その違いを明らかにしておく。

2.6 関連研究

谷は、意識の解明を目的とし、行動計画中の意識について研究を行っている。生理学的な学習過程に関する仮説に基づいた学習により、ロボットに中長期記憶を獲得させ、ロボットがそれを用いて行動計画を行っている際の意識について研究する。このための問題として、ロボットの視点で迷路を解かせることを行った。ロボットにはリカレントネットが備えられており、ニューロン・モデ

ルには Rate-based Model が用いられている。

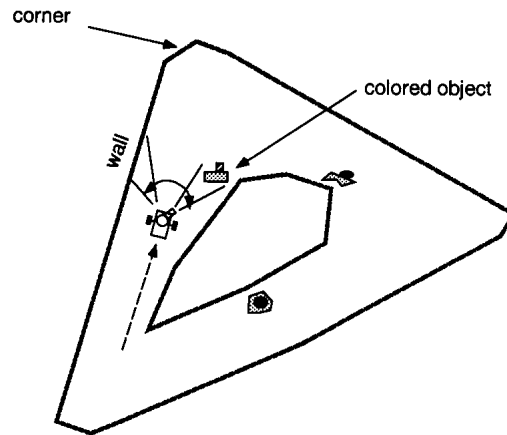


図4: ロボットの動く環境

ロボットが動き回る環境を図4に示す。環境には、壁と障害物が存在している。障害物の位置を色々変えることで、環境を変えることができるようにする。ロボットのリカレントネットへは、目に見える範囲内の、壁や障害物までの距離情報が入力される。

リカレントネットの行うタスクは、次時刻における入力を予測することである。リカレントネットが、次時刻における最も開けている方向、すなわち物体までの距離が遠い方向を予測し、ロボットはその方向に進む。物体までの距離が同じくらい遠い方向が複数存在する場合は、リカレントネットの予測の段階で、1方向を選択する。

前述したように、中長期記憶の生理学的な学習過程に基づいて学習させたいことから、リカレントネットの重みの調整には、2.5.1で述べた、通常の時系列的誤差逆伝搬法を用いない。通常の時系列的誤差逆伝播法を行うためには、全てのニューロンの過去の発火パターンを覚えておく必要があった。これは、以下に述べる生理学的な学習過程とはあまりに似ていない、と谷は述べている。

生物では、海馬が中長期記憶の1つであるエピソード記憶を一時的に蓄えていると考えられており [17]、また、一時的に蓄えられた記憶を変換して新皮質に蓄えさせるのは睡眠中に行われていると考えられている。さらに最近では、活動中に要求されたエピソード記憶に関する情報を取り出すために、新皮質の情

報を変換するのも海馬が行っていることを示す証拠 [18] が挙げられている。そこで McClelland [13] は、海馬が新皮質のトレーニングにおける教師の役割を担っており、また新皮質に記憶された中長期記憶のデコードを行っているとする仮説を立てた。

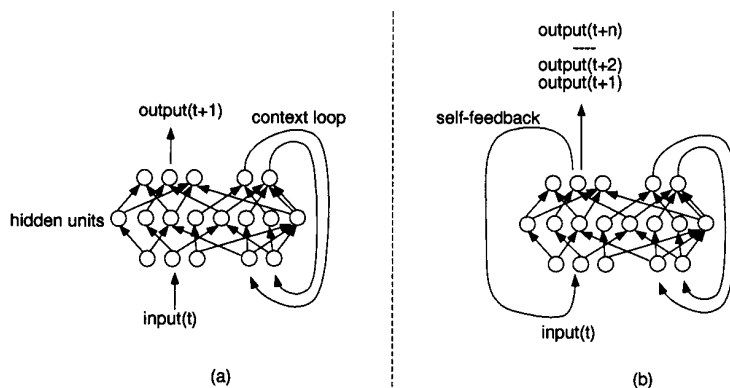


図5: prediction part の重みの調整

この仮説に基づき、prediction part のリカレントネットの重みを調整する。活動中、すなわち迷路内を動くフェーズ (図 5(a)) では重みの調整は一切行わず、その時の入力系列をどこかに保存しておく。重みを調整するフェーズでは、ロボットは迷路を動くことをやめ、リカレントネットの出力をリカレントネットの入力に接続して (図 5(b)) 通時的誤差逆伝播法を行う。すなわち、外部入力を断ち、リカレントネット自身が予測した結果で次の予測を行うことで、睡眠中であるような状態を作り出す。その時の教師信号としては、活動中に集めた入力系列を用いて重みの更新を行う。この2つのフェーズを繰り返すことで、迷路を解けるようにする。

谷は、Rate-based Model と生理学的な学習過程に関する仮説を基にした重み調整法により、迷路を解くリカレントネットを構成している。重み調整法における実際の重み調整は、通時的誤差逆伝播法が行っている。

彼の構成法では、教師信号は環境から得られた入力系列である。つまり、次の入力を正しく予測させることが目的である。このため、こうしてできたリカレントネットは、その重みで、つまり中長期記憶で迷路の形を覚えていることになる。

一方、このリカレントネットでは、我々が考える短期記憶、すなわち、ループの状態には、何が記憶されているのかはよくわからない。迷路の形を重みが覚えているのであれば、活動中におけるループは迷路内の現在地を表しているであろう。しかし、これが本当であるかどうかは、よくわからない。

短期記憶は通った経路の情報であり、中長期記憶は迷路を解くアルゴリズムに相当するだろうと、2.4において我々ははっきりと示した。そして、このアルゴリズムに当たる情報を重みが獲得するように、構成法を考えた。

彼の構成法は、そのような、短期記憶とループの状態の対応をはっきりさせていないため、現在の構成法で得られるリカレントネットのループの状態が何を表現しているのかは、よくわからない。短期記憶について考えるためには、構成法を考える際に、迷路問題における短期記憶について意識しておく必要があるだろう。

第3章 リカレント シンプルスパイクングニューラルネットワークを用いた状態機械の実現

Pulse-based Model にはパルス駆動型ニューロンモデル (以下パルス駆動型モデルとする)[16] とシンプルスパイクングニューロンモデル (以下シンプルスパイクングモデルとする)[12] が存在する. そこで次節では, まず各モデルを説明する. そして, モデルの比較を行い, シンプルスパイクングモデルを用いる理由について述べる. 続く 3.2 では, シンプルスパイクングモデルで論理素子を実現する方法について説明する. なお, 節末において, ここで実現した論理素子を用いて順序回路を構成する際の注意点についても述べる. 最後に, それを用いて実現した迷路を解く同期式順序回路の構成について述べる.

3.1 パルス駆動型モデル vs. シンプルスパイクングモデル

Pulse-based Model にはパルス駆動型モデルとシンプルスパイクングモデルの2つが存在する. そこで以下では, 各モデルを説明する. また, 両モデルの違いが入出力特性によく現れていることから, これについて詳しく説明する. そして, 2.1.1 で述べた現実のニューロンの振る舞いや, 入出力特性両の点から両モデルを比較する.

3.1.1 パルス駆動型モデル

パルス駆動型モデルは, 膜電位, 閾値電位と呼ばれる2つのパラメータをもち, これらの関係によってニューロンの発火が決まるモデルである. ニューロン i が以前に発火してから絶対不応期を経過した時刻を \hat{t}_i とする. 時刻 $t \geq \hat{t}_i$ におけるニューロン i の膜電位 $P_i(t)$, 閾値電位 $H_i(t)$ は, 次式によって与えられる.

$$P_i(t) = P_{i\infty} + (P_{i0} - P_{i\infty})e^{-(t-\hat{t}_i)/T_p} + \sum_{j \in \Gamma_i} w_{ij} \sum_{t_j^{(f)} \in F_j} e^{-(t-t_j^{(f)})/T_p}$$

$$H_i(t) = H_{i\infty} + (H_{i0} - H_{i\infty})e^{-(t-\hat{t}_i)/T_h}$$

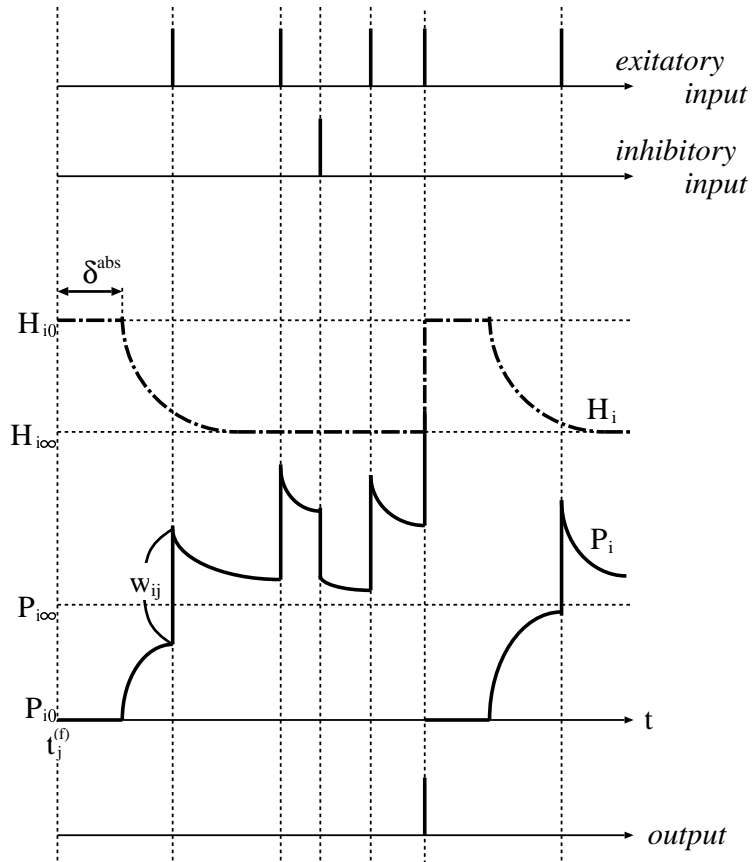


図 6: パルス駆動型モデル

- P_{i0}, H_{i0} 初期リセット膜電位, 初期リセット閾値電位
- $P_{i\infty}, H_{i\infty}$ 静止膜電位, 静止閾値電位
- T_p, T_h 膜電位, 閾値電位の時定数
- w_{ij} ニューロン $j-i$ 間の結合重み
- Γ_i シナプス前細胞の集合
- F_j ニューロン j の発火時刻の集合
- $t_j^{(f)}$ ニューロン j の発火時刻
- \hat{t}_i ニューロン i が以前発火してから絶対不応期 δ^{abs} を経た時刻

$P_i(t), H_i(t)$ は, 通常, それぞれの初期値 P_{i0}, H_{i0} から始まり, それぞれの収束値 $P_{i\infty}, H_{i\infty}$ に収束しようとする. ただし, $P_{i0} < P_{i\infty}, H_{i0} > H_{i\infty}$ である. T_p, T_h は, それぞれの時定数である.

図 6 にパルス駆動型モデルの動作を示す. $P_i(t)$ は, ニューロン j からパルスが到着すると, その結合重み w_{ij} に応じて変化する. 抑制性入力, 結合重み

を負にすることで表現される．興奮性パルス の到着時に $P_i(t)$ が $H_i(t)$ を越えれば，ニューロンは発火する．発火後， $P_i(t)$ ， $H_i(t)$ は，絶対不応期 δ^{abs} を経て，それぞれ，初期値 P_{i0} ， H_{i0} にリセットされる．後電位は P_{i0} の低さによって，相対不応期は H_{i0} の高さによって表現される．

インパルスの発生は，他の事象に比べて非常に短い時間で起こるので，その幅は通常無視できる．すなわち，電位感受性 Na チャネルの働きは無視できる． T_P ， T_H は，それぞれ，リガンド感受性 Na チャネル，電位感受性 K チャネルによって決まる．

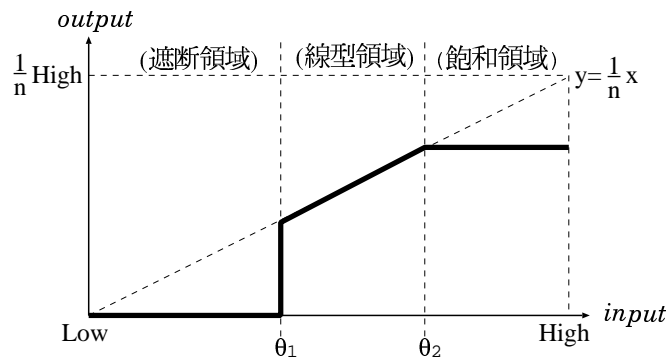


図7: パルス駆動型モデルの入出力特性

入出力特性 図7に，パルス駆動型モデルの入出力特性を示す．図中，横軸が入力頻度，縦軸が出力頻度である．各 θ_i は，入力枝の重み，膜電位，閾値電位の時定数，及び絶対不応期の長さによって決まる．以下では，各領域について詳しく見ていく．

入力頻度が低い場合(図9)には，次のパルスの到着までに $P(t)$ は実際上0に戻ってしまい，ニューロンは発火しない．この区間を遮断領域と呼ぶことにする．

一方，入力頻度が非常に高い場合(図8)でも，絶対不応期のため，出力頻度はある一定値以下に抑えられる．この区間を飽和領域と呼ぶことにする．

遮断領域と飽和領域の間には，入力と出力の関係が線型となる線型領域が存在する．図9, 10に線型領域における $P(t)$ ， $H(t)$ の変化を示す．入力頻度を θ_1 から微増させた場合，図10に示すように， n ($n \geq 2$) 回目のパルスで発火することになる．入力頻度を更に増加させていった場合には，出力頻度は，(入力頻

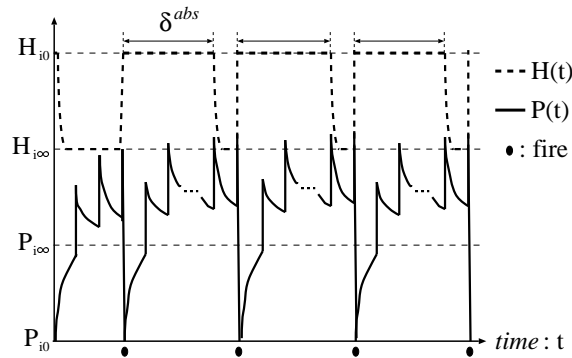


図8: 入力頻度 $x \geq \theta_2$

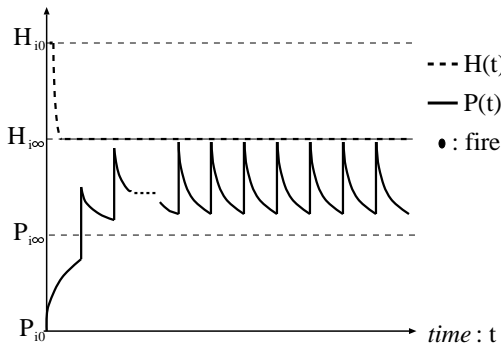


図9: 入力頻度 $x = \theta_1 - \epsilon$

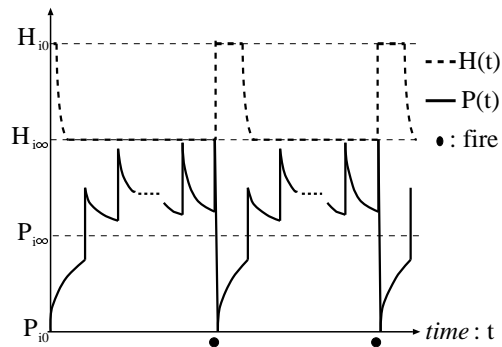


図10: 入力頻度 $x = \theta_1 + \epsilon$

度) / n ($n \geq 2$) の直線で与えられる。

この定数 n は、以下に示す理由により、 $n \geq 2$ となる。ある入力頻度において、発火直後の入力で発火する結合加重を与えよう。それより低い入力頻度では、パルス到着時の $H(t) - P(t)$ はより小さい値となるから、その一回の入力によって発火が起こるようになる。すなわち、遮断領域が存在しなくなり、飽和特性が失われる。従ってパルス駆動型モデルの出力頻度は、最大でも入力頻度の $1/2$ となる。これは、パルス駆動型モデルでは発火後ニューロンの状態がリセットされるためである。

以上まとめると、パルス駆動型モデルの入出力特性は遮断領域、線型領域、飽和領域に分けることができる。そして、線型領域における直線の傾きは $1/n$ 以下となる。このことは、パルス駆動型モデルでは出力頻度が入力頻度以上にはならないことを示している。すなわち、パルス駆動型モデルには出力レベルが低下するという特徴がある。

3.1.2 シンプルスパイクモデル

シンプルスパイクモデルは状態と閾値電位という2つのパラメータをもち、これらの関係によって発火が決まるモデルである。この状態とは、ニューロンにおけるNaコンダクタンス、及びリガンド感受性KチャンネルがKコンダクタンスに与える影響を考慮したものである。時刻 t におけるニューロン i の状態 $u_i(t)$ 、閾値電位 $H_i(t)$ は、次式によって与えられる。

$$u_i(t) = \sum_{j \in \Gamma_i} \sum_{t_j^{(f)} \in F_j} w_{ij} \epsilon_{ij}(t - t_j^{(f)})$$

$$H_i(t) = \vartheta - \eta_i(t - \hat{t}_i)$$

- ϑ 閾値
- w_{ij} ニューロン $i - j$ 間の結合重み
- $\epsilon_{ij}(x)$ リガンド感受性チャンネルの閉門による電位減少関数
- $\eta_i(x)$ 抑制性の電位感受性チャンネルの閉門による電位減少関数
- Γ_i シナプス前細胞の集合
- F_j ニューロン j の発火時刻の集合
- $t_j^{(f)}$ ニューロン j の発火時刻
- \hat{t}_i ニューロン i の以前の発火時刻

図11にシンプルスパイクモデルの動作を示す。状態 u_i は、ニューロン j からパルスが到着すると、その結合重み w_{ij} に応じて変化する。抑制性入力、結合重みを負にすることで表現される。時刻 t において、状態 $u_i(t)$ が閾値電位 $H_i(t)$ を上回ると、ニューロン i は発火する。パルス駆動型モデル同様、パルスの発生過程は無時刻的に起こるものとするが、パルス駆動型モデルとは異なり、発火によって状態 u_i がリセットされることはない。

時刻 t におけるニューロン i の状態 $u_i(t)$ は、全てのシナプス前細胞で現在までに発生したパルスが状態に与える影響の総和で表現される。 $w_{ij} \epsilon_{ij}(t - t_j^{(f)})$ は、シナプス前細胞 j で時刻 $t_j^{(f)}$ に発生したパルスが、ニューロン i のリガンド感受性のNa($w_{ij} > 0$)またはK($w_{ij} < 0$)チャンネルを開くことに相当する。通常、 $\epsilon_{ij}(x)$ には、 $\epsilon_{ij}(x) = \exp(-x/\tau)$ のような単調減少関数が用いられる。

閾値電位 $H_i(t)$ は、2.1.1におけるニューロンの閾値電位及び電位感受性のKチャンネルの影響を表しており、ニューロン i が以前に発火した時刻により決ま

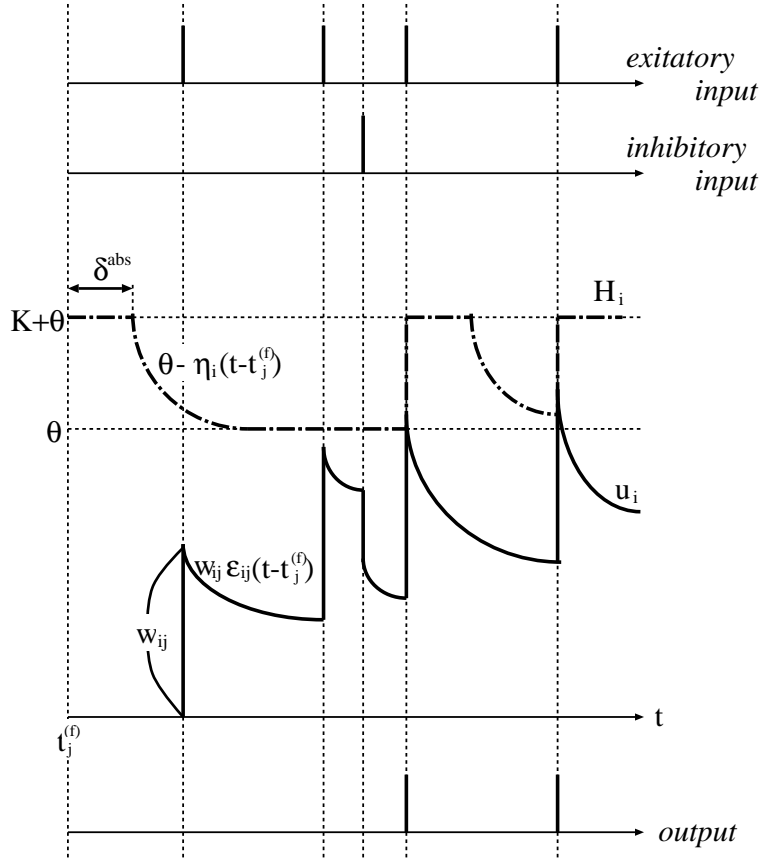


図11: SNNMにおける電位の変化の様子

る. ϑ は 2.1.1 におけるニューロンの閾値電位に相当する. $\eta_i(t - \hat{t}_i)$ は, 以前ニューロン i が発火した時刻 \hat{t}_i からの電位感受性の K チャンネルが与える影響である.

$\eta_i(x)$ には, 絶対不応期を考慮するものとしがないものがあるが, 我々は絶対不応期を考慮することにする. 絶対不応期を考慮した $\eta_i(x)$ は以下のように書ける.

$$\eta_i(x) = \eta_0 \exp\left(-\frac{x - \delta^{abs}}{T}\right) H(x - \delta^{abs}) + K H(x) H(\delta^{abs} - x)$$

δ^{abs} は絶対不応期の時間を表し, K は絶対不応期中の閾値電位 (無限大) を表す. $H(x)$ は $x \leq 0$ の時 0, $x > 0$ の時 1 を出力するステップ関数である.

入出力特性 図12 に, シンプルスパイクモデルの入出力特性を示す. 各 θ_i は, 入力枝の重み, 状態, 閾値電位の時定数, 閾値 ϑ , 絶対不応期の長さによって決まる. 遮断領域, 及び飽和領域に関してはパルス駆動型モデルと同様であるため, 以下では線型領域に関して説明する.

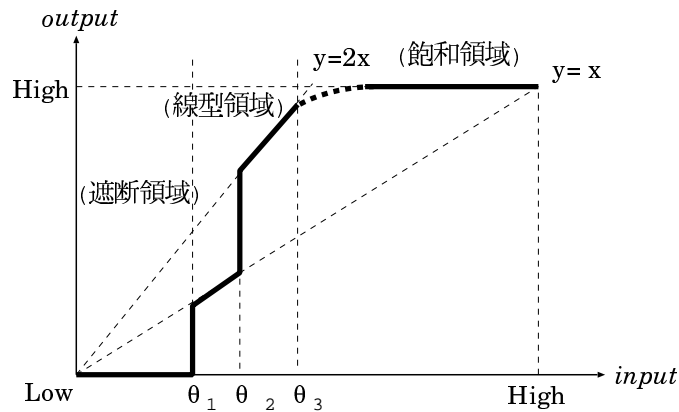


図12: シンプルスパイクングモデルの入出力特性

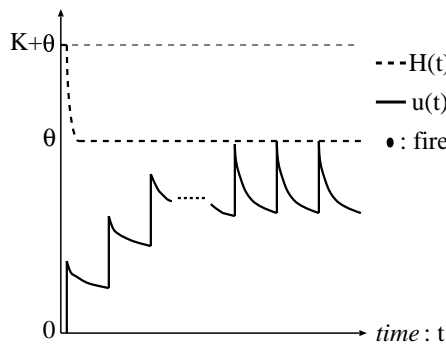


図13: 入力頻度 $x = \theta_1 - \epsilon$

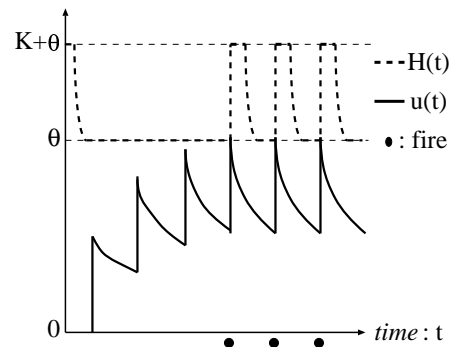


図14: 入力頻度 $x = \theta_1 + \epsilon$

パルス駆動型モデルと同様に遮断領域と線型領域の境界となるある頻度 θ_1 が存在する. それよりもわずかに頻度が低い場合は, ニューロンは図13のような状態にある. つまり, 現在の頻度では, 次にパルスが来て状態 u が押し上げられても, ぎりぎり閾値電位には届かないような状態である.

入力頻度が θ_1 よりもわずかに高くなると, 図14のように, 入力パルス1つにつき1回発火するようになる. 従って, 入力頻度が θ_1 を超えると, 入力頻度 x と出力頻度 y の関係は, $y = x$ となる.

同様にして, ぎりぎり $y = x$ で発火する頻度 θ_2 というものも考えられる. それを超えると, 図15のように, 1回のパルスで2回発火するようになるため, 入出力の関係は $y = 2x$ となる.

すなわちシンプルスパイクングモデルの線型領域では, 入力頻度を高くすると次々と傾きの大きな直線にのっていく. そして最終的には, 入力頻度が θ_3 を超えた時点で, 出力は一定となる. $y = 2x$ にのった後, さらに傾きの大きな直

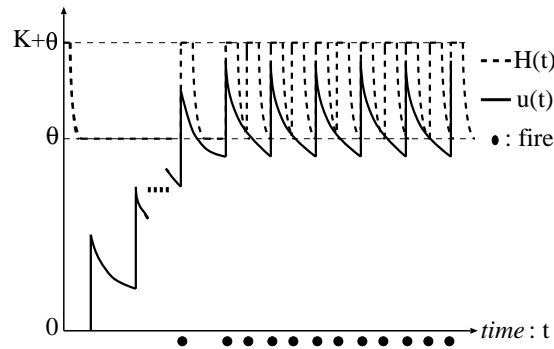


図15: 入力頻度 $x = \theta_2 + \epsilon$

線 $y = 3x$ や $y = 4x$ にのる可能性もあるが、具体的にはわからないため、図12では点線によって表現している。

以上まとめると、シンプルスパイクングモデルの入出力特性は遮断領域、線型領域、飽和領域に分けることができる。そして、線型領域における直線の傾きは n 以上となる。このことは、シンプルスパイクングモデルでは出力頻度が入力頻度以上にもなり得ることを示している。すなわち、シンプルスパイクングモデルには出力レベルが低下するという特徴はない。

3.1.3 モデルの比較

パルス駆動型モデルは、以下の点において、2.1.1で述べた、現実のニューロンの『正しい』モデルとは言い難い：

- 発火後膜電位がリセットされる。これは、発火によってリガンド感受性Naチャンネルも閉じられることを意味する。現実のニューロンでは、リガンド感受性チャンネルは発火の影響を受けない。
- パルス駆動型モデルでは、各チャンネル、特に、電位感受性チャンネルとリガンド感受性チャンネルの時定数の違いを表現することができない。
- 現実のニューロンでは、膜電位が低い状態での抑制性入力、膜電位を静止電位より降下させるのではなく、以降の興奮性入力を抑制する働きを持つ。パルス駆動型モデルでは、過去の抑制性入力の蓄積を表現することができない。

そして実際に前節までの結果から、シンプルスパイクングモデルとパルス駆動型モデルとは有意に異なる挙動を示すことが分かった。また、シンプルスパイクングモデルでは、パルス駆動型モデルより有意に計算量が多いということはない。従って、Pulse-based Modelにおけるパルス駆動型モデルの意義はないと

結論づけることができる。

よって、シンプルスパイクングモデルを用いて、Pulse-based に基づく構成法により、迷路を解くりカレントネットを構成する。次節からは、シンプルスパイクングモデルを用いた、論理素子と順序回路の設計について述べる。シンプルスパイクングモデルの出力レベルが低下しないという性質は、論理素子の設計において、非常に都合がよい。

3.2 論理素子の構成

2.5.2で述べたように、シンプルスパイクングモデルを用いたリカレントネット(以下リカレントシンプルスパイクングニューラルネットワーク)で順序回路を実現するために、シンプルスパイクングモデルで論理素子を構成する。最低頻度 / 最高頻度を論理素子の LOW / HIGH に対応させて、デジタルな論理素子を構成する。

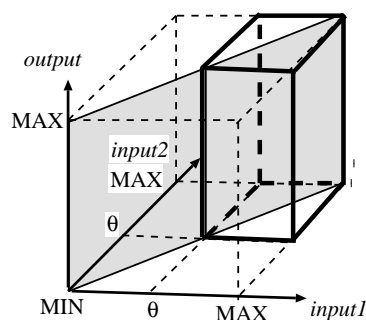


図 16: 理想的な 2 入力 AND の入出力特性

AND ニューロン まず、AND 素子について考える。2 入力 AND の理想的な入出力特性は、図 16 のようなものである。図の立方体における底面の 2 本の軸が 2 つの入力頻度を表し、高さ方向の軸が出力頻度を表している。電気的な AND 素子では、基本的な信号の強さは HIGH(V_{cc}) と LOW(0) の 2 つであり、その間の値は立ち上がり / 立ち下がりにおいて一瞬示すだけである。しかし、シンプルスパイクングモデルでは最低頻度から最高頻度までの何如なる値も長期に渡って取り得ることから、図では間の区間も描いている。頻度が θ を超えなければ LOW, 超えたら HIGH と見なすものとする。MIN/MAX は、LOW/HIGH のそれぞれ最小値 / 最大値を表す。

入力線が2本のニューロンの入出力特性を図16のような形に近づけることを考える。2入力のニューロンの入出力特性を考えるのは難しいので、2本の入力頻度は等しいものとして、1入力で考える。すなわち、図の斜線の(対角線の)平面における入出力特性のみを考える。2本の枝の重みは、ニューロンの入力を1本と考えるとANDニューロンを作り、その時の重みを2等分してやればよい。

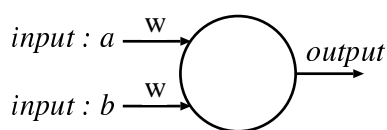


図17: 重みが等しいニューロン

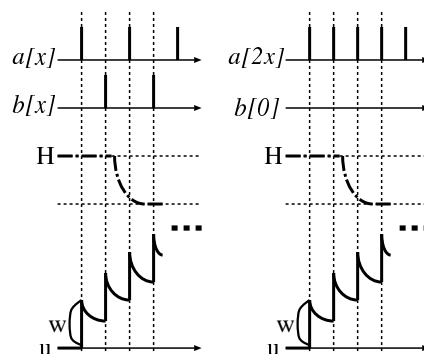


図18: $a, b = x$ と $a = 2x, b = 0$ の場合の状態 u の推移

このように考える場合に、注意しなければならないことが1つある。それは、2入力 a, b の重みが等しいニューロン(図17)において、 a, b の入力頻度が共にある頻度 x である場合と、 $a = 2x, b = 0$ の場合とを区別できないということである。これは、 a と b の入力パルスの位相が丁度半分ずれている場合の状態 u の推移(図18左)と、 a に入力パルスが2倍の頻度で入ってくる場合の状態の推移(図18右)を比較してみれば、明らかである。

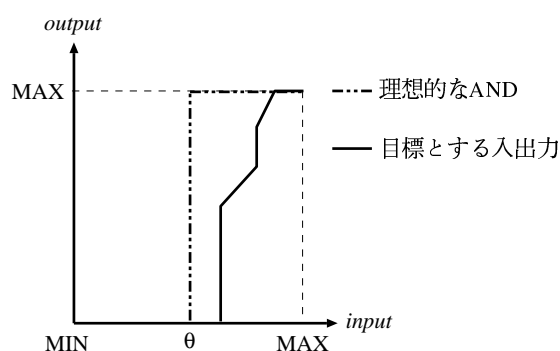


図19: 対角線上の入出力特性 (AND)

図16の斜線の平面をとりだしてきたのが、図19である。横軸が全ての入力頻度が等しい時の入力1本あたりの頻度であり、縦軸が出力頻度である。理想としては、 θ はMAXとMINの丁度中間、すなわちMINを0とする時 $\theta = \text{MAX}/2$ (図の破線)となるようにしたい。しかし、前述のようにニューロンでは $\text{input1} = \text{MAX}/2, \text{input2} = \text{MAX}/2$ の場合と $\text{input1} = \text{MAX}, \text{input2} = 0$ の場合を区別できないため、このように作った場合、入力の1つがHIGHでも出力がHIGHのようになり、ANDの働きをしなくなる。よって、ANDとして働かせるためには、閾値 θ はなるべく図の右に寄せる($\text{MAX}/2 < \theta < \text{MAX}$, 図の実線)ようにする。

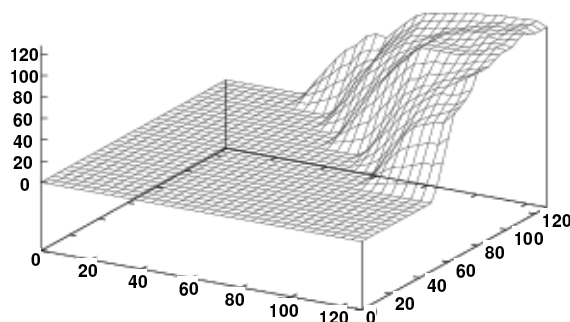


図20: AND ニューロン

図12に示す1入力のシンプルスパイクングモデルの入出力特性を、重み、時定数を調整することで、図19の実線のような入出力特性を示すようにする。こうしてできたニューロンの重みを2等分した、2入力のニューロンの入出力特性を図20に示す。図においてはMAXを128とし、MINを0としている。このような入出力特性を示すニューロンをANDニューロンと呼ぶことにする。

OR ニューロン ORについても同様に考える。2本の入力等しいものと考え、理想的なORの入出力特性は図21の破線ようになるが、ANDと同様、 $\theta = \text{MAX}/2$ とするのは好ましくない。ORは、複数入力の内1つでもHIGHのようになった時に、出力が必ずHIGHのようになって欲しい。よって、 θ はなるべく図の左に寄せる($\text{MIN} < \theta < \text{MAX}/2$, 図の実線)ようにする。

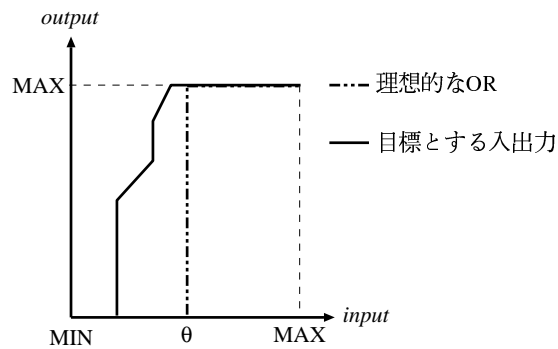


図 21: 対角線上の入出力特性 (OR)

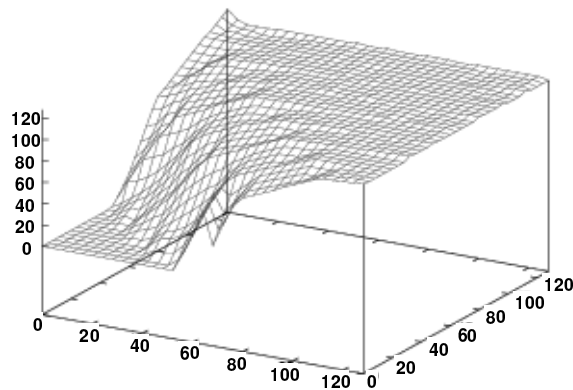


図 22: OR ニューロン

このような入出力特性を示すように 1 入力のニューロンの重み, 時定数を調整し, できたニューロンの重みを 2 等分した, 2 入力のニューロンの入出力特性を図 22 に示す. このような入出力特性を示すニューロンを OR ニューロンと呼ぶことにする.

否定包含ニューロン 一方 NOT であるが, NOT 素子は入力が LOW である時 HIGH を出力し, 入力が HIGH である時 LOW を出力するものである. これをニューロンに当てはめると, NOT ニューロンは次のようなものであると言える.

- 入力パルスが来ない (低頻度) 時, 高頻度で発火する
- 入力パルスが高頻度の時, 発火しない.

1 つ目の性質を満たすためには, 閾値電位の収束値である θ が 0 でなければならないことが, 容易に想像がつく. また $\theta = 0$ の時, 2 つ目の性質を満たすためには抑制性入力, すなわち入力枝の重みが負である必要がある.

x	y	$\overline{x \supset y}$
0	0	0
0	1	0
1	0	1
1	1	0

表1: 否定包含の真理値表

このようなニューロンは、実際のニューロンの振舞いからはかけ離れてしまっている。2.1.1 で述べたように、一般に抑制性入力とは興奮性入力の影響を打ち消す働きをするものであり、興奮性入力とセットになっている。抑制性入力のみニューロンは、一部の例外を除いて存在しない。

その例外とは、バースト発火ニューロンと呼ばれるニューロンである。バースト発火ニューロンは、常に神経伝達物質を放出しているニューロンであり、そこに抑制性入力がある場合は、その放出量を下げる働きをする。しかし、このニューロンは多数存在する訳ではなく、極く少数発見されている ([3][4][14]) だけである。

仮にNOTニューロンがこのバースト発火ニューロンをモデル化したものと捉えたとしても、NOTニューロンを用いて構成したニューラルネットは脳のような構造からはかけ離れてしまっている。順序回路のニューラルネットにはNOT数分のバースト発火ニューロンが存在することになる。順序回路を作った後に行う何らかの変換により、幾つかのバースト発火ニューロンが壊れ、通常の抑制性入力が興奮性の働きを抑えるニューロンになる可能性がない訳ではないが、その確率は低いと考えられる。よって、バースト発火ニューロンを用いるならば、実際の脳にならい、少数に限定した方がよい。

以上の理由により、我々はNOTニューロン、すなわち入力が1本でそれは抑制性であるようなニューロンを用いない。よって、順序回路をリカレントネットを実現するためには、NOTに変わる素子をニューロンで実現する必要がある。我々はそれを否定包含素子によって実現する。

否定包含の真理値表を表1に示す。論理学では $0 \supset 1$ を1とし、 $1 \supset 0$ を0としている。 $0 \supset 0, 1 \supset 1$ は共に1である。表のような素子を実現できれば、 x をHIGHにセットしておくことで、NOTと同様の結果が得られる。HIGHの信号

としては、バースト発火ニューロンを1つ用意しておけばよい。

このような真理値表をもつニューロンを実現する。 x のみがHIGHの時HIGHを出力しなければならないことから、入力枝 x の重みはORニューロンの重みと一緒にあることが想像できる。また、 x, y が共にHIGHである時LOWを出力することから、入力 y は抑制性であり、 y の重みとしてはORニューロンの重みを正負反転させたものを用いればよさそうである。

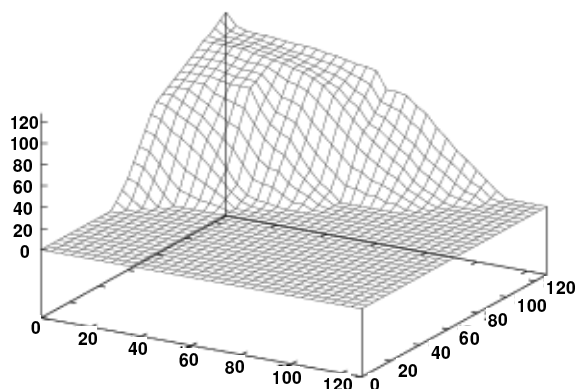


図 23: 否定包含ニューロン

このようにして、2入力ORニューロンの入力の内、1本の枝の重みの符号を反転させたニューロンの入出力特性を図 23 に示す。図の奥方向が x 軸であり、横方向が y 軸である。このような入出力特性を示すニューロンを否定包含ニューロンと呼ぶことにする。

順序回路を実装する上での注意点 ニューラルネットの論理回路で注意すべき点が1つある。それは、各論理ニューロンの入出力特性から判るように、入力を変化させていった時の出力の変化が滑らかな点である。順序回路を実現する上で、電気的な論理素子ではその変化が十分に急峻であるために問題はないが、ニューロンではこれが滑らかなために、値がLOWからHIGH(HIGHからLOW)に変化しきらない内に、その出力がフィードバックされてきてしまう可能性がある。これは誤動作の原因となるため、リカレントネットで順序回路を実現する場合は、ループに十分な遅延が必要となる。

以上示したように、AND, OR, 否定包含の各ニューロンを実現できた。こ

の3つと1つのバースト発火ニューロン, さらにループに十分な遅延を入れることで, 何如なる順序回路も実現できる. 次節からは, これらのニューロンを用いて, 迷路を解く順序回路を作る方法について述べる.

3.3 順序回路の構成

前節で構成した論理素子を用いて, 迷路を解く同期式順序回路を構成する. 同期は, フリップ・フロップを用いてとることにする. ただし, 脳にもフリップ・フロップが存在し, 同期をとっているとは考え難いことから, このフリップ・フロップをもつニューラルネットからフリップ・フロップを消す変換を考えるようにする. この変換法については, 後程考えることにする.

2.5.2で述べた方法に基づき, まず, 実装する, 短期記憶を使って迷路を解くアルゴリズムについて考える. 以下では, アルゴリズムを考える前に, まず, 環境である迷路の構成, 及びネットワークへの入出力について説明する. 続いて, その入出力の下で, 迷路を解くアルゴリズムについて考える. 最後に, そのアルゴリズムを実装した, フリップ・フロップを用いた同期式順序回路の構成について説明する.

3.3.1 迷路問題

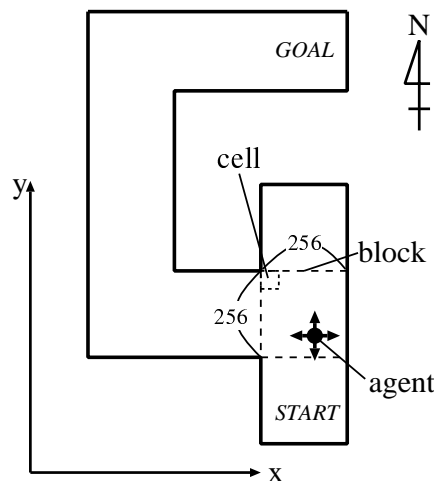


図24: エージェントが動く環境

エージェントが動き回る環境として, 図24のようなもの考える. すなわち, 迷路を2次元とし, しかも, エージェントにとって前後左右にしか壁が存在し

ないような環境である。このように問題を単純化しても、短期記憶という観点からは何ら問題がない。以下では、図24の上方、すなわち、 y 軸の正の方向を北(N)と呼ぶことにする。同様に、東(E)、南(S)、西(W)を定義する。

迷路空間を構成する最小単位を cell と呼び、エージェントはいずれかの cell 上に存在するものとする。cell が $256 * 256$ 個からなる空間を block と呼ぶ。迷路は複数の block から構成され、壁は block の境界線上に存在するものとする。なお、壁は厚みを持たないものとする(図24)。

エージェントは出力に応じて、現在の cell から N,E,S,W の4方向のいずれか隣接する cell へ動くものとする。従って、エージェント、すなわちニューラルネットの外部出力線は4本必要となる。4本の出力線それぞれが、N,E,S,Wのそれぞれの方向に対応し、各方向を示すニューロンがパルスを出した時点で、エージェントはその方向の隣接する cell へ移動する。

エージェントはあるスタート地点に置かれ、現在の入力と内部状態を用いて出力を計算しながらゴールを目指す。現在の入力とは、迷路内の現在地から割り出される情報であり、以下の2つを用意する。従って、ニューラルネットの外部入力線は計8本となる。

壁の入力 各方向(N,E,S,W)に壁があるかないか

匂いの入力 ゴールはどの方向(N,E,S,W)にあるか

壁の入力については異論がないだろうが、匂いの入力については、やや説明が必要であろう。我々は、エージェントが短期記憶を使うような解法で迷路を解くようになって欲しいのであり、そのために匂いの入力が必要となる。なぜなら、我々が巨大迷路の中に入って迷路を解く場合でも、ゴールがどこにあるかという情報がないと、左手の法則のような盲目的な方法で解かざるを得ない。そして、そのような解法を修得したニューラルネットは、我々の望むところではないからである。

壁の入力は図25のように計算する。N,E,S,Wの各方向に対して壁までの距離を計算し、それをパルスの頻度に変換して、ニューラルネットへの入力とする。壁に近い程、頻度が高くなるものとする。その際、エージェントが見える範囲を規定し、範囲を超えるような遠くに壁が存在する方向については、ニューラルネットへの入力パルス頻度を0とする。

匂いの入力は図26のように計算する。ゴールとエージェントを結ぶ線を算出

し、それとエージェントを中心とした円との交点を求める。交点からN-S軸、E-W軸にそれぞれ垂線を降ろした交点を、各方向を示すニューロンへの入力頻度とする。すなわち、ゴールとエージェントを結ぶ線のなす角度を、パルス頻度というアナログ値(0°で最高頻度、180°で最低頻度)で入力する。

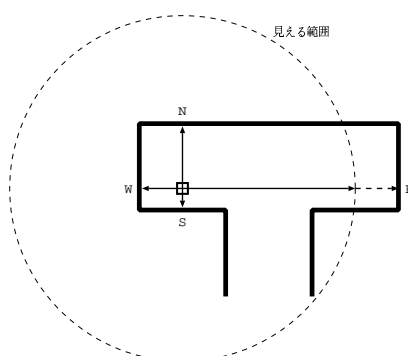


図 25: 壁の入力

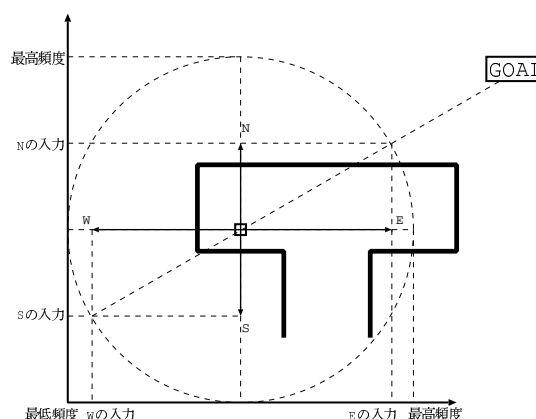


図 26: 匂いの入力

次項からは、このように定義された入出力環境の基で、記憶を用いて迷路を解くアルゴリズムについて説明する。

3.3.2 アルゴリズム

2.4で述べたように、実装するアルゴリズムは、短期記憶を使って迷路を解くアルゴリズムでなければならない。すなわち、左手の法則のような、入力情報(壁の情報)のみでゴールに辿りつけるようなアルゴリズムは適していない。

記憶を使うアルゴリズムとしてまず思い付くのは、通ってきた経路を全て覚えておくようなものである。しかし、我々が迷路を解く際に、そのように経路を全て覚えておくようなことはできない。我々が覚えておくことができるのは、せいぜい以前通った分岐の情報ぐらいである。

そこで、覚えておくのは分岐の情報のみ限定する。分岐を通過する度に、どちらから来たかを覚えておくことにする。次に分岐に戻って来た時には、記憶されている情報を見て、まだ進んでいない方向を選択すればよい。そのアルゴリズムは以下ようになる。

1. 現在地がゴールかどうかを判断。ゴールでなければ直進する。
2. 直進した結果、壁の状況が変化したら、「分岐点」「曲がり角」「行き止まり」

を判断し、以下を行う

分岐点 来た方向を記憶し、以下を行う

- (a) 初めて来た分岐点ならば、来た方向を「最初に分岐に入って来た方向」として記憶し、壁のない方向の内、ゴールに近づく方向を選択する
- (b) 一度来たことがある分岐点ならば、
 - i. まだ行ったことのない方向があれば、それらの方向の内、ゴールに近づく方向を選択する
 - ii. 全ての方向に進んだならば、最初にその分岐に入って来た方向に戻る

曲がり角 道なりに進む

行き止まり 来た方向に戻る

分岐において選択する方向の優先順位は、1.) まだ進んでない方向、2.) ゴールの匂いが強い方向、の順に選択する。1.) が存在しない時は、初めに分岐に入って来た方向に戻るものとする。従って、分岐毎に記憶されている情報は、進んだ方向、及び最初にその分岐に入って来た方向の2種類である。これは、進んだ方向のみを記憶していると、全ての方向に進んで分岐点に戻って来た時に、次に進むべき方向が分からなくなるためである。

3.3.3 迷路を解く順序回路の構成

前項で述べたアルゴリズムを順序回路で実現する。本節の冒頭で述べたように、同期はとりあえずフリップ・フロップでとる。ただし、次に述べるこのアルゴリズムの状態は、分岐点、曲がり角、行き止まりにおいて1回更新されるだけであるため、write-enable とセレクトを用いて回路を構成する。

一方、外部入力是非同期に変化する。このため、入力の変化タイミングとクロックの立ち上がりが重なる可能性があり、その結果 meta-stable が起こりうる。そこで、入力を同期化することで、入力は必ずクロックの立ち上がりから少し遅れて変化するようにする。

同期化は図27のようにして行う。すなわち、入力(*input*)と回路の間にフリップ・フロップを複数個(図では2個)挿入する。これは、フリップ・フロップが1個だと、そこで meta-stable が起こるためである。このようにすることで、*input'* はクロックの立ち上がりで変化する信号となる。

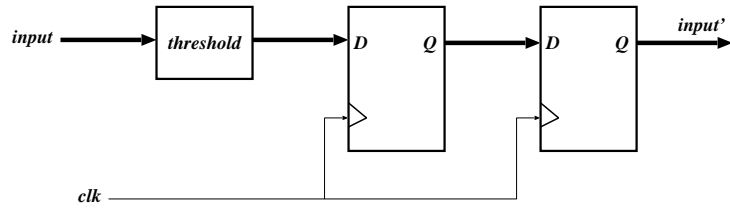


図 27: 同期化

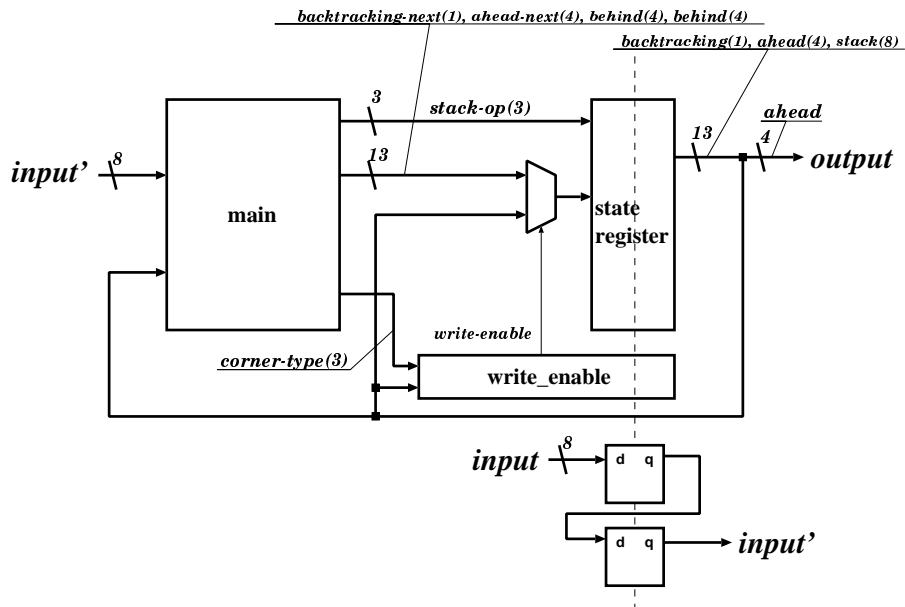


図 28: 回路構成

write-enable とセレクタを用いて、回路を構成する。その構成を図28に示す。図の *input* が3.3.1で述べた壁と匂いの入力に対応し、*output* がどの方向に進むかという4本の出力に対応している。状態レジスタの更新を *write-enable* によって制御する。*write-enable* は、分岐点、曲がり角、行き止まりにおいて、クロックの立ち上がり1回分、HIGH となるような信号である。この時、状態レジスタの値が新しい値に更新される。それ以外の時は現在の値を上書きするだけである。

状態レジスタ 回路の内部状態は *backtracking*, *ahead*, *stack* の3つである。*backtracking* は現在行き道か帰り道かを、*ahead* は前を向いている方角がどちらであるかを表している。*stack* はたどった道、戻るべき道を覚える。*stack* の段数は7段とする。これは人間が一度に記憶できる無意味な事象の数、すなわち、マジカルナンバーが7であるためである [7]。

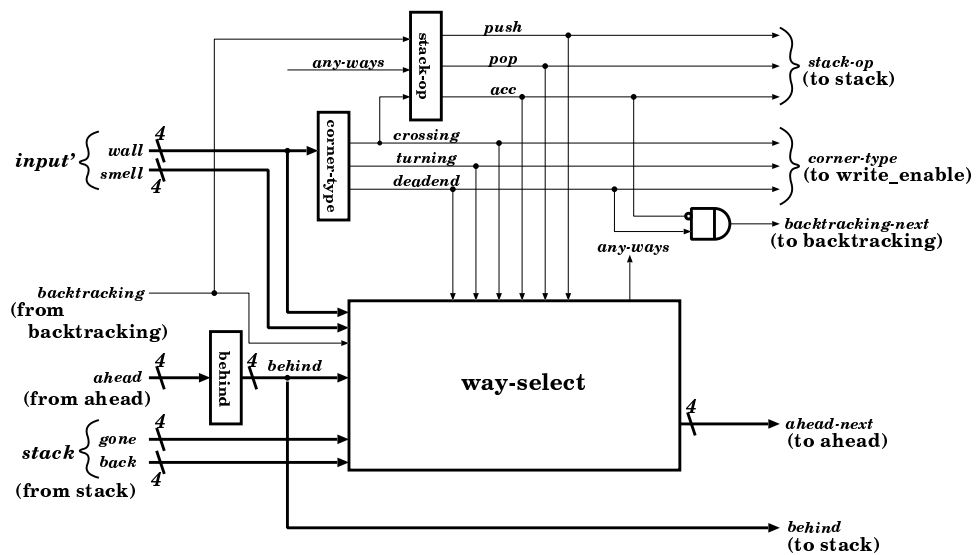


図 29: main の構成

main main の回路構成を図29に示す。main は、同期化された外部入力として8本の入力線からなる *input* (同期化された壁の入力 (*wall*)4本と匂いの入力 (*smell*)4本) と、状態レジスタから *backtracking* 1本、*ahead* 4本、*stack* 8本 (進んだ方向を示す *gone* 4本と初めに分岐に来た方向を示す *back* 4本) を受け取る。そして、6つのオペレーション (*stack-op* 3本と *corner-type* 3本) と9つの次状態 (*backtracking* 1本と *ahead-next* 4本と *behind* 4本) を出力する。

オペレーションは、*wall*から決まる *corner-type* と、*wall* と内部状態によって決まる *stack-op* がある。

corner-type は *crossing*, *turning*, *deadend*, の3つである。*crossing* は分岐点を、*turning* は曲がり角を、*deadend* は行き止まりを表す。

stack-op は *push*, *acc*, *pop* の3つである。これらは *crossing* の時に、内部状態 *backtracking* と、まだ進んでいない方向があるかを表す *any-ways* によって決まる。

push は、この分岐点に始めて来た時に行われる。従って、 $push = crossing \cdot \overline{backtracking}$ である。

acc は、この分岐点に戻って来たが、まだ他に行っていない道がある時に行われる。従って、 $acc = crossing \cdot backtracking \cdot any-ways$ である。

pop は、この分岐点に戻って来たが、もう他に行っていない道はない時に行われる。従って、 $pop = crossing \cdot backtracking \cdot \overline{any-ways}$ である。

これら3つの信号は、明らかに排他的である。*crossing* と *turning* と *deadend* が排他的であることから、*push*, *pop*, *acc*, *turning*, *deadend* の5つの排他的な信号によって、オペレーションは完全に指定される。

帰りがけを表す次状態 *backtracking-next* は、行き止まりで立ち上がり、分岐点に戻って来た時に立ち下がる。よって、 $backtracking-next = deadend \cdot \overline{acc}$ である。

次に進む方向を表す次状態 *ahead-next* は、*way-select* が決定する。*way-select* については後で述べる。

来た方向を表す *behind* は、現在進んでいる方向 *ahead* によって決まる。すなわち、*ahead* の逆方向が *behind* である。

stack スタックのエントリは、3.3.2で述べたように2種類(*back* と *gone*)用意する。*back* は最初に分岐に入って来た方向を、*gone* は既に進んだ方向を表している。

スタックのエントリは *push*, *acc* によって更新される。書き込まれるデータは *behind* によって表される。*gone* の更新は、行きがけではなく、帰りがけに行われる。

push 信号が入った時は、スタック・トップを1つ上げ、その *back*・*gone* に *behind* が書き込まれる。

acc 信号が入った時は、現在のスタック・トップの *gone* に *behind* を OR する。
 スタックのデータ・アレイの全体図(更新側のみ)とそのセルの構成を、それぞれ図 30・図 31 に示す。

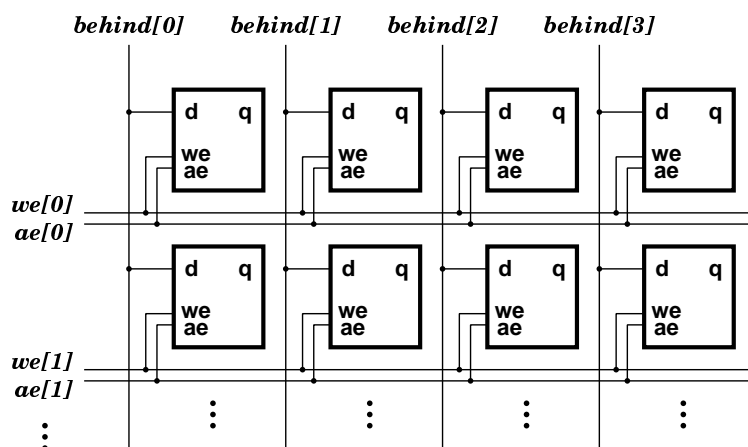


図 30: スタック・データ・アレイ

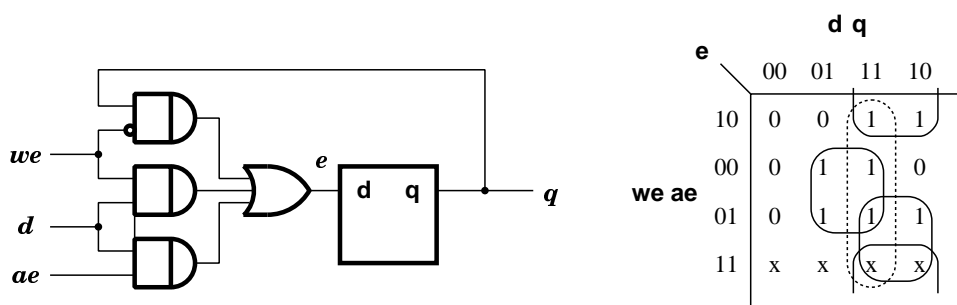


図 31: スタックの構成

way-select way-select の構成を図 32 に示す。way-select は、まだ行ってない道があるかを示す *any-ways* と次に進む方向を示す *ahead-next* を出力する。
push, *pop*, *acc*, *turning*, *deadend* の 5 つのオペレーションにより、*ahead-next* を出力するための処理は以下の 4 通りに分かれる。

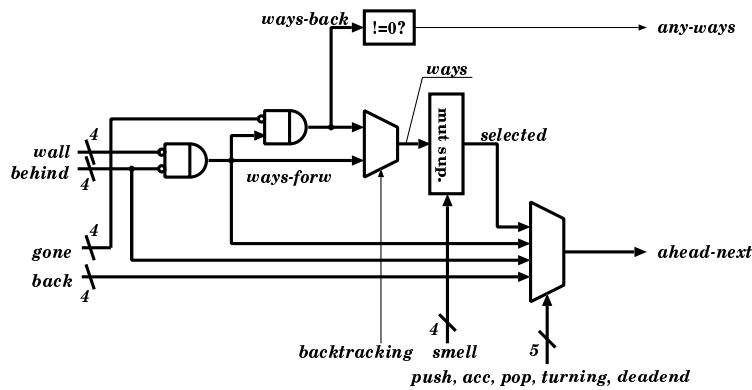


図 32: way-select の構成

- *push + acc* 現在分岐点において、まだ進んでない方向があるケース。進む方向の候補 (*ways*) の内、最も匂いの強い方向を選択 (*mutsup*) し、出力する。
- *turning* 来た方向以外の進める方向 (*ways-forw*) を出力する
- *deadend* 来た方向 (*behind*) を出力する
- *pop* 現在分岐点において、全ての方向に進んだケース。最初に分岐点に来た方向 (*back*) を出力する。

gone の更新は帰りがけに行われるため、戻ってきて次に行く道を選択する際 (*acc*) には、今戻って来た道 (*behind*) を除外する必要がある。よって、進む方向の候補 (*ways*) は、行き (*ways-forw*) と帰り (*ways-back*) で、それぞれ次のようになる。

$$\begin{aligned} \text{ways-forw} &= \overline{\text{wall}} \cdot \overline{\text{behind}} \\ \text{ways-back} &= \overline{\text{wall}} \cdot \overline{\text{behind}} \cdot \overline{\text{gone}} \end{aligned}$$

any-ways は、帰りがけの進む方向の候補である *ways-back* が全て LOW の時のみ LOW となる。

mutsup *mutsup* は、出力を相互抑制 (*mutual suppression*) することで、入力の最も強いものを選択して出力するモジュールである。その構成を図 33 に示す。図は、ニューロンを接続行列で表現している。横線が 1 つのニューロン及び出力線に相当し、縦線は入力線を表している。交点は、そのニューロンがその入力線と結合していることを表し、黒点は興奮性結合 (重みが正) を、白点は抑制

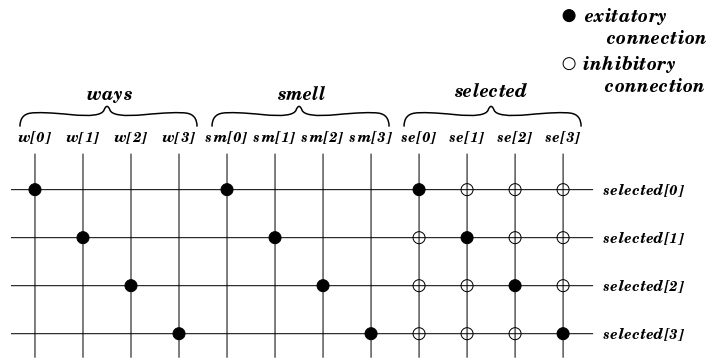


図 33: mutsup の構成

性結合 (重みが負) を表している。

mutsup は進む道の候補 (*ways*) の内、最も匂いの強い方向を選択する。従って、各方向に対応するニューロン (*selected*) は、各方向に対応する *ways* と *smell* を興奮性の入力とする。各ニューロンの出力 (*selected*) が相互に抑制結合することで、出力が1つに絞られる。

第4章 課題

前章において、迷路を解く順序回路をリカレントシンプルスパイキングニューラルネットワーク上に実現した。2.5.1で述べたように、Pulse-based Modelを用いたリカレントネットの1つであるリカレントシンプルスパイキングニューラルネットワークは、脳における同期の難しさと同じような問題を抱えている。しかも、そもそも脳において同期がとられているかどうかすらわからない。そこで以下では、リカレントシンプルスパイキングニューラルネットワークにおける同期について考察することにする。

4.1 パルスと同期

2.3.1で述べたように、Pulse-based Modelのリカレントネットには同期の問題が存在するが、本論では同期をとることを仮定してきた。そして、同期をとる方法としてフリップ・フロップを用いることにし、そうしてできたりカレントネットにフリップ・フロップを消す変換を施すことにした。

変換法の1つとして、現在のリカレントネットは順序回路であるのだから、全てのパスの高さを揃えることでフリップ・フロップを消す方法が考えられる。順序回路では、パスの高さを揃えれば、全てのパスの遅延が揃う。遅延が揃えば、全てのパスの信号のウェーブ・フロントが常に揃い、フリップ・フロップなしに同期をとることができる。

そこで、ウェーブ・フロントが常に揃っている状態を考える。2.1.2で述べたように、Pulse-based Modelではニューロンの時間発展過程もモデル化されているため、パルスの時間変化に時間がかかる。すなわち、Pulse-based Modelではウェーブ・フロントは幅をもっている。

この幅をもつウェーブ・フロントが常に揃っているリカレントネットについて考える。前述のように、ウェーブ・フロントが揃い続けるためには、少なくとも、パスの遅延が全て揃っていなければならない。

このパスの遅延が揃ったリカレントネットで、ウェーブ・フロントが揃い続けるための条件を考える。幅をもつウェーブ・フロントが、揃ってループを周回している状況を考える。ここに入力が入ってくるものとする、図34に示すように、入力のウェーブ・フロントとループのウェーブ・フロントが揃わない場合には、ループのウェーブ・フロントの幅が広がっていくことになり、最終

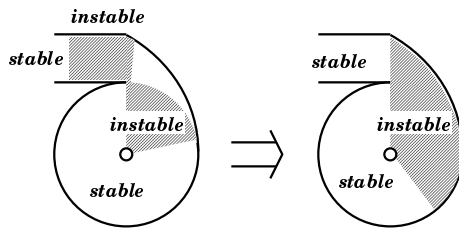


図 34: リカレントネットにおけるウェーブ・フロント

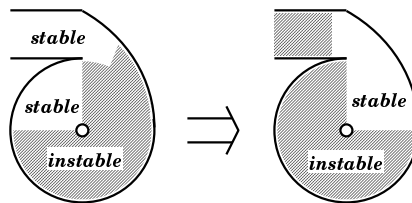


図 35: リカレントネットが安定に動作するための条件

的には、ループは全てウェーブ・フロントとなってしまふ。

このことから、ループのウェーブ・フロントの幅が広がらないための条件として、以下の2つが導かれる。

- ループのウェーブ・フロントの先端が入力との合流点に到達する前に、入力のウェーブ・フロントの終端が合流点を通過している (図 35 左)
- ループのウェーブ・フロントの終端が合流点を通過するまでは、入力のウェーブ・フロントの先端が合流点を通過してはならない。 (図 35 右)

どちらか一方の条件でも見たされない場合は、入力のウェーブ・フロントがループのウェーブ・フロントでない部分と重なるため、ループのウェーブ・フロントの幅は広がっていく。

以上の考察から、全てのパスの高さを揃えることでウェーブ・フロントを揃え、先に述べた2つの条件を満たすように入力の変化タイミングとループの長さの関係を調整してやれば、安定して動作するリカレントネットが得られそうである。

しかし、現在までに、全てのパスの高さを揃えても、パスの遅延は揃わないことがわかっている。これは、そもそも シンプルスパイキングモデルの入力と出力の遅延を揃えることができないためである。この入出力間で遅延が揃えることができないという性質は、シンプルスパイキングモデルの立ち上がり時間と立ち下がり時間が大きく異なるという性質に起因する。

シンプルスパイクモデルの出力を立ち上げるものは、興奮性入力であればその立ち上がりである。つまり、入力頻度が高くなることで出力頻度は高くなる。しかし、それには少々時間を要することは、3.1.2の図14を見れば明らかだろう。すなわち、出力頻度が高くなるためには、ニューロンの状態 u が定常状態となる必要があり、そのためにはある程度のパルスの蓄積を必要とするからである。

一方、出力の立ち下がりも、興奮性入力であればその立ち下がりによって引き起こされる訳だが、それは立ち上がりに比べれば極く短い時間で起こる。入力頻度がある頻度から0に変化する、すなわち、急に入力パルスが無くなることを考える。すると、出力を立ち下げる基であるニューロンの状態 u は、入力パルスが無くなった時点で下がり始める。従って、出力の立ち下がりも、入力の立ち下がりとはほぼ同時に起こる。

以上のように、シンプルスパイクモデルの立ち上がり時間と立ち下がり時間の違いから、リカレントシンプルスパイクニューラルネットワークでは遅延を揃えることができない。従って、ウェーブ・フロントを揃えることもできず、ウェーブ・フロントを揃えることで同期をとる方法は実現できない。

4.2 今後の課題

本論では、短期記憶について考えるために、記憶をもつネットワークを、Pulse-basedに基づく構成法によって実現することについて述べた。Pulse-basedに基づく構成法には、重みを調整する一般的な方法が存在しないことから、Pulse-based Modelで論理素子を構成し、それを用いて順序回路を作り、何らかの変換を施すという手法をとることにした。

Pulse-based Modelを用いたリカレントネットでは、同期をとるのが難しい。しかも、脳において同期をとっているかどうかすらわからない。しかし、順序回路を構成する際には同期式の方が容易であるため、同期式順序回路を構成することにした。同期は、不自然ではあるがフリップ・フロップによってとることとし、後から何らかの変換によってフリップ・フロップを消すことにした。

Pulse-based Modelとしては、その自然さから、シンプルスパイクモデルを用いることにした。そして、このシンプルスパイクモデルで、論理素子としてAND, OR, NOTの代わりとして否定包含ニューロンを構成した。これ

らの素子を用いて、分岐を記憶しながら迷路を解くアルゴリズムを、同期式順序回路として実装した。

前述したように、Pulse-based Model を用いたリカレントネットの1つであるリカレントシンプルスパイキングニューラルネットワークでは、同期をとるのが難しい。そこで、リカレントシンプルスパイキングニューラルネットワークにおける同期について考察を行った。

ウェーブ・フロントを揃えることで同期をとる方法について考えてきたが、考察の結果、ウェーブ・フロントを揃えることができないことがわかった。これは、シンプルスパイキングモデルの立ち上がり時間 / 立ち下がり時間が大きく異なることに起因する。

今後は、そもそも脳において同期が重要かどうかについて、もう少し考察してみる必要がある。また、脳において同期がとられていると仮定した場合、前節で述べたウェーブ・フロントを揃える方法以外の同期の方法について、考えてみる必要がある。

謝辞

本研究の機会を与えて下さり，適切な御指導を賜りました富田眞治教授に深甚な謝意を表します。

また，貴重な御助言をいただいた森眞一郎助教授，北村俊明助教授，中島康彦助教授，五島正裕助手，津邑公暁助手，津田晃寿氏に深く感謝致します。

さらに，日頃暖かく御鞭撻下さった京都大学工学部情報学科富田研究室の諸兄に感謝致します。

ありがとうございました。

参考文献

- [1] Adrian, E.: The impulses produced by sensory nerve endings, *Journal of Physiology (Lond.)*, No. 61, pp. 49–72 (1926).
- [2] Armstrong, C.: Voltage-dependent ion channels and their gating, *Physiological Reviews*, Vol. 72(suppl.), pp. S5–S13 (1992).
- [3] Chagnac-Amitai, Y., L. H. and Prince, D.: Burst generating and regular spiking layer V pyramidal neurons of rat neocortex have different morphological features, *J Comp Neurol*, No. 296, pp. 598–613 (1990).
- [4] Chagnac-Amitai, Y. and Connors, B.: Synchronized excitation and inhibition driven by intrinsically bursting neurons in neocortex, *Journal of Neurophysiology*, No. 62, pp. 1149–1162 (1989).
- [5] Delcomyn, F., 小倉 昭彦・富永恵子訳: ニューロンの生物学, Vol. 2, トッパン (1999).
- [6] Garzon, M. and Franklin, S.: Neural Computability II, No. 1 (1994).
- [7] George, A. M.: The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information, *The Psychological Review*, Vol. 63, pp. 81–97 (1956).
- [8] Hava, T. and Eduardo, D.: Analog Computation via Neural Networks, *Theoretical Computer Science*, No. 131, pp. 331–360 (1994).
- [9] Hodgkin, A. and Huxley, A.: A quantitative description of membrane current and its application to conduction and excitation in nerve, *Journal of Physiology (Lond.)*, Vol. 117, pp. 500–544 (1952).
- [10] Jun, T.: An Interpretation of the "Self" From the Dynamical Systems Perspective: A Constructivist Approach, Sony csl technical report.
- [11] Jun, T.: Model-Based Learning for Mobile Robot Navigation from the Dynamical Systems Perspective, *IEEE Trans. on System, Man and Cybernetics Part B*, Vol. 26, No. 3, pp. 421–436 (1996).
- [12] Maass, W. and Bishop, C. M.(eds.): *Pulsed Neural Networks*, MIT Press (1999).
- [13] McClelland, J.L., M. B. and O'Reilly, R.: Why there are complementary learning systems in the Hippocampus and Neocortex, Technical Report

- PDO.CNS.94.1, Carnegie Mellon University (1994).
- [14] McCormick, D.A., C. B. L. J. and Prince, D.: Comparative electrophysiology of pyramidal and sparsely spiny stellate neurons of the neocortex, *Journal of Neurophysiology*, No. 54, pp. 782–805 (1985).
 - [15] McNaughton, B.: Neural mechanisms for spatial computation and information storage, *Neural Connections and Mental Computations* (Nadel, L.(ed.)), MIT Press, pp. 285–349 (1989).
 - [16] Segundo, J.P., P. D. W. H. G. H. and Moore, G.: Input-output relations in computer-simulated nerve cells, *Kybernetik*, Vol. 4, No. 5, pp. 157–171 (1968).
 - [17] Squire, L.R., C. N. and Nadel, L.: The Medial temporal region and memory consolidation : A new hypothesis, *Memory consolidation* (Weingartner, H. and Parker, E.(eds.)), Erlbaum, pp. 185–210 (1984).
 - [18] Wilson, M.: Reactivation of hippocampal ensemble memories during sleep, *Science*, No. 265, pp. 676–679 (1994).