

博士論文

スパイキングニューラルネットワーク  
を用いた時系列シンボルに対する頻出  
パターンマイニングに関する研究

森田 賢太



# 目次

<b>第1章</b>	<b>序章</b>	<b>1</b>
1.1	本研究の背景	1
1.1.1	頻出パターンマイニング	3
1.1.2	ニューラルネットワーク	3
1.2	本研究の目的	4
1.3	本論文の構成	5
<b>第2章</b>	<b>2シンボル長の頻出順序パターンを抽出するニューラルネットワーク</b>	<b>7</b>
2.1	まえがき	7
2.2	頻出な順序パターンを抽出するネットワーク	8
2.2.1	概要	8
2.2.2	構造・動作	9
2.2.3	学習	11
2.2.4	ユニット	14
2.3	抽出の確認	15
2.3.1	実験条件・方法	15
2.3.2	頻出な順序パターンの抽出の確認	16
2.3.3	順序パターンとしての抽出しているかの確認	18
2.4	むすび	19
<b>第3章</b>	<b>音声データから3シンボル長以上の頻出な順序パターンを抽出するニューラルネットワーク</b>	<b>21</b>
3.1	まえがき	21
3.2	学習	22
3.3	ネットワークの構造	22
3.4	抽出の確認	30
3.4.1	実験方法・条件	30

3.4.2	実験結果 . . . . .	32
3.5	むすび . . . . .	33
<b>第4章</b>	<b>任意の出現頻度の順序パターンを抽出するニューラルネットワーク</b>	<b>35</b>
4.1	まえがき . . . . .	35
4.2	任意の出現頻度以上の順序パターンを抽出するスパイクングニューラルネットワーク . . . . .	36
4.2.1	ネットワークの構造 . . . . .	36
4.2.2	抽出後のふるまい . . . . .	38
4.2.3	学習 . . . . .	42
4.3	抽出の確認 . . . . .	45
4.3.1	実験目的 . . . . .	45
4.3.2	頻出な順序パターンのみを抽出できるかの確認 . . . . .	45
4.3.3	パラメーターによるしきい値の設定 . . . . .	50
4.3.4	考察 . . . . .	52
4.4	むすび . . . . .	52
<b>第5章</b>	<b>総論</b>	<b>54</b>
5.1	本研究で得られた成果 . . . . .	54
5.2	今後の課題 . . . . .	56
	<b>謝辞</b>	<b>58</b>
	<b>参考文献</b>	<b>59</b>

# 第1章 序章

## 1.1 本研究の背景

近年、人工知能に関する研究が盛んに行われている。人工知能とはコンピュータ上で人間の知能を再現しようとする技術である。人工知能技術には、機械に音声の内容を理解させる音声認識技術や、提示された画像に映る物体名を答える画像認識、データ集合から頻出パターンを抽出する頻出パターンマイニングなどがある。これらの技術は、音声対応アプリケーションや自動運転システムなどの商品及びサービスに組み込まれ、我々の日常で活用され始めている。

上記で述べた人工知能技術は、主に機械学習と呼ばれる手段が用いられる。機械学習は、大量のデータを機械に与えることで、特定のルールやパターンを発見したり、未知のデータに対して分類や識別または予測をすることができる手法である。機械学習では、パターンの発見や識別を行うアルゴリズムなどを識別器もしくは分類器と呼ぶ。識別器にデータを与えることを学習と呼び、学習には主に教師あり学習、教師なし学習、強化学習の3つがある。

教師あり学習は、人がデータを分類し、入力データと正解となる出力をペアで教える学習方法である。例えば、画像認識を行う識別器を作成する際には、学習データの画像に対して、物体の名前をラベル付けして学習させる。学習が進んだ識別器は、データを与えられると与えられた入力データに対して正解となる出力を出すことができる。また、学習した識別器は、未知のデータを与えると、そのデータが与えられたラベルの中のどれに属するか出力を行うことができる。教師あり学習は、認識や予測などに向いており、SVM(サポートベクターマシン)[1, 2]やニューラルネットワーク、線形回帰、ランダムフォレスト [3, 4] などがある。

教師なし学習は、教師あり学習とは異なり、データにラベル付けをしない学習方法である。教師なし学習では、アルゴリズムによりデータを自動分類を行う。教師なし学習には、 $K$ 平均法や自己組織化マップ(SOM)[5]などがあり、類似データのグルーピングや次元削減に向いている。

強化学習は、試行錯誤をして、報酬が多く得られるような行動を学習方法である [6]。強化学習では教師あり学習とは異なり、入力に対する正解の行動出力を教えない。強化学習は、ある状態において行動した結果、その報酬を得る。一つの行動だけでなく複数の行動を行った際に、もらえる報酬が最大になる行動の過程を学習する。強化学習には、TD 学習や Q 学習などがある。

教師あり学習では、ディープラーニング (深層学習)[7, 8, 9] と呼ばれる機械学習の手法があり、近年、この技術により画像認識技術は急速な発展をした。この技術は、画像認識だけでなく動画認識や音声認識などにも大きな影響をあたえている [10, 11, 12, 13]。図 1.1 はディープラーニングの構成図である。ディープラーニングは、人間の神経細胞のを工学的に模したシステムであるニューラルネットワークの一種である。ディープラーニングの登場前は、ニューラルネットワークの層は3層までであり、4層以上のニューラルネットワークは学習が上手くできていなかった。4層以上のニューラルネットワークでは、ニューラルネットワークにおける教師あり学習の誤差伝搬 (BP)[14] において、勾配消失問題などがあり、十分に学習ができなかったためである。Hinton らは、少ない層のニューラルネットワークを教師なし学習にて初期化し、それらの層を組み合わせて教師あり学習することでこれらの問題を解決した [15]。ディープラーニングは大量の画像、テキスト、音声データなどを与えると、各データに含まれる特徴を各層で自動で学習する。画像認識分野では、畳み込みニューラルネットワーク (Convolutional Neural Network)[16] により、画像の特徴を抽出することができ、人間と同等以上の認識をすることができるようになった。

ディープラーニングにより、画像認識分野では、物体を高精度で認識できるようになった。そこで近年、画像の集合体である動画などの時系列データに対してディープラーニングを活用する研究が行われている [19, 20]。Simonyan らは、動画を空間情報と時間情報に分け、それぞれを二つの畳み込みニューラルネットワークに与えて学習させる手法を提案している [21]。空間情報では画像のフレームを一枚ずつ与え物体などの学習を行い、時間情報ではフレームとフレームの差分を入力として与えて時系列の学習をしている。しかし、これらの手法は画像認識のように認識率は高くない。この原因は、時間情報を学習するネットワークが状態の遷移などの時間的特徴を学習しているのではなく、ただ識別しやすいフレームを選択しているだけだと考えられている。そのため、ディープラーニングの画像認識のように、教師なし学習により特徴を自動で学習するメカニズムが必要である。時系列データにおいて、ラベル付けなしで時系列的特徴を学習できる方法を模索

する。

時系列データにおいて、ラベル付けなしで時系列的特徴を模索するために、人間の語彙の偶発的獲得に注目する。古樋らは、人間の偶発的な語彙獲得について調査した [22]。その結果、子供が映画 1 本の中で頻度の高い語彙を獲得したことが分かった。音素をシンボルとすると、語彙はシンボルの順序パターンと捉えることができる。このように、ストリーミング形式で与えられる時系列シンボルから、頻繁に出現する順序パターンを抽出することができれば、ラベル付けなしで時系列的特徴を学習できると考えられる。

### 1.1.1 頻出パターンマイニング

時系列シンボルから頻出な順序パターンを抽出する方法として、頻出パターンマイニングに注目する。頻出パターンマイニングは、制約を満たすパターンにおいて、データ集合から高い頻度のパターンをすべて列挙する技術である。データ集合としては、アイテム集合や系列データ、時系列などがある。

頻出パターンマイニングに、相関ルール抽出がある。相関ルールは、ある条件  $X$  が発生する場合には、同時に条件  $Y$  が発生する場合が頻繁に起こる。桜井らは特徴的なイベントの並びを注目頻出パターンを抽出するために Apriori アルゴリズムに基づく手法を提案した [23]。Apriori アルゴリズムは、相関ルールを用いた計算によって頻出な順序パターンを抽出する手法である。桜井らはこの Apriori アルゴリズムを拡張し、上位  $k$  個の非類似な順序パターンを抽出できるようにした。この手法は、Apriori アルゴリズムは抽出する際にデータの個数などを数える必要があるため、リアルタイム処理に向いていない。

### 1.1.2 ニューラルネットワーク

ストリーミング形式で与えられる手法にニューラルネットワークを用いた手法が考えられる順序パターンを学習する有名なニューラルネットワークのモデルとして RNN(Recurrent Neural Network)[24, 25, 26] による手法がある。これらはエルマンネット [27] と呼ばれる単純再帰型ニューラルネットワークを基にした手法である。入力層の一部に中間層の状態を戻すことにより、過去の状態を入力とすることができる。この構造により過去の入力履歴を保持する事ができるため、順序列を学習する事ができる。この RNN をさらに拡張した LSTM(Long short-term

memory)[28, 29, 30]などが提案されている。これらのニューラルネットワークは、教師あり学習により時系列を学習する。

時間的特徴に特化したニューラルネットワークとして、スパイキングニューラルネットワーク [31] がある。教師なし学習による手法として、田中らの連想記憶を用いた手法がある。この手法は、青木らが提案したモデル [32] を改良したスパイキングニューラルネットワークによる手法である。頻繁に入力される2シンボル長の順序パターンをスパイキングニューラルネットワークの連想記憶により記憶させることで、頻出な順序パターンを学習する。しかし、この手法はニューラルネットワークの構造の問題により3シンボル長以上の頻出な順序パターンを抽出することができない。

## 1.2 本研究の目的

本論文では、人間の学習を模倣し、時系列シンボルから頻出な順序パターンを抽出するスパイキングニューラルネットワークを提案する。人間の学習を模倣するため、頻出な順序パターンを抽出する際に、4つ条件がある。1つ目はリアルタイム処理である。これは、識別器が学習と認識を同時にすることを目的としているからである。2つ目はラベル付けをしないことである。人間の語彙の獲得は、教師あり学習のようにこの順序パターンが頻出であると教えられるのではなく、多くの順序パターンから自動で頻出な順序パターンを抽出し学習する。そして、3つ目は抽出される順序パターンはシンボルの長さに依存しないことである。語彙の長さは、2シンボル長、3シンボル長、4シンボル長とさまざまであるためである。4つ目の条件は、抽出される順序パターンに個数制限をもうけないことである。これは、各系列データには複数の順序パターンがあるが、その個数は同じではないからである。

また、時系列シンボルとして音声や動画を選んだ場合、さらに条件が加わる。それは、シンボルの出現タイミングを考慮することである。これらのデータでは、シンボルが一定間隔で出現しているように見えるが、実際にはばらつきがある。そのため、シンボルの出現間隔にゆらぎをもつ時系列シンボルから頻出な順序パターンを抽出できることが望ましい。

さらにこれらの条件に加えさらに、抽出する順序パターンの出現頻度のしきい値を調整できるようにする。各系列データによってその順序パターンの出現頻度は



さまざまであり、同じ出現頻度でもそれを頻出とするかは異なる。そのため、どんな系列データにおいても、特定の出現頻度以上の場合、頻出な順序パターンとして抽出されるようにする。具体的には、頻出とする出現頻度のしきい値を決め、すべてのシンボル長において出現頻度がしきい値以上の順序パターンのみ抽出する。

### 1.3 本論文の構成

本論文は5章で構成される。

第1章では、本研究の背景として人工知能技術における学習などについて述べた。頻出な順序パターンを抽出する手法についていくつか手法を紹介した。また、教師なし学習としてニューラルネットワークについてもいくつかモデルを紹介した。そして、本研究の目的であるすべてのシンボル長において頻出な順序パターンのみを抽出することを述べた。

第2章では、本論文の基礎となるスパイクングニューラルネットワークについて説明する。このスパイクングニューラルネットワークは、4つ条件のうち、リアルタイム処理かつラベル付けなしで2シンボル長の順序パターンを抽出することができる。

第3章では、第2章のスパイクングニューラルネットワークを拡張し、残りの条件であるシンボルの長さに依存しないこと、抽出される順序パターンに個数制限をもうけないことをできるようにする。また、基となるスパイクングニューラルネットワークは、シンボルの出現タイミングを考慮することができる。実験では、音声データを用いて抽出の確認を行う。

第4章では、さらに本論文のスパイクングニューラルネットワークを拡張する。頻出とする出現頻度のしきい値を決め、すべてのシンボル長において出現頻度がしきい値以上の順序パターンのみ抽出できるネットワークについて説明する。

第5章では、本研究で得られた成果についてまとめ、最後に今後の課題について触れた。

なお、参考文献は最後にまとめて掲載した。

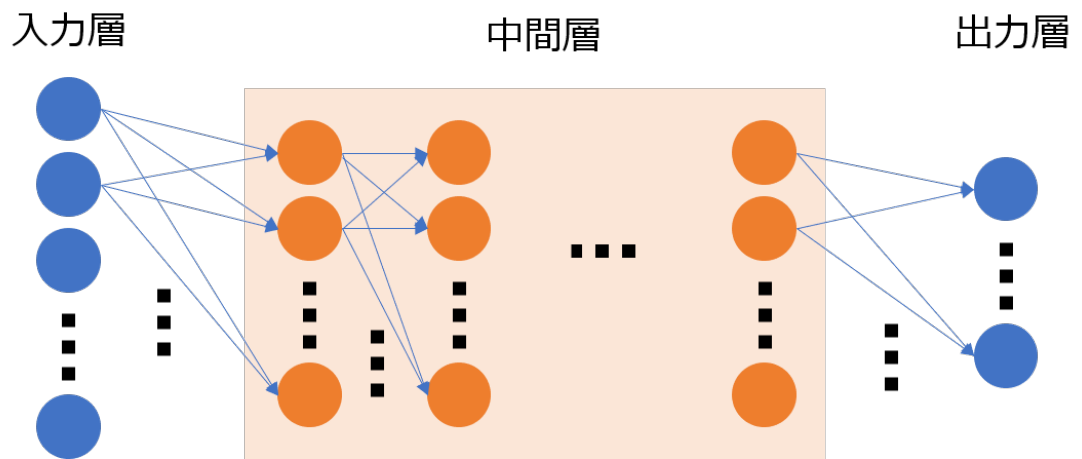


図 1.1: ディープラーニングの構成図

# 第2章 2シンボル長の頻出順序パターンを抽出するニューラルネットワーク

## 2.1 まえがき

1章で述べたように，本論文の目的は時系列シンボルから頻出な順序パターンを抽出することが目的である．この順序パターンの抽出は，人間の学習を模倣しており，様々な条件を満たすようにする．

めざす頻出な順序パターンの抽出について説明する．時系列シンボルとは，あらかじめ定められたシンボルを時間ごとに1つずつ順に並べた列である．この時系列シンボルを提示することで，頻出な順序パターンを抽出することを目的としている．ここでは，順序パターンとはシンボルとその次のシンボルの時間が決まった間隔で出現するシンボルのパターン列である．たとえば図 2.1 は，時系列シンボルであり，下の数字は現在からの経過した秒数で，上の英字はそのときの出現シンボルである．図 2.1 において，1 秒間隔でシンボル列を見ると，現在から 30 秒前の時間内では，シンボル “X” の次に 1 秒後に “Y” が出現することが 4 回ある．このようにある間隔で見たときに，シンボル “X” のあとに “Y” が出現する事を順序パターン “XY” とする．シンボル “X” と “Y” の出現の間隔が 2 秒の物もあるが，これは \_ (無入力) との順序パターンである “X \_” や “\_ Y” として捉えるため，別の順序パターンとして捉える．図 2.1 の時系列シンボルの中では，順序パターン “YX” や “ZA” よりも “XY” の順序パターンが多く出現している．このような時系列シンボルを提示したときに，“XY” という順序パターンを抽出することが目的である．

このような頻出な順序パターンの抽出にあたって，いくつか注意しなくてはならない点がある．1つ目は抽出したいシンボル列の出現間隔が決まっていないことである．図 2.1 の例では，順序パターン “XY” が出現してから次の “XY” が出現す

るまでに、いくつかの関係ないシンボル“A”や“B”などが出現している。このような状況において、抽出対象の順序パターンの出現間隔が一定ではないため、抽出対象の順序パターンの頻度だけでなく、頻出な順序パターンを抽出するためには、どのシンボル列を学習すべきか捉えにくいことを考える必要がある。2つ目は各シンボルの出現が決まった瞬間に来ないことである。たとえば、11秒前のところを見ると、シンボル“X”は11秒ちょうどに入力されていない。このように各シンボルは一定間隔から少しずれて出現したりと決まった瞬間に入力されるとは限らない。

本章ではこれら2つの注意点を考慮にいて、ストリーミング形式で与えられる時系列シンボルから頻出な順序パターンの抽出をめざす。今回、第一歩として4つ条件のうち、1つ目のリアルタイム処理であることと、2つ目のラベル付けをしないことで抽出することを解決する。そのため、3つ目の条件に対しては、2シンボル長に固定し、次のような空白で単語として区切られた時系列シンボルから頻出な順序パターンの抽出をめざす。

…;ZA, ,X,Y, ,A,X, ,Z,Y, ,X,Y, ,B,C, ,X,Z, ,X,Y, ,C,A, ,X,Y, …

頻出な順序パターンを抽出する際に、人間の学習を模倣しているため、リアルタイム処理、ラベル付けをしない、抽出される順序パターンはシンボルの長さに依存しない、抽出される順序パターンに個数制限をもうけないことの4つの条件がある。本章では、この研究の第一歩として、リアルタイム処理、ラベル付けをしないことの条件を満たすことをめざす。

## 2.2 頻出な順序パターンを抽出するネットワーク

### 2.2.1 概要

頻出な順序パターンの抽出を実現するために、時系列シンボルを入力として受け取り、頻出な順序パターンが入力されたときにのみ反応するニューラルネットワークを教師なし学習を用いて構築する。ニューラルネットワークを用いることで、リアルタイム処理に対応することができる。また、教師なし学習によりラベル付けをしないという条件にも対応できる。

入力線はシンボルの種類数分用意し、シンボルの入力はシンボルに対応する入力線にのみ与える。出力は抽出したいシンボル列数分だけ用意し、それぞれ異なる

るシンボル列に対して反応するものとする。時系列シンボルは、出現時刻とおり  
に1シンボルずつ与えられる。

図2.2は、順序パターン“XY”の抽出に成功したときの、入力と出力の関係を表  
したものである。図2.2(a)は順序パターンの“XY”のシンボル“X”を入力した時  
のようすであり、図2.2(b)は“XY”の“Y”を入力したようすである。そして、図  
2.2(c)はシンボル“Y”を与えたあとのようすである。シンボル“X”を入力した際  
には出力が行われず、“X”を入力した後にシンボル“Y”を入力すると“XY”専用  
の出力線から出力が行われる。このように頻出な順序パターンのみに対応し、特  
定の出力線から出力される状態が頻出な順序パターンを抽出した状態となる。こ  
のような状態を作るためのニューラルネットワークの構成、学習方法、ユニット  
について説明する。

### 2.2.2 構造・動作

提案するニューラルネットワークは、2つのブロックから構成される。図2.3は  
ネットワークの構成図である。ネットワークにはシンボルの入力を受け付ける入  
力ブロックと頻出な順序パターンを抽出する出力ブロックがある。入力ブロック  
は、入力端子と2つのユニットが鎖状に接続されたものがシンボルの種類数分あ  
る。各鎖は、シンボル1種に対して1対1で対応している。また、各鎖の数は、シ  
ンボルの種類数分ある。入力ブロックでは、1層目に最後に与えられたシンボル  
の情報、2層目にその1ステップ前のシンボルの情報が保持される。これにより、2  
シンボル長の順序パターンにおいて、1層目は順序パターンの後ろのシンボルに  
対応し、2層目は順序パターンの前のシンボルに対応する。出力ブロックは出力  
層が1つあり、各ユニットは入力ブロックのすべてのユニットと結合荷重で接続  
されている。出力層のユニットの個数は、ユニットの頻出な順序パターンを抽出  
したい数だけ用意する。これらのユニットは発火すると、次のユニットへと信号  
を伝搬する。ネットワークのすべての結合には遅延があり、信号の伝搬には一定  
時間がかかる。その遅延時間はシンボルの入力間隔と同等である。

具体的な動作例として、頻出な順序パターンを抽出したネットワークに頻出な  
順序パターンを与えた時の動作を説明する。このネットワークの出力層のユニット  
は、初期は順序パターンに対してランダムに発火する。シンボルがネットワーク  
に与えられるたびに結合荷重が調整され、頻出な順序パターンが与えられたとき  
のみ特定の1つのユニットが発火するようになる。図2.4は、順序パターン“XY”



図 2.1: 時系列シンボルの例

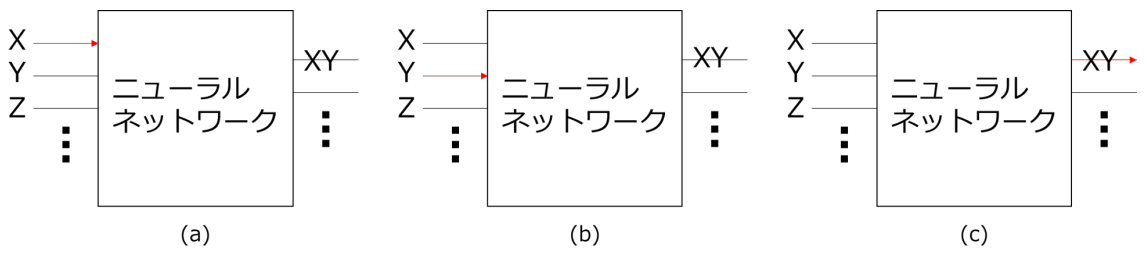


図 2.2: 抽出成功後の入出力イメージ

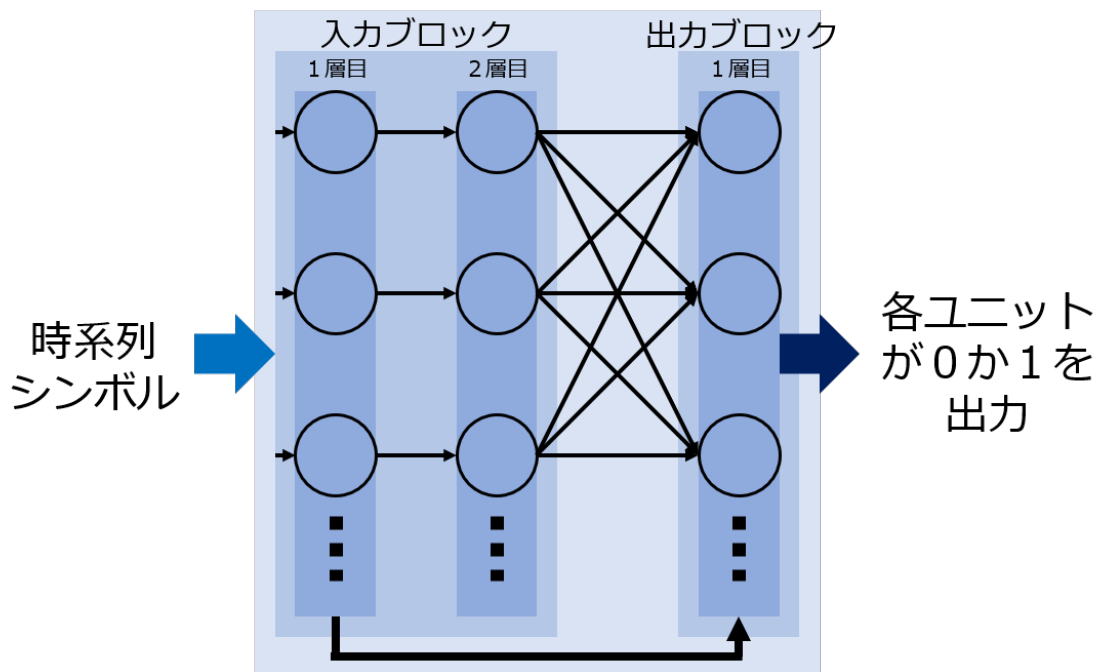


図 2.3: ネットワークの構成図

を抽出したネットワークネットワークに“XYZ”が入力されたときのネットワークの応答を示している。各図の入力端子は上からシンボル“X”，“Y”，“Z”に対応しており，入力ブロックの1層目は上から“○X”，“○Y”，“○Z”に対応し，2層目は“X○”，“Y○”，“Z○”に対応している。そして，出力ブロックの出力層の一番上のユニットは“XY”に対応している。初期のユニットは前の層と全結合しているが，順序パターンに反応するようになったユニットは，2つの層から1本ずつのみ接続されている状態になる。以下に4ステップのネットワークの状態を示す。

1. 図2.4(a)は“XYZ”を与える直前の状態を示している。
2. 図2.4(b)はiシンボル“A”を与えた後の状態を示している。入力ブロックの第1層のシンボル“○X”に対応するユニットのみが発火する。
3. 図2.4(c)はシンボル“Y”を与えた後の状態を示している。入力ブロックではシンボル“○Y”に対応する第1層のユニットと信号の伝搬により第2層の“X○”に対応するユニットが発火する。
4. 図2.4(d)はシンボル“Z”を与えた後の状態を表す。入力ブロックでは第1層のシンボル“○Z”に対応するユニットと第2層のシンボル“Y○”に対応するユニットが発火し，出力ブロックでは，“X○”と“○Y”に対応するユニットからの信号の伝搬により順序パターン“XY”に対応するユニットが発火する。

このように適切に学習したネットワークは頻出な順序パターンに反応する。また，ネットワークの結合荷重を調べることにより，どのユニットが何の順序パターンに反応するようになったか知ることができる。次節ではこの状態になる手法を述べる。

### 2.2.3 学習

2.2.2のような頻出な順序パターンを抽出した状態を，ニューラルネットワークの教師なし学習により結合荷重を調整することで実現する。学習による結合荷重の調整は，頻出な順序パターンに対して強化を，そうでない順序パターンに対して減衰を行う。この方法により層間の結合荷重を調整することで，出力層の各ユニットが頻出な順序パターンの入力に呼応して反応する自己組織化を行う。

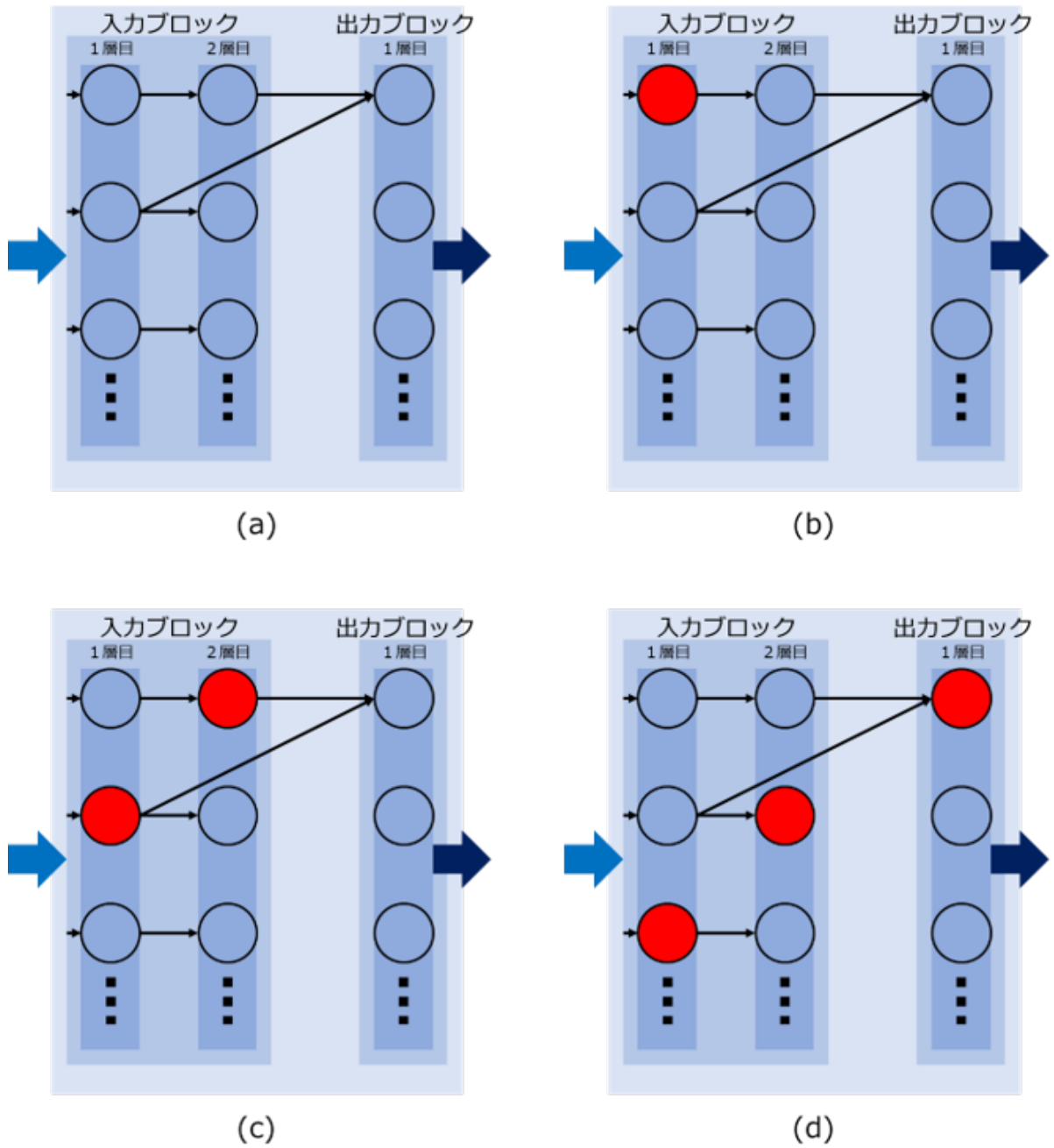


図 2.4: 抽出時のネットワークのふるまい



このネットワークを、ニューラルネットワークの代表的な強化学習の手法である Hebb 則 [34] に基づいて学習する。本提案のモデルではこの Hebb 則をシンボルブロックと出力ブロック間の結合に対して行う。Hebb 則では二つのユニット間の結合をそれらの発火時刻に基づいて調整する。前ユニット (シンボルブロックのユニット) が発火した後に後ユニット (出力ブロックのユニット) が発火した場合、式 (2.1) に従い、その結合荷重を一定値増やすことで強化する。

$$W_{ij} \leftarrow W_{ij} + A_+ \quad (2.1)$$

$W_{ij}$  はユニット  $j$  からユニット  $i$  に接続している結合荷重で、 $A_+$  は定数である。

具体的な強化例を説明する。順序パターン “XY” の入力により 図 2.4(c) のように “X○” に対応するユニットと “○Y” に対応するユニットが同時に発火する。この2つのユニットから出力ブロックのユニットへ信号が与えられ、図 2.4(c) のようにユニットが発火した場合、そのユニットにつながる “X○” に対応するユニットと “○Y” に対応するユニットの結合荷重が共に上がる。

本提案の学習方法は Hebb 則に 2 点工夫を加えることにより、頻出な順序パターンを抽出する自己組織化を実現することができる。1 点目の工夫は結合荷重の強化の制限である。結合荷重に上限をつけ、結合荷重をそれ以上あげない。また、下限  $W_{\min}$  を 0 とする。強化により結合荷重の上限を達したとき、その出力ユニットへ接続する他の同じシンボルブロックの層のユニットとの結合荷重を 0 にする。具体的には式 (2.2) に従う。

$$\begin{cases} W_{ij} \leftarrow W_{max} & (\text{if } W_{ij} > W_{max}) \\ W_{else} \leftarrow 0 & (\text{if } W_{ij} > W_{max}) \end{cases} \quad (2.2)$$

$W_{max}$  は結合荷重の最大値である。また、 $W_{else}$  はユニット  $i$  に接続するユニット  $j$  と同じ層にあるユニット  $j$  以外のすべての結合荷重である。この工夫により、値が調整された結合荷重はその後学習を行わない。ユニット  $i$  に対して、シンボルブロックの2つの層からそれぞれ  $W_{max}$  の結合荷重が1つできたとき、ユニット  $i$  は頻出な順序パターンを抽出した状態となる。2 点目は工夫を結合荷重の下げ方である。シンボルブロックのユニットが発火していない場合、以下の式 (2.3) に従って指数関数的減衰を行う。

$$W_{ij} \leftarrow W_{ij} + A_- \cdot \exp(t/\tau) \quad (2.3)$$

$A_-$  は定数であり、 $\tau$  は学習則での時定数である。 $t$  はユニット  $j$  が前回発火してからの経過時間であり、発火していない時間が長いほど結合荷重が下がる。Hebb 則

では出力ユニットが発火しないと結合荷重が下がらないため、まったく入力されていない入力ユニットの結合荷重は下がらない。よって結合荷重を指数関数的に減少することにより入力されていない入力ユニットの結合荷重を下げる事が出来る。

## 2.2.4 ユニット

このネットワークのユニットは、シンボルブロックと出力ブロックでユニットの動作が異なる。シンボルブロックのユニットは、信号を与えられるとすぐに発火する。

出力層の各ユニットは、シンボルブロックにあるすべてのユニットからの信号を結合荷重で重み付けして受け取る。また、出力層内の各ユニットは互いに接続しており、側抑制を行う。本手法では、ユニットの発火判定・側抑制の発動は、常に出力層ユニットの番号順に行うこととする。従って、ユニット番号  $n$  のユニットが発火することで、ユニット番号  $n + 1$  以降のユニットに側抑制の刺激を与える。ユニットの発火動作は、Leaky Integrate-and-Fire (LIF) モデルに基づいており、式 (2.4) に従う。

$$\begin{cases} V(t) \leftarrow V(t) + (V_r - V(t))\exp(-t/\tau_m) + V_{input} \\ V(t) \leftarrow V_r \end{cases} \quad (\text{if } V(t) > \theta) \quad (2.4)$$

LIF は内部電位  $V(t)$  を時間ごとに加算していき、閾値  $\theta$  を超えたときに発火する。 $\theta$  は  $W_{max}$  の 2 倍にすることで、2 つの信号が同時に来た時のみ発火するようになる。LIF は単純に積分発火するだけでなく、内部電位の漏れもおこなう。 $V_{input}$  は入力層からの信号に結合荷重を掛けたものの総和および側抑制からの入力である。 $V_r$  はユニットの内部電位の初期値であり、ユニットが発火した場合、内部電位  $V(t)$  は  $V_r$  にもどる。 $\tau_m$  は時定数でありこの数値が高いほどユニット内部の電圧が下がりにくい。

LIF モデルの特徴である積分発火と内部電位の漏れの量により、シンボルの入力はずれても、出力ユニットは発火することができる。積分発火により 0.9 秒後でも 1.1 秒後に入力されても、内部電位が下がりきらなければ、発火する事ができる。内部電位の漏れの量によりシンボルの入力のずれを対応することができるが、内部電位の漏れは少なすぎるとシンボル列の抽出に失敗する。たとえば、内部電位の漏れが少ないとき、“X○”と“○Y”の順序パターンが頻出だと、出力ユニットの発火により、シンボルブロック内の“X○”と“○Y”に対応するユニットの結合

荷重が上がる。このように出力ユニットの発火頻度が高いと，“XY”の頻度に関係なく“XY”に反応するネットワークになってしまう。これは内部電位の漏れが少ないと、閾値を超える事が多くなるため，“X○”と“○Y”の入力時に出力ユニットが発火する事が多いからである。よってシンボル列として抽出するには出力ユニットの発火頻度を押さえる必要がある。発火頻度を押さえる方法としては即抑制を用いる事や結合荷重の上限を低く設定する方法などがある。しかし、即抑制は発火を押さえたいユニットではないユニットが発火する必要があり、どのように構成するかが難しい。また、結合荷重の上限を下げると、反応したいシンボル列の入力ユニットが反応しても内部電位が閾値を超えないため使う事ができない。そのため、ユニット自身に頻度を押さえる仕組みを与える。具体的には出力ユニットの時定数 $\tau_m$ を小さくすることである。時定数 $\tau_m$ を小さくすると、蓄積していた内部電位の減少量が増加する。従って、内部電位の漏れが多くなる。そのため、積分発火モデルであるLIFモデルの出力ユニットは発火しにくくなる。よって，“X○”の“X”に反応することが減り、頻出な順序パターンを学習できる確率が上がる。しかし、時定数を下げすぎると内部電位の漏れが多いため、シンボルの時間のずれを吸収できなくなる可能性がある。よって時定数のパラメータを適切にする必要がある。

## 2.3 抽出の確認

この章では、提案手法により頻出な順序パターンを抽出できるのかを、プログラムを用いた実験により確認する。時系列シンボルの抽出において、順序パターンの出現間隔とシンボルの入力時間のずれの2点を注意する必要がある。入力時間のずれについては積分発火モデルのLIFにより対応できる可能性が高い。そのためもう一つの注意点である、シンボル列の出現間隔に関係なく頻出のシンボル列を抽出できるかを優先で確認する。

### 2.3.1 実験条件・方法

実験では、6種のシンボル(A, B, C, D, E, F)を使用し、これらのシンボルを2つ組み合わせた順序パターンを36種(AA, AB, AC, ..., FE, FF)用意した。時系列シンボルを作成する際に、各順序パターンの生成確立を変化させることで、各順序パターンの頻度を調整した。これらの順序パターンを空白(無入力)で区切

表 2.1: ネットワークのパラメーター

項目	値
$W_{\text{init}}$	15.0
$V_r$	-68.0
$\theta$	-10.0
$\tau_m$	5.0

表 2.2: 学習のパラメーター

項目	値
$W_{\text{max}}$	30.0
$W_{\text{min}}$	0.0
$A_+$	1.0
$A_-$	0.01
$\tau_d$	10.0

り，ランダムな順序 (使用回数は問わない) で連結することで，ネットワークに提示する時系列シンボルを作成した．これを 1 秒に 1 シンボルずつネットワークに与える．

ネットワークの構成は，入力ブロックの各層のユニットを 6 個，出力層のユニットを 3 個とした．ユニット間の伝搬時間は 1 秒とした．結合荷重の初期値と出力層のユニットの各パラメータは表 2.1 とした．また，学習のパラメーターは表 2.2 とした．これらのパラメータは試行錯誤で決定した．

### 2.3.2 頻出な順序パターンの抽出の確認

この節では，提案手法により頻出な順序パターンを正しく，指定数分だけ抽出できるかを確認する．

実験は，空白を含めた時系列シンボルをネットワークに提示しながら，各出力ユニットが抽出した順序パターンを調査した．表 2.3 は，その結果である．

表 2.3: 各ケースにおける頻出な順序パターンの抽出結果

順序パターン		出力ユニット		
		1 番目	2 番目	3 番目
ケース 1	「AB」 10.0% 「CD」 10.0% 「EF」 10.0% 他各 2.1%	EF(359)	AB(728)	CD(1190)
ケース 2	「AB」 10.0% 「CB」 10.0% 「EF」 10.0% 他各 2.1%	AB(326)	CB(509)	BF(1070)
ケース 3	「AB」 42.0% 「CD」 42.0% 「EF」 16.0% 他各 0.0%	AB(161)	CD(281)	なし
ケース 4	「AB」 10.0% 他各 2.5%	AB(458)	DA(947)	なし

この表は、入力として与えた時系列シンボルを生成した際の各順序パターンの生起確率と、それを学習したネットワークの3つの出力ユニットが抽出した順序パターンを示したものである。順序パターンの右の数字は、ネットワークにシンボルを与え始めた時刻を0として、その順序パターンを抽出したときの秒数である。またユニットの番号は側抑制の優先順位である。

以下、各ケースについて詳細に分析する。1番目、2番目のケースでは、高い頻度の3種の順序パターンをすべて抽出することができた。3番目のケースでは、ケース1よりもユニット番号1番のユニットと2番のユニットの抽出が速かったが、3番目に頻出である順序パターン“EF”を抽出することができなかった。提案手法では、側抑制の関係から、ユニット番号1番のユニットと2番目のユニットにどちらかが発火したとき、3番目のユニットは発火することができない。このケースでは、頻出な順序パターンである“AB”と“CD”の出現する割合が大きすぎるため、3番目のユニットへの結合荷重の減衰に、増加が間に合わず抽出に失敗した。4番目のケースでは、特に頻度の高いABは正しく抽出することができた。もう1つ抽出した順序パターン“DA”は、特に頻度は高くなかったが、乱数の関係で、たまたま抽出されたと考えられる。

以上より、提案手法によりストリーミング形式で与えられる時系列シンボルから頻度が高い順序パターンを抽出できることが確認した。

### 2.3.3 順序パターンとしての抽出しているかの確認

次に、学習法で述べたように、出力ユニットが頻繁に発火する場合、順序パターン“A○”が頻出であった場合、シンボル“A”に対応する入力線に接続している1つ目のユニットの結合荷重が増加していく。同様に、順序パターン“○B”が頻出であった場合、シンボル“B”に対応する入力線に接続している2つ目のユニットの結合荷重が増加していく。これにより、順序パターン“A○”と“○B”が頻出であった場合、出力ユニットが誤って順序パターン“AB”を学習してしまう可能性がある。そこで先ほどと同じパラメータを用いて、出力ユニットの発火を抑えることで、順序パターン“AB”を誤って抽出しないかを確認する。

表2.4は、順序パターン“AB”の出現を0回の時系列シンボルをネットワークに与えた結果である。

表 2.4: シンボル列として抽出しているかの実験結果

順序パターン		出力ユニット		
		1 番目	2 番目	3 番目
ケース 1	「A ○」 15.0% 「○ B」 15.0% 「AB」 0.0% 他各 2.8%	AA(353)	BE(773)	なし

順序パターン“A ○”と“○ B”の出現頻度は高いが、順序パターン“AB”を誤って抽出することはなかった。この結果より、頻出シンボルによる頻出な順序パターンの抽出への影響はないと考えられる。

## 2.4 むすび

本章では、時系列シンボルから頻出な順序パターンを抽出することをめざした。頻出な順序パターンを抽出する際に、4つの条件があった。1つ目はリアルタイム処理、2つ目はラベル付けをしないこと、3つ目はシンボルの長さに依存しないこと、そして、4つ目は、抽出される順序パターンに個数制限をもうけないことである。今回、第一歩として4つ条件のうち、1つ目のリアルタイム処理であることと、2つ目のラベル付けをしないことで抽出することを解決することを目的とした。

ストリーミング形式で与えられる時系列シンボルから頻出な順序パターンの抽出をするために、スパイクニューラルネットワークを用いた手法を提案した。ニューラルネットワークを用いることで、1つ目の条件であるリアルタイム処理に対応することができる。また、2つ目のラベル付けをしないに対しては、ニューラルネットワークの教師なし学習を用いることで解決をめざした。

提案したスパイクニューラルネットワークは、シンボルブロックと出力ブロックの2つのブロックをもった特殊な構造となっている。シンボルブロックはシンボルの入力を受け付け、シンボルの与えられた瞬間だけでなく、その1ステップ前のシンボルの情報を保持することができる。このシンボルブロックに頻出な順序パターンが与えられたときのみ、出力ブロックの出力層のユニットが発火するように、随時学習を行っている。学習は教師なし学習である Hebb 側を用いた。頻出な順序パターンを抽出する自己組織化を実現するために、この Hebb 側に2つの

工夫を取り入れた。1つ目は、結合荷重に上限値と下限値を設定することである。2つ目は、発火していないユニットの結合荷重を発火していない時間に依存して下げることである。

提案したニューラルネットワークの有効性を確認するために、2つの実験を行った。1つ目は、頻出な順序パターンを抽出できるか確認した。さまざまなデータを6種のシンボルを使って時系列シンボルとして作成した。すべてのデータに対して、出現率が10%以上の頻出な順序パターンを抽出できることを確認した。2つ目は、頻出な順序パターンとして抽出を行っているかを確認した。本提案手法では、シンボルの出現率が高いとそのシンボルでできた順序パターンを誤って抽出する可能性がある。そのため、“A○”と“○B”の出現頻度が高いデータを与え、“AB”を抽出しないことを確認した。以上の実験結果から、提案手法は、頻出な順序パターンを抽出できたと考える。

本章では、3つ目と4つ目の条件を解決していない。そのため、次章ではこのネットワークを改良し、シンボルの長さに依存しないこと、抽出される順序パターンに個数制限をもうけないこと、この2つの条件を解決する。また、本章ではシンボルの出現タイミングについての議論は不十分である。次章では、これについても議論していく。



# 第3章 音声データから3シンボル長以上の頻出な順序パターンを抽出するニューラルネットワーク

## 3.1 まえがき

2章では、時系列シンボルから頻出な順序パターンを抽出するスパイクングニューラルネットワークを提案した。このネットワークは、リアルタイム処理でラベル付けされていないデータから頻出な順序パターンを抽出することができる。しかし、抽出できる順序パターンは2シンボル長までであり、出力層に用意したユニットの個数と同じ数の順序パターンしか抽出することができない。

本章では、2章のネットワークを拡張し、本論文の抽出する際の条件である、順序パターンのシンボル長に依存しないこと、抽出される順序パターンに個数制限をもうけないことの2つの条件を満たすネットワークを提案する。基となるネットワークは、スパイクングニューラルネットワークを用いることでシンボルの出現タイミングに対応し、シンボルが一定間隔から少しずれて出現しても頻出な順序パターンを抽出することが可能としている。シンボルの出現タイミングについても議論する。

本章では、シンボルの出現間隔にゆらぎをもつ時系列シンボルとして音声データを対象とする。図3.1は時系列シンボル“ABCDE”を示している。横軸は時間を表しており、各シンボルの出現タイミングは、横軸に黒い点で表されている。各シンボルの内容は、黒い点の上に表されている。各シンボルは、ほぼ等間隔で出現しているように見えるが、実際には等間隔から少しずれて出現する。音声データから頻出な順序パターンを抽出するには、このような少しのずれならば、ずれを無視し、一定間隔の時系列シンボルと見なし、頻出な順序パターンを抽出する

必要がある。

本章では、本論文での抽出の際の4つ条件を満たし、シンボルとシンボルの時間間隔が一定でない時系列シンボルから頻出な順序パターンを抽出することをめざす。ネットワークの学習則と構造を変えることで、順序パターンのシンボル長に依存しないこと、抽出される順序パターンに個数制限をもうけないことの2つの条件を満たすようにする。また、音声データから頻出な順序パターンを抽出できるかを確認する。

## 3.2 学習

基となるネットワークの学習則はHebb側に基づいていた。音声データを時系列シンボルの対象とした場合、2での実験データよりも頻出な順序パターンの出現頻度はかなり低くなる。Hebb側は、出力ユニットが発火した時刻の1ステップ前の入力ブロックのユニットにしか結合荷重を調整しない。そのため、頻出な順序パターンの出現頻度が5%以下の場合、結合荷重の増加が時間的減衰の減少に負けてしまい、抽出が困難となる。

時系列シンボルの順序パターンをより、安定して抽出できるようにするために、ユニットが発火した瞬間だけでなく、数ステップ前後の発火に依存して結合荷重を調整するSTDP学習則[38]に基づいて行う。具体的には、式(4.3)に従う。

$$\begin{cases} W_{ij} \leftarrow W_{ij} + A_+ \exp\left(\frac{-x}{\tau_{\text{stdp}}}\right) & (\text{if } x > 0) \\ W_{ij} \leftarrow W_{ij} - A_- \exp\left(\frac{x}{\tau_{\text{stdp}}}\right) & (\text{if } x < 0) \\ W_{ij} \leftarrow W_{ij} & (\text{if } x = 0) \end{cases} \quad (3.1)$$

$A_+$  と  $A_-$  は定数、 $x$  はユニット  $i$  とユニット  $j$  の発火時刻の差で、 $\tau_{\text{stdp}}$  は時定数である。ユニット  $j$  が発火した後にユニット  $i$  が発火した場合は結合荷重の値が増加し、逆の順番で発火した場合は値は減少する。増減量は  $x$  が小さいほど大きくなる

## 3.3 ネットワークの構造

2のネットワークは、出力ブロックの出力層が1つのため、2シンボル長までの順序パターンしか抽出することができなかった。3シンボル長以上の頻出な順序

パターンを抽出するために、出力層を増やす必要がある。出力層の増やし方について議論する。

単純に出力層を増やしても3シンボル長以上の頻出な順序パターンを抽出することは難しい。図3.2は、出力ブロックに出力層を増やした場合のネットワークの構造図である。出力層ブロックの $n$ 層目は $(n+1)$ シンボル長の順序パターンに対応する。第 $n$ ( $n \geq 2$ )層目のユニットは、入力ブロックの1層目のユニットと出力ブロックの $(n-1)$ 層のユニットと結合する。このように接続することで、たとえば順序パターン“ABC”は、出力ブロックの1層目の“AB”を抽出したユニットと、入力ブロックの“OC”に反応するユニットとの結合荷重が最大値に達することで、ネットワークは“ABC”を抽出することができる。

しかし、この構造には、3シンボル長以上の順序パターンを抽出できない場合がある。1つ目の理由は、3シンボル長以上の順序パターンを抽出する層のユニットは、異なる発火頻度の層と接続しているからである。3シンボル長以上の順序パターンを抽出する層のユニットは、出力ブロック内の一つ前の層と入力ブロックの第1層目のユニットと全結合している。入力ブロックのユニットは、信号が与えられると発火するため、シンボルがネットワークに入力されるたびに発火する。一方、出力ブロックのユニットは頻出な順序パターンが入力された場合のみ発火する。このように、入力ブロックと出力ブロックのユニットでは発火頻度が異なる。STDP学習則による結合荷重の増減量は発火タイミングに依存するため、発火頻度の多い入力ブロックからの結合荷重は発火頻度が少ない出力ブロックの結合荷重に比べて増加しやすい。発火頻度が少ない出力ブロックに学習則のパラメータを合わせてしまうと、2シンボル長を抽出するユニットへ接続する結合荷重が過剰に値を増加させ、頻出ではない順序パターンを抽出してしまう。

このような問題を防ぐために、3シンボル長以上の順序パターンを抽出する層において、接続元の2層の発火頻度を等しくするために、2シンボル長の順序パターンを抽出する構造に着目する。2シンボル長の順序パターンを抽出する出力ブロックのユニットは、入力ブロックの1層目と2層目のユニットと全結合している。入力ブロックのこの2層はユニットが鎖状に接続されており、1層目のユニットが発火した場合には2層目のそれに接続しているユニット1つだけが発火する。このように出力ブロックの1層目のユニットに接続している入力ブロックの2つの層のユニットの発火頻度は変わらない。

本稿では、3シンボル長以上の順序パターンを抽出する出力ブロックの層に対して、2シンボル長を抽出する構造を取り入れる。図3.3は、本章で提案するネッ

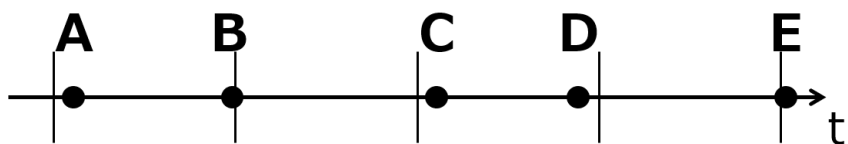


図 3.1: 音声データにおけるシンボルの出現例

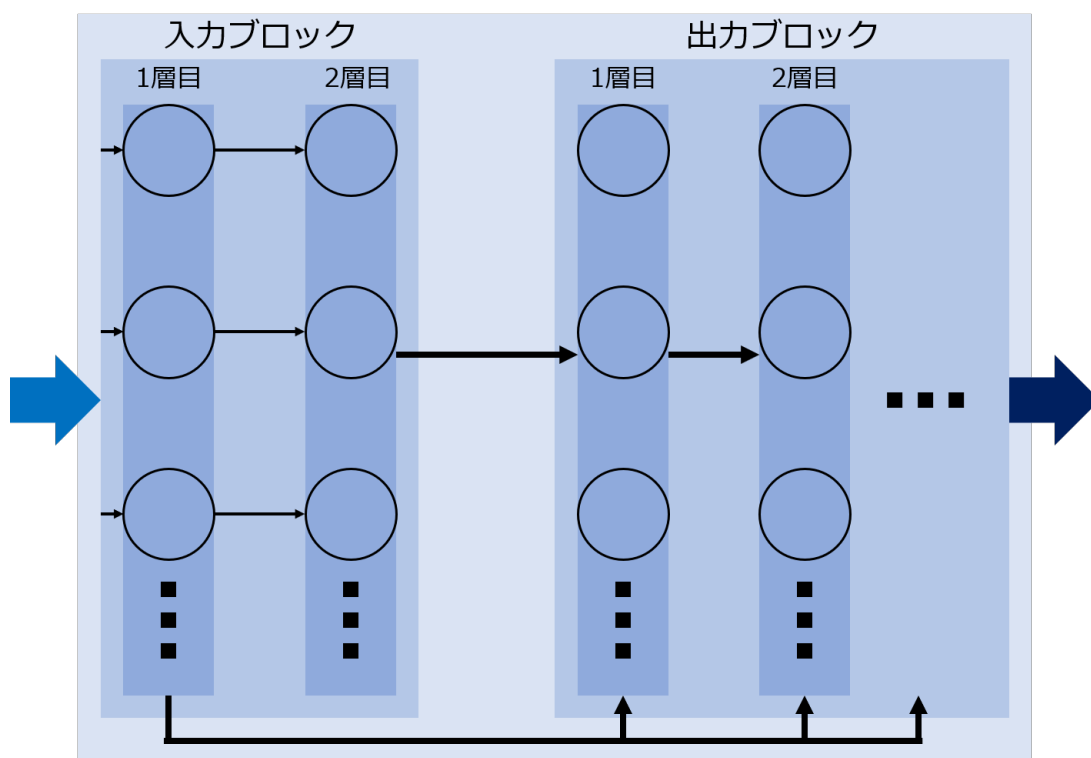


図 3.2: 出力ブロックに出力層を増やした場合

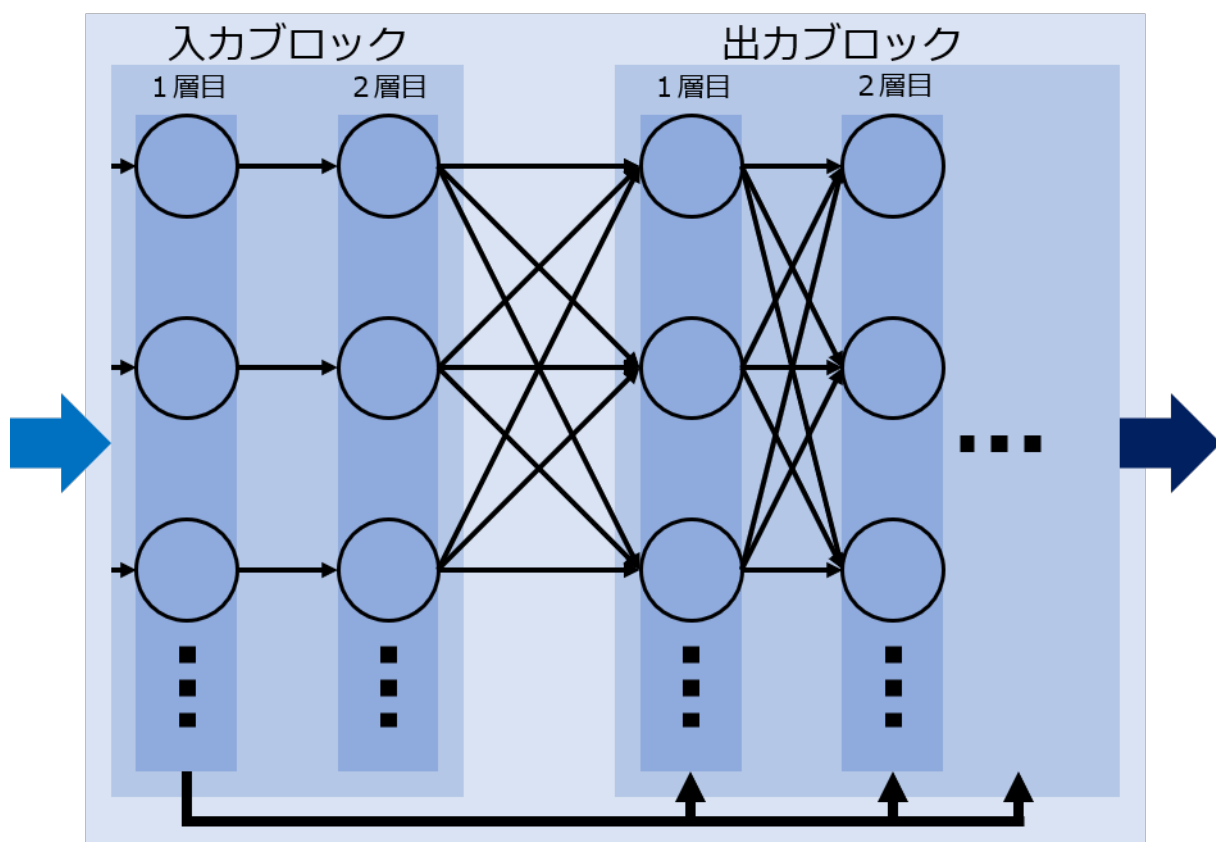


図 3.3: 提案するネットワークの構成

トワークの構造を表している。ブロック  $n$  ( $n \geq 2$ ) のユニットは  $n$  シンボル長の順序パターンに対応している。各ブロックには、既存手法の入力ブロックと同じように2つの層があり、ユニットが鎖状に接続されている。ブロック 1 は既存手法の入力ブロックと同じである。一方、ブロック 2 以降では、1 層目は LIF モデルのユニットであり、2 層目は入力ブロックのユニットと同じである。

ブロック  $n$  ( $n \geq 2$ ) では、1 層目のユニットはブロック ( $n - 1$ ) の2つの層と結合する。ネットワークが、頻出な順序パターンを抽出した後に、その抽出した順序パターンが与えられたときのネットワークのふるまいを説明する。図 4.2, 図 4.3, 図 4.4, 図 4.5, 図 4.6, 図 4.7, は、順序パターン“ABC”を抽出したネットワークネットワークに“ABCCA”が入力されたときのネットワークの応答を示している。図内の入力端子は上からシンボル“A”, “B”, “C”に対応している。そして、ブロック 1 は上から“AB”, “BC”に対応し、ブロック 2 は“ABC”に対応している。初期のユニットは前の層と全結合しているが、順序パターンに反応するようになったユニットは、2つの層から1本ずつのみ接続されている状態になる。以下に6ステップのネットワークの状態を示す。

1. 図 4.2 は“ABCCA”を与える直前の状態を示している。
2. 図 4.3 はシンボル“A”を与えた後の状態を示している。ブロック 1 の第 1 層のシンボル“○A”に対応するユニットのみが発火する。
3. 図 4.4 はシンボル“B”を与えた後の状態を示している。ブロック 1 ではシンボル“○B”に対応する第 1 層のユニットと信号の伝搬により第 2 層の“○A”に対応するユニットが発火する。
4. 図 4.5 は 1 個目のシンボル“C”を与えた後の状態を表す。ブロック 1 では第 1 層のシンボル“○C”に対応するユニットと第 2 層のシンボル“B○”に対応するユニットが発火し、ブロック 2 の第 1 層では、“A○”と“○B”に対応するユニットからの信号の伝搬により順序パターン“AB”に対応するユニットが発火する。
5. 図 4.6 は 2 個目のシンボル“C”を与えた後の状態を表している。ブロック 2 では第 1 層で順序パターン“BC”に対応するユニットが発火し、第 2 層では第 1 層の“AB○”に対応するユニットが発火する。

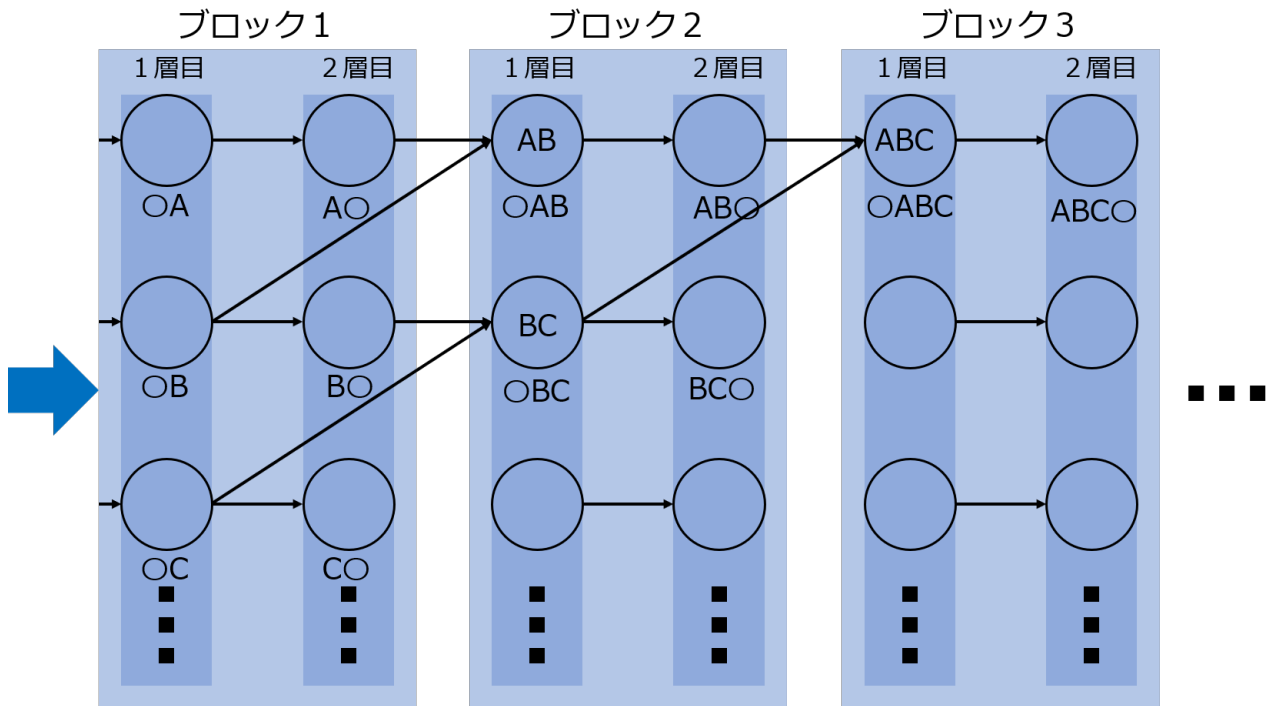


図 3.4: 抽出時のネットワークのふるまい : ステップ 1

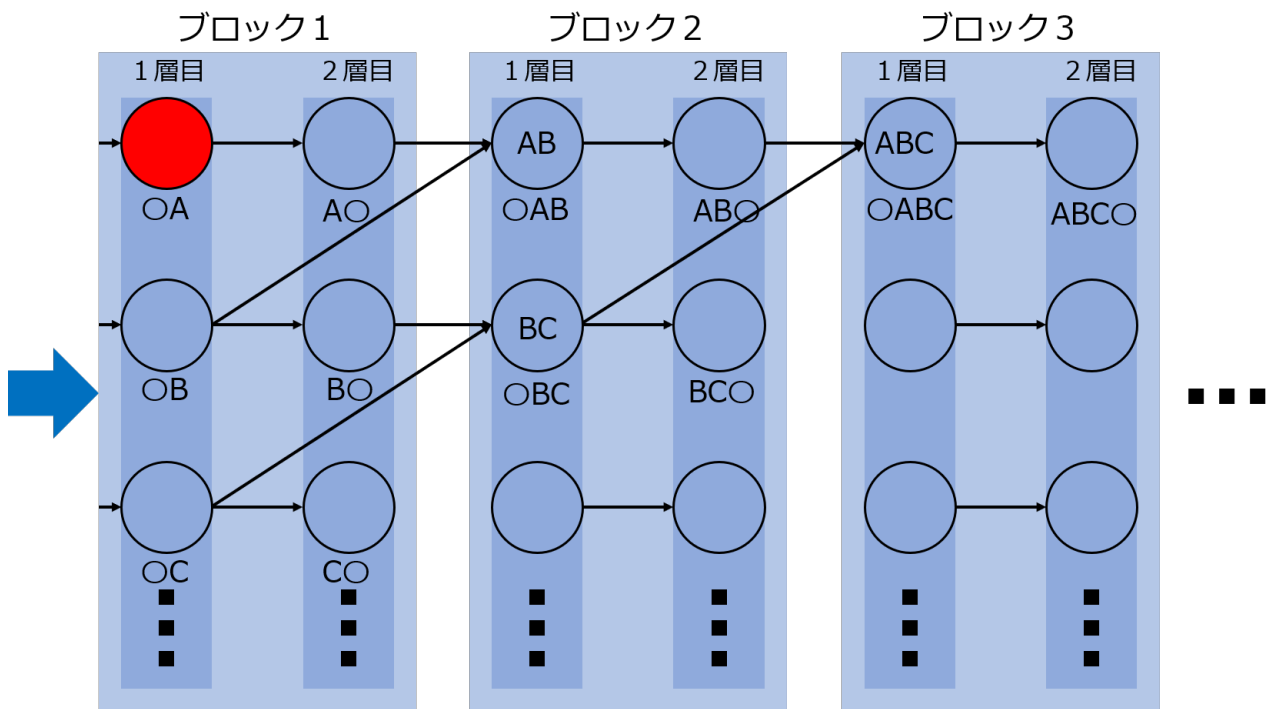


図 3.5: 抽出時のネットワークのふるまい : ステップ 2

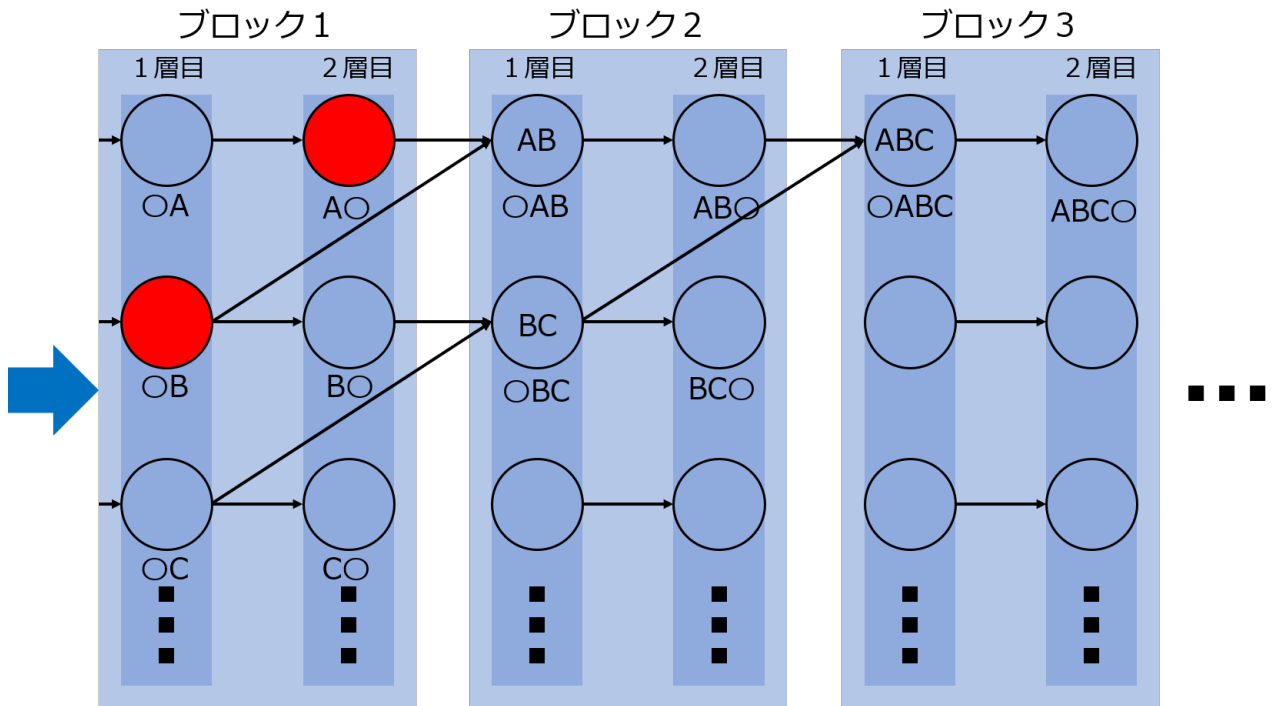


図 3.6: 抽出時のネットワークのふるまい : ステップ 3

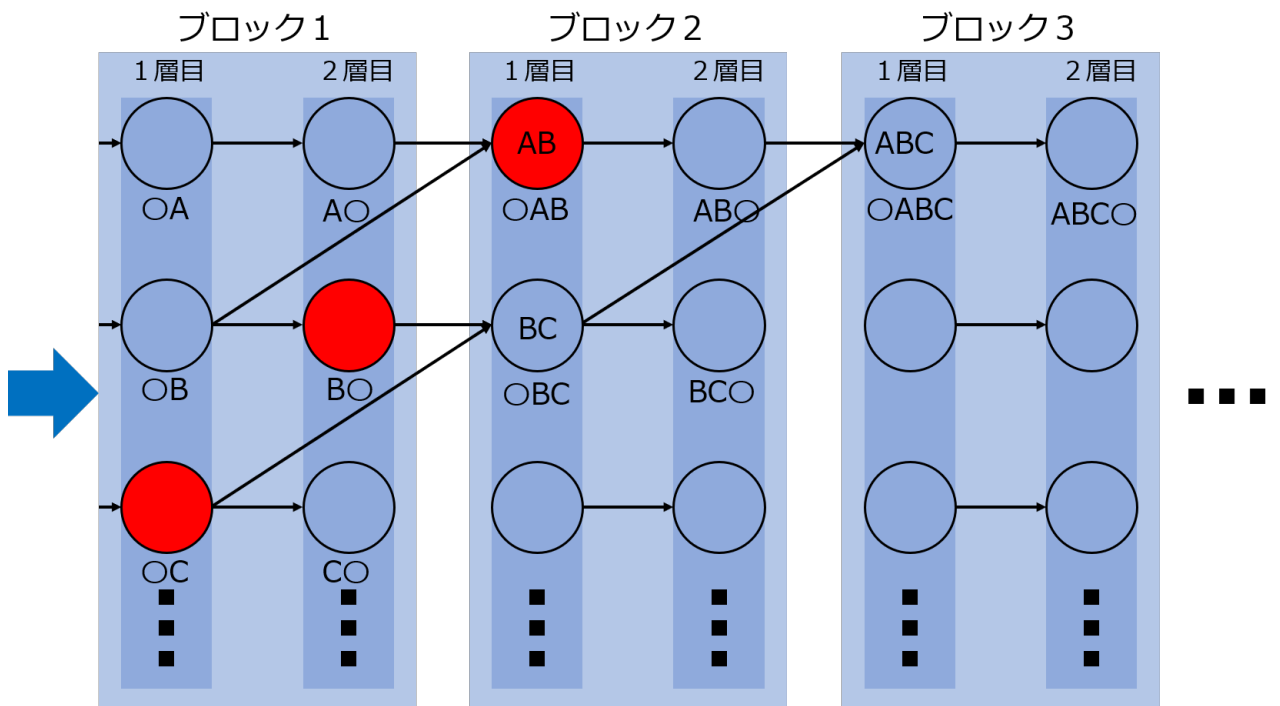


図 3.7: 抽出時のネットワークのふるまい : ステップ 4



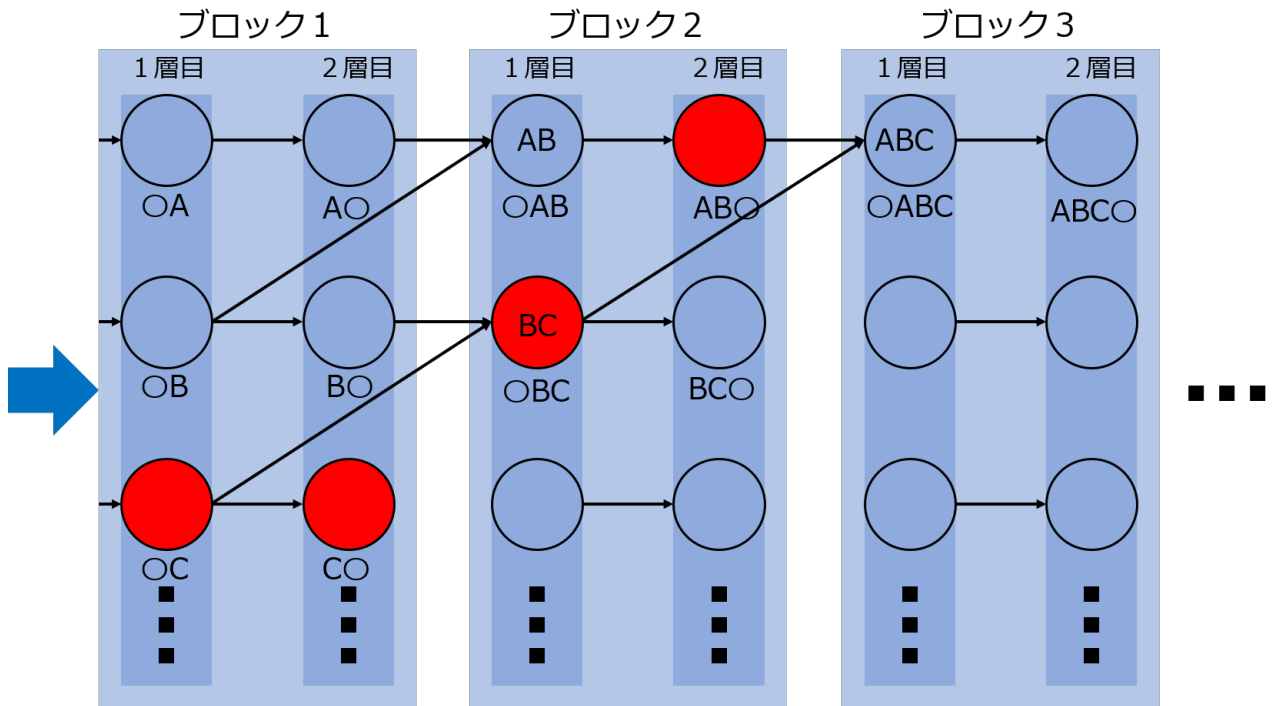


図 3.8: 抽出時のネットワークのふるまい : ステップ 5

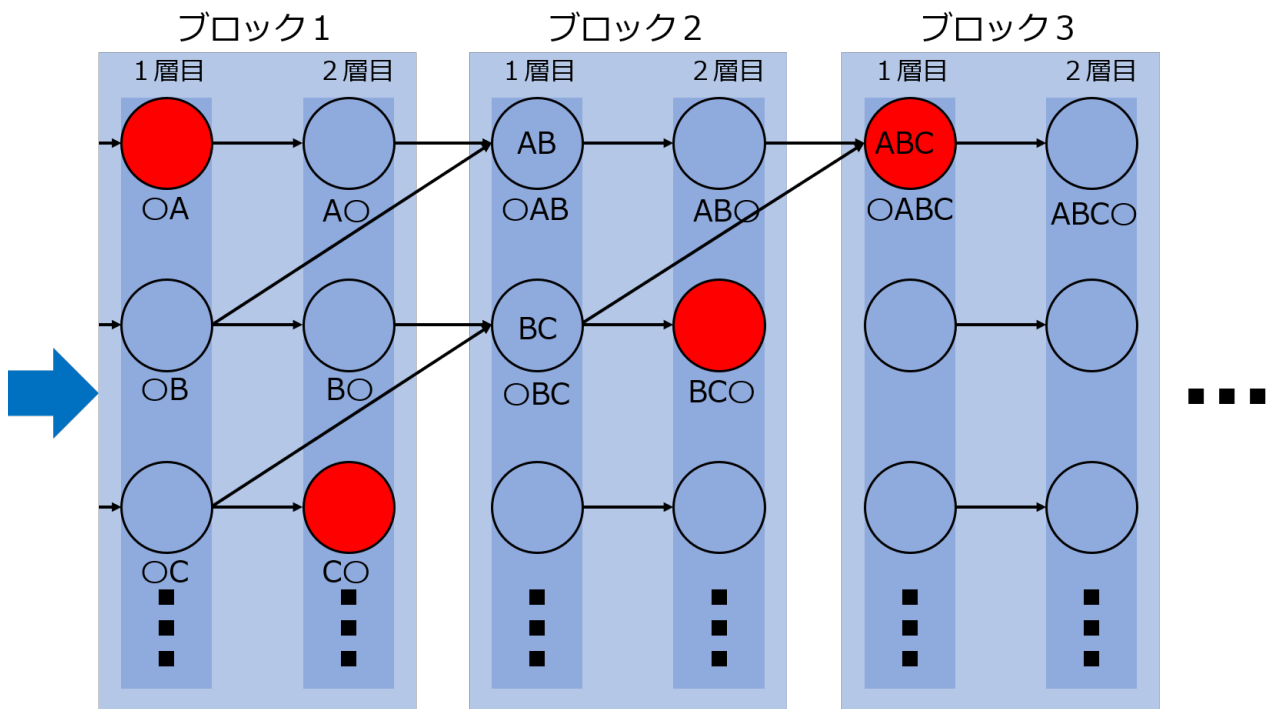


図 3.9: 抽出時のネットワークのふるまい : ステップ 6

6. 図 4.6 は 2 個目のシンボル “A” を与えた後の状態を表している。ブロック 3 の 1 層目では，“AB○” に対応するユニットと “○BC” に対応するユニットからの信号の伝搬により順序パターン “ABC” に対応するユニットが発火する。

本提案では，“AB” に対応するユニットは “AB○” と捉えることができ、既存手法の 2 シンボル長を抽出するような仕組みで抽出する。つまり “AB○” + “○BC” で “ABC” を抽出する。ブロック  $n$  のユニットが順序パターンを抽出したとき、ブロック  $n$  のユニットのペアを追加する。抽出する順序パターンの個数はわからないので、新たな順序パターンを抽出するたびこのネットワークはブロックとユニットを追加して成長する。自動成長させることで、複数の頻出な順序パターンを抽出できるようにする。初期のネットワークはブロック 1 とブロック 2 のみであり、ブロック 2 のユニットの個数は  $n_a$  とする。また、すべての結合荷重は初期値  $W_{init}$  とする。ブロック  $n$  のユニットが順序パターンを抽出した場合、ブロック  $n$  が一番最後のブロックでなければブロック  $n$  とブロック  $(n+1)$  にユニットのペアを 1 つ追加する。また、ブロック  $n$  が一番最後のブロックならば、ブロック  $(n+1)$  が追加され、ブロック  $(n+1)$  にユニットのペアを  $n_b$  個作成する。ユニットの追加は、1 層目のユニットが抽出不可能になったときにも行われる。この抽出不可能とは、 $n$  ブロックのユニットにおいて  $(n-1)$  のブロックのどちらかの層の結合荷重がすべて最小値 0 になったときのことである。

ユニットが追加された時、結合が作成される。ブロック  $n$  にユニットが追加された場合、ブロック  $n-1$  の抽出済みであるすべてのユニットと結合する。また、そのときの値は初期値  $W_{init}$  とする。

### 3.4 抽出の確認

本節では、提案するスパイキングニューラルネットワークが音声データから頻出な順序パターンを抽出できるかを確認する。抽出する順序パターンは 3 シンボル長以上をめざす。また、複数の順序パターンを抽出できるかを確認する。

#### 3.4.1 実験方法・条件

音声データから音素をシンボルとして取り出し、シンボルの出現時刻に従って、提案するネットワークに 1 シンボルずつ与える。シンボルを与えながらネットワー

表 3.1: 各シンボル長における順序パターンの出現頻度

2 シンボル長	3 シンボル長	4 シンボル長
よう (2.5)	ました (2.5)	たいよう (1.9)
まし (2.5)	たいよ (1.9)	きたかぜ (1.2)
した (2.5)	いよう (1.9)	たびびと (1.2)
たい (2.0)	きたか (1.2)	いようが (0.9)
きた (1.9)	たかぜ (1.2)	になりま (0.9)

クの結合荷重を調べ、抽出した順序パターンを確認する。また、抽出した順序パターンが、データを与え始めてから何秒後に抽出されたのかも確認する。

本実験では、だれでも使用可能な音声コーパスを音声データとして使用する。この音声コーパスは、6人の女性による日本語の音声データが含まれている。人によって音素の速さはことなるため、この6人のデータを順番に統合し、1つの音声データとした。

音声コーパスの朗読データを使用した。この朗読データの内容は、イソップ童話の北風と太陽である。統合した音声データの長さは447.785秒であった。この秒数は短く、ネットワークが十分に学習できない可能性があるため、ネットワークに音声データを5回与える。

音声データからのシンボルの取り出しは、オープンソースソフトウェアであるJuliusの音素セグメンテーションを使用した。シンボルを取り出した結果、この音声データは53種類のシンボルがあった。表3.1は、音声データの各シンボルにおける順序パターンの出現頻度を表している。4シンボル長において、最も出現頻度が高い順序パターン“たいよう”の出現頻度が1.9%である。本実験では各シンボル長において、1.9%以上の順序パターンを抽出することをめざす。

表4.3は、ネットワークの各パラメータの値を示している。これらのパラメータは次の理由により決定した。日本語における音素の出現間隔の平均が0.145秒[41]であるため、ユニットとユニットとの信号の伝搬にかかる時間を同じ0.145秒とした。ユニットのしきい値 $\theta$ 及び初期値 $V_r$ は、神経細胞の値を参考にした。ユニットの時定数 $\tau_m$ は、2シンボル分の出現間隔である0.290秒で内部電位 $V_m$ が0になるように設定した。結合荷重の初期値は、上限値 $W_{max}$ と最小値 $W_{min}$ の中間の値とした。STDP学習則のパラメーターは、試行錯誤で設定した。

表 3.2: ネットワークのパラメーター

項目	値
$V_r$	-68.0mV
$\theta$	-50.0mV
$\tau_m$	0.290 秒
伝搬にかかる時間	0.145 秒
$N_a$	1
$N_b$	1
$W_{max}$	10.0
$W_{min}$	0.0
$W_{init}$	5.0
$A_+$	5.0
$A_-$	5.0
$\tau_{stdp}$	1450 秒

### 3.4.2 実験結果

ネットワークにすべてのシンボルを与えた後，出力ブロックの各層のユニットによって抽出された順序パターンを確認した．3シンボル長に対応する層では，順序パターン“ました”，“たいよ”，“いよう”を抽出した．また，4シンボル長に対応する層では，順序パターン“たいよう”を抽出した．

提案手法は，2つのステップで長い順序パターンを抽出する．(1)短い順序パターンを抽出する．(2)短い順序パターンを組み合わせてより長い順序パターンを抽出する．本実験でこのステップで抽出されているかを確認する．2シンボル長では，47秒に順序パターン“した”，234秒に“まし”を抽出した．そして，334秒で“ました”を抽出した．同様に，提案するネットワークは順序パターン“たいよう”を抽出した．具体的は，まず“たい”，“いよ”，“よう”を抽出し，その後，“たい”と“いよ”の組み合わせで817秒に“たいよ”を抽出した．さらに，1318秒に“いよ”と“よう”の組み合わせにより“いよう”を抽出した．そして，1522秒に“たいよ”と“いよう”の組み合わせにより“たいよう”を抽出した．

これらの結果から，提案するネットワークは順序パターンのシンボル長に依存しないこと，抽出される順序パターンに個数制限をもうけないことの2つの条件

を満たすネットワークであると考えられる。また、本実験では音声データを用いており、シンボルの出現間隔にゆらぎをもっている。音声データから頻出な順序パターンを抽出することができたため、シンボルの出現タイミングにも対応できていると考えられる。

提案するネットワークは頻出な順序パターンを抽出することができた。しかし、2シンボル長において頻出としていた1.9%以下の順序パターンを抽出してしまっていた。そのため、頻出な順序パターンのみを抽出するようにネットワークをさらに拡張する必要がある。

### 3.5 むすび

本章では、2のネットワークを拡張し、本論文の抽出する際の条件である、順序パターンのシンボル長に依存しないこと、抽出される順序パターンに個数制限をもうけないことの2つの条件を満たすネットワークを提案した。また、本章では時系列シンボルとして音声データを対象とした。音声データは、シンボルの出現間隔がほぼ一定に出現するようにみえるが、出現タイミングにばらつきがある時系列シンボルである。このようにシンボルの出現間隔にゆらぎがある時系列シンボルである音声データから頻出な順序パターンを抽出することを目的とした。

音声データから頻出な順序パターンを抽出するために、基のネットワークで用いていたHebb則を変更した。音声データを時系列シンボルの対象とした場合、2での実験データよりも頻出な順序パターンの出現頻度はかなり低くなる。Hebb側は、出力ユニットが発火した時刻の1ステップ前の入力ブロックのユニットにしか結合荷重を調整しないため、頻出な順序パターンの出現頻度が5%以下の場合、結合荷重の増加が時間的減衰の減少に負けてしまい、抽出が困難となる。頻出な順序パターンの出現頻度が低くても抽出できるようにするために、Hebb則をユニットが発火した瞬間だけでなく、数ステップ前後の発火に依存して結合荷重を調整するSTDP学習に変更した。

順序パターンのシンボル長に依存しないこと、抽出される順序パターンに個数制限をもうけないことの2つの条件を満たすために、新しいネットワークの構造を提案した。3シンボル長以上の順序パターンを抽出するには、出力ブロックに出力層を追加する必要がある。しかし、単純に追加しても長い順序パターンを抽出することは困難である。そこで、安定して抽出できる2シンボル長を抽出する

メカニズムを繰り返して長い順序パターンを抽出する構造を提案した。また、ネットワークを自動成長させることで、抽出される順序パターンに個数制限をもうけないようにした。ネットワークの層とユニットは、順序パターンを抽出するたびに、自動で追加される。

提案するネットワークが音声データから頻出な順序パターンを抽出することができるかを確認するために、日本語の朗読データをネットワークに与えた。この朗読データは、4シンボル長でもっと出現頻度が高い順序パターンの出現頻度が1.9%であった。そのため、各シンボルにおいて、1.9%以上の順序パターンを抽出できるかを確認した。実験の結果、各シンボル長において、1.9%以上の順序パターンを抽出できることを確認した。

提案するネットワークは、音声データから頻出な順序パターンを抽出することができた。しかし、2シンボル長において頻出としていた1.9%以下の順序パターンを抽出してしまっていた。そのため、頻出な順序パターンのみを抽出するようにネットワークをさらに拡張する必要がある。

# 第4章 任意の出現頻度の順序パターンを抽出するニューラルネットワーク

## 4.1 まえがき

本論文では、時系列シンボルから頻出な順序パターンを抽出することを目的としている。この抽出は、人間の学習を模倣しており、頻出な順序パターンを抽出する際に、4つ条件がある。1つ目はリアルタイム処理である。これは、識別器が学習と認識を同時にすることを目的としているからである。2つ目はラベル付けをしないことである。人間の語彙の獲得は、この順序パターンが頻出であると教えられるのではなく、多くの順序パターンから自動で頻出な順序パターンを抽出し学習する。そして、3つ目はシンボルの長さに依存しないことである。語彙の長さは、2シンボル長、3シンボル長、4シンボル長とさまざまであるためである。4つ目の条件は、抽出される順序パターンに個数制限をもうけないことである。これは、各系列データには複数の順序パターンがあるが、その個数は同じではないからである。

また、時系列シンボルとして音声や動画を選んだ場合、さらに条件が加わる。それは、シンボルの出現タイミングを考慮することである。これらのデータでは、シンボルが一定間隔で出現しているように見えるが、実際にはばらつきがある。そのため、シンボルの出現間隔にゆらぎをもつ時系列シンボルから頻出な順序パターンを抽出できることが望ましい。

上記ができるスパイクニューラルネットワークを3章で提案した。音声データを時系列シンボルとしてこのネットワークに与えたところ、頻出な順序パターンを抽出することができた。しかし、頻出ではない順序パターンを抽出してしまう場合があった。

本章では、時系列シンボルから頻出な順序パターンのみを抽出することを目的

とする。しかし、頻出といっても各系列データによってその順序パターンの出現頻度はさまざまであり、同じ出現頻度でもそれを頻出とするかは異なる。そのため、どんな系列データにおいても、特定の出現頻度以上の場合、頻出な順序パターンとして抽出されるようにする。具体的には、頻出とする出現頻度のしきい値を決め、すべてのシンボル長において出現頻度がしきい値以上の順序パターンのみ抽出する。また、頻出とする出現頻度のしきい値は可変できるようにする。

本章では、3章で述べたスパイキングニューラルネットワークを改良することで、頻出とみなす出現頻度以上の順序パターンのみ抽出するスパイキングニューラルネットワークを提案する。このネットワークは本論文の集大成であるため、ネットワークの構造、ユニット、学習について説明する。

## 4.2 任意の出現頻度以上の順序パターンを抽出するスパイキングニューラルネットワーク

### 4.2.1 ネットワークの構造

提案手法の基となるスパイキングニューラルネットワーク (以下、ネットワーク) は2つのブロックから構成される。図4.1はネットワークの構成図である。ブロック  $n$  ( $n \geq 2$ ) のユニットは  $n$  シンボル長の順序パターンに対応している。各ブロックには、2つの層があり、ユニットが鎖状に接続されている。この二つの層があることで各ブロックは入力の前後関係を把握することができる。

簡易化のため、ブロック1から説明する。ブロック1は、入力端子と2つのユニットが鎖状に接続されたものがシンボルの種類数分ある。各鎖は1つのシンボルに1対1で対応している。また、各鎖の数は、新規のシンボルが与えられるたびに追加され、自動で調整される。入力ブロックでは、1層目に最後に与えられたシンボルの情報、2層目にその1ステップ前のシンボルの情報が保持されている。また、2シンボル長を抽出するユニットへの結合において、1層目は順序パターンの後ろのシンボルに対応し、2層目は順序パターンの前のシンボルに対応している。たとえば、一番上の鎖をシンボル“A”に対応する鎖だとすると、ブロック1の1層目が“○A”，2層目が“A○”に対応する。

ブロック  $n$  ( $n \geq 2$ ) では、ブロック1のようにユニットが1対1で鎖状に接続されている。また、1層目のユニットはブロック  $(n-1)$  の2つの層と結合するブ



ブロック2以降の1層目のユニットは順序パターンの抽出を行う。頻出な順序パターン“AB”があった場合、ブロック2の1層目のユニットが“A○”と“○B”に対応するユニットからの接続される結合のみが最大値になることで、抽出状態になり、順序パターン“AB”がネットワークに与えられたときのみ発火するようになる。ブロック2では、ABを抽出したユニットは“○AB”に対応し、2層目でこの接続しているユニットは“AB○”に対応する。頻出な順序パターン“ABC”があった場合、“AB○”と“○BC”に対応するユニットからの結合荷重が最大値になる。1層目のユニットは、同じ層のユニット同士で側抑制を行い、発火した場合には他のユニットへの入力を抑制する。この側抑制により、1つ順序パターンを複数のユニットが抽出することを防ぐ。

ネットワークのすべての結合には遅延があり、信号の伝搬には一定時間がかかる。その遅延時間はシンボルの入力間隔と同等である。

ブロック1のユニットは、信号を受け取るとすぐに発火し、接続先の次のユニットへ信号を伝搬させる単純なユニットモデルである。一方ブロック2以降では、1層目はスパイクニューロンであり、2層目は単純なユニットである。スパイクニューロンはモデルの一つであるLIF(Leaky Integrate and Fire)モデル[35]に基づいて発火動作をする。LIFモデルのユニットは内部電位 $V_i(t)$ をもっており、その内部電位が閾値 $\theta$ を超えると超えると発火し、次のユニットへ信号を伝搬させる。発火すると内部電位は初期値 $V_r$ へ戻る。本手法では、提案する学習則に合わせるために、LIFモデルのユニットは受容体を使ったモデル[40]を用い、発火動作は式(4.1)と式(4.2)に従う。

$$R_{ij}(t) \leftarrow R_{ij}(t) \exp\left(-\frac{t_d}{\tau_m}\right) + W_{ij}O_j(t) \quad (4.1)$$

$$\begin{cases} V_i(t) \leftarrow V_r & (\text{if } V_i(t) > \theta) \\ V_i(t) \leftarrow \sum_j R_{ij}(t) & (\text{otherwise}) \end{cases} \quad (4.2)$$

$R_{ij}(t)$ は時刻 $t$ の時のユニット $i$ にあるユニット $j$ から接続される受容体の値で $t_d$ は最後の入力を受けてからの経過時刻である。 $V_i(t)$ がしきい値 $\theta$ を超えた場合発火する。このユニットモデルは既存手法のユニットモデルに比べ、時定数を小さくしても発火することができる。

ブロック1とブロック2以降のユニットは発火頻度が異なるため、結合荷重はブロック1とブロック2以降とで異なるパラメーターで学習により調整される。ブ

ブロック  $n$  のユニットが順序パターンを抽出したとき、ブロック  $n$  のユニットのペアを追加する。

抽出する順序パターンの個数はわからないので、新たな順序パターンを抽出するたびこのネットワークはブロックとユニットを追加して成長する。初期のネットワークはブロック 1 とブロック 2 のみであり、ブロック 2 のユニットの個数は  $n_a$  とする。また、すべての結合荷重は初期値  $W_{init}$  とする。ブロック  $n$  のユニットが順序パターンを抽出した場合、ブロック  $n$  が一番最後のブロックでなければブロック  $n$  とブロック  $(n+1)$  にユニットのペアを 1 つ追加する。また、ブロック  $n$  が一番最後のブロックならば、ブロック  $(n+1)$  が追加され、ブロック  $(n+1)$  にユニットのペアを  $n_b$  個作成する。ユニットの追加は、1 層目のユニットが抽出不可能になったときにも行われる。この抽出不可能とは、 $n$  ブロックのユニットにおいて  $(n-1)$  のブロックのどちらかの層の結合荷重がすべて最小値 0 になったときのことである。

ユニットが追加された時、結合が作成される。ブロック  $n$  にユニットが追加された場合、ブロック  $n-1$  の抽出済みであるすべてのユニットと結合する。また、そのときの値は初期値  $W_{init}$  とする。

#### 4.2.2 抽出後のふるまい

このネットワークのユニットは、初期は順序パターンに対してランダムに発火するが、データのシンボルがネットワークに与えられるたびに結合荷重が調整され、頻出な順序パターンが与えられたときのみ特定の 1 つのユニットが発火するようになる。図 4.2, 図 4.3, 図 4.4, 図 4.5, 図 4.6, 図 4.7, は、順序パターン“ABC”を抽出したネットワークネットワークに“ABCCA”が入力されたときのネットワークの応答を示している。図内の入力端子は上からシンボル“A”, “B”, “C”に対応している。そして、ブロック 1 は上から“AB”, “BC”に対応し、ブロック 2 は“ABC”に対応している。初期のユニットは前の層と全結合しているが、順序パターンに反応するようになったユニットは、2 つの層から 1 本ずつのみ接続されている状態になる。以下に 6 ステップのネットワークの状態を示す。

1. 図 4.2 は“ABCCA”を与える直前の状態を示している。
2. 図 4.3 はシンボル“A”を与えた後の状態を示している。ブロック 1 の第 1 層のシンボル“○A”に対応するユニットのみが発火する。

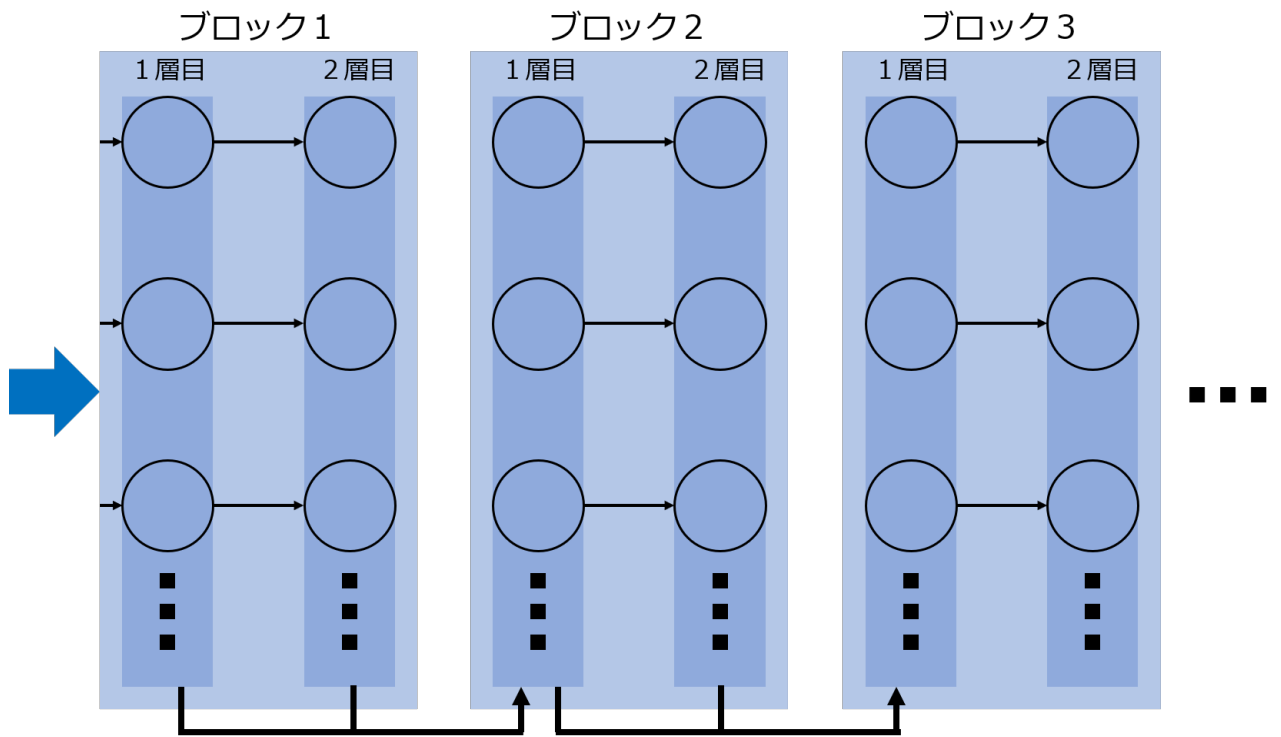


図 4.1: 提案手法のネットワークの構成

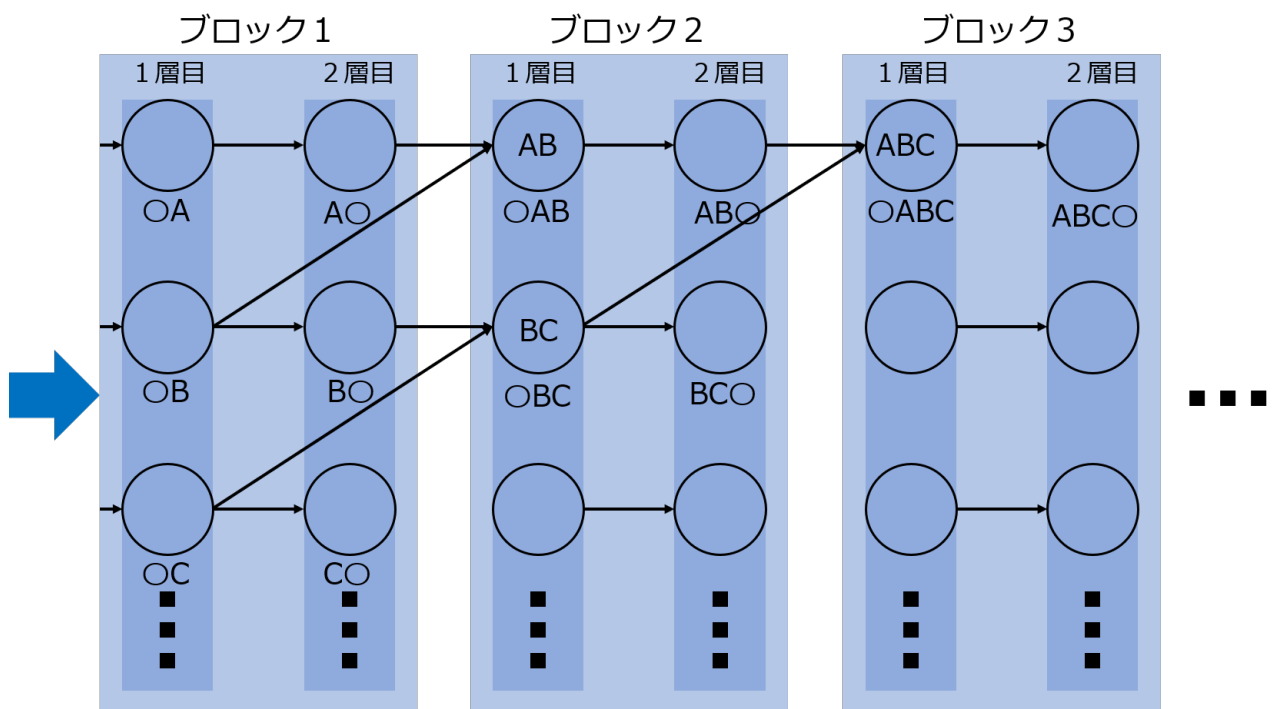


図 4.2: 抽出時のネットワークのふるまい : ステップ 1

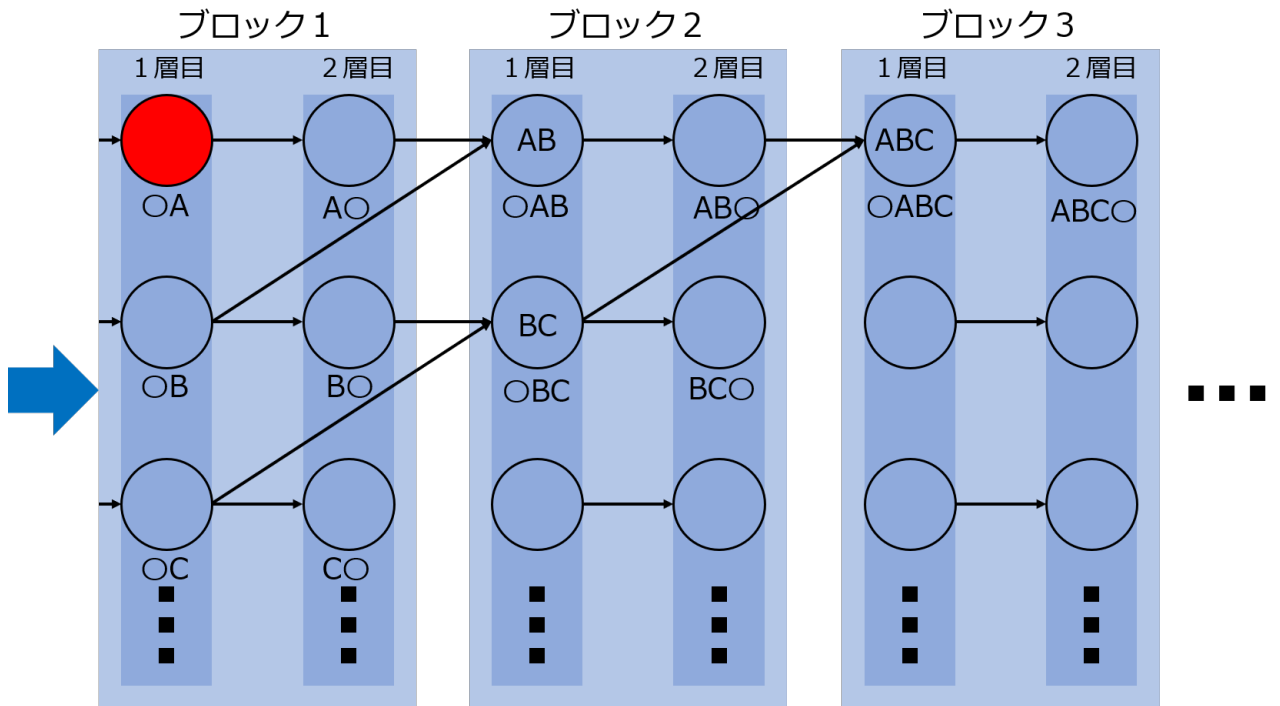


図 4.3: 抽出時のネットワークのふるまい : ステップ 2

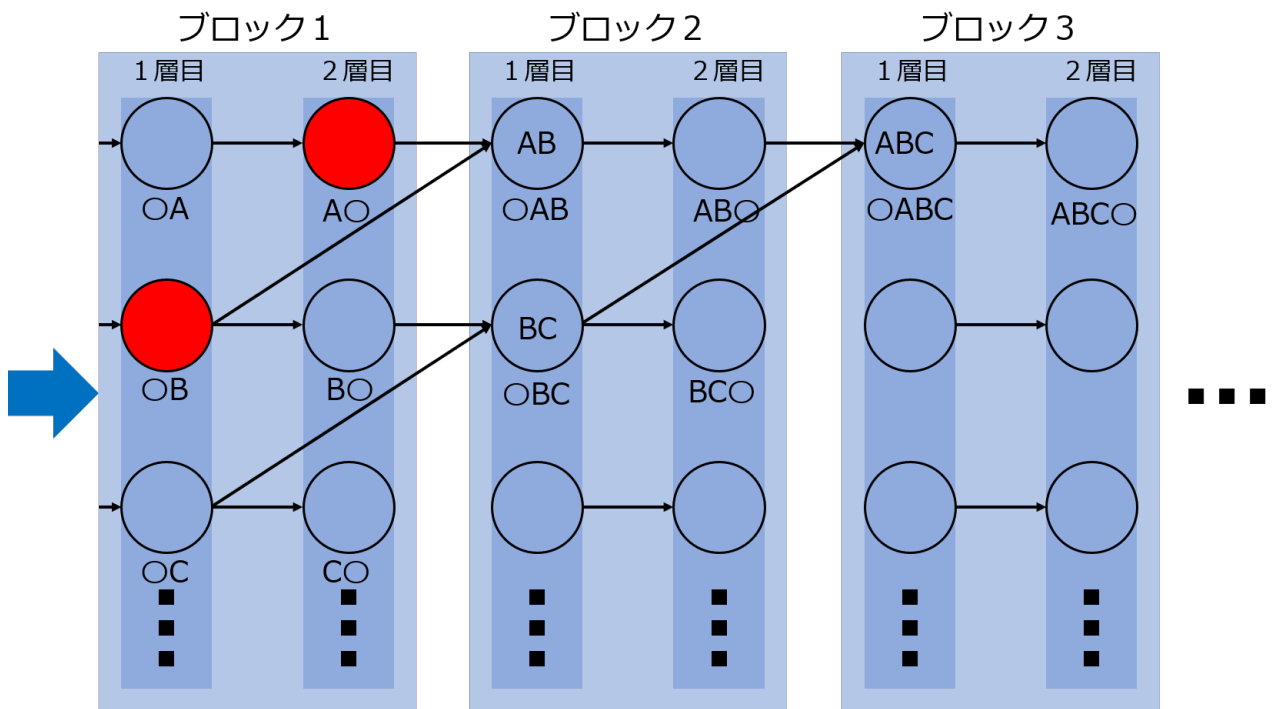


図 4.4: 抽出時のネットワークのふるまい : ステップ 3

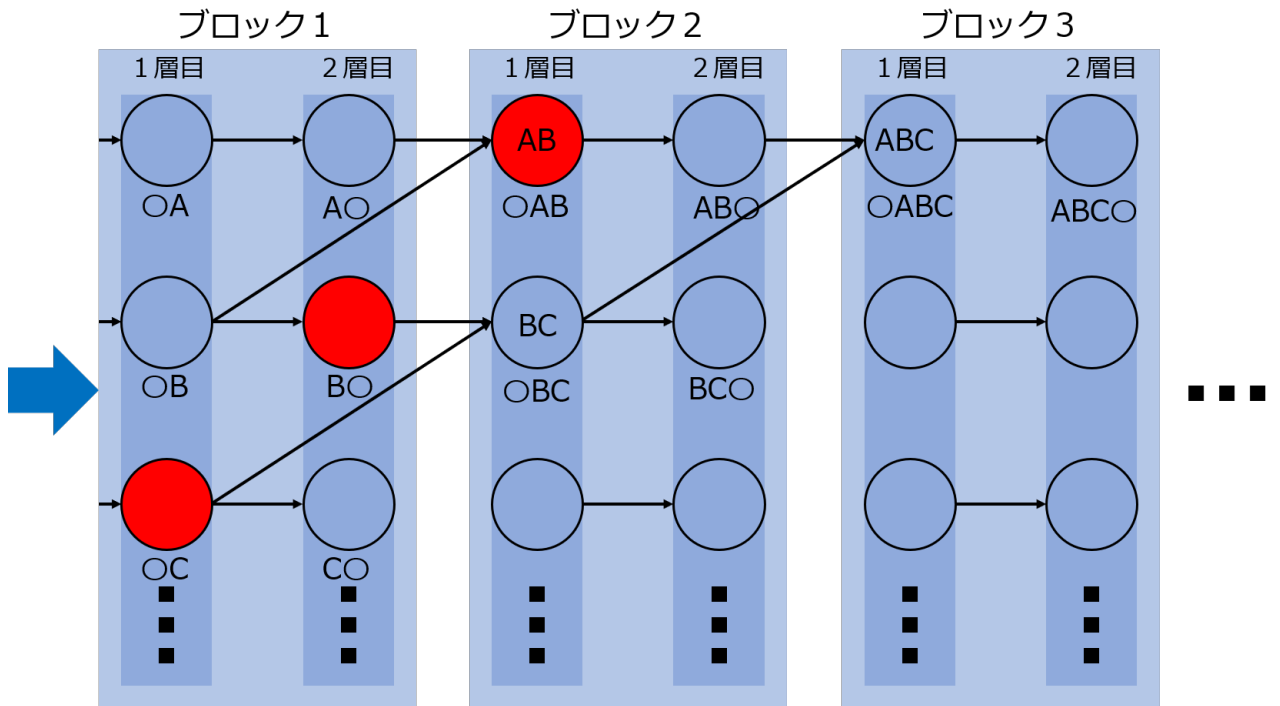


図 4.5: 抽出時のネットワークのふるまい : ステップ 4

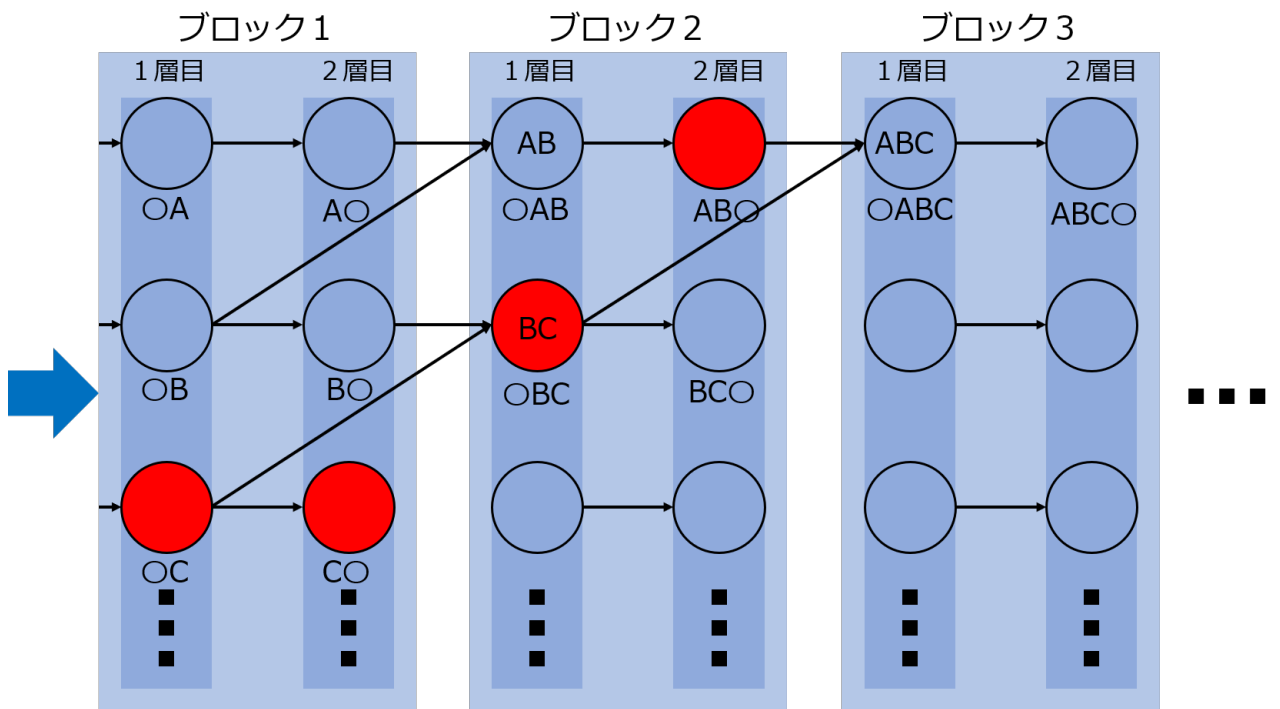


図 4.6: 抽出時のネットワークのふるまい : ステップ 5

3. 図 4.4 はシンボル “B” を与えた後の状態を示している。ブロック 1 ではシンボル “○B” に対応する第 1 層のユニットと信号の伝搬により第 2 層の “○A” に対応するユニットが発火する。
4. 図 4.5 は 1 個目のシンボル “C” を与えた後の状態を表す。ブロック 1 では第 1 層のシンボル “○C” に対応するユニットと第 2 層のシンボル “B○” に対応するユニットが発火し、ブロック 2 の第 1 層では、“A○” と “○B” に対応するユニットからの信号の伝搬により順序パターン “AB” に対応するユニットが発火する。
5. 図 4.6 は 2 個目のシンボル “C” を与えた後の状態を表している。ブロック 2 では第 1 層で順序パターン “BC” に対応するユニットが発火し、第 2 層では第 1 層の “AB○” に対応するユニットが発火する。
6. 図 4.6 は 2 個目のシンボル “A” を与えた後の状態を表している。ブロック 3 の 1 層目では、“AB○” に対応するユニットと “○B C” に対応するユニットからの信号の伝搬により順序パターン “ABC” に対応するユニットが発火する。

このように適切に学習したネットワークは頻出な順序パターンに反応する。また、ネットワークの結合荷重を調べることにより、どのユニットが何の順序パターンに反応するようになったか知ることができる

### 4.2.3 学習

提案するニューラルネットワークが 4.2.2 で述べたふるまいをするためには、学習が必要である。学習は、シンボルがネットワークに与えられるたびに行われる。

学習により、出力層の各ユニットは対応する順序パターンが入力された際にのみ発火するようにオンライン学習で自己組織化する。学習は出力ユニットに接続するすべての結合荷重を調整する。3 章のネットワークでは、この調整は STDP 学習則 [38] に基づいて行う。一般的な STDP 学習則は、式 (4.3) に従う。

$$\begin{cases} W_{ij} \leftarrow W_{ij} + A_+ \exp\left(\frac{-x}{\tau_{stdp}}\right) & (\text{if } x > 0) \\ W_{ij} \leftarrow W_{ij} - A_- \exp\left(\frac{x}{\tau_{stdp}}\right) & (\text{if } x < 0) \\ W_{ij} \leftarrow W_{ij} & (\text{if } x = 0) \end{cases} \quad (4.3)$$

$A_+$  と  $A_-$  は定数,  $x$  はユニット  $i$  とユニット  $j$  の発火時刻の差で,  $\tau_{\text{stdp}}$  は時定数である. ユニット  $j$  が発火した後にユニット  $j$  が発火した場合は結合荷重の値が増加し, 逆の順番で発火した場合は値は減少する. 増減量は  $x$  が小さいほど大きくなる.

しかし, この式は, 発火時刻との差が1ステップ分の場合と2, 3ステップ分との場合とで増減量ほぼ同じで, その差が大きくなることはない. そのため, 頻出な順序パターンの出現場所が近い場合, それら順序パターンに対応する結合荷重がほぼ同じように上昇してしまう. これを解決するために発火時刻との差が1ステップ分の場合とそれ以外との場合とで増減量の差を大きくなるように, STDP 学習則の式を改良する. 具体的には, 式 (4.4) に従う.

$$\begin{cases} W_{ij} \leftarrow W_{ij} + A_+ \exp\left(\frac{-x^2}{\tau_{\text{stdp}}}\right) & (\text{if } x > 0) \\ W_{ij} \leftarrow W_{ij} - A_- \exp\left(\frac{x^2}{\tau_{\text{stdp}}}\right) & (\text{if } x < 0) \\ W_{ij} \leftarrow W_{ij} & (\text{if } x = 0) \end{cases} \quad (4.4)$$

抽出を効率的に行うために上記に加えて以下の二つの工夫がされている.

1つ目は, 各結合荷重の値に最大値と最小値をつけることである. 本提案では, 二つの結合荷重から信号が与えられたときのみ発火するように, 工夫をする. 結合荷重に適切な最大値 ( $W_{\text{max}}$ ) を設定することにより, 抽出したユニットは二つの層から信号が与えられたときのみ発火するようにしている. また, 最小値 ( $W_{\text{min}} = 0$ ) を下回った結合荷重は  $W_{\text{min}}$  になり, その結合荷重は学習による調整がおこなわれない. 学習により, ある出力層のユニットへ接続している結合荷重の1つが最大値に達した場合, その接続元であるユニットと同じ層に所属しているユニットからその出力層のユニットへの他の結合荷重の値をすべて最小値にする. ある出力層のユニットへ接続している二つの層からの最大値に達した結合荷重ができたとき, そのユニットは順序パターンを抽出した状態となり, 特定の順序でシンボルがネットワークに与えられたときのみ発火するようになる. 抽出した出力層のユニットへ接続している結合荷重は学習しない.

2つ目は, 結合荷重をリセットすることである. 片方の層の結合荷重が最大値に達した場合, もう片方の層の結合荷重をリセットするとにより最大値に達したペアとの順序パターンを捨てるようにする. リセットが無い場合, つぎのような問題が発生する. ある系列データの頻出な順序パターンが “AB” と “BC” だったとする. この場合, 2シンボル前の A, 1シンボル前および2シンボル前の B, 1シ

ンボル前の C に対応するユニットからの結合荷重が増加する。このような状況下で、順序パターン AC が入力されても、当該ユニットは発火してしまい、場合によっては、その結果順序パターン AC を抽出するようになる。しかし、ユニットの時定数を小さくし、片方の結合荷重をリセットすることでこの問題を防ぐことができる。ユニットの時定数を小さくしていれば、“OC” に対応する結合荷重から信号が来たときのみ発火することができる。“AC” と “BC” がそれぞれ与えられたときに発火するが、“BC” のほうが “AC” より頻出なので、“BO” に対応する結合荷重のほうが上昇しやすく、先に最大値に達することができる。このリセットは、式 (4.5) に従う。

$$\begin{cases} W_{ij} \leftarrow W_{max}/2 & (\text{if } W_{ij} > W_{max}/2) \\ W_{ij} \leftarrow W_{min} & (\text{if } W_{ij} < W_{max}/2) \end{cases} \quad (4.5)$$

頻出とする出現頻度のしきい値を決め、そのしきい値以上の順序パターンのみ抽出するために、時間的減衰を行う。STDP 学習則による結合荷重の上昇する速さは、順序パターンの出現頻度に依存する。そこで、ユニットが順序パターンを抽出するのに制限時間をかけることで、出現頻度が低い順序パターンの抽出を防ぐ。具体的には式 (4.6), 式 (4.7) を適用させる。

$$x_{lim} = t - t_{lim} \quad (4.6)$$

$$W_{ij} \leftarrow W_{ij} \times \left(1 - \exp\left(\frac{x_{lim} - \tau a_{lim}}{\tau b_{lim}}\right)\right) \quad (4.7)$$

$t$  は現在時刻で、 $t_{lim}$  は接続元の片方の結合荷重が最大値に達した時刻である。ただし最大値に達してない場合、 $t_{lim}$  はユニットが作成された時刻とする。

式 (4.6), 式 (4.7) に合わせるために従来の時間的減衰の式を変更する。式 (4.6), 式 (4.7) は、頻出なシンボルに対しては結合荷重の減少を少なく、そうではない結合荷重は多く時間的減衰を行うのが望ましい。頻出ではない順序パターンにおいて、その順序パターンの各シンボルの出現頻度は低い。そこで、一定時間に 1 回も現れていなければそのシンボルからの結合を 0 にする。発火していない時間に従って徐々に結合荷重が減少するように時間的減衰の式を以下の式とする。

$$W_{ij} \leftarrow W_{ij} \times \left(1 - \exp\left(\frac{x_{exp} - \tau a_{exp}}{\tau b_{exp}}\right)\right) \quad (4.8)$$

$\tau a_{exp}$ ,  $\tau b_{exp}$  は時定数で、発火していない時刻  $x_{exp}$  が  $\tau a_{exp}$  に近づけば近づくほど結合荷重が 0 に近づく。



## 4.3 抽出の確認

### 4.3.1 実験目的

本章では、実データを用いた実験により、提案手法の有効性を確認する。そのために、2つの実験を行う。1つ目は、すべてのシンボル長において頻出な順序パターンのみを抽出することができるかである。2つ目は、抽出する順序パターンの出現頻度のしきい値を調整できるかである。

### 4.3.2 頻出な順序パターンのみを抽出できるかの確認

#### 実験条件

本節では、提案手法の有効性の確認に、系列データとしてテキストデータを用いた。用いたテキストデータは2種類で、データ1はイソップ童話の北風と太陽のテキストデータで、データ2は対話データのテキストデータである。

表4.1、表4.2はそれぞれデータ1とデータ2の各シンボル長の頻度が高い上位の順序パターンの出現率を表している。それぞれの表の各順序パターンの右にある括弧内の数値はその順序パターンの出現率(%)を表している。なお、同じ出現率の場合、順序パターンの並びは順不同である。データ1のシンボル種は55種で、データ2のシンボル種は44種であった。

本確認では、頻出とするしきい値を1.70%とする。ネットワークの結合荷重などのパラメータは表4.3とした。各ブロックにおける学習則のパラメータについては表4.4とした。

#### 抽出結果

表6はデータ1を与えたとき、表7はデータ2を与えたときのしきい値1.7%の学習パラメータでの抽出結果である。括弧内の数値は、その順序パターンを抽出したときの時刻を示している(ネットワークに最初のシンボルを与えたときを0として)。順序パターンの並びは、上からそのユニットが作成された順番に並んでいる。2つのデータににおいてどちらとも、5シンボル長の順序パターンは1つも抽出されなかった。頻出な順序パターンは数百秒間隔で抽出されている、最後に抽出された順序パターンから1000秒以上シンボルを与え続けたが、しきい値未満の順序パターンが抽出されることはなかった

表 4.1: データ 1 の各シンボル長における出現頻度

1 シンボル長	2 シンボル長	3 シンボル長	4 シンボル長	5 シンボル長
た (7.84)	よう (2.33)	ました (2.33)	ました。(2.34)	たいようが (0.88)
と (5.81)	まし (2.33)	した。(2.33)	たいよう (1.75)	になりまし (0.88)
い (4.94)	した (2.33)	たいよ (1.75)	きたかぜ (1.17)	なりました (0.88)
か (4.65)	た。(2.33)	いよう (1.75)	たびびと (1.17)	りました。(0.88)
し (4.65)	きた (1.74)	きたか (1.16)	いようが (0.87)	、たびびと (0.88)
う (4.06)	たい (1.74)	たかぜ (1.16)	になりま (0.87)	がいとうを (0.88)
ま (3.77)	いよ (1.74)	たびび (1.16)	なりまし (0.87)	たびびとは (0.88)
、 (3.48)	たか (1.45)	びびと (1.16)	りました (0.87)	きたかぜと (0.58)
く (3.19)	かぜ (1.45)	ようが (0.87)	、たびび (0.87)	たかぜとた (0.58)
。(3.19)	から (1.16)	になり (0.87)	がいとう (0.87)	かぜとたい (0.58)
は (2.90)	、た (1.16)	なりま (0.87)	いとうを (0.87)	ぜとたいよ (0.58)
き (2.61)	たび (1.16)	りまし (0.87)	びびとは (0.87)	とたいよう (0.58)
よ (2.61)	びび (1.16)	、たび (0.87)	たかぜと (0.58)	ちからくら (0.58)
が (2.61)	びと (1.16)	がいと (0.87)	かぜとた (0.58)	からくらべ (0.58)
を (2.61)	うを (1.16)	いとう (0.87)	ぜとたい (0.58)	いとうをぬ (0.58)
あ (2.32)	ると (0.87)	とうを (0.87)	とたいよ (0.58)	よう。」と (0.58)
て (2.32)	うが (0.87)	。」と (0.87)	ちからく (0.58)	はじめまし (0.58)
び (2.32)	にな (0.87)	びとは (0.87)	からくら (0.58)	じめました (0.58)
こ (2.03)	なり (0.87)	かぜと (0.58)	らくらべ (0.58)	めました。(0.58)
り (2.03)	りま (0.87)	ぜとた (0.58)	とうをぬ (0.58)	はげしくふ (0.58)
ら (1.74)	がい (0.87)	とたい (0.58)	よう。」 (0.58)	びびとは、 (0.58)
な (1.74)	いと (0.87)	ちから (0.58)	う。」と (0.58)	ました。こ (0.58)
っ (1.45)	とう (0.87)	からく (0.58)	はじめま (0.58)	はたいよう (0.58)
ぜ (1.45)	。」 (0.87)	らくら (0.58)	じめまし (0.58)	たいようの (0.58)
る (1.45)	」と (0.87)	くらべ (0.58)	めました (0.58)	ると、たび (0.58)

表 4.2: データ 2 の各シンボル長における出現頻度

1 シンボル長	2 シンボル長	3 シンボル長	4 シンボル長	5 シンボル長
い (7.40)	か? (6.21)	すか? (5.00)	ですか? (3.14)	か? そのほ (1.26)
か (7.40)	すか (4.96)	ですか (3.12)	そのほん (1.88)	? そのほん (1.26)
ん (6.79)	ほん (3.10)	か? そ (1.87)	ますか? (1.88)	そのほんを (1.26)
? (6.17)	です (3.10)	そのほ (1.87)	あります (1.25)	ましたか? (1.26)
す (5.55)	ます (2.48)	のほん (1.87)	か? その (1.25)	したか? そ (1.26)
は (4.32)	? そ (1.86)	ますか (1.87)	? そのほ (1.25)	いますか? (1.26)
し (4.32)	その (1.86)	ありま (1.25)	のほんを (1.25)	いとおもい (1.26)
ま (3.70)	のほ (1.86)	ります (1.25)	ましたか (1.25)	とおもいま (1.26)
ほ (3.08)	いま (1.86)	? その (1.25)	したか? (1.25)	ここにほん (0.63)
そ (3.08)	おも (1.86)	ほんを (1.25)	たか? そ (1.25)	ここにほんが (0.63)
で (3.08)	んが (1.24)	ました (1.25)	いますか (1.25)	にほんがい (0.63)
お (3.08)	あり (1.24)	したか (1.25)	いとおも (1.25)	ほんがいつ (0.63)
た (3.08)	りま (1.24)	たか? (1.25)	とおもい (1.25)	いっさつあ (0.63)
も (2.46)	それ (1.24)	います (1.25)	おもいま (1.25)	っさつあり (0.63)
あ (2.46)	んは (1.24)	いとお (1.25)	ここにほ (0.62)	さつありま (0.63)
の (2.46)	しゃ (1.24)	とおも (1.25)	ここにほん (0.62)	つあります (0.63)
こ (1.85)	んを (1.24)	おもい (1.25)	にほんが (0.62)	ありますそ (0.63)
に (1.85)	まし (1.24)	もいま (1.25)	ほんがい (0.62)	りますそれ (0.63)
が (1.85)	した (1.24)	ここに (0.62)	んがいつ (0.62)	ますそれは (0.63)
れ (1.85)	たか (1.24)	ここにほ (0.62)	がいつさ (0.62)	すそれはあ (0.63)
な (1.85)	かい (1.24)	にほん (0.62)	いっさつ (0.62)	それはあつ (0.63)
と (1.85)	いと (1.24)	ほんが (0.62)	っさつあ (0.62)	れはあつい (0.63)
っ (1.23)	とお (1.24)	んがい (0.62)	さつあり (0.62)	はあついほ (0.63)
さ (1.23)	もい (1.24)	がいつ (0.62)	つありま (0.62)	あついほん (0.63)
つ (1.23)	ここ (0.62)	いっさ (0.62)	りますそ (0.62)	ついほんで (0.63)

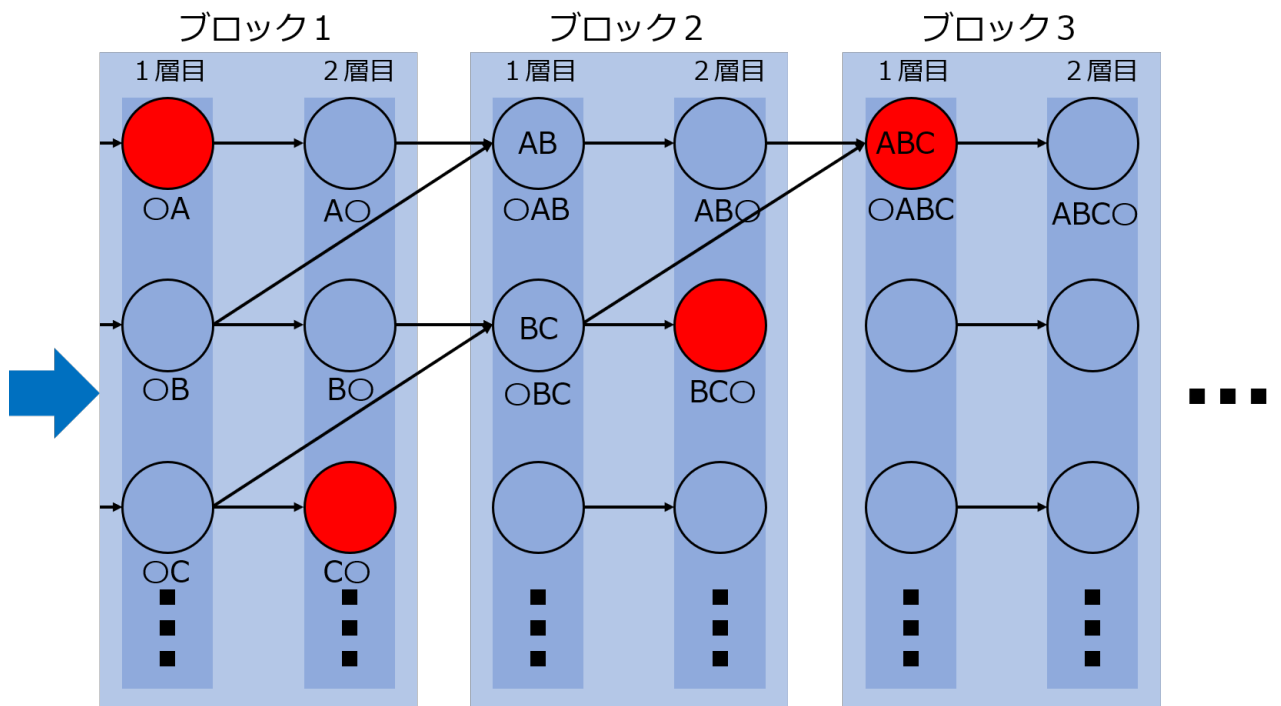


図 4.7: 抽出時のネットワークのふるまい : ステップ 6

表 4.3: 実験 1 におけるネットワークのパラメーター

項目	値
$W_{init}$	7.5
$W_{max}$	10.0
$\theta$	-55.0
$V_r$	-68.0
$R_{max}$	10
$\tau_m$	25.0
$N_a$	20
$N_b$	1

表 4.4: 実験 1 における学習則のパラメーター

項目	値	
	1 ブロック 1-2 間	ブロック 2 以降
$A_+$	0.15	0.30
$A_-$	0.15	0.30
$\tau_{stdp}$	15000	15000
$\tau a_{exp}$	30000	60000
$\tau b_{exp}$	3000	3000
$\tau a_{lim}$	500000	125000
$\tau b_{lim}$	1000	1000

表 4.5: データ 1 における抽出結果

2 シンボル長	3 シンボル長	4 シンボル長	5 シンボル長
た。(335.6)	した。(468.9)	ました。(673.6)	-
した(398.1)	ました(600.0)	たいよう(1035.5)	-
まし(526.6)	たいよ(847.6)	-	-
たい(744.2)	いよう(949.5)	-	-
よう(749.1)	-	-	-
きた(871.0)	-	-	-

表 4.6: データ 2 における抽出結果

2 シンボル長	3 シンボル長	4 シンボル長	5 シンボル長
すか (181.3)	すか? (230.1)	ですか? (508.2)	-
か? (194.5)	ですか (456.8)	そのほん (1029.6)	-
いま (485.8)	のほん (818.9)	ますか? (1988.1)	-
です (395.5)	そのほ (938.4)	-	-
ほん (473.4)	か?そ (1467.8)	-	-
のほ (727.7)	ますか (1892.4)	-	-
その (845.8)	-	-	-
おも (678.2)	-	-	-
?そ (137.5)	-	-	-
ます (1798.1)	-	-	-

表 4.7: 実験 1 における学習則のパラメーター

項目	値	
	1 ブロック 1-2 間	ブロック 2 以降
$\tau a_{lim}$	300000	70000
$\tau b_{lim}$	1000	1000

### 4.3.3 パラメーターによるしきい値の設定

#### 実験条件

本提案は時間的減衰により、しきい値の調整ができるとしている。その有効性を確認するために、時間的減衰のパラメーター  $\tau a_{lim}$  だけを変更して、実験 1 と同様の実験を行う。

各ブロックにおける学習則のパラメータについては表 4.7 とした。

#### 抽出結果

表 4.8 はデータ 1 を与えたとき、表 4.9 はデータ 2 を与えたときの抽出結果である。どちらのデータにおいても、各シンボル長にて出現頻度が 2.3% 以上の順序パ

表 4.8: データ 1 における抽出結果

2 シンボル長	3 シンボル長	4 シンボル長	5 シンボル長
た。(335.6)	ました (1622.0)	ました。(1845.0)	-
まし (526.6)	した。(1774.3)	-	-
よう (1286.2)	-	-	-
した (1548.0)	-	-	-

表 4.9: データ 2 における抽出結果

2 シンボル長	3 シンボル長	4 シンボル長	5 シンボル長
すか (181.3)	すか? (230.1)	ですか? (508.2)	-
か? (194.5)	ですか (456.8)	-	-
です (395.5)	-	-	-
ほん (473.4)	-	-	-
ます (2462.3)	-	-	-

ターンのみを抽出している。

#### 4.3.4 考察

4.3.2の実験では、2つのデータとも、頻出であると決めた出現頻度が1.7%以上の順序パターンをすべて抽出することができている。また、シンボルを与え続けても1.7%未満の順序パターンを抽出することはなかった。この結果から、提案するネットワーク頻出な順序パターンのみを抽出できていると考える。

4.3.2の実験では、1.7%以上の順序パターンのみを抽出している。4.3.3の実験では、2.3%以上の順序パターンのみを抽出している。これらの実験の違いはパラメーター $\tau_{a_{lim}}$ だけである。したがって、学習則のパラメーター $\tau_{a_{lim}}$ だけを変更するだけで、頻出とするしきい値を調整することができる。提案手法の抽出するのに制限時間を設ける工夫が有効であると考えられる。

以上から、本章で提案するネットワークは、頻出とみなす出現頻度以上の順序パターンのみ抽出することができると考えられる。また、本論文では、人間の学習を模倣した抽出をめざしており、頻出な順序パターンを抽出する際に、4つ条件がある。これらの4つの条件に対しても満たしていると考えられる。

## 4.4 むすび

本論文では、時系列シンボルから頻出な順序パターンを抽出することを目的としている。この抽出は、人間の学習を模倣しており、頻出な順序パターンを抽出する際に、4つ条件がある。それは、リアルタイム処理、ラベル付けをしない、さまざまなシンボル長の順序パターンを抽出、抽出される順序パターンに個数制限をもうけないことである。また、音声や動画を時系列シンボルと対象とした場合、シンボルの出現タイミングを考慮する必要がある。そのため、シンボルの出現間隔にゆらぎをもつ時系列シンボルから頻出な順序パターンを抽出できることが望ましい。これらの条件を満たすスパイキングニューラルネットワークを3で提案したが、頻出ではない順序パターンを抽出してしまう場合があった。

本章では、時系列シンボルから頻出な順序パターンのみを抽出することを目的とした。しかし、頻出といっても各系列データによって頻出とみなす出現頻度がことなる。そのため、どんな系列データにおいても、特定の出現頻度以上の場合、



すべてのシンボル長において頻出な順序パターンとして抽出されるようにし、頻出とする出現頻度のしきい値は可変できるようにすることをめざした。

本章では、すべてのシンボル長において頻出な順序パターンを抽出するために、スパイクングニューラルネットワークを用いる手法を提案した。スパイクングニューラルネットワークを用いたことで、1つ目の条件であるリアルタイム処理に対応した。学習については、教師なし学習の一種である STDP 学習則に基づいた学習法を用いることで、2つ目の条件のラベル付けをしないで頻出な順序パターンを抽出する。3つの条件に対しては、スパイクングニューラルネットワークの構造を特殊な階層構造にすることで対応した。提案するスパイクングニューラルネットワークは2つの層をもつブロックが階層構造に積みあがった構造をしている。これらのブロックはそれぞれのシンボル長の順序パターンを抽出する。ブロック内の2つの層がパターンの前後関係を捉えることができるため、順序パターンを抽出することができる。4つ目の条件である抽出される順序パターンに個数制限に対しては、スパイクングニューラルネットワークのユニットの個数を学習に応じて増やしていくことで対応した。最後に、抽出する順序パターンの出現頻度のしきい値を調整できるようにするために、学習則に工夫をした。具体的には、抽出にかかる時間に制限時間をつけられるように、時間的減衰をつけた。この時間的減衰のパラメータを変えることで、任意の出現頻度以上の順序パターンを抽出することができる。

提案するネットワークの有効性を確認するために、2つのテキストデータを用いて、2つの実験を行った。1つ目は、すべてのシンボル長において頻出な順序パターンのみを抽出することができるかである。2つ目は、抽出する順序パターンの出現頻度のしきい値を調整できるかである。1つ目の実験においては、頻出とする出現頻度を1.7%として、行った。その結果、各シンボル長において、1.7%以上の順序パターンのみを抽出したことを確認した。2つ目の実験では、1つ目の実験でのパラメータを1つだけ変更して行った。その結果、各シンボル長において、2.3%以上の順序パターンのみを抽出したことを確認し、パラメータ1つを変更するだけで頻出とするしきい値を調整できることを確認した。

## 第5章 総論

### 5.1 本研究で得られた成果

本研究の目的は、系列データから頻出な順序パターンを抽出することである。

近年、人工知能に関する研究が盛んに行われている。人工知能技術には、機械に音声の内容を理解させる音声認識技術や、提示された画像に映る物体名を答える画像認識、データ集合から頻出パターンを抽出する頻出パターンマイニングなどがある。

人工知能技術は、主に機械学習と呼ばれる手段が用いられる。機械学習は、大量のデータを機械に与えることで、特定のルールやパターンを発見したり、未知のデータに対して分類や識別または予測をすることができる手法である。機械学習において、ディープラーニング(深層学習)と呼ばれる手法があり、この技術により画像認識技術は急速な発展をした。このディープラーニングを動画認識などの時系列データの問題に適応させる研究が行われている。しかし、これらの認識精度は画像認識のように高い認識率ではなかった。それは、時間的な特徴を自動で抽出するメカニズムがなかったためである。

時間的な特徴を自動で抽出するために、人間の語彙の獲得に注目する。人間の偶発的な語彙獲得について調査した論文によると、出現頻度が高い語彙を獲得しやすいことがわかった。音素をシンボルとすると語彙はシンボルの順序パターンと捉えることができる。このように、ストリーミング形式で与えられるシンボル情報から、頻繁に出現する順序パターンを抽出することができれば、時系列特徴をネットワークが学習することができるようになる。

本論文では、系列データからすべてのシンボル長において頻出な順序パターンのみを抽出することを目的とする。またその際、人間の語彙の獲得に合わせて条件が4つある。1つ目はリアルタイム処理である。これは、識別器が学習と認識を同時にすることを目的としているからである。2つ目はラベル付けをしないことである。人間の語彙の獲得は、教師あり学習のようにこの順序パターンが頻出で

あると教えられるのではなく、多くの順序パターンから自動で頻出な順序パターンを抽出し学習する。そして、3つ目はシンボルの長さに依存しないことである。語彙の長さは、2シンボル長、3シンボル長、4シンボル長とさまざまであるためである。4つ目の条件は、抽出される順序パターンに個数制限をもうけないことである。これは、各系列データには複数の順序パターンがあるが、その個数は同じではないからである。4つの条件に加えさらに、抽出する順序パターンの出現頻度のしきい値を調整できるようにする。各系列データによってその順序パターンの出現頻度はさまざまであり、同じ出現頻度でもそれを頻出とするかは異なる。そのため、どんな系列データにおいても、特定の出現頻度以上の場合、頻出な順序パターンとして抽出されるようにする。

本論文の目的に対して、スパイキングニューラルネットワークによる手法を提案する。スパイキングニューラルネットワークを用いることで、1シンボルずつシンボルが与えられることができ、1つ目の条件であるリアルタイム処理に対応することができる。また、このスパイキングニューラルネットワークを教師なし学習で自己組織化することで、2つ目の条件のラベル付けをしないで頻出な順序パターンを抽出することができる。3つの条件に対しては、スパイキングニューラルネットワークの構造を特殊な階層構造にすることで対応する。4つ目の条件である抽出される順序パターンに個数制限に対しては、スパイキングニューラルネットワークのユニットの個数を学習に応じて増やしていくことで対応する。最後に、抽出する順序パターンの出現頻度のしきい値を調整できるようにするために、教師なし学習の1種であるSTDP学習則に工夫をする。

提案するスパイキングニューラルネットワークは、いくつかのブロックが階層的に積みあがった構造になっている。最下層のブロックのユニットは1シンボル長に対応し、上がるごとに長いシンボル長に対応している。各ブロックは2つの層をもっており、1層目が一つ前に入力された情報を、2層目がそのさらに1つ前に入力された情報を保っている。各ブロックの1層目のユニットは1つ下のブロックの2つの層のユニットと結合しており、下のブロックで頻繁に発火するパターンに対して発火するようになる。各ユニットは1つにつき1つの順序パターンに対応している。そのため、ユニットを増やすことができるため、複数の順序パターンを抽出することができる。各ブロックは学習により、自動的に増えるため、さまざまな長さの順序パターンを抽出することができる。

学習はシンボルが与えられるたびに行われる。学習は、STDP学習則に基づいて、1つのユニットが1つの頻出な順序パターンが、スパイキングニューラルネット

トワークに与えられたときのみ反応するようになる。STDP 学習則は、ユニットが発火することで結合荷重を調整する。そのため、頻出な順序パターンの出現頻度が高いほど、その学習は速い。そこで、時間的減衰という工夫を取り入れることで、ユニットが抽出にかかる時間に制限をかける。それにより、出現頻度が低い順序パターンの抽出を防ぐことができる。この時間的減衰のパラメーターを調整することにより、しきい値の調整をすることができる。

提案したスパイクングニューラルネットワークの有効性を確認するために、2つの実験を行った。1つ目は、すべてのシンボル長において頻出な順序パターンのみを抽出することができるかである。本実験では2種類のテキストデータを用意した。同じパラメーターのスパイクングニューラルネットワークにこれらのテキストデータを与えたところ、どちらのデータにおいても各シンボル長で出現頻度が1.7%以上の順序パターンのみが抽出された。2つ目は、抽出する順序パターンの出現頻度のしきい値を調整できるかである。1つ目の実験において、時間的減衰のパラメーターだけを変更して、再度実験を行った。その結果、どちらのデータにおいても2.3%以上の順序パターンのみが抽出された。この結果から、提案するスパイクングニューラルネットワークは、系列データからすべてのシンボル長において頻出な順序パターンのみを抽出ことができ、任意のしきい値以上の順序パターンを抽出できると考える。

## 5.2 今後の課題

本研究では、系列データからすべてのシンボル長において頻出な順序パターンのみを抽出することを目的とした。実際に時系列シンボルから頻出な順序パターンを抽出する際に、課題がある。それは、抽出した頻出な順序パターンが、意味のある順序パターンであるかである。たとえば、“みかん”といった順序パターンがあったとき、順序パターンの構成順序パターンである“みか”や“かん”なども抽出される。“みかん”という順序パターンは、単語として意味がある順序パターンであるが、“みか”などは場合によっては単語とは限らない。このように、抽出した順序パターンが、意味のある順序パターンであるか判断する方法が今後必要となる。

今後の新しい応用例がいくつか考えられる。例えば、コンビナート企業などの化学プロセスで使用されている時系列で与えられる温度、湿度、流量、ガス濃度、

液体濃度などのデータ処理についての応用が考えられる。このような時系列信号を使用した予測システムでは、人間の見落としとしていた変化を本システムで見出される可能性もあり、今後の発展に寄与すると思われる。また、患者のバイタルなど、老人ホームに入居している高齢者の健康状態をモニタリングする必要がある、体温、脳波、血圧などの各種バイタル信号の時系列データから健康状態を推定することが重要である。本研究を今後これらのバイタルデータに応用することが日本の高齢社会に貢献すると考えられる。

## 謝辞

本論文は，著者が三重大学大学院にて行った研究をまとめたものである．本論文を進めるにあたり，懇切丁寧なご指導と御督励を賜った三重大学大学院の高瀬教授，鶴岡教授，川中准教授，北准教授，および東海大学の森田准教授に感謝いたします．また，日頃熱心に討論していただいた計算機研究室の皆様型に厚く御礼申し上げます．

最後に，本論文をまとめるにあたり，助言，討論，その他お世話になったすべての方々に感謝いたします．

## 参考文献

- [1] Cortes, Corinna, and Vladimir Vapnik, “Support-vector networks.” *Machine learning*, Vol.20, No.3 ,pp273–297, 1995.
- [2] Drucker, Harris and Burges, Christopher JC and Kaufman, Linda and Smola, Alex J and Vapnik, Vladimir, “Support vector regression machines.’’ *Advances in neural information processing systems*, pp.155–161, 1997.
- [3] 波部齊, “ランダムフォレスト.” *研究報告コンピュータビジョンとイメージメディア (CVIM)*, Vol.2012-CVIM-182, No.31, pp.1–8, 2012.
- [4] 波部齊, “ランダムフォレストの基礎と最近の動向.” *映像情報メディア学会誌*, Vo.70, No.9, pp.788–791. 2016.
- [5] 道端正裕, 三好力, 増山博, “自己組織化マップの自動追加学習システム (SOM とその応用).” *日本ファジィ学会誌*, Vol.14, No.2, pp147–156, 2002.
- [6] 畝見達夫, “強化学習 (i 小特集j 「最近の機械学習」 )” *人工知能学会誌*, Vol.9, No.6, pp830–836, 1994.
- [7] LeCun, Yann and Bengio, Yoshua and Hinton, Geoffrey, “ Deep learning.” *nature*, Vol.521, No.7553, pp.436–444, 2015.
- [8] Goodfellow, Ian and Bengio, Yoshua and Courville, Aaron, “Deep learning.” 2016.
- [9] 中山英樹, “深層畳み込みニューラルネットワークによる画像特徴抽出と転移学習.” *信学技報*, Vol.115, No.146, pp.55-59, 2015.
- [10] Yue-Hei Ng, Joe and Hausknecht, Matthew and Vijayanarasimhan, Sudheendra and Vinyals, Oriol and Monga, Rajat and Toderici, George, “Beyond short snippets: Deep networks for video classification.” *Proceedings of the*

- IEEE conference on computer vision and pattern recognition, pp.4694–4702, 2015.
- [11] Karpathy, Andrej and Toderici, George and Shetty, Sanketh and Leung, Thomas and Sukthankar, Rahul and Fei-Fei, Li, “Large-scale video classification with convolutional neural networks.” Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp.1725–1732, 2014.
- [12] Amodei, Dario and Ananthanarayanan, Sundaram and Anubhai, Rishita and Bai, Jingliang and Battenberg, Eric and Case, Carl and Casper, Jared and Catanzaro, Bryan and Cheng, Qiang and Chen, Guoliang and others, “Deep speech 2: End-to-end speech recognition in english and mandarin.” International conference on machine learning, pp.173–182, 2016.
- [13] Hannun, Awni and Case, Carl and Casper, Jared and Catanzaro, Bryan and Diamos, Greg and Elsen, Erich and Prenger, Ryan and Satheesh, Sanjeev and Sengupta, Shubho and Coates, Adam and others, “Deep speech: Scaling up end-to-end speech recognition.” arXiv preprint arXiv:1412.5567, 2014.
- [14] LeCun, Yann and Boser, Bernhard E and Denker, John S and Henderson, Donnie and Howard, Richard E and Hubbard, Wayne E and Jackel, Lawrence D, “Handwritten digit recognition with a back-propagation network.” Advances in neural information processing systems, pp.396–404, 1990.
- [15] Hinton, Geoffrey E and Salakhutdinov, Ruslan R, “Reducing the dimensionality of data with neural networks.” science, Vol.313, No.5786, pp.504–507, 2006.
- [16] Krizhevsky, Alex and Sutskever, Ilya and Hinton, Geoffrey E, “Imagenet classification with deep convolutional neural networks.” Advances in neural information processing system, pp.1097–1105, 2012.
- [17] Simard, Patrice Y and Steinkraus, David and Platt, John C and others, “Best practices for convolutional neural networks applied to visual document analysis.” Icdar, Vol.3, No. 2003, 2003.



- [18] Xu, Li and Ren, Jimmy SJ and Liu, Ce and Jia, Jiaya, “Deep convolutional neural network for image deconvolution.” *Advances in neural information processing systems*, pp.1790–1798, 2014.
- [19] Gamboa, John Cristian Borges, “Deep learning for time-series analysis.” *arXiv preprint arXiv:1701.01887*, 2017.
- [20] Långkvist, Martin and Karlsson, Lars and Loutfi, Amy, “A review of unsupervised feature learning and deep learning for time-series modeling.” *Pattern Recognition Letters*, Vol.42, pp.11-24, 2014.
- [21] Simonyan, Karen and Zisserman, Andrew, “Two-stream convolutional networks for action recognition in videos.” *Advances in neural information processing systems*, pp.568–576, 2014.
- [22] 古樋直己: 偶発的語彙習得と英語力、語の頻度、コンテキストの関係: 英語字幕付き邦画を用いた場合, *映画英語教育研究: 紀要*, Vol.14, pp.29–40, 2009
- [23] S.Sakurai, N.Nishizawa : A NEW APPROACH FOR DISCOVERING TOP-K SEQUENTIAL PATTERNS BASED ON THE VARIETY OF ITEMS, *Journal of Artificial Intelligence and Soft Computing Research*, Vol. 5, No. 2, pp. 141-153, 2015
- [24] Mikolov, Tomáš and Karafiát, Martin and Burget, Lukáš and Černocký, Jan and Khudanpur, Sanjeev, “Recurrent neural network based language model.” *Eleventh annual conference of the international speech communication association*, 2010.
- [25] Dorffner, Georg, “Neural networks for time series processing.” *Neural network world*, 1996.
- [26] McLaughlin, Niall and Martinez del Rincon, Jesus and Miller, Paul, “Recurrent convolutional network for video-based person re-identification.” *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.1325-1334, 2016.
- [27] J L.Elmán : Finding structure in time, *Cognitive Science*, Vol. 14, pp. 179-211, 1990

- [28] Hochreiter, Sepp and Schmidhuber, Jürgen, “Long short-term memory.” *Neural computation*, Vol.9, NO.8, pp1735–1780, 1997.
- [29] Sundermeyer, Martin and Schlüter, Ralf and Ney, Hermann, “LSTM neural networks for language modeling.” *Thirteenth annual conference of the international speech communication association*, 2012.
- [30] Tai, Kai Sheng and Socher, Richard and Manning, Christopher D, “Improved semantic representations from tree-structured long short-term memory networks.” *arXiv preprint arXiv:1503.00075*, 2015.
- [31] 田中一穂, 矢野慎一郎, 山本野人: 連続した入力パタンのあいだの順序関係を認識する神経回路モデル—情報の予測・抽象化に向けて, *日本応用数理学会論文誌*, Vol.18, No.1, pp.87–105, 2008
- [32] T. Aoki and T. Aoyagi: A Possible Role of Incoming Spike Synchrony in Associative Memory Model with STDP Learning Rule, *Progress of Theoretical Physics Supplement*, No.161, pp.152–155, 2006
- [33] K.Morita, H.Takase, N.Morita, H.Kawanaka, S.Tsuruoka : Stable Extraction of Frequent Sub-sequences from Sequential Symbol Input, *Procedia Computer Science*, Vol. 112, pp 1901–1908, 2017
- [34] Hebb, Donald Olding, “The organization of behavior: A neuropsychological theory.” *The organization of behavior: A neuropsychological theory*.
- [35] A. N. Burkitt, “A review of the integrate-and-fire neuron model: I Homogeneous synaptic input.” *Biol Cybern.* 95, pp.119, 2006.
- [36] Liu, Ying-Hui and Wang, Xiao-Jing, “Spike-frequency adaptation of a generalized leaky integrate-and-fire model neuron.” *Journal of computational neuroscience*, Vo.10, No.1, pp25–45, 2001.
- [37] Markram, Henry and Gerstner, Wulfram and Sjöström, Per Jesper, “Spike-timing-dependent plasticity: a comprehensive overview.” *Frontiers in Synaptic Neuroscience*, Vol.4, No.2, 2012.

- [38] S. Song, K. D. Miller and L. F. Abbott: Competitive Hebbian learning through spike-timing-dependent synaptic plasticity, *Nature Neuroscience*, Vol. 3, No. 9, pp. 919–926, 2000.
- [39] K. Morita, H. Takasea, M.Sawamura, N.Morita, H. Kita, “Extraction of Frequent Sequential Patterns from Sequence at Uneven Intervals.” *Proceedings of SCIS&ISIS2018*, pp.1449–1453, 2018.
- [40] 森田賢太, 高瀬治彦, 川中普晴, 森田直樹: 順序列から任意の出現頻度以上の部分列を抽出するニューラルネットワーク, *知能と情報*, Vol.31, No.1, pp.592–596, 2019.
- [41] M. Kohno, “Mora, Syllable, and Rhythm - A Psycholinguistic Study.” *Journal of the Phonetic Society of Japan*, Vol.2 No.1 , pp.16–24, 1998.