

修士論文

リカレントニューラルネットワークにおける
時間連続状態機械の実現

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

津田 晃寿

平成15年2月7日

リカレントニューラルネットワークにおける 時間連続状態機械の実現

津田 晃寿

内容梗概

リカレント・ニューラル・ネットワーク (RNN) は フィードバック・ループを持つニューラル・ネットワークである。このフィードバック・ループによって、RNN は時系列情報を処理することができる。しかも連続値を扱えることから、RNN は連続状態機械と呼ばれる能力の高い機械クラスを提供する。連続状態機械とは、連続な状態空間において離散的な状態遷移を行うものである。

しかし、RNN は時間を連続的に扱える。そこで我々は、時間連続状態機械という新しい機械クラスを提唱した。時間連続状態機械は、時間連続な状態空間とその上の状態遷移によって定義される。これをエポック毎通時的誤差逆伝搬法と教師強制を用いて、RNN に学習させる方法を提案する。

時間連続な入出力系列を学習する際の問題は2つある。1つ目は、時間連続に変化する入力に対し、教師信号をいかに定義するかである。2つ目は、時間局所的に同一な入力に対し異なる教師信号をいかに学習するかである。

1つ目の問題に対し我々は、学習中に教師信号をその時のRNNの出力に応じて適応的に変化させることにする。出力を教師信号とすることで、入力が連続的に変化している区間における誤差を0とし、積極的に出力を変化させないようにする。2つ目の問題に対し我々は、一定時刻前の教師信号を入力とすることにする。時間局所的に入力が同一で教師信号が異なる区間においても、異なる入力が入ることで、識別しやすいようにする。

本稿では、時間連続状態機械として、時間連続な入力のポジティブエッジトリガの1bit-モジュロ・カウンタをRNNに学習させた。まず、入力の变化時間一定の7パターンをトレーニング・セットとして、通常のRNNと一定時刻前の教師信号を入力に追加したRNNの学習の収束率を比較した。その結果、ニューロン数16個において、前者が全く収束しなかったのに対し、後者では55%収束した。続いて、入力の变化時間が異なるパターンもトレーニング・セットに加えて、本手法を用いて学習したところ、学習に用いた変化時間内の任意のポジティブエッジに対し1bit-モジュロ・カウンタが実現できた。

Implementation of Temporally Graded State Machine based on Recurrent Neural Network

Akihisa Tsuda

Abstract

A Recurrent Neural Network is a Neural Network with feedback loops. These feedback loops enable RNN to handle time series information. And RNN can deal with real numbers, and is classified as powerful machine: Graded State Machine. GSM has continuous states and discrete transitions.

But RNN can deal with time continuously. We supposed a new machine class : Temporally Graded State Machine. TGSM has continuous states and continuous transition. We supposed the method to make RNN learn TGSM with Epochwise Back Propagation Through Time and Teacher Forcing.

RNN has two problems to learn continuous input and output sequences. The first is how we define teacher signals for inputs varying temporally. The second is how RNN learns different teacher sequences for same input sequences in local time.

For the first, we suppose that teacher signals are adaptive for outputs of RNN on each time in learning. Mean Square Error is zero in the terms in which inputs vary temporally by which outputs substitute for teacher signals in those. Therefore, outputs don't vary positively in those. For the second, we suppose RNN of which inputs are added teacher signals before some times. It helps RNN for the discrimination in the terms in which input sequences are same and teacher sequences are different.

In this paper, RNN learned a positive edge trigger 1bit modulo counter with continuous input sequences. We compared the convergency rate of learning in the normal RNN with that in the RNN of which inputs are added teacher signals before some times, in the 7 training sets whose inputs vary temporally constantly. Then, the former was 0 % and the latter was 55 % in 16 neurons. Next, RNN learned the training sets added patterns whose input vary temporally not-constantly, with above method. Then, RNN became 1bit modulo counter in arbitrary input patterns of above varying time.

リカレントニューラルネットワークにおける 時間連続状態機械の実現

目次

第1章	はじめに	1
第2章	背景	4
2.1	連続状態機械とリカレントネット	4
2.1.1	連続状態機械	5
2.1.2	リカレントネット	6
2.1.3	関連研究	11
2.2	リカレントネットの学習	17
2.2.1	BPTT	18
2.2.2	EBPTT	20
2.2.3	RTBPTT	22
2.2.4	教師強制	23
第3章	時間連続状態機械とその学習方法	27
3.1	時間連続状態機械	27
3.2	学習における諸問題と解決法	29
3.2.1	教師信号	30
3.2.2	採用するリカレントネットのモデル	32
3.2.3	学習方法	34
第4章	リカレントネットにおける時間連続状態機械の実現	39
4.1	遅延の効果	39
4.2	時間連続状態機械の実現	44
第5章	おわりに	49
	謝辞	51
	参考文献	52

第1章 はじめに

ニューラル・ネットワーク (Neural Network、以下ニューラルネットとする)・モデルは、知的な情報処理に対する従来の計算的なアプローチでは、思考や言語といった人間の持つ高次の情報処理能力のいくつかの重要な側面をうまく説明することができないだろうという感觸から生まれてきた。

それにも関わらず通常のニューラルネットは、高次の情報処理に必要ないくつかの機能をうまく実現することができない。それらの中には、可変長データの取り扱いがある。3層モデルに代表される通常のニューラルネットは、固定長のパターン処理を基本としており、可変長のデータをうまく扱うことができない。

一方、記号処理に代表される従来の計算的なアプローチでは、LISPのリストなどの例を持ち出すまでもなく、可変長のデータを容易に取り扱うことができる。特に記号処理は、人工知能の基本技術として用いられてきたが、可変長データの取扱いが容易であったことがその理由の1つとして挙げられよう。

しかし、記号処理などでも、可変長データがそのままの形で処理されているわけではない。それらはオートマトンにその基礎を置いているが、そのオートマトンも可変長のデータをそのまま扱うことはできない。オートマトンの入力には有限個のアルファベットであり、やはり固定長である。オートマトンが可変長のデータを扱うときには、まずそれを固定長のデータの時系列に分解する必要がある。オートマトンは、固定長データの時系列に対する処理として、可変長データに対する処理を実現しているのである。

リカレント・ニューラル・ネットワーク(Recurrent Neural Network, 以下リカレントネットとする)は、フィードバック・ループを持つニューラルネットのことである。リカレントネットのフィードバック・ループは、状態機械におけるフィードバック・ループと同様に、入力の履歴をその状態として代表させることができる。したがって、リカレントネットは通常の組み合わせ回路的なニューラルネットとは異なり、オートマトンと同様の時系列処理を行うことができる [1]。

思考や言語といった人間の高次情報処理も時系列処理であると考えられる。したがって、リカレントネットの時系列処理は、こういった人間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。

また、このような人間の持つ高次情報処理を時系列処理としてとらえる見方は、リカレントネット と オートマトン の対比という重要な視点を提供する。オートマトンは生成文法理論と密接な関係を持っているため、リカレントネット を オートマトン としてとらえることによって、生成文法との関係を議論することができるようになる。

実際、McClelland ら ([2] [3]) は、連続状態機械 (Graded State Machine) というクラスを提案している。連続状態機械 とは、連続的な状態空間とその上の離散的な状態遷移によって定義される オートマトン である。単純回帰結合回路網 (Simple Recurrent Network) は、リカレントネット の最も簡単な例の1つであるが、連続状態機械 であることが分かっている。

連続状態機械 は、その連続的な性質によって、有限状態機械 より高い能力を持つ。そのため 連続状態機械 は、自然言語の処理に代表される人間の高次情報処理に十分な能力を持つ可能性があると考えられている。

しかし一般には、処理させたい入力時間が時間方向に離散的な時系列であることは稀で、例えば、音声認識 や 株価予測 などのような、時間方向に連続的に変化しながら連続的な値をとる時系列であることが多い。そして、リカレントネット は時間方向に連続な入力を処理することができる。

それにも関わらず、従来の リカレントネット の研究では、処理する入力時間が時間方向に離散的な時系列であることが多く、時間方向に連続な場合はほとんど研究されてこなかった。そこで我々は、時間方向にも連続的な状態空間と、その上の、時間方向にも連続的な状態遷移を持つ 連続状態機械 を 時間連続状態機械 (Temporally Graded State Machine) と呼ぶことを提唱し、これをリカレントネット に学習させる方法を提案する。

もちろん、計算機は連続値を扱うことができない。よって計算機上でモデルを扱う以上、時間の最小単位時間を定める必要があり、厳密には時間方向に離散化されることになる。

しかし、例えば有限個のアルファベットで構成される離散列を処理するのは本質的に異なる。ある瞬間に入力がとり得る値の種類は、計算機の精度に制限を受けるものの無限である。また極小時間で入力の変化を見た場合、それは入力がとり得る値の全域に比べると僅かな変位に収まる。離散列のように、大きく異なる値に瞬間的に変化することはない。

以下2章では、連続状態機械 と リカレントネット の関係について述べる。続

いて、リカレントネットの一般的な構成法である学習アルゴリズムについて述べる。次に3章で、時間連続状態機械と、それを実現するための学習方法について述べる。我々は、一般的なリカレントネットと同様に、学習によりリカレントネットを構成するが、一般的なリカレントネットと異なり、入力時間が時間連続であることから、当然その学習法も異なる。そこで、リカレントネットに時間連続状態機械を学習させる際の諸問題と解決法について述べる。4章では、その解決法の効果、および、実際に時間連続状態機械の簡単な例として、時間連続な入力に対してポジティブエッジトリガの1bit-モジュロ・カウンタとして動作するリカレントネットを構成したので、その結果を述べる。最後に、まとめを行う。

第2章 背景

フィードバック・ループを持ったニューラルネットであるリカレントネットは、時系列処理を行えるため、状態機械との対比をすることができる。実際、McClellandらによる連続状態機械の提案も、状態機械との対比という視点から生まれたものである。

そこで以下では、まず次節で連続状態機械とリカレントネットの関係について述べる。連続状態機械について、有限状態機械との比較を交えて説明し、またその能力についても述べる。続いて、リカレントネットについても概説し、それが連続状態機械とどのような関係にあるか説明する。さらに、リカレントネットの能力や、具体的な応用例についても述べる。

続いて2.2節において、リカレントネットの学習アルゴリズムについて述べる。リカレントネットの学習アルゴリズムは階層型ニューラルネットと同様に最急降下法に基づくものが一般に用いられる。そこで、その代表的なものについて述べる。

2.1 連続状態機械とリカレントネット

連続状態機械とは、連続値の状態情報を利用できる状態機械である。この連続値を扱える、という点において、連続状態機械は有限状態機械よりも高い能力を持つ。

また、連続状態機械は一般にリカレントネットによって実現される。これは、リカレントネットが先に述べたように時系列処理を行うことができ、かつ連続値を扱うことができることによる。

そこで以下、まず2.1.1節で連続状態機械について有限状態機械との比較を交えて説明し、またその能力についても述べる。続いて2.1.2節で、連続状態機械を実現する際に一般に用いられるリカレントネットについて概説する。リカレントネットの構成要素はニューロンである。このニューロンのモデルについてまず説明する。そして、そのニューロン・モデルがどのようにリカレントネットを構成しているかについて説明する。また、リカレントネットと有限状態機械の動作を対比する。リカレントネットは有限状態機械を模倣でき、さらに連続値を扱えることから、連続状態機械が構成できることを説明する。そして2.1.3節において、連続状態機械の従来の研究について述べる。まず、

連続状態機械を構成するリカレントネットの能力について述べる。そして、その具体的な応用例として McClelland らによって行なわれた研究について述べる。

2.1.1 連続状態機械

連続状態機械とは、第1章で述べたように、連続的な状態空間とその上の離散的な状態遷移によって定義されるオートマトンである。そこで以下では、これを具体的に説明するために、まずその入出力や状態遷移の様子を、有限状態機械と比較する。続いて連続状態機械の能力について述べる。

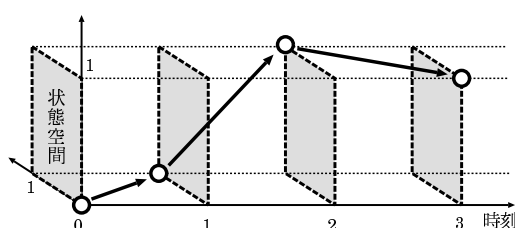


図1: 有限状態機械における状態遷移

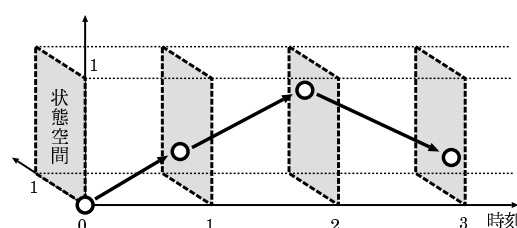


図2: 連続状態機械における状態遷移

有限状態機械 有限状態機械における入出力は、共に0,1の離散値である。そして、その状態遷移は、例えば図2.1.1のようになる。ここで、横軸は時刻、奥向きの軸と縦軸によって2次元の状態空間を表している。

有限状態機械では、状態の取り得る値は0,1の離散値である。そのため、状態空間においてとる状態は、座標を(奥向きの軸の値, 縦軸の値)として表現すると、(0,0),(0,1),(1,0),(1,1)の4状態である。

そして、時間も離散的に扱われているため、時刻が0,1,...と進む毎に、先の4つの状態のうちの1つに遷移することになる。図の例では、白抜きの丸が各時刻の状態の位置とすると、時刻0で(0,0)、時刻1で(0,1)、時刻2で(1,1)、時刻3で(1,0)のように遷移している。

連続状態機械 これに対し、連続状態機械においては、入出力は有限状態機械と同様に0,1の離散値であるが、状態は連続値である。そしてその状態遷移は、例えば図2.1.1のようになる。ここで、各軸は図2.1.1の場合と同じである。

連続状態機械では、状態の取り得る値は[0,1]の範囲の連続値であるので、状態空間においてとる状態は、先と同様に座標を(奥向きの軸の値, 縦軸の値)として表現すると、(0,0)から(1,1)までの非可算無限個の状態が存在する。

ただし、連続状態機械における時間は有限状態機械の時と同様、離散的に扱われているため、時刻が $0,1,\dots$ と進む毎に、連続状態機械のとり状態は先の非可算無限個のうちの1つに遷移することになる。

以上をまとめると、次のようになる。

1. 有限状態機械
入出力・状態： 離散値
時間： 離散
2. 連続状態機械
入出力： 離散値
状態： 連続値(実数)
時間： 離散

連続状態機械の能力 連続状態機械は連続的な状態空間として、有理数だけでなく無理数も含む実数を扱える。一方、チューリング機械は有理数までしか扱うことができない。したがって、連続状態機械はチューリング機械よりも能力が高く、任意の言語が計算可能である([4] [5] [6])。このような連続状態機械の機械クラスにおける位置付けは図3のようになる。

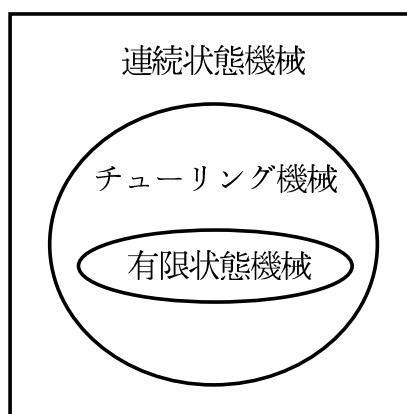


図3: 連続状態機械の機械クラスにおける位置付け

2.1.2 リカレントネット

前節のような連続状態機械は、一般にリカレントネットによって構成される。そこで以下では、まずはリカレントネットの構成要素であるニューロン・モデルについて説明する。リカレントネットで用いられるニューロン・モデル

ルも、有名な階層型ニューラルネットにおいて誤差逆伝搬法を用いる場合と同様に、準線形素子モデルと呼ばれるものである ([7] [8])。続いて、その準線形素子モデルを用いて構成されるリカレントネットについて説明する。リカレントネットはフィードバック・ループを有することにより順序回路と同様に時系列処理が行える。そして、さらに連続値を扱えることから、リカレントネットは連続状態機械を構成できることについて述べる。

ニューロン・モデル リカレントネットにおいて一般によく用いられるニューロン・モデルが、準線形素子モデル (図4) である。

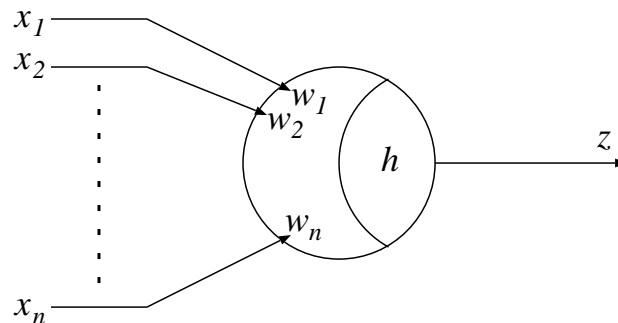


図4: ニューロン・モデル

このモデルでは一つのニューロンは n 個のニューロンからそれぞれ x_1, x_2, \dots, x_n の強さの入力信号を受けとり、その内部状態 u および出力 z は

$$u = \sum_{i=1}^n w_i x_i \quad (1)$$

$$z = f[u - h] \quad (2)$$

で表される。

ここで h は閾値であり、 w_i は i 番目のニューロンとの間の結合効率を表す量で結合荷重と呼ばれる。

ニューロンの評価関数 f は、一般に微分可能な単調増加関数および単調非減少関数が用いられる。これは、ニューラルネットに学習させる際に、誤差逆伝搬法のような最急降下法に基づく学習アルゴリズムを用いる場合、結合荷重の変量を求める際に微分されるからである。一般には、

$$f[u - h] = \frac{1}{1 + \exp(-(u - h))} \quad (3)$$

のようなシグモイド関数 が用いられ、ニューロンの入出力は(0,1)の範囲の連続値をとる。

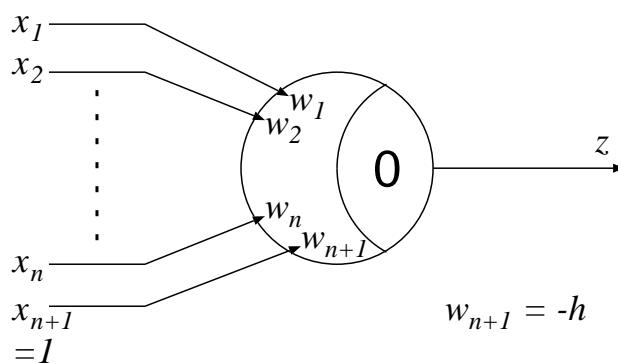


図5: 閾値を結合荷重に含めたニューロン・モデル

ところで、数式のと扱ひの簡単化のために、閾値 h の代わりに、仮想的に常に 1 を与える $n+1$ 番目の入力 x_{n+1} を設けて、新たに $n+1$ 番目の結合荷重 w_{n+1} を $-h$ と定義すれば、式(1)と(2)で定義される入出力関係は、次式のように簡潔に表される。

$$u - h = \sum_{i=1}^n w_i x_i + w_{n+1} = \sum_{i=1}^{n+1} w_i x_i, \quad w_{n+1} = -h \quad (4)$$

$$z = f \left(\sum_{i=1}^{n+1} w_i x_i \right) \quad (5)$$

この表現式に対応するニューロン・モデルは、図5のようになる。この w_{n+1} はバイアス(bias)と呼ばれる。また、それまで閾値としていたところは0となっている。以上のようにすることで、閾値の学習も、結合荷重の場合と同様に行うことができる。以降、この表現式で表されるニューロン・モデルを用いる。

リカレントネット リカレントネットとは、フィードバック・ループを持つニューラルネットのことである。例えば図6は、ニューロン数3個のリカレントネットを表している。図6の四角は外部入力を、丸と二重丸はニューロンを表している。3個のニューロンの内の1つ(図6の二重丸)が外部出力となっている。

リカレントネットのフィードバック・ループは、順序回路のフィードバック・ループと同様に、入力の履歴をその状態として代表させることができる。

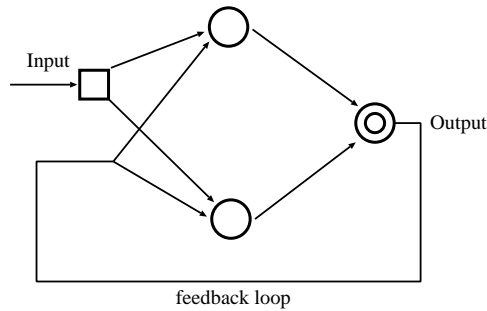


図6: リカレントネットの例

これに対し、通常のニューラルネットは、出力がその時の入力から一意に決まることから、組み合わせ回路的に動作する、と言える。したがってリカレントネットは、通常の組み合わせ回路的なニューラルネットとは異なり、順序回路と同様の時系列処理を行うことができる。

具体的にそのことを示すために、以下リカレントネットが順序回路を模倣できる例について述べる。例として、入力系列のパリティ(XOR)を求める順序回路(図7)を考える。

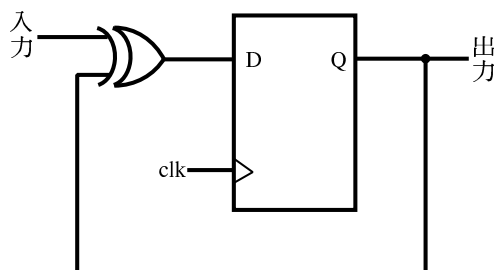


図7: 入力系列のパリティを求める順序回路

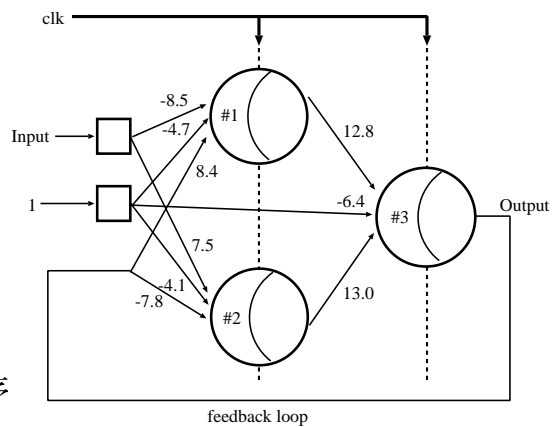


図8: 入力系列のパリティを求めるリカレントネット

これに対し、図8のような結合荷重を持つリカレントネットならば、図7と同様に入力系列のパリティ(XOR)を求める処理を行える。ここで、図8の四角は外部入力を表している。2つの外部入力のうちの1つは常に1が与えられる、バイアス用の入力である。

また、図8の丸はニューロンを表している。#1~#3はニューロンのインデックス

クスである。ニューロンの評価関数 f はここではシグモイド関数とし、前述の式(3)で与えられると仮定する。

3つのニューロンのうちの1つ(#3)は外部出力となっている。その外部出力からは、フィードバック・ループが#1と#2のニューロンそれぞれの入力となっている。

また、外部入力とニューロン、および、ニューロン同士を結合する各矢印は、向きは信号が伝搬する向きを、そこに付された数値は結合荷重を表している。ただし、常に1が与えられる外部入力から各ニューロンへの矢印に付された数値はバイアスである。

具体的に図7の順序回路と図8のリカレントネットの動作を比較してみる。入力の系列の例として、1が最初に入り、0がその次に来るとする。

図7において、まずXORの入力が外部入力側の1とフィードバック・ループ側の0なので、XORの出力は1となる。そこでクロック(図のclk)が入ることによってDフリップ・フロップにこの1が蓄えられる。Dフリップ・フロップの出力1がフィードバック・ループを伝わると、次のXORの外部入力として0が入ってきているので、XORの出力は1となる。その後、またクロックが入ることによってDフリップ・フロップにこの1が蓄えられる。

一方、図8において、まず外部入力(図のInput)の1とバイアス用入力の1、そしてフィードバック・ループからの入力の初期値である0が、それぞれ最初のニューロン2つに伝搬し、先に述べたニューロンの評価関数の式(3)に基づいて出力が計算される。このように矢印方向への信号の伝搬に基いた各ニューロンの出力の計算を行うことを、一般に**Forward Propagation**と呼ぶ(以降、FPとする)。ここでは#1のニューロンの内部状態 u は

$$\begin{aligned} u &= \sum_{i=1}^{n+1} w_i x_i \\ &= -8.5 \times 1 + 8.4 \times 0 + -4.7 \times 1 \\ &= -13.2 \end{aligned}$$

であるから、#1の出力 z は

$$\begin{aligned} z &= f[u] \\ &= \frac{1}{1 + \exp(-(u))} \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{1 + \exp(-(-13.2))} \\
&= 1.9 \times 10^{-6}
\end{aligned}$$

という 0 に近い値となる。同様に、#2 の出力は 0.97 という 1 に近い値となる。これらとバイアス用の入力 1 が次のニューロン #3 に伝搬して、同様に出力が計算され、0.99 という 1 に近い値が出る。このおよそ 1 という出力値がフィードバック・ループを伝搬し、次の外部入力 0 とあわせて出力の計算が同様に行われ、出力として 0.99 という 1 に近い値が出る。

この時、図 8 の clk が指す同じ点線の上にあるニューロンを、リカレントネットにおける同一の段とみなし、それぞれの段において同期がとられる。以下、この一つの段において FP が 1 回行われる時間を 1 時刻とする。

例えば、#1 と #2 を同一の段として計算の同期がとられる。つまり、同一時刻において #1 と #2 の FP がなされる。そして次の時刻にその結果が #3 のニューロンに伝搬され、その位置を一つの段として同期がとられ、計算がなされる。さらに次の時刻において、#3 からフィードバックされた値と次の外部入力との計算を、再び #1 と #2 を同一の段として行うための同期がとられる。

したがって、図 7 の順序回路の clk と同様に、図 8 の clk のような仮想的なクロックが存在する。

以上のように、リカレントネットは、順序回路と同様に時系列処理が行える。さらに、図 8 のようなリカレントネットは、2.1.1 節で述べたように入出力および状態に連続値を扱えることから、連続状態機械を実現することができる、と言える。

2.1.3 関連研究

2.1.1 節において述べたように、連続状態機械はチューリング機械よりも能力が高く万能である。そして、このような連続状態機械は一般にリカレントネットによって構成されることも述べた。したがって、連続状態機械とリカレントネットは同じ能力を持つことが考えられる。そして、実際にそれを証明するリカレントネットの能力についての研究が Hava らによってなされている。

そこで以下では、まず次節で Hava らの研究を紹介する。Hava らは、リカレントネットの結合荷重の型が、整数、有理数、実数の場合に、それぞれに対

応する言語のクラスを求める研究を行った ([9] [10])。その結果について概説する。

続いて、このような高い能力を持つリカレントネットを応用する研究の例として、McClelland らの研究を紹介する。

リカレントネットの能力 リカレントネットの計算理論として、チューリング機械との計算能力の対応が多く研究されている。ここでは、離散時間、連続状態のリカレントネット・モデルについて、その結合荷重に制約を与えた場合のリカレントネットの計算能力に関する Hava らの結果を紹介する。

ここで扱うリカレントネットの定義は、

$$x_i(t+1) = \sigma \left(\sum_{j=1}^N a_{ij}x_j(t) + \sum_{j=1}^M b_{ij}u_j(t) + c_i \right), \quad i = 1, \dots, N$$

$$\sigma(x) = \begin{cases} 0 & x < 0 \\ x & 0 \leq x \leq 1 \\ 1 & x > 1 \end{cases}$$

である。

ここで、関数 σ はニューロンの評価関数であり、最も単純に実現できるシグモイド関数である飽和線形関数を用いている。 i 番目のニューロンの $x_i(t)$ は時刻 t での平均発火頻度、 N はニューロンの数、 $u_j(t)$ は外部入力、 M は外部入力の個数、 a_{ij}, b_{ij} は結合荷重、 c_i は閾値を表す。

このリカレントネットについて、言語 $L \subseteq \{0,1\}^+$ の決定問題をネットワーク上で定義し計算論的視点を導入する。言語 $L \subseteq \{0,1\}^+$ の決定問題とは、入力される文字列 $\omega \in \{0,1\}^+$ が言語 L に含まれるか否かを決定する問題である。

ネットワークへの入力を $u(t) \equiv (D(t), V(t)) \in \{0,1\} \times \{0,1\}$ とし、ネットワーク中の2つのニューロン $x_{o1}(t), x_{o2}(t)$ を用いて、その出力を $(O_d(t), O_v(t)) \equiv (x_{o1}(t), x_{o2}(t))$ とする。ここで、 $D(t)$ は入力データを表し、 $D(t)$ にデータが入力されていることを $V(t) = 1$ で表す。出力についても同様に $O_v(t) = 1$ で、 $O_d(t)$ にデータが出力されていることを表す。

この入力 $u(t)$ を用いて文字列 $\omega = \alpha_1\alpha_2 \cdots \alpha_n$ を

$$D(t) = \begin{cases} \alpha_t & t = 1, 2, \dots, n \text{ のとき} \\ 0 & \text{その他} \end{cases}$$

$$V(t) = \begin{cases} 1 & t = 1, 2, \dots, n \text{ のとき} \\ 0 & \text{その他} \end{cases}$$

としてネットワークに入力する。ネットワークにおける計算は、各ニューロンの初期値を $x_i(1) = 0$ として行う。

ここで、閾値 c_i は $\sigma(c_i) = 0$ となるように選ぶ。計算結果 ($\omega \in L$ か否か) は、時間 τ で計算が終了するとき、

$$O_d(t) = \begin{cases} \eta_\omega & t = \tau \text{ のとき} \\ 0 & \text{その他} \end{cases}$$

$$O_v(t) = \begin{cases} 1 & t = \tau \text{ のとき} \\ 0 & \text{その他} \end{cases}$$

として出力する。ここで、 η_ω は 1 または 0 であり、受理・非受理を表す。

この入出力が定義されたネットワークをここでは、形式ネットワークと呼ぶ。

$T: \mathbf{N} \rightarrow \mathbf{N}$ を自然数上の関数とする。言語 $L \subseteq \{0, 1\}^+$ について、形式ネットワーク N が存在して、任意の $\omega \in \{0, 1\}^+$ について、時間 $\tau \leq T(|\omega|)$ で計算が終了し、かつ $\omega \in L$ のとき $\eta_\omega = 1$, そうでないとき $\eta_\omega = 0$ であるとき、言語 L は、形式ネットワーク N によって時間 T で認識されると言う。

さて、このように定義した形式ネットワークについて、次のことが示されている。

- (1) 整数係数の形式ネットワーク N_Z (結合荷重が整数) において各ニューロンは、 $\{0, 1\}$ の 2 値をとり、 N_Z が計算可能な言語のクラスは 正規言語 に対応する。
- (2) 有理数係数の形式ネットワーク N_Q の計算能力はチューリング機械 に等しい。これは、計算時間についても同様に等しく、チューリング機械 M で時間 T で計算される関数を時間 T で計算する N_Q を構成でき、 N_Q のサイズは時間 T に依存しない。そして、万能チューリング機械 に対応する有限個のニューロンからなる有理数係数の形式ネットワークが存在する。
- (3) 実数係数の形式ネットワーク N_R は実数を無限の精度で指定できるという意味で チューリング機械 を超える計算能力を持つ。このとき計算可能な言語のクラスは、計算時間を制限しなければ任意の言語が計算可能である。

一方で、実験的には、リカレントネットは状態機械より能力が高く、文脈自由文法に及ばないまでも、それに近い言語クラスを処理することができることが分かっている。そこで以降では、そのようなリカレントネットの応用例として、McClellandらの研究を紹介する。

応用例 以下では、まずリカレントネットの応用研究において従来よく用いられてきた単純回帰結合回路網について述べる。続いて、その応用例として、McClellandらの研究について説明する。

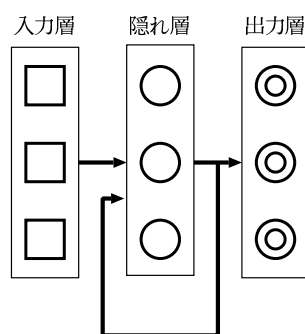


図9: SRN

連続状態機械をリカレントネットで実現する従来の研究では、Elman [11] によって提案された、図9のような単純回帰結合回路網 (Simple Recurrent Network, 以下シンプルリカレントネットとする) がリカレントネットのモデルとして一般に用いられる。

シンプルリカレントネットは、入力層、隠れ層、出力層による一般的な3層モデルに、隠れ層から隠れ層へのフィードバック・ループを加えたものである。一般的な3層モデルと同様に、隣接する層間のニューロンは結合しているが、層内の結合は存在しない。図9の矢印は、その結合と、信号の伝搬する方向を示している。

McClellandらは、Reber文法(図10)と呼ばれる単純な有限状態機械により生成されたアルファベット列を受理する状態機械をシンプルリカレントネットに学習させる研究を行った[3]。ここで、このReber文法を「受理する」とは、与えられたアルファベット列のすべての文字が、文法の最後の文字Eにいたるまで文法にかなっていることを言う。

以下では、最初に、シンプルリカレントネットが動作と状態表現の両方に

において有限状態機械をうまく模倣するよう学習できることを示す。次に、シンプルリカレントネットならば、有限状態機械にはない性質であり、かつ連続状態機械にはある性質として、アルファベット列の前の部分の系列を後で必要になるまで保持できることを示す。

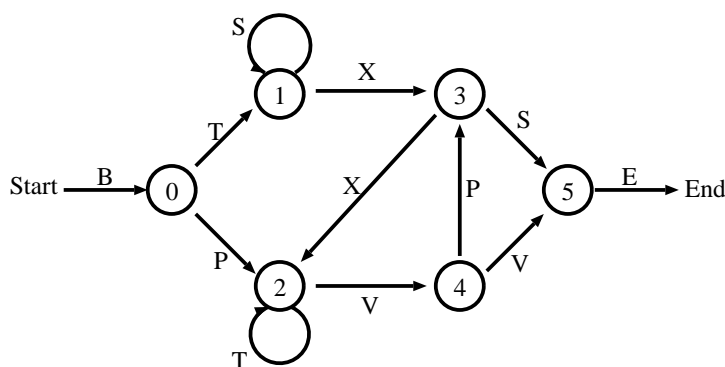


図10: Reber 文法

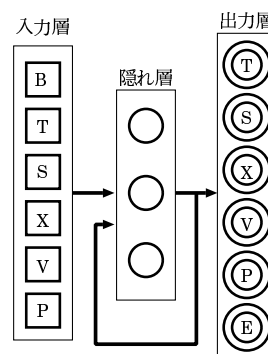


図11: シンプルリカレントネットの構成

実験に用いたシンプルリカレントネットは図11のような構成をしている。入力層には6種類の文字(B,T,S,X,V,P)それぞれに対応するニューロンを6個、出力層には6種類の文字(T,S,X,V,P,E)それぞれに対応するニューロンを6個、そして隠れ層にはニューロンを3個用いた。

3~30文字からなる、Bから始まり図10の文法に従う、ランダムに生成された60,000パターンの文字列を入力とし、BPによる学習を行った。文法に従う時、次の入力として取り得る文字を教師信号として用いる。例えば、現在Bという入力を与えられているならば、出力としてTとPが発火するように学習させる。

この学習後のネットワークに、図10の文法に従う、ランダムに生成された20,000パターンの文字列を入力として加えた。各文字列の最後の文字が入力された結果、Eに対応するニューロンが発火すれば、その系列は受理されたものとみなすとする、これら全ての系列を受理することが確認された。

このように、有限状態機械を非常によく模倣できるようシンプルリカレントネットを訓練することができた。この時、シンプルリカレントネットの隠れ層のパターンとしてReber文法の各状態に相当するクラスターが形成されていることが分かった。

しかし、この Reber 文法 を獲得した シンプルリカレント ネット の内部表現 を子細に見てみると、有限状態機械 とは異なっていることが分かる。実際、この シンプルリカレント ネット を含め、リカレント ネット は連続値を用いているため、本質的に 有限状態機械 とは異なる。このことから、このような連続値の状態情報を利用することができる 状態機械 を 連続状態機械 と呼ぶことを McClelland らは提唱した。

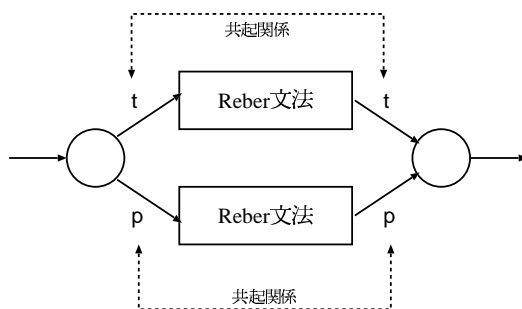


図 12: 埋め込み構造を持つ文法の例

さらに、彼らは、図 12 に示すような同一の部分が複数埋め込まれた構造を持つ文法をシンプルリカレント ネット に学習させる実験を行った [12]。実験の結果、シンプルリカレント ネット は、埋め込み部分が長いなどのいくつかの場合を除いて、ほぼ正しい予測を行うように訓練させることができた。

その際 シンプルリカレント ネット の内部では、その同一部分内の同じ位置にある状態を、全く別の状態として表現しているわけではない。それらの状態は、ほぼ同じ内部表現をとる。つまり最初のアルファベットの違いを、状態表現の僅かな差異として保持している。そうすることで、長時間にわたる入出力の依存関係を処理することができているのである。

このような通常長時間にわたる入出力の依存関係を処理するためには、有限状態機械 ならば図 12 のような埋め込まれた同一の部分に関して、該当する部分の状態のコピーを持たざるを得ない。もっと効率良く扱うのであれば、プッシュ・ダウン・オートマトン を用いることで、埋め込み部分はサブルーチン呼出しのように処理することができる。これに対し、シンプルリカレント ネット はコピーを持つことなく、サブルーチン呼出しを行うこともなく、図 12 のような埋め込み文を学習することができている。

以上のように、リカレントネットの最も簡単な例の一つであるシンプルリカレントネットは有限状態機械よりも能力が高く、文脈自由文法には及ばないまでも、それに近いクラスを処理することができることがMcClellandらによって実験的に示された。

2.2 リカレントネットの学習

リカレントネットの学習アルゴリズムには様々なものが存在するが、一般には、通常のニューラルネットにおける一般的な学習アルゴリズムである誤差逆伝播法(**Back Propagation**, 以下**BP**とする)と同様、最急降下法に基づいた学習アルゴリズムが用いられる。

最急降下法に基づく学習アルゴリズム BP は局所解に収束することは保証されているが、必ずしも最適解に収束する訳ではない。したがって、理論的には階層型ニューラルネットは3層あれば任意の非線型関数を実現できることが知られているが([13] [14])、BPのためには、層を増やした方が最適解に収束し易い場合もあると言われている。しかし、あるパターンを学習させる場合、層を何層にし、層のニューロン数を何個にすれば最適解に収束し易くなるかを知ることには一般に困難である。

リカレントネットにおける最急降下法に基づいた学習アルゴリズムには、様々なものが存在する。ニューラルネットにおけるBP同様、リカレントネットにおいて最急降下法に基づいたこれらの学習アルゴリズムを用いる場合にも、同様の問題がある。この問題については、3.2.2 節で述べる。

リカレントネットの学習アルゴリズムは、厳密に最急降下法に基づいたものから近似したものまで様々なものが存在する。厳密に最急降下法に基づくアルゴリズムには、代表的なものとして実時間リカレント学習法 (Real Time Recurrent Learning) があるが、その計算量は $O(N^4)$ 、メモリ量は $O(N^3)$ で膨大である。そのため、次節以降で述べるように、近似を行うことで計算量を $O(N^2)$ 、およびメモリ量を $O(N)$ へ減らした[15]、通時的誤差逆伝播法(**Back Propagation Through Time**, 以下**BPTT**とする)[16]が一般に用いられる。

以下では、まずBPTTについて説明する。そこで、BPTTには、その目的により2種類のア​​ルゴリズムが存在することを述べる。続いて、その2種類のア​​ルゴリズムである、エポック毎通時的誤差逆伝播法(**Epochwise BPTT**,

以下EBPTTとする)、およびリアルタイム通時的誤差逆伝播法(Real-Time BPTT, 以下RTBPTTとする)、についてそれぞれ述べる。そして、さらにそれらの学習アルゴリズムを用いてより学習を容易に行うための方法である、教師強制 (teacher forcing) について説明する。

2.2.1 BPTT

一般にリカレントネットの学習には、先に述べたようにBPTTアルゴリズムが用いられる。BPTTは、リカレントネットを時間展開することでフィードバック・ループを持たないニューラルネットに変換し、BPを適用する手法である。

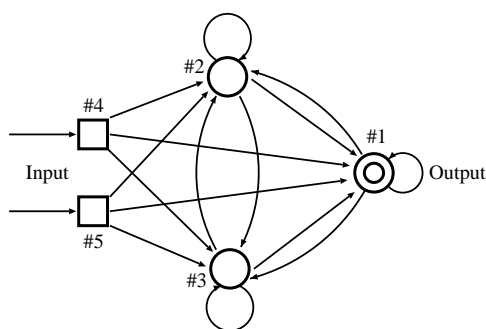


図13: リカレントネット

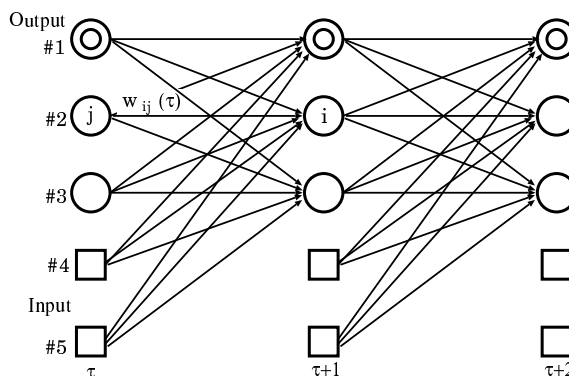


図14: 時間展開されたリカレントネット

図13にリカレントネット および図14にそれを時間展開したニューラルネットを示す。図の2個の四角は外部入力を、3個の丸がニューロンを表しており、そのうち二重丸のニューロンは外部出力となっている。図14における各層は同一時刻のニューロンであり、各ニューロンおよび各外部入力のインデックス(#1~#5)は図13のものと対応している。図14に示すように、リカレントネットを時間展開することで、フィードバック・ループを持たないニューラルネットを得ることができる。

この時得られるニューラルネットは、通常のニューラルネットとはやや異なっている。通常のニューラルネットは最初の層(入力層)にのみ外部入力を持ち、最後の層(出力層)にのみ外部出力を持つ。それに対し、得られたニューラルネットは各層に外部入出力を持つ。

このため、通常のBPをそのまま適用することはできない。通常のBPは、

出力層に入力信号が伝播した時点における出力信号と教師信号との二乗誤差を最小化するアルゴリズムである。一方、得られたニューラルネットには全ての層に外部入出力が存在するため、BPTTにおける最小化する対象は通常のBPとは異なる。

最小化する対象の違いにより、BPTTには以下の2つのアルゴリズムが提案されている。

Epochwise BPTT(EBPTT) 最後の層までの各時刻における出力信号と教師信号との二乗誤差の総和を最小化

Real-Time BPTT(RTBPTT) 現時刻における出力信号と教師信号との二乗誤差を最小化

2つのBPTTにおける最小化する対象の違いは、それぞれの学習用途に起因する。

ロボット工学等の分野では、学習はoff-lineとon-lineの2つに大別される。前者は学習用の時系列パターンを用意し、それに対し行う学習である。それに対し、後者は学習するシステムの稼働(入力に対する処理)と並行して、その時系列パターンに対し行う学習である。

off-line学習を行う場合は、当然ではあるが、学習時に学習する時系列パターン(以降、トレーニング・パターンと呼ぶ)が明らかになっている。一方、本来最小化したい対象はニューラルネットの出力系列のトレーニング・パターンに対する二乗誤差であり、これを求めることは可能である。従って、off-line学習にはEBPTTが用いられる。

これに対し、on-line学習時には、トレーニング・パターンが明らかでない。明らかになっているのはその時刻までの系列のみであり、今後の入力系列および教師系列はその時点では明らかでない。従って、上述のように本来最小化したい対象である、トレーニング・パターンに対する二乗誤差を求めることができない。そこで、各時刻においてその時刻の二乗誤差を最小化することにより将来的な誤差を小さくするRTBPTTが用いられる。

これらEBPTTとRTBPTTのそれぞれのアルゴリズムについて、次節以降で詳しく述べる。

2.2.2 EBPTT

リカレントネットにおいて、時刻 t におけるユニット（外部入力またはニューロン） i の出力を $z_i(t)$ とする。ここで、時刻 t における i 番目の外部入力を $x_i(t)$ 、ニューロンの出力を $y_i(t)$ とすれば、 $z_i(t)$ は

$$z_i(t) = \begin{cases} x_i(t), & i \in I \text{ のとき} \\ y_i(t), & i \in H \cup O \text{ のとき} \end{cases} \quad (6)$$

である。ここで I, H, O はそれぞれ外部入力、外部出力以外のニューロン、外部出力のニューロンのインデックスの集合を表す。

各ニューロンの入出力関係は以下のように表せる。

$$y_i(t+1) = f_i(s_i(t+1)), \quad i \in H \cup O \quad (7)$$

$$s_i(t+1) = \sum_{j \in I \cup H \cup O} w_{ij} z_j(t) \quad (8)$$

ここで、 w_{ij} はニューロン j からニューロン i への結合荷重、 $s_i(t+1)$ はニューロン i の内部状態を表す。評価関数 $f_i(x)$ は、2.1.2節で前述したように、最急降下法に基づく学習アルゴリズムを用いる場合は結合荷重の変数を求める際に微分されることから、微分可能な単調増加関数が用いられる。一般には

$$f(x) = \frac{1}{1 + \exp(-\beta x)} \quad (9)$$

で表される、シグモイド関数が用いられる。ここで β は関数の変化する急激さを定める正の定数であり、 $\beta = 1.0$ と設定されることが多い[7]。

リカレントネットに対して、時刻 t_0, t_0+1, \dots, t_1-1 において、それぞれ入力として $x_i(t_0), x_i(t_0+1), \dots, x_i(t_1-1)$ を与え、それぞれに対応する教師信号として、時刻 t_0+1, \dots, t_1-1, t_1 において、 $d_i(t_0+1), \dots, d_i(t_1-1), d_i(t_1)$ を与えたとする。

このときネットワーク全体の誤差関数は、各時刻における誤差の総和

$$J(t_0, t_1) = \sum_{\tau=t_0+1}^{t_1} E(\tau) = \frac{1}{2} \sum_{\tau=t_0+1}^{t_1} \sum_{i \in H \cup O} e_i(\tau)^2 \quad (10)$$

で与えられる。

ただし、 $e_i(t)$ は

$$e_i(t) = \begin{cases} d_i(t) - y_i(t), & i \in O \text{ のとき} \\ 0, & \text{その他のとき} \end{cases}$$

である。

最急降下法に基づいて結合荷重を変更するためには、結合荷重の変量を Δw_{ij} として

$$\Delta w_{ij}(t) = -\eta \frac{\partial J(t_0, t_1)}{\partial w_{ij}} \quad (11)$$

に従って結合荷重を変更すればよい。 η は学習率と呼ばれる。

ここで、時間展開されたリカレントネットの各層の結合荷重は独立であると見なし、時刻 τ のニューロン j から時刻 $\tau + 1$ のニューロン i への結合荷重を $w_{ij}(\tau)$ と表す。これにより、 $J(t_0, t_1)$ の w_{ij} に対する傾き $\partial J(t_0, t_1)/\partial w_{ij}$ は、以下のように表せる。

$$\frac{\partial J(t_0, t_1)}{\partial w_{ij}} = \sum_{\tau=t_0}^{t_1-1} \frac{\partial J(t_0, t_1)}{\partial s_i(\tau+1)} \frac{\partial s_i(\tau+1)}{\partial w_{ij}(\tau)} \frac{\partial w_{ij}(\tau)}{\partial w_{ij}} \quad (12)$$

ここで、各時刻 τ ごとの結合荷重 $w_{ij}(\tau)$ は時刻 τ には依存しない定数 $w_{ij}(\tau) = w_{ij}$ であると仮定して、誤差関数 $J(t_0, t_1)$ を結合荷重で偏微分すれば

$$\frac{\partial J(t_0, t_1)}{\partial w_{ij}} = \sum_{\tau=t_0}^{t_1-1} z_j(\tau) \frac{\partial J(t_0, t_1)}{\partial s_i(\tau+1)} \quad (13)$$

と求められる。

この結合荷重の仮定に基づく近似により、以下の計算を行うことで、EBPTT ではニューロン数を N とした場合、その計算量は $O(N^4)$ から $O(N^2)$ へ、メモリ量は $O(N^3)$ から $O(N)$ へ減らすことができる。

式(13)において

$$\delta_i(\tau) = \frac{\partial J(t_0, t_1)}{\partial s_i(\tau)}, \quad t_0 + 1 \leq \tau \leq t_1$$

とおけば、 $J(t_0, t_1)$ を極小化するような学習則は、 $\tau = t_1, t_0 + 1 \leq \tau \leq t_1 - 1$ に依存して

$$\delta_i(t_1) = (y_i(t_1) - d_i(t_1)) f'_i(s_i(t_1)) \quad (14)$$

$$\delta_i(\tau) = f'_i(s_i(\tau)) * \left[(y_i(\tau) - d_i(\tau)) + \sum_{l \in H \cup O} w_{li} \delta_l(\tau + 1) \right] \quad (15)$$

で与えられる。

したがって式(11)から、結合荷重の更新は

$$\Delta w_{ij}(t) = -\eta \sum_{\tau=t_0}^{t_1-1} \delta_i(\tau+1) z_j(\tau) \quad (16)$$

に従って行えばよい。

また、結合荷重の初期値(これを以降、初期重みと呼ぶ。)は、一般に小さな乱数に設定される。これは、もしすべての結合荷重を、例えば0のような同じ値に設定して学習を開始すれば、前述したように、誤差は結合荷重に比例して伝搬されるので、すべての結合荷重は同じように変化してしまい、非対称な解は決して得られないからである。

2.2.3 RTBPTT

本節では、RTBPTT のアルゴリズムを、前節の EBPTT と異なる点について説明する。リカレントネットにおける、時刻 t におけるユニット(外部入力またはニューロン) i の出力 $z_i(t)$ は式(6)、各ニューロンの入出力関係は式(7)から式(9)で定義されるものと、それぞれ同じであるとする。

RTBPTT では、トレーニング・パターン全体の入力と教師信号を与える EBPTT の場合とは異なり、時刻 t_0 における入力 $x_i(t_0)$ と時刻 t のみにおける教師信号 $d_i(t)$ を与えたとき、時刻 t における出力ユニット $i \in O$ の出力 $y_i(t)$ が教師信号 $d_i(t)$ になるように学習を行う。

このときネットワーク全体の誤差関数は、

$$E(t) = \frac{1}{2} \sum_{i \in O} (e_i(t))^2 \quad (17)$$

で与えられる。

ただし、 $e_i(t)$ は

$$e_i(t) = \begin{cases} d_i(t) - y_i(t), & i \in O \text{ のとき} \\ 0, & \text{その他のとき} \end{cases}$$

である。

最急降下法に基づいて結合荷重を変更するためには、結合荷重の変量を Δw_{ij} として

$$\Delta w_{ij}(t) = -\eta \frac{\partial E(t)}{\partial w_{ij}} \quad (18)$$

に従って結合荷重を変更すればよい。 η は前節と同様、学習率を表している。

この誤差関数 $E(t)$ を極小化するために、各時刻 τ ごとの結合荷重 $w_{ij}(\tau)$ は時刻 τ には依存しない定数 $w_{ij}(\tau) = w_{ij}$ であると仮定して、誤差関数 $E(t)$ を結合荷重で偏微分すれば

$$\frac{\partial E(t)}{\partial w_{ij}} = \sum_{\tau=t_0}^{t-1} z_j(\tau) \frac{\partial E(t)}{\partial s_i(\tau+1)} \quad (19)$$

と求められる。

そこで、

$$\delta_i(\tau) = \frac{\partial E(t)}{\partial s_i(\tau)}, \quad t_0 + 1 \leq \tau \leq t_1$$

とおけば、 $E(t)$ を極小化するような学習則は、 $\tau = t, t_0 + 1 \leq \tau \leq t - 1$ に依存して

$$\delta_i(t) = (y_i(t) - d_i(t)) f'_i(s_i(t)) \quad (20)$$

$$\delta_i(\tau) = f'_i(s_i(\tau)) \sum_{l \in H \cup O} w_{li} \delta_l(\tau+1) \quad (21)$$

で与えられる。

したがって式 (18) から、結合荷重の更新は

$$\Delta w_{ij}(t) = -\eta \sum_{\tau=t_0}^{t-1} \delta_i(\tau+1) z_j(\tau) \quad (22)$$

に従って行えばよい。

また、初期重みは、前節同様、一般に小さな乱数に設定される。

2.2.4 教師強制

前節までの EBPTT や RTBPTT のような リカレント ネット の学習アルゴリズムに対して、より学習が容易になる方法として、Williams と Zipser による教師強制 (teacher forcing) [16] [15] と呼ばれる学習法がある。そこで以下では、この学習法について説明する。

階層型ニューラルネット の BP による学習においては、教師信号を出力との誤差としてのみ用いていたが、リカレント ネット においてはもう 1 つ別の利用法がある。リカレント ネット においては、出力ニューロンの出力も フィードバック・ループ によって他のニューロンの出力に影響を与える。そのため、1 時刻前の正確な出力値 (教師信号) を入力として用いることが考えられる。このよ

うな観点から、Williams と Zipser は、教師強制 と呼ばれる学習法を提案した。

教師信号を上述のように利用する場合には、EBPTT・RTBPTT とともに2.2.2節の式(7),(8) で用いる $z_i(t)$ として、式(6) の $z_i(t)$ の代わりに

$$z_i(t) = \begin{cases} x_i(t), & i \in I \text{ のとき} \\ y_i(t), & i \in H \text{ のとき} \\ d_i(t), & i \in O \text{ のとき} \end{cases} \quad (23)$$

を用いればよい。

この時、出力のフィードバックの代わりに与える教師信号は、外部入力として扱う。以下、これを図15 と図16 を用いて具体的に説明する。

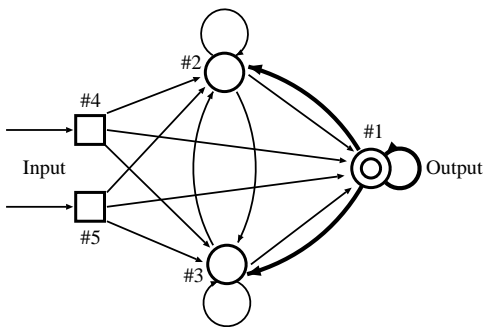


図15: リカレントネットの例

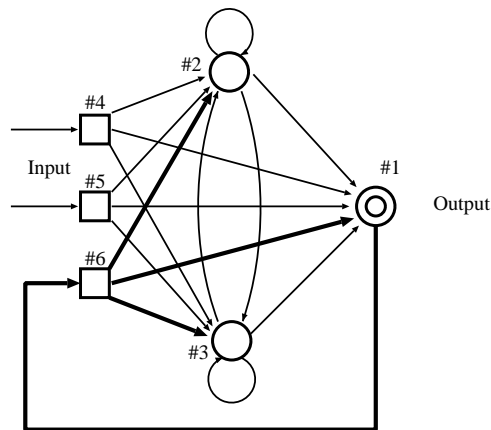


図16: 教師強制の際のリカレントネット

例えば、図15 は、ニューロン数3個のリカレントネットであり、四角や丸などの各記号の意味は、これまでの図と同様である。このリカレントネットに教師強制を用いる場合、図15の太い矢印で示すフィードバックは図16における太い矢印と等価である。ここで、図15と図16における各インデックスはそれぞれ対応している。

図16において、FPを行う際の外部出力#1からのフィードバックは、その値が#6の外部入力にコピーされ、そこから他の外部入力と同様に各ニューロンに結合されることによって実現される。これに対し、教師強制による学習時は、#6の外部入力にコピーされるのは、外部出力#1からフィードバックされる値に対応する教師信号である。

上記のような教師強制と呼ばれる学習法を EBPTT および RTBPTT に適用する場合について述べる。

EBPTT における教師強制では、これまでの EBPTT の学習則の式 (15) は

$$\frac{\partial d_l(t)}{\partial w_{ij}} = 0 \quad (l \in O) \quad (24)$$

に注意すれば、 $\tau = t_0 + 1 \leq \tau \leq t_1 - 1$ において

$$\delta_i(\tau) = f'_i(s_i(\tau)) * \left[(y_i(\tau) - d_i(\tau)) + \sum_{l \in H} w_{li} \delta_l(\tau + 1) \right] \quad (25)$$

と変更されることがわかる。

一方、RTBPTT における教師強制でも、EBPTT の場合と同様に式 (24) に注意すれば、これまでの RTBPTT の学習則の式 (21) は $\tau = t_0 + 1 \leq \tau \leq t_1 - 1$ において

$$\delta_i(\tau) = f'_i(s_i(\tau)) \sum_{l \in H} w_{li} \delta_l(\tau + 1) \quad (26)$$

と変更されることがわかる。

以上のようにすることで、ネットワークの出力がフィードバックされた入力に関しては正しい値が保証されるため、学習の際に出力を正しい値にすることに集中することができる。このため、学習がより早く収束することが知られている。

本章のまとめ 以上のように本章において説明してきた連続状態機械とリカレントネットの関係について、および、リカレントネットの学習についてまとめると、以下のようになる。

1. 連続状態機械とリカレントネット

(a) 連続状態機械

- i. 入出力は離散値、状態は連続値であり、時間は離散的に扱われる。
- ii. その能力は、連続値を扱えるため、チューリング機械よりも能力が高く万能である。

(b) リカレントネット

- i. ニューロン・モデルとして、準線形素子モデルが用いられる。
- ii. フィードバック・ループを持つニューラルネットであり、順序回路と同様に時系列処理を行うことができる。

- iii. 結合荷重として連続値を扱うことができるので、チューリング機械を超える能力を持つ。
- iv. 実験的には、有限状態機械よりも能力が高く、文脈自由文法には及ばないまでも、それに近いクラスを処理できることが示されている。

2. リカレントネットの学習

(a) 採用する学習アルゴリズム

- i. BP と同様の最急降下法に基づくアルゴリズムのうち、厳密なものより計算量・メモリ量が少ない BPTT を用いる。
- ii. BPTT には、最小化する対象の違いにより、EBPTT と RTBPTT の2つのアルゴリズムがある。

(b) **EBPTT**

最後の層までの各時刻における出力信号と教師信号との二乗誤差の総和を最小化するアルゴリズムである。

(c) **RTBPTT**

現時刻における出力信号と教師信号との二乗誤差を最小化するアルゴリズムである。

(d) 教師強制

学習中に、出力をフィードバックさせる代わりに、その出力に対応する教師信号をフィードバックさせることで、より学習を容易にする方法である。

次章では、以上の背景を踏まえ、我々の研究の目的である 時間連続状態機械について述べ、その実現の方法について説明する。

第3章 時間連続状態機械とその学習方法

前章で述べたように、従来研究されてきた連続状態機械は、連続な状態空間とその上の離散的な状態遷移によって定義された状態機械である。そして、これはリカレントネットによって実現される。

しかし、実はリカレントネットは時間連続な入力も処理することができる [17]。それにも関わらず、従来のリカレントネット研究においては、2.1.3 節のように、Reber 文法が生成するアルファベット列という時間方向に離散的な入力を扱っている。

そこで、我々の研究の目的は、時間方向にも連続な連続状態機械を実現することとする。我々は、そのような状態機械を時間連続状態機械 (Temporally Graded State Machine) と呼ぶことを提唱する。リカレントネットは時間連続な入力を処理できることから、リカレントネットによって時間連続状態機械を実現できる可能性があると考えられる。

そこで以下、まず次節では、その時間連続状態機械について、従来の有限状態機械や連続状態機械との比較を交えて説明する。また、その能力についての考察も行う。

続いて3.2節では、時間連続状態機械をリカレントネットによって実現するための、研究の方法について説明する。リカレントネットは一般に学習によって構成されるので、実際に用いる学習アルゴリズムについて述べる。さらに、リカレントネットに時間連続状態機械を学習させる上での諸問題とその解決法を述べる。

3.1 時間連続状態機械

本節では、時間連続状態機械とは何かを説明する。そこで以下では、まずその入出力と状態遷移について、有限状態機械や連続状態機械との比較を交えて述べる。続いて、その能力についても、それらの能力との比較を交えて説明する。

入出力および状態遷移 時間連続状態機械を説明するために、2.1 節で有限状態機械と連続状態機械の入出力や状態遷移を比較したように、有限状態機械や連続状態機械との対比を行う。

まず、時間連続状態機械の入出力は共に連続値を扱える。そして、その状態

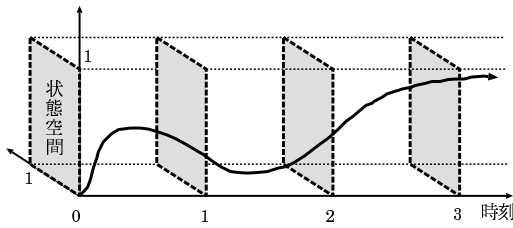


図 17: 時間連続状態機械における状態遷移

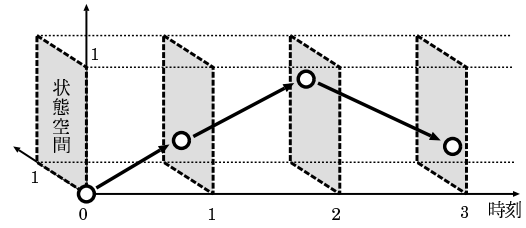


図 18: 連続状態機械における状態遷移 (再掲)

遷移は、例えば図 3.1 の曲線の矢印で示した軌跡のようになる。ここで、横軸は時刻、縦軸と奥向きの軸によって 2 次元の状態空間を表している。比較のために、連続状態機械の状態遷移の様子を表した図を図 3.1 に再掲しておく。

図 3.1 と図 3.1 を比較するとわかるように、連続状態機械の状態遷移が図 3.1 の白抜きの丸で示すように時間方向に離散的に行われるのに対し、時間連続状態機械では、時間方向にも連続的に状態遷移が行われる。すなわち、連続状態機械において、ある時刻 t の現状態が $s(t)$ であった時の次状態が $s(t+1)$ で表現されるのに対し、時間連続状態機械における次状態は微小時間 Δt 後の状態 $s(t+\Delta t)$ で表現される。

以上から、時間連続状態機械の入出力、状態、時間の扱いについて、有限状態機械と連続状態機械との対比をまとめると以下のようになる。

1. 有限状態機械
 - 入出力・状態： 離散値
 - 時間： 離散
2. 連続状態機械
 - 入出力： 離散値
 - 状態： 連続値 (実数)
 - 時間： 離散
3. 時間連続状態機械
 - 入出力・状態： 連続値 (実数)
 - 時間： 連続

時間連続状態機械の能力 上記のような時間連続状態機械の能力について考察する。

2.1.1 節で述べたように、従来研究されてきた連続状態機械は、有理数だけで

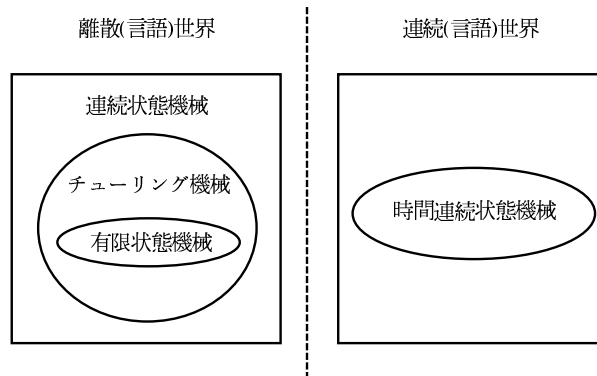


図 19: 時間連続状態機械の機械クラスにおける位置付け

なく無理数も扱えることからチューリング機械よりも能力が高く万能である。

しかし、前述したように、連続状態機械において時間は離散的に扱われる。これは、従来の連続状態機械までの状態機械においては、例えばアルファベットのようないち離散的なシンボルによって構成される、時間離散的な言語を処理するためである。

これに対し、時間連続状態機械は入出力および状態に連続値が扱えるだけでなく、時間の扱いも連続的である。つまり、これは時間連続的な言語を処理するものである。

このように処理する言語の違いにより、時間連続状態機械はその計算能力において従来の有限状態機械や連続状態機械と同列に比較することができないため、図 19 のようにそれらとは異なる枠組を提供すると言える。

以上のような時間連続状態機械は、十分ゆっくり変化する連続な入力をリカレントネットに与えることによって実現することができる。そこで、時間連続状態機械をリカレントネットで実現するための方法について、次節で説明する。

3.2 学習における諸問題と解決法

時間連続状態機械をリカレントネットで実現するにあたり、第 1 章で述べたように、今回その簡単な例として、時間連続な入力に対するポジティブエッジトリガの 1bit-モジュロ・カウンタをリカレントネットに学習させることにした。

ここで、2.2.1 節で述べたように、BPTT の学習アルゴリズムには、主に on-

line 学習に用いられる RTBPTT と、off-line 学習に用いられる EBPTT の2つがあり、どちらを学習に用いるかを決める必要がある。

1bit-モジュロ・カウンタのトレーニング・パターンは長さが有限長のものを学習前に用意できることから、off-line 学習を行えばよい。また、現時刻の教師信号と出力との誤差を最小化する RTBPTT に対して、トレーニング・パターン全体の誤差を最小化する EBPTT の方がより厳密に出力を教師信号に近づけるアルゴリズムであると言える。

したがって、先の2つのうち、有限長のトレーニング・パターン全体の誤差を最小化する EBPTT アルゴリズムを用いることにする。

そこで以降、EBPTT アルゴリズムを用いてリカレントネットに学習を行う際の諸問題と解決法について述べていく。まず次節では、与える教師信号について述べる。続いて3.2.2節では、採用するリカレントネットのモデルについて述べる。そして3.2.3節では、我々が採用する学習の方法について説明する。

3.2.1 教師信号

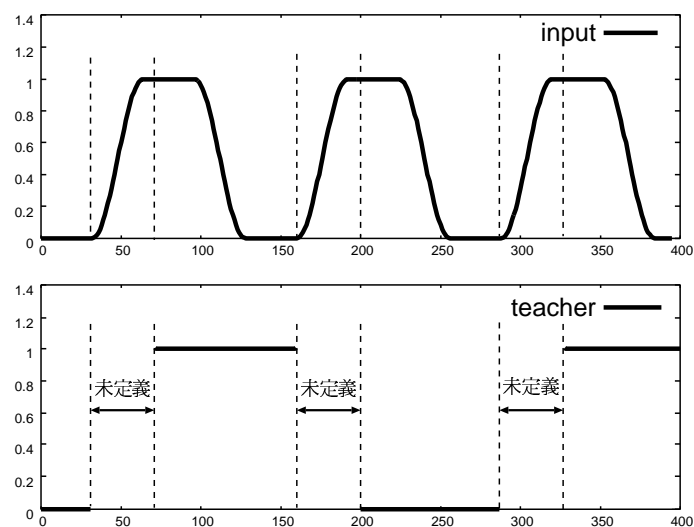


図 20: 与える入力と教師信号

本節では、時間連続な入力のポジティブエッジトリガの1bit-モジュロ・カウンタを学習させる際に、時間連続に変化する入力に対応する教師信号を定義する方法について説明する。そこで以下では、まず入力について述べ、続いてその入力に対応する教師信号について述べる。

入力 ポジティブエッジトリガの1bit-モジュロ・カウンタにおける時間連続な入力とは、図20のようなものである。入力は一度0(low値)をとるとその値でしばらくの間安定し、その後、例えばsin関数で示されるような緩やかなカーブを描きながら1(high値)へと連続的に変化する。high値からlow値への変化も同様とする。

教師信号 このような入力に対して望ましい出力、すなわち教師信号を定義する必要がある。このとき、入力が一定の値で安定している場合には望ましい出力は存在するが、入力が連続的に変化している間の1bit-モジュロ・カウンタとしての正しい動作というものは存在しない。

そのため、望ましい出力値が存在しないにもかかわらず、その間の教師信号を予め定義することによって、入力が安定している際の、望ましい出力が存在する場合の学習にまで影響を及ぼすことは避けるべきである。

また次のような理由からも、入力が連続的に変化している間の教師信号を予め定義することは避けるべきである。

仮に、入力が連続的に変化している間の教師信号を予め定義したとする。

一方、前述したように、学習に用いるEBPTTは厳密に出力を教師信号に近づけようとするアルゴリズムである。そのため、EBPTTは望ましい出力が存在する区間と存在しない区間の両方の出力を同時に満たすようなリカレントネットを構成するように働くが、そのような解は存在しない可能性がある。

そのような場合には、望ましい出力が存在する区間のみ、出力がそれに十分近くなるようなリカレントネットでさえ、学習によって構成することは困難となる。

以上のような理由から、我々は、入力が連続的に変化している間にその時刻における出力を教師信号として用いることにする。この時、その時刻における誤差がトレーニング・パターン全体の学習に影響を及ぼさないことは、2.2.2節の式(15)より明らかである。このようにすることで、望ましい出力が存在しない区間の出力を適応的に学習することができる。

また、図21のように、入力信号が立ち上がり、安定状態に入るのを見て、リカレントネットはやはり出力を連続的に変化させたのち次の安定状態へと遷移することが望まれる。つまり入力信号が安定状態に入った瞬間ではなく、そのしばらくのちに出力が安定していればよいと考える。

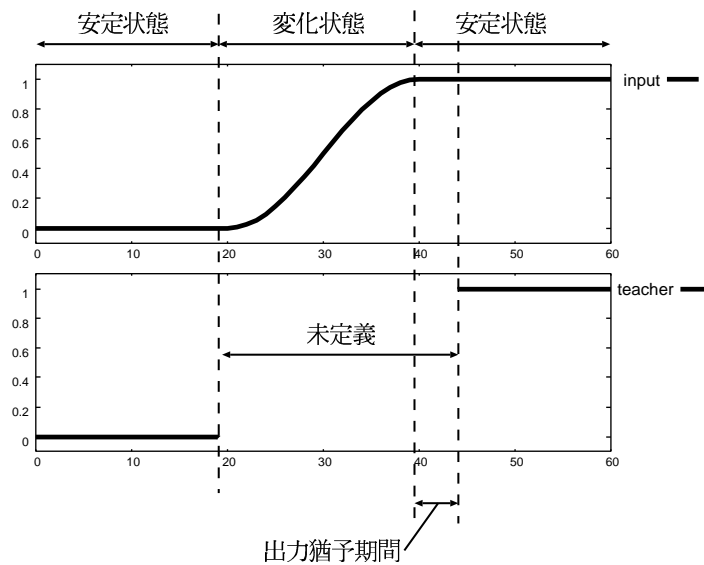


図 21: 入力の立ち上がりと教師信号

以上のことから教師信号は、入力が立ち上がってある一定期間を経た後から、次に入力が立ち上がり始めるまでの間のみ予め定義することにする(図20)。なお、以降ではこの一定期間のことを便宜的に出力猶予期間と呼ぶ(図21)。

3.2.2 採用するリカレントネットのモデル

前節で定義したような教師信号をリカレントネットにEBPTTで学習させる際には、2.2節で述べたように、最急降下法に基づく学習アルゴリズムを用いる場合には、学習の収束に最適な層の数やニューロン数が予めわからない、という問題がある。そこで本節では、この問題を解決できるリカレントネットのモデルについて説明する。

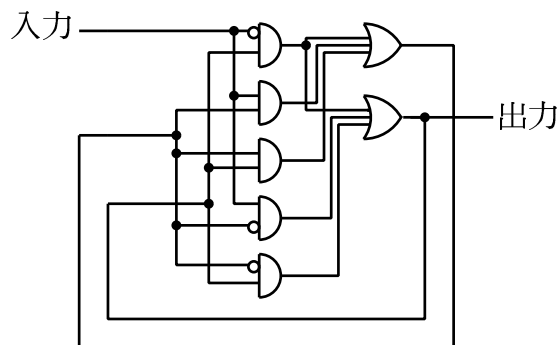


図 22: 順序回路で実現された離散の1bit-モジュロ・カウンタ

1bit-モジュロ・カウンタを順序回路で実現する場合は、図22のように、論理素子がいくつ必要で、何段の回路となるのかが、設計段階でわかる。例えば、図22では、7つの論理素子を用いた2段の回路となることがわかる。

しかし、リカレントネットにおいて、例えば同様の離散的なカウンタを実現する場合でさえ、学習するために必要なニューロンの個数や層の段数をあらかじめ与えることはできない。

そこで我々は、完全結合のリカレントネットを用いることにした。完全結合のリカレントネットならば、ニューロン数が十分であれば、任意の段数のネットワークを構造的に包含できるからである。

以下、この理由について説明する。

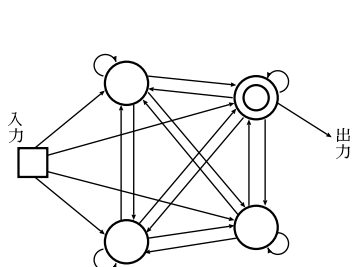


図23: 完全結合のリカレントネットの例

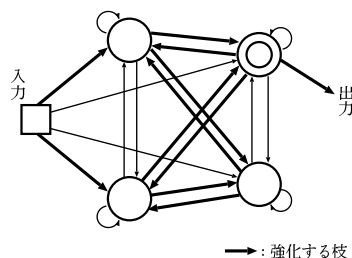


図24: 先の例のリカレントネットの一部の結合枝を強化したもの

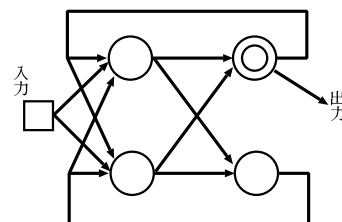


図25: 強化しなかった枝を取り除いたもの

例えば、図3.2.2のような、外部入力(図の四角)が1つで、4つのニューロン(図の丸)が完全結合となっているリカレントネットがあったとする。ただし、4つのニューロンのうちの1つは外部出力(図の二重丸)となっている。そして、このネットワークの結合枝の全ての重みの初期値は、2.2.2節で述べた理由により、十分0に近いランダムな数値で与えられているとする。

このネットワークの結合枝の重みのうち、図3.2.2の太い枝の重みだけを強化するように学習したとする。そうすると、強化されなかった枝の重みは十分0に近い値なのでその影響を無視することができ、図3.2.2のような2段の層を持ったリカレントネットを実現したことになる。

以上のように、完全結合のリカレントネットならば、ニューロンの個数を増やすだけで、任意の段数のネットワークを実現できる。

しかし、完全結合のリカレントネットに対し、3.2.1節で定義した教師信号

をEBPTTを用いて学習させる際には、次節で述べるような問題があると考えられる。そこで次節では、この問題について説明し、さらにその解決法について述べていく。

3.2.3 学習方法

3.2.1節で述べた入力と教師信号は、周期的に変化する入力において、ある時刻の入力が同じ場合に、異なる教師信号を与えている。

この時、そのままEBPTTによる学習を行う際には、通常の完全結合のリカレントネットでは、ある時刻で同一の入力に対して異なる教師信号を識別する必要があるため、目的とする動作を獲得するのが困難であることが予測される。

そこで以下、まずこの問題点について詳細に述べ、さらにその解決法を提示する。続いて、その解決法に対し、さらに学習を容易にするために、2.2.4節で説明した教師強制を適用するので、その方法について説明する。

問題 EBPTTによって、3.2.1節で定義したような入力と教師信号を、完全結合のリカレントネットにそのまま学習させる際に、以下に述べるような問題が生じることが考えられる。

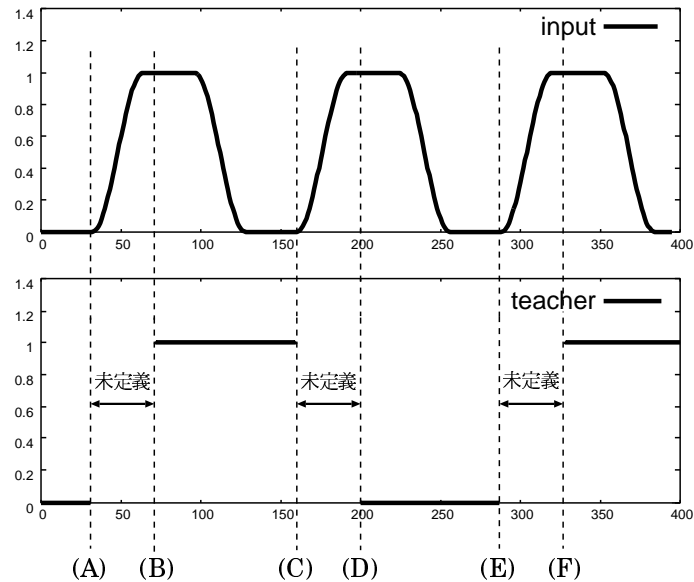


図 26: 入力と教師信号の例

3.2.1節で定義したものと同様の、図 26 のような入力と教師信号が与えられたとする。

図の通り、入力は周期的な信号であるので、全く同じ立ち上がりによって、出力の値を変化させる必要がある。

そこで、同一の入力の値に対して出力の値を変化させるには、入力が立ち上がる前におけるリカレントネットの出力の違いを利用する必要がある。例えば、図の(B)点では、リカレントネットの外部出力が(A)点でlowであったという情報を利用して、(A)点と(B)点の入力の立ち上がりにより、外部出力はhighである必要がある。同様に、図の(D)点では、リカレントネットの外部出力が(C)点でhighであったという情報を利用して、(C)点と(D)点の入力の立ち上がりにより、外部出力はlowである必要がある。

この時、図の(B)点と(D)点のリカレントネットの外部出力となっているニューロンの全ての入力を比較した場合、それらは異なっている必要がある。もし異なっていなければ、図の(B)点と(D)点で異なる出力を出すという判断をすることができなくなってしまうからである。

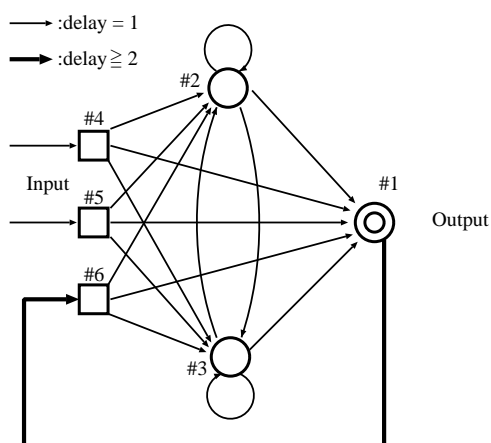


図27: 出力の遅れを入力とするリカレントネット

解決法 そこで我々は、このような予測される問題に対し、学習を容易にするため、3.2.2節で述べた完全結合のリカレントネットという構造に対して、図27のように、出力の遅れを入力に回すことにした。ここで、図27はニューロン3つの例である。図の四角は外部入力を、丸と二重丸はニューロンを表しており、二重丸は外部出力となっている。このようにすることにより、図26において、(B)点では(A)点の、(D)点では(C)点の出力を情報の識別に利用することができるようになる。

また、出力を遅らせる際の遅延の大きさは、教師信号が未定義である時間とすることにした。図27の出力ニューロン(#1)の値は、その時間分の遅延の後外部入力(#6)にコピーされる。

ここで、図27のように、通常の完全結合のリカレントネットの出力のフィードバックに遅延を与えるのではなく、出力の遅れを外部入力にまわす構造となっているのは、2.2.4節で述べた教師強制を用いることを前提としているためである。以下、このリカレントネットにおける教師強制を用いた学習について説明する。

教師強制 以上のような、出力の遅れを入力とするリカレントネットを用いる際には、2.2.4節でも述べたように、リカレントネットにおいては出力ニューロンの出力も他のニューロンの出力に影響を与える、ということに注意する必要がある。そのために、2.2.4節では、リカレントネットのそのような点に着目し、学習の過程で1時刻前の正確な出力値(教師信号)を入力として用いるという教師強制がWilliamsらにより提案されていたのであった。

そこで、我々は出力の遅れを入力とするリカレントネットに教師強制を応用することにした。以下、その方法について説明する。

2.2.4節で述べた教師強制では、リカレントネットの「1時刻前の教師信号」を、フィードバックされる出力の代わりに入力としていた。

それに対し、我々は、出力の遅れを入力とするリカレントネットを用いるので、その遅れの分だけ前の時刻の教師信号を出力の遅れの代わりに用いるのである。

このとき、教師信号を遅らせる際の遅延の大きさは、前で述べたのと同様に、教師信号が未定義である時間である。その時間分の遅れを経て、出力の代わりにその出力に対応する教師信号が図27の $delay \geq 2$ の太線で表されたフィードバック・ループを伝搬して入力となる。

我々は、この教師強制を用いたEBPTTアルゴリズムを出力の遅れを入力に回すリカレントネットに応用して学習を行う。以下、その方法について説明する。

図27のリカレントネットを時間展開すると、例えば遅延の大きさを8とすると図28のようになる。図28のインデックスは、図27の各ニューロン及び外部入力に対応している。

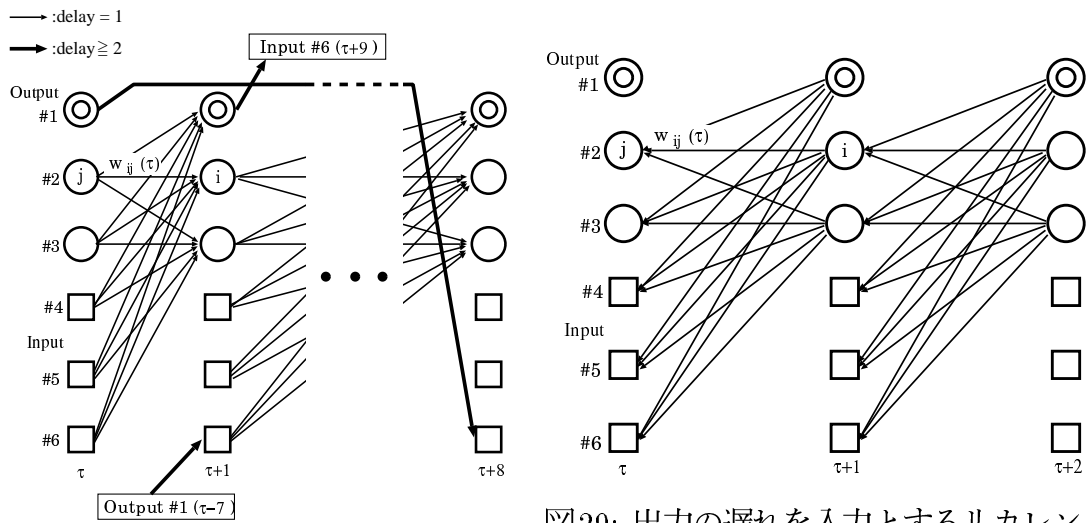


図28: 出力の遅れを入力とするリカレントネットの時間展開
 図29: 出力の遅れを入力とするリカレントネットのEBPTT

図28のようなネットワークに対して教師強制を用いたEBPTTを行う場合、誤差の伝搬の方向は図29における矢印の向きのようになる。教師強制のために出力の遅れを外部入力として扱っていることから、すべてのニューロンから出力以外のニューロン(#2,#3) および外部入力(#4~#6) への誤差の伝搬は、通常のリカレントネットの場合と同様に行われる。

しかし、通常のリカレントネットの場合と異なり、出力ニューロンへの誤差の伝搬は行われない。これは、FPにおける出力ニューロン(#1)から各ニューロン(#1~#3)への結合は、出力の遅れがコピーされる外部入力(#6)から各ニューロン(#1~#3)への結合に代替されているからである。

実際に以上の方法が有効であることを、4.1節で後述する。

本章のまとめ 以上のように本章において説明してきた時間連続状態機械について、および、その簡単な例として時間連続な入力の1bit-モジュロ・カウンタを実現する方法についてまとめると、以下のようなになる。

1. 時間連続状態機械
 - (a) 入出力、状態は連続値であり、かつ時間も連続的に扱われる。
 - (b) その能力は、連続的な言語を処理できるため、従来の離散的な言語を処理する状態機械とは異なる枠組を提供する。
2. 学習における諸問題と解決法

- (a) 採用する学習アルゴリズム トレーニング・パターン全体の誤差を最小化できる EBPTT を用いる。
- (b) 教師信号
 - i. 入力立ち上がりがある一定期間を経た後から、次に入力立ち上がり始めるまでの間のみ、望ましい出力が存在するので、予め定義する。
 - ii. それ以外の部分では、望ましい出力が存在しないので、学習中にその時刻の出力を教師信号とすることにより、望ましい出力が存在する区間の学習に望ましい出力が存在しない区間の影響を与えないようにする。
- (c) 採用するリカレントネットのモデル
 - 十分なニューロン数の完全結合のリカレントネットを用いることで、どのようなニューロン数と層数の構造でも包含できるようにした。
- (d) 学習方法
 - i. 出力の遅れを入力とするリカレントネットを用いることにより、ある程度過去の出力を利用することができ、同一の外部入力であっても、過去の異なる出力を利用して異なる教師信号を識別することができる。
 - ii. さらに、学習を容易にするために教師強制を応用し、教師信号の遅れを入力とするリカレントネットを用いることにした。

次章では、以上の方法によって実際に時間連続状態機械が実現できることを示す。

第4章 リカレント ネット における時間連続状態機械 の実現

前章で述べた方法に基づき、時間連続状態機械の簡単な例として、時間連続な入出力を行うポジティブエッジトリガの1bit-モジュロ・カウンタを実現する。1bit-モジュロ・カウンタにおける入力および教師信号のパターンは、3.2.1節で述べたようなものである。

そこで以下、まず次節では、単に通常の完全結合のリカレントネットに教師強制を用いた場合と比較することにより、出力の遅れを入力に回したリカレントネットに教師強制を用いた場合の効果を示す。続いて4.2節では、時間連続状態機械の簡単な例として、時間連続な入力に対する1bit-モジュロ・カウンタを実現するための具体的なパラメータ等を述べ、その結果としての入出力のグラフを提示する。また、その時の内部状態の入出力のグラフも提示する。

なお、以下ではFP 1回分の時間を1ステップと呼ぶことにする。

4.1 遅延の効果

本節では、単に通常の完全結合のリカレントネットに教師強制を用いた場合と、3.2.3節で述べた出力の遅れを入力に回したリカレントネットに教師強制を用いた場合とを比較する。

そこで以下では、本実験における学習に用いた種々のパラメータについて具体的に説明し、最後にそのような条件で学習を行った上記の実験の結果について述べる。

トレーニング・セット 図30のような、時間連続に変化する入力があるとする。その入力が0または1で安定している状態を**安定状態**と呼び、その時間を**安定時間**と呼ぶとする。また同様に、入力が0と1の間を連続的に変化している状態を**変化状態**と呼び、その時間を**変化時間**と呼ぶとする。

この時、比較に用いるトレーニング・セットは次のTSのようなものである。

TS 変化時間が一定で、安定時間のみが異なる入力パターンによって構成

このようなトレーニング・セットを学習することは、単にある長さのポジティブエッジで変化するカウンタを学習することである。

しかし、通常の完全結合のリカレントネットに教師強制を用いて学習を

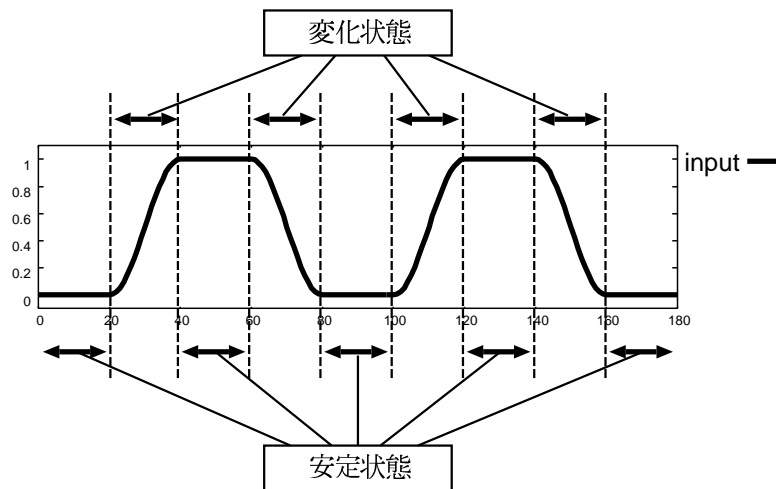


図30: 入力の安定状態と変化状態

行っても、このような簡単な問題であっても3.2.3節で述べたように過去の出力の違いを利用することができなければ、学習することは難しいと考えられる。よって、それを確かめるためには、TSのようなトレーニング・セットで十分である。

このトレーニング・セットに含まれる各トレーニング・パターンは、変化時間・安定時間共に一定の周期信号とする。1つのトレーニング・パターンは入力の4周期分、すなわち出力の2周期分とする。

このトレーニング・セットにおける安定時間の最小値は8ステップ、最大値は32ステップとする。安定時間は最小値から最大値まで4ステップ刻みの値をとる。また、変化時間は4ステップで一定とする。

1つのトレーニング・セットは、安定時間・変化時間の組み合わせによって生成される全てのトレーニング・パターンによって構成される。TSでは(安定時間, 変化時間)が(8, 4), (12, 4), ..., (32, 4)の7種類のトレーニング・パターンから成る。

なお、教師信号における出力猶予期間は、どのトレーニング・パターンにおいても4ステップとする。

学習方法 前述のトレーニング・セットを用いてEBPTTを行う。トレーニング・セットに対する学習は、その各トレーニング・パターンをサイクリックに行った。また、トレーニング・セットを学習できたか否かはEBPTT 100,000回分行った時のトレーニング・パターン全体の二乗誤差が全てのトレーニング・

パターンにおいて目標値 0.001 を下回ったかどうかで判定した。なお、学習率 η は 0.025 とした。

リカレント ネット ニューロン数は、出力の遅れを入力に回す場合、そうでない場合、それぞれ 4 個、8 個、16 個(その内の 1 個は外部出力)の完全結合のリカレント ネットを用いる。入力全部で 3 種類存在し、1 つは 2.1.2 節で述べた閾値の学習のために恒常的に 1 である入力であり、1 つは、トレーニング・パターンのための入力であり、もう 1 つが出力の遅延のための入力である。遅延は、変化時間 4 ステップと出力猶予期間 4 ステップの和である 8 ステップとした。リカレント ネットの初期重みは、-0.1 から 0.1 の範囲でランダムに定める。シグモイド関数の定数項 β は 1.0 とする。

比較 以上のようなパラメータを用いて、

RNN1 出力の遅れを入力に回した完全結合のリカレント ネット

RNN2 通常の完全結合のリカレント ネット

の 2 種類のネットワークに対して学習を 100 種類の相異なる初期重みに対して行った。その結果、前述のトレーニング・セット TS を学習できた回数は、表 1 のようになった。

	ニューロン数		
	4 個	8 個	16 個
RNN1	4	31	55
RNN2	0	0	0

表 1: トレーニング・セットを学習できた回数

表 1 が示すように、通常の完全結合のリカレント ネットでは、前述のトレーニング・セット TS を全く学習することができなかった。しかし、出力の遅れを入力に回すリカレント ネットでは、ニューロン数 4 個の場合の結果では、学習の成功率が 4% となりあまり効果が出ていないように見えるかもしれないが、それはニューロン数の問題であり、ニューロン数を 8 個、16 個と増やすことにより、有意な差が得られた。この結果より、出力の遅れを入力に回すリカレント ネットの学習における有効性は明らかである。

一方、通常の完全結合のリカレントネットの場合の学習できる回数は、それ以下であろうことは予測できたものの、全く学習できなかったことについては、何か原因があると考えられる。そこで以下ではその原因を調べるために、学習中のFPの様子を見てみることにする。

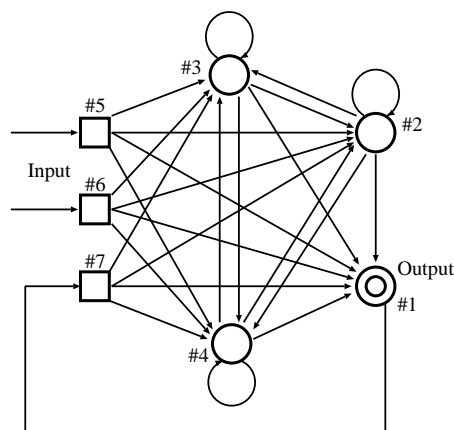


図 31: ニューロン4個の通常の完全結合のリカレントネット

通常の完全結合のリカレントネットの学習中のFP 前述のパラメータを用いて、通常の完全結合のリカレントネットに学習を行った。学習中のリカレントネットの内部を解析しやすいように、前述の3種類のうちのできるだけ少ないニューロン数を用いることにし、図31のように4個とした。

学習中(EBPTT 20,000回目)における外部入力と教師信号および各ニューロンの出力は、図32のようになった。各図のニューロンのインデックス(#1～#4)は図31のインデックスに対応している。なお、横軸はステップ、縦軸は入力・出力の値である。入力を点線、出力を実線で示してある。

出力ニューロンの出力(図32の#1)を見ると、一見入力の立ち上がりで出力を変化させることができているように見える。

しかし、実際は、他の隠れニューロン(図32の#2～#4)を見るとわかるように、出力猶予期間が終わった直後の同じ入力では、隠れニューロンの出力にほとんど差がない。例えば、入力が立ち上がって出力猶予期間が終わった時刻である16ステップ目と40、64、88ステップ目における隠れニューロンの出力を比較すると、その値にほとんど差がないことがわかる。

さらに、図32の教師信号と#1を比較すると、教師強制で入力として利用さ

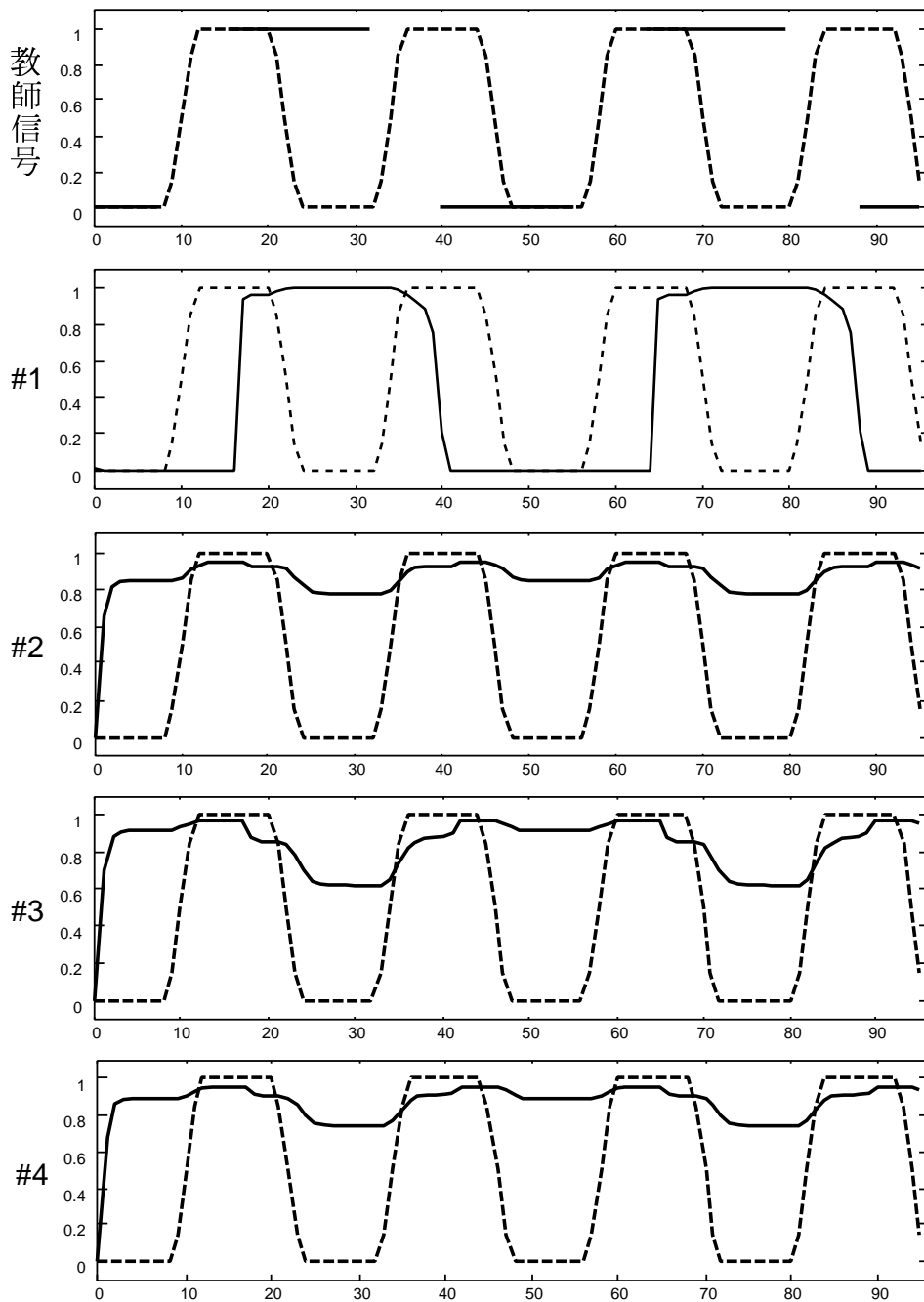


図 32: 学習中の教師信号と各ニューロン(#1~#4)の入出力: 入力(点線), 教師信号および出力(実線)

れる教師信号の変化の後、出力ニューロンの値を変化させていることがわかる。例えば、入力が立ち上がって出力猶予期間が終わった時刻である 16 ステップ目に教師信号は 1 という値を出しており、その後の 17 ステップ目に、出力ニュー

ロンの値はおよそ1となっている。これは、16ステップ目の各隠れニューロンの出力が、先に述べたように、ほとんど他の場合と違いがないため、出力ニューロンが変化するためには、16ステップ目の教師信号の1という値によるしかないからだと考えられる。

以上から、このまま学習を続けても、外部入力としての教師信号が存在しない通常のFPを行った時に、異なる教師信号を識別できないことがわかる。

しかし、なぜ学習によって隠れニューロンの出力に差が出るようにならないかは、未だ不明であり、今後の課題である。

以上のように、通常の完全結合のリカレントネットでは、時間連続な入力に対する1bit-モジュロ・カウンタを実現できず、実現するためには出力の遅れを入力に回したリカレントネットを用いた方がよいことがわかった。そこで次節では、実際にこの出力の遅れを入力に回したリカレントネットを用いて、時間連続状態機械を実現する。

4.2 時間連続状態機械の実現

本節では、実際に、時間連続状態機械の簡単な例として、時間連続な入力を処理するポジティブエッジトリガの1bit-モジュロ・カウンタを実現する。そこで、前節でその有効性を示した出力の遅れを入力に回したリカレントネットに教師強制を用いることにより、これを実現することにする。さらに、その内部の動作を調べやすいよう、できるだけ少ないニューロン数で実現したので、リカレントネットの各ニューロンの入出力も合わせて提示して、その動作を見てみることにする。

そこで以下では、本実験における学習に用いた種々のパラメータについて具体的に説明し、最後にそのような条件で学習を行った実験の結果について述べる。

トレーニング・セット トレーニング・セットは2種類用意する。それは以下のようなものである。

TS1 変化時間が一定で、安定時間のみが異なる入力パターンによって構成

TS2 変化時間・安定時間共に異なる入力パターンによって構成

TS1のトレーニング・セットを学習することは、前節のTSを学習することと同じであり、単にある長さのポジティブエッジで変化するカウンタを学習することである。そして、TS2のトレーニング・セットを学習することは何如な

る長さのポジティブエッジに対しても働くカウンタを学習することである。この2つの学習には別の難しさが存在すると考えられるため、トレーニングセットを2つに分け、段階的に学習を行う。

各トレーニング・セットに含まれる各トレーニング・パターンは、前節と同様に、変化時間・安定時間共に一定の周期信号とし、入力4周期分、すなわち出力の2周期分とする。

各トレーニング・セットにおける安定時間・変化時間のそれぞれ最小値・最大値を表2に示す。安定時間・変化時間はそれぞれ最小値から最大値まで4ステップ刻みの値をとる。

1つのトレーニング・セットは、安定時間・変化時間の組み合わせによって生成される全てのトレーニング・パターンによって構成される。例えば、TS2は(安定時間, 変化時間)が(8, 4), (8, 8), ..., (8, 128), (12, 4), ..., (32, 128)の、安定時間7種類と変化時間32種類の積である224種類のトレーニング・パターンから成る。

なお、教師信号における出力猶予時間は、前節と同様に、どのトレーニング・パターンにおいても4ステップとする。

	安定時間		変化時間	
	最小値	最大値	最小値	最大値
TS1	8	32	4	4
TS2	8	32	4	128

表2: 学習に用いたトレーニング・セット

学習方法 前述の2つのトレーニング・セットを用いてEBPTTによる学習を行う。まず

1.) TS1を用いて学習を行い、
続いて

2.) 1.)を学習済みのリカレントネットに対しTS2を用いて学習を行う。

各トレーニング・セットに対する学習は、それぞれに含まれる各トレーニング・パターンをサイクリックに行った。各トレーニング・セットに対する学習は、TS1については前節と同様にEBPTT 100,000回分を行い、TS2については、TS1よ

りもトレーニング・パターン数が多いため、EBPTT 400,000 回分行った。なお、学習率 η は 0.025 とした。

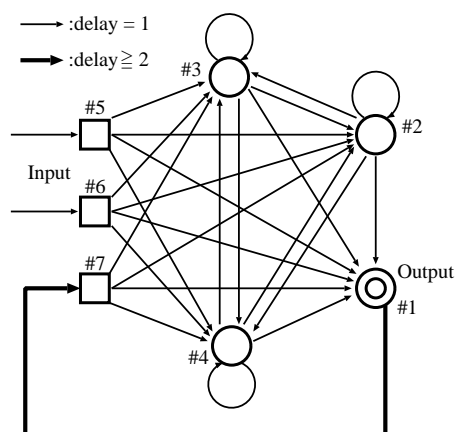


図 33: ニューロン4個の出力の遅れを入力に回した完全結合のリカレントネット

リカレント ネット 図 33 のような、ニューロン数 4 個 (その内の 1 個が外部出力) の完全結合のリカレントネットを用いる。実験後、内部の動作を見やすいようにできるだけ少ないニューロン数とした。入力の前節と同様に全部で 3 種類存在し、1 つは 2.1.2 節で述べた閾値の学習のために恒常的に 1 である入力であり、1 つは、トレーニング・パターンのための入力であり、もう 1 つが出力の遅延のための入力である。

遅延は、変化時間の最小値 4 ステップと出力猶予期間 4 ステップの和である 8 ステップで一定とした。これは、次の理由による。TS1 でまず変化時間 4 ステップで一定のポジティブエッジのカウンタを学習する際に、十分な遅延を与えて学習を可能にすることにより、過去の出力の違いを利用できるネットワークが得られることが考えられる。この後、TS2 の学習を行うことによって、遅延の大きさによらずに時間連続な 1bit-モジュロ・カウンタとしての動作を行えるようにするためである。

リカレントネットの初期重みは、-0.1 から 0.1 の範囲でランダムに定める。シグモイド関数の定数項 β は 1.0 とする。

実験結果 以上のようなパラメータを用いて学習を行った結果、変化時間が 4 から 128 ステップならば、任意の安定時間の入力に対してポジティブエッジトリガの 1bit-モジュロ・カウンタとして働くリカレントネットを実現すること

ができた。さらに、変化時間における入力の変化関数を \sin 以外に、未学習の線型・指数・対数に変えて試した。

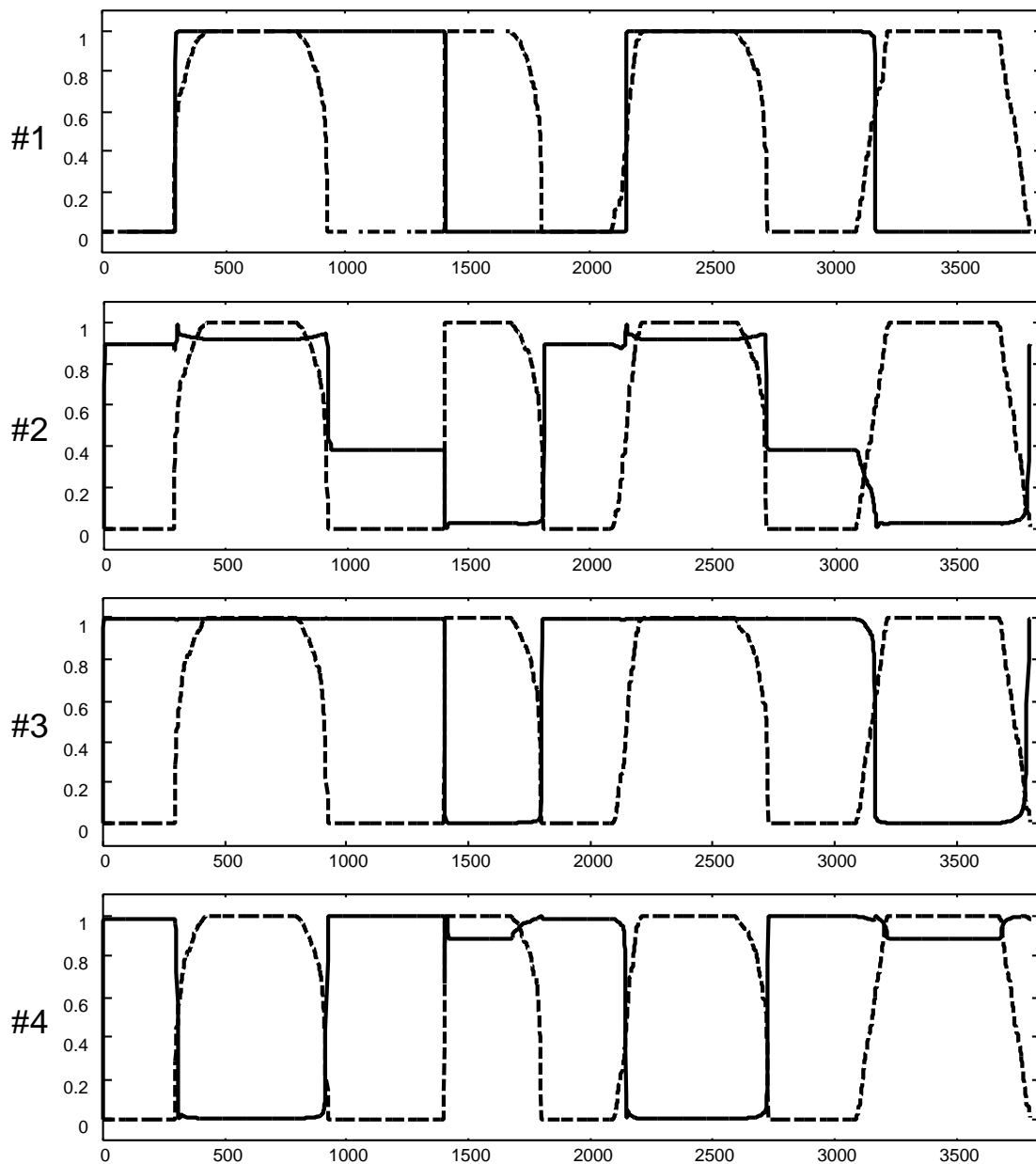


図 34: 学習後の各ニューロン (#1~#4) の入出力: 入力 (点線), 出力 (実線)

その結果、そのような入力パターンに対してもポジティブエッジトリガのカウンタとして働くことを確認した。図 34 は、変化時間を 128 ステップとし、さらに変化関数を先の 4 種類からランダムに与え、入力の安定時間を学習時の最

大値である 32 ステップよりも大きな 256 ステップから 512 ステップの間のランダムな値で与えた時の各ニューロン (#1 ~ #4) の入出力である。横軸はステップ、縦軸は入力・出力の値である。入力を点線、出力を実線で示してある。

図 34 の #1 のグラフに示したように、今回 時間連続状態機械 を学習させた リカレント ネット は時間連続な入力を処理できる。そこで以下、このような 時間連続状態機械 の動作がどのように実現されているかを、実現した リカレント ネット の隠れニューロンの動作に基づいて述べていく。

図 34 の #1 の入出力を行う際の隠れニューロン (図 33 の インデックス #2 ~ #4) の入出力を図 34 の #2 ~ #4 にそれぞれ示す。これらのグラフを見るとわかるように、図 34 の #1 のように時間連続な出力をするために、リカレント ネット は各ニューロンの出力が表す時間連続的に変化する連続値の状態を利用している。

さらに、各隠れニューロンは、同一の入力に対して異なる出力を行うことができている。これは、前節において通常の完全結合のリカレントネットに教師強制を用いた EBPTT による学習を行っても、同一の入力に対してほとんど同じ出力値を出していたことと対照的である。したがって、通常の完全結合のリカレントネットに教師強制を用いた EBPTT による学習では、いつまでも内部状態を利用した識別ができるようにならないのに対し、我々の提案した出力の遅れを入力に回したリカレントネットではそれが可能になっていると言える。

第5章 おわりに

フィードバック・ループを持つニューラルネットであるリカレントネットは、時系列情報を処理することができる。さらに連続値を扱えるため、従来の研究ではチューリング機械よりも能力が高く万能である連続状態機械という機械クラスに分類されてきた。そのため、リカレントネットは自然言語などのさまざまな応用分野が考えられる。

一方、リカレントネットは一般に実世界におけるような時間連続な入力を処理することができる。それにも関わらず従来のリカレントネット研究では、時間の扱いが離散的であった。

そこで我々は、時間方向にも連続的な状態空間と、時間方向にも連続的な状態遷移を持つ連続状態機械を、時間連続状態機械と呼ぶことを提唱した。そして、これをリカレントネットに学習させる方法を提案した。

まずリカレントネットに時間連続状態機械を学習させる際のアルゴリズムとして、BPと同様の最急降下法に基づくアルゴリズムのうち、厳密なものより計算量・メモリ量ともに少ないBPTTを用いることについて述べた。そして、RTBPTTとEBPTTという2つのBPTTアルゴリズムのうち、トレーニング・パターン全体の誤差を最小化できるEBPTTを採用することについて述べた。さらに、リカレントネットの学習をより容易にするために、教師強制を用いることにした。

そのままではリカレントネットに学習を適用することができない理由として、時間連続な入力に対して、望ましい出力が存在する部分と存在しない部分があることを述べた。そして、望ましい出力が存在する部分の教師信号のみを学習前に定義しておき、望ましい出力が存在しない部分の教師信号は、学習中にその時刻のリカレントネットの出力と一致させることで、望ましい出力が存在する部分に影響を与えないようにする、という解決法をとった。

また、採用するリカレントネットのモデルとして、十分なニューロン数を持つ完全結合のリカレントネットを用いることを述べた。これは、論理回路が設計前に段数や論理素子の個数がわかるのとは異なり、EBPTTのような最急降下法に基づく学習アルゴリズムにおいては、学習の収束に最適なリカレントネットの層の段数や、ニューロン数が事前にわからないからである。そのため、完全結合のリカレントネットを用いることにより、任意の構造が包含でき

ることについて述べた。

さらに、出力から入力への遅延枝を導入することで、リカレントネットに同一の入力に対して異なる出力を出すように、識別しやすくさせる方法について述べた。さらに、この時の教師強制を用いた学習方法についても述べた。

以上の結果として、まず通常の完全結合のリカレントネットと出力の遅れを入力に回したリカレントネットの両方に教師強制を用いた時の学習の結果を比較し、後者の有効性を示した。そして、上述のアプローチに基づき、連続的に変化する入力に対して1bit-モジュロ・カウンタとして動作するリカレントネットを学習により実現し、時間連続に変化する状態情報を利用することによって時間連続状態機械を実現していることについて述べた。

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、中島康彦助教授、五島正裕助手、津邑公暁助手、三輪忍氏、永野貴宣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] Omlin, C. and Giles, C.: Extraction of rules from discrete-time recurrent neural networks, *Neural Networks*, Vol. 9, No. 1, pp. 41–52 (1996).
- [2] McClelland, J. L., Cleeremans, A. and Servan-Schreiber: Parallel Distributed Processing: Bridging the Gap between Human and Machine Intelligence, *人工知能学会誌*, Vol. 5, No. 1, pp. 2–14 (1990).
- [3] Cleeremans, A., Servan-Schreiber, D. and McClelland, J.: Finite state automata and simple recurrent networks, *Neural computation*, Vol. 1, No. 3, pp. 372–381 (1989).
- [4] Copeland, B.J., S. R.: Beyond the Universal Turing Machine, *Australasian Journal of Physiology* (1998).
- [5] Siegelmann, H.: Computation Beyond the Turing Limit, *Science*, Vol. 268, pp. 545–548 (1995).
- [6] Stannett, M.: X-Machines and the Halting Problem: Building a Super-Turing Machine, *Formal Aspects of Computing*, Vol. 2, pp. 331–341 (1990).
- [7] Rumelhart, D., G.E., H. and R.J., W.: Learning internal representation by error propagation, *Parallel Distributed Processing*, Vol. 1, pp. 318–362 (1986).
- [8] Rumelhart, D., G.E., H. and R.J., W.: Learning representation by back-propagation errors, *Nature*, Vol. 323, pp. 533–536 (1986).
- [9] Hava, T. and Eduardo, D.: Analog Computation via Neural Networks, *Theoretical Computer Science*, No. 131, pp. 331–360 (1994).
- [10] Hava, T.: On the computational power of probabilistic and faulty neural networks., *Automata, Languages and Programming. 21st International Colloquium ICALP 94. Proceedings*, pp. 23–34 (1994).
- [11] Elman, J.: Finding structure in time, *Cognitive Science*, Vol. 14, pp. 179–211 (1990).
- [12] Servan-Schreiber, D., C. A. and McClelland, J.: Learning sequential structure in simple recurrent networks, No. CMU-CS-183 (1988).
- [13] Kolmogorov, A.: On the representation of continuous functions of many variables by surposition of continuous functions of one variables and addi-

- tion, *Dokl. Acad. Nauk USSR*, Vol. 114, pp. 956–956 (1956).
- [14] Funabashi, K.: On the approximate realization of continuous mapping by neural networks, *Neural Networks*, Vol. 2, pp. 183–192 (1989).
- [15] Williams, R. and Zipser, D.: Gradient-based learning algorithms for recurrent networks and their computational complexity, in Y. Chaubin and D.E. Rumelhart (eds.), *Backpropagation: Theory, Architectures, and Applications*, pp. 433–486 (1995).
- [16] Williams, R. and Zipser, D.: A learning algorithm for continually running fully recurrent neural networks, *Neural computation*, Vol. 1, No. 2, pp. 270–280 (1989).
- [17] 津田晃寿, 永野貴宣, 三輪忍, 津邑公暁, 五島正裕, 富田眞治: 学習による非同期連続状態機械の構成, 信学技報 CPSY2002-50 (SWoPP 2002), 信学会, pp. 59–64 (2002).