

Title	確率的学習アルゴリズムを用いた有限状態オートマトンの抽出に関する研究
Author(s)	近藤, 雅之
Citation	
Issue Date	2002-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/361
Rights	
Description	Supervisor:林 幸雄, 知識科学研究科, 修士

修 士 論 文

確率的学習アルゴリズムを用いた
有限状態オートマトンの抽出に関する研究

北陸先端科学技術大学院大学
知識科学研究科知識システム基礎学専攻

近藤 雅之

2002 年 3 月

修 士 論 文

確率的学習アルゴリズムを用いた 有限状態オートマトンの抽出に関する研究

指導教官 林 幸雄 助教授

北陸先端科学技術大学院大学
知識科学研究科知識システム基礎学専攻

050036 近藤 雅之

審査委員主査	林 幸雄 助教授
審査委員	中森 義輝 教授
審査委員	藤波 努 助教授
審査委員	佐藤 賢二 助教授

提出年月: 2002 年 2 月

目次

第1章	はじめに	1
1.1	まえがき	1
1.2	本研究の概要	3
第2章	確率的学習アルゴリズム	4
2.1	確率的学習アルゴリズムの特徴	4
2.2	S-MLP	5
2.2.1	結合荷重更新手続き	5
2.2.2	S-MLP のアルゴリズム	5
2.2.3	素子選択確率	6
2.3	PS-MLP	7
2.3.1	PS-MLP のアルゴリズム	7
2.3.2	素子選択確率	7
2.4	予備実験	8
第3章	シンプルリカレントネットワークへの適用	9
3.1	SRN	9
3.2	SRN への適用	10
第4章	実験と結果	11
4.1	実験環境	11
4.2	富田文法	11
4.3	実験手法	13
4.3.1	初期値設定	13
4.3.2	入力と出力の表現	13
4.3.3	実験方法	13

4.4	結果	14
4.4.1	ランダムデータ	14
4.4.2	網羅データ	17
第5章	考察	21
5.1	実験結果まとめ	21
5.2	今後の課題	22
第6章	まとめ	23
	謝辞	24

第1章 はじめに

1.1 まえがき

これまで、ニューラルネットワークの研究が数多くなされてきた。主にパターン認識、信号処理、制御などの、非記号的情報処理を対象とした適用が試みられてきた。ニューラルネットワークの学習能力、汎化能力、並列化による高速な計算、ロバスト性などに優れているという点に期待したものである。一方、それらを積極的に記号処理に応用しようとする研究が行なわれている。その中には再帰型神経回路網 (recurrent neural network:RNN) を用いた研究がある。RNN とは、再帰結合を持つニューラルネットワークであり、時系列のデータを取り扱うことができるという特徴がある。これを用いて、文法獲得の研究にも用いられている。記号処理を扱った研究には次のようなものがある。

Elman[3] は単純再帰型回路網 (simple recurrent network:SRN) を提案した。現在では、この SRN は RNN の最も基本的なモデルの一つとされており、様々な研究が行なわれている。Elman は SRN に、0、1 の時系列入力に対して次のビットを予測するタスクを学習させることに成功した。すなわち、SRN が時系列データを扱えることを示した。これはある種の有限状態オートマトン (finite-state automata:FSA) を学習させるのに相当する。

Servan-Schreiber ら [2] は、正規言語を認識する RNN について議論し、Reber 文法により生成される系列の逐次的予測という課題を学習させた。この結果から、隠れ層の素子群における発火パターンが過去の履歴をあらわしていること、および獲得された活性パターンのクラスタが、学習された FSA の各状態を表していると、Schreiber は主張している。

Giles ら [4] は、2 次の結合を含む相互結合型回路網を用いて、オートマトンの動作を学習する方法について報告している。この中で、ニューラルネットワークが状態数最小の FSA を獲得する場合があることを報告している。

田中ら [8] は再帰型高次結合神経回路網を用いて、Giles らと同様の実験を行ない比較を行なった。その結果、学習の成功率は向上したと報告している。

また、Giles は次のようにも述べている [7]。『Of particular concern - and an open issue - is fidelity of the extraction process,i.e. how accurately the extracted knowledge

corresponds to the knowledge stored in the network.」つまり、ニューラルネットワーク内に格納されている、情報をどのようにして正確に抽出するかが関心事であるということである。

以上のように、記号処理にはRNNが多く用いられている。どのような内部表現が隠れ層の上に形成されているかを明らかにしようとする研究がなされている。しかし、隠れ層からの抽出 (FSA 抽出も含む) は、非常に困難な問題である。通常のフィードフォワード型のネットワークにおいて、多項式時間内で FSA を正確に獲得できるアルゴリズムが存在しないことが証明されている [5]。したがって、記号処理に RNN を用いた研究が多くなされている。また、隠れ層から FSA を抽出するという課題は文法獲得の研究において、重要な課題であると考えられる。

従来の FSA 抽出を行なっている研究では、素子の出力関数にシグモイド関数を用いている。こうしたものは、閾値素子をシグモイド関数によって近似し、通常の誤差逆伝搬法 (back propagation:BP 法) やそれを改良したものをを用いている。そして、学習したネットワークの隠れ層から FSA の抽出を行なっているのである。

しかし、このシグモイド関数は連続値をとるために、その活性値パターンから、学習したと考えられる FSA の状態遷移を抽出することは非常に困難であった。一方、ネットワークを構成する素子としてシグモイド関数の代わりに線形閾値素子を用いれば、FSA の抽出は容易に行なえると考えられる。その理由は、線形閾値素子の出力値は離散値であるからである。しかし、RNN のような多層神経回路網 (multilayer perceptron:MLP) の学習アルゴリズムとして提案されたものには、線形閾値素子を対象とするものはなかった。というのも、線形閾値素子の出力値は離散値で表現されるため、出力関数は微分可能でなく、BP 法が適用できないからである。しかし、素子一個に対するアルゴリズムは存在していた。最近、櫻井 [1] が線形閾値素子を用いた MLP の学習アルゴリズム、確率的学習アルゴリズム (stochastic learning algorithm for MLP:S-MLP) を提案した。このアルゴリズムは通常の BP 法とは異なり、大域的な収束性を示すと報告している。また、高速に収束するとも報告されている。そして、線形閾値素子を用いているために、解が離散値で表現されるために解の解釈が容易に行なえるとも主張している。本研究では、素子の出力値が離散である回路網が学習できるという特徴に着目した。

そこで、本研究では S-MLP の SRN への適用を試みる。その検証として簡単な正規言語を用いて学習を行ない、中間層の状態から形成されているであろう FSA の抽出を行なう。

1.2 本研究の概要

本研究の目的としてはまず、線形閾値素子を用いた学習アルゴリズムである S-MLP を SRN に適用することにある。そして、実際に FSA を学習可能かどうかを明らかにすることにある。そのための実験として、まず、正規言語を生成する FSA の学習を試みる。また、S-MLP は線形閾値素子を用いているために、出力値が離散値であり、解の解釈が容易であることから、実際に学習終了後のネットワークの隠れ層の状態から FSA の抽出を試みる。その結果、どのような FSA を獲得できたかについても調べる。実際に、研究では正規言語の一種である富田文法を用いる。学習に際しては、ある時刻の入力が受理状態か非受理状態かを教師信号として与える。すなわち、(実際には有限個の例の提示ではあるが) 極限としては完全提示を行なっていることになる。通常の言語学習の枠組みとしてよく用いられる、統語的に正しい文の次単語を予測する枠組みや、各文が文か非文 (正例か負例) を教示する枠組みとは異なる。

本論文ではまず、第 2 章で S-MLP の特徴やアルゴリズムについて詳しく述べる。次に、第 3 章で SRN について述べた後、S-MLP を SRN へ適用したときの学習アルゴリズムを述べる。そして、第 4 章で富田文法と実験の手法について述べた後、第 5 章で実験結果の考察と今後の課題を述べる。最期の第 6 章では本研究のまとめを示す。

第2章 確率的学習アルゴリズム

この章では、本研究で用いている学習アルゴリズムについて、詳しく述べる。

2.1 確率的学習アルゴリズムの特徴

線形閾値素子からなる多層神経回路網 (multilayer perceptrons:MLP) を対象とした学習アルゴリズムであり、櫻井によって提案された [1]。MLP を対象とした学習アルゴリズムとして提案されたものには、線形閾値素子を対象とするものはなかった。というのも、線形閾値素子は出力値が離散値であるために、出力関数が微分ができず、BP 法など従来の学習アルゴリズムを適用することができないからである。しかし、素子一個の回路網の場合には、線形閾値素子を対象としたアルゴリズムが存在しており、非常に高速であることが知られている。そこで、櫻井は、線形閾値素子を対象とした MLP の学習アルゴリズムを提案した。このアルゴリズムは on-line 型であり、データが提示される毎に回路網内の素子を確率的に選択し、それへの結合荷重を Rosenblatt のパーセプトロン・アルゴリズムに従い更新するものである。さらに、通常の BP 法とは異なり、大域的な収束性を有する。すなわち、解が存在するならば確率 1 で収束する。また、解が存在しない場合は、結合荷重が構成する有限な空間の中を巡回する。

また、線形閾値素子を用いているために、素子の出力が離散値で表現される。これにより、解の解釈が容易になると櫻井は主張している。

以上のような特徴を持つ S-MLP であるが、本研究では、S-MLP が線形閾値素子からなる MLP を対象にした学習アルゴリズムであることから、出力値が離散値である回路網を学習できるという点に着目した。

この章では、S-MLP、S-MLP を並列処理を可能にした PS-MLP について詳しく述べる。

2.2 S-MLP

2.2.1 結合荷重更新手続き

素子一個の回路網の場合、線形閾値素子を用いた高速な学習アルゴリズムである、Rosenblatt のパーセプトロン・アルゴリズムが存在する。S-MLP では結合荷重の更新にこのパーセプトロン・アルゴリズムを用いる。

$$\begin{array}{ll} \text{if } \mathcal{H}(\mathbf{w} \cdot \mathbf{x}) < 0 & \text{then } \mathbf{w} \leftarrow \mathbf{w} + \mathbf{x} \\ & \text{else } \mathbf{w} \leftarrow \mathbf{w} - \mathbf{x} \end{array}$$

ここで、 $\mathcal{H}$ は Heaviside 関数 (図 2.1) であり、正值の入力に対しては 1、負値または 0 の入力に対しては -1 を返す関数である。また、 $\mathbf{w}$ は結合荷重ベクトル、 $\mathbf{x}$ は入力ベクトルである。以後、この手続きをパーセプトロン更新手続きと呼ぶ。荷重更新はすべてこの手続きに従って行なう。

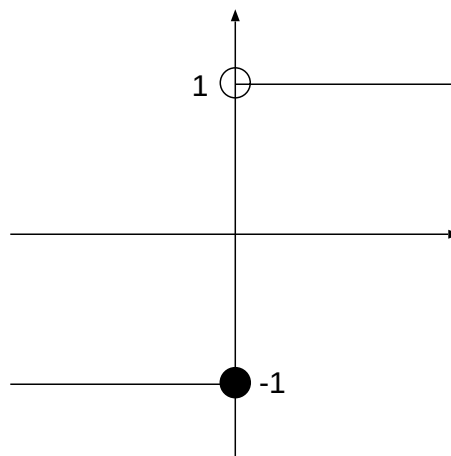


図 2.1: Heaviside 関数

2.2.2 S-MLP のアルゴリズム

S-MLP のアルゴリズムは図 2.2 のようなステップで行われる。図中の 5 で用いられる確率は、その時点での結合荷重と外部入力にのみ依存して定められる。この確率については 2.2.3 で述べる。

1. 全ての結合荷重を初期化し、全てのデータのラベルを `unsatisfied` とする。
2. 全てのデータが収束する (`satisfied` とラベル付けされる) まで 3 から 7 の処理を繰り返す。
3. 未収束 (`unsatisfied` とラベル付けされている) データから 1 つランダムに選択し、計算を行う。
4. 回路網の出力が教師信号と異なる場合は 5 以下の処理を行い、同じであった場合はそのデータのラベルを `satisfied` とし、3 へ。
5. 全ての素子に確率を割り当て、その確率に従ってランダムに素子を一個選択する。
6. 選んだ素子への荷重結合をパーセプトロン更新手続きに従って更新する
7. 全てのデータを `unsatisfied` とラベル付けし、3 へ。

図 2.2: S-MLP アルゴリズム

このアルゴリズム中において、あるデータが `unsatisfied` とラベル付けされているとは、そのデータでの学習は終了していない (未収束) ということを表す。逆に、学習が終了 (収束) した場合はそのデータは `satisfied` とラベル付けされる。このようにラベル付けを行うことによって、学習と同時にテストを行っていることとなる。

2.2.3 素子選択確率

ある素子への結合荷重の更新であって、その更新を行なうことにより、回路網の出力が期待出力値 (つまり教師信号) に変わる可能性がある更新を、`effectual` な更新と定義する。アルゴリズムの 5 では、この `effectual` な更新をもつ素子にのみ確率を割り当てる。割り当てる確率は、選択のためのバイアスを割り当て、それを全素子について合計し正規化することによって、得ることができる。そのバイアスには試行錯誤により求められた次の式を各素子に用いる。

$$\exp \left(- \left(\frac{\mathbf{w} \cdot \mathbf{x}}{\|\mathbf{w}\| \cdot \|\mathbf{x}\|} \right) (1.9n) \right) \quad (2.1)$$

ここで、 n はこの素子への入力数であり、また、 $n = 2, 3$ の時は、 $1.9n$ のかわりに $2.2, 3.0$ を用いる。

2.3 PS-MLP

S-MLP では、荷重更新を行う素子を決定するためには、全素子間で調整を行う必要があるが、PS-MLP はこの問題をなくしたアルゴリズムである。

本研究では、この PS-MLP を用いた。

2.3.1 PS-MLP のアルゴリズム

学習は図 2.3 の手順で行われる。

1. 全ての結合荷重を初期化し、全てのデータのラベルを `unsatisfied` とする。
2. 全てのデータが収束する (`satisfied` とラベル付けされる) まで 3 から 7 の処理を繰り返す。
3. 未収束 (`unsatisfied` とラベル付けされている) データから 1 つランダムに選択し、計算を行う。
4. 回路網の出力が教師信号と異なる場合は 5 以下の処理を行い、同じであった場合はそのデータのラベルを `satisfied` とし、3 へ。
5. 全ての素子に確率を割り当て、その確率に従って結合荷重を更新するかどうかを決定する。
6. 荷重結合を更新する場合はパーセプトロン更新手続きに従って更新する。
7. 更新を行っていてもいなくても、全てのデータを `unsatisfied` とラベル付けし、3 へ。

図 2.3: PS-MLP アルゴリズム

S-MLP と異なる点は、結合荷重更新時にそれぞれの素子について、その素子への荷重更新を行うか否かを割り当てた確率によって決定する点である。これにより、各素子は極めて独立に動作する。また、更新確率が 1 であるような素子があってはならない。このアルゴリズムも大域的な収束性を持つ。

2.3.2 素子選択確率

PS-MLP では、`effectual` な更新をもつ素子に割り当てる確率は

$$\exp\left(-\left(\frac{\mathbf{w} \cdot \mathbf{x}}{\|\mathbf{w}\| \cdot \|\mathbf{x}\|}\right)(kn)\right) \quad (2.2)$$

を用いる。ここで、 n はこの素子への入力数であり、 n が偶数のときは $k = 1.6$ 、 n が奇数のときは $k = 1.7$ とする。また、 $n = 2, 3$ の時は、 $kn = 1.5, 2.5$ とする。

この確率は試行錯誤の上得られたものであり、最も早く収束させるものであるかどうかは明らかになっていない。櫻井は、どのような確率分布を与えれば最も高速に収束するのかを明らかにすることを今後の課題と挙げている。

2.4 予備実験

このアルゴリズムが大域的な収束性を有し、また、高速かどうかを確認するために、予備実験として櫻井の追実験を行った。 n -ビットパリティ関数を n 個の隠れ素子を持った 3 層神経回路網で実現するようその結合荷重を学習で獲得させるという、パリティ問題を扱った。追実験で用いたアルゴリズムは PS-MLP であり、比較として通常の BP 法でも行った。

その結果、PS-MLP は全ての試行で収束し、データ提示回数も櫻井の実験と同程度であった。また、比較として行った BP 法では収束しない試行もあった。

第3章 シンプルリカレントネットワークへの適用

この章では、SRN について述べたのち、PS-MLP を SRN へ拡張したときのアルゴリズムについて述べる。

3.1 SRN

SRN は、リカレントニューラルネットワーク (recurrent neural network: 以下 RNN と略す) のもっとも単純なものの 1 つであり、Elman [3] により提案された。SRN は、通常のフィード・フォワード型のネットワークにおいて、1 時刻前の隠れ層 (hidden units) の活性パターンをそのまま現在の入力文脈層 (context units) にコピーして利用する (図 3.1)。

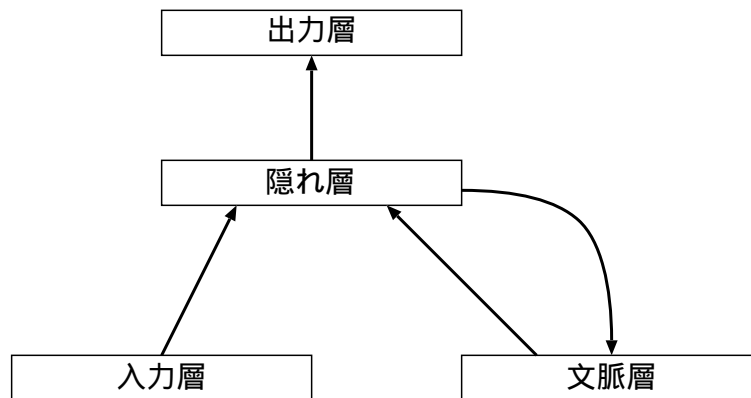


図 3.1: SRN の基本形

文脈層には、1 時刻前の隠れ層の出力がそのままコピーされる。
これにより SRN は、時系列パターンを学習することができる。

学習アルゴリズムには BP 法を利用することができる。過去の情報を持つ文脈層を入力

に加えることにより、SRN は時系列パターンを学習することが出来る。Elman はこのネットワークに、XOR の時系列入力に対して次のビットを予測させるタスクを BP 法を用いて学習させた。これはある種の有限オートマトンを学習させることに相当する。

3.2 SRN への適用

本研究では SRN に PS-MLP を適用し、その能力を測る。SRN に適用したときのアルゴリズムは図 3.2 のようになる。

1. 全てのデータのラベルを `unsatisfied` とし、結合荷重、文脈層に初期値を与える
2. 以下の処理を全てのデータのラベルが `satisfied` となるまで繰り返す
3. ラベルが `unsatisfied` のデータからランダムに一つ選択する
4. 文の最期まで以下の処理を繰り返す、そのデータのラベルを `satisfied` にする
5. 計算を行う
6. ネットワークの出力が教師信号と異なる場合は以下の処理を繰り返す。同じ場合は中間層の出力を文脈層にコピーして、次の入力を入れる。
7. 図 2.3 の 5～6 と同じ処理で結合荷重更新を行う
8. 文脈層をクリアする（全て 1 にする）
9. 文の最初にもどり、5 へ

図 3.2: PS-MLP を SRN に適用したときのアルゴリズム

出力層の素子数が 2 個以上の場合は、それぞれの素子の出力に対して教師信号との比較を行い、異なる出力の場合はその素子への荷重更新を行う。同じ場合は次の素子の出力を評価する。そして、全ての素子の評価が終了し、荷重更新が 1 度も行われなければ中間層の出力値を文脈層にコピーする。荷重更新が行われた場合は 8、9 の処理を行う。

荷重更新を行った後に文脈層のクリアを行うのは、文脈層に過去の間違った出力があると、正しい学習ができないと考えたためである。

本研究では、この学習アルゴリズムを実験で用いた。

第4章 実験と結果

正規文法的一种である富田文法により生成される言語の学習実験を行い、オートマトンの抽出を行った。その研究手法と結果について述べる。

4.1 実験環境

数値実験は Sun Fire 280R(UltraSPARC3 750MHz)、gcc コンパイラを用いて行なった。

4.2 富田文法

富田文法は、富田 [9] によって提案された、正規文法的一种である。この富田文法は正規言語の学習において、ベンチワ - ク的に用いられる典型的な問題である。従って、本研究でもこの富田文法を用いて、学習を行なう。そして、PS-MLP を適用した SRN がどのような FSA を学習するかについて検討する。富田文法には 7 つの文法があるが、本研究では第 3 文法、第 4 文法、第 6 文法、第 7 文法の 4 つを用いた。それぞれの文法についての詳しい説明は表 4.1 に、状態図は図 4.1 に示す。ただし、実験では 0 ではなく -1 としている。ここでは見やすさの便宜上 0 と表現している。

表 4.1: 富田文法

文法	説明	最小状態数
第 3 文法	奇数個の 1 の後に奇数個の 0 が存在しない文字列集合	5
第 4 文法	三つ以上の 0 が連続しない文字列集合	4
第 6 文法	1 の個数と 0 の個数の差が 3 の倍数になる文字列集合	3
第 7 文法	$0^*1^*0^*1^*$	5

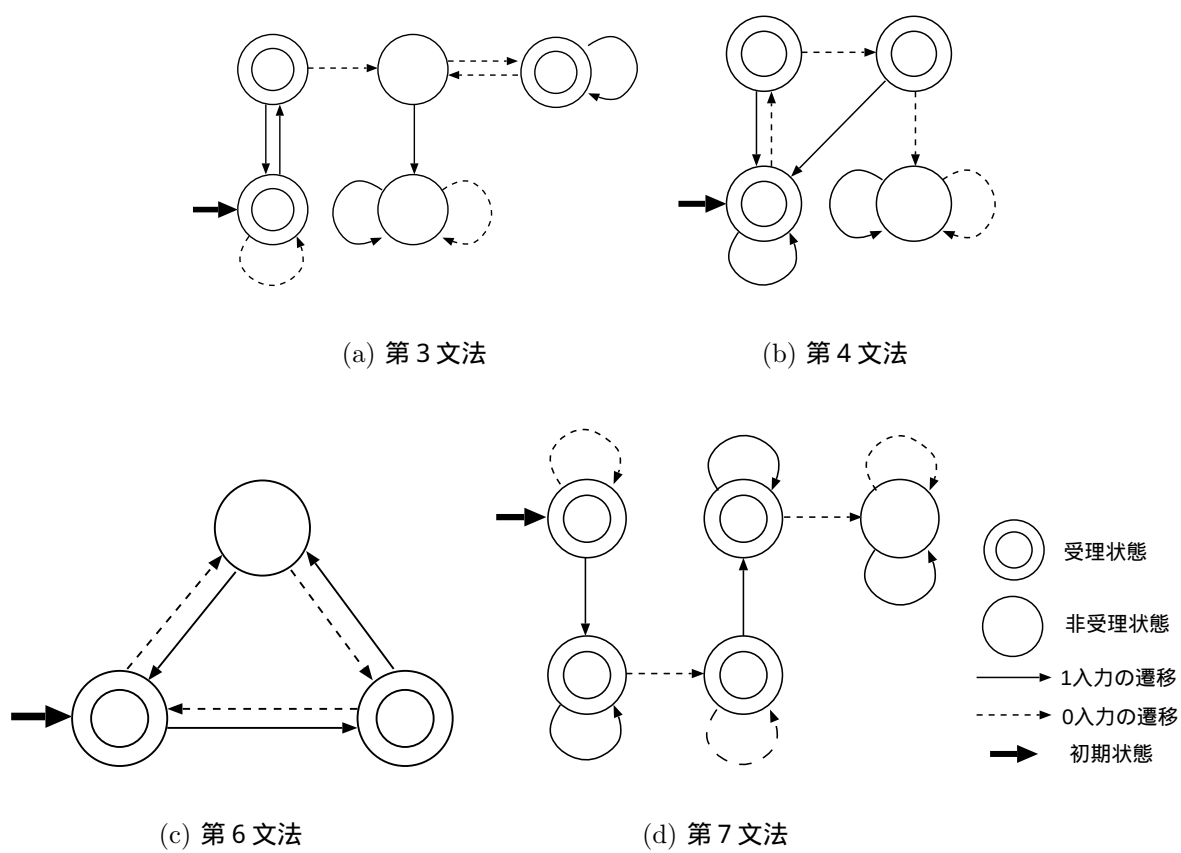


図 4.1: 富田文法:各文法の状態図
それぞれの文法における状態図である。

4.3 実験手法

4.3.1 初期値設定

結合荷重の初期値は $-2.0 \sim 2.0$ の整数値をランダムに設定した。また、文脈層の各素子の初期値は全て 1.0 とした。今後、文脈層のクリアするとは全て 1.0 にすることを意味する。

4.3.2 入力と出力の表現

学習は現在の入力に対して、その入力値が受理状態にあるか非受理状態にあるかを教師信号 (表 4.2) として与えた。すなわち、(実際には有限個の例の提示ではあるが) 極限としては完全提示を行っており、通常の言語学習の枠組みとは異なる。このデータを用いて、図 3.2 のアルゴリズムに従い正しい解が出力されるまで荷重更新を繰り返すという実験を行った。また、100 万回データを提示しても収束しない場合は学習失敗とみなした。

表 4.2: 入力と教師信号の表現

現在の入力	教師信号
受理状態	1
非受理状態	-1

扱う文法と教師信号の例を表 4.3 ~ 表 4.6 に示す。

4.3.3 実験方法

本研究では、2 種類の富田文法の学習をおこなった。一つはランダム生成したデータを用い、100 回の試行で何度学習が収束するかを測った。二つめは正例を網羅したデータを用いて 1 度だけ試行を行い、その隠れ層の状態から有限オートマトンを抽出を試みた。また、それぞれの実験を隠れ層の素子数を変えて行った。

表 4.3: 第 3 文法

入力	1	-1	-1	1	-1	-1
教師信号	1	-1	1	1	-1	1

表 4.4: 第 4 文法

入力	1	-1	-1	1	-1	-1	-1
教師信号	1	1	1	1	1	1	-1

表 4.5: 第 6 文法

入力	1	-1	1	1	1	-1
教師信号	-1	1	-1	-1	1	-1

表 4.6: 第 7 文法

入力	1	-1	1	1	-1	-1
教師信号	1	1	1	1	-1	-1

4.4 結果

4.4.1 ランダムデータ

1 と -1 からなる系列をランダムに生成し、それを手本となる有限オートマトンに入れ、教師信号を作成した。データの個数は第 3 文法、第 4 文法、第 7 文法は 40 個で行い、第 6 文法のみ 20 個で行った。また、文の長さは第 3 文法、第 4 文法、第 7 文法は 3 ~ 6 個、第 6 文法は 3 ~ 20 個で行った。それぞれの文法で、データ提示回数と収束時間を測り、平均、最大、最小を求めた。結果を表 4.7 ~ 表 4.10 にまとめた。

実験の結果、全ての試行で学習が収束した。収束速度は隠れ層の素子数が多い方が早かった。

表 4.7: 第 3 文法

中間層の素子数	3	4	5
平均データ提示回数	7877	4892	3836
最大データ提示回数	42449	30016	36645
最小データ提示回数	219	69	62
平均収束時間 (秒)	1.92	3.16	1.01
正例/負例	30/10	30/10	30/10

全ての試行において成功。
 しかし、隠れ層の数が少ないとデータ提示数が多かった。
 これは自由度が小さいと学習が収束しにくいと判断される。

表 4.8: 第 4 文法

中間層の素子数	2	3	4
平均データ提示回数	44816	1571	1711
最大データ提示回数	601433	8020	8718
最小データ提示回数	93	42	45
平均収束時間 (秒)	39.04	0.64	0.64
正例/負例	31/9	33/7	35/5

全試行で成功。
 隠れ層の素子数が 2 個の時は非常に時間がかかった。
 これは、この文法の最小状態数が 4 であり、
 学習を終了するには常に、最小の状態数でなければならないからである。

表 4.9: 第 6 文法

中間層の素子数	3	4	5
平均データ提示回数	3006	1756	28
最大データ提示回数	24544	1756	10189
最小データ提示回数	54	36	30
平均収束時間 (秒)	1.92	3.16	1.01
正例/負例	5/11	10/10	5/15

全試行で成功。
やはり、最小の状態数に近い表現力しか持たない場合は
学習が収束するまでに時間がかかる。

表 4.10: 第 7 文法

中間層の素子数	3	4	5
平均データ提示回数	15669	7948	78131
最大データ提示回数	118285	38069	483733
最小データ提示回数	94	69	102
平均収束時間 (秒)	11.52	1.92	10.24
正例/負例	35/5	34/6	35/5

全試行で成功。
全体的に他の文法より収束にかかる時間が長かった。

4.4.2 網羅データ

文の長さ3から5(第7文法のみ6)までの全てのパターンを網羅したデータで試行を10回行い、有限オートマトンを抽出した。得られた有限オートマトンから状態数を数え、平均と最小状態数を表4.11～表4.14に示す。また、各文法で抽出できた最小状態数の有限オートマトンの状態図を図4.2～図4.5に示す。この結果、第3文法、第4文法、第7文法において最小状態数のオートマトンが得られた。

表 4.11: 第3文法

中間層の素子数	3	4	5
平均状態数	6.3	7.6	8.0
最小状態数	5	5	6
正例/負例	43/13	43/13	43/13

表 4.12: 第4文法

中間層の素子数	3	4	5
平均状態数	5	6.7	7.4
最小状態数	4	4	4
正例/負例	45/11	45/11	45/11

表 4.13: 第6文法

中間層の素子数	3	4	5
平均状態数	4.3	6.2	5.3
最小状態数	4	4	4
正例/負例	18/38	18/38	18/38

表 4.14: 第7文法

中間層の素子数	3	4	5
平均状態数	5.7	7.2	7.5
最小状態数	5	5	6
正例/負例	80/30	80/30	80/30

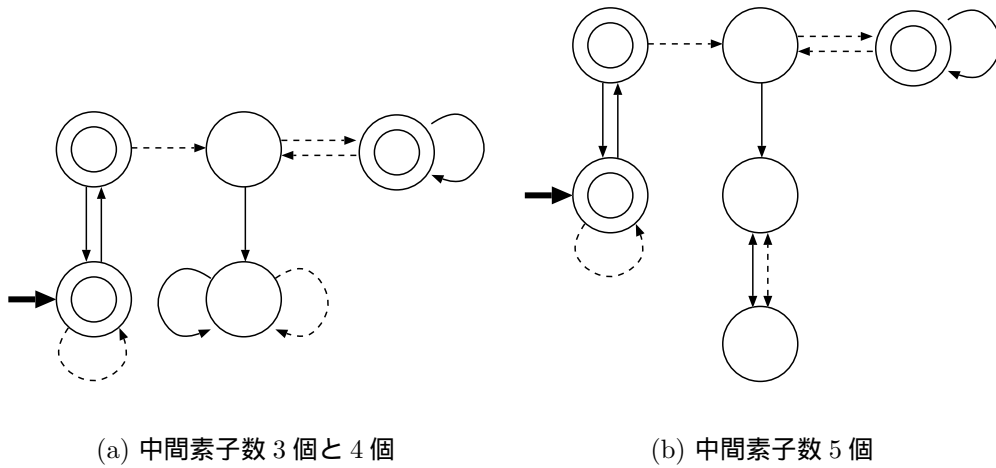


図 4.2: 第 3 文法

- (a) は最小状態数であったときの抽出できた FSA、
 (b) も第 3 文法を生成する正しい FSA。

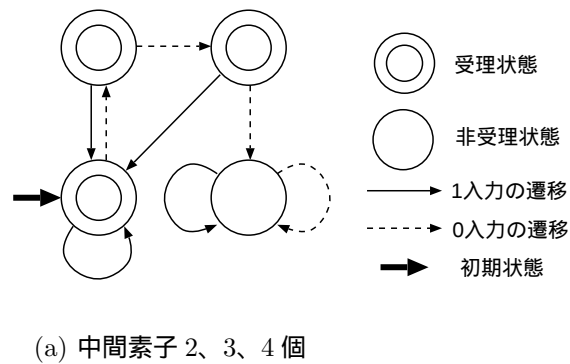
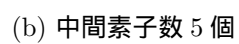
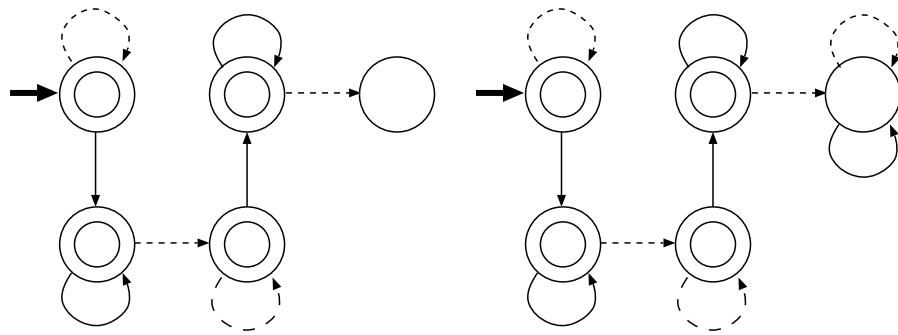


図 4.3: 第 4 文法

全てにおいて、最小状態数の FSA が得られた。
 しかし、素子数が増えると平均状態数が増えた。
 素子数 2 個で常に正しい FSA を抽出できた。

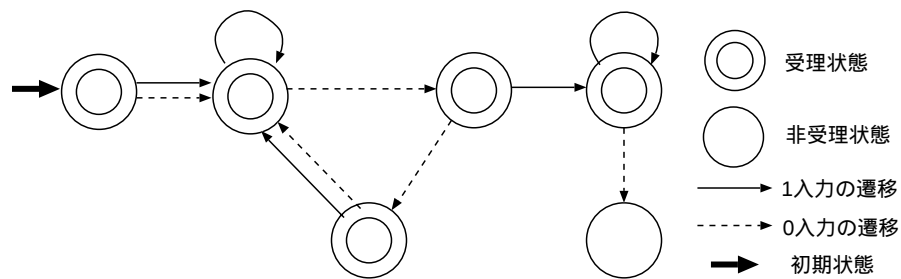


最小状態数の FSA を得ることはできなかった。
しかし、比較的状态数が少ない FSA を得ることができた。



(a) 中間素子数 3 個

(b) 中間素子数 4 個



(c) 中間素子数 5 個

図 4.5: 第 7 文法

最小状態数の FSA を得ることはできた。

(c) も正しい文を生成する FSA である。

第5章 考察

5.1 実験結果まとめ

本研究では、PS-MLP を適用した SRN が、実際に FSA を学習できるかを明確にするために、正規言語の一種である富田文法を用いて実験を行なった。ここではまず、実験結果をまとめ、考察を行なう。

試行数は少ないが、その全てにおいて成功したことから、PS-MLP を適用した SRN は FSA を学習することができると考えられる。実際に隠れ層の状態から学習された FSA の抽出を試みたところ、最小状態数の FSA が第 3、4、7 文法で得られた。また最小状態数ではないが、第 6 文法も、正しい文を出力できる FSA を学習していた。以上のことから、S-MLP を用いた SRN は、富田文法の各文を生成するような FSA を学習することができたといえる。

また、第 7 文法をのぞいた他の文法では、中間層の素子数が多い程、収束速度が早かった。これは素子数を増やすことによって、自由度が高くなったため、表現力が増したためである。しかし、自由度が高くなったにも関わらず、比較的状态数の少ない FSA が得られた (隠れ層の素子数 n 個の時、 2^n 個の状态数の FSA ができる可能性がある)。

一方、収束速度に関しては高速とはいえないかもしれない。例えば、第 4 文法の場合、中間層の素子数が 2 個の場合は平均収束時間が 40 秒近くかかっている。データの提示回数に注目しても、必ずしも小さい提示回数で終るとは限らず、場合によっては 10 万回以上提示しなければ収束しないこともあった。これは S-MLP が全数検索に近いことを行なっているからだと考えられる。

抽出された FSA については、今回行なった実験では最小状態数のものが得られた試行があった。しかし、文の長さが短いために網羅的行なったとはいえず常に正しい FSA を学習することはできなかった。

5.2 今後の課題

実験結果より、S-MLP が SRN に適用でき、その SRN は実際に FSA を学習することがわかった。しかし、課題は残る。

本研究ではデータを網羅的に与えて学習を行なった。しかし、実際にはもっと少ないデータ数でも収束する可能性がある。また、負例がいくつあれば収束するのかということも明確にする必要があると考える。つまり、どのようなデータを与えてやれば、完全な FSA を得ることができるのかを明らかにする必要がある。さらに、短い文で学習したネットワークに、学習させていない長い文を入力したときに、正しい出力をするかを調べなければならない。

また、教示の仕方について、本研究での教示方法は現在の入力を受理状態にあるか非受理状態にあるかを教師信号として与えていた。FSA の教示方法としてはこの方法でも間違いはないが、通常の言語学習の枠組みとしてよく用いられる、統語的に正しい文の次単語を予測する枠組みや、各文が文か非文かを教示する枠組みとはことなる。今後はこのような言語学習的な教示方法でも学習が収束するかを調べ、文法学習の可能性を明らかにする必要がある。

また、今回扱った富田文法は非常に簡単な FSA で表現される。もう少し状態数の多い正規文法を学習させたときに、どのような FSA が抽出されるのか調べる必要もあるだろう。また、学習を短い文で行ない、長い文でテストを行なってみる必要もある。

最期に、S-MLP の課題として櫻井 [1] も述べているように、各素子に割り当てる確率をどのようにすればよいかということがある。どのような確率分布を与えれば高速に安定した速度で収束するかを明らかにしたい。

第6章 まとめ

本研究では、線形閾値素子を用いた学習アルゴリズム、PS-MLP を SRN に適用した。さらに、その検証として富田文法の学習を行ない、第 3、4、7 文法において最小状態数の FSA を獲得することにも成功した。

実験により、PS-MLP を SRN に適用可能であることが明らかになった。したがって、S-MLP も同様に適用することができると考えられる。本研究が、今後の記号処理の分野において多少なりとも寄与することを願う。

謝辞

本研究を行なうにあたり、熱心に御指導して頂きました櫻井彰人教授、林幸雄助教授、荒木修助手に心から厚くお礼申し上げます。本研究を進めるにあたり、助言をして頂いた先生方に感謝いたします。また、様々な面で協力して頂いた、同講座のみなさまに感謝いたします。

参考文献

- [1] Akito Sakurai. A fast and convergent stochastic mlp learning algorithm. *International Journal of Neural Systems*, 2001.
- [2] McClelland, J.L. Cleeremans, A., Servan-Schreiber, D. Finite state automata and simple recurrent networks, 1989. *Neural Computation* 1, 372-381.
- [3] J.L. Elman. Finding structure in time, 1990. *Cognitive Science*, 14; 179-211.
- [4] C.L. Giles and C.B. Miller. Learning and extracting finite state automata with second-order recurrent neural networks. *Neural Computation*, Vol. 4, No. 3, pp. 393-405, jun 1992.
- [5] Golea, M. On the complexity of rule-extraction from neural networks and network-querying, 1996. Tech. Rep., Department of Systems Engineering, Australian National University.
- [6] 原田哲治. 再帰型回路網による文法の獲得. Master's thesis, 北陸先端科学技術大学院大学, March 2001.
- [7] Lee Giles Christian Omlin. Symbolic knowledge representation in recurrent neural networks: Insights from theoretical models of computation.
- [8] 田中賢, 熊沢逸夫, 小川英光. 再帰型高次結合ニューラルネットワークによる正規言語の学習, 1996. 電気情報通信学会論文誌, D-2 Vol. J79-D-2 No. 5, 899-907.
- [9] M. Tomita. Dynamic construction of finite-state automata from examples using hill-climbing. *Proceedings of the Fourth Annual Cognitive Science Conference*, pp. 105-108, 1982.
- [10] 小沢誠一 馬場則夫. ニューラルネットの基礎と応用. 共立出版, 1996.

[11] 宮野悟有川節夫. オ - トマトンと計算可能性. 培風館, 1996.