<u>卒業論文</u>

<u>題目</u>

リカレントニューラルネットワークの学習と力学構造

<u>指導教授</u>

所 真理雄 教授

慶應義塾大学

理工学部 電気工学科

1996 年度

69301037

伊藤 真人

論　文　要　旨

学科	電気工学	学籍番号	69301037	氏名	伊藤 真人

(論文題目)

リカレントニューラルネットワークの学習と力学構造

リカレントニューラルネットワークは，フィードフォワードネットワークにはない時系列情報を扱う能力を持っている．しかし，リカレントニューラルネットワークを，単にニューラルネットワークとして眺めるだけでは，その本質的な計算能力を明らかにすることはできない．本論文では，時間情報を記述することのできる力学系の理論を導入することにより，リカレントニューラルネットワークを力学系の枠組の 1 つとして捉え，その力学構造よりリカレントニューラルネットワークの学習について解析する．

本論文では，リカレントニューラルネットワークの学習に関する実験を行う．まず，時系列情報として単純なリミットサイクルのパターン列を学習させると，リカレントニューラルネットワークの内部自体に力学構造としてリミットサイクルのアトラクタが形成されていく．一方，時系列情報として有限状態機械で表現されるパターン列を学習させると，ネットワーク内部に形成されていく力学構造はフラクタルなものとなる．この力学構造の違いは，有限状態機械の学習の際にリミットサイクルの学習に比べて保持しなければならない時系列情報が複雑であることに起因する．次に，こうした学習過程を観察した上で，このフラクタル構造の中に，時系列情報が埋め込まれていることをその記号力学系を解析することにより示す．そして，フラクタルの生成アルゴリズムである Barnsley の *Iterated Function Systems* とリカレントニューラルネットワークの関係について議論し，リカレントニューラルネットワークの本質的な計算能力を明らかにする．

慶 應 義 塾 大 学 理 工 学 部

◎内容の要旨は約 600 字

Summary of Bachelor's Thesis

Course	Student No.	SURNAME, Firstname
Electrical Engineering	69301037	ITO Masato

Title

The Dynamical Structure of Recurrent Neural Networks

目 次

図 目 次

表 目 次

第 1 章

序論

1.1 はじめに

人間社会および生態系は，無数の自律的な知的主体 (エージェント) が複雑な相互作用を絶えず行っている結果として形成されている系である．これらの系は外部との明確な境界を持たず，エネルギーの外部から内部への流入あるいは内部から外部への流出が絶えず起こる開放系である．この無限に広がる開放系に対して，系を構成するエージェントは有限の能力しか持たない．個々のエージェントが系全体の状態を把握することは不可能であり，常に最適な行動を取ることはできない．しかしエージェントはその存在を維持していくために，致命的な過ちを犯すことなく活動を続けている．有限の経験から，エージェントはその内部に何らかの形で系の適切なモデルを構築し，それに基づいて行動を決定しているのである．

計算機上にこのエージェントを創り出そうとするとき，エージェントが行っている認知について考えなければならない．その際には，まずモデルが必要となる．人工知能の分野では，知識を完全に記号で表現し，その記号を操作するという手法で認知のモデルを構築してきた．つまり，エージェントはその内部に置かれた環境のモデルを記号を用いて表現しているのである．しかし人間および生物が認知の底辺において記号的処理を行っているとは考えにくい．もちろん，認知過程の上位レベルでは記号処理的な側面を持つものもあるが，そこでも，その記号自体はどのように表現されているのかといった問題が残されている．

一方，知性はカオスを含む力学系によるものであるという解釈の下に，認知のモデルを力学系の枠組の上に構築しようとする試みがなされている [1, 2, 3]．エージェントが行う認知を力学系の時間発展の上に考えようとしているのである．エージェントは，置かれた環境を 1 つの力学系として捉えて，それに自分が持つ内部力学系を同調させることにより環境のモデルを表現しているのである．最近の脳研究において，脳の情報処理にカオスが深く関わっているという見解 [4, 5] が示されており，力学系によるモデルは認知を理解する上で非常に有力なモデルと考えられる．

本研究では，力学系としてリカレントニューラルネットワークを取り挙げる．そのリカレントニューラルネットワークの学習の実験を通して，認知の問題について考察していく．

リカレントニューラルネットワークと力学系

生物のなかには，同じ刺激に対して常に同じ反応を示すものもいる．しかし，われわれ人間をはじめとする多くの生物は，過去の経験の有無によって，同様の状況の下でも異なる行動を取り得る．生物は過去の経験を何らかの形で記憶し，それに基づいて行動を決定するのである．

さて，人工ニューラルネットワークは，生物の神経系の特徴的な機能に着目してそのモデル化を行ったものである．その人工ニューラルネットワークの中でも，リカレントニューラルネットワークは，時系列情報を扱うためにネットワークにフィードバック結合を持たせたモデルである．

リカレントニューラルネットワークは，その内部に過去の入力の履歴を何らかの形で保持する機構を有する．つまり，リカレントニューラルネットワークの出力は現時点での入力のみでなく，過去の入力の履歴により決定される．このリカレントニューラルネットワークにおける情報処理を，単にニューラルネットワークの振舞いとして外から眺めるだけでは，その本質的な計算能力を明らかにすることはできない．

リカレントニューラルネットワークの振舞いは，フィードバック結合を持つネットワークの中に保持された内部状態を，ユニット間の結合の重みで決定される非線形写像により，更新していく過程である．これは力学系の系が (1.1) 式のような差分方程式により時間発展していくことと同様であると考えられる．

$$x_{t+1} = f(x_t) \tag{1.1}$$

つまり，リカレントニューラルネットワークを力学系の枠組として捉えることができる．したがって，その力学構造を解析することにより，リカレントニューラルネットワークの情報処理の機構を明らかにすることができると考えられる．

1.2 本研究の目的

本研究の第一の目的は，力学系の視点からリカレントニューラルネットワークの学習を解析し，リカレントニューラルネットワークにおける情報処理の機構を明らかにすることである．そして，そこに見られる力学構造が，リカレントニューラルネットワークの計算能力を決定づけていることを示す．

また本研究の別の目的は，力学系による認知モデルの構築において新たな可能性を与えるとともに，その認知モデルについて考察することである．

1.3 本論文の構成

本論文の構成は以下の通りである．

まず，第 2 章では，本研究の背景としてリカレントニューラルネットワークの研究についていくつか紹介した後に，本研究の大きな柱となる力学系の理論について説明する．第 3 章では，力学系の視点からリカレントニューラルネットワークについて説明する．第 4 章では，実験としてリカレントニューラルネットワークに学習を行わせ，それを力学系の視点から解析する．第 5 章では，リカレントニューラルネットワークの力学系としての情報処理や力学系による認知モデルについて考察し，議論する．最後に第 6 章で結論を述べる．

第 2 章

背景

2.1 リカレントニューラルネットワーク

本節では，本研究において解析対象となるリカレントニューラルネットワークについていくつか関連研究を紹介しながら説明する．

人工ニューラルネットワーク (Artificial Neural Networks) は，生物の神経系の特徴的な機能に着目してモデル化したものである．この人工ニューラルネットワークの歴史は，McCulloch と Pitts による閾値素子のニューロンモデルにまでさかのぼる．その後，Rosenblatt によって提案されたパーセプトロンが学習する機械として注目された．近年では，様々な問題に応用するために，数多くの人工ニューラルネットワークのモデルが提案されている．

リカレントニューラルネットワークは，その人工ニューラルネットワークの中でも時系列情報を扱うことのできるモデルである．

2.1.1 リカレントニューラルネットワークとは

人工ニューラルネットワークの 1 つであるフィードフォワードネットワーク (Feedforward Networks) は，誤差逆伝播 (Back Propagation) アルゴリズム [6] の適用により，パターン認識や未知関数の学習などに幅広く応用されている．しかし，その出力は現在の入力のみから決められ，過去の履歴が考慮されずに現在の入力のみから決められてしまうため，時系列情報を学習・適切に処理することができない．この問題に対して，時系列パターンを空間パターンに変換することにより，時系列情報を扱うことのできるフィードフォワードネットワークのモデルも提案されているが，この方法には考慮する履歴の大きさが固定されてしまうという問題がある．一方，これとは別に，リカレントニューラルネットワークというモデルも提案されている．これは，ネットワークにフィードバックループを持たせ，そこに保持される内部状態に基づいて処理を行うことで時系列情報を扱うことを可能にしたものであり，履歴の大きさが固定されるといった問題もない．

2.1.2 モデル

本項では，リカレントニューラルネットワークのモデルとして，*Jordan Network*，*Elman Network*，*Sequential Cascaded Network* の 3 つを紹介する．

Jordan Network

図 2.1 に Jordan によって提案された Jordan Network を示す．Jordan Network は，出力ユニットからコンテクストユニットの間に 1 対 1 に対応するフィードバック結合があり，出力ユニットの出力が対応するコンテクストユニットへそのまま入力される．また，コンテクスト層の各ユニットには，結合の重みが γ の自己結合があり，これによって過去の履歴を保持することができる．

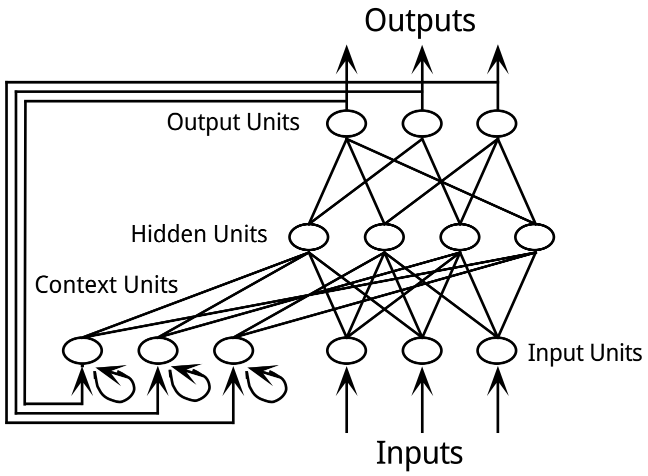


図 2.1: Jordan Network

Elman Network

図 2.2 に Elman によって提案された Elman Network [7] を示す．Jordan Network は出力層からコンテクスト層へのフィードバック結合があるのに対し，Elman Network では隠れ層からコンテクスト層へのフィードバック結合があり，コンテクストユニットに自己結合がない．過去の履歴は隠れ層が保持することになる．なお，フィードバック側の重みが出力側とは別に決定されるので，出力層の制約を受けずに履歴情報を保持することができる．

Sequential Cascaded Network

Jordan Network や Elman Network のユニット間の結合は，1 次元結合である．つまり，あるユニットから次のユニットへのアクティベーションは $\sum w_{ij} o_i$ で算出される．これに対して，ユニット間の結合を多項式結合にしたネットワークである Sequential Cascaded

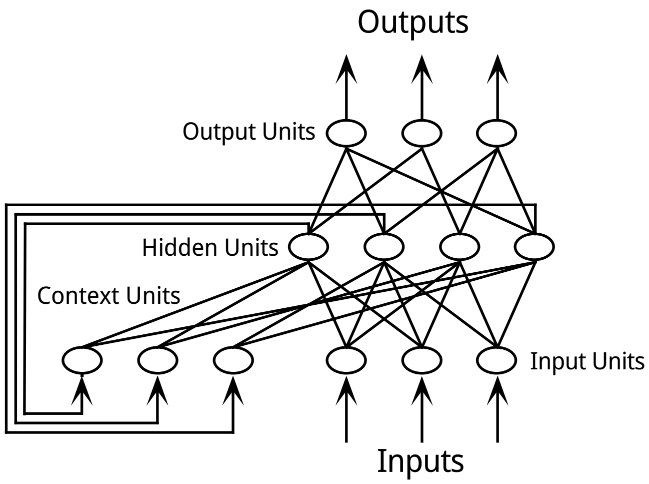


図 2.2: Elman Network

Network (SCN) が Pollack により提案された (図 2.3)．SCN では，あるユニットから次のユニットへのアクティベーションは，$\sum w_{ij} o_i o_j$ で算出される．これにより少ない計算資源でより多くの機能を表現することができる．

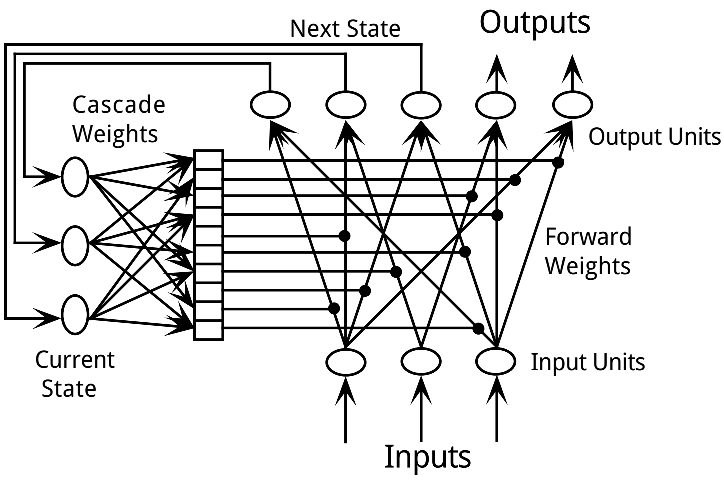


図 2.3: Sequential Cascaded Network

どのモデルも，ネットワークにフィードバックを持たせることにより，過去の入力の履歴を何らかの形で現在の出力に反映させており，本質的な計算能力に違いはない．

2.1.3 学習アルゴリズム

リカレントニューラルネットワークは，時間をまたいだ入出力の列を学習することが可能であるが，その学習アルゴリズムはフィードフォワードネットワークの場合に比べて，複雑なものとなる．

ここでは，リカレントニューラルネットワークの学習アルゴリズムとして典型的な，*Back Propagation Through Time (BPTT)* と *Real Time Recurrent Learning (RTRL)* の 2 つのアルゴリズムについて概説する．

Back Propagation Through Time

Rumelhart らによって提案された Back Propagation Through Time (BPTT) [6] では，多ステップにわたるネットワークの動作を時間軸方向に展開し，多層フィードフォワードネットワークの 1 回の動作に置き換えた上で，通常の誤差逆伝播アルゴリズムを適用して学習を行う．このとき，展開されたネットワークの各層の間の重みが実際には 1 つの結合であることを考慮して重みの修正を行う．BPTT のアルゴリズムの詳細は，Appendix Aにて述べる．

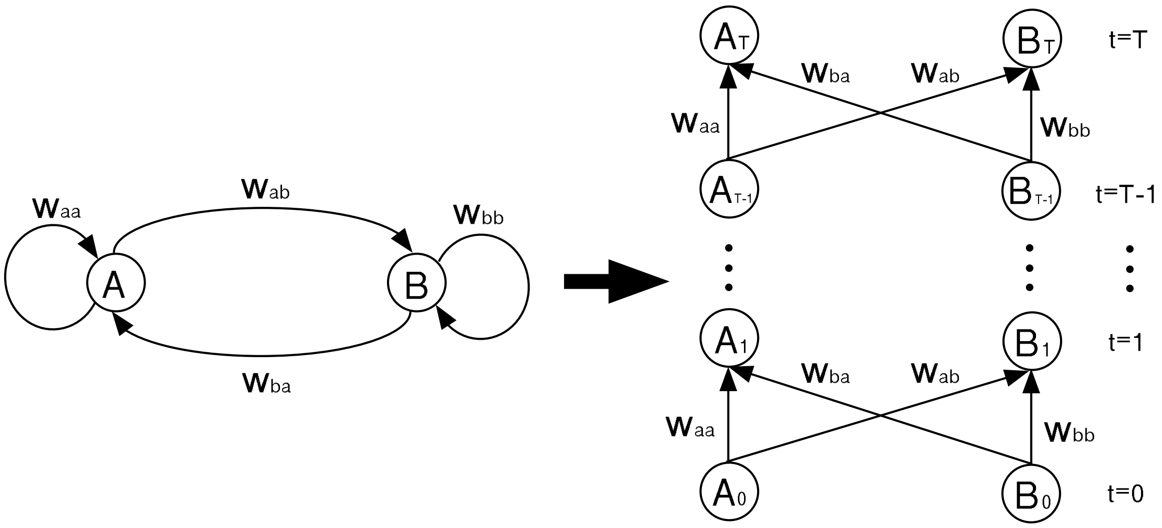


図 2.4: Back Propagation Through Time

全ユニット数を N ，学習時間ステップ数を T とするとき，$O(NT)$ の空間計算量と $O(N^2T)$ の時間計算量を必要とする．このように，BPTT は長時間の学習には適さないため，時間ステップ数をある適切な間隔で区切って使用している場合が多い．

Real Time Recurrent Learning

BPTT が誤差の勾配を計算する際に誤差情報を時間を遡る方向に伝搬させるのに対して，Williams らの提案した Real Time Recurrent Learning (RTRL) [8] は，勾配情報を時間の進む方向に伝搬させるアルゴリズムである．ネットワークの状態変化の方程式を結合の重みによって偏微分することによって，各時刻のユニットの出力の結合の重みに関する偏微分係数を時刻ごとに順に求めていくための式が得られる．この偏微分係数は，1 つの結合の重みが微小に変化したときの各時刻におけるユニットの出力の変化率を表すものである．結合の重みの変化の影響は，(t_0, T) の全期間にわたって累積されて，時刻 T でのユニットの出力の変化が決まる．この計算において，ネットワークの初期状態は一定であり，各時刻の入力も一定であると仮定している．この計算式を用いることによって，ネットワークの動作と並行して偏微分係数を計算していくことができる．したがって，この係数を用いることによって，実際の出力と正解出力の間に誤差が生じた場合には，即座に結合の重みを修正することができる．

問題点としては，全ユニット数を N とすると $O(N^3)$ の空間計算量と $O(N^4)$ の時間計算量を必要とする点が挙げられる．したがって，大規模なネットワークには向かないが，順時間方向に解くことができる実時間性を要求されるものへの適用が可能であるという利点がある．

2.2 力学系

ニューラルネットワークを 1 つの系として捉えた場合，フィードフォワードネットワークでは，系の出力は現在の入力のみから決められる．一方，リカレントネットワークでは，系の出力は現在の入力だけでなく，過去における入力の履歴にも依存して決定される．そこで，リカレントネットワークの時間を越えた振舞いを理解するために，時間情報を記述できる枠組の 1 つとして力学系の理論について述べる．本節では，力学系一般について触れ，次 2.3 節で力学系の中でもリカレントネットワークと関係の深い *Iterated Function Systems* について述べる．

2.2.1 力学系 (Dynamical Systems) とは

自然界の諸現象は，時間の経過に伴い状態が変化していく．これらの現象を表現するために，時間情報を記述できる枠組の 1 つとして，力学系 (Dynamical Systems) がある．ある現象を力学系の枠組で捉えた場合，系はその運動状態を記述する状態変数 (state variable) の組を用いて表現される．この状態変数で張られる空間を相空間 (phase space)

と呼ぶ．系のある状態は各状態変数の値の組で記述されるので，相空間の中では1つの点で表現される．この点の時間経過に伴う振舞いを追跡すると，相空間の中に系固有の幾何学模様が描かれる．これが系の力学構造を表している．

古典力学系の理論では，系の状態は決定論的なものである．ある時刻での状態が特定されると未来のある時点での状態は一意に決定される．あるいは，過去にさかのぼったある時点での状態を正確に知ることができる．つまり，古典力学系の枠組においては初期状態が特定できれば，理論的にはその遠い未来まで正確に予測することができる．しかし，系の時間発展を決定する法則に非線形要素を含む場合，決定論的な系であるにもかかわらず，一見ランダムあるいは確率的で複雑な振舞いを呈することがある．この状態をカオス (chaos) と呼ぶ．一般に非線形力学系ではパラメータの変化によって，安定性などの系の力学的性質が不連続的に変化することがある．このパラメータの変化の前後で，系が位相的に変化する現象を分岐 (bifurcation) と呼ぶ．

これまでに述べてきたことは力学系一般についてであるが，時間および状態空間をそれぞれ連続もしくは離散のものに分類すると4つのタイプに分けることができる．状態空間と時間が共に連続であるような系は微分方程式で表現されるし，状態空間が連続で時間が離散である系は繰り返し写像で表現される．一方，状態空間が離散で時間が連続なものはスピングラス，状態空間と時間が共に離散なものはオートマトンなどがある．

2.2.2 トラジェクトリ，トランジェント，アトラクタ

相空間における系の運動を追跡するとその系の力学的性質を視覚的に見ることができる．この相空間における系の運動の道筋をトラジェクトリ (trajectory) と呼ぶ．ある力学的性質をもつ系のトラジェクトリは，初期値によらず似たトラジェクトリを持つ．この，系に特有のトラジェクトリをその系のアトラクタ (attractor) と呼び，アトラクタで表現される力学的性質を力学構造と呼ぶ．

ただし，初期状態からアトラクタに至るまでのトラジェクトリは初期値により異なる．これをトランジェント (transient) と呼ぶ．系の完全なアトラクタを求めるには，無限の時間を必要とするが，有限時間のトラジェクトリからトランジェントを取り除くことで，近似的にアトラクタを求めることができる．

系の力学構造は，アトラクタで全て表現されつくされているわけではなく，トラジェクトリがアトラクタにたどり着くまでのトランジェントもそれを表現している．しかし，そのトランジェントもアトラクタの性質に支配されているので，系の大域的力学構造は系のアトラクタを見ることにより十分に知ることができる．

アトラクタはその力学的性質からみて，不動点，リミットサイクル，トーラス，スト

レンジアトラクタの 4 つに分類することができる．以下で，それぞれについて説明する．

不動点 (**Fixed Point**)

相空間上で 1 つの点で表されるアトラクタを不動点と呼ぶ．不動点をアトラクタとして持つ系のトラジェクトリは一点へと収束していく．いったん，系の運動がアトラクタに落ちると，その系の運動が止まっているように見える．不動点には，安定なものと不安定なものがある．安定なものは，その近傍の状態を引き込む性質を持つ．一方，不安定なものは，不動点からほんのわずかでも離れると，トラジェクトリがどんどんその点から離れる性質を持つ．このような性質の不動点は，アトラクタとは反対の意味でリペラーと呼ぶ．

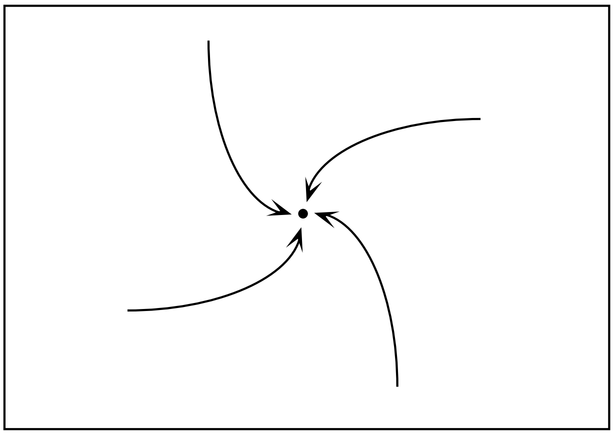


図 2.5: 不動点

リミットサイクル (**Limit Cycle**)

リミットサイクルは，相空間上で状態が離散の場合，複数の点で表され，状態が連続の場合，1 つのループで表される．リミットサイクルをアトラクタとして持つ系のトラジェクトリは周期的に同じ状態を巡回する．

トーラス (**Torus**)

系の全ての状態変数が同一周期を取るときは，系のアトラクタがリミットサイクルになることは直観的に分かる．しかし，いつも全ての状態変数が同一周期を取るとは限らない．状態変数の周期の比が有理数で表される場合は，その最小公倍数が周期となり，系

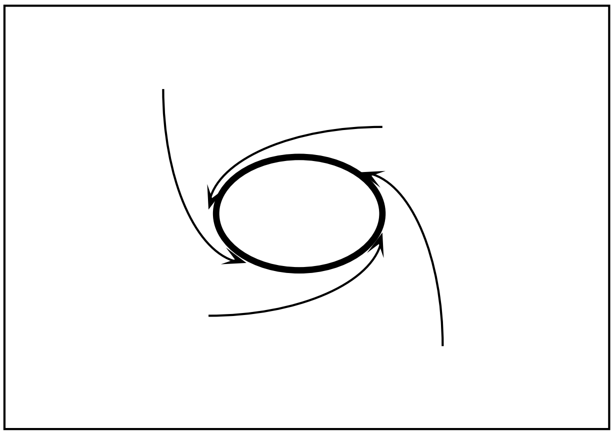


図 2.6: リミットサイクル

のアトラクタはリミットサイクルとなる．しかし，状態変数の周期の比が無理数となる場合は，最小公倍数が計算できない．つまり，周期が存在しない．したがって，系のトラジェクトリは同一の点を 2 度通ることはない．この複雑なトラジェクトリのアトラクタをトーラスと呼ぶ．3 次元トーラスの場合，トラジェクトリはドーナツ状の表面を万べんなくかつ交わることなく覆いつくす．

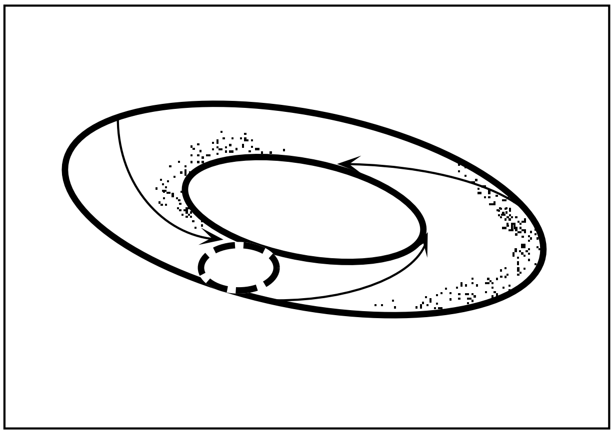


図 2.7: トーラス

ストレンジアトラクタ (Strange Attractor)

ストレンジアトラクタもトーラスと同様に同一の点を 2 度通ることのないトラジェクトリを持つアトラクタである．両者の相違点は，トーラスでは近傍にある 2 点はその後

もごく似通ったトラジェクトリを取るのに対して，ストレンジアトラクタでは近傍にある 2 点はその後全く異なるトラジェクトリを取る．つまり，ストレンジアトラクタは初期値のほんのわずか違いが時間発展とともにどんどん大きくなるような，初期値鋭敏性を持っているのである．したがって，系のアトラクタがストレンジアトラクタである場合には，この系はカオス状態的な振舞いを呈する．

2.2.3　分岐

線形力学系では，系のパラメータを微小変化させてもその力学構造が変化することはない．しかし，非線形力学系においては入力の大きさ・種類によって系の力学構造が変化するばかりでなく，同じ入力であってもパラメータの大きさによって系の力学構造が位相的に変化し得る．このように系のパラメータが微小変化したとき，その前後の系の力学構造が位相的に変化する現象を分岐現象という．分岐とは，『2 つに分かれる』という意味で，本来，非線形システムの不動点がパラメータの微小変化により発生・消滅する現象を示すのに用いられた．しかし，様々な分岐の種類があることから，パラメータの微小変化により，系の力学構造が位相的に変化する現象を総称して，分岐現象と呼んでいる．非線形力学系ではパラメータの微小変化により，不動点が消滅してカオスが発生したり，リミットサイクルが次々に分岐を続けてカオスになったりする．また不動点やリミットサイクルの性質が変化したり，消滅・発生したりする．分岐現象とカオスは密接な関係があり，非線形システムの構造を明らかにするうえで重要なものである．

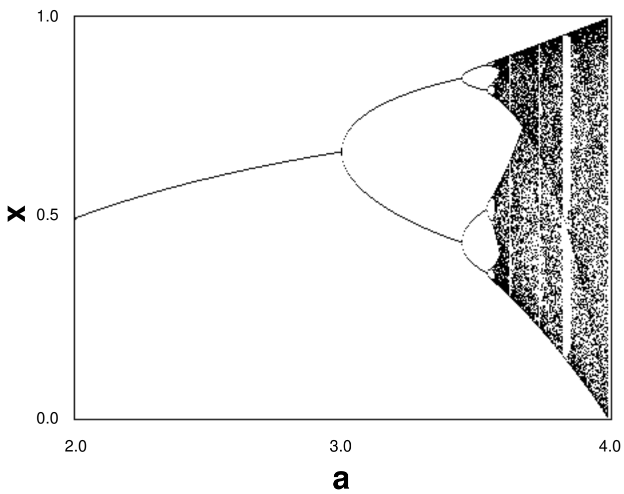


図 2.8: 分岐現象

2.2.4 記号力学系

系の力学構造を調べるときに，まず用いる方法は系のアトラクタを調べることである．系のアトラクタが 2 次元や 3 次元といった比較的低次元の相空間に現れる場合は，そのアトラクタを調べることにより系の力学構造を特定することができる．しかし，系のアトラクタが高次元の相空間に現れる場合，そのアトラクタを視覚的に表現することはもちろん直観的に理解することも難しいため，系の力学構造を特定することは困難である．そこで，トラジェクトリを離散的な記号列に置き換えて，その記号列を新たな力学系として見ることにより，系の構造を理解しようというのが記号力学系の考え方である．この記号は相空間の領域に対する名前であると考えることができる．

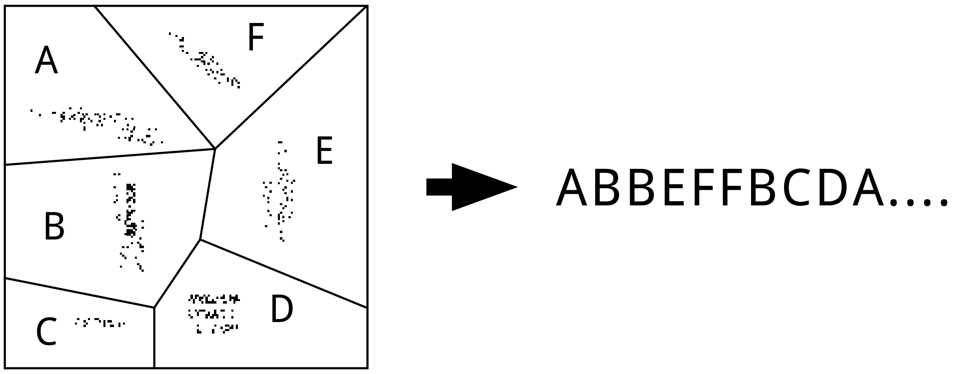


図 2.9: 記号力学系

2.3 フラクタルと Iterated Functions Systems

力学系の中でもリカレントネットワークと関係が深いのは，繰り返し写像で表現される系である．Barnsley は，繰り返し写像の組みを *Iterated Functions Systems* として捉え，フラクタルとの関係を述べている [9]．

2.3.1 フラクタル (Fractal)

数学でよく扱われるのは，三角形，円，正方形，立方体などの図形または点の集合である．しかし，自然界にはある図形の大部分は単純なな規則正しいものではない．例えば，海岸線，山，雲，樹木，さらには肺の血管網など．これらの図形を，数学的に扱うには複雑すぎる．

さて，ここで，次のようなことを考えてみる．ある正三角形の各辺の中点を結び，中心の三角形を取り除き，残った三角形にこれと同じ操作を無限に繰り返す．すると，図 2.10 のような集合が得られる．この集合，シェルピンスキーギャスケットと呼ばれるものである．今世紀初頭に発見され，長い間，きわめて病理的なものであると思われてきた．単純に三角形あるいはその集合として扱うには複雑すぎる．それでもなお，実は極めて簡単な規則から生成されるている．

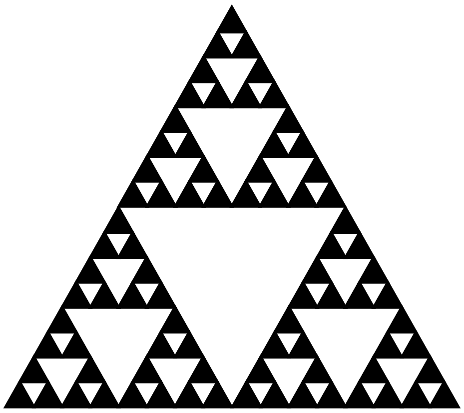


図 2.10: シェルピンスキーギャスケット

1974 年，Mandelbrot は図 2.10 のような集合をフラクタルと名付けた．この集合のハウスドルフ次元 (Hausdorff dimension)[1]は $\log 3/\log 2$，(約 1.58) である．ここで，ハウスドルフ次元を複雑さを示す指標として見ると，正方形は 2，線分は 1 となるため，この集合の次元はその間の存在すると言える．一方，これを位相次元で見ると，正方形は 2，線分は 1 で，シェルピンスキーギャスケットは 1 となる．そこで，Mandelbrot は同一図形におけるハウスドルフ次元と位相次元の差に着目して『集合のハウスドルフ次元がその位相次元より大きいものをフラクタルという』というようにフラクタルを定義した．

定義自体は直観的には分かりにくいが，フラクタルに共通な性質は自己相似性 (self-similarity) である．この点から見ると，シェルピンスキーギャスケットと海岸線を同様に扱うことができる．

2.3.2 Iterated Functions Systems

Iterated Functions Systems(IFS)[9] は，フラクタル生成アルゴリズムの 1 つである．IFS は幾つかの縮小写像 (contraction map) から構成される．例えば， 以下の 3 つの式

[1]非整数次元を表す次元としてよく用いられる．

で表されるような 3 つの写像を考える．

$$W_1(x) = \left(\frac{x}{2} + \frac{1}{4}, \frac{y}{2} + \frac{\sqrt{x}}{4}\right) \tag{2.1}$$
$$W_2(x) = \left(\frac{x}{2} + \frac{1}{2}, \frac{y}{2}\right) \tag{2.2}$$
$$W_3(x) = \left(\frac{x}{2}, \frac{y}{2}\right) \tag{2.3}$$

個々の写像は，縮小写像になっている．したがって，この 3 つのうち 1 つだけを用いて繰り返し写像を行うと，集合はただ 1 つの不動点となる (図 2.11(a))．しかし，この 3 つの縮小写像を全て用いて映し出される集合はコンパクトな集合となり，これは元の自己相似的な構造を持つフラクタルとなる (図 2.11(b))．

Barnsley の提案した IFS はそれ自身が力学系である．したがって，IFS から生成されるフラクタル図形は IFS のアトラクタであり，そのアトラクタの構造は，個々の写像のトランジェントに依存した構造となる．

2.3.3 カオスゲーム (Chaos Game)

理論的には，IFS によりフラクタルを生成することができるが，多大な時間を要するため現実的ではない．そこで，Barnsley はカオスゲーム (Chaos Game) と呼ばれる手法を考案した [9]．カオスゲームは，IFS から生成されるフラクタルが力学系のアトラクタであることを利用している．

シェルピンスキーギャスケットの例で考えてみる．シェルピンスキーギャスケットを生成する IFS は，(2.1) 式，(2.2) 式，(2.3) 式の 3 つの写像から成る．したがって，集合の各点について n 回の写像を行うとすると，フラクタルを生成するには 3 つの写像の全組合せ，3^n 回の計算を行わなければならない．一方，カオスゲームでは集合の各点について n 回の写像を行う際，3 つの写像からランダムに 1 つを選んで写像を行うことで，近似的にフラクタルを生成することができる．これは，IFS から生成されるフラクタルが力学系のアトラクタであるため，初期値によらず同じ集合 (不変集合) が生成されるのである．

2.4 まとめ

本章では，本研究において解析対象となるリカレントニューラルネットワークについて説明した．また，本研究でリカレントニューラルネットワークを解析する際に用いる

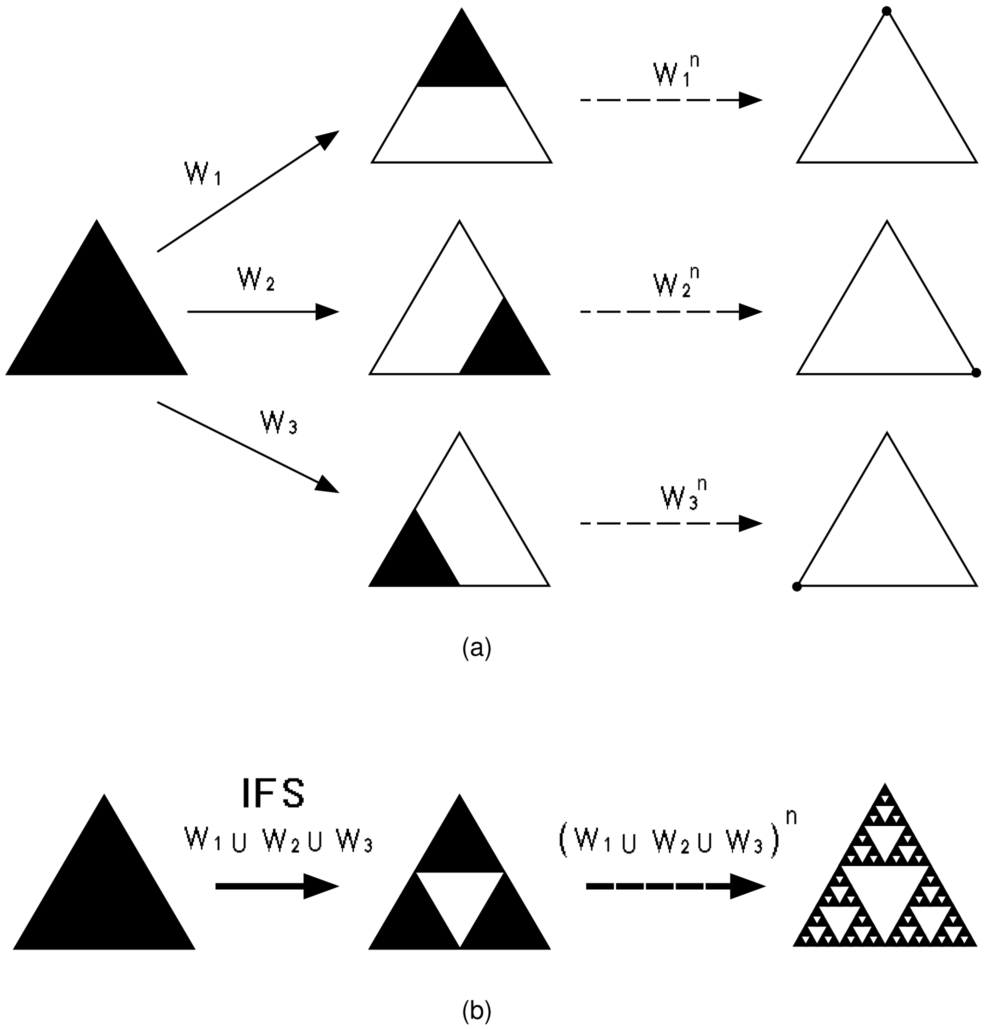


図 2.11: Iterated Functions Systems

力学系の理論について説明し，さらにリカレントニューラルネットワークに関係の深い Iterated Functions Systems について説明した．

第 3 章

リカレントニューラルネットワークと力学系

3.1　リカレントニューラルネットワークと力学系

リカレントニューラルネットワークに関する多くの研究は，学習効率の良いアルゴリズムやより複雑な問題の学習などに重点がおかれている．これらの研究は工学的応用を考える上では非常に重要であるが，その振舞い自体に言及する研究事例は少ないため，その情報処理の機構については未だ明らかにされているとは言い難い．そこで，本研究ではリカレントニューラルネットワークを力学系として捉え，その振舞いを解析した上で．その情報処理の機構を明らかにする．

本章では，リカレントニューラルネットワークを力学系として捉えるにあたり，力学系の中でもリカレントニューラルネットワークと関係の深い離散力学系と非線形力学系について説明する．最後に，リカレントニューラルネットワークを離散力学系の時間発展方程式に重ね合わる．

3.2　数学モデル

リカレントニューラルネットワークの離散時間モデルは以下の 2 式で表現できる．ネットワークの動作つまり状態 S_t の更新は (3.1) 式で，出力 O_t の決定は (3.2) 式で表現できる．

$$S_{t+1} = F_s(S_t, W) \tag{3.1}$$

$$O_{t+1} = F_o(S_t, W) \tag{3.2}$$

ここで W はユニット間の結合の重みのマトリックス，F_s，F_o はシグモイド関数である．

3.3　離散力学系

力学系の中でも，離散時間を用いる系を離散力学系と呼ぶ．離散力学系では，系の時間発展は差分方程式で表現される．ここでは，差分方程式の中でも最も簡単な 1 次差分方程式で表現される系を考える．例えば，ある系において時刻 t における状態変数を x_t として，系の時間発展が (3.3) 式の 1 次差分方程式で表現されるとする．

$$x_{t+1} = f(x_t) \tag{3.3}$$

この系の長期的な時間発展は，$f(x)$ の繰り返し写像 (iterated map) で決定される．系の初期状態を x_0 とすると時刻 T の系の状態 x_T は (3.4) 式で表現される．

$$x_T = \underbrace{f(f(\ldots f}_{T}(x_0)\ldots)) \tag{3.4}$$

学習後のリカレントニューラルネットワークの動作は，コンテクストユニットの初期状態に (3.1) 式を繰り返し適用して，それと並行して (3.2) 式により出力を決定する過程として考えることができる．したがってリカレントニューラルネットワークの動作を離散力学系の時間発展と同様に考えることができる．

3.4　非線形力学系と複雑適応系

系の時間発展を決定する法則に非線形要素を含むものを非線形力学系と呼ぶ．非線形力学系においては，系のパラメータの僅かな変化により系の力学構造が大きく変化する．これが複雑な分岐現象を生むことは 2.2.3 項でも述べた (図 2.8)．リカレントニューラルネットワークの時間発展は (3.1) 式で表現される．これは，リカレントニューラルネットワーク全体の振舞いを表現する式である．全体の振舞いは，リカレントニューラルネットワークを構成する個々のユニットの複雑な相互作用により決定される．個々のユニットの振舞いは，次のシグモイド関数により決定される．

$$x_i(t+1) = \frac{1}{1+\exp(-(\sum_j w_{ij}x_j(t)+\theta_i))} \tag{3.5}$$

ここで，シグモイド関数は図 3.1 に示すように非線形な性質を持つので，リカレントニューラルネットワークの時間発展には非線形要素が当然含まれることになる．したがって，リカレントニューラルネットワークを非線形力学系として捉えることができる．

リカレントニューラルネットワークの学習は，(3.1) 式においてユニット間の結合の重みのマトリックス W を出力エラーを減少させる方向に修正していく過程である．リカレントニューラルネットワークの時間発展の方程式は，この W により決定される．したがって，リカレントニューラルネットワークを力学系として捉えたときの系の制御パラメータであると考えることができる．よって，リカレントニューラルネットワークの学習は，非線形力学系において系のパラメータを変化させることと等価である．このように，動作中に系のパラメータ自体を変化するような系は特に，複雑適応系 (Complex Adaptive System) と呼ばれている．

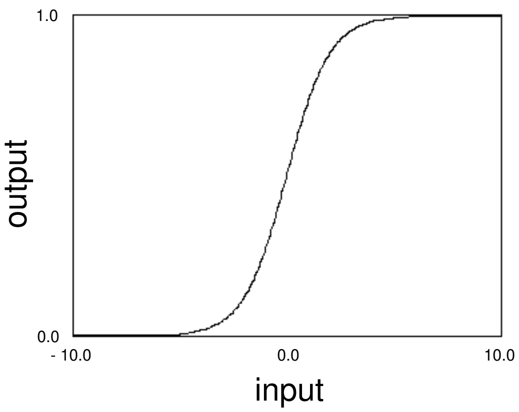


図 3.1: シグモイド関数

3.5　力学系の枠組としての見方

リカレントニューラルネットワークを力学系の枠組で捉えたとき，系の状態を記述する状態変数をコンテクストユニットの取る値とすることができる．このコンテクストユニットで張られる相空間上において，リカレントニューラルネットワークの状態を特定することができるため，その動作はこの相空間におけるトラジェクトリとして表現される．つまり，あるリカレントニューラルネットワークの動作の性質は，この相空間上のトラジェクトリが示す力学構造により反映されることになる．なお，リカレントニューラルネットワークの動作の性質は，学習による結果である．したがって，力学構造が学習の結果を表現していると考えられる．

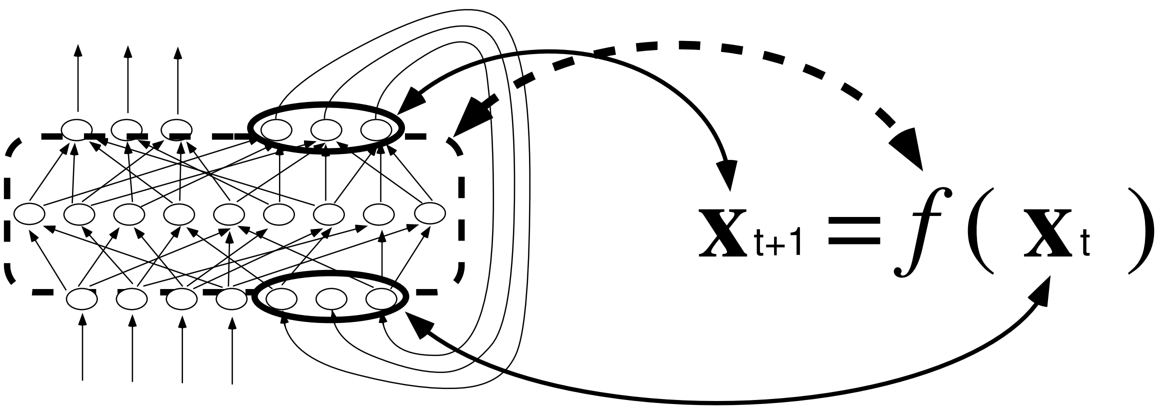


図 3.2: 力学系の視点からのリカレントニューラルネットワーク

3.6　まとめ

本章では，リカレントニューラルネットワークの離散力学系と非線形力学系との関係について説明した．

力学系の視点からのカレントニューラルネットワークをまとめると，その学習は非線形力学系におけるパラメータを変化させる複雑適応系であり，学習結果は系の力学構造となる．学習後の動作つまり状態の更新は，離散力学系の時間発展と同様である．

第 4 章

リカレントニューラルネットワークの学習

4.1 リカレントニューラルネットワークの学習

本章では，リカレントニューラルネットワークの学習に関して行った実験について述べる．時系列情報としてリミットサイクルのパターンと有限状態機械で表現されるパターンの 2 種類をそれぞれ学習させる．その過程を力学系の視点から解析する．なお，本研究で行う実験は，リカレントニューラルネットワークをエージェントと見立てて，そのエージェントがリミットサイクルや有限状態機械で表現される環境を学習するという状況を想定している．さらに，エージェントが未学習の環境に遭遇したときに，内部に構築していく力学構造や，学習中に環境を追加した場合の力学構造の変化についての実験と解析を行う．

4.2 リミットサイクルの学習

まずはじめに時系列情報としてリミットサイクルの学習を行わせる．リミットサイクルとはある周期で同じ情報列が繰り返されるもので，時系列情報の中でも比較的単純なものである．

4.2.1 実験の設定

リミットサイクルの学習では，リカレントニューラルネットワークとしてのエージェントがリミットサイクルで表現される環境を学習するという想定の下で実験を行う (図 4.1)．エージェントは環境に関する情報をセンサ入力とし，その入力のサンプルを基に学習を行う．ここでのエージェントにとっての学習の目的は，エージェントがその環境中を動作するときに，環境から得られるであろうセンサ入力を順次予測できるようになることである．

リカレントニューラルネットワークの設定

リミットサイクルの学習で用いるリカレントニューラルネットワークの構造は，以下の通りである (図 4.1)．各ユニットの数を以下に列挙する．

- 入力ユニットなし．
- 隠れユニット 4 つ．
- 出力ユニット 1 つ．
- コンテクストユニット 3 つ．

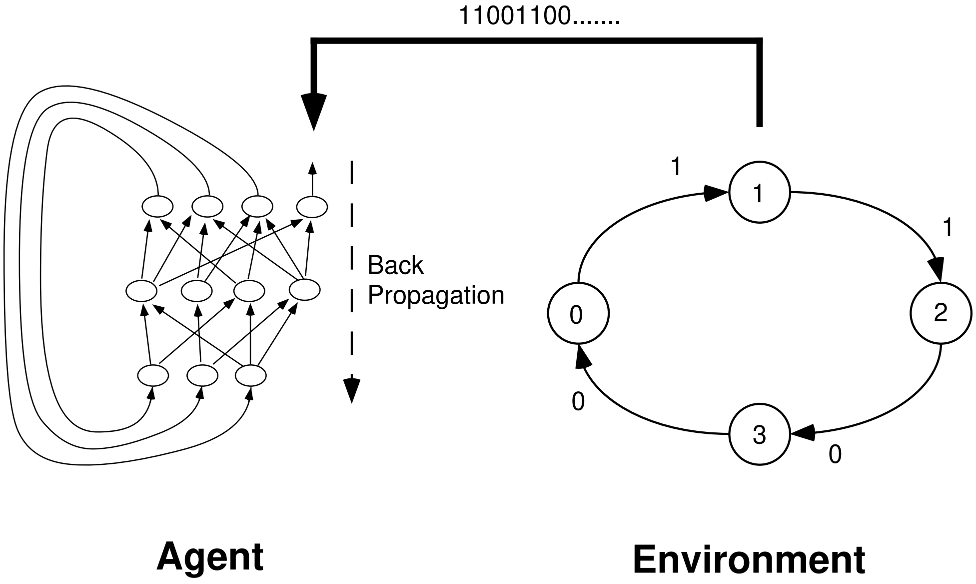


図 4.1: リミットサイクルの学習

学習アルゴリズム

本研究では実験の枠組上，学習の実時間性は必要ではないため，学習アルゴリズムとしては，計算量の少ない BPTT を用いる．BPTT アルゴリズムの詳細は Appendix A にて説明する．

BPTT に必要な各パラメータを以下に列挙する．

- 学習率 η は 0.05 とする．
- 慣性係数 α は 0.96 とする．
- 減衰率 ν は 0.95 とする．
- 1 回の学習ステップにおける時間ステップ 12 とする．
- 学習ステップは $10,000$ とする．

学習対象

学習対象として，図 4.1 に示すリミットサイクルを与える．このリミットサイクルは 4 状態の有限状態機械で表現されており，周期 4 の記号列を生成する．しかし，リカレントニューラルネットワークには，周期 4 であることは明示的には示されていない．こ

こで，学習ステップ 1 に対する時間ステップを 12 としているので，リカレントニューラルネットワークの学習サンプルには 4 個の記号列が 3 周期含まれていることになる．

実験の手順

実験の手順を以下に示す．

1. 学習サンプルの収集
   エージェントを環境中で動作させ，センサ入力列を集める．

2. 環境の学習
   センサ入力列を教師信号として，BPTT により学習を行う．

3. 環境中の動作
   エージェントを再び環境中で動作させる．

4.2.2 実験結果

リカレントニューラルネットワークにリミットサイクルを学習させたときの学習の様子を図 4.2 に示す．(a) は，学習ステップに対する学習エラーの推移を示している．(b)，(c)，(d) は，各学習ステップごとにリカレントニューラルネットワークを動作させときに，それぞれ 3 つのコンテクストユニットの出力を縦軸に取ったものである．(e) は，そのときの出力ユニットの出力を縦軸に取ったものである．

(a) を見ると学習エラーは学習ステップ 4000 付近で急激に減少している．(b)，(c)，(d) を見ると，ちょうどこのときプロットが 4 つに分かれている．最初，各コンテクストユニットの取る値は 1 つだけである．これに対して 4000 ステップ付近から，各コンテクストユニットは 4 つの値を取っている．(e) を見ると，出力ユニットのプロットも同様の傾向を示している．

このときのリカレントニューラルネットワークの力学構造について考えてみる．コンテクストユニットが 1 つの値だけを取るというのは，力学構造的には不動点のアトラクタになっていると考えることができる．一方，4 つの値をとるというのは，リカレントニューラルネットワークの力学構造が周期 4 のリミットサイクルのアトラクタになっていると考えることができ，これは学習する環境からのセンサ入力が $11001100\ldots$ の周期 4 のリミットサイクルであることに対応している．この不動点からリミットサイクルへの力学構造の変化は，学習による重みの変化によるものである．これは，非線形力学系に特有の分岐現象であると考えられる．

リカレントニューラルネットワークの力学構造が不動点のアトラクタであるので，相空間で安定となるのはこの点だけである．つまり，リカレントニューラルネットワークの取る状態は 1 つだけである．一方，力学構造が周期 4 のリミットサイクルのアトラクタであるということは，相空間では 4 つの点の周期的に巡回するトラジェクトリが安定となり，これはリカレントニューラルネットワークが 4 つの状態を取ることができることを示す．

次に出力について考えてみる．リカレントニューラルネットワークが 1 つの状態しか取ることができないときは，それに対応した 1 種類の出力しか出すことができない．しかし，リカレントニューラルネットワークが 4 つの状態を取ることができるようになると，各状態に対して別々の出力を出すことができるようになる．したがって，この分岐時に学習エラーが急激に減少するのである．

学習後のリカレントニューラルネットワークの相空間を図 4.3 に示す．それぞれリカレントニューラルネットワークのコンテクストユニットにランダムな値を 50 個与え，リカレントニューラルネットワークを 10,000 ステップ動作させ，そのトラジェクトリをプロットしたものである．(b) と (d) は，動作ステップが 200 ステップ以降のトラジェクトリだけをプロットした．(a) と (b) は，コンテクストユニット 0 の出力を横軸に，コンテクストユニット 1 の出力を縦軸に取ったものである．(c) と (d) は，コンテクストユニット 0 の出力を横軸に，コンテクストユニット 2 の出力を縦軸に取ったものである．

(b) と (d) をみると，相空間のトラジェクトリは完全に 4 点になっている．このことからも，リカレントニューラルネットワークの力学構造が周期 4 のリミットサイクルのアトラクタになっていることが確認できる．各点は，リミットサイクル中の各状態に対応しており，この 4 点を周期的に巡回することにより，4 つの状態を作り出している．

(a) と (c) は，ランダムに決定した初期値から 10000 ステップの全ての状態が記録されたトラジェクトリで，トラジェクトリがアトラクタに落ち込むまでのトランジェントの部分を見ることができる．相空間のいかなる初期状態から出発しても最終的には，(b) と (d) のアトラクタに落ちていることが分かる．

4.3 有限状態機械の学習

次に，リカレントニューラルネットワークに時系列情報として，有限状態機械で表現されるパターンを学習させる．

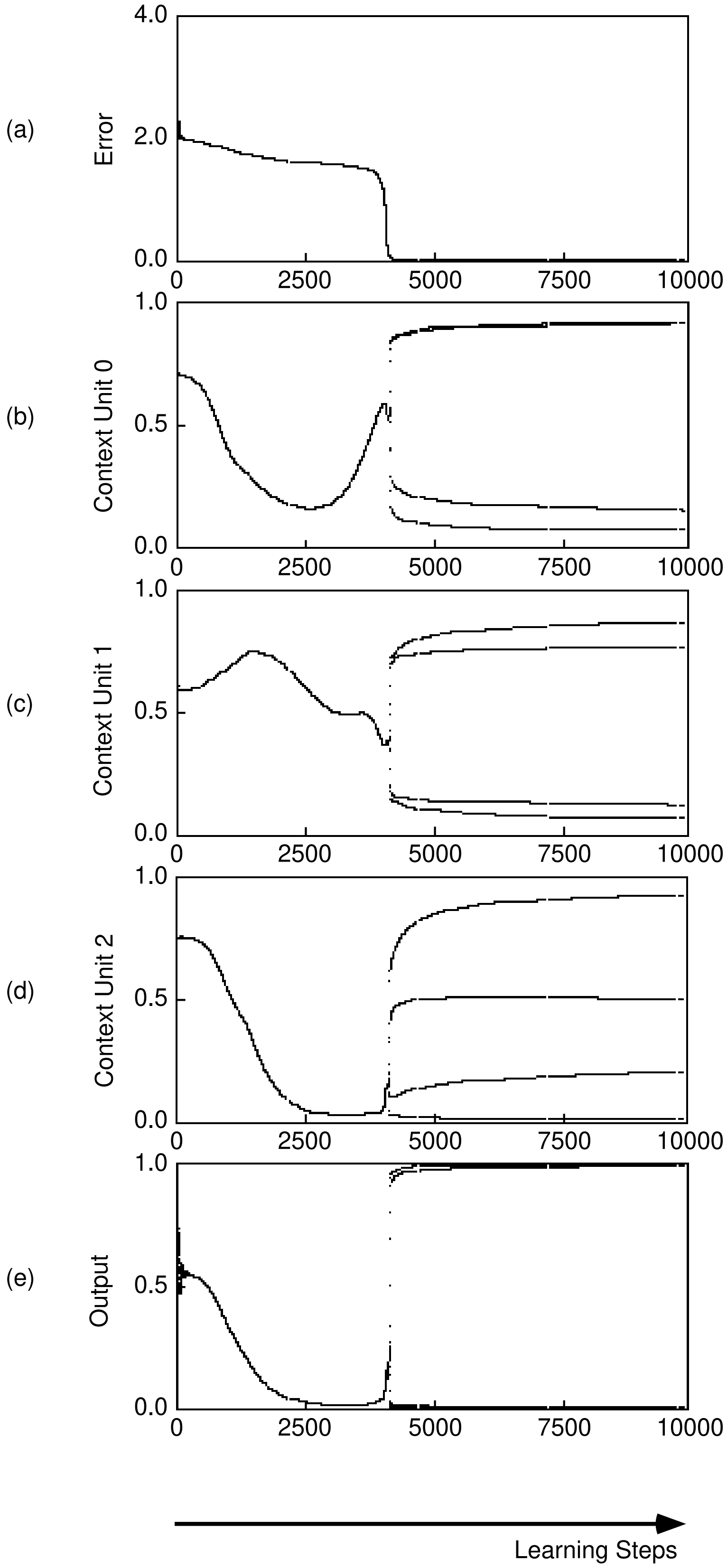


図 4.2: リミットサイクルの学習の様子

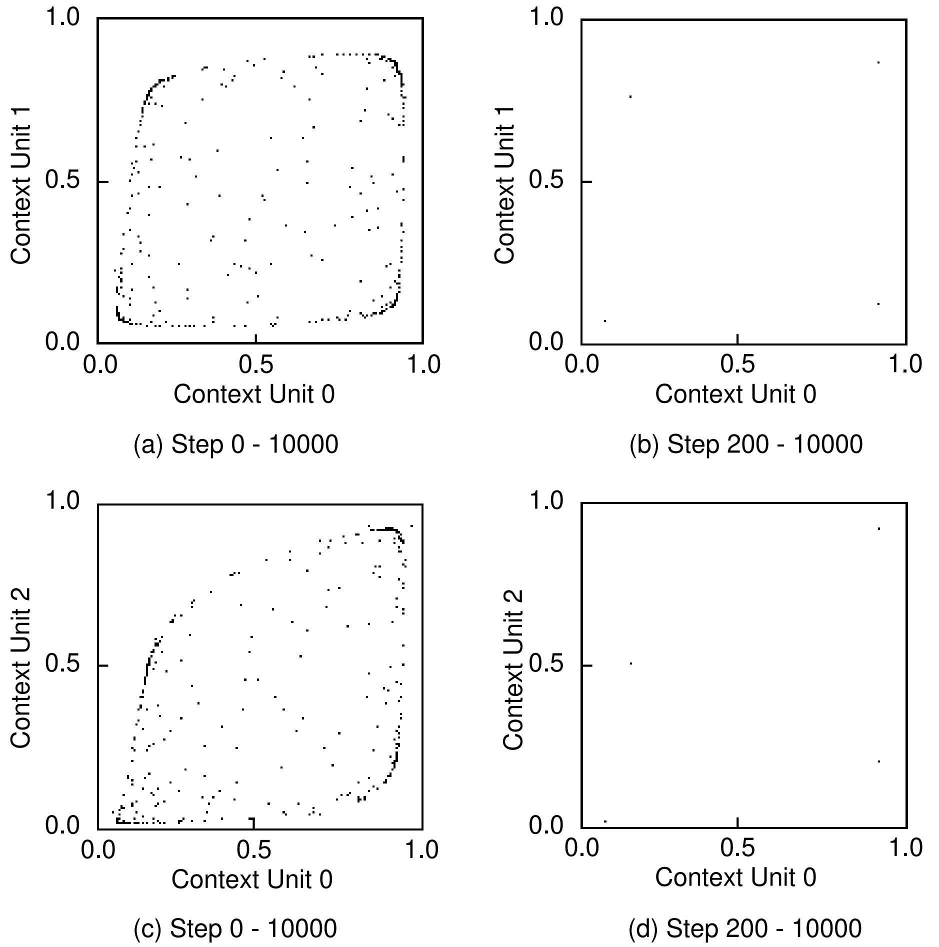


図 4.3: リミットサイクルの学習後の動作

4.3.1 実験の設定

有限状態機械の学習も，リミットサイクルの学習同様，リカレントニューラルネットワークのエージェントが有限状態機械で表現される環境を学習するという想定の下で実験を行う (図 4.4)．エージェントは環境に関する情報をセンサ入力として得て，そのサンプルを基に学習を行う．ただし，有限状態機械の学習では，エージェントはモータコマンドとして行動を選択し，その行動の選択により得られるセンサ入力が決定されるものとする．ここでのエージェントにとっての学習の目的は，現時点でのセンサ入力において，選択した行動を取ることによって環境から得られるであろう次のセンサ入力を，順次予測できるようになることである．

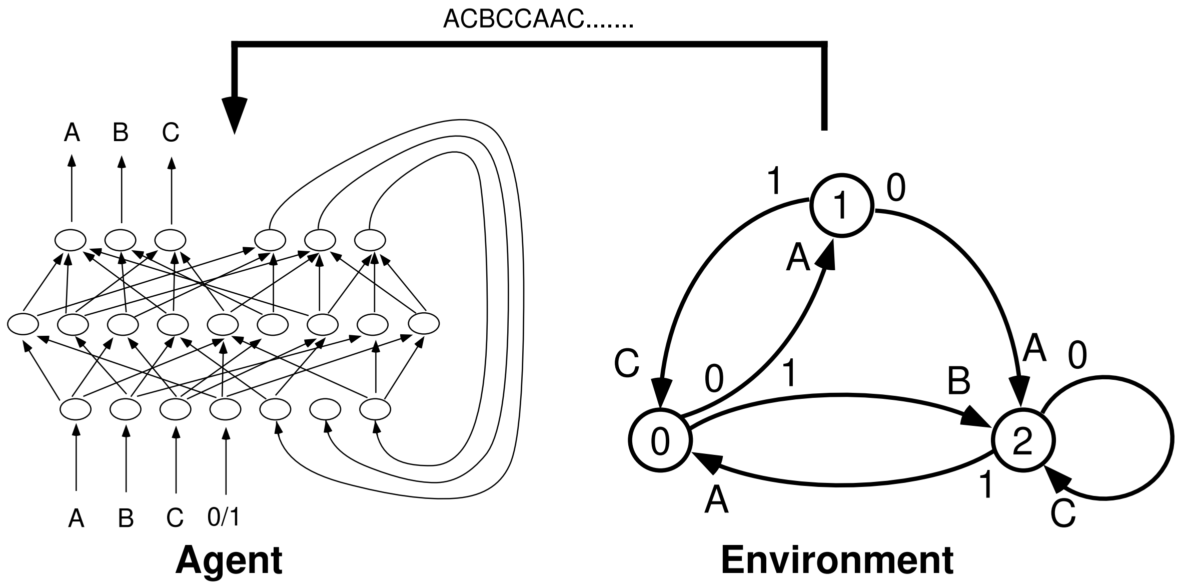


図 4.4: 有限状態機械の学習

リカレントニューラルネットワークの設定

有限状態機械の学習で用いるリカレントニューラルネットワークの構造は図 4.4 に示す通りである．各ユニットの数を以下に列挙する．

- 入力ユニット 4 つ
- 隠れユニット 8 つ．
- 出力ユニット 3 つ．

- コンテクストユニット 3 つ．

学習アルゴリズムの設定

リカレントニューラルネットワークの学習アルゴリズムとして BPTT を用いる．ただし BPTT を用いる際，リカレントニューラルネットワークを並列に並べて学習させ，結合の重みの修正はその並列に並べたリカレントニューラルネットワークの平均を用いる．こうすることにより，リカレントニューラルネットワークが学習サンプルを暗記的に学習することを防ぐことができる．

BPTT に必要な各パラメータを以下に列挙する．

- 学習率 η は 0.02 とする．
- 慣性係数 α は 0.96 とする．
- 減衰率 ν は 0.95 とする．
- 1 回の学習ステップにおける時間ステップは 12 とする．
- 並列に学習させる数は 4 とする．
- 学習ステップは 20000 とする．

学習対象

学習対象として，図 4.4 に示す有限状態機械を学習させる．この有限状態機械の状態数は 3 である．この有限状態機械の動作をエージェントが環境中を動作するという視点から説明する．

エージェントは各状態において，モータコマンドとして 2 種類の選択肢 $\{0, 1\}$ から行動を選択することができる．ある状態においていずれかの行動を取ったときに，3 種類のセンサ入力 $\{A, B, C\}$ のうち 1 つを得て，次の状態へと移動する．こうしてエージェントは環境中を動作する．エージェントが得ることのできる情報は，ある行動を選択したときに与えられるセンサ入力だけであり，エージェント自身は現在自分がどの状態にいるかるのかを知ることはできない．また，センサ入力は状態ごとにユニークなものではないため，現在のセンサ入力だけから現在の状態を知ることはできない．

実験の手順

実験の手順を以下に示す．

1. 学習サンプルの収集
エージェントを環境中で動作させ，センサ入力列を集める．このとき，エージェントはランダムにモータコマンド (行動) を決定する．

2. 環境の学習
モータコマンド列に対するセンサ入力列を教師信号として，BPTT を用いて学習を行う．

3. 環境中の動作
エージェントを再び環境中で動作させる．

4.3.2 実験結果

リカレントニューラルネットワークに有限状態機械を学習させたときの様子を図 4.5 に示す．(a) は，学習ステップに対する学習エラーの推移を示している．(b) は，学習ステップごとにリカレントニューラルネットワークを動作させたときのコンテクストユニットの出力を縦軸に取ったものである．

(a) を見ると，学習エラーの現象は一定の傾きで減少するのではなく，急激な変化や緩やかな変化を繰り返しながら減少している．リミットサイクルの学習同様，学習エラーが急激に減少しているところで分岐が生じている (b)．学習エラーの現象が急激な変化や緩やかな変化を繰り返すのは，何度も分岐が生じているためと考えられる．

ここで力学構造の変化についてより深く考察してみる．200 ステップ付近で分岐が生じて，不動点から数周期のリミットサイクルに変化している．リカレントニューラルネットワークはここでいくつかの状態を取れるようになり，学習エラーが急激に減少するが，完全には減少しきらない．これは，リカレントニューラルネットワークの状態の分割が，学習対象の有限状態機械に対して適切ではないためである．しばらく学習が進むと，2000 ステップ付近でまた新たな分岐が生じている．ここでは，リカレントニューラルネットワークは状態を学習対象に対してより適切に分割している．

リカレントニューラルネットワークの学習は，2 つの段階に分けることができる．1 つ目の段階では，リカレントニューラルネットワークの取り得る状態の数を変えることで学習エラーを減少させる．2 つ目の段階では，分割した状態ごとにその状態の出力が適切になるように重みの変更を行う．

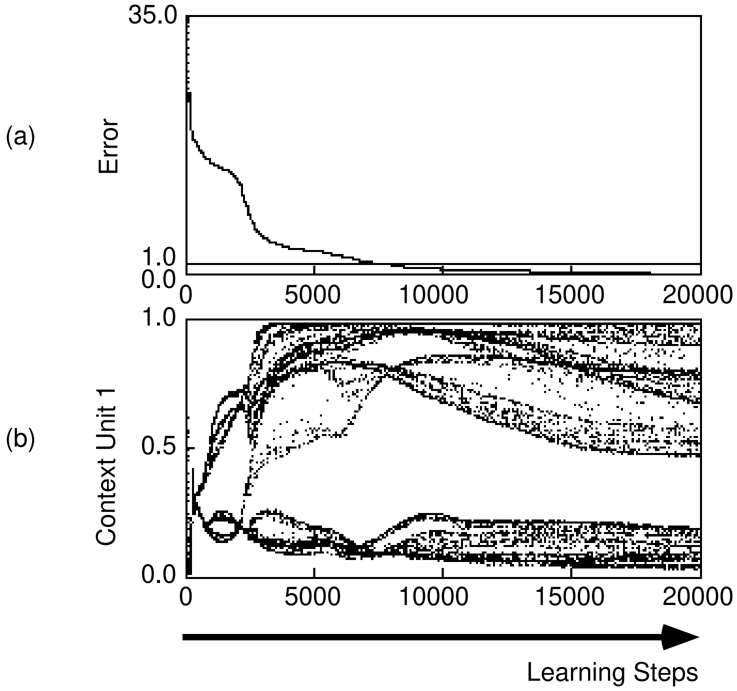


図 4.5: 有限状態機械の学習の様子

学習後のリカレントニューラルネットワークの相空間を図 4.6 に示す．コンテクストユニットにランダムな値を与えて，10000 ステップ動作させて，そのトラジェクトリをプロットしたものである．ただし，1000 ステップ以降のトラジェクトリのみをプロットしている．

これを見ると，相空間のトラジェクトリはいくつかのセグメントになっていることが分かる．このセグメントはコンテクストユニットの初期状態を変えて動作させても同様の場所に現れる．したがって，このセグメントはリカレントニューラルネットワークの力学系としてのアトラクタである．各セグメントにあるトラジェクトリを，有限状態機械の状態と対応させてみると，1 つのセグメント内のトラジェクトリは全て有限状態機械において同一の状態に結び付いていることが確認できる．したがって，この各セグメントは学習対象の有限状態機械の各状態に対応していると考えることができる．ただし，セグメントの数は 5 つあるのに対して，有限状態機械の状態数が 3 つであるため完全に対応がるわけではない．これは，リカレントニューラルネットワークが有限状態機械を学習する際，その有限状態機械の構造そのものを学習するのではなく，その有限状態機械が生成し得る記号列の構造を学習する．この記号列は有限状態機械のサンプルにすぎず，全く同じ記号列を生成し得る有限状態機械は他にも無数に考えられる．そもそも，

エージェントは与えられた有限状態機械の正しい構造そのものを知るすべはなく，むしろエージェントにとってはあり得る記号列を生成できる有限状態機械はいかなる構造を持っていようとも，正しいものとして認識される．この実験において，学習エラーが十分に減少していることから，状態数は多いけれども学習対象と等価な有限状態機械を学習していると考えられる．Pollack は，リカレントニューラルネットワークが最小状態数の有限状態機械を学習することは $\mathcal{NP}$ 完全な問題であることを確認している [10]．

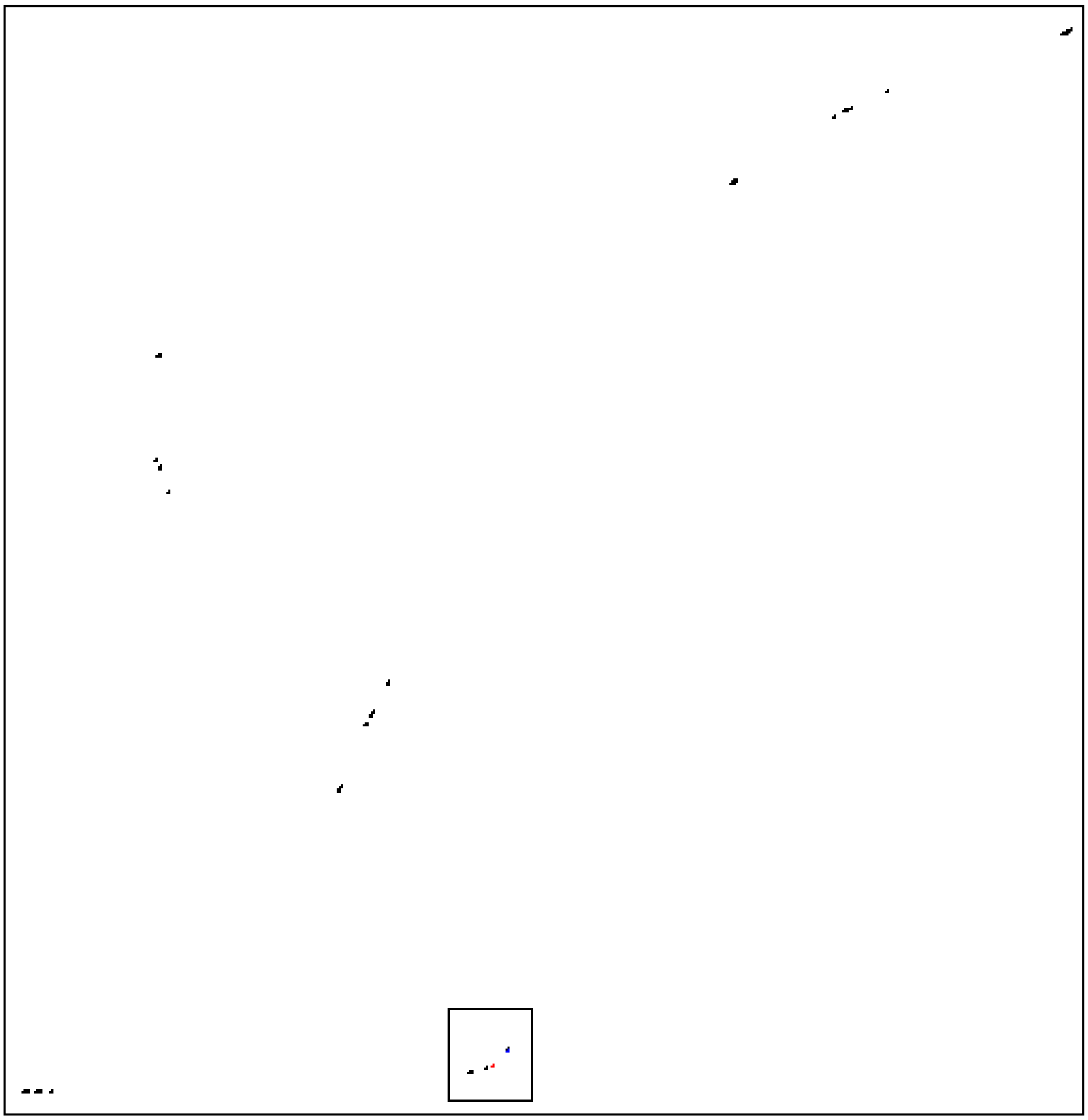


図 4.6: 有限状態機械学習後の動作 (1)

4.4 未学習の環境

有限状態機械の学習では，リカレントニューラルネットワークは相空間においてセグメントを形成することにより学習を行った．また，リカレントニューラルネットワークは学習した環境に対しては学習によって獲得した力学構造でうまく対処することができた．では，未学習の環境に対してこの力学構造はどうなるのだろうか．ここでは，未学習の環境に対するリカレントニューラルネットワークの力学構造について実験と解析を行う．

4.4.1 実験の設定

この実験も有限状態機械の学習同様，リカレントニューラルネットワークのエージェントが有限状態機械で表現される環境を学習するという想定の下で行う (図 4.7)．ただし，エージェントが学習する環境と学習後にエージェントが動作するは環境は異なるものとする．リカレントニューラルネットワークの設定，学習アルゴリズムの設定は有限状態機械の学習と同様とする．

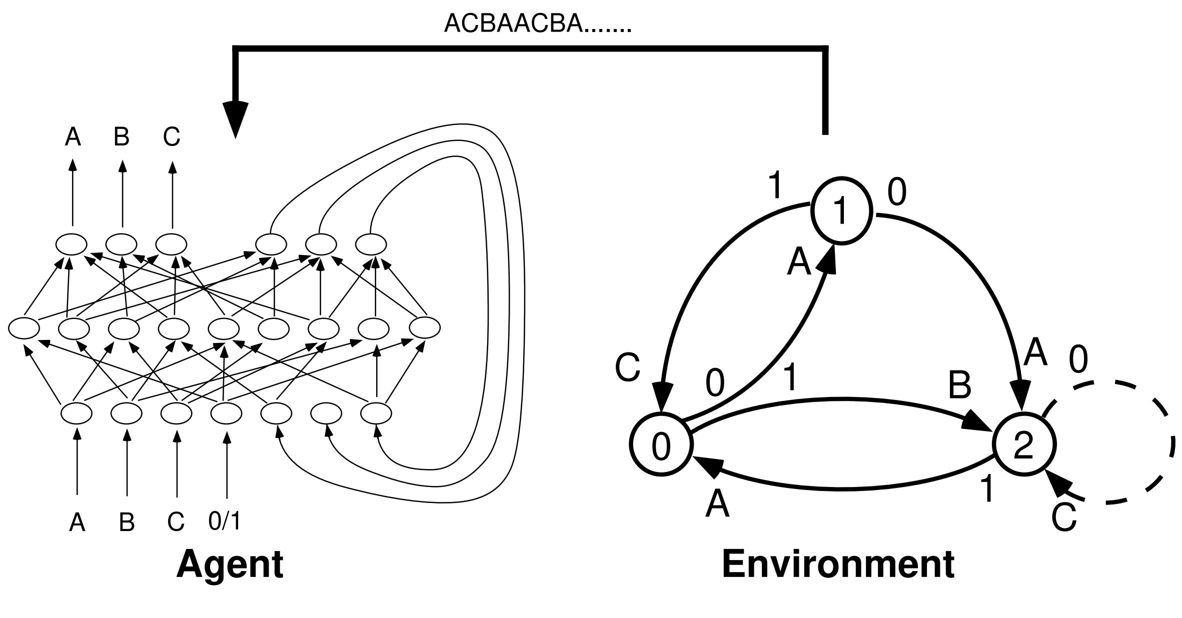


図 4.7: 未学習の環境

学習対象

学習対象として，図 4.7 に示す有限状態機械を学習させる．ここでは，実線で示されているパスのみから成る環境を学習させる．学習後，エージェントを環境が動作するときには，状態 2 に点線で示されているパスを新たに付け加えた環境で動作させる．エージェントにとってはこのパスが未学習の環境となる．

実験の手順

実験の手順を以下に示す．

1. 学習サンプルの収集
   エージェントを環境 (点線のパスなし) 中で動作させ，センサ入力列を集める．このとき，エージェントはランダムにモータコマンド (行動) を決定する．
2. 環境の学習
   モータコマンド列に対してセンサ入力列を教師信号として，BPTT を用いて学習を行う．
3. 環境中の動作
   エージェントを異なる環境 (点線のパスあり) 中で動作させる．

4.4.2 実験結果

学習後のリカレントニューラルネットワークの相空間を図 4.8 に示す．(a) はリカレントニューラルネットワークを学習した環境中で，(b) は未学習の環境中で動作させたものである．これらの図はコンテクストユニットにランダムな値を与えた後に，10000 ステップ動作させてそのトラジェクトリをプロットしたものである．なお，1000 ステップ以降のトラジェクトリだけをプロットしている．

(a) と (b) を比較してみると，全体的に大きな構造上の違いはない．ただし，(b) の方がセグメントの数が多く，個々のセグメントの大きさも大きい．新たに生じたセグメントは，エージェントが未学習の環境 (点線のパス) を通った直後の状態に対応している．この状態では，エージェントは自分の状態を見失っているのでその出力にエラーを生じている．

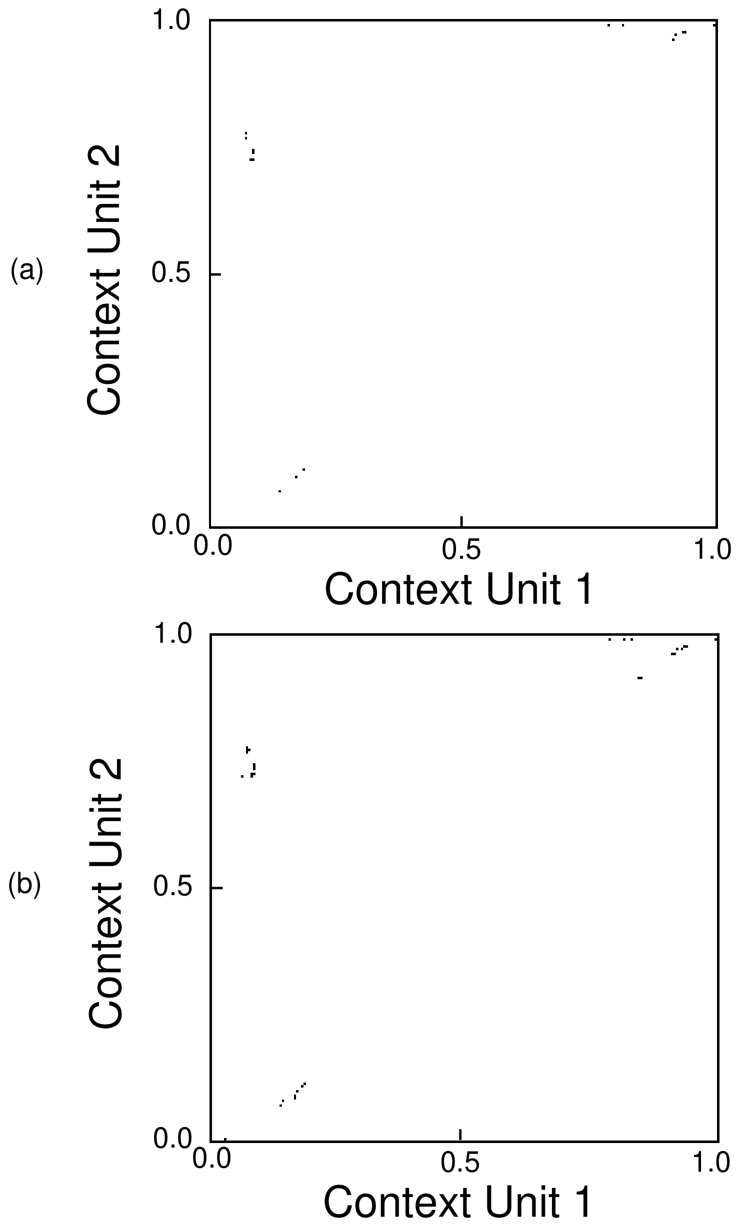


図 4.8: 学習後のリカレントニューラルネットワークの動作

4.5 追加学習

前節では，未学習の環境に対するリカレントニューラルネットワークの力学構造の解析を行った．本節では，ある環境を学習した後に，環境を追加して再び学習させた場合にリカレントニューラルネットワークの力学構造がどのように変化するかを解析する．

4.5.1 実験の設定

この実験もリカレントニューラルネットワークのエージェントが有限状態機械で表現される環境を学習するという想定の下で行う (図 4.9)．ただし，エージェントが環境を学習中に学習エラーが十分に減少したところで，新たに環境を追加して再び学習させる．

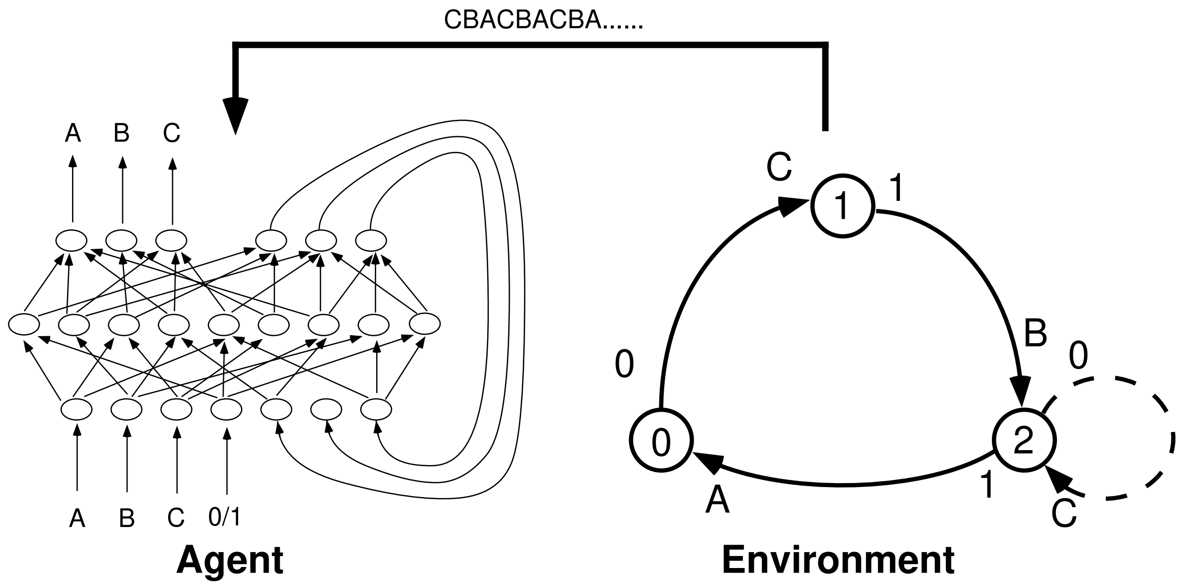


図 4.9: 追加学習

リカレントニューラルネットワークの設定，学習アルゴリズムの設定は有限状態機械の学習と同様とする．ただし環境追加後は，追加前の環境と追加後の環境を並列に学習させる．

学習対象

学習対象として，図 4.9 に示す有限状態機械を学習させる．まず，エージェントは実線のみで示されている環境を学習し，学習エラーが十分減少したところで，状態 2 に点線で示されているパスを新たに付け加えた環境を学習させる．

実験の手順

実験の手順を以下に示す．

1. 学習サンプルの収集 (1)
エージェントを環境 (点線のパスなし) 中で動作させ，センサ入力列を集める．このとき，エージェントはランダムにモータコマンド (行動) を決定する．

2. 環境の学習 (1)
モータコマンド列に対するセンサ入力列を教師信号として，BPTT を用いて学習を行う．

3. 学習サンプルの収集 (2)
学習エラーが十分に減少したところで，エージェントを変化した環境 (点線のパスあり) に入れて動作させ，センサ入力列を集める．このとき，エージェントはランダムにモータコマンド (行動) を決定する．

4. 環境の学習 (2)
モータコマンド列に対するセンサ入力列を教師信号として，BPTT を用いて学習を行う．

4.5.2 実験結果

以上の実験を行ったときの結果を図 4.10 に示す．(a) は，学習ステップに対する学習エラーの推移を示している．(b) は，各学習ステップごとにリカレントニューラルネットワークを動作させたときのコンテクストユニットの出力を縦軸に取ったものである．ただし，学習エラーが十分に減少したところで，環境を追加している．

5000 ステップ付近までは，状態 2 に点線で示されるパスがなく，環境を表現する有限状態機械は周期 3 のリミットサイクルである．したがって，リカレントニューラルネットワークの内部にも 3 つの状態を繰り返し取るような力学構造が構築される．5000 ステップ付近で状態 2 に点線で示されるパスが付け加わった環境に変化する．リカレントニューラルネットワークにとってこの新しいパスは未学習なので，それまでに構築した力学構造ではうまく対処できず，相空間のトラジェクトリを表す点が散らばってしまう．しかし，学習が進むにつれて点の散らばりが段々小さくなっている．

なお，図 4.11 において，(a)，(b)，(c) はそれぞれ学習環境追加後に，学習ステップが 0，1000，6000 ステップのリカレントニューラルネットワークを，追加した環境で動

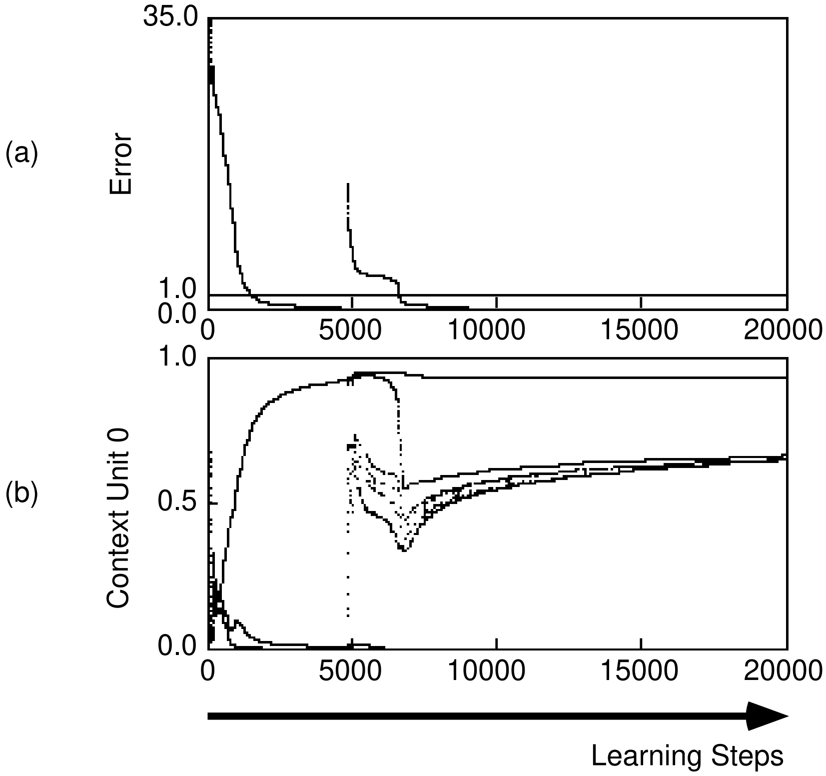


図 4.10: 追加環境の学習の様子

作させたときの相空間のトラジェクトリを表したものである．これらの図はコンテクストユニットにランダムな値を与えてリカレントニューラルネットワークを 10000 ステップ動作させたときのトラジェクトリをプロットしたものである．ただし，2000 ステップ以降のトラジェクトリだけをプロットしている．

環境追加後 0 ステップの (a) では，追加した環境を学習していないので，状態 2 にある点線で示されているパスを通る場合，リカレントニューラルネットワークはうまく対応することができない．したがって，状態 2 に対応するトラジェクトリのセグメントが相空間の広い範囲にまたがって存在している．(b)，(c) と学習が進むに連れて広い範囲に存在していたトラジェクトリは，徐々に 1 箇所に集まり最終的に有限状態機械の学習で見られたセグメントになっている．

新しく環境を追加した時点では，リカレントニューラルネットワークにとってその環境は未学習の環境である．それに対する力学構造が未学習の環境の実験の結果と異なることについては，5.1.3 項で考察する．

4.6 実験結果のまとめ

本章では，リカレントニューラルネットワークの学習の実験を行った．まず，リミットサイクルの学習において，リカレントニューラルネットワークの内部に構築される力学構造もリミットサイクルのアトラクタになることを示した．次に，有限状態機械の学習において，リカレントニューラルネットワークの獲得する力学構造が複数のセグメントになることを示した．未学習の環境に対するリカレントニューラルネットワークの力学構造は，学習した環境に対する力学構造と比較して大きな構造の違いがないことを示した．最後に，学習する環境を学習途中で追加した場合の力学構造の変化について解析し，セグメントの形成の様子を示した．

(a)
(b)
(c)

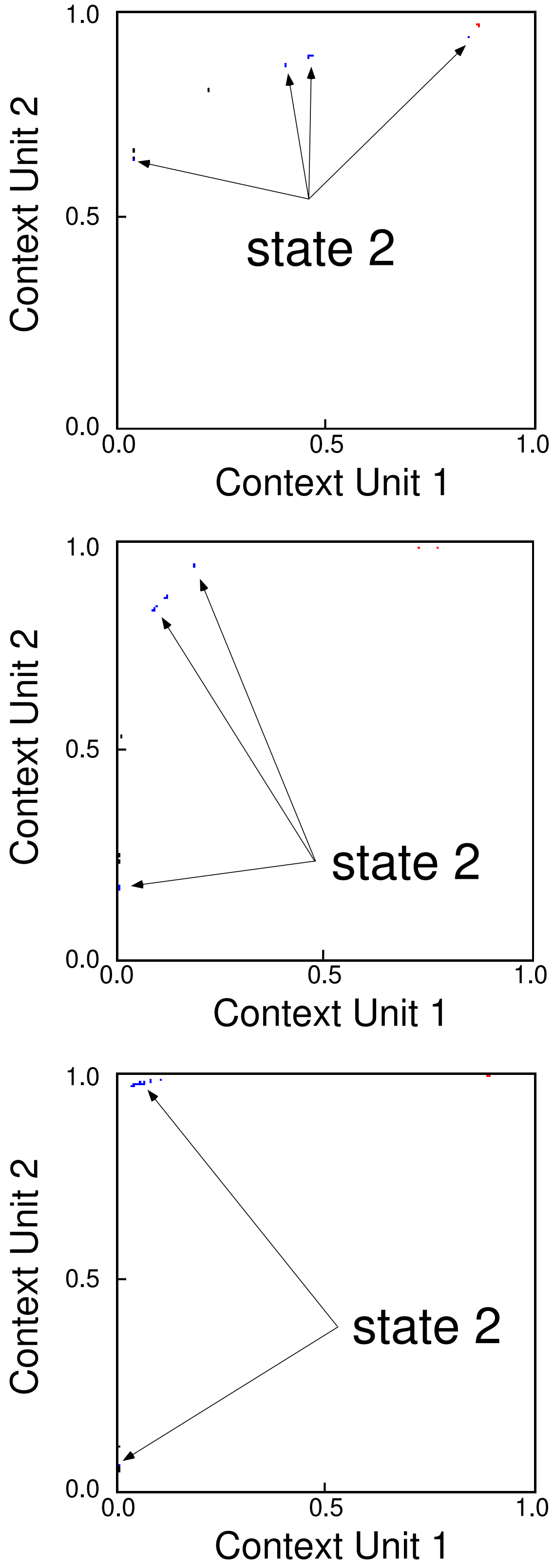


図 4.11: 追加学習における力学構造の変化

第 5 章

議論および今後の課題

5.1 実験結果の考察

本節では，前章で行った実験結果について力学系の視点からの解析を加えた上で，より細部にわたって考察する．

5.1.1 力学構造の比較

リカレントニューラルネットワークにリミットサイクルを学習させた場合，その力学構造はリミットサイクルのアトラクタになる．一方，有限状態機械を学習させた場合，その力学構造は幾つかのセグメントにより構成されるアトラクタになる．リミットサイクルの学習では，リミットサイクルの各状態に対応するものが点で表現されている．一方，有限状態機械の学習では，有限状態機械の各状態に対応するものが完全な 1 点には成らず，いくつかの点から成るセグメントとして表現されている．この違いがどこから生じているのかについて考察してみる．

リミットサイクルの生成する記号列は，必ず周期的に繰り返される．したがって，リカレントニューラルネットワークが現在の状態を知るためには，そのリミットサイクルの 1 周期分の履歴情報を保持することのできる力学構造を形成すればよい．一方，有限状態機械の生成する記号列には，文法的な規則で束縛されているだけで明確な周期は存在しない．逆に言うと，有限状態機械はリミットサイクルと違いあらゆる周期を含む．したがって，有限状態機械を学習したリカレントニューラルネットワークの力学構造は，その全ての周期を取り込むことのできる力学構造でなければならない．リカレントニューラルネットワークが現在の状態を知るために，過去どの状態を通ってきたかという全履歴情報が必要である．

有限状態機械の学習において，リカレントニューラルネットワークがどのように履歴情報をそのセグメントに埋め込んでいるか見るために，形成された力学構造 (図 4.6) の細部についてさらに解析を行った．図 5.1 は，図 4.6 下方の四角い枠で囲まれた部分を拡大したものである．さらに，図 5.1 の四角い枠で囲まれた部分を拡大したものが図 5.2 で，この中の四角い枠をまた拡大したものが図 5.3 である．

図 5.1 を見ると，図 4.6 では 1 つに見えたセグメントが，ここでは大まかに見て 3 つのセグメントから構成されている．さらに，その図 5.1 の 1 つのセグメントは図 5.2 の 3 つのセグメントから構成されていて，そのなかの 1 つのセグメントもまた図 5.3 にある 3 つのセグメントから構成されている．

各図の 1 つのセグメントを小さな 3 つのセグメントの集まりと見ると，全てのセグメントは似た形状をしており，またそれぞれの位置関係はどれも似ている．3 つ並んだセグ

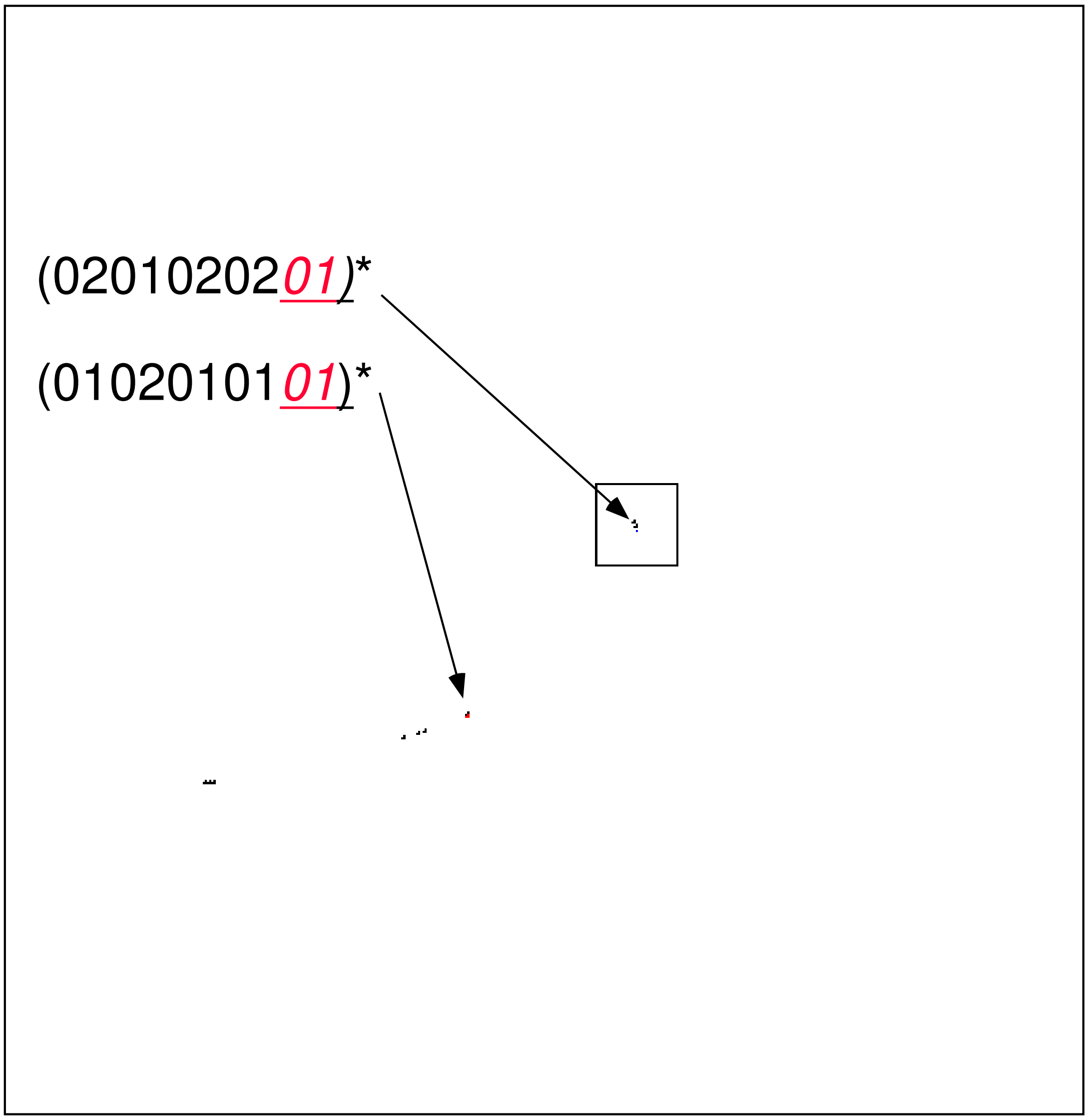


図 5.1: 有限状態機械の学習後の動作 (2)

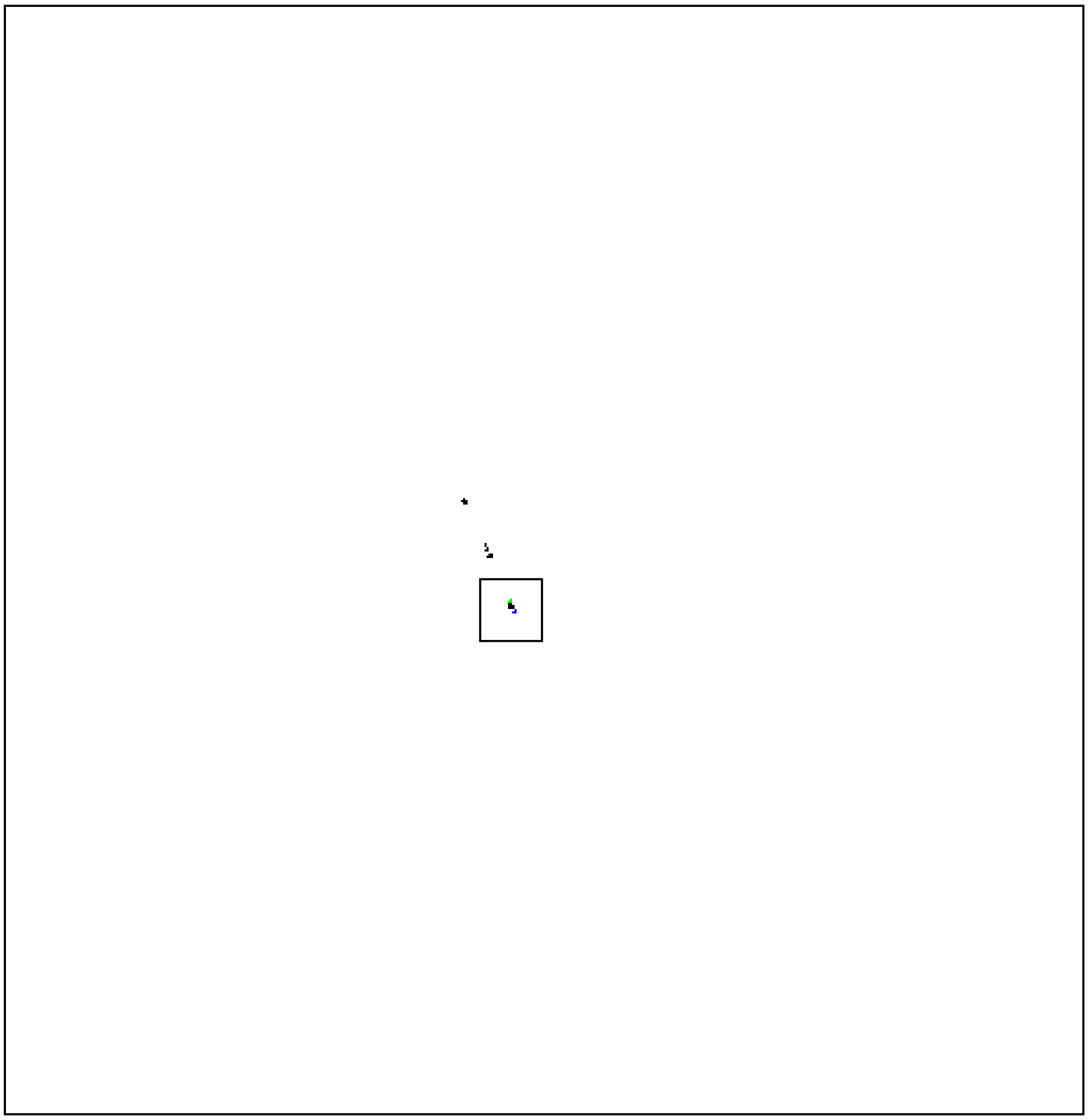


図 5.2: 有限状態機械の学習後の動作 (3)

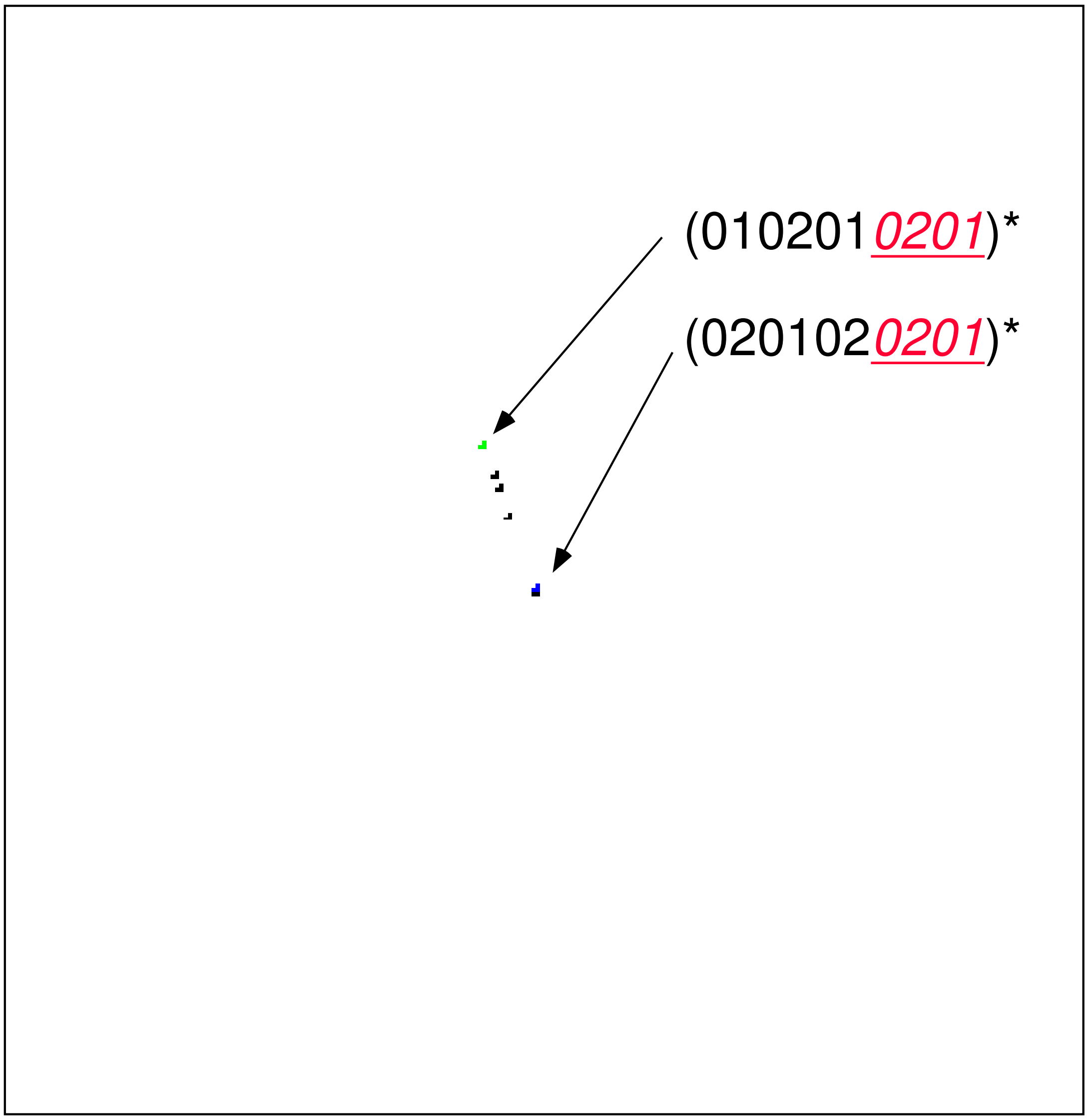


図 5.3: 有限状態機械の学習後の動作 (4)

メントのうち，中央にあるセグメントは外側にあるセグメントに比べて広がりを持っている．また，図 4.6 の他のセグメントでも同様の繰り返し構造が確認できる．つまり，相空間上の全てのセグメントは自己相似構造を持つフラクタル構造になっているのである．

リカレントニューラルネットワークが，このフラクタル構造に履歴情報をどのように埋め込んでいるのかを見るために，この力学系の記号力学系について解析を行った．

リカレントニューラルネットワークのトラジェクトリの中で有限状態機械における状態の履歴が $(02010202\underline{01})^{*}$[1] と $(01020101\underline{01})^{*}$ となる点を探すと，どちらも図 5.1 で示した範囲内に存在している．この 2 つの状態の履歴を比較すると，現在の状態 (1) と 1 ステップ過去にさかのぼった状態の履歴 (0) が等しく，それ以前の状態は異なる状態を通っている．つまり，この 2 つの点は 1 ステップ過去にさかのぼった状態からの履歴は同じであるが，それ以前の履歴は異なる．次に状態の履歴が $(010201\underline{0201})^{*}$ と $(020102\underline{0201})^{*}$ となる点を探すと，どちらも図 5.3 に示した範囲内に存在していることが分かる．この 2 つの状態の履歴を比較すると，現在の状態 (1) から 3 ステップ過去にさかのぼった状態までの履歴が同じで，それ以前の履歴が異なる．

解析の結果をまとめると，1 ステップ過去までの履歴が同じ点は遠くから見ると同じセグメント内にあるが，そのセグメントの中ではそれぞれ別のセグメントとして区別されている．さらに，3 ステップ過去までの履歴が同じ点は一番大きなセグメントから 2 回拡大したセグメントまでは同じセグメントにあるが，そのセグメントの中では別のセグメントとして区別される．つまり，フラクタル構造における階層と履歴が一致しているのである．フラクタル構造は部分に全体の縮図を持つので，その縮図の中に前状態が埋め込まれている．フラクタル構造全体で考えると，全履歴が埋め込まれていることになる．

Kolen はこのリカレントニューラルネットワークによる Finite State Machine の学習において形成される力学構造がフラクタル構造になることを，リカレントニューラルネットワークを Barnsley の Iterated Functions Systems (IFS)[9] として見ることにより説明できることを示した [11]．

IFS により生成されるフラクタルは，全ての点に対して一意なアドレスをつけることができる．シェルピンスキーギャスケットの例を用いて説明する．シェルピンスキーギャスケットは，(2.1) 式，(2.2) 式，(2.3) 式の 3 つの縮小写像からなる IFS により生成される．この 3 つの写像を組み合わせて繰り返し写像を行うことにより，フラクタル構造を持つシェルピンスキーギャスケットを形成する (図 2.11)．

このとき，シェルピンスキーギャスケットの各三角形は写像の組合せにより一意に定

[1] ()*は，() 内の記号列の繰り返しを表す．

まる．図 5.4 の (a) を 3 つの写像により写したものが (b) である．各写像により写された先に用いられた写像の番号を付けている．(b) をまた 3 つの写像により写したものが (c) である．12 という番号は，(a) をはじめに W_1 の写像で写したものを次に W_2 の写像で写したことを表して n いる．(c) の各三角形は，2 つの写像により番号付けをすることができる．この番号をアドレスと呼ぶ．IFS により n 回写像されるとすると，各三角形は n 桁のアドレスを持つことになる．このアドレスを三角形の履歴と考えると，(c) の 9 つの三角形はそれぞれ 1 状態前までの履歴を持つ．ここで，(c) の 9 つの三角形を一回り大きな 3 つの三角形で区切ると，それぞれの中には，現在の状態 (一番最近に写された写像の番号) が同じものが入っている．そして，その中では 1 状態前の状態により区別されている．これを n 回の写像をした場合を考えると，各三角形の中で繰り返し区別されていると考えることができる．

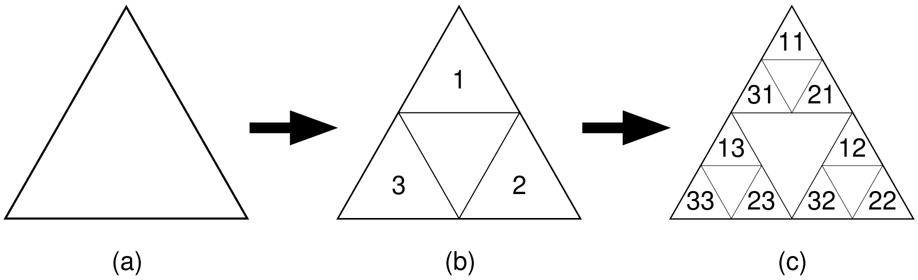


図 5.4: IFS におけるアドレス

リカレントニューラルネットワークがフラクタルな力学構造に履歴を埋め込んでいることは，この IFS により生成されたシェルピンスキーギャスケットにおけるアドレスと同じ機構なのである．

さて，学習によりユニット間の結合の重みが適切に調節された後は，リカレントニューラルネットワークは動作中にその重みを変更することはない．したがって，コンテクストユニットの状態をユニット間の結合の重みで決定される 1 つの写像で繰り返し更新していくことになる．しかし，有限状態機械の学習に用いたリカレントニューラルネットワークは，入力ユニットを持つ．したがって入力ユニットに与えられる値により，コンテクストユニットの状態が同じであっても異なる点に写像され得る．エージェントは環境中でランダムにモータコマンド (行動) を選択するため．リカレントニューラルネットワークにはランダムな値が入力されるため，その写像はランダムに選択される．これは，IFS におけるカオスゲームの原理と同じである．したがって，リカレントニューラルネットワークの動作により生成されたフラクタル構造は不変集合となる．

5.1.2 未学習の環境に対する力学構造

リカレントニューラルネットワークの学習した環境に対する力学構造と未学習の環境に対する力学構造を比較した．図 5.5 は，リカレントニューラルネットワークを学習した環境中を動作させたときのトラジェクトリと，未学習の環境中を動作させたときのトラジェクトリとを重ねてプロットしたものである．未学習の環境の中に学習した環境が含まれているので，学習したトラジェクトリの下には必ず未学習のトラジェクトリがある．未学習の環境中を動作したときのトラジェクトリが，学習した環境中を動作したときのトラジェクトリに比べて大きく崩れることはない．リカレントニューラルネットワークは，未学習の環境中においてもその動作を大きく崩すこともない．未学習の入力により力学構造から外れても，力学系の時間発展にあたる動作 (状態の更新) を続けていく間に，大域的力学構造の性質により再びその力学構造の中に戻ってくるからである．

さて，これまで述べてきたとおり，未学習の環境中を動作させたときのトラジェクトリと学習した環境中を動作させたときのトラジェクトリは相空間上において混在している．図 5.6 は図 5.5 の四角い枠でで囲まれた部分を拡大したものである．これを見ると，2 つのトラジェクトリが相空間上においてフラクタル的に混在している様子をはっきり見ることができる．

ここでも，有限状態機械の学習における力学構造の解析で用いた記号力学系による解析を行った．トラジェクトリの中で，状態の履歴が $(01\underline{2012010})^*$ と $(02\underline{2012010})^*$ の点を探し，この 2 つの状態の履歴を比較すると，現在の状態 (0) から 6 ステップ過去にさかのぼった状態までの履歴が同じで点である．しかし，$(02\underline{2012010})^*$ はその 1 つ前の状態が 2 であるので，2 の状態において未学習の環境を通っている．どちらの点も図 5.6 で示した範囲内に存在している．つまり，図 5.5 のように遠くから眺めると四角い枠で囲まれた狭い範囲内に存在している．未学習の環境を通ったトラジェクトリは学習した環境だけを通ったもののうち良く似た履歴を持つトラジェクトリのすぐ近くに存在するのである．学習した環境だけを通った履歴のどれを考えても，その 1 つ前に未学習の環境を通った履歴を考えることができる．学習した環境だけを通った履歴を持つどの点の近傍にも，必ず未学習の履歴を通った点が存在するのである．

リカレントニューラルネットワークは，学習したものと未学習のものを相空間において，明確な線を引いて区別しているわけではない．一見混在しているが，それをフラクタル構造を持つ力学構造のなかで区別しているのである．学習した環境中を動作させたときの力学構造と未学習した環境中を動作させたときの力学構造を比較したときに，未学習における個々のセグメントが若干大きいのは，未学習の状態を通った履歴を保持する分余計な力学構造を必要とするためであると考えられる．

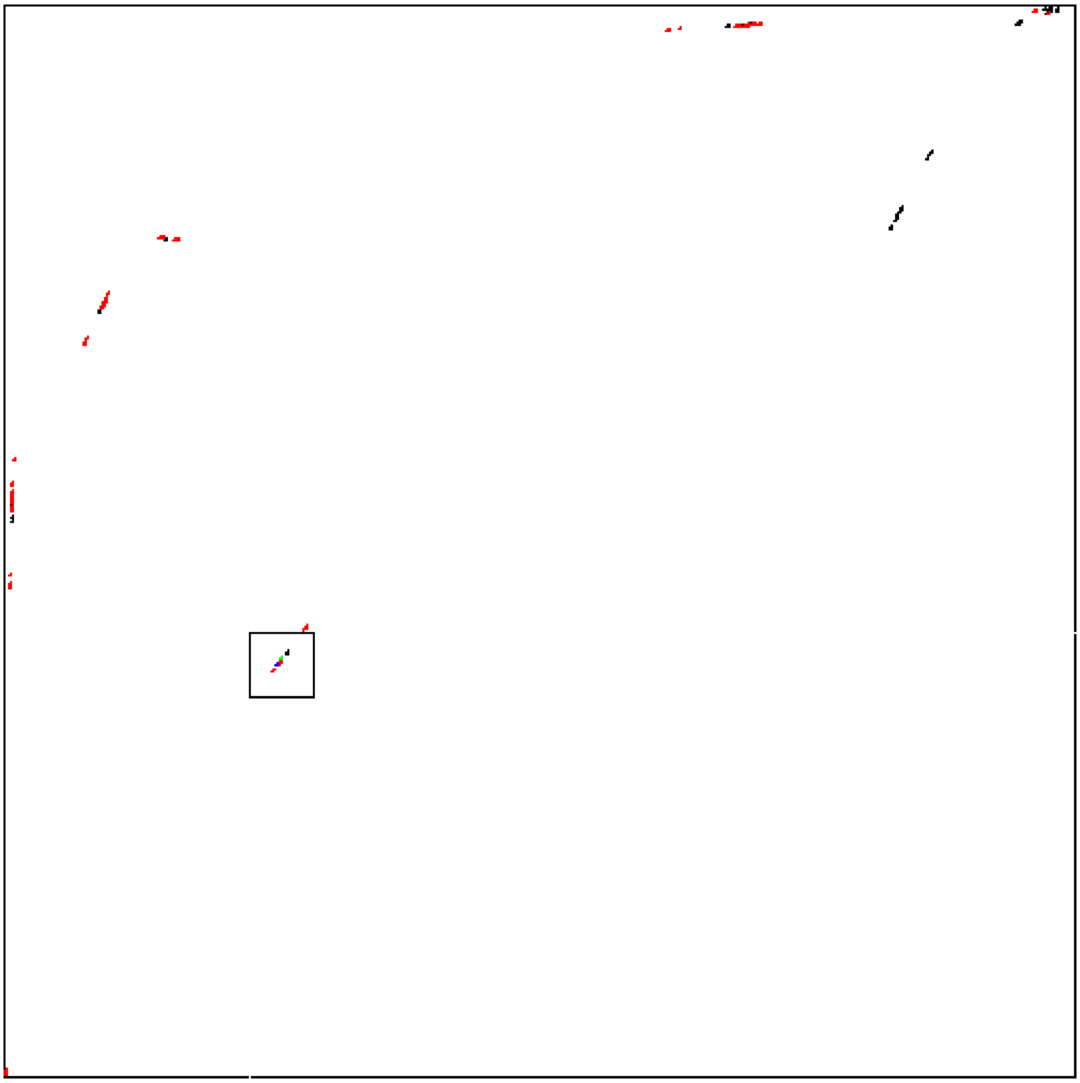


図 5.5: 未学習の環境における力学構造 (1)

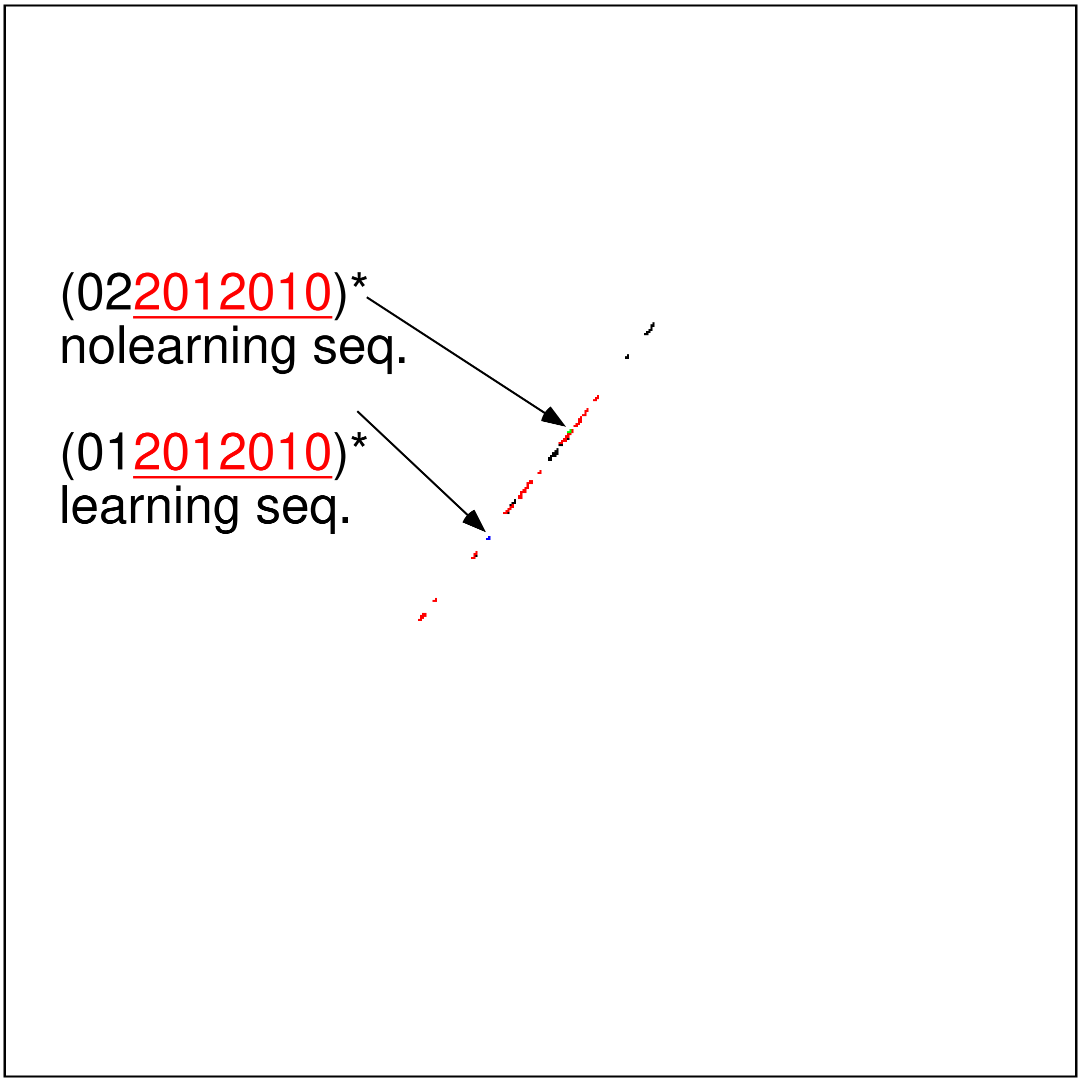


図 5.6: 未学習の環境における力学構造 (2)

5.1.3 追加学習の実験

リカレントニューラルネットワークは未学習の環境に対してもフラクタル構造を持つ力学構造の中で学習した環境とうまく区別している．未学習の状態に対応する状態が相空間上にセグメントとして形成されている．しかし，追加学習の実験においては未学習の環境に対応する状態が相空間において広い範囲に散在している．これは学習する環境に対する力学構造によるものである．未学習の環境に対する力学構造を解析した実験では，学習対象が有限状態機械で表現されるものであった．一方，追加学習の実験では，最初リミットサイクルを学習させて，その後有限状態機械を学習させている．有限状態機械を学習させた力学構造はその構造を保ったまま，多量の情報をしまい込む収容力に優れるため，未知の要素が入りこんでもある程度対象可能である．一方リミットサイクルの場合には，そういった無限の情報をしまい込む構造はなく，ロジカルに一意に定まってしまう構造のため，未知の要素が入り込むと不安定化するのである．

5.1.4 リカレントニューラルネットワークの学習

本研究では，力学系の視点からリカレントニューラルネットワークの学習の過程を解析した．リカレントニューラルネットワークにおける学習は，非線形力学系において系のパラメータを変化させることと同じである．非線形力学系においては，系のパラメータの変化がその系の力学構造の変化を生む．リカレントニューラルネットワークは学習対象に対応して適切に力学構造を変化させて，学習対象のモデルをその内部力学系のなかに表現することにより学習を行っているのである．リカレントニューラルネットワークの学習の過程は，非線形力学系の分岐現象と同じ過程であり，学習それ自体は複雑適応系なのである．

5.2 関連研究との比較

学習の意味付けが異なるが．本研究の他にもリカレントニューラルネットワークに有限状態機械を学習させて，それを力学系の視点から解析するという研究がいくつかなされている [10, 7, 11]．これらの研究では，学習対象である有限状態機械を正則文法として捉えている．正則文法をリカレントニューラルネットワークにある学習させた後に入力列を与えて，受理可能かどうか判断させたり，次の入力の予測をさせる．このような実験を行い力学系の視点からの解析を行っている．一方，本研究では，リカレントニューラルネットワークのエージェントが有限状態機械で表現される環境を学習するという想定で実験を行いその力学構造を解析している．

この 2 つのアプローチには，リカレントニューラルネットワークの情報処理の機構を解析する上で，本質的な違いはない．しかし，本研究ではいくつかの点で他の研究にない新しい方向からの取り組みを試みている．まず 1 つ目は，未学習の入力に対するリカレントニューラルネットワークの反応を解析したことである．有限状態機械を文法として捉えるアプローチでは，入力列を受理するかどうかだけが重要であり，未学習の入力に対してどう反応するかについては特に議論がなされていなかった．一方，本研究のアプローチでは，リカレントニューラルネットワークの学習をエージェントの学習の問題として捉えているため，エージェントが未学習の入力に対してどのような反応を示すかということは非常に重要な問題である．次に 2 つ目として，追加学習における力学構造の変化を解析したことである．文法の学習においては追加学習する必要性はほとんどない．しかし，エージェントの学習においては追加学習も重要な問題となる．

5.3 今後の課題

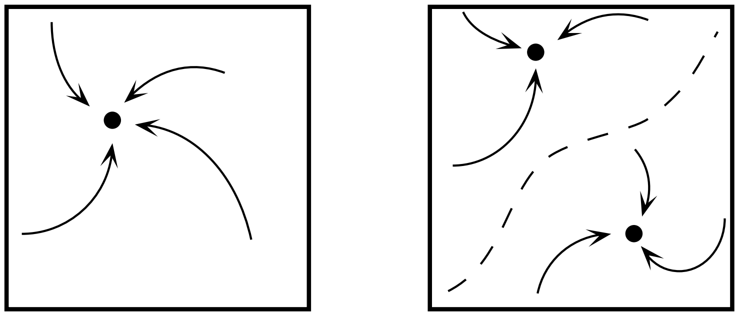
本節では，今後の課題についていくつか述べる．

5.3.1 複数の環境の学習

本研究の実験では，リカレントニューラルネットワークの学習対象は 1 つであった．追加学習の実験においても，追加した環境の中に，最初の環境が含まれているため，最初に学習した環境と追加して学習した環境を区別して学習する必要はない．しかし，リカレントニューラルネットワークを何らかの形で応用することを考えると，当然複数のものを学習する必要が生じる．エージェントのモデルとしてリカレントニューラルネットワークを用いる場合にも複数の環境を学習する場面は多いにあるだろう．本研究の実験では，学習結果がリカレントニューラルネットワークの大域的アトラクタとして表現されていた．これでは，複数のものを分けて学習することはできない．しかし，力学系の全てが大域的アトラクタを持つものばかりではなく，複数のアトラクタを持つ場合も考えられる．この場合，相空間には複数のベイスン[2]が存在し，系の初期状態によりトラジェクトリが最終的に引き込まれていくアトラクタが異なる (図 5.7)．このことは，リカレントニューラルネットワークに複数のものを学習させることが可能であることを示している．相空間を複数のベイシンに分割して，そこにそれぞれの学習対象に適した力学構造を形成することができれば良いのである．

リカレントニューラルネットワークは，力学系の中に複数の学習対象に適した力学構

[2]アトラクタの勢力範囲．



(a) (b)

図 5.7: ベイスン

造を形成できることが分かった．しかし，それをどのように学習させるかがまた大きな問題である．まずあるものを学習させて，その後新たなものを学習させようとすると前に学習した力学構造を破壊してしまい，うまく複数のものを学習することができない可能性がある．つまり，逐次的な方法では学習できないと考えられる．何らかの形で同時に学習させる必要がある．まず最初に考えられる方法としては，BPTT の学習に用いるサンプルデータに複数のものを混ぜて学習させる方法である．しかしこの方法ではそれぞれのデータが別々のものとして認識できないので，1 つものとして学習してしまう恐れがある．次に，本研究の実験で用いたリカレントニューラルネットワークを並列に学習させる方法を，複数の環境の学習に応用することが考えられる．この方法は，同時に学習しつつも学習対象を区別できる点で適切であるかもしれない．

5.3.2 次元の高いアトラクタ

リカレントニューラルネットワークに有限状態機械を学習させた場合，その力学構造はフラクタル構造を持ち，複数のセグメントから成る比較的低次元のアトラクタとなる．しかし時折，その力学構造が図 5.8 のような次元の高いアトラクタとなることがある．学習エラーは十分に減少しているので学習は完了していると考えてよい．ところがこのリカレントニューラルネットワークを動作させてみると，ごく稀にその出力にエラーを生じる．ある意味で完全に学習できていないとも考えられる．学習が完全にできてない場合，より頻繁にエラーを出すはずである．

この原因について考察してみる．図 4.6 と図 5.8 を比較すると，明らかにアトラクタの性質が異なる．アトラクタの性質が異なるということは，リカレントニューラルネットワークが力学系としての観点から異なる性質を持つということである．このことを IFS

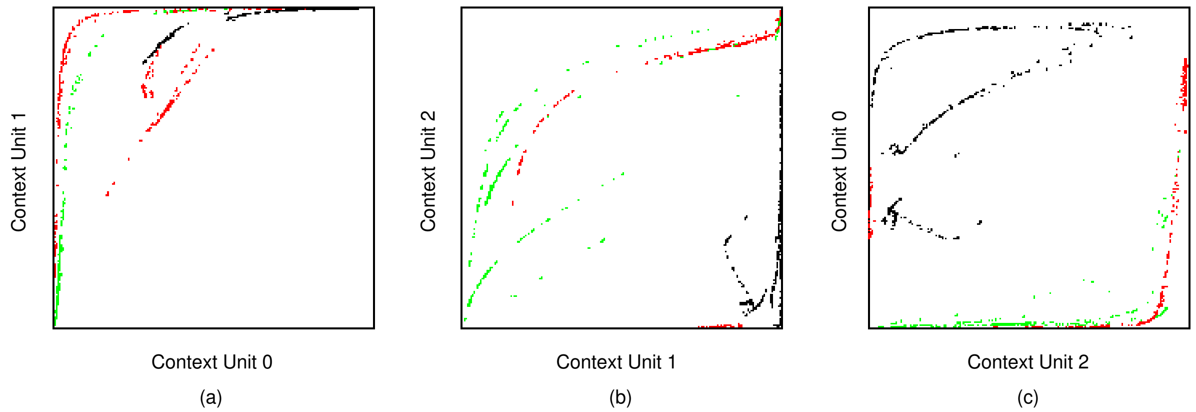


図 5.8: 次元の高いアトラクタ

との関連において考えてみる．IFS により生成されるフラクタル集合の各点は，一意に決まるアドレスが付けられることを説明した．アドレスに対して点は一意に決定されるが，点に対してアドレスが一意に決定されない場合がある．シェルピンスキーギャスケットの例で考えてみると，三角形同士が接する点には，複数のアドレスが付けられている．これを Barnsley は *just-touching* と呼んでいる．リカレントニューラルネットワークにおけるフラクタル構造のアドレスは過去の履歴を表しているので，フラクタル構造が *just-touching* となる場合過去の履歴を一意に特定できなくなり，リカレントニューラルネットワークは現在の状態を把握できないことになる．したがって，リカレントニューラルネットワークはその出力にエラーを生じると考えられる．Barnsley は，IFS を構成する複数の写像による写像先が接するだけでなく，さらに重なっているような場合を *overlapping* と呼ぶ．*just-touching* の場合，複数のアドレスを持つ点の全体に占める割合が少ないので，IFS より生成されるフラクタル構造が大きく崩れることはない．*overlapping* の場合，複数のアドレスを持つ点の割合が増加するので，生成されるフラクタル構造が大きく崩れる可能性がある．これが次元の高いアトラクタを生む原因の 1 つとして考えることができる．一方，低次元のアトラクタとなるのは，複数の写像による写像先が重ならない *totally disconnected* の場合であると考えられ，このとき各点には 1 つのアドレスだけがつけられる．したがって，全履歴を保持できることになり，リカレントニューラルネットワークは現在の状態を一意に把握できる．

IFS における *just-touching*，*overlapping* は決定論的力学系でありながら非決定的な過程を生む 1 つの要因と考えられる．リカレントニューラルネットワークでは，学習結果が非決定性有限状態機械として表現されてる可能性も考えられる．

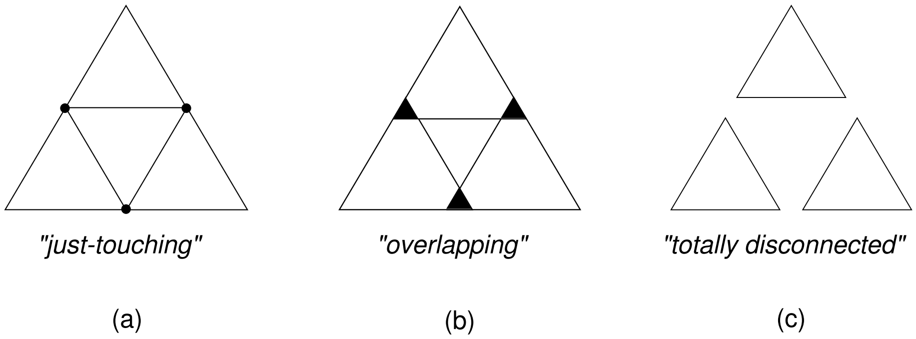


図 5.9: IFS の分類

5.4 力学系と認知

本節では，本研究の実験における力学系と認知について考察する．本研究では，力学系としてリカレントニューラルネットワークを用いた学習の実験を行った．そこで，このリカレントニューラルネットワークの学習において，力学系による認知の解釈を考えることができる．

リカレントニューラルネットワークの学習は，学習対象が生成する情報を適切に取り込み保持し，それに基づいて出力を決定できる力学構造を形成することである．この力学構造を学習対象の内部モデルとして考えることができる．この内部モデルは，力学系の時間発展として表現される．リカレントニューラルネットワークは力学系の時間発展にあたる動作を繰り返すうちに，学習対象に関する履歴情報を記憶として保持しつつ，それに基づいて出力を決定している．学習対象に関する履歴情報の記憶はリカレントニューラルネットワークが動作を続ける中で，自己組織化的に形成されるものである．履歴情報を保存するための操作を別に行っているわけではない．したがって，リカレントニューラルネットワークがその動作の中で，自律的に自分の行動に必要な情報を力学構造に埋め込んでいると考えることができる．

一方，人工知能の分野では記号的記述を用いた認知モデルが用いられてきた．このモデルでの学習は，学習対象が生成する情報を記号で表現して保持し，その記号を利用しやすいように編集することである．モデルの動作はこの保持された記号を，操作して出力を決定することにより行われる．このモデルにおいて，記憶の形成とその操作は別の処理として行われるのである．ここで忘れてはならないことは，学習とは別に保持された記号を操作する存在を考えなくてはいけないことである．この記号を操作する存在は設計者が作らなければならない．設計者が設計する以上のことをこれをモデルとするエー

ジェントに期待することはできないのである．したがって，このモデルに自律性を求めることはできない．

第 6 章

結論

本研究では，力学系の視点からリカレントニューラルネットワークについて説明した．そして，リカレントニューラルネットワークの学習の実験を行い，それを力学系の視点から解析した．

リカレントニューラルネットワークの学習過程は非線形力学系に特有の分岐現象によりその力学構造を変化させながら，学習対象のモデルをその内部力学系に力学構造として形成していく複雑適応系であり，実験により以下のことを確認した．

- リミットサイクルの学習においては，リカレントニューラルネットワークの力学構造がリミットサイクルのアトラクタとなった．
- 有限状態機械の学習においては，リカレントニューラルネットワークの力学構造が，フラクタル構造を持つアトラクタとなった．

この学習対象による力学構造の違いを記号力学系により解析し，リカレントニューラルネットワークは有限状態の学習に必要な履歴をフラクタル構造の中に埋め込んでいることが分かった．

次に，未学習の入力を入れた場合のリカレントニューラルネットワークの力学構造について解析し，リカレントニューラルネットワークの相空間において学習した入力に対するトラジェクトリのすぐ近傍に未学習の入力に対するトラジェクトリが存在していることを確認した．そして，記号力学系の解析により学習したものと未学習のものをフラクタル構造の中で区別していることを分かった．有限状態機械の学習により形成された力学構造は，多量の情報を保持できる構造を持ち，未学習の入力も受け入れることができることを確認した．

これらの実験結果より，リカレントニューラルネットワークの情報処理の機構は力学系としての性質に起因することが分かった．特に，フラクタル構造のアトラクタについて，Iterated Functions Systems との関係について考察した．

本研究で得られた知見は，リカレントニューラルネットワークを対象として考察および議論したが，力学系における情報処理に有益な示唆を与えるものである．さらに，本研究の最大の目的である力学系による認知の解釈に大きく貢献するものである．

謝辞

本研究の機会を与えられ，また日頃から懇切な御指導を頂きました慶應義塾大学理工学部教授 所 真理雄 博士に心から深く感謝致します．

本研究全般に渡り，多大なる御指導を頂きましたソニーコンピュータサイエンス研究所 谷 淳 博士に深く感謝致します．

本研究に貴重な助言を下さっただけでなく，様々な場面において適切かつユーモアのある御指導を頂いた本研究室博士課程 佐々木 貴宏 氏に深く感謝します．

お忙しい所を本論文の査読を引き受けて下さった．本研究室修士課程 1 年 池谷直紀 氏，修士課程 1 年 桑折 隆之 氏，修士課程 1 年 北脇 晴之 氏に心から感謝します．

また，貴重な助言をして頂いた本研究室博士課程 松林 慶 氏，修士課程 2 年中里 宗弘 氏，修士課程 2 年 蔀 秀久 氏，修士課程 1 年 久曽神 宏 氏，そして，共に研究をした本研究室学科 4 年生をはじめ，本研究室の諸氏に心から感謝します．

平成 9 年 2 月

Appendix A

Back Propagation Through Time (BPTT)

Back Propagation Through Time(BPTT) アルゴリズムの詳細を以下に示す．

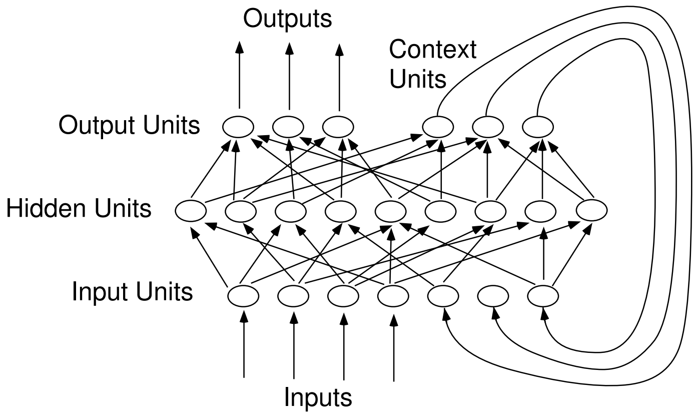


図 A.1: リカレントニューラルネットワーク

1. 各ユニット間の結合の重み w ，各ユニットの閾値 b ，$t = 0$ のコンテクストユニットの状態 s_0 をランダムな値で初期化する．

2. 入力 i と理想出力 (教師信号) y の組みを選び，入力を入力層に与える．(A.1) 式，(A.2) 式，(A.3) 式を順次用いて，隠れユニット h_i ，コンテクストユニット s_i ，出力ユニット o_i の各ユニットの出力を計算する．

$$h_i = f(\sum_j w^h_{i,j} s_{j,t} + b^h_i) + f(\sum_j w^h_{i',j} i_{j,t} + b^h_{i'}) \tag{A.1}$$

$$s_{i,t+1} = f(\sum_j w^s_{i,j} h_j + b^s_i) \tag{A.2}$$

$$o_{t+1} = f(\sum_j w^o_j h_j + b^o_i) \tag{A.3}$$

3. 2 の操作を時間ステップ T まで繰り返し，時刻 T までの各ユニットの出力を計算する．

4. 時刻 t の出力層のユニットにおける出力エラーを (A.4) 式を用いて計算する．

$$\delta^o_i = o_j(y_i - o_i) f'_i(net_o) \tag{A.4}$$

5. 時刻 t のコンテクスト層のユニット，隠れ層における出力エラーδ を (A.5) 式を用いて計算する．ここで，ν は減衰率を表す．

$$\delta_i = \nu f_i'(net_i) \sum_k w_{i,k} \tag{A.5}$$

6. 4，5 の操作を時刻 $t = T$ から時刻 $t = 0$ まで繰り返し，時刻 t の結合の重みの修正値を計算する．

7. 学習ステップ n での結合の重みの修正値 Δw を (A.6) 式を用いて計算する．ここで η は学習率， α は慣性係数を表す．

$$\Delta w_{i,j}(n+1) = \eta \delta_i a_j + \alpha \Delta w_{i,j}(n) \tag{A.6}$$

8. 時刻 $t = 0$ のコンテクストの状態の修正値 Δs_0 を (A.7) 式を用いて計算する．

$$\Delta s_{i,0}(n+1) = \eta \delta_i + \alpha \Delta s_{i,0}(n) \tag{A.7}$$

9. 8，9 で計算した修正値を基に各値を修正する．

10. 2 から 9 の操作を学習ステップ 1 として行う． (A.8) 式，(A.9) 式を用いて，学習エラーを計算してこれが十分小 さくなるまで繰り返す．

$$e_{i,t+1} = y_{i,t+1} - o_{i,t+1} \tag{A.8}$$

$$E = \frac{1}{2} \sum_{t=0}^{T} \sum_i (e_{i,t+1})^2 \tag{A.9}$$

参 考 文 献

[1] Jun Tani and Naohiro Fukumura. Embedding a grammatical description in deterministic chaos: an experiment in recurrent neural learning. *Biological Cybernetics*, 1995.

[2] Jun Tani. Model-based learning for mobile robot navigation from the dynamical ssystems perspective. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 1996.

[3] 谷　淳．ロボットにおける認知と自律性の構造：力学系の見地から．日本ロボット学会誌 *Vol.14 No.4*，1996．

[4] 合原　一幸．ニューラルシステムにおけるカオス．電機大出版局，1993．

[5] 津田　一郎．カオス的脳観．サイエンス社，1990．

[6] David E. Rumelhart, James L. McClelland, and the PDP Research Group. *PDP モデル: 認知科学とニューロン回路網の探索 (甘利俊一 (監訳))*. 産業図書 (1989), 1986. (原題) Parallel Distributed Processing: Explorations in the Microstructure of Cognition.(MIT Press).

[7] Jeffrey L. Elman. Finding structure in time. *Cognitive Science 14*, 1990.

[8] Williams Ronald J. Real-time reccurent learning algorithm. In 脳と学習のメカニズム, 丸善, 1992.

[9] Michael Barnsley. *FRACTALS EVERYWHERE*. ACADEMIC PRESS, 1988.

[10] Jordan B. Pollack. The induction of dynamical recognizers. *Machine Learning*, 1991.

[11] John F. Kolen. *Exploring the Computational Capabilities of Recurrent Neural Networks*. PhD thesis, The Ohio State University, 1994.

[12] Stevan Harnad. The symbol grounding problem. *Physica D 42*, 1990.

[13] James P. Crutchfield and Karl Young. Inferring statistical complexity. *PHYSICAL REVIEW LETTERS*, 1989.

[14] 合原　一幸．カオス - カオス理論の基礎と応用．サイエンス社，1990．

[15] 合原　一幸．応用 カオス．サイエンス社，1994．

[16] Denny Gulick. カオスとの遭遇. 産業図書, 1995.

[17] 長島　弘幸．カオス入門．培風館，1992．

[18] R. L. Devaney. カオス力学系入門. 共立出版, 1990.

[19] Edwin Altee Jackson. 非線形力学の展望 *I* カオスとゆらぎ. 共立出版, 1991.

[20] 吉永　良正．複雑系とは何か．講談社，1997．

[21] Tamas Vicsek. フラクタル成長現象. 朝倉書店, 1989.

[22] 山口　昌哉，畑　政義，木上　淳．フラクタルの数理．岩波書店，1993．

[23] 高安　秀樹，高安　美佐子．フラクタルって何だろう．ダイヤモンド社，1988．

[24] Randall D. Beer. A dyanmical systems perspective on agent-environment interation. 1990.

[25] 安西　祐一郎，出村　公成．時系列を学習するニューラルネットワーク．，脳とニューラルネットワーク，朝倉書店，1994．

[26] 橋田　浩一，安西　祐一郎．認知科学の基礎．岩波書店，1995．

[27] 市川　神一，伊東　祐司．記憶と学習．岩波書店，1995．

[28] 下條　信輔．サブリミナル・マインド - 潜在的人間観のゆくえ．中公新書，1996．