

進化的計算の動的問題への適用法と
その応用に関する研究

佐藤 俊二



①

進化的計算の動的問題への適用法と
その応用に関する研究

佐藤 裕二

要 旨

従来の情報処理は閉鎖型システムに力点が置かれている。すなわち、問題の領域を定義し、切り取り、抽象化することにより問題を扱うことが一般的である。一方、実世界の問題、例えば社会、経済現象や生命の問題は定義しきれない要素や切り取ることでできない問題領域を含んでいる。また、広域分散、移動コンピューティング環境やマルチメディア環境では、利用環境が時々刻々と変化している中で安定した性能を供給する必要がある。このような状況を考え本研究の目的は、学習過程における部分的な情報を基に有効な処理を行うシステムの実現およびこのようなシステム実現に有効と思われる技術の開発を行うことにある。ここでは、近年、実用上の準最適解を速やかに得ることのできる適応範囲の広い探索アルゴリズムとして注目されている進化的計算に着目し、その動的問題への適用法と応用に関する基礎的な検討を行う。特に、進化的計算とニューラルネットワークの融合によるアプローチを中心に論じる。

第1に、進化的計算自体が新しい技術であるため、現状の技術に関する整理を行う。すなわち、進化的計算の概要および進化的計算を用いたニューラルネットワークの合成に関する従来技術に関してまとめる。第2に、ニューラルネットワークの合成に有効と思われる、表現型レベルで遺伝的操作を施す手法について述べる。すなわち、ニューラルネットワーク自体を2次元の染色体と考え、ニューラルネットワーク上の2次元部分構造に着目した交差を提案する。これは、進化的計算をニューラルネットワークに適用する従来手法が、基本的に1次元染色体上でのランダムな位置での交差を仮定しているために、ニューラルネットワーク上の2次元スキーマを交差で破壊する確率が高いと考えるからである。すなわち、進化的計算では交差によりスキーマと呼ばれる部分構造を的確に組み合わせて効率的な探索を実現する可能性を有するが、このような部分構造は染色体の表現形式と交差の単位に依存することが多く、進化的計算をニューラルネットワークに適用する場合に効果的な染色体の表現形式と交差の単位の検討が重要と考えるためである。また、1次元染色体上で標準的な遺伝的操作を施した場合との比較から、その有効性を示す。第3に、行動を決定するために十分な情報が得られない状況下での自律適応システム行動制御に有効と思われる、2つのリカレント型ニューラルネットワークが相互作用を持ちながら学習を行うモデルを提案するとともに、このモデルの学習に進化的計算を適用する基礎的な実験について述べる。進化的計算を用いたリカレントニューラルネットワークの生成自体が、世界的に見ても、まだ始まったばかりの段階である。従って、2つのリカレントニューラルネットワークが相互作用を持ちながら学習を行うモデルに進化的計算を適用することは今までにない新しい試みと考える。第4に、動的で複雑な問題の一例としてカオスダイナミックスの学習を取り上げ、既に著者らが開発した専用のニューロコンピュータ上に実装した高速シミュレータによる学習と進化的計算により生

成したりカレント型ニューラルネットワークによる学習の比較を行う。教師信号として用いたローレンツ軌道は常にカオス的に変化し続ける非周期軌道である。したがって、教師信号の波形そのものの学習が完了することはない。ストレンジアトラクタを形成するというカオスの隠れた特徴の学習を試みる。最後に、実世界における動的問題の応用例として音声情報処理を取り上げ、進化的計算の適応能力を調べる基礎的な実験を行う。すなわち、ピッチ（声の高低情報）、パワー、時間構造などの韻律情報を進化的計算により変換することにより、声質の変換が自然に行える可能性があることを示す。

これらの研究成果により、動的で複雑な問題を対象とした進化的計算の適用可能性を実証し、動的な問題の応用拡大を考える上で有効な結果を示すことができた。また、これらの技術は、今後飛躍的な発展が予想されるマルチメディア技術の柔軟なユーザインタフェース実現にも応用することができると考える。例えば、ユーザ自身が自分の好みの声質に調整可能な音声ユーザインタフェースを持つマルチメディアPCなどに応用することができると考えられる。

目次

第1章 序論	1
1.1 研究の背景と動機	1
1.1.1 閉鎖型システムの限界	1
1.1.2 開放型システムへの期待	3
1.2 研究の目的	4
1.2.1 最適解が確率的に変化する離散的空間の問題への適用	4
1.2.2 連続実数空間の動的問題への適用	4
1.2.3 音声情報処理への適用	5
1.3 論文の構成	6
第2章 進化的計算を用いたニューラルネットワーク合成に関する従来手法	9
2.1 はじめに	10
2.2 進化的計算の概要	11
2.2.1 遺伝的操作	11
2.2.2 選択・交配	13
2.2.3 交差	14
2.2.4 突然変異	15
2.3 スキーマ定理と積木仮説	16
2.4 フィードフォワードニューラルネットワークの合成	17
2.4.1 進化的計算を用いた結合荷重値の学習	17
2.4.2 進化的計算とBPのハイブリッド化による結合荷重値の学習	17
2.4.3 進化的計算を用いたニューラルネットの構造決定	19
2.4.4 進化的計算を用いた構造決定と結合荷重値の学習	19
2.5 リカレントニューラルネットワークの合成	22
2.6 まとめ	23
第3章 表現型レベルの遺伝的操作を用いたニューラルネットワークの合成	24
3.1 はじめに	25
3.2 表現型レベルの遺伝的操作を用いたニューラルネットワーク合成の提案	26
3.3 スキーマ定理と表現型レベルでの遺伝的操作	29
3.4 遷移行列を用いた遺伝的操作	31
3.5 標準GAとの比較評価実験	34
3.5.1 評価用問題	34
3.5.2 ニューラルネットワークの初期構造	35
3.5.3 遺伝的操作の設定	35

3.5.4 実験結果	37
3.5.5 考察	37
3.6 適応度依存並列化の提案	47
3.6.1 適応度依存並列GAの提案	47
3.6.2 SIMD型並列計算機への実装案	51
3.7 まとめ	54
第4章 自律システム制御のための先読みモデルの進化的獲得	55
4.1 はじめに	56
4.2 先読みモデルの提案	57
4.2.1 基本とするリカレントネットワークの特徴	57
4.2.2 行動を決定するネットワークと内部モデルの相関学習	57
4.3 進化的計算を用いた先読みモデルの学習法	59
4.3.1 行動を決定するネットワークの学習	59
4.3.2 内部モデルの学習	61
4.4 Tic-Tac-Toeゲームを用いた評価	63
4.4.1 実験方法	63
4.4.2 ネットワークの初期設定とGAの適用方法	65
4.4.3 実験結果	66
4.5 考察	69
4.6 まとめ	71
第5章 カオスダイナミクスを用いた進化的計算の適応力評価実験	72
5.1 はじめに	73
5.2 ローレンツアトラクタ	75
5.3 非周期軌道の学習方法	78
5.4 進化的アルゴリズムによるニューラルネットワークの合成	79
5.4.1 染色体の定義	79
5.4.2 適応度の計算	79
5.4.3 遺伝的操作	80
5.5 計算機実験	87
5.6 考察	90
5.7 まとめ	93
5.8 付録：リカレントニューラルネットワーク高速シミュレータの開発	94
5.8.1 はじめに	94
5.8.2 システム概要	95
5.8.3 学習アルゴリズム	97
5.8.4 計算機実験	99
5.8.5 まとめ	103

第6章 進化的計算を用いた韻律係数フィッティングによる声質変換に関して ...	109
6.1 はじめに	110
6.2 音声の構成要素と声質変換	111
6.2.1 音声の構成要素と特性	111
6.2.2 韻律調整による声質変換	111
6.3 進化的計算を用いた韻律係数フィッティング	114
6.3.1 声質変換システム構成概略	114
6.3.2 進化的計算の適用法	114
6.3.3 遺伝的操作	116
6.4 評価実験	119
6.4.1 音声刺激	119
6.4.2 実験方法	119
6.4.3 実験結果および考察	119
6.5 まとめ	122
第7章 結論	123
7.1 研究成果	123
7.1.1 研究成果の概要	123
7.1.2 進化的計算のニューラルネットワークへの適用法の提案とその有効性の 実証	124
7.1.3 先読みモデルの提案とその有効性の実証	124
7.1.4 離散的空間の動的問題を対象とした進化的計算の適用可能性の実証...	125
7.1.5 連続実数空間上の動的問題を対象とした進化的計算の適用可能性の 実証	125
7.1.6 進化的計算を用いた声質変換技法の開発	125
7.2 結論	126
7.3 今後の課題	127
7.3.1 進化的計算のニューラルネットワークへの適用法	127
7.3.2 先読みモデルの有効性の実証	127
7.3.3 離散的空間の動的問題を対象とした進化的計算の適用可能性の実証...	127
7.3.4 連続実数空間上の動的問題を対象とした進化的計算の適用可能性の 実証	128
7.3.5 進化的計算を用いた声質変換技法の開発	129
7.3.6 他の課題とまとめ	129
謝辞	130
参考文献	131

付録	138
著者の主な出版物	151
章と論文の対応	155

第1章 序 論

この章では、本論文における研究の背景と動機、研究の目的、論文の構成について順にのべる。

1.1 研究の背景と動機

この研究を開始した背景と動機は閉鎖型システムの限界と開放型システムへの期待にまとめられる。これらの点について、以下説明する。

1.1.1 閉鎖型システムの限界

従来の情報処理の状況を図1.1に示す。従来の情報処理は閉鎖型システムに力点が置かれている。すなわち、問題の領域を定義し、切り取り、抽象化することにより問題を扱うことが一般的である。これは、WinogradやSchankなどに代表されるトイワールドでシステムを作り、後で実世界に向けたスケールアップを考えるアプローチと捕えることができる。ここでは、いくつかの理想化された前提が暗黙の了解として置かれている。即ち、以下に示す前提である。

- ・問題を解決するために十分な知識（アルゴリズム）を予め人間が獲得できる。
- ・それらの知識は（シンボルとして）表現可能である。
- ・問題が扱うデータは完全なデータ（情報の欠落がない、一貫性がある、誤りを含まない）である。

上記アプローチは、多くの工学的応用分野において成功を収めている。例えば、解法のためのアルゴリズムが存在しプログラム言語で明確に記述可能な良設定問題は、コンピュータを使って人間よりもはるかに高速かつ正確に解くことが可能となった。また、航空会社の乗員スケジュールや注文住宅の組立計画などでは、エキスパートシステムが普及している。乗員スケジュールでは、すべての乗員、路線、機材は完全に記号化されていて、スケジュール規則や制約も完全に記述可能であるなど、上記前提が満たされるためである。

一方、実世界の問題では必ずしも上記前提がすべて成立するとは限らない。例えば社会、経済現象や生命の問題は定義しきれない要素や切り取ることでできない問題領域を含んでいる。また、広域分散、移動コンピューティング環境やマル

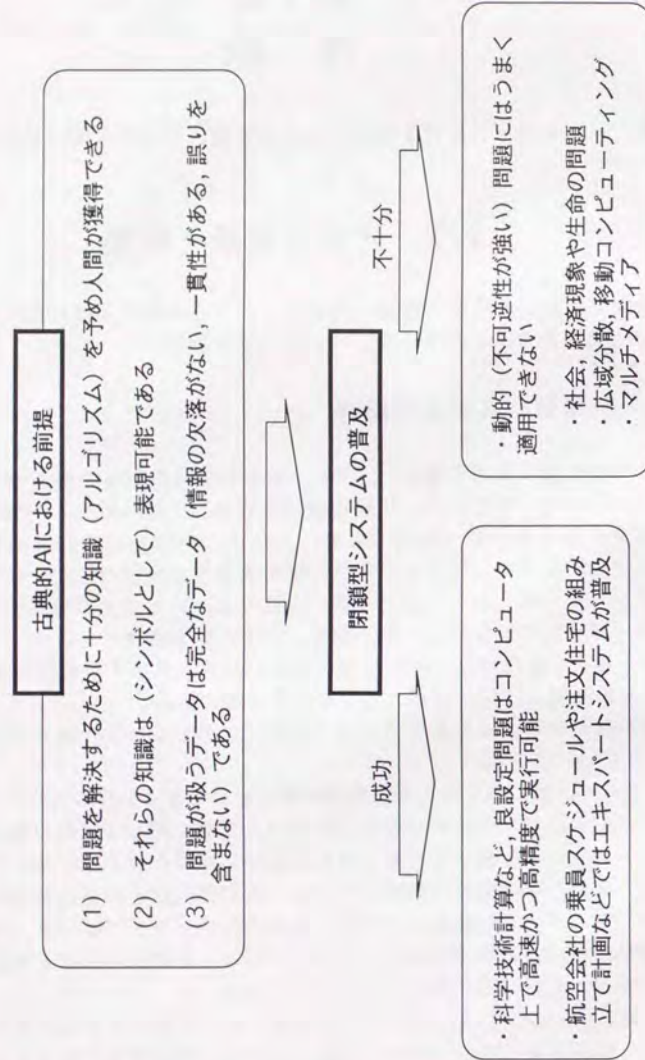


図1.1 従来の情報処理の状況

マルチメディア環境では、利用環境が時々刻々と変化している中で安定した性能を供給する必要がある。このような問題に対して上記アプローチを適用する場合、極めて限定された領域で非常に複雑な処理を行い、それが一定の成功を収めた後で、領域を次第に広げていくという手法が取られる。例えば、WinogradのSHRDLU [Winograd 72]やYale大学時代のRoger Schank一派の研究のように、限定された世界での言語理解を達成したら、その方法を実世界に応用して広い領域に関する自由な自然言語入力を理解するシステムを作るという手法である。このような手法は、現在広く一般的に用いられている手法である。しかし、限定された世界から実世界に移行する際のスケールアップの難しさ、および、情報の不確定性の劇的増大の問題を考えると、理想化された世界で動作するシステムが実世界でも動作する保証はない。むしろ、ほとんどうまくいっていないのが実情である。実世界では必ずしも成立しない前提から出発しているために、対応できる問題領域が限られてしまうためと考えられる。

1.1.2 開放型システムへの期待

上記閉鎖型システムは一部の問題領域に対しては成果を残したと考えられる。一方、近年急速に発展しているマルチメディア分野では、定義しきれない問題、切り取ることでできない問題を解かなければならない状況に至っており、上記閉鎖型システムだけでは十分に対応できなくなっている。そこで、閉鎖型システムに対立する概念として開放型システムの必要性が望まれ始めている。上記閉鎖型システムの大前提を否定して問題を解決するアプローチと捉えることができる。例えば、時々刻々とネットワークテクノロジーが変化し、接続されるコンピュータやそのサービスが変わる環境の下で、マルチメディア情報ベースにダイナミックにアクセスしながら仕事を行う環境を考えると、仮に個々の要素システムについて十分な知識を持っていたとしても全体の振る舞いは予測不能であり、開放型システムと捉えることができる。常に変化する環境に対して安定した性能を保証するシステム設計開発は今後急速にニーズが高まると予想され、そのための方法論や技術を確立していく必要があると考えられる。また、開放型システム実現のための方法論や要素技術を確立することにより、今までにない新しい応用分野を開き開くことができる可能性があると考えられる。

1.2 研究の目的

従来の科学技術は閉鎖型システムに力点が置かれている。すなわち、問題の領域を定義し、切り取り、抽象化することにより問題を扱うことが一般的である。一方、実世界の問題、例えば社会、経済現象や生命の問題は定義しきれない要素や切り取ることでできない問題領域を含んでいる。また、広域分散、移動コンピューティング環境やマルチメディア環境では、利用環境が時々刻々と変化している中で安定した性能を供給する必要がある。このような状況を考え本研究の目的は、学習過程における部分的な情報を基に有効な処理を行うシステムの実現およびこのようなシステム実現に有効と思われる技術の開発を行うことにある。ここでは、近年、実用上の準最適解を速やかに得ることのできる適応範囲の広い探索アルゴリズムとして注目されている進化的計算に着目し、その動的問題への適用法と応用に関する基礎的な検討を行う。特に、進化的計算とニューラルネットワークの融合により、最適解が確率的に変化する離散的空間の問題および連続実数空間の動的問題の学習能力を探索する基礎的な実験を試みる。

1.2.1 最適解が確率的に変化する離散的空間の問題への適用

この研究により、最適解が確率的に変化する離散的空間の問題に対して、進化的計算とニューラルネットワークの融合モデルの持つ学習能力を探索することを目的とする。一般的に、進化的計算は多数の局所解が存在する離散的な問題に対して有効であることが知られている。突然変異により局所解からの脱出が可能で、および、複数の個体を用いた多点探索を行う機能があるためである。ここではさらに、例えば時系列性や複数の局所解の中で最適解が確率的に変動する等の動的条件が加わった場合に対しても有効なアルゴリズムを提案することを目的としている。特に、単純に進化的計算を適用するよりも、進化的計算とニューラルネットワークの融合により学習が効率的に行える可能性があることを示すことを目的としている。また、本研究を通して新しい応用技術の発展に寄与することを目指す。当面は大規模な応用問題への適用は考えず、基礎技術の確立を優先して考える。

1.2.2 連続実数空間の動的問題への適用

1.2.1節で述べた離散的な問題に留まらず、この研究は、連続実数空間の動的に変化する問題に対して、進化的計算とニューラルネットワークの融合モデルの

持つ学習能力を探索することを目的とする。一般的に、進化的計算は連続実数空間上の局所的な探索には不向きであると言われている。探索の方向が明確に決定される最急降下法などの学習アルゴリズムとは異なり、基本的にランダムサーチの一種と考えられるために、局所的な探索能力に弱いと考えられるからである。一方、進化的計算には、(i) 最適解の周辺に早く近づく、(ii) 初期値に比較的左右されない、(iii) 目的関数の性質が良く分からない問題にも適用できる、という利点がある。この研究では、進化的計算とニューラルネットワークの融合により、これら進化的計算の特徴をニューラルネットワークの学習にうまく取り込むことを目的としている。これは表1.1に示すように、進化的計算とニューラルネットワークがお互いの弱点を補いあう関係にあるためである。すなわち、進化的計算とニューラルネットワークをうまく融合することができれば、両者の利点を兼ね備え、連続実数空間の動的問題にも対応可能な強力なアルゴリズムを実現できる可能性があると考えられるからである。

表1.1 進化的計算とニューラルネットワークの比較

学習能力に関連する項目	進化的計算	ニューラルネット
最適解の近傍に収束する早さ	○	△
初期値依存性	○	×
強化学習	○	△
ネットワーク構造の自動獲得	○	×
局所探索能力	×	○
非線形問題への適応力	△	○
動的問題への適応力	△	○

1.2.3 音声情報処理への適用

1.2.1節および1.2.2節では、意図的に設定した問題を用いて進化的計算の動的な問題への適応力を探索することを試みる。一方ここでは、具体的な応用例を用いて進化的計算の動的問題への適応力を探索することを目的とする。特に、マルチメディア時代への移行が急速に進んでいることを考慮して、音声情報処理を実世界の問題の一例として取り上げる。すなわち、ピッチ（声の高低情報）、パワー、時間構造などの韻律情報を進化的計算により変換することにより、声質の変換が自然に行える可能性があることの実証を目的とする。

1.3 論文の構成

本論文では、学習過程の部分的な情報を基に有効な処理を行うシステムを実現するために有効と思われる基礎技術について提案と評価を行う。特に、進化的計算とニューラルネットワークの融合により、動的に変化する問題に対する学習能力を高める基礎的な技術に関する提案と評価を中心的に行う。

まず第2章で、進化的計算自体が新しい技術であるため、現状の技術に関する整理を行う。すなわち、進化的計算の概説を行うとともに、進化的計算を用いたフィードフォワード型ニューラルネットワークの生成およびリカレント型ニューラルネットワークの生成に関する従来技術に関してまとめる。進化的計算をニューラルネットワークに適用する場合、以下に示す4種類のアプローチが考えられる。第1の方法は、ニューラルネットワークの構造は人間が与え、ニューロン間の結合荷重値の学習に対して進化的計算を用いる方法である。第2の方法は、ニューラルネットワークの学習アルゴリズムは人間が与え、進化的計算によってニューラルネットワークの構造を決定する方式である。第3の方法は、ニューロン間の結合荷重値の学習に対して進化的計算とBPなどの学習アルゴリズムとのハイブリッド化を考える方法である。そして第4の方法は、ニューラルネットワークの構造決定と各ニューロン間の結合荷重値の大きさの決定の両方に対して進化的計算を適用する方法である。第2章では、進化的計算をニューラルネットワークに適用する場合の、これら4種類の方法的な研究例を紹介する。また、従来の方法の問題点について示す。

第3章では、ニューラルネットワークの合成に有効と思われる、表現型レベルで遺伝的操作を施す手法について述べる。すなわち、ニューラルネットワーク自体を2次元の染色体と考え、ニューラルネットワーク上の2次元部分構造に着目した交差を提案する。これは、進化的計算をニューラルネットワークに適用する従来手法が、基本的に1次元染色体上でのランダムな位置での交差を仮定しているために、ニューラルネットワーク上の2次元スキーマを交差で破壊してしまう確率が高いと考えるからである。すなわち、進化的計算では交差によりスキーマと呼ばれる部分構造を的確に組み合わせて効率的な探索を実現する可能性を有するが、このような部分構造は染色体の表現形式と交差の単位に依存することが多く、進化的計算をニューラルネットワークに適用する場合に効果的な染色体の表現形式と交差の単位を提案する。また、1次元染色体上で標準的な遺伝的操作を施した場合との比較から、その有効性を示す。さらに、環境との情報のやり取りで得られる適応度に依存して、遺伝的操作の並列度を動的に変化させるアイデアおよびSIMD (Single Instruction stream and Multiple Data stream)型並列計算機上への実装案について述べる。

第4章では、行動を決定するために十分な情報が与えられない状況下での自律適応システム行動制御を考える。実世界における多くの問題では、適応システムや機械学習で必要となる試行回数や学習時間が十分に与えられるとは限らず、従って、学習過程における部分的な情報を基に有効な行動を行う必要があると考えるからである。ここでは、このような部分的な情報を基に有効な行動を必要とする問題に対して、2つのリカレント型ニューラルネットワークが相互作用を持ちながら学習を行うモデルを提案するとともに、このモデルの学習に進化的計算を適用する基礎的な実験について述べる。進化的計算を用いたリカレントニューラルネットワーク合成の研究自体が、世界的に見ても、まだ始まったばかりの段階である。従って、2つのリカレントニューラルネットワークが相互作用を持ちながら学習を行うモデルに進化的計算を適用することは今までにない新しい試みと考える。

第5章では、動的で複雑な問題の一例としてカオスダイナミックスの学習を取り上げ、既に著者らが開発したニューロコンピュータ上に構築した高速シミュレータによる学習と進化的計算により生成したリカレント型ニューラルネットワークによる学習の比較を行う。教師信号として用いたローレンツ軌道は常にカオス的に変化し続ける非周期軌道である。したがって、教師信号の波形そのものの学習が完了することはない。ストレンジアトラクタを形成するというカオスの隠れた特徴の学習を試みる。結果として、動的で複雑な問題を対象とするリカレントニューラルネットワーク生成への進化的計算の適用可能性を実証し、自律適応システムの行動制御を考える上で有効な実験結果を示すことができたと考える。

また、付録として、大規模リカレントニューラルネットワークを高速に学習するために開発した高速シミュレータシステムについて述べる。進化的計算を用いてニューラルネットワークを合成するというソフト的なアプローチとともに、ニューロコンピュータや超並列計算機などの力を借りたハード的なアプローチにより動的かつ複雑な問題解決に取り組むことも重要と考えられること、また、動的かつ複雑な問題を対象として進化的計算の有効性を高める際の比較実験とすることが目的である。既に著者らが開発した専用のニューロコンピュータ上に実装したために、SUN Sparc10などのワークステーション上のソフトシミュレータと比較して数10~100倍程度的高速学習が可能である。学習アルゴリズムは高速性を考慮してBPTT (Back Propagation Through Time)アルゴリズムを採用した。最大512物理ニューロンまで対応可能である。

第6章では、実世界における動的問題の応用例として音声情報処理を取り上げ、進化的計算の適応能力を高める基礎的な実験を行う。すなわち、ビッチ (声の高低情報)、パワー、時間構造などの韻律情報を進化的計算により変換することによ

り、声質の変換が自然に行える可能性があることを示す。声質変換に関する研究は既に幾つか報告されているが、「明瞭な」などの限られた感性表現語を目的とする変換に関して定性的な変換ルールが分かっている程度である。すなわち、任意の感性表現語に対して具体的にどのような値に韻律情報の変換を行うかは全く不明であり、その設定は試行錯誤による困難な作業である。ここでは、従来研究から特に明示的なルールが得られず、かつ、動的な情報処理である声質変換を取り上げ、進化的計算の適用方法および適用した結果について報告する。

最後に、第7章で上記全ての研究成果をまとめる。

尚、この論文の第1章は佐藤ら[Sato 95b]を参照してあらたに記述した。第2章は佐藤[Sato 95a]を下敷きにし、北野[Kitano 93]および安居院ら[Agui 93]を参照して記述した。第3章は佐藤ら[Sato 95a, Sato 95e]および佐藤[Sato 95d]を参照して記述した。また、第3章の内容をもとにした論文を投稿中[Sato 97b]である。第4章は佐藤ら[Sato 95b, Sato 95c]を下敷きにし、佐藤ら[Sato 93b, Sato 94a, Sato 94b, Sato 94c, Sato 94d]を参照して記述した。第5章は佐藤[Sato 96d]および佐藤ら[Sato 96a]を下敷きにし、佐藤ら[Sato 93a, Sato 95f, Sato 96b, Sato 96c]を参照して記述した。第6章は佐藤ら[Sato 96e, Sato 97a]をもとに記述した。また、あらたに記述した部分も含めて、各部分を日立製作所中央研究所の研究報告書に記載している。

第2章 進化的計算を用いたニューラルネットワーク 合成に関する従来手法

要旨

本章では、進化的計算の概要および進化的計算を用いたニューラルネットワークの合成に関する従来研究のまとめを行う。進化的計算は、生物進化から着想を得た確率的探索手法であり、従来手法では解決が困難であったさまざまな最適化問題や探索問題に対して、実用上の準最適解を速やかに得ることができる適応範囲の広い多点探索アルゴリズムとして注目され始めている。特に、ニューラルネットワークとの融合により、お互いの弱点を補う効果的な探索を行える可能性がある。すなわち、初期値依存性が比較的少なく、かつ、局所解の近傍に素早く収束する進化的計算の利点と、非線形問題や時系列問題に有効なニューラルネットワークの利点を統合した強力な探索モデルを実現する可能性がある。ここでは、進化的計算を利用したニューラルネットワークの合成に関する代表的な従来研究に関してまとめる。また、これら従来研究の持つ問題点を示す。

2.1 はじめに

Evolutionary Computation (以下、進化的計算と称す) [Fogel 94a]は、Genetic Algorithms (GA) [Holland 75], Evolutionary Programming (EP) [Fogel 66], Evolution Strategies (ES) [Schwefel 65], Genetic Programming (GP) [Koza 92]の4つの研究分野を総称して、1994年にIEEEが命名した名称である。いずれも生物進化から着想を得た確率的探索手法であり、従来手法では解決が困難であったさまざまな最適化問題や探索問題に対して、実用上の準最適解を速やかに得ることができる適応範囲の広い多点探索アルゴリズムとして注目され始めている。歴史的には、図2.1に示すように、GAはHolland、EPはFogel、ESはSchwefelらによって、1960年代後半から、それぞれ独立に発展してきた。GPは、Hollandの弟子の一人であるKozaが、GAをLISPプログラムの自動合成用に拡張したアルゴリズムである。研究の歴史は意外と古いが、これらの研究が一般に知られるようになったのは最近のことである。GAの研究結果が報告されるようになったのは、1985年に開始されたInter. Conf. on Genetic Algorithms (ICGA)からであり、一般に知られるようになったのは、[Goldberg 89]が刊行された1989年以降のことである。EP、ES、GPは、1994年に開始されたIEEE Inter. Conf. on Evolutionary Computation (ICEC)以降になって、にわかに注目を浴びるようになってきた。以下の節では、進化的計算の中で一般に最も普及しているGAの概説を行う。GAと他のアルゴリズムとの違いに関しては文献[Fogel 94, Koza 92]を参照されたい。また、進化的計算を用いたニューラルネットワークの合成に関する代表的な従来研究を紹介する。進化的計算とニューラルネットワークとの融合により、お互いの弱点を補う効果的な探索を行える可能性があるためである。尚、特に断わりのない限り、以下の章ではGAと進化的計算の区別を特に行わずに使う。

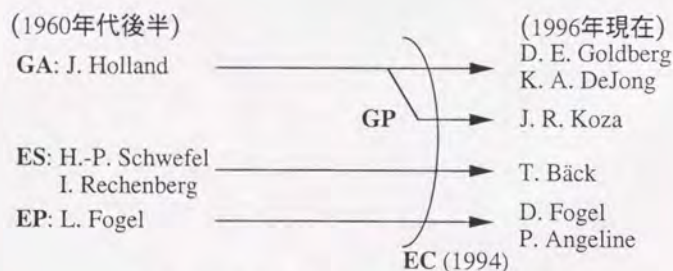


図2.1 進化的計算の派閥と代表者

2.2 進化的計算の概要

2.2.1 遺伝的操作

図2.2に遺伝的操作の基本的なフローチャートを示す。GAでは、まず染色体(chromosome)の定義を行う。すなわち、対象とする探索問題を、最適な染色体を求める問題に置き換える。染色体は、図2.3に示すように、一般的には複数の遺伝子(gene)から構成される。各々の個体の染色体の内部的表現を遺伝子型(genotype)という。遺伝子型を表わす記号列は任意であるが、一般的に、GAでは0または1の二値表現を用いる場合が多い。一方、EP、ESでは実数値表現を用いる場合が多い。次に、以下の処理手順により最適解に対応する染色体を求める。

1. 初期集団の生成
2. 終了条件が満たされるまで以下のループを繰り返す
 - (a) 適応度の評価
 - (b) 選択
 - (c) 交差
 - (d) 突然変異

まず、初期集団の生成を行う。一般的には、決められた個体数の染色体をランダムに生成する。ここで、個体数の決定や染色体の長さ、コーディングの方法は重要な研究課題であるが、現在は、経験と試行錯誤により決定しているのが実状である。初期集団の生成が終わると、次に各々の個体に対して適応度の評価を行う。一般的には、解こうとする問題を考慮したうえで、設計者が評価関数を明示的に与えて適応度の評価を行う。各々の個体に対して適応度が決まると、この適応度を基に選択・交配を行う。基本的には、適応度の高い個体により多くの子孫を残す機構となるため、より良い個体を形成する遺伝子が集団中に広がる。選択・交配を行う個体対が決定されると、染色体の交差を行う。交差の方法も複数提案されているが、基本的には、双方の染色体の一部ずつを採ってきて子孫の染色体を作る。次に、突然変異を加える。突然変異は、ある確率で染色体の一部の値を変える操作である。これら一連の操作が終了すると、新しい世代の個体集団が生成されたことになる。この新たな集団に対して、また適応度評価、選択・交配、交差、突然変異を行い、さらに新しい世代を作っていく。世代更新の終了条件は、一般的には設計者が明示的に与える。ある一定値以上の適応度を持つ個体が生成された時点を経験的操作の終了とする場合や世代数を終了条件とする場合などがある。



図2.2 遺伝的アルゴリズムのフローチャート

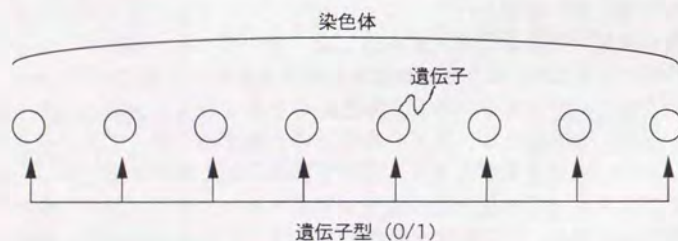


図2.3 染色体・遺伝子・遺伝子型

2.2.2 選択・交配

選択・淘汰を行う際に、どの個体同士を交配させるかを決める必要がある。現在までに幾つかの方法が提案されている中から、代表的な方法を紹介する。

(1) 適応度比例戦略

適応度比例戦略は、ルーレット選択モデルまたはモンテカルロ・モデルとも呼ばれる。各個体の適応度に比例した確率で子孫を残すためのモデルであり、最も一般的に使用される選択方式である。ある個体 i が各々の選択で選ばれる確率 P_i は次式で表わされる。但し、 f_i は個体 i の適応度とする。

$$P_i = f_i / \sum_{j=1}^n f_j \quad (2.1)$$

(2) ランク戦略

ランク戦略は、適応度によって各個体をランクづけし、予め各ランクに対して決められた確率で子孫を残す戦略である。表2.1に一例を示す。各個体は、その適応度ごとにランク付けされており、ランクに依存して選択確率が決まる。この方式では、ランクによって決まる選択確率をどのように設定するかがポイントとなる。

表2.1 ランク戦略の例

ランク	個体	元の 適応度(f)	スケール後の 適応度(f^*)	選択確率
1	8	3.0	9.00	0.35
2	1	2.5	6.25	0.20
3	7	2.5	6.25	0.15
4	5	2.1	4.41	0.10
5	6	2.0	4.00	0.06
6	3	1.8	3.24	0.05
7	10	1.2	1.44	0.04
8	9	1.0	1.00	0.03
9	2	0.9	0.81	0.02
10	4	0.7	0.49	0.00

(3) エリート保存戦略

エリート保存戦略は、集団中で最も適応度の高い個体をそのまま次世代に残す方法である[DeJong 75]。この方法では、その時点で最も良い解が交差や突然変異で破壊されないという利点がある一方、エリート個体の遺伝子が集団中に急速に広がる可能性が高いために局所解 (local minima) に陥る危険性もある。

(4) トーナメント選択戦略

トーナメント選択は、集団から一定数の個体を無作為に選択し、その中で最も適応度の高い個体を次世代に残すという手続きを次世代に残したい数の個体を選択されるまで繰り返す方法である。

2.2.3 交差

交差 (crossover) は2つの親の染色体を組み替えて子の染色体を作る操作であり、幾つかの手法が提案されている。ここでは、代表的な交差に関して概説する。

(1) 1点交差

最も一般的な手法は、交差位置を一個所決めて、その前後でどちらの親の遺伝子型を受け継ぐかを定める方法であり、1点交差 (one-point crossover) と呼ばれている。単純交差 (simple crossover) と呼ぶこともある。図2.4の例では、2番目と3番目の遺伝子座の間に交差位置があり、個体Aの染色体の先頭から2番目までと個体Bの3番目から最後までの遺伝子が、新しい個体の遺伝子となる。また、その逆が、他の新しい個体の遺伝子となる。

個体A	1 0	0 1 1 1 1	→	1 0 1 0 0 1 0
個体B	0 0	1 0 0 1 0	→	0 0 0 1 1 1 1

図2.4 1点交差の例

(2) 多点交差

多点交差 (multipoint crossover) は、交差位置が複数ある方法である。一般的には、2点交差が良く使われている。一例を図2.5に示す。例えば交差位置が2と4なら、新たな個体の一つは個体Aの先頭から2番目まで、個体Bの3番目から4番目まで、個体Aの5番目から最後までによって遺伝子が作られる。同時に、その逆の組合せで、もう一つの新たな個体の遺伝子が作られる。

個体A	1 0	0 1	1 1 1 1	→	1 0 1 0 1 1 1
個体B	0 0	1 0	0 1 0	→	0 0 0 1 0 1 0

図2.5 多点交差 (2点交差) の例

(3) 一様交差

一様交差 (uniform crossover) は、交差時にマスクをかけてどちらの親の遺伝子を受け継ぐかを決定する方法であり、Syswerdaによって提案された[Syswerda 89]。一例を図2.6に示す。二つの親となる個体およびマスクを設定する。マスクビットが0の時は、子-1には親-1の遺伝子をコピーし、1の時は、親-2の遺伝子をコピーする。子-2に関しては、この逆を行う。多点交差の一種と考えられるが、1点交差や2点交差よりも有効な場合が多いとの報告[Spears 91]もされている。

親-1	1 0 0 1 1 1 1
親-2	0 0 1 0 0 1 0
マスク	0 1 0 1 0 1 0
子-1	1 0 0 0 1 1 1
子-2	0 0 1 1 0 1 0

図2.6 一様交差の例

(4) その他の交差

その他、問題の性質により、幾つかの交差手法が提案されている。例えば、Goldbergは巡回セールスマン問題などを対象として、部分一致交差 (Partially Matched Crossover) を提案している。この方法は、2点交差を行い、順路の一部分を交換した上で、順路としての意味を保つように、部分的な入れ換えを行う方法である。また、2次元画像を対象として、各画素を遺伝子とみなし、一つの画像を2次元の染色体とみなした上での2次元交差も提案されている[Sawai 93]。

2.2.4 突然変異

突然変異は、遺伝子を一定の確率で変化させる操作である。突然変異による探索は、ランダムサーチの一種と考えることができる。単純なランダムサーチと比較すると、複数の個体が並列に探索を行うために比較的初期値に依存されないという特徴を持つ。また、並列化の効果も期待できる。突然変異の目的は、交差によって生成される遺伝子型の多様性維持と考えられる。すなわち、交差だけでは生成不能な遺伝子型を突然変異によって生成することにより、探索空間を拡張することができる。また、局所解に陥った場合に、そこから脱出する確率を高める効果を持つ。

2.3 スキーマ定理と積木仮説

ここでは、上記遺伝的操作を効率的に行う上で重要と思われるスキーマ定理と積木仮説を示す。

染色体が1次元の文字列で表現されている時、優れた形質の生成に貢献する文字列の集まりをパターン化したものをスキーマ (schema) と呼ぶ。スキーマ定理 (schemata theorem) は、このようなパターンが次世代で生き残る期待値を示す定理であり、GAの基本定理とも言われている。

スキーマ H の中で最初の定数と最後の定数の距離をスキーマ H の定義長 (defining length) といい、 $\delta(H)$ で表す。スキーマ H 中の定数の数をオーダ (order) といい、 $O(H)$ で表す。例えば、 $1**00$ のスキーマは、定義長が4でオーダは3である。また、 $*1*0*$ のスキーマは定義長が2でオーダは2である。世代 t の集団において、スキーマ H を含む個体数を $m(H, t)$ 、 H を含む個体の平均適応度を $\bar{f}(H)$ 、集団内の全個体の平均適応度を \bar{f} で表すとき、以下に示す式(2.2)をスキーマ定理という。但し、 L は染色体長、 P_c は交差率、 P_m は突然変異率を表す。

$$m(H, t+1) \geq m(H, t) \frac{\bar{f}(H)}{\bar{f}} [1 - P_d] \quad (2.2a)$$

$$P_d = P_c \frac{\delta(H)}{L-1} + P_m O(H) \quad (2.2b)$$

式(2.2)において、第1項は選択・交配によりスキーマ H を含む個体を選ばれる確率を示し、第2項は交差によりスキーマ H が破壊される確率、第3項は突然変異によりスキーマ H が破壊される確率を示している。したがって、この定理は一見すると、突然変異および交差をできるだけ減らした方が、適応したスキーマが集団中に拡散していくように思える。しかし、突然変異や交差を行わなければ、進化 (新しい領域の探索) もしない。これは、スキーマ定理では、突然変異や交差で生成される新たなスキーマに関する分析がされていないためである。この定理は、あるスキーマに着目したときに、そのスキーマの個数が集団中でどのように変化するかに関する予測を与えるに過ぎない。

一方、式(2.2)から、構成長 $\delta(H)$ が短く、オーダ $O(H)$ が低く、平均適応度 \bar{f} より高い適応度を示すスキーマ H が次の世代で生き残りやすいことがわかる。このようなスキーマを積木 (building block) と呼ぶ。また、このような積木単位に交差した方が、ランダムな交差よりも、最適な遺伝子が得られやすいと思われる。この希望的観測を積木仮説 (building block hypothesis) と呼ぶ。

2.4 フィードフォワードニューラルネットワークの合成

ある目的に適した能力を持つニューラルネットワークを作るためには、以下の2点に関して考慮する必要がある。

- (1) ニューラルネットワークの構造決定。すなわち、各ニューロン間の接続方法の決定。
- (2) ニューラルネットワークの学習。すなわち、各ニューロン間の結合荷重値の大きさを、与えられた問題に適した値に収束させる。

GAをフィードフォワード型ニューラルネットワークの合成に適用する場合、表2.2に示す4種類のアプローチが採られている。

2.4.1 進化的計算を用いた結合荷重値の学習

第1の方法は、ニューラルネットワークの構造は人間が予め与え、ニューロン間の結合荷重値の学習に対してGAを用いる方式である。この方式は、Whitelyら [Whitely 89] や Montanaら [Montana 89] の研究を始めとする。図2.7に例を示す。この研究では、1次元染色体上に直接、ネットワークの重みやノードのバイアスを表現してトレーニングを行う。MontanaとDavisは実値表現 (real value encoding) を使い、WhitelyとHansonは二値表現 (binary encoding) を使っている。いずれの表現でも、初期集団がランダムに生成され、各々が違ったウエイトのニューラルネットワークを表現する。GAを利用してニューラルネットワークの学習を行う際には、各染色体をネットワークにマップして、1エポック分のパターンを提示し、その際の二乗和誤差 (Total Sum Square error: TSS) の逆数などを適応度として使用する。全てのネットワークの適応度が決定すると、次の世代を残すために、選択・交配、交差、突然変異という遺伝的操作が加えられる。このような手続きを繰り返すうちに適応度が向上する。すなわち、二乗和誤差が低減する。しかし、この手法はBack Propagation (BP) 等の通常の学習法よりも、必ずしも効率的ではないとの報告 [Kitano 90] がされている。特に、Fahlmanが提唱したQuick Prop (QP) [Fahlman 88] という高速化されたBPとの比較実験では、GAによる学習は一貫してQPより非効率であるという実験結果が示されている。

2.4.2 進化的計算とBPのハイブリッド化による結合荷重値の学習

第2の方法は、GAとBPのハイブリッドにより、ニューロン間の結合荷重値の学習を行う方式である。すなわち、学習の初期段階をGAで行い、途中からBPに

表2.2 進化的計算を用いたニューラルネットワークの合成

ニューラルネットワークの構造	結合荷重値の学習
人手	進化的計算
進化的計算	BP
人手	進化的計算 + BP
進化的計算	進化的計算

BP: Back Propagation

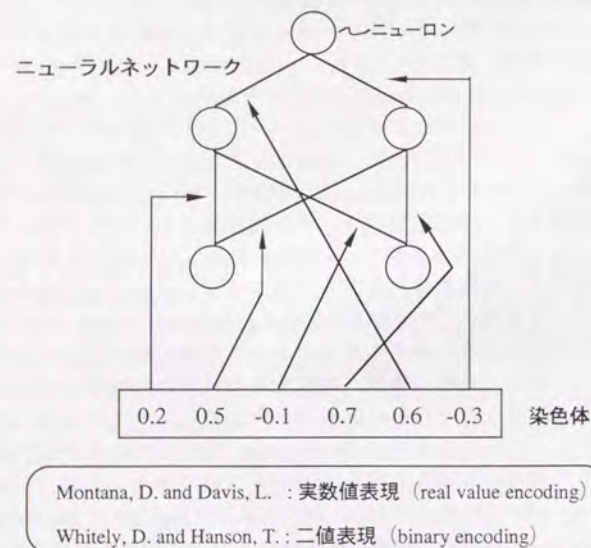


図2.7 GAによる結合荷重値学習の例

切り換える方式[Kitano 90]である。これは、最適解の近傍まで速く収束することができるが、そこから最適解への局所的探索が苦手なGAの特徴と、局所探索を得意とするBPの特徴を組合せた有効な方式であると考えられる。例えば、XOR問題を解くニューラルネットワークの学習を用いた実験[Kitano 90]では、単純にGAのみで学習を行う場合、あるいは、BPのみで学習を行う場合よりも明かに速く収束することが示されている。但し、2スパイラル問題を用いたGAとQPの初期収束の速さの比較では、GAが特に有効という結果はでていない。ここで、2スパイラル問題とは、二つの渦巻きによって構成されている解空間の194の点があり、各々どちらの渦巻きに属するかを判定する問題であり、非線形性の高い問題である。また、上記の手法と同様に、ニューラルネットワークの構造は予め設計者が経験を基に試行錯誤で決定する必要がある。

2.4.3 進化的計算を用いたニューラルネットの構造決定

第3の方法は、ニューラルネットワークの学習方法は人間が与え、GAによってニューラルネットワークの構造を決定する方式である。GAを使ってニューラルネットワークの構造を決定する代表的な手法では、まず、染色体上にネットワークの構造を表現し、それを学習や評価が可能なネットワークに変換する。次に、このネットワークにランダムにウエイトを設定し、BP等である程度学習を行い、適応度を計算する。この方式の代表的な研究としては、Millerらの報告[Miller 89]およびHarpらの報告[Harp 89]が挙げられる。例えば、Millerらの方法では、ニューラルネットワークの全ての結合に関して、結合行列を用いて指定する。この結合行列は、図2.8に示す方法により、1次元染色体上にエンコードされる。この方法は、ネットワーク決定に関する設計者の負担を低減する可能性を有するが、ネットワーク規模の増大に対して最適なネットワーク構造に到達するまでの計算量が爆発的に増大するために、大規模ネットワークの決定に適用するのは適さないとの報告[Kitano 90]がある。この原因の一つは、交差の際に、ニューラルネットワーク上の2次元部分構造が保持されにくいと考えられる。すなわち、Millerらの手法では、ニューラルネットワーク上に2次元スキーマが形成されても、1次元染色体上では分散配置されてしまうために定義長が非常に大きくなり、2次元スキーマが交差で破壊される確率が非常に高いと考えられる。

2.4.4 進化的計算を用いた構造決定と結合荷重値の学習

第4の方法は、ニューラルネットワークの構造決定とニューロン間の結合荷重値の学習の両方に対してGAを適用する方式である。上記2種類の適用方法に比べると研究の歴史は浅く、有効性の可否に関しては未知数である。この方法の代

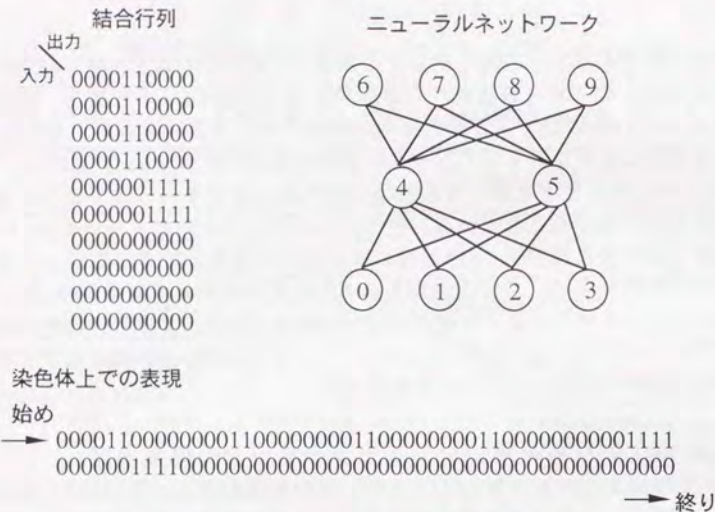


図 2.8 GAによるニューラルネットワークの構造表現の例

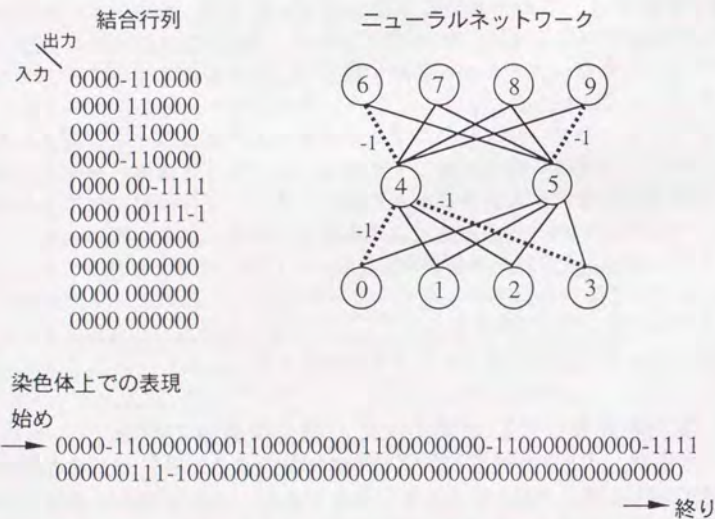


図 2.9 GAによるニューラルネットワークの構造と結合荷重値表現の例

表的な研究としては長尾らの研究[Nagao 92, 93]が挙げられる。この方法では、図2.9に示すように、入力ユニット群、隠れユニット群、出力ユニット群から構成される層状の階層型ニューラルネットワークを扱う。各ニューロンは自分自身の状態として0（非励起状態）または1（励起状態）のいずれかの状態をとり、ある時点での入力信号の総和が0よりも大きい場合は自分自身の次の状態を1にし、それ以外の場合は次の状態を0にする。ユニット間の結合荷重値は1, -1, または0のいずれかをとり、0はニューロン間に結合がないことを示す。後は、Millerらの方法と同じように結合行列を作り、1次元染色体上にエンコードする。ここで、Millerらの手法との差は、結合行列の要素として、結合の有／無し（"1", "0"）ではなくて、結合の荷重値（"1", "-1", "0"）を用いている点にある。原理的には、結合荷重値として実数を扱うことも可能である。したがって、遺伝的操作を行うことにより、ニューラルネットワークの構造決定とニューロン間の結合荷重値の学習の両方を並行して行うことができる。但し、1次元染色体上での遺伝的操作を仮定しているために、Millerらの手法と同様に、ニューラルネットワーク上の2次元スキーマが交差で破壊される確率が非常に高くなると思われる。

2.5 リカレントニューラルネットワークの合成

前節で紹介したように、GAを用いたフィードフォワード型ニューラルネットワークの合成に関しては、1989年頃から精力的な研究がなされている。一方、GAを用いてリカレントニューラルネットワークを合成する研究は、世界的に見ても、まだ始まったばかりの段階である。GAなどの進化的計算を用いたリカレントニューラルネットワーク合成に関する初期の研究報告としては、例えば、sunspot prediction problemを対象としたリカレントニューラルネットワークの合成に関する報告[Saravanan 93]、finite state automataの学習を対象としたリカレントニューラルネットワークの合成に関する報告[Angeline 94]、5-bit even parity problemを対象としたリカレントニューラルネットワークの合成に関する報告[Santos 94]、Tic-Tac-Toeゲームの戦略獲得を目的としたJordan型リカレントニューラルネットワークの合成に関する報告[Sato 94a~94d]などがある。しかし、これらの問題は必ずしもリカレントニューラルネットワークを必要としない問題である。すなわち、フィードフォワード型のニューラルネットワークでも学習可能な問題である。佐藤の報告[Sato 94a]を一般的なりカレントニューラルネットワークの学習用に拡張した方法[Ogihara 94]も報告されているが、十分な実験結果は得られていない。

2.6 まとめ

本章では、進化的計算の概要および進化的計算を用いたニューラルネットワークの合成に関する従来研究のまとめを行った。進化的計算は、生物進化から着想を得た確率的探索手法であり、従来手法では解決が困難であったさまざまな最適化問題や探索問題に対して、実用上の準最適解を速やかに得ることができる適応範囲の広い多点探索アルゴリズムとして、最近になって俄かに、注目され始めている。特に、ニューラルネットワークとの融合により、お互いの弱点を補う効果的な探索を行える可能性がある。すなわち、初期値依存性が比較的少なく、かつ、局所解の近傍に素早く収束する進化的計算の利点と、非線形問題や時系列問題に有効なニューラルネットワークの利点を統合した強力な探索モデルを実現する可能性がある。しかし一方で、進化的計算を利用したフィードフォワード型ニューラルネットワークの合成に関する従来研究では、ニューラルネットワークという2次元の構造を1次元の染色体に対応させた上でランダムな位置での交差を行っているために、ニューラルネットワーク上の2次元部分構造が保持されにくいという問題点が存在する。すなわち、進化的計算では、building blockと呼ばれる部分構造をうまく組合せて効率的な探索を実現する可能性を持つが、このような部分構造は用いられる表現形式と問題空間に応じて暗黙のうちに決まることが多いという点に関して、従来研究では十分な考慮がなされていない。また、リカレントニューラルネットワークの合成に対する進化的計算の適用に関しては、研究が始まったばかりの段階であり、進化的計算の効果的な適用方法が明かになっていない。進化的計算をどのようにニューラルネットワークの合成に適用するかに関する継続的な研究が必要と考えられる。

第3章

表現型レベルの遺伝的操作を用いた ニューラルネットワークの合成

要旨

この章では、ニューラルネットワークの構造および結合荷重値の学習を目的として、表現型レベルで遺伝的操作を施す手法について検討する。すなわち、ニューラルネットワーク自体を2次元の染色体と定義し、ニューラルネットワーク上の2次元部分構造を活かした交差を提案し、その有効性を検討する。これは、進化的計算をニューラルネットワークに適用する従来手法が、基本的に1次元染色体上でのランダムな位置での交差を仮定しているために、ニューラルネットワーク上の2次元スキーマを交差で破壊する確率が高いと考えるためである。また、提案する手法では、ニューラルネットワークの構造決定と結合荷重値の学習の両方同時に進化的計算を適用することを特徴とする。

この手法を、幾つかの論理合成問題を用いて、1次元染色体上で1点交差、2点交差あるいは一様交差を行う標準的な遺伝的アルゴリズムと比較し、以下の結論を得た。提案する表現型レベルでの遺伝的操作は、Exclusive-OR論理を含んだ論理合成問題や教師データの一部分が動的に変化する問題など対して有効である。

これによって、Exclusive-OR論理のような「だまし問題」（進化的計算での学習が困難とされている問題）やノイズなどの影響を考慮する必要のある実世界の動的な問題への進化的計算の効果的適用法の一例を示すことができたと考える。

また、上記遺伝的操作を個体総数の変化に比較的影響されないロバストな手法に改良する一つの試みとして、適応度に依存して遺伝的操作の並列度を学習過程において動的に切り換える手法を提案する。

3.1 はじめに

Back Propagation (BP)学習[Rumelhart 86]を始めとしてニューラルネットワークの学習は、予めニューラルネットワークの構造を与えておき、ニューロン間の結合荷重値を学習する方式が一般的である。すなわち、ニューロン数、ニューロン間の結線情報および学習率などは予め思考錯誤により人間が与える必要があり経験に依存する負担の大きい作業である。

このような設計者の負担を軽減する一つ的手段として、Genetic Algorithms (GA)を用いたニューラルネットワークの合成が着目され始めている。例えば、Montanaら[Montana 89]やWhitleyら[Whitley 89]は、構造が既知であるニューラルネットワークの結合荷重値の学習にGAを用いる研究を行っている。Millerら[Miller 89]やHarpら[Harp 89]は、GAを用いたニューラルネットワークの構造設計の研究を行っている。長尾ら[Nagao 93]はニューラルネットワークの構造決定と結合荷重値の学習の両方に対してGAを適用する研究を行っている。一方、これらの研究では、ニューラルネットワークという2次元の構造を1次元の染色体に対応させた上でランダムな位置での交差を行っている。したがって、ニューラルネットワーク上の2次元部分構造が保持されにくいという問題点が存在する。例えば、ニューラルネットワーク上のある2次元部分構造が非常に有効（すなわち、期待する出力信号を生成するために効果的）だとしても、それは1次元染色体上では、分散配置されてしまう。あるいは、非常に大きな定義長を持った構造になってしまう。すなわち、交差の時に破壊される確率が非常に高くなる。また、有効な2次元部分構造のコピーやシフトも困難である。

このような問題点をふまえ、Kitano[Kitano 90]はグラフ生成システムを利用したニューラルネットワークの合成を提案している。この方法は、2次元構造を保持する能力に関しては改善されている。しかし、ネットワーク構造そのものではなく構造生成の文法規則を1次元染色体として定義しているため、染色体を見ても対応するニューラルネットワークが直観的にイメージできない。また、通常のGAの操作の他に、文法の成長操作および複数種類の生成ルールの記憶が必要である。すなわち、複雑な操作と多くの記憶容量が必要である。

本章では、上記従来手法の問題点を解決するために、表現型レベルでの遺伝的操作の適用を提案し、その有効性を示す。以下、3.2節では、提案する表現型レベルでのGAを示す。3.3節では、スキーマ定理との関係から提案する手法の有効性を検討する。3.4節では、遷移行列を用いて表現型レベルでの遺伝的操作を簡単に実現する一例を示す。3.5節において、幾つかの論理合成問題を用いて、1次元染色体上で1点交差、2点交差あるいは一様交差[Syswerda 89, Spears 91]を行う標準的なGAとの比較を行い[Sato 95a]、最後に考察とまとめを行う。

3.2 表現型レベルの遺伝的操作を用いたニューラルネットワーク合成の提案

図3.1に、表現型レベルでの遺伝的操作の基本的な考え方を示す。図3.1は、ニューラルネットワーク自身を2次元の染色体と考え、ニューラルネットワーク上の2次元部分構造単位にニューロン間の結線情報と結合荷重値の置き換えを行う交差を表している。ニューロン間の矢印は、ニューロン間に結線があり、矢印の方向に信号が伝搬することを示している。実線が+1の結合荷重値、点線が-1の結合荷重値を表す。ここでは結合荷重値として、0, +1, -1の3値の場合を想定したが、実数値を扱うことも可能である。交差の単位となる2次元部分構造には、ランダムに選んだあるニューロンを頂点として入力層を底辺とするピラミッド構造を用いる。ピラミッド構造の頂点となるニューロンに対して、直接または間接的に、入力源となる可能性のあるニューロンは全てこのピラミッド構造の中に含める。図3.1に示す交差はGenetic Programming (GP)[Koza 92]の交差に似ているが、以下の点で大きく異なる。(i) 標準的なGPで扱うノードはリンクとの間に結合強度という概念を持たない。したがって、結合荷重値の学習もない。(ii) GPの交差はノードの移動を伴う。提案する手法では、ノード間の結線情報と結合荷重値の交換を行い、ノード自体は固定である。(iii) GPの交差は実際にノード間に結合の存在する木構造単位の置き換えを行う。提案する手法ではピラミッド構造の頂点となるニューロンに対して、直接または間接的に、入力源となる可能性のあるノード（ニューロン）は全て交差の単位となるピラミッド構造内に含める。したがって、ニューラルネットワークの入力層全体が必ずピラミッド構造の底辺となる。提案する手法のこれらの特徴は以下の理由による。

今、あるニューロンが N 本の入力を持つと仮定して各入力の信号の強さを $S_1 \sim S_N$ 、各入力信号に対する結合荷重を $w_1 \sim w_N$ 、このニューロン自体の出力の強さを x とすると、ニューロンの動作は次式の離散時間の差分方程式で記述される。 h は、しきい値である。

$$x(t+1) = F \left[\sum_{i=1}^N w_i S_i(t) - h \right], \quad (3.1)$$

$$\begin{aligned} F[u] &= 1 & (u \geq 0) \\ &= 0 & (u < 0) \end{aligned} \quad (3.2)$$

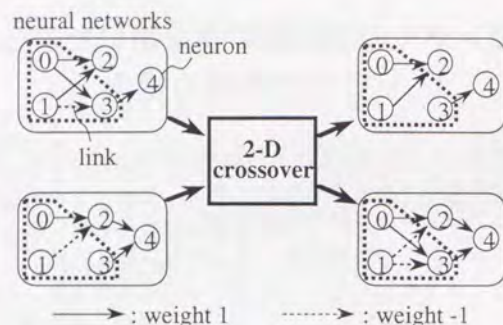


図 3.1 ピラミッド構造単位の2次元交差の例

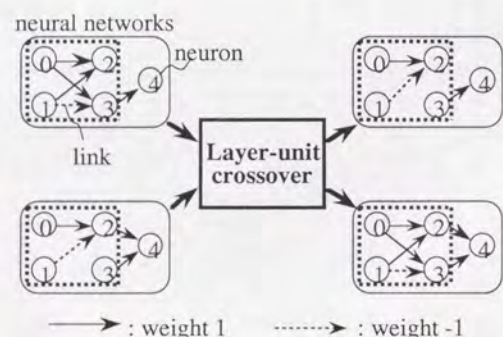


図 3.2 階層単位の2次元交差の例

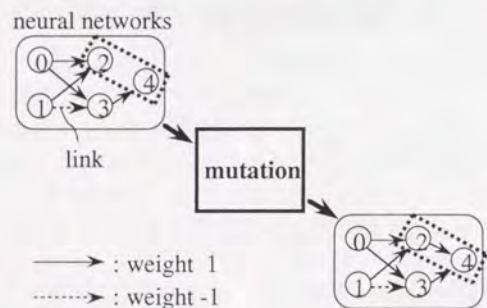


図 3.3 突然変異の例

ニューロンの各入力および出力は、1または0の値を取り、1はニューロンが興奮・発火した状態、0はニューロンが興奮しない状態に各々対応する。式(1)から、あるニューロンの出力が意味ある定常状態（すなわち、期待するニューラルネットワークの出力値を生成するために有効な値に確定している状態）になるための条件は、そのニューロンの入力源となるニューロンとの間の結線情報および結合荷重値が確定し、かつ、入力信号が定常状態になっていることである。あるニューロンの入力信号とは、そのニューロンの入力源となるニューロンの出力信号である。今着目しているニューロンの入力源となるニューロンの出力信号が意味ある定常状態となる条件についても同様のことが言える。この条件を入力層まで再帰的に繰り返して考えると、結局、あるニューロンの出力が意味のある定常状態になるためには、このニューロンを頂点として入力層を底辺とするピラミッド構造内の結線情報およびニューロン間の結合荷重値が確定している必要がある。すなわち、期待する出力を得るために有効な2次元部分構造は、あるニューロンを頂点として入力層を底辺とするピラミッド構造、あるいは、これら幾つかのピラミッド構造の集合体になる可能性が高い。したがって、このようなピラミッド構造を単位とした交差を行えば、有効な2次元部分構造を破壊する可能性が低下すると考えられる。本手法は、基本的にはMillerらの提案する直接法に属する手法である。しかし、ニューラルネットワークの構造決定にGAを適用した後で結合荷重値の学習をBPなどにより行う操作を何回も繰り返すMillerらの手法[Miller 89]と比較すると、ニューラルネットワーク合成に要する時間を短縮できる可能性を持つ。また、前記理由から、結線情報と結合荷重値を合わせて操作することにより、ニューラルネットワーク上の特徴ある部分構造を活かした交差を行える可能性を持つ。

有効なピラミッド構造が複数存在する場合は、幾つかのピラミッド構造をまとめて扱えると効率が良い。図3.2では、交差の単位となる2次元部分構造に、幾つかのピラミッド構造を組み合わせた階層単位の構造を採用している。図3.2の例では、ノード2を頂点とするピラミッド構造とノード3を頂点とするピラミッド構造を組み合わせた階層からなる交差を示している。

図3.3は、突然変異の例を示している。突然変異は、ノード間の結線情報の付加もしくは削除、あるいは、結合荷重値の変異に対応する。図3.3の例では、ノード2とノード4の間に結線がない状態から結合荷重値+1の結線がある状態に変異した例を示している。

本手法では交差の単位となるピラミッド構造の多様性が必要である。しかし、多様性に関しては、初期集団数を適当に多く設定することで対応できる。突然変異率を少し高めに設定することも考えられる。ピラミッド構造内部に関して、従来の1次元の交差を並行して適用することも考えられる。

3.3 スキーマ定理と表現型レベルでの遺伝的操作

スキーマ定理に関しては既に2.3節で示したが、ここでもう一度最初にスキーマ定理[Goldberg 89]を簡単に示し、次に提案する表現型レベルでの遺伝的操作の有効性をスキーマ定理との関係から考える。

染色体が1次元の文字列で表現されている時、優れた形質の生成に貢献する文字列の集まりをパターン化したものをスキーマ(schema)と呼ぶ。スキーマ H の中で最初の定数と最後の定数の距離をスキーマ H の定義長(defining length)といい、 $\delta(H)$ で表す。スキーマ H 中の定数の数をオーダー(order)といい、 $O(H)$ で表す。世代 t の集団において、スキーマ H を含む個体数を $m(H, t)$ 、 H を含む個体の平均適応度を $f(H)$ 、集団内の全個体の平均適応度を \bar{f} で表すとき、以下に示す式(3.3)をスキーマ定理という。

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{\bar{f}} [1 - P_d] \quad (3.3a)$$

$$P_d = P_c \frac{\delta(H)}{L-1} + P_m O(H) \quad (3.3b)$$

但し、 L は染色体長、 P_c は交差率、 P_m は突然変異率を表す。

式(3.3)において、第1項は選択・交配によりスキーマ H を含む個体が選ばれる確率を示し、第2項は交差によりスキーマ H が破壊される確率、第3項は突然変異によりスキーマ H が破壊される確率を示している。表現型レベルでの遺伝的操作の基本的考え方は、図3.1に示したピラミッド構造単位の交差を行うことにより、スキーマが破壊される確率を低下させることである。すなわち、1次元スキーマに対応するニューラルネットワーク上の2次元構造が図3.1に示したピラミッド構造、あるいは、これらピラミッド構造の集合体になる確率が高いと考えるのである。2次元GAを用いた場合、世代 t の集団におけるスキーマ H を含む個体数を $m(H, t)^*$ と表すと、以下に示す式(3.4)が成り立つ。

$$m(H, t+1)^* \geq m(H, t) \frac{f(H)}{\bar{f}} [1 - P_d^*] \quad (3.4a)$$

$$P_d^* = k_c P_c \frac{\delta(H)}{L-1} + P_m O(H), \quad (0 \leq k_c \leq 1) \quad (3.4b)$$

但し、 k_c はスキーマがピラミッド構造に対応しない確率であり問題に依存する変数である。

例えば、 $L=9$ 、 $\delta(H)=6$ 、 $O(H)=4$ 、 $P_c=1$ 、 $P_m=0.005$ の場合、 k_c の値に依存して、

$$1 \leq \frac{m(H, t)^*}{m(H, t)} \leq 4$$

となる。すなわち、提案する手法では、1世代当たりのスキーマの保持確率が最大4倍に向上する。但し上記は、次世代においてスキーマ H を含む個体数の下限に関する議論である。実際には、交差および突然変異によりスキーマが獲得されることを考慮する必要がある。

3.4 遷移行列を用いた遺伝的操作

ここでは、遷移行列を用いて、表現型レベルでの遺伝的操作を簡単に実現する一例を紹介する[Sato 94a, Sato 95b]。図3.4にニューラルネットワークと対応する染色体の一例を示す。図3.4において、○印中を通し番号を付けたシンボルがニューロンを表す。図3.4では、各ニューロン間の遷移状態を表す行列 M11, M12を各層ごとに考え、これらの遷移行列から構成される列 [M11, M12]を染色体と定義している。一般に、K層からなるネットワークでは、遷移行列は全部でK-1個存在する。このK-1個の遷移行列から構成される列を一つの染色体と考える。

遷移行列から構成される染色体を用いた表現型レベルでの遺伝的操作を図3.5～図3.7に示す。図3.5～図3.7は、それぞれ図3.1～図3.3の操作に対応している。すなわち、ランダムに選んだあるニューロンを頂点とするピラミッド構造単位の交差は、ランダムに選んだ遷移行列からある行を選んで置き換えるという操作に対応する。層単位の交差は、遷移行列単位の置き換えに対応する。突然変異は、遷移行列内のあるビットをランダムに変異させることに対応する。

ここで示した遷移行列を用いた遺伝的操作は、オペレーションレベルで考えると、基本的には結合行列の行または列単位の置換を行うMillerらの手法[Miller 89]に属する。しかし、3.2節で示したように、ニューラルネットワーク上の2次元スキーマはネットワーク構造と結合荷重値の双方から決定づけられるものであり、GAによる結線情報の獲得とBPによる結合荷重値の学習を独立に行うMillerらの手法では、交差において2次元スキーマを破壊する確率を低減させることは困難である。すなわち、結線情報と結合荷重値の情報を交差で同時に置換すること、かつ、ニューラルネットワーク上の2次元部分構造に着目した交差を行うことが2次元スキーマを交差で破壊する確率を低減するために有効であると提案したのは筆者の報告[Sato 94a]が最初であるとかんがえる。尚、筆者はJordan型のリカレントニューラルネットワークを仮定して上記操作方法を提案したが、筆者の手法を一般的なりカレントニューラルネットワークに拡張する場合のアイデア[Ogihara 94]も提案されている。また、荻原らの手法[Ogihara 94]をBPTTアルゴリズムを用いて結合荷重値の学習を行う際の初期状態の獲得に利用した報告[Kato 95]も最近見られる。

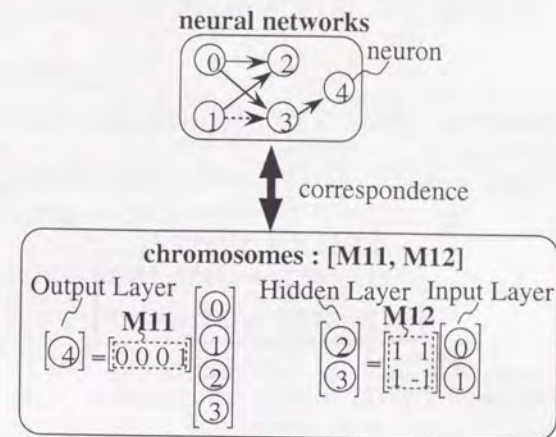


図3.4 遷移行列を用いた染色体表現の例

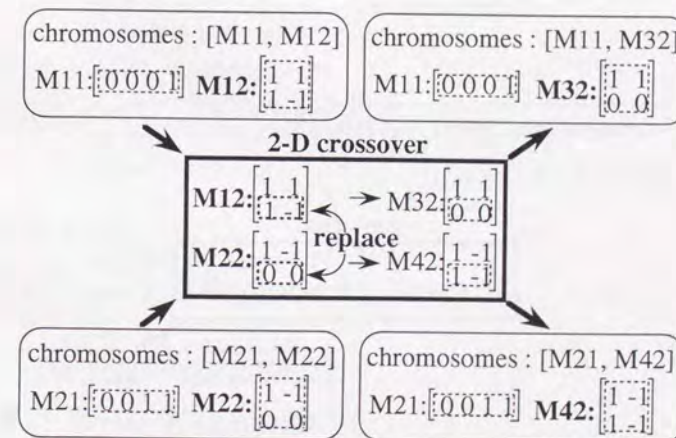


図3.5 遷移行列を用いた2次元交差の例

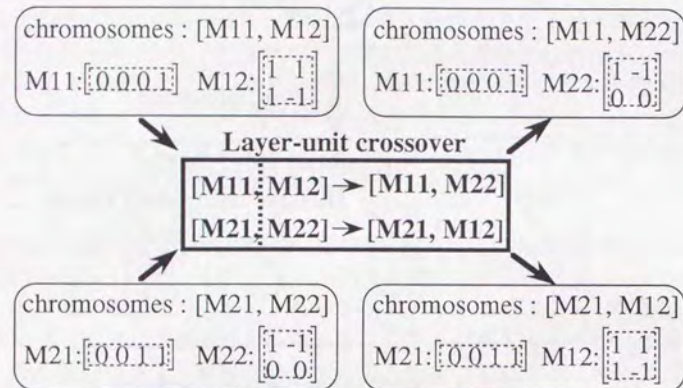


図3.6 遷移行列を用いた階層単位の交差の例

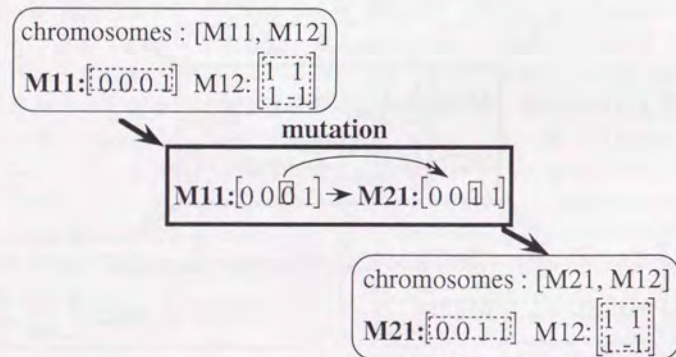


図3.7 遷移行列を用いた突然変異の例

3.5 標準GAとの比較評価実験

3.5.1 評価用問題

提案する手法の有効性評価のための実験には、以下に示す論理合成問題を用いた。特に、ニューラルネットワーク合成を目的としたGAを評価するための問題として最も広く用いられている[Whitley 89, Miller 89, Harp 89, Nagao 93, Kitano 90]Exclusive-OR (XOR)論理の学習を中心とした。

(a) 線形論理の学習

[実験 1] 入力: [0, 1, 2, 3, 4, 5, 6, 7]
出力期待値: [0, 1, 2, 3, 4, 5, 6, 7]
 $y1 = x1, y2 = x2, y3 = x3$

[実験 2] 入力: [0, 1, 2, 3, 4, 5, 6, 7]
出力期待値: [7, 6, 5, 4, 3, 2, 1, 0]
 $y1 = \overline{x1}, y2 = \overline{x2}, y3 = \overline{x3}$

(b) XOR論理の学習

[実験 3] 入力: [0, 1, 2, 3, 4, 5, 6, 7]
出力期待値: [0, 3, 6, 5, 5, 6, 3, 0]
 $y1 = x1 \oplus x2, y2 = x2 \oplus x3, y3 = x3 \oplus x1$

(c) 部分的にXOR論理を含む論理の学習

[実験 4] 入力: [1, 2, 3, 4, 5, 6]
出力期待値: [3, 4, 1, 2, 4, 5]
 $y1 = x2\overline{x3} + x1(x2 \oplus x3)$
 $y2 = \overline{x2}(x1 \oplus x3)$
 $y3 = \overline{x1}x3 + x2(x1 \oplus x3)$

[実験 5] 入力: [1, 2, 3, 4, 5, 6]
出力期待値: [3, 4, 1, 2, 6, 5]
 $y1 = x2\overline{x3} + x1(x2 \oplus x3)$
 $y2 = \overline{x2}(x1 + x3)$
 $y3 = \overline{x1}x3 + x2(x1 \oplus x3)$

[実験 6] 入力: [1, 2, 3, 4, 5, 6]
出力期待値: [3, 4, 1, 2, 6, 3]
 $y1 = \overline{x1}x2\overline{x3} + x1\overline{x2}x3$
 $y2 = x1\overline{x2} + \overline{x2}x3 + x1\overline{x3}$
 $y3 = \overline{x1}x3 + x2(x1 \oplus x3)$

(e) ノイズなどにより、教師データの一部が動的に変化している論理の学習

【実験 7】 入力：[1, 2, 3, 4, 5, 6, 5, 6]

出力期待値：[3, 4, 1, 2, 4, 5, 6, 5]

$$y1 = x2x3 + x1(x2 \oplus x3)$$

$$y2 = \overline{x2}(x1 \oplus x3) \text{ or } y2 = \overline{x2}(x1 \oplus x3)$$

$$y3 = \overline{x1}x3 + x2(x1 \oplus x3)$$

(f) 断続的に教師データの一部が変化する論理の学習

【実験 8】 入力：[1, 2, 3, 4, 5, 6]

出力期待値：[3, 4, 1, 2, 6, 5] --> [3, 4, 1, 2, 6, 3]

(240 世代目で出力期待値の一部を変化)

3.5.2 ニューラルネットワークの初期構造

ここでは、ある特定の論理に最適化したニューラルネットワークの獲得ではなく、最終的には動的な環境に自律的に適応するニューラルネットワークの実現を目指している。したがってここでは、冗長性を有する共通のニューラルネットワークで、上記全ての問題の学習を行うことにした。このネットワークの初期構造はランダムに決定した。結果的には、図3.8に示すように19ニューロンを6層（入力層から順に、3-7-2-2-2-3ニューロン）に配置した初期構造を用いた。ニューロン間の結合荷重値は、0, 1および-1の中からランダムに選択した。

3.5.3 遺伝的操作の設定

まず適応度について示す。適応度は、GAを用いて合成したニューラルネットワークが、与えられた問題に対してどの程度適しているかを示す必要がある。ここでは入力 X および出力 Y をそれぞれ3ビットで表現した上で、個体 I_i ($i=1, 2, \dots, N$)に対応するニューラルネットワークに対して、ある入力値に対する出力値が出力期待値と何ビット一致していたかを学習データの数だけ累積して適応度とした。次に遺伝的操作の設定について示す。ある世代の個体群から次世代の個体群を生成するための方法として、適応度の下位の一定の割合を淘汰し、ルーレット選択による子孫を補って個体総数を一定に保った。この遺伝規則による子孫の生成方法を図3.9に示す。すなわち、淘汰される個体の数を M 個とすると、今の世代の N 個の個体 $I_1 \sim I_N$ から二つの個体のペアを M 組だけ重複を許してランダムに選択し、それぞれに対して交差と突然変異を施して不足を補うための M 組の個体を決定した。ここで、ある個体 I_i が次の世代の個体を作るための親として選択される確率 $P(I_i)$ は次式によって決定する。

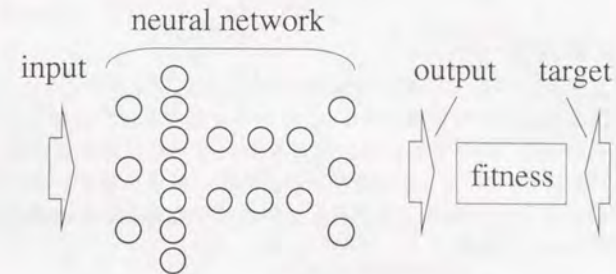


図3.8 ニューラルネットワークの初期構造

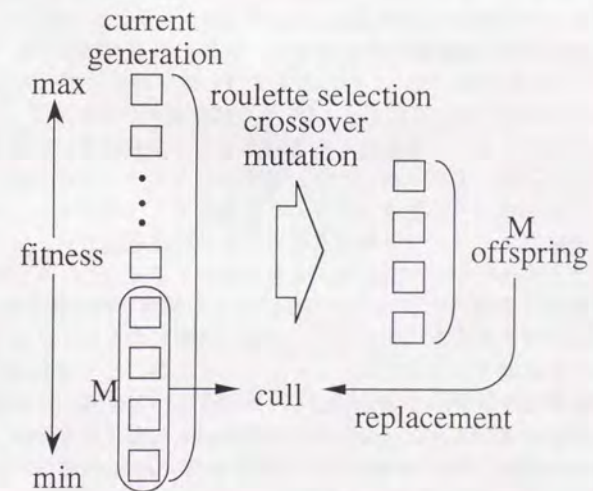


図3.9 子孫の生成方法

$$P(I_i) = f(I_i) / \left\{ \sum_{j=1}^N f(I_j) \right\} / N \quad (3.5)$$

3.5.4 実験結果

提案する表現型レベルでの遺伝的操作と1次元染色体上での1点交差, 2点交差あるいは一様交差を用いる標準的なGAとの比較を前記論理合成問題を用いて行った. 個体総数は400, 突然変異率は0.01とした. 世代数対最大適応度の関係を図3.10~図3.17に示す. また, 世代数対平均適応度の関係も示す. 変化する環境への適応性を評価するためには, 最大適応度の他に平均適応度も意味を持つ可能性があると考えられるためである.

3.5.5 考察

実験結果から, 以下のことが考えられる. 実験1, 実験2など単純な線形論理では, 提案した表現型レベルでの交差は1次元染色体上で1点交差, 2点交差あるいは一様交差を行う標準GAに比べて顕著な差は認められなかった. 実験3におけるXOR論理の学習では, 提案した手法および一様交差が他の二つよりも有効であった. 実験3の結果は, 多くの問題において, 一様交差が1点交差や2点交差よりも有効であるというSyswerdaの主張[Syswerda 89]と一致している. 一方, 実験4~実験6など, 部分的にXOR論理を含む場合, 提案した表現型レベルでの交差は標準GAに比べて有効であった. この結果をもう一度スキーマ定理を念頭において考えてみる. 次世代においてスキーマ H を含む個体数の下限を考慮して, ピラミッド構造単位の交差により, 交差による2次元スキーマ破壊確率を低減させることが提案する手法の狙いであった. 一方, 実際にはスキーマ定理では, 交差および突然変異によりスキーマが獲得されることを考慮する必要がある. 一般的に考えて, ここで提案するピラミッド構造単位の交差は, 2次元スキーマを破壊する確率を低減させる可能性を有する反面, 交差の単位を限定しているために, 新しいスキーマを生成する能力ではランダムな位置での交差を行う標準的なGAに劣る可能性があると考えられる. すなわち, 問題の性質に依存して, 交差において2次元スキーマを破壊する確率を抑えるメリットと交差により新しいスキーマが獲得される確率の低下のデメリットのどちらが大きく作用するかで本手法の有効・無効が決定すると考えられる. 今回の実験において提案する手法が効果的であった部分的にXOR論理を含む場合について考えると, XOR論理は「だまし問題」と呼ばれるGAでは学習が困難とされている問題である. すなわち, 「だまし問題」であるXOR論理の生成に有効に作用する2次元スキーマが学習の途中で獲得されたために, このような2次元スキーマを破壊しにくい交差の方が, 新しいス

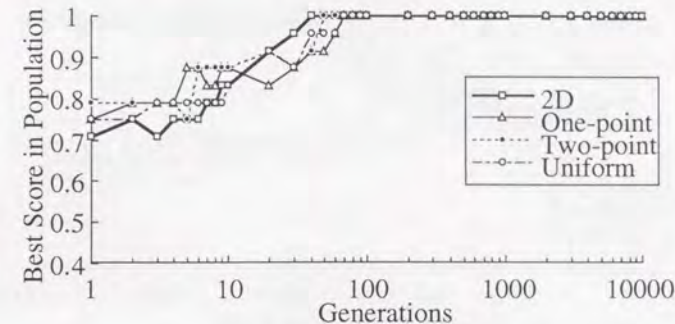


図3.10(a) 世代数と最大適応度との関係 [実験1]
Fig. 3.10 (a) The relationship between the best fitness score and the number of generations for learning of linear logic.

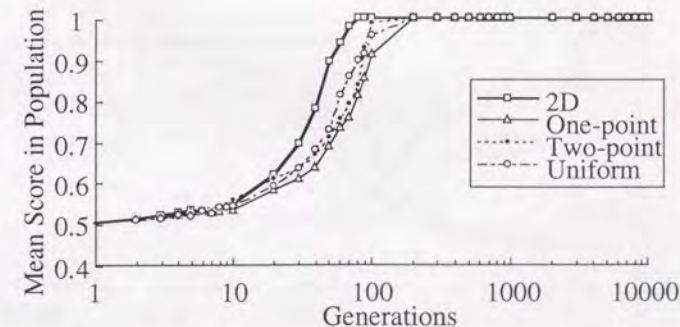


図3.10(b) 世代数と平均適応度との関係 [実験1]
Fig. 3.10 (b) The relationship between the mean fitness score and the number of generations for learning of linear logic.

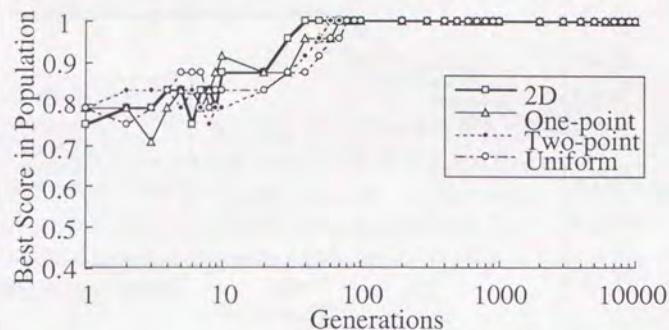


図 3.1 1 (a) 世代数と最大適応度との関係 [実験 2]
Fig. 3.11 (a) The relationship between the best fitness score and the number of generations for learning of linear logic.

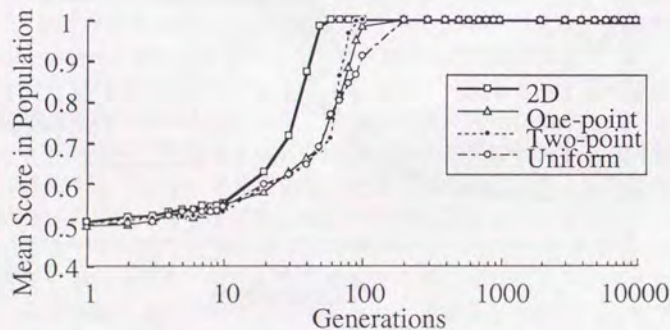


図 3.1 1 (b) 世代数と平均適応度との関係 [実験 2]
Fig. 3.11 (b) The relationship between the mean fitness score and the number of generations for learning of linear logic.

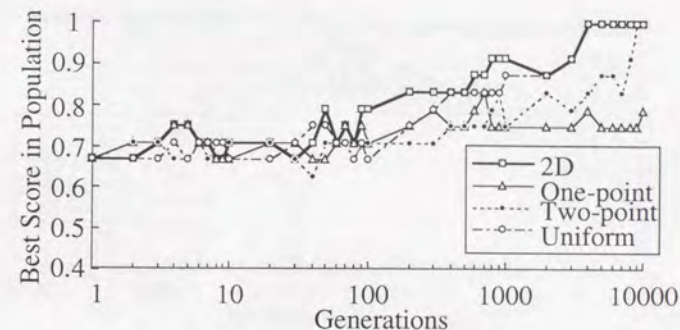


図 3.1 2 (a) 世代数と最大適応度との関係 [実験 3]
Fig. 3.12 (a) The relationship between the best fitness score and the number of generations for learning of XOR logic.

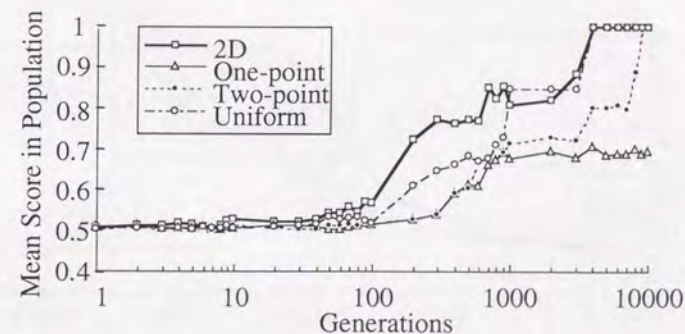


図 3.1 2 (b) 世代数と平均適応度との関係 [実験 3]
Fig. 3.12 (b) The relationship between the mean fitness score and the number of generations for learning of XOR logic.

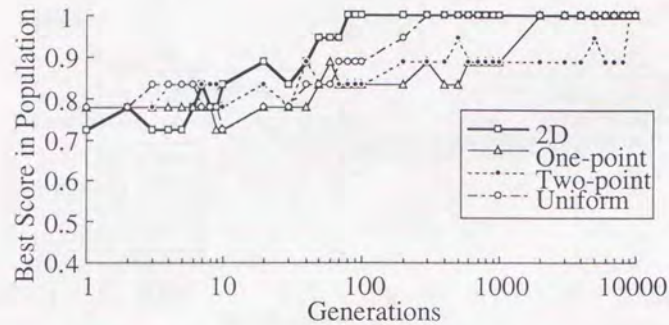


図 3.1 3 (a) 世代数と最大適応度との関係 [実験 4]
Fig. 3.13 (a) The relationship between the best fitness score and the number of generations for learning logic that partially includes XOR logic.

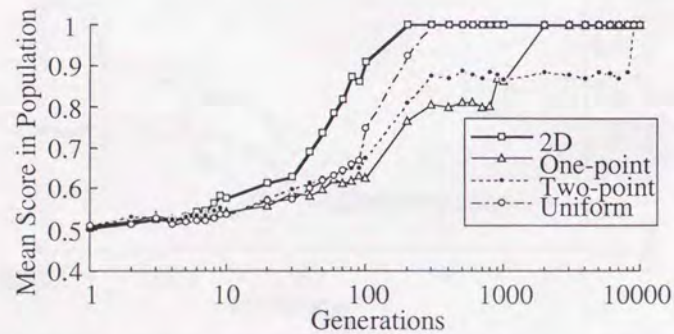


図 3.1 3 (b) 世代数と平均適応度との関係 [実験 4]
Fig. 3.13 (b) The relationship between the mean fitness score and the number of generations for learning logic that partially includes XOR logic.

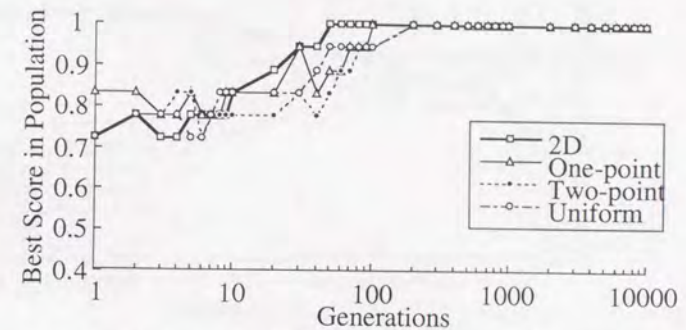


図 3.1 4 (a) 世代数と最大適応度との関係 [実験 5]
Fig. 3.14 (a) The relationship between the best fitness score and the number of generations for learning logic that partially includes XOR logic.

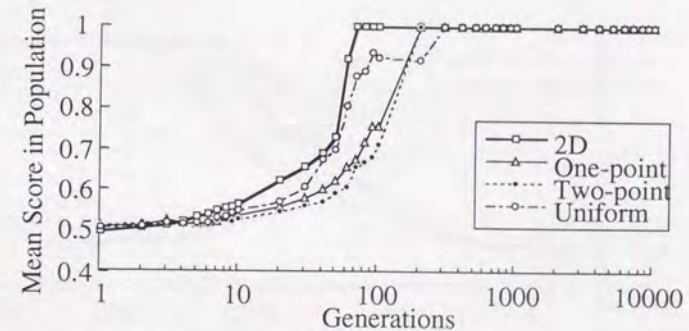


図 3.1 4 (b) 世代数と平均適応度との関係 [実験 5]
Fig. 3.14 (b) The relationship between the mean fitness score and the number of generations for learning logic that partially includes XOR logic.

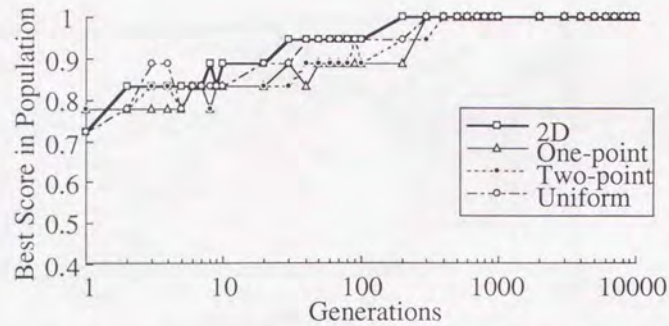


図 3.15 (a) 世代数と最大適応度との関係 [実験 6]
Fig. 3.15 (a) The relationship between the best fitness score and the number of generations for learning logic that partially includes XOR logic.

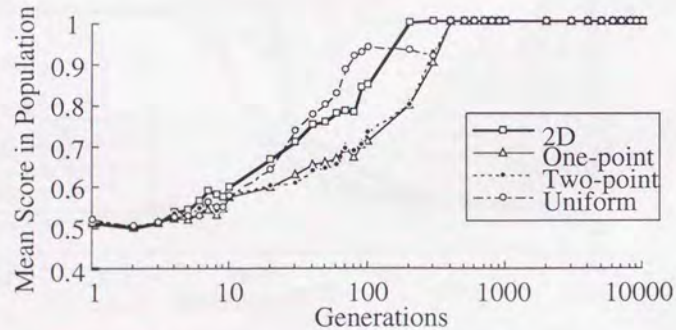


図 3.15 (b) 世代数と平均適応度との関係 [実験 6]
Fig. 3.15 (b) The relationship between the mean fitness score and the number of generations for learning logic that partially includes XOR logic.

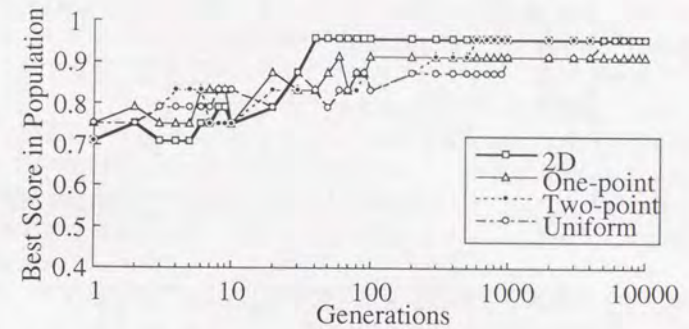


図 3.16 (a) 世代数と最大適応度との関係 [実験 7]
Fig. 3.16(a) The relationship between the best fitness score and the number of generations for learning logic involving partially dynamically changing training data.

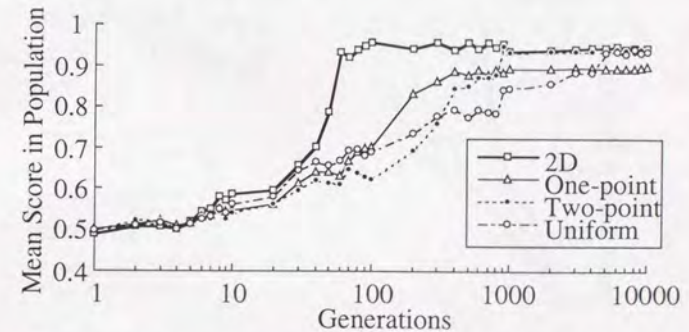


図 3.16 (b) 世代数と平均適応度との関係 [実験 7]
Fig. 3.16(b) The relationship between the mean fitness score in population and the number of generations for learning logic involving partially dynamically changing training data.

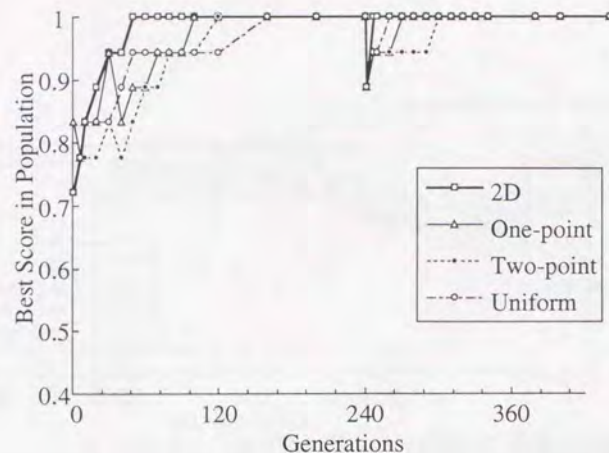


図 3.17 (a) 世代数と最大適応度との関係 [実験 8]

Fig. 3.17(a) The relationship between the best fitness score and the number of generations for learning logic with part of the training data is continuously changing.

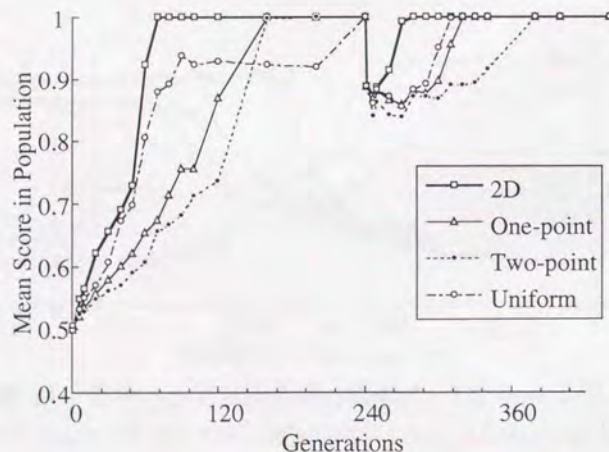


図 3.17 (b) 世代数と平均適応度との関係 [実験 8]

Fig. 3.17(b) The relationship between the mean fitness score in population and the number of generations for learning logic with part of the training data is continuously changing.

キーマを生成する能力が高い交差よりも、結果的には提案する手法が有効に作用したと考えられる。

また、実験7および実験8など、教師データの一部分が動的に変化している論理および断続的に教師データの一部が変化する論理に関して、提案した表現型レベルでの交差の有効性は顕著であった。この原因を解析するために、教師データが変化した後の入出力の関係を調査した。標準GAでは、一部の教師データの変化に対応するために、教師データの変化のない部分に関しても入出力関係を破壊していた。一方、表現型レベルでの交差では教師データの変化のない部分に関しては入出力関係を維持している傾向が高いことが分かった。世代数対平均適応度のグラフはこの調査結果を反映している。これは、ピラミッド構造単位の交差の効果と思われる。

すなわち、一般的には、GAが学習を苦手とする「だまし問題」を部分的に含んだ論理の合成や繰り返し構造の多い論理の合成にここで提案する手法は有効に働く可能性を持つと考えられる。ただし、一般的に証明するためには、さらに多くの実験が必要である。

今回、論理合成問題を扱ったために、遺伝子表現に2進表現を用い、ニューロン間の結合荷重値として0, 1, -1の3値を扱った。実数値を基本とする問題に対しては、遺伝子表現に実数ベクトル表現を用いガウシアン分布に従う突然変異を扱うEvolutionary ProgrammingやEvolution Strategies[Bäck 93]の考え方が有効である可能性を持つ。今後調査が必要である。また、大規模なニューラルネットワークで結合荷重値の学習のみにBPなどのアルゴリズムを適用した場合との比較が必要と考える。

3.6 適応度依存並列化の提案

3.6.1 適応度依存並列GAの提案

前節では、個体総数は400として実験を行った。これは、提案する表現型レベルでの交差は2次元スキーマを破壊しにくい傾向があると考えられる反面、ランダムな交差と比較して、初期収束の影響を受ける可能性が高いと考えたためである。すなわち、突然変異率を高め設定するとともに、個体総数を400という大きな数に設定して、初期集団の多様性を高める工夫のもとに実験を行った。実際、個体総数を極端に減らした場合、前節で示した程の効果が得られない場合が観測された。ハードウェア上の物理的な記憶容量の制限から必ずしも大きな個体総数を設定できない場合も考えると、個体総数の変化に比較的影響されない、ロバストな手法が望まれる。そこでここでは、表現型レベルでの遺伝的操作を個体総数の変化に対してロバストになるように改良する一つの手段として、適応度に依存して遺伝的操作の頻度を動的に変化することを提案する。学習の初期段階では、遺伝的操作を頻繁に行うことにより集団の多様性向上を優先的に考え、ある程度適応度が高くなったら、遺伝的操作の頻度を抑えて2次元スキーマを破壊させる確率を抑えることを優先するように切り換える手法である。本アイデアの実現手段としては複数の手段が考えられるが、ここでは一例として、2次元交差により置き換えるピラミッドの数を適応度に依存して動的に変化させる場合に関して検討する。異なるニューロンがピラミッド構造の頂点として選ばれる限り、2次元交差では全てのピラミッド構造が独立に処理できるために、これら複数のピラミッド構造の置き換えは並列に実行可能である。すなわち、学習過程の適応度に応じて、遺伝的操作の並列度を動的に変化させる手法の提案と考えることもできる。ここで、適応度と並列度の関係をどのように定義するかは難しい問題である。まず、線形関係を仮定して、適応度に逆比例して並列度を低下させる場合に関して実験を行った。しかし、最初から一つのピラミッド構造の置き換えしか行わない前記手法よりも、逆に収束が遅くなった。また、学習のかなり早い段階で急激に並列度を下げることが観測された。そこで次に、式(3.6)に示す正規分布の関係を仮定して、適応度の増加にともない並列度が急激に減少する場合に関して実験を行った。適応度 f と並列度 Par の関係は表3.1および図3.18に示す。

$$Par(x) = k_1 \cdot \phi(x), \quad k_1: \text{constant} \quad (3.6a)$$

$$\phi(x) = e^{-x^2/2} / (2\pi)^{1/2} \quad (3.6b)$$

$$x = k_2 \cdot f, \quad k_2: \text{constant} \quad (3.6c)$$

表3.1 適応度と並列度の関係

f	t	$\phi(t)$	$Par(t)$
0	0.00	0.39894228	24
1	0.12	0.39604047	24
2	0.24	0.38746106	23
3	0.36	0.37357307	22
4	0.60	0.33238970	20
6	0.73	0.30674115	18
7	0.85	0.27896874	17
8	0.97	0.25003340	15
9	1.09	0.22085108	13
10	1.21	0.19224720	12
11	1.33	0.16492235	10
12	1.45	0.13943057	8
13	1.57	0.11617040	7
14	1.69	0.09538761	6
15	1.81	0.07718758	5
16	1.93	0.06155480	4
17	2.05	0.04837661	3
18	2.18	0.03746864	2
19	2.30	0.02859956	2
20	2.42	0.02151344	1
21	2.54	0.01594848	1
22	2.66	0.01165166	1
23	2.78	0.00838910	1
24	2.90	0.00595253	0

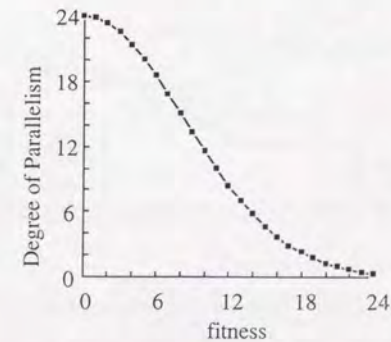


図3.18 適応度と並列度の関係

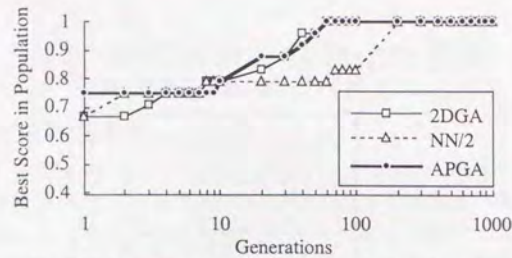


Fig. 3.19 (a) The relationship between the best fitness score and the number of generations for learning of XOR logic.
[Exp. 3] (Population size=400, seed=1013)

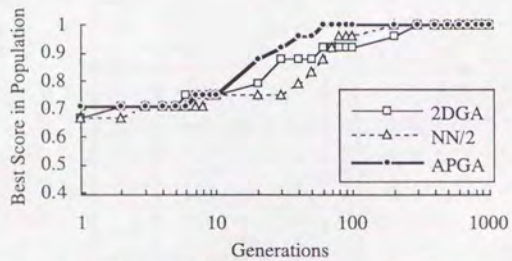


Fig. 3.19 (b) The relationship between the best fitness score and the number of generations for learning of XOR logic.
[Exp. 3] (Population size=140, seed=1013)

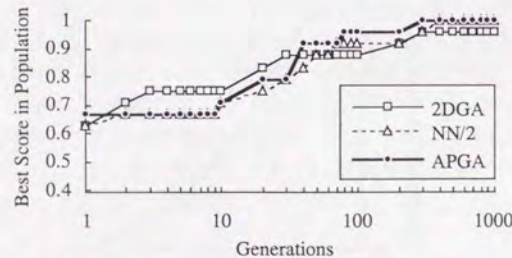


Fig. 3.19 (c) The relationship between the best fitness score and the number of generations for learning of XOR logic.
[Exp. 3] (Population size=70, seed=1013)

図 3.19 世代数と最大適応度の関係

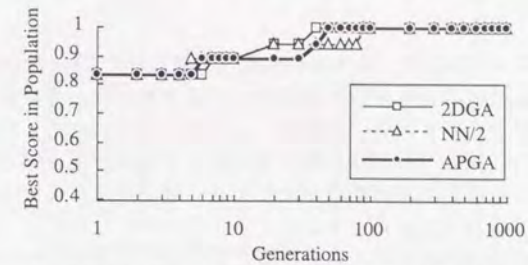


Fig. 3.20 (a) The relationship between the best fitness score and the number of generations for learning logic that partially includes XOR logic.
[Exp. 4] (Population size=400, seed=941230)

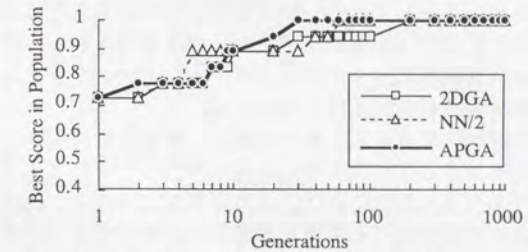


Fig. 3.20 (b) The relationship between the best fitness score and the number of generations for learning logic that partially includes XOR logic.
[Exp. 4] (Population size=140, seed=941230)

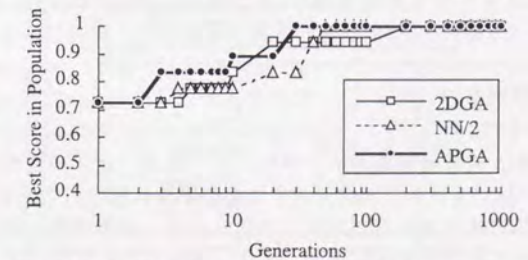


Fig. 3.20 (c) The relationship between the best fitness score and the number of generations for learning logic that partially includes XOR logic.
[Exp. 4] (Population size=70, seed=941230)

図 3.20 世代数と最大適応度の関係

図3.19および図3.20に、3.5節の問題 [実験3]および[実験4]を用いて、常に一つだけピラミッド構造を選択する場合、正規分布に従い選択するピラミッド数を変化させる場合、常に全体の半数のニューロン数分に対応したピラミッド構造を選択する場合に関して、個体総数を変化させて比較評価を行った結果を示す。これらの実験結果は、個体総数が十分に大きい場合には、常に一つだけピラミッド構造を選択する場合が有効であることを示している。これは、初期集団として十分な多様性が確保されている場合には、始めから2次元スキーマを破壊させる確率を抑える効果の高い交差を行うことが有効に作用したと予想される。一方、個体総数が少なくなるのに従い、正規分布に従い選択するピラミッド数を変化させる場合の有効性が顕著になる現象が観測された。正規分布に従い選択するピラミッド数を変化させる方法は初期集団として十分な多様性が確保されている場合にも比較的有効な結果を示しており、個体総数の変化に対してロバストな手法を提案するという狙いを満たす現実的な手法の一つと考えられる。但し、一般的な有効性を示すためには実験データが不足している。また、正規分布のパラメータ調整に関しても継続的な検討が必要と思われる。

一方、実世界を対象としたシステムでは、環境変化に依存して適応度が常に変化しているとみなすことができる。すなわち、ここで提案する適応度に依存して遺伝的操作の並列度を動的に変化させる手法は、環境変化に依存して遺伝的操作の並列度を動的に変化させる手法と捉えることもできる。このような適応度依存並列遺伝的操作は、特に、自律システム研究分野への適用が有効と思われる。また、第4章で示す、先読みモデルとの併用も今後の課題として重要と考える。

3.6.2 SIMD型並列計算機への実装案

次に、上記適応度依存型並列GAを、実際にSIMD (Single Instruction stream and Multiple Data stream)型並列計算機に実装する一実現手段に関して検討する。図3.21はニューロン番号 i ($i=1 \sim M$)をプロセッサ番号 i ($i=1 \sim M$)に対応させて実装する場合の例を示している。プロセッサ i ($i=1 \sim M$)のローカルメモリには、全ての個体（すなわち、ニューラルネットワーク）のニューロン i ($i=1 \sim M$)に関する結線情報と結合荷重値情報を格納する。また、全ての個体に対する適応度を格納する。ローカルメモリのポインタは個体番号に対応している。図3.21を基に、基本的な遺伝的操作である、淘汰、選択、交差、突然変異が並列計算機上でどのように実現されるかを簡単に示す。まず、淘汰について述べる。淘汰する個体番号は、ローカルメモリに格納している適応度を基に決定する。例えば、ある基準値以下の個体を淘汰の対象とする。実装の一手段としては、基準値をレジスタなどに予め格納しておき、演算器群を使って適応度との差をとった後の符号（正/負）から判定する。あるいは、ある一定の割合の個体（例えば、適応度が低い方

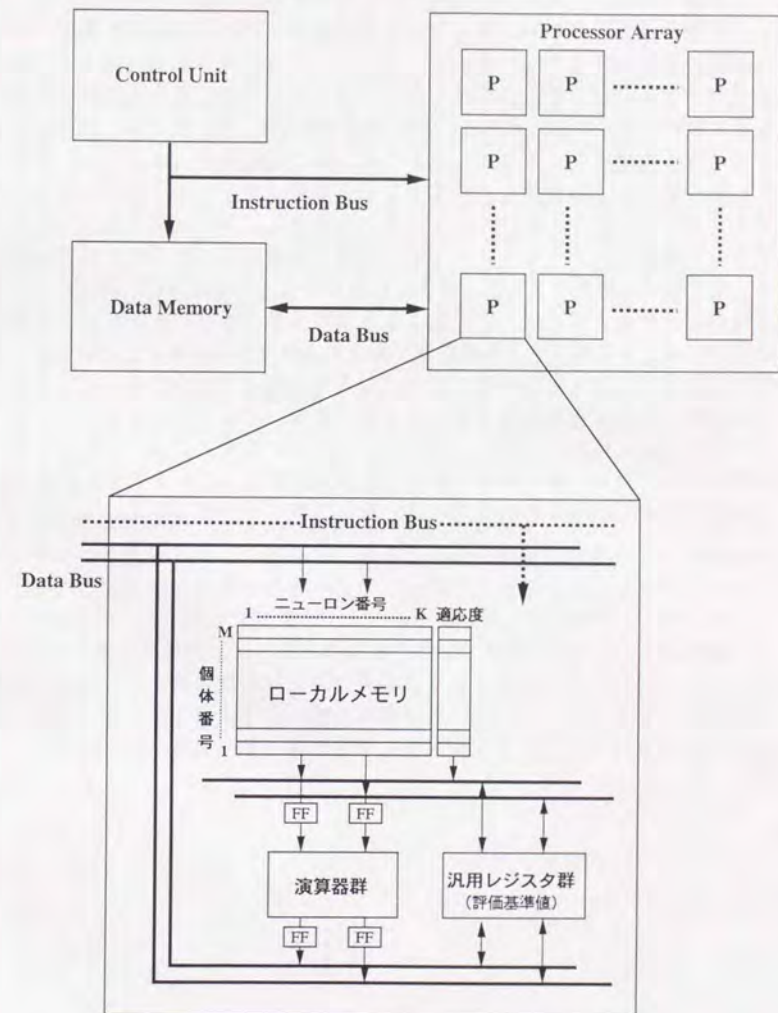


図2.2.1 並列計算機への実装例

から半数の個体)を淘汰の対象とする。淘汰する個体番号が決定したら、次に親となる個体番号を求める。親となる個体番号は、例えば適応度を基に、ルーレット選択により決定する。あるいは、全くランダムに決定する。淘汰する個体の番号と親となる個体の番号が決定したら、淘汰する個体に関する結線情報と結合荷重値の情報を親の個体のそれらの情報で置き換える。図3.21では、個体番号はローカルメモリのポインタとなっているため、親の個体番号(ポインタ)で示されるローカルメモリ上の情報を、淘汰する個体番号(ポインタ)で示されるローカルメモリにコピーする。次に、交差について述べる。2次元交差を行うために、ピラミッド構造の頂点となるニューロンの番号を並列度の数だけランダムに決定する。ニューロン番号はプロセッサ番号に対応させているために、選ばれたプロセッサ内のローカルメモリにおいて親となる個体番号で示される内容同士を演算器群を用いて交換して淘汰する個体番号で示されるローカルメモリに格納する。また、突然変異として、一定の確率でローカルメモリ内の情報をランダムに変更する。上記処理を淘汰する個体数を補うまで繰り返す。ここで示した方式は、並列度がプロセッサ数を越えない限り、選ばれたプロセッサ全てにおいて独立に並列実行可能である。但し、適応度の計算(ニューラルネットワークで考えると、出力信号の誤差計算)は個体数の数だけ繰り返し計算する必要がある。

3.7 まとめ

本章では、ニューラルネットワークの構造および結合荷重値の学習を目的とした表現型レベルで遺伝的操作を施す手法を提案し、その有効性を評価した。提案した手法は、ニューラルネットワーク自体を2次元の染色体とみなし、ニューラルネットワーク上の2次元部分構造を活かした交差を行うことを特徴とする。また、ニューラルネットワークの構造決定と結合荷重値の学習の両方同時に進化的計算を適用することを特徴とする。XOR論理を含んだ論理合成問題などを用いて、1次元染色体上で1点交差、2点交差あるいは一様交差を行う標準的な遺伝的アルゴリズムと比較し、以下の結論を得た。提案した表現型レベルでの遺伝的操作は、Exclusive-OR論理を含んだ論理合成問題や教師データの一部分が動的に変化する問題などに対して有効である。これによって、Exclusive-OR論理のような「だまし問題」(進化的計算での学習が困難とされている問題)やノイズなどの影響を考慮する必要のある実世界の動的な問題への進化的計算の効果的適用法の一例を示すことができたと考える。但し、今後、実数値表現を扱う問題での調査およびBPなどの学習アルゴリズムとの比較が必要である。

また、上記遺伝的操作を個体総数の変化に比較的影響されないロバストな手法に改良する一つの試みとして、適応度に依存して遺伝的操作の並列度を学習過程において動的に切り換える手法を提案した。実世界を対象としたシステムでは、環境変化に依存して適応度が常に変化しているとみなすことができる。すなわち、ここで提案する適応度に依存して遺伝的操作の並列度を動的に変化させる手法は、環境変化に依存して遺伝的操作の並列度を動的に変化させる手法と捉えることもできる。このような適応度依存並列遺伝的操作は、特に、自律システム研究分野への適用が興味深い。また、第4章で示す、先読みモデルとの併用も今後の課題として重要と考える。

第4章 自律システム制御のための先読みモデル の進化的獲得

要旨

この章では、リカレント型ニューラルネットを基本とした先読みモデルを提案し、その有効性を検討する。先読みモデルは、行動を決定するネットワークと環境の内部モデルからなり、それぞれリカレント型ニューラルネットで構成する。環境の内部モデルを学習により構築しながら先読みを行い、投機的な行動を実際に環境に対して行う。また、先読み結果を、行動を決定するネットワークの学習に利用する。すなわち、2つのネットワークは、相互作用を持ちながら学習を行う。ニューラルネットワークの構造およびウェイトの学習には、遺伝的アルゴリズムを適用した。

このモデルの評価には、最適解が確率的に変化する離散的問題を用いた。具体的には、tic-tac-toeゲームを用いて有効性の評価を行い、以下の結論を得た。

(1) 環境の内部モデルを用いた先読みを行うことにより、実際の環境を介した学習回数の削減につながる可能性を持つ。(2) ある程度学習が進んでから、内部モデルを用いた先読みを行うべきである。(3) 内部モデルの学習方式として、ニューラルネットワークを用いてアルゴリズムを学習する方式の方が、入出力の対応表をClassifier Systemを用いて学習する方式よりも、先読み効果が高い可能性を持つ。

これによって、行動を決定するために十分な情報が予め得られない状況下での行動制御を対象として、先読みモデルの有効可能性がしめされた。また、最適解が確率的に変化する離散空間上の動的問題を対象とする場合、単純に進化的計算をClassifier Systemとして適用するよりも、進化的計算とニューラルネットワークの融合により学習が効果的に行える可能性を示唆している。

4.1 はじめに

今日、コンピュータは、解法のためのアルゴリズムが存在しプログラム言語で明確に記述可能な良設定問題を解く能力では、人間をはるかに上回る。一方、解くべき多くの問題が不良設定でありアルゴリズムとして記述困難な実世界における情報処理能力に関しては、人間と比べて依然として柔軟ではない。この問題を解決する一つの可能性として、適応システムや機械学習の研究が考えられる。しかしながら、実世界における多くの問題では、適応システムや機械学習で必要となる試行回数や学習時間が十分与えられるとは限らない。十分な試行回数を得る前に何らかの行動を必要とされる場合が考えられる。すなわち、学習過程における部分的な情報を基に有効な行動を行う必要がある。

この問題の解決策の一つとして、Holland[Holland 91]やRiolo[Riolo 91]はClassifier System (CS)[Holland 75]を用いた先読みの利用を提案している。しかしながら、CSでは、入力と出力の対応表という形で内部モデルを構成するために、静的環境における先読みが基本となり、学習過程における深い先読み（数手先まで読むこと）や動的に変化する問題を扱うことには適さない。なぜなら、CSでは、未学習の入力データに対する適応能力（すなわち、出力の信頼度）が学習過程で向上しないからである。

この問題を解決するために、本稿では、リカレント型のニューラルネットワークを基本とした先読みモデルを提案し、その有効性を検討する。すなわち、行動を決定するネットワークと環境の内部モデルをそれぞれリカレント型のニューラルネットワークで構成し、学習過程における先読みを有効に活用することを考えた。リカレント型ニューラルネットワークは、非線形関数の学習を行う、規則的な処理の学習からカオスダイナミックスの学習[Sato 90]まで幅広い適応能力を持つ、外界を介したフィードバックループを用いた学習が可能であるなど、適応・進化システムに有効と思われる性質を有するからである。以下、4.2節では、提案する先読みモデルの概要を示す。4.3節では、遺伝的アルゴリズム (GA) [Goldberg 89]を用いたニューラルネットワークの構造および結合荷重の学習方法を示す。4.4節において、tic-tac-toeゲームを用いて、このモデルの有効性の評価を行ない[Sato 94b]、最後に考察とまとめを行う。

4.2 先読みモデルの提案

4.2.1 基本とするリカレントネットワークの特徴

本章で扱うリカレント型ニューラルネットワークは、自己帰還ループの他に、環境（外界）を介した帰還ループを持つことを特徴とする。図4.1は、例えばフィードフォワード型のニューラルネットワークに時間遅れのある自己帰還ループを加えたJordan型の時系列発生モデル[Jordan 89]（図中の点線で囲んだ部分）に、環境を介した帰還ループを付加した場合の例である。図4.1のニューラルネットワーク部が N 個のニューロンを持つ単一層から構成される場合には、離散時間モデルの動作方程式は以下の式で表わすことができる[Hect 92]。

$$X_i(t+1) = f \left[\sum_{j=1}^n w_{ij} Y_j(t) + \sum_{j=n+1}^{n+N} w_{ij} X_{j-n}(t) \right] \quad (4.1)$$

$$Y_j(t) = F[X_1(t), X_2(t), \dots, X_N(t)] \equiv F[X(t)] \quad (4.2)$$

ここで、 $X_i(t)$ 、 $Y_j(t)$ はそれぞれ、 i 番目のユニットの時刻 t での出力、外部入力を表し、 w_{ij} は j 番目のユニットから i 番目のユニットへの結合荷重を表す。また、 f はシグモイド関数、しきい関数などの適当な関数、 F は環境からの影響を表す未知の関数である。すなわち、時刻 $t+1$ の出力 $X(t+1)$ は、時刻 t の出力 $X(t)$ および時刻 t の出力に対する環境（外界）からの応答 $Y(t) = F[X(t)]$ から決定される。

4.2.2 行動を決定するネットワークと内部モデルの相関学習

本章で提案するモデルは、先読みを実現するために、行動を決定するネットワーク (Action Network) の他に、環境の内部モデル (Internal Model) を持つ。いずれも、前記リカレント型のニューラルネットワークを基本とする。内部モデルも含めた基本的構成を図4.2に示す。行動を決定するネットワークと内部モデルは相互結合関係にある。内部モデルも N' 個のニューロンを持つ単一層から構成されると仮定すると、先読みモデルの動作方程式は以下の式で表すことができる。

$$X_i(t+1) = f \left[\sum_{j=1}^n w_{ij} Y_j(t) + \sum_{j=n+1}^{n+N'} w_{ij} Y'_{j-n}(t) + \sum_{j=n+N'+1}^{n+N'+N} w_{ij} X_{j-n-N'}(t) \right] \quad (4.3)$$

$$Y'_j(t) = F' \left[\sum_{k=1}^N w'_{jk} X_k(t) + \sum_{k=N+1}^{N+N'} w'_{jk} Y'_{k-N}(t-1) \right] \quad (4.4)$$

ここで、 $Y'_j(t)$, $Y_j(t)$ はそれぞれ、内部モデルを構成する j 番目のユニットの時刻 t での出力、および式(2)で定義した外部入力である。 F' はシグモイド関数、しきい関数などの適当な関数を表す。提案した先読みモデルでは、環境との応答を通して、環境の内部モデルを自己学習しながら先読みを利用する。また、先読み結果に従い、投機的な行動を（実際に外界に対して）行う。先読み結果は、行動を決定するネットワークの学習にも利用する。すなわち、本稿で提案する先読みモデルは、行動を決定するネットワークと内部モデルが相互作用を持ちながら成長するモデルとなっている。

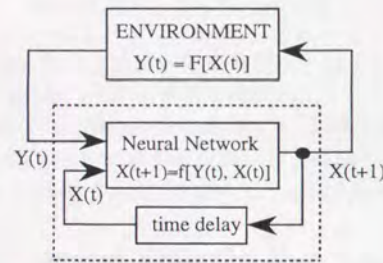


図4.1 基本とするリカレント型ニューラルネットワークの例

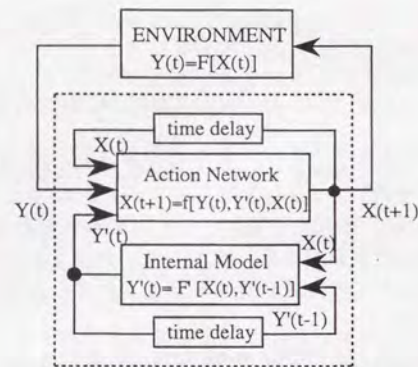


図4.2 提案する先読みモデル

4.3 進化的計算を用いた先読みモデルの学習法

ここでは、GAによるニューラルネットワークの構造および結合荷重の決定を内部モデルの合成に適用する。これは、GAが持つグローバル・サンプリングの特徴を活かして、局所解に落ち込むことを防ぐことを狙ったためである。GAを用いたニューラルネットワークの学習に関しては、既に幾つかの研究報告がされている[Miller 89, Kitano 90, Nagao 93]。以下、行動を決定するネットワークの学習方式および内部モデルの学習方式について示す。

4.3.1 行動を決定するネットワークの学習

行動を決定するネットワークでは、時刻 t の出力（状態） $X(t)$ 、時刻 t の出力に対する環境からの応答（入力） $Y(t)$ と時刻 $t+1$ の出力（行動） $X(t+1)$ の関係を学習する。今回実験に使用したネットワーク構造を図4.3に示す。

図4.3では、前章で説明したリカレントネットワークを空間的に展開した、2層構造を用いて学習を行っている。入力層16ニューロン、出力層8ニューロンの合計24ニューロンから構成される。入力層の中で、8ニューロンは状態記憶に用い、残り8ニューロンは入力信号の記憶に用いる。出力層のニューロンはしきい関数により入出力変換を行う。出力 $X(t+1)$ は、基本的には、前節の式(4.1)に従う。出力信号が"1"となるニューロンの番号が行動に対応する。但し、出力層を構成するニューロンの出力が複数個"1"となる場合には、以下の式により、あるニューロン番号を1個だけ選択する。例えば、8個のニューロンの全ての出力が"1"の場合、 i 番目のニューロン番号が選択される確率 P_i は以下の式で定義する。ただし、ネットワークの初期状態は、4.2.1節で述べるように、明示的に与える。

$$P_i = W_i / \sum_{i=0}^7 W_i \quad (4.5a)$$

$$W_i = \left[\sum_{j=0}^7 w_{ij} Y_j(t) + \sum_{k=0}^7 w_{ik} X_k(t) \right] \quad (4.5b)$$

学習は2種類の方法で行う。第1は、内部モデルを用いた先読み結果から予測される勝敗による評価を行い、選択したニューロン番号に対応する重み値 w_{ij} の増減を行う。第2は、あるニューロン番号を選択したことにより実際にゲームに勝ったか、負けたかという評価を行い、選択したニューロンに対応した重み値 w_{ij} の増減を行う。すなわち、実際のゲームの勝敗および先読み結果の勝敗を強化信号とする、強化学習を行う。

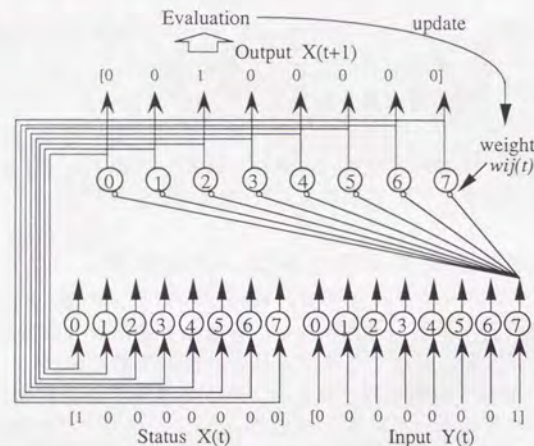


図4.3 行動決定ネットワークの学習方式の例

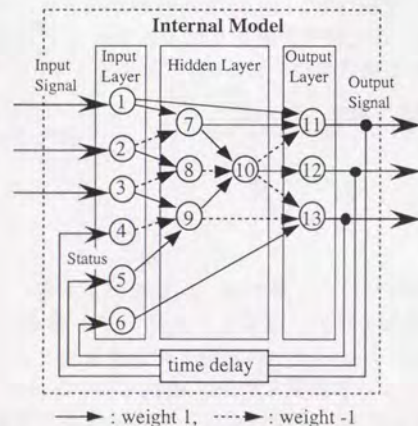


図4.4 ニューラルネットワークを用いた内部モデルの例

4.3.2 内部モデルの学習

ここでは、2種類の方式に関して検討を行った。いずれも、前節で説明したりカレントネットワークを空間的に展開した多層構造を用いて学習を行う。

4.3.2.1 学習方式1：ニューラルネットワーク方式

まず、学習方式1で扱うニューラルネットワークの条件を示す。一例を図4.4に示す。○印中に通し番号を付けたシンボルがニューロンを表す。

- (i) 各ニューロンは、自分自身の状態として0（非励起状態）または1（励起状態）のいずれかをとる。ある時点における入力信号の総和が0よりも大きい場合は自分自身の次の状態を1にし、それ以外の場合は次の状態を0にする。
- (ii) 回路網は、入力層、隠れ層群、出力層から構成される層状結合型のニューラルネットワークである。入力層は、環境からの入力信号および自己帰還信号から構成される。
- (iii) ニューロン間の結合荷重は1、-1または0のいずれかをとる。0の時は、ニューロン間に結合がないことを表す。

次に、GAを適用するために、回路網の構造と結合荷重を表す染色体を定義する。ここでは、図4.5に示すように、各ニューロン間の遷移状態を表す行列M1, M2, M3を各層ごとに考え、これらの遷移行列から構成される列[M1, M2, M3]を染色体とする。行列の各要素は、あるニューロンから別のニューロンへの結合荷重（1、-1または0）を表している。一般に、 m 層からなるネットワークでは、遷移行列は全部で $m-1$ 個存在する。この $m-1$ 個の遷移行列から構成される列を一つの染色体と考える。GAでは、通常、1次元の染色体を定義し遺伝オペレータの操作を行う。しかし、1次元の染色体では、ニューラルネットワーク上のある2次元部分構造が非常に有効だとしても、その構造が染色体上では分散されてしまい、遺伝オペレータにより壊されてしまう確率が高い。そこで、各層ごとの2次元遷移行列から構成される列を染色体として定義した。

4.3.2.2 学習方式2：入出力の対応表を学習する方式

学習方式2では、入力から出力への2次元遷移行列を染色体と考える。入力または出力として取りうる状態をそれぞれ異なるユニットに割り当てることにより、CSと同値になる。図4.6に一例を示す。図4.6は、時刻 t の時の出力 "7" に対する "1" ~ "6" の入力（応答）に対して、それぞれ "1" ~ "7" のいずれを出力するかの対応表を学習する場合の例である。

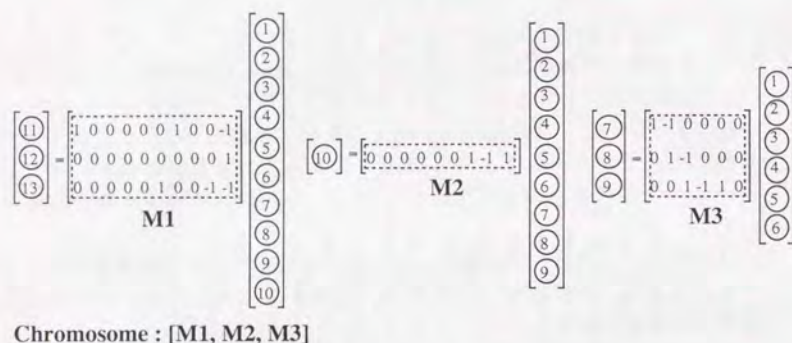


図4.5 図4.4のネットワークに対応した遷移行列

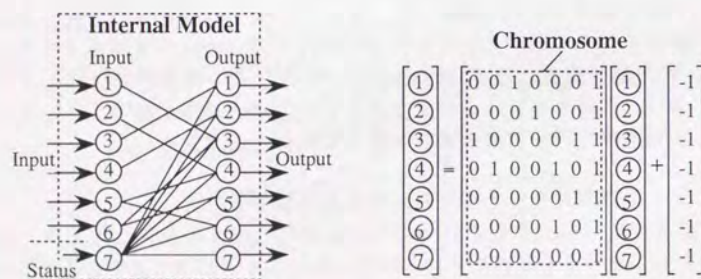


図4.6 入出力対応表を学習する内部モデルの例と対応する遷移行列

4.4 Tic-Tac-Toeゲームを用いた評価

4.4.1 実験方法

4.4.1.1 ゲームの設定

tic-tac-toeゲームを用いてモデルの評価を行った。ルールは以下の通りである。

(ルール) : 2人が交互に自分のシンボル、すなわち○とX、を3行3列のマトリックスの中に置いていく。縦、横、斜めのいずれかの方向に、先に自分のシンボルを3つ連続して置いた方が勝ちである。

遺伝オペレータを適用するために、図4.7に示すように、中央を除く3行3列のマトリックス上の各位置に、"0" ~ "7" (すなわち、"000" ~ "111") までの位置番号を設定した上で、解析を簡単化するために、以下の仮定の基に評価を行った。

(1) 先手は、図4.7に示すように、自分のシンボル○を最初に中央に置く。

後手は、自分のシンボルXを、最初に位置 "0" ("000") に置く。

(2) 先手は、熟練者であり、負けることはない。

(3) 後手(シンボルX)は、先手のアルゴリズムを知らない。

(4) 後手は、初期状態として、上記ゲームのルールから以下の知識を予め得ている。

(知識1) : 自分の番の時に、自分のシンボルを3個続けて一直線上に置ければ置く。

(知識2) : (知識1)が適用できない時、もし次に相手が、相手のシンボルを3個続けて一直線上に置けるようなら阻止する。

(知識3) : (知識1), (知識2)共に適用できない場合は、空いている場所にランダムに打つ戦略をとる。

以上の仮定の基に、先手を環境と考え、後手に対して図4.2に示した先読みモデルを適用した。すなわち、先手には負けることのないアルゴリズムを知識ベースとして与える。後手は、(知識1) ~ (知識3)を行動を決定するネットワークの初期状態として、ゲームの試行を通して、先手のアルゴリズムを内部モデルとして学習しながら、行動を決定するネットワークの内の(知識3)の戦略のみ更新する。

4.4.1.2 ゲームの流れと教師データ

上記仮定に基づくゲームの流れを図4.8に示す。第3手目に先手は、空いている場所の中から、ランダムに選択する設定とした。第3手目に先手が "1" ~ "6" の位置に自分のシンボルを置いた場合、ゲームの結果は引き分けとなる。すなわ

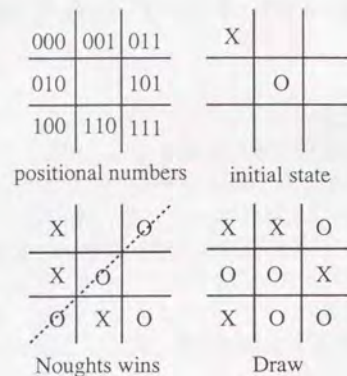


図4.7 tic-tac-toeゲームの説明図

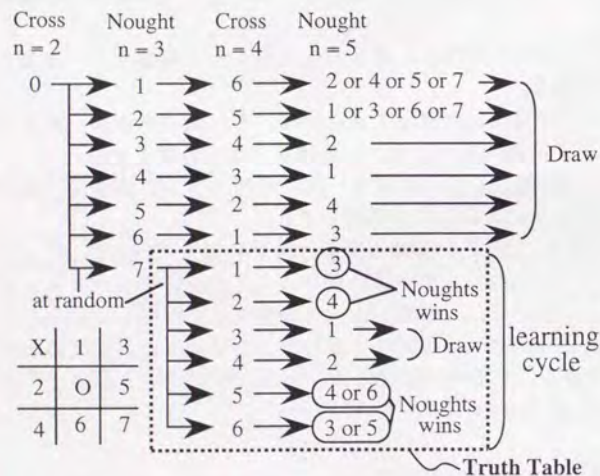


図4.8 (知識1) ~ (知識3) に基づくtic-tac-toeゲームの流れ

ち、後手の行動を決定するネットワークの更新は行われない。第3手目に先手が"7"の位置に自分のシンボルを置いた場合のみ勝敗が存在し、行動を決定するネットワーク中の(知識3)の更新が行われる。すなわち、行動を決定するネットワークと内部モデルは、それぞれ以下の学習を行う。行動を決定するネットワークは、4.3.1で示した学習方法に従い、"7"の手に対して"3"または"4"の位置に自分のシンボルを打つことを学習する。内部モデルは、先手の"7"の手に対する自分(後手)の"1" ~ "6"の応答に対して、先手が次に返してくる手(すなわち、点線内の真理値表)を学習する。ここで、"5"と"6"に対する先手の次の手はそれぞれ2種類存在し、ゲームごとにランダムに変化するアルゴリズムに設定した。すなわち、内部モデルの学習ターゲットとなる先手のアルゴリズムには不確定性を持たせる設定とした。

4.4.2 ネットワークの初期設定とGAの適用方法

4.4.2.1 行動を決定するネットワークの初期設定

図4.3に示した、入力層16ニューロン、出力層8ニューロンから構成される2層構造を用いて実験を行った。初期状態は、(知識1)、(知識2)および(知識3)の初期戦略に対応した構造を明示的に与えた。すなわち、状態"0"(2手目)の時、入力"1" ~ "6"(3手目)に対して出力"6" ~ "1"(4手目)、入力"7"(3手目)に対して出力(4手目)はランダムに選択という対応(図4.8に記述)を初期状態とした。また、状態"0"、入力"7"と出力"1" ~ "6"の間の重み値 w_{ij} を、3.1に示す学習方式で更新した。入力"1" ~ "6"と出力の間の重み値情報((知識1)と(知識2)に対応)はゲームの間更新されない。

4.4.2.2 内部モデル獲得の為のGAの適用方法

a. 学習方式1の場合

まず適応度について示す。適応度は、内部モデルを構成するニューラルネットワークが与えられた問題にどの程度適しているかを示す必要がある。すなわち、ある入力に対してどの程度正しい出力が得られるかを示す必要がある。ある程度十分な試行回数を得られることを前提とした学習の場合は、数十~数百回の試行のうち何回出力値が期待値と一致したかを適応度と定義することも考えられる。しかし、ここでは十分な試行回数を得られない場合の学習問題を対象としているため、1回のゲームの試行において、予測した手(すなわち、内部モデルを構成するニューラルネットワークの出力)と実際の相手の手(すなわち、先手の応答)が何ビット一致していたかを適応度とした。一致していたビット数を n_c とすると、個体 I_i の適応度 $f(I_i)$ は次式で定義される。

$$f(I_i) = n_i \quad (4.6)$$

次に遺伝オペレータの設定について示す。ある世代の個体群から次世代の個体群を生成するための方法として、適応度の下位の一定の割合を淘汰し、ルーレット選択による子孫を補って個体総数を一定に保った。すなわち、淘汰される個体の数を M 個とすると、今の世代の N 個の個体 $I_1 \sim I_N$ から二つの個体のペアを M 組だけ重複を許してランダムに選択し、それぞれに対して交差と突然変異を施して不足を補うための M 個の個体を決定した。ここで、ある個体 I_i が次の世代の個体を作るための親として選択される確率 $P(I_i)$ は、次式によって定義した。

$$P(I_i) = f(I_i) / \left\{ \sum_{j=1}^N f(I_j) \right\} \quad (4.7)$$

交差は、染色体を構成する遷移行列中のランダムに選んだある行を、ペアとして選んだ個体の遷移行列の対応する行と置き換える方式とした。行の置き換えを中心とした交差を行った理由は、学習途中で出来たニューラルネットワーク上の有効な2次元部分構造を交差により破壊する確率を抑えるためである[Sato 95a]。学習方式1では、6層、2-2ニューロン構造（入力層から順に、6-7-2-2-2-3ニューロン）を初期状態とした。状態、入力、出力をそれぞれ3ビットで表現し、 $f(I_i)=1$ 以下の個体を淘汰した。個体数 N は100、突然変異率は0.01に設定した。上記遺伝的操作は、全ての個体の適応度が2以上になるまで、繰り返し行った。

b. 学習方式2の場合

学習方式2における適応度および遺伝オペレータの設定は基本的に上記学習方式1と同一に設定した。但し、図4.6に示すように、学習方式2では入力7ユニット、出力7ユニットで構成した上で、取りうる手をそれぞれ異なるユニットに対応させている。従って、状態1ビット、入力6ビット、出力7ビットで表現し、 $f(I_i)=5$ 以下の個体を淘汰した。交差は、学習方式1との対応から、遷移行列中のランダムに選んだある行をペアとして選んだ個体の遷移行列の対応する行と置き換える方式とした。上記遺伝的操作は、全ての個体の適応度が6以上になるまで、繰り返し行った。

4.4.3 実験結果

以下の4つのケースに関して実験を行った。

(実験1)：(知識3)の学習による更新を、先読み無しで、行う場合。内部モデルは使用しない。

(実験2)：(知識3)の学習による更新を、先読みも利用して、行う場合。

但し、内部モデルの学習は前章で述べた、学習方式1を用いる。

(実験3)：(知識3)の学習による更新を、先読みも利用して、行う場合。

但し、内部モデルの学習は前章で述べた、学習方式2を用いる。

(実験4)：(知識3)の学習による更新を、先読みも利用して、行う場合。

但し、内部モデルとして、先手(シンボル○)の完全なアルゴリズムを最初から使用する。

ゲームの試行回数と後手がそれまでに負けた試合総数の割合との関係を、幾つか乱数の初期値を変えて求めた。結果を、図4.9～図4.11に示す。

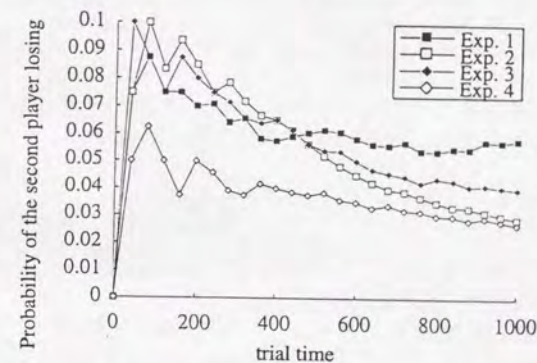


図4.9 後手が負ける確率(累積平均)と試合数の関係

Fig. 4.9 Probability of second player losing as a function of the number of trial games played. (seed = 7301)

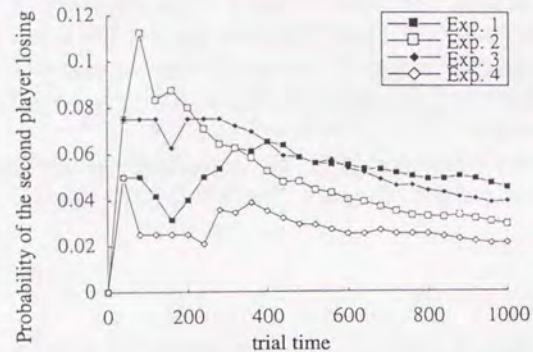


図4.10 後手が負ける確率（累積平均）と試合数の関係
Fig. 4.10 Probability of second player losing as a function of the number of trial games played. (seed = 94417)

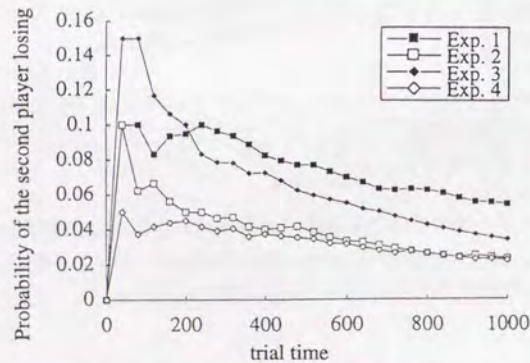


図4.11 後手が負ける確率（累積平均）と試合数の関係
Fig. 4.11 Probability of second player losing as a function of the number of trial games played. (seed = 1994629)

4.5 考察

実験結果から、以下のことが考えられる。(実験1)と他の3つの実験結果の比較から、今回扱った問題に対しては、内部モデルを用いた先読みが勝率向上に有効に作用していたと考えられる。内部モデルを用いた先読み結果を学習に利用することにより、学習に必要な試行回数を削減できる可能性を持つ。(実験2)と(実験3)では、学習開始直後は先読み効果にバラツキがあるが、試行回数が進むにつれて安定して高い効果を維持する。一方、(実験4)では学習開始直後すぐに高い先読み効果を示している。これは、内部モデルの学習が進み、先読みの信頼性が向上して初めて安定した効果が得られるためと推測される。従って、工学的な応用を考える場合は、内部モデルの学習進行状況をチェックし、途中から先読み結果を学習に利用するように切り換える方式が有効である。(実験2)と(実験3)では、(実験2)の方が高い先読み効果を示す傾向にあった。すなわち、多層型ニューラルネットワークを用いてアルゴリズムを学習するタイプの学習方式1が、CSを用いて入出力の対応表を学習するタイプの学習方式2よりも、高い先読み効果を示す傾向にあった。この原因を調査するために、内部モデルの学習ターゲットの詳細な調査、および、(実験2)と(実験3)で得られた内部モデルの解析を行った。表4.1は、学習ターゲットとなる真理値表の入/出力をそれぞれ3ビットで表した例である。表4.1から、内部モデルは以下のブール式を学習することになる。

(1) "5" --> "4", "6" --> "3" の時

$$y_1 = \overline{x_1}x_2\overline{x_3} + x_1\overline{x_2}x_3, \quad y_2 = \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 \\ y_3 = \overline{x_1}x_3 + x_2(x_1 \oplus x_3)$$

(2) "5" --> "4", "6" --> "5" の時

$$y_1 = x_2\overline{x_3} + x_1(x_2 \oplus x_3), \quad y_2 = \overline{x_2}(x_1 \oplus x_3) \\ y_3 = \overline{x_1}x_3 + x_2(x_1 \oplus x_3)$$

(3) "5" --> "6", "6" --> "3" の時

$$y_1 = \overline{x_1}x_2\overline{x_3} + x_1\overline{x_2}x_3, \quad y_2 = \overline{x_1}x_2 + \overline{x_2}x_3 + x_1\overline{x_3} \\ y_3 = \overline{x_1}x_3 + x_2(x_1 \oplus x_3)$$

(4) "5" --> "6", "6" --> "5" の時

$$y_1 = x_2\overline{x_3} + x_1(x_2 \oplus x_3), \quad y_2 = \overline{x_2}(x_1 + x_3) \\ y_3 = \overline{x_1}x_3 + x_2(x_1 \oplus x_3)$$

すなわち、内部モデルの学習ターゲットは排他的論理和 (XOR) を部分的に含んでおり、内部モデルは、ターゲットとなる真理値表のランダムな変化に伴い、XOR論理の生成と消滅を動的に繰り返す必要があったことが分かる。図4.12に(実験2)で得られた内部モデルの一例を示す。学習方式1で得られた内部モデルは、XOR論理(すなわち、GA-deceptiveな論理[Goldberg 89])を確実に獲得していることが分かる。このことから、動的に変化する非線形論理の学習には、入出力の対応表をCSで学習する方式よりも、ニューラルネットを用いて学習する方式の方が有効であると推測される。

表4.1 内部モデルの真理値表

x1	x2	x3	y1	y2	y3
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	0	0	1
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	0	1	1

XOR

XOR

XOR

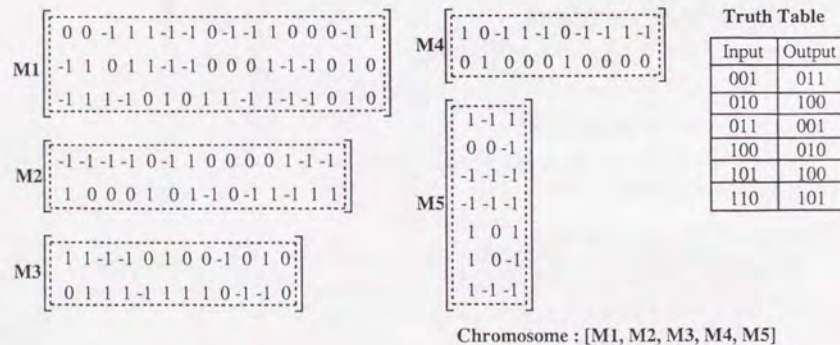


図4.12 ニューラルネットワークを用いた内部モデルで得られた遷移行列の例
(簡略化のため、自己帰還ループの情報は省略している)

4.6 まとめ

本章では、リカレント型ニューラルネットワークを基本とした先読みモデルを提案し、その有効性を検討した。行動を決定するネットワークと環境の内部モデルをそれぞれリカレント型ニューラルネットワークで構成し、内部モデルを学習により構築しながら先読みを利用する。先読み結果は、行動を決定するネットワークの学習回数の削減に活用する。tic-tac-toeゲームを用いて有効性の評価を行い、以下の結論を得た。(1) 環境の内部モデルを用いた先読みを行うことにより、実際の環境を介した学習回数の削減につながる可能性を持つ。(2) ある程度学習が進んでから、内部モデルを用いた先読みを行う方が効率的である。(3) 今回の実験結果では、内部モデルの学習方式として、ニューラルネットワークを用いてアルゴリズムを学習する場合の方が、CSにより入出力の対応表を学習する場合よりも、先読みの効果が高かった。今後、より探索空間が広く、深い先読みを必要とする大規模問題での調査が必要である。

第5章 カオスダイナミックスを用いた進化的計算の 適応力評価実験

要旨

この章では、連続実数空間の動的問題に対して、進化的計算とニューラルネットワークの融合モデルの持つ学習能力を探るための基礎的な実験を行う。すなわち、カオスダイナミックスを学習可能なリカレントニューラルネットワークを進化的計算を用いて合成する一実現手段を提案し、その有効性を検討する。教師信号として用いるローレンツ軌道は常にカオス的に変化し続ける非周期時系列軌道である。したがって、教師信号の波形そのものの学習が完了することはない。ここでは、進化的計算により合成する学習過程のリカレントニューラルネットワークの出力が、不動点、リミットサイクル、ローレンツアトラクタと変化することを計算機実験により検証する。すなわち、進化的計算により合成したリカレントニューラルネットワークが、カオスの隠れた秩序を学習できることを示す。現段階では、Back Propagation Through Time (BPTT) アルゴリズムを用いた場合と比較して、収束性、再現性ともに劣っている。しかし、連続実数空間の動的問題を対象として、進化的計算の適用法の一例を示すとともに、進化的計算の学習能力を探る上で貴重な実験結果を得ることができたと考える。

5.1 はじめに

我々は、複雑かつ動的に変化する環境に対して自律的に適応・進化するシステムの実現を目的として、複数のリカレントニューラルネットワークが相互作用を持ちながら学習を行うモデルの研究を行っている[Sato 94b, Sato 95b, Sato 95c]. なぜなら、リカレントニューラルネットワークは周期的な軌道の学習からカオスダイナミックスの学習[Sato 90]まで幅広い適応能力を持つため、複数のリカレントニューラルネットワークの相互作用を考えることにより、今までにない動的情報処理を実現する可能性があると考えからである。あるいは、動的で複雑な問題に対応するには大規模なリカレントニューラルネットワークが有効になる可能性が高いと考え、問題に適した特別な構造を持たない完全結合型リカレントニューラルネットワークよりも、問題の特性を考慮した疎結合部分が学習効率の向上に貢献すると考えるからである。

一方、リカレントニューラルネットワークのような非線形力学系では、パラメータのわずかな変化に対して、システムの振舞いが不連続に変化する分岐という現象が起こることが知られている[Guckenheimer 83]. すなわち、ネットワークの構造決定および学習率などのパラメータ調整は難しい問題であり、学習にも多くの時間を必要とする。したがって、このような設計者の負担を低減することを目的として、近年、Genetic Algorithms (GA)などの進化的アルゴリズムを用いたりリカレントニューラルネットワーク合成の研究が始まっている。例えば、N. Saravanan[Saravanan 93]はsunspot prediction problemを対象としたリカレントニューラルネットワークの合成に進化的アルゴリズムを適用している。P. Angelineら[Angeline 94]はfinite state automataの学習を対象としたリカレントニューラルネットワークの合成に進化的アルゴリズムを適用している。J. Santosら[Santos 94]は5-bit even parity problemを対象としたリカレントニューラルネットワークの合成に進化的アルゴリズムを適用している。しかし、これらの問題は必ずしもリカレントニューラルネットワークを必要としない。フィードフォワード型のニューラルネットワークでも学習可能な問題である。

ここでは、カオスダイナミックスを対象として、進化的アルゴリズムの学習能力を探る基礎的な実験を行う。すなわち、カオスダイナミックスを学習可能なリカレントニューラルネットワークを進化的アルゴリズムを用いて合成する一実現手段を提案し、その実現可能性を計算機実験により検証した結果について報告する。以下、5.2節では、カオスダイナミックスの一例としてローレンツ軌道とその特徴を示す。5.3節で、実験で用いた非周期軌道の学習法を示し、5.4節において、進化的アルゴリズムを用いたりリカレントニューラルネットワーク合成法に

関して述べる。5.5節において計算機実験結果と考察を示し、最後にまとめを行う。また、付録として、我々が開発したニューロコンピュータ上に構築したリカレントシミュレータを用いた学習結果を示す。学習アルゴリズムは、高速性を考慮して、Back Propagation Through Timeアルゴリズムを採用した。

5.2 ローレンツアトラクタ

カオスダイナミクスの例として、ローレンツ軌道を取り上げる。対流のモデル [Lorenz 63] として古くから知られていること、および、Back Propagation Through Time (BPTT) アルゴリズム [Pearlmutter 89] を用いた学習例 [Sato 90, Sato 95f] が既に報告されているため、進化的アルゴリズムによるカオスダイナミクスの学習能力を探る上で適当な例題と判断したためである。ローレンツ方程式は外部入力のない3自由度の自律系で、運動方程式は次式で与えられる。

$$\frac{dx}{dt} = 10(x - y) \quad (5.1a)$$

$$\frac{dy}{dt} = -y + (28 - z)x \quad (5.1b)$$

$$\frac{dz}{dt} = -\frac{8}{3}z + xy \quad (5.1c)$$

この方程式をEuler法により離散的な時系列データに変換した軌道を図5.1に示す。時間の刻み幅 dt は0.01とした。図5.1は無限に続く軌道の一部 (2000点分) を表示している。ローレンツ方程式の解軌道の最大リアプノフ指数の値は0.90であり、解軌道が不安定なカオス状態にあることを示している。例えば、初期値を1%だけずらした異なる2つの解軌道の時間波形を図5.2に示す。2つの解軌道は時間とともに指数関数的に離れて行き、ある一定時間以上たつと2つの軌道は全く異なる時間的な振る舞いをする。すなわち、ある時刻での状態変数 (x, y, z) の値が分かっても、この値に対するわずかな誤差が時間とともに拡大され、ある一定時間以上先の解軌道を予測計算することは原理的に不可能である。

一方、カオス軌道は単なるランダム運動ではなく、隠れた秩序を持っている。例えば、ローレンツ方程式のカオス軌道を3次元空間の中で描いてみると、どのような初期値から出発しても、時間がたてば必ずローレンツアトラクタと呼ばれる領域の中に収束して行き、以後このローレンツアトラクタの中を永遠に動き回ることが知られている。ローレンツアトラクタを2次元平面 $x-y$, $y-z$, $z-x$ に投影した一例を図5.3に示す。

以下、進化的アルゴリズムによって合成したりカレントニューラルネットワークが、ローレンツアトラクタを形成するという、カオスダイナミクスの隠れた特徴を学習できる能力があるかどうかを計算機実験により検証する。

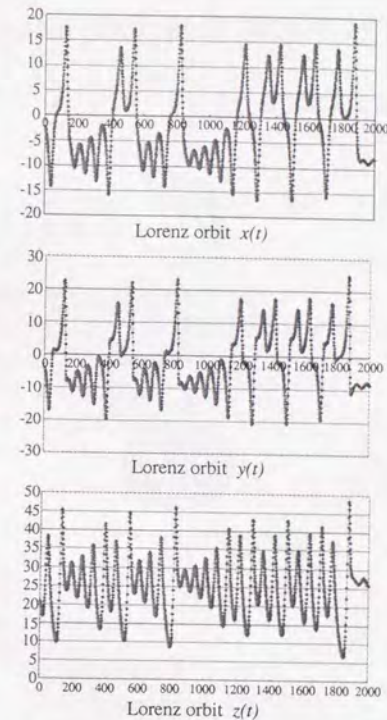


図5.1 ローレンツ軌道の一部

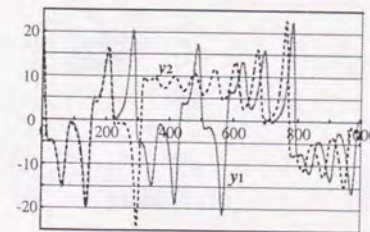
Fig. 5.1 Lorenz orbit $x(t)$, $y(t)$, $z(t)$. ($dt=0.01$, $t=0$ to 20)

図5.2 1%ずれた初期値から始めた2つのローレンツ軌道

Fig. 5.2 Lorenz orbit $y1$ & $y2$. ($dt=0.01$, $t=0$ to 10)

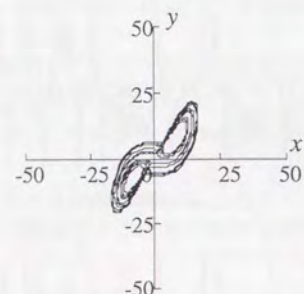
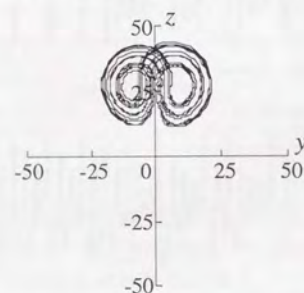
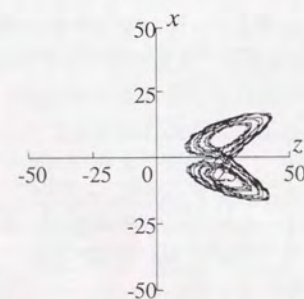
(a) $x - y$ plane(b) $y - z$ plane(c) $z - x$ plane

図5.3 ローレンツアトラクタの例

5.3 非周期軌道の学習法

非周期軌道を学習するため、エラーを評価する区間をある長さ TB に限り、この区間を $TF (< TB)$ ずつずらしてゆく方式[Sato 90]を採用した。学習方法を以下に示す。

- 時刻 T から $(T+TB)$ までリカレントニューラルネットワークの出力と教師信号との平均2乗誤差の累積を求める。
- 次章で示す進化的手法により、リカレントニューラルネットワークの構造とニューロン間の結合荷重値の更新を行う。
- 新しい結合荷重値と隠れユニットの初期値を用いて、時刻 TF 後の隠れユニットの出力を計算する。
- 時刻 T を TF だけ進めて(a)に戻る。

上記方法は、無限に続く非周期時系列データを対象とした実時間学習則である。無限に続く非周期時系列データを学習対象としているため、教師信号の波形そのものの学習が完了することはない。教師信号を生成している時間発展則を学習することになる。

5.4 進化的アルゴリズムによるニューラルネットワークの合成

ここでは、表現型レベルでの遺伝的操作を用いてニューラルネットワークの合成を行う。1次元染色体上で1点交差、2点交差あるいは一様交差などを行う標準的なGAよりも、ニューラルネットワークの合成に有効という実験結果[Sato 95a, Sato 95e]を得ているためである。

5.4.1 染色体の定義

表現型レベルでの遺伝的操作では、ニューラルネットワーク自身を2次元の染色体と定義する。ここでは図5.4に示すように、シグモイドユニットとダイナミックユニットから構成されるネットワークを考える。そして、シグモイドユニットダイナミックユニット間のニューラルネットワークとダイナミックユニットーシグモイドユニット間のニューラルネットワークをそれぞれ別の染色体IA, IBと定義する。前者は主に3つの出力x, y, zに対応したモジュール化構成に関連し、後者はこれら3つのモジュール間の相互作用に関連するという異なる役割分担があると考えたためである。リカレントニューラルネットワークはこれら2種類の染色体の組み合わせで定義する。

5.4.2 適応度の計算

ニューロンの状態方程式（離散時間モデル）は以下の式で表わせる。

$$S_i(t+1) = S_i(t) + \sum_{j=1}^M w_{ij} Z_j(t) \quad (5.2a)$$

$$Z_j(t) = G \left(\sum_{i=1}^N w_{ji} S_i(t) + B_j \right) \quad (5.2b)$$

ここで、 $S_i(t)$ はi番目のダイナミックユニットの時刻tでの出力を示し、 $Z_j(t)$ はj番目のシグモイドユニットの時刻tでの出力を示す。 w_{ij} はj番目のユニットからi番目のユニットへの結合荷重値、 $G(\cdot)$ はシグモイド関数、 B_j はj番目のユニットでのバイアスを示す。

エラー評価関数Eは、外部出力ユニットiに対して、出力 $S_i(t)$ と教師データ $T_i(t)$ の平均2乗誤差を時刻TからT+TBまでの間累積した値とした。

$$E = \sum_{t=T}^{T+TB} \sum_{i \in \text{output}} (S_i(t) - T_i(t))^2 / 2 \quad (5.3)$$

遺伝的操作を行う場合の適応度fはエラー評価関数の逆数で定義する。

$$f = 1 / E \quad (5.4)$$

5.4.3 遺伝的操作

各世代、確率1/2でIA, IBのうちどちらか一方の染色体Iを選び、遺伝的操作を施した。

(1) 淘汰および増殖

図5.5は、淘汰および増殖の方法を示す。現世代のN個の個体 $I_1 \sim I_N$ のうち適応度が下位のN/2個の個体を淘汰する。N個の個体の中からルーレット選択によりN/2個の親を選択し、交差と突然変異を施して補充するN/2個の個体を生成する。

(2) 交差

ニューラルネットワーク上の2次元部分構造単位にニューロン間の結線情報と結合荷重値を置き換える2次元交差を行う。交差の単位となる2次元部分構造には、ランダムに選んだあるニューロンを頂点として入力層全体を底辺とするピラミッド構造を用いる。一例を図5.6(a)および図5.6(b)に示す。ここで示す交差はGenetic Programming (GP)[Koza 92]の交差と似ているが、標準的なGPの交差とは以下の点で異なる。(i) GPで扱うノードは一般的にリンクとの間に結合強度という概念を持たない。したがって、結合荷重値の学習もない。(ii) GPの交差ではノードの移動を伴うが、2次元交差では、ノード間の結線情報と結合荷重値の交換を行い、ノード自体は固定である。(iii) GPの交差では実際にノード間に結合の存在する木構造単位の置き換えを行うが、2次元交差ではピラミッド構造の頂点となるニューロンに対して、直接または間接的に、入力源となる可能性のあるノード（ニューロン）は全て交差の単位となるピラミッド構造内に含める。したがって、ニューラルネットの入力層全体が必ずピラミッド構造の底辺となる。これらの2次元交差の特徴は、ニューラルネットワーク上の2次元スキーマを交差で破壊する確率を低下させることを目的としている[Sato 95a]。

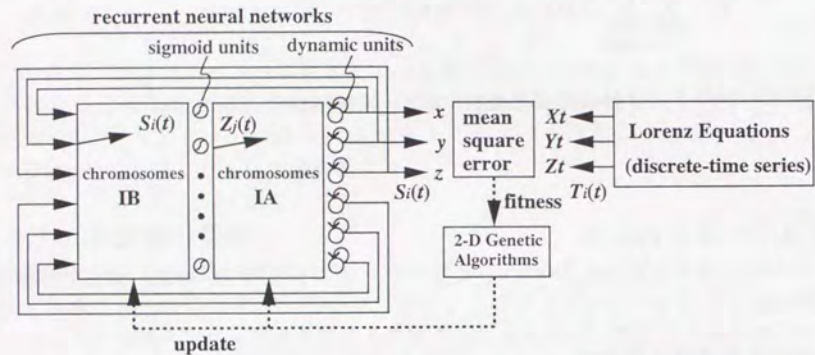


図5.4 学習に用いたリカレントニューラルネットワークの初期構造
51ニューロン（ダイナミックユニット：6，シグモイドユニット：45）構成として，
結線情報および結合荷重値の初期値はランダムに決定

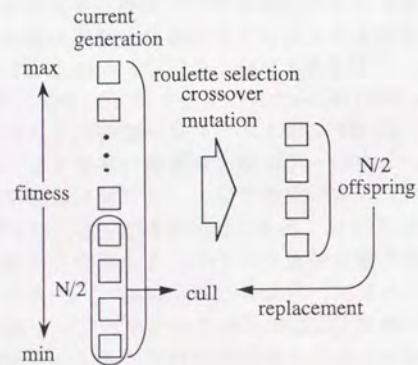


図5.5 選択と淘汰の方法

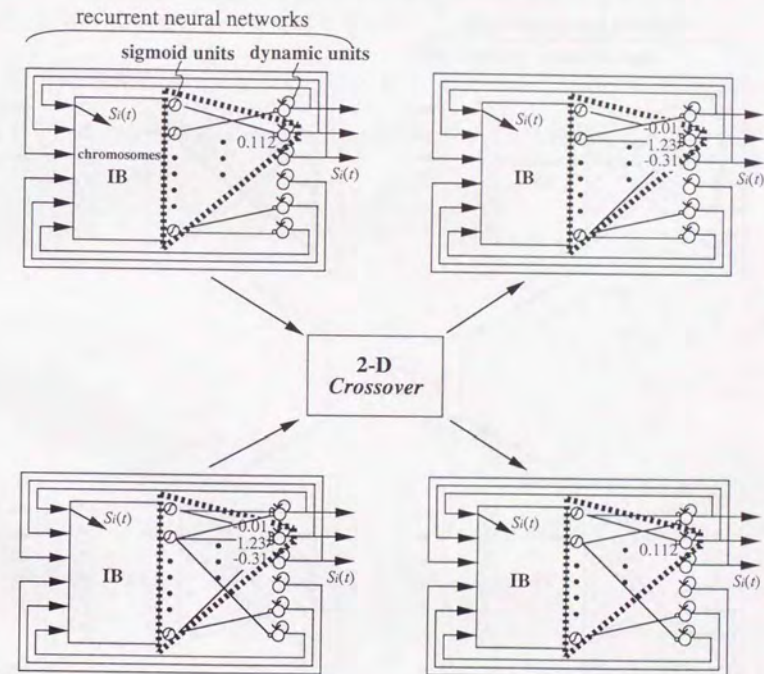


図5.6 (a) ダイナミックユニットに対する交差の例

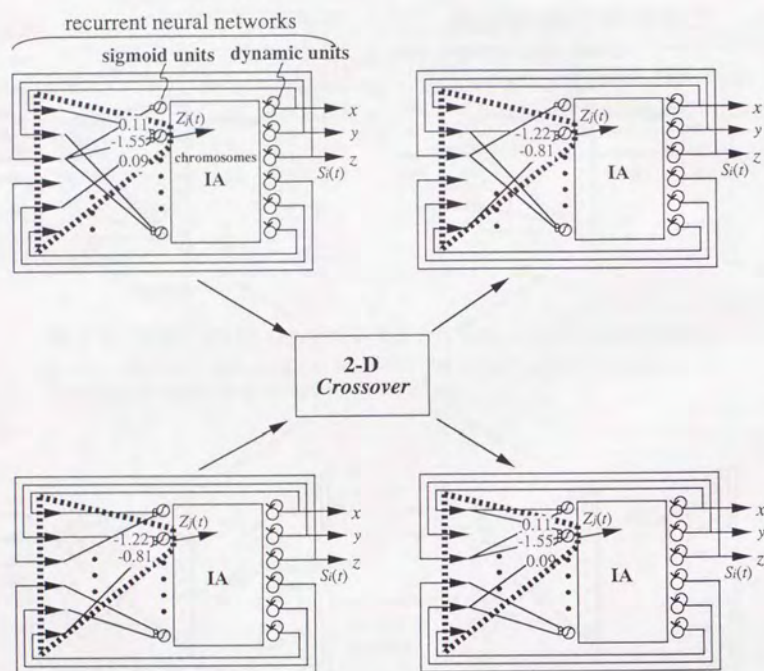


図5.6 (b) シグモイドユニットに対する交差の例

(3) 突然変異

突然変異に関する一例を，図5.6(c)および図5.6(d)に示す．結線情報に関する突然変異は，ニューロン間の結線情報の付加もしくは削除に対応する．結合荷重値に対する突然変異は，現在の結合荷重値 w_{ij} からの微小な変異を考える．ここでは，以下の式で表わされる変異を用いた．

$$w_{ij} = w_{ij} + \alpha w_{ij}, \quad (-0.1 \leq \alpha \leq 0.1) \quad (5.5)$$

ただし， w_{ij} は64ビット浮動小数点， α は乱数によって決定する16桁の実数を仮定した．

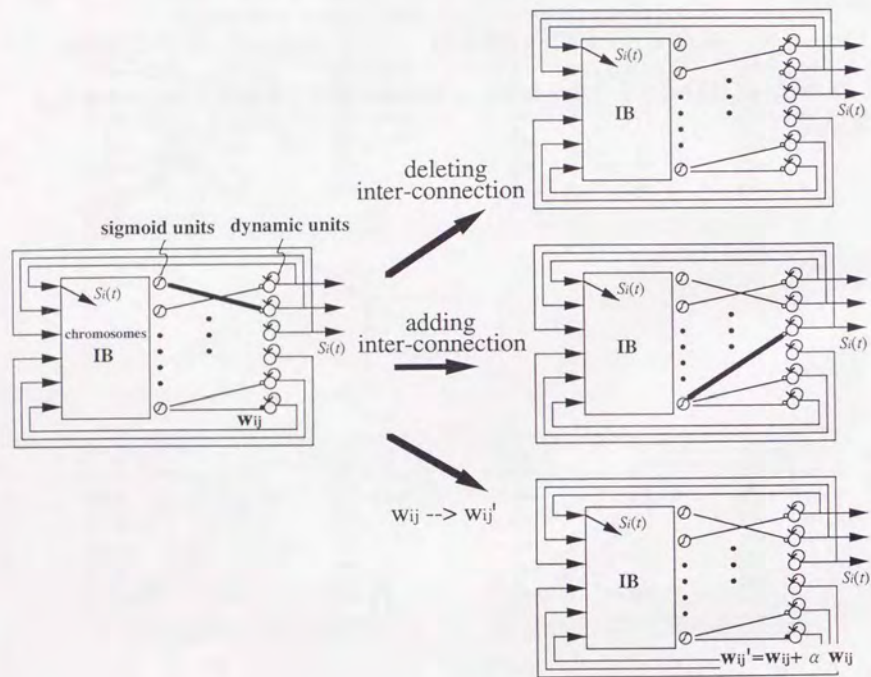


図5.6 (c) ダイナミックユニットに対する突然変異の例

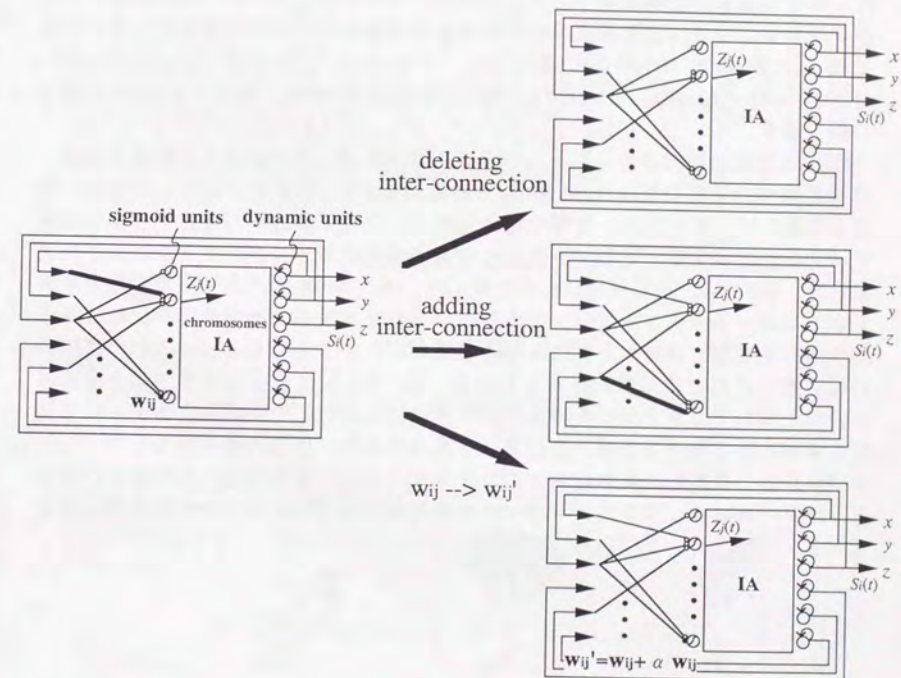


図5.6 (d) シグモイドユニットに対する突然変異の例

5.5 計算機実験

個体総数 $N=200$, 交差率 $\mu_c=1$, 結線に関する突然変異率 $\mu_{m1}=0.002$, 結合荷重値に関する突然変異率 $\mu_{m2}=0.2$ とした. 結線情報の初期値は層間全結合を仮定した. 結合荷重値の初期値はランダムに選んだ小さな値を用いた. 教師信号の生成および学習の計算は, 倍精度浮動小数点演算を基本とした. 初期条件をランダムに変えた実験を100回繰り返した. その中から初期値 $x(0)=-4.360352$, $y(0)=-4.402954$, $z(0)=22.149292$, エラー評価区間 $TB=8$, シフト量 $TF=4$ の場合の例を示す.

図5.7は学習途中のリカレントニューラルネットワークが出力した軌道を示す. 学習を途中で中断した時から2000回分の出力を2次元平面上に表示している. 学習を開始して, 比較的早い段階で安定平衡点に収束する. 学習が進むとリミットサイクルが出現する. そして, さらに学習が進むとローレンツアトラクタが観測された. 図5.8に, 同じ初期値から出発した, 200,000回のウエイト変更後のリカレントニューラルネットワークの出力とローレンツ軌道の時間波形の比較(時系列予測)を示す. 時間とともに2つの軌道のずれが大きくなっているが, 短期的には定性的に似た振る舞いを示している. 図5.9は学習曲線を示す. 結合荷重値の変化に対してネットワークの出力が不連続的に変化して一時的にエラー評価関数の値が大きく増大するが, 長期的には次第にエラー関数の値が減少していることが分かる. これらの実験結果からカオスダイナミックスの隠れた特徴を学習するリカレントニューラルネットワークが進化的アルゴリズムにより合成できたと考える.

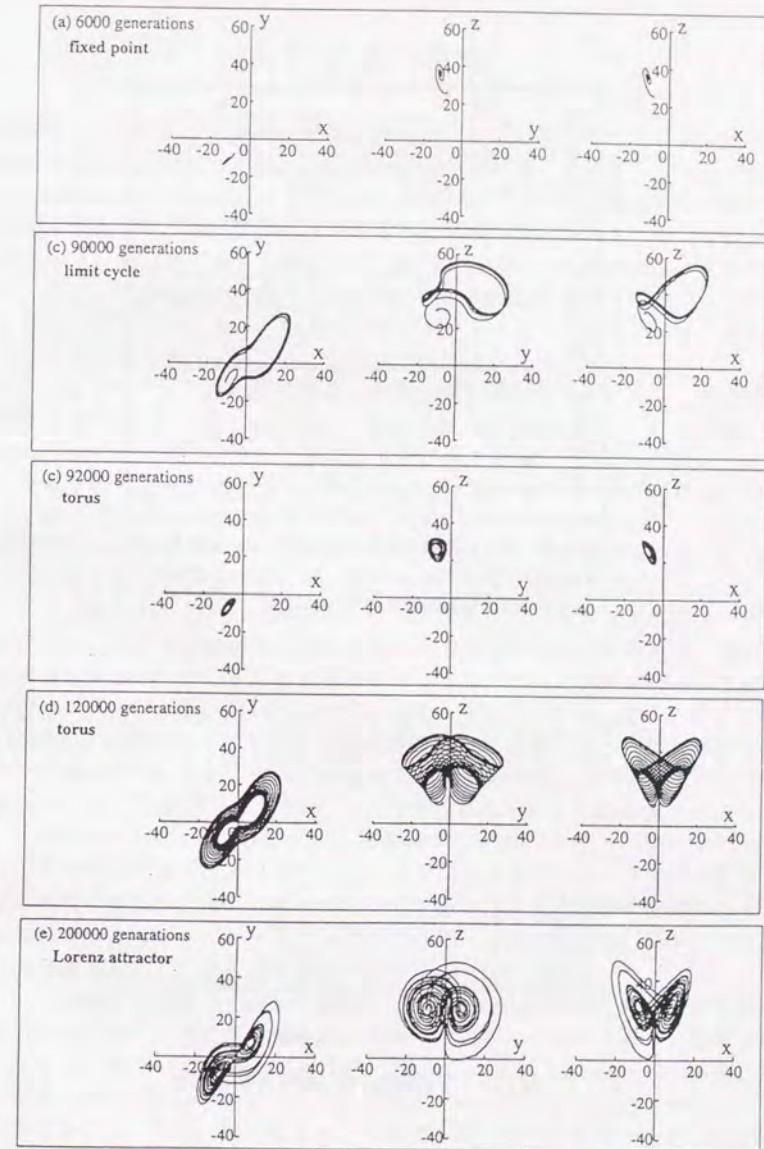


図5.7 学習途中のリカレントニューラルネットワークが出力した軌道

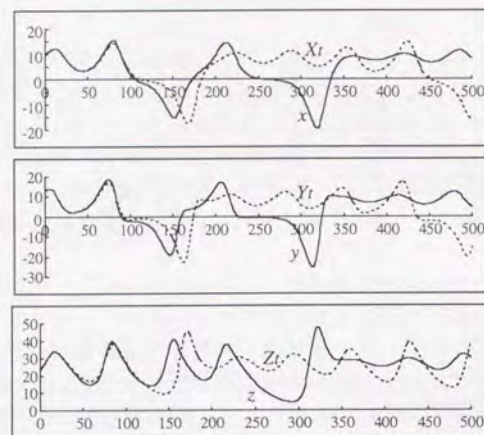


図5.8 同じ初期値から出発したローレンツ軌道と学習過程のリカレントニューラルネットワークの出力 (200,000回学習後の時系列予測の比較)

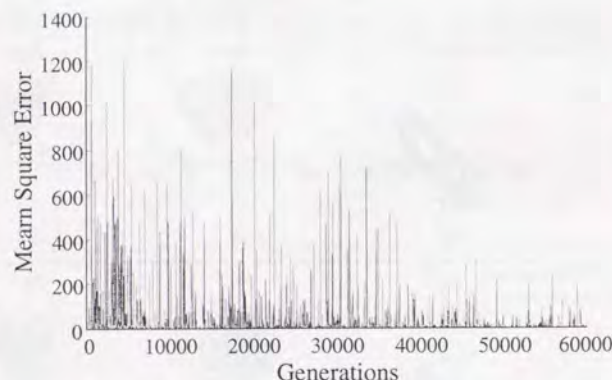


図5.9 学習曲線の例 (最良個体の場合)

5.6 考察

進化的アルゴリズムを用いた時系列データの学習に関しては既に幾つか報告されている[Saravanan 93, Iba 93, Angeline 94, Kato 95]. しかし、いずれも周期性を持つ時系列データを対象としている. あるいは、カオスの軌道の一部分を学習区間として固定し、周期軌道として繰り返し学習を行っている. ここでは、無限に続く非周期時系列データを対象とし、学習を打ち切った後のニューラルネットワークの出力に着目した評価 (すなわち、時系列予測) を行っており、今までにない新しい試みと考える. また、カオスダイナミックスを対象として、進化的アルゴリズムの学習能力を探る上で貴重な実験結果を示せと考える.

一方、ネットワークの結線情報を過去の経験に基づき明示的に与え、結合荷重値の学習にBPTTアルゴリズムを用いた場合[Sato 90, Sato 95f]との比較を考える. 今回の実験結果に関する限り、進化的アルゴリズムによりニューラルネットワークの合成を行った場合には、リカレントニューラルネットワークの構造を明示的に与え結合荷重値の学習にBPTTアルゴリズムを用いた場合よりも、アトラクタ出現までに10倍程度の学習回数を必要とした. また、初期条件を変更した場合の再現性に関しても現段階では十分とは言えず、遺伝的操作に必要なパラメータ調整が困難であった. 図5.7も最良のデータである. これは、カオスダイナミックスに代表されるように動的かつ長期予測不能な問題を対象とした場合、遺伝的操作の設定方法やパラメータ調整がニューラルネットワークの学習における学習率などのパラメータ調整に比べて必ずしも単純でないことを示していると思われる.

一般的に、進化的アルゴリズムは最適解の周辺には早く近づくが局所探索能力が弱いと言われている. これは、進化的アルゴリズムでニューロン間の結合荷重値を決めていく探索過程が基本的にrandom searchと考えられるためである. 一方、BPTTでの学習は最急降下法による探索の方向性が学習データにより与えられる. すなわち、進化的アルゴリズムによる探索は離散空間で局所解が数多く存在する問題に威力を発揮し、今回例題として用いた連続実数空間の探索はもともと進化的アルゴリズムには不向きな問題であり、BPTTと比較して学習効率が悪いのは自明と考えることもできる.

一方、進化的アルゴリズムは多点探索を行うという特徴を有する. すなわち、探索の過程において複数の解集団を保持し、これらの間で共有される解空間の中の超平面に基づいて処理が進行する. 従って、単純なrandom searchと比較して並列化の効果が期待できるとともに学習が比較的初期値に左右されにくいという特徴を有する. また、交差により2つの個体間で染色体を組み合わせる新しい個体を生成するため、両親の優れた部分形質をうまく組み合わせることで継承すること

に成功すれば、探索における飛躍が期待できる。ここでは、これら進化的アルゴリズムの特徴を積極的に活かすことにより、学習効率がBPTTなどのアルゴリズムに近づくことを期待している。局所探索能力の問題が改善されれば、ニューラルネットワーク構造の自動的獲得が可能である、初期値に比較的左右されない、適応度を利用するだけで他の（勾配などの）情報を使わないため目的関数の性質が良く分からない問題にも適用できる、などの進化的アルゴリズムの利点を連続実数空間の探索問題にも活かせると考えるからである。そこで今回、次に示す工夫のもとに実験を行った。(a)一般的に結合荷重値に関する突然変異は全探索空間内でのrandomな変異を仮定するが、ここでは式(5)に示すように現在(親)の結合荷重値の近傍内に変異先を限定することにより、局所探索能力の強化を図った。(b)今回の実験では、SUN Sparc10を用いて、1回の実験に4~6日のTATを必要とする時間のかかる学習であったが、個体総数200という比較的大きな集団サイズを用いて並列化の効果を狙った。(c)一般的には1次元染色体上でのrandomな位置での交差を行うが、ここでは2次元部分構造に着目した交差を行うことにより、ニューラルネットワーク上の2次元スキーマを交差で破壊する確率を低下させる[Sato 95a]ことを試みた。これらの工夫を施して実験を行ったが、結果的には、BPTTに匹敵する学習効率は得られなかった。但し、予めネットワークの構造を明示的に与えて結合荷重値の学習のみにBPTTを適用する場合と比較して、ネットワーク構造も同時に進化的計算で獲得することを狙っているために、問題の設定が非常に難しくなっている。また、BPTTの場合と異なりパラメータ調整に関する過去のノウハウがないため、パラメータの調整により学習効率が大きく改善される可能性を残している。さらに、カオスダイナミックスの隠れた特徴を学習するリカレントニューラルネットワークを獲得できたことから上記工夫がある程度学習効率の向上に寄与したと考えることができる。今後の対策案の一つとしては、進化的アルゴリズムによる広域探索とBPTTによる局所探索のハイブリッド化が考えられる。すなわち、初期段階を進化的アルゴリズムで探索し途中からBPTTなどのアルゴリズムに切り換えることが考えられる。但し、フィードフォワード型ニューラルネットワークで進化的アルゴリズムとBPなどの学習アルゴリズムのハイブリッド化[Kitano 90]を考えると異なり、分岐などの現象を考えると、進化的アルゴリズムからBPTTに切り換えるタイミングの選定が困難である。例えば、図5.9に示す学習曲線を見ても分かるように、誤差関数の値がある一定値以下になったらBPTTに切り換えるという単純な方式では対応できない。進化的アルゴリズムをどのようにリカレントニューラルネットワークの学習に取り込むかの検討が引き続き必要と思われる。

一方、今回の実験では、ローレンツアトラクタやリミットサイクルなど特徴的な出力を示す学習過程のネットワーク構造を調べてみると、必ず出力 x, y, z に対応したモジュール化の傾向が観測された。このことは、問題の特徴を考慮したモジュール化(すなわち、疎結合)構造が学習において重要な意味を持つことを示唆しているとも考えられる。ローレンツ軌道では基になる生成規則(すなわち、ローレンツ方程式)が予め分っている。したがって、リカレントニューラルネットワークの構造をある程度最初から明示的に決定することも可能である。しかし、実世界における情報には、カオス的なふるまいを示す信号であることは分かっている。基になる生成規則が不明な場合も多い。例えば、人間の声帯振動波形にはカオス的なゆらぎが含まれているとの報告[Sato 90]がある。しかし、基になる生成規則は不明である。自然に聞こえる音声合成を考える場合、進化的アルゴリズムを用いた学習が有効になる可能性があると考えられる。

5.7 まとめ

本論文では、カオスダイナミックスを学習するリカレントニューラルネットワークの合成に進化的アルゴリズムを適用することを試みた。教師信号として用いたローレンツ軌道は、常にカオス的に変化し続ける非周期軌道であり、進化的アルゴリズムの学習能力を探る上で新しい試みと考える。結果として、進化的アルゴリズムにより合成した学習過程のリカレントニューラルネットワークの出力が、不動点、リミットサイクル、ローレンツアトラクタと変化することを計算機実験により示した。すなわち、進化的アルゴリズムにより合成したニューラルネットワークがカオスの隠れた秩序を学習できることを示した。一方、ネットワーク構造を明示的に与えBPTTアルゴリズムにより結合荷重値の学習を行った場合と比較すると、アトラクタ出現までに多くの学習回数を必要とした。また、再現性も現段階では不十分である。遺伝的操作の適用方法およびパラメータの設定などに関して引き続き検討が必要である。また、モジュール化と学習効率の関係に関しても引き続き調査が必要と考える。

5.8 付録：リカレントニューラルネットワーク 高速シミュレータの開発

5.8.1 はじめに

我々は、複雑かつ動的に変化する環境に対して自律的に適応・進化するシステムの実現を目的として、複数のリカレントニューラルネットワークが相互作用を持ちながら学習を行うモデルの研究を行っている[Sato 94b, Sato 95b, Sato 95c]. なぜなら、リカレントニューラルネットワークは周期的な軌道の学習からカオスダイナミックスの学習[Sato 90]まで幅広い適応能力を持つため、複数のリカレントニューラルネットワークの相互作用を考えることにより、今迄にない動的情報処理を実現する可能性があると考えからである。あるいは、動的で複雑な問題に対応するには大規模なリカレントニューラルネットワークが有効になる可能性が高いと考え、問題に適した特別な構造を持たない完全結合型リカレントニューラルネットワークよりも、問題の特性を考慮した疎結合部分が学習効率の向上に貢献すると考えるからである。

一方、リカレントニューラルネットワークのような非線形力学系では、パラメータのわずかな変化に対して、システムの振舞いが不連続に変化する分岐という現象が起こることが知られている[Guckenheimer 83]. 即ち、ネットワークの構造決定および学習率などのパラメータ調整は難しい問題であり、学習にも多くの時間を必要とする。

従って、リカレントニューラルネットワークを用いた学習では、小規模ネットワークで完全結合を仮定した実験が一般的である[Sato 93, Takagi 94]. 我々が研究対象とするモデルでは、複数のリカレントニューラルネットワークの相互作用を扱うために、システム全体が大規模になる場合が多い。また、どのようにモジュール分割するか(すなわち、どのように疎結合部分を作るか)を決める必要もあり、上記問題はさらに深刻である。設計者の経験と試行錯誤によりネットワークの構造決定とパラメータの調整を行っている現状では、1回の学習に必要な時間を短縮することにより、学習結果の迅速なフィードバックが必要となる。

以上のような背景から我々は、複数のリカレントニューラルネットワークが相互作用を持ちながら学習を行うモデルを高速に評価するためのシミュレータをニューロコンピュータ[Sato 93a]上に実装することを試みた。以下、5.8.2節では、本シミュレータのシステム概要を示す。5.8.3節では、学習アルゴリズムを示す。5.8.4節において、ローレンツアトラクタなどの問題を用いて機能検証を行った結果および性能評価結果[Sato 95f]を示し、最後に考察とまとめを行う。

5.8.2 システム概要

5.8.2.1 シミュレータシステムの入出力

本シミュレータはユーザインタフェースとなるワークステーション (Workstation, WS) およびカレントニューラルネットワーク高速学習用ニューロコンピュータから構成される。ニューラルネットワークの構造、学習パラメータ、教師データ、外部入力データをWS上で指定する。これらの情報はSCSIバスを介してニューロコンピュータ上に転送される。ニューロコンピュータ上では、最大512物理ニューロンが並列学習を行う。学習曲線、ニューラルネットワークの出力値、各ニューロンの状態値などがWSを介して得られる。ネットワークの構造としては、全結合型、階層型、出力層から入力層へのフィードバック付き階層型の他、これら幾つかのニューラルネットワークを任意に接続したネットワークを指定可能である。

5.8.2.2 ニューロコンピュータの構成

図5.10にニューロコンピュータ内の構成概略を示す。ニューロコンピュータは1枚の制御ボード (control board) および2枚のニューロンボード (neuron board) から構成される。制御ボードは制御記憶 (control storage)、制御チップ (control chip)、データバッファ (data buffer) などから構成され、ニューロンボードは32個のニューロンチップから構成される。SCSIを介してWSからニューロコンピュータに転送されたデータの内、学習パラメータ、教師データ (Training Data)、外部入力データは制御ボード上のデータバッファに格納される。ニューラルネットワーク構造は制御ボード上の制御記憶に格納される。学習アルゴリズムは、マイクロプログラムに変換後、制御記憶に格納される。

本ニューロコンピュータ上には512個の物理ニューロンが実装されており、理論上の最大性能は、Back Propagation (BP) 学習アルゴリズムを用いた場合、学習時: 1.26 GCUPS (Giga Connections Updates Per Second)、非学習時: 12.8 GCPS (Giga Connections Per Second)である。仮想化機能 (複数の仮想ニューロンが、時分割で1個の物理ニューロンを共有して演算を行う) により最大4096ニューロンまで対応可能であるが、今回開発したシミュレータでは開発工数の関係から仮想化のための変換機能をサポートしていない。また、物理ニューロンの拡張もサポートしていない。したがって、512ニューロンを越えるネットワークサイズには現状対応できないという制限を有する。但し、最大512ニューロンという制限を満たすかぎり、必要なメモリは確保されており、メモリスワップなどにより性能が低下する心配はない。一方、シミュレーション対象となるニューラルネットワークを構成する全ニューロン数 N が512個より少ない場合、 $(512 - N)$ 個の余剰ニューロンはNOP (No Operation) 状態となる。

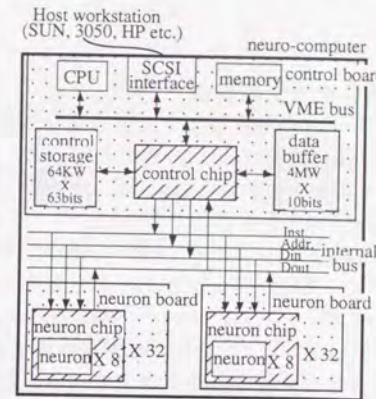


図5.10 ニューロコンピュータの構成概略

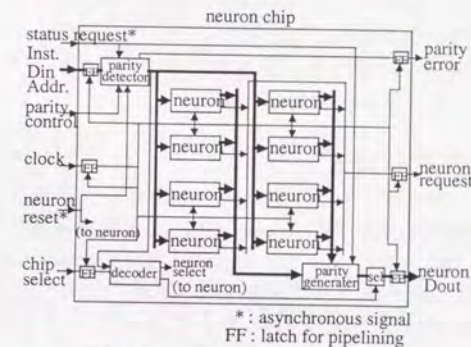


図5.11 ニューロンチップの構成

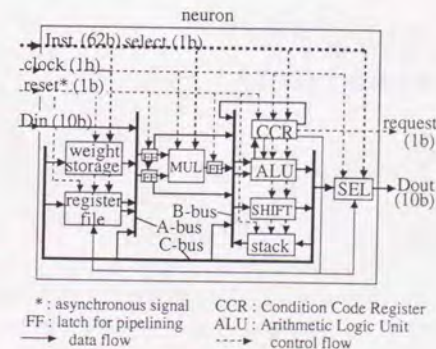


図5.12 ニューロンのブロック図

5.8.2.3 ニューロンチップの構成

図5.11にニューロンチップの構成を示す。各ニューロンチップには8個の物理ニューロンが1.0- μm CMOS技術により搭載されている。従って、1ボード当り256物理ニューロン、システム全体では512物理ニューロンが搭載されている。

図5.12にニューロンのブロック図を示す。各ニューロンは乗算器(MUL)、算術論理演算ユニット(ALU)、シフタ(SHIFT)などの演算器群および結合荷重値記憶用ローカルメモリ、汎用レジスタ群などから構成される。演算器群では、倍精度機能を用いて、乗算: 32bits x 20bits, 和算: 64bits + 64bitsの精度で演算を行う。結合荷重値は32bits, ニューロンの出力値の精度は20bitsとした。ニューロコンピュータ上の演算は全て固定小数点演算である。汎用レジスタ群は学習率などのパラメータ情報の一時記憶や仮想ニューロン用の結合荷重値の記憶などに利用される。ニューロン制御用の信号は、各クロックサイクルごとに制御記憶から読み出されたマイクロプログラムが、制御チップ内でデコードされて生成される。この制御信号に従い最大512物理ニューロンが並列動作を行う。積和演算は、パイプライン処理を行うことにより見かけ上、1クロックサイクルで実行可能である。クロックサイクルは25MHzである。

5.8.3 学習アルゴリズム

リカレントニューラルネットワークに対する教師付き学習法は複数提案されている[Takagi 94, Williams 89, Pearlmutter 89]。ここでは高速化を最重要視し、計算量がニューロン数の2乗のオーダーと少ないBack Propagation Through Time (BPTT)[Pearlmutter 89]アルゴリズムを基本とした。但し、学習アルゴリズムに対応したマイクロプログラムを作成することにより、Real-Time Recurrent Learning (RTLTL)[Williams 89, Doya 92]などの他の学習アルゴリズムに対応することも可能である。

5.8.3.1 ニューロンの動作

以下に示す離散時間の動作方程式を用いた。

$$S_i(t+1) = \sum_{j=1}^n w_{ij} y_j(t) + \sum_{j=1}^m b_{ij} I_j(t), \quad (5.6)$$

$$y_i(t) = f(S_i(t))$$

ここで、 $S_i(t)$ および $y_i(t)$ ($i = 1, \dots, n$)はそれぞれ i 番目のユニットの時刻 t での内部状態と出力を示し、 w_{ij} は j 番目のユニットから i 番目のユニットへの結合

荷重値を示す。 $f(\cdot)$ はシグモイド関数または線形関数を示す。 $I_j(t)$ ($j = 1, \dots, m$)はネットワークへの外部入力を示す。

5.8.3.2 エラー評価関数

出力ユニット i に対して、出力 y_i と教師データ T_i の2乗誤差を時刻 t_0 から t_1 の間累積した値をエラー評価関数 E と定義した。

$$E = \sum_{t=t_0}^{t_1} \sum_{i \in \text{output}} (y_i(t) - T_i(t))^2 / 2 \quad (5.7)$$

5.8.3.3 結合荷重値の更新

出力ユニットで得られた誤差をもとに、全てのユニットでの等価的な誤差を求め、最急降下法により結合荷重値 w_{ij} を更新する。 E の w_{ij} に対するこう配 $\partial E / \partial w_{ij}$ を求めるために、次の微分方程式を満たす随伴係数 q_i を導入する。

$$q_i(t-1) = \sum_{j=1}^n q_j(t) w_{ji} f'(S_i(t)) + \delta_i(t), \quad (5.8)$$

$$q_i(t_1) = 0 \quad (i = 1, \dots, n)$$

$$\delta_i(t) = (y_i(t) - T_i(t)) f'(S_i(t)), \quad (\text{output units}),$$

$$= 0, \quad (\text{hidden units}).$$

但し、 $f'_i(y_i)$ は $\partial f_i(y_i) / \partial y_i$ を表す。

E の w_{ij} に対するこう配 $\partial E / \partial w_{ij}$ および結合荷重値の変分 Δw_{ij} は、式(5.6), (5.8)より、以下の式で求められる。

$$\partial E / \partial w_{ij} = \sum_{t=t_0}^{t_1} q_i(t) f(S_j(t)) \quad (5.9)$$

$$\Delta w_{ij}(t+1) = -\eta \partial E / \partial w_{ij} + \alpha \Delta w_{ij}(t) \quad (5.10)$$

ここで、 η は学習率、 α は慣性項を示す。

5.8.3.4 強制力学習

本シミュレータシステムでは強制力学習 (teacher forcing) [Doya 92] と呼ばれる教師付き学習方法も可能である。強制力学習では、学習時に教師付きユニットの出力が教師信号に等しくなるように、外部から強制力を教師付きユニットに加える。加える強制力の強さが大きいほどリカレントニューラルネットワークのダイナミックスが学習対象のダイナミックスからずれていると考えられるので、エラー評価関数 E には外部から加える強制力を用いる。5.8.4節で示すローレンツアトラクタの学習には強制力学習を用いている。強制力学習は一般的に自律系の自励振動パターン学習に有効との報告 [Williams, Doya 89] があるためである。強制力学習の場合の動作方程式を以下に示す。

$$S_i(t+1) = \sum_{j=1}^n w_{ij} T_j(t) + \sum_{j=1}^m v_{ij} I_j(t) \quad (5.11)$$

$$y_i(t) = f(S_i(t))$$

5.8.4 計算機実験

ここでは、計算機実験を通して学習機能および性能を観察することにより、提案するリカレントニューラルネットワークのシミュレータがニューロコンピュータ上に正しく実装されていることを検証する。

尚、実験に用いたニューロコンピュータの理論上の最大性能は、BP学習アルゴリズムを用いた場合、学習時: 1.26 GCUPS, 非学習時: 12.8 GCUPSである。ニューロコンピュータ上の演算は全て固定小数点演算であり、計算精度は、ニューロン間の結合荷重値に関しては32ビット (整数部: 7ビット, 小数点以下: 25ビット), 教師信号は20ビット (整数部: 7ビット, 小数点以下: 13ビット), シグモイドユニットの出力は20ビット (整数部: 1ビット, 小数点以下: 19ビット), ダイナミックユニットの出力は20ビット (整数部: 7ビット, 小数点以下: 13ビット), 積和演算は64ビットの精度とした。これらの計算精度は試行錯誤的な実験を何度も繰り返した結果から決定した。

一方、性能比較評価に用いたWSは、SUN SS10は17.2MFLOPS (クロック速度40.3MHz), 3050RX/200は7MFLOPS (クロック速度33MHz) の性能のマシンである。WS上のプログラムは全てC言語で作成し、最適化オプション (最適化レベル=2, 関数のインライン展開は行わない) を指定したコンパイルを行った。WS上の計算精度はdouble型の浮動小数点 (64ビット) を仮定した。

5.8.4.1 機能検証

(1) 周期的軌道の学習

周期的な軌道を学習する機能の検証にはサイン関数を用いた。全てシグモイドユニットを用い、自己結合を含む全結合構成で実験を行った。ニューロン総数をパラメータとして幾つか実験を行った。いずれの場合も、かなり忠実に教師データを再現していた。50ニューロンの場合を例に、学習曲線および50000回のウェイト変更後のリカレントニューラルネットワークの出力をそれぞれ図5.13および図5.14に示す。但し、学習率 η を0.1, 慣性項 α を0.0, $t_0=0$, $t_1=50$ として実験を行った。誤差の値は0.01よりも小さく、見た目にも学習後のニューラルネットワークの出力は教師データと重なって見える。周期的な軌道を学習する機能に関しては正しく実装されたと考えられる。

(2) カオスダイナミックスの学習

カオスダイナミックスを学習する機能の検証にはローレンツアトラクタを用いた。これは外部入力のない自律系で、運動方程式は次式で与えられる。

$$\frac{dx}{dt} = F_1(x, y, z) = 10(x - y) \quad (5.12a)$$

$$\frac{dy}{dt} = F_2(x, y, z) = -y + (28 - z)x \quad (5.12b)$$

$$\frac{dz}{dt} = F_3(x, y, z) = -\frac{8}{3}z + xy \quad (5.12c)$$

ローレンツ方程式の3つの状態変数に対応して3つのダイナミックユニットを用いた。また、45個のシグモイドユニットを用いた。ネットワーク構造は、ローレンツアトラクタの学習に有効な構造が現時点では明確でないために、全結合を仮定した。ローレンツ方程式を解くには、オイラー法を用い、時間ステップ幅は0.01とした。ローレンツ方程式の解軌道は非周期軌道であるため、学習法はBPTTのエラー評価区間 TB を TF ($< TB$) ずつずらしながら学習を進める方法 [Murakami 90] を採用した。全てのダイナミックユニットに式 (5.12) で計算した $(x(t), y(t), z(t))$ の値を教師信号として与えた。結合荷重値の初期値はランダムに選んだ小さな値を用いた。幾つかパラメータを変えて実験を行い、ほぼ同一の定性的な傾向を得た。その中から初期値 $x(0)=-4.360352$, $y(0)=-4.402954$, $z(0)=22.149292$, 学習率 η を $1/2048$, 慣性項 α を0.5, $TB=10$, $TF=5$ の場合の例を示す。

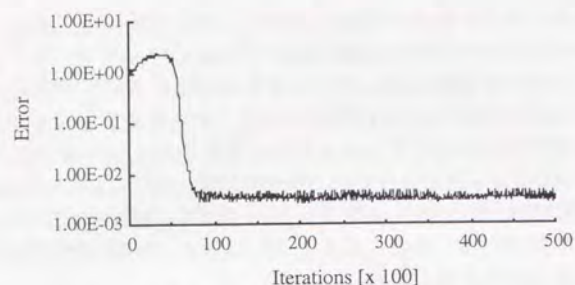


図5.13 サイン関数の学習曲線

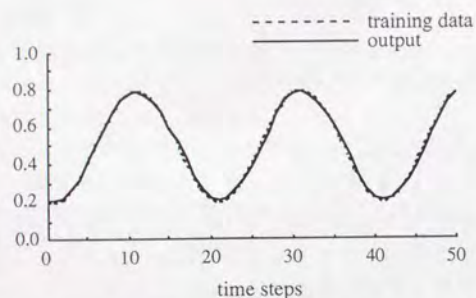


図5.14 学習(50,000回)後のリカレントニューラルネットの出力と教師データ

図5.15は学習途中のリカレントニューラルネットワークが出力した軌道を示す。図5.15(a)に示すようにリカレントニューラルネットワークの出力は学習を開始して比較的早い段階で安定平衡点に収束する。学習が進むと、図5.15(b)に示すように、リミットサイクルが出現する。さらに学習が進むと、図5.15(c)に示すように、このリミットサイクルが不安定になる。図5.15(d)は50000回のウェイト変更後のリカレントニューラルネットワークが出力したストレンジアトラクタを示す。このアトラクタのかたちは図5.16に示す教師のローレンツアトラクタに似ている。図5.17に、同じ初期値から出発した、50000回のウェイト変更後のリカレントニューラルネットワークの軌道とローレンツ軌道の時間波形の比較を示す。時間とともに2つの軌道のずれが大きくなっているが、長時間にわたって定性的に似た振る舞いを示している。図5.18は学習曲線を示す。ウェイトの変化に対してネットワークの出力が不連続的に変化して一時的にエラー評価関数の値が大きく増大するが、長期的には次第にエラー関数の値が減少している。これらの実験結果から、カオスダイナミックスを学習する機能に関しても正しく実装されたと考えられる。

(3) モジュール化機能の検証

ここでは、上記カオスダイナミックスの学習を通してモジュール化機能の検証を行った。すなわち、モジュール化記述を用いた場合の学習結果を上記全結合(すなわち、1モジュール)の場合と比較した。ローレンツ方程式の3つの状態変数に対応して3つのモジュールを定義した。各モジュールは1つのダイナミックユニットと15個のシグモイドユニットを全結合した。モジュール間の接続は、各モジュールのダイナミックユニットの出力が45個のシグモイドユニットに入力される構成とした。ネットワーク構造以外の条件は全て上記モジュール化を考えない場合と同一に設定し実験を行った。

上記と同一パラメータの場合に関して、学習途中のリカレントニューラルネットワークが出力した軌道を図5.19に示す。軌道の定性的な振る舞いは上記モジュール化を行わない場合と一致しており、モジュール化した場合にも学習機能に障害はなかったと考えられる。ここで興味深いのは図5.15(b)と図5.19(b)の比較である。13000回のウェイト変更後と比較すると、モジュール化を考えない全結合ではリミットサイクル状態であったのに対して、モジュール化記述を用いた場合には既にストレンジアトラクタへの移行が始まっている。この傾向は他のパラメータの組み合わせを用いた場合でも多く観測された。これは問題に適した特別な構造を持たない完全結合型リカレントニューラルネットワークよりも、問題の特性を考慮した疎結合部分が学習時間の短縮につながった結果と考えられる。但し、今回の実験結果のみで断定するには情報量が不十分である。今後、本シミュレー

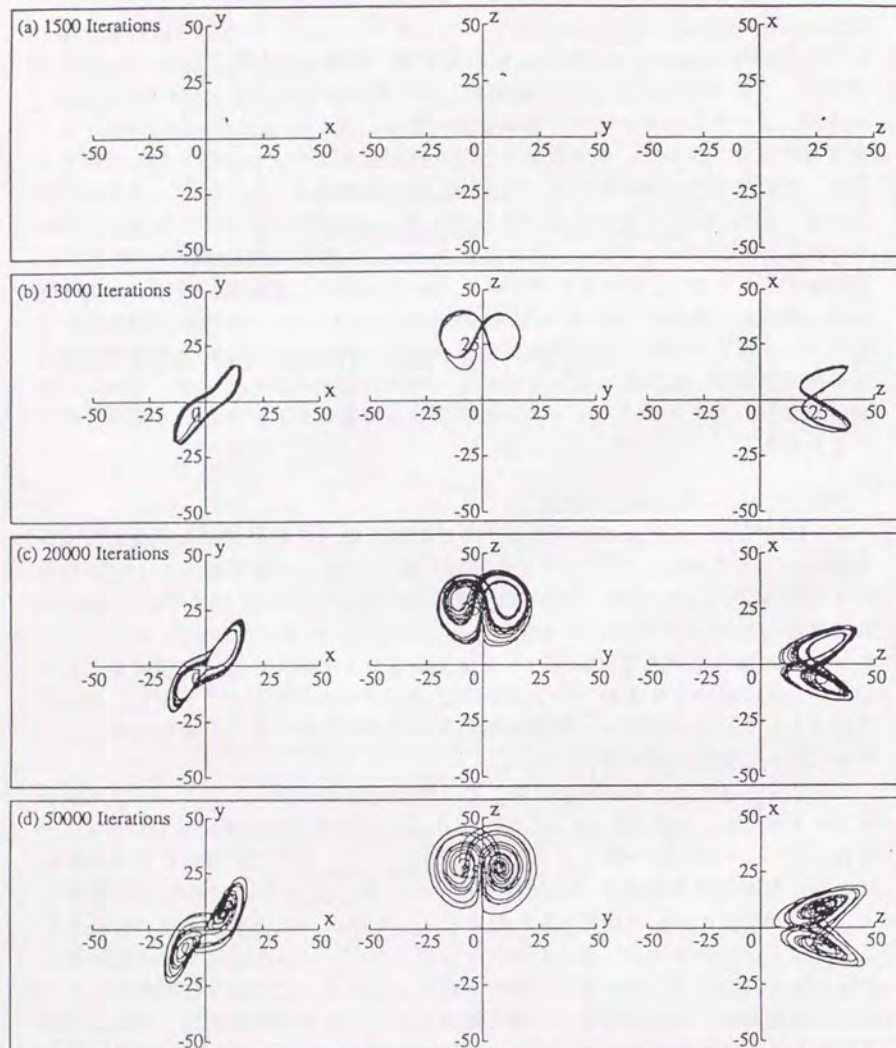


図5.15 学習途中のリカレントニューラルネットワークが出力した軌道
(モジュール化を行わない場合)

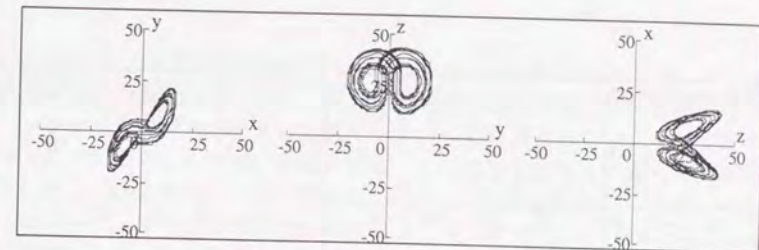


図5.16 教師のローレンツアトラクタ

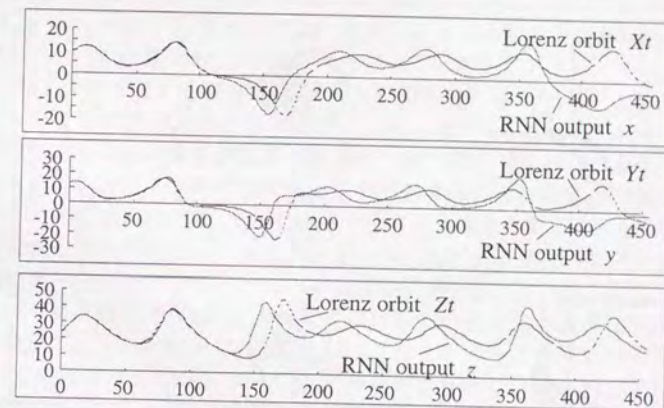


図5.17 同じ初期値から出発したローレンツ軌道とリカレントニューラルネットワークの出力 (50,000回学習後)

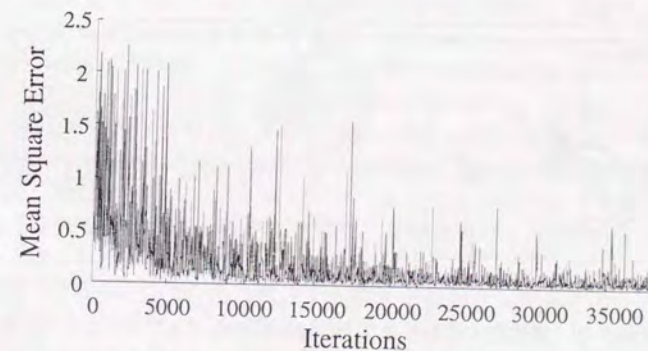


図5.18 ローレンツアトラクタの学習曲線

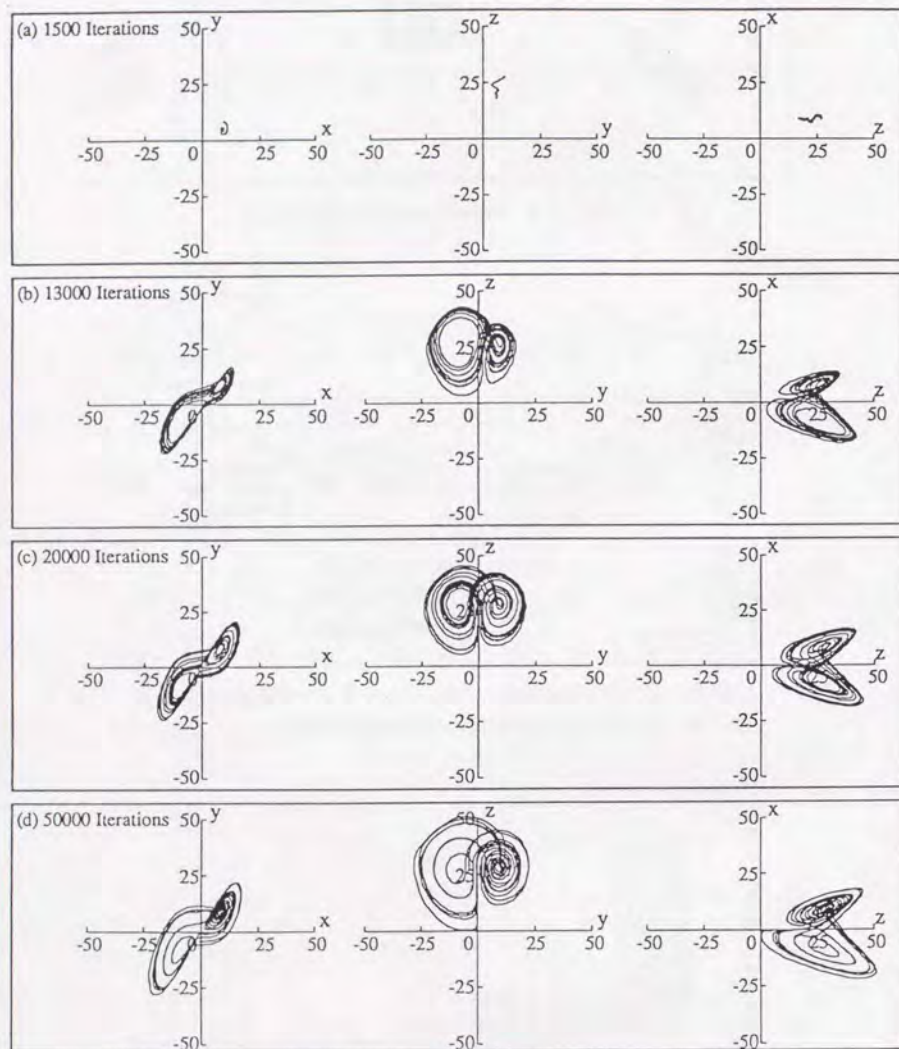


図 5.19 学習途中のリカレントニューラルネットワークが出力した軌道
(モジュール化記述を行った場合)

タを用いたモジュール記述の実験を通して検証していく必要がある。一方、モジュール化記述を用いた場合のアトラクタは、全結合の場合と比較して、教師のローレンツアトラクタとのずれが大きいと思われる。この原因としては、今回与えたモジュール構造が必ずしも問題に最適な構造になっていないことが考えられる。学習率やエラー評価区間の大きさに依存していることも考えられる。また、ニューロコンピュータ上での演算精度が必ずしも十分でない可能性がある。今後、4倍精度記述などを用いて演算精度を上げた場合との比較が必要と思われる。

5.8.4.2 性能評価

ここでは、ニューロコンピュータ上に実装したシミュレータとWS上に実装したシミュレータの性能比較を示す。ローレンツアトラクタの学習の場合、100000ステップの学習に必要な時間は、今回開発したニューロコンピュータ上のシミュレータでは約15分程度であったのに対して、WS上のシミュレータを用いた場合はおよそ半日の実行が必要であった。そこで、ニューロン数と実行時間の関係をローレンツアトラクタを例題として比較することは非効率的であると判断し、サイン関数の学習に関して幾つかネットワーク規模を変えて性能調査を行った。調査結果を図5.20に示す。図5.20において、MY-NEUPOWERが今回シミュレータを実装したニューロコンピュータを示す。WS版では、シミュレーション時間がニューロン数のほぼ2乗に比例して増加している。この結果は、荷重値行列の要素数がニューロン数の2乗に比例することを考えると自然である。一方、MY-NEUPOWERでは、シミュレーション時間がニューロン数に対してはほぼニアに増加している。これは、MY-NEUPOWERがSIMD (Single Instruction stream Multiple Data stream) 型の並列計算機に属するためである。すなわち、最大512ニューロンという使用上の制限を満たす限り、全てのニューロンは並列に演算可能であり、逐次処理でニューロン数の2乗に比例する演算は、ほぼニューロン数に比例した演算時間で実行可能である。例えば、64ニューロンの場合、MY-NEUPOWERでは208秒、SS10では1531秒、3050RXでは5376秒であるが、100ニューロンになると、MY-NEUPOWERでは315秒、SS10では3713秒、3050RXでは13612秒となりシミュレーション時間の差が著しく拡大する。今回開発したシミュレータは、WS上のシミュレータに対して、数10倍の性能向上が確認できる。複数のリカレントニューラルネットワークモジュールが相互作用を持ちながら学習を行う相関学習の場合には、相乗効果により数100倍の性能差が生じると予想される。

以上の実験結果から、周期的な軌道を学習する機能、カオスダイナミックスを学習する機能そしてモジュール化機能を有する高速なりカレントニューラルネットワークシミュレータがニューロコンピュータ上に実装出来たと考える。

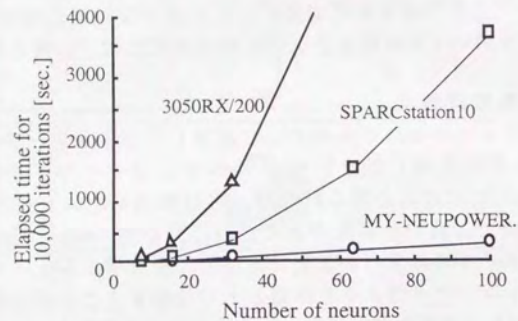


図5.20 ニューロコンピュータ版シミュレータとワークステーション実装の性能比較

5.8.5 まとめ

複数のリカレントニューラルネットワークが相互作用を持ちながら学習を行うモデルの評価を目的としたシミュレータを開発した。学習アルゴリズムにはBPTTを採用した。本シミュレータはニューロコンピュータ上で高速実行可能である。サイン関数の学習やローレンツアトラクタの学習などを用いて機能検証を行った。サイン関数の学習による評価から、ワークステーション上のシミュレータと比較し、数10～数100倍の高速化が予想される。今後、より複雑な問題を用いて、相関学習機能の詳細な検証が必要である。また、近年、進化的計算を用いたリカレントニューラルネットワーク合成の研究が始まっている。学習能力や精度などに関して、BPTTアルゴリズムを用いた場合と進化的計算を用いた場合の比較が必要と思われる。

第6章 進化的計算を用いた韻律係数フィッティング による声質変換に関して

要旨

第4章および第5章では意図的に設定した問題を用いて、進化的計算の動的問題への適応力を探る検討を行った。一方この章では、実世界の動的応用問題を用いて進化的計算の動的問題への適応力を探ることを試みる。特に、マルチメディア時代への移行が急速に進んでいることを考慮して、音声情報処理を実世界の問題の一例として取り上げることにする。すなわち、ピッチ（声の高低情報）、パワー、時間構造などの韻律情報を進化的計算により調整し、自然な声質の変換を試みる。

「明瞭な」、「子供っぽい」、などの感性表言語で示される幾つかの変換目的に対して声質の変換を行い、人手による試行錯誤で韻律情報を変換する係数の調整を行った場合よりも、多くの感性表現後に対して効率良く、かつ、自然に声質の変換ができる見通しを得た。

ここで示す進化的計算の応用技術は、例えば、ユーザ自身が自分の好みの声質に調整可能な音声ユーザインタフェースを持つマルチメディアPCなどに利用することができると考える。

6.1 はじめに

マルチメディア時代への突入により、Personal Computer (PC)を中心としたマルチメディア情報機器の市場が急速に伸びている。また、マルチメディアアプリケーションソフト市場も立ち上がりを見せ、ユーザが手軽に画像や音声を取り扱う環境が現実のものになってきた。特に、音声合成技術は電子メールやWorld Wide Webのテキスト情報、電子新聞等の読み上げから道路交通情報等のテキスト情報を音声出力できるナビゲーション装置など大きな市場が期待されている。しかし、規則合成方式など機械的に合成した音声は、音素片間の接続部の不連続感、信号処理を繰り返すことによる音質の劣化、音源・調音分離モデル（近似モデル）の限界などの問題が存在する。即ち、自然な音質の音声合成はきわめて困難であり、現状技術では機械的な音声や籠った音声になる傾向があり、音質の問題から音声合成製品の普及は遅れている。

一方、プレゼンテーションなどのマルチメディアコンテンツの編集に、音声加工技術を利用したナレーション付与を行う研究が始まっている[Kitahara 95]。プレゼンテーション等の編集においてナレーション付与は重要な作業の一つであるが、一般ユーザには表情豊かで明瞭な声質でナレーションを行うことは困難であり、声質の変換（加工）が必要と考えられるためである。例えば、素人が行うナレーションには、以下に示す問題点が存在する。

- (1) ナレーション原稿を棒読みする傾向が強い。
- (2) クリアな発声、雰囲気合わせた発声には訓練が必要。
- (3) 一定のピッチ、パワーを維持しながら原稿を読むことは困難。

しかし、明瞭な声質に変換するための音声加工技術に関しても、定性的なノウハウが得られているのみであり、具体的なパラメータ設定は試行錯誤的作業であり、調整が困難である。

以上のような背景から、本研究では自然な音質の音声を実現するために有効な技術を確立することを目的とする。本章ではその第一段階として、ユーザが発声した音声を、クリアで、韻律の強弱に富んだ声質に変換する技術に関して報告する。特に、声質変換のためのパラメータ調整に進化的計算を適用することを提案し、韻律係数のフィッティングに適用した幾つかの実験結果について報告する。ここで報告する技術はナレーションにおける音声加工技術としてだけでなく、規則合成システムと組み合わせることにより、テキストからの音声合成の明瞭性向上にも有効となる可能性があると考えられる。

6.2 音声の構成要素と声質変換

話者間の声質変換技術に関しては、既に幾つかの研究報告がなされている[Kuwabara 86, Takagi 86, Abe 89]。これらの研究報告では、話者間の変換は個人性パラメータが特定できないために、コードブック変換方式などが主流になっている。しかし、ベクトル量子化に起因する音質劣化などの問題が存在する[Abe 94]。ここでは、同一話者内での声質の変換を考える。まず、声質変換を行うために必要な特徴量の整理を行い、次に、韻律調整による声質変換について示す。

6.2.1 音声の構成要素と特性

人間の場合、喉の奥にある声帯の振動が音源となる。声帯は非常に弾性のある一種の筋肉に相当し、これが肺からの呼気によって振動を開始し、空気の振動（すなわち、音波）となる。この音源信号は一般に三角波あるいはのこぎり波に近い振動波形を持ち、振動の基本周波数の整数倍の倍成分から構成される。この一定間隔の倍音成分を広く持った音波（音源信号）が、声帯から唇までの声道を伝搬する間に、その形状によって共鳴が起こり、/a/や/i/の音といった音韻が作られ、最終的に唇から放射されて音声になる。すなわち、人間の音声は音源、声道の伝搬、放射の3要因で規定され[Kuwabara 86]、声帯で声の高さを、声道の形で音韻を制御している。また、音色をはじめとする声の質に関する性質を声質と定義すると、声質は声帯と声道の両方の状態から決定される[Sato 74, Ito 82, Kuwabara 86, Klatt 90]と考えることができる。すなわち、声質を制御する特徴量としては、ピッチ情報、パワー情報、時間構造、および、スペクトル情報を考えることができる。

6.2.2 韻律調整による声質変換

アナウンサーの聞きやすい音声の特徴を調べた研究[Kuwabara 83]などから、ナレーション向きの明瞭性の高い音声に関する定性的傾向はある程度明らかになっている。この特徴量の制御ルール[Kitahara 95]を表6.1に示す。また、感情音声の分析結果[Kitahara 89, Kitahara 92]から、典型的な感情を伴った音声に変換する制御パラメータも明かになっている。しかし、制御パラメータを具体的にどのような値に設定すべきかに関しては不明である。他の感性表現語を目的とした変換を扱う場合にも、制御パラメータの具体的な値に関しては不明である。また、一般的に、スペクトルの動的特徴をリアルタイムで制御することは困難である。すなわち、全ての制御パラメータの調整を初めから考えることは困難である。そこで、制御すべき情報を、ピッチ情報、パワー情報、時間構造の韻律情報に限定

することにより声質の変換を行うことを考える。

ピッチの変更方法を図6.1に示す。ピッチの上昇は1ピッチ内の波形の部分カットを行う。ピッチの下降は1ピッチ波形に無音部を挿入することで対応する。次に、継続長の変換方法を図6.2に示す。継続長の変換はTDHSの改良方式[Morita 86]によりピッチの変更を行うことなく時間長の伸縮を行う。パワーは対数パワー上での変換を行う。現在の値を W_i 、変換係数を k として、次式(6.1)に従い変換する。

$$\log_{10} W_{i+1}^2 = \log_{10} W_i^2 + k \quad (6.1)$$

表 6.1 明瞭性向上のための音声制御ルール

特徴			制御
スペクトル情報	静的特徴	平均スペクトル包絡	中域 (3~4KHz) のレベルを上昇
	動的特徴	ホルマントボタン	時間軌跡の振れ幅を拡大 各ホルマントのダイナミックレンジ拡大 第3ホルマントの最大値を上昇
韻律情報	静的特徴	平均ピッチ	全体的にやや上昇
	動的特徴	ピッチボタン	時間変動のダイナミックレンジを拡大
	パワー情報		子音部をやや上昇 無声子音のバズバーを除去
	時間構造		先行文に合わせたポーズ長

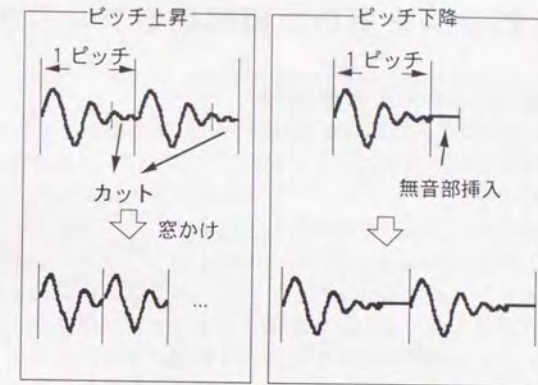
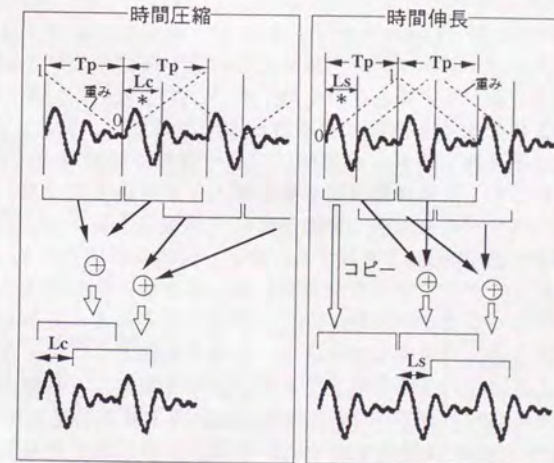


図 6.1 時間長を変更せずにピッチのみを制御する方法



Tp : ピッチ周期 R : 圧縮・伸長率 $Lc = R Tp / (1 - R)$ $Ls = Tp / (R - 1)$
* 圧縮/伸長率に応じてシフト

図 6.2 ピッチを変更せずに時間長のみを制御する方法

6.3 進化的計算を用いた韻律係数フィッティング

6.3.1 声質変換システム構成概略

図6.3に声質変換システムの概略構成を示す。声質変換システムは音声加工部と韻律変換係数学習部から構成される。音声加工部では、韻律情報の変換を行うことにより、“明瞭な”、“かわいい”などの感性表現語を目的とした声質の変換を行う。韻律情報の変換は韻律変換係数を用いて行う。音質の劣化を防ぐために、一般的な分析合成系[Nakagawa 91]のようにパラメータレベルまで変換せず、上述したように波形レベルで処理を行っている。本システムでは変換係数学習系を用いて、感性表現語などの定性的な変換目的を与えて、韻律を変換する際のパラメータを学習により自動的に獲得することの特徴とする。ここでは、学習アルゴリズムとして進化的計算を採用している。大規模組み合わせ最適化問題に有効な手法として一般的に知られているためである。

6.3.2 進化的計算の適用法

図6.4は韻律変換係数学習アルゴリズムとして遺伝的アルゴリズム[Goldberg 89]を用いた場合のフローチャートを示している。遺伝的アルゴリズムは生物の進化にならったアルゴリズムであり、適応範囲の広い多点探索アルゴリズムとして注目され始めている。まず染色体の定義を行う。すなわち、対象とする探索問題を、最適な染色体を求める問題に置き換える。ここでは、図6.5に示した、声質変換目的（感性表現語）とピッチ情報、パワー情報、時間構造の3種類の韻律変換係数の対応を示す1次元実数配列を染色体として定義している。すなわち、ピッチの拡大率、パワーの拡大率、継続長時間の伸縮率を各々 α 、 β 、 γ とすると、配列 $[\alpha, \beta, \gamma]$ を染色体と定義する。次に、個体の生成を行う。ここでは、韻律変換係数値をランダムに変えた20種類の染色体を各声質変換目的ごとに定義し、それぞれを個体と定義している。次に、満足できる声質に変換されるまで、評価、淘汰・選択、遺伝的操作を繰り返す。評価の方法としては、人間の主観に基づき得点を与える方法、予め目標とする音声波形を準備し、目標の波形と声質変換装置の出力音声波形との平均二乗誤差を評価結果とする方法他が考えられる。

一般的に、進化的計算では設計者が予め評価関数を明示的に定義し、この評価関数を使って自動的に世代交代を繰り返す。しかし、エンドユーザが評価関数を明示的に設定し評価する運用形態は考えにくいいため、ここでは、一世代ごとに人間が主観（感性）で評価を行う方式を採用する。進化的計算を人間との感性インタフェースに適用する新しい試みと考える。

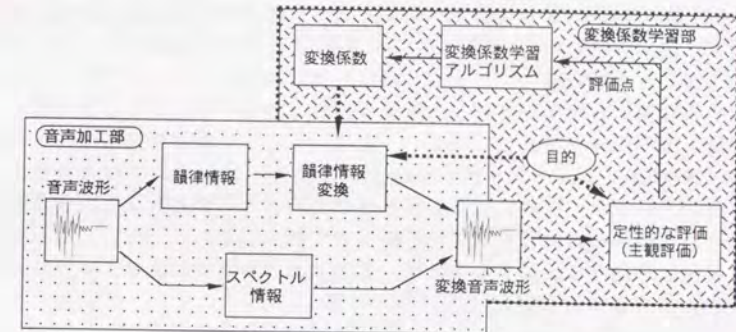


図6.3 声質変換システム構成概略

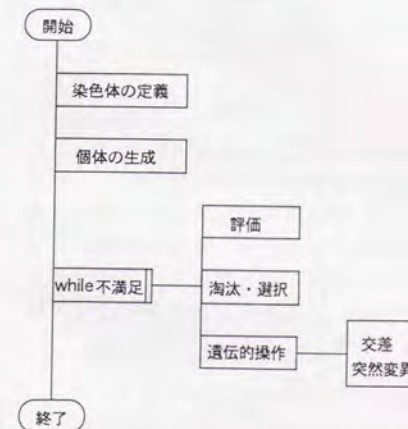


図6.4 遺伝的アルゴリズムのフローチャート

韻律変換係数に関する知識

声質変換目的	韻律変数		
	ピッチ	パワー	継続長
明瞭な声	1.172	1.365	0.918
かわいい声	1.383	-1.366	0.907
落ち着いた声	0.992	1.074	1.015

変換係数
(染色体)

図6.5 染色体の例

6.3.3 遺伝的操作

(1) 淘汰および増殖

評価点に基づき図6.6に示す淘汰・選択を行う。まず評価点に基づき個体のソートを行う。図6.6の例では、明瞭な声への変換という声質変換目的を持つ20個体について評価点の高い順にソートしている。次に淘汰を行う。ここでは、評価点の低い半数の個体を淘汰している。淘汰の割合は特に半数である必要はない。ある基準値以下の個体を淘汰してもよい。次に、淘汰した個体を補う操作を行う。図6.6の例ではルーレット選択[Goldberg 89]を採用している。2つの染色体を親として選び、後で示す交差と突然変異を施してできた子供で淘汰した個体を置き換えるという操作を淘汰した個体数を補うまで繰り返す。ここで、2つの親を選択する確率は評価点に比例する。また、選択における重複は許す。親は淘汰する個体も含めた全個体を候補とする。例えば図6.7に示すように、評価点の高い個体のみを親の候補とした場合、直ぐに局所解に収束（初期収束）してそこから脱出できなくなる可能性があるためである。すなわち、淘汰される個体の数を M 個とすると、今の世代の N 個の個体 $I_1 \sim I_N$ から二つの個体のペアを M 組だけ重複を許してランダムに選択し、それぞれに対して交差と突然変異を施して不足を補うための M 組の個体を決定する。ここで、ある個体 I_i が次の世代の個体を作るための親として選択される確率 $P(I_i)$ は次式によって決定する。

$$P(I_i) = f(I_i) / \left\{ \sum_{j=1}^N f(I_j) / N \right\} \quad (6.2)$$

ここで示した淘汰・選択は評価点の低い個体のみ遺伝的操作で生成した個体に置き換える。従って、淘汰・選択後の個体に対する評価点の最大値が淘汰・選択前の評価点の最大値を下回ることが有りえない。尚、ここでは各個体に評価点を付け、評価点の低い個体を淘汰したが、特に評価点を設けず、無作為に選んだ2つの個体の内どちらを使った方がより目的に合っているかを判断し、トーナメント制で淘汰する個体を決定することも可能である。この場合、親の選択は（評価点を参考にできないので）ランダムに行うことが考えられる。

(2) 交差と突然変異

図6.8と図6.9を用いて遺伝的操作の説明を行う。図6.8は交差の一例を示している。任意に選んだ列から後を2つの親の間で置き換えるという操作を行う。図6.8の例では、継続長に関する変換係数を、2つの親の間で、置き換える例を示している。交差という操作は、高い評価点につながるビット列（染色体の部分構

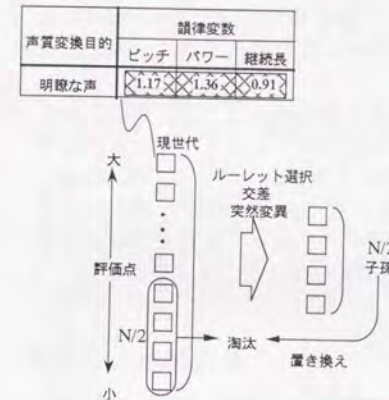


図6.6 個体の淘汰・選択方法の例

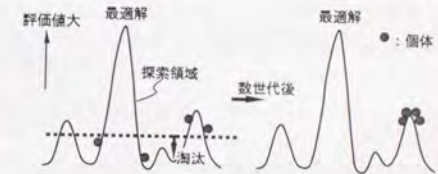


図6.7 評価値の高い個体のみ親として選ぶ探索の例

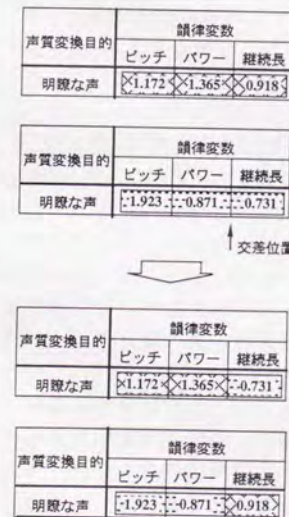


図6.8 韻律変換係数学習アルゴリズムにおける交差の例

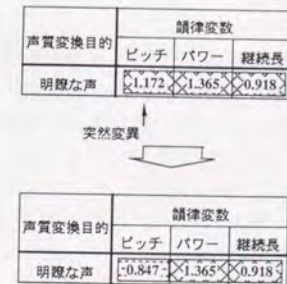


図6.9 韻律変換係数学習アルゴリズムにおける突然変異の例

造)を他の個体に伝搬する効果を持つ。このような部分構造 (building block [Goldberg 89]と呼ばれる)を的確に組み合わせることに成功すれば、効果的な探索に繋がる。

図6.9は突然変異の例を示している。任意に選んだ要素の値(変換係数)をランダムに変化させるという操作を行う。図6.9の例では、ピッチに関する変換係数が選ばれ、変換係数の値が1.17から0.84に変異した例を示している。ここでは、親の近傍に突然変異先を限定して局所探索能力を向上させるために式(6.3)で示す変異を用いている。 W_i は世代*i*の変換係数を表わす。 α は、突然変異の範囲を親の近傍に限定するための係数で、-0.1から0.1の範囲で乱数により決定する。

$$W_{i+1} = W_i + \alpha W_i \quad (-0.1 \leq \alpha \leq 0.1) \quad (6.3)$$

突然変異という操作は、局所解から脱出する効果があるとともに多様性の創出に効果を持つ。交差と突然変異という操作を組み合わせることで評価点の高かった、すなわち声質変換目的により近かった、変換係数の近傍を効率的に探索できる。また、複数の個体が異なる初期値から並行して探索を行うため、初期値依存性が少ない。また、並列化による効果も期待できる。尚、以下の実験では、交差率0.5、突然変異率0.3として実験を行う。

6.4 評価実験

6.4.1 音声刺激

女性話者1名に” 当社の紹介を致しましょう”の一文を発声させた音声 S_0 と、上述の進化的計算を用いた学習方式により得られた韻律変換係数を用いて、” 明瞭な”, ” かわいい”, ” はずんだ”, ” 落ち着いた”, ” 怒った”という5種類の感性表現語に対応する声質変換を行った音声 $SA_1 \sim SA_5$ と、設計者が経験に基づき試行錯誤で決定した韻律変換係数を用いて、同一の感性表現語に対応する声質変換を行った音声 $SB_1 \sim SB_5$ を音声刺激とした。

6.4.2 実験方法

被験者として、実験の目的を知らない20代~30代の10名の男女をランダムに選択した。刺激対 $S_0 - SA_i$ ($i = 1 \sim 5$) および $S_0 - SB_i$ ($i = 1 \sim 5$)を上記10名の被験者に対して提示(スピーカ受聴)し、各々両刺激が前記5種類の感性表現語に対応する声質変換が行われたかどうかを判断させた。判定は、” 感性表現語で示される目的に近づいた”, ” どちらも言えない”, ” 返って不自然感の方が強い”の3段階の中から選択させた。また、定量的な比較を行うために” かわいい”という感性表現語に対しては、併せて両刺激の達成度および好悪度に対してもそれぞれ9段階で評定させた。各々の刺激については複数回受聴することを許した。

6.4.3 実験結果および考察

$S_0 - SA_i$ ($i = 1 \sim 5$) および $S_0 - SB_i$ ($i = 1 \sim 5$)について、全被験者の判定結果を図6.10に示す。図6.10に示すグラフでは、” 感性表現語で示される目的に近づいた”, ” 返って不自然感の方が強い”の2種類の判定に対してヒストグラム表示を行っている。図6.10から、設計者が経験に基づき試行錯誤で決定した韻律変換係数を用いて声質変換を行った場合には、” 明瞭な”, ” かわいい”, ” はずんだ”などの一部の感性表現語に対応する声質変換には成功しているが、” 落ち着いた”, ” 怒った”などの感性表現語に対しては返って不自然感の方が強くなっていることが分かる。一方、進化的計算を用いた学習により獲得した韻律変換係数を用いて声質変換を行った場合には、全ての感性表現語に対応する声質変換に対して平均して良好な結果を示している。これは、韻律変換係数の組み合わせを考える場合、組み合わせの数が天文学的な数になり、人手で最適な組み合わせを探索することが非常に困難なためと考えられる。例えば、韻律変数間の組み合わせによっては、小数点以下4桁目や5桁目の変換係数の修正が声質の変化に影響

を与える場合が観測された。このことは、最適解を見つけるためにはある程度大きな有効数字を考える必要があり、人手で扱うには探索空間が大きくなりすぎる場合があることを示していると思われる。また、ピッチ、パワー、継続長の各韻律情報はそれぞれ独立ではなく、相互に関連しているため、順番に種類ずつ最適な係数を求めて行くことができない。すなわち、“明瞭な”など、過去の研究から声質の変換を行うための特徴量の制御ルールが定性的に分かっている感性表現語に対しては、人手でも試行錯誤を繰り返して適切な韻律変換係数の組み合わせを求めることが可能と思われる。一方、“怒った”など、声質の変換を行うための特徴量の制御ルールが不明確な感性表現語に関しては、人手での変換係数の探索は困難であり、本論文で紹介したように、進化的計算などの学習をうまく適用した変換係数の獲得が必要と思われる。

次に、達成度評定値に関する全被験者の平均を表わしたグラフおよび好悪度評定値に関する全被験者の平均を表わしたグラフを図6.11に示す。達成度では、経験に基づき人手で韻律変換係数の組み合わせを求めた場合の達成度+1.0に対して、進化的計算を用いて韻律変換係数の最適な組み合わせを求めた場合には+1.6となり、+0.6の改善がみられた。また、好悪度に対しては、-0.8から+0.6に、+1.4の改善がみられた。これは、人手の場合、ピッチを大きく（すなわち、声を高く）することで“かわいい声”に変換する傾向が強いために、機械的な音質となり、不自然感が残るためと思われる。一方、進化的計算を用いた場合には、目的に合った変換が行われ、しかも、音質の劣化は知覚されず加工した印象を与えないことが確認された。話速や声の高さをユーザが直接調整する機能は、一部製品化されている規則合成システム（任意文章から機械的に音声合成するシステム）でサポートされているが、変数の精度および変数間の相互作用を考えると、自然性を維持したまま声質を変換するためのパラメータを素人が直接調整することは非常に困難である。本研究では、“明瞭な”、“かわいい”等の定性的な目的を与えることでパラメータの調整を行う新しいアプローチと考える。

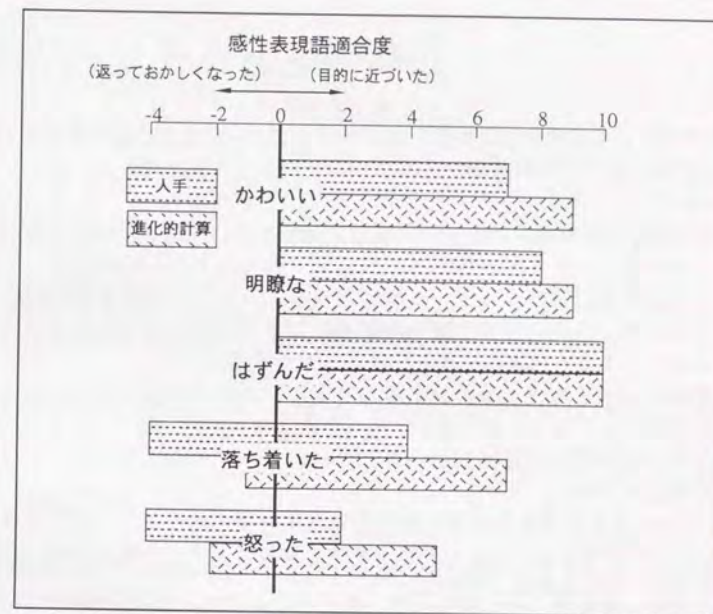


図6.10 感性表現語適合度の評価実験結果

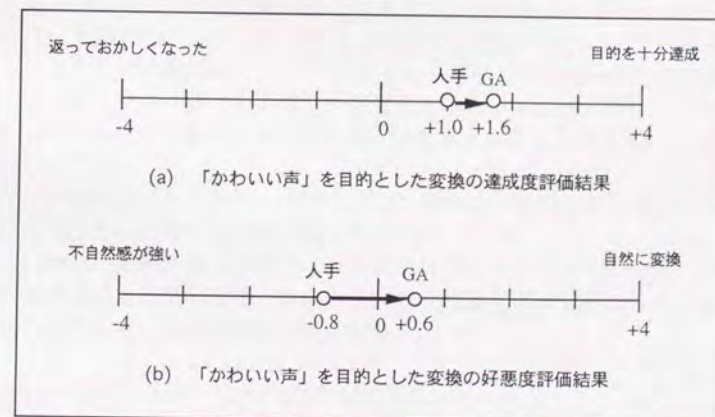


図6.11 「かわいい声」を目的とした変換の達成度と好悪度の評価実験結果

6.5 まとめ

素人の声質を表情豊かで明瞭性の高いナレーションに自動変換する技術として、進化的計算を用いた韻律係数フィッティングによる声質変換方式を提案した。

“明瞭な”，“かわいい”，“はずんだ”，“落ち着いた”，“怒った”という5種類の感性表現語に対応する声質変換実験を行い，以下の結論を得た。

- (1) “明瞭な”，“かわいい”等，声質の変換を行うための特徴量の制御ルールが定性的には分かっている感性表現語に対して，韻律変換係数のフィッティングに進化的計算を用いることにより，人手で変換係数を求める場合よりも効率良く声質の変換を行うことができる。
- (2) 声質の変換を行うための特徴量の制御ルールが不明確であり人手で変換係数の探索を行うことが困難な感性表現語に関しても，韻律変換係数のフィッティングに進化的計算を用いることにより，声質変換を行うことができる可能性を持つ。

今後，スペクトル情報の変換も併用した声質変換の精度向上，任意文章から自然性の高い音声合成するためのパラメータ調整に関する知識の獲得，などに関して検討する必要があると考える。

第7章 結論

この章では，研究成果をまとめて結論をのべるとともに，今後の課題をまとめる。

7.1 研究成果

この節ではまずこの研究によって得られた成果の概要を示し，その後，各項目について説明する。

7.1.1 研究成果の概要

この研究の目的は，動的問題を対象として進化的計算の有効な適用法を検討し，かつ，その具体的応用問題を探ることであった。すなわち，最適解が確率的に変化する離散空間上の問題および連続実数空間上の動的問題を対象として，進化的計算の有効な適用法を検討し，また，その探索能力を探ることを主な目標としてきた。具体的な応用問題としては，近年，マルチメディア情報処理に対するユーザニーズが急速に高まりつつあることを考慮して，音声情報処理への進化的計算の適用を狙った研究を行ってきた。特に，ニューラルネットワークとの融合を考えることにより，進化的計算とニューラルネットワークのお互いの欠点を補い合うことのできるモデルの提案を目指して研究してきた。

この研究で得られた主要な成果は次のようにまとめることができる。

- (1) 進化的計算のニューラルネットワークへの適用法の提案とその有効性の実証
- (2) 先読みモデルの提案およびその有効性の実証
- (3) 離散的空間の動的問題を対象とした進化的計算の適用可能性の実証
- (4) 連続実数空間上の動的問題を対象とした進化的計算の適用可能性の実証
- (5) 進化的計算を用いた声質変換技法の開発

これらの各項目について，次節以降で説明する。

7.1.2 進化的計算のニューラルネットワークへの適用法の提案とその有効性の実証

従来、進化的計算をニューラルネットワークに適用する場合、基本的に1次元染色体上でのランダムな位置での交差を仮定している。その結果、ニューラルネットワーク上の2次元スキーマを交差で破壊する確率が高くなる傾向がある。また、ニューラルネットワーク上のまとまった構造が1次元染色体上では分散配置されてしまうために、繰り返し構造の多い論理は生成しにくいと考えられる。これは、進化的計算では交差によりスキーマと呼ばれる部分構造を的確に組合せて効率的な探索を実現する可能性を有するが、このような部分構造は染色体の表現形式と交差の単位に依存する可能性が高いことを考慮されていないためであった。これに対して本研究は、ニューラルネットワーク自体を2次元の染色体と考え、ニューラルネットワーク上の2次元部分構造に着目して結線情報と結合荷重値の情報を同時に置き換える交差を提案し、例えば、排他的論理和を部分的に含んだ論理の合成に有効であることを示した。あるいは、ノイズなどの影響により、教師信号の一部が動的に変化する論理の合成に有効であることを示した。この研究により、ニューラルネットワークの合成を目的として、進化的計算の有効な適用法の一つを示すという目的を達成することができたと考える。

7.1.3 先読みモデルの提案とその有効性の実証

この研究の目的は、学習過程の部分的な情報を基に有効な処理を行うシステムを実現するために有効と思われる技術の確立であった。そのために、ここでは、2つのリカレントニューラルネットワークが相互作用を持ちながら先読み（外界からの応答の予測）を行うモデル（先読みモデル）を提案し、このモデルの有効性を示した。先読みモデルは、行動を決定するネットワークと環境の内部モデルからなり、それぞれリカレント型ニューラルネットワークで構成する。環境の内部モデルを学習により構築しながら先読みを行い、投機的な行動を実際に環境に対して行う。また、先読み結果を、行動を決定するネットワークの学習に利用する。すなわち、2つのネットワークは、相互作用を持ちながら学習を行う。このモデルの評価には、tic-tac-toeゲームを用いて、最適解が確率的に変化する離散的な問題を人工的に設定した。この評価実験から、行動を決定するために十分な情報が予め得られない状況下での行動制御を対象として、先読みモデルの有効性を示した。すなわち、この研究により、学習過程の部分的な情報を基に有効な処理を行うシステムを実現するために効果的と思われる技術の一つを提案することができたと考える。

7.1.4 離散的空間の動的問題を対象とした進化的計算の適用可能性の実証

本研究では、多数の局所解が存在する離散的な問題に、例えば時系列性や複数の局所解の中で最適解が確率的に変動するなどの動的条件が加わった場合に対しても有効なアルゴリズムを提案することを目的としていた。この論文においては、上記2つのリカレントニューラルネットワークが相互作用を持ちながら学習を行う先読みモデルの生成に進化的計算を適用し、最適解が確率的に変化する離散的な問題を人工的に設定して評価を行った。その結果、単純に進化的計算を適用するよりも、進化的計算とニューラルネットワークの融合により、最適解が離散空間上で確率的に変化する問題に対して、学習が効率的に行える可能性があることを示した。

7.1.5 連続実数空間上の動的問題を対象とした進化的計算の適用可能性の実証

本研究では、連続実数空間上の動的に変化する問題に対しても有効なアルゴリズムを提案することを目的としていた。本論文では、非周期時系列問題の一つとしてカオスダイナミックスの学習問題を取り上げ、進化的計算により合成する学習過程のリカレントニューラルネットワークの出力が、不動点、リミットサイクル、ストレンジアトラクタと変化することを計算機実験により検証した。すなわち、進化的計算とニューラルネットワークの融合により、カオスダイナミックスの隠れた特徴を学習できる強力なアルゴリズムを実現できる可能性があることを示した。これにより、例えば実世界において、カオスのなふるまいを示す信号であることが分かっているにもかかわらず、基になる生成規則が不明な問題に対して進化的計算の適用可能性があることを示すことができたと考える。

7.1.6 進化的計算を用いた声質変換技法の開発

本研究の最終的な目的は、実社会における具体的な動的応用問題に対して有効なアルゴリズムを提供することであった。本論文では、移動コンピューティングや広域ネットワークなど、マルチメディア時代への移行が急速に進んでいることを考慮して、音声情報処理に着目して進化的計算の適応力調査を試みた。その結果、ピッチ（声の高低情報）、パワー、時間構造などの韻律情報を進化的計算を用いて変換することにより、声質の変換が自然に行える可能性があることを示した。ここで示した技術は、例えば、ユーザ自身が自分の好みの声質に調整可能な音声ユーザインタフェースを持つマルチメディアPCなどに利用することができると考える。

7.2 結論

この研究の目的は、動的問題を対象として進化的計算の有効な適用法を検討し、かつ、その具体的応用問題を探ることであった。これらに関して順にのべる。

まずここで扱う動的問題としては、最適解が確率的に変化する離散空間上の問題および連続実数空間上の動的問題を対象とした。最適解が確率的に変化する離散空間上の問題に関しては、2つのリカレントニューラルネットワークが相互作用を持ちながら学習を行う先読みモデルを提案した。先読みモデルは、行動を決定するネットワークと環境の内部モデルからなり、それぞれリカレント型ニューラルネットワークで構成する。環境の内部モデルを学習により構築しながら先読みを行い、投機的な行動を実際に環境に対して行う。また、先読み結果を、行動を決定するネットワークの学習に利用する。この先読みモデルの獲得に進化的計算を適用し、行動を決定するために十分な情報が予め得られない状況下での行動制御を対象として、先読みモデルの有効性を示した。また、単純に進化的計算を適用するよりも、進化的計算とニューラルネットワークの融合により、最適解が離散空間上で確率的に変化する問題に対して、学習が効率的に行える可能性があることを示した。すなわち、この研究により、学習過程の部分的な情報を基に有効な処理を行うシステムを実現するために効果的と思われる技術の一つを提案することができたと考える。

一方、連続実数空間上の動的問題に関しては、非周期時系列問題の一つとしてカオスダイナミックスの学習問題を取り上げ、進化的計算により合成する学習過程のリカレントニューラルネットワークの出力が、不動点、リミットサイクル、ストレンジアトラクタと変化することを計算機実験により検証した。すなわち、進化的計算とニューラルネットワークの融合により、カオスダイナミックスの隠れた特徴を学習できるアルゴリズムを実現できる可能性があることを示した。これにより、例えば実世界において、カオス的なふるまいを示す信号であることが分かっているにもかかわらず、基になる生成規則が不明な問題に対して進化的計算の適用可能性を示すことができたと考える。

つぎに、実社会における具体的な動的応用問題に関していえば、本論文では、マルチメディア時代への移行が急速に進んでいることを考慮して、音声情報処理に着目して進化的計算の適応力調査を試みた。その結果、ピッチ（声の高低情報）、パワー、時間構造などの韻律情報を進化的計算を用いて変換することにより、声質の変換が自然に行える可能性があることを示した。ここで示した技術は、例えば、ユーザ自身が自分の好みの声質に調整可能な音声ユーザインタフェースを持つマルチメディアPCなどに利用できると考える。

7.3 今後の課題

各項目に対応させて、今後の課題を説明し、最後にそれらをまとめる。

7.3.1 進化的計算のニューラルネットワークへの適用法

ニューラルネットワーク自体を2次元の染色体と考え、ニューラルネットワーク上の2次元部分構造に着目して結線情報と結合荷重値の情報を同時に置き換える交差を行う表現型レベルでの遺伝的操作が、例えば、排他的論理和を部分的に含んだ論理の合成や、ノイズなどの影響により教師信号の一部分が動的に変化する論理の合成に有効である可能性が高いことを示した。しかし、ここで示した結果が、初期条件に依存せずどのような場合でも必ず成立するかどうかに関しては引き続き調査が必要である。例えば、ネットワークの初期構造や問題の性質などを変化させて傾向を調べる必要があると考える。また、一般的にどのような問題に対して有効になる可能性が高いかを継続的に調査し明確化していく必要があると考える。

7.3.2 先読みモデルの有効性の実証

2つのリカレントニューラルネットワークが相互作用を持ちながら学習を行う先読みモデルを提案し、環境の内部モデルを用いた先読みを行うことにより、実際の環境を介した学習回数の削減につながる可能性のあることを示した。しかし、ここでは解析の都合上、数回先の環境とのやり取りまで考慮する深い先読みを必要としない問題設定となり、リカレントニューラルネットワークの特徴を十分活かしていない。今後、より深い先読みを必要とする問題や時系列予測問題での評価が必要と考える。また、三つ以上のリカレントニューラルネットワークが相関学習を行うモデルでの評価が必要と考える。動的環境に自律的に適応するシステムの制御には、複数のモジュールの相互作用が必要と考えるからである。

7.3.3 離散的空間の動的問題を対象とした進化的計算の適用可能性の実証

先読みモデルの獲得に進化的計算を適用し、単純に進化的計算を適用するよりも、進化的計算とニューラルネットワークの融合により、最適解が離散空間上で確率的に変化する問題に対して、学習が効率的に行える可能性があることを示した。しかし、他の学習アルゴリズムを用いた場合との比較評価や、他のGAの適用方法との比較が必要と考えられる。例えば、Back Propagation Through Time (BPTT)などのアルゴリズムにより結合荷重値の学習を行った場合との比較が必

要と考える。すなわち、大規模なニューラルネットワークで結合荷重値の学習のみにBPTTなどのアルゴリズムを適用した場合、GAによるニューラルネットワークの構造決定とBPTTなどのアルゴリズムによる結合荷重値の学習を併用した場合、そして今回のようにニューラルネットワークの構造決定と結合荷重値の学習の両方にGAを適用した場合の比較が必要と考える。

7.3.4 連続実数空間上の動的問題を対象とした進化的計算の適用可能性の実証

ここでは、無限に続く非周期時系列データを対象とし、学習を打ち切った後のニューラルネットワークの出力に着目した評価（すなわち、時系列予測）を行っており、今までにない新しい試みと考える。また、カオスダイナミクスを対象として、進化的アルゴリズムの学習能力を探る上で貴重な実験結果を示せと考える。一方、ネットワークの結線情報を過去の経験を基に明示的に与え、結合荷重値の学習にBPTTアルゴリズムを用いた場合との比較を考えると、今回の実験結果に関する限り、進化的アルゴリズムによりニューラルネットワークの合成を行った場合には、リカレントニューラルネットワークの構造を明示的に与え結合荷重値の学習にBPTTアルゴリズムを用いた場合よりも、アトラクタ出現までに10倍程度の学習回数を必要とした。また、初期条件を変更した場合の再現性に関して現段階では十分とは言えず、遺伝的操作に必要なパラメータ調整が困難であった。ここでは、進化的アルゴリズムの特徴を積極的に活かすことにより、ニューラルネットワーク構造の自動的獲得が可能である、初期値に比較的左右されない、適応度を利用するだけで他の（勾配などの）情報を使わないため目的関数の性質が良く分からない問題にも適用できる、などの進化的アルゴリズムの利点を連続実数空間の探索問題にも活かすことを考えて実験を行ったが、結果的には、BPTTに匹敵する学習効率は得られなかった。結合荷重値の他にニューラルネットワークの構造も同時に獲得しようとしているために、明示的にネットワークの構造を与えた状態から結合荷重値の学習のみBPTTアルゴリズムで行う場合よりも、問題の設定が極端に難しくなっていることと、遺伝的操作におけるパラメータ設定に検討の余地を残しているためと思われる。他の対策案の一つとしては、進化的アルゴリズムによる広域探索とBPTTによる局所探索のハイブリッド化が考えられる。すなわち、初期段階を進化的アルゴリズムで探索し途中からBPTTなどのアルゴリズムに切り換えることが考えられる。但し、フィードフォワード型ニューラルネットワークで進化的アルゴリズムとBPなどの学習アルゴリズムのハイブリッド化を考える場合と異なり、分岐などの現象を考えると、進化的アルゴリズムからBPTTに切り換えるタイミングの選定が困難である。例えば、学習曲線からも分かるように、誤差関数の値がある一定値以下になったらBPTTに

切り換えるという単純な方式では対応できない。進化的アルゴリズムをどのようにリカレントニューラルネットワークの学習に取り込むかの検討が引き続き必要と思われる。

一方、今回の実験では、ローレンツアトラクタやリミットサイクルなど特徴的な出力を示す学習過程のネットワーク構造を調べてみると、必ず出力 x, y, z に対応したモジュール化の傾向が観測された。このことは、問題の特徴を考慮したモジュール化（すなわち、疎結合）構造が学習において重要な意味を持つことを示唆しているとも考えられる。今後、モジュール化に関する調査・研究が引き続き必要と考える。

7.3.5 進化的計算を用いた声質変換技法の開発

実社会における具体的な動的応用問題に関していえば、マルチメディア時代への移行が急速に進んでいることを考慮して、音声情報処理に着目して進化的計算の適応力調査を試みた。その結果、ピッチ（声の高低情報）、パワー、時間構造などの韻律情報を進化的計算を用いて変換することにより、声質の変換が自然に行える可能性があることを示した。しかし、本論文では、あくまでも一部の感性表現語に関する評価しか行っていない。今後、数多くの感性表現語を用いた評価が必要と考える。また、任意の感性表現語に関して自然な声質変換を行うためには、スペクトル情報の変換も併用した声質変換の精度向上が必要になる可能性があると考えられる。さらに、声質変換実験から、任意文章から自然性の高い音声を作成するためのにも進化的計算を利用できる可能性があると考えられるが、そのためにはパラメータ調整に関する知識の獲得などに関して今後検討する必要があると考える。

7.3.6 他の課題とまとめ

この論文においては、近年、実用上の準最適解を速やかに得ることのできる適応範囲の広い探索アルゴリズムとして注目されている進化的計算に着目し、その動的問題への適用法と応用に関する基礎的な検討と評価を行ってきた。特に、進化的計算とニューラルネットワークの融合によるアプローチを中心にまとめた。しかし、実社会において具体的な応用問題への適用を考える場合には、この研究成果を新たな目でみなおして新たなアプローチの可能性を探ることも必要だと考える。たとえば、進化的計算では、処理時間が実用上の問題となる場合がある。超並列計算機上での実行方法の検討や、進化的計算の処理を高速に実行可能な専用のハードウェアの研究も今後重要になると考えられる。

謝 辞

この研究に協力して頂いた以下の方々に深謝する。

東京大学の岡部 洋一教授には、指導教官として、この論文を洗練するためにご助力いただいた。また、東京大学の吉澤 修治教授、同合原 一幸助教授、同相澤 清晴助教授、同廣瀬 明助教授には、本論文を洗練するために、貴重な助言をいただいた。

現東邦大学教授の古谷 立美氏、新情報処理開発機構理論・新機能研究部長の岡隆一氏、同研究企画部長の大野 美恵氏には、筆者が新情報処理開発機構に出向していた当時の部長として本研究の方向付けに関して、さまざまな示唆をいただいた。筑波大学の星野 力教授には、人工生命に関する輪講会をとおして、本研究を進めるうえで多くの助言をいただいた。日立マイコンシステム第2設計部の落合 辰男氏、同ソフト開発部の阿部 行雄氏には、一部の評価用プログラムの開発および評価実験に関して著者と共同作業をしていただいた。現日立製作所中央研究所マルチメディアシステム研究部の金田 泰氏には、著者と同じく新情報処理開発機構に出向していた当時、本研究全般に渡り広く議論いただくとともに、さまざまな示唆をいただいた。現Stanford大教授のJohn R. Koza氏およびNatural Selection, Inc.のDavid B. Fogel氏には、本研究の中で特に、ニューラルネットワークの合成を目的とした表現型レベルの遺伝的操作に関して議論していただくとともに、さまざまな助言をいただいた。現日立製作所中央研究所マルチメディアシステム研究部の北原 義典氏および同安藤 ハル氏には、本研究の中で特に、進化的計算を用いた声質変換に関して一部実験環境の構築に協力していただいた。

また、現中央研究所プロセッサシステム研究部長杉江 昇氏、現日立ダブリン研究所の阿部 正博氏、現中央研究所主幹研究員藤澤 浩道氏、現中央研究所マルチメディアシステム研究部畑岡 信夫氏、現日立アメリカの大島 義光氏には、この研究開始当時から現在に至るまでの中央研究所におけるユニット・リーダーとしてこの研究をバックアップしていただいた。さらに、現新情報処理開発機構つくば研究所所長の島田潤一氏、現日立製作所理事の堀越 彌氏、同基礎研究所所長の浅井 彰二郎氏、同中央研究所所長の中村 道治氏、同中央研究所企画室長の長島 重夫氏、同中央研究所マルチメディアシステム研究部長横山 哲夫氏のほか中央研究所の方々には、この研究を管理者の立場から支援していただいた。

最後に、妻 未来子はこの論文をまとめるにあたってさまざまな面でバックアップしてくれた。

参考文献

- [Abe 89] 阿部匡伸, 田村震一, 桑原尚夫: "FFTスペクトルからの信号再生法による声質変換手法", 信学論(D-II), Vol. J72-D-II, No. 8, pp. 1180-1186 (1989).
- [Abe 94] 阿部匡伸: "音声変換処理技術", 信学技報, SP93-137, pp. 69-75 (1994).
- [Angeline 94] Angeline P.J., Saunders G.M., and Pollack J.B.: "An Evolutionary Algorithm that Constructs Recurrent Neural Networks", *IEEE Trans. on Neural Networks*, vol. 5, no. 1, pp. 54-65 (1994).
- [Bäck 93] Bäck T., Rudolph G., and Schwefel H.-P.: "Evolutionary Programming and Evolution Strategies: Similarities and Differences", *Proc. of the Second Ann. Conf. on Evolutionary Programming*, pp. 11-22 (1993).
- [Baluja 93] Baluja S.: "Structure and Performance of Fine-Grain Parallelism in Genetic Search", *Proc. of the Fifth Inter. Conf on Genetic Algorithms*, pp. 155-162 (1993).
- [Casdagli 92] Casdagli M., and Eubank S. (Eds.), "Nonlinear modeling and forecasting," Addison-Wesley (1992).
- [DeJong 75] DeJong K.: "An analysis of the behavior of a class of genetic adaptive systems", Ph.D.Thesis, University of Michigan (1975).
- [Doya 89] Doya K. and Yoshizawa S.: "Adaptive neural oscillator using continuous-time back-propagation learning", *Neural Networks*, 2, 375-385 (1989).
- [Doya 92] Doya K.: "Bifurcations in the Learning of Recurrent Neural Networks", *Proc. of Inter. Sym. on Circuits and Systems*, pp. 2777-2780 (1992).
- [Fahlman 88] Fahlman S.: "Faster-Learning Variations on Back-Propagation: An Empirical Study", *Proc. of the Connectionist Models Summer School*, Morgan Kaufmann (1988).
- [Fogel 66] Fogel L., Owens A., and Walsh M.: "Artificial intelligence through simulated evolution", New York, John Wiley (1966).
- [Fogel 94a] Fogel D.B. and Fogel L.J.: "Evolutionary Computation", *IEEE Trans. on Neural Networks*, vol. 5, no. 1, pp. 1-2 (1994).
- [Fogel 94b] Fogel D.B.: "An Introduction to Simulated Evolutionary Optimization", *IEEE Trans. on Neural Networks*, vol. 5, no. 1, pp. 3-14 (1994).

- [Furuya 93] Furuya T., Sato Y., Ito H., Higuchi T., and Suzuki Y. : "Self-Programming Network (SPN) - A Computational Model for Adaptive Evolutionary Computers", *Proc. of the Inter. Joint Conf. on Neural Networks*, pp. 733-737 (1993).
- [Goldberg 89] Goldberg D.E. : "Genetic Algorithms in Search, Optimization, and Machine Learning", Reading, MA: Addison-Wesley (1989).
- [Guckenheimer 83] Guckenheimer J., Holmes P. : Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields, Springer (1983).
- [Harp 89] Harp S.A., Samad T., and Guha A. : "Towards the Genetic Synthesis of Neural Networks", *Proc. of the Third Inter. Conf. on Genetic Algorithms*, pp. 360-369 (1989).
- [Hatano 94] Hatano S., Sato Y., Hatano H., and Furuya T. : "The Inverse Method for Recurrent Neural Networks", *Proc. of Inter. Conf. on Neural Networks'94*, (1994).
- [Hecht 92] Hecht-Nielsen, R. : "ニューロコンピューティング", 234-247, トッパン (1992).
- [Holland 75] Holland, J. H. : "Adaptation in Natural and Artificial Systems", The University of Michigan (1975), and MIT Press (1992).
- [Holland 91] Holland, J. H. : "Concerning the Emergence of Tag-Mediated Lookahead in Classifier Systems", *Emergent Computation*, pp. 188-201, MIT Press (1991).
- [Iba 93] Iba H., Kurita T., de Garis H., and Sato T. : System Identification using Structured Genetic Algorithms, *Proc. of the Fifth Inter. Conf. on Genetic Algorithms*, pp. 279-286 (1993).
- [Ito 82] 伊藤憲三, 斎藤収三 : "音声の音響的特徴パラメータが個人性の知覚に及ぼす影響", 信学論(A), Vol. J65-A, No. 1, pp. 101-108 (1982).
- [Jordan 89] Jordan, M. I. : "Serial order : A parallel distributed processing approach", *Advances in Connectionist Theory : Speech*, Erlbaum (1989).
- [Kato 95] 加藤裕之, 須貝康雄, 川瀬太郎 : GAを用いたリカレントニューラルネットワークによる翌日最大電力需要予測, 電学論B, 115-B, 8, pp. 875-882 (1995).
- [Kitahara 87] 北原義典, 武田昌一, 市川薫, 東倉洋一 : "音声言語認知における韻律の役割", 信学論(D), Vol. J70-D, No. 11 (1987).
- [Kitahara 89] 北原義典, 東倉洋一 : "音声の韻律情報と感情表現", 信学技報, SP88-158, pp. 27-32 (1989).
- [Kitahara 92] Kitahara, Y and Tohkura Y. : "Prosodic Control to Express Emotions for Man-Machine Speech Interaction, *IEICE Trans. Fundamentals*, Vol. E75, No. 2 (1992).

- [Kitahara 95] 北原義典, 長谷川隆 : "ユーザのマルチメディア感性演出支援方式の提案", 信学技報, HC94-95, pp. 63-70 (1995).
- [Kitano 90] Kitano H. : "Empirical Studies on the Speed of Convergence of the Neural Network Training by Genetic Algorithms", *Proc. of AAAI-90*, Boston (1990).
- [Kitano 90] Kitano H. : "Designing Neural Networks using Genetic Algorithms with Graph Generation System", *Complex Systems*, Vol. 4, No. 4 (1990).
- [Klatt 90] Klatt, D.H. and Klatt, L.C. : "Analysis, Synthesis, and Perception of Voice Quality Variations among Female and Male Talkers", *Journal of Acoustic Society America*, 87(2) (1990).
- [Koza 92] Koza J.R. : *Genetic Programming : On the Programming of Computers by Means of Natural Selection*, MIT Press (1992).
- [Kuwabara 83] 桑原尚夫, 大串健吾 : "アナウンサー音声の音響的特徴", 信学論(A), Vol. J66-A, No. 6, pp. 545-552 (1983).
- [Kuwabara 86] 桑原尚夫, 大串健吾 : "ホルマント周波数・バンド幅の独立制御と個人性判断", 信学論(A), Vol. J69-A, No. 4, pp. 509-517 (1986).
- [Kuwabara 86] 桑原尚夫, 都木徹 : "分析合成による声質変換とさ声改善への応用", 信学研資, SP86-57, pp. 45-52 (1986).
- [Lorenz 63] Lorenz E.N. : Deterministic non-periodic flow, *Journal of Atmospheric Sciences*, 20, pp. 130-141 (1963).
- [Manderick 89] Manderick B. and Spiessens P. : "Fine-Grained Parallel Genetic Algorithms", *Proc. of the Third Inter. Conf. on Genetic Algorithms*, pp. 428-433 (1989).
- [Matsuo 93] 松尾和洋, 安達統衛 : "遺伝的アルゴリズムと生態系の自己組織化", 北野宏明編 "遺伝的アルゴリズム", 産業図書, pp. 305-322 (1993).
- [Miller 89] Miller G.F., Todd P.M., and Hegde S.U. : "Designing Neural Networks using Genetic Algorithms", *Proc. of the Third Inter. Conf. on Genetic Algorithms*, pp. 379-384 (1989).
- [Montana 89] Montana D. and Davis L. : "Training Feedforward Neural Networks Using Genetic Algorithms", *Proc. of the Inter. Joint Conf. on Artificial Intelligence*, pp. 762-767 (1989).
- [Morita 86] 森田直孝, 板倉文忠 : "自己相関法による音声の時間軸での伸縮方法とその評価", 信学技報, EA86-5, pp. 9-16 (1986).

- [Mühlenbein 89] Mühlenbein H. : "Parallel Genetic Algorithms, Population Genetics and Combinatorial Optimization", *Proc. of the Third Inter. Conf. on Genetic Algorithms*, pp. 416-421 (1989).
- [Mühlenbein 91] Mühlenbein H., et al. : "The Parallel Genetic Algorithm as Function Optimizer", *Proc. of the Fourth Inter. Conf. on Genetic Algorithms*, pp. 271-278 (1991).
- [Murakami 90] 村上由彦, 佐藤雅昭 : "リカレントネットによる非線形ダイナミックスの近似とカオスの学習", 信学技法, **NC90-81** (1990).
- [Nagao 93] Nagao T., Agui T., and Nagahashi H. : "Structural evolution of neural networks having arbitrary connections by a genetic method", *IEICE Trans. INF & SYST.*, Vol. E76-D, No. 6, pp. 689-697 (1993).
- [Nakagawa 91] 中川聖一, 鹿野清宏, 東倉洋一 : "音声・聴覚と神経回路網モデル", オーム社 (1991).
- [Ochiai 95] 落合辰男, 佐藤裕二 : "遺伝的アルゴリズムにおけるエリート生命戦略の提案", 1995年信学会総合大会, D-16, p. 16 (1995).
- [Ogihara 94] 荻原拓也, 中西正和 : "遺伝的アルゴリズムを用いたリカレントニューラルネットの学習について", 情処第49回全大会, pp. 2-229-230 (1994).
- [Pearlmutter 89] Pearlmutter B.A. : Learning state space trajectories in recurrent neural networks, *Neural Computation*, Vol. 1, pp. 263-269 (1989).
- [Riolo 91] Riolo, R. L. : "Lookahead Planning and Latent Learning in a Classifier System", *From animals to animats*, pp. 316-326, MIT Press (1991).
- [Rumelhart 86] Rumelhart D.E., McClelland J.L., and the PDP Research Group : "Parallel Distributed Processing", Cambridge, MA: MIT Press (1986).
- [Santos 94] Santos J., Duro R.J. : Evolutionary Generation and Training of Recurrent Artificial Neural Networks, *Proc. of the First IEEE Inter. Conf. on Evolutionary Computation*, pp. 759-763 (1994).
- [Saravanan 93] Saravanan N. : Evolving Neural Networks: Application to a Prediction Problem, *Proc. of the Second Ann. Conf. on Evolutionary Programming*, pp. 72-78 (1993).
- [Sato 90] Sato, M., Murakami, Y. and Joe, K. : "Learning chaotic dynamics by recurrent neural networks", *Proc. of the Inter. Conf. on Fuzzy Logic & N. N.* pp. 601-605 (1990).
- [Sato 93] 佐藤省三, 久野弘明, 郷原一寿 : "リカレントネットワークによる音波波形の学習", 信学技法, **NC93-56** (1993).

- [Sato 74] 佐藤大和 : "女声を特徴づける音響パラメータの研究", 信学論(A), Vol. J57-A, No. 1, pp.23-30 (1974).
- [Sato 93a] Sato Y., Shibata K., Asai M., and Sugie M. : "Development of a High-Performance, General Purpose Neuro-Computer Composed of 512 Digital Neurons", *Proc. of the Inter. Joint Conf. on Neural Networks*, pp. 1967-1970 (1993).
- [Sato 93b] Sato Y., Furuya T. : "Artificial Thought Model for Realizing Flexible Information Processing", *Proc. of the Inter. Joint Conf. on Neural Networks*, pp. 2909-2912 (1993).
- [Sato 94a] 佐藤裕二 : "リカレントニューラルネットワークにおける先読みと共進化", 第4回並列人工知能研究会, pp. 1-6 (1994).
- [Sato 94b] Sato Y., Hatano S., Hatano H., and Furuya T. : "Lookahead Planning and Co-Evolution in Recurrent Neural Networks", *Proc. of the First IEEE Inter. Conf. on Evolutionary Computation*, pp. 764-769 (1994).
- [Sato 94c] 佐藤裕二, 落合辰男, 波多野祥二 : "リカレントニューラルネットワークを基本とした先読みモデルの提案", 情処第49回全大会, pp. 2-265-266 (1994).
- [Sato 94d] Sato Y., Ochiai T., and Hatano S. : "Lookahead Planning Model based on Recurrent Neural Networks and 2-D Genetic Algorithm Operation", *Proc. of the First Inter. Conf. on Neural Information Processing*, pp. 798-803 (1994).
- [Sato 95a] Sato Y., Ochiai T. : "2-D Genetic Algorithms for Determining Neural Network Structure and Weights", in *Evolutionary Programming IV (Proc. of the Fourth Ann. Conf. on Evolutionary Programming)*, McDonnell J.R., Reynolds R.G., and Fogel D.B., Eds. MIT Press, pp. 789-804 (1995).
- [Sato 95b] 佐藤裕二, 古谷立美 : "GAを用いたリカレントニューラルネットワーク共進化モデルの学習", 信学論 (D-II), J78-D-II, No. 5, pp. 836-844 (1995).
- [Sato 95c] 佐藤裕二, 古谷立美 : "適応システムにおける先読みモデルの進化的獲得", 北野宏明編 "遺伝的アルゴリズム 2", 産業図書, pp. 147-173 (1995).
- [Sato 95d] Sato Y. : "Adaptive Parallel Genetic Algorithms that Depend on Fitness Landscapes for the Synthesis of Neural Networks", *Abstracts book of the Third European Conf. on Artificial Life*, p. 79 (1995).
- [Sato 95e] 佐藤裕二, 落合辰男 : "ニューラルネットワーク合成のための2次元GAの提案", 1995年信学会総合大会, D-40, p. 44 (1995).
- [Sato 95f] Sato Y., Hatano S., Abe Y., Ochiai T., and Oka R. : Development of a High-performance Simulator for Cooperative Clustered Recurrent Neural Networks, *Proc. of the Second Inter. Conf. on Neural Information Processing*, pp. 637-641 (1995).

- [Sato 96a] Sato Y., Nagaya S. : "Evolutionary Algorithms that Generate Recurrent Neural Networks for Learning Chaos Dynamics", *Proc. of the Third IEEE Inter. Conf. on Evolutionary Computation*, pp. 144-149 (1996).
- [Sato 96b] 佐藤裕二, 落合辰男, 阿部行雄 : "群リカレントニューラルネットワーク評価用高速シミュレータの開発", 信学論 (D-II), J79-D-II, 9, pp. 1594-1602 (1996).
- [Sato 96c] Sato Y., Kuwabara Y., and Oka R. : "Real-Time Learning of Chaos Dynamics Using a High-Performance Simulator on a Neuro-Computer", *Proc. of the Third Inter. Conf. on Neural Information Processing*, pp. 1262-1267 (1996).
- [Sato 96d] 佐藤裕二 : "進化的手法を用いて合成したリカレントニューラルネットワークによるカオスダイナミックスの学習に関して", 情処学論, Vol. 37, No. 11, pp. 1960-1968 (1996).
- [Sato 96e] 佐藤裕二, 安藤ハル, 北原義典 : "進化的計算を用いた韻律係数フィッティングによる声質変換に関して", 第12回ヒューマンインタフェースシンポジウム論文集, pp. 469-475 (1996).
- [Sato 97a] Sato Y. : "Voice Conversion Using Evolutionary Computation of Prosodic Control", *Proc. of the Australasia-Pacific Forum on Intelligent Processing and Manufacturing of Materials* (1997). (掲載予定)
- [Sato 97b] 佐藤裕二 : "表現型レベルの遺伝的操作を用いたニューラルネットワークの合成", 日本神経回路学会投稿中
- [Sawai 93] 澤井秀文 : "遺伝的アルゴリズムの画像復元への応用", 北野宏明編 "遺伝的アルゴリズム", 産業図書, pp. 209-231 (1993).
- [Schwefel] Schwefel H.-P. : "Kybernetische evolution als strategie der experimentellen forschung in der strömungstechnik", Diploma thesis, Technical Univ. of Berlin (1965).
- [Sebald 92] Sebald A.V. and Fogel D.B. : "Design of Fault Tolerant Neural Networks For Pattern Classification", *Proc. of the First Ann. Conf. on Evolutionary Programming*, pp. 90-99 (1992).
- [Spears 91] Spears W.M. and De Jong K.A. : "On the Virtues of Parameterized Uniform Crossover", *Proc. of the Fourth Inter. Conf. on Genetic Algorithms*, pp. 230-236 (1991).
- [Suzuki 81] 鈴木光男 : "ゲーム理論入門", 共立出版 (1981).
- [Syswerda 89] Syswerda G. : "Uniform Crossover in Genetic Algorithms", *Proc. of the Third Inter. Conf. on Genetic Algorithms*, pp. 2-9 (1989).

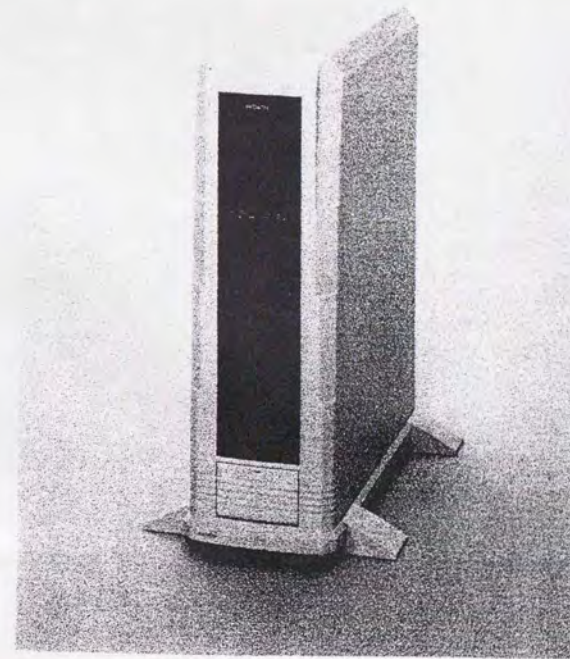
- [Takagi 94] 高木秀彦, 水野政司, 郷原一寿 : "リカレントニューラルネットワークのBPTTを応用した時間前向き計算による学習法とアトラクタ", 信学論 (D-II), J77-D-II, 2, pp. 422-430 (1994).
- [Takagi 90] 都木徹, 梅田哲夫 : "ピッチ変更時のひずみをスペクトル領域で修正する声質変換方式とその品質の心理評価", 信学論 (A), Vol. J73-A, No. 3, pp. 387-396 (1990).
- [Weigend 93] Weigend A.S., and Gershenfeld N.A. (Eds.), "Time series prediction: forecasting the future and understanding the past," Addison-Wesley (1993).
- [Whitley 89] Whitley D. and Hanson T. : "Optimizing Neural Networks Using Faster, More Accurate Genetic Search", *Proc. of the Third Inter. Conf. on Genetic Algorithms*, pp. 391-396 (1989).
- [Williams 89] Williams R.J. and Zipser D. : "Experimental analysis of the real-time recurrent learning algorithms", *Connection Science*, 1, pp. 87-111 (1989).
- [Winograd 72] Winograd T. : "Understanding Natural Language", Academic Press, New York (1972).
- [Zhang 93] Zhang B.T., Mühlenbein H., "Genetic Programming of Minimal Neural Nets Using Occam's Razor," *Proc. of the Fifth Inter. Conf. on Genetic Algorithms*, pp. 342-349 (1993).

付録 1

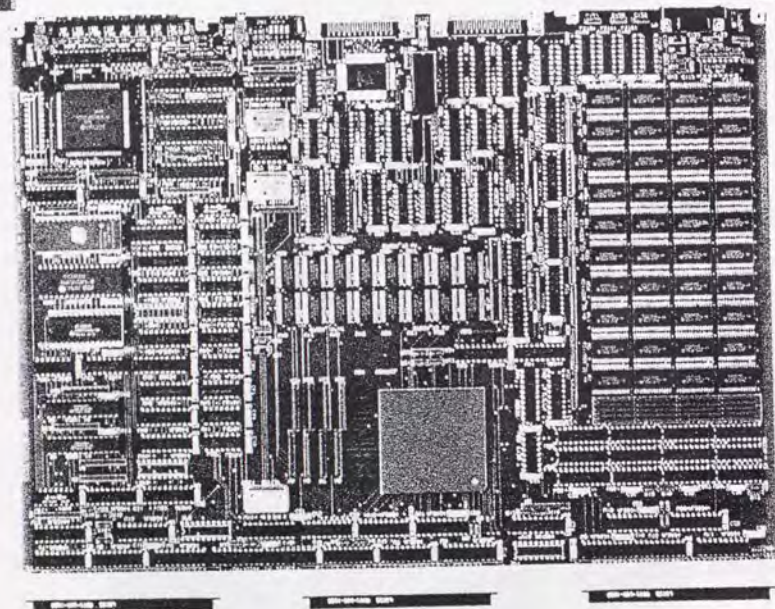
以下、付録として、第5章で紹介した、筆者らが開発した国内初のニューロコンピュータ（MY-NEUPOWER）の仕様を付表1に示す。製品化されたニューロコンピュータとしては、現在、世界最高レベルの性能を示している。また、外観写真、制御ボード写真、ニューロボード写真およびニューロチップ写真を付図1～付図4に示す。

付表 1 MY-NEUPOWERの仕様一覧表

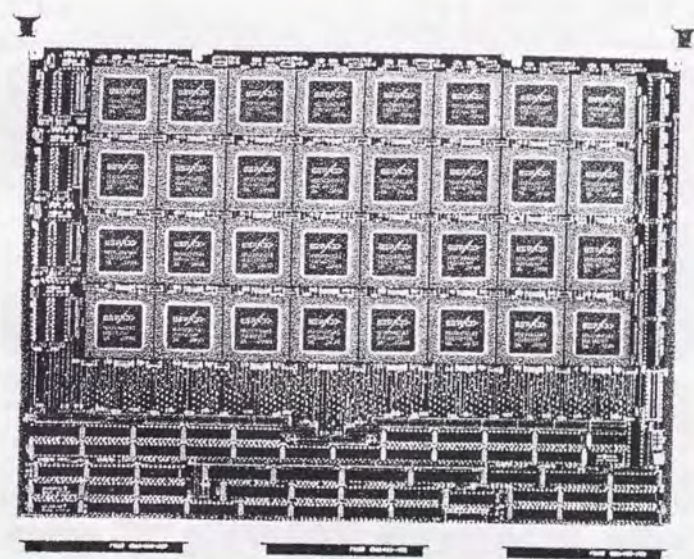
Circuit	Completely Digital
Architecture	Micro-programming Architecture Horizontal Microinstruction Format Nanomemory for Parameters Time-Sharing Digital Bus
Complexity	512 Physical Neurons/System 4096 Virtual Neurons/System 1024 Synapses/Neuron
Performance	1.26 GCUPS (MAX) 12.8 GCPS (MAX)
Learning Algorithm	General Purpose (BP, HF, LVQ etc.)
Neuron Output	10 bits / 16 bits
Synapse Weight	8 bits / 16 bits
Clock Rate	25 MHz
Process	1.0 μ m CMOS
Interface	SCSI
Host	SUN, 3050, etc.



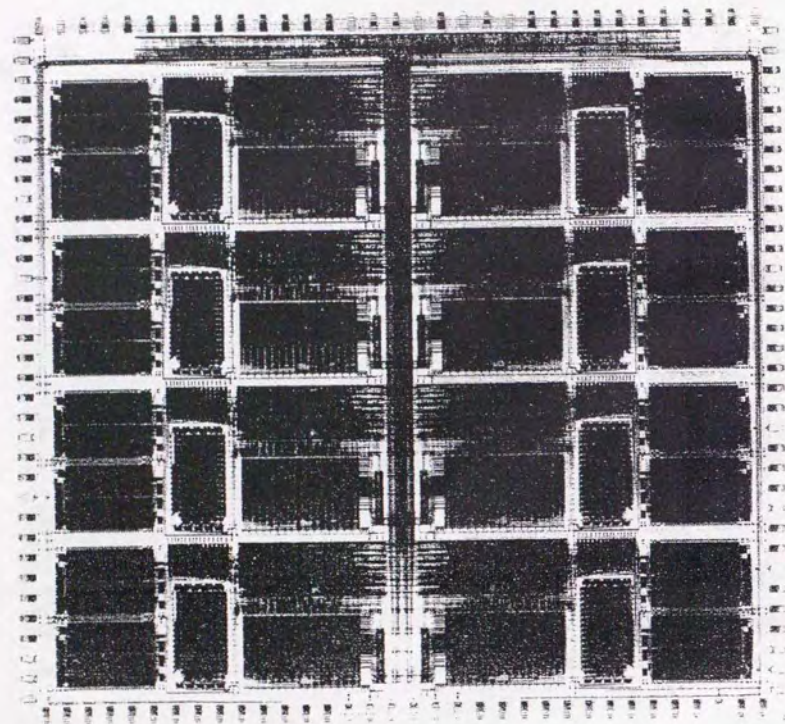
付図 1 MY-NEUPOWERの外観写真



付図2 制御ボード写真



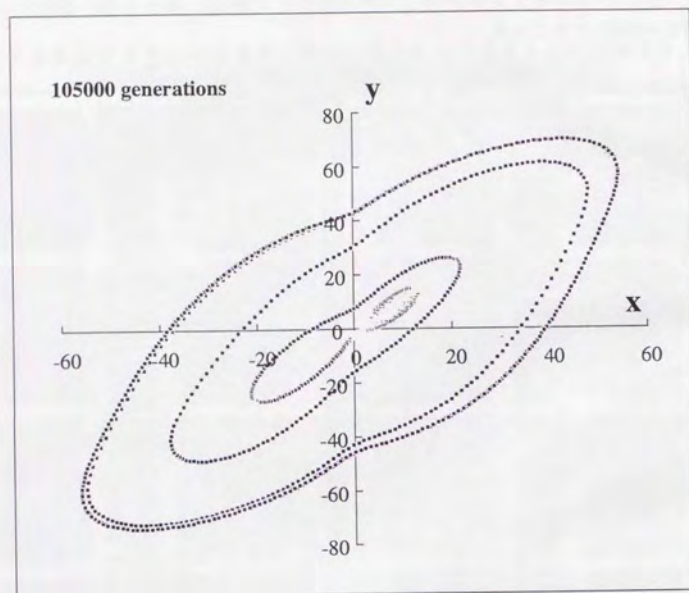
付図3 ニューロボード写真



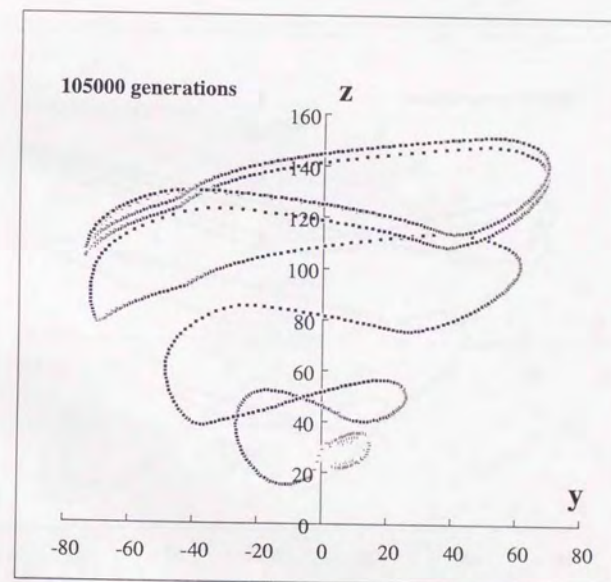
付図4 ニューロチップ写真

付録2

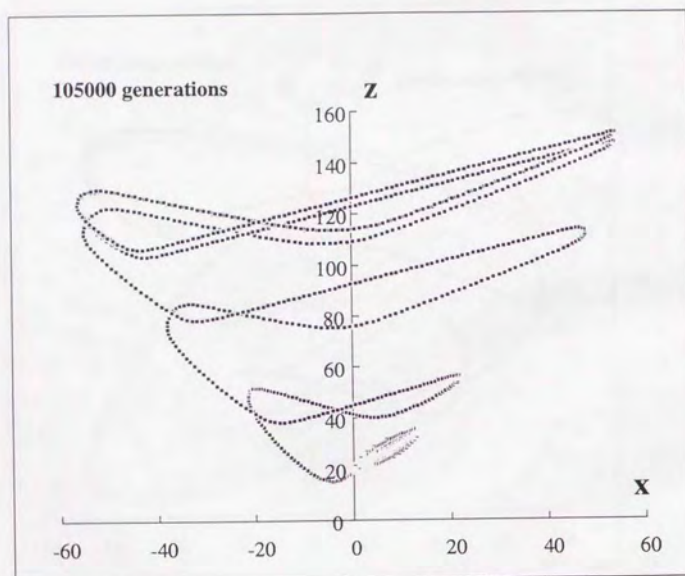
以下、付録として、第5章（図5.7）で紹介した、進化的計算により合成した学習途中のリカレントニューラルネットワークが出力した軌道の中から他の特徴的な出力の例を付図5～付図10に示す。



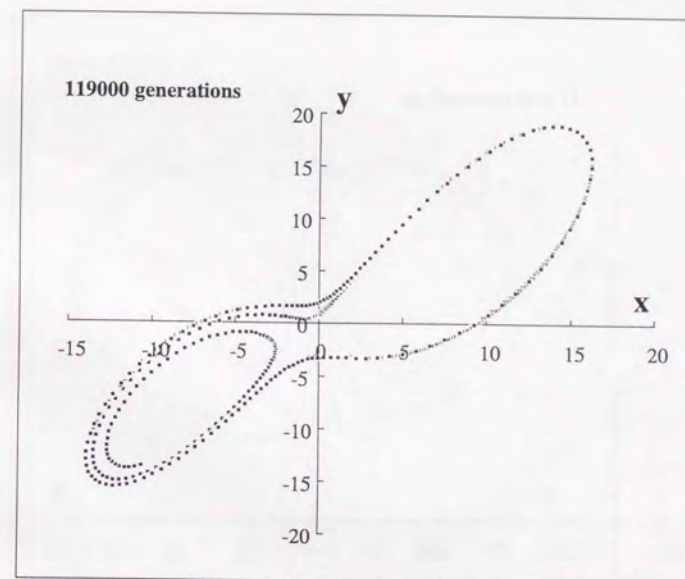
付図5 進化的計算により合成した学習途中のリカレントニューラルネットワークが出力した軌道の例



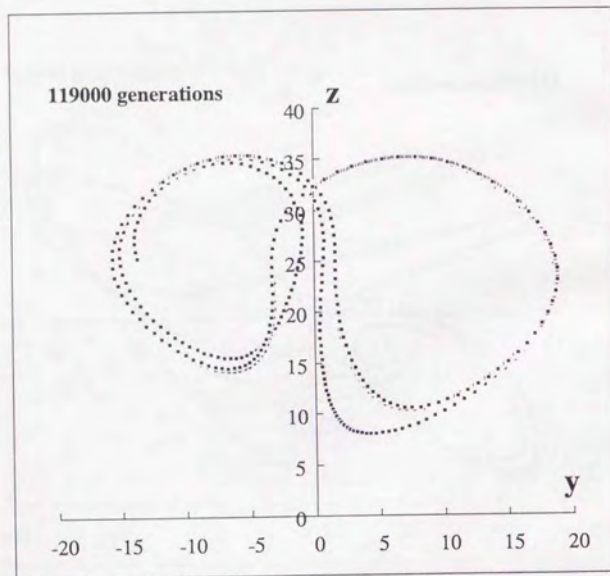
付図6 進化的計算により合成した学習途中のリカレントニューラルネットワークが出力した軌道の例



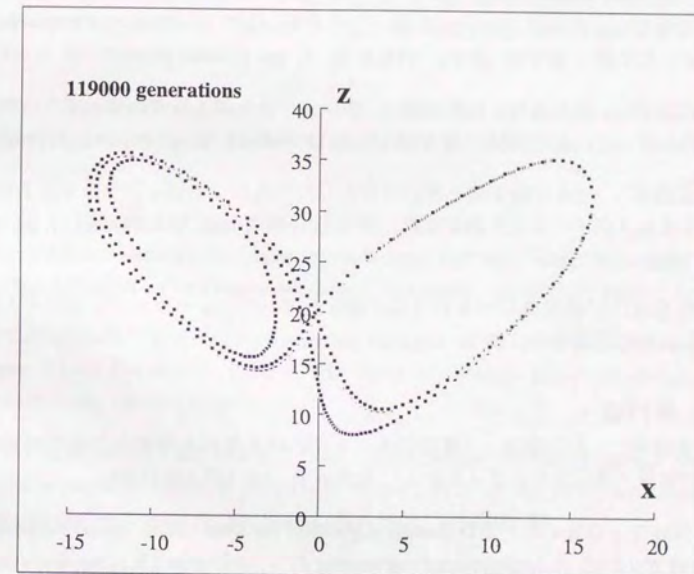
付図7 進化的計算により合成した学習途中のリカレントニューラルネットワークが出力した軌道の例



付図8 進化的計算により合成した学習途中のリカレントニューラルネットワークが出力した軌道の例



付図 9 進化的計算により合成した学習途中のリカレントニューラルネットワークが出力した軌道の例



付図 10 進化的計算により合成した学習途中のリカレントニューラルネットワークが出力した軌道の例

著者の主な出版物

以下、本論文に直接関係する著者の主な出版物を示す。

1. 論文誌:

- [1] 佐藤裕二, 古谷立美: "GAを用いたリカレントニューラルネットワーク共進化モデルの学習", 信学論 (D-II), **J78-D-II**, 5, pp. 836-844 (1995).
- [2] 佐藤裕二, 落合辰男, 阿部行雄: "群リカレントニューラルネットワーク評価用高速シミュレータの開発", 信学論 (D-II), **J79-D-II**, 9, pp. 1594-1602 (1996).
- [3] 佐藤裕二: "進化的手法を用いて合成したリカレントニューラルネットワークによるカオスダイナミックスの学習に関して", 情処学論, Vol. 37, No. 11, pp. 1960-1968 (1996).
- [4] 佐藤裕二: "表現型レベルの遺伝的操作を用いたニューラルネットワークの合成", 日本神経回路学会投稿中

2. 専門書:

- [1] 佐藤裕二, 古谷立美: "適応システムにおける先読みモデルの進化的獲得", 北野宏明編 "遺伝的アルゴリズム 2", 産業図書, pp. 147-173 (1995).
- [2] Sato Y., Ochiai T.: "2-D Genetic Algorithms for Determining Neural Network Structure and Weights", in *Evolutionary Programming IV*, McDonnell J.R., Reynolds R.G., and Fogel D.B., Eds. MIT Press, pp. 789-804 (1995).

3. 国際会議:

- [1] Sato Y., Shibata K., Asai M., and Sugie M.: "Development of a High-Performance, General Purpose Neuro-Computer Composed of 512 Digital Neurons", *Proc. of the Inter. Joint Conf. on Neural Networks*, pp. 1967-1970 (1993).
- [2] Sato Y., Furuya T.: "Artificial Thought Model for Realizing Flexible Information Processing", *Proc. of the Inter. Joint Conf. on Neural Networks*, pp. 2909-2912 (1993).
- [3] Sato Y., Hatano S., Hatano H., and Furuya T.: "Lookahead Planning and Co-Evolution in Recurrent Neural Networks", *Proc. of the First IEEE Inter. Conf. on Evolutionary Computation*, pp. 764-769 (1994).

- [4] Sato Y., Ochiai T., and Hatano S.: "Lookahead Planning Model based on Recurrent Neural Networks and 2-D Genetic Algorithm Operation", *Proc. of the First Inter. Conf. on Neural Information Processing*, pp. 798-803 (1994).
- [5] Sato Y., Ochiai T.: "2-D Genetic Algorithms for Determining Neural Network Structure and Weights", in *Evolutionary Programming IV (Proc. of the Fourth Ann. Conf. on Evolutionary Programming)*, McDonnell J.R., Reynolds R.G., and Fogel D.B., Eds. MIT Press, pp. 789-804 (1995).
- [6] Sato Y.: "Adaptive Parallel Genetic Algorithms that Depend on Fitness Landscapes for the Synthesis of Neural Networks", *Abstracts book of the Third European Conf. on Artificial Life*, p. 79 (1995).
- [7] Sato Y., Hatano S., Abe Y., Ochiai T., and Oka R.: "Development of a High-performance Simulator for Cooperative Clustered Recurrent Neural Networks", *Proc. of the Second Inter. Conf. on Neural Information Processing*, pp. 637-641 (1995).
- [8] Sato Y., Nagaya S.: "Evolutionary Algorithms that Generate Recurrent Neural Networks for Learning Chaos Dynamics", *Proc. of the Third IEEE Inter. Conf. on Evolutionary Computation*, pp. 144-149 (1996).
- [9] Sato Y., Kuwabara Y., and Oka R.: "Real-Time Learning of Chaos Dynamics Using a High-Performance Simulator on a Neuro-Computer", *Proc. of the Third Inter. Conf. on Neural Information Processing*, pp. 1262-1268 (1996). (招待講演)
- [10] Sato Y.: "Voice Conversion Using Evolutionary Computation of Prosodic Control", *Proc. of the Australasia-Pacific Forum on Intelligent Processing and Manufacturing of Materials* (1997). (掲載予定, 招待講演)
- [11] Furuya T., Sato Y., Ito H., Higuchi T., and Suzuki Y.: "Self-Programming Network (SPN) - A Computational Model for Adaptive Evolutionary Computers", *Proc. of the Inter. Joint Conf. on Neural Networks*, pp. 733-737 (1993).
- [12] Hatano S., Sato Y., Hatano H., and Furuya T.: "The Inverse Method for Recurrent Neural Networks", *Proc. of the Inter. Conf. on Neural Networks*, (1994).

4. 国内講演会, 研究会論文:

- [1] 佐藤裕二, 古谷立美: "柔らかな情報処理を実現する為の人工的思考モデルの提案", 1993年信学会秋季大会, **D-38**, 6-40 (1993).
- [2] 佐藤裕二: "リカレントニューラルネットワークにおける先読みと共進化", 第4回並列人工知能研究会, pp. 1-6 (1994).
- [3] 佐藤裕二, 落合辰男, 波多野祥二: "リカレントニューラルネットワークを基本とした先読みモデルの提案", 情処第49回全大会, pp. 2-265-266 (1994).
- [4] 佐藤裕二, 落合辰男: "ニューラルネットワーク合成のための2次元GAの提案", 1995年信学会総合大会, **D-40**, p. 44 (1995).
- [5] 佐藤裕二, 安藤ハル, 北原義典: "進化的計算を用いた韻律係数フィッティングによる声質変換に関して", 第12回ヒューマンインタフェースシンポジウム論文集, pp. 469-475 (1996).
- [6] 落合辰男, 佐藤裕二: "遺伝的アルゴリズムにおけるエリート生命戦略の提案", 1995年信学会総合大会, **D-16**, p. 16 (1995).

以下, 本論文以外の著者の主な出版物を示す.

5. 論文誌:

- [1] Nakamura H., Sawase T., Akao Y., Masumura S., Hayashi M., Ohsuga H., Sato Y. and Aizawa T.: "An Intelligent Subprocessor for Hardware Emulation with 20-MOPS Performance", *IEEE Journal of Solid-State Circuits*, Vol. 26, No. 11, pp.1662-1668 (1991).
- [2] 岩崎一彦, イゼリ・クリスチャン, 佐藤裕二: "超並列計算機用VLSIに適した結合網の一提案", 信学論 (D-I), 並列/分散処理特集号, **J75-D-I**, 8, pp. 583-591 (1992).
- [3] Hancu M., Iwasaki K., Sato Y. and Sugie M.: "Experimental results on the error detection capability of a concurrent test architecture for massively-parallel computers", *Parallel Computing* 18, pp. 1079-1103 (1992).
- [4] Hancu M., Iwasaki K., Sato Y. and Sugie M.: "A Concurrent Test Architecture for Massively Parallel Computers and Its Error Detection Capability", *IEEE Trans. on Parallel and Distributed Systems*, Vol. 5, No. 11, pp. 1169-1184 (1994).

6. 国際会議:

- [1] Nakamura H., Sawase T., Akao Y., Masumura S., Hayashi M., Ohsuga H. and Sato Y.: "An Intelligent Subprocessor for Hardware Emulation with 20 MOPS Performance", 1991 *IEEE Inter. Solid-State Circuits Conference (ISSCC91) Digest of Technical Papers* (1991).
- [2] Hancu M., Iwasaki K., Sato Y. and Sugie M.: "A Concurrent Test Architecture for Massively-Parallel Computers and Its Error Detection Capability", *Proc. of the IEEE Inter. Test Conf.* 1991, pp. 758-767 (1991).

7. 国内講演会, 研究会論文:

- [1] 佐藤裕二, 岩崎一彦, 澤瀬照美, 山口昇: "冗長技術を用いた並列マシンの歩留向上に関する検討", 信学技法, **FTS90-17**, pp. 67-72 (1990).
- [2] 佐藤裕二, ハンク・マリウス, 杉江衛, 岩崎一彦: "超並列マシンのネットワークの自己テスト方法に関する検討", 信学技法, **FTS91-21**, pp. 17-23 (1991).
- [3] 佐藤裕二, 岩崎一彦, ハンク・マリウス, 杉江衛: "超並列マシンのネットワークのオンライン自己テスト方式の検討", 1991年信学会秋季大会, **SD-8-1**, 6-323 (1991).
- [4] 野口孝樹, 佐藤裕二: "機能レベル論理設計の論理圧縮のための制御コード割当問題に関する考察", 1989年信学会秋季大会, **A-243**, 1-246 (1989).
- [5] 澤瀬照美, 中村英夫, 赤尾泰, 大須賀宏, 佐藤裕二, 増村茂樹, 林誠: "20MOPS性能のインテリジェント・サブプロセッサ", 信学技法, **ICD91-5**, pp. 29-34 (1991). (招待論文)
- [6] ハンク・マリウス, 岩崎一彦, 佐藤裕二, 杉江衛: "The Error Detection Capability of A Concurrent Test Architecture for Massively-Parallel Computers", 信学技法, **FTS91-2**, pp. 9-16 (1991).
- [7] 岩崎一彦, 佐藤裕二, イゼリ・クリスチャン: "VLSI超々並列計算機に適した結合網の一提案", 信学技法, **CPSY91-12**, pp. 67-73 (1991).
- [8] 岩崎一彦, イゼリ・クリスチャン, 佐藤裕二: "超々並列計算機用PEの演算サイズに関する一検討", 1991年信学会秋季大会, **D-55**, 6-57 (1991).
- [9] 岩崎一彦, 佐藤裕二: "バケット数カウントによる超並列計算機のセルフチェック", 1992年信学会春季大会, **D-244**, 6-246 (1992).

章と論文の対応

以下、各章と著者の主な論文との対応を示す。

第1章：序論

第2章：進化的計算を用いたニューラルネットワークの合成 - 1 従来の手法

第3章：進化的計算を用いたニューラルネットワークの合成 - 2 表現型レベルでの遺伝操作

- [1] 佐藤裕二："表現型レベルの遺伝的操作を用いたニューラルネットワークの合成", 日本神経回路学会投稿中
- [2] Sato Y., Ochiai T. : "2-D Genetic Algorithms for Determining Neural Network Structure and Weights", in *Evolutionary Programming IV (Proc. of the Fourth Ann. Conf. on Evolutionary Programming)*, McDonnell J.R., Reynolds R.G., and Fogel D.B., Eds. MIT Press, pp. 789-804 (1995).
- [3] Sato Y. : "Adaptive Parallel Genetic Algorithms that Depend on Fitness Landscapes for the Synthesis of Neural Networks", *Abstracts book of the Third European Conf. on Artificial Life*, p. 79 (1995).

第4章：自律システム制御のための先読みモデルの進化的獲得

- [1] 佐藤裕二, 古谷立美："GAを用いたリカレントニューラルネットワーク共進化モデルの学習", 信学論 (D-II), **J78-D-II**, 5, pp. 836-844 (1995).
- [2] Sato Y., Hatano S., Hatano H., and Furuya T. : "Lookahead Planning and Co-Evolution in Recurrent Neural Networks", *Proc. of the First IEEE Inter. Conf. on Evolutionary Computation*, pp. 764-769 (1994).
- [3] Sato Y., Ochiai T., and Hatano S. : "Lookahead Planning Model based on Recurrent Neural Networks and 2-D Genetic Algorithm Operation", *Proc. of the First Inter. Conf. on Neural Information Processing*, pp. 798-803 (1994).
- [4] 佐藤裕二, 古谷立美："適応システムにおける先読みモデルの進化的獲得", 北野宏明編 "遺伝的アルゴリズム 2", 産業図書, pp. 147-173 (1995).

第5章：カオスダイナミックスを用いた進化的計算の適応力評価実験

- [1] 佐藤裕二："進化的手法を用いて合成したリカレントニューラルネットワークによるカオスダイナミックスの学習に関して", 情処学論, Vol. 37, No. 11, pp. 1960-1968 (1996).
- [2] 佐藤裕二, 落合辰男, 阿部行雄："群リカレントニューラルネットワーク評価用高速シミュレータの開発", 信学論 (D-II), **J79-D-II**, 9, pp. 1594-1602 (1996).
- [3] Sato Y., Shibata K., Asai M., and Sugie M. : "Development of a High-Performance, General Purpose Neuro-Computer Composed of 512 Digital Neurons", *Proc. of the Inter. Joint Conf. on Neural Networks*, pp. 1967-1970 (1993).
- [4] Sato Y., Hatano S., Abe Y., Ochiai T., and Oka R. : "Development of a High-performance Simulator for Cooperative Clustered Recurrent Neural Networks", *Proc. of the Second Inter. Conf. on Neural Information Processing*, pp. 637-641 (1995).
- [5] Sato Y., Nagaya S. : "Evolutionary Algorithms that Generate Recurrent Neural Networks for Learning Chaos Dynamics", *Proc. of the Third IEEE Inter. Conf. on Evolutionary Computation*, pp. 144-149 (1996).
- [6] Sato Y., Kuwabara Y., and Oka R. : "Real-Time Learning of Chaos Dynamics Using a High-Performance Simulator on a Neuro-Computer", *Proc. of the Third Inter. Conf. on Neural Information Processing*, pp. 1262-1268 (1996). (招待講演)

第6章：進化的計算を用いた韻律係数フィッティングによる声質変換に関して

- [1] 佐藤裕二, 安藤ハル, 北原義典："進化的計算を用いた韻律係数フィッティングによる声質変換に関して", 第12回ヒューマンインタフェースシンポジウム論文集, pp. 469-475 (1996).
- [2] Sato Y. : "Voice Conversion Using Evolutionary Computation of Prosodic Control", *Proc. of the Australasia-Pacific Forum on Intelligent Processing and Manufacturing of Materials* (1997). (掲載予定, 招待講演)

第7章：結論

