

博士学位論文

逆想起を用いた形態学的連想記憶モデルに関
する研究

原田 秀隆

目次

第 1 章	序論	1
第 2 章	連想記憶モデル	5
2.1	概要	5
2.2	連想記憶モデル	5
2.2.1	ニューラルネットワーク	5
2.2.2	ニューラルネットワークを用いた連想記憶	7
2.2.3	連想記憶モデルの歴史	7
2.2.4	ホップフィールドネットワーク	9
2.3	ホップフィールドネットワークによる連想記憶	9
2.4	形態学的連想記憶モデル(Morphological Associative Memories : MAM)	12
2.4.1	数理形態学について	12
2.4.2	MAM の数学的基礎	14
2.4.3	MAM におけるノイズの種類	15
2.4.4	MAM における記憶配列 W 、 M の性質	16
2.4.5	核パターンを用いた形態学的連想記憶モデル	16
第 3 章	記銘パターンから独立した核パターンを用いた MAM	19
3.1	概要	19
3.2	記銘パターンから独立した核パターンを用いた MAM	19
3.3	自己相関想起実験による従来モデルとの比較	21
3.4	記銘パターンから独立した核パターンを用いた MAM のハードウェア化	23
3.5	処理の並列化	23
3.6	記憶配列 W 、 M の各要素の 1bit 表現	24
3.7	MAM ハードウェアの仕様	27
3.8	MAM ハードウェアの構成	27
3.8.1	MAM ハードウェアの全体ブロック	27
3.8.2	記銘ユニット	27
3.8.3	前段想起ユニット	28
3.8.4	多数決ユニット	30
3.8.5	後段想起ユニット	31
3.9	MAM ハードウェアの性能評価	32
3.10	記銘パターンから独立した核パターンを用いた MAM の問題点	34
第 4 章	逆想起を用いた MAM	35

4.1	概要	35
4.2	逆想起を用いた MAM のアルゴリズム	36
4.3	相互相関想起実験に対する逆想起の効果	37
4.4	逆想起の演算の簡単化	41
4.5	相互相関想起実験に対する簡単化した逆想起の効果	43
4.6	簡単化した逆想起を用いた MAM のハードウェアモデル	45
4.7	ハードウェアとソフトウェアモデルとの速度比較	48
第 5 章	分割記録により計算量を抑えた逆想起を用いた MAM	50
5.1	概要	50
5.2	分割記録を用いた MAM	51
5.3	分割記録を用いた MAM の評価	53
5.3.1	完全想起率の記録パターン数依存性	53
5.3.2	記録パターンから独立した核パターン構成法の導入	55
5.4	逆想起手法の導入	59
第 6 章	結論	61
	謝辞	63
	参考文献	64

図目次

図 1.1	不完全な情報からの想起(例)	1
図 2.1	形式ニューロンモデル	6
図 2.2	ホップフィールドネットワーク連想記憶のエネルギー関数	11
図 2.3	図形と構造要素、dilation と erosion の演算結果	13
図 2.4	記銘パターンとノイズの例	15
図 2.5	記銘パターンと核パターン(アルファベット大文字 20 文字)	18
図 2.6	核パターンを用いない MAM と核パターンを用いた MAM における自己相関 想起実験のノイズ耐性	18
図 3.1	記銘パターンから独立した核パターンを用いた MAM	20
図 3.2	記銘パターンから独立した核パターンを用いた MAM における自己相関想起実 験のノイズ耐性	21
図 3.3	記銘パターン(アルファベット大文字 26 文字)	22
図 3.4	記銘パターン(アルファベット小文字 26 文字)	22
図 3.5	我々のモデルにおける相互相関想起実験のノイズ耐性	23
図 3.6	記憶配列 M の算出例	25
図 3.7	記憶配列 W の算出例	25
図 3.8	記憶配列 M の各要素の 1bit 表現例	26
図 3.9	記憶配列 W の各要素の 1bit 表現例	26
図 3.10	MAM ハードウェアの全体ブロック図	27
図 3.11	記銘ユニット内部ブロック図	28
図 3.12	記憶配列 M 、 W 更新ユニット内部ブロック図	28
図 3.13	前段想起ユニット内部ブロック図	29
図 3.14	前段想起演算ユニット内部ブロック図	29
図 3.15	多数決ユニット内部ブロック図	30
図 3.16	後段想起ユニット内部ブロック図	31
図 3.17	8×8 の記銘パターン(アルファベット大文字 20 文字)	32
図 3.18	ハードウェアシミュレーション結果	33
図 3.19	包含関係にあるパターン	34
図 4.1	複数の核パターンが想起される様子	36
図 4.2	逆想起を用いた MAM の処理の流れ	37
図 4.3	2 個の包含関係にあるパターン 10 組	38
図 4.4	3 個の包含関係にあるパターン 10 組	38

図 4.5 従来モデルにおける相互相関想起実験のノイズ耐性(2 個の包含関係にあるパターン 10 組の記録の場合).....	39
図 4.6 逆想起を用いた MAM における相互相関実験のノイズ耐性(2 個の包含関係にあるパターン 10 組を記録の場合).....	39
図 4.7 従来モデルにおける相互相関想起実験のノイズ耐性(3 個の包含関係にあるパターン 10 組を記録した場合).....	40
図 4.8 逆想起を用いた MAM における相互相関想起実験のノイズ耐性(3 個の包含関係にあるパターン 10 組を記録した場合).....	40
図 4.9 単純化した逆想起を用いた MAM の逆想起過程.....	42
図 4.10 逆想起を用いた MAM における相互相関想起実験のノイズ耐性.....	43
図 4.11 単純化した逆想起を用いた MAM における相互相関想起実験のノイズ耐性.....	44
図 4.12 単純化した逆想起を用いた MAM ハードウェアモデル.....	45
図 4.13 逆想起ユニットのブロック図.....	46
図 4.14 相互相関想起実験に用いた記録パターン対.....	47
図 4.15 逆想起を用いた MAM ハードウェアのハードウェアシミュレーション結果.....	48
図 5.1 記憶配列のユニット数(100 ユニットの 2 値パターンの例).....	50
図 5.2 2 値、5 ユニット 3 組の自己相関想起における記憶配列 M.....	51
図 5.3 分割記録を用いた際の複数の記憶配列の構成.....	52
図 5.4 自己相関想起における分割想起.....	52
図 5.5 完全想起率の記録パターン数依存性(Ritter のモデル).....	53
図 5.6 完全想起率の記録パターン数依存性(分割記録を用いたモデル).....	54
図 5.7 記録パターン数を 1000 個とした場合のノイズ耐性.....	55
図 5.8 分割記録を用いていないモデルにおける完全想起率の記録パターン数依存性.....	56
図 5.9 分割記録を用いたモデルにおける完全想起率の記録パターン数依存性.....	57
図 5.10 記録パターン数 1000 個の場合の独立した核パターンを用いたモデル、分割記録を用いたモデルの完全想起率.....	58
図 5.11 独立した核パターンを用いたモデルおよび分割記録を用いたモデルにおける相互相関想起を行った場合の完全想起率の記録パターン数依存性.....	59

表目次

表 3-1 MAM ハードウェアモデルとソフトウェアの比較	34
表 4-1 単純化した逆想起を用いた MAM と従来の逆想起との一回の想起時間比較 .	44
表 4-2 逆想起を用いた MAM ハードウェアモデルとソフトウェアとの比較	49

第1章 序論

普段生活している際、人はある物を見たとき、「対象物の一部が隠れていても、その対象物の全体を想起」、または「対象物に関連する別の物を想起する」、等の柔軟な想起を行うことができる。この機能は“連想”と呼ばれ、重要な脳機能の一つである。この連想は、人間にとっては簡単に処理することが可能であるが、現代のコンピュータには難しい情報処理の一つである。

連想記憶において、入力データから記憶したものを想起する際の検索する方法によって、大きく2種類に分けることができる。

1. 完全一致検索型

入力データと完全に一致するデータを記憶から検索し、想起する

2. 最少距離検索型

入力データと最も類似したデータを記憶から検索し、想起する

1.の連想記憶は Content Addressable Memory(CAM)であり、CAM は、ルーターの IP テーブルなどに応用されている。2.の連想記憶は、入力データから最も類似したデータを想起するため、不完全な情報から完全な情報(図 1.1)、あるいは関連する情報を想起するような、人間により近い連想処理であると、我々は考えており、本研究では 2.の連想記憶に着目している。この連想記憶システムを実現することで、パターン認識、画像や音声認識、データ修復、人間のような柔軟な情報処理などへの応用が期待される。

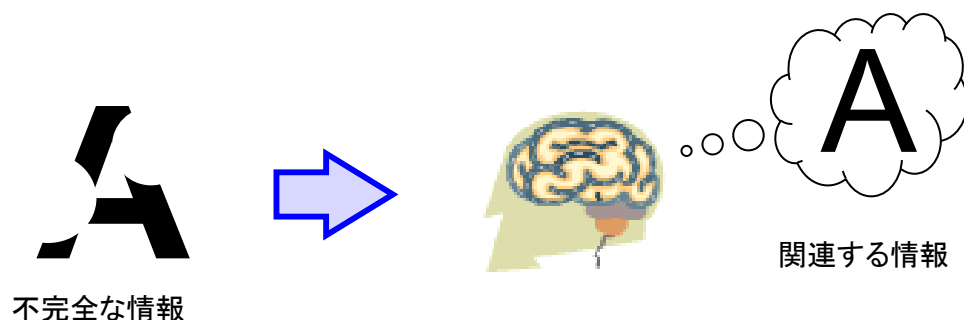


図 1.1 不完全な情報からの想起(例)

これまでに多くの連想記憶モデルが研究されているが、記憶したパターンとは違う“偽記憶”を想起してしまうこと（ローカルミニマム問題）や、デジタルメモリと比較し、記憶容量の小ささ等の問題点が指摘されている[1][2][3]。連想記憶モデルの一つとして、Ritter

らが、数理形態学の論理で記銘、想起を行う、形態学的連想記憶モデル(Morphological Associative Memory : MAM)を提案した[4]。その後、この MAM モデルについて、さまざまな研究がなされている[[5]-[20]]。Chen と Liu は Complex MAM(CMAM)を提案している。CMAM は、FFT を行った複素データのような複素信号を直接処理することができる。2003 年には、形態学とファジー演算を基礎とした、Fuzzy MAM(FMAM)が、Wang らによって提案された。Wang と Chen は FMAM に焦点をあて、改良を行った Enhanced FMAM(EFMAM)を提案している。MAM の研究は多くなされているが、後述するように、従来の MAM は、Ritter の MAM をベースとしているため、包含関係にあるパターンには対応できない。

MAM モデルは、Hopfield のような従来の連想記憶モデルと比較して、収束するまで繰り返し計算する必要がなく、一回の計算で想起可能である。また、2 値のパターンを扱う場合、その記憶容量は 2^N と、従来モデルと比較し、記憶容量が大きいという性質を持っている。その一方で、想起率の改善のため Ritter が取り入れた核パターンという、パターンを想起するためのインデックスのような概念において、その核パターンの設計が記銘パターン数の増加と共に難しくなる。また、核パターンを構成するユニットにノイズが載ると正しく想起できない。そこで我々は、新しい核パターンの構成法である、記銘パターンに依存しない核パターン構成法を提案し、その構成法を用いた新しい MAM を提案した[21][22]。我々の核パターン構成法では、核パターンを記銘パターンに依存することなく決定できるため、核パターンの設計が容易である。また、ブロック分割の手法を取り入れ、分割したサブブロック内にノイズの影響をととめることによる想起率の改善もなされている。しかしながら、従来の連想記憶モデルと同様、このモデルでは、包含関係にあるパターンを記銘後、そのパターンを入力されると、ノイズの有り無しに関わらず、正しく想起できない。

本論文では、新しい想起方法である、逆想起手法を導入し、包含関係にあるパターンへも対応できる新しい MAM を提案する[23]。記銘パターンから独立した核パターン構成法を用いた MAM において、包含関係にあるパターンを記銘後、包含パターンが入力された際、前段想起にて複数の重なった核パターンが想起される。前段想起にて想起された、その複数の重なった核パターンを用いて、後段想起を行うため、複数の重なった出力パターンが想起される。逆想起手法を導入したモデルでは、記銘したパターンを前段想起とは逆方向に想起させる。その逆想起で得られた記銘パターンと入力とを比較することにより、尤もらしい核パターンを決定し、出力パターンを想起することにより、入力に対応する尤もらしい出力パターンを想起できる。包含しているパターンを、重なった核パターンをそれぞれに分離し、それぞれの核パターンに対応する記銘したパターンを想起し入力との比較が可能のため、複数のパターンが重なったパターンが記銘・入力されても、尤もらしいパターンを想起できる。また、我々が提案した、記銘パターンに依存しない核パターン構成法を用いた MAM の利点も含んでいるため、記銘パターンが増加しても、核パターンを簡単に構成可能である。

MAM はシンプルな演算を用いているため、ハードウェア化に適した連想記憶モデルとなっている。MAM のハードウェアはいくつか提案されている。Guzman らは、画像認識へ応用した MAM ハードウェアベースのプロセッサを提案している[24]。しかしながら、従来の Ritter の MAM をベースとして用いているため、包含関係にあるパターンが入力された場合は対応できない。

逆想起を用いた MAM もまた、非常にシンプルな演算を用いていること、また、並列に処理が可能なアーキテクチャとなっているため、ハードウェア化に適した連想記憶モデルとなっている。我々は、逆想起に用いている演算を、よりシンプルな演算に簡略化することで、よりハードウェア化に適したモデルを提案する[25]。従来の逆想起を用いた MAM では、包含関係にあるパターンが入力された場合、前段想起において、重なった核パターンが想起され、重なったパターンを分離し、記銘パターンを逆想起することにより、入力との比較を行っているが、重なっているパターンの数だけ、記銘パターンを逆想起する必要がある。2 値のパターンを取り扱った場合、この逆想起の演算を簡略することが可能であり、簡略化することで、重なった核パターンを分離することなく、記銘パターンを逆想起することが可能である。そのため、一回の逆想起で記銘パターンを想起でき、その分想起時間を減らすことが可能である。

この簡略化した逆想起を用いた MAM のハードウェアモデルを提案することにより、より高速に想起が可能な MAM ハードウェアを開発する[26][27]。従来の連想記憶では複数のパターンが重なったパターンには対応できなかったが、この逆想起を用いた MAM ハードウェアモデルでは、たとえ複数のパターンが重なったパターンが入力されても、尤もらしいパターンを高速に想起することが可能である。実応用を考えた場合、記銘パターンが増加すると、包含するパターンや、類似パターンの存在も増えていくことが予想される。そのような状況下でも、我々が提案した MAM ハードウェアモデルを応用することにより、高速に尤もらしいパターンを想起可能となる。

従来の MAM モデルでは、パターンを記憶している記憶配列 M 、 W の構成に必要な空間計算量は、記銘パターンのユニット数を n とすると、 $O(n^2)$ となってしまうことがわかっている[4]。そこで、これまでの MAM モデルの問題点を解決した分割記銘を用いた新しい MAM モデルが提案された[28]。このモデルの記憶配列 M 、 W に必要な空間計算量は、 $O(n^{1.5})$ まで削減できる。しかしながら、分割記銘を用いたモデルでは、入力および出力パターン対を分割し、それぞれの分割したパターンを記銘するため、記銘するパターンのユニット数が少なくなり、包含関係にあるパターンが増加する。そのため、逆想起を用いた MAM と比べ、記銘可能なパターン数が減少してしまう。そこで、分割記銘を用いたモデルに、逆想起を導入することで、従来の MAM と比較して空間計算量を削減しながらも、記銘可能なパターン数の減少を抑えることを検討する。

本論文の構成は下記の通りである。

第 1 章は、序論である。

第 2 章では、連想記憶モデルについて述べる。

第 3 章では、記銘パターンから独立した核パターンを用いた **MAM** モデルとそのハードウェアモデルについて述べ、ソフトウェアシミュレーション、及びハードウェアシミュレーションを用いて、性能を評価する。

第 4 章では、逆想起を用いた **MAM** モデルとそのハードウェアモデルについて述べ、ソフトウェアシミュレーション、及びハードウェアシミュレータを用いて、性能を評価する。

第 5 章では、分割記銘を用いた **MAM** モデルによる計算量の削減、および逆想起を導入した分割記銘モデルについて検討する。

第 6 章は、結論である。

第2章 連想記憶モデル

2.1 概要

人の脳機能の中には、一部が欠けているような情報から完全な情報の想起、ある情報からその情報に関連する情報を想起等、連想とよばれる処理がある。この連想という処理は、コンピュータにとっては、不得意とする情報処理の一つである。この連想の処理は、入力されるデータに対応する、記録されたデータの中から最も関連するデータを想起する、連想記憶システムとして考えることができる。

これまで、この連想とよばれる処理を実現するために、人間の神経回路網を模擬したような連想記憶モデルの研究が行われてきた[29][30][31]。初期の連想記憶モデルとしては、中野がアソシアトロン[29]を提案している。また、ほぼ同時期に、Kohonen、Andersonによって、連想記憶モデル[30][31]がそれぞれ提案されている。その連想記憶モデルの中で、最も代表的なモデルとして、ホップフィールドネットワークは広く知られている[32]が、いくつかの問題点も持つ。連想記憶としてホップフィールドネットワークを用いた場合、記憶可能な容量が少なく、記憶していない誤ったパターン(偽記憶)を想起してしまうこともしばしばある[1][2][3]。

連想記憶モデルの一つとして、数理形態学の概念を用いて、パターンの記録および想起を行う、形態学的連想記憶モデル(Morphological Associative Memories : MAM)が、Ritterらにより提案された[4]。このモデルは、Hopfield のような従来の連想記憶モデルと比較して、収束するまで繰り返し計算する必要がなく、一回の計算で想起可能である。また、2 値のパターンを扱う場合、その記憶容量は 2^N と、従来モデルと比較し、記憶容量が大きいという特長を持っている。

本章ではまず、ニューラルネットワークの基本素子となる、ニューロンについて述べる。また、連想記憶モデルの中の代表的なモデルとして、ホップフィールドネットワークについて述べる。次に、本研究の元となっている形態学的連想記憶モデルについて述べる。

2.2 連想記憶モデル

2.2.1 ニューラルネットワーク

人の脳には 140 億個もの神経細胞(ニューロン)があるといわれており、その膨大な数のニューロンが並列かつ非同期的に動作を行う。それぞれのニューロンの動作により、全体的

な動作としてみると、人は複雑な計算や記憶などの機能を同時に行うことができる。

ニューラルネットワークは、脳機能の神経回路網の数理モデルとして研究されている。人の実際のニューロンは複雑な構造もつが、数理モデルとしてのニューラルネットワークにおいては、生体のニューロンを厳密に模擬せず、ニューロンの特定の機能を抽出し、単純化した形式ニューロンモデルが基本素子となる。

1943 年、McCulloch と Pitts によりニューロンモデル[33]が提案された。McCulloch-Pitts ニューロンモデルは、実際のニューロンの特性である、シナプスによる情報修飾、細胞体による電位加算、インパルス発生 of 閾値特性をモデル化している。

McCulloch-Pitts ニューロンモデルを図 2.1 に示す。この場合、このニューロンモデルの出力値は式(2.1)を用いて算出される。

$$y = \begin{cases} 0, & \sum_{i=1}^n w_i x_i \leq \theta \\ 1, & \sum_{i=1}^n w_i x_i > \theta \end{cases} \quad (2.1)$$

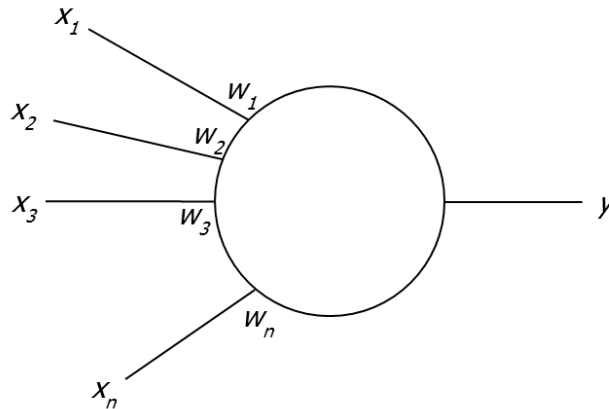


図 2.1 形式ニューロンモデル

ニューラルネットワーク中のある一つのニューロンに着目すると、そのニューロンの入力には、他のニューロンあるいは外部から信号が与えられる。ここで、 x_i は i 番目のシナプス前ニューロンの出力、 w_i をそれらに対するシナプス荷重値、 θ は閾値を表している。シナプス荷重値は任意の実数値をとり、正の場合は興奮性のシナプスに対応する。また、負の場合には抑制性のシナプスに対応し、0の場合は、シナプス結合がないものと考えられる。

$\sum_{i=1}^n w_i x_i$ は、生体ニューロンの内部電圧を意味し、内部電圧が閾値 θ を超えると、インパルスが出力される。

2.2.2 ニューラルネットワークを用いた連想記憶

連想記憶とは、「りんご」という入力から、「赤い」や「甘酸っぱい」を出力したり、顔の一部分を入力して、顔全体を出力したりする情報処理のことを指す。このとき、記憶させる入力パターンと出力パターンのパターン対により、2 パターンの記憶方法がある。1 つは、入力パターンと出力パターンが同じである場合、この連想記憶のことを自己相関連想記憶と呼ぶ。もう一方は、入力パターンと出力パターンが異なる場合、この連想記憶のことを相互相関連想記憶と呼ぶ。自己相関連想記憶では、入力パターンと出力パターンが同じなため、単純に複数のパターンを記憶する。相互相関連想記憶では、入力パターンと出力パターンの異なるパターン対で、複数のパターンを記憶する。この方法は通常の計算機と異なり、記憶されたものと同じような情報を入力することによって、記憶した情報の検索がおこなわれる。

ニューラルネットワークにおける連想記憶モデルの特徴としては、以下が挙げられる。

1. 分散多重記憶

入出力パターン対を記憶するとき、その情報はニューラルネットのシナプス全体に分散して記憶される。複数の入出力パターン対を記憶する際には、それぞれの入出力パターン対の情報はシナプス全体に重なって記憶される。これにより、ニューラルネットの一部分が失われても、ある入出力パターン対が全て記憶から失われることはない。

2. 並列的な想起方法

記憶するパターン対の数が増加しても、ある 1 つのパターンが入力され、それに対応する出力パターンを想起するために、各ニューロンの動作が増加することはない。

3. 不完全な入力からの想起

ニューロンに閾値作用を持たせると、不完全な入力パターンが入力された場合、正しい出力パターンを想起可能な能力(誤り訂正能力)を持つ。例として、ニューラルネットに $(\mathbf{x}^{(\xi)}, \mathbf{y}^{(\xi)})$ というパターン対を記憶したと仮定する。ここで、 ξ はパターン番号を表す。記憶したパターン $\mathbf{x}^{(\xi)}$ にノイズを付加したパターン $\mathbf{x}'^{(\xi)}$ を、ニューラルネットワークに入力した場合、ニューラルネットワークは正しい出力 $\mathbf{y}^{(\xi)}$ を出力する。

2.2.3 連想記憶モデルの歴史

1972 年に中野、kohonen、Anderson らがそれぞれ、同時期に連想記憶モデルを提案した[29][30][31]。これらの提案された連想記憶モデルは相関行列を用いており、後に提案さ

れる連想記憶モデルのベースとなっている。

1982 年、物理学者である J.J.Hopfield は、ニューラルネットワークモデルの一つとして、エネルギーの概念を導入したホップフィールドネットワークを提案した[32]。ホップフィールドネットワークは、連想記憶にも応用され、最も代表的な連想記憶モデルとして広く用いられている。連想記憶モデルとして以外では、このニューラルネットワークは巡回セールスマン問題のような最適化問題にも応用されている[34]。

1983 年、平井は、従来の連想個億モデルでは困難であった 1 対多の連想が可能な **human associative processor (HASP)** を提案した[35]。HASP は、2 つの連想記回路で構成されており、相関行列を用いた相互想起型連想記憶回路、相互抑制回路から成る。キーおよび連合ベクトルがスパースである場合、従来の連想記憶モデルと比較して、HASP は 2 倍以上の記憶容量をもつことが示されている[36]。[37]では、HASP における、最も簡単な相関型連想記憶モデルの提案、そのモデルの想起過程の議論がなされている。

1985 年、Ackley らは、ホップフィールドネットワークをベースにした、ボルツマンマシンを提案した[38]。ホップフィールドネットワークでは、最急降下法により、エネルギーが最小となるようネットワークの状態を変化させるが、一般に学習を行わないため、ローカルミニマム(局所最小解)に収束する場合がある。ボルツマンマシンでは、ホップフィールドネットワークに新しく、パラメータ温度 T を取り入れ、確率的に状態変化の動作繰り返し、平衡を保つことが可能である。また、各ユニットは 0 または 1 を出力するが、0、1 どちらを出力するかが決定論的に決まるのではなく、エネルギーが変化したときの差分および温度 T の 2 つのパラメータによって確率を求め、各ユニットの出力を変化させている。ボルツマンマシンでは、パラメータである温度 T が各ユニットの出力に大きな影響を与える。ボルツマンマシンの欠点としてはいかが挙げられる。

(1)学習に要する時間が多い

(2)構造的に対称結合ネットワークに制限

この欠点に対し、平均場近似法[39][40][41]、ヘルムホルツマシン[42]等が提案されている。

1987 年には、Kosko より 2 層構造を持つ連想記憶モデルとして、双方向連想記憶 (**BAM: Bidirectional Associative Memory**) が提案された[43][44]。BAM の記憶容量としては、 $N/4 \log N$ であることが導かれている(N はユニット素子数)[44]。

1990 年には、萩原が、2 層構造である BAM を多数の層に拡張を行った、多方向連想記憶 (**Multidirectional Associative Memory**) を提案した[45]。服部は擬似緩和学習法を提案することで、多方向連想記憶モデルの記憶容量を大きくしている[46]。また、小林らによって多方向連想記憶モデルの記憶容量の議論がなされている[47]。さらに、小林は多方向連想記憶に隠れ層を追加したモデルとその学習法を提案した[48][49]。

1990 年、森田らはそれぞれの神経素子の出力が内部状態の非単調関数によって出力される神経回路網、非単調神経回路網を用いた連想記憶モデルを提案した[50]。非単調神経回路網は、非単調入出力特性(神経素子への入力増加に対して出力が単調に増加せず、入力の絶

対値が大きいときには出力が減少するという性質)を持つ神経素子から構成される回路網である。非単調神経回路網は、従来の神経回路網にない特長をもっており、連想記憶の能力が非常に優れている。従来の連想記憶モデルの欠点であった、記憶容量の小ささ、低い連想能力、偽記憶および情報表現上の制約が改善されている[51][52]。記憶容量では、理論的な解析によって、従来のモデルの3倍以上の容量があること示させている[53][54]。

また、合原らがカオスニューロンを提案した[55][56]。このカオスニューロンを導入し、その動的遷移能力により、1対多の想起が可能なカオス連想メモリが提案された[57][58][59]。

その他の連想記憶モデルとして、従来モデルと比較して記憶容量が大きな積結合を有する連想記憶モデル[60][61][62]、忘却の構造を持つ連想記憶モデル[63][64]、階層型ネットワーク構造の連想記憶[65]、スパースコーディング[66][67][68]など、様々な連想記憶モデルが研究されている。

2.2.4 ホップフィールドネットワーク

1982年に物理学者 J.J.Hopfield は、ホップフィールドネットワークを提案した。Hopfield は、相互結合型ニューラルネットワークモデルでは、物理学でのスピン・モデルと類似していると指摘している[69]。スピン・モデルでは、「上向き」と「下向き」の2つの状態を持つ、磁気的基本的単位である。物質中の多数のスピン間の相互作用により、物質の磁気的性質が発生していると考えられている。

Hopfield は、スピンをニューロンに、スピン間の相互作用をニューロン間の結合に置き換え、相互結合型のニューラルネットワークモデルが得られると考えた。スピンの上向きの場合は、ニューロンが興奮している、下向きの場合は、ニューロンが興奮していないことに対応している。

さらに Hopfield は、このニューラルネットワークモデルに、「エネルギー」という量に対応した、「ニューラルネットワークのエネルギー」という概念を取り入れている。Hopfield は、この「エネルギー」という量を使って、ニューラルネットワークの動作を調べ、ニューラルネットワーク全体がどのように振る舞い、どのような情報処理能力を持っているのかを示している。

2.3 ホップフィールドネットワークによる連想記憶

神経回路網による連想記憶では、不完全なパターンに対応する状態を神経回路網の初期状態として与え、最終的に平衡状態に対応したパターンが、記録したパターンの想起結果

である。連想記憶の記録とは、記録したパターンに対応した状態が平衡状態になるよう、神経回路網を動作させることである。この節では、離散モデルのホップフィールドネットワークを用いた連想記憶について述べる。

記録するパターンを、-1、1 の 2 値の要素をもつ N 次元ベクトルで表す。ここで N 個の -1、1 の 2 値をとるニューロンからなる離散モデルの神経回路網を考える。神経回路網のそれぞれのニューロンを、パターンベクトルの各要素と考え、神経回路網の状態と記録パターンは、1 対 1 に対応する。 p 個のパターンベクトル

$$X^{(1)}, X^{(2)}, \dots, X^{(p)}, \quad (2.2)$$

を記録することを考える。最も簡単な記録方法は、シナプス荷重を式(2.3)で算出することである。

$$W_{ik} = \sum_{p=1}^P X_i^{(p)} X_k^{(p)}, \quad (2.3)$$

この場合、式のエネルギー関数は式(2.4)で示すことができる。

$$\begin{aligned} E(x) &= -\frac{1}{2} \sum_{i=1}^N \sum_{k=1}^N W_{ik} X_i X_k \\ &= -\frac{1}{2} \sum_{i=1}^N \sum_{k=1}^N \sum_{p=1}^P X_i^{(p)} X_k^{(p)} X_i X_k \\ &= -\frac{1}{2} \sum_{p=1}^P \sum_{i=1}^N X_i^{(p)} X_i \sum_{k=1}^N X_k^{(p)} X_k \\ &= -\frac{1}{2} \sum_{p=1}^P \left(\sum_{i=1}^N X_i^{(p)} X_i \right)^2 \\ &= \sum_{p=1}^P E^{(p)}(X) \end{aligned} \quad (2.4)$$

ここで

$$E^{(p)}(X) = -\frac{1}{2} \left(\sum_{i=1}^N X_i^{(p)} X_i \right)^2 \quad (2.5)$$

式(2.5)は、エネルギー関数 $E(X)$ での、パターン $X^{(p)}$ のエネルギーである。 X_i は、-1 または 1 をとるため、 $X=X^{(p)}$ の場合に $E^{(p)}(X)$ は最小値をとる。したがって、それぞれのパターンのエネルギーを足し合わせた $E(X)$ は、 $X=X^{(p)}$ ($p=1, \dots, P$) の場合に極小点を持つことが近似

的に成り立つ(図 2.2)。しかしながら、 $X=X^{(p)}$ のパターン数 p の増加や、それぞれのパターン $X^{(p)}$ の分布により、式(2.4)は成り立たなくなる場合がある。この場合、正しいパターンを想起することができない。また、式(2.5)の $E^{(p)}(X)$ は、 $X^{(p)}$ の -1 と 1 を反転したパターン $-X^{(p)}$ の場合でも、 $X^{(p)}$ と同じ最小値をとるため、反転したパターンも同時に記録してしまう。

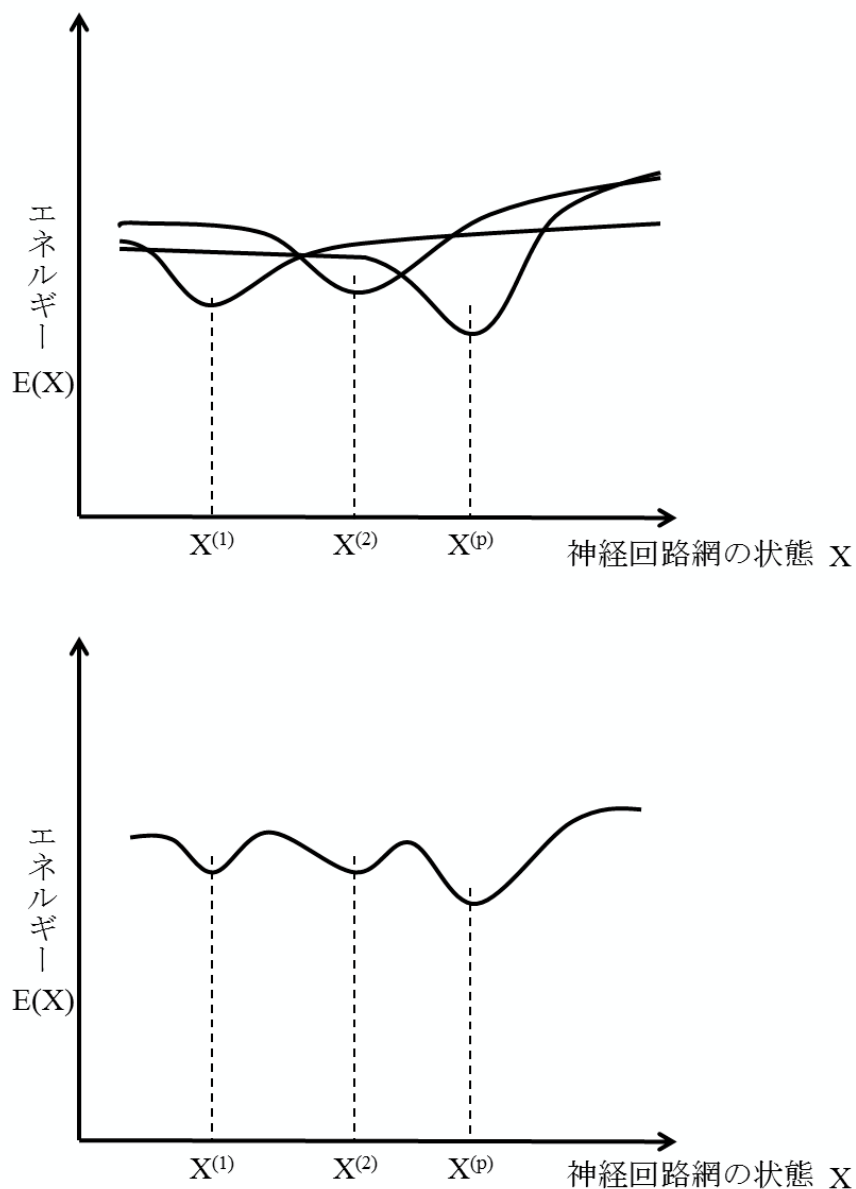


図 2.2 ホップフィールドネットワーク連想記憶のエネルギー関数

2.4 形態学的連想記憶モデル(Morphological Associative Memories : MAM)

2.4.1 数理形態学について

画像に対する非線形解析手法の一つとして、数理形態学(Mathematical Morphology : モルフォロジー)がある[70][71]。数理形態学では、典型的形状の小図形である「構造要素」を用いて、画像中にある物体の「形、大きさ」を、その対比で定量的に表現する。構造要素を用いて、画像中の図形全体を「膨張」(dilation)「収縮」(erosion)と呼ばれる基本的集合演算、およびそれらを組み合わせた opening、closing により、さまざまな画像演算を構成することが可能である。

dilation : 対象図形を構成要素の分だけ膨張

$$A \oplus B = \{x \in E^N \mid x = a + b, a^{\vee} \in A, b^{\vee} \in B\} \quad (2.6)$$

erosion : 対象図形を構成要素の分だけ収縮

$$A \ominus B = \{x \in E^N \mid x + b \in A, b^{\vee} \in B\} \quad (2.7)$$

opening : 対象図形中の構成要素より小さな凸部を削る

$$AB = (A \ominus B) \oplus B \quad (2.8)$$

closing : 対象図形中の構成要素より小さな凹部を削り、孔を潰す

$$AB = (A \oplus B) \ominus B \quad (2.9)$$

A、B は、対象画像と構成要素の座標を表す位置ベクトル a 、 b の集合となる。

図 2.3 に対象図形と構造要素、および dilation と erosion の演算を行った際の結果を示す。構造要素により、さまざまな画像処理が可能である。

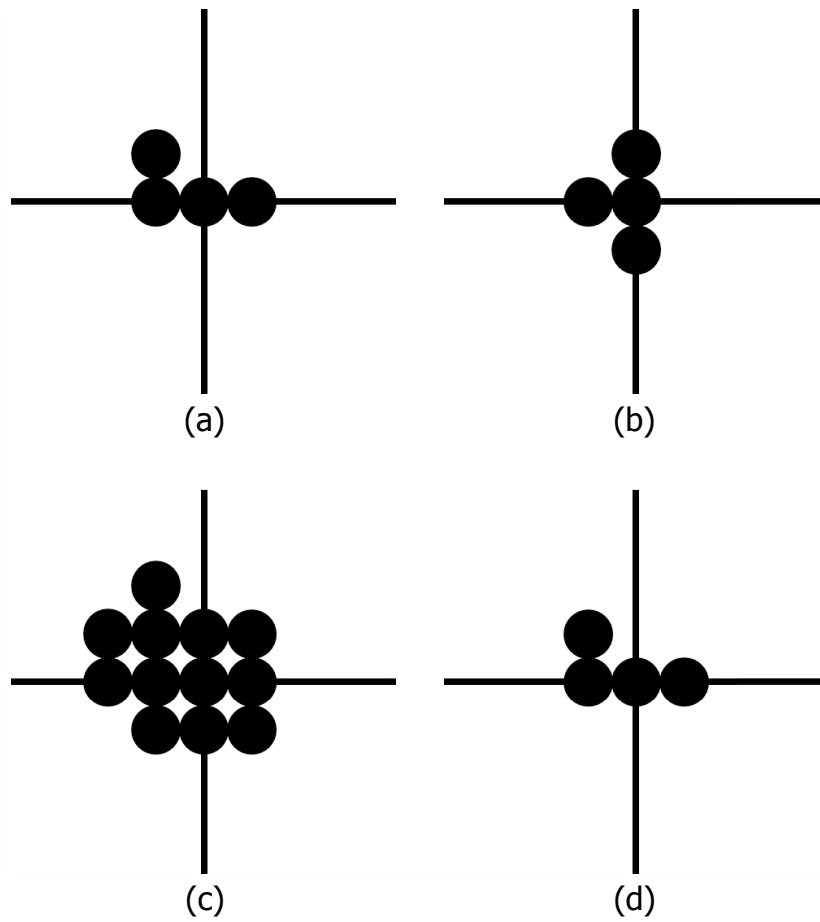


図 2.3 図形と構造要素、dilation と erosion の演算結果
 (a)対象図形、(b)構造要素、(c)dilation、(d)erosion

2.4.2 MAM の数学的基礎

記録させるパターンが P 組あるとすると、 $(X^1, Y^1), \dots, (X^P, Y^P)$ のベクトル対を、連想記憶モデルに記録されることになる。あるベクトル対 $X = (x_1, \dots, x_n)$ 、 $Y = (y_1, \dots, y_m)$ のパターンに対して、MAM では

$$W_{XY} = Y \wedge (-X)', \quad (2.10)$$

$$W_{XY} = \begin{pmatrix} y_1 - x_1 & \cdots & y_1 - x_n \\ \vdots & \ddots & \vdots \\ y_m - x_1 & \cdots & y_m - x_n \end{pmatrix} \quad (2.11)$$

とすると

$$W_{XY} \vee X = \begin{pmatrix} \bigvee_{i=1}^n (y_1 - x_i + x_i) \\ \vdots \\ \bigvee_{i=1}^n (y_m - x_i + x_i) \end{pmatrix} = Y \quad (2.12)$$

が成立する。ここで、 \vee 、 \wedge の演算子は \max 演算、 \min 演算となる。

2次元配列 A、B、C、D において、 $C = A \vee B$ が成り立つとする。行列 C の ij 成分 c_{ij} は

$$\begin{aligned} c_{ij} &= \bigvee_{k=1}^P (a_{ik} + b_{kj}) \\ &= (a_{i1} + b_{1j}) \vee (a_{i2} + b_{2j}) \vee \cdots \vee (a_{iP} + b_{Pj}), \end{aligned} \quad (2.13)$$

となる。

同様に $D = A \wedge B$ とすると、行列 D の ij 成分 d_{ij} は

$$\begin{aligned} d_{ij} &= \bigwedge_{k=1}^P (a_{ik} + b_{kj}) \\ &= (a_{i1} + b_{1j}) \wedge (a_{i2} + b_{2j}) \wedge \cdots \wedge (a_{iP} + b_{Pj}), \end{aligned} \quad (2.14)$$

となる。

2.4.3 MAM におけるノイズの種類

本論文で扱うパターンは、“1”、“0”の2値のパターンを取り扱い、黒が“1”を、白が“0”を表している。本論文でのノイズとは、記銘パターンのある要素をビット反転することを意味する。図 2.4 に記銘パターン、およびノイズを加えたパターンの例を示している。ノイズの種類として、付加ノイズ、欠落ノイズ、一般ノイズがある。付加ノイズでは、記銘パターンの白(“0”)のある要素を、黒(“1”)にビット反転させる(図 2.4(b))。欠落ノイズでは、記銘パターンの黒(“1”)のある要素を、白(“0”)にビット反転させる(図 2.4(c))。一般ノイズでは、付加、欠落ノイズの両方のノイズを加える(図 2.4(d))。本研究では、一般ノイズを取り扱う。

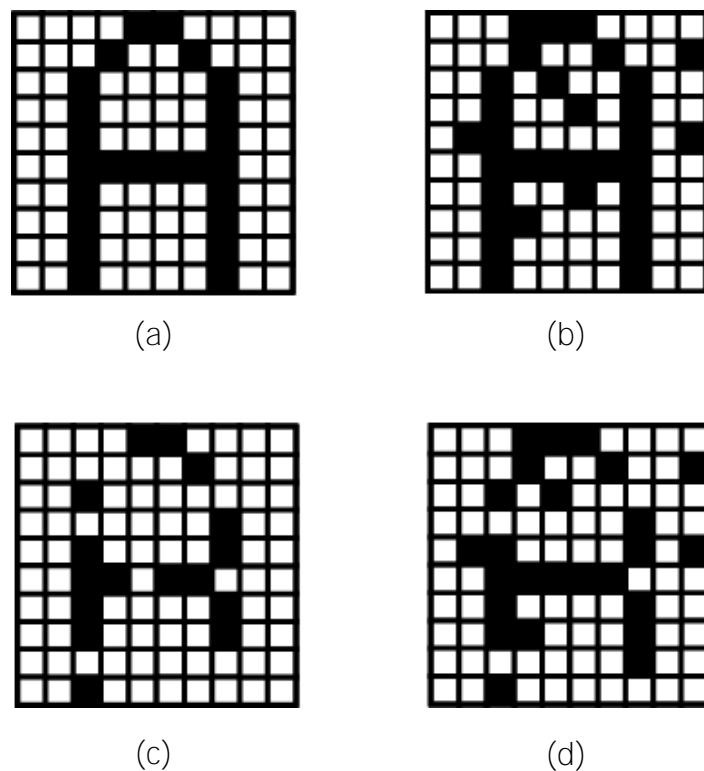


図 2.4 記銘パターンとノイズの例
(a)記銘パターン、(b)付加ノイズを加えたパターン、
(c)欠落ノイズを加えたパターン、(d)一般ノイズを加えたパターン

2.4.4 MAM における記憶配列 W 、 M の性質

形態学的連想記憶モデル(MAM)では2つの重みである記憶配列 W と M を用いて、加算、最大値および、最小値の演算を行い、想起させる。

ここで、 p 個のベクトル対 $X^r = (x_1^r, \dots, x_n^r)$, $Y^r = (y_1^r, \dots, y_m^r)$, $r=1, \dots, p$ を MAM に記録させる場合を考える。このとき、記録した情報は記憶配列 W と M に蓄えられる。

$$w_{ij} = \bigwedge_{r=1}^p (y_i^r - x_j^r) \quad i=1, \dots, m, \quad j=1, \dots, n \quad (2.15)$$

$$m_{ij} = \bigvee_{r=1}^p (y_i^r - x_j^r) \quad i=1, \dots, m, \quad j=1, \dots, n \quad (2.16)$$

w_{ij} 、 m_{ij} はそれぞれ記憶配列 W 、 M の i 行 j 列の要素を表している。また、このように記録を行った MAM において、入力 X が与えられたとき、想起は式(2.17)、(2.18)に従う。

$$y_i^r = \bigvee_{j=1}^n (w_{ij} + x_j^r) \quad i=1, \dots, m \quad (2.17)$$

$$y_i^r = \bigwedge_{j=1}^n (m_{ij} + x_j^r) \quad i=1, \dots, m \quad (2.18)$$

記憶配列 W と M は、それぞれ欠落ノイズ、付加ノイズのみに対して有効であるという性質を持っている[4]。しかし、記憶配列 W は付加ノイズに対して、 M は欠落ノイズに対して全く無力で、わずかのノイズに対しても正しい想起は行えない。また、一般ノイズに対しても、記憶配列 W 、 M は正しい想起を行うことができない。

2.4.5 核パターンを用いた形態学的連想記憶モデル

記憶配列 M および W はそれぞれ入力パターンに含まれる付加ノイズ、または欠落ノイズには有効であるが、欠落ノイズ、または付加ノイズに対しては無力となる。また、付加ノイズ、欠落ノイズの両方が含まれる一般ノイズに対しては、正しいパターンを想起できない。そこで Ritter は、核パターンという概念を導入している。記憶配列 M が付加ノイズに、 W が欠落ノイズに対して有効であるという性質を利用し、2段階の想起を行い、中間に核パターンを用いる。核パターン“Z”は、パターン対を想起するための index のようなもので、記録パターン“X”の一部で構成される。核パターンを構成する際は、他の核パターンと完全に重ならないことが必要である。Ritter の核パターン構成法では、核パターンを式(2.19)に

より求める。また、式(2.19)で求められない場合、式(2.20)を満たす Z^ξ を試行錯誤によって求める。

$$\begin{cases} Z^\xi \leq X^\xi \\ Z^\xi \wedge X^\gamma = 0, \quad \gamma \neq \xi, \end{cases} \quad (2.19)$$

$$\begin{cases} M_{ZZ} \wedge X^\xi = Z^\xi \\ W_{ZX} \vee Z^\xi = X^\xi \end{cases} \quad (2.20)$$

核パターンを用いた MAM では、核パターン、および記憶行列 M と W を用いて次式で表される 2 段階の想起を行うよう構成される。

$$z_i^r = \bigwedge_{j=1}^N (m_{ij} + \tilde{x}_j^r), \quad (2.21)$$

$$y_i^r = \bigvee_{j=1}^M (w_{ij} + z_j^r) \quad (2.22)$$

核パターンの有効性を示すために、図 2.5 に示す記銘パターンを用いて、核を用いない MAM と核パターンを用いる MAM の自己相関想起実験を行った。ここで完全想起率とは、ノイズを含む入力パターンより、対応する記銘パターンを 1 ユニットも間違えずに想起した確率を表す。記銘パターンは $10 \times 10 = 100$ 個の 2 値のユニットからなる。図 2.5 に示すパターン中の赤いユニットは、Ritter の核パターン構成法によって求めた、核パターンである。核パターンを用いない MAM と核パターン用いた MAM における自己相関想起実験のノイズ耐性を、図 2.6 に示す。ここで、横軸はノイズの混入率、縦軸は完全想起率を表している。結果より、核パターンを用いない MAM と比較して、核パターンを用いた MAM の完全想起率が高いことがわかる。

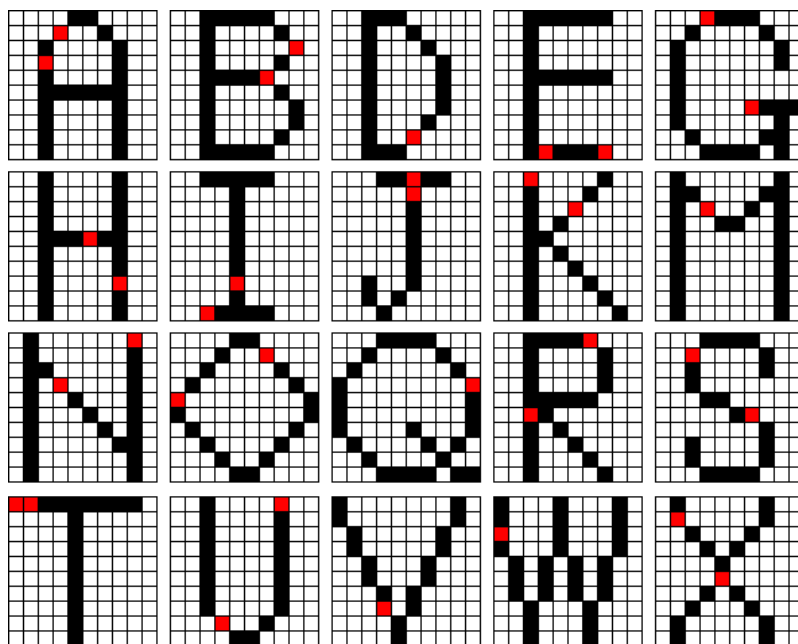


図 2.5 記銘パターンと核パターン(アルファベット大文字 20 文字)
図中の赤いユニットは核パターンを表す

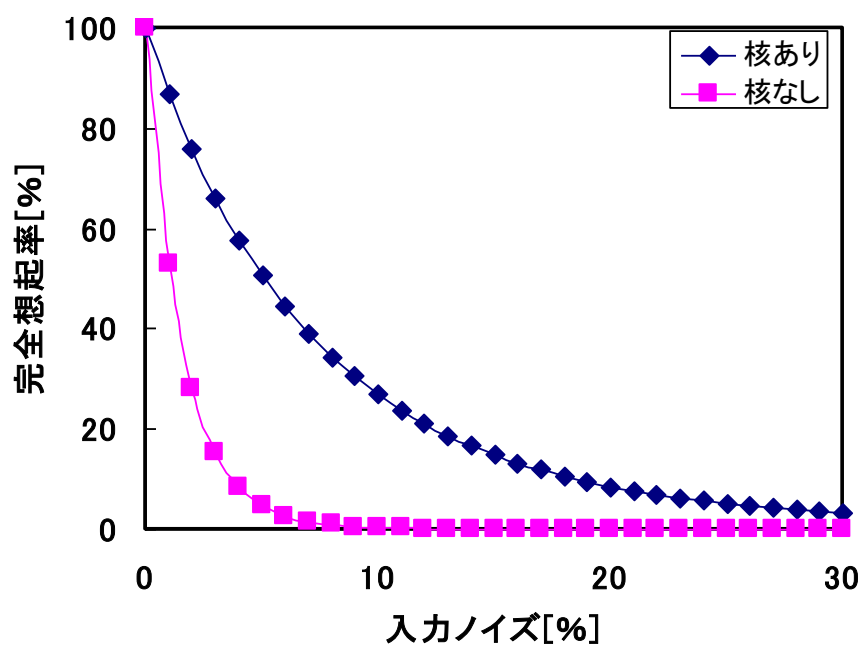


図 2.6 核パターンを用いない MAM と核パターンを用いた MAM における自己相関想起
実験のノイズ耐性

第3章 記銘パターンから独立した核パターンを用いた MAM

3.1 概要

Ritter のモデルの記銘パターンの一部で核パターンを構成する方法は、記銘パターン数が多くなると核パターン設計が難しくなる。また、核パターンに相当するユニットにノイズが載ると想起できなくなる。そこで我々は、この MAM を改良し、記銘パターンから独立した核パターンを用いる MAM を提案した[21][22]。

本章では、記銘パターンに依存しない核パターンの構成法を示し、自己相関想起実験を行いその有効性を確認する。また、その MAM のハードウェアモデルを提案する。

3.2 記銘パターンから独立した核パターンを用いた MAM

記銘パターンから独立した核パターンを用いた MAM の特徴として、

1. 核パターンを記銘パターンに依存せず構成可能である。
2. 記銘パターン対に包含関係がない場合、入力パターンは必ず、記銘パターンの一つに分類可能である。

が挙げられる。特徴 1 として、記銘されたパターンの中に核パターンが存在しないため、想起率のパターン依存性が低く、ノイズに弱い特定のユニットが存在しない。また、核パターンが自由に決定できるため、核パターン設計が容易である。特徴 2 として、多数決を行うことにより、複数の核の中から、必ず一つの核が決定される。

記銘パターンから独立した核パターンを用いた MAM モデルの想起過程を以下に示す。

1. 入力パターンを複数のサブブロックに等分割する。
2. 各々のサブブロックで独立に前段想起を行う。
3. 全てのサブブロックの結果を足しこむ。
4. 足しこまれた核パターンで多数決を行い、核パターンを決定する。
5. 最後に、核パターンを入力として、後段想起において出力パターンが想起される。

図 3.1 に本モデルの処理の流れを示す。

前段の想起における sb 番目のサブブロックの i 番目のユニットは式(3.1)で与えられる。

$$\tilde{y}_i^{r, sb} = \bigwedge_{j=1}^k (m_{ij}^{sb} + \tilde{x}_j^{r, sb}), \quad (3.1)$$

ここで k は各サブブロックのユニット総数、 $\tilde{x}_j^{r, sb}$ は入力パターン x^r の sb 番目のサブブロックの j 番目のユニットである。

前段の想起の後の足し込みと多数決は式(2.24)で表される。

$$\tilde{y}_j^r = \begin{cases} 1 & , \sum_{sb=1}^s \tilde{y}_j^{r, sb} = \bigvee_{i=1}^h \left(\sum_{sb=1}^s \tilde{y}_i^{r, sb} \right) \\ 0 & , \sum_{sb=1}^s \tilde{y}_j^{r, sb} < \bigvee_{i=1}^h \left(\sum_{sb=1}^s \tilde{y}_i^{r, sb} \right) \end{cases}, \quad (3.2)$$

ここで h は核パターンの総ユニット数を、 $\tilde{y}_i^{r, sb}$ は入力パターン $\tilde{x}_j^{r, sb}$ に対する sb 番目のサブブロックの i 番目のユニットの値であり、 \tilde{y}_j^r は入力に対する核パターンである。 s はブロック分割数を表す。 $s=1$ は特別なケースでブロック分割しないことを意味する。

後段の想起は従来の MAM と同様に行う。

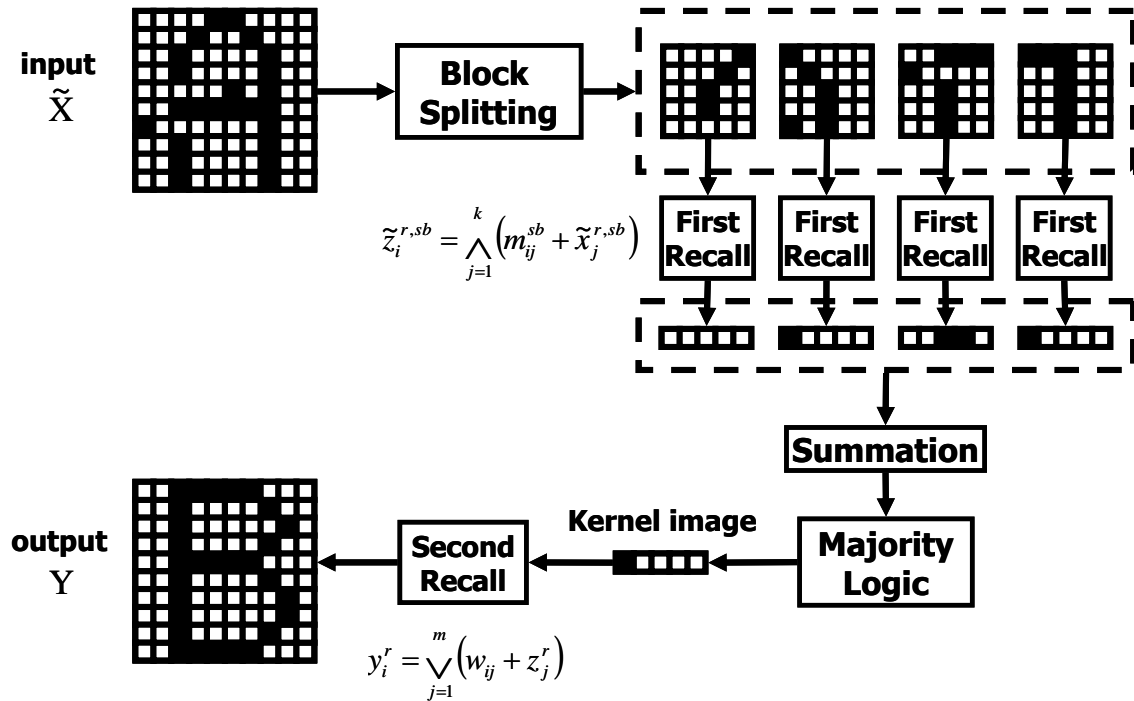


図 3.1 記銘パターンから独立した核パターンを用いた MAM

3.3 自己相関想起実験による従来モデルとの比較

本モデルの性能を評価するため、図 2.5 に示す記録パターンを用いて自己相関想起実験を行った。図 3.2 に我々のモデルと Ritter のモデルにおける自己相関想起実験のノイズ耐性を示す。横軸はノイズ混入率、縦軸は完全想起率を表す。一試行においてシミュレーションを 10000 回行い、その平均値を用いた。図 3.2 中の Ritter とは Ritter のモデルのことである。本モデルは、Ritter のモデルと比較して、想起率が高いことが分かる。

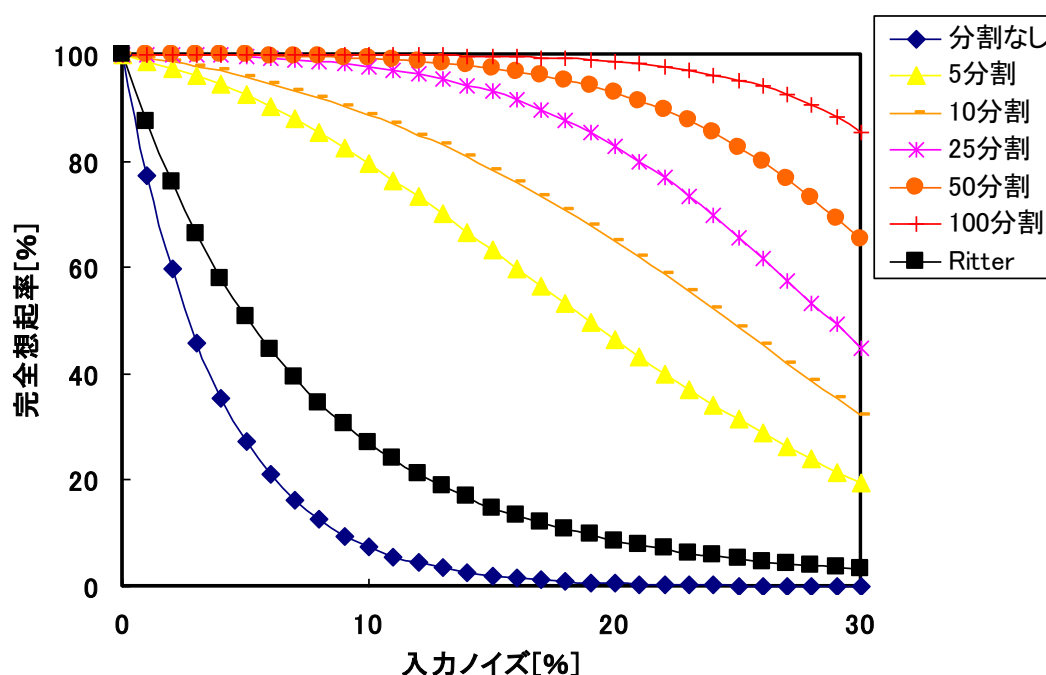


図 3.2 記録パターンから独立した核パターンを用いた MAM における自己相関想起実験のノイズ耐性

また、アルファベット大文字 26 文字(図 3.3)と、対になるパターン(図 3.4)を記録パターンとして用いて相互相関想起実験を行った。図 3.5 に我々のモデルにおける相互相関想起実験のノイズ耐性を示す。

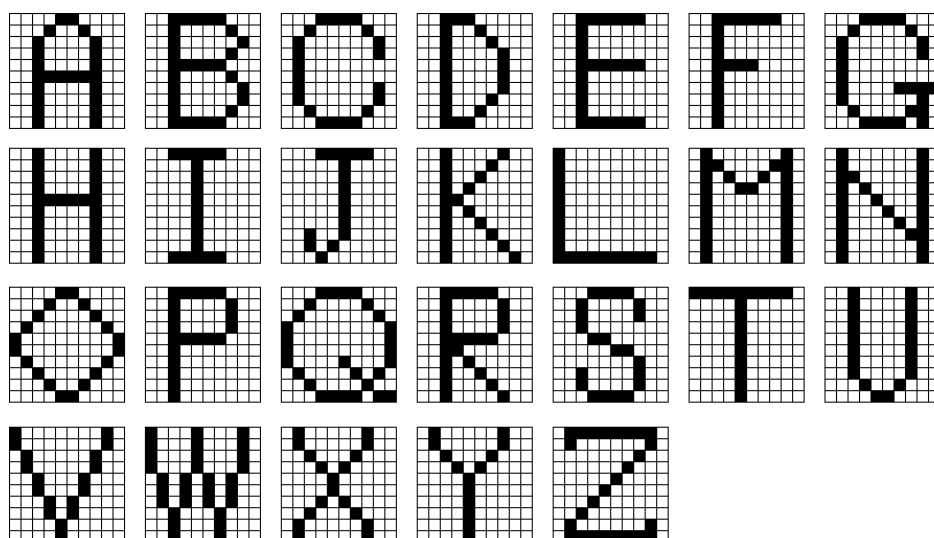


図 3.3 記銘パターン(アルファベット大文字 26 文字)

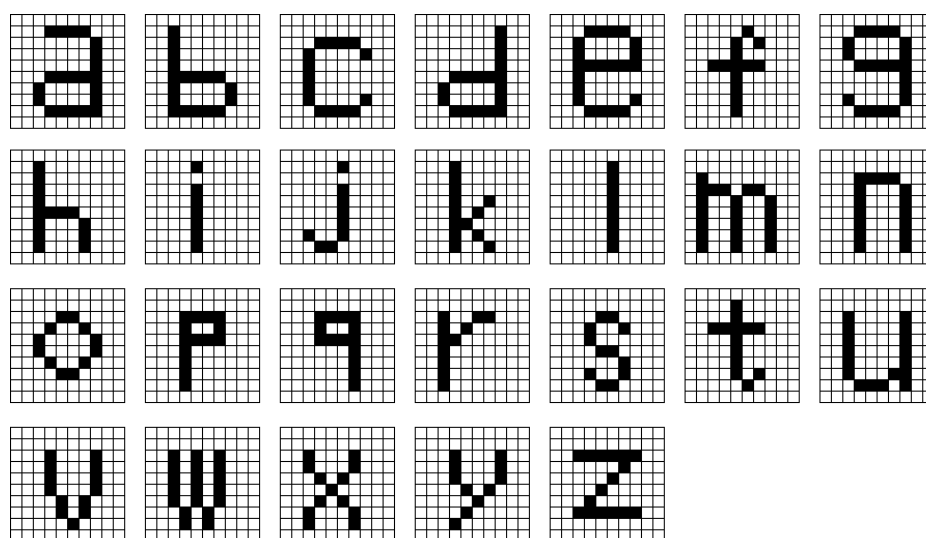


図 3.4 記銘パターン(アルファベット小文字 26 文字)

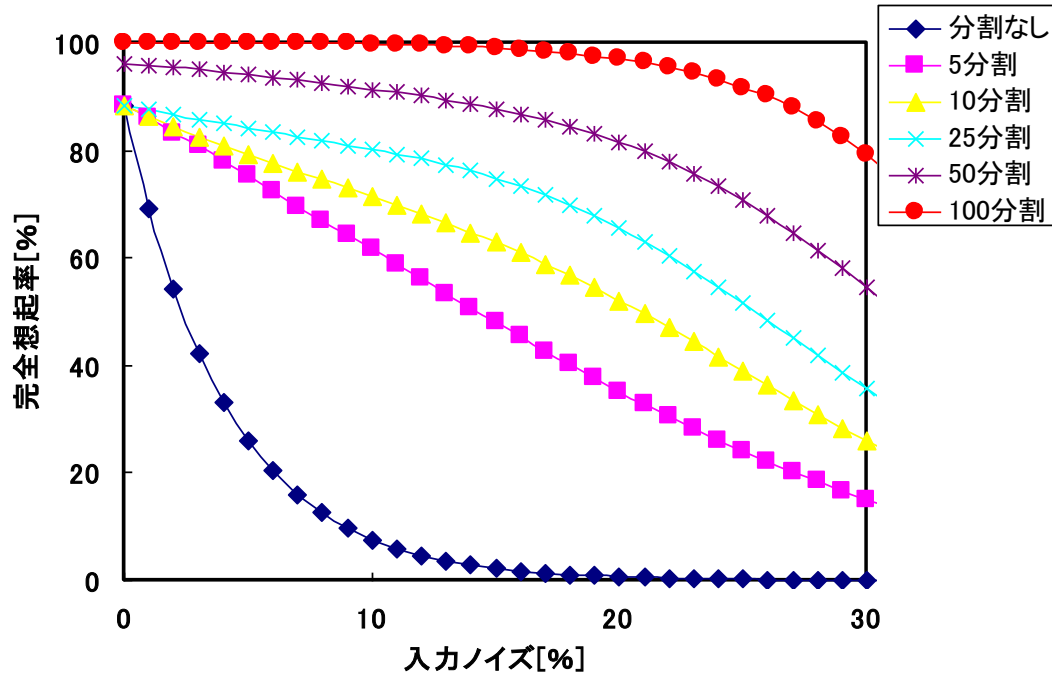


図 3.5 我々のモデルにおける相互相関想起実験のノイズ耐性

3.4 記銘パターンから独立した核パターンを用いた MAM のハードウェア化

MAM モデルの将来の実応用を考え、記銘パターンから独立した核パターンを用いた MAM のハードウェア化を行い、その妥当性を検証する。

記銘パターンから独立した核パターンを用いた MAM の前段想起および後段想起の処理には並列処理が可能な部分があり、ハードウェア化による高速化が期待できる。本モデルのハードウェア化により処理を並列化し、効率的に処理をさせることで高速化を試みる。それに加え、記憶配列 W 、 M の各要素を 1bit で再符号化することにより、バス線およびメモリ削減する。本モデルのハードウェア化は、既存の機器やシステムへの組み込みを容易にするものと期待される。

3.5 処理の並列化

本モデルでの前段想起では、記憶配列 M と入力 X を加算し、サブブロックのユニット内で最小値をとる処理を行う(式(3.3))。また、後段想起では、記憶配列 W と核 Z を加算し、最大値をとる処理を行う(式(3.4))。

$$z_i^{r, sb} = \bigwedge_{j=1}^k (m_{ij}^{sb} + \tilde{x}_j^{r, sb}), \quad (3.3)$$

$$y_i^r = \bigvee_{j=1}^m (w_{ij} + z_j^r) \quad (3.4)$$

これらの処理において、並列化が可能であり、前段想起内での演算または後段想起内での演算を同時に行う手法をとることによって効率的な処理が可能となり、高速化が図れる。

3.6 記憶配列 W 、 M の各要素の 1bit 表現

記銘パターンから独立した核パターンを用いた MAM で 2 値のデータを取り扱った場合、記憶配列 M 、 W は”-1”、”0”、”1”の 3 つの値を取り得るので 2bit で表現しなければならない。しかし、記憶配列 M 、 W を求める際は、式(3.5)、(3.5)に従い、最大値および最小値を求めていく。

$$m_{ij} = \bigvee_{r=1}^R (z_i^r - x_j^r) = (z_i^1 - x_j^1) \vee (z_i^2 - x_j^2) \vee \cdots \vee (z_i^R - x_j^R) \quad (3.5)$$

$$w_{ij} = \bigwedge_{r=1}^R (y_i^r - z_j^r) = (y_i^1 - z_j^1) \wedge (y_i^2 - z_j^2) \wedge \cdots \wedge (y_i^R - z_j^R) \quad (3.6)$$

ここで、 i 、 j はユニット番号、 r はパターン番号、 R は記銘パターン数、 x は入力、 z は核パターン、 y は期待出力を表す。

記憶配列 W 、 M は式(3.5)、(3.6)に従うため、記銘するごとに、次第に記憶配列 M の各要素の値は”1”と”0”、記憶配列 W の各要素の値は”-1”と”0”の、2 つの値を取る。図 3.6 と図 3.7 にその例を示す。

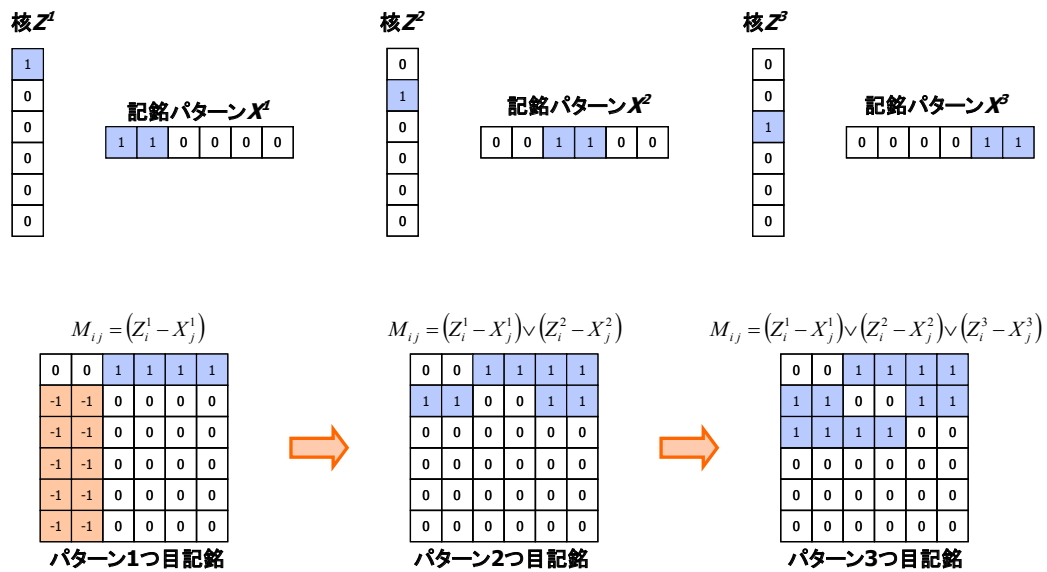


図 3.6 記憶配列 M の算出例

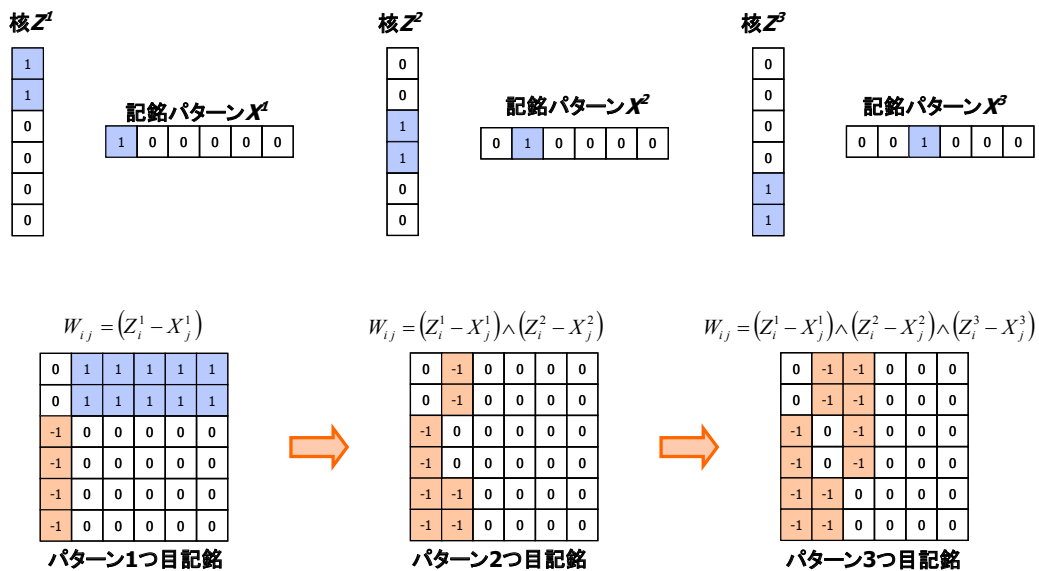


図 3.7 記憶配列 W の算出例

そこで、以下のように記憶配列を再符号化することによって、記憶配列の各要素を 1bit で

表現することが可能となる。

記憶配列 M : “0”、“-1” → ‘0’ “1” → ‘1’
 記憶配列 W : “0”、“1” → ‘0’ “-1” → ‘1’

図 3.8 と図 3.9 に記憶配列の各要素を 1bit で再符号化した例を示す。

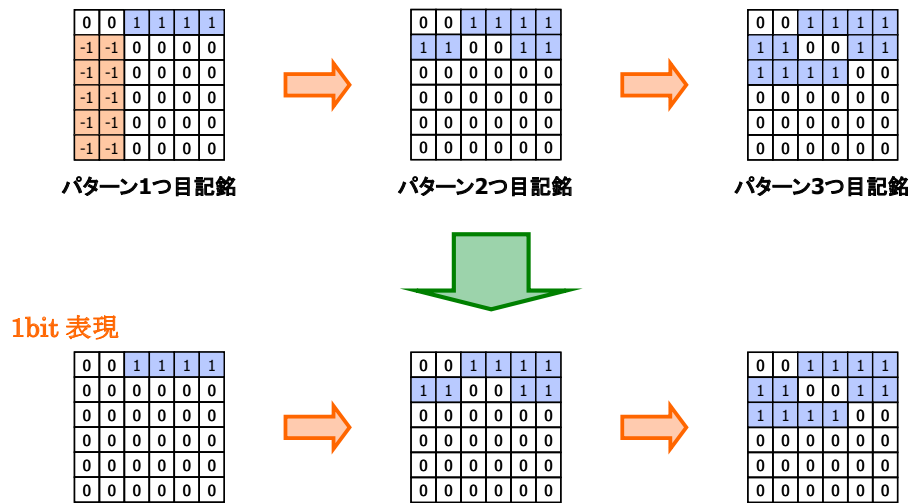


図 3.8 記憶配列 M の各要素の 1bit 表現例

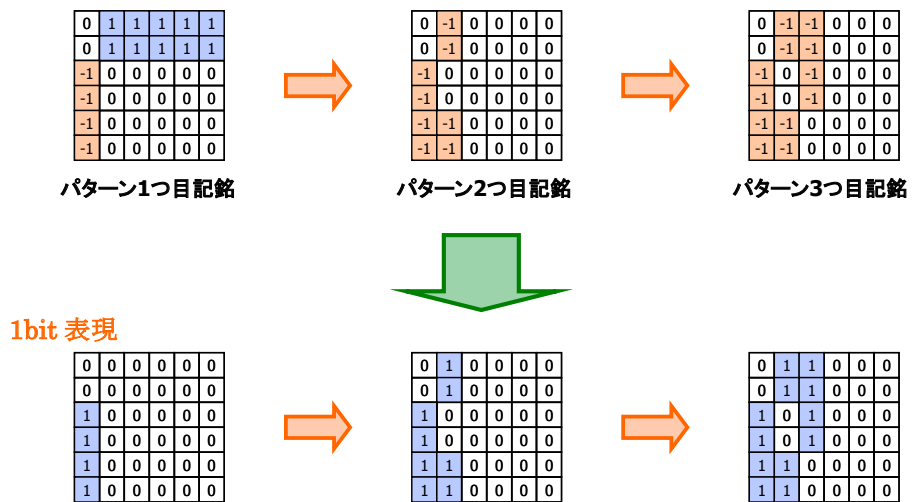


図 3.9 記憶配列 W の各要素の 1bit 表現例

3.7 MAM ハードウェアの仕様

構築した MAM ハードウェアの仕様を示す。本ハードウェアは $8 \times 8 = 64$ ユニットの 2 値 (64bit) のデータを取り扱う。本 MAM ハードウェアモデルでは、最大 32 個まで記録を行うことができる。

3.8 MAM ハードウェアの構成

3.8.1 MAM ハードウェアの全体ブロック

設計した MAM ハードウェアの全体ブロックを図 3.10 に示す。MAM ハードウェアは大きく分けて、パターンを記録する“**Memory Unit(記録ユニット)**”、入力と記憶配列 M を加算し最小値をとり、核を足しこむ“**First Recall Unit(前段想起ユニット)**”、足しこまれた核から多数決を行い、核パターンを決定する“**Majority Logic Unit(多数決ユニット)**”、核パターンと記憶配列 W を加算し最大値をとり想起結果を出力する“**Second Recall Unit(後段想起ユニット)**”からなる。

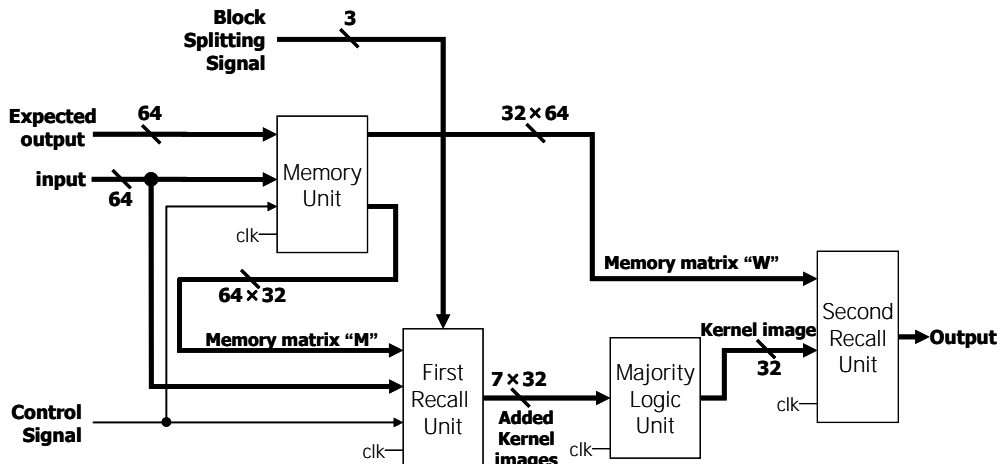


図 3.10 MAM ハードウェアの全体ブロック図

3.8.2 記録ユニット

記録ユニット内部ブロック図を図 3.11 に示す。記録ユニットは、核パターン生成ユニット、記憶配列 M 更新ユニット、記憶配列 W 更新ユニットからなる。図 3.12 に記憶配列 W 、 M 更新ユニット内部のブロック図を示す。

核パターン生成ユニットは、Control Signal(ND 信号)の'1'に対し、
 $"100\cdots 0" \rightarrow "0100\cdots 0" \rightarrow "0010\cdots 0" \cdots$
 を出力する回路である。

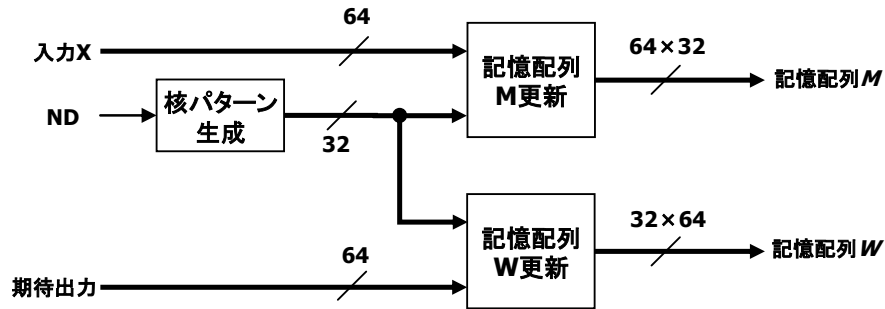


図 3.11 記銘ユニット内部ブロック図

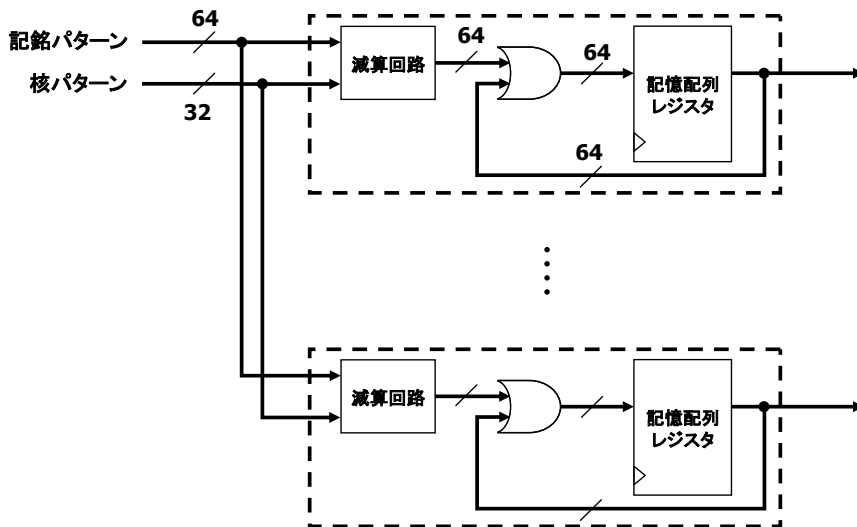


図 3.12 記憶配列 M 、 W 更新ユニット内部ブロック図

3.8.3 前段想起ユニット

前段想起ユニット内部のブロック図を図 3.13、図 3.14 に示す。前段想起ユニットでは、入力 X と記憶配列 M レジスタから送られてくる値を加算し、分割数に応じて最小値をとる。最小値を求める際は、分割数によって最小値を求めるユニット数が異なるため、分割数に応じて最小値を算出する最小値算出ユニットの出力をセレクトによって切り替えている。

核足しこみユニットでは、最小値算出ユニットから出力を受け取り、'1'となる値をカウントし、出力する。

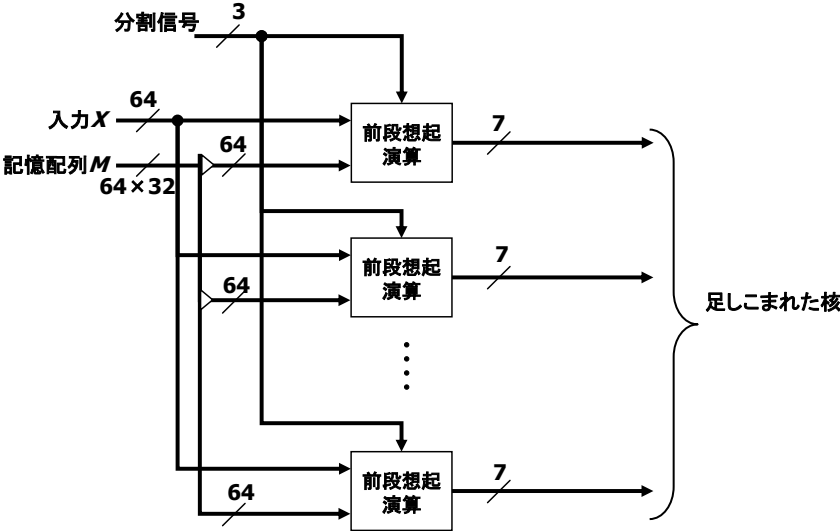


図 3.13 前段想起ユニット内部ブロック図

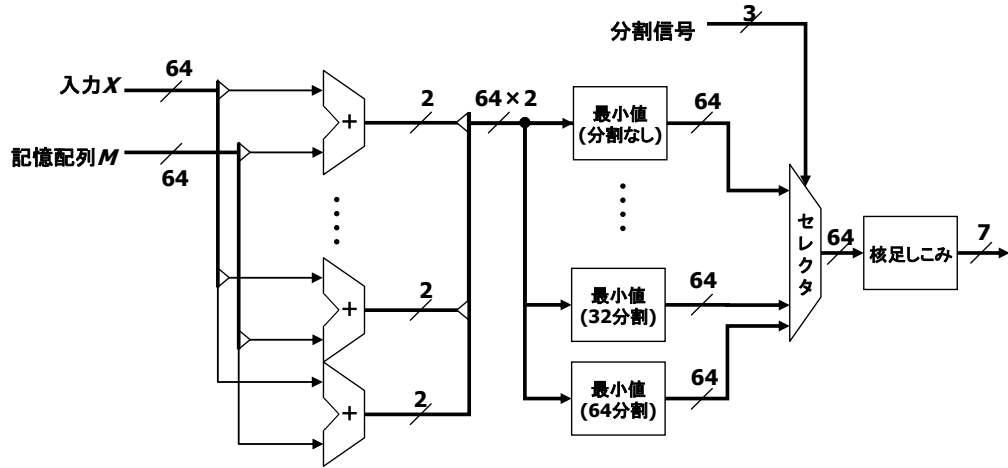


図 3.14 前段想起演算ユニット内部ブロック図

3.8.4 多数決ユニット

多数決ユニットの内部のブロック図を図 3.15 に示す。多数決ユニットでは、まず、前段想起ユニットから送られてくる、核の足しこまれた値から最大値をとる。次に、その最大値と等しくなる核ユニットを探し、等しければ'1'を、それ以外は'0'とし、出力する。出力した値は、多数決によって決定された核パターンとなっている。

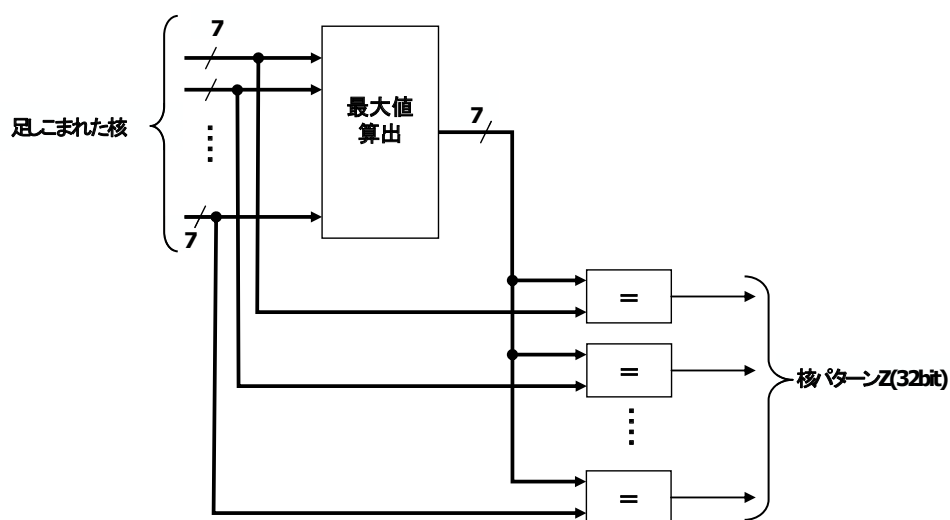


図 3.15 多数決ユニット内部ブロック図

3.8.5 後段想起ユニット

後段想起ユニット内部のブロック図を図 3.16 に示す。後段想起ユニットでは、多数決ユニットから送られてくる核パターン Z と記憶配列 W レジスタから送られてくる値を減算し、最大値をとる。出力された値が想起結果である。

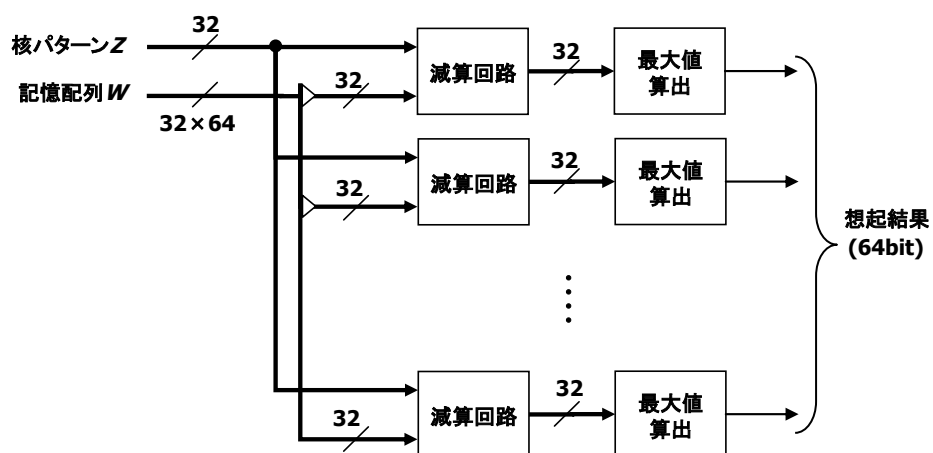


図 3.16 後段想起ユニット内部ブロック図

3.9 MAM ハードウェアの性能評価

MAM ハードウェアは、ハードウェア記述言語の VHDL により設計した。ハードウェアシミュレータを用いてクロック精度の動作確認および、処理能力の確認を行った。ここでは、ターゲットデバイスとして、汎用機器に良く用いられる Xilinx 社製の Spartan3 を想定した。8×8=64 の 2 値のデータ(64bit)、核ユニット数は 32 個、図 3.17 に示す記銘パターンを用いて自己相関想起を行った。

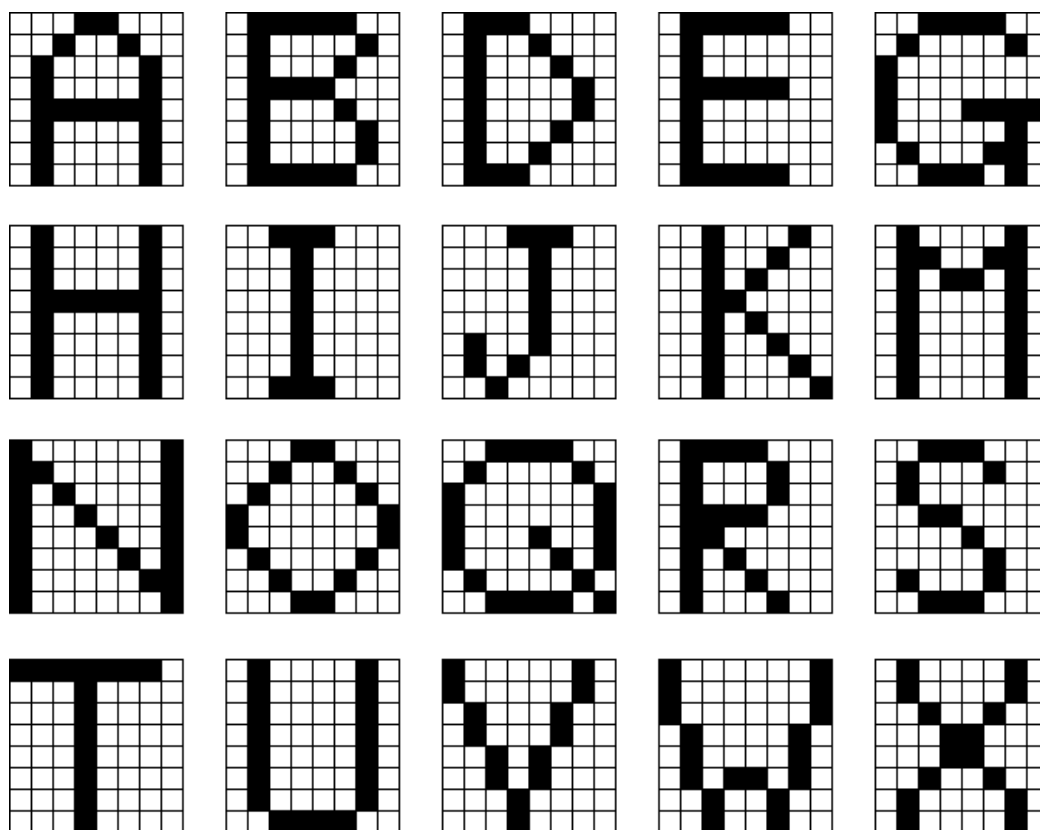
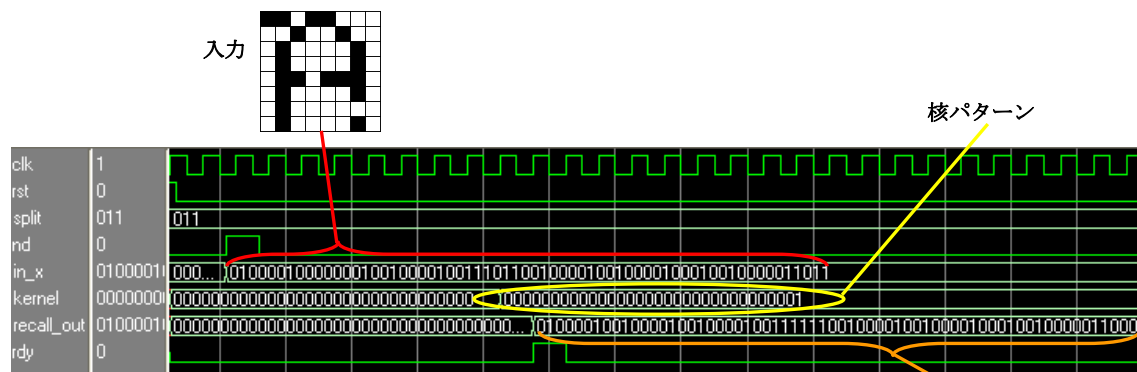
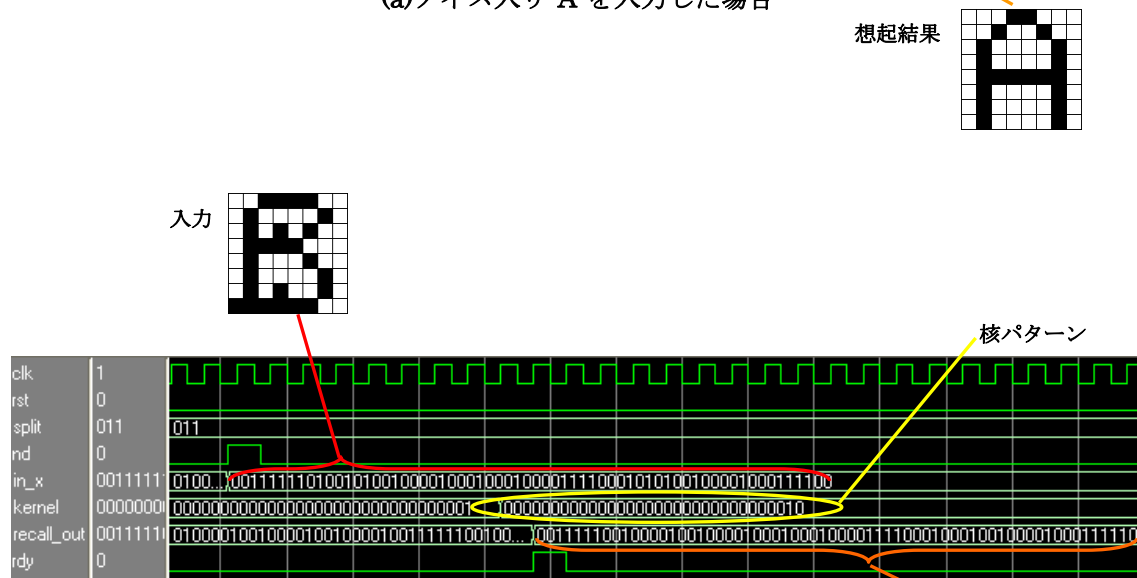


図 3.17 8×8 の記銘パターン(アルファベット大文字 20 文字)

図 3.18 にシミュレーション結果を示す。分割数は 8 分割である。



(a) ノイズ入り”A”を入力した場合



(b) ノイズ入り”B”を入力した場合

図 3.18 ハードウェアシミュレーション結果

図 3.18 中の nd は入力データが有効であるか、rdy は出力データが有効であるかの信号である。split は分割信号のことであり、値は 8 分割に相当する。図 3.18(a)では、ノイズ入り”A”を入力している。核パターンは、”A”に対応する核パターン一つが想起されており、記録された”A”を正しく想起していることが分かる。また、図 3.18(b)では、ノイズ入り”B”を入力している。この場合も、”B”に対応する核パターン一つが想起されており、記録された”B”を正しく想起していることが分かる。

以上の結果から、開発した MAM ハードウェアモデルが正常に動作していることを確認した。また、データが入力されてから 10 クロックで想起されていることが分かる。この MAM ハードウェアモデルを 40MHz で動作させた場合と、ソフトウェア処理との比較結果

を表 3-1 に示す。ソフトウェア処理には CPU Intel Xeon 3.0GHz の PC を用いた。

表 3-1 MAM ハードウェアモデルとソフトウェアの比較

モデル	規模 (核ユニット数/slice)	動作周波数 (MHz)	処理速度 (μ sec)
ソフトウェア	32 個/-	3000	26
ハードウェア	32 個/13310	40	0.25

本ハードウェアモデルは、ソフトウェア処理と比較して、104 倍の高速化が図れることを確認した。ターゲットデバイスとして Spartan3(xc3s1500)を想定した場合、本ハードウェアモデルの slice 数は 13310 となり、FPGA 全体(slice 数 13312)の 99%を使用する。このデバイスでは、64bit のデータで 32 個までの記銘が可能である。

3.10 記銘パターンから独立した核パターンを用いた MAM の問題点

佐伯も指摘しているように[21]、図 3.5 において、相互相関想起では自己相関想起とは異なり、入力パターンにノイズを含まない場合でも、完全想起しない場合が存在する。これは図 3.19 に示すように、あるパターンに完全に含まれるパターン(包含関係にあるパターン)が入力された場合、あるパターンとそのパターンに完全に含まれるパターンの核パターンが同時に想起されるためである。記銘パターンから独立した核パターンを用いた MAM では、核パターンを多数決で決定しているが、同数の場合は複数の核パターンを勝者としている。そのため、分割数が少ない場合、全てのサブブロックで包含関係が成立し、複数の核パターンが出力される。

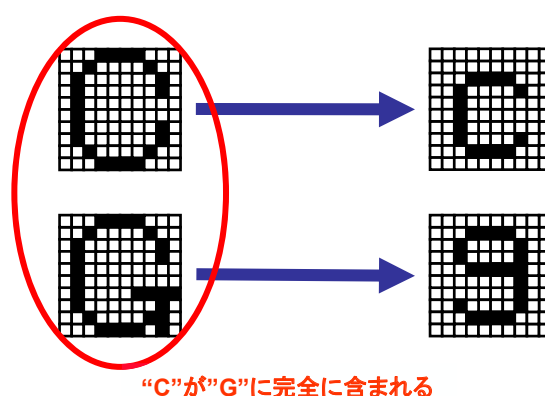


図 3.19 包含関係にあるパターン

第4章 逆想起を用いた MAM

4.1 概要

記銘パターンから独立した核パターンを用いた MAM モデルでは、Ritter の MAM モデルの欠点であった核パターン設計の改善がなされている。核パターンを自由に決定することが可能で、核パターン設計が容易となっており、Ritter のモデルよりも、想起率の改善がなされている。しかし、本モデルの問題点として、図 3.19 のように、あるパターンに完全に含まれているパターン(包含関係にあるパターン)が記銘されると、相互相関想起において正しく想起できない[21]。

本章では、このモデルの問題点を解決した包含パターンが想起可能な逆想起を用いた MAM モデルを提案する[23]。核パターンから記銘パターンを逆想起し、入力と比較させ、核パターンを一つに絞ることにより正しく想起できることを示す。また、逆想起を用いた MAM では、非常にシンプルな演算を用いていること、並列に処理が可能なアーキテクチャとなっているため、ハードウェア化に適した連想記憶モデルとなっている。我々は、逆想起に用いている演算を、よりシンプルな演算に簡略化することで、よりハードウェア化に適したモデルを提案する[25]。この単純化した逆想起を用いた MAM ハードウェアモデルを提案することにより、より高速に想起が可能な MAM ハードウェアを開発する[26][27]。

4.2 逆想起を用いた MAM のアルゴリズム

核パターンから独立した核パターンを用いた MAM において、包含関係にあるパターンが入力されると、前段想起で、あるパターンの核パターンとそのパターンに完全に含まれるパターンの核パターンが同じ数だけ想起される。このモデルでは核パターンを多数決で決定されているが、同数の場合は複数の核パターンが出力される。そのため、複数の核パターンが想起され、複数の核パターンから包含関係でない出力パターンが重なって想起される。これにより、包含関係にあるパターンは、相互相関想起において正しく想起されない(図 4.1)。

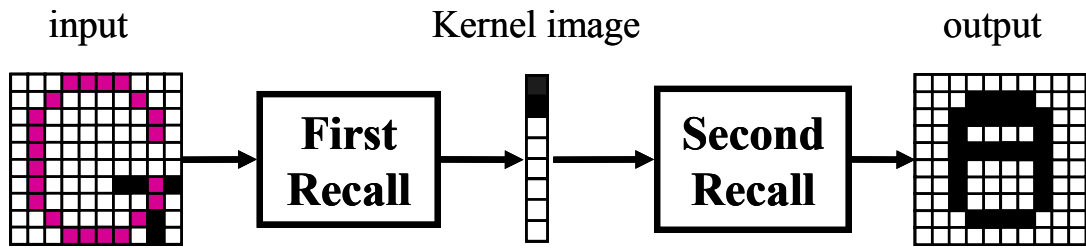


図 4.1 複数の核パターンが想起される様子

この問題を解決するために、多数決によって複数の核パターンが決定された場合、さらに複数の核パターンから一つの核パターンに絞ることを考えた。我々の従来のモデルでは、“C” や “G” のように包含関係にあるパターンが入力されると複数の核パターンが想起される。まず、この複数想起された核パターンをそれぞれ一つの核パターンに分割する。その分割したそれぞれの核パターンから記銘パターンを逆想起させ、その逆想起した記銘パターンと入力とのハミング距離をとる。ハミング距離が最も小さかった核パターンを、入力に対応する核パターンとすることにより、複数の核パターンから一つに絞ることが可能である。

核パターン $Z^r = (z_1^r, \dots, z_m^r)$ から記銘パターン $X^r = (x_1^r, \dots, x_n^r)$ の逆想起は式(4.1)に従う。

$$x_j^r = \bigvee_{i=1}^m (z_i^r - m_{ij}), \quad (4.1)$$

ここで、 i, j はユニット番号、 r はパターン番号を表す。

図 4.2 に逆想起を用いた MAM の処理の流れを示す。
本モデルの想起過程を以下に示す。

1. 記銘パターンから独立した核パターンを用いた MAM と同様に、足しこまれた核パターンを多数決で決定する。

2. 複数核パターンが想起された場合、複数の核パターンを分割する。
3. それぞれの核パターンから、記銘パターンを逆想起する。
4. 逆想起したパターンと入力とのハミング距離を計算する。
5. ハミング距離が最も小さかった核パターンを勝者とし、核パターンに決定する。
6. 最後に、一つに絞られた核パターンを入力とし、後段想起において出力パターンが想起される。

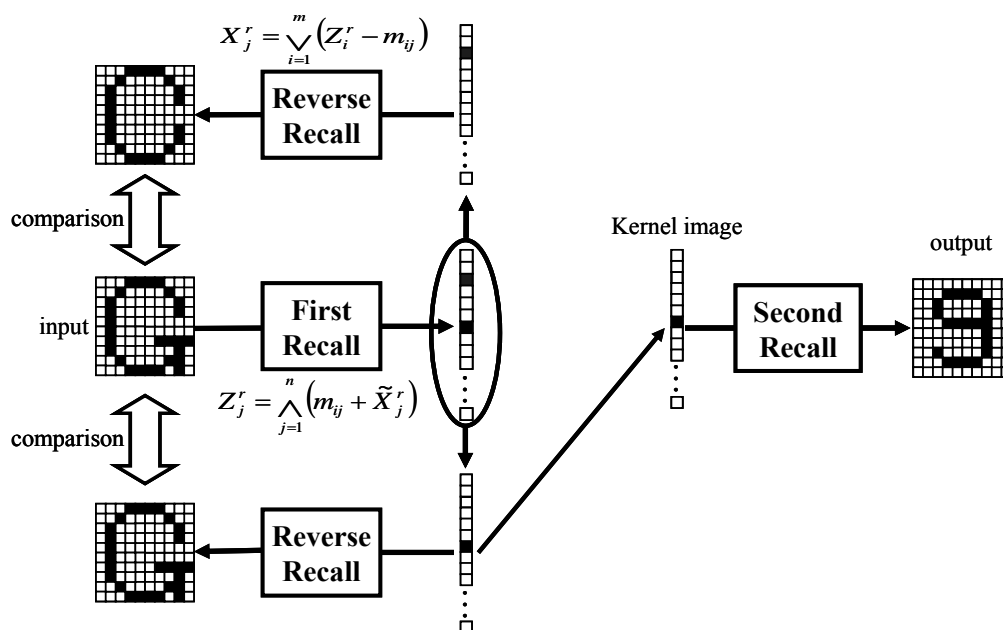


図 4.2 逆想起を用いた MAM の処理の流れ

4.3 相互相関想起実験に対する逆想起の効果

逆想起を用いた MAM の性能を評価するために、包含関係にあるパターンのみを用いて相互相関想起実験を行い、我々の従来モデルと完全想起率を比較した。一試行においてシミュレーションを 10000 回行い、その平均値を用いた。

図 4.3 に 2 個の包含関係にあるパターン 10 組、図 4.4 に 3 個の包含関係にあるパターン 10 組、対になるパターンはそれらのパターンとは異なるパターンとし、これらのパターンを記銘パターンとする。各々のパターンは $10 \times 10 = 100$ のバイナリユニットからなる。図 4.5、図 4.6 に 2 個の包含関係にあるパターンを記銘パターンに、図 4.7、図 4.8 に 3 個の包含関係にあるパターンを記銘パターンに用いた際の、逆想起を用いた MAM と従来モデルのノイズ耐性を示す。横軸はノイズ混入率、縦軸は完全想起率を表す。

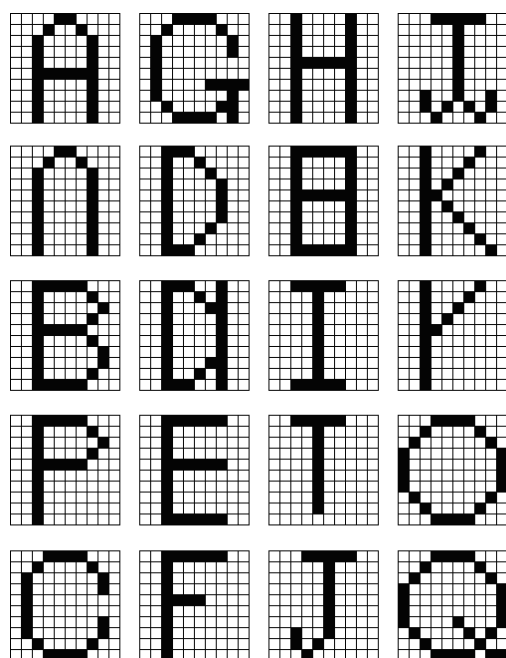


図 4.3 2 個の包含関係にあるパターン 10 組

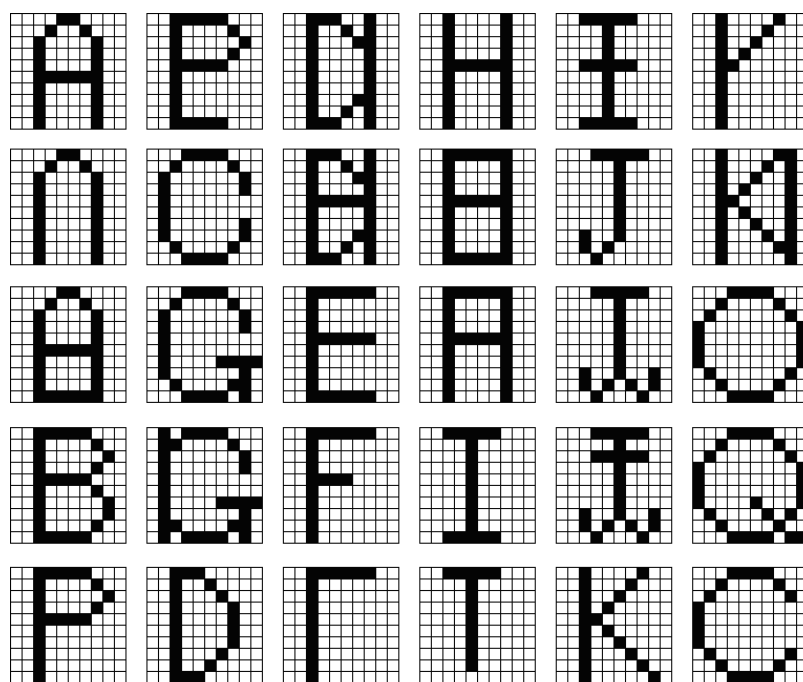


図 4.4 3 個の包含関係にあるパターン 10 組

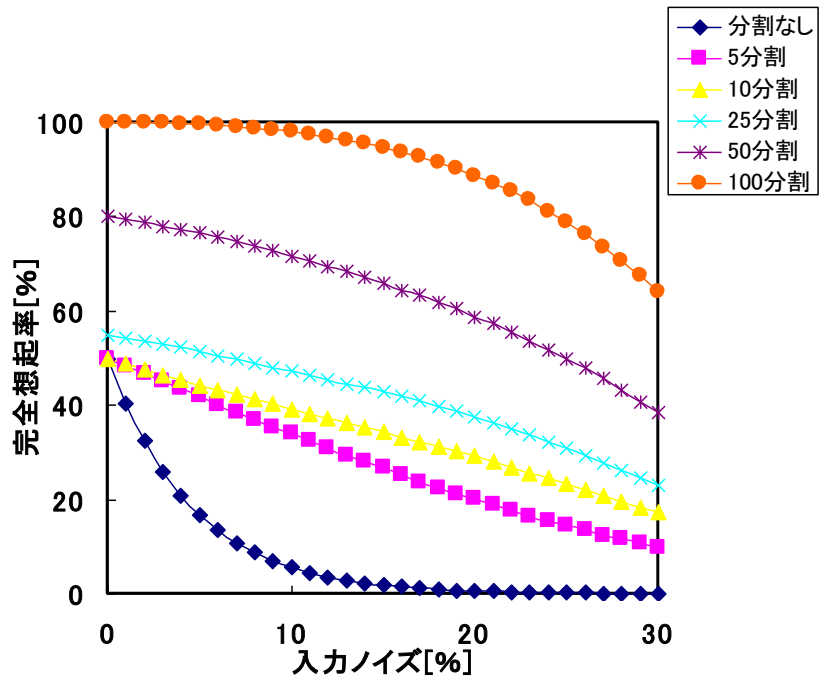


図 4.5 従来モデルにおける相互相関想起実験のノイズ耐性(2 個の包含関係にあるパターン 10 組の記録の場合)

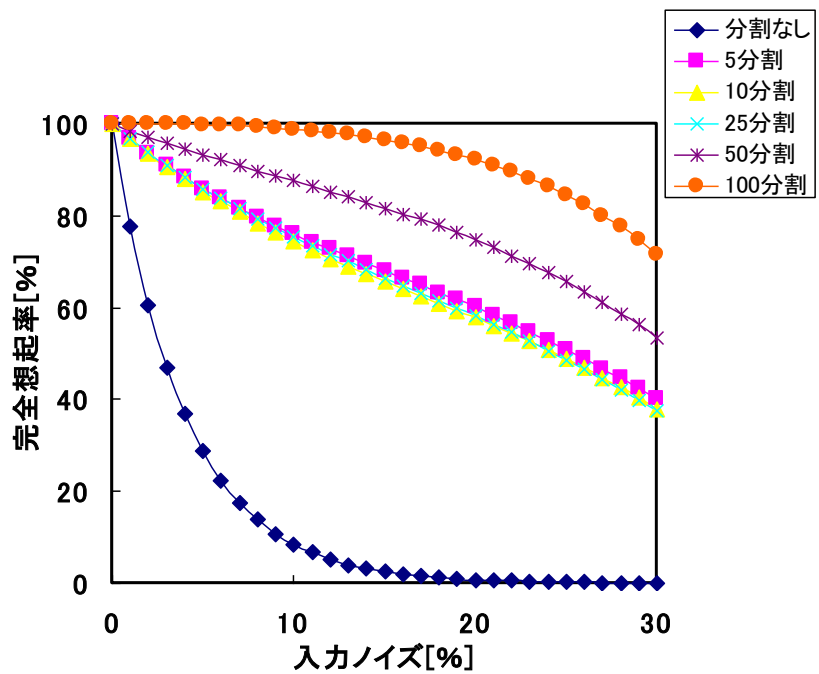


図 4.6 逆想起を用いた MAM における相互相関実験のノイズ耐性(2 個の包含関係にあるパターン 10 組を記録の場合)

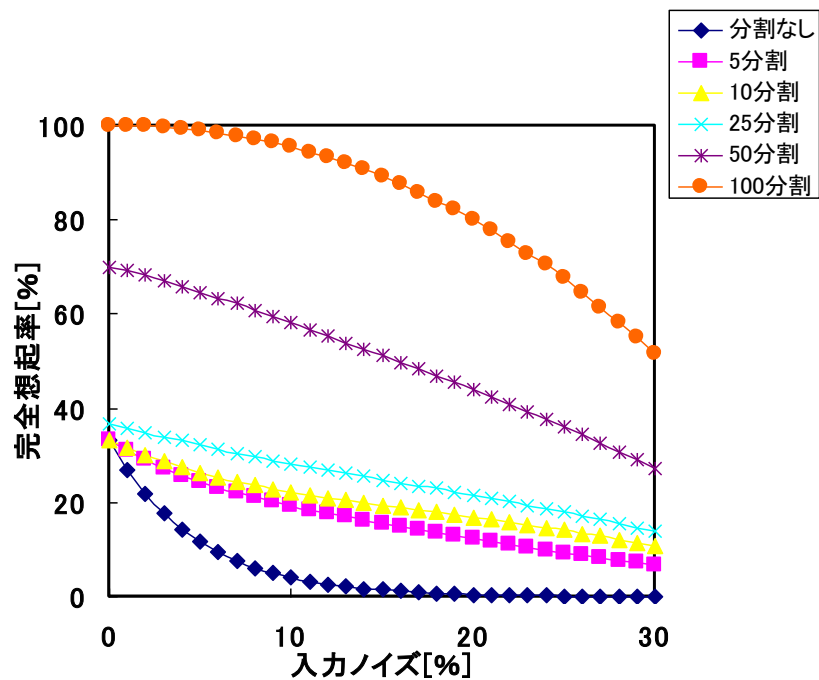


図 4.7 従来モデルにおける相互相関想起実験のノイズ耐性(3 個の包含関係にあるパターン 10 組を記録した場合)

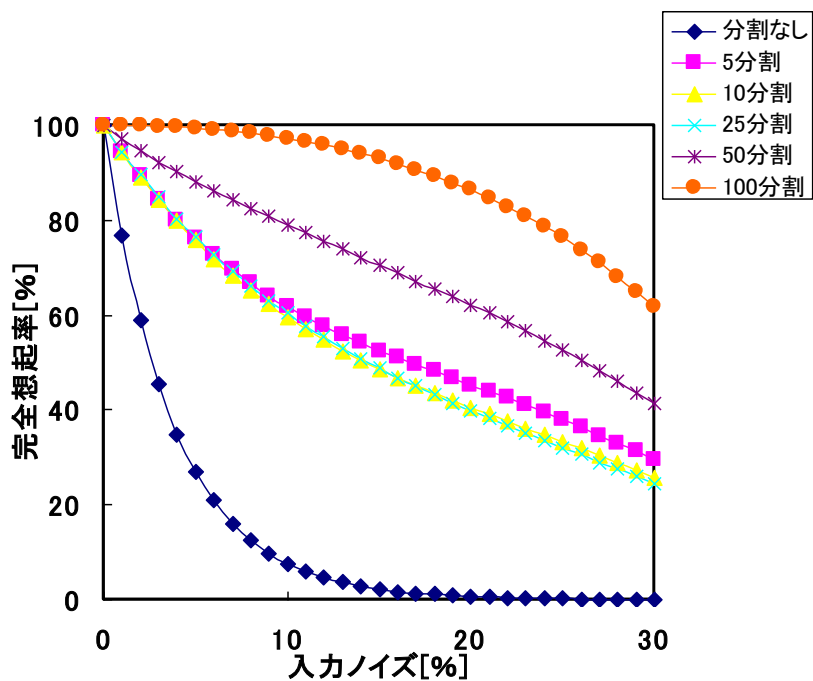


図 4.8 逆想起を用いた MAM における相互相関想起実験のノイズ耐性(3 個の包含関係にあるパターン 10 組を記録した場合)

図 4.5～図 4.8 の結果から、従来モデルではノイズが 0% の場合でも包含関係にあるパターンは正しく想起できなかったのに対して、逆想起を用いた MAM ではどの分割数でもノイズ 0% の場合、完全想起率が 100% になっていることが分かる。したがって、提案手法では、包含関係にあるパターンでも正しく想起可能である。

4.4 逆想起の演算の簡単化

逆想起を用いた MAM において、記録パターンから独立した核パターンを用いた MAM と比較して、記録パターンを逆想起するだけの、追加の演算が必要となる。このモデルでは、包含パターンにおける包含しているパターンの数が増加すると、追加の想起時間も増加してしまう。追加の想起時間 t は式(4.2)に従い増加する。

$$t = 0.5 \times N_{ip} \times t_{recall}, \quad (4.2)$$

ここで、 N_{ip} は包含しているパターン数、 t_{recall} は包含パターンがない場合の想起時間を表す。逆想起を用いた MAM では、重なった核パターンが想起された場合、重複した核パターンを分離し、分離した核パターンを用いて、それぞれの核パターンに対応した記録パターンを逆想起している。そのため、核パターンを分離した分、それに対応する記録パターンを想起する必要がある。

ここで、逆想起を用いた MAM では、核パターン Z^r に対応する記録パターン X^r を式(4.3)に従い、逆想起する。

$$x_j^r = \bigvee_{i=1}^R (z_i^r - m_{ij}), \quad (4.3)$$

ここで、 x_j^r は記録パターン X^r の j 番目のユニット、 z_i^r は核パターン Z^r の i 番目のユニット、 R は記録パターン数を表す。2 値パターンの場合、式(4.3)は式(4.4)に簡略化することが可能である。

$$x_j^r = z_r^r - m_{rj}. \quad (4.4)$$

式(4.4)は、分離したそれぞれの核パターンに対応した記録パターンを、核パターンの一つのユニットで想起可能であることを示している。従って、逆想起部分に式(4.4)を用いることにより、記録パターンを一回の想起で、逆想起可能となる。

図 4.9 は簡単化した逆想起を用いた MAM の逆想起過程を表す。簡単化した逆想起を用いた MAM では、前段想起において重なった核パターンが想起された場合、式(4.4)を用い

て、核パターンの発火したユニットに対応した記録パターンを逆想起する。従来の逆想起を用いた MAM と同様、逆想起された記録パターンと入力パターンとのハミング距離を比較し、入力に対応した尤もらしいパターンを決定し、その尤もらしい核パターンを用いて、後段想起において出力を想起する。

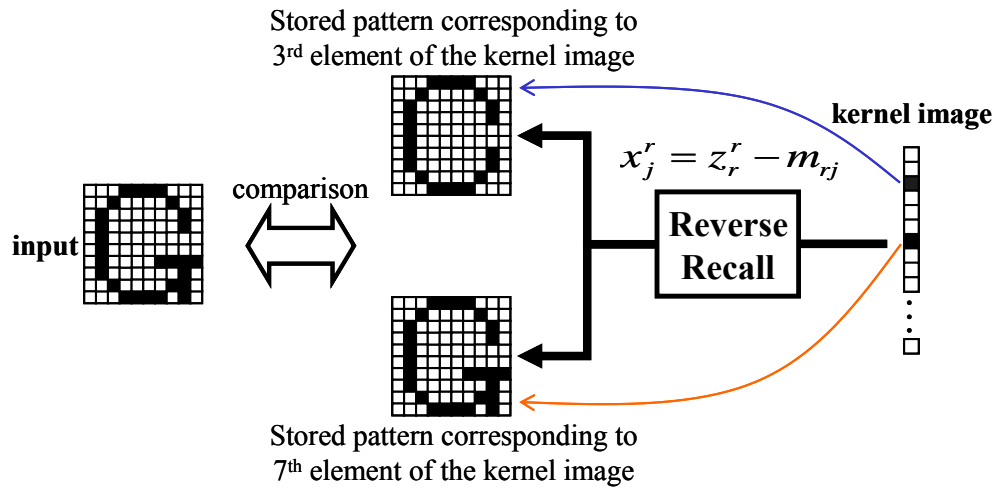


図 4.9 簡単化した逆想起を用いた MAM の逆想起過程

本モデルの想起過程を以下に示す。

1. 従来の逆想起を用いた MAM と同様に、前段想起において核パターンを想起する。
2. 複数核パターンが想起された場合、図 4.9 に示されるように、その核パターンの発火しているユニットに対応する核パターンを逆想起する。
3. 逆想起したパターンと入力とのハミング距離を計算する。
4. ハミング距離が最も小さかった核パターンを勝者とし、核パターンに決定する。
5. 最後に、一つに絞られた核パターンを入力とし、後段想起において出力パターンが想起する。

4.5 相互相関想起実験に対する簡単化した逆想起の効果

簡単化した逆想起を用いた MAM の性能を評価するために、包含関係にあるパターンのみを用いて相互相関想起実験を行い、従来の逆想起を用いた MAM と完全想起率を比較した。一試行においてシミュレーションを 10000 回行い、その平均値を用いた。記録パターンには、図 4.3 の 2 個の包含関係にあるパターン 10 組、図 4.4 の 3 個の包含関係にあるパターン 10 組を用いて、相互相関想起実験を行った。

我々は初めに、図 4.3、図 4.4 を記録パターンに用いた際の、逆想起を用いた MAM と従来モデルのノイズ耐性を調べた。図 4.10 は従来の逆想起を用いた MAM のノイズ耐性を、図 4.11 は簡単化した逆想起を用いた MAM のノイズ耐性を示している。横軸はノイズ混入率、縦軸は完全想起率を表す。

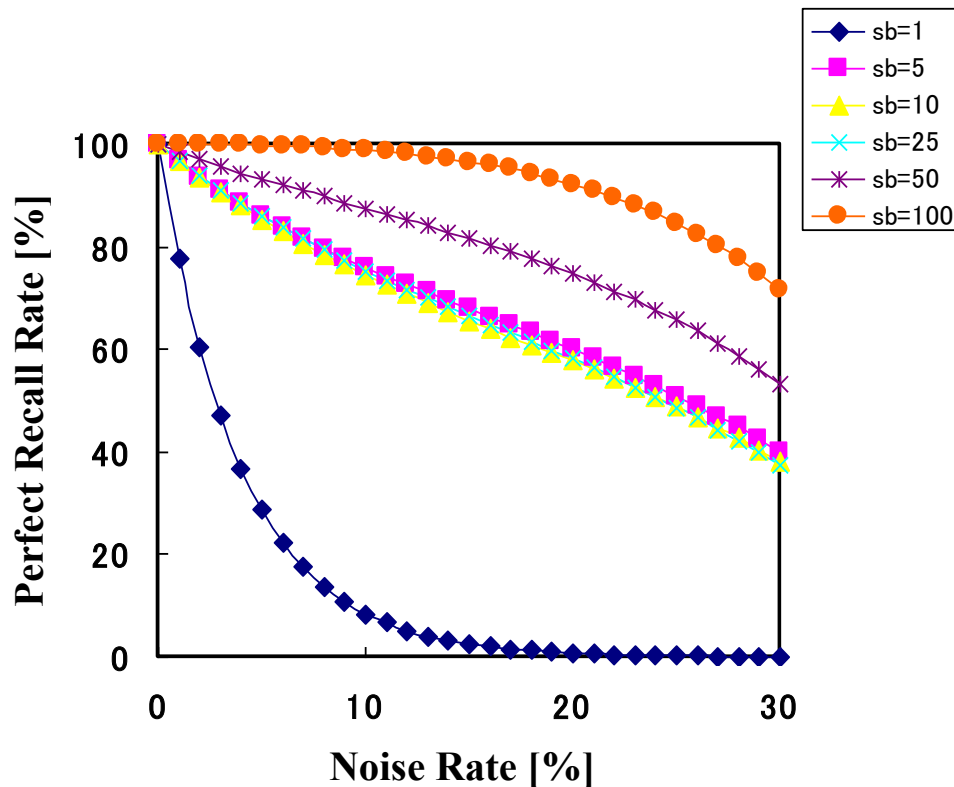


図 4.10 逆想起を用いた MAM における相互相関想起実験のノイズ耐性

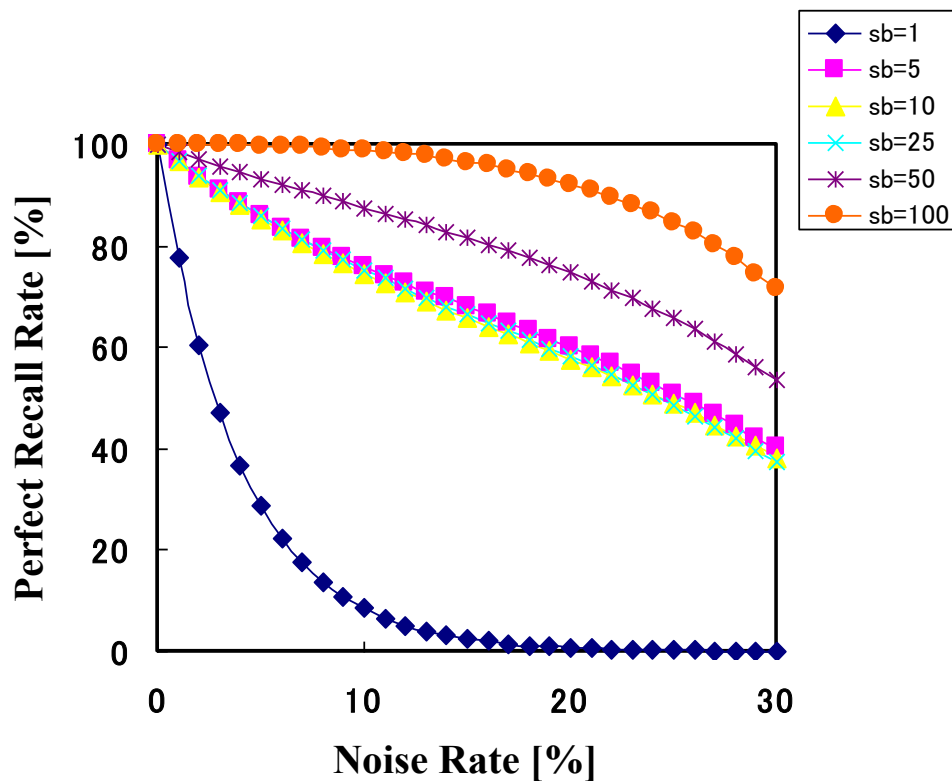


図 4.11 単純化した逆想起を用いた MAM における相互相関想起実験のノイズ耐性

図 4.10、図 4.11 の結果より、逆想起の演算を単純化した場合でも、ノイズ耐性が低下していないことが分かる。

次に、本モデルにおける想起時間の実験を行った。本実験において、一回の想起にかかる時間は、一試行においてシミュレーションを 10000 回行い、その平均値を用いた。記録パターンには、図 4.3、図 4.4 に示すパターンを用いている。

表 4-1 単純化した逆想起を用いた MAM と従来の逆想起との一回の想起時間比較

	一回の想起時間 (μ sec)	
	従来モデル	単純化した逆想起
2個の包含関係にあるパターン	83.2	62.4 (25% 削減)
3個の包含関係にあるパターン	106.5	63.5 (40% 削減)

表 4-1 は単純化した逆想起を用いた MAM と従来の逆想起との、一回の想起時間比較を示している。表 4-1 の結果より、従来モデルの一回の想起時間と比較して単純化した逆想起では、2 個の包含関係にあるパターンに対しては 25%削減、3 個の包含関係にあるパターンに対しては 40%削減されていることが分かる。

4.6 単純化した逆想起を用いた MAM のハードウェアモデル

単純化した逆想起を用いた MAM では、非常にシンプルな演算を用いていること、また、並列に処理が可能なアーキテクチャとなっているため、ハードウェア化に適した連想記憶モデルとなっている。そこで、我々はこの単純化した逆想起を用いた MAM のハードウェアモデルを提案する。

本研究で設計した逆想起を用いた MAM ハードウェアの全体ブロックを図 4.12 に示す。本ハードウェアモデルでは、記録パターンから独立した核パターンを用いた MAM のハードウェアモデル[22]に、逆想起ユニットを導入している。逆想起を用いた MAM ハードウェアは大きく分けて、パターンを記録する“**Memory Unit(記録ユニット)**”、入力と記憶配列 M を加算し最小値をとり、核を足しこむ“**1st Recall Unit(前段想起ユニット)**”、足しこまれた核から多数決を行い、核パターンを決定する“**Majority Logic Unit(多数決ユニット)**”、複数の核パターンが想起された場合、記録パターンを逆想起し入力と比較を行い、入力に対応する核パターンを決定する“**Reverse Recall Unit(逆想起ユニット)**”、核パターンと記憶配列 W を加算し最大値をとり想起結果を出力する“**2nd Recall Unit(後段想起ユニット)**”からなる。

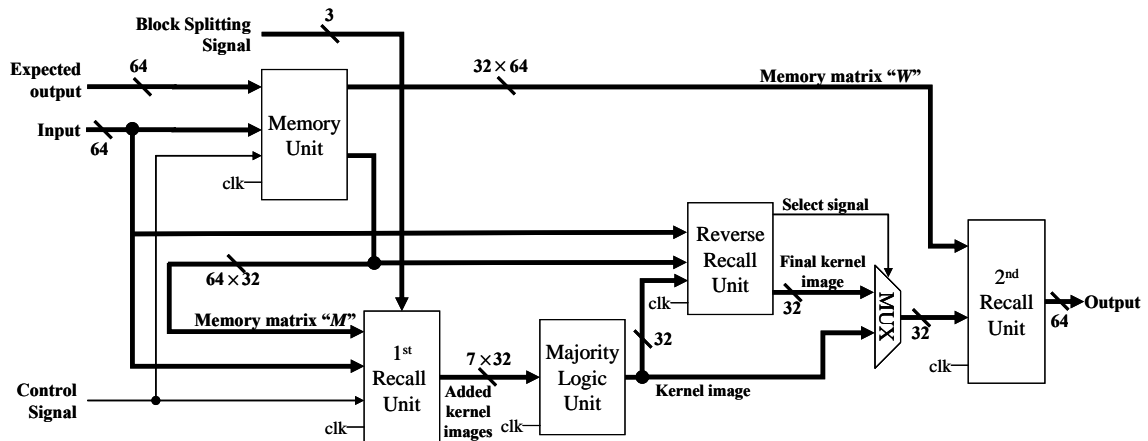


図 4.12 単純化した逆想起を用いた MAM ハードウェアモデル

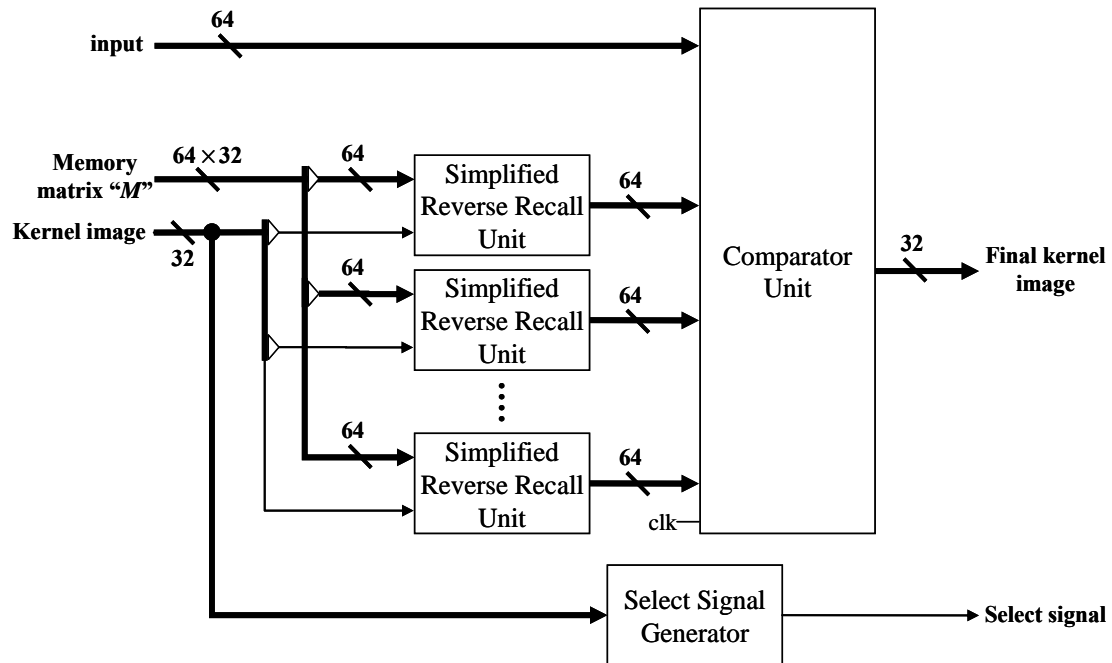
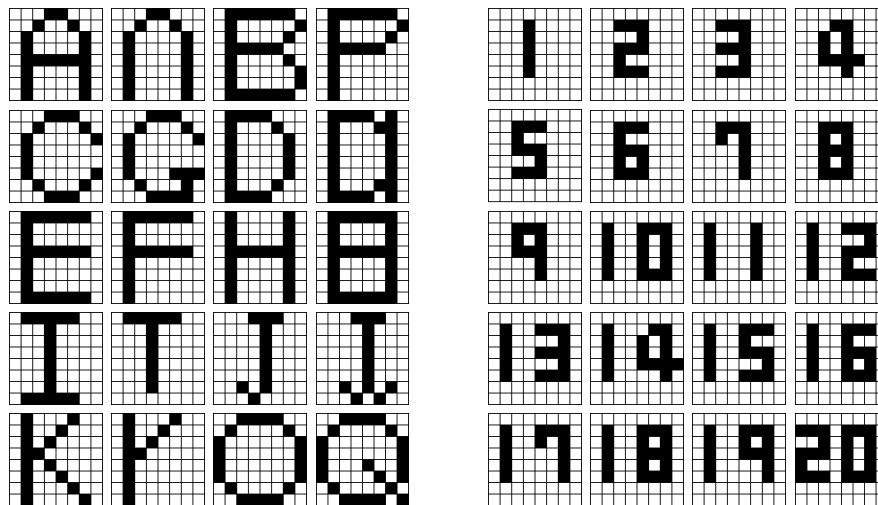
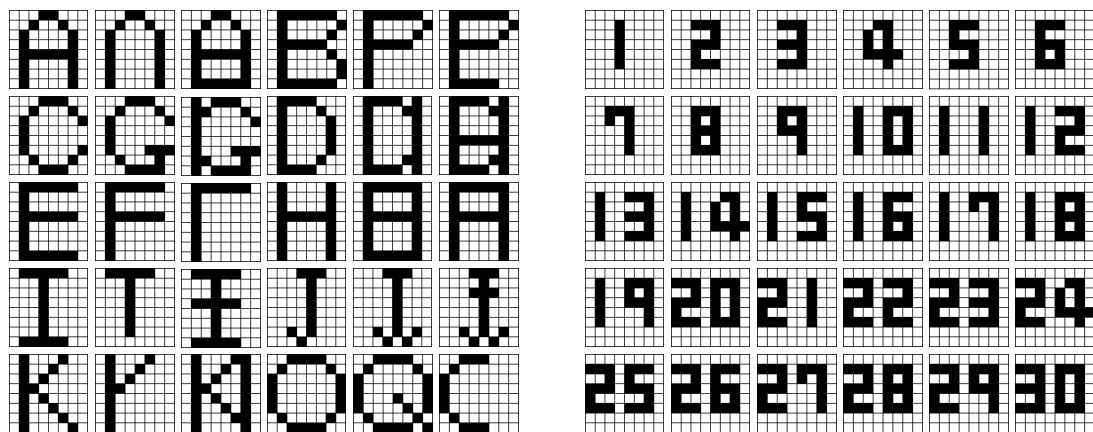


図 4.13 逆想起ユニットのブロック図

図 4.13 は逆想起ユニットを構成するブロック図を示している。“**Simplified Reverse Recall Unit**”では、簡単化した逆想起の演算式 (4.4) を用いて、記録したパターンの逆想起を行っている。“**Comparator Unit**”では、“**Simplified Reverse Recall Unit**”において、逆想起された記録パターンと入力パターンとの比較を行い、入力パターンとのハミング距離が最も短い記録パターンに対応する核パターンを出力する。“**Select Signal Generator**”は、核パターンが、重なっているかどうかを検出するユニットである。



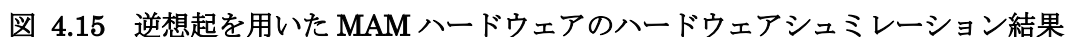
(a) 2 個の包含関係にあるパターン対 10 組



(b) 3 個の包含関係にあるパターン 10 組

図 4.14 相互相関想起実験に用いた記銘パターン対

逆想起を用いた MAM ハードウェアは、ハードウェア記述言語の VHDL により設計している。ハードウェアシミュレータを用いてクロック精度の動作確認および、処理能力の確認を行った。ここでは、ターゲットデバイスとして、汎用機器に良く用いられる Xilinx 社製の Virtex5(xc5vsx95t)を想定した。 $8 \times 8 = 64$ の 2 値のデータ(64bit)、核ユニット数は 32 個、図 4.14 に示す記銘パターンを用いて相互相関想起を行った。



次に本ハードウェアモデルとソフトウェアモデルとの想起時間を比較した。ソフトウェア処理には CPU Intel Core i5 2.8GHz の PC を用いた。表 4-2 は逆想起を用いた MAM のハードウェアモデルとソフトウェアモデルとの、1 回の想起時間を比較した表である。

表 4-2 逆想起を用いた MAM ハードウェアモデルとソフトウェアとの比較

モデル		最大動作周波数 (GHz)	処理速度 (μ sec)	
			2 個の 包含パターン	3 個の 包含パターン
ソフトウェア	従来の 逆想起	2.8	43.2	54.1
	簡単化した 逆想起	2.8	31.4	32.1
ハードウェア		0.14	0.11	0.11

表 4-2 の結果より、本ハードウェアモデルはソフトウェアモデルと比較して、約 280 倍以上もの高速化が図れることを確認した。

第5章 分割記録により計算量を抑えた逆想起を用いた

MAM

5.1 概要

逆想起を用いた MAM モデルでは、従来 MAM モデルの欠点であった、相互相関想起において、包含パターンが想起できないという問題を改善したことを示した。逆想起を用いた MAM は包含関係にあるパターンが入力され、前段想起で重複した核パターンを想起された場合でも、逆想起という手法により核パターンを 1 つに絞り込み、後段想起より入力パターンに対応する尤もらしいパターンの想起を行うことが可能である。このモデルは、包含パターンに対応することで、従来の MAM モデルと比較し、想起率を改善している。しかし、従来の MAM 同様、記録パターンの情報を記憶している記憶配列 M 、 W の構成に必要な空間計算量は、パターンのユニット数を n とすると、 $O(n^2)$ となる[4](図 5.1)。ここで、記録パターンがバイナリパターンとすると、記録パターン数は 2^n となる。そこで、従来の MAM モデルの問題点を解決した分割記録を用いた新しい MAM モデルが提案された[28]。このモデルでは、記憶配列 M 、 W に必要な空間計算量は、 $O(n^{1.5})$ まで削減できる。

しかしながら、分割記録を用いたモデルでは、記録パターンを分割し、それぞれの分割したパターンを記録するため、記録するパターンのユニット数が少なくなり、包含関係にあるパターンが増加する。そのため、逆想起を用いた MAM と比べ、記録可能なパターン数が減少してしまう。そこで、分割記録を用いたモデルに、逆想起を導入することで、従来の MAM と比較して空間計算量を削減しながらも、記録パターン数の減少を抑えることを検討する。

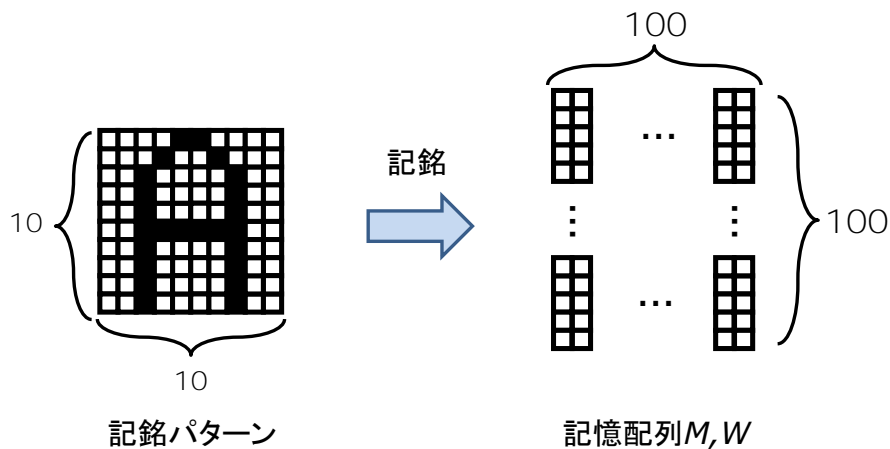


図 5.1 記憶配列のユニット数(100 ユニットの 2 値パターンの例)

5.2 分割記銘を用いた MAM

従来の MAM モデルにおいて、記銘パターンのユニット数が 256×256 である場合、空間計算量は $O(n^2)$ となるため、 $(256 \times 256)^2 = (2^{16})^2 = 2^{32} = 4(GB)$ ものメモリが必要である。MAM の記憶配列の演算では、記銘パターン数には依存せず、パターンのユニット数に依存する(図 5.2)。

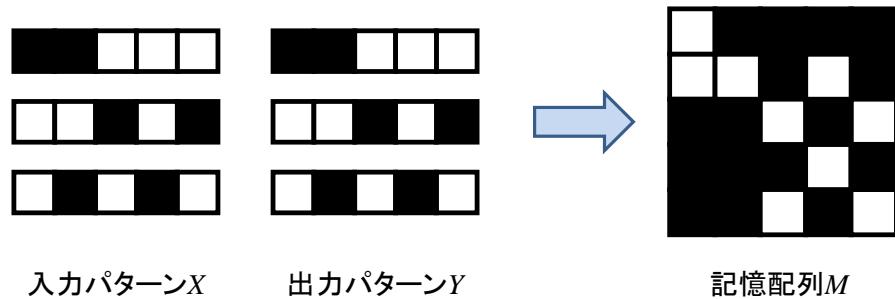


図 5.2 2 値、5 ユニット 3 組の自己相関想起における記憶配列 M

この問題を解決するために、パターンを記銘する際、パターンを分割し銘、複数の記憶配列 M 、 W を構成する、分割記銘を用いた MAM が提案された。従来の MAM では、記憶配列を構成する際、記銘パターンの全ユニットを用いて演算を行う。分割記銘を用いた手法では、記銘する際、パターンのユニットを複数のサブブロックに分割し、分割されたサブブロック毎に記憶配列を構成する。想起する場合には、入力パターンのユニットを、記銘させた際と同様にユニットを分割し、それぞれのサブブロックに対応した記憶配列を用いて想起を行う。最後に、サブブロックごとに想起したパターンを 1 つのパターンに結合し、出力する。

図 5.3 に分割記銘を用いた際の記銘過程、また図 5.4 に分割記銘を用いた際の想起過程を示す。

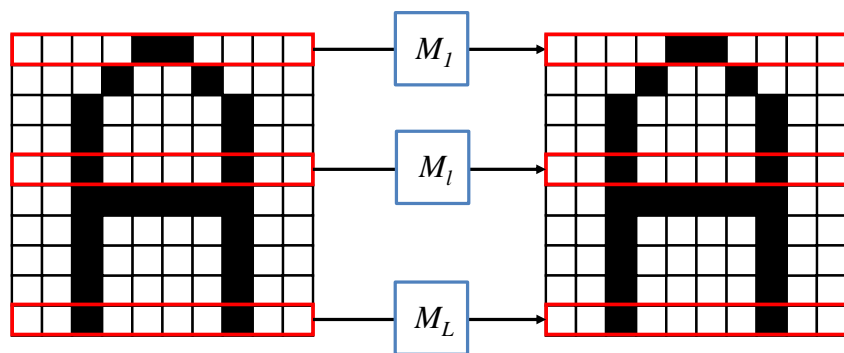
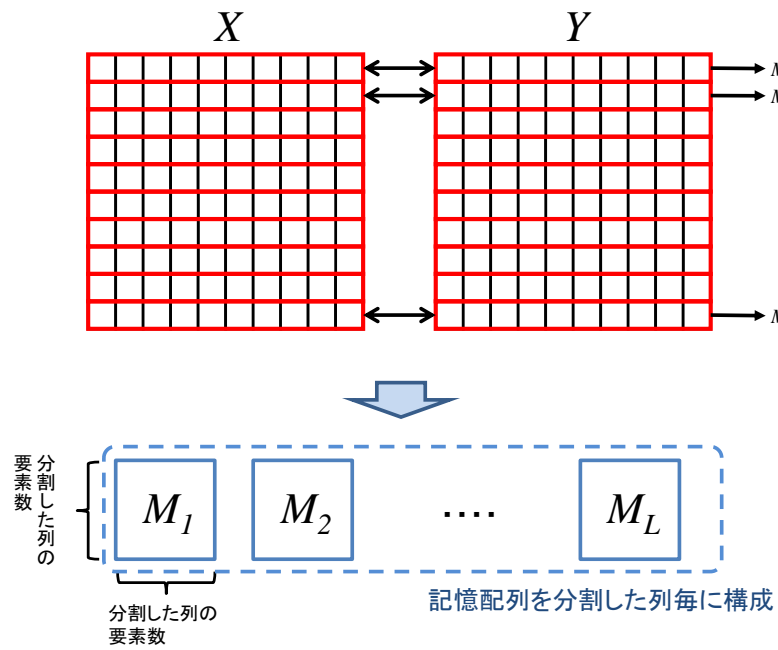
分割記銘を用いた MAM の記銘、および想起過程のアルゴリズムを以下に示す。

記銘

1. 各パターンを 1 つ目のサブブロックに分割し、その要素を集めた X を構成する。
2. X と、それに対応する出力 Y とのパターン (自己相関想起の場合 $X=Y$) を用いて、記憶配列 M 、 W を構成する。
3. 次のサブブロックに移り 2. を行う。
4. すべてのサブブロックを実行するまで繰り返す。

想起

1. 入力された X をサブブロックに分割する。
2. 1. のサブブロックを用いて M 、 W 、それぞれ演算を行う。
3. 出力された各サブブロックを一つのパターンに結合、 Y として出力する。



5.3 分割記録を用いた MAM の評価

5.3.1 完全想起率の記録パターン数依存性

分割記録を用いた MAM の性能を評価するため、記録パターン数、入力ノイズ率を変化させ、自己相関想起実験を行った。実験に用いた記録パターンは、“0” か “1” の 2 値をランダムに入力する。それぞれのパターンは $8 \times 8 = 64$ のバイナリユニットからなる。

図 5.5 に Ritter の MAM、図 5.6 に分割記録を用いた MAM モデルの完全想起率の記録パターン数依存性を示す。ここで、横軸が記録パターン数、縦軸が完全想起率を表している。

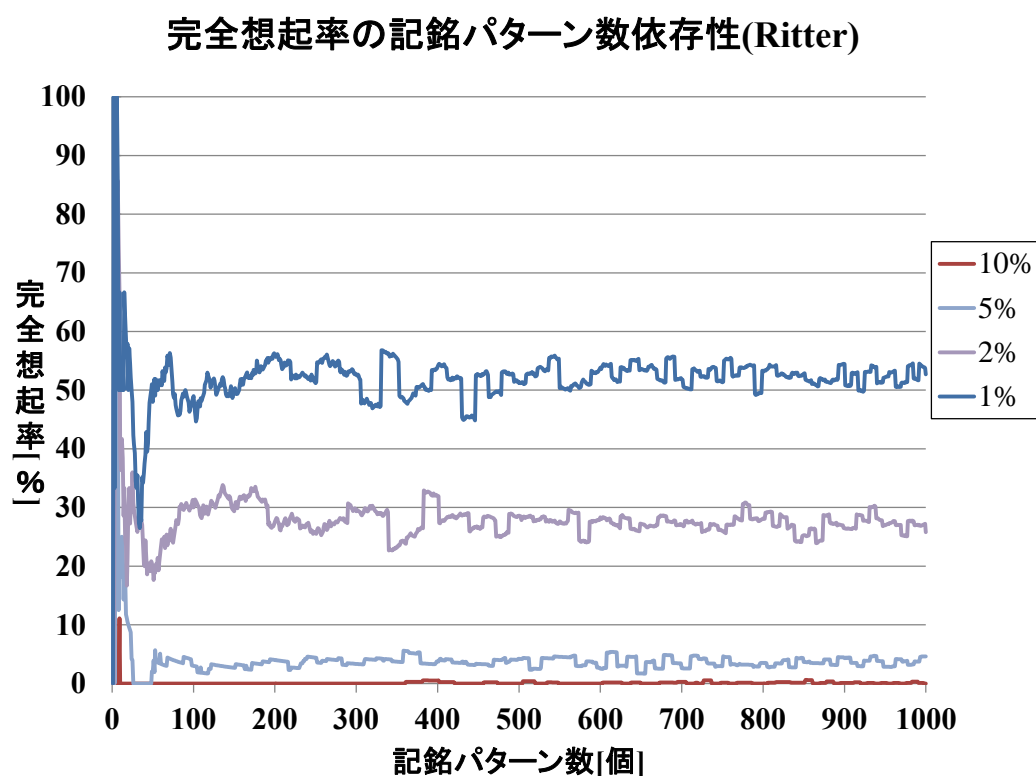


図 5.5 完全想起率の記録パターン数依存性(Ritter のモデル)

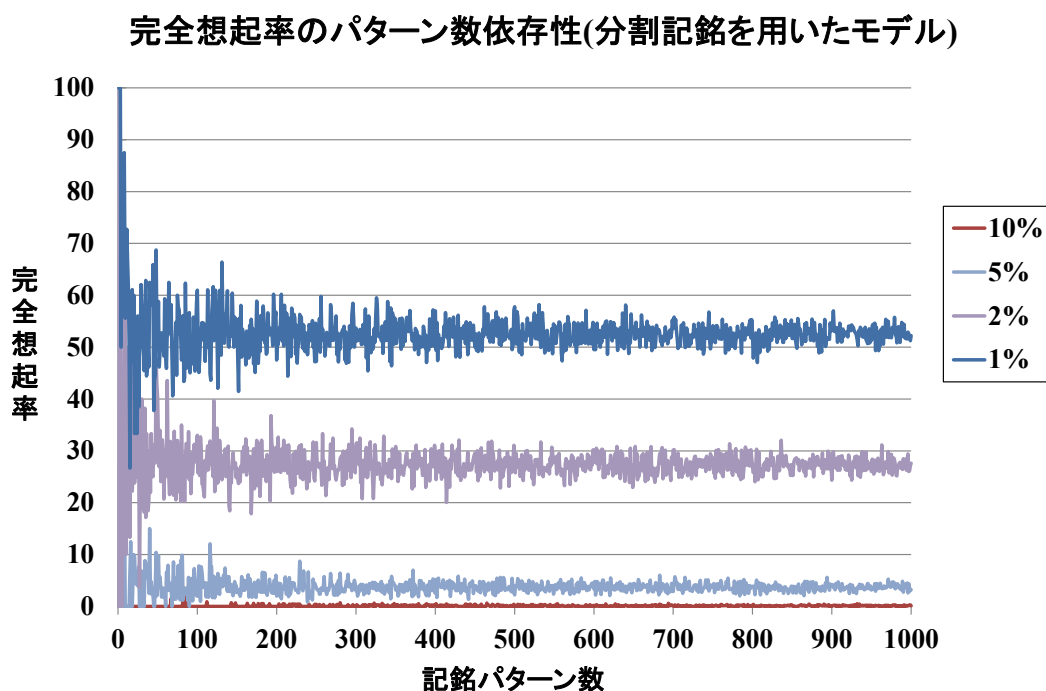


図 5.6 完全想起率の記録パターン数依存性(分割記銘を用いたモデル)

図 5.5 より、Ritter の MAM は、入力ノイズが 1[%]の場合、記録パターン数が増加すると、完全想起率は約 50%に収束している。入力ノイズが増加すると、完全想起率は低くなり、ノイズ率が 10[%]では正しい想起ができなくなることが分かる。図 5.6 では、入力ノイズが 1[%]の場合、完全想起率は約 50[%]に収束している。入力ノイズが増加すると、完全想起率が低下し、入力ノイズが 10[%]の場合、Ritter の MAM と同様、完全想起率が 0[%]に収束していることが分かる。このことから、分割記銘を用いた MAM は、Ritter の MAM と同等の性能を持っていることが分かる。したがって、分割記銘を用いた MAM は使用する空間計算量を削減しながらも、従来の MAM と同等の性能を持っていることが分かる。

次に、入力ノイズと完全想起率との関係を調べるため、記録パターン数を 1000[個]とし、自己相関想起実験を行った。図 5.7 に記録パターン数を 1000[個]とした場合の、分割記銘を用いたモデル、および Ritter のモデルの完全想起率を示す。

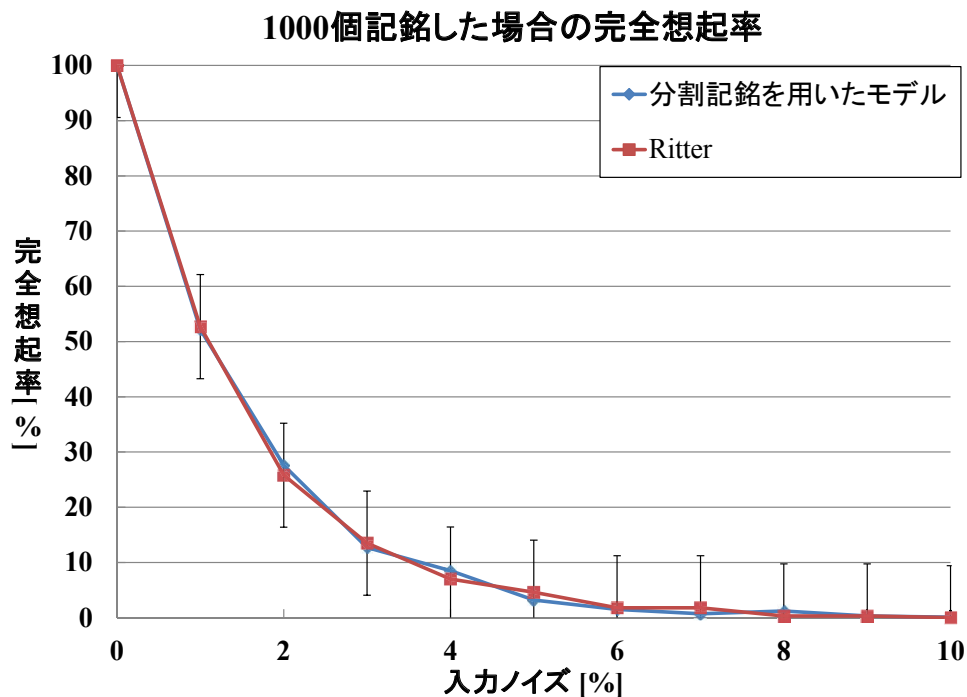


図 5.7 記憶パターン数を 1000 個とした場合のノイズ耐性

図 5.7 より、分割記憶を用いたモデルと Ritter のモデルを比較すると、ノイズ耐性はほぼ同等の性能ということが分かる。ここで、図 5.7 のエラーバーは、Ritter モデルの標準誤差を表している。分割記憶を用いたモデルでは、この誤差内に収まっていることがわかる。

5.3.2 記憶パターンから独立した核パターン構成法の導入

分割記憶を用いた MAM モデルに、記憶パターンから独立した核パターン構成法を導入したモデルの評価を行う。

この核パターンを導入することによって、一般ノイズ(付加、および欠落ノイズ)にも有効である。図 5.8 は、記憶パターンから独立した核パターンを用いた MAM モデルの結果である。また、図 5.9 には、分割記憶を用いたモデルに、独立した核パターン構成法を導入したモデルの結果を示している。記憶パターンから独立した核パターンを用いた MAM では、想起過程で、ノイズの影響をサブブロック内に抑えるため、入力パターンのユニットをサブブロックに分割し、それぞれのサブブロックから核パターン想起する。前段想起で想起された核パターンの多数決をとり、核パターンを 1 つに決定する。ここでは、分割数

を“1”とする(入力パターンの分割なし)。ここで、図 5.8、図 5.9 では、横軸は記銘パターン数、縦軸は完全想起率を表す。

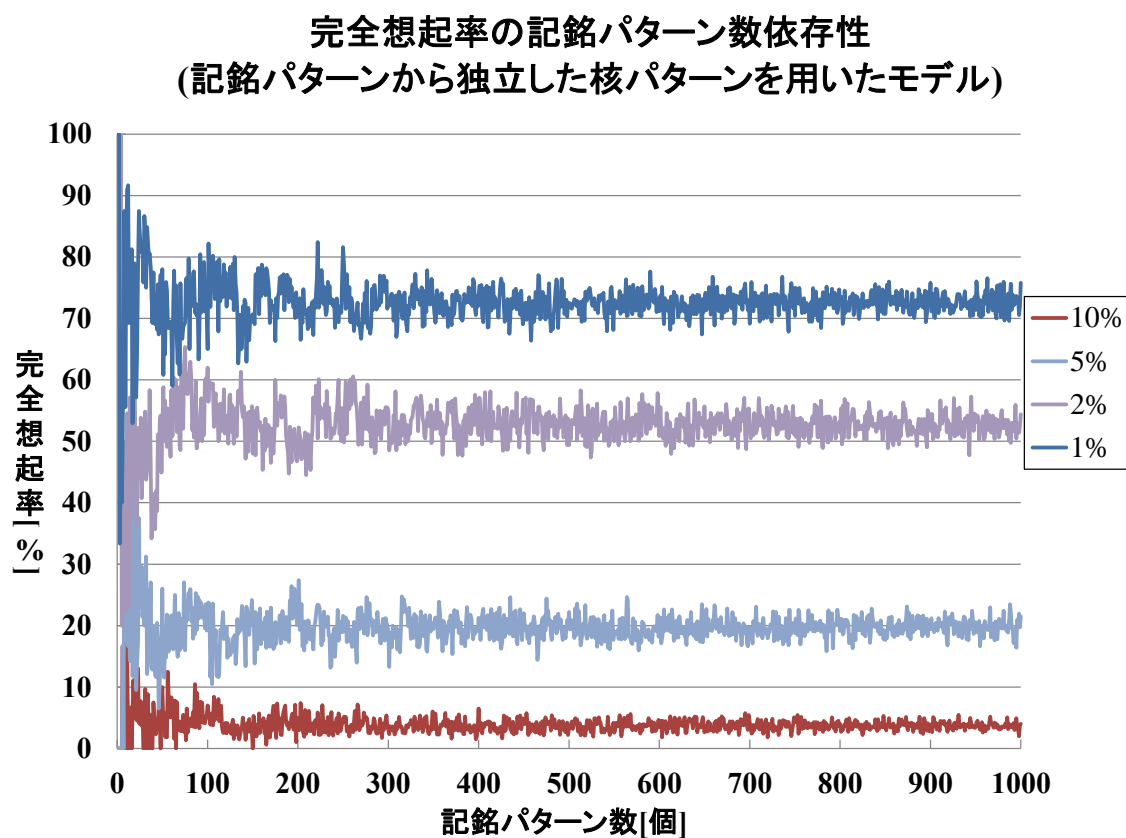


図 5.8 分割記銘を用いていないモデルにおける完全想起率の記銘パターン数依存性

完全想起率の記録パターン数依存性 (分割記録を用いたモデル)

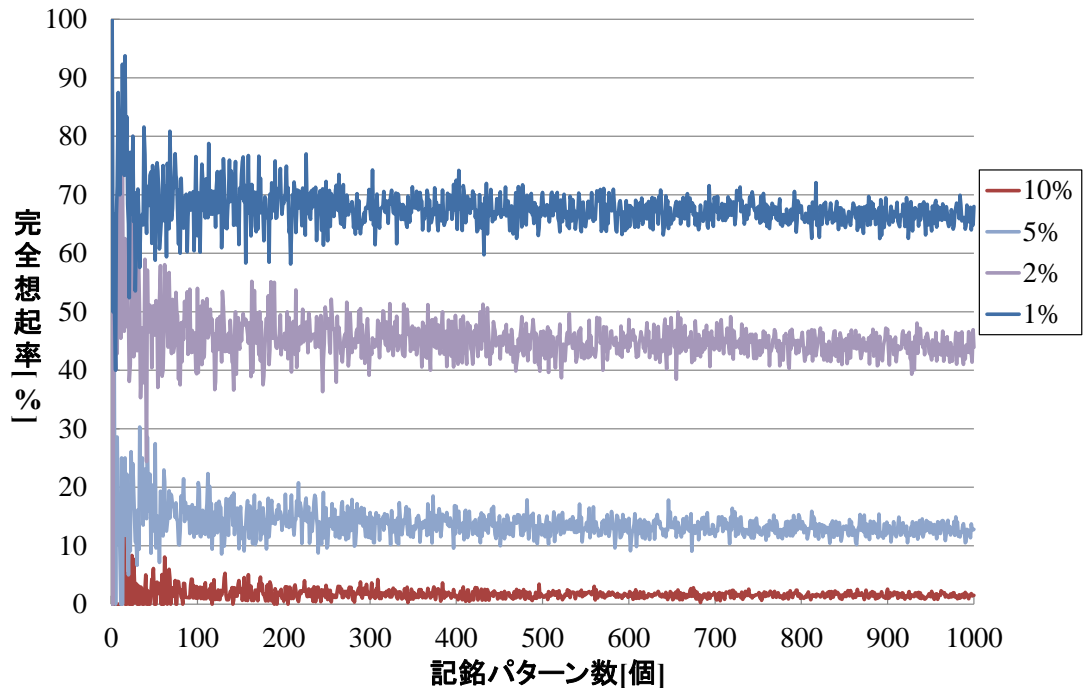


図 5.9 分割記録を用いたモデルにおける完全想起率の記録パターン数依存性

図 5.8、および図 5.9 より、核パターンを用いることで、核パターンを用いない MAM(図 5.5、図 5.6)と比較し、収束する完全想起率が高くなっている。ノイズ率が 1[%]の場合、核パターンを用いることで、完全想起率が約 70[%]に収束していることが分かる。

また、図 5.10 に、記録パターン数が 1000[個]の際の、記録パターンから独立した核パターン構成法を用いた MAM、および分割記録を用いたモデル(核パターン有り、無し)のノイズ耐性を示す。

1000個記録した場合の完全想起率

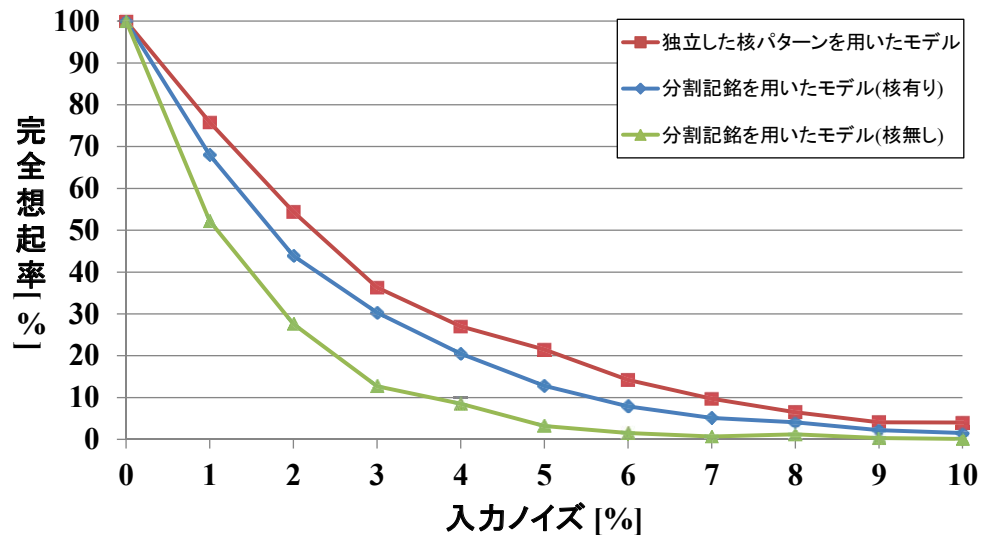


図 5.10 記録パターン数 1000 個の場合の独立した核パターンを用いたモデル、分割記録を用いたモデルの完全想起率

図 5.10 より、記録パターンから独立した核パターンを用いたモデルと比較し、分割記録を用いたモデルのノイズ耐性が低くなっていることが分かる。これは、分割記録を用いたモデルでは、パターンを分割して記録および想起を行うため、用いる記録パターンのユニット数が減少することで、包含関係にあるパターンが発生する確率が高くなるためであると考えられる。

次に、分割記録を用いたモデル、および記録パターンから独立した核パターンを用いたモデルの相互相関想起実験を行った。図 5.11 では、分割記録を用いたモデル、および記録パターンから独立した核パターンを用いたモデルでの、完全想起率のパターン数依存性を示している。ここで、記録パターンには、総ユニット数 64 のバイナリパターンを用いている。

完全想起率のパターン数依存性 (相互相関想起)

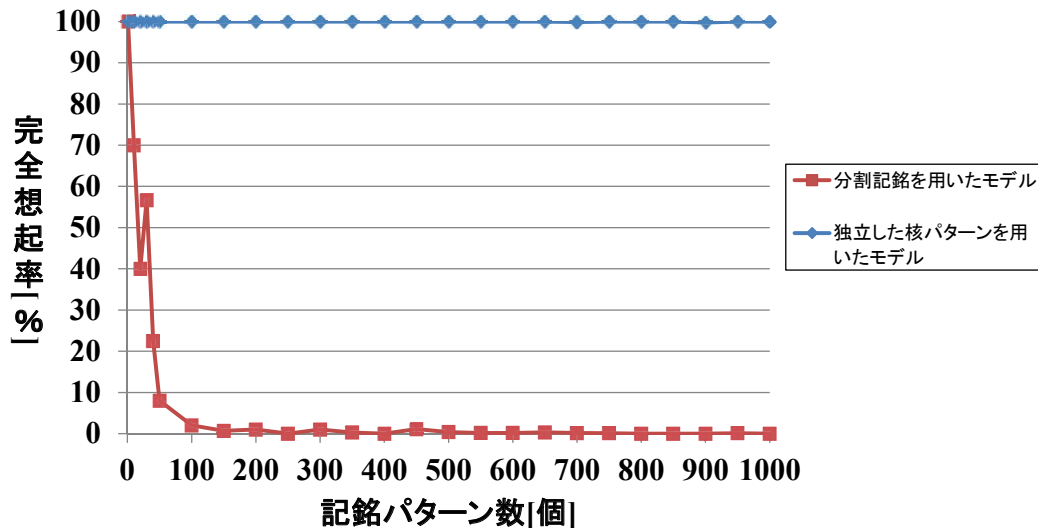


図 5.11 独立した核パターンを用いたモデルおよび分割記録を用いたモデルにおける相互相関想起を行った場合の完全想起率の記録パターン数依存性

図 5.11 より、入力ノイズ 0[%]の場合、記録パターンから独立した核パターンを用いたモデルでは、完全想起率がほぼ 100%となる。また、記録パターン数が 700 以上では、包含パターンが発生し、それにより正しく想起できない場合があった。分割記録を用いたモデルでは、記録パターン数が増加すると、完全想起率が急激に減少してしまう。これは、分割記録を用いたモデルでは、記録、想起に用いるパターンをサブブロックに分割し、そのサブブロックのユニット数が 8 ユニットとなるため、包含関係にあるパターンの発生する確率が高くなり、正しいパターンを想起できなくなることが考えられる。

5.4 逆想起手法の導入

分割記録を用いたモデルでは、連想に用いるパターンをサブブロックに分割し、それぞれの分割したパターンを記録するため、記録するパターンのユニット数が少なくなり、包含関係にあるパターンが増加する。そのため、逆想起を用いた MAM と比べ、記録可能なパターン数が減少してしまう。そこで、分割記録を用いたモデルに、逆想起を導入することで、従来の MAM と比較して空間計算量を削減しながらも、記録可能なパターン数の減少を抑えることを検討する。

分割記銘を用いたモデルへ、逆想起を導入した際の、記銘および想起過程を以下に示す。

記銘

1. 各パターンを 1 つ目のサブブロックに分割し、その要素を集めた X を構成する。
2. X と、それに対応する出力 Y とのパターン（自己相関想起の場合 $X=Y$ ）および、記銘パターンから独立した核パターン構成法で作成した核パターンを用いて、記憶配列 M 、 W を構成する。
3. 次のサブブロックに移り 2. を行う。
4. すべてのサブブロックを実行するまで繰り返す。

想起

1. 入力された X をサブブロックに分割する。
2. 1 のサブブロックを用いて M と演算を行い、核パターンを算出する。
3. 重複した核パターンが出力された場合、逆想起を用いて、記銘した入力パターンを逆に想起し、入力パターンと比較を行い、尤もらしいパターンに対応する核パターンを決定する。
4. 尤もらしい核パターンを用いて W と演算を行う。
5. 出力された各サブブロックを一つのパターンに結合、 Y として出力する。

分割記銘を用いた MAM へ、逆想起を導入することで、従来の MAM モデルと比較して、空間計算量を $O(n^{1.5})$ まで削減でき、逆想起を用いた MAM モデルと同等の記銘パターン数をもつ連想記憶モデルとなることが期待される。

第6章 結論

本論文では、記憶したパターンと入力されたパターンと比較を行い、より尤もらしいパターンを想起可能な、連想記憶モデルを考案し、連想記憶モデルの実応用へ向けて、そのハードウェア化を行った。従来の連想記憶モデルでは、相互相関想起において包含関係にあるパターンは正しく想起できないという問題点を解決するために、包含パターンが想起可能な新しい MAM モデルを提案し、その有効性を確認した。逆想起を用いた手法では、3 個の包含関係にあるパターンでも想起可能であることを示した。また、逆想起を用いた MAM では、非常にシンプルな演算を用いていること、また、並列に処理が可能なアーキテクチャとなっているため、非常にハードウェア化に適していることからハードウェアモデルを提案した。我々は、逆想起に用いている演算を、よりシンプルな演算に簡略化することで、よりハードウェア化に適したモデルを提案し、ハードウェア化を行った。本論文では、形態学的連想記憶モデルの工学的応用に向けて、下記の新しい手法を提案し、その効果を確認した。

・ 記銘パターンから独立した核パターンを用いた MAM

Ritter の MAM を改良し、記銘パターンから独立した核パターンを用いた MAM を提案した。Ritter のモデルの記銘パターンの一部で核パターンを構成する方法は、記銘パターン数が増えると核パターン設計が難しくなり、核パターンに相当するユニットにノイズが載ると想起できなくなる。そこで我々は、記銘パターンから独立した核パターン構成法を考案し、核パターンの設計の単純化を行った。また、核パターンは記銘パターンとは独立して構成するため、核パターンに相当するユニットにノイズが載ることがなく、ノイズ耐性を向上することができた。

また、この記銘パターンから独立した核パターンを用いた MAM のハードウェアモデルを提案した。データが入力されてから 10 クロックで想起可能であり、この MAM ハードウェアモデルを 40MHz で動作させた場合と、ソフトウェア処理との比較を行い、104 倍の高速化が図れることを確認した。

・ 逆想起を用いた MAM

従来の連想記憶モデル同様、包含関係にあるパターンに対しては、正しく想起できない MAM の問題点を解決した、包含パターンが想起可能な逆想起を用いた MAM モデルを提案した。従来の MAM では、あるパターンに完全に含まれているパターン(包含関係にあるパターン)が記銘されると、相互相関想起において正しく想起できなかった。逆想起を用いた MAM では、核パターンから記銘パターンを逆想起し、入力と比較させ、核パターンを一つに絞ることにより正しく想起できることを示し、また、ノイズ耐性が向上したことを確認

した。

また、逆想起を用いた MAM では、非常にシンプルな演算を用いていること、並列に処理が可能なアーキテクチャとなっているため、ハードウェア化に適した連想記憶モデルであるため、このモデルのハードウェア化を行った。ハードウェア化を行う前にまず、逆想起に用いている演算を、よりシンプルな演算に簡略化することで、よりハードウェア化に適したモデルを提案した。逆想起の演算の簡単化を行うことで、前段想起において重なった核パターンが想起された場合でも、核パターンを分離することなく、一回の逆想起で記銘パターンを想起できることを示し、従来の逆想起と比較して、1 回の想起時間を 25%以上削減できることを確認した。

この簡単化した逆想起を用いた MAM ハードウェアモデルを提案することにより、より高速に想起が可能な MAM ハードウェアの開発を行った。入力に対応する出力を 15 クロックで想起できていることを示した。この逆想起を用いた MAM ハードウェアモデルは汎用 PC (Intel Core i5 2.8GHz) と比較して、約 280 倍以上もの高速化が図れることを確認した。

本論文で提案した逆想起を用いた MAM ハードウェアモデルは、たとえ複数のパターンが重なったパターンが入力されても、尤もらしいパターンを高速に想起することが可能である。実応用を考えた場合、記銘パターンが増加すると、包含するパターンや、類似パターンの存在も増えていくことが予想される。そのような状況下でも、我々が提案した MAM ハードウェアモデルを応用することにより、尤もらしいパターンを高速に想起することが可能である。今後、実応用としては、パターン認識、パターン修復、ノイズ除去等に用いられることを期待している。手書き文字や写真画像等、実環境化から特徴量を求め、その特徴量を用いた想起等、実用的な連想記憶システムの構築を検討していく。また、自律ロボットの画像認識等へ、MAM ハードウェアが用いられることを期待している。

謝辞

本研究を進めるにあたり、多忙な中幾度と相談に乗って頂き、適切な御助言を賜り、丁寧に指導して下さった本学大学院生命体工学研究科神酒勤教授、及び田向権准教授に謹んで感謝の意を表します。また、本研究を進める過程で、有益な御意見ならび様々な御協力を頂いた本学大学院生命体工学研究科神酒研究室の皆様にも心より感謝致します。

参考文献

- [1] S. Amari, "Statistical Neurodynamics of Associative Memory," Neural Networks, 1, pp.63-73, 1988.
- [2] D.J. AMIT, "Storing infinite numbers of patterns in a spin-glass model of neural networks," Physical Review Letters, 55, pp.1530-1533, 1985.
- [3] R.J. McEliece, "The capacity of the Hopfield associative memory," IEEE Trans. Information Theory, IT-33, 4, pp.461-482, 1987.
- [4] G.X. Ritter, P. Sussner, and J.L. Diaz-de-Leon, "Morphological associative memory," IEEE Trans. Neural Networks, Vol.9, No.2, pp.281-293, 1998.
- [5] G.X.Ritter, J.L.Diaz-de-Leon and P.Sussner, Morphological bidirectional associative memories, Neural Networks, 12(6), pp.851-867, 1999.
- [6] M.Wang, S.C.Chen, Enhanced FMAM based on empirical kernel map, IEEE Trans. Neural Networks, 16(3), pp.557-564, 2005.
- [7] P.Sussner, M.E.Valle, Implicative fuzzy associative memories, IEEE Trans. Fuzzy Syst., 14(6), pp.793-807, 2006.
- [8] M.E.Valle, P.Sussner, A general framework for fuzzy morphological associative memories, Fuzzy Sets Syst., 159(7), pp.747-768, 2008.
- [9] P.Sussner, M.E.Valle, Classification of fuzzy mathematical morphologies based on concepts of inclusion measure and duality, J. Math. ImagingVision, 32(2), pp.139-159, 2008.
- [10] P.Sussner, M.E.Valle, Gray-scale morphological associative memories, IEEE Trans. Neural Networks, 17(3), pp.559-570, 2006.
- [11] M.E.Valle, P.Sussner, Fuzzy associative memories based on uninorms, in:

Proceedings of the IEEE International Conference on Computational Intelligence and Proceedings of FUZZ-IEEE, HongKong, China, pp.1582–1589, 2008.

- [12] M.E.Valle, P.Sussner, Fuzzy associative memories from the perspective of mathematical morphology, in: Proceedings of the IEEE International Conference on Fuzzy Systems, London, UnitedKingdom, pp.829–834, 2007.
- [13] P.Sussner, M.E.Valle, Fuzzy associative memories and their relationship to mathematical morphology, Handbook of Granular Computing, W.Pedrycz, A.Skowron, V.Kreinovich, (Eds.), Wiley-InterscienceNewYork, NY, USA, (Chapter 33).
- [14] M.Nachtegaele, P.Sussner, T.Melange, E.E.Kerre, Fuzzy set theory in mathematical morphology: from tool to uncertainty model, in: Proceedings of WMSCI2009, 13th WorldMulti-Conference on Systemics, Cybernetics and Informatics, Orlando, FL, USA, vol.IV, pp.149-155, 2009.
- [15] P.Sussner, M.E.Valle, Morphological and certain fuzzy morphological associative memories with applications in classification and prediction, computational intelligence based on lattice theory, in: V.G.Kaburlasos, G.X. Ritter(Eds.), Series: Studies in Computational Intelligence, vol.67, Springer Verlag, pp.149–173, 2007.
- [16] N.Q.Feng, Y.H.Qiu, F.Wang, Y.Q.Sun, A unified frame work of morphological associative memories, in: D.-S.Huang, K.Li, G.W.Irwin(Eds.), Proceedings of the 2nd International Conference on Intelligent Computing, Kunming, China, pp.1–11, 2006.
- [17] N.Q.Feng, C.H.Liu, C.P.Zhang, et al., Research on the framework of morphological associative memories, Chin.J.Comput, 33(1), pp.157–166, 2010.
- [18] N.Q.Feng, X.Z.Cao, S.J.Li, et al., A new method of morphological associative memories, in: Proceedings of the 5th International Conferenceon Intelligent Computing, Ulsan, SouthKorea, pp.407–416, 2009.
- [19] N.Q.Feng, S.X.Wang, J.C.Xu, et al., Research on the dual kernel based on the unified framework of morphological associative memories, in: Proceedings of 2010

International Conference on computer application and system modeling, Taiyuan, China, vol.II, pp.643–648, 2010.

- [20] G.X.Ritter, G.Urcid, M.S.Schmalz, Autonomous single-pass endmember approximation using lattice auto-associative memories, *Neurocomputing*, 72(10–12), pp.2101–2110, 2009.
- [21] 佐伯崇, “形態学的連想記憶モデルのネットワーク構造に関する研究,” 博士論文(九州工業大学), 2008.
- [22] H. Harada, T. Saeki and T. Miki, A morphological associative memory employing a stored pattern independent kernel image and its hardware model, *IWCIA 2009*, pp.219-224, 2009.
- [23] H. Harada and T. Miki, A morphological associative memory employing a reverse recall, *17th International Conference on Neural Information Processing, Part II*, pp.321-328, 2010.
- [24] E. Guzman, S. Alvarado, O. Pogrebnyak and C. Yanez, Image Recognition Processor based on Morphological Associative Memories, *CERMA 2007*, pp.260-265, 2007.
- [25] H. Harada and T. Miki, Recall Time Reduction of a Morphological Associative Memory Employing a Reverse Recall, *18th International Conference on Neural Information Processing, Part III*, pp.332-339, 2011.
- [26] H. Harada and T. Miki, A Hardware Model of a Morphological Associative Memory Employing a Simplified Reverse Recall, *ICIC Express Letters*, Vol.6, No.3, pp.833-838, 2012.
- [27] H. Harada and T. Miki, A Morphological Associative Memory Employing Simplified Reverse Recall, and Its Hardware Model, *IJICIC*, Vol.10, No.5, 2014.
- [28] 古賀謙介, “分割記銘手法を用いた形態学的連想記憶モデル,” 修士論文(九州工業大学), 2014.
- [29] K. Nakano, “Associatron-a model of associative memory,” *IEEE Trans. Syst., Man,*

Cybernetics 12, Vol.SMC-2, pp.380-388, 1972.

- [30] T. Kohonen, "Correlation matrix memories," IEEE Trans. Computers, C-21, pp.353-359, 1972.
- [31] J.A. Anderson, "A simple neural network generating an interactive memory," Math. Biosciences, 14, pp.197-220, 1972.
- [32] J.J. Hopfield, "Neural network and physical systems with emergent collective computational abilities," Proc. of the national Academy of Sciences of the United States of America 79 (8), pp.2554-8, 1982.
- [33] W.S. McCulloch, and W.H. Pitts, "A logical calculus of the ideas immanent in nervous activity," Bullet. Math. Biophysics, 5, pp.115-133, 1943.
- [34] J.J. Hopfield and D.W. Tank, " "Neural" computation of decisions in optimization problems," Biological Cybernetics, 52, pp.141-152, 1985.
- [35] Y. Hirai, "A model of human associative processor (HASP)," IEEE Trans. Sys., Man, Cyb. , SMC-13, No.5, pp.851-857, 1983.
- [36] 川村正樹, 平井有三, "連想記憶回路 HASP の記憶容量の解析," 電子情報通信学会論文誌, D-II, Vol.J78, No.11, pp.1692-1700, 1995.
- [37] 川村正樹, 岡田真人, 平井有三, "1 対多の連合を記憶する相関型連想記憶モデルの解析," 電子情報通信学会論文誌, D-II, Vol.J81, No.6, pp.1336-1344, 1998.
- [38] D.H. Ackley, G.E. Hinton, and T.J. Sejnowski, "A learning algorithm for Boltzmann machines," Cognitive Science, Vol.9, pp.147-169, 1985.
- [39] G.L. Bilbro, W.E. Snyder, S.J. Garnier, J.W. Gault, "Mean field annealing: a formalism for constructing GNC-like algorithms," Neural Networks, IEEE Trans. Neural Networks, Vol.3 (1), pp.131-138, 1992.
- [40] G. E. Hinton, "Deterministic Boltzmann learning performs steepest descent in weight-space," Neural Computation, 1, pp.143-150, 1989.

- [41] C. Peterson, and J. Anderson, “Mean field theory learning algorithm for neural networks,” *Complex systems* 1, pp.995–1019, 1987.
- [42] P. Dayan, G. E. Hinton, and Radford M. Neal, “The Helmholtz machine,” *Neural Computation*, 7, pp.889-904, 1995.
- [43] B. Kosko, “Bidirectional Associative Memories,” *IEEE Trans. Systems, Man and Cybernetics*, Vol.18, No.1, pp.49-60, 1988.
- [44] B. Kosko, “Adaptive Bidirectional Associative Memories,” *Applied Optics*, Vol. 26, No.23, pp.4947-4960, 1987.
- [45] M. Hagiwara, “Multidirectional associative memory,” *International Joint Conference on Neural Networks*, I, pp.3-6, 1990.
- [46] M. Hattori, and M. Hagiwara, “Quick learning for Multidirectional Associative Memories,” 1995 *IEEE International Conference on Neural Networks*, pp.1949-1954, 1995.
- [47] 小林正樹, 服部元信, 山崎晴明, “多方向連想記憶の記憶容量,” *電子情報通信学会技術研究報告*, No.NC2003-33, pp.67-72, 2003.
- [48] 小林正樹, 服部元信, 山崎晴明, “隠れ層をもつ多方向連想メモリ,” *電子情報通信学会論文誌*, Vol.J84-DII, No.7, pp.1495-1502, 2001.
- [49] M. Kobayashi, M. Hattori, H. Yamazaki, “Multidirectional Associative Memory with a Hidden Layer,” *Systems and Computers in Japan*, Vol.33, No.6, pp.1-9, 2002.
- [50] M. Morita, S. Yoshizawa, and K. Nakano, “Memory of correlated patterns by associative neural networks with improved dynamics,” *Proc. of the 1990 International Neural Network Conference*, Paris, 2, pp.868-871, 1990.
- [51] 森田昌彦, 吉澤修治, 中野馨, “非単調ダイナミクスを用いた構造をもつパターンの連想記憶,” *電子情報通信学会論文誌*, Vol.J75-D-II, pp.1884-1891, 1992.

- [52] M. Morita, "Associative memory with nonmonotone dynamics," *Neural Networks*, 6, pp.115-126, 1993.
- [53] S. Yoshizawa, M. Morita, and S. Amari, "Capacity and spurious memory of associative memory using a non-monotonic neuron model," *Proc. of the 1992 International Conference on Artificial Neural Networks*, Brighton, 2, pp.445-448, 1992.
- [54] S. Yoshizawa, M. Morita, and S. Amari, "Capacity of associative memory using a nonmonotonic neuron model," *Neural Networks*, 6, pp.167-176, 1993.
- [55] K. Aihara, T. Takabe, and M. Toyoda, "Chaotic neural networks," *Physics Letters A*, Vol.144, No.6, 7, pp.333-340, 1990.
- [56] 合原一幸, "ニューラルシステムにおけるカオス," 東京電気大学出版局, 1993.
- [57] 長名優子, 服部元信, 萩原将文, "カオス双方向連想メモリ," 電気学会論文誌, Vol.116-C, No.7, pp.741-747, 1996.
- [58] Y. Osana and M. Hagiwara, "Separation of superimposed pattern and many-to-many associations by chaotic neural networks," *IJCNN*, Vol.1, pp.514-519, 1998.
- [59] 今井久夫, 長名優子, 萩原将文, "カオスアナログ連想メモリ," 電子情報通信学会論文誌, Vol.J85-DII, No.3, pp.514-522, 2002.
- [60] D.E. Rumelhart, J.L. McClelland, and PDP Research Group, "Parallel Distributed Processing," the MIT Press, Cambridge, Massachusetts, 1986.
- [61] S.B. Chan and J.L. Michael, "Memory capacity of artificial neural networks with high order node connections," *Proc. IJCNN*, pp.207-216, 1998.
- [62] 屋附秀司, 宮島廣美, 村島定行, "積結合をもつ神経回路網による連想記憶能力," 電子情報通信学会論文誌, Vol.J79-DII, No.11, pp.1929-1939, 1996.
- [63] M. Mézard, J.P. Nadal, G. Toulouse, "Solvable models of working memories," *J.*

Physique, 47, pp.1457-1462, 1986.

- [64] 岡田真人, 三村和史, 倉田耕治, “指数忘却をもつ連想記憶モデル—統計神経力学による解析—,” 電子情報通信学会論文誌, Vol.J77-DII, No.6, pp.1178-1180, 1994.
- [65] 水野尚, “連想記憶機能を実現するニューラルネットワークモデル,” 電子情報通信学会論文誌, Vol.J76-DII, No.7, pp.1463-1467, 1993.
- [66] D.J. Amit, H. Guyfreund, and H. Sompolinsky, “Information storage in neural networks with low levels of activity,” Phys. Rev. A, 35, pp.2293-2303, 1987.
- [67] S. Amari, “Characteristics of Sparsely Encoded Associative Memory,” Neural Networks, 2, pp.451-457, 1989.
- [68] 岡田真人, 三村和史, 倉田耕治, “相関型連想記憶モデルのスパースコーディングと素子のしきい値,” 電子情報通信学会論文誌, Vol.J77-DII, No.9, pp.1934-1936, 1994.
- [69] J.J. Hopfield, “Neurons with grabbed response have collective computational properties like those of two state neurons,” Proc. National Academy Sciences, 81, pp.3088-3092, 1982.
- [70] G. Matheron, “Random Sets and Integral Geometry,” John Wiley & Sons, New York, 1975.
- [71] J. Serra, ed., “Image analysis and mathematical Morphology,” Academic Press, 1982.

研究業績

投稿論文

1. Hidetaka Harada, Tsutomu Miki, A Morphological Associative Memory Employing Simplified Reverse Recall, and Its Hardware Model, International Journal of Innovative Computing, Information and Control, Vol.10, No. 5, 2014.
2. Hidetaka Harada, Tsutomu Miki, A Hardware Model of a Morphological Associative Memory Employing a Simplified Reverse Recall, ICIC Express Letters, Vol.6, No.3, pp.833-838, 2012.

国際会議

1. Hidetaka Harada, Takashi Saeki, Tsutomu Miki, A Morphological Associative Memory Employing A Stored Pattern Independent Kernel Image and Its Hardware Model, IWCIA 2009, pp.219-224, 2009.
2. Hidetaka Harada, Tsutomu Miki, A morphological associative memory employing a reverse recall, 17th International Conference on Neural Information Processing, Part II, pp.321-328, 2010.
3. Hidetaka Harada, Tsutomu Miki, Recall Time Reduction of a Morphological Associative Memory Employing a Reverse Recall, 18th International Conference on Neural Information Processing, Part III, pp.332-339, 2011.