自己増殖オートマトンネットワークの開発と酵素機能解析への応用に関する研究

東京大学大学院 農学生命科学研究科
応用生命工学専攻 生物情報工学研究室

久原　泰雄

①

自己増殖オートマトンネットワークの開発と酵素機能解析への応用に関する研究

Development of Self-Reproducing Automata Network And Application to Analysis of Enzyme Functions

東京大学大学院 農学生命科学研究科
応用生命工学専攻 生物情報工学研究室
久原 泰雄

Department of Biotechnology, The University of Tokyo
Bioinformation Engineering Laboratory
Yasuo Kuhara

目次

第 1 章

緒言

脳の構造や動作についての知見を基に神経回路モデルを計算機上に構築する動きは，1940 年代の McCulloch と Pitts の閾値素子や D.Hebb のシナプス強化則に始まった．それ以来，様々な Neural Network が研究されてきた．

たびたび脳と対比されるのが計算機である．プログラムストア型計算機は，問題を Algorithm として記述し，分解された個々の命令を直線的に実行するマシンモデルであるが，ハードウェアの目覚しい進歩によって，高速，大量，および正確な情報処理が可能となった．これに対して，神経回路モデルへの期待は，経験的知識による学習能力であり，厳密さを要求しない柔軟な処理や素子の並列動作また自己組織が特長となっている．

しかしながら，学習対象のデータが多くなるとネットワークの規模が大きくなり，計算時間が増大し，収束性も低下する．学習速度を上げるための試みが数多く提案されてきたが，学習環境が動的に変化すると，最適なネットワークの決定に困難が伴い，学習能力を保持することが難しい．このような問題点はハードウェアの改良によってある程度まで補われてきた．超高度並列計算機を実現することも可能となってきたが，従来の Algorithm に基づく手法では並列計算機はかえって効率が悪くなる．脳が低速な素子で構成されながら，効率の良い学習をするという事実は，新しいマシンモデルの構築方法の必要性を示唆している．

本論文では，神経回路の情報処理の次のような特徴に注目した．(1) 膨大な数の低速なユニットが互いに連結してネットワークを形成し，相互に情報交換する．(2) 個々のユニットは，各々の変移則に従って並列に動作する．(3) ユニットの増殖によってネットワーク構造を変化させながら処理を行う．(4) 集団内における遺伝操作によって個体が変化する．この分析に基づいて新しいマシンモデルである自己増殖オートマトンネットワーク（Self-**Repro**ducing Automata **Net**work：以下 ReproNet と記す）を提案した．実際にこのマシンモデルを用いて，学習中にデータが動的に

変化する環境下において柔軟に適応して構造を変化させるネットワークを構築した．また，種々のパターン認識の実例によって，学習能力を検証した．

経験的知識を用いた学習を特色とするマシンモデルには，対象とする問題を包含する大量の学習データが必要である．この点で，タンパク質／核酸の分野では，アミノ酸や塩基の膨大な配列データが蓄積されており，適用分野としては大きな可能性を秘めている．実際に，1988 年には Qian と Sejnowski はタンパク質の 2 次構造予測に Neural Network を適用して，従来の Chou と Fasman の方法より精度の高い予測が可能であることを報告した．以来，Neural Network を核酸，タンパク質分野へ応用した例が数多く報告されてきた．これは予め大量のデータを入力して学習を済ませておけば，1 次配列を入力するだけで瞬時に回答が得られ，ホモロジー検索などの手続きが不要であるという Neural Network の有用性が要因となっている．

本論文では，ReproNet を使用して，酵素の機能予測を試み，その有用性を検討した．酵素は，触媒として働く作用を選択する性質が強い．この酵素の特異性は基質と酵素の構造上の相互関係によって発揮される．このことは，酵素のアミノ酸 1 次配列上の局所的な情報を用いて，酵素機能の分類が可能であることを示唆している．そこで ReproNet を使用して，酵素のアミノ酸 1 次配列を入力し，EC 番号（1：酸化還元酵素，2：転移酵素，3：加水分解酵素，4：除去付加酵素，5：異性化酵素，6：合成酵素）を出力する機能予測システムを構築した．結果として，ホモロジー検索よりも有効な予測手法であることを示した．さらに，ReproNet の 1 次配列に対する出力値を解析し，酵素の特徴的な部位との比較検討を行い，酵素機能部位の抽出および予測に関する可能性を論じた．

第 2 章

総論

2.1 マシンモデルの概念

ヒトが行う知的な情報処理には大きく 2 つに分類できる．1 つ目としては，問題を Algorithm[1]として記述し，分解された個々の命令を順次実行するモデルである．Turing はヒトが行う論理的な処理過程として Algorithm を取り上げ，それを実行するモデルとして，Turing マシンを構成した．Turing マシンは Algorithm を記述するための数学的モデルとして知られている．これが直線的な情報処理を基礎とするプログラムストア型計算機の基本原理である．2 つ目としては，神経回路のように，単純な要素を多数結合して，並列の相互作用によって情報処理を実現する．問題の手続き化や論理的な厳密性は要求されないが，学習と自己組織化によって環境に適用して自己を調整する能力や柔軟な情報処理に優れている．これらは，Neural Network に代表される神経回路モデルによって構成されている．以下にこれらの研究の流れを述べる．

2.1.1 Turing マシン

1943 年から 1946 年の期間に，ペンシルバニア大学でアメリカ陸軍の援助の下に，主として弾道計算に利用する計算機 ENIAC の開発が行われた [1]．演算論理制御だけでなく，高速記憶装置にも真空管が用いられていたので，ENIAC は 18,000 本もの真空管からなっていた．個々の特定の問題のためのプログラミングは手で行われた．すなわち，解こうとしている問題に必要な演算装置の各々のプログラム制御の機械スイッチ，ケーブルによるこれらのプログラム制御間の接続，および関数表のスイッチの設定である．これがいわゆる第一世代計算機である．計算機はトランジ

[1]ある問題を解く際に，必ず正しい答えを出すことができる定まった手続きを **Algorithm** と呼ぶ．ここでいう手続きとは機械的に実行可能な有限の命令の列である．

スタを素子とした第二世代，LSI を素子とした第三世代，超 LSI を素子とした第四世代，そして人工知能を目標とした第五世代へと発展していった．ENIAC から半世紀経過した今日，計算機は個人で所有するパーソナルコンピュータから高度な科学技術計算を必要とするスーパコンピュータまで，様々な形態で使用されているが，これらはすべて Turing マシンという仮想的な機械に集約できる．

Turing マシンの基本型は，有限制御部，有限個の記号が記されたテープ，テープをアクセスするヘッドで構成されている（図 2.1）．制御部は有限個の内部状態をもつ．制御部はテープに対して左右に移動し，またテープの内容を書き換えることができる．Turing マシンに対する命令は次のように与えられる．

$$Q_i S_j \rightarrow S_k D Q_l \tag{2.1}$$

ここで，$S_0, S_1, \cdots, S_m$ は有限個のテープ記号，$Q_0, Q_1, \cdots, Q_n$ は有限個の内部状態，$D : \{L, R\}$ は制御部の移動，L : は左へ移動 R : は右へ移動である．これは，「Turing マシンの制御部が状態 Q_j で記号 S_j を読んだとき，記号 S_k に書き換え，D 方向に移動し，状態 Q_l になる．」ことを意味する．この式は，(i, j) の組に対して機能表 2.1によって定義されている．

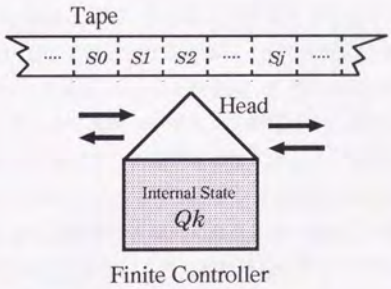


図 **2.1** Turing マシン

ある論理的な過程を Algorithm で表すことができるならば，その機械的な命令の列を Turing マシンの動作に置き換えることができる．したがって，Algorithm で記述できるすべての論理的な過程は Turing マシンで実行可能であり，現在の計算機で解決可能であるといえる．

表 2.1 機能表

	S_0	$\cdots$	S_j	$\cdots$	S_m
Q_0					
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Q_i			$S_k D Q_l$		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Q_n	$\cdots$		$\cdots$		$\cdots$

2.1.2 Neural Network の概念

神経回路に着目したモデルは古くから提唱されており，McCulloch と Pitts[2] は 1943 年に神経細胞をモデルとしたしきい素子を発表している．Neural Network の初期のモデルとしては 1958 年の Rosenblatt[3] による Perceptron が有名である．これは，3 層に配置された神経素子（ユニット）が一方向に信号を伝達していくネットワークで、単一の出力ユニットに外部から教師信号を与えて結合している前層との間の重みを変化させることにより学習が可能なモデルであった．その後、1986 年に Rumelhart と McClelland[4] は，3 層以上の階層型 Neural Network で，全てのユニット間の重みを変化させる逆伝播法を考案し、現在の主流となっている．1985 年には，Hopfield[5] らは Neural Network を使用してエネルギー関数を最適化する方法を考案した．これによって巡回セールスマン問題などの組合せ最適化問題を求めることができる．これに加えて，物理学者の参入があり，確率的な動作をするボルツマンマシン [6] やシミュレーテッドアニーリングの手法が提唱された．一方，いくつかの組合せ問題を並列回路網で近似的に高速に解く解法，画像処理への応用，さらに学習回路網の新しい応用など，この分野は急速に注目を浴びるようになった．

Neural Network の素子であるニューロンは以下のモデルがあげられる．

しきい素子モデル

内部状態をもたず，2 値出力を出す．自分の入力の総和を計算して，入力の総和がしきい値より大きい場合には，出力 1 を，そうでない場合は 0 を出す．

$$r = sign[\sum_n w_n a_n - \theta] \tag{2.2}$$

$$sign[x] = \begin{cases} +1 & (x \geq 0) \\ 0 & (x < 0) \end{cases} \tag{2.3}$$

ここで，a_n は n 番目の入力，w_n は n 番目の入力に対する結合の強度，θ はしきい値である．

準線形素子モデル

内部状態なしで，連続の出力をとる．入力の総和を計算して，その値を自分の特性関数によって変換して出力値を得る．特性関数の例として次のものがある．

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{2.4}$$

微分方程式／差分方程式モデル

ニューロンの状態変化を時間発展する微分／差分方程式によって記述する．次の方程式が使用される．

$$\begin{cases} \frac{\partial}{\partial t}\mathbf{u} = -\frac{1}{\tau}\mathbf{u} + i \\ v = g(\mathbf{u}) \end{cases} \tag{2.5}$$

ここで，$\mathbf{u}$ はユニットの内部状態，v は出力値，i はユニットへの入力の総和，g は Sigmoid などの連続関数である．

確率的なモデル

ニューロンの動作を確率的な状態変化規則や確率微分方程式などでモデル化する．たとえば，ボルツマンマシンと呼ばれる Neural Network のモデルでは次のような式を使用して出力値を求める [6].

$$p = \frac{1}{1 + \exp(-i)} \tag{2.6}$$

ここで，i はユニットへの入力の総和，p は出力値を 1 にする確率である．

Neural Network のニューロン間の結合に関するモデルとして階層型ネットワークと相互結合型ネットワークがあげられる．いずれもニューロン間をシナプスの荷重で結合し，シナプスの物理的形状や伝達効率をモデル化している．

階層型ネットワークでは，ニューロンがいくつかの層に分かれており，各層に順番が付いていて，前の層からしか入力を受け付けない．第 1 層から最終層に達する

と1回の動作が終了する．一方，相互結合型ネットワークでは，任意の2つのニューロン間に互いに結合があるようなネットワークである．ある初期状態から出発したネットワークは状態変化を繰り返すうちに平衡状態に達する．

2.1.3 誤差逆伝播法

一般的に学習で使用する Neural Network は，入力層，隠れ層，出力層よりなる階層型ネットワーク構造（hierarchy neural network structure）を基本とする．各層はニューロンに該当するユニットを，対象とする問題に応じた数だけもっており，固有の重みである伝達効率を伴って結合している．

データが入力層より入力されると，設定した動作関数を基に処理され出力層より答が出される．出力値を規定するのが各ユニット間の結合荷重値であり，この荷重値を期待する出力が得られるように変化させることが学習に相当する．そして，この階層型 Neural Network において，出力層の出力値と，教師信号との誤差が最小となるように結合荷重を変化させるのが，誤差逆伝播法である．

いま，ユニット j が荷重 w_{ij} で結合した前層のユニット i から出力 x_i を入力として受けるとする．このとき，j の出力 o_j は前層からの入力の総和 y_j に動作関数 f を施したもので，

$$o_j = f(y_j) \tag{2.7}$$

$$y_j = \sum_i w_{ij} x_i + b_j \tag{2.8}$$

となる．ここで，b_j はしきい値（bias）で，前層からの入力の総和がこれを越えたときに 1 に近い値を出力し，それ以外は 0 に近い値を出力するように，動作関数には Sigmoid 関数

$$f(y) = \frac{1}{1+e^{-y}} \tag{2.9}$$

を用いることが多い（図 2.2）．

ネットワークがある入力に対し o_j を出力したとする．教師信号を t_j とすると，出力と教師信号との誤差の2乗和が最小になるように学習をさせるため，次のような評価関数 E を定義する．

$$E = \frac{1}{2} \sum_j (o_j - t_j)^2 \tag{2.10}$$

E は o_j の，o_j は w_{ij} の関数である．ここで，E を w_{ij} のつくる多次元空間の曲面と考えれば，これを極小化するには，w_{ij} 点における曲面の傾きをもとめ，その負の傾きが最大となる方向へ進むように w_{ij} を変化させればよい．w_{ij} の変化量は定数

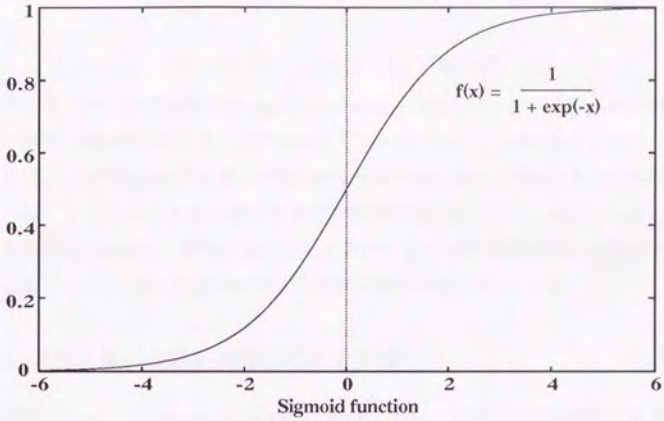


図 2.2 Sigmoid 関数

ϵ を用いて

$$\delta w_{ij} = -\epsilon \frac{\partial E}{\partial w_{ij}} \tag{2.11}$$

で表される．ここで，式 (2.10) を w_{ij} で偏微分する．

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \cdot \frac{do_j}{dy_j} \cdot \frac{\partial y_j}{\partial w_{ij}} \tag{2.12}$$

式 (2.10) より

$$\frac{\partial E}{\partial o_j} = o_j - t_j \tag{2.13}$$

式 (2.8) より

$$\begin{aligned} \frac{do_j}{dy_j} &= f'(y_j) \\ \frac{\partial y_j}{\partial w_{ij}} &= x_i \end{aligned} \tag{2.14}$$

であるので，式 (2.11) は，式 (2.12)，(2.13)，(2.14) より

$$\delta w_{ij} = -\epsilon(o_j - t_j) f'(y_j) x_i \tag{2.15}$$

となる（ここで，簡単のため bias $b_j = 0$ とした）．特に，評価関数 f に Sigmoid 関数（式 2.9）を採用した場合，α を定数として

$$f'(y_j) = f(y_j)[1 - f(y_j)]\alpha \tag{2.16}$$

となるので，δw_{ij} は

$$\delta w_{ij} = -\epsilon(o_j - t_j)f(y_j)[1 - f(y_j)]\alpha x_i \tag{2.17}$$

と表される．これは最急降下法（gradient descent method）と呼ばれる極小値の導出法であり，最急降下法を使った学習則はデルタルールとして知られている．

ここで，ϵ は学習過程で多次元空間を極小解の方向へ進ませる度合を示す係数である．一般に ϵ を小さくとると極小解までの到達時間が長くなり，大きくとると極小解から逸脱して振動し，収束しないことがある．α は学習の収束速度を規定する係数である．この逆伝播法の改良に関する研究は数多く報告されている．

2.2 アミノ酸／塩基配列解析による予測

一般に，Neural Network を効果的に活用するには，対象とする問題を包括する大量の学習データが必要である．タンパク質，核酸の分野では，データベース化されたアミノ酸や塩基の膨大な 1 次配列データを使用して，多くの問題に応用が試みられてきた [7].

2.2.1 翻訳開始部位

Neural Network を配列解析に初めて適用したのは，1982 年の Stormo ら [8] である．彼らは，隠れ層を持たない Perceptron を使用して *E. coli* 内の翻訳開始部位を予測した．彼らの目的は *E. coli* のリボゾームが翻訳開始コドンを選択する際のヌクレオチドの役割を定義づけることであった．

2.2.2 スプライス部位

1985 年には，Nakata ら [9] は，スプライス部位周辺の配列パターンを含んだ情報の判別解析によって，ヒトの mRNA 内のスプライス部位を予測した．1991 年に，Brunak ら [10] は，コード領域と非コード領域の識別問題を逆伝播法によって学習した Neural Network を使用して，DNA 内のヒト mRNA ドナーおよびアクセプター部位を予測した．小さな配列のセグメントを多層型 Neural Network に学習させて，イントロン／エクソンおよびエクソン／イントロン境界を識別した．

2.2.3 プロモータ部位

核酸に関しては，DNA のタンパク質コード領域の判別 [11][12]，プロモーター部位の判別 [13] 等の適用例があり，一定の成果を得ている．

2.2.4 タンパク質 2 次構造予測

初めて Neural Network をタンパク質 2 次構造予測の分野に適用したのは，Qian と Sejnowski(1988)[14] らのタンパク質の２次構造予測と言われている．彼らは，106 のタンパク質について，13 のアミノ酸シーケンスを入力，α，β，coil の３つの２次構造を出力として，１度の入力アミノ酸シーケンスの中心に位置するアミノ酸がどの２次構造に属するかを学習・テストし，Chou と Fasman(1974) ら [15] の従来の手法と比較して精度の高い（約 10%）予測が可能であることを示した．Holley と Karplus[16] も 1989 年に同様の手法で 2 次構造予測を行っている．

2.2.5 特定のタンパク質の 2 次構造予測

Andressen と Bohr(1990) ら [17] は，特定の２次構造を多く含むタンパク質（all-α，all-β）に限定して行えば正答率が向上することから，HIV の糖タンパク（gp120，gp41）の２次構造予測に適用して，いくつかの特徴的部位の存在を指摘した．

2.2.6 その他のタンパク質構造への適用

また，β - turn のタイプ分けに適用した例 (McGregor ら, 1989 [18]) や，ジスルフィド結合の有無の判別 (Muskal と Holbrook, 1990[19]) や，タンパク質の表面露出度によるアミノ酸の分類 (Holbrook ら, 1990[20])，ATP 結合モチーフの判別 (Hirst と Sternberg, 1991[21])，免疫グロブリン様ドメインの検出 (Benjio と Pouliot, 1990[22])，N 末端シグナルペプチドの判別 (Ladunga ら, 1991[23]) などの適用例がある．

2.2.7 立体構造予測

３次構造予測に適用した例 (Bohr ら, 1990, Wilcox ら, 1990)[24][25] もあるが，学習用データ量が少ないことや，使用タンパク質間のホモロジーが高く，一般的な３次構造予測には向かない．

表 2.2に，これまでになされた研究例をまとめる．その精度もさることながら，予め大量のデータを入力して学習を済ませておけば 1 次配列を入力するだけで瞬時に

回答が得られ，ホモロジー検索などの手続きが不要であることも Neural Network の有用性を高める要素となっている．

表 2.2 Neural Network のアミノ酸／塩基配列解析への応用例

		Reference	Problem	Result	Comparison	Network Design (units) input-hidden-output	Data proteins (residues) train test
Protein	secondary structure	Qian & Sejnowski, 1988	prediction of helix / sheet / coil	64%	50% (Chou & Fasman, 1978, non-ANN) 53% (Garnier et al., 1978, non-ANN)	13*21-40-3	106
		Holly & Karplus, 1989	prediction of helix / sheet	63%	48% (Chou & Fasman, 1978, non-ANN) 55% (Garnier et al., 1978, non-ANN)	17*21-2-2	48 (8315) 14 (2441)
		Bohr et al., 1988	prediction of helix / non-helix	73%	none	51*20-40-2	48 (about 10,000) 8
		McGregor et al., 1989	classification of beta-turn	26%	21% (Wilmot & Thornton, 1978, non-ANN)	4*20-8-4	58
		Kneller et al., 1990	prediction of helix / sheet / coil in all-helix and all-sheet	79% (all-helix) 70% (all-sheet)	64% (Qian & Sejnowski, 1988, ANN)	13*21-0-3	all helix : 22 all sheet :24
		Anderssen, Bohr et al., 1990	prediction of HIV protein gp120 and gp41	some particular region detected	(Chou & Fasman, 1978, non-ANN)	4*20-8-4	56 (about 10,000)
	tertiary structure	Bohr et al., 1990	prediction of tertiary structure of backbone	3.0 rms	74% homologous	61*20-300-33	13
		Wilcox et al., 1990	classification of backbone	no generalization	none	70*20-30-140*140	15
	other	Muskal et al., 1988	disulfide-bonding state of cystein	81%	none	10*21-0-2	128 (669sets) (20sets)
		Holbrook, 1989	surface exposure of amino acids	72% (2 types) 54% (3 types)	70% (Rose, 1985, non-ANN)	9*21-0-2 7*21-10-3	20 (3581) 5 (963)
		Hirst & Sternberg, 1991	recognition of an ATP/GTP-binding motif	78%	80% (Hirst & Sternberg, 1991, non-ANN)	17*20-0-1	348 motifs 1 motif
		Benjio & Poulot, 1990	recognition of immunogloblin domains	98%	none	5*20-8-4	30+ domains 56
		Ladunga et al., 1991	prediction of signal peptide	93%	77% (Heijne, 1989, non-ANN)	20*?- $\frac{3}{2}$-?	464 signal peptides 116
Nucleic Acid		Stormo et al., 1982	distinction of E. coli mRNA translational initiation sites (perceptron)	70%	60% (Stormo et al., 1982)	no specification	no specification
		Nakata et al., 1985	prediction of splice junctions in mRNA (perceptron)	73-91%	61-74% (Fickett, 1982)	4-0-?	78,000 nucleotides
		Farber & Lapedes, 1992	determination of protein coding regions in DNA	99.4%	90.9% (Baysian method, 90 codon)	6-?-1	about 30,000 1,000 codons
		Uberbacher & Mural, 1991	determination of protein coding regions in DNA	92%	none	7-$\frac{10}{5}$1	no specification
		O'Neill, 1991	E. coli promoter recognition	80%	70% (O'Neill, 1989, non-ANN)	4-0-?	about 15,000 36 promoter

Table 1.1 Neural Network Applications to Structural and Functional Analysis of Protein and Nucleic Acid Sequences

第 I 部

自己増殖オートマトンネットワークの開発

第 3 章

マシンモデルの設計

Hubel ら [26, 27] は視覚の脳内機構に関する研究によって神経回路の情報処理の仕組みを特徴付けた．Hubel によると情報処理の機械としての脳は，1) 膨大な数の低速なニューロンが互いに連結し，2) 個々のニューロンは各々の規則に従って並列に動作し，3) 外部の環境に応じたユニットの増殖によってネットワーク形態が変化し，4) 集団内における遺伝操作によって個体が変化する．これらの特徴が脳に特有の情報処理，例えば事前の凡例学習によって蓄積した経験的知識による問題解決を可能にしている．

脳の構造や動作についての知見を基に神経回路モデルを構築する動きとして，1940 年代の McCulloch と Pitts[2] の閾値素子や D.Hebb[28] のシナプス強化則に始まり，Rosenblatt[3] の Perceptron や Rumelhart と McClelland[4] の逆伝播法による学習，さらに Hopfield[5] の相互結合型ネットワークによる最適化などの機能を持つ様々な Neural Network が研究されてきた．Neural Network への期待は学習能力であるが，学習対象のデータが多くなるとネットワークの規模が大きくなり，計算時間が増大し，収束性も悪化する．

学習速度を改良した試み [29, 30, 31, 32, 33, 34] が数多く提案されてきたが，変化する学習環境に対してネットワークを決定する際に困難が伴い，最適な学習能力を保持することが難しい．ネットワーク形態を変化させて改善を試みた例として，Reilly ら [35] の RCE モデル，岩田ら [36] の CombNET-II，さらに Fahlman ら [37, 38, 39] の Cascade-Correlation のようにユニット数を増加させた例や，Hagiwara[40] や藤井ら [41] による冗長なユニットを削減する例が見られるが，主に学習効率を改善するための手段であり，環境の変化に動的に対応してネットワークを構築することは目的とされていない．Genetic Algorithm を用いて改善を試みた例として，Whitley ら [42] のネットワークの重み付けの最適化，Kitano ら [43] の提案した NGL 法によるネットワーク構造の最適化が報告されている．また，Fogel ら [44, 45, 46] は

Evolutionary Programming によって，進化による適応的学習を実現した．しかし，いずれもネットワークの集団の中から遺伝操作によって適応度の高い個体を淘汰していくため，数多くのネットワークを用意する必要が生じるばかりか，ネットワークの最適化が完了するのは何世代も経過した後に持ち越される．

このような問題点はハードウェアの改良によってある程度まで補うことができるが，脳が低速な素子で構成されながら，効率の良い学習をするという事実は，新しいマシンモデルの構築方法の必要性を示唆している．本論文では，新しいマシンアーキテクチャの試みとして，自己増殖オートマトンネットワーク（Self-**Repro**ducing Automata **Net**work：以下 ReproNet と記す）を提案する．ReproNet は，ユニット，ネットワーク，変移則から構成される．このモデルでは前述の Hubel の 1)2)3) の分析に着目して，1) 複数のユニットが相互に結合してネットワークを構成し，相互に情報交換する．2) 個々のユニットは，各々の規則に従って並列に動作する．3) ユニットの増殖によってネットワーク構造を変化させながら処理を行う．これは多数の素子が連結した並列マシンとしての枠組を提供するアーキテクチャである．なお，4) については，同一世代内で常時ネットワーク構造の最適化を完了させるため，本研究では遺伝操作による固体の変化は扱わなかった．

3.1 脳内機構の計算機による実現化

Hubel が指摘した脳の情報処理の特徴は次の通りである [27].

1. 脳では低速ではあるが膨大な量の素子が複雑なネットワークを構成する．ニューロンの数は 10^{11}，シナプス結合の数は 10^{14} である．

2. 個々の命令を直線的に順次実行するプログラムストア型計算機とは対照的に，脳は各素子が持つ規則に従って並列に動作する．

3. 脳では素子が動的に増殖（produce）する．

4. 集団内における遺伝操作によって個体が変化する．

本研究では以上に着目して，ReproNet を，ユニット，ネットワーク，変移則の 3 要素で構成した．（1）として，複数のユニットは相互に結合してネットワークを構成し，相互に情報交換する．（2）として，個々のユニットは，各々の動作を規定した変移則を持ち，それに従って独立に並列動作する．（3）として，ユニットの増殖によってネットワーク構造を変化させながら処理を行う．なお，（4）について，本

論文では現在の学習性能に基づく情報により，同一世代内でネットワーク構造の最適化を完了させるため，遺伝操作による固体の変化は扱わず，今後の課題とした．

3.2 ReproNet の定義

本研究では独自に ReproNet の各要素を次のように設計した．

3.2.1 ユニット

ユニットは独立して並列に動作する 1 つの自己増殖オートマトンである[1]．図 3.1にユニットの構造を示す．入出力と内部状態をもち，変移則によって動作が規定される．有限オートマトンが相互に結合してオートマトンネットワークを構成する．ユニット M は次のように定義する．

$$M = (Q, U, v, g, R, q_0) \tag{3.1}$$

ここで，Q：内部状態の集合，U：入力値の集合，v：出力値，g：出力値を対応させる関数（出力関数），R：変移則の集合，q_0：初期状態である．

ユニットは外部からの入力値を伝達効率によって重み付けを与えてから内部に取り入れる．ユニットは内部状態を持ち，それに応じて変移則を決定する．変移則はその時刻におけるユニット内の情報から次のユニットの動作や内部状態を決定し，出力関数を用いてユニットの出力値を決定する．ユニット相互の干渉はネットワークのパスを通した信号の伝達だけであり，個々のユニットは独立しており並列に動作する．

3.2.2 ネットワーク

ネットワークは，有向グラフ G を用いて定義する．図 3.2にネットワークの構造を示す．グラフ G は次のように定義する [48]．

$$G = (V', E) \tag{3.2}$$

$$V' = V \cup V_i \tag{3.3}$$

$$E = (E_1, E_2, E_3, \ldots, E_k) \tag{3.4}$$

[1] von Neumann の提唱した自己増殖オートマトンは方形のセルが無限に並び，各セルは 29 の状態をもつが [47]，ここではセルは考慮しない．

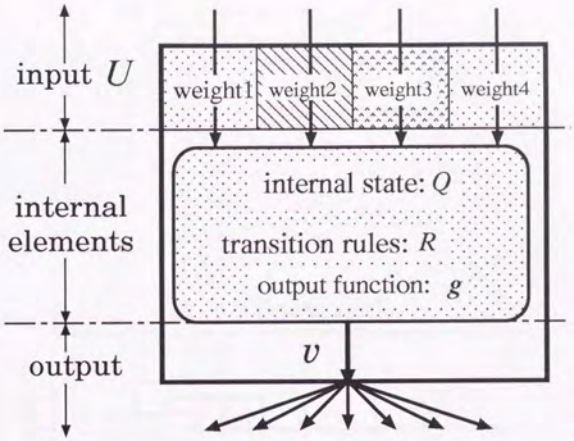


図 3.1　ユニットの構造．内部状態 Q，変移則 R を持つ．外部からの信号は重み付けが与えられてから内部に取り入れられる．

$$E_i \subset V \times V' \tag{3.5}$$

ここで，V'：ノードの集合，V：内部ノードの集合，V_i：入力ノードの集合，E：枝の集合である．グラフ G のノード V' 上にユニット M を配置したものがネットワークである．ユニット間の信号が通るパスはグラフの枝である．このとき，ノード V' に入ってくる枝の数はユニット M の入力数に等しい．ネットワーク A は次のように定義する．

$$A = (G, f) \tag{3.6}$$

$$f : V \to M(Q) \tag{3.7}$$

ここで，G：グラフ，$M(Q)$：状態集合 Q のユニット全体の集合．ただしノード $v \in V$ の入力数が k のとき $f(v)$ は k 入力のユニットである．

3.2.3　変移則

変移則は個々のユニットがもつ動作規則である．図 3.3に内部状態の遷移の例を示す．内部状態は変移則に基づいて次の状態に変化する．図 3.4に変移則の流れ図を

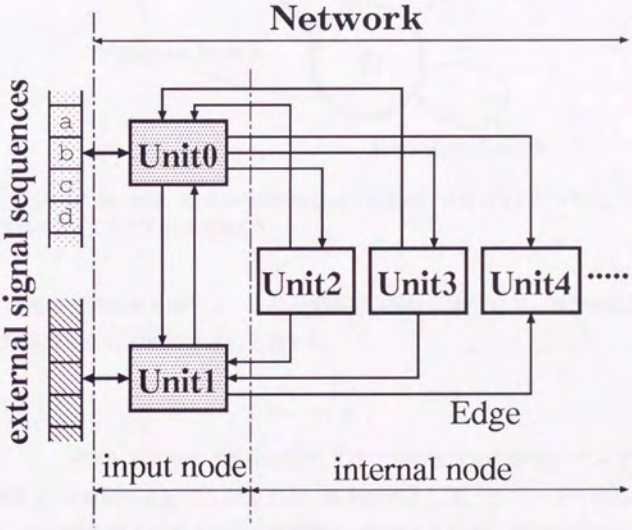


図 3.2 ネットワークの構造．入力ノードを通してネットワーク外部と信号を交換する．

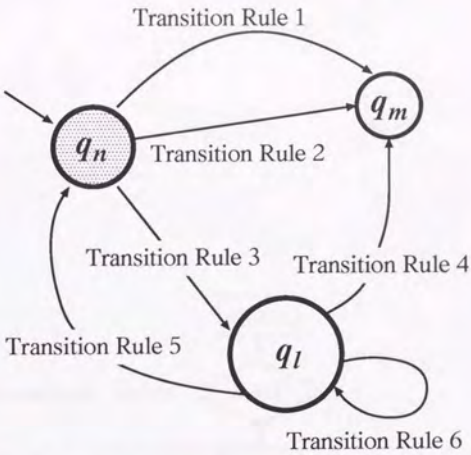


図 3.3　内部状態の遷移．現在の内部状態は q_n．変移則 1 または 2 によって q_m に変化する．また変移則 3 によって q_l に変化する．

示す．最初に内部状態を調べる．次に，変移則の条件に基づいて，動作が選択される．変移則の集合 R は次のように定義する．

$$R = (C, W) \tag{3.8}$$

ここで，C：変移則を適用する条件の集合，W：実行する動作の集合である．条件 C と動作 W は適用例に応じて決定する．条件の例として，ユニット外部との入出力関係，内部状態およびネットワーク形態などが考えられる．動作の例として，出力値の決定，伝達効率の変化，信号の出力，内部状態の変化，ユニットの増殖（除去），パスの追加（削除），ネットワーク形態の変化などが考えられる．

ネットワークの形態変化は Graph Grammar[49][50] の生成規則を用いる．図 3.5に生成規則の例を示す．Graph Grammar の生成規則 G^2 は次のように定義する．

$$G^2 = (\alpha, C, \beta, E, S) \tag{3.9}$$

ここで，α, β：ラベル，C：生成規則の適用条件．真のとき α に β を埋め込む．E：β を埋め込む方法を規定する連結規則，S：初期ネットワークである．

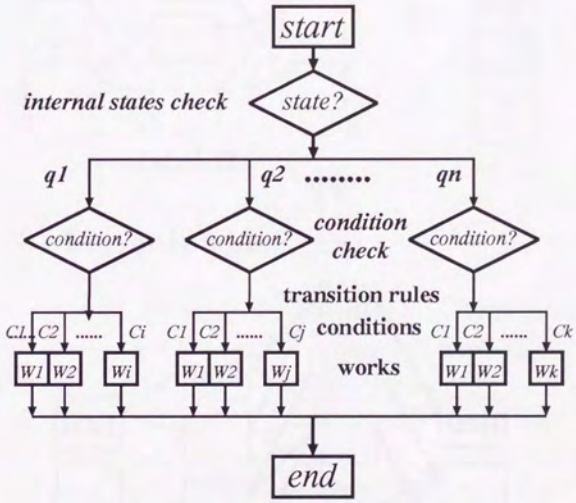


図 3.4　変移則の流れ図．現在の内部状態（$q_1 \cdots q_n$ のうちの 1 つ）にしたがって変移則を決定する．

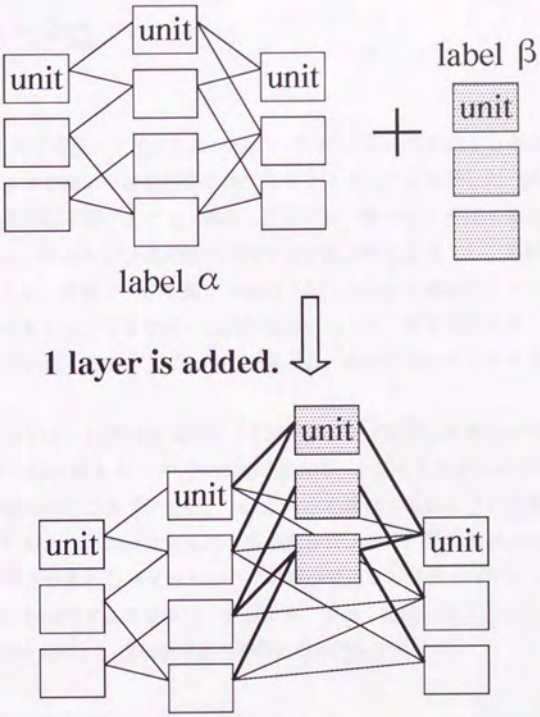


図 3.5　ネットワークの生成規則．この例では，1 層 3 ユニットの追加である．最初のラベル α に，網がけのラベル β を追加して形態が変化する．α と β 間の枝の結合方法は連結規則によって規定される．

第 4 章

学習能力の検証

ReproNet の学習能力を検証するために，学習中に動的に変化する環境に対して，構造を適応させて高い学習性能を維持するネットワークを構築した．適用例としてパターン認識問題を採り上げた．構築したモデルの汎用性を検証するために，2 種の例を扱った．具体的には暗号解読問題と文字認識問題を例とし，動的に変化する環境を作るため，学習データを徐々に追加した．暗号文の解読ではネットワークの学習能力を検証した．文字認識では学習能力に加えて，学習済みのネットワークに対してノイズを含んだデータによってテストし，未知のデータに関する識別能力を検証した．

このモデルでは，1) 学習性能の低下を出力ユニットにおける誤差から評価する変移則，2) その評価値をネットワーク内の全ユニットに伝播させる変移則，3) 各ユニットが性能評価値に基づいて新しいユニットを増殖するかどうか決定する変移則をユニットに組み込んだ．これらの 3 種の変移則よって，隠れ層のユニットを必要に応じて増加させることが可能となり，環境が変化して性能が低下する場合でも，高い学習能力を保持するネットワークが実現できた．比較の基準として momentum 法の逆伝播法を使用し，実験結果から特徴と優位性を示す．

4.1 暗号解読問題

暗号化された文字列を入力して原文文字列を出力する階層型 Neural Network を用いて実験を行った．暗号文を作成するために，2 通りの手法を用いた．

1. ビットごとの排他的論理和（以下 XOR と記す）による暗号化．原文の 1 バイト文字 m に対して，鍵 k を seed として得られた 1 バイトの乱数 r との XOR を取り，暗号文の 1 バイト文字 c を得る．r は原文の 1 文字ごとに変化する一様乱数であり，互いに独立している．暗号文に対して同じ手続きを適用すると原

```
原文文字列
496e 205b 7468 655d 2062 6567 696e 6e69
 I n    [  t h  e ]    b  e g  i n  n i
6e67 2047 6f64 2063 7265 6174 6564 2074
 n g    G  o d    c  r e  a t  e d    t
6865 2068 6561 7665 6e73 2061 6e64 2074
 h e    h  e a  v e  n s    a  n d    t
6865 2065 6172 7468 2e20 4e6f 7720 7468
 h e    e  a r  t h  .    N o  w    t h
排他的論理和による暗号化文字列
49c9 6ef7 6fec 18c7 7e23 3ab4 4522 caa3
 I    n    o         ~ #  :    E "
6ec0 6eeb 74e0 5df9 2c24 3ea7 4928 84be
 n    n    t    ]    , $  >    I (
68c2 6ec4 7ee5 0bff 3032 7fb2 4228 84be
 h    n    ~         0 2       B (
68c2 6ec9 7af6 09f2 7061 11bc 5b6c d0a2
 h    n    z   \t    p a       [ l
crypt による暗号化文字列
39e7 138b 0adf a529 aba2 ded4 b60e 2aff
 9        \n      )                  *
a606 132c e814 d8df c19d a1f5 2928 1c9f
        ,                        ) (
0784 13a8 1580 3213 712f 240f bd28 1c9f
                2    q /  $      (
0784 1304 4913 c3e1 299e 0cbe a217 9f09
          I         )   \f          \t
```

図 4.1 暗号問題で使用した文字列の例

文に復号される．乱数列 $r_1 r_2 \cdots r_n$ を用いて，原文文字列 $m_1 m_2 \cdots m_n$ に対して，暗号化文字列 $c_1 c_2 \cdots c_n$ を作成する方法は次の通りである．

$$c_i = m_i \oplus r_i \ \ (i = 1, 2, \ldots, n) \tag{4.1}$$

2. UNIX の crypt による暗号化．使用した UNIX は，HP-UX A.09.05 である．

4.1.1 使用データ

今回の適用例では，学習用の原文データとして，16 文字からなる文字列を 120 組（合計 1,920 文字）用いた．また，テスト用の原文データとして，16 文字からなる文字列を 100 組（合計 1,600 文字）用いた．使用した学習用文字列の 16 進表記とテキストの例を図 4.1に示す．

4.1.2 学習方法

データの増加と学習方法は次の通りである．

1. ネットワークに対して，初期の学習データ 8 組を与える．入力として暗号化文字列を，教師信号として原文文字列を与える．

2. 学習規則に従ってパターンを学習させる．学習方法は momentum 法による結合荷重の更新である．学習の完了の条件は式（4.9）におけるネットワークの出力誤差の平均を用いて次のように決める（4.1.4節を参照）．

$$\bar{\varepsilon}_A < 0.4 \tag{4.2}$$

学習の完了の可否にかかわらず，規定回数（10,000 回ループ）に達した場合は学習を打ち切り，次のステップに進む．

3. 未知のデータ 100 組を使用してテストする．暗号化文字列を入力し，ネットワークの出力した文字列を原文文字列と比較する．

4. 新たに学習用データを 2 組追加する．

5. データがなくなるまで 2. から 4. を繰り返す．

学習能力の低下を反映した自己増殖機能に関して，上記の手順のステップ 2. を詳述すると次の通りである．

2.1 出力層のユニットにおいて，現在のネットワークの学習能力を評価するために，各ユニットごとに教師信号との誤差から性能評価値を求める．

2.2 その評価値をネットワーク内の全ユニットに対して，入力信号と逆方向に伝播させる．

2.3 各ユニットは各自の内部状態や性能評価値に基づいて新しいユニットを増殖するかどうか決定する．

4.1.3 ユニットの構築

ReproNet のユニットを構築した．内部状態は次の通り．

$$Q = (in, wait, ready, out, bias) \tag{4.3}$$

ここで，in：入力信号を受け入れる状態，out：教師信号を受け入れる状態，$bias$：常に一定のしきい値を出力する状態，$wait$：ユニットを増殖させない状態，$ready$：ユニットを増殖させる状態である．ユニットの初期状態は，入力層：in，隠れ層：$wait$，出力層：out，しきい値ユニット：$bias$ とした．内部状態 $ready$ は増殖機能を持つユニットだけに与える．

出力関数には，Sigmoid 関数 $g(x) = 1/(1 + e^{-x})$ を用いた．

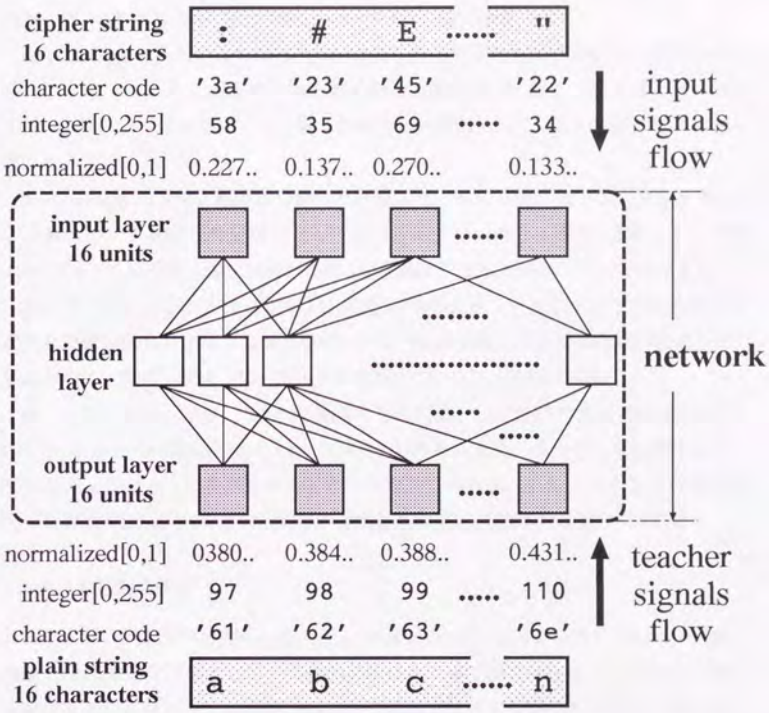


図 4.2 ネットワークの構成（暗号解読）

4.1.4 ネットワークの構築

ReproNet のネットワークを構築した．ネットワークの構成を図 4.2の点線内に示す．ネットワークの入力ノードは入力層と出力層であり，内部ノードは隠れ層である．

入力層と出力層のユニット数は固定である．通常の Neural Network の隠れ層は固定であが，ReproNet の隠れ層は動的に変化する．実験では，隠れ層のユニット数が異なる 4 個の Neural Network を使用して，ReproNet と比較した．具体的には表 4.2に示す．

入力層では，一度に 1 組の暗号化文字列（16 文字）を入力し，1 文字に対して 1 つのユニットが対応する．すなわち入力層は 16 個のユニットで構成する．出力層で

は，原文文字列の 1 文字に対して 1 つのユニットが対応する．一度に入力される文字が 1 組 16 文字なので，出力層のユニットも 16 個である．

図 4.2の点線の上部に入力信号の流れを示した．暗号化文字の文字コード（"00"から"ff"）を 0 から 1 に正規化した実数値を入力信号として扱う．例えば，暗号化文字"#"に関して，文字コード"23"を 0 から 1 の実数値0.137... に正規化して入力層のユニットに与える．

図 4.2の点線の下部に教師信号の流れを示した．原文文字の文字コードを 0 から 1 に正規化した実数値を教師信号として扱う．例えば，原文文字"a"に関して，文字コード"61"を 0 から 1 の実数値0.380... に正規化して出力層のユニットに与える．

教師信号と出力ユニットの誤差は，教師信号の文字コードを 0 から 255 の整数に変換した値と出力ユニットの出力値を 0 から 255 の実数値に変換した値の差とした．したがって，誤差1 とは，文字コードの差が1 であることを意味する．

表 4.1に文字"a"，"b"，"c"の例を示す．例えば，"a"と"b"の誤差は97 と98 の差の1 で，"a"と"c"の誤差は97 と99 の差の2 である．また，誤差が0.5 未満のとき，出力ユニットは正しい文字を出力したとみなす．例えば，0 から 255 の実数値に変換した出力が98.4 や97.7 のときはいずれも整数値98 で表される文字"b"である．

4.1.5 変移則の構築

ReproNet の変移則を構築した．図 4.3に変移則の流れ図を示す． ReproNet は，増殖に関係する変移則として，現在の環境における学習性能をネットワーク内部に反映させる変移則 r_2 と r_5，層とユニットの増殖を制御する変移則 r_3 と r_6 を持つ．

ReproNet において，隠れ層のユニットは 2 種の内部状態を持つ．通常は *wait* 状態だが，*ready* 状態になると，新しいユニットを 1 つ増殖する．出力層のユニットはすべて *out* 状態であるが，層の増殖を制御する変移則 r_3 を持つ．以下に適用される内部状態ごとに個々の変移則を説明する．

表 4.1 1 バイト文字とユニット内部の信号（暗号解読）

character	code "00"-"ff"	integer [0,255]	normalize [0,1]
a	"61"	97	0.380...
b	"62"	98	0.384...
c	"63"	99	0.388...

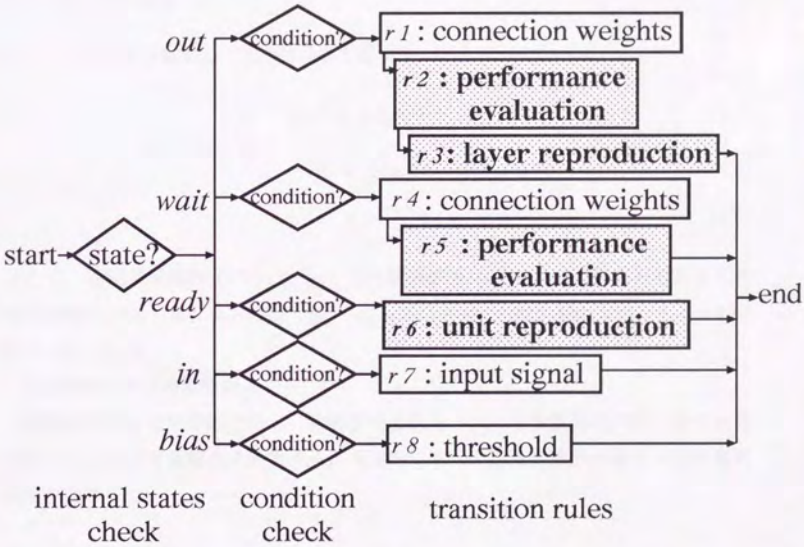


図 4.3　変移則の流れ図．増殖に関係する変移則を網掛けで示した．まず，内部状態を調べる．次に，条件を満たした変移則を実行する．

内部状態 out のときの変移則 $r_1 r_2 r_3$

結合荷重に関する変移則 r_1

通常の Neural Network の結合荷重を計算する．逆伝播法を改良した数多くの手法 [29, 30, 31, 32, 33, 34] が研究されてきたが，ここでは 1 次偏微分の第一項と第二項によって加速化された逆伝播法である momentum 法を使用した．他の改良型逆伝播法を使用する際には，この変移則 r_1 に組み込む．変移則の条件と動作は次の通りである．

C_1： 常に真とする．

W_1： ユニット j に対して結合荷重を更新する．

$$\Delta w_{ij}^T = \eta_j \delta_j o_i + \alpha_j \Delta w_{ij}^{T-1} \tag{4.4}$$

$$\delta_j = g_j'(y_j)(t_j - o_j) \tag{4.5}$$

$$y_j = \sum_i w_{ij} o_i \tag{4.6}$$

ここで，η_j：学習係数（0.75），α_j：慣性項の係数（0.8），t_j：外部から与えられる教師信号，o_j：ユニットの出力値，w_{ij}：ユニット j の前の層のユニット i からの結合荷重である．

性能評価に関する変移則 r_2

増殖に関係する変移則である．教師信号と出力ユニットの誤差の平均に基づいてユニットごとに学習能力を評価する．具体的に定式化した変移則の条件と動作は次の通りである．

C_2： 常に真とする．

W_2： ユニット j に対する性能評価値 $\tilde{\epsilon}_j$ を次のように計算する．

$$\epsilon_j = |t_j - o_j| \tag{4.7}$$

$$\tilde{\epsilon}_j = (\epsilon_{j_{\max}} + \epsilon_{j_{\min}})/2 \tag{4.8}$$

ここで，ϵ_j：ユニット j の教師信号と出力値の誤差，$\tilde{\epsilon}_j$：現在の環境における誤差 ϵ_j の平均に基づくユニット j の性能評価値である．$\tilde{\epsilon}_j$ は変移則 r_5 によって前の層に伝播される．$\tilde{\epsilon}_j$ の値が大きいほど，性能が悪いことを意味する．

層の増殖に関する変移則 r_3

ネットワーク全体の学習能力を評価し，層の増殖を制御する変移則である．π_A によって指定した期間，ネットワークの性能が改善されない状態が続くと，現ネットワークは現在の環境を学習する能力がないと判断し，ネットワークの層を増殖させる．層の増殖は π_A によって制御する．

1 層の隠れ層で任意の境界領域を構成できることが知られているが [51]，2 層以上の隠れ層でより優れた学習性能を示すことがあるので，層の増殖を導入した．具体的に定式化した変移則の条件と動作は次の通りである．

C_3： π_A によって指定した期間，ネットワークの性能評価値 $\bar{\varepsilon}_A$ が改善されない状態が続くと真．$\bar{\varepsilon}_A$ は次のように計算する．

$$\bar{\varepsilon}_A = \sum_{j}^{n} \bar{\epsilon}_j / n \tag{4.9}$$

$\bar{\varepsilon}_A$ は出力層のユニット j の性能評価値の平均に基づくネットワークの性能評価値である．$\bar{\varepsilon}_A$ の値が大きいほど，性能が悪いことを意味する．

W_3： α に β を追加（図 3.5 参照）．β の初期ユニット数は，表 4.2の隠れ層の初期ユニット数と同じ．E：枝は前後の層の全ユニットを連結する．なお，隠れ層の上限は 2 層までとした．

内部状態 $wait$ のときの変移則 $r_4 r_5$

結合荷重に関する変移則 r_4

通常の Neural Network の結合荷重を計算する．r_1 と同様に momentum 法を使用した．変移則は次の通りである．

C_4： 常に真とする．

W_4： ユニット j に対して結合荷重の更新，出力値の計算をする．

$$\Delta w_{ij}^{T} = \eta_j \delta_j o_i + \alpha_j \Delta w_{ij}^{T-1} \tag{4.10}$$

$$\delta_j = g_j'(y_j) \sum_{k} w_{jk} \delta_k \tag{4.11}$$

ここで，w_{jk}：ユニット j の後の層のユニット k への結合荷重である．

性能評価に関する変移則 r_5

増殖に関係し，現在の環境に対する学習能力をユニットごとに評価し，その評価値をネットワークに反映させる変移則である．π_j によって指定した期間，ユニット

の性能が改善されない状態が続くと，現ネットワークには現在の環境を学習する能力がないと判断し，内部状態を $ready$ する．$ready$ 状態のユニットは後述の変移則 r_6 に従って，即座に新しいユニットを 1 つ増殖する．ユニットの増殖は π_j によって制御する．具体的に定式化した変移則の条件と動作は次の通りである．

C_5： π_j によって指定した期間，ユニット j の性能評価値 $\tilde{\epsilon}_j$ が改善されない状態が続くと真．$\tilde{\epsilon}_j$ は次のように計算する．

$$\epsilon_j = \sum_k w_{jk}\tilde{\epsilon}_k \tag{4.12}$$

$$\tilde{\epsilon}_j = (\epsilon_{j_{\max}} + \epsilon_{j_{\min}})/2 \tag{4.13}$$

W_5： 内部状態を $ready$ にする．

ユニット j は出力先のすべてのユニット k に影響を与えるので，ユニット k が持つ性能評価値 $\tilde{\epsilon}_k$ に結合荷重 w_{jk} を与えて加算した値をユニット j の性能評価値 $\tilde{\epsilon}_j$ とする．$\tilde{\epsilon}_j$ の値が大きいほど，性能が悪いことを意味する．隠れ層が複数存在する場合，性能評価値 $\tilde{\epsilon}_j$ は 1 つ前の隠れ層に伝播される．

なお，ユニットを減少させる機能をこの変移則に組み込むことも可能である．具体的に定式化した変移則の条件と動作は次の通りである．

$C_{5'}$： 規定した期間ユニットが増殖しない状態が続くと真．

$W_{5'}$： 規定した個数だけ古い順番にユニットを削除する．

このユニットを減少させる変移則は，本節では用いず，6.4.2節で実装した．

内部状態 $ready$ のときの変移則 r_6

ユニットの増殖に関する変移則 r_6

$ready$ 状態のユニットは，無条件に新しいユニットを 1 つ増殖し，内部状態を $wait$ に戻す．図 4.4にユニットの増殖の例を示す．変移則の条件と動作は次の通りである．

C_6： 常に真とする．

W_6： ユニットを 1 つ増殖する．増殖したユニットは前後の層のすべてのユニットと連結する．内部状態を $ready$ から $wait$ に戻す．

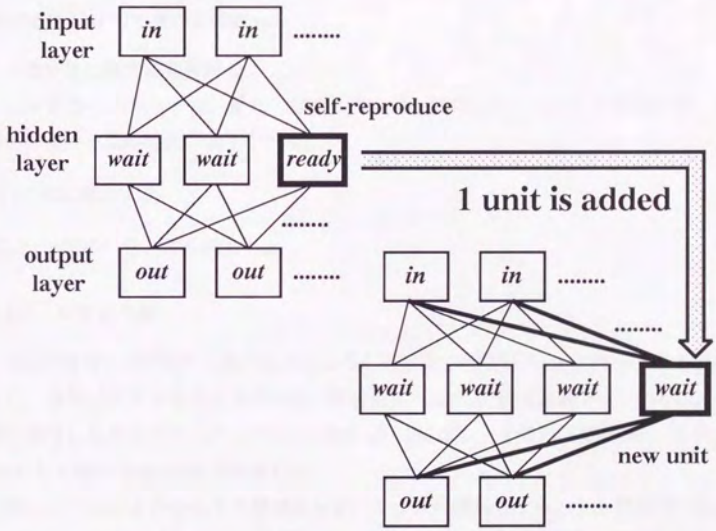


図 4.4　ユニットの増殖．この例では，1ユニットの追加である．ネットワークの隠れ層の *ready* 状態のユニットが自己増殖して，ユニットが追加される．枝の結合方法は連結規則によって規定される．

内部状態 in のときの変移則 r_7

入力信号に関する変移則 r_7

in 状態のユニットは，常に外部からの入力信号を接続先のユニットに伝達する．変移則の条件と動作は次の通りである．

C_7： 常に真とする．

W_7： ネットワーク外部からの入力信号（暗号文字列に相当）を出力する．

内部状態 $bias$ のときの変移則 r_8

しきい値に関する変移則 r_8

$bias$ 状態のユニットは，常に一定のしきい値を接続先のユニットに伝達する．変移則の条件と動作は次の通りである．

C_8： 常に真とする．

W_8： 一定のしきい値を出力する．

4.1.6 結果と考察

XOR の暗号文を学習する能力を ReproNet と通常の階層型 Neural Network で比較した．横軸は学習対象の文字列の数，縦軸は式（4.9）で求まる誤差 $\bar{\varepsilon}_A$ である．実験で使用したネットワークのユニット数を表 4.2に示す．4 個の通常の Neural Network と 1 個の ReproNet を比較した．

図 4.5に Neural Network の結果を示す．サイズが隠れ層 1 ユニット数 40 の NeuroNetA および隠れ層 2 ユニット数 16-16 の NeuroNetB は，学習するデータの少ない初期の環境（各々文字列 38 までと文字列 16 まで）では良い学習性能を示すが，学習データが大きくなるにしたがって，極端に性能が低下する．サイズを隠れ層 2

表 4.2 ネットワークのユニット数（暗号解読）

network name	input	hidden	output
NeuroNetA	16	40（1 層）	16
NeuroNetB	16	16 - 16（2 層）	16
NeuroNetC	16	36 - 36（2 層）	16
NeuroNetD	16	100 - 100（2 層）	16
ReproNet	16	16（1 層）	16

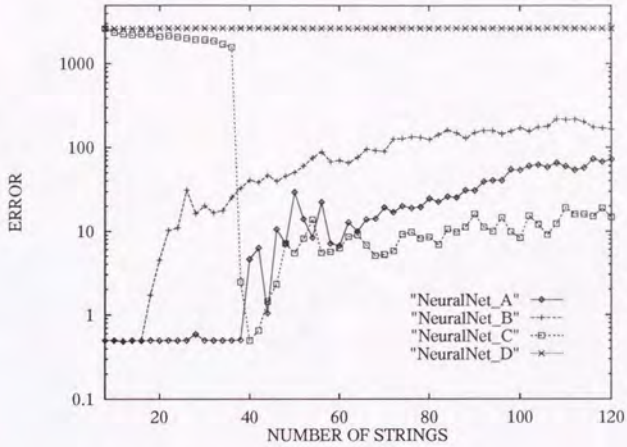


図 4.5 Neural Network の学習能力（排他的論理和）

ユニット数 36-36 の中程度にした NeuroNetC は，学習環境が大きくなるまで，効率の悪い学習を強いられ（文字列 36，38 付近まで），早い段階ではネットワークに無駄が見られる．その後，学習環境が大きくなると性能が低下する．さらにサイズが大きい隠れ層 2 ユニット数 100-100 の NeuroNetD ではネットワークに無駄が多く，学習がほとんど不可能である．

図 4.6に ReproNet の結果を示す．ReproNet は，刻々と変化する環境に応じて動的にネットワーク形態を変化させているので，常に効率的な学習を実現している．従来の Neural Network では，学習性能が初期のネットワーク構造に依存しており，環境の変化に対応することができないが，ReproNet は安定した学習性能を維持している．

UNIX の crypt による暗号文を学習する能力を ReproNet と Neural Network で比較した結果を図 4.7に示す．全体的な傾向は XOR の場合と同様である．小さいサイズの NeuroNetB では XOR の場合よりも性能が若干低下する．中程度のサイズの NeuroNetC は早い段階の効率の悪い学習の期間が XOR の場合より長くなり，しかも，その後の性能の低下が著しくなる．一方，ReproNet はいずれも効率よく学習できている．

これらの実験結果は，通常の Neural Network では学習対象データが，XOR の暗号文より解読の困難な UNIX の crypt の暗号文に変化すると，解を得ることがより難

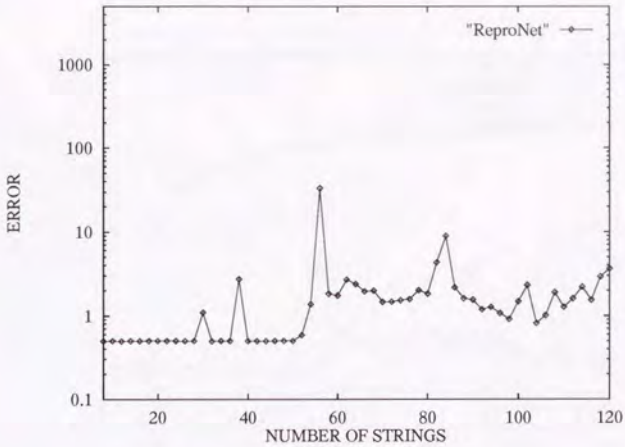


図 4.6 ReproNet の学習能力（排他的論理和）

しくなるのに対し，ReproNet は暗号化の差異に影響されず，安定した学習が可能であることを意味している．

ReproNet が Neural Network に比べて優れた学習能力をもつ理由として，学習環境から得られる情報に基づいて，ネットワークの学習性能を監視し，ネットワーク形態が動的に変化することが挙げられる．ReproNet の隠れ層は 1 層から開始したが，いずれの場合も隠れ層は 2 層に増殖した．図 4.8には第 1 層のユニット数の増加を示した．学習環境の拡大に応じてユニットが動的に増殖している様子が観察される．

図 4.9に時刻に応じた，学習文字列の数，誤差，隠れ層のユニット数の変化を示した．ループ 210000 手前付近で学習対象の文字列が 80 から 82 に増えると，誤差が急激に増加するが，ユニットが学習状況に応じて動的に増殖しているため，瞬時に回復している様子が分かる．ループ 220000 手前付近では文字列は 82 から 84 に増加しているが，ここでも同様のことがいえる．

未知の暗号文をネットワークに入力してテストした結果については，Neural Network の誤差が 1000 を超えるのに対して，ReproNet は平均して，500 から 700 程度だったので，ReproNet がよい性能を示した．しかし，ReproNet と Neural Network のいずれも全体的に誤差が大きい．これは，暗号文の入力パターンが全部で

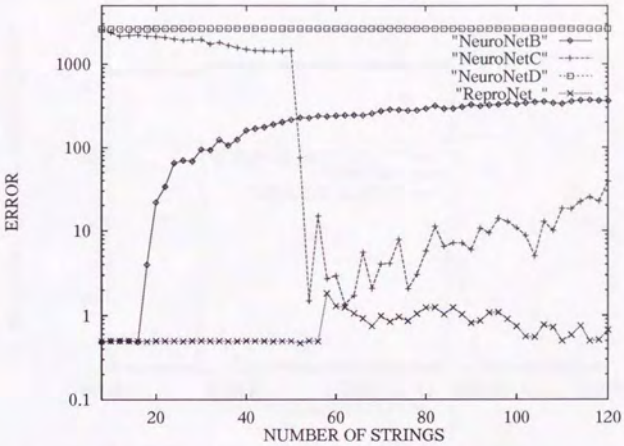


図 4.7 ReproNet と Neural Network の学習能力の比較 (crypt)

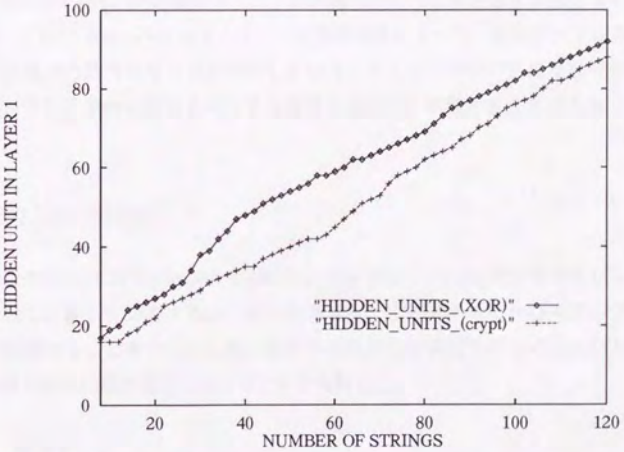


図 4.8 ReproNet の環境に応じたユニットの増殖

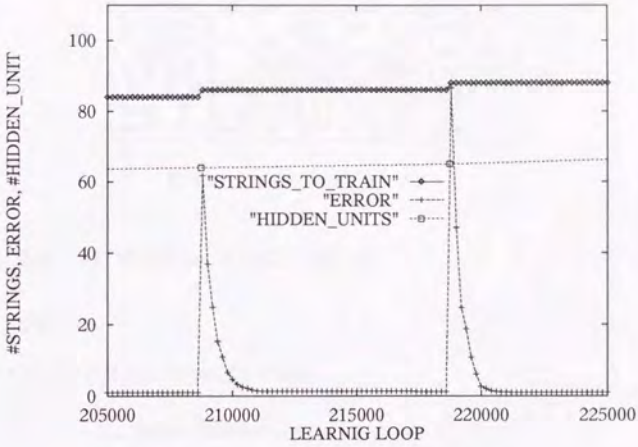


図 4.9 ReproNet の環境に応じた形態変化（crypt）

256^{16} 組あり，わずか 120 組の学習データだけで膨大な未知のデータを解読することは困難であると考えられる．

Hagiwara は [40]，隠れ層のユニット数の変化によって過学習を回避する可能性を示した．これは ReproNet のネットワーク形態変換によって，未知データに対する認識率の改善が可能であることを示唆している．ネットワークの冗長な部分を縮小するなど，さらに柔軟に形態を変化する機能を変移則に構築する必要がある（6.4.2節参照）．

4.2 文字認識問題

Neural Network と ReproNet を用いて，アルファベット 26 文字を表すビットマップを入力し，正しく認識するように学習させた．またノイズを含む未知のデータを正しく認識することをテストした．モデルの汎用性を実証するため，入出力の部分以外は暗号解読問題の場合と同じモデルで実験した．

4.2.1 使用データ

1 文字 5×5 の 25 ドットの学習用入力データ 26 種類を使用する（図 4.10 参照）．ノイズを含むテスト用データは，上記の 1 ドットにノイズを加えたものを 1 文字に

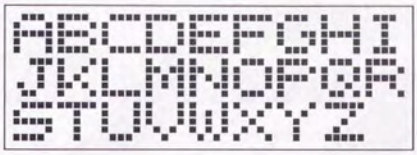


図 4.10 25 ドットのアルファベット

つき 25 種類（合計 $25 \times 26 = 650$ 種類）用意した．

4.2.2 学習方法

データ増加と学習方法は次の通りである．

1. ネットワークに初回の学習用データとして文字 A を与える．
2. 規定回数（20,000 回ループ）を除いて，暗号解読問題と同じ．
3. 未知のデータとして，1 文字につき 25 個のノイズデータを使用してテストする．
4. 学習済みのデータに加えて新たに学習用データとして，次の 1 文字を追加する．
5. 全データ（A から Z の 26 文字）の学習が終了するまで 2. から 4. を繰り返す．

4.2.3 ユニットと変移則の構築

ユニットの構成は前述のモデルと同じである．変移則に関しては，学習係数 α（0.5）と慣性項の係数 η（0.5）の値以外はすべて同じである．すなわち，入出力の部分を書き換えるだけで，異なるパターン認識のための ReproNet を構築することができるので，このモデルは汎用的であると言える．

4.2.4 ネットワークの構築

ネットワークの構成を図 4.11に示す．入力層と出力層のユニット数は固定である．通常の Neural Network の隠れ層は固定であるが，ReproNet の隠れ層は動的に変化する．実験では，隠れ層のユニット数が異なる複数の Neural Network を使用して，ReproNet と比較した．具体的には表 4.3に示す．

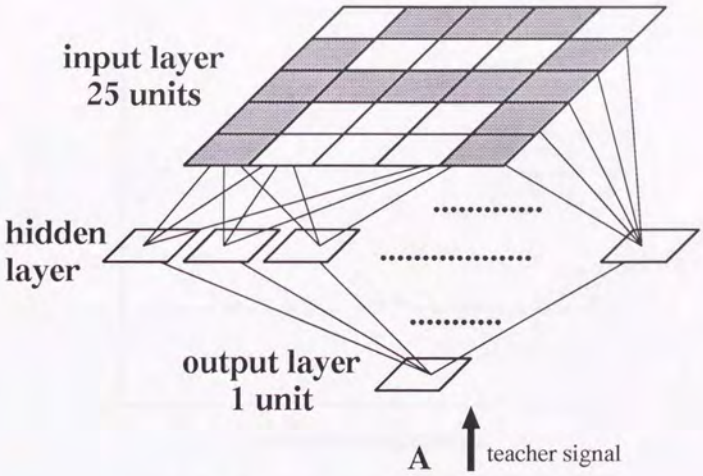


図 4.11 ネットワークの構成（文字認識）

入力層ではアルファベット 1 文字を表す 25 ドットのビットマップが入力される．1 ドットに対して 1 つのユニットが対応するので，入力層のユニットは 25 個である．1 ドットは，黒点を 1.0，白点を 0.0 として入力層のユニットに与えた．テストデータ用のドットのノイズは，黒点 1.0 と白点 0.0 の中間色であるが，ここでは 0.5 として実験した．

出力層では教師信号としてビットマップに対応する文字が入力される．1 文字の識別のために 1 つのユニットが対応するので，出力層のユニットは 1 個である．教師信号として，学習する文字を 0.0 から 1.0 の実数値に均等に配分した値に対応させた．例えば，A,B,C,D,E の 5 文字を学習する場合，出力ユニットの出力値 0.0, 0.25, 0.5, 0.75, 1.0 を各文字の教師信号に対応させた．したがって，n 個の文字列 $a_0a_1\cdots a_{n-1}$ を学習する場合，文字 a_i に対応する教師信号 $f(a_i)$ は，次の通りである．

$$f(a_i) = i/(n-1) \ \ (i = 0, 1, \ldots, n-1) \tag{4.14}$$

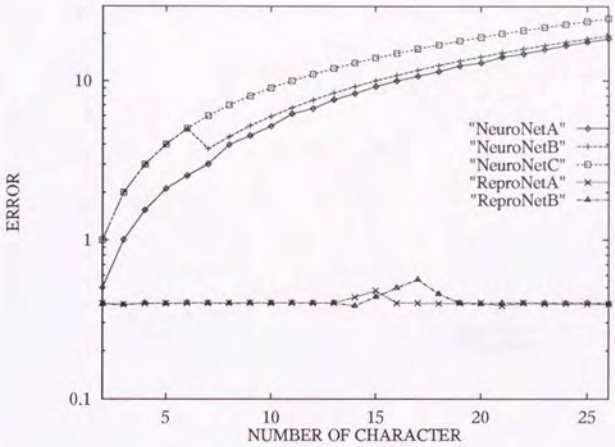


図 4.12 ReproNet と Neural Network の学習能力の比較（文字認識）

4.2.5 結果と考察

実験で使用したネットワークのユニット数を表 4.3に示す．3 個の通常の Neural Network と 2 個の ReproNet を比較した．

アルファベット 26 文字を学習する能力を ReproNet と Neural Network で比較した結果を図 4.12に示す．学習能力に関して暗号解読問題と同様の傾向が見られ，ReproNet は常に安定し性能を維持している．Neural Network はいずれも学習データに比例して誤差が上昇する．

図 4.13に，学習環境に応じて，ユニットが動的に増殖している様子を示す．

図 4.14に，ノイズデータに対する Neural Network と ReproNet の正答率を示す．

表 4.3 ネットワークのユニット数（文字認識）

network name	input	hidden	output
NeuroNetA	25	20 - 20 （2 層）	1
NeuroNetB	25	26 - 26 （2 層）	1
NeuroNetC	25	80 - 80 （2 層）	1
ReproNetA	25	6 （1 層）	1
ReproNetB	25	8 （1 層）	1

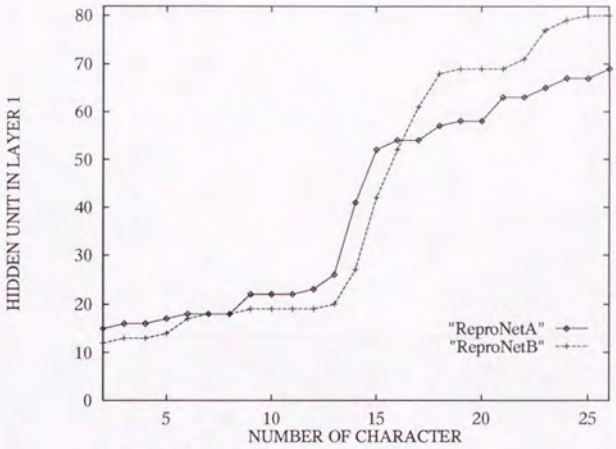


図 4.13 ReproNet の環境に応じたユニットの増殖

未知のデータに対する正答率も，ReproNet は環境の変化にネットワークが適応しているため，安定して 90% 程度の正当率を出しているが，Neural Network は 8 文字目に 85% を超えた後は，徐々に正当率が下る．

4.3 まとめ

変化する環境における ReproNet の学習能力を暗号解読問題と文字認識問題によって検証した．このモデルによって，ネットワークを柔軟に変化させる Neural Network を構築し，動的に変化する学習環境に対しても常に効率の良い学習を可能にした．また事前にネットワークの規模を決定する際の困難を削減できる．このことを暗号解読と文字認識問題という例によって示した．特に，UNIX の crypt に代表される暗号文の解読といった定式化の困難な非線形の系に対しても，優れた学習能力をもつことが示された．

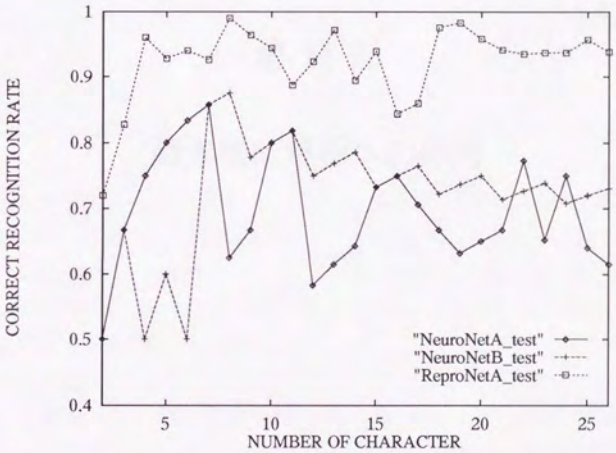


図 4.14 ReproNet と Neural Network の未知データに対する正答率

第 II 部

酵素機能解析への応用

第 5 章

酵素機能分類

酵素は生物の営むほとんど全ての反応に関係し，それらの反応をその生体の生存可能な穏和な条件下で円滑に実行させ，生命の維持に役立っている．酵素は，触媒として働く作用を選択する性質が強い．この酵素の特異性は基質と酵素の構造上の相互関係によって発揮される．このことは，酵素のアミノ酸 1 次配列上の局所的な情報を用いて，酵素機能の分類が可能であることを示唆している．

本章では，ReproNet を使用して局所的な 1 次配列を入力とした酵素の機能分類を試み，学習能力を検証した．生命分子配列の解析は，Neural Network の応用例の中でも最も成果を上げた分野となっている．しかしながら，タンパク質のデータは年々増加しており，計算時間や収束率の悪化が問題となっている．したがって，Neural Network はタンパク質のアミノ酸 1 次配列データのように増加しつつあるデータに対して構造を動的に変化させることが求められる．ここで，4章で構築した ReproNet を生命分子配列に適用し，変化する環境下において，アミノ酸 1 次配列から酵素を分類することを試みた．計算機実験では，学習データを徐々に増加させ，タンパク質データベースを模倣した．これによって，動的に増加する学習環境下での学習性能を評価した．結果として，ReproNet は Neural Network よりも優れた学習能力をもつことが示された．

5.1 酵素データ

酵素反応は，酵素が基質と一定の条件で相互作用し，触媒する化学反応と定義される．現在，一般的になっている NC-IUBMB[1]が提唱した酵素命名法 [52] では，酵素を反応の種類によって分類し，EC 番号（酵素クラス番号）が決められている．番号は 4 個の数字で表示し，第 1 の数によって表 5.1で示した 6 つのクラスのいずれ

[1]the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology

表 5.1 Enzyme Classes

EC-number	enzyme name	we used	database
1.	酸化還元酵素 Oxidoreductase	25	1600
2.	転移酵素 Transferase	28	1763
3.	加水分解酵素 Hydrolase	31	1923
4.	除去付加酵素 Lyase	10	646
5.	異性化酵素 Isomerase	2	147
6.	合成酵素 Ligase	4	276
	total	100	6355

に属するかを示す．第2，3の数字は反応をさらに細かく分類して付ける．第4の数字は1から3番目の数字によって分類された一群の酵素の中の通し番号である．新しく抽出された酵素には，NC-IUBMBによって，EC番号が割り当てられる．Enzyme Data Bank Release 18.0 1995に基づく酵素クラスの定義を付録Aに掲載する．

PIR(Protein Information Resource) [2] 等のデータベースにもEC番号，系統名，アミノ酸配列などとともに，多数の登録がある．表5.1の数値は，PIRに登録されている酵素（抽出源の異なる同一番号酵素を含む）の数とその割合を検索により求めたもので，各酵素の割合はEnzyme Nomenclatureのそれとほぼ一致している．これをみると分かるように，機能別の酵素の割合はEC番号が1から3の酸化還元酵素，転移酵素，加水分解酵素で全体の8割以上を占めている．

表5.1に主要6クラスごとの酵素数を示した．第3欄には本研究の学習で使用した酵素数を示したが，第4欄に示したデータベースから抽出した酵素数と同じ存在比率にしてある．学習では合計100個の酵素を使用した．学習に使用した全酵素の一覧を付録B.1に掲載する．アミノ酸配列の学習において，タンパク質間のホモロジーが精度に大きく影響する．このため，学習用の酵素間で偏りがないように選択し，平均20%のホモロジーを持つデータを使用した．

5.2 ReproNetの構築

変化する環境に対して，ネットワーク形態を動的に変化させる階層型ネットワークをReproNetを用いて構築し，momentum法による階層型NNと比較した．Re-

[2] PIR is a registered mark of NBRF and partially supported by the National Library of Medicine

proNet 変移則に従って，隠れ層のユニットを増殖し，高い性能を維持することが可能となっている．構築した ReproNet によって，酵素の分類を行い，学習能力を検証した．酵素データは年々増加しているので，動的に変化する環境下におけるネットワークの性能を評価した．

5.2.1 評価方法

最も普及している評価方法は，単純な正答率 q_4 である．これは 4 つのグループに対して酵素を正しく分類する率である．正答率は次の通り．

$$q_4 = \frac{p_1 + p_2 + p_3 + p_4}{n} \tag{5.1}$$

ここで，n は分類された酵素の合計数，p_1，p_2，p_3，および p_4 は，EC1，EC2，EC3，およびその他（EC4，EC5，EC6）の各クラスについて正しく分類された酵素の数である．

5.2.2 学習方法

1. 初期の学習データとして 10 個の酵素を学習データから無作為に選ぶ．入力として，アミノ酸配列を，教師信号として EC 番号を与える．

2. 学習規則に従ってパターンを学習させる．学習方法は momentum 法による結合荷重の更新である．次の条件を満たすと，学習が完了する．

$$q_4 > 80.0 \tag{5.2}$$

   ここで，q_4 は式 (5.1) によって定義される正答率である． 学習の完了の可否にかかわらず，規定回数（500 回ループ）に達した場合は学習を打ち切り，次のステップに進む．

3. 10 個の酵素を学習データから無作為に選択し，新しい学習データとして，ネットワークに追加する．

4. 100 個の酵素が学習されるまで，2. から 3. を繰り返す．

環境の変化を反映した自己増殖機能に関して，上記の手順のステップ 2. を詳述すると次の通りである．

2.1 出力層のユニットにおいて，現在のネットワークの学習能力を評価するために，各ユニットごとに教師信号との誤差から性能評価値を求める．

2.2 その評価値をネットワーク内の全ユニットに対して，入力信号と逆方向に伝播させる．

2.3 各ユニットは各自の内部状態や性能評価値に基づいて新しいユニットを増殖するかどうか決定する．

5.2.3 ユニットの構築

内部状態は次のように構築した．

$$Q = (in, wait, ready, out, bias) \tag{5.3}$$

ここで，in：入力信号を受け入れる状態，out：教師信号を受け入れる状態，$bias$：常に一定の閾値を出力する状態，$wait$：ユニットを増殖させない状態，$ready$：ユニットを増殖させる状態である．ユニットの初期状態は，入力層：in，隠れ層：$wait$，出力層：out，閾値ユニット：$bias$ とした．内部状態 $ready$ は増殖機能を持つユニットだけに与える．

出力関数には，次の Sigmoid 関数を用いた．

$$g(x) = 1/(1 + e^{-x}) \tag{5.4}$$

5.2.4 ネットワークの構築

ネットワークの構成を図 5.1に示す．ネットワークの入力ノードは入力層と出力層であり，内部ノードは隠れ層である．入力層と出力層のユニット数は固定である．通常の NN の隠れ層は固定であが，ReproNet の隠れ層は動的に変化する．実験では，隠れ層のユニット数が異なる 7 個の NN と 3 個の ReproNet を使用し比較した．具体的には表 5.2に示す．入力層のユニットを上部に，出力層のユニットを下部に配置した．入力層のユニットは隠れ層のユニットに結合し，隠れ層のユニットは出力層のユニットに結合している．

ネットワークには連続した 11 のアミノ酸配列を与える．ネットワークは中央のアミノ酸に関して正しく酵素を分類するように学習される．ネットワークは一度に 11 個のアミノ酸を入力するウィンドウを持ち，酵素のアミノ酸配列上を走査する．入力をウィンドウに限定するのは，ある特定のローカルなアミノ酸配列が全体の酵素の機能を決定しており，それ以外の配列はノイズ的な情報であることが考えられるためである．タンパク質の 2 次構造予測などでは，一般的によく用いられている方法である．

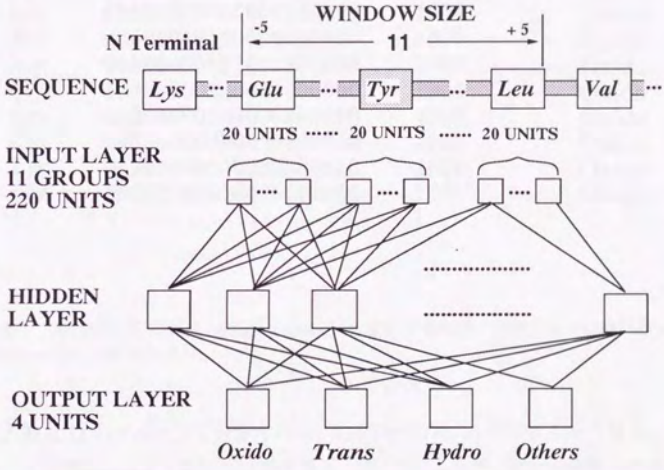


図 5.1　酵素機能分類用に構築した ReproNet のネットワーク構造．11 の連続するアミノ酸残基が 1 度に入力される．1 残基につき 20 の入力ユニットが割り当てられる．入力された情報は隠れ層を経て，出力層に伝達される．出力層では 4 個のユニットは各々 EC 番号 1，2，3，4 から 6 を表し，出力値にしたがって，酵素クラスを分類する．

Sequences	Input Data	Teacher Signal	Enzyme Class
.	.....	.	.
.	.....	.	.
.	.....	.	.
Lys	00000000000000100000	1000	Oxido
Glu	00000000000010000000	1000	Oxido
Gly	10000000000000000000	1000	Oxido
Tyr	00000000000000000010	1000	Oxido
Leu	00010000000000000000	1000	Oxido
Val	00100000000000000000	1000	Oxido
Lys	00000000000000100000	1000	Oxido
Lys	00000000000000100000	1000	Oxido
Ser	00000010000000000000	1000	Oxido
Asp	00000000001000000000	1000	Oxido
Gly	10000000000000000000	1000	Oxido
Cys	00000000100000000000	1000	Oxido
Lys	00000000000000100000	1000	Oxido
Tyr	00000000000000000010	1000	Oxido
Gly	10000000000000000000	1000	Oxido
.	.....	.	.
.	.....	.	.
.	.....	.	.

図 5.2　入力データ（local coding scheme）と教師信号の例．酵素クラスは酸化還元酵素（Oxidoreductase）である．.

入力層は 11 のグループに分類される．各グループは 20 ユニットからなる．すなわち，入力層のユニット数は 220 個である．図 5.2に入力データと教師信号の例を示す．各グループ内の 1 個のユニットがその位置にあるアミノ酸に対応し，入力値として 1 が与えられる．残りの 19 のユニットには 0 を入力する．

例えば，グリシンは 10000000000000000000，アラニン 01000000000000000000 である．これは，local coding scheme と呼ばれている．

出力層は 4 個のユニットで構成され，各ユニットが酵素クラスを表す．酵素クラスは，1) 酸化還元酵素 (Oxidoreductase)，2) 転移酵素 (Transferase)，3) 加水分解酵素 (Hydrolase)，および 4) 残りの除去付加酵素 (Lyase)，異性化酵素 (Isomerase)，合成酵素 (Ligase) のように 4 つに分類した．4 番目のグループは自然界に存在する割合が他の 3 クラスと比較して少ないので同じグループにまとめた．与えられた入力値とネットワークの結合荷重に対して，出力ユニットは 0 から 1 の値を出力する．教師信号は，EC1 に対して 1000，EC2 に対して 0100，EC3 に対して 0010，残り

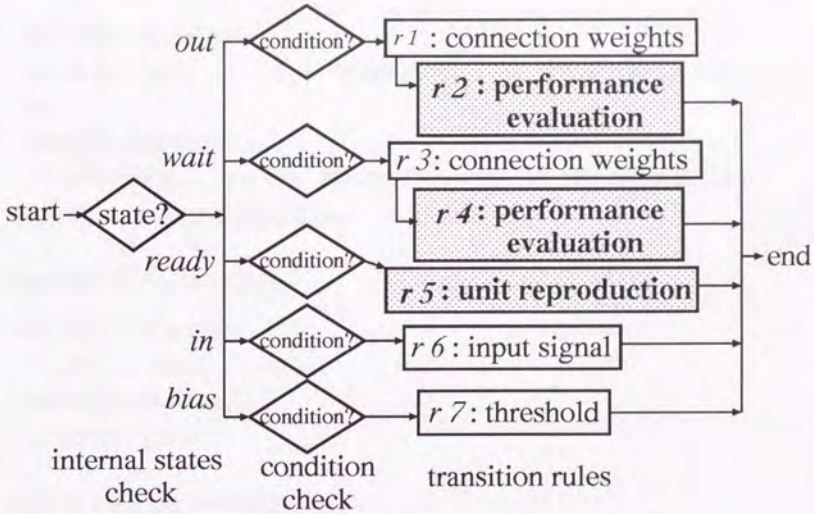


図 5.3 変移則の流れ図．増殖に関係する変移則を網がけで記した．

の EC4，EC5，EC6 に対して 0001 を与えた．ウィンドウ内の中央のアミノ酸残基に対して，最大値を出力する出力ユニットの酵素クラスを選択する．従って，1 の酵素に対して，アミノ酸残基の数だけ各酵素クラスの出力値が存在することになる．各アミノ酸残基ごとに出力された値を 1 次配列全体で合計して，最大値の酵素クラスを選択する．

5.2.5 変移則の構築

図 5.3 に変移則の流れ図を示す．変移則 r_2, r_4, r_5 はユニットの増殖に関係している．変移則 r_2 と r_4 は現在の環境におけるネットワークの性能を評価する．変移則 r_5 はユニットの増殖を制御する．

ReproNet の隠れ層のユニットは 2 つの内部状態を持つ．通常は *wait* であるが，学習性能が乏しいと *ready* に変化し，新しいユニットを増殖する．

以下に適用される内部状態ごとに個々の変移則を説明する．

内部状態 out のときの変移則 $r_1 r_2$

結合荷重に関する変移則 r_1

4.1.5節の r_1 と同じ．但し，η_j：学習係数 (0.0005)，α_j：慣性項の係数 (0.0005) である．

性能評価に関する変移則 r_2

4.1.5節の r_2 と同じ．本節では，層は増殖しないので，4.1.5節の層の増殖に関する変移則 r_3 に相当する変移則はない．

内部状態 $wait$ のときの変移則 $r_3 r_4$

結合荷重に関する変移則 r_3

4.1.5節の r_4 と同じ．

性能評価に関する変移則 r_4

4.1.5節の r_5 と同じ．

内部状態 $ready$ のときの変移則 r_5

ユニットの増殖に関する変移則 r_5

4.1.5節の r_6 と同じ．

内部状態 in のときの変移則 r_6

入力信号に関する変移則 r_6

4.1.5節の r_7 と同じ．

内部状態 $bias$ のときの変移則 r_7

閾値に関する変移則 r_7

4.1.5節の r_8 と同じ．

5.3 結果と考察

隠れ層のユニット数が異なる ReproNet と Neural Network を複数個用意して，momentum 項を持つ逆伝播法により学習を行った．表 5.2に隠れ層のユニット数の初期値を示す．ネットワークの隠れ層は 1 個である．ReproNet の隠れ層のユニットは変移則に従って増殖する．

表 5.2 初期ネットワーク内のユニット数

network name	input	hidden	output
NeuroNet10	220	10	4
NeuroNet20	220	20	4
NeuroNet30	220	30	4
NeuroNet40	220	40	4
NeuroNet50	220	50	4
NeuroNet100	220	100	4
ReproNet10	220	10	4
ReproNet20	220	20	4
ReproNet50	220	50	4

ReproNet は環境に応じた学習能力および計算機資源の効率的使用において顕著な性能向上を示した.

5.3.1 学習能力

ReproNet と Neural Network の酵素クラスを学習する能力を比較した.

NeuroNet10,20,30,40,50 の学習能力を図 5.4に示す. NeuroNet はすべて学習能力に限界がある. 学習能力を超えると, 分類能力が低下する. 例えば, NeuroNet10 は学習データの酵素が 10 個の環境では, 正しく分類する率は 80% であるが, 酵素データが徐々に増加すると, 性能が低下する. このことは従来の Neural Network の性能は初期ネットワーク構造に依存していることを意味している.

これに対して, ReproNet10,20,50 の学習能力を図 5.5に示す. 学習する酵素データが増加すると, 式 (4.8) および (4.13) で表される性能評価値 $\tilde{e}_j$ が大きくなり, ネットワークの学習能力が落ちていることを表す. ReproNet は環境のデータサイズの増加に応じて, 隠れ層のユニットを動的に 10 から 74 程度まで増殖させている. 結果として, ReproNet は性能を良い状態に保っている.

5.3.2 ユニットの増殖

図 5.6に学習環境の増加に応じたユニットの動的な増殖を示す. 隠れユニット数が 10 である ReproNet10 は, 学習酵素データが 10 から 100 に増えるまで, 一定の勾配で隠れユニットを増殖している.

隠れユニット数が 20 である ReproNet20 は, 学習酵素データが 30 個になった後,

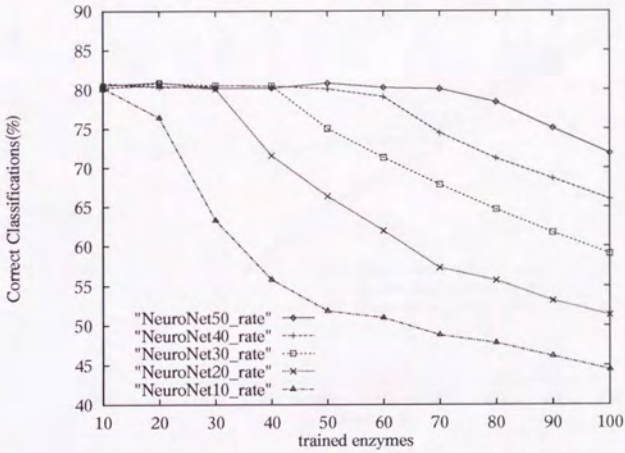


図 5.4　Neural Network の学習能力．正答率と学習データの酵素数をプロットした．学習する酵素データが増加するにしたがって，性能が低下する．

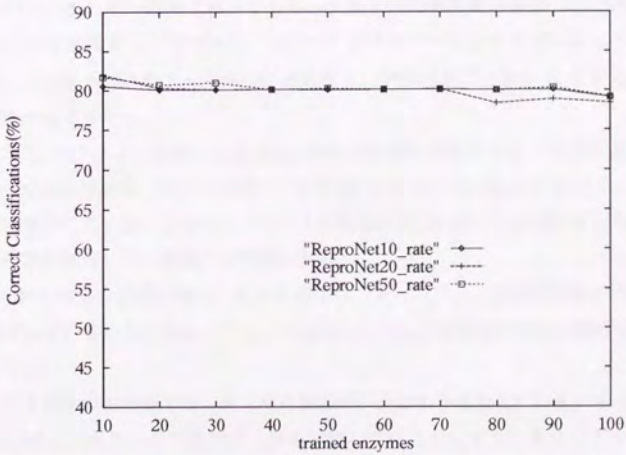


図 5.5　ReproNet の学習能力．ReproNet は環境が変化しても高い性能を維持している．

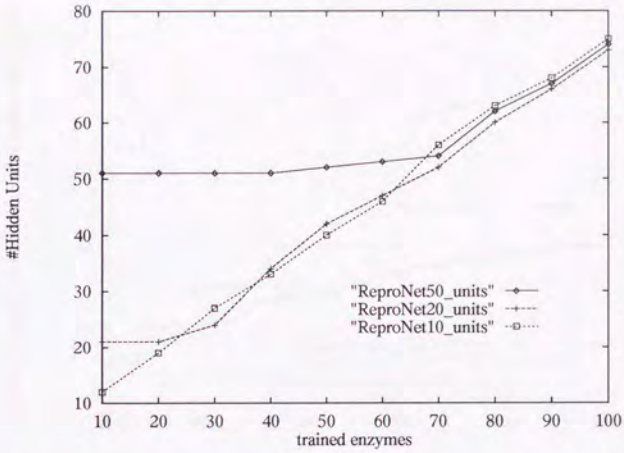


図 5.6　環境に応じたユニットの増殖．隠れユニット数と学習酵素データ数をプロットした．ReproNet10 の初期隠れユニット数は 10 で，ReproNet20 および 50 は，各々 20 および 50 である．

ReproNet10 と同じ勾配でユニットを増殖している．これは，学習酵素データ数に対して 10 から 30 の早い段階ではネットワークは十分の学習能力を持つので増殖が起きないことを意味する．すなわち，ユニットは学習データのサイズに応じて増殖している．ReproNet50 も ReproNet30 と同様に，学習酵素データが 70 以降にユニットの増殖が起きている．

図 5.7に，ユニットの増殖による ReproNet の性能の回復を示す．反復回数 (iteration times)180 付近で，新しく学習データが 30 から 40 に増加しているので，エラーが急激に増加している．しかし，ユニットが環境の変化に応じて増加しているので，瞬時に性能が回復している様子が観察される．

ReproNet が高い性能を維持している理由は，ネットワーク構造が環境の変化に動的に適応しているからである．一方，Neural Network は環境の変化に適応することができない．

特に分子生物学の分野では，タンパク質のデータは年々増加している．ネットワークがその時点でのタンパク質のデータを学習していたとしても，数多くのタンパク質がデータベースに加わり，すぐにネットワークの学習能力を超えてしまう．ネットワーク形態が柔軟に環境に適用することがネットワークに必要であり，ReproNet

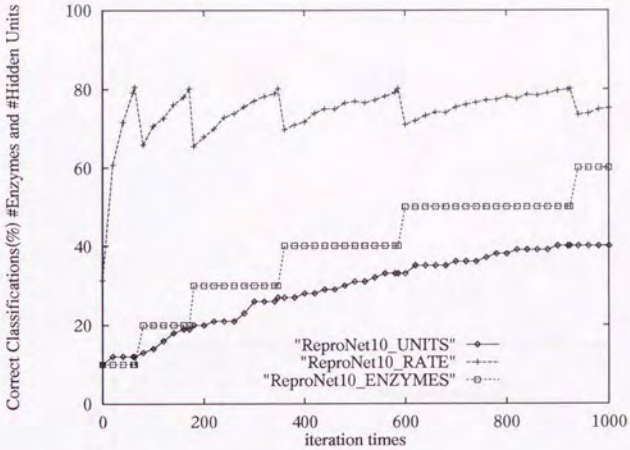


図 5.7　環境の変化に対する ReproNet の隠れユニット数の推移．正答率と学習回数 (iteration loops) をプロットした．学習データが増加すると，隠れユニットが増殖し，正答率が即座に回復する．

はそれを実現している．

5.3.3　計算機資源の消費

隠れユニット数が 100 である NeuroNets100 は，100 の酵素データを学習するのに十分の能力を持つ．しかしながら，余分のユニット数を持っていると，計算機資源の負荷が高くなる．図 5.8に学習酵素データが増加するにしたがって，1 学習ステップ当たりの CPU 時間の変化を示した．従来の Neural Network は ReproNet よりも多くの CPU パワーを必要とすることが分かる．また，使用記憶容量もユニット数に比例するため，隠れ層のユニットに関しては，NeuroNet100 は ReproNet10 の 10 倍のメモリを消費することになる．これは計算機資源の観点から見て，NeuroNet100 は，この環境下ではサイズが過剰であることを意味する．さらに，ネットワーク形態が固定されていると学習能力に限界がある．したがって，学習データが増加し，学習能力を超えると，正答率が急速に低下すると思われる．タンパク質データが年々増加している事実からすると，計算機資源の有効利用は考慮に値するといえる．

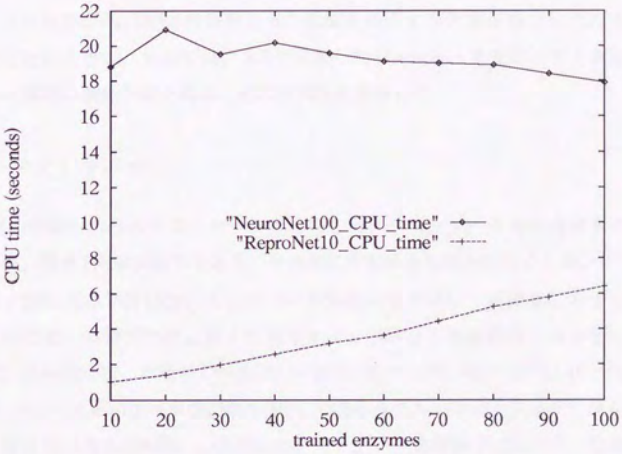


図 5.8 1 酵素，1 学習ステップ (iteration time) 当たりの CPU 時間（秒）

第 6 章

酵素機能予測

酵素の機能が直接そのアミノ酸配列から予測できれば，新たに発現した酵素に対して，1 次配列から目的の機能をもった酵素を抽出する作業が簡素化されるなど益するところが大きい．本章では，5章で構築した ReproNet を使用して 1 次配列を入力とした酵素の機能予測を試み，その有用性を検討した．

6.1 テストデータ

50 個の酵素からなるテストデータを使用した．テストデータ内の各酵素クラス数の比率は，表 6.1に示す通りであり，自然界に存在する比率と等しくした．

アミノ酸配列の学習において，タンパク質間のホモロジーが精度に大きく影響する．このため，学習データに対するホモロジーが異なる複数個のテストデータを用意した．具体的には，ホモロジーが 10 ～ 20%, 20 ～ 30%, 30 ～ 40%, 40 ～ 50%, 50 ～ 60%, 60 ～ 70%, 70 ～ 80%, 80 ～ 90% である 9 グループのテストデータに対して予測を行った（表 6.2参照）．使用したテストデータを付録 B.2に示す．なお，2 次

表 6.1 Enzyme Classes

EC-number	enzyme name	test data	database
1.	酸化還元酵素 Oxidoreductase	13	1600
2.	転移酵素 Transferase	14	1763
3.	加水分解酵素 Hydrolase	15	1923
4.	除去付加酵素 Lyase	5	646
5.	異性化酵素 Isomerase	1	147
6.	合成酵素 Ligase	2	276
	total	50	6355

表 6.2 Enzyme Classes

test dataset	homology for training data
test data1	10 ～ 20%
test data2	20 ～ 30%
test data3	30 ～ 40%
test data4	40 ～ 50%
test data5	50 ～ 60%
test data6	60 ～ 70%
test data7	70 ～ 80%
test data8	80 ～ 90%

構造予測の場合に関しては，Zhang らは [53] ホモロジーが 50% 以下であれば，予測精度への影響は少ないことを報告している．

6.2 入力アミノ酸配列の限定

酵素は特定の基質の遷移状態に強く結合して化学反応速度を増大させる．この酵素の特異性は，特定のアミノ酸が酵素の基質特異性と触媒反応の両方について重要な役割を示すことを意味している．この事実は酵素のアミノ酸配列の中で，特定の部位が酵素機能を特徴付けることを示唆している．従って 1 次配列の中で，ネットワークが高い出力を出す部位を抽出し，そうでないものを学習対象から外すという操作によって，酵素機能の特徴的な部位を抽出し，余分な残基を排除できる可能性がある．具体的な手順は次の通りである．

1. 学習用酵素データに対して学習する．入力として，アミノ酸配列を，教師信号として酵素番号を与える．ここでは，5.2.5節で構築した ReproNet の変移則を使用する．

2. 各酵素のアミノ酸配列の出力値と教師信号の誤差が最も大きい残基を 1 つ選び，学習対象から外す．ただし，前もって残基数の最小値を決めておく．すなわち，各酵素はその最小値の残基数だけ残して学習を行う．

3. 規定された回数まで，1. から 2. を繰り返す．

6.2.1 学習能力

全ての残基列を学習したネットワーク，100 個の残基列を残して学習したネットワーク，および 200 個の残基列を残して学習したネットワークの学習性能を比較した．図 6.1に上記の 3 種の ReproNet の学習能力の推移を示す．図 6.2には，隠れ層のユニットの増殖を示す．学習データである残基数が少ないほど，ネットワークは学習能力を高く保つことができる．100 残基を学習するネットワークはユニット数が 58 で増殖が止まっている．200 残基および全残基を学習するネットワークは各々 65，68 まで断続的に増殖している．これは，ネットワークが 100 残基のデータを学習するには，58 程度の隠れユニット数で十分であることを示している．

学習対象の残基数が少ないほど，学習データに特異的な情報を学習することになるので，未知のデータに対する予測率の低下を招くことになる．これとは逆に，学習対象の残基数を増やすと，より広範なアミノ酸配列の情報を学習することになるが，学習データの増加に伴う学習性能の低下が問題となる．図 6.2に示した ReproNet の増殖状況から判断すると，全残基を学習させたネットワークについては，隠れ層の増殖が継続している様子が示されている．これは，学習が十分になされていないことを意味する．一方，200 残基を学習するネットワークはイタレーションが 1000 を超える当たりで，ユニットの増殖が抑えられており，この段階で学習が十分になされていることが示唆される．これは，イタレーションが 300 ほどでユニット数が安定している 100 残基を学習したネットワークより，学習時間がかかっていることを意味する．

6.2.2 予測性能

前述の 3 種のネットワークについて，テストデータに対する予測性能を比較した．表 6.2に示した学習データに対するホモロジーが異なるテストデータの予測率の推移を図 6.3に示した．また，図 6.4には，ホモロジーが高い（50% から 90%）データと低い（10% から 50%）データの予測率の平均を示した．

100 残基および 200 残基を学習したネットワークが高い予測率を出しているのに対して，全残基を学習したネットワークは全体的に低い予測率に留まっている．100 残基のネットワークは，ホモロジーが高い酵素に対して 200 残基のネットワークよりも高い予測率を出している．しかし，ホモロジーが 50% を境に，ホモロジーの低い酵素に対しては，200 残基のネットワークが高い予測率を出している．

6.2.1節で考察した学習能力の比較と総合すると，学習対象として限定する残基数が少ないほど，局所的なアミノ酸配列の情報を学習しやすくなる．このため学習能

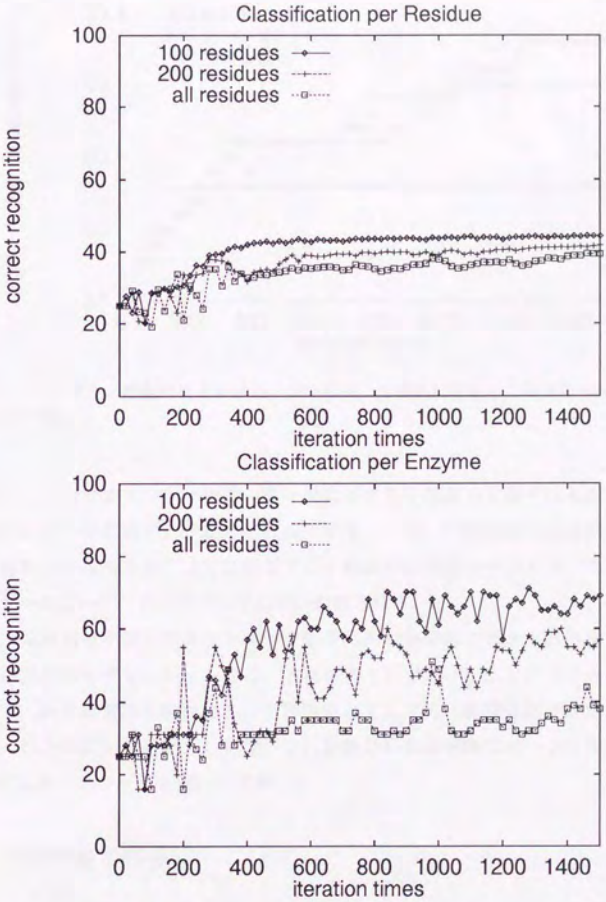


図 6.1　入力アミノ酸配列を 100 残基，200 残基，全残基に限定した場合の ReproNet の学習能力の推移．上が残基ごとの正答率で，下が酵素ごとの正答率．

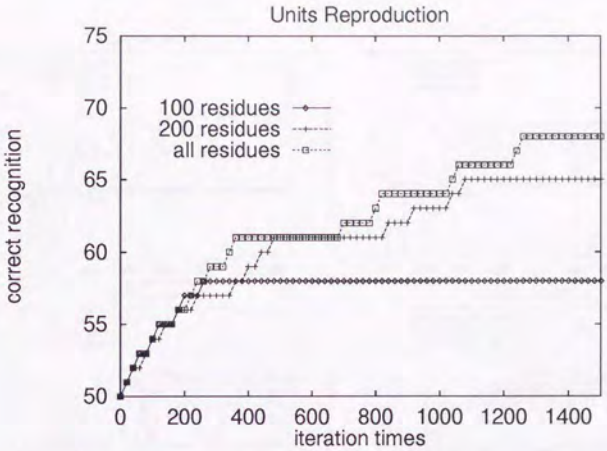


図 6.2　入力アミノ酸配列を 100 残基，200 残基，全残基に限定した場合の ReproNet のユニットの増殖．

力が増加し，またホモロジーの高いデータに対する予測能力も高くなるが，ホモロジーの低いデータに対する予測性能は低下する．一方，学習対象の残基数を増やすと，学習能力が落ちるが，より広範なアミノ酸配列の情報を学習する．このため，ホモロジーの低いデータに対する予測率が改善される．

なお，全残基を学習したネットワークの予測率が全体的に低かった理由は，今回の計算機実験のイタレーションでは，学習が不十分であったことが考えられる．したがって，計算機資源を検討して，学習対象とするアミノ酸残基数を決定する必要がある．以上のことを検討した結果，これ以降の計算機実験では，200 残基を残して学習するネットワークを用いて考察した．

6.3　予測性能の評価

ReproNet と Neural Network の予測性能を比較する計算機実験を行った．いずれも ReproNet の予測性能が優れていた．

図 6.5には各々 ReproNet と隠れ層に 100 個のユニットを持つ Neural Network の学習と予測率の推移を示した．Neural Network が学習率の増加に対して過学習が発生し，予測率の上昇が停止しているが，ReproNet は過学習を回避し，予測率が変

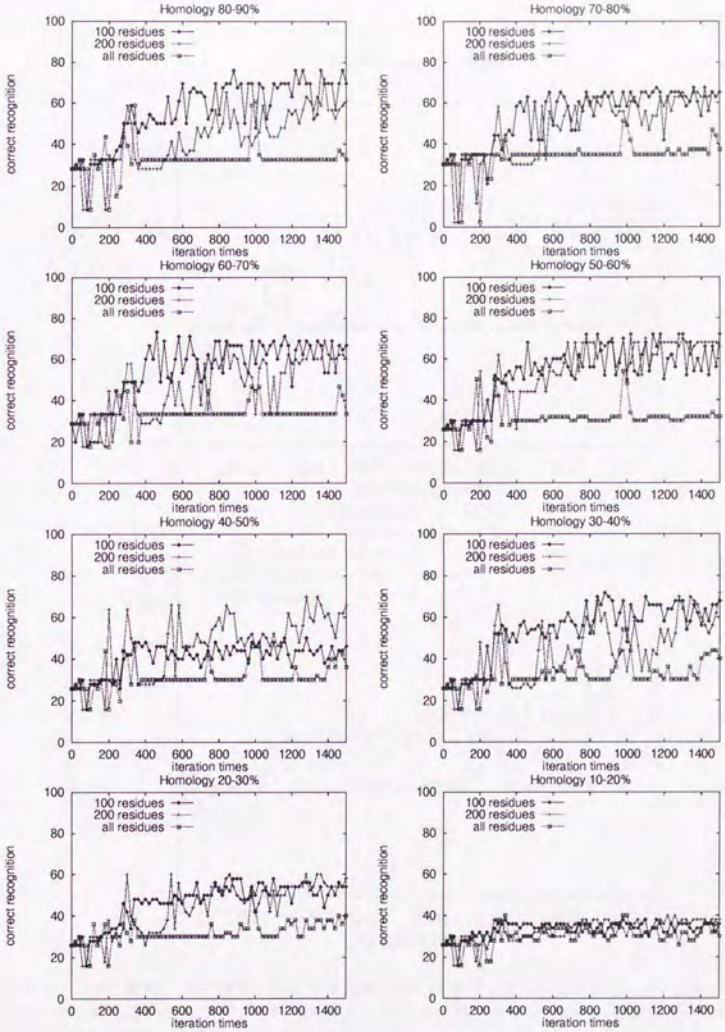


図 6.3　100 残基，200 残基，および全残基を残して学習した ReproNet による予測．学習データに対するホモロジーが 10% から 90% までのテスト用酵素の予測率の推移を示した．

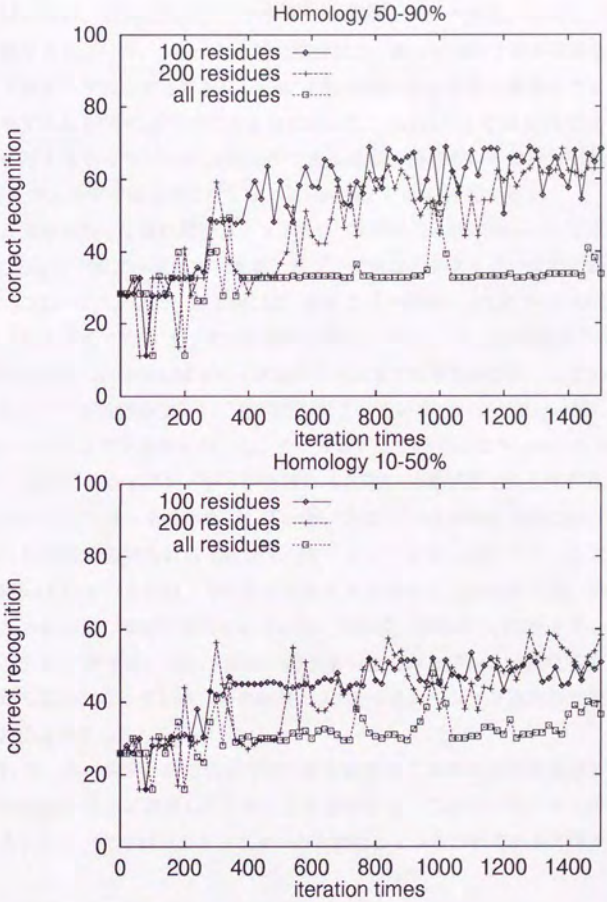


図 6.4　100 残基，200 残基，および全残基を残して学習した ReproNet による予測．学習データに対するホモロジーが 50% から 90% まで（上図）および 10% から 50% まで（下図）のテスト用酵素の予測率の平均の推移を示した．

動しながら上昇している様子が観察された．図 6.6にはホモロジーが 40% 未満のテストデータの予測率の推移を示した．テストデータのホモロジーが低くなるにしたがって，予測率も低下した．ホモロジー検索による予測では，長い計算時間が必要であるばかりか，既存の酵素とテストデータのホモロジーが低い場合は，機能の予測が困難である．一方，ReproNet では短時間で，ある程度の予測が可能である．例えば，学習データに対するホモロジーが 50% 未満である酵素の機能をホモロジー検索で予測すると 46% の正答率であるのに対して，ReproNet では 64% であり，学習データに対するホモロジーが 20% 未満である酵素の機能をホモロジー検索で予測すると 30% の正答率であるのに対して，ReproNet では 44% である．

次に，ReproNet と隠れ層のユニット数が 70 個の Neural Network を比較した．表 6.2に示した学習データに対するホモロジーが異なるテストデータの予測率の推移を図 6.7には示した．また，図 6.8には，ホモロジーが高い（50% から 90%）データと低い（10% から 50%）データの予測率の平均とユニットの増殖の関係を示した．

従来の Neural Network は 400 イタレーションまでは予測率が向上しているが，その後，徐々に予測性能が低下し，過学習を起こしている．一方，ReproNet は，300 イタレーションまで予測率が向上し，その後低下している点は Neural Network と類似しているが，500 イタレーション付近から性能の回復が見られ 800 から 900 イタレーション付近で高い予測率を出している．ユニットの増殖は性能の回復と関連している．性能低下の見られる 300 から 500 イタレーションにかけて，ユニットが活発に増殖している．その後，500 から 800 イタレーションにかけては，予測率の回復が見られるので，増殖は抑制されている．その後，900 から 1000 イタレーションにおいて，再び過学習による予測率の低下が生じているが，これに応じて，ユニットの増殖も活発になっている．その結果，1000 イタレーション以降の予測率の回復が見られるようになる．

すなわち，ネットワークの学習性能の変化に基づくユニットの増殖がテストデータの予測性能の向上に寄与していることを意味する．これは，ユニットが増殖する際に，ネットワークがローカルミニマムから脱出しているためであると考えられる．

6.4 過学習の回避方法

過学習の原因の 1 つは，ネットワークが学習データの持つ局所的な特徴を学習し，テストデータに見られる一般性を失うことが考えられる．一般的に過学習を回避し，予測性能を向上させるために，問題の性質に関する情報を広範囲に包含する学習データを用意する方法がある．そこで，学習データを 2 倍にして学習を行い，予測率の

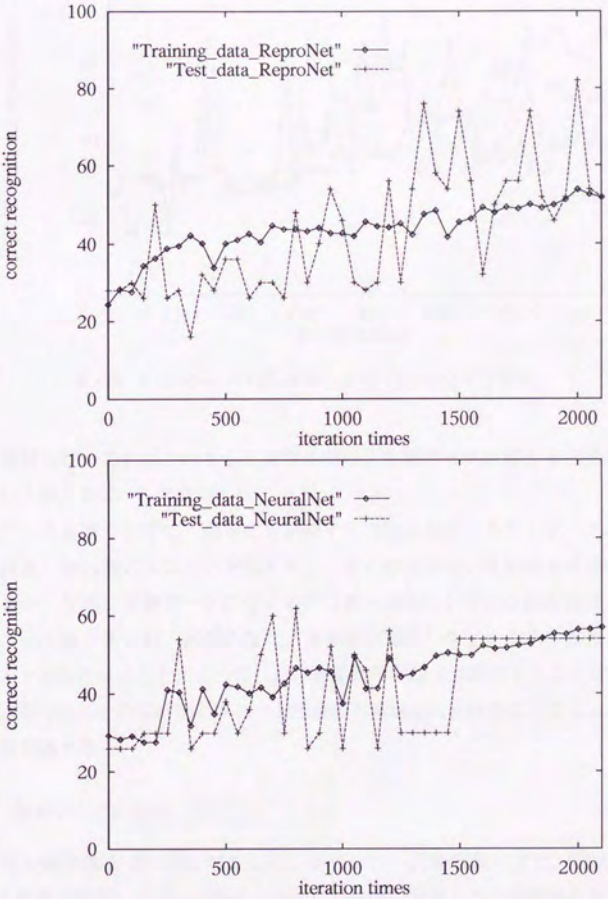


図 6.5 ReproNet（上）と Neural Network（下）の学習と予測.

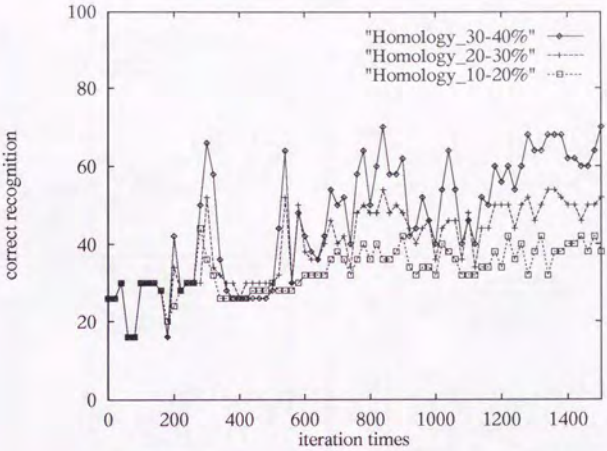


図 6.6 ReproNet の 40% 未満のホモロジーごとの予測率.

改善を観察した．これは手法としては簡単だが，学習データが増えると学習に時間が比例して増えるという欠点がある．

学習データを増やさずに，過学習を回避する方法として，ネットワークの収縮が挙げられる．隠れ層のユニットが増えると，多くのパターンを学習させることが可能であるが，反面，学習データに強く依存し偏ったネットワークが形成される．これはユニット数が多い程，顕著になり，過学習が発生しやすくなる．したがって，ユニットを減少させることによって，過学習をある程度まで回避することができる．そこで，ReproNet の変移則にユニットの削除と増殖を組み込むことによって，予測率の改善を図った．

6.4.1 学習データの増加

学習用の酵素数を 200 個に増やして，ネットワークを学習させた．図 6.9には，200 個の酵素で学習した ReproNet と 100 個の酵素で学習したの予測率の変化を示した．また，図 6.10には，ホモロジーが高い（50% から 90%）データと低い（10% から 50%）データの予測率の平均を示した．

1000 イタレーションまでは，両者ともほぼ同等の予測率を示しており，大きな差は見られない．1000 から 1500 イタレーションにかけて，200 個の酵素で学習した

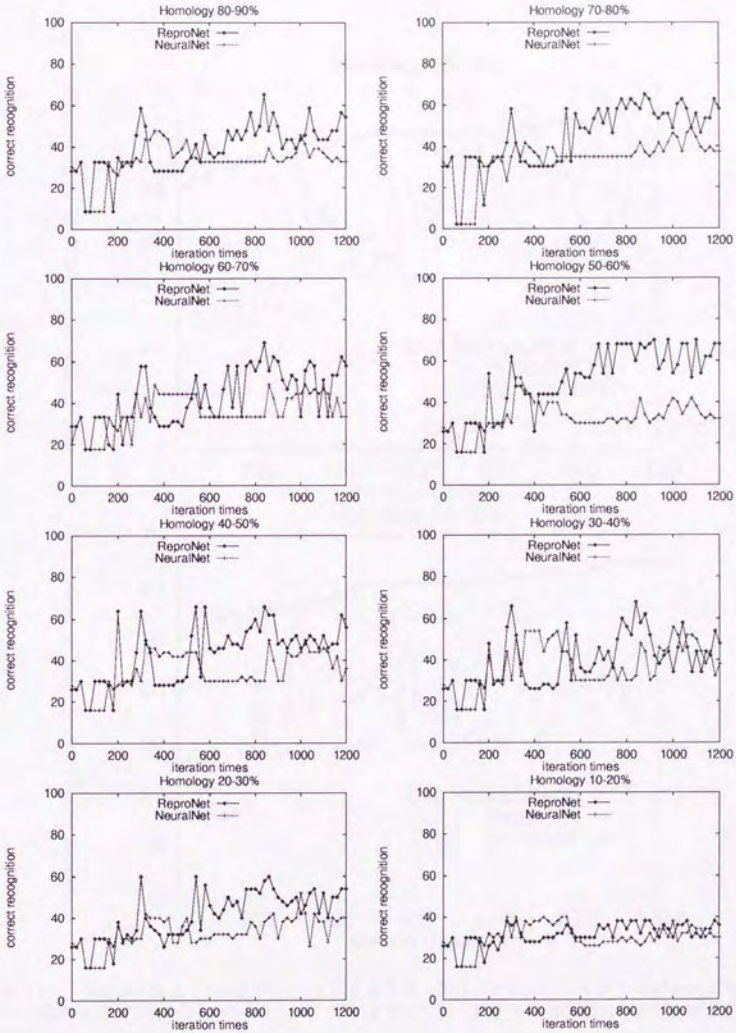


図 6.7 ReproNet と Neural Network による予測．学習データに対するホモロジーが 10% から 90% までのテスト用酵素の予測率の推移を示した．

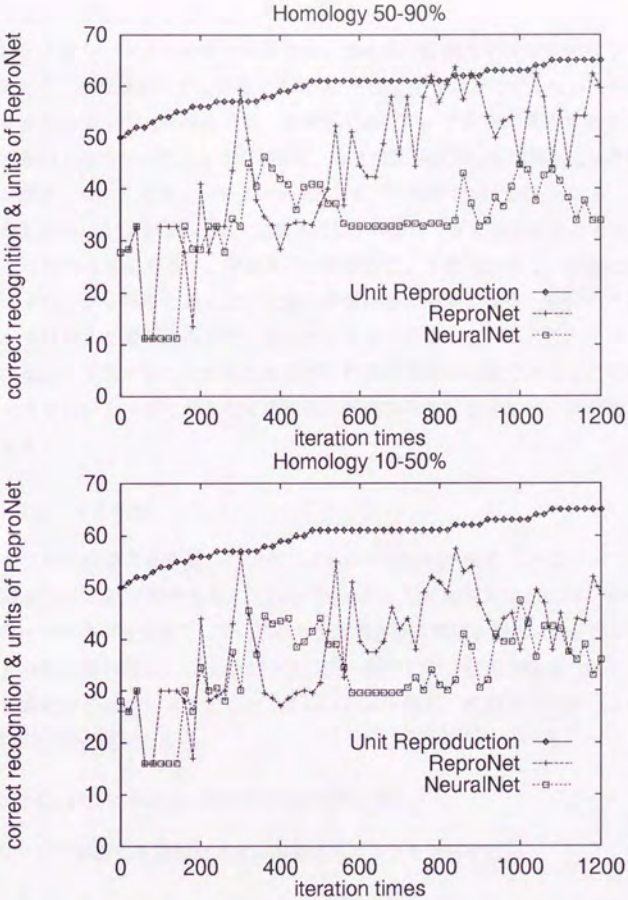


図 6.8　ReproNet と Neural Network による予測．学習データに対するホモロジーが 50% から 90% まで（上図）および 10% から 50% まで（下図）のテスト用酵素の予測率の平均の推移と ReproNet ユニットの増殖を示した．

ReproNet の予測率が低下している．しかし，1500 イタレーションを越えると，100 個の酵素で学習した ReproNet の予測率が低下し始め，以降，目だった予測率の回復が見られないが，200 個の酵素で学習した ReproNet では，予測率の回復が見られ，より優れた性能を示している．

これは，イタレーションの早い段階では，200 個の酵素を学習するには不十分であるため，予測の性能が十分に発揮できない．一方，学習が進行するにしたがって，学習データの少ない ReproNet では，過学習を起こし，予測率が低下するが，学習データの多い ReproNet は，より広範囲にアミノ酸配列の情報を学習しているので，過学習が起きにくく，イタレーションが進んでも，予測率の向上が見られる．

過学習を回避して，予測率の向上を計るには，学習データを増やすことによって，ある程度は解決可能であるが，学習の早い段階では，予測率が低く，実際に使用可能なネットワークを得るためには，大量の学習時間が必要となる．実際のタンパク質のデータは膨大な量があるので，理想としては現存するタンパク質のアミノ酸配列データを全て学習することが望まれるが，計算機資源が有限であることを考慮すると，より少ないデータで偏りなく広範囲の配列情報を包含するデータを選択する必要がある．

6.4.2 ユニットの削除

ネットワークが過学習を起こしたら，ユニットを削減させることによって，過学習を回避することが可能である．これを ReproNet で実現するためには，変移則にネットワークが過学習を起こしていることを監視させる機能を持たせる必要がある．そこで，今回は隠れ層のユニットが一定期間，増殖しない状態が続くとユニットを規定した数だけ削減するようにした．古いユニットほど，削除されやすいようにした．変移則は次の通りである．

- 条件 C: 200 iterations 間ユニットが増殖しない．
- 動作 W: 規定した個数だけ古い順番にユニットを削除する．

図 6.11には，ユニットを 5 個ずつ減少させる機能を持つ ReproNet の予測率とユニット数の変化を示した．また，図 6.12には，ホモロジーが高い（50% から 90%）データと低い（10% から 50%）データの予測率の平均を示した．イタレーションが 480 から 660 である間，ユニット数が一定の状態が継続する．そこで変移則に従って，イタレーションが 680 のときにユニット数が 5 個減少している．

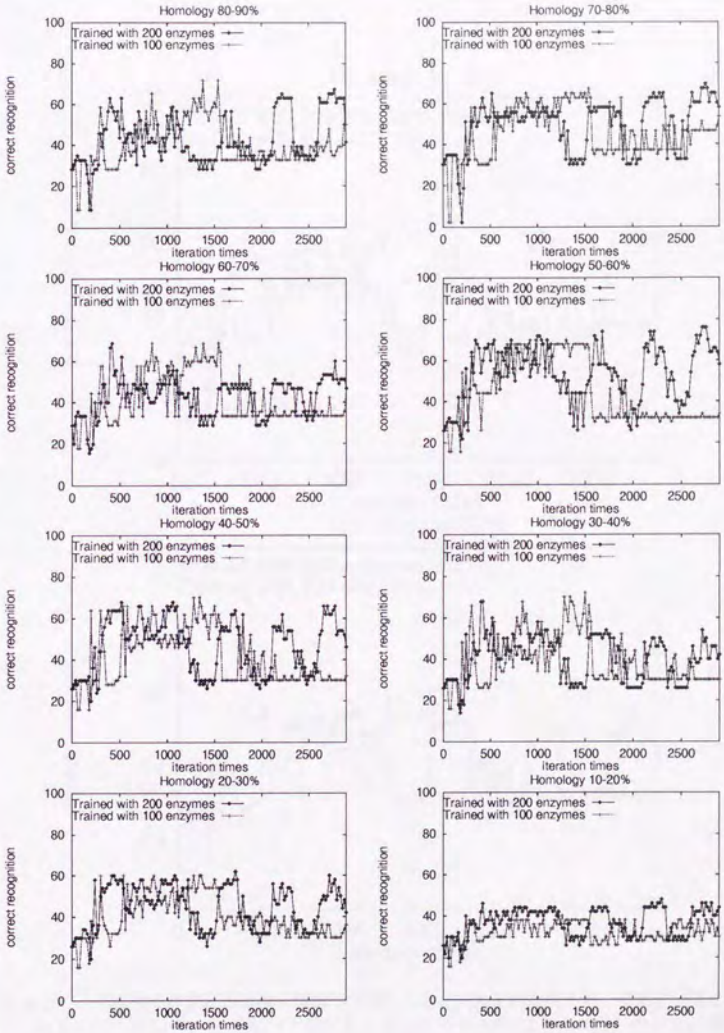


図 **6.9**　200 個の酵素で学習した ReproNet による予測と 100 個の酵素で学習した ReproNet の予測率．学習データに対するホモロジーが 10% から 90% までのテスト用酵素の予測率とユニット数の推移を示した．

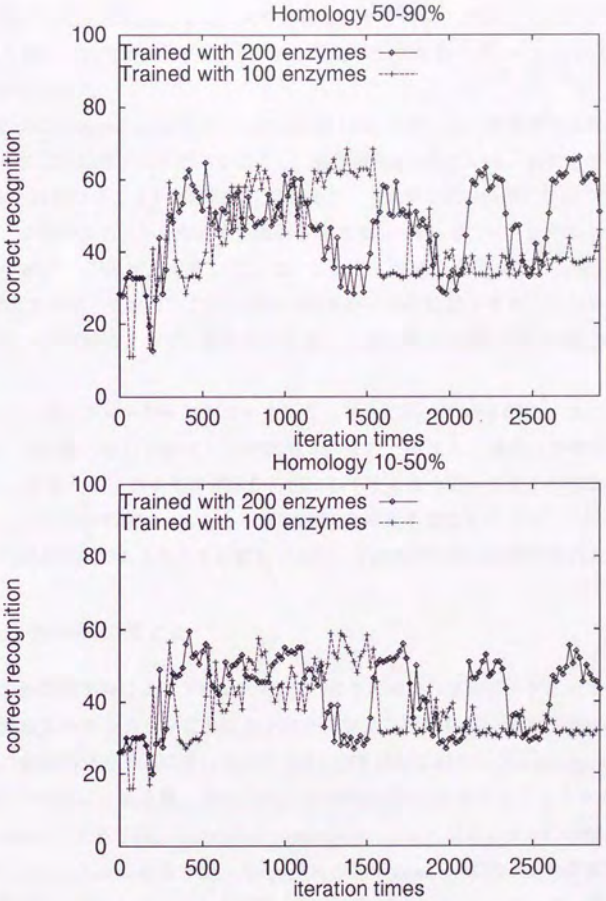


図 **6.10**　200 個の酵素と 100 個の酵素で学習した ReproNet による予測．学習データに対するホモロジーが 50% から 90%（上図）および 10% から 50% まで（下図）のテスト用酵素の予測率の平均の推移を示した．

それに伴い，一時的に予測率が低下するが，その後，予測率が急速に回復し，増殖機能しか持たない ReproNet より高い予測率を出すことがしばしば観察される．

図 6.13には，ユニットを 2 個ずつ減少させる機能を持つ ReproNet の予測率とユニット数の変化を示した．また，図 6.14には，ホモロジーが高い（50% から 90%）データと低い（10% から 50%）データの予測率の平均をイタレーション 1000 回以降について示した．

増殖のみの ReproNet はイタレーションが 1500 回までは，予測率の上昇が見られるが，それ以降は極端に予測率が低下し，過学習を起こしている．特にイタレーション 2000 回以降はユニットが増殖してはいるが，予測率は低い状態に停滞している．

一方，2 個のユニットを削除する機能を持つ ReproNet は，イタレーションが 2000 回後も引続き，予測率が上昇している．ユニットの減少がイタレーション 1600 と 2000 付近で発生しており，これに伴い予測率が一時的に低下する．しかし，その後のユニットの増殖によって，予測率が回復し，過学習を回避している様子が観察される．

このことは，ネットワークがローカルミニマムに陥っている際に，ユニットが削除され，その後の新しいユニットが増殖されることにより，通常の増殖のみの ReproNet よりも，ローカルミニマムから脱出しやすくなっていることが考えられる．これは，より効率の良いユニットの削減方法を考案することによって，効果的な過学習の回避が可能であることを示唆しており，予測性能の向上が期待される．

6.5 予測結果のまとめ

本章の計算機実験による予測結果を表 6.3にまとめた．学習データに対するホモロジーの異なるテストデータごとに各予測手法が出した最も高い予測正答率の一覧を示した．表の第一カラムに関しては各予測手法を示しており，Homology Search はホモロジー検索による予測，NeuralNet 70 は隠れ層に 70 個のユニットをもつ Neural Network による予測，ReproNet reproduction only はユニットの増殖のみの機能を持つ ReproNet による予測，ReproNet 200 enzymes は 200 個の学習用酵素によって学習した ReproNet による予測，ReproNet 5 units reduction は一度に 5 個のユニットを削減する機能を持つ ReproNet による予測，ReproNet 2 units reduction は一度に 2 個のユニットを削減する機能を持つ ReproNet による予測である．本手法によって，学習データに対してホモロジーが低い酵素である場合，ホモロジー検索による予測よりも高い正答率が得られている．

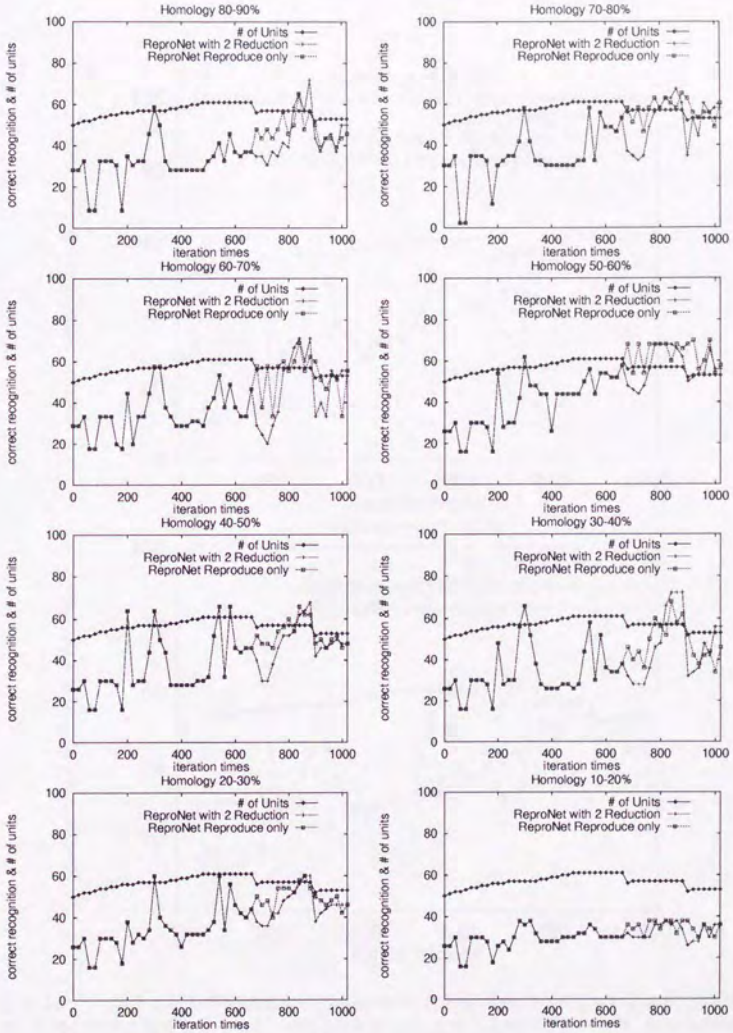


図 6.11　ユニット数を一度に 5 個づつ削減する機能を持つ ReproNet による予測とユニット数の変化および削減機能を持たない ReproNet の予測率．学習データに対するホモロジーが 10% から 90% までのテスト用酵素の予測率とユニット数の推移を示した．

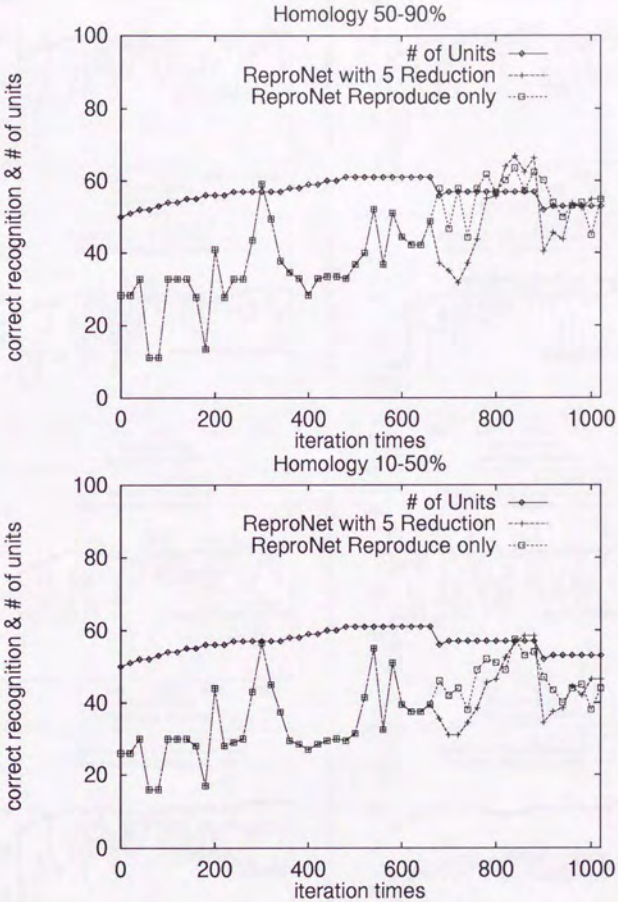


図 6.12　ユニットの削減機能を持つ ReproNet と増殖機能のみの ReproNet による予測．学習データに対するホモロジーが 50% から 90% まで（上図）および 10% から 50% まで（下図）のテスト用酵素の予測率の平均とユニット数の推移を示した．

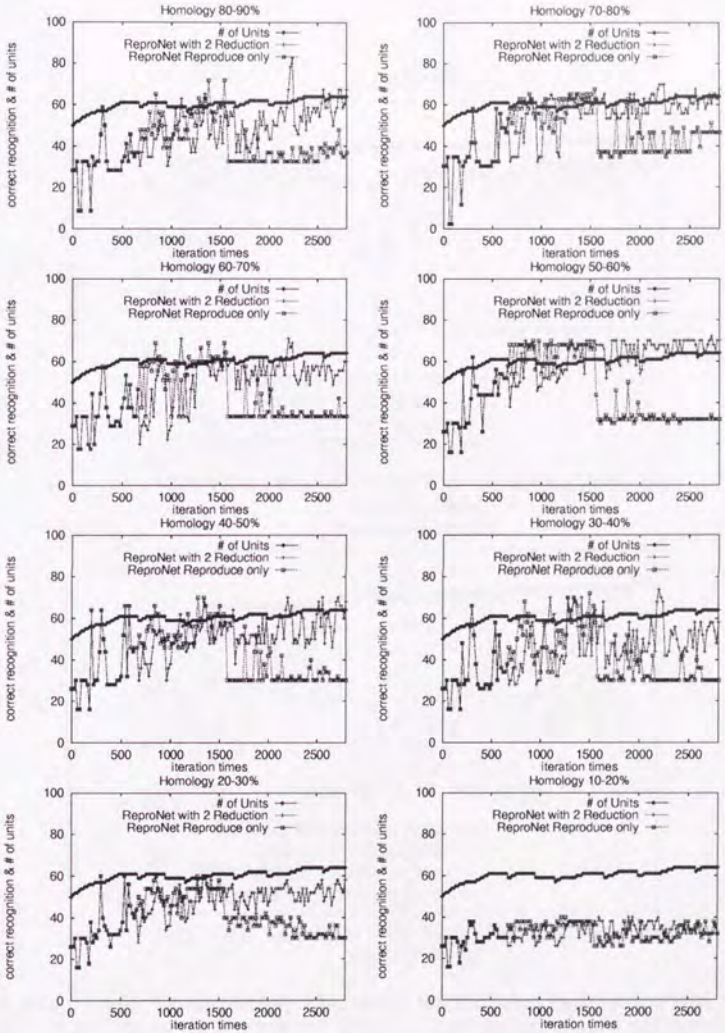


図 6.13　ユニット数を一度に 2 個ずつ削減する機能を持つ ReproNet による予測とユニット数の変化および削減機能を持たない ReproNet の予測率．学習データに対するホモロジーが 10% から 90% までのテスト用酵素の予測率とユニット数の推移を示した．

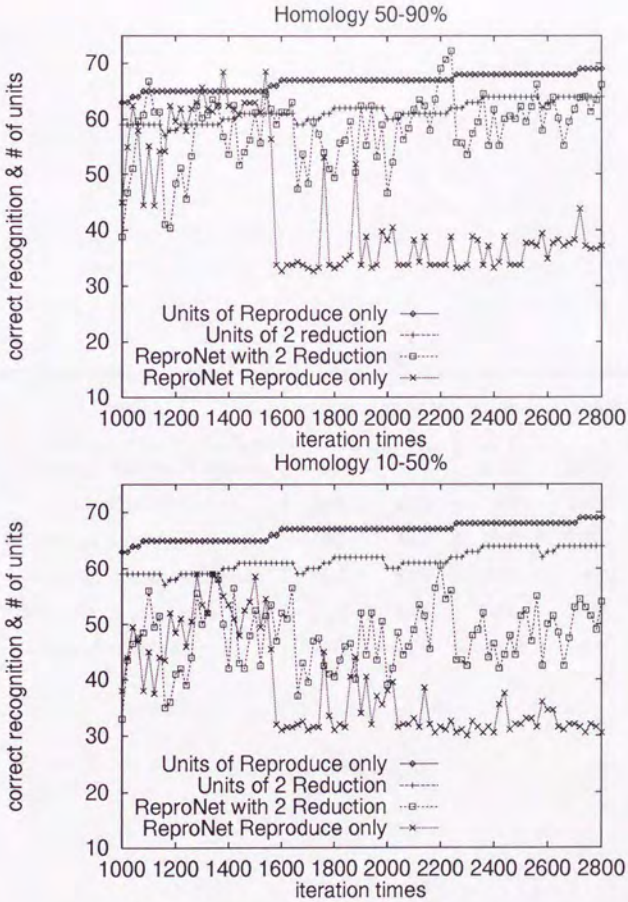


図 6.14　ユニットの削減機能を持つ ReproNet と増殖機能のみの ReproNet よる予測．学習データに対するホモロジーが 50% から 90% まで（上図）および 10% から 50% まで（下図）のテスト用酵素の予測率の平均とユニット数の推移を示した．

表 6.3 予測結果の一覧

Methods \ Homologies	10-20%	20-30%	30-40%	40-50%
Homology Search Prediction	*33%*	*55%*	**94%**	**96%**
NeuralNet 70 units	*40%*	*60%*	*72%*	*66%*
ReproNet reproduction only	*40%*	*60%*	*72%*	**70%**
ReproNet 5 units reduction	*38%*	*60%*	*72%*	*68%*
ReproNet 2 units reduction	*40%*	*60%*	**74%**	**70%**
ReproNet 200 enzymes	**48%**	**62%**	*68%*	*68%*

第 7 章

アミノ酸 1 次配列の解析

ネットワークは，11 残基からなる局所的な 1 次配列ごとに酵素の機能を予測するので（図 5.1参照），正しく機能を予測した 1 次配列部位は，その酵素の特徴的な残基列であることが考えられる．特徴的な部位の同定には，特異的化学修飾法（アフィニティーラベル），X 線結晶解析法，部位特異的突然変異導入法が用いられるが，精度やコストなどの点で限界がある．学習済の ReproNet によって，1 次配列から酵素の特異的な部位に関する知見が得られるなら大きな助けになる．本章では，酸化還元酵素 (EC1) として NAD 依存性脱水素酵素，転移酵素 (EC2) としてヌクレオチドポリメラーゼ，加水分解酵素 (EC3) としてセリンプロテアーゼのファミリーの実例 [54, 55] を用いて，6章で予測に使用した ReproNet の出力値に基づいたアミノ酸配列の解析を行った．ReproNet の配列に対する出力値を，実際の活性部位残基，基質結合部位，補酵素の結合部位，阻害物の結合部位などと比較分析し，機能部位予測システムの開発の基礎となる知見を得た．

7.1 酵素の特異性とアミノ酸 1 次配列

酵素は，触媒として働く作用を選択する可能性が非常に強い．この酵素の特異性は基質と酵素の構造上の相互関係（鍵と鍵穴モデル）によって発揮される．次の知見は，アミノ酸 1 次配列から酵素機能の特徴的な部位を抽出できる可能性を示唆している．

1. 酵素の活性部位は表面にある明瞭なくぼみである場合が多く，反応に直接関与する官能基が一定の配置で並んでおり，その周辺には基質の様々な部分と多様に相互作用するアミノ酸残基列が並んでいる．

2. 核酸の塩基やアミノ酸の側鎖のような基質の突出した部分に対しては，酵素表面上に対応するポケット上のくぼみが部位として存在する．

3. 活性部位は補酵素や補欠分子族あるいは金属などを含む場合もあるが，一般的には少数個のアミノ酸残基（活性中心残基）から構成され，これらが相互に特異的空間配置をとって存在している．

4. 酵素表面には，補酵素の結合部位やアロステリックエフェクターの結合部位が存在し，酵素反応を特徴付けている．

5. ある特定の阻害物は特定の条件で特定の酵素を阻害する．こうした阻害の特異性は酵素の立体構造の中に阻害物質と特異的に結合する部位が存在するためである．

ネットワークは局所的なアミノ酸残基列ごとに酵素の機能を予測するので，ネットワークが高い値を出力した残基列は，その酵素の特徴的な配列であることが考えられる．

7.2 酵素データとの照合

酸化還元酵素 (EC1)，転移酵素 (EC2)，および加水分解酵素 (EC3) から代表的な酵素として，各々 NAD 依存性脱水素酵素，ヌクレオチドポリメラーゼ，およびセリンプロテアーゼを選び，6章で検討した ReproNet による酵素機能予測を適用し，出力結果と実際のアミノ酸残基の特徴的部位とを比較分析した．

なお，使用した NAD 依存性脱水素酵素，ヌクレオチドポリメラーゼ，およびセリンプロテアーゼの付録 B.1に示した学習データに対するホモロジーは最大で，各々 21.3%，18.2%，および 40.2% である．

7.2.1 NAD 依存性脱水素酵素 (EC1)

ヌクレオチドは細胞内代謝において，エネルギー運搬の中心的役割を担っている．ヌクレオチドのエネルギー転移にかかわるのが酸化還元過程における酸化還元酵素である．このうち，代謝におけるヌクレオチドの基本的な役割はヌクレオチド結合酵素が果たしている．

NAD 依存性脱水素酵素はヌクレオチド結合酵素の中で最も大きなファミリーであるが，本節の計算機実験ではウマの肝臓から取られた NAD 依存性脱水素酵素であるアルコール脱水素酵素 (LADH) を用いた．LADH のサブユニットのドメイン (PDB コード 1ADG) の 3 次構造を図 7.1に示す．

LADH は二量体であり，ポリペプチド鎖は 374，375 残基からなり，2 つのドメインから構成されている．一方のドメインは補酵素である NAD に結合し，他方のド

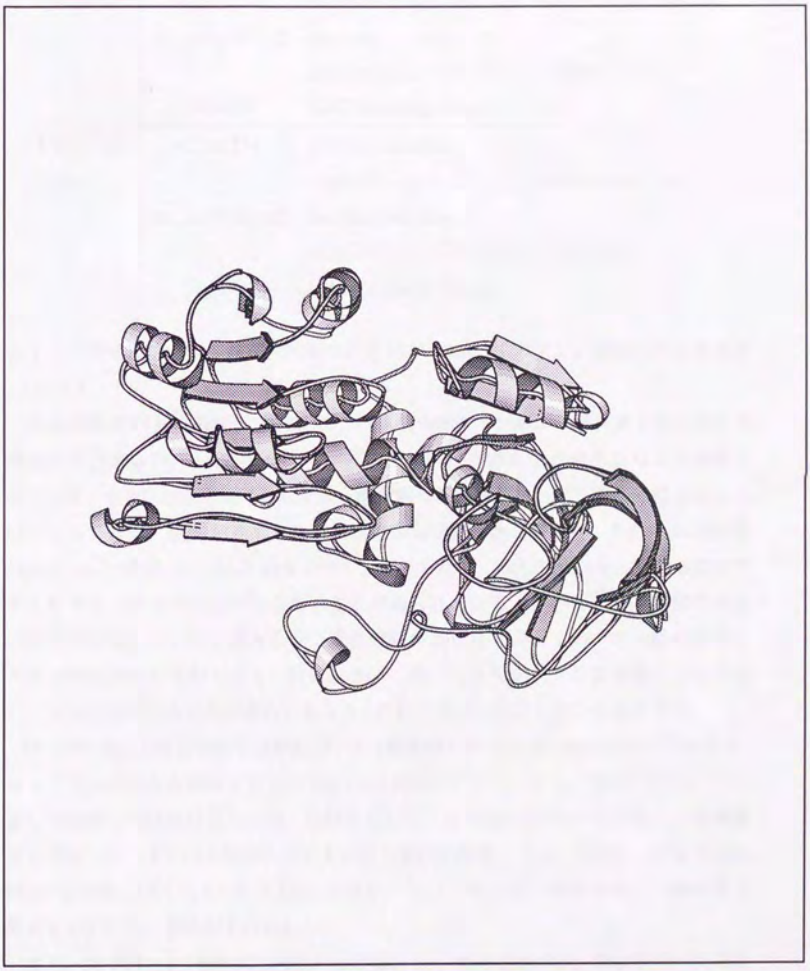


図 7.1 LADH のの NAD 結合ドメインと基質結合ドメインの構造．PDB コード 1ADG.

表 7.1 LADH の特徴的部位．（PIR より）

PIR code	residue number	feature
DEHOAL	47,68,175	binding site zinc
(horse E)		catalytic (Cys, His, Cys) status experimental
	98,101,104,112	binding site zinc
		noncatalytic (Cys) status experimental
	176-318	NAD-binding domain
DEHOAS	47,68,174	binding site zinc
(horse S)		catalytic (Cys, His, Cys) status predicted
	98,101,104,112	binding site zinc
		noncatalytic (Cys) status predicted
	176-318	NAD-binding domain

メインは基質に結合して触媒作用をつかさどる．LADH のアミノ酸配列の特徴を表 7.1に示す．

脱水素酵素では，機能的に類似した NAD 結合ドメインがポリペプチド鎖の異なる領域に存在するが，LADH では C 末端に近い領域である 176 残基から 318 残基に相当する．NAD 結合ドメインはアミノ酸配列に相同性がないが，3 次元構造はよく似ている．一方，触媒ドメインの構造は酵素間でまったく異なり，それぞれが独特のトポロジーを持つ．NAD 結合ドメインだけでなく，NAD が結合している部位でさえもアミノ酸配列に保存性がないことが知られている．しかし，この部位の残基の多くが変化していても鍵となる不変の残基があるので，ポリペプチド鎖の領域をアミノ酸配列から予測できる．特に，$\beta 1 - \alpha A - \beta 2$ モチーフはよく保存されており，3 次元構造が未知の酵素内にある NAD 結合領域の同定に用いられてきた．

図 7.2に NAD 依存性脱水素酵素のアミノ酸配列に対する ReproNet の出力を示す．ネットワークの出力層のうち EC1 の出力を実線で示した．また，実際に正しく EC1 として認識した残基に対しては，破線を記した．全残基のうち 34% が正しく酵素機能を予測した．また出力層のユニットの出力値の平均は，EC1: 0.2929, EC2: 0.2804, EC3: 0.2584, EC4 〜 EC6: 0.1801 となり，EC1 が最も高い出力を示し，酸化還元酵素として正しく機能が予測された．

また，図 7.2には，活性部位残基 Cys, His, Cys，亜鉛結合部位，および NAD 結合ドメインの位置を矢印で記した．亜鉛結合部位に関しては，ネットワークの出力値は高い値を示していないが，3 つの活性部位の周辺に関しては，高い出力を出し，

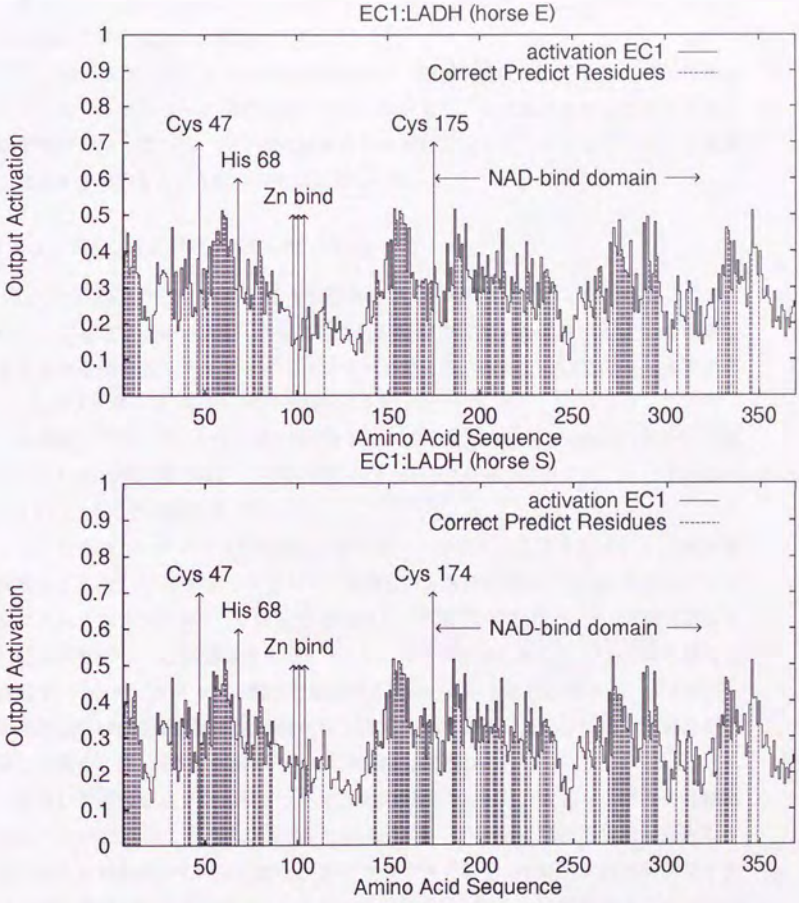


図 7.2　NAD 依存性脱水素酵素 (LADH) のアミノ酸配列に対する ReproNet の出力．上図が LADH（ウマ E）で，下図が LADH（ウマ S）

酵素機能の予測に寄与している．また，150 残基付近から 300 残基までの部位は比較的高い値を出力している．これは NAD 結合ドメインと整合しており，NAD 依存性脱水素酵素の特徴を抽出している．

図 7.3に LADH のサブユニットのドメイン構造について，正しく予測した部位を Spacefill で表示した 3 次構造を示す．

図 7.4に LADH のグラフィクス表示を示す．NAD 結合ドメインおよび活性部位に対して，ネットワークが高い出力を出しているが，それ以外の部分に対しても，過予測が発生している．過予測は酵素全体の機能予測としては望ましいが，特徴的な部位を予測するという観点からは望ましくない．

7.2.2 ヌクレオチドポリメラーゼ (EC2)

DNA 合成は，生細胞で起こる最も基本的で複雑な酵素反応の 1 つである．この反応の核心をなすのは，新生 DNA 鎖に鋳型が指定するヌクレオチドを正しく結合する反応を触媒するヌクレオチドポリメラーゼである．本節では，このうち大腸菌の DNA ポリメラーゼ I を用いて計算機実験を行った．

大腸菌の DNA ポリメラーゼ I は 928 個のアミノ酸からなる 1 本のポリペプチド鎖でできたタンパク質である．DNA ポリメラーゼ I クレノウフラグメント (PDB コード 1KFD) の 3 次構造を図 7.5に示す．

この酵素は 3 つのドメインを持ち，各ドメインが別々の機能を担当する．N 末端側ドメインは，5'-3' エキソヌクレアーゼ活性による校正機能をもつ．中央のドメイン（クレノウフラグメント）は α/β 型であり，3'-5' エキソヌクレアーゼ活性による校正機能をもつ．C 末端側ドメイン（クレノウフラグメント）は $\alpha+\beta$ 型の独特な構造で，DNA プライマーや鋳型と結合する深いくぼみを形成している．ポリメラーゼ活性部位は校正機能の活性部位から 30Å も離れており，この 2 つの活性部位が強調して働くには，DNA が酵素に対して移動する必要がある．

標識した基質を用いたポリメラーゼ反応の実験によると，大きなくぼみの内部にあるヘリックスのアミノ酸残基 758 と 766 がポリメラーゼの活性部位の一部となっていることがわかっている．また，3'-5' エキソヌクレアーゼ活性を持ちヌクレオチドと結合するクレノウフラグメントの小さいドメインはアミノ酸残基 330 ～ 420 である．アミノ酸配列の特徴を表 7.2に示す．

図 7.6にヌクレオチドポリメラーゼのアミノ酸配列に対する ReproNet の出力を示す．ネットワークの出力層のうち EC2 の出力を実線で示した．また，実際に正しく EC2 として認識した残基に対しては，破線を記した．全残基のうち 60% が正しく酵

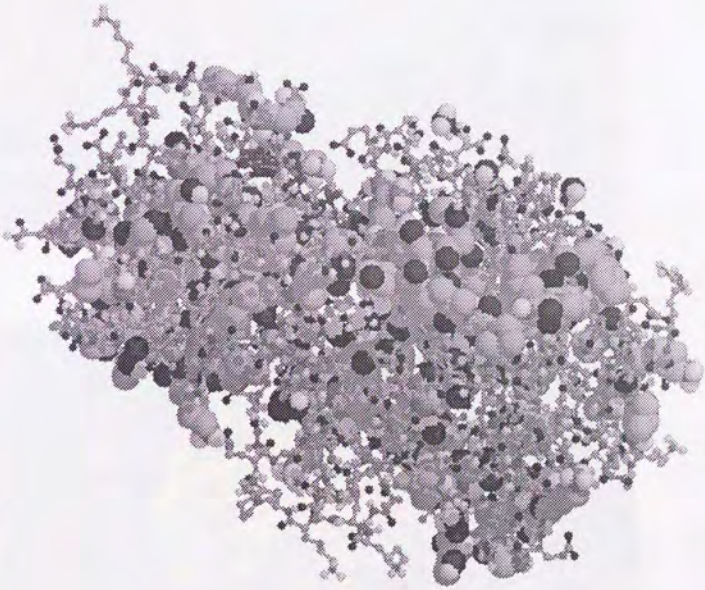


図 7.3　LADH について，正しく機能を予測した部位を Spacefill で表示し，その他を Ball&Stick で表示した．

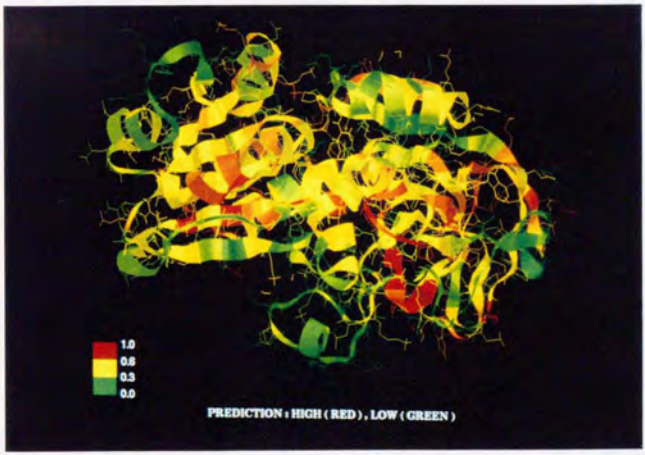


図 7.4　LADH のグラフィクス表示．上図は，NAD 結合ドメインと活性部位を赤，それ以外をシアンで表示．下図は，ネットワークの高い出力を赤，低い出力を緑で表示．

図 7.5 DNA ポリメラーゼ I クレノウフラグメントの 3 次構造．PDB コード 1KFD

表 7.2 DNA ポリメラーゼ I の特徴的部位．（PIR より）

PIR code	residue number	feature
DJECI	1-323	N-terminal fragment
(*E.coli*)		5'-3' exonuclease activities
	324-517	3'-5' exonuclease activities
		(Klenow fragment)
	518-928	C-terminal fragment
		carry the polymerase (Klenow fragment)
	758, 766	polymerase active site

素機能を予測した．また出力層のユニットの出力値の平均は，EC1: 0.2800, EC2: 0.3474, EC3: 02199, EC4 ～ EC6: 0.1643 となり，EC2 が最も高い出力を示し，転移酵素として正しく機能が予測された．

3 つのドメインのうちクレノウフラグメント C 末端側ドメインが最も高い出力を示したが，これは，このドメインが大きいくぼみを持ち，ポリメラーゼ活性と関連していることと合致している．ポリメラーゼ活性の一部となっていることが知られている 758 残基と 766 残基に関しては，その周辺の残基は高い出力を示したが，活性部位そのものは，低い値に留まっている．このことは，活性部位の周辺のアミノ酸残基列が活性に何らかの影響を与えているのではないかということを示唆するものである．3'-5' エキソヌクレアーゼ活性に関係する残基 330 ～ 420 について，ネットワークの出力は，340 残基の周辺と 390 残基から 420 残基までは高い値を示したが，他の部位は低い値に留まった．このことは，既知の活性部位のうち，ネットワークが高い出力を示した部位が特に，ヌクレアーゼ活性に関係している可能性があることを示唆している．

図 7.7に DNA ポリメラーゼ I について，正しく予測した部位を Spacefill で表示した 3 次構造を示す．

図 7.8に DNA ポリメラーゼ I のグラフィクス表示を示す．活性部位周辺とポリメラーゼ活性に関係のある大きなくぼみの上部に対してネットワークが高い出力を出しているが，それ以外の部分に対しても，過予測が発生している．

7.2.3 セリンプロテアーゼ (EC3)

セリンプロテアーゼはタンパク質のペプチド結合を加水分解する一群の酵素である．セリンプロテアーゼはすべて触媒反応に不可欠な 4 つの特徴を備えている．そ

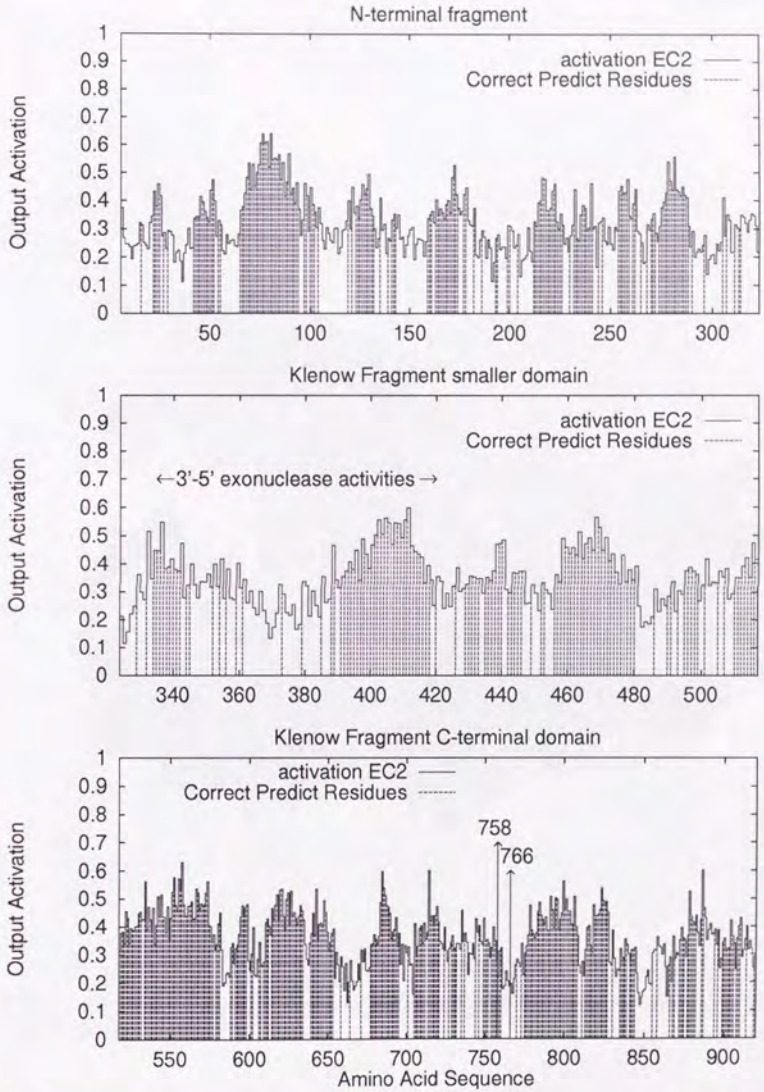
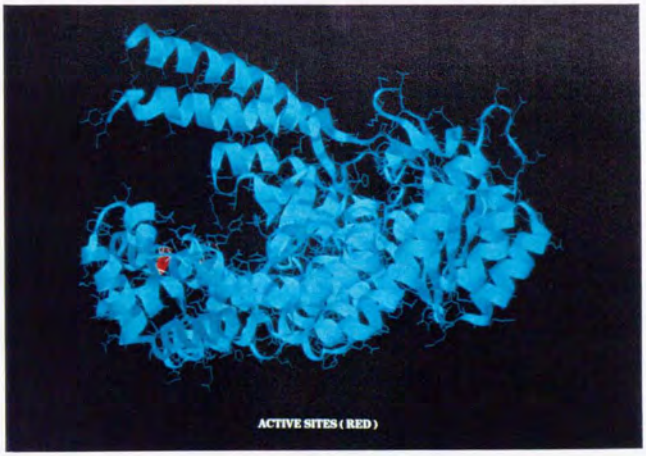


図 7.6　DNA ポリメラーゼ I のアミノ酸配列に対する ReproNet の出力．上図は N 末端フラグメント．中図はクレノウフラグメントの小さい方のドメイン．下図はクレノウフラグメント C 末端側ドメイン．

図 7.7　DNA ポリメラーゼについて，正しく機能を予測した部位を Spacefill で表示し，その他を Ball&Stick で表示した．



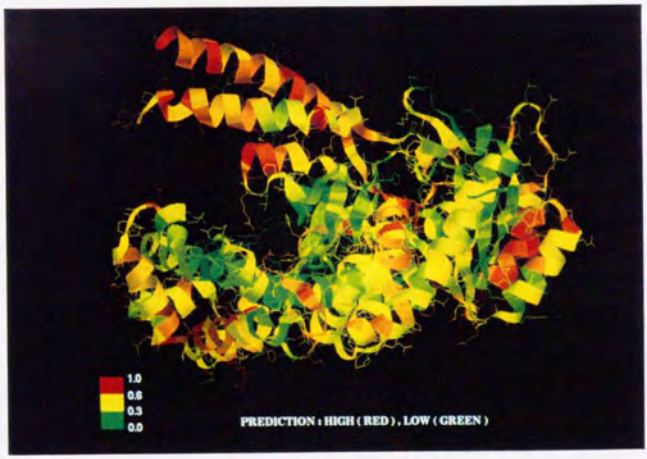


図 7.8　DNA ポリメラーゼ I のグラフィクス表示．上図は，活性部位を赤，それ以外をシアンで表示．下図は，ネットワークの高い出力を赤，低い出力を緑で表示．

表 7.3 キモトリプシンの特徴的部位．（PIR より）

PIR code	residue number	feature
KYBOA	1-13,16-146,	
(A precursor	149-245	product alpha-chymotrypsin label MPT
- bovine)	16-238	domain trypsin homology label TRP
	1-122,42-58,	
	136-201,168-182,	
	191-220	disulfide bonds status experimental
	57,102,195	active site His, Asp, Ser
KYBOA	16-238	domain trypsin homology label TRP
(B precursor	1-122,42-58,	
- bovine)	136-201,168-182,	
	191-220	disulfide bonds status experimental
	57,102,195	active site His, Asp, Ser

の 4 つとは触媒トライアード，オキシアニオンホール，基質特異性ポケット，基質であるポリペプチドに対する非特異的結合部位である．本節では，セリンプロテアーゼのうち，ほ乳類の 2 種のキモトリプシンを使用して計算機実験を行った．この 2 種のキモトリプシンの相互のホモロジーは 79% であり，類似したタンパク質である．キモトリプシン (PDB コード 1GCT) の 3 次構造を図 7.9に示す．また，キモトリプシンのアミノ酸配列の特徴を表 7.3に示す．

キモトリプシンでは，触媒トライアードは互いに近くにある Asp，His，Ser の側鎖からなる．Ser 残基は基質と共有結合を作って反応が特定の経路を通るようにする．His は 2 つの役割をもつ．第一は Ser からの水素原子を受け取って，共有結合を形成しやすくすること，第二は負の電荷をもつ遷移状態を安定化することである．この 2 つの残基のどちらかに変異が起きると，触媒反応の速度は極端に低下してしまう．Asp は His のもつ正電荷を安定化して反応速度を増大させる．活性部位は 2 個のドメインの避け目にあり，触媒トライアードの 2 つの残基，His57 と Asp102 は C 末端側のドメインに属し，反応性セリン Ser195 は N 末端側のドメインに属する．同様に，阻害剤もこのドメイン間の避け目に結合する．

図 7.10にキモトリプシンのアミノ酸配列に対する ReproNet の出力を示す．ネットワークの出力層のうち EC3 の出力を実線で示した．また，実際に正しく EC3 として認識した残基に対しては，破線を記した．全残基のうち, 各々 54%，41% が正

図 7.9 キモトリプシンの 3 次構造 (BOVINE PANCREAS). PDB コード 1GCT.

しく酵素機能を予測した．また出力層のユニットの出力値の平均は，各々 EC1: 0.3046, EC2: 0.1965, EC3: 0.3755, EC4 ～ EC6: 0.1562 および EC1: 0.3074, EC2: 0.2148, EC3: 0.3322, EC4 ～ EC6:0.1685 となり，いずれも EC3 が最も高い出力を示し，加水分解酵素として正しく機能が予測された．

また，図 7.10には，活性部位残基 His, Asp, Ser を矢印で記した．Asp57 については，上下の両方の図ともネットワークの出力値と一致しなかった．Asp102 に関しては，上図のみ周囲の残基列が高い出力を示した．Ser195 については，上下図とも周辺の残基が高い出力を示した．

具体例として，ドメイン間に結合する小型のペプチド性阻害剤である Ac-Pro-Ala-Pro-Tyr-COOH の例を考察する．この阻害剤は，触媒トラーアード，基質性ポケット，オキシアニオンホール，非特異性基質結合領域と関連をもっている．Ser195 と共有結合を作らないが，カルボキシル基の酸素原子はオキシアニオンホールの中で，193 番と 194 番の残基の主鎖の NH 基と水素結合している．またチロシンの側鎖は 189 番残基に関係する特異性ポケットの内部に納まっている．このポケットでの特異性を決めているのは，主に 216, 226, 189 番の 3 つの残基である．阻害剤の主鎖は，酵素の残基 215 の NH 基と残基 216 の CO 基に水素結合して，短い逆平行 β シートを形成している．ネットワークの出力と比較すると，Ser195 の直前の 192, 193, 194 番残基はいずれも高い出力値を持ち，オキシアニオンホールと一致している．またポケットでの結合に関与している 216, 226, 189 番残基に関しても高い値を出力している．

図 7.11にキモトリプシンについて，正しく予測した部位を Spacefill で表示した 3 次構造を示す．

図 7.12にキモトリプシンのグラフィクス表示を示す．活性部位周辺，触媒トライアード，および阻害剤結合部位に対してネットワークが高い出力を出しているが，それ以外の部分に対しても，過予測が発生している．

7.3 機能部位抽出の可能性と有用性

以上の解析から酵素の機能部位の抽出の可能性を検討する．7.2節の計算機実験で得たアミノ酸配列に対するネットワークの出力値を分析すると，活性部位およびそれに関連する結合部位の残基に共通の傾向が見られる．図 7.2, 7.6, 7.10に示されている機能部位とネットワークの出力値にも共通の類似点が見られる．図 7.13に，その関係を示した．これは，機能部位を抽出する際の指針を提案するものであり，定量的に波形を求めることを目的としていないが，いまだ仮説の段階である．

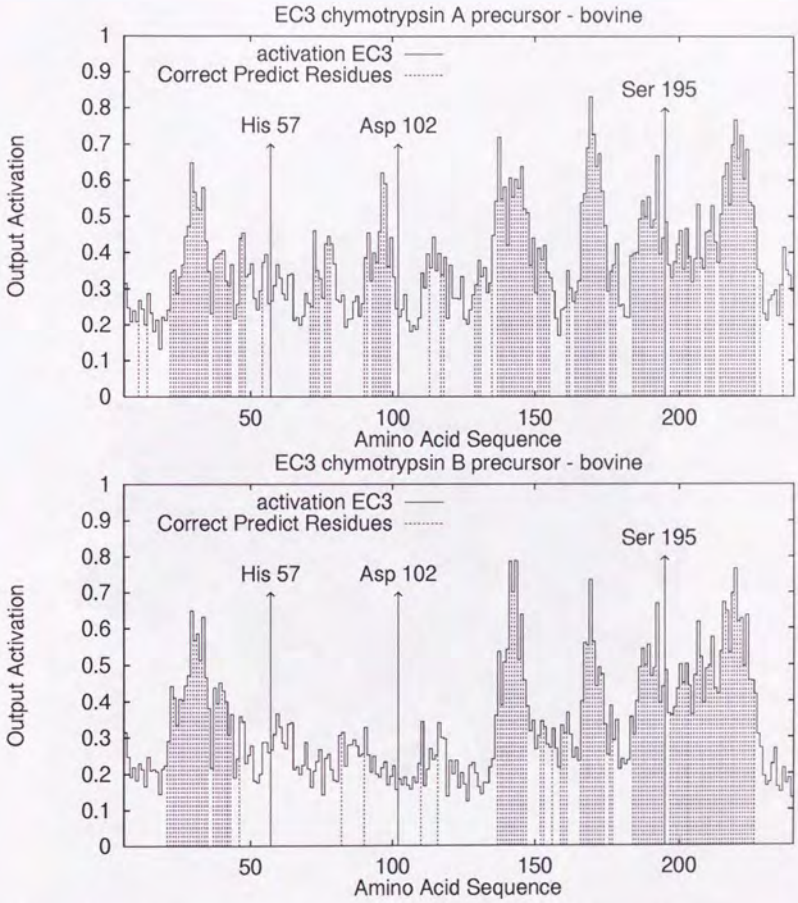


図 7.10　キモトリプシンのアミノ酸配列に対する ReproNet の出力．上図は，A precusor で，下図は，B precusor．

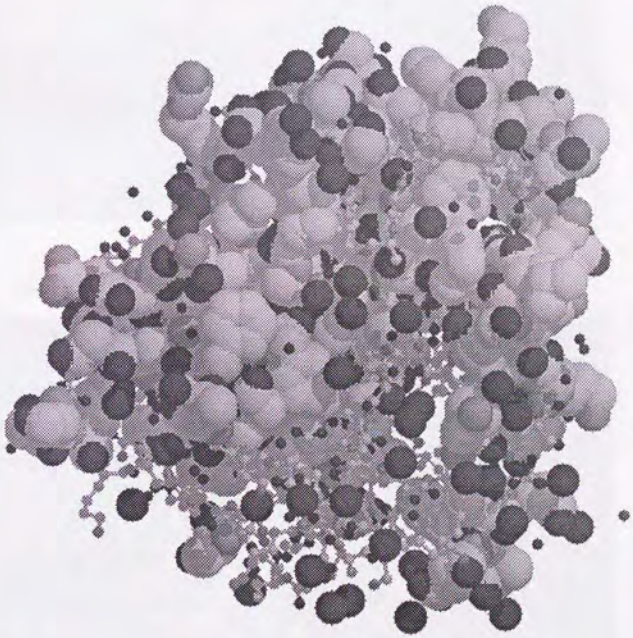


図 7.11　キモトリプシンについて，正しく機能を予測した部位を Spacefill で表示し，その他を Ball&Stick で表示した．

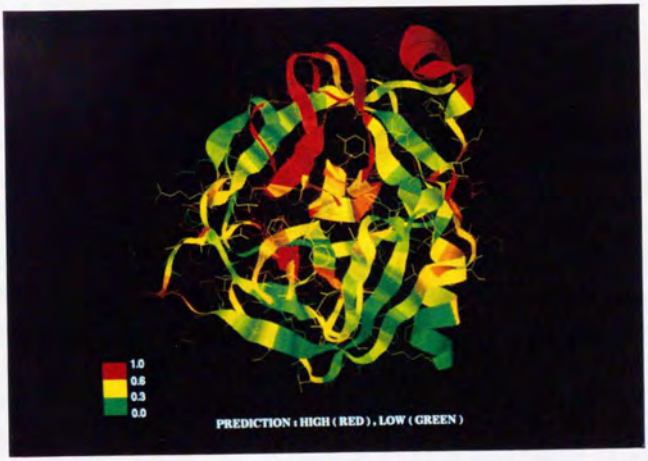


図 7.12　キモトリプシンのグラフィクス表示．上図は，阻害剤の結合部位と活性部位を赤，それ以外をシアンで表示．下図は，ネットワークの高い出力を赤，低い出力を緑で表示．

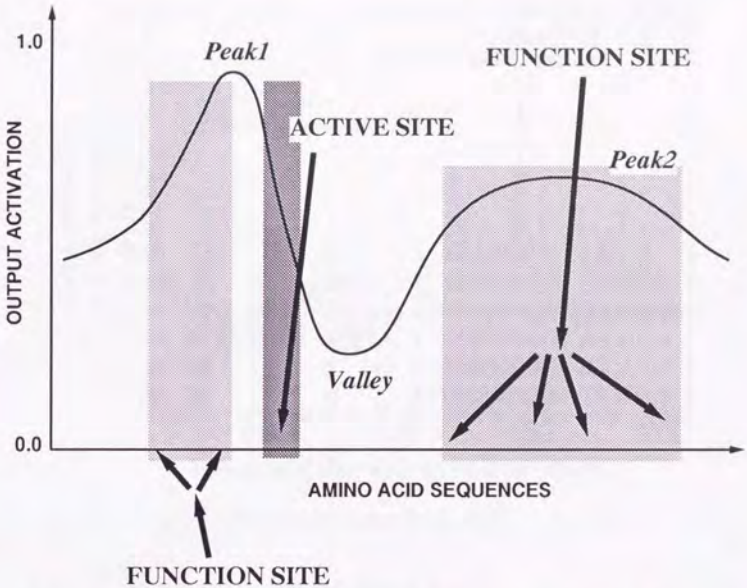


図 7.13 ReproNet のアミノ酸配列の出力から活性部位および他の機能部位を予測する.

活性部位の存在するアミノ酸残基のネットワーク出力は必ずしも高い値ではない. むしろ出力値の曲線の谷に相当する位置に活性部位が存在している. 一方, 活性部位の直前の残基の出力は鋭いピークを示す傾向がある. 活性部位の直後では, 出力値は上昇し, 直前のピークと比較すると緩やかな丘を形成する. また酵素の特異的な機能に関連する他の機能部位や結合部位は, 活性部位の前後のピークに広い領域で存在する傾向が見られる (特に図 7.2, 7.10). したがって, 同様な出力曲線を示す傾向のあるアミノ酸残基列は, 何らかの酵素機能に関係する可能性があるといえる. 例えば, 図 7.2の 150 番, 170 番残基前後, 図 7.10の 300 番残基前後, 図 7.6の 810 番, 580 番残基前後がこれに相当する.

7.3.1 chymotrypsin II - oriental hornet(EC3) の解析例

図 7.13のモデルを使用して, 未知の酵素について活性部位を予測する例を示す. ここでは, キモトリプシンである chymotrypsin II - oriental hornet を用いた. 7.2.3節

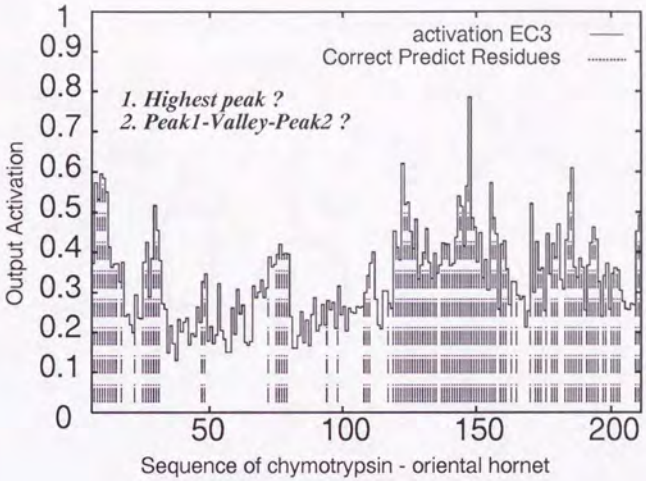


図 7.14 ReproNet の出力の分析

の chymotrypsin - bovin とのホモロジーは 30% である．

ReproNet の出力を図 7.14に示す．分析の指針は次の通りである．

1. 最も高いピークを探す．

2. 図 7.13の Peak1-Valley-Peak2 モデルに適応する部位を探す．

ReproNet の出力から，以上の分析に基づいて，推測された機能部位の候補を図 7.15に示す．図には探索された順番を記した．

図 7.16に，実際の活性部位を下向き矢印で示し，ReproNet の出力と実際の活性部位を照合した．この例では，75% の割合で予測できている．

今回の仮説は非常に限られた少数の酵素のみの解析であるので，機能部位予測の信憑性は決して高いものとは言えない．

たとえば，Highest Peak は順次，一意的に求まる．しかし，Valley Peak2 の決定が決して一意的ではない．円の幅（半径）をどうするか，左右の対象をどうするか，などが決定の際に問題となる．したがって，波形のスムージングの手法と 2Peak の決定方法の開発が今後の課題である．

さらに多くの酵素を用いたアミノ酸配列の解析によってネットワーク出力と酵素機能部位の相関関係に関する見地を得ることが必要である．また，ネットワーク出

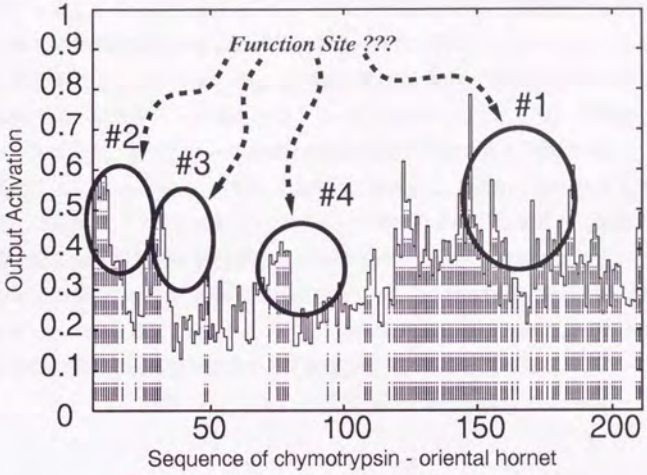


図 7.15 ReproNet の出力から機能部位の候補を推測

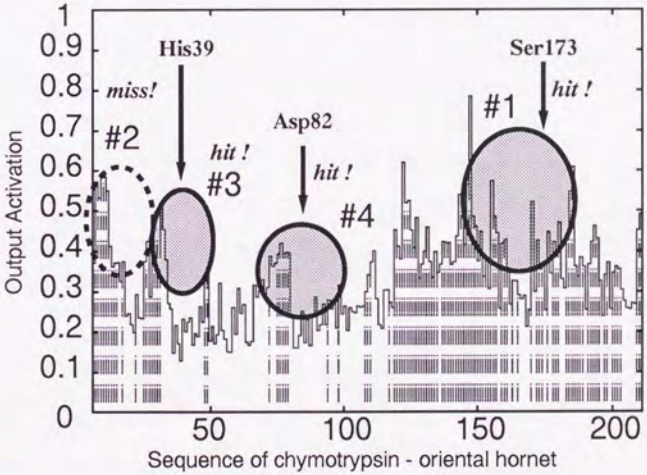


図 7.16 ReproNet の出力と実際の活性部位の照合

力がアミノ酸配列の特徴に反応するためには，より広範な酵素を学習用データとして包含させることも必要である．一方，特定の機能に限定して部位抽出を行う場合は，目的とする機能を持つ酵素の残基のみを局所的に集中してネットワークに学習させる必要があると思われる．例えば，酸化還元用，加水分解用のネットワークや脱水素酵素用，プロテアーゼ用のネットワークなど特定のファミリーに限定して学習することにより，そのグループの酵素の機能部位を予測できる可能性がある．

本手法によって ReproNet のアミノ酸配列に対する出力値から得られた見地は，特異的化学修飾法（アフィニティーラベル）や部位特異的突然変異導入法を行う際に意義のある有用な情報を提供すると思われる．ホモロジー検索では，高い値が得られなかった場合，ほとんど無作為に部位を調査しなければならないという現実を考慮すると，本手法は，ReproNet のアミノ酸配列に対する出力値によって，ある程度，意義のある有用な情報を提供する可能性が十分にあると思われる．

第 8 章

結言

本研究を要約すると，次の通りである．

1. 脳の情報処理機構の分析に基づいた自己増殖オートマトンネットワークによるマシンアーキテクチャ ReproNet を提案した．このマシンはユニット，ネットワーク，変移則からなり，単純な要素を多数結合して，並列の相互作用によって情報処理を実現する．また，学習と自己増殖によって環境に適用した自己を調整する能力を持ち，柔軟な情報処理に優れている．

2. 変化する環境における ReproNet の学習能力を暗号解読問題と文字認識問題によって検証した．このモデルによって，ネットワークを柔軟に変化させる Neural Network を構築し，動的に変化する学習環境に対しても常に効率の良い学習を可能にした．また事前にネットワークの規模を決定する際の困難を削減できる．このことを暗号解読と文字認識問題という例によって示した．特に，UNIX の crypt に代表される暗号文の解読といった定式化の困難な非線形の系に対しても，優れた学習能力をもつことが示された．

3. アミノ酸 1 次配列を入力して，酵素機能の分類を学習させた．増加するタンパク質データに対しても高い学習性能を維持することを示した．

4. 学習済の ReproNet を用いて酵素機能予測を行った．ホモロジー検索よりも有効な手法であることを示した．冗長なユニットを削除する機能などを取り入れることによって，極小値や過学習を回避し，予測性能を改善することができた．

5. ReproNet の 1 次配列に対する出力値を解析し，酵素の特徴的な部位との比較検討を行った．ReproNet のアミノ酸配列に対する出力に基づいて，機能部位を抽出する可能性を論じた．

また，脳の集団内における遺伝操作による個体の変化に対応する機能としては，ユニット，ネットワーク形態，変移則の情報をコード化し，複数のネットワークからなる集団を用いることによって，世代間のネットワーク変化も可能となる．今後，Genetic Algorithm 的な手法を用いた世代間の最適化を予定している．

未知データに対する予測システムには常にローカルミニマムや過学習の回避といった問題が付きまとう．これは，ネットワークの冗長な部分を縮小することによって，ある程度，解決されることが知られている．今回の酵素機能予測における計算機実験では，ユニットの削除の方法はユニット数がある期間増殖しない状態が継続するとユニットを幾つか削除するという単純な規則にすぎなかった．さらに柔軟にネットワーク形態を変化する機能も考慮することによって，予測性能の向上が期待されるため，今後の検討する必要がある．

さらに，ReproNet を並列マシンとしてハードウェア化することは，個々のプロセッサに変移則を持たせ，ネットワーク上に配置して情報交換する構造を実現することによって可能となるが，今後の課題として検討していきたい．

さらに，酵素の特徴的なアミノ酸 1 次配列を局所的に学習し，機能部位を抽出または予測するシステム，またタンパク質の 2, 3 次構造予測システムなどにも適用する予定である．

付録 A

酵素クラスの定義

EC-number	Definition of enzymes classes, subclasses and sub-subclasses
1. -. -.-	Oxidoreductases.
1. 1. -.-	acting on the CH-OH group of donors.
1. 1. 1.-	with NAD^+ or $NADP^+$ as acceptor.
1. 1. 2.-	with a cytochrome as acceptor.
1. 1. 3.-	with oxygen as acceptor.
1. 1. 4.-	with a disulfide as acceptor.
1. 1. 5.-	with a quinone or similar compound as acceptor.
1. 1.99.-	with other acceptors.
1. 2. -.-	acting on the aldehyde or oxo group of donors.
1. 2. 1.-	with NAD^+ or $NADP^+$ as acceptor.
1. 2. 2.-	with a cytochrome as acceptor.
1. 2. 3.-	with oxygen as acceptor.
1. 2. 4.-	with a disulfide as acceptor.
1. 2. 7.-	with an iron-sulfur protein as acceptor.
1. 2.99.-	with other acceptors.
1. 3. -.-	acting on the CH-CH group of donors.
1. 3. 1.-	with NAD^+ or $NADP^+$ as acceptor.
1. 3. 2.-	with a cytochrome as acceptor.
1. 3. 3.-	with oxygen as acceptor.
1. 3. 5.-	with a quinone or related compound as acceptor.
1. 3. 7.-	with an iron-sulfur protein as acceptor.
1. 3.99.-	with other acceptors.
1. 4. -.-	acting on the $CH\text{-}NH_2$ group of donors.
1. 4. 1.-	with NAD^+ or $NADP^+$ as acceptor.
1. 4. 2.-	with a cytochrome as acceptor.
1. 4. 3.-	with oxygen as acceptor.
1. 4. 4.-	with a disulfide as acceptor.
1. 4. 7.-	with an iron-sulfur protein as acceptor.
1. 4.99.-	with other acceptors.
1. 5. -.-	acting on the CH-NH group of donors.
1. 5. 1.-	with NAD^+ or $NADP^+$ as acceptor.
1. 5. 3.-	with oxygen as acceptor.
1. 5. 4.-	with a disulfide as acceptor.
1. 5. 5.-	with a quinone or similar compound as acceptor.
1. 5.99.-	with other acceptors.
1. 6. -.-	acting on NADH or NADPH.
1. 6. 1.-	with NAD^+ or $NADP^+$ as acceptor.
1. 6. 2.-	with a cytochrome as acceptor.
1. 6. 4.-	with a disulfide as acceptor.
1. 6. 5.-	with a quinone or similar compound as acceptor.
1. 6. 6.-	with a nitrogenous group as acceptor.
1. 6. 8.-	with a flavin as acceptor.
1. 6.99.-	with other acceptors.
1. 7. -.-	acting on other nitrogenous compounds as donors.
1. 7. 2.-	with a cytochrome as acceptor.
1. 7. 3.-	with oxygen as acceptor.
1. 7. 7.-	with an iron-sulfur protein as acceptor.
1. 7.99.-	with other acceptors.
1. 8. -.-	acting on a sulfur group of donors.
1. 8. 1.-	with NAD^+ or $NADP^+$ as acceptor.
1. 8. 2.-	with a cytochrome as acceptor.
1. 8. 3.-	with oxygen as acceptor.
1. 8. 4.-	with a disulfide as acceptor.
1. 8. 5.-	with a quinone or similar compound as acceptor.
1. 8. 7.-	with an iron-sulfur protein as acceptor.
1. 8.99.-	with other acceptors.
1. 9. -.-	acting on a heme group of donors.
1. 9. 3.-	with oxygen as acceptor.
1. 9. 6.-	with a nitrogenous group as acceptor.

1. 9.99.-	with other acceptors.
1.10. -.-	acting on diphenols and related substances as donors.
1.10. 1.-	with NAD^+ or $NADP^+$ as acceptor.
1.10. 2.-	with a cytochrome as acceptor.
1.10. 3.-	with oxygen as acceptor.
1.10.99.-	with other acceptors.
1.11. -.-	acting on a peroxide as accepor (peroxidases).
1.12. -.-	acting on hydrogen as donor.
1.12. 1.-	with NAD^+ or $NADP^+$ as acceptor.
1.12. 2.-	with a cytochrome as acceptor.
1.12.99.-	with other acceptors.
1.13. -.-	acting on single donors with incorporation of molecular oxygen.
1.13.11.-	with incorporation of two atoms of oxygen.
1.13.12.-	with incorporation of one atom of oxygen.
1.13.99.-	miscellanous (requires further characterizaton).
1.14. -.-	acting on paired donors with incorporation of molecular oxygen.
1.14.11.-	with 2-oxoglutarate as one donor, and incorporation of one atom each of oxygen into both donors.
1.14.12.-	with NADH or NADPH as one donor, and incorporation of two atoms of oxygen into one donor.
1.14.13.-	with NADH or NADPH as one donor, and incorporation of one atom of oxygen.
1.14.14.-	with reduced flavin or flavoprotein as one donor, and incorporation of one atom of oxygen.
1.14.15.-	with a reduced iron-sulfur protein as one donor, and incorporation of one atom of oxygen.
1.14.16.-	with reduced pteridine as one donor, and incorporation of one atom of oxygen.
1.14.17.-	with ascorbate as one donor, and incorporation of one atom of oxygen.
1.14.18.-	with another compound as one donor, and incorporation of one atom of oxygen.
1.14.99.-	miscellanous (requires further characterizaton).
1.15. -.-	acting on superoxide radicals as acceptor.
1.16. -.-	oxidizing metal ions.
1.16. 1.-	with NAD^+ or $NADP^+$ as acceptor.
1.16. 3.-	with oxygen as acceptor.
1.17. -.-	acting on $-CH_2$ groups.
1.17. 1.-	with NAD^+ or $NADP^+$ as acceptor.
1.17. 3.-	with oxygen as acceptor.
1.17. 4.-	with a disulfide as acceptor.
1.17.99.-	with other acceptors.
1.18. -.-	acting on reduced ferredoxin as donor.
1.18. 1.-	with NAD^+ or $NADP^+$ as acceptor.
1.18. 6.-	with dinitrogen as acceptor.
1.18.99.-	with H^+ as acceptor.
1.19. -.-	acting on reduced flavodoxin as donor.
1.19. 6.-	with dinitrogen as acceptor.
1.97. -.-	other oxidoreductases.
2. -. -.-	Transferases.
2. 1. -.-	transferring one-carbon groups.
2. 1. 1.-	methyltransferases.
2. 1. 2.-	hydroxymethyl-, formyl- and related transferases.
2. 1. 3.-	carboxyl- and carbamoyltransferases.
2. 1. 4.-	amidinotransferases.
2. 2. -.-	transferring aldehyde or ketone residues.
2. 3. -.-	acyltransferases.
2. 3. 1.-	acyltransferases.
2. 3. 2.-	aminoacyltransferases.
2. 4. -.-	glycosyltransferases.
2. 4. 1.-	hexosyltransferases.
2. 4. 2.-	pentosyltransferases.
2. 4.99.-	transferring other glycosyl groups.
2. 5. -.-	transferring alkyl or aryl groups, other than methyl groups.
2. 6. -.-	transferring nitrogenous groups.
2. 6. 1.-	transaminases (aminotransferases).
2. 6. 3.-	oximinotransferases.
2. 6.99.-	transferring other nitrogeneous groups.
2. 7. -.-	transferring phosphorous-containing groups.
2. 7. 1.-	phosphotransferases with an alcohol group as acceptor.
2. 7. 2.-	phosphotransferases with a carboxyl group as acceptor.
2. 7. 3.-	phosphotransferases with a nitrogeneous group as acceptor.
2. 7. 4.-	phosphotransferases with a phosphate group as acceptor.
2. 7. 6.-	diphosphotransferases.
2. 7. 7.-	nucleotidyltransferases.
2. 7. 8.-	transferases for other substituted phosphate groups.
2. 7. 9.-	phosphotransferases with paired acceptors.
2. 8. -.-	transferring sulfur-containing groups.
2. 8. 1.-	sulfurtransferases.
2. 8. 2.-	sulfotransferases.
2. 8. 3.-	coa-transferases.
2. 9. -.-	transferring selenium-containing groups.
3. -. -.-	Hydrolases.
3. 1. -.-	acting on ester bonds.
3. 1. 1.-	carboxylic ester hydrolases.
3. 1. 2.-	thiolester hydrolases.
3. 1. 3.-	phosphoric monoester hydrolases.
3. 1. 4.-	phosphoric diester hydrolases.
3. 1. 5.-	triphosphoric monoester hydrolases.
3. 1. 6.-	sulfuric ester hydrolases.
3. 1. 7.-	diphosphoric monoester hydrolases.
3. 1. 8.-	phosphoric triester hydrolases.

3. 1.11.- exodeoxyribonucleases producing 5'-phosphomonoesters.
3. 1.13.- exoribonucleases producing 5'-phosphomonoesters.
3. 1.14.- exoribonucleases producing other than 5'-phosphomonoesters.
3. 1.15.- exonucleases active with either ribo- or deoxyribonucleic acid and producing 5'-phosphomonoesters.
3. 1.16.- exonucleases active with either ribo- or deoxyribonucleic acid and producing other than 5'-phosphomonoesters.
3. 1.21.- endodeoxyribonucleases producing 5'-phosphomonoesters.
3. 1.22.- endodeoxyribonucleases producing other than 5'-phosphomonoesters.
3. 1.25.- site-specific endodeoxyribonucleases specific for altered bases.
3. 1.26.- endoribonucleases producing 5'-phosphomonoesters.
3. 1.27.- endoribonucleases producing other than 5'-phosphomonoesters.
3. 1.30.- endonucleases active with either ribo- or deoxyribonucleic acid and producing 5'-phosphomonoesters.
3. 1.31.- endonucleases active with either ribo- or deoxyribonucleic acid and producing other than 5'-phosphomonoesters.
3. 2. -.- glycosidases.
3. 2. 1.- hydrolysing O-glycosyl compounds.
3. 2. 2.- hydrolysing N-glycosyl compounds.
3. 2. 3.- hydrolysing S-glycosyl compounds.
3. 3. -.- acting on ether bonds.
3. 3. 1.- thioether hydrolases.
3. 3. 2.- ether hydrolases.
3. 4. -.- acting on peptide bonds (peptide hydrolases).
3. 4.11.- aminopeptidases.
3. 4.13.- dipeptidases.
3. 4.14.- dipeptidyl-peptidases and tripeptidyl-peptidases.
3. 4.15.- peptidyl-dipeptidases.
3. 4.16.- serine-type carboxypeptidases.
3. 4.17.- metallocarboxypeptidases.
3. 4.18.- cysteine-type carboxypeptidases.
3. 4.19.- omega peptidases.
3. 4.21.- serine endopeptidases.
3. 4.22.- cysteine endopeptidases.
3. 4.23.- aspartic endopeptidases.
3. 4.24.- metalloendopeptidases.
3. 4.99.- endopeptidases of unknown catalytic mechanism.
3. 5. -.- acting on carbon-nitrogen bonds, other than peptide bonds.
3. 5. 1.- in linear amides.
3. 5. 2.- in cyclic amides.
3. 5. 3.- in linear amidines.
3. 5. 4.- in cyclic amidines.
3. 5. 5.- in nitriles.
3. 5.99.- in other compounds.
3. 6. -.- acting on acid anhydrides.
3. 6. 1.- in phosphorous-containing anhydrides.
3. 6. 2.- in sulfonyl-containing anhydrides.
3. 7. -.- acting on carbon-carbon bonds.
3. 7. 1.- in ketonic substances.
3. 8. -.- acting on halide bonds.
3. 8. 1.- in C-halide compounds.
3. 9. -.- acting on phosphorus-nitrogen bonds.
3.10. -.- acting on sulfur-nitrogen bonds.
3.11. -.- acting on carbon-phosphorus bonds.
3.12. -.- acting on sulfur-sulfur bonds.

---

4. -. -.- Lyases.
4. 1. -.- carbon-carbon lyases.
4. 1. 1.- carboxy-lyases.
4. 1. 2.- aldehyde-lyases.
4. 1. 3.- oxo-acid-lyases.
4. 1.99.- other carbon-carbon lyases.
4. 2. -.- carbon-oxygen lyases.
4. 2. 1.- hydro-lyases.
4. 2. 2.- acting on polysaccharides.
4. 2.99.- other carbon-oxygen lyases.
4. 3. -.- carbon-nitrogen lyases.
4. 3. 1.- ammonia-lyases.
4. 3. 2.- amidine-lyases.
4. 3. 3.- amine-lyases.
4. 3.99.- other carbon-nitrogen-lyases.
4. 4. -.- carbon-sulfur lyases.
4. 5. -.- carbon-halide lyases.
4. 6. -.- phosphorus-oxygen lyases.
4.99. -.- other lyases.

---

5. -. -.- Isomerases.
5. 1. -.- racemases and epimerases.
5. 1. 1.- acting on amino acids and derivatives.
5. 1. 2.- acting on hydroxy acids and derivatives.
5. 1. 3.- acting on carbohydrates and derivatives.
5. 1.99.- acting on other compounds.
5. 2. -.- cis-trans-isomerases.
5. 3. -.- intramolecular oxidoreductases.
5. 3. 1.- interconverting aldoses and ketoses.
5. 3. 2.- interconverting keto- and enol- groups.
5. 3. 3.- transposing C=C groups.
5. 3. 4.- transposing S-S bonds.
5. 3.99.- other intramolecular oxidoreductases.
5. 4. -.- intramolecular transferases (mutases).

5. 4. 1.-	transferring acyl groups.
5. 4. 2.-	phosphotransferases (phosphomutases).
5. 4. 3.-	transferring amino groups.
5. 4.99.-	transferring other groups.
5. 5. -.-	intramolecular lyases.
5.99. -.-	other isomerases.
6. -. -.-	Ligases.
6. 1. -.-	forming carbon-oxygen bonds.
6. 1. 1.-	ligases forming aminoacyl-trna and related compounds.
6. 2. -.-	forming carbon-sulfur bonds.
6. 2. 1.-	acid–thiol ligases.
6. 3. -.-	forming carbon-nitrogen bonds.
6. 3. 1.-	acid–ammonia (or amine) ligases (amide synthases).
6. 3. 2.-	acid–amino-acid ligases (peptide synthases).
6. 3. 3.-	cyclo-ligases.
6. 3. 4.-	other carbon–nitrogen ligases.
6. 3. 5.-	carbon–nitrogen ligases with glutamine as amido-N-donor.
6. 4. -.-	forming carbon-carbon bonds.
6. 5. -.-	forming phosphoric ester bonds.

付録 B

計算機実験で使用した酵素データ

B.1 学習用データ

EC-number	Code	Enzyme name
1.-.-.-	B32344	isopenicillin N synthase - Streptomyces anulatus
1.1.1.140	DEECSP	sorbitol-6-phosphate 2-dehydrogenase - Escherichia coli
1.1.1.153	JQ1176	sepiapterin reductase - human
1.1.1.1	S09634	alcohol dehydrogenase - fruit fly
1.1.1.1	S25449	alcohol dehydrogenase - fruit fly
1.1.1.1	S26780	Alcohol dehydrogenase - Fruit fly
1.1.1.21	A60603	aldehyde reductase - rat
1.1.1.27	DEMSLM	L-lactate dehydrogenase chain M - mouse
1.1.1.62	DEHUE7	glucose-6-phosphate 1-dehydrogenase - fruit fly
1.11.1.7	A39889	Catalase - Mouse
1.11.1.7	S22087	Peroxidase precursor - Rice
1.13.11.3	S12962	Protocatechuate 3,4-dioxygenase - Escherichia coli
1.14.99.3	S00325	heme oxygenase (decyclizing) - human
1.17.4.1	RDVZVV	ribonucleoside-diphosphate reductase small chain - vaccinia virus
1.17.4.1	WMBEB2	ribonucleoside-diphosphate reductase small chain - human herpesvirus 2 (strain 333)
1.18.6.1	S15745	Nitrogenase iron protein - Azospirillum brasilense
1.2.1.12	DEBSG	glyceraldehyde-3-phosphate dehydrogenase - Bacillus subtilis
1.2.1.12	DELOG3	glyceraldehyde-3-phosphate dehydrogenase - lobster
1.2.1.12	S12696	Glyceraldehyde-3-phosphate dehydrogenase - Bacillus megaterium
1.2.1.13	DESPGA	glyceraldehyde-3-phosphate dehydrogenase (NADP+) (phosphorylating) A, chloroplast - spinach
1.2.4.4	DEPSEB	3-methyl-2-oxobutanoate dehydrogenase (lipoamide) chain E1-beta - Pseudomonas putida
1.3.1.33	S17823	protochlorophyllide reductase 35.5K chain - Rhodobacter capsulatus
1.9.3.1	A39653	cytochrome-c oxidase chain II precursor - cowpea
1.9.3.1	OTRZ3M	cytochrome-c oxidase chain III - rice mitochondrion
1.9.3.1	S14157	cytochrome-c oxidase chain II - sugar beet mitochondrion
2.1.1.-	XYECRO	rRNA (adenine-N6,N6-)-dimethyltransferase - Escherichia coli
2.1.1.48	A27741	rRNA (adenine-N6-)-methyltransferase - Streptomyces fradiae
2.1.1.72	S09361	site-specific DNA-methyltransferase (adenine-specific) dam - phage T4
2.1.3.3	OWECI	ornithine carbamoyltransferase chain I - Escherichia coli
2.1.4.2	B26984	inosamine-phosphate amidinotransferase - Streptomyces griseus
2.3.1.39	S20443	[Acyl-carrier-protein] malonyltransferase - Escherichia coli
2.3.1.5	XYCHY0	arylamine N-acetyltransferase (clone p-NAT-10) - chicken
2.3.1.74	S21444	Naringenin-chalcone synthase - Soybean
2.3.1.81	S09651	aminoglycoside N3'-acetyltransferase isozyme II - Enterobacter cloacae
2.3.2.5	A41535	glutaminyl-peptide cyclotransferase precursor - bovine
2.4.1.-	JC1275	UPD-N-acetylglucosamine–N-acetylmuramyl-(pentapeptide)pyrophosphoryl-undecaprenol N-acetylglucosamine transferase - Bacillus subtilis
2.4.1.27	XUBPB4	DNA beta-glucosyltransferase - phage T4
2.4.2.17	XREBT	ATP phosphoribosyltransferase - Salmonella typhimurium
2.4.2.9	JH0147	uracil phosphoribosyltransferase chain FUR1 - yeast (Saccharomyces cerevisiae)
2.5.1.15	D37854	dihydropteroate synthase homolog - Bacillus subtilis
2.7.1.-	S15519	Protein kinase ERK1 - Human
2.7.1.-	TVBEG1	kinase-related transforming protein - equid herpesvirus 1 (strain Ab4p)
2.7.1.37	A25334	protein kinase, cAMP-dependent, catalytic chain - bovine
2.7.1.37	A34724	protein kinase, cAMP-dependent, catalytic chain - bovine
2.7.1.6	KIECGG	galactokinase - Escherichia coli
2.7.1.95	PKSOJF	kanamycin kinase - Enterococcus faecalis plasmid pJH1
2.7.2.11	KISEEM	glutamate 5-kinase - Serratia marcescens
2.7.3.2	A42078	creatine kinase B - mouse
2.7.7.49	S16653	RNA-directed DNA polymerase - Escherichia coli
2.7.7.6	A25968	DNA-directed RNA polymerase 40K chain - yeast (Saccharomyces cerevisiae)
2.7.7.6	RNECA	DNA-directed RNA polymerase alpha chain - Escherichia coli
2.7.7.7	A25445	DNA-directed DNA polymerase beta chain - rat
2.8.2.1	S10329	Aryl sulfotransferase - Rat
3.1.1.11	A25010	pectinesterase - tomato

3.1.3.11	PAEC	fructose-bisphosphatase - Escherichia coli
3.1.3.16	B36076	phosphoprotein phosphatase 2A, ppa2 - yeast (Schizosaccharomyces pombe)
3.1.3.16	PART2B	phosphoprotein phosphatase 2A-beta catalytic chain - rat
3.1.3.16	S30854	phosphoprotein phosphatase - yeast (Saccharomyces cerevisiae)
3.1.3.25	S24343	myo-Inositol-1(or 4)-monophosphatase - African clawed frog
3.1.3.48	A33897	protein-tyrosine-phosphatase 1B, placental - human
3.1.3.48	S28392	protein-tyrosine-phosphatase pyp3 - yeast (Schizosaccharomyces pombe)
3.2.1.-	A40176	xylanase - Thermoascus aurantiacus
3.2.1.14	A38664	chitinase precursor - barley
3.2.1.14	S19794	chitinase III, basic - common tobacco
3.2.1.35	HUBPHA	hyaluronoglucosaminidase - Streptococcus pyogenes phage H4489A
3.2.1.39	S12013	glucan endo-1,3-beta-D-glucosidase sp41a precursor - common tobacco
3.2.1.73	A25455	licheninase precursor - barley
3.2.2.20	A41230	DNA-3-methyladenine glycosidase I - human
3.4.17.2	CPBOB	carboxypeptidase B - bovine
3.4.21.-	A32340	tonin precursor - rat
3.4.21.-	PRMVJA	proteinase - sheep pulmonary adenomatosis virus
3.4.21.35	KQRTP	tissue kallikrein precursor, pancreatic - rat
3.4.21.36	ELRT2	pancreatic elastase II precursor - rat
3.4.21.50	A32960	lysyl endopeptidase - Achromobacter lyticus
3.4.22.16	KHRTH	cathepsin H precursor - rat
3.4.23.21	B41415	rhizopuspepsin II - Rhizopus chinensis
3.4.23.6	PEPLBJ	Microbial aspartic proteinase - Penicillium janthinellum
3.4.24.27	HYBST	thermolysin - Bacillus "thermoproteolyticus"
3.4.99.46	SNHUC8	multicatalytic endopeptidase complex chain C8 - human
3.5.1.28	S16016	N-acetylmuramoyl-L-alanine amidase - Streptococcus pneumoniae phage HB-3
3.5.1.52	A38365	peptide-N4-(N-acetyl-beta-glucosaminyl)asparagine amidase precursor - Flavobacterium meningosepticum
3.6.1.34	JQ1144	H+-transporting ATP synthase chain b precursor, mitochondrial - human
3.6.1.34	PWECA	H+-transporting ATP synthase alpha chain - Escherichia coli
3.6.1.37	S09601	Na+/K+-transporting ATPase beta chain - mouse
4.1.1.28	A43758	aromatic-L-amino-acid decarboxylase - bovine
4.1.1.64	A35173	2,2-dialkylglycine decarboxylase (pyruvate) - Pseudomonas cepacia
4.1.2.15	A35253	2-dehydro-3-deoxyphosphoheptonate aldolase - Salmonella typhimurium
4.1.3.5	A35865	hydroxymethylglutaryl-CoA synthase - rat
4.2.1.1	A35163	carbonate dehydratase precursor - spinach chloroplast
4.2.1.1	A35795	carbonate dehydratase precursor - Chlamydomonas reinhardtii
4.2.1.20	A35407	tryptophan synthase beta chain - Thermus aquaticus
4.3.2.2	A35291	adenylosuccinate lyase - chicken
4.4.1.14	A35516	1-aminocyclopropane-1-carboxylate synthase(clone pcVV4A) - tomato
4.4.1.14	A40960	1-aminocyclopropane-1-carboxylate synthase 2 - tomato
5.2.1.8	A41581	eptidylprolyl isomerase 3 precursor - human
5.3.1.24	A34091	phosphoribosylanthranilate isomerase - Acinetobacter calcoaceticus
6.3.1.2	A43754	glutamate-ammonia ligase - rat
6.3.1.2	A44095	glutamate-ammonia ligase
6.3.4.15	A24989	Biotin-[acetyl-CoA-carboxylase] ligase
6.3.5.5	A35111	carbamoyl-phosphate synthase (glutamine-hydrolyzing) small chain - Pseudomonas aeruginosa

B.2 テスト用データ

使用したテスト用酵素について，PIR のコードを以下に示す．

EC1 Oxidoreductase

B23724 C23724 D23724 E23724 S07439 S09633 S18273 S20715 S25918 S26768
S26769 S26770 S26779 S26784 S26785 S06591 A36070 A43944 DEDFLM DELBLA
JQ0183 JU0280 S00019 S06290 S22492 DEBSXS JH0531 S15315 A32937 CCBO17
S16572 JC1249 JU0457 OPNB7 S04763 S22505 B31047 JX0071 JX0151 OHRTD A26916
S06735 S24585 WMBE18 WMBE32 WMBEB4 WMBES7 A26931 JS0238 NIZRF NIZRFM NIZRFT
S06984 S08048 S15747 A22366 B22366 DEBSGF DEKWG1 DEKWG3 DEZMGC JH0769
JH0770 JQ1285 JQ1286 JQ1287 S06879 S16508 S25596 S26976 DEPMNA DEPMNB
DESPGB DEZMG3 S19255 S19721 DNOBU3 S08425 S10199 S16157 S16887 S25944
C34285 S28250 A29180 B41260 E34284 OBBO2 OBHU2 OBJJMP OBNC2 OBOB2M OBPC2N
OBRZ2 OBWT2 OBXL2 OBZM2 OTHU3 OTMS3 S01503 S01785 S01960 S04753 S05493
S10193 S10303 S11029 S13105 S14207 S14455 S17911 S20147 S21343 S22198
S26023 S26035 S26949

EC2 Transferase

JS0635 XYSMRE XYBPT2 XYBPT4 OWBS OWECF OWNHG OWPSCA OWPSY S24718 S24719
S24723 S24727 S24728 S24734 S24746 S24749 XJSMIG S13174 A28168 XYCHY3
S12224 S18136 S20931 S20933 S26414 S26415 SYPJCA SYPJCB SYPJCD SYPJCJ
SYPJCN SYSKCD SYSYC1 SYSYC3 SYSYCN SYZMW1 JS0652 S18730 A43661 S10555
S14347 S27111 A34740 A38578 A41227 JQ0967 KIHUC1 KIMSCE KIMVT8 KIRBC1
KIRTC1 S00217 S15714 S19051 S22258 S23428 S25011 S28548 S30095 TVBEKA
TVBEPS S17552 S18012 S18013 S18014 S18015 A23690 A23716 A27070 A31248
A34106 A35755 A36371 A39360 A40444 B27070 B34724 B40033 C26911 C27070
JQ1150 JS0178 OKBOG S05034 S05702 S17999 S18000 S29478 S13434 S16071 C37760
S27988 PKSAF A41893 KIECEG A23590 A24686 A24793 A26387 A30789 A31431 A31793
A35682 A35756 B37059 KIRBCM KIRTCB KIRTCM KIRYCT S17188 S17189 S24612
A39879 E32307 S13348 S14232 A27112 S22212 S25763 JN0479

EC3 Hydrolase

S00629 S25171 S25172 PSNJ1W PSNJ2W PSNJ3W PAWTF S14060 S16582 A28029 A36076
A40942 A41152 B41152 JX0157 PABY21 PABY22 PAFF1A PAHU2A PARB11 PARB2B
PARBA1 PART2A S09417 S10371 S12500 S12501 S13827 S13828 S13829 S14773
S16808 S16809 S20348 S29396 S29985 S31252 S31265 S14991 A33899 A34845

A35992 A60345 JN0317 S12053 S23126 A25898 B45511 JQ0965 S05426 S08627
S13322 S14948 S15997 S18750 S20981 S25634 S25637 S26625 S12522 B38257
S13734 S13735 S13830 S25126 JS0589 A32128 A34487 A39246 B32129 CPRTA S29127
A32129 A41204 A30971 B33359 C23863 EGMSB KQRTTN PRMVMM A31136 A33359 B31136
D23863 KQMS1 S01971 A25528 A26823 B26823 A32687 JS0597 KHCHL S15844 S26871
A41415 A28672 JU0343 PS0140 S21358 S22136 B36706 HYBS HYBSN HYBSS S22690
A41042 A41335 A42464 A43575 PN0114 S21934 SNRTC8 A35759 A35760 S28866
A39732 B39732 C36493 D25797 EWBY8 LWBYA LWZMA PWBYG PWFF8 PWFF8Y PWNTG
PWYCG S00763 S04751 S07188 S17721 S17994 A24862 A25768 S21088

EC4 Lyase

JS0754 A34890 S11492 S12989 DCHUA DCJAAP DCRTA DEGPA A36505 C36044 ADECH
JN0322 S22840 S26055 S12040 S12736 S13000 JE0045 S16798 S10200 S13189
S14188 S28412 S28796 S28797 A31393 A32959 A33929 JQ1073 JU0401 TSBYAB
S24974 A41141 B35516 B41141 S19677 S19678 S19679 S31450 B29010

EC5 Isomerase

A40515 CSHYAC CSRTA CSYC42 S13758 ISECTP

EC6 Ligase

SYGPPP SYRTPP A39827 B39827 A42224 AJBHQ AJFF1M AJFF2C AJMSQ AJRTQ S18600
S01319 SYBYCS SYECCS

謝辞

本研究を進めるにあたり，御指導していただいた土井淳多教授に心から感謝の意を表します．また，適切な助言をいただいた清水謙多郎助教授，池口満徳助手，中村周吾助手，大学院生の河野秀俊氏，升屋正人氏，川端猛氏，田崎康一氏，木下宏氏をはじめ生物情報工学研究室の諸氏，株式会社ソニー木原研究所の石井隆寛氏，貴重なデータを準備してくださった大阪大学医学部の桑原裕祐氏に感謝致します．

参考文献

[1] W. Burks. *Electronic Computing Circuits of ENIAC.* the University Illinois Press, 1943.

[2] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bullet. Math. Biophysics*, Vol. 5, pp. 115–113, 1943.

[3] F. Rosenblatt. The perceptron : A probablistic model for information storge and organization in the brain. *Psychol. Rev.*, Vol. 65, No. 6, pp. 386–408, 1958.

[4] D. E. Rumelhart and J. L. McClelland. *Parallel Distributed Processing.* MIT Press, 1986.

[5] J. J. Hopfield and D. W. Tank. Neural computation of decisions in optimizaition problems. *Biological Cybernetics*, Vol. 52, pp. 141–152, 1985.

[6] Sejnowski T.J. Hilton G.E. and Ackley D.H. Boltzmann machines:cosraint satisfaction networks that learn. *Tech,Rep,CMUCS-84-119*, 1984.

[7] Jonathan D. Hirst and Michael J. E. Sternberg. Prediction of structural and functional features of protein and nucleic acid sequences by artificial neural networks. *Biochemistry*, Vol. 31, No. 32, pp. 7211–7218, 1992.

[8] G. D. Stormo, Gold L. Schneider, T. D., and A. Ehrenfeucht. Use of the 'perceptron' algorithm to distinguish translational initiation sites in E. coli. *Nucleic Acids Research*, Vol. 10, No. 9, pp. 2997–3011, 1982.

[9] K. Nakata, M. Kanehisa, and C. DeLisi. Prediction of splice junctions in mrna sequences. *Nucleic Acids Research*, Vol. 13, No. 14, pp. 5327–5340, 1985.

[10] S. Brunak, J. Engelbrecht, and S. Knudsen. Prediction of splice junctions in mrna sequences. *J. Mol. Biol*, Vol. 220, pp. 49–65, 1991.

[11] R. Farber and A. Lapedes. Determination of eukaryotic protein coding reigions using neural network and information theory. *J. Mol. Biol*, Vol. 226, pp. 471–479, 1992.

[12] E. C. Uberbadher and R. J. Mural. Locating protein-coding regions in human dna sequences by a multiple sensor neural network approach. *Proc. Natl. Acad. Sci. USA*, Vol. 88, pp. 11261–11265, 1991.

[13] M. C. O'Neill. Training back-propagation neural networks to define and detect dna-binding sites. *Nucleic Acids Ressearch*, Vol. 19, pp. 313–318, 1991.

[14] N. Qian and J. Sejnowski. Prediction the secondary structure of globular proteins using neural network models. *J. Mol. Biol*, Vol. 202, pp. 865–884, 1988.

[15] P. Y. Chou and G. D. Fasman. Conformational parameters for amino acids in helical,sheet,and random coil regions calculated from proteins. *Biochemistry*, Vol. 13, pp. 211–222, 1974.

[16] L. H. Holley and M. Karllus. Protein secondary structure prediction with a neural network. *Proc. Natl. Acad. Sci. USA*, Vol. 86, pp. 152–156, 1989.

[17] H. Bohr, J. Bohr, S. Brunak, R. M. J. Cotterill, B. Lautrup, L. Norskov, O. H. Olsen, and S. B. Petersen. Protein secondary structure and homolgy by neural networks - the alpha-helices in rhodopsin. *FEBS Lett.*, Vol. 241, pp. 223–228, 1988.

[18] M. J. McGregor and Sternberg. Prediction of bata-turns in proteins using neural networks. *Protein Engineering*, Vol. 2, No. 7, pp. 521–526, 1989.

[19] S. M. Muskal and S. R. Holbrook. Prediction of the disulfide-bonding state of cystein in protein. *Protein Engineering*, Vol. 3, No. 8, pp. 667–672, 1990.

[20] S. R. Holbrook, Muskal S. M., and Kim S. H. Predicting surface exposure of amino acids from protein sequence. *Protein Engineering*, Vol. 3, No. 8, pp. 659–665, 1990.

[21] J. D. Hirst and M. J. Sternberg. Prediction of atp-binding motifs : a comparison of a perceptron-type neural network and a consensus sequence method, protein engineering. *Protein Engineering*, Vol. 4, No. 6, pp. 615–623, 1991.

[22] Y. Benjio and Y. Pouliot. Efficient recognition of immunogloblin domains from amino acid sequences using a neural network. *CABIOS*, Vol. 6, No. 4, pp. 319–324, 1990.

[23] Ladunga I., Czako F., Csabai I., and Geszti T. Improving singal peptide prediction accuracy by simulated neural network. *CABIOS*, Vol. 7, No. 4, pp. 485–487, 1991.

[24] H. Bohr, J. Bohr, S. Brunak, R. M. J. Cotterill, H. Fredholm, B. Lautrup, and S. B. Petersen. A novel approach to prediction of the 3-dimensionalstructures of protein backbones by neural networks. *FEBS Lett.*, Vol. 261, No. 1, pp. 43–46, 1990.

[25] G. L. Wilcox, M. Poliac, and M. N. Liebman. Neural network analysis of protein tertiary structure. *Tetrahedron Computer Methodology*, Vol. 3, pp. 191–221, 1990.

[26] David H. Hubel and Torsten N. Wiesel. Brain mechanisms of vision. *Scientific American*, Vol. 241, No. 3, pp. 130–144, 9 1979.

[27] David H. Hubel. The brain. *Scientific American*, Vol. 241, No. 3, pp. 39–47, 9 1979.

[28] D. O. Hebb. *The Organization of Behavior.* Wilkey, New York, 1949.

[29] R. A. Jacobs. Increased rates of convergence through learning rate adaptation. *Neural Networks*, Vol. 1, pp. 295–307, 1988.

[30] Scott E. Fahlman. An empirical study of learning speed in back-propagation networks. Technical Report Computer Science Technical Report, 1988.

[31] W. Schiffmann, M. Joost, and R. Werner. Comparison of optimized backpropagation algorithm. In *Proceedings of the European Symposium on Artificial Neural Networks*, pp. 97–104, San Diego, CA, 1993. ESANN.

[32] Etienne Barnard. Optimization for training neural nets. *IEEE Transactions on Neural Networks*, Vol. 3, No. 2, pp. 232–240, March 1992.

[33] T. Battiti. First- and second-order methods for learning: Between steepest descent and newton's method. *Neural Computation*, Vol. 4, No. 2, pp. 141–166, 1992.

[34] Martin Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, Vol. 6, No. 4, pp. 525–533, June 1993.

[35] D. L. Reilly, L. N. Cooper, and C. Elbaum. A neural model for category learning. *Biological Cybernetics*, Vol. 45, pp. 35–41, 1982.

[36] Akira Iwata, Yukata Ino, Ken ichi Hotta, and Nobuo Suzumura. *Hand-written Japanese Kanji character recognition by a structured self-growing neural network "CombNET-II"*, Vol. 2, pp. 1189–1192. Elsevier Science Publishers B.V., 1992.

[37] Scott E. Fahlman and Christian Lebiere. The Cascade-Correlation learning architecture. Technical Report CMU-CS-90-100, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, February 1990.

[38] Scott E. Fahlman. The recurrent cascade correlation architecture. Technical Report CMU-CS-91-100, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, May 1991.

[39] Markus Hoehfeld and Scott E. Fahlman. Learning with limited numerical precision using the cascade-correlation algorithm. *IEEE Transactions on Neural Networks*, Vol. 3, No. 4, pp. 602–611, 1992.

[40] Masafumi Hagiwara. Novel back propagation algorithm for reduction of hidden units and acceleration of convergence using artificial selection. In *Proc. Int. Joint Conf. on Neural Networks, vol. 1*, pp. 625–630, 1990.

[41] 藤井善行, 増田達也. ユニットの合成による多層型ニューラルネットワークのコンパクト構造化. 電気学会全国大会講演論文集, Vol. 13, pp. 18–19, 1991.

[42] D. Whitley. Applying genetic algorithms to neural net problems. *Neural Networks*, Vol. 1, p. 230, 1988.

[43] H. Kitano. Designing neural networks using genetic algorithms with graph generation sytem. *Complex Systems*, Vol. 4, No. 4, pp. 203–222, 1990.

[44] D. B. Fogel, L. J. Fogel, and V. W. Porto. Evolving neural networks. *Biological Cybernetics*, Vol. 63, pp. 487–493, 1990.

[45] David B. Fogel. Evolving behaviors in the iterated prisoner's dilemma. *Evolutionary Computation*, Vol. 1, No. 1, pp. 77–97, 1993.

[46] David B. Fogel. An introduction to simulated evolutionary optimization. *IEEE Transactions on Neural Networks*, Vol. 5, No. 1, pp. 3–14, 1994.

[47] John von Neumann. *Theory of Self-Reproducing Automata*. the University of Illinois Press, Urbana, 1966.

[48] 伊理正夫, 白川功. 演習グラフ理論. コロナ社, 1983.

[49] Proc. Int. Joint Conf. Art. Intelligence. *Web grammars*. Washigton, 1969.

[50] J. L. Pfaltz. Web grammars and picture description. *Computer Graphics and Image Process.*, Vol. 1, No. 2, pp. 193–219, 1972.

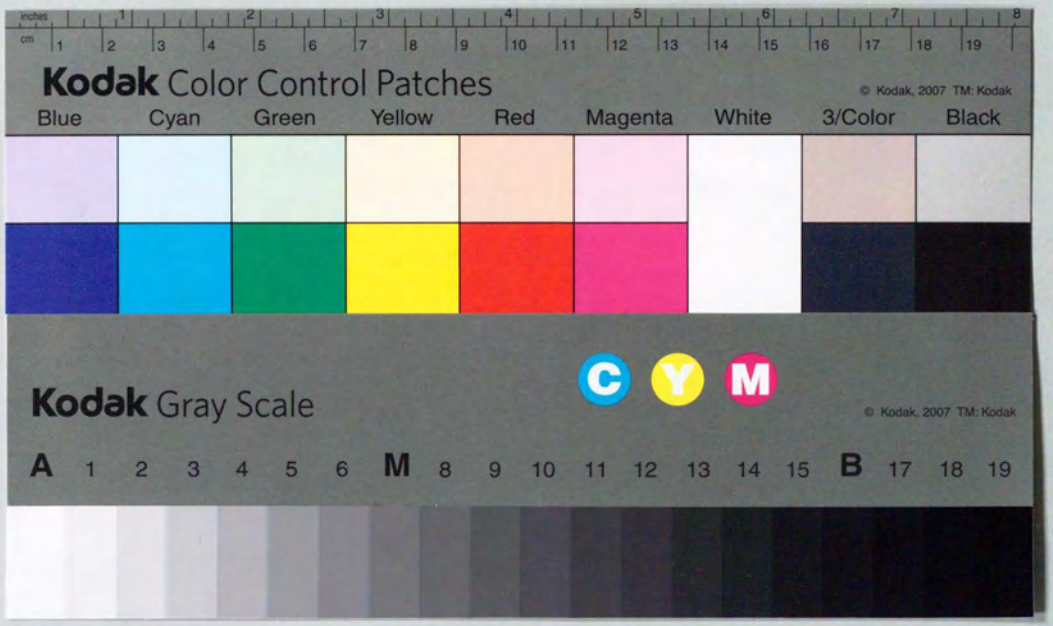
[51] B. Irie and S. Miyake. Capabilities of three–layered perceptrons. In *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 1, pp. 641–648, San Diego, CA, 1988. IEEE.

[52] The Nomenclature Committee of the International Union of Biochemistry, Molecular Biology on the Nomenclature, and Classification of Enzymes. *Enzyme Nomenclature*. Academic Press, Inc., 1992.

[53] X. Zhang, J. P. Mesirov, and D. L. Waltz. Hybrid system for protein secondary structure prediction. *J. Mol. Biol.*, Vol. 225, pp. 1049–1063, 1992.

[54] C. Branden and J. Tooze. タンパク質の構造入門. 教育社, 1991.

[55] Carl Branden and John Tooze. *Introduction to Protein Structure*. Garland Publishing, Inc., 1991.


inches
1 2 3 4 5 6 7 8
cm
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
Kodak Color Control Patches
© Kodak, 2007 TM: Kodak
Blue
Cyan
Green
Yellow
Red
Magenta
White
3/Color
Black
C
Y
M
Kodak Gray Scale
© Kodak, 2007 TM: Kodak
A 1 2 3 4 5 6 M 8 9 10 11 12 13 14 15 B 17 18 19