修 士 論 文

階層型混成ニューラルネットワークモデルによる追記学習に関する研究

指導教官　日比野靖 教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

小川知之

1999 年 2 月 15 日

目 次

第 1 章

諸言

一般にニューラルネットワークはモデルの形態の如何に関わらず、後から新規パターンのみを学習させる追記学習を行なうことが非常に難しいという問題点を抱えている。これは、学習済みのニューラルネットワークに新しいパターンを記憶させる場合において、学習済みのネットワークに新しいパターンのみを学習させると、以前に学習したパターンを想起するためのネットワークの状態を破壊してしまう可能性があるため、通常、記憶させたい新規のパターンだけではなく、先の学習時に提示したすべてのパターンと共に学習を行なう必要がある、ということに由来する。

このような記憶の安定性と可塑性ジレンマに関する問題に対処するため、これまで多くの研究が行なわれてきてはいるが、新規パターン以外の情報の必要性、学習の収束に関する問題等により現在までに十分な成果が上がっているとは言い難い [6][7][8][9][10][11]。

このような状況の中で、Grosbarg、Carpenter らはこのジレンマを解決するための 1 つの方法として競合学習を用いた適応共鳴理論（ART：Adaptive Resonance Theory）を提案した [1][2][3]。

ART は競合学習モデルをベースにしているが、自律的に信号の流れを制御するための制御機構を内蔵することで、上述のジレンマを解決している。しかし、本質的な部分は単層の競合学習モデルであることから、線形分離不可能なパターン群の分類ができないといった問題を抱えている。

一方、ニューラルネットワークを一つの処理単位として考え、同種のアーキテクチャを持つネットワークをいくつか組み合せたり、或は異なるアーキテクチャのモデルを組み合わせることで、単体のモデルの持っている機能の向上や、新しい機能を実現するという試みがなされている。特に、Hecht-Nielsen による Counterpropagation Network は、バックプロパゲーション等の学習法に比べて、10〜100 倍の学習速度持ち比較的良い精度の学習

を行なうことで知られており、さまざまな応用研究がなされている [12][13][14][15][16]。

このような背景を踏まえ、本論文では、ART と教師ありの学習層を組み合わせることにより安定した追記が可能で、かつ、任意のパターンの分離能力を有するネットワークモデルを提案し、その学習速度、追記性能等について検討した。

また、多くの文献で分類され語のパターンの組を表現する語として、主に “カテゴリ”、“クラスタ”、“クラス” の 3 語を使用しているが区別されていない。そこで、今後の議論で混乱しないようこれらの 3 語について定義を与え、それにより区別して使用することを提案する。

第 2 章

ニューラルネットワーク概説

本章では、本論文で頻繁に出てくる用語 “カテゴリ” と “クラス” を定義し、教師あり学習を行なう単層ネットワークモデルの単純パーセプトロン、及び教師なし学習におけるオヤ則と競合学習モデルについて、その学習アルゴリズムや問題点について述べる。

ニューラルネットワークにおける「層」の数え方は文献によってまちまちであるが、本論文ではこの章以降、入力を受ける一連の素子群は層として数えないことにする。このことは、入力素子は入力を次の層へ配分しているだけで実質的には何も仕事をしていないことから、合理的な解釈であると考えられる[1]。

2.1 カテゴリとクラスの定義

さまざまな文献で入力空間の分類について述べられているが、分類されてできた塊の呼び方は統一されていない。一般的には “カテゴリ (category)”、“クラス (class)”、“クラスタ (cluster)” の 3 つの語のいずれかが使用されているものの、文献によってまちまちなのが現状である。

本論文では、これらの曖昧さをなくすため、これらの 3 つの語に以下のような意味を持たせ適切に使い分けることを提案する。

カテゴリ (category)： 基本的、或は最小の分類単位。1 つのパターンが 1 つのカテゴリになり得る。

クラスタ (cluster)： いくつかの似ているカテゴリの集合。他のクラスタと重ならないカテゴリを含むことができるが、含まれるカテゴリは何らかの

[1] 現在では、この考え方が主流になりつつあるようだ。

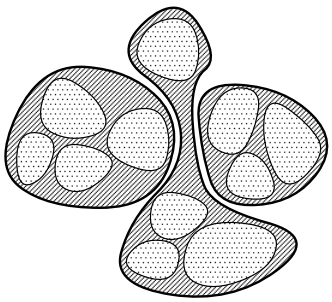


図 2.1: **カテゴリとクラス**　点塗の集合がカテゴリであり、斜線で塗られているのがクラスである。クラスは他のクラスと重ならない任意のカテゴリを含むことができる。

尺度（ハミング距離等）により似ている判断されたものでなければならない。

クラス (class)：いくつかの任意のカテゴリの集合。クラスタと似ているが、1 つのクラスには他のクラスと重ならない任意のカテゴリを含むことができ、カテゴリの空間配置等による制限を受けない[2]。

図 2.1にこれらの違いを摸式的に示す。左右にある 2 つのクラスは、クラスタであるかも知れない。

2.2　教師あり学習モデル

教師あり学習モデルには、単層ネットワークである単純パーセプトロンや、バックプロパゲーションモデルに代表される階層的に処理層を配置した多層型ネットワークがあるが、以下では特に本研究で必要になる単純パーセプトロンについて簡単に説明する。

2.2.1　パーセプトロン

形式ニューロン

まんず、パーセプトロンを構成する基本処理ユニットについて簡単に述べる。

この基本処理ユニットは、1943 年に McCullock と Pitts により提案された形式ニューロン (基本ニューロン、図 2.2) と呼ばれるものである。

[2] この定義から、“クラスタ（cluster）” はクラスの特別な場合であることが分かる。

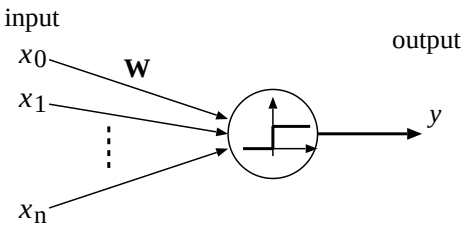


図 2.2: **形式ニューロン**

形式ニューロンは、いくつかの入力素子群からの入力信号の重み付き総和が、ある閾値を越えると発火し、出力を出す。入力を x_i $(1 \leq i \leq N)$、結合荷重を w_i $(-1 \leq w_i \leq +1)$、出力を o とすると、o は下式で与えられる。

$$o = f\left(\sum_{i=1}^{N} w_i x_i\right) \tag{2.1}$$

ここで、$f(x)$ はステップ関数であり、

$$f(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{2.2}$$

と表される。

要約すれば、式 (2.2) のステップ関数を用いた場合、形式ニューロンは、入力の重み付き総和が 0 を越えれば発火し、越えなければ発火しない，と言うことである。

アーキテクチャ

パーセプトロンは、1962 年に Frank Rosenblatt によって提案された単層の教師あり学習モデルであり、形式ニューロンをいくつか組み合わせた形をしており、入力素子群と出力層は完全に結合している (図 2.3)。

パーセプトロンは、与えられた入力のある種のコーディングされたものを出力とする。このとき、外部からの教師信号によって、この出力結果が正しいか否かを決定し、すべての入力に付いて正しい出力が出せるように学習を行なう。

学習アルゴリズム

パーセプトロンの学習 (訓練) は、入力パターンと教師信号の対を 1 組としたいくつかの組を順番に繰返し提示し、出力層素子が各入力パターンに対する正しい出力を出すよう

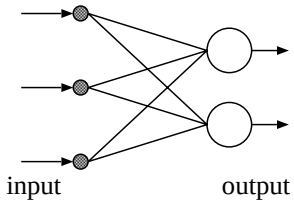


図 2.3: **perceptron**

になるまで、結合荷重を修正していくことで行なわれる。

この時、出力が教師信号と一致していない場合はその素子の出力が正しくなるように結合を強化し、一致していれば何もしない、というように結合荷重を変更していくことにすれば、j番目の出力素子 o_jと i 番目の入力素子 x_iとの間の結合荷重 w_{ij}の更新式は、

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij} \tag{2.3}$$

である。現在の出力 o_jに対する教師信号を t_jとすれば、Δw_{ij}は、

$$\Delta w_{ij} = \begin{cases} 2\eta\, t_j x_i & (\text{if } t_j \neq o_j) \\ 0 & (\text{otherwise}) \end{cases} \tag{2.4}$$

或は、

$$\Delta w_{ij} = \eta\,(t_j - o_j) x_i \tag{2.5}$$

で与えられる (但し、η は学習の速さを決める学習率である)。

パーセプトロンの限界

良く知られているように、パーセプトロンでは線形分離不可能なパターンを分類することができない。これは、適応 (変更) 可能な結合荷重を持つ層を 2 つ以上持つことができないことに由来する。しかし、線形分離可能なパターンであれば、有限回の学習規則の適用によって必ず正しい結合荷重が得られることは保証されている。

2.3　教師なし学習モデル

教師なし学習においては、回路網の出した出力に対してそれが正しいかどうかを判定するための (入力と対になる) データは存在しない。したがって、回路網自身が (自分の為に) 入力データのパターン、特徴、規則性、相関を発見し、それによって自分自身を変更し自己組織化していく必要がある。

回路網が教師なし学習による自己組織化によって、出力がどのような意味を持つようになるのかを以下に示す。

親しさ： 新しい入力パターンが、過去に見られた典型的、平均的なパターンとどのくらい似ているのかを示す出力を出す。したがって、回路網は学習によって典型的なものを学んでいく。

集団化： 入力パターンの相関に基づいて適切なカテゴリが回路網により見い出されなければならない。出力は入力パターンがどのカテゴリに属しているのかを示す。

コード化： 出力は入力より少ないビットを用い、関連する情報をできるだけ多く保存する。

特徴写像： 幾何学的配列を持つ出力層に入力の位相的な像を作っていくことで、類似の入力パターンがいつも近くの出力素子を刺激するようにする。これにより、出力が組織化されていく。

これらは、はっきりと区別されるものではないが、ネットワークが与えられた入力から発見した特徴、規則性とは何かを知る一つの目安となる。

以下ではこのような自己組織化を行なう、ヘッブ則を元にしたオヤ則と**勝者がすべてを得る** (winner-take-all) 競合学習の 2 つの学習方法における結合荷重の更新則について述べる。

2.3.1　オヤ則

図 2.4に示されているような 1 つの線形出力素子を持つ場合について考えると、出力 O は、

$$O = \sum_j w_j x_j = \boldsymbol{w} \cdot \boldsymbol{x} \tag{2.6}$$

である。この時出力 $|O|$ は、ある入力ベクトル ($\boldsymbol{x}$) に対して平均的に大きな値を出力するようになることが望まれる。

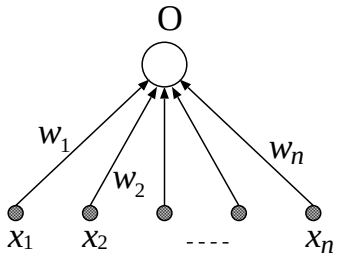


図 2.4: **線形出力素子**

したがって、単純なヘッブ則

$$\Delta w_i = \eta\, O\, x_i \qquad (\eta : \text{学習率}) \tag{2.7}$$

により学習を行なうことが考えられる。しかし、式 (2.7) による学習では、結合荷重 $\boldsymbol{w}$ が際限なく大きくなり続けて学習が収束しない。

オヤ [17] は、ヘッブ則に重みの崩壊項を加えることで発散を防ぎ、$|\,\boldsymbol{w}\,| = 1$ に近付けるオヤ則を提案した。つまり、

$$\Delta w_i = \eta\, O(x_i - O w_i) \tag{2.8}$$

である。この式 (2.8) により $\boldsymbol{w}$ の方向は、O^2 の平均が最大になるように修正されていく。このことは、主成分解析における入力データの第 1 主成分の抽出に対応する。

このように、オヤ則による学習における出力は、入力パターンの特徴を表す。

2.3.2 競合学習

入力素子群 $\boldsymbol{x}$ と 1 層の出力層 ($\boldsymbol{O}$) が結合荷重 $\boldsymbol{W}$ によって完全結合していて、2 進 (binary) の入出力を行なう単純な競合学習モデルを考える。競合学習モデルにおいては、ある入力パターンに対して出力層は、**勝者**と呼ばれるただ 1 つの素子のみが出力を出すことができる。

つまり、ある入力パターン $\boldsymbol{x}$ に対して、

$$O_j = \sum_i w_{ij} x_i = \boldsymbol{w}_j \cdot \boldsymbol{x} \tag{2.9}$$

とすると、

$$\boldsymbol{w}_{j^*} \cdot \boldsymbol{x} \;\geq\; \boldsymbol{w}_j \cdot \boldsymbol{x} \qquad (\text{for all } j) \tag{2.10}$$

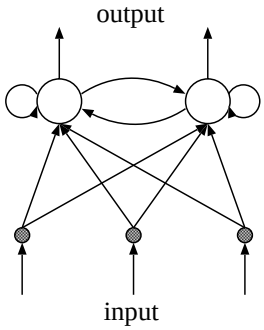


図 2.5: **競合学習モデル**　入力素子群と出力層は完全結合で、
出力層の各素子は側抑制性の結合を持っている。

となる j を勝者 j^*と定義し、勝者素子 j^*の出力を $O_{j^*} = 1$ とする。

ここで、式 (2.10) においてすべての jに対し重みが規格化 ($|\boldsymbol{w}_j| = 1$) されていれば、これは、

$$|\boldsymbol{w}_{j^*} - \boldsymbol{x}| \leq |\boldsymbol{w}_j - \boldsymbol{x}| \tag{2.11}$$

に等しい

ここで決定した勝者素子は、将来同じ入力が与えられた時にも当然勝者になるべきであるので、ベクトル $\boldsymbol{w}_j^*$を現在の入力 $\boldsymbol{x}$ に近くするように、勝者素子のみに対して結合荷重を更新するように学習を進行することが望ましい。

したがって、結合荷重の更新則は、

$$\Delta w_{ij} = \begin{cases} \eta\,(x_i - w_{ij}) & (\text{ if } \quad j = j^* \,) \\ 0 & (\text{ otherwise }) \end{cases} \tag{2.12}$$

或は、勝者の定義より、$O_{j^*} = 1$、$j \neq j^*$である出力素子に対しては $O_j = 0$ であるから、

$$\Delta w_{ij} = \eta\, O_j (x_i - w_{ij}) \tag{2.13}$$

とすることができる。

このような学習を行なうことで、ネットワークは入力パターンをカテゴリに分類するようになる。

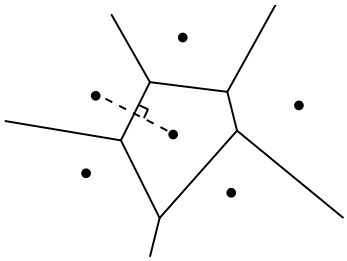


図 2.6: **Voronoi tessellation**　カテゴリの境界は、それを代表する 1 対の隣接するパターン (黒点) を結ぶ直線の垂直 2 等分線 (面) である。

2.4　パターンの分類

パーセプトロンの節でも述べたように、1 層の単純パーセプトロンでは線形分離不可能な、つまり、1 本の直線或は 1 つの超平面で 2 つに分けることのできない入力パターン群をうまく分類することはできない。そのようなパターンをうまく分類するには、非線形素子を用いた多層モデルにする必要がある。

では、競合学習モデル[3]ではどうだろうか。入力をいくつかのカテゴリに分類するような競合学習モデルでは、一般に入力空間を図 2.6に示されるボロノイ・テセレイション (Voronoi tessellation) に代表されるような空間に分割する。つまり、学習の結果、入力を最も良く表現する結合荷重ベクトルを求めることで入力空間を分類する。この場合の分類では、例えば、2 入力 XOR 問題においては、入力空間を 4 つに分類することになる。

結論としては、教師ありの単純パーセプトロンでは線形分離不可能なパターンをうまく分類することは不可能であるのに対して、競合学習モデルでは分類することはできるが、どのカテゴリが同じクラスに属するのかまでは知ることはできない，と言うことになる。

[3]入力パターンを分類するという目的には非競合学習モデルよりは向いていると考えるので、競合学習モデルを比較対象とする。

第 3 章

適応共鳴理論 (ART)

1987 年に G.A.Carpenter と S.Grossberg によって発表された適応共鳴理論 (Adaptive Resonance Theory)[1] は、電気生理学的な知見に基づいて考案された学習モデルである。適応共鳴理論には、この 1987 年に発表された 2 値入力に対して動作する ART-1 の他に、アナログ信号を処理できるように一般化された ART-2[2]、及び、より生体の動作に近付くように改善された ART-3[3] の 3 つのモデルが存在する。本研究では、これらの基本である ART-1 を利用しているので、以後、ART-1 を ART と記述することにする。

3.1 概要

人間の脳は外界からの情報を過去の記憶 (経験) を元に整理、分類し、新しい概念があれば、それをすでにある記憶が失われたり変更されたりしないように保存することができる。ここに、どのように過去の記憶を変更せず安定性を保ったまま、新しい記憶を保存するのかという、安定性と可塑性ジレンマが生じる。ART はこのジレンマを解決する 1 つの方法である。

ART モデルは、基本的には 2 層の競合学習ネットワークモデルであるが、これに自律的な制御機構を付加することによって自己組織的に適応的なパターン認識を行なう。このモデルには、パターンの認識、比較、探索という 3 つのフェーズがあり[1]、これらのフェーズを経て次のような動作をする。

- 入力パターンと貯えられている原型 (パターン) が似ている時：認識

[1] これらのフェーズは自律的な制御機構によりフェーズの移行をコントロールされる。

- 入力パターンが貯えられているどの原型にも似ていない時：使用されていない任意の出力層のノードで新しいカテゴリを形成
- すべての出力層ノードが使用されている：新しいカテゴリは形成されない

ART における学習には、後で説明される “高速学習” と “低速学習” の 2 つの学習方式があるが、いずれの方式で学習したとしても学習が収束した段階で、既知の入力パターンを回路網に与えると、探索フェーズに移行することなく直ちに認識される (direct access)。

以下で、ART モデルのアーキテクチャ、動作、そして特徴と問題点について述べる。

3.2 ART アーキテクチャ

ART の基本的な構造を図 3.1に示す。入力素子群 (以後本論文では**入力フィールド** と呼ぶ[2]。) と、出力層 (以後本論文では**カテゴリ表現層**と呼ぶ[3]。) は、完全に結合している。入力フィールドには、“入力” と “比較” の 2 つのモードがあり、これは本来、制御信号によって切替えられるのだが、データの流れから自明であるので本論文では省略する。また、ある時点でカテゴリ表現層のノードを有効にするか無効にするかの切替を行なうのが、制御信号 $\boldsymbol{C}$ である。ρ (vigilance parameter) は、入力パターンが記憶しているカテゴリに属するかどうかの指標を与えるものである。また、$\boldsymbol{W}$ 、$\boldsymbol{D}$ は、それぞれ入力フィールドからカテゴリ表現層への経路 (bottom-up pathway) と、カテゴリ表現層から入力フィールドへの経路 (top-down pathway) の結合荷重ベクトルである。

3.3 ART の動作

ART の動作を、初期化、認識、比較 (訓練を含む)、探索フェーズの 4 つに分けて簡単に説明していく[4]。

3.3.1 初期化フェーズ

実際にパターンを入力する前に、各パラメータの初期化を行なう必要がある (入力フィールドのノード数を N、カテゴリ表現層のノード数を M とする)。

[2] オリジナルの論文では feature representation field, F_1 と呼ばれている。

[3] この層は原文では category representation field, F_2 と呼ばれている。

[4] 詳細は [1][18][19] を参照のこと

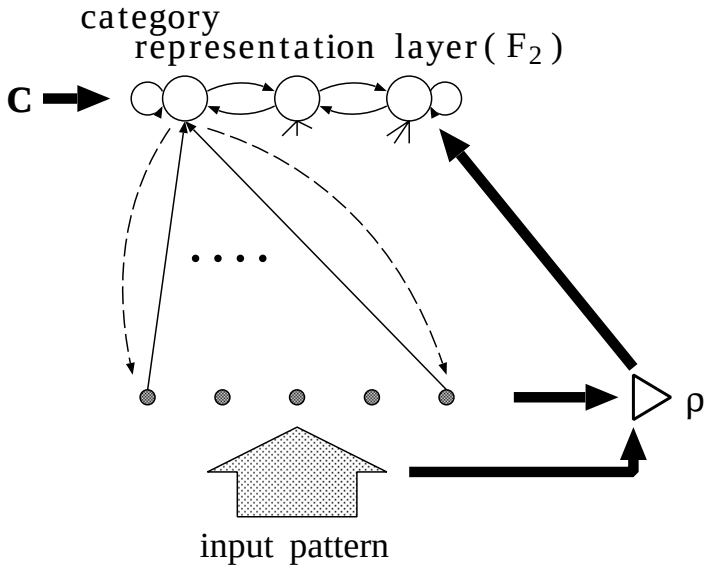


図 3.1: **ART アーキテクチャ**　図中の破線は top-down 経路を、実線は bottom-up 経路を示している。

結合荷重

bottom-up 経路における結合荷重の初期値 $w_{ij}(0)$ (但し、$0 \leq i < N,\ 0 \leq j < M$) の与え方は特に重要であり、すべて小さな同じ値に初期化しなければならない。論文 [1] によれば、この値は式 (3.1) の範囲にする必要がある。このようにすべて同じ値にすることで、どのカテゴリ表現層の素子にも均等に勝者となる機会を与えることができる。式 (3.1) の上限は direct access を保証するための制限である。

$$0 < w_{ij}(0) < \frac{L}{L-1+N} \qquad \text{但し、} L > 1 \tag{3.1}$$

ここで L は典型的には $L = 2$ とすることが多い。

top-down 経路における結合荷重 $\boldsymbol{D}$ は $\{0, 1\}$ の 2 値をとり、学習したパターンの保存のために使用される。初期値 $d_{ij}(0)$ は、

$$d_{ij}(0) = 1 \tag{3.2}$$

とし、新しいカテゴリが生成された時には常に入力パターンとマッチするようにする。

カテゴリ表現層

入力パターンが与えられていないため、どの素子も発火していない、即ち、勝者がいない状態にしておく。したがって、カテゴリ表現層の素子群$\boldsymbol{U}$の初期値は、

$$\boldsymbol{U} = 0 \tag{3.3}$$

制御線

制御信号$\boldsymbol{C}$のとる状態は0か1の2つで、入力が与えられた時は全て1になるが、その入力が記憶パターンとマッチしなかった時は、その時点の勝者ノードが次の探索で再び勝者にならないようにするため、0になり、そのノードを無効とする。これは、新しい入力が与えられるまで固定される。

$$\boldsymbol{C} = 0 \tag{3.4}$$

vigilance parameter

vigilance parameter ρ は、

$$0 \leq \rho \leq 1 \tag{3.5}$$

の範囲で任意の値をとり、入力パターンと記憶パターンがどの程度似ているかという許容範囲を指定する。

ρ を1に近い大きな値にすれば入力パターンは細かく分類され、逆に小さな値にすれば大雑把に分類されカテゴリはあまり生成されない。

3.3.2 認識フェーズ

入力パターン（入力ベクトル）$\boldsymbol{X}$を入力フィールドに与えると、それがbottom-up経路を通ってカテゴリ表現層へ伝達される。この時カテゴリ表現層のノードの出力ベクトル$\boldsymbol{U}$は、

$$\boldsymbol{U} = \boldsymbol{W} \cdot \boldsymbol{X} \tag{3.6}$$

である。この中で最も大きい値を持ったものが勝者ノード u_{j^*}となり、$u_{j^*} = 1$にセットされ、$j \neq j^*$なノード u_jは0になる。

$$u_{j^*} = \max_j[u_j] \quad (0 \leq j < M) \tag{3.7}$$

この状態の出力ベクトル$\boldsymbol{U}^*$から

$$\boldsymbol{X}^* = \boldsymbol{D} \cdot \boldsymbol{U}^* \tag{3.8}$$

の式により、入力パターン$\boldsymbol{X}$に対する想起パターン$\boldsymbol{X}^*$が生成される。

3.3.3 比較フェーズ

認識フェーズで想起されたパターン$\boldsymbol{X}^*$と入力パターン$\boldsymbol{X}$を次式によって比較を行なう。

$$\frac{|\boldsymbol{X}^*|}{|\boldsymbol{X}|} \geq \rho \tag{3.9}$$

もし式(3.9)が真であれば正しく分類されたということになり分類動作が終了するので、両経路の結合荷重を次式にしたがって更新する。

$$d_{ij^*}(t+1) = d_{ij^*}(t)x_i \tag{3.10}$$

$$w_{ij^*}(t+1) = \frac{Lx_i}{L-1+\sum_{k=0}^{N-1} d_{kj^*}(t)x_k} \tag{3.11}$$

マッチしない、つまり入力パターンがその勝者ノードの示すカテゴリに属していない、と判定されると、リセット信号を発行し、探索フェーズへ移行する。

3.3.4 探索フェーズ

比較フェーズでのリセット信号を受けて、現在の勝者ノードの値を0にし、探索フェーズを抜けるまで勝者にならないようにする。結局、以後このノードは新しい入力が与えられるまで無効で固定される。

それから再び入力パターンをカテゴリ表現層へ送り、新しい勝者ノードを選びマッチングが行なわれる。この過程は、マッチするノードが見つかるまで続けられる。もし、そのようなノードがなかった場合、その入力パターンは新しいカテゴリに属するものと判断され、カテゴリ表現層の新しい(使用されていない)ノードを割り当てる。そのようなノードがない時は、学習は行なわれない。

3.4 ART の特徴と問題点

ART の持つ特徴の主なものについて簡単にまとめると下のようになる。

- 安定性対可塑性 (stability-plasticity) のジレンマの克服
- 全記憶容量 (カテゴリ表現層のノード数) を効果的に利用可能
- 高速学習の場合、一度の提示で新規パターンの学習を終了
- ローカルミニマム等の収束性の問題がない

このように他のニューラルネットワークモデルにはないような優れた特徴を持ってはいるが問題がないわけではない。特に 1 番目に挙げた “安定性対可塑性のジレンマの克服” は難しい問題含んでいる。例えば、(ρ の設定にも依るが) ある入力パターンを与え、次にそれを少し変化させたものを提示し、さらにそれと少し違ったパターンを与え、ということを何回か続けると、ART の保持している記憶パターンの中からは完全に元の入力パターンは消えてしまい、全く別のカテゴリとして元の入力パターンを分類してしまう。このようなことが起こり得るのは、ART が記憶パターンを、現在の入力に対して最適なマッチングが起こるように “適応的に” 変化させる、ということに起因する。とはいえ、このような動作は記憶を壊しているわけではなく、あくまでも入力 (環境) に適応しているためであるから、一概に欠点といえるものではない。

また ART はジレンマを克服するため、通常のニューラルネットワークモデルと違い、2 種類の結合経路、入力がどのカテゴリに属するかを決めるための経路 (bottom-up pathway) と記憶を保持している経路 (top-down pathway) を持っている。つまり、1 層のモデルにも関わらず 2 層モデルの複雑さを持っている。

他によく言われることに、「入力のちょっとした変動に弱い」という問題がある。これは、マッチングの基準を与える vigilance 変数の設定に関する問題で、vigilance 変数の設定の仕方の如何によっては、入力のちょっとした変動が新しいカテゴリを形成させてしまう、ということである。さらに、カテゴリ表現層では競合学習を用いており勝者がカテゴリを保持しているため、データのコーディングが非分散的であり、耐故障性に問題がある。

以下に、ART の主な問題点を簡単にまとめておく。

- 入力の変動に弱い
- データのコーディングが非分散的
- 耐故障性の能力に欠ける

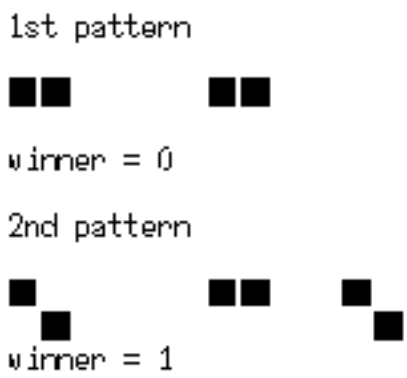


図 3.2: **分離可能なパターンの例**　最左列のパターンが入力であり、右側のパターンが$\rho = 1$ の元で記憶されたパターンである。1 回目の提示で完全に分類されている。

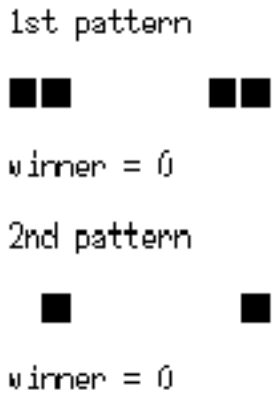


図 3.3: **分離不可能なパターンの例**　図 3.2 と同様に$\rho = 1$ での結果であるが、1 回目の提示で分類されず 1 つ目の入力パターンの記憶を破壊している。

3.4.1　完全学習とパターン分離問題

ρ が十分 1 に近くない時、入力パターン群を数回繰返し提示すると、カテゴリの生成が終了し direct access するようになる。これはつまり入力パターン群を完全に学習したということである。ART は入力パターンの提示順序によって生成されるカテゴリ数が変わってくるが、完全に学習した段階で生成されるカテゴリの数、及びその記憶パターンは提示順序によらず一定であり、同じである。

次に、$\rho = 1$ の場合について考えてみると、通常、任意の入力パターン群は 1 回の提示で完全に学習される (図 3.2) が、1 回の提示で分離できないパターンが存在する。実はこれらのパターンは、任意の値のρ で分離できないことが分かっている (付録 A を参照のこと)。この分離 “できない” という表現は適切ではなく、“安定性が維持されない” のである。分離できないパターンの提示した時のシミュレーション結果を図 3.3に示す。

図 3.3から明らかであるが、安定性が維持できないパターンとは、任意のパターンとその subset のパターンであり、提示順序も関係がある。したがって、以下の条件を満たす入力パターン $\boldsymbol{X}_a$ と $\boldsymbol{X}_b$ において、$\boldsymbol{X}_a$、$\boldsymbol{X}_b$ の順序で提示すると分離することができない。

$$\boldsymbol{X}_a \supset \boldsymbol{X}_b \tag{3.12}$$

$$|\boldsymbol{X}_a| > |\boldsymbol{X}_b| \tag{3.13}$$

この問題を **subset 問題**と呼ぶことにする。とはいえ、これらの subset 問題に抵触するようなパターン群は 1 回の提示で分類できないということにしか過ぎない。なぜなら、2 回目の提示の時は勝者素子が記憶しているパターンが $\boldsymbol{X}_b$ であるため、式 (3.13) が成立しなくなるからである。言い替えれば、提示順序が変わってしまうということである。

一般的な他のニューラルネットワークモデルでは、提示回数等に依存するようなこの種の問題はあまりないが、ART では、そのパターンの分類規則によってこのような問題が生じてしまう。

第 4 章

階層型混成ニューラルネットワークモデル

節 2.4で述べたように、競合学習モデルでは線形分離不可能なパターン群を複数のカテゴリに分類することはできても、どのカテゴリがどのクラスに属するのかまでは分からない。例えば、有名な 2 入力 XOR 問題について考えてみると、競合学習では 4 つのパターン $(0,0)$、$(0,1)$、$(1,0)$、$(1,1)$ を、それぞれ 1 つの点を含む 4 つのカテゴリに分類するが、分類を行なわせる側からしてみれば $(0,1)$、$(1,0)$ という組と $(0,0)$、$(1,1)$ という組の 2 グループに分けて欲しいわけである。

このようなカテゴリの統合を行なうには競合学習だけでは難があるため、教師信号を用いて適切なクラス分けを行なう学習が必要になる。本論文では、ART を用いたカテゴリの統合を行なうことができ、かつ追記学習が可能なニューラルネットワークモデル、適応型カテゴリ統合回路網（$\mathrm{AC_TUN}$: Adaptive Category Unifying Network）を提案する。

4.1 概要

競合学習のみで学習ではカテゴリの統合ができない。ART も例外ではなく、任意のパターン群を任意の粗さで分類することができるだけである。しかし、ART のカテゴリ表現層の上に、得られたカテゴリが適切なクラスに属するよう教師信号によって学習させることのできる層を追加することで、この問題を解決することができる。適応型カテゴリ統合回路網 $\mathrm{AC_TUN}$ は、この考え方にしたがって構築されたものである。

$\mathrm{AC_TUN}$ の構造はシンプルで、ART によって分類され生成されたカテゴリを入力として、単純パーセプトロン (Simple Perceptron) によってカテゴリの統合を行なう。カテゴリ表現層は競合学習層であるため、単純パーセプトロンの入力は、2 進入力（binary input）

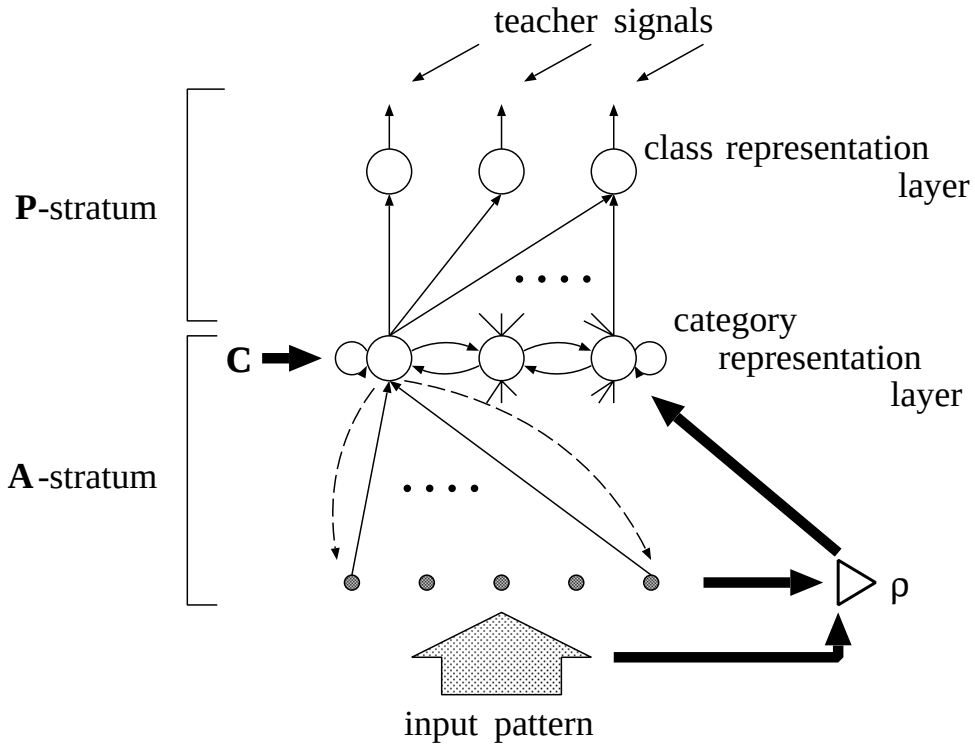


図 4.1: **ACTUN アーキテクチャ**　A 層で探索が終了し真の勝者が決定した時点で、その時のカテゴリ表現層の出力が P 層の入力となる。P 層では教師信号を用いてカテゴリの統合を行なう。

となる[1]。このことにより、下層の ART と同様にパーセプトロンの層においても、新規パターンの追記学習によって記憶パターンが破壊されることなく安定性を保つことができる。また、単純ヘッブ則を用いて学習を行なうため、学習の収束性に関する問題もない。

4.2　アーキテクチャ

図 4.1 に ACTUN のアーキテクチャを示す。ACTUN は、ART によって構成される **A 層**（A-stratum）とその出力層であるカテゴリ表現層を入力層とするパーセプトロンによって構成される **P 層**（P-stratum）からなり、カテゴリ表現層と P 層の出力素子群であるクラス表現層とは完全に結合している。したがって、パーセプトロンを構成する P 層は入力として、A 層の（学習した記憶パターンではなく）カテゴリ表現層の出力を受け取る。stratum は、本論文では、各入力素子群を意味する layer と区別し、layer とそこへの結

[1]立っているビットの数（つまり 1 の数）は、常に 1 である。

合経路等を含む一連の構造のことを指すために使用する。クラス表現層は単純パーセプトロンの出力であるから 2 進値 (binary output) であり、教師信号も同じである。

クラス表現層の素子数は、カテゴリ表現層の素子数と同じである。これは、クラス表現層ではカテゴリ表現層で生成されたカテゴリに対して教師信号を発するためである。

4.3 動作

以下で ACTUN の動作をカテゴリ分類、カテゴリ統合フェーズの 2 つに分けて層毎に説明する。

4.3.1 カテゴリ分類フェーズ

カテゴリ分類フェーズでは、通常の ART による学習を行なう（第 3章参照)。

このフェーズで最も重要なことは、ノイズに左右されない程度にできるだけ細かく入力パターンを分類することである。したがって、通常、vigilance parameter ρ の範囲は $0.8 \leq \rho < 1$ である。このように細かく分類する必要があるのは、分類が粗いと教師信号を用いたカテゴリの統合を行なう際に問題が生じるからである。つまり、 ρ の値が小さめに設定されていると、入力パターンがどのように、或はどのようなカテゴリに分類されるのかが事前に分からないため、カテゴリ統合の際の教師信号を適切に与えることができない。したがって、ρ は生成されるカテゴリについてある程度目安がつくくらいの値に設定する必要がある。$\rho = 1$ に関しては、ちょっとした入力の変動に対しても非常に敏感になるので必ずしも適切であるとはいえない。

カテゴリの探索中は、カテゴリ表現層に勝者が存在していたとしてもその時点での $\boldsymbol{U}$ は意味を持っていないので、P 層へ送出されない。したがって、探索フェーズ終了後、或は direct access により勝者が決定した後で、その時のカテゴリ表現層のベクトル $\boldsymbol{U}^*$ がカテゴリ統合層へ送出され、同時に top-down 経路の結合荷重 $\boldsymbol{D}$ 、bottom-up 経路の結合荷重 $\boldsymbol{W}$ を更新する。

4.3.2 カテゴリ統合フェーズ

P 層の初期化

カテゴリ統合フェーズでの実際の動作について説明する前に、ACTUN 動作開始時における P 層の初期化、即ち初期結合荷重 $\boldsymbol{P}(0)$ について述べておく。

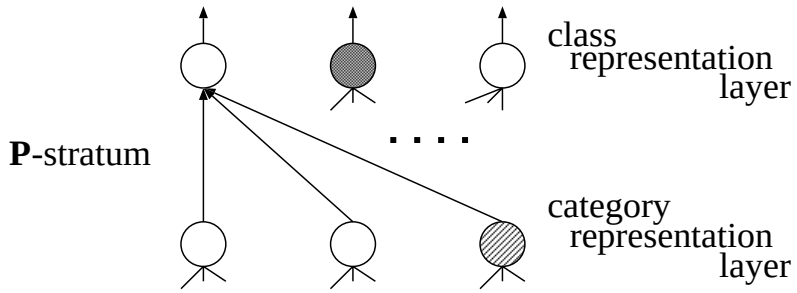


図 4.2: **クラスの表現**　クラス表現層の薄く色のついた素子は、カテゴリ表現層の斜線のついた素子のカテゴリが属すべきクラスを示しており、教師信号により与えられる。

カテゴリ表現層の素子 u_j とカテゴリ統合層の素子 p_k における初期結合荷重 $p_{jk}(0)$ は、

$$-0.5 \leq p_{jk}(0) \leq +0.5 \tag{4.1}$$

の範囲の乱数で与える。

教師信号

教師信号は、各入力パターンがどのクラスに属すのか、或はどのパターンとどのパターンが同じクラスなのかを教えるものである。

ART がカテゴリ表現層の素子がその入力パターンのカテゴリを示しているのと同様に、P 層のクラス表現層の素子は、その時の入力パターンのカテゴリが属しているクラスを示す（図 4.2）。教師信号 t_k は 2 進値をとり、かつ $|\boldsymbol{T}| = 1$ である。

カテゴリ統合

カテゴリ表現層からのカテゴリ表現ベクトル $\boldsymbol{U}^*$ は、結合荷重 $\boldsymbol{P}$ を持つ P 層の経路を通ってカテゴリ統合層へ送出される。この時、カテゴリ統合層の出力 $\boldsymbol{O}$ は、

$$\boldsymbol{O} = f(\boldsymbol{P} \cdot \boldsymbol{U}^*) \tag{4.2}$$

或は、

$$o_k = f\left(\sum_j p_{jk} u_j\right) \qquad (\text{但し、} 0 \leq k < M) \tag{4.3}$$

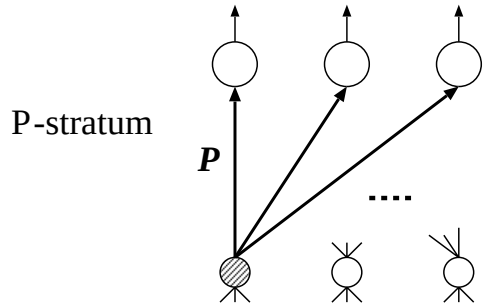


図 4.3: **結合荷重の更新**　斜線のついた勝者素子からの経路のみが更新の対象であり、他のカテゴリ表現層の素子は全く関与しない。

と書くことができる。また、カテゴリ表現層は競合学習を行なう層であることから、勝者素子 u_{j^*}のみが 1（$|\boldsymbol{U}| = 1$）であることを利用すれば、

$$o_k = f\left(p_{j^*k}\right) \tag{4.4}$$

となる。ここで、$f(x)$ なる関数は式 (2.2) で示されるヘビサイド関数である。

得られた出力ベクトル $\boldsymbol{O}$ は、その時の入力パターン $\boldsymbol{I}$ に対応する教師信号ベクトル $\boldsymbol{T}$ と同じであることが望ましい。したがって、節 2.2.1で説明した学習規則、

$$\Delta p_{jk} = \eta\left(t_k - o_k\right) u_j \tag{4.5}$$

によって $\boldsymbol{O} = \boldsymbol{T}$ となるように結合荷重を更新し学習が行なわれる。ここで、式 (4.5) は、$\boldsymbol{U}$ の性質から、

$$\Delta p_{j^*k} = \eta\left(t_k - o_k\right) u_{j^*} \qquad (\text{for all } k) \tag{4.6}$$

と書き換えることができる。この式の示すところは、結合荷重の更新は、カテゴリ表現層の勝者素子とクラス表現層との間の経路でのみ行なわれるということである（図 4.3）。

P 層の学習収束判定

次に、P 層の学習の収束について述べる。ここでは分離可能な場合についてのみ考えることにする。P 層での学習に失敗する場合については、この後に述べられる。

P 層の学習は、教師信号と P 層の出力との差がなくなったところで終了する。つまり、学習誤差を err とすると、

$$err = \eta\left(t_k - o_k\right) = \pm\eta \tag{4.7}$$

において、$err = 0$ となった時点でそのパターンにおける学習が終了する。パーセプトロンは有限回の回数で解を得ることができることが証明されているが、P 層ではどうだろうか。

P 層の結合経路における初期結合荷重 $p_{jk}(0)$ は式 (4.1) によって定められており、クラス表現層素子の閾値は式 (2.2) で示される通り 0 である。これらの事実から、誤った出力を出したクラス表現層素子につながっている経路の結合荷重の符合が反転すれば学習が終了することになる。したがって、結合荷重の更新回数をτとすれば、学習終了時までの総変更量Δ は、

$$|\Delta| = \tau\,\eta \tag{4.8}$$

である。ここで、結合荷重を反転させるために必要な総変更量は、最悪の場合でも $|\Delta| = 0.5$ で反転可能であるから、$|\Delta| \leq 0.5$ である。したがって、収束に必要な更新回数τ は、

$$\tau = \frac{|\Delta|}{\eta} \leq \left\lceil \frac{0.5}{\eta} \right\rceil \tag{4.9}$$

となる。よって、P 層での学習収束はτに依存する[2]。

これらの事実から、η の値は大きめに設定した方が良いことが分かる。P 層では勾配降下則等の局所解を導くような学習則を用いているわけではないので、あえて小さい値にする必要はない。

4.4 追記学習

新規パターンと学習済みのパターンとで取り得るカテゴリとクラスの関係は、

1. カテゴリが同じで、かつクラスも同じ。
2. カテゴリが同じで、かつクラスが異なる。
3. カテゴリが異なり、かつクラスは同じ。
4. カテゴリが異なり、かつクラスも異なる。

の 4 ケースである。以下でこの 4 種の状況における結合荷重の変化について述べる。

[2]入力の大きさではない。入力の大きさに依存するのは、収束に要する時間である。

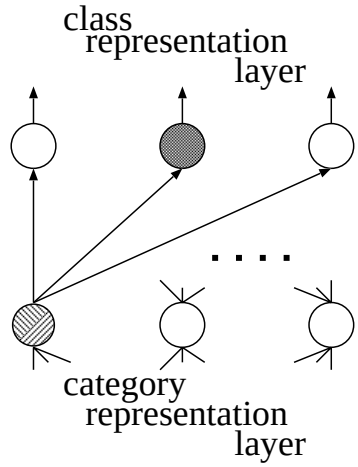


図 4.4: **カテゴリ、クラスとも同じ場合**　あるパターンの属すカテゴリ (右上がりの斜線) と他のパターンが属すカテゴリ (左上がりの斜線) が同じで、両者とも同じクラスに属す場合、使用する結合経路は同じである。

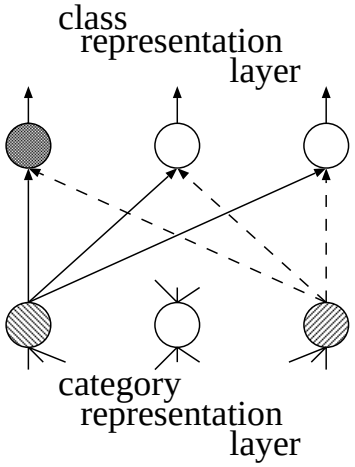


図 4.5: **クラスが同じ場合**　あるパターンの属すカテゴリ (右上がりの斜線) と他のパターンが属すカテゴリ (左上がりの斜線) が、それぞれ異なるクラスに属す場合、使用する結合経路は異なる。

ケース 1 の場合

異なる入力パターンが同じカテゴリに分類され、かつクラスも同じ場合、P 層での学習に関係する経路を図 4.4 に示す。

この場合、結合荷重の更新に関与するカテゴリ表現層からクラス表現層への経路は、1 種類しかない。このことは、P 層から見れば以前の入力と同じ入力信号が入ってきたということと同じことになる。したがって、結合荷重は変更されない。

ケース 3 の場合

関係する結合経路を図 4.5 に示す。2 種類の入力、A 及び B の属すカテゴリが、u_{j^A}及び u_{j^B}であるとすると、$j^A \neq j^B$であるから、学習規則、式 (4.6) によって変更される結合経路は異なる。

ケース 4 の場合

4 つ目のケースで関与する結合経路を図 4.6 に示す。

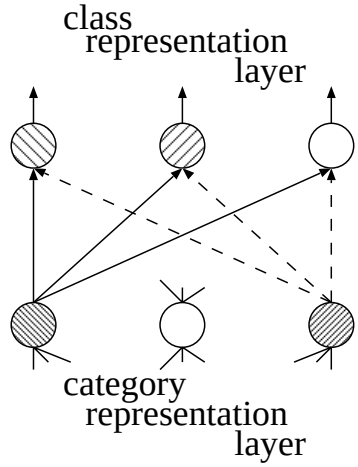


図 4.6: **カテゴリ、クラスとも異なる場合** 異なるカテゴリに分類されていて、かつ異なるクラスに属しているので、図 4.5の場合と同様、使用する経路は異なる。

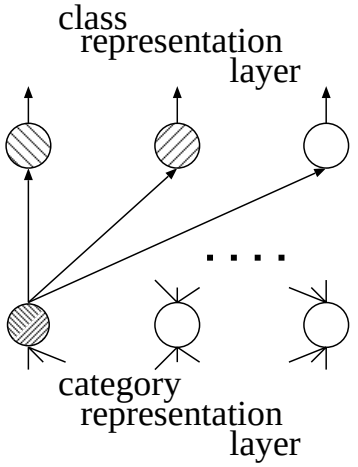


図 4.7: **クラスが異なる場合** 複数のパターンが同じカテゴリに属すのにも関わらず、それぞれが異なるクラスに属す場合、カテゴリ表現層素子から伸びている複数の結合経路の荷重を平等に増減する。

この場合も、ケース 3 の時と同様、属するカテゴリが異なるため、式 (4.6) に示される学習規則によって変更される結合経路は異なり、まったく排他的である。したがって、新規パターンが既存のパターンのクラス表現を破壊することはない。

ケース 2 の場合

最後に、2 番目のケースのような状況について考える。このような状況下において変更される可能性のある結合経路を図 4.7 に示す。

はじめに、どのような時にこのような状況に陥るのかを考えると、大きく分けて 2 つの場合が存在する。

1. 節 3.4.1 で述べた subset 問題に抵触する入力パターンを提示した時。
2. vigilance parameter ρ が十分に大きくなかった時。

1 つ目の場合について考えると、subset 問題に抵触するのは 1 回目の提示の時だけであり、その時はクラス表現が破壊されるが、2 回目以降の提示においては、変更される可能性のある結合経路は図 4.6 に示されるような経路になり、クラス表現が破壊されることはなくなる。

表 4.1: ケース 2 の場合の真理値表

input1	input2	output
0	1	0
0	1	1

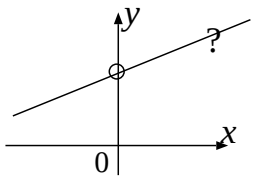


図 4.8: **入力空間**

次に 2 つ目の場合について考えてみる。

まず、ある入力パターンのクラスを表現する素子が k^Cであるとすると、結合荷重の 1 回の変更量は式 (4.6) より、

$$\Delta p_{jk} = \begin{cases} \eta & (\text{ if } \quad j = j^*,\ k = k^C\ ) \\ -\eta & (\text{ if } \quad j = j^*,\ k \neq k^C\ ) \\ 0 & (\text{ otherwise }) \end{cases} \tag{4.10}$$

である。

ここで、教師信号 t_kは、

$$t_k = \begin{cases} 1 & (\text{ if } \quad k = k^C\ ) \\ 0 & (\text{ otherwise }) \end{cases} \tag{4.11}$$

である。学習の収束回数がτ回であったとすれば、カテゴリ表現素子 j^*とクラス表現素子 k^Cとの間の経路の結合荷重の変更量は、

$$\Delta p_{j^*k^C} = \tau\,\eta \tag{4.12}$$

で、結合が強化されるように更新される。j^*、k^C間以外の経路の変更量は、

$$\Delta p_{j^*k} = \begin{cases} -\tau\,\eta & (\text{ if } \quad t_k \neq o_k\ ) \\ 0 & (\text{ otherwise }) \end{cases} \tag{4.13}$$

であり、教師信号と同じでない出力を出す経路の結合荷重は弱められる。

次に同じカテゴリに属す別のパターンを入力した時、このパターンのクラスが $k^{C'}$に属すものとすると、$k^{C'} \neq k^C$であるから、このパターンの学習においては、式 (4.13) で発火を抑制するように変更された経路を逆に式 (4.12) を用いて正の方向に強化することが行なわれる。この結果、例え 2 つの入力パターンを繰返し提示したとしてもクラス表現素子 k^C、$k^{C'}$とカテゴリ表現素子 j^*との間の特定の 2 本の経路のみが強化されたり、抑制されたりを繰り返すだけで、収束せず振動してしまう (参考：図 B.2)。

この状況を入力を 2 次元と仮定して考えてみると[3]、P 層のパーセプトロンが取るべき動作は表 4.1 のような真理値表に表される。これを幾何学的に示したのが図 4.8 である。この図から明らかなように、このケース 2 の場合は、分類されるべきカテゴリが同じ位置にあり、いわゆる**一般の位置**にあるのではないため、パーセプトロンはクラス分けするための直線を定めることができない。したがって、先に述べたように学習が収束しない。

上で述べたように、1.、3.、4. に示したような場合は、式 (4.6) から結合荷重 $\boldsymbol{P}$ が変更されることがないため、クラス表現を破壊するような学習は起こらない。したがって、少なくとも上述の 3 つの場合においては、過去の記憶を破壊しない安定的な追記が可能であることが分かる。

2. のケースだけは特別でクラス表現の破壊を引き起こすが、このようなパターン自体は原因が明らかであり、記憶の修復を行なうことが可能である。subset 問題に抵触するパターンを入力した場合については、関連するパターンだけを再度提示すれば解決する問題であるし、そもそも ART での分類失敗はρの設定ミスである。本来、AC$_{\mathrm{T}}$UN では、節 4.3.1で述べたように異なる入力パターンが同じカテゴリに分類されることは望ましいことではない。とはいえ、AC$_{\mathrm{T}}$UN(ART もそうであるが) は他のニューラルネットワークモデルと異なり、破壊されたパターンがどの入力パターンのものであるかが分かるため、そのパターンのみを再度提示するだけで、正しいカテゴリ、クラスに分類することができるようになるという決定的な利点がある。また、ART の学習過程として高速学習法を用いているため、これらの学習、更にはパターンマッチング、カテゴリの生成等は非常に高速に行なわれる。これらのことについての考察を以下で行なう。

4.5 考察

本節では、本論文で提案したニューラルネットワークモデル AC$_{\mathrm{T}}$UN について、“学習速度”、“線形分離問題”、“追記学習と subset 問題”、“耐ノイズ性” の 4 つのテーマについて検討する。

[3]P 層の入力で意味のあるビットは、1 ビットのみであるのでこの仮定に問題はない。

表 4.2: **AND の真理値表**

in1	in2	in3	out
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0

表 4.3: **XOR の真理値表**

in1	in2	in3	out
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0

4.5.1 学習速度

本節では、AC_TUN の学習速度と言う観点から、バックプロパゲーション法との比較を行なう。

バックプロパゲーション法は、学習規則、つまり結合荷重の変更方法として勾配降下則を用いているため、一般的に学習の収束に時間がかかる。また、初期値の設定に失敗すると学習に失敗することがある。AC_TUN では、初期値の設定についての心配をする必要はないし、加えて P 層の学習においては局所的最少解に捕らわれることもない。

表 4.2に示される AND 関数と、表 4.3の XOR 関数のシミュレーションを行ないその収束速度について調べた結果を図 4.9、図 4.10に示す。学習率は$\eta = 0.2$、vigilance parameter は $\rho = 0.9$ とした。`Ptns` が 7 のところからは 2 回目の提示が始まっている（図の詳しい見方については付録を参照のこと）。

図から明らかなように、提示 1 回目には 2、3 回の結合荷重の更新で学習が収束しており、2 回目の提示からは学習は必要としておらず、完全にクラス分けが完了している[4]。このような分類をバックプロパゲーションで行なったとすれば、初期値に左右されるが、学習の収束回数で数十回以上、実時間で数秒から数時間、或は数日間かかるだろう。また、学習回数が入力パターンやそのクラス分けに依存しないことが明らかである。

4.5.2 線形分離問題

線形分離問題として、XOR 問題を取り上げよう。シミュレーション結果は前節で述べたように図 4.10で示される。真理値表 4.3で示されるように、A 層でそれぞれのパターンが完全に分類されている。また、P 層での学習も数回で終了し目的のクラスにクラス分け

[4]この程度の規模の入力の学習であれば、一瞬で学習は終了する。

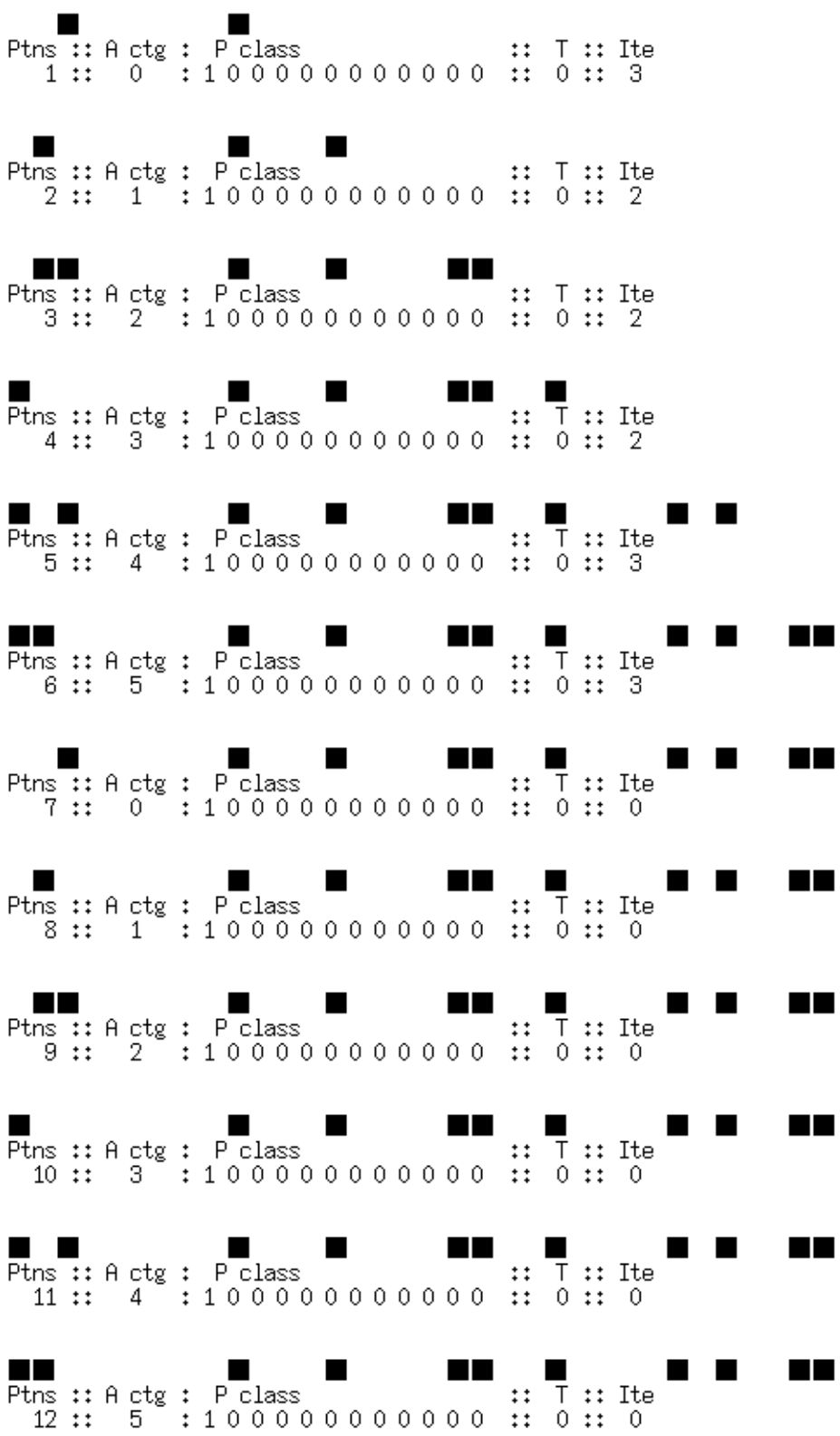


図 4.9: **AND のシミュレーション結果**

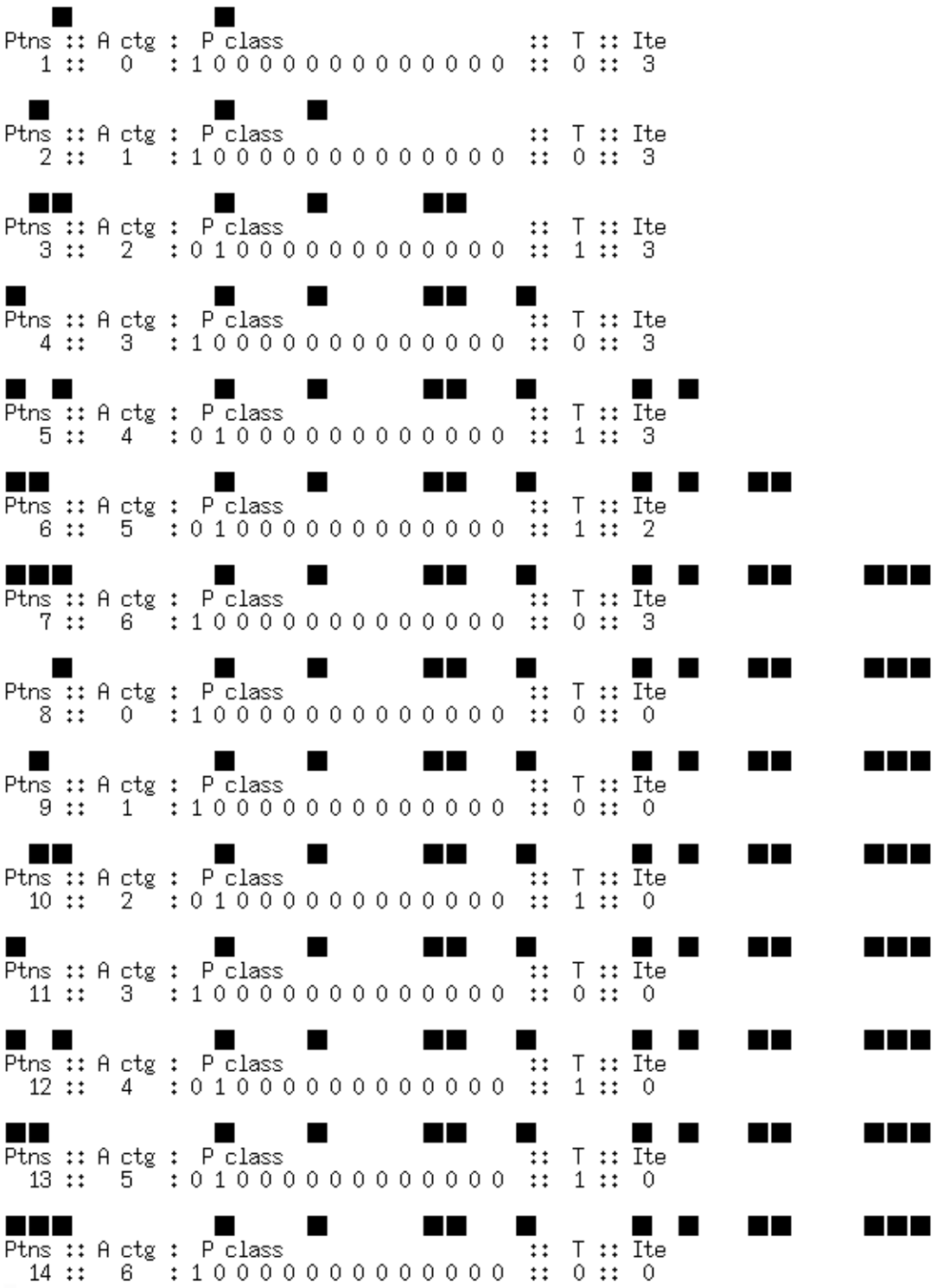


図 4.10: **XOR のシミュレーション結果**

されている。2 回目の提示においては、P 層では結合荷重の更新は行なわれておらず完全にクラス分けが完成していることが分かる。

4.5.3 追記学習と subset 問題

節 4.3.2で述べたように ACTUN では A 層での分類に失敗しない限り追記学習で問題が生じるようなことはない。この節では、subset 問題との関係から考察する。

まず、図 4.11に 3 つのパターンがそれぞれの subset であるような入力について 4 回、順に提示した場合のシミュレーション結果を示す。1 回目の提示では、3 番目のパターンの学習が完了している[5]。同様に提示 2 回目で 2 番目のパターンの学習が終了し、3 回目の提示の時にすべてのパターンが分類され、クラス分けされていることが分かる。このことは提示 4 回目の結果から明らかである。

ここで話を追記学習に戻すと、このシミュレーション結果 (図 4.11) から実にうまく追記学習が遂行されていることが分かる。つまり提示 1 回目では、ネットワークは複雑な概念からより抽象化された概念へと記憶の抽象化を行なうとともに、そのクラスを変更する。提示回数が増えるとともにより複雑な概念をネットワークに形成し、またその属すクラスの修正を行なっている，と考えることができる。

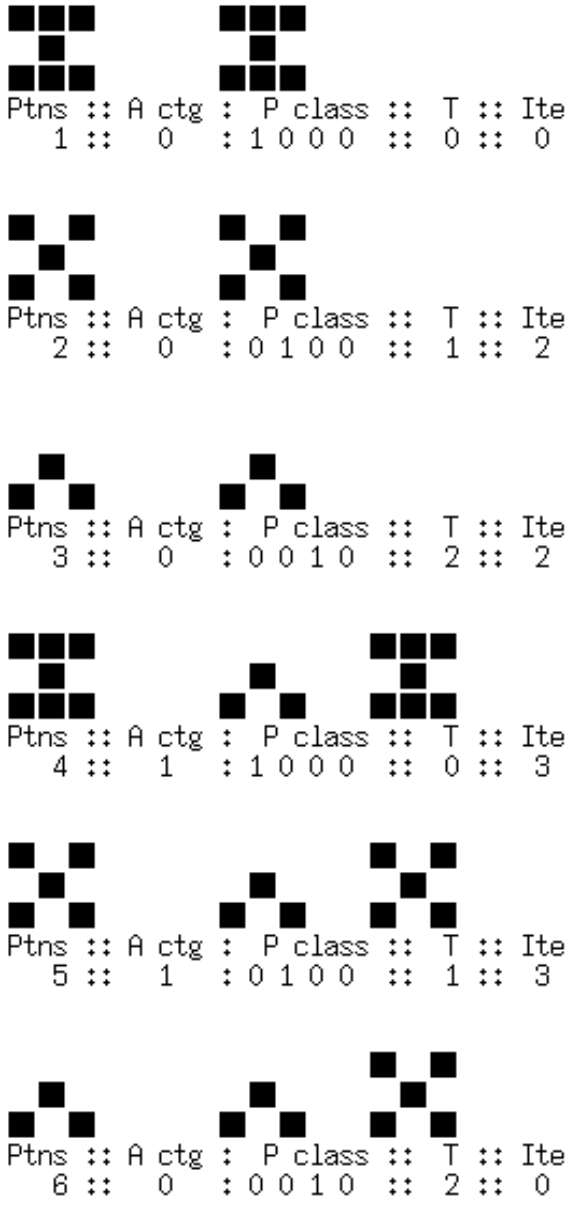
図 4.12は、図 4.11で用いた 3 つのパターンを逆の順序で提示した場合の結果である。この場合は 1 回目の提示ですべてのパターンを分類し、かつカテゴリの統合も完了している。

以上の結果から、すべての入力パターンが始めに分かっているのであれば、subset 問題に対処するのは簡単で、立っているビット数の少ないノッペリとしたパターンから提示していけばよいことが分かる。とはいえ、予め入力が分かっているような状況は ACTUN の運用から考えるとあまり一般的ではない。しかし、干渉を起こすパターンは事後にでも分かるため、いつでも再学習が可能である。

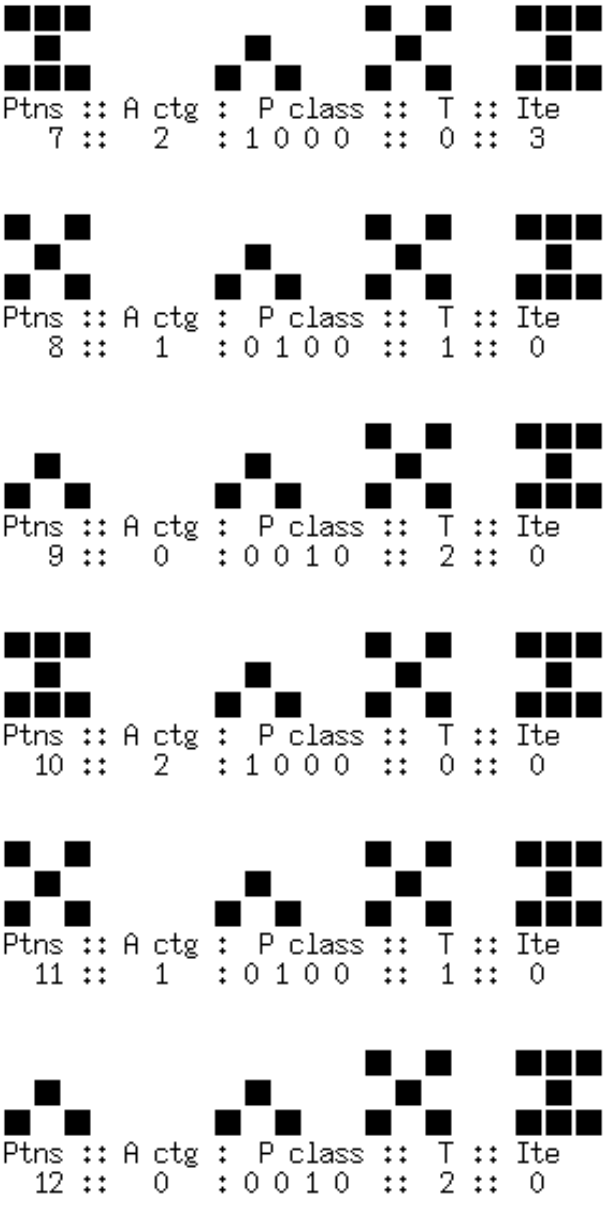
4.5.4 耐ノイズ性

ART は入力の変動に対して弱いといわれるが、ACTUN ではカテゴリの統合という動作により覆い隠すことができる。最も簡単な対処法は、$\rho = 1$ としてノイズを含めたすべてのパターンを記憶し、それを P 層の機能により統合することでパターン認識を行なうことである。この場合、カテゴリ表現層の素子数が非常に多くなるが、それは入力ベクトル

[5]提示 2 回目の結果で、`Ite` を見れば明らかである。



(a) 提示 1、2 回目



(b) 提示 3、4 回目

図 4.11: **subset 問題におけるシミュレーション結果**

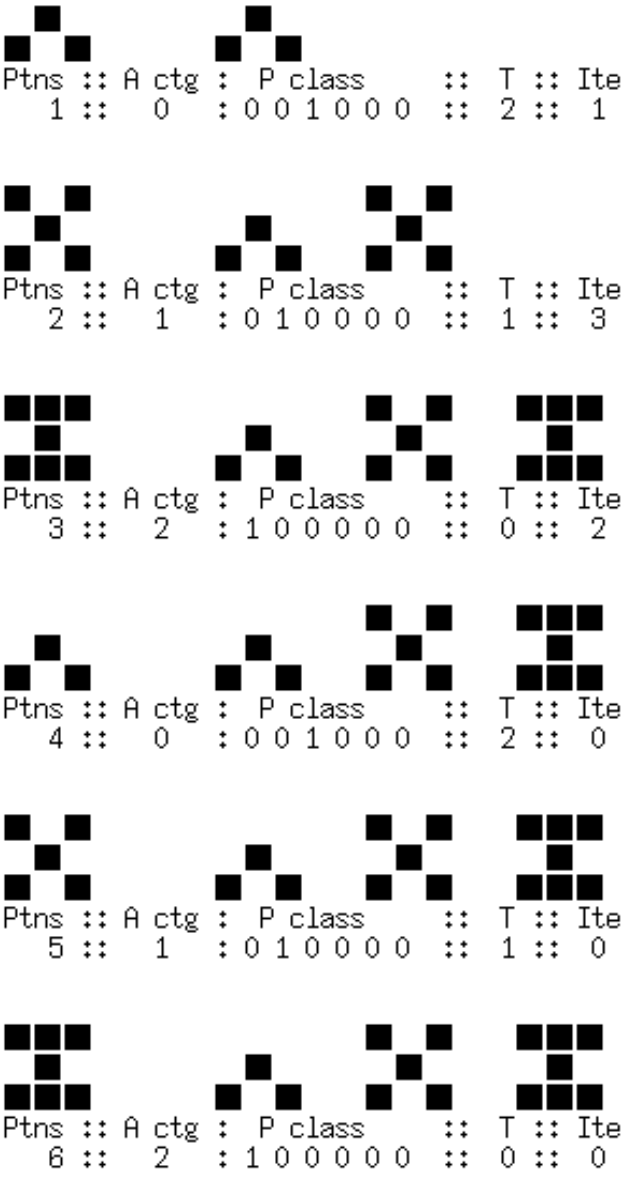


図 4.12: **図 4.11の逆順で提示した結果**

の次元数に制限されるし、ACTUN の各層は並列性が高いので時間的な制約はそれほど大きなものではない。

最後に Counterpropagation Network(CPN) との差異について触れておく。

CPN は、競合学習を行なう Kohonen 層と、その出力を受け教師あり学習を行なう Grossbarg 層との混成ニューラルネットワークモデルである。この CPN と ACTUN との主な違いは、用いるアーキテクチャ、信号の流れの方向性、学習モードである。ACTUN では単なるパターンの分類だけでなくパターンマッチングを行なうことが可能である。CPN には一般的なニューラルネットワークモデルと同じように学習モードと認識モードに分かれており、追記学習はできない。また、CPN はどちらかというと多層パーセプトロンよりの動作（パターン認識、データ圧縮等）を行なうことができる。

4.6 まとめ

4.6.1 特徴

本章で提案した階層型混成ニューラルネットワークモデル、適応型カテゴリ統合回路網（ACTUN）について、特徴という観点からまとめる。

ACTUN は、適応共鳴理論（ART）を用いたカテゴリ生成層、A-stratum と単純パーセプトロンを用いたカテゴリ統合層、P-stratum よりなる、追記学習が可能なニューラルネットワークモデルである。A 層と P 層は直結してはいるが互いに独立であり、それぞれの層内で学習は完結している。A 層では入力パターンからさまざまなカテゴリを生成し、その上位層である P 層で生成されたカテゴリを統合していくつかのクラスに分類する。また、ART と同様に学習モードと想起モードとの区別はなく、常に想起しながら学習を行なうため、無限の入力に対して有効である。これに関連して、学習が高速に行なわれるということも考え併せれば、Real-time システムにおけるパターンマッチングを行なうことも可能である。

vigilance parameter ρ を十分に大きい値に設定していれば、A 層によって適切なカテゴリが生成され、それにより振動を起こすことなく完全に追記学習が可能になる。

4.6.2 問題点

ACTUN で問題となる点は、基本的にはその構成上の問題として生じる。つまり、A 層、P 層で生じるそれぞれの問題が全体としての問題となる。

ART と同様に、「入力のノイズに対する耐性」というρ に関連する問題が存在する。しかし、ACTUN ではカテゴリの統合を行なうことができるため対処することが可能である。

第 5 章

結言

本論文では、まず、一般的なニューラルネットワークにおける学習規則とそのパターンの分類能力についてまとめた。また、適応共鳴理論 (ART) について、vigilance parameter とパターンの分類能力の観点から考察し、分類不可能なパターンについて検証した。

またに、現在曖昧なまま用いられている分類されたパターン群に使用される語、“カテゴリ”、“クラスタ”、“クラス” について以下のような意味を持たせることで区別して使用することを提案した。

- カテゴリ (category)： 基本的、或は最小の分類単位。
- クラスタ (cluster)： いくつかの似ているカテゴリの集合。
- クラス (class)： いくつかの任意のカテゴリの集合。

次に、従来の一般的な多層パーセプトロンでは非常の困難であった追記学習が可能な、また、教師なし学習モデルにはできないカテゴリの統合を行なうことのできるニューラルネットワークモデル、適応型カテゴリ統合回路網 (AC_TUN: Adaptive Category Unifying Network) を提案した。

AC_TUN が ART の持つネットワークの可塑性を保ったまま安定した記憶の保持ができる能力を生かしつつ、単純パーセプトロンによる教師あり学習でカテゴリの統合を行なうことができることを示した。

謝辞

本研究をまとめるにあたっては、日頃御指導を賜わりました日比野靖教授に心より感謝致します。また、ニューラルネットワークの基礎について御指導頂きました丹康雄助教授に深く感謝致します。

最後に、平成 9 年度卒業の日比野研究室ならびに横田研究室の皆様、さらに平成 10 年度現在日比野研究室在籍の皆様のおかげで、楽しい研究生活を送ることができました。ここに深く感謝致します。

参考文献

[1] G.A.Carpenter, S.Grossberg, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine", Computer Vision, Graphics, and Image Processing, Vol.37, 1987, pp.54-115

[2] G.A.Carpenter, S.Grossberg, "ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns", Applied Optics, Vol.26, Dec 1, 1987, pp.4919-4930

[3] G.A.Carpenter, S.Grossberg, "ART3: Hierarchical Search Using Chemical Transmitters in Self-Organizing Pattern Recognition Architectures", Neural Networks, Vol.3, 1990, pp.129-152

[4] G.A.Carpenter, S.Grossberg(阪口 豊訳), "ART-自己組織的神経回路網による適応的パターン認識", 人工ニューラルシステム (bit 臨時増刊) Vol.21, No.11, September, 1989, pp.1393-1408

[5] Richard P.Lippmann "An Introductioin to Computing with Neural Nets", IEEE ASSP Magazine, April, 1987, pp.4-22

[6] 福田敏男, 塩谷成敏ら, "新規パターンを追加学習するための新しいニューロン・モデル", 計測自動制御学会論文集, vol.29, no.3, March 1993, pp.356-364

[7] 荒井正之, 奥田健三, 渡辺博芳、宮道壽一, "追加学習が可能な大規模ニューラルネット Honeycomb ネット III", 信学論 (D-II), vol.J80-D-II, no.7, July 1997, pp.1955-1963

[8] 山内康一郎, 石井直宏, "干渉を受ける記憶パターンの想起と追加学習", 信学技報, NC94-101, 1995

[9] 池田和司 "追加学習の漸近論的解析" 信学論 (D-II), vol.J80-D-II, no.7, July 1997, pp.1913-1918

[10] 山川宏, 増本大器ら, “逐次学習における学習データの選択と追加的出力補正”, 信学技報, NC92-99, 1993

[11] 池原啓介, 中山謙二, “追加学習と記憶内容の自己組織化を目的とした連想記憶モデルの一検討”, 信学技報, NC91-42, 1991

[12] R.Hecht-Nielsen “Counterpropagation networks” Proc. of the Int. Conf. on Neural Networks, vol.2, June, 1987, pp.19-23

[13] R.Hecht-Nielsen “Counterpropagation networks” Applied Optics, Vol.26, Dec 1, 1987, pp.4979-4984

[14] R.Hecht-Nielsen “Applications of counterpropagation networks” Neural Networks,1, 1988, pp.131-139

[15] 藤村喜久朗, 徳高平蔵, 石川眞澄 “変形カウンタープロパゲーションの層間情報に関する考察” 信学技報, NC97-62, 1991, pp.55-62

[16] “変形カウンタープロパゲーションの応用 - 天気予測システム構築へのアプローチ-” 信学技報, NC97-63, 1991, pp.62-69

[17] Oja.E “A Simplified Neuron Model As a Principal Component Analyzer”, Journal of Mathematical Biology, 3, 1982, pp.267-273

[18] R.Beale,Tom Jackson(八名 和夫監訳), “ニューラルコンピューティング入門”, 1993, 海文堂

[19] Philip D.Wasserman(石井直宏, 塚田稔訳) “ニューラル・コンピューティング (理論と実際)” 森北出版株式会社, 1993

[20] J.Hertz, A.Krogh , R.G.Palmer(笹川辰弥, 呉勇訳) “ニューラルコンピュータ (統計物理学からのアプローチ)” トッパン, 1994

[21] R.Hecht-Nielsen “NEUROCOMPUTING” Addison-Wesley Publishing Company, 1990

第 A 章

ART における完全一致分類に関する問題

A.1 証明

論理積を用いたパターンマッチングを行なっているという理由から、ART にはρ の値に関わらず、1 回の提示では分離不可能なパターンが存在する。以下でこのようなパターンの定義と証明を行なう。

Subset Problem:
次のような条件のパターン $\boldsymbol{X}_a$、$\boldsymbol{X}_b$を

$$\boldsymbol{X}_a \supset \boldsymbol{X}_b \tag{A.1}$$

$$|\boldsymbol{X}_a| > |\boldsymbol{X}_b| \tag{A.2}$$

$\boldsymbol{X}_a$、$\boldsymbol{X}_b$の順に提示すると、ART は 1 回目の提示では安定したパターンの保持ができない。

(証明)
学習の開始時、即ち記憶しているパターンが 1 つもない時、簡単のためどの入力に対してもカテゴリ表現層の最初のノード u_0が勝者となるとすると、最初の入力パターン X_aに対して u_0が勝者となる。よって、u_0から伸びている top-down 結合荷重を $\boldsymbol{D}_0$とすれば、想起されたパターンと入力パターン X_aとの比較は、$D_{ij}(0) = 1$ であるから、

$$\frac{|\boldsymbol{D}_0 \wedge \boldsymbol{X}_a|}{|\boldsymbol{X}_a|} = \frac{|\boldsymbol{X}_a|}{|\boldsymbol{X}_a|} = 1 \tag{A.3}$$

であり、ρ の値が何であれ、完全に一致という結果になる。したがって、式 (A.4)、(A.5)

によって結合荷重が更新される。

$$\boldsymbol{D}_0 = \boldsymbol{D}_0 \wedge \boldsymbol{X}_a = \boldsymbol{X}_a \tag{A.4}$$

及び、

$$\boldsymbol{W}_0 = \frac{L}{L-1+|\boldsymbol{D}_0 \wedge \boldsymbol{X}_a|} \times \boldsymbol{D}_0 \wedge \boldsymbol{X}_a \tag{A.5}$$

ここで、$\xi = |\boldsymbol{D}_0 \wedge \boldsymbol{X}_a|$ とすると、

$$0 < \xi < N \tag{A.6}$$

であるので、初期 bottom-up 結合荷重 $\boldsymbol{W}_0(0)$ と式 (A.5) を比較すると、

$$L-1+\xi < L-1+N \tag{A.7}$$

であるから、

$$\boldsymbol{W}_0 > \boldsymbol{W}_0(0) \tag{A.8}$$

となる。

次に、入力パターン $\boldsymbol{X}_b$ を提示すると、式 (A.8) であるから、カテゴリ表現層での競合の結果、勝者素子は u_0 となる。u_0 が記憶しているパターンと入力パターン X_b との比較式は、

$$\frac{|\boldsymbol{D}_0 \wedge \boldsymbol{X}_b|}{|\boldsymbol{X}_b|} \tag{A.9}$$

である。

ここで、式 (A.4) より、

$$|\boldsymbol{D}_0 \wedge \boldsymbol{X}_b| = |\boldsymbol{X}_a \wedge \boldsymbol{X}_b| \tag{A.10}$$

であり、かつ式 (A.1)、(A.2) の関係から、

$$|\boldsymbol{X}_a \wedge \boldsymbol{X}_b| = |\boldsymbol{X}_b| \tag{A.11}$$

となる。したがって、

$$\frac{|\boldsymbol{D}_0 \wedge \boldsymbol{X}_b|}{|\boldsymbol{X}_b|} = \frac{|\boldsymbol{X}_b|}{|\boldsymbol{X}_b|} = 1 \tag{A.12}$$

であり、パターン $\boldsymbol{X}_b$ は勝者 u_0 の記憶パターン ($\boldsymbol{X}_a$) と完全に一致する。この結果 $\boldsymbol{D}_0$ が更新されれば、$\boldsymbol{D}_0 = \boldsymbol{X}_b$ となり、vigilance parameter ρ が 1 に設定されていても、以前の記憶を保持することはできない。 ∎

A.2 完全な学習

一般的なニューラルネットワークモデルがそうであるように、ART もパターンを何回か繰返し与えることによって学習を行なう[1]。

しかし、本質的に ART は、固定した入力に対して学習を行なわせるためのモデルではなく、自然に存在するような時々刻々と変化する環境においても記憶の安定性と可塑性を保ったまま、学習と想起を行なうことができるように設計されたモデルである。したがって、$\rho = 1$ の下で分類動作を行なった結果 1 回で分類できなかったからといってモデルが欠陥であるわけではない[2]。

[1] 先に述べた subset 問題を除けば、$\rho = 1$ で学習を行なえば 1 回の提示で学習は終了する。

[2] 人間でさえ 1 回では覚え切れないのだ。すべての人間が何事も 1 回で覚えることができるのであれば、今頃は……

第 B 章

シミュレーション結果

節 4.4での検証のためのシミュレーション結果を示す。図 B.1、B.2、B.3、B.4は、それぞれ 1、2、3、4 番目のケースのシミュレーション結果である。ρ はすべて 0.9 とした。但し、図 B.5のシミュレーションでは、$\rho = 0.5$ である。また、P 層での学習における学習率 η は、0.1 とした。

B.1　図の見方

図中の文字列についての説明する。

Ptns： 何番目に提示されたパターンかを示す。

A ctg： A 層による分類結果で、どのカテゴリ表現層素子のカテゴリかを示す。

P class： P 層での学習終了語の出力結果。ビットの立っている素子がクラスを表す。

T： クラス表現層の何番目の素子がクラスを表現すべきかを示す、教師信号。

Ite： P 層での学習終了までの繰返し回数。0 であれば、期待通りの出力を出したことになる。

パターンの見方については、第 3章で述べたように、最左列のパターンが入力パターンであり、それより右のパターンが記憶されたパターンである。

B.2 解説

節 4.4におけるケース 1、2 のシミュレーションでは、入力が同一カテゴリに分類される必要があることから、subset 問題に抵触する 2 種類のパターンを入力として与えた。ケース 3、4 のシミュレーションでは、全く別の 2 種類のパターンを与えている。

さて、節 4.4で、全く追記学習による結合経路への影響がないとされたケース 1、3、4 の場合の結果、図 B.1、B.3、B.4をみると、2 回目の提示となる Ptns = 3,4 において、Ite = 0 であり、1 回目の提示で学習が終了していることが分かる。ここで、図 B.1では、1、2 番目に提示したパターンにおいて、双方とも Ite が 0 となっているが、偶然である。

図 B.2は、2 種類の入力パターンが同じカテゴリであるが、2 回目の提示で 2 つのカテゴリに分類されているため、Ptns = 1 のパターンだけが再学習されている。

図 B.5は、ケース 2 の特別の場合で、A 層での分類に失敗した場合である（ρ を低くすることで失敗させた)。節 4.4で述べたように、何回提示しても Ite = 1 であり、学習が収束していない。

(再掲)

新規パターンと学習済みのパターンとで取り得るカテゴリとクラスの関係：

1. カテゴリが同じで、かつクラスも同じ。
2. カテゴリが同じで、かつクラスが異なる。
3. カテゴリが異なり、かつクラスは同じ。
4. カテゴリが異なり、かつクラスも異なる。

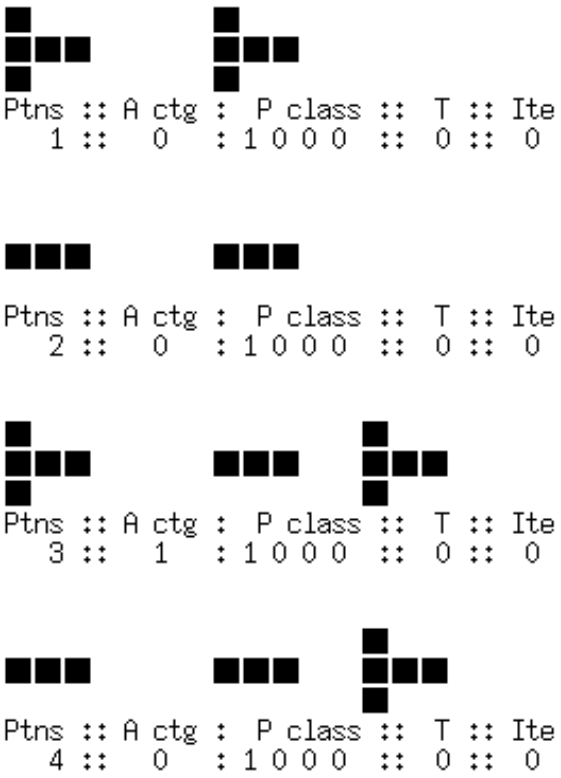


図 B.1: **ケース 1**

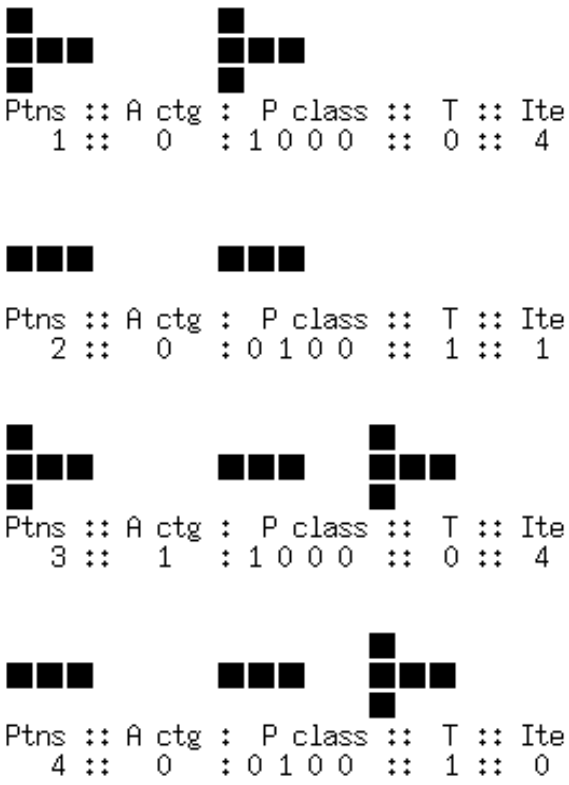


図 B.2: **ケース 2**

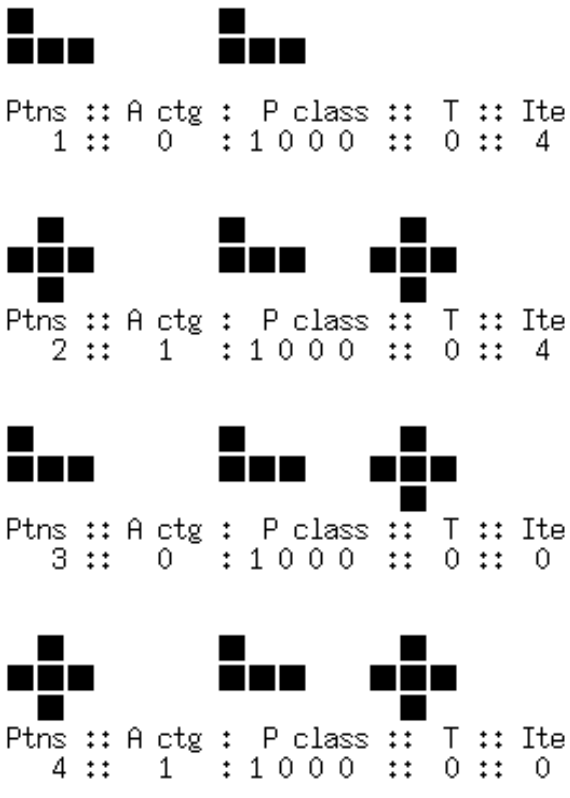


図 B.3: **ケース 3**

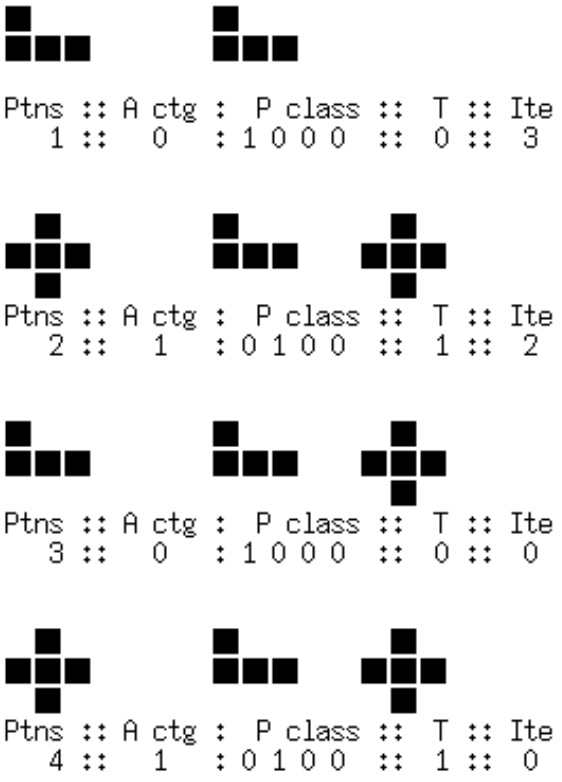


図 B.4: **ケース 4**

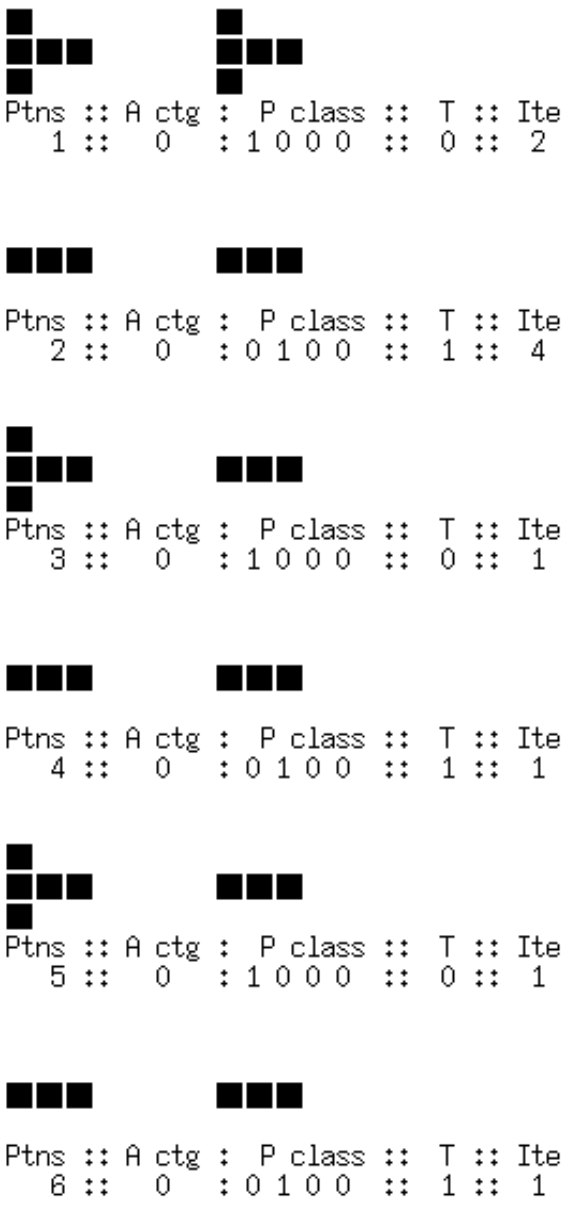


図 B.5: **ケース 2**　同一カテゴリで、異なるクラスに属すため学習が収束しない。