

多層ネットワークの瞬時学習法

塚本 義明[†] 生天目 章[†]

本論文では、多層ネットワークの新しい学習アルゴリズムとして、瞬時学習法を提案する。瞬時学習法とは、類似行列の性質から多層ネットワークの構造を事前に決定し、そのようにして構造化されたネットワーク上にユニット間の結合係数を求める方法である。入出力がブール値で定義される学習例間に類似度を導入し、学習例の集合上に2次元ベクトルの集合として求められる類似行列を新しく定義する。学習例の類似行列の構造から多層ネットワークの構造(中間ユニットの数やそれらの内部表現)について決定できることを示す。従来の学習法は、ネットワークの構造やユニット間の結合係数が一度決定されてしまうと、新しい学習例の追加や入出力ユニットの変更を伴う学習環境の動的な変化に容易に適応することが困難である。しかしながら、瞬時学習法は、新しい学習例の追加、入力情報の新しい属性の追加および学習例の新しい分類法の追加といった動的な学習問題に対しても、容易に適用できる特長を有する。

Flash Learning for Multi-layer Networks

YOSHIAKI TSUKAMOTO[†] and AKIRA NAMATAME[†]

Neural networks in nature are not designed but evolved, and it learns its structure through with its environment. Because the structure of neural networks is evolved and developed rather than pre-specified, we have to develop a framework for an adaptable process. This paper proposes a general framework for structural adaptation to allow a neural network to change its structure autonomously. We describe a two-stage structured training procedure for multi-layer networks based on the group-similarity measure. We propose a new learning algorithm to capture the weights of the connections in the network with a single presentation of the training set. A connectionist learning in a dynamic environment is then described as the process of the self-modification process of the group-similarity measure under a new learning environment. The group-similarity measure poses some invariant property for the change of the environment. In the case the previously acquired the learning parameters should be modified, the new learning parameters are obtained by modifying the group-similarity measures.

1. はじめに

コネクショニスト情報処理の基木モデルは、ユニットと呼ぶ神経細胞のような働きをする単純な素子が、重み付きの結合係数により相互結合されたネットワークによる情報処理モデルである。ユニットの内部パラメータ(結合係数)をいかに獲得するかはコネクショニストモデルの基本的課題の一つで、今までさまざまな学習アルゴリズムが提案されている。また、多層ネットワークモデルは、任意の入出力関係について学習・表現することができ、コネクショニストモデルの中で最も活発に研究されている分野の一つである^{19),20)}。

パセプトロンや誤差逆伝播法^{17),22)}は、結合係数を

逐次修正するために誤差訂正法を用いており、フィードバックのない入力層から出力層まで逐次配置されたユニット内部の適切なパラメータを獲得するために適用することができる。しかしながら、それらのアルゴリズムはさまざまな欠点を抱えている。例えば、誤差訂正法における学習戦略は、信号誤差に比例するようなある一定の量によってすべての結合係数を調整しながら、誤差を最小化する方向へと漸次降下するような学習を行う。しかしながら、その収束は必ずしも最小点に到達するとは限らず、またその収束の過程において学習例を繰り返し提示しなければならない。したがって、こうした誤差訂正型学習アルゴリズムは、入力の特徴数や学習例が数多い場合には非効率的な学習戦略になる¹²⁾。また、これらアルゴリズムは汎化を促進するのに必要な入力の特徴間に重要または重要でないものの識別を十分に行えるかどうか不明確である。また、これらの学習アルゴリズムは、ネットワーク

[†] 防衛大学校情報工学教室
Department of Computer Science, National
Defense Academy

の構造について事前に規定するための方法がなく、特に中間層の構成法については試行錯誤により決定される。このような学習方法を非構造化によるアプローチ^{3),4),9)}と呼ぶ。

本論文では、多層ネットワークの新しい学習アルゴリズムとして、瞬時学習法を提案する。入出力がブール値で定義される学習例間に類似測度を導入し、学習例の集合上に2次元ベクトルの集合として求められる類似行列を新しく定義する。学習例の類似行列の構造から多層ネットワークの構造（中間ユニットの数やそれらの内部表現）について決定できることを示す。また、学習例の類似行列が線形分離可能であるならば、学習例は線形分離可能になり、学習例の線形識別関数の結合係数は類似行列より決定でき、したがって学習例を一度提示するだけで求められることを示す。構造化学習法とは、類似行列の性質から多層ネットワークの構造を事前に決定し、そのようにして構造化されたネットワーク上にユニット間の結合係数を求める方法である。

多層ネットワークは、ネットワークの学習メカニズムによりネットワークの構造やユニット間の結合係数が一度決定されてしまうと、新しい学習例の追加や入力ユニットの変更を伴う学習環境の動的な変化に容易に適應することが困難であり、ネットワークの構造を自己再編し自己成長するための機能を有していない¹⁴⁾。コネクショニストモデルの環境への適應能力や自己成長機能は高次の認知処理を実現していくために不可欠である^{15),20)}。新しい学習例の追加、入力情報の新しい属性の追加および学習例の新しい分類法の追加といった学習環境の変化に対しても、瞬時学習法は容易に適應できることを示す。

2. 基本モデルと定義

n 次元の属性ベクトル $x_t = (x_{t1}, x_{t2}, \dots, x_{tn})$, $t = 1, 2, \dots, T$, の集合 X で学習例の集合を表す。ここで x_{ij} は、2値のブール変数を表す。学習例の集合 X 上に定義される概念 C とは学習例の数 T に対応した要素 c_t , $t = 1, 2, \dots, T$, をもつベクトルである。ここで、 $c_t \in \{0, 1\}$ のブール値をとる。すなわち、学習例 X の概念 C とは、その要素が $c_t = 1$ となるような X の部分集合を真（正の学習例）とし、 $c_t = 0$ となる集合を偽（負の学習例）とする。学習例の集合 X は二つの部分集合に分離するためのカテゴリ関数として定義される。概念 C により分類された学習例 X の正および

負の学習例を以下のように表す。

$$\begin{aligned} C^+ &= \{x_t \in X : \langle x_t, c_t \rangle \in \langle X, C \rangle, c_t = 1\} \\ C^- &= \{x_t \in X : \langle x_t, c_t \rangle \in \langle X, C \rangle, c_t = 0\} \end{aligned} \quad (2.1)$$

k 個の隠れユニットをもつ3層ネットワークでは、線形識別関数 f_j , $j = 1, 2, \dots, k$, が各層の活性化関数（ブール関数）となり新しい概念 H_j を定義し、それを中間概念という。中間概念が定義する中間層の属性空間を、 $H = (H_1, H_2, \dots, H_k)$ で表す。ここで H_j の t 番目の要素はブール値で $h_{tj} = f_j(x_t)$, $t = 1, 2, \dots, T$, で表す。すなわち中間層においては、学習例の集合 X を入力し中間概念 (H_1, H_2, \dots, H_k) により X を変換する。概念 C は、中間層が定義する属性空間 H を同じように正および負の学習例に分類し、それを次のように表す。

$$\begin{aligned} C^+ &= \{h_t \in H : \langle h_t, c_t \rangle \in \langle H, C \rangle, c_t = 1\} \\ C^- &= \{h_t \in H : \langle h_t, c_t \rangle \in \langle H, C \rangle, c_t = 0\} \end{aligned} \quad (2.2)$$

学習例の集合 X に対して、次のような学習例のグループ化関数を定義する。

$$\begin{aligned} G^+(x_t) &= \sum_{x_s \in C^+} S(x_t, x_s) \\ G^-(x_t) &= \sum_{x_s \in C^-} S(x_t, x_s) \end{aligned} \quad (2.3)$$

ここで $S(x_t, x_s)$ は、

$$S(x_t, x_s) = \sum_{i=1}^n x_{ti} x_{si} \quad (2.4)$$

で定義される学習例 x_t と x_s 間の類似測度である。すなわち、 $G^+(x_t)$ はある学習例 x_t と概念 C のすべての正の学習例との類似測度の総和を表し、正の学習例の特徴が集約的に表現される。一方、 $G^-(x_t)$ は概念 C のすべての負の学習例との類似測度の総和で、負の学習例の特徴について集約して表現したものである。

3. 学習例のグループ化関数

学習例は、中間層での中間概念に基づいてグループ化されるが、従来のコネクショニストモデルの学習アルゴリズムでは学習例の部分集合を中間層にどのように割り当てるか、明らかになっていない^{14),21)}。しかし、式(2.3)で定義したような類似測度を用いれば学習例のグループ分類が容易になる。ここで、学習例 X の類似行列を以下のように定義する。

定義 3.1 式(2.3)で定義した学習例のグループ化関数 $G^+(x_t)$, $G^-(x_t)$ を学習例の集合 X のすべてについて求めた $T \times 2$ 行列

$$T(X, C) = [\{G^+(x_t), G^-(x_t)\} : t = 1, 2, \dots, T] \quad (3.1)$$

を概念 C の下で学習例 X の類似行列と定義する。

補題 3.1 式 (3.1) の類似行列は次式で与えられる。

$$T(X, C) = XX^T(C, \mathbf{1} - C) \quad (3.2)$$

ここで、 $\mathbf{1}$ は要素として 1 をもつ列ベクトルを表す。

証明 $T(X, C)$ の t 番目の要素である $G^+(x_t)$ は、式 (3) の定義より

$$\begin{aligned} G^+(x_t) &= \sum_{x_s \in C^+} S(x_t, x_s) \\ &= \sum_{s=1}^T S(x_t, x_s)c_s \end{aligned} \quad (3.3)$$

ここで、 $S(x_t, x_s)$ は式 (3.1) の定義より行列 XX^T の (t, s) 要素になる。同様に、

$$\begin{aligned} G^-(x_t) &= \sum_{x_s \in C^-} S(x_t, x_s) \\ &= \sum_{s=1}^T S(x_t, x_s)(1 - c_s) \end{aligned} \quad (3.4)$$

となるので式 (3.2) を得る。 □

以上で定義された類似行列 $T(X, C)$ は次のような性質をもつ。

定理 3.1 類似行列 $T(X, C)$ が線形分離可能ならば、すなわち、学習例の集合 X のそれぞれの要素 $x_t, t=1, 2, \dots, T$ に対し

[i] $x_t \in C^+$ ならば
$$\alpha G^+(x_t) - \beta G^-(x_t) + \theta > 0 \quad (3.5)$$

[ii] $x_{t'} \in C^-$ ならば
$$\alpha G^+(x_{t'}) - \beta G^-(x_{t'}) + \theta < 0 \quad (3.6)$$

となるパラメータ α, β, θ が存在するならば学習例の集合 X は概念 C の下で線形分離可能である。また、各属性 x_i に対する結合係数 w_i を

$$w_i = \alpha \sum_{x_s \in C^+} x_{si} - \beta \sum_{x_s \in C^-} x_{si} \quad (3.7)$$

とする次式で定義される関数

$$F(x) = \sum_{i=1}^n w_i x_i + \theta \quad (3.8)$$

は、学習例 X 上の概念 C の線形識別関数になる。すなわち、

$$\begin{aligned} (i) \quad &x_t \in C^+ \text{ ならば } F(x_t) > 0 \\ (ii) \quad &x_t \in C^- \text{ ならば } F(x_t) < 0 \end{aligned} \quad (3.9)$$

証明 式 (3.7) を満たす α, β, θ が存在するならば式 (3.9) で定義される関数が学習例 X の線形識別関数になることを示せば十分である。

(i) ある正の学習例 $x_t \in C^+$ に対して

$$\begin{aligned} F(x_t) &= \sum_{i=1}^n w_i x_{ti} + \theta \\ &= \alpha \sum_{s \in C^+} S(x_t, x_s) - \beta \sum_{s' \in C^-} S(x_t, x_{s'}) + \theta \\ &= \alpha G^+(x_t) - \beta G^-(x_t) + \theta \end{aligned}$$

> 0

(ii) ある負の学習例 $x_{t'} \in C^-$ に対して

$$\begin{aligned} F(x_{t'}) &= \sum_{i=1}^n w_i x_{t'i} + \theta \\ &= \alpha \sum_{s \in C^+} S(x_{t'}, x_s) - \beta \sum_{s' \in C^-} S(x_{t'}, x_{s'}) + \theta \\ &= \alpha G^+(x_{t'}) - \beta G^-(x_{t'}) + \theta \\ &< 0 \end{aligned}$$

式 (3.7) (3.8) で定義される関数は線形識別関数となり、学習例 X は概念 C の下で線形分離可能である。 □

定理 3.1 の不等式は、学習上のクレジット量 (正しい出力の場合の利得 (α) および間違った出力の場合のペナルティ (β)) を求めるものである。しかしながら、定理 3.1 は、十分条件であり、線形分離可能な問題の中に、学習上のクレジット量が定理 3.1 によって決定できない問題が存在する。そのような問題は一つの直線上に正および負の学習例の類似尺度がのってしまうような問題である。本来、線形分離可能でありながら学習上のクレジットの量が決定できないような問題を少なくするために、以下のような補完表現法により瞬時学習法の学習性を強化する。

定義 3.2 (補完表現) 補完表現は、 n 個のブール変数 $x = (x_1, x_2, \dots, x_n)$ に対し、それぞれのブール変数の否定 $\bar{x}_i = 1 - x_i$ をもつ。 $2n$ 個の変数 $\underline{x} = \{x_1, 1 - x_1, x_2, 1 - x_2, \dots, x_n, 1 - x_n\}$ を x の補完表現と定義する。また、補完表現されたブール変数 \underline{x}_i および \underline{x}_s の類似測度 $S(\underline{x}_i, \underline{x}_s)$ をその内積

$$S(\underline{x}_i, \underline{x}_s) = \sum_{i=1}^n x_{i1} x_{s1} + \sum_{i=1}^n (1 - x_{i1})(1 - x_{s1}) \quad (3.10)$$

で定義する。 □

性質 3.1 \underline{x}_i および \underline{x}_s のハミング距離 $H(\underline{x}_i, \underline{x}_s)$ に対して、次の関係が成り立つ。

$$S(\underline{x}_i, \underline{x}_s) = 2n - H(\underline{x}_i, \underline{x}_s) \quad (3.11) \quad \square$$

補題 3.2 学習例の集合 X のそれぞれの要素 $x_t, t=1, 2, \dots, T$ を補完表現して形成される学習例 X の類似行列 $T(\underline{X}, C)$ は次式で与えられる。

$$T(\underline{X}, C) = \{XX^T + (I - X)(I - X^T)\} [C, \mathbf{1} - C] \quad (3.12) \quad \square$$

証明 $T(\underline{X}, C)$ の t 番目の要素である $G^+(\underline{x}_t)$ は、

$$\begin{aligned} G^+(\underline{x}_t) &= \sum_{\underline{x}_s \in C^+} S(\underline{x}_t, \underline{x}_s) \\ &= \sum_{s=1}^T S(\underline{x}_t, \underline{x}_s)c_s \end{aligned}$$

ここで、 $S(\underline{x}_t, \underline{x}_s)$ は式 (3.2) より、行列 $XX^T + (I - X)(I - X^T)$ の (t, k) 要素になる。同様に

$$\begin{aligned} G^-(x_i) &= \sum_{x_i' \in C^-} S(x_i, x_i') \\ &= \sum_{s=1}^T S(x_i, x_s)(1-c_s) \end{aligned}$$

となるので式(3.12)が得られる。□

定理 3.2 学習例の集合 X の補完表現 \bar{X} の類似行列 $T(\bar{X}, C)$ が線形分離可能ならば, i 番目の属性 x_i に対応した入力ユニットから概念 C に対応する出力ユニットへの結合係数 w_i^+ およびその否定変数 $\bar{x}_i = 1 - x_i$ からの結合係数 w_i^- は次式で与えられる。

$$w_i^+ = \alpha \sum_{x_i \in C^+} x_i - \beta \sum_{x_i' \in C^-} x_i' \quad (3.13)$$

$$w_i^- = \delta - w_i^+ \quad (3.14)$$

ここで, α, β は類似行列 $T(\bar{X}, C)$ を線形分離にする学習パラメータである。また, $\delta = \alpha * C^+ - \beta * C^-$, $*C^+$ および $*C^-$ は学習例の集合 X に含まれる正の学習例および負の学習例の数を表す。□

証明 定理(3.1)より, 式(3.4)は明らかである。また, 否定変数 \bar{x}_i からの結合係数は,

$$\begin{aligned} w_i^- &= \alpha \sum_{x_i \in C^+} (1 - x_i) - \beta \sum_{x_i' \in C^-} (1 - x_i') \\ &= (\alpha * C^- - \beta * C^-) - w_i^+ \\ &= \delta - w_i^+ \end{aligned} \quad (3.15) \quad \square$$

定理 3.3 学習例の集合 X の補完表現 \bar{X} の類似行列 $T(\bar{X}, C)$ が線形分離可能ならば, 学習例の集合 X は概念 C の下で線形分離可能で, その線形識別関数は次式で与えられる。

$$f(x, w) = \sum_{i=1}^n (2w_i^+ - \delta)x_i + n\delta - \sum_{i=1}^n w_i^+ + \theta \quad (3.16)$$

ここで, w_i^+, δ は定理(3.2)で与えたパラメータで, θ は $T(\bar{X}, C)$ に対し式(3.5)(3.6)を満足するパラメータである。□

証明 補完表現の下で線形識別関数は,

$$f(x, w) = \sum_{i=1}^n w_i^+ x_i + \sum_{i=1}^n w_i^- (1 - x_i) + \theta \quad (3.17)$$

で与えられるので, 定理3.2の関係を用いると式(3.17)は,

$$f(x, w) = \sum_{i=1}^n (2w_i^+ - \delta)x_i + n\delta - \sum_{i=1}^n w_i^+ + \theta \quad (3.18)$$

で与えられる。したがって, 式(3.18)は学習例 X の線形識別関数となり, X は概念 C の下で線形分離可能である。□

後述するように概念学習においては, 学習例の集合

X の補完表現を用いることにより, より汎化能力の高い学習が可能である。しかしながら, 入力ユニットが2倍になりネットワークの規模に問題が生じる。しかし, 定理3.3により補完表現に対しても入力ユニットの数は変わらないことを示した。

4. 瞬時学習法

4.1 瞬時学習アルゴリズム

生物には, ある環境下で“One-Shot Learning”^{2), 16)}といわれている能力がある。特異なパターンの単一の表現を一度見たあとで, 他のいく千もの中から選び出すことができる。ここで学習例が線形分離可能な場合の One-Shot Learning の性質をもつ新しい学習アルゴリズムとして, 瞬時学習法¹⁴⁾を提案する。このアルゴリズムは, 2層間の学習法である。入力ユニットを $x_i, i=1, 2, \dots, n$, 出力ユニットを $o_j, j=1, 2, \dots, k$ とし x_i と o_j 間の結合係数を w_{ij} とする。結合係数の修正学習戦略とは, t 番目の学習例 x_i に対し, i 番目の入力ユニットの値が x_{it} , j 番目の出力ユニットの望ましい値が o_{jt} のとき, w_{ij} を次のように修正する方法である。

$$\begin{aligned} \text{[i]} \quad \Delta w_{ij} &= \alpha_j \text{ if } x_{it}=1 \text{ and } o_{jt}=1 \\ \text{[ii]} \quad \Delta w_{ij} &= -\beta_j \text{ if } x_{it}=1 \text{ and } o_{jt}=0 \\ \text{[iii]} \quad \Delta w_{ij} &= 0 \text{ if } x_{it}=0 \end{aligned} \quad (4.1)$$

$\alpha_j, \beta_j, j=1, 2, \dots, k$, は, 定数である。すべての学習例を一度与えることにより, 式(4.1)の修正ルールによって学習例の集合をすべて提示した後の結合係数を次のように決定できる。

$$\begin{aligned} w_{ij} &= \alpha_j \sum_{t=1}^T x_{it} o_{jt} - \beta_j \sum_{t=1}^T x_{it} (1 - o_{jt}) \\ &= \alpha_j \sum_{x_i \in C_j^+} x_{it} - \beta_j \sum_{x_i' \in C_j^-} x_{it}' \end{aligned} \quad (4.2)$$

ここで, C_j^+ および C_j^- は j 番目の出力ユニット O_j の値によって決定される概念 C_j が定義する学習例 X の正および負の学習例を表す。

4.2 瞬時学習アルゴリズムの性質

瞬時学習アルゴリズムは, 相関学習¹⁰⁾や学習行列アルゴリズム¹⁸⁾と類似しているが, いずれの方法も結合係数が負の値にならないように式(4.1)の(ii)に基づく修正パラメータは小さく抑えられている。瞬時学習法は負のパラメータに対する修正は正のレベルと同じように大きなものとし, 抑制型の結合係数の獲得を学習において重視している点に特徴がある。

特に, 学習例の集合 X の補完表現による式(3.12)の類似行列を用いた場合の結合係数の修正は,

$$\begin{aligned}
 [i] \quad & \Delta w_{ij} = \alpha_j \text{ if } x_i = 1 \text{ and } o_j = 1 \\
 & \text{or } x_i = 0 \text{ and } o_j = 0 \\
 [ii] \quad & \Delta w_{ij} = -\beta_j \text{ if } x_i = 1 \text{ and } o_j = 0 \\
 & \text{or } x_i = 0 \text{ and } o_j = 1
 \end{aligned} \tag{4.3}$$

で与えられる。学習例 X が線形分離可能な場合の識別関数の結合係数を求める方法として、ベイズの最適化ルールがよく知られている。ベイズの最適化ルールでは、結合係数 w_{ij} は次式で与えられる¹⁵⁾。

$$w_{ij} = \ln O^+(x_i) - \ln O^-(x_i) \tag{4.4}$$

ここで、

$$\begin{aligned}
 O^+(x_i) &= \Pr(x_i = 1 | C_j^+) + \Pr(x_i = 0 | C_j^+) \\
 O^-(x_i) &= \Pr(x_i = 1 | C_j^-) + \Pr(x_i = 0 | C_j^-)
 \end{aligned} \tag{4.5}$$

したがって、瞬時学習法において学習パラメータ α_j , および β_j を

$$\begin{aligned}
 \alpha_j &= \ln O^+(x_i) / O^+(x_i) \\
 \beta_j &= \ln O^-(x_i) / O^-(x_i)
 \end{aligned} \tag{4.6}$$

で与えると、ベイズの決定ルールに基づく結合係数は、

$$w_{ij} = \alpha_j O^+(x_i) - \beta_j O^-(x_i) \tag{4.7}$$

となり、瞬時学習法で求められる特別な場合になり、瞬時学習法はベイズの最適化ルールよりも汎化性の高い学習結果を導くことができる。さらに、次節で示すように瞬時学習法は線形分離不可能な問題にも適用できることからより一般性の高い学習アルゴリズムである。

入力ユニットの数を n 個、出力ユニットの数を k 個とすると、瞬時学習アルゴリズムによって結合係数およびしきい値を求めるために必要な計算量は $O(nk^2)$ である。一方、誤差逆伝播法による場合に必要計算量は $O(nk^2L)$ である。ここで L は、誤差逆伝播法の学習回数とする。したがって、瞬時学習法の計算量は誤差逆伝播法の1回の学習回数に必要な計算量と同程度である。

4.3 多層ネットワークの学習

学習例の集合 X を概念 C に基づいて分類するための多層ネットワークを構築する方法として図1に示す構造決定過程と学習過程の二つの過程からなる学習方法について提案する。構造決定過程では、学習例 X の類似行列 $T(X, C_j)$, $j=1, 2, \dots, k$, からネットワーク構造すなわち中間ユニットの数やその活性化値を決定する。学習例の類似行列 $T(X, C_j)$ が線形分離でない場合はいくつかの中間概念を生成する。学習過程では、入力層と中間層および中間層と出力層との間の結合係数を4.1節で示した瞬時学習法によって逐次決定す

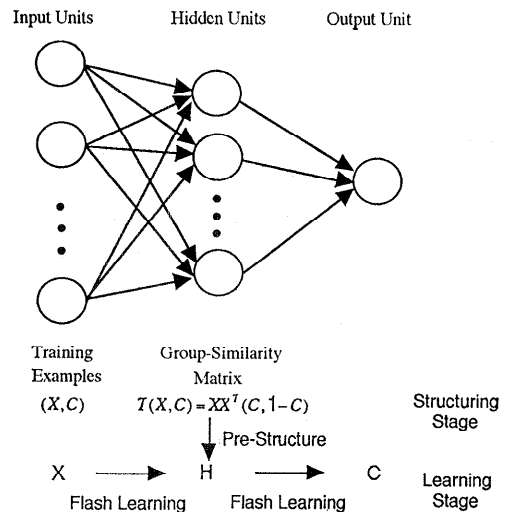


図1 瞬時学習法の手順
Fig. 1 Procedures for structured connectionist learning.

る。このような多層ネットワークの学習法を構造化学習法と呼ぶ。

$T(X, C_j)$ が線形分離可能でない場合、類似行列 $T(X, C_j)$ を線形分離不可能にしている学習例の集合を求める。すなわち、概念 C_j に対して求められるグループ化関数 $G_j^+(x_i)$, $G_j^-(x_i)$ に対して式(3.5)および(3.6)を満足しない学習例(一般性を失うことなくこれを正の学習例と仮定できる) x_s を抽出し、これを正の学習例から負の学習例へ分類変更するような新しい $\bar{C} = C - H_s$, および中間概念 H_s を生成する。ここで、 H_s は s 番目の要素を1他を0とする列ベクトルである。新しい中間概念 \bar{C} の下でのグループ化関数は、

$$\begin{aligned}
 \bar{G}_j^+(x_i) &= G_j^+(x_i) - S(x_i, x_s) \\
 \bar{G}_j^-(x_i) &= G_j^-(x_i) + S(x_i, x_s)
 \end{aligned} \tag{4.8}$$

で与えられ、概念 C の下で線形分離不可能となる学習例の集合 X は、新しい概念 $\bar{C} = C - H_s$ の下では線形分離可能な状態に近づく。以上のような手順で、類似行列 $T(X, C_j) = \{G_j^+(x_i), G_j^-(x_i)\}$ を線形分離不可能にしている正の学習例 $Y = \{x_1, x_2, \dots, x_{s-1}\}$ をいくつか抽出し、それらを負の学習例に分類するような中間概念を生成する。すなわち、 Y を局所表現とするような k 個の中間概念 H_s , $s=1, 2, \dots, k$, を生成する。 H_s は Y の s 番目の学習例 $x_s \in Y$ だけを唯一の正の学習例と識別し、他の学習例は負の学習と分類する。すなわち、 H_s は s 番目の要素が1で他は0とな

るような単位列ベクトルである。また、最後の中間概念 H_k を次式で与える。

$$H_k = C - \bigcup_{s=1}^{k-1} H_s \quad (4.9)$$

すなわち、 H_k は類似行列 $T(X, C_j)$ を線形分離不可能としている正の学習例を負の学習例のグループに分類するよう生成された概念である。また、中間概念 H_k は、線形分離不可能な類似行列 $T(X, C)$ の中からいくつかの正の学習例を抽出し、それを線形分離可能にするように選ばれた中間概念であり、したがって類似行列 $T(X, H_k)$ は、線形分離可能になる。各中間概念 $H_j, j=1, 2, \dots, k-1$, はある一つの学習例を正の学習例および残りのすべての学習例を負の学習例と識別するもの(局所表現)であり、明らかに線形分離可能である。したがって、学習例 X はそれぞれの中間概念 $H_j, j=1, 2, \dots, k$, の下で線形分離可能である。また概念 C は中間概念 $H = \{H_1, H_2, \dots, H_{k-1}\}$ の選言形式で表現でき、したがって C はそれぞれの中間概念のとり値によって定義される中間概念の学習空間 H に対し線形分離可能である。

学習例の集合 X が概念 C_j の下で線形分離不可能な場合、何個の中間概念(すなわち中間ユニットの数)を生成すべきかは類似行列 $T(X, C_j)$ の構造に依存し、最低2個は必要で、また最高 C_j の正(または負)の学習例の数だけあれば線形可能になる。類似行列が線形分離不可能な場合、それを線形分離可能にするよう学習例をグルーピングするための計算量も学習例を生成している学習問題の構造に依存する。すなわち、類似尺度が決定する2次元空間において線形分離不可能になっている学習例の数に比例する。本論文が対象としている 0/1 表現の学習例をから成る学習問題は、それを生成しているブール関数を同定する問題として定式化できる。例えば、学習例を生成しているブール関数の選言項が少なく、学習例が比較的構造化されている場合には、類似行列を線形分離不可能にする学習例は少ない。逆に、ブール関数の選言項が多く学習例が構造化されていない parity 問題では、線形分離不可能にしている学習例が多くなる。この場合、一つの直線上に正および負の学習例の類似尺度が位置することになる。また、それぞれの学習例の類似尺度は、その学習例のもつ一つの数に応じて学習例はグルーピングされて一つの直線上に位置することになる。しかしながら、この場合でも、類似行列を線形分離不可能にする学習例を求めるための計算量は、 n 変

数のブール値をもつ学習例が構成する学習空間に対し、 $n+1$ 個の中間ユニットを求めるための計算量で済む。

4.4 学習環境の変化に対する適応

初期の学習例の集合 $[D|C]$ に基づき学習した後、新しく学習例が追加された場合や新しく入力属性が追加された場合には、新しい学習例の集合の下で類似行列を求めそれに応じて式(3.5)および式(3.6)を満たすように学習パラメータ α, β, θ を修正する。また新しい概念の追加に対しては、新しく出力ユニットを生成しそれに対応したグループ化関数や学習パラメータを決定をする。この場合は、既存の出力ユニットの学習パラメータは不変である。以下に、学習環境が変化した場合の瞬時学習アルゴリズムを示す。

Step 1 類似行列の修正

$$T(X_{new}, C_{new}) = X_{new} X_{new}^T T_{new} [C_{new}, 1 - C_{new}] \quad (4.10)$$

Step 2 学習パラメータ $r_{new} = (\alpha_{new}, \beta_{new}, \theta_{new})$ の決定

$T(X_{new}, C_{new})$ に対して式(3.5)および式(3.6)が成立するような r_{new} を求める。

Step 3 重み結合係数の決定

新しい学習パラメータに対して重み結合係数

$$w_i = \alpha_{new} \sum_{x_i \in C^+} x_{ii}^{new} - \beta_{new} \sum_{x_i \in C^-} x_{ii}^{new} \quad (4.11)$$

を求める。

このように、瞬時学習法は学習環境の変化に対しても容易に適用でき、また学習環境が変化する以前に獲得した類似行列や学習パラメータの再利用が可能である。

5. 適用例

コネクショニストモデルのアプローチは、非シンボリックな学習により知識を獲得し、それをネットワークの内部表現として自動的にネットワークの内部モデル表現される点に特徴がある^{7), 19)}。多層ネットワークによるアプローチは、ネットワークの入力から出力までを一種のブラックボックスと見立て、学習例パターンを提示し、知識を獲得、学習させるものである。知識は、ネットワーク全体の重みの集合として分散表現されるので、獲得した知識を外部にルール形式などで明示的に表現することは困難である。しかしながら、ネットワーク構造が比較的単純でありまた静的であるために、シンボルに基づく知識表現モデルと比較し、構造データの表現力に乏しいという問題がある⁸⁾。した

がってコネクショニストモデルを用いて実用的な知識処理システムを実現する場合は、問題領域に関する知識や概念をいかにネットワークで表現するかが重要である^{1),6),20)}。例として図2に示す16個の知識⁹⁾からなる概念を構成するネットワークを構築する。これは、

- C₁: (dog, hates, monkey)
- C₂: (dog, eats, meat)
- C₃: (monkey, hates, dog)
- C₄: (monkey, eats, banana)
- C₅: (cat, hates, dog)
- C₆: (cat, eats, milk)
- C₇: (cat, eats, mouse)
- C₈: (mouse, hates, cat)
- C₉: (mouse, eats, cheese)
- C₁₀: (mouse, eats, milk)
- C₁₁: (banana, isa, food)
- C₁₂: (banana, color, yellow)
- C₁₃: (milk, isa, food)
- C₁₄: (milk, color, white)
- C₁₅: (cheese, isa, food)
- C₁₆: (cheese, color, yellow)

図2 学習対象の知識集合
Fig. 2 Knowledge of test world.

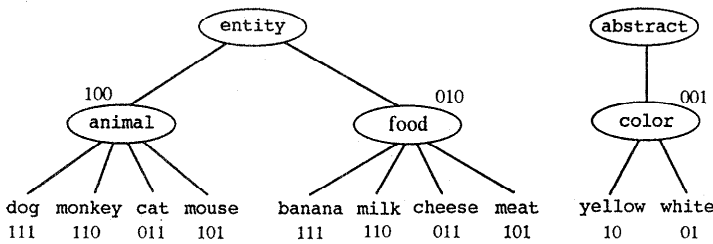


図3(a) 知識の構成要素と階層表現
Fig. 3(a) Components and hierarchy representation of knowledge.

表1(a) 対象の意味表現コード
Table 1(a) Semantic code for objects.

object	semantic coding
dog	(10011100000)
monkey	(10011000000)
cat	(10001100000)
mouse	(10010100000)
food	(01000000000)
banana	(01000011100)
milk	(01000011000)
cheese	(01000011000)
meat	(01000010100)
yellow	(00100000010)
white	(00100000001)

その構成要素を入力として3対1で表現される知識を連想できるような概念ネットワークを学習させる問題である。

表現法には、局所表現や分散表現⁷⁾である。ここでは、クラス階層を考慮した意味表現を用いる。意味表現は、図3(a)に示すように対象を階層化し、意味コード=((クラスコード)(動物コード)(植物コード)(色コード))で表現する方法である。そのように対象を表現したものが表1(a)である。また同様に図3(b)に示すように知識を関係情報をもとにして分類し、概念=((hate Class)(eat Class)(isa Class)(color Class))で知識を意味表現したものが表1(b)である。

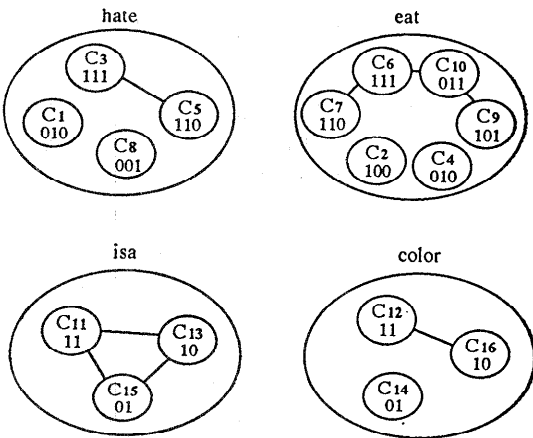


図3(b) 分類された知識
Fig. 3(b) Classified knowledge.

表1(b) 概念(知識)の意味表現コード
Table 1(b) Semantic code for concept.

concept	semantic coding
C ₁	(0100000000)
C ₂	(0001000000)
C ₃	(1110000000)
C ₄	(0000100000)
C ₅	(1100000000)
C ₆	(0001110000)
C ₇	(0001100000)
C ₈	(0010000000)
C ₉	(0001010000)
C ₁₀	(0000110000)
C ₁₁	(0000001100)
C ₁₂	(0000000011)
C ₁₃	(0000001000)
C ₁₄	(0000000001)
C ₁₅	(0000000100)
C ₁₆	(0000000010)

表 2 意味表現コードによる図2の知識の学習例 (入力ユニット: 26個, 出力ユニット: 10個) および非線形分離不可能な出力ユニットに生成された中間概念の個数とその値

Table 2 The set of the training examples for the learning problem in Fig. 2 with the semantic coding (26 inputs and 10 outputs) and the values of H_4 and H_5 that represent the intermediate-concepts for the target-concepts C_4 and C_5 that are not linearly separable.

Input		Intermediate		Output	
$X_1 X_2 X_3 \dots$	$\dots X_{26}$	H_4	H_5	$C_1 C_2 \dots$	$\dots C_{10}$
100111000000100010011000000		00	00	0100000000	
10011100000010001000010100		10	00	0001000000	
100110000000100010011100000		00	00	1110000000	
10011000000010001000011100		00	10	0000100000	
100011000000100010011100000		00	00	1100000000	
10001100000010001000011000		01	10	0001110000	
10001100000010010010100000		01	01	0001100000	
100101000000100010001100000		00	00	0010000000	
10010100000010001000001100		10	00	0001010000	
10010100000010001000011000		00	10	0000110000	
010000111000001001000000000		00	00	0000001100	
01000011100000100100000010		00	00	0000000011	
010000110000001001000000000		00	00	0000001000	
01000011000000100100000001		00	00	0000000001	
010000011000001001000000000		00	00	0000000100	
01000001100000100100000010		00	00	0000000010	

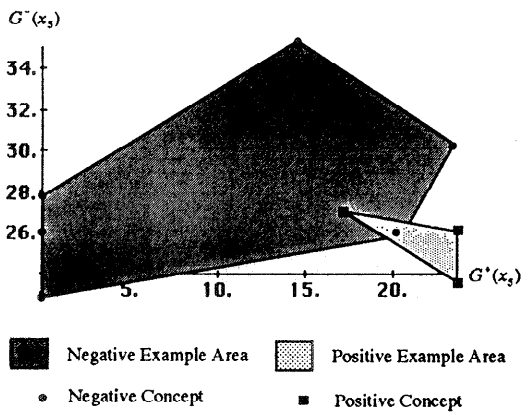


図 4(a) 第5出力ユニットの類似行列の描写 (中間ユニットなし)
Fig. 4(a) Similarity matrices of the 5th output unit without hidden units.

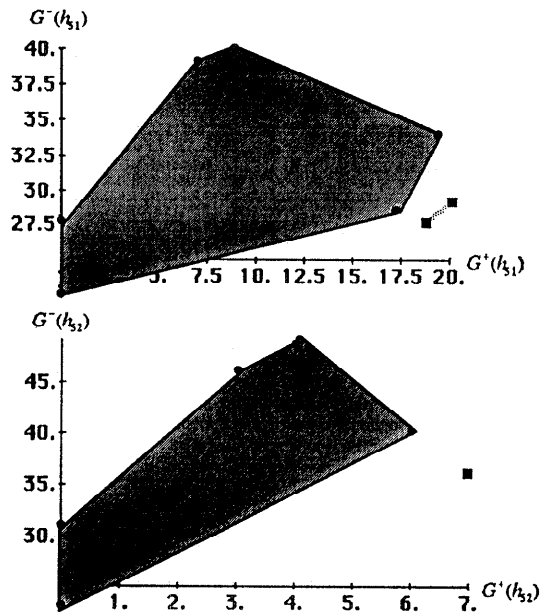


図 4(b) 中間ユニットの学習例の類似行列の描写
Fig 4(b) Similarity matrices of the hidden units for the output unit.

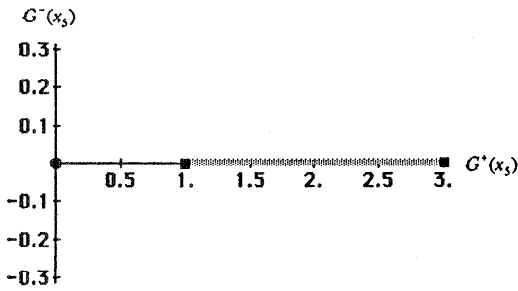


図 4(c) 第5出力ユニットの類似行列の描写 (中間ユニットあり)

Fig. 4(c) Similarity matrices of the 5th output unit with hidden units.

図2に示す知識の意味表現による学習例は表2のようになる。ネットワークの構造は、意味コードおよび概念コードよりビット数に相当するので入力ユニット26個、出力ユニット10個となる。表2の学習例に基づきそれぞれの概念(出力ユニット)の類似行列を求めると、第4・第5出力ユニットを除くがユニットが線形分離可能である。図4(a)に、線形分離不可能である第5出力ユニットの類似行列の描写図を示す。図4(a)は、類似行列を2次元表現(x軸: 正の類似尺度, y軸: 負の類似尺度)したもので、類似行列の線形分離可能性が視覚的に判別可能である。第4・第5出力ユニットのそれぞれの類似行列を線形分離可能にするためには中間ユニットが必要である。表2に示すようにそれぞれが2個の中間ユニットによって表現される中間概念 H_4 および H_5 を生成することにより線形分離可能になる。新しく生成された中間概念 H_6 の類似行列を図4(b)におよび中間概念が形成する学習例を入力とする第5出力ユニットの類似行列を図4(c)に示すが、いずれも線形分離可能である。表2の

学習例に基づいて獲得した多層ネットワークを図5に示す。従来の学習アルゴリズムでは、中間ユニットはすべての出力ユニットに結合されるが、瞬時学習法では線形分離不可能な出力ユニットだけに限定される。

6. おわりに

入出力がブール値で定義される概念学習に類似尺度を導入し、それによって定義される学習例の類似行列の構造から多層ネットワークの構造(中間ユニットの数やそれらの内部表現)について決定できることを示した。瞬時学習法は、一度学習例を提示するだけで多層ネットワークの結合係数を獲得でき、また、学習環境が変化した場合でもすでに学習して獲得した学習パラメータを事後学習に再利用できる点に特長があり、環境の変化に対しても容易に適応できることを示した。

本モデルは、Macintosh上に数式処理ソフト Mathematica および Smalltalk/V で実装されている。一つの大きな学習例をいくつか分割させて学習させる分散学習やそれぞれ独立して学習した複数のネットワークモジュールを複合せせるための複合学習に本モデルを拡張していくことが今後の課題である。

参考文献

- 1) Barnden, J. A. and Pollack, J. B. (eds) : *High-Level Connectionist Models*, Ablex Publishing (1991).
- 2) Brown, R. and Fulik, J. : *Flashbulb Memories, Cognition*, Vol. 5, pp. 73-99 (1977).
- 3) Fianty, A. : *Structured Connectionist Network Learning*, TR-252, University of Rochester (1988).

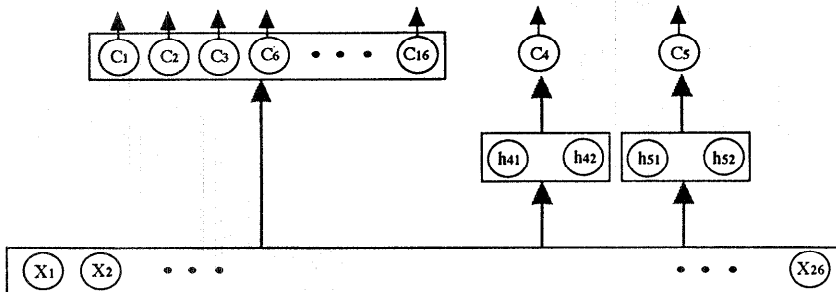


図5 表2の学習例が獲得したネットワーク構造(線形分離不可能な出力ユニットだけに中間ユニットが生成され結合されている)

Fig. 5 The network architecture for concept learning in Table 2 (The generated intermediate-concepts $h_{4i}, h_{5i}, i=1,2$ are specialized for nonlinearly separable concepts C_4 and C_5).

- 4) Feldmann, J. A.: Structured Connectionist Network Learning, TR-121, University of Rochester (1986).
- 5) 萩原 馨: コネクショニストモデルを用いた知識ベースシステム, 人工知能学会研究会資料, pp. 1-9 (1987).
- 6) Hampson, S. E. and Volper, D. J.: Disjunctive Models of Boolean Category Learning, *Biological Cybernetics*, Vol. 56, pp. 121-137 (1987).
- 7) Hinton, G. E.: Implementing Semantic Networks in Parallel Hardware, *Parallel Models of Association Memory*, Hinton, G. E. and Anderson, A. (eds.), Erlbaum (1981).
- 8) Hinton, G. E.: Mapping Part-Whole Hierarchies into Connectionist Networks, *Artif. Intell.*, Vol. 46, pp. 47-75 (1990).
- 9) 石川真澄: 忘却を用いたコネクショニストモデルの構造学習アルゴリズム, 人工知能学会論文誌, Vol. 5, No. 3, pp. 596-603 (1990).
- 10) Kohonen, T.: *Self-Organization and Associative Memory*, Springer-Verlag (1988).
- 11) Lee, T.: *Structure Level Adaptation for Artificial Neural Networks*, Kluwer Academic Publishers (1991).
- 12) Nadal, J.: Study of a Growth Algorithm for a Feedforward Network, *Int. J. of Neural Systems*, Vol. 1, pp. 55-59 (1989).
- 13) 生天目章, 大沢洋一: 並列推論ネットワーク: スキーマの表現と並列探索, 情報処理学会論文誌, Vol. 32, No. 12, pp. 1552-1565 (1991).
- 14) Namatame, A. and Tsukamoto, Y.: Structural Connectionist Learning with Complementary Coding, *Int. J. of Neural Systems*, Vol. 3, No. 1, pp. 19-30 (1992).
- 15) Nilsson, N. J.: *Mathematical Foundation of Learning Machines*, Morgan Kaufmann (1990).
- 16) Reiley, D. L., Cooper, L. N. and Wlbaum, C.: A Neural Model for Category Learning, *Biological Cybernetics*, Vol. 45, pp. 35-41 (1982).
- 17) Rumelhart, D., Hinton, G. E. and Williams, R. J.: Learning Representation by Backpropagation, *Nature*, Vol. 323, pp. 533-536 (1986).
- 18) Steinbuchi, K. and Piske, U. A.: Learning Matrices and Their Applications, *IEEE Trans*, Vol. EC-12, pp. 846-862 (1963).
- 19) Shastri, L.: *Semantic Networks: An Evidential Formalization and Its Connectionist Realization*, Morgann Kaufmann (1988).
- 20) Touretzky, D. S.: Special Issue on Connectionist Approaches to Natural Language Learning, *Machine Learning*, Vol. 7, No. 2-3 (1991).
- 21) Volper, D. L. and Hampson, S. E.: Connectionistic Models of Boolean Category Representation, *Biological Cybernetics*, Vol. 54, pp. 393-406 (1986).
- 22) Worbos, P. J.: Generalization of Backpropagation with Application to a Recurrent Gas Market Model, *Neural Network*, Vol. 1, pp. 339-356 (1989).

(平成 4 年 12 月 15 日受付)
(平成 5 年 7 月 8 日採録)



塚本 義明 (正会員)

1963年12月3日生。1986年防衛大学校卒業(応用物理学専攻)修了。現在、防衛大学校情報工学教室研究員。人工知能、日本ソフトウェア科学会各会員。



生天目 章 (正会員)

1950年10月4日生。1973年防衛大学校卒業(応用物理学専攻)。1977年および1979年スタンフォード大学大学院修士および博士課程修了(ph. D)。同年航空幕僚監部勤務。1987-1988年ジョージメイソン大学客員教授。現在、防衛大学校情報工学教室助教授。人工知能、ニューラルネットワーク、意思決定工学等の研究に従事。人工知能学会、ソフトウェア科学会、神経回路学会、AAAI、ACM、IEEE 学会各会員。