

# 自己増殖型ニューラルネットワークを用いたパターン情報ベースの推論機

須藤 明人<sup>†</sup>      張    晨犁<sup>†</sup>      坪山    学<sup>†</sup>      佐藤 彰洋<sup>†</sup>  
 長谷川 修<sup>††a)</sup>

Pattern Based Reasoning Using Self-Organizing Incremental Neural Network

Akihito SUDO<sup>†</sup>, Chenli ZHANG<sup>†</sup>, Manabu TSUBOYAMA<sup>†</sup>, Akihiro SATO<sup>†</sup>,  
 and Osamu HASEGAWA<sup>††a)</sup>

あらまし ロボットの知能に関する研究の突破口となり得るパターン情報ベースの推論について、新たな手法を提案する。シンボルをベースとした既存の推論機を知能ロボットに適用する際には(1)シンボル化装置の性能に限界があるという問題、(2)自律的なシンボルを生成する際に生じる問題、(3)シンボル化になじまないタイプのパターン情報が存在するという問題がある。本論文では、これらの問題の解決のためにパターン情報ベースの推論機が必要となることを指摘し、新しいパターン情報ベースの推論機を提案する。提案手法は「連言、選言、否定を含む任意の形の論理式を扱うことができる」、「パターン情報を多値ベクトルで表現する」、「汎化性能をもつ」、「推論結果の重複を回避する」、「雑音への耐性をもつ」といった知能ロボットに適用する際に重要な機能を併せ持っている。これらの機能を実現するために、提案手法はオンライン教師なし学習手法である Self-organizing Incremental Neural Network (SOINN) や連想記憶モデルである SOINN Associative Memory を拡張したアルゴリズムを採用している。

キーワード 自己増殖型ニューラルネットワーク、パターン情報、推論、知能ロボット

## 1. ま え が き

人工知能分野における論理・推論に関する研究の歴史は長い。人間の知能を計算機上で実現する上で、推論は人間の思考プロセスとして極めて重要な機能であると考えられ、多くの研究がなされてきた。推論に関する研究は多様であるが、あえて共通項でくくろうとすれば、知識をシンボルで表現しそれを利用して新しい知識を生成する記号処理の仕組みの研究であるといえるだろう [1]。しかし非正常な実環境下において汎用的なタスクを扱う必要のある知能ロボット等にとって、このような既存の推論システムの枠組みでは不十分で

ある。既存の推論システムが不十分である理由はいくつかあるが、筆者らは既存のシステムのほとんどがシンボル化された知識のみを操作する推論しか行えないことが重要な理由の一つであると考えている。

シンボルを操作する推論機として代表的なものにプロダクションシステムがあるが、これは事前に人間の専門家が if-then ルールの形でシステムにシンボルベースの知識を与えておく必要がある。システムの扱うタスクの範囲が限定されていれば、必要とされる知識を事前にシステムに与えることでタスクを解決できる。一方、非正常な実環境下で自律的に活動することを要求される知能ロボット等にとっての十分な知識を事前にすべて列挙することは不可能であり、新たな環境に適應するために自律的に知識を追加学習していくことが必要となる。このようなシステムが実世界からセンサを通じて獲得する情報はパターン情報であり、シンボル化はされていない。この情報を利用してシンボルを操作する推論機によって推論を行うためにはパターン情報をシンボル化する必要があるが、現在のパ

<sup>†</sup> 東京工業大学総合理工学研究所知能システム科学専攻, 横浜市  
 Department of Computational Intelligence and System Science, Tokyo Institute of Technology, 4259 Nagatsuta-cho, Midori-ku, Yokohama-shi, 226-8503 Japan

<sup>††</sup> 東京工業大学像情報工学研究施設, 横浜市  
 Imaging Science and Engineering Laboratory, Tokyo Institute of Technology, 4259 Nagatsuta-cho, Midori-ku, Yokohama-shi, 226-8503 Japan

a) E-mail: hasegawa@isl.titech.ac.jp

ターン識別機の性能はまだ不十分であり、例えば猫と犬を識別することすら簡単なことではない。そのため現在のところパターン情報をシンボル化する装置と従来のシンボルベースの推論機を組み合わせても性能に限界がある。

もし将来的に人間と同等の能力を有する識別機が開発されたとしても、二つの理由からシンボルベースの推論機だけでは不十分である。第1にシステムが自分でシンボルを生成する際に生じる問題がある。非定常な環境ではシステムに事前にすべてのシンボルを与えることは不可能であり、システムは環境に対応して自らシンボルを生成することが求められる。このときシステムによるシンボル化が人間が普段行っているように対象を用いた行動や思考の結果なされるものであるとすれば、シンボルを操作する推論機しかもたないシステムが適切なシンボル化を行うことは困難となってしまう。第2にシンボル化になじまないパターン情報が存在するという問題がある。例えば風景や絵などの画像の色彩や構図は我々人間にとってもシンボル化することは難しく、恐らくパターン情報のまま処理されているものと考えられる。

以上のように、非定常な実環境下において汎用的なタスクを扱う将来の知能ロボットにとって、シンボルを操作する推論機だけでは不十分であり、センサ情報であるパターン情報をシンボル化することなく推論を行うパターン情報ベースの推論を実現する必要がある。

ここまで述べてきたような観点以外にも、計算機上で熟練者の模倣を行うエキスパートシステムを実現する際に生ずる知識獲得問題の解決のためという観点[2]や、シンボルグランディング問題及びフレーム問題の回避という観点[3]からもパターン情報ベースの推論の実現は現在の人工知能の研究の限界を克服できる可能性をもっている。

## 2. 関連研究

上述のように大きな可能性を秘めているパターン情報ベースの推論であるが、実現するためのアプローチとして定まったものはまだ存在しない。

月本はパターン情報を関数としてとらえ、非古典論理である中間論理 LC を用いることで、関数ととらえたパターン情報の入出力関係を学習した3層のフィードフォワード型ニューラルネットワークを命題として扱う推論が行えることを詳細に示し、これをパターン推論として提案した[2]。また山根らはパターン情報を

2値ベクトルとして扱い、2値ベクトル間の関係を非単調神経回路網に学習させることで構成される大自由度力学系のダイナミクスを利用した推論をパターンベース推論として提案した[3]。月本の手法が連言(かつ)、選言(または)、否定を扱えることは、これらを扱うことができない山根らの手法に比べて大きな優位点である。

一方、知能ロボットへの応用を視野に入れたとき、パターン情報を関数とみなし、その入出力関係を学習したフィードフォワード型ニューラルネットワークを命題として扱う月本の手法よりも、2値ベクトルをそのまま命題として扱う山根らの手法の方がよい。月本の手法では異なるパターンごとに新しくフィードフォワード型ニューラルネットワークを用意する必要があるが、知能ロボットのセンサ情報は時間的にも空間的にも連続的なものであるから新しいニューラルネットワークを用意するタイミングを決めることが非常に難しい。これを失敗すると、ニューラルネットワークの数が爆発し知識の学習にもそれを用いた推論にもばく大な時間がかかってしまうか、知識として全く異なるパターンを同じ命題として扱ってしまうことにより学習した知識が無意味なものになってしまうおそれがある。一方、山根らの2値ベクトルでパターン情報を表現するアプローチでは、単一の非単調神経回路網に複数の知識を学習することになるためこのようなことは避けられる。これら両手法の利点をともに満たすような手法はこれまで提案されてこなかった。

## 3. 提案手法の特徴

### 3.1 従来研究を踏まえた提案手法の利点

本論文では[2]及び[3]の利点をかんがみ、連言、選言、否定を扱うことができ、かつ、実数値ベクトルとして扱われるパターン情報を命題とする if-then ルールを自己増殖型ニューラルネットワークで学習するシステムを提案する。すなわち、提案するニューラルネットワークはベクトルとして表現されたパターン情報を原子命題とし、これらの原子命題に連言、選言、否定を付加した命題を含意(ならば)で結ぶことで作られる任意の if-then ルールを学習することができる。例えば、A から F をパターン情報を表現するベクトルとしたとき、「 $A \rightarrow B$ 」や「 $((A \wedge B) \vee C) \rightarrow (D \wedge \neg E \wedge F)$ 」といった知識が学習の対象となる<sup>(注1)</sup>。このような if-then ルー

(注1): 本論文では連言、選言、否定、含意を  $\wedge, \vee, \neg, \rightarrow$  を用いて表す。

表 1 提案手法と従来手法の比較．括弧内の × は知能ロボットへの応用を見据えたときの利点及び欠点．  
Table 1 Comparative table between the proposed method and the relative works.

	論理演算子の取扱い	パターン情報の取扱い
提案手法	含意, 連言, 選言, 否定を扱える ( )	実数値ベクトルとして扱う ( )
月本 [2]	含意, 連言, 選言, 否定を扱える ( )	関数ととらえニューラルネットワークに学習させる ( × )
山根ら [3]	連言, 選言, 否定を扱えない ( × )	2 値ベクトルとして扱う ( )

ルを学習したシステムに対しファクトにあたるパターン情報が入力されると, 学習した if-then ルールを利用して演繹推論を行い, いくつかの結果を導き出すことができる. もし「 $(A \wedge B) \rightarrow (C \vee D)$ 」と「 $C \rightarrow (E \wedge \neg F)$ 」を学習したニューラルネットワークに対し「 $A \wedge B$ 」を入力すると「 $C \vee D$ 」「 $(E \wedge \neg F) \vee D$ 」を出力することができる. 提案手法が扱う範囲はシンボルベースの推論でいえば命題論理の枠組みで演繹推論を行うことに対応する. パターン情報を実数値ベクトルとして表現し, 連言, 選言, 否定を扱うことのできるパターンベースの推論機はこれまでに提案されていない. 参考までに, 提案手法と従来研究の比較を表 1 にまとめた.

### 3.2 知能ロボットへの応用を見据えた提案手法の利点

パターン情報ベースの推論機を将来の知能ロボット等へ応用する場合, 上段で述べた機能に加えて追加学習, 汎化性能, 雑音耐性, 推論結果の重複を防ぐ仕組みの 4 点も必要になる. 提案手法はこれら 4 点も実現している.

#### 3.2.1 追加学習性能

システムが学習すべき if-then ルールを人間が事前にすべて用意し与えることはできないため, システムは知識を追加的に学習する必要がある. その際, 新たな知識を学習することで既存の知識が破壊されてはならないし, 知識量や推論時間の爆発を防ぐために既存の知識と同じか極めて似ている知識は排除されなければならない.[3] では, 既存の知識の再学習による知識量の爆発を防ぐことはできるが, 新たな知識の学習による既存の知識の破壊が生じてしまう.[3] で提示されている実験によれば, 既学習知識の約 2 倍を追加学習すると 20% 弱の推論が失敗してしまう. 提案手法は知識の追加に伴って自律的にノードが増えていく自己増殖型ニューラルネットワークに基礎をおくアルゴリズム

を採用しているため, 追加学習で既存の知識が破壊されることはなく, また既存の知識と同じか類似した知識は新たに知識として追加することは行わない. 前者の性能が追加学習をする上で重要なことはもちろんだが, 後者の性能も, 知識量や推論時にかかる計算量の爆発を防ぐ上で重要な性能である.

#### 3.2.2 汎化性能

知能ロボットが環境から得るセンサ情報は, たとえ同じ対象であっても全く同じであるということは考えづらい. 提案手法は学習後に得たデータが学習データと多少異なっても対応可能である. しかも, どの程度の違いを許容するかを, パラメータの調整によって変化させることができるという柔軟性をもっている.[3] の手法も汎化性能をもっているが, 学習データからどの程度の違いを許容するかを変化させることはできない.

#### 3.2.3 推論結果の重複の防止

汎化性能を利用し, 獲得した知識と類似したデータを用いて推論を行えるにしても, 当然限界は存在する. そのため, シンボルに落とせば単一の if-then ルールで表すことができる知識もパターン情報ベースでは複数の if-then ルールとして蓄積することが必要になってくるだろう. これは, パターン識別の教師付き学習で, 汎化性能をもつ識別機であっても, あるラベルをもつパターンを複数用意してそれらを学習する必要があることに対応する(教師なし学習では, ある分布に属するデータを複数用意して学習しなければならないことに対応する). しかし, このことによって推論結果に重複が生じてしまうおそれがある. 例えばシステムが環境から  $\{a_i \rightarrow b_j | i, j = 1, 2, 3, \dots\}$  といった複数の if-then ルールを学習したとする. これらが別々の知識であるとシステムが認識すると, ファクトとして  $a_1$  を入力すれば推論結果として  $b_1, b_2, \dots$  が得られる.  $b_1, b_2, \dots$  が本当に別の推論結果として解釈されるべきであればそれでよいが, そうでなければ推論結果の重複を避けて一つだけ推論結果を出力することができた方がよい.[3] では非単調神経回路網の性質によりこの問題を回避可能であり, 提案手法では学習する if-then ルールをクラスタリングしながら蓄積することで推論結果の重複を回避している.

#### 3.2.4 雑音耐性

実環境下ではシステムへの入力に雑音が混じることは避けがたく, 一定の雑音耐性をもっていることが求められる. 実環境における雑音には 2 種類あると考



図 1 2 種類の雑音  
Fig.1 Two types of noise.

えられる [5] . 一つは、もとのデータに混入するケースであり、もう一つはもとのデータとは無相関なデータが入力されてしまうケースである . それぞれの例が図 1 (a) 及び (b) に示されている . [3] は図 1 (a) に示されるタイプの雑音に対する耐性を持っているが、(b) に示されるタイプの雑音が入力された場合は無意味な出力をしてしまう . 提案手法はどちらのタイプの雑音に対しても耐性をもっていることが特徴である .

3.1 及び 3.2 で述べたような機能を実現するため、提案手法はクラスタリングやトポロジー学習を目的としたニューラルネットワークモデルである Self-organizing Incremental Neural Network (SOINN) [4] や連想記憶モデルである SOINN Associative Memory (SOINN-AM) [5] のアルゴリズムを拡張して用いている . 提案手法の詳細を説明する 4.1 及び 4.2 においては、アルゴリズムを厳密に説明するために、ニューラルネットワークモデルからの観点ではなく知識工学の観点でモデルを説明するが、4.3 で提案手法が [4] や [5] を拡張した自己増殖型のニューラルネットワークとして解釈できることを示す .

#### 4. 提案手法

図 2 に示したように、提案手法のアーキテクチャは短期記憶、長期記憶、学習機、推論機からなる . 短期記憶には入力データが一時的に保存される . 長期記憶には学習した if-then ルールがエッジで結ばれることでクラスタリングされながら蓄積されていく . 提案手法は、if-then ルールを獲得する学習フェーズと、獲得した if-then ルールと入力されたファクトをもとに推論を行う推論フェーズに分けることができる . 学習フェーズでは実数値ベクトルで表現されるパターン情報を命題とする if-then ルールが入力され、所定のアルゴリズムに従ってそのデータをオンラインで学習する . その際、学習データを長期記憶に追加すべきかどうかや、既に長期記憶にあるデータで削除すべきものがあるかどうかを判断することができる . また、長期

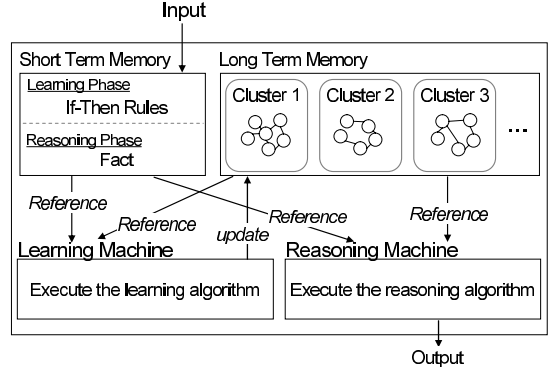


図 2 知識工学の観点でのアーキテクチャ  
Fig.2 The architecture.

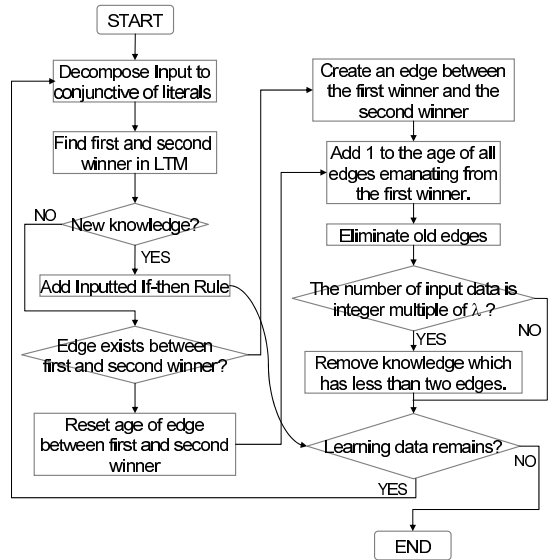


図 3 学習フェーズのフローチャート  
Fig.3 The flowchart of the learning phase.

記憶の if-then ルールは学習中にクラスタリングされる . これらの機能により追加学習及び推論結果の重複の回避を同時に行うことができる . 入力された if-then ルールのすべてをただ長期記憶に蓄積していくのではこれらを実現することはできない .

##### 4.1 学習フェーズ

ここでは、図 3 のフローチャートで表される学習フェーズのアルゴリズムの詳細を述べる .

学習データとして if-then ルールが入力されると、前処理として入力された if-then ルールを「リテラルの連言 → リテラルの連言」の形をした複数の if-then ルールに分解する . 分解するために、まず入力され

た if-then ルールの条件部と結論部をそれぞれ選言標準形に変形する．任意の命題論理式は選言標準形に変形できることに注意されたい．次に，選言標準形に変形された if-then ルールの条件部と結論部の選言肢を一つずつ選びそれらを含意で結んだ if-then ルールをすべての組合せについて作ると「リテラルの連言  $\rightarrow$  リテラルの連言」の形の if-then ルールを複数得ることができる．もし選言標準形に変形したとき条件部と結論部の選言肢の数がそれぞれ  $M$  個， $N$  個であったとすると  $M \times N$  個の「リテラルの連言  $\rightarrow$  リテラルの連言」の形の if-then ルールが得られる．例えば，条件部と結論部を選言標準形に変形した結果が  $(A \wedge \neg B) \vee C \vee (D \wedge E \wedge F) \rightarrow \neg G \vee (H \wedge I)$  であれば，条件部と結論部の選言肢を組み合わせることで  $A \wedge \neg B \rightarrow \neg G$ ， $A \wedge \neg B \rightarrow H \wedge I$ ， $C \rightarrow \neg G$ ， $C \rightarrow H \wedge I$ ， $D \wedge E \wedge F \rightarrow \neg G$ ， $D \wedge E \wedge F \rightarrow H \wedge I$  という 6 個の if-then ルールを得ることになる．このように分解された if-then ルールは一時的に短期記憶に保存される．

以上の前処理を行った後，以下の手順を分解された if-then ルールすべてについて実行する．本節ではこれ以後分解された if-then ルールのそれぞれを単に学習データと呼ぶ．まず，学習データ  $k_i$  と長期記憶の if-then ルール  $k_m$  との距離  $d(k_i, k_m)$  を計算する．もし， $k_i$  と  $k_m$  の条件部と結論部の肯定命題及び否定命題の数が異なっていれば  $d(k_i, k_m) = \infty$  とする．例えば， $k_i$  が  $A \wedge \neg B \rightarrow C$  で， $k_m$  が  $D \wedge E \rightarrow F$  のときは， $k_i$  の条件部の肯定命題が 1 個，否定命題が 1 個なのに対し， $k_m$  の条件部の肯定命題が 2 個，否定命題が 0 個であるので， $d(k_i, k_m) = \infty$  とする．一方， $k_i$  と  $k_m$  の条件部と結論部の肯定命題及び否定命題の数が完全に一致するとき（例えば， $k_i$  が  $A \wedge \neg B \rightarrow C \wedge \neg D$  で， $k_m$  が  $E \wedge \neg F \rightarrow G \wedge \neg H$  のとき）には， $d(k_i, k_m)$  を次式で計算する．

$$d(k_i, k_m) = \frac{1}{\sqrt{D}} \left\{ \min_{\sigma_1 \in S_M} \sum_{i=1}^M \|P_i - P'_{\sigma_1(i)}\| + \min_{\sigma_2 \in S_N} \sum_{i=1}^N \|Q_i - Q'_{\sigma_2(i)}\| + \min_{\sigma_3 \in S_{\tilde{M}}} \sum_{i=1}^{\tilde{M}} \|\tilde{P}_i - \tilde{P}'_{\sigma_3(i)}\| + \min_{\sigma_4 \in S_{\tilde{N}}} \sum_{i=1}^{\tilde{N}} \|\tilde{Q}_i - \tilde{Q}'_{\sigma_4(i)}\| \right\} \quad (1)$$

ここで， $S_M, S_N, S_{\tilde{M}}, S_{\tilde{N}}$  はそれぞれ  $M$  次， $N$  次， $\tilde{M}$  次， $\tilde{N}$  次の対称群， $D$  はパターン情報を表現するベクトルの次元であり，また，学習データを

$$\left( \bigwedge_{i=1}^M P_i \right) \wedge \left( \bigwedge_{j=1}^N \neg Q_j \right) \rightarrow \left( \bigwedge_{i=1}^{\tilde{M}} \tilde{P}_i \right) \wedge \left( \bigwedge_{j=1}^{\tilde{N}} \neg \tilde{Q}_j \right) \quad (2)$$

長期記憶の if-then ルールを

$$\left( \bigwedge_{i=1}^M P'_i \right) \wedge \left( \bigwedge_{j=1}^N \neg Q'_j \right) \rightarrow \left( \bigwedge_{i=1}^{\tilde{M}} \tilde{P}'_i \right) \wedge \left( \bigwedge_{j=1}^{\tilde{N}} \neg \tilde{Q}'_j \right) \quad (3)$$

とおいた．

このように距離計算を行った後，学習データと最も距離の小さい長期記憶の知識（第 1 勝者）と 2 番目に小さい知識（第 2 勝者）を見つけ（条件部と結論部の肯定命題及び否定命題の数が学習データと異なる if-then ルールは距離が  $\infty$  であるから，第 1 勝者や第 2 勝者になることはない），本節の最後に述べる方法で学習データが長期記憶にある知識と同じクラスタに属するか判定される．もし長期記憶の知識と同じクラスタには属しないと判定されれば，学習データを新たに長期記憶に追加し，その学習データについての処理を終了する．もし既存の知識と同じクラスタに属すると判定されたなら，学習データの類似知識が既に長期記憶にあると判断して長期記憶への追加はせず，次のように長期記憶を更新する．まず，学習データと最も距離の小さい長期記憶の知識（第 1 勝者）と 2 番目に小さい知識（第 2 勝者）の間にエッジがなければ，それらと同じクラスタとするためそれらの間に年齢 1 のエッジを生成する．もし既にエッジが存在していればそのエッジの年齢を 1 に戻す．第 2 勝者以外と第 1 勝者との間にエッジがあればそれらのエッジの年齢を 1 増やす．このとき  $\Lambda_{edge}$  を超えたエッジはすべて削除される．学習回数がパラメータ  $\lambda$  の整数倍であるならば，エッジをもたないか 1 本しかもたない知識を削除する．

本節の最後に，入力された学習データが既存の知識と同じクラスタに属するかを判断する方法を述べる．学習データが入力されると，第 1 勝者及び第 2 勝者の

それぞれのクラスタリング距離しきい値  $s$  を次式で計算する .

$$s = \begin{cases} \max_{k \in N} d(k, k_w) & (\text{if } N \neq \emptyset) \\ \min_{k \in A} d(k, k_w) & (\text{if } N = \emptyset) \end{cases} \quad (4)$$

ここで  $N$  は第 1(2) 勝者とエッジで結ばれている知識の集合,  $A$  は長期記憶に存在するすべての知識の集合,  $k_w$  は以前の手順で見つけた第 1(2) 勝者の知識を表す . すなわち, 第 1 勝者のクラスタリング距離しきい値を求める際には,  $k_w$  を第 1 勝者として計算し, 第 2 勝者のクラスタリング距離しきい値を求める際には,  $k_w$  を第 2 勝者として計算するというのである . もしも学習データと第 1 勝者の距離が第 1 勝者のクラスタリング距離しきい値よりも小さく, かつ第 2 勝者についても同様のことが成り立てば, 学習データは第 1 勝者及び第 2 勝者と同じクラスタであると判断する . 第 1 勝者または第 2 勝者のクラスタリング距離しきい値よりも入力ベクトルまでの距離が離れていれば, 入力ベクトルは既存のノードとは同じクラスタではないと判断する .

#### 4.2 推論フェーズ

ここでは, 図 4 のフローチャートで表される推論フェーズのアルゴリズムの詳細を述べる .

推論フェーズではパターン情報に連言, 選言, 否定を付与した論理式がファクトとして入力される . ファクトが入力されると, 学習フェーズで長期記憶に保存された if-then ルールと入力されたファクトを利用して推論を行い, 入力したファクトを根とする OR 木が出力される . この木の各節はリテラルの連言を保持し

ており, これらを選言で結んで作られる複数の選言標準形を推論結果とみなすことができる . 以下にアルゴリズムの詳細を述べる . なお, 学習フェーズと同様に, 入力されたファクトは複数のリテラルの連言の形に前処理として分解される . 以下で述べるアルゴリズムはこのような形に分解された後, 個別のリテラルの連言の形の命題について行う .

まず, 上記のように分解したファクトを根とし, それ以外の節はもたない木を生成する . 次に, 長期記憶に蓄えられている if-then ルールの条件部とファクトとの距離を計算し, しきい値  $\delta_r$  以下の距離になった if-then ルールの結論部のみを先の木に入力されたファクトの子ノードとして追加する . ここで, ファクト  $f$  と if-then ルールの条件部  $c$  の距離  $d(f, c)$  は次のように計算する . もし,  $f$  と  $c$  の肯定命題及び否定命題の数が異なっていれば  $d(f, c) = \infty$  とする . 例えば,  $f$  が  $A \wedge B$  で,  $c$  が  $C \wedge \neg D$  であれば  $d(f, c) = \infty$  とする . 一方,  $f$  と  $c$  の肯定命題及び否定命題の数が一致するとき (例えば  $f$  が  $A \wedge \neg B$  で  $c$  が  $C \wedge \neg D$  のとき) は  $d(f, c)$  を次式で計算する .

$$d(f, c) = \frac{1}{\sqrt{D}} \left\{ \min_{\sigma_1 \in S_M} \sum_{i=1}^M \|P_i - P'_{\sigma_1(i)}\| + \min_{\sigma_2 \in S_N} \sum_{i=1}^N \|Q_i - Q'_{\sigma_2(i)}\| \right\} \quad (5)$$

ここで, ファクトを

$$\left( \bigwedge_{i=1}^M P_i \right) \wedge \left( \bigwedge_{j=1}^N \neg Q_j \right) \quad (6)$$

if-then ルールの条件部を

$$\left( \bigwedge_{i=1}^M P'_i \right) \wedge \left( \bigwedge_{j=1}^N \neg Q'_j \right) \quad (7)$$

とおいた . ただし, 同一のクラスタから複数の if-then ルールを追加することを避けるため, しきい値を下回る if-then ルールが同じクラスタに複数ある場合は距離の最も小さい if-then ルールのみが子ノードに追加される . もし, しきい値以下の距離になるような if-then ルールが長期記憶に存在しない場合は処理は終了する . その後, 追加された子ノードを入力されたファクトと同様に扱い, 次々に子ノードを生成していく . この手順は, 入力されたファクトと長期記憶の断片的な知識

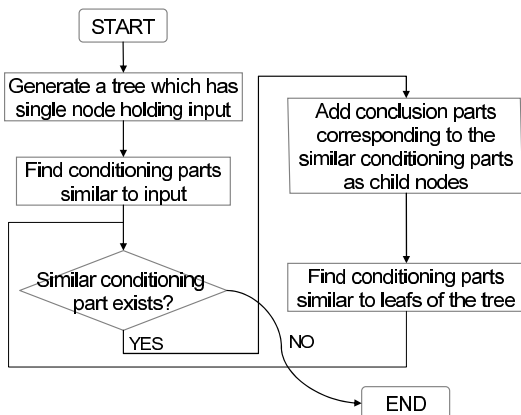


図 4 推論フェーズのフローチャート  
Fig. 4 The flowchart of the reasoning phase.

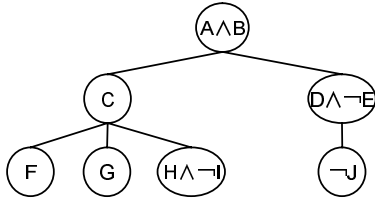


図 5 出力される OR 木の例  
Fig. 5 An example of OR-tree.

を組み合わせると多段の推論を行っていることに相当する．例として，長期記憶として  $A \wedge B \rightarrow C \vee (D \wedge \neg E)$ ， $C' \rightarrow F \vee G \vee (H \wedge I)$ ， $D' \wedge \neg E' \rightarrow \neg J$  を保持しているときに  $A' \wedge B'$  がファクトとして入力されたときに出力される木を図 5 に示した．ここで  $(A, A')$ ， $(B, B')$ ， $(C, C')$ ， $(D, D')$ ， $(E, E')$  はしきい値に対して十分に距離が小さいパターン情報のペアである．

出力される木の各節にはデータとしてリテラルの連言が保持されているが，これらは単独では推論結果として正しくない．なぜならば，各節はもともと選言で結ばれたものを分解して学習されたものだからである．図 5 で例えば  $A \wedge B \rightarrow C \vee (D \wedge \neg E)$  なのであって， $A \wedge B \rightarrow C$  ではない．そこで，出力された木を OR 木として扱い，節を選言で結んだ選言標準形が推論結果として正しいものであるとみなす．例えば図 5 の木が出力された場合は，推論結果が  $C \vee (D \wedge \neg E)$ ， $C \vee \neg J$ ， $F \vee G \vee (H \wedge I) \vee (D \wedge \neg E)$ ， $F \vee G \vee (H \wedge I) \vee \neg J$  であつたとみなす．このように，推論結果は選言標準形をしているが，これは提案手法が可能性についての推論を行っていると同解釈することができる．例えば現在地に関する情報をファクトとして入力した結果，出力として「スーパーまたはコンビニエンスストアまたは薬局」に相当する結果を得たとしよう．これは現在地からどのような場所に移動できるかという可能性を推論によって発見したと考えられる．

#### 4.3 ニューラルネットワークモデルとしての解釈

本手法は知識工学の視点では図 2 のようなアーキテクチャであるとみなせるが，SOINN-AM [5] を拡張した図 6 のようなニューラルネットワークモデルとしての解釈も可能である．このニューラルネットワークモデルは，SOINN-AM のように学習データの入力に応じて，同じクラスであることを表現するエッジでノードが結合されながら自律的にノードが増えていくモデルであるが，ノードによるデータの表現方法が SOINN-AM と異なる．以下に提案手法の自己増殖型

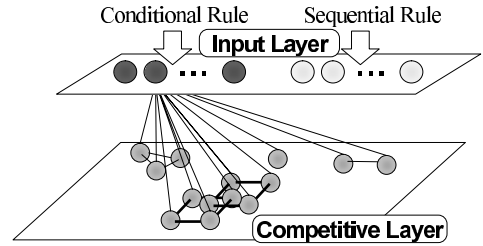


図 6 提案手法のニューラルネットワークモデル  
Fig. 6 The neural network model of the proposed method.

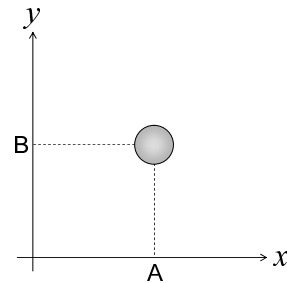


図 7  $A \rightarrow B$  を表現するノード  
Fig. 7 A node representing  $A \rightarrow B$ .

ニューラルネットワークが，どのようにノードによって含意（ならば），選言（または），連言（かつ）を表現しているかを以下に述べる．提案手法のニューラルネットワークは，新たな if-then ルールを学習することで，以下で述べたような形で if-then ルールを表現するノード群が自己組織的に増えていく．

##### 4.3.1 含意のみを含む if-then ルールの表現

SOINN-AM の一つのノードは連想対に対応する実数値ベクトルのペアを重みとして保持している．新たに学習すべき連想対が入力されたときは，SOINN-AM はこのノードを自己組織的に増やしていくことによって，学習すべき連想対を獲得していく．提案手法の  $A \rightarrow B$  といった形の if-then ルールの学習は，SOINN-AM が学習する連想対を，二つの命題を含意で結んだ if-then ルールと解釈し直して表現している．もし，パターン情報が一次元であり，ノードの保持するベクトルのペアをノードの位置する点の  $x$  座標及び  $y$  座標で表現することにすれば， $A \rightarrow B$  を表すノードは図 7 のようになる．推論フェーズにおいて  $A$  が入力されれば，このノードが発火し  $B$  を出力する．そのときには  $B$  を  $A$  の子ノードとして追加することになる．

##### 4.3.2 選言を含む if-then ルールの表現

選言を含む if-then ルールのノードによる表現は，

SOINN-AM が多対多の連想対を学習できることを利用している。SOINN-AM は複数のノードを利用することで多対多の連想対を学習しているが、上と同様に、提案手法は連想対を含意で結んだ if-then ルールと解釈し直すことで選言を含んだ if-then ルールを表現している。例えば連想キー A に対して B と C を連想結果として出力することを表現しているノードによって、提案手法は  $A \rightarrow B \vee C$  を表現している。図 7 のときと同様の設定で  $A \rightarrow B \vee C$ ,  $A \vee B \rightarrow C$  を表すノード群を図 8 に示した。

4.3.3 選言を含む if-then ルールの表現

提案手法では、選言を含む if-then ルールを表現するために、クラスタを表現するために SOINN-AM で用いられていたエッジに加えて新たなエッジを導入している。このエッジは抑制性をもっており、結合されている二つのノードが同時に発火しない限りそのノードの発火を抑えるものである。これによって二つの命題が同時に真になって初めて真になる選言を表現している。例えば、 $A \rightarrow B \wedge C$ ,  $A \wedge B \rightarrow C$  を表すノード群は、図 7 のときと同様の設定では図 9 のようになる。

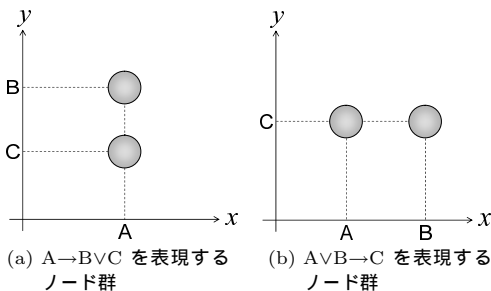


図 8 選言を含む if-then ルールを表現するノード群  
Fig. 8 Nodes representing if-then rules including disjunction.

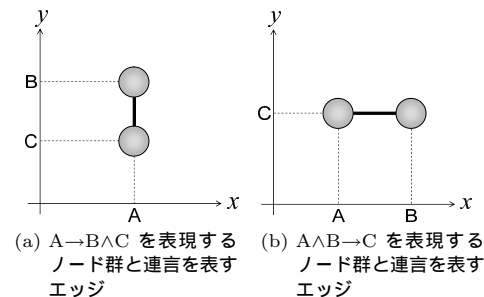


図 9 選言を含む if-then ルールを表現するノード群と選言を表すエッジ  
Fig. 9 Nodes and edges representing conjunction.

4.3.4 否定を含む if-then ルールの表現

上述のように、提案手法のニューラルネットワークは一つのノードが多次元ベクトルのペアを保持している。これらの多次元ベクトルで表されるパターン情報が肯定なのか否定なのかを表現するために、各ノードに肯定・否定かを表すフラグを二つ設けることで否定を含む if-then ルールを表現することが可能になる。

5. 実験

図 10 のような見取り図で表される建物から取得した実画像を用い、環境の因果関係や可能性について学習させ、学習した知識を用いて推論を行わせた。学習の方法は、バッチ学習ではなく追加学習で行った。学習時のパラメータは  $\Lambda_{edge} = 100, \lambda = 50$  とした。用いた画像は図 11 に示したような  $56 \times 46$  ピクセルの 14 種類の画像についてそれぞれアングルを変えて撮影した画像を各 20 枚ずつ合計 280 枚を用意した。20 枚の画像は、カメラのアングルをおよそ 1 度ずつ変化させながら撮影して得た。学習させた if-then ルールは「 $A \rightarrow B, B \rightarrow D \vee E, E \rightarrow (C \wedge N) \vee F \vee M, (C \wedge N) \rightarrow (G \wedge I) \vee (H \wedge J), (G \wedge I) \rightarrow K, (H \wedge J) \rightarrow L$ 」である。これらをシンボルで書くならば「閉じたドア → 開いたドア, 開いたドア → 壁 ∨ 廊下, 廊下 → (研究室 ∧ 表札) ∨ エレベータ ∨ 階段 (研究室 ∧ 表札) → (机 1 ∧ 閉まった引出し 1) ∨ (机 2 ∧ 閉まった引出し 2) (机 1 ∧ 閉まった引出し 1) → 開いた引出し 1 (机 2 ∧ 閉まった引出し 2) → 開いた引出し 2」である。学習データとして実際に入力されたデータは、各命題を表す 20 枚の画像をそれぞれランダムに 1 枚ピックアップしてできる if-then ルールであり、そのようにランダムに選択してできる if-then ルールの入力をそれぞれの if-then ルールについて 4400 回繰り返した。例

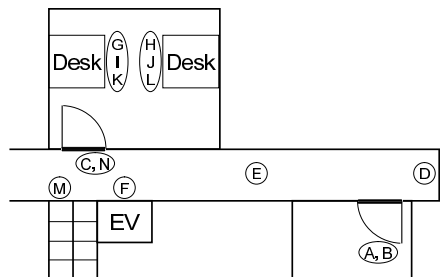


図 10 学習した環境の見取り図。円で囲まれたアルファベットは図 11 と対応している。  
Fig. 10 The layout of the building.



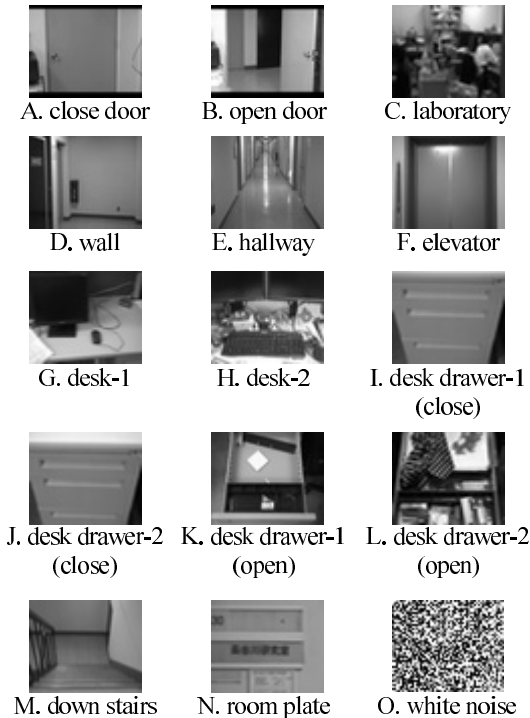


図 11 実験に用いた対象  
Fig. 11 Objects used in the experiment.

例えば  $A \rightarrow B$  を学習させる際には、A, B に対応する画像をそれぞれ 20 枚の画像から 1 枚ずつ選び、それを if-then ルールとして学習させることを繰り返した。

### 5.1 学習結果

学習によって 980 個の if-then ルールが長期記憶に蓄積され、それらは 10 個のクラスタに分類された。クラスタに属する if-then ルールはそれぞれ「 $A \rightarrow B$ ,  $B \rightarrow D$ ,  $B \rightarrow E$ ,  $E \rightarrow (C \wedge N)$ ,  $E \rightarrow F$ ,  $E \rightarrow M$ ,  $(C \wedge N) \rightarrow (G \wedge I)$ ,  $(C \wedge N) \rightarrow (H \wedge J)$ ,  $(G \wedge I) \rightarrow K$ ,  $(H \wedge J) \rightarrow L$ 」に対応する知識だけであった。これはシンボルベースで表せば単一のルールとなるような if-then ルールをうまくクラスタリングできていることを意味する。

表 2 に各クラスタが保持する if-then ルールの数 (クラスタのメンバ数) とデータ圧縮率を示した。データ圧縮率は「クラスタのメンバ数 ÷ 与えた学習データの種類の数」で評価した。学習データとして合計で 4000 種類の if-then ルールが与えられたが、そのうちの 24.5% にあたる、980 個の if-then ルールが長期記憶に蓄積されたことが表から分かる。つまり、75.5% にあたる 3020 個の if-then ルールは長期記憶には蓄積されなかったことになる。これは類似した知識を長期

表 2 クラスタのメンバ数と圧縮率

Table 2 The number of members in the clusters and compression rate.

Cluster No.	If-then rule	Number of members	Compression rate (%)
1	$A \rightarrow B$	87	21.75
2	$B \rightarrow D$	145	36.25
3	$B \rightarrow E$	50	12.5
4	$E \rightarrow (C \wedge N)$	64	16
5	$E \rightarrow F$	31	7.75
6	$E \rightarrow M$	146	36.5
7	$(C \wedge N) \rightarrow (G \wedge I)$	84	21
8	$(C \wedge N) \rightarrow (H \wedge J)$	111	27.75
9	$(G \wedge I) \rightarrow K$	140	35
10	$(H \wedge J) \rightarrow L$	122	30.5
Total	—	980	24.5

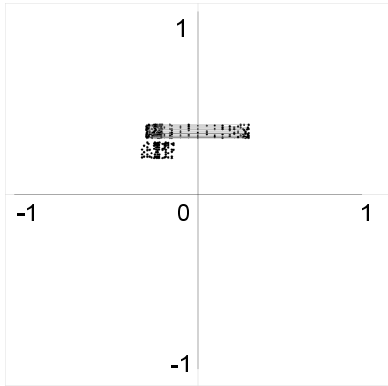
記憶に蓄積しないために導入したアルゴリズムがうまく働いていることを意味している。

ここで、上記の学習手順がバッチ学習ではなく追加学習になっていることに注意されたい。上記の手順で正しく知識を学習できたことは、いったん学習が終了した後に新たな学習データが追加されたとしても以前に学習した知識を忘却することなく新たな知識を学習することや、同じか極めて類似したデータの長期記憶への追加によるメモリの浪費を避けることが可能であることを示している。

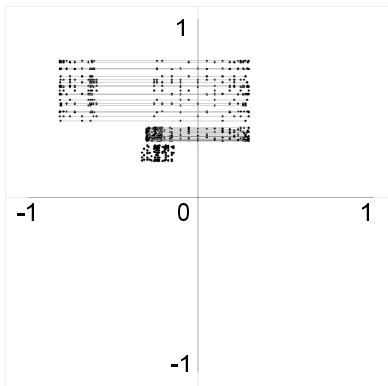
また、提案手法をニューラルネットワークとして解釈したとき、学習によって自己組織的に増殖するノードの様子を図 12 に示した。図 12 のようにノードを可視化するため、まず各ノードの保持する多次元ベクトルのペアを平均値を計算した。その後、図 7 のようにノードの位置の横軸と縦軸の座標を、そのノードの保持するベクトルの平均値とした。図 12 でのノード間のエッジは、クラスタを表現するためのエッジではなく、連言を表すためのエッジである。この図から、新たな知識を学習すると自己組織的にノードが増えていく様子が分かる。

### 5.2 推論結果

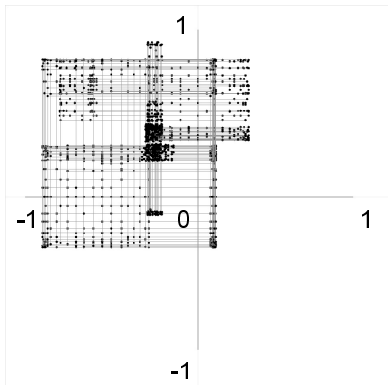
学習後、ファクトとして図 13(a) の画像を入力して  $\delta_r = 0.3$  として推論を行わせたところ、if-then ルールとして断片的に学習したデータを組み合わせて図 14 の OR 木が推論結果として出力された。これは入力したファクトと学習させた if-then ルールから得られる結果として不足も重複もない必要十分なものであるといえる。この結果は、閉じたドアから階段、エレベータ、引出し等に到達可能であることと、それらの場所に至るまでの経路をシステムが推論によって見出すこ



(a)  $A \rightarrow B$ ,  $H \wedge J \rightarrow L$  の順に追加的に学習



(b)  $A \rightarrow B$ ,  $H \wedge J \rightarrow L$  に加え,  $G \wedge I \rightarrow K$  を追加的に学習



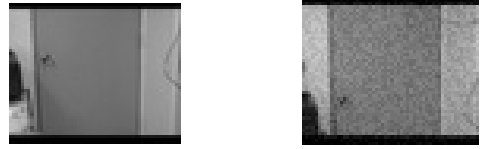
(c)  $A \rightarrow B$ ,  $H \wedge J \rightarrow L$ ,  $G \wedge I \rightarrow K$  に加え, その他すべてのデータを順に追加的に学習

図 12 自己組織的に増殖するノードの様子

Fig. 12 Nodes growing in learning phase.

とができたと解釈できる。

引出しに関する推論の部分では提案手法が連言を扱うことができることが有効に働いている。もし連言を扱えないと、机の画像を知識として与えずに閉じた引



(a) (b)

図 13 ファクトとして入力した画像

Fig. 13 The images for facts.

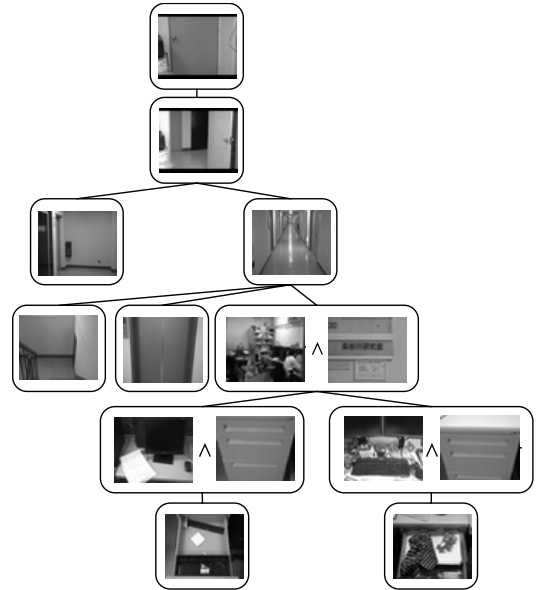


図 14 雑音のない画像をファクトとして与えたときの推論結果

Fig. 14 The reasoning result when the noiseless image was provided as the fact.

出しの画像だけを学習することになり、異なる引出しでも見た目は同じであるため正しく推論を行うことができない。

また、推論結果に重複がないのは if-then ルールをクラスタリングしながら学習できたためである。もしクラスタリング結果を無視して、しきい値よりも小さい if-then ルールの結論部をすべて節として追加するようにして推論を行わせたところ、3,804,840,166 個もの節が生成された。これらの節のうち、根の子ノードである節のもつ画像はすべて図 11 の B に対応する 20 枚の画像に一致し、シンボルベースで解釈するならば一つの節で表現すべき内容であった。この結果は if-then ルールをクラスタリングすることの有効性を示している。

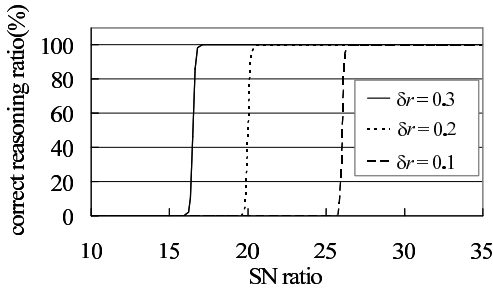


図 15 雑音を加えられたファクトに対して正しく推論できた割合

Fig. 15 The ratio of correct reasoning when noisy data given.

### 5.3 雑音耐性と汎化性能

図 1(a) のタイプの雑音への耐性を確かめるため、閉じたドアを異なったアングルで撮影した画像 20 枚にガウス雑音を加えたものを同一の雑音レベルについて 10 枚ずつ用意し、それらの画像をファクトとして与えた。各雑音レベルについて“正しく推論した回数 ÷ 推論を試みた回数”をプロットしたところ、図 15 のような結果を得た。図 15 の横軸は下式で定義される SN 比であり、数値が大きいほど雑音が少ないことを意味する。

$$\text{SN 比} = 10 \log_{10} \left( \frac{S}{\sigma} \right)^2 \quad (8)$$

ここで  $S$  は入力画像の各ピクセルがとり得る値の幅であり、本実験では  $S = 2$  である。また  $\sigma$  は加えたガウス雑音に従う正規分布の偏差である。図 15 に示した結果は、ある SN 比を境界にして、それより雑音の多い画像は未知画像だと判断されて推論が行われず、雑音が少ない画像に対しては原画像のときと同様の正しい推論を行っていることを意味する。本手法は  $\delta_r$  の設定で境界になる SN 比を調整できる柔軟性をもつ。パターン推論機の先行研究である [3] はこのようなことを行えない。参考までに、雑音を加えられた図 13(b) に示されている画像をファクトとして入力した場合に出力された OR 木を図 16 に示す。図 14 と図 16 を比較すると、雑音のある画像をファクトとしてもその SN 比が一定の値よりも大きければ、雑音のないファクトからの推論と同様の結果を得られることが分かる。なお、図 15 では  $\delta_r$  を大きくすればするほど、より雑音の多い画像に対して正しい推論が行えるという結果になっているが、 $\delta_r$  を大きくしすぎると、本来異なるはずの画像を正しいものと認識して推論を失敗してし

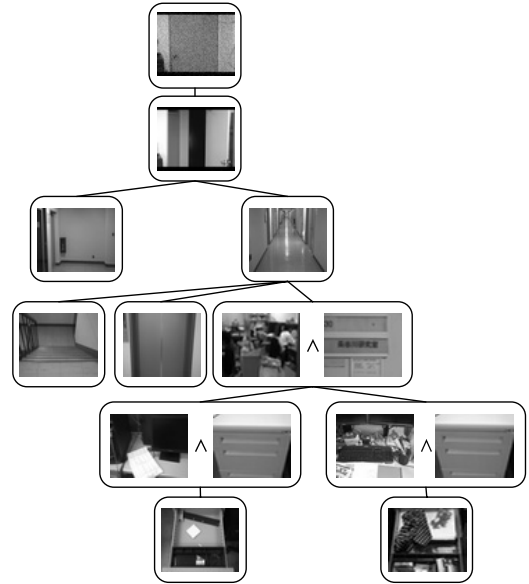


図 16 雑音を加えられた画像をファクトとして与えたときの推論結果

Fig. 16 The reasoning result when the noisy image was provided as the fact.



図 17 テストデータとして入力した画像  
Fig. 17 Images used for test data.

まうため注意が必要である。実際、 $\delta_r \geq 0.4$  のときはこの現象が起きてしまうため、雑音の多寡によらず正しい推論を行うことができない。

次に、図 1(b) のタイプの雑音への耐性を確かめるため、図 11 M の画像を入力したところ、システムは入力画像を保持するノードを根とし、それ以外のノードをもたない木を出力した。これは、学習していない画像を未知画像として適切に判断できていることを示している。[3] の手法では、学習していないデータを入力すると、無意味なデータが出力されてしまう。

なお、図 1(a) のタイプの雑音の入力に対して正しい推論が行えることは、本システムが汎化性能をもっていることにほかならない。実際 [3] においては、図 1(a) のタイプの雑音の入力に対して正しく推論ができる実験を行うことで、汎化性能の有無を議論して

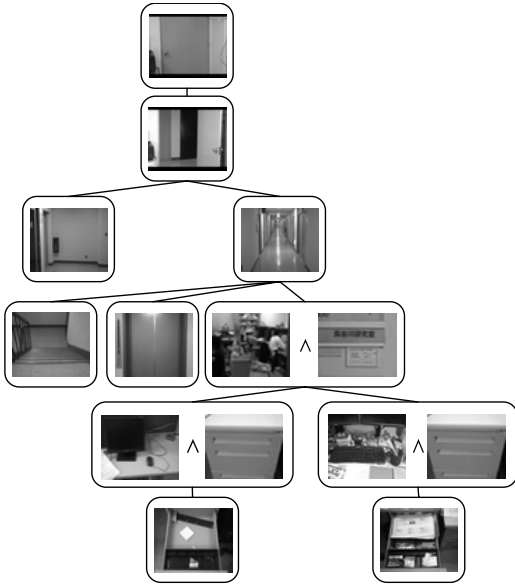


図 18 テストデータに対する推論結果  
Fig. 18 Reasoning result for test data.

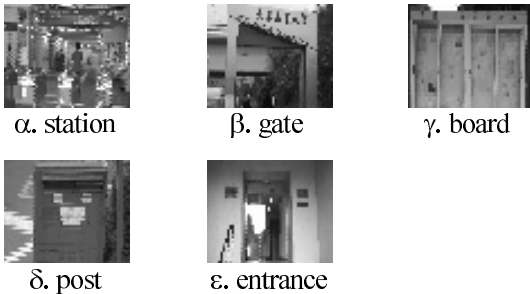


図 19 新たに学習させた画像.  
Fig. 19 The images newly provided to the system.

いる。念のため、 $\delta_r = 0.3$  のもとで、学習データと類似してはいるが異なる画像である図 17(a) を、テストデータとして入力し、推論させてみたところ図 18 に示すような正しい推論結果を得た。ただし、学習データと同じ対象を撮影した画像であっても、アングルがある程度以上異なると望ましい推論は行えない。例えば、 $\delta_r = 0.3$  のもとで、図 17(b) の画像をファクトとして入力した場合の推論結果は、入力した画像を保持するノードが生成されたのみであった。

#### 5.4 追加学習性能の確認

本手法の追加学習性能を確実に示すため、これまでの節で提示した実験に伴う学習や推論を行った後のシステムに対して、別の環境から得られた if-then ルー

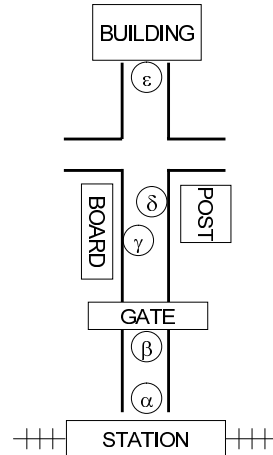


図 20 新たに学習させた環境の見取り図。円で囲まれたギリシャ文字は図 19 と対応している。

Fig. 20 The layout of the environment newly provided to the system.

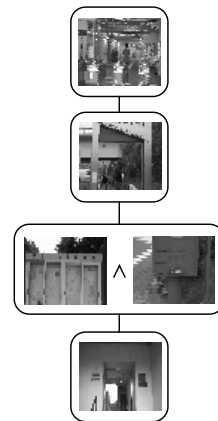


図 21 新たに学習させた環境についての推論結果  
Fig. 21 Reasoning result for the new environment.

ルを更に追加学習することを試みた。学習に用いた画像は図 19 に示したような  $56 \times 46$  ピクセルの 5 種類の画像であり、それぞれの種類について角度を変えて 20 枚ずつ画像を撮影した。20 枚の画像は、カメラのアングルをおよそ 1 度ずつ変化させながら撮影して得た。画像を取得した環境は図 20 のような環境である。

学習させた if-then ルールは「 $\alpha \rightarrow \beta, \beta \rightarrow \gamma \wedge \delta, \gamma \wedge \delta \rightarrow \epsilon$ 」である。これらのルールをシンボルベースで書くならば、「駅  $\rightarrow$  校門, 校門  $\rightarrow$  掲示板  $\wedge$  ポスト, 掲示板  $\wedge$  ポスト  $\rightarrow$  エントランス」となる。学習回数やパラメータは図 10 の環境を学習する際に用いたものと同じとした。学習はバッチ学習でなく、追加

学習で行った。

学習後、新たに学習した環境から得られた画像である図 19  $\alpha$  をファクトとして推論を行わせたところ、図 21 に示す推論結果を得た。また、最初に学習した環境から得られた画像である図 13 (a) をファクトとして推論を行わせたところ、図 14 の推論結果を得た。これらは推論結果として妥当なものである。この結果は、既存の知識を用いた推論を何度か行った後でも、提案手法が新たな環境を追加学習できることを示している。

## 6. 今後の課題

### 6.1 パラメータ及び学習回数の自動決定

本手法は、パラメータ  $\Lambda_{edge}, \lambda, \delta_r$  及び、学習回数を事前に設定しておく必要がある。 $\Lambda_{edge}, \lambda$  や学習回数を適切に設定しないと if-then ルールの学習がユーザの望みどおりに行えず、また、 $\delta_r$  の設定を適切に行わないと推論を失敗してしまう。これらのパラメータの適切な値は学習する対象によって異なることが分かっているが、学習する対象がシステムに与えられたとき、その対象に対する適切なパラメータは、現時点では試行錯誤によって探すことになる。例えば、本論文で提示した実験においては、 $\Lambda_{edge} = 100, \lambda = 50$ 、学習回数を 4400 回とすれば与えられた if-then ルールを正しく学習することができ、 $\delta_r$  を 0.1 から 0.3 とすれば正しく推論を行うことができたが、これらの値は試行錯誤によって見出したものであり、これらのパラメータがすべての問題に対して適用できるわけではない。パラメータを試行錯誤によってではなく、学習対象によって自動的に決定する手法を見つけることは今後の課題として重要であると考えている。

### 6.2 学習フェーズにおける雑音

本論文の実験では、推論フェーズにおける雑音への耐性を確認したが、学習フェーズにおける雑音については実験を行っていない。学習フェーズに図 1 (a) のタイプの雑音が存在すると、雑音の混じったデータが長期記憶に蓄積されてしまい、推論フェーズにおいて雑音の混じった形で推論結果が出力されてしまうと考えられる。一方、学習フェーズに図 1 (b) のタイプの雑音が存在する場合にはある程度の耐性をもつことが予想される。というのも、学習時に図 1 (b) のタイプの雑音が入力されたとき、いったんはそのデータが長期記憶に蓄積されるが、もしも雑音が入力される頻度が正しい学習データが入力される頻度比べて十分に少

ない場合は、雑音である知識がエッジをもつことはなく、学習回数が  $\lambda$  の整数倍のときに実行される、エッジをもたないか 1 本しかもたないデータを削除するというステップにおいて、そのデータが削除されると期待されるからである。学習フェーズにおける図 1 (a) のタイプの雑音への耐性を実現すること、及び、学習フェーズにおける図 1 (b) のタイプの雑音にどの程度の耐性があるかを確認する実験を行うことは今後の課題である。

## 7. むすび

非定常な実環境下で自律的に活動することが要求されるシステムにとって従来のシンボルベースの推論機が不十分であることを踏まえ、パターン情報ベースの推論機の提案を行った。提案手法は、パターン情報を命題としそれらの命題に連言・選言・否定を付与した任意の形の if-then ルールを学習することができる。また学習した知識を利用して多段の推論を行うことができる。これら推論機としての基本的な機能以外にも if-then ルールの追加学習、不要な知識の削除、if-then ルールのオンラインでのクラスタリングを実現している。if-then ルールのクラスタリングは推論結果の重複や計算量の爆発を防ぐために重要であるだけでなく、システムの内部的なシンボル獲得につながる可能性もある。

今後は、6. で述べた課題を解決するとともに、提案手法を知能ロボットに応用することを考えている。提案手法により、環境から自律的にパターン情報ベースの if-then ルールを学習し、その知識をもとにした推論によってタスクを解決する知能ロボットが開発できるものと考えている。

謝辞 本研究の実施にあたり NEDO 産業技術研究助成事業から支援を頂きました。記して感謝致します。

## 文 献

- [1] 新田克己, 知識と推論, サイエンス社, 2002.
- [2] 月本 洋, “パターン推論—ニューラルネットワークの論理的推論,” 信学論 (D-II), vol.J83-D-II, no.2, pp.744–753, Feb. 2000.
- [3] 山根 健, 蓮尾高志, 末光厚夫, 森田昌彦, “軌道アトラクタを用いたパターンベース推論,” 信学論 (D), vol.J90-D, no.3, pp.933–944, March 2007.
- [4] F. Shen and O. Hasegawa, “An incremental network for on-line unsupervised classification and topology learning,” Neural Netw., vol.19, no.1, pp.90–106, 2006.
- [5] A. Sudo, A. Sato, and O. Hasegawa, “Associative

memory for online incremental learning in noisy environments,” Proc. 2007 International Joint Conference on Neural Networks (IJCNN’07), 2007.

- [6] B. Fritzke, “A growing neural gas network learns topologies,” Advances in Neural Information Processing Systems (NIPS), pp.625–632, 1995.

(平成 19 年 7 月 3 日受付, 12 月 20 日再受付)



須藤 明人 (学生員)

2002 早大・理工・物理卒。2004 同大大学院物理学及応用物理学専攻修士課程了。現在, 東工大大学院知能システム科学専攻博士後期課程に在学中。ニューラルネットワーク, ヒューマノイドロボティクス, 信号処理に興味をもつ。



張 晨犁

2006 清華大・工・自動車工学卒。現在, 東工大大学院知能システム科学専攻修士課程在学中。時系列データの認識に興味をもつ。



坪山 学

2007 東京農工大・電気電子工学卒。現在, 東工大大学院知能システム科学専攻修士課程在学中。ニューラルネットワークとヒューマノイドロボティクスに興味をもつ。



佐藤 彰洋 (学生員)

2006 電通大・電気通信・情報工学卒。現在, 東工大大学院知能システム科学専攻修士課程在学中。ニューラルネットワークとヒューマノイドロボティクスに興味をもつ。



長谷川 修 (正員)

1993 東京大学大学院博士課程了。博士(工学)。同年電子技術総合研究所入所。1999 年 6 月より 1 年間カーネギーメロン大学ロボティクス研究所滞在研究員。2001 産業技術総合研究所主任研究員。2002 年 5 月東京工業大学大学院理工学研究科付属像情報工学研究施設助教授。2002 から 3 年間科技団さきがけ研究 21 研究員。パターン認識, ニューラルネット, 認知ロボティクスなどの研究に従事。人工知能学会, 日本認知科学会, 日本顔学会, IEEE CS 等各会員。