人　工　知　能　74－4
（1991．　1．　16）

環境適応のためのチャンキングメカニズム

早勢欣和　畝見達夫　川田重夫
長岡技術科学大学
hayase@voscc.nagaokaut.ac.jp

安定な状態変化を示す環境に適応するための学習メカニズムのモデルを構築し、簡単な試験例題への応用を通して検討を行う。入力素子からのデータの集合で表される状態の遷移における各データ間の単純遷移関係に重みをつけ、値を経験に依存した方法で変更し、小さなものは忘却することにすれば、必要なものだけが記憶中に保持されることになる。記憶方式として単純遷移関係を記憶する関係記憶と最近の時系列データを記憶する作業記憶を用いる。必要に応じて、データの空間方向の抽象化やチャンキングを行うなどして記憶データの構造を変更し、環境に適応していく。

A chunking mechanism to adapt to an environment

Yoshikazu Hayase　Tatsuo Unemi　Shigeo Kawata
Nagaoka University of Technology
hayase@voscc.nagaokaut.ac.jp

This paper proposes a learning mechanism to adapt to an environment which shows a stable state transition controlled by some rule. The learner memorizes an experiential state transition represented by a set of relations with weights in a relation memory and in a working memory in order of time. The weight is changed in accordance with experience and the learner forgets a relation which has a low weight. Referring two memories, the leaner associates a state confronting with some state, makes a plan, and makes macros if needed.

1 はじめに

人間をはじめとする幾つかの生体は、特定の環境の中においてのみ見られるわけではなく、全く異なる環境においても生息している。あまりにも様々な環境が存在することから、そうした生体が生得的に生活する環境についての知識を持っているとは考え難い。また、個々の持つ感覚受容器などの各器官の機能は同じであっても、その精度は遺伝などの影響により個体差があるものと考えられるので、仮にあらかじめ知っていたとしてもそれが使えるといった絶対的な保障はない。すなわち、生体が環境の中で生きていくためには、環境にうまく適応しなければならないわけである。環境適応ための一つの方法として、チャンキングをあげることができる。例えば、目標地点までの到達の問題について考えたとき、目標地点に行くまでには様々な状態の変化が見られることになるが、途中に障害物などが無いとわかれば、そうした全ての状態変化をいちいち確認する必要はないはずである。このような場合、一目散に進むといった風にチャンキングを行い、それを記憶しておけば、以後同じ状況であればすぐに到達できるようになるであろう。チャンキングを行うためには環境における空間や時間の方向のデータのまとまりを学習する必要がある。

本論文では、こうした問題を踏まえて、記憶のデータ構造を変化させることのできる学習メカニズム、すなわちデータの空間方向、時間方向の表現単位の抽象化を行うといったメカニズムを提案する。なお、プリミティブな知識の表現形式が環境についての一種の知識とも言えなくもないが、本論文ではこれを生得的なものとして固定することにし、また、学習の対象となる環境は、なんらかのルールにより安定な状態変化を示すものとする。

2 学習メカニズム

学習モデルの概要を図 2-1 に示す。外界および内界の状態は感覚受容器から入力され、経験した状態遷移関係が作業記憶、関係記憶に記憶される。記憶中には必要なものは保持され続けるが、そうでないものは忘却することにする。経験に則して、必要であればデータの抽象化を行うなどして記憶のデータ構造を変更すべく制御機構が働く。作業記憶には推論結果やプランなども記憶されており、その内容に応じて効果器に命令が出される。

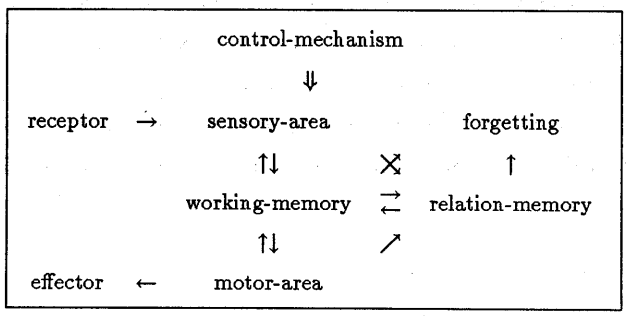

図 2-1: 学習モデル

2.1 記憶モデル

外界や内界の状態 S は、目などの感覚受容器からの入力データ y、及びそれまでの遷移経過 f を値として持つノード n の集合により表される。状態は要因 x 、例えば、前に1歩進むといったような状態遷移を引き起こすものによって次の状態に遷移する。関係記憶 $\mathcal{RM}$ には、こうした状態遷移における各ノード間の遷移を表す f に重みを付加した関係 r が記憶される。作業記憶 $\mathcal{WM}$ には重みを伴う最近に経験した関係 wm の列が時間順に記憶され、また予想される状態などもここに時間順に記憶される。

$$\mathcal{RM} \subset \mathcal{R} \tag{2-1}$$

$$\mathcal{WM} \subset \{wm_1 wm_2 \cdots wm_n \mid wm_i = \langle r_i, w_{i1}, w_{i2}\rangle,\ r_i \in \mathcal{R},\ w_{i1}, w_{i2} \in [0,1]\} \tag{2-2}$$

$$\mathcal{S} \subset \mathcal{N} \tag{2-3}$$

$$\mathcal{R} = \{\langle f, w_1, w_2\rangle \mid f \in \mathcal{F},\ w_1, w_2 \in [0,1]\} \tag{2-4}$$

$$\mathcal{F} = \{\langle n_1, x, n_2\rangle \mid n_1, n_2 \in \mathcal{N},\ x \in \mathcal{X}\} \quad \mathcal{N} \times \mathcal{X} \to \mathcal{N} \tag{2-5}$$

$$\mathcal{N} = \{\{a_1, a_2, \ldots, a_n\} \mid a_i \in \mathcal{Y} \vee a_i \in \mathcal{F}\} \tag{2-6}$$

$$\mathcal{Y} = \{\langle receptor_{id}, data_{id}\rangle \mid \text{可能なデータの集合}\} \tag{2-7}$$

$$\mathcal{X} = \{x \mid \text{可能な状態遷移要因の集合}\} \tag{2-8}$$

ここで遷移関数 f を

$$f = \langle n_1, x, n_2\rangle \overset{\text{def}}{\Longleftrightarrow} \begin{cases} \text{ノード } n_1 \text{ から要因 } x \text{ により次にノード } n_2 \text{ になる} \\ \text{要因 } x \text{ によりノード } n_2 \text{ になる前はノード } n_1 \text{ である} \end{cases}$$

のように定義する。また関係 r を

$$r = \langle f, w_1, w_2\rangle \overset{\text{def}}{\Longleftrightarrow} \begin{cases} w_1 \text{ は } f \text{ における次のノードの決定に対する重み} \\ w_2 \text{ は } f \text{ における前のノードの決定に対する重み} \end{cases}$$

とし、作業記憶の記憶要素 wm を

$$wm = \langle r, w_1', w_2'\rangle \overset{\text{def}}{\Longleftrightarrow} \begin{cases} w_1' \text{ は } r \text{ の } w_1 \text{ に対する重み} \\ w_2' \text{ は } r \text{ の } w_2 \text{ に対する重み} \end{cases}$$

とする。各々の重みは、その遷移に対する確信度、信頼度として、あるいは印象度のように扱う。

2.1.1 記銘

感覚受容器から入力されてくるデータが状態 S_1 から要因 x によって状態 S_2 に遷移したとする。それぞれに含まれるノード間の遷移関数 f の集合 F

$$F = \{f \mid f = \langle n_i, x, n_j\rangle,\ n_i \in S_1,\ n_j \in S_2\}$$

を過去の記憶と照合して、必要な関係 r について以下に示す方法で重みを変更し記憶する。必要かどうかは、過去の経験、特に作業記憶 $\mathcal{WM}$ に記憶されている最近の経験の内容によって決定する。初めて経験したことや最近よく起きることに対してよく信じるといった、ヒューリスティクスを用いる。

2.1.2 重みの変更

状態が S_1 から要因 x によって S_2 に遷移したとする。このとき関係記憶$\mathcal{RM}$ に記憶されている関係 r

$$r = \langle f, w_1, w_2 \rangle \in \mathcal{RM}, \quad f = \langle n_i, x, n_j \rangle$$

について以下のように重みを変更する。

$$w_1^{t+1} = \begin{cases} g_1(w_1^t) & (n_i \in S_1) \wedge (n_j \in S_2) \\ g_2(w_1^t) & (n_i \in S_1) \wedge (n_j \notin S_2) \\ w_1^t & \text{otherwise} \end{cases} \qquad w_2^{t+1} = \begin{cases} g_1(w_2^t) & (n_i \in S_1) \wedge (n_j \in S_2) \\ g_2(w_2^t) & (n_i \notin S_1) \wedge (n_j \in S_2) \\ w_2^t & \text{otherwise} \end{cases} \tag{2-9}$$

ここで、w^t はそれまでの重みを、w^{t+1} は新しい重みをそれぞれ表している。また、関数 g_1 は非減少関数であり、関数 g_2 は非増加関数とする。初めて経験した関係についての重みは、あらかじめ設定してある初期値とする。また、抽象化を行う際にも、もととなる関係の重みを変更する。

ただし、幾つかのノード間遷移については過去に経験しており、

$$R_i = \{r_i \mid r_i = \langle f_i, w_1, w_2 \rangle,\ f_i = \langle n_{i1}, x, n_{i2} \rangle,\ r_i \in \mathcal{RM},\ n_{i1} \in S_1,\ n_{i2} \in S_2\}$$

は作業記憶に記憶されているが、

$$R_j = \{r_j \mid r_j = \langle f_j, w_1, w_2 \rangle,\ f_j = \langle n_{j1}, x, n_{j2} \rangle,\ r_j \in \mathcal{RM},\ n_{j1} \in S_1,\ n_{j2} \in S_2\} \neq R_i$$

は記憶されていないといった場合には、R_j の各関係について、

$$w_1^{t+1} = g_2(w_1^t), \quad w_2^{t+1} = g_2(w_2^t), \qquad (n_{1j} \in S_1) \wedge (n_{2j} \in S_2) \tag{2-10}$$

のように変更することにする。作業記憶要素 wm の関係 $r = \langle f, w_1, w_2 \rangle$ に対する重みの初期値 w_0 は、

$$w_{01} = g_3(w_1), \qquad w_{02} = g_3(w_2) \tag{2-11}$$

のように r の重みの関数により決定する。この重みは、時間の経過 (新しい遷移関係の経験) とともに減少していく。この関数を g_4 とする。

本論文では、関係 r の初期の重みは 3/4 とし、重みの変更は、

$$g_1(w) = w, \quad g_2(w) = \begin{cases} w - 1/4 & \text{if } w < 1 \\ w & \text{otherwise} \end{cases}, \quad g_3(w) = w, \quad g_4(w) = w - 1/4 \tag{2-12}$$

により行う。抽象化の際には、そのもととなる各関係は忘却する。また、重み w の値が 1 となるのは、抽象化などにより遷移関係に OR が生じる場合のみとする。

2.1.3 忘却

あらかじめ記憶し続けることができる重みの最小値を設定しておき、重みがその値よりも小さくなったものについては忘却することにする。また、抽象化の際に、そのもととなる関係は忘却することにする。本論文では、$w = 0$ となったものを忘却することにする。

2.2 推論メカニズム

2.2.1 連想

関係 $r = \langle f, w_1, w_2 \rangle$, $f = \langle n_1, x, n_2 \rangle$ におけるノード n_1, n_2 間の連想距離 ℓ を

$$\ell \stackrel{\text{def}}{=} 1 - w_1 \tag{2-13}$$

のように w_1 の関数として定義する。

連想における探索の深さは、連想可能な最長距離を設定することにより制限し、最長距離となった、あるいは自分自身を連想した場合、そのパスについてはそこで連想を止めることにする。また、目標状態などシステムに影響のあるものを連想した場合にはそのパスについてプランニングを行うようにする。距離の近いものほど早く連想されることになる。

2.2.2 プランニング

連想によって得られたパスのそれぞれの遷移について空間文脈を満たしているかどうかをチェックする。このとき、時間文脈についてもそれらを一つのノードしているので、空間文脈として扱うことができる。連想したものがシステムにとって陽の価値のものであれば空間分脈を満たすように要因の列を作成し、陰なるものであればそうならないようにする。

2.3 コントロールメカニズム

説明のために、任意のノード $n \in \mathcal{N}$ と要因 $x \in \mathcal{X}$ について以下のオペレータを定義する。

$$_{nx}\mathrm{F} \stackrel{\text{def}}{\Longleftrightarrow} \{f \mid r = \langle f, w_1, w_2 \rangle \in \mathcal{RM},\ f = \langle n, x, n_i \in \mathcal{N} \rangle\} \tag{2-14}$$

$$\mathrm{F}_{xn} \stackrel{\text{def}}{\Longleftrightarrow} \{f \mid r = \langle f, w_1, w_2 \rangle \in \mathcal{RM},\ f = \langle n_i \in \mathcal{N}, x, n \rangle\} \tag{2-15}$$

$$_{nx}\overset{\alpha}{\mathrm{N}} \stackrel{\text{def}}{\Longleftrightarrow} \{n_i \mid r = \langle f, w_1, w_2 \rangle \in \mathcal{RM},\ f = \langle n, x, n_i \in \mathcal{N} \rangle,\ w_1 \geq \alpha\} \tag{2-16}$$

$$\overset{\alpha}{\mathrm{N}}_{xn} \stackrel{\text{def}}{\Longleftrightarrow} \{n_i \mid r = \langle f, w_1, w_2 \rangle \in \mathcal{RM},\ f = \langle n_i \in \mathcal{N}, x, n \rangle,\ w_2 \geq \alpha\} \tag{2-17}$$

式 (2-14) は、ノード n と要因 x から次の状態を連想するために用いる遷移関数の集合で、式 (2-15) は、要因 x でノード n になる前の状態を連想するための遷移関数の集合である。また、式 (2-16) は、次の状態と信じられるノードの集合であり、式 (2-17) は、前の状態であると信じているノードの集合を表す。ここで α は、学習システムが遷移関係を信じる目安となる値とする。本論文では、$w \leq 3/4$ であれば信じるものとして説明する。

2.3.1 データの空間方向の抽象化

状態決定に必要な空間文脈をひとまとまりとして扱うために、新しくノードを作成するように制御する。

$$N_0 = \overset{1}{\mathrm{N}}_{xn}, \quad N_1 = (\overset{3/4}{\mathrm{N}}_{xn} - N_0) \neq \phi$$

という記憶状態のときに、

$$N_1 \cap N_2 = \phi$$

である N_2 から要因 x によりノード n となったとする。ここで、N_1 を値とするノード n_a と N_2 を値とするノード n_b を作成し、

$$\overset{3/4}{\mathrm{N}}_{xn} = \phi, \quad \overset{1}{\mathrm{N}}_{xn} = \{N_0, n_a, n_b\}$$

のように N_0, n_a, n_b のいずれかを空間文脈とするように記憶内容を変更する。例えば、

$$r_1 = \langle f_1, 3/4, 3/4\rangle, \quad f_1 = \langle n_1, x, n\rangle$$
$$r_2 = \langle f_2, 3/4, 3/4\rangle, \quad f_2 = \langle n_2, x, n\rangle$$
$$r_3 = \langle f_3, 3/4, 3/4\rangle, \quad f_3 = \langle n_3, x, n\rangle$$

という記憶状態のときに、状態 $S = \{n_4, n_5, n_6\}$ から要因 x によってノード n となったとすると以下のようになる。

$$r_4 = \langle f_4, 3/4, 0\rangle, \quad f_4 = \langle n_4, x, n\rangle, \quad r_1 = \langle f_1, 3/4, 0\rangle$$
$$r_5 = \langle f_5, 3/4, 0\rangle, \quad f_5 = \langle n_5, x, n\rangle, \quad r_2 = \langle f_2, 3/4, 0\rangle$$
$$r_6 = \langle f_6, 3/4, 0\rangle, \quad f_4 = \langle n_6, x, n\rangle, \quad r_3 = \langle f_3, 3/4, 0\rangle$$
$$r_7 = \langle f_7, 3/4, 1\rangle, \quad f_7 = \langle n_7, x, n\rangle, \quad n_7 = \{n_1, n_2, n_3\}$$
$$r_8 = \langle f_7, 3/4, 1\rangle, \quad f_8 = \langle n_8, x, n\rangle, \quad n_8 = \{n_4, n_5, n_6\}$$

2.3.2 過った空間方向の抽象化の修復

空間方向の条件と信じていたものに不必要なものが含まれていた場合、必要なものとそうでないものに分離するように制御を行う。

$$N_1 = \overset{1}{\mathrm{N}}_{xn} \neq \phi$$

という記憶状態のときに、

$$(n_1 \cap N_2) \neq n_1 \neq \phi, \quad n_1 \in N_1, \quad N_2 \subset \mathcal{N}$$

である N_2 から要因 x によってノード n となったとき、$(n_1 \cap N_2) \neq n_1 \neq \phi$ であるノード n_1 の値を $n_1 \cap N_2$ として記憶内容を変更し、$n_1 - (n_1 \cap N_2)$ の値については忘却することにする。例えば、2.3.1 において抽象化された記憶状態であるときに、状態 $S = \{n_1, n_2, n_9\}$ から要因 x でノード n となったとすると、以下のように変更される。

$$r_7 = \langle f_7, 3/4, 1\rangle, \quad f_7 = \langle n_{10}, x, n\rangle, \quad n_{10} = \{n_1, n_2\}$$
$$r_8 = \langle f_8, 3/4, 3/4\rangle, \quad f_8 = \langle n_9, x, n\rangle$$

2.3.3 データの時間方向の抽象化

状態決定に必要な時間文脈をひとまとまりとして扱うために、新しくノードを作成するように制御する。システムは環境を文脈に依存しない、つまり直前の入力データから次のデータが一意に決まるようなもの (*0-contextual*) として扱うが、それでは説明できなくなったときに一時的に *1-contextual* として1つ前の状

態との関係についても記憶することにする。すなわち遷移関数 f をノードの値とするわけである。遷移関数 f の値をノードとして扱うことにより、環境を再び *0-contextual* と扱うことができることになる。いま、

$$F_1 = \mathrm{F}_{x_0 n}, \qquad N_1 = {}_{nx}\overset{3/4}{\mathrm{N}} \neq \phi$$

という記憶状態のときに、

$$F_2 = \{f \mid f = \langle n_i, x_0, n\rangle,\ n_i \in \mathcal{N}\} \neq F_1$$

を経てノード n になり、さらに要因 x によって

$$N_1 \cap N_2 = \phi$$

である N_2 になったとする。このとき、一時的に *1-contextual* として

$$r = \langle f, 3/4, 3/4\rangle, \quad f = \langle n_{j1}, x, n_{j2}\rangle, \quad n_{j1} = \{f_j\}, \quad n_{j2} \in N_j, \quad f_j \in F_j$$

となる各関係を作成する。ノード n_{j1} の集合を N_{j1} で表すとすると、

$$\overset{3/4}{\mathrm{N}}_{xn_j} = N_{j1}, \quad n_j \in N_j$$

となるように記憶内容を変更し、*0-contextual* とする。例えば、

$$r_1 = \langle f_1, 3/4, 3/4\rangle, \quad f_1 = \langle n_0, x_0, n_1\rangle, \qquad r_2 = \langle f_2, 3/4, 3/4\rangle, \quad f_2 = \langle n_1, x_1, n_2\rangle$$

という記憶状態のときに、ノード n_3 から要因 x_0 によってノード n_1 となり、

$$r_3 = \langle f_3, 3/4, 1/2\rangle, \quad f_3 = \langle n_3, x_0, n_1\rangle, \qquad r_1 = \langle f_1, 3/4, 1/2\rangle$$

次に要因 x_1 によってノード n_4 となったとすると、以下のように変更される。

$$r_2 = \langle f_2, 3/4, 0\rangle, \qquad r_4 = \langle f_4, 3/4, 0\rangle, \quad f_4 = \langle n_1, x_1, n_4\rangle, \qquad n_5 = \{f_1\},\ n_6 = \{f_3\}$$
$$r_5 = \langle f_5, 3/4, 3/4\rangle, \quad f_5 = \langle n_5, x_1, n_2\rangle, \quad r_6 = \langle f_6, 3/4, 3/4\rangle, \quad f_6 = \langle n_6, x_1, n_4\rangle$$

この操作を繰り返すことによって、有限である k について *k-contextual* であるものも学習可能となる。

2.3.4 過った時間方向の抽象化の修復

時間方向の条件と信じていたものに不必要なものが含まれていた場合、必要なものとそうでないものに分離するように制御を行う。例えば、

$$n_1 * n_2 * n_3 \rightarrow n_4$$

という時間方向に抽象化された記憶があったとする。このとき、

$$n_5 * n_2 * n_3 \rightarrow n_4$$

となったとすると、ノード n_4 の決定において、順に n_3 と n_2 までのノードが同じであるので、

$$n_2 * n_3 \rightarrow n_4$$

のように変更する。

2.3.5 チャンキング

記憶中に保持されている単純遷移関係を順に追ってみたとき、一意に決定可能なパスが複数個連結しているものがあったとする。例えば、

$$n_1 \xrightarrow{x_1} n_2 \xrightarrow{x_2} n_3$$

のようにそれぞれ一意に決定できる場合、ノード n_1 の状態からひとまとまりの要因 $x_1 * x_2$ によってノード n_3 になると推論することができる。ノード n_3 が重要である場合など、必要に応じて

$$n_1 \xrightarrow{x_1 * x_2} n_3$$

のように新しいパスを作成する。チャンキングを行うことができるものは状態を一意に決定できるものでなければならない。そのためには空間方向のデータの抽象化ならびに時間方向についてもデータを抽象化する必要がある。

3 試験例題への応用

問題設定として以下のような実験を想定する。被験者にあるパターン (状態 S) を見せ、一定時間 (状態遷移要因 x) が経過したところで次のパターンを見せる。さらに一定時間が経過したところで次のパターンを見せるといったことを繰り返す。被験者には、パターンを見せられる度に次の状態を予想することが要求され、正解の場合には報酬が与えられ、不正解の場合には罰が与えられるものとする。

本論文では簡単のために、□と■の 2 種類のタイルを

L R

のように 2 枚並べて作られるパターンを用いた系列データ

$$\cdots \rightarrow \square\square \xrightarrow{x_1} \square\blacksquare \xrightarrow{x_1} \blacksquare\square \xrightarrow{x_1} \blacksquare\square \xrightarrow{x_1} \square\square \xrightarrow{x_1} \square\blacksquare \rightarrow \cdots$$

を試験例題として用いる。

3.1 学習経過

初期の記憶状態は、$\mathcal{RM} = \phi, \mathcal{WM} = \phi$ である。左に□があることを LW で、右にあるときは RW で表し、■が左のときは LB 、右の場合は RB で表すことにする。また、状態遷移において、前の状態を S_1、遷移後を S_2 で表すことにする。

$$\cdots \rightarrow \square\square$$

というパターンを最初に見たとき、以下のように記憶される。

$$n_0 = \{\text{未知のデータ}\}, \quad n_1 = \{LW\}, \quad n_2 = \{RW\}, \quad x_0 = \text{未知の要因}$$

$$S_1 = \{n_0\}, \; S_2 = \{n_1, n_2\}, \quad {}_{n_0 x_0}\overset{3/4}{\mathrm{N}} = \phi, \; \overset{3/4}{\mathrm{N}}_{x_0 n_1} = \phi, \; \overset{3/4}{\mathrm{N}}_{x_0 n_2} = \phi$$

$$r_1 = \langle f_1, 3/4, 3/4 \rangle, \; f_1 = \langle n_0, x_0, n_1 \rangle, \quad r_2 = \langle f_2, 3/4, 3/4 \rangle, \; f_2 = \langle n_0, x_0, n_2 \rangle$$

一定時間が経過したところで、

$$\square\square \xrightarrow{x_1} \square\bullet$$

のようにパターンが変わると、次のようになる。

$$n_3 = \{RB\}, \quad x_1 = \text{一定時間の経過}$$

$$S_1 = \{n_1, n_2\},\ S_2 = \{n_1, n_3\}, \quad {}^{3/4}_{n_1x_1}\mathrm{N} = \phi,\ {}^{3/4}_{n_2x_1}\mathrm{N} = \phi,\ {}^{3/4}_{x_1n_1}\mathrm{N} = \phi,\ {}^{3/4}_{x_1n_3}\mathrm{N} = \phi$$

$$r_3 = \langle f_3, 3/4, 3/4\rangle, \quad f_3 = \langle n_1, x_1, n_1\rangle, \quad r_4 = \langle f_4, 3/4, 3/4\rangle, \quad f_4 = \langle n_1, x_1, n_3\rangle$$

$$r_5 = \langle f_5, 3/4, 3/4\rangle, \quad f_5 = \langle n_2, x_1, n_1\rangle, \quad r_6 = \langle f_6, 3/4, 3/4\rangle, \quad f_6 = \langle n_2, x_1, n_3\rangle$$

次に、

$$\square\bullet \xrightarrow{x_1} \bullet\square$$

のようにパターンが変わると、

$$n_4 = \{LB\},\ n_5 = \{f_1\},\ n_6 = \{f_3\},\ n_7 = \{f_5\}$$

$$S_1 = \{n_1, n_3\},\ S_2 = \{n_2, n_4\}, \quad {}^{3/4}_{n_1x_1}\mathrm{N} = \{n_1, n_3\},\ {}^{3/4}_{n_3x_1}\mathrm{N} = \phi,\ {}^{3/4}_{x_1n_2}\mathrm{N} = \phi,\ {}^{3/4}_{x_1n_4}\mathrm{N} = \phi$$

$$r_7 = \langle f_7, 3/4, 3/4\rangle, \quad f_7 = \langle n_5, x_1, n_1\rangle, \quad r_8 = \langle f_8, 3/4, 3/4\rangle, \quad f_8 = \langle n_5, x_1, n_3\rangle$$

$$r_9 = \langle f_9, 3/4, 3/4\rangle, \quad f_9 = \langle n_6, x_1, n_2\rangle, \quad r_{10} = \langle f_{10}, 3/4, 3/4\rangle, \quad f_{10} = \langle n_6, x_1, n_4\rangle$$

$$r_{11} = \langle f_{11}, 3/4, 3/4\rangle, \quad f_{11} = \langle n_7, x_1, n_2\rangle, \quad r_{12} = \langle f_{12}, 3/4, 3/4\rangle, \quad f_{12} = \langle n_7, x_1, n_4\rangle$$

$$r_{13} = \langle f_{13}, 3/4, 3/4\rangle, \quad f_{13} = \langle n_3, x_1, n_2\rangle, \quad r_{14} = \langle f_{14}, 3/4, 3/4\rangle, \quad f_{14} = \langle n_3, x_1, n_4\rangle$$

$$r_{15} = \langle f_{15}, 3/4, 0\rangle, \quad f_{15} = \langle n_1, x_1, n_2\rangle, \quad r_{16} = \langle f_{16}, 3/4, 0\rangle, \quad f_{16} = \langle n_1, x_1, n_4\rangle$$

$$r_3 = \langle f_3, 3/4, 0\rangle, \quad r_4 = \langle f_4, 3/4, 0\rangle$$

のように変更される。このとき、

$${}^{3/4}_{n_1x_1}\mathrm{N} \cap S_2 = \phi$$

であるので、*1-context* として時間方向にデータを抽象化している。さらに、

$$\bullet\square \xrightarrow{x_1} \bullet\square$$

のようにパターンが変わると、

$$n_8 = \{f_{15}\},\ n_9 = \{f_{13}\},\ n_{10} = \{n_2, n_4\},\ n_{11} = \{n_3, n_6, n_7\}$$

$$S_1 = \{n_2, n_4\},\ S_2 = \{n_2, n_4\}$$

$${}^{3/4}_{n_2x_1}\mathrm{N} = \{n_1, n_3\},\ {}^{3/4}_{n_4x_1}\mathrm{N} = \phi,\ {}^{3/4}_{x_1n_2}\mathrm{N} = \{n_3, n_6, n_7\},\ {}^{1}_{x_1n_2}\mathrm{N} = \phi,\ {}^{3/4}_{x_1n_4}\mathrm{N} = \{n_3, n_6, n_7\}\ {}^{1}_{x_1n_4}\mathrm{N} = \phi$$

$$r_{17} = \langle f_{17}, 3/4, 3/4\rangle, \quad f_{17} = \langle n_8, x_1, n_1\rangle, \quad r_{18} = \langle f_{18}, 3/4, 3/4\rangle, \quad f_{18} = \langle n_8, x_1, n_3\rangle$$

$$r_{19} = \langle f_{19}, 3/4, 3/4\rangle, \quad f_{19} = \langle n_9, x_1, n_1\rangle, \quad r_{20} = \langle f_{20}, 3/4, 3/4\rangle, \quad f_{20} = \langle n_9, x_1, n_3\rangle$$

$$r_{21} = \langle f_{21}, 3/4, 1\rangle, \quad f_{21} = \langle n_{10}, x_1, n_2\rangle, \quad r_{22} = \langle f_{22}, 3/4, 1\rangle, \quad f_{22} = \langle n_{10}, x_1, n_4\rangle$$

$$r_{23} = \langle f_{23}, 3/4, 1\rangle, \quad f_{23} = \langle n_{11}, x_1, n_2\rangle, \quad r_{24} = \langle f_{24}, 3/4, 1\rangle, \quad f_{24} = \langle n_{11}, x_1, n_4\rangle$$

$$r_{25} = \langle f_{25}, 3/4, 0\rangle, \quad f_{25} = \langle n_4, x_1, n_2\rangle, \quad r_{26} = \langle f_{26}, 3/4, 0\rangle, \quad f_{26} = \langle n_4, x_1, n_4\rangle$$

$$r_5 = \langle f_5, 3/4, 0\rangle, \quad r_6 = \langle f_6, 3/4, 0\rangle, \quad r_{11} = \langle f_{11}, 3/4, 0\rangle, \quad r_{12} = \langle f_{12}, 3/4, 0\rangle$$

$$r_{13} = \langle f_{13}, 3/4, 0\rangle, \quad r_{13} = \langle f_{13}, 3/4, 0\rangle$$

のように変更される。このときも、*1-context* として時間方向にデータを抽象化しており、同時に空間方向に対しても抽象化を行っている。

4 おわりに

本論文では、環境に適応するための学習モデルを提案した。このモデルは、空間文脈、および時間文脈に依存して状態変化を示す環境に対して学習することが可能である。チャンキングを行うためには、確実な状態遷移を見つけなければならないが、そのためには空間、時間文脈を学習できなければならないことになる。このモデルの学習可能な問題として、1状態の空間文脈情報のみで次の状態が決定可能なもの、例えば対称性の検出問題など、*0-contextual* であるものをまずあげることができる。また、有限の k の場合における *k-contextual* な問題、電卓の表示とボタンの関係といったものにも学習は可能である。理論的には空間と時間の広がりが非常に大きな場合でも学習は可能であると考えられる。しかしながら、学習の段階において作成される関係の数は非常に大きなものとなってしまい、計算不可能となってしまうであろう。

今回のモデルでは、記憶を長期記憶に固定するといったことも行っていない。システムがある程度学習した段階で、突然、環境におけるルールが変わった場合、古い知識を壊して忘却し、改めて学習を行うので環境の変化にも追随することができるわけであるが、しばらくしてもとのルールに戻った場合にはもう一度学習を繰り返さなければならないといった問題がある。記憶の固定においてはルールの一般化などを行うことも必要となると考えられるが、今回のモデルでは環境におけるルールのようなものを学習するが、ルールの一般化は行っていない。

今後こうした問題について検討を行う必要がある。

参考文献

[1] 早勢欣和, 畝見達夫：系列学習における表現単位の抽象化レベルの適応的変更,Workshop On Lerning '90, 北海道手稲 (1990)

[2] 畝見達夫：抽象化リンクによる信号と記号の融合, 日本認知科学会第5回大会·発表論文集, B-1, (1988)

[3] Laird, J.E, P. S. Rosenbloom and A. Newell : *Chunking in Soar : The Anatomy of a General Learnig Mechamism*, Machine Learning, Vol. 1, pp. 11–46, (1986)

[4] Muggleton, S. : *Inductive Acquisition of Expert Knowledge*, Addison-Wesley (1990)

[5] 麻生英樹：ニューラルネットワーク情報処理–コネクショニズム入門、あるいは柔らかな記号に向けて–, 産業図書, (1988)

[6] Rumelhart, D. E., McClelland, J.L., and The PDP Research group. *Paralel Distributed Processing – Explorations in the Microstructure of Cognition Volume 1: Foundations.* Cambridge, MA:MIT press. (1986)