手続き的知識を獲得する連想記憶モデル
Associative Memory Model to Acquire Procedural Knowledge

郷古 学 † 下村 正夫 † 菅谷 至寛 † 阿曽 弘具 †
Manabu GOUKO Masao SHIMOMURA Yoshihiro SUGAYA Hirotomo ASO

1. まえがき

手続き的知識の記憶を行なう従来の連想記憶モデルの多くは，学習により記憶すべき知識が予め準備されている．そのため，このようなモデルを基にシステムを構築する場合，記憶した知識を逸脱する環境下ではシステムが機能しないという問題が生じる．そこで本稿では，周囲の環境からモデルに対して入力される情報の順序，入力間隔及び頻度を基に，手続き的知識を獲得する連想記憶モデルを提案する．

2. 手続き的知識の構築

本稿において「A から B」という手順を手続き的知識の 1step と考え，「A → B」と表す．このような知識はモデルに対して情報 A が入力された時に，その次の手順である情報 B を出力することで表現される．従来の連想記憶モデルに，「A → B」という手続き的知識を記憶させる場合，入力 A に対して出力 B となるように予め学習させることで実現していたが，本研究では「A → B」という関係が事前に与えられない場合を考える．提案モデルは，周囲の環境から入力される情報を基に「A → B」という手続き的知識を自律的に獲得する．モデルが「A → B」という知識を獲得するにあたり，モデルへ入力される情報 A，B はそれぞれ一定時間持続するものとし，以下のような傾向があると仮定する．

(1) モデルに対して B よりも以前に A が入力される．

(2) A が入力されてから，B が入力されるまでの時間が短いほど，「A → B」という知識が獲得されやすい．

(3) モデルに対して情報 A，B の入力頻度が高く，かつ入力の時間的傾向が (1)，(2) に従うものであれば，モデルの周囲の環境における「A → B」という知識の一般性が高い．

(1) は，A と B の発生する順序を規定する．また (2) は二つの情報の入力される時間が短いほど，それらの情報の因果関係が強い傾向にあることを意味し，逆にそれぞれの情報が入力される間隔が長いほど，二つの情報の因果関係は弱く，独立した情報と考える．(3) は知識の一般性について考慮したものであり，入力頻度が高い場合は一般的な知識とし，逆に頻度が低い場合は，特別な知識と考える．本稿では，より一般的な知識ほど，獲得すべきものと考える．

芳澤ら [1] は逐次記憶学習を目的とした自己連想記憶モデルを提案している．本稿で提案するモデルは，それを基に，新たに相互連想記憶処理を加え，両者を連携させる機構を組み込んだもので，より一般性の高い手続き的知識の獲得を実現している．

†東北大学大学院工学研究科電気・通信工学専攻

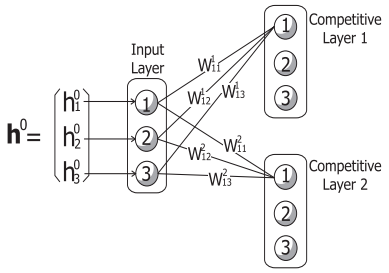
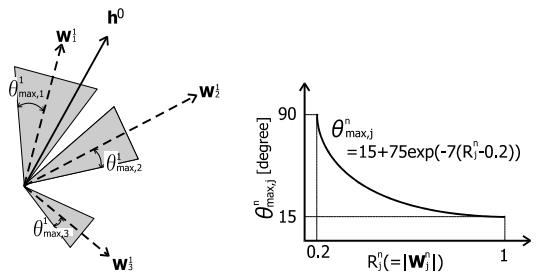


図 1: 提案モデルの構造 ($n_0 = 3, n_1 = 3$)



(a) 重みベクトル (b) 角度閾値

図 2: 重みベクトルと角度閾値

3. 提案モデル

提案モデルの構造を図 1 に示す．モデルは n_0 個のニューロンからなる入力層，n_1 個のニューロンからなる 2 つの競合層からなり，入力層の各ニューロンは各競合層の全てのニューロンと全結合している．時刻 t における入力層の i 番目のニューロンと，競合層 $n(= 1, 2)$ の j 番目のニューロンとの結合重みを $W^n_{ji}(t)$ と表す．j 番目のニューロンへの結合重みをベクトル表現し，$\boldsymbol{W}^n_j(t) = (W^n_{j1}(t), W^n_{j2}(t), \ldots, W^n_{jn_0})$ とおく．また，各競合層のニューロンは，記憶パターンの信頼度 $R^n_j(t)$，角度閾値 $\theta^n_{max,j}(t)$，発火パラメータ $p^n_j(t)$ の 3 つのパラメータを持つ．競合層 1 から 2 へは発火パラメータを通じて影響を与える．

提案モデルは記憶処理と想起処理の 2 つの処理モードで動作する．

3.1 記憶処理

提案モデルにおける記憶処理は，情報 (以下パターンと呼ぶ) が提示された場合に，結合重みを更新する処理である．記憶処理はモデルに対して提示されたパターンそのものを記憶する自己連想記憶処理と，提示されたパターンの次に提示される可能性の高いパターンを記憶する相互連想記憶処理の二つの処理からなる．

3.1.1 自己連想記憶記処理

自己連想記憶記処理ではモデルに対して提示されるパターンの記憶を行う．この処理では，モデルに対して提

示される回数の多いパターンほど重要なパターンであるとし，優先的に記憶するようにしている [1]．自己連想記憶情報は競合層 1 のニューロンの持つ結合重みに蓄えられる．本処理は 2 段階の処理を繰り返すことで行われ，第 1 段階では提示される入力パターンを記憶するニューロンを決定する．第 2 段階ではそのニューロンの結合重みベクトルの更新を行う．また同時に「記憶パターンの信頼度」の更新を行う．

入力パターンはベクトルとして扱い，ベクトルの大きさで正規化したものを入力層に入力するとする．この入力パターンベクトルを $\boldsymbol{h}^0 = (h_1^0, h_2^0, \ldots, h_{n_0}^0)$ とし，以下の式で競合層 1 の出力を定める．

$$V_i^0(t) = h_i^0(t) \tag{1}$$

$$h_j^1(t) = \sum_{i=1}^{n_0} W_{ji}^1(t) V_i^0(t) \quad (j = 1, 2, \ldots, n_1) \tag{2}$$

$$V_j^1(t) = h_j^1(t) \tag{3}$$

ここで $h_i(t)$ は時刻 t における i 番目のニューロンへの入力値，$V_j(t)$ は j 番目のニューロンの出力値を表す．肩の添え字 $0, 1$ はそれぞれ入力層，競合層 1 を表す．

第 1 段階として，入力パターンを記憶するニューロン j^* を次のように決定する．j^* を更新ニューロンと呼ぶ．$\boldsymbol{h}^0(t)$ と $\boldsymbol{W}_j^1(t)$ のなす角度を $\theta_j^1(t)$ とおく．すなわち，

$$\theta_j^1(t) = \left| \cos^{-1} \left(\frac{h_j^1(t)}{\left\| \boldsymbol{W}_j^1(t) \right\|} \right) \right| \tag{4}$$

(1) $h_j^1(t)$ を最大とする j を選択する．

(2) $\theta_j^1(t) \leq \theta_{max,j}^1(t)$ ならば，$j^* = j$ とする．

(3) そうでない場合は j を除いて，(1)，(2) の操作を繰り返し，更新ニューロンを決定する．

(4) 全ての j で $\theta_j^1(t) > \theta_{max,j}^1(t)$ ならば，重みベクトルの更新及び後述する記憶パターンの想起は行わない．

図 2(a) の例では，$\boldsymbol{W}_1^1(t)$ を持つニューロンが更新ニューロンとなる．第 2 段階は，更新ニューロン j^* について，式 (5) に示す Oja 則 [2] を用いて，重みベクトルを更新する．

$$W_{j^*i}^1(t+1) = W_{j^*i}^1(t) + \eta_1 V_{j^*}^1(t) \left(V_i^0(t) - V_{j^*}^1(t) W_{j^*i}^1(t) \right) \tag{5}$$

ここで，η_1 はある定数で，学習係数である．

重みベクトル $\boldsymbol{W}_j^n(t)$ の大きさをニューロン j の記憶パターンの信頼度 $R_j^n(t)$ と呼ぶ．すなわち，

$$R_j^n(t) = \left\| \boldsymbol{W}_j^n(t) \right\| \tag{6}$$

である．重みベクトルの大きさは，重みベクトルが (5) 式で更新されるごとに増加し 1 に収束する [2]．すなわち，記憶パターンの信頼度 $R_j^1(t)$ は入力パターンの提示回数に対応して増加する．

次に角度閾値 $\theta_{max,j}^n(t)$ を図 2(b) に示すような記憶パターンの信頼度の関数に従って更新する．これにより，信頼度が高い記憶パターンは信頼度が低い記憶パターンと比較して小さな角度閾値を持つため，新しい入力パターンに対して影響を受けにくくなり頑健になる．また信頼度が低い記憶パターンのニューロンで新たに重要な入力パターンを記憶することが可能になる．

モデルは提示パターンによる重みの更新に伴い，記憶パターンの信頼度が増加し，各重みベクトルの持つ角度閾値が小さくなるために，モデルに対して新たに重要なパターンが提示された場合，学習を行うことが出来ないという問題が生じる．この問題を解決するため，モデルはパターンが提示される度に，各競合層の全てのニューロンの持つ結合重みの大きさを減少させるという処理を行う．減少量は適切な値を求める必要があり，これに関しては文献 [1] を利用した．

3.1.2 相互連想記憶記処理

提案モデルにおいて手続き的知識は，入力パターンに対し次に入力される可能性の高いパターンを出力する相互連想記憶として表現される．手続き的知識の獲得メカニズムを以下に述べる．本処理は 2 段階からなる．第 1 段階では発火パラメータの値を基に，競合層 2 のニューロンの結合重みを更新する．第 2 段階では提示される入力パターンを基に，次の時点で用いるために発火パラメータの値を更新する．

第 1 段階では競合層 2 の全ての重みベクトル $\boldsymbol{W}_j^2(t)$ に対し，入力ベクトルとの角度を計算し，角度閾値 $\theta_{max,j}^2(t)$ の範囲内であれば，その重みベクトルを式 (7) で更新する．

$$W_{ji}^2(t+1) = W_{ji}^2(t) + \eta_2 f_j(t) V_j^2(t) \left(V_i^0(t) - V_j^2(t) W_{ji}^2(t) \right) \tag{7}$$

ここで $V_j^2(t)$ は競合層 2 の j 番目のニューロンの出力値であり，$V_j^1(t)$ と同様に定める．η_2 は定数で学習係数である．$f_j(t)$ は以下の値をとる．

$$f_j(t) = \begin{cases} 0 & \text{if } \ j = j^* \\ p_j^2(t) & \text{otherwise} \end{cases} \tag{8}$$

時刻 t における競合層 n の j 番目のニューロンが持つ発火パラメータを $p_j^n(t)$ と表す．$p_j^n(t)$ は 0 から 1 の値とるものとする．$p_j^n(t) > 0$ である時，ニューロン j を発火状態と呼ぶ。発火パラメータは次式で更新する.

$$p_j^1(t+1) = \begin{cases} R_j^1(t) & \text{if } \ j = j^* \\ p_j^1(t) - e & \text{otherwise} \end{cases} \tag{9}$$

$$p_j^2(t+1) = p_j^1(t+1) \tag{10}$$

ただし，$p_j^n(t+1) < 0$ の場合は，0 に再設定する．この更新は，モデルに対して入力されたパターンに対して，競合層 1 の更新ニューロン j^* の持つ発火パラメータをその記憶パターンの信頼度 $R_{j^*}^1(t)$ に一致させ，そのほかのニューロンの発火パラメータを減衰定数 e により減衰させている．

全てのニューロンが持つ発火パラメータが 0 の状態をモデルの初期状態とする．2 つの異なるパターンが続いて入力される場合を例に，記憶処理について説明する．

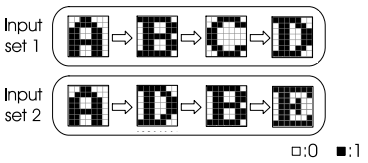


図 3: 入力パターン

まず最初に一定時間持続して入力されたパターンに対して自己連想記憶処理および相互連想記憶処理が行われる．すなわち，競合層 1 のニューロン j^* の持つ重みベクトルが更新され，同時に相互連想記憶処理において発火パラメータの更新がなされる．しかし全てのニューロンにおいて $f_j(t) = 0$ であるために，競合層 2 の重みベクトルの更新は行われない．一般的には，各競合層の $j(= j^*)$ 番目のニューロンが発火状態となり，競合層 2 のニューロンが持つ重みベクトルの更新が行われる．モデルに対して入力が終了すると，発火パラメータは式 (9)(10) により減衰を始める．

続いてモデルに対して別のパターンが一定時間持続して入力されると，競合層 1 において自己連想記憶処理が実行され，競合層 2 において，先に入力されたパターンに対して発火していたニューロンの発火パラメータが $0 < p_j^2(t)$ を保っている場合，そのニューロンの持つ重みベクトルが式 (7) により更新される．つまり競合層 2 において，先に入力されたパターンに対して発火したニューロンの持つ重みベクトルが後から入力されたパターンに更新されることになる．この様にして提案モデルでは，入力されるパターンの入力間隔および頻度に応じて，過去の入力パターンに対応するニューロン位置に，現在の入力パターンを競合層 2 に記憶する，つまりモデルは手続き的知識の 1step を獲得したことになる (相互連想記憶)．

3.2 想起処理

自己連想想起パターン y_i^1 は，モデルへ提示された入力パターンに対して，記憶処理において選択された競合層 1 における更新ニューロン j^* の重みベクトルとする．すなわち，出力パターンは次式で与えられる．

$$y_i^1 = W_{j^*i}^1 \quad (i = 1, 2, \ldots, n_0) \tag{11}$$

また相互連想想起パターン y_i^2 も同様にして，次式で与えられる．

$$y_i^2 = W_{j^*i}^2 \quad (i = 1, 2, \ldots, n_0) \tag{12}$$

4. 実験結果及び考察

提案モデルが，手続き的知識を獲得していく様子を数値実験により示す．実験で用いたネットワークは $n_0 = 49, n_1 = 4$ とした．実験では図 3 に示す 7×7 の 5 種類のドットパターン (A，B，C，D，E) を用い，モデルに対して A → B → C → D(入力セット 1) と，A → D → B → E(入力セット 2) の，入力セットを用いて学習を行った．それぞれの入力セットは，異なる手続き的知識を表現している．実験では，各パターンの入力が持続する時間を 50 とした．すなわち，入力セット 1 の場合，A が 50 回入力された後 B が 50 回，続いて C が 50 回，D が 50 回入力され，200 ステップで 1 つの入力セットの入力を終了する．これに続く入力セットは確率 s_1 で入力セット 1 を，

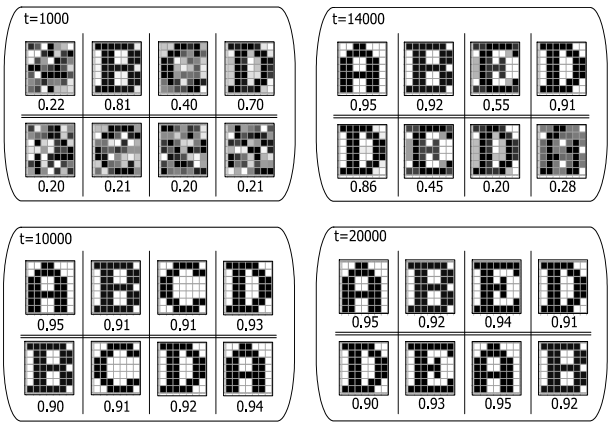


図 4: 実験結果

確率 s_2 で入力セット 2 を選択する．実験は $t = 0$ から 20000 まで行った．実験の前半 ($t = 0$ から 10000) では，$(s_1, s_2) = (0.8, 0.2)$ とし，後半 ($t = 10001$ から 20000) では，$(s_1, s_2) = (0.2, 0.8)$ とした．この様にすることで，実験の前半と後半で，異なる手続き的知識が存在する環境を作り出している．前，後半部の環境をそれぞれ env1，env2 と呼ぶ．

実験結果を図 4 に示す。この図は重みベクトルの各成分の大きさを，信頼度で正規化して得られた記憶パターンを表したものである．図中パターンの下に記した数字は，その記憶パターンの信頼度である．各時刻の結果で，上段が競合層 1 の記憶パターン，下段が競合層 2 の記憶パターンである．上下の各パターンは同じ番号のニューロンを表している．モデルは上段のパターンが入力として与えられた場合，下段の記憶パターンは次に入力されるパターンを記憶するようになることが期待されていた．結果を見ると，実験の前半において，自己連想記憶の形成が始まり，遅れて相互連想記憶が形成されていく様子が分かる．$t = 10000$ の結果を見ると env1 に主に存在する A → B → C → D という手続き的知識が獲得できている．また，環境が env2 に変化すると，モデルは自らの持つ自己連想記憶と相互連想記憶を更新し ($t = 14000$)，env2 に主に存在する A → D → B → E という手続き的知識を獲得している ($t = 20000$)．

5. まとめ

本稿では，適応的に環境から手続き的知識の獲得が可能な連想記憶モデルを提案した．提案モデルが，モデルに入力される情報の順序，入力間隔及び頻度に応じて，手続き的知識を獲得していくことを実験により確認した．特に環境の変化に適応して，手続き的知識が獲得できることが示された．

参考文献

[1] 芳澤伸一，道木慎二，大熊繁，”記憶容量の制約下での追加学習を目的とする新しい連想記憶システムの提案，”信学論 (D-II)，vol．J82-D-II，No．6，pp1072–1081，1999．

[2] J.Hertz，A.Krogh，R.G.Palmer 著，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．