

構造化ニューラルネットによる運動スキーマの構成

正 員 辻 敏夫[†] 非会員 石田 祐介[†] 正 員 伊藤 宏司[†]
 正 員 長町 三生[†] 非会員 西野 達夫^{††}

Motor Schema Model Learned by Structured Neural Networks

Toshio TSUJI[†], *Member*, Yusuke ISHIDA[†], *Nonmember*, Koji ITO[†],
 Mitsuo NAGAMACHI[†], *Members and*
 Tatsuo NISHINO^{††}, *Nonmember*

あらまし 人間はよく生じる定型的な運動に関するプランをスキーマとして記憶している。そして、状況に応じて適切なスキーマを選択し、ほとんど無意識のうちにその動作に心要な筋運動を発生させる。運動スキーマに表現される動作は定型的ではあるが、把持動作や歩行運動に見られるように動作の並列性や直列性を考慮した優れたプラン構造を備えている。しかも、そのプラン構造は学習により獲得される。本論文ではニューラルネットを用いてこのような運動スキーマのモデル化を試みる。そのため、並列性を含んだ形で動作系列を生成する部分(並列動作生成部)とその動作系列に対して並列に実行できない制約を満たすように修正をかける部分(制約表現部)にネットワークをあらかじめ構造化する。そして各部ごとに学習を行い、実行時に両者を結合することにより、動作の直列性と並列性を考慮した運動系列を生成できるモデルを提案する。最後に、本モデルをタイプライティング動作の運動スキーマ形成に適用し、動作の実行に伴う運動系の拘束を考慮した動作系列を生成できることを示す。

1. ま え が き

人間は、異なる目的をもついくつかの行動を同時に実行することができる。例えば、食事をしながら会話をするような場合には、意識は会話の方に向けられており、食事のための動作はほとんど無意識のうちに実行される。これは、よく生じる定型的な活動に関するプランを運動スキーマとして記憶していることによる。その場の状況に応じて、適切なスキーマを選択し、適宜、切り換えて行くことによってほとんど無意識のうちにその動作に必要な筋運動を発生できるのである。

もちろん、このような無意識の動作は定型的であるけれども、かなり複雑なプラン構造を備えていなければならない⁽¹⁾。例えば、コップをつかむというプランを考えると、腕を伸ばす動作の間に手をつかむ

ための形にしておけば、前の動作が完全に終了してから次の動作を始める場合に比べて、効率的な実行ができる(図1(a))。一方、つかんでいるコップをテーブルに置く場合には、コップを置く動作が完全に終わらないうちに手を離すことはできない(図1(b))。また、歩行運動などのように、静的には安定でない動作からなるプランの場合には、各動作間の依存関係をプランニングに表現しない限り、動作と動作の間で安定を失い、プランが実行不可能になってしまう。

このように人間の運動スキーマは、プランを構成する各動作の並列性と直列性を、行動目的に応じてうまく組み合わせることがわかる。しかも、そのプラン構造は定型的な活動を何度も繰り返して実行しているうちに、学習的に形成される。

本研究では、自律的な運動能力を有するロボット制御への応用を目的として、運動スキーマのモデル化を試みる。このような運動スキーマを学習に基づいて実現するには、逐次記号操作的な情報処理を用いるよりも、ニューラルネットのように、並列的なメカニズムが相互作用的に動作する機構の方が適していると考え

[†] 広島大学工学部, 東広島市
 Faculty of Engineering, Hiroshima University, Higashi-Hiroshima-shi, 724 Japan

^{††} 広島工業大学, 広島市
 Department of Industrial Engineering, Hiroshima Institute of Technology, Hiroshima-shi, 731-51 Japan

られる。ニューラルネットは、

- (1) 高度に並列な処理の実現が可能
- (2) 学習によって自然に知識を獲得できる
- (3) ノイズや故障に対するロバストネスをもつ

などといった利点を備えており、直列記号処理に代わる新しい並列分散型情報処理機構として注目されている⁽²⁾⁽³⁾。しかしながら、その高度に並列化されたメカニズムにより、逆に時間的に変化するデータを扱う能力が低く、高次の推論に必要な直列・論理的な情報処理は不得意である⁽⁴⁾。

従って、ニューラルネットを用いて運動スキーマを実現するためには、

- (1) 動作の並列性と直列性をどのように記述するか
- (2) 時間的な動作の系列をどのようにして生成するか
- (3) 問題領域の制約をどのように表現するか

といった点が問題となる。

本論文では、このようなニューラルネットに基づいて運動スキーマを構成するアプローチをとる。そのため、並列性を含んだ形で動作の系列を生成する部分(並列動作生成部)と、その動作系列に対して、並列に実行できない制約を満たすように修正をかける部分(制約表現部)にネットワークをあらかじめ構造化する。そして、各部分ごとに学習を行い、実行時に両者を結合することにより、動作の直列性と並列性を考慮した動作系列を生成できるモデルを提案する。以下、2.では、ニュー

ラルネットを用いて直列・時系列処理を行うために提案されている従来の手法について概説し、3.で運動スキーマを構成するためのニューラルネットモデルを提案する。そして、4.では、このモデルをタイプライティング動作のスキーマ形成に応用し、動作の実行に伴う運動系の拘束を考慮した動作系列を生成できることを示す。

2. ニューラルネットによる直列処理

直列・時系列処理をニューラルネットで行うため、従来2通りの方法が考えられている。一つは、時系列を空間パターンに展開する方法である⁽⁵⁾。この方法は、単位時間の遅延特性をもつユニットを用いることによって、各時刻に相当する階層を用意し、入力を順次、次の階層に送っていくものである。但し、この方法では、認識したい時系列の長さの分だけの階層を用意する必要があり、あまり長い時系列を扱うことができない。

もう一つの方法は、モデルがその内部状態を明示的にもつ方法で、その内部状態を表現し保持するための状態ユニット群を用意する。そして、1単位時間前の出力ユニット群や状態ユニット群からのフィードバックにより内部状態を更新していくことによって一種のオートマトンとして動作し、時系列データを扱う。

Rumelhartら⁽⁶⁾は後者のアプローチに基づき、外界に対する行動を決定する解釈ネットワーク部と、外部世界のシミュレートを行う世界モデル部からなるニューラルネットを提案している(図2)。ここで解釈ネットワーク部は入力を解釈し、その入力に対応する行動を決定する。世界モデル部は、環境をシミュレートすることによって、決定された行動がどのような環境の変化を起こすかを予測する。そして、その予測の結果を解釈ネットワーク部の入力にフィードバックすることで、人間の思考のような逐次処理を実行できる可能性を示している。

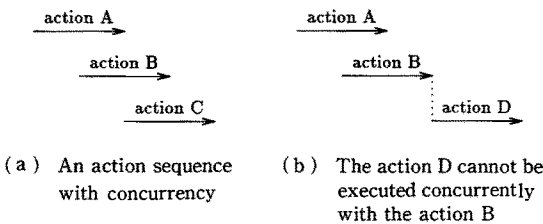


図1 並列性を含む動作系列
Fig. 1 Action sequences with concurrency.

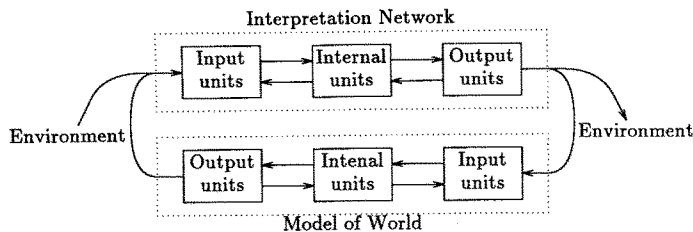


図2 メンタルシミュレーションモデル⁽⁶⁾
Fig. 2 The model of mental simulations⁽⁶⁾.

更に Rumelhart と Norman⁽⁷⁾ は、ニューラルネットに直列・逐次のな処理を行う能力があることを示すモデルとして、タイプライティング動作のシミュレーションを行っている。これは、単語や文字の打けん動作のスキーマに相当するユニットからなるニューラルネットをあらかじめ単語のつづりに応じて構造化し、フィードバックを用いて人間のタイプライティング動作の動作系列を生成するものである。特に、このモデルでは、次の文字の打けんスキーマのユニットを部分的に活性化することによって、次の打けんに備えてあらかじめ指を動かしておくといった動作の並列性を表現している。そして、活性化値にノイズを付加することによって、人間のようなタイプミスを再現することもできる。但し、このモデルでは、単語のスペリングをネットワークの構造として事前に与えておく必要がある。

一方、Jordan⁽⁸⁾ は多階層ニューラルネットにフィードバックと内部状態を付加することによって、学習した時系列を並列性を含む形で生成できるモデルを提案

している。このモデルの構成を図3に示す。状態ユニット群は内部状態を明示的に表現し、この内部状態を入力として用いて次の出力を計算することにより、時系列を扱う。また、出力の計算は基本的にフィードフォワード型であるから、誤差逆伝搬法⁽⁵⁾を用いることにより、教師信号として与えられる時系列を学習できる。

このモデルの特徴は、順序関係だけを与える教師信号に対して並列的な動作系列を生成できることにある。これは、誤差逆伝搬法の補間能力とフィードバックループの働きにより時間的に近い状態が、似た(ハミング距離が近い)表現をもつためである。例えば、「1」を教えられたユニットは、その時刻の前後で、部分的に活性化(例えば0.8)する。更に、Miyata⁽⁹⁾ は Jordan のモデルを二つ直列に結合し、運動系列の習熟過程をモデル化している。

しかしながら Jordan のモデルでは、動作を並列に行うことができないという直列性の制約を教師信号として明示的に与えている。そのため、同じプランに対する系列を生成する場合でも、動作制約が異なれば、再度学習を行う必要がある。また、誤差逆伝搬法をそのまま用いているため、複雑な動作系列に対しては学習が難しく、モデルに含まれるパラメータの設定が極めて困難である。

そこで本論文では、並列性を含む形で動作系列を生成する部分(並列動作生成部)と、問題領域から要求される制約に合わせてその系列を修正する部分(制約表現部)にあらかじめネットワークを構造化する。そして、各部分ごとに学習を行い、実行時に両者を結合する。これにより、領域依存な制約表現部をモジュールとして切り換えることができ、他の問題領域(環境, 作業条件, 目的)の運動スキーマとして用いることが可能となる。

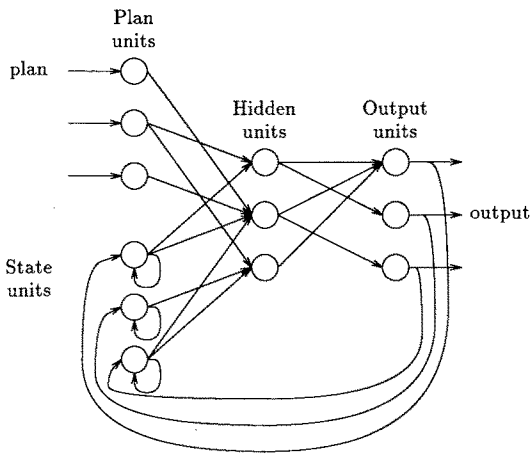


図3 逐次ネットワーク⁽⁸⁾
Fig. 3 The sequential network⁽⁸⁾.

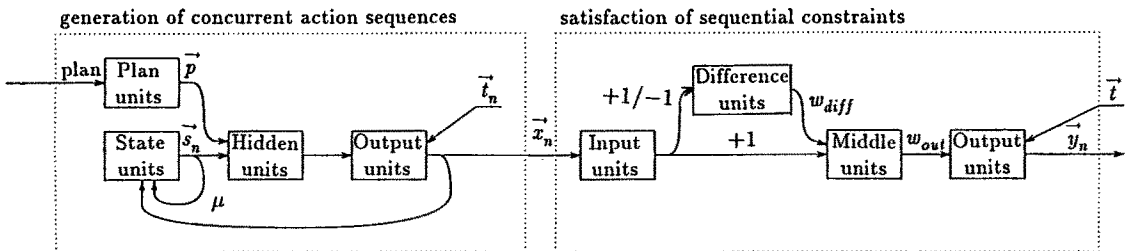


図4 構造化ニューラルネットワーク
Fig. 4 The structured neural network for motor schemas.

3. 運動スキーマのための構造化ニューラルネット

本モデルは、並列動作生成部と制約表現部からなる(図4)。並列動作生成部は、あるプランに対する動作系列の順序関係を学習し、並列性を含んだ動作系列(図1(a)参照)を生成する。制約表現部は、問題領域に依存する制約を学習し、その制約を満足しない並列動作生成部の出力に対して抑制をかけ、制約を満たすように修正する。そして、領域に無関係な表現形式をもつ入力運動系への目標値に相当する表現形式に変換する。

但し、本論文では以下の性質をもつ問題を考えることにする。

(1) 各動作の順序関係が入れ替わることはない。例えば、単語のスプリングのように文字列の順序が与えられている。

(2) 並列実行できる動作と並列実行できない動作があり、それは問題領域に依存して決まっている。例えば、タイプライティングにおいては、同じ指に割り当てられた複数の文字を並列に打つことはできない。

3.1 並列動作生成部

本論文では、この部分に Jordan のモデル⁽⁸⁾と同じ構造のニューラルネットを用いる。但し、Jordan のモデルとは異なり、教師信号としては動作の順序関係だけを与え、並列に実行できないという制約を与えることは行わない。その結果、並列動作生成部は完全に並列的な動作系列を生成することになり、Jordan のモデルがもつ学習の非線形性やパラメータ設定の難しさをある程度回避できる。

図4に並列動作生成部の構成を示す。プランは、プランユニット群にベクトル \vec{p} (単位ベクトル)として与える。また、時刻 n における出力ユニット群の出力ベクトル要素 $x_{n,i}$ は各動作の活性値を表し、0から1の値をとる。

時刻 n における出力ベクトル \vec{x}_n は、ベクトル \vec{p} と状態ユニット群の状態ベクトル \vec{s}_n の関数として得られる。

$$\vec{x}_n = \vec{f}(\vec{s}_n, \vec{p}) \quad (1)$$

この関数 \vec{f} は、プランユニット群、状態ユニット群から出力ユニット群への多階層ネットワークで表現されることになる。

各ユニットの動作を以下のように定義する。但し、 $u_{n,i}$ 、 $v_{n,i}$ はユニット i の入力、出力で、 w_{ij} はユニット j からユニット i のリンクの重みである。

$$u_{n,i} = \begin{cases} p_i & (\text{プランユニット}) \\ s_{n,i} & (\text{状態ユニット}) \\ \sum_j w_{ij} v_{n,j} & (\text{隠れユニット, 出力ユニット}) \end{cases} \quad (2)$$

$$v_{n,i} = \begin{cases} u_{n,i} & (\text{プランユニット, 状態ユニット}) \\ \frac{1}{1+e^{-u_{n,i}}} & (\text{隠れユニット, 出力ユニット}) \end{cases} \quad (3)$$

すなわち、プランユニット群、状態ユニット群の各ユニットは入力をそのまま出力し、隠れユニット群、出力ユニット群の各ユニットはシグモイド型の特性関数に従って活性値を出力する。

状態ユニット群の状態 \vec{s}_n は、1単位時間前の自分自身と出力からのフィードバックを用いて、次式のように更新する。

$$\vec{s}_{n+1} = \mu \vec{s}_n + \vec{x}_n \quad (4)$$

但し、 $\vec{s}_0 = [1, 1, \dots, 1]^T$ で、 μ は状態ユニットの自己フィードバックループの重みである。

本モデルの学習則は基本的に誤差逆伝搬法⁽⁹⁾に基づくが、フィードバックや状態ユニットをもつため以下の方法を用いる。

(1) 各リンクの重みを、乱数で初期化する。

(2) 各プランに対して以下の(2 a)~(2 f)の手続きを行う。

(2 a) プランユニット群にプラン入力 \vec{p} を与えると共に、状態ユニット群を初期化する。

(2 b) 出力 $\vec{x}_n = \vec{f}(\vec{s}_n, \vec{p})$ を求める。

(2 c) 出力 \vec{x}_n とその時刻に対する教師信号 \vec{t}_n を用いて誤差逆伝搬学習を行い、リンクの重み w_{ij} を更新する。

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial v_{n,i}} \frac{dv_{n,i}}{du_{n,i}} v_{n,i} \quad (5)$$

但し、出力ユニットに対しては、

$$\frac{\partial E}{\partial v_{n,i}} = \begin{cases} v_{n,i} - t_{n,i} & (t_{n,i} = 0, 1) \\ \max(v_{n,i} - 0.8, 0) & (t_{n,i} : \text{不完全活性値}) \end{cases} \quad (6)$$

で、隠れユニットに対しては、

$$\frac{\partial E}{\partial v_{n,i}} = \sum_k \frac{\partial E}{\partial v_{n,k}} \frac{dv_{n,k}}{du_{n,i}} w_{ki} \quad (7)$$

とする。 $t_{n,i}$ が不完全活性を表す教師信号の場合には、出力が0.8以下になるよう学習する。

(2 d) 新しい w_{ij} を用いて $\vec{x}_n = \vec{f}(\vec{s}_n, \vec{p})$ を求める。

(2 e) 式(4)により, 内部状態 \vec{s}_{n+1} を更新する.

(2 f) そのプランに対する教師信号の系列が終わるまで, (b)~(e)を繰り返す. この間, \vec{p} は固定しておく.

(3) (2)で学習したプランのうち, 最も誤差の大きいものに対して, 再度(a)~(f)の学習手続きを適用する(強化学習).

(4) すべてのプラン, 時刻, 動作に対して誤差が ϵ より小さくなるまで, (2)~(3)を繰り返す.

ここで教師信号としては, 次の値を用いる.

(i) プラン l , 時刻 n で完全に活性化すべき動作:

$$t_{n,i}^{(l)} = 1$$

(ii) プラン l に含まれる動作:

$$t_{m,i}^{(l)} = * \text{ (不完全活性化値) } (m \neq n)$$

(iii) プラン l には出現しない動作:

$$t_{m,i}^{(l)} = 0 \quad (m = 1, 2, \dots, N)$$

(iv) プランの開始時と終了時のすべての動作:

$$t_{m,i} = 0 \quad (m = 1, N)$$

例えば, "abc" というプランに対しては, 図5のような教師信号の系列を与える. 図中, *印は不完全活性化値を表す.

3.2 制約表現部

タイプライティング動作のような運動を考えたとき, 同じ指に割り当てられた文字を打つ動作は並列実行できないという制約がある. すなわち, 運動系の同じ出力器に対応する動作群では, 群内で最も強い活性化値をもつ動作を実行し, 他の動作は抑制する必要がある.

このような制約は, 並列動作生成部と違って時間的なダイナミクスを考える必要がないので, 理論上は多階層ネットワークによる写像で実現できる. しかし, 写像の非線形性がかなり強いので学習に時間がかかったり, 学習がローカルミナマに落ち込む可能性が高い. そこで, 学習を効率的にするために, 制約を与える部分と表現変換を行う部分に分離し, それぞれの部分であらかじめネットワークを構造化しておく.

図4に制約表現部の構成を示す. 図中, 入力ユニット群から中間ユニット群までの写像で領域からの制約を実現し, 中間ユニット群から出力ユニット群までの写像で表現変換を行う.

$m =$	1	2	3	4	5	6	7
a:	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} * \\ * \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ * \\ * \end{bmatrix}$	$\begin{bmatrix} * \\ 1 \\ * \end{bmatrix}$	$\begin{bmatrix} * \\ * \\ 1 \end{bmatrix}$	$\begin{bmatrix} * \\ * \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
b:	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} * \\ * \\ * \end{bmatrix}$	$\begin{bmatrix} * \\ * \\ * \end{bmatrix}$	$\begin{bmatrix} * \\ 1 \\ * \end{bmatrix}$	$\begin{bmatrix} * \\ * \\ 1 \end{bmatrix}$	$\begin{bmatrix} * \\ * \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
c:	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} * \\ * \\ 0 \end{bmatrix}$	$\begin{bmatrix} * \\ * \\ * \end{bmatrix}$	$\begin{bmatrix} * \\ * \\ * \end{bmatrix}$	$\begin{bmatrix} * \\ * \\ 0 \end{bmatrix}$	$\begin{bmatrix} * \\ * \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

図5 教師信号系列の例

Fig. 5 An example of target activity signals.

差検出ユニット群は, 入力ユニット群のすべてのユニット対に対して用意し, そのユニット対の活性化値 x_i, x_j の差 diff_{ij} を検出する.

$$\text{diff}_{ij} = \begin{cases} 0 & (\text{if } x_i \leq x_j) \\ 1 & (\text{if } x_i > x_j) \end{cases} \quad (8)$$

中間ユニット群は出力ユニットに対応してグループを形成し, グループ内に自分より強い活性化値をもつユニットがある場合, 差検出ユニットから w_{diff} を通して抑制を受ける.

$$x_k' = \min(x_k + \sum_{ij} w_{\text{diff}k,ij} \text{diff}_{ij}, 0) \quad (9)$$

但し, $w_{\text{diff}k,ij} \leq 0$ は, 差検出ユニット ij から中間ユニット k へのリンクの重みで, グループ内の動作間の制約を表現している. 出力ユニット群は, x_j' を w_{out} を通して受け取り, その重み付き和 y_i を出力する.

$$y_i = \sum_j w_{\text{out}ij} x_j' \quad (10)$$

制約表現部の動作を図6を用いて説明する. 動作 a と動作 b は同じ出力ユニット O_1 に, c は別の出力ユニット O_2 に対応している. 今, 並列動作生成部から入力ユニットに与えられる各動作の活性化値が, $a=1.0, b=0.8, c=0.6$ であるとする. このとき, 差検出ユニットのうち, a-b, a-c, b-c が活性化する. 中間ユニット b' は, a-b からリンク w_{diff} を通して抑制がかけられ, $b'=0$ となる. しかし, ユニット a', c' には, リンクが接続されていない ($w_{\text{diff}}=0$) ので, 抑制がかかるとはならず, 入力値がそのまま中間ユニットの活性化値に伝えられる. そして, これらの活性化値は, リンク w_{out} を通して出力ユニットに伝達される.

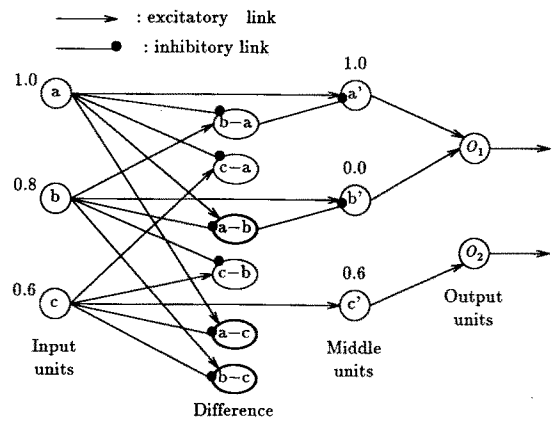


図6 制約表現部の構造

Fig. 6 The network structure for the constraint satisfaction.

ネットワークの学習則としては、以下の方法を用いる。

- (1) w_{diff}, w_{out} を 0 に初期化する。
- (2) 並列動作生成部の出力であるすべての動作に対して、(2 a)~(2 b) の手続きを適用する。

(2 a) 動作 c を表す単位ベクトル $\vec{x}^{(c)}$ を入力ユニット群に、 c に対する教師信号 $\vec{i}^{(c)}$ を出力ユニット群に与える。

(2 b) 式(11)を用いて、 w_{out} を修正する。このとき、 w_{diff} はすべて 0 なので $\vec{x}' = \vec{x}^{(c)}$ である。これにより中間ユニットを暗黙的にグループ化する。

$$\Delta w_{out,ij} = x_j' t_i \tag{11}$$

- (3) すべての動作に対して、(3 a)~(3 c) の手続きを適用する。

(3 a) $\vec{x}^{(c)}$ を入力ユニット群に与え、出力 \vec{y} を計算する。

(3 b) このときの y_j を $w_{out,ji}$ を通して中間ユニットに伝搬する。

$$d_i = \sum_j w_{out,ji} y_j \tag{12}$$

(3 c) ここで、 $d_i \neq 0$ となる動作 i は動作 c と同じグループに属し、動作 c によって抑制される可能性がある。その抑制を表現するために、次式に従って $w_{diff,k,ij}$ を更新する。

$$\Delta w_{diff,k,ij} = \begin{cases} -1 & (\text{diff}_{ij} d_k \neq 0) \\ 0 & (\text{diff}_{ij} d_k = 0) \end{cases} \tag{13}$$

以上の手続きにより、並列動作生成部の出力を修正し、運動系への目標値に相当する表現形式に変換することが可能となる。

4. タイプライティング動作への応用

以上のモデルの応用例として、タイプライティング動作に関する運動スキーマの生成を試みた。

表 1 にタイプライティング動作の入出力を示す。並列動作生成部のプラン入力としては、 M 個の単語を想定する。また、並列動作生成部の出力動作(制約表現

部の入力となる)は、ある一つの文字を打けんする動作(アルファベット 26 文字)とする。一方、制約表現部の出力および教師信号としては、各指の作業座標における目標位置および打けん情報を考える。このとき、人差指は x 座標および y 座標が、中指から小指までは y 座標のみが必要となるので、右手と左手で合わせて 10 個の位置を出力する。表 2 は各指に対応するアルファベットである。1 本の指に対応する文字は並列に動作できない。ただし、実際に指をこの目標位置まで動かすことに関してはここでは扱わない。また、打けんユニットは指の打けんのタイミングを表現する。この打けんユニットの活性化値が 1 になったときに、実際の打けん動作が発現する。なお、並列動作生成部のパラメータは $\mu=0.8$, $\eta=1.0$, $\epsilon=0.01$ とし、隠れユニットは 52 個用意した。

以下に単語数 $M=10$ として行ったシミュレーション結果の一部を示す。用いた単語は、“ein”, “zwei”, “drei”, “vier”, “fuenf”, “sechs”, “sieben”, “acht”, “neun”, “zehn” という独語の 10 個の数詞である。まず、並列動作生成部で強化学習(3.1 参照)を行わずに学習を行った後の単語 “neun” に対する出力を示す。図 7(a) は並列動作生成部の出力ユニット、(b) は制約表現部の中間ユニット、(c) は制約表現部の出力ユニットの一部である。

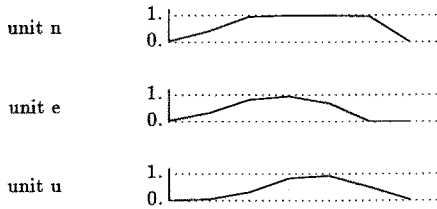
図 7(a)を見ると、ユニット ‘n’ の活性化値が 4 回続けて 1 となっている。この結果、制約表現部の中間ユニット群では、ユニット ‘n’ がユニット ‘u’ を抑制し(図 7(b))、誤った打けん動作系列を出力している(図 7(c))。そこで、並列動作生成部の学習則に基づき、強化学

表 1 タイプライティング動作の入出力

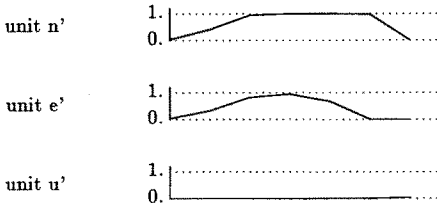
	並列動作生成部	制約表現部
入力	単語 (M 種類)	文字を打つ動作 (26種類)
出力	文字を打つ動作 (26種類)	指の位置 (10個), 打けん情報 (8 個)
教師信号	単語のスベリング	文字に対する指の位置

表 2 指と対応する文字

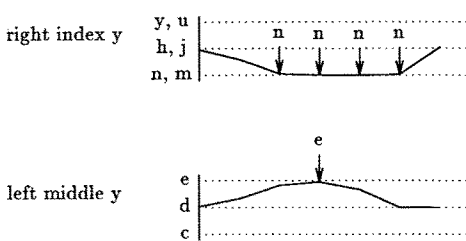
y	左 手				右 手				
	小指	薬指	中指	人差指 $x =$ 0 +1	人差指 $x =$ -1 0	中指	薬指	小指	
+1	q	w	e	r t	y u	i	o	p	
0	a	s	d	f g	h j	k	l		
-1	z	x	c	v b	n m				



(a) output units in the generation part of concurrent action sequences



(b) activation values of Middle units



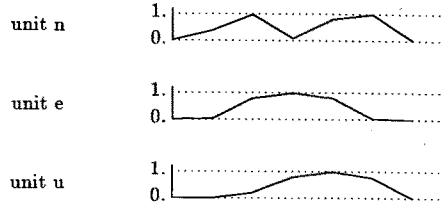
(c) output action sequences of the motor schema

図7 単語“neun”に対するシミュレーション結果 (強化学習なし)

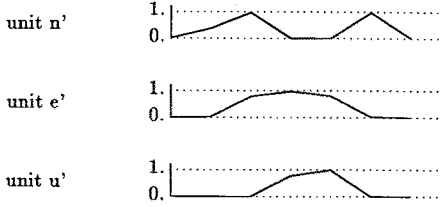
Fig. 7 Simulation results for a word “neun” (without reinforcement learning).

習を用いて再度シミュレーションを行った。学習後(約6,000回)の結果を図8に示す。強化学習の結果、並列動作生成部の出力ユニット‘n’は活性化値最大をとったのち一度減少し、最後に再び最大となっており、うまく学習が行われている(図8(a))。これにより、制約表現部の中間ユニット群では、図7(b)とは逆に、ユニット‘u’がユニット‘n’を抑制する部分が現れ、この結果、正しい打けん動作系列が出力されている。なお、他の9個の単語についても、正しい打けん動作系列を実現することができた。

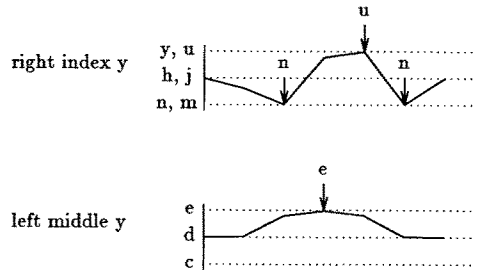
以上のシミュレーション結果より、本論文で提案したモデルはタイプライティング動作の直列性と並列性をうまく表現することができ、効率的な打けん動作系列を生成できることが明らかになった。本モデルでは、各単語に対応したネットワーク構造をあらかじめ用意



(a) output units in the generation part of concurrent action sequences



(b) activation values of Middle units



(c) output action sequences of the motor schema

図8 単語“neun”に対するシミュレーション結果 (強化学習あり)

Fig. 8 Simulation results for a word “neun” (with reinforcement learning).

する必要はなく、任意の単語を学習により記憶することができる。更に、制約表現部の学習が一度完了すれば、タイプライティング動作に関する運動系の拘束を一般的な形で表現することができ、あらゆる文字列に対応できるという利点も持っている。

5. むすび

本論文では、直列性と並列性を組み合わせた効率的な運動スキーマを、ニューラルネットを用いて実現することを試みた。そして、並列動作生成部と制約表現部に構造を分割したニューラルネットモデルおよびその学習則を提案し、タイプライティング動作の運動スキーマを構成した。

本モデルは、制約表現部をモジュールとして取り替

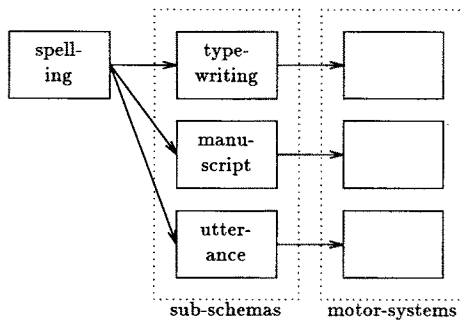


図9 運動スキーマの階層構造

Fig. 9 The hierarchical structure of motor schemas.

えることにより、複数の対象領域に対応することができる。例えば、並列動作生成部で単語のスペリングに応じた動作系列を生成し、制約表現部でその動作系列をタイプライティング、手書き、発声といった作業のための運動指令に修正する(図9)。逆に、ある対象領域に関する制約表現部を一度、学習的に獲得すれば、新たなプランが与えられても並列動作生成部を学習するだけで済む。このようなプランニング機能の柔軟性が、ネットワークの構造化により得られたのである。

しかしながら、いくつかの問題点が本モデルには残されている。

(1) 並列動作生成部の学習に時間がかかり、学習がローカルミニアに陥る可能性がある。これは、誤差逆伝搬法を用いるモデルに共通して言うことである。本論文では、学習則に強化学習を導入することで、この問題をある程度解決した。しかしながら、この学習が常に最適解に収束する保証はなく、今後、更に優れた学習則を検討する必要がある。

(2) 制約表現部で実現できる制約の種類が限られている。本論文では、問題領域からの制約を表現する際、その制約の性質をモデルの構造にあらかじめ埋め込むことによって実現した。これにより、かなり学習が簡単化できたが、逆に対象とする問題領域がかなり限定される。さまざまな問題領域からの制約を表現できるようなモデル構造を考察する必要がある。

(3) プランに含まれる動作の順序を入れ替えることができない。本論文では、プランの目的として動作の順序が決まっているものを扱い動作の順序を入れ替えることは考慮していない。

今後、より広い範囲の運動スキーマを実現し、ロボット制御等の問題に応用していくためには、このような問題を考察する必要がある。

謝辞 本研究の一部は文部省科学研究費(62460142, 63750415)の研究助成によるもので、ここに改めて謝意を表します。

文 献

- (1) M. A. Arbib: "Schema and Perception: Perspectives from Brain Theory and Artificial Intelligence", Pattern Recognition by Humans and Machines Vol. 2, eds. E. C. Schewb and H. C. Nusbaum, pp. 121-157, Academic Press, New York (1986).
- (2) D. E. Rumelhart and J. L. McClelland: "Parallel Distributed Processing, Vol. 1 & 2", MIT Press, Cambridge (1986).
- (3) 麻生秀樹: "ニューラルネットワーク情報処理", 産業図書(昭63).
- (4) 國藤 進: "コネクショニストモデルとその周辺", 計測と制御, 27, 10, pp. 935-938 (昭63-10).
- (5) D. E. Rumelhart, G. E. Hinton and R. J. Williams: "Learning internal representations by error propagation", Parallel Distributed Processing, Vol. 1, eds. D. E. Rumelhart and J. L. McClelland, MIT Press, Cambridge (1986).
- (6) D. E. Rumelhart, P. Smolensky, J. L. McClelland and G. E. Hinton: "Schemata and Sequential Thought Processes in PDP Models", Parallel Distributed Processing, Vol. 2, eds. D. E. Rumelhart and J. L. McClelland, pp. 7-57, MIT Press, Cambridge (1986).
- (7) D. E. Rumelhart and D. A. Norman: "Simulating a skilled typist: A study of skilled cognitive-motor performance", Cognitive Sci., 6, 4, pp. 1-36 (1982).
- (8) M. I. Jordan: "Attractor dynamics and parallelism in a connectionists sequential machine", Proc. of the 8th Annual Conference of the Cognitive Science Society, pp. 531-546 (1986).
- (9) Y. Miyata: "Organization of action sequence in motor learning", Proc. 9th Annual Conference of the Cognitive Science Society, pp. 496-507 (1987).

(平成元年4月6日受付)



辻 敏夫

昭60 広島大学院工学研究科博士課程前期システム工学専攻了。同年同大学工学部助手、現在に至る。生体工学、ロボット工学、マン・マシンシステムに関する研究に従事。計測自動制御学会、電気学会、日本ロボット学会、日本人間工学会、IEEE

各会員。



石田 祐介

平成元年広島大大学院工学研究科博士課程前期情報工学専攻了。同年4月富士ゼロックス㈱入社。在学中、ニューラルネットワークに関する研究に従事。計測自動制御学会会員。



伊藤 宏司

昭44名大大学院工学研究科修士課程了。昭45同工学部自動制御研究施設助手。昭54広島大学工学部第2類(電気系)助教授。現在に至る。工博。生体システム、ロボティクス、マン・マシンインタフェースの研究に従事。電気学会論文賞受賞。計測自動制御学会、電気学会、日本エム・イー学会、日本ロボット学会、IEEE等各会員。



長町 三生

昭38広島大大学院博士課程了。同年文学博士。広島大学助手。助教授を経て現在工学部教授。情報工学、認知心理学、産業用ロボットと安全管理などの研究に従事。計測自動制御学会、日本人間工学会、情報処理学会、人工知能学会等各会員。



西野 達夫

昭51広島大大学院工学研究科修士課程了。同年広島工業大学経営工学科助手。昭60同大学経営工学科講師。現在に至る。認知情報処理、ニューラルネットワークに関する研究に従事。人工知能学会、日本人間工学会、日本経営工学会各会員。