

# スパース再帰神経回路モデルによる人物の行動学習

西出俊 (京都大学) 奥乃博 (京都大学) 尾形哲也 (早稲田大学)

## 1. はじめに

近年, スパースなニューロン間結合を持つネットワークを用いた学習系が注目されている。Jaeger らは大規模な再帰神経回路モデルを用いたスパースなエコーステートネットワーク(ESN)を構築し, カオス時系列の学習における有効性を示した[1]。ESN はネットワークの出力部への重みのみを学習することで収束の早い学習を実現しており, ネットワークの内部ニューロンをランダムで接続している。Jordan 型[2]の再帰神経回路(RNN)モデルにおけるスパースなネットワークの有効性も示されている[3]。[3]では離散的なアルファベットを入力することで文章を学習し, スパース化による学習性能向上が確認された。スパースなネットワークを用いる手法は動的な対象だけではなく, 画像などの静的な対象に対しても有効性が確認されている。Wang らは画像分類を行う際, 局所的な制約を用いることでスパースなネットワークを実現し, スパースコーディングによる学習性能向上を示した[4]。

我々はこれまで Jordan 型 RNN を用いたロボットの環境知覚・行動生成に関する研究を行ってきた[5]。Jordan 型 RNN を用いることにより, 下記のような利点が挙げられる。

1. 連続的な時系列データを学習可能。
2. モデルの解析が可能である。

本研究ではスパース化した Jordan 型 RNN を用い, 連続的な時系列データを学習することでより効率的な行動学習を目指している。本稿では様々な RNN の構成(ニューロン数など)でモデルのスパース化を行い, 学習性能と未学習動作に対する汎化性能を評価する。

## 2. MTRNN の概要

本研究では人物の行動学習モデルとして再帰神経回路モデルの一種である Multiple Timescales Recurrent Neural Network (MTRNN)[6] を用いる。

MTRNN(図1)は Jordan 型 RNN を改良したものであり, 2層で構成される時系列学習器である。各層は入出力ニューロン群  $S$ , Fast Context ニューロン群  $C_F$ , Slow Context ニューロン群  $C_S$  から成り, 現状態  $S(t)$  を入力することで次の状態  $S(t+1)$  を出力する予測器になっている。一般的な MTRNN は, 入出力ニューロン群と Slow Context ニューロン群間は連結せず, それ以外のニューロン間は全結合となっている。

MTRNN の各ニューロンには時定数が設定されており, 時定数が小さいほどニューロン値は時間変化と共に大きく変化する。入出力ニューロン群の時定数が最も小さく, Fast Context ニューロン群, Slow Context ニューロン群の順に時定数が大きくなる。時定数が無限大のニューロンは時間変化に対して一定の値を持つ。異なる時定数を持つニューロン群の組み合わせにより,

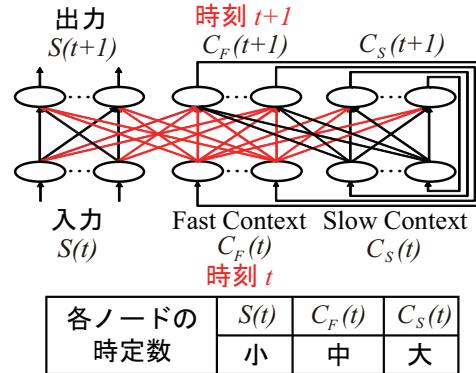


図 1 MTRNN の構成

MTRNN は情報を構造化して時系列を学習することが可能であり, 従来の RNN より高い学習性能を持つ。

### 2.1 MTRNN の学習

MTRNN の学習は Forward 計算と誤差逆伝搬(Back Propagation)によって行う。学習開始時, ニューロン間の結合重み,  $C_F(0), C_S(0)$  を乱数で初期化する。次に学習データ系列の各時刻における値を  $S(t)$  に入力し, 出力値  $S(t+1)$  を計算する(Forward 計算)。出力値と教師信号の誤差を入力に逆伝搬することでネットワークを更新する。Forward 計算と誤差逆伝搬を交互に繰り返し, 出力誤差が収束すると学習を終了する。

Forward 計算ではまず各ニューロンの内部値  $u_{in}(t)$  を以下の式によって計算する。

$$u_{in}(t) = \left(1 - \frac{1}{\tau_i}\right) u_{in}(t-1) + \frac{1}{\tau_i} \left[ \sum_{j \in N} w_{ij}(n) x_{jn}(t) \right]. \quad (1)$$

ここで,  $\tau_i$  はニューロン  $i$  の時定数を,  $w_{ij}(n)$  は  $n$  番目の計算ループにおけるニューロン  $j$  からニューロン  $i$  への重み値を,  $x_{jn}(t)$  はニューロン  $j$  の入力値を表す。ニューロン  $i$  の出力値  $y_{in}$  は内部値  $u_{in}(t)$  にシグモイド関数を適用して計算する。

$$y_{in}(t) = \frac{1}{1 + \exp(-u_{in})}. \quad (2)$$

入力値  $x_i(t)$  は教師信号  $T_i(t)$  と 1 時刻前の出力値  $y_i(t-1)$  を用い, 以下のように計算する。

$$x_i(t) = \begin{cases} 0.9 \times y_i(t-1) + 0.1 \times T_i(t) & i \in F \\ y_i(t-1) & \text{otherwise} \end{cases}, \quad (3)$$

式(3)のように 1 時刻前の出力値を教師信号にフィードバックして入力値を計算することにより, 学習誤差が急激に増加することを防ぐことができる。 $C_F$  と  $C_S$

ノードについては 1 時刻前の出力値をそのまま入力値として入力する。

誤差逆伝搬では、Forward 計算で求めた出力値を用い、出力誤差を逆伝搬することで MTRNN の結合重みと  $C_S$  初期値  $C_S(0)$  を更新する。まず、各時刻の出力誤差（教師信号と出力値の差分）の総和を学習誤差  $E(n)$  として求める。

$$E(n) = \sum_t \sum_i (y_{in}(t-1) - T_i(t))^2. \quad (4)$$

第  $i$  ニューロンから第  $j$  ニューロンへの結合重み  $w_{ij}$  は学習係数  $\alpha$  を用いて下記のように計算される。

$$w_{ij}(n+1) = w_{ij}(n) - \alpha \frac{\partial E(n)}{\partial w_{ij}}. \quad (5)$$

学習時、 $C_S$  の初期値である  $C_S(0)$  も式 (5) と同様に更新する。

MTRNN の学習は Forward 計算と誤差逆伝搬を繰り返し行い、学習誤差  $E(n)$  が収束するまで行う。本稿では  $n$  を繰り返し計算を行う計算ループとして定式化している。

## 2.2 MTRNN による行動認識と生成

MTRNN の学習により、各学習時系列データに対応する  $C_S(0)$  が得られる。 $C_S(0)$  は時系列を表現しており、MTRNN を用いることで  $C_S(0)$  と時系列を相互に変換することができる。

時系列から  $C_S(0)$  を求める過程は認識と呼ばれる。時系列を認識する際、前節の学習方法で示したものと同様に Forward 計算と逆誤差伝搬を用いる。ただし学習時と異なり、認識時は結合重みを更新せず、 $C_S(0)$  のみ更新する。収束によって得られた  $C_S(0)$  が MTRNN によって認識された時系列となる。

$C_S(0)$  から時系列を求める過程は生成と呼ばれる。生成には Open loop 生成と Closed Loop 生成がある。Open loop 生成は  $C_S(0)$  と各時刻の状態  $S(t)$  が与えられている時に MTRNN から時系列を再現する方法である。計算は前節で述べた Forward 計算と同様の手法で計算する。Closed loop 生成は  $C_S(0)$  と初期状態  $S(0)$  から時系列を計算する方法であり、状態  $S(t)$  と Fast Context  $C_F(t)$ 、Slow Context  $C_S(t)$  の逐次入力で計算する。まず初期状態  $S(0)$  と  $C_S(0)$  を入力し、次時刻の  $S(1)$ 、 $C_F(1)$ 、 $C_S(1)$  を出力として得る。出力を再度 MTRNN に入力し、計算を繰り返し行うことで時系列全体を計算する。

## 3. 評価実験

### 3.1 実験設定

本稿では MTRNN のスパース化を評価するために、[7] で用いられた人物の行動データ系列を用いてモデルを学習する。データ系列は人物の行動をモーションキャプチャで取得されたものである。行動は足踏み、バンザイ、踊り、蹴り、パンチ、四股、スクワット、投げる、前屈の 9 種で 20 自由度で構成される。各データのデータ数とデータ長は表 1 の通りである。

本実験では  $(C_F, C_S)$  は  $(40, 2)$ ,  $(40, 5)$ ,  $(40, 10)$ ,  $(80, 2)$ ,  $(80, 5)$ ,  $(80, 10)$  の 6 通りでモデルを学習した。

表 1 データ数とデータ長

データ種類	データ数	データ長
足踏み	28	66 ~ 77
バンザイ	15	80 ~ 103
踊り	7	124 ~ 134
蹴り	19	58 ~ 68
パンチ	14	96 ~ 108
四股	13	112 ~ 140
スクワット	13	95 ~ 109
投げる	13	119 ~ 142
前屈	15	86 ~ 111
計	137	58 ~ 142

各  $(C_F, C_S)$  の構成に対し、図 1 の赤色の結合をランダムで 10%, 20%, …, 100% (全結合) だけ残すことで MTRNN をスパース化したもので学習した。本稿では 10% や 20% などの結合の割合を結合スパースネスと呼ぶ。図 1 の赤色で示したもの以外の結合については、低い結合スパースネスでは他のニューロンとの結合がないニューロンができる可能性があるため、黒色の結合は全結合とした。それぞれの結合スパースネスについて、10 種類の異なる乱数シードを用いて初期結合重み、 $C_F(0)$ ,  $C_S(0)$  を決定し、10 回ずつ学習した。各ニューロンの時定数については、入力ニューロンは 2,  $C_F$  ニューロンは 5,  $C_S$  ニューロンは 70 とした。時定数は過去の実験経験から学習効率がよいものを設定した。

### 3.2 評価方法

MTRNN の学習性能を評価するために、本稿では学習後の MTRNN を用いて評価系列の認識、生成性能を評価する。まず、MTRNN を用いて評価系列をそれぞれ認識し、各評価系列に対応した  $C_S(0)$  を求める。認識によって得られた  $C_S(0)$  を用いて評価系列を Open loop 生成し、その時の生成誤差  $G$  (評価系列の各時刻の値と出力値の絶対誤差和) を式 (6) のように計算する。

$$G = \sum_t \sum_i (y_i(t-1) - F_i(t))^2. \quad (6)$$

ここで、 $y_i(t)$  はステップ  $t$  における出力ニューロン  $i$  の値であり、 $F_i(t)$  は評価系列の時刻  $t$  におけるニューロン  $i$  に対応する値である。生成誤差  $G$  が小さいほど MTRNN は評価系列を正確に表現しているといえる。

本実験では評価系列を

1. 学習に用いた系列 (学習系列)
2. 学習に用いていない系列 (未学習系列)

の二つのグループに分ける。それぞれのグループで生成誤差  $G$  の平均・分散を  $(C_F, C_S)$ 、結合スパースネスごとに計算する。同じ  $(C_F, C_S)$  の中で学習系列の生成誤差平均が最も小さい結合スパースネスを  $(C_F, C_S)$  で最も学習効率がよい結合スパースネス ( $\alpha\%$ ) とする。

スパース化したモデルの評価として、下記の二種類の評価を行う。

1. 結合スパースネス  $\alpha\%$  と全結合 (結合スパースネス 100%) の性能比較

## 2. 結合スパースネス $\alpha\%$ におけるモデルの汎化性能評価

1. について、結合スパースネス  $\alpha\%$  のモデルと全結合（結合スパースネス 100%）のモデルの学習系列の生成誤差平均間の有意差の有無を有意水準 1% の t 検定によって検定する。両者の間に有意差が認められる場合は結合スパースネス  $\alpha\%$  と全結合モデルの生成誤差平均の間に差があり、MTRNN の結合をスパース化することの有効性が統計的に示されたことを意味する。2. について、結合スパースネス  $\alpha\%$  のモデルにおける学習系列と未学習系列の生成誤差間の有意差の有無を有意水準 1% の t 検定（F 検定によって分散が等しい判定された場合）、あるいはウェルチの t 検定（F 検定で分散が異なると判定された場合）で検定する。両者の間に有意差が認められない場合は学習系列と未学習系列をモデルが同等に扱えていると考えることができ、未学習系列も汎化できていると考えられる。

### 3.3 実験結果

前節で述べた評価結果を図 2 に示す。図 2 では、 $(C_F, C_S)$  ごとに生成誤差平均をまとめており、標準偏差を縦棒でグラフに描画した。各  $(C_F, C_S)$  について左から順に最も学習性能がよかった  $\alpha\%$  の結合スパースネスにおける学習データの生成誤差平均（青色グラフ）、同結合スパースネスにおける未学習データの生成誤差平均（赤色グラフ）、全結合モデルにおける学習データの生成誤差平均（緑色グラフ）を描画した。前節で述べた検定 1. と 2. の結果は各グラフの下の印の有無で示している。印がついているものは結合スパースネス  $\alpha\%$  のモデルの生成誤差と比べ、統計的に有意差があることを意味している。緑色グラフに対する印が検定 1. の結果であり、赤グラフに対する印が検定 2. の結果である。

まず、最適な結合スパースネス  $\alpha$  と  $C_F, C_S$  の関係について述べる。本実験の結果、 $\alpha$  は  $C_S$  が 2 の時は 90,  $C_S$  が 5 の時は 80,  $C_S$  が 10 の時は 90 となった。このように  $\alpha$  は  $C_F$  よりも  $C_S$  のニューロン数に影響を受け、 $C_S$  ニューロン数が増えるにつれ最適な結合スパースネスは下がる傾向があることが示された。

検定 1. の結果については、青色グラフと緑色グラフ

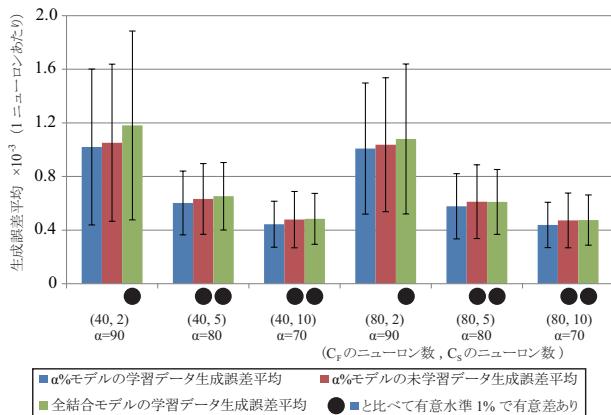


図 2 生成誤差平均

の生成誤差の有意差に注目する。全ての  $(C_F, C_S)$  組み合わせについて印がついており、統計的に有意差が認められる結果になった。これにより、最適な結合スパースネスのモデルを用いることで全結合のモデルより効率的な学習が可能であることが示された。

検定 2. の結果については、青色グラフと赤色グラフの生成誤差の有意差に注目する。図 2 より、 $C_S$  が 2 の時には印がなく、統計的に学習データと未学習データの生成誤差の間に有意差が認められなかった。 $C_S$  が 5, 10 の時には印があり、統計的に学習データと未学習データの生成誤差の間に有意差が認められる結果になった。前節で定義した汎化能力について、 $C_S$  のニューロン数が 2 の時は未学習データが汎化できており、 $C_S$  のニューロン数が 5, 10 の時は汎化できていないという結果になった。

最後に、学習した MTRNN を用いて動作を認識することで得られる  $C_S(0)$  の分布を用いての動作分類性能を調べる。 $(C_F, C_S)$  が (80, 2) のスパースネス 90% の時の  $C_S(0)$  空間を図 3 に示す。本稿では  $C_S$  のニューロン数が 2 の場合は 2 次元で  $C_S(0)$  空間を描画することができ、視覚的に解析することが容易であるため、一例として  $(C_F, C_S)$  が (80, 2) の場合について評価する。 $C_S(0)$  空間とはモデル学習後に学習・未学習系列を認識することで求められた  $C_S(0)$  の値を描画したものである。 $C_S$  ニューロン数は 2 であるので、横軸に第 1 の  $C_S$  ニューロンの初期値 ( $C_{S1}(0)$ ) を、縦軸に第 2 の  $C_S$  ニューロンの初期値 ( $C_{S2}(0)$ ) をとる。図 3 において、青色で描画した項目は学習系列を表しており、赤色で描画した項目は未学習系列を表している。図 3 より、各項目は動作ごとに学習・未学習系列間わずクラスタを形成しており、モデルのスパース化を行っても動作の分類は  $C_S(0)$  空間にによって可能であることを示している。

### 4. 考察

本章では実験の結果を考察する。

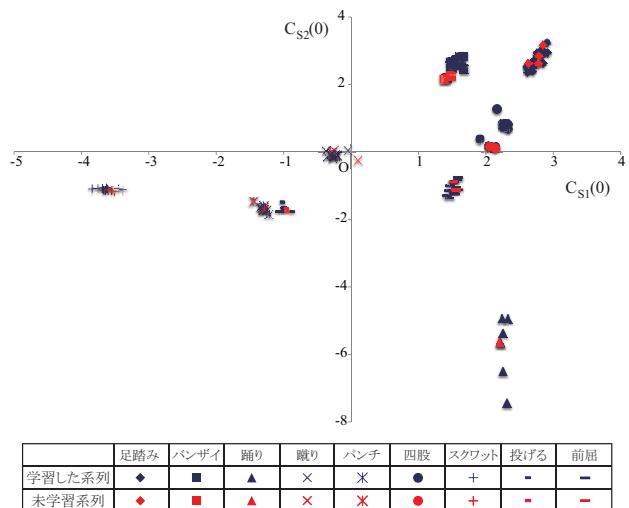


図 3 学習した MTRNN の  $C_S(0)$  空間

#### 4.1 モデル構成に対する最適なスパースネス

本実験では  $C_S$  ニューロン数が 2 の時には 90% , 5 個の時には 80% , 10 個の時には 70% が最適な結合スパースネスとなり ,  $C_F$  ニューロンの個数に影響されず ,  $C_S$  ニューロンの個数によってのみ最適な結合スパースネスが決定される結果になった . また , 図 2 の生成誤差に注目しても ,  $C_S$  ニューロン数によって大きく変わつてあり ,  $C_F$  ニューロン数の影響はほとんど受けていることが分かる . 以上からモデルの結合スパースネスを決定する際 , モデルの構成は  $C_F$  ニューロン数よりも  $C_S$  ニューロン数に注目する必要があると考えることができる .

#### 4.2 モデル構成と汎化能力

本実験では  $C_S$  ニューロン数が 2 の時のみ未学習系列が汎化できており , 5 と 10 の時には未学習データが汎化できなかったという結果になった . しかし ,  $C_S$  ニューロン数が 2 の時と比べ , 5 の時は生成誤差平均が約半分になっており , 分散も小さくなっていることが分かる .  $C_S$  のニューロン数が 10 の時はさらに生成誤差平均・分散も小さくなっている . このことから  $C_S$  のニューロン数によって生成誤差と汎化能力間のトレードオフが生じることが分かる .

本実験で  $C_S$  ニューロン数が大きい場合に未学習データを汎化できなかったのは  $C_S$  ニューロン数が多いことによる過学習の影響と  $C_S$  ニューロンから出る結合をスパース化しなかったために過学習を抑制できなかったこと原因として考えられる . これらの結合もスパース化した場合の性能評価は今後の課題として残されている .

#### 4.3 動作の種類と学習性能

本実験では , 学習動作系列に対して全ての ( $C_F$ ,  $C_S$ ) 構成においてスパース化の有効性を統計的に示した . 評価は全動作の平均と分散で行ったが , 本節では一つ一つの動作に注目し , 動作の種類とスパース化の有効性の関係について考察する .

動作ごとの評価は検定 1. と同様の方法で行い , 動作ごとに全結合モデルと比較する . まず , 各動作の生成誤差平均を式 (6) と同様に求め , 動作ごとに最適な結合スパースネス ( 最も生成誤差平均が小さくなるスパースネス ) を求めた . 次に動作ごとに , 最適な結合スパースネスにおける学習動作の生成誤差平均と全結合モデルにおける学習動作の生成誤差平均の間に有意差があるかどうかを  $t$  検定で検定した .

検定の結果 , 54 通りのうち ( 全 9 動作  $\times$  6 モデル構成 = 54 通り ) , バンザイ ( $C_F, C_S$ ) = (80, 2) , バンザイ ( $C_F, C_S$ ) = (80, 5) , 四股 ( $C_F, C_S$ ) = (80, 2) , 前屈 ( $C_F, C_S$ ) = (40, 10) , 前屈 ( $C_F, C_S$ ) = (80, 5) , 前屈 ( $C_F, C_S$ ) = (80, 10) の 6 通りにおいて有意差がない ( スパース化の有効性が確認できない ) という結果になった . 特に前屈動作は 6 種の ( $C_F$ ,  $C_S$ ) 構成のうち 3 通りでスパース化の有効性が統計的に見られなく , 本結果はスパース化が有効な動作系列とそれほど有効ではない動作系列が存在することを示唆している . 今後は動作 ( 学習系列 ) の複雑さとスパース化の関係性についても解析していきたい .

#### 5. おわりに

本稿では再帰神経回路モデルの一種である MTRNN を用い , 人間動作を学習する際の結合スパース化の有効性を確認した . 実験は足踏み , バンザイ , 踊り , 踊り , バンチ , 四股 , スクワット , 投げる , 前屈の 9 種類の動作を用いて行い , 6 種の MTRNN モデル構成全てにおいてスパース化による学習性能向上を統計的に示した . 性能評価はモデルの生成誤差をもとに行い ,  $C_S$  ニューロン数が多くなるほど誤差は下がるが汎化性能が悪くなることが示された . また , ネットワークをスパース化しても  $C_S(0)$  空間によって動作の分類が可能であることを確認した .

今後は  $C_S$  ニューロン群の結合をスパース化することによる学習性能を評価し , 学習系列の複雑さとスパース化の関係性について解析する予定である . その結果をもとに自動的なニューロン間結合の切断・連結アルゴリズムによる効率的な学習モデル構築 , さらにはロボットの動作学習に応用していきたいと考えている .

#### 6. 謝辞

本研究は JST さきがけ領域「情報環境と人」 , 文科省科研費新学術領域研究「構成論的発達科学」( No. 24119003 ) , 科研費若手研究 B ( No. 25730159 ) , 柏森情報科学振興財団 , 立石科学技術振興財団の助成を受けた .

#### 参考文献

- [1] H. Jaeger, H. Haas, "Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication," *Science*, Vol. 304, No. 5667, pp. 78-80, 2004.
- [2] M. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine," *Eighth Annual Conference of the Cognitive Science Society*, pp. 513-546, 1986.
- [3] H. Awano, S. Nishide, H. Arie, J. Tani, T. Takahashi, H. G. Okuno, T. Ogata, "Use of a Sparse Structure to Improve Learning Performance of Recurrent Neural Networks," B.-L. Lu, L. Zhang, and J. Kwok (Eds.), *International Conference on Neural Information Processing*, Part III, LNCS 7064, pp. 323-331, Springer-Verlag Berlin Heidelberg, 2011.
- [4] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained Linear Coding for Image Classification," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 3360-3367, 2010.
- [5] S. Nishide, J. Tani, T. Takahashi, H. G. Okuno, and T. Ogata, "Tool-Body Assimilation of Humanoid Robot using a Neurodynamical System," *IEEE Transactions on Autonomous Mental Development*, Vol. 4, Issue 2, pp. 139-149, 2012.
- [6] Y. Yamashita and J. Tani, "Emergence of Functional Hierarchy in a Multiple Timescale Neural Network Model: a Humanoid Robot Experiment," *PLoS Computational Biology*, Vol. 4, No. 11, 2008.
- [7] D. Kulić, W. Takano, and Y. Nakamura, "Incremental Learning, Clustering and Hierarchy Formation of Whole Body Motion Patterns using Adaptive Hidden Markov Chains," *International Journal of Robotics Research*, Vol. 27, Issue 7, pp. 761-784, 2008.