パルスニューラルネットワークにおける 破局的な忘却の抑制を考慮したヘブ型学習則

学生員 元木 誠* 正員 濱上 知樹*
正員 小坪 成一* 正員 平田 廣則*

Hebbian Learning Rule Restraining Catastrophic Forgetting in Pulse Neural Network

Makoto Motoki*, Student Member, Tomoki Hamagami*, Member, Seiichi Koakutsu*, Member,
Hironori Hirata*, Member

In this paper, a Hebbian learning rule restraining “catastrophic forgetting” is proposed on pulse neural network (PNN) with leaky integrate-and-fire neurons. The strong point of this learning rule is that a learning of new pattern does not destroy past ones, and that an efficient use of synapses is enabled. First, in order to consider the function of the learning rule, a fundamental experiment is made. Next, to compare the performance between the proposed learning rule and conventional ones on the application, simulation experiments are examined using autonomous behavior robots which are forced to learn concurrently two different environments. The results of the experiments show that the proposed learning rule clearly restrains “catastrophic forgetting” and enables working of more efficient than conventional PNN learning.

キーワード：パルスニューラルネットワーク，ヘブ型学習則，破局的な忘却

Keywords: pulse neural network, Hebbian learning, catastrophic forgetting

1. まえがき

近年，ニューロンの時空間的な振る舞いが，ニューロン集団の機能の決定に重要な役割を果たすと考えられている。特にスパイクの発火タイミングや発火頻度による情報処理はテンポラルコーディングと総称され，パルスニューラルネットワーク（Pulse Neural Network：PNN）によるテンポラルコーディングに関する研究がさかに行われている^{(1)～(4)}。

PNNは従来の出力にシグモイド素子等のアナログニューロンを用いるネットワークに比べ時系列処理に適している。例えば，PNNをロボットのコントローラに用いる場合，ネットワークへの入力である各センサからの値の時間的変化を考慮した出力が可能となる。その結果，時間軸を考慮した制御が可能になる。対して，アナログニューロンを用いる場合，ネットワークの構造をリカレント型にするか，ネットワークへの入力として数ステップ前までのセンサの値を一緒に入力しなければ，時間的変化を考慮した出力を行うことができない。また，ネットワークをハードウェア化する

場合，アナログニューロンを用いるネットワークに比べ，PNNは演算が簡易であり搭載しやすいというメリットもある⁽⁵⁾。

従来のニューロンモデルと同様にPNNにもヘブ型学習則が提案されている。Gerstnerら⁽¹⁾は，減衰のある積分発火モデルのPNNに結合荷重を調節するヘブ型学習則を応用することで，ネットワークが一定周波数の音のデコードとなり得ることを示している。また，Eurichら⁽²⁾や櫛谷ら⁽³⁾は前ニューロンで発生したパルスが後ニューロンに到達するまでの時間遅れを調節することでも発火タイミングの学習が可能であることを示している。

しかし，これらのヘブ型学習則は，過去に記憶した時系列パターンを破壊してしまう「破局的な忘却」が生じてしまうことが知られている^{(6)～(13)}。そこで本論文では，PNNにおける破局的な忘却の抑制を考慮したヘブ型学習則を提案する。

また，PNNの応用として，PNNを自律移動ロボットのコントローラに用いる自律移動ロボットの走行実験を行う。具体的には，自律移動ロボットに異なる2種類の環境中を走行させ，それぞれのスタート地点からゴール地点まで移動させることをロボットのタスクとする計算機実験を行う。そして，従来法のヘブ型学習則を使用するPNNと，提案するヘブ型学習則を使用するPNNをそれぞれコントロー

* 千葉大学

〒263-8522 千葉市稲毛区弥生町1-33

Chiba University

1-33, Yayoicho, Inage-ku, Chiba 263-8522

ラに用いた場合のロボットの振る舞いを比較する。

以降, 2. では本研究で用いる PNN の基本的動作, 従来法のヘブ型学習則について, 3. では提案手法である破局的な忘却の抑制を考慮したヘブ型学習則について, 4. では, PNN の応用として, 自律移動ロボットの自律走行実験を行い, その結果と考察を述べる。最後に 5. で本研究を総括し, 今後の課題を述べる。

2. パルスニューラルネットワーク (PNN)

ここでは, 本論文で用いる PNN の基本的動作, ヘブ型学習則とその問題点について述べる。

〈2・1〉 基本動作 本論文で用いるニューロンモデルは, 図 1 に示すような減衰のある積分発火ニューロンとする。なお, 本論文では, 計算機によるソフトウェア実装を前提にしている。そのため, 以降の関係式は離散化した式で表現している。

ニューロン j をシナプス前ニューロン, ニューロン i をシナプス後ニューロンとする。ニューロン j が時刻 t_0^j で発火したとすると, ニューロン i にパルスが到達する時刻 $t_d^{i,j}$ は式 (1) のように表せる。

$$t_d^{i,j} = t_0^j + \tau_d^{i,j}(t_0^j) \dots \dots \dots (1)$$

ここで, $\tau_d^{i,j}(t_0^j)$ は時刻 t_0^j にニューロン j で発生したパルスがニューロン i に到達するまでの時間遅れである。

パルスが到達すると, ニューロン i の膜電位 $P_m^i(t)$ は, 式 (2) により変化する。

$$P_m^i(t) = \mathcal{H}(t - t_\gamma^i) \left\{ \frac{\tau_m^i}{\tau_m^i + \Delta t} P_m^i(t - \Delta t) + \frac{\Delta t}{\tau_m^i + \Delta t} \sum_{j \in J} w_0^{i,j}(t) \delta(t - t_d^{i,j}) \right\} \dots \dots (2)$$

$$t_\gamma^i = t_0^i + \tau_\gamma^i \dots \dots \dots (3)$$

ここで, τ_m^i は減衰時定数, Δt は時間の刻み幅, J はニューロン i に接続する前ニューロンの集合, $w_0^{i,j}(t)$ は結合荷重, τ_γ^i は不応期, $\mathcal{H}(t)$ はヘビサイド関数, $\delta(t)$ は Dirac のデルタ関数である。

時刻 t において, 膜電位 $P_m^i(t)$ が θ^i を越えると, 式 (4) によりニューロン i は発火し, 式 (5) に従い発火時刻 t_0^i を更新する。また, ニューロン i が発火した場合, 膜電位 $P_m^i(t)$ は, 不応期 τ_γ^i の間 0 となる。

$$y^i(t) = \mathcal{H}(P_m^i(t) - \theta^i) \dots \dots \dots (4)$$

ここで, $y^i(t)$ はニューロン i の出力, θ^i は閾値である。

$$t_0^i = \begin{cases} t & \text{for } y^i(t) = 1 \\ \text{no} & \text{for } y^i(t) = 0 \\ \text{changed} & \end{cases} \dots \dots \dots (5)$$

〈2・2〉 ヘブ型学習則 実ニューロンにおける長期的な伝達効率の変化は, シナプス前ニューロンとシナプス後ニューロンの活動電位発生のタイミングに強く依存してい

図 1 ニューロンモデル

Fig. 1. Neuron model.

図 2 学習窓 $W_d(s)$

Fig. 2. A learning window function.

ることが知られている⁽¹⁴⁾。ヘブ型学習則はこの性質に基づく手法である。PNN におけるヘブ型学習則は前ニューロンと後ニューロンの活動相関に依存して, 結合荷重 $w_0^{i,j}(t)$ や信号の伝達遅れ $\tau_d^{i,j}(t)$ 等のパラメータの調節を行う。

Eurich らの時間遅れを調節する学習則について説明する⁽²⁾。ネットワークの構造は入力層と出力層の 2 層構造とし, 入力層のニューロンは周期 T で発火する。ニューロン i (出力層のニューロン) の発火した時刻を t_0^i , ニューロン j (入力層のニューロン) のパルスがニューロン i に到達した時刻を $t_d^{i,j}$ とすると, 時間遅れ $\tau_d^{i,j}(t)$ は式 (6) で更新される。

$$\tau_d^{i,j}(t + \Delta t) = \tau_d^{i,j}(t) + \alpha_d^{i,j} W_d(t_0^i - t_d^{i,j}) y^j(t) \dots \dots \dots (6)$$

$$W_d(s) = \frac{2s/T}{1 + (2s/T)^2} \dots \dots \dots (7)$$

ここで, $\alpha_d^{i,j}$ は学習率, $W_d(s)$ は図 2 に示すような学習窓である。

続いて, Gerstner らの結合荷重を調節するヘブ型学習則を説明する⁽¹⁾。ネットワークの構造は入力層と出力層の 2 層構造とし, 入力層のニューロンは周期 T で発火するとする。ニューロン i (出力層のニューロン) の発火した時刻を t_0^i , ニューロン j (入力層のニューロン) のパルスがニューロン i に到達した時刻を $t_d^{i,j}$ とすると, 結合荷重 $w_0^{i,j}(t)$ は

式 (8) で更新される。

$$w_0^{i,j}(t + \Delta t) = w_0^{i,j}(t) + \alpha_w^{i,j} W_w(t_0^i - t_d^{i,j}) y^i(t) \quad (8)$$

$$W_w(s) = \begin{cases} 0.3 \exp\left(\frac{s+0.05}{0.5}\right) & \text{for } s < -0.05 \\ 0.5 \exp\left(-\frac{s+0.05}{0.5}\right) & \\ -0.2 \exp\left(-\frac{s+0.05}{5}\right) & \\ \text{for } s \geq -0.05 \end{cases} \quad (9)$$

ここで、 $\alpha_w^{i,j}$ は学習率、 $W_w(s)$ は学習窓である。

〈2・3〉 従来法の問題点 結合荷重を調節するヘブ型学習則は前ニューロンの集合 J から、結合荷重の値に関係なく後ニューロン i の発火にもっとも寄与している前ニューロン j との結合荷重を強め、発火にさほど関与していない前ニューロンとの結合荷重を弱める。そのため、前ニューロンの集合 J から後ニューロン i への入力パターンが変わり、後ニューロン i の発火時刻がわずかに違う場合等に、過去に後ニューロン i の発火にもっとも寄与していたシナプスの結合荷重も変化させてしまう可能性がある。すなわち、過去に記憶した時系列パターンが破壊されてしまう「破局的な忘却」^{(6)~(13)} が生じる可能性がある。これは時間遅れを調節するヘブ型学習則においても同様の問題が生じる。

3. 破局的な忘却の抑制を考慮したヘブ型学習則

ここでは、提案手法である破局的な忘却の抑制を考慮したヘブ型学習則と提案手法における記憶容量の低下を防ぐための時間遅れを調節するヘブ型学習則について述べる。また、提案手法が破局的な忘却を抑制することができるかについての予備実験について述べる。

〈3・1〉 破局的な忘却の抑制を考慮した結合荷重を調節するヘブ型学習則 結合荷重を調節するヘブ型学習則では、結合荷重の値が大きいシナプスは、過去の記憶に関係している可能性が大きい。結合荷重の値が大きいものほど、新たな結合荷重の調節を行わないようにすれば、破局的な忘却を抑制しつつ、新たなパターンを記憶することが可能になる。

一方、実ニューロンでは、前ニューロンの活動電位に同期して後ニューロンにも通電により活動電位を発生させること（ベアリング）を数十回繰り返すと、前ニューロンの活動電位が少し先行する場合に伝達効率の長期増強が生じ、シナプス後電位が大きくなる。逆に後ニューロンの活動電位が少し先行する場合は長期抑圧が生じ、シナプス後電位が小さくなる⁽¹⁴⁾。

以上のことをふまえ、破局的な忘却の抑制を考慮した後ニューロン i と前ニューロン j との結合荷重 $w_0^{i,j}(t)$ を調節するヘブ型学習則は式 (10) のようになる。

$$w_0^{i,j}(t + \Delta t) = w_0^{i,j}(t) + \alpha^{i,j} \left\{ 1.0 - \Gamma\left(w_0^{i,j}(t)\right) \right\} y^i(t) W_p(t_0^i - t_d^{i,j}) \quad (10)$$

図3 提案手法の学習窓 $W_p(s)$

Fig.3. A learning window function in proposed method.

$$W_p(s) = \frac{2s/\tau_\gamma^i}{1 + (2s/\tau_\gamma^i)^2} + \frac{\delta(s)}{2} \quad (11)$$

ここで、 $\alpha^{i,j}$ は学習率、 $y^i(t)$ は時刻 t でのニューロン i の出力、 t_0^i はニューロン i の発火時刻、 $t_d^{i,j}$ はニューロン j のパルスがニューロン i に到達する時刻、 $W_p(s)$ は図3に示す学習窓、 τ_γ^i はニューロン i の不応期、 $\delta(t)$ は Dirac のデルタ関数である。 $\Gamma\left(w_0^{i,j}(t)\right)$ は $w_0^{i,j}(t)$ の関数であり、本論文では $\Gamma\left(w_0^{i,j}(t)\right) = \left\{w_0^{i,j}(t)\right\}^2$ とする。また、結合荷重 $w_0^{i,j}(t)$ の値域を本論文では、 $0 \leq w_0^{i,j}(t) \leq 1$ とする。

式 (10) において、 $\left\{1.0 - \Gamma\left(w_0^{i,j}(t)\right)\right\}$ とすることで、結合荷重 $w_0^{i,j}$ が小さいシナプスほど、大きく値が変化する。活動電位の発生にさほど関与していないシナプスを積極的に更新することで、破局的な忘却を抑制しつつ、新たな発火タイミングの学習が可能となる。また、関数 $\Gamma\left(w_0^{i,j}(t)\right)$ を $\left\{w_0^{i,j}(t)\right\}^\sigma$ とすると、 $w_0^{i,j}$ の値が大きいとき、 σ が小さい場合に比べ σ が大きい場合の方が、関数 $\Gamma\left(w_0^{i,j}(t)\right)$ の値が大きくなる。つまり、 σ が大きくなるにつれ、活動電位の発生にある程度寄与しているシナプスの結合荷重も更新されるようになる。

〈3・2〉 提案手法における記憶容量の低下を防ぐための時間遅れを調節するヘブ型学習則 提案手法は、活動電位の発生にさほど関与していないシナプスを積極的に更新することで、破局的な忘却を抑制しつつ、新たな発火タイミングの学習を可能としている。しかし、実質的には、後ニューロンの発火に不必要な前ニューロンとの結合荷重も増加させてしまう。そのため、学習した発火タイミングが多くなるにつれ、全体的に結合荷重が大きくなる傾向がある。その結果、従来法に比べ、相対的に記憶容量が低下してしまう問題が生じる。

一方、実ニューロンでは、軸索の直径が増加するとパルスの伝達速度が上昇することが知られている⁽¹⁵⁾。長期的な伝達効率の変化に伴い、軸索の直径が変化すると仮定すると、前ニューロン j で発生したパルスが後ニューロン i に到達するまでの時間遅れ $\tau_d^{i,j}$ を調節する学習則は式 (12) のようになる。

表 1 PNN のパラメータ

Table 1. Parameters for PNN.

ネットワークの時間の刻み幅	Δt	0.1
入力層のニューロンの発火周期 $[\Delta t]$	T	100
閾値	θ^i	0.05
減衰時定数 $[\Delta t]$	τ_m^i	15
不応期 $[\Delta t]$	τ_r^i	50
ニューロン i, j 間の初期時間遅れ $[\Delta t]$	$\tau_d^{i,j}(0)$	15
ニューロン i, k 間の初期時間遅れ $[\Delta t]$	$\tau_d^{i,k}(0)$	20
ニューロン i, l 間の初期時間遅れ $[\Delta t]$	$\tau_d^{i,l}(0)$	10
ニューロン i, j 間の結合荷重	$w_0^{i,j}$	0.8
ニューロン i, k 間の結合荷重	$w_0^{i,k}$	0.7
ニューロン i, l 間の結合荷重	$w_0^{i,l}$	0.5
学習率	$\alpha^{i,j}, \alpha^{i,k}, \alpha^{i,l}$	0.8

図 4 τ_d の時間特性

Fig. 4. Time characteristics of τ_d .

図 5 提案手法における w_0 の時間特性

Fig. 5. Time characteristics of w_0 in the proposed method.

$$\tau_d^{i,j}(t + \Delta t) = \tau_d^{i,j}(t) - \alpha^{i,j} \left\{ 1.0 - \Gamma \left(w_0^{i,j}(t) \right) \right\} \cdot y^i(t) W_p(t_0^i - t_d^{i,j}) \dots (12)$$

ここで、時間遅れ $\tau_d^{i,j}$ の値域を本論文では、 $\Delta t \leq \tau_d^{i,j}(t) \leq \tau_r^i$ とする。

式 (12) により、後ニューロンの発火に関係している前ニューロンからのパルスの時間遅れ $\tau_d^{i,j}$ を小さくし、後ニューロンの発火タイミングを早くすることができる。すると、前ニューロンからのパルスの内、後ニューロンの発火タイミングより遅く到達するパルスが現れ、後ニューロンの発火に関係しているシナプス数を減少させることが可能となる。したがって、後ニューロンの発火に不必要なシナプスの結合荷重を増加させずに、記憶容量の低下を防ぐことが可能となると考えられる。

〈3・3〉 予備実験 従来法である Eurich らのヘブ型学習則と、破局的忘却の抑制を考慮したヘブ型学習則を比

図 6 y および P_m^i の時間特性

Fig. 6. Time characteristics of y and P_m^i .

較し、提案手法が破局的な忘却を抑制することができるかについて、予備実験を行う。その際、提案手法は、〈3・2〉で述べた記憶容量の低下を防ぐためのヘブ型学習則も併用する。

ニューロンおよびシナプスのパラメータは表 1 のように設定する。入力層のニューロンが 3 個 (ニューロン j, k, l)、出力層のニューロンが 1 個 (ニューロン i) である 2 層の PNN とする。入力層のニューロンはすべて周期 T で発火するが、ニューロン k の発火周期に対し、ニューロン j は $-10\Delta t$ 、ニューロン l は $+20\Delta t$ 位相がずれて発火するものとする。また、 t が 0 から $1000\Delta t$ の間はニューロン j は発火せず、ニューロン k, l だけが発火する。 t が $1000\Delta t$ から $2000\Delta t$ の間は入力層のニューロン j, k, l すべて発火する。 t が $2000\Delta t$ から $3000\Delta t$ の間は、再びニューロン j は発火せず、ニューロン k, l だけが発火する。つまり、ニューロン k, l だけが発火するパターン 1 とニューロン j, k, l が発火するパターン 2 の 2 種類のパターンを学習する。

〈3・4〉 予備実験結果 従来法および提案手法の結果を図 4~6 に示す。

図4より、従来法では、パターン1からパターン2へ変化すると ($t = 1000\Delta t$), $\tau_d^{i,k}, \tau_d^{i,l}$ ともに変化していることがわかる。そのため、再び、 $t = 2000\Delta t$ 以降でパターン2からパターン1へ変化するとニューロン i が発火しなくなる (図6 [c])。これより、従来法では破局的な忘却が生じていることが確認できる。

これに対し、提案手法では、図4, 5より、パターン1からパターン2へ変化しても時間遅れ $\tau_d^{i,k}$ および結合荷重 $w_0^{i,k}$ は変化しない。再び、 $t = 2000\Delta t$ 以降でパターン2からパターン1へ変化しても、ニューロン i は発火する。よって、提案手法は破局的な忘却を抑制することができている。

この予備実験により、従来法は破局的な忘却が生じてしまうことが確認でき、提案手法は破局的な忘却を抑制することが明らかとなった。

4. PNNの自律移動ロボットのコントローラへの応用

PNNの応用として、PNNをコントローラに用いた、自律移動ロボットの走行実験を行う。

具体的には、自律移動ロボットに異なる2種類の環境中を走行させ、それぞれのスタート地点からゴール地点まで移動させることをロボットのタスクとする計算機実験である。そして、出力にシグモイド関数を用いた階層型3層ニューラルネットワーク (以下、Analog NN) をコントローラとするロボット、および時間遅れを調節するヘブ型学習則を用いたPNN (以下、PNN 0) をコントローラとするロボットの行動が、提案手法のヘブ型学習則を用いたPNNをコントローラとする場合と比較して、どのような違いが現れるかを計算機シミュレーションを通して確認する。

また、提案手法をコントローラとして用いる場合、〈3・2〉で述べた記憶容量の低下を防ぐためのヘブ型学習則を併用しないPNN (以下、PNN 1) をコントローラとする場合と、記憶容量の低下を防ぐためのヘブ型学習則を併用するPNN (以下、PNN 2) をコントローラとする場合について実験を行う。なお、Analog NNの出力に局所的な反応特性をもつRadial基底関数を用いるのが一般的だが、Radial基底関数を用いたネットワークを使用する場合、本論文の学習は教師なし学習であること、後述する遺伝的アルゴリズムでのネットワークのパラメータ調整が難しいこと、同じく後述する移動モジュールを選択する方式との相性が悪い等の問題が生じてしまう。そのため、Radial基底関数を使用せず、大局的な反応特性をもつシグモイド関数を用いることにする。また、Analog NNにおける学習は、一般的に教師なしのヘブ型学習則を用いる。

〈4・1〉 自律移動ロボットシミュレータ 実験環境として、独自に開発した自律移動ロボットシミュレータを使用する⁽¹⁶⁾。このシミュレータは、図7に示すようなロボットを想定しており、以下のような特徴がある。1. ロボットは円形で、大きさを任意に設定可能。2. 左右の車輪を駆動す

図7 自律移動ロボット

Fig. 7. An autonomous mobile robot.

表2 自律移動ロボットシミュレータのパラメータ

Table 2. Parameters for a simulator of autonomous mobile robot.

環境の大きさ	$E_x \times E_y$	200×200
ロボットの半径	R_b	10
近接センサ反応半径	R_p	15
視覚センサ解像度 [bit]	-	15
視覚センサ反応角 [deg.]	θ_v	45
視覚センサ反応半径	R_v	50

るモータへの出力を制御することで、ロボットの移動が可能。3. ロボットの行動フィールド (マップの大きさ、障害物等) を任意に設定可能。4. 近接センサを搭載し、反応半径を任意に設定可能。5. 視覚センサを搭載し、反応半径、解像度を任意に設定可能。

本実験では表2のようにパラメータを設定する。

〈4・2〉 環境およびタスク設定 図8に示すmapA, mapBの2種類の環境でロボットにタスクを実行させる。mapA, mapBの左側の黒色の円がロボットで、ロボットのいる地点がスタート地点となる。また、mapAの右上側にある円、mapBの中央下の円がゴール地点であり、mapAの中央の円はライトである。

以上のような、2種類の異なる環境mapA, mapBにおいて、それぞれの環境内のスタート地点からゴール地点まで、mapA, mapB, mapA, mapBと連続して移動することをロボットのタスクとする。ロボットは規定ステップ内に環境内を自由に動き回り、タスクの達成度合いに応じて報酬を得る。

このタスクは、mapA, mapBを繰り返し移動するため、破局的な忘却が生じやすい。また、mapAとmapBはライトの有無とゴール以外、全く同じ環境である。そのため、コントローラである各ニューラルネットワークにとって、この2種類の環境は区別することは難しいと考えられる。

〈4・3〉 適応度 ロボットはスタート地点からゴール地点に移動するまでに、図9に示すサブゴール (点線で囲まれた部分) 内に移動することで、同図中の各サブゴール

図 8 実験環境

Fig. 8. Experimental environments.

図 9 サブゴールおよび報酬

Fig. 9. Sub goals, and its rewards in the environments.

に割り当てられた値だけ報酬を獲得する。また、ゴール地点に到着すると規定ステップ数から、ゴール到着に要したステップ数を引いた値を報酬として獲得する。また、後述する移動モジュールの選択の際、前進モジュールを選択した場合に報酬 1 を獲得する。

タスクを終了した時点で、獲得している報酬の合計をロボットの適応度とする。すなわち、環境内を動き回り、より早くゴールに到達するロボットは適応度が高くなる。

〈4・4〉ニューラルネットワークの設定とネットワーク構造 ネットワークの出力はニューロン数を 3 個とし、後述する 3 個の移動モジュールを選択する際に用いる。また、ネットワークへの入力、ロボットの障害物センサの 0~1 の実数値（距離に応じて変化）および視覚センサの 0~1 の実数値（ゴール：0.95、ライト：0.5、壁：0.35）、壁に衝突したか、ライトに接触したか、ゴールに到達したかのロボット内の 3 種類のフラグ（0, 1 の 2 値）、および 1 ステップ前のネットワークの出力とする。

PNN のネットワーク構造を図 10 に示す。本論文では、一つのニューロンの軸索が次層のニューロンすべてにシナプスを形成するのではなく、ネットワーク全体で形成できる総シナプス数を設定する。つまり、ニューロンの軸索が形成しているシナプス数はそれぞれのニューロンで異なる。総シナプス数を設定することでネットワークをハードウェア化する場合、次層すべてのニューロンとシナプスを形成する場合に比べ、メモリや配線数の消費量を軽減することが可能となる。

図 10 ネットワーク構造とニューロン h の軸索範囲 R_a Fig. 10. Network structure, and axon region R_a of neuron h .

表 3 移動モジュールのパラメータ

Table 3. Parameters for movement modules.

前進モジュール移動速度 $[R_h/\Delta t]$	0.5
旋回モジュール旋回角度 $[\text{deg.}]$	9

ネットワークの初期設定において、個々のシナプスのシナプス前ニューロンをランダムに決定し、以下のようなシナプス形成ルールによりシナプス後ニューロンを決定する。

- 入力層のニューロンの軸索は、隠れ層の第 1 層だけにシナプスを形成する。
- 隠れ層の最終層のニューロンの軸索は、出力層だけにシナプスを形成する。
- 隠れ層のニューロン h の軸索は、図 10 に示す軸索範囲 $R_a = (r_x, r_y)$ 内の隠れ層のニューロンにシナプスを形成する。図 10 に示すように隠れ層のニューロンには座標を設定している。隠れ層の層数を l_{ym} (l_y 方向)、一層辺りのニューロン数を l_{xm} (l_x 方向) とすると、隠れ層のニューロン $h(h_x, h_y)$ の軸索がシナプス形成可能な座標範囲 $R_h(h_x, h_y) = (r_{hx}, r_{hy})$ は、

$$h_x - r_x/2 \leq r_{hx} \leq h_x + r_x/2, \\ h_y + 1 \leq r_{hy} \leq h_y + r_y$$

となる。

- 隠れ層のニューロン h の座標が、 $h_x < r_x/2$ および、 $h_x > l_{xm} - r_x/2$ の場合、 l_x 方向のシナプス形成範囲をそれぞれ、 $1 \leq r_{hx} \leq h_x + r_x/2$ および、 $h_x - r_x/2 \leq r_{hx} \leq l_{xm}$ とする。
- 隠れ層のニューロン h の座標が、 h_y が $l_{ym} - r_y$ から $l_{ym} - 1$ の場合、 l_y 方向のシナプス形成範囲を $h_y + 1 \leq r_{hy} \leq l_{ym}$ とする。

ニューロン i が入力層のニューロンである場合、入力層のニューロンの出力 $y^i(t)$ は、式 (4) の代わりに式 (13) を、式 (5) の代わりに式 (14) を用いる。ここで、 $x_i(t)$ は時刻 t における入力層のニューロン i への入力（0~1 の実数値）である。また、式 (3) における τ_{γ}^i は、 Δt とする。

$$y^i(t) = \mathcal{H}(t - t_\gamma^i) \left\{ \frac{2}{1 + \exp(-\beta x_i^i(t))} - 1 \right\} \dots\dots\dots (13)$$

$$t_0^i = \begin{cases} t & \text{for } \mathcal{H}(x_i(t) - \theta^i) = 1 \\ \text{no} & \text{for } \mathcal{H}(x_i(t) - \theta^i) = 0 \\ \text{changed} & \end{cases} \dots\dots\dots (14)$$

〈4・5〉 移動モジュール ロボットは、左右の車輪を駆動するモータへの出力の値を、直接制御することで移動するのではなく、あらかじめ設計した前進、右旋回、左旋回の3種のモジュールを選択することにより移動する。前進モジュールは、車輪を駆動する左右のモータへ等しい値を出力するモジュールである。また、右旋回、左旋回モジュールは、左右のモータへ絶対値の等しい正負が逆の値を出力し、その場で旋回するモジュールである。本実験では、前進モジュールの移動速度および旋回モジュール旋回角度を表3のように設定する。

モジュールの選択方法は、式(15)で計算される $M_s^{mod}(t)$ の値が最も大きいモジュールを選択する。

$$M_s^{mod}(t) = \frac{\tau_m^o}{\tau_m^o + \Delta t} M_s^{mod}(t - \Delta t) + \frac{\Delta t}{\tau_m^o + \Delta t} y^o(t) \dots\dots\dots (15)$$

ここで、 $y^o(t)$ はネットワークの出力層の時刻 t での各ニューロンの出力、 τ_m^o は出力層の各ニューロンの減衰時定数、 mod は前進、右旋回、左旋回モジュールのいずれかである。モジュール1個につき、出力層のニューロン1個が対応する。

〈4・6〉 遺伝的アルゴリズムによるネットワークの調整 自律移動ロボットのコントローラとしてニューラルネットワークを使用する場合、ニューロンおよびシナプスの個数が多くなる。そのため、トップダウン的にニューロンおよびシナプスの各パラメータを適切に設定することは困難となる。また、本論文で提案する学習則は、「教師なし」学習であり、個々のニューロンの相互作用にしたがって学習を行う。そのため、ある時系列パターンがネットワークへ入力された場合、出力の時系列パターンは個々のニューロンの相互作用に依存する。そこで、入力に対する望ましい出力が得られるように遺伝的アルゴリズム（以下、GA）等の最適化手法を用いて、ネットワークの構造やパラメータを調整する。その結果、入力に対する望ましい出力が得られ、提案する学習則を3層以上のネットワークに適用することが可能となる。また、ニューラルネットワークをハードウェア化する場合、ハードウェア化が容易なPNNとGAを組み合わせ、FPGA等の再構成可能なデバイスに搭載することで、進化型ハードウェアを構築することが可能になると考えらる。

ニューロンのパラメータ (τ_m^i, τ_γ^i)、およびシナプスのパラメータ ($i, j, \tau_d^{i,j}, w_0^{i,j}$) を実数値として並べた二本の一次元配列を染色体とする。染色体のニューロンに関する部分とシナプスに関する部分は、別々に遺伝操作を施す。シナプスパラメータ ($i, j, \tau_d^{i,j}, w_0^{i,j}$) のうち、 $\tau_d^{i,j}, w_0^{i,j}$ は、学

表4 PNNのパラメータ

Table 4. Parameters for PNN.

ネットワークの時間の刻み幅	Δt	0.1
入力層ニューロン数 [個]	N_i	52
隠れ層ニューロン数 [個]	N_h	120
出力層ニューロン数 [個]	N_o	3
隠れ層の層数 [層]	L_h	3
隠れ層の軸索範囲	$R_a = (r_x, r_y)$	(30, 3)
総シナプス数 [個]	N_s	500
閾値	θ^i	0.1
学習率	$\alpha^{i,j}$	0.2
減衰時定数 [Δt]	τ_m^i	2~12
不応期 [Δt]	τ_γ^i	8~10
初期時間遅れ [Δt]	$\tau_d^{i,j}(0)$	1~7
初期結合荷重	$w_0^{i,j}(0)$	0.1~1.0

表5 GAのパラメータ

Table 5. Parameters for GA.

総個体数	N	40
保存個体数	E_n	5
交叉確率	C_r	0.80
突然変異確率	C_m	0.002
規定世代数	G_{max}	500

習則で修正されるが、これらの初期値をGAにより調整する。また、遺伝的操作により親個体から次世代の子個体へ受け継がれる $\tau_d^{i,j}, w_0^{i,j}$ は、親個体であるロボットが各環境を走行する際に学習則で修正される前の初期値であり、学習後の $\tau_d^{i,j}, w_0^{i,j}$ は親個体から子個体には反映されない。

遺伝的操作には、二点交叉と突然変異を用いる。世代交代は、集団 N 個体中、適応度の高い順に E_n 個体を保存し、そのまま次世代に残す。ここで、保存の対象となる個体と適応度が等しい個体が複数いる場合、遺伝子に多様性を持たせるため、ランダムに選択した一個体だけ保存し、次に適応度が高い個体を順次保存する。また、前世代で保存した個体が保存の対象となる場合で、前世代で保存した個体と適応度が等しい個体が複数存在するとき、進化の停滞を防ぐため、前世代で保存した個体は保存の対象から除く。残りの個体を交叉確率 C_r で交叉を行う。交叉を行う際、適応度に基づきルーレット選択により親個体を選択する。また、染色体を複製する際に、ニューロンおよびシナプス一個につき、突然変異確率 C_m で突然変異を施し、新たなニューロンおよびシナプスを生成する。

〈4・7〉 実験手順 実験の手順を以下に示す。

- (1) 初期集団を N 個体生成する。
- (2) すべてのロボットについて以下の処理を実行する。
 - (a) mapAにおいて、ロボットを規定ステップ数走行させる。壁に衝突した場合は、手順(3)に進む。
 - (b) mapBにおいて、(a)と同様の処理を行う。
 - (c) 再びmapAにおいて、同様の処理を行う。
 - (d) 再びmapBにおいて、同様の処理を行う。
- (3) 適応度に基づき遺伝的操作を施し、新集団を生成

図 11 各世代に対する最良個体の適応度の変化 (20 試行の平均)

Fig. 11. Comparison of generation change on averaged fitness of each best individual in each method (20 trials).

図 12 各世代に対するゴール到達回数の変化 (20 試行の平均)

Fig. 12. Comparison of generation change on averaged count to accomplish goal in each method (20 trials).

表 6 タスク成功率

Table 6. Task success rates.

Analog NN	0.05
PNN 0	0.40
PNN 1	0.70
PNN 2	0.80

する。

(4) 規定世代数 $G_{\max}$ まで手順 (2), (3) を繰り返す。ここでいう、ステップとは時間の刻み幅 Δt に等しい。規定ステップ数を 200 ステップとする。

〈4・8〉パラメータ設定 実験における PNN および GA のパラメータを、それぞれ表 4, 5 のように設定する。また、各パラメータは経験的に設定する。

〈4・9〉実験結果 図 11 に各世代に対する最良ロボットの適応度の変化 (20 試行の平均) を示す。横軸は世代、縦軸はロボットの適応度の値を表す。

また、図 12 に各世代に対する最良ロボットのゴール到達回数の変化 (20 試行の平均) を示す。横軸は世代、縦軸はゴール到達回数を表す。すべての環境において、ゴールへ到達すると 4 になる。また、表 6 に 20 試行中、タスクを成功した試行の割合を示す。

Analog NN, PNN 0 および PNN 2 をコントローラとする場合のロボットの行動軌跡について、代表的な行動パ

mapA-手順 (2) (a) mapB-手順 (2) (b)

図 13 Analog NN をコントローラとする場合の行動軌跡 (適応度 467)

Fig. 13. Trace of behavior of the robot using Analog NN in controller (Fitness 467).

mapA-手順 (2) (a) mapB-手順 (2) (b)

mapA-手順 (2) (c) mapB-手順 (2) (d)

図 14 PNN 0 をコントローラとする場合の行動軌跡 (適応度 1106)

Fig. 14. Trace of behavior of the robot using PNN 0 in controller (Fitness 1106).

ターンを図 13~15 に示す。

〈4・10〉考 察 図 11, 表 6 より, Analog NN に比べ, PNN をコントローラとするロボットの性能が良いことがわかる。ネットワークへ入力される時系列パターンに対する Analog NN の時間分解能よりも, PNN の時間分解能のほうが高いため, PNN を用いるコントローラは, より適切なタイミングで移動モジュールの選択が可能になる。ゆえに, Analog NN および PNN をコントローラとするロボットの性能差が現れると考えられる。

また, PNN 1 および PNN 2 をコントローラとする場合の適応度とタスク達成率が, PNN 0 をコントローラとする場合より高いことがわかる。そして, 提案手法を用いる中でも, PNN 2 をコントローラとする場合がもっとも適応度とタスク達成率が高い。これより, PNN 2 における, 記憶容量の低下を防ぐためのヘブ型学習則が有効に働いていることが確認できる。

図 13 より, Analog NN をコントローラとする場合,

図 15 PNN 2 をコントローラとする
場合の行動軌跡 (適応度 1167)

Fig.15. Trace of behavior of the robot using
PNN 2 in controller (Fitness 1167).

mapA ではゴールに到達できるものの、mapB では壁に衝突してしまい、タスクを失敗していることがわかる。また、図 14 より、PNN 0 をコントローラとする場合、手順 (2) (a), (c) の 2 回の mapA の走行はいずれもゴールに到達しているが、手順 (2) (d) で、再び mapB を走行すると壁に衝突し、タスクを失敗してしまう。これはヘブ型学習則により時間遅れを調節することで、ネットワークの内部状態が変わってしまう、つまり破局的な忘却が生じてしまうことが原因であると考えられる。

これらに対し、図 15 の PNN 2 をコントローラとする場合は、mapA, mapB ともに手順 (2) (c), (d) で再び走行しても軌跡に違いがさほど見られず、タスクを成功している。これより、破局的な忘却の抑制を考慮したヘブ型学習則が有効に働いていることが確認できる。

5. むすび

本論文では、減衰のある積分発火ニューロンを用いる PNN において、破局的な忘却の抑制を考慮したヘブ型学習則を提案した。予備実験の結果、本提案手法により、破局的な忘却の抑制が可能であることが明らかとなった。

また、PNN の応用として、PNN を自律移動ロボットのコントローラに用いる自律移動ロボットの走行実験を行った結果、出力にシグモイド関数を用いた階層型 3 層ニューラルネットワークをコントローラに使用するロボットに比べ、PNN をコントローラに使用するロボットの性能が良いことが明らかとなった。さらに、PNN において、従来法のヘブ型学習則を用いる PNN に比べ、提案手法のヘブ型学習則を用いる PNN をコントローラとするロボットの性能がより良いことが明らかとなった。

今後の課題として、本提案手法の雑音、および故障等に対する頑健性についての考察が必要である。

(平成 14 年 5 月 29 日受付, 平成 14 年 11 月 14 日再受付)

文 献

- (1) W. Gerstner, R. Kempter, J.L. van Hemmen, and H. Wagner: "A neuronal learning rule for sub-millisecond temporal coding", *Nature*, Vol.383, pp.76-78 (1996-9)
- (2) C. W. Eurich, J. D. Cowan, and J. G. Milton: "Hebbian Delay Adaptation in a Network of Integrate-and-Fire Neurons", *Artificial neural networks - ICANN'97*, pp.157-162, Springer, Berlin (1997-8)
- (3) D. Keyakidani, K. Amemori, and S. Ishii: "Spike-based Hebbian Delay Adaptation", *Technical report of IEICE*, NLP99-172, pp.83-90 (2000-3) (in Japanese)
櫻谷 大・雨森賢一・石井 信: 「スパイク系列に依存した時間遅れのヘブ型学習」, 電子情報通信学会技術研究報告, NLP99-172, pp.83-90 (2000-3)
- (4) K. Amemori and S. Ishii: "Self-organization and association for spatio-temporal spike sequences", *ISCIE Transactions*, Vol. 13, No. 7, pp.308-317 (2000-7) (in Japanese)
雨森賢一・石井 信: 「精緻な時空間スパイク列の自己組織化と想起」, システム制御情報学会論文誌, 13, 7, pp.308-317 (2000-7)
- (5) N. Fatamata, S. Kuroyanagi, and A. Iwata: "An Implementation Method of Pulsed Neuron Model for FPGA Devices", *Technical report of IEICE*, NC2001-211, pp.121-128 (2002-3) (in Japanese)
二俣宣義・黒柳 奨・岩田 彰: 「FPGA のためのパルスニューロンモデルの実装手法」, 電子情報通信学会技術研究報告, NC2001-211, pp.121-128 (2002-3)
- (6) M. McCloskey and N. J. Cohen: "Catastrophic interference in connectionist networks: The sequential learning problem", In G. H. Bower, ed. *The psychology of learning and motivation*, Vol. 24, pp.109-165, New York: Academic Press (1989)
- (7) R. Ratcliff: "Connectionist Models of Recognition Memory: Constraints Imposed by Learning and Forgetting Functions", *Psychological Review*, Vol. 97, pp.285-308 (1990)
- (8) R. M. French: "Using Semi-Distributed Representations to Overcome Catastrophic Forgetting in Connectionist Networks", *Proc. of the 13rd Annual Cognitive Science Society Conf.*, pp.173-178, NJ: LEA (1991)
- (9) R. M. French: "Dynamically constraining connectionist networks to produce distributed, orthogonal representations to reduce catastrophic interference", *Proc. of the 16th Annual Cognitive Science Soc. Conf.*, NJ: LEA, pp.335-340 (1994)
- (10) R. M. French: "Using pseudo-recurrent connectionist networks to solve the problem of sequential learning", *Proc. of the 19th Annual Cognitive Science Soc. Conf.*, NJ: LEA, pp.921-926 (1997)
- (11) R. M. French: "Catastrophic Forgetting in Connectionist Networks", *Trends in Cognitive Sciences*, Vol. 3, No. 4, pp.173-178 (1999-4)
- (12) J. P. Sougné and R. M. French: "Synfire chains and catastrophic interference", *Proc. of the 23rd Annual Cognitive Science Soc. Conf.*, NJ: LEA, pp.270-275 (2001)
- (13) S. Kondo, R. Futami, and N. Hoshimiya: "Pattern recognition with feature map and the improvement on sequential learning ability", *Tec. report of IEICE*, NC95-161, pp.55-60 (1996-3) (in Japanese)
近藤修平・二見亮弘・星宮 望: 「特徴マップを用いたパターン認識における追加学習能力の向上」, 電子情報通信学会技術研究報告, NC95-161, pp.55-60 (1996-3)
- (14) 久野 宗, 監修: 脳を知る, 秀潤社 (1999)
- (15) 藤 義博・高畑雅一: 脳と行動の生物学, 講談社 (2000)
- (16) M. Motoki, T. Hamagami, S. Koakutsu, and H. Hirata: "Acquisition of Recognition Ability of Autonomous Mobile Robots Using Pulse Neural Network Considering Working Memory", 2001 Proceeding of Electronics, Information and Systems Conference Electronics and Systems Society, IEE Japan, Vol. I, pp.519-522 (2001-9) (in Japanese)

元木 誠・濱上知樹・小坏成一・平田廣則：「ワーキングメモリを考慮したパルスニューラルネットワークによる自律移動ロボットの識別能力の獲得」，電気学会電子・情報・システム部門大会講演論文集 I, pp.519-522 (2001-9)

元 木 誠 (学生員) 1977 年 8 月 10 日生。2002 年，千葉

大学大学院自然科学研究科電子機械科学専攻博士前期課程修了。現在，同大学大学院自然科学研究科人工システム科学専攻博士後期課程に在学。パルスニューラルネットワークの研究，および，その応用に関する研究に従事。

濱 上 知 樹 (正員) 1966 年 3 月 7 日生。88 年，千葉大学工

学部電気工学科卒業。セコム(株)勤務を経て，99 年，在職のまま千葉大学大学院自然科学研究科生産科学専攻博士後期課程修了。2001 年，同大学院自然科学研究科助手となり現在に至る。大規模システムの数理，マルチエージェント，機械学習，創発システムの応用研究に従事。博士(工学)。IEEE，電子情報通信学会。

小 坏 成 一 (正員) 1964 年 7 月 7 日生。92 年，千葉大学大学院自然科学研究科生産科学専攻博士課程修了。同年，千葉大学工学部助手，97 年同助教授，同年同大学院自然科学研究科助教授となり現在に至る。94~95 年，カリフォルニア大学サンタクルズ校客員研究員。VLSI レイアウト，確率的最適化手法，ニューラルネットワークの研究に従事。博士(工学)。IEEE，INNS，電子情報通信学会，計測自動制御学会会員。

平 田 廣 則 (正員) 1948 年 6 月 2 日生。76 年，東京工業大学大学院電気工学専攻博士課程修了。同年，千葉大学工学部助手，81 年同助教授，94 年同教授，97 年同大学院自然科学研究科教授となり現在に至る。大規模システムのモデリング，解析と設計，特に生態システム，VLSI レイアウト，並びに分散システムの基礎的理論に興味を持つ。工学博士。2001 年電気学会業績賞受賞。IEEE(fellow)，ISEM，INNS，電子情報通信学会，情報処理学会，計測自動制御学会，システム制御情報学会，人工知能学会会員。