

SpikeProp において余分なスパイクを抑制する学習法
 —ネットワーク構造の自動調整による構造依存性の軽減—
 A Learning Method for SpikeProp Suppressing Redundant Spikes
 —Reduce Dependency of Learning Performance for Network Structure—

松本 崇* 高瀬 治彦*
 Takashi Matsumoto Haruhiko Takase

川中 普晴* 鶴岡 信治*
 Hiroharu Kawanaka Shinji Tsuruoka

1 はじめに

近年, ユニット間のやりとりをスパイク(パルス)で行うスパイクニューラルネットワーク(SNN)を用いた時系列パターン処理が注目を集めている. 生体のニューロンはスパイクにより電気信号をやり取りしていることが知られており, SNN はニューラルネットワークの中では生体に近いモデルである. SNN は情報を表現する手法により大きく二種に分けることができる. 一つはスパイクの密度を用いるモデルである. もう一つはスパイクのタイミングによるモデルである. 前者はスパイク密度を観測するために時間がかかるため, 応答の速さの観点において, 後者のほうが好ましいとされている [1].

スパイクのタイミングにより情報表現を行う SNN の学習法としてさまざまな手法が提案されている [2]. その一つに Booiij らによって提案された SpikeProp がある [3]. これは, Bohte らにより提案された SpikeProp [4] を拡張したモデルである. Bohte らのモデルでは, ネットワークの構成ユニットが高々 1 回しか発火しないのに対し, Booiij らのモデルではこれらが複数回発火することを許容している. このモデルは, ネットワークが出力する複数の出力スパイクの発火時刻を, 誤差逆伝播法に基づく手法により学習できる. その結果, 時系列信号から時系列信号への任意の変換を学習でき, 時系列情報処理システムへの応用が期待されている. しかし, Booiij らの提案した学習法は, 先頭のいくつか(教師パターンで指定した分)のスパイクの発火時刻を調整するのみである. そのためスパイク数を調節できず, 必要なパターンを出力した後に過剰なスパイクが出力されるのを防ぐことはできない. これは SpikeProp の機能をスパイク列の変換であるとみなしたときに好ましくない動作である.

本研究の最終目的は, Booiij らのモデルにおいて, 出力スパイクの発火時刻だけでなく, それらの数も学習できるようにすることである. これまで我々のグループでは, 余分な発火を抑制する方法を提案してきた [5, 6, 7]. これらは, 余分な結合荷重を全体的に弱める手法 [5], 学習性能が初期ネットワーク構造に依存していることに着目し, 自動的にユニットを削除する手法 [6], あるいは時間遅れを削除する手法 [7] である. 本稿では, この成果をう

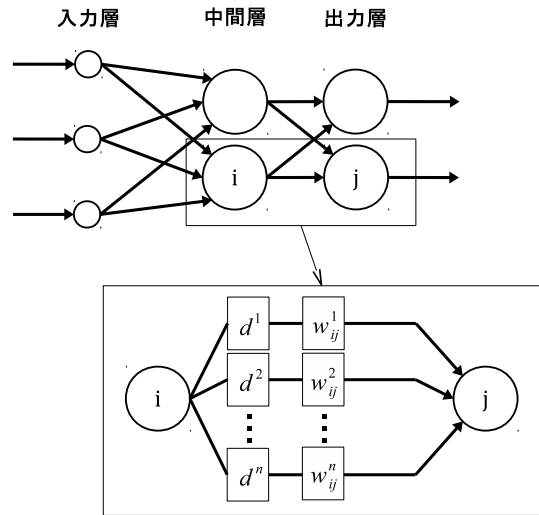


図 1 典型的な SpikeProp のネットワーク構造

けて, これらの手法を組み合わせることでその効果を検証し, 適切なネットワーク構造を自動で獲得することをめざす.

2 従来研究

この章では Booiij らによって提案された SpikeProp モデルの概略と筆者らのグループで行ってきた従来研究について簡単に述べる.

2.1 SpikeProp

SpikeProp は入出力関係にスパイク(パルス)を用いるモデルである. ネットワーク構造は, フィードバック結合を持たない形で定義される. 典型的なネットワーク構造は, 図 1 に示すような階層型の構造を持つ. ユニット間の結合は 1 本ではなく多数の副結合からなっており, それぞれが結合荷重と時間遅れという二種のパラメータを持つ. 時間遅れの値は副結合番号 k により決まる定数として d^1, d^2, \dots, d^m のように表す. ユニット i からユニット j への k 番目の副結合の結合荷重は, w_{ij}^k のように表す. 各ユニットの内部状態は, 式 (1) に従い変化する. ここで, 式 (2) はスパイクに対する応答を示すスパイク応答関数を, 式 (3) は自身の発火による不応期を実現するための刺激を表す不応期関数である. 内部状態が, あらかじめ決められたしきい値 θ を超えた時, ユニットは発火しスパイクを出力する.

* 三重大学大学院 工学研究科

$$x_i(t) = \sum_{t_j^f \in F_i} \kappa(t - t_j^f) + \sum_{j \in \Gamma_i} \sum_{t_j^f \in F_j} \sum_k w_{ij}^k \epsilon(t - t_j^f - d^k) \quad (1)$$

$$\epsilon(t) = \left(\exp\left(\frac{t}{\tau_m}\right) - \exp\left(\frac{t}{\tau_s}\right) \right) H(t) \quad (2)$$

$$\kappa(t) = -\theta \exp\left(-\frac{t}{\tau_r}\right) H(t) \quad (3)$$

ここで、 $H(t)$ はヘビサイドのステップ関数であり、 $t \leq 0$ で 0、それ以外で 1 となる。時定数 τ_m, τ_s, τ_r は、スパイク応答関数および不応期関数の時定数である。 F_i はユニット i が出力したスパイクの発火時刻の集合であり、 f 番目のスパイクの時刻を t_j^f と表す。 Γ_i はユニット i に接続しているユニットのユニット番号の集合である。また、一部のユニットは抑制性ユニットとして振る舞い、そのユニットから出力されたスパイクに対するスパイク応答関数の値は -1 倍されるものとする。

出力層ユニットの発火時刻を調整するために式 (4) のように誤差を定め、これを最小化する。

$$E = \frac{1}{2} \sum_{p \in P} \sum_{j \in J} \sum_l \left(t_j^{p,l} - \hat{t}_j^{p,l} \right)^2 \quad (4)$$

ここで、 J は出力層ユニットの集合を、 P は教師パターンの集合を表す。また、 $t_j^{p,l}, \hat{t}_j^{p,l}$ はパターン p におけるユニット j の l 番目のスパイクの出力時刻および希望する出力時刻を表す。学習は誤差逆伝搬法に基づいて行う。結合荷重 w_{ij}^k の更新量 Δw_{ij}^k は式 (5) により求められる。

$$\Delta w_{ij}^k = -\eta \frac{\partial E}{\partial w_{ij}^k} \quad (5)$$

ここで、 η は学習率であり正の実数である。なお、学習を繰り返すと、ユニットの発火回数が不足することがある。学習中に一つでもそのようなユニットが生じた場合には、そのユニットへのすべての結合の結合荷重を大きくする。

2.2 不要な発火を抑制する従来手法

この節では、不要な発火を抑制する従来手法について簡単に説明する。

2.2.1 不要な発火

Booij らのモデルでは、式 (4) を学習により最小化するため、ネットワークが出力する最初の何個かのスパイクについてのみ、それらの発火時刻を学習する。そのため、それ以降の挙動については学習できない。その結果、期待した以上に出力されるスパイク (余分なスパイク) を Booij らの学習法ではなくすることはできない。実際、Booij らの論文にある時間版 XOR 問題の学習を行ったところ、教師パターンとして一つの出力スパイクを与えたのに対して

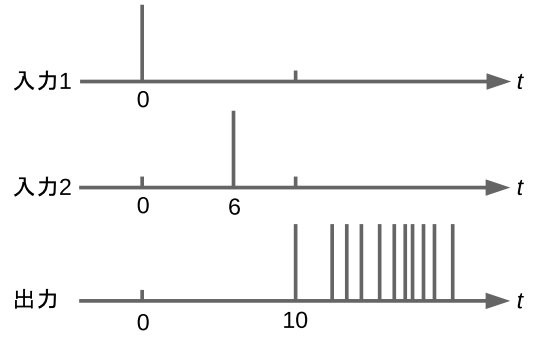


図 2 余分なスパイクを出力する学習後の SpikeProp ネットワーク

数十個のスパイクを出力するネットワークを得た (図 2)。入力 1 を時刻 0 に入力 2 を時刻 6 に入力に対して、期待する出力は時刻 10 のみであり、出力の最初のスパイク以外は、余分なスパイクである。

2.2.2 調整 WD

それに対し、我々のグループでは余分なスパイクはネットワークの余分な構成要素によりもたらされると考え、これを削除するために調整荷重減衰 (WD: Weight Decay) 法 [5] および、ユニットの自動削除 [6]、不要な時間遅れの削除法 [7] を提案してきた。

調整 WD 法では不要な結合荷重が 0 になるように式 (6) により定義された評価関数 E' を、誤差 E の代わりに最小化する。第一項は SpikeProp の誤差 E であり、第二項は荷重減衰によるペナルティ項である。 E' を最小化することで、誤差を減少させることに寄与しない結合荷重は弱められる。

$$E' = E + \frac{\rho}{\eta} \sum_{w \in W} w^2 \quad (6)$$

ここで、 W はネットワーク内の全結合荷重の集合、 ρ は荷重減衰の影響の強さを調整する定数である。

このとき、式 (5) の替わりに、式 (7) に従い結合荷重を調整する。

$$\Delta w_{ij}^k = -\eta \frac{\partial E}{\partial w_{ij}^k} - 2\rho w_{ij}^k \quad (7)$$

η と ρ を学習の進行状況に合わせて変化させることで、効果的に余分な発火を減らすことができた。

2.2.3 中間層ユニット数の自動調整

文献 [6] では、学習中にネットワークの冗長な中間層ユニットを削除することで、調整 WD 法の学習性能が初期中間層ユニット数に依存してしまうことを解消した。この手法は、ある程度学習した時点で、発火していない中間層ユニットは冗長なユニットであると考え、それを削除する。その手順は以下のとおりである。

1. ネットワークを動作させ、その出力を得る。

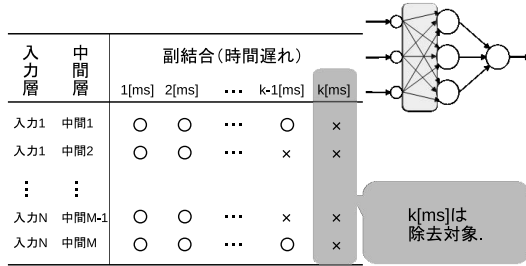


図 3 不要な時間遅れの削除 (入力層-中間層間)

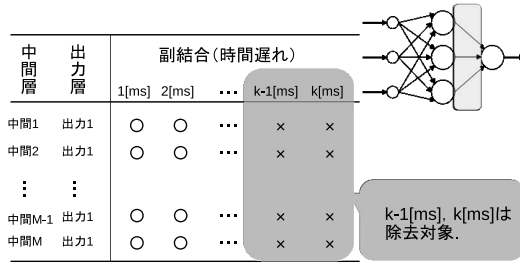


図 4 不要な時間遅れの削除 (中間層-出力層間)

2. 誤差が十分に減少し、かつ、余分なスパイクを出力していないならば学習終了。
3. 誤差 E がしきい値 E_d 以下となった場合は手順 4 へ、そうでない場合は手順 5 へ。
4. 発火していない中間層ユニットを一つ削除し、手順 1 へ。
5. 調整 WD 法により結合荷重を更新し、手順 1 へ。

2.2.4 時間遅れの自動調整

文献 [7] では、学習中に不要な時間遅れをもつ結合を削除することで、調整 WD 法の学習性能が学習初期に用意した時間遅れの範囲に依存してしまうことを解消する。この手法はネットワーク内部で教師出力パターンを出力することに貢献していない結合群を検出し、大域的に共通した時間遅れを不要な時間遅れとして削除する。ここで、式 (7) の第一項が 0 となるような結合は、その結合の結合荷重を変化させても出力が変化しないことを意味する。そこで、そのような結合はすべて不要な結合とみなす。図 3、図 4 に、この概略を示す。図 3 は、入力層と中間層の間の結合について、それぞれ教師出力パターンを出力することに貢献している結合に ○ を、貢献していない結合に × をつけた表である。この表において、時間遅れが $k[ms]$ の結合すべてに × が付いている。そこで、入力層と中間層の間の結合で、時間遅れが $k[ms]$ である結合をすべて削除する。同様に、図 4 にしたがって、中間層と出力層の間の結合で、時間遅れが $k[ms]$ および $k-1[ms]$ である結合をすべて削除する。なお、一度削除された結合は復活することはない。

表 1 初期ネットワーク構造と学習成功率の関係 (調整 WD)

		初期最大時間遅れ		
		8	12	16
初期中間層	5	96	96	×
ユニット数	10	76	80	×
	15	72	82	×

表 2 初期ネットワーク構造と学習回数の関係 (調整 WD)

		初期最大時間遅れ		
		8	12	16
初期中間層	5	1,639	1,548	×
ユニット数	10	2,484	3,868	×
	15	2,595	5,541	×

表 3 初期ネットワーク構造と学習成功率の関係 (調整 WD+ 中間層ユニット削除)

		初期最大時間遅れ		
		8	12	16
初期中間層	5	92	94	×
ユニット数	10	80	90	×
	15	72	90	×

表 4 初期ネットワーク構造と学習回数関係 (調整 WD+ 中間層ユニット削除)

		初期最大時間遅れ		
		8	12	16
初期中間層	5	1,657	1,452	×
ユニット数	10	2,591	3,224	×
	15	2,581	3,996	×

2.3 従来法の適用結果

これらの方法をそれぞれ適用した結果を以下に示す (実験条件は 4 章に準じる)。なお、中間層ユニット数の調整および時間遅れの調整については、調整 WD と組み合わせて適用した。学習成功率を表 1, 3, 5 に、学習回数を表 2, 4, 6 に示す。各表で、最高の学習成功率に対して 0.8 倍未満の学習成功率のセルは灰色に着色し、最低の学習回数に対して 1.5 倍以上の学習回数のセルも灰色に着色した。また、成功しなかったもの × とした。

調整 WD だけを適用した場合、表 1, 2 より、初期中間層ユニット数・初期最大時間遅れのいずれかが大きくなると、学習成功率が低下しかつ学習回数が増加した。調整 WD に加え、中間層ユニット数を調整することで、表 3, 4 より、調整 WD 単体の時と比べ全体的に学習性能の低下は押さえられたが、初期最大時間遅れの増加に対する学習性能の低下は依然大きい。また、調整 WD に加

表 5 初期ネットワーク構造と学習成功率の関係 (調整 WD+ 時間遅れ削除)

		初期最大時間遅れ		
		8	12	16
初期中間層	5	88	92	92
ユニット数	10	92	80	78
	15	78	64	66

表 6 初期ネットワーク構造と学習回数 (調整 WD+ 時間遅れ削除)

		初期最大時間遅れ		
		8	12	16
初期中間層	5	1,615	2,035	1,956
ユニット数	10	2,625	2,625	1,967
	15	2,826	3,509	3,045

え、時間遅れを調整することで、表 5, 6 より、調整 WD 単体の時と比べ学習性能の低下は押さえられたが、初期中間層ユニット数の増加に対しての学習性能の低下は依然大きい。

このように、それぞれの手法は効果を上げているが、効果が上がらない場合をそれぞれ含んでいる。次節ではこれらの手法を組み合わせ、各手法が干渉することなく、学習性能の初期ネットワーク構造への依存を少なくすることをめざす。

3 提案法：ネットワーク構造の自動調整

この節では SpikeProp において中間層ユニット数および時間遅れを自動で調整する方法を組み合わせることで、学習性能の初期ネットワーク構造への依存を少なくする手法について検討する。

2.3 節で示した結果では、中間層ユニット数の自動調整で効果が上がらない場合は、時間遅れの自動調整を適用することで効果が上がった、また逆に、時間遅れの自動調整を適用することで効果が上がらない場合は、中間層ユニット数の自動調整で効果が上がった。そのため、これらの手法を組み合わせることで、より多くの場合に学習性能の低下を防ぐことができると考える。

これらの手法はいずれも、ネットワークの冗長な部分を削除することで学習性能を維持することをめざしている。そのため、これらの手法を単純に組み合わせると、削除しすぎてしまいかえって学習できなくなる可能性がある。また、互いの効果を打ち消さないように、適用順序に注意する必要がある。

中間層ユニットの削除では、学習がある程度進んだときに発火していない中間層ユニットを削除する。また、時間遅れの削除では、式 (7) の第 1 項が 0 となる結合に共通している時間遅れを削除する。このように、これらの

手法は異なる基準で削除を行っている。そのため、手順としては互いに干渉することなく、二つの手法をともに適用できる。そこで、適用順序が問題となる。中間層ユニットの削除は、それにつながるすべての結合の削除と等価である。そのため、過度に中間層ユニットを削除しないように留意する必要がある。ここでは、時間遅れの削除は常に行われる可能性があるが、中間層ユニットの削除は学習がある程度進んでからしか行われないことに着目する。適用の機会が少ない、中間層ユニットの削除を先に行うこととする。

以上をまとめると、提案法の具体的な手順は以下のとおりとなる。

1. ネットワークに教師パターンを入力を与え動作させ、その出力を得る。
2. 誤差 E が十分に減少し、かつ、余分なスパイクを出力していなければ学習終了。
3. 誤差 E が E_d 以下となったら手順 4 へ、そうでなければ手順 5 へ。
4. 発火していない中間層ユニットを一つ選択し削除し、手順 1 へ。削除するユニットが存在しない場合は手順 5 へ。
5. 式 (7) に従い結合荷重を更新する。
6. 式 (7) の第 1 項が 0 となる結合を不要な結合と見なし、不要な結合に大域的に共通する時間遅れを不要な時間遅れと判定する。
7. 不要な時間遅れを持つ結合をすべて削除し、手順 1 へ。

4 実験

この節では、簡単な実験をつうじ、各手法が干渉せず、有効に働いたか確認する。4.1 節で実験条件を示す。また、4.2 節では学習性能の初期ネットワーク構造への依存性を示し、依存性が解消できたかどうかを議論する。4.3 節では調整後のネットワーク構造を示し、ネットワーク構造の調整が過度に行われなかったかを確認する。

4.1 実験条件

実験は、文献 [3] でベンチマーク問題として取り上げられていた時間版 XOR 問題を用いた。これは、論理値をスパイクの発火時刻によりエンコード (早い時刻が論理値 1, 遅い時刻が論理値 0) し、排他的論理和を表したものである。本来これは 2 入力 1 出力の問題だが文献 [3] にならない、バイアス入力として時刻 0 に発火するスパイクをパターンにかかわらず入力 3 に与えている。なお、4 つの教師パターンのいずれも、一つの出力スパイクを期待している。具体的な教師パターンは、表 7 に示す。ネットワークのパラメータは文献 [3] に準じて、表 8 に示すように設定した。中間層-出力層の初期結合荷重は興奮性については $[0.5]$ の一様乱数を、抑制性については $[-1, 0]$ の一

表 7 時間版 XOR 問題

入力	出力
0 0 0	16
0 0 6	10
0 6 0	10
0 6 6	16

表 8 ネットワークのパラメータ

入力層ユニット数	3
中間層ユニット数 (うち抑制性ユニット数)	5, 10, 15 (1, 2, 3)
出力層ユニット数	1
1 結合あたりの副結合数	6, 12, 16
入力層-中間層の初期結合荷重	$[-1, 5]$ の一様乱数
中間層-出力層の初期結合荷重	$[-1, 5]$ の一様乱数
第 k 副結合の時間遅れ d^k	k
発火しきい値 (θ)	7.0
時定数 (τ_m, τ_s, τ_r)	5, 10, 10

表 9 学習のパラメータ

調整 WD	η_H, η_L	0.005, $\eta_H/4$
	ρ_H, ρ_L	0.2, $\rho_H/4$
	$E_{th\uparrow}, E_{th\downarrow}$	1.5, 0.7
中間層ユニット削除	E_d	1.0

様乱数を割り振った。各学習法のパラメータも過去の文献に準じて、表 9 に示すように設定した。

学習の終了条件は、 $E < 1$ かつ出力のスパイク数がすべてのパターンで 1 となったとき (余分なスパイクがなくなったとき) とした。なお、学習回数が 1 万回を超えた場合は、学習失敗とした。以下の結果は、異なる 50 種の初期荷重から学習した結果をまとめたものである。

4.2 学習性能の初期ネットワーク構造への依存性

3.5 節と同様に、さまざまな初期ネットワーク構造から提案法により学習した際の学習成功率と学習回数をそれぞれ表 10, 11 に示す。いずれの結果も、中間層ユニットの削除・時間遅れの削除の一方を適用した場合と比べ、性能が低下しがちであった初期ユニット数・初期時間遅れが大きい場合に向上している。実際、学習成功率が初期ネットワーク構造を下回ったのは、9 ケース中 1 ケースのみとなった。これより、提案法を用いることで、初期のネットワークにほとんど依存せず、おおむね学習が成功するようになったといえる。学習回数についても悪化することなく、従来法と同程度の結果を得ることができた。

以上の結果より、検討の余地はあるものの、学習性能

表 10 初期ネットワーク構造と学習成功率の関係 (提案法)

		初期最大時間遅れ		
		8	12	16
初期中間層	5	88	92	90
ユニット数	10	92	86	78
	15	90	70	78

表 11 初期ネットワーク構造と学習回数 (提案法)

		初期最大時間遅れ		
		8	12	16
初期中間層	5	1,839	2,035	1,960
ユニット数	10	2,641	2,625	2,248
	15	2,931	3,707	3,371

(成功率・学習回数) の初期ネットワーク構造への依存性をより軽減できたといえる。

4.3 学習後のネットワーク構造

この節では学習後のネットワーク構造を調査することで、提案法によりネットワークの構成要素を過剰に削除しなかったかどうかを検討する。

前節の結果より、いずれの場合も、初期中間層ユニット数が少なく初期最大時間遅れが小さいときは、少ない学習回数で確実に学習できている。そのため、これらの場合については、過剰な削除は生じていないと考える。表 12 に、そうでない場合について、学習後に残った中間層ユニットの数と、最大の時間遅れを示す。

表には、最終の中間層ユニット数・入力層-中間層 (I-H) における最終の最大時間遅れ、中間層-出力層 (H-O) の最終最大時間遅れの平均値を示す。一番左の列は初期ネットワークの構造を (初期ユニット数, 初期最大時間遅れ) として表す。例えば、(10, 12) であれば、初期中間層ユニット数が 10 個、副結合数 (最大時間遅れ) が 12 としたことを意味する。表内の従来と提案は、それぞれ従来法と提案法を意味する。提案法は、調整 WD と中間層ユニットの自動調整と時間遅れの自動調整のすべてを組み合わせた方法を指す。従来法は、最終中間層ユニット数の列では調整 WD と中間層ユニットの自動調整を組み合わせた方法を、最終の最大時間遅れの列では調整 WD と時間遅れの自動調整を組み合わせた方法を指す。なおすべての場合において、時間遅れの削除はその値が大きいものから行われており、最少の時間遅れはすべて 1 であった。

この結果より、提案法は最大時間遅れ、中間層ユニットについても個別に適用した場合と同程度に削除を行っているといえる。

以上より、時間遅れ削除、中間層ユニット削除が干渉しなかったことを確認した。

表 12 学習後のネットワーク構造

	中間層		最大時間 遅れ (I-H)		最大時間 遅れ (H-O)	
	従来	提案	従来	提案	従来	提案
(10, 8)	9.8	9.8	8.0	8.0	4.3	4.3
(10, 12)	9.2	9.9	9.1	8.9	4.3	4.1
(10, 16)	×	9.8	9.1	8.9	4.2	4.3
(15, 8)	14.9	14.8	8.0	8.0	4.1	4.1
(15, 12)	13.5	14.7	9.1	8.7	4.0	4.0
(15, 16)	×	14.8	9.2	8.8	4.1	4.1

5 まとめ

本稿では, SpikeProp において, 学習後に出力される余分なスパイクを抑制する手法について検討した. これまでに我々は, 学習中にネットワーク構造を調整する手法 (調整 WD, 調整 WD+ 中間層ユニット削除, 調整 WD + 時間遅れ削除) を提案してきた. これらの手法は独立して提案されており, それらの効果は限定的なものであった. 本稿ではこれらの方法を組み合わせ, 個別で対処できなかったネットワークの依存性の軽減を, 各手法が干渉せず実現できるか検討した. 具体的には, 調整 WD, 中間層ユニット削除, 時間遅れ削除を組み合わせる手法を提案し, その効果を検討した. これにより, 学習中にて中間層ユニット数・時間遅れの調整, 不要な結合を削除を同時に行うことができ, 各手法が干渉せず, 初期ネットワーク構造の依存性を軽減できることがわかった.

参考文献

- [1] Sander M. Bohte: The evidence for neural information processing with precise spike-times : A survey, *Natural Computing*, Vol.3, pp.195–206 (2004).
- [2] Andrzej KASŃISKI and Filip PONULAK: Comparison of Supervised Learning Methods for Spike Time Coding in Spiking Neural Networks, *International Journal of Applied Mathematics and Computer Science*, Vol.16, No.1, pp.101–113 (2006).
- [3] Olaf Booi, Hieu tat Nguyen: A gradient descent rule for spiking neurons emitting multiple spikes, *Information Processing Letters*, Vol.95, pp.552–558 (2005).
- [4] S. M. Bohte, J. N. Kok, and J. A. La Poutre: Error-backpropagation intemporally encoded networks of spiking neurons, *Neurocomputing*, Vol.48, pp.17–37 (2002).
- [5] 松本崇, 新友太, 高瀬治彦, 川中普晴, 鶴岡信治: 拡張 SpikeProp における余分な出力スパイクの抑制 — 調整 WD 法による学習性能の向上 —, 平成 26 年度 電

気・電子・情報関係学会東海部連合大会, J5-1, (2014).

- [6] 松本崇, 新友太, 高瀬治彦, 川中普晴, 鶴岡信治: スパイクングニューラルネットワークにおける複数スパイクの発火時刻の学習法 — ユニット削減による学習成功率の向上 —, 第 30 回ファジィシステムシンポジウム, pp.190–195, (2014)
- [7] 松本崇, 高瀬治彦, 川中普晴, 鶴岡信治: SpikeProp において余分なスパイクを抑制する学習法 — 時間遅れに着目した一検討 —, *電子情報通信学会技術研究報告 (NC)*, Vol.114, No. 362, pp.49–53 (2014).