

## スパイクングニューラルネットにおける マルチスパイク誤差逆伝搬学習法

井上 大輔<sup>†</sup> 服部 元信<sup>††</sup>

<sup>†</sup>山梨大学大学院医学工学総合教育部 <sup>††</sup>山梨大学大学院医学工学総合研究所

### 1 はじめに

近年、生物神経細胞の自然なスパイクの振る舞いが注目を集めている。スパイクとは、神経細胞の興奮が閾値を越えた際に生じる活動電位であり、実際の神経細胞網ではスパイクによって情報が伝達される。また、スパイクはそれぞれに電位差がほとんどない。そのため神経細胞は、スパイクの数やタイミングに基づき情報を処理していると考えられている。しかしながら、一般的なニューロンモデルでは、よりマクロな視点から神経細胞網をモデル化しており、ニューロン間で伝達される情報も実数値などの一次元の情報に集約されている。そこで、スパイクに基づく情報伝達をモデル化（スパイクングニューロンモデル：SNN）することで、多次元の情報を扱うことができ、情報処理能力が向上すると期待されている。

中でも我々は、Bohte らが提案した Spikeprop[1] に注目した。Spikeprop は、SNN を用いた教師あり学習手法の 1 つである。教師あり学習は応用性が高く、比較的速く学習が収束することから有用な学習手法であると言える。また Spikeprop は、導出が BP(Backpropagation) と良く似ており、複雑な非線形タスクの学習が可能という特徴を持つ。しかしながら、Bohte らの手法は単独のスパイクしか考慮しておらず、1 つのニューロンが複数のスパイクを発することができない。そのため、全てのニューロンが最初のスパイクを発するまでの狭い時間範囲しか扱えず、適用可能な問題が限定される他、学習の安定性に大きな問題を抱えていると言える。そこで本研究では、複数のスパイクから成るマルチスパイクパターンを学習できる Spikeprop の改良手法を提案し、パターン認識性能と安定性が向上することを示す。

### 2 Spikeprop

ここでは、Spikeprop を基にマルチスパイクに対応させた提案法の説明を行う。ネットワーク構造と基本的な式の導出は文献 [1] を参照されたい。また提案法

では、ニューロンの膜電位の計算に不応期を導入している。不応期とは、ニューロンが発火した後しばらく発火できなくなる期間を指し、膜電位の計算における負の指数関数によって表される。そのため、結合荷重の修正における誤差の求め方が、従来法と異なり、複数のスパイクに対する誤差を含み、また不応期の影響を考慮したものとなっている。以下に式の導出を示す。

まず、時刻  $t$  におけるニューロンの膜電位を式 (1) に示す。 $x$  が閾値を越えた時刻  $t$  をスパイクを発した時刻として扱う。

$$x_j(t) = \sum_{m \in M} \sum_{ms \in MS} \sum_{k \in L} w_{mj}^k y_{m,ms}^k(t) + \eta'(t - t_{j,last}) \quad (1)$$

$$y_{m,ms}^k(t) = \epsilon(t - t_{m,ms} - d^k) \quad (2)$$

$$\epsilon(t) = \frac{t}{\tau} \exp(1 - \frac{t}{\tau}) \quad (3)$$

$$\eta'(t - t_{j,last}) = -\rho \cdot \exp(-\frac{t - t_{j,last}}{\tau'}) \quad (4)$$

$$\frac{\partial \eta'(t - t_{j,last})}{\partial t} = \frac{\rho}{\tau'} \exp(-\frac{t - t_{j,last}}{\tau'}) \quad (5)$$

ここで、 $i, m, j$  と  $is, ms, js$  は、それぞれ入力、中間、出力層のニューロンとスパイクを指し、大文字はその集合を指す。 $k$  は遅延結合、 $L$  はその集合を指す。 $t_{j,last}$  は、時刻  $t$  の時点でニューロン  $j$  が最後に発したスパイクの時刻を指す。また、式 (2) は遅延を考慮したニューロン活性関数、式 (3) はスパイク反応関数、式 (4) は不応期、式 (5) は不応期の微分を示す。

#### 2.1 出力-中間層間の結合荷重修正

修正量の求め方は最急降下法に則る。まず、出力層での実際のスパイク時刻  $t^a$  と教師  $t^d$  との誤差を求める。

$$E = \frac{1}{2} \sum_{j \in J} \sum_{js \in JS} (t_{j,js}^a - t_{j,js}^d)^2 \quad (6)$$

結合荷重の修正量は次式のようにして求める。

$$\begin{aligned} \Delta w_{mj}^k &= -\eta \frac{\partial E}{\partial w_{mj}^k} \\ &= -\eta \sum_{js \in JS} \left\{ \frac{\partial E}{\partial t_{j,js}^a} (t_{j,js}^a) \frac{\partial t_{j,js}^a}{\partial x_j(t)} (t_{j,js}^a) \frac{\partial x_j(t)}{\partial w_{mj}^k} (t_{j,js}^a) \right\} \\ &= -\eta \sum_{js \in JS} \delta_{j,js} \left( \sum_{ms \in MS} y_{m,ms}^k(t_{j,js}) \right) \end{aligned} \quad (7)$$

Error back propagation for multiple spikes in spiking neural networks

<sup>†</sup> Daisuke INOUE

<sup>††</sup> Motonobu HATTORI

Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi (<sup>†</sup>)

Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi (<sup>††</sup>)

$$\begin{aligned} \delta_{j,j_s} &= \frac{\partial E}{\partial t_{j,j_s}^a} (t_{j,j_s}^a) \frac{\partial t_{j,j_s}^a}{\partial x_j(t)} (t_{j,j_s}^a) \\ &= \frac{-(t_{j,j_s}^a - t_{j,j_s}^d)}{\sum_m \sum_{m_s} \sum_k \{w_{mj}^k \frac{\partial \{y_{m,m_s}^k(t)\}}{\partial t} (t_{j,j_s}^a)\} + \frac{1}{\tau} \exp(-\frac{t_{j,j_s}^a}{\tau})} \end{aligned} \quad (8)$$

2.2 中間-入力層間の結合荷重修正

次に入力層-中間層間の結合荷重の修正量を示す.

$$\begin{aligned} \Delta w_{im}^k &= -\eta \frac{\partial E}{\partial w_{im}^k} \\ &= -\eta \sum_{m_s} \left\{ \frac{\partial E}{\partial t_{m,m_s}^a} (t_{m,m_s}^a) \frac{\partial t_{m,m_s}^a}{\partial x_m(t)} (t_{m,m_s}^a) \frac{\partial x_m(t)}{\partial w_{im}^k} (t_{m,m_s}^a) \right\} \\ &= -\eta \sum_{m_s} \delta_{m,m_s} \left\{ \sum_{i_s} y_{i_s}^k (t_{i_s}^a) \right\} \end{aligned} \quad (9)$$

$$\begin{aligned} \delta_{m,m_s} &\equiv \frac{\partial E}{\partial t_{m,m_s}^a} (t_{m,m_s}^a) \frac{\partial t_{m,m_s}^a}{\partial x_m(t)} (t_{m,m_s}^a) \\ &= \frac{\sum_j \sum_{j_s} \delta_{j,j_s} \sum_k \{w_{mj}^k \frac{\partial \{y_{m,m_s}^k(t)\}}{\partial t} (t_{j,j_s}^a)\}}{\sum_i \sum_{i_s} \sum_k \{w_{im}^k \frac{\partial \{y_{i,i_s}^k(t)\}}{\partial t} (t_{m,m_s}^a)\} + \frac{1}{\tau} \exp(-\frac{t_{m,m_s}^a}{\tau})} \end{aligned} \quad (10)$$

3 計算機実験

提案法の有効性を検証するため, Iris データセットを用いたクラス分類実験を行った. データは, 150 個のサンプルデータからなり, 各データは 4 つの特徴量を持ち, 3 つのクラスに分類される. また, その内 2 つのクラスが線形分離不可能となっている. 実験条件を表 1 に示す. 比較対象には, Bohte らの Spikeprop と Ghosh らが提案した手法 [2] を用いる.

表 1: 実験条件

学習率 $\eta$	0.01	入力ニューロン数	17
遅延結合	16, 4 (*1)	中間ニューロン数	10 (*2)
発火閾値	1	出力ニューロン数	1
訓練データ数	90	活性化関数パラメータ $\tau$	7.0
テストデータ数	60	不応期パラメータ $\tau'$	4.0
		不応期係数 $\rho$	100.0

ここで, (\*1) の 16 は提案法と Bohte の Spikeprop における遅延結合数を, 4 は Ghosh 手法による遅延結合数を指す. (\*2) 中間ニューロンのうち 8 個は興奮性ニューロン, 2 個は抑制性ニューロンとなる. 興奮性ニューロンからのスパイクを受け取ると膜電位が上昇するのに対し, 抑制性ニューロンからのスパイクを受け取ると膜電位が減少する. 図 1 に実験結果のテストデータの認識率の推移を示す. 結果は 10 試行の平均を取ったものである. 図 1 より, 提案手法の学習性能が向上していることが分かる. また, 式 (6) の誤差値の標準偏差を取り, 学習の安定性を調査した. 結果を表 2 に示す. 値は, 学習回数 4000~20000 の間における 10 試行の標準偏差の平均を取ったものである.

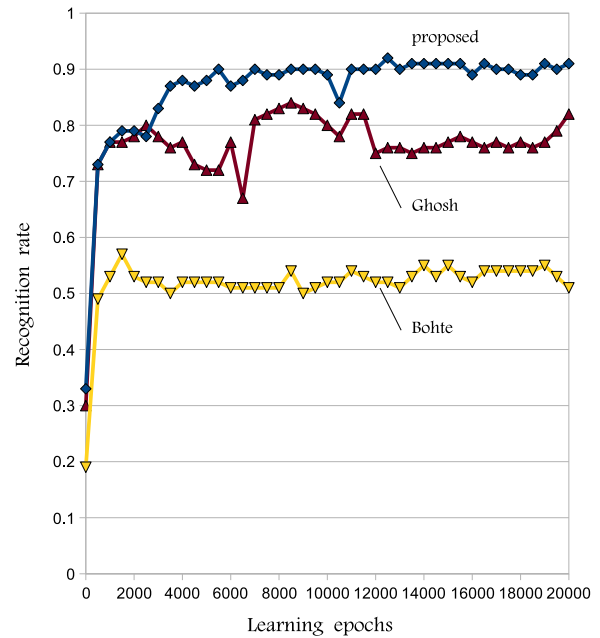


図 1: テスト認識率の推移

表 2: 誤差値 (式 (6)) の標準偏差

proposed	Bohte	Ghosh
0.31	4.07	1.67

結果より, 提案手法ではより学習が安定することが示された.

4 まとめ

SNN の教師あり学習手法である Spikeprop をマルチスパイクに対応させた手法を提案した. また, クラス分類実験により, 従来法よりも高いパターン分離性能と安定性を持つことを示した. ただし, マルチスパイクに対応させている分だけ学習に遅れが発生している. また, SNN は従来のシグモイドニューロンに比べて計算コストが高いため, その応用対象には今後検討が必要である.

参考文献

[1] Sander M. Bohte, Joost N. Kok, Han La Poutre, "Error-backpropagation on temporally encoded networks of spiking neurons," Neurocomputing, Vol. 48, pp.17-37, 2002.

[2] Samanwoy Ghosh-Dastidar, Hojjat Adeli, "A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection," Neural Networks, Vol. 22, pp.1419-1431, 2009.