

修士論文

SpikeProp の学習過程における誤差の
振動の抑制に関する研究

平成22年度修了

三重大学大学院工学研究科

博士前期課程 電気電子工学専攻

泰間 智也

目次

第1章	はじめに	1
第2章	SpikeProp	3
2.1	スパイクニューロン	3
2.2	ネットワーク構造	5
2.3	学習	7
2.4	学習過程における誤差の振動	9
第3章	時間遅れの学習による誤差の振動の抑制	12
3.1	誤差の振動の抑制方法	12
3.2	検証実験	15
3.2.1	誤差の振動の抑制に適した時間遅れの学習率 η_d の検証	15
3.2.2	時間遅れの学習による誤差の振動の抑制効果の検証	17
第4章	遺伝的アルゴリズムを用いた学習による誤差の振動の抑制	19
4.1	誤差の振動の抑制方法	19
4.2	検証実験	20
4.2.1	実数値 GA と SpikeProp の計算量の比較	21
4.2.2	実数値 GA と SpikeProp を切り替えるタイミングの検証	23
第5章	まとめ	26
	謝辞	27
	参考文献	28
付録A	SpikeProp の学習則の導出	30
付録B	XOR 問題	33

第1章 はじめに

近年，生体の脳の情報処理機構をモデル化し，コンピュータによりその動作をシミュレートするスパイクングニューラルネットワーク (Spiking Neural Network) が注目されている [1]．ニューラルネットワークには，その構成ユニットのモデル化手法，ネットワークの構成方法，学習方法の違いにより，さまざまなモデルが存在する．スパイクングニューラルネットワークはその一つであり，スパイク (パルス) を入出力するユニットにより構成されている．その構成ユニット間でやりとりする情報は，スパイクの発火密度や発火タイミングなどを用いて表現される．スパイクを用いた情報表現は，生体の神経回路網でも行われており，生体に近いニューラルネットワークとして期待されている．これに対して，一般に広く用いられているシグモイド型のユニットで構成されるニューラルネットワークでは，これをそのままアナログ値で表現している．シグモイド型のニューラルネットワークは株価予測やロボット制御など幅広く利用されており，その研究も古くから進められている [2][3]．

しかし，スパイクングニューラルネットワークの研究が加速するのは1990年代に入ってからである．スパイクングニューラルネットワークはニューロンの内部電位の時間的な変化を模式化しているため，ニューラルネットワークの中でも時系列処理に適しているという特徴をもつ [4]．それにもかかわらず，注目されるのが遅れたのはコンピュータ上でのシミュレーションが難しかったためである．そもそも，シグモイド型のニューラルネットワークであっても，多数のニューロンの並列処理と繰り返し計算が必要となるために，シミュレーションの負荷が非常に大きく，研究者の障害となっていた．スパイクングニューラルネットワークの場合，内部電位の精密な計算を行うために，シミュレーションの負荷がさらに大きく，深刻な問題であった．

しかし，計算機の性能が向上し，以前に比べてこの問題が深刻なものではなくなったことにより，近年では，スパイクングニューラルネットワークに関する研究も盛んに行われている [5][6]．

スパイクニューラルネットワークを構成するユニット間でやりとりする情報の表現方法には、スパイクの発火密度を用いる表現方法とスパイクの発火タイミングを用いる表現方法がある。スパイクの発火密度を用いた代表的なモデルとしてはSTDP(Spike Timing Dependent Plasticity)と呼ばれるモデルが挙げられる。スパイクニューラルネットワークに関する研究の多くがこのSTDPに関する研究であり、STDPを利用してさまざまなものに応用しようと研究が進められている [7][8]。一方、構成ユニット間でやりとりする情報にスパイクの発火タイミングを用いるモデルは需要がないのかと言うと、そのようなことはない。現にロボット制御にはタイミングの情報が重要であるため、国際的ロボット競技大会であるRoboCupにおいても構成ユニット間でやりとりする情報にスパイクの発火タイミングを用いるモデルが用いられている [9]。しかし、スパイクの発火密度を扱うモデルが研究の主流を占めているのが現状である。そこで、本研究では、構成ユニット間でやりとりする情報にスパイクの発火タイミングを用いる代表的なモデルである SpikeProp[10] に着目する。

SpikeProp は、Bohte らが提案した積分発火モデルによるニューロンにより構成された階層型ニューラルネットワークおよびその教師あり学習法である。これにより、出力層ユニットの発火時刻を、誤差逆伝搬により学習できる。しかし、Takase らにより、シグモイド型ユニットを用いたニューラルネットワークにはない原因で、SpikeProp の学習過程で誤差が振動し、学習が不安定になっていることが指摘された [11]。

そこで、本研究では、学習を阻害する原因である学習過程で生じる誤差の振動を抑制する学習法について検討する。

以下に本論文の構成を示す。2章では研究の対象としている SpikeProp と学習過程における誤差の振動について述べる。3章ではネットワークのパラメータである時間遅れに注目した誤差の振動を抑制する方法について検討する。4章では3章の結果をふまえて誤差の振動を抑制するために遺伝的アルゴリズムを用いた学習法を提案および検討し、最後に5章で本研究をまとめる。

第2章 SpikeProp

本章では、Bohteらが提案したSpikePropについて、それを構成するスパイクングニューロン、そのネットワーク構造、学習法、および、学習過程で生じる誤差の振動について説明する。

2.1 スパイクングニューロン

スパイクングニューロン(以下ユニット)とは、入出力にスパイクを用いるニューロンのモデルである。SpikePropにおけるユニットは、積分発火モデルにもとづいた動作を行う。シナプス前ニューロンからスパイクを受け取り、シナプス強度(結合荷重)に応じた強さでユニットの内部電位が変化する。内部電位があらかじめ定めておいたしきい値 θ を最初に超えた瞬間に、そのユニットは発火し、次のユニット(出力層ユニットの場合は外部)へとスパイクを出力する。多くのスパイクングニューロンのモデル化と異なる点は、各ユニットの発火回数が高々1回であることである。

SpikePropにおけるユニットの具体的な動作を以下に示す。シナプス前ユニット i から時刻 t_i にスパイクが出力されると、シナプス毎に定められた時間遅れ d_{ij} だけ遅れてユニット j に入力される。ニューロン j にこのスパイクが入力されると、ユニット i とユニット j のシナプス結合の種類に応じたシナプス後電位 $y_{ij}(t)$ が発生する。シナプス結合には、興奮性シナプス結合と抑制性シナプス結合があり、興奮性シナプス結合の場合は興奮性シナプス後電位(EPSP:Excitatory PostSynaptic Potential)が発生し、抑制性シナプス結合の場合は抑制性シナプス後電位(IPSP:Inhibitory PostSynaptic Potential)が発生する。それぞれのシナプス後電位の時間変化は式(2.1)で表される。

$$y_{ij}(t) = \begin{cases} \varepsilon(t - t_i - d_{ij}) & (\text{興奮性シナプス後電位}) \\ -\varepsilon(t - t_i - d_{ij}) & (\text{抑制性シナプス後電位}) \end{cases} \quad (2.1)$$

式 (2.1) 中の $\varepsilon(t)$ はスパイク応答関数 (図 2.1) と呼ばれる単峰性の関数であり，次のように定義される．

$$\varepsilon(t) = \begin{cases} \frac{t}{\tau} e^{1-\frac{t}{\tau}} & (t \geq 0) \\ 0 & (t < 0) \end{cases} \quad (2.2)$$

ここで τ は時定数である．

ユニット j の時刻 t における内部電位 $x_j(t)$ は，スパイク応答関数 $\varepsilon(t)$ に結合荷重 w_{ij} を乗じたものとなる．ここで，ユニット j と結合しているシナプス前ユニットの集合を Γ_j とすると，ユニット j の内部電位は式 (2.3) で表される．

$$x_j(t) = \sum_{i \in \Gamma_j} w_{ij} \varepsilon(t - t_i - d_i) \quad (2.3)$$

内部電位 $x_j(t)$ がしきい値 θ を最初に超えたとき，ニューロン j はスパイクを出力する．

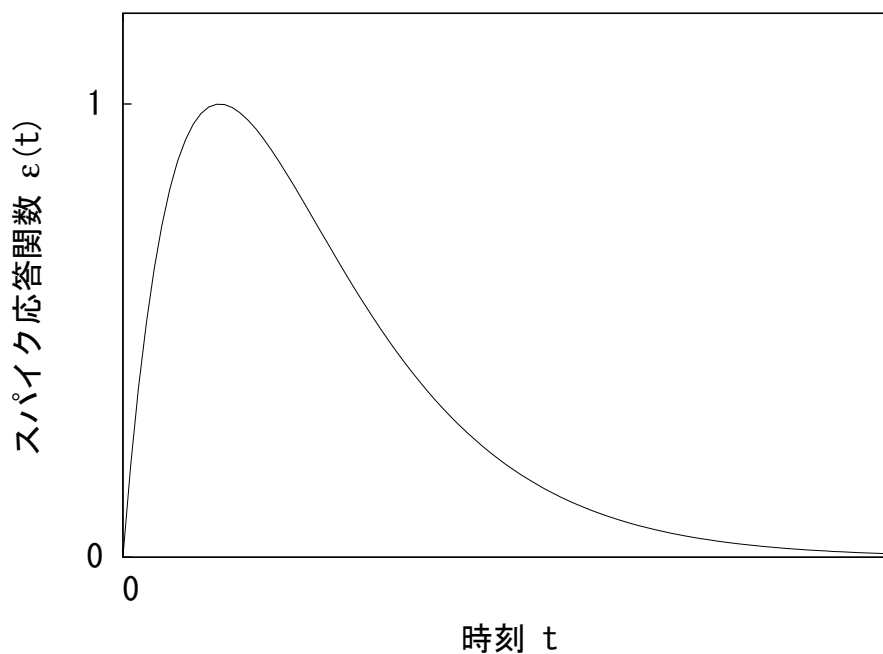


図 2.1: スパイク応答関数の形状

2.2 ネットワーク構造

SpikeProp のネットワーク構造は、図 2.2 に示すような特殊な構造をもつ三層の階層型である。入力されたスパイクは、入力層で分配され中間層ユニット・出力層ユニットを経由して、出力される。

シグモイド型のニューラルネットワークと異なる点は、構成するユニットがスパイクニューロンであることと、ユニット間の結合が、実際には多数の副結合の集合となっている点である。SpikeProp のネットワークは各結合が m 本の副結合から構成されている。各副結合は、調整可能な結合荷重 w_{ij}^k と、定数の時間遅れ d^k をもつ。 w_{ij}^k はユニット i からユニット j への結合における k 番目の副結合の結合荷重を意味する。ユニット i が興奮性である場合はすべての w_{ij}^k を 0 以上とし、抑制性である場合には 0 以下とする。また簡単のために、副結合の時間遅れは、全ての結合 (全てのユニット間) で共通の定数とし、($d^k = k$ のように) 副結合間で一定の差を持つように設定される。副結合も考慮すると、式 (2.3) は式 (2.4) のように書き改められ、ユニットの内部電位は図 2.3 のような変化をする。

$$x_j(t) = \sum_{i \in \Gamma_j} \sum_{k=1}^m w_{ij}^k \varepsilon(t - t_i - d^k) \quad (2.4)$$

他の点については、シグモイド型のニューラルネットワークと同じである。

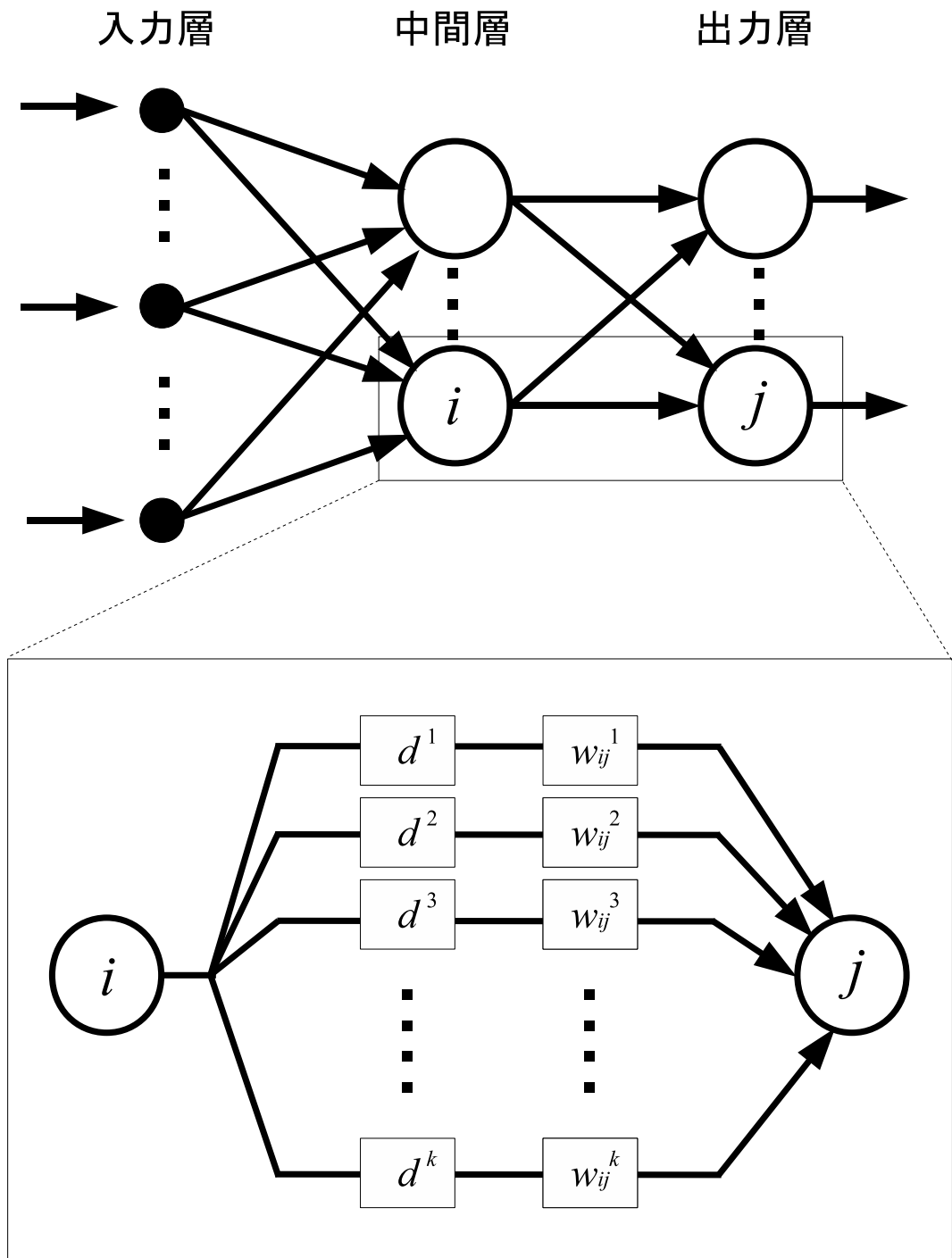


図 2.2: SpikeProp のネットワーク構造

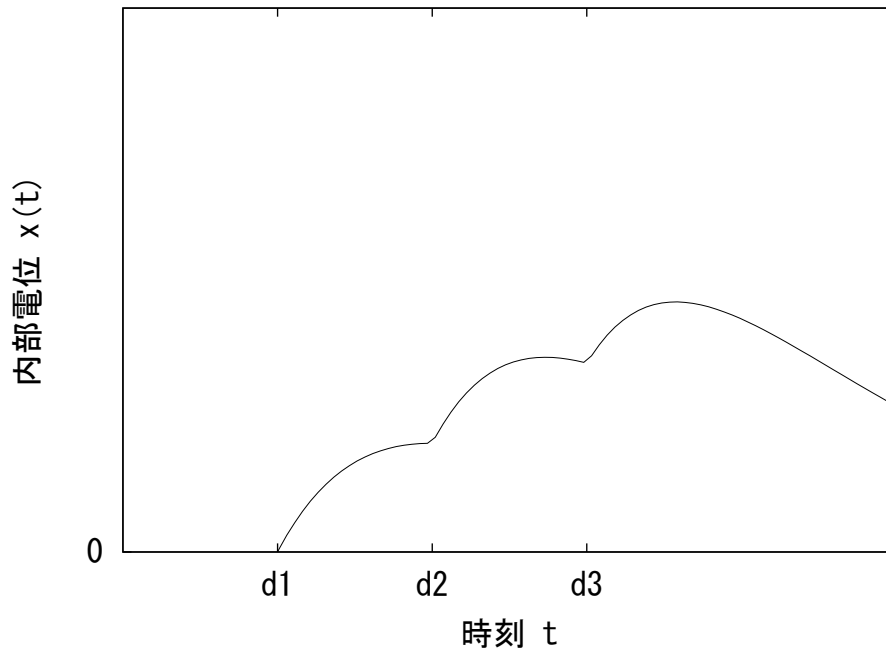


図 2.3: ニューロンの内部電位

2.3 学習

SpikeProp では、誤差逆伝搬法にもとづいて結合荷重を調整することで、出力層ユニットの発火時刻を調整する。SpikeProp での誤差 E は、出力層ユニット j の所望の出力時刻 t_j^d と実際の出力時刻 t_j^a を用いて、式 (2.5) のように定義される (図 2.4)。

$$E = \frac{1}{2} \sum_{j \in J} (t_j^a - t_j^d)^2 \quad (2.5)$$

ここで、 J は出力ユニットの集合である。また、式 (2.6) にしたがって各回の学習を行う。

$$\Delta w = -\eta \frac{\partial E}{\partial w} \quad w \leftarrow w + \Delta w \quad (2.6)$$

ここで η は学習率と呼ばれ、 $\eta > 0$ の定数である。

シグモイド型ユニットを用いたニューラルネットワークと同様に、式 (2.6) 中の勾配 $\frac{\partial E}{\partial w}$ を計算すると、中間層-出力層間の結合荷重 w_{ij}^k に対する勾配と入力層-中間層間の結合荷重 w_{hi}^k に対する勾配は以下のように書き表すことができる。

中間層–出力層間の結合荷重 w_{ij}^k に対する勾配

$$\frac{\partial E}{\partial w_{ij}^k} = y_i^k(t_j^a) \delta_j \quad (2.7)$$

ここで δ_j は式 (2.8) である .

$$\delta_j = \frac{-(t_j^a - t_j^d)}{\sum_{i \in \Gamma_j} \sum_{k=1}^m w_{ij}^k \left\{ \frac{\partial \{y_i^k(t)\}}{\partial t} \Big|_{t=t_j^a} \right\}} \quad (2.8)$$

入力層–中間層間の結合荷重 w_{hi}^k に対する勾配

$$\frac{\partial E}{\partial w_{hi}^k} = y_h^k(t_i^a) \delta_i \quad (2.9)$$

ここで δ_i は式 (2.10) である .

$$\delta_i = \frac{\sum_{j \in \Gamma^i} \left\{ \delta_j \sum_{k=1}^m w_{ij}^k \left\{ \frac{\partial \{y_i^k(t)\}}{\partial t} \Big|_{t=t_j^a} \right\} \right\}}{\sum_{h \in \Gamma_i} \sum_{k=1}^m w_{hi}^k \left\{ \frac{\partial \{y_h^k(t)\}}{\partial t} \Big|_{t=t_i^a} \right\}} \quad (2.10)$$

式 (2.8) , 式 (2.10) の導出方法については付録 A に示す .

式 (2.6) にもとづく学習では, 最急降下方向 (勾配の逆方向) に学習率により定められた量だけ, 結合荷重をわずかに変化させることを繰り返す . そのため, 誤差が十分に減少するまでに, 多数の学習回数 (パラメータ更新) が必要となる . 一般に, ニューラルネットワークの学習には時間がかかることが知られており, スパイキングニューラルネットワークでも例外ではない . そのため, 学習の高速化手法の研究は, 重要な研究課題の一つとされている [12] . 本研究で対象としている SpikeProp に関して, 学習の高速化に関する研究が, いくつか行われている [13][14] . しかし, Takase らにより, シグモイド型ユニットを用いたニューラルネットワークにはない原因で, SpikeProp の学習過程で誤差が振動し, 学習過程が不安定になっていることが指摘された [11] .

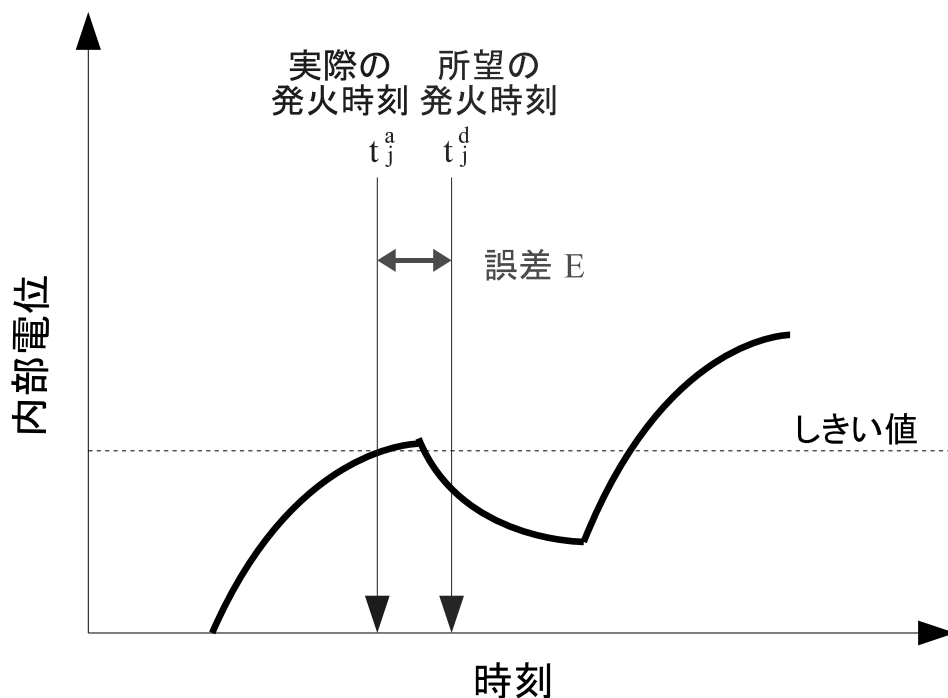


図 2.4: SpikeProp での誤差

2.4 学習過程における誤差の振動

SpikeProp の学習過程では、一般のシグモイド型ユニットを用いたニューラルネットワークの学習時と比べて、誤差が振動しがちである。実際に SpikeProp にて簡単なベンチマーク問題を学習した際の、学習回数に対する誤差の変化の代表例を図 2.5 に示す。この図では、学習開始後しばらくすると、誤差が振動を始めていることがわかる。これは、結合荷重の微小な変化により大きな出力の変化がもたらされる領域を、学習の過程で通過していることを意味する。このような誤差の振動が学習過程に含まれていると、学習の高速化の障害となるだけでなく、学習に失敗したり学習が停滞する (学習回数が増加する) 原因ともなる。シグモイド型ユニットを用いたニューラルネットワークでもこのような誤差の振動は発生するが、その主な原因は大きすぎる学習率によるものである。それに対して、SpikeProp では学習率によらず誤差の振動が発生する。

文献 [11] によると、SpikeProp でのこの振動は、SpikeProp のユニットの発火方法に起因する。以下に簡単に説明する。ユニットは、式 (2.4) にしたがって内部電位を変化させ、内部電位が定められたしきい値 θ を最初に超えた瞬間に発火す

る．スパイク応答関数は連続な関数であるため，スパイク応答関数と結合荷重の積 $w_{ij}\varepsilon(t - t_i - d^k)$ がしきい値 θ を超える時刻は，結合荷重 w_{ij} の変化にともない連続的に変化すると予想できる．しかし，実際には，しきい値 θ を超える時刻は不連続に変化することがある．以下にその原因について説明する．実際の内部電位は多数のスパイク応答関数と結合荷重の積の重ね合わせであるため，多峰性の関数となる．また，SpikeProp では，内部電位がいったん極大値となり再びその値を超えるまでの区間では，しきい値をどのような値にしてもユニットは発火することはない(図 2.6)．そのため，パラメータ更新の結果，内部電位の極大値がしきい値をまたぐように変化する場合，内部電位が最初にしきい値 θ を超える時刻が，連続的に変化しなくなる．その結果，発火時刻が不連続に変化し，誤差が増加することがある．

こういった誤差の増加が発生した後，再度内部電位の極大値がしきい値をまたぐように変化を繰り返すことで，図 2.5 のような誤差の振動が発生する．また，発火時刻が不連続に変化することで，SpikeProp の学習則を導出した際の仮定を満たさなくなる(付録 A 参照)．そのため，シグモイド型のニューラルネットワークの学習の高速化に有効だった方法が，必ずしもうまく働かない．

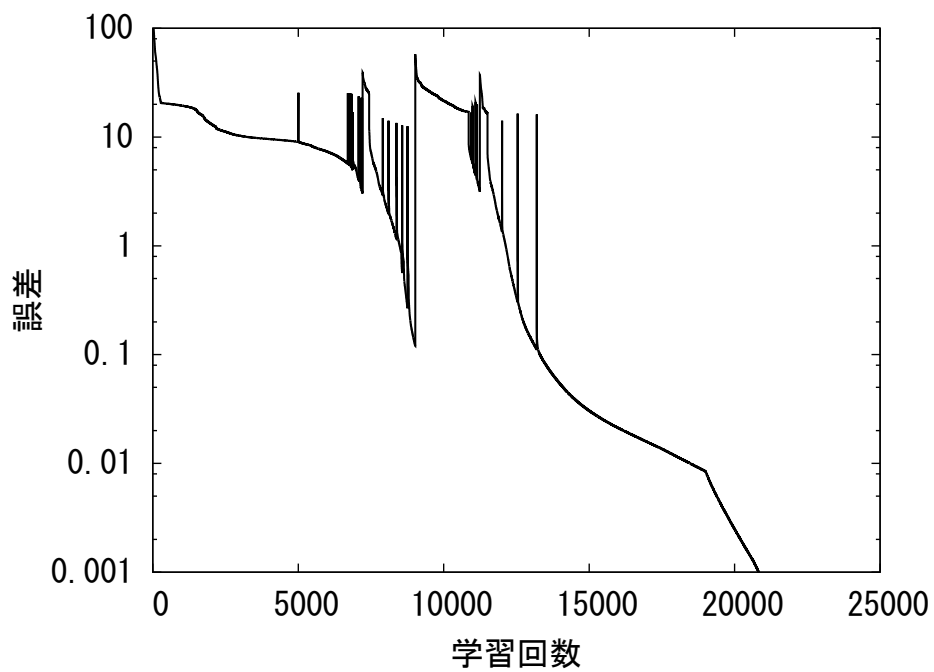


図 2.5: SpikeProp による XOR 問題の学習過程

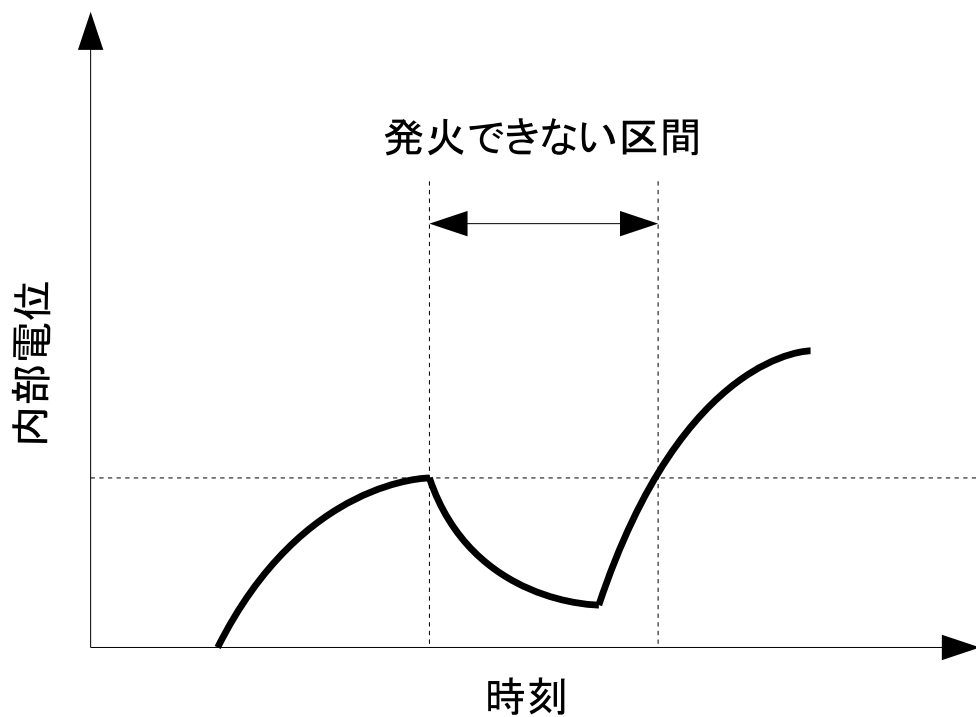


図 2.6: ニューロンの発火できない区間

第3章 時間遅れの学習による誤差の振動の抑制

本章では，SpikeProp の学習過程における誤差の振動を抑制するために，時間遅れに注目した学習法について検討する．

3.1 誤差の振動の抑制方法

Takase らにより指摘された問題は，結合荷重のみを学習している限り避けることができない．結合荷重を調整するだけでは内部電位の時間変化 (図 2.6) は，その高さ方向にしか変化しないため，内部電位が一度極大値になってから再度極大値より大きくなるまでの区間に所望の発火時刻があった場合には，内部電位の極大値がしきい値をまたぐように変化してしまう．そのため，実際の発火時刻が所望の発火時刻を飛び越えて急激に変化し，誤差が増加してしまう (図 3.1)．この誤差の増加が，2.4 節の述べた誤差の振動となる．このような変化を引き起こすのは，当該のユニットに直接接続している結合の結合荷重の変化である．出力層ユニットの出力に対して，中間層ユニットへの結合の結合荷重の変化は，中間層ユニットの発火時刻の変化につながるため，時間軸方向の変化も引き起こす．そのため，振動しつつも誤差を小さくすることはできるが，誤差の振動をなくすことはできない．

そこで，結合荷重だけでなく時間遅れも学習させ，パラメータ更新前後での内部電位の時間変化の多様性を高めることで，この問題の解消を試みる．時間遅れも学習させることで，内部電位の変化がその時間軸方向にも大きく変化するようになるため，誤差の振動 (誤差の増加) を起こすことなく誤差を小さくできることが期待できる (図 3.2) ．

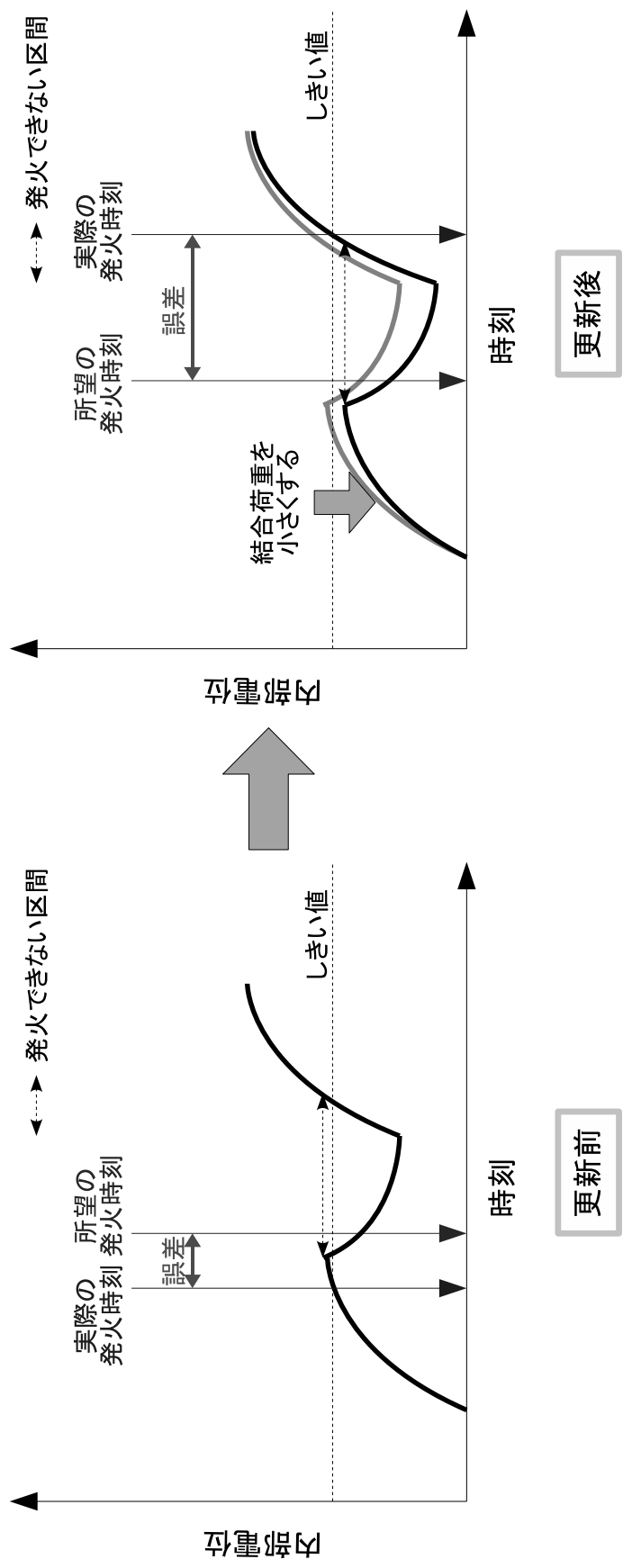


図 3.1: 結合荷重のみ学習する場合のパラメータ更新前後のニューロンの内部電位の時間変化

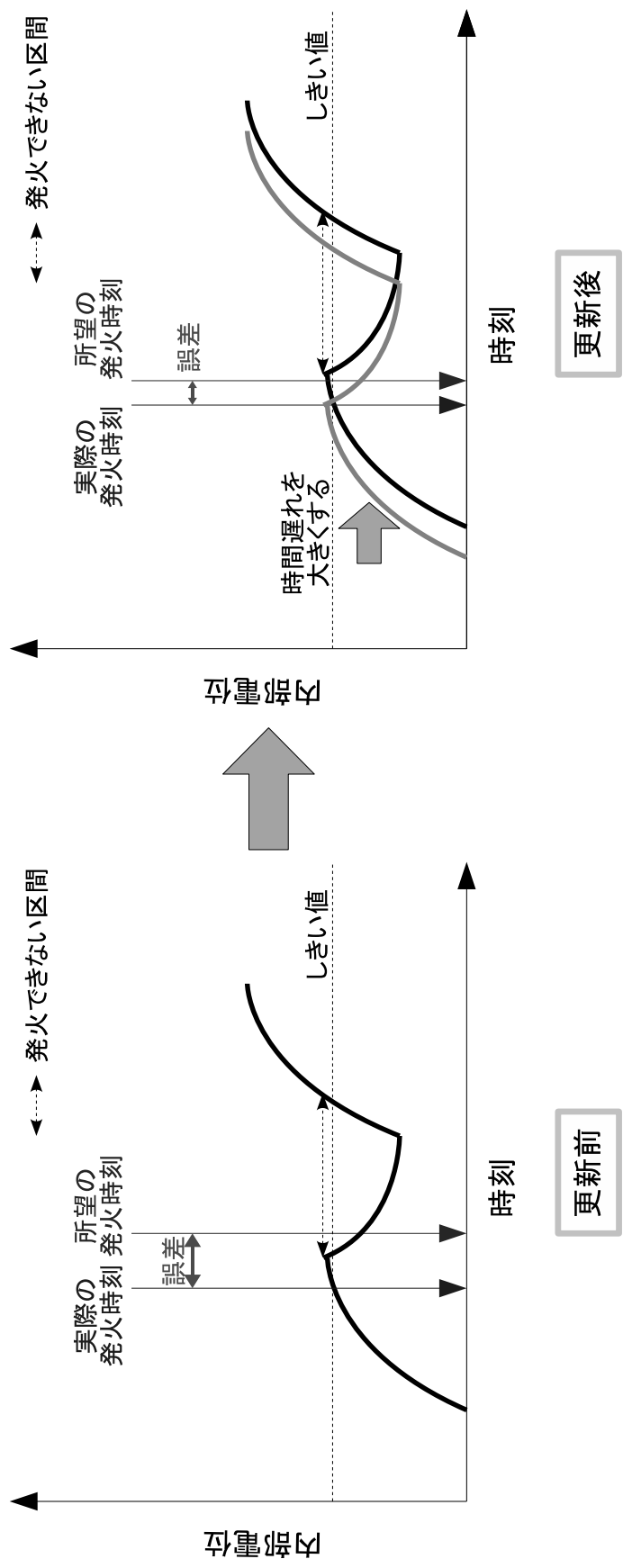


図 3.2: 時間遅れも学習する場合のパラメータ更新前後のニューロンの内部電位の時間変化

SpikePropにおいて、結合荷重以外のネットワークのパラメータを学習させる手法がSchrauwenによって提案されている[15]。この手法では結合荷重以外にも、時間遅れ・しきい値・時定数も学習することで、学習を高速化している。たとえば、時間遅れの学習は以下のように定式化されている。各回の学習による時間遅れの更新は式(3.1)(3.2)にしたがって行う。

$$\Delta d = -\eta_d \frac{\partial E}{\partial d} \quad d \leftarrow d + \Delta d \quad (3.1)$$

$$\begin{aligned} \frac{\partial E}{\partial d_{ij}^k} &= \frac{\partial E}{\partial t}(t_j^a) \frac{\partial t}{\partial x_j(t)}(t_j^a) \frac{\partial x_j(t)}{\partial d_{ij}^k}(t_j^a) \\ &= -w_{ij}^k \frac{\partial \varepsilon}{\partial t}(t_j^a - t_i^a - d_{ij}^k) \delta_j \end{aligned} \quad (3.2)$$

ここで、学習率 η_d は結合荷重の式の学習率 η とは異なるものとする。

文献[15]では、学習の高速化については検証されているものの、学習過程における誤差の振動については検証されていない。そこで、時間遅れも学習することが、学習時の誤差の振動を抑制することにも効果があるのかを次節で検証する。

3.2 検証実験

時間遅れも学習することで、学習過程における誤差の振動を抑制できるかどうかを実験により検証する。学習対象は、Bohteらの論文で用いられたXOR問題を対象とした。XOR問題とは、線形分離不可能な最も簡単なパターン分類問題であり、多くの難しい問題はXOR問題を部分問題として内包するとされている。XOR問題の詳細については付録Bに示す。入力は時刻0または6のスパイクであり、出力は時刻10または16のスパイクである。いずれも、早い時刻のスパイクが論理値1を表し、遅い時刻のスパイクが論理値0を表す。SpikePropのネットワーク構造の概略は、表3.1に示す。なお、XOR問題の二つの入力と、シグモイド型ユニットによるネットワークでバイアスに相当するものとして時刻0のスパイクを入力するために、入力数を3とした。学習に関するパラメータは、SpikePropの結合荷重の学習率 η を1.0とした。

3.2.1 誤差の振動の抑制に適した時間遅れの学習率 η_d の検証

文献[15]によると、SpikePropの学習の高速化に効果的な結合荷重の学習率 η と時間遅れの学習率 η_d の比率は $10^4:1$ である。しかし、この比率が誤差の振動の抑

表 3.1: SpikeProp ネットワークの諸元

入力層ユニット数	3
中間層ユニット数	5
うち興奮性	4
出力層ユニット数	1
副結合での時間遅れ	$1, 2, \dots, m$
スパイク応答関数の τ	7.0
しきい値	10.0

表 3.2: 誤差の振動の抑制に効果的な時間遅れの学習率 η_d の検証

時間遅れの学習率 η_d	誤差の振動回数 [回]	学習失敗回数 [回]
10^{-2}	108	300
10^{-3}	33.1	48
10^{-4}	31.2	47
10^{-5}	30.0	43
10^{-6}	29.6	41

制に対しても効果的かどうかは検証されていない。そこで、結合荷重の学習率 η を 1.0 としたときに、誤差の振動の抑制に効果的な時間遅れの学習率 η_d の値について検証する。

具体的に時間遅れの学習率 η_d の値は、学習の高速化に効果的である比率である 10^{-4} を中心として、 10^{-2} 、 10^{-3} 、 10^{-5} 、 10^{-6} とした場合についてそれぞれ比較を行う。

1000 種類の異なる結合荷重から学習した結果、誤差が 0.01 未満となるまでに発生した誤差の振動の回数の平均を表 3.2 に示す。ここで、誤差の振動の回数とは、1 回のネットワークのパラメータの更新により、誤差が上昇した回数を意味する。

表 3.2 より、結合荷重の学習率 η を 1.0 としたときに、誤差の振動の抑制に効果的な時間遅れの学習率 η_d は 10^{-6} であることがわかった。ただ、時間遅れの学習率 η_d が 10^{-3} 以下のときには、どの場合についても大差はみられなかった。

3.2.2 時間遅れの学習による誤差の振動の抑制効果の検証

結合荷重のみを学習する従来法と時間遅れも学習する方法との差を実験により検証する．時間遅れも学習すると，その分可調整パラメータ数が増えてしまう．しかし，オッカムの剃刀として知られる，「事物を説明するにあたり，必要ない仮定をすべきではない」という考え方にもあるように，調整すべきパラメータ数が多いと，ノイズに過敏になり，過学習を起こしてしまう [16]．そこで，従来法と同じネットワーク構造で時間遅れも学習する場合だけでなく，従来法と同じ可調整パラメータ数で時間遅れも学習する場合についても比較を行う．具体的には，時間遅れの学習率 η_d と 1 結合あたりの副結合数をそれぞれ 0, 16 とした場合， 10^{-6} , 16 とした場合， 10^{-6} , 8 とした場合について比較を行う．

1000 種類の異なる結合荷重から学習した結果，誤差が 0.01 未満となるまでに発生した誤差の振動の回数を表 3.3 に示す．

表 3.3 では，従来法と比べ，従来法と同じネットワーク構造で時間遅れも学習した場合には誤差の振動回数は少ない．しかし，有意水準 1% で t 検定を行った結果，有意差はなかった．つまり，従来法と同じネットワーク構造で時間遅れも学習しても，誤差の振動の抑制に効果があったとはいえない．また，従来法と同じ可調整パラメータ数で時間遅れを学習した場合には，誤差の振動がかえって増加してしまった．これは，時間遅れを学習することよりも，結合荷重の数を半分にしたことによる影響が大きく現れていると考えられる．

以上の結果より，単純に時間遅れを学習させても誤差の振動を抑制できないことが分かった．誤差の振動を抑制するには，結合荷重および時間遅れの調整方法に工夫が必要と考えられる．特に，新たに時間遅れを学習することによるメリットよりも元々学習していた結合荷重の数を減らすことによるデメリットの方が大きかったことから，結合荷重の調整方法を工夫することが有効と考えられる．

表 3.3: 時間遅れの学習と学習過程との関係

	学習率		可調整パラメータ数	誤差の振動の回数 [回]
	結合荷重 η	時間遅れ η_d		
結合荷重のみ学習 従来法	1.0	0	320	34.7
時間遅れも学習 同じネットワーク構造	1.0	10^{-6}	640	29.6
時間遅れも学習 同じ可調整パラメータ数	1.0	10^{-6}	320	55.2

第4章 遺伝的アルゴリズムを用いた学習による誤差の振動の抑制

本章では，SpikeProp の学習過程における誤差の振動を抑制するために遺伝的アルゴリズムを用いた学習法について検討する．

4.1 誤差の振動の抑制方法

前章での結果をふまえ，誤差逆伝搬法以外の学習法に注目する．誤差の勾配情報のみでパラメータを調整する SpikeProp で結合荷重を学習する場合，時間遅れも学習しても，パラメータの更新前後で内部電位の極大値がしきい値をまたぐような変化を避けられなかった．

そこで，遺伝的アルゴリズムに注目する．誤差の勾配情報を必要としない進化的計算手法である遺伝的アルゴリズムであれば，式 (2.8)・式 (2.10) を導出した仮定を満たす必要がなくなる．そのため，SpikeProp 固有の原因による誤差の振動が顕在化しなくなる．

SpikeProp の学習は，実数値のパラメータである結合荷重を調整することで誤差を最小化する非線形最適化問題と見なすことができる．このような問題には実数値遺伝的アルゴリズム (実数値 GA) が適用可能である．しかし，実数値 GA によりニューラルネットワークを学習する場合，多数の個体 (ニューラルネットワーク) が必要であるため，SpikeProp (誤差逆伝搬法) での学習時にも増して，ニューラルネットワークの出力計算を行う必要がある．そのため，実数値 GA による学習と勾配法による学習 (SpikeProp) を組み合わせることで，その計算量を軽減する．

ここで，図 2.5 に着目する．この図では，誤差が振動しているのは誤差が一定値 (図では 0.01) より大きいときである．これは，図 2.5 の例だけでなく，予備実験として行った 10 試行の結果，全てにあてはまった．この結果は，いったん誤差が振動する領域を抜ければ，その後の勾配法にもとづいた学習では再び誤差が振動することがないことを示唆している．よって，誤差が振動する領域のみを実数値 GA

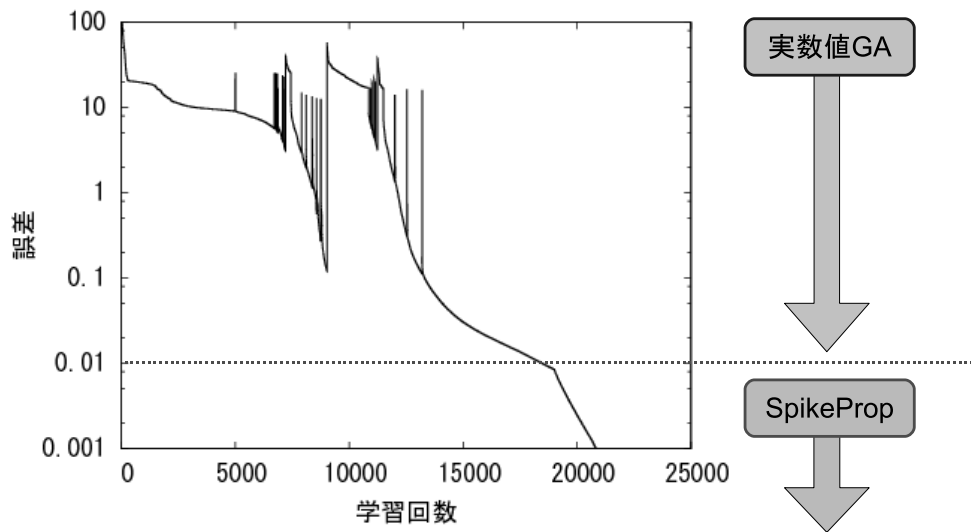


図 4.1: 提案法の概要

を用いて学習すれば，誤差の振動を起こすことなく学習できると考えられる．では，基本的には SpikeProp で学習し，誤差が振動 (増加) したときのみ実数値 GA を用いて学習すればいいかということ，それは難しい．なぜならば，実数値 GA から SpikeProp に切り替えるときは，実数値 GA での最適解をそのまま SpikeProp に用いればよいが，SpikeProp から実数値 GA に切り替えるには，実数値 GA での初期個体をどうするかという問題があるためである．

そのため，まず実数値 GA を用いて誤差がある程度減少するまで学習し，続けて SpikeProp で誤差を十分に減少させる手法を提案する (図 4.1)．これにより，学習行程全体を SpikeProp で学習した場合に発生していた誤差の振動は実数値 GA により抑制し，誤差が振動する領域を抜けたあとは SpikeProp で (実数値 GA と比べて) 高速に学習を進める．また，実数値 GA で誤差が振動する領域を回避することで，その後の学習では，ネットワーク出力が結合荷重の変化にともない連続的に変化すると考えられる．そのため，各種非線形最適化手法により，後半の SpikeProp の学習をさらに高速化することも可能である．

4.2 検証実験

遺伝的アルゴリズムを用いて結合荷重を調整することで，学習過程における誤差の振動を抑制できるかどうかを実験により検証する．

表 4.1: SpikeProp ネットワークの諸元

入力層ユニット数	3
中間層ユニット数	5
うち興奮性	4
出力層ユニット数	1
1 結合あたりの副結合数	16
副結合での時間遅れ	1, 2, \dots , 16
スパイク応答関数の τ	7.0
しきい値	10.0

提案手法では、まずスパイクニューラルネットワークを実数値 GA により学習する。ここでは、世代交代モデルとして MGG[17] を、交叉法としてシンプレクス交叉 (SPX)[18] を組み合わせた手法を適用した。なお、詳細については樋口らの論文 [18] の実験条件に準じた。

学習対象は、前章と同様に Bohte らの論文で用いられた XOR 問題を対象とした。SpikeProp のネットワーク構造の概略を表 4.1 に示す。学習に関するパラメータは、SpikeProp の学習率 η を 0.01、SPX の拡張率 ϵ を $\sqrt{322}$ 、SPX で 1 世代あたりに生成する子個体数を 960 とした。また、学習終了条件は誤差が 10^{-3} 未満となったときとした。この条件は、XOR 問題に対しては過剰であるが、より複雑・大規模な問題への適用を考え厳しく設定した。

4.2.1 実数値 GA と SpikeProp の計算量の比較

4.1 節では、実数値 GA で学習するよりも SpikeProp で学習する方が計算量が少なくすむものとして話を進めていた。ここで、実際に SpikeProp のネットワークを実数値 GA、SpikeProp のそれぞれで学習する際の計算量を比較する。

比較するのは、次の 2 点とする。

- (1) SpikeProp の学習と、実数値 GA を用いた学習でそれぞれ 1 回のパラメータ更新にかかる計算量
- (2) 一定の誤差を減少させるために必要なパラメータの更新回数

表 4.2: パラメータ更新にかかる計算量

	実数値 GA	SpikeProp
1 回のパラメータ更新にかかった時間 [s]	1.8	9.7×10^{-3}

表 4.3: 一定量の誤差を減少させるのに必要なパラメータ更新回数

	実数値 GA	SpikeProp
パラメータ更新回数 [回]	3,254	2,210

(1) では, 1,000 回の SpikeProp の学習と, 1,000 世代分の実数値 GA による学習を行い, SpikeProp と実数値 GA それぞれについて 1 回のパラメータ更新にかかる時間を求める. また, (2) では学習時に誤差が振動してしまうと, 学習を阻害してしまう. そこで, 予備実験の結果 SpikeProp で誤差の振動が発生しなかった区間 (誤差が 0.01 より小さい区間) である誤差を 10^{-2} から 10^{-3} にするのに必要なパラメータの更新回数を求める.

(1) の結果を表 4.2 に示す. 10 試行の平均として, 1 回の SpikeProp の学習には 9.7×10^{-3} 秒かかり, 1 世代の実数値 GA には 1.8 秒かかることが分かった. この結果は, 実装方法に多少は左右されるものの, 1 世代分の実数値 GA の計算量は, 100 回分以上の SpikeProp の学習に要する計算量に相当することが分かる.

(2) の結果を表 4.3 に示す. 10 回の試行の平均として, SpikeProp では 2,210 回の学習を要し, 実数値 GA では 3,254 世代の学習を要することが分かった. (1) の結果と合わせると, SpikeProp と実数値 GA で誤差を 10^{-2} から 10^{-3} にするのに, それぞれ 21 秒, 5.9×10^4 秒の時間を要することが分かった. この結果から, 一定の誤差を減少させるのに必要な計算量も実数値 GA よりも SpikeProp を用いた方が少ないことがわかる.

以上の結果より, SpikeProp で学習する方が, より短時間に学習ができることがわかる. しかし, SpikeProp だけでは学習過程で誤差の振動が起きてしまう. このことから, 誤差の振動が発生しなくなるまで実数値 GA で学習させ, そこから SpikeProp に切り替えるという方法により, 学習過程の安定性の向上と, 学習時間の短縮を両立できるといえる.

4.2.2 実数値 GA と SpikeProp を切り替えるタイミングの検証

実数値 GA と SpikeProp を切り替えるタイミングと、学習過程における誤差の振動との関係について検討する。

4.1 節で述べたように、誤差が振動しているのは誤差が一定値より大きい区間のみである。このことから、以下の 4 種類の条件で学習する際の学習過程を比較する。

- (1) 最初から SpikeProp を用いる方法
- (2) 学習中盤 (振動が起こる可能性のある領域, 最初に誤差が 0.1 を下回った時点) で SpikeProp に切り替える方法
- (3) 学習終盤 (振動が完全に起きなくなった領域, 最初に誤差が 0.01 を下回った時点) で SpikeProp に切り替える方法【提案法】
- (4) 切り替えない方法 (実数値 GA のみで学習)

10 回の試行結果の平均として、誤差が振動した試行の回数 (失敗試行数)、実数値 GA での学習の計算量、SpikeProp での学習の計算量、学習全体の計算量を表 4.4 に示す。ここで、失敗試行数とは、SpikeProp で学習中に 1 回のネットワークのパラメータの更新により、誤差が増加した試行の回数を意味する。また、計算量とは、実数値 GA の 1 世代にかかる計算量を 1 としたときの計算量である。なお、実数値 GA を適用した区間では、世代交代モデルとして実数値 GA での最適解を必ず残す MGG を採用したため、誤差は単調に減少する。

表 4.4 より、学習終盤で SpikeProp に切り替えた場合、誤差の振動が無く、実数値 GA のみの場合と比べて計算時間も少ないことが分かる。ただし、SpikeProp への切り替えのタイミングが早かった場合、SpikeProp のみで学習した場合とかわらない程度の誤差の振動がみられる。この場合は、SpikeProp のみの場合と比べて、実数値 GA を用いている分だけ計算時間が長くなっており、提案法の悪い面のみが表れている。また、SpikeProp のみで学習する場合と比べると、いずれの場合も多大な実数値 GA の計算が必要な点には、今後の検討が必要である。

図 4.2 に学習終盤で SpikeProp に切り替える方法による学習過程の例を示す。横軸は実数値 GA の 1 世代にかかる計算量を 1 としたときの計算量、縦軸は誤差を表す。この例では、実数値 GA の 4,410 世代目で SpikeProp に切り替わった。この結果からも、誤差が振動する区間は実数値 GA により確実に学習し、誤差が振動

しなくなった時点で SpikeProp に切り替えることで、実数値 GA のみの場合と比べ学習時間を短縮できていることが確認できる。

以上の結果から、提案を用いるうえで、実数値 GA と SpikeProp の切り替えのタイミングを適切に指定することが必要となることがわかる。今回の実験では、予備実験として行った SpikeProp のみでの学習結果からこのタイミングを定めたが、この時点でいったん学習を完了しているので、再び学習する必要がなく、実用的ではない。今後の課題として、実数値 GA と SpikeProp の適切な切り替えのタイミングを導出する方法の検討が挙げられる。

表 4.4: 実数値 GA と SpikeProp を切り替えるタイミングの検証

切り替え時の 誤差	失敗 試行数	計算量 (世代数相当)		
		GA	SP	全体
SpikeProp のみ	5	—	54	54
0.1	6	2,630	57	2,687
0.01	0	4,547	12	4,559
実数値 GA のみ	0	7,560	—	7,560

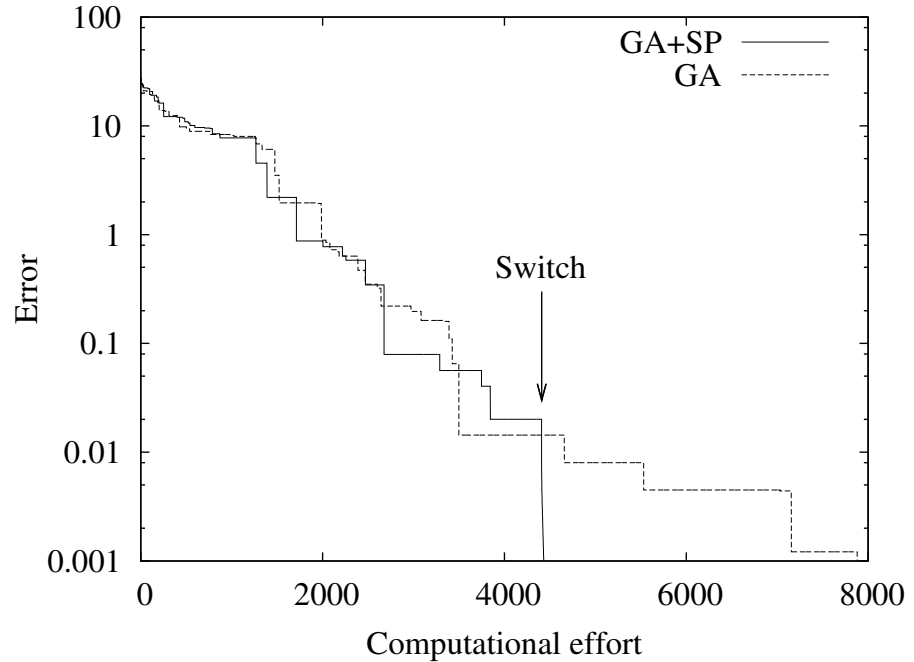


図 4.2: 提案法による XOR 問題の学習結果

第5章 まとめ

本研究では，SpikeProp の学習過程における誤差の振動を抑制することを目的とした．

学習時に調整するパラメータとして時間遅れを増やすことで，学習前後での内部電位の時間変化の多様性を高め，Takase らによって指摘された SpikeProp 固有の原因による誤差の振動の抑制を試みた．しかし，実験の結果，時間遅れを学習させても誤差の振動を抑制するのに効果が得られないことが分かった．

この結果をふまえて，遺伝的アルゴリズムを用いた学習法に注目した．誤差逆伝搬法のように勾配情報のみを考慮して学習する場合，時間遅れも学習しても内部電位の極大値がしきい値をまたぐような変化を避けられなかった．しかし，誤差の勾配情報を必要としない遺伝的アルゴリズムであれば，SpikeProp 固有の原因による誤差の振動が顕在化しなくなる．ただ，単純に遺伝的アルゴリズムによりニューラルネットワークを学習する場合，SpikeProp での学習時にも増して，ニューラルネットワークの出力計算を何度も行う必要がある．そのため，遺伝的アルゴリズムによる学習と SpikeProp による学習を組み合わせた学習法を提案した．実験の結果，提案法を用いることで誤差の振動を起こすことなく学習できることが分かった．また，遺伝的アルゴリズムのみで学習した場合と比べて計算時間を少なくすることができた．

今後の課題としては，遺伝的アルゴリズムと SpikeProp との適切な切り替えのタイミングを導出する方法の検討が挙げられる．

謝辞

本論文は、著者が三重大学大学院工学研究科博士前期課程時に行った研究をまとめたものである。本論文を進めるにあたり、懇切丁寧な御指導と御督励を賜った三重大学の鶴岡信治教授，高瀬治彦准教授，北英彦准教授，川中助教に深く感謝いたします。また，日頃熱心に討論していただいた情報処理研究室の皆様方に厚く御礼申し上げます。

最後に，本論文をまとめるにあたり，助言，討論，その他お世話になったすべての方々に感謝いたします。

参考文献

- [1] Wolfgang Mass and Christopher M. Bishop: Pulsed Neural Networks, The MIT Press, pp.3–85 (2001)
- [2] Yamashita T, Hirasawa K, Jinglu Hu: Application of multi-branch neural networks to stock market prediction, Proceedings of International Joint Conference on Neural Networks 2005, pp. 2544–2548 (2005).
- [3] Watanabe K, Jun Tang, Nakamura M, Koga S, Fukuda T: A fuzzy-Gaussian neural network and its application to mobile robot control, IEEE Transactions on Control Systems Technology, Vol.4, Issue2, pp.193–199(1996)
- [4] 瀧田航一郎: パルスニューラルネットワークによる時系列情報処理に関する研究, 慶応義塾大学博士論文 (2005)
- [5] Alnajjar F, Murase K: Sensor-fusion in spiking neural network that generates autonomous behavior in real mobile robot, Proceedings of International Joint Conference on Neural Networks 2008, pp. 2200–2206 (2008)
- [6] Eugene M. Izhikevich: Solving the Distal Reward Problem through Linkage of STDP and Dopamine Signaling, Cerebral Cortex Vol.17, pp.2443–2452(2007)
- [7] 谷口忠大 榎木哲夫: シェマデルと STDP 則の結合による記号過程の創発, 人工知能学会全国大会論文集 Vol. 20, 3C1–3(2006)
- [8] 田中秀樹, 森江隆, 合原一幸: STDP を有する CMOS スパイキングニューラルネットワーク LSI の評価, 電子情報通信学会技術研究報告, NC107(328), pp37–42(2007)
- [9] Xiuqing Wang, Zeng-Guang Hou, Anmin Zou, Min Tan, Long Cheng: A behavior controller based on spiking neural networks for mobile robots, Neurocomputing 71, pp.655–666 (2008)

- [10] Sander M. Bhoite, Joost N. Kok, and Han La Pouté: Error-backpropagation in temporally encoded networks of spiking neurons, *Neurocomputing*, Vol.48, pp.17–37 (2002)
- [11] H. Takase, M. Fujita, H. Kawanaka, S. Tsuruoka, H. Kita, and T. Hayashi: Obstacle to Training SpikeProp Networks — Cause of Surges in Training Process —, *Proceedings of International Joint Conference on Neural Networks 2009*, pp. 3062–3066 (2009)
- [12] 黒江康明: スパイクニューラルネットワーク—学習法を中心として, システム制御情報学会誌, Vol.48, No.2, pp.57–62 (2004)
- [13] Sam McKennoch, Dingding Liu, and Linda G. Bushnell: Fast Modifications of the SpikeProp Algorithm, *Proceedings of International Joint Conference on Neural Networks 2006*, pp.3970–3977 (2006)
- [14] Benjamin Schrauwen, and Jan Van Campenhout: Improving SpikeProp: Enhancements to An Error-Backpropagation Rule for Spiking Neural Networks, *Proceedings of the 15th ProRISC Workshop*, pp.301–305 (2004)
- [15] Benjamin Schrauwen, and Jan Van Campenhout: Extending SpikeProp, *Proceedings of International Joint Conference on Neural Networks 2004*, pp. 471–475 (2004).
- [16] 若山正人, 小西貞則, 竹内純一: 現代技術への数学入門シリーズ 統計的モデリング / 情報理論と学習理論-データと上手につきあう法, 講談社, pp.102–103 (2008)
- [17] 佐藤浩, 小野功, 小林重信: 遺伝的アルゴリズムにおける世代交代モデルの提案と評価, *人工知能学会誌*, Vol. 12, No.5, pp. 734–744 (1997)
- [18] 樋口隆英, 筒井茂義, 山村雅幸: 実数値 GA におけるシンプレクス交叉の提案, *人工知能学会論文誌*, Vol. 16, No. 1, pp. 146–155 (2001)

付録 A SpikeProp の学習則の 導出

ここでは, SpikeProp の結合荷重の更新式で用いた δ_j および δ_i (式 (2.8), 式 (2.10)) の導出方法について説明する.

中間層-出力層間の結合荷重 w_{ij}^k に対する勾配について, ユニット j の実際の発火時刻を t_j^a とすると

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}^k} &= \frac{\partial E}{\partial t_j}(t_j^a) \frac{\partial t_j}{\partial w_{ij}^k}(t_j^a) \\ &= \frac{\partial E}{\partial t_j}(t_j^a) \frac{\partial t_j}{\partial x_j(t)}(t_j^a) \frac{\partial x_j(t)}{\partial w_{ij}^k}(t_j^a) \end{aligned} \quad (\text{A.1})$$

ここで, $t = t_j^a(x_j(t))$ がしきい値) 付近の微小な区間において, 出力層ユニットの発火時刻 $t = t_j^a$ が連続的に変化するとき, 図 A.1 のようにユニットの内部電位 $x_j(t)$ も連続的に変化する仮定すると,

$$\begin{aligned} \frac{\partial t_j}{\partial x_j(t)}(t_j^a) &= \left. \frac{\partial t_j(x_j)}{\partial x_j(t_j)} \right|_{x_j=\theta} \\ &= \frac{1}{-\frac{\partial x_j(t)}{\partial t}(t_j^a)} \\ &= \frac{-1}{\sum_{i \in \Gamma_j} \sum_{k=1}^m w_{ij}^k \left\{ \left. \frac{\partial \{y_i^k(t)\}}{\partial t} \right|_{t=t_j^a} \right\}} \end{aligned} \quad (\text{A.2})$$

となる. また,

$$\frac{\partial E}{\partial t_j}(t_j^a) = t_j^a - t_j^d \quad (\text{A.3})$$

$$\frac{\partial x_j(t)}{\partial w_{ij}^k}(t_j^a) = y_j^k(t_j^a) \quad (\text{A.4})$$

でとなるので, 式 (2.7) の δ_j は式 (2.8) となる.

入力層–中間層間の δ_i は一般的に次のように定義される .

$$\begin{aligned}
\delta_i &\equiv \frac{\partial t_i^a}{\partial x_i} (t_i^a) \frac{\partial E}{\partial t_i^a} \\
&= \frac{\partial t_i^a}{\partial x_i} (t_i^a) \sum_{j \in \Gamma_i} \frac{\partial E}{\partial t_j^a} \frac{\partial t_j^a}{\partial x_j(t_j^a)} \frac{\partial x_j(t_j^a)}{\partial t_i^a} \\
&= \frac{\partial t_i^a}{\partial x_i} (t_i^a) \sum_{j \in \Gamma_i} \delta_j \frac{\partial x_j(t_j^a)}{\partial t_i^a}
\end{aligned} \tag{A.5}$$

中間層–出力層間と同様に , 中間層ユニットの発火時刻 $t = t_i^a$ が連続的に変化する と , ユニットの内部電位 $x_i(t)$ も連続的に変化する仮定すると ,

$$\begin{aligned}
\frac{\partial t_i^a}{\partial x_i} (t_i^a) &= \left. \frac{\partial t_i(x_i)}{\partial x_i(t_i)} \right|_{x_i=\theta} \\
&= \frac{1}{-\left. \frac{\partial x_i(t)}{\partial t} \right|_{t=t_i^a}} \\
&= \frac{-1}{\sum_{h \in \Gamma_i} \sum_{k=1}^m w_{hi}^k \left\{ \left. \frac{\partial \{y_h^k(t)\}}{\partial t} \right|_{t=t_i^a} \right\}}
\end{aligned} \tag{A.6}$$

$$\frac{\partial x_j(t_j^a)}{\partial t_i^a} = - \sum_{k=1}^m w_{ij}^k \left\{ \left. \frac{\partial \{y_i^k(t)\}}{\partial t} \right|_{t=t_j^a} \right\} \tag{A.7}$$

となるので , 式 (2.9) の δ_i は式 (2.10) となる .

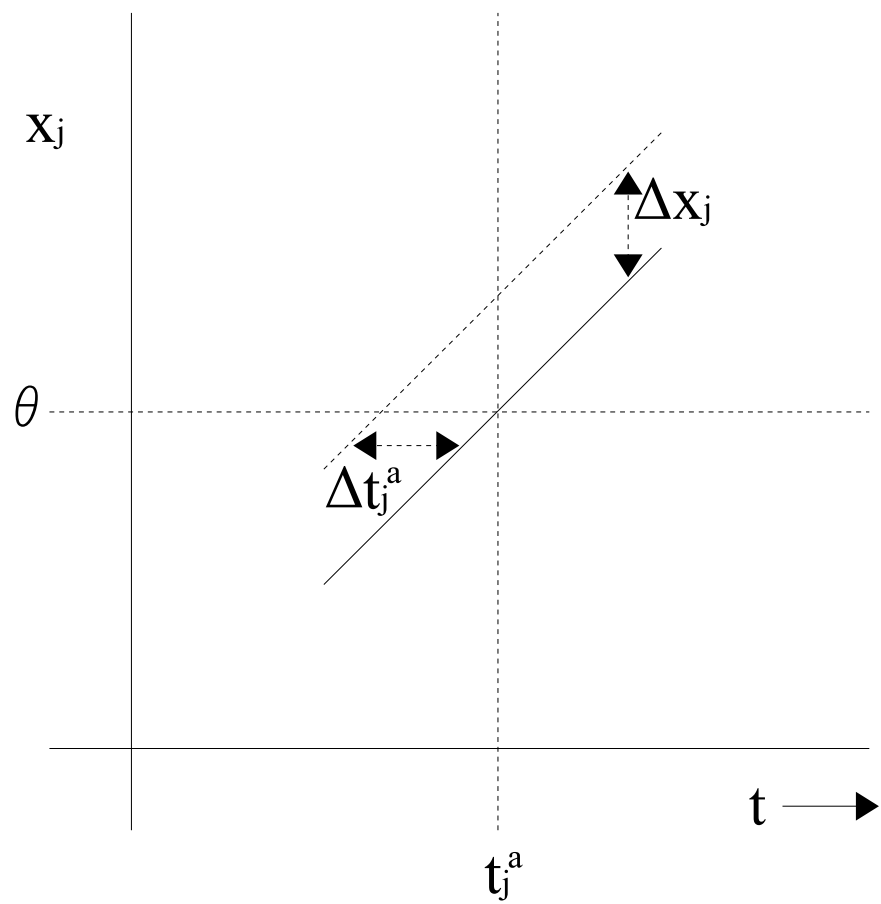


図 A.1: 内部電位がしきい値付近の区間における内部電位と発火時刻の関係

付録B XOR問題

ここでは、SpikeProp の学習対象とした XOR 問題について説明する。XOR 問題とは、線形分離不可能なパターン分類問題の中で、最も簡単な問題である。多くの難しい問題は XOR 問題を部分問題として内包するとされていることから、学習対象としてよく用いられる。

XOR 問題の入出力関係を表 B.1 に示す。シグモイド型ユニットを用いたニューラルネットワークであれば、これをそのまま学習対象として用いる。しかし、SpikeProp はユニットの入出力として、アナログ値ではなく、スパイクの発火タイミングを用いるため、時間版 XOR 問題 (表 B.2) を学習対象とした。

本文における実験結果の誤差 E は、時間版 XOR 問題の 4 つの入力パターンをそれぞれ入力したときの誤差の合計を記した。時間版 XOR 問題では、学習する (ネットワークのパラメータを調整する) ことにより、この誤差を小さくしていき、理想の入出力関係を実現する。

表 B.1: XOR 問題の入出力関係

入力 1	入力 2	出力
0	0	0
0	1	1
1	0	1
1	1	0

表 B.2: 時間版 XOR 問題の入出力関係

入力時刻 1	入力時刻 2	出力時刻
0	0	10
0	6	16
6	0	16
6	6	10

発表論文

- [1] 泰間智也, 高瀬治彦, 川中普晴, 鶴岡信治: 遺伝的アルゴリズムを用いた SpikeProp の確実性向上に関する一考察, 第 7 回情報学ワークショップ 2009(WiNF2009) 論文集, pp.85-88, 2009 (WiNF2009 奨励賞受賞)
- [2] 泰間智也, 高瀬治彦, 川中普晴, 鶴岡信治: SpikeProp の学習過程における誤差の振動を抑制する一手法, 平成 22 年度電気関係学会東海支部連合大会講演論文集, p.P3-6, 2010
- [3] 泰間智也, 高瀬治彦, 川中普晴, 鶴岡信治: SpikeProp の学習過程における誤差の振動と学習するパラメータとの関係に関する調査平成 22 年度三重地区計測制御研究講演会講演論文集, pp.A11-1-A11-2, 2010