

# 特別研究報告書

## 題 目

学習率を変動させる階層型ハードウェアニューラルネットワーク  
A Multi-Layered Hardware Neural Network with a Variable  
Learning Rate

---

## 指 導 教 員

矢野 政顕 教授

---

## 報 告 者

学籍番号: 1055082

氏名: 木村 知史

---

平成 14 年 1 月 27 日

高知工科大学 電子・光システム工学コ - ス

## 序文

この論文は、著者が高知工科大学大学院前期課程在学中に行ってきたニューラルネットワークのハードウェア化に関する研究成果についてまとめたものである。

ニューラルネットワークは、ニューロンという入力と出力を持つ最小の情報処理単位で構成されるネットワークである。ニューロンは神経細胞のことであり、神経細胞の集合体が人間の脳である。しかしニューラルネットワークの目的は脳の再現ではない。脳を構成するニューロンの数は多く、それらの結合は複雑であり、また脳の機能も複雑であるため、再現するのは困難である。ニューラルネットワークは、脳をあくまで手本とし、その知的機能を実現することで工学的応用を図るものである。したがってニューロンの振る舞いは実際よりも単純化される。単純化されたニューロンの基本的な動作は、入力値と結合の強さを表す重みの積和演算を行い、その演算結果を出力関数により変換して出力を決めることである。このような処理単位で構成されるネットワークの特徴が、並列性と学習機能である。ネットワークの学習は、ニューロン間の結合強度を正しい出力が得られるように徐々に更新して、結合強度を調整することでなされる。この学習機能に注目し、ニューラルネットワークは多方面に応用されている。ニューラルネットを応用する上で、ハードウェア化は必要である。本研究は、ハードウェア化されたニューラルネットワークの学習をより効率よく動作させるための手法を提案している。

本論文の内容は5つの章から構成されている。第1章では、研究の背景と、並列性と学習というニューラルネットワークの特徴、そしてハードウェア化の必要性について述べる。その後、ハードウェア化に関する研究の動向と本論文の位置付けを記している。第2章では、ニューロンのモデル化とその単純化された動作について、また本論文が対象としているネットワークの形態について述べる。続いて、そのネットワークの振る舞いと、一般的に使用される誤差逆伝播法という学習方法について説明する。第3章は、従来手法として具体的なハードウェア化の方法を説明し、そして問題点と改善点について述べるとともに、改善方法を提案する。第4章では、従来手法と、それを改善した提案手法をソフトウェアによるシミュレーションにより比較した結果について説明する。第5章では、設計ツールを用いて提案手法をVHDLにより記述し、シミュレーションを行った結果について説明する。第6章ではこれまでのまとめと今後の研究課題について述べる。

# 目次

## 序文

<b>第 1 章 はじめに</b> . . . . .	1
1.1 背景 . . . . .	1
1.2 ニューラルネットワークの特徴 . . . . .	1
1.3 ハードウェア化の必要性 . . . . .	3
1.4 アナログ回路とデジタル回路 . . . . .	4
1.5 現状と位置付け . . . . .	4
<b>第 2 章 ニュ - ロンのモデルと学習</b> . . . . .	8
2.1 ニューロンについて . . . . .	8
2.2 ニューロンの数理モデル . . . . .	10
2.2.1 MacCulloch と Pitts の数理モデル . . . . .	11
2.2.2 シグモイド関数 . . . . .	12
2.3 パーセプトロン . . . . .	13
2.3.1 パーセプトロンの構造 . . . . .	14
2.3.2 パーセプトロンの学習 . . . . .	17
2.3.3 排他的論理和 . . . . .	20
2.4 誤差伝播学習 . . . . .	21
2.4.1 勾配法について . . . . .	22
2.4.2 誤差伝播学習の概要 . . . . .	23
2.4.3 誤差伝播学習の導出 . . . . .	25
<b>第 3 章 ニューラルネットワークのハードウェア化</b> . . . . .	28

3.1 フォワードプロセス	28
3.2 バックワードプロセス	34
3.3 問題点と改善方法について	38
<b>第 4 章 ソフトウェアシミュレーション</b>	<b>41</b>
4.1 EXOR の学習実験	41
4.1.1 実験方法	41
4.1.2 実験結果と考察	42
4.2 文字の学習実験	48
4.2.1 実験方法	48
4.2.2 実験結果と考察	52
<b>第 5 章 VHDL による設計</b>	<b>55</b>
5.1 論理シミュレーションによる学習実験	55
<b>第 6 章 まとめ</b>	<b>62</b>
<b>参考文献</b>	<b>64</b>
<b>謝辞</b>	<b>66</b>

# 第 1 章

## はじめに

第 1 章では、まず研究の背景とニューラルネットワークの特徴について述べ、続いてハードウェア化の必要性と現状、及びこの研究の位置付けについて説明する。

### 1.1 背景

ニューラルネットワークの研究は、MacCulloch と Pitts が 1943 年にニューロンの数学モデルを提案したことから始まった[1]。そして 1961 年には、Rosenblatt によって、学習アルゴリズムの基礎であるパーセプトロンが発明された[2]。パーセプトロンは、入力と出力を規則的に与え、ニューロン間の結合重みを修正するという形でネットワークを学習させることを可能にしたものである。しかし、1969 年に Minsky と Papert によってパーセプトロンの限界が示された[3-4]。パーセプトロンは線形分離可能な問題に対しては有効であるが、逆に線形分離不可能な問題に対しては必ずしも有効でないということが証明された。それ以来、ニューラルネットワークの研究は下火になったが、1986 年に Rumelhart らによってバックプロパゲーション( back-propagation : 誤差逆伝播法 ) という新しい学習方法が提案された[5-6]。この方法によって、パーセプトロン型のネットワークにおいて線形分離不可能な問題に対しても学習が可能になった。その結果、これまで難しかった画像や音声の認識などにも応用されるようになり、ニューラルネットワークの可能性が大きく広がった。

### 1.2 ニューラルネットワークの特徴

ニューラルネットワークの特徴として、演算の並列性と学習機能がある。この 2 つの特徴について、現在のノイマン型コンピュータと脳の処理を比較しながら説明する。

#### (a) 演算の並列性

コンピュータと脳は構成と情報を処理する速度が違う。人間の脳は、約20億の神経細胞（ニューロン）からなる。また、それぞれのニューロンは約1万個の他のニューロンと結合している。しかし処理装置としてのニューロンは、1秒間の処理ステップはせいぜい数百のステップである。現在のコンピュータは中央処理装置（CPU）と記憶装置（メモリ）で構成され、CPUは1秒あたり数千万から億単位の処理ステップをこなせる。処理速度という面で比較すると、ニューロンとCPUとではその性能差は大きい。しかし人間は人の顔を瞬時に判断でき、また判別するだけでなく、微妙な表情までも読み取ることができる。それに対し、コンピュータでは顔の判別が難しい。また判別できたとしても表情まで読み取るようなことは困難である。このように、コンピュータと脳では処理速度に圧倒的な違いがあるにもかかわらず、脳がコンピュータよりも優れている部分がある。この理由は、コンピュータは基本的に逐次処理であり、CPUはメモリから受け取ったデータを順に処理するのに対して、脳は各ニューロンが独立して動作する、つまり並列処理をしているという点にある。したがってニューラルネットワークは、高い並列性を持つため高度な機能を実現できる可能性がある。

#### （b）学習機能

脳は外部の環境変化に適応できる。その例として言語の獲得が挙げられる。日本に生まれ、日本で育った子供が成長とともに日本語をしゃべるようになるのはごく自然のことである。それは、その子供が日本人夫婦の子供であり、日本人としての脳をもって生まれたからではない。その子供の周囲に日本語を使う人が多いからである。その子供は単語を発音し、もし間違いであれば周囲の人間（親）が教師としてその発音を訂正する。そして単純な文章を使い始め、同様にその文章が間違いであれば修正する。これを繰り返して、ついには日本語という複雑な言語体系を獲得するにいたる。それは、日本語を話す周囲の人間との会話という具体的な経験を繰り返すことによって学習したといえる。またそのことは、訓練によって脳が環境に適応したことを意味する。このとき脳の内部では、学習の過程においてニューロン間の結合が強まったり弱まったりすることで、間違いを少しずつ訂正していく。最終的に教師による修正が不要なくなったとき、ニューロン間の結合強度の度合いは正解の出力が得られるように調節された状態にある。

より具体的な例として、コンピュータに音読させるために、コンピュータプログラムによって文字と音を変換する規則体系を構築した例と、ニューラルネットワークに学習させることで音読させた例がある[7]。

視覚障害者が印刷物を利用できるように、英語で書かれた文章を発音するよ

うなプログラムが開発された。英語は、同じ文字でも場合によって発音が違うことが起こる。例えば「ADAM」の最初の A と 3 文字目の A では発音が異なる。文字を音へ変換するには、その文字の周辺を考慮しなければならず、プログラムは複雑なものとなった。そして速い計算機の導入によって期待する機能を実現し、コンピュータに本を音読させた。その後、同じ機能を実現するためにニューラルネットワークが導入された。文字の読み取りには同じスキャナーを用い、同じ音声合成装置が使用された。文字の入力と対応する出力を用いて、ネットワークにおけるニューロンの結合強度を自動的に調節することで学習させた。その結果、文字の音読は高い正解率を達成した。

この例から言えることは、問題によってはその解法が複雑であり、プログラムとして記述することが困難な場合があるが、そのようなとき、ニューラルネットワークが有効であるという可能性を示しているということである。なぜならニューラルネットワークは、入力パターンとそれに対応する出力を与える作業を繰り返すだけだからである。ニューラルネットワークには複雑な解法を記述する必要がないためである。

### 1.3 ハードウェア化の必要性

ニューラルネットの特徴を生かした応用例は多数存在する。例えば、特殊なカメラを用いた連続画像を処理する装置がある[8]。その装置は、人間の視覚システムを取り入れたもので、手のジェスチャーをカメラで取り込み、形状を認識し、学習済みの特徴と照らし合わせて判断する。また画像中の人間の数を数えるために、人間の顔を検出するシステムもある。このシステムは人間の肌の特徴として熱に注目し、顔と認識できるようなパターンを生成し、学習させることでパターン認識を行っている。他にも、手書き文字の文字認識や音声認識にも応用されている。手書き文字認識では形の歪みや変形、音声ではアクセントやイントネーションといったものを含む情報処理を行っている。

これらのようなニューラルネットを応用したシステムを実現するには、ソフトウェアによるものとハードウェアによるものの2つが考えられる。ソフトウェアによる実現は、ネットワークの構造が自由であり、変更もハードウェアに比べて容易である。しかし大規模化すると処理時間が長くなる。また小型化が難しい。それに対し、ハードウェアによる実現は、ネットワークの構造を変更しにくいのが、大規模化しても処理時間が長くないし小型化もしやすい。またネットワークを構成するニューロンの数が多いほど、ニューラルネットの実

現できる機能は高くなる。よってニューラルネットをシステムの一部として組み込む場合、ハードウェアでの実現が望ましい。ただし、ネットワークの構造は変更しにくい、またニューロンの数は増やしにくいので、使用に制限がでる可能性はある。

## 1.4 アナログ回路とデジタル回路

ニューラルネットワークをハードウェア化するために、アナログ回路かデジタル回路かという問題がある。アナログ回路での実現は、ニューロンの動作である入力値とニューロン間の結合強度の積和演算がデジタル回路に比べて容易である。オペアンプを用いた加算回路であれば、入力電圧と抵抗の積が加算されるとき、抵抗が入力電圧に重みをつける役割をするためである。またこのようなニューロンを MOS トランジスタで構成し、LSI 化した例や、入力信号として直流電圧ではなく交流電圧を用いて実現したものなどがある。しかし、アナログ回路での実現は、素子のばらつきが問題となったり、設計が難しいという面がある。また集積化する場合でも、上記の理由から規模に限界がある。

一方、デジタル回路での実現は、アナログ回路に比べて配線が多く複雑であるが、設計が容易であり集積化しやすい。ある機能を実現するニューラルネットワークを考えると、ニューロンの数が少なくて困ることはあっても、多すぎて困ることはない。ニューロンの数が多いほどより高い機能を実現できる。設計の容易さと集積化しやすいという点でデジタル回路での実現が望まれる。よって本研究ではニューラルネットワークをデジタル回路で実現する。

## 1.5 現状と位置付け

デジタル回路でニューラルネットワークを実現するには、単純にニューロンへの入力信号とニューロン間の結合重みを乗算するための乗算器と、その乗算結果の総和をとるための加算器があればよい。しかし、結合数だけ乗算器を用意すれば、ニューロン数が増加とともに結合数やそれにとまなう配線が増加し、結局 LSI の面積増加となるため、ネットワークの規模を大きくできない。そのため、ニューラルネットワークのハードウェア化においては、データを並列ではなく直列で扱うことによって配線の数を抑え、また乗算器を用いない方法を採用する必要がある。

現在では、直列データを乗算器なしに演算する方法[9-10]と、データをパルス



で表現し、乗算器を不要にする方法[11-15]が用いられている。

(1) 直列データを用いる方法

この方法では、ニューロンの出力値として、ニューロンの出力を連続的な値をとらずに、閾値を用いて離散値にする。その結果、その出力値と結合重みの乗算は単純になり、乗算器を用いることなく実現できる。乗算結果は直列データとして後段のニューロンの入力となる。

(2) ニューロンの出力をパルスで実現する方法

(i) パルス幅変調方式

パルス幅変調方式[11-12]は、ニューロンの出力の大きさに応じてパルス幅を変化させる。出力が大きければパルスの幅を広く、逆に出力が小さければパルス幅を狭くする。パルス幅は、減衰信号によって決定される。ニューロンの出力は“ 1 ”か“ 0 ”であるため、この出力値と結合重みの乗算は論理積で実現できる。その積は加算器で加算することで加重和を計算する。ニューロンの出力は加算結果を閾値と比較して決定する。このときの加算結果を時定数に応じて減衰させる。加算結果が大きければ、出力が“ 1 ”である時間が長くなり、加算結果が小さければ出力が“ 0 ”となる時間が長い。

(ii) パルス密度変調方式

パルス密度変調方式は、基準パルスからの入力パルスの遅延時間を用いる方法[13]と、データの大きさに対応したパルス密度で表現する方法[14-15]がある。

基準パルスからの遅延を用いる方法は、基準パルスからの遅延が正規分布に従う乱数となるような入力パルス列を与える。入力パルスと結合重みの乗算は、基準パルスからの入力パルスの遅延の大きさに応じて乱数を発生させ、結合重みと比較することで実現する。乱数を発生させるために、LFSR (線形フィードバックシフトレジスタ) を用いる。この回路は、フリップフロップと EXOR で構成される回路である。

他にパルス密度を用いる方法として、データの大きさをパルス密度に変換し、レートマルチプライヤを用いて乗算する方法[14]と、入力信号と結合重みをパルスの出現確率で計算する方法[15]がある。まず、レートマルチプライヤを用いる方法について説明する。入力データはある周波数の方形波であり、これをクロックとするカウンタがレートマルチプライヤである。レートマルチプライヤにはさらに結合重みが入力されている。レートマルチプライヤは、クロックとしてのデータが入力されるたびに状態遷移する。このとき、入力されている結合重みは、レートマルチプライヤの各状態で“ 1 ”を出力するかどうかを決める。

結合重みの値が小さければ、レートマルチプライヤが“1”を出力する状態が少ない。よって状態遷移するレートマルチプライヤから出力されるパルス密度は小さい。逆に結合重みが大きければ、レートマルチプライヤは“1”を出力する状態が多くなることによってパルス密度が高くなる。このような方法で乗算を実現する。この各乗算結果はパルス列であり、この論理和をカウンタに入力することで和をとっている。このように積和演算を実現している。

ニューロンへの入力信号と結合重みをパルスの出現確率（パルス列の中でビットが“1”になる確立）で表現する方法は、積和演算をパルス列の論理積と論理和で実現する。レートマルチプライヤのようなカウンタを用いないため、回路を小型化できる。

パルスを用いて演算する方法[11-15]では、シナプスやニューロンが非同期的な回路になりやすく、設計が容易でない。また、現在良く用いられている学習方法であるバックプロパゲーションアルゴリズムを実装することが難しい。バックプロパゲーションには乗算の繰り返しや微分演算が含まれるからである。パルス幅変調方式は、この学習方法を実装するために、カウンタやシフト演算を多用しなくてはならない。パルス密度変調方式は、結合重みの値が-1から+1までしかとれず、十分な学習ができないという問題がある。基準パルスとの遅延と確率的演算を組み合わせる手法では、計算精度を上げるためにLFSRを多用しなくてはならない。またニューロン間はパルスの送受信であっても、ニューロン内部での演算はパラレルデータの演算であり、小型化には不向きである。

直列データを用いる方法[9-10]では、シナプスやニューロンが同期式回路で実現が可能であり、またニューロンの出力を離散化しているため乗算が容易である。バックプロパゲーションアルゴリズムの実装においても、直列データを用いる方法はパルス方式より実現しやすい。

本研究では、ニューラルネットワークのハードウェア化において、パルス変調方式ではなく、シリアルデータを用いた離散値ニューロンの方式を採用した。非同期的な回路になりやすいパルス変調式よりも、同期式回路で構成しやすいシリアルデータを用いた離散値ニューロンの方式の方が容易に設計できるからである。離散値ニューロンは、結合重みをカウンタで実現しているが、計算精度や学習の速度向上には、この結合重みのビット数を増加させることになる。しかしそうすると学習時間の増加を招く。本研究では、離散値ニューロンの学習性能の改善を主眼においている。

これまで説明したように、この章では、まず研究の背景としてニューラルネットワークの歴史を紹介した。次に、ニューロンが同時に動作することで、ニューロンの数が増えても処理時間が変わらないという高い並列性をもつことと、ニューロン間の結合の強度を調節することによって、画像認識やパターン認識などの複雑な情報処理を可能とするという 2 つのニューラルネットワークの特徴と、その応用に向けてのハードウェア化の必要性について説明した。最後にハードウェア化についての研究動向と本研究の位置付けについて述べた。

## 第 2 章

### ニューロンのモデルと学習

この章では、生物的なニューロンについての簡単な説明と、それを MacCulloch・Pitts モデルという代表的な数理モデルで表現したものについて説明する。さらに、それを組み合わせたネットワークであるパーセプトロンについて、その動作と学習について述べた後、Minsky と Papert によるパーセプトロンの問題の指摘と、Rumelhart らによる、それを解決するバックプロパゲーションアルゴリズム（誤差逆伝播法）について説明する。

#### 2.1 ニューロンについて

細胞は、熱や特定の化学物質の存在や接触によって細胞膜のイオンに対する透過性を変化させ、膜電位の変化を生じさせて興奮状態となる。その膜電位の変化は周囲の細胞に伝播される。神経細胞（ニューロン）は、興奮能力と伝播能力において特に優れた細胞である。

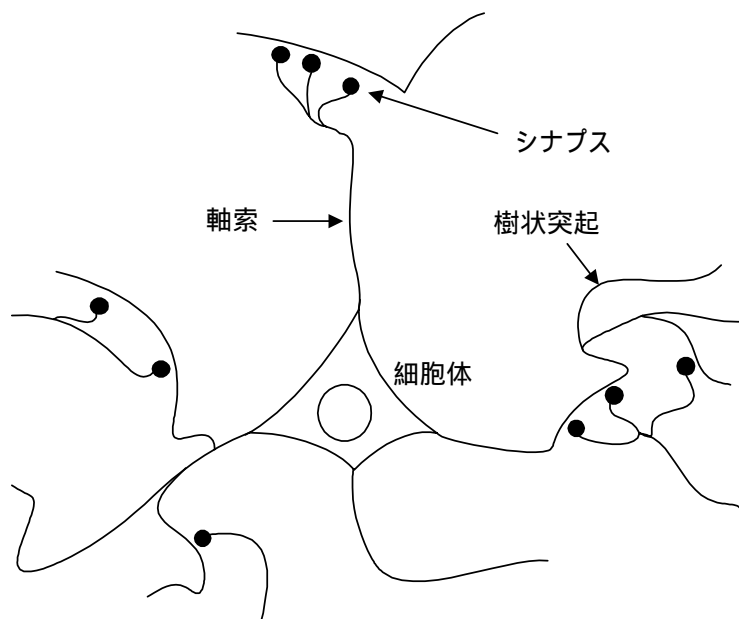


図 2.1 ニューロン

図 2.1 に示すように、ニューロンは主に 3 つの部分からなる。細胞本体となる細胞体と、複雑に枝分かれした樹状突起、そして細胞体から 1 つだけ長く伸びて末端で枝分かれした軸索と呼ばれる部分である。軸索は細胞体からの信号を他のニューロンに伝えるためのもので、いわばニューロンの出力用繊維である。樹状突起は、他のニューロンからの信号を受け取る部分であり、他のニューロンの軸索の末端と結合する。この結合部分をシナプスと呼ぶ。この結合部分において、軸索と樹状突起と直接結合しているわけではない。シナプスでは、軸索の末端において化学物質を放出することによって情報伝達を行う。1 つのニューロンで数千本から数万本の軸索から信号を受け取る。したがって多数の軸索と結合するには、樹状突起のような枝をつくって軸索との結合箇所を広げる必要がある。細胞体は、その多数の入力信号をもとになんらかの計算を行い、計算結果は軸索を伝わって他のニューロンへ伝達する。

ニューロンの興奮と情報の伝達は、イオンの移動に基づく。そのイオンは特にナトリウムイオンとカリウムイオンである。すべての細胞はナトリウムを細胞外へ出し、カリウムを細胞中へ入れることによって、わずかな電位（約  $70 \mu\text{V}$ ）を生じる。この電位は静止膜電位と呼ばれる。細胞はある特定の状況においてその電位が不安定になる仕組みとなっている。その静止膜電位が乱されると、電位は急激に増大したり減少したりする。ニューロンは、このような電位の変化を情報の処理と伝達に利用するよう専門化されている。

静止膜電位に対して、活動電位と呼ばれる電位がある。例えば細胞がある状況でイオンを移動させ、静止膜電位が減少しはじめて、その電位が  $20 \mu\text{V}$  以上減少すると、膜電位は急速に減少し、さらに静止膜電位のプラスマイナスが逆転する。そのあと  $1 \text{ms}$  から  $10 \text{ms}$  後には再び静止膜電位に戻る。結果として細胞膜の内側と外側の電位差が時間的に変化することによってパルス状の電位変化を生じる。このような爆発的な現象が活動電位である。活動電位が発生するためには、決まった電位（閾値）を下回らなくてはならない。ニューロンが興奮状態にある、または発火するという状態は、静止膜電位が閾値に近づき、閾値に到達したニューロン自らが活動電位を発生させた状態である。発生した膜電位は、軸索にそって急速に伝播し、軸索はさらに別のニューロンへと活動電位を伝える。

軸索を伝播した信号は、シナプスという軸索の末端に到達する。そこには化学的な連絡場所があり、電気的な刺激は、アセチルコリンやカテコールアミンといった神経伝達物質とよばれる化学物質へと変換される。軸索の末端にはシナプス小胞というこれらの伝達物質を含む塊がいくつかある。軸索を伝播されてきた活動電位は、このシナプス小胞を刺激し、伝達物質を細胞膜外へと放出

させる。信号を受け取る側となる細胞は、樹状突起でこの伝達物質を受け取り、電位へ変換して細胞体へと伝達する。このとき重要な点は、シナプスにおいて情報伝達が行われるとき、信号伝達の強さがシナプスによって異なることである。ニューロンにとって、軸索を通じて送られてくる信号は入力信号である。この入力信号はシナプスを介して伝達されるとき、その結合強度によって重みづけされているといえる。

これまで述べたように、ニューロンは細胞体・軸索・樹状突起の3部分からなり、信号の処理と伝達に特化した細胞である。細胞体はイオンの濃度の違いを活動電位として電位差を生じ、その電位は軸索を伝わる。シナプスまで到達した電位は伝達物質を放出するための刺激となり、放出された伝達物質は樹状突起で検知され、電位となって伝達される。シナプスでは結合の強さによって伝達強度が異なる。ニューロンは情報の伝達と処理において、複雑な電氣的な伝播と化学的な伝播を駆使している。

## 2.2 ニューロンの数理モデル

実際のニューロンの構造や情報伝達の仕組みは複雑である。脳はそのようなニューロンの集合体であり、その接続も複雑なものである。ニューラルネットワークは、脳をモデルにしたものではあるが、実際のニューロンを完全に模倣することを目的としていない。目的はニューロンの情報伝達の特徴を抽出し、それを数学的に表現して、それらを用いたネットワークを工学的に応用することである。

ニューロンモデルは様々であるが構造は同じである。ニューロンが、多数の入力信号をひとつの出力信号に変換する単位制御素子であると仮定する。するとニューロンの機能は、入力の強度とシナプス加重と活性化関数によって表現できる。活性化関数は、ニューロンが発火するかどうかを決定するための関数である。図 2.2 にニューロンの構造を示す。図 2.2 の  $x_1, x_2, \dots, x_n$  はニューロンへの入力信号である。 $w_1, x_2, \dots, w_n$  はシナプス結合重みである。この結合重みによって刺激が促進されたり抑制されたりする。入力と結合重みで決まる値を  $z$  で表し、その関数値と活性化関数  $f$  によって出力  $y$  が決まる。出力  $y$  はさらに他のニューロンへの入力となる。

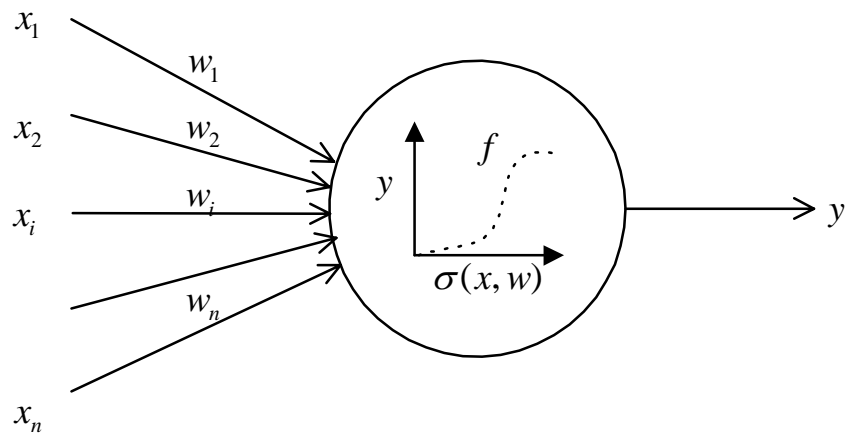


図 2.2 ニューロンモデル

### 2.2.1 MacCulloch と Pitts の数理モデル

MacCulloch と Pitts は、ニューロンが論理的な制御素子であると仮定した。そのモデルでは「and」「or」「not」といった基本的な論理演算が可能であり、さらに複雑な結合子を構築できることを示した。

MacCulloch と Pitts の提案したニューラルネットワークモデルにおいて、ニューロンへの入力、そのニューロンに接続する他のニューロンからの出力に結合重みを乗じたものの総和で表される。他のニューロンからの出力を  $x$ 、結合重みを  $w$  とし、ニューロンへの入力を  $\sigma$  とすると、式 (2.1) となる。

$$\sigma = \sum_{i=1}^n x_i \cdot w_i \quad (2.1)$$

さらにこのモデルでは、重み付けされた入力信号の総和である  $\sigma$  と、活性化関数  $f$  を適用してニューロンの出力を求める。関数  $f$  は図 2.3 のような階段状をしたステップ関数である。この関数は閾値を持っており、その閾値をこえると“ 1 ”、そうでなければ“ 0 ”という値を返す。式で表すと式 (2.2) のようになる。式 (2.2) における閾値は 0 としている。関数  $f$  はマッカロック・ピッツ関数と呼ばれる。

$$y = f(\sigma) = \begin{cases} 1, & \sigma > 0 \\ 0, & \sigma \leq 0 \end{cases} \quad (2.2)$$

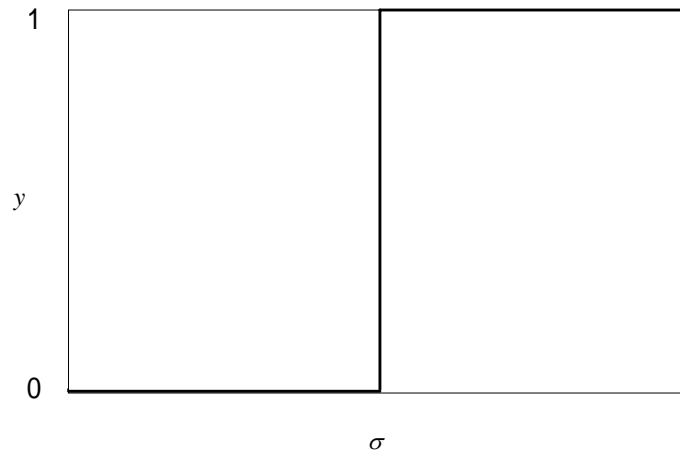


図 2.3 マッカロック・ピッツ関数

### 2.2.2 シグモイド関数

活性化関数にステップ関数を使うかわりに、図 2.4 のような S 字型をしたシグモイド関数がよく用いられる。ニューラルネットワークの学習においてよく使われる誤差伝播学習は微分可能な関数が望ましいためである。ステップ関数は微分不可能である。図 2.4 を式で表すと、式 (2.3) のようになる。また の値は、式 (2.1) で求められる。

$$y = f(\sigma) = \frac{1}{1 + \exp^{-T\sigma}} \quad (2.3)$$



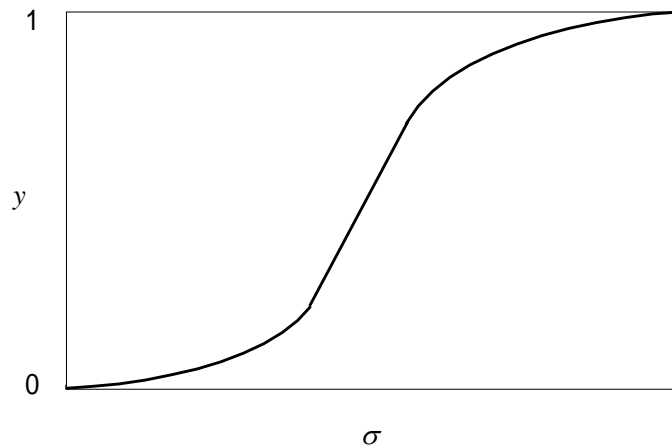


図 2.4 シグモイド関数

また、シグモイド関数は式 (2.4) のように  $\tan$  の逆関数を用いて表されることもある。

$$y = f(\sigma) = \frac{1}{2} \{ \tan^{-1}(T\sigma) + 1 \} \quad (2.4)$$

式 (2.3) と式 (2.4) のパラメータ  $T$  を変えれば、線形関数やステップ関数で近似できる。

これまではニューロンの数理モデルについて述べた。ニューロンは論理関数を表現する素子として仮定されている。そのため活性化関数として階段状をしたステップ関数が用いられる。活性化関数は、入力とシナプス結合重みの乗算と、その総和を閾値と比較する。その和が閾値より大きければニューロンが発火、小さければ何も起こらない。現在では、活性化関数として、微分可能なシグモイド関数が用いられることが多い。

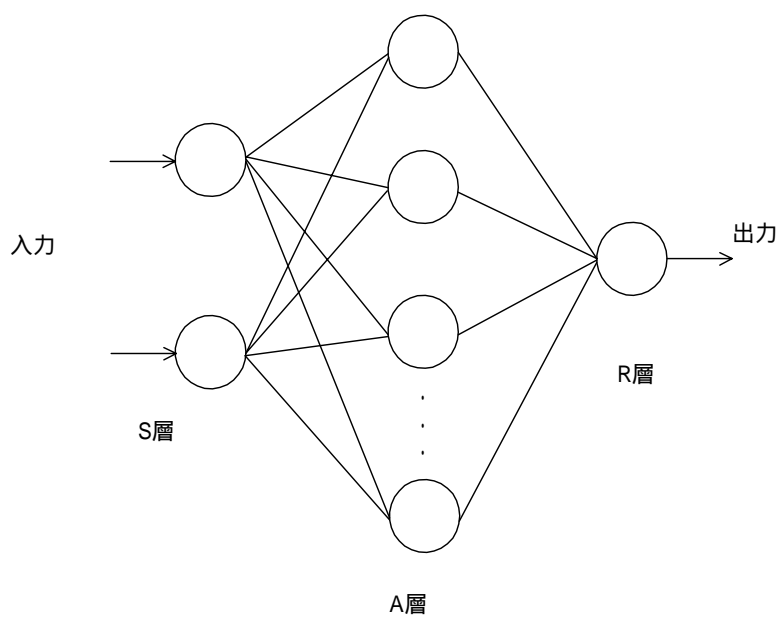
## 2.3 パーセプトロン

パーセプトロンは、Rosenblatt によって発明され、学習機械の基礎となったネットワークであり、それは MacCulloch・Pitts のモデルを採用したニューロンで構成される。ネットワークの構造と、その学習方法を述べ、そしてパーセプトロンの持つ問題について説明する。学習とは、ネットワークへの入力信号

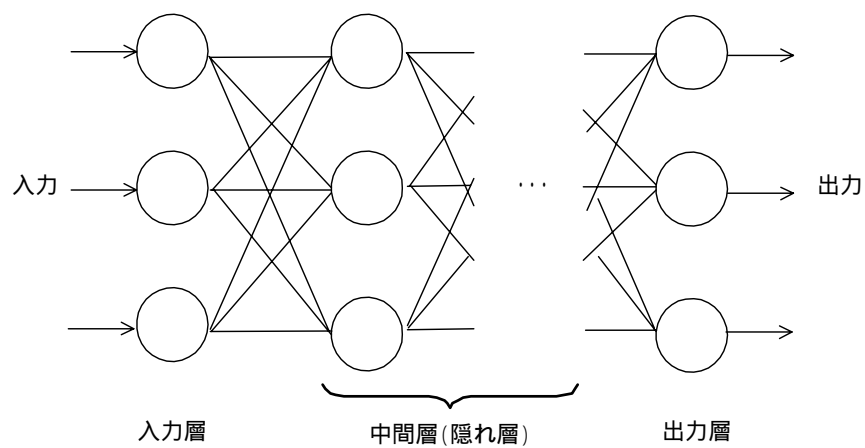
に対して、出力させたい値が出るようにシナプス結合重みを調整することである。

### 2.3.1 パーセプトロンの構造

パーセプトロンの構造を図 2.5 (a) に示す。



(a) パーセプトロン



(b) 階層型ネットワーク

図 2.5 ネットワークの形態

パーセプトロンは3層からなるネットワークである。その層はS層(Sensory unit)、A層(Associative unit)、およびR層(Response unit)と呼ばれる。ネットワークの入力から出力にいたるまで、信号の流れは単一方向である。また各層のニューロン数は制限されない。S層は入力信号をそのまま通過させるニューロンである。A層とR層のニューロンは活性化関数としてステップ関数を持つMacCulloch・Pittsモデルである。このようにフィードバックや相互結合を持たないようなネットワークは一般的に階層構造ネットワークと呼ばれる。したがって、パーセプトロンは図2.5(b)のような階層構造ネットワークの一つである。階層型は、入力層と中間層、そして出力層で構成される。各層のニューロン数に制限はなく、中間層の層数も制限は無い。

信号の流れを説明する。まず入力信号はS層を通過し、S層からの信号をA層に伝達するが、このときS層とA層の結合に重みがかかっている。A層のニューロンでは、S層の出力とこの重みの加重和がとられ、その和と閾値を比較することで出力を得る。こうして得たA層の出力がR層の入力となるが、R層への入力、A層とR層の結合重みの加重和と閾値の比較でR層のニューロンの出力が決定され、その値がネットワークの出力値となる。図2.6と図2.7にネットワークの計算例を示す。

図2.6はS層からA層の計算、図2.7はA層からR層の計算を示している。ネットワークはS層、A層、R層はそれぞれ3つのニューロンで構成されている。またA層とR層の活性化関数の閾値は0であるとする。今S層(S1、S2、S3)に{1 0 1}という入力を与えたとする。S層のニューロンは入力値をそのまま出力する。この値がA層(A1、A2、A3)の入力となる。図2.6はA1のみの計算を示している。S層とA1の結合重みが{-3.7 0.7 4.1}である。その結果、加重和は

$$\begin{aligned}\sigma &= \{1 \times (-3.7)\} + \{0 \times 0.7\} + \{1 \times 4.1\} \\ &= 0.4\end{aligned}$$

となる。は閾値を上回っているので、A1の出力は1となる。A2とA3の出力も同様に求められる。

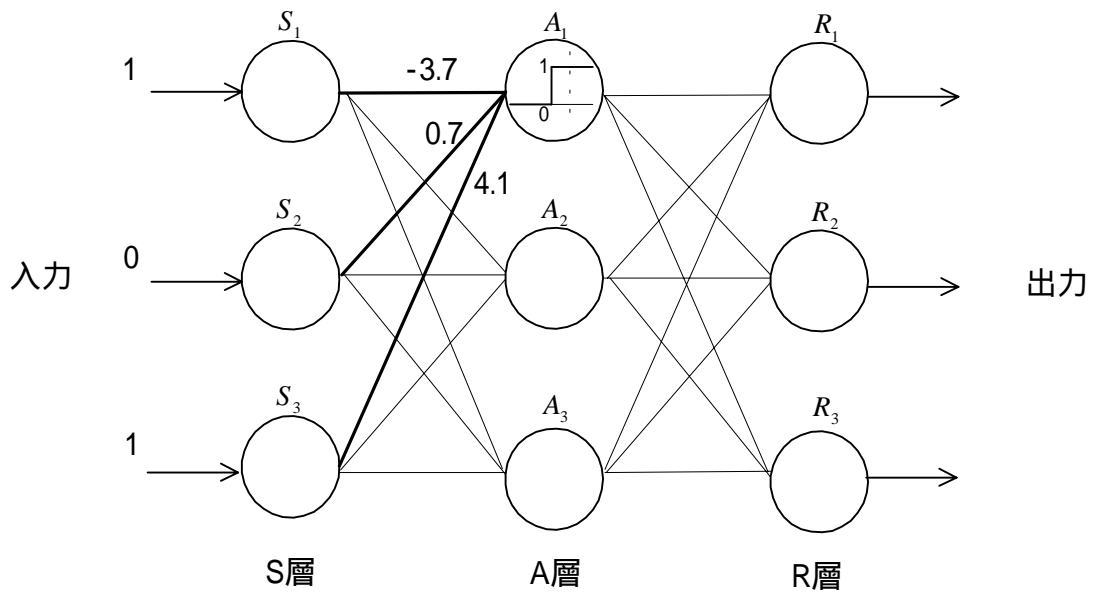


図 2.6 S層からA層の計算

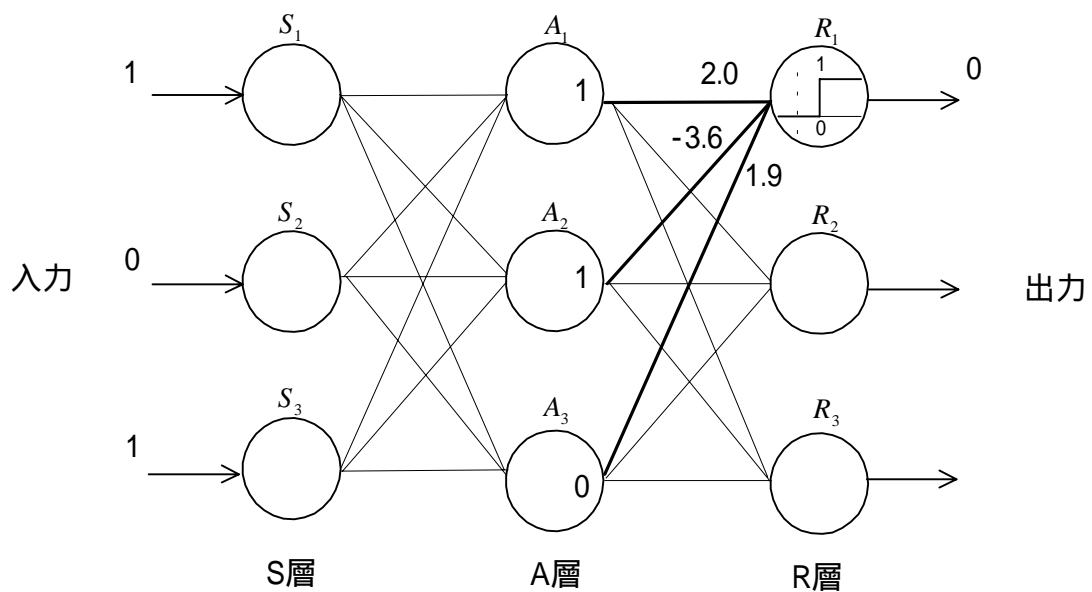


図 2.7 A層からR層の計算

図 2.7 は R 層にあるニューロン R1 についての計算例を示している。A1 を求めたものと同様に A2、A3 を求め、その結果 A 層 (A1、A2、A3) の出力がそれぞれ { 1 1 0 } となったと仮定する。この値が R 層の入力となり、A 層と R 層の結合重みの加重和と閾値で R 層の出力を決定する。図 2.7 において、A 層と R1 の結合重みが { 2.0 -3.6 1.9 } であるとする、加重和は

$$\begin{aligned}\sigma &= \{1 \times 2.0\} + \{1 \times (-3.6)\} + \{0 \times 1.9\} \\ &= -1.6\end{aligned}$$

となり、閾値を下回るのでニューロン R1 の出力は 0 となる。R2 と R3 も同様にして求められる。

### 2.3.2 パーセプトロンの学習

パーセプトロンの出力から、必ずしも期待する値が得られるとは限らない。このため、期待する出力が得られるようにシナプス結合重みを変更しなくてはならない。変更するためには、結合重みを増減させる量を決定しなくてはならない。図 2.8 に パーセプトロンの結合重みの更新手順を示した。

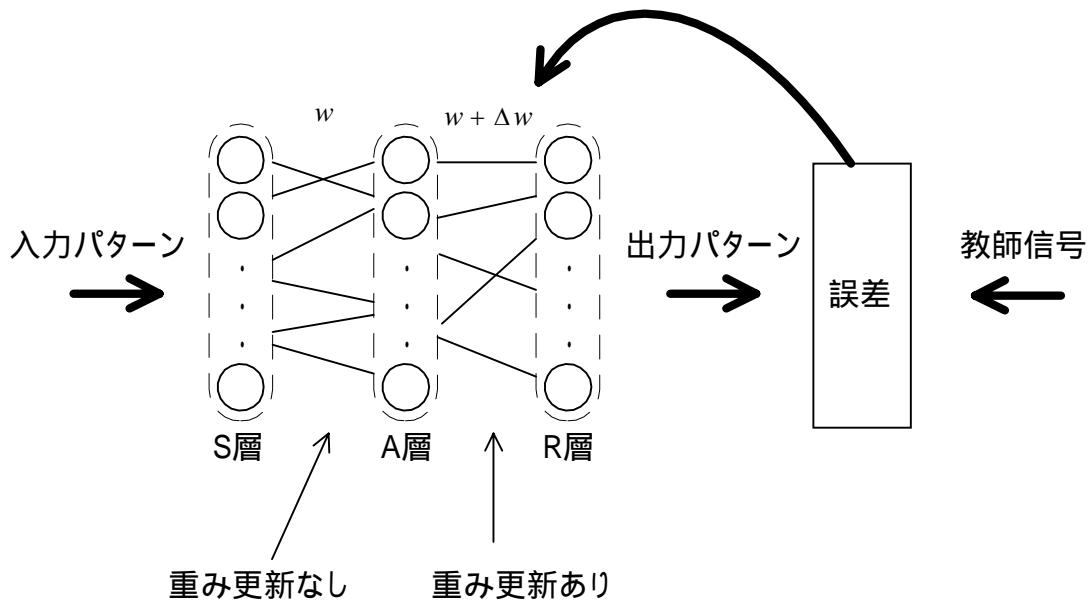


図 2.8 パーセプトロンのシナプス結合重み更新手順

パーセプトロンは、ニューロンの出力値と得たい出力（目標値）の誤差を誤差信号とし、それをを用いてその変更量を定める。このとき目標値は、誤った出力を出すニューロンに正解を教える信号であるから「教師信号」と呼ばれる。また、パーセプトロンにおける重み更新は、A層とR層を結合するシナプスの重みだけである。S層とA層の結合重みは固定したまま更新されない。

重みの変更と変更量は、式(2.5)、(2.6)のように表される。

$$w_i = w_i + \Delta w_i \quad (2.5)$$

$$\Delta w_i = \rho(d - y)x_i \quad (2.6)$$

ここで、 $w_i$  は  $i$  番目の結合重みを表し、 $\Delta w_i$  は重みの変更量を表す。また  $\rho$  は学習効率、 $d$  は教師信号であり、 $y$  はニューロンの出力である。さらに、 $x_i$  は  $i$  番目のニューロンの入力である。式 (2.6) の  $(d - y)$  は教師信号と出力の誤差信号であり、重みの更新量  $\Delta w_i$  は、出力誤差と入力信号を反映したものである。

学習は、ネットワークへの入力信号に対応する出力値と、その出力に対する教師信号との誤差を利用して重みを更新するが、そのとき 1 回の入力パターンの提示で目標値が得られるようになるわけではない。入力信号と教師信号を繰り返し与えることで徐々に目標出力に近づけていく。このような学習は、重みの更新が無くなるまで行われる。全ての入力パターンに対して出力誤差が無くなったとき、学習は完了する。

学習効率  $\rho$  は、1 回の学習ステップでどれくらい重みを更新するかを決定する。 $\rho$  が大きければ学習の終了を早めることができるが、大きすぎると、入力を与えるたびに重みが大きく変更され、逆に学習されにくい。学習効率は適度な値に設定しなくてはならない。

パーセプトロンの論理積の学習例を図 2.9 と図 2.10 に示す。これら 2 つの図は  $S$  層への入力と目標値の関係を表す真理値表と、ネットワークの構成を示している。ネットワークは、 $S$  層が 2 つ ( $S_1, S_2$ )、 $A$  層が 3 つ ( $A_1, A_2, A_3$ ) および  $R$  層が 1 つ ( $R_1$ ) のニューロンで構成されている。ニューロンは MacCulloch・Pitts のモデルに従い、活性化関数は式 (2.2) のような閾値を 0 とするステップ関数である。また学習効率  $\rho$  を 1 とする。

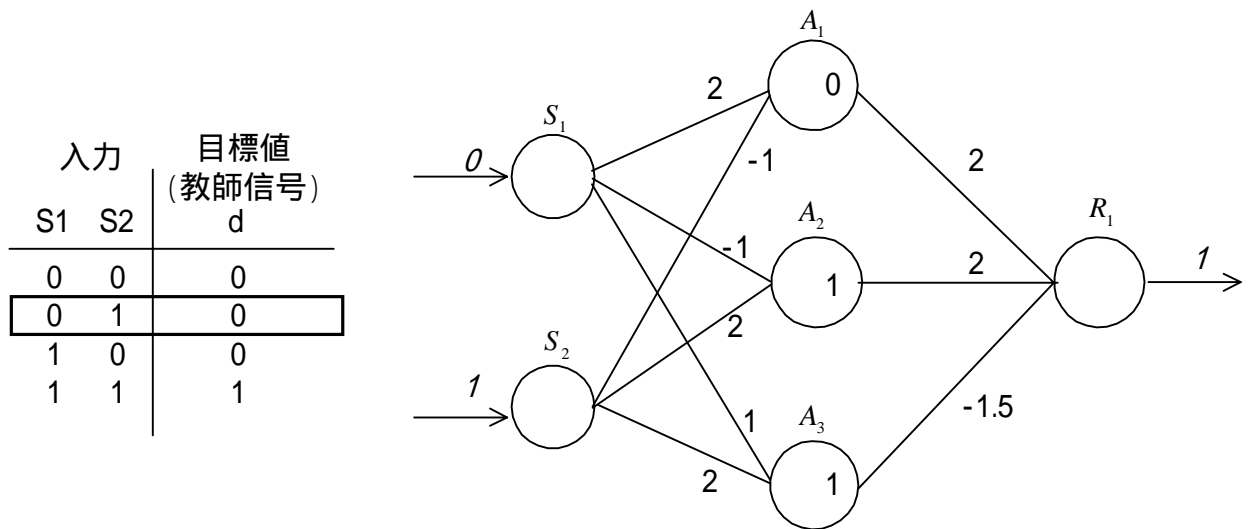


図 2.9 パーセプトロンによる論理積の学習 ( 1 )

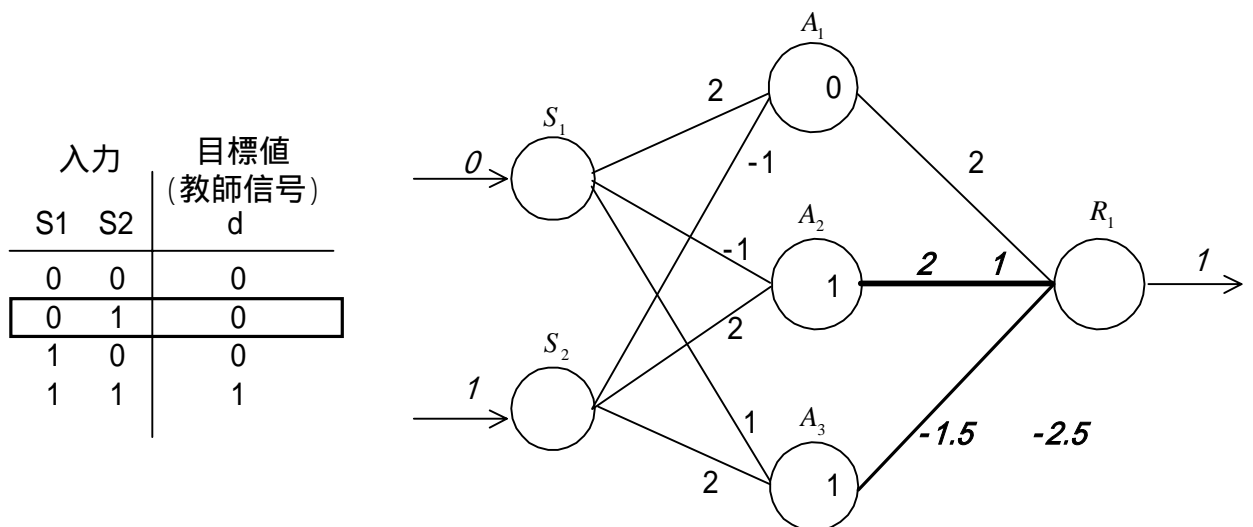


図 2.10 パーセプトロンによる論理積の学習 ( 2 )

図 2.9 において、まず S 層に { 0 0 } を入力したとする。このときの R1 の出力は 0 である。このとき、教師信号と出力信号の誤差は 0 であるため、学習は行われぬ。次に { 0 1 } を入力すると、R1 の出力は 1 となり、これは望まれる出力値ではない。そのため結合重みを更新しなくてはならない。図 2.10 に変更後の結合重みを示している。結合重みを更新するために式 ( 2.6 ) に従って更

新量を求める。A1 と R1 の結合重みは、A1 の出力値が 0 であるため、 $w=0$  となるため変更されない。更新されるのは A2 と R1、A3 と R1 の結合重みである。更新量はそれぞれ-1 であるので、A2 と R1 の結合重みは 2 から 1 へ、A3 と R1 の重みは-1.5 から-2.5 に変更される。あとの入力信号  $\{1\ 0\}$   $\{1\ 1\}$  については出力誤差がないので、これ以上結合重みを変更する必要はない。よって学習は終了である。

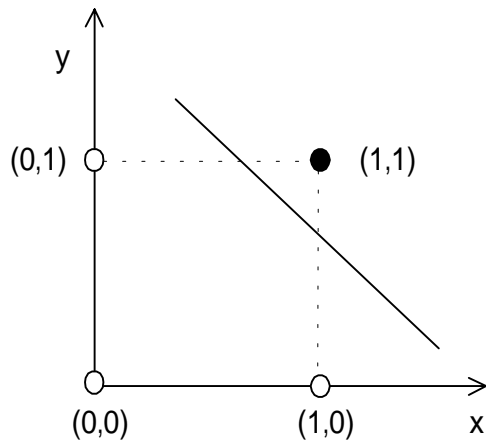
### 2.3.3 排他的論理和

パーセプトロンは、A 層と R 層のシナプス結合重みを変更することで学習を可能にした。しかし、Minsky と Papert によってパーセプトロンの限界が示された。それは A 層の出力が線形分離可能な条件を満たす場合ならば、有限回数の学習でパターンを分類できるが、逆に線形分離不可能ならば分類できるとは限らないということである。分類できる状態とは、入力パターンに対する出力を正しく得られる状態であり、学習とはその正しい出力が得られるまで結合重みを更新していくことである。図 2.9 および図 2.10 で示したような論理積 (AND) の学習は、線形分離可能な例である。図 2.11 を用いて説明する。

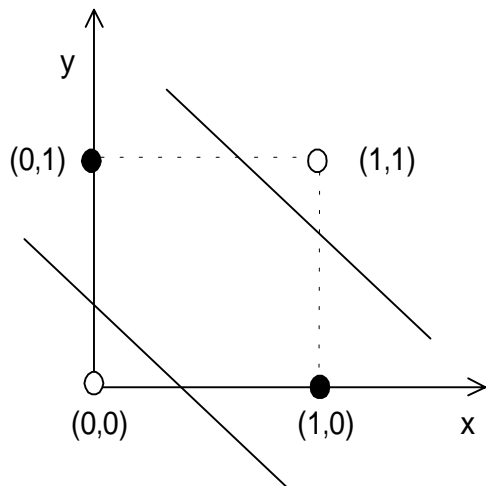
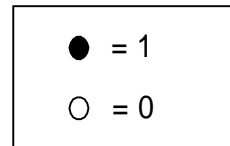
図 2.11 (a) は AND の線形分離を示している。図の座標は入力パターンであり、出力パターンは  $\begin{matrix} 0 \\ 1 \end{matrix}$  で表している。AND の場合、 $\begin{matrix} 0 \\ 1 \end{matrix}$  が 1 本の直線で分類できる。図 2.11 (b) は排他的論理和 (EXOR) の場合を示している。同様に入力パターンは座標、出力パターンは  $\begin{matrix} 0 \\ 1 \end{matrix}$  で表している。この場合では、1 本の直線で  $\begin{matrix} 0 \\ 1 \end{matrix}$  を区別することはできない。分類するには最低でも 2 本の直線が必要である。このような 1 本の直線、あるいは平面で分類できる信号の集合を線形分離可能であるという。EXOR は線形分離不可能である。

以上のように、パーセプトロンは 3 層のニューラルネットワークであり、それを構成するニューロンは MacCulloch・Pitts モデルを基本にしている。しかし、パーセプトロンには制約がある。1 つは活性化関数として閾値を持つステップ関数を使用していることである。もう 1 つは、ネットワークが入力パターンと出力パターンを提示されることでシナプス結合重みを調整していく際、可変である結合重みは、A 層と R 層を結合するシナプスのみであるという点である。これによってパーセプトロンは、EXOR のような線形分離不可能なものを学習によって認識することができない。





(a) 線形分離可能な例



(b) 線形分離不可能な例

図 2.11 AND と EXOR の線形分離

## 2.4 誤差伝播学習

Rumelhart らによって誤差逆伝播法が考え出された。それはパーセプトロンの拡張である。パーセプトロンは S 層と A 層の結合が固定であったが、この方法では S 層と A 層の結合重みについても更新するようにしている。これによってパーセプトロンでは不可能であった排他的論理和の学習が可能になった。またこの方法は、階層型ニューラルネットワークについて、複数の中間層をもつネットワークにも適用できる。この誤差伝播学習について説明するため、まず

この学習法のもとである勾配法について述べる。そしてその後、誤差逆伝播法の概要と導出について述べる。

### 2.4.1 勾配法について

勾配法は、関数の極小、あるいは極大値を見つけ出すためのアルゴリズムである。この方法は、山下り法（山登り法）とも呼ばれる。例えば、登山中に深い霧の中で立ち往生したとき、自分の周囲の傾斜を見て、より低くなっているほうに進む。すると少なくとも1つの窪み（極小点）に到達できる。原理を図 2.12 に示す。

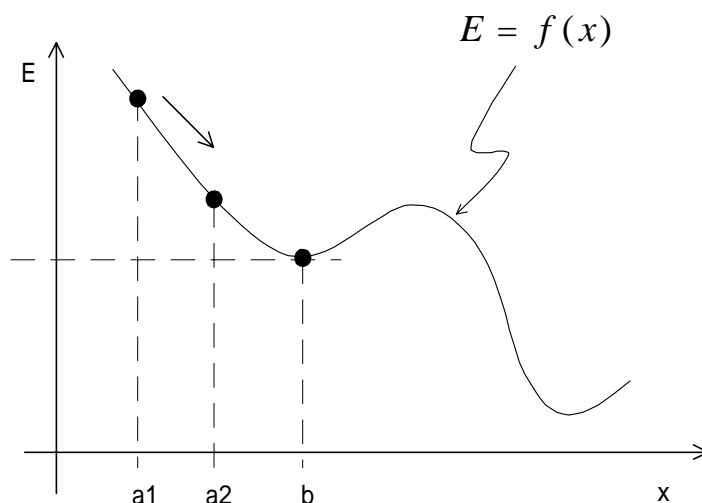


図 2.12 勾配法の原理

今、ある  $x$  についての関数  $E$  の極小値を求めるとする。適当な初期値を設定し、局所探索を繰り返すことによって極小値を求める。まず  $x$  の初期値として  $x=a_1$  を与える。このときの  $E$  の傾きを調べ、 $E$  の値が減少する方向に  $x$  を変化させる。変化した結果  $x=a_2$  となったとする。さらにその時の  $E$  の傾きを調べ、 $E$  が減少するよう  $x$  を変化させる。同様の操作を繰り返すと、極小点に到達する。このとき  $x=b$  であるとする、 $E$  の傾きは 0 であり、 $x$  をどちらに移動させても  $E$  の値は増加する。これによって繰り返しおこなう計算は終了する。関数  $E$  の傾きを  $g$  とすると、 $x$  の更新は式 (2.8) のように表される。

$$g = f'(x) \quad (2.7)$$

$$x = x - \alpha g \quad (2.8)$$

式 (2.8) の  $\alpha$  は  $x$  を更新する際の更新する度合いを定める定数である。

## 2.4.2 誤差逆伝播法の概要

パーセプトロンでは、R 層（出力層）の素子は出力が誤りであれば、その誤りは教師信号が教える。しかし A 層（中間層）の素子は、どの素子の動作が悪いために出力層の結果がおかしいのか判定するのが難しい。よって回路網全体で、総合的にどう修正するのか決めなくてはならない。そこで誤差伝播学習は、ニューラルネットワークを結合重みという可変パラメータをもつシステムとして考える。それはネットワークの入力から出力までの関係が、そこに含まれる結合重みのすべてに依存して決まるためである。ここでネットワークの出力と目標出力についての評価関数を導入し、それによって重みを変更する。図 2.13 は、誤差伝播学習の概要を示したものである。

図 2.13 の入力パターンと目標出力パターンは、ネットワークに学習させたい入出力関係、すなわち訓練データである。パラメータ  $w$  は結合重みである。ある入力データ ( $x_1, x_2, \dots, x_l$ ) が入力され、それに対する出力 ( $z_1, z_2, \dots, z_n$ ) が得られたとする。そして、その出力と目標データ ( $d_1, d_2, \dots, d_n$ ) を用いて、学習達成度を測る。その尺度となるものが評価関数  $E$  である。勾配法のように、 $E$  の値が小さくなるようにパラメータ  $w$  の値を微小変化させる。この手続きを図 2.14 に示す。

図 2.14 の  $L$  は変数である。1 つの入力データに対して評価関数  $E$  の勾配を求め、パラメータ  $w$  を更新する。この操作を入力パターンの数だけ行う。入力パターンが  $p$  個あるとすると、 $L = p$  となるときすべての入力パターンをネットワークに提示したことになる。そして収束条件を満たしていれば学習は終了であるが、収束条件は、すべての入力パターンにおいて、 $E$  の微分値が 0 となるときである。それはすべての入力パターンを提示したとき、重みの更新が 1 回もないことと同じである。

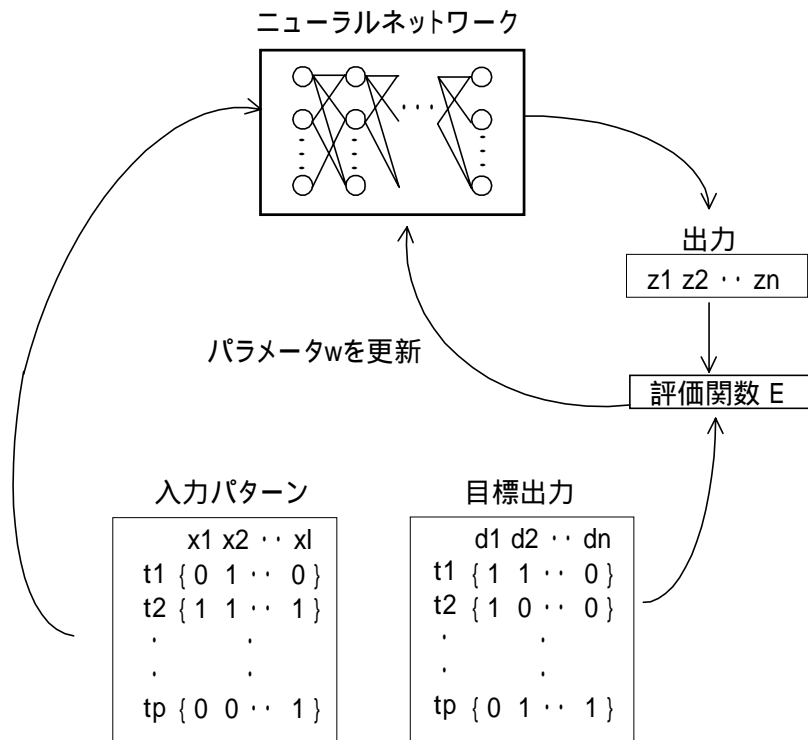


図 2.13 ニューラルネットワークの学習の概要

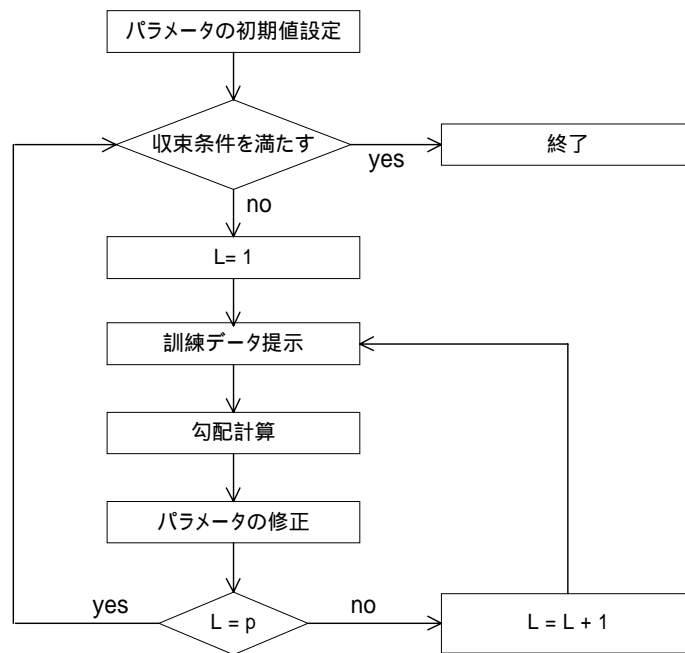


図 2.14 学習アルゴリズム

### 2.4.3 誤差逆伝播法の導出

図 2.15 のような階層型ニューラルネットワークについての誤差伝播学習を考える。図 2.15 のネットワークは 3 層構造であり、入力層の素子数は  $l$  個、中間層は  $m$  個、そして出力層は  $n$  個のニューロンである。今、 $L$  番目の入力パターンが入力され、出力  $z$  ( $z_1, z_2, \dots, z_i, \dots, z_n$ ) が得られたとする。中間層出力  $y$  と出力層の値  $z$  はそれぞれ式(2.9)、式(2.10)を用いて表される。

$$y_j = f(\sigma_j) = f(\sum x_i w_{ij}) \quad (2.9)$$

$$z_k = f(\sigma_k) = f(\sum y_j w_{jk}) \quad (2.10)$$

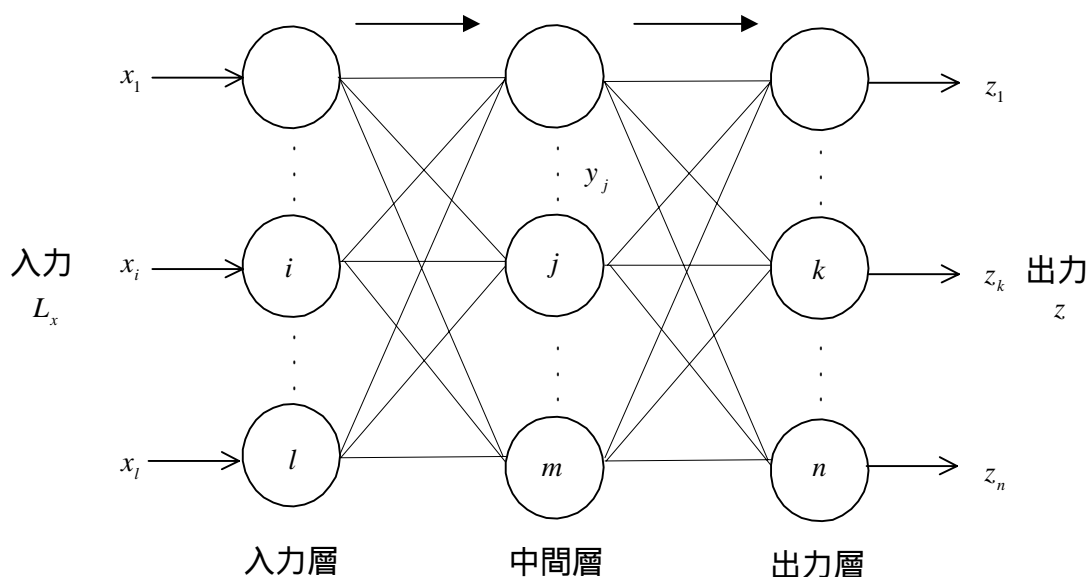


図 2.15 誤差伝播学習の導出

$L$  番目の入力データに対する入力目標を  $d$  ( $d_1, d_2, \dots, d_k, \dots, d_n$ ) とする。評価関数を式(2.11)のように定義する。

$$E = \frac{1}{2} \sum_{k=1}^n (d_k - z_k)^2 \quad (2.11)$$

この  $E$  の傾きを用いて重みパラメータ  $w$  を変更する。中間層と出力層の重み  $w_{jk}$

は式 (2.12) のようになる。

$$\frac{\partial E}{\partial w_{jk}} = -(d_k - z_k) z_k' \cdot y_j \quad (2.12)$$

ここで、出力層の誤差信号を

$$\delta_{jk} = -(d_k - z_k) z_k' \quad (2.13)$$

とおくと、中間層と出力層を結合する重み  $w_{jk}$  は、この誤差信号 と学習率 を用いて式 (2.14) のように表せる。

$$w_{jk} = w_{jk} + \rho \cdot \delta_{jk} \cdot y_j \quad (2.14)$$

そして入力層と中間層の結合重み  $w_{ij}$  については、E を用いて、

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{w_{ij}} \quad (2.15)$$

となる。式 (2.15) は

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}} &= \frac{\partial E}{\partial y_j} \frac{\partial y_j}{w_{ij}} \\ &= -\sum_k (d_k - z_k) \frac{\partial E}{\partial y_j} \frac{\partial y_j}{w_{ij}} \\ &= -\left(\sum_k \delta_k w_{jk}\right) y_j' \cdot x_i \end{aligned} \quad (2.16)$$

となる。式 (2.17) は、中間層の誤差信号を  $\delta_{ij}$  とすると、 $\delta_{ij}$  は、

$$\delta_{ij} = -\left(\sum_k \delta_k w_{jk}\right) y_j' \quad (2.17)$$

とおける。よって入力層と中間層を結合する重みは、式 (2.18) のように表せる。

$$w_{ij} = w_{ij} + \rho \cdot \delta_{ij} \cdot x_i \quad (2.18)$$

2.4 節では、Rumelhart らによって発明された誤差逆伝播法について述べた。これはパーセプトロンの拡張であり、排他的論理和を学習できないという弱点を克服するものである。またこの方法では、評価関数の導入と、勾配法を活用することによってすべての結合重みを変更できる。

## 第3章

### ニューラルネットワークのハードウェア化

この章では、ニューラルネットワークの新しいハードウェア化について提案する。ハードウェア化の従来手法として、活性化関数であるシグモイド関数を3値に単純化したものがある[9-10]。この方法では、活性化関数を単純にすることで乗算を論理積で行うことができる。また学習には誤差逆伝播法を使用する。この学習時においても活性化関数の導関数を単純化することで、複雑な乗算を無くしている。シナプス結合重みはアップダウンカウンタで実現されている。出力誤差はパルス数で表現され、学習時にはこのパルスを利用してカウンタをインクリメント、またはデクリメントさせる。このカウンタ動作が結合重みの更新であり、カウンタはパルス数に応じた変化をする。カウンタのビット長が小さすぎると、パルス当りの変化量が大きくなる。その結果、計算精度が低くなり、学習が達成されない。ビット長が大きければ、パルス当りの変化量は小さく、計算精度は上がるが、ネットワークの規模が大きくなると計算時間の増大につながる。

このような従来手法の欠点を取り除くため、新たなハードウェア化手法を提案する。提案手法は、カウンタのビット長を大きくしても計算時間の増大を抑えるものである。通常はニューロンの出力と教師信号の誤差量によってカウンタの変化量が決まるが、提案手法ではニューロンへの入力信号と重みの加重和を利用することで誤差量をより小さくするようカウンタの変更をおこなう。

この章では、従来手法について説明した後、提案手法を説明する。従来手法の説明は、入力から出力にいたる通常の計算（フォワードプロセス）と、学習時の誤差を重みに反映させる計算（バックワードプロセス）に分けて行う。

#### 3.1 フォワードプロセス

図 3.1 にニューラルネットワークのハードウェア構成を示す。図 3.1 はニューラルネットワークと教師ユニットから成る。ネットワークの入出力は入力（input,  $\theta$ ）と出力（one, half）となっている。input はネットワークの入力で



あり、それに対する出力が (one1, half1, ..., one n, half n) である。その出力は教師ユニットの入力となり、教師ユニットはそれぞれの入力信号に対する誤差量を計算し、誤差に応じたパルス信号 ( $+\theta_1, -\theta_1, \dots, +\theta_n, -\theta_n$ ) を出力する。教師ユニットからの出力信号はニューラルネットワークへの入力となり、この入力信号は、ネットワーク内部の結合重みであるカウンタを更新するために使用される。

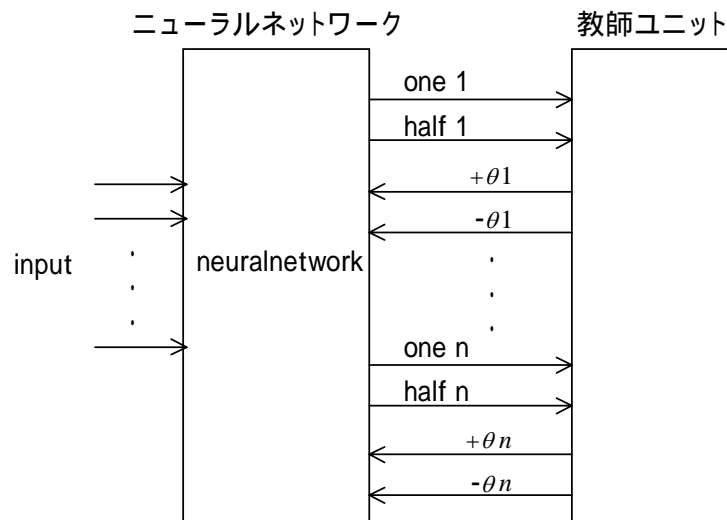


図 3.1 ニューラルネットワークのハードウェア構成

ニューラルネットワークユニットはニューロンユニットとシナプスユニットで構成される。図 3.2 に 2 つのユニットの入出力を示す。

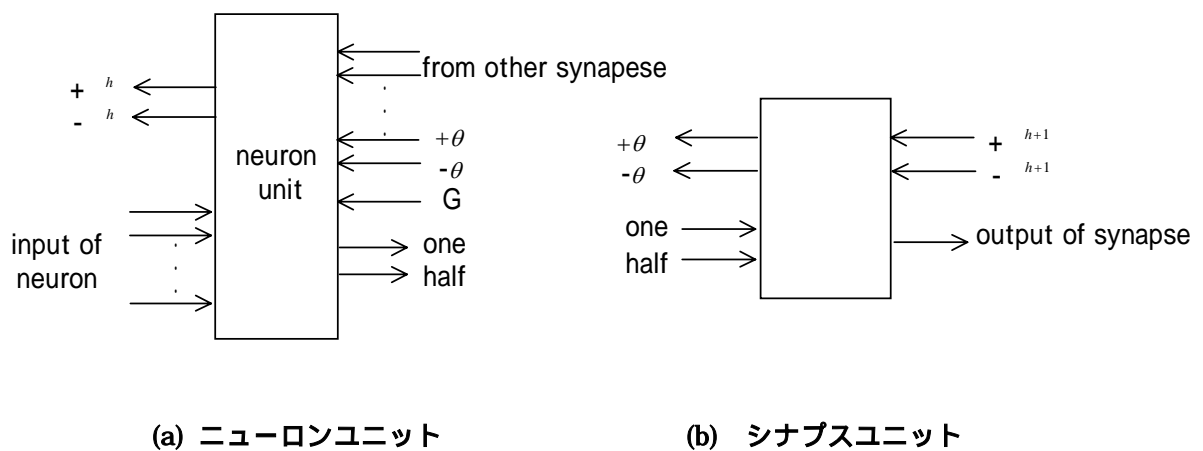


図 3.2 ユニットのブロック図

図 3.2(a)はニューロンユニット、(b)はシナプスユニットである。シナプスユニットはフォワードプロセス時の動作とバックワードプロセス時の動作を持つ。フォワードプロセス時において、ニューロンの入力信号 (input of neuron) からのデータが計算され、結果は出力信号 (one, half) から出力される。バックワードプロセス時には、誤差信号 を生成するための信号 ( $\pm\theta, G$ ) が入力される。出力信号  $\delta^h$  は第 h 層のニューロンの誤差信号であることを示している。

図 3.2(b)のシナプスユニットもニューロンユニットと同様にフォワードプロセス時の動作とバックワードプロセス時の動作を持つ。フォワードプロセス時はニューロンユニットからの信号が入力信号 (one, half) となる。この入力信号と内部のアップダウンカウンタの値との積がシナプスユニットの出力 (output of synapse) となる。バックワードプロセス時にはニューロンユニットからの誤差信号 が入力信号となり、この  $\pm$  とシナプス結合重みの積が出力信号 ( $\pm\theta$ ) となる。

フィードフォワード時におけるニューロンユニットの動作について説明する。ニューロンの計算は式 (2.1) (2.3) のように、入力と結合重みの加重和とシグモイド関数で表される。シナプスユニットで入力信号  $x$  と結合重み  $w$  の乗算が計算される。ニューロンユニットでその総和 とシグモイド演算を行う。

$$\sigma = \sum_{i=1}^n x_i \cdot w_i$$

$$y = f(\sigma) = \frac{1}{1 + \exp^{-r\sigma}}$$

この計算をハードウェア化するために、シグモイド関数の簡単化を行う。簡単化したものを図 3.3 に示す。

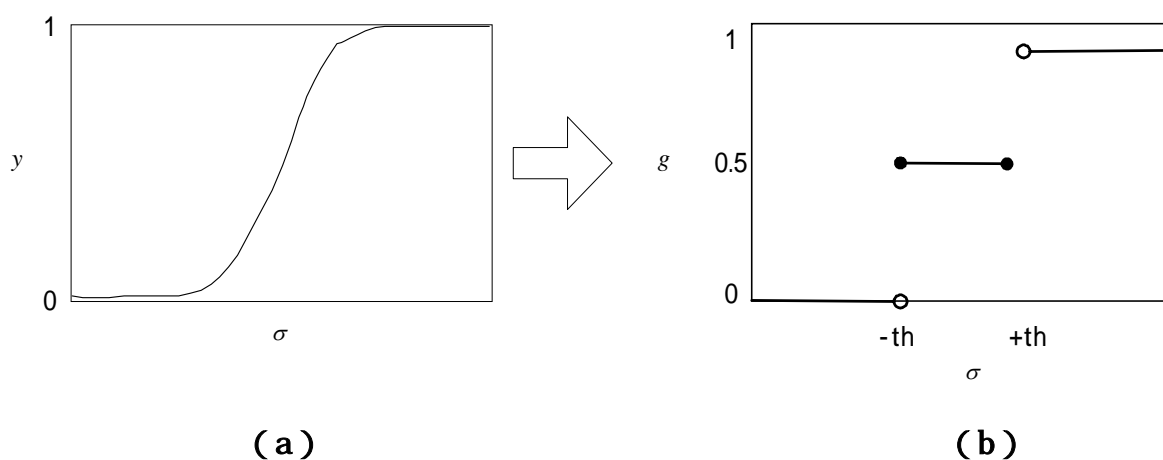


図 3.3 シグモイド関数の簡単化

図 3.3(a)はシグモイド関数であり、閾値を用いて出力を 3 値に簡単化したものが図 3.3(b)である。式で表すと式 (3.1) のようになる。

$$g(\sigma) = \begin{cases} 0 & : \sigma < -th \\ 0.5 & : -th \leq \sigma \leq th \\ 1 & : \sigma > th \end{cases} \quad (3.1)$$

図 3.4 にニューロンユニットのブロック図を示す。フィードフォワード時において、ニューロンへの入力 (input of neuron) をシリアルデータで受け取り、そのデータをビットシリアルアダプタで加算する。加算結果はシフトレジスタに取り込まれ、閾値  $\pm th$  と比較される。その結果、0、0.5、または 1 のうち 1 つが出力として決定される。表 3.1 にニューロンユニットの出力と の真理値表を示す。表 3.1 に示すように、出力は 3 値であるので、出力を one, half の 2 ビットで表すことができる。

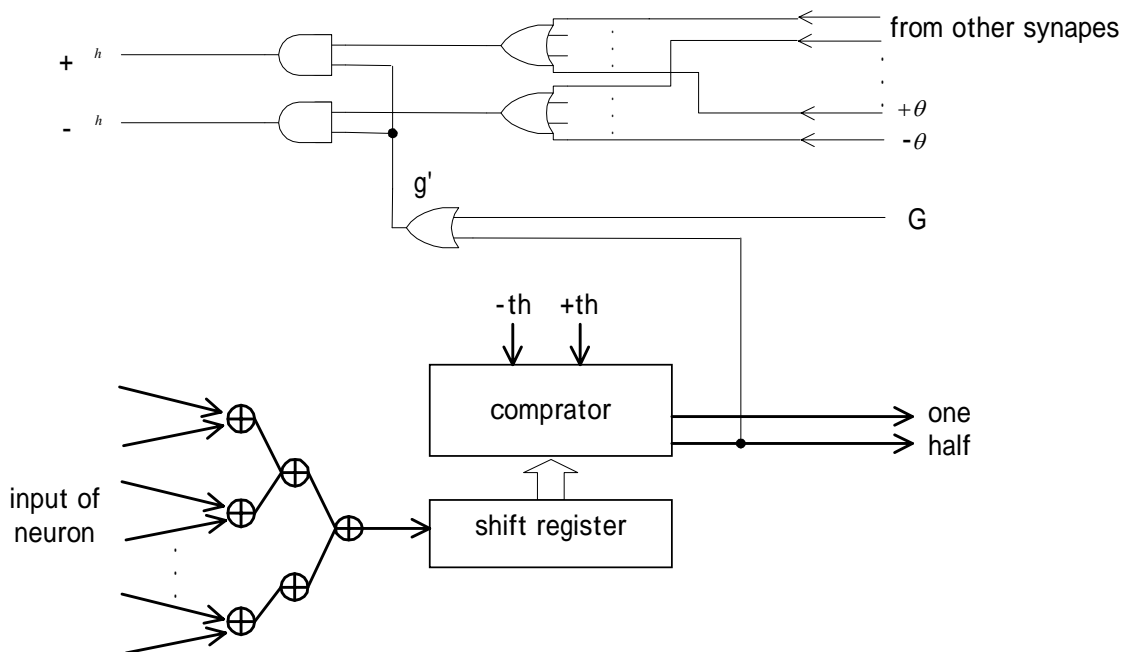


図 3.4 フィードフォワード時のニューロンユニット

表 3.1 ニューロンユニットの真理値表

	one	half	$u(\ )$
$\sigma < -th$	0	0	0
$-th \leq \sigma \leq th$	0	1	1
$\sigma > th$	1	0	0

シナプスユニットのブロック図を図 3.5 に示す。フィードフォワード時には、ニューロンユニットからの出力 (one, half) がシナプスユニットの入力となる。結合重みは固定小数点フォーマットでアップダウンカウンタに格納されており、乗算はそのカウンタの値と入力信号との間で行う。乗算はマルチプレクサ (mux3) と AND で実現されている。その結果がシナプスユニットの出力 (output of synapse) となる。入力 (one, half) はカウンタに入力されているが、これは学習時 (フィードバックプロセス) においてカウンタの更新に使用される。ニューロンの出力が 0.5 のとき、mux3 はカウンタの出力をシリアル変換した値  $Q$  を出力する。ニューロンの出力が 1 のとき  $Q$  は 1 ビット分の遅延を入れた  $Q'$  を出力する。 $Q$  は  $Q'$  に比べて 1 ビット分早く出力される。よってカウンタの値を 1 ビット分だけ右シフトしたことになる。この結果ニューロンの出力が 0.5 のとき、 $Q'$  はカウンタの値を 1/2 倍したものに等しい。シナプスユニットの入力 (one, half) と出力 (output of synapse) の真理値表を表 3.2 に示す。

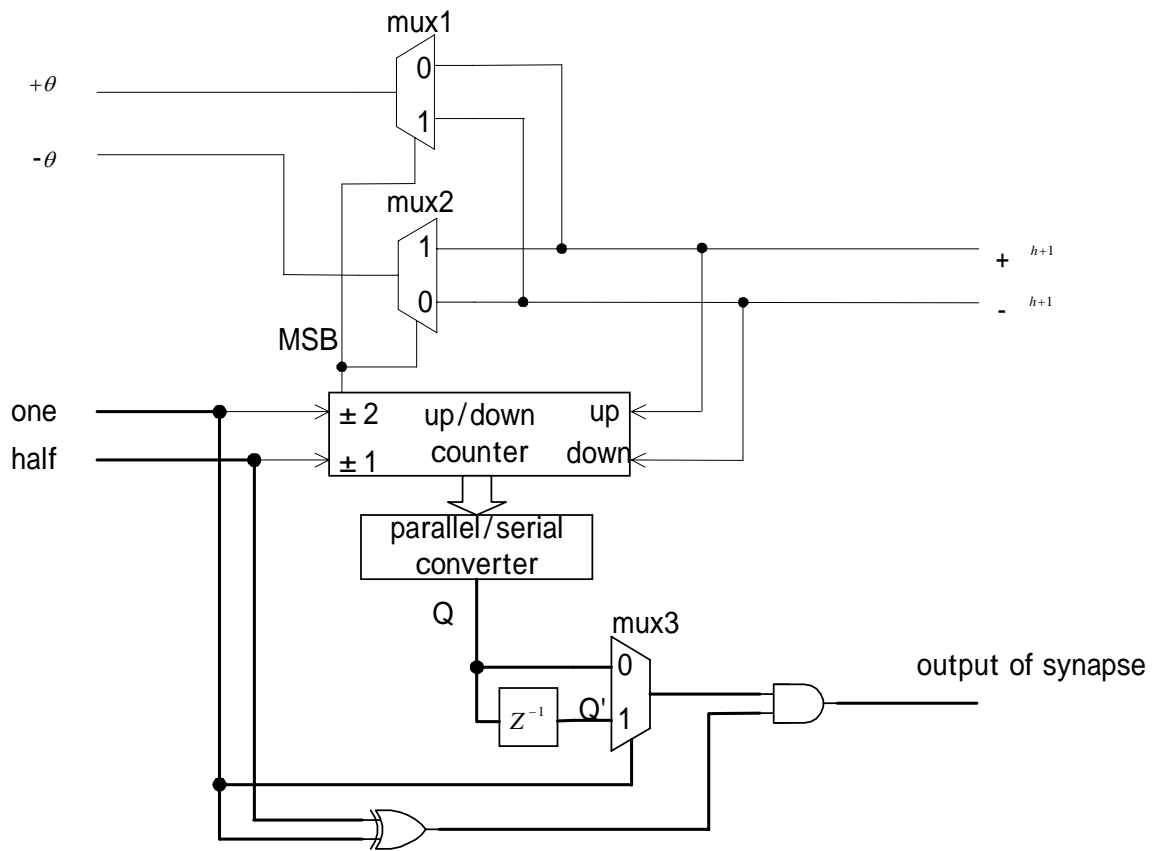


図 3.5 フィードフォワード時のシナプスユニット

表 3.2 シナプスユニットの真理値表

$g(\ )$	one	half	output of synapse
0	0	0	0
1/2	0	1	Q
1	1	0	Q'

## 3.2 バックワードプロセス

バックワードプロセスにおけるニューロンユニットとシナプスユニットの動作について説明する。バックワードプロセスは誤差信号を生成し、下位層に伝播させて結合重みを更新する作業である。バックワードプロセスにおけるニューロンユニットは下位層に出力誤差を生成し、伝播させる働きをする。その動作は式(2.13)(2.17)で表現される。式(2.13)は出力層の誤差信号であり、式(2.17)は中間層の誤差信号である。

$$\delta_{jk} = -(d_k - z_k)z_k'$$

$$\delta_{ij} = -\left(\sum_k \delta_k w_{jk}\right)y_j'$$

ここで $\theta$ を式(3.2)のようにおく。

$$\theta = \begin{cases} d_k - z_k & : \text{output layer} \\ \delta_k w_{jk} & : \text{hidden layer} \end{cases} \quad (3.2)$$

すると式(2.13)(2.17)は式(3.3)(3.4)のようになる。

$$\delta_{jk} = -\theta \cdot z_k' \quad (3.3)$$

$$\delta_{ij} = -\left(\sum \theta\right) \cdot y_j' \quad (3.4)$$

ニューロンユニットでは、この式(3.3)(3.4)を実現している。図3.6にバックワード時のニューロンユニットのデータパスを示す。図3.6は中間層のニューロンユニットである。式(3.4)のように、ニューロンユニットの出力は、 $\theta$ の総和とニューロンユニットの出力関数の微分値との乗算である。図3.6では、シナプスユニットで生成される $\theta$ の総和はOR、乗算はANDで実現されている。これは、 $\theta$ がパルスで表現されていることと、活性化関数の導関数を単純化したことによる。単純化した導関数を図3.7に示す。ニューロンユニットの出力が0、または1のときは微分値Bをとり、0.5のときは微分値1をとる。もしBの値が0であれば、下位層に誤差信号が全く伝わらないので $0 < B < 1$ にしなくてはならない。これは、ランダム信号Gとニューロンユニットの出力halfのORをとることで実現する。Gは線形フィードバックシフトレジスタ(LFSR)によって生成される。図3.7に訓練データとGの関係を示す。p個の訓練データ $t_1, t_2, \dots, t_p$ の提示を1回の学習とする。Gが8ビットのLFSRであったとすると、GはN/255の割合で1を出力する。よってニューロンユニットの出力が0、

または1のときは、255回の学習のうちN回誤差を伝達させることになる。もしニューロンユニットが出力層であった場合にはORをとる必要はない。なお図3.6において $\theta$ の正負の区別をつけるため、 $+\theta$ と $-\theta$ の2本の線を用いている。

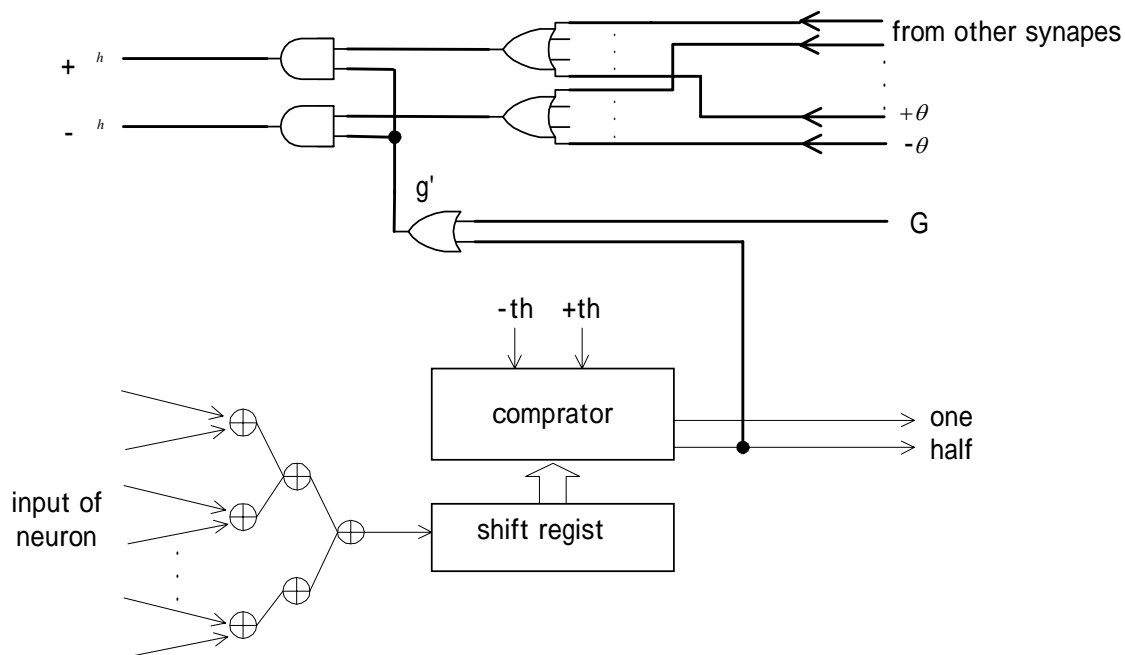


図 3.6 バックワード時のニューロンユニット

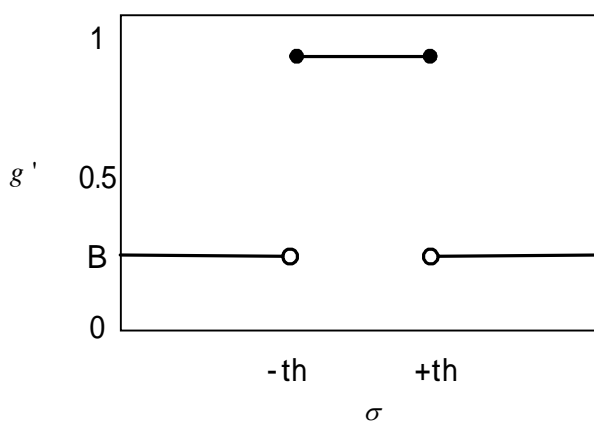


図 3.7 活性化関数の導関数

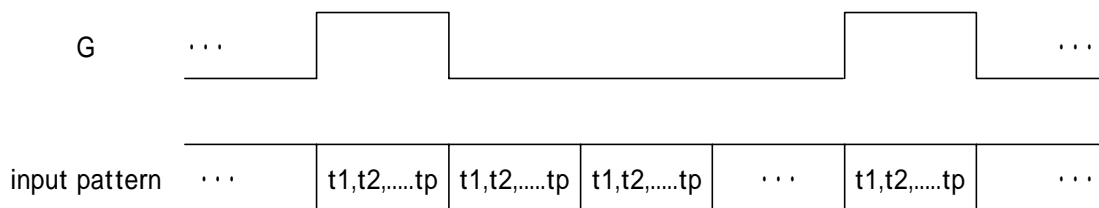


図 3.8 G の発生と訓練データ

バックワードプロセス時のデータパスを図 3.9 に示す。シナプスユニットは、バックワードプロセスにおいて上位層のニューロンユニットからの誤差信号  $\delta^{h+1}$  と、下位層のニューロンユニットからの出力信号(one, half)を用いてカウンタを更新し、 $\theta$  を出力する。

$\theta$  の計算は式 (3.2) のように、出力層ニューロンであるか、中間層ニューロンであるかによって計算が異なる。出力層における  $\theta$  の計算は、教師信号と出力層ニューロンの出力値の差の計算である。ニューロンユニットの出力は 3 値であるので、教師信号との差は、0、 $\pm 0.5$ 、 $\pm 1$  しかない。出力層ニューロンユニットへの入力となる  $\theta$  は、教師ユニットから差に応じたパルスとして出力される。差が 0 のときはパルス 0 個、差が  $\pm 0.5$  のときはパルス 1 個、差が  $\pm 1$  のときはパルス 2 個を出力する。パルスは符号を表現できないので、正と負で信号線を別にしている。中間層における  $\theta$  の計算は、結合重みと誤差信号の乗算で表される。図 3.9 のように、 $\theta$  の計算は結合重みの符号のみで行っている[16-17]。その動作を表 3.3 に示す。重み値の MSB を符号ビットとして、重みが負であれば誤差信号の符号を入れ替え、重みが正であれば誤差信号の符号はそのままにする。

カウンタの更新は式 (2.14)(2.18) で表される。

$$w_{jk} = w_{jk} + \rho \cdot \delta_{jk} \cdot y_j$$

$$w_{ij} = w_{ij} + \rho \cdot \delta_{ij} \cdot x_i$$

式(2.14)は中間層と出力層の結合重み、式 (2.18) は入力層と中間層の結合重みである。中間層と出力層の結合重みは、出力層 (上位層) ニューロンユニットで生成された誤差信号 と中間層 (下位層) ニューロンユニットの出力値の積である。入力層と中間層の結合重みは中間層 (上位層) ニューロンユニットで生成された誤差信号 と入力層 (下位層) ニューロンユニットの出力値の積である。入力層ユニットの出力値はユニットへの入力値と等しい。図 3.9 のカウンタの動作を表 3.4 に示す。カウンタの更新量をニューロンユニットの出力値の 2



倍に等しくなるようにする。これによって、カウンタは誤差信号と出力値の乗算に比例した大きさに変化。重み更新に学習率が用いられているが、図 3.9 においてはカウンタのビット長で決まる。

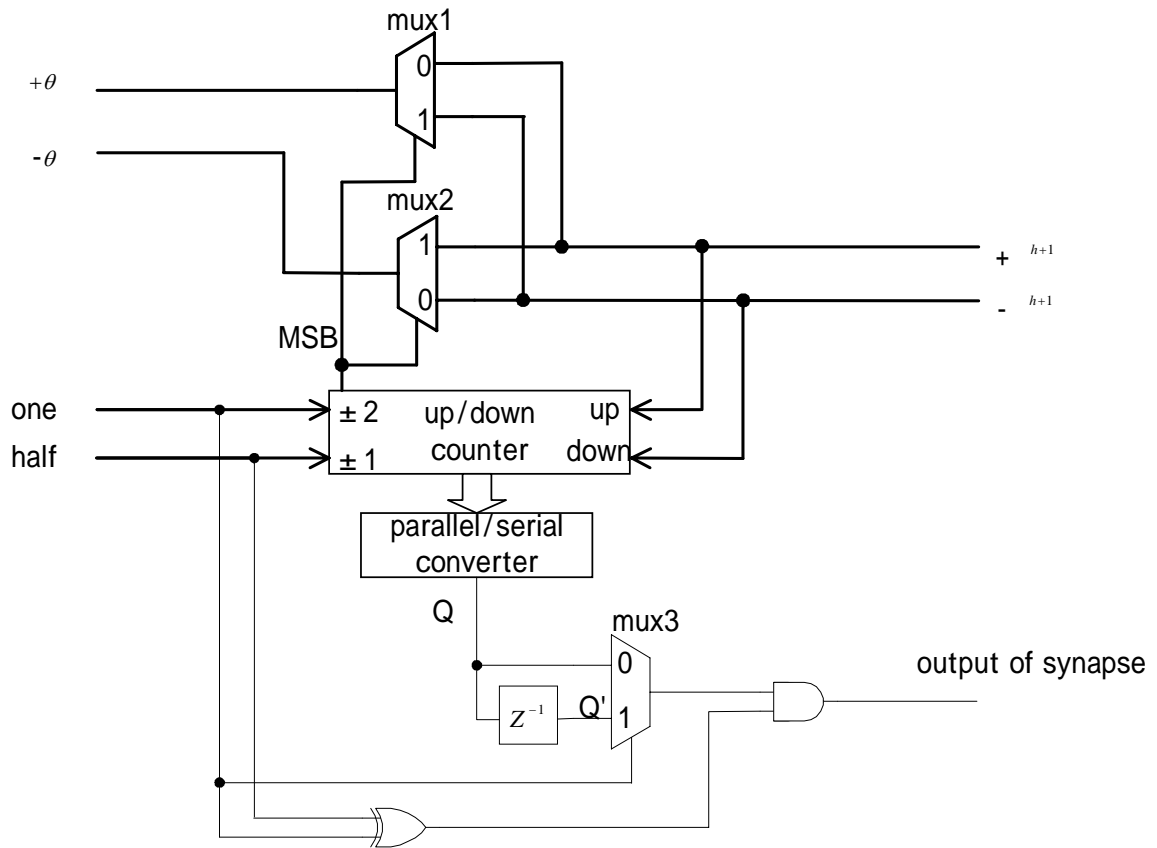


図 3.9 バックワード時のシナプスユニット

表 3.3 符号ビットと誤差信号

MSB of counter	$+\theta$	$-\theta$
0	$+_{h+1}$	$-_{h+1}$
1	$-_{h+1}$	$+_{h+1}$

表 3.4 アップダウンカウンタの動作

one $\pm 2$	half $\pm 1$	$+^{h+1}$ up	$-^{h+1}$ down	up /down counter
0	0	no pulse		no change
0	1			increment by one
1	0			increment by two
0	1			decrement by one
1	0			decrement by two

### 3.3 問題点と改善方法について

従来手法では結合重みをカウンタで実現しており、そのビット長が小さいと学習は達成されにくく、ビット長を大きくすれば計算精度は向上するが、学習時間の増大となる。その増大を抑えるために、図 3.7 の B の値を大きくすることで、下位層への誤差伝播頻度を上げるという方法が考えられる。しかし B の値を大きくしすぎると、学習時間の増大と学習達成率が下がる。そこで学習率を変動させて学習時間の増大を抑える手法を提案する。その方法は、出力層のニューロンユニットに新たな閾値を設け、それによって誤差信号のパルス数を増加させるというものである。図 3.10 に新たな閾値を導入した活性化関数を示す。図 3.10 の  $\pm th'$  が追加された閾値である。図中の領域 a' と領域 c' のように、加重和が  $-th'$  以下、あるいは  $+th'$  以上であるなら学習率  $\rho$  を大きくする。それ以外の領域 a, b, c の場合は  $\rho$  は通常値である。図 3.11 の出力 u は領域 a', c' の場合に 1 となる。このとき誤差信号のパルス数を増加させることでシナプスユニットのカウンタの更新量を増加させる。増加された更新量を  $\alpha$  とすると、学習率  $\rho$  は式(3.5)のようになる。

$$\rho = \frac{\Delta w + \alpha}{\delta \cdot y} \quad (3.5)$$

式(3.5)は信号線 u が '1' のときの  $\rho$  である。出力層の出力誤差は教師信号とニューロン出力の差であるので、u='1' であっても、出力誤差とニューロンの出力には影響しない。したがって式(3.5)の分母である  $\delta$ 、y は u='1' であっても変化しない。また  $\alpha$  はパルス数を増やすことで増加させることができる。

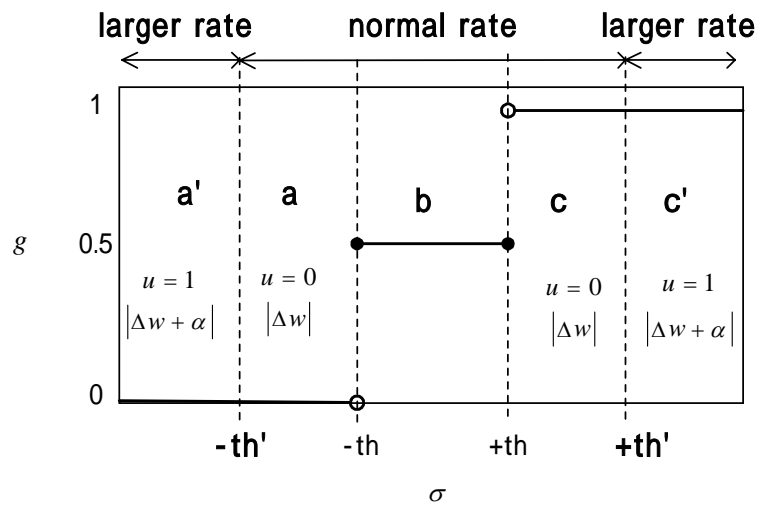


図 3.10 閾値の導入と活性化関数

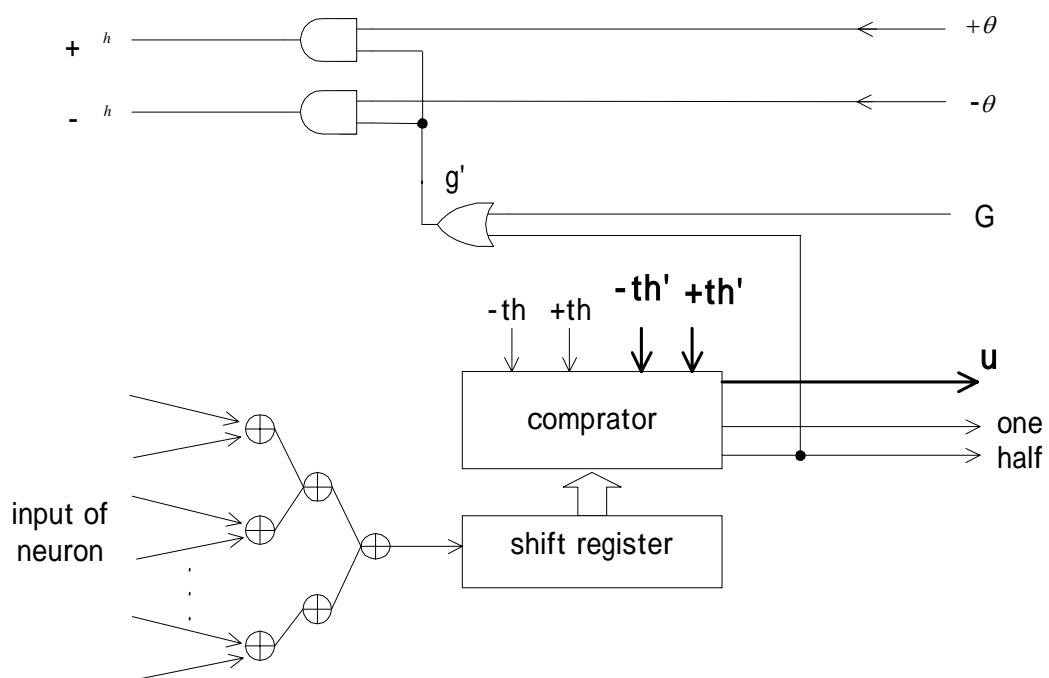


図 3.11 出力層ニューロンの信号線の追加

図 3.12 は出力誤差のパルスが伝播する様子を示したものである。出力層において  $u=1$  であったニューロンの誤差パルスは通常よりも増加されている。その増加されたパルスは結合重み  $w_a$  を更新させ、かつ下位層である中間層の結合重み  $w_b$  にも伝播する。中間層の結合重み  $w_b$  の更新は、 $w_a$  と  $\delta$  によって決まるが、 $\delta$  の伝播には  $w_a$  の符号をもちいて  $\delta$  の正、負を決めているだけである。そのため中間層に伝播するパルスの数は変化しない。よって出力層から伝播する増加されたパルスは中間層にも伝播する。

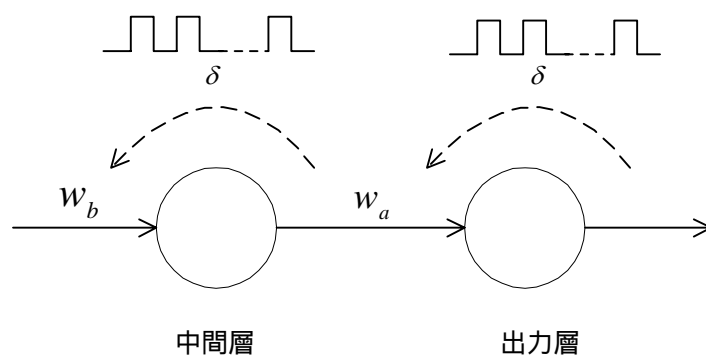


図 3.12 パルスの伝播

## 第 4 章

# ソフトウェアシミュレーション

この章では、第 3 章で説明した従来手法と提案手法[18]の学習時間を比較するために行ったソフトウェアシミュレーションの結果について説明する。実験は EXOR の学習とアルファベットの学習の 2 つである。

学習時間は、出力誤差がなくなるまでにかかった時間である。この実験では、学習が終了するまでに提示した入力パターンの回数を比較している。学習回数は、全入力パターンをネットワークに提示し終えて 1 回である。学習の終了は、全パターンを入力し終わった段階で、1 度も結合重みの更新がなかったときである。

### 4.1 EXOR の学習実験

#### 4.1.1 実験方法

この実験では、2 入力の EXOR の学習について学習時間の比較を行った。ネットワークの構成を図 4.1 に示す。ネットワークは、3 層の階層型ネットワークである。ニューロン数は、入力層 2、中間層 5、出力層 1 となっている。5 つの中間層のうち 1 つはオフセットニューロンである。オフセットニューロンとは、常に "1" を出力するニューロンである。このオフセットニューロンは、学習を進めやすくするために導入している。このオフセットニューロンがあれば、オフセット以外の中間層ニューロンの出力がすべて "0" であった場合でも、オフセットニューロンの結合重みの更新をおこなうことで学習を進めることができる。

結合重みの初期値を -16.000000 から 16.000000 の範囲でランダムな値に設定し、従来手法と提案手法について学習を行い、学習時間を比較した。結合重みのデータ形式を固定小数点とし、ビット長の下位 6 ビットを小数部と考え、重みの更新量を最小で  $1/64$  とした。従来手法の閾値は  $\pm 2$ 、提案手法で新たに付加した閾値を  $\pm 3$  とした。学習は 10000 回を超えても終了しなかった場合は失

敗とみなした。

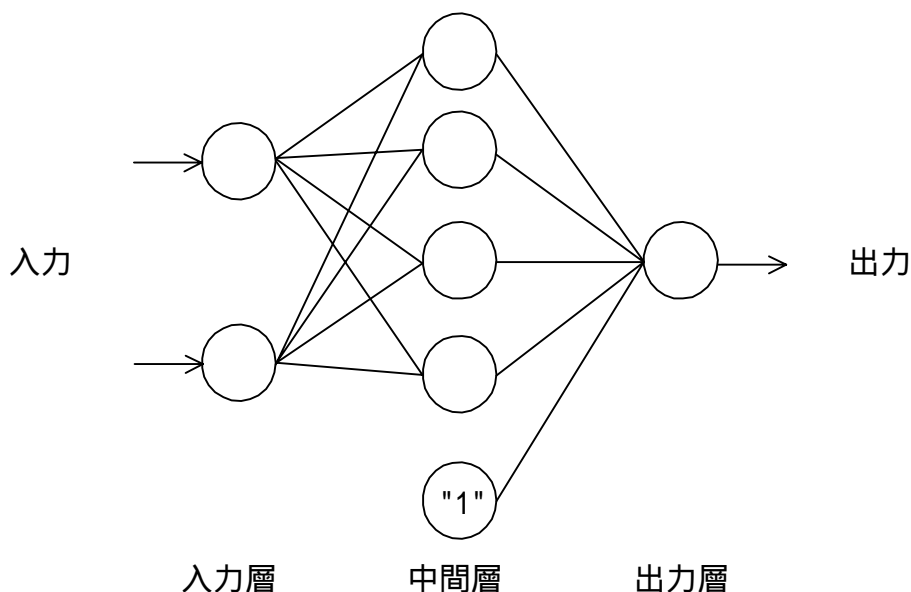


図 4.1 EXOR の学習実験

#### 4.1.2 実験結果と考察

学習実験は 177 回行った。その結果を表 4.1 と表 4.2 に示す。表 4.1 と表 4.2 は試行回数ごとの従来手法と提案手法の学習回数、および両者の差を示している。表中の学習回数が 10000 と示されているものは、学習が終了しなかったものである。従来手法による学習が失敗すると提案手法による学習も失敗に終わることが多い。また表の学習回数差は、両手法による学習回数の差をとったものである。数字が負の値になっているものは、提案手法による学習回数の方が少なかったことを示している。学習失敗したものを除けば、ほとんどの試行において提案手法の学習回数は、従来手法の学習回数より少ない。試行回数が 40 回目と 41 回目のような、元々学習回数が少ないものについては学習回数の改善はなされていない。

表 4.3 は、従来手法と提案手法の学習成功率と平均学習回数、学習回数の最大値、最小値を示している。成功率は両手法ともに同じである。平均学習回数は、学習が失敗したものを除いて算出している。これによると、従来手法の学習回数は提案手法によって、学習回数は平均で約 60 回程度改善している。しかし、最大学習回数と最小学習回数には大きな差があることと、学習回数差にはかなりのばらつきがあることを考えると、常に 60 回程度の改善を見込むことは難し

い。表 4.4 に学習回数差の最大値と最小値を示している。学習回数差は 926 が最大値であり、12 回目の試行時の値である。表 4.4 の< >内の 9803 回は 138 回目の試行時の値である。これは従来手法において、10000 回の学習で成功しなかったが、提案手法によって 197 回の学習で成功したものである。両手法ともに学習が成功している場合の最大回数差は、12 回目の試行時におけるもので、その値は 926 である。最小学習回数差は、学習回数が提案手法によって改善された場合の差で、169 回目の試行時の学習回数差である。学習回数差は、最大値と最小値で大きな差がある。学習回数差は、結合重みの初期値で大きく異なると思われる。

表 4.1 EXOR 学習の実験結果(1)

試行回数(回)	学習回数(回)		学習回数(回)
	従来手法 a	提案手法 b	b - a
1	1195	570	-625
2	597	570	-27
3	2614	2564	-50
4	969	624	-345
5	613	581	-32
6	287	167	-120
7	422	350	-72
8	3301	3219	-82
9	10000	10000	0
10	4165	10000	5835
11	294	247	-47
12	1687	761	-926
13	10000	10000	0
14	10000	10000	0
15	324	316	-8
16	10000	10000	0
17	10000	10000	0
18	1855	1269	-586
19	1533	1280	-253
20	1151	843	-308
21	515	381	-134
22	10000	10000	0
23	662	637	-25
24	72	63	-9
25	10000	10000	0
26	227	182	-45
27	199	152	-47
28	593	435	-158
29	10000	10000	0
30	970	939	-31
31	253	248	-5
32	635	518	-117
33	10000	10000	0
34	10000	10000	0
35	490	492	2
36	722	258	-464
37	799	667	-132
38	532	467	-65
39	10000	3561	-6439
40	58	58	0
41	44	44	0
42	459	451	-8
43	527	480	-47
44	987	1383	396
45	6972	6941	-31
46	2262	2242	-20
47	579	496	-83
48	10000	10000	0
49	10000	10000	0
50	10000	10000	0
51	552	439	-113
52	201	188	-13
53	1733	1717	-16
54	10000	10000	0
55	396	376	-20
56	10000	10000	0
57	548	502	-46
58	554	382	-172
59	10000	10000	0
60	581	513	-68
61	108	108	0
62	10000	10000	0
63	2117	2042	-75
64	10000	10000	0
65	548	508	-40
66	10000	10000	0
67	579	528	-51
68	392	338	-54
69	400	385	-15
70	476	458	-18
71	1126	1115	-11
72	10000	726	-9274
73	431	281	-150
74	10000	10000	0
75	10000	10000	0
76	426	296	-130
77	1363	1325	-38
78	265	250	-15
79	816	471	-345
80	284	295	11
81	269	269	0
82	417	417	0
83	759	738	-21
84	284	277	-7
85	723	645	-78
86	400	390	-10
87	158	145	-13
88	324	308	-16
89	395	360	-35
90	2379	2115	-264
91	279	279	0
92	266	270	4
93	622	587	-35
94	320	311	-9
95	436	436	0
96	10000	10000	0
97	10000	10000	0
98	432	430	-2
99	835	719	-116
100	402	438	36



表 4.2 EXOR 学習の実験結果(2)

試行回数(回)	学習回数(回)		学習回数差(回)
	従来手法 a	提案手法 b	b - a
101	403	403	0
102	10000	10000	0
103	376	372	-4
104	158	146	-12
105	798	729	-69
106	10000	10000	0
107	175	173	-2
108	10000	10000	0
109	10000	10000	0
110	317	305	-12
111	504	499	-5
112	349	337	-12
113	10000	10000	0
114	381	381	0
115	10000	10000	0
116	10000	10000	0
117	272	263	-9
118	308	301	-7
119	217	215	-2
120	509	481	-28
121	404	384	-20
122	286	10000	-9714
123	10000	10000	0
124	323	256	-67
125	900	900	0
126	284	279	-5
127	336	340	4
128	419	416	-3
129	675	642	-33
130	10000	10000	0
131	342	339	-3
132	10000	374	-9626
133	764	672	-92
134	320	320	0
135	377	377	0
136	336	348	12
137	227	223	-4
138	10000	197	-9803
139	329	284	-45
140	386	375	-11
141	611	605	-6
142	542	498	-44
143	10000	10000	0
144	1194	1328	134
145	399	393	-6
146	833	827	-6
147	692	639	-53
148	355	355	0
149	631	10000	9369
150	476	447	-29
151	534	534	0
152	306	308	2
153	587	596	9
154	706	692	-14
155	242	251	9
156	420	449	29
157	289	281	-8
158	290	295	5
159	345	335	-10
160	818	819	1
161	10000	10000	0
162	488	336	-152
163	10000	10000	0
164	187	188	1
165	439	445	6
166	501	502	1
167	541	537	-4
168	392	394	2
169	271	270	-1
170	586	304	-282
171	673	670	-3
172	258	263	5
173	282	285	3
174	462	462	0
175	240	228	-12
176	605	10000	9395
177	10000	10000	0

表 4.3 従来手法との比較 1

	成功率 (%)	平均学習回 数(回)	最大学習回 数(回)	最小学習回数 (回)
従来手法	77.4	663.0	6972	44
提案手法	77.4	603.6	6941	44

表 4.4 従来手法との比較 2

	最大回数差 (回)	最小回数差 (回)
学習回数差	926 <9803>	1

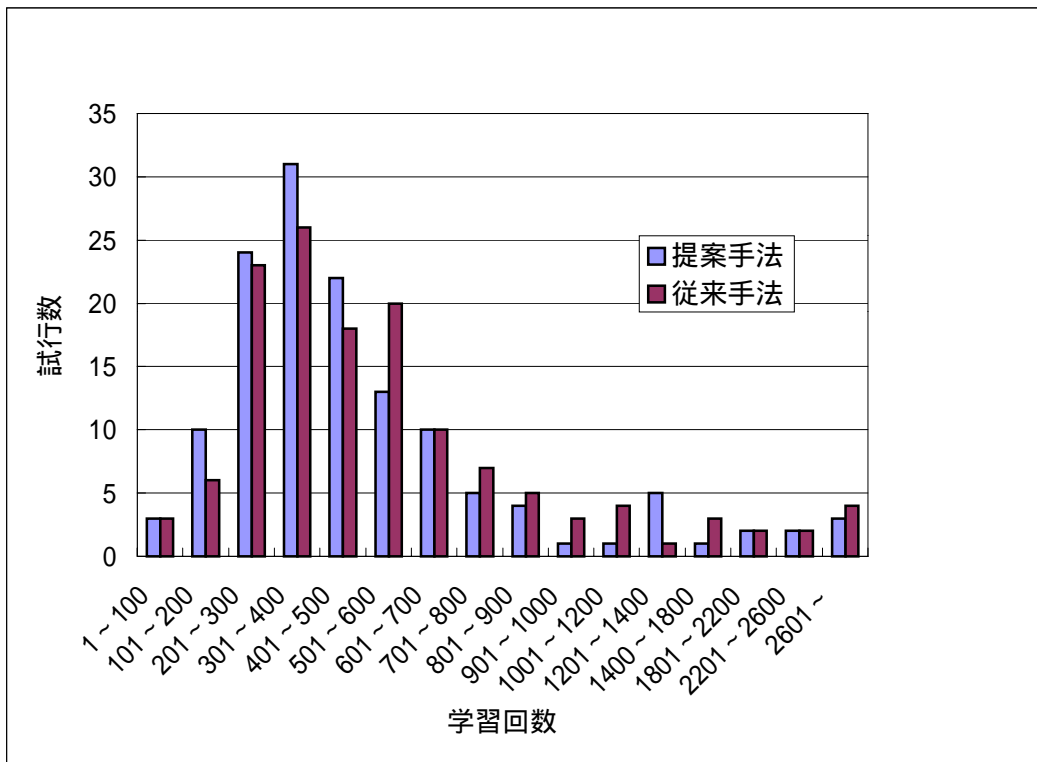


図 4.2 学習に成功した試行の度数分布

図 4.2 は、表 4.1 及び表 4.2 の EXOR の学習実験結果を学習回数ごとの度数分布である。平均学習回数は、従来手法で 660 回、提案手法で 600 回であるが、

図 4.2 の度数分布では、平均学習回数で終了した試行が最も多いグラフにはなっていない。両手法とも 301 ~ 400 回で学習終了した試行が最も多い。全体的には、平均学習回数である 600 回を下回る学習回数では、提案手法による試行が従来手法の試行より多い。学習回数が 600 回を超えると、逆に従来手法による試行が提案手法を上回るという傾向が見られる。

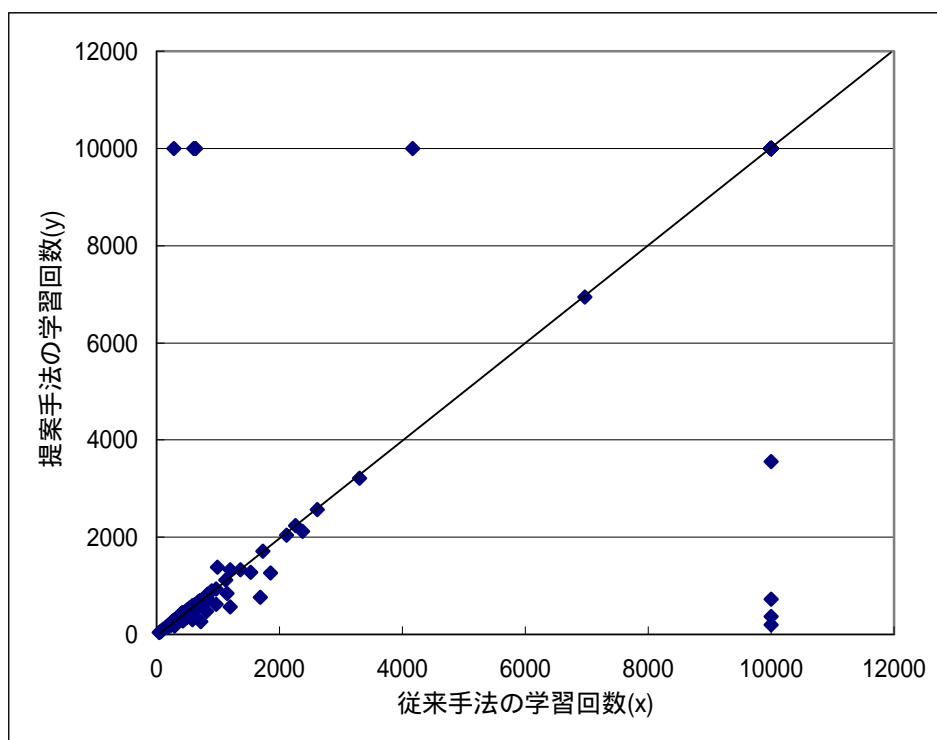


図 4.3 従来手法と提案手法の分布図

図 4.3 は提案手法と従来手法の学習回数の分布である。図 4.3 の直線は傾きが 1 の直線( $y = x$ )である。傾きが 1 未満であれば、学習回数が提案手法のほうが少ないことを示す。この分布では、学習回数が 10000 回以上のもの、すなわち学習が失敗した試行を除けば、 $y = x$  の直線に沿うような傾向が見られる。これは提案手法と従来手法の学習回数差が少ないことを示す。しかし学習回数が 2000 回程度までは、 $y/x < 1$  の領域に分布する傾向がある。これによって全体の平均学習回数が 1 割程度削減されたと思われる。

提案手法は、個々の訓練パターンとそれぞれの出力層ニューロンの誤差に対して学習率を変動させるものである。よって提案手法による改善効果は結合重みの初期値で異なる。

## 4.2 文字の学習実験

### 4.2.1 実験方法

この実験では、A~G のアルファベットの学習実験をおこなった。アルファベットは、図 4.2 のような縦 7 つ、横 5 つのマスキ目で表現した。この 35 のマスキ目の 1 つ 1 つが入力層ニューロンに相当する。図 4.2 に示された文字で、文字を形成する部分（色が濃い部分）は”1”、それ以外は”0”とする。35 のマスキ目に番号を割り当て、A~G の入力パターンを決める。番号の割り当ては図 4.3、A~G の入力パターンを図 4.4 に示す。ネットワークの出力層には 7 つのニューロンを用意し、それぞれのニューロンには A~G の文字が対応している。つまり、1 つの入力パターンに対して発火する出力層のニューロンは 7 つのうちただ 1 つである。図 4.5 に目標出力パターンを示す。例えば A の入力パターンに対して発火する（出力”1”となる）のは 1 番のニューロンのみである。図 4.6 に A の入力パターンに対する出力の様子を示す。図 4.6 のネットワークの入力層に A のパターンを入力し、出力としてただ 1 つのニューロンを発火する。もしそうでなければ、目標出力との誤差を結合重みに反映させる。

ネットワークの構成を図 4.7 に示す。ネットワークの構成は、3 層の階層型ネットワークであり、ニューロン数は入力層 35、中間層 11、出力層 7 とした。EXOR の実験と同様に、結合重みの初期値を-16 から 16 の範囲でランダムな値に設定し、従来手法と提案手法について学習を行い、学習時間の比較をした。従来手法の閾値は  $\pm 2$ 、提案手法で新たに付加した閾値を  $\pm 3$  とした。

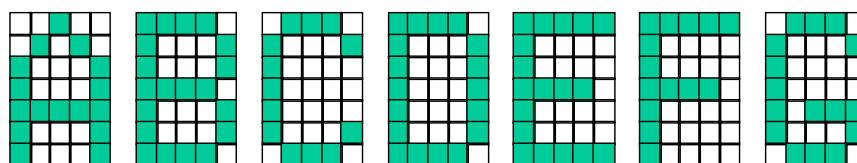


図 4.2 アルファベットの学習実験

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35

図 4.3 番号の割り当て

		入力																																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
入力文字	A	0	0	1	0	0	0	1	0	1	0	1	0	0	0	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	1	1	0	0	0	1
	B	1	1	1	1	0	1	0	0	0	1	1	0	0	0	1	1	1	1	1	0	1	0	0	0	1	1	0	0	0	1	1	1	1	1	0
	C	0	1	1	1	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	1	1	1	0
	D	1	1	1	1	0	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	1	1	1	0
	E	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	1	1	1	1	0	1	0	0	0	0	1	0	0	0	0	1	1	1	1	1
	F	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	1	1	1	1	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
	G	0	1	1	1	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	0	1	0	1	1	1	1	0	0	0	1	0	1	1	1	0

図 4.4 ネットワークの入力信号

		出力						
		1	2	3	4	5	6	7
入力文字	A	1	0	0	0	0	0	0
	B	0	1	0	0	0	0	0
	C	0	0	1	0	0	0	0
	D	0	0	0	1	0	0	0
	E	0	0	0	0	1	0	0
	F	0	0	0	0	0	1	0
	G	0	0	0	0	0	0	1

図 4.5 ネットワークの目標出力信号

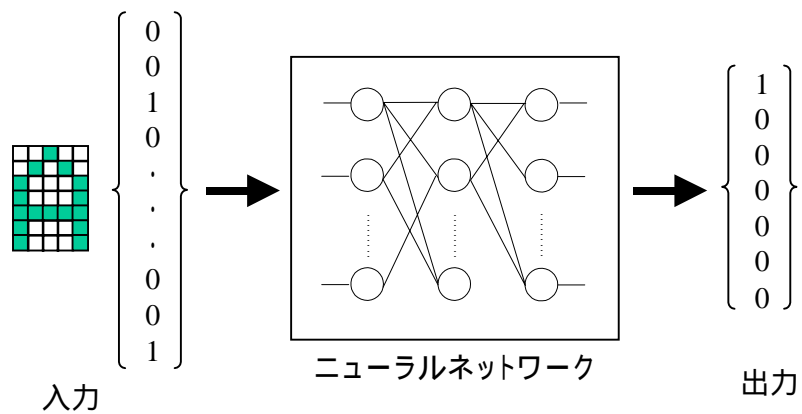


図 4.6 アルファベットの学習

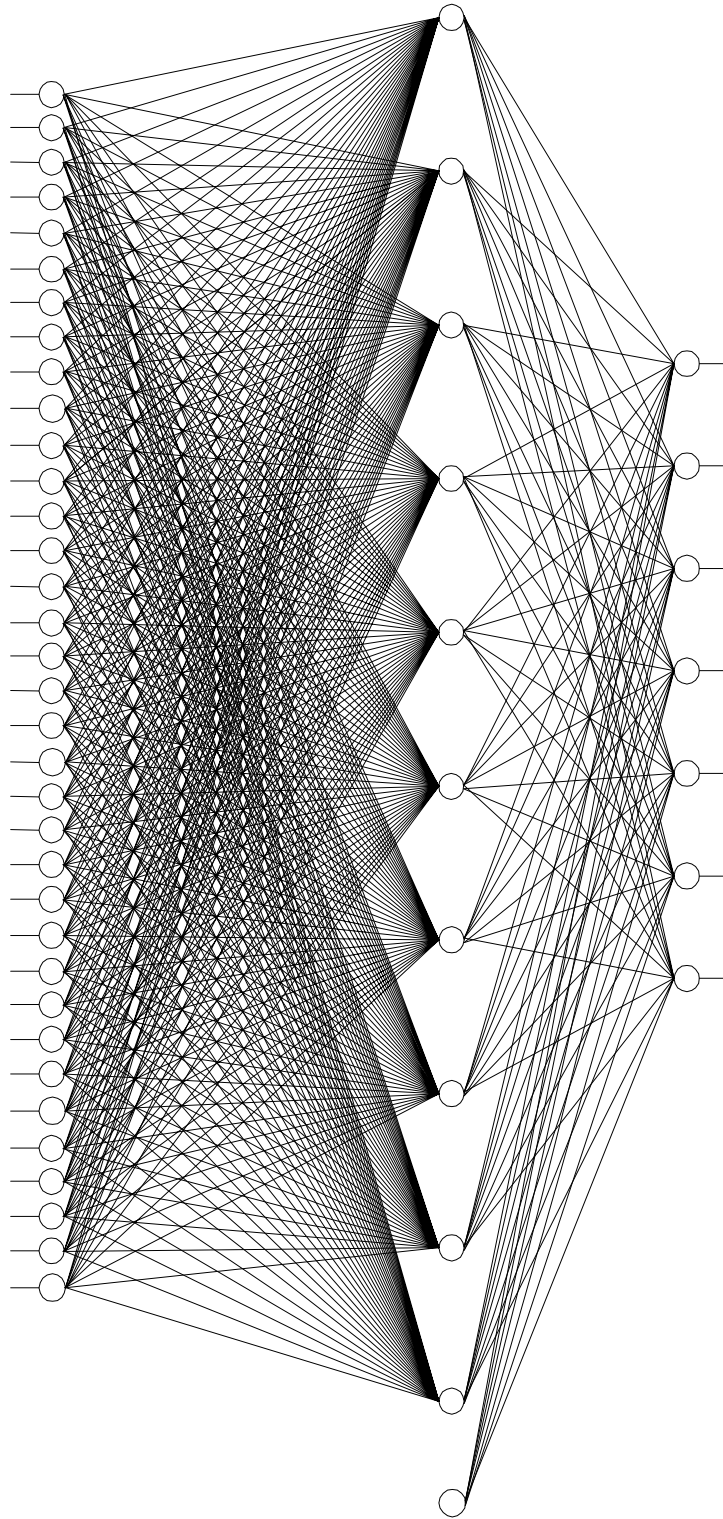


図 4.7 学習実験のネットワーク構成

#### 4.2.2 実験結果と考察

実験結果を表 4.5 に示す。実験は 30 回行ったが、すべての試行において学習は成功した。また、ほとんどの試行において学習速度が向上している。表 4.6 に従来手法との比較を示す。従来手法の平均学習回数と提案手法の平均学習回数を比べると、約 30 回程度の学習回数の差がある。しかし、EXOR の学習実験と同様に学習回数差には大きなばらつきがあるため、常に 30 回程度の改善効果を期待することはできない。

表 4.5 文字学習の結果

試行回数 (回)	学習回数(回)		学習回数差
	従来手法 a	提案手法 b	b - a
1	269	217	-52
2	231	176	-55
3	631	284	-347
4	262	314	52
5	53	43	-10
6	132	239	107
7	116	87	-29
8	167	85	-82
9	140	120	-20
10	135	129	-6
11	499	355	-144
12	87	86	-1
13	168	108	-60
14	139	107	-32
15	146	103	-43
16	209	201	-8
17	432	417	-15
18	161	149	-12
19	477	1278	801
20	405	137	-268
21	387	375	-12
22	632	577	-55
23	497	476	-21
24	472	258	-214
25	577	417	-160
26	769	829	60
27	350	375	25
28	125	123	-2
29	328	274	-54
30	560	286	-274



表 4.6 従来手法との比較

	成功率 (%)	平均学習回数 (回)	最大学習回数 (回)	最小学習回数 (回)
従来手法	100	318.5	632	53
提案手法	100	287.5	1278	43

図 4.8 に 1 回目の試行の学習経過を示す。図 4.8 のグラフは横軸に学習回数、縦軸に出力誤差をとっている。1 回目の試行において、従来手法による学習は 269 回、提案手法では 217 回の学習で誤差が 0 となり学習が終了している。提案手法によって学習回数に改善効果があったものは、図 4.8 と同じように学習回数の早い段階で誤差をより少なくする経過をたどった。図 4.9 は、逆に提案手法による学習のほうが学習時間がかかった例である。図 4.9 は 19 回目の試行における誤差の経過を示している。

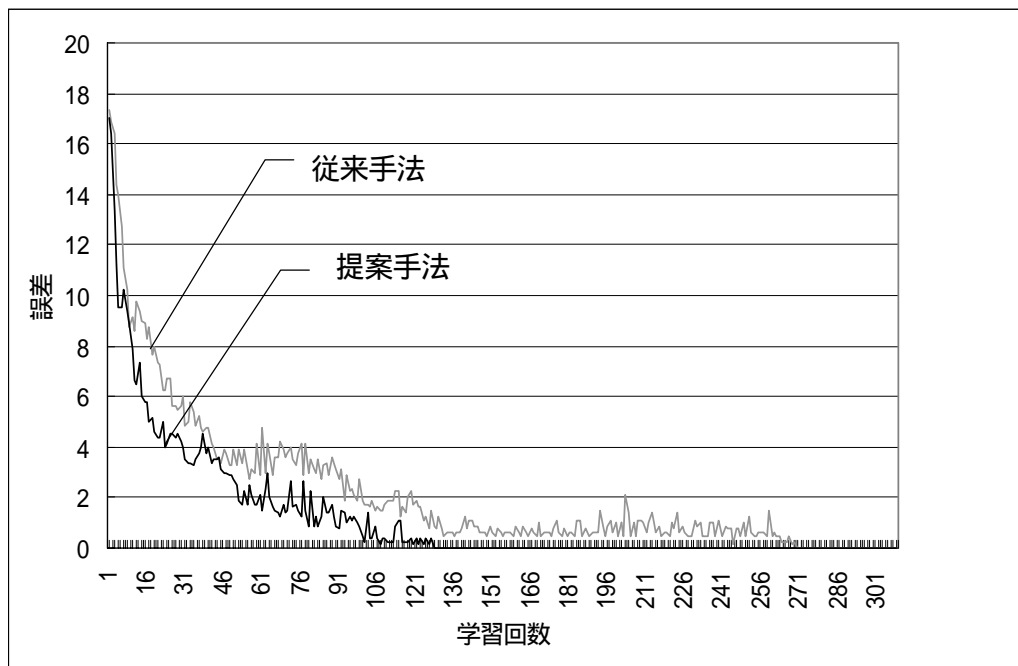


図 4.8 1 回目の試行結果

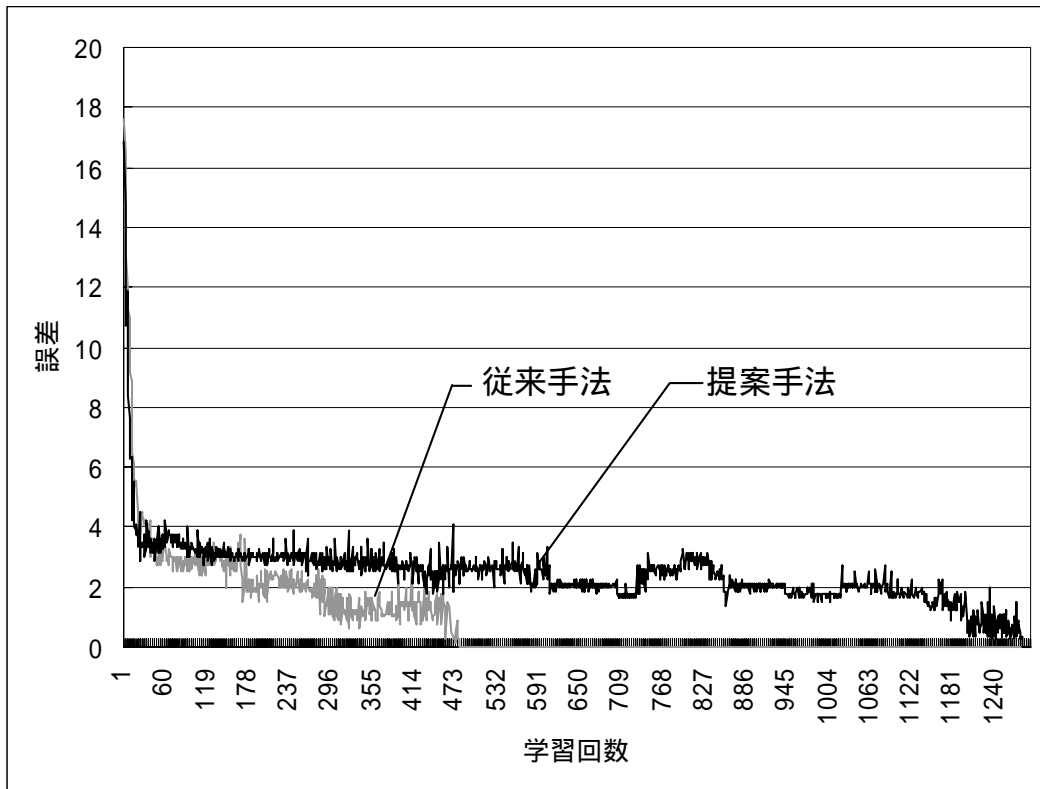


図 4.9 19 回目の試行結果

19 回目の試行では、従来手法では 477 回、提案手法は 1278 回の学習が行われた。この試行結果は、提案手法の学習回数が従来手法より多かったものの中で、学習回数差が最大であったものである。これは、閾値  $th'$  の値や結合重みの初期値、あるいは重みの更新量によって、出力が振動したためであると思われる。

EXOR の学習実験で提案手法は学習時間短縮に有効であることは示されたが、この実験で、さらに規模の大きなネットワークにおいても学習時間短縮の有効性を示している。

## 第 5 章

### VHDL による設計

提案手法を備えたニューラルネットワークを VHDL で記述し、シミュレーションを行った。設計ツールとして ALTERA 社の MAX plus II を使い、ターゲットデバイスは EPM9320LC84-15 とした。シミュレーションは単一ニューロンによる論理和と論理積の学習を行い、これに成功した。

#### 5.1 論理シミュレーションによる学習実験

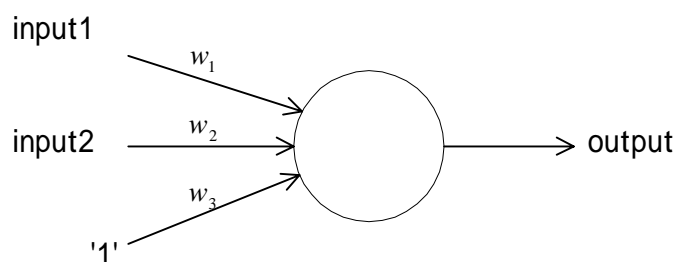


図 5.1 設計した単一ニューロン

図 5.1 に設計した単一ニューロンを示す。ニューロンの入力は input1 と input2、出力は output である。入力には常に '1' を入力するオフセット入力も備えている。このニューロンに論理和を学習させた。もし入力 input1 と input2 が '0' であったとき、 $w_1$ 、 $w_2$  の結合重みが更新されない。そこでオフセット入力を加えることで学習を進めやすくしている。

図 5.2 に動作を波形で示している。1 つの入力パターンに対し、回路はフォワードプロセスとバックワードプロセスの 2 つの状態をとる。フォワードプロセスは、回路に訓練データを入力してから出力するまでの過程であり、バックワードプロセスは、出力された値と教師信号の比較によって生成される誤差信号を結合重みに反映させる過程である。2 つのプロセスは制御回路によって管理される。フォワードプロセスの状態において、制御回路は、その状態であることを知らせる信号が '1' とする ( $\text{fwd} = '1'$ )。計算が終了すれば、制御回路はその

状態を終了させる信号を受け取り (fwd\_end = '1')、状態を遷移する。バックワード時も同様に、バックワードプロセスを知らせる信号線が '1' (back = '1') のとき、誤差に応じたパルス ( $\pm\theta$ ) によって結合重みを更新し、終了した時点で制御回路は終了信号を受け取る (back\_end = '1')。このような動作を学習終了まで繰り返す。

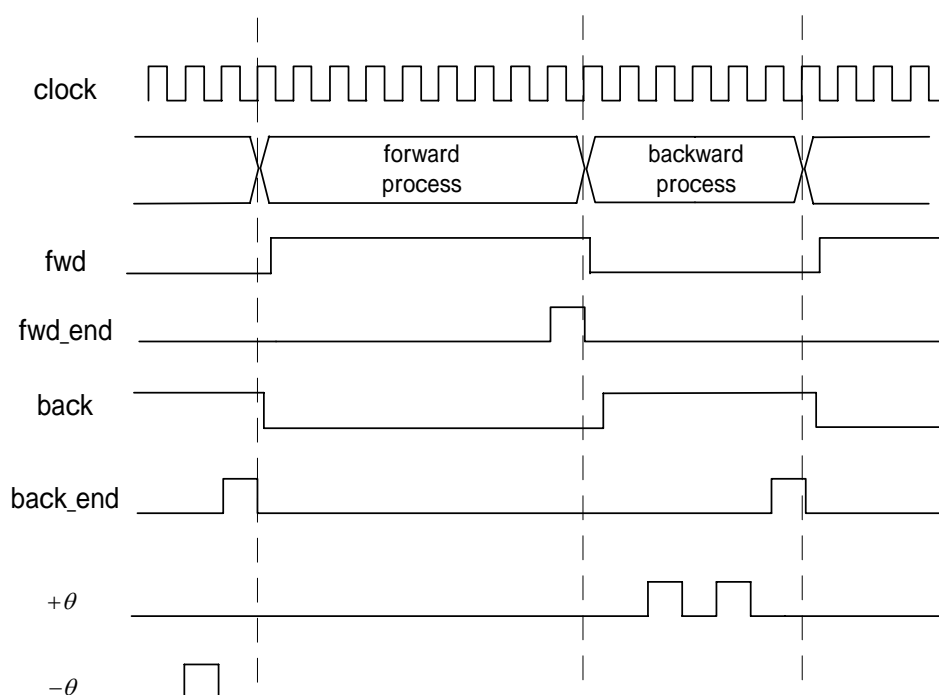


図 5.2 学習の動作

図 5.3 に回路のブロック図を示した。回路の入力には、input1,2 のほかに教師信号 (training\_data) とクロック (clock)、リセット (reset) 信号が付加されている。出力は one、half である。回路は、シナプスユニット (synapse unit)、ニューロンユニット (neuron unit)、バックワードプロセス制御回路 (backward\_control)、システム制御回路 (system\_control) からなる。外部入力 (input1, input2, clock, reset, training\_data) と外部出力 (one, half) を接続したものを図 5.4 に示す。Training\_data は教師信号である。入力 input1 と input2 はそれぞれシナプスユニット 1、2 に入力されている。シナプスユニットからの出力はニューロンユニットの入力 (input\_a, b, c) へ接続される。これらニューロンユニットとシナプスユニットは、制御回路からの信号 fwd によってフォワードプロセス処理を行う。

次にバックワードプロセスとなるが、そのときの信号の経路を加えたものを図 5.5 に示す。ニューロンユニットから出力された信号(one, half)は、バックワードプロセス制御回路へ入力され、教師信号と比較される。その結果の誤差がパルスとなって信号線 $\pm\theta$ から出力される。バックワードプロセス制御回路から出力されたパルスは、ニューロンユニットへ入力されて誤差信号  $e$  を出力する。この  $e$  がシナプスユニットへ入力されて、その内部にあるアップダウンカウンタを動作させる。

図 5.6 は MAXplusII によって論理和の学習をシミュレーションした波形である。フォワードプロセスとバックワードプロセスが交互に行われおり、バックワードプロセス時にはパルスが送出されている。結合重みを表すアップダウンカウンタは 12 ビットとし、データ形式は固定小数点で上位 6 ビットは整数値、下位 6 ビットは小数値とした。また閾値  $th=2$ 、 $th'=3$  とした。 $th'$ をこえてニューロンユニットの出力  $u$  が '1' となったときは誤差パルスを 1 個増加させてパルスを 3 個出力するようにした。デバイスは EPM9320LC84-15、クロック周波数 5MHz で、学習は約 1.55ms で終了した。また、論理積は約 2.63ms で終了した。

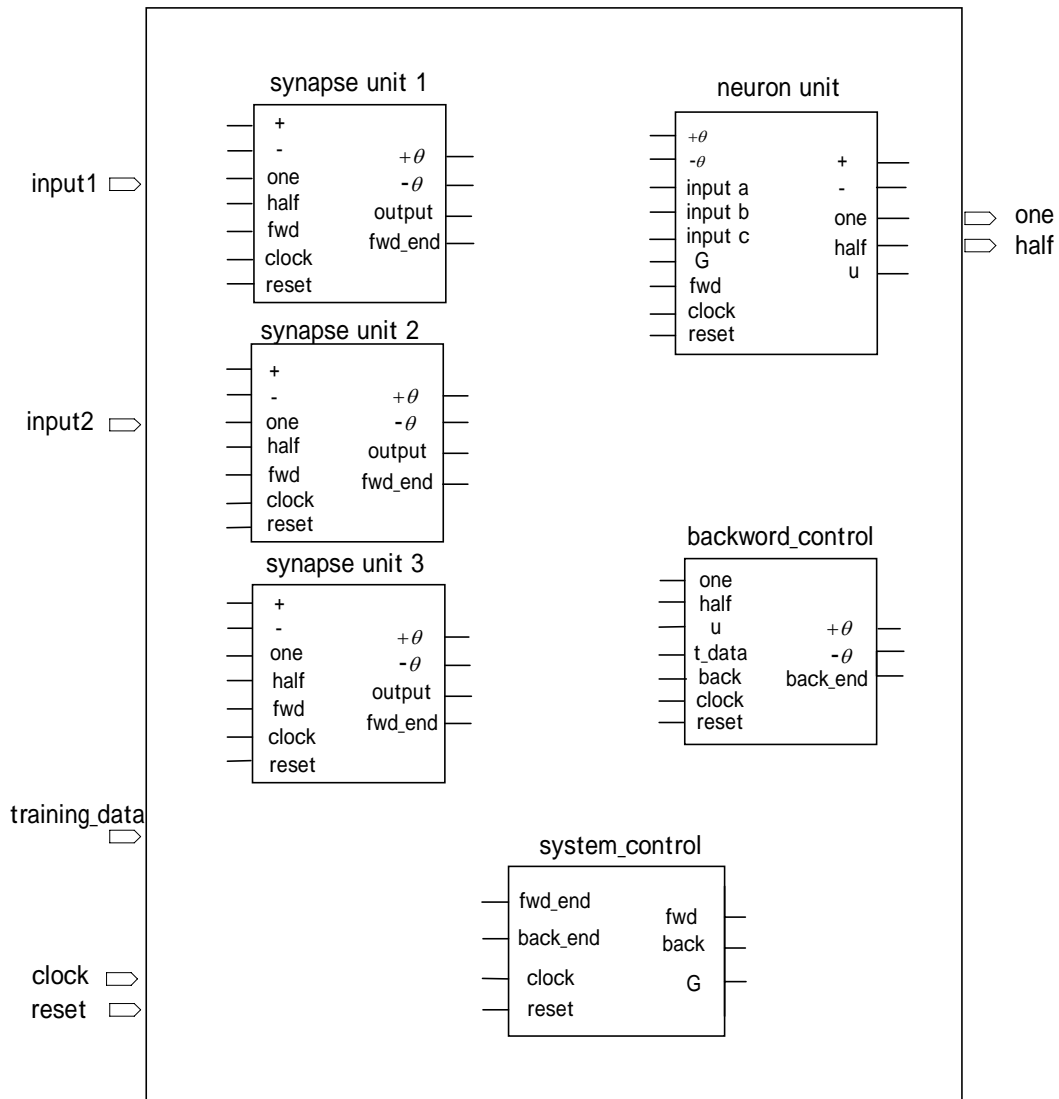


図 5.3 回路のブロック図 1

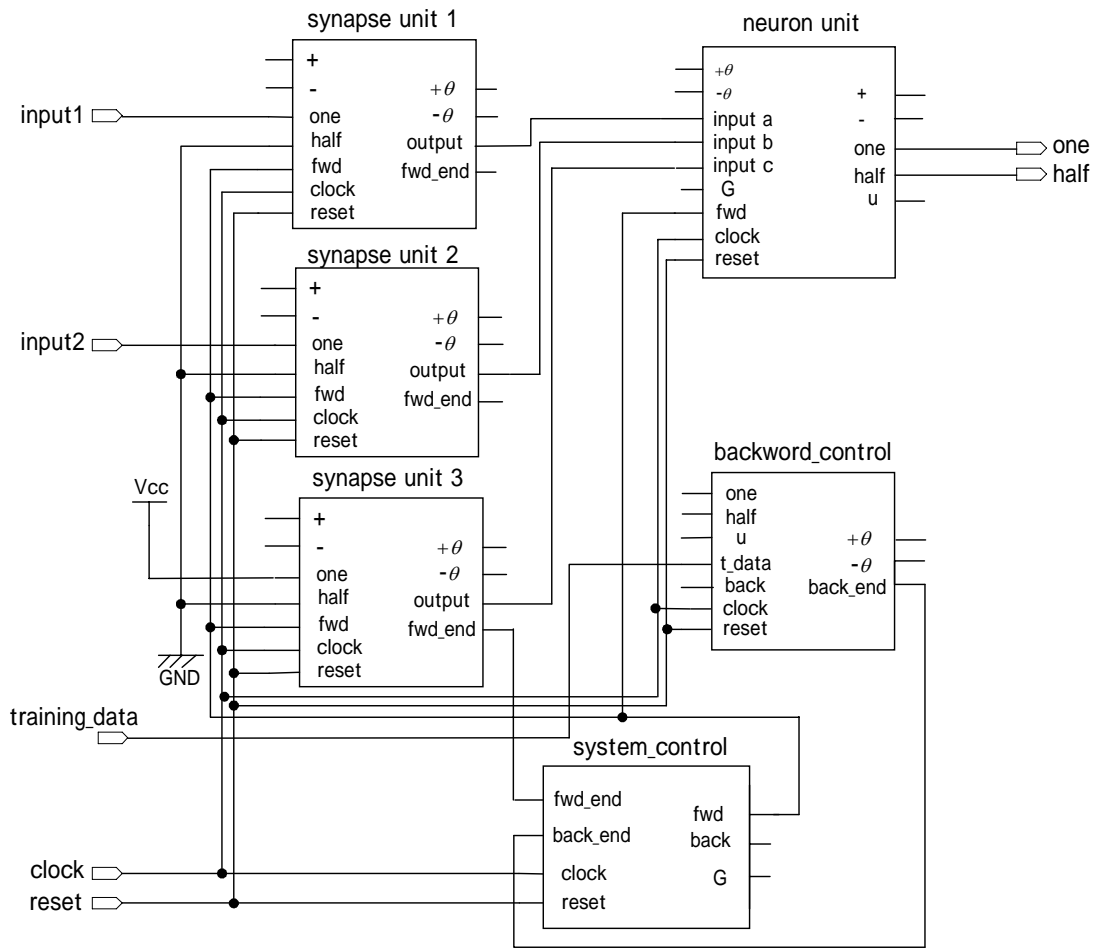


図 5.4 回路のブロック図 2

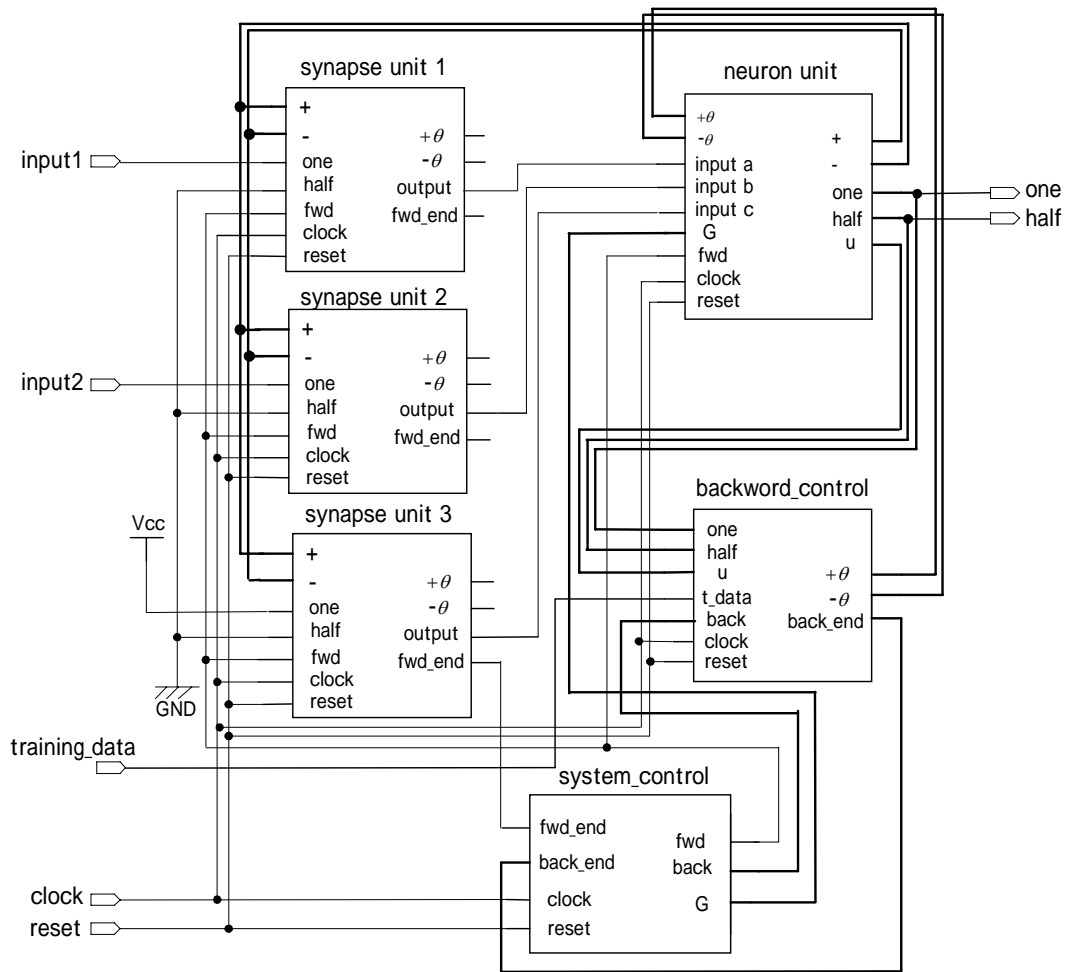


図 5.5 回路のブロック図 3



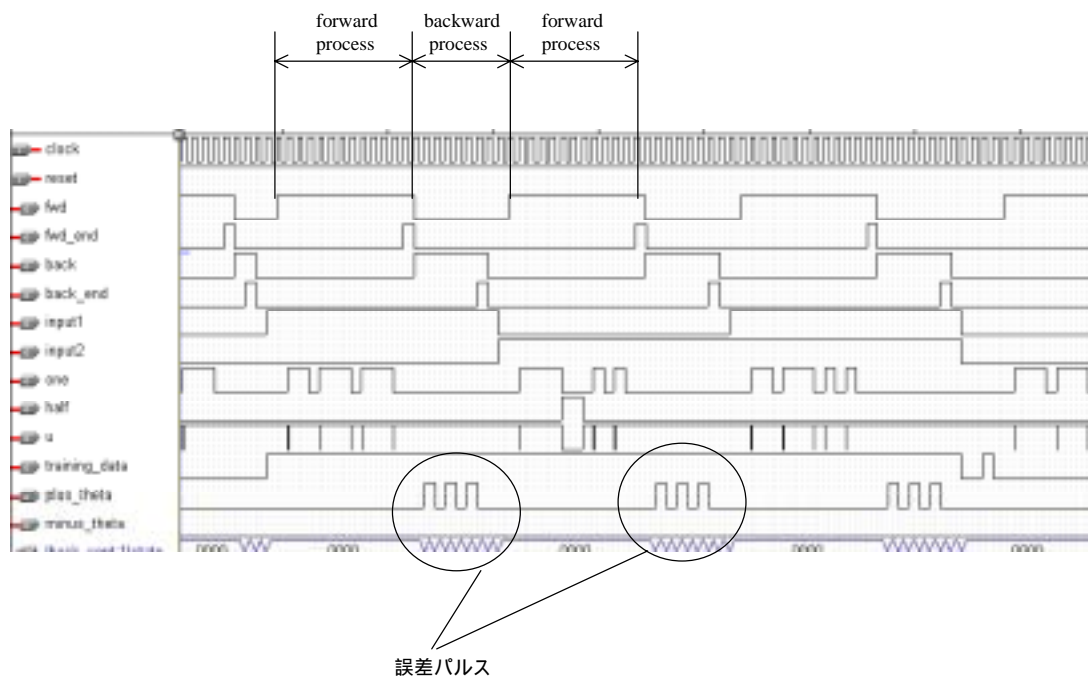


図 5.6 シミュレーション波形

## 第 6 章

### まとめ

ニューラルネットワークのハードウェア化の研究成果について、5 章にわたって述べた。

第 1 章では、研究の背景としてニューラルネットワークの歴史を紹介し、ハードウェア化についての研究動向と本研究の位置付けについて述べた。

第 2 章では、ニューロンの構造と数理モデル、そしてパーセプトロンと誤差伝播学習法について説明した。ニューロンの出力は、入力とシナプス結合重みの加重和を活性化関数の適用によって決定する。活性化関数にはシグモイド関数が用いられる。このような処理単位としてのニューロンを用いたパーセプトロンは、学習機械として注目されたが、EXOR のような線形分離不可能なものを学習によって認識することができなかった。しかし Rumelhart らによって発明された誤差逆伝播法によって、その弱点は克服された。

第 3 章では、ニューラルネットワークの新しいハードウェア化について提案した。ハードウェア化の従来手法は、活性化関数であるシグモイド関数を 3 値に単純化することで、複雑な乗算を無くした。シナプス結合重みはアップダウンカウンタで実現されており、誤差伝播学習では、結合重みの更新をカウンタの増減で行う。カウンタのビット長が小さすぎると、変化量が大きくなるが計算精度が低くなり、学習が達成されない。ビット長が大きければ、パルス当りの変化量は小さく、計算精度は上がるが、ネットワークの規模が大きくなると計算時間の増大につながる。このような従来手法の欠点を取り除き、カウンタのビット長を大きくしても計算時間の増大を抑える方法を示した。

第 4 章では、従来手法と提案手法の学習時間を比較するために行ったソフトウェアシミュレーションの結果について説明した。実験は EXOR の学習とアルファベットの学習の 2 つについて行い、学習時間短縮に有効であるという結果を得た。

第 5 章では、VHDL で提案手法を用いたハードウェアを記述した。設計は単一ニューロンによるもので、2 入力の論理和の学習を確認した。

本研究では、活性化関数を単純化したハードウェアの学習時間短縮を目的と

した。本研究の有効性はソフトウェアシミュレーションによって確認できた。しかし提案手法による学習時間は、すべての試行において短縮に成功したわけではなかった。学習時間は、結合重みの初期値と閾値に影響を受ける。これは従来手法であっても提案手法であっても同じである。結合重みの初期値と閾値は、学習時間の改善効果にばらつきが生じる原因になっていると思われる。実験結果では、提案手法による学習時間が従来手法より長くなったものがある。この原因も結合重みの初期値と提案手法に導入した閾値  $th'$  の影響であると思われる。 $th'$  を大きくすれば従来手法より学習時間が長くなることは少なくなるが、逆に時間短縮にもつながりにくいと考えられる。この  $th'$  と学習時間の短縮度についての説明は今後の課題として残されている。

## 参考文献

- [1] Warren S. McCulloch and Walter Pitts , “A logical calculus of the ideas immanent in nervous activity” , bulletin of mathematical biophysics, vol.5 , pp.115-133 , 1943
- [2] F.Rosenblatt , *Principles of Neurodynamics Perceptrons and the Theory of Brain Mechanisms*, Spartan, 1961
- [3] M.Minsky and S.Papert, *Perceptrons*, MIT Press, 1969
- [4] M.Minsky and S.Papert 著, 中野馨、坂口豊 訳 「パーセプトロン」 パーソナルメディア、1993
- [5] D.E.Rumelhart, J.L.McClelland, and PDP Reseach Group, *Parallel Distributed Processing*, MIT Press, 1986
- [6] D.E.Rumelhart, G.E.Hinton and R.J.Williams, “Learning representations by back-propagating errors”, NATURE, vol.323, pp.533-537, October 1986
- [7] M.Spitzer 著、村井俊哉、山岸洋 訳「回路網のなかの精神」 新曜社 pp.34-35 、2001
- [8] 武藤佳恭、斎藤孝之「ニューラルコンピューティング」 共立出版社 pp.281-294 、2001
- [9] H.Hikawa and K.Sato “Multilayer neural network with threshold neurons”, IEICE Trans. Fundamentals, vol.E81-A, no.6, pp.1105-1113, 1998
- [10] 肥川宏臣 “ハードウェア化に適した学習機能付き 3 値多層ニューラルネットワーク”、電子情報通信学会論文誌 D-II, vol.J81-D-II, no.12, pp.2811-2818, 1998
- [11] 黒柳奨、岩田彰 “パルスニューロンモデルのための教師あり学習則”、信学技報 NC97-151, pp.95-102, 1998
- [12] 田中愛久、黒柳奨、岩田彰 “FPGA のためのニューラルネットワークのハードウェア化手法” 信学技報 NC2000-179, pp.175-182, 2001
- [13] 川島毅、石黒章夫、大熊繁 “小規模回路で実現可能なニューラルネットワークのハードウェア化手法”、信学技報 NC99-90, 2000
- [14] 平井有三、落合辰男、安永守利 “1000 ニューロン 100 万シナプスで構成されたニューラルネットワークハードウェアシステム”、電子情報通信学会論文誌 D-II, vol.J84-D-II, no.6, pp.1185-1193, 2001

- [15] 江口裕俊、古田俊之、堀口浩幸、樗木杉高 “学習機能を有するパルス密度型ニューロンモデルとそのハード化”、電子情報通信学会論文誌 C-II, vol.J74-C-II, no.5, pp.369-376, 1991
- [16] Tom Baker and Dan Hammerstrom , “Characterization of Artificial Neural Network Algorithms”, Proc. of IEEE ISCAS'89, pp.78-81, 1989
- [17] CA Anderson , Ph.D. Dissertation, “Learning and Problem Solving with Multilayer Connectionist Systems”, Amherst, MA, Sept. 1986
- [18] 木村知史、矢野政顕 “学習率変動ハードウェアニューロンモデル”、電気関係学会四国支部連合大会講演論文集, pp.14, 2002

## 謝辞

本研究の過程において、御指導戴いた高知工科大学電子・光システム工学科矢野政顯教授に感謝致します。また適宜御助言戴いた電子・光システム工学科原央教授、橘昌良助教授に厚く御礼申し上げます。さらに矢野研究室の中村基継氏、石川純平氏、そして松村暢也氏に感謝致します。