

センシング情報処理機構に関する基礎的検討 ～ニューラルネットワークによる学習的情報処理～

平成23年度卒業論文

千葉大学工学部都市環境システム学科

指導教員：塩田 茂雄 教授

08T0276S

橋本 崇樹

目次

第 1 章 序論

1. 1	はじめに	1
1. 2	概要	1

第 2 章 バイナリセンサ

2. 1	センサノード	2
2. 2	センサネットワーク	2
2. 3	バイナリセンサ	3

第 3 章 ニューラルネットワークによる対象物輪郭推定

3. 1	ニューラルネットワーク	4
3. 1. 1	生体のニューラルネットワーク	4
3. 1. 2	ニューロンの工学的モデル	5
3. 1. 3	ネットワークの形態	11
3. 2	バックプロパゲーション（誤差逆伝搬）法	12
3. 3	検討の概要	17

第 4 章 シミュレーション評価

4. 1	概要	18
4. 2	シミュレーション条件	18
4. 3	シミュレーション結果	21
4. 3. 1	ケース 1	21
4. 3. 2	ケース 2	23
4. 3. 3	ケース 3	25
4. 3. 4	ケース 4	27
4. 3. 5	ケース 5	29
4. 3. 6	ケース 6	31

第 5 章 考察

5. 1	まとめ	32
------	-----	----

第1章

序論

1.1 はじめに

ある領域を通過する物体の輪郭を知るのに、高機能なセンサであれば高度な通信機能や情報分析（大きさや形だけでなく、色、温度、速度など）能力、耐久性など良い条件で検知できるだろう。しかし、低機能だが安価なセンサを用いても物体の輪郭を知るくらしいのことはできないだろうか。

例えば、人間が足を踏み入れ難い場所（監視領域）にセンサを撒いておき、通過する物体の輪郭だけでも知れば良い、という要求ならば安価なセンサであれば可能であろうし、また、わざわざ危険を冒して回収に向かわなくてもいいだろう。そういった限定的な用途に限れば高機能で高価なセンサよりは低機能だが安価なセンサが実用的ではないだろうか。そういった考えにより本研究を行うに至った。

1.2 概要

ある特定の監視領域を通過する対象物に関する情報を得る方法として、様々な手法が考えられる。例えば、赤外線モニタを搭載した監視装置や、レーダやソナーを用いた画像解析、デジタル画像を基に人の顔を認識する顔認証システム、複数のセンサにGPSを搭載させ位置情報から割り出すもの等々、色々な手法が考えられる。しかし、より安価な手法としてバイナリセンサとニューラルネットワークを用いた対象物の形状推定を検討した。

ある特定の監視領域に多数のバイナリセンサを撒き、対象物が通過した際のバイナリセンサの反応から、積分幾何学の公式に基づいて対象物の周囲長・面積を推定する手法が提案されている[1]。本研究では、バイナリセンサの反応パターンと対象物の輪郭との関係性を幾つかの教師信号を用いてニューラルネットワークにバックプロパゲーション（誤差逆伝搬）法による学習をさせ、積分幾何学の公式を用いることなく、未知の対象物の輪郭を推定することが可能か否かについて基礎的検討を実施した。

第2章

バイナリセンサ

2.1 センサノード

センサノードとは、ノードの周囲の状況を感知し、そのデータになんらかの加工を加えネットワークを経由し情報を伝えるための端末である。一般的には、この検出する部分であるセンサ自身と検出したデータを加工したり他の処理を指令したりするプロセッサ、その結果得られて情報をネットワークへ流すための通信モジュール、また動作するための電源が備わっている。以下に例を示す。

センサが感知するために用いる対象は様々である。

対象：光／温度／音／磁気／ガス／速度／加速度／位置／距離

光センサ フォトダイオード 磁気センサ ガス濃度 タコジェネレータ 光位置センサ 超音波距離計 光学式測距 電磁波測距 ジャイロセンサ 加速度センサ 地震センサ 液検知センサ レーザドップラー流速計…などなど。

2.2 センサネットワーク

センサネットワークとは、センサが検出したデータをネットワーク経由で収集するシステムのことである。

センサノードは時代とともにサイズが小さくなり、価格も下がり、消費電力も低消費電力化され、また高精度化されてきている。基地局や固定通信網を持たない自動車などの移動体に搭載する利用方法が検討されている無線でのアドホックネットワークでは、センサに搭載された無線通信機が自律的にルーティングやマルチホップ通信を行う。この場合、自身が直接的に目的の受信機へ送信するよりも、最寄の（同規格の）センサに中継させる方が、消費電力は低くできる。中継するセンサ自身もまた最寄の他のセンサへ送信し中継させ目的の受信機へ向けて受け渡されて行く。

また、防災の観点から災害情報の収集や危険予測のため、線路への車両や人の侵入、火災警報器や消火設備・スプリンクラー、気象観測や、潮の満ち引き、河川の水位、大気汚染の観測、また、防犯や医療や健康管理などにも幅広く利用されている。

2.3 バイナリセンサ

バイナリセンサとは、1ビット信号（対象物を検知すれば1，しなければ0）のみを通知できるが，アドホックのように自身がルータとなってマルチホップ通信を行う機能やGPSのような測位機能を持たない低機能（つまり対象物の検出のみ可能）なセンサのことである．複数のゲートウェイを広域に配置し，センサといずれか1つのゲートウェイを狭帯域ではあるが長距離（数 km）の無線リンクにより1ホップで接続するユビキタスネットワーク用のインフラストラクチャの概念（広域ユビキタスネットワーク）が提案されており[2]，このようなインフラストラクチャのもとでは，多数のバイナリセンサを監視領域に配置し，各バイナリセンサの信号をゲートウェイ（ゲートウェイと1対1の対応）経由で遠隔収集することが可能となる．低機能であるため，安価であり多数のバイナリセンサを監視領域に撒くことができるという利点がある．

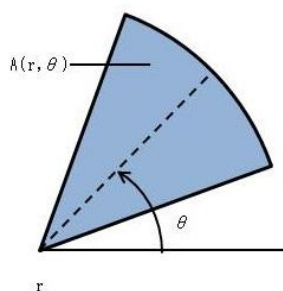


図1：センシングエリアの例.

図1に示したように，バイナリセンサのセンシングエリアを平面で捉えた場合，図のような扇型になる．本研究でシミュレーションに用いるバイナリセンサはこの半径を 2π ，つまりセンシングエリアを円形としている．このセンシングエリアに対象物の輪郭が触れただけでもバイナリセンサはこれを感知するものとする．

第3章

ニューラルネットワークによる対象物輪郭推定

3.1 ニューラルネットワーク

3.1.1 生体のニューラルネットワーク

生体のニューラルネットワーク（神経回路網）では、神経細胞（ニューロン）が他の細胞からの刺激により細胞内電位を変化させ、繋がり（実際は僅かの隙間がある）のある他の神経細胞へ向けて伝達物質（化学物質）を放出し通信する。この伝達物質を（化学的作用として）受け取った神経細胞もまた細胞内電位を変化させる。この細胞内電位の変化がある値（閾値）を越えると興奮状態となり、細胞内電位が急激な立ち上がりを見せる。この急激な立ち上がりはインパルス（速度は数百 m/s 程度）と呼ばれる現象で、これが細胞内を流れ神経細胞の先端に到達すると、先端から先程の伝達物質が放出される仕組みである。

また、神経細胞の興奮状態は on (1) か off (0) の2値であり、他の多数の神経細胞から伝達物質を受け取っていたとしても、閾値を越えなければ伝達物質を放出しない。つまり、他の細胞へ影響を及ぼすか及ぼさないかである。

なお、細胞同士の作用には2種類あり、細胞内電位を高める働きをする興奮性と、細胞内電位を抑える働きをする抑制性がある。

このような性質は、おおよそその細胞に当てはまる。ただし、例外もある。網膜中の水平細胞は細胞内電位を他の細胞にそのまま伝達するため、先程の2値のみではなく中間値も存在する。

上記は信号の伝達のメカニズムであるが、知覚したものからの学習のメカニズムが脳内には存在する。

人間は、ある一連の動作を行うとき、それが初めての場合は、自分の手や足など体の動きを意識して行う。動作に慣れ始めると、無意識にその動作を行えるようになる。ここで大脳は視覚、聴覚、触覚等により外界からの情報を動作の調整のために用いる。小脳は、この大脳の働きから動作に対する制御信号の規則性を次第に学習する。小脳の学習が完了すると、大脳が行う制御を小脳が引き受けるようになり、大脳はこの制御から解放される。

つまり、大脳が小脳の教師であると言える。

3.1.2 ニューロンの工学的モデル

ニューロンの工学的モデルにおいては、ニューロン（神経細胞）に代わる人工の素子で構成される。生体のニューロンはその構成や働きは複雑であるが、工学的モデルでは、必要な機能を抽出し、これを単純化したものである。単なる模倣ではない。現在の工学的モデルは以下の2つの項目により分類される。

- (1) 素子の動作 決定的 / 確率的
- (2) 出力値 2値(0, 1) / アナログ

(1)の「素子の動作が決定的」ということは、入力値と出力値の関係が常に一意であるということである。表1にこの関係の例を示す。この例の場合、入力値が0のとき出力値は必ず1になり、入力値が1のとき出力値は必ず0となる。常にこのような関係にある素子を決定的素子と呼ぶ。

表1：決定的素子の入出力関係の例。

入力	出力
0	1
1	0

次に確率的素子の例を表2に示す。「素子の動作が確率的」ということは、この場合、入力値が0のとき出力値は0かも知れないし1かも知れない。また入力値が1のときでも出力値は0かも知れないし1かも知れない。入力値が0のとき出力値が0になる確率は、出力値が1になる確率よりも高い。

表2：確率的素子の入出力関係の例。

入力	出力=0の確率	出力=1の確率
0	0.7	0.3
1	0.2	0.8

また、入力値が1のとき出力値が0になるよりも出力値が1になる確率の方が高い。しかし、出力値の結果は偶然により決まるため、確率が低い方になることもある。つまり、入力値と出力値の関係が一意に決まらないのである。このような素子を確率的素子と呼ぶ。

確率的素子の入出力関係において、入力値が0のときの出力値が0の確率が1で出力値が1の確率が0、入力値が1のときの出力値が0の確率が0で出力値が1の確率が1という関係は決定的であり、決定的素子は確率的素子の特別な状態であると言える。

(2)の「出力値が2値」とは、生体のニューロンは他の細胞等へ興奮するか興奮しないかの作用しか及ぼさない。中間的作用は及ぼさないのである。この2通りの作用をモデル化したものが2値モデルと呼ばれる。

また「出力値がアナログ」とは、ニューロン内のインパルスの発生頻度の観点から言う。つまり、単位時間当たりの興奮状態になる回数を出力値と捉えることである。ただし、ニューロン自体は外的要因によらず自発的に興奮状態になることもある。これを含めて考えると、細胞内電位 (s) とインパルス頻度 (y) は図1に示されるようにS字のグラフとなる。細胞内電位が閾値 (θ) に達していない状態でも他の細胞には、ある程度の影響を及ぼす。これが中間的作用と言えるものになる。ここで、この作用は頻度であって、インパルスそのものは依然として0か1かの法則によって成り立っている。

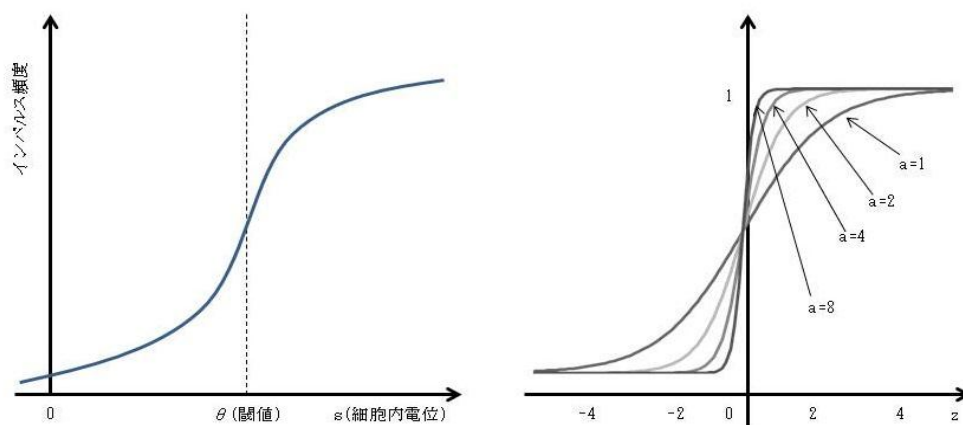


図2：ニューロンの細胞内電位とインパルス発生頻度の関係。

図2に表されたS字のグラフはシグモイド関数と呼ばれる飽和関数である。この曲線の下になる式は以下である。これはニューロンの入出力特性を表す重要な関数である。

$$y = \frac{1}{1 + e^{-\alpha z}}$$

閾値 θ を考慮した場合には以下の式になる。

$$y = \frac{1}{1 + e^{-\alpha(z - \theta)}}$$

式中の α はゲインと呼ばれ、この値により曲線の立ち上がり具合が変化する。図3の右のグラフに表されるように、このゲインの値が変化することで曲線の立ち上がりが変化しているのが判る。

-1- 決定的 2 値モデル

ニューロンを $\{0, 1\}$ の 2 値 (興奮するかしないか) の出力を持つ決定的素子でモデル化したものが決定的 2 値モデルである.

このモデルの仕組みを説明した図を図 3 に示す.

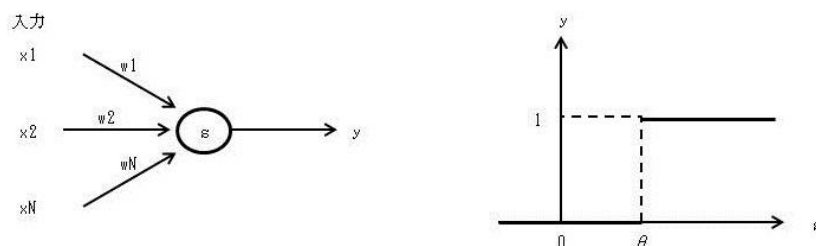


図 3 : 決定的 2 値モデル.

他の素子, または外部からこの素子へ $\{x_1, x_2, \dots, x_N\}$ の信号が入力されたとして, これらに $\{w_1, w_2, \dots, w_N\}$ の各重みを掛け加えて s を得る. この s は細胞内電位に相当する. $\{w_1, w_2, \dots, w_N\}$ の各重みはシナプス結線重みに相当する. 細胞内電位 s が閾値 θ 以上になった場合を出力 y を 1 とする. 細胞内電位 s が閾値 θ 未満の場合は出力 y を 0 とする. これは以下の式で表される.

$$s = \sum_{i=1}^N w_i x_i$$

$$y = \begin{cases} 1 & s \geq \theta \\ 0 & s < \theta \end{cases}$$

-2- 決定的アナログモデル

ある入力からの信号に対し出力 y を $0 < y < 1$ の範囲のアナログ値に決定的に定めるモデルが決定的アナログモデルである.

このモデルでも同様に, 他の素子, または外部からこの素子へ $\{x_1, x_2, \dots, x_N\}$ の信号が入力されたとして, これらに $\{w_1, w_2, \dots, w_N\}$ の各重みを掛け加えて s を得る. ここまでは同じだが, s から y を求める際にシグモイド関数に s と θ を与え求める. シグモイド関数内に表記されている α がゲインと呼ばれるもので, このゲインを大きくすればシグモイド関数は出力を $\{0, 1\}$ の値しか取らないステップ関数に近づいていく. つまり, アナログモデルはゲインを十分大きな値 (極限值) にすると, 2 値モデルとなる. これらを図 5 と式に表す.

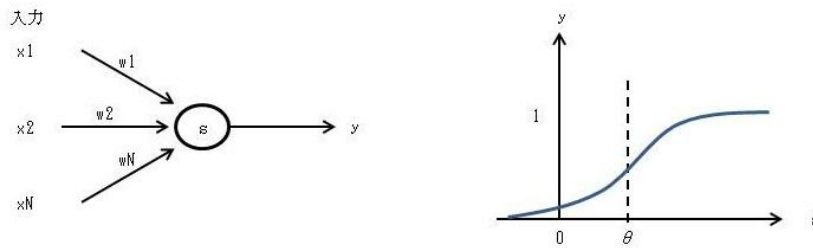


図 4 : 決定的アナログモデル.

$$s = \sum_{i=1}^N w_i x_i$$

$$y = \begin{cases} 1 & s \geq \theta \\ 0 & s < \theta \end{cases}$$

-3- 確率的 2 値モデル

次に出力 y が $\{0, 1\}$ の 2 値を確率的に取るように以下の式で定められる. この場合, 入力と同じ値でも出力は異なる値を取ることがある. 細胞内電位 s を入力値に重みを掛け加える計算を行い, 次に s に対しシグモイド関数より p を求める. この p とは, 出力 y が p の確率で 1 になり $1-p$ の確率で 0 となるものである.

$$s = \sum_{i=1}^N w_i x_i$$

$$p = \frac{1}{1 + e^{-s}}$$

$$y = 1 \text{ (確率 } p)$$

$$y = 0 \text{ (確率 } 1 - p)$$

s と確率 p の関係を図 5 のグラフに示す.

細胞内電位 s が高くなると, y が 1 になる確率は高くなっていく. 実際, 生体のニューロンは自発的に (細胞内電位の揺らぎにより) 興奮することがある. この傾向がこのモデルでも言えるのである.

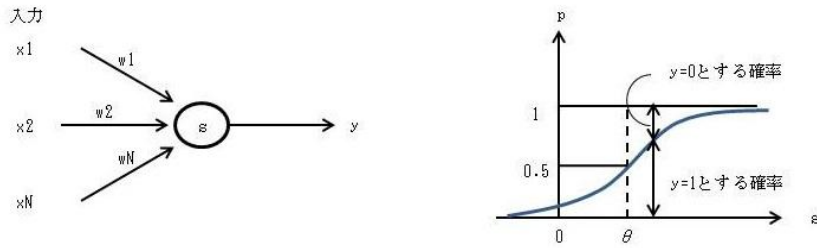


図5：確率的2値モデル.

厳密にはパソコンやワークステーションなども決定的機械であるため、本当の意味での確率的な試行は行えない。そのため、先に説明したような計算も行えない。ただ、パソコンやワークステーションなどには疑似乱数と呼ばれるものがあるので、これを用いて近似させ計算することができる。以下に疑似乱数を用いて確率 p で出力 y を 1 とする方法を C 言語の場合で説明する。

C 言語で疑似乱数を生成させる発生させる関数は `rand()` である。この関数は 0 から `RAND_MAX` の範囲の整数を等確率（等頻度）で不規則に生成させる。ただし、`RAND_MAX` の値は用いる C コンパイラにより異なる。各整数の生成される確率は $1/\text{RAND_MAX}$ であり等確率である。この `rand()` 関数が頻繁に呼び出された場合、次式が成り立つのは、どのくらいの回数かを見てみる。

$$\text{rand}() < \text{RAND_MAX}(\theta - 1) \times \text{RAND_MAX_2} / \text{RAND_MAX}$$

`rand()` が生成する整数でこの式を満たすものは、0 から $\text{RAND_MAX}(\theta - 1) \times \text{RAND_MAX_2} / \text{RAND_MAX}$ の間にあり、個数は $\text{RAND_MAX}(\theta - 1) \times \text{RAND_MAX_2} / \text{RAND_MAX}$ 個である。また、それぞれの出現確率は $1/\text{RAND_MAX}$ である。よって確率 p は次式である。

$$\begin{aligned} \theta &= \text{RAND_MAX}(\theta - 1) \times \text{RAND_MAX_2} / \text{RAND_MAX} \times (1/\text{RAND_MAX_2}) \\ &= \text{RAND_MAX}(\theta - 1) \end{aligned}$$

つまり、`rand()` 関数を呼び出して得た値は $\text{rand}() < \text{RAND_MAX}(\theta - 1) \times \text{RAND_MAX_2} / \text{RAND_MAX}$ を満たすとき、 y の値を 1 であるとすれば、確率 $\theta = \text{RAND_MAX}(\theta - 1)$ で y を 1 とすることができる。

$$s = \frac{\text{rand}()}{\text{RAND_MAX_2}}$$

if `rand() <= RAND_MAX(\theta - 1) * RAND_MAX_2 / RAND_MAX` then $y = 1$ else $y = 0$

以上より，疑似乱数を生成する関数 rand() を用いて確率的 2 値モデルの出力を計算する方法は上記の 2 つの式のようになる．

-4- 時間発展モデル

前述の 3 つモデルがある時点での細胞内電位を固定した値を扱っていたが，このモデルは細胞内電位の時間変化を扱うものである．このモデルでは細胞内電位 s の時間変化を次に示す微分方程式で表す．

$$\frac{ds}{dt} = -s + \frac{1}{\tau} \sum_{i=1}^N w_i x_i$$

素子の出力 y を，この式に従い時間変化する細胞内電位を用いて次式で計算する．

$$y = \sigma(s - \theta)$$

先程の微分方程式は s が 0 の場合，右辺の右項の値が正ならば，左辺の値が正となり s は増加する．このとき，右辺の左項の $-s$ は負の側へと振れていく．そのため，右辺全体の値は 0 に近づくことになる．さらに次式が成り立つとき $ds/dt=0$ となり s の値は変化しなくなる．

$$s = \frac{1}{\tau} \sum_{i=1}^N w_i x_i$$

また先程の微分方程式において右辺の右項の値が負の場合，やはり s の値は変化しなくなる．よって，変化しなくなった後の素子の入出力関係は，決定的アナログモデルと同等になる．

$$\Delta s = \tau(-s + \sum_{i=1}^N w_i x_i)$$

$$s = s + \Delta s$$

コンピュータに計算させる場合には，上の 2 つの離散時間化した近似式で計算させられる．なお， ϵ は十分小さい正の数である．

3.1.3 ネットワークの形態

これまでに述べた素子を以下の図のように接続し、信号が入力から出力へ向けて一方方向に進んでいくネットワークをフィードフォワード型ニューラルネットワークという。

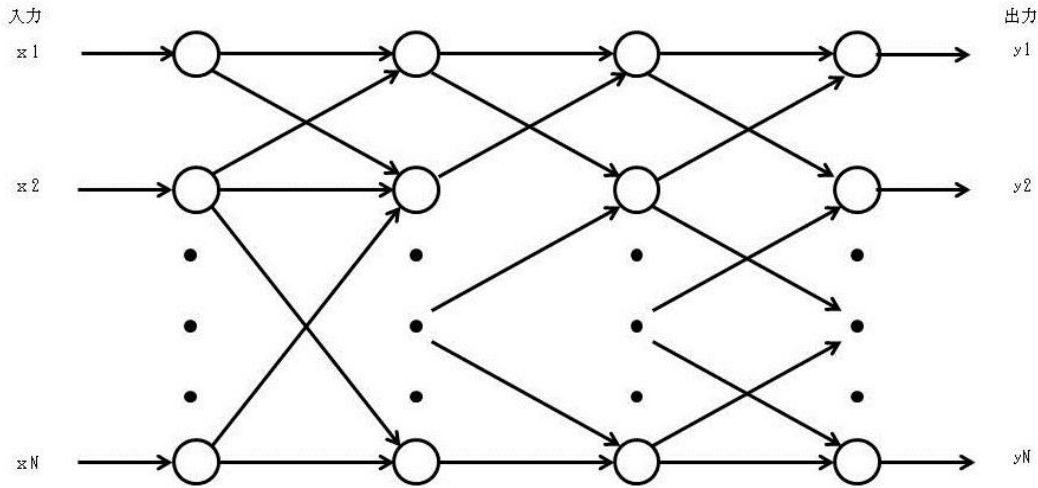


図6：フィードフォワード型ニューラルネットワーク。

入力より与えられた値（信号）が次々と出力側方向の素子へと渡され値が決まる。入力側へと信号が戻ることはない。

一方、出力側の素子から入力側の素子へ向けて信号がフィードバックしてくるネットワークを次の図に示す。このようなネットワークでは入力側の素子の値を決めるとき、出力側の素子の値を考慮する必要がある。入力側の素子と出力側の素子の依存関係から信号の流れに一種のループが起こる。しかし実際には入力よりも出力が遅れて値が定まる。これがリカレント型ニューラルネットワークである。決定的2値モデルで構成される離散時間モデルにおいて、時刻 t における素子 i の値を $s_i(t)$ とすると時刻 $t+1$ における素子の値 $s_i(t+1)$ を次式で定める。

$$s_i(t+1) = \sum_j w_{ij} s_j(t)$$

$$y_i = \begin{cases} 1 & s_i \geq \theta \\ 0 & s_i < \theta \end{cases}$$

初期値 $s_i(0)$ を適当に与えれば各時点の素子の値が順次定まる。

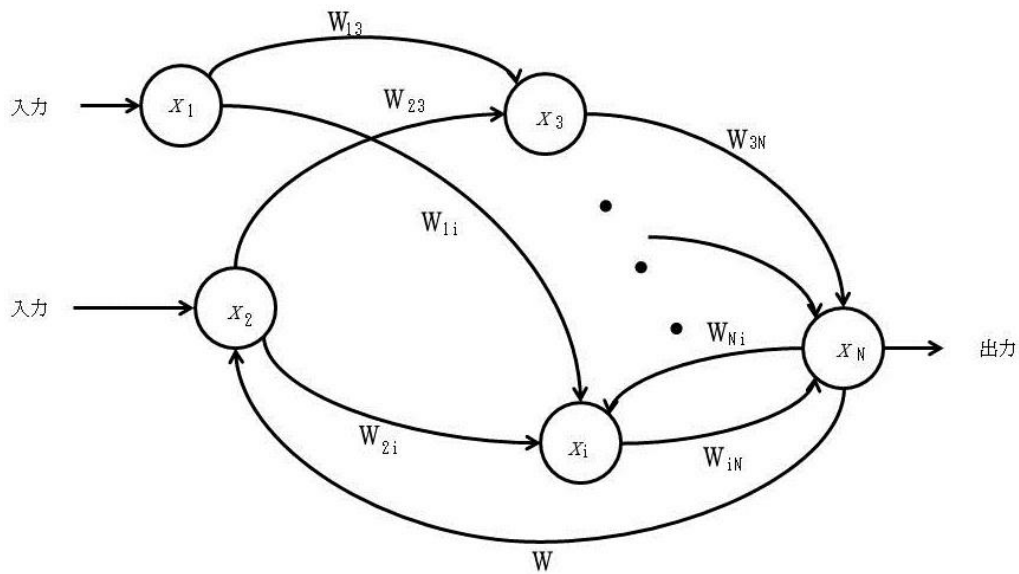


図7：リカレント型ニューラルネットワーク。

3.2 バックプロパゲーション（誤差逆伝搬）法

フィードフォワード型ニューラルネットワークの学習手法の1つである[3]。目的の関数を近似的に実現するための手段である。勾配法に基づいて、ネットワーク内部のパラメータを定める。目的の到達の度合を計るための評価尺度を用いる。これによりネットワークが目的の関数を近似する際に生じる誤差を計る。ネットワーク内部のパラメータに適切な初期値を与えた上で、誤差尺度を小さくするように、パラメータの微小修正を繰り返す。逐次更新学習法の1つであり、訓練データが与えられる都度、結線重みを微小修正する。ただし、あくまで目的の関数の近似であって、必ず目的の関数を満たすというものではない。

内部に可変なパラメータを複数持ち、これらを変化させることにより様々な入出力関係を実現するシステムの簡易図を図8に示した。このシステムが線形である場合は、線形代数を用いて逆行列を解くことで直接パラメータを指定することができた。

ところが、非線形システムの場合、関数を近似しようとしてもパラメータを求める線形代数のような理論は存在しない。そのため、最適解が得られる保証はないが、勾配法による繰り返し計算が行われてきた。しかし、評価関数の偏微分の計算ですら困難な場合が多い。よって非線形システムは単純なものが使われてきた。

バックプロパゲーション法は、複雑かつ高度な写像能力を持つ非線形システムを用いて関数近似を行う初の実用的な手法であるとされている。

次にそのアルゴリズムについて、以下の3つのステップに分けられる。これらのステップを全訓練データの誤差評価尺度が十分小さくなるまで行う。

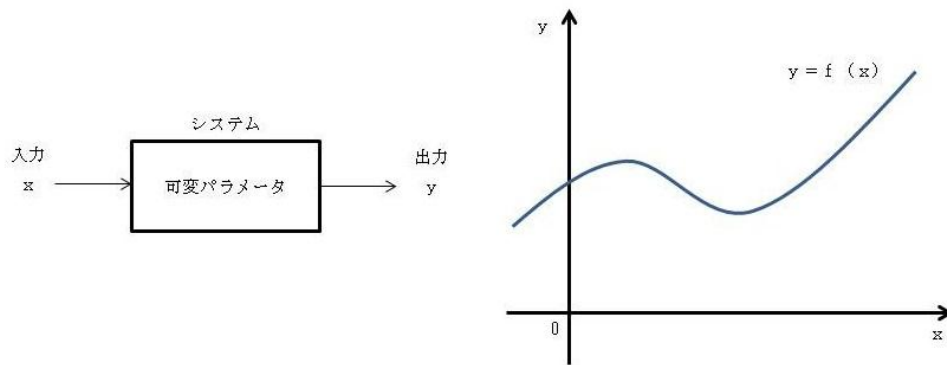


図 8：関数近似の枠組.

Step1 訓練データの選択

図 9 の内側ループにおいて、最初に l 番目の訓練データを選択する．この訓練データに対し以下の 2 つのステップを実行する．なお右肩の添え字は第何層かを表す．

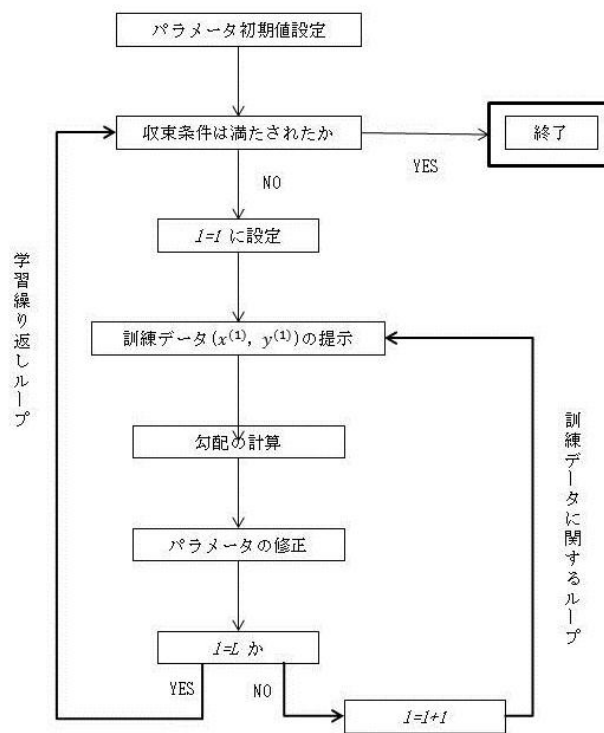


図 9：逐次更新学習法のアルゴリズム.

Step2 出力の計算

3 層から成るフィードフォワード型ニューラルネットワークにおいて、中間層と出力層の各素子は演算機能を持つ．これらの素子の動作は次の式で定式化される．

$$s = \sum_{j=1}^K w_{ij}^{(1)} x_j^{(1)}$$

$$x_i^{(2)} = \sigma(s)$$

このように動作する素子を組み合わせ、フィードフォワード型ニューラルネットワークの入力層の i 番目のノードに入力 $x_i^{(1)}$ を加える。入力ノードは $i = 1, 2, \dots, N$ の N 個あるものとし、これらの全てに同時に入力を加える。

ここで、ネットワーク内の結線重みは予め初期値として乱数で発生させた値を割り当てておく。初期値は必ず 0 以外でなければならない。対称性のため勾配値が 0 となる。その場合、学習手続きは進行しない。初期値を乱数で与えるのは、この対称性を壊すためである。

入力ノードに入力した後、 j 番目の中間層素子は、入力 $x_i^{(1)}$ に結線重み $w_{ij}^{(1)}$ を掛けて $i = 1, 2, \dots, N$ について総和し、次の値を得る。これを重み付け総和と呼ぶ。

$$s_j^{(2)} = \sum_{i=1}^N w_{ij}^{(1)} x_i^{(1)}$$

次に $s_j^{(2)}$ をシグモイド関数に通して、素子の出力の次式を得る。

$$x_j^{(2)} = \sigma(s_j^{(2)})$$

中間層素子が $j = 1, 2, \dots, K$ の K 個あるとき、これらの全てについて上述の手続きにより出力を求める。このように中間層素子の出力を求めた後、 j 番目の出力層素子は、

中間層素子の出力 $x_j^{(2)}$ 、 $i = 1, 2, \dots, K$ をその入力として受け取り、これに結線の重みをかけて総和し、次式を得る。

$$s_i^{(3)} = \sum_{j=1}^K w_{ij}^{(2)} x_j^{(2)}$$

続いて $s_i^{(3)}$ をシグモイド関数に通して次式を得る。

$$x_i^{(3)} = \sigma(s_i^{(3)})$$

出力層素子が $j = 1, 2, \dots, M$ の M 個あるとき、これらの全てについて上述の手続きにより出力を求める。

以上の手続きでネットワークの出力を計算した後、これらを学習目標（教師信号）の出力と比較する。初回は重みがデタラメな値であるため一致しない。そこで勾配法の原理により次式の誤差評価尺度 E を小さくするようにネットワーク内部の結線重みを修正する。なお、閾値は結線の1つとして表されている。また以上で求めた各素子の出力値は次の Step で利用するため、各素子に記憶させておく。

$$E = \sum_{i=1}^n |y_i - t_i|^2$$

Step3 結合重みの修正

結線重みの修正について示す。

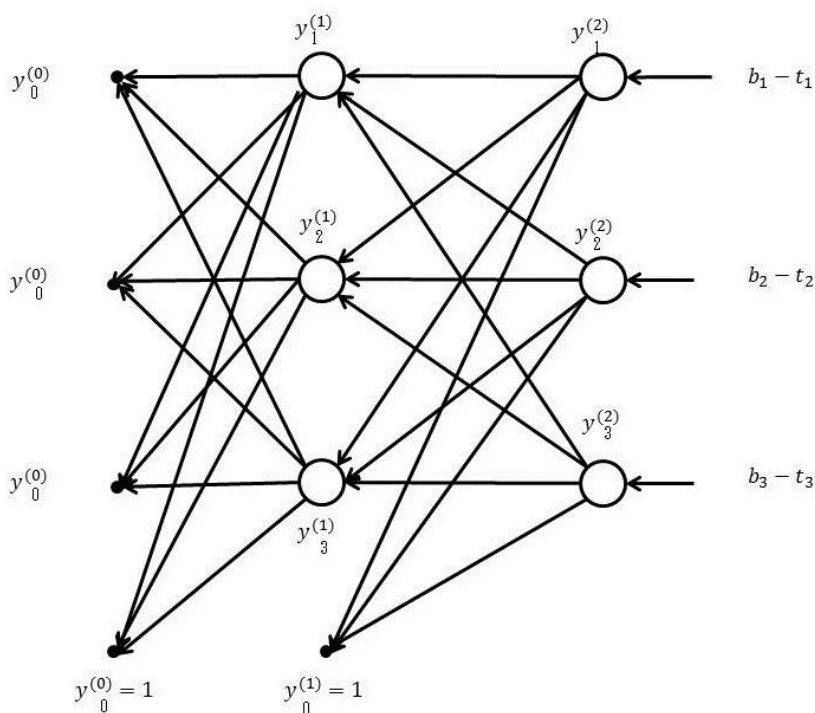


図 10：誤差逆伝搬モードの信号の流れ。

結線重み修正時の（誤差逆伝搬モード）のネットワークの信号の流れを図 10 に示す。ノード数、素子数、素子間の結線構造、および結線の重みは出力の計算時（出力モード）と変わらないが、素子の機能と信号の流れる方向が異なる。

誤差逆伝搬モードにおける素子機能の図を以下に示す。

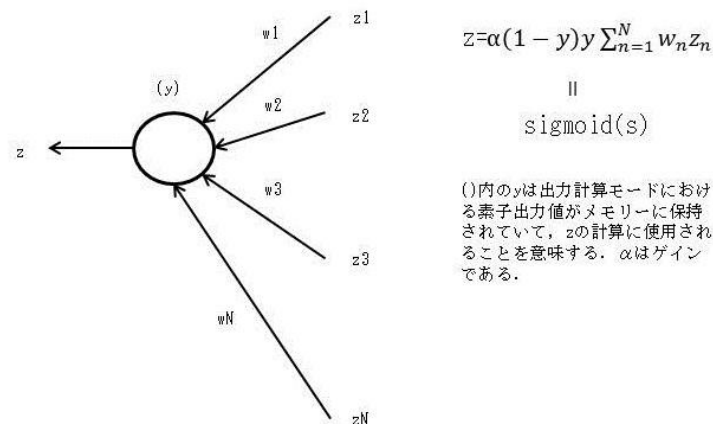


図 11：誤差逆伝搬モードの素子機能.

素子内部にはメモリーがあり，Step2 で求めた自分の出力を保持しておく．図中の () 内の値は，Step2 で求めた値が保持されていることを意味する．素子の右側から与えられる入力に素子の右側の結線の重みをかけ，重み付け総和

$$z = \sum_{n=1}^N w_n z_n$$

を求めた後，メモリーに保持されている y を用いて $\alpha(1-y)y$ を計算し，これに u を掛け

$$z = \alpha(1-y)y u$$

を出力する．これは左側の素子への入力となる．ここで $\alpha(1-y)y$ は $\frac{dz}{dy} = \alpha(1-y)y$ のときに $\frac{dz}{dy} = \alpha(1-y)y$ なる関係でシグモイド関数の微分が求まることから出てきたものである．

図 10 において出力と目標出力の誤差を加える．この入力に基づき各層で説明してきた素子機能が働き，素子出力が右側の層から次々と定まる．こうして，誤差逆伝搬モードで各層の素子出力値を求めた後，これと出力計算モードで求めた各層の素子出力を用いて次式で結線重みを修正するのである．

$$\Delta w_{ij} = \Delta z_j - \Delta z_i w_{ij}$$

3.3 検討の概要

バイナリセンサが多数配置された監視領域を対象物が通過するとき、バイナリセンサの反応の仕方（どのバイナリセンサが同時に1の信号を通知したか）と対象物の通過経路や輪郭との間には明確な関係が存在するはずである。輪郭が既知の少数個の対象物を既知の経路に沿って通過させたときのバイナリセンサの反応から何らかの普遍的な関係性が学習できれば、未知の経路に沿って侵入する未知の対象物に対しても、その輪郭を推定できると考えられる。

本研究では各センサの反応を入力する入力素子と、対象物の輪郭（円形、長方形、正方形、正三角形）を出力する出力素子と、両者を結合する中間素子からなる3層ニューラルネットワークを構成し（図12）、バックプロパゲーション（誤差逆伝搬）法により入力側へ与える教師信号と出力側へ与える教師信号とを用いてニューラルネットワークに学習させ、輪郭推定を行う検討を行った。

中間素子の総数は、入力素子の総数の2倍から1つ引いたものとした。

$$n_{\text{中間素子}} = n_{\text{入力素子}} \times 2 - 1$$

また、出力素子の総数は対象物の輪郭（円形、長方形、正方形、正三角形）の種類の数であるので4とした。

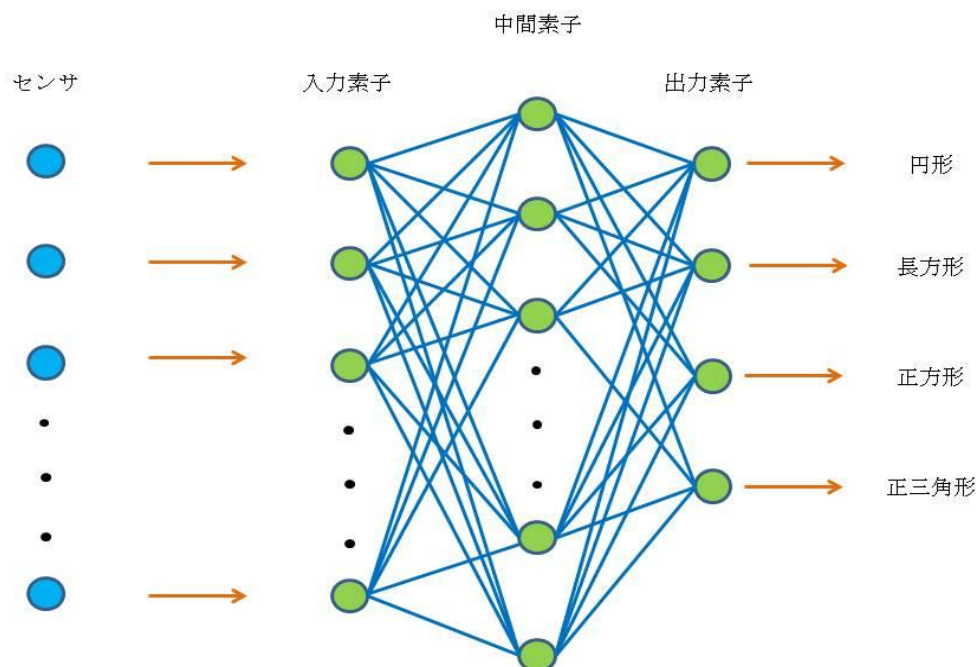


図12：3層ニューラルネットワーク。

第4章

シミュレーション

4.1 概要

4.2 に示す各シミュレーション条件の下で、センシングエリアを対象物が通過するとき、対象物をセンシングエリア内に感知したバイナリセンサ群の ID を抽出し、そのデータをバックプロパゲーションをシミュレートするニューラルネットワークプログラムに入力し学習させた。

さらに学習させたプログラムに対し未学習のデータを与え、その対象物が円形か長方形か正方形か正三角形を判定させた。

以下のシミュレーションは平面でのものであり、バイナリセンサのセンシングエリアの半径を 3.1523 とし、中心角を 2π (円形) とした。

また、監視領域にはバイナリセンサがある一定量撒かれているとする。この撒かれているセンサノードの密度も 3 通りに変化させた。

対象物は円形と長方形、正方形、正三角形の 4 通りで行った。

4.2 シミュレーション条件

シミュレーションは、監視領域を 100×100 (X 座標 : 50~150, Y 座標 : 50~150) としバイナリセンサを密度 0.01, 0.05, 0.10 の 3 通りとし、ランダムに撒いた。このランダムに撒いたバイナリセンサの位置は、乱数を発生させる素になる seed の値を 3 通りに設定したため、やはり 3 通りとなる。学習データとして形状が既知の対象物を X 座標に並行で Y 座標が異なる 11 の経路 (Y=50, 60, ..., 140, 150) に沿って秒速 10 で動かし、1 秒毎のセンサの反応をニューラルネットの学習用教師信号として用いた。学習データと判定データは共に円形、長方形、正方形、正三角形の 4 種類で、これらについて試みた。主な条件は次項の通りである。

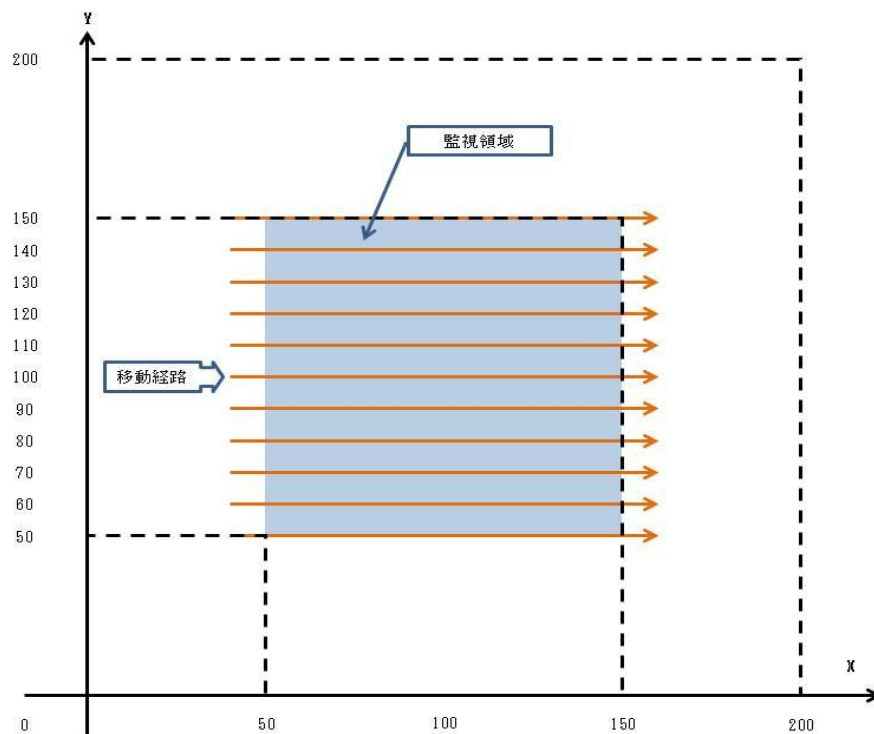


図 13 : 監視領域.

対象物が 4 種類であるため、出力素子は 4 つである。それぞれが対象物の形状に対応している。即ち、対象物が円形の場合は 1 つ目の出力素子が、対象物が長方形の場合は 2 つ目の出力素子が 1 に、対象物が正方形の場合は 3 つ目の出力素子が 1 に、対象物が正三角形の場合は 4 つ目の出力素子が 1 になる。よって複数の出力素子が 1 となることはない。また、教師信号の学習にはバックプロパゲーション法により素子間の結線重みを教師信号への近似として繰り返し計算させる。ただし、計算回数の上限は、5 千回とした。

シミュレーションするにあたり、条件を大きく 6 つのケースに分けた。表 4 の「密度」とはバイナリセンサが監視領域に撒かれたときの単位面積あたりの数である。また「半径」はバイナリセンサのセンシングエリアの半径である。「大きさ」は対象物の各形状のそれぞれの 3 通りの大きさである。

表 3 : 対象物

形状	大きさ
円形	半径=3
円形	半径=6
円形	半径=12
長方形	縦=2.5 横=10
長方形	縦=5 横=20
長方形	縦=10 横=40
正方形	縦=5 横=5
正方形	縦=10 横=10
正方形	縦=20 横=20
正三角形	一辺=7.6
正三角形	一辺=15.2
正三角形	一辺=30.4

表 4 : シミュレーション条件

ケース	密度	半径	大きさ
1	0.01	1	小
	0.01	1	中
	0.01	1	大
2	0.01	3	小
	0.01	3	中
	0.01	3	大
3	0.01	6	小
	0.01	6	中
	0.01	6	大
4	0.05	1	小
	0.05	1	中
	0.05	1	大
5	0.05	3	小
	0.05	3	中
	0.05	3	大
6	0.05	6	小
	0.05	6	中
	0.05	6	大

4.3 シミュレーション結果

4.3.1 ケース 1

センサ密度が 0.01 でセンシングエリアの半径が 1 のときの結果を以下に示す.

表 5 : 形状推定結果(ケース 1) .

ID	形状	大きさ	推定結果
1	円形	半径=3	円形
2	円形	半径=6	円形
3	円形	半径=12	円形
4	長方形	縦=2.5 横=10	円形
5	長方形	縦=5 横=20	長方形
6	長方形	縦=10 横=40	長方形
7	正方形	縦=5 横=5	円形
8	正方形	縦=10 横=10	円形
9	正方形	縦=20 横=20	正方形
10	正三角形	一辺=7.6	円形
11	正三角形	一辺=15.2	正三角形
12	正三角形	一辺=30.4	正三角形

ID が 1, 4, 7, 10 は対象物の大きさが最も小さいため, 推定結果が円形となってしまうている. センサ密度も 0.01 と低く, このケースではセンシングエリアの半径も最も短く, よってセンシングエリアは最も狭い. このためこのケースでの最も小さい対象物の形状を推定するまでには至っていない.

計算によって出された値も, この付近の推定値は全ての図形にほぼ同じ値を返している. つまり全部に反応している.

対象物の大きさが中クラスになるとようやく正三角形や長方形が推定された. 計算結果では, 正解の図形が確かに他の図形より値は高い. しかし, はっきりと判別したといえる程の差はない.

対象物の大きさが大クラスでは, 中クラスの大きさよりは, はっきりと判別できているようだ.

半径1

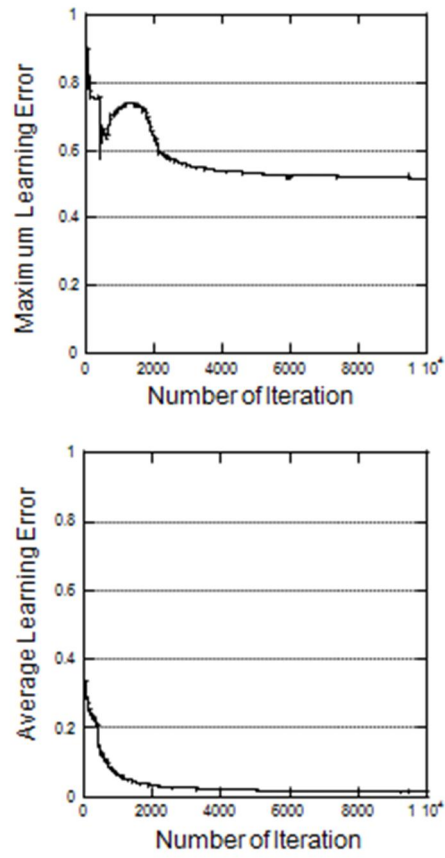


図 14：計算回数に対する最大誤差と平均誤差の推移.

計算回数が 2 千回くらいから，誤差が落ち着いてきているのが判る.

4.3.2 ケース 2

センサ密度が 0.01 でセンシングエリアの半径が 3 のときの結果を以下に示す.

表 6 : 形状推定結果 (ケース 2) .

ID	形状	大きさ	推定結果
1	円形	半径=3	円形
2	円形	半径=6	円形
3	円形	半径=12	円形
4	長方形	縦=2.5 横=10	正三角形
5	長方形	縦=5 横=20	長方形
6	長方形	縦=10 横=40	長方形
7	正方形	縦=5 横=5	正方形
8	正方形	縦=10 横=10	正方形
9	正方形	縦=20 横=20	正方形
10	正三角形	一辺=7.6	正三角形
11	正三角形	一辺=15.2	正三角形
12	正三角形	一辺=30.4	正三角形

半径3

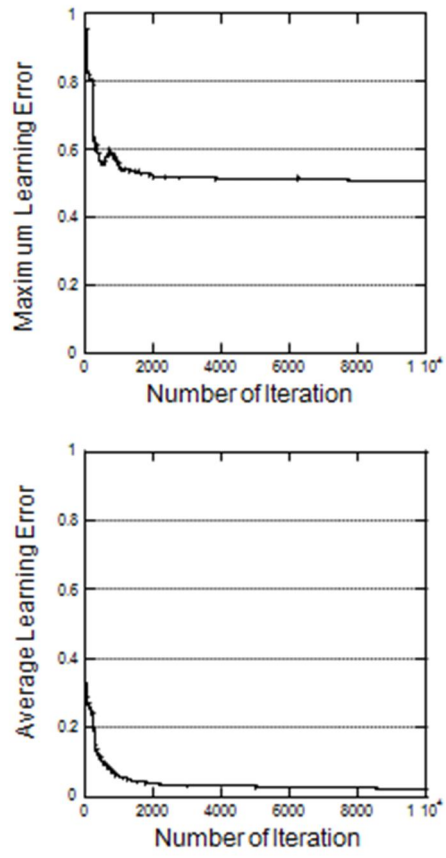


図 15 : 計算回数に対する最大誤差と平均誤差の推移.

4.3.3 ケース 3

センサ密度が 0.01 でセンシングエリアの半径が 6 のときの結果を以下に示す.

表 7 : 形状推定結果 (ケース 3) .

ID	形状	大きさ	推定結果
1	円形	半径=3	円形
2	円形	半径=6	円形
3	円形	半径=12	円形
4	長方形	縦=2.5 横=10	長方形
5	長方形	縦=5 横=20	長方形
6	長方形	縦=10 横=40	長方形
7	正方形	縦=5 横=5	正方形
8	正方形	縦=10 横=10	正方形
9	正方形	縦=20 横=20	正方形
10	正三角形	一辺=7.6	正三角形
11	正三角形	一辺=15.2	正三角形
12	正三角形	一辺=30.4	正三角形

半径6

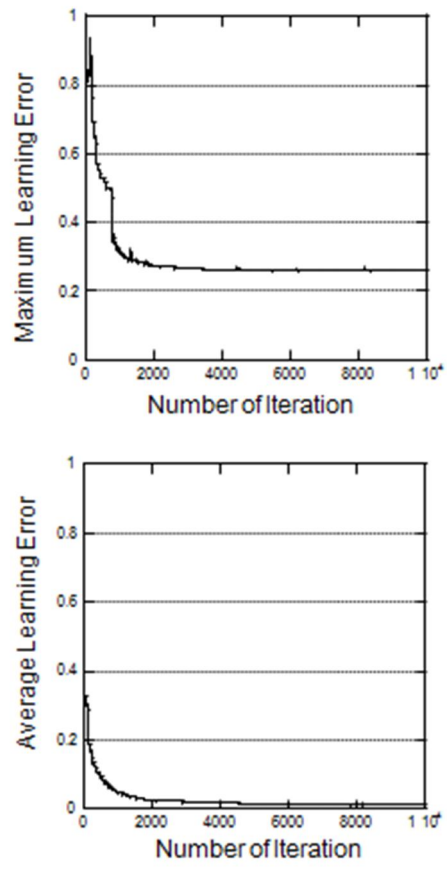


図 16 : 計算回数に対する最大誤差と平均誤差の推移.

4.3.4 ケース 4

センサ密度が 0.05 でセンシングエリアの半径が 1 のときの結果を以下に示す.

表 8 : 形状推定結果 (ケース 4) .

ID	形状	大きさ	推定結果
1	円形	半径=3	円形
2	円形	半径=6	円形
3	円形	半径=12	円形
4	長方形	縦=2.5 横=10	長方形
5	長方形	縦=5 横=20	長方形
6	長方形	縦=10 横=40	長方形
7	正方形	縦=5 横=5	円形
8	正方形	縦=10 横=10	正方形
9	正方形	縦=20 横=20	正方形
10	正三角形	一辺=7.6	正三角形
11	正三角形	一辺=15.2	正三角形
12	正三角形	一辺=30.4	正三角形

半径1

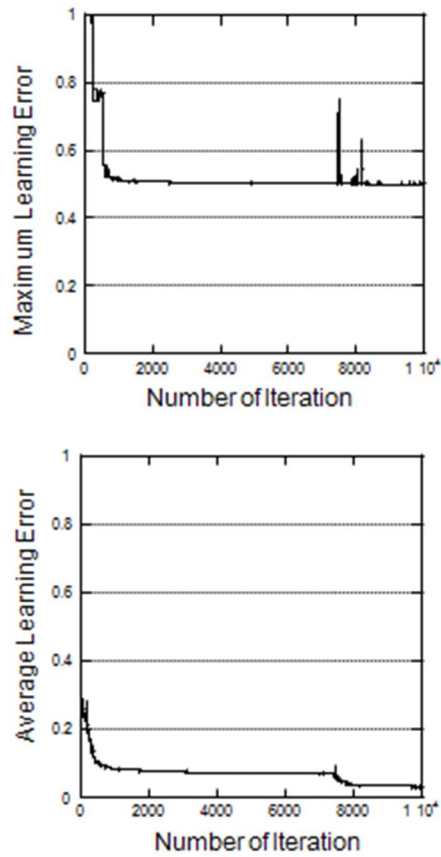


図 17：計算回数に対する最大誤差と平均誤差の推移.

最大誤差は 8 千回付近にインパルスのようなものがあるが、その他は平衡状態に見える。平均誤差も 8 千回手前で少し落ちている。入力素子数や中間素子数、あるいはゲインの値に関係があるかも知れない。今後の課題だ。

4.3.5 ケース 5

センサ密度が 0.05 でセンシングエリアの半径が 3 のときの結果を以下に示す.

表 9 : 形状推定結果 (ケース 5) .

ID	形状	大きさ	推定結果
1	円形	半径=3	円形
2	円形	半径=6	円形
3	円形	半径=12	円形
4	長方形	縦=2.5 横=10	長方形
5	長方形	縦=5 横=20	長方形
6	長方形	縦=10 横=40	長方形
7	正方形	縦=5 横=5	円形
8	正方形	縦=10 横=10	円形
9	正方形	縦=20 横=20	正方形
10	正三角形	一辺=7.6	正三角形
11	正三角形	一辺=15.2	正三角形
12	正三角形	一辺=30.4	正三角形

結構な好条件なケースであるはずが, 正方形の判別にまだやや難がある. やはり円形と正方形は間違え易いといえる.

半径3

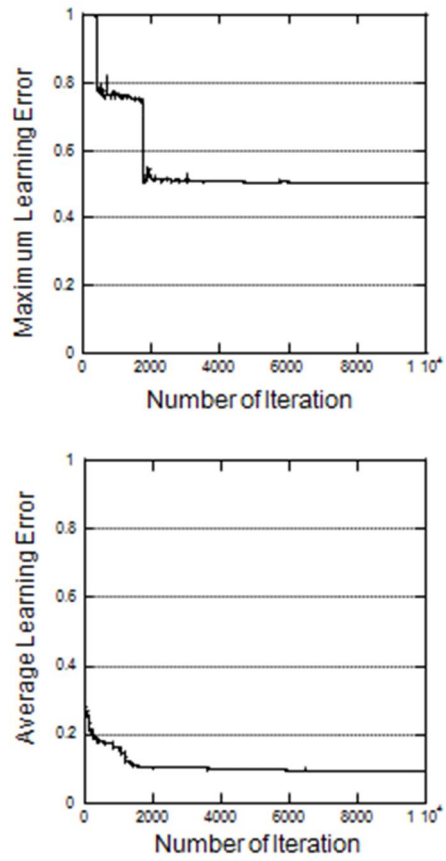


図 18 : 計算回数に対する最大誤差と平均誤差の推移.

最大誤差では 5 百回近くと千 5 百回近くできれいに落ちている.

4.3.6 ケース 6

センサ密度が 0.05 でセンシングエリアの半径が 6 のときの結果を以下に示す.

表 10 : 形状推定結果 (ケース 6) .

ID	形状	大きさ	推定結果
1	円形	半径=3	円形
2	円形	半径=6	円形
3	円形	半径=12	円形
4	長方形	縦=2.5 横=10	長方形
5	長方形	縦=5 横=20	長方形
6	長方形	縦=10 横=40	長方形
7	正方形	縦=5 横=5	正方形
8	正方形	縦=10 横=10	正方形
9	正方形	縦=20 横=20	正方形
10	正三角形	一辺=7.6	正三角形
11	正三角形	一辺=15.2	正三角形
12	正三角形	一辺=30.4	正三角形

今回のシミュレーションで最も好条件なケースである. 推定結果も満足のいくものである. ただ, 実際の計算結果はやはり一部に根拠となる数値が弱いものがある.

あ

半径6

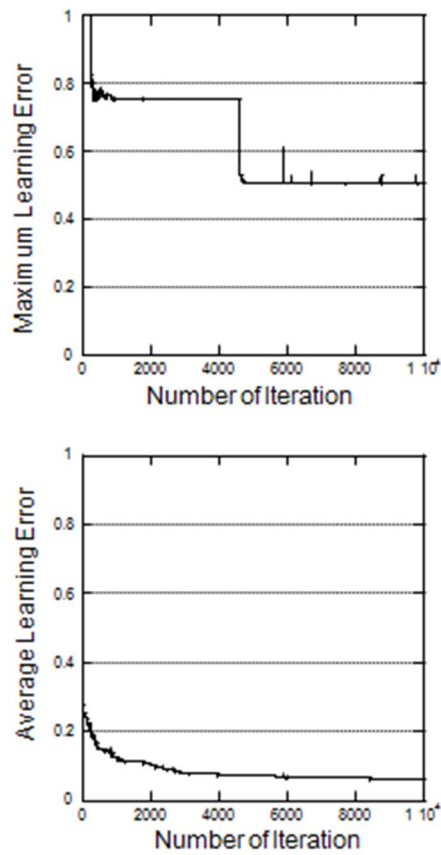


図 19：計算回数に対する最大誤差と平均誤差の推移.

最大誤差が計算回数 5 百回近くと 4 千 5 百回近くできれいに落ちているのが判る.

第5章

考察

5.1 まとめ

この研究の対象となるニューラルネットはセンサと同数の入力素子が必要であるが、はじめにニューラルネットの学習の手間を軽減させるために密度を 0.01 にし入力素子の総数を 100 程度に制限した。しかし、この密度では、より小型の物体の場合、検知するセンサの数がごくわずかであり、特に小さいの対象物の場合に、正方形を円形と間違えるケースが見受けられた。バイナリセンサの密度が低いため、形状の違いが明らかなものでなければ（長方形や正三角形）推定は難しいといえる。

また密度を上げればそれだけ入力素子の数が増える。入力素子が増えれば、それだけ中間素子の数が倍に増える。このため計算に膨大な時間を要することになる。センサの数が増えた場合でも学習のための計算がより短時間で終わられるような手法が必要である。

ただし、そういった新しい手法が見つからない内は、何度も試行錯誤を重ね取り扱っている問題に対する適切なゲインや重みを見つけていくしかない。

文 献

- [1]山下賢也, 塩田茂雄, 斉藤洋, “センサ密度が未知のバイナリセンサネットワークによる対象物形状推定,” 電子情報通信学会論文誌, vol. J95-D, No. 1, pp. 1-12, 2012.
- [2]H. Saito, O. Kagami, and M. Umehira, “Wide area ubiquitous network: the network operator’s view of a sensor network,” IEEE Commun. Magazine, vol. 46, pp. 112-120, 2008.
- [3]熊沢逸夫, “学習とニューラルネットワーク,” 森北出版株式会社, p 51, 1998