

階層型ニューラルネットワークを用いた移動ロボットの障害物回避

日高良和* 久保田智彦**

A Method of Obstacle Avoidance for A Mobile Robot Using Multi-Layered Neural Networks

Yoshikazu HITAKA* Tomohiko KUBOTA**

Abstract :

Generally, when a robot execute a task, it is necessary to make the motion plan in detail. In making this motion plan, the robot designer must understand every motion of a robot. However, it is difficult for the robot designer due to a complex motion of a robot and its environment. This paper proposes a new method of design a robot motion by using multi-layered neural networks. This method is that the robot designer just decides a basic motion of a robot and the complicated motion of a robot is created by the neural networks. The proposed method was tested by simulation of a mobile robot that has two wheels and two infrared sensors. The simulation was demonstrated the method is useful for the purpose.

Keywords: motion planing, mobile robot, neural networks

1. まえがき

ロボットにいろいろな作業をさせるためには、一般に設計者がロボットの行動を細部まで設計するというトップダウン的な手法が用いられる。しかし、近年、ロボットを利用する分野の拡大によってロボットに要求される行動が複雑化してきており、トップダウン的な手法によるロボットの行動設計が困難になってきている^{[1][2]}。一方、人間が行動を決定する場合は、あいまいな情報をもとにしていることがある。このような情報処理を行なうためにニューラルネットワークの研究がさかんに行なわれている。ニューラルネットワークは、入力値に対して適当な値を計算できる情報処理能力をもっている。そのため、基本的な最小限の行動設計を行なうだけで、高度な行動を実現できる可能性がある。そこで本研究は、二輪移動ロボットを対象としてロボットに取り付けられたセンサ出力に応じた車輪駆動用モータ指令値を階層型ニューラルネットワークによって生成することを試みた。

まず2章では、本研究に用いる二輪移動ロボットの構造と仕様について述べる。また、車輪駆動用のモータコントローラにニューラルネットワークを利用し、行動設計の簡略化を提案している。

次に3章は、ニューラルネットワークの基本的な構造とその学習アルゴリズムについて述べている。ロボットのモータ駆動指令値を決定させるためのニューラルネットワークは2入力2出力で、入力層、中間層、出力層の3層からなる階層型のニューラルネットワークである。ネットワークは Visual Basic により構築した。入力層には赤外線センサ出力値が入力され、出力層からはそれに対応する各モータへのモータ駆動指令値が出力される。ニューラルネットワークの学習は、センサ出力値に対するモータ駆動指令値を教師データとして与え、バックプロパゲーション法（誤差逆伝播法）^{[3][4]}を用いて行なっている。

4章では学習の様子から決定した、最終的なニューラルネットワークの構造と構築したネットワークの入出力機能について述べている。学習が十分に行なわれたニューラルネットワークは、教師データとして与えている場合はもちろんのこと、与えていないセンサ出力値に対してもモータ駆動指令値を生成することができた。

最後に5章で本稿のまとめを述べている。

(2000年12月22日受理)

* 宇部工業高等専門学校電気工学科

** 宇部工業高等専門学校専攻科生産システム工学

2. 二輪移動ロボットの構造と制御系

2.1 二輪移動ロボットの構造

移動ロボットは、研究分野で多く使用されている AAI 製のケペラロボットを想定している。ケペラロボットは、直径 55 mm、高さ 30 mm の円柱形で、重量 70g の小型ロボットであり、図 1 に示すように外界センサと左右独立の駆動輪を有している。また、このロボットは表 1 にあるように 32 ビットマイクロコントローラやシリアルポートが搭載されており、ユーザーが自由にロボットの行動をプログラミングできるようになっている。本研究では、ケペラロボットに前方の障害物を避けながら前進させることを想定している。この行動を実現するために、最小限のハードウェアを利用することとする。すなわち、前方の障害物を検出する赤外線センサは、配置にある 8 個のセンサのうち前方左右の 2 個を利用し、駆動輪は前進のみを行う。赤外線センサは障害物に対する距離に応じ、前方に障害物がない状態を 0、障害物に最接近した状態で 1 となる 0~1 までの値を出力する。2 個の駆動輪は、左右のセンサ出力値に応じて回転数を制御されながら前進を行なう。その回転指令値は、モータが回転していない状態を 0、定格で回転している状態を 1 として 0~1 までの値が指定される。

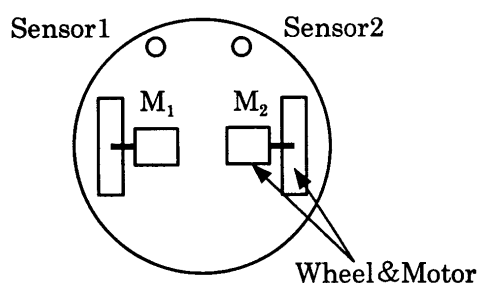


図 1. 二輪移動ロボットの構造

表 1. ケペラロボットの仕様

サイズ	直径 55mm、高さ 30mm
自重	約 70g
基本センサ	8個の赤外線センサ
電源	充電可能な Ni-Cd バッテリ
プロセッサ	モトローラ社製 MC68331 16Hz 32ビットマイクロコントローラ
ROM	256Kバイト
RAM	526Kバイト
バス構造	K-bus (35ピン汎用入力バス)
ユーザI/O	シリアルポート (ミニチュアコネクタ)

2.2 二輪移動ロボットの制御系

ロボットの制御系ブロック線図を以下に示す。移動ロボットは、前方の障害物を検出すると左右の駆動輪の回転数を制御して進行方向を調節する。このとき、大きく曲がり障害物を回避すると到達地点に達する時間がかかってしまう。そのため駆動輪の回転数は、赤外線センサの出力値 S_1, S_2 に応じた最適な値を取る必要がある。簡易的にこのような制御を実現させようとすると、図 2 のようにモータドライバコントローラによってセンサ出力 S_1, S_2 に応じた指令値 M_1, M_2 を決めなければならない。しかしこの方法では、設計者が各センサ値に対するモータ駆動指令値を詳細まで決定する必要がある。

本研究は、図 2 のドライバコントローラ部に各赤外線センサ出力値 S_1, S_2 と最適なモータ駆動指令値との関係を学習したニューラルネットワークを適用する。このニューラルネットワークは、任意のセンサ出力値に対して障害物を回避する行動を瞬時に決定できることが期待される。また、設計者は代表的なセンサ出力 S_1, S_2 とモータ駆動指令値 M_1, M_2 の教師データを用意するだけであり、行動設計の省力化ができる。

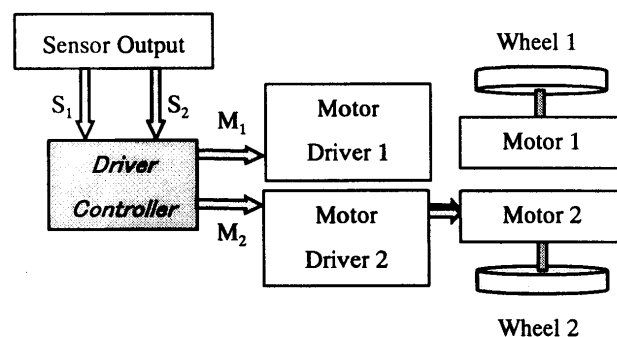


図 2. 一般的な二輪移動ロボットのブロック線図

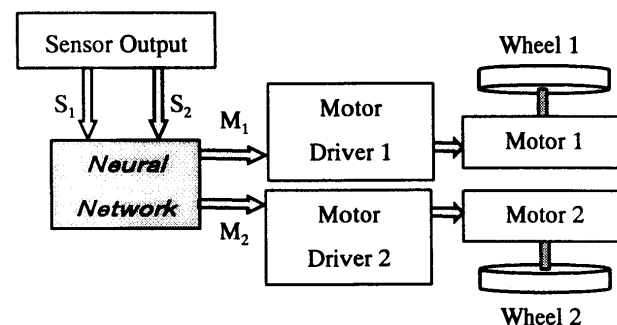


図 3. ニューラルネットワークを用いた二輪移動ロボットのブロック線図

3. 二輪移動ロボット用階層型ニューラルネットワーク

3.1 ニューラルネットワークと学習アルゴリズム

3.1.1 ニューラルネットワーク

脳の主構成要素はニューロンであり、そのニューロンがシナプス結合と呼ばれる結合を行うことによって、あいまいな情報の処理や学習を行っていることが知られてきた。このような、脳に関する生理学的知見に脳機能の工学的実現を行なうために構成された人工的なネットワークをニューラルネットワークと呼んでいる^[9]。

ニューラルネットワークは、図4のような多入力-出力非線形要素としてモデル化されたニューロンを多数結合した神経回路網であり、ニューロンの接続のしかたによって、階層構造型と相互結合型の2つの種類がある。階層構造型は図5に示すように、ニューロンを層状に並べ、前の層から次の層へ一方方向にのみ信号が伝わる。このタイプのネットワークは、入力層のニューロンに加えた信号(入力信号)に対して出力層のニューロンの出力(出力信号)が一意的に定まる。相互結合型は図6に示すように、信号の流れがフィードバックされる構造であり、ネットワークの状態量変化を設計することによって種々の情報処理が行なえる。

本研究では、制御分野で広く利用されている階層型ニューラルネットワークを適用し、その学習はバックプロパゲーション法を導入する。

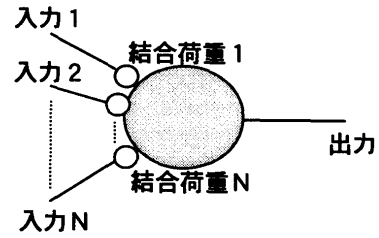


図4. ニューロンモデル

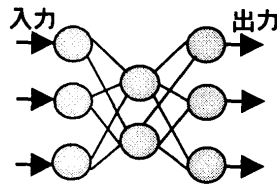


図5. 階層構造型

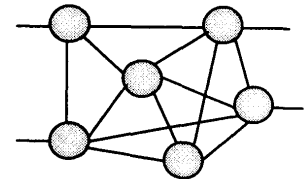


図6. 相互結合構造型

3.1.2 バックプロパゲーション法

バックプロパゲーション法は、ネットワークの出力で所望の値が教師データとして与えられる教師付き学習法の1つである。この学習は、ネットワークの出力値と教師データとの学習誤差に基づきシナプス結合を逐々に変更し、誤差が無くなるまで繰り返す方法である。

図7において、ニューロンの出力関数を f 、第 m 層の i 番目のニューロンの出力を q_i^m 、そのニューロンへの入力の総和を p_i^m とする。ただし、 $m=1,2,\dots,M$ 、 $i=1,2,\dots,K$ とする。今、第 $(m-1)$ 層の j 番目のニューロンから m 層の i 番目のニューロンへの結合係数を w_{ij}^m とすると q_i^m と p_i^m は以下の式のようにになる。

$$q_i^m = f(p_i^m), \quad p_i^m = \sum_j w_{ij}^m q_j^{m-1} \quad (1)$$

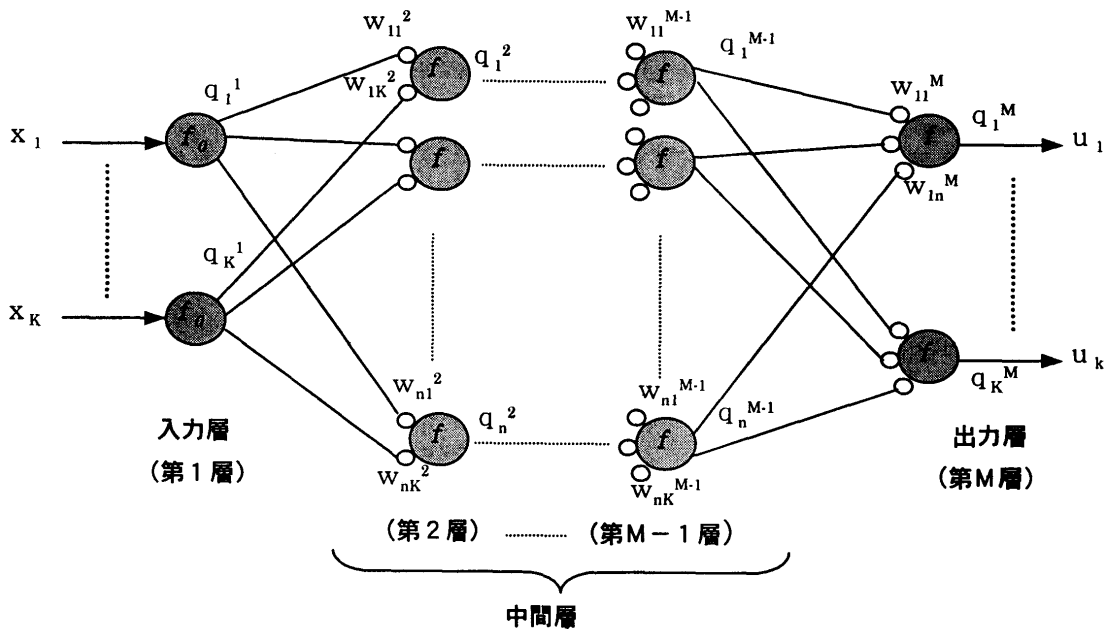


図7. 階層型ニューラルネットワーク

一方、学習誤差は、階層型ニューラルネットワークの最終層である出力層の出力を u_k 、その目標値（教師データ）を t_k ($k=1,2,\dots,K$) とするとき、

$$E = \frac{1}{2} \sum_{k=1}^K (t_k - u_k)^2 \quad (2)$$

として定義する。学習は、この学習誤差がなくなるように結合荷重の変更を繰り返すことになる。結合荷重の修正は、結合荷重 w_{ij}^m に対する誤差 E の変化を見ればよい。ため、 E を w_{ij}^m で偏微分した、

$$\Delta w_{ij}^m = -\eta \frac{\partial E}{\partial w_{ij}^m} \quad \eta > 0 \quad (3)$$

の変化量を用いて行なわれる。ここで η は学習係数であり、学習速度を上げるためには η を大きな値とすればよい。

i) $m=M$ 層（出力層）の場合

出力層、すなわち $m=M$ のとき、(3)式の $\frac{\partial E}{\partial w_{ij}^m}$ は

$$\frac{\partial E}{\partial w_{ij}^M} = \frac{\partial E}{\partial p_i^M} \frac{\partial p_i^M}{\partial w_{ij}^M} = \frac{\partial E}{\partial p_i^M} q_j^{M-1} \quad (4)$$

である。ここで

$$\delta_i^M \triangleq - \frac{\partial E}{\partial p_i^M} \quad (5)$$

とおくと、(4),(5)式を代入した(3)式は

$$\Delta w_{ij}^M = \eta \delta_i^M q_j^{M-1} \quad (6)$$

となる。また $u_i = q_i^M$ であることに着目すると、 δ_i^M は、(1),(2)式から

$$\delta_i^M = - \frac{\partial E}{\partial q_i^M} \frac{\partial q_i^M}{\partial p_i^M} = (t_i - u_i) \dot{f}(p_i^M) \quad (7)$$

である。ただし $\dot{f}(p_i^M) = df(p_i^M)/dp_i^M$ を示す。

ii) $m < M$ 層の場合

まず、出力層の1つ前の層である、 $m=M-1$ 層の場合について考える。この場合、 w_{ij}^m の値が $p_i^{M-1}, q_i^{M-1}, p_k^M$ ($k=1,2,\dots,K$) に順次影響してくることを考慮する

と、(7)式の関係が少し異なったものとなる。

まず、結合荷重の変化量は、

$$\frac{\partial E}{\partial w_{ij}^m} = \frac{\partial E}{\partial p_i^m} \frac{\partial p_i^m}{\partial w_{ij}^m} = \frac{\partial E}{\partial p_i^m} q_j^{m-1} \quad (8)$$

で与えられ、(5)式と同様に δ_i^m を導入すると

$$\delta_i^m \triangleq - \frac{\partial E}{\partial p_i^m} = - \frac{\partial E}{\partial q_i^m} \frac{\partial q_i^m}{\partial p_i^m} = - \frac{\partial E}{\partial q_i^m} \dot{f}(p_i^m) \quad (9)$$

となる。(9)式を代入した(8)式を用いると、(3)式は

$$\Delta w_{ij}^m = \eta \delta_i^m q_j^{m-1} \quad (10)$$

とできる。

また、(9)式の $\frac{\partial E}{\partial q_i^m}$ は、(2),(5)式を代入し、

$m=M-1$ であることに着目し整理すると

$$\begin{aligned} \frac{\partial E}{\partial q_i^m} &= \sum_{k=1}^K \frac{\partial E}{\partial p_k^M} \frac{\partial p_k^M}{\partial q_i^m} = - \sum_{k=1}^K \frac{\partial E}{\partial p_k^M} \frac{\partial (\sum_j w_{kj}^M q_j^{M-1})}{\partial q_i^{M-1}} \\ &= - \sum_{k=1}^K \delta_k^M w_{ki}^M \end{aligned} \quad (11)$$

となる。したがって、 δ_i^m は (11)式を(9)式に代入した、

$$\delta_i^m = \delta_i^{M-1} = \dot{f}(p_i^m) \sum_{k=1}^K \delta_k^M w_{ki}^M \quad (12)$$

から計算できる。ただし $\dot{f}(p_i^m) = df(p_i^m)/dp_i^m$ を示す。

(12)式より、 $m=M-1$ 層の δ_i^m 、すなわち δ_i^{M-1} を求めるためには、出力層 (M 層) の情報を必要とすることがわかる。同様に、 $m=M-2, M-3, M-4, \dots$ の場合も、それぞれ $m=M-1, M-2, M-3, \dots$ で求めた $\delta_i^{M-1}, \delta_i^{M-2}, \delta_i^{M-3}, \dots$ を使い、出力層から入力層へ向かって順次計算することで、 $\delta_i^{M-2}, \delta_i^{M-3}, \delta_i^{M-4}, \dots$ を決定することができる。ここで求めた δ_i^m を(10)式に代入し、結合荷重の修正を行なう。

各層の結合荷重の修正は、出力層から入力層へ逆の経路をたどる必要がある。そのため、このような結合荷重の修正法はバックプロパゲーション(誤差逆伝播)法と呼ばれている。

以上をまとめ、各層においてのバックプロパゲーションアルゴリズム (一般化 δ ルール) を以下に示す。

$$\Delta w_{ij}^m = \eta \delta_i^m q_j^{m-1} \quad (13)$$

$$\delta_i^m = \begin{cases} \dot{f}(p_i^m)(t_i - u_i), & m = M \\ \dot{f}(p_i^m) \sum_{k=1}^K \delta_k^{m+1} w_{ki}^{m+1}, & m < M \end{cases} \quad (14)$$

このルールに従い(15)式によって w_{ij}^m を計算し、ニューロン間の結合係数の修正を行う。

$$w_{ij}^m(k+1) = w_{ij}^m(k) + \Delta w_{ij}^m(k) \quad (15)$$

3.2 モータ駆動指令値生成のためのニューラルネットワーク

二輪移動ロボットが前方の障害物を回避しながら前進できるように、車輪駆動のためのモータ駆動指令値をニューラルネットワークによって与えるシステムを考える。

このネットワークは入力層、中間層、出力層からなる階層型ニューラルネットワークとし、その学習はバックプロパゲーション法を用いて行なう。ネットワークの入力層は、障害物検出用の2個の赤外線センサから0~1の値を持つ信号がそれぞれ S_1, S_2 として入力され、出力層はネットワークにより決定された各モータドライバへの0~1のモータ駆動指令値 M_1, M_2 が出力される。

入力層のニューロンの出力関数 $f(x)$ は、入力値が0~1の値なので

$$f(x) = x \quad (16)$$

と入力値をそのまま出力するようにした。また、中間層、出力層の出力関数は、ニューラルネットワークの出力値が0~1の範囲内であるため

$$f(x) = \frac{1}{1 + e^{-ax}} \quad (17)$$

で表されるシグモイド関数とした。ここで、 a はシグモイド関数の傾きを決定する定数である。

中間層、出力層の出力関数をシグモイド関数とすることにより

$$\dot{f}(p_i^m) = a q_i^m (1 - q_i^m)$$

となり、(14),(15)式の一般化 δ ルールの微分計算が省略できる。

次に、誤差関数 E は

$$E = \frac{1}{2} \{(M_1^T - M_1^0)^2 + (M_2^T - M_2^0)^2\} \quad (18)$$

として与える。ここで M_1^T, M_2^T はモータドライバへの教師データであり、 M_1^0, M_2^0 はネットワークの出力値である。

ロボットに取り付けられた赤外線センサの出力値は、障害物を検出していないときは0、障害物を検出し、それに近づくにつれて大きな値となり最接近した場合に1の値をとる。教師データは、このセンサ出力値に応じた左右の駆動輪への指令値を決めることで与えられる。センサ出力値のすべての組み合わせに対してモータ駆動指令値を設計できないので、ここではロボットの行動を代表する表2の4パターンの教師データを使用する。

ロボットの前方に何も障害物が無くセンサが反応していない状態である $(S_1, S_2) = (0, 0)$ の場合には、モータ駆動指令値は $(M_1, M_2) = (1, 1)$ とし、両方のモータを定格で回転させ前進を続ける。 S_1 または S_2 の前方に障害物があれば $(S_1, S_2) = (1, 0)$ や $(S_1, S_2) = (0, 1)$ のセンサ出力値に対して、片側だけのモータを駆動させ障害物を回避できるように、それぞれ $(M_1, M_2) = (1, 0)$, $(M_1, M_2) = (0, 1)$ を与える。そしてロボットの直前に障害物があり、センサの出力が $(S_1, S_2) = (1, 1)$ の場合には、モータ駆動指令値を $(M_1, M_2) = (0, 0)$ としてモータの回転を止めてロボットを停止させるようにする。

表2. 教師データ

S_1	S_2	M_1	M_2
0	0	1	1
0	1	0	1
1	0	1	0
1	1	0	0

4. 階層型ニューラルネットワークの構築と学習結果

4.1 ネットワークの構築

モータ駆動指令値 M_1, M_2 を生成するための階層型ニューラルネットワークを 3.1.2 のバックプロパゲーション法に基づいて、Visual Basic により構築した。作成したプログラムの実行画面を図 8 に示す。設計者は、シグモイド関数の定数 a 、学習係数 η の 2 つの定数を任意に設定でき、また教師データのパターン数と 1 パターンあたりの学習回数も変化できるようになっている。モータ駆動指令値生成にあたって、各定数、データのパターン数、学習回数を各々変化させてネットワークの構築を行った。

試行錯誤の中でネットワークの汎化性が最も高かった、シグモイド関数の定数 $a=3$ 、学習係数 $\eta=0.5$ を採用することにした。

4.2 学習誤差

ネットワークの構造を決めるために、各層のユニット数が問題となる。入力層と出力層のユニット数はセンサとモータがともに 2 個であることから、各層のユニット数も各々 2 個と決定できる。残された中間層のユニット数を決定するために、ユニット数の違いによるネットワークの学習の様子を調べた。学習は、表 2 に示す 4 パターンの教師データを 1 パターン当たり 100 回学習させ、パターンを変えて 300 回繰り返す、計 3 万回行なった。図 9 (a) は中間層のユニット数が 2 個の場合であり、2 万回を超えた付近から誤差は減少している。

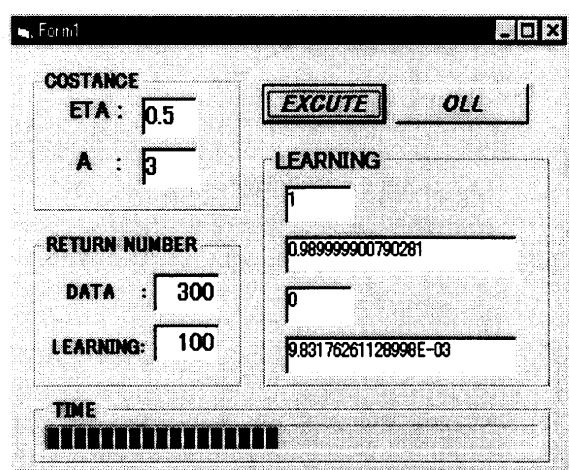


図 8. モータ駆動指令値生成プログラム実行画面

また、ユニットの数を増やし図 9 (b) の中間層 6 個の場合や、図 9 (c) の中間層 10 個とした場合は、それぞれ学習が 1 万回程度から誤差は減少し、3 万回あたりで誤差は十分収束するというほぼ同様の結果が得られた。そのため、ネットワークの汎化性確保と計算時間短縮を考慮して、中間層のユニット数は 6 個とした。

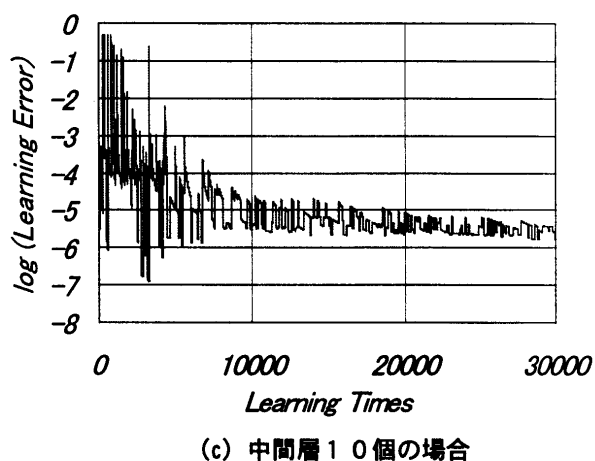
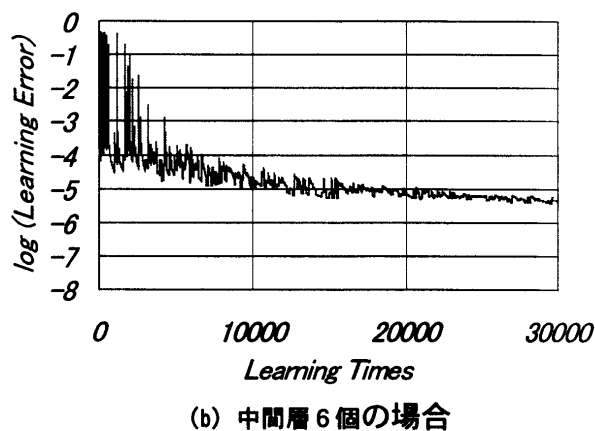
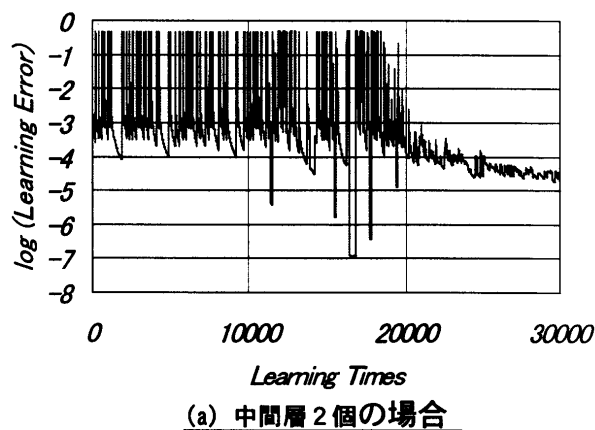


図 9. 学習誤差

4.3 学習結果

学習が十分に行われたニューラルネットワークの出力値は表3(b)のようになった。モータ駆動指令値は、教師データとして与えられている場合はもちろんのこと、任意のセンサ出力値の場合においても最適なモータ駆動指令値が得られている。

例えば、センサ1前方に障害物が無く、センサ2前方の少し離れた場所に障害物がある場合を想定したセンサ出力(S_1, S_2)=(0, 0.5)は教師データとして与えられていないが、モータ1を定格の30%で回転させ、モータ2を定格で回転させるというモータ駆動指令値(M_1, M_2)=(0.3, 1)を決定している。また、ロボット正面前方の少し離れた場所に障害物があると想定したセンサ出力(S_1, S_2)=(0.5, 0.5)に対しては、それぞれのモータを定格の40%で回転させ、ロボットをゆっくりと前進させるというモータ駆動指令値(M_1, M_2)=(0.4, 0.4)を決定することができた。

このことから、ニューラルネットワークを用いることにより、設計者は単純な行動設計をするだけで、ロボットに複雑な行動を実行させられる可能性があると考えられる。

表3. 階層型ニューラルネットワークの教師データと出力値

(a) 教師データ

S_1	S_2	M_1	M_2
0	0	1	1
0	1	0	1
1	0	1	0
1	1	0	0

(b) ネットワークの出力値

S_1	S_2	M_1	M_2
0	0	1	1
0	0.5	0.3	1
0	1	0	1
0.5	0	1	0.3
0.5	0.5	0.4	0.4
0.5	1	0	0.5
1	0	1	0
1	0.5	0.5	0
1	1	0	0

5. まとめ

ロボットの利用分野の拡大により、ロボットに要求される行動が複雑化してきている。このため、設計者がロボットのすべての作業を把握し細部まで設計するということが非常に困難になっている。そこで本研究は、移動ロボットの障害物回避にあたり小型の二輪移動ロボットを対象として、ロボットに取り付けられた赤外線センサ出力に応じた車輪駆動用モータ指令値をニューラルネットワークによって生成することで行動設計の省力化を試みた。

ロボットの車輪駆動用モータ指令値を決定させるためのニューラルネットワークは2入力2出力で、ネットワークはVisual Basicにより構築した。入力層には赤外線センサ出力信号が入力され、出力層からはそれに対応するモータ駆動指令値が各モータへ出力される。ニューラルネットワークの学習は、ロボットの行動を決定する代表的な値を教師データとして与え、その学習はバックプロパゲーション法を用いて行なった。

最終的なニューラルネットワークは、試行錯誤を行ない、中間層のユニット数を6個とする3層からなる階層型ニューラルネットワークとした。学習が十分に行われたニューラルネットワークの出力値、すなわちモータ駆動指令値は、種々の赤外線センサ出力値に対して適当なものであった。

このことから、設計者は単純なロボットの行動設計を行なうだけで、ロボットに高度な作業を実行させ得ることが確認できた。

参考文献

- [1] 寒河江 政史, 小塚 成一, 平田 廣則, “遺伝子評価を導入した遺伝的アルゴリズムによる自律移動ロボットの行動ルール獲得”, 電気学会論文誌, Vol.119-C, No.12, pp.1535-1541, 1999
- [2] 日高 良和, 泉 照之, 久保田 智彦, “ニューラルネットワークによるフレキシブル・リンクハンマーの駆動信号生成”, 平成11年電気情報関連学会中国支部連合大会 講演論文集, pp101, 1999
- [3] 大松 繁, “ニューラルネットワークと適応制御”, コンピュートロール, No.32, コロナ社
- [4] 中野 馨, “ニューロンコンピューターの基礎”, コロナ社, 1990