

## ニューラルネットとその高次元化 —複素数・四元数へ—

拓殖大学 工学部 電子システム工学科 小川 毅彦

### 1. はじめに

コンピュータの発展は目覚ましく、スーパーコンピュータからスマートフォンに至るまで、我々の生活に無くてはならないものになっている。いわゆるノイマン型コンピュータは記憶装置に格納したデータを順番に読み込み、アルゴリズムに従って処理を実行する。その処理は、厳密な数学的体系に基づくボトムアップ型の処理とみなすことができる。近年のCPUの高速化とメモリの大容量化により、コンピュータに不可能な処理は無いようにも見えるが、一方で苦手な処理も存在する。

コンピュータの厳密な処理に対して、現実の複雑な現象を人間が行うようなあいまいかつ柔軟なアプローチで解こうとするものがソフトコンピューティングである。それらには、ニューラルネットやファジー理論、進化的アルゴリズム、カオス理論などが含まれるが、本稿では人間の脳の働きを模倣した情報処理モデルの1つであるニューラルネットについて紹介する。

ニューラルネット研究の歴史は古く、1940年代のノイマンによるプログラム内蔵方式の提唱とほぼ同じ頃、ニューロンの数理モデルが発表されている。これまでに様々なニューラルネットモデルが提案され、さらに工学的な応用も見られる<sup>(1)</sup>。その研究も近年では広がりを見せ

ており、複素数や四元数など高次元への拡張や、層を重層化させたディープラーニングなどが盛んに研究されている。本稿では、ニューラルネットの原理を説明し、さらに最近研究が進められている高次元のニューラルネットについて紹介する。

### 2. ニューラルネットの原理

人間の脳には約140億個のニューロン（神経細胞）があり、それらが互いに結合して、人間の記憶や判断などの精神活動を実現している。ニューロンは、入出力を持った情報処理素子であり、その出力が他のニューロンの入力となることで、複雑に結合したニューラルネット（神経回路網）を構成している。

人工ニューラルネットは、人間の脳の計算処理機構を模倣した計算モデルである。なぜ人間の脳のモデルを用いるのか？コンピュータにも、画像などからの直観的判断や、あいまい情報の判断、誤りの多いデータの処理など苦手な処理がある。人間の方がコンピュータよりも優れている処理も多い。そこで、人間の脳を模倣して、その機構を取り入れ、人間の優れた処理能力を利用しようというのが主な動機である。

ニューロンのモデル化やネットワーク接続の構成は、その目的と着目点によって異なってくる。人工システムと生体システムには差異があ

るものの、これらの特徴は従来のコンピュータとは異なるものである。ニューラルネットを用いることで従来のコンピュータとは異なる特徴を持った処理が可能となる。

ニューラルネットの代表的なモデルとして、多層型、相互結合型、自己組織型、連想記憶型などがある。本稿では、もっとも基本的な教師つき写像学習型の、多層型ニューラルネットについて説明する。このモデルは、入力と出力の間の非線形関係を学習することができる。

### 3. 多層型ニューラルネット

まず、ニューロンのモデルを考える。多層型ニューラルネットで用いるモデルでは、シナプスによる他の神経細胞からの入力の制御と、他の神経細胞からの入力の集積、ある閾値以上で興奮し情報を出力する点に着目する。他の神経細胞からの入力を $x_i$ 、荷重によって表現する情報入力の制御を $w_i$ 、閾値を $\theta$ 、他の神経細胞への出力を $y$ とすると、神経細胞の入出力関係は、

$$y = f\left(\sum_{i=1}^n w_i x_i - \theta\right) \quad (1)$$

と表現できる。 $f$ は入出力特性を与える関数である。ニューロンモデルの模式図を図1に示す。

これらのモデルを層状に接続したものが多層型ニューラルネットである。図2に、入力層・中間層・出力層を持つ3層型のモデルを示す。ここで中間層と出力層に上記のニューロンモデルを用いる。このモデルを用いて、多数の入出力データから、多入力多出力の非線形入出力関係を学習できる。以下では、勾配法に基づく誤差逆伝搬法による学習の原理を示す。

1つの学習入力に対する出力層 $r$ 番目の素子

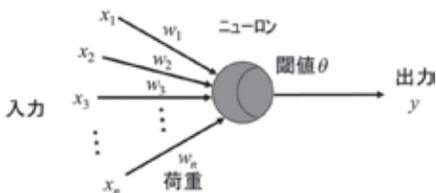


図1 ニューロンモデル

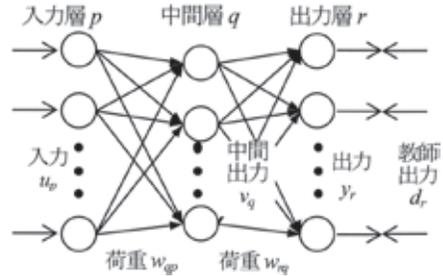


図2 3層型ニューラルネットの例

出力 $y_r$ とその教師 $d_r$ 値に対し、誤差関数を

$$E = \frac{1}{2} \sum_r (d_r - y_r)^2 \quad (2)$$

と定義する。また、中間層 $q$ 番目の素子出力 $v_q$ と、中間層 $q$ 番目の素子と出力層 $r$ 番目の素子間の荷重 $w_{rq}$ とし、出力層 $r$ 番目の素子への入力の総和を表す荷重和項を $S_r$ とすると、

$$y_r = f(S_r), \quad S_r = \sum_q w_{rq} \cdot v_q, \quad (3)$$

と表すことができる。さらに、入力層 $p$ 番目の入力 $u_p$ と、入力層 $p$ 番目の素子と中間層 $q$ 番目の素子間の荷重 $w_{qp}$ をとし、中間層 $q$ 番目の素子への入力の総和を表す荷重和項を $S_q$ とすると、

$$v_q = f(S_q), \quad S_q = \sum_p w_{qp} \cdot u_p, \quad (4)$$

と表すことができる。ここで、 $f$ は素子の特性関数で、ここでは

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (5)$$

で表されるシグモイド関数を考える。このとき、荷重に対する誤差関数の傾きの逆方向に微小移動を繰り返すことで、誤差 $E$ の最小化を図る。これを元に、中間層と出力層の間の荷重を、

$$w_{rq}(t+1) = w_{rq}(t) + \eta \delta_r v_q$$

$$\delta_r = (d_r - y_r) y_r (1 - y_r) \quad (6)$$

の式で、入力層と中間層の間の荷重を、

$$\begin{aligned} w_{qp}(t+1) &= w_{qp}(t) + \eta \delta_q u_p \\ \delta_q &= v_q(1-v_q) \sum_r \delta_r w_{rq} \end{aligned} \quad (7)$$

の式で、それぞれ繰り返し修正することで、誤差関数を最小化する荷重値を求めることができる。ここで $\eta$ は微小係数である。これにより、入力に対し与えられた教師値を出力する関係が得られる<sup>(1)</sup>。

応用例として、多層型ニューラルネットを用いたパターン判別プログラムを紹介する。このモデルは、10個のパターンを学習し、与えた1個のパターンがどのパターンに対応するか判断する。パターンは7×7のグリッドで表現し、入力層49個、出力層10個、中間層10個とする。なお、このソフトは文献(1)付録をもとにWindowsプログラムとして作成したものである。

図3に、10個の数字パターンの学習・判別を行った場合の実行例を示す。結果によると、それぞれの場合で、多少の誤りを含むパターンからも正しく判別が行われていることがわかる。このプログラムでは10個の基本パターンを学習しているだけであるが、実際のパターン認識に用いる場合は、ノイズや誤差を含む多数の学習データを与える必要がある。また精度向上のために、入力データ点数の増加や、入力前処理の工夫、さらにはネットワークのサブネット化などさまざまな工夫が行われる。

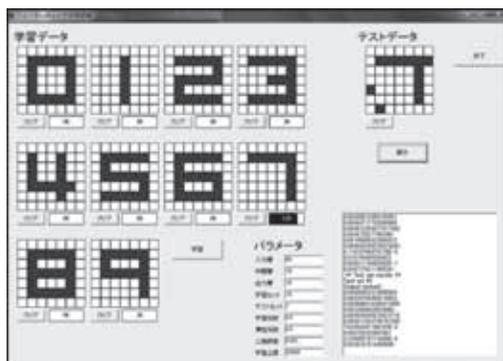


図3 3層型ニューラルネットによるパターン判別の実行例

#### 4. 高次元ニューラルネット

近年、複素数や四元数などの高次元数や、その一般化としてクリフォード数に拡張されたニューラルネットモデルが研究されている<sup>(2)~(4)</sup>。複素数については、回路解析や制御理論をはじめとして、さまざまな工学分野において利用されている。また、高次複素数の1つである四元数は、3次元空間における代数表現を容易に行うことができるため、コンピュータグラフィクスや姿勢制御などの分野で注目されている。

四元数はW. R. Hamiltonによって考案された高次元数で、実部と3つの虚部から成る。3つの虚数単位を $i, j, k$ とすると、四元数は

$$x = x_R + ix_I + jx_J + kx_K \quad (8)$$

と表現される。 $x_R, x_I, x_J, x_K$ は実数で四元数の各成分を表す。また、

$$\begin{aligned} i^2 = j^2 = k^2 = ijk = 1, \quad ij = -ji = k, \\ jk = -kj = i, \quad ki = -ik = j \end{aligned} \quad (9)$$

である。

ニューラルネットの高次元への拡張として、神経細胞モデルや学習方法にさまざまなものが提案されている。本稿では四元数を対象とし、最も単純な各部独立型の神経細胞モデルと、それに基づく誤差逆伝搬学習法を紹介する。

各部独立型のニューロンモデルでは、入力の重み付き総和を $s = s_R + is_I + js_J + ks_K$ とすると、

$$f_Q(s) = f(s_R) + if(s_I) + jf(s_J) + kf(s_K),$$

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (10)$$

と表現する。四元数学習入力に対する出力層 $r$ 番目の四元数出力 $y_r$ とその四元数教師値 $d_r$ に対し、誤差関数を

$$E = \frac{1}{2} \sum_r (d_r - y_r) \overline{(d_r - y_r)} \quad (11)$$

と定義する。 $\bar{x}$ は $x$ の共役四元数を表す。中間層 $q$ 番目の四元数出力 $v_q$ と、中間層 $q$ 番目の素子

と出力層 $r$ 番目の素子間の四元数荷重を $w_{rq}$ とし、出力層 $r$ 番目の素子への入力 $u_p$ の総和を表す四元数荷重和項を $S_r$ とすると、

$$y_r = f_Q(S_r), \quad S_r = \sum_q w_{rq} \cdot v_q, \quad (12)$$

と表せる。さらに、入力層 $p$ 番目の四元数出力 $u_p$ と、入力層 $p$ 番目素子と中間層 $q$ 番目素子の間の四元数荷重を $w_{qp}$ 、中間層 $q$ 番目の素子への入力 $v_q$ の総和を表す四元数荷重和項

$S_q = S_{qR} + iS_{qI} + jS_{qJ} + kS_{qK}$ に対し、

$$v_q = f(S_{qR}) + if(S_{qI}) + jf(S_{qJ}) + kf(S_{qK}),$$

$$S_q = \sum_p w_{qp} \cdot u_p, \quad (13)$$

と表せる。四元数荷重各成分に対する誤差関数の傾きの逆方向に微小移動を繰り返すことで、誤差 $E$ の最小化を行い、中間層と出力層の間の荷重を、

$$\begin{aligned} w_{rq}(t+1) &= w_{rq}(t) + \eta \delta_r \cdot v_q, \\ \delta_r &= (d_r - y_r) \cdot y_r \cdot (1 - y_r) \end{aligned} \quad (14)$$

入力層と中間層の間の荷重を、

$$\begin{aligned} w_{qp}(t+1) &= w_{qp}(t) + \eta \delta_q \cdot u_p \\ \delta_q &= v_q \cdot (1 - v_q) \cdot \sum_r \delta_r \cdot w_{rq} \end{aligned} \quad (15)$$

に従って、それぞれの四元数荷重を繰り返し修正する。ここで、 $x \cdot y$ で四元数各成分の積を、 $x \cdot y$ で四元数の積を表すことにする。 $\eta$ は微小係数である。これによって、四元数入力に対し与えられた四元数教師値を出力するような関係を学習する。なお、上記の手順や式において、四元数の実数部と最初の虚数部のみを考えれば、複素ニューラルネットの学習法となる。さらに上記の手順は、複素数と四元数だけでなく、一般的な高次元数に拡張可能である。

## 5. 計算機実験

高次元ニューラルネットの効果を示すために、高次元写像関係の学習を、高次元ニューラルネットと実数型ニューラルネットで比較する。

まず、複素平面上に与えた写像関係の学習を、複素ニューラルネット (Complex-valued Neural Network: CNN) と実数型ニューラルネット (Real-valued Neural Network: RNN) で行い比較する。学習データは、原点を通る直線上の点と、点 (0.1, 0.1) を中心に  $\theta = \pi/6$  回転した点を入出力として学習する。推定時には、 $ax^2/a^2 + bx^2/b^2 = 1$  ( $a = 0.8, b = 0.4$ ) の楕円上の点を入力として与え、対応する出力を推定する。これらを図4に示す。学習・推定には、入力層、中間層、出力層がそれぞれ1, 10, 1のCNNと、それぞれ2, 20, 2のRNNを用いた。学習係数 $\eta$ はともに0.01で、出力誤差が0.00001を下回るまで荷重の繰り返し修正を行い、学習の収束を確認した。

推定結果を図5に示す。CNNでは、回転移動を正しく学習し、楕円上に与えた点を正しく写像しているのに対し、RNNでは正しく推定されていない。これは、CNNでは、複素型の変換に拘束されるため、次元の低い学習データからも学習が可能であるのに対して、RNNではその変換を学習によって獲得できていないためである。

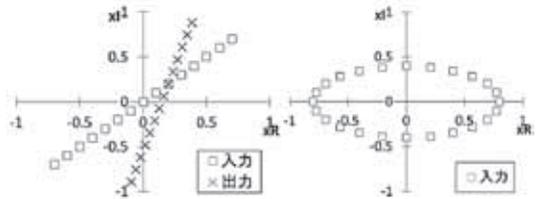


図4 複素平面上のデータ

(a) 学習入出力, (b) 推定対象入力

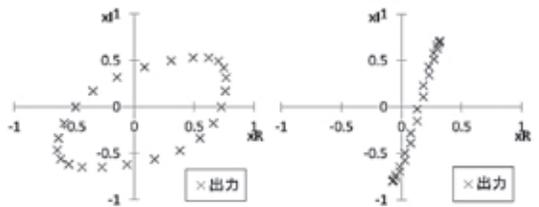


図5 推定結果 (a) CNN, (b) RNN

次に、3次元空間内に与えた写像関係の学習を、四元数ニューラルネットワーク(Quaternion-valued Neural Network: QNN)とRNNで行う。具体的には、原点を通り方向ベクトル(1,1,1)に平行な直線の周りの角度 $\theta = \pi/6$ の回転移動を考える。なお、四元数の実部はすべて0として行った。

学習データは原点を通る直線上の点と、それらを回転移動した点を入出力として学習する。推定時には、

$$x_2^2/a^2 + x_3^2/b^2 + x_4^2/c^2 = 1, \quad (a = 0.8, b = 0.2, c = 0.2)$$

の楕円体上の点を入力として与え、対応する出力を推定する。これらを図6に示す。学習・推定には、入力層、中間層、出力層がそれぞれ1, 10, 1のQNNと、それぞれ4, 40, 4のRNNを用いた。学習係数 $\eta$ はともに0.0005で、出力誤差が0.0001を下回るまで荷重の繰り返し修正を行い、ともに学習の収束を確認した。

推定結果を図7に示す。QNNでは、回転移動を正しく学習し、楕円体上に与えた点を推定しているのに対し、RNNでは正しく推定されていない。QNNにおいては、四元数間の変換に拘束されるため、次元の低い学習データからも十分に学習が可能である。

多層型ニューラルネットワークは、理論的にはどんな非線形変換でも学習が可能であるため、十分な中間層を用意し、次元の高い学習データを与えれば、実数型でも学習は可能である。例えば、2次元平面上や3次元空間内の非線形写像関係を、平面上や空間内に分布するデータから学習できる場合は、実数型でも十分に学習できる。しかし、高次元空間内の学習を効率よく行える点が、高次元ニューラルネットワークの有効性の1つであると考えられる。

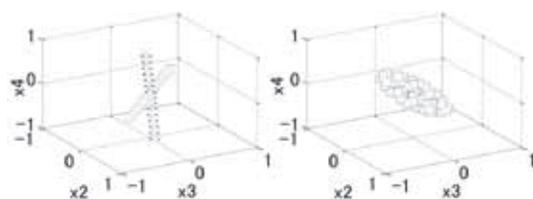


図6 3次元空間内のデータ  
(a) 学習入出力, (b) 推定対象入力

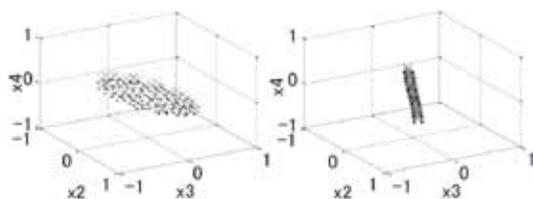


図7 推定結果 (a) QNN, (b) RNN

## 6. まとめ

本稿では、ニューラルネットワークの特徴や原理について、多層型ニューラルネットワークを中心に説明した。さらに、高次元数への拡張について、その原理とともに計算機実験の結果を元に効果の1つを示した。高次元数はさまざまな工学的な場面で用いられるため、その非線形写像関係を学習できる高次元ニューラルネットワークの活用範囲も広がるものと考えられる。

## 参考文献

- (1) 小杉幸夫, 「神経回路システム—基礎と応用—」 コロナ社 (1995年)
- (2) 廣瀬 明, 「複素ニューラルネットワーク」 臨時別冊・数理科学, サイエンス社 (2005年)
- (3) 松井, 磯川, 西村, 「四元数ニューロコンピューティングとその応用」 計測と制御, 第51巻第4号, pp.358 - 363, (2012年)
- (4) 黒江, 新田, E. Hitzer 「クリフォード代数の応用—サーベイ—」 計測と制御, 第51巻第4号, pp.328 - 337, (2012年)