

第5回 ニューラルネットワークと複雑系

時永 祥三

1. はじめに

ニューラルネットワーク（以下では、適宜 NN と略する）の構成理論は、本来、複雑系の理論とは直接的に関連ない分野として発展してきた。しかし、最近、ニューロンにカオス理論をとりいれて能力を高める試みがなされ、また、ニューラルネットワークは、もともと多数の単純なニューロンの集合により、複雑な情報処理の機能を作りだすものであり、複雑系の理論におけるエージェントに通じるものがあることなどを考慮して、とりあげることにした。以下では、次のような分野に関して述べる。

- (1) 階層型 NN による企業分析（倒産予測など）
- (2) 結合型 NN による最適化、カオス NN による記憶
- (3) CNN による進行波のシミュレーション

2. 階層型 NN

2.1 階層型 NN の構成と逆伝播法

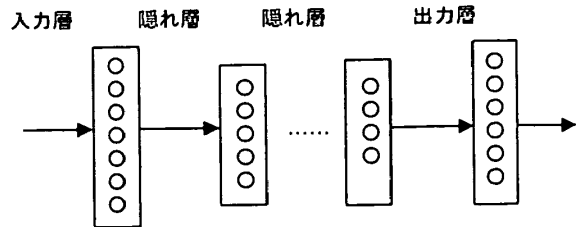
いま、図1に示すように、ユニット（ニューロン）の集合からなる層を N 層作り、 $n-1$ 番目の層に含まれる第 j 番目ユニットと、 n 番目の層に含まれる第 i 番目のユニットが、ある重み w_{ij}^{n-1} をもった回線により結合されているとする。第1層はニューラルネットワークへ送られる入力信号を受け付ける入力端子であり、最後の第 N 層はニューラルネットワークにより解析された出力信号を取り出す出力端子である。

いま、第 $n-1$ 層の第 j 番目のユニットの出力信号を x_j^{n-1} とし、これが第 n 層の第 i 番目のユニットへの入力となっているので、入力信号の和（ユニット i の状態 u_i^n ）は以下のように記述される。

$$u_i^n = \sum w_{ij}^{n-1} x_j^{n-1} \quad (1)$$

ときなが しょうぞう

九州大学 大学院経済学研究院経済工学部門
〒812-8581 福岡市東区箱崎 6-19-1



ニューラルネットワークのユニット結合

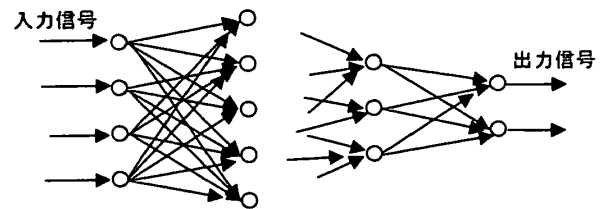


図1 階層型 NN の構成

この入力信号の和は、一定のしきい値 h_i を持った関数により制御されて出力される。第 n 層の第 i 番目のニューロンの出力値 x_i^n は次のように計算される。

$$x_i^n = f(u_i^n - h_i^n) \quad (2)$$

ここで、関数 $f(\cdot)$ はユニットの動作を規定する特性であり、通常は数値計算上の収束などを考慮して次のシグモイド関数が用いられる。

$$f(x) = 1 / (1 + \exp(a - x)) \quad (3)$$

ここで a はシグモイド関数に含まれるパラメータで適当に（ゼロでもよい）に設定しておく。

準備された学習データを用いて NN を調整する手順（重みとしきい値の最適化）は、逆伝播法（back propagation algorithm）として整理されている[10]。NN における学習とは、ネットワークへ特定の入力（第1層のユニットへの入力 x_i^1 ）を加えたときに、対応して望ましい出力信号（第 N 層の出力信号 x_i^N ）が得られるように、ネットワークのパラメータ w_{ij}^{n-1} および h_i^n を調整していくことである。

入力信号 x_i^1 、そのときに得られるべき出力信号 x_i^N との組はネットワークに対する教師信号と呼ばれる。教師信号の組を多数用意し、これらを繰り返しネット

ワークに与えることにより、学習により最適なパラメータ値が計算される。

最小化すべき評価関数をネットワークの出力と教師信号との間の最小2乗誤差とすると、次に示される逆伝播法による最適化の式が得られる。出力層 N のユニット i から戻される学習信号は、出力層からの出力値 x_i^n と教師信号 d_i^n との差を使って次のようになる。

$$\delta_i^n = (d_i^n - x_i^n)g(u_i^n),$$

$$g(x) = f(x)(1 - f(x)) \quad (4)$$

第 n 層のユニット i から第 $n-1$ 層のユニットへ向かって戻される学習信号は

$$\delta_i^n = g(u_i^n) \sum_k \delta_k^{n+1} w_{k,i}^{n+1,n} \quad (5)$$

以上の量を用いて、重みとしきい値との修正は、計算の現在のステップと次のステップを t , $t+1$ とすると、次のようになる（導出は省略する）。

$$\Delta w_{i,j}^{n,n-1}(t) = \eta \delta_i^n x_j^{n-1} + \alpha \Delta w_{i,j}^{n,n-1}(t-1) \quad (6)$$

$$w_{i,j}^{n,n-1}(t+1) = w_{i,j}^{n,n-1}(t) + \Delta w_{i,j}^{n,n-1}(t) \quad (7)$$

$$\Delta h_i^n(t) = \eta \delta_i^n + \alpha \Delta h_i^n(t-1) \quad (8)$$

$$h_i^n(t+1) = h_i^n(t) + \Delta h_i^n(t) \quad (9)$$

ここで収束のためのパラメータ α, η は通常1以下であり、 $\alpha=0.5, \eta=0.5$ 程度の値を用いる。

2.2 階層型 NN による倒産予測

倒産予測は1940年代にAltmanらにより開始された伝統的な経営分析の手法であり、代表的な財務指標を選択して、これらにある係数をかけて加えた関数（線形結合による判別関数）を計算し、この値があるしきい値より大きくなれば良好な企業であり、そうでない場合には倒産の危険性があると判断する方法である[2, 9]。これらの判別関数が、財務指標の線形結合からなる関数であるのに対して、NNを用いた判別分析では、非線形の判別関数となるので、判別の能力が向上することが期待できる[8, 11, 12]。以下では、日本の戦後の倒産企業のデータをもとにして、階層型NNの判別能力を多変量解析の結果と比較検討してみる。

調査すればわかるように、1956年以降、1995年まで一部上場企業のうち148社の企業が倒産している。倒産した企業に対して、同じ条件に置かれながら倒産しなかった企業（健全企業）を抽出し、これらに対比データとして確定する必要がある。これをペアサンプルとよぶ。まず、倒産した企業と同じ業種に属する企業であることが必要であり、その次に、倒産発生時点で倒産企業と同じ程度の資本金を有していた企業をペ

アサンプルとして設定する。これが基本的な検索条件になる。

このように企業の財務データが入手できたら、財務指標の分布の形状を統計解析し、形状が極端なもの（1つに集中しているなど）は入力データの候補から除外する。この場合、同時に、企業の選択についても検討を加え、あまりにも分布からはずれている企業（外れ値）はサンプルから除外する。

外的基準として、倒産企業にはゼロ、健全企業には1を与え、出力がそれぞれ、できるだけゼロ、1に近くなるように重みの調整がなされる。一方、学習が終了したあとに判別が不明な企業の財務指標を入力した場合には、NNの出力は整数ではなく実数である。出力はゼロと1の間にあるので、この値が0~0.5なら倒産、0.5~1なら健全といったような判断を行う。

ニューラルネットワークの判別能力を評価するには、学習に用いるサンプルと判別のテストを実施するサンプルを選択する必要がある。本来は、これらのサンプルは重複しない企業サンプルであることが必要である。

また、倒産が発生した年度から数年前のデータまで全部用いることにより、性能が向上することも期待できる。以上のようなことを考慮し、次のようなサンプルの組合せを用いて評価実験を行った。

（ケース1）単年度の財務データによる倒産予測
単年度の財務指標を用いて倒産予測を行う。

（ケース2）3年間の財務指標による倒産予測
倒産の数年前から経営悪化の傾向があると仮定し、倒産の前の3年間の財務指標を用いて倒産予測を行う。したがって入力端子の数はケース1の3倍となる。

これらのケースごとの判別結果の一部を表1に示している。結果は、ニューラルネットワークの出力（その離散化）と、外的基準とが一致する場合を正しい判別として、テストに用いた全部のサンプルに対するこの比率を判別率としている。なお、比較のため、通常

表1 ケース2の倒産予測の性能（認識率%）

	倒産企業と予測	健全企業と予測
倒産企業	96%	4%
健全企業	10%	90%

表2 判別分析（多変量解析）による予測（ケース2）

	倒産企業と予測	健全企業と予測
倒産企業	82%	18%
健全企業	22%	78%

の多変量解析により倒産予測をした結果を表2に示している。結果から分かるように表2による判別は極めて良好である。

このような倒産予測のほかに、財務指標を入力、転換社債の格付を出力とするシステムを構成することができる[6]。一般に、複数の格付を同時に判別する必要があるので、例えば、ランクが4段階である場合に、これらのランクの外的基準を0.125, 0.375, 0.625, 0.875という代表値にしておき、NNの学習を行う。格付が未知の企業の財務指標を入力した場合のNNの出力による認識(判別)では、0.0~0.25をランク1に割り当てるなどの離散化を行う。これにより債券の格付の予備的な判断を行うシステムが構成できる。

また、時系列を生成するシステムの関数を、NNが近似できると仮定し、株価や為替レートを予測する方法が開発されている[15, 16]。

3. 結合型 NN とその応用

3.1 結合型 NN による最適化問題の解法

結合型ニューラルネットワークは、図2に示すように自分自身へのフィードバックを含む多入力-出力のニューロンからなるネットワークであり、最適なニューロンの状態に達したときのニューロンの集合が、与えられた問題に対する解を与えている。 w_{ij} はユニット j からユニット i への結合の重み(結合係数)である。ニューラルネットワークの動作は時刻を1つずつ進めることにより変更されていく。ニューロンの状態を表す変数を x_i とし、最終的には1かゼロをとる。まず、入力の値が与えられていると仮定すると、ニューロンへの入力合計され、しきい値による演算を実行した結果が出力の値とされる。

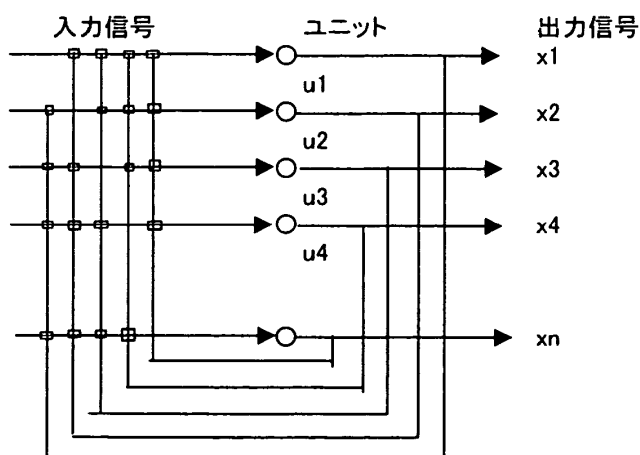


図2 結合型ニューラルネットワークの構成

$$u_i = \sum_j w_{ij} x_j \quad (10)$$

これで現在の時刻における計算は終了する。このとき、 i 番目のニューロンは同期的に、現在の状態 x_i を、次に規則に従って変更する。

$$u_i - h_i > 0 \quad \text{なら} \quad x_i \text{ を } 1 \text{ にする}$$

$$u_i - h_i < 0 \quad \text{なら} \quad x_i \text{ を } 0 \text{ にする}$$

$$u_i - h_i = 0 \quad \text{なら} \quad x_i \text{ を変化させない}$$

次の時刻には、出力(状態) x_i は結合された先のニューロンの入力として伝達される。伝達された出力は、この時刻における新しい入力となるので、再度、ニューロンにおける入力合計の計算、しきい演算が実施される。この繰り返しにより、順次、入力の値の更新が行われていく。

このような動作を繰り返すことにより、次に示すエネルギーが最小になることが示されている。

$$E = -0.5 \sum_{ij} w_{ij} x_i x_j - \sum_i h_i x_i \quad (11)$$

したがって、与えられた最適化問題における解と状態変数 x_i 、問題の構造と重みとの対応関係をうまくとると、状態を更新して最終ステップにいたった段階で、問題の解を求めることができる。

実際には状態の値をゼロか1に切り替えるのは安定的ではないので、次のような変換を用いた演算で代用される。

$$u_i = \sum_j w_{ij} x_j - h_i \quad (12)$$

$$x_i = 0.5(1 + \tanh(u_i/0.5)) \quad (13)$$

この式に従ってユニットの状態が遷移していく。最終的に、これ以上遷移が起こらない状態になったら、次の規則で状態をゼロか1に読み替える。

$$x_i \geq 0.5 \quad \text{なら} \quad x_i \text{ を } 1 \text{ にする}$$

$$x_i < 0.5 \quad \text{なら} \quad x_i \text{ を } 0 \text{ にする}$$

例題として、次のような連立1次方程式の解法を考える。 $5x_1 + 8x_2 = 47$, $3x_1 + 2x_2 = 17$ 。連立1次方程式は、変数 x_i に値を与えたときに、左辺と右辺の値の差が小さくなる方向へ変数の値を修正していけば、最終的に解が求まる。表現の都合上、解を次のような5つの、1か0をとる数値ビットの合計として表すことにする。

$$x_1 = X_{11} + X_{12} + X_{13} + X_{14} + X_{15}$$

$$x_2 = X_{21} + X_{22} + X_{23} + X_{24} + X_{25}$$

連立方程式の左辺と右辺の差である次式がエネルギー関数となる。

$$E = -0.5 \sum_i \sum_j \sum_k \sum_l w_{ij,kl} X_{ij} X_{kl} - \sum_i \sum_j h_{ij} X_{ij} \quad (14)$$

一方、最小化したい目的関数は次のものである。

$$|Ax - b|^2 \rightarrow \min \quad (15)$$

ここで、 $A=[a_{ij}]$, $b=(b_1, b_2)^T$ は方程式の左辺の係数行列、右辺の係数行列である。式(15)を展開して、 X_{ij} の1次、2次の項を整理して式(14)と比較すると、しきい値と重みは、次のようになることが分かる。

$$h_{jm} = 2 \sum_i b_i a_{ij} \quad (16)$$

$$w_{jm,kl} = -2 \sum_i a_{ij} a_{ik} \quad (17)$$

これらを用いた状態更新の関係式は、次のようになる。

$$u_{ij} = \sum_k \sum_l w_{ij,kl} X_{kl} + h_{ij} \quad (18)$$

$$X_{ij} = 0.5(1 + \tanh(u_{ij}/0.5)) \quad (19)$$

最終的には、解 $x_1=3$, $x_2=4$ が正確に計算される。

結合型 NN は、生産計画など最適な資源配分を求める問題のほか、商品の配送問題（理論的には巡回セールスマン問題として定式化される）、あるいは生産スケジューリングなど、いわゆる組合せ問題を解く場合に用いられる[13]。

3.2 カオス NN と連想記憶

結合型 NN の1つの拡張としてカオス NN が提案されており、時系列や2次元平面上のパターンを記憶して、連想する機能をもつシステムとして研究されている[1]。カオス NN は類似したパターンを検索する機能をもっているため、似たような株価パターンを検出するような場合への応用が可能であろう。

カオス NN は、次のような方程式で記述される。

$$y_i(t+1) = ky_i(t) + \sum_{j=1}^N W_{ij} x_j(t) - \alpha x_i(t) + a_i \quad (20)$$

$$x_i(t+1) = 1/[1 + \exp(-y_i(t+1)/\epsilon)] \quad (21)$$

添字の i ($i=1, 2, \dots, N$) はニューロンの番号を表し、 W_{ij} はニューロン j から i への結合係数である。

カオス NN は、通常の NN と異なり、時間的に変化する（動的な）特性を持っている。いま、式(20)において、重みをゼロ ($W_{ij}=0$) とし、単独のカオス NN を考察する。次のようにパラメータを設定する。

$$k=0.7, \alpha=1.0, \epsilon=0.02$$

定数である a_i を変化させると、システムの状態である $y_i(t)$ は大きく変動する。いわゆる、カオス的な挙動を示す。これを分岐点図として図3に示す。 a_i のとり方によって、状態は、一定の値となったり、複数の値をとったり、周期的な変動をしたりする。階層型 NN、結合型 NN では、このような状態の振動は起こらない（あるいは望ましくない）。

このような動的な特性はカオス NN によるパター

状態 $y_i(t)$ の取り得る値

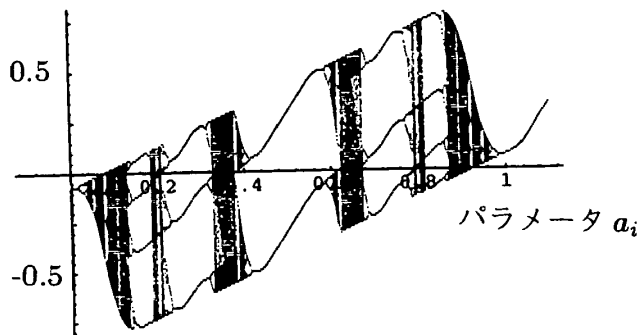


図3 カオス NN における分岐過程

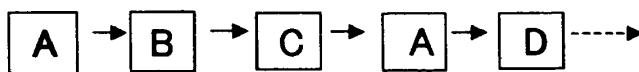


図4 カオス NN におけるパターン記憶と連想

ンの記憶に利用される。いま、100個のカオス NN を 10×10 の平面に配置しておき、それぞれのニューロンの状態が画像の濃淡を表すとしておく。このカオス NN に、例えば、A, B, C, D という4つの文字を記憶させることができる。4つの2次元的なパターンをカオス NN に記憶させる場合には、結合係数 W_{ij} を、これら4つのパターンの自己相関係数を用いて計算しておく。カオス NN は時間の経過とともに、不規則なカオス的な変動を開始する。パラメータを調整すると、図4に概略を示すように、この4つのパターンが、 10×10 の平面に次々と現れる。

4. CNN による拡散分析

4.1 CNN の原理

カリフォルニア大学の Chua 教授らにより提案されている CNN (Cellular Neural Network) による2次元的な拡散モデルによるパターン生成の分析について述べる[3~5]。CNN は、もともと、2次元的なオートマトンである、セルラーオートマトン (Cellular Automaton) を拡張した概念として提案されたものである。しかし、そのパラメータの取り方によりカオス的な挙動が発生することで注目を集めることとなった。CNN はニューロンの挙動を微分方程式で記述したものであり、次の方程式となる。2次元平面上のある座標 ij のセル $c(i, j)$ について

$$dx_{ij}/dt = -x_{ij} + \sum a_{kl} f(x_{kl}) + \sum b_{kl} u_{kl} + z_{ij} \quad (22)$$

$$y_{ij} = f(x_{ij}) \quad (23)$$

$$f(x) = 0.2(|x+1| - |x-1|) \quad (24)$$

ここで、 x_{ij} , u_{kl} , y_{ij} はニューロンにおける状態、入力、

出力である。\$z_{ij}\$ はしきい値、\$f(x)\$ は変換関数、\$a_{kl}\$、\$b_{kl}\$ は重み係数である。なお、式(2)の合計をとる範囲を示す添字 \$k, l\$ は、座標 \$(i, j)\$ から一定の半径 \$r\$ の範囲内にあるセルを意味する。

この方程式を用いた応用例としては、画像信号を入力として与えた場合に、その輪郭（エッジ）部分を自動的に抽出するなどの処理がある。

一方、入力をゼロとした場合の自律的な挙動についても応用が示される。この CNN は波の進行を表現することができ、ネットワークにおける障害の拡散や、企業の関係関係の伝搬を表現するのに都合がよい[14]。以下では、このケースを考察する。

$$dX_{ij}/dt = f(X_{ij}) + D_{ij}\nabla^2 X_{ij} \quad (25)$$

ここで、変数のベクトルは \$X_{ij} = (u_{ij}, v_{ij}, \dots)\$ であるとし、変数 \$u_{ij}, v_{ij}, \dots\$ はセル \$c(i, j)\$ の内部状態を表す（添字 \$i, j\$ は 2次元平面における座標）。ここで、ラプラシアン \$\nabla^2\$ を次のような近隣のセルの状態を用いた近似で置き換える。

$$\nabla^2 u_{ij} \rightarrow u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{ij} \quad (26)$$

以下では \$Diff(u_{ij}) = u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{ij}\$ と表す。次の 2 つの CNN を考察する。

(CNN-1)

$$du_{ij}/dt = \alpha[v_{ij} - f(u_{ij})] + D_u Diff(u_{ij}) \quad (27)$$

$$dv_{ij}/dt = u_{ij} - v_{ij} + w_{ij} \quad (28)$$

$$dw_{ij}/dt = -\beta v_{ij} \quad (29)$$

$$f(x) = 0.5[(s_1 + s_2)x + (s_0 - s_1) \times (|x+1| - |x-1|)] + \epsilon \quad (30)$$

$$\alpha = 9, \beta = 30, s_1 = s_2 = 2/7, s_0 = -1/7, \epsilon = 1/14.$$

(CNN-3)

$$du_{ij}/dt = \alpha[v_{ij} - u_{ij} - g(u_{ij})] + D_u Diff(u_{ij}) \quad (31)$$

$$dv_{ij}/dt = u_{ij} - v_{ij} + w_{ij} + D_v Diff(v_{ij}) \quad (32)$$

$$dw_{ij}/dt = -\beta v_{ij} + D_w Diff(w_{ij}) \quad (33)$$

$$g(x) = s_1 x + 0.5(s_0 - s_1)(|x+1| - |x-1|) \quad (34)$$

$$\alpha = 9, \beta = 19, s_0 = -1.143, s_1 = 0.714.$$

CNN-1 は次の 2 つの安定状態（不動点）をもつ。

$$S_A = (u_a, v_a, w_a) = (-1.25, 0, 1.25)$$

$$S_B = (u_b, v_b, w_b) = (1.75, 0, -1.75)$$

また、CNN-3 は安定な limit cycle（周期的な軌道）をもつ。

CNN のような離散的な結合素子の配列（discrete coupled excitable cells）によって進行波を表現する場合において、拡散係数 \$D\$ が小さい場合には進行波が阻害される（propagation failure）ことが知られている[7]。逆に、係数がどれ位大きくないと進行波と

ならないかを判断できる。これまで、1次元のセル配列における進行波の存在条件について導出されている。

以下では、CNN-1 における進行波の例をとりあげる。考察するセル範囲の 1 か所、例えばセル \$c(10, 10)\$ を含むすべてのセルの初期値を、1 つの安定均衡点 \$S_A\$ に設定しておき、シミュレーションを開始する。次に、セル \$c(1, 1)\$ の状態を別の安定均衡点 \$S_B\$ に設定し、この安定均衡点がセル \$c(10, 10)\$ にまで達するかを調べる。定められた時間経過の後に、セル \$c(10, 10)\$ の場所において、現在の状態であるる値 \$S = (u, v, w)\$ と初期値である \$S_A = (u_a, v_a, w_a)\$ との変移

$$r_d = (u - u_a)^2 + (v - v_a)^2 + (w - w_a)^2 \quad (35)$$

を求め、その値が大きいか小さいかで進行波の伝搬を推定する。この変移 \$r_d\$ が小さい場合には進行波が阻止されたと考えられる。

図 5 にはこの例を示す。図 5 ではセル位置を示す \$i-j\$ 平面における状態 \$u_{ij}\$ の値を縦軸に示している。

図 5 の上の図では、拡散係数が大きいのでセルに加えられた状態の変化は一定の時間経過後はすべての

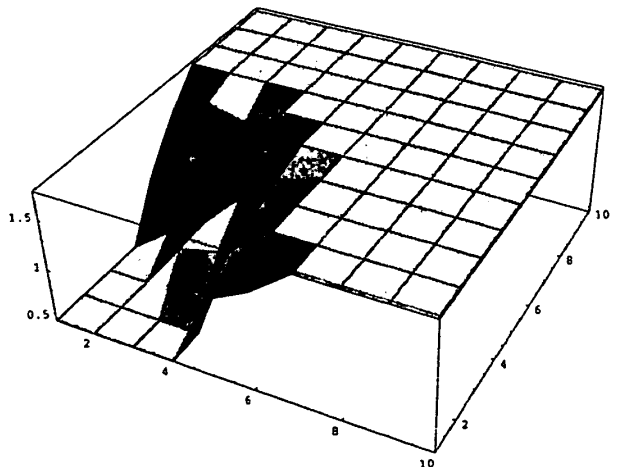
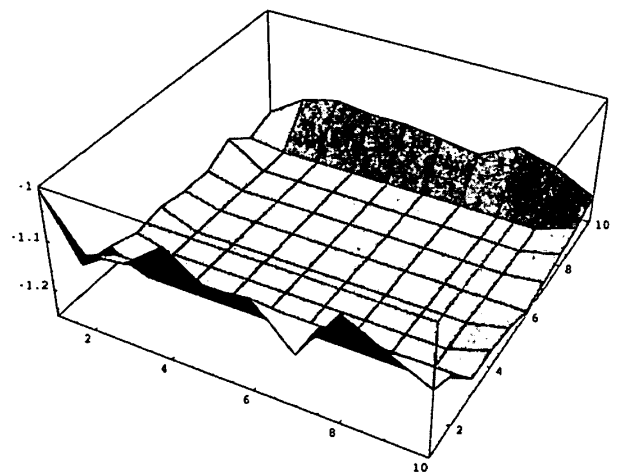


図 5 進行波の伝搬（上：伝搬，下：阻止）

セルに進行している。

一方、図5の下の図ではセルの範囲、 $c(1,1) \sim c(5,5)$ および $c(5,5) \sim c(5,1)$ にあるセルの結合を切断している拡散係数をゼロにするので、進行波はこのライン上で阻止され、これ以降には拡散していない。

4.2 CNNにおけるカオスとフィードバック制御

次に、CNNが不規則な振動をしている場合に、安定点へと移行させる制御について考察する。例えば、ネットワークの構成要素が乱雑な状態に入ったときに、正常な状態にリセットすることに用いる。目標値との差をフィードバック制御入力として加えることにより、同期化が可能である。目標とする不動点あるいはリミットサイクルを $\hat{x}(t)$ とした場合に同期化の制御は $K[\hat{x}(t) - x(t)]$ を微分方程式の右辺に加えたフィードバック制御により可能である[14]。

$$dx(t)/dt = f(x(t)) + K[\hat{x}(t) - x(t)] \quad (36)$$

ここで、 K はフィードバックゲインである。しかし、一般には観測データしか与えられていないため、システムは未知でありフィードバック制御はできない。

しかし、この問題はカオスシステムを推定あるいは近似することにより解決できる。前回の解説で説明したように、カオス時系列からそれを生成する方程式力学系を推定する方法として遺伝的プログラミング(GP: Genetic Programming)を用いる方法がある。以下では、GPによりCNNのシステム方程式の近似形を推定していると仮定する。これにより、次のような制御が可能となる[14]。

アルゴリズムとしてまとめると、次のようになる。

(ステップ1) 不動点あるいはリミットサイクルの検出

いま、関数の形が推定されているので、これを用いて不動点あるいはリミットサイクル $\hat{x}(t)$ を推定しておく。

(ステップ2) 制御入力を加える

時刻 t に $\hat{x}(t)$ に移動するには、フィードバックゲイン λ を小さい値から一定の間隔で増加させながら

$$dx(t)/dt = \hat{f}(x(t)) + \lambda[\hat{x}(t) - x(t)] \text{ の結果} \\ \rightarrow \hat{x}(t) \quad (37)$$

となるように決めればよい。 $x(t)$ が $\hat{x}(t)$ に近くなると入力 $u(t)$ は小さくなり、制御が完了すればゼロになる。

(ステップ3) 制御の終了の判断

制御を終了させるのは、目的としている均衡点、リミットサイクルと、現在の状態との誤差が、ある設定

にしたしきい値より小さくなった時点とする。状態が不動点、リミットサイクルに移行しない場合には、ステップ2へと戻って制御を繰り返す。

CNN-1について、制御される前、および制御後のセルの状態を示したのが図6~8である。ここで、セルは 10×10 の範囲で2次的に配置されているが、この図6~8では1列にならべて表示している。すなわち、 $u_{1,1}, u_{1,2}, \dots, u_{10,10}, v_{1,1}, v_{1,2}, \dots, v_{10,10}, w_{1,1}, w_{1,2}, \dots, w_{10,10}$ の順に表示している。

最初の図6では、セルの初期状態としてランダムな数値を与えた場合を示している。途中の図7は制御の

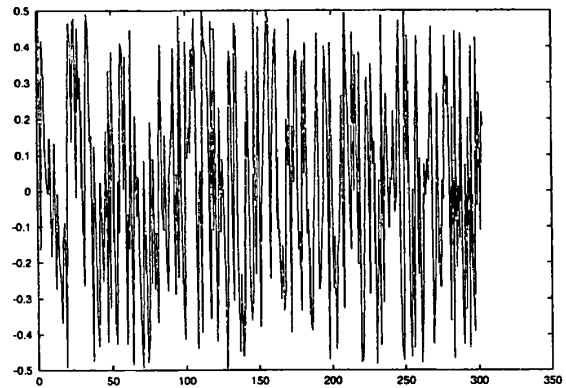


図6 セルの初期状態 (u_{ij}, v_{ij}, w_{ij})

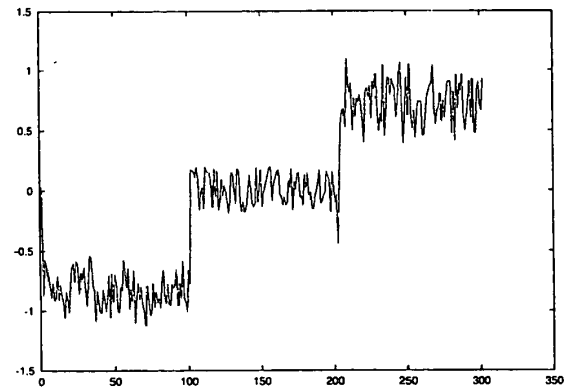


図7 一定の時間後のセルの状態 (u_{ij}, v_{ij}, w_{ij})

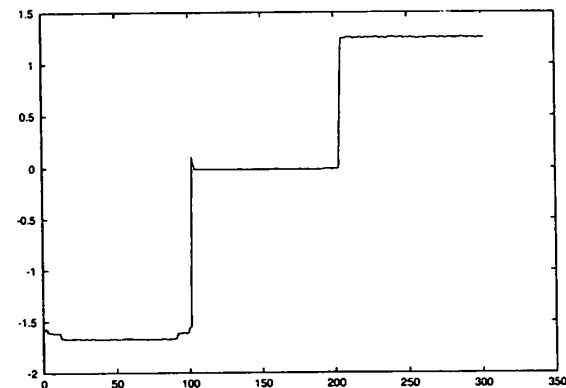


図8 セルの最終状態 (u_{ij}, v_{ij}, w_{ij})

中間の段階であり、状態が同じレベルに集まり始めている。最後の図8は、入力印加による制御を実施し安定化させた例であり、このシミュレーションにおいては、制御を始めて約30ステップ目に収束をしている。

参考文献

- [1] K. Aihara, T. Takabe and M. Toyoda: "Chaotic neural networks", Phys. Lett, A, vol. 144, pp. 333-340 (1990).
- [2] E. I. Altman: Corporate Bankruptcy in America, D. C. Heath and Company (1971) (邦訳: 南部二三雄訳『企業倒産』, 文雅堂銀行研究社 (1975)).
- [3] L. O. Chua: CNN: A Paradigm for Complexity, Scientific Publishers (1998).
- [4] L. O. Chua and T. Roska: "The CNN paradigm", IEEE Trans., Circuits Syst., vol. 40, no. 3, pp. 147-156 (1993).
- [5] L. O. Chua et. al: "Autonomous cellular neural networks: A unified paradigm for pattern formation and active wave propagation", IEEE Trans. Circuits Syst., vol. 42, no. 10, pp. 559-578 (1995).
- [6] S. Dutta and S. Shekhar: "Bond rating: A non-conservative application of neural networks", discussion paper, Computer Science Division, University of California, Berkeley (1989).
- [7] J. P. Keener: "Propagation and its failure in coupled systems of discrete excitable cells", SIAM J. Appl. Math, vol. 47, no. 3, pp. 556-572 (1987).
- [8] 李鋼浩, 時永祥三: "ニューラルネットワークによる経営情報解析—倒産分析と時系列解析", 経営情報学会論文誌, vol. 1, no. 2, pp. 32-43 (1991).
- [9] B. Lev: Financial Statement Analysis: A new Approach", Prentice-Hall (1974) (邦訳: 柴川林也, 寺田徳訳『現代財務諸表分析』, 東洋経済新報社 (1978)).
- [10] D. E. Rumelhart, G. E. Hinton and R. J. Williams: "Learning internal representation by error propagation, Paralell distributed processing", McClelland and the PDP Research Group ed. MIT Press (1989).
- [11] S. Tokinaga and K. Lee: "A knowledge based system for corporate financial analysis based upon neural networks", Proc. APORS '94, pp. 180-187 (1994).
- [12] S. Tokinaga: "Learning models in the neural networks in comparison to the learning classifier system with the genetic algorithm—applications in financial management", Proc. of NOLTA '91, pp. 73-76 (1991).
- [13] 時永祥三: 『複雑系による経済モデル分析』, 九州大学出版会 (2000).
- [14] 矢加部正幸, 時永祥三: "CNNによるネットワーク決済リスク拡散のモデル化と応用", 日本オペレーションズリサーチ学会 2001 春期全国大会予稿 (2001).
- [15] "予測や検出, 制御な用途で実用システムの開発に挑む先進ユーザ企業", 日経インテリジェントシステム別冊 1992 年夏号, pp. 52-57 (1992).
- [16] "様々な相場展開に対応できるニューロ応用相場予測システム", 日経インテリジェントシステム別冊 1992 年夏号, pp. 64-71 (1992).