

Title	A Study on Information Economics
Author	橋本, 文彦
Citation	経済学雑誌. 別冊. 99 卷 1 号
Issue Date	1998-04
ISSN	0451-6281
Type	Learning Material
Textversion	Publisher
Publisher	大阪市立大学経済学会
Description	

Placed on: Osaka City University Repository

A Study on Informational Economics

橋本文彦

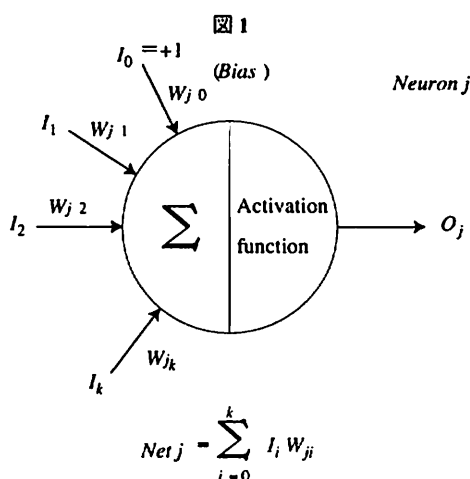
1 はじめに

何らかの未知の要因によって時系列的に変化するデータを扱おうとする場合、その時系列を生み出す要因となるパラメタを組み合わせてモデルを作成し、これを現実の観測データにうまく当てはめることで、未来の任意の時点におけるデータの値を予測しようという形式を取る手法が従来から存在している。これに対して、本研究で取り上げる Artificial Neural Network (ANN) の手法では、「適切に」設計された ANN に過去の経験データを「学習」させることで、このデータを構成するパラメタそのものが明らかでない場合でも、その学習をもとにして、未来のデータの値を予測することを可能とするものである。本研究では、このうちの「適切」な ANN とはどのようなものであり、またその「学習」はどのように行われ、どの程度の信頼性があるものかを検討するものとする。最終的には、解くべき時系列データに対して特定の最適な ANN モデルを構築することを目標とする。

2. Artificial Neural Network

Neural Network は Neuron と呼ばれる「素子」(図 1) が、ネットワーク状態に複数個組み合わせられたものである。

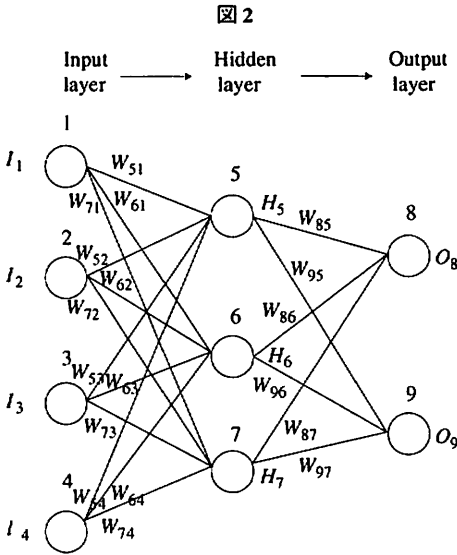
このようなネットワークは全体として「入力」と「出力」を持つ。また、個々の Neuron は複数のチャンネルから入力された情報（スカ



ラーデータ) をその時点での Neuron の内部状態に応じて「加重」し、統合した上で出力する。ある Neuron から出力されたデータは、さらに一つ上の階層の Neuron に新たな入力として伝達され、この Neuron で処理されたデータがさらに上の階層の Neuron に伝達され…、と順々に進んでいき、最終的な出力層の Neuron からネットワークの外部に対して「出力」がなされる。(図 2)

通常 Artificial Neural Network (ANN) には二つの種類のネットワークがある。その一つは「教師信号なし学習 (Unsupervised Learning)」であり、今ひとつが「教師信号付き学習 (Supervised Learning)」である。

前者の構造の ANN は、ホップフィールド型と呼ばれる ANN で代表される。この型の ANN は、すべての Neuron の出力は他の



Neuron への入力となっており、Network は外部から切り離されて閉じた形式となっている。また、解くべき問題の種類によってネットワークの構造が確定しており、Network 全体のエネルギー（活性化量）が平衡状態に達することで、多次元空間を構成するような多パラメタ問題を解く際に大きな威力を発揮するものであるが、本研究ではこの構造の ANN については取り扱わない。（ホップフィールド型のネットワークについて詳しく知りたい読者は、橋本(1995)を参照のこと）。

一方、後者の構造の ANN は「階層型 ANN」と呼ばれているもので、既に述べたように、Network の外部から「入力」信号を受け取り、Network の中の階層を順番に進んでいき、最終的に出力層から Network の外へと出力される、というように外部に対して開かれた構造を持っている。さらにこの出力値が、外部の（当該入力に対する）「現実の値」（これが「教師信号」である）と比較された上で、その誤差の大きさに応じて現在の各 Neuron 間の荷重が変更されることで、新たな入力データに備えて過去のデータが「学習」されることになる。学習の結果は、したがって Network 上の特定

の領域にではなく、「現在の荷重」という形で Network の内部に保持されるのである。このような「入力」「出力」「荷重の変更」が繰り返されることで、時系列データの学習が行われ、この Network に対して未知なる新たな入力が行われた場合にも、その学習経験を活かして出力値を予測することができることになる。

このような ANN の学習機能は、初期のニューラルネットである「パーセプトロン」にも備えられていたが、パーセプトロンの学習システムでは XOR 問題のような非線型の解を持つような学習が行えないことが Minsky らによって指摘されており、より複雑な解を持つような問題を現実的にとくためには Rummelhart & McClelland の PDP モデルにおいて示された誤差逆伝播法という学習方法を用いることになる。この学習方法（以下 BP）の学習過程は以下ようになる。

いま、先の図 2 で示した三階層からなる ANN モデルを例としてみよう。

外部からの入力 $I_1 - I_4$ が与えられ、各 Neuron 間の初期の荷重が図の値で与えられているとすると、 H_5, H_6, H_7 の出力はそれぞれ

$$H_5 = \frac{1}{1 + e^{-(w_{51}I_1 + w_{52}I_2 + w_{53}I_3 + w_{54}I_4)}}$$

$$H_6 = \frac{1}{1 + e^{-(w_{61}I_1 + w_{62}I_2 + w_{63}I_3 + w_{64}I_4)}}$$

$$H_7 = \frac{1}{1 + e^{-(w_{71}I_1 + w_{72}I_2 + w_{73}I_3 + w_{74}I_4)}}$$

として与えられる。

更に同様に、 O_8, O_9 からの出力はそれぞれ

$$O_8 = \frac{1}{1 + e^{-(w_{85}H_5 + w_{86}H_6 + w_{87}H_7)}}$$

$$O_9 = \frac{1}{1 + e^{-(w_{95}H_5 + w_{96}H_6 + w_{97}H_7)}}$$

で与えることができる。この時、 $I_1 - I_4$ の入力に対して本来あるべき値（あるいは現実に観

測された値) を T_1, T_2, \dots とすると, $E(W)$ を

$$E(W) = \frac{1}{2} \sum_{i=1}^I \sum_{j=1}^J (T_{ij} - O_{ij})^2$$

であらわしたとき, 荷重の変更量は,

$$\Delta_i w_{ji} = -\alpha \frac{\partial E_i}{\partial w_{ji}}$$

を計算すればよいことから (これは, 現在の自分の位置と, 傾斜角だけを頼りに, より低い方向に向かうことから, 「山下り法」と呼ばれている), まず, 出力層と, 隠れ層との間の荷重の変更量は,

$$\frac{\partial E_i}{\partial w_{ji}} = \frac{\partial E_i}{\partial O_{ij}} \frac{\partial O_{ij}}{\partial w_{ji}}$$

これを計算すると (計算の手順は省略)

$$\frac{\partial E_i}{\partial w_{ji}} = -(T_{ij} - O_{ij}) O_{ij} (1 - O_{ij}) H_{ij}$$

で与えられることになる。

この値をもとにして, さらに隠れ層と入力層との間の荷重は (同様の計算によって)

$$\frac{\partial E_i}{\partial w_{ji}} = H_{ij} (1 - H_{ij}) I_{ij} \sum_{k=1}^K (T_{ik} - O_{ik}) O_{ik} (1 - O_{ik}) w_{kj}$$

で与えることができる。

より大きな階層を持つような構造の ANN でも, このような形で次々と出力層の側から入力層へと向かって荷重の調整を変更していくことができる。このように, 入力層側から与えられた信号が順方向に出力層側まで進み, その出力値と現実の値との誤差が, 今度は出力層側から逆方向に進んでいき, 入力層にまでいたる逆方向の伝播が行われるという点で, この学習方法が「誤差逆伝播」型と呼ばれているのである。

BP を利用した学習方法では, α の値によって学習を完了するまでの収束の速さが変わってくる。

BP 法の持つ欠点として, 入力として与えるデータ系列によっては, 本当の収束値 (グローバル・ミニマム) ではなく, 局所的な値 (ロー

カル・ミニマム) に収束して学習を完了してしまう場合がある。このような欠点を取り除くために, 「焼きなまし法」をはじめとするさまざまな BP の改良型の学習方法が提唱されている。

3. Genetic Algorithms

前章で述べた ANN の方法の他に, 多パラメタ問題を解くための方法としてより新しい手法である Genetic Algorithms (GA) の手法がある。GA の具体的な手法は後に述べるが, ANN に比べた場合のこの手法の長所は, (1) ANN の様に問題ごとに Network を設計し直す必要がない。(2) 計算時間が短い, というものであり, 一方短所は (1) 得られた解に対する数学的な根拠がない, というのが最大のものである。

このような特徴を持つ GA の有効な利用法として考えられるのが, 先の ANN の荷重調整の方法に BP の代わりに GA を使う方法である。このような利用の仕方をすれば, Network の収束値の数学的な根拠は Neural Net の理論によって既に与えられており, 一方, 荷重の修正方法自体には数学的な根拠は不要であり, より速く確実にグローバル・ミニマムに導く方法であることだけが要請されるため, ANN の GA の両者の長所を生かすことができると考えられる。

ここで, GA の手法について簡単に紹介をしておこう。GA とは, 生物に関する「進化論」から, そこで用いられる用語と概念を借用して, 数学的な「解の進化」をプログラム上で実現しようとするものである。したがってその用語は多分に「生物学的」であるが, その意味するところは, もっぱら数学的なものである。その点には十分に注意をする必要がある。

与えられた問題を遺伝的アルゴリズムで解くためには, 当該の問題を遺伝的アルゴリズム流に解釈することから始める必要がある。このアルゴリズムでは一般に, 問題のパラメタの値

(ANN への適用問題の場合は各 Neuron 間の荷重の一つ一つがパラメタとして与えられるものと考えられる)は、「遺伝子」として解釈され、パラメタ(すなわち遺伝子座)に適当な初期値を代入した上でそれらとは少しずつ異なった初期パラメタ値組をもつ種を複数生成し、それらの種ごとに数百の個体を持つ初期集団を生成する。

この様な初期集団中の各「種」について、与えられたパラメタ組が問題に要求されている条件をどの程度満たしているのかについて、「適応度」という表現で評価を行う。この評価によって、適応度が高いと判定された種は次世代で個体の数を増やし、適応度が低いと判定された種は次世代で個体の数を減らす、もしくは絶滅するという「選択 (Selection)」をおこなう仕組みである。ただし、この条件だけでは、あらかじめ用意されていたパラメタの組み合わせ以外の解は得られないので、ここで、生き残った種にたいしてある確率で「交叉 (Cross Over)」を行う。すなわち、一つの種のパラメタ(遺伝子)のある部分と、別の種のパラメタの残りの部分とを組み合わせる新しいパラメタ組を作り出すのである。これによって、ある場合には前世代の二つの種の両方のよいところを兼ね備えた種が誕生することになるし、またある場合には二つの種の悪いところを兼ね備えた種が誕生することになる。勿論、前者の種はさらに次の世代で個体数を増やし、後者の種はいずれ絶滅することになる。さらに、ある確率で種に対して「突然変異 (Mutation)」を起こさせるものとする。すなわち、パラメタ内のある値をランダムに書き換えることで全く新奇な種を誕生させるというものである。このことによって、初期値としてランダムに決定した種の遺伝子の条件に制限されることなく、これらの種は解空間の中を目的とする最適解に向かって進化していくという事になるのである。

このようにして、初期集団から評価を経て選

択・交叉・突然変異によって新たな種の集団が生成されるが、ここまですべてを1世代として、何百世代かを繰り返していくと、次第にパラメタの組み合わせが特定のものに収束していく状況が見られる。こうして収束に至った種は、初期集団に比べて遙かに評価の高い、すなわち、目的解の条件を最適に満たしているようなパラメタの組として得ることができるのである。

わかりやすくするためにここで具体的な例を数学の問題上で探ってみよう。

いま、 $2x^2 - 4x - 6 = 0$ なる方程式の解を探索するものとする。勿論この程度の問題であれば、「二次方程式の解の公式」を用いればたどころに計算できるが、これを遺伝的アルゴリズムを用いて解いてみることにする。

いま、パラメタとして遺伝子を定める。ここでは、解を0から31の範囲の中で探すこととすると、遺伝子座は次のように定めることができる。

- 第1ビット……16の桁
- 第2ビット…… 8の桁
- 第3ビット…… 4の桁
- 第4ビット…… 2の桁
- 第5ビット…… 1の桁

このような順位のもとで、あらかじめランダムに各ビットに1と0を割り振った種を用意する。(この例では100種は多すぎるので5種とする)

- A種：1 0 1 0 1
- B種：1 0 0 1 0
- C種：0 1 0 0 1
- D種：0 1 1 0 0
- E種：0 0 0 1 0

これらが「環境に適応しているか否かを調べる」には実際にこの数値を方程式に代入して答えを調べればよい。

- A種=21：f (21) =792
- B種=18：f (18) =570

C種 = 9 : f (9) = 120

D種 = 12 : f (12) = 234

E種 = 2 : f (2) = -6

となり、E種とC種およびD種は比較的環境に適應しているがその他はあまり適應していないと見ることができる。したがって、これらの種が次世代では個体数を増加させることになるが、その際に一定の確率で互いの種の遺伝子の一部づつを交換する「交配」を行うものとする。また同様に一定の確率で互いの遺伝子の一部をランダムに書き換えることで「突然変異種」を生み出すこととする。

例えば、E種の遺伝子の前半3ビットとC種の遺伝子の後半2ビットとが交配されるとあたりに生ずる種の遺伝子は

EC種 : 0 0 0 0 1

となり、またD種の第3ビットが突然変異すると新たな種の遺伝子は

D'種 : 0 1 0 0 0

となる。

次世代ではこのような種が適当な個体数だけ配分され、再び環境による適應度の評価を受けて世代を繰り返し、次第に方程式の解の近辺に近づいていき、

0 0 0 1 1

なる遺伝子を持った種が登場すると、この遺伝子は当該方程式を満たすためにそこで計算が終了するということになるのである。

以上の方法を ANN において BP の代わりに使用することで、ANN の欠点であるローカル・ミニマムに陥らないための計算量の増大から免れることが可能となり（このような GA の手法は「数百世代」を必要とするため、長い時間を必要とするように思われるが、BP の代表的な改良型である「焼きなまし法」などでは、GA に要するよりもはるかに多くの時間（計算量）が必要となる）、また GA の欠点である数学的な根拠の不足（荷重が完璧に最適なものでなかったとしても、結果として必要なのは

ANN 出力の適切さであり、GA の欠点は、最終的な問題の欠点には結びつかない）をもカバーするものであるため、本研究で用いる学習機構として、この形式の ANN を利用するものとする。

4. Agent Models

先に述べた ANN を用いて以下のようなモデルを考察することにしよう。

すなわち、与えられた環境の中で Agent が、環境と他の Agents から情報を受け取り、意思決定し、行為する。但し、環境と他の Agents は所与のものとし、Agent はこれらに adapt するようにみずからの action を定めることができるが、その action によって環境や他の Agent に影響を与えることはできないものとする。（図 3）

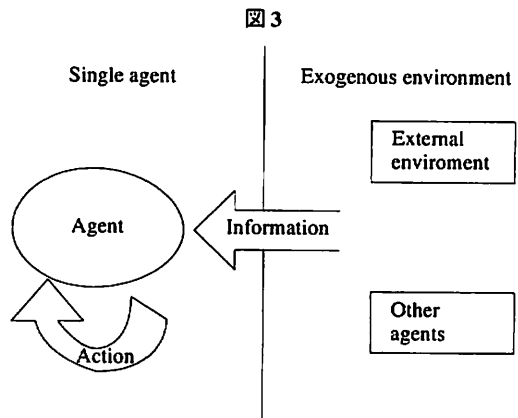


図 3

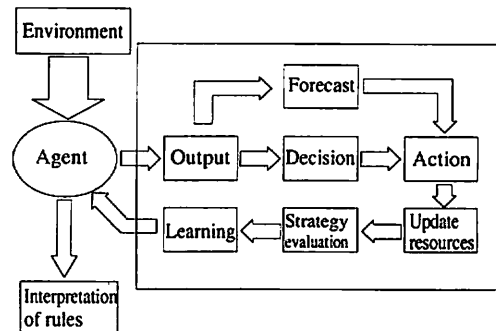
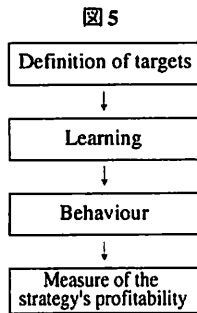


図 4

Agnentの意思決定の過程を図式化すると(図4)のようになる。ここで、すぐに分かるように、図4内の実線で囲った部分はANNの動作過程そのものであるとみなすことができる。ここで、さらに図4の一部を、より詳細に見ると(図4), 図4の最終段階の決定がGAの役割であるということになる。



このような Agent の意思決定モデルを具体的な ANN の構造に当てはめるには、しかし未だ残されたフリーパラメタの問題がある。それは、Agent が環境と他の Agent から収集した過去の情報によって学習した結果、環境の未来の値の予測値を得ることと、その予測値に対してどのような action をとるかという点を同時に扱うのか、それとも独立に扱うのかという問題である。

このような予測値と action の扱いを ANN の構造で表現したものが、以下の図6, 図7, 図8である。但し、図6において、「外部ルール」とは、「予測値が0より低いときは『待て』, 予測値が正のときは『買い』, 予測値が負のときは『売り』」などといったルールのことを指す。

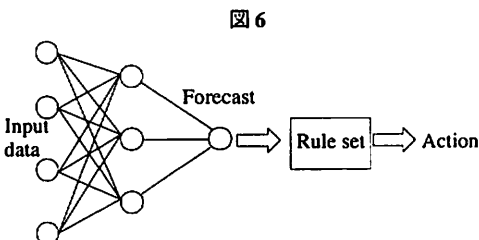


図7

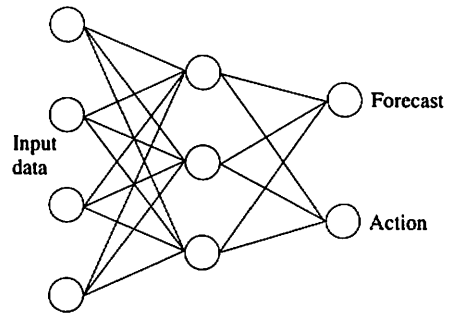
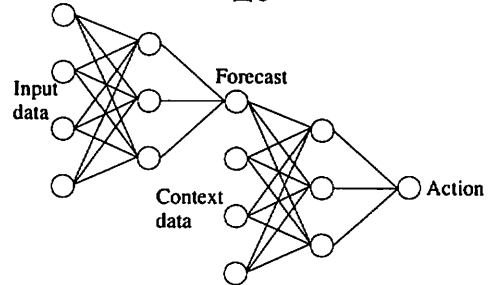


図8



以上の議論をもとに、現実の時系列データに対して、未来のデータ値を予測し、これに対して action を行う Agent を ANN でモデル化することにしよう。

- (1) まず、ANN の階層数を定める必要がある。特に最適な階層数に関する根拠がない場合には、この値を「3」とすることが多い (BP の手法を用いるためには3以上の階層を必要とする。本研究ではBPを用いるわけではないが、一般にはBPを用いることのできる最小の階層数として「3」階層型のANNが用いられることが多い)。
- (2) 各層の Neuron の数を定める。Input 層の数は入力されるデータの数であるが、特に時系列の場合は Auto Regressive なモデルを参考として、過去 n 時点分のデータを入力することが多い。また、Neuron の一つには、出力層の1時点前のデータを帰還して入力する場合がある。

このデータとして何を用いるかは、ANNの設計者に任されていることであり、逆に言えば、試行錯誤を要する部分である。Hidden層のNeuronの個数は、GAによって最適な個数を定めればよい。

- (3) 初期のNetwork内部の荷重をランダムに定めたのち、現実のデータをANNに入力し、出力を「期待されるべき出力(教師信号)」と比較する。比較の結果はGAによって、Neuron間の荷重の調整、さらにNeuronの個数の決定(荷重が0のNeuronはNetworkから切り離される)を行う。
- (4) 十分な入力データによって学習を繰り返す、ANNを訓練したのち、未知のデータ値の推測に適用する。

5. Population Models

Agent Modelでは、環境と他のAgentは外部からの所与として与えられていたが、Population Modelでは環境としての変数はAgent同士の相互作用によって導かれるものとして定められ、また他のAgentは当該Agentとの相互作用によって影響を受け、そのActionを代えうるものであると考える。このPopulation Modelには、さらに二つの場合がある。

(A) One-Population Model

このモデルは、すべてのAgentが同一の構造を持っていると考えてよい場合のモデルである。このモデルでは、通常のANNの構造を二次元化することで、さ

ほどの困難なく形式的に拡張することができる。

(B) Multi-Population Model

このモデルでは、(A)のOne Population Modelを更に一般化して、おのおののAgentがそれぞれ異なる構造を持っていると仮定する。

これによって、より一般的な構造をNeural Networkの形式で記述することができるが、そのNetworkの構造はかなり複雑なものとなる。

(注) 本稿中の図はA. Beltratti et. al. 1996(本研究で使用するテキスト)を利用した。

参考文献

- A. Beltratti, Sergio Margarita & Pietro Terna (1996), "Neural Networks for Economic and Financial Modelling", Thomson Computer Press.
- D. E. Rummelhart & J. L. McClelland (1986), "Parallel distributed processing, explorations in the microstructure of cognition", The MIT Press.
- H. Dawid (1996), "Adaptive Learning by Genetic Algorithms", Springer
- J. Kingdon (1997), "Intelligent Systems and Financial Forecasting", Springer.
- M. J. Wooldridge & N. R. Jennings (Eds.) (1995), "Intelligent Agents", Springer-Verlag.
- W. S. Macculloch & W. H. Pitts (1943), "A logical calculus of the ideas immanent in neural nets", Bull. Math. Biophys., 5, 115-133.
- 橋本文彦 (1995), 「ニューラルネットを利用したソフトバリエーションデータの解析」, 『経済学雑誌』, 第96巻, 3, 4, 60-68.