

ニューロコンピューティングを用いたカオスのデータの短期予測

創価大学大学院 木戸 和彦 (kazuhiko kido)
創価大学大学院 菊池 洋一 (yoichi kikuchi)

1. はじめに

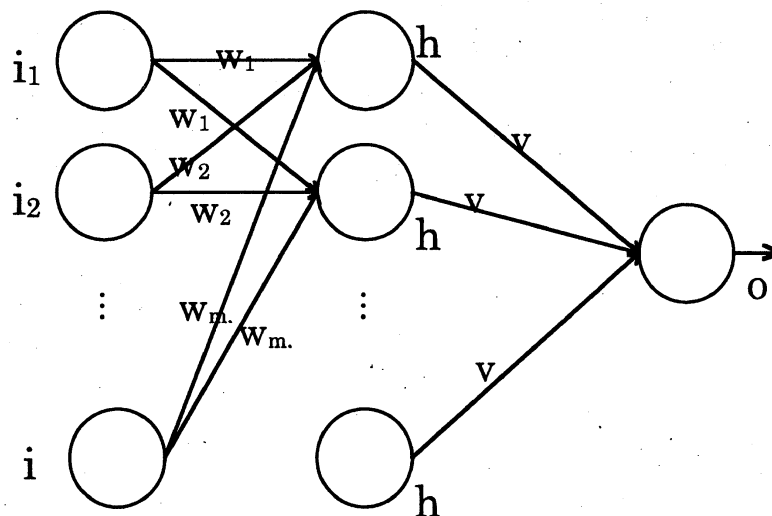
近年、カオスの研究が注目されている。その応用の1つにカオスの時系列データの短期予測問題がある。カオスの振る舞いをする時系列は、決定論に支配されているので、予測可能である。しかし、その特徴の1つに「初期値に対して鋭く依存する」というものがあり、短期間しか予測ができないということがわかっている。その短期予測をする方法についての研究を行った。

短期予測の方法には、グラムシュミットの直交化法やテセレーション法などがあるが、パラメータの取り方によっては予測不可能になったり、次元が高くなるに伴って計算時間が急激に増大するといった多数の問題がある。

そこで、比較的問題点の少ないニューロコンピューティングを用いる予測法に着目した。

2. ニューラルネットワークの構造

まず、予測を行う際に用いるニューラルネットワークの構造を示す。図の i は入力層への入力、 h は中間層からの出力、 o は出力層からの出力、 w は入力層と中間層の間の結合荷重、 v は中間層と出力層の間の結合荷重を示す。



(図 1)

このニューラルネットワークは、入力層と中間層のユニット数を m ，出力層のユニット数を 1 とした，3 層からなる階層型神経回路網であり，それぞれの層の入出力値の計算式は次のようになる．ただし，関数 f はシグモイド関数と呼ばれるもので，1 次元の場合は，式 (4) であらわされる．

入力層への入力値

$$i_k = x_k \quad (k=1, 2, \dots, m) \quad \dots\dots\dots (1)$$

中間層からの出力値

$$\begin{aligned} h_k &= f(\bar{h}_k) \\ \bar{h}_k &= \sum_{j=1}^m i_j w_{kj} \quad (k, j=1, 2, \dots, m) \quad \dots\dots\dots (2) \end{aligned}$$

出力層からの出力値

$$\begin{aligned} o_k &= f(\bar{o}_k) \\ \bar{o}_k &= \sum_{j=1}^m h_j v_k \quad (k=1, 2, \dots, m) \quad \dots\dots\dots (3) \end{aligned}$$

シグモイド関数

$$f(y) = \frac{1}{1 - e^{-y}} \quad \dots\dots\dots (4)$$

3. タケンスの埋め込み定理

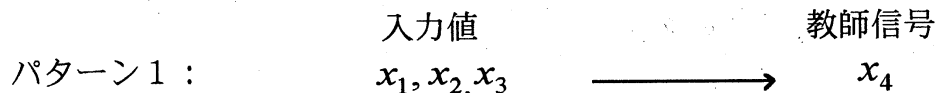
今，観測されたある時系列データ $y(t)$ から，ベクトル $(y(t), y(t-\tau), y(t-2\tau), \dots, y(t-(n-1)\tau))$ をつくる (τ は遅れ時間)．このベクトルは n 次元再構成状態空間 R^n の 1 点を示すことになる．従って t を変化させると，この n 次元再構成状態空間に軌道が描ける．もし，対象のシステムが決定論的力学系であって，観測時系列データがこの力学系の状態空間から 1 次元ユークリッド空間 R への C^1 連続写像に対応した観測系を介して得られたものと仮定すれば，この再構成軌道は元の決定論力学系の埋め込みになっている．つまり，元の力学系に何らかのアトラクタが現れているならば，再構成状態空間にはこのアトラクタの位相構造を保存したアトラクタが再現することになる．

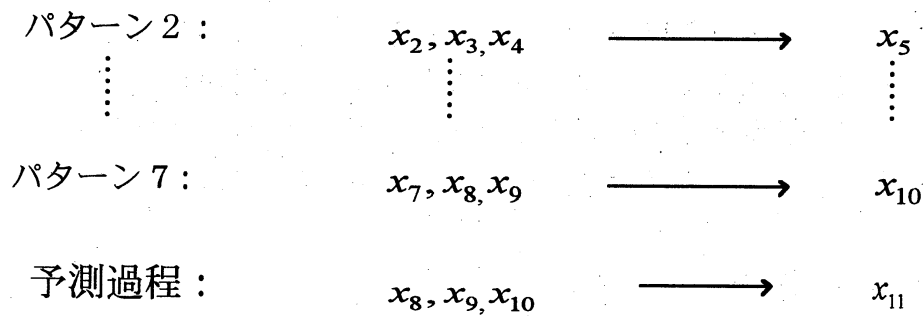
4. ニューロコンピューティングを用いた時系列データの予測法

今，ある時系列データを x_i ($i=1, 2, \dots, 10$) とし， x_i から x_{11} を予測する問題を考える．

4-1 従来の予測法

従来の予測法では，まず次のパターンを先程のニューラルネットワークに学習させる．





ここで、パターン 7 まで全ての学習を終了した時点での結合荷重を保存しておく。そして予測過程として入力値を x_8, x_9, x_{10} 、結合荷重を今保存したものとし、同じニューラルネットワークで 1 度だけ計算させる。その結果を予測値 x_{11} とする。これが、従来の予測法である。

しかし、この従来の方法では、 x_{11} を予測する時点で、それに関連するはずの x_7 以前のデータの影響がほとんど無視されるという問題点がある。

そこで、この問題を解決するために新しい方法を 2 つ提案する。

4-2 新しい予測法

まず、元の時系列データにタケンスの埋め込み定理を適用し、3次元再構成状態空間に埋め込みを行う。すると、元の時系列データは、 $[x_{i-2}, x_{i-1}, x_i]$ ($i=3, 4, \dots, n$) と書き直すことができる。

4-2-1 提案 1

ここで、最初のニューラルネットワークに学習させるが、この場合入力データが 3 次元なので、それぞれの層への入出力値の計算式は、次のようになる。

入力層への入力値

$$i_k = [x_k, x_{k+1}, x_{k+2}] \quad (k=1, 2, \dots, m-2) \quad \dots\dots\dots (5)$$

中間層からの出力値

$$\begin{aligned} h_k &= [f(\bar{h}_k), f(\bar{h}_{k+1}), f(\bar{h}_{k+2})] \\ \bar{h}_k &= \sum_{j=1}^{m-2} i_k w_{kj} \quad \dots\dots\dots (6) \\ &\quad (k, j=1, 2, \dots, m-2) \end{aligned}$$

出力層からの出力値

$$\begin{aligned} o_k &= [f(\bar{o}_k), f(\bar{o}_{k+1}), f(\bar{o}_{k+2})] \\ \bar{o}_k &= \sum_{j=1}^{m-2} h_j v_k \quad \dots\dots\dots (7) \\ &\quad (k=1, 2, \dots, m) \end{aligned}$$

シグモイド関数

$$f(y) = \frac{1}{1 - e^{-y}} \quad \dots\dots\dots (8)$$

従来の方法との相違点は、式(6)の中間層からの出力値 h と式(7)の出力層からの出力値 o を3次元扱いにするところである。

また、この場合、データが3次元なので、誤差関数はそれぞれ E_1 , E_2 , E_3 と3つ求められる。そこでそれぞれに比率を持たせて誤差関数 E を次のように決める。

$$E = \lambda E_1 + \mu E_2 + (1 - \lambda - \mu) E_3$$

$$(0 < \lambda < 1, 0 < \mu < 1)$$

ただし、数値計算をする時は λ , μ を共に 0.1 きざみで 0.9 までの全通りを計算し、最も良いものを採用する。

この計算式を使い、最初のニューラルネットワークに次のパターンを学習させる。

入力値	教師信号
パターン1: $[x_1, x_2, x_3], [x_2, x_3, x_4], [x_3, x_4, x_5]$	$[x_4, x_5, x_6]$
パターン2: $[x_2, x_3, x_4], [x_3, x_4, x_5], [x_4, x_5, x_6]$	$[x_5, x_6, x_7]$
$\vdots$	$\vdots$
パターン5: $[x_5, x_6, x_7], [x_6, x_7, x_8], [x_7, x_8, x_9]$	$[x_8, x_9, x_{10}]$

予測過程: $[x_6, x_7, x_8], [x_7, x_8, x_9], [x_8, x_9, x_{10}] \longrightarrow [x_9, x_{10}, x_{11}]$

ここで、パターン5まで全ての学習を終了した時点での結合荷重を保存しておく。そして予測過程として入力値を $[x_6, x_7, x_8], [x_7, x_8, x_9], [x_8, x_9, x_{10}]$, 結合荷重を今保存したものとし、同じニューラルネットワークで1度だけ計算させる。その結果の内、1番最後のものを予測値 x_{11} とする。

次に、埋め込み次元を 4, 5, ... と上げていき、7次元に埋め込みする。すると元の時系列データは、 $[x_{i-6}, x_{i-5}, \dots, x_i]$ ($i = 7, 8, 9, 10$) と書き直すことができる。そして、3次元の場合と同様に、次のパターンを学習させ、終了した時点での結合荷重を保存しておく。

入力値	教師信号
パターン: $[x_1, \dots, x_7], [x_2, \dots, x_8], [x_3, \dots, x_9]$	$[x_4, \dots, x_{10}]$
予測過程: $[x_2, \dots, x_8], [x_3, \dots, x_9], [x_4, \dots, x_{10}]$	$[x_5, \dots, x_{11}]$

次に、予測過程として入力値を $[x_2, \dots, x_8], [x_3, \dots, x_9], [x_4, \dots, x_{10}]$, 結合荷重を先程保存したものとして、1度だけ計算する。その結果の内、1番最後のものを予測値 x_{11} とする。

以上が提案1の予測法であるが、この方法だと、 x_{11} を予測する時、初期値以外の全てのデータの影響が繁殖され、従来の方法での問題が解決できる。

4-2-2 提案2

提案1と同様に、元の時系列データにタケンスの埋め込み定理を適用し、3次元再構成状態空間に埋め込みを行う。そして、最初のニューラルネットワークに学習させるのだが、この場合入力データが3次元なので、それぞれの層への入出力値の計算式は、次のようになる。

入力層への入力値

$$i_k = [x_k, x_{k+1}, x_{k+2}] \quad (k=1, 2, \dots, m-2) \quad \dots\dots\dots (9)$$

中間層からの出力値

$$h_k = f(\bar{h}_k)$$

$$\bar{h}_k = [\sum_{k=1}^{m-2} i_k w_{kj}, \sum_{k=1}^{m-2} i_{k+1} w_{kj}, \sum_{k=1}^{m-2} i_{k+2} w_{kj}] \quad \dots\dots\dots (10)$$

$$(k, j=1, 2, \dots, m-2)$$

出力層からの出力値

$$o_k = f(\bar{o}_k)$$

$$\bar{o}_k = \sum_{k=1}^{m-2} h_k v_k \quad \dots\dots\dots (11)$$

$$(k, j=1, 2, \dots, m-2)$$

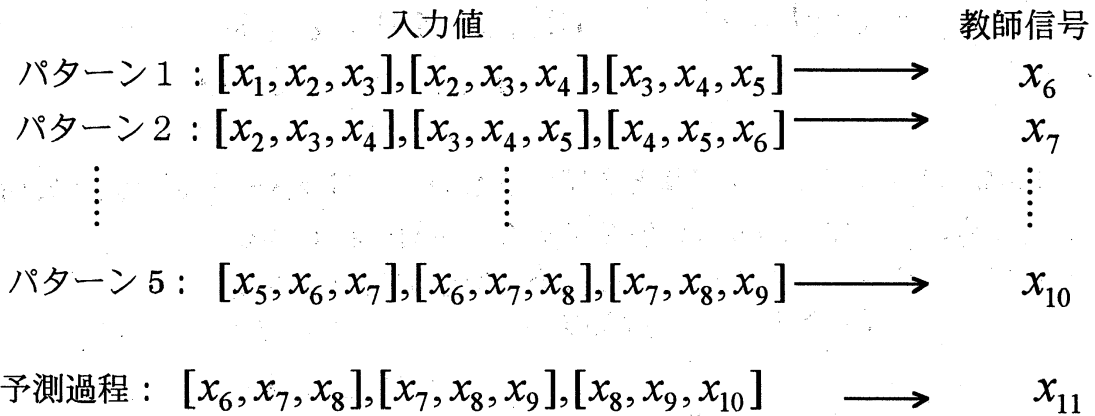
シグモイド関数

$$f(y_1, y_2, y_3) = \frac{1}{1 + e^{-(y_1, y_2, y_3)}} \quad \dots\dots\dots (12)$$

ただし、関数 f はシグモイド関数であり、3次元データなので、式(12)であらわされる。

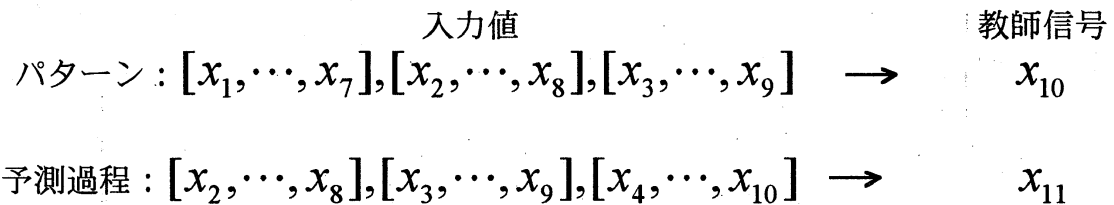
提案1との相違点は、入力はどちらも3次元だが、出力層からの出力がこちらの場合は、1次元になるという点である。そのため提案1は式(6)でシグモイド関数を通したあとの h が3次元になるのに対して、提案2ではシグモイド関数を通すまえの $\bar{h}$ が3次元になる。

この計算式を使い、最初のニューラルネットワークに次のパターンを学習させる。



ここで、提案 1 と同様にパターン 5 まで全ての学習を終了した時点での結合荷重を保存しておく。そして予測過程として入力値を $[x_6, x_7, x_8], [x_7, x_8, x_9], [x_8, x_9, x_{10}]$ 、結合荷重を今保存したものとし、同じニューラルネットワークで 1 度だけ計算させる。その結果を予測値 x_{11} とする。

次に、埋め込み次元を 4, 5, $\dots$ と上げていき、7 次元に埋め込みをする。すると、元の時系列データは、 $[x_{i-6}, x_{i-5}, \dots, x_i]$ ($i = 7, 8, 9, 10$) と書き直すことができる。そして、3 次元の場合と同様に、次のパターンを学習させ、終了した時点での結合荷重を保存しておく。



次に、予測過程として入力値を $[x_2, \dots, x_8], [x_3, \dots, x_9], [x_4, \dots, x_{10}]$ 、結合荷重を先程保存したものとし、1 度だけ計算する。その結果を予測値 x_{11} とする。

以上が提案 2 の予測法であるが、この方法だと、 x_{11} を予測する時、初期値以外の全てのデータの影響が繁殖され、従来の方法での問題が解決できることになる。

5. 数値実験とその結果

以上のように、従来の方法では x_{11} を予測する時点で x_7 以前のデータの影響が無視されるという問題点があったが、それを解決すべく、2 つの提案をした。

次に、この 2 つの方法の有効性を確認するために代表的なカオス的振る舞いをする時系列モデルの短期予測に適用し、その結果を示すことにする。

まず、時系列モデルを式 (13) とする。

$$x_{n+1} = ax_n(1 - x_n) \quad (0 \leq a \leq 4, 0 \leq x_0 \leq 1) \quad \dots\dots (13)$$

集積点 $\bar{a} = 3.5699\dots$

これは、ロジスティック写像と呼ばれるもので、パラメータ a の値が集積点 $\bar{a}$ をこえると、カオス的振る舞いをする事が知られている。

そこで、パラメータを次のように決めて数値実験をしてみた。

時系列データの初期値 $x_0 = 0.1$

$a = 4.0$

入力層、中間層のユニット数 $m = 10$

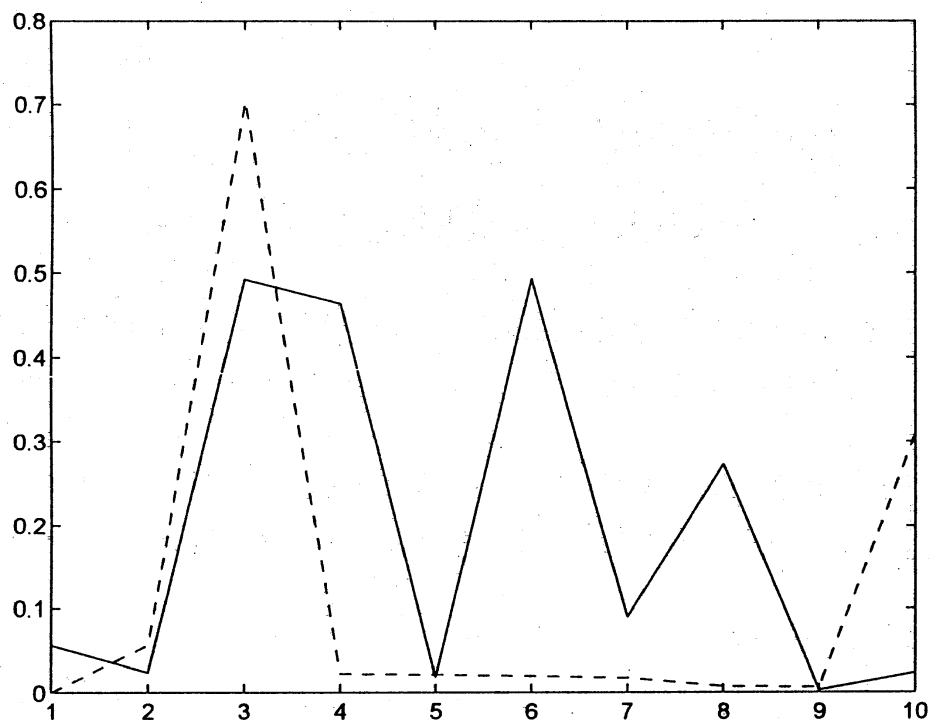
出力層のユニット数 1

埋め込み次元 3

学習パターン数 10

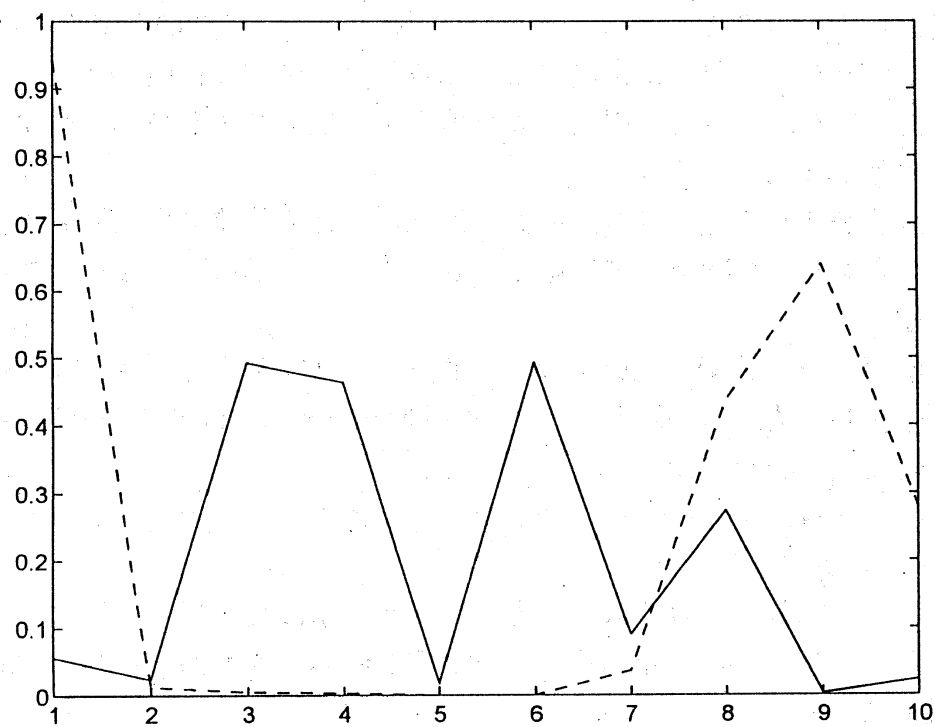
以下に示すグラフは、横軸に時間ステップ、縦軸に (実測値 - 予測値)² をプロットしたものである。また、提案1と従来の方法とを比較したものを図2に、提案2と従来の方法とを比較したものを図3に、提案1と提案2を比較したものを図4に示す。

提案1と従来の方法との比較



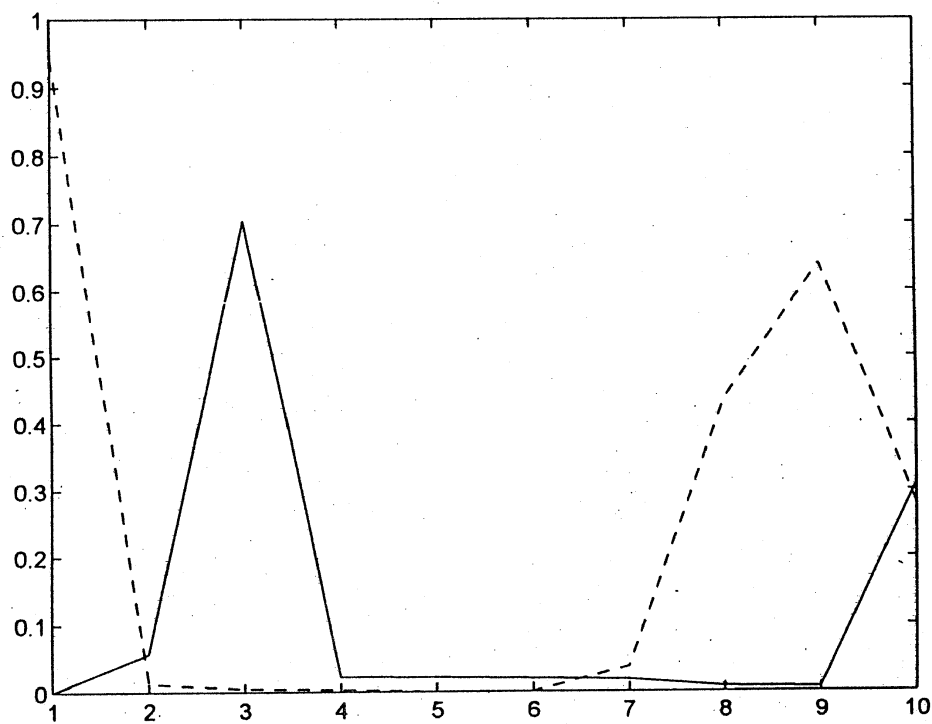
(図2) 実線：従来の方法 点線：提案1

提案2と従来の方法との比較



(図3) 実線：従来の方法 点線：提案2

提案1と提案2との比較



(図4) 実線：提案1 点線：提案2

6. 最後に

今回提案した2つの方法で、カオス的時系列データの予測が可能であるということがわかった、また、従来の方法と比較して、予測の精度が良くなるという結果も得られた。さらに、提案1より提案2のほうがより長く予測が可能であることがわかった。

以上の方法を株価や最高気温、最低気温などの社会現象や自然現象でカオス的振る舞いをする時系列データに応用させることができるのではないだろうか。

<参考文献>

1. 五百旗頭 正, 菅家 正康, 藤本 泰成, 鈴木 新伍: “カオス的時系列の短期予測のための局所ファジィ再構成法”, 日本ファジィ学会誌 vol1.7, No1, pp.186-194
2. 合原 一幸編著: “ニューロ・ファジィ・カオス” 新世代アナログコンピューティング入門
3. 合原 一幸編著: “カオス “ カオス理論の基礎と応用
4. D.E.Rumelhart, G.E.Hinton and R.J.Williams : Nature, 323, pp.533-536(1986)