

双線形時間遅れニューラルネットワークによる 視覚情報処理について

○永嶋史朗 (富士通研究所)

An Example of Vision Process using Bi-linear and Time Delay Neural Network

*Fumio NAGASHIMA (FUJITSU LABORATORIES LTD.)

Abstract— There is a lot of variety of tasks for human friendly robot. A human friendly robot software system is a technologies combination of huge range. I proposed a notation for such software. In this paper, I discuss an application of this notation for vision process.

Key Words: Recurrent Neural Network, Vision Process, Human Friendly Robot

1. はじめに

人間共存型ロボットのシステム記述手法として、双線形時間遅れニューラルネットワークを提案した [1]. このニューラルネットワークは、人間共存型ロボットに必要な様々の処理を記述するために用いられることを前提としている。本稿は、このニューラルネットワークの特徴の1つである双線形性を利用して、視覚情報処理にどのように応用できるかを示し、それを数字認識に応用した結果を示す。なお、この応用例は、一般的なパターンマッチングに関しても同様の議論ができる。

2. 双線形時間遅れニューラルネットワーク

文献 [1] において、双線形時間遅れニューラルネットワークを用いると、ロボットに必要な様々のタスクができることを示した。このニューラルネットワークは、非線形性をニューロンではなく結線を持つことを特徴とする。そのため、線形システムを用いた制御等から非線形効果を用いたフィードバック等の様々のシステムを統一的に記述することができる。基本的なモデルは、

$$\varepsilon_i \frac{dy_i}{dt} + y_i = \sum_j C_{ij} y_j + \sum_{j,k} C_{ijk} y_j y_k + \sum_{j,k} \bar{C}_{ijk} f(y_j) y_k \quad (i = 1, 2, \dots, n) \quad (1)$$

と表すことができる。ただし、 $f(x)$ は

$$f(x) = \begin{cases} 0 & (x < 0) \\ 1 & (0 \leq x) \end{cases},$$

n はニューロン数、 ε_i は遅れパラメータ、 y_i, y_j, y_k はニューロンの状態量、 $C_{ij}, C_{ijk}, \bar{C}_{ijk}$ は重み係数、 t は時間である。この Eq.(1) において、左辺は遅れを表し、右辺第1項は線形項、第2項が双線形項、第3項はデジタルスイッチ項となる。

このニューラルネットワークモデルを言語として実装した [2]. Fig.1 にシステム構成を示す。このシステムは、主に運動生成・制御を目的として設計された。そ

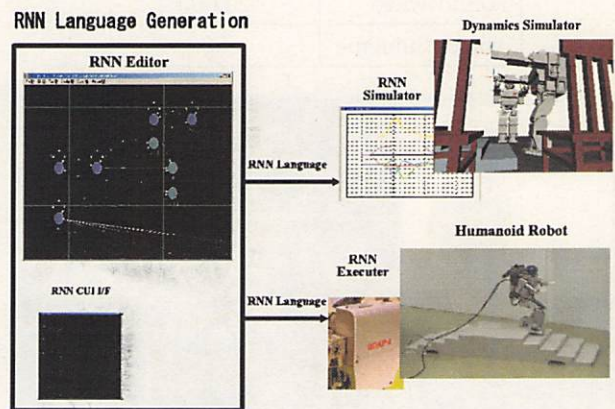


Fig.1 NueROMA system

のため、このシステムには、様々のタスクを行う際に、実行効率の面から問題があった。まず、この問題点について議論し、改善したシステムについて述べる。

3. デジタル信号処理用整数 rnn システム

様々の処理を同一のシステムで扱うためには、各処理の帯域やサイクルタイムに依存しないシステムとする必要がある。各処理の帯域とサイクルタイムを Table 1, 2 に示す。これらの数値は、文献 [3] 等からの推定である。従来、この帯域とサイクルタイムが問題となり、各々の処理は独立に開発され、上位階層で統合化されてきた。しかしながら、すべての情報は、同一の表記方法で表現されない限り、統合化システムとはいえない。個々の処理が統合できない理由のもっとも厄介なものに、積分にある。各処理のサイクルタイムの違いによって、最適な積分幅が異なることによって、打ち切り誤差と丸め誤差がバランスしなくなるか、サイクルタイムの短い処理にすべてが巻き込まれ、計算量の増大化する。これらの問題点は従来あまり議論されなかった。

このように RNN 言語における最大の問題点は、様々の処理のサイクルタイムがまちまちであることである。

Table 1 Band width

sensory organ	band width (bits/sec)
vision	10,000,000
hearing	100,000
touch	1,000,000
taste	1,000
smell	100,000

Table 2 Cycle time

process	cycle time (Hz)
walking	1
running	4
taste,smell	1
vision	30 - 60
touch	10
sound wave	20,000
voice phoneme	50

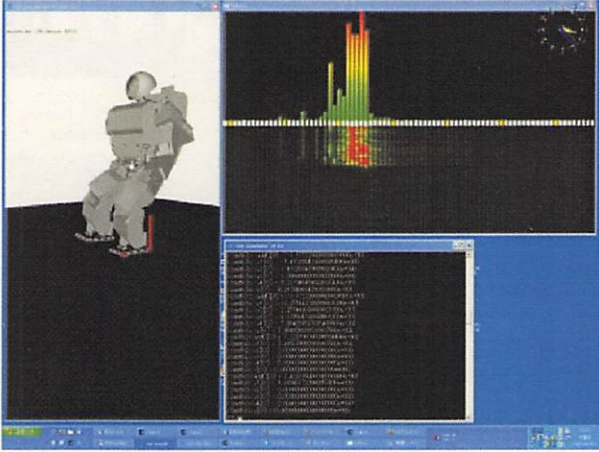


Fig.2 Screen image of irnn

そのため、積分の時間刻みが一定であると、ある処理については、丸め誤差が大きくなり、ある処理に対しては、打ち切り誤差が大きくなり、実用的な計算ができない。本稿では、すべての数値は、固定小数点で行われていると仮定する。積分をデジタル計算機上で行う時には、離散化を行う必要がある。この離散化は、たとえば、以下のような積分公式が用いられる。

$$\frac{dy}{dt} = f(t, x) \quad (2)$$

において、Adams-Bashforth 法は、

$$y^{(i+1)} = y^{(i)} + \delta t \sum_{j=0}^k \beta_j f(t^{(i-j)}, y^{(i-j)}) \quad (3)$$

である。このような公式と、本稿の RNN 言語モデルを比較する。本稿の RNN 言語モデルは、

$$\frac{dy_i}{dt} = \frac{1}{\varepsilon_i} g_i(y_j) \quad (4)$$

と表現できる。ただし、 $g_i(y_j) = (\sum_j^n C_{ij} y_j + \sum_{j,k}^{n,n} C_{ijk} y_j y_k + \sum_{j,k}^{n,n} \bar{C}_{ijk} f(y_j) y_k) - y_i$ である。 $g_i(y_j)$ は、 ε_i の関数ではないため、下記のように、積分ステップ δt と遅れ ε_i の2つのパラメータは、 $\delta t/\varepsilon$ の形でのみあらわれる。下式のように、 δt と ε_i というパラメータの代わりに、その比の $\Delta t (\equiv \delta t/\varepsilon)$ をパラメータと考えることができる。

$$y^{(i+1)} = y^{(i)} + \Delta t \sum_{j=0}^k \beta_j g(j)(y^{(i-j)}) \quad (5)$$

ここで、適切な Δt について考察する。Eq.(5) 右辺第2項の中の Δt は、丸め誤差と打ち切り誤差のバランスによって決定される。両誤差が同程度である計算が理想のデジタル計算であると考えられる。打ち切り誤差は、積分法の次数と刻み幅に依存する。丸め誤差は、刻み幅と計算の有効桁数に依存する。

今、すべての計算が有効桁数 m の固定小数点の計算で行われ、さらに、 $y^{(i)}$ と $g(j)(y^{(i-j)})$ がともに同じオーダであったとする。事実、 $g(j)(y^{(i-j)})$ は、自他のニューロン値から導かれるので、同オーダである仮定は妥当であると考えられる。今、デジタル計算を意識して、 Δt を $1/2^n$ とし、積分の打ち切り誤差が $O(\delta t^k)$ であるとする。このとき、丸め誤差、打ち切り誤差は、簡単な計算により

$$\text{丸め誤差} = O\left(\frac{1}{2^{m-n}}\right) \quad (6)$$

$$\text{打ち切り誤差} = O\left(\frac{1}{2^{kn}}\right) \quad (7)$$

と評価できる。このオーダが等しいとすると、

$$n \approx \frac{m}{1+k} \quad (8)$$

という関係が導き出される。この条件を満足すると、両誤差がほぼ等しくなり、デジタル誤差のもっとも小さな状態となる。たとえば、符号ビット付4バイト整数 ($m = 31$) で、2次 Adams-Bashforth 法 ($k = 2$) を用いるとすると、 $n \approx 10$ 、すなわち、 $g(j)(y^{(i-j)})$ の値を10ビット右シフトし、加算すればよいことになる。この計算は、すべてのニューロンについて同じであるが、すべてのニューロンを同期させるためには、実際の積分幅時間幅を考慮する必要がある。1回の積分で、ニューロンの時間は、 $\delta t = \varepsilon/2^n$ だけ進む。よって、ある時間だけ時間を進めるには、 $t/\delta t = (2^n/\varepsilon)t$ 回だけ Eq.(5) の演算を繰り返せばよい。ニューラルネットワーク上のすべてのニューロンの $\varepsilon_i (> 0)$ の最大公約数を ε_{gcd} とし、この ε_{gcd} から求まる $\delta t_{gcd} (\equiv \varepsilon_{gcd}/2^n)$ を基本時間積分幅とすると、すべてのニューロンの δt は、 δt_{gcd} の整数倍 c となる。よって、各ニューロンの計算は、計算ループの c 回毎に計算すればよいことになる。

4. 単純なパターンマッチング問題

本稿のニューラルネットワークを用いた画像処理について説明するために、まず、非常に簡単な例として、Fig. 4 のような単純な4つの画素から成る画像を考える。各画素は、1または0の値をとるとすると、そのパターンは、全部で16となる。

非常に単純な場合、最もよく用いられるパターンマッチングの方法は、「相関演算」である。この相関演算は、入力とテンプレートの内積として計算される。そのためニューラルネットワークとしては、Fig. 5の最上段のようなものになる。各画素の値が0,1のような場合には、この相関演算のみで充分区別できる。しかしながら、各画素が中間の値を取れるようになると、話が違って来る。各画素が中間の値を取れると、まったく区別のつかないパターンが存在する。

そこで、多項式によって、区別することを考える。画素数が4なので、可能な多項式は、下記のような4次多項式である。

$$y_i = \sum_i^4 C_i x_i + \sum_{i,j}^4 C_{ij} x_i x_j \dots \quad (9)$$

摂動的な考え方から、内積計算で十分でなかった場合、順次次数をあげて精度を上げることを考える。このときに2次、3次のネットワークはFig.5の下図のようになる。

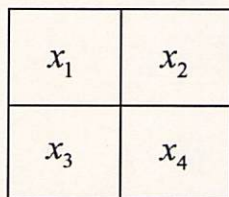


Fig.3 Simple image

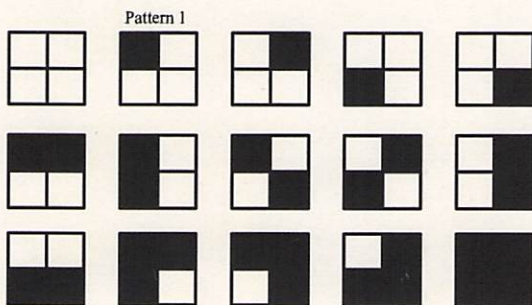


Fig.4 16 patterns

5. 手書き数字認識実験

本稿のニューラルネットワークを用いた応用問題として、手書き数字の認識実験を行った。Fig.6に示すような手書き数字のデータ1000個に対して、0,1,...,9の区別を次数3次まで考慮して、勾配法で学習をさせた。実験の結果をTable 3に示す。最終的に、誤認率は0%となった。学習にかかった時間は、8分28秒であった。

6. 最後に

提案したニューラルネットワークシステムを用いた画像情報処理について議論した。運動生成や制御においては、重要だった遅れではなく、双一次結合が重要な役割を果たす。手書き数字を用いて実験を行った。その結果、ほぼ100%の確率で認識できることを確認した。

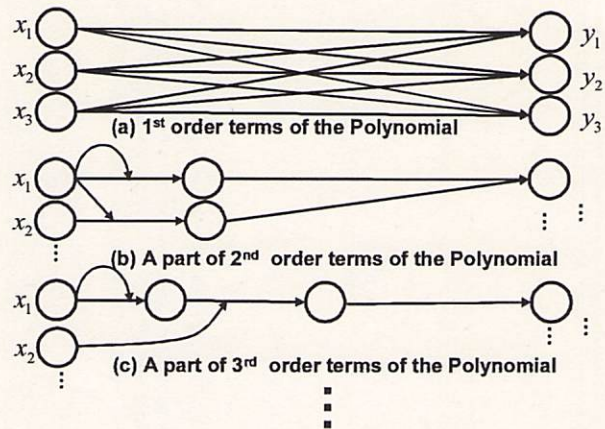


Fig.5 Perturbed neural network

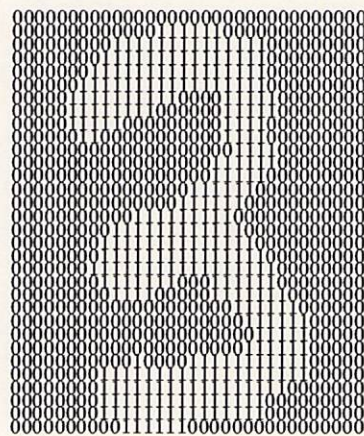


Fig.6 Example of hand written number data

Table 3 Experiment result

trial	wrong classification (%)
1000	18.309859
2000	6.539235
3000	3.018109
4000	2.716298
5000	1.509054
6000	1.408451
7000	0.402414
8000	0.201207
9000	0.201207
10000	0.000000

- [1] 永嶋史朗: “双線形時間遅れニューラルネットワークによるロボットソフトウェアシステム”, 日本ロボット学会誌, 24, 6, 2006(予定).
- [2] 永嶋史朗: “ヒューマノイドロボット動作生成システム - NueROMA -”, 日本ロボット学会誌, 22, 2, pp.34-37, 2003.
- [3] ジェフ・ホーキンス他: “考える脳 考えるコンピュータ”, ランダムハウス講談社, 2005.