ホップフィールドネットワークによるニューラルネットPTV

近江 和生[○] (大阪産業大学)

Neural network PTV using a Hopfield network

Kazuo OHMI

ABSTRACT

The Hopfield neural network is applied to the tracking algorithm of individual particles in the PIV measurement of fluid flows. In order that the particle tracking should work well with relatively large numbers of individual particles, the object function of the network is assigned with some elaborately defined constraint conditions. The new network is applied to synthetic particle images (the VSJ's PIV standard images) as well as experimental images from the water channel experiment.

Keywords: PIV, Particle tracking velocimetry, Neural network, Hopfield network,

1. 緒 論

ニューラルネットの手法を結合方式により分類すると、階層型と相互結合型の2種類に分けられるが、2画面間で比較的多数の粒子を追跡する PTV アルゴリズムにおいては、相互結合型のニューラルネットがより効果的であると考えられる。粒子対応関係の評価において、周囲の粒子の配置や動きを考慮に入れることが比較的容易であるという点もメリットであるが、とくに重要な特徴であるのは、第1画面の個々の粒子に対して、第2画面の対応候補を距離の閾値等によって事前に選別せずとも（すなわち流れの速度場に関する大略情報がなくとも）、比較的短時間で粒子追跡を行なうことができる点である。

本研究では、このような相互結合型ネットワークの利点を生かして、ホップフィールド型ニューラルネットによる PTV のアルゴリズムを考案し、それを代表的な PIV 用の粒子画像 (PIV 標準画像) に適用して、粒子追跡の結果について検討を加える。

2. ホップフィールド型ニューラルネット

ホップフィールド型ネットワーク¹⁾の特徴は、すべての(ニューロン)ユニットが相互に結合しあったネットワークを考え、その状態量であるエネルギー関数が最小値となるような状態変化を繰り返すことで、いわゆる最適値問題の解を得ることができる点である。このネットワークを用いて PTV における粒子対応問題を解く場合、その問題自体をいかなる数のユニットにより、いかなる状態量の変化として表すかが大きな問題であるが、ここでは巡回セールスマン問題への応用²⁾との類似性も考慮して、Fig.1 に示すように、第1画面と第2画面間のすべての粒子の対応状態を、ユニットの状態量として表すことにする。

Fig.1 Neuron units arrangement for a PTV particle pairing problem.

ここで、ニューロンの状態量は基本的に1か0、つまり発火しているか否かであり、2画面間の粒子の正しい対応関係は発火(●)、そうでないものは非発火(●)で表している。なお、原則として両画面間での粒子の喪失や、粒子1個が他画面の粒子2個以上に対応する状況は考えないので、第1画面の各ニューロンは必ず1つ、しかも1つだけ発火し、また第2画面の各ニューロンについても同様である。

さて、ホップフィールド型ネットワークにより最適値問題の解を得るためには、問題の何らかの拘束条件を表すような目的関数をネットワークの状態量の関数として与え、そのエネルギー量が最小となるように状態量を変化させていく必要がある。このとき、エネルギー量最小の条件を与えるエネルギー関数 E は、

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{k=1}^M \sum_{j=1}^N \sum_{l=1}^M T_{ik,jl} \cdot V_{ik} \cdot V_{jl} - \sum_{i=1}^N \sum_{k=1}^M I_{ik} \cdot V_{ik} \quad (1)$$

で与えられる¹⁾。ここで V_{ik} は、Fig.1 中で i 行 k 列目のユ

ニットの状態量を表し(V_{jl} も同様)、 $T_{ik,jl}$ はこれら両ユニットの結合の重みを、 I_{ik} は i 行 k 列目のユニットの閾値を、また N, M はFig.1中での行および列の総数(つまり両画面での粒子数)を表す。これに対して、各ユニットの状態量の変化は、単純に閾値と大小比較して1または0とする方法でなく、通常、以下のように正規化した更新式で計算される。

$$V_{ik} = \frac{1}{2} \cdot \left(1 + \tanh \left(\frac{u_{ik}}{0.5} \right) \right) \quad (2)$$

$$\text{ただし } u_{ik} = \sum_{j,l} T_{ik,jl} \cdot V_{jl} + I_{ik}$$

相互結合型ネットワークの状態変化とは、実質的には(2)式のようなユニットの状態量の変化であり、この変化を繰り返すことでエネルギー量が最小化へと進み、その時点でユニット値の分布が問題の最適解を示すようになる。

ここで重要であるのは、(1)のエネルギー式に対応して最適解を得るのに有効と考えられる拘束条件を検討し、それを目的関数という形で与えることである。PTVでは粒子の正しい対応関係を得るために、基本的な拘束条件として

- (a) 対応粒子間の距離の総和が最小
- (b) 第1画面の1個の粒子に対応する第2画面の粒子は1個のみであり、逆もまた同様である

の2種類を考えるのが有効であると考えられる。条件(a)は、粒子対応問題での最適解の基準条件として一般的によく用いられ、画面内の粒子密度が十分に小さければ、それなりに有効な条件である。条件(b)は Fig.1 のようなユニット形式から必然的に必要となる条件であり、実際の実験粒子画像の状況とは相容れないものであるが、後述のように今回のネットワークでは、この拘束条件を特に絶対条件として扱う訳ではなく、他の拘束条件との相対的な重み付けによって、この拘束条件の重要度(貢献度)を調整することができる。

すでに他の粒子追跡手法^{3),4)}でも指摘されているように、上記の拘束条件(a)は、画面内の粒子密度が増大すると、誤対応の結果を多数生じる結果となりやすい。これに対する有効な対処方法として、2画面間での粒子移動速度の局所的類似性とか、粒子分布パターンの類似性といった条件が追加される。そこで今回のネットワークにおいても、これらの類似性を与える拘束条件として、

- (c) 粒子の移動距離や移動方向は、画面内でスムーズに変化するので、個々の粒子移動ベクトルと周囲の粒子移動ベクトルとの差ベクトルの総和が最小
- (d) 同一粒子のまわりの粒子分布パターンは2画面間で類似するので、個々の粒子とその周囲粒子との距離の偏差(2画面間偏差)の総和が最小

の2条件を追加し⁵⁾、それらをエネルギー式に対応すべき目的関数として定義した。この目的関数を数式化すると、

$$\phi_0 = C_0 \cdot \sum_{i=1}^N \sum_{k=1}^M f_R(i,k) \cdot V_{ik} \quad (3)$$

$$\phi_1 = C_1 \cdot \sum_{i=1}^N \sum_{k=1}^M \sum_{j=1}^N \sum_{l=1}^M |f_X(i,k) - f_X(j,l)| \cdot V_{ik} \cdot V_{jl} \quad (4)$$

$$+ C_2 \cdot \sum_{i=1}^N \sum_{k=1}^M \sum_{j=1}^N \sum_{l=1}^M |f_Y(i,k) - f_Y(j,l)| \cdot V_{ik} \cdot V_{jl}$$

$$\phi_2 = C_3 \cdot \sum_{i=1}^N \sum_{k=1}^M \sum_{j=1}^N \sum_{l=1}^M |g(i,j) - g(k,l)| \cdot V_{ik} \cdot V_{jl} \quad (5)$$

$$\phi_3 = C_4 \cdot \sum_{i=1}^N \sum_{k=1}^M \left(\sum_{\substack{j=1 \\ j \neq i}}^N V_{ik} \cdot V_{jk} + \sum_{\substack{l=1 \\ l \neq k}}^M V_{ik} \cdot V_{il} \right) \quad (6)$$

のようになる。ここで $f_R(i,k)$ は異なる2画面の粒子 i と k との距離、 $f_X(i,k)$ は同じく粒子 i と k との x 方向変位、 $f_Y(i,k)$ は粒子 i と k の y 方向変位、そして $g(i,j)$ は同一画面内の粒子 i と j との距離をそれぞれ表している。また係数 C_0, C_1, C_2, C_3, C_4 は、ネットワークを効率的に収束させるためのもので、実際には各拘束条件に対する重み係数である。すなわち、最終の目的関数

$$\phi = \phi_0 + \phi_1 + \phi_2 + \phi_3 \quad (7)$$

により与えるものとする。

この目的関数の2次の項を集めて(1)のエネルギー式と係数比較を行うことにより、結合の重み $T_{ik,jl}$ の大きさが定まり、1次の項を集めて比較すると閾値 I_{ik} の値が定まる。

$$\begin{aligned} T_{ik,jl} = & -\frac{C_1}{2} \cdot (1 - \delta_{ij}) \cdot (1 - \delta_{kl}) \cdot |f_X(i,k) - f_X(j,l)| \\ & -\frac{C_2}{2} \cdot (1 - \delta_{ij}) \cdot (1 - \delta_{kl}) \cdot |f_Y(i,k) - f_Y(j,l)| \\ & -\frac{C_3}{2} \cdot (1 - \delta_{ij}) \cdot (1 - \delta_{kl}) \cdot |g(i,j) - g(k,l)| \\ & -\frac{C_4}{2} \cdot \delta_{ij} \cdot (1 - \delta_{kl}) - C_4 \cdot \delta_{kl} \cdot (1 - \delta_{ij}) \end{aligned} \quad (8)$$

$$I_{ik} = -f_R(i,k) \cdot C_0 \quad (9)$$

上式において δ_{ij}, δ_{kl} はクロネッカーのデルタであり、 δ_{ij} は $i=k$ のとき1、それ以外の場合は0の値をとる。上の2種の計算結果を(2)式に代入することにより u_{ik} が求まり、そこからユニット値 V_{ik} の更新値が求められる。この一連の処理を繰り返し計算することにより、目的関数 ϕ の値は次第に最小値へと近づき、最終段階で各ユニット値が示す発火あるいは非発火の状態によって、粒子の対応関係が明らかになる。ただし(2)式が示すように、ここでのユニット値は単純な2値表現でないため、Fig.1のユニット配置で各行または各列において V_{ik} の値が最大となる行列位置が、求める粒子の対応関係である。

なお、重み係数 C_0, C_1, C_2, C_3, C_4 の値は、試行錯誤的に経験値を決定して与えた。また、 V_{ik} や V_{jl} の初期値は0.0~1.0の範囲の乱数で与えた。一方、 $\phi_0, \phi_1, \phi_2, \phi_3$ の値を評価する際、各粒子間の距離の値は、生の計測値でなく正規化した値を用いて、各目的関数の値の偏りをできる限り少なくした。

3. 実験結果

3.1 追跡粒子数の影響

可視化情報学会の PIV標準画像(#301)の粒子データを用いて、粒子追跡の実験を行った。最初の実験では、異なる6通りの追跡粒子数に対して、Table 1 のような重み係数および繰り返し数の設定により計算を行った。その実験結果における誤対応数と、要した計算時間（Pentium III 866MHz による）も合わせて Table 1 に記した。なお、この表での実験結果を粒子の移動ベクトルの形で表示したものが、Fig. 2 である。

Table 1 Experimental parameters and results

particles	C_0	C_1	C_2	C_3	C_4	iterations	errors	time
40	100	1	1	100	3.25	4	0	0'
50	100	1	1	100	3.25	10	0	4'
70	1000	1	1	1000	3	5	0	7'
100	1000	1	1	1000	3.25	7	0	41'
130	2000	1	1	2000	6	8	0	2''48'
150	2000	1	1	3000	6	12	0	6''00'

Fig. 2 PTV results with different numbers of particles.

3.2 重み係数の影響

次の実験では、追跡粒子数を 100 に固定し、各々の重み係数の値を変化させることによって、実行結果にどのような影響が現れるかを調べた。具体的には C_0 , C_1 , C_2 , C_3 , C_4 の値をそれぞれ 2 通りに変化させ、粒子の誤対応数や計算時間を比較した。重み係数および繰り返し数の設定値と、実験結果における誤対応数と計算時間を要約したものが Table 2 である。使用した PC 等の条件は、最初の実験と同一である。この表での実験結果の幾つかを粒子の移動ベクトルの形で表示したものが、Fig. 3 である。

Table 2 Experimental parameters and results

particles	C_0	C_1	C_2	C_3	C_4	iteration	errors	time
100	100	1	1	1000	3.25	30	2	2''58'
100	10000	1	1	1000	3.25	30	9	2''58'
100	1000	1000	1	1000	3.25	30	2	2''58'
100	1000	1	1000	1000	3.25	30	1	2''58'
100	1000	1	1	100	3.25	30	5	2''58'
100	1000	1	1	10000	3.25	30	4	2''58'
100	1000	1	1	1000	0.1	30	5	2''58'
100	1000	1	1	1000	10	30	2	2''58'

(a) $C_0 = 100$

(b) $C_0 = 10000$

(c) $C_4 = 0.1$

(d) $C_4 = 10$

Fig.3 PTV results with different weight coefficients.

4. 結 論

ホップフィールド型ネットワークによるPTVの特徴は、遺伝アルゴリズムを用いたPTV⁴⁾と同じく、粒子移動距離に対する閾値設定等により、対応候補粒子を事前に選別する必要がない（流れ場に関する予備知識を必要としない）点である。しかしこのようなメリットの代償として、これも遺伝アルゴリズムと同様、計算処理時間は必然的に長くなる。遺伝アルゴリズムと異なる特徴としては、目的関数の個々の拘束条件に適切な重みを付けて、計算過程でそれらのバランスを調整しやすいこと、そしてそのことによって、画面の一方で粒子像が消われる現象に対して、より柔軟に対応できる可能性があること、以上の2点である。反対にホップフィールド型ネットワークによるPTVの現状での問題点は、追跡粒子数の多い（200個以上）画像では正しい粒子対応関係を得にくく、よって画面分割等の手法に頼らざるを得ないことである。この点に関しては、さらにアルゴリズムの工夫と改良が必要であると思われる。

参 考 文 献

- 1) Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities, Proc. Nat'l Acad. Sci. USA, Vol.97 (1982), pp.2554-2558.
- 2) 中野馨, 飯沼一元: ニューロコンピュータ, 技術評論社(1989), pp.259-263.
- 3) Ohmi, K., Li, H.-Y.: Particle tracking velocimetry with new algorithms, Measurement Science and Technology, Vol.11, Issue 6 (2000), pp.603-616.
- 4) Ohmi, K., Yoshida, N., Huynh Thai Hoang: Genetic algorithm PIV, Proc. 4th International Symposium on Particle Image Velocimetry (2001), P1051.
- 5) Knaak, M., Rothlubbers, C., Orglmeister, R: A Hopfield neural network for flow field computation based on Particle Image Velocimetry / Particle Tracking Velocimetry image sequences, IEEE International Conference on Neural Networks, ICNN 97 (1997)