特集論文 SOM とその応用

自己組織化マップの 自動追加学習システム†

道端 正裕*・三好 力*・増山 博*

従来の Kohonen の自己組織化マップ(SOM)では入力データの特徴が変化していく場合にはその変化に対応できず、適用が難しいと考えられる。

本稿では入力データの特徴が変化した場合、それを検出し、その変化に対応して自動的に特徴の追加学習を行い、マップを再構成させる追加学習手法と、元のマップにつけられていたラベルから再構成後のマップに対応したラベルの位置を自動的に検出し、ラベル付けを行う半自動ラベル付け手法の2つをそなえた SOM の自動追加学習システムを提案する。実験によりデータの特徴変化検出が可能であり、追加学習が有効である事を示した。また、追加学習後のラベルに関する検証を行い、半自動ラベル付け手法によってマップ再編成後のラベルに関する問題を解決した。

keyword: ニューラルネットワーク、自己組織化マップ、教師なし学習、評価

1. はじめに

人工知能研究において、複数の属性からなる入力データの分類をするため、入力データの特徴を自動的に抽出する方法が研究されている。このような手法のなかにニューラルネットワークを用いたものがある。ニューラルネットワークを用いた手法は学習手法によって教師付き学習(バックプロパゲーション法等)と教師なし学習(自己組織化マップ(SOM)等)の二つに分けられる[1]~[5]。ニューラルネットワークは汎化能力を持つため、学習を行ったデータに関して、ある程度の誤差については適応できる。しかし、学習を行っていないデータ、つまり、学習データから汎化能力以上の差があるデータについては、学習を行っていないために適応できない。すなわち、上記の2種類の手法とも、学習終了後、適用段階において入力データの特徴が学習データのそれから変化していくと、多くの場合適用を続けていくのは難しい。例えば文献[1]においては、「多くのニューラルネットワークのパラダイムは、安定-可塑性のジレンマとして知られる問題に悩まされている。これはニューラルネットワークが古い知識を維持しながらその上に新しい情報を学習する困難さを意味する。」などと報告されている。しかし、教師なし

学習については、教師あり学習とは異なり、学習に際して教師信号を必要としないために、入力データがそのまま学習データとして使用できる。教師なし学習手法で、入力の変化に適応的に対応していく手法として、適応共鳴理論(ART)[1]が挙げられるが、ARTは全く未知の入力がされると逐次学習していく、つまり一度のパターン提示で学習が行われる。しかしこの場合、入力の雑音に非常に弱いという欠点が挙げられる[1]。すなわち、雑音を含む入力を直ちに反映するため、雑音の性質によっては一時的にせよ本来示すべき汎化能力が阻害される可能性がある。

しかし、他の教師なし学習を行うニューラルネットワークの代表である SOM においては、耐雑音性も強く、アルゴリズムも単純であるため、適用段階においてデータの特徴が変化していったとき、その変化を検出できれば、自動的に追加学習を行うことによって、その変化に対応していけると考えた。

そこで、SOM において、入力データのもつ特徴が動的に変化していく場合においても適用できるようにするため、入力データの特徴が変化した場合、それを検出し、自動的に特徴の追加学習を行い、期待通りのクラスタリングを行うマップを再構成する様な追加学習システムを考案した。

このシステムでは、追加学習を起動するタイミングを調整できるので、未知の入力が来れば直ちに追加学習を始める ART と同様な設定とすることもでき、また、ある程度時間の余裕をもって統計的に有意になることを見極めてから追加学習を始めることもできる。つまり、汎化能力と追加学習のバランスをとることができ

† Automatic Incremental Learning System on Self-Organizing Map
Masahiro MICHIMATA, Tsutomu MIYOSHI, Hiroshi MASUYAMA

* 鳥取大学大学院工学研究科知能情報工学専攻
Course in Information and Knowledge Engineering, The Graduate school of Engineering, Tottori University

るシステムとなる。

また、SOM で追加学習を行うと、追加学習後はマップが学習によって変化し、各ノードに付加されているラベルは新しいマップに対応しなくなるため、追加学習前のマップにつけられたラベルから新しいマップ上でのラベルの位置を自動的に検出するシステムを考案した。これにより、データの変化していないクラスにつけられていたラベルについては、新しいマップに自動的にラベル付けされる。また、変化したクラスに対しては、変化したことと、マップ上での領域がわかり、新しいラベルを付けることが必要であることが提示され、ラベル付け作業の半自動化が実現される。

これにより、文献[1]に記されるような、「既に学習されたネットワークに新しい学習ベクトルを付け加えることによって、重みが変わり、極端な場合、以前の学習のすべてを壊してしまう破滅的な副作用が生じないとは保証できない。新しい情報を取り入れるために再学習が必要になれば、学習に大変量の計算時間(数時間、数日)を要する大規模なネットワークの場合、深刻な制約となる。」といった、入力情報が変化したときに対応しようとして起こりうる問題に対しても、計算時間に対しては、追加学習を行うことにより再学習に要する時間よりも大幅に短い時間で学習を終了でき、また、半自動ラベル付けにより、変化した重みに対する問題点にも対応できる。

本稿では、上記の追加学習システムと半自動ラベル付けシステムをそなえた、SOM の自動追加学習システムを提案する。実験により、提案システムによってデータの特徴変化検出が可能であり、追加学習が有効である事を示した。そして、追加学習後のラベルに関する検証を行い、半自動ラベル付け手法によってマップ再編成後のラベルに関する問題を解決した。

以下、2 章において SOM の説明および問題点について、3 章で追加学習手法について、4 章で半自動ラベル付け手法について述べ、5 章、6 章でそれぞれの手法の実験と結果の考察を行い、7 章において結論を述べる。

2. 自己組織化マップ(SOM)

自己組織化マップ (SOM) は、ニューラルネットワークをモデル化したもののひとつで、教師無し競合強化学習及び近傍学習により、ある分布に従う多次元のデータに対してその分布を近似した特徴マップを生成する。特徴マップは 2 次元平面に表示され、同じ様な特徴をもつ入力データはマップ上の近い位置に出力される。入力データに対してマップのどの位置が出力されたかにより、どのデータと類似した特徴をもつかと

図2.1 SOM のモデル図

いうことがわかり、視覚的にも理解しやすい。

SOM のネットワークは、図2.1のようにデータの次元に対応した入力ノードを持つ入力層と、2次元平面上に出力ノードが格子状に配置された出力層の2層からなり、各入力ノードはすべての出力ノードと結合している。また、各々の出力ノードには入力データである n 次元のベクトルに対応する n 次元の結合ベクトルが格納されており、入力ベクトルと1番近い結合ベクトルをもつ出力ノードと、その近傍にある出力ノードの結合ベクトルを入力ベクトルに近づける事で学習を行う。このような動作を近傍を縮小しながら繰り返すことにより、次第に、同じような特徴をもつデータ同士が集まった領域がつくられていく。すなわち、マップが自己組織化されていき、入力ベクトルの特徴を反映したマップが生成される。

2.1. SOM の学習アルゴリズム

以下に SOM の学習アルゴリズムを示す。

(1) 初期化

学習回数 T を設定する。(ここで学習回数とは、学習データセットをネットワークに提示する回数) t は学習時における時間変化で、 T は t のとりうる最大値となる。

学習データ(その集合を X とし、その数を S とする)を設定する。

$$X = \{x_i(s) \mid i = 1, 2, \dots, n; s = 1, 2, \dots, S\}$$

マップのサイズ(出力ノード数: M)を設定する。各出力ノードにおける n 次元の入力ベクトルに対する結合ベクトル(その集合を W とする)を乱数によって初期

化する。また、出力ノードは位置ベクトル $\mathbf{w}$ をもつ。 $Rand_i$ は $0.0 \leq Rand_i \leq 1.0$ の一様乱数である。

$$\begin{aligned} W &= (w_{ij}(t, s) | i=1, 2, \dots, n; j=1, 2, \dots, M; \\ &\quad t=0, 1, \dots, T; s=1, 2, \dots, S) \\ w_{ij}(0, 1) &= Rand_i \\ \mathbf{w} &= \{\lambda_j(p, q) | p=1, 2, \dots, P; q=1, 2, \dots, Q\} \\ M &= P \times Q \end{aligned}$$

学習率の初期値と更新率を決定するパラメータ $\alpha(0)$ 、 C_1 を設定する。

近傍領域の初期範囲と更新率を決定するパラメータ $\rho(0)$ 、 C_2 を設定する。

(2) 学習データの提示

ネットワークに学習データを提示する。

(3) 距離の計算

以下の式により、学習データと各出力ノードの距離を計算する。

$$d_j(t, s) = \sum_{i=1}^n (x_i(s) - w_{ij}(t, s))^2 \dots\dots\dots (*)$$

(4) 最小距離の選択

$d_i(t, s)$ が最小となる出力ノードを勝利ノード v とする。

$$d_v(t, s) = \min_j (d_j(t, s))$$

(5) 結合ベクトルの更新、および近傍の縮小

勝利ノードとその近傍領域にある全てのノードの結合ベクトルを以下の式により更新する。

$$\begin{aligned} w_{ij}(t, s+1) &= w_{ij}(t, s) + h(t) (x_i(s) - w_{ij}(t, s)) \\ &\quad (1 \leq s \leq S-1) \\ w_{ij}(t+1, 1) &= w_{ij}(t, S) + h(t) (x_i(S) - w_{ij}(t, S)) \\ h(t) &= \alpha(t) \cdot \exp(-\|\lambda_v - \lambda_j\|^2 / 2\rho^2(t)) \\ \alpha(t) &= \alpha(0) C_1 / (C_1 + t) \\ \rho(t) &= \rho(0) C_2 / (C_2 + t) \end{aligned}$$

ここで、 $h(t)$ は近傍関数とよばれ、 λ_v は勝利ノードの位置ベクトル、 $\alpha(t)$ は学習率、 $\rho(t)$ は近傍の半径に相当する。近傍の半径とは勝利ノードからのハミング距離を指し、位置ベクトルがその値以内の位置にあるノードは結合ベクトルの更新を行う。学習率 $\alpha(t)$ は初めは高く保ち(0.8~0.5程度)無秩序なマップを粗く構成させる。そして学習率 $\alpha(t)$ を時間と共に減少させることによりマップを細かく調整していく。この細かい調整にはマップを粗く構成させる時間の10倍から100倍かかり、最終的には学習率 $\alpha(t)$ を0.1以下にまで減少させる。同様に近傍の半径 $\rho(t)$ も時間と共に減少させ、近傍領域を縮小して結合ベクトルが更新される領域を狭めていく。

(6) 繰り返し

(2)から(5)にかけての行為を T 回、繰り返す。

2.2. ラベル付け

SOMは学習終了後、入力データの特徴をマップの出力位置により把握するため、利用者がマップ(各出力ノードもしくは領域)にラベルを付ける。

その作業としては、ラベルの代表となるデータが勝利するノードをそのラベルの代表点とする。代表点とならなかったノードのラベルは、各ノードと代表ノードとのマッチ度を調べ、一番マッチ度の高い代表ノードのラベルをそのノードのラベルとする。このとき、マッチ度には閾値をもうけるため、どのラベルにも属さないノードも存在する。この閾値は学習データの分散具合や学習時の学習率の減少具合により推定する事が可能である。

2.3. SOMの問題点

世の中一般には予測不能な変化を伴うデータが存在する。例えば、「中学生の平均身長の上昇と正常範囲の変化」や「地球温暖化による異常気象判定基準の変化」などと多数挙げることができる。

SOMは学習フェーズと分類フェーズとに分けられ、学習フェーズでのみ重み w の更新を行い、学習終了後は重み w を固定し(更新しない)入力データの分類を行う(分類フェーズ)。つまり、SOMは学習終了後、分類フェーズにおいて、学習データと同じ特徴を持つ入力データに関しては、適切なクラスタリングを行う。しかし、入力データが、学習データとは異なった特徴をもつように変化すると、そのデータは適切にクラスタリングされとは限らなくなる。すなわち、入力されるデータが時間と共に、初期のころ入力していたデータと特徴が異なっているようなもの、また、時系列に変化することをわかっているものではなく、変化が起こるかどうかわからない場合はその変化が予測しにくい入力データの場合、しまいには分類器としての機能を果たさなくなる危険性がある。このように、入力データのもつ特徴が予測不能な変化を伴う場合においては、SOMを適用するのは難しいと言わざるをえない。SOMがその変化に対応できれば、応用範囲の広いクラスタリング手法となるのではないかと考えた。

そこで、本研究では入力データの特徴が変化した場合、それを検出し、その変化に対応して自動的に特徴の追加学習を行い、変化しなかったデータに対しては自動的にラベル付けを行うことが可能な、SOMの自動追加学習システムを提案する。

3. 異常検出と自動追加学習

入力データの特徴が変化する場合においても、変化に対応し、SOM を適用していけるようにするため、以下のような検討を行った。

3.1. 異常検出

マップを適用するにあたり、好ましくない状態というのは入力したデータと出力されたノードとの距離が期待値よりも遠いことである。つまり、 $d_v \geq Th$ (Th は閾値) の状態である。そこで、 $d_v \geq Th$ になった場合を異常と判定し、逐次に追加学習を施せば常に入力データと出力ノードとの距離が一定以下の、好ましい状態を維持できるのではないかと考えた。しかし、「初期入力データに対して自己組織化したマップを基に、変化する入力をその都度マップに入力してマップを適応的に自己組織化させる」という単純な方法では、ART での検討と同様に、雑音を含む入力を直ちに反映するため、雑音の性質によっては一時的にせよ本来示すべき汎化能力が阻害される可能性がある。また、学習時間が増大する等の支障が出てくる事が容易に考えられる。そこで、入力データ一つ一つで異常を検知するのではなく、入力データ群をまとめて全体的に異常を検出する形式を考えた。この形式だと、逐次学習を施す形式と比べて大幅に学習回数を減らす事ができるのではないかと考えられる。

ここで、入力データ群で異常を検出するとき、入力データの特徴が変化した場合には、下記のような事柄が、異常として出力結果に現れてくると推測した。

指標 1：入力データとそれに対する勝利ノードの結合ベクトルとの距離が異常に離れているものが多数存在する。

$$\frac{\sum_{t_1}^{t_2} \delta(d_v(t) - Th_1)}{t_2 - t_1} > Th_2$$

$$\delta(\psi) = \begin{cases} 0 & (\psi \leq 0) \\ 1 & (\psi > 0) \end{cases}$$

指標 2：全く勝利しないノードが多数存在する。

$$\frac{M - \sum_{j=0}^M \delta(cf(j, t_1, t_2))}{M} > Th_3$$

指標 3：勝利頻度が異常に多いノードが存在する。

$$\max_j \left(\frac{cf(j, t_1, t_2)}{t_2 - t_1} \right) > Th_4$$

ここで、 t はデータを入力した時点、 $cf(j, t_1, t_2)$ は t_1 から t_2 の間にノード j が勝利した回数、そして、 Th_1 、 Th_2 、 Th_3 、 Th_4 はそれぞれ閾値を表している。

Th_1 は分類過程の分解能の役割であり、この値が高いと分解能が低くなる。逆にこの値が低いと分解能は高くなり、少しの入力の違いにも反応する。 Th_2 は入力されたデータの中で異常と判断されるデータがどれだけ含まれるか、その割合を示している。 Th_3 はマップ (出力ノード) の有効活用率、すなわち、マップ全体における境界ノードの割合を示しており、通常、マップの 50% 以下が妥当であると考えられる。 Th_4 に関しては、一つのノードに出力が集中するのはマップの無駄使いである、つまり、それだけのデータが出力されるのであればそのようなデータに対する分解能を高くするために他のデータよりも多くのノード (領域) を使用するべきであるとの考えから設定したものである。

各閾値はデータの種類、分散の具合、マップを用いる環境等、様々な要因により決定方法が異なってくる。今回の実験では予備実験による試行錯誤の結果を考慮して値を決定した。

このような指標によって異常が検出できれば、データの特徴の変化にも 3.2 章に記す追加学習等によって対応できるのではないかと考えた。

異常検出についての実験および結果・考察は 5.1 章に示す。

3.2. 自動追加学習

入力データの変化が大幅な場合、そのデータを適正に分類するようなマップは、元のマップとは全く異なったものになると考えられるので、マップは全く初めから再学習させるのが適当であろう。

入力データの特徴が少しずつ変化していく場合、はじめは学習データと同様の特徴を持つため、入力データは適切にクラスタリングされ、データの特徴が変化するにつれて徐々に適切なクラスタリングが行われなくなってくると思われる。だが、その変化がゆるやかであれば、それらのデータを適正に分類するようなマップは、元のマップを少し修正するだけで得られるのではないかと推測される。また、多数クラスのうちの少数のクラスのデータの特徴のみが変化する場合、入力データの大部分は適正に分類されるので、これも元のマップを少し修正するだけでそれらのデータを適正に分類するマップを得られるのではないかと推測される。

再学習の場合はネットワークを初めから構成し直すが、追加学習では元のネットワークに追加的に学習を施す。そのため、学習に多大な計算時間を要する大規模なネットワークの場合、再学習と追加学習では計算時間に大きな差が生まれる。よって、データの変化が大幅な場合以外はマップを初めから再学習させるよりも現在のマップに追加的に学習を施す方が妥当だと考えられる。

また、SOM は教師なし学習であるので、入力データを学習に必要な数だけ蓄えておき、異常の検出後それを学習データとして利用すれば、学習を自動的に行う事ができる。

そこで、入力データの特徴が上記のような変化をする場合、3.1章において記した3つの指標で異常を検出し、変化した入力データを使って現在のマップに追加学習を自動的に行うことで、変化後のデータを適正に分類するマップを自動的に得られるのではないかと考えた。

ここで、そのアルゴリズムを示す。

(1) 学習データ

初期学習に用いた学習データを初期の追加学習用データ Y とする。

$$Y = \{y_i(s) | i=1, 2, \dots, n; s=1, 2, \dots, S\}$$

(2) 適用

入力されたデータ x と予め用意しておいた(1)のデータとを置き換えていく。

$$y_i(s+1) = y_i(s)$$

$$y_i(1) = x_i$$

(3) 異常検出

3.1異常検出の3つの指標により、異常の発生を監査し、異常が検出されたらデータ Y を用いた追加学習の処理に移る。

(4) 追加学習

学習アルゴリズム自体は2.1SOMの学習アルゴリズムに示したものと同一のため省略する。

ただし、結合ベクトルは初期化を行わずに現在のマップの値に追加的に学習を行う。

(5) 繰り返し

(1)～(4)を繰り返す。

この手法は入力の変化に逐次適応していくのではなく、ある程度の時間的余裕を持って変化を検出し、適応していくという方法であるため、雑音に弱いARTなどの方法と異なり、耐雑音性に関しても問題はないと考えられる。

その上、異常検出の指標1において Th_2 を0.0とすると、ARTと同様に未知の入力が来れば直ちに追加学習を施す事ができる。つまり、必要に応じて汎化能力と追加学習のバランスをとることができる。

自動追加学習についての実験および結果は5.2章に示す。

4. 半自動ラベル付け

追加学習を施したマップは競合および近傍学習というSOMの学習の特徴によって元のマップとは当然違ったものになる。したがって、各ノードに固定されているラベルは新しいマップに対応しなくなるため、追加学習後にラベルの付け替えが必要になる。追加学習を行う度に人手で古いマップのラベルと対応させながらラベル付けを行う事は、せっかく追加学習を自動的行えたにも関わらず大きな手間となってしまう。

そこで、動的な変化がないデータに対しては新しいマップ上の位置を自動的に検出し、また、追加学習によって新しく学習されたデータに対してはそのことを検出できれば、新しいラベル付けに対する指標が与える事が可能であると考えた。そのアルゴリズムを以下に示す。

- (1) 追加学習の際、学習率が十分小さくなり ($\alpha(t_s) \leq 0.1$ 程度)、マップが概ね構成されたと考えられる時点 t_a から学習終了までの各ノードの勝利回数 $cf(j, t_a, T)$ を記録しておく。
- (2) 追加学習後、古いマップから各クラスの代表点データと新しいマップの各ノードの結合ベクトルとの距離 d_v を(*)式を用いて計算し、距離が一番近いクラスのラベルをそのノードのラベルとする。ただし距離が一定以上離れている場合 ($d_v > Th_5$: Th_5 は閾値) ラベルはつかないものとする。つまりそのクラスのデータは変化したと考え、そのクラスに属するノードにはラベルは付けない。
- (3) (1)と(2)により、データが変化したクラスは新しいマップ上ではラベルの付けられていない空白の領域として表示されるので、必要に応じてその領域にだけ人手で新しいラベルをつける。

これにより変化していないデータのラベルは新しいマップの位置および領域が検出され、変化したデータに対してはその領域が空白になることから新しいラベルを付ける必要があることを提示する。すなわち、追加学習後のラベル付けはデータの変化したクラスのラベルだけでよく、ラベル付けの半自動化が達成される。

半自動ラベル付けについての実験および結果は6章

に示す。

5. 異常検出及び自動追加学習の実験

実験は、3章において述べたようなデータの特徴が変化したときにおける異常の検出が可能であるか、異常の検出後のマップの追加学習が可能であるかの2点を検証することを目的に行った。

5.1. 実験1：異常の検出

実験1では、3.1章において推測したような異常が、データの特徴変化に際して実際に現れるのかどうか、また、現れたときにはそれを適切に検出することができるのかどうか検証した。

実験は、データの特徴変化に対応できないオリジナルのSOMに対して、特徴が変化していく入力データを提示し、データの特徴の変化によって出力結果と各指標が変化するか、そして、それが検出されるかについて観察を行った。

ここで、マップサイズ 5×5 、 10×10 、 15×15 、 20×20 において、学習回数1000、5000、10000回、結合ベク

トルの初期値はランダム関数により与えたものについての実験を行ったが、いずれの場合においても同様の傾向の結果が得られたため、その代表としてマップサイズ 10×10 、学習回数は10000回の場合の結果を示す。また、閾値 Th_1 は5.1.1実験データに記すように学習データが分散0.1の正規分布をとっているので、データの95%は平均からの距離が0.5以内に存在するとして、0.5と設定した。近傍関数の各変数は以下のように設定した。

$$\alpha(0) = 0.8, \rho(0) = 10.0$$

$$C_1 = T/50, C_2 = T/500$$

$$Th_1 = 0.5, t_2 - t_1 = 1000$$

5.1.1. 実験データ

図5.2 データ変化前の各ノードの勝利頻度

図5.3 データ変化後の各ノードの勝利頻度

表5.1 入力データの中心 $U(u_1, u_2, u_3)$ の推移

	Step0	Step1	Step2	...	Step5
A	(0.0,0.0,0.0)	(0.0,0.0,0.0)	(0.0,0.0,0.0)	...	(0.0,0.0,0.0)
B	(5.0,0.0,0.0)	(6.0,0.0,0.0)	(7.0,0.0,0.0)	...	(10.0,0.0,0.0)
C	(0.0,5.0,0.0)	(0.0,6.0,0.0)	(0.0,7.0,0.0)	...	(0.0,10.0,0.0)
D	(0.0,0.0,5.0)	(0.0,0.0,6.0)	(0.0,0.0,7.0)	...	(0.0,0.0,10.0)
E	(5.0,5.0,0.0)	(6.0,6.0,0.0)	(7.0,7.0,0.0)	...	(10.0,10.0,0.0)
F	(5.0,0.0,5.0)	(6.0,0.0,6.0)	(7.0,0.0,7.0)	...	(10.0,0.0,10.0)
G	(0.0,5.0,5.0)	(0.0,6.0,6.0)	(0.0,7.0,7.0)	...	(0.0,10.0,10.0)
H	(5.0,5.0,5.0)	(6.0,6.0,6.0)	(7.0,7.0,7.0)	...	(10.0,10.0,10.0)

図5.1 入力データの中心 $U(u_1, u_2, u_3)$ の変化

使用した学習データおよび入力データ X は、3次元空間において下式のように中心 $U = \{\mu_i | i=1, 2, 3\}$ とその周りにばらつく点の集合と想定した。このとき、ばらつきとして分散 σ^2 が0.1の正規乱数を発生させた。

$$X(x_1, x_2, x_3), U(u_1, u_2, u_3) \\ x_i = \sqrt{-2\sigma^2 \text{Rand}_2} \sin 2\pi \text{Rand}_3 + \mu_i \dots\dots\dots (**)$$

ここで、 Rand_2 および Rand_3 は0.0から1.0の範囲の一樣乱数を表している。そして、8個のクラスに分類されるようなデータを、学習データとして各クラス100個、計800個。入力データとして各クラス1000個、計8000個を用意した。

入力データは、中心 U が始め学習データと同じであるが、A以外のクラスのデータがベクトルの向きを変えず等幅に緩やかに増加していくようにした。表5.1および図5.1に各クラスの中心の推移を示す。

5.1.2. 結果、考察

図5.2は、はじめのマップに step0のデータを、図5.3は、はじめのマップに step5のデータを入力したときの各ノードの勝利頻度を表している。なお、縦軸はその出力ノードの勝利回数である。すなわち、その数字が大きいほど出力頻度が高い。

ここで、図5.4では入力データが Step0から Step1、Step2、…と順に Step 間を小刻みに変化していく過程における、指標1、2、3の値の変化を示している。

図5.2と図5.3を比較してみると明らかなように、図5.2はある程度満遍なく出力されているのに対し、図5.3は各クラスのデータが少数のノードに片寄って出力されていることがわかる。図5.3の手前左側の領域(クラスAにあたる領域)以外は大抵の領域が出力されておらず、出力のあるノードは異常な頻度で出力されてい

ることが容易に見てとれる。これは3.1異常検出の指標2及び指標3によって検出された異常を視覚化したものといえる。

図5.4で明らかなように、データ変化がおこると急激に各指標の値が増大し、その後は飽和している。これらから、閾値として、 Th_2 を0.5、 Th_3 を0.5、 Th_4 を0.075と設定することで指標の変化から異常を検出できることがわかる。

5.2. 実験2：自動追加学習

実験1で、データの特徴変化が検出できることが言えた。実験2では、3.2章の検討に基づき、特徴変化後のデータを利用して自動的に追加学習を行い、変化後のデータに対して適切なクラスタリングを行うことができるのかどうかについて検証した。また、提案手法である追加学習と、マップを初期から学習させる再学習及び異常があれば逐次に追加学習を行う逐次学習との比較を行い、提案手法の有効性についても検証した。また、再学習とは異常検出後、マップを初期化して改めて学習を行ったものであり、逐次学習とは異常検出の指標1の Th_2 を0.0と設定して追加学習を行ったものである。

ここで、「適切なクラスタリング」とは、

- (1) 初期状態のマップに学習データを入力とした場合のクラスタリングを「適切なクラスタリング」の基準とする。
- (2) データが変化した場合、それらのデータに対してマップは適切なクラスタリングができない。

という2つの認識に基づき、変化後のデータを初期状態と同程度にクラスタリングできる状態を適切なクラ

図5.4 指標の変化

図5.5 追加学習後の各ノードの勝利頻度

表5.2 各学習方法における平均距離

	追加学習	再学習	逐次学習
平均距離	0.292814	0.292831	0.294084
学習回数	1000	10000	1000
初期近傍	3	10	3
初期学習率	0.5	0.8	0.5
処理回数	1	1	347

スタリングと考えた。

実験はマップサイズ 5×5 、 10×10 、 15×15 、 20×20 において、学習回数1000、5000、10000回、追加学習回数100、500、1000回、結合ベクトルの初期値はランダム関数により与えたものについてを行ったが、いずれの場合においても同様の傾向の結果を得られたため、その代表としてマップサイズ 10×10 とし、初めのマップ生成の際の学習回数、学習率および近傍領域の更新率は実験1と同様とした結果を以下に示す。

異常の検出は実験1で決定した閾値を用いた。

5.2.1. 実験データ

入力データは、オリジナルの SOM との比較のため、実験1において使用したデータと同じものを使用した。追加学習のための学習データ Y として、最新の入力データから学習データと同じ800個を保存して使用した。

5.2.2. 結果、考察

図5.5に追加学習後の各ノードの勝利頻度を、図5.4の「re」に追加学習後の各指標の値を示す。

表5.2には追加学習と再学習および逐次学習との比較を示すために、Step0から Step5までの間にそれぞれの追加および再学習を行った回数と学習後の平均距離を示す。ここで、平均距離とは再学習および追加学習後の入力データと勝利ノードとの距離の平均値を指す。

図5.5、図5.4の「re」をみるとわかる通り、追加学習後のマップは変化後のデータを変化前と同様に適切にクラスタリングしている。よって、追加学習によりデータの特徴変化後も適切なクラスタリングが行われることが確認できた。

表5.2をみると、追加学習は再学習、逐次学習と比べても平均距離に差は見られないことが分かる。また、追加学習では初期化されたマップによる再学習の10分の1程度である1000回の学習で十分であった。同様に学習率および近傍の初期値も再学習よりも小さい値である $\alpha(0)=0.5$ 、 $\rho(0)=3.0$ で十分であった。追加学習では再学習と比べて、追加学習時の学習回数、初期学習率、初期近傍が小さく、学習時間が大幅に短縮され

ている。

また、逐次学習は他の2手法が1回であるのに比べて347回と明らかに追加学習の処理回数が突出している。

これらのことから、提案手法である追加学習の有効性、及び入力の変化に逐次適応していくのではなく、ある程度の時間的余裕を持って変化を検出することについての有効性を示すことができた。

6. 半自動ラベル付けの実験

5章の実験で異常の検出と自動追加学習が可能である事がいえた。この章では追加学習後、4章で述べた手法により適切にラベルの位置を検出することができるかどうかを以下の実験で検証した。

ここで、マップサイズ 5×5 、 10×10 、 15×15 、 20×20 において、学習回数1000、5000、10000回、結合ベクトルの初期値はランダム関数により与えたものについての実験を行ったが、いずれの場合においても同様の傾向の結果を得られたため、その代表としてマップサイズ 10×10 とし、学習回数 T は10000回、追加学習回数は1000回の場合の結果を示す。また、学習率と近傍は5章と同様に減少および縮小させた。

6.1. 実験データ

使用したデータは5章と同様、3次元空間において、中心とその周りにばらつく点の集合と想定した。この

表6.1 実験データの中心の変化の推移

	Step0	Step1	Step2	...	Step5
A	(5.0,0.0,0.0)	(5.0,0.0,0.0)	(5.0,0.0,0.0)	...	(5.0,0.0,0.0)
B	(0.0,0.0,5.0)	(0.0,0.0,5.0)	(0.0,0.0,5.0)	...	(0.0,0.0,5.0)
C	(5.0,5.0,5.0)	(5.0,5.0,4.0)	(5.0,5.0,3.0)	...	(5.0,5.0,0.0)
		(4.0,5.0,5.0)	(3.0,5.0,5.0)	...	(0.0,5.0,5.0)

図6.1 実験データの中心の変化の推移

図6.2 はじめのマップにつけられたラベル

とき、そのばらつきとして、式(**)により分散が0.1の正規乱数を発生させた。そして、クラスが3つに分類されるようなデータを学習データとして各クラス100個、計300個、入力データとして各クラス1000個、計3000個を用意した。このとき、入力データとしてデータCだけが2つに分かれて変化していくようにした。表6.1および図6.1に各クラスの中心の推移を示す。また、ラベリングにおける閾値 Th_5 は1.0と設定した。

6.2. 結果、考察

実験では、入力データが変化する度に異常が検出され追加学習が行われた。

図6.2は初めの特徴マップにつけたラベルである。「-」はラベル付けされなかった空白部分をあらわし「#」は勝利頻度が低い、各クラスの境界線を表している。図6.3はstep5で追加学習後に自動的につけたラベルの図である。データの変化していないクラス(AとB)については自動的にラベル付けされており、変化したクラス(C)は表示されず、空白部分が発生している。さらに空白部分は2つに分かれており、2つのクラスについてラベル付けをすればよいことがわかる。

7. 結論

本研究では、SOMにおいてデータの特徴変化に対する自動追加学習システムを提案した。

5章の実験結果から、データの特徴変化が緩やかな場合や、多数クラスの中の少数のクラスの特徴のみが変化する場合、このシステムの追加学習法によって期待通りにマップを更新させることが確認できた。これは、時間的变化に伴うデータの特徴変化に対応できなかったSOMに新しい可能性を示したといえる

図6.3 追加学習後のマップにつけられたラベル

であろう。

また、6章の実験結果から、このシステムの半自動ラベル付け法によって、追加学習後のラベル付けは、特徴が変化したクラスのラベルだけでよくなり、ラベル付けの半自動化が実現された。

しかし、改善していかなければならない点が実験を通して明らかになった。今回は、半自動ラベル付けにおけるクラスの境界を表す際の閾値 Th_5 を経験的に妥当と考えた数値としたが、これは、データの分散等が既知であったために設定することができたものである。また、分散等が分かっているデータである場合でも、その分類において求められる精度よりも離れた数値に Th_5 を設定する事ができる。しかし、データによってはそういった条件が異なってくることも考えられる。クラス数が変化するとき空白部分での境界線は重要となってくるので、十分な実験および考察が必要と考える。

なお、先行研究との比較として、従来のSOMとの比較に関しては追加学習前までは従来のSOMそのものなので異常が現れることを示すことによって従来のSOMとの比較とした。また、ARTに関しては、提案する手法において、3.1異常検出の章で記した指標1の Th_2 を0.0とすることにより、異常が出ればすぐ追加学習するようにすることでARTと同様の特性となる。従って、提案する手法はARTの特性を包含する手法と考えた。

参考文献

- [1] R. ビール T. ジャクソン 著 八名和夫 監訳:「ニューラルコンピューティング入門」海文堂(1993)ISBN4-303-72640-0
- [2] T. コホネン 著 徳高平蔵 岸田悟 藤村喜久朗 訳:「自己組織化マップ」シュブリンガーフェアラーク東京(1996)ISBN4-431-70700-X C3055
- [3] T. コホネン 著 中谷和夫 訳:「自己組織化と連想記

憶」シュプリングーフェアラーク東京(1993) ISBN4-431-70636-4 C3055

[4] R. ヘクト, ニールセン 著 袋谷賢吉 訳: 「ニューロコンピューティング」トッパン(1992) ISBN4-8101-8035-2

[5] PHILIP D. WASSERMAN 著 石井直宏 塚田稔 訳: 「ニューラルコンピューティング」森北出版(1993) ISBN4-627-82200-6

(2001年3月27日 受付)

(2002年1月29日 採録)

[問い合わせ先]

〒680-8552 鳥取市湖山町南4-101

鳥取大学工学部知能情報工学科

三好 力

Tel & Fax : 0857-31-5221

E-mail : mijosxi@ike.tottori-u.ac.jp

著者略歴

みちはた まさひろ
道端 正裕 [正会員]

2001年 鳥取大学大学院工学研究科知能情報工学専攻修了。同年 東京大学生産技術研究所 技術官。現在、自己組織化ネットワーク等の研究に従事。日本ファジィ学会会員。

みよし つとむ
三好 力 [正会員]

1985年 大阪府立大学工学部電気工学科卒業。同年 シャープ株式会社技術本部研究員。1989年-1993年 技術研究組合 国際ファジィ工学研究所 主任研究員。1993年-1997年 シャープ株式会社技術本部研究員。1997年より 鳥取大学工学部知能情報工学科助教授。この間、構造化マルチメディア文書処理、人工知能、知識工学、あいまい理論、知的インタフェース、医療情報処理、携帯端末のネットワーク活用、自己組織化ネットワーク等の研究に従事。情報処理学会、日本ファジィ学会、ACM 各会員。博士 (工学)。

ますやま ひろし
増山 博 [非会員]

昭和44年 広島大学大学院修士課程(電気工学専攻) 修了後、広島大学助手、宮崎大学助教授・教授、Stanford・Boston 両大学客員教授及び大阪大学医学系研究科教授(併任)を経て、現在、鳥取大学・知能情報工学科教授。非同期論理回路、計算機アーキテクチャ等の研究を経て、現在、主に並列・分散処理システムやコンピュータ・ネットワークの高信頼化に関する研究に従事。工博。電子情報通信学会、情報処理学会及び IEEE 各会員。

Automatic Incremental Learning System on Self-Organizing Map

by

Masahiro MICHIHATA, Tsutomu MIYOSHI, Hiroshi MASUYAMA

Abstract :

Kohonen's Self-Organizing Map (SOM) is a kind of neural networks that learns the feature of multi input data without supervision. Therefore, it seems difficult to apply the SOM, in the case that the feature of input data changes from learning example data. In this paper, the authors proposed that "Automatic Incremental Learning System of Self-Organizing Map" is effective for its solution. This system consists of two techniques. The one works automatic learning of the input data feature, when the input data has partly been shifted from features of learning example, and the other is "semi-automatic labeling" on new SOM feature map that adjust territory labels by using information from the former map.

Key Words : Neural Network, Self-Organizing Map, Unsupervised Learning, Evaluation