

怠け者の割合を考慮した Lazy Self-Organizing Map とその振る舞い

原口 卓[†] 松下 春奈[†] 西尾 芳文[†]

[†] 徳島大学工学部 〒 770-8506 徳島県徳島市南常三島 2-1
E-mail: †{taku,haruna,nishio}@ee.tokushima-u.ac.jp

あらまし 自己組織化マップ (Self-Organizing Map: SOM) は Kohonen 氏によって提案された教師なしニューラルネットワークであり, 近年クラスタリングへの応用として注目されている. 過去の研究で, SOM の学習率係数が更新回数に依存するのに対し, ニューロンの性格にのみ依存する学習率係数を持つ Lazy Self-Organizing Map (LSOM) を提案した. さらに, その LSOM の振る舞いを調査することで, SOM よりも入力データの分布状態をより反映したマップを取得できる反面, ねじれが生じやすいという傾向を確認した. そこで本研究では, 学習率係数がニューロンの性格, 怠け者の割合に依存し, さらに更新回数に従って減少する新しい Lazy Self-Organizing Map を提案する. そして, 改良版 LSOM の怠けニューロンの割合を変えて, 様々な入力データに適用しその振る舞いと有効性を確認する.

キーワード 自己組織化マップ, クラスタリング, 特徴抽出

Behaviors of Lazy Self-Organizing Map Considering Lazy-Neuron Rate

Taku HARAGUCHI[†], Haruna MATSUSHITA[†], and Yoshifumi NISHIO[†]

[†] Department of Electrical and Electronic Engineering, Tokushima University 2-1, Minami-Josanjima,
Tokushima, 770-8506 Japan
E-mail: †{taku,haruna,nishio}@ee.tokushima-u.ac.jp

Abstract The Self-Organizing Map (SOM) is unsupervised neural networks introduced by Kohonen. The SOM attracts attentions for clustering in these years. In the previous study, we have proposed the Lazy Self-Organizing Map (LSOM), whose learning rate depends on only each neuron's character. Meanwhile, the learning rate of the conventional SOM depends on learning step. We investigated its behaviors, however, we confirmed that the previous LSOM tend to obtain effective map reflecting the distribution state of the input data and a strongly twist map than the conventional SOM. In this study, we propose an improved LSOM, whose learning rate depends on each neuron's character, lazy-neuron rate and decreases according to learning step. We apply the improved LSOM considering lazy-neuron rate to various input data set and confirm its behaviors and effectiveness.

Key words self-organizing map (SOM), clustering, feature extraction

1. ま え が き

情報技術の急速な発達に伴い, 世界中でデータの量が複雑かつ膨大になっていることから, データマイニングは重要な課題である. またデータマイニングにおいて, クラスタリングは代表的な分析手法の一つであり, 統計, パターン認識, 画像分析など多くの分野で盛んに研究されている. そこで近年, 自己組織化マップ (Self-Organizing Map: SOM) [1] を利用したクラスタリングに関する研究が注目されている [2]. SOM は 1982 年に Kohonen 氏によって提案された教師なしニューラルネットワークであり, 教師信号がなくても自分自身で内部構造を変化できることから脳の自己組織化過程を簡略化したモデルであるとも言われている. また SOM は, ニューロン間のつながりで

ある位相的順位を保持できるという利点を持ち, 類似性のあるデータを分類するのに適している.

私たちは過去の研究で, より効率の良い自己組織化を行うために, SOM にアリの世界を適用することを提案した [3]. その背景に, まだ明確な理由や確実性は立証されていないが, 働きアリの約 20% が働いていないという興味深い報告 [4] がある. また, これを報告した研究者はこの怠けアリにもなんらかの役割があるのではないかと考えている. その裏づけに, シミュレーション結果ではあるが, 優秀なアリだけのグループよりも怠けアリを含むグループの方がより効率的に餌を集められるという報告がある. これらのことから, 私達は怠けニューロンを含んだ自己組織化マップ (Lazy Self-Organizing Map: LSOM) を提案した. LSOM の重要な特徴は, 3 種類のニューロン (働き

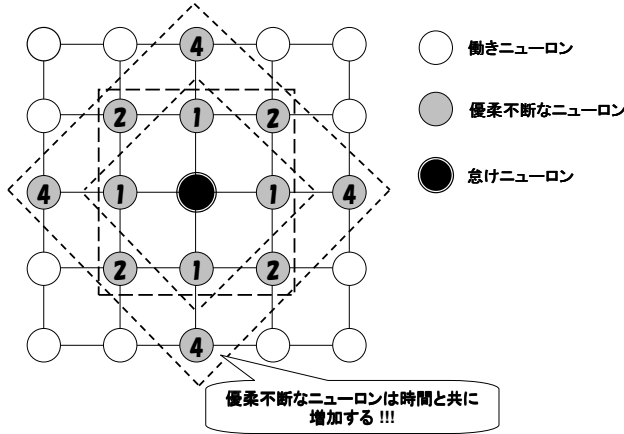


図 1 3 種類のニューロンを含む LSOM: 働きニューロン, 怠けニューロン, 優柔不断なニューロン. それぞれの数字は怠けニューロンとそれぞれのニューロン間の近傍距離を表す.

ニューロン, 怠けニューロン, 優柔不断なニューロン) を持つということである. 怠けニューロンは働きニューロンよりも学習率係数が小さく, 更新幅が少ないニューロンである. 優柔不断なニューロンは怠けニューロンの近傍であり, その影響を受けて学習率係数が働きニューロンよりも小さくなるニューロンである. このように, LSOM の学習率係数はニューロンの性格に依存することから, SOM よりも入力状態をより反映したマップを取得することができた. しかし, その得られたマップにはねじれが生じているなどの位相的順位が正確に保持されていないという欠点があった.

そこで本研究では, 従来版 LSOM の特徴を残しつつ, 位相的順位を保持できるように改良した新しい LSOM を提案する. 改良版 LSOM の学習率係数は, ニューロンの性格, 怠けニューロンの割合に依存し, さらに更新回数に従って減少する. これは, 改良版 LSOM が従来版 LSOM よりも従来の SOM に近い特性を持つことを意味する. さらに, それをさまざまな入力データに適用することで, 改良版 LSOM のアルゴリズムの有効性を確認する. また, 異なる怠けニューロンの割合においてもシミュレーションを行い, 怠けニューロンの割合に対する改良版 LSOM の振る舞いを調査する.

改良版 LSOM の新しいアルゴリズムと怠けニューロンの割合を考慮したときの有効性を調査し, その振る舞いを確認する.

2. 改良版 Lazy Self-Organizing Map (LSOM)

過去の研究で, 3 種類のニューロン (働きニューロン, 怠けニューロン, 優柔不断なニューロン) を含む LSOM を提案した (図 1). この 3 種類のニューロンには, それぞれ学習率係数を用いることで個性を持たせた. 働きニューロンは学習率係数が最も大きい値であり, 更新幅が大きいことが特徴である. 怠けニューロンは働きニューロンよりも学習率係数が小さく, 更新幅が少ないニューロンである. 優柔不断なニューロンは怠けニューロンの近傍に位置するニューロンであり, その影響を受けて学習率係数が働きニューロンよりも小さくなるニューロンである. 一方, 従来の SOM の学習率係数は更新回数に従って

減少するため, 学習の最終段階では小さな値となり, ニューロンはほとんど更新されなくなる. しかし, 従来版 LSOM は SOM と違い, 更新回数に従って減少せずニューロンの性格のみ依存する. つまり, 学習率係数は固定の値となり, 学習の最終段階でも SOM に比べ大きな値をとるためにニューロンは大きく更新される. これらの理由から, 従来版 LSOM は SOM よりも入力の分布状態をより反映したマップを取得できる反面, ねじれが生じやすいという欠点があった.

そこで本研究では, 従来版 LSOM の特徴を残しつつ, 位相的順位を保持できるような新しい LSOM を提案する. この改良版 LSOM の学習率係数は, 従来版 LSOM と同様にニューロンの性格によって各ニューロンが異なる値を持つのに加え, 怠けニューロンの割合にも影響を受ける. また, この値は固定値ではなく, 更新回数に従い, 徐々に減少する. これは, 改良版 LSOM が従来版 LSOM よりも SOM に近い特性を持つことを意味する.

改良版 LSOM の学習アルゴリズムについて詳しく説明する. 改良版 LSOM は入力層と競合層の 2 層構造であり, 競合層では M 個のニューロンが 2 次元格子状に配列されている. それぞれのニューロンは重みベクトル $w_i = (w_{i1}, w_{i2}, \dots, w_{id})$ ($i = 1, 2, \dots, M$) を持ち, その初期値は, 0 から 1 の範囲でランダムに与えられる. さらに, ランダムに選ばれた $(p \times M)$ 個のニューロンは, 怠けニューロンとしての集合 S_{lazy} に分類される.

(LSOM1) 入力ベクトル $x_j = (x_{j1}, x_{j2}, \dots, x_{jd})$ ($j = 1, 2, \dots, N$) を全てのニューロンに同時に入力する.

(LSOM2) 入力ベクトル x_j とニューロン i の重みベクトル w_i との距離を計算し, 入力ベクトル x_j に最も近い重みベクトルを持つニューロンを勝者ニューロン c に決定する.

$$c = \arg \min_i \{\|w_i - x_j\|\}, \quad (1)$$

本研究では, $\|\cdot\|$ はユークリッド距離を表している.

もし $c \in S_{\text{lazy}}$ なら (LSOM3) を, そうでないなら (LSOM4) を実行する.

(LSOM3) 勝者ニューロン c になった怠けニューロンは, S_{lazy} から取り除かれる. さらに, 入力データ x_j から最も遠く, S_{lazy} の中に存在しないニューロン f が S_{lazy} に入るように選ばれる.

$$f = \arg \max_i \{\|w_i - x_j\|\}, \quad i \notin S_{\text{lazy}}. \quad (2)$$

つまり, 勝者ニューロン c になった怠けニューロンは働きニューロンになり, もう一方のニューロン f が怠けニューロンになる (図 2).

(LSOM4) 怠けニューロンの近傍に位置する働きニューロンが, 怠けニューロンの影響を受けた優柔不断なニューロンになる. 優柔不断なニューロンの集合を N_{lazy} とする.

$$N_{\text{lazy}} = \{i \mid \|r_i - r_l\|^2 \leq D(t), \\ i \neq c, i \notin S_{\text{lazy}}, l \in S_{\text{lazy}}\}, \quad (3)$$

$\|r_i - r_l\|$ は, マップ上におけるされるニューロン i とニューロン l 間の近傍距離であり, $D(t)$ は近傍範囲に一致する. また,

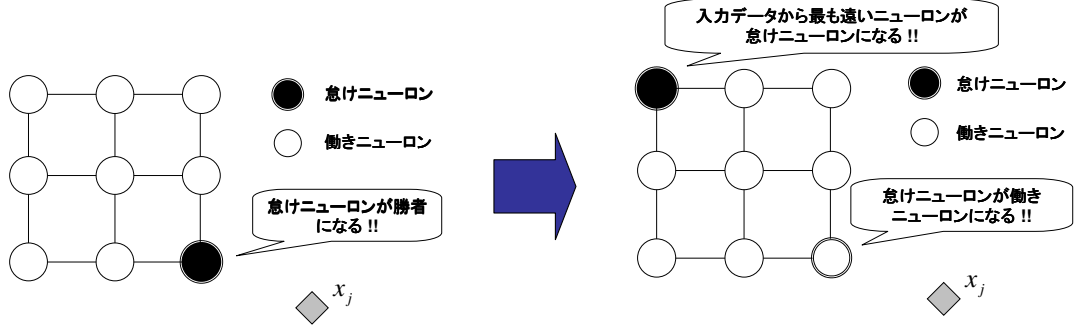


図 2 怠けニューロンが勝者になったとき働きニューロンとの交代方法

$D(t)$ は式 (4) に従って減少する.

$$D(t) = \left[D_{\max} \frac{t}{T} \right], \quad (4)$$

$[\cdot]$ はガウス記号であり, $D(t)$ の最大値である D_{\max} は固定のパラメータで決定される. また, t は現在の学習回数であり, T は最大学習回数である.

(LSOM5) すべてのニューロンの重みベクトルを式 (5) に従って更新する.

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + h_{L_{c,i}}(t)(\mathbf{x}_j - \mathbf{w}_i(t)), \quad (5)$$

$h_{L_{c,i}}(t)$ は LSOM の近傍関数と呼ばれ, 式 (6) のように表される.

$$h_{L_{c,i}}(t) = \alpha(t) \exp\left(-\frac{\|\mathbf{r}_i - \mathbf{r}_c\|^2}{2\sigma^2(t)}\right), \quad (6)$$

$\alpha(t)$ は改良版 LSOM の学習率係数を表し, 式 (7) のようにニューロンの性格と怠けニューロンの割合 p ($0 < p < 1$) によって決定され, 更新回数に従って減少する.

$$\alpha(t) = \begin{cases} \alpha_{\text{lazy}} \left(1 - p \frac{t}{T}\right), & \text{if } i = l, l \in S_{\text{lazy}} \\ \alpha_N \left(1 - (1-p) \frac{t}{T}\right), & \text{if } i \in N_{\text{lazy}} \\ \alpha_w \left(1 - (1-p) \frac{t}{T}\right), & \text{それ以外} \end{cases} \quad (7)$$

$\alpha_w, \alpha_{\text{lazy}}, \alpha_N$ はそれぞれ働きニューロン, 怠けニューロン, 優柔不断なニューロンの初期学習率係数を表す. つまり, $\alpha_{\text{lazy}} < \alpha_N < \alpha_w$ となる.

(LSOM6) (LSOM1) から (LSOM5) までのステップを全ての入力データに対して繰り返す.

3. シミュレーション結果

3.1 ターゲットデータ

まず, 図 3(a) に示すような 2 次元の入力データであるターゲットデータ [5] に対してシミュレーションを行う. 入力データ数 N は 770 個, 2 つのクラスタの他に 4 つの離れた入力データを持ち, 全てのデータはランダムに並び替えている. SOM, 従来版 LSOM, 改良版 LSOM はそれぞれ 100 個 (10×10) のニューロンを持ち, 各 LSOM では全体の 10% のニューロン ($p = 0.1$) を怠けニューロンとする. 学習は全ての入力に対して 20 回繰り返す ($T = 15400$). 学習パラメータは以下のように設定する.

$$\alpha_w = 0.5, \alpha_{\text{lazy}} = 0.05, \alpha_N = 0.25,$$

$$\alpha(0) = 0.5, \sigma(0) = 4.0, D_{\max} = 5.$$

怠けニューロンの有効性を確認するために, α_w は SOM の初期学習率係数 $\alpha(0)$, $\sigma(0)$ は SOM の近傍関数の幅と一致させる.

図 3(b), (c), (d) に SOM, 従来版 LSOM, 改良版 LSOM のシミュレーション結果をそれぞれ示す. 図 3(b) において, SOM は全ての離れた入力データを自己組織化できていないが, 図 3(c) における従来版 LSOM では, ねじれが生じてはいるものの全ての離れた入力データを自己組織化できている. しかし, ねじれは位相的順位が保持できていない現象であり, ねじれのためにニューロン間の距離が遠いにも関わらず, 距離が近いというような誤認が生じる. これはデータ抽出の際, 正確な抽出を困難にするなどの問題を引き起こす. すなわち, ねじれは回避すべき現象であるといえる. 一方, 図 3(d) において, 改良版 LSOM は従来版 LSOM ほど正確ではないが, SOM より離れた入力データを自己組織化し, さらにそのマップにはねじれが生じていない. これは, 改良版 LSOM が, ねじれが生じにくい SOM と, 離れた入力データを自己組織化し易い従来版 LSOM の両方の特徴を持っているからである. 怠けニューロンは 2. 節における (LSOM3) の特徴を持つため, その他のニューロンが勝者になった場合でもあまり動かないという性質がある. すなわち, 離れた入力データを自己組織化し易い. また, 改良版 LSOM の学習率係数は更新回数に従い減少するため, 従来版 LSOM に比べて, 従来の SOM に近い自然な学習が行える. よって, 位相的順位が保持でき, ねじれが生じないのである. これらの理由から, 改良版 LSOM のシミュレーション結果は最も有効であるとわかる.

さらに, LSOM の学習能力を数値的な方法で比較するために以下の利用頻度の高い 3 種類の測定値を用いる.

量子化誤差 Q_e [1]: 量子化誤差 Q_e はそれぞれの入力ベクトルとその勝者との距離の平均を計算した値である.

$$Q_e = \frac{1}{N} \sum_{j=1}^N \|\mathbf{x}_j - \bar{\mathbf{w}}_j\|, \quad (8)$$

$\bar{\mathbf{w}}_j$ は入力データ \mathbf{x}_j に対する勝者ニューロンの重みベクトルである. 従って, Q_e は 0 に近いほど入力状態に近いことを示す. トポロジー誤差 T_e [6]: トポロジー誤差 T_e は SOM がどのくらい入力データのトポロジーを保存できているかを示す値である.

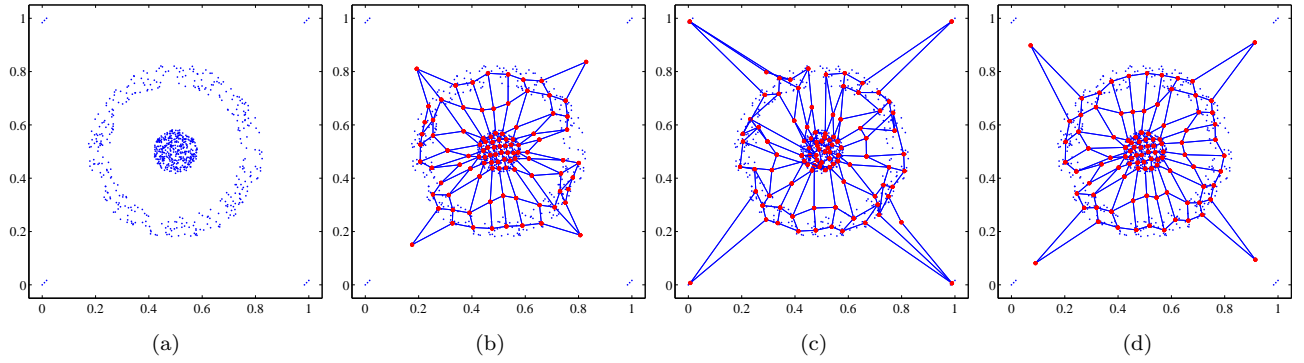


図3 ターゲットデータに対するシミュレーション. (a) 入力データ. (b) SOM. (c) 従来版 LSOM. (d) 改良版 LSOM.

$$Te = \frac{1}{N} \sum_{j=1}^N u(x_j), \quad (9)$$

N は入力データの総数である. また, 入力データ x_j に対する 1 番目の勝者と 2 番目の勝者が互いに 1 近傍以内なら $u(x_j)$ は 0, それ以外なら $u(x_j)$ は 1 となる. つまり, Te は 0 に近いほど入力データのトポロジーを保存できていることを示す.

ニューロン利用率 U [7]: ニューロン利用率 U は 1 度以上勝者になったニューロンの割合を示す値である.

$$U = \frac{1}{nm} \sum_{i=1}^{nm} u_i, \quad (10)$$

もし, ニューロン i が一度以上勝者になったなら $u_i = 1$ で, 一度も勝者にならなかったなら $u_i = 0$ となる. つまり, U は 1 に近いほどより多くのニューロンを有効利用できており, 不活性ニューロンが少ないことを示す.

表 1 に学習結果から計算された 3 種類の評価値をそれぞれ示す. 従来版 LSOM の Q_e は 3 つの中で最も小さな値をとり, U に関しても最も大きな値をとっている. しかし, Te に関しては最も悪い値をとり, SOM に比べて 128.47% も悪くなっている. Q_e とは異なり Te はマップの位相保持の度合を示すので, たとえ Q_e が小さな値をとろうとも, ねじれのあるマップでは Te は大きな値となる. つまり, Q_e と U が SOM よりもどんなに良い値をとろうとも, Te が悪いとそれは有効な結果とはいえない. 一方, 改良版 LSOM の U は SOM と同じ値であるけれども, Q_e は小さな値をとり, 13.7% も改善されている. さらに, Te に関しては SOM に比べ小さな値をとり, 54.74% も改善されている. これらの結果から, 改良版 LSOM は 3 つの中で最もバランスがとれており, 有効であるといえる.

図 4 はニューロン間の距離をグレースケールで表したもので, マップのクラスタ構造を可視化する. マップノード間の距離が遠いほど四角形は黒く表されるため, 白の四角形が一樣にある部分がクラスタである. 各マップはそれぞれの距離で正規化している. 図 4(b) において, 従来版 LSOM の離れた入力は鮮明であるけれども, 入力データの構造を認識するのは困難である. 一方, 図 4(c) における改良版 LSOM では, マップの状態を保持しつつ, 入力データを正確にクラスタリングしているの

表 1 ターゲットデータに対する量子化誤差 Q_e , トポロジー誤差 Te , ニューロン利用率 U .

	Q_e	Te	U
SOM	0.0190	0.1779	0.8100
従来版 LSOM	0.0144	0.4065	0.8300
改良版 LSOM	0.0164	0.0805	0.8100
SOM と改良版 LSOM の改善率	13.6970%	54.7445%	0%

で, 離れた入力データおよび入力データの構造の確認が容易である. つまり, 改良版 LSOM は入力データの構造と位相空間を最も正確に抽出しているといえる.

3.2 アトムデータ

次に, 図 5(a) に示すような 3 次元の入力データであるアトムデータ [5] に対してシミュレーションを行う. アトムデータは線形的に区別できないクラスタリング問題 (異なる密度と分散) を持ち, すべてのデータはランダムに並び替えている. 入力データ数 N は 800 個, 学習は全ての入力データに対して 20 回繰り返す ($T = 16000$). 学習条件はターゲットデータに対するシミュレーションと同じ値にする.

SOM, 従来版 LSOM, 改良版 LSOM の結果を図 5(b), (c), (d) にそれぞれ示す. これらの結果から, 従来版 LSOM は SOM に比べ, 球の表面に分散した入力データを自己組織化できている. しかし, ターゲットデータのときと同様, ニューロン間のつながりを保持できていないことがわかる. 一方, 改良版 LSOM では位相的順位を保持しつつ, 球の表面に分散した入力データを自己組織化できていることがわかる.

表 2 アトムデータに対する量子化誤差 Q_e , トポロジー誤差 Te , ニューロン利用率 U .

	Q_e	Te	U
SOM	0.0558	0.2725	0.8300
従来版 LSOM	0.0509	0.4065	0.8200
改良版 LSOM	0.0483	0.2537	0.8500
SOM と改良版 LSOM の改善率	13.4116%	6.8807%	2.4096%

表 2 に, 学習結果から計算された 3 種類の評価値をそれぞれ示す. 改良後の Q_e は 3 つの中で最も小さな値をとり, U に関し

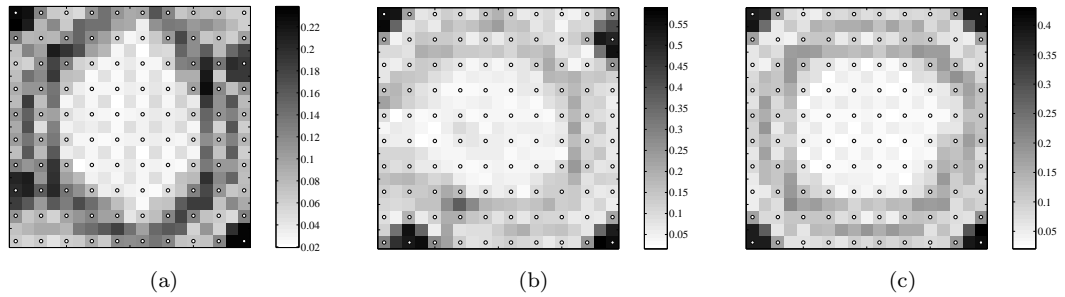


図4 ターゲットデータに対する結果のグレースケール表示. (a) SOM. (b) 従来版 LSOM. (c) 改良版 LSOM.

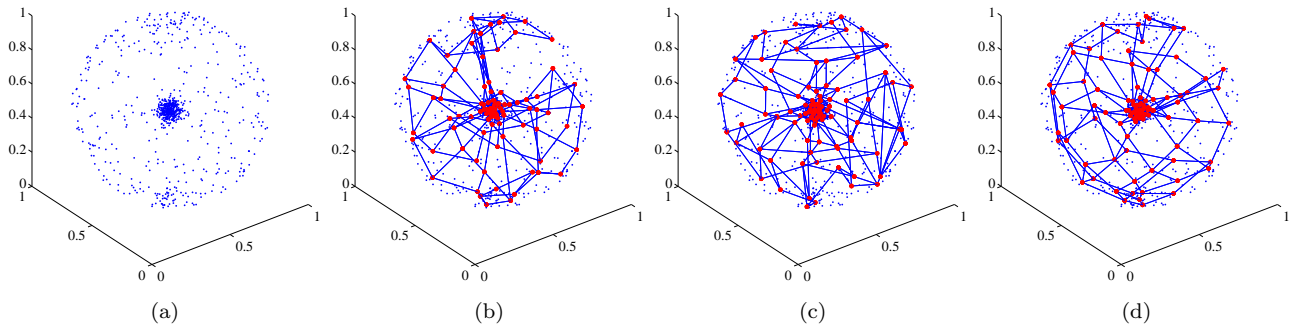


図5 アトムデータに対するシミュレーション結果. (a) 入力データ. (b) SOM. (c) 従来版 LSOM. (d) 改良版 LSOM.

でもまた最も良い値をとっている。さらに、SOM との相対誤差はそれぞれ 13.41%, 2.41% となり大きく改善されていることがわかる。これは入力データと学習後のマップにおけるニューロンとの間に、誤差が少ないことを意味する。つまり、改良版 LSOM は入力データの分布状態を最も反映しているといえる。加えて、改良版 LSOM は $U = 0.85$ 、つまり全体の 85% のニューロンが 1 度以上勝者になっていることから、不活性ニューロンが少ないといえる。不活性ニューロンは、入力データの存在しない所にあるニューロンである。従って、不活性ニューロンが少なければ少ないほど、より多くのニューロンが入力データを自己組織化することになる。その結果、入力データとその入力データに対する勝者ニューロン間の距離がそれぞれ小さくなる。一方、改良版 LSOM は更新回数に依存するため、学習とともに学習率係数は減少する。しかし、SOM とは異なり学習率係数が 0 にはならないために学習の最終段階でも更新が行われる。これらの理由から、 Q_e は小さな値となる。改良版 LSOM の T_e は 3 つの中で最も小さな値をとっていて、SOM に比べて 6.88% 改善されている。これは、学習率係数がニューロンの性格にのみ依存する従来版 LSOM とは違い、改良版 LSOM の学習率係数は SOM の学習率係数のように更新回数に従い減少するため、よりなめらかな学習ができるからである。これらの理由から、改良版 LSOM は最も有効であることがわかる。

4. 怠けニューロンの割合を考慮した LSOM の振る舞い

さらに、怠けニューロンの割合に対する改良版 LSOM の振

る舞いを調査するために、怠けニューロンの割合 p を変えて、ターゲットデータに対してシミュレーションを行う。学習条件はターゲットデータに対するシミュレーションと同じ値にする。

図 6 はそれぞれの p を用いてシミュレーションしたときの SOM から改良版 LSOM の改善率である。改良版 LSOM の Q_e は p に関わらず、SOM よりも良くなっている。しかし T_e に関しては、 p が 0.1 から 0.2 のときは SOM に比べて良くなっているけれども、 p が増加するにつれて T_e は悪くなっている。また、 p が 0.5 から 0.9 のとき U も SOM に比べて悪くなっている。これらの結果から、 p が増加するにつれて位相的順位を保持できなくなることがわかる。つまり、改良版 LSOM は 10% から 20% の怠けニューロンを含むとき位相的順位を保持しつつ、SOM よりも正確に入力データの分布状態を反映したマップを取得できる。

5. まとめ

本研究では、過去の研究で提案した怠けニューロンを含む自己組織化マップ (LSOM) を改良した新しい LSOM を提案した。その LSOM の学習率係数は、ニューロンの性格、怠けニューロンの割合に依存し、さらに更新回数に従って減少する。また、その LSOM を様々な入力データへ適用することで、改良版 LSOM のアルゴリズムの有効性を確認した。さらには、怠けニューロンの割合を変えて、怠けニューロンの割合に対する改良版 LSOM の振る舞いを調査した。その結果、10% から 20% の怠けニューロンを含むとき、改良版 LSOM の有効性が確認できた。

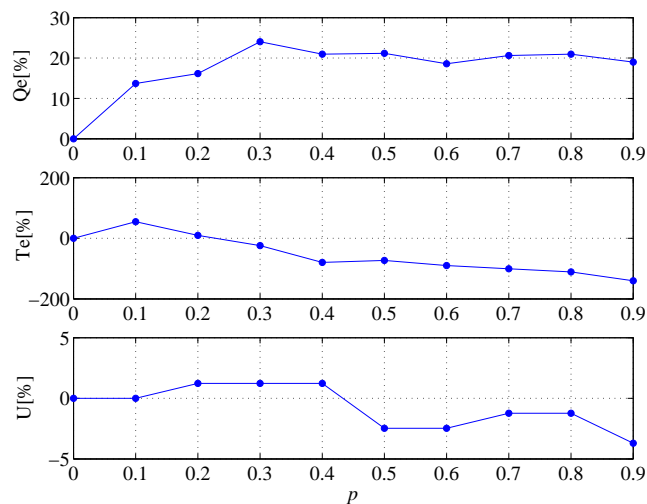


図6 怠けニューロンの割合 p に対する SOM から改良版 LSOM による量子化誤差 Q_e 、トポロジー誤差 T_e 、ニューロン利用率 U の相対誤差 [%].

文 献

- [1] T. Kohonen, *Self-Organizing Maps*, Berlin, Springer, vol. 30, 1995.
- [2] J. Vesanto and E. Alhoniemi, "Clustering of the Self-Organizing Map," *IEEE Trans. Neural Networks*, vol. 11, no. 3, pp. 586–600, 2002.
- [3] T. Haraguchi, H. Matsushita and Y. Nishio, "Lazy Self-Organizing Map and its Behaviors, Proc. of IJCNN'08, pp.2276-2281, June 2008.
- [4] 長谷川 英祐, "「集団」行動の最適化," 日本動物行動学会 Newsletter, no. 43, pp. 22–23, 2004.
- [5] A. Ultsch, "Clustering with SOM: U*C," Proc. Workshop on Self-Organizing Maps, pp.75-82, 2005.
- [6] K. Kiviluoto, "Topology Preservation in Self-Organizing Maps," Proc. of International Conference on Neural Networks, pp. 294-299, 1996.
- [7] Y. Cheung and L. Law, "Rival-Model Penalized Self-Organizing Map," *IEEE Trans. Neural Networks*, vol. 18, no. 1, pp. 289–295, 2007.