



Title	GHSOMのパラメータ設定による計算時間の比較
Author(s)	竹内, 尚; 鈴木, 育男; 山本, 雅人 他
Citation	情報処理北海道シンポジウム講演論文集, 2009, 66-67
Issue Date	2009-10-03
Doc URL	https://hdl.handle.net/2115/51067
Rights	ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うをお願いいたします。
Type	journal article
File Information	Hokkaidosympo20096667.pdf



GHSOMのパラメータ設定による計算時間の比較

竹内尚* 鈴木育男 山本雅人 古川正志

(北大情報)[†]

1 はじめに

画像や音楽などの多次元データの分類に現在広く用いられているものに、自己組織化マップ (Self-Organizing Map, 以下 SOM) がある。SOM は、ユニットという重みベクトルにネットワークの関連性をもたせ、近傍学習という概念を与えることで、入力の特ポロジカルマッピングを行うことのできる、教師なしニューラルネットワークの一種である。

SOM は様々な問題に応用できるアルゴリズムであり、様々な用途に特化したアルゴリズムが考え出されている。一方で SOM は学習を始める前にユニットの個数を決める必要があり、これによって得られる結果が異なってくることが知られている。この問題を解決するための手法として Growing Hierarchical SOM (以下 GHSOM) があり、入力データ分布が密である部分にユニット、または、階層を新たに追加することで、参照ベクトル数の初期値に依存しない学習を可能とするアルゴリズムである。また、GHSOM は成長過程におけるパラメータによってマップの構造を変えることができ、パラメータ次第で SOM の計算時間を減少させることができる。

そこで本研究では SOM との計算時間の比較を行いその結果から考察を行う。

2 自己組織化マップ SOM

自己組織化マップ (SOM) は、T. Kohonen [3] が提案した教師なしニューラルネットワークである。SOM の学習は、参照ベクトルが初期化された状態から始まる。まず、 n 次元の入力データからランダムに選出したデータを SOM に与える。ユニット m_i のうち、入力データの距離が最も近いものを勝者 (Best Matching Unit, BMU) と呼び c で表す。 n 次元の入力を $x \in \mathbb{R}^n$ とすれば、勝者の決定条件は以下で表される。

$$c = \arg \min_i \{ \|x - m_i\| \} \quad (1)$$

$d(x, m_i)$ には、ユークリッド距離やマンハッタン距離などが目的によって使い分けられる。勝者の参照ベクトルと、その近傍に位置するユニットは、入力されたデータに近づくように更新される。更新則は以下で表される。

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)] \quad (2)$$

$$h_{ci}(t) = \alpha(t) \cdot \exp \left(-\frac{d_{ci}^2}{2\sigma(t)^2} \right) \quad (3)$$

$$\sigma(t) = \sigma_f + (\sigma_i - \sigma_f) \cdot \exp \left(-\frac{t}{\tau} \right) \quad (4)$$

ここで t は学習時間、 $h_{ci}(t)$ は学習の条件付重み、 $\alpha(t)$ は学習係数であり、どちらも t の単調減少である。 $\exp$ は指数関数、 d_{ci} は勝者 c とユニット i 間の SOM のトポロジで定義されるユークリッド距離である。 σ_i は学習開始時の

近傍半径、 σ_f は学習終了時の近傍半径、 τ は時定数であり、大きいほど近傍関数 $h_{ci}(t)$ は緩やかに減少する。

SOM の性能は、マップのトポロジと参照ベクトルの数によって変化し、入力に応じてこれらを決定する必要がある。これは参照ベクトルの成長を伴わない全ての SOM アルゴリズムについて言えることである。

3 Growing Hierarchical SOM

Growing Hierarchical SOM (GHSOM) は Michael Dittenbach [1][2] が考案した成長型階層構造 SOM アルゴリズムである。GHSOM はまず最上位層である単一のユニットからなる第 0 層からスタートする。第 0 層の参照ベクトルを m_0 として全入力ベクトルの各要素それぞれの平均値を持っている。その m_0 を以下のように全入力ベクトルの偏差平均を求める。

$$mqe_0 = \frac{1}{d} \cdot \|x - m_0\| \quad (5)$$

d は全入力データ x の数である。得られた mqe_0 は第 1 層以降の層におけるユニットの成長およびマップの階層化を制御するパラメータとして利用する。

第 0 層では成長は行わず、第 1 層に 2×2 のマップを作成し、SOM を行う。学習終了後に成長規則を満たせば、1つのマップが完成し、満たさなければユニットを追加し SOM を行う。完成したマップから勝者ユニットを探索し、ユニットそれぞれの偏差平均を求める。その求めた値が階層化規則を満たせば下層にマップを追加する。最終的に階層化を満たすユニットがなくなるまで処理を行う。また、各層の各マップは独立したユニットから構成され、学習も独立で行われる。

成長判定を行う際に各マップ上で以下のパラメータを算出する。

$$MQE_m = \frac{1}{u} \cdot \sum_i mqe_i \quad (6)$$

u はユニットの数であるが、これはマップ中の全てのユニット数ではなく、入力データを担当する勝利ユニットの数である。下式の mqe_i は各ユニットの平均偏差をとる。また、各ユニットの平均偏差をとる d_i は各ユニットの BMU となった入力ベクトルの数である。

$$mqe_i = \frac{1}{d_i} \cdot \|x - m_i\| \quad (7)$$

3.1 成長規則

各マップで求めたパラメータを成長判定として以下の式を満たすまでユニットの挿入を行い成長させる。 T_m は閾値を表し、この値のが高いと成長しにくくなり、によって成長度合いを操作する。

$$MQE_m < T_m \cdot mqe_0 \quad (8)$$

上式を満たした場合、成長を行う。まず、各マップ中最も偏差の大きいユニットを検索し、見つけたユニットを error unit とし、そこから 1 近傍の中でユニット同士の 2 乗距離が一番離れているものを dissimilar unit とする。

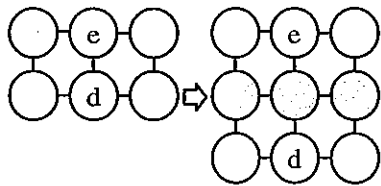


Fig. 1 ユニットの挿入図

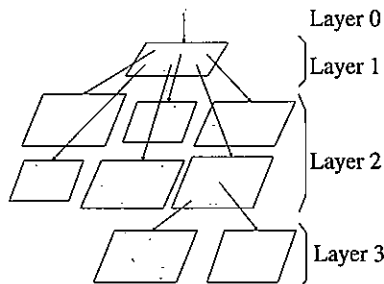


Fig. 2 階層化イメージ

Table 1 実験設定

	GHSOM	SOM
入力信号の総数	16	16
入力信号のベクトル長	17	17
ニューロン数	-	15 × 15
初期学習率	0.3	0.3
初期近傍数	5	5

求めた2つのユニットの間にそれぞれ向かい合ったユニット同士の間値のユニットを挿入する。

ユニットの挿入後, SOMによる学習を行い, 適切なマップサイズになるまで成長を繰り返す。

3.2 階層化規則

マップの成長停止条件を満たすと, 階層化規則により階層構造をもった SOM が生成される。

$$mqe_i > Tu \cdot mqe_0 \quad (9)$$

T_u は閾値を表し, この値の増減によって階層化の度合いを操作する。上式の条件を満たすと, 下層に 2×2 のマップが生成され, 最終的にどのマップも階層化を行わなくなった時点で終了する。SOM 階層化のイメージを Fig.2 に示す。

4 数値実験

4.1 実験概要

データ数 16, 次元数 16 の自己組織化マップの応用 [4] から得た animal データセットを用いて T_m, T_u を Table1 のように設定し, SOM と GHSOM の実行時間と, 比較を行った。

4.2 実験結果・考察

Fig.3 の結果から SOM の計算時間が少なくなることが確認できた。また, 学習結果は Fig.4 となった。GHSOM はマップサイズが少数から始まることで, マップの検索にかかる計算コストが少なくなり, 実行時間が小さくなっ

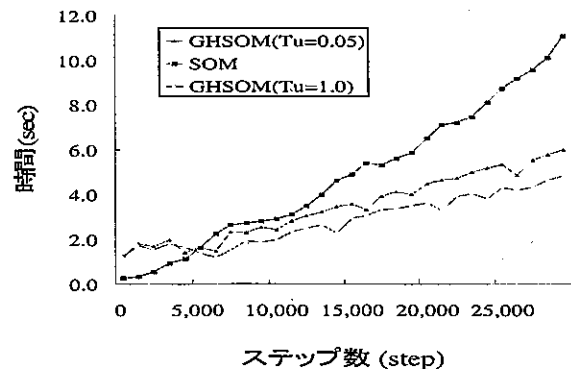


Fig. 3 ステップ数の変化による実行時間の推移

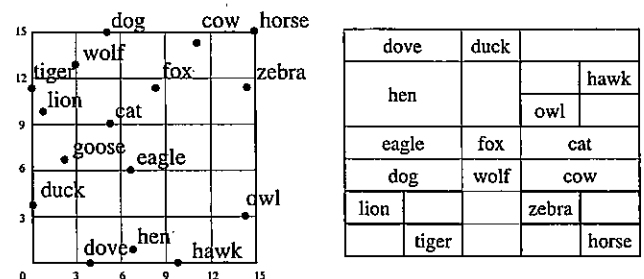


Fig. 4 30000step 学習後の SOM(左) と GHSOM(右)

ていると考えられる。これにより入力ベクトル自体の数と次数が多いものを用いた場合, 実行時間の差が大きくなると考えられる。

また, ステップ数が大きくなっても実行時間が大きくならない点を見るとステップ数が増えたことにより, ユニット間の距離が小さくなると同時に, ユニットと入力ベクトルの差が小さくなる。このことから成長条件と階層化条件を満たしにくくなり, 成長や階層化が抑えられて検索にかかる計算回数が減ったことが考えられる。

5 終わりに

本研究では, GHSOM と SOM の実行時間を比較し, GHSOM を用いることで SOM よりも結果を速く導くことが可能なことを示すことができた。今後は GHSOM の評価する指標を見つけ出し, GHSOM のパラメータの最適化を行うと考えている。

参考文献

- [1] M.Dittebach, D.Merkel and A.Rauber, "The Growing Hierarchical Self-Organizing Map" S.Amari and C.L.Giles and M.Gori and B.puri, editors, Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000), vol.6, pp.15-19, july 24-27, 2000, Como, Italy, IEEE Computer Society. 2000.
- [2] GHSOM - The Growing Hierarchical Self-Organizing Map - Homepage, <http://www.ifs.tuwien.ac.at/andi/ghsom/>
- [3] T. コホネン: 自己組織化マップ 改訂版, シュプリンガー・ジャパン (2005)
- [4] 徳高 平蔵: 自己組織化マップの応用, 海文堂出版 (1999)