

高階 SOM を用いた手書き文字認識

日大生産工 ○野田 翔太 日大生産工 山内 ゆかり

1 まえがき

近年スマートフォン端末等の爆発的な普及に伴い、手書き文字を認識する手法に対し注目が集まっている。

文字認識の手法の一つとして、弾性マッチング[1]がある。弾性マッチングとは、一方の画像をゴムのように変化させてもう一方の画像との一致を図る手法である。しかし、弾性マッチングによる認識においては、アルゴリズムの複雑さや計算量の多さが指摘される。そのため、自己組織化マップ (Self-Organizing Map: SOM)[2]を用いた文字認識手法[3]にも注目が集まっている。SOMは、1982年にKohonen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る。

SOMは簡単なアルゴリズムで実装でき、文字全体をとらえることが出来るため、形の変化にも柔軟に対応できる。それゆえ、他の手法と比べ認識率の向上や、計算時間の削減などにもつながる。しかし、SOMを用いる手法について問題点が挙げられる。SOMによる文字認識では、文字1つの認識において、1つの手書きデータをを用いると、新しい手書き文字の入力データを認識する際に、識別が正しく行われない可能性が高い。

そこで、従来型のSOMが作成するマップをさらに自己組織化し、写像することで解決を図る高階自己組織化マップ(SOM²)[4]が提案された。SOM²では手書きデータ「1」を識別する際、様々な形の「1」を入力データとして、自己組織化することで、「1」のマップを作成する。さらにそのマップ1つを入力データとして、自己組織化することで、それぞれの1において近い形状のものは近くに、形状が異なるものは遠くに配置されるマップが作成される。このことで、「1」を多面的にとらえ、文字認識における識別率を向上させることが出来る。図1にSOM²の概念図を示す。

以下、モジュールの中に入る従来のSOMを「子SOM」、子SOMをモジュールとして用いる上位のSOMを「親SOM」と示す。

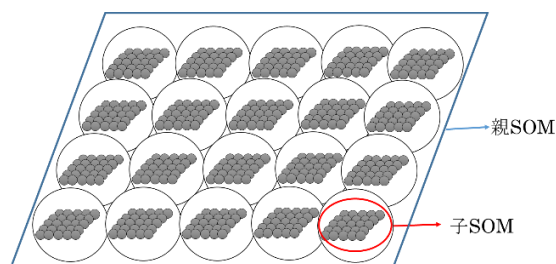


図1 SOM²の概念図

大谷らは、トポロジー拘束のないTopology Free SOM(TFSOM)を開発し、そのTFSOMを高階SOMに組み込みTFSOM×SOMを提案している[5]。この手法は、手書き文字の識別においてSOM²よりも識別率が高いことが報告されている。しかし、1文字に対して1つTFSOM×SOMでマップを構築するため、複数の文字に対応する場合、学習時だけでなく認識時にも膨大な計算コストがかかる。

本研究では、複数の文字に対応するTFSOM×SOMの適用方法を提案し、複合的に用いることで、メモリの使用量および計算コストを削減することを試みる。

2 従来研究

2.1 自己組織化マップ(SOM)

自己組織化マップ (SOM) [2]はKohonenによって提案されたニューラルネットワークの一種で、入力層と出力層の2層から構成されたものであり、高次元データを2次元平面状への非線形写像するデータ解析法である。SOMのアルゴリズムを以下に示す。

- A) すべての競合層のノード n の参照ベクトル y^n をランダムな値で初期化する。
- B) 1個の入力データからランダムに選んだものを入力ベクトル x_i とする。
- C) 式(1)により入力ベクトル x_i と競合層のノード n の参照ベクトル y^n とのユークリッド距離 E_i^n を求め、その距離が最小となる勝者ノード n_i^* を見つける。

$$E_i^n = \frac{1}{2} \|x_i - y^n\|^2 \quad (1)$$

$$n_i^* = \arg \min_n E_i^n \quad (2)$$

D) (3)を用いて、勝者ノード自身と周りの出力層 y^n を更新する。

$$y^n := (1 - \varepsilon)y^n + \varepsilon \frac{1}{B^n} \sum_{i=1}^I \beta_i^n x_i \quad (3)$$

$$B^n = \sum_{i=1}^I \beta_i^n \quad (4)$$

$$\beta_i^n = \exp \left[-\frac{1}{2\lambda^2} d^2(n_i^*, n) \right] \quad (5)$$

$$d(n_1, n_2) = \|\xi^{n_1} - \xi^{n_2}\| \quad (6)$$

ここで、 ε は1以下の正定数、 ξ^n は第 n ノードのマップ上の位置である。

2.2 高階自己組織化マップ(SOM²)

高階自己組織化マップSOM²は、従来型のSOMが作成するマップを一つのユニットとみなし、その要素を参照ベクトルとして、自己組織化を行う、階層的に適用された自己組織化マップである。例えば、下位のSOMはある文字を形成する点の位置座標を用いて、その文字がどのような点の集合で構成されているか、自己組織的に分布を学習する。上位のSOMは、下位のSOMである文字の分布の特徴に変換されたものを用いて、文字同士の類似や識別の目的で、文字空間を構成する。

なお、以降で示す従来研究及び本研究では、各文字の点の座標における次元を2次元で統一する。

2.3 Topology Free SOM(以下TFSOM)

大谷らは以下に示す二点の問題点を解決するベクトル量子化アルゴリズムTFSOMを開発した。

まず、従来のSOMの問題点は、マップを作るノード間の隣接関係が固定されているため、隣接するノードに引っ張られることで各ノードの学習に影響が出てしまう点にある。

一方ここで、隣接するノード間にエッジが存在しないニューラルガス(以下NG[6])を用いた場合の問題点は、各ノード間の隣接に規制がなく、それぞれ自由にふるまってしまうことであり、ノード間に位相空間が構成されない点である。

以下にSOMとNGの比較を示す。

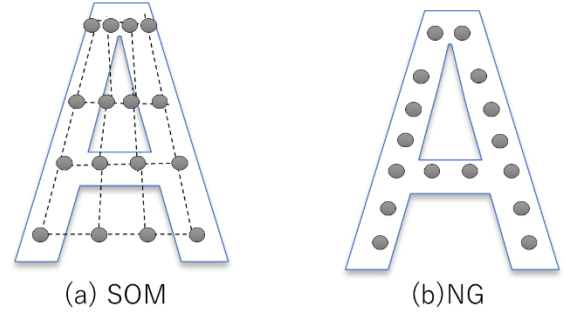


図2 SOMとNGの比較

ゆえに、TFSOMは次に示す二点の条件を満たす必要がある。一点目に、ノード間の距離がデータ空間の近傍関係にある程度維持する。つまり、各点にある程度の規制を持たせたい。二点目に、一方で各点が互いに近傍かどうかは事前に決めず、データを学習することで推定したい。SOMは一点目の条件のみ満たし、NGは二点目の条件のみ満たす。

そこで、大谷らは式(6)の従来のSOMにおけるノード間距離の計算部分を次のように修正したTFSOMを提案した。

$$d(n_1, n_2) = \text{RANK}_{n'=1, \dots, N} (E^{n_1, n_2}; E^{n_1, n'}) \quad (7)$$

$$E^{n_1, n_2} = \frac{1}{2} \|y^{n_1} - y^{n_2}\| \quad (8)$$

2.4 TFSOM×SOM

大谷らは次に、TFSOMを高階SOMに組み込んだ、TFSOM×SOMの構造を提案し、文字認識に適用した。図3に概念図を示す。

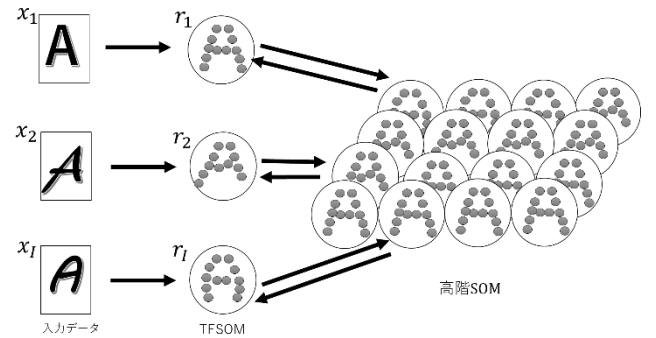


図3 TFSOM×SOMの概念図

I 個の文字データをそれぞれドットに分解したものが入力データ $\{x_1, x_2, \dots, x_I\}$ である。下層である TFSOM $\{r_1, r_2, \dots, r_I\}$ は、それぞれ対応する入力データ (文字) の形状を座標点の情報をを用いて学習する。その学習後の形状ベクトル r_i を入力ベクトルとし、高階SOMで形状の類似関係を通常のSOMのアルゴリズム

ムで学習する。ここで、高階 SOM の参照ベクトルを $u^m = \{z^{m1}, \dots, z^{mN}\} \in \mathbb{R}^{D \times N}$ とする。

以下に TFSOM×SOM のアルゴリズムを示す。

- A) I 個の学習用入力データを J 個のドットに分解し、IJ 個のデータベクトル x_{ij} を作る。
- B) TFSOM×SOM の各ユニットを乱数で初期化する。初期化が終了したら学習回数 $t=0$ とする。

初期化が終了したら、以下の学習ステップを収束するまで繰り返す。

C) 準備

各ループの開始にあたり TFSOM における平均ユニット間距離行列 D を求める。 D に関しては、式(7)、(8)を置き換えた以下の式から算出する。

$$E^{mn_1 m, n_2} = \frac{1}{2} \|z^{mn_1} - z^{mn_2}\| \quad (9)$$

$$d(n_1, n_2) =$$

$$\frac{1}{M} \sum_{m=1}^M \text{RANK}(E^{mn_1, mn_2}; E^{mn_1, mn'}) \quad (10)$$

TFSOM、高階 SOM それぞれの近傍半径を式(11)、(12)によって予め算出する。

$$\lambda_1(t) = (\lambda_1^{\max} - \lambda_1^{\min}) \exp\left[-\frac{t}{\tau_1}\right] + \lambda_1^{\min} \quad (11)$$

$$\lambda_2(t) = (\lambda_2^{\max} - \lambda_2^{\min}) \exp\left[-\frac{t}{\tau_2}\right] + \lambda_2^{\min} \quad (12)$$

ここで、 τ_1 、 τ_2 は縮小時定数である。さらに、前回ループにおける各ロードマーク点の勝率を求める。

$$p_i^n = \frac{\sum_j \delta(n_{ij}^*, n)}{J} \quad (13)$$

$$\bar{p}^n = \frac{\sum_i \sum_j \delta(n_{ij}^*, n)}{IJ} \quad (14)$$

尚、 $t=0$ の時は、すべて等確率 $p_i^n = \bar{p}^n = 1/N$ とする。

D) TFSOM の学習を行う。

従来の SOM と同様、勝者ノードを見つける。ただし式(15)、(16)を用いる。

$$E_{ij}^n = \frac{1}{2} \|x_{ij} - y_i^n\|^2 \quad (15)$$

$$n_{ij}^* = \arg \min_n E_{ij}^n \quad (16)$$

式(17) を用いて、勝者ノード自身と周りの出力層 y_i^n を更新する。

$$y_i^n := (1 - \varepsilon) y_i^n + \varepsilon \frac{1}{B_i^n} \sum_{j=1}^J \beta_{ij}^n x_{ij} \quad (17)$$

$$\beta_{ij}^n = \frac{\bar{p}_{ij}^n}{p_i^n} \exp\left[-\frac{1}{2\lambda(t)^2} d^2(n_{ij}^*, n)\right] \quad (18)$$

$$B_i^n = \sum_{j=1}^J \beta_{ij}^n \quad (19)$$

式(20)を用い、TFSOM の出力層の値により形状ベクトル r_i を更新する。

$$r_i = \bigoplus_{n=1}^N y_i^n \quad (20)$$

E) 高階 SOM の学習を行う。

式 (21) 、(22)を用いて、 r_i を入力ベクトルとして、高階 SOM における勝者ユニット m_i^* を決定する。

$$E_i^m = \frac{1}{2} \|r_i - u^m\|^2 \quad (21)$$

$$m_i^* = \arg \min_m E_i^m \quad (22)$$

式(23)を用い、高階 SOM における出力層の値を更新する。

$$u^m := \frac{1}{A^m} \sum_{i=1}^I \alpha_i^m r_i \quad (23)$$

$$\alpha_i^m = \exp\left[-\frac{1}{2\lambda^2} d^2(m_i^*, m)\right] \quad (24)$$

$$A^m = \sum_i \alpha_i^m \quad (25)$$

F) Copy Back

高階 SOM の学習結果を、TFSOM へフィードバックし、C)準備に戻る。

$$r_i := u^{m_i^*} \quad (26)$$

$$t = t + 1 \quad (27)$$

以上が TFSOM×SOM の学習アルゴリズムである。

TFSOM×SOM は、0～9 までの 10 種類の数字における手書き文字認識において、識別率 95%以上の結果が得られたと報告されている。

3 提案手法

大谷らが提案した TFSOM×SOM は非常に高い識別率を出している一方、1 文字につき 1 つの TFSOM×SOM を対応させ、学習を行うため、識別できる文字の種類を増やすと、膨大な計算コストがかかり、処理時間やメモリを多く使用するという問題がある。

そこで本研究では、複数の種類の文字を入力データとして、一つの TFSOM×SOM で学習し、より多くの種類の文字の識別を現実的な計算コストで行うことを目的とする。

この場合、一つの出力層において複数の文字を扱うため、ユニット毎に対応する文字のラベル付けを行う必要がある。また、すべての文字を一つの高階 SOM で学習させる場合、出力層のユニット数が膨大になるので、ある程度はグループ化し部分的に複数的高階 SOM で学習することが望ましいと考えられる。

そのため、第 1 段階の高階 SOM で、酷似している文字ごとに集合を作り、各集合に対してさらにもう一段階高階 SOM を行うこと（以下多段高階 SOM）で、識別率を下げることなく、課題を解決できるのではないかと考えた。

4 実験環境

4.1 予備実験

本研究では、まず従来手法の TFSOM×SOM の文字認識における識別率、計算にかかった時間、総計算コスト、メモリの使用量を出力する必要がある。取得したデータと提案手法の有用性を検討していく。

4.2 複数文字の同時形状空間推定

本研究における実験用の入力データは、自分で手書きした「1」から「9」の数字である。また各数字において 9 種類の手書き文字データを用意した。

まず結果として、計算コストの削減、メモリの使用量の削減を実現するため、複数の文字を同時に読み込むことで、様々な文字の形状表現マップを作成する。その際に、同じ文字は近くに、形が大きく違う文字は遠くに配置されるマップが完成していることを確認する。

4.3 多段高階 SOM

提案手法に記載の通り、ユニット間のユークリッド距離を計算した際に、ユークリッド距離が非常に近く、似ている文字同士で集合を作る。

本研究では、「1」と「7」と「2」、「5」と「6」、「3」と「8」、「4」と「9」のような複数の集合ができると想定している。それぞれのグループを入力データとして、グループ毎に高階 SOM で学習を行う。

得られたそれぞれのマップを用いて、文字認識を行った際の識別率、計算にかかった時間、総計算コスト、メモリの使用量を出力し、予備実験において求めた結果と比較し、有用性を検討する。

5 まとめ

本研究では、複数の文字に対応する多段高階 SOM を用いた文字認識の手法を提案した。従来研究で 1 文字毎に構成していた TFSOM×SOM を、似ている文字をグループ化し、複合的に用いることで、メモリの使用量および計算コストを削減する効果が得られると考えられる。

複数の文字を同時に学習する過程で TFSOM×SOM を 1 回、続いて似ている文字同士で集合を作り、それぞれに対し TFSOM×SOM を 1 回ずつ実行するため、従来研究のおおよそ半数の高階 SOM で、学習および識別が行える。この手法の有効性が確認出来れば、数字のみでない文字認識に幅広く応用が可能になると考えられる。

「参考文献」

- 1) S.Omachi, S.Megawa, and H.Aso, "Decorative Character Recognition by Graph Matching", IEICE Trans. Inf. Syst., Vol.ED90-D, No.10, 2007.
- 2) T.Kohonen, "Self-Organizing Maps", Springer, vol.30, 1995.
- 3) 増原士, 折居英章, 河野英昭, 前田博, 「自己組織化マップを用いた変形を許容する文字認識法の提案」, 電気関係学会九州支部連合大会, pp.269-270, 2010.
- 4) 古川徹夫, 「SOM の SOM : SOM 集合をマップする SOM」, 知能と情報, Vol.19, No.6, pp.618-626, 2007.
- 5) 大谷誠, 古川徹夫, 「高階 SOM による形状表現マップの自己組織化—トポロジー拘束のない形状空間法—」, 知能と情報, Vol.25, No.2, pp.701-720, 2013.
- 6) T.M.Martinetz, S.G.Berkovich, K.J.Schulte, "Neural Gas: Network for Vector Quantization and its Application to Time-Series Prediction", IEEE Trans. of NEURAL NETWORKS, Vol.4, pp.558-567, 1993.