

二種類のニューロンを含むニューラルガスとその振る舞い

神田 恵子[†] 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: †{kanda,haruna,nishio}@ee.tokushima-u.ac.jp

あらまし 本研究では、二種類のニューロンを含むニューラルガス (Neural Gas Containing Two Kinds of Neurons: TN-NG) と、二種類の更新関数を含むニューラルガス (Neural Gas Containing Two Kinds of Update Functions: TF-NG) の2つの手法を提案する。TN-NG では、全てのニューロンがそれぞれ個々の性質に従って学習する。一方、TF-NG では、全てのニューロンが勝者ニューロンの性質に従って学習する。TN-NG および TF-NG の振る舞いを調査し、従来のニューラルガスよりも効果的な学習結果を得ることを確認する。

キーワード ニューラルガス (NG), 特徴抽出, ノイズ除去

Neural Gas Containing Two Kinds of Neurons and its Behaviors

Keiko KANDA[†], Haruna MATSUSHITA[†], and Yoshifumi NISHIO[†]

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

Abstract In this study, we propose two new Neural Gas algorithms. One method is the Neural Gas Containing Two Kinds of Neurons (called TN-NG). In TN-NG, all neurons learn according to own character at each learning. The other method is the Neural Gas Containing Two Kinds of Update Functions (called TF-NG). In TF-NG, all neurons learn according to winner's character at each learning. The behavior of TN-NG and TF-NG is investigated with computer simulation. We confirm that TN-NG and TF-NG can obtain the more effective learning results than the conventional Neural Gas.

Key words neural gas, feature extraction, noise reduction

1. はじめに

近年、私たちは画像や統計などの多次元量かつ線形分離不可能な膨大な量のデータを扱うことから、比較的小さい計算量で良好な解を得ることが求められている。そこで、脳を簡略化したモデルであるといわれている様々なニューラルネットワークが注目されている。ニューラルネットワークは、教師信号の入力によって問題の最適化を行う教師あり学習と、教師信号を必要としない教師なし学習に分けられ、パターン認識やデータマイニングなど、さまざまな分野に応用されている。そのなかで、画像認識やクラスタリング手法の1つとして注目されているものに、ニューラルガス (Neural Gas: NG) [1] がある。NG は1991年にMartinetzとSchuldenによって提案された順位学習型教師なしニューラルネットワークの1つで、生物学に端を発するモデルであるといわれている。NGの学習過程では、入力ベクトルに近い重みベクトルを持つニューロン順に順位付けを行い、その順位に従ってニューロンの重みベクトルの更新を繰り返すことで学習が行われる。しかし、入力の形や密度によっては、離れた複数のクラスタを学習することや、クラスタの端まで分布状態を反映した学習が困難であるといった問題点があ

る。また、全てのニューロンが全ての入力に反応してしまうため、ノイズを含むデータに対しては正確な学習を行えないことも問題点として挙げられる。

その一方で、実社会においては、例えばゾウは遅い心拍速度、ネズミは早い心拍速度を持ち、それぞれ時間に対する感覚が異なっていると言われているように、それぞれが個々に個性を持ちながら行動している。このように、実社会では様々な異なる特徴や感覚を持ったものが共存し、様々な得意不得意といったそれぞれの特徴を補い合っている。また、同じ個体のあいだでも、年齢やそのときの状況などに応じて、感覚は様々なに変化するのが常である。しかし、ニューラルガスネットワークにおいては、全てのニューロンが常に同じ性質を持ち、同じ方式に従って学習している。そこで本研究では、二種類のニューロンを含むニューラルガス (Neural Gas Containing Two Kinds of Neurons: TN-NG) と、二種類の更新関数を含むニューラルガス (Neural Gas Containing Two Kinds of Update Functions: TF-NG) の2つの手法を提案する。TN-NGでは、近傍順位関数の異なる二種類のニューロンが存在し、それぞれが個々の性質に従って学習する。一方TF-NGでは、全てのニューロンの学習率が、勝者ニューロンの性質に従って変化する。つまり、

TN-NG は二種類のニューロンが個々の特徴を補い合う学習を行う手法であり、TF-NG はその状況に応じて個々のニューロンの能力が変化する手法だと言える。

これらの手法をシミュレーションすることで、TN-NG および TF-NG の振る舞いを調査する。また、それぞれの学習結果の特徴から、TN-NG はノイズリダクションへ、TF-NG は特徴抽出への応用を行い、従来のニューラルガスよりも効果的な学習結果を得ることを確認する。

2. ニューラルガス (NG)

従来の NG の学習アルゴリズムについて説明する。NG ネットワークには M 個のニューロンの集合 $A = \{c_1, c_2, \dots, c_M\}$ が存在し、各ニューロン c_i ($i = 1, 2, \dots, M$) は d 次元の重みベクトル $\mathbf{w}_i = (w_{i1}, w_{i2}, \dots, w_{id})$ を持っている。重みベクトルの初期値は入力空間全体にランダムに配置する。

(NG1) d 次元の入力ベクトル $\mathbf{x}_j = (x_{j1}, x_{j2}, \dots, x_{jd})$ ($j = 1, 2, \dots, N$) を 1 つ選び、全てのニューロンに入力する。

(NG2) 入力ベクトル \mathbf{x}_j と全ての重みベクトル \mathbf{w}_i との距離 d_{ji} を計算する。

$$d_{ji} = \|\mathbf{x}_j - \mathbf{w}_i\| \quad (1)$$

ここで $\|\cdot\|$ はユークリッド距離を表す。

(NG3) 距離 d_{ji} が短い順に、それぞれのニューロン c_i に順位 $k = 0, \dots, (M-1)$ を与える。ここで k_i はニューロン c_i の順位とする。順位 0 は \mathbf{x}_j に最も近いニューロン (勝者ニューロン) を、順位 $(M-1)$ は \mathbf{x}_j に最も遠いニューロンを意味する。

(NG4) ニューロンの重みベクトルを更新する。

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + E(t)h_{\lambda}(t, k)(\mathbf{x}_j - \mathbf{w}_i(t)) \quad (2)$$

t は学習回数、 $E(t)$ は学習率、 $h_{\lambda}(t, k)$ は近傍順位関数である。

$$h_{\lambda}(t, k) = \exp(-k_i/\lambda(t)) \quad (3)$$

$\lambda(t)$ は近傍順位関数の収束速度を制御する関数であり、 $E(t)$ と $\lambda(t)$ は時間と共に減少する。

$$E(t) = E_0 (E_0/E_f)^{\frac{t}{T}}, \quad \lambda(t) = \lambda_0 (\lambda_0/\lambda_f)^{\frac{t}{T}} \quad (4)$$

ここで E_0 と E_f は $E(t)$ の、 λ_0 と λ_f は $\lambda(t)$ の初期値と最終値をそれぞれ表し、 T は最大学習回数である。

(NG5) $t < T$ を満たす間、全入力データに対して (NG1) から (NG4) を繰り返す。

3. 提案手法

3.1 二種類のニューロンを含むニューラルガス (TN-NG)

1 つ目の提案手法である、TN-NG について説明する。TN-NG では、ネットワークに存在する M 個のニューロンのうち、 m_1 個のニューロンを集合 A_1 に、 m_2 個のニューロンを集合 A_2 にランダムに分類する ($M = m_1 + m_2$)。 A_1 と A_2 に属するニューロンに対し、近傍順位関数の収束速度を変えることで、 A_1 に属するニューロン群は急速に収束し、 A_2 に属するニューロン群はゆっくりと収束するというように、それぞれに異なる特徴を持たせる。重みベクトルの初期値は、従来の NG と同様に、入力空間全体に広がるようにランダムに設定する。

(TN-NG1) d 次元の入力ベクトル \mathbf{x}_j を 1 つ選び、全てのニューロンに入力する。

(TN-NG2) 入力ベクトル \mathbf{x}_j と、全ての重みベクトル \mathbf{w}_i の間の距離 d_{ji} を、式 (1) に従って計算する。

(TN-NG3) それぞれのニューロン c_i を、距離 d_{ji} の短い順に順位付け $k = 0, \dots, (M-1)$ を行う。

(TN-NG4) ニューロンの重みベクトルを更新する。

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + E(t)h_{\lambda_{TK}}(t)(\mathbf{x}_j - \mathbf{w}_i(t)) \quad (5)$$

$h_{\lambda_{TK}}(t)$ は TN-NG における近傍順位関数である。

$$h_{\lambda_{TK}}(t, k) = \exp(-k_i/\lambda_{TK}(t)) \quad (6)$$

このとき、次式に従い、 A_1 に属するニューロンと A_2 に属するニューロンでは異なった $\lambda_{TK}(t)$ を用いる。

$$\lambda_{TK}(t) = \begin{cases} \lambda_0 \left(\frac{\lambda_f}{\lambda_0}\right)^{\frac{t}{T}}, & c_i \in A_1 \\ (p + \lambda_0 - \lambda_f) \left(\frac{p}{p + \lambda_0 - \lambda_f}\right)^{\frac{t}{T}} \\ + \lambda_f - p, & c_i \in A_2 \end{cases} \quad (7)$$

ここで、 p は λ_{TK} の収束速度を調整するパラメータであり、 $p < \lambda_f$ とする。これにより、 A_1 に属するニューロンは急速に収束し、 A_2 に属するニューロンはゆっくりと収束するという特徴を持つ。

(TN-NG5) $t < T$ を満たす間、全入力データに対して (TN-NG1) から (TN-NG4) を繰り返す。

3.2 二種類の更新関数を含むニューラルガス (TF-NG)

2 つ目の提案手法である、TF-NG について説明する。TF-NG では、ネットワークに存在する M 個のニューロンのうち、 m_1 個のニューロンを集合 A_1 に、 m_2 個のニューロンを集合 A_2 にランダムに分類する ($M = m_1 + m_2$)。学習率の収束速度を変化させることで、 A_1 に属するニューロンが勝者ニューロンになった場合は全てのニューロンが急速に収束し、 A_2 に属するニューロンが勝者ニューロンになった場合はゆっくりと収束するというような特徴を持たせる。重みベクトルの初期値は、従来の NG と同様に、入力空間全体に広がるようにランダムに設定する。

(TF-NG1) d 次元の入力ベクトル \mathbf{x}_j を 1 つ選び、全てのニューロンに入力する。

(TF-NG2) 入力ベクトル \mathbf{x}_j と、全ての重みベクトル \mathbf{w}_i の間の距離 d_{ji} を、式 (1) に従って計算する。

(TF-NG3) それぞれのニューロン c_i を、距離 d_{ji} の短い順に順位付け $k = 0, \dots, (M-1)$ を行う。ここで、 $k_i = 0$ 、すなわち \mathbf{x}_j に最も近いニューロンを、勝者ニューロン c とする。

(TF-NG4) ニューロンの重みベクトルを更新する。

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + E_F(t)h_{\lambda}(t)(\mathbf{x}_j - \mathbf{w}_i(t)) \quad (8)$$

このとき、次式に従い、勝者ニューロン c が A_1 と A_2 のどちらに属するかで、全てのニューロンの学習率 $E_F(t)$ を決定する。

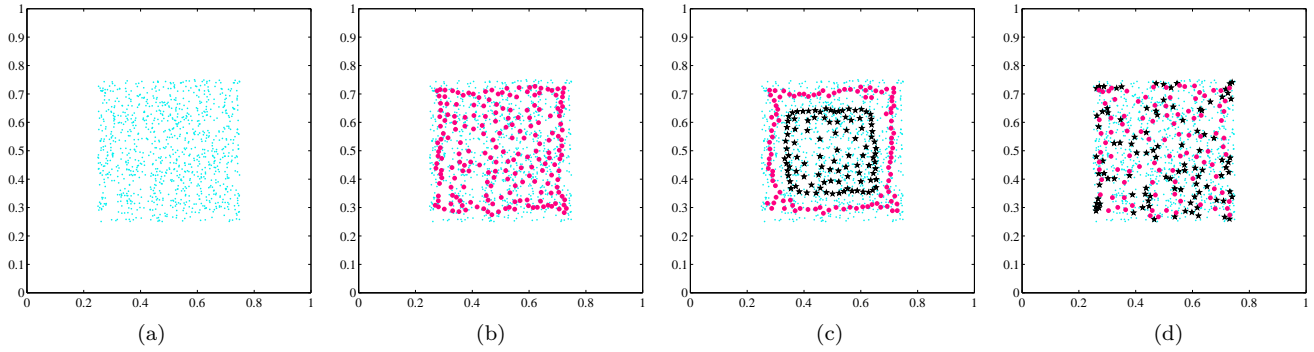


図 1 2次元データに対するシミュレーション．(a) 入力データ．(b) 従来の NG の学習結果．(c) TN-NG の学習結果．●は A_1 ，★は A_2 に属するニューロンを表す．(d) TF-NG の学習結果．●は A_1 ，★は A_2 に属するニューロンを表す．

$$E_F(t) = \begin{cases} E_0 \left(\frac{E_f}{E_0} \right)^{\frac{t}{T}}, & c \in A_1 \\ (q + E_0 - E_f) \left(\frac{q}{q + E_0 - E_f} \right)^{\frac{t}{T}} \\ + E_f - q, & c \in A_2 \end{cases} \quad (9)$$

ここで、 q は E_F の収束速度を調整するパラメータであり、 $p < E_f$ とする．これにより、 E_F の値が勝者ニューロンの性質に従って学習ごとに毎回変化する．

(TF-NG5) $t < T$ を満たす間、全入力データに対して (TF-NG1) から (TF-NG4) を繰り返す．

4. シミュレーション結果

まず、図 1(a) のような二次元の入力データに対して、従来の NG、TN-NG および TF-NG のシミュレーションを行う．入力データ数は 1000 個で、全てのデータはランダムに並べ替えている．従来の NG は $M = 200$ 個のニューロンを持ち、TN-NG および TF-NG では $m_1 = m_2 = 100$ 個のニューロンを用いる．学習は全ての入力データに対して 5 回繰り返す ($T = 5000$)．学習パラメータは、以下のように設定する．

$$(NG) \quad E_0 = 0.7, E_f = 0.01, \lambda_0 = 70, \lambda_f = 0.2$$

$$(TN-NG) \quad E_0 = 0.7, E_f = 0.01, \lambda_0 = 70, \lambda_f = 0.2, p = 50$$

$$(TF-NG) \quad E_0 = 0.7, E_f = 0.01, \lambda_0 = 70, \lambda_f = 0.2, q = 50$$

NG と TN-NG、TF-NG の学習結果を、それぞれ図 1 (b), (c), (d) に示す．これを見ると、TN-NG では、二種類のニューロンがそれぞれクラスタの中心部分と外側の部分に分かれ、ニューロンの種類によって領域を持っていることがわかる．これは、二種類のニューロンでは、入力に対する集束力が異なるからである．また、従来の NG と TF-NG では、学習結果の視覚的違いが少なく判断が難しいため、量子化誤差を用いてそれぞれの学習精度を評価する．

ここで、量子化誤差 Q_e とは、それぞれの入力ベクトルと勝者ニューロンとの距離の平均を求めたものであり、 Q_e の値が小さいほど学習精度がよいと言える [3]．

$$Q_e = \frac{1}{N} \sum_{j=1}^N \|x_j - \bar{w}_j\| \quad (10)$$

\bar{w}_j は、 x_j に対する勝者ニューロンの重みベクトルである．

表 1 2次元データに対する量子化誤差 Q_e

	従来の NG	TN-NG	TF-NG
Q_e	0.0146	0.0155	0.0138

表 1 に示されるように、量子化誤差の点においては、従来の NG に比べ TF-NG は優れた値を出しており、相対誤差で 5.47% の改善がみられるが、TN-NG では逆に学習精度が劣っている．

5. TN-NG のノイズリダクションへの応用

図 1 に対するシミュレーション結果より、二種類のニューロンでクラスタの中心への集束力が異なることを利用して、TN-NG をノイズリダクションへ応用する．

図 2(a) に示す、ノイズを含む入力データに対してシミュレーションを行う．全データ数は 1000 で、そのうちクラスタの入力データ数は 700 個、ノイズの入力データ数は 300 個であり、全てのデータはランダムに並べ替えている．学習は全ての入力データに対して 5 回繰り返す ($T = 5000$)．学習パラメータは、図 1 に対するシミュレーションと同様の値を使用する．

NG と TN-NG の学習結果を、それぞれ図 2(b), (c) に示す．これを見ると、従来の NG では全てのニューロンがノイズの影響を受けながら学習してしまっていることがわかる．対して TN-NG は、入力に対する集束力が強い A_2 のニューロンがクラスタを、集束力が弱く A_2 のニューロンによってクラスタから追い出された A_1 のニューロンがノイズを学習している．

図 2(d) は、図 2(c) の TN-NG の学習結果より、各入力データの勝者ニューロンを調べ、その勝者ニューロンが A_2 に属する入力データのみを抽出した結果である．これを見ると、入力データからノイズが除去され、クラスタのみが抽出されていることがわかる．よって、TN-NG はノイズリダクションに利用可能であると言える．

6. TF-NG の特徴抽出への応用

図 3(a) に示されるような、2次元のターゲットデータ [2] を考える．図 1 に対するシミュレーションにおいて、TF-NG が量子化誤差を減少できたことから、TF-NG を特徴抽出へ応用する．このデータは中心の円とその周りを囲む輪、そして四隅の点の計 6 つのクラスタからなっているが、各クラスタは互いに密度や大きさがそれぞれ異なるため、特徴抽出が困難な問題を含んでいる．データ数 N は 770 個であり、学習条件は図 1 の場合と同じとする．

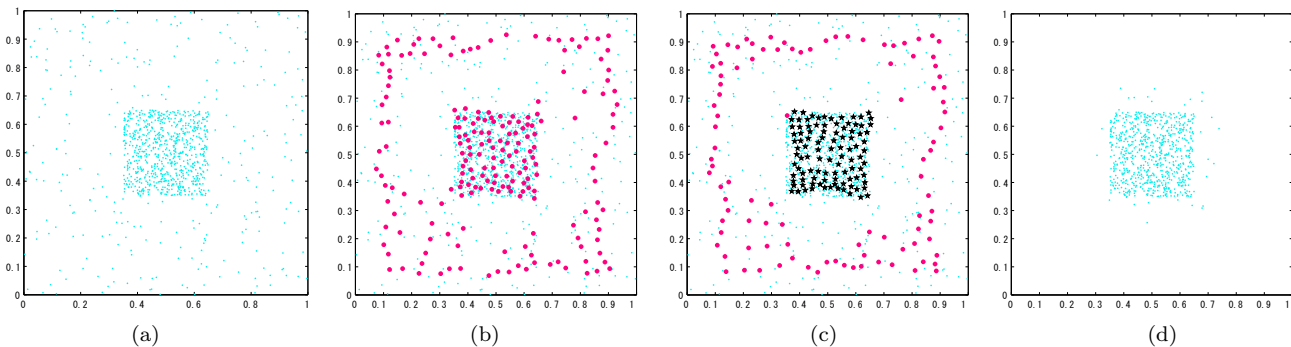


図2 ノイズを含むデータに対するシミュレーション . (a) 入力データ . (b) 従来の NG の学習結果 . (c) TN-NG の学習結果 . (d) TN-NG によって抽出されたクラスタ .

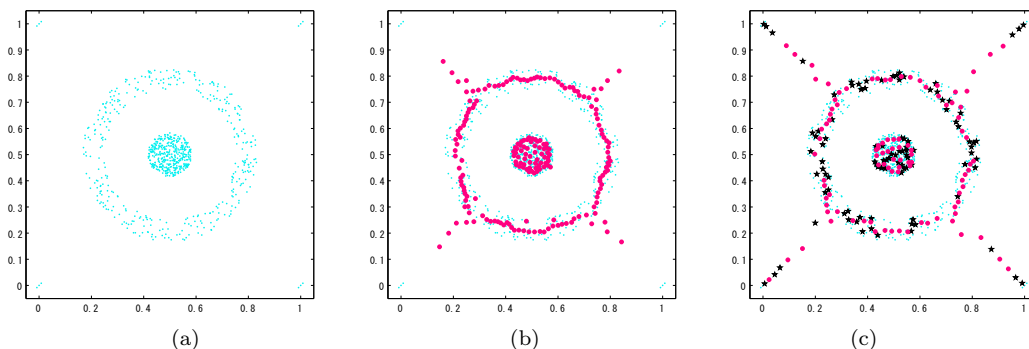


図3 ターゲットデータに対するシミュレーション . (a) 入力データ . (b) 従来の NG の学習結果 . (c) TF-NG の学習結果 .

表2 ターゲットデータ Q_e に対する量子化誤差

	従来の NG	TF-NG
Q_e	0.0162	0.0128

従来の NG と TF-NG のシミュレーション結果を図 3(b), (c) に示す . これを見ると , NG の結果では四隅のクラスタまでニューロンが学習し切れていないのに対し , TF-NG の結果では同じ学習回数でニューロンが四隅のクラスタまで達しているのがわかる . これは , 全てのニューロンの学習率関数が学習ごとに変化することにより , 入力への集束力と安定性の両方の特徴を持つからである .

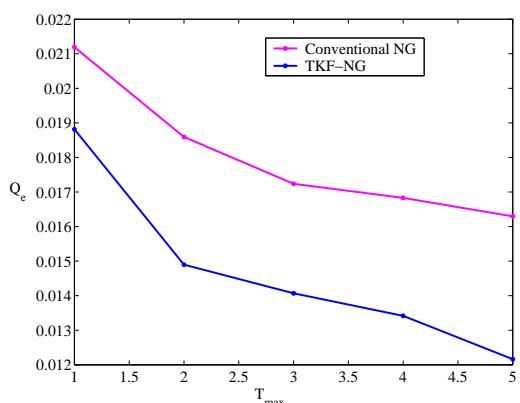


図4 入力の繰り返し回数 T_{\max} を変化させたときの量子化誤差 .

表2 は , 従来の NG と TF-NG のターゲットデータの学習結果に対する量子化誤差を計算したものである . これを見ると , TF-NG のほうが NG よりも優れた値を持ち , 相対誤差で

21.0% もの改善が見られることがわかる .

また , 図4 は入力データを繰り返す回数 T_{\max} を変化させ , シミュレーションを行ったときの , 量子化誤差をグラフにしたものである ($T = N \times T_{\max}$) . これを見ると , いずれの回数でも TF-NG のほうが小さい , すなわち優れた値が出ていることがわかる . また , 従来の NG に対し入力を5回繰り返したときよりも , TF-NG に対し1回しか入力しなかったときの方が , 良い結果が得られている . したがって , TF-NG は NG に比べて , 収束が早く , 少ない学習でよい結果が得られることから , 特徴抽出に有効であると言える .

7. ま と め

本研究では , 二種類のニューロンを含むニューラルガス (TN-NG) および二種類の更新関数を含むニューラルガス (TF-NG) を提案した . TN-NG では , それぞれのニューロンが個々に二種類どちらかの近傍順位関数に従って学習し , TF-NG では , 毎回の学習ごとに全てのニューロンが勝者ニューロンの属する学習率に従って学習する . また , シミュレーションを行うことで , TN-NG および TKF-NT の振る舞いを調査し , その結果から , TN-NG はノイズリダクション , TF-NG は特徴抽出への有効性があることを確認した .

文 献

- [1] T. M. Martinez and K. J. Schulten, "A "neural-gas" network learns topologies. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors," *Artificial Neural Networks*, pp. 397–402, 1991.
- [2] A. Ultsch, "Clustering with SOM: U*C", *Proc. Workshop on Self-Organizing Maps*, pp. 75–82, 2005.
- [3] T. Kohonen, *Self-organizing Maps*, 2nd ed., Berlin, Springer, 1995.