

複雑ネットワークの特徴量を用いた SOINN アルゴリズムの改善

日大生産工（院） ○高萩 悠 日大生産工 山内 ゆかり

1. まえがき

実世界のデータは分布の形状が変化するなど非定常な場合がある。それらの非定常性に対処するために追加学習の機能が必要となる。山崎らは自己増殖型ニューラルネットワーク (Self-Organizing Incremental Neural Network: SOINN)[1]を提案した。この手法は自己組織化マップ (Self-Organizing Map: SOM)[2] と Growing Neural Gas (GNG)[3] に着想を得て構築された教師なし学習手法の1種である。動的にネットワーク構造を構築するため、ネットワークの事前定義が必要なく、追加の学習が可能である。また、ノイズへの耐性も獲得している。しかし、生成されるネットワークがデータの入力順などに依存しているため、同じデータから学習を行っても別のネットワークが構築される。そのため、毎回同様の結果が求められるとは限らないという問題がある。

一方で柳井らはノード順位による選択を用いたスケールフリーネットワークモデル[4]を提案している。これは成長戦略をとらずにスケールフリーネットワークを生成するモデルである。

本研究では複雑ネットワークの構成モデルからノードやエッジの再構成手法を取り入れることでSOINNのネットワーク構成を安定させるアルゴリズムを提案する。

2. 従来研究

2.1. 自己増殖型ニューラルネットワーク

自己増殖型ニューラルネットワーク (Self-Organizing Incremental Neural Network : SOINN)とは、1層構造をなし、ノードの削除周期である λ と、ノードとノードをつなぐエッジを削除する指標である age_{max} の二つのパラメータを持つ学習アルゴリズムである。まずは基本となるAdjusted SOINNのアルゴリズムについて説明する。

Step1: 式(1)に基づき、入力に対して既存のノードとユークリッド距離から勝者ノードを2つ決定する。それぞれを第一勝者、第二勝者とする。 Win_1 、 Win_2 はそれぞれ第一勝者、第二勝者ノードを表し、 X_t は t 回目の入力値を表す。 i はノード番号を表し、 A はSOINN空間上の全ノード

の集合を表し、 W_i は i 番目のノードの重みを表す。第二勝者を求めるときは第一勝者は除いたSOINN空間上で入力値とのユークリッド距離が最も小さいものを求める。

$$Win_{1,2} = \min_{i \in A \setminus Win_1} |W_i - X_t| \quad (1)$$

Step2: 類似度閾値 T_i を超えるかを判断する。類似度閾値の計算は勝者ノードがエッジを持つ場合と持たない場合によってそれぞれ違うものを扱う。勝者ノードがエッジを持たない場合を式(2)に、持つ場合を式(3)に示す。 C は勝者ノードのノード番号を表し、 N_i は i 番目のノードに連結しているノード集合を表す。

$$T_i = \min_{c \in A \setminus \{i\}} |W_i - W_c| \quad (2)$$

$$T_i = \max_{c \in N_i} |W_i - W_c| \quad (3)$$

式(2)及び式(3)で求めた類似度閾値のいずれかが入力値との距離を超える場合はStep3に進み、そうでない場合はStep4へ進む。式(2)では自身を含まないすべてのノードから自身までのユークリッド距離を類似度閾値とし、式(3)では自身とエッジによってつながっているノード群とのユークリッド距離を類似度閾値としている。

Step3: 入力値を新規ノードとしてノードを追加する。その後Step10に進む。

Step4: 第一勝者ノードと第二勝者ノードの間にエッジの接続があるかを確認する。エッジを持つ場合はStep5に進み、エッジを持たない場合はStep6に進む。

Step5: 第一勝者と第二勝者の間のエッジの年齢を0に初期化する。その後Step7に進む。

Step6: 第一勝者と第二勝者の間にエッジを挿入し、そのエッジの年齢を0に初期化する。

Step7: 第一勝者ノードに繋がるエッジの年齢をインクリメントする。

Step8: 第一勝者ノードは式(4)で重みを更新し、第一勝者とエッジで繋がっているノードを式(5)で更新する。式(4)及び式(5)で使われる t_i とはノード i が第一または第二勝者に選ばれた回

数である。 s_1 は第一勝者ノードのノード番号である。

$$\Delta W_{s_1} = \frac{1}{t_{s_1}} (X_t - W_{s_1}) \quad (4)$$

$$\Delta W_i = \frac{1}{100t_i} (X_t - W_i) \quad (i \in N_{s_1}) \quad (5)$$

Step9: エッジの年齢を確認し、エッジの年齢が age_{max} 以上のエッジは削除する。エッジの削除処理後に孤立ノードとなったノードはエッジの削除処理後に削除される。

Step10: ノードの入力回数が λ の倍数回の場合ノイズノードの削除処理を行う。ノイズノードの削除処理は次数が1以下のノードに対して行われる。次数1以下のノードを削除し、また削除ノードに繋がっていたエッジも削除する。

Step11: 学習が終了していない場合、Step1に戻り新たな入力を行う。

2.2. ランク選択モデル

ランク選択モデルとは成長原理を用いず、スケールフリーネットワークを生成することができるノンパラメトリックなネットワーク生成モデルである。具体的なアルゴリズムについて説明する。

Step A: n 個のノードを用意し、各ノードに適合度を任意に割り当てる。

Step B: ランダムに s 個のノードを選択し、 s 個のノードの中で最も適合度の高いノードを選び、これを $node_1$ とする。

Step C: もう一度ランダムに s 個のノードを選択し、その s 個のノードの中で最も適合度の高いノードを選び、これを $node_2$ とする。この時 $node_1$ と $node_2$ は一致しないようにする。

Step D: $node_1$ と $node_2$ にリンクを生成する。すでに $node_1$ と $node_2$ の間にリンクが存在する場合はリンクを新たに生成しない。

Step E: 学習終了まで繰り返す。

ランク選択モデルはネットワーク生成の過程でノード数が増加せず一定である。

3. 提案手法

3.1. 提案手法のアルゴリズム

SOINN はそのアルゴリズムの性質上、SOINN 空間上に存在するノードのうち疎なデータ領域に存在しているノードをノイズノードとして削除する。それらのノードのうち重要であると考えられるノードを削除しないためにノイズノードの削除処理前にランク選択モデルによるエッジの追加を行う。提案手法のアルゴリズムを従来手法の各 Step に則って説明する。

基本的には SOINN アルゴリズムの Step10 であるノイズノードの削除処理前に提案手法の処理を加える。

Step10'-1: 提案手法ではノイズノードの削除前に SOINN 空間上のノードに適合度を割り当てる。適合度は SOINN 空間上の頂点間距離や、入力値との平均距離などが挙げられる。

Step10'-2: 次に削除対象のノードのうち最も適合度の高いノードを求める。

Step10'-3: 最も適合度の高いノードとその近傍 2 ノードにエッジを繋げる。その際新しく作るエッジの年齢は 1 とし、年齢のインクリメントは行わないものとする。

Step10'-4: 適合度の最も高いノードは削除対象から除外する。

Step10'-5: 通常の SOINN アルゴリズムと同様に次数 1 以下のノードとそのノードに繋がるエッジを削除する。

Step10'-6: SOINN 空間上に残ったノードの第一勝者になった回数を 1 にリセットする。

また従来の SOINN アルゴリズムでは Step8 によって第一勝者ノードの近傍ノードも重みの更新を行っていたが、提案手法では近傍ノードの更新を行わず第一勝者ノードの更新のみ行う。そのため Step8 ではなく Step8'で重みの更新を行う。

Step8': 提案手法におけるノードの重みの更新は式(6)で行う。 $t_{s_1}^1$ は第一勝者ノードが第一勝者に選ばれた回数である。

$$\Delta W_{s_1} = \frac{1}{t_{s_1}^1} (X_t - W_{s_1}) \quad (6)$$

3.2. 適合度について

提案手法ではいくつかの特微量で適合度を与えた。まずは頂点間距離を適合度にする場合について説明する。

$$F_i = \sum_{c \in A \setminus \{i\}} |W_i - W_c| \quad (7)$$

式(7)は頂点間距離を適合度にした際の式である。 F_i はノード i の適合度を表す。本来ネットワーク理論における頂点間距離というのは各ノード間におけるエッジの最短経路である。しかし SOINN 空間においてエッジで繋がれた最短経路を用いた場合、削除対象になるノードは次数が1以下であるため本来の意味での頂点間距離は不適切になる。そこで本研究では SOINN 空

間において頂点間距離をユークリッド距離で求め、それを適合度として用いる。

次に入力値との平均距離を適合度にする場合について説明する。式(8)、式(9)、式(10)は入力値との距離の平均を適合度にするための式である。

$$IV_i \leftarrow \sum_{t, i=s_t} |W_{s_t} - X_t| \quad (8)$$

$$F_i = \frac{IV_i}{t_i^1} \quad (9)$$

$$IV_i \leftarrow F_i \quad (10)$$

式(8)において IV_i はノード i が第一勝者ノードに選ばれた際のノード i と入力値 X_t とのユークリッド距離の差の総和を表す。式(9)より適合度 F_i は IV_i をそのノードが第一勝者に選ばれた回数 t_i^1 で割った値となる、勝者に選ばれた際における入力値とノードの重みの差の平均値を割り当てる。入力値との距離の平均を適合度とした場合はStep10⁻⁶にて t_i^1 を1にリセットすると同時に式(10)により IV_i を上書きする。ここで上書きした IV_i は次の λ 周期のノイズノード削除の際に再び利用する。

次にコサイン類似度を適合度にする場合について説明する。コサイン類似度は式(11)によって求めることが出来る。

$$\begin{aligned} \cos(a, b) &= \frac{\langle a, b \rangle}{\|x\| \|y\|} \\ &= \frac{\sum_{k=1}^n a_k b_k}{\sqrt{\sum_{k=1}^n a_k^2} \sqrt{\sum_{k=1}^n b_k^2}} \end{aligned} \quad (11)$$

式(11)において a 、 b はそれぞれベクトルを表しており、それぞれ n 次元の同じサイズのベクトルである。コサイン類似度は二つのベクトルがどれほど似ているかを表すことができる。似ている場合は1に近づき、似ていない場合は-1に近づく。また、SOINN空間上で似ていないノードの多いノードが全体的に疎なデータ空間のノードであると考えられる。そのためコサイン類似度が負の値の時のほうが大きくなるようにしたい。そこで本研究では式(12)を適合度として割り当てた。本研究ではこの3つの適合度で実験を行う。

$$F_i = - \sum_{c \in A \setminus \{i\}} \cos(W_i, W_c) \quad (12)$$

4. 実験および検討

実験は手書き文字データセットMNISTを用いて行う。MNISTは図1のような"0"~"9"までの手書き文字のデータセットである。手書き文字データ1枚当たりの次元数は $28 \times 28 = 784$ 次元である。学習データとして60000枚のデータをテストデータとして10000枚のデータを用いて識別率、SOINN空間上のノード数、 λ 周期ごとのノードの削除数の平均を求める。



図1 MNIST の例

表1は $\lambda = 2000$ 、 $age_{max} = 100$ で実験を行った結果である。表2は $\lambda = 3000$ 、 $age_{max} = 100$ で実験を行った結果である。図1は $\lambda = 2000$ 、 $age_{max} = 100$ の場合の λ 周期ごとのノードの削除数の推移である。

表1 L2000_A100 の結果

	識別率	ノード数	削除平均
通常	89.54	436.67	1064.92
適合度 1	86.53	329.67	439.49
適合度 2	86.17	342.00	455.26
適合度 3	86.18	343.00	476.93

表2 L3000_A100 の結果

	識別率	ノード数	削除平均
通常	90.35	589.50	1558.36
適合度 1	88.33	462.20	806.06
適合度 2	88.32	468.10	836.76
適合度 3	88.20	453.40	836.67

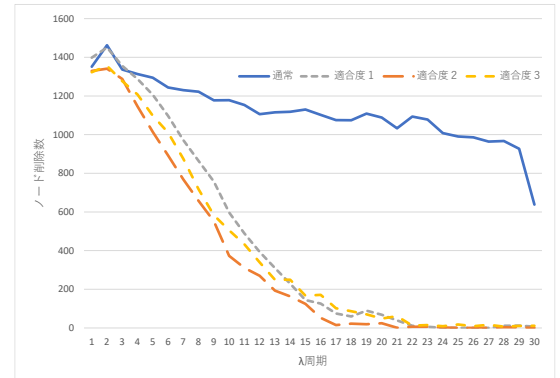


図1 λ 周期ごとのノード削除数

それぞれの適合度は適合度1が式(7)で表したユークリッド距離であり、適合度2は式(9)で表した入力値との距離の平均値である。また適合度3はコサイン類似度を適合度にしたものである。

通常のSOINNと比較すると λ 周期あたりのノードの削除数の平均が40~50%ほどに抑えられ、ノード数80%程に削減された。一方で識別率が2~3%程落ちてしまった。特に図1でみられるように λ 周期ごとにノイズノードとして削除されるノードの数が大幅に減っているため提案手法はSOINNネットワークの構築に関して安定させる効果があると考えられる。一方で通常のSOINNと比較した際、全体の識別率は落ちてしまっているため識別率を上げる必要がある。また各適合度において識別率の差はそこまで出なかったが $\lambda = 2000$ 、 $\lambda = 3000$ のどちらでも適合度1である頂点間距離が一番識別率が高かった。

そこで追加実験としてSOINN空間上のノード数を増やすためにStep10'-1からStep10'-4までの処理をする回数を変化させた。データの計測は適合度1でのみ行った。表3は $\lambda = 2000$ 、 $age_{max} = 100$ とし、適合度を頂点間距離としてStep10'-1からStep10'-4まで繰り返したものである。

表3 複数ノードを残した結果

	識別率	ノード数	削除平均
通常	89.54	436.67	1064.92
1回	86.53	329.67	439.49
5回	86.59	400.00	435.10
10回	86.98	618.50	390.71
20回	87.50	678.00	380.00
λ 周期回	86.98	618.50	390.71
削除の5%	86.17	513.00	321.97
削除の1%	86.18	321.33	410.94

Step10'-1からStep10'-4を繰り返すことで複数の疎なデータのノードを残すようにした。 λ 周期回残すというのは1回目の λ 周期は1つ、3回目の λ 周期回の場合は3ノードのように λ 周期ごとに残すノードの数を増やすようにした。結果として1つのみノードを残すようにするよりも複数個ノードを残すようにした方がノードの数も増え識別率も上がった。しかし、複数のノードを増やすようにしてもいたずらにノイズノードが増えただけになり、識別率の大きな向上に繋

がらなかった。このことから疎なデータ空間上に存在しているノードはノイズノードになる可能性を含めていると考えられる。

5. まとめ

本研究ではSOINNアルゴリズムにおけるネットワークの生成を安定させるために、複雑ネットワークの特徴量を用いてノイズノードの削除時に疎なデータ領域のノードを残すようにした。実験では手書き文字データMNISTを使用し、識別率、ノード数、 λ 周期ごとのノードの削除平均で評価を行った。

従来手法のSOINNに比べて提案手法では最終的なノード数を20%ほど削減し、 λ 周期ごとに削除されるノイズノードの数を40%程削減することに成功した。このことからSOINN空間上のネットワークの生成に関して従来手法のSOINNと比較して大幅に安定させることが出来たと考えられる。

一方で従来手法に比べて識別率が3%程下がってしまうという問題点が発生した。識別率低下の原因としてノード数が少なくなりすぎていることが考えられる。

また、追加実験として提案手法のStep10'-1からStep10'-4を複数回繰り返すものを実験した。 λ 周期ごとに追加で残すノードを増やした場合、1ノードのみを残すときよりは識別率が高くなった。しかし従来手法のSOINNよりも高い識別率を出すことが出来なかった。これは疎なデータ空間に存在しているノードはノイズになる可能性を含めていると考えられる。

今後は提案手法で得られた安定したSOINNネットワーク構成を生かしつつ学習後の識別率を上げる手法を考える必要がある。

参考文献

- [1] 山崎和博, 巻渕有哉, 申富饒, 長谷川修, 「SOINNとその実践」, 日本神経回路学会誌 Vol.17, No.4(2010), 187-196
- [2] Kohonen, T., "Self-organized formation of topologically correct feature maps", Biological Cybernetics, Vol.43(1982), pp.59-69.
- [3] Fritzke, B., "A growing neural gas network learns topologies", Advances in Neural Information Processing Systems(1995), pp.625-632
- [4] 柳井孝介, 伊庭斉志, 「ノード順位による選択を用いたスケールフリー・ネットワークモデル」, コンピュータソフトウェア, Vol24, No.1(2007), pp.50-61