

# Self-Organizing Fusional and Incremental Neural Network

日大生産工(院) ○種田 祥吾 日大生産工 山内 ゆかり

## 1. まえがき

教師なし学習の種類の一つとして、クラスタリングが存在し、クラスタリングの代表的な手法として、 $k$ -means法がある。 $k$ -means法は分類のためにクラス数を事前に定める必要がある。しかし、実世界のデータはクラス数が定められていない。また、実世界のデータは非定常な場合があり、非定常性の対処のために追加学習が必要である。Self-Organizing Fusional and Incremental Neural Network(SOINN)[1]は新しいクラスからの入力や、分布の形状変化に対応し動的にネットワークを構成し続けるアルゴリズムである。学習データを一括に処理するのではなく、一つ一つ入力されるデータに対して逐次学習することが可能である。

adjusted SOINN はSOINNのアルゴリズムで共通して利用されている。その中で、「ノードの削除」という処理があるが、これは $\lambda$ 周期ごとに連結ノードが1以下のノードはノイズの影響を受けている可能性が高いという仮定に基づいて削除を行っている。しかし、必ず削除ノードがノイズであるかは曖昧である。また、 $\lambda$ 周期の手前のステップにおいて入力されたノードは、エッジを他のノードと繋げず、学習を十分に行わずに、 $\lambda$ 周期で消去してしまうという問題がある。本研究では、ノイズの可能性のある曖昧なノードに対して、両端のノード情報を融合するNode Fusionを提案する。また、それぞれのノードについて年齢の概念を取り入れることで、 $\lambda$ 周期を考慮せずに、一つ一つのノードを十分に学習させるノードの加齢処理を提案する。未学習ノードを減らし、融合したノードを再学習に利用することで、安定したネットワークの生成を目指す。

## 2. 従来研究

adjusted SOINNはSOINNシリーズのアルゴリズムでよく用いられており、オンライン学習による自己増殖的なネットワークの構成と、ネットワーク構造に基づく入力分布の近似を目的としている。適切な分布の近似のために入力に対して、「ノードの挿入」、「エッジの挿入・削除」、「ノードの位置ベクトル更新」、「ノードの削除」を組み合わせる実行する。

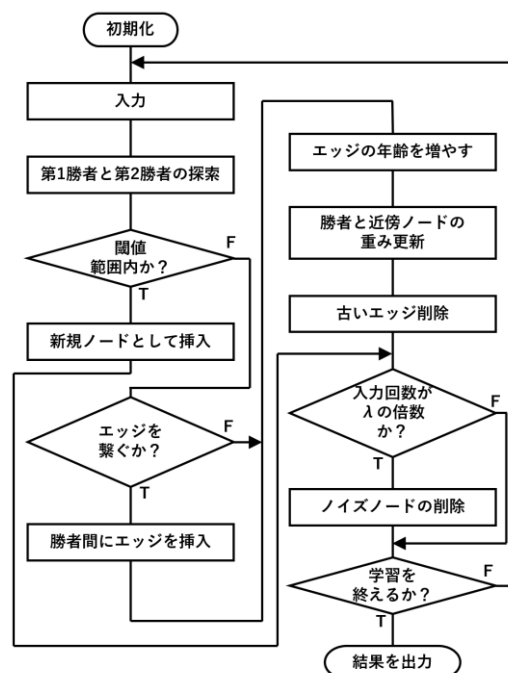


Figure 1 adjusted SOINN のフローチャート

Figure 1に adjusted SOINN のフローチャートを示す。入力されたデータに対し、ネットワークのそれぞれのノードとのユークリッド距離を比較し、一番似ているノードを第1勝者 $s_1$ 、次に似ているノードを第2勝者 $s_2$ とする。それぞれの勝者に対し類似度閾値  $T$  を求める。Figure 2にノードの挿入処理を示す。

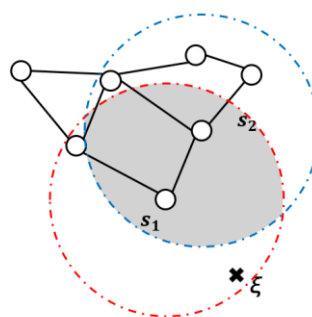


Figure 2 ノードの挿入処理

類似度閾値範囲外であるとき、分布の近似が不十分であるため新規ノードとしてネットワークに挿入する。類似度閾値は連結ノードが存在し

ない場合は式(1)の最近傍ノードとの距離を、連結ノードが存在する場合は式(2)の最も遠いノードとの距離とする。

$$T_i \leftarrow \max_{c \in N_i} \|W_i - W_c\| \quad (1)$$

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

閾値範囲外となった時、エッジの挿入と削除を行う。まず、勝者同士にエッジが存在しなければ、2つのノードをエッジで連結する。次に学習によって更新されたネットワークにおいて初期に構築された連結関係は不適切になる可能性があるため、エッジの加齢処理によって参照されなくなったエッジを削除する。

エッジの処理後、ノードの位置ベクトルの更新と削除を行う。入力パターンに対して、勝者ノードと勝者ノードに連結するノードの位置ベクトルが更新される。位置ベクトル更新後、連結ノード数が1以下のノードはノイズの影響を受けている可能性が高いという仮定に基づき、削除する。

### 3. 提案手法

SOINNの問題点として曖昧なノードと未学習のノードの学習が挙げられる。Figure 3にネットワーク上における曖昧なノードを示す。

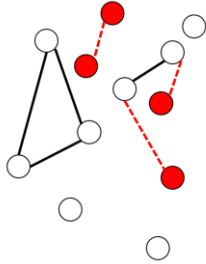


Figure 3 曖昧なノード

通常、連結ノード数が1以下のノードはノイズである可能性が高いという仮定に基づき削除するが、対象ノードがノイズであるかは曖昧である。また、SOINNはノイズノードの削除を入力パターンが一定数を超えるたびに実行するが、入力パターンが一定数になる直前に挿入されたノードは位置ベクトルの更新が行われる可能性が低く、未学習のままノイズとして削除される。本研究では、この2つの問題に対して提案を行う。

#### 3.1. Node Fusion

種田らは画像認識における畳み込み学習の全結合層において十分に小さくなった重みにつながるノードを他のノードに付加するNode Fusion[2]を提案した。このNode Fusionを、SOINN用アルゴリズムとして改良し、Self-Organizing Fusional and Incremental Neural

Networkを提案する。以降、このモデルをSOFINNと呼ぶ。

まず、削除条件 $\theta$ を定義する。連結ノード数 $N_i$ が $\theta$ 以下のノード $i$ をSOINNのアルゴリズムに従って削除する。 $\theta = 0$ は孤立ノードのみの削除を意味する。次に $N_i = \theta + 1$ のノードを探し、融合ノード集合を作成する。融合ノード集合は対象ノード $a$ と連結ノード番号 $b$ を記憶する。

融合ノード集合を作成後、ノードの融合を行う。融合は対象ノード $a$ が連結ノード $b$ に融合される。位置ベクトルの更新式を式(3)に示す。 $W$ はノードの位置ベクトルで第1勝者に選択された回数 $M$ を用いて更新する。

$$W_b = \frac{M_a}{M_a + M_b} W_a + \frac{M_b}{M_a + M_b} W_b \quad (3)$$

次に勝者回数を融合する。勝者回数の更新式を式(4)に示す。勝者回数融合後、融合対象ノードは不要となるため、ノード集合から削除する。

$$M_b = M_a + M_b \quad (4)$$

#### 3.2. ノードの加齢処理

個々のノードに年齢の処理を追加する。Figure 4にネットワーク上の加齢処理を示す。新規ノードとして追加されたノードは0で初期化される。入力データに対して全てのノードの年齢を1増加し、第1勝者のノードの年齢は0に初期化する。個々のノードで年齢が $\lambda$ 倍数となったときノイズノードであるかどうかを判別し、削除処理を行う。ノードの削除判断は個々に実行するため、 $\lambda$ 周期での全ノードの削除判断処理は実行しない。

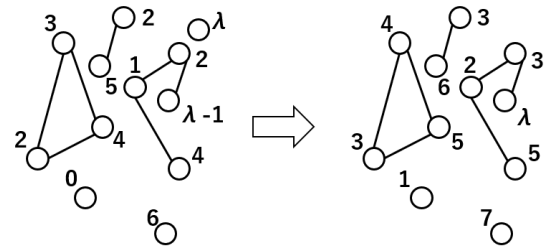


Figure 4 ノードの加齢処理

#### 3.3. Add Edge

土田らは新規ノードにエッジがつかず、削除されてしまう問題に対し、Add Edge[3]を提案した。Figure 5にAdd Edgeのノード挿入処理を示す。Add Edgeは新規ノードをネットワークに挿入する際に、第1勝者と年齢1のエッジを繋ぐ。エッジを繋ぐことで、孤立ノードの生成を抑制する。

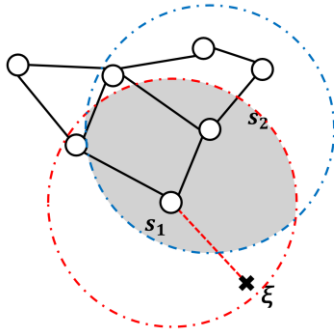


Figure 5 Add Edge のノード挿入処理

#### 4. 実験および検討

adjusted SOINN を基として Optdigit と MNIST の手書き文字データセットを用いた実験を行う。Optdigit の  $\lambda$  は 400、 $Age_{max}$  を 10 とし、MNIST の  $\lambda$  は 600、 $Age_{max}$  を 100 とする。まず従来手法の adjusted SOINN [1] を Baseline とする。提案手法は、削除条件  $\theta = 0$  を手法 A、 $\theta = 1$  を手法 B とする。次に新規ノードの追加時の Add Edge を手法 C とする。さらに、ノードの加齢処理を手法 D とし、排他的な A と B 以外のすべての条件の組み合わせについて実験を行う。提案一覧を Table 1 に示す。

Table 1 提案一覧

|     | A | B | C | D |
|-----|---|---|---|---|
| 従来  |   |   |   |   |
| 提案① | ✓ |   |   |   |
| 提案② |   | ✓ |   |   |
| 提案③ |   | ✓ | ✓ |   |
| 提案④ | ✓ |   | ✓ |   |
| 提案⑤ |   |   |   | ✓ |
| 提案⑥ | ✓ |   |   | ✓ |
| 提案⑦ |   | ✓ |   | ✓ |
| 提案⑧ |   | ✓ | ✓ | ✓ |
| 提案⑨ | ✓ |   | ✓ | ✓ |

実験の評価については学習精度、生成されたネットワークのノード数、新規に追加されたノード数、削除されたノード数、融合ノード数を比較する。また学習精度とノードの標準偏差により生成された SOINN 空間の安定性についても検討する。

手法 B と手法 D の組み合わせは、ノードの加齢処理でノードを一つずつ確認していくため、ノイズノードの削除は可能だが、ノイズノード削除のみで融合を行わない可能性が高い。この組み合わせに関して、融合を促すために、連結ノード数 1 のノードを見つけたとき、繋がる連結ノードを融合対象ノードとし、ノイズノード削除後に融合を行う。

Optdigit の 10 回平均の実験結果 I を Table 2 に示す。

Table 2 実験結果 I

|     | accuracy | node  | new    | delete | fusion |
|-----|----------|-------|--------|--------|--------|
| 従来  | 91.47    | 89.7  | 2326.8 | 2239.1 | 0.0    |
| 提案① | 91.55    | 82.6  | 2413.7 | 2333.1 | 401.9  |
| 提案② | 88.32    | 75.4  | 2474.8 | 2401.4 | 54.2   |
| 提案③ | 95.15    | 251.3 | 1436.2 | 1186.9 | 161.5  |
| 提案④ | 95.47    | 308.2 | 1507.8 | 1201.6 | 1197.5 |
| 提案⑤ | 92.16    | 161.7 | 2685.6 | 2525.9 | 0.0    |
| 提案⑥ | 92.84    | 168.6 | 2615.7 | 2449.1 | 412.4  |
| 提案⑦ | 91.76    | 154.5 | 2686.5 | 2534.0 | 62.3   |
| 提案⑧ | 95.86    | 449.7 | 1404.9 | 957.2  | 129.1  |
| 提案⑨ | 96.05    | 498.7 | 1439.6 | 942.9  | 936.3  |

従来、提案③、提案④、提案⑧、提案⑨を比較する。提案 4 つは従来より精度が約 4% 向上した。ノード数は全て 2 倍以上となったが、ノードの加齢処理を取り入れた提案⑧、提案⑨はネットワークのノード数が 5 倍程となり、過剰にノードを生成した。これは、 $\lambda$  周期前のノードについて考慮できるようになったが、学習終わりより前の部分で学習しきれなかったノードが残ったためだと考えられる。次に融合ノードについて確認する。融合条件によって融合ノード数に 10 倍近い差が出た。提案④、提案⑨はデータ数に対して 3 分の 1 のデータを融合している。これは学習データに対して過剰融合していると考えられる。ノード生成・削除数を抑え、融合回数を抑えていることから実験結果①の提案③はもっとも有効であるといえる。

次に、MNIST の 5 回平均の実験結果 II を Table 3 に示す。

Table 3 実験結果 II

|     | accuracy | node   | new     | delete  | fusion  |
|-----|----------|--------|---------|---------|---------|
| 従来  | 84.70    | 175.6  | 36136.6 | 35963.0 | 0.0     |
| 提案① | 82.17    | 141.4  | 41847.8 | 41708.4 | 2503.8  |
| 提案② | 82.73    | 157.6  | 38820.6 | 38665.0 | 92.6    |
| 提案③ | 90.89    | 582.4  | 25719.2 | 25138.8 | 902.4   |
| 提案④ | 91.67    | 616.6  | 33001.4 | 32386.8 | 32386.8 |
| 提案⑤ | 88.27    | 354.6  | 36803.8 | 36451.2 | 0.0     |
| 提案⑥ | 87.24    | 301.4  | 40259.6 | 39960.2 | 2794.4  |
| 提案⑦ | 87.73    | 331.0  | 38422.4 | 38093.4 | 171.4   |
| 提案⑧ | 92.08    | 1142.8 | 23089.4 | 21948.6 | 968.4   |
| 提案⑨ | 92.54    | 1231.8 | 28624.6 | 27394.8 | 27394.6 |

提案③が多くのノードを融合しすぎず、新規ノード数と削除ノード数を減少させながら、従来モデルを 5% 程上回る結果を示した。手法 A①⑥か手法 B②⑦のみの効果について確認すると、従来及び⑤よりも新規ノード数と削除ノード数が増加し、学習後のノード数が減少したことで、学習精度が下がってしまった。このことから、SOFINN の単体モデルは高次元データとの相性が悪いことがいえる。ノードの加齢処理を加えた提案⑦はノード数を提案③より抑えつつ、融合しすぎずに精度向上できているため、提案③に加え、高次元データにおいてノードの加齢処理を加える場合は連結ノード 2 のノードを融

合する手法が有効であるといえる。2つのデータセットを用いた実験結果から、ノード数、新規ノード数、削除ノード数が少なく、ノードを過剰に融合しないモデルが最適であると考えられる。

実験結果IIより、従来モデルと提案③、提案⑦の学習時のノード数の変化についてFigure 6に示す。

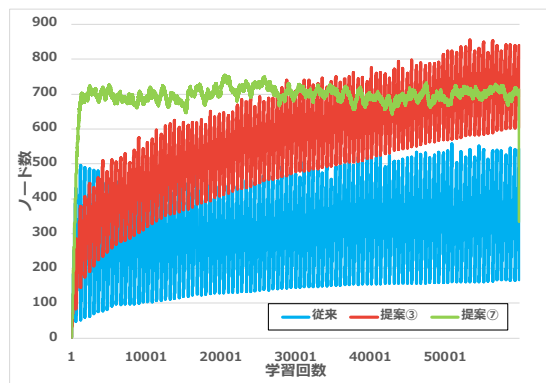


Figure 6 ノード数の変化

3つの学習時間はそれぞれ56秒、90秒、141秒であった。Figure 6から提案③は徐々に従来のノード数を超えていくため、学習時間が従来より遅くなり、提案⑦は初期で従来よりノード数が超え、同じノード数付近を維持するため、学習時間が3倍近くかかった。

融合回数の多かった提案④についてFigure 7にMNIST学習後のネットワークのノードを示す。

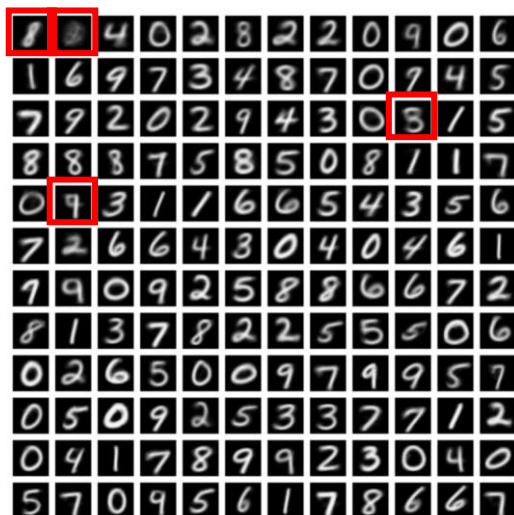


Figure 7 学習後の SOFINN ノード

各ノードの画像データを確認すると、融合による同一ラベルの特徴を大まかに捉えたノードは学習できているが、他ラベルのノードと融合した時、ラベル判別が不可能なノードをネットワークに残し、判別不可ノードを含めた学習を行ってしまう問題があることが分かった。

Table 4に2つのデータセットにおける学習精度とノードの標準偏差を示す。

Table 4 学習精度とノードの標準偏差

|     | accuracy① | node①   | accuracy② | node②   |
|-----|-----------|---------|-----------|---------|
| 従来  | 0.0084    | 7.0875  | 0.0136    | 17.4550 |
| 提案① | 0.0114    | 10.4265 | 0.0364    | 8.9821  |
| 提案② | 0.0367    | 8.5271  | 0.0343    | 12.1239 |
| 提案③ | 0.0051    | 9.9560  | 0.0044    | 20.6583 |
| 提案④ | 0.0055    | 24.7736 | 0.0029    | 20.9221 |
| 提案⑤ | 0.0104    | 8.9449  | 0.0054    | 15.9196 |
| 提案⑥ | 0.0109    | 16.5072 | 0.0060    | 9.4304  |
| 提案⑦ | 0.0122    | 12.2859 | 0.0209    | 11.8397 |
| 提案⑧ | 0.0060    | 10.3285 | 0.0057    | 15.7906 |
| 提案⑨ | 0.0050    | 27.8011 | 0.0028    | 32.3137 |

従来モデルと提案③について比較すると、倍のノード数である提案③は同程度のノードの標準偏差で学習精度を小さい標準偏差で得ることができている。つまり、提案③のSOFINNは安定したネットワークの生成ができ、安定した学習精度を得ることができるといえる。

## 5. まとめ

本研究では、SOINNの問題点である曖昧なノードと未学習ノードに焦点を当て、Node Fusionと複数の改善手法を取り入れたSOFINNを提案した。実験結果として、Add Edgeによって孤立ノードをできにくくし、連結ノード数1以下のノイズノードを削除した後、連結ノード数2のノードを融合することで、安定したネットワークの生成とともに、学習精度を向上するモデルを得ることができた。

今後は高次元データにおけるSOFINNにおいて、融合条件を考慮することで判別不可能ノードの生成を抑えるモデルについて検討したい。また、未学習ノードに対して提案したノードの加齢処理は学習初期にノード数が増加することで学習時間がかかってしまう問題が確認できたため、ノードの加齢処理を取り入れる場合の高速化について検討したい。

## 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) 種田祥吾, 野口尚馬, 山内ゆかり, 「スパース化DenseNetにおけるNode Fusionの提案」, 信学技報, vol. 122, no. 425, NC2022-110, pp. 105-108, 2023年3月.
- 3) 土田悠介, 山内ゆかり, 「教師ありSOINNにおける高次元耐性の向上」, 日本大学, 生産工学部, 学術講演会. No.55, (2022), 2-5.