

V-SOINN を利用した SOINN 空間の最適化

日大生産工 ○相木 翼 日大生産工 山内 ゆかり

1. まえがき

近年、ビッグデータの活用が注目されている。ビッグデータのような実世界でのデータ処理では、データの増大やデータに含まれるノイズへの対処が必要である。山崎らが提案した自己増殖型ニューラルネットワーク (Self-Organizing Incremental Neural Network : SOINN)[1]は、教師なし学習の一つであり、追加学習が可能で、高いノイズ耐性を有しているという特徴がある。しかし、SOINNには高次元データに対する性能が低いという問題がある。また、Hui Douらが提案したV-SOINN[2]は、高次元SOINN空間のトポロジーを維持しつつ、低次元空間への変換を可能にした。

本研究では、V-SOINNを利用し、SOINN空間の改善を図る。MNISTでの実験により提案手法と通常のSOINNを比較する。

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はいくつかの手法が提案されているが、基本的な Adjusted SOINN について説明する。Adjusted SOINN は1層構造であり、オンライン学習を用いて自己増殖的にネットワークを構成し、このネットワーク構造に基づく入力分布の近似を行うことができる。Adjusted SOINN のフローチャートを図1に示す。

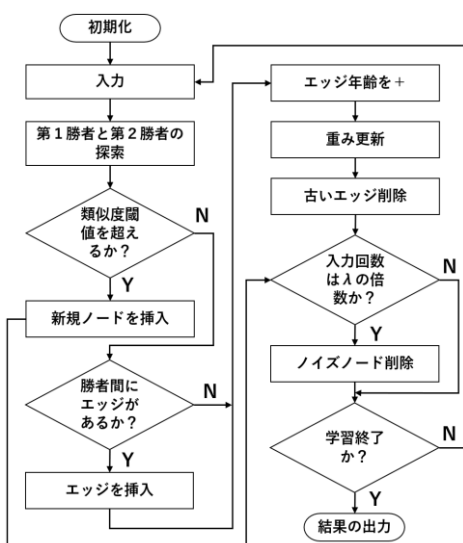


図1 Adjusted SOINN のフローチャート

Adjusted SOINNでは、学習データが入力として与えられると、既存のノードとのユークリッド距離を計算し、最も距離の近いノードを第1勝者、次に近いノードを第2勝者とする。それから、第1・第2勝者ノードについて類似度閾値 T_i を式(1)により計算する。ただし、 i はノード番号、 N_i はノード i の連結ノード集合、 A は全ノードの集合、 W_i はノード i の位置ベクトル(重み)である。

$$\text{if } N_i \neq \emptyset \begin{cases} T: T_i \leftarrow \max_{c \in N_i} \|W_i - W_c\| \\ F: T_i \leftarrow \min_{c \in A \setminus \{i\}} \|W_i - W_c\| \end{cases} \quad (1)$$

入力と第1・第2勝者ノードとの距離がいずれかの類似度閾値を超える場合、入力が新規ノードとして挿入される。類似度閾値を超えない場合、重みの更新を行う。まず勝者間にエッジを年齢0として挿入し、第1勝者に繋がるエッジの年齢を1加算する。そうして、式(2)により第1勝者ノードの重みの、式(3)により第1勝者と繋がるノードの重みの修正分を求めて重みを更新する。ただし、 s_1 は第1勝者ノード番号、 ξ は入力の重み、 t_i はノード i が勝者ノードに選択された回数である。

$$\Delta W_{s_1} \leftarrow \frac{1}{t_i} (\xi - W_{s_1}) \quad (2)$$

$$\Delta W_i \leftarrow \frac{1}{100t_i} (\xi - W_i), \forall i \in N_{s_1} \quad (3)$$

重み更新後、エッジ年齢が年齢閾値 age_{max} を超えたエッジを削除する。さらに、削除したエッジで連結されていたノードのうち、エッジを持たなくなったノードを削除する。この後と新規ノード挿入後、入力回数が削除周期 λ の倍数である場合、エッジが1本以下のノードを削除する。これを学習終了まで繰り返す。

age_{max} 、 λ はともにネットワーク構成に影響するハイパーパラメータである。 age_{max} はエッジの削除頻度に、 λ はノードが疎な領域での削除頻度に影響を与える。

2.2 V-SOINN

V-SOINNでは、高次元データによるSOINN空間をもとに、低次元空間に描画する。高次元データによるSOINN空間で、特定の2次元を描画したものを図2に示す。

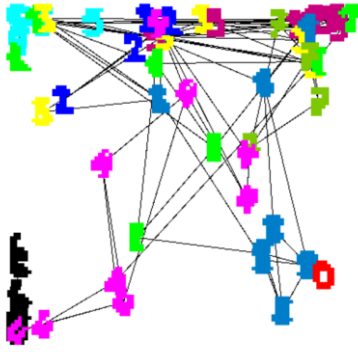


図2 SOINN空間の特定の2次元の描画

図2では、SOINN空間上のネットワーク構造が表示されているが、その特徴をとらえているとは言えない。このように、高次元SOINN空間は、そのままでは位相構造を保ちつつ可視化することができない。

これに対して、V-SOINNの学習後の低次元空間を図3に示す。

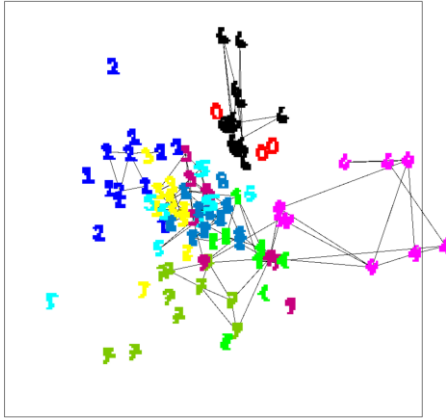


図3 V-SOINNによる描画空間(2次元)

図3を見ると、V-SOINNによって類似したノードが近くに配置され、位相構造を保ったまま2次元空間上に描画されていることがわかる。

V-SOINNは、通常のSOINNの学習が終了した後、以下の手順で低次元空間ノードの重みを学習する。

まず、高次元空間ノードの重み W_a と、対応する低次元空間ノードの重み W'_a を式(4)により正規化する。ただし、 μ, σ は高次元空間ノードの重みの平均と分散、 μ', σ' は低次元空間ノードの重みの平均と分散である。

$$W_a = \frac{W_a - \mu}{\sigma}, W'_a = \frac{W'_a - \mu'}{\sigma'} \quad (4)$$

次に、低次元空間ノードの重みを更新する。式(5)と式(6)により、全てのノード間における高次元空間と低次元空間の距離の差を求める。ただし、 d はユークリッド距離である。

$$S(a, b) = (d(W_a - W_b) - d(W'_a - W'_b))^2 \quad (5)$$

$$S_1(a) = \sum_{b \in A} S(a, b) \quad (6)$$

この距離差 $S_1(a)$ を最小化するように、式(7)で偏微分計算を行い、低次元空間の重み W'_a を更新(学習)する。なお、 η_1 は学習率である。

$$W'_a \leftarrow W'_a - \eta_1 \frac{\partial S_1(a)}{\partial W'_a} \quad (7)$$

これを学習回数分行って、高次元空間のノード間の距離関係と低次元空間のノードの距離関係を対応させる。これにより、高次元空間のトポロジカルなネットワーク構造を維持しつつ、低次元空間に変換することができる。

3. 提案手法

通常のSOINNでは、データが高次元になるほど、入力が勝者ノードと類似しているにもかかわらず、2つの類似度閾値の間に存在せずに新規ノードとして追加されてしまう。そのため余計なノードが増えてしまい、学習が安定しないという問題点が存在する。本研究では、V-SOINNによる低次元空間の情報をSOINN空間のネットワーク形成に利用し、学習の安定化を図る。

4. 実験および検討

実験では、学習データおよびテストデータとして手書き文字画像データセットMNISTを使用する。高次元なSOINN空間では、余分なノードが追加され、学習が安定しないという問題がある。V-SOINNにより生成される空間はSOINN空間のトポロジーを維持しており、低次元空間であるため、この情報を利用することで学習が安定すると考えられる。

5. まとめ

本研究では、V-SOINNを用いてSOINN空間のネットワーク生成の最適化を目指す。V-SOINNによって生成される低次元空間の情報を利用することで、学習の安定化が可能になると考えられる。

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] Hui Dou, Baile Xu, Furaio Shen and Jian Zhao, “V-SOINN: A topology preserving visualization method for multidimensional data”, Neurocomputing, 449, 280–289 (2021).