

# Multi-Scale Batch Learning SOINN の提案

日大生産工 ○甲木 智香 日大生産工 山内 ゆかり

## 1. まえがき

現在では、通信ネットワークやセンサーの発展・普及に伴い、実環境からリアルタイムで膨大な量のデータが生成され続けている。この膨大なデータをビッグデータという。それらの実世界のデータはノイズが多く、分布の形状が変化するなど非定常であるため、対処するには追加学習の機能が必要となる。長谷川修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]という教師なし学習アルゴリズムを提案した。この手法は、自己組織化マップ (Self-Organizing Map: SOM) と Growing Neural Gas (GNG) に着想を得て構築された手法である。これは、ノイズに強く、SOMのように事前に分布の形状を決定する必要がなく、追加学習が可能という特徴がある。しかし、ネットワークの構成が動的であるため安定しない。また、高次元数に応じて教師ラベルの学習への影響が変わってしまうという問題点がある。

一方、通常のGNG[2]とBatch Learning Growing Neural Gas (BL-GNG)は、ノードの特徴、グループ化、トポロジカルマッピングを抽出することにより、必要な情報を迅速に見つけることを目的としているため、学習を大幅に高速化させることは困難である。そこでFernando Ardillaらは、学習と収束を劇的に加速させる画期的な成長戦略を提案した[3]。それは、BL-GNGに対して、3番目に近いノードと1番目と2番目に近いノードの距離が最大になるように推定されたノード追加の尤度に基づいて、データのサンプルを新しいノードとして既存のネットワークに即座に追加する。

しかし、この手法のパラメータはデータ構造に依存するため、オンライン漸進教師なし学習アルゴリズムを使用することは困難であるという問題点がある。そこで本研究では、SOINNにMSBLの追加を行い、安定したネットワークにすることを目的とし、提案する (MSBL-SOINN)。文字認識における計算機実験により提案手法と精度を比較し、報告する。

## 2. 従来研究

MSBL-GNGのアルゴリズムをフローチャートとしてFigure1に示し、概要を説明する。

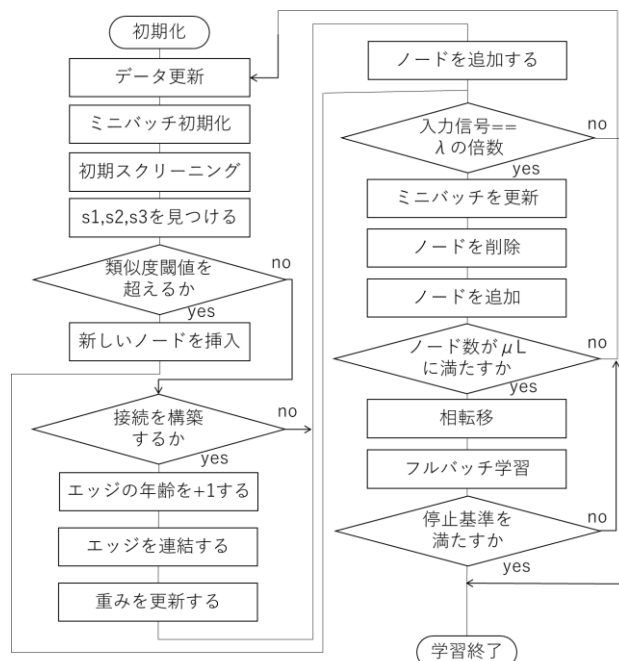


Figure1. MSBL-GNGのアルゴリズム

エッジに3つの接続ノードを作成し、学習段階のレベルを初期化する ( $L=1$ )。入力値をランダムに選択し、次に初期サンプリングデータID ( $it$ ) を1に設定する。このとき成長段階は1 ( $K=1$ ) である。

次に入力値を更新する。一時的な重みの更新と選択時間を初期化し、時間的エッジ連結性を作成してミニバッチを初期化する。

最初のミニバッチのノード率が高ければ、現在のデータセットを用いて初期化を再開する。または、最初のミニバッチから前回のタイステップまでのノードの割合が事前に定義された閾値に達していれば、現在のデータセットを使って初期化を再開し、初期スクリーニングを行う。

次にデータのサンプルに従って重みと時間エッジ接続を更新する。一番ユークリッド距離が近いノードを第1勝者 ( $s_1$ ) 式(1)として、同じように第2勝者 ( $s_2$ ) 式(2)と第3勝者 ( $s_3$ ) 式(3)も作成する。

$$s_1 = \underset{i \in A}{\operatorname{argmin}} d_i \quad (1)$$

$$s_2 = \underset{i \in A \setminus \{s_1\}}{\operatorname{argmin}} d_i \quad (2)$$

$$s_3 = \underset{i \in A \setminus \{s_1, s_2\}}{\operatorname{argmin}} d_i \quad (3)$$

式(4)では、入力ベクトルと参照ベクトル間の距離を累積誤差に加える。

$$E_{s1} \leftarrow E_{s1} + \eta 1 d_{s1} i \quad (4)$$

勝者とノードの重みを式(5)、式(6)で計算して更新する。ここでの入力数は学習率である。

$$\Delta w_{s1} \leftarrow \Delta w_{s1} + \eta 1 (v - w_{s1}) \quad (5)$$

$$\Delta w_j \leftarrow \Delta w_j + \eta 2 (v - w_j) \text{ if } c_{s1,j} = 1 \quad (6)$$

さらに、エッジの年齢をインクリメントする。選択回数を増加させ、時間的エッジ連結性を増加させる。そして、ネットワークに新規ノードを追加する。

入力値をインクリメントして  $\mu L$ (入力数)の整数倍でなければ、ミニバッチ初期化に進む。

そして、ミニバッチを更新する。MSBL式(7)式(8)により重みを更新及びノードを削除する。

$$\Delta w_j \leftarrow w_j + \frac{\Delta w_j}{(x_i^1 + x_i^2)}, \text{ if } (x_i^1 + x_i^2) > 0 \quad (7)$$

$$c_{i,j} = \begin{cases} 1 & \text{if } c'_{i,j} \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

選択ベースの削除戦略に従ってノードの削除を行う。エラーに基づく追加戦略に従い、新規ノードを追加する。

ノード数がプリセット値  $\mu L$  に満たない場合は新たな入力値を追加する。

次に相転移を行う。学習段階のレベルを上げ、 $L_{max}$ と成長段階のレベル  $k$  を上げて、次のミニバッチデータセットの先頭に移動させる。

フルバッチ学習を行う。

最後に停止基準（ネットワークのサイズや性能など）を満たさない場合は新たな入力値を追加する。

MSBL-GNGとオリジナルのGNGでは年齢が異なる。ミニバッチデータセットはエッジの接続をリフレッシュし、エッジとエッジのないノードを削除する。MSBL-GNGはバッチ学習では年齢パラメータを用いず、学習率と参照ベクトルを更新する。MSBL-GNGは、データの分散をカバーするために、最初に2番目に近いノードに高い学習率を適用する。この手法はノード数が多いほど遅くなる。

$$\eta 2 = \alpha_2 \cdot (L_{max} - L) \quad (9)$$

$$\eta 1 = 1.0 - \alpha_2 \cdot (L_{max} - L) \quad (10)$$

ここで、 $L_{max}$ は最大学習段階レベル、 $\alpha$ は係数である。最後に、フルバッチ学習は最後の学習フェーズ ( $L = L_{max}$ )でフルバッチ学習が行われ、参照ベクトルとそのエッジ結合性を更新する。段階レベルが増加していく様子をFigure2に示す。

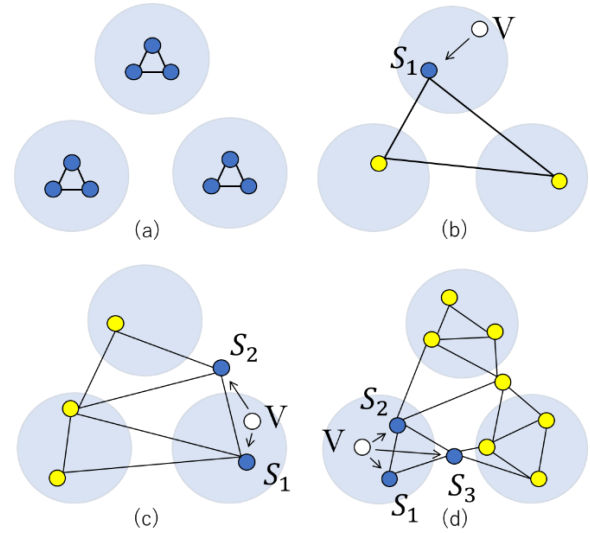


Figure2. AISに基づく付加戦略  
(a) Topological clustering. (b) 第1段階レベル(L=1).  
(c) 第2段階レベル(L=2). (d) 第3段階レベル(L=3).

### 3. 提案手法

従来のSOINNはネットワークの構成が動的であるため安定しないという問題点がある。そのため本研究では、SOINNにマルチスケールバッチ学習(MSBL)を追加することによって、これらの問題を解決する。

### 4. 実験および検討

従来のSOINNの問題点であった、ネットワークの不安定や高次元データへの対応が考慮され、学習が安定すると考えられる。

### 5. まとめ

SOINNにMSBLの学習を追加することにより、MSBL-SOINNと従来研究のSOINNの比較を行った。

### 参考文献

- [1] F. Shen, O. Hasegawa, “An incremental network for on-line unsupervised classification and topology learning,” *Neural Networks*, vol.19, No.1, 2005, pp90-106
- [2] Y. Toda, T. Matsuno, and M. Minami, “Multilayer batch learning growing neural gas for learning multiscale topologies,” *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 25, no. 6, pp. 1011–1023, 2021.
- [3] F. Ardilla, A. A. Saputra, N. Kubota “Batch Learning Growing Neural Gas for Sequential Point Cloud Processing”, Tokyo Metropolitan University (2022).