Self-Organizing Incremental Neural Network and Its Application

Furao Shen[1,2] and Osamu Hasegawa[3]

[1] National Key Laboratory for Novel Software Technology, Nanjing University, China
frshen@nju.edu.cn
http://cs.nju.edu.cn/rinc/
[2] Jiangyin Information Technology Research Institute, Nanjing University, China
[3] Imaging Science and Engineering Lab., Tokyo Institute of Technology, Japan
hasegawa@isl.titech.ac.jp

Abstract. Self-organizing incremental neural network (SOINN) is introduced. SOINN is able to represent the topology structure of input data, incrementally learn new knowledge without destroy of learned knowledge, and process online non-stationary data. It is free of prior conditions such as a suitable network structure or network size, and it is also robust to noise. SOINN has been adapted for unsupervised learning, supervised learning, semi-supervised learning, and active learning tasks. Also, SOINN is used for some applications such as associative memory, pattern-based reasoning, word-grounding, gesture recognition, and robotics.

Keywords: Self-organizing incremental neural network; Incremental learning; Competitive learning.

1 Introduction

Incremental learning addresses the ability of repeatedly training a network using new data without destroying the old prototype patterns. The fundamental issue for incremental learning is how a learning system can adapt to new information without corrupting or forgetting previously learned information: the so-called Stability-Plasticity Dilemma [1].

Numerous online incremental learning algorithms based on competitive neural networks have been proposed and applied in many applications. A salient advantage of competitive neural networks is their capability of operating with information of new data incrementally. Well known examples of competitive neural networks include Kohonen's self-organizing map (SOM) and its modification for supervised learning, Learning Vector Quantization (LVQ) [2]. SOM and LVQ are unsuitable for incremental learning, the number of nodes of such methods is predefined. Self-growing type SOM [3] can improve their performance by insertion of nodes and the gradual increase of their structural complexity. In addition, the Growing Neural Gas (GNG) architecture [4] is a modification to the GCS, in which the dimensionality of topological structures is not predefined, but is instead discovered during training. In self-growing SOM, GCS, and GNG, the

K. Diamantaras, W. Duch, L.S. Iliadis (Eds.): ICANN 2010, Part III, LNCS 6354, pp. 535–540, 2010.

number of nodes is not predefined and, by insertion of nodes, new information can be learned. However, these methods require predefinition of the maximum number of nodes because insertion of nodes continues interminably as long as new input patterns come. In this regard, if the number of nodes reaches the maximum, further inputs engender the possibility of destruction.

In this paper, we introduce SOINN [5] for online incremental learning. In SOINN, the insertion of nodes is stopped and restarted automatically with new input patterns. Thereby, it avoids the indefinite increase of nodes, and it intends to present a balance of stability and plasticity [5].

2 Self-Organizing Incremental Neural Network (SOINN)

A SOINN adopts a two-layer network. The first layer learns the density distribution of the input data and uses nodes and edges to represent the distribution. The second layer separates clusters by detecting the low-density area of input data, and uses fewer nodes than the first-layer to represent the topological structure of input data. When the second-layer learning is finished, SOINN reports the number of clusters and gives typical prototype nodes of every cluster. It also adopts the same learning algorithm for the first and second layers. Fig. 1 shows a flowchart of the SOINN learning process.

When an input vector is given to SOINN, it finds the nearest node (winner) and the second nearest node (runner up) of the input vector. It subsequently judges if the input vector belongs to the same cluster of the winner or runner up using the similarity threshold criterion. The first layer of SOINN adaptively updates the similarity threshold of every node because the input data distribution is unknown. If node i has neighbor nodes, the similarity threshold T_i is calculated using the maximum distance between node i and its neighboring nodes.

$$T_i = \max_{j \in N_i} \|\boldsymbol{W}_i - \boldsymbol{W}_j\| \tag{1}$$

Therein, N_i is the set of neighbor nodes of node i and $\boldsymbol{W}_i$ is the weight vector of node i. A similarity threshold T_i is defined as the minimum distance between node i and other nodes in the network if node i has no neighbor nodes.

$$T_i = \min_{j \in N \setminus \{i\}} \|\boldsymbol{W}_i - \boldsymbol{W}_j\| \tag{2}$$

Here, N is the set of all nodes.

The input vector will be inserted to the network as a new node to represent the first node of a new class if the distance between the input vector and the winner or runner up is greater than the similarity threshold of a winner or runner up. This insertion is called a between-class insertion because this insertion will engender the generation of a new class, even if the generated new class might be classified to some older class in the future.

If the input vector is judged as belonging to the same cluster of winner or second winner, and if no edge connects the winner and second winner, connect

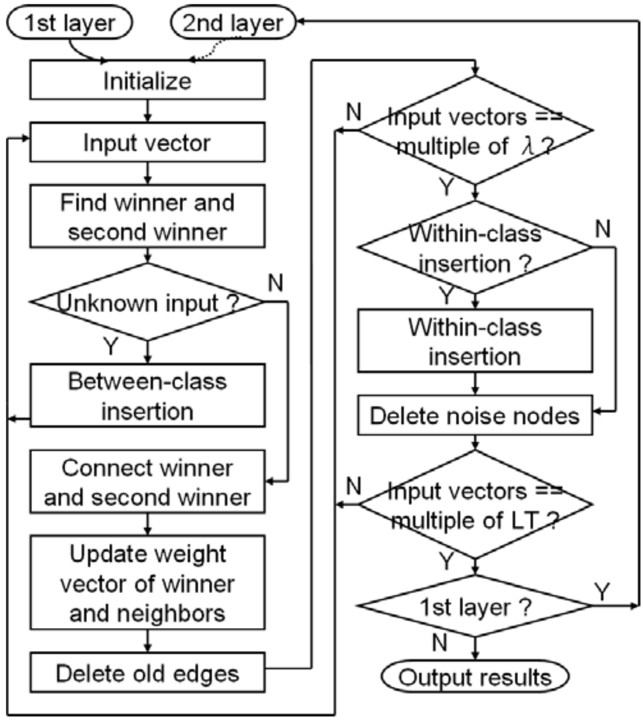

Fig. 1. Flowchart of SOINN

the winner and second winner with an edge, and set the 'age' of the edge as '0'; subsequently, increase the age of all edges linked to the winner by '1'.

Then, update the weight vector of the winner and its neighboring nodes. We use i to mark the winner node, and M_i to show the times for node i to be a winner. The change to the weight of winner $\varDelta \boldsymbol{W}_i$ and change to the weight of the neighbor node $j(\in N_i)$ of i $\varDelta \boldsymbol{W}_j$ are defined as $\varDelta \boldsymbol{W}_i = \frac{1}{M_i}(\boldsymbol{W}_s - \boldsymbol{W}_i)$ and $\varDelta \boldsymbol{W}_j = \frac{1}{100 M_i}(\boldsymbol{W}_s - \boldsymbol{W}_j)$, where $\boldsymbol{W}_s$ is the weight of the input vector.

Then, SOINN removes the edges whose age is larger than the value of threshold parameter $\mathrm{age}_{\max}$.

After λ learning iterations, the SOINN inserts new nodes into the position where the accumulating error is extremely large. Cancel the insertion if the insertion cannot decrease the error. The insertion here is called within-class insertion because the new inserted node is within the existing class; also, no new class will be generated during the insertion.

To delete the nodes created by noise, SOINN uses the following strategy: if the number of input signals generated so far is an integer multiple of a parameter λ, remove those nodes that have no neighbor or only one topological neighbor.

After LT learning iterations of the first layer, the learning results are used as the input for the second layer. The second layer of SOINN uses the same learning algorithm as the first layer. For the second layer, the similarity threshold is constant; it is calculated using the within-cluster distance and between-cluster distance [5]. With a large constant similarity threshold, different from that of the first layer, the accumulation error for nodes of the second layer will be very high, and within-class insertion plays a major role in the learning process. With a large constant similarity threshold, the second layer also can delete some "noise nodes" that remain undeleted during first-layer learning.

3 Applications with SOINN

After the publishing of original SOINN, we adopted SOINN for some machine learning tasks and some applications.

1. Unsupervised learning
   The original SOINN [5] and its enhanced version [6] are about unsupervised learning. SOINN is used to learn the topology structure of input data, it is able to grow incrementally and to accomodate input patterns of online non-stationary data distribution. It can separate classed with low-density overlap and detect the main structure of clusters that are polluted by noise. It automatically learns number of nodes and structure of the network, reports the number of clusters, and give the typical prototypes of every cluster.
2. Supervised learning
   In [7], a prototype-based classifier based on SOINN is proposed. We firstly use SOINN on every class separately and generate typical prototypes for each class; then we do k-means clustering to tune the results of SOINN; then we adopt a k-Edited Neighbors Classifier like technique to reduce the noise influence of input data; at last, the input data of all classes and the results of noise-reduction part are used to clean the central part of every class and only boundary prototypes are remained. This method automatically learns the number of prototypes needed to determine the decision boundary. For different classes, the learned prototypes may be different. It is robust to noisy training data, and it realized very fast classification.
3. Semi-supervised learning
   In [8], an online incremental semi-supervised learning method based on SOINN is presented. Using labeled data and a large amount of unlabeled data, the proposed method automatically learns the topology of input data distribution with no prior knowledge, it subsequently labels all generated nodes and divide the learned topology structure into sub-structures corresponding to classes. Weights of nodes are used as prototype vectors to realize classification. During the learning, new labeled or unlabeled data is able to incrementally add to the system.
4. Active learning
   In [9], an online incremental active learning algorithm based on SOINN is proposed. It uses SOINN to represent the topology structure of input data, and then separates the generated nodes into different groups and subclusters. It then actively labels some teacher nodes and uses such teacher nodes

to label all unlabeled nodes. It queries the labels of some important samples rather than selecting the labeled samples randomly. It automatically learns the number of nodes and teacher vectors required for a current task. Experiments using artificial data and real-world data show that the proposed method works effectively and efficiently.

5. Associative memory
   Associative memory operating in a real environment must perform well in online incremental learning and be robust to noisy data because noisy associative patterns are presented sequentially in a real environment. In [10], an associative memory is proposed to satisfy these requirements. New associative pairs that are presented sequentially can be learned accurately without forgetting previously learned patterns. The memory size of the proposed method increases adaptively with learning patterns. Therefore, it suffers neither redundancy nor insufficiency of memory size, even in an environment in which the maximum number of associative pairs to be presented is unknown before learning. The proposed associative memory performs as a bidirectional one-to-many or many-to-one associative memory and deals not only with bipolar data, but also with real-valued data.
6. Pattern-based reasoning
   In [11], an architecture for reasoning with pattern-based if-then rules is proposed. By processing patterns as real-valued vectors and classifying similar if-then rules into clusters in the long-term memory, the proposed system can store pattern-based if-then rules of propositional logic, including conjunctions, disjunctions, and negations. It also achieves some important properties for intelligent systems such as incremental learning, generalization, avoidance of duplicate results, and robustness to noise. Experiments show that the proposed method is very effective for intelligent systems solving varying tasks autonomously in a real environment.
7. Other applications
   SOINN is also used for humanoid robots [12], online robot navigation in the real-world [13], word acquisition and grammar learning [14], word grounding [15], and so on. We will not give the detailed discussion here.

4 Conclusion

In this paper, we introduced self-organizing incremental neural network and its application. There are still lots of problems to be solved. Such problems include how to build a theoretical basis of SOINN and how to use SOINN as a data- and feature-extraction method for the solving of large-scale problems. In addition, we need to discuss how SOINN can be used for different types of incremental learning, such as example-incremental, class-incremental, and attribution-incremental learning.

Acknowledgement

This work was supported in part by the Fund of the National Natural Science Foundation of China (Grant No. 60975047, 60723003, 60721002) and Jiangsu NSF grant (#BK2009080).

References

1. Grossberg, S.: Nonlinear neural networks: principles, mechanisms, and architectures. Neural Networks 1, 17–61 (1988)
2. Kohonen, T. (ed.): Self-Organizing Maps. Springer Series in Information Sciences. Springer, Heidelberg (1997)
3. Bauer, H., Villmann, T.: Growing a hypercubical output space in a self-organizing feature map. IEEE Transactions on Neural Networks 8(2), 218–226 (1997)
4. Fritzke, B.: A Growing Neural Gas Network Learns Topologies. In: Neural Information Processing Systems, vol. 7, pp. 625–632. MIT Press, Denver (1995)
5. Shen, F., Hasegawa, O.: An Incremental Network for On-line Unsupervised Classification and Topology Learning. Neural Networks 19(1), 90–106 (2006)
6. Shen, F., Ogura, T., Hasegawa, O.: An enhanced self-organizing incremental neural network for online unsupervised learning. Neural Networks 20, 893–903 (2007)
7. Shen, F., Hasegawa, O.: A Fast Nearest Neighbor Classifier Based on Self-organizing Incremental Neural Network. Neural Networks 21, 1537–1547 (2008)
8. Kamiya, Y., Ishii, T., Shen, F., Hasegawa, O.: An Online Semi-Supervised Clustering Algorithm Based on a Self-organizing Incremental Neural Network. In: The 2007 International Joint Conference on Neural Networks (IJCNN 2007), Orlando, FL, USA (August 2007)
9. Shen, F., Sakurai, K., Kamiya, Y., Hasegawa, O.: An Online Semi-supervised Active Learning Algorithm with Self-organizing Incremental Neural Network. In: The 2007 International Joint Conference on Neural Network (IJCNN 2007), Orlando, FL, USA (August 2007)
10. Sudo, A., Sato, A., Hasegawa, O.: Associative Memory for Online Learning in Noisy Environments Using Self-organizing Incremental Neural Network. IEEE Transactions on Neural Networks 20(6), 964–972 (2009)
11. Shen, F., Sudo, A., Hasegawa, O.: An online incremental learning pattern-based reasoning system. Neural Networks 23, 135–143 (2010)
12. Okada, S., Kobayashi, Y., Ishibashi, S., Nishida, T.: Incremental learning of gestures for human-robot interaction. AI & Society Journal of Knowledge, Culture and Communication (2009)
13. Kawewong, A., Honda, Y., Tsuboyama, M., Hasegawa, O.: Reasoning on the Self-Organizing Incremental Associative Memory for Online Robot Path Planning. IEICE Transactions on Information and Systems E93-D(3), 569–582 (2010)
14. He, X., Ogura, T., Satou, A., Hasegawa, O.: Developmental Word Acquisition And Grammar Learning by Humanoid Robots through A Self-Organizing Incremental Neural Network. IEEE Trans. SMC-Part B 37, 1357–1372 (2007)
15. He, X., Kojima, R., Hasegawa, O.: Developmental Word Grounding through A Growing Neural Network with A Humanoid Robot. IEEE Trans. SMC-Part B 37, 451–462 (2007)