Application of neural network based on self-organizing incremental learning in anti-interference

Shaodong Hu1, Xuechen Wu2, Bo Xue3
1School of Management Science, Chengdu University of Technology, Chengdu, Sichuan, 610059, China
2School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China
3School of Physics, University of Electronic Science and Technology of China, Chengdu, Sichuan, 610054, China

Abstract. In view of the shortcomings of traditional neural network in the actual information fusion process, a data anti-interference method based on self-organizing incremental learning neural network is proposed, which is used to cluster and represent the dynamic input data online without prior knowledge. In this paper, the neuron distribution, dynamic node adjustment, topology representation and denoising process of the network are described in detail. Finally, the multi-source heterogeneous data collected by the sensor can be self-adaptive dimensionality reduction and self-organization learning. At the same time, it has strong robustness to noise data, so that it can continuously learn new patterns in data flow. SOINN model is suitable for supervised learning, associative memory, pattern-based reasoning, manifold learning and other learning scenarios. It can also be extended to the application fields of unbalanced, highly complex and nonlinear big data prediction.

1. Introduction

The demand problems brought by social development make the design of intelligent human-computer interaction system more and more important. If robots can accurately recognize human actions and understand the information expressed by human actions, it will further improve the communication ability between machines and humans. Incremental learning refers to a learning system that can learn new knowledge from new samples and save most of the knowledge that has been learned before. It is similar to human's own learning mode, because in the process of growing up, people learn and receive new things every day, and will not forget what they have learned.

Self-organizing incremental learning neural network (SOINN) is a kind of neural network model based on competitive learning [1, 2], which can do incremental unsupervised learning for data without fixed topology structure. SOINN uses a group of neurons distributed in the feature space to approximate the density distribution of input data. SOINN's learning algorithm is fully incremental, which is not only reflected in the online learning process, but also the incremental learning results. It has the following advantages: (1) it can learn new knowledge from new data; (2) the previously processed data does not need to be repeated Processing; (3) only one training observation sample is seen and learned at a time; (4) most of the previously learned knowledge can be saved while learning new knowledge; (5) the training observation sample is discarded once the learning is completed. Since the adaptive similarity...
threshold is used to identify the input patterns that have not been learned before, the network can dynamically generate new neurons to represent the input patterns without affecting the previous learning results.

Because of these characteristics, SOINN, as a general competitive neural network model, has been applied in various fields, including robot intelligence, computer vision, and anomaly detection and so on. It is also more suitable for all kinds of learning problems, including associative memory, density estimation, and manifold learning and so on. This paper combs and summarizes the basic idea of SOINN and its application in artificial intelligence.

2. The basic principle of SOINN algorithm

SOINN [3] is a two-layer competitive neural network (excluding the input layer), which clusters and represents the input data in a self-organized way. The working process is shown in Figure 1.

![Figure 1](image1.png)

**Figure 1.** Two level competitive learning of SOINN

(1) The first layer network receives the input of the original data and adaptively generates prototype neurons to represent the input data online. These nodes and their connections reflect the distribution and topology of the original data. It should be noted that neurons in SOINN, SOM, GNG networks are different from neurons in common feed forward neural networks. They can be regarded as vectors in a certain feature space, and their weights are coordinate representations of neurons in that space.

(2) The second layer estimates the distance between classes and the distance within classes of the original data according to the results of the first layer network, and then uses the neurons generated in the first layer as input to run the SOINN algorithm again to stabilize the learning results. When there are multiple clusters and noises in the input data, SOINN can still generate reliable neuron nodes to represent each cluster in the input data; at the same time, the topological structure of the sub graph reflects the nature of the original data distribution, and its working effect is shown in Figure 2 below.

![Figure 2](image2.png)

**Figure 2.** Effect demonstration on artificial data set
3. Training process of SOINN algorithm

From the point of view of specific algorithm, the training process of SOINN is mainly divided into four steps: neuron distribution, dynamic node adjustment, topology representation and denoising process. Here are the basic ideas of each part.

3.1. Distribution of neurons

Using a certain number of representative data to approximate the original complete data set, and then making decisions on the new data based on these representative points is the basic idea of the prototype based learning algorithm. These representative data are called prototypes. They usually reflect the distribution information of the original data in some way. For example, each clustering center of K-means is a typical prototype. SOINN is also a prototype based algorithm. Its prototype consists of neurons distributed in the input space. Each neuron represents the most similar input pattern around it. From this point of view, distributed neurons are essentially a vector quantization process.

Vector quantization (VQ) [4] was originally used to compress and code the digital signal, so that the signal can be transmitted with a lower bandwidth. In short, the process of VQ is to use a set of fixed number of codebook vectors \( w_i \) (\( i = 1, 2, \ldots, K \)) the original signal \( x \) is encoded to achieve the purpose of data compression, so as to minimize the expected error value \( E(w) \) of signal reconstruction at the decoding end to:

\[
E(w) = \mathbb{E}[||x - w_i(x)||^2] p(x) \, dx
\]  

Where \( p(x) \) is the probability density function of signal \( x \). \( E(w) \) usually has no closed optimal solution, so it needs to be optimized by iterative algorithm, among which LBG algorithm based on Lloyd-Max condition is the most widely used one. The formula (1) is optimized by using random gradient descent to get the following updated formula, but it usually needs to traverse multiple training data to make the algorithm converge to the local optimal solution:

\[
w_i^{(t+1)} \leftarrow w_i^{(t)} + \alpha(t)(x^{(t)} - w_i^{(t)})
\]  

On the basis of formula (2), a neighbourhood function is added to control the sensitivity of different codebook vectors to the input, which can form a mapping with spatial order. The formula is as follows:

\[
w_i^{(t+1)} \leftarrow w_i^{(t)} + \alpha(t) h(x^{(t)}, W, \sigma)(x^{(t)} - w_i^{(t)})
\]

In general, the existence of \( h \) enables only those codebook vectors closest to the input mode to compete and move to the input data, which may be adjacent in the input space. The result of network convergence is that the output codebook vector and the input pattern form a spatial ordered mapping relationship.

For SOINN, its neuron is equivalent to codebook vector in vector quantization. Every time the data is input, only the nearest neuron and its neighbour nodes in the topology have the chance to move, that is, the topology neighbourhood. SOINN gives each neuron a different learning rate parameter \( (t) \), which has the same effect as that in VQ. This makes the movement of neurons eventually tend to be stable and eventually converge without continuous oscillation. In general, the form of \( (t) \) needs to meet certain constraints to ensure the final learning effect:

\[
\sum_{t=1}^{\infty} \alpha(t) = \infty, \sum_{t=1}^{\infty} \alpha^2(t) < \infty
\]

The learning rate of formula (4) is satisfied, which ensures that each node always maintains a certain learning ability when it is gradually stable.
3.2. Dynamic adjustment
Dynamic adjustment is the key to realize self-organization and incremental learning of SOINN. It makes the weight vector of neuron and topological structure of network dynamically adjust with the arrival of input mode to optimize the expression accuracy of input data. At the same time, it can adapt to the input mode that has not been learned before without affecting the previous learning results.

SOINN defines intra class node insertion and inter class node insertion respectively to achieve these two purposes. In order to reduce the quantization error of neurons adaptively and approximate the distribution of the original data as accurately as possible, the node insertion in the class is mainly used. It records the cumulative quantization error of each neuron, finds out the two nodes with the largest cumulative quantization error among all nodes, inserts a new node between them, updates their cumulative quantization error value by interpolation, and constantly increases the number of nodes. Therefore, SOINN will judge whether the insertion has significantly reduced the quantization error after each node insertion in the class. If not, the insertion will be cancelled.

3.3. Topological representation
Dynamic adjustment of neurons is for incremental learning of data. In addition, SOINN uses competitive Hebb learning rules to establish connections between neurons. At the same time, when the neurons are dense enough, the topological neighborhood of the original data can be perfectly maintained.

Figure 3. Interclass node insertion

For each input data sample, two neurons most similar to the input pattern are found in the feature space, and then they are connected. The bold connection in Figure 3 (b) is the new connection established according to ChL. Considering that the position of the nodes will be adjusted during the running process, the two nodes that are relatively close before may not be activated by an input signal at the end of the algorithm at the same time, and there is no connection between them. Reset the parameter to 0 every time two endpoints of the edge connection are activated at the same time; otherwise, the parameter will be deleted if it exceeds a certain threshold.

3.4. Network denoising
In practical application, random input sequence will also make the results of SOINN unstable, and there is often noise in the input data, and unnecessary nodes are generated. In order to alleviate this problem, we should find and delete some nodes that may not reflect the distribution information of the original data, they are often in the low density area.

So how to judge? In SOINN, the number of times a node becomes the winner and the number of its neighbors are used to determine whether it is in a low data density area. If a node has no or only one
neighbor node, and the number of times it becomes the winner is lower than the average number of current neurons, a certain proportion will be deleted, and all the edges associated with it will be removed.

3.5. Simulation test

The SOINN algorithm is realized by MATLAB software, and the corresponding machine learning model is established. In order to test the performance of the model, the artificial data set is used to simulate and test the machine learning model. Using the rand function of MATLAB software, 50000 random data points are generated and divided into six groups for simulation. The self-organizing incremental learning neural network algorithm can extract the features of sample data and replace the original redundant data with fewer nodes and topology. When the amount of sample data is small, the number of nodes is large, and there is no obvious advantage. However, with the increase of data volume, the number of nodes does not show a linear growth, and the data compression ratio increases gradually. When the two-dimensional data points are 50000, there are only 85 nodes after learning, and the data compression ratio is as high as 588, that is, a large number of nodes can be used to represent the distribution characteristics of data.

The three-dimensional data set generated manually is input into the machine learning model and simulated. The simulation result is shown in Figure 4, which is the simulation result of the three-dimensional data. In the figure, the gray bubble points represent the distribution of the original data, the white red circle represents the nodes learned by the SOINN algorithm, and the blue line represents the topological relationship between them. It can be seen that the model has a good ability to extract features from 3D data. Using the rand function and radon function of MATLAB software, 50000 artificial data sets with noise points are generated, which are divided into six groups for simulation.

![Figure 4. Learning simulation results of 10000 sample points](image-url)
From the simulation results, the neural network model based on self-organizing incremental learning has a very good data extraction ability for three-dimensional data, that is, it can replace the distribution characteristics of a large number of redundant data with fewer nodes and network topology, and has a certain anti-interference ability.

4. Conclusion
SOINN has been applied in many fields, including medical expert system, computer vision, anomaly detection, robot intelligence, etc. For example, in the aspect of artificial intelligence, the robot's learning system can describe the current physical scene with some simple natural languages through the multimodal training process such as image display and voice guidance, and at the same time, it can communicate with people with simple vocabulary; in the aspect of medical diagnosis, the hybrid expert system includes data processing, classification and knowledge extraction To assist doctors in the diagnosis process. This shows that SOINN makes robots have certain autonomy and learning ability in the face of constantly changing external environment.

In general, self-organizing incremental learning is a competitive neural network which can cluster data online and represent topology, and robot intelligence is its main application direction. The ability of incremental learning is very suitable for the training process of robot intelligence. It does not need too many people's intervention, and does not need to specify the number of clusters. As a result, SOINN can get the approximate density distribution and topology information of data without prior knowledge.

References
[1] Shen F, Hasegawa O. An incremental network for on-line unsupervised classification and topology learning. Neural Networks, 2006, 19 (1): 90 - 106.
[2] Shen F, Ogura T, Hasegawa O. An enhanced self-organizing incremental neural network for online unsupervised learning. Neural Networks, 2007, 20 (8): 893 - 903.
[3] Shen F, Hasegawa O. Self-Organizing incremental neural network and its application. In: Proc. of the Int'l Conf. on Artificial Neural Networks (ICANN 2010). Berlin, Heidelberg: Springer-Verlag, 2010. 535 - 540.
[4] Gray RM. Vector quantization. IEEE ASSP Magazine, 1984, 1 (2): 4 - 29.