Load Predicting Algorithm Based on Improved Growing Self-organized Map

Nawaf Alharbe

Abstract With the development of big data and data stream processing technology, the research of load predicting algorithm has gradually become the research hotspot in this field. Nevertheless, due to the complexity of data stream processing system, the accuracy and speed of current load predicting algorithms are not meet the requirements. In this paper, a load predicting algorithm based on improved Growing Self-Organizing Map (GSOM) model is proposed. The algorithm clusters the input modes of the data stream processing system by neural network, and then predicts the load according to its historical load information, optimizes it according to the characteristics of stream processing system, and a variety of strategies are introduced to better meet the load predicting needs of stream processing systems. Based on experimental results, the proposed algorithm achieved higher prediction accuracy rate and speed significantly compared to other prediction algorithms.

Keywords GSOM · Load predicting · Stream processing

1 Introduction

The massive data are generated all the time through a variety of devices around the world such as driverless cars, mobile terminals, humidity sensors, computer clusters and so on. Big data has brought tremendous impact on people's life. In order to meet the requirement of real-time data, a series of data stream processing platform came into being. Data stream processing system [1] provides a way to deal with big data and greatly improves the real-time performance of data processing. Reducing the operating cost as much as possible and ensuring the stable operation of the system has become a research hotspot. Load predicting technology [2] can solve the above problems to a certain extent. Therefore, load predicting has become one of the research hotspots. Thus, the difficulty of load predicting is that data stream in

N. Alharbe (✉)
College of Community, Taibah University, Badr, Kingdom of Saudi Arabia
e-mail: nrharbe@taibahu.edu.sa

J. J. Park et al. (eds.), *Advanced Multimedia and Ubiquitous Engineering*, Lecture Notes in Electrical Engineering 518,
https://doi.org/10.1007/978-981-13-1328-8_5

data stream processing system is temporary compared with traditional processing system. The scale of data to be processed is also complex and unpredictable, and it also needs to meet the requirements of the real-time performance of the stream processing system. Prediction models for load can be divided into linear and nonlinear prediction. The linear prediction mainly includes ARMA [3] model and FARIMA [4] model. The nonlinear prediction mainly includes neural network [5], wavelet theory [6] and support vector machine (SVM) [7]. Due to the uncertainty of the rule of flow processing load fluctuation, the method of nonlinear prediction is more concerned by researchers. The basic principle of the most of the prediction algorithms is based on the existing load time series data [8] and on other historical rules to predict. Recently, some achievements have been made: Box-Jenkins's classic model [9], load predicting by utilizing time series of dynamic load change of processor, prediction of processor behavior by using thread execution time slice in CPU as a parameter. Warren et al. [10] used of job execution time and queue latency as a basis for predictions. Wolski [11] proposed CPU utilization for time-based UNIX systems. The above prediction algorithm has high theoretical value, but did not study the characteristics of the data stream processing system.

In this paper, a load forecasting algorithm based on improved GSOM model is proposed. The rest of this paper is organized as follows. Our proposed algorithm is detailed in Sect. 2. The corresponding comparative experiments are carried out and the experimental results are analysed in Sect. 3. Finally, this paper concludes with Sect. 4.

2 Methodology

2.1 Related Works

Self-organization mapping (SOM) [12] is widely used in pattern recognition as clustering algorithm. The research goal of this paper is to predict the load of stream processing system. Whereas, the traditional SOM consists of three phases: Competition, Cooperation and Adaption processes. In Competition process; the discriminant function is calculated to meet most matching neurons by computing the Euclidean distance using Eq. (1). Where n is the number of output neurons.

$$i(\hat{x}) = argmin_j \|\hat{x} - \hat{w}\|, \quad j = 1, 2, \ldots, n \tag{1}$$

In the cooperation process; the adjacent neurons will cooperate with each other and the weight vectors will be adjusted in the neighborhood of the winning neurons. The Gaussian functions are used using the Eq. (2). The neighborhood function reflects are computed using Eq. (3). Where, σ_0 is the initial neighborhood radius which is generally set to half the output plane and τ is the time constant.

$$h_{i(\hat{x})}(n) = e^{-\frac{d_{i,j}^2}{2\sigma^2(n)}} \tag{2}$$

$$\sigma(n) = \sigma_0 e^{-\frac{n}{\tau}} \tag{3}$$

where in adaptation process started after neighborhood function is determined. The winning vector of neurons in the winning neurons and their topological neighborhoods can be updated using Eq. (4). Where w_j the weight of $_j$ neuron, and t represent the winning vector of neurons.

$$w_j(n+1) = w_j(n) + \eta(n)h_{j,i(x)}(n)\big(X(n) - w_j(n)\big), \quad n = 1, 2, \ldots, T \tag{4}$$

Learning efficiency computed using Eq. (5), where η is a constant greater than 0 and less than 1, and T is the total number of iterations, η(0) is the initial learning efficiency.

$$T_\eta(n) = \eta(0)\left(1 - \frac{n}{T}\right), \quad n = 1, 2, \ldots, T \tag{5}$$

The algorithm has three basic steps after initialization: sampling, similarity matching, and updating. Repeat these three steps until the feature mapping is completed.

2.2 Improved Growing Threshold Setting Method

The load predicting of stream processing system has higher requirements for real-time response. The predicted effect depends not only on the output of the algorithm, but also on the response speed. This paper presents an improved GSOM algorithm. There are improvements in network parameter initialization, clustering prediction mode, new node initialization, and operation efficiency and so on, which can better meet the demand of load predicting of stream processing system.

GSOM determines when to increase a new neuron according to the growth threshold (GT), so the value of GT should be set reasonably. If the threshold is too trivial, the neuron will be added frequently which will increase the training burden. If the threshold is too huge, the prediction of the load will be inaccurate. As the network grows, the addition of neurons should be more and more prudent. Therefore, the value of the threshold should be closely related to the current network condition. Drawing on the general idea of clustering algorithm: the points in the same category should be as close as possible, and the points in different categories can be as far away as possible, A new method to adjust growing threshold dynamically is presented in this paper.

$$\begin{aligned} j &= \arg\min_{j} \left\| x_n - w_j \right\|, \quad j = 1, 2, \ldots, m \\ GT &= \min \left\| w_i - w_j \right\|, \quad i = 1, 2, \ldots, m, \ i \neq j \end{aligned} \tag{6}$$

where j is the number of winner neuron and w_j is the weight of it. w_i is the weight of neuroni's nearest neighbor and GT is set to be the distance of neuron j and its' nearest neighbor i. The main idea of the method is that if vector x_i belongs to a category represented by w_j, then the distance from x_i to w_j is at least less than the distance between w_j and its nearest neighbor. Equation (7) below shows how the growing threshold works:

$$GT < \min \left\| x_n - w_j \right\|, \quad j = 1, 2, \ldots, m \tag{7}$$

where m is the number of competition neurons. The network considers input x as a new input pattern and grows itself only when the distance between x and its' winner neuron j larger than the growing threshold.

2.3 Initial Parameter Optimization

1. Neuron number initialization algorithm

Each time a new input pattern arrives; the network will dynamically add neurons and adjust parameters until it reaches steady. Therefore, if the initial neuron number is too small, it will lead to frequent adding neurons in the training phase, which will affect the response speed of the system. On the other hand, a too large number which will cause excessive death neurons and brings unnecessary interference to the training process. Accordingly, setting up a proper number of initial neurons can accelerate the training process of SOM network. The following algorithm draws lessons from the idea of dichotomy, and calculates the average distance $\mathrm{dist}_{\mathrm{mean}}$ for the input sample set X. If the Euclidean distance $\mathrm{dist}_{\mathrm{ij}}$ between the two input X_i and X_j is smaller than the average distance mean, it indicates that the two inputs are very likely to belong to the same category. By pre-processing the set of input vectors by probabilistic analysis and dichotomy method, a rough number of M is obtained and used as the number of initial neurons. Compared with traditional methods which based on experience or simply choose fixed m, this method can greatly accelerate the training process and reduce the number of iterations.

2. Initialize neurons' weights

The weights of neurons should be initialized first, then they'll be adjusted gradually to reflect the characteristics of the input data set during the training process. Traditional SOM networks used to initialize neurons' weights with random numbers, and the weights generated randomly do not contains any characteristic of the training data. A method which initialize neuron's with typical input vectors is proposed in this paper. As the number of neurons is knows as m, the problem of initializing m neurons' weights is converted to the problem of finding m typical input vectors that can represent the characteristics of their respective categories. The general criterion for clustering problem is to make the distance between nodes in the same category as close as possible, and the distance between nodes of different categories as far away as possible. Therefore, in our work, we use the greedy algorithm to select m vectors from the input data set, which has the farthest distance from each other, then initialize neurons' weights with these vectors.

2.4 Computational Performance Optimization

SOM requires repeated iteration during the training process, and after that, the weights of the whole network need to be adjusted each time a new neuron added in. The prediction is timeliness. Thus the complexity of traditional SOM algorithm is intolerable in load prediction problem of stream processing system. To solve the problem, a caching-based load prediction mechanism is proposed in this section. On the one hand, the new prediction mechanism uses SOM as a classifier to predict load accurately, and on the other hand, it improves the computational efficiency of load prediction. The algorithm improves computational efficiency with the following three methods.

1. New neuron weight vector assignment strategy

The efficiency of network learning process is greatly influenced by the initial value of the network connection weight. In order to speed up the retraining process after a neuron added in, the weight of winning neuron and the input pattern itself are used to assign new neuron node's weight. The weight initialization formula is shown as follows:

$$w_{new} = a * w + b * X_i + c * Random \tag{8}$$

where w_{new} is a linear combination of the winning neuron weight w_i and the new pattern vector X_i. A random quantity is imported in order to ensure that the weight does not bias the current vector X_i too much. According to the experimental result, it works well when a takes 1/5, B takes 3/5 and C takes 1/5. The initial weight of the new neuron need not be very precise, because it will be constantly adjusted the subsequent iteration process, but a rational initial value do help reduce the iterations and make the network stable.

2. Predicting strategy after pattern recognition

When the input vector does not conform to the winning neuron constraints, which means the distance of input vector and its' wining neuron larger than the growing threshold, a new neuron need to be added, but the new empty neurons do not contain any known load information. In order to solve this problem, a prediction mechanism is proposed. When the input vector is considered belongs to a knowing cluster, predict the load according to the historical data of that cluster. Otherwise predict it with linear regression algorithm based on all the historical data. After the real load arrives, add the information to the new neuron.

The linear regression algorithm works as following: For input matrix X, the regression coefficient is stored in the vector w, and the result of the prediction will be given by Y = XTw. In order to make the best prediction, the square error is used to measure the effect:

$$Err = \sum_{i=1}^{m} \left(y_i - x_i^T w\right)^2 \tag{9}$$

The equation can be represented in the form of a matrix as (y − Xw)T(y − Xw), find the derivative of w and make it equal to zero, solve the equation and get w as follows:

$$\hat{w} = (X^T X)^{-1} X^T y \tag{10}$$

With the weight vector $\hat{w}$ and the input data set X, the predicted load can be given by $Y = X^T \hat{w}$.

3. Cold backup strategy

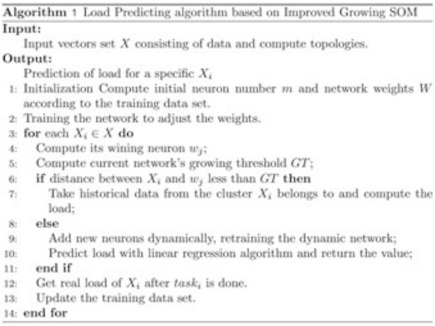
As is said above, adding new neurons will cause the SOM network to reiterate to adjust parameters, and the high time complexity of the iteration process can not meet the real-time requirement of stream processing system. To this end, the SOM cold backup strategy is proposed, System maintain two SOM networks of dynamic and static. The static network is responsible for receiving input and predicting the load, and the dynamic network is responsible for adding new neurons and retraining to make the network stable. The synergy process of the two networks is as follows:

(1) In the initial stage, two networks are the same.
(2) When an input vector X_i comes, the static SOM calculates the winning neuron and compare it with the threshold GT. If the input belongs to an existing cluster, take the historical data and predict load for input $\mathrm{X_i}$.
(3) If X_i belongs to a new cluster, the dynamic SOM network performs the operation of adding neurons and retrains the network parameters. The static network remains the same, using linear regression algorithm to calculate the results.
(4) After retraining process of the dynamic SOM completed, replicate the dynamic network to replace the static classifier.

The process of training iteration is responsible for the dynamic SOM network. This strategy can avoid the problem of failing to meet the real-time requirement of the stream processing system because of the network updates.

2.5 Implementation of the Improved GSOM Based Load Predicting Algorithm

The existing GSOM algorithm can meet the requirement of dynamic adding of neurons and recognizes new classification of input vectors. However, the algorithm iterates frequently, and the computing speed can not meet the requirements of the real-time performance of the stream processing system. In order to recognized input task' cluster and predict its' load requirement accurately and quickly, this paper proposes a LP-IGSOM (Load Predicting based on Improved Growing Self-Organizing Map) algorithm. Compared with the existing GSOM, the LP-IGSOM has improvements in the initializing neuron numbers, optimizing calculate efficiency and some other ways. The specific process of the LP-IGSOM algorithm is showing as Algorithm 1:



Algorithm 1 Load Predicting algorithm based on Improved Growing SOM

```
Input:
    Input vectors set $X$ consisting of data and compute topologies.
Output:
    Prediction of load for a specific $X_i$
Initialization Compute initial neuron number $m$ and network weights $W$
    according to the training data set.
Training the network to adjust the weights.
for each $X_i \in X$ do
    Compute its wining neuron $w_j$;
    Compute current network's growing threshold $GT$;
    if distance between $X_i$ and $w_j$ less than $GT$ then
        Take historical data from the cluster $X_i$ belongs to and compute the
        load;
    else
        Add new neurons dynamically, retraining the dynamic network;
        Predict load with linear regression algorithm and return the value;
    end if
    Get real load of $X_i$ after $task_i$ is done.
    Update the training data set.
end for
```

(1) Initialization phase:

   (a) According to the known input mode, calculate the rough class number m, which used to initialize the number of neurons.
   (b) Select m input vectors with the largest distance from each other and initialize *m* neuron weights.
   (c) According to current network status, calculate the growing threshold.

(2) Growth phase:

   (a) Add input to the network.
   (b) Use the Euclidean distance to find the winning neuron in the traditional SOM algorithm.
   (c) Determine whether the winning neuron is greater than the threshold GT, if not, skip to step f.
   (d) If the winning neuron is a boundary node, add a neuron and initialize the weight of the new neuron using the current input mode X, the winning neuron weight W, and the random quantity. If not, skip to step f.
   (e) Reset the learning rate to the initial value and adjust the neighborhood to the initial value.

(f) Update the neighborhood vector of neurons.

(g) Repeat step b to step f until the clustering effect stabilizes for the existing data.

(3) Prediction phase:

(a) Find the winning neuron, if there is no need to add new nodes, take all the known loads from the winning neurons and calculate the average as the result of load predicting.

(b) If a node needs to be added, the static network uses linear regression to predict the load on the input mode, and the dynamic SOM network adds nodes to re-train.

(c) Visit the real information of the load and add to the new neurons.

3 Experiment Result

The experimental data in this paper simulates the data set proposed by the pavement sensor network. The statistical data arrival speed is 5000 pieces per second. Due to the particularity of data stream, little research has been done on load predicting. Here we choose the classic linear regression prediction algorithm and the classical clustering algorithm K-means for comparison, respectively predict the load on the data stream sent by the sensor network and compare it with the real load situation. Experiment related parameters are set as follows; The maximum learning rate parameter is 0.9, the minimum learning rate parameter is 1E-5 and The number of iterations of training neural network is 1000. The initial neighborhood radius is 5.

Figure 1 shows the actual changes of the load over time during the operation of the stream processing system. Under standard data source and fixed computing topology, samples are taken every two minutes from 0 to 20, and the prediction of the calculated load by GLP-SOM and linear regression, k-means clustering is recorded and compared with the real load. Figure 2 shows the actual load curve and the load curve predicted by each algorithm. Under fixed computing topology,

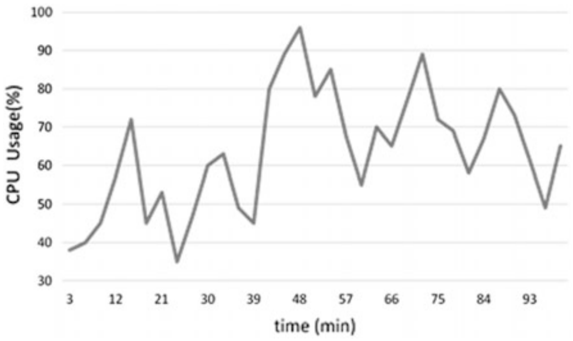


Fig. 1 The real load situation of the nodes

the input modes are known modes and no new mode enters the system. Therefore, linear regression, K-means and GLP-SOM algorithms are better able to predict the load, as shown in Fig. 2, The predicted curve is closer to the actual load curve. In this case, the main effect of the affection prediction is the performance of the algorithm and the fluctuation of the data source. Among them, the MSE of LR algorithm is 81, the errors of k-means and GLP-SOM are smaller, which are 32.7 and 21.5 respectively. LR algorithm has a relatively poor prediction effect when the data source fluctuates greatly, while the prediction effect based on clustering algorithm is relatively stable.

On the basis of the existing calculation rules, new calculation topologies are continuously generated, corresponding to new calculation modes. In Fig. 3, the face of the new calculation rules, the data source and the calculation topology have no prior knowledge in the historical data. With the method of linear regression prediction, the situation can not be handled and predicted well. The prediction error is too large to reach 322.5. Among the three classifiers algorithms, K-means of fixed clustering has the worst prediction effect, and the MSE reaches 383.9. Due to the fixed value of K, the new input mode will be forcibly classified into existing clusters, and the current clustering characteristics will be affected. Making k-means no matter dealing with simple mode or new mode, the prediction effect has a greater error. The proposed algorithm based on LP-IGSOM can identify and dynamically grow neurons in the face of new input. The overall prediction effect is more accurate, and the actual load error is smaller, MSE is 77.6. The experimental results show that the load forecasting algorithm based on the proposed GSOM model can effectively deal with the new input mode. The accuracy and speed of load predicting are superior to other methods.

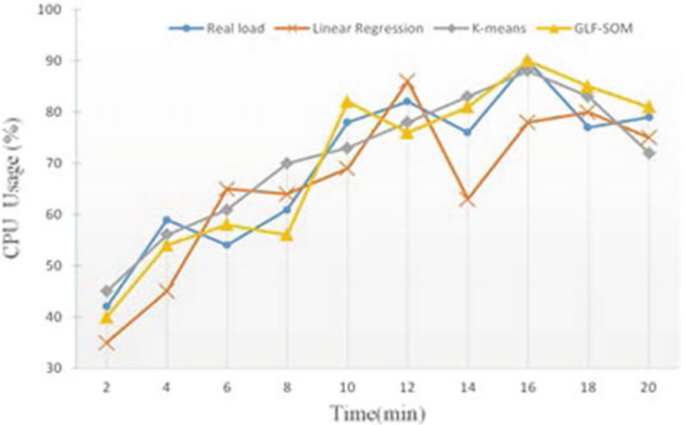


Fig. 2 The load predicting curve based on standard data source and fixed computing topology

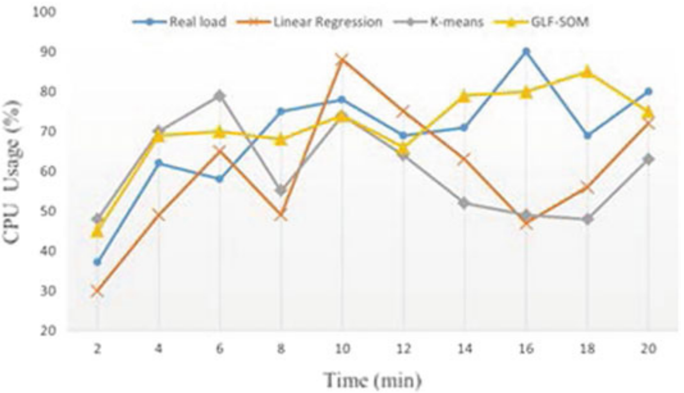


Fig. 3 The load predicting curve based on standard data source and customized computing topology

4 Conclusion

In this paper, we propose an effective load prediction algorithm based on the improved GSOM model for data stream processing system. Compared with other traditional predicting algorithms, we optimize the GSOM algorithm for the complex features of the stream processing system. The proposed load prediction algorithm LP-IGSOM achieved higher prediction accuracy rate and speed efficiency with a significant improvement than the traditional load predicting algorithms.

References

1. Salehi A (2010) Design and implementation of an efficient data stream processing system
2. Moghram I, Rahman S (1989) Analysis and evaluation of five short-term load forecasting techniques. IEEE Trans Power Syst 9(11):42–43
3. Riise T, Tjozstheim D (2010) Theory and practice of multivariate ARMA forecasting. J Forecast 3(3):309–317
4. Shu Y, Jin Z, Zhang L, Wang, L (1999) Traffic prediction using FARIMA models. In: IEEE international conference on communications, vol 2, pp 891–895
5. Haykin S, Network N (2001) A comprehensive foundation. Neural Netw
6. Wen HY (2004) Research on deformation analysis model based upon wavelet transform theory. PhD Thesis, Wuhan University
7. Cristianini N, Shawe-Taylor J (2004) An Introduction to support vector machines and other kernel-based learning methods. Publishing House of Electronics Industry, Beijing, China
8. Hagan MT, Behr SM (1987) The time series approach to short term load forecasting. IEEE Trans Power Syst 2(3):785–791
9. Lowekamp B, Miller N, Sutherland D, Gross T, Steenkiste P, Subhlok J (1998) A resource monitoring system for network-aware applications. In: Proceedings of the 7th IEEE international symposium on high performance distributed computing (HPDC)

10. Smith W, Wong P, Biegel, BA (2001) Resource selection using execution and queue wait time predictions. NASA Ames Research Center TR NAS
11. Wolski R, Spring N, Hayes J (2000) Predicting the CPU availability of time-shared unix systems on the computational grid. Clust Comput 3(4):293–301
12. Vesanto J, Alhoniemi E (2000) Clustering of the self-organizing map. IEEE Trans Neural Netw 11(3):586