

A new ART-counterpropagation neural network for solving a forecasting problem

Tzu-Chiang Liu*, Rong-Kwei Li

*Department of Industrial Engineering and Management, National Chiao-Tung University,
1001 Ta Hsueh Road, Hsinchu, Taiwan, ROC*

Abstract

This study presents a novel Adaptive resonance theory-Counterpropagation neural network (ART-CPN) for solving forecasting problems. The network is based on the ART concept and the CPN learning algorithm for constructing the neural network. The vigilance parameter is used to automatically generate the nodes of the cluster layer for the CPN learning process. This process improves the initial weight problem and the adaptive nodes of the cluster layer (Kohonen layer). ART-CPN involves real-time learning and is capable of developing a more stable and plastic prediction model of input patterns by self-organization. The advantages of ART-CPN include the ability to cluster, learn and construct the network model for forecasting problems. The network was applied to solve the real forecasting problems. The learning algorithm revealed better learning efficiency and good prediction performance.

© 2004 Elsevier Ltd. All rights reserved.

Keywords: Adaptive resonance theory; Counterpropagation; Neural network

1. Introduction

Many neural networks are used for solving forecasting problems. The forecasting network model is a supervised learning algorithm on a neural network. The network models include the backpropagation (BP), radial basis function (RBF) and counterpropagation network (CPN) models (Chun & Kim, 2004; Kim, Jeong, & Lee, 2003; Shi, Xu, & Liu 1999). The counterpropagation network was introduced by Hecht-Nielsen (1988). CPN was designed to provide an efficient learning algorithm for solving the function approximation problem $y=f(x)$ and the forecasting problem (Chang & Chen, 2001). The full CPN works best only when the inverse function f^{-1} exists. The forward-only CPN is designed to approximate $y=f(x)$ when f^{-1} does not need. The forward-only CPN consists of three layers: input, cluster (Kohonen), and output (Grossberg) layers. The learning of CPN can be split into two stages that combine unsupervised and supervised learning. During the first stage,

the input vectors are clustered and the weight of cluster nodes is determined. During the second stage, the weights from the cluster nodes to the output nodes are adapted to produce the desired response (target output). The supervised learning reduces the errors between the CPN outputs and the desired target.

Adaptive resonance theory (ART) was developed by Carpenter and Grossberg. ART nets are a famous unsupervised learning algorithm. This algorithm can automatically find the adaptive clusters based on training patterns. The ART net clustering result is affected by a lower change value of the vigilance parameter. CPN is a supervised neural network that based on the Kohonen learning vector quantization. The learning vector quantization algorithm depends on the approximately optimal number of codebook vectors assigned to each cluster and their initial weights (Kohonen, Hynninen, Laaksonen, & Torkkola, 1995). This study proposes a new neural network called ART-CPN. This network uses the vigilance parameter to generate the cluster layer. In the train process, the learning algorithm sets the vigilance parameter of the input layer (ρ_x) to generate new nodes of cluster layer. The net can automatically create the nodes of the cluster layer and their initial weights.

* Corresponding author. Tel.: +886-4-23924505-7626; fax: +886-4-23934620.

E-mail address: liuzc@chinyi.nctit.edu.tw (T.-C. Liu).

Moreover, the weights of each layer are adjusted based on the CPN leaning rule.

This study introduces a new neural network for solving forecasting problems using a modified CPN algorithm. ART-CPN was applied to forecast the Box-Jenins furnace data and the lead frame dimension in the etching process for semiconductors. Compared with the conventional neural network, that is the standard backpropagation network (BPN). ART-CPN showed better learning efficiency and good prediction performance for solving forecasting problems.

The rest of this paper is organized as follows. Section 2 describes ART and CPN neural network architectures and the learning algorithm. Section 3 then describes ART-CPN neural network. Subsequently, Section 4 examines the performance of proposed method by computer simulation on benchmark Box-Jenins furnace data and predicting the lead frame dimension. Finally, Section 5 discusses the results and draws conclusions.

2. Adaptive resonance theory and forward-only conterpropagation network

Adaptive resonance theory was developed by Carpenter and Grossberg. ART nets are designed to control the degree of similarity of patterns place on the same cluster unit. The system is sufficiently stable against noise to enable learning, and is sufficiently plastic to learn new input vectors without affecting already learned results. ART networks can develop stable and plastic clustering of arbitrary sequences of input patterns by self-organization. Upon receiving an input pattern, the network attempts to categorize it by first comparing it against the stored weight vectors of existing categories. If a category with the required matching level (vigilance parameter) is identified, then the network training enters a so-called resonant state, and learns by modifying its weight vectors in the learning process. ART has since led to an evolving series of real-time neural network models for unsupervised and supervised leaning. These neural models are capable of learning stable recognition categories in response to arbitrary input sequences with either fast or slow learning (Yang, Han, & Kim, 2004). Model families include ART1 (Carpenter & Grossberg, 1987a), which can stably learn to categorize binary input patterns presented in an arbitrary order; ART2 (Carpenter & Grossberg, 1987b), which can learn to categorize either analog or binary input patterns; ART3 (Carpenter & Grossberg, 1990), which can carry out parallel search, or hypothesis testing of distribution recognition code in a multilevel network hierarchy and Fuzzy ART (Carpenter, Grossberg, & David, 1991) developed herein generalizes ART1 as being capable of learning stable recognition categories in response to both analog and binary input patterns.

The forward-only conterpropagation network is a combination of a portion of the Kohonen self-organizing

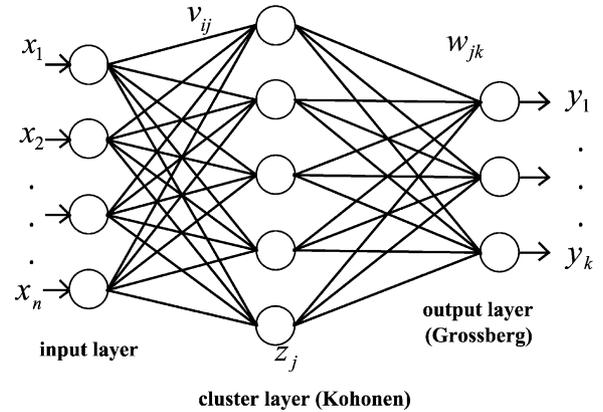


Fig. 1. Forward-only conterpropagation.

map and the output layer. Fig. 1 illustrates the architecture of the CPN, which appears to be same that of the backpropagation net. The net consists of three layers: input layer, cluster layer (Kohonen layer) and output layer (Grossberg layer). The training procedure for the CPN comprises two steps. First, an input vector is presented to the input node. The nodes in the cluster layer then compete (winner take all) for the right to learn the input vector. The weights of the network are adjusted automatically during the learning process. Unsupervised learning is used in this step to cluster the input vector to separate distinct clusters of input data. Second, the weight vectors between the cluster and output layers are adjusted using supervised learning to reduce the errors between the CPN outputs and the corresponding desired target outputs.

During the first step, the Euclidean distance between the input and weight vectors is calculated. The winner node is selected based on comparing the input vector $X(x_1, x_2, \dots, x_n)^T$ and the weight vectors $v_{ij}(w_{1j}, w_{2j}, \dots, w_{nj})^T$. The winning node z_j has the weight vector $w_{jk}(w_{1j}, w_{2j}, \dots, w_{nj})^T$, winner-take-all operation that permits this cluster node J to be the most similar to the input vector. The weights of the cluster node J are adjusted. The weight vector of the winner is updated according to

$$v_{ij}^{\text{new}} = v_{ij}^{\text{old}} + \alpha(x_i - v_{ij}^{\text{old}}) \quad (1)$$

where α denotes the learning rate and x_i represents the i th node of input layer.

After training the weights from the input layer to the cluster layer, the weights from the cluster layer to output layer are trained. Each training pattern inputs the input layer, and the associated target vector is presented to the output layer. The competitive signal is a binary variable, assuming a value of 1 for the winning node and a value of 0 for other nodes of the cluster layer. Each output node k has a calculated input signal w_{jk} and target output y_k . The weights between the winning cluster node and the output layer nodes are updated as follows:

$$w_{jk}^{\text{new}} = w_{jk}^{\text{old}} + \beta(y_k - w_{jk}^{\text{old}}) \quad (2)$$

where w_{jk} denotes the weights from the cluster layer to output layer, and β represents the learning rate. The competitive signal of cluster layer z_j is computed by

$$z_j = \begin{cases} 1 & \text{if } j = J, J \text{ is winning node} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

and the output node k is given by

$$\hat{y}_k = \sum_{j=1}^p w_{jk} z_j \quad (4)$$

\hat{y}_k is the CPN k th computed output.

The CPN classifies the input vector to most similar cluster nodes, and then outputs the prediction result. The learning speed of CPN is fast compared to other neural networks owing to the use of the efficient learning algorithm for solving forecasting problems. The CPN can compress the m input patterns to p clusters where in general $p < m$. The adaptive p cluster nodes determine the accuracy of the network output. The next paragraph develops a new ART-CPN network to automatically generate the adaptive nodes of the cluster layer. The network was applied to solve the real forecasting problems.

3. ART-CPN neural network

Adaptive resonance theory nets are designed to allow the user to control the degree of similarity of patterns placed on the same cluster. The adaptive resonance theory algorithm proposed by Grossberg is a special neural network that can cluster the training patterns. Meanwhile, the CNN-ART algorithm (Lin & Yu, 2003) can dynamically generate the nodes of the cluster layer. The appropriate initial weight vectors (codebook vectors) can be obtained in contrast to the conventional unsupervised learning algorithms. The ART-CPN combines the adaptive resonance theory and counterpropagation network to develop a new prediction network model. The algorithm redesigns the relative similarity between the input vector and the weight vectors for a cluster node. The vigilance parameters make the network automatically generate the nodes of the cluster layer and adaptive initial weights between the input layer, cluster layer and output layer. The ART-CPN algorithm can derive the Kohonen and Grossberg learning rule for updating the weights of the winning node. The network has fast learning speed and good prediction performance.

3.1. Similarity and architecture

Pattern recognition systems are designed to find the similarly vector from random training vectors. From the geometric perspective, the similarity of two vectors is the distance metric of the two different vectors (patterns). The p -norm metric is commonly used, and was defined as

follows (Han & Kamber, 2001)

$$X = [x_1, x_2, x_3, \dots, x_n]^T \text{ and } Y = [y_1, y_2, y_3, \dots, y_n]^T$$

p -norm metric

$$\|X - Y\| \equiv \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}, \quad 1 \leq p \leq \infty. \quad (5)$$

The following conditions must be satisfied:

$$C1 : d(x, y) \geq 0 \quad (6)$$

$$C2 : d(x, y) = d(y, x) \quad (7)$$

$$C3 : d(x, y) \leq d(x, z) + d(z, y) \quad (8)$$

The p -norm is generally used to determine whether two patterns are the same class. The distance between two vectors is as follows (Han & Kamber, 2001):

$$\begin{aligned} p = 1, \|X - Y\|_1 \\ = \sum_{i=1}^n |x_i - y_i|, \text{ is the Manhattan distance.} \end{aligned} \quad (9)$$

$$\begin{aligned} p = 2, \|X - Y\|_1 \\ = \sqrt{\sum_{i=1}^n |x_i - y_i|^2}, \text{ is the Euclidean distance.} \end{aligned} \quad (10)$$

The competition learning law uses the Euclidean distance to determine the winner. The distance calculates the Euclidean distance between the input vector and weight vectors, and the winner unit whose weight vector has the smallest Euclidean distance from the input vector. ART-CPN uses the mean of Manhattan distance to calculate the similarity between the input and weight vectors. Consider m unlabeled training patterns to have n -dimensional attributes using a set of the vectors (X_1, X_2, \dots, X_m) , the similarity between two vectors X_1 and X_2 is calculated as follows:

$$\begin{aligned} X_1 = [x_{11}, x_{12}, x_{13}, \dots, x_{1n}]^T \text{ and} \\ X_2 = [x_{21}, x_{22}, x_{23}, \dots, x_{2n}]^T \text{ have } n - \text{dimensional attributes.} \end{aligned}$$

Distance components of two vectors for x_{11}, x_{21} :

$$\begin{aligned} D_{s(x_{11}, x_{21})} = \frac{|x_{11} - x_{21}|}{\max x_{m1} - \min x_{m1}}, \\ x_{11}, x_{12} \subset X_1, X_2 \text{ for first attribute of } X_1, X_2 \end{aligned} \quad (11)$$

Similarity between two vectors $(V_{s(X_1, X_2)})$ between X_1 and X_2 :

$$\begin{aligned} D_{s(X_1, X_2)} = \frac{\sum_{i=1}^n D_{s(x_i, x_{2i})}}{n}, \\ \text{the mean of } D_{s(X_1, X_2)} \text{ for } n - \text{dimensional vector.} \end{aligned} \quad (12)$$

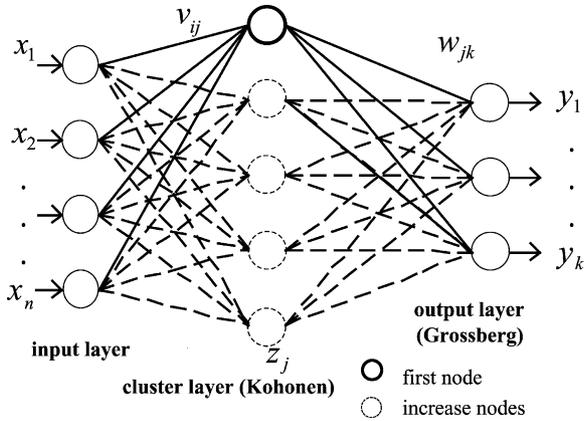


Fig. 2. The architecture of ART-CPN.

$$V_{s(x_1, x_2)} = 1 - D_{s(x_1, x_2)},$$

$$V_s \text{ is defined as the similarity between } X_1 \text{ and } X_2. \quad (13)$$

The major difference between ART and unsupervised neural networks is the vigilance parameter (ρ). The ART defines the similarity between a new pattern and a stored pattern. This similarity is compared to ρ as a measure to ensure that the new pattern is properly classified or that a new cluster is generated. ART-CPN defines the vector similarity (V_s) that $V_s \geq \rho$ creates the new node for the CPN cluster layer. Fig. 2 shows the architecture of ART-CPN network. The net develops a new method to generate the adaptive nodes of the cluster layer. The first input vector (first v_{ij}) and target vector (first w_{jk}) are used directly to establish the initial cluster node. The similarity of the vector ($V_{s(x_j)}$) is calculated between the input vector (X) and weight vector (v_{ij}). The maximum $V_{s(x_j)}$ is the winning node. Moreover, the winning node is index J . ρ_X is the vigilance parameters of the weight vectors (v_{ij}). If the $V_{s(x_j)} < \rho_X$ is true, the cluster nodes are added one node to cluster layer. Additionally, if $V_{s(x_j)} \geq \rho_X$ is true to represent the input vector that belongs to weight vector (same cluster), the Kohonen learning algorithm adjusts the weights of v_{ij} and the Grossberg learning algorithm adjusts the weights of (w_{jk}). The network uses real-time learning to generate dynamically the nodes of cluster layer in the training process.

3.2. The ART-CPN algorithm

The training procedure for the forward-only counter-propagation net includes two steps in learning process. The ART-CPN simultaneously trains the weights of the input layer, cluster layer and output layer. An input vector (X) presents to the cluster node that the $V_{s(x_j)}$ of the weight vector (v_{ij}) is calculated. Each training vector is presented to the input layer, and the associated target vector is presented to the output layer. The nodes in the cluster layer compete (winner-take-all) for the right to learn the input vector. The maximum $V_{s(x_j)}$ is the winning node (call its index J).

The winning node sends a signal of 1 to the output layer. Each output node k has a calculated input signal w_{jk} and target vector. The learning rule updates the weights of the winning nodes. Meanwhile, the learning rule updates the weights from the input layer to the cluster nodes:

$$v_{ij}^{\text{new}} = v_{ij}^{\text{old}} + \alpha(x_i - v_{ij}^{\text{old}}) \quad (14)$$

J denotes the winning node.

The learning rule updates the weights from the cluster nodes to the output layer:

$$w_{jk}^{\text{new}} = w_{jk}^{\text{old}} + \beta(y_k - w_{jk}^{\text{old}}) \quad (15)$$

The competitive signal of cluster layer z_j is computed by

$$z_j = \begin{cases} 1 & \text{if } j = J, J \text{ is winning node} \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

The learning rule for the weights from the cluster nodes to the output nodes can be expressed using the delta rule:

$$w_{jk}^{\text{new}} = w_{jk}^{\text{old}} + \beta z_j (y_k - w_{jk}^{\text{old}}) \quad (17)$$

The training of the weights from the input nodes to the cluster nodes continues at a low learning rate with gradually reducing learning rate for the weights from the cluster nodes to the output nodes. The nomenclature used is as follows:

- $X(x_1, \dots, x_n)$ input vector.
- $Y(y_1, \dots, y_k)$ target vector.
- $V_{s(x_j)}$ similarity of vector between the input and weight vectors of cluster node j .
- ρ_X vigilance parameter of input vector (0.8–0.9).
- C learning epochs.
- α, β learning rate (0.04–0.1).
- v_{xj}, w_{jk} weights of the winner cluster nodes (J).

Table 1 lists a pseudo-code of the ART-CPN algorithm. Setting the vigilance parameter can generate the number of the cluster nodes. If the vigilance is set to be high, then it gathers a large number of cluster nodes. After training, the weights of the cluster nodes are distributed in a statistically optimal manner that improves the accuracy performance. The learning speed of ART-CPN is extremely fast due to the one step learning process and the efficient learning algorithm.

In the testing process, only input data is required for the network model to operate when the ART-CPN is used for the prediction. The application procedure for ART-CPN is as follows:

- Step 1. Present input vector X .
- Step 2. Compute $V_{s(x_j)}$;
Find the node J that is maximum $V_{s(x_j)}$.
- Step 3. Set the activations of output nodes (\hat{y}_k):

Table 1
The ART-CPN algorithm

Step 1. Set the initialize weights (first input vector and first target vector); initialize learning rate (α, β) ; $0 \leq \rho_X \leq 1$, number of epochs

Step 2. While stopping condition is false, do steps 3–8

Step 3. For each input vector $X(x_1, x_2, \dots, x_n)$ and target vector $Y(y_1, y_2, \dots, y_k)$, do Steps 4–8

Step 4. Compute $V_{s(X_j)}$ of each cluster node
Find the winning cluster node (J) that is a maximum $V_{s(X_j)}$, called its index J

Step 5. If $V_{X_j} \geq \rho_X$ update w_{ij} using

$$v_{ij}^{\text{new}} = v_{ij}^{\text{old}} + \alpha(x_i - v_{ij}^{\text{old}}), \quad i = 1, \dots, n.$$

$$w_{jk}^{\text{new}} = w_{jk}^{\text{old}} + \beta(y_k - w_{jk}^{\text{old}}), \quad k = 1, \dots, k.$$

Step 6. The maximum $V_{s(X_j)} < \rho_X$ then adds one cluster node ($j = j + 1$).

$$v_{ij}^{\text{new cluster node}} = x_i (i = 1, \dots, n), \text{ the weights between input layer and cluster layer.}$$

$$w_{jk}^{\text{new cluster node}} = y_k (i = 1, \dots, k), \text{ the weights between cluster layer and output layer.}$$

Step 7. Reduce learning rate

Step 8. Test stopping condition
The condition may specify a fixed number of epochs or the learning rate reaches a small value sufficiently

The competitive signal of cluster layer z_j is computed by

$$z_j = \begin{cases} 1 & \text{if } j = J, J \text{ is winning node} \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{y}_k = \sum_j z_j w_{jk} \quad (18)$$

4. Application of the ART-CPN to solve the forecasting problems

The above methodology is applied to two forecasting problems. The first problem is the benchmark Box and Jenkins gas furnace data (Lin & Cunningham, 1995). Meanwhile, the second problem is an application of predicting the lead frame dimension for semiconductors. The performance of ART-CPN and BPN are evaluated and compared using the root-mean-square error (RMSE) and the P_1 performance index. The root mean of the square errors of prediction are calculated from the target output (y_i) and predicted values (\hat{y}_i) according to the equation:

$$RMSE = \sqrt{\frac{\sum_{i=1}^m (y_i - \hat{y}_i)^2}{m}}, \text{ for } m \text{ training patterns} \quad (19)$$

P_1 is calculated based on the target output and predicted values according to the equation (Lin & Cunningham, 1995):

$$P_1 = \frac{\sqrt{\sum_{i=1}^m (y_i - \hat{y}_i)^2}}{\sum_{i=1}^m |y_i|}, \text{ for } m \text{ training patterns} \quad (20)$$

4.1. Box and Jenkins gas furnace data

The Box and Jenkins gas furnace data were obtained from the literature (Lin & Cunningham, 1995). The process is a gas furnace with a single input (gas flow rate) $u(t)$ and single output (CO_2 concentration) $y(t)$. The dataset considers the variable $y(t-1), y(t-2), \dots, y(t-4), u(t-1),$

$u(t-2), \dots, u(t-6)$ as input nodes. This investigation trained 250 data points and predicted the next 40. The ART-CPN sets the $\rho_X = 0.92$, $\alpha = \beta = 0.04$ and $C = 6$ for network training. It took 1500 iterations to train. The forecasting performance measure was $RMSE = 0.65722$, $P_1 = 0.00077$ on the training set, and $RMSE = 1.61294$, $P_1 = 0.00473$ on the test set. The training data is used to train the back-propagation network. Training the network model required 15,000 iterations. The BPN performance measure was $RMSE = 0.4284$, $P_1 = 0.0005$ on the training set and $RMSE = 1.4795$, $P_1 = 0.00436$ on the test set.

4.2. Forecasting the dimension of lead frame for semiconductor

The electrical industry is rapidly developing, creating high demand for IC production. Etched semiconductor lead frame is the basic material used in IC packaging. The dimensions of the pilot hole are generally required to be highly precise in the lead frame manufacturing. Fig. 3 shows the dimension of the pilot hole of the etching lead frame. The photo etching process must control the dimension of the pilot hole, and records the manufacturing parameters of the etching machine and also the inspection data. These data were used to construct the ART-CPN network model. The network model forecasts the dimension of the pilot hole to control the dimension on target value in the etching process. The model effectively maintains process stability and supports the adjustment parameters of

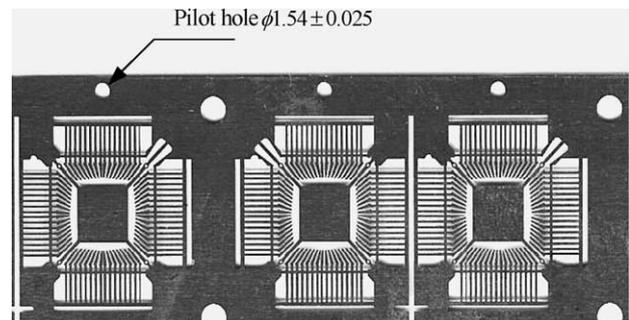


Fig. 3. The pilot hole of the lead frame.

Table 2

The training data is used to train the neural network

Be	PH	ORP	ET	Speed	Roll number	$y(t-1)$	$y(t-2)$	$y(t-3)$	$y(t-4)$	$y(t)$
41.70	486.90	534.50	50.10	1905	97.00	1551.40	1549.60	1551.00	1549.20	1550.20
41.70	488.90	534.60	50.20	1905	97.00	1550.20	1551.40	1549.60	1551.00	1551.40

the subsequent etching process. The input parameters that significantly influence the dimensions of the pilot hole during the etching process must be identified. The input data is mapped onto the output to the desired accuracy. Ten input parameters are used to construct the forecasting model. The input parameters are identified as follows:

Manufacture data of etching process:

1. $Be(t-1)$: the Blaume value of etching solution in time period $t-1$.
2. $PH(t-1)$: the PH value of etching solution in time period $t-1$.
3. $OPR(t-1)$: the ORP value of etching solution in time period $t-1$.
4. $ET(t-1)$: the temperature of the etching solution ($^{\circ}C$) in time period $t-1$.
5. $Speed(t-1)$: the speed of the material (mm/min) in time period $t-1$.
6. $RON(t-1)$: the etching open roller number in time period $t-1$.

Inspection data:

7. $y(t-1)$: the sample mean the dimension of the pilot hole in time period $t-1$.
8. $y(t-2)$: the sample mean the dimension of the pilot hole in time period $t-2$.
9. $y(t-3)$: the sample mean the dimension of the pilot hole in time period $t-3$.
10. $y(t-4)$: the sample mean the dimension of the pilot hole in time period $t-4$.

Table 2 illustrates the input data for training process. $y(t)$ predicts the sample dimension of the pilot hole in time period t . The 72 etching process data is used to establish the forecasting network model. Network training took 1008 iterations using the ACT-CPN. Moreover, the ART-CPN forecasting performance measure was $RMSE=1.5525$, $P_1=0.00012$ on the training set. Network training required 100,000 iterations using BPN. The performance measure used on the training set was $RMSE=1.65795$, $P_1=0.00013$.

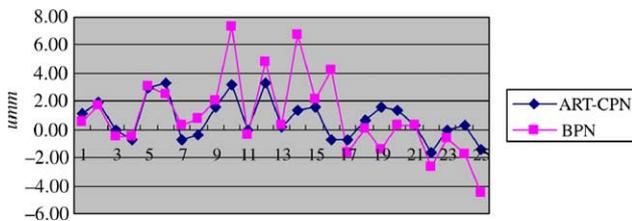


Fig. 4. The predicting errors of ART-CPN and BPN.

This study used the trained network model to predict 25 samples of the pilot hole in the etching process. Fig. 4 shows the prediction errors for the test process. The maximum prediction error is $ART-CPN=4 \mu m$ and $BPN=7 \mu m$. Moreover, the performance measure was $RMSE=1.60609$, $P_1=0.00020$ for ART-CPN and $RMSE=2.86667$, $P_1=0.00037$ for BPN. The pilot hole has a tolerance of $25 \mu m$. The ART-CPN network can precisely forecast the dimension of the pilot hole according to manufacturing parameters and inspection data in the etching process. The algorithm provides good learning efficiency and prediction performance to improve the lead frame quality in the etching process.

5. Conclusions

This investigation proposed a new neural network for solving the forecasting problem. ART-CPN uses the vigilance (ρ_x) to generate the cluster layer nodes. The adaptive cluster nodes can enhance the traditional CPN performance. The net successfully applies to predict the gas furnace data and the lead frame dimension in the forecasting system. A good forecasting performance could improve the lead frame quality in the etching process. ART-CPN requires the one step learning process. The learning speed is faster than the CPN and BPN. ART-CPN based on Adaptive Resonance Theory and CPN that is supervised real-time learning by a self-organizing neural network. If the vigilance is fixed, there is no need to retrain all patterns when adding the new training patterns. In the dynamic database, the real-time learning algorithm can adapt the environmental change to produce a new network model for new training patterns. The learning characteristic is critical for dynamic forecasting problems. ART-CPN is successfully applied to solve the forecasting problems. The algorithm can reduce the learning time and obtain good prediction performance for solving forecasting problems.

References

- Carpenter, G. A., & Grossberg, S. (1987a). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphic, and Image Processing*, 37, 54–115.
- Carpenter, G. A., & Grossberg, S. (1987b). ART2: Stable self-organization of pattern recognition codes for analog input patterns. *Applied Optic*, 26, 4919–4930.
- Carpenter, G. A., & Grossberg, S. (1990). ART3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Networks*, 3, 129–152.

- Carpenter, G. A., Grossberg, S., David, B. R., & Fuzzy, ART. (1991). Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4, 759–771.
- Chang, F. J., & Chen, Y. C. (2001). A counterpropagation fuzz-neural network modeling approach to real time streamflow prediction. *Journal of Hydrology*, 245, 153–164.
- Chun, S. H., & Kim, S. H. (2004). Data mining for financial prediction and trading: Application to single and multiple markets. *Expert Systems with Applications*, 26, 131–139.
- Han, J., & Kamber, M. (2001). Data mining: Concept and techniques. *Morgan Kaufmann*, 338–344.
- Hecht-Nielsen, R. (1988). Application of counterpropagation networks. *Neural Networks*, 1, 131–139.
- Kohonen, T., Hynninen, J., Laaksonen, J., & Torkkola, K. (1995). *The learning vector quantization program package reference guide*. www.cis.hut.fi/research/som-research/nnrc-programs.shtml, 1995 (pp.1-10).
- Kim, I. S., Jeong, Y. J., Lee, C. W., & Yarlalagadda, P. K. D. V. (2003). Prediction of welding parameters for pipeline welding using an intelligent system. *International Journal of Advanced Manufacturing Technology*, 22, 713–719.
- Lin, Y., & Cunningham, G. A. (1995). A new approach to fuzzy-neural system modeling. *IEEE Transaction on Fuzzy System*, 3(2), 190–198.
- Lin, T. C., & Yu, P. T. (2003). Centroid neural network adaptive resonance theory for vector quantization. *Signal Processing*, 83, 649–654.
- Shi, S. M., Xu, L. D., & Liu, B. (1999). Improving the accuracy of nonlinear combined forecasting using neural networks. *Expert Systems with Applications*, 16, 49–54.
- Yang, B. S., Han, T. H., & Kim, Y. S. (2004). Integration of ART-Kohonen neural network and case-based reasoning for intelligent fault diagnosis. *Expert Systems with Applications*, 26, 387–395.