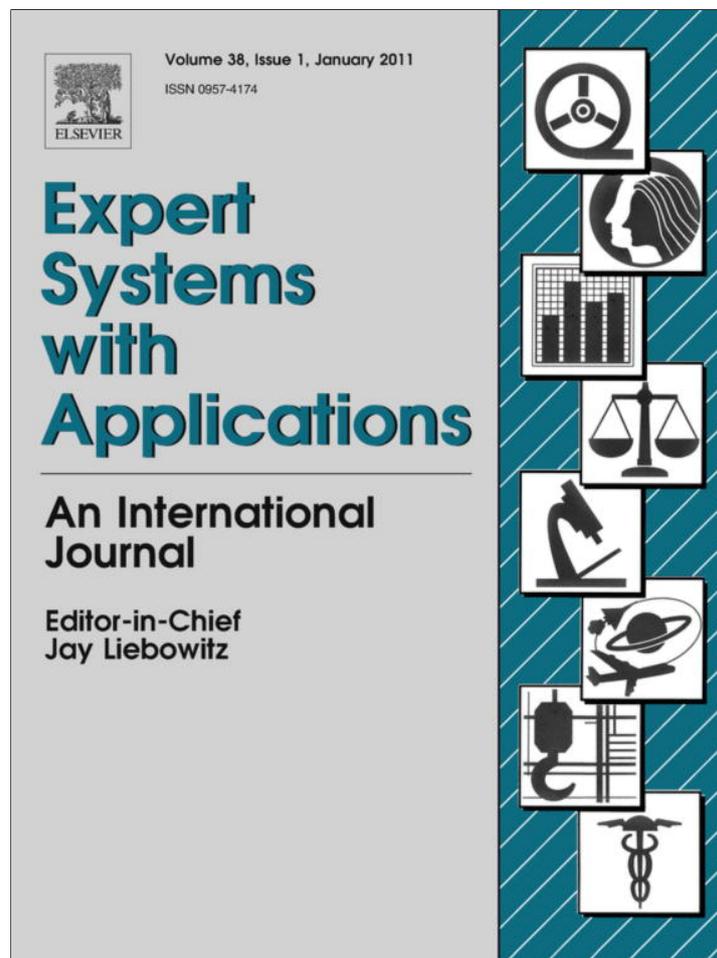


Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A Hybrid Higher Order Neural Classifier for handling classification problems

Mehdi Fallahnezhad^{a,*}, Mohammad Hassan Moradi^a, Salman Zaferanlouei^b^a Bioelectronics Department, Lab of Biomedical Signal Processing, Faculty of Biomedical Engineering, Center of Excellence in Biomedical Engineering, Amirkabir University of Technology (Tehran Polytechnic), 424 Hafez Avenue, Tehran, Iran^b Nuclear Engineering and Physics Department, Faculty of Nuclear Engineering, Amirkabir University of Technology (Tehran Polytechnic), 424 Hafez Avenue, Tehran, Iran

ARTICLE INFO

Keywords:

Higher order neural network (HONN)
 Classification problems
 High-order unit
 Model selection
 Feature subset selection

ABSTRACT

In this paper, we propose a novel Hybrid Higher Order Neural Classifier (HHONC) which contains different high-order units. In contrast with conventional fully-connected higher order neural networks (HONN), our proposed method uses fewer learning parameters and allocates the best fitted model in dealing with different datasets by modifying the orders of different high-order units and updating the learning parameters. Structure, model selection and updating the learning parameters of HHONC is introduced and is applied in classification of the Iris data set, the breast cancer data set, the Wine recognition data set, the Glass identification data set, the Balance scale data set, and the Pima diabetes data set. Acquired results are compared with the methods presented in [Chen and Shie \(2009\)](#). It is observed that the fewer features the dataset contains, the more accurate the HHONC performs, however the accuracy of datasets with more features are acceptable. Experimental results show about 3.5% and 0.6% improvements compared to the best accuracy obtained in previously methods for classifying the Pima diabetes and Iris datasets, respectively. In addition, by using a same method for reducing the feature number, it's shown the proposed method perform more accurate than methods presented in [Shie and Chen \(2008\)](#). In this case, improvements compared to the best acquired accuracy of mentioned methods are about 1.7%, 1.3% and 0.2% in classification of Pima, Iris and Breast cancer datasets, respectively.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Varied structures of Artificial Neural Networks (ANNs) have been proposed to solve classification problems. They are basically used as supervised learning neural classifiers and unsupervised competitive learning classifiers. Back-Propagation Network (BPN) is proposed by [Rumelhart, Hinton, and Williams \(1986\)](#) and caused revolutionary effects on neural network learning algorithm to solve nonlinear classification problems. Radial Basis Function (RBF) neural networks introduced by [Moody and Darken \(1989\)](#) were also examined as a hybrid layer scheme to learn classification tasks more computationally efficient which combines self-organized and supervised learning faster than BPNs. Self-Organization Map (SOM) introduced by [Kohonen \(1989, chap. 5\)](#) and Adaptive Resonance Theory (ART) of [Grossberg \(1988\)](#) are the most famous unsupervised neural networks applied in classification field.

Furthermore, many different NN-based methods are investigated in pattern classification of statistical datasets. [Ou and Murphey \(2007\)](#) used a system of multiple (or a single) neural networks for classification of the Glass and Shuttle datasets in different ap-

proaches of selection of training and testing instances. To overcome the problem of finding optimal ANN model, [Rocha, Cortez, and Neves \(2007\)](#) apply evolutionary computation (EC) algorithm for the optimization of ANN and use the proposed EC/ANN method to 16 real-world datasets from UCI machine learning (e.g. Balance scale, Wisconsin breast cancer). [Lee, Chen, and Lu \(2003\)](#) propose a fast self-organizing hierarchical cerebellar model arithmetic computer (HCMAC) neural network classifier for resolving high-dimensional classification problems via low memory requirement. This method is applied to different statistical datasets (e.g. Iris, Wisconsin breast cancer, Wine, etc.). [Furao and Hasegawa \(2008\)](#) employs a very fast self-organizing incremental neural network (SOINN) classifier based on fast nearest neighborhood in some artificial and real-world (e.g. Iris, Wisconsin breast cancer, Pima Indians, Glass, Wine, etc.) datasets. They have shown that the proposed method, in contrast with conventional ANN, is robust to noisy dataset. Additionally, ANN has been used along other methods for handling classification problems. Wavelet neural networks are implemented for prediction of bankruptcy and benchmark datasets (e.g. Iris, Wine, Wisconsin breast cancer) ([Chauhana, Ravi, & Chandra, 2009](#)). [Quteishat, Lim, Tweedale, and Jain \(2009\)](#) proposed a NN-based multi-agent classifier system (MACS) including two agent teams; fuzzy min-max (FMM) NN and fuzzy ARTMAP (FAM) NN for the classification of five benchmark datasets (e.g. Pima Indian diabetes, Glass, Wine, Breast cancer and Sonar).

* Corresponding author. Tel.: +98 912 2703292; fax: +98 216 6468186.

E-mail addresses: mfallah@aut.ac.ir, mehdi.fallah@gmail.com (M. Fallahnezhad), mhmoradi@aut.ac.ir (M.H. Moradi), zaferanlooei@aut.ac.ir (S. Zaferanlouei).

Among many different structures of ANNs, BPNs are most widely used for many different areas (Tsai, 2009). Although, these networks are recognized for their excellent performance in mathematical function approximation, nonlinear classification, pattern matching and pattern recognition, they have several limitations. They do not excel in discontinues or non-smooth (small changes in inputs cause large changes in outputs) data mining and knowledge learning (Fulcher, Zhang, & Shuxiang, 2006; Zhang, Xu, & Fulcher, 2002). Also, they cannot deal well on incomplete or noisy data (Dong & Pei, 2007; Peng & Zhu, 2007; Wang, 2003). A traditional ANN with single connections having no hidden layer just can map neuron inputs linearly to neuron outputs. Single hidden layer FNNs are known for their ability of function approximation and learning pattern recognition. In spite of these, they cannot realize every nonlinear mapping. FNNs incorporating more hidden layers are well-known for their long time convergence and usually stuck in local minima rather than global minima. They have more complex structure than others so that initializing the weights will be difficult and also investigating of the input–output mapping might be a disaster. Furthermore, most often the purpose of using an intelligent unit is to have an open-box model of input–output mapping. BPNs are acting like black-box models and as the number of hidden layers increases, this black-box becomes harder to understand.

In order to improve the various limitations of traditional neural network, higher order neural networks (HONNs) can be considered. By combining inputs, nonlinearly, HONNs make a higher order correlation among inputs (Zurada, 1992). Unlike BPNs, HONN successfully provides an efficient open-box model of nonlinear input–output mapping which provides easier understanding of data mining. By leaving necessity of hidden layers, HONN structures become simpler than FNNs and initialization of learning parameters (weights) will not be catastrophic. Furthermore, activation function in some structures of HONNs is adaptive (sigmoid, sine, cosine, SINC, etc.) or for better fitting specifications, model may have more than one activation function type. Moreover, HONNs most often run faster than FNNs. Examples in implementation of two-input and three-input XOR functions by using second-order neural network (SONN) proved that SONN is several times faster than FNN (Gupta, Homma, Hou, Solo, & Goto, 2009; Zhang, 2008). Regarding mentioned limitations of ANNs and advantages of HONNs, we are motivated to use higher order neural networks for handling classification dilemmas.

Various types of HONNs have been applied widely in different research areas including time series data, business, finance and economics (Fulcher et al., 2006; Ghazali, Hussain, Al-jumeily, & Merabti, 2007; Hussain & Liatsis, 2008; Xu, 2008), clustering (Ramanathan & Guan, 2007), classification (Abdelbar, 1998), pattern recognition and mathematical function approximation (Artyomov & Yadid-Pecht, 2005; Foresti & Dolso, 2004; Rovithakis, Maniadakis, & Zervakis, 2004; Tsai, 2009; Wang & Lin, 1995; Zhang, 2008). A second-order neural network was trained to distinguish between two objects was applied to handle classification problems by Reid, Spirkovska, and Ochoa (1989); also, higher order nets format as a higher-order processing unit (HPU) was suggested by Lippmann (1989). Different strategies for non-fully-connected HONNs are discussed for pattern recognition (Spirkovska & Reid, 1990). Thereafter, a higher order neural network called pi-sigma type for reducing the number of weights and processing units were proposed by Shin and Ghosh (1991) and it was used for pattern classifications; besides, a polynomial connectionist network called ridge polynomial network (RPN) that is a generalization of the pi-sigma network and provides more efficient architecture in comparison with ordinary higher-order feed-forward networks was implemented by Shin and Ghosh (1995) to classify high-dimensional data and better results was obtained. Chow and Yu (1995) propose

an estimation theory and optimization algorithm for identifying the number of hidden units of HONN applied for system identification. Several works have been done to implement, and improve HONN's parameters performance in pattern recognition and classification (Abdelbar, 1998; He & Siyal, 1999; Jakubowski, Kwiatos, Chwaleba, & Osowski, 2002; Kaita, Tomita, & Yamanaka, 2002). A new approach for determining the number of hidden neurons was also proposed and applied in data mining using adaptive HONN model by Xu and Chen (2008).

In this paper, we introduce a novel classifier based on HONN models, and then we use the proposed classifier over various benchmark statistical datasets including Iris data set, Breast cancer data set, Wine recognition data set, Glass identification data set, Balance scale data set and Pima diabetes data set. We test the proposed method over original datasets and also datasets which their features are reduced by the use of a feature subset selection method. We compare the results with priority presented methods.

2. Higher order neural network structures and models

In recent years, many different types of HONN structures have been developed. Many different explanations of high-order neural networks are given by different peoples. The simplest structure is that inputs of network in addition to original inputs are some product of them (Lee et al., 1986). However, “high-order” concept is widely used in neuron type (linear, power, multiplicative, sigmoid, etc.), neuron activation function type (polynomial, sigmoid, cosine, sine, SINC, etc.), (Fulcher et al., 2006; Zhang, 2008) and as higher order neural networks with adaptive functions (Xu, 2008), etc. In summary, HONN uses a higher correlation of input neurons for better fitting properties which often leads to a higher number of learning parameters (weights). The greater the order is used the higher the number of parameters will be. Many efforts for decreasing the number of parameters in different areas have been done. In summary, two major forms of HONN have been considered, sigma-pi network known as higher-order processing unit (HPU) and pi-sigma network (PSN). The former uses all of the combination of inputs up to the specified order. As a result, HPU models need more number of learning parameters to be updated which causes a longer run time. However, it provides an investigation of all correlations among different inputs. In contrast, the latter is developed to comprise a high order correlation between inputs by using lower number of weights which causes faster learning while avoids the high loss of performance. Shin and Ghosh (1991) have introduced an efficient higher order neural network as a PSN model and they have studied several pattern classification and function approximation problems. They claimed that a generalization of PSN can approximate any measurable function. Homma and Gupta (2002) developed a sigma-pi artificial second-order neural unit (ASONU) without losing the higher performance. Recurrent PSN (RPSN) and converging challenges is proposed and used in predicting the foreign currency exchange rates and results are compared with feed-forward and other recurrent structures (Hussain & Liatsis, 2008). Since, PSN is not a universal approximator, a generalization of PSN is the Ridge Polynomial Higher Order Neural Network (RPHONN) proposed by Shin and Ghosh (1995). Dynamic RPHONN is used in prediction of the exchange rate time series (Ghazali et al., 2007). Zhang (2008) introduces the general sigma-pi models in three structures named 1b, 1 and 0 models. Eqs. (1)–(3) show the presented models for a network including two features, x_1 and x_2 , in every sample.

HONN Model 1b:

$$Z = \sum_{ij=0}^N a_{ij}^0 \cdot \left\{ a_{ij}^{kx_1} \cdot f_i^{x_1} \left(a_i^{x_1} \cdot x_1 \right) \right\} \cdot \left\{ a_{ij}^{hx_2} \cdot f_j^{x_2} \left(a_j^{x_2} \cdot x_2 \right) \right\} \quad (1)$$

HONN Model 1:

$$Z = \sum_{ij=0}^N a_{ij}^0 \cdot \{f_i^{x_1}(a_i^{x_1} \cdot x_1)\} \cdot \{f_j^{x_2}(a_j^{x_2} \cdot x_2)\} \quad (2)$$

HONN Model 0:

$$Z = \sum_{ij=0}^N a_{ij}^0 \cdot \{f_i^{x_1}(x_1)\} \cdot \{f_j^{x_2}(x_2)\} \quad (3)$$

Zhang (2008) also develops six different HONN including: Polynomial Higher Order Neural Network (PHONN), Trigonometric Higher Order Neural Networks (THONN), SINC Higher Order Neural Network (SINCHONN), Sigmoid Polynomial Higher Order Neural Network (SPHONN), Ultra High Frequency Cosine and Sine Higher Order Neural Network (UCSHONN), SINC and Sine Polynomial Higher Order Neural Network (SXSPHONN).

In fact, each of these six models can be implemented in one of the structures 1b, 1 and 0. By changing one of these six models to another, neuron activation function inside the sigma symbol changes. Zhang (2008) proves mathematically that presented HONN models always converge and have better accuracy than Statistical Analysis System Nonlinear (SAS NLIN) models. Zhang (2008) presents the advantageous of HONN models over SAS NLIN and shows that different structures of model 0 get smaller residual mean squared error than SAS NLIN for the US population growth data and for the Quadratic with Plateau data. Fig. 1 shows block diagram of an HONN implemented as model 0.

By using model 0, Zhang (2008) reports the order, MSE and run time of the best models for the exchange rate Yen vs. US dollar year 2004 in different structures including PHONN, THONN, UCSHONN (Ultra High Frequency Cosine and Sine Higher Order Neural Networks), SXSPHONN (SINC and Sine Polynomial Higher Order Neural Networks), SINCHONN. The Best model 0 is reported to be a UCSHONN model of order 5. Also, Zhang (2008) uses these structures

for US Consumer Price Index (1992–2004) and reports the order of each model.

3. Structure and learning algorithm of Hybrid Higher Order Neural Classifier

As Eqs. (1) and (2) demonstrate, in spite of better fitting properties, the number of learning parameters in HONN model 1b and 1 is more than HONN model 0 and as the order of model gets higher, updating the parameters will become more difficult. Thus, in this study, we propose a two hidden-layered higher order neural classifier which includes online updating orders of four PHONN, SPHONN (involving both LOGSIG and TANSIG), SINCHONN and THONN models implemented by HONN model 0 in first layer. In second layer, linear combinations of these units are chosen.

Different high-order units in the first layer are organized as follows (Zhang, 2008).

3.1. PHONN model

Polynomial Higher Order Neural Networks (PHONNs) are developed by choosing the activation function as a polynomial function. The learning formula of PHONN and other five models are presented by Zhang (2008). Accordingly, in this model neuron activation function, $f(x)$, in model presented in Fig. 1, is linear.

3.2. THONN model

In Trigonometric Higher Order Neural Networks (THONN) the activation functions are trigonometric (sine & cosine) functions. We use cosine and sine alternately over the features e.g. for data set including two features: $f(x_1) = \cos(x_1)$, $f(x_2) = \sin(x_2)$.

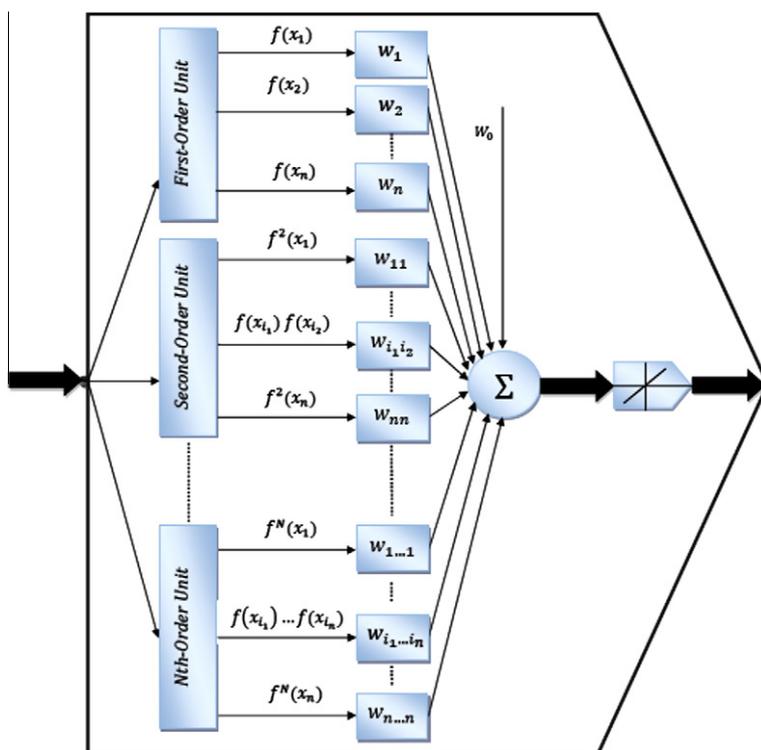


Fig. 1. Block diagram of each higher order unit in HHONC.

3.3. SINCHONN

In SINC Higher Order Neural Networks (SINCHONN) all of the activation functions are SINC functions; $f(x) = \frac{\sin(x)}{x}$.

3.4. SPHONN

In Sigmoid Polynomial Higher Order Neural Networks (SPHONN) the activation functions are SIGMOID functions. Both LOGSIG and TANSIG functions can be used

$$TANSIG : f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad LOGSIG : f(x) = \frac{1}{1 + e^{-x}}$$

As it's cited in previous section, each HONN unit (PHONN, THONN, SPHONN and SINCHONN) converges solely, so each linear combination of these units will converge too. In addition, by adding a hidden layer including linear neurons, convergence of network will not change, so proposed network always converges.

Fig. 2 shows a general aspect of HHONC. Generally, it contains various high-order units. The order of each high-order unit dynamically changes in training process (decrease and increase) to obtain a desired accuracy for validation samples (or MSE). As shown in Fig. 2, at the output stage there is a shifted hard-limit function. This unit is just used in testing process and it's because of our binary bit coding (0, 1) for identifying different classes. So as Fig. 2 illustrates, the HHONC output during the training process is shown by Z and the HHONC output during test samples process is shown by Ẑ.

Eqs. (4) and (5) mathematically demonstrate HHONC outputs as a function of input features

$$y_j(\bar{x}) = w_{j0}^{(1)} + \sum_{i_1=1}^P w_{ji_1}^{(1)} f_1(x_{i_1}) + \sum_{i_1=1}^P \sum_{i_2=i_1}^P w_{ji_1 i_2}^{(1)} f_1(x_{i_1}) f_1(x_{i_2}) + \dots + \sum_{i_1=1}^P \dots \sum_{i_{N_1}=i_{N_1-1}}^P w_{ji_1 \dots i_{N_1}}^{(1)} f_1(x_{i_1}) \dots f_1(x_{i_{N_1}}) + w_{j0}^{(2)} + \sum_{i_1=1}^P w_{ji_1}^{(2)} f_2(x_{i_1}) + \sum_{i_1=1}^P \sum_{i_2=i_1}^P w_{ji_1 i_2}^{(2)} f_2(x_{i_1}) f_2(x_{i_2}) + \dots + \sum_{i_1=1}^P \dots \sum_{i_{N_2}=i_{N_2-1}}^P w_{ji_1 \dots i_{N_2}}^{(2)} f_2(x_{i_1}) \dots f_2(x_{i_{N_2}}) + \dots + w_{j0}^{(K)} + \sum_{i_1=1}^P w_{ji_1}^{(K)} f_K(x_{i_1}) + \sum_{i_1=1}^P \sum_{i_2=i_1}^P w_{ji_1 i_2}^{(K)} f_K(x_{i_1}) f_K(x_{i_2}) + \dots + \sum_{i_1=1}^P \dots \sum_{i_{N_K}=i_{N_K-1}}^P w_{ji_1 \dots i_{N_K}}^{(K)} f_K(x_{i_1}) \dots f_K(x_{i_{N_K}}) \quad (4)$$

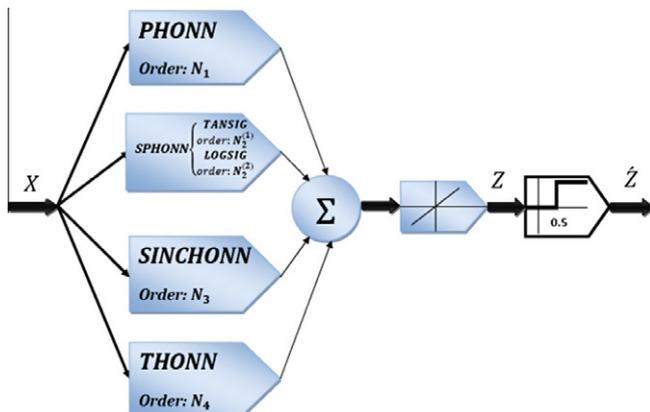


Fig. 2. General block diagram of the HHONC.

$$z_l = Y \times V = v_{j0} + \sum_{j=1}^K v_{jl} y_j(\bar{x}), \quad Z = [z_1, \dots, z_l, \dots, z_L] \quad (5)$$

where, L is the number of outputs (bit number of output binary code), P is the number of features, K is the number of higher order units which have the order greater than 0 and N_1, N_2, \dots, N_K are optimum acquired order of higher order units 1, 2 to K.

The number of hidden neurons in first layer is directly specified by the order of high-order units. Second hidden layer of HHONC is a linear order and experiments reveal that the desired accuracy can be reached if the number of hidden neurons of this layer is chosen to be same as the number of bits that output is coded by, for example when we have 2, 3 or 7 class, the number of hidden neurons in second layer are 1, 2, 3, respectively.

The first and second layer weights are updated by Eqs. (6) and (7), respectively

$$w_{ji_1 \dots i_{N_k}}^{(k)}(t+1) = w_{ji_1 \dots i_{N_k}}^{(k)}(t) - \eta \cdot \left(\frac{\partial E}{\partial w_{ji_1 \dots i_{N_k}}^{(k)}} \right) \\ \frac{\partial E}{\partial w_{ji_1 \dots i_{N_k}}^{(k)}} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial w_{ji_1 \dots i_{N_k}}^{(k)}} = \frac{\partial E}{\partial y_j} \cdot f_{k_1}(x_{i_1}) \dots f_{k_{N_k}}(x_{i_{N_k}}) \\ \frac{\partial E}{\partial y_j} = \frac{\partial}{\partial y_j} \left[\frac{1}{2} \sum_{l=1}^L (d_l - z_l)^2 \right] = \sum_{l=1}^L (d_l - z_l) \frac{\partial}{\partial y_j} (Y \times V) = \sum_{l=1}^L (d_l - z_l) v_{jl} \quad (6)$$

$$v_{jl}(t+1) = v_{jl}(t) - \eta \cdot \left(\frac{\partial E}{\partial v_{jl}} \right) \\ \frac{\partial E}{\partial v_{jl}} = \frac{\partial E}{\partial z_l} \cdot \frac{\partial z_l}{\partial v_{jl}} = \frac{\partial E}{\partial z_l} \cdot y_j = y_j \cdot \sum_{l=1}^L (d_l - z_l) \quad (7)$$

where, t is training state, η is learning coefficient, E is error, W is weight matrix of first hidden layer and V is weight matrix of second hidden layer, L is the number of outputs (Bit number of output binary code), P is the number of input samples, K is the number of higher order units which have the order greater than 0 and N_1, N_2, \dots, N_K are optimum acquired orders of higher order units 1, 2 to K.

In summary, learning algorithm of HHONC is organized as follows:

- Step 1.** Initialize $N_1, N_2^{(1)}, N_2^{(2)}, N_3, N_4$ by 0; set order of model to $N = [N_1 \ N_2^{(1)} \ N_2^{(2)} \ N_3 \ N_4]$.
- Step 2.** Increase one the order of model (N). (e.g. when order is [0 0 0 1], the next orders will be [0 0 0 1 0], [0 0 0 1 1], [0 0 1 0 0], [0 0 1 0 1], ...)
- Step 3.** Set weights to current model and initialize them.
- Step 4.** Update weights by using Eqs. (7) and (8); calculate the accuracy and mean squared error of training and especially validation samples.
- Step 5.** If the accuracy (or MSE) of validation samples reached to the desired value, stop. Otherwise go to step 2.

As an example, in Fig. 3 we have shown outputs of different stages including before (Z) and after (Ẑ) the shifted hard-limit unit of the one HHONC implemented for breast cancer dataset. This dataset includes 699 samples and 9 attributes in 2 classes; Malignant (241 instances) and Benign (458 instances). This dataset contains 16 samples with missing attributes which are not used in our process.

4. Experimental results

We use six different kinds of UCI datasets (ftp://ftp.ics.uci.edu/pub/machine-learning-databases/) including the Iris data set, the Breast cancer data set, the Wine recognition data set, the Glass identification data set, the Balance scale data and the Pima diabetes data set. For each dataset we normalize the data over each

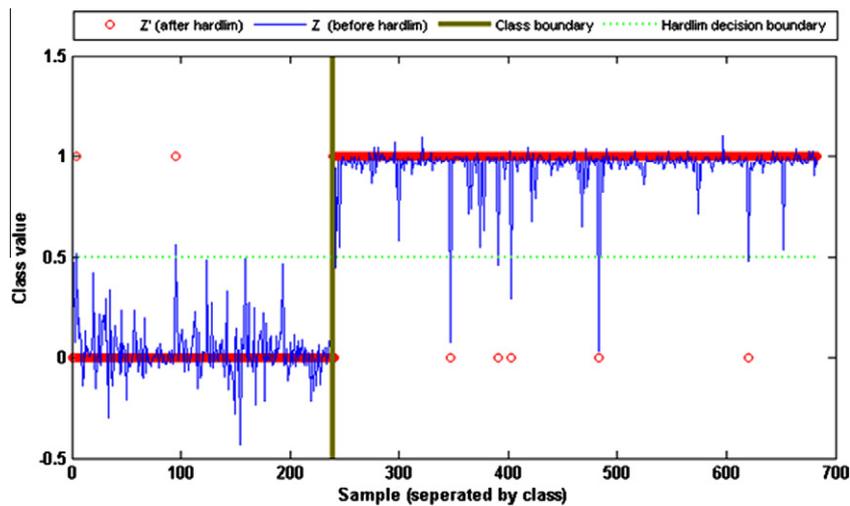


Fig. 3. Output of different stages of HHONC for breast cancer dataset.

feature value to have mean value of 0 and variance value of 1. Eq. (8) covers this concept:

$$\hat{f}_i = \frac{f_i - \text{mean}(F_j)}{\sqrt{\text{var}(F_j)}}$$

$$\text{mean}(F_j) = \frac{\sum_{i=1}^n f_i}{n}$$

$$\text{var}(F_j) = \frac{\sum_{i=1}^n (f_i - \text{mean}(F_j))^2}{n}, \quad F_j = [f_1, \dots, f_i, \dots, f_n]$$

where, \hat{f}_i is the normalized value of feature j th of sample i th, f_i is the original value of feature j th of sample i th, F_j is the values of feature j th of all samples.

Meant for providing excellent comparison with other classifiers, two kinds of experiments are carried out on different datasets. At the first one, we use the original datasets including all of features and at the second experiment we apply our proposed HHONC to a selected subset of features. In both experiments we compare the proposed method with other classifiers.

Table 1
HHONC configuration; order of different high-order units for original datasets.

Data sets	Orders of different units of HHONC				
	PHONN	SPHONN TANSIG	SPHONN LOGSIG	SINCHONN	THONN
Iris data set	0	0	2	0	1
Breast cancer data set	0	1	0	1	1
Wine recognition data set	1	0	1	0	0
Glass identification data set	1	1	0	1	0
Balance scale data set	1	1	3	0	0

4.1. Experiments on original datasets

In this experiment, we use original datasets and try to find a HHONC model for each one. For each dataset, we use 65% of data as training instances, 10% as validation instances and 25% as testing instances which are randomly selected over dataset. In fact, for providing a good comparison with previous works we use 25% of dataset as test samples; however, in spite the other compared classifiers, we need to choose validation samples over the dataset for our training stop epoch. For probing the robustness of the method over the various testing instances, we repeat the method 200 times on each data set by differing train, validation and test instances.

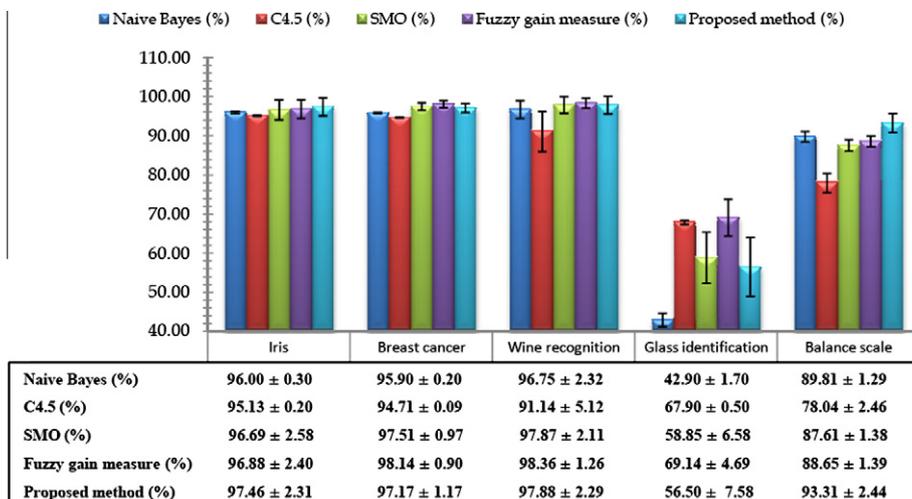


Fig. 4. Comparison of the average classification accuracy rates of proposed method by prior presented classifiers; for original datasets.

Table 1 presents the resultant orders of different high-order units for each HHONC.

Fig. 4 presents the comparison of presented method with the Naive Bayes method (John & Langley, 1995), the C4.5 method (Quinlan, 1993), the sequential minimal optimization (SMO) (Platt, 1999) and fuzzy gain measure (Chen & Shie, 2009; Shie & Chen, 2006). The reported values are organized as: Mean value ± Standard deviation, which are computed from 200 independent trials.

From Fig. 4 we can see that while the number of features of a data set is lowenough (about four features), the proposed method gets higher average classification accuracy rates than the others. Meanwhile, the accuracy of data sets with more features is acceptable.

4.2. Experiments on feature subset selected datasets

In this section, we use a method based on fuzzy entropy measures proposed by Shie and Chen (2008) for feature subset selection.

Table 2
HHONC configuration; order of each high-order units for feature subset selected datasets.

Data sets	Orders of different units of HHONC				
	PHONN	SPHONN TANSIG	SPHONN LOGSIG	SINCHONN	THONN
Iris data set	2	1	0	0	0
Breast cancer data set	0	1	0	1	0
Pima diabetes data set	3	3	0	1	0

Then we apply again the proposed HHONC to the new datasets with selected features. In this experiment, we use the 10 times 10-fold cross validation method for acquiring the average accuracy of different classifiers. Table 2 indicates the order of different high-order units for each dataset. Fig. 5 covers a comparison of acquired accuracy of using HHONC with the other classifiers reported in Shie and Chen (2008) including LMT (Landwehr, Hall, & Frank, 2003), Naive Bayes (John & Langley, 1995), C4.5 (Quinlan, 1993) and SMO (Pal, De, & Bsask, 2000). All of the classifiers use the same subset of data features selected by fuzzy entropy measures.

As the detailed results given in Fig. 5 demonstrate, the lower the number of features of dataset is, the higher the accuracy of HHONC will be.

For providing more detailed information of the strength of proposed classifier, we have recorded all 10 test, train and validation degrees of accuracy of tested models for these three datasets. We examined 624 different models for Breast cancer dataset. Fig. 6 shows the test, validation and train accuracy of each model sorted by highest test accuracy. It can be clarified that accuracy of validation samples is varying similar to that of test samples which certifies our procedure for finding the best model. Fig. 7A and B respectively show the accuracy and complexity of each model and Fig. 7C shows the accuracy of each model with respect to the model complexity over the Pima diabetes dataset. We assume the model complexity (N) of each HHONC to be the summation of high-order units ($N = N_1 + N_2^1 + N_2^2 + N_3 + N_4$).

As mentioned above, one of the main advantages of proposed classifier, compared with other neural classifiers and most of

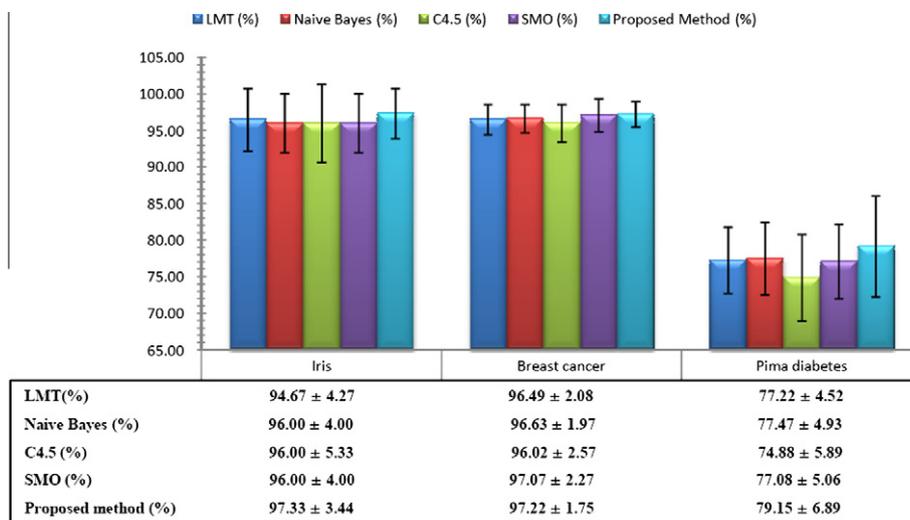


Fig. 5. Comparison of average classification accuracy rates of proposed method by prior presented classifiers; for feature subset selected datasets.

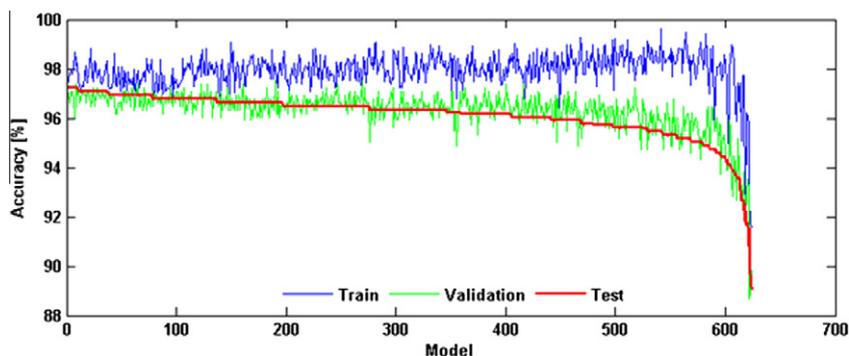


Fig. 6. Accuracy of train, validation and test samples of 623 models for Wisconsin breast cancer dataset.

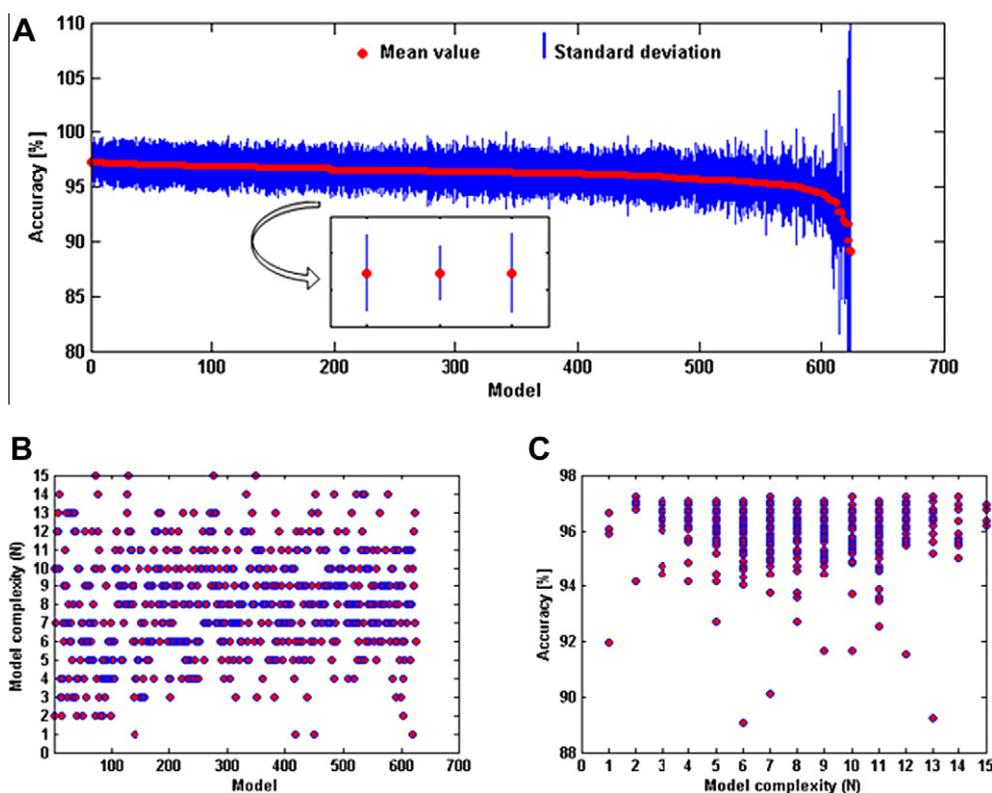


Fig. 7. Accuracy and model complexity of 624 models for Wisconsin breast cancer dataset.

Table 3
HHONC model coefficients for the Iris dataset.

	a_{i0}	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	a_{i6}	a_{i7}
$i = 1$	0.1220	1.3861	0.7877	-0.8811	0.5141	-0.5153	-0.2855	-0.4799
$i = 2$	1.3314	-1.3341	-2.3523	3.5471	-3.9824	2.7494	-0.0502	3.1171

commonly-used classifiers, is its open-box property. For example, as shown in Table 2, the best open-box model achieved is given by (9). The coefficients of each output are given in Table 3 (since the Iris dataset includes 3 classes, it's coded by 2 bit output). This capability can be more notable whenever the dataset includes 2 classes (i.e. Breast cancer dataset and Pima diabetes dataset)

$$\begin{aligned}
 Z &= [Z_1 Z_2] \\
 Z_i &= a_{i0} + a_{i1} \cdot x_1 + a_{i2} \cdot x_2 + a_{i3} \cdot x_1^2 + a_{i4} \cdot x_1 \cdot x_2 \\
 &\quad + a_{i5} \cdot x_2^2 + a_{i6} \cdot T(x_1) + a_{i7} \cdot T(x_2) \\
 T(x) &= \frac{1 - e^{-x}}{1 + e^{-x}}, \quad i = 1, 2
 \end{aligned}
 \tag{9}$$

where x_1 is 3rd feature and x_2 is 4th feature of Iris dataset.

5. Conclusion

In this study, we have introduced a new Hybrid Higher Order Neural Classifier for pattern classification of different statistical datasets. We have shown the proposed method gets high average classification accuracy rates in comparison to the methods presented in John and Langley (1995), Quinlan (1993), Platt (1999) and Chen and Shie (2009). Also we have shown that HHONC acts excellently over the dataset with lower number of features. In a special case, we have shown by using a feature subset selection method, that when dimension of datasets are reduced, HHONC gets

higher accuracy rather than the other classifiers including LMT (Landwehr et al., 2003), Naive Bayes (John & Langley, 1995), C4.5 (Quinlan, 1993) and SMO (Pal et al., 2000). In summary, it is concluded that the proposed method performs excellently over datasets with less features, however by the trade-off between both open-box property and run time with achieved accuracy, it can be used with more features. The main reason of acquiring better accuracy over low-feature dataset is using the various correlations of features. By diminishing the number of features, the method will be able to check more correlations over features.

As mentioned, by using hybrid components of different HONN units and so decreasing the number of learning parameters (weights) instead of fully-connected HONN, we find a HHONC model accurately fitted over each dataset. In future, it may become useful to use adaptive higher order functions or different wavelet functions as the kernel of HHONC instead of each neuron activation function. In this case, just a few learning parameters will be added while it may provide fewer orders for different units of HHONC. Another future trend can be considered in selecting best model of HHONC. The evolutionary algorithm (Genetic algorithm) may also be useful for diminishing the processing time of model selection.

References

Abdelbar, A. (1998). Achieving superior generalization with a high order neural network. *Neural Computing and Applications*, 7(2), 141–146.

- Artyomov, E., & Yadid-Pecht, O. (2005). Modified high-order neural network for invariant pattern recognition. *Pattern Recognition Letters*, 26(6), 843–851.
- Chauhana, N., Ravi, V., & Chandra, D. K. V. (2009). Differential evolution trained wavelet neural networks: Application to bankruptcy prediction in banks. *International Journal of Expert System with Application*, 36(4), 7659–7665.
- Chen, S. M., & Shie, J. D. (2009). Fuzzy classification systems based on fuzzy information gain measures. *International Journal of Expert Systems with Applications*, in press.
- Chow, J. L., & Yu, T. W. S. Y. (1995). The estimation theory and optimization algorithm for the number of hidden units in the higher-order feedforward neural network. In *Proceedings, IEEE international conference on neural networks* (Vol. 3, pp. 1229–1233).
- Dong, G., & Pei, J. (2007). *Sequence data mining*. Springer.
- Foresti, G. L., & Dolso, T. (2004). An adaptive high-order neural tree for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(2), 988–996.
- Fulcher, J., Zhang, M., & Xu, S. (2006). Application of higher-order neural networks to financial time series prediction. In J. B. Kamruzzaman (Ed.), *Artificial neural networks in finance and manufacturing* (pp. 80–108). New York: Idea Group Inc. (IGI).
- Furao, S., & Hasegawa, O. (2008). A fast nearest neighbor classifier based on self-organizing incremental neural network. *Neural Networks*, 21(10), 1537–1547.
- Ghazali, R., Hussain, A. J., Al-jumeily, D., & Merabti, M. (2007). Dynamic ridge polynomial neural networks in exchange rates time series forecasting. In B. Beliczynski, A. Dzielinski, M. Iwanowski, & B. Ribeiro (Eds.), *Adaptive and natural computing algorithms* (pp. 123–132). Berlin/Heidelberg: Springer.
- Grossberg, S. (1988). Nonlinear neural networks: principles, mechanisms, and architectures. *Neural Networks*, 1, 17–62.
- Gupta, M. M., Homma, N., Hou, Z., Solo, A. M. G., & Goto, T. (2009). Fundamental theory of artificial higher order neural networks. In M. Zhang (Ed.), *Artificial higher order neural networks for economics and business* (pp. 368–388). Idea Group Inc. (IGI).
- He, Z., & Siyal, M. Y. (1999). Improvement on higher-order neural networks for invariant object recognition. *Neural Processing Letters*, 10(1), 49–55. ISSN: 1370-4621 (11 ref.).
- Homma, N., & Gupta, M. M. (2002). A general second order neural unit. *Bulletin of College Medical Sciences, Tohoku University*, 11(1), 1–6.
- Hussain, A., & Liatsis, P. (2008). A novel recurrent polynomial neural network for financial time series prediction. In Zhang, M. (Ed.), *Higher order neural networks for economics and business* (pp. 190–211). Idea Group Inc. (IGI).
- Jakubowski, J., Kwiatos, K., Chwaleba, A., & Osowski, S. (2002). Higher order statistics and neural network for tremor recognition. *IEEE Transactions on Biomedical Engineering*, 49(2), 152–159.
- John, G. H., & Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Proceedings of the 11th conference on uncertainty in artificial intelligence, Montreal, Canada* (pp. 338–345).
- Kaita, T., Tomita, S., & Yamanaka, J. (2002). On a higher-order neural network for distortion invariant pattern recognition. *Pattern Recognition Letters*, 23(8), 977–984.
- Kohonen, T. (1989). *Self-organization and associative memory* (3rd ed.). Berlin: Springer-Verlag (pp. 119–157).
- Landwehr, N., Hall, M., & Frank, E. (2003). Logistic model trees. In *Proceedings of the 14th European conference on machine learning, Cavtat-Dubrovnik, Croatia* (pp. 241–252).
- Lee, H. M., Chen, C. M., & Lu, Y. F. (2003). A self-organizing HCMAC neural network classifier. *IEEE Transaction on Neural Networks*, 14(1), 15–27.
- Lee, Y. C., Doolen, G., Chen, H., Sun, G., Maxwell, T., Lee, H., et al. (1986). Machine learning using a higher order correlation network. *Physica D: Nonlinear Phenomena*, 22, 276–306.
- Lippmann, R. P. (1989). Pattern classification using neural networks. *Communications Magazine, IEEE*, 27(11).
- Moody, J., & Darken, C. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1, 281–294.
- Ou, G., & Murphey, Y. L. (2007). Multi-class pattern classification using neural networks. *Pattern Recognition*, 40(1), 4–18.
- Pal, S. K., De, R. K., & Bsask, J. (2000). Unsupervised feature selection: A neuro-fuzzy approach. *IEEE Transactions on Neural Networks*, 11(2), 366–376.
- Peng, H., & Zhu, S. (2007). Handling of incomplete data sets using ICA and SOM in data mining. *Neural Computing and Applications*, 167–172.
- Platt, J. C. (1999). Using analytic QP and sparseness to speed training of support vector machines. In *Proceedings of the 13th annual conference on neural information processing systems, Denver, Colorado* (pp. 557–563).
- Quinlan, J. R. (1993). *C4.5 programs for machine learning*. San Francisco: Kaufmann.
- Quteishat, A., Lim, C. P., Tweedale, J., & Jain, L. C. (2009). A neural network-based multi-agent classifier system. *Neurocomputing*, 72(7–9), 1639–1647.
- Ramanathan, K., & Guan, S. U. (2007). Multiorder neurons for evolutionary higher-order clustering and growth. *Neural Computation*, 19(12), 3369–3391.
- Reid, M. B., Spirkovska, L., & Ochoa, E. (1989). Rapid training of higher-order neural networks for invariant pattern recognition. In *International joint conference on neural network* (Vol. 1, pp. 689–692).
- Rocha, M., Cortez, P., & Neves, J. (2007). Evolution of neural networks for classification and regression. *Neurocomputing*, 70(16–18), 2809–2816.
- Rovithakis, G. A., Maniadakis, M., & Zervakis, M. (2004). High-order neural network structure selection for function approximation applications using genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(1), 150–158.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representation by back-propagation errors. *Nature*, 323, 533–536.
- Shie, J. D., & Chen, S. M. (2006). A new approach for handling classification problems based on fuzzy information gain measures. In *Proceedings of the 2006 IEEE international conference on fuzzy systems, Reno, Nevada* (pp. 939–946).
- Shie, J. D., & Chen, S. M. (2008). Feature subset selection based on fuzzy entropy measures for handling classification problems. *Journal of Applied Intelligence*, 28, 69–82.
- Shin, Y., & Ghosh, J. (1991). The pi-sigma networks: An efficient higher-order neural network for pattern classification and function approximation. In *Proceedings of international joint conference on neural networks* (Vol. 1, pp. 13–18).
- Shin, Y., & Ghosh, J. (1995). Ridge polynomial networks. *IEEE Transactions on Neural Networks*, 6(3), 610–622.
- Spirkovska, L., & Reid, M. B. (1990). Connectivity strategies for higher-order neural networks applied to pattern recognition. In *International joint conference on neural network* (Vol. 1, pp. 21–26).
- Tsai, H.C. (2009). Hybrid high order neural networks. *International Journal of Applied Soft Computing*, in press.
- Wang, S. H. (2003). Application of self-organising maps for data mining with incomplete data sets. *Neural Computing and Applications*, 42–48.
- Wang, J. H., & Lin, J.H. (1995). Qualitative analysis of the BP composed of product units and summing units. In *IEEE international conference on systems, man and cybernetics* (Vol. 1, pp. 35–39).
- Xu, S. (2008). Adaptive higher order neural network models and their applications in business. In M. Zhang (Ed.), *Artificial higher order neural networks for economics and business* (pp. 314–329). IGI Global.
- Xu, S., & Chen, L. (2008). Application of new adaptive higher order neural networks in data mining. In *International conference on computer science and software engineering* (Vol. 1, pp. 115–118).
- Zhang, M. (2008). *Artificial higher order neural networks for economics and business*. Hershey: IGI Global.
- Zhang, M., Xu, S., & Fulcher, J. (2002). Neuron-adaptive higher order neural-network models for automated financial data modeling. *IEEE Transactions on Neural Networks*, 13(1), 188–204.
- Zurada, J. M. (1992). *Introduction to artificial neural systems*. St. Paul, West Publishing Company.