Design of Polynomial Neural Network Classifier for Pattern Classification with Two Classes

Byoung-Jun Park*, Sung-Kwun Oh† and Hyun-Ki Kim**

Abstract – Polynomial networks have been known to have excellent properties as classifiers and universal approximators to the optimal Bayes classifier. In this paper, the use of polynomial neural networks is proposed for efficient implementation of the polynomial-based classifiers. The polynomial neural network is a trainable device consisting of some rules and three processes. The three processes are assumption, effect, and fuzzy inference. The assumption process is driven by fuzzy c-means and the effect processes deals with a polynomial function. A learning algorithm for the polynomial neural network is developed and its performance is compared with that of previous studies.

Keywords: polynomial networks, pattern classification, spiral, two classes.

1. Introduction

In the past few years, various approaches have been investigated for pattern classification. There are statistical classifiers and neural classifiers [1]. Statistical classifiers include the linear discriminate function (LDF), the quadratic discriminate function (QDF), the nearest neighbor (1-NN), and k-NN rules, etc. Neural classifiers can be divided into relative density models and discriminative models depending on whether they aim to replicate the manifolds of each class or to discriminate the patterns of different classes [2]. Discriminative neural classifiers include the multi-layer perceptron (MLP), the radial basis function (RBF) network, and the polynomial classifier (PC) [3, 4], etc. The classification performances of statistical and neural classifiers have been evaluated in many comparison experiments. The results show that when trained with a large sample set, neural classifiers yield higher accuracies than statistical classifiers, whereas statistical classifiers are robust against small sample size [1].

The MLP have been widely used to solve pattern classification problems. It is shown that the MLP can be trained to approximate complex discriminant functions [5, 6]. One of the most widely applied neural classifiers is the RBF network, which is applied to both function approximation [7] and pattern classification [8, 9]. RBF networks have some advantages; global optimal approximation characteristic, favorable classification capability, and rapid convergence of learning procedure, etc. [10, 11].

Polynomial networks have been known in the literature for many years [1, 6, 12]. The polynomials have powerful approximation properties and excellent properties as classifiers. Because of the Weierstrass theorem, polynomial classifiers are universal approximators to the optimal Bayes classifier [13]. The Weierstrass theorem [14] states that any continuous function can be approximated to an arbitrary accuracy by polynomials. The polynomial classifier is also known as higher-order neural network [15] or functional link net [16]. With binomial expansion on linear subspace features, the polynomial classifier has shown superior performance to multilayer neural networks [1, 4].

Neural classifiers can deal with many multivariable nonlinear problems for which an accurate analytical solution is difficult to obtain [17]. It is found however that the use of neural classifiers depends on several parameters that are crucial to the accurate predictions of the properties sought. The appropriate neural architecture, the number of hidden layers, and the number of neurons in each hidden layer are issues that can greatly affect the accuracy of the prediction. Unfortunately, there is no direct method to specify these factors as they need to be determined on an experimental and trial basis [17]. In addition to that, it is difficult to understand and interpret a given problem for obtained neural classifiers that become increasingly complex if more variables and hidden layers are introduced [8, 18].

To improve the mentioned problems, we propose a polynomial neural network (PNN) classifier based on fuzzy c-means clustering and polynomials in this paper. The

† Corresponding author: Dept. of Electrical Engineering, University of Suwon, Korea. (ohsk@suwon.ac.kr)
* The Technology Research Institute, GM TECH CO., LTD., Korea (lcap2005@gmail.com)
** Dept. of Electrical Engineering, University of Suwon, Korea. (ohsk@suwon.ac.kr, hkkim@suwon.ac.kr)
Received 12 July, 2007 ; Accepted 12 January, 2008

proposed PNN classifier is a trainable device consisting of some rules and three processes. The three processes are assumption, effect, and fuzzy inference. The assumption process is driven by fuzzy c-means clustering and the effect process deals with a polynomial function. These processes contribute directly to if-then rules that provide intuitional interpretation using the linguistic analysis for a given problem. From the viewpoints of neural classifiers, fuzzy c-means clustering is used as an activity node in the same way that a sigmoid or radial based function in the hidden layer of networks and polynomials are as the connection weights between hidden and output layers. Use of fuzzy c-means clustering helps decision problem of activity nodes in hidden layers of MLP or RBF networks and polynomials provide the improved classification performance and understanding for certain problems that are difficult to solve. A learning algorithm for the PNN classifier is developed. The proposed PNN classifier is applied to a two-class problem and evaluated.

2. Polynomial Neural Network

2.1 Architecture of polynomial neural network

Lots of neural network models and learning algorithms have been proposed and studied for many decades. Multi-layer perceptron (MLP) and radial basis function (RBF) are two representative networks used in neural networks. However, the basic architecture of general neural networks always consists of three layers; input layer, hidden layer, and output layer, and it is fully connected with weights as shown in Fig. 1.

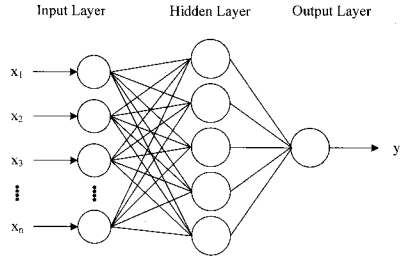


Fig. 1. Basic architecture of general neural networks

MLP trained by back-propagation learning can have one or more hidden layers for the basic architecture. There are no recurrent connections in the network. Every node except for those in the input layer has its own activation function. The activation functions are used to introduce nonlinearity into the network. Sigmoidal and logistic functions are commonly used as activation functions. The RBF network has one hidden layer and only the weights between the output layer and the hidden layer are trained. The hidden nodes compute their activation using radial basis functions. Gaussian function is one of the most popular radial basis functions [18].

The proposed polynomial neural network (PNN) has the same structure as the RBF networks. However, unlike RBF networks, the nodes in the hidden layer are activated by fuzzy c-means clustering. Namely, the network structure is automatically formulated by assigning each cluster to a node in the hidden layer. In addition to that, weights between the output layer and the hidden layer are represented by a polynomial with input variables as shown in Fig. 2.

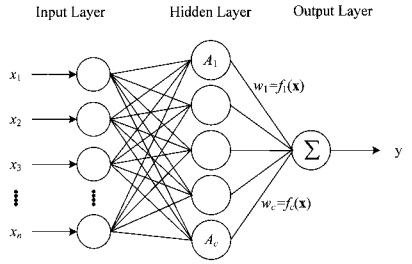


Fig. 2. PNN architecture as a general neural network

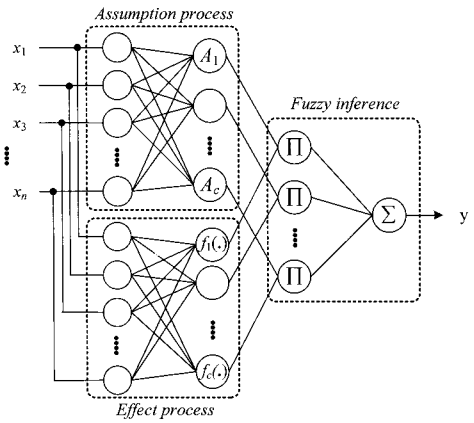


Fig. 3. PNN architecture sketched by two processes

From the viewpoint of linguistic analysis, PNN can be expressed by two processes; the assumption process driven by fuzzy c-means clustering, and the effect processes fired by the polynomial, which has some rules written as Eq. (1). Therefore, the number of rules is the number of clusters and the output of PNN is gotten by fuzzy inference. Eq. (1) is sketched as a neural network shown in Fig. 3, which is the equivalent to Fig. 2.

$$\text{If } \mathbf{x} \text{ is } A_i \text{ then } f_i(\mathbf{x}) \tag{1}$$

where, $\mathbf{x}$ is input vector $[x_1,\ldots x_n]$, A_i is membership function of i cluster, f_i is a polynomial function, n is the number of input variables and i is rule number (cluster no.). In Fig. 3, the nodes are connected by 1, namely all connection weights are 1. c is the number of clusters (rules).

2.2 Three processes of PNN

Assumption process: The assumption process of PNN is handled with fuzzy c-means clustering. In this section, we briefly review the objective function-based fuzzy clustering with intent of highlighting its key features. The fuzzy c-means comes as a standard mechanism aimed at the formation of 'c' fuzzy sets (relations) in R^n. The objective function Q guiding the clustering process is expressed as a sum of the distances of individual data from the prototypes $\mathbf{v}_1$, $\mathbf{v}_2$, ..., and $\mathbf{v}_c$,

$$Q = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{m}\left\|\mathbf{x}_k - \mathbf{v}_i\right\|^2 \tag{2}$$

Here, $\|\ \|$ denotes a certain distance function; 'm' stands for a fuzzification factor (coefficient), m>1.0. The resulting partition matrix is denoted by U=[u_{ik}], i=1, 2,...,c; k=1, 2, ..., N (Number of patterns). While there is a substantial diversity as far as distance functions are concerned, here we adhere to a weighted Euclidean distance taking on the following form

$$\left\|\mathbf{x}_k - \mathbf{v}_i\right\|^2 = \sum_{j=1}^{n}\frac{(x_{kj} - v_{ij})^2}{\sigma_j^2} \tag{3}$$

with σ_j being a standard deviation of the j-th variable. While not being computationally demanding, this type of distance is still quite flexible. The minimization of Q is realized in successive iterations by adjusting both the prototypes and entries of the partition matrix, min Q (U, $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_c$). The properties of the optimization algorithm are well documented in the literature, cf. [19, 20]. In the context of our investigations, we note that the resulting partition matrix is a clear realization of 'c' fuzzy relations with the membership functions U_1, U_2, ..., U_c forming the corresponding rows of the partition matrix U, that is U = [U_1^T U_2^T .... U_c^T].

Effect process: Polynomial functions are dealt with in the effect process. These functions are activated with partition matrix and lead to local regression models for the assumption process in each linguistic rule. We consider hear a polynomial function as presented below

Constant; $$f_i(\mathbf{x}) = a_{i0} \tag{4}$$

Linear; $$f_i(\mathbf{x}) = a_{i0} + \sum_{j=1}^{n} a_{ij}x_j \tag{5}$$

Quadratic; $$f_i(\mathbf{x}) = a_{i0} + \sum_{j=1}^{n} a_{ij}x_j + \sum_{j=1}^{n}\sum_{k=1}^{n} a_{ijk}x_j x_k \tag{6}$$

where, n is the number of input variables and i=1,...,c (the number of clusters).

As taking into consideration the three types of polynomial functions as a local model, the proposed PNN classifier with multi-input variables is available effectively. Accordingly, it is possible to consider the nonlinearity characteristics of process and to obtain better output performance. The use of constant Eq. (4) as a polynomial function in the PNN leads to a general neural network, namely, RBF network.

Fuzzy Inference: Let us consider the PNN structure by considering the fuzzy partition realized in terms of fuzzy c-means clustering as given in Fig. 3. The node denoted by Π is realized as a product of the corresponding fuzzy set and polynomial function. Making use of the language of the rule-based systems, the structure translates into Eq. (1). The format of the if-then rules is similar to fuzzy neural networks [21]. The output of each node generates a final output y using fuzzy inference method as the following form

$$y = g(\mathbf{x}) = \sum_{i=1}^{c} u_i f_i(\mathbf{x}) \tag{7}$$

where, $g(\mathbf{x})$ is a representation of PNN as a function, and u is a membership value (partition matrix) for A_i as a fuzzy set for input variables.

3. Polynomial Neural Network Classifier

3.1 Classification for the two-class problem

There are many different ways to represent pattern classifiers. One of the most useful is in terms of a set of discriminant functions $g_i(\mathbf{x})$, i=1,...,m (number of classes). The classifier is said to assign an input vector x to class ω_i if

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \qquad \text{for all } j \neq i. \tag{8}$$

Thus, the classifier is viewed as a network or machine that computes m discriminate functions and selects the category corresponding to the largest discriminant [22]. The two-class case is just a special instance of the multi-class case; it has traditionally received separate treatment. The classifier that places a pattern in one of only two categories has the special name of dichotomizer [22]. Instead of using two discriminant functions g_1 and g_2 and assigning x to ω_1 if $g_1 > g_2$, it is more common to define a single discriminate function $g(\mathbf{x})$ and to use the following decision rule

Decide ω_1 if $g(\mathbf{x}) > 0$; otherwise decide ω_2. (9)

Thus, a dichotomizer can be viewed as a machine that computes a single discriminant function g(x), and classifies x according to the algebraic sign of the result. In this paper, the proposed PNN classifier is used as a dichotomizer for the two-class type. The final output of PNN, Eq. (7), is used as a discriminant function g(x) that is a linear combination written as

$$g(\mathbf{x}) = \mathbf{a}^T\mathbf{f}\mathbf{x} \tag{10}$$

where, **a** is the coefficients of polynomial functions in Eqs. (4)- (6) and **fx** is a composition matrix of U and **x**. These can be defined as follows for each polynomial function.

i) Constant;

$$\mathbf{a}^T=[a_{10},\ldots,a_{c0}],\ \mathbf{fx}=[u_1,\ldots,u_c]^T$$

ii) Linear;

$$\mathbf{a}^T=[a_{10},\ldots,a_{c0},a_{11},\ldots,a_{c1},\ldots,a_{cn}]$$
$$\mathbf{fx}=[u_1,\ldots,u_c,u_1x_1,\ldots,u_cx_1,\ldots,u_cx_n]^T$$

iii) Quadratic;

$$\mathbf{a}^T=[a_{10},\ldots,a_{c0},a_{11},\ldots,a_{c1},\ldots,a_{cn},\ldots,a_{cnn}]$$
$$\mathbf{fx}=[u_1,\ldots,u_c,u_1x_1,\ldots,u_cx_1,\ldots,u_cx_n,\ldots,u_cx_nx_n]^T$$

For a discriminant function of the form of Eq. (10), a two-class classifier implements the decision rule Eq. (9). Namely, x is assigned to ω_1 if the inner product $\mathbf{a}^T\mathbf{fx}$ exceeds zero and to ω_2 otherwise. The equation $g(\mathbf{x})=0$ defines the decision surface that separates points assigned to ω_1 from points assigned to ω_2 .

3.2 Learning using linear discriminant analysis

In order to identify the effect process of the PNN classifier, we take care of the learning algorithm using linear discriminate analysis. Linear discriminate analysis achieves a significant improvement in class separation, since it separates the class means while attempting to sphere the data classes [23].

Let us consider that a set of N samples $\mathbf{fx}_1,\ldots,\mathbf{fx}_N$ is given and these are labeled as ω_1 or ω_2 . We want to use these samples to determine the coefficients a in a discriminate function Eq. (10). A sample X_i is classified correctly if $\mathbf{a}^T\mathbf{fx}_i>0$ and $\mathbf{fx}_i$ is labeled ω_1 or if $\mathbf{a}^T\mathbf{fx}_i<0$ and $\mathbf{fx}_i$ is labeled ω_2. This suggests a normalization that simplifies the treatment of the two-class case, namely, the replacement of all samples labeled ω_2 by their negatives. With this normalization we can forget the labels and look for a coefficient vector, a, such that $\mathbf{a}^T\mathbf{fx}_i>0$ for all of the samples. Such a coefficient vector is called a separating vector or more generally a solution vector [22]. Seeking a vector making all of the inner products $\mathbf{a}^T\mathbf{fx}_i$ positive, $\mathbf{a}^T\mathbf{fx}_i=b_i$ can be considered. b_i is some arbitrarily specified positive constant. Then the problem is to find coefficient vector a satisfying

$$\mathbf{Xa}=\mathbf{b} \tag{11}$$

where, $\mathbf{X}=[\mathbf{fx}_1, \mathbf{fx}_2,\ldots,\mathbf{fx}_N]^T$ and $\mathbf{b}=[b_1,\ldots,b_N]^T$.

The coefficient vector a in the effect process of the PNN classifier can be determined by the standard least square method. Namely, the coefficient can be estimated by solving the optimization problem

$$\mathrm{Min}_a\ V(\mathbf{a},N) \tag{12}$$

$$V(\mathbf{a},N)=\|\mathbf{Xa}-\mathbf{b}\|^2=\sum_{i=1}^{N}(\mathbf{a}^T\mathbf{fx}_i-b_i)^2 \tag{13}$$

where N is the number of patterns.

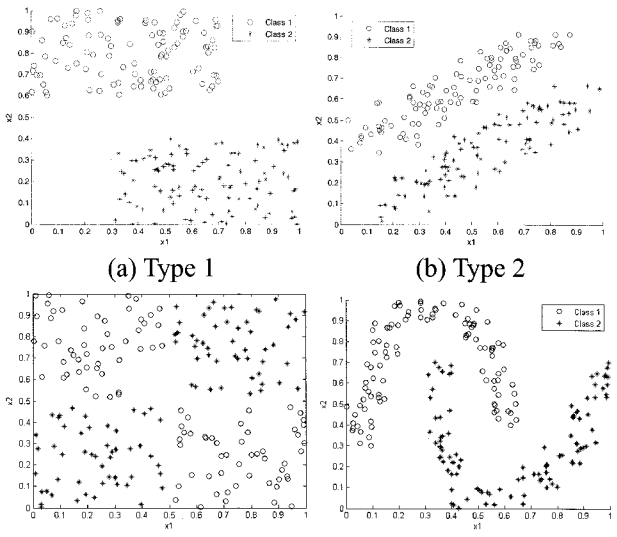
Owing to the criterion function in (13), the minimal value produced by the least square method is governed by the following expression:

$$\mathbf{a}=(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{b} \tag{14}$$

4. Experiments Studies

4.1 Synthetic datasets

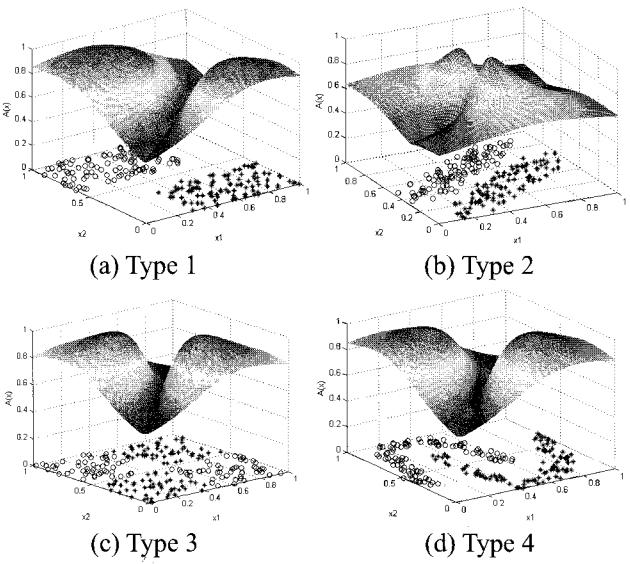
We start with a series of two-dimensional synthetic examples. Our primary objective is to illustrate the classification of the proposed PNN classifier. Four types of synthetic datasets are shown in Fig. 4.



(a) Type 1 (b) Type 2

(c) Type 3 (d) Type 4

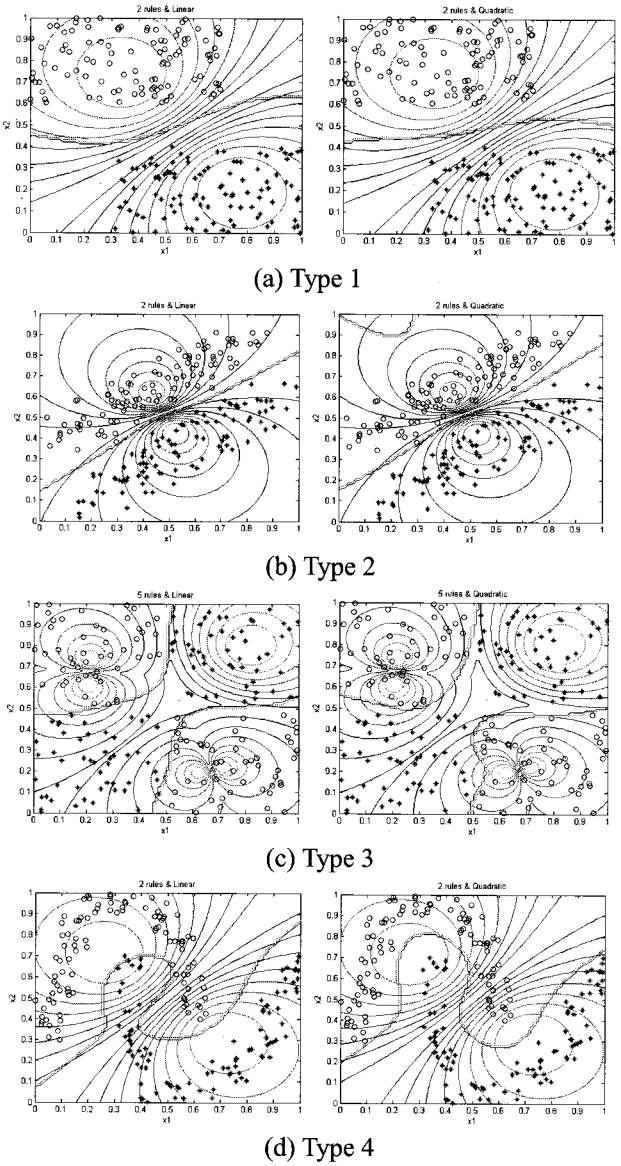
Fig. 4. Four types of synthetic datasets



(a) Type 1 (b) Type 2

(c) Type 3 (d) Type 4

Fig. 5. Two membership functions and datasets



(a) Type 1

(b) Type 2

(c) Type 3

(d) Type 4

Fig. 6. Results of classification of PNN classifier

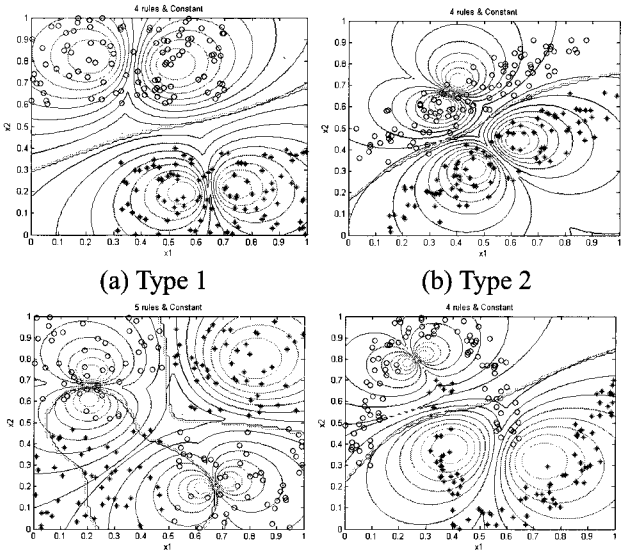
Fig. 5 presents two membership functions for each dataset. The values of membership functions are gotten by fuzzy c-means clustering, namely, two membership functions are the partition matrix for two clusters. The assumption process of the PNN classifier takes these functions for each dataset.

Table 1 shows the classification rate of the proposed PNN classifier compared with the RBF classifier for each dataset. The PNN classifier has better performance on a few rules than the RBF classifier.

The results of the best classification of the proposed PNN classifier are shown in Fig 6. There are two classes, boundary of classification and membership values. Membership values are contoured. Those of the RBF classifier are compared in Fig. 7. The PNN classifier is better than RBF for forming a boundary of classification.

Table 1. Classification rate of PNN classifier

Dataset	No. of Rules	RBF	Linear	Quadratic
Type 1	2	95.5 %	100 %	100 %
	3	99.5 %	100 %	100 %
	4	100 %	100 %	100 %
	5	97 %	100 %	100 %
Type 2	2	97.5 %	100 %	100 %
	3	86.5 %	100 %	100 %
	4	99.5 %	100 %	100 %
	5	98 %	100 %	100 %
Type 3	2	52 %	93.5 %	96.5 %
	3	67 %	94 %	98.5 %
	4	63 %	97.5 %	98.5 %
	5	82 %	99 %	99.5 %
Type 4	2	86.5 %	99 %	100 %
	3	84 %	98.5 %	100 %
	4	87.5 %	98.5 %	100 %
	5	87 %	98 %	100 %



(a) Type 1 (b) Type 2

(c) Type 3 (d) Type 4

Fig. 7. Results of classification of RBF classifier

4.2 Two-Spiral dataset

A two-spiral dataset has been chosen as it consists in delineating complex classification boundaries. In addition to the interest related to the problem complexity, this case study enables data visualization due to the low data dimensionality. It is a very well-known set taken as reference in many papers to show the ability of proposed classifiers [8]. It is a non-overlapping 2-dimensional set based on two classes, each of them composed of 97 patterns as shown in Fig. 9

Fig. 8 shows the classification rate of the proposed PNN and RBF classifiers according to number of rules. In the accuracy of classification, the PNN classifier has 89.7 %

on 26 rules and 100 % on 24 rules for the linear function and quadratic function, respectively. The RBF classifier has 66 % on 29 rules. The output of the PNN classifier as discriminant function g(x) and boundary of classification are shown in Fig. 9. Fig. 10 provides the outcome of classification of the RBF classifier with 29 rules. The proposed PNN classifier works very well for this application.

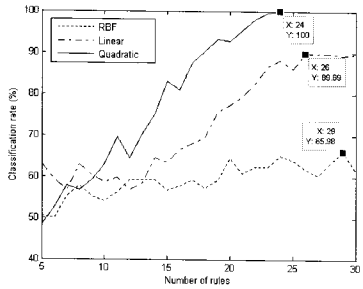
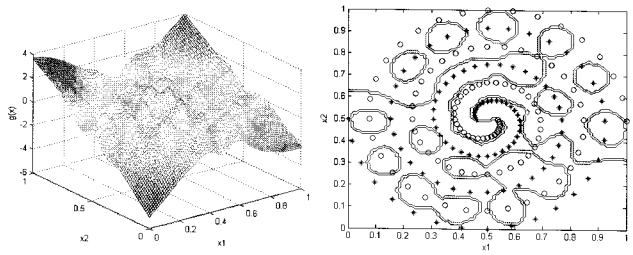
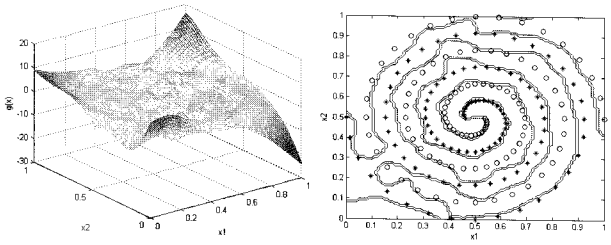


Fig. 8. Classification rate of PNN classifier for two-spiral dataset

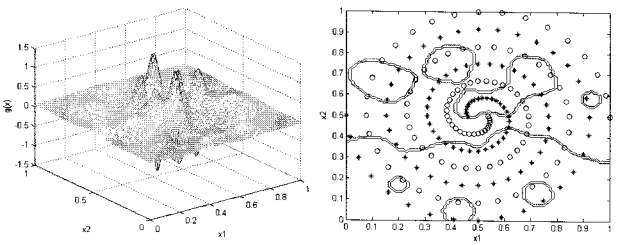


(a) 26 rules and linear function



(a) 24 rules and quadratic function

Fig. 9. g(x) and boundary of classification for PNN classifier



(a) discriminant function (b) boundary of classification

Fig. 10. Result of classification for RBF classifier with 29 rules

5. Conclusions

In this paper, a polynomial neural network (PNN) classifier based on fuzzy c-means clustering and polynomials has been proposed as a neural classifier such as multi-layer perceptron (MLP) and the radial basis function (RBF) network. The proposed PNN classifier is a trainable device consisting of some rules and two processes. The two processes are namely the assumption and effect processes. The assumption process is driven by fuzzy c-means clustering and the effect process deals with a polynomial function. These processes contribute directly to if-then rules that provide intuitional interpretation using the linguistic analysis for a given problem. The coefficients of a polynomial in the effect process of the PNN classifier are learned on the least squares method. The proposed PNN is applied and evaluated to four synthetic datasets and a spiral dataset. As shown in the results on the experimental studies, the proposed PNN performs better classification than RBF. The boundary of classification is formed very well by the PNN classifier for these applications. The successful results indicate that the proposed PNN classifier can be used as a reliable technique for developing neural classifiers for two-class problems.

Acknowledgements

This work has been supported by KESRI (R-2007-2-044), which is funded by MOCIE (Ministry of Commerce, Industry and Energy).

References

[1] C.-L. Liu and H. Sako, "Class-specific feature polynomial classifier for pattern classification and its application to handwritten numeral recognition", *Pattern Recognition*, vol. 39, pp. 669-681, 2006.

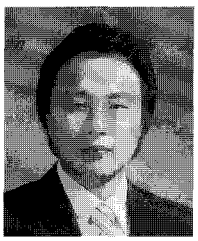
[2] G.E. Hinton, P. Dayan and M. Revow, "Modeling the manifolds of images of handwritten digits", *IEEE Trans. Neural Networks*, vol. 8, no. 1, pp. 65-74, 1997.

[3] J. Shurmann, *Pattern Classification: A Unified View of Statistical and Neural Approaches*, Wiley Interscience, New York, 1996.

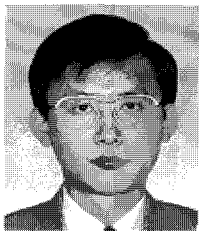
[4] U. Kreßel and J. Schurmann, "Pattern classification techniques based on function approximation", *in Handbook of Character Recognition and Document Image Analysis*, H. Bunke and P.S.P. Wang, Eds. World Scientific, Singapore, pp. 49–78, 1997.

[5] R. P. Lippman, "An introduction to computing with neural nets", *IEEE ASSP Magazine*, vol.4, no. 2, pp. 4-22, 1981.

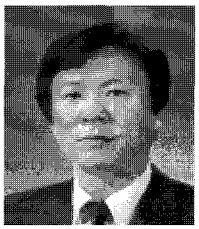
[6] A. Patrikar and J. Provence, "Pattern classification using polynomial networks", *Electronics Letters*, vol. 28, no. 12, pp. 1109-1110, 1992.
[7] A. Esposito, M. Marinaro, D. Oricchio and S. Scarpetta, "Approximation of continuous and discontinuous mappings by a growing neural RBF-based algorithm", *Neural Networks*, vol. 13, no. 6, pp. 651-665, 2000.
[8] F. Ros, M. Pintore and J.R. Chretien, "Automatic design of growing radial basis function neural networks based on neighborhood concepts", *Chemometrics and Intelligent Laboratory Systems*, vol. 87, pp. 231-240, 2007.
[9] H. Sarimveis, P. Doganis and A. Alexandridis, "A classification technique based on radial basis function neural networks", *Advances in Engineering Software*, vol. 37, pp. 218-221, 2006.
[10] X.-Y. Jing, Y.-F. Yao, D. Zhang, J.-Y. Yang and M. Li, "Face and palmprint pixel level fusion and Kernel DCV-RBF classifier for small sample biometric recognition", *Pattern Recognition*, vol. 40, pp. 3209-3224, 2007.
[11] M.J. Er, S.Q. Wu, J.W. Lu and H.L. Toh, "Face recognition with radical basis function (RBF) neural networks", *IEEE Trans. Neural Networks*, vol. 13, no. 5, pp. 697-710, 2002.
[12] W.M. Campbell, K.T. Assaleh and C.C. Broun, "Speaker recognition with polynomial classifiers", *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 4, pp.205-212, 2002.
[13] L. Devroye, L. Gyorfi and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer-Verlag, 1996.
[14] W. Rudin, *Principles of mathematical analysis*, McGraw-Hill, 1976.
[15] G.L. Giles and T. Maxwell, "Learning, invariance, and generalization in high-order neural networks", *Appl. Opt.*, vol. 26, no. 23, pp. 4972-4978, 1987.
[16] Y.H. Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison -Wesley, Reading, MA, 1989.
[17] Y. Al-Assaf and H. El Kadi, "Fatigue life prediction of composite materials using polynomial classifiers and recurrent neural networks", *Composite Structures*, vol. 77, pp. 561-569, 2007.
[18] C. Zhang, J. Jiang and M. Kamel, "Intrusion detection using hierarchical neural networks", *Pattern Recognition Letters*, vol. 26, pp. 779-791, 2005.
[19] A. Aiyer, K. Pyun, Y.Z. Huang, D.B. O'Brien and R.M. Gray, "Lloyd clustering of Gauss mixture models for image compression and classification", *Signal Processing: Image Communication*, vol. 20, pp. 459-485, 2005.
[20] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, N. York, 1981.
[21] S.-K. Oh, W. Pderycz and B.-J. Park, "Self-organizing neurofuzzy networks in modeling software data", *Fuzzy Sets and Systems*, vol. 145, pp. 165-181, 2004.
[22] R.O. Duda, P.E. Hart and D.G. Stork, *Pattern Classification*, 2nd ed., Wiley-Interscience, 2000.
[23] J.E. Munoz-Exposito, S. Garcia-Galan, N. Ruiz-Reyes, P. Vera-Candeas, "Adaptive network-based fuzzy inference system vs. other classification algorithms for warped LPC-based speech/music discrimination", *Engineering Applications of Artificial Intelligence*, vol. 20, pp. 783-793, 2007.



Byoung-Jun Park
He received his B.S., M.S., and Ph.D. degrees in Control and Instrumentation Engineering from Wonkwang University. His research interests include fuzzy modeling, neurofuzzy systems, intelligent control, pattern classification and recognition, biologically-inspired optimization strategies, and computational intelligence.



Sung-Kwun Oh
He received his B.S., M.S., and Ph.D. degrees in Electrical Engineering from Yonsei University. He is currently a Professor in the Dept. of Electrical Engineering, Suwon University, Korea. His research interests include automation systems, advanced Computational Intelligence, and intelligent control. He currently serves as an Associate Editor of KIEE Transactions on Systems & Control, International Journal of Control Automation and Systems of the ICASE, and International Journal of Fuzzy Logic and Intelligent Systems of KFIS, South Korea.



Hyun-Ki Kim
He received his B.S., M.S., and Ph.D. degrees in Electrical Engineering from Yonsei University. During 1999-2003, he worked as Chairman at the Korea Association of Small Business Innovation Research. He is currently a Professor in the Dept. of Electrical Engineering, Suwon University, Korea. His research interests include system automation and intelligent control.