J Electr Eng Technol.2018; 13(4): 1724-1731
http://doi.org/10.5370/JEET.2018.13.4.1724

ISSN(Print) 1975-0102
ISSN(Online) 2093-7423

Design of Space Search-Optimized Polynomial Neural Networks with the Aid of Ranking Selection and L2-norm Regularization

Dan Wang*, Sung-Kwun Oh† and Eun-Hu Kim**

Abstract – The conventional polynomial neural network (PNN) is a classical flexible neural structure and self-organizing network, however it is not free from the limitation of overfitting problem. In this study, we propose a space search-optimized polynomial neural network (ssPNN) structure to alleviate this problem. Ranking selection is realized by means of ranking selection-based performance index (RS_PI) which is combined with conventional performance index (PI) and coefficients based performance index (CPI) (viz. the sum of squared coefficient). Unlike the conventional PNN, L2-norm regularization method for estimating the polynomial coefficients is also used when designing the ssPNN. Furthermore, space search optimization (SSO) is exploited here to optimize the parameters of ssPNN (viz. the number of input variables, which variables will be selected as input variables, and the type of polynomial). Experimental results show that the proposed ranking selection-based polynomial neural network gives rise to better performance in comparison with the neuron fuzzy models reported in the literatures.

Keywords: Space Search-Optimized Polynomial Neural Network (ssPNN), Ranking Selection-Based Performance Index (RS_PI), Polynomial Neural Network (PNN), Space Search Optimization (SSO), L2-norm Regularization.

1. Introduction

Nowadays, computational intelligence techniques including neural network, fuzzy models, and evolutionary algorithm have been successfully used in many fields [1-6]. As one of the classical neural network, Group Method of Data Handling (GMDH) initialized by Imasaki [7] have been proposed to alleviate highly dimensional problem encountered in many application fields such as nonlinear model. In the design of GMDH [8-10], each new neuron is generated based on two neurons coming from previous layer, while the output neuron is calculated in form of polynomial. However, GMDH has its own drawbacks: (1) Unreasonable selection of input variables resulting in ineffective dimensionality; (2) Unreasonable parameter learning mechanism. To alleviate these problems, Oh et al. Proposed polynomial neural networks (PNN) [11]. As an artificial neural network model, PNN has more flexible selection mechanism in comparison to GMDH. Unlike each neuron consists of only two input variables in the GMDH, each neuron can consists of three and more input variables in the PNN. In the design of PNN, neuron contains different polynomial expressions (such as linear polynomials, quadratic polynomials, cubic polynomials, etc.) and the parameter learning mechanism is achieved by the least squares method (LSE). In spite of advantages, PNN is not free from limitation: its parameter learning mechanism leads to overfitting problem, and its neuron selection mechanism is not able to be optimized by self-organizing limited ability.

In this paper, we propose a space search-optimized polynomial neural network (ssPNN) to overcome these aforementioned problems. A ranking selection-based performance index (RS_PI) that is combination of the traditional performance index (PI) and coefficient based performance index (CPI) are proposed to select the neurons of each layer. The stable performance index is realized by the sum of squared coefficient which is the coefficient of polynomial and calculated by using LSE with L_2-norm regularization [12]. L_2-norm regularization is exploited here to estimate the value of coefficient. By adding the penalty term to the cost function, L_2-norm regularization can alleviate the overfitting problem. Moreover, space search optimization (SSO) is exploited here to optimize the structure parameters of neurons (viz. the number of input variables, which variables will be selected as input variables and the type of polynomial) in the ssPNN.

The structure of this paper is organized as follows. Section 2 introduces motivation and related works. Section 3 presents the design of ssPNN. Section 4 proposes ssPNN by means of SSO. Section 5 reports the experimental results. At last, conclusions are shown in section 6.

† Corresponding Author: Dept. of Electrical Engineering, The University of Suwon, Korea and Key Laboratory of Complex Systems and Intelligent Computing in Universities of Shandong, Linyi University, China. (ohsk@suwon.ac.kr)
* School of Computer Science and Information Engineering, Tianjin University of Science &Technology, China. (wanghzc@163.com)
** Dep. of Electrical Engineering, The University of Suwon, Korea. (wdkim@suwon.ac.kr)
Received: December 12, 2017; Accepted: February 13, 2018

2. Motivation and Related Works

To construct the ssPNN, we first illustrate the motivation and contribution of this study, and then recall the classical polynomial neural network studied previously and space search optimization. The mainly four important parts in the design of classic PNN are first introduced, and then the overall flowchart of SSO are summarized.

2.1 Motivation and contribution of this study

The motivation of this study is to overcome the limitation of overfitting problems encountered in the design of conventional Polynomial neural networks. In short, the contributions can be summarized as follows:

First, coefficients-based performance index (CPI) and L2-norm regularization are invoked in the design of polynomial neural networks. Unlike the conventional performance index (PI) that is used as selection criteria of polynomial neurons for the next layer, CPI can reflect the complexity of the polynomial neurons. Furthermore, L2-norm regularization can alleviated the overfitting problem when estimating the coefficients of polynomials by using least square method.

Second, a ranking selection-based mechanism for the choice of polynomial neurons is proposed. The proposed ranking selection-based performance index (RS_PI) is realized by combination of PI and CPI. In this case, the RS_PI takes into consideration both performance (PI) and complexity (CPI), and results in the diversity of the polynomial neurons.

Third, the overall architecture of ssPNN by space search optimization is proposed. On the one hand, space search optimization initialized by our previous study is applied to the design of ssPNN. Such new optimization used in the ssPNN brings about some novelty. On the other hand, the ssPNN can be regarded as the conventional PNNs with novel ranking selection-based polynomial neurons (RS_PNs). That is, in some senses the proposed ssPNN are essentially generalized PNNs.

2.2 L2-norm regularization

Sometimes the polynomial neural networks lead to overfitting problem with the increasing the number of layers [11]. In fact, lots of fuzzy neural models have been encountered the overfitting problem. In this case, L2-norm regularization is one of effective method to alleviate this problem. Assume that $(x_1, x_2, ..., x_k)$ and y represent the input and output of training data, then the L2-norm regularization-based least square error algorithm can be summarized as shown in Algorithm 1.

Algorithm 1. L2-norm regularization-based least square error algorithm

Input: $(x_1, x_2, ..., x_k)$, Y
Output: Coefficient matrix A
1.Set regularization factor λ
2.Set $\mathrm{X} = (1, x_1, x_2, ..., x_k)$
3.Estimate the coefficients A by using the formula
A= $(\mathrm{X}^{\mathrm{T}}\mathrm{X}+\lambda\mathrm{I})^{-1}\mathrm{X}^{\mathrm{T}}\mathrm{Y}$

2.3 Ranking-based selection of neurons

In the conventional PNNs, the polynomial neurons with the best performance (viz. the smallest error) in each layer. Such mechanism can guarantee the rapid convergence of the polynomial neural networks. However, the excessive rapid convergence also leads to local optimum. That is, the polynomial neurons in the same layer have height similarity. To overcome this limitation, we use the following selection mechanism: some polynomial neurons are the neurons with the best performance, the other polynomial neurons are those neurons with the small sum of coefficients that represent the simplest neurons.

With this idea, there still exist one problem: how to define the selection criteria (function) of neurons. The difficulty of selection criteria is that the ranges between performance index and the sum of coefficient are great difference. In this study, we propose ranking-based selection mechanism to address this problem. Algorithm 2 describes the overall flowchart of ranking-based selection algorithm.

Algorithm 2. Ranking-based selection algorithm

Input: Polynomial neurons in each layer
Output: Selected polynomial neurons in each layer
1. Set ranking factor $\gamma\%$ and number of selection neurons NS
2. Sort all polynomial neurons based on performance index
3. Select the $\gamma\% \times NS$ neurons with the best performance
4. Sort all polynomial neurons based on sum of coefficients
5. Select the $(1-\gamma\%) \times NS$ neurons with the smallest sum of coefficients

2.4 Space search optimization

Space Search optimization (SSO) is an evolutionary algorithm that is based on the space search mechanism of to get the candidate solution [13-14]. As one of the evolutionary algorithms, SSO is also obtained the optimal individual by mean of selection mechanism. However, the SSO utilize the space search mechanism, which are quite different from the evolutionary mechanisms used in the conventional evolutionary algorithms. The space search mechanism are realized by means of generating a new space and searching the new space, in which space search operators are mainly two parts, namely local space search (part I) and global search (part II).

The two parts of operator can be summarized as follows:

Part I : Space search is based on the selection of M solutions, the new generated space is:

$$V = \{X^{new} \mid x_i^{new} = \sum_{k=1}^{M} a_i x_i^{new} \cup X^{new} \in S, \quad where \sum_{k=1}^{M} a_i = 1, -0.5 \le a_i \le 1.5\} \tag{1}$$

Where M is the predefined number, X^{new} denotes the new generated solution, a_i are randomly generated and the ranges are from -0.5 to 1.5, S represents the entire solution space.

Part II: Space search is based on only one the best solution, the new generated space is:

$$V_1=\{(x_1^{new},x_2^{new},...,x_n^{new}) | x_j^{new}=x_j\,(j \ne i) \cup x_i^{new} \in [l_i,u_i]\} \tag{2}$$

where $X^{new} = (x_1^{new}, x_2^{new}, ..., x_n^{new})$. The value of x_i is range from l_i to u_i.

3. Design of Polynomial Neural Network Using Ranking Selection and L2 Norm Regularization

This section elaborates on the design of ssPNN. The main comparison of the classical PNN and the proposed ssPNN are summarized as shown in Table 1.

Table 1. Comparison of constructing methodologies in the design of PNN and ssPNN

	Constructing approaches	PNN	ssPNN
Common point	Type of neurons	Polynomials	Polynomials
Different point	Learning method	LSE	LSE+ L2-norm regularization
	Selection criteria of neurons	PI	*RS_PI

*RS_PI represents ranking selection-based performance index defined in section 3.3.

3.1 Architecture of polynomial neural networks

Polynomial neural network is an artificial neural network model which can estimate the relationship of multivariate high order polynomials between input and output. PNN [11] is a dynamic structure of the neural network, the network topology layer can also be amended by the new layer.

3.2 Polynomial type of neurons

Like polynomial types used in the classic polynomial neural network contains, the polynomial types used in the ssPNN are as follows:

1) (Constant): $f_j = a_{j0}$
2) (Linear Polynomials): $f_j = a_{j0} + a_{j1}x_1 + \cdots + a_{jk}x_k$
3) (Quadratic Polynomial):

$$f_j = a_{j0} + a_{j1}x_1 + \cdots + a_{jk}x_k + a_{j(k+1)}x_1^2 + \cdots + a_{j(2k)}x_k^2 + a_{j(2k+1)}x_1x_2 + \cdots + a_{j((k+2)(k+1)/2)}x_{k-1}x_k$$

4) (ModifiedQuadratic Polynomial) :

$$f_j = a_{j0} + a_{j1}x_1 + \cdots + a_{jk}x_k + a_{j(k+1)}x_1x_2 + a_{j(k(k+1)/2)}x_{k-1}x_k f_j = a_{j0}$$

From the four polynomial type, where $a_{j0}, a_{j1}, \cdots, a_{jk}, a_{j(k+1)}, a_{j(k(k+1)/2)}$ are the coefficients of the polynomial, respectively.

It is apparent that the highest order of four polynomial is equal to 2, which is lower in comparison with the 3-order polynomial used in the conventional PNN. With the use of relative low polynomial order, the proposed ssPNN gives rise to flexible structure.

3.3 Learning method using L2 norm regularization

For convenience, we consider a linear format of polynomial in the following way:

$$\mathrm{y}=\mathrm{a}_0 + \mathrm{a}_1 x_1 + \mathrm{a}_2 x_2 + ... + \mathrm{a}_k x_k \tag{3}$$

Where y denotes the output data, $x_1, x_2, ..., x_k$ represents the input variables, respectively. And $\mathrm{a}_0, \mathrm{a}_1, \mathrm{a}_2, ..., \mathrm{a}_k$ denotes the coefficient of polynomial, respectively. For convenience, the expression can be described as follows:

$$\mathrm{Y=AX} \tag{4}$$

where

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1k} \\ 1 & x_{21} & x_{22} & \ldots & x_{2k} \\ & & \ldots & & \\ 1 & x_{i1} & x_{i2} & \ldots & x_{ik} \\ . & & . & & . \\ 1 & x_{n1} & x_{n2} & \ldots & x_{nk} \end{bmatrix} \quad A = \begin{bmatrix} a_0 \\ \ldots \\ a_i \\ \ldots \\ a_k \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ \ldots \\ y_i \\ \ldots \\ y_n \end{bmatrix}.$$

According to the LSE method, the coefficient can be calculated by the following expression:

$$\mathrm{A=(X^TX)^{-1}X^TY} \tag{5}$$

To alleviate the overfitting problem, we use L_2–norm regularization method that adds the penalty term to the cost function in the following way:

$$\mathrm{A=(X^TX+\lambda I)^{-1}X^TY} \tag{6}$$

where I is the unit matrix, λ denotes a predetermined number. In this study, λ is fixed as 0.01.

The coefficients of polynomial are estimated by the L_2–norm regularization-based LSE method.

3.4 Ranking-based selection criteria of neurons

The neuron selection criteria for the next layer in the design of ssPNN is realized by a combination of performance index and coefficients based performance index. The expression of PI is as follows [14-15]:

$$PI = \begin{cases} \dfrac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 (MSE) \\ \sqrt{\dfrac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} (RMSE) \end{cases} \tag{7}$$

Here MSE denotes mean squared error, and RMSE stands for the root mean squared error.

The stable performance index is essentially the sum of squared coefficient of polynomial. The expression of CPI is as follows:

$$CPI = \sum_{i=1}^{m} a_i^2 \tag{8}$$

The expression of RS_PI combined with PI and CPI is in the following way:

$$RS\_PI = \begin{cases} 1, & if \ V(PI \ or \ CPI) \le N/2, \\ 0, & otherwise. \end{cases} \tag{9}$$

where $V(PI)$ represents the ranking number of performance index, $V(CPI)$ denotes the ranking number of stable performance index. N stands for the number of selected neurons for the next layer.

For any neurons, If $RS\_PI = 1$ denotes the neuron will be selected for the next layer, while $RS\_PI = 0$ represents the neuron will not be selected.

4. Space Search Optimization of Ranking Selection-based Polynomial Neural Network

Space search optimization has been optimized fuzzy systems. In this study, SSO is explored here to optimize the ssPNN. In what follows, we first introduce the design of SSO-based polynomial neurons, and then discuss the construction of ssPNN by means of SSO.

4.1 Design of space search-optimized polynomial neurons

In the process of design ssPNN, the structure of polynomial neuron should be optimized. The structure parameters of polynomial neuron contains the number of input variables, which variables will be selected as input variables, and the type of polynomials. Figure 1 describes arrangement of solution for the optimization neuron using

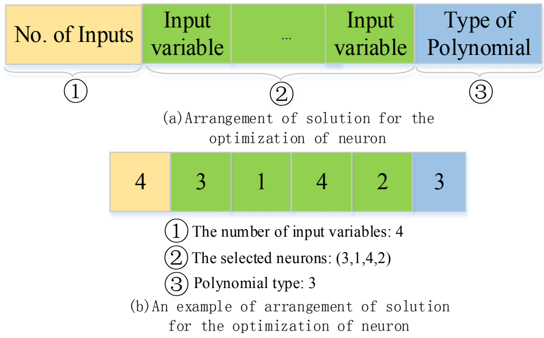

Fig. 1. Arrangement of solution for the optimization of polynomial neuron

SSO.

1) ***Number of input variables***. The number of input variables would be selected as the next layer's input. The value is randomly generated from 2 to 5.

2) ***Input variables to be selected***. The one of each input variables represents the index of input variables. The largest number is set as the maximum column coming from data set.

3) ***Polynomial types***. There are four types of polynomial can be selected. The polynomial type is randomly selected from one to four.

4.2 Design of space search-optimized polynomial neural networks

In this part, we elaborate the details of SSO optimized ssPNN structures. Fig. 2 describes the overall optimization process of ssPNN with the aid of SSO.

[Step 1] *Partition data sets into training data and testing data.*

Training data and testing data. In this study, the original dataset is divided into two parts. The first 60% data is considered as training data, while the remaining 40% is regarded as testing data.

[Step 2] *Initialize structure parameters of ssPNN.*

Two structure parameters will be initialized. They are as follows: 1) the number of layers; and 2) the number of selected neurons for each layer. In this study, the number of layers is set as 10, while the number of neurons is fixed as 30.

[Step 3] *Constructing polynomial neurons by space search optimization.*

Step 3.1. Initialization. The population is initialized by randomly generated individuals. The structure of a general individual is shown in Figure 3.

Step 3.2. Evaluation. The performance of each individual is evaluated by using the aforementioned RS_PI. Here an individual is essentially denotes a neuron.

Step 3.3. Selection. The individuals with best performance (viz. RS_PI) are selected.

Step 3.4. Space search (Case I). This space search operator is realized based on four individuals.

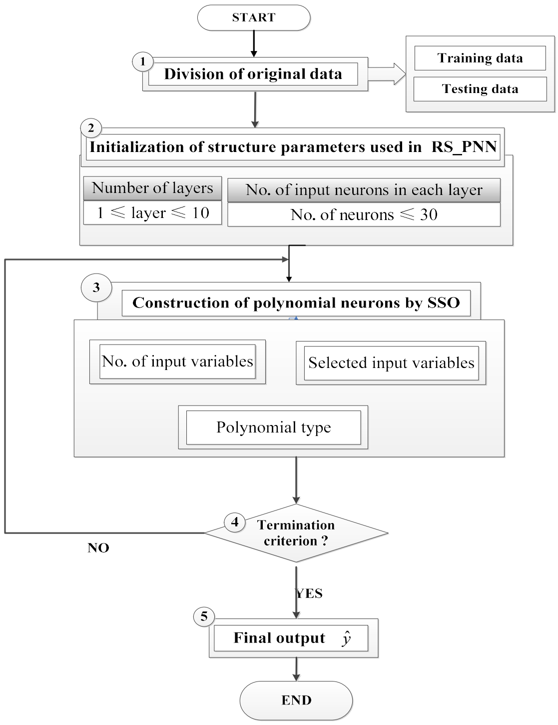

Fig. 2. Overall flowchart of ssPNN by SSO

Step 3.5. Space search (Case II). This space search operator is realized by means of the best individual in the current population.

[Step 4] *Implement the iterations until the terminal condition is satisfied*.

If the terminal condition is not satisfied, go to Step 3, otherwise go to Step 5. Here the terminal condition is as follows:

(1) If the number of iterations arrives at a predetermined number, then the algorithm will be terminated; or

(2) If the performance of the current layer is worse than the previous layer, then the algorithm will be terminated.

[Step 5] *Report the output*

Output the optimal ssPNN.

5. Experimental Studies

In this study, three well-known datasets (viz. MO, IZ, ABA) are utilized to evaluate the performance of the proposed method. All experiments on datasets are based on five-fold cross-validation. The data sets are divided into two parts: the 80% of data is used as training data and the remaining 20% is considered as testing data. In order to compare to other models, RMSE and MSE/2 are viewed as the performance index for data sets. SSO are used to optimize the ssPNN. According to the values of parameters in the literature [13], the values of parameters in SSO are selected as shown in Table 2. Here NP denotes the number of population, NS represents the number of solutions for SSO operators, and NI stands for the number of iterations.

Table 2. Settings of parameters in the SSO algorithm

Parameters	NP	NS	NI
Setting values	100	8	30

Table 3. Experimental results of ssPNN and PNN (MO)

Layers	PNN		ssPNN	
	PI	EPI	PI	EPI
LF=1	0.018 ±0.0007	0.018 ±0.003	0.020 ±0.005	0.021 ±0.004
LF=2	0.014 ±0.001	0.014 ±0.003	0.010 ±0.0004	0.010 ±0.003
LF=3	0.013 ±0.002	0.014 ±0.002	0.010 ±0.0005	0.010 ±0.003
LF=4	0.013 ±0.002	0.014 ±0.002	0.009 ±0.0007	0.010 ±0.004
LF=5	0.013 ±0.002	0.014 ±0.002	0.009 ±0.0007	0.010 ±0.004

Table 4. Comparative results of ssPNN and some previous models (MO)

Model		PI	EPI
MEA-FIS[18]	$NSGA\text{-}II_{RB}$	0.11±0.06	0.16±0.16
	$NSGA\text{-}II_{KB}$	0.09±0.04	0.13±0.12
	$PAES_{RB}$	0.06±0.03	0.08±0.05
	$PAES_{KB}$	0.05±0.02	0.09±0.10
	$SOGA_{RB}$	0.38	0.64
	$SOGA_{KB}$	0.28	0.66
MEA-FIS[18]	$W_M(3)$	0.985±0.129	0.973±0.09
	$W_M(5)$	0.128±0.05	0.134±0.012
	$W_M(7)$	0.095±0.006	0.137±0.056
	$G_{R\text{-}MF}$	0.03±0.002	0.176±0.28
	$G_{A\text{-}WM}$	0.02±0.003	0.093±0.147
	$G_{LD\text{-}WM}$	0.016±0.002	0.022±0.005
	FS_{MOGFS}^{e}	0.033±0.004	0.034±0.007
ssPNN	LF= 1	0.020±0.005	0.021±0.004
	LF = 2	0.010±0.0004	0.010±0.003
	LF = 3	0.010±0.0005	0.010±0.003
	LF = 4	0.009±0.0007	0.010±0.004
	LF = 5	0.009±0.0007	0.010±0.004

5.1MO Data

Mortgage (MO) data contains the economic information of USA [16-17]. It owns 15 input variables, and 1049 patterns. Table 3 illustrates the experiment results of ssPNN with two types. Table 3 describes the comparative analysis of the performance of the other models and the proposed ssPNN model.

Table 3 describes the values of the performance index of testing data with the increasing number of layers. It shows that both of PNN and ssPNN are stable in case of EPI (testing error). These results demonstrate that the overfitting problem are not happened in this data set.

The comparison result of ssPNN and some other models are summarized in Table 4. As shown in Table 4, the proposed ssPNN obtains the better performance when compared with some conventional models. Fig. 3 depicts the optimal architecture of ssPNN as shown in Table 4.

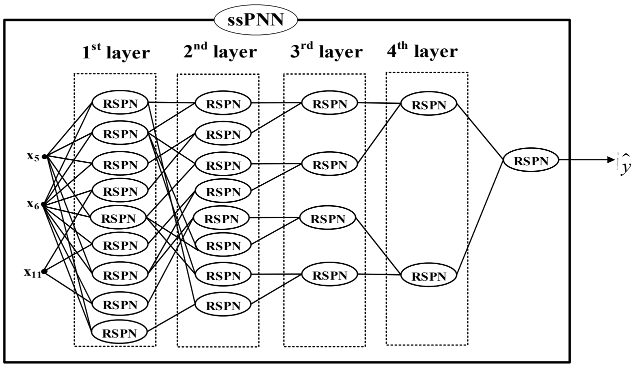

Fig. 3. Optimal ssPNN architecture for MO data

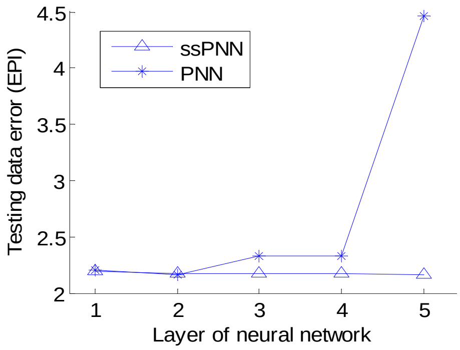

Fig. 4. Comparison of test error with the increasing number of layers (ABA)

5.2ABA data

Abalone machine learning (ABA) data is concerned with the estimate the age of abalone when encounter the physical measurements [20]. In this data set, it contains 4177 input-output pairs and 7 input variables.

Fig. 4 depicts the relationship between the performance index of testing data (viz. EPI) and the number of layers for the proposed ssPNN and the conventional PNN. As shown in Fig. 4, EPI of ssPNN are almost not changed with the increasing number of layers when compared with the PNN Such phenomenon illustrates the good stability of the structure of the ssPNN.

The result of our proposed method ssPNN compared to other models is further described in Table 5. It is apparent that the performance of ssPNN much better in comparison with that of the conventional models. Unlike the conventional fuzzy models in reference [25], the proposed ssPNN are realized based on the architectures of neural networks. Moreover, instead of the conventional evolutionary algorithms, space search optimization initialized by us [13] is used as the optimal algorithm in the ssPNN. These results demonstrate the outstanding feature of the ssPNN.

5.3 IZ data

Weather Izmir (IZ) data is the data about the weather information [18-19]. It contains 9 input variables and 1461 patterns. Fig. 5 depicts the results of ssPNN and the conventional PNN in five layers. As shown in Fig. 5, it is evident that the proposed ssPNN leads to better performance than the conventional PNN in case of overfitting problem.

Table 5. Comparative results of ssPNN and some previous models (ABA)

Model		PI	EPI
Linear regression[21]		14.15±0.07	17.22±0.20
Boosting of granular model [21]	P=5, C=6	8.39±0.008	8.68±0.014
RBFNN with context-free clustering [22]		5.52±0.25	6.91±0.45
PSO-based PNN [22]		4.338±0.231	7.997±9.465
RBFPNN[22]		3.605±0.169	4.710±0.224
Linguistic Modeling[23]	Without optimization	5.21±0.12	6.14±0.28
	One-loop optimization	4.80±0.52	5.22±0.58
	Multi-step optimization	4.12±0.35	5.32±0.96
Evolutionary Fuzzy Systems [24]		2.41±0.085	2.51±0.186
L2-norm TSK fuzzy models [25]		1.879	2.558
LEL-TSK fuzzy models [25]		2.040	2.412
METSK+HD [25]		2.205	2.392
ssPNN	LF = 1	2.195±0.024	2.198±0.095
	LF = 2	2.168±0.026	2.172±0.100
	LF = 3	2.167±0.027	2.171±0.101
	LF = 4	2.165±0.025	2.169±0.104
	LF = 5	2.161±0.023	2.166±0.106

Table 6. Comparative results of ssPNN and some previous models (IZ)

Model		PI	EPI
MEA-FIS [18]	$NSGA\text{-}II_{RB}$	2.05±0.71	2.38±0.96
	$NSGA\text{-}II_{KB}$	1.64±0.34	1.91±0.48
	$PAES_{RB}$	1.62±0.43	1.89±0.54
	$PAES_{KB}$	1.30±0.27	1.49±0.26
	$SOGA_{RB}$	4.17	4.81
	$SOGA_{KB}$	3.65	4.89
MGA-FIS [19]	$W_M(3)$	6.944±0.72	7.368±0.909
	$W_M(5)$	3.107±0.27	5.961±2.498
	$W_M(7)$	2.036±0.048	10.56±2.197
	$G_{R\text{-}MF}$	1.176±0.077	9.602±8.879
	$G_{A\text{-}WM}$	1.233±0.065	3.529±4.023
	$G_{LD\text{-}WM}$	0.926±0.041	1.150±0.123
	FS_{MOGFS}^{e}	1.519±0.094	1.571±0.168
FWPNN [26]	CFWN	0.703±0.064	1.285±0.876
	FFWN	0.667±0.052	0.855±0.133
SsPNN	LF= 1	0.850±0.036	0.858±0.145
	LF = 2	0.762±0.019	0.778±0.120
	LF = 3	0.751±0.017	0.771±0.116
	LF = 4	0.750±0.016	0.779±0.108
	LF = 5	0.747±0.016	0.798±0.100

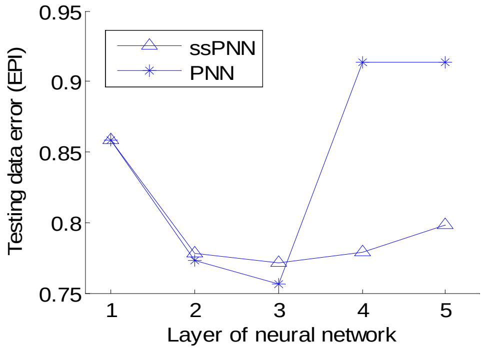

Fig. 5. Comparison of test error with the increasing number of layers (IZ)

The tendency of EPI and the number of layers is depicted as shown in Fig. 5. It illustrates that the ssPNN are relative not dramatically changed in case of EPI. These results demonstrate that the overfitting problem in the ssPNN are alleviated in comparison with the conventional PNN.

Table 6 summarizes the comparative results between the proposed ssPNN and some other previous models.

6. Conclusions

This study aims at a design of stable polynomial neural networks that can alleviate the overfitting problems, which are universal problems in the design of conventional neural networks. The main differences between the ssPNN and the conventional PNN are summarized as follows:

First, instead of the conventional PI, a concept of RS_PI is further proposed to be viewed as the selection criteria of neurons for the next layer in the design of neural network.

Second, L_2–norm regularization is invoked when estimating the polynomial coefficients by least square error method.

Finally, space search optimization is further exploited here to realize the structure optimization and parameter optimization in the design of ssPNN.

For future study, fuzzy polynomial neurons can be included. By constructing different fuzzy neurons, one can obtain space search optimized fuzzy polynomial neural networks.

Acknowledgements

This work was supported by the Foundation of Educational Commission of Tianjin City, China (Grant no. 20140803), and supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) by the Ministry of Education (Grant no. NRF-2017R1D1A1B03032333).

References

[1] C. Riziotis and A.V. Vasilakos, “Computational intelligence in photonics technology and optical networks: a survey and future perspectives,” *Information Sciences*, vol. 177, pp. 5292-5315, 2007.

[2] W. Huang and L. Ding, “Project-Scheduling problem with random time-dependent activity duration times,” *IEEE Transactions on Engineering Management*, vol. 58, pp. 377-387, 2011.

[3] B. Wu, S. Han, J. Xiao, X. Hu, and J. Fan, “Error compensation based on BP neural network for airborne laser ranging,” *International Journal for Light and Electron Optics* vol. 127, pp. 4083-4088, 2016.

[4] W. Huang, J. Wang, and J. Liao. “A granular classifier by means of context-based similarity clustering,” *Journal of Electrical Engineering & Technology*, vol. 11, pp. 1383-1394, 2016.

[5] S. Yilmaz and Y. Oysal. “Fuzzy wavelet neural network models for prediction and identification of dynamical systems,” *IEEE Transactions on Neural Networks*, vol. 21, pp. 1599-1609, 2010.

[6] E. Kaslik and L.R. Radulescu. “Dynamics of complex-valued fractional-order neural networks,” *Neural Networks*, vol. 89, pp. 39-49, 2017.

[7] A.G. Ivakhnenko, “Polynomial theory of complex systems,” *IEEE Trans Systems, Man, and Cybernetics*, SMC-vol. 1, pp. 364-378, 1971.

[8] K.T. Parveen, B. Sanghamitra, and K.P. Sankar, “Multi-Objective Particel Swarm Optimization with time variant inertia and acceleration coefficients,” *Information Sciences*, vol. 177, pp. 5033-5049, 2007.

[9] S. Masato, and G. Mitsuo, “Fuzzy multiple objective optimal system design by hybrid genetic algorithm,” *Applied Soft Computing,* vol. 2, pp. 189-196, 2003.

[10] M. Delgado, M.P. Ceullar, and M.C. Pegalajar, “Multiobjective Hybrid Optimization and Training of Recurrent Neural Networks,” *IEEE Trans. Syst., Man cybern. –Part B*, vol. 38, pp. 381-403, 2008.

[11] S.-K. Oh, W. Pedrycz, and B.-J. Park, “Polynomial Neural Networks architecture: analysis and design,” *Computers and Electrical Engineering*, vol. 29, pp. 703-725, 2003.

[12] SH Zhou, XW Liu, Q Liu, SQ Wang, CZ Zhu, and JP Yin,“Random Fourier extreme learning machine with l(2,1)-norm regularization,” *Neuro Computing*, vol. 174, pp. 143-153, 2016

[13] W. Huang and S.-K. Oh. “Identification of fuzzy inference systems using multi-objective space search algorithm and information granulation,” *Journal of Electrical Engineering & Technology*, vol. 6, pp. 853-866, 2011.

[14] W. Huang and J. Wang, “Design of polynomial fuzzy radial basis function neural networks based on non-symmetric fuzzy clustering and parallel optimi-

zation," *Mathematical Problems in Engineering*, vol. 2013, pp. 1-11, 2013

[15] Z. Yong, W. Xiaobei, X. Zongyi, and H. Weili, "On generating interpretable and precise fuzzy systems based on Pareto multi-objective cooperative co-evolutionary algorithm," *Applied Soft Computing,* vol. 11, pp. 1284-1294, 2011

[16] S.-K. Oh and W. Pedrycz, "Fuzzy polynomial neuron-based self-organizing neural networks," *Int. J. of General Systems*, vol. 32, pp. 237-250, 2003.

[17] J.-N. Choi, S.-K. Oh, and W. Pedrycz, "Identification of fuzzy models using a successive tuning method with a variant identification ratio," *Fuzzy Sets and Systems*, vol. 159, pp. 2873-2889, 2008

[18] R. Alcala, P. Ducange, F. Herrera, B. Lazzerini, and Marcelloni F, "A multiobjective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy-rule-based systems," *IEEE Transactions on Fuzzy Systems,* vol. 17, pp. 1106-1122, 2009.

[19] R. Alcala, M. J. Gacto, and F. Herrera," A fast and scalable multiobjective genetic fuzzy System for linguistic fuzzy modeling in high-dimensional regression problems," *IEEE Transactions on Fuzzy Systems*, vol. 19, pp. 666-681, 2011.

[20] W. Pedrycz, and K.-C. Kwak, "The Development of Incremental Models," *IEEE Trans. Fuzzy Systems*, vol. 15, pp. 507-518, 2007.

[21] W. Pedrycz, and K.-C. Kwak, "Boosting of granular models," *Fuzzy Sets and Systems*, vol. 157, pp. 2943-2953, 2006.

[22] J.S.R. Jang and C.T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems," *IEEE Trans. Neural Networks*, vol. 4, pp. 156-158, 1993

[23] W. Pedrycz, and K.-C. Kwak, "Linguistic models as a framework of user-centric system modeling," *IEEE Trans. SMC-A*, vol. 36, pp. 727-745, 2006

[24] M. Antonelli, P. Ducange, and F. Marcelloni, "An efficient multi-objective evolutionary fuzzy system for regression problems," *International Journal of Approximate Reasoning*, vol. 54, pp. 1434-1451, 2013.

[25] J. Liu, F. Chung, and S. Wang, "Bayesian zero-order TSK fuzzy system modeling," *Applied Soft Computing*, vol. 55, pp. 253-264, 2017.

[26] W. Huang, S.-K. Oh, and W. Pedrycz, "Fuzzy wavelet polynomial neural networks: analysis and design," *IEEE Trans. Fuzzy Systems*, vol. 25, no. 5, pp. 1329-1341, 2017.

Dan Wang She received her Ph.D. degree in School of Computer, Wuhan University. She is currently a full time lecturer in the School of Computer Science and Information Engineering, Tianjin University of Science and Technology. Her research interests include evolutionary computation and fuzzy neural networks.

Sung-Kwun Oh He received his B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Yonsei University, Seoul, Korea, in 1981, 1983, and 1993, respectively. During 1983-1989, he was a Senior Researcher of R&D Lab. of Lucky-Goldstar Industrial Systems Co., Ltd. From 1996 to 1997, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada. He is currently a Professor with the Department of Electrical Engineering, University of Suwon, Suwon, South Korea. His research interests include fuzzy system, fuzzy-neural networks, automation systems, advanced computational intelligence, and intelligent control. He currently serves as an Associate Editor of the *KIEE Transactions on Systems and Control*, *International Journal of Fuzzy Logic* and *Intelligent Systems of the KFIS*, and *Information Sciences*.

Eun-Hu Kim received his B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Suwon University, Hwaseong-si, Korea, in 2009, 2011, and 2016, respectively. He is currently a postdoctoral fellow at electrical engineering, Suwon University, Korea. His research interests include fuzzy set, neural networks, deep learning, evolutionary computing, computational intelligence and statistical learning.