

Feature Selection Using Evolutionary Functional Link Neural Network for Classification

Amaresh Sahu¹, Sabyasachi Pattnaik²

¹Computer Science Engineering Department, SOA University Bhubaneswar, Odisha 751030, India

²Information & Communication Technology Department, Fakir Mohan University, Balasore, Odisha 756019, India

Article Info

Article history:

Received Sep 17, 2017

Revised Nov 18, 2017

Accepted Nov 23, 2017

Keyword:

Functional link neural network

Genetic algorithm

Particle swarm optimization

ABSTRACT

Computational time is high for Multilayer perceptron (MLP) trained with back propagation learning algorithm (BP) also the complexity of the network increases with the number of layers and number of nodes in layers. In contrast to MLP, functional link artificial neural network (FLANN) has less architectural complexity, easier to train, and gives better result in the classification problems. The paper proposed an evolutionary functional link artificial neural network (EFLANN) using genetic algorithm (GA) by eliminating features having little or no predictive information. Particle swarm optimization (PSO) is used as learning tool for solving the problem of classification in data mining. EFLANN overcomes the non-linearity nature of problems by using the functionally expanded selected features, which is commonly encountered in single layer neural networks. The model is empirically compared to MLP, FLANN gradient descent learning algorithm, Radial Basis Function (RBF) and Hybrid Functional Link Neural Network (HFLANN). The results proved that the proposed model outperforms the other models.

Copyright © 2017 Institute of Advanced Engineering and Science.

All rights reserved.

Corresponding Author:

Amaresh Sahu,
Computer Science Engineering Department,
SOA University Bhubaneswar,
Odisha 751030, India.
Email: amaresh_sahu@yahoo.com

1. INTRODUCTION

Classification task is one of the most important research area of data mining. A lot of research has focused on the field over the last several years [1- 3]. The Data mining is a process of discovering knowledge from large gamut of data. Various classification models in this regard. M. James [4] has used a linear or quadratic discriminates techniques for solving problems in classification. Another procedure has been employed by using decision trees [5, 6]. In the same perspective Duda et al. [7] has proposed discriminant analysis based on the Bayesian decision theory, however he has mentioned traditional statistical models are built mainly on various linear assumptions. To overcome such disadvantage, artificial intelligent tools have been emerged to solve classification problems in data mining. For this reason, many genetic algorithms models were used [8]. In a recent study, Zhang has introduced the neural networks technique as a powerful classification tool [9-10]. In recent year Artificial Neural Networks (ANNs) have gained expensive popularity in research activities. ANNs are generally implemented as a data-based techniques in performing nonlinear system identification of complex processes. ANNs have been used increasingly as a promising modeling tool in almost all human related activity areas where quantitative approaches can be used to help in decision making e.g functional approximation, rule extraction, pattern recognition and classification, forecasting and prediction, business electrical engineering, medical and civil engineering. Multilayer

perceptrons (MLPs) are the most common type of neural networks applied successfully for identification and control of a variety of nonlinear processes. Various research activities have been done from Mcculloch-pitts model of neuron [11] to higher order neuron [12] are considerable and literature is growing massively day by day. Feed forward networks which based on single layer of linear threshold logic units exhibit fast learning. e.g the simple perceptron [13] can solve a linearly separable problem in a finite number of learning steps, however, it does not converge or approximate nonlinear decision making problems. By using gradient descent learning algorithm to adjust the weights adline [14-15] tries to find a solution which minimise a mean square error criterion. But it discrimination capability is also limited to linearly separable problems. Many real problems involve approximating nonlinear functions or forming multiple nonlinear decision regions limits the applicability of simple single layered networks of linear threshold unit. By adding of a layer of hidden units dramatically increased the power of layered feed forward networks. In MLP using the back propagation (BP) learning algorithm has been successfully applied to many application problems. However, the training speed of MLPs are typically slower than those for feed forward networks consist of a single layer of linear threshold units due to back propagation of error introduced by multi layering. However problems occur in MLP training are local minima trapping, saturation, weight interference, initial weight dependence and over fitting. Also it is very difficult to fix the parameters like number of neurons in a layer and number of hidden layers in a network. So deciding a proper architecture is difficult. An easy way to avoid these mentioned problems consisting of removing the hidden layers. The removing process should be carried out without giving up non linearity. Provided that the input layer is en-downed with additional higher order units [16-18]. In other words higher order correlations among input components can be used to construct a higher order networks to perform nonlinear mapping using only a single layer of units [16]. All these indicate the field of HONs [19-20] like functional link neural networks [21-24], pi-sigma neural networks [25-26], radial basis functions [27,28] and polynomial neural networks (RPNNs) [29,30] and so on.

A direction is given by pao and pao et al, that functional links neurons may be conveniently used for function approximation with faster convergence rate and lesser computational load than MLP [31]. FLANN with gradient descent method has achieved good results in classification task of data mining. Also, different set of orthogonal basis functions have been suggested for feature expansion [32]. FLANN is basically a flat network without requiring hidden layers and has simple learning rule. For enhancing the classification accuracy two FLANN based classifiers with genetic algorithms have been developed [33-34]. In FLANN the dimensionality of input vector is also effectively increased by using functional expansion of the input vector and hence the hyper planes generated by the FLANN providing greater discriminating capability of input patterns. Although FLANN using the gradient descent gives promising results, sometimes it may be trapped in local optimal solutions. Moreover FLANN coupled with genetic algorithms may suffer with problems like heavy computational burdens and large number of parameters tuning. Then PSO has been implemented for classification [35] suffers from problems like parameter tuning and complexity in network architecture. The discovered knowledge from the data should be predictive and comprehensible. The architectural complexity of FLANN [22] is directly proportional to number of features and the functions used for expansion of the given feature value. Knowledge comprehensibility is useful for at least two related reasons. First, the knowledge discovery process usually assumes that the discovered knowledge will be useful for supporting a decision to be made by a user. Second if the discovered knowledge is not comprehensible to the user, he/she will not be able to validate it, obstructing the interactive feature of the knowledge discovery process includes knowledge validation and refinement. The proposed method for classification is given an equal importance to both predictive accuracy and comprehensibility. We measure comprehensibility of the proposed method by reducing the complexity in architecture. The extracted knowledge is used by us for supporting a decision making process that is the ultimate goal of data mining. We choose MLP, FLANN with gradient descent learning [22], Radial Basis Function (RBF) and Hybrid FLANN (HFLANN) [43] for results comparison with Proposed method.

2. CONCEPTS AND DEFINITIONS

2.1. Genetic algorithms

Genetic algorithms defined as a search technique was inspired from Darwinian Theory. The scheme is based on the theory of natural selection. Here we presume that there is a population composed with different characteristics. Inside the population the stronger will be able to survive and they pass their characteristics to their offsprings.

The process of GA is described as follows:

- a. Initial population is generated randomly;
- b. Fitness function are used to evaluate the population;
- c. Genetic operators such as selection, crossover, and mutation are applied;
- d. Repeat the process "Evaluation Crossover mutation" until stopping criteria is met

2.2. Particle Swarm Optimization

PSO is relatively a new population based search technique for solving numerical optimization problems. It inspired from the complex social behavior that is shown by the natural species like flock of birds. Each member of swarm called particle associated with a position and a velocity. In the multidimensional search space the particles fly and look for a potential solution. Each particle adjusts its position in the search space frequently according to the flying experience of its own and its neighbors [36]. The movement of each particle is governed by the following two Equations (1) and (2), in n-dimensional search space.

$$v_{ij}(t+1) = w.v_{ij}(t) + c_1.rand1(pbest_{ij}(t) - x_{ij}(t)) + c_2.rand2(gbest_j(t) - x_{ij}(t)) \quad (2.1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (2.2)$$

Where, x_i represents the position of the i^{th} particle; v_i represents the velocity of the i^{th} particle; $pbest_i$ be the previous best particle of the i^{th} particle and $gbest$ is the global best particle found by all particles until now. $rand1$ and $rand2$ are two random vector values within $[0, 1]$, w represents a inertia weight, c_1 and c_2 are two learning factors, the value of each velocity vector is compressed within the range $[v_{\max}, v_{\min}]$ to decrease the likelihood of the particle leaving the search space, and $t=1,2,3\dots$, indicates the number of iterations.

The constriction factor approach of PSO (CPSO) is an important variant of standard PSO, It was proposed by Clerc and Kennedy [37-38]. The velocity of CPSO is updated by the following Equation:

$$v_{ij}(t+1) = \chi.(v_{ij}(t) + c_1.rand1.(pbest_{ij}(t) - x_{ij}(t)) + c_2.rand2.(gbest_j(t) - x_{ij}(t))) \quad (2.3)$$

Where, χ is called a constriction factor, given by Equation:

$$\chi = \frac{2}{|2 - \varphi + \sqrt{\varphi^2 - 4\varphi}|} \quad (2.4)$$

Where,

$$\varphi = c_1 + c_2, \quad \varphi > 4 \quad (2.5)$$

The CPSO ensures the convergence of the search procedures and be able to generate higher quality solutions than the standard PSO is used in this study.

2.3. Functional Link Neural Network

The FLANN architecture was first proposed by Pao et al. [31]. The idea of this model is to apply an expansion function to increases the dimensionality of input vector. Hence the hyper-planes generated provide superior discrimination capability in the input pattern space. The need of hidden layer is removed by applying expansion function thus the learning algorithm become simpler. This model has the benefit to have faster convergence rate and lesser computational cost than MLP structure. The conventional nonlinear functional expansions can be employed in FLANN are trigonometric, power series or Chebyshev type. R. Majhi et al. [39], illustrates that use of trigonometric function expansion provides superior prediction capability of the model. Therefore, in the present situation, trigonometric expansion is employed. Let us assume each element of the input pattern before expansion is represented as $Z(i)$, $1 < i < I$ where I is the total number of features and each element $Z(i)$ is functionally expanded as $Xn(i)$, $1 < n < N$, where N is the total number of expanded points for each input element. Let us take, $N=5$ in this example. The expansion of each input pattern is represented in Figure 1.

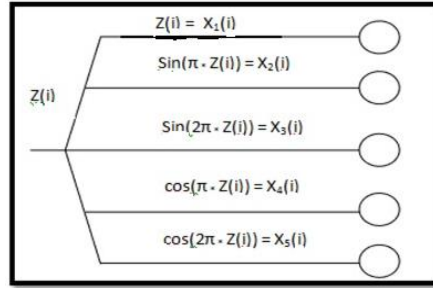


Figure 1. Represents the Expansion of Each Input Pattern

The expanded inputs are fed to the single layer neural network and then the network is trained by using training algorithm to obtain the desired output.

2.4. Proposed Method

The proposed single layer ANN named as EFLANN with a genetically optimized set of features. It has the potential of generating complex decision regions by nonlinear improvement of hidden units referred to as functional links. Figure 2 shows the topological structure of the EFLANN. This method is characterized by a set of FLANN with a different subset of features. Let take the number of original features of the data domain is n and the number of features selected as a chromosome of the genetic population is d_f , where $d_f \leq n$. The d_f varies in the genetic population from chromosomes to chromosomes (i.e. $1 \leq d_f \leq n$). For simplicity, let us discuss how a single chromosome with d_f features is working cooperatively for EFLANN.

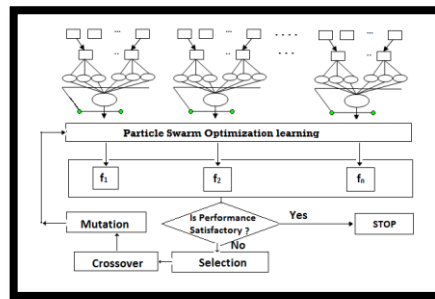


Figure 2. Topological Structure of the EFLANN

In this proposed method, the general trigonometric function is used for mapping the d_f features from low to high dimension. In order to use a function that is very close to the underlying distribution of the data some prior domain knowledge is required. Here five functions are taken out of which four functions are trigonometric and one function is linear (i.e., keeps the original form of the feature value). Within the four trigonometric functions two functions are sine and other two functions are cosine. By using the trigonometric functions the domain is feature values and range is a real number lies between $[-1,1]$. It can be represented as

$$f: \text{Domain} \rightarrow \text{Range}^{[-1,1] \cup \{x\}} \tag{4.1}$$

Where, $\text{Domain} = \{ x_{i1}, x_{i2}, \dots, x_{idf} \}$, and d_f represents the number of features. Let us take f_1, f_2, \dots, f_v be the number of functions that are used to expand each feature value of the pattern. Hence, now each input pattern can be represented as

$$\bar{x}_i = \{ x_{i1}, x_{i2}, \dots, x_{idf} \} \rightarrow \{ \{ f_1(x_{i1}), f_2(x_{i1}), \dots, f_v(x_{i1}) \}, \{ f_1(x_{i2}), f_2(x_{i2}), \dots, f_v(x_{i2}) \} \dots \{ f_1(x_{idf}), f_2(x_{idf}), \dots, f_v(x_{idf}) \} \} = \{ \{ y_{11}, y_{21}, \dots, y_{v1} \}, \{ y_{12}, y_{22}, \dots, y_{v2} \} \dots \{ y_{1df}, y_{2df}, \dots, y_{vdf} \} \} \tag{4.2}$$

The weight vector exist between hidden layer and output layer is multiplied with the resultant sets of nonlinear outputs and are supplied to the output neuron as an input. Therefore, the weighted sum is calculated as follows:

$$\text{Sum} = \sum_{j=1}^m y_{ij} \cdot w_j \quad (4.3)$$

Where, $i=1,2,\dots,N$ and m represents the total number of expanded features.

The network has the ability to learn by using PSO training process. The network training needs a set of training data, i.e., a series of input and related output vectors. During the training process, the data is repeatedly presented to the network and weights are adjusted by PSO from time to time till the desired input to output mapping is obtained. Therefore the estimated output is calculated as follows:

$$\text{yout}_i(t) = f(\text{sum}_i), \text{ where, } i = 1,2,\dots,N \quad (4.4)$$

The error is calculated for i^{th} pattern of training set as

$$\text{error}_i(t) = y_i(t) - \text{yout}_i(t), \text{ where, } i = 1,2,\dots,N \quad (4.5)$$

So, the error criterion function is calculated as

$$\text{Er}(t) = \sum_{i=1}^N \text{error}_i \quad (4.6)$$

Here, our objective function is to minimize the error by adjusting weights through CPSO until $\varepsilon \leq \text{Er}$, where ε is very small value. This process is applied repeatedly for each chromosomes of the GA and subsequently, based on the performance of each chromosome will be assigned with a fitness value. Using this fitness value the usual process of GA is executed until we find some good topology which can achieve an acceptable predictive accuracy.

2.5. High Level Algorithms for EFLANN

The requirements for the near optimal EFLANN architecture and related parameters can be obtained by using both genetic algorithms and Particle Swarm Optimization algorithm for learning. Evolutionary genetic algorithm uses stochastic search and optimization methods. So GA based on fundamental process, such as reproduction, recombination and mutation. An algorithm of this type starts with a set (population) of estimates (genes), called individuals (chromosomes) encoded properly. For solving the classification task of data mining the fitness of each one is evaluated correctly. The best fit individuals are allowed to make and bear offspring during each iteration of algorithm. The PSO is a population based algorithm used here to update the weights in learning process. During evolutionary process the length of each particle has upper bound n , represents the feature vector. Each cell of the chromosome holds binary value either 0 or 1, where the cell value 1 represents the activation and 0 represents deactivation in functional expansion of individuals. During evolution each individual measures its effectiveness by the error criterion function that is given in Equation (4.6) and then the predictive accuracy is assigned as it corresponding fitness.

2.6. Pseudo code for EFLANN

The steps that are followed by EFLANN can be described as follows:

1. Division of Dataset
Divide the dataset into training and testing parts
2. Random Initialization of Individual
Each individual Initialize each individual randomly from the domain $\{0,1\}$
3. While (Termination criterion is not met)
4. For The Population
 - 4.1 For each sample of the training set
 - 4.2 Mapping of Input Pattern
Map each pattern from low to high dimension by expanding each feature value according to the predefined set of functions.
 - 4.3 Calculation of Weighted Sum
Calculate the weighted sum and feed as an input to the node of the output layer.
 - 4.4 Calculate The Error
Calculate the error and accumulate it.
 - 4.5 Use Particle Swarm Optimization for Learning
Minimize the error by adjusting weights using PSO.
 - 4.6 Assign The Fitness Value
5. For The Population
 - 5.1 Roulette Wheel Selection is performed to obtain the better chromosomes.
6. For The Population Obtained
 - 6.1 Perform recombination
 - 6.2 Mutation
7. While End

3. EXPERIMENTAL STUDIES

The performance of the EFLANN model was evaluated using a set of seven benchmark classification datasets like IRIS, WINE, PIMA, BUPA, ECOLI, GLASS and VOWEL from the University of California at Irvine (UCI) machine learning repository [40]. For classifying six overlapping vowel classes, we have also taken VOWEL dataset to show the performance of EFLANN [41]. We have used MLP, FLANN with gradient descent learning, RBF and HFLANN results comparison with proposed method. Table 1 summarizes the main characteristics of the databases that have been used in this paper.

Table .1. Summary of the Datasets used in Simulation Studies

Serial No.	Name of Dataset	No. of patterns	No. of attribute	No. of classes
1	IRIS	150	4	3
2	WINE	178	13	3
3	PIMA	768	8	2
4	BUPA	345	6	2
5	ECOLI	336	7	8
6	GLASS	214	9	6
7	VOWEL	871	3	6
8	ZOO	101	16	7

3.1. Parameter Setup

Before evaluating the proposed EFLANN algorithm, we need to set the user defined parameters and protocols related to the dataset. Here we have used two fold cross validation for all the dataset by randomly dividing the total set of patterns of dataset into two equal parts as dataset1.dat and dataset2.dat. Each of these two sets of dataset was alternatively used either as a training set or a test set [22]. The quality of each individual of dataset is measured by the predictive performance obtained during training. We need to set the optimal values of the following parameters to reduce the local optimality. The parameters we have used are described as follows: Population size: The size of the population designated as $|P|=50$ is fixed for all the datasets. The number of input features is n . Hence length of the individuals is fixed to n . The probability for crossover is taken as 0.7 and the probability for mutation is 0.02. The total number of iterations we have applied is 1000 for all the datasets. Parameters of PSO for EFLANN training process are set similarly as neural network training with PSO [42]. The MLP uses two hidden layers for all datasets. In each hidden layer $P \times Q$ numbers of neurons are taken. Where, P and Q are the number of input neurons and output neurons.

3.2 Comparative Performance

The predictive performance achieved from proposed EFLANN for the above mentioned datasets were compared with the results obtained from MLP, FLANN with gradient descent learning, RBF and HFLANN results. Table 2 and Table 3 summarize the comparative average training and test performances of EFLANN with other above mentioned methods.

Table 2. Average Comparative Performance of EFLANN, MLP and RBF

Datasets	Algorithms					
	EFLANN		MLP		RBF	
	Training	Testing	Training	Testing	Training	Testing
IRIS	98.4710	97.4505	97.1510	94.0000	38.5000	38.5000
WINE	99.7560	90.5600	96.1410	92.2920	85.3935	79.2130
PIMA	80.8220	72.6343	76.6100	77.1920	77.4740	76.0415
BUPA	77.9830	69.3798	67.5220	67.3900	71.0125	66.9530
ECOLI	55.3630	50.9730	46.9600	45.2650	31.1780	26.1100
GLASS	63.7590	51.7525	43.2840	40.9800	48.9865	34.6440
VOWEL	40.8210	38.6635	38.2500	37.4500	25.2555	24.3250
ZOO	99.4532	86.4624	96.3500	92.3200	96.1540	81.0830

Table 3. Average Comparative Performance of EFLANN, HFLANN and RBF

Datasets	Algorithms		HFLANN		FLANN	
	EFLANN Training	Testing	Training	Testing	Training	Testing
IRIS	98.4710	97.4505	98.0001	97.3335	96.6664	96.6664
WINE	99.7560	90.5600	99.4380	90.4495	97.1910	88.7640
PIMA	80.8220	72.6343	80.7290	72.1355	79.5570	72.1355
BUPA	77.9830	69.3798	77.6820	69.2785	77.9725	69.2800
ECOLI	55.3630	50.9730	55.1670	50.8020	49.9625	47.3075
GLASS	63.7590	51.7525	63.5565	51.5075	60.7510	50.3800
VOWEL	40.8210	38.6635	40.4395	38.1965	27.9250	24.7220
ZOO	99.4532	86.4624	99.0385	86.1850	97.1155	85.1645

From Table 2 and Table 3, we can easily verify that except BUPA case in all other cases on an average the proposed method EFLANN is giving assuring results in both training and test cases. FLANN is performing better in case of BUPA dataset. Figure 3 shows the percentage of feature selected by the EFLANN compare to HFLANN in the context of comprehensibility. The datasets are represented by using X-axis and percentage of active bits in the optimal chromosome obtained during the training are represented by using Y-axis.

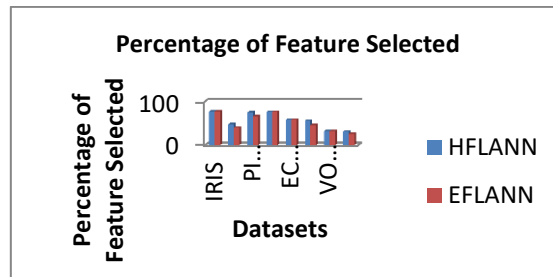


Figure 3. Represents the Percentage of Optimal set of Selected Features in EFLANN and HFLANN

3.3. Predictive Accuracy by Knowledge Incorporation

Let us take n be the total number of features exist in the dataset, F₁ and F₂ denote the number of feature selected by using the training set 1 and testing set 2 alternatively. Notations with Meaning: |N| denotes the total number of features in the dataset, |F₁| represents the total number of selected features in test set 2. |F₂| represents the total number of selected features in test set 1 and |PA| represents the predictive accuracy.

The fitness of the chromosome related to F₁ is

$$f(F_1) = \frac{|PA| \times |N| - Tr \times |F_1|}{|N|} \tag{5.1}$$

Similarly, the fitness of the chromosome related to F₂ is

$$f(F_2) = \frac{|PA| \times |N| - Tr \times |F_2|}{|N|} \tag{5.2}$$

Tr represents the predictive accuracy and tradeoff between two criteria and its value is taken as Tr = 0.01.

Table 4. Predictive Accuracy of EFLANN by Knowledge Incorporation with Tr =0.01

Datasets	N	Predictive Accuracy of Test Set 1 Chromosome	
		Predictive Accuracy of Test Set 1	Predictive Accuracy of Test Set 1 Chromosome
IRIS	4	96.1001	97.3260
WINE	13	90.0110	91.1264
PIMA	8	71.6525	72.9648
BUPA	6	70.5346	68.9602
ECOLI	7	54.8939	46.9020
GLASS	9	57.4374	45.9824
VOWEL	3	35.1263	40.1395
ZOO	16	87.9492	84.9212

4. CONCLUSION

In this study, the proposed EFLANN method is used for the task of data mining. It has given importance to the selection of optimal set of features and it has also enhanced the classification accuracy. The classification performance of EFLANN model is promising as demonstrated in experimental studies. Here the architectural complexity is low but, training time is more than FLANN. Also it is shown that EFLANN has better feature selection capability than HFLANN. Therefore, this model can be used in classification tasks of data mining.

REFERENCES

- [1] Fayyad, U. And Uthurusamy, R.: Data Mining and Knowledge Discovery in Databases. Commun.ACM, 39: 24-27. 1996.
- [2] Fayyad, U. M., Piatetsky-Shapiro, G., And Smyth, P.: From Data Mining to Knowledge Discovery: An Overview. In U. M. Fayyad, G. Piatetsky- Shapiro, & P. Smyth (Eds.), Advances in Knowledge Discovery and Data Mining ,Pp. 1–34,. Menlo Park, CA: AAAI Press ,1996.
- [3] Agrawal, R., Imielinski, T., And Swami A.: Database Mining: A Performance Perspective. IEEE Trans. Knowledge Data Eng, 5, Pp. 914-925, 1993.
- [4] James, M.: Classification Algorithms.Wiley, 1985.
- [5] Breiman, L., Friedman, J.H., Olshen, R.A., And Stone C.J.: Classification and Regression Learning. Morgan Kaufman, 1984.
- [6] Kriegel, H.P., Et Al.: Future Trends in Data Mining. Data Mining and Knowledge Discovery, 15(1), 87–97. Netherlands: Springer, 2007.
- [7] Duda, R.O., Hart, P.E. and Stork D.G.: Pattern Classification. Wiley, New York, 2001.
- [8] Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Morgan Kaufmann, 1989.
- [9] Zhang, G.P.: Neural Networks for Classification, A Survey. IEEE Transactions on Systems, Man, Cybernetics-Part C: Application and Reviews, 30(4), Pp. 451–461, 2000.
- [10] Zhang, G.P.: Avoiding Pitfalls in Neural Network Research. IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews, 37(1), Pp. 3–16, 2007.
- [11] McCulloch, W., Pitts, W.: A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics 7, 115-133, 1943.
- [12] Giles C. L., Maxwell, T., Learning Invariance, and Generalization in A Higher Order Neural Network. Applied Optics 26(23), 4972-4978, 1987.
- [13] Minsky, M., Papert, S.: Perceptrons.The MIT Press, 1969.
- [14] Widrow, B., Hoff, M. E.: Adaptive Switching Circuits. IRE WESCON Convention Record, 96-104, 1960.
- [15] Widrow, B., Lehr, M.: 30 Years Of Adaptive Neural Networks: Perceptron, Madaline, and Back-Propagation. Proceedings Of IEEE 78(9), 1415-1442, 1990.
- [16] Giles, C. L., Maxwell, T.: Learning, Invariance and Generalization in Higher-Order Neural Networks. Appl. Opt. 26(23), 4972-4978, 1987.
- [17] Pao, Y. H.: Adaptive Pattern Recognition and Neural Network. Reading, MA: Addison-Wesley, 1989.
- [18] Young, M.: The Technical Writer's Handbook. Mill Valley, CA University Science, 1989.
- [19] Venkatesh, S. S. Baldi, P.: Programmed Interactions in Higher Order Neural Networks: Maximal Capacity. J. Complexity 7, 316-337, 1991.
- [20] Antymov, E. And Pecht, O. Y.: Modified Higher Order Neural Network for Invariant Pattern Recognition. Pattern Recognition Letters 26 , 843-851, 2005.
- [21] Pao, Y.H., Philips, S. M.: The Functional Link Net Learning Optimal Control. Neuro Computing 9,149-164, 1995.
- [22] Misra, B. B., Dehuri, S.: Functional Link Neural Network For Classification Task in Data Mining. Journal of Computer Science 3(12), 948-955, 2007.
- [23] Mirea, L., Marcu, T.: System Identification Using Functional Link Neural Networks With Dynamic Structure. 15th Triennial World Congress, Barcelona, Spain, 2002.
- [24] Cass, R., Radl, B.: Adaptive Process Optimization Using Functional Link Networks and Evolutionary Algorithms. Control Eng. Practice 4(11), 1579-1584, 1996.
- [25] Shin, Y., Ghosh, J.: The Pi-Sigma Networks: An Efficient Higher Order Neural Network for Pattern Classification and Function Approximation. Proceedings of International Joint Conference on Neural Networks, 13-18, 1991.
- [26] Shin, Y., Ghosh, J.: Computationally Efficient Invariant Pattern Recognition With Higher Order Pi-Sigma Networks. The University of Texas At Austin, Tech. Report, 1992.
- [27] Zhu, Q., Cai, Y., Liu, L.: A Global Learning Algorithm For A RBF Network. Neural Networks 12, 527-540, 1999.
- [28] Li, M., Tian, J., Chen, F.: Improving Multiclass Pattern Recognition With A Co-Evolutionary RBFNN. Pattern Recognition Letters 29, 392-406, 2008.
- [29] Shin, Y., Ghosh, J.: Ridge Polynomial Networks. IEEE Transactions Neural Networks 6(2), 610-622. 1995.
- [30] Shin, Y., Ghosh, J.: Approximation Of Multivariate Functions Using Ridge Polynomial Networks. Proceedings of International Joint Conference on Neural Networks II, 380-385, 1992.
- [31] Pao, Y.H., Phillips, S.M., Sobajic, D.J.: Neural-Net Computing and Intelligent Control Systems, International Journal of Control Systems 56(2), 263-289, 1992.

- [32] Mishra, B.B., Dehuri, S.: Functional Link Artificial Neural Network For Classification Task in Data Mining. *Journal of Computer Science* 3, 948-955, 2007.
- [33] Dehuri, S., Cho S.B.: Evolutionary Optimized Features in Functional Link Neural Network For Classification. *Expert System And Applications* 37, 4379-4391, 2010.
- [34] Dehuri, S., Cho S.B.: A Hybrid Genetic Based Functional Link Artificial Neural Network With A Statistical Comparison of Classifiers Over Multiple Datasets. *Neural Computing Applications* 19, 317-328, 2010.
- [35] Dehuri, S., Roy R., Cho S.B., Ghosh A.: An Improved Swarm Optimized Functional Link Artificial Neural Network (ISO-FLANN) For Classification. *The Journal of Systems And Software* 85, 1333-1345, 2012.
- [36] Eberhart, R.C., Kennedy J.: A New Optimizer Using Particle Swarm Theory. *Proceedings Of The Sixth International Symposium On Micro Machine And Human Science, Nagoya, Japan*, Pp. 39-43, 1995.
- [37] Clerc M., Kennedy J.: The Particle Swarm Explosion, Stability and Convergence In A Multidimensional Complex Space. *IEEE Trans Evol Comput* 2002,6(1), 58-73, 2002.
- [38] Eberhart, R., Shi, Y.: Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. *Proceedings of The Congress On Evolutionary Computation (CEC2000)*, Pp. 84-88, 2000.
- [39] Majhi R., Panda G. And Sahoo G.: Development and Performance Evaluation Of FLANN Based Model For Forecasting of Stock Markets, *Expert Systems With Applications* 36, Pp. 6800-6808, 2009.
- [40] Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases, [Http://www.Ics.Uci.Edu/~Mlearn/MLrepository.Html](http://www.ics.uci.edu/~mllearn/mlrepository.html)
- [41] Pal, S.K., Majumdar, D.D.: Fuzzy Sets and Decision Making Approaches in Vowel and Speaker Recognition. *IEEE Transactions on Systems, Man, And Cybernetics* 7, 625-629, 1977.
- [42] Carvalho, M. And Ludermir, T.B.: Hybrid Training of Feed-Forward Neural Networks With Particle Swarm Optimization, I. King Et Al. (Eds.): *ICONIP 2006, Part II, LNCS 4233*, Pp. 1061-1070, 2006.
- [43] Dehuri, S., Mishra, B.B., Cho S.B.: Genetic Feature Selection For Optimal Functional Link Artificial Neural Network in Classification. *IDEAL 2008, LNCS 5326*, Pp.156-163, 2008.
- [44] Hsu, C.W., Lin, C.J.: A Comparison Of Methods For Multi-Class Support Vector Machines. *IEEE Transaction on Neural Networks*.3 (2), Pp. 415-425. 2002.