ISSN 2007-9737

Improving the Multilayer Perceptron Learning by Using a Method to Calculate the Initial Weights with the Similarity Quality Measure Based on Fuzzy Sets and Particle Swarms

Lenniet Coello[1], Yumilka Fernandez[1], Yaima Filiberto[1], Rafael Bello[2]

[1] Universidad de Camagüey, Department of Computer Sciences,
Cuba

[2] Universidad Central de Las Villas, Department of Computer Sciences,
Cuba

{lenniet.coello, yumilka.fernandez, yaima.filiberto} @reduc.edu.cu, rbellop@uclv.edu.cu

Abstract. The most widely used neural network model is Multilayer Perceptron (MLP), in which training of the connection weights is normally completed by a Back Propagation learning algorithm. Good initial values of weights bear a fast convergence and a better generalization capability even with simple gradient-based error minimization techniques. This work presents a method to calculate the initial weights in order to train the Multilayer Perceptron Model. The method named PSO+RST+FUZZY is based on the similarity quality measure proposed within the framework of the extended Rough Set Theory that employs fuzzy sets to characterize the domain of similarity thresholds. Sensitivity of BP to initial weights with PSO+RST+FUZZY was studied experimentally, showing better performance than other methods used to calculate feature weights.

Keywords. Multilayer perceptron, weight initialization, similarity quality measure, fuzzy sets.

1 Introduction

The development of information technologies has permitted a rapid growth in digital information. This has brought about a great demand of automated analysis of data to improve decision-making process in every aspect of human life. Rough Set Theory (RST) proposed by Pawlak in 1982 offers measures for analysis of data. The measure called classification quality is applied when the RST is used to construct the evaluation function. This measure allows calculating the consistency of a decision system. Its main limitation is its use only for decision systems where the feature domain is discrete.

A new measure for the case of decision systems in which the feature domain, including the decision feature, does not have to be necessarily discrete, is proposed in [1]. This new measure named Similarity Quality Measure represents the similarity degree among the objects of a heterogeneous decision system.

Besides, a method for constructing similarity relations from the combination of metaheuristic optimization based on particles (Particle Swarm Optimization, PSO) is proposed in [2] and [3]; as well as the similarity quality measure, which is used as an heuristic evaluation function. The process of construction of the similarity relation includes feature weight calculation. The impact of the method called PSO+RST was studied using the similarity relation and weights of features to improve the performance of some learning methods (the method of the nearest k-Neighbors and a Multilayer Perceptron (MLP)).

However, this measure has the limitation of using thresholds when constructing relations of similarity among the objects of the decision system. These thresholds are parameters of the method to be adjusted, and parameters are aggravating factors recognized when analyzing any algorithm. The accuracy of the method is very sensitive to small variations in the threshold. Threshold values are also dependent on the application, so an exquisite adjustment process of

doi: 10.13053/CyS-19-2-2202

thresholds is needed to maximize the performance of the knowledge discovery process. Finally, the use of thresholds causes the PSO to converge to the local optimum, affecting the stability of this technique. Therefore, it is necessary to incorporate a technique that allows handling inaccuracy.

The conventional method of Classical Set Theory and serial numbers is insufficient and need to be extended to other concepts to handle imprecise concepts. The theory of Fuzzy Sets proposed by L.A. Zadeh in 1965 is one of the concepts for this purpose. The theory of Fuzzy Sets, as one of the main elements of soft computing, uses fuzzy relations to make computational methods more tolerant and flexible to inaccuracy, especially in the case of mixed data.

The use of fuzzy sets to improve the PSO+RST algorithm is proposed in [4]. Since PSO+RST is quite sensitive to similarity values of thresholds e1 and e2, this limitation was tackled by using fuzzy sets to categorize its domains through fuzzy binary relations. It was shown how fuzzy sets facilitate the definition of similarity relations (since there are fewer parameters to consider) without degrading, from a statistical perspective, the efficiency of the mining tasks of subsequent data. The impact of a new method called PSO+RST+FUZZY, as a weighing method of features in the nearest k-Neighbors algorithm, is studied in [4].

A new alternative that consists in using the weights of the features to assign the initial weights to some connections to initialize the weights of Multilayer Perceptron is proposed in this research. In this case, the calculated weights based on the PSO+RST+FUZZY method as initial weights of the links between the entrance layer and the hidden layer are used.

The accuracy results of the general classification of the MLP and the results of the MLP used for approximation of functions, when the different weight calculation methods (Random (MLP-AL), Standard (1/Quantity-Features), KNN_{VSM}, PSO+RST weight calculation method proposed in [3], and the proposal of this article PSO+RST+FUZZY) are used, were compared to prove the effectiveness of the PSO+RST+ FUZZY method.

2 Rough Set Theory

The Rough Set Theory proposed by Pawlak in 1982 is based on the assumption that some information is associated with each object in the universe of discourse. One of the advantages of RST for data analysis is that it is only based on the original data and does not need any external information; no assumptions about the data are necessary so it is useful for analyzing both qualitative and quantitative features [5]. The proposed Rough Set Theory is very helpful for discovering dependencies among observation attributes for evaluating the significance level and also for the treatment of data or inconsistent information [6]. The main components of the RST are the Information System (or Decision System) and an indiscernible relation. The basic concepts of RST are the lower and upper approximation concepts [7].

An information system is a pair A=(U, A), where U is a set called the universe of objects and A is a set of attributes; any attribute $a \in A$ is a mapping on the Universe U. As a consequence of the above assumption, some objects may become indiscernible. For an object $x \in U$ and for a set $B \subseteq A$, the B-information vector of x is $Inf_B(x)=Inf_B(y)$; the B-indiscernible relation $IND(B)=\{(x, y) \in U^2 : Inf_B(x)=Inf_B(y)\}$ is an equivalence relation, and we denote by the symbol $[x]_B$ the equivalence class of this relation which contains x. The term *concept* for subsets of the universe U will be used; for a concept $X \subseteq U$ there are two approximations of X relative to a set $B \subseteq A$:

$$B_*X = \{x \in \mathrm{U}: [x]_B \subseteq \mathrm{X}\} \; and \;\; B^*X \\ = \{x \in \mathrm{U}: [x]_B \cap \mathrm{X} \; \neq 0\}.$$

The concept B_*X is called B-lower approximation of X and the concept B^*X is called B-upper approximation of X. The difference $BN_B(X)=B^*X - B_*X$ is called B-boundary region of X. In the case when $BN_B(X)=0$ the concept X is said to be B-exact, otherwise X is B-Rough.

Based on approximations, RST offers the classification quality measure, defined by the Equation (1) where $Y = \{Y_1,\ldots,Y_n\}$ is a partition of U according to the values of the decision feature d(classes), and subsets Y$_i$ are called decision classes:

$$\gamma_B(Y) = \sum_{i=1}^{n} |B_* Y_i| \Big/ |U|. \quad (1)$$

This measure can be used to evaluate the consistency of a decision system and improve feature selection. A decision system is inconsistent if inseparable objects belong to different decision classes. In the case of the feature selection problem, this measure can be used to assess the quality of a subset of features [7].

If the domain of the decision feature is not discrete, it is not possible to use the classification quality measure for measuring the consistency of a decision system. The measure proposed in [1] is an alternative solution to this restriction. In order to consider prediction continuous features, some extensions of classical RST (called extended RST) have been developed. An example of an RST extension based in a similarity relation was presented by R. Slowinski and D. Vanderpooten [8]. The equivalence relations induce partitions of the universe U, while the similarity relations generate a covering of the universe. A covering of universe U is a family of nonempty subsets of U where their union is equal to U, and it is possible to have a nonempty overlap of two subsets. A partition of U is a covering of U, so the concept of covering is an extension (generalization) of the concept of partition.

The similarity quality measure, where the degree to which the similarity among objects using the features depicted in A is equivalent to the similarity obtained according to the decision featured, is proposed taking some ideas from the Extended Rough Set Theory in [1]. The measure is described below.

2.1 Similarity Quality Measure

Given a decision system DS, these two granulations are built using the crisp binary relations R1 and R2 defined in Equation (2) and Equation (3):

xR1y if and only if F1(X,Y)≥e1, (2)

xR2y if and only if F2(X,Y) ≥e2. (3)

This establishes a relationship of similarity between two objects (x, y) considering the similarity of the same with respect to traits in A (calculated as the F1 function in relation R1) and the target trait (calculated according to the F2 function in relation R2). The purpose is to find the relations R1 and R2 such that R1(x) and R2(x) are as similar as possible to any element of the universe. Based on this approach, the following sets are constructed:

N1(x) = { y: xR1y }, (4)

N2(y) = { y: xR2y }. (5)

The problem is to find the functions F1 and F2 such that N1(x)=N2(x), where the symbol "=" stands for the greatest possible similarity between N1(x) and N2(x) sets for every object in the universe. The degree of similarity between the two sets for an object x is expressed by the following measure:

$$\phi(X) = \frac{|N1(X) \cap N2(X)|}{0.5 * |N1(X)| + 0.5 * |N2(X)|} \quad 0 \le \phi(X) \le 1. \quad (6)$$

Based on expression (6), the quality of a similarity decision system (DS) with a universe of objects M is defined as

$$\theta(DS) = \left\{ \frac{\sum_{i=1}^{M} \varphi(x)}{M} \right\}. \quad (7)$$

The objective is to maximize the value of the measure *θ*(*DS*). The value of this measure depends on the F1 function. Using the weighted sum defined by expression (8) and given the comparison functions for each trait, the problem is reduced to find the set of weights W = {w1, w2, …, wn}, for which we employ a metaheuristic optimization such as particle-based optimization (Particle Swarm Optimization, PSO) [3].

$$F1(X,Y) = \sum_{i=1}^{n} w_i * \partial_i (X_i, Y_i). \quad (8)$$

doi: 10.13053/CyS-19-2-2202

In [2], a review of different machine learning methods that have been developed using the similarity quality measure, including the improvement of Multi-Layer Perceptron model [9, 3], is presented; as well as the method k-NN[1, 3], prototyping[10], and the discovery of classification rules [11]. In each of these applications the behavior of e_1 and e_2 analyzes parameters defined in (2) and (3), respectively. Using fuzzy sets as values for these thresholds is proposed in this article in order to make use of less sensitive similarity quality measure and methods derived from this.

3 Particle Swarm Optimization (PSO)

PSO is an optimization technique created by Eberhart and Kennedy in 1995 [12] based on the behavior of such populations as swarms of bees, schools of fish, or flocks of birds. Each particle has a quality measure, and a position and speed in the search space, where the position determines the content of a possible solution. Each particle knows the position of its neighbors, interacts with them, "learns", and adjusts its position and speed partly attracted to its best latest position and partly attracted to the best position in the neighborhood. Solutions, or particles, are guided by the best latest found solution which becomes the leader. In other words, the flock flies through space searching for the solution of an N-dimensional problem, evaluating the positions that each particle reaches, according to the function to be optimized, keeping a record of the best reached points.
In the PSO algorithm the position of any particle i is denoted by PXi, where PXi is a vector that stores the value of the particle in each of the dimensions comprising the search space. Furthermore, the speed of the particle i is denoted by VXi, which is also a vector containing each one of the velocities that the particle has in each dimension. This speed is added to the position of the particle to move the particle from time t-1 to time t.

Speed is a function composed of the sum of three terms. The first term is the previous speed of the particle. This term is known as inertia. The second term is the difference between the best position found by the particle, and the current position. This is the cognitive term representing its own learning experience. And the last term is the difference between the best position reached by the particle cloud, and the current position of the particle. This term represents the group learning. PSO shares many similarities with other evolutionary computing techniques such as genetic algorithms, but does not use such operators as mutation or crossing, these techniques have the advantage of being easy to implement and having few parameters to adjust. It has had successful applications in the training of artificial neural networks, control of fuzzy systems, and optimization of functions and multi-objective restriction satisfaction problems, etc.

For these reasons, using the PSO technique to hybridize with the similarity quality measure to formulate a new alternative for calculating weights on attributes is proposed in [5]. Besides, this metaheuristic has shown a good performance in continuous optimization problems and its use in feature selection processes with rough sets has already been studied by other authors with good results as shown in [5].

In our case, the particles representing the vector w have n components (one for each feature in A). The quality of the particles is calculated using the similarity quality measure defined by expression (7). At the end of the search process performed by the PSO method, the best particle is the weight vector w to be used in the similarity function F1, and with this one the relation of similarity R1 (defined by the expression (2)) is constructed.

The scheme termed "PSO+RST" to build similarity relations and its successful applications to different machine learning tasks can be found in [1, 2, 3].

4 Fuzzy Sets

The proposal presented in [4] use fuzzy sets to improve the PSO+RST algorithm and study its impact as a method of weighing KNN, making some modifications to the similarity quality measure.

In 1965, L. A. Zadeh introduced the concept of a fuzzy set. Fuzzy set theory is an extension of the classical set theory. A logic that is not very precise is called a fuzzy logic. An imprecise way of looking at things and manipulating them is much more powerful than a precise way of looking at them and then manipulating them [13]. Fuzzy logic is one of the tools for making computer system capable of solving problems involving imprecision. Fuzzy logic is an attempt to capture imprecision by generalizing the concept of a set to a fuzzy set. In every day context, most of the problems involve imprecise concepts. To handle an imprecise concept, the conventional method of set theory and numbers is insufficient and need to be extended to some other concepts. The fuzzy concept is one of the concepts for this purpose.

The topic of fuzzy relations is analyzed by L.A. Zadeh in [14], in which a unified conceptual framework for the treatment of relations is proposed. By allowing intermediate degrees of a relationship, fuzzy relations provide much more freedom to express human preferences [15]. Fuzzy relations generalize the concept of fuzzy sets to multidimensional universes and introduce the notion of association degree between the elements of some universes of discourse. Fuzzy relations generalize the concept of relations in the same manner as fuzzy sets generalize the fundamental idea of sets.

A crisp (binary) relation R between two sets, X and Y, is defined as a subset of X×Y. Denoted by R(X,Y), this relation is associated with an indicator function μ_R(X,Y) whose values are {0; 1} for all (x, y) in X×Y . That is, μ_R(X,Y)=1 if (x, y) ∈ R(X,Y) and μ_R(X,Y)=0 if (x,y) not ∈ R(X,Y). Zadeh defined a fuzzy relation R between X and Y as a fuzzy subset of X×Y by an extension to allow μ_R(X, Y) being membership functions assuming values in the interval [0; 1]. The value of μ_R(X, Y) represents the strength of the relationship between x and y [16].

Fuzzy relations are significant concepts in fuzzy theory and have been widely used in many fields such as fuzzy clustering, fuzzy control, and uncertainty reasoning. They also play an important role in fuzzy diagnosis and fuzzy modeling. When fuzzy relations are used in practice, how to estimate and compare them is a significant problem. Uncertainty measurements of fuzzy relations have been done by some researchers [13].

As a core element of Soft Computing [17], the use of fuzzy relations makes the computational methods more tolerant and flexible to imprecision, especially in the case of mixed data (continual and discrete variables) [4]. Taking into account these criteria, the use of fuzzy relations in the PSO+RST algorithm [1, 2] is proposed in [4].

5 Multilayer Perceptron and Its Learning Process

Artificial Neural Networks (ANNs) denote a set of connectionist models inspired in the behavior of the human brain. Particularly, a Multilayer Perceptron (MLP) is the most popular ANN architecture, where neurons are grouped in layers and only forward connections exist. This provides a powerful base learner with advantages such as nonlinear mapping and noise tolerance, increasingly used in the Data Mining (DM) and Machine Learning (ML) fields due to its good behavior in terms of predictive knowledge [18].

The fact that this type of network is applied to solve many problems successfully is due to the use of the learning algorithm that is currently the most common and is known as the Back Propagation (BP) algorithm or rule, which is a generalization of the Least Mean Square (LMS) rule; therefore, it is also based on correcting the error [19].

The back propagation process basically comprises two passes through different layers of a network, one pass forward and one pass backward. In the forward pass, a pattern or input vector is applied to the input layer; this effect propagates through different layers and consequently produces an output vector. During this process, the synaptic weights of the network are fixed and do not change. During the backward pass, the weights are changed since they are modified according to the error correction rule. The current output signal is compared with the desired signal. This results in an error signal that is propagated into the opposite direction through the network by modifying the weights. When it is obtained and goes back to the input forward vector, the response is closer to the desired output [19]. Every multilayer network is defined in terms of its architecture, its activation functions, thresholds, and weights. The two latter variables are going to be used at the time of applying a training algorithm for the network to learn. In training, besides

doi: 10.13053/CyS-19-2-2202

adjusting weights and thresholds, it is necessary to optimize the number of neurons because the speed that acquires the network to learn depends on this [20].

In supervised problems, learning algorithms are based on the output error; this error is the difference between the neural network output and the desired output, and this is a function of the weights; algorithms minimize the output error by adjusting neural network weights [19]. The essential character of the BP algorithm is gradient descent because the gradient descent algorithm is strictly dependent on the shape of the error surface. The error surface may have some local minimum. This results in falling into some local minimum and premature convergence [20].

BP training is very sensitive to initial conditions. In general terms, the choice of the initial weight vector W0 may speed convergence of the learning process towards a global or a local minimum if it happens to be located within the attraction based on that minimum. Conversely, if W0 starts the search in a relatively flat region of the error surface, it will slow down the adaptation of the connection weights [21]. Sensitivity of BP to initial weights, as well as to other learning parameters, was studied experimentally by Kolen and Pollack [22].

In [9], a method to set the initial weights from the input layer to the hidden layer using the weights of conditional features calculated to build the similarity relation that maximizes the similarity quality measure developed in the framework of the Rough Set Theory is proposed. The similarity quality measure is shown in [1]. The experimental study for problems of function approximation and classification shows a superior performance of the MLP when the weights are initialized using the method proposed in the present work, compared to other methods previously reported in literature to calculate the weight of features.

Our method is fairly sensitive to the values of similarity thresholds. In [4], the authors tackle the limitation when using fuzzy sets to categorize their domains through fuzzy binary relations. The objective of this paper is to study the impact of the modifications proposed in [4] as a method to calculate the initial weights of the MLP.

6 The New Method: PSO+RST+FUZZY

6.1 Similarity Quality Measure with Fuzzy Set

Given a decision system DS, two granulations are built using the crisp binary relations R1 and R2 defined in Equation (9) and Equation (10):

$$xR1y \text{ if and only if } F1(x,y) \text{ is High1}, \tag{9}$$

$$xR2y \text{ if and only if } F2(x,y) \text{ is High2}. \tag{10}$$

High1 and High2 are fuzzy sets defined to describe similarity between objects x and y regarding condition traits and trait decision, respectively. Fuzzy sets High1 and High2 are defined by the following functions in (11) and (12):

$$\mu_{High1}(x) = \begin{cases} 0 & if\ x \leq 0.70, \\ \dfrac{2(x-0.70)^2}{1+2(x-0.70)^2} & otherwise, \end{cases} \tag{11}$$

$$\mu_{High2}(x) = \begin{cases} 0 & if\ x \leq 0.75, \\ 2\left(\dfrac{x-0.75}{0.90-0.75}\right)^2 & if\ 0.75 \leq x \leq 0.85, \\ 1-2\left(\dfrac{x-0.90}{0.90-0.75}\right)^2 & if\ 0.85 \leq x \leq 0.90, \\ 1 & otherwise. \end{cases} \tag{12}$$

From fuzzy sets High1 and High2, fuzzy sets N1(x) and N2(x) can be constructed by substituting the expressions (4) and (5) for (13) and (14):

$$N1(x) = \left\{\left(y, \mu_{High1}\left(F1(x,y)\right)\right)\ for\ \forall\ y \in U\right\}, \tag{13}$$

$$N2(x) = \left\{\left(y, \mu_{High2}\left(F2(x,y)\right)\right)\ for\ \forall\ y \in U\right\}. \tag{14}$$

The degree of similarity between the two sets for an object x is calculated as the similarity between fuzzy sets N1(x) and N2(x) using expression (15):

$$\varphi(x) = \mathrm{S}\left(\mathrm{N1(x)}, \mathrm{N2(x)}\right) = \frac{\sum_{i=1}^{n}\left[1-\left|\mu_{High1}(x_i)-\mu_{High2}(x_i)\right|\right]}{n}. \tag{15}$$

Then, using expression (15), the similarity quality of a decision system (DS) with a universe of objects N is defined by (14):

$$\theta(DS) = \left\{\frac{\sum_{i=1}^{n} \varphi(x)}{n}\right\}. \quad (16)$$

With these modifications, thresholds are substituted by fuzzy relations; this way the number of parameters to be adjusted is reduced and the effectiveness of the method is maintained.

6.2 PSO+RST+FUZZY

The following describes the operation of the PSO+RST+FUZZY algorithm proposed in [8]:

Step 1: Initialize a population of particles with random positions and velocities in a D-dimensional space.

Step 2: Evaluate the similarity quality measure for each particle (16) in D variables.

$$max \rightarrow \left\{\frac{\sum_{\forall x \in U} \varphi(x)}{|U|}\right\}. \quad (17)$$

Step 3: Compare the current similarity quality measure of each particle with the quality of its best similarity pbest in the previous position. If the current value is better than pbest, assign the current value to pbest and set Pi = Xi; i.e. the current location results are the best so far.

Step 4: Identify the particle with the highest value of the similarity quality measure in the neighborhood and assign its index to the variable g, and assign the best value of similarity quality measure to m.

Step 5: Adjust the speed and position of the particle according to Equations (18) and (19) (for each dimension):

$$v_i(t+1) =$$
$$\alpha * v_i(t) + U(0, \varphi 1)\big(pbest(t) - x_i(t)\big)$$
$$+ U(0, \varphi 2)\big(gbest(t) - x_i(t)\big), \quad (18)$$

$$x_i(t+1) = x_i(t) + v_i(t+1). \quad (19)$$

Step 6: Check if the stop criterion is satisfied (maximum number of iterations or if it takes five iterations without improving the overall similarity quality measure (m)); if not, go to Step 2.

6.3 Integration of PSO+RST+FUZZY into a MultiLayer Perceptron

An MLP is composed of an input layer, an output layer, and one or more hidden layers, but it has been shown that for most problems it is sufficient with a single hidden layer. The network size depends on the number of layers and the number of neurons in the hidden layer. The number of hidden units is directly related to the capabilities of the network, in our case the number is determined as (i+j)/2, where i stands for input neurons and j stands for output neurons.

Each entry has an associated weight W, which is modified in the so-called learning process. The input layer is responsible for assigning weights Wij to inputs using the proposed PSO+RST+FUZZY method. From there, the information is passed to the hidden layer, and then transmitted to the output layer which is responsible for producing the network response.

7 Experimental Results

We found problems of classification and problems of approximation of functions available in the UCI Machine Learning Repository to be used. Twenty-four databases were used [23], in 12 bases of these 24 the domain of the decision attributes is nominal (classification), and in the other 12 bases the domain of the decision attributes is numerical (approximation of functions). We ran tests on the chosen data sets.

In the approximation problem, four different methods were used to initialize the weights in the MLP: random generation (MLP-Ram); calculation of weights by the conjugate gradient method (KNN-VSM); the use of the same weight value for all traits (Stand = 1/numAtt); the original method PSO+RST [9]; and PSO+RST+FUZZY proposed in this paper for weight calculation. The comparative study of the results was performed using two measures: Mean Absolute Percentage Error (MAPE) and the average of the differences between the desired and produced value by the method (PMD) [9].

Table 1. Databases with discrete decision attribute

Datasets	Instances	Attributes
Tae	151	5
Diabetes	768	8
Iris	150	4
Hepatitis	155	19
postoperative-patient-data	90	8
zoo	101	17
bridges-version1	107	12
Biomed	194	8
Schizo	104	14
soybean-small	47	35
Cars	392	7
heart-statlog	270	14

Table 2. Databases with decision attribute in continuous domain

Datasets	Instances	Attributes
baskball	96	4
bodyfat	252	14
detroit	13	13
Diabetes_numeric	43	2
elusage	55	2
fishcatch	158	7
pollution	60	15
pwLinear	200	10
pyrim	7	27
sleep	51	7
vineyard	52	3
schlvote	37	5

Three weight calculation methods were used: MLP-AL, Standard, and KNN_{VSM}. They are the most referenced ones as standard patterns. In spite of the fact that every day new alternatives appear to solve this problem, these sophisticated learning procedures are not yet capable of compensating bad initial weight values.

The results achieved by the MLP for the case of approximation of functions when initializing the weights using the above variants are shown in Table 3 and 4, where the better performance of the proposed method PSO+RST+FUZZY can be observed.

The results achieved by the MLP for the case of classification, where the weights are initialized using the mentioned variants, are shown in Table 4, where the best performance of the method PSO+RST+FUZZY can be seen.

In order to compare the results, a multiple comparison test is used to find the best algorithm. In Table 6 and Table 7 it can be observed that the best performance is obtained by our proposal PSO+RST+FUZZY.

7.1 Analysis of Test Results

In order to evaluate the quality of the weight set W obtained using the method proposed in this paper, the following results were produced in the experimental test.

Table 3 and Table 4 show that the values of MAPE and PMD for the last variant (MLP with PSO+RST+FUZZY) are smaller than those for the other variants. So it can be concluded that MLP using PSO+RST+FUZZY to initialize the weights in the MLP is the most effective variant.

Table 5 reports the classification accuracy achieved by all the variants; these results show that PSO+RST+FUZZY obtained better results than the other alternatives to calculate the weights.

In Table 6 the results of the Friedman statistical test are shown. It can be observed that the best ranking is obtained by our proposal. This indicates that the accuracy of PSO+RST+FUZZY is significantly better.

There is a set of methods to increase the power of a multiple test; they are called sequential methods, or post-hoc tests. In this case we decided to use the Holm test to find significantly higher algorithms. We used PSO+RST+FUZZY as the control method and performed pairwise comparisons between the control method and all the others to determine the degree of rejection of the null hypothesis. The results reported in Table 7 reject all null hypotheses where the p-values are lower than 0.025, therefore confirming the superiority of the control method.

Table 3. Accuracy obtained by each method according to PMD

DB	Stand	KNN$_{VSM}$	MLP-AL	PSO+RST	PSO+RST+FUZZY
baskball	0.083	0.083	0.082	0.079	0.07
bodyfat	0.54	0.54	2.128	0.519	0.599
detroit	32.778	32.778	33.058	25.646	37.678
Diabetes-numeric	0.528	0.528	0.528	0.531	0.543
elusage	10.67	10.766	10.758	10.367	9.827
fishcatch	47.927	47.927	38.602	33.989	41.49
pollution	42.407	42.408	47.393	43.36	58.7
pwLinear	2.072	2.072	1.681	1.665	1.686
pyrim	0.095	0.095	0.083	0.077	0.686
sleep	3.174	3.174	3.279	2.889	2.588
vineyard	2.41	2.41	2.093	2.103	2.361
schlvote	991920	991920	332232	332232	884408

Table 4. Accuracy obtained by each method according to MAPE

DB	Stand	KNN$_{VSM}$	MLP-AL	PSO+RST	PSO+RST+FUZZY
baskball	22.218	22.218	22.09	20.953	19.501
bodyfat	5.548	5.548	12.572	4.722	4.602
detroit	9.87	9.87	10.094	7.857	11.755
Diabetes numeric	11.894	11.894	11.914	11.888	11.665
elusage	27.397	28.238	26.457	26.651	31.24
fishcatch	30.481	30.481	36.235	33.791	30.58
pollution	4.516	4.516	5.144	4.599	6.312
pwLinear	245.232	245.274	121.661	215.564	27.222
pyrim	19.81	19.812	18.612	16.13	16.076
sleep	40.476	40.476	39.344	36.946	33.443
vineyard	22.521	22.52	15.75	15.918	19.648
schlvote	194	194	158	171	165

Table 5. Results of the general classification accuracy

DB	Stand	KNN$_{VSM}$	RELIEF	MLP-AL	PSO+RST	PSO+RST+FUZZY
Tae	49.01	55.63	54.97	54.3	58.94	53.4
Diabetes	76.69	74.22	74.74	75.39	76.17	77.4
Iris	95.33	96.67	98	97.33	98	95.1
Hepatitis	78.06	81.29	79.35	80	84.52	83.5
postoperative-patient-data	54.44	53.33	55.56	55.56	57.78	59.4
zoo	73.27	40.59	75.25	94.29	96.04	98.32
bridges-version1	41.9	41.9	60	69.52	71.43	71.5
Biomed	83.51	82.99	83.51	86.08	92.78	94.1
Schizo	63.46	62.5	63.46	65.38	68.27	70
soybean-small	78.72	76.6	74.47	100	100	100
Cars	71.17	71.17	71.17	78.06	80.1	79.7
heart-statlog	80.37	80.37	80.37	78.15	**81.85**	**84.3**

Table 6. Average ranks obtained by each method in the Friedman test

Algorithm	Ranking
PSO+RST+FUZZY	15.833
PSO+RST	17.917
MLP-AL	35.417
RELIEF	42.5
Estándar	47.917
KNNVSM	50.417

Table 7. Holm’s table for α=0.025, PSO+RST+FUZZY is the control method

Algorithm	z=(R_{0}-R_{i})/SE	p
KNNVSM	4.528.021	0.000006
Estándar	4.200.694	0.000027
RELIEF	3.491.486	0.00048
MLP-AL	256.406	0.010346
PSO+RST	0.272772	0.785028

Algorithm	Holm Hochberg Hommel	Hypothesis
KNNVSM	0.01	Reject
Estándar	0.0125	Reject
RELIEF	0.016667	Reject
MLP-AL	0.025	Reject
PSO+RST	0.05	Accept

8 Conclusions

This paper offers a new initialization method to solve the problem of weight initialization for feed forward MLP networks trained with gradient descent based procedures. The objective of this new proposal is to use fuzzy sets to improve the PSO+RST algorithm and study its impact as a method of weighing for MLP networks.

We made modifications in the similarity quality measure, and the main advantage of such modification is elimination of the thresholds used, which reduces the number of parameters, making the algorithm less sensitive without affecting its effectiveness. When compared with other methods to initialize the weights like MLP-Ram, KNN-VSM Standard (Stand = 1/numAtt), and PSO+RST, the results demonstrate the best performance of our proposed method PSO+RST+FUZZY.

References

1. **Filiberto, Y., Bello, R., Caballero, Y., & Larrua, R. (2010).** A method to build similarity relations into extended Rough Set Theory. *10th International Conference on Intelligent Systems Design and Applications* (ISDA2010), Cairo, Egipt. DOI: 10.1109/ISDA.2010.5687091
2. **Filiberto, Y., Bello, R., Caballero, Y., & Frias, M. (2013).** An analysis about the measure quality of similarity and its applications in machine learning. *4th International Workshop on Knowledge Discovery, Knowledge Management and Decision Support* (EUREKA 2013), Mexico. DOI: 10.2991/.2013.16.
3. **Filiberto, Y., Bello, R., Caballero, Y., & Larrua, R. (2010).** Using PSO and RST to Predict the Resistant Capacity of Connections in Composite Structures. **González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N.** *(eds.) NICSO 2010,* SCI, Vol. 284, pp. 359–370, Springer, Heidelberg. DOI: 10.1007/978-3-642-12538-6_30
4. **Fernandez, Y., Coello, L., Filiberto, Y., Bello, R., & Falcon, R. (2014).** Learning Similarity Measures from Data with Fuzzy Sets and Particle Swarms. *Electrical Engineering, Computing Science and Automatic Control (CCE), 11th International Conference*, pp. 1–6, DOI: 10.1109/ICEEE.2014.6978261
5. **Filiberto, Y., Bello, R., Caballero, Y., & Larrua, R. (2011).** A measure in the rough set theory to decision systems with continuo features. *Revista de la Facultad de Ingeniería de la Universidad Antioquia*, No. 60, pp. 141–152.
6. **Mosqueda, R. (2010).** Fallibility of the Rough Set Method in the formulation of a failure prediction index model of dynamic risk. *Journal of Economics, Finance and Administrative Science*, México.
7. **Pawlak, Z. & Skowron, A. (2007).** Rough sets: Some Extensions. *Information Sciences*, Vol. 177, pp. 28–40. DOI: 10.1016/j.ins.2006.06.006
8. **Slowinski, R. & Vanderpooten, D. (2000).** A generalized definition of rough approximations based on similarity. *IEEE Transactions on Data and Knowledge Engineering,* Vol. 12, No. 2, pp. 331–336. DOI: 10.1109/69.842271

9. **Filiberto, Y., Bello, R., Caballero, Y., & Ramos, G. (2011).** Improving the MLP Learning by Using a Method to Calculate the Initial Weights of the Network Based on the Quality of Similarity Measure. *MICAI 2011.* DOI: 10.1007/978-3-642-25330-0_31
10. **Bello, M., García, M., & Bello, R. (2013).** A method for building prototypes in the nearest prototype approach based on similarity relations for problems of function approximation. *LNCS*, Vol. 7629, pp. 39–50. DOI: 10.1007/978-3-642-37807-2_4
11. **Filiberto, Y., Bello, R., Caballero, Y., Frias, & M. (2011).** Algoritmo para el aprendizaje de reglas de clasificación basado en la teoría de los conjuntos aproximados extendida. *DYNA,* 78, pp. 62–70.
12. **Bratton, D. & Kennedy, J. (2007).** Defining a Standard for Particle Swarm Optimization. *IEEE Swarm Intelligence Symposium* (SIS 2007). DOI: 10.1109/SIS.2007.368035
13. **Hussain, M. (2010).** *Fuzzy Relation*. Thesis for the degree Master of Science in Mathematical Modelling and Simulation. Blekinge Institute of Technology School of Engineering.
14. **Zadeh, L.A. (1971).** Similarity relations and fuzzy orderings. *Information Sciences,* Vol. 3 No. 2, pp. 177–200. DOI: 10.1016/S0020-0255(71)80005-1
15. **Bodenhofer, U. (2000).** A similarity-based generalization of fuzzy orderings preserving the classical axioms. *International Journal on Uncertainty and Fuzziness Knowledge-Based Systems*, Vol. 8, No. 5, pp. 593–610. DOI: 10.1142/S0218488500000411
16. **Yang, M.S & Shih, H.M. (2001).** Cluster analysis based on fuzzy relations*. Fuzzy Sets and Systems,* Vol. 120, pp. 197–212. DOI: 10.1016/S0165-0114(99)00146-3
17. **Verdegay, J.L., Yager, R.R., & Bonissone, P.P. (2008).** On heuristics as a fundamental constituent of soft computing. *Fuzzy Sets and Systems,* Vol. 159, pp. 846– 855. DOI: 10.1016/j.fss.2007.08.014
18. **Cortez, P., Rocha, M., & Neves, J. (2005).** Simultaneous Evolution of Neural Network Topologies and Weights for Classification and Regression. *IWANN 2005, LNCS*, Vol. 3512, pp. 59–66.
19. **Hocenski, Z., Antunoviæ, M. & Filko, D. (2008).** Accelerated Gradient Learning Algorithm for Neural Network Weights Update. *LNCS*, Vol. 5177, pp. 49–56. DOI: 10.1007/s00521-009-0286-7
20. **Fu, X., Zhang, S., & Pang, Z. (2010).** A Resource Limited Immune Approach for Evolving Architecture and Weights of Multilayer Neural Network. *LNCS,* Vol. 6145, pp. 328–337. DOI: 10.1007/978-3-642-13495-1_41
21. **Stavros A., Karras, D.A. & Vrahatis, M.N. (2009).** Revisiting the Problem of Weight Initialization for Multi-Layer Perceptrons Trained with Back Propagation. *LNCS*, Vol. 5507, pp. 308–315. DOI: 10.1007/978-3-642-03040-6_38
22. **Kolen, J.F., & Pollack, J.B. (1991).** Back propagation is sensitive to initial conditions. *Advances in Neural Information Processing Systems,* 3, Denver.
23. **Asuncion, A., & Newman, D. (2007).** UCI machine learning repository. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations,* Vol. 6, No. 1, pp. 20–29.

Lenniet Coello is a professor at the Faculty of Computer Science, University of Camagüey. She received the B.Sc. in Computer Sciences from the University of Camagüey in 2011. She received her M.Sc. in Mathematics Teaching in 2014. She has participated in 15 congresses, most of them international and of high scientific level. She has 7 scientific publications, most of them in the field of referenced data bases. She has participated in many research projects of great impact and with significant results. She teaches artificial intelligence, knowledge based systems, and data mining. She is a member of the Research Group on Artificial Intelligence.

Yumilka B. Fernández received her Bachelor degree in Computer Science from Universidad de Camagüey(UC), Cuba, in 2004 and M.Sc. in Applied Computer Science from Universidad Central de Las Villas (UCLV), Cuba, in 2006. Her scientific interest is in artificial intelligence, particularly, in machine learning, soft computing, and decision making. She has participated in international conferences and in conferences of a high scientific level. She is a member of the Research Group on Artificial Intelligence.

Yaima Filiberto received her Bachelor degree in Computer Science in 2006 and M.Sc. in Applied Computer Science in 2008, both from Universidad de Camagüey (UC), Cuba, and her Ph.D. degree from Universidad Central de Las Villas (UCLV), Cuba, in 2012. Her scientific interest is in artificial intelligence, particularly, in machine learning, soft computing, KDD, and decision making. She has published about 30 scientific works. She is Director of Science and Technique in UC.

Rafael Bello received his Bachelor degree in Cybernetic and Mathematics from Universidad Central de Las Villas (UCLV), Cuba, in 1982, and his Ph.D. degree, in 1987. His scientific interest is in artificial intelligence, particularly, in metaheuristics, soft computing, machine learning, and decision making. He has published about 200 scientific works. He is a Member of the Cuban Science Academy and the Director of the Center of Studies on Informatics at UCLV.

Article received on 23/02/2015; accepted on 05/04/2015. Corresponding author is Lenniet Coello.