Applied Soft Computing Journal 76 (2019) 607–614

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

A novel method for structure selection of the Recurrent Random Neural Network using multiobjective optimisation

Erivelton G. Nepomuceno

Control and Modelling Group (GCOM), Department of Electrical Engineering, Federal University of São João del-Rei, Praça Frei Orlando, 170 - Centro São João del-Rei, MG, 36307-352, Brazil

HIGHLIGHTS

- A novel scheme for multiobjective optimisation structure selection of the RRNN.
- A systematic way to specify the number of neurons in a RRNN.
- A stochastic nondominated algorithm to exclude dominated solutions of the Pareto-set.

ARTICLE INFO

Article history:
Received 19 January 2018
Received in revised form 22 July 2018
Accepted 29 October 2018
Available online 6 January 2019

Keywords:
Recurrent Random Neural Network
Structure optimisation
Stochastic nondominated algorithm
Multiobjective optimisation
System identification and modelling
Classification problem

ABSTRACT

The Random Neural Network (RNN) has extensively investigated over the past few decades; this research has resulted in a considerable number of theoretical and application papers. Although, great effort has been done to develop a systematic procedure to train the recurrent fashion of the RNN, the choice of the number of neurons remains an open question. To overcome this problem, at least partially, this paper uses multiobjective optimisation (MOP) to select the number of neurons. The MOP framework used the mean square error (MSE) and the number of neurons (N) as the objectives to be minimised. The stochastic nondominated algorithm (SNA) to exclude dominated solutions of the Pareto-set has been also introduced. Instead of using only the best solution, candidates to the Pareto-set are excluded by statistical comparison among mean values of the two objectives in all training runs. The SNA allows a statistically correct exclusion of dominated solutions; the best solution can be picked up using classical decision-making procedures. Numerical and real examples illustrate the potentiality of the proposed method in two areas: classification problems and system identification.

1. Introduction

The Random Neural Network (RNN) is propelled by the spiking behaviour of the physics of biological processes in a real neuron [1–3]. Signals propagate between neurons, and enter or leave the network, both as inhibitory and excitatory spikes of amplitude -1 and $+1$, respectively. The two most common topologies of RNN are multilayer feedforward [4] and recurrent networks (RRNN) [5].

During recent decades, much research has been done on building up the RNN and on applications in various territories of engineering and science. In [6], a few components that clarify the accomplishment of the RNN are outlined: (i) it introduces an effective calculation strategy by means of an analytical equation for its steady-state probability distribution [6]; (ii) it has low complexity for standard learning algorithms [7]; (iii) it has a close connection with biological neuronal network [8]; (iv) and it can serve as a universal approximation for bounded continuous functions [9]. Probably the most vital uses of the RNN concern on optimisation [10], detection of malicious behaviour in mobile networks [11] and supervised learning problems [4]. More recently, a new type of RNN based on Tsallis statistics has been proposed [12]. The reader can also found an interesting survey on applications of the RNN in [13]. Very recent works have been published on RNN. For more information the reader can refer to [14–18].

Although, great effort has been done to develop a systematic procedure to train the recurrent fashion of the RNN, the choice of the number of neurons remains an open question. According to [19] there is no endeavour to simultaneously optimise the weights and the structure of the RNN by means of using a multiobjective technique [20]. This problem has also been pointed in works such as [4,21,22]. This reveals a different research stage in comparison to Multilayer Perceptron, in which there are a considerable number of works that deals with optimisation of topology and number of neurons [23,24]. The number of neurons for the hidden layer has been successfully determined using a stochastic search algorithm for radial basis function neural network in [25]. Innovative methods, such as shark smell optimisation has also

E-mail address: nepomuceno@ufsj.edu.br.

https://doi.org/10.1016/j.asoc.2018.10.055
1568-4946/

been considered to increase the performance of structure selection of neural networks approaches [26]. It is important to point out that the choice of the number of neurons, although showing some advance, has been receiving great attention. There are still works in which the topology is tested by different set of parameters. This is the case of work done by Ullah et al. [27], wherein a range of accuracy within 80% and 92% has been obtained simulating different combinations of training functions and number of neurons. The choice of number of neurons is still a problem to be overcome even in control application, such as described in [28], where the authors use a variable structure neuro fuzzy to reduce the number of neurons, and consequently heavy complex computations. Similar worry has been investigated in [29]. The authors have experimentally observed a reduction of around 40% in the number of neurons, but keeping the accuracy in 99% of the original topology. Also, it has been possible to notice in [29], a significant improvement in the speed of the FPGA which received the investigated neural network. On the other hand, it is possible to find recent works where optimisation tools have been efficiently applied. For instance, the authors in [30] have used genetic algorithm to optimise the results in a deep neural network, where the number of neurons are the input parameters for the optimisation tool. Another example of optimised structure can be seen in [31], where structural parameters, such as the number of neurons are automatically selected to minimise the prediction error criterion according to Akaike's information criterion. Nevertheless, in works such as in [32], the authors have limited to say that the number of neurons is proportional to other works in literature. Certainly, this is a topology that deserves further investigation in this direction. Moghaddari et al. [33] have adopted similar procedure to find the best solution, by means a heuristic approach.

To overcome this problem, at least partially, this paper uses multiobjective optimisation (MOP) to select the number of neurons for a RRNN. The MOP framework used the mean square error (MSE) and the number of neurons (N) as the objectives to be minimised. Numerous works have established valuable procedures to select structures for neural networks using MOP [34–43]. However, considering stochastic representations, such as the RRNN, little attention has been paid for the statistical implications in the stage of removing dominated solutions. In such situations, the comparison of two candidates in the Pareto-set using nominal values can be shown statistically incoherent. The contributions of this paper are twofold. First, it has been introduced the stochastic nondominated algorithm (SNA) to exclude dominated solutions of the Pareto-set. Instead of using only the best solution, candidates to the Pareto-set are excluded by a comparison of means of the two objectives in all training runs using statistic test at the 0.05 level of significance. Second, using the Pareto-set, we present a systematic approach to select the number of neurons for the RRNN. The best solution can be picked up using classical decision-making procedures. Here, three methods to choose the solution among the Pareto-set are applied: (i) minimum N; (ii) minimum MSE; and (iii) minimum norm.

The proposed method is applied in four case studies. The first one is a theoretical and it aims at showing that Pareto-set obtains the precise number of neurons. Two classification problems and a system identification for a piecewise nonlinear system complete the four case studies. In all cases it was possible to present a Pareto-set and offer a systematic way to evaluating the number of neurons for the RRNN. To show the run time and standard deviation properties of the solutions other four classification cases have been presented.

This paper is organised as follows. In Section 2, the RRNN model is described and the gradient descent learning algorithm is briefly presented. The MOP is defined in Section 3, where the SNA is detailed. The steps of the algorithm to select the number of neurons in the RRNN is given in Section 4. Lastly, the results are discussed in Section 5 and final remarks are presented in Section 6.

2. The RRNN model

In this section, the RRNN model based on [1,5,6,44] is described. The RRNN is a recurrent network of N completely associated neurons which exchange *negative* and *positive* impulse signals. Whenever t, the signal potential $k_i(t)$ represents the state of neuron i. If $k_i(t) > 0$, then a neuron is excited and its excitation probability is $q_i(t) = Pr[k_i(t) > 0] \leq 1$.

At the point when a neuron is excited, it can arbitrarily fire as indicated by the exponential distribution with rate r_i bringing about the lessening of its potential by 1. The let go spike may carry on in three distinctive ways. It might achieve another neuron as a negative flag with probability $p^-(i,j)$, or as positive flag with probability $p^+(i,j)$, or it leaves from the network with probability $\ell(i)$.

Consider $k(t) = [k_i(t), \ldots, k_n(t)]$ be the array of signal potentials at time t, and $k = (k_i, \ldots, k_n)$ be a specific value of the vector. $p(k)$ is the steady state probability distribution $p(k) = \lim_{t\to\infty} Pr[k(t) = k]$, which must be in accordance with global balance equations:

$$\begin{aligned} &p(k)\sum_{i=1}^{N}[\Phi(i) + [\phi(i) + r(i)]\mathbf{1}_{\{k_i>0\}}] \\ &= \sum_{i=1}^{N}\{p(k_i^+)r(i)\ell(i) + p(k_i^-)\Phi(i)\mathbf{1}_{\{k_i>0\}} \\ &+ p(k_i^+)\phi(i) + \sum_{j=1}^{N}[p(k_{ij}^{+-})r(i)p + (i,j)\mathbf{1}_{\{k_j>0\}} \\ &+ p(k_{ij}^{++})r(i)p^-(i,j) + p(k_i^+)r(i)p^-(i,j)\mathbf{1}_{\{k_j=0\}}]\} \end{aligned} \tag{1}$$

The Eq. (1) has been solved in [1],

$$p(k) = \prod_{i=1}^{N}[1-q_i]q_i^{k_i} \tag{2}$$

where

$$q_i = \frac{\phi^+(i)}{r(i) + \phi^-(i)} \tag{3}$$

and $\phi^+(i), \phi^-(i)$ for $i = 1, \ldots, N$ fulfil the next system of nonlinear equations:

$$\begin{aligned} \phi^+(i) &= \sum_{j=1}^{N} q_j r(j) p^+(j,i) + \Phi(i), \\ \phi^-(i) &= \sum_{j=1}^{N} q_j r(j) p^+(j,i) + \phi(i) \end{aligned} \tag{4}$$

Let the next rates:

$$w^+(i,j) = r_i p^+(i,j) \tag{5}$$

$$w^-(i,j) = r_i p^-(i,j). \tag{6}$$

Furthermore, from Eqs. (5)–(6) and considering that the total of the three probabilities for all j neurons is 1, the accompanying expression for r_i is determined:

$$r_i = (1-\ell(i))^{-1}\sum_{j=1}^{N}[w^+(i,j) + w^-(i,j)]. \tag{7}$$

Positive and negative signals can also arrive from the outside world according to Poisson processes of rates Φ_i and ϕ_i, respectively. Each neuron accumulates signals as they arrive, and fires if its total signal count at a given instant of time is positive.

2.1. The RRNN learning process

The author of [5] has formulated a gradient descent supervised learning algorithm for the RRNN. In the RRNN, the mth input data $\vec{x}_m$ is described by the vectors $\vec{\Phi} = [\Phi_{m1}, \ldots, \Phi_{mN}]$ and $\vec{\phi} = [\phi_{m1}, \ldots, \phi_{mN}]$. The technique to attribute the input to exogenic rates is detailed in [6].

The output of the mth pattern, $\vec{y}_m$ are described by the stationary excitation probabilities of the neurons $\vec{q}_m = [q_{m1}, \ldots, q_{mN}]$ evaluated from employing input training pattern m to the RRNN. The learning method comprises in updating the RRNN weights $w^+(i,j) \in \mathbb{R}^{N\times N}$ and $w^-(i,j) \in \mathbb{R}^{N\times N}$. On the other hand in a more broad manner, it comprises in updating the matrix $W \in \mathbb{R}^{2N\times N}$, such that

$$W = \begin{bmatrix} w^+(i,j) \\ w^-(i,j) \end{bmatrix} \tag{8}$$

3. Multiobjective optimisation

According to [45], the multiobjective optimisation problem (MOP) may be defined as:

$$\begin{aligned} &\text{minimise} \quad \vec{f}(\vec{\kappa}) = \begin{bmatrix} f_1(\kappa_1, \kappa_2, \ldots, \kappa_n) \\ f_2(\kappa_1, \kappa_2, \ldots, \kappa_n) \\ \vdots \\ f_A(\kappa_1, \kappa_2, \ldots, \kappa_n) \end{bmatrix} \\ &\text{subject to} \quad \vec{\kappa} \in \Omega, \end{aligned} \tag{9}$$

where $\vec{f} : \mathbb{R}^n \to \mathbb{R}^A$ and $\Omega \subset \mathbb{R}^n$ is the constraint set.

Consider the definition presented in [46]

Definition 3.1. Let $\vec{f} : \mathbb{R}^n \to \mathbb{R}^A$ and $\vec{\kappa} \in \Omega$ be given. For the optimisation problem

$$\begin{aligned} &\text{minimise} \quad \vec{f}(\vec{\kappa}) \\ &\text{subject to} \quad \vec{\kappa} \in \Omega \end{aligned}$$

a point $\vec{\kappa}^* \in \Omega$ is called a *Pareto minimiser* or *nondominated solution* if there is no $\vec{\kappa} \in \Omega$ such that $\vec{f}(\vec{\kappa}) \leq \vec{f}(\vec{\kappa}^*)$ and $\vec{f}(\vec{\kappa}) \neq \vec{f}(\vec{\kappa}^*)$. ■

Therefore, if there is no other $\vec{\kappa} \in \Omega$ that diminishes other objectives without a simultaneous increase in at least one other variable, then $\vec{\kappa}^*$ is a *Pareto minimiser*. The set of $\vec{\kappa}^*$ is nominated the *Pareto-set* ($\mathcal{P}$) [45]. Let Υ be defined as objective function space of the *Pareto-set*, such that $\vec{f} : \mathcal{P} \to \Upsilon$. A multiobjective function specifies to each variable $\vec{\kappa}$ a multiobjective vector function value in the objective function space. In general, the MOP are applied to nonconvex problems, which show local solutions. The following definition in such case is useful:

Definition 3.2. A solution $\vec{\kappa}_\ell$ is a *local Pareto minimiser* in a region $\mathcal{N}(\vec{\kappa}_\ell, \delta) \subset \Omega$ if exists $\delta > 0$ and there is no exist any other solution $\vec{\kappa} \in \mathcal{N}(\vec{\kappa}_\ell, \delta)$ such that for $\vec{f}(\vec{\kappa}) \leq \vec{f}(\vec{\kappa}_\ell)$ and $\vec{f}(\vec{\kappa}) \neq \vec{f}(\vec{\kappa}_\ell)$. ■

The main concern of a MOP is to estimate the Pareto-set [47,48]. The determination of the *Pareto-set* is usually undertaken by means of finding a subset of nondominated solutions of all *local Pareto-set*. The choice of a unique solution of $\mathcal{P}$ is made by a *decision maker* (DM) [48]. This technique is denominated *decision-making task*.

As mentioned, a MOP solution uses a stochastic optimisation process. In such situation, each point of Pareto-set is a vector of solutions, as the optimisation procedure runs several times. Let the following definition for a biobjective case of N and MSE, which describes a procedure to exclude dominated solutions for such case.

Table 1
Exclusion of dominated solutions. Applying the concept of stochastic nondominated described in Definition 3.3, only solutions A and D are stochastic nondominated.

Solution	Neuron	Mean	Runs				
			1	2	3	4	5
A	2	0.100	0.10	0.09	0.08	0.11	0.12
B	3	0.094	0.09	0.08	0.07	0.11	0.12
C	4	0.130	0.11	0.15	0.14	0.12	0.13
D	5	0.014	0.01	0.02	0.01	0.01	0.02

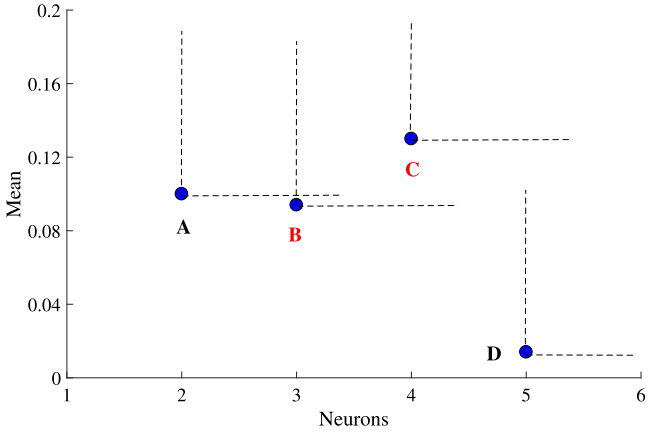

Fig. 1. Solutions of hypothetical biobjective problem. The objectives are number of neurons and mean of MSE. In general, only solution C would be excluded. Following Definition 3.3 stochastic nondominated solution, the solution B is also excluded. Thus, the Pareto-set contains solutions A and D.

Definition 3.3 (*Stochastic Nondominated*)**.** Consider a pair of solutions in objective function space given by $(f_{1i}, \bar{f}_{2i})$, where f_{1i} is N and $\bar{f}_{2i}$ is mean value of MSE for all runs of gradient descent simulation. If one of the following conditions is satisfied, then a *ith* solution is excluded.

(a) It exists at least one pair $(f_{1j}, \bar{f}_{2j})$, with $j \neq i$ that $f_{1j} < f_{1i}$, $\bar{f}_{2j} < \bar{f}_{2i}$ and $\bar{f}_{2j} \neq \bar{f}_{2i}$ at the 0.05 level of significance.
(b) It exists at least one f_{1j} with $j \neq i$ that $f_{1j} < f_{1i}$ and $\bar{f}_{2j} = \bar{f}_{2i}$ at the 0.05 level of significance.

Table 1 shows an example of the application of this definition. In this hypothetical Pareto-set, for each number of neurons $N = [2; 3; 4; 5]$ there are 5 runs of MSE. The mean of MSE was presented in the third column. In the definition of dominance described in [36], only the solution C would be excluded. Nevertheless, let us suppose that the mean of solutions A and B are equal. Using a p-value of 0.05. These tests have been performed in the free statistical package PSPP.[1] The two tailed significance for the MSE is 0.621. As this is greater than 0.05 we must reject the null hypothesis and conclude that there is insufficient evidence to indicate that the mean of solution A and B are different. Thus, the solution B should be also excluded, and according to Definition 3.3, only the solutions A and D belongs to Pareto-set. For the sake of clarity the solutions are also shown in Fig. 1. Solutions B and C are excluded from the Pareto-set.

4. Structure optimisation of the RRNN

In this section, the structure optimisation algorithm of the RRNN is presented. In general terms, the algorithm consist of

[1] http://www.gnu.org/software/pspp/.

minimising simultaneously two functions: mean square error and number of neurons as stated in the following equations

$$\underset{W}{\text{minimise}} \begin{cases} f_1 = \dfrac{1}{2}\displaystyle\sum_{m=1}^{M}\sum_{i=1}^{N} c_i(g_i(q_{i,m}) - y_{i,m})^2 \\ f_2 = N \end{cases}$$

$$\text{subject to } [W, N] \in \Omega, \tag{10}$$

where f_1 is the error function for mth input–output pair, $c_i \in \{0, 1\}$, g_i is a differentiable function of neuron i and f_2 is the number of neurons.

4.1. Steps of the algorithm

Here, the steps for the structure optimisation algorithm are summarised as follows:

(1) *Initialise the matrices W and define minimum and maximum values of N.* The initiation of W is made at random among non-negative matrices. The learning rate is empirically chosen as $\eta = 0.1$ for all simulations. The minimum value of N should reckon of the number of inputs/outputs.
(2) *Define stop criteria.* In this paper, the tolerance of MSE and a maximum number of interactions are defined as stop criteria. For all simulations, the maximum number of interactions is 600. For each case study, an appropriate tolerance for MSE has been defined.
(3) *Perform an optimisation process to find a Pareto-set.* This step consist in a MOP, in which may be solved by several forms [48]. In this approach, the gradient descent method with multiple initial conditions has been adopted. In this method the system is initiated in different areas of parameter space. As the system is nonconvex, it is possible to achieve a sub-optimal solution. We proceed in such way for each value of N in a predefined range in step 2.
(4) *Stochastic nondominated exclusion.* As the system is nonconvex and there is no guarantee of global solution, dominated solutions should be removed. Here, as for each N there is a vector of values for MSE, it has been developed a comparison regarding the mean values of such vectors. Moreover, the claim that two means of MSE vectors are or are not different have been also tested [49]. This procedure is undertaken as stated in Definition 3.3.
(5) *Apply a decision-making task.* In this work, three strategies to choose the most suitable solution from a Pareto-set have been investigated. The first procedure stresses the minimum size of the network, that is, the solution of the Pareto-set chosen will be that one which presents the minimum value of N. This may be suitable for implementations of RRNN in hardware in which the number of variables may be a constraint. The second possible solution relies on minimum of mean square error. In this case, it is stressed the fitness to system. Finally, the solution may be chosen as a Euclidean norm of the two objectives. Objective function has been normalised into a scale of 0 to 1. This solution represents the minimum distance to utopian solution, that is a solution which minimises both f_1 and f_2. This may be used in occasions that the user should consider the complexity and the fitness of the RRNN simultaneously.

The pseudocode of this algorithm is presented in Algorithm 1. This algorithm was implemented in Matlab R2017a using a modified version of toolbox proposed in [50]. All simulations have been performed in a computer with Windows 8, Intel Core i5, RAM 8 GB, 64-Bits.

Algorithm 1: Pseudocode of the structure optimisation of the RRNN following the steps described in Section 4.1, which are emphasised in bold face. B and M are the number Pareto candidates and number of input/output pairs. W^* is the best solution according a decision criteria, which is identified by the operator $DM(\cdot)$. See [44] for a complete description of Multiobjective learning algorithm using the gradient method.

```
/* Step 1                                                    */
Input : Define the minimum and maximum values of N
        Set the learning rate μ ← 0.1 (default)
        Set the maximum number of interactions
τ_max ← 600 (default)
        Set the tolerance tol for each case study
        Load training and validation data
for b ← 1 to B do
    /* Step 2                                                */
    Initialise randomly the matrix W defined by Eq. (8)
    τ ← 1
    while (stop) do
        /* Step 3                                            */
        for m ← 1 to M do
            Δ_{b,0} ← 0
            for a ← 1 to A do
                Δ_{b,a} ← Δ_{b,a−1} + Θ_{b,a} ∂f_a/∂W
            end
            W_{b,τ+1} ← W_{b,τ} − ηΔ_{b,a}
        end
        τ = τ + 1
        if (τ > τ_max or MSE < tol then
            stop ← true
        end
        P_b^ℓ ← W_{b,τ+1}
        Ψ_b^ℓ ← f(W_{b,τ+1})
    end
    /* Step 4                                                */
    β ← [ ]
    for b ← 1 to B do
        if nand (Ψ^ℓ(b, 1) > Ψ^ℓ(:, 1), ··· , Ψ^ℓ(b, A) > Ψ^ℓ(:, A))
        then
            β ← [β  b]
        end
    end
    P ← P^ℓ(:, :, β)
    /* Step 5                                                */
    Decision-making task: W* ← DM(P)
end
Output: Chosen number (N) of Neurons and respective MSE
        Weights W* for the selected RNNN.
```

5. Case studies

In this section, four case studies illustrate the potentiality of the proposed method. For each value of N the optimal solution following the gradient descent with random initial condition was undertaken. The vertical bars represent one standard deviation, which means that 95% of solutions are expected to be inside this range. These results have presented in general a standard deviation for the solutions less than 2%. To present more comparisons with the proposed technique, other four classification cases have been presented in Section 5.5.

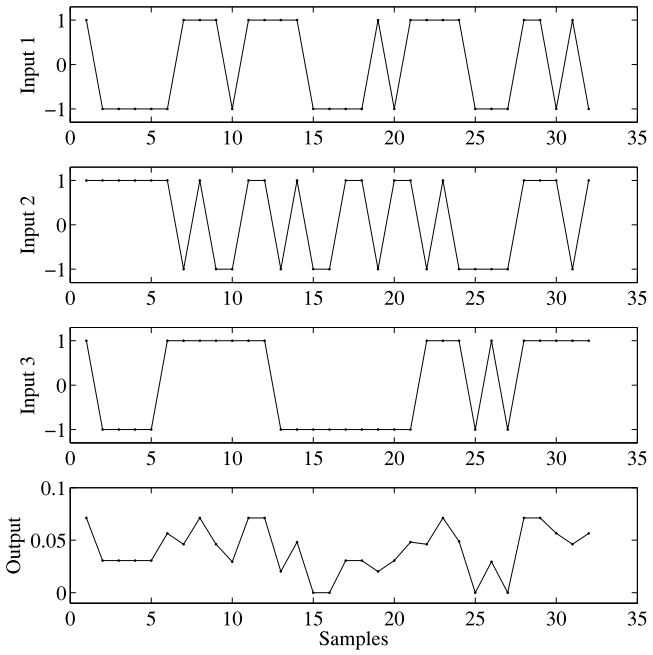

Fig. 2. Input and output of the RRNN system. The RRNN contains 14 neurons. Input is applied in first three neurons and output is collected from the 14*th* neuron.

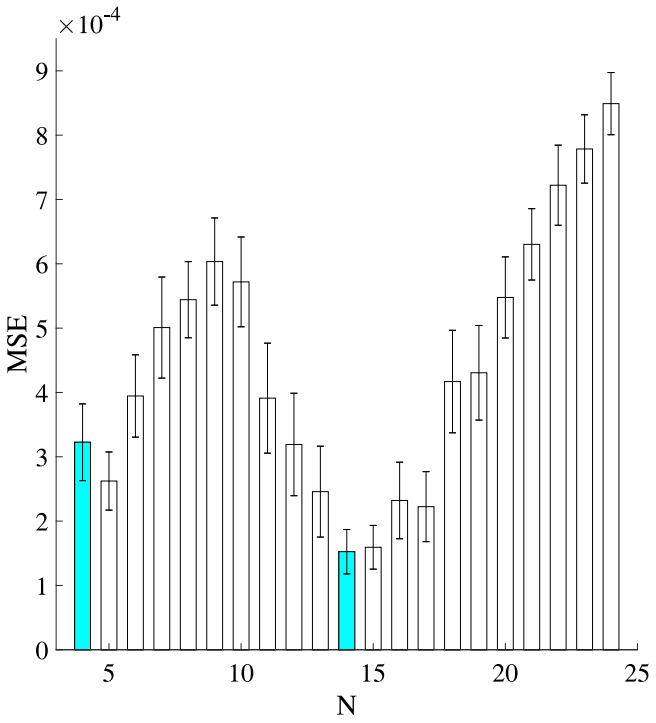

Fig. 3. The RNN system. Tolerance 0.0010. The x-axis represents the number of neurons N and the y-axis represents the mean square error (MSE) of training. The solutions marked with ∘ are the final solutions belonged to the Pareto-set.

5.1. The RRNN system

This theoretical case study aims at showing the property of the proposed method in producing the Pareto-set with optimal solution. This system has been designed with three inputs and one output. The RRNN contains 14 neurons and weights of the RRNN were generated arbitrary following a uniform distribution ranging from 0 to 0.2. The input consists of random values of $+1$ and -1 as shown in Fig. 2. The three inputs are applied in the first three neurons. It has been used 32 samples of the input. The output collected from 14*th* neuron of the RRNN is shown in Fig. 3. For this case, a range from 4 to 24 neurons has been examined. Applying the concept of stochastic nondominated (Definition 3.3), only two values of neurons were considered nondominated: 4 and 14. As one can see, if only the mean is compared, the Pareto-set would also include numbers 5 and 13. However, as these means do not present significant difference compared with other means in this set, the values with lower number of neurons were considered. It is important to emphasise that the precise number, that is 14, was obtained by our method, which suggests an interesting behaviour of the RRNN. Other feature is related to choose of the minimum number of neurons. That may indicate the ability of RRNN to represent a system using minimum information. Finally, the behaviour of MSE according to number of neurons can be used to design some policy in situations where resilience is required, for instance, in a real application, if one of the neurons is disconnected, according to Fig. 3, it could be better to reduce the number of neurons from 13 to 4, which is not an obvious policy. Moreover, it suggests the optimal N has the minimum value of MSE.

5.2. Diabetes classification

The Diabetes dataset was obtained from Proben1 [51]. This consist of 8 inputs settled on personal data and medical examinations. The output 1 or 0 indicates if the individual is diabetes positive or not, respectively. The original dataset is composed of 768 examples. In this study, 70 examples were used for training and 192 examples for validation. In this case, only the minimum structure possible, that is, with 9 neurons (one for output and eight for inputs) was chosen to Pareto-set, as seen in Fig. 4. Values of N was from 9 to 20. This best solution presents a MSE of 0.219 and regarding its capacity to classify, it was observed that for a different set of examples, the error of classification was 25.3%. This result is close to values obtained by [52], where the authors found 21.8% and 25.8% for Backpropagation and Levenberg–Marquardt respectively.

5.3. Liver disorder classification

The liver disorder classification is based on dataset from Irvine Machine Learning Database Repository called BUPA Liver Disorders. It comprises in 345 data points with 6 features [53]. The first five variables are blood clinical exams related to alcohol consumption. The sixth variable measures the number of alcohol drinks per day. The first 50 points have been used to train and the following 50 points to validate.

This case has been investigated using values of N from 7 to 26. In this case, only the first value was pointed out as a Pareto Minimise. It can be seen in Fig. 5 Values of MSE stayed in a close range. As a consequence, there was no significant difference between the mean of MSE vectors of the chosen result and other solutions. Comparing to published solution, this method achieved an error of 26.19%, while in the authors in [19] has achieved 35%. In both two cases of classification, this method was suitable to point out the minimum feasible number of N as the best solution for the RRNN.

5.4. Function approximation

The fourth case study is piecewise function described as in Eq. (11). For this single-input and single-output system we have used 25 examples.

$$f(x) = \begin{cases} 0.25x & \text{if} \quad 0 \leq x < 0.5 \\ 0.75x & \text{if} \quad 0.5 \leq x \leq 1 \end{cases} \tag{11}$$

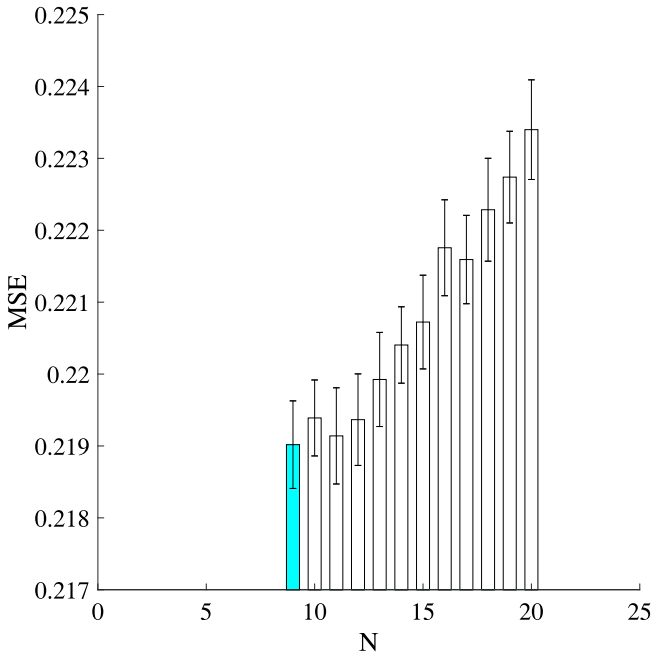

Fig. 4. Diabetes Classification. Tolerance 0.5000. The *x*-axis represents the number of neurons *N* and the y-axis represents the mean square error (MSE) of training. The solutions marked with ○ are the final solutions belonged to the Pareto-set.

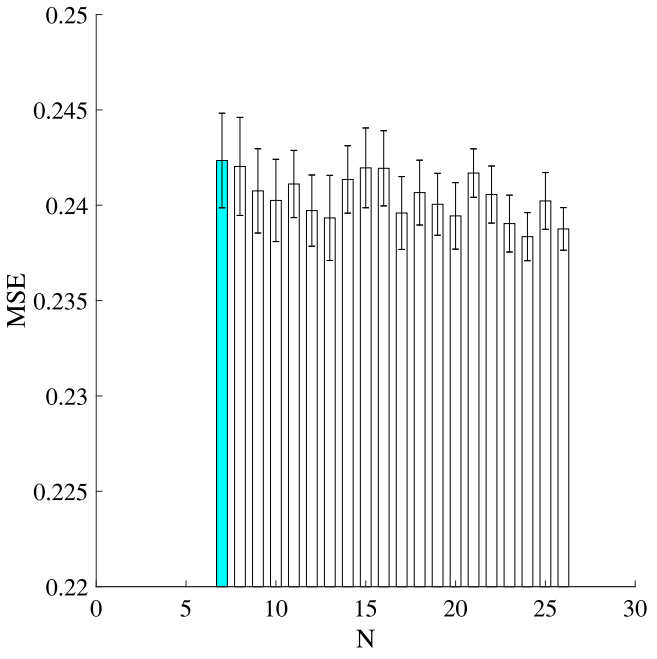

Fig. 5. Liver Disorder Classification. Tolerance 0.100. The *x*-axis represents the number of neurons *N* and the y-axis represents the mean square error (MSE) of training. The solutions marked with ○ are the final solutions belonged to the Pareto-set.

The fourth example was piecewise function. In this case, the Pareto-set was presented with three solutions, as shown in Fig. 6. $N = 2$ represents the solution in which decision-making task applies the minimum value for N as criteria. If one applied the minimum value for MSE, the best solution would be $N = 4$, while the $N = 3$ is the best solution for the case that Euclidean norm of the normalised objective functions is considered as criteria.

5.5. Additional examples

In this section, a comparison of the proposed RRNN with other techniques in the literature to classify patterns has been presented. We have presented four cases for classification, which results are summarised in Table 2. The Glass Identification presents 7 different values for the output and other three are binary classification. The second method of the decision making, that is, the minimum of MSE, has been adopted to chose the best solution.

Three cases presented performance similar with values in the literature. The exception is for the Breast Cancer Ljubljana, which our method has presented significant higher error. For this case, the level of 33% is not so different from the majority of results found in many works as reported by Dr. Duch[2] which is reported as 30%. Nevertheless, one of the reasons that it is possible to explain such difference is the fact that this is the unique dataset with missing values. This fact is certainly a feature that deserves more attention in future investigations. Here in this section, we have also reported the Liver Disorder, but now we have followed the same number of training data and test data as in [54]. It was possible to obtain a slight different value, but inferior to that presented in Section 5.3. This is another feature that should be analysed, that is the impact of the number of sample for the training and test. It has also been shown the Run Time for training the best solution. The RRNN is very sensitive to the number of attributes. With 34 attributes, the Johns Hopkins Ionosphere dataset has spent the longest time, compared to the Glass Identification dataset (10 attributes), which was 8 times less.

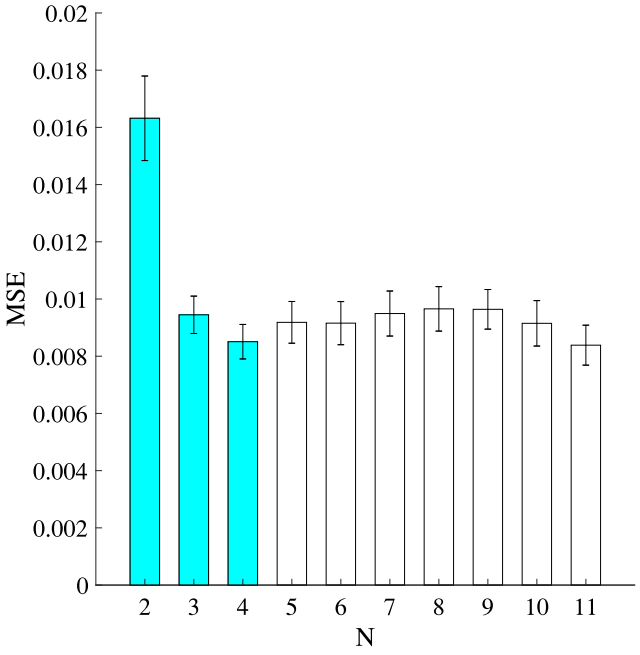

Fig. 6. Function Approximation. Tolerance 0.0001. The *x*-axis represents the number of neurons *N* and the *y*-axis represents the mean square error (MSE) of training. The solutions marked with ○ are the final solutions belonged to the Pareto-set.

6. Final remarks

This paper presents a novel method for structure selection of the RRNN using the MOP. This method aims at finding solutions in the Pareto-set based on two objectives, namely the number of neurons and the means square error (MSE). Usually, the exclusion

[2] See more details in http://www.is.umk.pl/~duch/projects/projects/datasets.html.

Table 2
Comparison of the proposed technique. It has been compared the classification accuracy of the Back Propagation Neural Network (BPNN) as proposed regarding the Ref. [54]. It has been presented the average and standard deviation of error and run time of the best solution.

Dataset	Literature		This paper			
	Error (%)	Ref.	Error (%)		Run time (s)	
			Avg.	Std.	Avg.	Std.
Johns Hopkins Ionosphere	9.7	[55]	8.9	0.8	256.5	32.9
Glass Identification	6.7	[56]	7.6	1.5	29.3	3.6
Breast Cancer Ljubljana	22.0	[57]	34.0	0.7	59.5	15.7
Liver Disorder	30.0	[54]	34.1	1.4	131.6	86.1

of dominated solutions in the Pareto-set has been addressed by the concept of dominance as described in [36]. However, this approach is not always suitable for situations, wherein the outcomes are stochastic, such as in the RRNN. To deal with the stochastic nature of the RRNN, the concept of stochastic nondominated has been introduced, which allows the exclusion of solution using statistical properties.

The results provide compelling evidence that the MOP is useful to replace heuristic approaches to choose the number of neurons for RRNN. The proposed technique has been applied in four examples. In the first case, we showed that this method is able to identify a Pareto-set in which the precise solution is present. Although, this example is a hypothetical one, it has been important to stress the ability of the technique to find the optimum number of neurons in a priori known scenario. This is not the common background, and thus, the other three cases were investigated to select the number of neurons, in which the best number of neurons are unknown. The two classification problems have shown the ability of MOP to provide a good number of neurons, which presents similar classification error when compared with other works in literature. It is important to stress that these examples present mean square error quite high. As the results are comparable with others in literature, it suggests that the independent variables are limited to a better explanation of the outcome. In fact, dealing with real data and in situations, such as diseases, this results are in good agreement with those found in literature. Durairaj and Kalaiselvi [58] have surveyed four different data mining algorithms. The error of classification ranges from 11% to 25% . In the fourth example, the RRNN has been applied to identify a system, a piecewise function, in which the Pareto-set presents three nondominated solutions. The choice of the best solution in this case, as well in the previous classification problems, can be easily obtained using a decision-making task described in Section 4.1. classification problems have shown the ability of MOP to provide a good number of neurons, which presents similar classification error when compared with other works in literature. It is important to stress that these examples present mean square error quite high. As the results are comparable with others in literature, it suggests that the independent variables are limited to a better explanation of the outcome. In fact, dealing with real data and in situations, such as diseases, this results are in good agreement with those found in literature. Durairaj and Kalaiselvi [58] have surveyed four different data mining algorithms. The error of classification ranges from 11% to 25% . In the fourth example, the RRNN has been applied to identify a system, a piecewise function, in which the Pareto-set presents three nondominated solutions. The choice of the best solution in this case, as well in the previous classification problems, can be easily obtained using a decision-making task described in Section 4.1.

Four additional classification cases have been investigated. The proposed technique has obtained comparable results in three of these four cases. Only in one case, Breast Cancer Ljubljana dataset the result has been distant from the best known outcome. Nevertheless, in this case, it has been possible to obtain results close to the average of other techniques (34% compared to the average of 30%). This dataset presents missing values, which can be inferred as one of the factors to explain the difference between these results. This fact should be investigated in a future work.

In this work, the learning process has been carried by means of adapted gradient optimisation method. It seems clear that future work should address multiobjective evolutionary algorithm [40, 59] to obtain the Pareto-set. Moreover, the influence of the number of samples and the number of attributes should be also examined as it presents an important factor to explain the increasing of run time.

Acknowledgments

This work was conducted amid a research visit at the Intelligent Systems and Networks Group, Dept. Electrical and Electronic Engineering, Imperial College London. It was financial aided by Postdoctoral Scholarship from CNPq/INERGE, Brazil.

References

[1] E. Gelenbe, Random neural networks with negative and positive signals and product form solution, Neural Comput. 1 (4) (1989) 502–510.
[2] E. Gelenbe, Stability of the random neural network model, Neural Comput. 2 (1990) 239–247.
[3] F.G.C. Cabarle, H.N. Adorna, M.J. Pérez-Jiménez, Sequential spiking neural p systems with structural plasticity based on max/min spike number, Neural Comput. Appl. 27 (5) (2016) 1337–1347.
[4] S. Basterrech, G. Rubino, Random neural network model for supervised learning problems, Neural Netw. World 25 (5) (2015) 457–499.
[5] E. Gelenbe, Learning in the recurrent random neural network, Neural Comput. 5 (1) (1993) 154–164.
[6] S. Timotheou, The random neural network: A survey, Comput. J. 53 (3) (2010) 251–267.
[7] E. Gelenbe, K.F. Hussain, Learning in the multiple class random neural network, IEEE Trans. Neural Netw. 13 (6) (2002) 1257–1267.
[8] E. Gelenbe, C. Cramer, Oscillatory corthico-thalamic response to somatosensory input, Biosystems 48 (1–3) (1998) 67–75.
[9] E. Gelenbe, Z. Mao, Y. Li, Function approximation with spiked random networks, IEEE Trans. Neural Netw. 10 (1) (1999) 3–9.
[10] E. Gelenbe, S. Timotheou, Synchronized interactions in spiked neuronal networks, Comput. J. 51 (6) (2008) 723–730.
[11] O.H. Abdelrahman, Detecting network-unfriendly mobiles with the random neural network, Probab. Engrg. Inform. Sci. 30 (3, SI) (2016) 514–531.
[12] D. Stosic, D. Stosic, C. Zanchettin, T. Ludermir, B. Stosic, QRNN: q-generalized random neural network, IEEE Trans. Neural Netw. Learn. Syst. 28 (2) (2017) 383–390.
[13] H. Bakırcıoğlu, T. Koçak, Survey of random neural network applications, European J. Oper. Res. 126 (2) (2000) 319–330.
[14] A. Vlontzos, The RNN-ELM classifier, in: 2017 International Joint Conference on Neural Networks (IJCNN), 2017-May, IEEE, 2017, pp. 2702–2707.
[15] A. Adeel, H. Larijani, A. Ahmadinia, Random neural network based cognitive engines for adaptive modulation and coding in LTE downlink systems, Comput. Electr. Eng. 57 (2017) 336–350.
[16] A. Javed, H. Larijani, A. Ahmadinia, D. Gibson, Smart random neural network controller for HVAC using cloud computing technology, IEEE Trans. Ind. Inf. 13 (1) (2017) 351–360.
[17] E. Gelenbe, Y. Yin, Deep learning with dense random neural networks, in: Advances in Intelligent Systems and Computing, vol. 659, Springer, 2018, pp. 3–18.
[18] W. Serrano, E. Gelenbe, The deep learning random neural network with a management cluster, in: Smart Innovation, Systems and Technologies, vol. 73, Springer, 2018, pp. 185–195.
[19] M. Georgiopoulos, C. Li, T. Kocak, Learning in the feed-forward random neural network: a critical review, Perform. Eval. 68 (4) (2011) 361–384.
[20] E.G. Nepomuceno, R.H.C. Takahashi, L.a. Aguirre, O.M. Neto, E.M.A.M. Mendes, Multiobjective nonlinear system identification: a case study with thyristor controlled series capacitor (TCSC), Internat. J. Systems Sci. 35 (9) (2004) 537–546.
[21] J. Sum, W.-K. Kan, G. Young, A note on the equivalence of NARX and RNN, Neural Comput. Appl. 8 (1) (1999) 33–39.
[22] Y. Suemitsu, S. Nara, A note on time delayed effect in a recurrent neural network model, Neural Comput. Appl. 11 (3–4) (2003) 137–143.
[23] J.-A. Désidéri, Multiple-gradient descent algorithm (MGDA) for multiobjective optimization, C.R. Math. 350 (5) (2012) 313–318.

[24] O. Abedinia, D. Raisz, N. Amjady, Effective prediction model for hungarian small-scale solar power output, IET Renew. Power Gener. 11 (13) (2017) 1648–1658.
[25] O. Abedinia, N. Amjady, Short-term load forecast of electrical power system by radial basis function neural network and new stochastic search algorithm, Int. Trans. Electr. Energy 26 (7) (2016) 1511–1525.
[26] M. Mohammadi, F. Talebpour, E. Safaee, N. Ghadimi, O. Abedinia, Small-scale building load forecast based on hybrid forecast engine, Neural Process. Lett. (2017) 1–23.
[27] I. Ullah, M.N.R. Baharom, H. Ahmad, F. Wahid, H.M. Luqman, Z. Zainal, B. Das, Smart lightning detection system for smart-city infrastructure using artificial neural network, Wirel. Pers. Commun. (2018) 1–22.
[28] Y. Solgi, S. Ganjefar, Variable structure fuzzy wavelet neural network controller for complex nonlinear systems, Appl. Soft Comput. 64 (2018) 674–685.
[29] T. Fujii, S. Sato, H. Nakahara., A threshold neuron pruning for a binarized deep neural network on an FPGA, IEICE Trans. Inf. Syst. E101.D (2) (2018) 376–386.
[30] F. Gaxiola, P. Melin, F. Valdez, J.R. Castro, Optimization of deep neural network for recognition with human iris biometric measure, in: Advances in Intelligent Systems and Computing, vol. 648, Springer, 2018, pp. 172–180.
[31] S. Takao, S. Kondo, J. Ueno, T. Kondo, Deep multi-layered GMDH-type neural network using revised heuristic self-organization and its application to medical image diagnosis of liver cancer, Artif. Life Robot. 23 (1) (2018) 48–59.
[32] B. Sen-Bhattacharya, S. James, O. Rhodes, I. Sugiarto, A. Rowley, A.B. Stokes, K. Gurney, S.B. Furber, Building a spiking neural network model of the basal ganglia on SpiNNaker, IEEE Trans. Cognit. Dev. Syst. 10 (3) (2018) 823–836.
[33] M. Moghaddari, F. Yousefi, M. Ghaedi, K. Dashtian, A simple approach for the sonochemical loading of Au, Ag and Pd nanoparticle on functionalized MWCNT and subsequent dispersion studies for removal of organic dyes: Artificial neural network and response surface methodology studies, Ultrason. Sonochem. 42 (2018) 422–433.
[34] H.A. Abbass, Speeding up backpropagation using multiobjective evolutionary algorithms, Neural Comput. 15 (11) (2003) 2705–2726.
[35] A. Kaylani, M. Georgiopoulos, M. Mollaghasemi, G.C. Anagnostopoulos, C. Sentelle, M. Zhong, An adaptive multiobjective approach to evolving ART architectures, IEEE Trans. Neural Netw. 21 (4) (2010) 529–550.
[36] D.A. Vieira, J.A. Vasconcelos, W.M. Caminhas, Controlling the parallel layer perceptron complexity using a multiobjective learning algorithm, Neural Comput. Appl. 16 (4–5) (2007) 317–325.
[37] I. Kokshenev, A.P. Braga, A multi-objective approach to RBF network learning, Neurocomputing 71 (7–9) (2008) 1203–1209.
[38] R.A. Teixeira, A.P. Braga, R.H. Takahashi, R.R. Saldanha, Improving generalization of MLPs with multi-objective optimization, Neurocomputing 35 (1–4) (2000) 189–194.
[39] V. Kadirkamanathan, G. Liu, Multiobjective criteria for neural network structure selection and identification of nonlinear systems using genetic algorithms, IEE Proc. Control Theory Appl. 146 (5) (1999) 373–382.
[40] C. Smith, Y. Jin, Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction, Neurocomputing 143 (SI) (2014) 302–311.
[41] S. Bureerat, K. Sriworamas, Simultaneous topology and sizing optimization of a water distribution network using a hybrid multiobjective evolutionary algorithm, Appl. Soft Comput. 13 (8) (2013) 3693–3702.
[42] N. Ghadimi, A. Akbarimajd, H. Shayeghi, O. Abedinia, Application of a new hybrid forecast engine with feature selection algorithm in a power system, Int. J. Ambient. Energy (2017) 1–10.
[43] O. Abedinia, N. Amjady, N. Ghadimi, Solar energy forecasting based on hybrid neural network and improved metaheuristic algorithm, Comput. Intell. 34 (1) (2018) 241–260.
[44] E.G. Nepomuceno, Multiobjective learning in the random neural network, Int. J. Adv. Intell. Paradig. 6 (1) (2013) 66–80.
[45] E.K. Chong, S.H. Zak, An Introduction to Optimization, vol. 76, John Wiley & Sons, New York, 2013.
[46] R.H. Takahashi, P.L. Peres, P.A. Ferreira, Multiobjective H2/H-inf guaranteed cost PID design, IEEE Control Syst. 17 (5) (1997) 37–47.
[47] R. Marler, J.S. Arora, Survey of multi-objective optimization methods for engineering, Struct. Multidiscip. Optim. 26 (6) (2004) 369–395.
[48] V. Chankong, Y. Haimes, Multiobjective Decision Making Theory and Methodology, North-Holland, New York, 1983.
[49] M.A. Shayib, Applied Statistics, Bookboon, 2013.
[50] H. Abdelbaki, Random neural network simulator for use with Matlab, Tech. Rep., University of Central Florida, 1999.
[51] L. Prechelt, PROBEN1 — A Set of Benchmarks and Benchmarking Rules for Neural Network Training Algorithms, Tech. Rep. 21/94, Fakultät für Informatik, Universität Karlsruhe, D-76128 Karlsruhe, Germany, 1994, Anonymous FTP: http://page.mi.fu-berlin.de/prechelt/Biblio//#Proben1.
[52] E. Alba, J.F. Chicano, Training neural networks with GA hybrid algorithms, in: Genetic and Evolutionary Computation–GECCO 2004, Springer, 2004, pp. 852–863.
[53] K. Bache, M. Lichman, UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences, 2013, http://archive.ics.uci.edu/ml.
[54] P. Jeatrakul, K. Wong, Comparing the performance of different neural networks for binary classification problems, in: 2009 Eighth International Symposium on Natural Language Processing, 2009, pp. 111–115.
[55] I. Tsoulos, D. Gavrilis, E. Glavas, Neural network construction and training using grammatical evolution, Neurocomputing 72 (1–3) (2008) 269–277.
[56] J. Siswantoro, A.S. Prabuwono, A. Abdullah, B. Idrus, A linear model based on Kalman filter for improving neural network classification performance, Exp. Syst. Appl. 49 (2016) 112–122.
[57] W. Duch, R. Adamczak, K. Grabczewski, G. Zal, Methodology of extraction, optimization and application of logical rules, in: Intelligent Information Systems VIII, vol. 659, Ustron, Poland, 2000, pp. 14–18.
[58] M. Durairaj, G. Kalaiselvi, Prediction of diabetes using soft computing techniques- A survey, Int. J. Sci. Technol. Res. 4 (3) (2015) 190–192.
[59] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.