Training of Feed-Forward Neural Networks for Pattern-Classification Applications Using Music Inspired Algorithm

Article · January 2011

CITATIONS
9

READS
422

2 authors:

Ali Kattan
Noble Institute, Erbil, Iraq
33 PUBLICATIONS **149** CITATIONS
SEE PROFILE

Rosni Abdullah
Universiti Sains Malaysia
109 PUBLICATIONS **615** CITATIONS
SEE PROFILE

Training of Feed-Forward Neural Networks for Pattern-Classification Applications Using Music Inspired Algorithm

Ali Kattan
School of Computer Science,
Universiti Sains Malaysia,
Penang 11800, Malaysia
kattan@cs.usm.my

Rosni Abdullah
School of Computer Science,
Universiti Sains Malaysia,
Penang 11800, Malaysia
rosni@cs.usm.my

***Abstract*—There have been numerous biologically inspired algorithms used to train feed-forward artificial neural networks such as generic algorithms, particle swarm optimization and ant colony optimization. The Harmony Search (HS) algorithm is a stochastic meta-heuristic that is inspired from the improvisation process of musicians. HS is used as an optimization method and reported to be a competitive alternative. This paper proposes two novel HS-based supervised training methods for feed-forward neural networks. Using a set of pattern-classification problems, the proposed methods are verified against other common methods. Results indicate that the proposed methods are on par or better in terms of overall recognition accuracy and convergence time.**

Keywords-harmony search; evolutionary methods; feed-forward neural networks; supervised training; pattern-classification

I. INTRODUCTION

Harmony Search (HS) is a relatively young meta-heuristic stochastic global optimization (SGO) method [1]. HS is similar in concept to other SGO methods such as genetic algorithms (GA), particle swarm optimization (PSO) and ant colony optimization (ACO) in terms of combining the rules of randomness to imitate the process that inspired it. However, HS draws its inspiration not from biological or physical processes but from the improvisation process of musicians. HS have been used successfully in many engineering and scientific applications achieving better or on par results in comparison with other SGO methods [2-6]. HS is being compared against other evolutionary based methods such as GA where a significant amount of research has already been carried out on the application of HS in solving various optimization problems [7-11]. The search mechanism of HS has been explained analytically within a statistical-mathematical framework [12, 13] and was found to be good at identifying the high performance regions of solution space within reasonable amount of time [14]. Enhanced versions of HS have been proposed such as the Improved Harmony Search (IHS) [15] and the Global-best Harmony Search (GHS) [16], where better results have been achieved in comparison with the original HS when applied on some integer programming problems. HS, IHS variants are being used in many recent works [17].

Evolutionary based supervised training of feed-forward artificial neural networks (FFANN) using SGO methods, such as GA, PSO and ACO has been already addressed in the literature [18-25]. The authors have already published a method for training FFANN for a binary classification problem (Cancer) [27] which has been cited in some recent works [28]. This work is an expanded version of the original work that includes additional classification problems and a more in depth discussion and analysis. In addition to the training method published in [27] this work presents the adaptation for the original IHS optimization method [15]. Then IHS is modified to produce the second method using a new criterion, namely the best-to-worst (BtW) ratio, instead of the improvisation count for determining the values of IHS's dynamic probabilistic parameters as well as the termination condition. Implementation considers pattern-classification benchmarking problems to compare the proposed techniques against GA-based training as well as the standard Backpropagation (BP) training.

The rest of this work is organized as follows. Section II presents a literature review of related work; Section III introduces the HS algorithm, its parameters and modeling; section IV introduces the IHS algorithm indicating the main differences from the original HS; section V introduces the proposed methods discussing the adaptation process in terms of FFANN data structure, HS memory remodeling and fitness function introducing a complete training algorithm and the initial parameters settings; section VI covers the results and discussion. Conclusions are finally made in section VII.

II. RELATED WORK

The supervised training of an artificial neural network (ANN) involves a repetitive process of presenting a training data set to the network's input and determining the error

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

between the actual network's output and the intended target output. The individual neuron weights are then adjusted to minimize such error and give the ANN its generalization ability. This iterative process continues until some termination condition is satisfied. This usually happens based on some measure, calculated or estimated, indicating that the current achieved solution is presumably good enough to stop training [29]. FFANN is a type of ANNs that is characterized by a topology with no closed paths and no lateral connections existing between the neurons in a given layer or back to a previous one [29]. A neuron in a given layer is fully connected to all neurons in the subsequent layer. The training process of FFANNs could also involve the network's structure represented by the number of hidden layers and the number of neurons within each [30-32]. FFANNs having a topology of just a single hidden layer, which sometimes referred to as 3-layer FFANNs, are considered as universal approximators for arbitrary finite-input environment measures [33-36]. Such configuration has proved its ability to match very complex patterns due to its capability to learn by example using relatively simple set of computations [37]. FFANNs used for pattern-classification have more than one output unit in its output layer to designate "classes" or "groups" belonging to a certain type [34, 38]. The unit that produces the highest output among other units would indicate the winning class, a technique that is known the "winner-take-all" [39, 40].

One of the most popular supervised training methods for FFANN is the BP learning [36, 41, 42]. BP is basically a trajectory-driven method, which is analogous to an error-minimizing process requiring that the neuron transfer function to be differentiable. The concept is illustrated in Fig. 1 using a 3-dimensional error surface where the gradient is used to locate minima points and the information is used to adjust the network weights accordingly in order to minimize the output error [43].

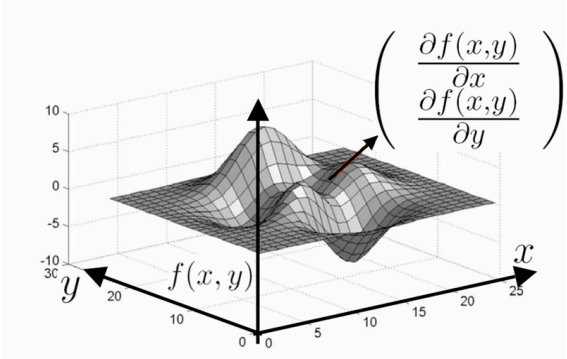

Figure 1. An illustration of the gradient-descent technique using a 3-dimensional error surface

However, BP is generally considered to be inefficient in searching for global minimum of the search space [44] since the BP training process is associated with two major problems; slow convergence for complex problems and local minima entrapment [36, 45]. ANNs tend to generate complex error surfaces with multiple local minima and trajectory-driven methods such as BP possess the possibility of being trapped in local solution that is not global [46]. Different techniques have been proposed to cure these problems to a certain extent including techniques such as simulated annealing and dynamic tunneling [36] as well as using special weight initialization techniques such as the Nguyen-Widrow method [39, 47, 48]. BP could also use a momentum constant in it's learning rule, where such technique accelerates the training process in flat regions of the error surface and prevents fluctuations in the weights [42].

Evolutionary supervised training methods offer an alternative to trajectory-driven methods. These are SGO techniques that are the result of combining an evolutionary optimization algorithm with the ANN learning process [31]. Evolutionary optimization algorithms are usually inspired form biological processes such as GA [44], ACO [49], Improved Bacterial Chemo-taxis Optimization (IBCO) [50], and PSO [51]. Such evolutionary methods are expected to avoid local minima frequently by promoting exploration of the search space. Their explorative search features differ from those of BP in that they are not trajectory-driven, but population driven. Using an evolutionary ANN supervised training model would involve using a fitness function where several types of these have been used. Common fitness functions include the ANN sum of squared errors (SSE) [20, 52, 53], the ANN mean squared error (MSE) [49-51], the ANN Squared Error Percentage (SEP) and the Classification Error Percentage (CEP) [18, 54]. The common factor between all of these forms of fitness functions is the use of ANN output error term where the goal is usually to minimize such error. Trajectory-driven methods such as BP have also used SSE, among others, as a training criterion [39, 43].

Many evolutionary-based training techniques have also reported to be superior in comparison with the BP technique [44, 49, 50, 54]. However, most of these reported improvements were based on using the classical XOR ANN problem. It was proven that the XOR problem has no local minima [55]. In addition, the size of the training data set of this problem is too small to generalize the superiority of any training method against others.

III. THE HARMONY SEARCH ALGORITHM

The process of music improvisation takes place when each musician in a band tests and plays a note on his instrument. An aesthetic quality measure would determine if the resultant tones are considered to be in harmony with the rest of the band. Such improvisation process is mostly noted in Jazz music where the challenge is to make the rhythm section sound as cool and varied as possible without losing the underlying groove [56].

Each instrument would have a permissible range of notes that can be played representing the pitch value range of that musical instrument. Each musician has three basic ways to improvise a new harmony. The musician would either play a totally new random note from the permissible range of notes, play an existing note from memory, or play a note from memory that is slightly modified. Musicians would keep and remember only good improvised harmonies till better ones are found and replace the worst ones.

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

The basic HS algorithm proposed by Lee and Geem [57] and referred to as "classical" [13] uses the above scenario as an analogy where note played by a musician represents one component of the solution vector of all musician notes and as shown in Fig. 2.

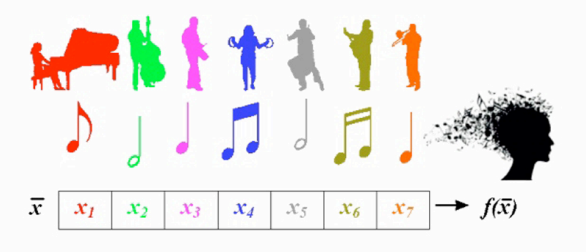

Figure 2. Music improvisation process for a harmony in a band of seven

The best solution vector is found when each component value is optimal based on some objective function evaluated for this solution vector [3]. The number of components in each vector N represents the total number of decision variables and is analogous to the tone's pitch, i.e. note values played by N musical instruments. Each pitch value is drawn from a pre-specified range of values representing the permissible pitch range of that instrument. A Harmony Memory (HM) is a matrix of the best solution vectors attained so far. The HM size (HMS) is set prior to running the algorithm. The ranges' lower and upper limits are specified by two vectors $\mathbf{x}^{\mathbf{L}}$ and $\mathbf{x}^{\mathbf{U}}$ both having the same length N. Each harmony vector is also associated with a harmony quality value (fitness) based on an objective function ***f(x)***. Fig. 3 shows the modeling of HM.

Improvising a new-harmony vector would consider each decision variable separately where HS uses certain parameters to reflect playing probabilistic choices. These are the Harmony Memory Considering Rate (HMCR) and the Pitch Adjustment Rate (PAR). The former determines the probability of playing a pitch from memory or playing a totally new random one. The second, PAR, determines the probability of whether the pitch that is played from memory is to be adjusted or not. Adjustment value for each decision variable is drawn from the respective component in the bandwidth vector **B** having the size N.

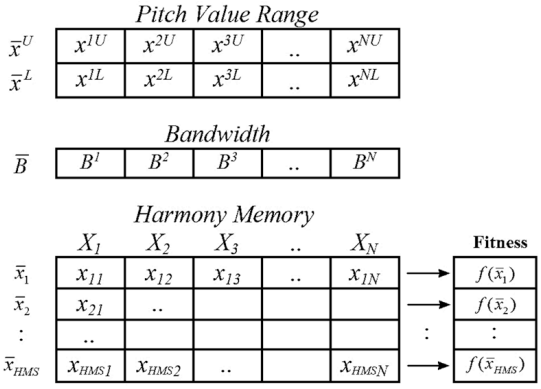

Figure 3. The modeling of HM with N decision variables

The adjustment process should guarantee that the resultant pitch value is within the permissible range specified by $\mathbf{x}^{\mathbf{L}}$ and $\mathbf{x}^{\mathbf{U}}$. The classical HS algorithm pseudo code is given in Algorithm 1.

1 Initialize the algorithm parameters (HMS, HMCR, PAR, **B**, MAXIMP)
2 Initialize the harmony memory HM with random values drawn from vectors $[\mathbf{x}^{\mathbf{L}},\mathbf{x}^{\mathbf{U}}]$
3 Iteration itr=0
4 **While** itr < MAXIMP **Do**
5 Improvise new harmony vector ***x'***
Harmony Memory Considering:
6 $x_i' \leftarrow \begin{cases} x_i' \in \{x_{i1}, x_{i2}, ..., x_{iHMS}\} & \text{with probability HMCR} \\ x_i' \in X_i & \text{with probability (1-HMCR)} \end{cases}$
If probability HMCR **Then**
Pitch adjusting:
7 $x_i' \leftarrow \begin{cases} x_i' \pm rand(0,1) \cdot B_i & \text{with probability PAR} \\ x_i' & \text{with probability (1-PAR)} \end{cases}$
Bounds check:
$x_i' \leftarrow \min(\max(x_i', x_i^L), x_i^U)$
EndIf
8 If x' is better than the worst harmony in HM Then Replace worst harmony in HM with x'
9 itr= itr+1
10 **EndWhile**
11 Best harmony vector in HM is the solution

Algorithm 1. Pseudo code for the classical HS algorithm

The improvisation process is repeated iteratively until a maximum number of improvisations MAXIMP is reached. Termination in HS is determined solely by the value of MAXIMP. The choice of this value is a subjective issue and has nothing to do with the quality of the best-attained solution [16, 58, 59].

The use of solution vectors stored in HM is similar to the genetic pool in GA in generating offspring based on past information [10]. However, HS generates a new solution vector utilizing all current HM vectors not just two (parents) as in GA. In addition, HS would consider each decision variable independently without the need to preserve the structure of the gene.

IV. The Improved Harmony Search Algorithm

Mahdavi et al. [15] have proposed the IHS algorithm for better fine-tuning of the final solution in comparison with the classical HS algorithm. The main difference between IHS and the classical HS is that the two probabilistic parameters namely PAR and B, are not set statically before run-time rather than being adjusted dynamically during run-time as a function of the current improvisation count, i.e. iteration (*itr*), bounded by MAXIMP. PAR would be adjusted in a linear fashion as given in equation (1) and shown in Fig. 4(a). B on the other hand would decrease exponentially as given in equation (2) and (3) and shown in Fig. 4(b). Referring to classical HS given in Algorithm 1, this adjustment process takes place just before improvising new harmony vector (line 5). PAR_{min} and PAR_{max} would replace the initial parameter PAR and B_{max} and B_{min} would replace the initial parameter B (line 1).

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

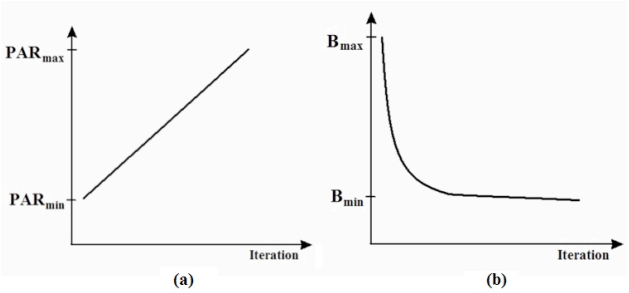

Figure 4. The adjustment of the probablistic parameters in IHS
(a) dynamic PAR value increases linearly as a function of iteration number,
(b) dynamic B value decreases exponentially as a function of iteration number

$$PAR(itr) = PAR_{\min} + \frac{PAR_{\max} - PAR_{\min}}{MAXIMP} \times itr \quad (1)$$

$$B(itr) = B_{\max} \exp(c \cdot itr) \quad (2)$$

$$c = \ln(\frac{B_{\min}}{B_{\max}}) / MAXIMP \quad (3)$$

PAR, which determines if the value selected from HM is to be adjusted or not, starts at PAR_{min} and increases linearly as a function of the current iteration count with a maximum limit at PAR_{max}. So as the iteration count becomes close to MAXIMP, pitch adjusting would have a higher probability. On the other hand B, the bandwidth, starts high at B_{max} and decreases exponentially as a function of the current iteration count with a minimum limit at B_{min}. B tends to be smaller in value as the iteration count reaches MAXIMP allowing smaller changes.

V. Proposed Methods

The proposed supervised FFANN training method considers the aforementioned IHS algorithm suggested in [15]. In order to adapt IHS for such a task, suitable FFANN data structure, fitness function, and training termination condition must be devised. In addition, the HM must be remodeled to suit the FFANN training process. Each of these is considered in the following sections.

A. FFANN data structure

Real-coded weight representation was used in GA-based ANN training methods, where such technique proved to be more efficient in comparison with the binary-coded one [52, 53]. It has been shown that binary representation is neither necessary nor beneficial and it limits the effectiveness of GA [46]. The vector representation from the Genetic Adaptive Neural Network Training (GANNT) algorithm originally introduced by Dorsey et al. [18, 20, 53, 60] was adopted for the proposed method. Fig. 5 illustrates such representation for a small-scale sample FFANN. Each vector represents a complete set of FFANN weights including biases where weight values are treated as genes. Neurons respective weights are listed in sequence assuming a fixed FFANN structure.

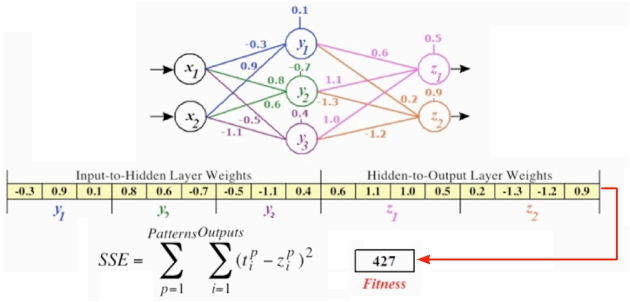

Figure 5. Harmony vector representation of FFANN weights

B. HM remodeling

Since FFANN weight values are usually within the same range, the adapted IHS model could be simplified by using fixed ranges for all decision variables instead of the vectors $\mathbf{x}^L$, $\mathbf{x}^U$ and $\mathbf{B}$. This is analogous to having the same musical instrument for each of the N decision variables. Thus the scalar range $[x^L, x^U]$ would replace the vectors $\mathbf{x}^L$, $\mathbf{x}^U$ and the scalar value B would replace the vector $\mathbf{B}$. The B value specifies the range of permissible weight changes given by the range [-B,B]. The remodeled version of HM is shown in Fig. 6 with one "Fitness" column. If the problem considered uses more than one fitness measure then more columns are added.

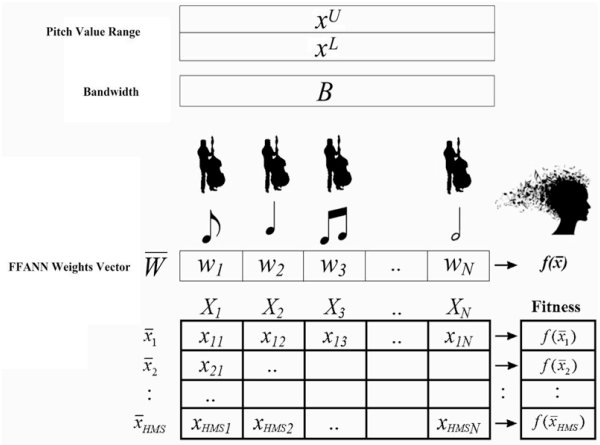

Figure 6. Adapted HM model for FFANN training

C. Fitness function & HS-based training

The proposed method uses SSE as its main fitness function where the goal is to minimize the amount of this error [43]. SSE is the squared difference between the target output and actual output and this error is represented as $(t\text{-}z)^2$ for each pattern and each output unit and as shown in Fig. 5. Calculating SSE would involve doing FFANN forward-pass calculations to compare the resultant output with target output. Equations (4) through (6) give these calculations assuming a bipolar sigmoid neuron transfer function [39].

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

Considering the use of FFANNs for pattern classification networks, CEP, given in (7), could be used to complement SSE's raw error values since it reports in a high-level manner the quality of the trained network [54].

$$SSE = \sum_{p=1}^{P}\sum_{i=1}^{S}(t_i^p - z_i^p)^2 \quad (4)$$

$$y = \sum_{i=1}^{n+1} w_i x_i \quad (5)$$

$$z = F(y) = \frac{2}{1+e^{-y}} - 1 \quad (6)$$

$$CEP = \frac{E_p}{P} \cdot 100\% \quad (7)$$

where

P	total number of training patterns
S	total number of output units (classes)
t	target output (unit)
z	actual neuron output (unit)
y	sum of the neuron's input signals
w_i	the weight between this neuron and unit i of previous layer (wn+1 represents bias)
x_i	input value from unit I of previous layer (output of that unit)
$n+1$	total number of input connections including bias
$F(y)$	neuron transfer function (bipolar sigmoid)
E_p	total number of incorrectly recognized training patterns

The flowchart shown in Fig. 7 presents a generic HS-based FFANN training approach that utilizes the HM model introduced above. The algorithm would start by initializing the HM with random harmony vectors representing candidate FFANN weight vector values. A separate module representing the problem's FFANN computes each vector's fitness individually. This occurs by loading the weight vector into the FFANN structure first then computing the fitness measure, such as SSE and CEP, for the problem's data set by performing forward-pass computations for each training pattern. Then each vector is stored in HM along with its fitness value(s). An HM fitness measure could be computed upon completing the initialization process. Such measure would take into considerations all the HM fitness values stored such as an average fitness. The training would then proceed in a similar fashion to Algorithm 1 by improvising new weight vector, finding its fitness and deciding whether to insert in HM or not. The shaded flowchart parts in Fig. 7 are to be customized by each of the IHS-based proposed training methods introduced next.

D. *The adapted IHS-based training algorithm*

The IHS algorithm is adapted to use the data structure and the remodeled HM introduced above. The newly improvised harmony is accepted if its SSE value is less than that of the worst in HM and its CEP value is less than or equal to the average value of CEP in HM. The latter condition would guarantee that the newly accepted harmonies would yield the same or better overall recognition percentage. The justification can be explained by considering the winner-take-all approach used for the pattern-classification problems considered. Lower CEP values are not necessarily associated with lower SSE values. This stems from the fact that even if the SSE value is small, it is the winner class, i.e. the one with the highest value, is what determines the result of the classification process.

Fig. 8 shows the flowchart of the adapted IHS training algorithm, which is a customized version of the one given earlier in Fig. 7. Improvising a new harmony vector, which is a new set of FFANN weights, is given as pseudo code of Algorithm 2.

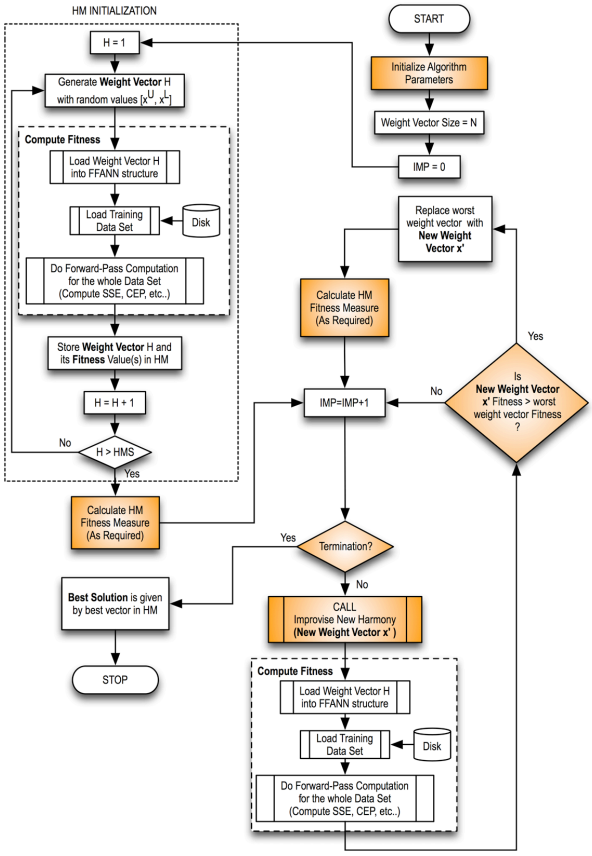

Figure 7. Generic FFANN training using adapted HS-based algorithm

E. *The modified IHS-based training algorithm using BtW ratio*

In the plain version of the adapted IHS training algorithm discussed in the previous section, MAXIMP value would affect the rate of change for PAR and B as well as being the only termination condition of the algorithm. Selecting a value for

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

MAXIMP is a subjective issue that is merely used to indicate the total number of times the improvisation process is to be repeated. The modified version of IHS uses a quality measure of HM represented by the BtW criterion. BtW is a new parameter representing the ratio of the current best harmony fitness to the current worst harmony fitness in HM. With SSE taken as the main fitness function, the BtW value is given by the ratio of the current best harmony SSE to the current worst harmony SSE and as given in equation (8).

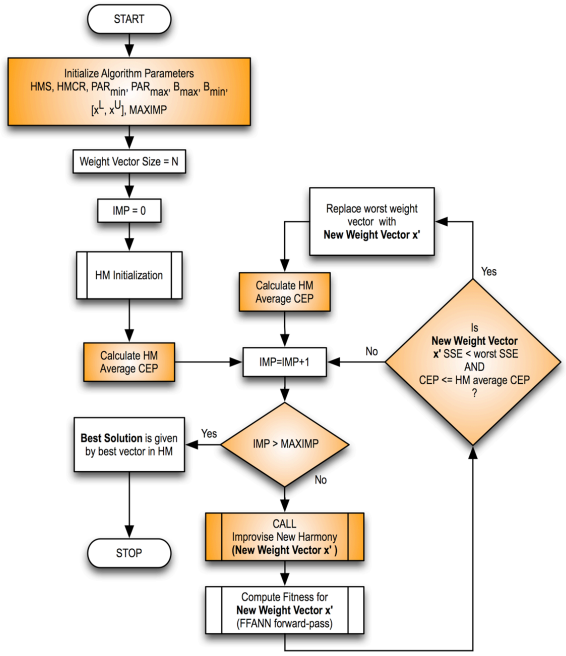

Fig. 8. FFANN training using adapted IHS algorithm

```
1   Create new harmony vector x' of size N
2   For i=0 to N do
3     RND= Random(0,1)
4     If (RND<=HMCR) //harmony memory considering
5        RND= Random(0,HMS)
6        x'(i)= HM(RND,i) //harmony memory access
7         PAR= PARmin+(PARmax-PARmin)/MAXIMP)*itr
8         C= ln(Bmin/Bmax)/MAXIMP
9         B= Bmax*exp(C*itr)
10        RND= Random(0,1)
11        If (RND<=PAR) //Pitch Adjusting
12          x'(i)= x'(i) + Random(-B,B)
13          x'(i)= min(max(x'(i),xL),xU)
14        EndIf
15    Else //random harmony
16        x'(i) = Random(xU,xL)
17    EndIf
18  EndFor
19  Return x'
```

Algorithm 2: Pseudo code for improvising new harmony vector in IHS

$$BtW = \frac{SSE_{BestHarmony}}{SSE_{WorstHarmony}} \quad (8)$$

The concept of Best-to-Worst was inspired from the fact that the words "best" and "worst" are part of the HS algorithm nomenclature. The algorithm basically tries to find the "best" solution among a set of solutions stored in HM by improvising new harmonies to replace those "worst" ones. At any time HM would contain a number of solutions including best solution and worst solution in terms of their stored quality measures, i.e. fitness function values. If the worst fitness value in HM is close to that of the best, then this basically indicate that the quality of all current harmony vectors are almost as good as that of the best. This is somewhat similar to GA-based training methods when the percentage of domination of a certain member in the population could be used to signal convergence. Such domination is measured by the existence of a certain fitness value among the population. The BtW value would range between zero and one where values close to one indicate that the average fitness of harmonies in the current HM is close to the current best; a measure of stagnation. From another perspective, the BtW ratio would actually indicate the size of the area of the search space that is currently being investigated by the algorithm. Thus values close to zero would indicate a large search area while values close to one would indicate smaller areas.

The modified version of the adapted IHS training algorithm is referred to as the HS-BtW training algorithm. The BtW ratio would be used for dynamically adjusting the values of PAR and B as well as determining the training termination condition. A threshold value BtW_{thr} controls the start of PAR and B dynamic change and as shown in Fig. 9. This is analogues to the dynamic setting for the parameters of IHS given earlier in Fig. 4. Setting BtW_{thr} to 1.0 would make the algorithm behave just like the classical HS such that PAR is fixed at PAR_{min} and B is fixed at B_{max}. The BtW_{thr} value is determined by calculating BtW of the initial HM vectors prior to training.

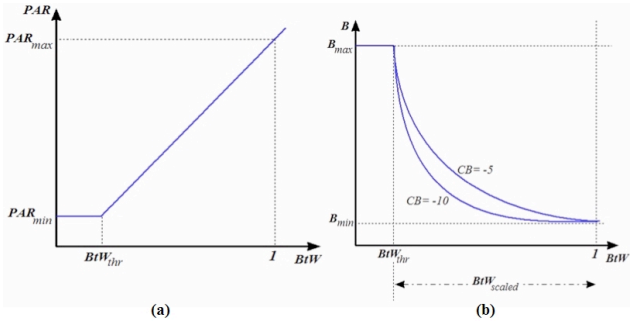

Figure 9. The dynamic PAR & B parameters of HS-BtW
(a) dynamic PAR value increases linearly as a function of the current HM BtW ratio, (b) dynamic B value decreases exponentially as a function of the current HM BtW ratio

PAR would be calculated as a function of the current BtW value and as given in equation (9) and (10) where *m* gives the line slop past the value of BtW_{thr}. B is also a function of the current BtW value and as given in equation (11) and (12) where *CB* is a constant controlling the steepness of change and it's in the range of [-10,-2] (based on empirical results) BtW_{scaled} is the value of BtW past the BtW_{thr} point scaled to be in the range [0,1].

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

The termination condition is based on $BtW_{termination}$ value that is set close to, but less than, unity. Training will terminate if $BtW >= BtW_{termination}$. MAXIMP is added as an extra termination criterion to limit the total number of training iterations if intended.

$$PAR(BtW) = \begin{cases} PAR_{min} & \text{if } BtW < BtW_{thr} \\ m(BtW - 1) + PAR_{max} & \text{if } BtW \geq BtW_{thr} \end{cases} \quad (9)$$

$$m = \frac{PAR_{max} - PAR_{min}}{1 - BtW_{thr}} \quad (10)$$

$$B(BtW) = \begin{cases} B_{max} & \text{if } BtW < BtW_{thr} \\ (B_{max} - B_{min})\exp(CB \cdot BtW_{scalled}) + B_{min} & \text{if } BtW \geq BtW_{thr} \end{cases} \quad (11)$$

$$BtW_{scalled} = \frac{(BtW - BtW_{thr})}{1 - BtW_{thr}} \quad (12)$$

where

BtW	Best-to-Worst ratio
BtW_{thr}	threshold value to start dynamic change
PAR_{min}	minimum pitch adjusting rate
PAR_{max}	maximum pitch adjusting rate
B_{min}	minimum bandwidth
B_{max}	maximum bandwidth
CB	constant controlling the steepness of B change

The flowchart shown in Fig. 10 shows the proposed HS-BtW training method, along with the pseudo code for improvising a new harmony vector in Algorithm 3. Both of these are analogous to adapted IHS flowchart given in Fig. 8 and improvisation process given in Algorithm 2. The IHS-based training method introduced earlier used two quality measures namely SSE and CEP where it was also indicated that SSE could be used as the sole fitness function. The HS-BtW method uses SSE only as its main fitness function in addition to using the BtW value as a new quality measure. Based on the BtW concept, the HS-BtW algorithm must compute this ratio in two places: after HM initialization process and after accepting a new harmony. The BtW value computed after HM initialization is referred to as BtW threshold (BtW_{thr}) used by equation (9) through (12). BtW is recomputed upon accepting a new harmony vector and the value would be used to dynamically set the value of PAR and B as well as to determine the termination condition. The HS-BtW improvisation process given in Algorithm 3 applies the newly introduced formulas given in equation (9) through (12).

F. Initial parameter values

The original IHS was used as an optimization method in many problems where the HMS value of 10 was encountered in many parameter estimation problems [61,9]. However it was indicated that no single choice of HMS is superior to others [16] and it is clear that in the case of FFANNs training more calculations would be involved if HMS were made larger.

HMCR was set to 0.9 or higher in many applications [58,59,57]. Based on the recommendations outlined by Omran et al [16], HMCR should be set such that HMCR≥0.9 for high dimensionality problems, which in this case resembles the total number of FFANN weights. It was also recommended to use relatively small values for PAR such that PAR≤0.5. The bandwidth B parameters values were selected based on several experimental tests in conjunction with selected $[x^L, x^U]$ range. Finally the termination condition would be achieved either if the value of $BtW \geq BtW_{termination}$, where $BtW_{termination}$ is set close to unity, or reaching the maximum iteration count specified by MAXIMP. Values like 5000, 10000 or higher were commonly used for MAXIMP in many applications [58,59,16].

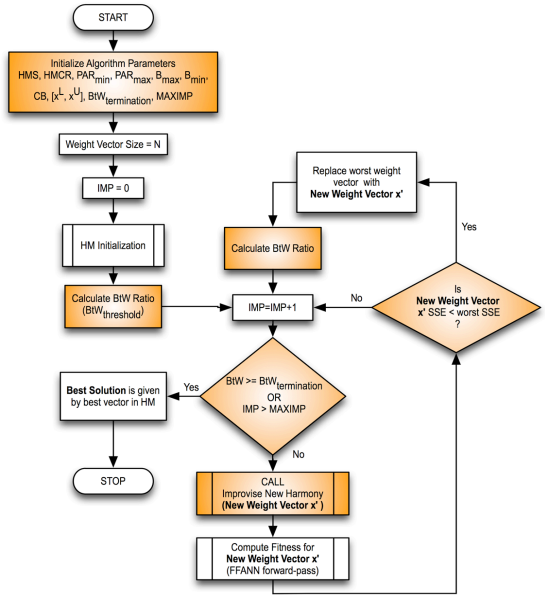

Figure 10. FFANN training using the HS-BtW algorithm

```
1   Create new harmony vector x' of size N
2   For i=0 to N do
3     RND= Random(0,1)
4     If (RND<=HMCR) //harmony memory considering
5       RND= Integer(Random(0,HMS))
6       x'(i)= HM(RND,i) //harmony memory access
7       If (BtW<BtW_threshold)
8         PAR= PAR_min
9         B= B_max
10      Else
11        m= (PAR_max-PAR_min)/(1-BtW_threshold)
12        PAR= m(BtW-1)+ PAR_max
13        BtW_scaled= CB(BtW-BtW_threshold)/(1-BtW_threshold)
14        B= (B_max- B_min)exp(BtW_scaled)+ B_min
15      EndIf
16      RND= Random(0,1)
17      If (RND<=PAR) //Pitch Adjusting
18        x'(i)= x'(i) + Random(-B,B)
19        x'(i)= min(max(x'(i),x^L),x^U)
20      EndIf
21    Else //random harmony
22      x'(i) = Random(x^U,x^L)
23    EndIf
24  EndFor
25  Return x'
```

Algorithm 3: Pseudo code for improvising new harmony vector in HS-BtW

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

VI. RESULTS AND DISCUSSION

In order to demonstrate the performance of the proposed methods, five different pattern-classification benchmarking problems were obtained from UCI Machine Learning Repository[1] [62] for the experimental testing and evaluation. The selected classification problems listed in Table (1) are taken from different fields including medical research, biology, engineering and astronomy. One of the main reasons behind choosing these data sets among many others is that they had no or very few missing input feature values. In addition these problems have been commonly used and cited in the neural networks, classification and machine learning literature [63-71]. All the patterns of a data set were used except for the Magic problem where only 50% out of the original 19,020 patterns of the data were used in order to perform the sequential computation within feasible amount of time. Some other pre-processing tasks were also necessary. For instance, in the Ionosphere data set there were 16 missing values for input attribute 6. These were encoded as 3.5 based on the average value of this attribute.

A 3-layer FFANN, represented by input-hidden-output units in Table 1, was designed for each to work as a pattern-classifier using the winner-take-all fashion [43]. The data set of each problem was split into two separate files such that 80% of the patterns are used as training patterns and the rest as out-of-sample testing patterns. The training and testing files were made to have the same class distribution, i.e. equal percentages of each pattern type. Data values of the pattern files where normalized to be in the range [-1,1] in order to suit the bipolar sigmoid neuron transfer function given in equation (6).

TABLE 1. BENCHMARKING DATA SETS

Data Set	Training Patterns	FFANN Structure	Weights
Iris	150	4-5-3	43
Magic	9,510	10-4-2	54
Diabetes	768	8-7-2	79
Cancer	699	9-8-2	98
Ionosphere	351	33-4-2	146

For implementation Java 6 was used and all tests were run individually on the same computer in order to have comparable results in terms of the overall training time. The programs generate iteration log files to store each method's relevant parameters upon accepting an improvisation. The initial parameters values for each training method considered in this work are given in Table 2. GANNT and BP training algorithms were used for the training of the five aforementioned pattern-benchmarking classification problems to serve as a comparison measure against the proposed method.

The results for each of the benchmarking problems considered are aggregated in one table and are listed in Table 3 through Table 7. For each problem, ten individual training tests were carried out for each training method (M) considered. The best result out of the ten achieved by each method is reported for that problem. The aim is to train the network to obtain maximum overall recognition accuracy within the least amount of training time. Thus all comparisons consider the overall recognition accuracy as the first priority and the overall all training time as a second. The "Overall Time" in these tables represents the overall computing time required by each method to complete the whole training process. Some fields are not applicable for some methods and these are marked with (N.A.). For BP and GANNT the "Total Accepted" column represents the total number of training iterations needed by these methods.

TABLE 2. INITIAL PARAMETER VALUES USED BY TRAINING ALGORITHMS

M	Parameter	Values
IHS	HMS	10, 20
	HMCR	0.97
	PAR_{min}, PAR_{max}	0.1, 0.45
	B_{max}, B_{min}	5.0, 2.0
	$[x^L, x^U]$	[-250, 250]
	MAXIMP	5000, 20000
HS-BtW	HMS	10, 20
	HMCR	0.97
	PAR_{min}, PAR_{max}	0.1, 0.45
	B_{max}, B_{min}	5.0, 2.0
	CB	-3
	$[x^L, x^U]$	[-250, 250]
	$BtW_{termination}$	0.99
	MAXIMP	20000
GANNT	Population Size	10
	Crossover	At k=rand(0,N), if k=0 no crossover
	Mutation Probability	0.01
	Value Range [min,max]	[-250, 250]
	Stopping Criterion	50% domination of certain fitness
BP	Learning Rate	0.008
	Momentum	0.7
	Initial Weights	[-0.5, 0.5]
	Initialization Method	Nguyen-Widrow
	Stopping Criterion	SSE difference<= 1.0E-4

A. The adapted IHS training method

Since MAXIMP would determine the algorithm's termination condition, two values were used for testing, a lower value of 5000 and a higher value of 20,000. More iterations would give better chances for the algorithm to improvise more accepted improvisations. Results indicated by the IHS rows of Table 3 through 7 show that there are generally two trends in terms of the overall recognition percentage. In some problems, namely Magic, Diabetes and Ionosphere given in Table 4, 5 and 7 respectively, increasing MAXIMP would result in attaining better overall recognition percentage. The rest of the problems, namely Iris and Cancer given in Table 3 and 6 respectively, the resultant overall recognition percentage has decreased. Such case is referred to as "overtraining" or "overfitting" [43,72]. Training the network more than necessary would cause it to eventually lose its generalization ability to recognize out-of-sample patterns since it becomes more accustomed to the training set used. In general the best results achieved by the adapted IHS method are on par with those achieved by BP and GANNT rival methods. The IHS method scored best in the Iris, Cancer and Ionosphere problems given in Table 3, 6 and 7 respectively. BP scored best in the Magic problem given in Table 4 and GANNT scored best in the Diabetes problem given in Table 5.

[1] For full citations and data sets download see http://archive.ics.uci.edu/ml

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

Tests were also conducted using a double HMS value of 20. However, the attained results for all problems were the same as those attained using an HMS value of 10 but with longer overall training time and hence not reported in the results tables. For the problems considered in this work such result seem to coincide with that mentioned in [16] stating that no single choice of HMS is superior to others. Unlike the GA-based optimization methods, the HMS used by HS is different from that of the population size used in GANNT method. The HS algorithm and its dialects replace only the worst vector of HM upon finding a better one. Increasing the HMS would allow more vectors to be inspected but has no effect on setting the probabilistic values of both PAR and B responsible for the stochastic improvisation process and fine-tuning the solution. These values are directly affected by the current improvisation count and the MAXIMP value.

B. The HS-BtW training method

The adapted IHS method introduced in the previous section has achieved on par results in comparison with BP and GANNT. However, termination as well as the dynamic settings of PAR and B depended solely on the iteration count bounded by MAXIMP. The HS-BtW method has been used for the training of the same set of benchmarking problems using the same HMS value of 10. The results are given in the HS-BtW rows of Table 3 through 7. In comparison with IHS, BP and GANNT, the HS-BtW method scored best in the Iris, Diabetes and Cancer problems given in Table 3, 5 and 6 respectively. Sub-optimal results were obtained in the Magic and Ionosphere problems given in Table 4 and 7 respectively. However, due to its new termination condition and PAR and B settings technique, HS-BtW achieved convergence in much less number of total iterations and hence overall training time. The overall training time is the same as the last accepted improvisation time since termination occurs upon accepting an improvisation that yields BtW value equal or larger than $BtW_{termination}$.

Unlike the former adapted IHS, the HMS would have a direct effect on the HS-BtW performance since it affects the computed BtW ratio. Having a higher HMS would increase the solution space and the distance between the best solution and the worst solution. Tests were repeated using a double HMS value of 20 for all problems. The method attained the same results but with longer overall training time for Iris, Diabetes and Cancer problems given in Table 3, 5 and 6 respectively, and hence these were not included in the relevant results tables. This indicates that the HMS value of 10 is sufficient for these problems. However, HS-BtW was able to score higher in both the Magic problem and the Ionosphere problem given in Table 4 and 7 respectively when using an HMS value of 20. For the Magic problem, BP still holds the best score. The justifications for this is that BP has an advantage over the other considered methods when the training data set is relatively larger (see Table 1). Such increase in the number of training patterns will enable BP to have better fine-tuning attributed to its trajectory-driven approach. Table 8 summarizes the best results achieved by the IHS training method against those of the HS-BtW training method for the problems considered. For all the pattern-classification problems considered, the HS-BtW training method outperforms IHS in terms of the overall recognition percent and the overall training time even if double HMS is used by HS-BtW.

The convergence graph given in Fig. 11, which is obtained from the Iris results, illustrates how the BtW value changes during the course of training. Each drop in the "Worst Fitness" curve represent accepting a new improvisation that replaces the current worst vector of HM while each drop in the "Best Fitness" curve represent finding a new best vector in HM. The SSE value would decrease eventually and the two curves become close to each other as convergence is approached, i.e. as BtW value approaches $BtW_{termination}$. Fig. 12 shows the effect of BtW ratio on PAR and B dynamic settings. The top graph is a replica of lower graph of Fig. 11 which is needed to show the effect of BtW on PAR and B. The lower graph is a two-vertical axis graph to simultaneously show PAR and B changes against the upper BtW ratio graph. PAR would increase or decrease linearly with BtW as introduced earlier in Fig. 9(a). B on the other hand is inversely proportional with BtW and would decrease or increase exponentially as given earlier in Fig. 9(b). Such settings enables the method to modify its probabilistic parameters based on the quality of solutions in HM. In comparison with the adapted IHS method, the changes are steady and bound to the current iteration count to determine PAR and B values. In HS-BtW whenever the BtW value increases PAR values tend to become closer to the PAR_{max} value and B becomes closer to the B_{min} value. In the adapted IHS such conditions occurs only as the current iteration count approaches MAXIMP. The values of PAR and B would approach PAR_{min} and B_{max} respectively as the BtW values decreases. The horizontal flat curve area in the lower graph of Fig. 14 correspond to the case when the BtW values goes below the initial $BtW_{threshold}$. In this case, PAR is set fixed at PAR_{min} as in equation (9), while B is set fixed at B_{max} as in equation (11). Theses dynamic settings of the probabilistic parameters of PAR and B would gave the method better capabilities over the adapted IHS in terms of improvising more accepted improvisations in less amount of for the benchmarking problems considered.

VII. CONCLUSIONS

By adapting and modifying an improved version of HS, namely IHS, two new FFANN supervised training methods are proposed for pattern-classification applications; the adapted IHS and modified adapted IHS referred to as HS-BtW. The proposed IHS-based training methods has showed superiority in comparison with both a GA-based method and a trajectory-driven method using the same data sets of pattern-classification benchmarking problems. The settings of the probabilistic values in the adapted IHS training method are functions of the current iteration count. The termination condition is bound by a subjective maximum iteration count value MAXIMP set prior to starting the training process. Choosing a high value might cause the method to suffer from overtraining in some problems while choosing a smaller value might prevent the algorithm from finding a better solution. Increasing HMS seems to have no effect on the adapted IHS solutions for the pattern-classification problems considered for this work.

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

The HS-BtW method utilizes the BtW ratio to determine its termination condition as well as to dynamically set the probabilistic parameter values during the course of training. Such settings are independent of the current iteration count and have resulted in generating more accepted improvisations in less amount of overall training time in comparison with the adapted IHS. Doubling the HMS have resulted in attaining better solutions for some of the pattern-classification problems considered with an overall training time that is still less in comparison with other rival methods. However, BP is still superior in terms of attaining better overall recognition percentage in pattern-classification problems having relatively larger training data sets. BP seems to benefit from such sets to better fine-tune the FFANN weight values attributed to its trajectory-driven approach.

For future work it would be also interesting to apply the proposed HS-BtW technique to optimization problems other than ANNs such as some standard engineering optimization problems used in [15] or solving some global numerical optimization problems used in [30].

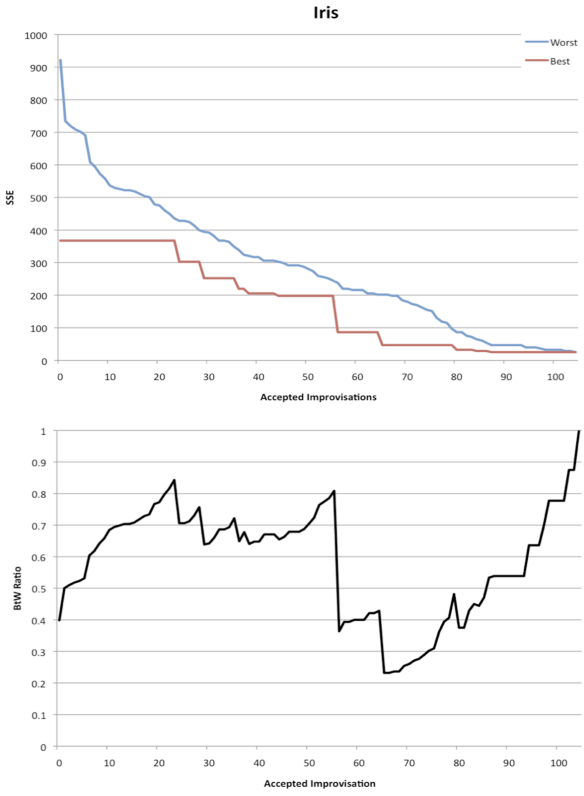

Figure 11. Convergence graph for the HS-BtW Iris problem

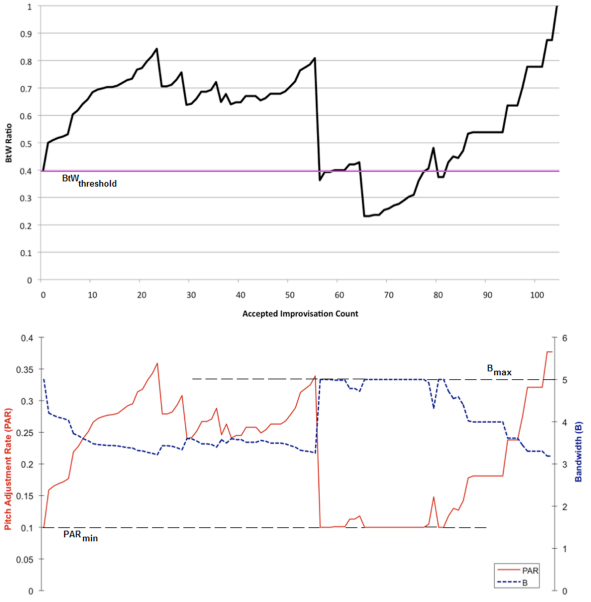

Figure 12. BtW value against PAR and B for the accepted improvisations of the HS-BtW Iris problem

ACKNOWLEDGMENT

The first author would like to thank Universiti Sains Malaysia for accepting as a postdoctoral fellow in the School of Computer Sciences. This work was funded by the Fundamental Research Grant Scheme from "Jabatan Pengajian Tinggi Kementerian Pengajian Tinggi" (Project Account number 203/PKOMP/6711136) awarded to the second author.

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

TABLE 3. RESULTS FOR BEST OUT OF TEN TRAINING SESSIONS FOR THE IRIS PROBLEM

M	Training							Testing	
	HMS/ Pop. Size	MAXIMP	SSE	Total Accepted	Last Accepted Iteration #	Last Accepted Time	Overall Time h:mm:ss	Overall Recog.%	Class Recog.%
IHS	10	5000	16	154	1826	0:00:58	0:02:39	96.67%	100.00% 100.00% 90.00%
	10	20000	7.08	287	10255	0:05:32	0:10:45	93.33%	100.00% 100.00% 80.00%
HS-BtW	10	20000	25.19	104	208	0:00:27	**0:00:27**	**100.00%**	100.00% 100.00% 100.00%
BP	N.A.	N.A.	7.85	1254	N.A.	N.A.	0:07:29	96.67%	100.00% 100.00% 90.00%
GANNT	10	N.A.	96	66	N.A.	N.A.	0:00:34	90.00%	100.00% 90.00% 80.00%

TABLE 4. RESULTS FOR BEST OUT OF TEN TRAINING SESSIONS FOR THE MAGIC PROBLEM

M	Training							Testing	
	HMS/ Pop. Size	MAXIMP	SSE	Total Accepted	Last Accepted Iteration #	Last Accepted Time	Overall Time h:mm:ss	Overall Recog.%	Class Recog.%
IHS	10	5000	12387.95	172	4574	1:49:43	1:59:13	77.39%	94.57% 45.74%
		20000	10647.98	413	19834	7:34:40	7:38:27	81.18%	93.27% 58.89%
HS-BtW	10	20000	11463.36	114	395	0:32:10	0:32:10	79.65%	86.62% 66.82%
	20	20000	9944.15	495	3190	4:10:01	4:10:01	81.44%	93.84% 58.59%
BP	N.A.	N.A.	6137.48	825	N.A.	N.A.	**4:35:42**	**83.97%**	82.97% 85.65%
GANNT	10	N.A.	12473.48	149	N.A.	N.A.	0:48:18	77.87%	89.62% 56.20%

TABLE 5. RESULTS FOR BEST OUT OF TEN TRAINING SESSIONS FOR THE DIABETES PROBLEM

M	Training							Testing	
	HMS/ Pop.Size	MAXIMP	SSE	Total Accepted	Last Accepted Iteration #	Last Accepted Time	Overall Time h:mm:ss	Overall Recog.%	Class Recog.%
IHS	10	5000	968	147	4835	0:10:48	0:11:10	76.62%	90.00% 51.85%
	10	20000	856	240	13001	0:27:11	0:41:47	77.27%	89.00% 55.56%
HS-BtW	10	20000	915.88	223	1316	0:11:42	**0:11:42**	**79.87%**	87.00% 66.67%
BP	N.A	N.A.	408.61	11776	N.A.	N.A.	5:30:42	78.57%	88.00% 61.11%
GANNT	10	N.A.	1108	1007	N.A.	N.A.	0:29:28	79.87%	89.00% 62.96%

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

TABLE 6. RESULTS FOR BEST OUT OF TEN TRAINING SESSIONS FOR THE CANCER PROBLEM

M	Training							Testing	
	HMS/ Pop.Size	MAXIMP	SSE	Total Accepted	Last Accepted Iteration #	Last Accepted Time	Overall Time h:mm:ss	Overall Recog.%	Class Recog.%
IHS	10	5000	124	155	4946	0:10:13	0:10:19	100.00%	100.00% 100.00%
	10	20000	99.76	212	19914	0:30:04	0:30:11	99.29%	100.00% 97.92%
HS-BtW	10	20000	126.37	217	1408	0:08:30	**0:08:30**	**100.00%**	100.00% 100.00%
BP	N.A.	N.A.	24.62	1077	N.A.	N.A.	0:27:55	95.71%	100.00% 87.50%
GANNT	10	N.A.	172	452	N.A.	N.A.	0:10:30	98.57%	100.00% 95.83%

TABLE 7. RESULTS FOR BEST OUT OF TEN TRAINING SESSIONS FOR THE IONOSPHERE PROBLEM

M	Training							Testing	
	HMS/ Pop.Size	MAXIMP	SSE	Total Accepted	Last Accepted Iteration #	Last Accepted Time	Overall Time h:mm:ss	Overall Recog.%	Class Recog.%
IHS	10	5000	72	181	4711	0:03:45	0:03:58	94.37%	100.00% 84.00%
	10	20000	64	225	19867	0:20:51	0:21:00	95.77%	97.83% 92.00%
HS-BtW	10	20000	113.6	327	1770	0:05:44	0:05:44	94.37%	100.00% 84.00%
	20	20000	70.23	584	7254	0:20:33	**0:20:33**	**97.18%**	100.00% 92.00%
BP	N.A.	N.A.	8.52	1628	N.A.	N.A.	0:24:43	95.77%	100.00% 88.00%
GANNT	10	N.A.	152	2244	N.A.	N.A.	0:35:57	94.37%	100.00% 84.00%

TABLE 8. IHS BEST TRAINING RESULTS VS. HS-BTW BEST TRAINING RESULTS

Problem	IHS Training			HS-BtW Training		
	HMS	Overall Time h:mm:ss	Overall Recog.%	HMS	Overall Time h:mm:ss	Overall Recog.%
Iris	10	0:02:39	96.67%	10	0:00:27	100.00%
Magic	10	7:38:27	81.18%	20	4:10:01	81.44%
Diabetes	10	0:41:47	77.27%	10	0:11:42	79.87%
Cancer	10	0:10:19	100.00%	10	0:08:30	100.00%
Ionosphere	10	0:21:00	95.77%	20	0:20:33	97.18%

REFERENCES

[1] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A New Heuristic Optimization Algorithm: Harmony Search", Simulation, vol. 72, pp. 60-68, 2001.

[2] Z. W. Geem, K. S. Lee, and Y. Park, "Applications of harmony search to vehicle routing", American Journal of Applied Sciences, vol. 2, pp. 1552-1557, 2005.

[3] Z. W. Geem, C.-L. Tseng, and Y. Park, "Harmony Search for Generalized Orienteering Problem: Best Touring in China," in Advances in Natural Computation. vol. 3612/2005: Springer Berlin / Heidelberg, 2005, pp. 741-750.

[4] Z. W. Geem, K. S. Lee, and C. L. Tseng, "Harmony search for structural design", in Genetic and Evolutionary Computation Conference (GECCO 2005), Washington DC, USA, 2005, pp. 651-652.

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

[5] R. Forsati, A. T. Haghighat, and M. Mahdavi, "Harmony search based algorithms for bandwidth-delay-constrained least-cost multicast routing", Computer Communications, vol. 31, pp. 2505-2519, 2008.

[6] R. Forsati, M. Mahdavi, M. Kangavari, and B. Safarkhani, "Web page clustering using Harmony Search optimization", in Canadian Conference on Electrical and Computer Engineering (CCECE 2008) Ontario, Canada: IEEE Canada, 2008, pp. 001601 – 001604.

[7] Z. W. Geem, "Harmony Search Applications in Industry," in Soft Computing Applications in Industry. vol. 226/2008: Springer Berlin / Heidelberg, 2008, pp. 117-134.

[8] W. S. Jang, H. I. Kang, and B. H. Lee, "Hybrid Simplex-Harmony search method for optimization problems", in IEEE Congress on Evolutionary Computation (CEC 2008) Trondheim, Norway: IEEE, 2008, pp. 4157-4164.

[9] H. Ceylan, H. Ceylan, S. Haldenbilen, and O. Baskan, "Transport energy modeling with meta-heuristic harmony search algorithm, an application to Turkey", Energy Policy, vol. 36, pp. 2527-2535, 2008.

[10] J.-H. Lee and Y.-S. Yoon, "Modified Harmony Search Algorithm and Neural Networks for Concrete Mix Proportion Design", Journal of Computing in Civil Engineering, vol. 23, pp. 57-61, 2009.

[11] P. Tangpattanakul and P. Artrit, "Minimum-time trajectory of robot manipulator using Harmony Search algorithm", in 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2009) vol. 01 Pattaya, Thailand: IEEE, 2009, pp. 354-357.

[12] Z. W. Geem, "Novel Derivative of Harmony Search Algorithm for Discrete Design Variables", Applied Mathematics and Computation, vol. 199, pp. 223-230, 2008.

[13] A. Mukhopadhyay, A. Roy, S. Das, S. Das, and A. Abraham, "Population-variance and explorative power of Harmony Search: An analysis", in Third International Conference on Digital Information Management (ICDIM 2008) London, UK: IEEE, 2008, pp. 775-781.

[14] Q.-K. Pan, P. N. Suganthan, M. F. Tasgetiren, and J. J. Liang, "A self-adaptive global best harmony search algorithm for continuous optimization problems", Applied Mathematics and Computation, vol. 216, pp. 830-848, 2010.

[15] M. Mahdavi, M. Fesanghary, and E. Damangir, "An Improved Harmony Search Algorithm for Solving Optimization Problems", Applied Mathematics and Computation, vol. 188, pp. 1567-1579, 2007.

[16] M. G. H. Omran and M. Mahdavi, "Globel-Best Harmony Search", Applied Mathematics and Computation, vol. 198, pp. 643-656, 2008.

[17] D. Zou, L. Gao, S. Li, and J. Wu, "Solving 0–1 knapsack problem by a novel global harmony search algorithm ", Applied Soft Computing, vol. 11, pp. 1556-1564, 2011.

[18] R. S. Sexton and R. E. Dorsey, "Reliable classification using neural networks: a genetic algorithm and backpropagation comparison", Decision Support Systems, vol. 30, pp. 11-22, 15 December 2000.

[19] K. P. Ferentinos, "Biological engineering applications of feedforward neural networks designed and parameterized by genetic algorithms", Neural Networks, vol. 18, pp. 934-950, 2005.

[20] R. E. Dorsey, J. D. Johnson, and W. J. Mayer, "A Genetic Algoirthm for the Training of Feedforward Neural Networks", Advances in A.pngicial Intelligence in Economics, Finance, and Management vol. 1, pp. 93-111, 1994.

[21] J. Zhou, Z. Duan, Y. Li, J. Deng, and D. Yu, "PSO-based neural network optimization and its utilization in a boring machine", Journal of Materials Processing Technology, vol. 178, pp. 19-23, 2006.

[22] M. Geethanjali, S. M. R. Slochanal, and R. Bhavani, "PSO trained ANN-based differential protection scheme for power transformers", Neurocomputing, vol. 71, pp. 904-918, 2008.

[23] A. Rakitianskaia and A. P. Engelbrecht, "Training Neural Networks with PSO in Dynamic Environments", in IEEE Congress on Evolutionary Computation (CEC '09) Trondheim, Norway: IEEE, 2009, pp. 667-673.

[24] H. Shi and W. Li, "Artificial neural networks with ant colony optimization for assessing performance of residential buildings", in International Conference on Future BioMedical Information Engineering (FBIE 2009): IEEE, 2009, pp. 379-382.

[25] C. Blum and K. Socha, "Training feed-forward neural networks with ant colony optimization: an application to pattern classification", in Fifth International Conference on Hybrid Intelligent Systems (HIS '05) Rio de Janeiro, Brazil, 2005, p. 6.

[26] Z. W. Geem, C.-L. Tseng, J. Kim, and C. Bae, "Trenchless Water Pipe Condition Assessment Using Artificial Neural Network", in Pipelines 2007, Boston, Massachusetts, 2007, pp. 1-9.

[27] A. Kattan, R. Abdullah, and R. A. Salam, "Harmony Search Based Supervised Training of Artificial Neural Networks", in International Conference on Intelligent Systems, Modeling and Simulation (ISMS2010), Liverpool, England, 2010, pp. 105-110.

[28] S. Kulluk, L. Ozbakir, and A. Baykasoglu, "Self-adaptive global best harmony search algorithm for training neural networks", Procedia Computer Science, vol. 3, pp. 282-286, 2011.

[29] N. P. Padhy, Artificial Intelligence and Intelligent Systems, 1st ed. Delhi: Oxford University Press, 2005.

[30] J.-T. Tsai, J.-H. Chou, and T.-K. Liu, "Tuning the Strucutre and Parameters of a Neural Network by Using Hybrid Taguchi-Genetic Algorithm", IEEE Transactions on Neural Networks, vol. 17, January 2006.

[31] W. Gao, "Evolutionary Neural Network Based on New Ant Colony Algorithm", in International Symposium on Computational Intelligence and Design (ISCID '08). vol. 1 Wuhan, China, 2008, pp. 318 - 321.

[32] S. Kiranyaz, T. Ince, A. Yildirim, and M. Gabbouj, "Evolutionary artificial neural networks by multi-dimensional particle swarm optimization", Neural Networks, vol. 22, pp. 1448-1462, 2009.

[33] C. M. Bishop, Pattern Recognition and Feed-forward Networks: MIT Press, 1999.

[34] X. Jiang and A. H. K. S. Wah, "Constructing and training feed-forwardneural networks for pattern classifcation", Pattern Recognition, vol. 36, pp. 853-867, 2003.

[35] F. Marini, A. L. Magri, and R. Bucci, "Multilayer feed-forward artificial neural networks for class modeling", Chemometrics and intelligent laboratory systems, vol. 88, pp. 118-124, 2007.

[36] T. Kathirvalavakumar and P. Thangavel, "A Modified Backpropagation Training Algorithm for Feedforward Neural Networks", Neural Processing Letters, vol. 23, pp. 111-119, 2006.

[37] K. M. Lane and R. D. Neidinger, "Neural networks from idea to implementation", ACM Sigapl APL Quote Quad, vol. 25, pp. 27-37, 1995.

[38] E. Fiesler and J. Fulcher, "Neural network classification and formalization", Computer Standards & Interfaces, vol. 16, pp. 231-239, July 1994.

[39] L. Fausett, Fundamentals of Neural Networks Architectures, Algorithms, and Applications. New Jersey: Prentice Hall, 1994.

[40] I.-S. Oh and C. Y. Suen, "A class-modular feedforward neural network for handwriting recognition", Pattern Recognition, vol. 35, pp. 229-244, 2002.

[41] A. T. Chronopoulos and J. Sarangapani, "A distributed discrete-time neural network architecture for pattern allocation and control", in Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'02), Florida, USA, 2002, pp. 204-211.

[42] Z. W. Geem and W. E. Roper, "Energy demand estimation of South Korea using artificial neural networks", Energy Policy, vol. 37, pp. 4049-4054, 2009.

[43] M. H. Hassoun, Fundamentals of Artificial Neural Networks. Massachusetts: MIT Press, Cambridge, 1995.

[44] D. Kim, H. Kim, and D. Chung, "A Modified Genetic Algorithm for Fast Training Neural Networks," in Advances in Neural Networks - ISNN 2005. vol. 3496/2005: Springer Berlin / Heidelberg, 2005, pp. 660-665.

[45] M. b. Nasr and M. Chtourou, "A Hybrid Training Algorithm for Feedforward Neural Networks ", Neural Processing Letters, vol. 24, pp. 107-117, 2006.

[46] J. N. D. Gupta and R. S. Sexton, "Comparing backpropagation with a genetic algorithm for neural network training", Omega, The International Journal of Management Science, vol. 27, pp. 679-684, 1999.

[47] B. Guijarro-Berdinas, O. Fontenla-Romero, B. Perez-Sanchez, and A. Alonso-Betanzos, "A New Initialization Method for Neural Networks Using Sensitivity Analysis", in International Conference on Mathematical and Statistical Modeling Ciudad Real, Spain, 2006, pp. 1-9.

[48] J. Škutova, "Weights Initialization Methods for MLP Neural Networks", Transactions of the VŠB, vol. LIV, article No. 1636, pp. 147-152, 2008.

[49] G. Wei, "Study on Evolutionary Neural Network Based on Ant Colony Optimization", in International Conference on Computational Intelligence and Security Workshops Harbin, Heilongjiang, China, 2007, pp. 3-6.

[50] Y. Zhang and L. Wu, "Weights Optimization of Neural Networks via Improved BCO Approach", Progress In Electromagnetics Research, vol. 83, pp. 185-198, 2008.

[51] J. Yu, S. Wang, and L. Xi, "Evolving artificial neural networks using an improved PSO and DPSO", Neurocomputing, vol. 71, pp. 1054-1060, 2008.

[52] M. N. H. Siddique and M. O. Tokhi, "Training neural networks: backpropagation vs. genetic algorithms", in International Joint Conference on Neural Networks (IJCNN '01), Washington, DC 2001, pp. 2673 - 2678.

[53] K. E. Fish, J. D. Johnson, R. E. Dorsey, and J. G. Blodgett, "Using an Artificial Neural Network Trained with a Genetic Algorithm to Model Brand Share ", Journal of Business Research, vol. 57, pp. 79-85, January 2004 2004.

[54] E. Alba and J. F. Chicano, "Training Neural Networks with GA Hybrid Algorithms," in Genetic and Evolutionary Computation (GECCO 2004). vol. 3102/2004: Springer Berlin / Heidelberg, 2004, pp. 852-863.

[55] L. G. C. Hamey, "XOR Has No Local Minima: A Case Study in Neural Network Error Surface Analysis", Neural Networks, vol. 11, pp. 669-681, 1998.

[56] R. Cutchin, C. Douse, H. Fielder, M. Gent, A. Perlmutter, R. Riley, M. Ross, and T. Skinner, The Definitive Guitar Handbook, 1st ed.: Flame Tree Publishing, 2008.

[57] K. S. Lee and Z. W. Geem, "A New Meta-heuristic Algorithm for Continuous Engineering Optimization: Harmony Search Theory and Practice", Computer Methods in Applied Mechanics and Engineering, vol. 194, pp. 3902-3933, 2005.

[58] Z. W. Geem, "Optimal Cost Design of Water Distribution Networks Using Harmony Search", Engineering Optimization, vol. 38, pp. 259-277, 2006.

[59] Z. W. Geem and J.-Y. Choi, "Music Composition Using Harmony Search Algorithm," in Applications of Evolutionary Computing. vol. 4448/2007: Springer Berlin / Heidelberg, 2007, pp. 593-600.

[60] R. S. Sexton, R. E. Dorsey, and N. A. Sikander, "Simultaneous Optimization of Neural Network Function and Architecture Algorithm", Decision Support Systems, vol. 30, pp. 11-22, December 2004 2004.

AUTHORS PROFILE

Ali Kattan, (Ph.D.): Dr. Kattan is a postdoctoral fellow at the School of Computer Sciences - Universiti Sains Malaysia. He completed his Ph.D. from the same school in 2010. He has a blended experience in research and industry. Previously, he served as an assigned lecturer at the Hashemite University in Jordan and as a senior developer working for InterPro Global Partners, an e-Business solution provider in the United States. He specializes in Artificial Neural Networks and Parallel & Distributed Processing. His current research interests include optimization techniques, parallel processing using GPGPU, Cloud Computing and the development of smart phone application. Dr. Kattan is an IEEE member since 2009 and a peer-reviewer in a number of scientific journals in the field

Rosni Abdullah (Ph.D.): Prof. Dr. Rosni Abdullah is a professor in parallel computing and one of Malaysia's national pioneers in the said domain. She was appointed Dean of the School of Computer Sciences at Universiti Sains Malaysia (USM) in June 2004, after having served as its Deputy Dean (Research) since 1999. She is also the Head of the Parallel and Distributed Processing Research Group at the School since its inception in 1994. Her main interest lies in the data representation and the associated algorithms to organize, manage and analyse biological data which is ever increasing in size. Particular interest is in the development of parallel algorithms to analyse the biological data using Message Passing Interface (MPI) on message passing architectures and multithreading on multicore architectures. Her latest research interests include Cloud Computing, GPGPU and Computational Neuroscience.

http://sites.google.com/site/ijcsis/
ISSN 1947-5500