Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IMPACT FACTOR: 6.017

IJCSMC, Vol. 8, Issue. 3, March 2019, pg.36 – 41

DIABETES DIAGNOSIS USING MACHINE LEARNING

Alaa Badr Eysa; Asst Prof. Dr. Sefer Kurnaz
Electrical and Computer Engineering
Electrical and Computer Engineering
alaabadr428@gmail.com; sefer.kurnaz@altinbas.edu.tr

Abstract— Artificial neural networks have been in the position of producing complex dynamics in control applications over the last decade, especially when they are linked to feedback. Although ANNs are strong for network design, the harder the design of the network, the more complex the desired dynamic is. Many researchers tried to automate the design process of ANN using computer programs. Search and optimization problems can be considered as the problem of finding the best parameter set for a network to solve a problem. Recently, the problem of optimizing ANN parameters to train different research datasets has been targeted by two commonly used stochastic genetic algorithms (GA) and particle swarm optimization (PSO). The process based on the neural network is optimized with GA and PSO to enable the robot to perform complex tasks. However, using such optimization algorithms to optimize the ANN training process cannot always be balanced or successful. These algorithms simultaneously aim to develop three main components of an ANN: synaptic weight, connections, architecture and transfer functions set for each neuron. Developed with the proposed approach, ANN is also compared with hand-designed Levenberg-Marquardt and Back Propagation algorithms.

Keywords— DIABETES; DIAGNOSIS; MACHINE LEARNING; Artificial neural networks; ANN

I. INTRODUCTION

Artificial Neural Networks (ANNs) are organized neurons connected to each other to form networks that contains input hidden and output layers which the connections are considered as synaptic weights. An ANN is a very powerful technique for pattern recognition classification prediction … etc. ANN based on two main processes the first one is the learning process and the second is testing process. In learning process the synaptic weights are continuously changes their values until the knowledge of the network is accurate enough or learning process ends. When learning process ends the testing process starts which in this process accuracy of the network will be collected to check how good the training was. Different algorithms were proposed for ANN training in last years which weren't that good in maintaining the required solution and getting the optimum solution. For this reason bioinspired algorithms were used to train ANNs.

Bioinspired algorithms due to their optimization power they gathered a lot of acceptance in the Artificial Intelligence community. ANN is a fundamental form of learning through various works using evolutionary and bioinhalated algorithms. Neural education methods based on local search population and other methods, such as coevolutionary co-operative models. In the ANN training process, the bioinspired algorithm optimizes the synaptic weights of the ANN until the ANN precision and mistake exceeds the necessary accuracy or when the testing begins. The network accuracy will be collected during the testing process to verify the quality of the training.

II. PARTICLE SWARM OPTIMIZATION

Eberhart et al. [16] proposed a bioinspired optimization algorithm called Particle Swarm Optimization (PSO) which this algorithm was inspired from bird flocks behavior movement when they search for food or when they're in survival situation same thing applies with fish schooling which the best member movement is the one that will going to be collected. The population represents their positions in the optimization. Saving the best position according to the best fitness value in the optimization is the particle evaluation in the optimization. The particles in each iteration in PSO change their position according to their velocity, which continues until they reach an optimal position in their search space. The particle speed I is updated with: At each time:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1\big(p_i(t) - x_i(t)\big) + c_2 r_2\left(p_g(t) - x_i(t)\right)$$

The random numbers r 1 and r 2 are distributed consistently between (0, 1) and r 2. Inertia c 1 and c 2 are an acceleration factor. The v1 speed is confined to the maximum residual v min range. This allows the speed update to look for its best individual location and calculate the world's best position.

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

III. ARTIFICIAL NEURAL NETWORK-PSO

The transfer function, architecture and synaptic weights and bias are the main key to have the best ANN with higher accuracy. Selecting the best individual for best ANN is based on the solutions that are gathered by the optimization according to fitness function measurement.

For each algorithm, this involves a large behavioral study. An additional concern is also the maximum number of the neurons (MNNs) directly linked to the unique dimension used in the ANN methodology. Due to information on the size of a given problem, a scheme for MNN in an ANN design was not necessary but only in line with intake and output patterns. In the individual section, this equation is explained.

IV. INDIVIDUAL

When the matrix information is obtained from individuals or possible solutions, it should be decoded in an NNA for assessment. In terms of synaptic weights and transmission, the first element to be decoded is the topology stored on the matrix. Therefore, it was proposed that certain rules should prevent the provision of repetitive links to ANN, the sole limit of ANN. This investigation is restricted to one feed ANN type. We shall be studying the behavior of such ANNs in future work, including recurring connections. Only three layers shall produce the architectures generated using the proposed method. The following three rules must be met in order to generate valid architectures. The ILNbe consists of the first set of HLN - input neurons, the neurons J which compose the hidden layer and the output layer K neurons.

(1) ILNi I = 1 is used for input neurons (ILN). I can only send HLNj and OLNk information.

(2) HLNj j = 1 For neurons of the hidden layer (HLN). J neuron can only send information with one last restriction to OLNk and HLNj. There is only a link to HLNj+1 for HLNj. NLCY.

(3) The OLNk k = 1 for the Neuron output layer (OLN). K neuron may only send data to other layer neurons, but there is only a connection to OLNk+1 with the limitation for OLNk. The North African Republic. OLNK.

The information in Wij with I= 1 to decode the architecture with these rules in mind. Based on binary square Z matrix, MNN and j= 1 (that is in the decimal base). This matrix represents a graph where the links between neuron I and j when zij= 1. are indicated by each component zij. The binary code shall be construed by transforming the binary code into a binary code "0111000" or "011001" between the ith neuron and seven neurons (number of bits). The neurotic connections (left to right) are only two, four, and seven. Architecture assesses the corresponding Wij component synaptic weights to I=0. The MNN= 2 and J. J. More NEM 1. NRM. NRM. For partiality, the component Wij has I= 1 encoded. J= MNN+ 2. MNN and j.

V. TRANSFER FUNCTIONS

The Wij component with I = 1 represents the TF. MNN + j = MNN + 3, respectively. The transfer features are within [0, 5]. For this research the transfer functions with their labels are sigmoid (LS) hyperbolic tangent (HT) sinusoidal (SN) Gaussian (GS) linear (LN)) and hardened limit (hL) functions.

VI. ARTIFICIAL NEURAL NETWORK OUTPUT

When the individual's data has been decoded, its effectiveness needs to be known to be evaluated for any fitness function. For this purpose, during training and generalization the ANN output needs to be calculated. This output is computed using algorithm 2, where I am the neuron output, i.e. the input module feeding the ANN n; m is the sizing of the desired input module, yi is the ANN output.

VII. FITNESS FUNCTION

MSE is the error of the ANN output and the pattern of desire. The minimum MSE generator (see the following equation) is the best person here:

$$F_1 = MSE = \frac{1}{p.M} \sum_{\xi=0}^{p} \sum_{i=0}^{M} (d_i^{\xi} - y_i^{\xi})$$

Where yi is the ANN's output.

VIII. IMPLEMENTATION AND RESULTS

The selection of these instances from a larger database has been subject to several constraints. In particular, all patients are women of Pima's Indian heritage aged at least 21 years. The number of instances and attributes are 768 and 8 respectively. All attributes are numeric-valued. Diazotic blood pressure attributes (mm Hg Triceps Skin Fold thickness (mm) 2-hour serum insulin (muU / ml). The diabetes pedigreed function Age(years) and the class variable(0) were used in this studying. Diastolic blood pressure index (mm Hg Triceps Skin Fold thickness (mm). Diagnosis of the body's glycoside is based on the diastolic gynecology. In this dataset there are Missing Attribute Values. If the class value 1 is interpreted as "tested positive for diabetes". Number of instances for class 0 and 1 are 500 and 268 respectively.

This experiment proves the performance of the PSO against BP algorithms the construction of BP and PSO algorithms consists mainly of 8 inputs one hidden layer of 8 neurons each and one output layer. Where in this study sigmoid function have been is used.

Table 2 shows that the classification accuracy of PSO algorithm are much better than BP algorithm has obtained that accuracy with only 250 iterations while BP needed 300 iterations. Furthermore comparing the MSE of both algorithms shows that the MSE of PSO is much less than BP algorithm. It also converges faster than the BP algorithms as shown in Fig. 2.

Table 1. Diabetes dataset

Dataset	**instances**	**Attributes**	**attribute types**	**Classes**
UCI-Diabetes	768	8	Numeric	2

Table 1 shows that diabetes dataset consist of 768 instances 8 attributes and 2 classes. The attributes are the number of pregnancies the patient has had their BMI insulin level age Glucose Blood Pressure Diabetes Pedigree Function and Skin Thickness. The classes that the data holds are either effected or not.

Table 2. Performances comparison of BP and PSO for diabetes dataset

Methods	Accuracy	Attributes	iteration	Error
BP	77.83	8	Numeric	2
PSO	88.26			

Figure 1, 2 and 3 are showing the fitness or MSE or error outcomes by using the existing ANN-PSO approach. Similarly, the figures 5.4 to 5.6 are showing the error results using same dataset for ANN-BP method.

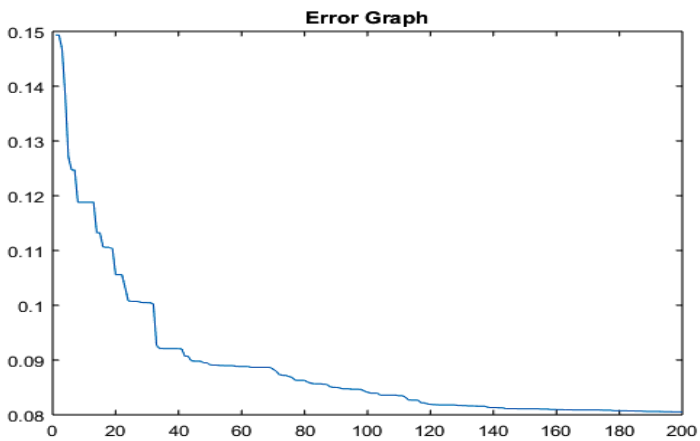

Fig 1: Error graph performance using ANN-PSO for Diabetes dataset

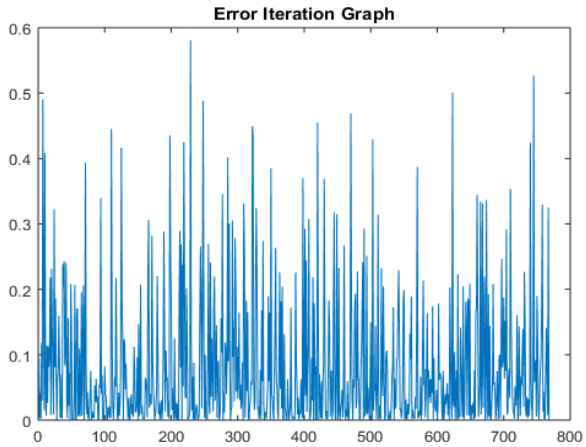

Fig 2: Error iteration graph performance using ANN-PSO for Diabetes dataset

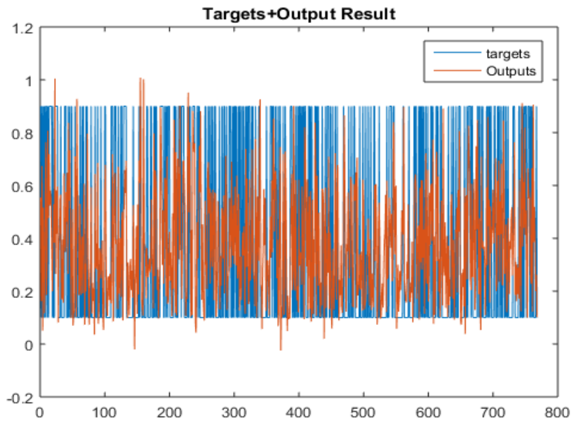

Fig 3: Error iteration graph performance using ANN-PSO for Diabetes dataset for original target and predicted outcomes.

IX. CONCLUSIONS

We have proposed three rules for connecting feed-forward ANN to the links between neurons. We found that the input neurons have not been connected to some ANNs developed with the methodology proposed. This means that ANN output was not computed with the function associated with this neuron. This is called input pattern dimensional reduction. Eight transmission characteristics are evaluated for each individual including combining MSE, CER validation failure and architectural reduction (connections and neurons). These experiments show that those using the CER classification error have the fitness functions that produced the highest weighted ANN recognition rate. The three bio-inspired algorithms based on PSOs were comparable to the weighted average recovery rate. The best results were obtained, in contrast to the core PSO and SGPSO algorithms. To statistically verify the exactness of the methodology proposed, parameters of the three bio-inspired algorithms were chosen. VCER with variable ranges from [−2, 2]. After we have tuned and selected the best parameter setting for each algorithm the parameters are different and set to 0.3, c1=1.0 and c2=1.5 for each parameter from those proposed in the literature. The best fitness feature of the SGPSO algorithm was CER with a variable range from [−2, 2]. The value of c3= 0.5 and geometric center P= 100 was defined. CER with a variable range of [−4, 4] was The NMPSO algorithm's best fitness function. The μ= 200 crossover rateα= 0.1 and β mutation rate= 0.1 was the best settings for this. Three runs have been performed on each of the 10 classification issues following the adaptation of 3 algorithm parameters. For each algorithm, the transmission functions were most frequently used: for the basic PSO algorithm, Gaussian for the SGPSO algorithm, and NMPSO for the Gaussian.

REFERENCES

[1] "A New Associative Model with Dynamical Synapses | SpringerLink." [Online]. Available: https://link.springer.com/article/10.1007/s11063-008-9089-6. [Accessed: 04-Mar-2019].
[2] "A new evolutionary system for evolving artificial neural networks - IEEE Journals & Magazine." [Online]. Available: https://ieeexplore.ieee.org/document/572107. [Accessed: 04-Mar-2019].
[3] "An Improved Particle Swarm Optimization for Evolving Feedforward Artificial Neural Networks | SpringerLink." [Online]. Available: https://link.springer.com/article/10.1007/s11063-007-9053-x. [Accessed: 04-Mar-2019].
[4] "An improved PSO-based ANN with simulated annealing technique - ScienceDirect." [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231204003364. [Accessed: 04-Mar-2019].
[5] "An Introduction to Neural Networks | The MIT Press." [Online]. Available: https://mitpress.mit.edu/books/introduction-neural-networks. [Accessed: 04-Mar-2019].
[6] "Artificial neural network synthesis by means of artificial bee colony (ABC) algorithm - IEEE Conference Publication." [Online]. Available: https://ieeexplore.ieee.org/document/5949637. [Accessed: 04-Mar-2019].
[7] "Back-Propagation vs Particle Swarm Optimization Algorithm: which Algorithm is better to adjust the Synaptic Weights of a Feed-Forward ANN? | Garro | International Journal of Artificial IntelligenceTM." [Online]. Available: http://www.ceser.in/ceserp/index.php/ijai/article/view/2260. [Accessed: 04-Mar-2019].
[8] "Backpropagation through time: what it does and how to do it - IEEE Journals & Magazine." [Online]. Available: https://ieeexplore.ieee.org/document/58337. [Accessed: 04-Mar-2019].
[9] "Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks - IEEE Conference Publication." [Online]. Available: https://ieeexplore.ieee.org/document/1202255. [Accessed: 04-Mar-2019].
[10] "Design of artificial neural networks using a modified Particle Swarm Optimization algorithm - IEEE Conference Publication." [Online]. Available: https://ieeexplore.ieee.org/document/5178918. [Accessed: 04-Mar-2019].
[11] "Designing artificial neural networks using differential evolution for classifying DNA microarrays - IEEE Conference Publication." [Online]. Available: https://ieeexplore.ieee.org/document/7969644. [Accessed: 04-Mar-2019].
[12] "Designing Artificial Neural Networks Using Particle Swarm Optimization Algorithms. - PubMed - NCBI." [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/26221132. [Accessed: 04-Mar-2019].
[13] "Earthquake classifying neural networks trained with random dynamic neighborhood PSOs." [Online]. Available: https://dl.acm.org/citation.cfm?id=1276958.1276974. [Accessed: 04-Mar-2019].
[14] "Empirical study of particle swarm optimization - IEEE Conference Publication." [Online]. Available: https://ieeexplore.ieee.org/document/785511. [Accessed: 04-Mar-2019].
[15] "Evolving artificial neural networks - IEEE Journals & Magazine." [Online]. Available: https://ieeexplore.ieee.org/document/784219. [Accessed: 04-Mar-2019].
[16] "Evolving Graphs for ANN Development and Simplification." [Online]. Available: https://www.researchgate.net/publication/314457474_Evolving_Graphs_for_ANN_Development_and_Simplification. [Accessed: 04-Mar-2019].
[17] "Evolving Neural Networks: A Comparison between Differential Evolution and Particle Swarm Optimization | SpringerLink." [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-21515-5_53. [Accessed: 04-Mar-2019].
[18] "Hybrid Particle Swarm Optimiser with breeding and subpopulations - Semantic Scholar." [Online]. Available: https://www.semanticscholar.org/paper/Hybrid-Particle-Swarm-Optimiser-with-breeding-and-L%C3%B8vbjerg/1e4cddd9d4b89a9dbdf854aaf84b584ba47386ab. [Accessed: 04-Mar-2019].
[19] "Learning internal representations by error propagation." [Online]. Available: https://dl.acm.org/citation.cfm?id=104279.104293. [Accessed: 04-Mar-2019].
[20] "Metaheuristic Procedures for Training Neural Networks | Enrique Alba | Springer." [Online]. Available: https://www.springer.com/gp/book/9780387334158. [Accessed: 04-Mar-2019].
[21] "Neural Networks Training Based on Differential Evolution Algorithm Compared with Other Architectures for Weather Forecasting 34 - Semantic Scholar." [Online]. Available: https://www.semanticscholar.org/paper/Neural-Networks-Training-Based-on-Differential-with-Kader/9380d2591edd48155352ffb90fa44f076f8641e3. [Accessed: 04-Mar-2019].
[22] "Particle swarm optimization feedforward neural network for modeling runoff | SpringerLink." [Online]. Available: https://link.springer.com/article/10.1007/BF03326118. [Accessed: 04-Mar-2019].
[23] "Particle swarm optimization with Gaussian mutation - IEEE Conference Publication." [Online]. Available: https://ieeexplore.ieee.org/document/1202250. [Accessed: 04-Mar-2019].

[24] "Second Generation Particle Swarm Optimization - IEEE Conference Publication." [Online]. Available: https://ieeexplore.ieee.org/document/4630781. [Accessed: 04-Mar-2019].
[25] "Swarm Intelligence - 1st Edition." [Online]. Available: https://www.elsevier.com/books/swarm-intelligence/eberhart/978-1-55860-595-4. [Accessed: 04-Mar-2019].
[26] "Swarm Intelligence in Cellular Robotic Systems | SpringerLink." [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-58069-7_38. [Accessed: 04-Mar-2019].