Hybrid Neurocomputing for Breast Cancer Detection

Yuehui Chen and Ajith Abraham

School of Information Science and Engineering
Jinan University, Jinan 250022, P.R.China
yhchen@ujn.edu.cn
School of Computer Science and Engineering
Chung-Ang University, Seoul, Republic of Korea
ajith.abraham@ieee.org

Abstract. Breast cancer is one of the major tumor related cause of death in women. Various artificial intelligence techniques have been used to improve the diagnoses procedures and to aid the physician's efforts. In this paper we summarize our preliminary study to detect breast cancer using a Flexible Neural Tree (FNT), Neural Network (NN), Wavelet Neural Network (WNN) and their ensemble combination. For the FNT model, a tree-structure based evolutionary algorithm and the Particle Swarm Optimization (PSO) are used to find an optimal FNT. For the NN and WNN, the PSO is employed to optimize the free parameters. The performance of each approach is evaluated using the breast cancer data set. Simulation results show that the obtained FNT model has a fewer number of variables with reduced number of input features and without significant reduction in the detection accuracy. The overall accuracy could be improved by using an ensemble approach by a voting method.

1 Introduction

Breast cancer is the most common cancer in women in many countries. Various artificial intelligence techniques have been used to improve the diagnoses procedures and to aid the physician's efforts [1][2][3][4]. Screening mammography is the best tool available for detecting cancerous lesions before clinical symptoms appear [5].

In this paper we evaluate the performance of a Flexible Neural Tree (FNT), Neural Network (NN), Wavelet Neural Network (WNN) and an ensemble method to detect breast-cancer. For FNT model, a tree-structure based evolutionary algorithm and the Particle Swarm Optimization (PSO) are used to find an optimal FNT. For the NN and WNN, the PSO is employed to optimize the free parameters. Simulation studies shown the effectiveness of the proposed method.

2 The PSO Algorithm

The PSO [6] conducts searches using a population of particles which correspond to individuals in an evolutionary algorithm (EA). A population of particles is randomly generated initially. Each particle represents a potential solution and has a position represented by a position vector $\mathbf{x_i}$. A swarm of particles moves through the problem space, with the moving velocity of each particle represented by a velocity vector $\mathbf{v_i}$. At each time step, a function f_i representing a quality measure is calculated by using $\mathbf{x_i}$ as input. Each particle keeps track of its own best position, which is associated with the best fitness it has achieved so far in a vector $\mathbf{p_i}$. Furthermore, the best position among all the particles obtained so far in the population is kept track of as $\mathbf{p_g}$. In addition to this global version, another version of PSO keeps track of the best position among all the topological neighbors of a particle.

At each time step t, by using the individual best position, $\mathbf{p_i(t)}$, and the global best position, $\mathbf{p_g(t)}$, a new velocity for particle i is updated by

$$\mathbf{v_i(t+1)} = \mathbf{v_i(t)} + c_1\phi_1(\mathbf{p_i(t)} - \mathbf{x_i(t)}) + c_2\phi_2(\mathbf{p_g(t)} - \mathbf{x_i(t)}) \quad (1)$$

where c_1 and c_2 are positive constant and ϕ_1 and ϕ_2 are uniformly distributed random number in [0,1]. The term $\mathbf{v_i}$ is limited to the range of $\pm\mathbf{v_{max}}$. If the velocity violates this limit, it is set to its proper limit. Changing velocity this way enables the particle i to search around its individual best position, $\mathbf{p_i}$, and global best position, $\mathbf{p_g}$. Based on the updated velocities, each particle changes its position according to the following equation:

$$\mathbf{x_i(t+1)} = \mathbf{x_i(t)} + \mathbf{v_i(t+1)}. \quad (2)$$

In this research, the PSO is employed to optimize the parameter vectors of FNT, NN and WNN.

3 Breast Cancer Detection Using FNT, NN and WNN

3.1 Flexible Neural Tree Classifier

In this research, a tree-structural based encoding method with specific instruction set is selected for representing a FNT model [7][8].

Flexible Neuron Instructor and FNT Model The function set F and terminal instruction set T used for generating a FNT model are described as follows:

$$S = F\bigcup T = \{+_2, +_3, \ldots, +_N\}\bigcup\{x_1, \ldots, x_n\}, \quad (3)$$

where $+_i(i = 2, 3, \ldots, N)$ denote non-leaf nodes' instructions and taking i arguments. x_1,x_2,...,x_n are leaf nodes' instructions and taking no other arguments.

The output of a non-leaf node is calculated as a flexible neuron model (see Fig.1). From this point of view, the instruction $+_i$ is also called a flexible neuron operator with i inputs. In the creation process of neural tree, if a nonterminal instruction, i.e., $+_i(i = 2, 3, 4, \ldots, N)$ is selected, i real values are randomly generated and used for representing the connection strength between the node $+_i$ and its children. In addition, two adjustable parameters a_i and b_i are randomly created as flexible activation function parameters. Some examples of flexible activation functions are shown in Table 1.

For developing the FNT classifier, the following flexible activation function is used.

$$f(a_i, b_i, x) = e^{-(\frac{x-a_i}{b_i})^2} \tag{4}$$

The output of a flexible neuron $+_n$ can be calculated as follows. The total excitation of $+_n$ is

$$net_n = \sum_{j=1}^{n} w_j * x_j \tag{5}$$

where $x_j(j = 1, 2, \ldots, n)$ are the inputs to node $+_n$. The output of the node $+_n$ is then calculated by

$$out_n = f(a_n, b_n, net_n) = e^{-(\frac{netn-an}{bn})^2}. \tag{6}$$

A typical flexible neuron operator and a neural tree model are illustrated in Figure 1. The overall output of flexible neural tree can be computed from left to right by the depth-first method, recursively.

The Optimization of FNT Model The optimization of FNT includes the tree-structure and parameter optimization. Finding an optimal or near-optimal neural tree is formulated as a product of evolution. A number of neural tree variation operators are developed as follows:

Mutation Four different mutation operators were employed to generate offspring from the parents.These mutation operators are as follows:

(1) Changing one terminal node: randomly select one terminal node in the neural tree and replace it with another terminal node;
(2) Changing all the terminal nodes: select each and every terminal node in the neural tree and replace it with another terminal node;
(3) Growing: select a random leaf in hidden layer of the neural tree and replace it with a newly generated subtree.

Table 1. The activation functions

Gaussian function	$f(x) = exp(-\frac{(x-a)^2}{b^2})$
Flexible unipolar sigmoid function	$f(x, a) = \frac{2\|a\|}{1+e^{-2\|a\|x}}$
Flexible bipolar sigmoid function	$f(x, a) = \frac{1-e^{-2xa}}{a(1+e^{-2xa})}$

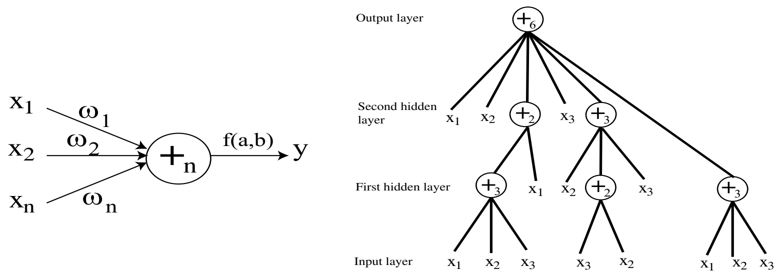

Fig. 1. A flexible neuron operator (left), and a typical representation of the FNT with function instruction set $F = \{+_2, +_3, +_4, +_5, +_6\}$, and terminal instruction set $T = \{x_1, x_2, x_3\}$ (right)

(4) Pruning: randomly select a function node in the neural tree and replace it with a terminal node.

The neural tree operators were applied to each of the parents to generate an offspring using the following steps: (a)A Poission random number N, with mean λ, was generated. (b)N random mutation operators were uniformly selected with replacement from above four mutation operator set. (c)These N mutation operators were applied in sequence one after the other to the parents to get the offsprings.
Crossover Select two neural trees randomly and select one nonterminal node in the hidden layer for each neural tree randomly, and then swap the selected subtree. The crossover operator is implemented with a pre-defined a probability 0.3 in this study.
Selection Evolutionary programming (EP) style tournament selection was applied to select the parents for the next generation. Pairwise comparison is conducted for the union of μ parents and μ offsprings. For each individual, q opponents are chosen uniformly at random from all the parents and offspring. For each comparison, if the individual's fitness is no smaller than the opponent's, it receives a selection. Select μ individuals out of parents and offsprings, that have most wins to form the next generation. This is repeated for each generation until a predefined number of generations or when the best structure is found.

Parameter Optimization by PSO Parameter optimization is achieved by the PSO algorithm as described in Section 2. In this stage, the architecture of FNT model is fixed, and it is the best tree developed during the end of run of the structure search. The parameters (weights and flexible activation function parameters) encoded in the best tree formulate a particle. The PSO algorithm works as follows:

(a) Initial population is generated randomly. The learning parameters c_1 and c_2 in PSO should be assigned in advance.
(b) The objective function value is calculated for each particle.

(c) Modification of search point. The current search point of each particle is changed using Eqn.(2) and Eqn.(1).
(d) If maximum number of generations is reached or no better parameter vector is found for a significantly long time (100 steps), then stop, otherwise goto step (b).

3.2 NN Classifier

A neural network classifier trained by PSO algorithm with flexible bipolar sigmoid activation functions at hidden layer were constructed for the breast-cancer data set. Before describing the details of the algorithm for training NN classifier, the issue of coding is presented. Coding concerns the way the weights and the flexible activation function parameters of NN are represented by individuals or particles. A float point coding scheme is adopted here. For NN coding, suppose there are M nodes in hidden layer and one node in output layer and n input variables, then the number of total weights is $n * M + M * 1$, the number of thresholds is $M + 1$ and the number of flexible activation function parameters is $M + 1$, therefore the total number of free parameters in a NN to be coded is $n*M+M+2(M+1)$. These parameters are coded into an individual or particle orderly.

The simple loop of the proposed training algorithm for neural network is as follows.

S1 Initialization. Initial population is generated randomly. The learning parameters c_1 and c_2 in PSO should be assigned in advance.
S2 Evaluation. The objective function value is calculated for each particle.
S3 Modification of search point. The current search point of each particle is changed using Eqn.(2) and Eqn.(1).
S4 if maximum number of generations is reached or no better parameter vector is found for a significantly long time (100 steps), then stop, otherwise goto step **S2**;

3.3 WNN Classifier

In terms of wavelet transformation theory, wavelets in the following form

$$\Psi = \{\Psi_i = |a_i|^{-\frac{1}{2}}\psi(\frac{x - b_i}{a_i}) : a_i, b_i \in R, i \in Z\} \tag{7}$$

$$\begin{aligned} x &= (x_1, x_2, \ldots, x_n) \\ a_i &= (a_{i1}, a_{i2}, \ldots, a_{in}) \\ b_i &= (b_{i1}, b_{i2}, \ldots, b_{in}) \end{aligned}$$

are a family of functions generated from one single function $\psi(x)$ by the operation of dilation and translation. $\psi(x)$, which is localized in both the time space and the frequency space, is called a mother wavelet and the parameters a_i and b_i are named the scale and translation parameters, respectively.

In the standard form of wavelet neural network, the output of a WNN is given by

$$f(x) = \sum_{i=1}^{M} \omega_i \Psi_i(x) = \sum_{i=1}^{M} \omega_i |a_i|^{-\frac{1}{2}} \psi(\frac{x - b_i}{a_i}) \quad (8)$$

where ψ_i is the wavelet activation function of ith unit of the hidden layer and ω_i is the weight connecting the ith unit of the hidden layer to the output layer unit. Note that for the n-dimensional input space, the multivariate wavelet basis function can be calculated by the tensor product of n single wavelet basis functions as follows

$$\psi(x) = \prod_{i=1}^{n} \psi(x_i). \quad (9)$$

Before describing details of the PSO algorithm for training WNN, the issue of coding is presented. Coding concerns the way the weights, dilation and translation parameters of WNN are represented by individuals or particles. A float point coding scheme is adopted here. For WNN coding, suppose there are M nodes in hidden layer and n input variables, then the total number of parameters to be coded is $(2n + 1)M$. The coding of a WNN into an individual or particle is as follows:

$$|a_{11}b_{11} \ldots a_{1n}b_{1n}\omega_1|a_{21}b_{21} \ldots a_{2n}b_{2n}\omega_2| \ldots |a_{n1}b_{n1} \ldots a_{nn}b_{nn}\omega_n|$$

The simple loop of the proposed training algorithm for wavelet neural network is as follows.

S1 Initialization. Initial population is generated randomly. The learning parameters, such as c_1, c_2 in PSO should be assigned in advance.
S2 Parameter optimization with PSO algorithm;
S3 if maximum number of generations is reached or no better parameter vector is found for a significantly long time (100 steps), then go to step **S4**; otherwise goto step **S2**;
S4 Parameter optimization with gradient descent algorithm;
S5 If the satisfactory solution is found then stop; otherwise goto step **S4**.

3.4 Ensemble Classifier

A ensemble classifier of FNT, NN and WNN is also constructed in order to test the performance of the mixture of different classification models. A simple voting method is employed in this research.

4 Results

As a preliminary study, we made use of the Wisconsin breast cancer data set from the UCI machine-learning database repository [9]. This data set has 32

Table 2. Comparative results of the four classification methods for the detection of breast cancer

Cancer type	FNT(%)	NN(%)	WNN(%)	Ensemble(%)
Benign	93.31	94.01	94.37	95.42
Malignant	93.45	95.42	92.96	96.14

Table 3. The important features selected by the FNT algorithm

Cancer type	Important variables
Benign	x_{12},x_{14},x_{15},x_{16},x_{17},x_{18},x_{23},x_{24},x_{26},x_{27},x_{29}
Malignant	x_1,x_5,x_6,x_7,x_{11},x_{12},x_{16},x_{17},x_{20},x_{21},x_{29}

attributes (30 real valued input features) and 569 instances of which 357 are of benign and 212 are of malignant type. We randomly divided the training and test data sets. The first 285 data is used for training and the remaining 284 data is used for testing the performance of the different models.

All the models were trained and tested with the same set of data. As the data set has two different classes we performed a 2-class binary classification. The classification results for testing data set are shown in Table 2. It should be noted that the obtained FNT classifier has smaller size and reduced features and without a significant reduce in the accuracy. The important features for constructing the FNT models are shown in Table 3. In general, the ensemble of FNT, NN and WNN shows the best classification rate. A comparison of implementation time of the three classifier models are shown in Table 4. Receiver Operating Characteristics (ROC) analysis of the FNT, NN, WNN and their ensemble model is shown in Table 5.

Table 4. Comparison of training/testing time for FNT, NN and WNN classifiers

Model	Training time (minute)		Test time (ms)	
	Benign	Malignant	Benign	Malignant
FNT	54	58	15	42
NN	35	39	31	26
WNN	52	55	45	37

5 Conclusion

In this paper, we presented some advanced artificial intelligence techniques for the detection of breast cancer. As depicted in Table 2, the preliminary results

Table 5. Comparison of false positive rate (fp) and true positive rate (tp) for FNT, NN, WNN and ensemble classifiers

Cancer	FNT		NN		WNN		Ensemble	
Type	fp(%)	tp(%)	fp(%)	tp(%)	fp(%)	tp(%)	fp(%)	tp(%)
Benign	3.88	91.71	4.85	93.37	6.8	98.34	4.85	95.58
Malignant	2.76	86.41	4.97	96.12	9.4	97.09	3.88	93.21

are very encouraging. The best accuracy was offered by the ensemble method followed by the wavelet neural network for detecting benign types and PSO trained neural network for detecting the malignant type of cancer. An important advantage of the FNT model is the ability to reduce the number of input variables as presented in Table 3. ROC analysis (Table 5) illustrates that wavelet neural network has the highest false positive rate and the FNT model has the lowest false positive rates for detecting benign and malignant cancer. The time required to construct these models are not very much and hope these tools would assist the physician's effort to improve the currently available automated ways to diagnose breast cancer.

References

1. David B. Fogel , Eugene C. Wasson , Edward M. Boughton and Vincent W. Porto, A step toward computer-assisted mammography using evolutionary programming and neural networks, Cancer Letters, Volume 119, Issue 1, pp. 93-97, 1997.
2. Charles E. Kahn, Jr , Linda M. Roberts, Katherine A. Shaffer and Peter Haddawy, Construction of a Bayesian network for mammographic diagnosis of breast cancer, Computers in Biology and Medicine, Volume 27, Issue 1, pp. 19-29, 1997.
3. Shinsuke Morio , Satoru Kawahara , Naoyuki Okamoto , Tadao Suzuki, Takashi Okamoto , Masatoshi Haradas and Akio Shimizu, An expert system for early detection of cancer of the breast, Computers in Biology and Medicine, Volume 19, Issue 5, pp. 295-305, 1989.
4. Barbara S. Hulka and Patricia G. Moorman, Breast Cancer: Hormones and Other Risk Factors, Maturitas, Volume 38, Issue 1, pp. 103-113, 2001.
5. Jain, R. and Abraham, A., A Comparative Study of Fuzzy Classifiers on Breast Cancer Data, Australiasian Physical And Engineering Sciences in Medicine, Australia, Volume 27, No.4, pp. 147-152, 2004.
6. Kennedy, J. and Eberhart, R.C., Particle Swarm Optimization. Proc. of IEEE International Conference on Neural Networks. **IV** (1995) 1942-1948
7. Chen, Y., Yang, B., Dong, J., Nonlinear systems modelling via optimal design of neural trees. International Journal of Neural ystems. **14**, (2004) 125-138
8. Chen, Y., Yang, B., Dong, J., Abraham A.: Time-series forcasting using flexible neural tree model. Information Science. In press.
9. Merz J., and Murphy, P.M., UCI repository of machine learning databases. http://www.ics.uci.edu/- learn/MLRepository.html, (1996).