Genetic Algorithm - Artificial Neural Network (GA-ANN) Hybrid Intelligence for Cancer Diagnosis

Fadzil Ahmad[1,2], Nor Ashidi Mat-Isa[1], Zakaria Hussain[2], Rozan Boudville[2], Muhammad Khusairi Osman[2]
[1]School of Electrical & Electronic Engineering, Universiti Sains Malaysia,
Penang, Malaysia.
[2]Faculty of Electrical Engineering, Universiti Teknologi MARA,
Penang, Malaysia.
fadzil_ahmad@ppinang.uitm.edu.my

***Abstract*— Artificial Neural Network (ANN) is one of the most promising biological inspired computational intelligence techniques. However designing an ANN is a difficult task as it requires setting of ANN structure and tuning of some complex parameter. On the other hand, Genetic Algorithm (GA) as a global search technique is useful for complex optimization problem where the numbers of parameters are large and difficult to obtain. In this paper GA has been used to simultaneously select significant features as input to ANN and automatically determine the optimal number of hidden node. Meanwhile the ANN training is done by Levenberg Marquardt (LM) algorithm. A new procedure in obtaining optimal ANN architecture is also described which based on feature importance determine by Genetic Algorithm. Simulation results on cancer dataset proved that the proposed method has achieved the highest 97% average percentage of correct classification with the absent of 2nd and 5th feature.**

Keywords - Genetic Algorithm, Artificial Neural Network, Computational Intelligence, Feature Selection and Hidden Node Optimization.

I. INTRODUCTION

ANN is getting popular among researchers in many applications. This is mainly due to its ability to learn from training data and employ what it has learned to classify pattern from unseen test data. This is known as generalization ability. It is also shown that ANN can virtually approximate any function in a stable and efficient manner [1]. However, designing an ANN is a difficult task involves setting of ANN structures such as number of hidden node (HN) and tuning of parameters such as connection weights and learning rate.

Determining the number of HN is one of the critical design issues in ANN applications. Less number of HN generally results in under-fitting problem. On the other hand if the HN are too big the network will suffer from over-fitting problem where it can only make good prediction during training phase but poor performance in testing phase [2]. Over-fitting can also produce wild predictions even with noise-free data.

Another important factor that affects the performance of an ANN is the selection of appropriate feature subset. Not all the features are equally important; some are noisy, redundant and irrelevant for a given task. By selecting only the relevant feature subset, the generalization ability could be improved. Besides, the ANN complexity will be reduced and thus minimizing computational effort especially during the training phase.

Optimization of input features and HN size should be done automatically and simultaneously (shaded area in Fig. 1). This is due to the fact that the optimal number of HN is highly depends on the number of input features [3]. Meanwhile the input features may consist of irrelevant and redundant information. By removing them, the generalization ability could be improved. Consequently the required number of HN is expected to be changed as well. So this is considered as multi parameter optimization and the use of GA is a promising alternative.

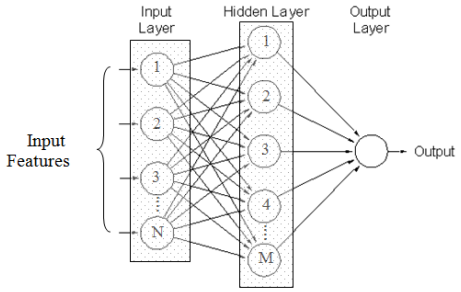

Figure 1. Structure of ANN and location for optimization (shaded area)

GA is a global search technique used in computing [4]. GA is categorized as global search heuristics. So it is not guaranteed to obtain the optimal solution. GA is a particular class of evolutionary algorithms (also known as evolutionary computation) that uses techniques inspired by evolutionary biology such as inheritance, selection, crossover, and mutation. Recently the trend to hybridize GA and ANN is getting popular among researchers [5-8]. The advantages offered by these techniques are forming a better GA-ANN hybrid intelligence system that can improve generalization and at the same time ease the ANN design process.

In this paper, GA is used to extract the importance of individual feature in the dataset as proposed by Kermani et al in [9]. In their work, '1' and '0' binary

978-0-7695-4158-7/10 $26.00 © 2010 IEEE
DOI 10.1109/CICSyN.2010.46

chromosomes were used as an indication to either a particular feature is selected or not. What is new in this work is that the role of GA has been extended to perform HN size optimization at the same time. For that purpose, additional bits are included in the chromosome to represent the HN size.

Beside, two different set of feature importance that is based on the best and average classification accuracy have been established. Based on the best and average feature importance, the work has been extended by some series of experiment by manual selection of feature subsets in which GA only explore the HN size. First feature subset is created by inserting the first rank feature, followed by second feature subset that consists of 1st and 2nd rank feature. The last subset will consist of all features. In this way, good feature subset that might be missed by GA can be manually explored. As previously mentioned, GA not always guarantee exact solution as it is not from a family of deterministic approach.

II. An Intelligence Cancer Diagnosis

It is a fact that cancer is one of the most deathly diseases in many countries around the world. Linked with the high use of tobacco, alcohol, low consumption of fruit and vegetable, the number of new cancer case is expected to increase annually from 10.9 million in 2002 to 16 million in 2020 [10]. In spite of a very intensive research effort there is still no concrete evidence of the root cause, preventive methods and the cure for cancer. In reality, some of the cancerous tissue appears to be very aggressive. The only way to reduce the mortality rate among cancer patients is through early detection and with a proper treatment the risk for the cancerous tissue to spread to other organ can be minimized. Usual tradition method is time-consuming and incurs unnecessary burden to radiologist. By the time it is detected, it may be at critical stage.

To date, there are many efforts to computerize the diagnosis of cancer as either benign (not cancerous) or malignant (cancerous). It is expected to enhance the classification accuracy and reduce the waiting time. At the same time, false classification due to human error can be reduced. In this work, computational intelligence method based on GA-ANN hybrid intelligence has been developed for breast cancer detection. This paper present some result achieves by applying this technique to a real world cancer dataset obtains from UCI machine learning repository [11].

III. Simultaneous Input Selection and HN Optimization: The Algorithm

The following outline summarizes how the proposed algorithm works.

1. Data preparation and setting of population size, maximum generation and mutation rate.
2. Create random initial population. A population is an array consists of binary sequence '0' and '1'. X-dimension represents chromosome length while Y-dimension represents population size.
3. Decode each chromosome in the population to obtain the selected feature and HN size. Feature that is not selected is removed from training, validation and test dataset.
4. Create the network and start LM training using training dataset.
5. Use validation data to simulate the resultant network to obtain mean square error (MSE). Compute the fitness for all individuals based on validation MSE. Keep track the best network from each generation which is with the smallest MSE network.
6. Perform reproduction operation (selection, crossover, replacement and mutation) to generate next population.
7. Repeat step 3 to 6 until maximum generation is reached.
8. Use test data to simulate the best network to obtain the final performance (Classification accuracy and HN size).

A. Representation

Binary code GA is used to represent the chromosome. The chromosome is divided into two parts as illustrated in Fig. 2. The first part consists of N bits represents N number of features of the dataset. Value '1' or '0' indicates either feature at that particular location is selected or not. The second part of the chromosome represents the M number of nodes at ANN hidden layer. The length of second part of the chromosome is automatically set so that equivalent decimal value is not smaller than twice of the feature number. The M binary bits allow GA to explore up to a maximum of 2^M HN sizes. This representation is preferred because it requires small code even when the network size is big.

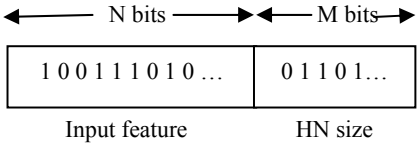

Figure 2. Binary coded GA representation

B. Fitness Function, Selection and Replacement

The fitness function is calculated based on Mean Square Error (MSE) of the validation dataset as in (1), where p is the number of output nodes, q is the number of examples, D_i^j is the output value of i^{th} output node for j^{th} pattern and O_i^j is target value of i^{th} output node for j^{th} pattern.

$$MSE = \frac{100}{pq} \times \sum_{i=1}^{p} \sum_{j=1}^{q} (D_i^j - O_i^j)^2 \quad (1)$$

Chromosome with smaller MSE will have better chance to be selected as parent for mating and survive for next generation. This is achieved via the selection process where the MSE value is first converted into its ranking. The bigger the ranking value the better the chromosome is. Next, the fitness of i^{th} chromosome is calculated based on their ranking value as in (2).

$$fitness(i) = \frac{100 \times rank(i)}{\sum_{j=1}^{population\ size} rank(j)} \quad (2)$$

Table I provides the example of the fitness value for each chromosome resulted from calculation using (2). After that, rank based roulette-wheel selection procedure is applied to select two chromosomes as parent and crossover operation will take place. This procedure produces two new children that inherit genetic information from their parent. The next step is the replacement process where the newly born offspring will replace the last two worst rank chromosomes. This is to mimic the biological evolution process where most fit chromosome will survive for next generation while the worst chromosome will die off. Finally all the chromosomes will go through mutation operation.

TABLE I. AN EXAMPLE OF FITNESS VALUE FOR EACH CHROMOSOME OBTAIN FROM VALIDATION MSE

chromosome no	val MSE	ranking	fitness value
1	0.0258	1	5
2	0.0246	3	14
3	0.018	5	24
4	0.0257	2	10
5	0.0172	6	29
6	0.0229	4	19

C. Crossover and Mutation

In this work, standard single point crossover and mutation operation is used (as illustrated in Fig. 3 and Fig. 4). For each generation, only two selected chromosomes (parent) will go through crossover operation process. In crossover operation, the first step is to generate random crossover location. Then, all bits after that location is swapped between the parents to produce two off-springs.

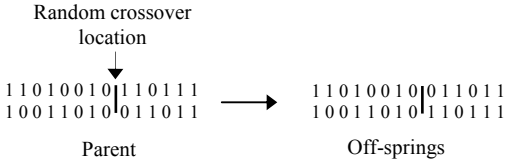

Figure 3. Single point crossover operation

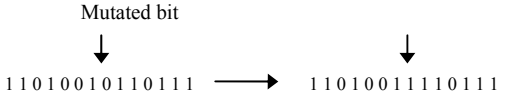

Figure 4. Mutation operation

The two off-springs will then replace the worst two chromosomes. Subsequently all the chromosomes will go through mutation operation. In mutation operation each bit in the chromosome will be flipped either from '1' to '0' or '0' to '1' based on mutation probability.

IV. METHODS

A. The dataset

The proposed algorithm is applied on the real world cancer dataset obtains from UCI machine learning benchmark repository [11]. The original cancer dataset is contributed by Dr. William H. Wolberg from the University of Wisconsin Hospitals, Madison [12]. The dataset requires a correct diagnostic of breast lumps as either benign or malignant. The original dataset consists of 699 examples or input patterns, each with 9 recorded criteria of lumps (input features) gathered by microscopic examination. They are (1) clump thickness, (2) uniformity of cell size, (3) uniformity of cell shape, (4) marginal adhesion, (5) single epithelial cell size, (6) bare nuclei, (7) bland chromatin, (8) normal nucleoli, and (9) mitosis. There are 16 examples with missing values in the dataset.

B. Experiment Setup.

In this study, fully connected feedforward multilayer perceptrons (MLP) Neural Network with one hidden layer has been used. The number of HN is determined by GA, while the input nodes are same as GA selected input features. Levenberg-Marquardt (LM) algorithm which is one of the variations of error back propagation algorithm is used for weight training. This algorithm is preferred as it appears to be the fastest method for training moderate-sized feedforward neural networks [13]. Regarding the output layer, it consists of two output nodes same as the number of classification in cancer data set.

The GA parameter that has been used in all the experiments is as follows: the population size (15), crossover probability (1 - always applied), mutation probability (0.15) and maximum generation (150).

The dataset was preprocessed before feeding into the system. The original value of input data that varies from 1 to 10 was rescaled within the range of 0 to 1. Concerning

the outputs, the original value is either 2 for benign or 4 for malignant case. The two output classes are encoded based on 1-of-C encoding approach, one binary variable for each class. For this specific problem, benign and malignant class is encoded with '0 1' and '1 0' respectively. When input data is supplied to the system, it will response to the class associated with the output node with the highest output value. For example, when a particular input pattern is presented, the actual value of output nodes could be '-0.1342 0.6119'. The system will classify this pattern as benign ('0 1'). This is referred as winner takes all approach.

16 missing value examples were removed from dataset. The balance 683 examples are divided into 10 sub examples, first three sub examples contain 69 examples and the remaining contain 68 examples. The sub examples are then partitioned into training, validation and test data. For instance, data partition 1-5,6-9,10 means sub examples 1,2,3,4 & 5 is used for training, 6,7,8 & 9 for validation and sub example 10 for testing. This is to ensure that all the data are used at least once for training, validation and testing.

As far as the performance evaluation is concern, 10 fold cross validation scheme is adapted. The final accuracy is based on the average of 10 independent runs. This should eliminate any suspicion that the result is influenced by the distribution or the order of the data sample.

V. Results and Discussion

The experiment on simultaneous GA selected feature and HN size consists of 10 independent runs where each run caters for different partition of training, validation and test data. The experiment is repeated 10 times that made a total of 100 independent runs has been executed. Table II shows the best result of the experiment. 'tr', 'val' and 'tst' MSE stand for training, validation and test MSE respectively. The average GA selected feature over the 10 runs indicates the feature importance. Mean while, the performance in term of HN size and classification accuracy (correct classification either for benign or malignant) are also based on the average of the 10 runs.

From the result, it is clear that different data partition exhibits different performance. Test classification accuracy varies from 92.75% (for data partition of 2-6,7-10,1) to 100% (for data partition of 7-1,2-5,6). Mean while, the average tests accuracy and HN size is 96.9% and 2.7 respectively. At the same time, feature number 1, 6, 7 and 8 are frequently selected by the GA from the 10 runs and this indicates that they are significant feature.

TABLE II. Result on Simultaneous Feature Selection and HN Size Determination

Data partition	tr MSE	val MSE	tst MSE	GA selected feature									GA selected no of HN	Test Classification Accuracy
1-5,6-9,10	0.034	0.011	0.007	1	0	1	1	1	1	1	1	1	2	98.53
2-6,7-10,1	0.025	0.01	0.052	1	1	1	0	0	1	1	1	1	3	92.75
3-7,8-1,2	0.037	0.014	0.014	1	0	1	0	1	0	1	1	0	1	98.55
4-8,9-2,3	0.035	0.023	0.026	1	1	1	0	1	0	0	1	0	2	97.1
5-9,10-3,4	0.019	0.02	0.059	0	0	1	0	0	1	1	1	1	3	94.12
6-10,1-4,5	0.016	0.031	0.045	1	0	0	1	1	1	0	0	0	2	94.12
7-1,2-5,6	0.016	0.029	0.004	1	1	0	1	0	1	1	1	1	3	100
8-2,3-6,7	0.019	0.025	0.026	1	1	0	1	0	1	1	0	1	3	97.06
9-3,4-7,8	0.02	0.027	0.014	1	0	0	1	0	1	1	1	0	4	98.53
10-4,5-8,9	0.029	0.018	0.01	1	0	1	1	0	1	1	1	1	4	98.53
Average	0.025	0.021	0.026	0.9	0.4	0.6	0.6	0.4	0.8	0.8	0.8	0.6	2.7	96.9

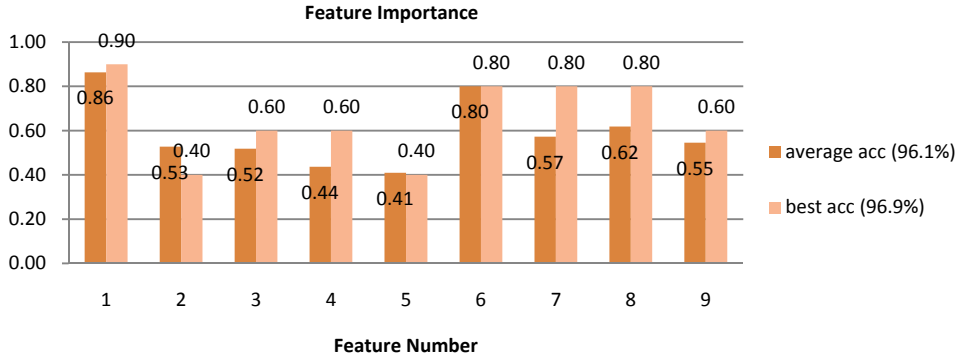

Figure 5. Comparison of feature importance between the best and average classification accuracy over 10 experiments.

The best and the average classification accuracy over 10 experiments is 96.9% and 96.1% respectively. While the average HN size for the best and the average classification accuracy experiment is 2.7 and 5.3 respectively. Feature importance or the relative frequencies of a particular feature selected by GA, for the best and average accuracy is depicted in Fig. 5.

Further experiment was carried out in order to validate the feature importance that automatically obtained by GA from previous experiment. The feature importance that are based on the best and average classification accuracy were first ranked, with most important feature receives number 1. If there is more than 1 feature that shared the same important value, the group received the same rank number. The feature ranking based on the best and average classification accuracy is tabulated in Table III. For average accuracy ranking, each feature receives different ranking value because all the feature importance value is different. On the other hand, there are 4 different ranks (feature 1 – ranking 1, feature 6,7,8 – ranking 2, feature 3,4,9 – ranking 3 and feature 2,5 – ranking 4) in best accuracy ranking since there are 4 different importance values. The number of ranking value will determine the number of run in next experiment.

TABLE III. FEATURE RANKING BASED ON THE BEST AND AVERAGE CLASSIFICATION ACCURACY

feature no		1	2	3	4	5	6	7	8	9
ranking	average accuracy (96.1%)	1	6	7	8	9	2	4	3	5
	best accuracy (96.9%)	1	4	3	3	4	2	2	2	3

At this stage, two set of feature ranking for each type of feature importance has been identified. The next experiment determined the optimum feature subset that gave the best performance. In this experiment, only the HN size is being evolved by the GA while the feature was manually selected. Separate experiment was carried out based on the best accuracy feature and average accuracy feature ranking. The experiment begins with the highest ranked feature as input to the NN and next ranked feature is inserted one after another in next run. At last run the subset will consist of all features. The result for these experiments is tabulated in Table IV and V.

From Table IV, it is obvious that initially, adding relevant feature as input to ANN causes the performance to improve significantly. In term of HN size, it is reduced from 8.5 to the minimum of 3.0. Mean while the classification accuracy has improved from 85.2% to the maximum of 96.6%. The best performance achieved by feature subset '100001111'. Adding more features did not improve the performance any further. In fact, the HN size started to deteriorate from 3.0 to 4.3 while the classification accuracy almost level.

TABLE IV. PERFORMANCE OF GA-ANN BASED ON AVERAGE ACCURACY FEATURE RANKING

Manually selected feature subset	GA selected HN size	Ave test set MSE	Ave classification acc on test set
100000000	8.5	10.8%	85.2%
100001000	8.8	4.3%	94.7%
100001010	8.3	3.3%	96.3%
100001110	4.4	3.0%	96.3%
100001111	**3.0**	**2.9%**	**96.6%**
110001111	3.1	2.7%	96.5%
111001111	5.2	2.8%	95.9%
111101111	3.9	2.6%	96.5%
111111111	4.3	2.7%	96.5%

Table V shows the experiment based on average accuracy. Feature subset '101101111' gave 3.2 HN sizes, 2.6% MSE and the highest classification accuracy of 97.0%.

TABLE V. PERFORMANCE OF GA-ANN BASED ON BEST ACCURACY FEATURE RANKING

Manually selected feature subset	GA selected HN size	Ave test set MSE	Ave classification acc on test set
100000000	8.5	10.8%	85.2%
100001110	4.4	3.0%	96.3%
101101111	**3.2**	**2.6%**	**97.0%**
111111111	4.3	2.7%	96.5%

The similarity between the two experiments is that, the peak performance of GA-ANN system will give the best classification accuracy and the minimum HN size.

Comparison between feature subset '100001111' and '101101111', in term of classification accuracy, '101101111' is better than '100001111' by 0.4% but in terms of HN size, '100001111' is better than '101101111' by 0.2. There must be a tradeoff between the number of selected feature and the performance.

From the result, it is clear that by integrating feature selection into the ANN classifier caused the performance to improve drastically. Performance comparison with (feature subset '101101111') and without (feature subset '111111111') feature selection in term of HN size is 3.2 and 4.3 respectively and in term of classification accuracy is 97.0% and 96.5% respectively.

TABLE VI. COMPARISON WITH OTHER METHOD

Method	C4.5	C4.5 rules	ITI	LMDT	CN2	MNNO	**GA-ANN**
Classification Accuracy %	94.3	94.7	91.1	95.7	94.4	96.6	**97.0**

Table VI compares the performance of the proposed GA-ANN method with C4.5, C4.5 rules, ITI, LMDT, CN2[14] and MNNO[15] for the specific cancer dataset. The comparison is in term of the average classification accuracy. Reference [15] reports the average percentage of misclassification which is 3.4%. The value is converted to classification accuracy which is 96.6% (100% minus 3.4%). The result clearly demonstrated that the proposed method is superior to other method.

VI. Conclusion

This paper has established a new procedure to obtain optimal ANN design. At first, simultaneous and automatic feature selection and determination of HN based on GA-ANN hybrid intelligence is used to find the feature importance. Based on the feature importance, a series of feature subset is manually created and the ANN is retrained using this subset.

The approach which is based on a simple binary coded GA representation has overcome the tedious trial and error design process of ANN especially in determining the HN size and the selection of significant feature. It is also shown that employing feature selection in ANN classifier improves the classification accuracy and at the same time reduces the network complexity. For this dataset it is concluded that feature subset '101101111' gives the optimal ANN in term of generalization ability and complexity.

In view of good result obtained, it is concluded that our approach that is based on GA-ANN hybrid intelligence, assisted with LM training capable to obtain high generalization ability ANN for cancer dataset. Further work can be carried out to evaluate the proposed algorithm on dataset from other application.

Acknowledgment

This research is partially supported by Universiti Sains Malaysia Research University Postgraduate Research Grant Scheme (USM-RU-PGRS) under grant no. 1001/PELECT/8043002.

References

[1] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks,* vol. 2, pp. 359-366, 1989.

[2] S. Lawrence, C. Giles, and A. Tsoi, "Lessons in neural network training: Overfitting may be harder than expected," 1997, pp. 540-545.

[3] G. Mirchandani and W. Cao, "On hidden nodes for neural nets," *Circuits and Systems, IEEE Transactions on,* vol. 36, pp. 661-664, 1989.

[4] D. Goldberg, *Genetic Algorithms in Search and Optimization*: Addison-wesley, 1989.

[5] T. Jingwen and G. Meijuan, "Network Intrusion Detection Method Based on High Speed and Precise Genetic Algorithm Neural Network," in *Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC '09. International Conference on*, 2009, pp. 619-622.

[6] G. Meijuan and T. Jingwen, "Wireless Sensor Network for Community Intrusion Detection System Based on Improved Genetic Algorithm Neural Network," in *Industrial and Information Systems, 2009. IIS '09. International Conference on*, 2009, pp. 199-202.

[7] G. Meijuan, T. Jingwen, and X. Jin, "Building Logistics Cost Forecast Based on High Speed and Precise Genetic Algorithm Neural Network," in *Intelligent Systems and Applications, 2009. ISA 2009. International Workshop on*, 2009, pp. 1-4.

[8] L. Xiao, Z. Wang, and X. Peng, "Research on Congestion Control Model and Algorithm for High-Speed Network Based on Genetic Neural Network and Intelligent PID," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, 2009, pp. 1-6.

[9] B. G. Kermani, M. W. White, and H. T. Nagle, "Feature extraction by genetic algorithms for neural networks in breast cancer classification," in *Engineering in Medicine and Biology Society, 1995., IEEE 17th Annual Conference*, 1995, pp. 831-832 vol.1.

[10] "World health organization," 2009.

[11] C. Blake and C. Merz, "UCI Repository of machine learning databases [http://www. ics. uci. edu/~ mlearn/MLRepository. html]. Irvine, CA: University of California," *Department of Information and Computer Science,* vol. 460, 1998.

[12] W. Wolberg and O. Mangasarian, "Multisurface method of pattern separation for medical diagnosis applied to breast cytology," *Proceedings of the national academy of sciences of the United States of America,* vol. 87, p. 9193, 1990.

[13] H. Demuth and M. Beale, "Neural network toolbox," *MathWorks Version,* vol. 4, 2001.

[14] A. Hoang, "Supervised classifier performance on the UCI database," University of Adelaide, 1997.

[15] P. Naval and J. Yusiong, "An Evolutionary Multi-objective Neural Network Optimizer with Bias-Based Pruning Heuristic," *Advances in Neural Networks–ISNN 2007,* pp. 174-183.