Feature Selection for ANNs using Intelligent Genetic Algorithms in Diagnostic of Hepatitis

A.A. Soltani nia
Department of Electrical Engineering and Electronics,
The University of Azad Islamic Tehran center, Tehran, I.R.I.
Abbas.Soltaninia@gmail.com

Abstract: *Artificial Neural Networks (ANNs) can be used successfully to detect the Hepatitis patient from healthy people using statistical estimates of blood factors and signs of the disease as input features. One of the main problems facing the use of ANNs is the selection of the best inputs to the ANN, allowing the creation of compact, highly accurate networks that require comparatively little pre-processing. This paper examines the use of a Genetic Algorithm (GA) to select the most significant input features from a large set of possible features in diagnostic of the Hepatic patients. Using a large set of 19 different features, the GA is able to select a set of 10 features that give 100% recognition accuracy.*

Keywords: Artificial Neural Networks, Genetic Algorithm, Hepatitis, MLP

1 Introduction

Research has shown that ANNs are promising solution to several different problem areas [11, 12, 13]; however many of the input features used require a significant computational effort to calculate.

A feature selection process using GAs is used in order to isolate features that are provide the most significant information for the neural network, whilst cutting down the number of inputs required for the network.

The work presented in this paper is based around database that performed on Hepatitis disease by Donor: G.Gong (Carnegie-Mellon University), Bojan Cestnik and Jozef Stefan in November in 1988 [4, 8]. this database includes 155 instances that classified with 19 features from each other to 2 classes
1_Healthy people class = 123 individuals
2_Petiant people class = 32 individuals

The features of the classification include:
1. AGE: 10, 20, 30, 40, 50, 60, 70, 80
2. SEX: male, female
3. STEROID: no, yes(Cortico Steroid drag usage)
4. ANTIVIRALS: no, yes(Antiviral drag usage)
5. FATIGUE: no, yes
6. MALAISE: no, yes
7. ANOREXIA: no, yes
8. LIVER BIG: (Increasing of the liver capacity) no, yes
9. LIVER FIRM: no, yes(Firming of the liver)
10. SPLEEN PALPABLE: no, yes
11. SPIDERS: no, yes(signs like spiders)
12. ASCITES: (collection of liquids in stomach) no, yes
13. (inflammation of vessels of the lower limb) VARICES: no, yes
14. BILIRUBIN: 0.39, 0.80, 1.20, 2.00, 3.00, 4.00
    (A feature for diagnostic of the jaundice)
15. ALK PHOSPHATE: 33, 80, 120, 160, 200, 250
    (A feature for liver function test)
16. SGOT: 13, 100, 200, 300, 400, 500,
    (A feature for liver function test)
17. ALBUMIN: 2.1, 3.0, 3.8, 4.5, 5.0, 6.0
    (A protein in blood)
18. PROTIME: 10, 20, 30, 40, 50, 60, 70, 80, 90
    (Window period)
19. HISTOLOGY: no, yes

This ANN has 77 patterns for training phase and 78 patterns for testing phase.
The experimental operations show that the best accuracy obtains when the average error is less than (0.29946).
So: Me1=average error=0.29946

2 Neural Networks

The MLP(Multi Layers Perceptrons) used in this work consists of three hidden layers and an output layer, the 3 hidden layers having a logistic activation function, whilst the output layer uses a linear activation function. The size of the hidden layer is determined by the genetic algorithm itself during training [5,10, 11].

This allows training to proceed at a faster rate than an exhaustive training process that checks different sizes of the each three layers. The size of the output layer is determined by the number of outputs required. This is set at 10 neurons for the particular application.

Training of the network is carried out using a standard back-propagation algorithm, and the network is trained for 1,000,000 epochs, using 50% of the data set as training data, and the remaining 50% as the test and validation set [14, 15].

3 Genetic Algorithms

GAs have been gaining popularity in a variety of applications which require global optimization of a solution. A good general introduction to genetic algorithms is given in [6]. The prime component of a genetic algorithm is the genome. The genome is an encoded set of instructions which the genetic algorithm will use to construct a new model or function (in this case the inputs to a neural network). The best type of encoding is very much problem dependent, and may require some form of combination of two or more encoding types (binary, real numbers, etc) in order to get the optimum results [1, 2, 3].

The GA is allowed to select subsets of various sizes in order to determine the optimum combination and number of inputs to the network. The emphasis in using the genetic algorithm for feature selection is to reduce the computational load on the training system while still allowing near optimal results to be found relatively quickly [17, 18].

4 Feature Selection & Encoding

Feature selection of the GA is controlled through the values contained within the genome generated by the GA [19].

On being passed a genome with (N + 1) values to be tested, the first N values are used to determine which rows are selected as a subset from the input feature set matrix. Rows corresponding to the numbers contained within the genome are copied into a new matrix containing N rows. The last value of the genome determines the number of neurons present in these layers of the network [9].

For this particular application, a simple real number based genome string was used. For a training run requiring 77 different inputs to be selected as a subset of Q possible inputs, the genome string would consist of (77 + 1) real numbers.

The maximum number of neurons permissible in the first layer is defined as S [16, 17].

Each of the first N numbers (x) in the genome is constrained
as $0 < x < (Q - 1)$, whilst the last number, (x), is constrained to lie within the bounds $1< x < S$. This means that any mutation that occurs will be bounded within the limits set at the definition of the genome. The classification performance of the trained network using the whole dataset was returned to the GA as the value of the fitness function.

The GA uses 30 populations size of 20 individuals, starting with randomly generated genomes. The probability of mutation was set to 0.005, whilst the probability of crossover was set to 0.95. An elitist population model is used, meaning that the best individual in the previous population is kept in the next population, and preventing the performance of the GA worsening as the number of generations' increase [7, 20].

5 Training and Simulation

Training was carried out using three data sets; one feature set comprised all the statistically based features (6 features) and this dataset was combined with all the statistical feature sets to form an input feature set of 19 inputs. Each feature set contained a total of 155 cases.

Using the genetic algorithm running for a total of 30 generations, each containing 20 members (meaning the training of 600 neural networks), eight separate cases were tested using

various numbers of inputs, varying from five to twelve.

As a comparison, a neural network was trained using each feature set. These were trained for a total of 1,000,000 epochs, and allowed to choose the best size of intermediate layer between 2 and 9 neurons.

5.1 Figures and Tables

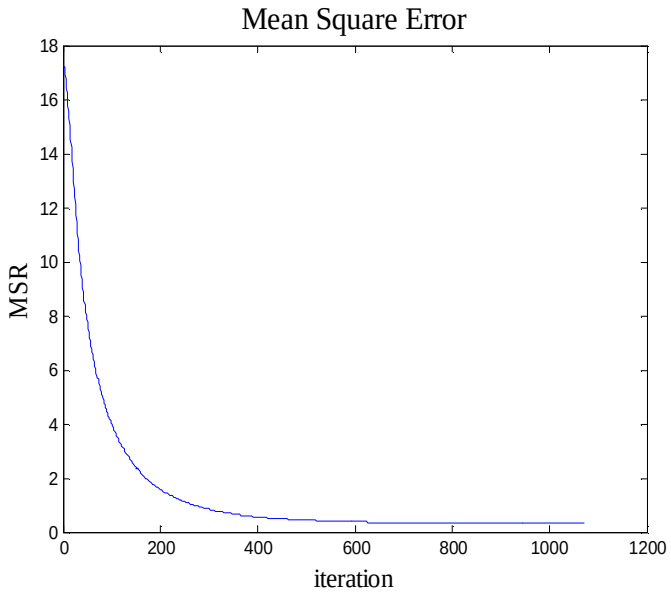

Figure1-1: the Mean Square of the MLP ANN was improved with GA in 1076 epochs

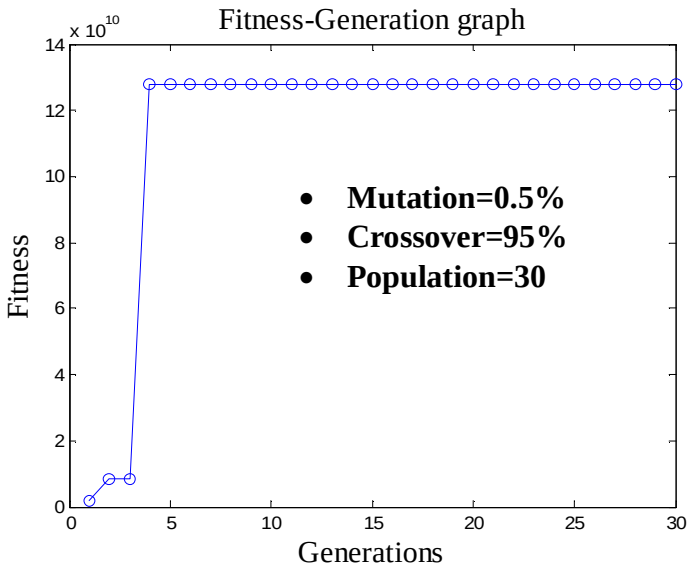

Figure 1-2: illustration of the Fitness-Generations which is used by GA

TP=66
TN= 11
FP=1
FN=0

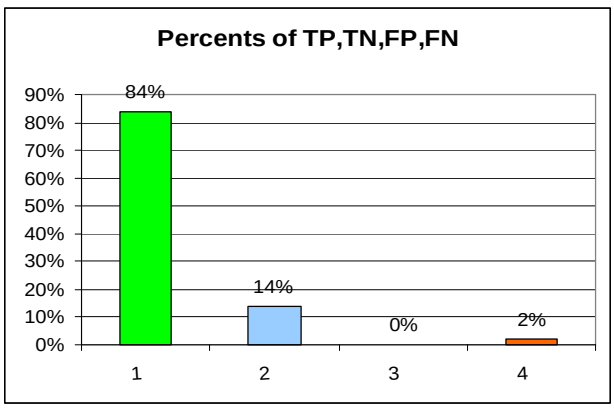

Figure 1-3: illustration of computational pattern of the TP, TN, FP and FN.

- **Sensitivity= %100**
- **Specificity=%92**
- **Accuracy=%99**

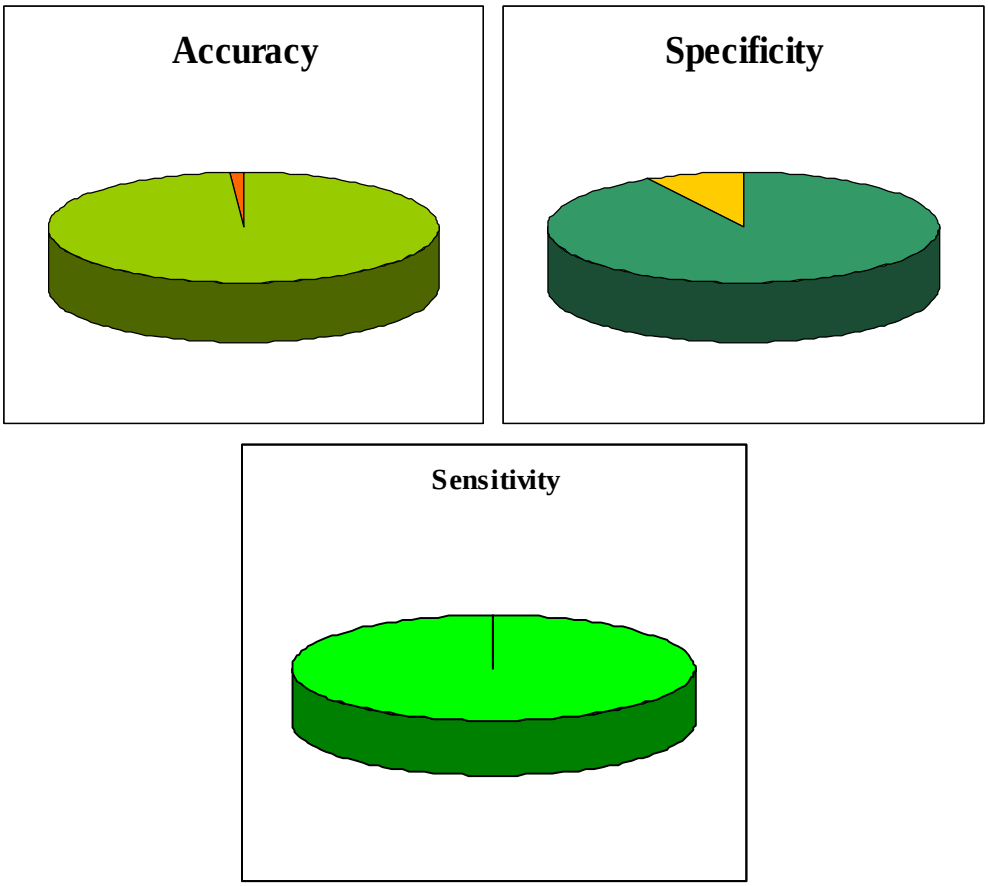

Figure 1-4: illustration of the statistic pattern of the ANN quality parameters.

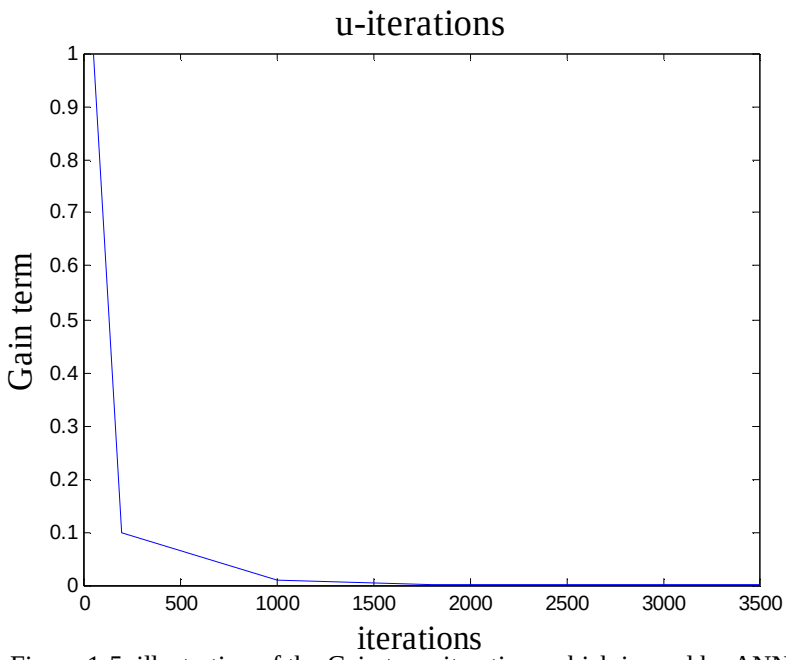

Figure 1-5: illustration of the Gain term-iterations which is used by ANN

6 Results

6.1. Results: ANN

Table 1 shows a summary of results for all three feature sets used. The " No. Neurons" quoted in the second column is the number of neurons used in the hidden layer of the best network in each training run.

"TP" represents the percentage success rate of the ANN using the complete dataset, which are healthy people and classified as normal people

The "TN" category details the number of patients that were classified as patient people again expressed as a percentage of the total dataset.

The "FN" rate details the number of "normal" people that were misclassified as patient, expressed as a percentage of the total dataset.

The "FP" category details the number of patients that were classified as normal people again expressed as a percentage of the total dataset.

The aggregate of the FN and FT rates are also the lowest of all the different feature sets.

Table 1: Comparison between standalone ANN and GA with ANN after 30 generations, for all three data sets

Straight ANN	**TP=65**	**TN=7**	**FP=2**	**FN=4**	**Sen=95%**	**Spe=78%**	**Acc=92%**
GA with the best ANN	**TP=66**	**TN=11**	**FP=1**	**FN=0**	**Sen=100%**	**Spe=92%**	**Acc=99%**
GA with ANN	**TP=65**	**TN=8**	**FP=2**	**FN=3**	**Sen=96%**	**Spe=80%**	**Acc=94%**

The feature set containing all the data has a larger number of features than any of the other sets and the drop in performance may be due to the limited amount of time that the training algorithm is allowed to arrive at a result.

It may be that due to the large number of features the training algorithm is unable to arrive at a better solution before the training cycle is stopped.

6.2 Genetic Algorithm with ANN after 30 Generations

Table 1 shows the performance of the different feature sets after running under the GA for 30 generations. All of the datasets have their best performance in excess of 92%. The feature set using all the available training data has managed to achieve an accuracy of 100%, indicating accurate classification.

This is achieved using only six inputs out of the possible 156. Using 9 neurons in the hidden layers, a relatively small network has been created that fulfills the criteria set earlier on.

A network of this size would be ideal for a real-time implementation on a small chip or micro-controller.

7 Conclusions

The use of the Genetic Algorithm allows feature selection to be carried out in an automatic manner, meaning that input combinations can be selected without the need for human intervention.

This technique offers great potential for use in diagnostic of the Hepatitis, where there are often hundreds and even thousands of different factors available to a diagnostic system, and selection of the most relevant features is often difficult.

It has been shown that the Genetic algorithm is capable of selecting a subset of 6 inputs from a set of 156 features that allow the ANN to perform with nearly 100% accuracy. The performance of networks trained using the feature selection was consistently higher than those trained without feature selection.

8 Acknowledgements

Thanks must be expressed to:

Dr. H. Ebrahimi rad

9 References

[1] "Genetic Algorithm" Marek Obitko, 1998 http://cs.felk.cvut.cz/~xobitko/ga

[2] "A Survey of Parallel Genetic Algorithms," E. Cantu-Paz, vol. 10, no. 2,Paris: Hermes, 1998.

[3] "How to Build a Beowulf: A Guide to Implementation and Application of PC Clusters", T. Sterling, J. Salmon, D. Becker, D. Savarese, Cambridge, MA: MIT Press, 1999
.

[4] "Digital Signal Process-ing Algorithms in Condition Monitoring" McCormick A. C., Nandi A. K., and Jack L. B., International Journal of COMA-DEM, vol 1., no. 3, pp 5-14, 1998.

[5] " Classification of Rotating Machine Condition using Artificial Neural Networks" McCormick A. C. and Nandi A. K ,Proceedings of the Institute of Mechanical Engineers, Part C, Vol 11, No. 6, pp 439-450, 1997.

[6] "'Four Problems for which a Computer Program Evolved byGenetic Programming is Competitive with human Performance", KOZA, J.R., BENNETT III, F.H., ANDRE,D., and KEANE, M.A. Proceedings of the 1996 IEEE International Conference on Evolutionary Computation, May 1996,Nagoya, Japan, pp. 1-10.

[7] "Genetic Programming: On the Programming of Computers by Means of Natural Selection', KOZA, J.R. (MIT Press, 1992).

[8] "Neural Networks in Process Fault Diagnosis, IEEE Transactions on Systems, Man, and Cybernetics" Sorsa, T. Koivo H. and Koivisto H., Vol 21,No. 4, 1991.

[9] "Genetic Algorithms in Search, Optimization and Machine Learning" Goldberg G. E., Addison Wesley, New York, 1989.

[10] “Parallel Distributed Processing” Volume 1, 2 and 3 J. L. Mc-clelland & D. E. Rumelhart

[11] "An Introduction to computing with Neural Networks Richard" P. Lippmann. IEEE ASSP Magazine

[12] "Pattern Recognition M. James BSP Professional Books", Oxford, 1987.

[13] "Self Organization and Association Memory", third edition T. Kohonen Springer-Verlag, 1990.

[14] "Organization of Behavior" Donald Hebb. 1949.

[15] "Perceptrons" M. Minsky & S. Papert MIT Press 1969

[16] "Parallel Models of Associative Memory" , second edition G. E. Hinton & J. A. Anderson Lawrence Erlbaum Associate

[17] "Neural Networks and Physical system with Emergent Collective Computational Abilities" J. J. Hopfield In Proc. Natl. Acad. Sci. USA volume 81 1984

[18] "A Learning Algorithm for Boltzman Machines" G. E. Hinton, T. J. Sejnowki & D. H. Ackley. May 1984

[19] "Adaptation in Natural and Artificial Systems" HOLLAND, J.H. (University of Michigan Press, Ann Arbor, Mich., 1975).

[20] "Genetic Algorithms in Search, Optimization, and Machine Learning" GOLDBERG, D.E. (Addison-Wesley Publishing Company, Reading, Mass., 1989).