

Research article

## Prediction models in the design of neural network based ECG classifiers: A neural network and genetic programming approach

Chris D Nugent\*, Jesus A Lopez, Ann E Smith and Norman D Black

Address: Medical Informatics Research Group, Faculty of Informatics, University of Ulster at Jordanstown, Northern Ireland, BT37 0QB

E-mail: Chris D Nugent\* - cd.nugent@ulst.ac.uk; Jesus A Lopez - lopez@nibec-s1.nibec.ulst.ac.uk; Ann E Smith - ae.smith@ulst.ac.uk; Norman D Black - nd.black@ulst.ac.uk

\*Corresponding author

Published: 11 January 2002

Received: 20 July 2001

*BMC Medical Informatics and Decision Making* 2002, 2:1

Accepted: 11 January 2002

This article is available from: <http://www.biomedcentral.com/1472-6947/2/1>

© 2002 Nugent et al; licensee BioMed Central Ltd. Verbatim copying and redistribution of this article are permitted in any medium for any purpose, provided this notice is preserved along with the article's original URL.

### Abstract

**Background:** Classification of the electrocardiogram using Neural Networks has become a widely used method in recent years. The efficiency of these classifiers depends upon a number of factors including network training. Unfortunately, there is a shortage of evidence available to enable specific design choices to be made and as a consequence, many designs are made on the basis of trial and error. In this study we develop prediction models to indicate the point at which training should stop for Neural Network based Electrocardiogram classifiers in order to ensure maximum generalisation.

**Methods:** Two prediction models have been presented; one based on Neural Networks and the other on Genetic Programming. The inputs to the models were 5 variable training parameters and the output indicated the point at which training should stop. Training and testing of the models was based on the results from 44 previously developed bi-group Neural Network classifiers, discriminating between Anterior Myocardial Infarction and normal patients.

**Results:** Our results show that both approaches provide close fits to the training data;  $p = 0.627$  and  $p = 0.304$  for the Neural Network and Genetic Programming methods respectively. For unseen data, the Neural Network exhibited no significant differences between actual and predicted outputs ( $p = 0.306$ ) while the Genetic Programming method showed a marginally significant difference ( $p = 0.047$ ).

**Conclusions:** The approaches provide reverse engineering solutions to the development of Neural Network based Electrocardiogram classifiers. That is given the network design and architecture, an indication can be given as to when training should stop to obtain maximum network generalisation.

### Background

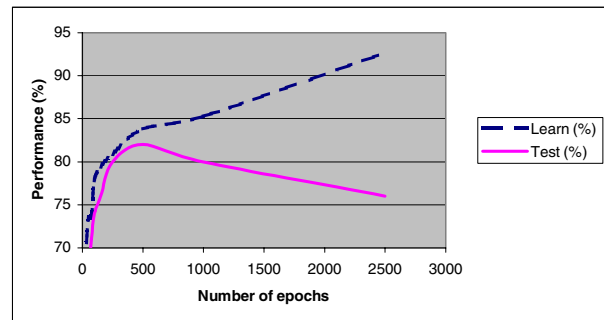
For more than 4 decades, computers have been used in the classification of the Electrocardiogram (ECG) resulting in a huge variety of techniques [1,2] all designed to enhance

the classification accuracy to levels comparable to that of a 'gold standard' of expert cardiology opinion. Included in these techniques are Multivariate Statistics, Decision Trees, Fuzzy Logic, Expert Systems and Hybrid approaches. The

recent interest in Neural Networks (NNs) coupled with their high levels of performance has resulted in many instances of their application in this field [2,3].

In designing an ECG classifier based on NNs, the normal procedure is to firstly train the network by presenting it with training data that is representative of the unknown data it is likely to experience during the classification process. A well-chosen training algorithm, results in a NN which is capable of generating a non-linear mapping function with the capability of representing relationships between given ECG features and cardiac disorders. A well designed NN will exhibit good generalisation when a correct input-output mapping is obtained even when the input is slightly different from the examples used to train the network [4]. In designing a NN, for example a multi-layered perceptron (MLP), the designer must make a number of choices with regard to the system architecture: what is the appropriate number of hidden layers to be included? How many nodes should each layer have? What activation function should be employed and in which configuration? Unfortunately, there is a shortage of evidence available to designers that would enable them to make specific design choices based on a clear scientific rationale and as a consequence, many designs are made on the basis of trial and error. There are other design issues associated with the level and extent of training required for such a network, in particular locating the point at which the network is considered to be sufficiently trained. Conventional methods of training MLPs involve a process whereby the network is trained to the point of minimum error based upon the training data. Subsequently, the network's internal parameters are fixed and it is tested with unseen data to evaluate its performance. This has been the most common approach in the development of NN ECG classifiers [3,5-7]. A danger exists with this approach in that the NN, during training, may memorise the training data. If this becomes the case, then the NN may be biased towards the training data and hence not fully represent the underlying function that is to be modeled. In such instances, poor generalisation is attained when unseen data is presented as input to the network. Such a phenomenon is referred to as over-fitting. Hence it is possible to over-fit the NN if training is not stopped at the correct point.

By employing the 'early stopping method of training' [8] it is possible to test the NN at various stages of training on a validation data set to ensure that over-fitting is avoided. With such an approach it is usual to find that the learning performance of the NN will increase monotonically for an increasing number of epochs in the usual fashion. The validation performance increases monotonically to a maximum, then it begins to decrease gradually as the training continues. With this approach the suggested point of stopping the learning is at the maximum point on the valida-

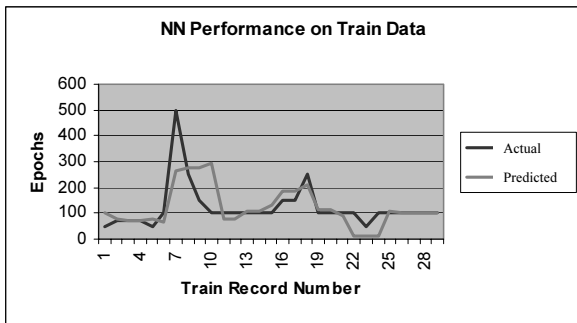


**Figure 1**  
Example of a NN trained with the early stopping method of training depicting the performance with learning and test data.

tion curve. Figure 1 shows an example of a NN trained in this fashion. As indicated in Figure 1 the validation performance increases monotonically to a maximum, occurring at just below 500 epochs. After this point, although the learn performance continues to increase, the validation performance begins to decrease gradually. Thus by employing the early stopping method of training, a network can be trained to a point of maximal generalisation based on a validation set and thus over-fitting is avoided. Although this increases the computational requirements of the NN learning process, benefit is obtained in terms of higher levels of generalisation.

The authors have previously developed a framework for classification of the 12-lead ECG based on a configuration of bi-group NNs (BGNNs) [9]. The framework has the capability to analyse a feature vector comprising approximately 300 features extracted from the 12-lead ECG and classify it into one of a possible 6 diagnostic categories: Inferior Myocardial Infarction, Anterior Myocardial Infarction, Combined Myocardial Infarction, Left Ventricular Hypertrophy, Combined Myocardial Infarction and Left Ventricular Hypertrophy and Normal. Each BGNN in the framework is represented by a single layer MLP with one output node. Each network has the ability to be trained to specifically detect the presence (or absence) of one of the aforementioned diagnostic categories and through a combination matrix an overall classification of the ECG currently under evaluation can be made.

The design procedures, as with similar studies, were largely based on trial and error. This increases the amount of effort required during the design process before the optimal solution can be identified. Additionally, to avoid over-fitting of the BGNNs, the early stopping method of training was employed, further increasing the computational requirements of the design process. Subsequently, a



**Figure 2**  
Comparison between actual and predicted values for the NN based prediction model for training data.

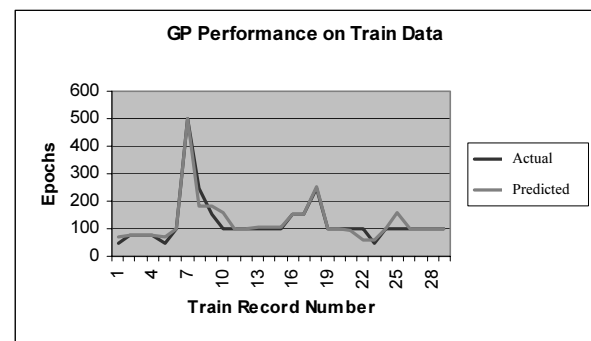
means of predicting a final solution and hence avoiding the aforementioned design constraints can be seen as being largely beneficial.

To address the needs of such a prediction approach to assist in the design process, two methods have been presented; one based on NNs and the other on Genetic Programming (GP). GP can be defined as a search method based on natural selection rules [10,11]. The process begins by selecting an initial set of contending solutions for a particular problem. This set is created by randomly assembling programs from a gene pool of program parts consisting of for example, operators, variables and constants. In order to obtain a good individual or solution (i.e. the program that solves the problem) a number of steps must be taken. Firstly, the set of contending solutions are exposed to the environment or problem to which they are trying to address to determine their 'fitness' i.e. how well they fit the problem at hand. Secondly, the evolutionary process in GP emulates natural evolution in that, the unfit members are removed, fit members remain and new members are generated [10,11]. The entire process is repeated over successive generations until a specific criterion is met and the best solution obtained [12].

The aims of the current study have been to generate two models based on NNs and GP techniques to estimate, given a NN design for ECG classification, the point at which training should stop. The following section describes the methods and approaches of the study.

## Methods

In order to develop the prediction models a number of variable attributes considered to affect the training and generalisation of the NNs were identified. These are variable conditions in the development of the NNs and hence are considered as having potential affects on the location



**Figure 3**  
Comparison between actual and predicted values for the GP based prediction model for training data.

of the point of maximum validation performance. The variables identified were:

1. Number of nodes in the hidden layer ( $n$ ).

(During training various network configurations are evaluated to locate the optimal solution.)

2. Feature Selection method employed ( $fs$ ).

(In an effort to maximise generalisation between classes, various feature selection methods are employed [13]. These aim to reduce the dimensionality of the input to the network, yet still maintain sufficient information to permit discrimination.)

3. Number of files in training set ( $N$ ).

(The content of the training set (i.e. the number of ECG records) affects generalisation. Hence different training set sizes are evaluated.)

4. Size of input feature vector ( $s$ ).

(Depending on the training set size and feature selection method employed, the input feature vector size varies.)

As a final variable, the point at which the network attained maximum performance during training, in the form of the number of epochs ( $m$ ), was also included.

These 5 variables can be considered, in the current study, to potentially contribute to the location of the point of maximum validation performance and hence be used as inputs to the prediction model to give the optimum number of epochs. As mentioned in the introductory sec-

**Table 1: Wilcoxon's signed rank sum results for NN and GP prediction models**

Method	N	Rank -ve Mean	Rank +ve Mean	z-value	2-tailed sig
NN train	29	17.73	13.33	-0.487	p = 0.627
NN test	15	7.00	8.67	-1.023	p = 0.306
GP train	29	13.08	16.56	-1.027	p= 0.304
GP test	15	4.17	10.56	-1.989	p = 0.047

**Table 2: Paired t-test of the mean absolute errors of the prediction models**

Method	MAE	t-value	d.f	2-tailed sig
NN train	40.05	2.859	28	p = 0.008
GP train	11.84			
NN test	36.92	0.057	14	p = 0.955
GP test	35.63			

tion, these are variable factors which will change during the design process, in order to attain the optimal solution.

As a starting point for the current study, only the data collected from the development of the BGNN classifier for the Anterior Myocardial Infarction was analysed. This involved the analysis of the design of 44 different BGNN classifiers developed to specifically classify Anterior Myocardial Infarction. For each classifier a record was created detailing the above 5 design parameters. In addition, the point at which the network attained maximum performance following analysis of the results attained from the early stopping method of training was included. This can be considered to be the desired output of the prediction model and hence be used to compare with the actual output from the prediction model to evaluate how well the model fits the problem. The data was partitioned with two thirds allocated as training data (29 records) and one third as test data (15 records).

Two approaches were investigated to develop the necessary prediction model: a NN approach and a GP approach.

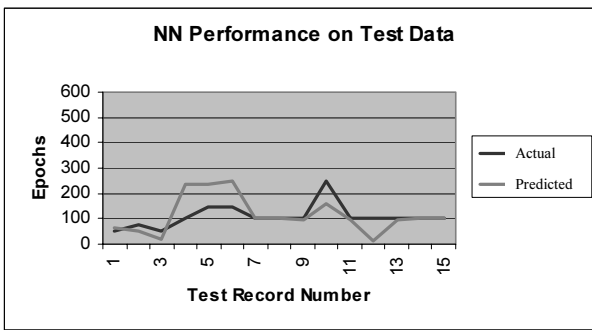
**NN Approach to the Implementation of the Prediction Model**

In an effort to develop a suitable prediction model, a NN based system has been employed. An MLP NN topology was adopted for the prediction model. During the development of the prediction model various architectures (in

terms of the numbers of hidden layers and neurons in each layer) were developed and evaluated. The back propagation training algorithm was employed. The input layer of the NN had 5 neurons, one neuron for each of the aforementioned variable design parameters. A single neuron was used in the output layer with a sigmoidal activation function. The output from this neuron was linearly de-normalised to produce a value for the predicted number of epochs to stop training in the range 0–5000. Following assessment of the different neural classifiers developed, in terms of comparison between desired and actual outputs, the optimal NN generated for the required prediction model had a 5-4-1 architecture. Results generated are presented in the next section.

**Genetic Programming Approach to the Implementation of the Prediction Model**

To further the investigations in terms of the development of suitable prediction models a GP approach was also employed. For the GP prediction model, populations of 3000 individuals were initially evolved and arithmetic functions: add, minus, protected division and product were defined as the function set [10]. A set of random float type constants between 0.0 and 5.0, 0.0 and 50.0 and, 0.0 and 500.0 were defined as the terminal set [10], as well as the aforementioned 5 input parameters (*n*, *fs*, *N*, *s*, *m*). The fitness function (i.e. the measure of how well the model fits the problem) was based on absolute errors for the desired output parameter and the complexity of each individual. Following the evolution process, the individu-



**Figure 4**  
Comparison between actual and predicted values, following exposure to the test data, for the NN based prediction model.

al (solution program) was found with raw fitness of 340.5 and complexity of 127.

**Results**

Figures 2 and 3 indicate the prediction capabilities of both models following training. These graphs indicate the predicted epoch cycles from the NN and the GP models and actual points at which the BGNN attained maximum performance.

Both models follow closely to the actual number of epochs at which maximum performance was attained. The range of values for which the given BGNN attained minimum error following evaluation with training data is in the range of 250–2500, in comparison with the range of 50–500 for the actual values of epochs based on the early stopping method of training and 12–275 for the NN prediction model or 60–500 for the GP prediction model.

Figures 4 and 5 indicate the ability of the prediction models following exposure to the test set of records. Both models exhibit a good level of performance in terms of prediction, indicated by their ability to closely predict the desired range of epochs, at which training should stop. The range of the test data, in terms of epochs at which the BGNN achieved minimum error with regards to the training data, was 150–2500, in comparison with 50–250, 43–222 and 70–500 for the actual epoch value, NN predicted value and GP predicted value respectively.

The data is not normally distributed, so a non-parametric test, the Wilcoxon's signed rank sum test for paired data was utilised. This tests the hypothesis that the two outputs have the same distribution without making any assumptions as to their shape. It was applied to the results for both the NN and GP models for comparison of the desired and predicted results for both training and test data

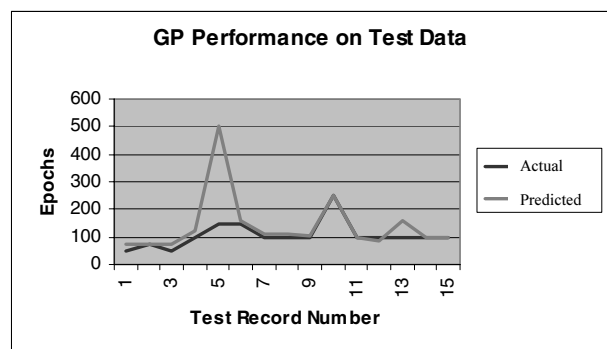
sets; these are given in Table 1 in the standard Wilcoxon's output fashion. These results indicate that there are no significant (sig.) differences with respect to predicted and actual, for both NN and GP methods based on the training data. For the test results, the NN model had no significant differences with respect to predicted and actual while the GP results on the test data are just marginally significantly different ( $p = 0.047$ ).

Another performance measure is that related to the mean absolute error (MAE) of the two train and the two test cases (NN and GP). These can be tested with each other by a t-test for paired samples for the means of the differences, given the degrees of freedom (d.f.). The results for this are given in Table 2. This indicates that the means of the errors of the training sets are significantly different ( $p = 0.008$ ), indicating that the GP training performs significantly better. For the test data, the means of the errors are not significantly different ( $p = 0.955$ ), hence the NN and the GP could be considered to perform equally well on the test sets.

Some discrepancy in test results may be accounted for by the difference in approaches. The Wilcoxon's test here indicates that although the GP performs as well as the NN as regards to the overall errors with respect to the test sets (shown here by the t-test), the GP consistently slightly overestimates the predictions. This is evidenced by the difference in mean ranks (4.17 -ve; 10.56 +ve) and, on closer examination of the graph (Figure 5), this effect can just be discerned.

**Discussion and Conclusions**

As shown in the previous section, both the NN and GP model have the ability to a certain extent to predict the epoch number at which the BGNN should stop training.



**Figure 5**  
Comparison between actual and predicted values, following exposure to the test data, for the GP based prediction model.

Both NN and GP prediction models demonstrated good abilities not only to model the training data as shown in Figures 2 and 3 and indicated in Table 1, but also exhibited good generalisation, Figures 4 and 5. Although the GP displayed a significant level of performance in terms of training in comparison with the NN, both were comparable following evaluation with the test data, with no significant differences in this given study. The range of values for attained minimum error indicates that not only is there a gain in the reduction of the computational intensity of the learning process by being able to indicate when a given network should stop its training, but as previously stated, a gain is also achieved in generalisation.

This study demonstrates that it is possible to generate prediction models to detect the point at which training should stop, in terms of epochs based on variable design parameters. This provides an indication of the point at which maximum performance and subsequently maximal generalisation can be attained and thus optimises these. The models presented in essence provide a means of reverse engineering to the heuristic problem of NN design, in that given the variable parameters of a network architecture and its associated optimal learning performance based on the training data, an indication can be given as to when the network should stop training in order to provide the maximum level of generalisation. This can be seen to be of benefit to developers of NNs, not only in the presented case of NN based ECG classifiers, but indeed any classification problem. Such an approach alleviates the intensity of a trial and error process to NN design and additionally ensures good generalisation.

Further work is planned to develop prediction models for the remaining BGNNs classifying different diagnostic classes and investigate further commonalities and differences in the results generated by both the NN and GP approaches. Initial results have indicated similar findings for the prediction models when applied to different diagnostic classes [14].

### Competing interests

None declared.

### References

- Kors JA, van Bommel JH: **Classification methods for computerized interpretation of the electrocardiogram.** *Methods of Information in Medicine* 1990, **29**:330-336
- Nugent CD, Webb JAC, Black ND, Wright GTH: **Electrocardiogram 2: Classification.** *Automedica* 1999, **17**:281-306
- Bortolan G, Brohet C, Fusaro S: **Possibilities of using neural networks for ECG classification.** *J Electrocardiol* 1996, **29** Suppl:10-16
- Haykin S: **Neural Networks: A Comprehensive Foundation.** Prentice-Hall 1999
- Harrison RF, Marshall SJ, Kennedy RL: **The early diagnosis of heart attacks: a neurocomputational approach.** *Proceedings of the International Joint Conference on Neural Networks* 1991, **1**:1-5
- Endenbrandt L, Devine B, MacFarlane PW: **Neural networks for classification of ecg st-t segments.** *Journal of Electrocardiology* 1992, **25**:167-173
- Baxt WG: **Use of artificial neural networks for the diagnosis of myocardial infarction.** *Annals of Internal Medicine* 1991, **115**:843-848
- Drucker H: **Boosting using neural networks.** In: *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems* 1999
- Nugent CD, Webb JAC, Black ND, Wright GTH, McIntyre M: **An Intelligent Framework for the Classification of the 12-Lead ECG.** *Artificial Intelligence in Medicine* 1999, **16**:205-222
- Koza JR: **Genetic Programming: on the Programming of Computers by Means of Natural Selection.** MIT Press 1992
- Koza JR: **Genetic Programming II: Autonomous Discovery of Reusable Programs.** MIT Press 1994
- Fogel DB: **What is evolutionary computation?.** *IEEE Spectrum* 2000, **37**:26-32
- Nugent CD, Webb JAC, Black ND, McIntyre M: **Bi-dimensional Feature Selection of Electrocardiographic Data.** *Proceedings of the VIII Mediterranean Conference on Medical and Biological Engineering and Computing, MEDICON '98*
- Nugent CD, Lopez JA, Smith AE, Black ND: **Reverse engineering of neural network classifiers in medical applications.** *Proceedings of the VII EFOMP Congress, Physica Medica* 2001, **XVII**:184

Publish with **BioMed Central** and every scientist can read your work free of charge

"BioMedcentral will be the most significant development for disseminating the results of biomedical research in our lifetime."

Paul Nurse, Director-General, Imperial Cancer Research Fund

Publish with **BMC** and your research papers will be:

- available free of charge to the entire biomedical community
- peer reviewed and published immediately upon acceptance
- cited in PubMed and archived on PubMed Central
- yours - you keep the copyright



Submit your manuscript here:

<http://www.biomedcentral.com/manuscript/>

[editorial@biomedcentral.com](mailto:editorial@biomedcentral.com)