One-pass training of optimal architecture auto-associative neural network for detecting ectopic beats

G. Clifford, L. Tarassenko, N. Townsend

We have previously described a method for ectopic beat detection in the electrocardiogram using an auto-associative neural network [1]. We present in this letter a method that utilises Principal Component Analysis to optimise the complexity of the neural network and uses Singular Value Decomposition to determine the initial values for the weights.

Introduction: Effective cardiology requires the identification of significant morphology variations in a patient's electrocardiogram (ECG), a record of the mV fluctuations of the heart's electro-potential. Abnormal beat identification is required for heart rate variability calculations and warnings of the onset of potentially fatal arrhythmias. In an earlier paper [1] we described a method for training an auto associative multi-layer perceptron (AAMLP) to perform QRS detection of normal beats and rejection of ectopic beats [1]. It is trained to reconstruct a *normal* waveform in the ECG (with the main features labelled P-QRS-T as in figure 2), centered on the R-peak as accurately as possible for that particular subject.

Once the AAMLP has been trained, new ECG data is passed, sample by sample, across the input nodes. The trained weights of the AAMLP then perform the auto-associative mapping of this data and produce an output pattern. When the input data is centered on the R-peak, the AAMLP will reconstruct the QRS complex accurately. Otherwise the reconstruction will be poor since the mapping was learned for R-peak centered waveforms only.

Since the QRS complexes are easily located by their R-peaks [2], which occur approximately half way between the start of the P-wave and the end of the T-wave, a 0.5s window centered on the R-peak effectively segments each heart beat. To reduce the input/output dimensionality for the auto-associative network and avoid learning fine (and potentially irrelevant) detail while still preserving the main features, the P-QRS-T waveform is down-sampled from 256Hz to 64Hz. A 0.5 second window then corresponds to 32 input and output nodes.

However, neural network training can be time consuming and the Association for the Advancement of Medical Instrumentation (AAMI) requires that any on-line ECG algorithm must be trained within the first five minutes [3]. There is therefore a trade-off between the number of training patterns that can be collected and the time required for training the AAMLP: the smaller the number of training vectors, the greater the number of iterations through the training set required. Even using a powerful modern PC it is difficult to collect data and train within five minutes. We therefore use Singular Value Decomposition (SVD) to calculate the weights in one pass.

Method: A typical P-QRS-T segment is composed of approximately 10 major turning points. Individual nodes in an AAMLP do not encode individual features, but this number can be used for order of magnitude calculations [4]. The AAMLP encodes the variance in the training set in the same manner as Principal Component Analysis (PCA), projecting the variance onto the same number of orthogonal axes as there are hidden units [4]. A more rigorous method for determining the number of hidden units is to perform PCA on the training set, rank the components in order of magnitude and identify the knee of the curve [6]. Figure 1 shows a knee at 2 components, with significant eigen-vectors up to 8 components. Beyond this any further components contribute very little further information, as they encode noise. In tests on the MIT-BIH database [7] and data collected from a local hospital we found that there was a high degree of inter-patient similarity between singular spectra of training sets, with 8 eigen-components being the upper limit of the boundary between signal and noise. The number of hidden units was therefore set to a value of 8. Figure 2 shows how a typical training set (above) is reconstructed (below) by the first 8 principal components.

For a two layer AAMLP with linear hidden units and a sum-of-squares error function, the weights can be calculated from the pseudo-inverse of a matrix [4]. Consider N training patterns presented to an AAMLP with i input and k output nodes ($i = k$) and $j \leq i$ hidden nodes. For the n^{th} ($n = 1...N$) input vector x_i of the $i \times N$ ($N \geq i$) real input matrix, X, formed by the N (i-dimensional) training vectors, the hidden unit output values are

$$h_j = F(W_1 x_i + w_{1b}) \tag{1}$$

where W_1 is the input-to-hidden layer $i \times j$ weight matrix, w_{1b} is a rank-j vector of biases and F is an activation function. The output of the AAMLP can then be written as

$$y_k = W_2 h_j + w_{2b} \tag{2}$$

where W_2 is the hidden-to-output layer $j \times k$ weight matrix and w_{2b} is a rank-k vector of biases. Now consider the SVD of X, given by [5] as $X_i = U_i S_i V_i^t$ where U is an $i \times i$ column-orthogonal matrix, S is an $i \times N$ diagonal matrix with positive or zero elements (the singular values) and V^t is the transpose of an $N \times N$ orthogonal matrix. The best rank-j approximation of X is given by [8] as $W_2 h_j = U_j S_j V_j^t$ where $h_j = T S_j V_j^t$ and

$$W_2 = U_j T^{-1} \tag{3}$$

with T being an arbitrary non-singular $j \times j$ scaling matrix. U_j has $i \times j$ elements S_j has $j \times j$ elements and V^t has $j \times N$ elements. Furthermore, Bourlard and Kamp [9] show

$$W_1 = \alpha_1^{-1} T U_j^t \tag{4}$$

where W_1 are the input-to-hidden layer weights and α is derived from a power series expansion of the activation function, $F(x) \approx \alpha_0 + \alpha_1 x$ for small x. For a linear activation function, as in this letter, $\alpha_0 = 0$, $\alpha_1 = 1$. The bias weights given by [9] reduce to

$$w_{1b} = -\alpha_1^{-1} T U_j^t \mu_X = -U_j^t \mu_X,$$

$$w_{2b} = \mu_X - \alpha_0 U_j T^{-1} = \mu_X \tag{5}$$

where $\mu_X = \frac{1}{N}\sum_N x_i$, the average of the training (input) vectors and T is set to be the $(j \times j)$ identity matrix since the output is unaffected by the scaling. (Equations (1) to (5) give the output of the AAMLP as $y_k = U_j U_j^t (X - \mu_X) + \mu_X$ where T has canceled).

Results: Equations (3), (4) and (5) represent an analytical solution which determine the weights of the AAMLP in one pass with as few as $i^3 + 6i^2 + O(i)$ multiplications [10]. With $i = 32$ this compares favourably with the usual error back-propagation training algorithm which requires $gNi^2j + O(gNij) + O(gN(i+j))$ multiplications, where g is the number of iterations through the $N \times i$ training set. For a typical value of $N \approx 300$ QRS complexes (approximately 5 minutes of data), g is of the order 2×10^3.

In experiments on patient 109 of the MIT-BIH database [1] where the AAMLP was coded in 'C' and SVD implemented in Matlab, SVD took of the order of 10^{-3} of the time of the back-propagation algorithm to compute the weights. Furthermore, for $N = 300$ the same accuracy was achieved for the network with the back-propagation trained weights; 99.3% of the 2067 normal beats and 31 of the 32 ectopic beats were correctly classified.

The trained weights can be updated on-line with back propagation with $g'i^2j + O(g'ij) + O(g'(i+j))$ floating point operations where g' is the number of iterations to encode the change in variance in the training set due to the new QRS complex. (Typically $g' \approx 10$).

Conclusions: The AAMI allows only five minutes for ECG data collection and classifier training [3]. In a previous paper [1] we described a method for effective identification of ectopic beats in the ECG in real time. However, the number of iterations required to train the algorithm meant that the AAMI guidelines could not be adhered to.

In this letter we have detailed a method for reducing the training time to a single epoch which takes less than one second on most modern PCs. Once the training is complete, the weights can then be updated to adapt to any gradual long term morphological changes.

References

[1] Tarassenko, L., Clifford, G., Townsend, N.: *Detection of Ectopic Beats in the Electrocardiogram Using an Auto-associative Neural Network*, Neural Processing Letters (in press), 2001.

[2] Pan, J. and Tompkins, W.: *A Real-time QRS Detection Algorithm*, IEEE Transactions on Biomedical Engineering, **32**(3) (1985), 230-236.

[3] AAMI ECAR.: *Recommended Practice for Testing and Reporting Performance Results of Ventricular Arrhythmia Detection Algorithms*, Association for the Advancement of Medical Instrumentation, 1987.

[4] Bishop, C.M.: *Neural Networks for Pattern Recognition*, OUP, Oxford, 1995.

[5] Golub, G.H., and Van Loan, C.F.: *Matrix Computations*, 2^{nd} ed. Johns Hopkins Univ. Press, Baltimore 1989.

[6] Ripley, B.D.: *Pattern Recognition and Neural Networks*, Camb. Univ. Press, 1996.

[7] Moody, G.B. : *Harvard-Massachusetts Institute of Technology Division of Health Science and Technology's MIT-BIH arrhythmia database.* Room E25-506A, Cambridge, MA 02139 USA, Updated 16 October 1999.

[8] Golub, G.H.: *Least squares, singular values and matrix approximations*, Applikace Matematiky, 13,44-51, 1968.

[9] Bourlard, H. and Kamp, Y.: *Auto-Association by Multilayer Perceptrons and Singular Value Decomposition*, Biol. Cybern. 59, 291-294, 1988.

[10] Bunch, J.R. and Nielsen, C.P.: *Updating the Singular Value Decomposition*, Numer. Math. 31, 111-129, 1978.

Authors' affiliations

G. Clifford, L. Tarassenko, N. Townsend (Neural Networks Research Group, Department of Engineering Science, Univ. of Oxford, Parks Road, Oxford, OX1 3PJ, UK)

Email: gari@robots.ox.ac.uk

Acknowledgments: The authors would like to thank Oxford BioSignals for their generous support.

Figure captions:

Figure 1 Size-ranked set of eigen-vectors from a SVD decomposition of a training set composed of approximately 200 P-QRS-T waveforms.

Figure 2 Reconstruction (below) of a set of $\approx$ 200 32-D training vectors (above) using the eigen-vectors with the 8 largest eigen-values derived from SVD. P, Q, R, S, and T refer to clinical labels - see [1].

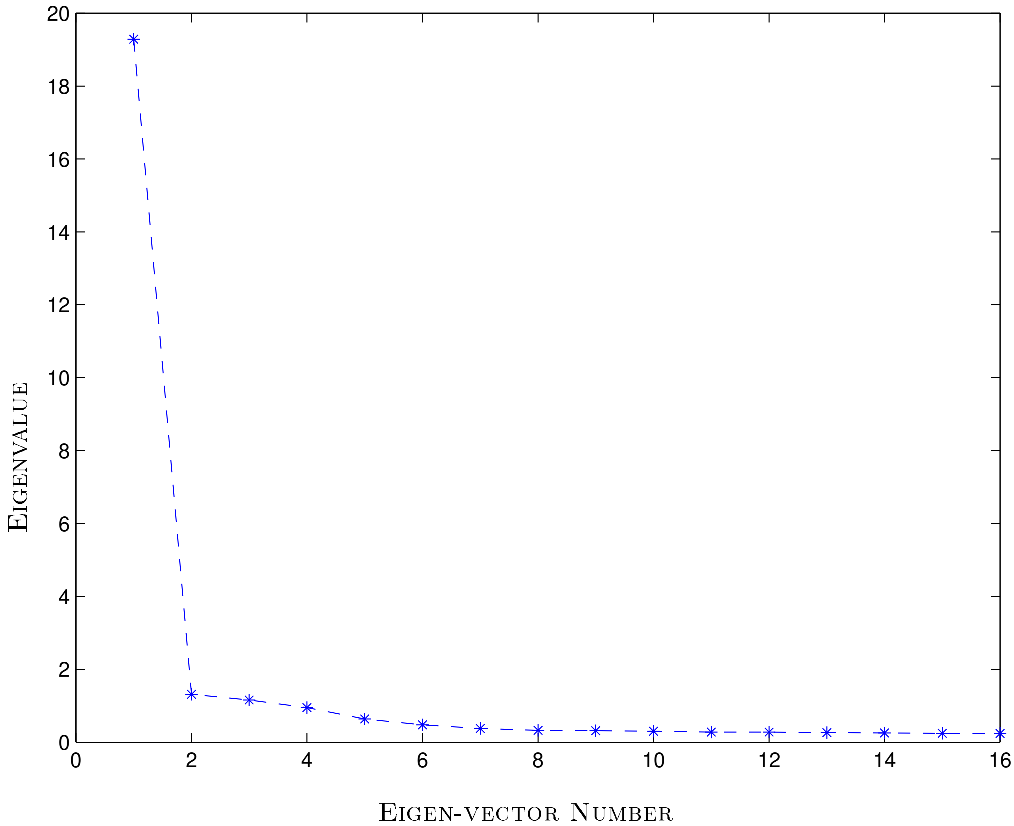
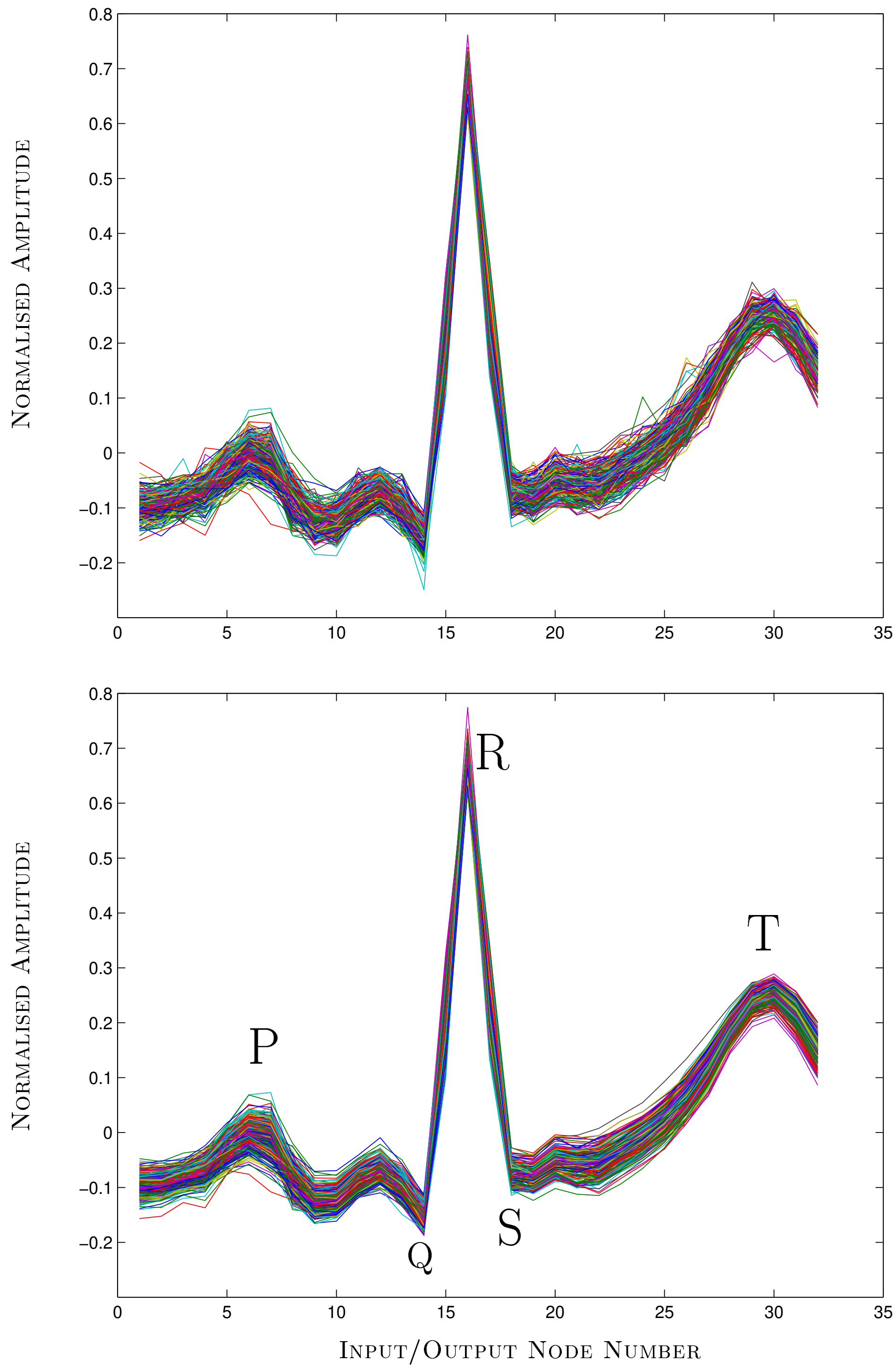


Figure 1:



Normalised Amplitude
Normalised Amplitude
P
Q
R
S
T
Input/Output Node Number

Figure 2: