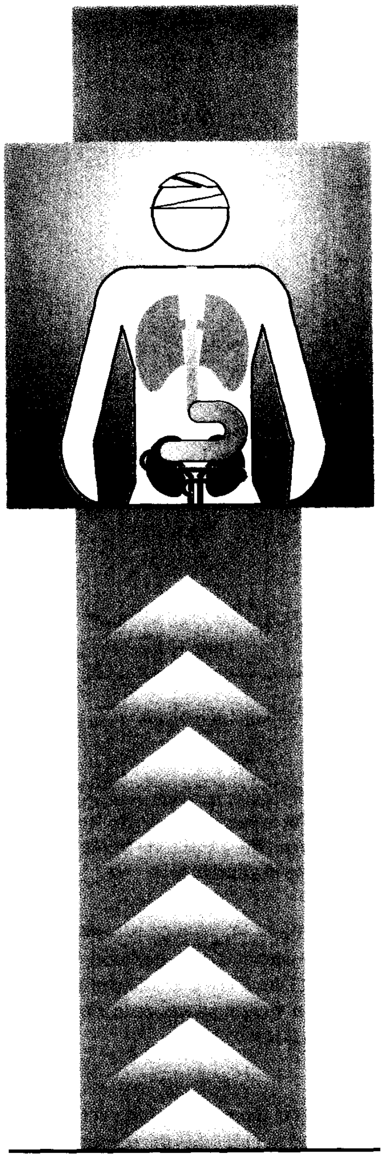
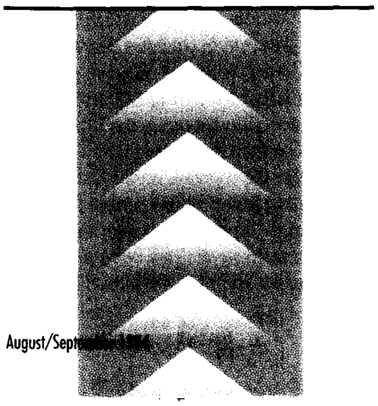


Determining Hearing Threshold from Brain Stem Evoked Potentials

Optimizing a neural network to improve classification performance

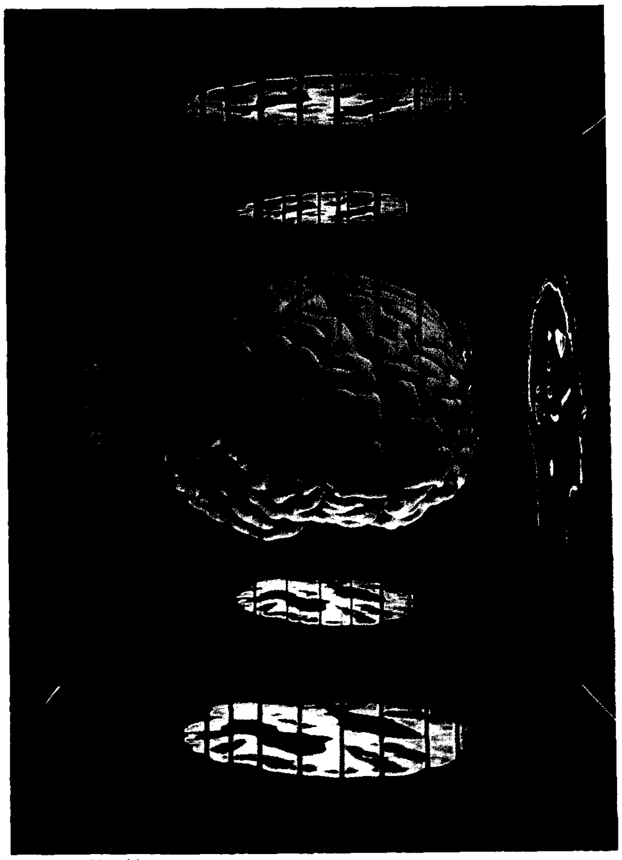
**D. Alpsan[1], M. Towsey[1], O. Ozdamar[2]
A. Tsoi[3] and D.N. Ghista[1]**
[1] Department of Biophysics
Faculty of Medicine and Health Sciences
United Arab Emirates University
[2] Department of Biomedical Engineering
University of Miami
[3] Department of Electrical Engineering
Faculty of Engineering
University of Queensland



Brainstem Auditory Evoked Potentials (BAEPs) are considered the most objective measure currently available with which to determine the functional integrity of the peripheral auditory nervous system. BAEPs are the early portion of the electrical activity of the brain in response to a brief auditory stimulus, typically recorded using electrodes attached to the scalp. A response signal usually consists of seven vertex positive waves within 10 ms of stimulus presentation.

Estimating hearing threshold from BAEP signals is a time consuming and labor intensive procedure, and therefore one which recommends itself to computerized automation. The important step is the classification of the signals into Response (R) and No Response (NR) classes (Fig. 1), the main difficulties being a poor signal-to-noise ratio and the differentiation of response peaks from artifacts. Artificial neural network (ANN) classifiers are an appropriate choice for this type of task because they are tolerant of noise and do not require a prior analytical description of the signal. Nevertheless, BAEP classification is a difficult task, having certain characteristics such as the requirement for good generalisation, unequal numbers in the two class populations, and a significant proportion of signals that even experts have difficulty in classifying. The difficult to classify signals are typically those at the threshold of hearing. In a clinical setting, the physician has the advantage of having a sequence of signals arranged in order of stimulus strength. Thus, it is comparatively easy to classify the difficult threshold signals.

We have already demonstrated the feasibility of using neural networks to classify BAEPs [1], and also that signal preprocessing and careful selection of the



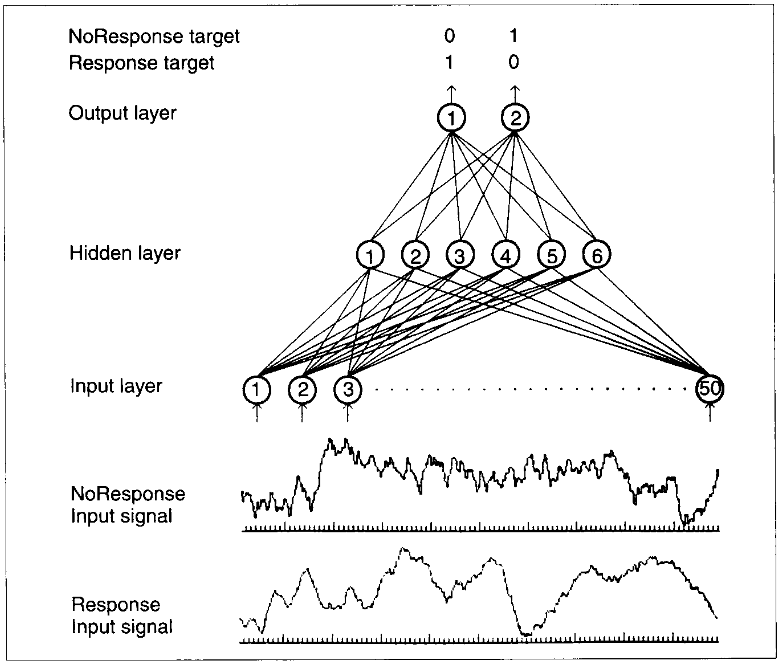
FPG Interntional/©Ralph Mercer 1994

training set can increase the accuracy rate of classification [2]. When experts were presented with solitary signals in the absence of any contextual or clinical clues, they performed only slightly better than a trained neural net classifier. Preprocessing the signals brought the net performance to levels comparable to those of experts. However, in these studies, no attempt was made to optimise the neural net learning parameters; in particular, the learning rate constant (LRC), the momentum constant (MOM) and the batch size.

The aim of this study was to optimize four learning parameters in order to improve the classification performance of the net on unprocessed signals as seen by experts, perhaps to a level equal to the experts with contextual clues. Our results show that proper tuning of learning parameters not only increases the speed of learning, but can also increase generalisation and reduce the occurence of false negative classifications. This last effect is of considerable importance in a clinical environment.

ANNs are computational systems whose internal structure and processing methods attempt to "imitate" some of the known features of biological nervous systems. As such, they have properties and capabilities quite different from those of traditional serial algorithms processed by serial computers. Indeed, the recent surge of interest in ANNs has come about because they can solve problems that are intractable with traditional serial methods [3]. They are rapidly finding many applications within medical science, not only as signal and image processors but also for computer-aided diagnosis [4].

An ANN is a network of simple computational elements known as neurons or units. Any ANN is defined by three features, its network architecture (the number of units and their pattern of interconnection), the input-output function of the units, and a learning rule by which the connection strengths or weights between units are changed. In particular, a three-layer (input, hidden, and output layers) feed-foward architecture in conjunction with the back-propagation learning algorithm has proved successful in many classification tasks, including biomedical signal classification. The network is required to learn a set of input patterns or signals (that is, produce correct target outputs when presented with each input pattern) by changing the weights between units. The net compares its actual outputs with target outputs and minimises the difference by adjusting its weights appropriately. Learning is terminated when the net



1. Sample BAEP signals and network architecture. The signal labeled "No Response" does not contain sound evoked activity, whereas the amplitude fluctuations of the signal labeled "Response" are the result of a sound stimulus.

can correctly classify all input patterns within a predetermined error tolerance.

From a mathematical point of view, the backpropagation algorithm is a non-linear least squares optimisation problem, whose solution is approached iteratively. The non-linearity in this case is a sigmoid function (either the logistic or tanh) chosen because it is thought to mimic the output function of a typical biological neuron. The error function to be minimised is known as the quadratic:

$$E = \frac{1}{2}\sum_{p}\sum_{k}(d_{pk} - o_{pk})^2 \qquad (1)$$

where p is the number of patterns in the training set, k is the number of output layer neurons, d is the desired or target value for any given output neuron presented with pattern p, and o is the actual output of that neuron. When the net is presented with an input signal, unit outputs are fed foward through successive hidden layers until output values are obtained for the units of the output layer. The output errors are calculated as in equation (1) and back-propagated through the network, so that an output error is assigned to each hidden unit. The network weights are adjusted according to the learning rule:

$$\Delta \mathbf{w}_n = -\eta \nabla E(\mathbf{w}_n) \qquad (2)$$

where $\Delta \mathbf{w}_n$ is the change in value of the weight vector at the nth iteration, and η is the LRC. *DELE* is the gradient of the error surface in weight space, each element of which is given by:

$$\partial E / \partial w_{ji} = \delta o_i$$

where w_{ji} is the weight of the connection from unit i to unit j, δ is the output error of unit j, and o_i is the output of unit i. Once training is complete, the ability of the net to generalise is tested by presenting a set of novel signals, that is, signals not seen by the net during training but drawn from the same pattern space.

Despite its popularity, one difficulty with the original backpropagation algorithm has been its slow rate of training convergence. Increasing the LRC will speed learning, because larger steps mean fewer need be taken to reach a solution. However, too large a value of LRC makes learning unstable In practice, learning can be slow even with the largest possible values of LRC. Many modifications have been proposed to speed up back-propagation. One, which produces faster learning, is to change the weights immediately after each training pattern is presented (pattern-mode weight updating) rather than after the entire training set has been presented (batch-mode weight updating). The addition of a momentum term to the calculation of weight changes is the most frequently used method to increase speed of convergence. It is usually incorporated into the learning rule as follows:

$$\Delta \mathbf{w}_n = -\eta \nabla E(\mathbf{w}_n) + \alpha \Delta(\mathbf{w}_{n-1}) \quad (3)$$

where α is the momentum constant (MOM). Momentum is now so widely used that it can be considered a part of the standard back-propagation algorithm. The value of MOM (usually between 0.0 and 0.9) determines the relative contribution of the previous gradient to the current weight change. Momentum smooths high frequency fluctuations of the error surface in the weight-space [5]. It improves the speed of convergence by augmenting weight changes when consecutive changes have the same sign, and by damping when they have alternating signs. For many problems, increasing MOM not only speeds up learning but also reduces the variability of learning times [6, 7].

While it is apparent from many studies that momentum can accelerate learning, there are few published reports on the interaction of momentum with other learning parameters or its effect on other performance indices, such as generalisation— the ability of a net to classify correctly novel inputs. In medical problems, it can be argued that generalisation is more significant than faster learning.

Generalisation is affected by training set selection [2], training set size [8], noise added to the training signals [9], and hidden layer size [9]. As a preliminary to their study on the relationship between generalisation and training set size, Cohn and Tesauro [8] state that no other training parameters apart from batch size (number of signals presented to the net before weights are updated) had a significant effect on generalisation. Sietsma and Dow [9] found that pruning a network down to the minimum number of hidden layer units that could correctly classify the training set produced networks which generalised poorly. The results presented here reveal a significant interaction between the four learning parameters (LRC, MOM, batch size, and hidden layer size) with respect to both learning speed and generalisation and that a network tuned for speed may yield poor generalisation.

The Signal

Raw BAEPs were amplified and bandpass filtered (100-3000Hz) to remove the EEG component and high frequency noise. A post stimulus signal of 12.8 ms was sampled at 40 kHz to give 512 data points. Since these raw signals are extremely noisy, standard procedure was to coherently average 1024 of such signals to give a single BAEP signal. This signal can be used for classification but in this study, the signals were further reduced by sampling every eighth value between 1 ms and 11 ms. The resulting signal of 50 data points was normalised between 0 and 1 and used as input to the neural network. A data set of 321 such input signals was obtained, which included various combinations of hearing impaired and normal subjects and varying stimulus intensities.

The Training and Test Sets

The training set consisted of 60 signals (45 response signals (R) and 15 no response signals (NR) selected by experts as being typical of their class. Previous studies had shown that an expert selected training set was superior to a randomly selected training set [1]. The ratio of class sizes (the R:NR ratio) in the training set was chosen as 3:1, reflecting the approximate ratio in a clinical setting.

The test set consisted of 261 signals with the same ratio of three R signals to one NR signal. No signals from any of the same subjects used in the training set were included, which added considerably to the difficulty of the learning task. The test set was presented to the net usually every 100 iterations and finally at convergence in order to monitor generalisation.

Standard Net Configuration

When optimising a net's performance, the number of possible parameter combinations is enormous. Not all can be tested, yet much care must be taken in drawing general conclusions from a few combinations because the interactions between parameters are complex. We therefore defined a standard net configuration and changed parameters one at a time from the standard configuration to test the effect of changing that parameter. The standard configuration was a three layer net with 50 input units, 6 hidden layer units, and 2 output units (a 50-6-2 net). Each unit had a bias weight and used the binary logistic output function. The net was initialised with random weights in the range [-0.5, +0.5]. The quadratic error function was used to measure learning error. The targets for the two output units were 1, 0 and 0, 1 for the R and NR signals, respectively (Fig. 1). The standard training batch size was 60, that is weights were updated after each complete presentation of the training set, which was counted as one learning iteration. Training was terminated either when all signals had been learned within a tolerance of 0.2 (for each output unit) or when the number of iterations equaled 10,000.

Performance Indices

In this study, we were primarily concerned with three indicators of net performance, time to learn the training set, generalisation, and learning stability. Learning time was measured as the number of iterations required for the net to converge on a solution at the given level of error tolerance. Speed of learning is inversely proportional to the number of iterations.

Generalisation is the ability of a net to classify correctly inputs it has not seen during training. In this study, generalisation was measured as the percentage of correctly classified signals in a test set of 261 signals.

The *stability* of learning for a given set of learning parameters is the ability of the net to converge, starting from different points in the weight space; that is, starting with different sets of initial random weights. Another measure of stability is the standard deviation of the iterations required for convergence. A net configuration that results in large SD of iterations is less stable.

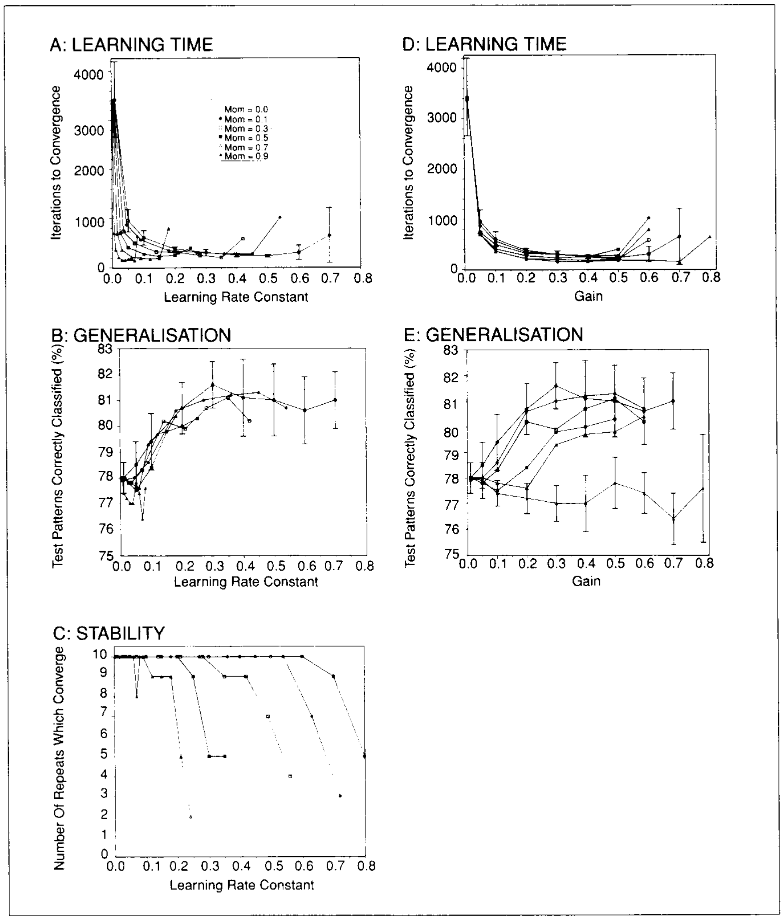
Learning was repeated 10 times for each combination of parameters (10 repeats = 1 trial), the net being initialised with a different set of random weights for each repeat. To enable more accurate comparisons, the same 10 sets of random weights were used for every trial. Where a run did not converge within 10,000 iterations, training was automatically terminated and the next repeat started. Averages and standard deviations of performance measures were calculated only for those repeats which converged within the 10,000 limit.

Plotting performance indices against LRC yields graphs that are difficult to interpret. Consequently, performance indices were plotted against gain, where gain = LRC/(1 - MOM). By so doing, the smoothing effect of the momentum term can be clearly disentangled from its contribution to gain.

Results

Interaction of Learning Rate Constant and Momentum

We trained a 50-6-2 network using different combinations of the parameters, LRC and MOM. Learning time and generalisation are shown in Figs. 2a and 2b, respectively. For any given value of MOM, learning time first decreased and then increased as LRC was progressively



2. Learning time, generalisation and convergence stability. The graphs in A, B and C are plotted against the LRC for different values of MOM. The graphs in D and E are plotted against gain for different values of MOM. For figures A, B, D and E, only those parameter combinations are plotted where 8 or more of the 10 repeats converged within 10,000 iterations. Error bars indicate ± one SD. For clarity, error bars are included only for selected traces.

increased. Variability of learning time (standard deviation of 10 repeats) also declined at first, and then increased. The optimum value of LRC (for which learning time was a minimum), became smaller as MOM was increased. Increasing MOM also reduced the range of LRC values over which learning remained stable. This is also apparent in Fig. 2c, which shows how the stability of learning (number of repeats out of 10 which converged within 10,000 iterations) varies for various combinations of LRC and MOM.

There was a significant increase in generalisation to a maximum of 81.5 percent as LRC was increased from 0.01 to 0.3, but it remained unchanged for higher values of LRC. Note that high values of MOM prevented the use of higher LRC values, which resulted in higher generalisation.

When network performance was plotted against gain (Figs. 2d and 2e), it was clear that gain has the dominant influence on learning time (Fig. 2d), although use of momentum still significantly reduced learning time. Instability problems arose with gains higher than 0.6.

Generalisation (Fig. 2e) at low gain was 78 percent for all values of MOM. With low MOM (0.0, 0.1), generalisation increased to about 81 percent as gain was increased to 0.3, but did not change significantly as gain was increased further. For intermediate values of MOM (0.3, 0.5, 0.7), generalisation increased with gain, but more slowly than with low MOM. With maximum MOM (0.9), generalisation remained low for all values of gain.

Batch Size

A 50-6-2 net was trained with batch sizes of 1, 4, 20, 30, and 60 (Fig. 3). A batch size of 1 is equivalent to training in pattern mode (weight updates after each pattern is presented), while a batch size of 60 was equivalent to weight updates after presentation of the entire training set. Once again, the dominant trend was for learning time to decline with increasing gain. Decreasing the batch size also decreased the learning time, but the effect was significant only with low momentum (0.1). The major effect of batch sizes smaller than 60 was to enable stable learning at gains higher than 0.5.

Batch size had a significant effect on generalisation. Once again, generalisation was low (78 percent) at low gain regardless of batch size. Generalisation increased with gain, most rapidly for batch size of 60, at a slower rate for batch sizes of 30 and 20, and not at all for batch sizes of 1 and 4. The net result is that for intermediate levels of gain, generalisation tended to increase as batch size increased. Using high momentum (0.9), generalisation did not change significantly with increase in gain or increase in batch size.

Hidden Layer Size

The combined effects of hidden layer size, gain, and momentum on learning time and generalisation are shown in Fig. 4. With MOM = 0.1, learning time decreased and generalisation increased as the hidden layer size was increased. Increasing the gain for any given hidden layer size increased learning speed and generalisation. There was an interesting interaction between hidden layer size and gain with respect to generalisation. With higher gain, the net required fewer hidden units to achieve optimum generalisation. For example, the optimum performance with gains of 0.1, 0.3, and 0.5 (and MOM = 0.1) was reached with hidden layer sizes of 8, 6, and 3, respectively.

When high momentum was used (MOM=0.9), learning time and generalisation did not significantly differ over the whole range of hidden layer sizes. In particular, generalisation remained at the same low level (around 77 percent). Trials were also run with MOM values of 0.0 and 0.5 (not shown in the graphs to preserve clarity). Generalisation scores for MOM = 0.5 were intermediate between those of low and high momentum.

Discussion

Interaction of Learning Rate Constant and Momentum

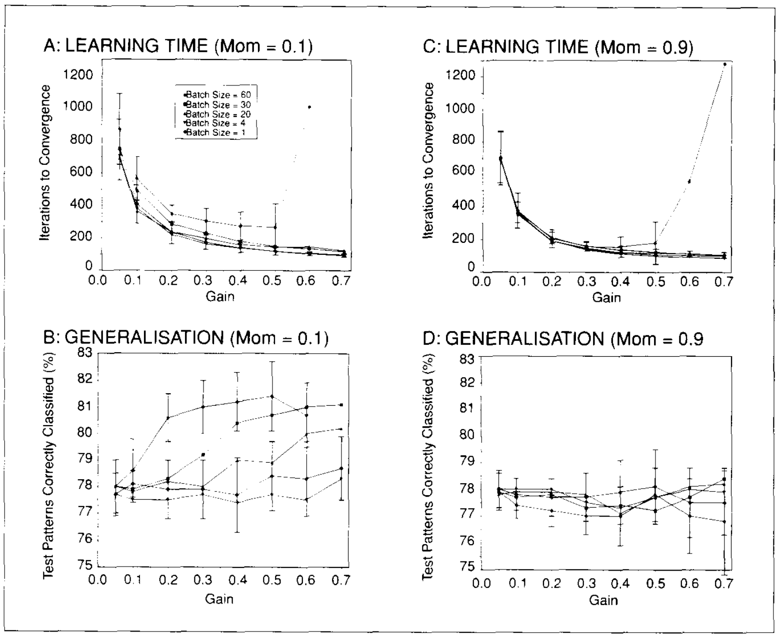
The learning time curves shown in Fig. 2a are similar to those obtained by Tollenaere [6] and Higashino, *et al.* [10]. The similarity is of interest because both these former studies used artificial bench mark tests, whose validity with respect to real world data (such as our BAEP data) is sometimes questioned.

While it is true that the optimum combination of LRC and MOM is very much problem specific and depends further on training set size, hidden layer size, etc., there has been unnecessary confusion in the literature over the interaction of these two learning parameters. The confusion arises because *both* LRC and MOM in the standard learning rule contribute to the magnitude of the weight changes. Momentum is usually incorporated into the back-propagation algorithm, as in Eq. 3. The problem with this formulation is that the momentum term has the additional effect of increasing the overall value of the weight changes and, therefore, the gain of the system. Momentum can also be incorporated into the weight update formula as follows:

$$\Delta \mathbf{w}_t = -(1 - \alpha)\eta_o \nabla E(\mathbf{w}_t) + \alpha \Delta \mathbf{w}_{t-1} \quad (4)$$

In this formulation, adding momentum does not increase the system gain, and η_o can be viewed as the total gain of a cascade of an integrator and a low pass filter. The integrator, which sums individual gradients over time, has a gain of η, and the low pass filter has a gain of $1/(1 - \alpha)$ [7]. Equations 3 and 4 are equivalent with gain $= \eta_o = \eta/(1 - \alpha)$. Rather than considering combinations of LRC and MOM, it is more useful to consider combinations of gain and MOM because we would expect learning speed to be proportional to gain. Indeed, Higashino, *et al.* found empirically that N is proportional to $(1 - \alpha)/\eta$, where N is the number of iterations required to converge on a solution, which is the inverse of learning speed. While high gains provide faster learning, training tends to become unstable.

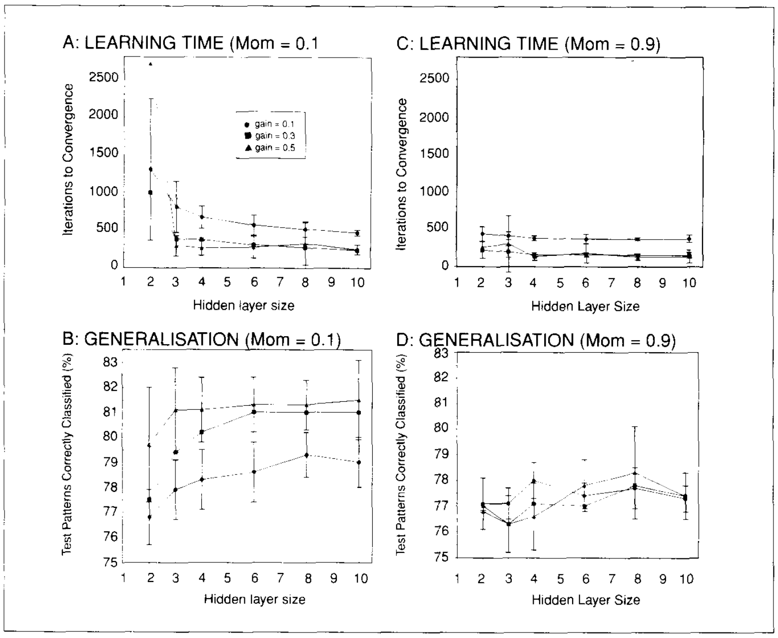
If learning speed were the only consideration, then high gain and high momentum (within the limits of stable learning) would be the most desirable combination. However, high momentum tends to inhibit generalisation, as shown in Fig. 2e, a result that is not immediately obvious when generalisation is plotted as a function of LRC (Fig. 2b). Taking both speed and generalisation into account, the optimum parameter values for our problem were: gain = 0.3 and MOM = 0.0, which gave both fast learning and good generalisation. The important point to emerge from these results is that parameter combinations that lead to fast learning do not necessarily lead to the best achievable generalisation.



3. Learning time and generalisation as a function of gain for different batch sizes. A and B: MOM = 0.1; C and D: MOM = 0.9.

In our original studies [1, 2], no investigation was made of the optimum LRC or MOM for the classification task. It turns out that the values used (LRC = 0.01, MOM = 0.9) are the worst combination for both speed and generalisation. By optimising LRC and MOM, average generalisation was significantly increased from 77 percent to 82 percent. The best individual generalisation value achieved was 84 percent.

While this may be considered a rather modest increase in generalisation, these total figures disguise another important



4. Learning time and generalisation as a function of hidden layer size. A and B: MOM = 0.1; C and D: MOM = 0.9.

consequence of optimising for LRC and MOM. In the classification of BAEP signals, one is concerned to minimise the occurence of false negatives, that is the misclassification of "no response" signals as "response" signals. This concern is undesirable because it increases the probability of a hearing impaired person not being admitted for further testing. Analysis of the generalisation test scores revealed that increases in generalisation with gain were achieved by an improved recognition of NR signals. As gain increased from 0.01 to 0.6 (with MOM = 0.0), correct recognition of NR signals increased from 40 to 70 percent, a value which is slightly better than experts in the absence of clinical and contextual clues [1].

The use of momentum is obviously detrimental to good generalisation in this particular classification task. Pedone and Parisi [11] state that too high a value for momentum "can block learning," but this statement must be interpreted carefully. Our results confirm that the use of momentum significantly increases the speed of learning a training set, provided that gain is kept low enough to avoid unstable learning. Furthermore, a momentum value of 0.9 helped to stabilise learning with high gains. In other words, momentum enhanced the learning of our training set. Rather, it was generalisation to unseen inputs that declined with the use of momentum, a phenomenon that Pedone and Parisi did not investigate.

The interactions between LRC and MOM in our study illustrate the difficulty in drawing general conclusions from a limited sample of parameter combinations. It would have been possible in our case, using a wide range of MOM with a single low value of LRC and a wide range of LRCs with a single high value of MOM, to conclude incorrectly that neither variable had an effect on generalisation. The optimum combination of LRC and MOM may of course depend on other factors such as network size and training set size. Nevertheless, our results suggest that for optimum generalisation, momentum should be kept low or not used at all.

An increase in step size means fewer steps to convergence and thus faster learnng, but it is more difficult to explain why a larger step size should increase generalisation. During gradient descent, the value of the error function gradually decreases, but this value does not in itself determine when training stops. Rather training stops when all input patterns are correctly classified within a predetermined level of output unit error. We have found that final error, that is the value of the error function at convergence, is significantly reduced when higher values of gain are used. The translation of reduced final error into higher generalisation depends on other parameters such as momentum and batch size. With low momentum, low final error correlates with higher generalisation. With high momentum, generalisation remains poor, regardless of the final error value.

Batch Size

Decreasing the training batch size significantly decreased learning times but also reduced generalisation (at low momentum). In this respect, small batch size had the same effect on network performance as high momentum. Observe, for example, the similarity between Figs. 2e and 3b.

In Fig. 3, a batch size of 1 is equivalent to what is usually called pattern mode weight updating. There are two versions of pattern mode weight updating; either the patterns can be presented in random order or in sequential order. In Fig. 3, the patterns were presented sequentially. We had previously compared sequential pattern mode with random pattern mode and found no difference in learning times or generalisation. Pattern mode is usually adopted in preference to full batch mode because it gives a faster rate of learning [12]. Our results suggest that in tasks where generalisation is important, both versions of pattern mode should be avoided, despite their faster training times.

Our results confirm the statement of Cohn and Tesauro [8] that batch size has an effect on generalisation. But contrary to our results, they further state that LRC and momentum had no effect on generalisation. They were primarily interested in the influence of training set size on generalisation and investigated other parameters only as possible sources of systematic error. It is probable that, like many other aspects of neural network performance, interactions between parameter values are very much task specific.

Hidden layer size

It is frequently stated that a training set learned with the minimum of hidden layer units leads to better generalisation [13]. Intuitively, one might expect that excess hidden units will learn idiosyncratic features of the training set that are not representative of the pattern space as a whole. However, this intuition is not supported by several studies. Sietsma and Dow [9] report that best generalisation occured when more hidden units were used than the minimum required to learn the training set. Our results lead to the same conclusion, but we have additionally observed that the optimum number of hidden layer units depended on gain. In the absence of momentum, fewer hidden layer units were required with higher gains, but increasing the number of hidden units could not compensate for the use of low gain. In other words, high gain is more important than large hidden layer size to achieve best generalisation. The better generalisation with large hidden layer size is due, once again, to improved recognition of the NR signals.

The use of high momentum completely negates the advantage of larger hidden layer size. (Sietsma and Dow [9] used MOM = 0.5 and LRC = 5/fan-in, which makes detailed comparisons more difficult.) Our results lead to the conclusion that the optimal hidden layer size for generalisation cannot be determined independently of the optimal gain and momentum. The interaction between gain and hidden layer size can be reduced, however, if the gain of each unit in the net is normalised for its connectivity [10].

It was noticed during training with large hidden layer sizes that several of the hidden layer units did not appear to learn, that is, their weight changes were very small compared to the units which were obviously learning pattern features. These excess or redundant hidden units did not, however, impair generalisation. We found no diminished generalisation even with a hidden layer size of 10.

Conclusions

Feed-foward neural networks in conjunction with back-propagation are an effective tool to automate the classification of biomedical signals. Most of the neural network research to date has been done with a view to accelerate learning speed. In the medical context, however, generalisation may be more important than learning speed. With the BAEP classification task described in this study, we found that parameter values that gave fastest learning could result in poor generalisation. In order to achieve maximum generalisation, it was necessary to fine tune the neural net for gain, momentum, batch size, and hidden layer size. Although this maximization could be time consuming, especially with larger training sets, our results suggest that fine tuning parameters can have important clinical consequences, which justifies the time involved. In our case, fine tuning parameters for high generalisation had the additional effect of reducing false negative classifications, with only a small sacrifice in learning speed.

Dogan Alpsan received the B.Sc. degree in physics from the Middle East Technical University (METU, Ankara, Turkey) in 1969. He received his M.Sc. degree in biophysics and bioengineering from the University of London and his Ph.D. degree in biology from METU. Until 1988, he was with the faculty in the Biology Department, METU, where he was involved in the study of somatosensory evoked potentials and selective activation of nerve fibres. From 1988 to 1989 he was a Fulbright fellow in the Department of Biomedical Engineering, University of Miami. Subsequently his research interests have extended to include automated recognition of medical signals by intelligent computing systems, and organ systems modelling. Currently he is with the Faculty of Medicine and Health Sciences, United Arab Emirates University.

Michael Towsey received his B.Sc. Hons in Biology from Auckland University (New Zealand) in 1970. He completed his M.Sc. in crop physiology in 1983 and has worked both in agricultural research and as a science teacher. In 1990, he shifted to the Biophysics Department in the Faculty of Medicine and Health Sciences of the United Arab Emirates University. Here, his research interests have extended to the automated recognition and processing of biomedical signals, and to computer aided learning (CAL). He is currently pursuing a Ph.D. degree in the Department of Electrical Engineering of Queensland University (Brisbane, Australia) on the classification of EEG signals using neural networks.

D.N. Ghista gained his Ph.D. degree from Stanford University in 1964. Currently, he is Professor and Chairman of the Biophysics Department at United Arab Emirates University. Previously, he had been Professor of Medicine and Engineering Physics, and Chairman of Biomedical Engineering at McMaster University, Canada. In recent years, he has also been involved in the indigenous development and industrialization of advanced medical technologies. His interests embrace the broad field of biomedical and health engineering. These include medical diagnostic indices and systems; customized prosthetic and rehabilitation systems; simulation and customized guidelines in cardiovascular, orthopaedic, gastrointestinal, and urological surgery; theory of microvita in disease and evolution process; and design of hospital and community health systems. Professor Ghista is the American editor of Automedica. He has published numerous papers in the fields of biomedical, engineering and societal sciences. He is also author/editor of several books on biomedical physics, physiology, cardiovascular physics and orthopaedic mechanics. **Professor Ghista can be reached at the Department of Biophysics, Faculty of Medicine and Health Sciences, United Arab Emirates University, PO Box 17666, Al Ain, United Arab Emirates.**

O. Ozdamar received his B.S. degree in 1971 in Electrical Engineering from Middle East Technical University in Ankara, Turkey. He obtained his M.Sc. and Ph.D. in Biomedical Engineering from Northwestern University, Evanston, Illinois, USA in 1973 and 1976, respectively. He is currently a Professor of Biomedical Engineering at the University of Miami, Coral Gables, Florida, USA, with secondary appointments in the departments of Otolaryngology and Pediatrics of the School of Medicine there. His current interests are applications of artificial neural networks to EEGs, EP and speech processing, intelligent biomedical devices, hearing prostheses for the deaf, and automated electrophysiologic testing of hearing.

Ah Chung Tsoi was born in Hong Kong. He received a Higher Diploma in Electronic Engineering, Hong Kong Technical College, in 1969, M.Sc. in Electronic Control Engineering, and Ph.D. in Control Engineering, from the University of Salford, in 1970 and 1972 respectively. He received a BD in Theology from the University of Otago, in 1980. From 1972 to 1974, he worked as a Senior Research Fellow at the Inter-University Institute of Engineering Control, University College of North Wales, Bangor. From 1974 to 1977, he was a Lecturer at the Paisley College of Technology, Renfrewshire, Scotland. From 1977 to 1984, he worked as a Senior Lecturer at the University of Auckland, New Zealand. From 1985 to 1990, he was a Senior Lecturer at the University College, University of New South Wales. He is presently a Professor in Electrical Engineering, University of Queensland. His research interests include aspects of neural networks, and their application to practical problems; adaptive signal processing; adaptive control.

References

1. **D Alpsan and O Ozdamar:** Auditory brainstem evoked potential classification for threshold detection by neural networks. I. Network design, similarities between human-expert and network classification, feasibility, *Automedica* 15, 67-82 (1992).

2. **D Alpsan and O Ozdamar:** Auditory brainstem evoked potential classification for threshold detection by neural networks. II. Effects of input coding, training set size and composition and network size on performance, *Automedica* 15, 83-93 (1992).

3. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.*, DE Rumelhart, JL McClelland Eds, MIT Press Cambridge, (1986).

4. **AS Miller, BH Blott and TK Hames:** Review of neural network applications in medical imaging and signal processing, *Med. & Biol. Eng. & Comput.* 30, 449-464 (1992).

5. **DE Rumelhart, GE Hinton and RJ Williams:** Learning internal representations by error propagation, in Parallel Distributed Processing: Explorations in the Microstructure of Cognition., *Foundations*, 1, 318, DE Rumelhart, JL McClelland Eds, MIT Press Cambridge, (1986)

6. **T Tollenaere:** SuperSAB: Fast adaptive back propagation with good scaling properties, *Neural Networks* 3, 561-573 (1990).

7. **HAC Eaton and TL Olivier:** Learning coefficient dependance on training set size, *Neural Networks* 5, 283-288 (1992).

8. **D Cohn and G Tesauro:** How tight are the Vapnik-Chervonenkis bounds?, *Neural Computation* 4, 249-269 (1992).

9. **J Sietsma and RJF Dow:** Creating artificial neural networks that generalize, *Neural Networks* 4, 67-79 (1991).

10. **J Higashino, BL deGreef and EHJ Persoon:** Numerical analysis and adaption method for learning rate of back propagation, *Proc. IJCNN*, Washington D.C., 1 627-630 (1990).

11. **R Pedone and D Parisi:** Learning the learning parameters, *Int. Joint Conf on Neural Networks* (Singapore)2033-2037 (1991).

12. **Y Lee and RP Lippmann:** Practical characteristics of neural network and conventional pattern classifiers on artificial and speech problems, *Advances in Neural Information Processing Systems* Vol 2, ed. D.S. Touretzky (1990).

13. **JK Kruscke:** Improving generalisation in back-propagation networks with distribution bottlenecks, *Int. Joint Conf. Neural Networks,* San Diego, I, 443-447 (1989).