A Multi-Computer Neural Network Architecture

R.J.Howlett and S.D.Walters

Abstract: A novel neural network architecture is presented which allows the efficient execution of the back-propagation learning algorithm on a multi-computer system which has a low communications bandwidth.

The Class-Distributed Neural Network: The multi-layer perceptron, which uses the back-propagation learning algorithm, is widely used as a classifier because of its good generalisation properties and compact internal structure. However, its convergence rate is poor, leading to extended training times. Multi-computer systems have appeared to be attractive platforms for the implementation of neural network algorithms, offering the potential for achieving faster convergence through increased processing power [1]. However, the *data-decomposition* procedure commonly described in the literature [2] for the concurrent execution of the back-propagation algorithm leads to a requirement for a high communications bandwidth which low-cost platforms are unable to satisfy. The Class-Distributed (C-D) neural network, shown in Fig. 1, is a fast-learning pattern-classifier based on a modified back-propagation gradient-descent algorithm based on a distributed architecture. The inter-processor communications requirement for this new network is minimal, making efficient execution possible on low-cost parallel processing systems which are characterised by restricted communications bandwidth.

The input layer passes a p-dimensional feature vector x, which is to be assigned to one of r different classes, to the category layer. The class-discriminators in the category layer are macro-nodes, each consisting of a feed-forward network having an input buffer, a hidden layer of active (weight-carrying) neurons, and a single active output neuron. Let there be a training file $F = \{S_1 \cup S_2 \cup ... \cup S_r\}$ where S_j is a subset of F containing all training vectors belonging to class j. During the training phase the macro-nodes perform total gradient descent independently in parallel, each concurrently executing a modified cumulative back-propagation algorithm. Each node trains towards a different convergence criterion; for example, at the termination of the learning phase, the output of the *jth* class discriminator is $y_j \geq 1 - T_c \ \forall \ x_n \in S_j$ and $y_j \leq T_c \ \forall \ x_m \in S_i \ (i = 1,...r) \ i \neq j$ where T_c is a convergence threshold.

Multi-computer execution is achieved by implementing each class-discriminator as a process and scheduling these processes on the available processors. Thus, the network is decomposed by class

and for this reason it is termed a *Class-Distributed* network. The inter-processor communications requirement consists merely of the distribution of the training vectors and collation of the output values. As this is negligible, wide communications bandwidth is not required and the choice of inter-connection topology does not influence performance.

Experimental evaluation of the C-D network: The air-fuel ratio (AFR) in a gasoline-fuelled internal-combustion engine is a quantity which must be maintained accurately at its *stoichiometric* value of 14.7:1 for engine operation which is efficient and has low exhaust emissions. However, this is made difficult by the unavailability of sensors at economic cost. The C-D network was evaluated as part of a *virtual sensor* system which estimated the AFR from other combustion data. The aim of the experiments was to compare the convergence time and accuracy of the C-D network with that of a monolithic MLP in this context.

The experimental work was conducted using a single-cylinder 98.2cc capacity four-stroke engine. The time varying voltage at the spark plug was recorded for values of AFR of stoichiometric and stoichiometric ±10% at a fixed value of engine speed and load torque. Training files were constructed from 250 spark voltage records corresponding to each lambda value, pre-processed and compressed into 12-element input vectors each associated with a 3-element output vector. Similar test files were constructed using data which had not been used during training. The execution platform was based on 166MHz Intel Pentium II processor nodes connected by a low-bandwidth (approximately 1.0 Mbyte/s) network.

Fig. 2 shows the convergence times the for C-D network compared with the monolithic network for different numbers of hidden nodes when the above training files were used. The iteration time and time to convergence of each class-discriminator is considerably lower than for a comparable MLP network executing on a serial processor (termed here a *monolithic* network). The weights in both types of network are adjusted by adding to them a *delta-weight* proportional to the neuron error. In a monolithic network, the error in element q of the hidden layer is related to the output X_q, the weight between the hidden and output layers, W_{qr}, and the output layer error E_r, by:-

$$E_q[hid] = X_q[hid]\left(1 - X_q[hid]\right)\sum_{r=0}^{R}\left(E_r[out]W_{qr}[out]\right) \qquad (1)$$

In the C-D network, however, the single output node results in the less computationally onerous procedure

$$E_q[hid] = X_q[hid]\left(1 - X_q[hid]\right)\left(E[out]W_r[out]\right) \qquad (2)$$

resulting in a lower per-iteration time during training than a monolithic network.

A second reason for the improved convergence rate of the C-D network is that the number of hidden neurons required by each class-discriminator is often less than that needed by the monolithic network. This parameter is dependant on the problem domain and the topology of the input space [3], but the number of hyper-planes needed to define a single region in input space is in many cases fewer than the number needed to separate multiple regions. Further, the number of weight adjustments needed for the network to converge from its random start condition in weight space, to the desired convergence condition, is statistically likely to be lower where there are fewer weights. For similar reasons, the C-D network achieves faster recall than an equivalent monolithic network.

The *speed-gain* is defined as the ratio of the times-to-convergence of the monolithic and C-D networks, $g_m = t_m / t_{cd}$. Table 1 shows that in the case of this application, g_m increases marginally with network size, while Table 2 shows that these is little appreciable difference in the classification capabilities of the two networks. Thus, the C-D network achieves a clear speed improvement over the monolithic network during training with no classification penalty.

If the monolithic network had been decomposed onto the multi-computer system by the data-decomposition method, instead of by class-distribution, the speed improvement which was obtained is defined as the *speed-up*, $g_s = t_m / t_{dd}$, the ratio of times-to-convergence of the monolithic network and the data-decomposed network. The maximum theoretical value of speed-up obtainable with three processors is 3.0. However, previous studies have shown that with this size of network and the available communications bandwidth, communications saturation is likely to reduce the practically achievable speed-up to unity or less [4] when the data decomposition method is used. Given these constraints, the comparative speed improvement which was achieved by the C-D network is very attractive.

Conclusion: The C-D neural network architecture, offers faster training and recall when executed on a multi-computer platform. The inter-processor communications requirement is negligible, permitting efficient execution on parallel processing system with a low communications bandwidth.

R.J.Howlett and S.D.Walters (*Engineering Research Centre, University of Brighton, Moulsecoomb, Brighton, BN2 4GJ, UK)*

References

[1] McLOONE, S. and IRWIN, G.W.: 'Fast parallel off-line training of multi-layer perceptrons', *IEEE Trans. on Neural Netw.,* 1997, **8**, (3), pp. 646-653

[2] YAMAZAKI, K., KANATANI, T., WANATANABE, T., AND TOKUMARU, H.: 'Parallelization of back-propagation learning using several interconnection networks on a multi-Transputer system', *in* GREBE, R (Ed.): 'Transputer Applications and Systems' (IOS Press, Amsterdam, 1993), pp.668-680.

[3] HUSH, D.R., HORNE, B. AND SALAS, J.M. 'Error surfaces for multilayer perceptrons'. *IEEE Trans. on Systems, Man and Cybernetics*., **22**, (5), pp.1152-1161. 1992.

[4] HOWLETT R.J., 'A distributed neural network for machine vision'. (PhD Thesis. University of Brighton. 1995).

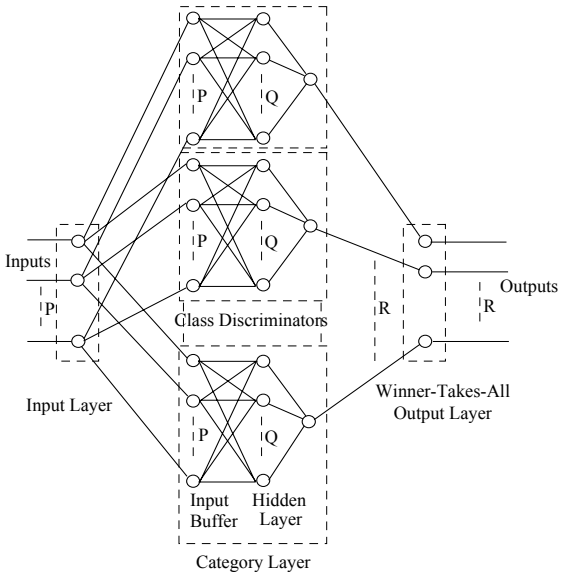

Fig. 1 *The Class-Distributed Network Architecture*

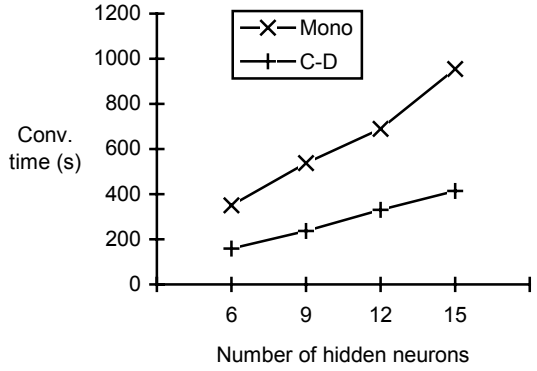

Fig. 2 *Convergence times for Monolithic and C-D networks*

No Hidden Nodes	6	9	12	15
Speed-gain	2.20	2.26	2.28	2.3

Table 1 *Speed gain for various numbers of hidden nodes*

No Hidden Nodes	6	9	12	15
Monolithic	97.8	97.6	98.2	97.6
C-D	97.8	97.3	98.0	97.3

Table 2 *Classification rates for Monolithic and C-D networks*