the given training data, it can possibly generalize under the same condition that (10) holds true. However, the size of a training set required for any empirical distribution to reflect appropriately the true underlying probability is another issue that is closely related to the Law of Large Numbers in statistics.

In summary, our (10) is justified by mathematically rigorous argument under the imposed condition of [2]. We agree that sufficiently many hidden neurons are required to satisfy that condition in general. If we have no *a priori* knowledge of the data generation mechanism, infinitely many neurons may truly be necessary to model every possible underlying probability. However, if we have available any knowledge to restrict possible probability distributions to a small hypothesis class, then finitely many hidden neurons may suffice to model every member of the class. Therefore, evaluation of the number of neurons required to provide the network with sufficient function capacity is relative to the *a priori* knowledge available in a specific situation, and is not considered to be clarified in general. Thus our feeling is that we do not have any definite answer to the question whether our required condition is impractical or not.


REFERENCES

[1] E. Barnard, "Comment on 'Bayes statistical behavior and valid generalization of pattern classifying neural networks'," *IEEE Trans. Neural Networks*, vol. 3, Nov. 1992, this issue.

[2] F. Kanaya and S. Miyake, "Bayes statistical behavior and valid generalization of pattern classifying neural networks," *IEEE Trans. Neural Networks*, vol. 2, pp. 471–475, July 1991.

[3] D. Ruck, S. Rogers, M. Kabrisky, M. Oxley, and B. Suter, "The multilayer perceptron as an approximation to a Bayes optimal discriminant function," *IEEE Trans. Neural Networks*, vol. 1, pp. 296–298, Dec. 1990.


## Comments on "Dynamic Programming Approach to Optimal Weight Selection in Multilayer Neural Networks"

Barak A. Pearlmutter

In the above paper, Saratchandran[1] presents an efficient algorithm using dynamic programming to find weights which load a set of examples into a feedforward neural network with minimal error. $NP$-complete problems have been reduced to special cases of this loading problem [2], [3], so an efficient algorithms implies $P = NP$, a significant result indeed.

Regrettably a confusion lies buried in the paper's notation. Since $w(k)$ is used to represent the weights from layer $k$ to layer $k+1$, it might seem reasonable to use $w^*(k)$ to represent the setting of these weights that minimizes the error. The author proposes to compute $w^*(n-1)$, then $w^*(n-2)$, etc., thus computing optimal settings for all the weights. However, $w^*(k)$ is a function of the settings of the weights at all the other layers, and should really be written $w^*(k, w(1), \ldots, w(k-1), w(k+1), \ldots, w(n-1))$.

Manuscript received February 8, 1992.

The author is with the Department of Computer Science and Engineering, Oregon Graduate Institute of Science & Technology, Beaverton, OR 97006–1999.

IEEE Log Number 9201745.

[1] P. Saratchandran, *IEEE Trans. Neural Networks*, vol. 2, pp. 465–467, July 1991.

This makes the problem with the algorithm clear. First the weights $w(n-1)$ are set optimally. Then the weights $w(n-2)$ are changed to optimal. At this point $w(n-1)$ is no longer optimal, as the assumption made when their optimal setting was calculated, namely that the unit states $y(n-1)$ are held constant, which necessitates holding $w(1), \ldots, w(n-2)$ constant, has been violated.


REFERENCES

[1] A. Blum and R. L. Rivest, "Training a 3-node neural net is NP-complete," in *Advances in Neural Information Processing Systems I*, D. Touretzky, Ed. New York: Morgan Kaufman, 1989.

[2] J. S. Judd, "Learning in networks is hard," in *IEEE First Int. Conf. on Neural Networks*, San Diego, CA, June 21–24, 1987, pp. 685–692.


## Authors' Reply

P. Saratchandran

The primary concern of Dr. Pearlmutter is that the algorithm proposed in my paper implies $P = NP$ although nowhere in the paper is such a claim made. In fact the algorithm only proposes an iterative layer by layer approach to training a multilayer feed forward network without claiming that it scales polynomially with problem size.

The concern seems to be due to some misunderstandings about the implementation of dynamic programming based algorithms. Such details were not included in the paper as they are well documented in text books [1]-[3] on dynamic programming. The weights $w^*(..)$ derived in the paper are for each quantized state of $y(..)$ and are not global optimal weights. Following explanation of implementation will further clarify any misgivings that the algorithm runs in polynomial time.

Because neuron activation functions are nonlinear the outputs $y$ for every layer has to be quantized [1]-[3] and the optimal weights $w^*$ computed for each quantized state. Thus for the $(n-1)$th layer, corresponding to each quantized state of $y(n-1)$, we can compute a $w^*(n-1)$ and a minimum error $I(y(n-1))$ using (8) and (9) in the paper. Store these in a table.

At $(n-2)$nd layer, for each quantized state of $y(n-2)$, we repeatedly calculate $w^*(n-2)$ and the error $I(y(n-2))$ corresponding to every quantized state of $y(n-1)$ using the appropriate $w^*(n-1)$ in (12) in the paper. Thus for every quantized $y(n-2)$ we will have as many $\{w^*(n-2), w^*(n-1)\}$ as the number of states of $y(n-1)$. For each state of $y(n-2)$ pick only that $w^*(n-2), w^*(n-1)$ which resulted in the lowest error $I(y(n-2))$ and store them in a table as we need them for the calculation of $w^*(n-3)$. Continue this procedure until we reach the first layer.

At the first layer there is no need to quantize $y(1)$ as these are the inputs to the network and consequently known. So we only need compute the $w^*(1)$ and the error $I(y(1))$ corresponding to every quantized state of $y(2)$ using the appropriate $\{w^*(2), \ldots, w^*(n-2), w^*(n-1)\}$ from the table generated at the 2nd layer. The $w^*(1)$ that resulted in the lowest error $I(y(1))$ and the corresponding

Manuscript received March 25, 1992.

The author is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 22263.

IEEE Log Number 9201746.