

## NEURAL NETWORK APPROACH TO ENERGY CONFINEMENT SCALING IN TOKAMAKS\*

L. ALLEN and C. M. BISHOP†

AEA Technology, Culham Laboratory, Oxon OX14 3DB, U.K.  
(Euratom/UKAEA Fusion Association)

(Received 8 July 1991; and in revised form 16 December 1991)

**Abstract**—Empirical studies of the scaling of Tokamak energy confinement times with machine parameters constitute a useful point of contact with physics-based transport theories. They also form the basis for the design of next-step and reactor grade Tokamaks. In most cases a simple power law (or sometimes offset linear) functional form is fitted to the data. Such linear regression techniques have the advantage of computational simplicity, but otherwise have little *a-priori* justification. Neural networks provide a powerful general purpose technique for nonlinear regression which exhibits no essential limitations on the functional form which can be fitted. In this paper we apply neural networks to the problem of energy confinement scaling and we illustrate the technique using data from the JET (Joint European Torus) Tokamak. The results show that the neural network approach leads to a substantial improvement in the ability to predict the energy confinement times as compared with linear regression. The significance of this result is discussed.

### 1. INTRODUCTION

SUSTAINED THERMONUCLEAR FUSION in a magnetically confined plasma requires that the product of temperature, density and energy confinement time should exceed a critical threshold (the Lawson criterion). The confinement time is therefore a quantity of central importance in the design of next-step and reactor grade Tokamaks. Improvements in the confinement time can be obtained by increasing the volume of the plasma. However, this greatly increases the cost of the system. Indeed, uncertainties in the expected confinement time of a large Tokamak system require the inclusion of extra margins in the design parameters, and therefore improvements in the ability to predict the confinement time accurately can result in substantial cost savings. However, due to the present limitations in the theoretical understanding of energy transport processes in a magnetically confined plasma, it is not possible to give a first principles calculation of the energy confinement time.

For these reasons, empirical studies of the dependence of global confinement time  $\tau_E$  on various physical parameters have received considerable attention in recent years (see, for example, CONNOR, 1988; KAYE *et al.*, 1990). The resulting scaling relations constitute a convenient point of contact with theoretical models of transport, and extrapolations of the scaling relations form the basis for the design of future Tokamaks. For most such studies a particular functional form for the confinement time, in terms of a number of physical variables, is assumed. Free parameters contained in the expression for  $\tau_E$  are determined by means of a least-squares fit to an experimental database. In general the functional form is taken to be a multivariate power law expression which (after taking logarithms) leads to a *linear* problem to determine the unknown exponents. Such a power law expression does not arise from

\* UKAEA copyright © 1992.

† Present address: AEA Technology, Harwell Laboratory, Oxon, OX11 0RA, U.K.

any general theoretical constraint, but rather is chosen for computational convenience. Offset linear forms, which also generate linear expressions for the free parameters, have also been used. In this paper we exploit neural network techniques to eliminate this restriction and thereby allow *nonlinear* regression using a very general class of functional forms.

In the next section we give a brief introduction to neural networks and describe the particular network architecture known as the multilayer perceptron which will form the basis of our approach to confinement scaling. Section 3 describes how neural networks can be applied to the analysis of confinement scaling in Tokamaks, and in Section 4 results obtained using data from the JET experiment are presented. We find that the neural network method leads to a significantly better prediction for the energy confinement time than does the conventional linear regression approach. Finally, conclusions are given in Section 5.

## 2. THE MULTILAYER PERCEPTRON

Neural networks are analogue computational systems whose structure is inspired by studies of the brain. Many different architectures of neural network have been developed to tackle a variety of problems, and research in this area continues at a rapid pace. For introductory reviews of neural networks see LIPPMANN (1987) or REALE and JACKSON (1990). In this section we give a brief overview of a relatively simple but very widely used network type known as the multilayer perceptron (MLP). This class of network will form the basis for our analysis of energy confinement scaling in the next two sections.

An MLP consists of a network of units (also known as processing elements or neurons) as illustrated in Fig. 1. Each unit is shown as a circle in the diagram, and the lines connecting them are known as weights or links. The network can be thought of as describing an analytic mapping between a set of real-valued input variables  $x_m$  ( $m = 1, \dots, M$ ) and a set of output variables  $y_n$  ( $n = 1, \dots, N$ ). The input variables are applied to the  $M$  input units at the bottom of the diagram ( $M = N = 4$  in the case of Fig. 1). These variables are multiplied by a matrix of parameters  $w_{lm}$  ( $l = 1, \dots, L$ ;  $m = 1, \dots, M$ ) corresponding to the first layer of links. Here,  $L$  is the number of units in the middle, or hidden, layer. ( $L = 3$  in the example shown in Fig. 1.) This results in a vector of inputs to the units in the middle layer. Each component of this vector

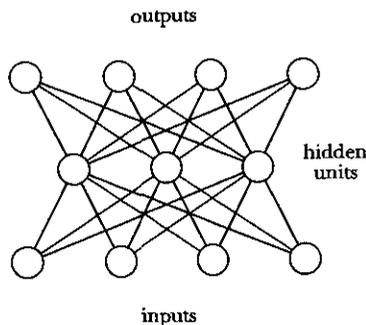


FIG. 1.—An example of a multilayer perceptron network, in this case having four input units, three units in the “hidden” layer, and four output units.

is then transformed by a nonlinear function  $f(\ )$  so that the outputs of the middle layer units can be written

$$z_l = f\left(\sum_{m=1}^M w_{lm}x_m + \theta_l\right) \quad (l = 1, \dots, L) \quad (1)$$

where  $\theta_l$  is an offset (or threshold). The function  $f(\ )$  is generally chosen to be the sigmoid defined by

$$f(x) \equiv \frac{1}{1 + e^{-x}}, \quad (2)$$

and this function is plotted in Fig. 2.

The outputs from the hidden layer units are now multiplied by a second matrix of parameters  $\hat{w}_{nl}$  ( $n = 1, \dots, N; l = 1, \dots, L$ ), and offsets  $\hat{\theta}_n$  are added to the components of the resulting vector to generate the network outputs:

$$y_n = \sum_{l=1}^L \hat{w}_{nl}z_l + \hat{\theta}_n \quad (n = 1, \dots, N). \quad (3)$$

Combining equations (1) and (3) we see that the entire network corresponds to a mapping from inputs  $x_m$  to outputs  $y_n$  which is specified by the analytic function

$$y_n(x_1, \dots, x_M) = \sum_{l=1}^L \hat{w}_{nl}f\left(\sum_{m=1}^M w_{lm}x_m + \theta_l\right) + \hat{\theta}_n \quad (4)$$

where  $f(\ )$  is defined by equation (2). This mapping is parameterized by the quantities  $w_{lm}$ ,  $\theta_l$ ,  $\hat{w}_{nl}$  and  $\hat{\theta}_n$ .

More complex architectures of MLP, having more than one layer of hidden units,

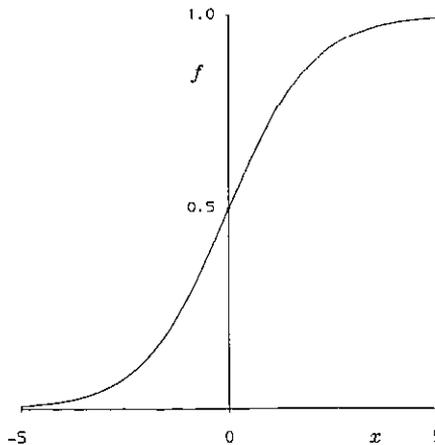


FIG. 2.—Plot of the sigmoid function  $f(x)$  defined in equation (2).

or different patterns of connectivity between the units, are also of interest, and an example of an alternative architecture will be introduced in Section 5. Note that MLPs can also be used to solve classification problems in which the network outputs are to be interpreted as binary quantities according to whether the output is above or below some threshold. In this case, it is usual to apply the nonlinear activation function of equation (2) to the outputs  $y_n$ . This would be inappropriate, however, in the present context where we are interested in fitting smooth continuous mappings. Neural networks, such as the MLP, in which information is fed from a set of inputs through a series of hidden units to a set of outputs are referred to as *feedforward networks* to distinguish them from more complex networks with feedback loops.

The functional form in equation (4) appears somewhat arbitrary. Its importance, however, stems from two crucial properties:

- (1) For suitable choices of the parameters  $w_{lm}$ ,  $\theta_l$ ,  $\hat{w}_{nl}$  and  $\hat{\theta}_n$  the mapping can approximate, with arbitrary accuracy, any given nonlinear multivariate mapping (subject to some mild restrictions) provided the number  $L$  of middle layer units is sufficiently large. Formal discussions of this property can be found in FUNAHASHI (1989) and HORNIK *et al.* (1989).
- (2) Given a set of  $P$  exemplar vector pairs  $\{x_m^p, y_n^p\}$ ,  $p = 1, \dots, P$ , characterizing a particular mapping, there exist procedures, based on the technique known as *error backpropagation*, for determining appropriate values for the parameters  $w_{lm}$ ,  $\theta_m$ ,  $\hat{w}_{nl}$  and  $\hat{\theta}_n$  so that the network function in equation (4) approximates the required mapping. (In the neural network terminology these are referred to as training or learning algorithms.)

Error backpropagation aims to minimize an error function  $E$  defined to be the root-mean-square (RMS) error between the output vector  $\mathbf{y}(\mathbf{x}^p)$  of the network (for given input vector  $\mathbf{x}^p$ ) and the corresponding target vector  $\mathbf{y}^p$ , summed over all exemplars  $p$ :

$$E = \left\{ \frac{1}{P} \frac{1}{N} \sum_{p=1}^P \sum_{n=1}^N (y_n(\mathbf{x}^p) - y_n^p)^2 \right\}^{1/2}. \quad (5)$$

Thus,  $E$  is a function of the values of  $w$  and  $\theta$ . Backpropagation is a recursive procedure for evaluating the derivatives of  $E$  with respect to these parameters. A detailed account of the backpropagation procedure can be found in RUMELHART and MCCLELLAND (1986). Knowledge of these derivatives allows  $E$  to be minimized using standard optimization procedures. We have compared a variety of techniques including enhanced gradient descent, conjugate gradients and quasi-Newton methods. The memoryless BFGS (Broyden-Fletcher-Goldfarb-Shanno) technique (SHANNO, 1978) was found to be both robust and relatively fast and was used to obtain the results presented in Section 4. It is described in the neural network context in BATTITI and MASULLI (1990).

The training of an MLP using a set of exemplar vectors can be regarded as analogous to the fitting of a polynomial curve through a set of data points. The coefficients of the polynomial correspond to the weights and thresholds in the network. In effect, the MLP represents an efficient generalization of curve fitting to allow for an arbitrary

number of independent (input) and dependent (output) variables together with a very general class of functional forms (BISHOP and ROACH, 1992). One of the penalties to be paid for this generality is that the determination of the weights and thresholds in the network (i.e. the training of the network) is a nonlinear optimization problem which is computationally intensive, and which can sometimes exhibit sub-optimal solutions (corresponding to local minima in the error function).

In practical applications of neural networks it is necessary to restrict the class of functions which is fitted in order to obtain satisfactory generalization to new data which was not included in the training data set. The usual procedure for achieving this is to restrict the number of hidden units. This is analogous to limiting the number of terms (and hence the number of degrees of freedom) in polynomial curve fitting. An alternative approach is to impose directly some constraint on the functional form which the network can generate, for instance that it should not have large curvature (BISHOP, 1990). In this paper we have chosen to limit the number of hidden units, and a suitable procedure for determining the appropriate value for this number is discussed in the next section.

### 3. ENERGY CONFINEMENT SCALING IN TOKAMAKS

In seeking empirical scaling laws it is convenient to consider expressions for  $\tau_E$  in terms of appropriate dimensionless parameters (CONNOR, 1988; THOMAS, 1987). Thus, for auxiliary heated plasmas, we would typically consider the form

$$\tau_E = \tau_{E0} F(q, v_*, \beta_p, \rho_c/a) \quad (6)$$

where  $q$  is the safety factor,  $R$  is the major radius,  $v_* \equiv v_{eff}/\omega_b$ ,  $v_{eff}$  is the effective electron collisionality for trapped electrons,  $\omega_b$  is the trapped electron bounce frequency,  $\beta_p$  is the poloidal beta,  $\rho_c$  is the electron Larmor radius,  $a$  is the minor radius and  $\tau_{E0} \equiv Rq/T_e^{1/2}$ , where  $T_e$  is the electron temperature in keV (note that  $T_e^{1/2} \propto v_{the}$  where  $v_{the}$  is the electron thermal velocity, and thus  $\tau_{E0}$  can be considered to have the same dimensions as  $\tau_E$ ). Since many of these regressor variables have spatial variations, appropriate profile-averaged quantities should be used. The function  $F(\ )$  is to be determined using an experimental database.

The conventional approach to this problem is to choose a form for  $F(\ )$  which is the product of powers of the independent variables, so that

$$\tau_E = \tau_{E0} C q^{\alpha_1} v_*^{\alpha_2} \beta_p^{\alpha_3} (\rho_c/a)^{\alpha_4} \quad (7)$$

in which  $C, \alpha_1, \dots, \alpha_4$  are unknown parameters. This functional form is computationally convenient since by taking logarithms of both sides we obtain an expression which is linear in  $C$  and  $\alpha_i$ :

$$\ln(\tau_E/\tau_{E0}) = \ln C + \alpha_1 \ln q + \alpha_2 \ln v_* + \alpha_3 \ln \beta_p + \alpha_4 \ln(\rho_c/a). \quad (8)$$

Given an experimental dataset containing sets of values of  $(q, v_*, \beta_p, \rho_c/a)$ , and the corresponding values of  $\tau_E$ , it is then straightforward to determine the values of  $C$  and

$\alpha_i$  which minimize the RMS error between the experimental values of  $\tau_E$  and those predicted by equation (8). Note that this corresponds to the solution of a set of linear equations. We shall refer to this approach as linear regression (LR).

The functional form in equation (7) is, however, largely arbitrary and is not the result of any known physics constraint. Neural networks of the multilayer perceptron type represent a powerful technique for exploring a much wider class of functions  $F(\ )$ . We shall consider network functions with four inputs  $\ln q$ ,  $\ln v_*$ ,  $\ln \beta_p$  and  $\ln(\rho_c/a)$ , and a single output  $\ln(\tau_E/\tau_{E0})$ . The corresponding neural network architecture is shown in Fig. 3. The use of logarithms compresses the dynamic range of the variables and also allows a closer comparison of the neural network technique with LR. Indeed, if we consider a network whose transfer function  $f(\ )$  in equation (4) is replaced by the identity function, then the network mapping reduces to a simple matrix multiplication and the network approach is then equivalent to LR.

The optimum value for  $L$ , the number of hidden units in the network, cannot be known in advance since it will depend on the properties of the dataset. In effect  $L$  determines the number of degrees of freedom (i.e. the number of weights and thresholds) which parameterize the functional form. We would expect that increasing this number would lead to a steadily smaller value of the residual error. This can, however, result in "over-fitting" in which the network tends to store individual data points rather than representing the underlying trends in the data. We are interested primarily in obtaining a network mapping with the greatest *predictive* capability, that is one which produces the smallest output error when applied to new data which was not part of the training set. For this reason we divide the experimental dataset into two (randomly partitioned and therefore nominally equivalent) parts which we shall refer to as the training and test sets. The training set is used to determine the values of the weights and thresholds in the network, and the test set is used to assess the predictive capabilities of the functional form represented by the trained network. We therefore compare networks with different numbers of hidden units and select the network giving the lowest RMS error with respect to the test data. For comparison, the same technique will be applied to linear regression, in which the regression exponents will be determined using the training data and the error evaluated for the test data.

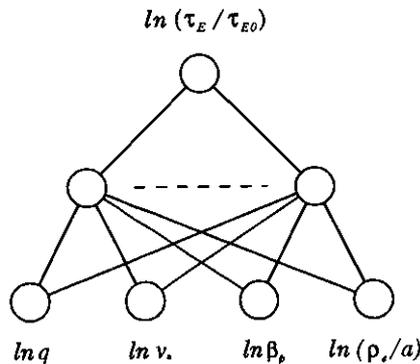


FIG. 3.—The neural network architecture used to implement the scaling relation of equation (6).

## 4. RESULTS FROM JET DATA

Energy confinement data from the JET Tokamak have been used to examine the capabilities of the network approach to confinement scaling introduced in the previous section. In this study we are interested primarily in auxiliary heated discharges and so data were selected for which the total heating power  $P_{\text{TOT}} \geq 2P_{\text{OH}}$ , where  $P_{\text{OH}}$  is the Ohmic heating power. Attention was also restricted to limiter or inner wall deuterium cases during the current flat top. The resulting dataset consisted of 1,147 observations which were divided into a training set of 574 points and a test set of 573 points. Linear regression gave the following results:

$$\tau_E/\tau_{E0} \propto q^{-1.69} v_*^{-0.50} \beta_p^{0.38} (\rho_c/a)^{-1.49}. \quad (9)$$

Figure 4 shows the training and test error values obtained from the neural network with various numbers  $L$  of hidden units. The results from linear regression are plotted at  $L = 0$ . Each network was trained for 500 cycles of the BFGS algorithm, each of which consists of  $\mathcal{N}$  line search minimizations, where  $\mathcal{N}$  is the total number of weights and thresholds in the network. For each cycle the search direction starts with gradient descent and is then updated using the BFGS algorithm.

It can be seen that the training set error falls as the number  $L$  of hidden units is increased, as would be expected since the number of degrees of freedom increases with  $L$ . The fact that the decrease in error is not strictly monotonic is a consequence of the fact that (unlike LR) the training of the network corresponds to the solution of a nonlinear optimization problem and so the solution must be found iteratively. Insufficient number of iterations, or the presence of local minima in  $E$ , can result in sub-optimal solutions. The test error also decreases with  $L$  at first but then begins to saturate. The smallest test error occurs at  $L = 22$ , and is significantly smaller (by

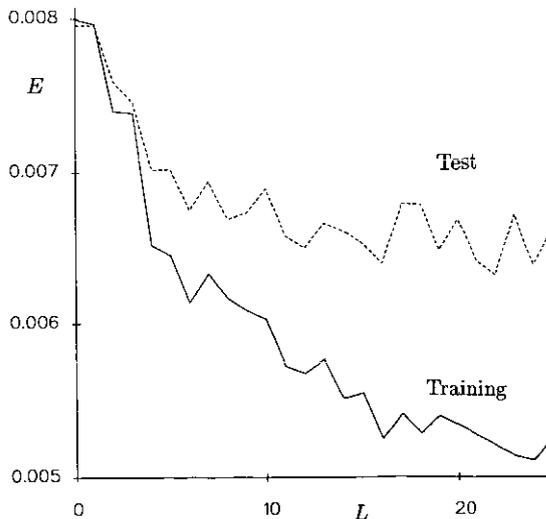


FIG. 4.—Plot of the root-mean-square error for both training and test data sets as a function of the number of hidden units in the network. The results from linear regression are plotted at  $L = 0$ .

about 20%) than the corresponding value obtained from linear regression. The error values from linear regression and from the optimal neural network are summarized in Table 1.

The RMS error obtained for the test data is about 20% smaller for the neural network compared with linear regression, corresponding to an increased capability to predict the confinement time of a new discharge. This suggests that the usual power law form used for many confinement scaling studies may be inadequate. The reason that the reduction in error is not larger than this is probably due to the considerable random errors (noise) associated with energy confinement data.

Unlike linear regression, the neural network does not lead to a simple analytic expression for the confinement time scaling law. This is hardly surprising since the point of using neural networks was to permit a very wide class of functional forms. Although an analytic form can be written down using equations (2) and (4), together with the values of the weights and thresholds from the trained network, this would provide little useful insight. Instead, we compare the LR and neural network results by plotting  $\ln(\tau_E/\tau_{E0})$  versus the regressor variables  $\ln q$ ,  $\ln v_*$ ,  $\ln \beta_p$  and  $\ln(\rho_e/a)$ .

Principal component analysis applied to the test regressor variables, after first normalizing the variables by their mean value, gave the following eigenvalues

$$(0.97, 0.24, 0.03, 0.02)$$

indicating a significantly nonuniform distribution of input data. Thus, a plot of  $\ln(\tau_E/\tau_{E0})$  versus one of the regressor variables, with the other regressor variables held fixed, would almost certainly lead to significant extrapolations outside the range of the training data. To ensure that we are interpolating within the range of the dataset, we plot  $\ln(\tau_E/\tau_{E0})$  along the first principal axis, given by

$$q = 4.09 + 1.48\eta \quad (10)$$

$$v_* = 0.072 + 0.059\eta \quad (11)$$

$$\beta_p = 0.270 + 0.118\eta \quad (12)$$

$$\rho_e/a = 1.72 \times 10^{-5} + 1.39 \times 10^{-6}\eta \quad (13)$$

where the parameter  $\eta$  spans the range  $(-1.0, 3.0)$ . Figure 5 shows  $\tau_E/\tau_{E0}$  along the principal axis for both linear regression (dashed curve) and for the 22-hidden-unit neural network (solid curve). It is clear that the neural network has found a functional form which is similar to that of LR. Figure 6 shows the value of  $\ln(\tau_E/\tau_{E0})$  for each point in the test data set plotted against the corresponding value as predicted by linear regression. A similar plot for the neural network prediction is shown in Fig. 7.

TABLE 1.

Error values for JET data		
Method	Train RMS	Test RMS
Linear regression	0.007995	0.007957
Neural net (22 hidden units)	0.005207	0.006319

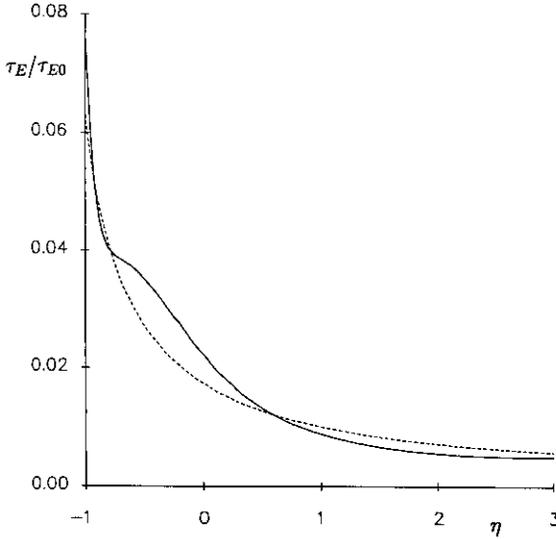


FIG. 5.—Graph of  $(\tau_E/\tau_{E0})$  against the parameter  $\eta$  along the first principal direction. The solid curve corresponds to the neural network and the dotted curve to linear regression.

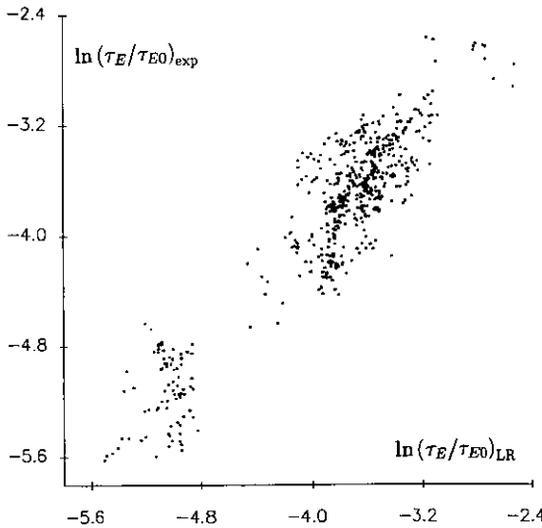


FIG. 6.—Plot of  $\ln(\tau_E/\tau_{E0})$  from the test dataset versus the corresponding quantity predicted by linear regression.

Comparison of Figs 6 and 7 shows the improved predictive capability of the neural network as compared with linear regression.

### 5. CONCLUSIONS

In this paper we have described the particular architecture of neural network known as the multilayer perceptron and we have shown how it goes beyond the method of

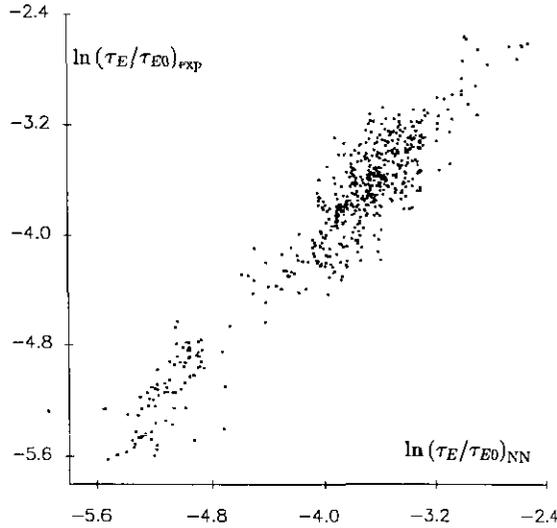


FIG. 7.—Plot of  $\ln(\tau_E/\tau_{E0})$  from the test dataset versus the corresponding quantity predicted by the neural network.

linear regression by allowing a much larger class of functional forms. We have applied this technique to the problem of energy confinement time scaling in Tokamaks. For a set of data from neutral beam heated JET plasmas we found a significant improvement in the confinement time prediction obtained from the neural network as compared with linear regression. The general form of the scaling relation was similar in the two cases.

One feature of the neural network approach is the lack of a simple analytic expression for the resulting “scaling law”. This is an unavoidable consequence of having access to a wide range of functional forms. Visualization of the scaling properties requires the network mapping to be displayed graphically.

A second difficulty with the use of neural networks in this context concerns the extrapolation of the results outside the range of values for the regressor variables spanned by the training data. While any extrapolation is prone to error, the neural network is much less likely to produce smooth extrapolations than linear regression as a consequence of the much less tightly constrained functional form, and also because of the tendency of the hidden units to “saturate” for large inputs (see Fig. 2).

In the present context, however, this problem can be largely circumvented. In extrapolating from JET results to a large scale fusion reactor many of the dimensional quantities differ significantly, although the dimensionless parameters  $q$ ,  $v_*$  and  $\beta_p$  for a reactor lie in the range spanned by the JET data. Thus, for these dimensionless variables, the extrapolation problem can be avoided. However, the dimensionless parameter  $\rho_e/a$  is about a factor 2 smaller in a reactor than in JET. Fortunately, there are strong theoretical reasons (CONNOR, 1988) to believe that the dependence of the confinement time on  $\rho_e/a$  should follow a simple power law. Thus, we are led to consider a scaling relation having the following functional form

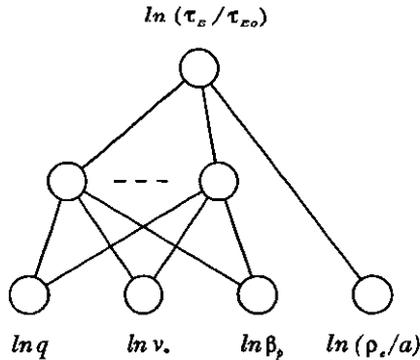


FIG. 8.—The neural network architecture corresponding to equation (14).

$$\tau_E = \tau_{E0}(\rho_e/a)^\gamma G(q, \nu_*, \beta_p) \quad (14)$$

which can be represented in the form of a neural network as shown in Fig. 8. This functional form satisfies the power law constraint on  $\rho_e/a$ , but is otherwise arbitrary. The weights and thresholds of the corresponding network can be found using the training algorithms discussed in Section 2, and the network function can be extrapolated to reactor conditions without encountering any of the aforementioned problems.

In addition, if the energy confinement data from a Tokamak follows the functional form of equation (14), and we fit a standard power law of the form of equation (7), then the value of  $\alpha_4$  obtained from (7) will not in general be equal to the value of  $\gamma$  in (14). Therefore, studies which use conventional linear regression may obtain an incorrect value for the exponent  $\alpha_4$ . This may obscure the true power law behaviour of  $\tau_E/\tau_{E0}$  with respect to  $\rho_e/a$ , and cloud the comparison of theoretically predicted scalings with experimental data. The neural network architecture of Fig. 8 will not suffer from this problem, and it is hoped to pursue this approach further in the future.

One situation in which the neural network would be expected to perform significantly better than linear regression is where the data spans some threshold for an additional transport mechanism to switch on or off. Resulting discontinuities in the derivatives of the confinement time with respect to one or more regressor variables can easily be represented by the neural network but lie outside the scope of standard linear regression.

*Acknowledgements*—The bulk of this work was carried out under JET Task Agreement No. 5 “Testing Theoretical Transport Models”, and the authors would like to thank JET staff for their assistance and advice. The authors would also like to thank I. G. D. STRACHAN for his contributions to the neural network analysis, G. P. MADDISON for his assistance in the use of the JET database, and J. W. CONNOR for a number of helpful suggestions in connection with Tokamak scaling laws.

#### REFERENCES

- BATTITI R. and MASULLI F. (1990) *Proc. Int. Neural Network Conference*, Paris, Vol. II, p. 757. Kluwer, Utrecht.  
 BEALE R. and JACKSON T. (1990) *Neural Computing; an Introduction*. Adam Hilger, London.

- BISHOP C. M. (1990) *Proc. Int. Neural Network Conference*, Paris, Vol. II, p. 749.
- BISHOP C. M. and ROACH C. M. (1992) Fast Curve Fitting Using Neural Networks, AEA Fusion Report AEA-FUS-162, submitted to *Rev. Sci. Inst.*
- CONNOR J. W. (1988) *Plasma Phys. Contr. Fusion* **30**, 619.
- FUNAHASHI K. (1989) *Neural Networks* **2**(3), 183.
- HORNICK K., STINCHCOMBE M. and WHITE H. (1989) *Neural Networks* **2**(5), 359.
- KAYE S. *et al.* (1990) *Physics Fluids* **B2**, 2926.
- LIPPMAN R. P. (1987) *An Introduction to Computing with Neural Nets*. IEEE ASSP Magazine, April.
- RUMELHART D. E. and MCCLELLAND J. L. (1986) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* Vol. 1: Foundations. Cambridge MA. MIT Press, MA.
- SHANNO D. F. (1978) *Math. Operations Res.* **3**, 244.
- THOMAS P. R. (1987) Preprint JET-P(87)-17, Jet Joint Undertaking, Abingdon, Oxfordshire.