ELSEVIER

BioSystems 38 (1996) 15–28

Bio Systems

On the analysis of the inverse problem of metabolic pathways using artificial neural networks

Pedro Mendes, Douglas B. Kell*

Institute of Biological Sciences, University of Wales, Aberystwyth, Dyfed SY23 3DA, UK

Received 4 April 1995; revision received 17 July 1995; accepted 14 July 1995

Abstract

Here we develop the use of artificial neural networks for solving the inverse metabolic problem, in other words, given a set of steady-state metabolite levels and fluxes in a pathway of known structure to obtain the parameters of the system, in this case the enzymatic limiting rate and Michaelis constants. This requires two main procedures: first the development of a computer program with which one can model metabolism in the forward direction (i.e. given the internal and parameters to determine the steady-state fluxes and metabolite concentrations), and second, given arrays of associated parameters and variables thereby obtained, to exploit artificial neural networks to form a model capable of obtaining the parameters from the variables. We studied 2-step pathways exhibiting first-order kinetics, 2-step pathways exhibiting reversible Michaelis-Menten kinetics and then 3-step pathways (again exhibiting reversible Michaelis-Menten kinetics), modelled using the program Gepasi. Whilst it was fairly easy for the networks to learn most of the parameters in the 2-step pathway, it was found helpful for the Michaelis-Menten case to vary the concentration of the starting pathway substrate for each set of internal parameters, and to train separate networks for each parameter. Some parameters were much easier to learn than others, reverse K_m and V_{max} values normally being the most difficult. For the 3-step pathway learning sometimes required as much as 3 days, and occasionally convergence was not obtained. Overall, neural networks of the present type, with fully interconnected feedforward architectures and trained according to the backpropagation algorithm, scaled poorly as the problem size was increased.

Keywords: Artificial neural networks; Metabolic control analysis; Enzyme kinetics; Parameter estimation

1. Introduction

Metabolic Control Analysis (MCA), which stems from the work of Kacser and Burns (1973) and Heinrich and Rapoport (1974), is a formalism which allows one to establish the extent to which individual enzymes in a metabolic pathway control both the flux through that pathway and the concentrations of intermediary metabolites (for a recent review see Fell, 1992). MCA is an exact algebraic formalism (Reder, 1988), valid for any arbitrarily complex metabolic pathway. MCA provides measures of how perturbations in metabolic steps (enzymes, etc.) affect the variables, i.e. the metabolite concentrations and the fluxes. This is

* Corresponding author, Fax: +44 1970 622354; E-mail: PRM,DBK@aber.ac.uk

SSDI 0303-2647(95)01565-3

done via the flux- and concentration-control coefficients. MCA can be an important tool for the rational optimization of metabolic pathways, where the objective is to optimize some metabolic variable. This is a particularly desirable goal in areas such as biotechnology (Kell and Westerhoff, 1986), where the field is nowadays commonly referred to as 'metabolic engineering' (Backman et al., 1990; Bailey et al., 1990; Bailey, 1991; Stephanopoulos and Vallino, 1991; Tong et al., 1991; Ikeda and Katsumata, 1992; Cameron and Tong, 1993; Katsumata and Ikeda, 1993). The general approach is based on the analysis of the relevant steady state using MCA followed by a choice of the steps with larger control coefficients to be the targets for manipulation (nowadays typically via cloning, Niederberger et al., 1992). One major result of the MCA formalism is that it is now accepted that the control of steady-state flux and metabolite concentrations is distributed through *all* the steps of metabolism, even enzymes of pathways other than the one leading directly to the output flux of primary interest, as long as there are links between them (Kacser and Burns, 1973). As a consequence of the summation theorem, which states that all flux-control coefficients add up to unity, most steps in fact have negligible control over any flux, a fact that has been confirmed in several independent experiments (see Fell, 1992 for a review). This reinforces the notion of systemic control, as each step alone has little control over a flux (or concentration), but a group of steps can have considerable control over the same flux, if acting in coordination. Niederberger et al. (1992) have elegantly shown this for the tryptophan system of yeast, and Kacser and Acerenza (1993) have indicated how for certain pathways the relative extent of the cloning required is a simple function of flux ratios. MCA's advantage in such applications is that it is an exact quantitative approach. Its drawback is that it is only valid for infinitesimal perturbations. Due to the nonlinearity of the kinetics of metabolic systems, however, extrapolations to large changes cannot in general be done with any degree of accuracy (see also Small and Kacser, 1993)

Consider a typical (generalised) metabolic pathway such as that in Fig. 1. An initial or 'external' metabolite X_0, present at a constant or 'clamped' concentration, is transformed by enzymes E_1 to E_n via metabolites S_1, S_2,..., S_{n-1} to form X_n (like X_0, present at a constant concentration). In the steady state, the concentrations of S_1, S_2, ... S_{n-1} are constant, so that the rate of production of X_n is constant as is the metabolic flux. Apart from implicit parameters such as the temperature, the *parameters* of the system (also known as independent variables) are the concentrations and kinetic constants of the enzymes, plus the concentration of the external metabolites, whilst the *variables* of the system (or dependent variables) are the steady-state metabolite concentrations and the flux(es). Because enzymatic rate equations are nonlinear, there may apparently be little relation between the properties of individual enzymes when studied in isolation and the behaviour of the system as a whole. In MCA, the so-called control coefficients describe quantitatively the role of individual enzymes in controlling the flux and metabolite concentrations. These coefficients depend solely on the parameters of the system defined as above. Our problem is that the factors controlling the pathway, which is what we wish to understand, are determined by the parameters, which are extremely difficult to measure in vivo, and not by the variables (whose determination requires only conventional (bio)chemical analyses).

A number of workers have devised computer programs for the simulation of the dynamics of metabolic pathways and their steady-state analysis within the framework of MCA (see e.g. Holzhütter and Colosimo, 1990; Cornish-Bowden and Hofmeyr, 1991; Sauro and Fell, 1991; Mendes, 1993). In such programs, one can input the structure of the pathway of interest, the rate equations and the values of the kinetic parameters of each enzyme, and the concentrations of the external metabolites. The computer then runs the system to a steady state, establishing the metabolite concentrations and the fluxes that result from the parameters chosen. The control coefficients and other control analytic parameters may then be obtained analytically or numerically. The problem is that constraints on computational time mean that whilst we can determine the variables from the parameters we cannot solve the inverse problem ('go backwards'), and test all possible combinations of parameters to see which would give us the best fit to an observed set of variables, i.e. effect a global minimisation of the parameter space. Since the parameter space of most biochemical systems has a large number of dimensions, one will frequently find instances in which the function to minimize has many local minima, and therefore is difficult to optimize. Although not formally proven to our knowledge for enzymatic reaction networks, it is likely that such inverse problems are *NP*-complete (see Garey and Johnson, 1979; Anthony and Biggs, 1992). Numerical methods such as steepest descent or other variations of the Newton method are not normally able to find the global optimum in such cases.

Artificial neural networks ('neural nets') are collections of identical but very simple 'computational units' which can take a numerical input and transform it into an output (e.g. Rumelhart et al., 1986; McClelland and Rumelhart, 1988; Cowan and Sharp, 1988; Wasserman, 1989; Amit, 1989; Kohonen, 1989; Pao, 1989; Aleksander and Morton, 1990; Beale and Jackson, 1990; Eberhardt and Dobbins, 1990; Hecht-Nielsen, 1990; Simpson, 1990; Freeman and Skapura, 1991; Hertz et al., 1991; Peretto, 1992; Gallant, 1993). The inputs and outputs may be to and from the external world or to other units within the network. The way in which each unit transforms its input depends on the so-called 'connection weight' (or 'connection strength') and 'bias' of the unit, which are modifiable. The output of each unit to another unit or the external world then depends on both its strength and bias and on the weighted sum of all its inputs, which are transformed by a (normally) nonlinear weighting function referred to as its activation function or squashing function. The great power of neural networks stems from the fact that it is possible to present ('train') them with known inputs (and outputs) and provide some form of learning rule which may be used, iteratively, to modify the strengths and biases until the outputs of the network as a function of the inputs correspond to the desired ('true') outputs.

We have seen that it is possible by computer simulation to determine steady-state variables such as fluxes and metabolite concentrations as a function of parameters such as the enzymatic rate constants and external metabolite concentrations. It is obviously then possible to change one or more of the parameters and to determine another set of associated variables, and so on. This can be done rapidly and automatically on a computer. Having acquired related sets of parameters and variables, we would then be in a position to train a neural network in which the (known) variables were the inputs and the parameters were the outputs. If the net successfully learned to reflect the correct parameters when presented with the variables, we would have solved our problem. We could then present the net with 'unknown' (i.e. experimentally determined) variables and ask it for the parameters. The correctness of the network's predictions would obviously be checked by running a simulation with the parameters provided by the network and seeing if they generated the variables used as the input to the net. Using this novel approach, we would in fact be able to obtain the (enzymatic) parameters of a metabolic network (and hence the control coefficients and elasticities) by measuring the variables alone. In this context, it is also worth mentioning that neural nets have been used to address cognate inverse problems for a number of dynamic systems (e.g. Chen and Billings, 1992; Masri et al., 1992, 1993; Veng-Pedersen and Modi, 1993; Brouwn et al., 1994).

We have therefore sought to assess the ability of simple feedforward neural networks updated via the backpropagation algorithm to solve the inverse metabolic problem described above, and to use the neural net approach to carry out control analyses that are not restricted to small changes to determine optimal (changes in) pathway parameters necessary for maximising fluxes (in biotechnology) or for minimising them (in pharmacology). It transpires that whilst such neural networks can indeed learn to predict parameters from variables for simple metabolic pathways, the method does not appear to scale well to large pathways. A preliminary account of this work has been presented (Sauro and Kell, 1992; Kell et al., 1993; Mendes and Kell, 1994).

2. Methods

2.1. Outline of the method

To be able to use this method, one must know in advance the structure of the metabolic pathway, i.e. the detailed sequence of reactions that the sub-

$$X_0 \xrightarrow[E_1]{} S_1 \xrightarrow[E_2]{} S_2 \xrightarrow[E_3]{} \cdots \xrightarrow[E_{n-1}]{} S_{n-1} \xrightarrow[E_n]{} X_n$$

Fig. 1. A generic metabolic pathway. X_0 and X_n are external metabolites, which are forced by some mechanism to have a constant concentration. S_1, S_2,...S_{n-1} and S_n are the 'internal' metabolites (note that these are variables). Arrow heads refer to the positive direction of fluxes; for generality, all steps are treated as kinetically reversible.

strate(s) molecules (X_0 in Fig. 1) undergo until the product(s) (X_n in Fig. 1) are formed. This includes any possible branches or cycles between the two. Additionally one must know the kinetic equations of each step in the metabolic sequence. The values of the parameters of these equations are the targets that the method tries to estimate. It is assumed that one can measure experimentally the concentrations of the intermediate metabolites (and the fluxes) in the steady state.

The method comprises two stages. In the first stage one generates a 'large' number of steady-state simulations of the pathway (we normally chose 100–500), each with a different set of values of the parameters (within some boundaries). All the unknown parameters must be varied, and any that are known must not. If some parameters are known to fall in a small interval one has the option of either not varying them (setting them to an estimate of the mean value), or to use that interval as the domain of variation of the parameter. For convenience, the result of this set of simulations is written to one single file as a table (one row per simulation). It is important that the parameters with unknown values be varied in a domain that contains the unknown value. As with conventional statistical methods neural networks are known to extrapolate poorly.

In the second stage one trains the neural network. This is the process of adjusting the weights of the connections between neurons. The data obtained in the previous stage are used to construct two sets of data: the first (the 'training set') is shown to the neural network repetitively until the later converges to a model, the second (the 'test set') is used to see how well the neural network performs with data it has not seen in the training stage. Both sets are composed of a group of columns that are fed to the input nodes of the neural network and another group which is the desired target of the neural network's outputs. For the inputs of the neural network we chose those parameters that we can measure or set together with the variables that we can measure and for the outputs the desired kinetic parameters. Note that parameters that are invariant need not be included in the neural network. Typically, therefore, we have at the inputs of the neural network the intermediate metabolite concentrations and the fluxes.

After these two stages are successfully completed, one can then use the trained neural network to estimate values of the desired parameters from measurements of the variables (those that are inputs of the network). This is a very rapid process (fractions of a second typically) and is well suited to be used routinely (Kell and Davey, 1992). If the predictions are accurate, the neural net is said to have generalised.

2.2. Metabolic pathways

The performance of the method proposed was tested using model metabolic pathways. The first of these (Fig. 2) is composed of two sequential steps with one intermediate. This pathway has only one true variable, although we use two variables in the neural network analysis (the flux and the concentration of the intermediate). In this case one variable is dependent on the other, in that the flux is here a simple function of the concentration of the intermediate. First we assume the two steps to follow first order kinetics. In this case the model has six parameters: the concentrations of the substrate and product of the pathway (which are clamped), the two equilibrium constants and two rate constants. Of these, the concentrations and

$$A \xrightarrow[E_1]{} B \xrightarrow[E_2]{} C$$

Fig. 2. A two-step metabolic pathway. A and C are external metabolites, which are forced by some mechanism to have a constant concentration. B is the only 'internal' metabolite (note that this is a variable). The two steps are named E_1 and E_2 but can be enzyme catalysed or simple first-order kinetics.

the equilibrium constants are easily determined, which leaves us with two unknown parameters that we will try to estimate with a neural network. This is a very simple example in the sense that one can actually express the values of the two unknown parameters in terms of the values of the other parameters and the variables in algebraic format. However one cannot do so in general, and indeed this is impossible when we consider the same pathway but with Michaelis-Menten kinetics for both steps. In this case the model has ten parameters as each Michaelis-Menten enzyme has three independent parameters (there is a fourth but this is the equilibrium constant which we can measure and is invariant in experiments)

2.3. The MLP architecture

In the training process, an algorithm is used to change the weights and bias of each node so that the output values are close enough to the desired values (obtained in the simulations). The algorithm that we have used is known as backpropagation of error (Rumelhart et al., 1986) and has been widely applied to multi-layer perceptrons. The number of units in the hidden layer(s) was varied as described in the text, and the squashing function used was the logistic function

$$O_j = 1/(1 + e^{-x}) \tag{1}$$

where O_j is the output of the node and x the weighted sum of its inputs.

2.4. Software and hardware platforms

All the results shown here were obtained on a Dell 4560/XE personal computer running the Microsoft Windows NT operating system. The neural network computations were made using the program WinNN, by Yaron Damon (danony@rebecca.its.rpi.edu). A trial version of this program is available free of charge on the internet from ftp://ftp.cica.indiana.edu/pub/ibmpc/windows3/programr/winnn093.z ip, or in the CICA CD-ROM. The simulations of the steady-state properties of metabolic pathways were carried out using the program GEPASI (Mendes, 1993), available free of charge on the internet from ftp://bmsdarwin.brookes.ac.uk/pub/software/ibmpc/gep208c.zip, in the CICA CD-ROM, or from one of us (PM).

3. Results

3.1. Two sequential first-order reactions

To create the data for training and checking the performance of the neural net, we proceeded as follows: the pathway of Fig. 2 was implemented on the metabolic simulator Gepasi (Mendes, 1993), where we set the concentration of A to 10 (in arbitrary units, but could easily be seen as e.g. mM). Gepasi was instructed to run 250 simulations, each of them with two random values for k_1 and k_2. Because their true values are unknown, it is important to allow these parameters to take values within a sufficiently large domain. This in most cases means through more than one order of magnitude, and possibly several. In the results shown here the boundaries were 0.01 and 100 (again in arbitrary units, but consistent with the units of concentration, so mM s^{-1} in the example above), covering four orders of magitude. Had we used a random uniform distribution for values of k_1 and k_2, we would have obtained many more points in the upper decade than in the three lower decades. However, because we want the neural network to learn the inverse representation within each of the four orders of magnitude (and not just the upper one), it is very important that we cover all the decades equally. Gepasi allows us to do so by generating pseudo-random numbers in logarithmic space (Mendes, 1993). The program puts the results of the simulations in a columnar file, one parameter or variable per column and one row per simulation. This is very close to the format of training and test set files required by most neural network programs, including WinNN. We only had to separate the original file in two and add one line at the top with the number of rows in the file and how many columns are inputs of the neural net. We chose to use the first 200 lines for the training set and the last 50 for the test set. These had to be post-processed in order to be used with the neural network. Because we are using the logistic as the squashing function, the outputs (k_1 and k_2) must be scaled between 0 and 1 as the logistic has these values as asymptotes. Again, just rescal-

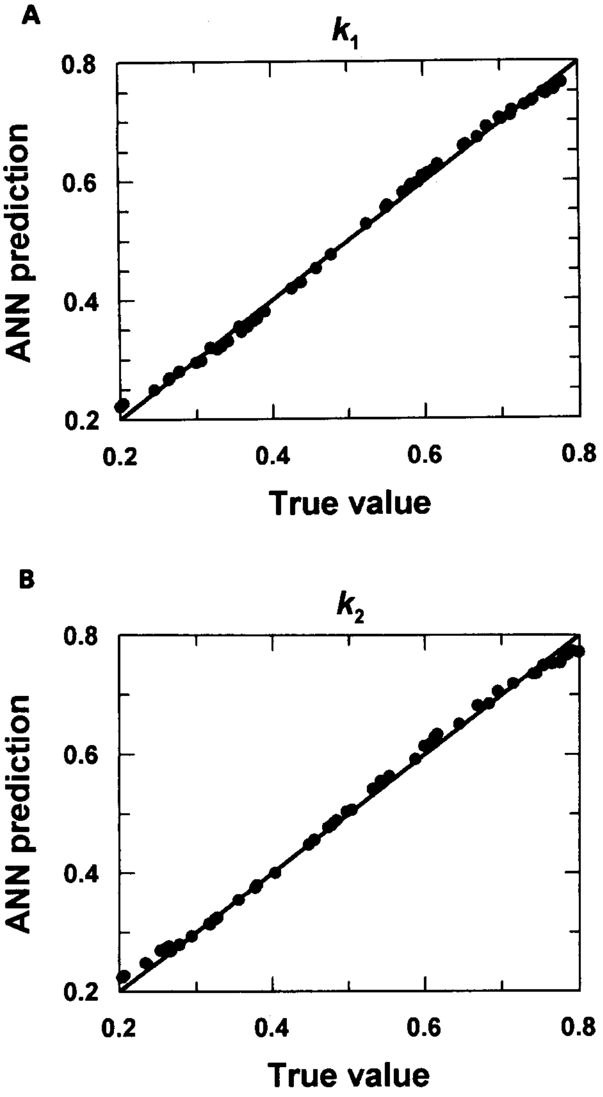


Fig. 3. Performance of the neural network model for the two step pathway with linear kinetics. Real values on the x-axis and the network prediction on the y-axis. Results are shown for the post-processed results of the test set (logged and rescaled between 0.2 and 0.8 covering values of k_1 and k_2 between 0.01 and 100. The 45° slope line indicates where the exact predictions would lie. A: k_1, B: k_2.

ing the values of k_1 and k_2 is not good enough; we have to rescale the logarithm of these values so as to cover all orders of magnitude equally. We have done so between the values of 0.2 and 0.8, trying to avoid saturation on the extremes of the logistic (see also Goodacre et al., 1993). Similar arguments apply to the inputs. Although there are no limits on the arguments that the logistic function can take, its output becomes saturated at around 3 (and −3). We therefore rescaled the logarithm of the inputs between −2 and 2. A very important point is that we rescaled the test and training sets together, not separately although each column was rescaled independently of the others (see also Neal et al., 1994).

The objective is for the neural network to learn the (inverse) relation between the rate constants and the metabolite concentration and the flux, so we must have the latter two applied at the inputs and the rate constants applied at the outputs of the net. We have chosen to use 10 hidden units between the inputs and outputs. Each unit in one layer is connected to all units in the next layer (a fully connected feedforward neural network). Random noise (±0.02 in the normalized scale) was added automatically (by the WinNN package) to the inputs of the neural network. In these conditions the learning algorithm converged in the order of 100 iterations; the performance of the trained neural network on the test set is displayed in Fig. 3. Note that these data were not those used to train the neural net, but were used only to test the predictive power of the network. As can be seen by inspection of Fig. 3 the neural network is quite capable of giving excellent predicitions for the values of the rate constants given measurements of the concentration of the intermediate and the steady-state flux. At the extremes of the scale there is some bias in the predictions (positive error in the low end and negative error in the high end). This is known to be an artefact introduced by the squashing function (e.g. Long et al., 1990; Goodacre et al., 1993; Jacobsson and Hagmann, 1993) and can be solved by using a linear function in the output nodes (Goodacre et al., 1995). Indeed we have confirmed that this does eliminate such a bias in the current example (results not shown).

3.2. *Two sequential Michaelis-Menten enzymes*

The next logical step up in the complexity of the pathway to be analysed by a neural net is to have the two steps catalysed by enzymes with Michaelis-Menten type kinetics. The reaction topology is the same as the previous example (Fig. 2). As in the previous case, we still only have two variables: the steady-state concentration of B and the flux, J; however there is now a larger number of parameters. Each Michaelis-Menten enzyme needs four parameters to be characterized: the forward limiting rate V^f, the reverse limiting rate V^r, the Michaelis constants for the substrate, K_s, and for the product, K_p. Because of thermodynamic constraints (the Haldane relationships, Haldane, 1930), one of these is not independent and so can be expressed as a function of the other three and the equilibrium constant. Doing this one can rearrange the Michaelis-Menten equation to use three of those parameters and the equilibrium constant:

$$v = \frac{V^f\left(S - \frac{P}{K_{eq}}\right)}{K_s + S + \frac{PV^f}{K_{eq}V^r}} \qquad (2)$$

This way, we are left with three unknown parameters per reaction (instead of one in the case of first order reactions), which makes a total of six. Preliminary results suggested that in this case we cannot easily train neural nets like those of the previous example to learn the inverse relation. This might be due to the fact that the relation may now be under-determined. For this limitation to be overcome we need to make available to the inputs of the neural network more information about the system. Fortunately there is one parameter of the pathway that can be manipulated experimentally: the concentration of the substrate, A. For one set of random values of the kinetic constants we can do several simulations with different, but known and fixed, values of [A]. The result is that we have various values of the steady-state concentration of B and flux, which we hope describe a set of kinetic constants uniquely (at least in the region we examine).

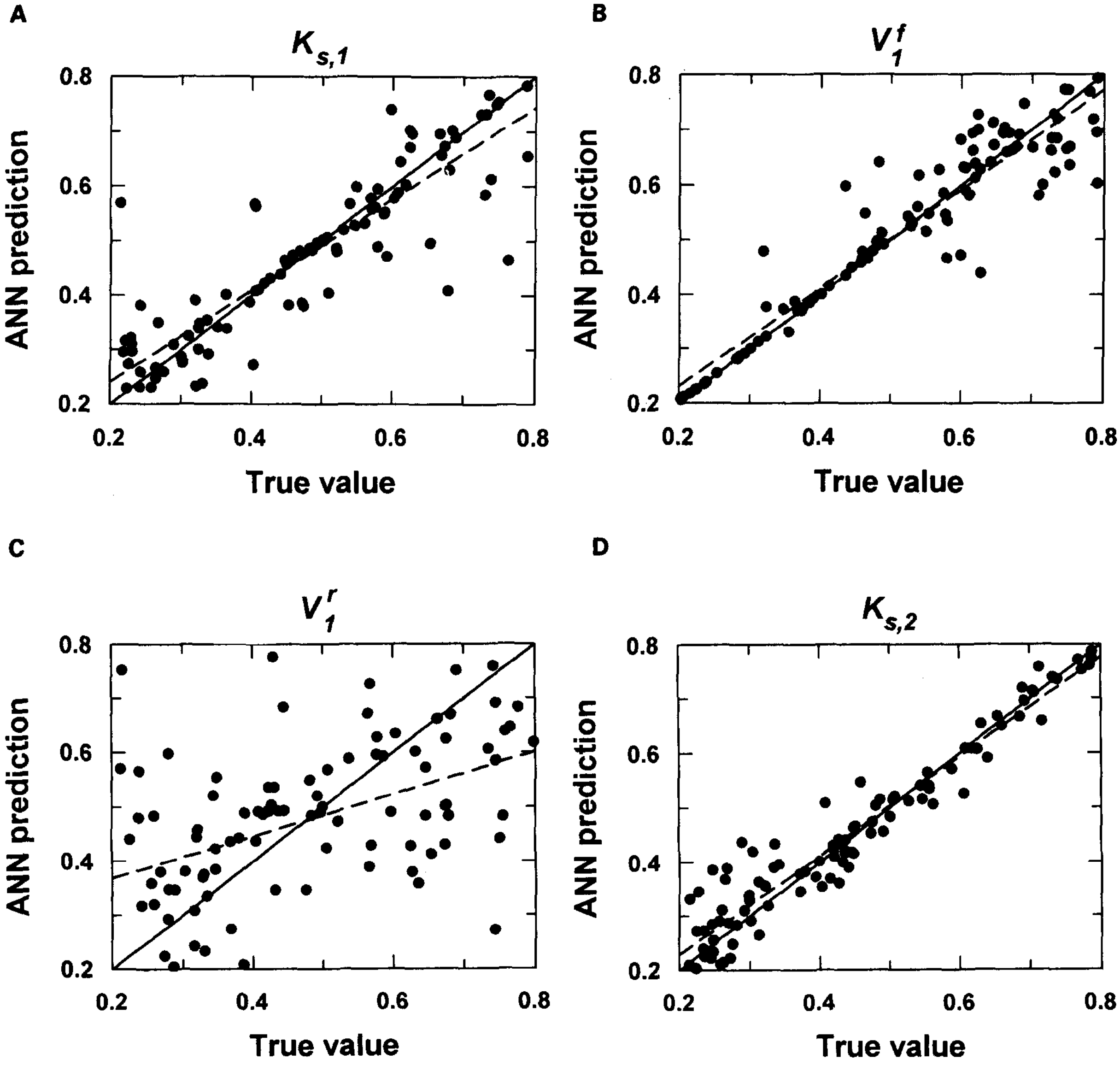
The simulations were set up as before with the following modifications: V^f_1, V^r_1, V^f_2, V^r_2, $K_{S,1}$ and $K_{S,2}$ varied between 0.1 and 10. We ran three simulations with [A] = 1, [A] = 3 and [A] = 5 respectively; 500 sets of kinetic parameters were generated (1500 simulations in total), where the first 400 were chosen for the training set and the last 100 for the test set.

With respect to the neural network model, we had to do some optimizations before obtaining a neural net model that could learn the inverse relationship. First we noticed that as the nonlinearity in the kinetics of the reactions increased, a single layer of hidden units connecting the inputs to the outputs was not enough (see also Cybenko, 1989). Four-layer fully connected feed-forward neural nets (four-layer perceptrons) appear to be better at learning the inverse relation and so we have used them in this and subsequent examples. We also observed quicker and better learning when we used one neural network to learn each parameter than when we used one single neural network to learn them all (see also Goodacre et al., 1994). Given this, we trained six neural networks each with a topology of 6-15-10-1 (6 inputs, 15 nodes on the first hidden layer, 10 nodes on the second hidden layer and 1 output). The performance of these networks with unseen data is depicted in Fig. 4. As can be seen from Fig. 4C and 4F, we did not manage to train neural networks that were able to relate the six inputs and each of the reverse limiting rates. This was not a problem of lack of generalisation but rather of lack of convergence. However, the inverse relations of the other kinetic parameters were modelled by the neural networks with considerable success (Fig. 4A, B, D, and E). We also repeated the whole process but this time setting the two reverse limiting rates to fixed arbitrary values (as if we knew their value a priori). In this case we managed to train a single neural network to learn the inverse relationship between the variables and the rest of the kinetic parameters (results not shown) even better than those in Fig. 4A, B, D and E.

3.3. Three sequential Michaelis-Menten enzymes

Following the logic of the two previous experiments, we increased the complexity of the pathway by adding one extra step catalysed by a Michaelian enzyme (Fig. 5). We now have nine kinetic constants, with only three variables: [B], [C] and the flux J. We repeated the procedure described in the previous section, from which we obtain data for nine neural networks with nine inputs and one output each. Each of these will be used to model the inverse relation between the nine apparent variables (they are truly only three but triplicated by the use of three different concentrations of the substrate A, as before) and each of the kinetic parameters. The results are depicted in Fig. 6A-I.

Once again we observed that the neural networks are not able to learn the inverse relationship between the concentrations and flux and the reverse limiting rates (Fig. 6C, F and I). In general the performance of the neural networks for this three-enzyme pathway is poorer than for the two-enzyme pathway, in particular with respect to the Michaelis constants. We also note that the parameter best modelled by the neural networks is the limiting rate of the last enzyme in the pathway. This might suggest that there is a simple relation between this parameter and the concentrations and flux; however this is not apparent by simple inspection of plots of [B], [C] and J versus V^f_3 (data not shown).



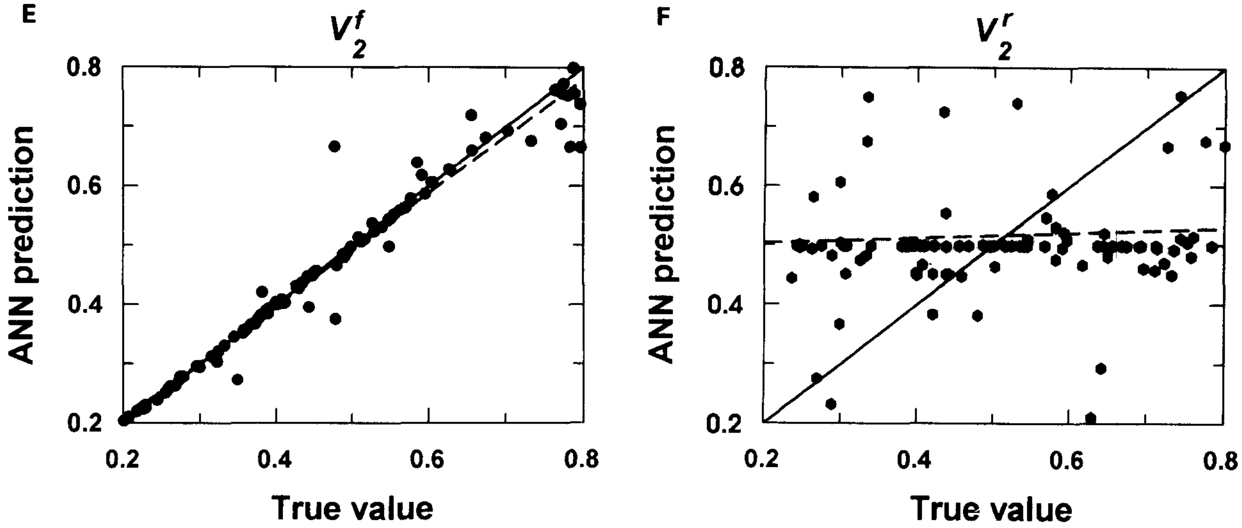


Fig. 4. Performance of the neural network model for the two step pathway with Michaelian kinetics. Real values on the x-axis and the network prediction on the y-axis. Results are shown for the post-processed results of the test set (logged and rescaled between 0.2 and 0.8 covering values of V^f_1, V^r_1, V^f_2, V^r_2, $K_{S,1}$ and $K_{S,2}$ between 0.1 and 10. The continuous 45° slope lines indicate where the exact predictions would lie, the broken lines are the least-squares linear fits from the plotted data. A: $K_{S,1}$ (best fit line with correlation coefficient of 0.8696); B: V^f_1 (best fit line with correlation coefficient of 0.9404); C: V^r_1 (best fit line with correlation coefficient of 0.4006); D: $K_{S,2}$ (best fit line with correlation coefficient of 0.9627); E: V^f_2 (best fit line with correlation coefficient of 0.9807); F: V^r_2 (best fit line with correlation coefficient of 0.06067).

4. Discussion

Inverse problems are those in which one knows values for a set of variables of a model and wants to deduce from that the values of the parameters that were responsible for the system to attain that state. One example, known as inverse kinematics, is that of calculating the motion of an object in space from a desired final position. This is a problem faced by the brain when it controls the limbs. Biological brains are very good at solving this particular inverse problem, while in robotics some acceptable solutions have also been developed, some based on artificial neural networks (Kuperstein, 1987; Miller, 1987; Jordan, 1992).

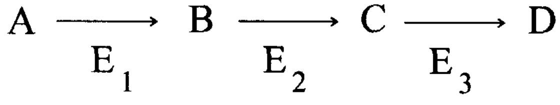
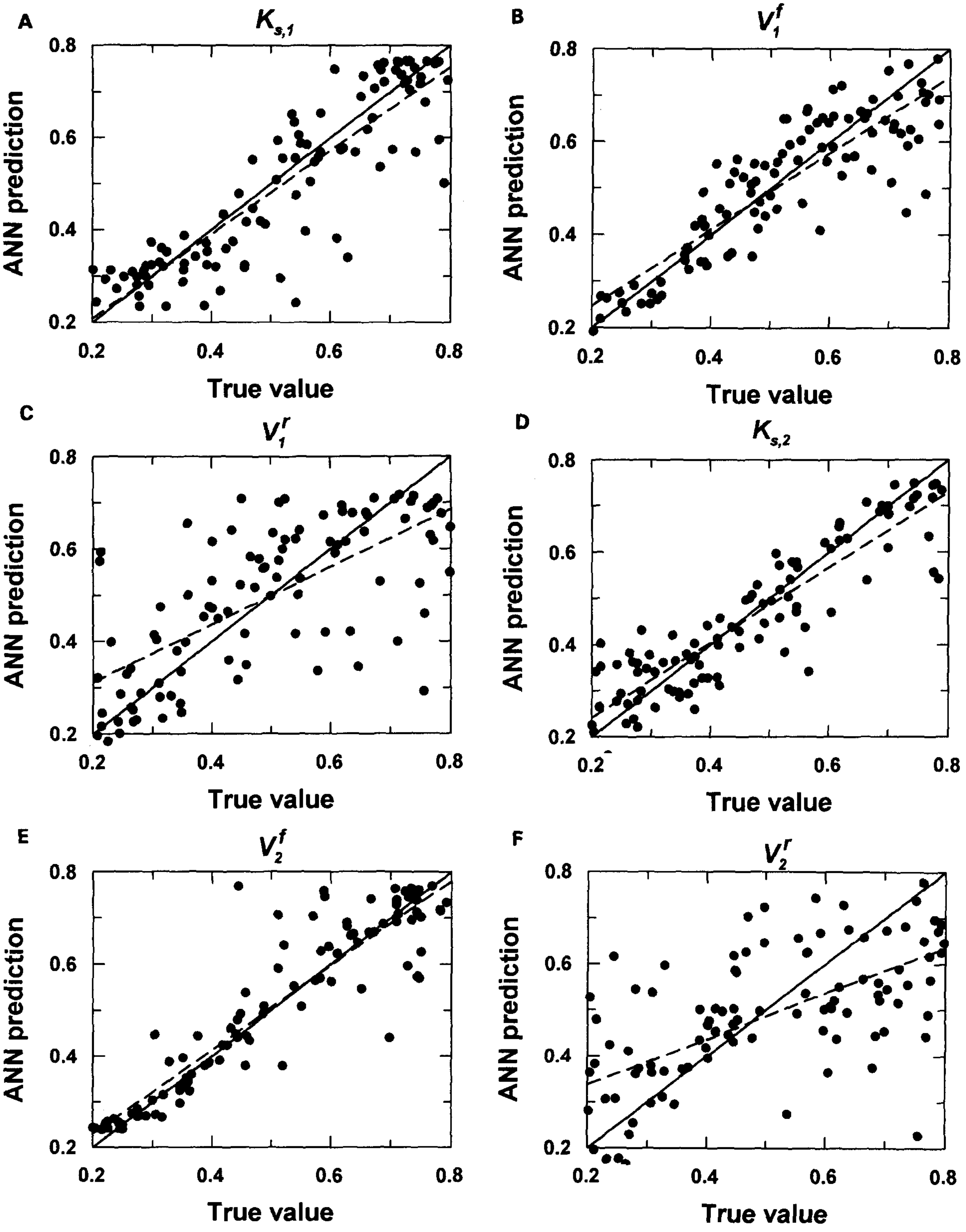


Fig. 5. A three-step metabolic pathway. A and D are external metabolites, which are forced by some mechanism to have a constant concentration. B and C are the 'internal' metabolites (variables). The three steps are catalysed by enzymes E_1, E_2 and E_3.

Here we discuss a particular inverse problem in metabolism, which is that of identifying the values of steady-state kinetic parameters from the values of measured concentrations and fluxes. The corresponding forward problem is easily solvable by integration of systems of differential equations (see e.g. Heinrich et al., 1977; Hayashi and Sakamoto, 1986), and this is a process easily carried out by using computer programs specifically designed for this purpose (Holzhütter and Colosimo, 1990; Cornish-Bowden and Hofmeyr, 1991; Sauro and Fell, 1991; Mendes, 1993). We have here developed a method aimed at solving this particular inverse problem that uses a combined approach: in the first stage a model of the metabolic system is set up and repeatedly simulated in the forward direction using pseudo-random values for the parameters, while in a second stage the data generated by the simulations are used to train a


A
$K_{s,1}$
B
V_1^f
C
V_1^r
D
$K_{s,2}$
E
V_2^f
F
V_2^r
ANN prediction
True value
0.2
0.4
0.6
0.8

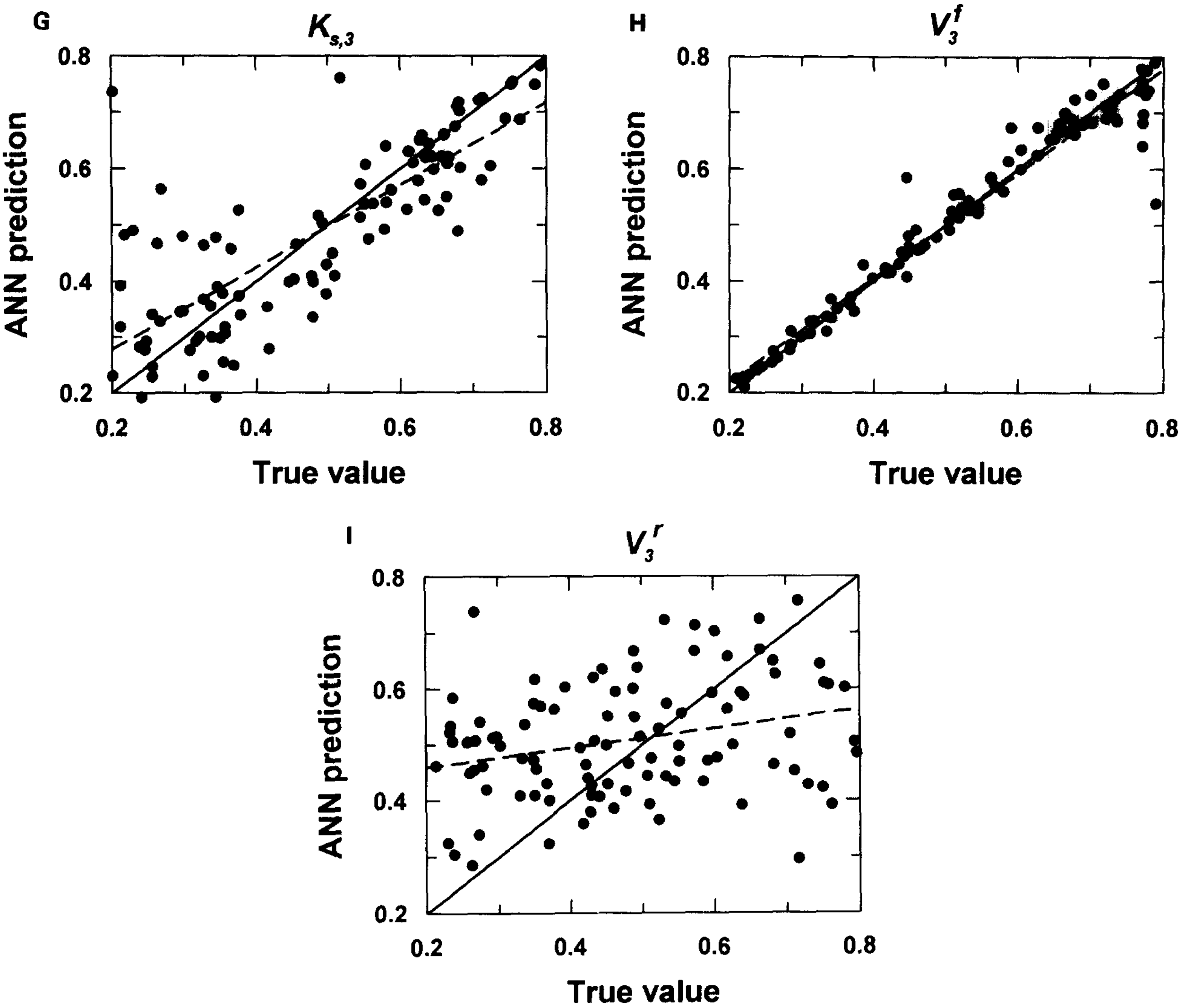


Fig. 6. Performance of the neural network model for the three step pathway with Michaelian kinetics. Real values on the x-axis and the network prediction on the y-axis. Results are shown for the post-processed results of the test set (logged and rescaled between 0.2 and 0.8 covering values of $K_{S,1}$, V^f_1, V^r_1, $K_{S,2}$, V^f_2, V^r_2, $K_{S,3}$, V^f_2 and V^r_3 between 0.1 and 10. The continuous 45° slope lines indicate where the exact predictions would lie, the broken lines are the least-squares linear fits from the plotted data. A: $K_{S,1}$ (best fit line with correlation coefficient of 0.8794); B: V^f_1 (best fit line with correlation coefficient of 0.8673); C: V^r_1 (best fit line with correlation coefficient of 0.6908); D: $K_{S,2}$ (best fit line with correlation coefficient of 0.9043); E: V^f_2 (best fit line with correlation coefficient of 0.9206); F: V^r_2 (best fit line with correlation coefficient of 0.6229); G: $K_{S,3}$ (best fit line with correlation coefficient of 0.7987); H: V^f_3 (best fit line with correlation coefficient of 0.9740); I: V^r_3 (best fit line with correlation coefficient of 0.2664).

feedforward neural network in which the inputs and outputs are now in the opposite order to that in the simulation, i.e. the variables (concentrations and fluxes) are at the inputs and the (kinetic) parameters at the outputs. If the neural network converges and generalises, one has then solved the problem within the domain in parameter space defined by the boundaries of the random values in the forward simulations. Moreover, if convergence is obtained one has solved the inverse problem of all metabolic pathways that have the same structure and kinetics as the model used. For example the results of Fig. 3 apply to all systems of two sequential first order reactions with values of forward rate constants between 0.01 and 100. Artificial neural networks are good candidates for this task as they have been demonstrated to fit any arbitrary nonlinear function given enough number of nodes in the hidden layers (Hornik et al., 1989; White, 1992). Methods designed for linear systems are not expected to perform well. We did in fact try to apply partial least squares (PLS, Martens and Naes, 1989) to the data corresponding to Fig. 6, but this technique indeed failed to model the data (results not shown).

The results described above show us that the method proposed here seems to perform well for some of the parameters but, apparently, not for all of them. The neural networks were not able to converge in reasonable time in the case of the reverse limiting rates (V^r). This is only a moderate limitation in fact, as these kinetic parameters tend to have only a small influence in determining the values of the variables in the forward problem (Mendes and Kell, 1994). This problem may also be alleviated if one knows that some of the other parameters lie within a narrower range of values. It is not unreasonable to think that in experimental settings one may know at least the order of magnitude of the V^f and K_S parameters.

The performance of the neural networks for the data from the three-enzyme pathway, in which there is a larger average error of the neural network estimates as compared to those of the two-enzyme pathway, suggest that this method does not scale well, however, at least using fully interconnected feedforward nets trained according to the standard backpropagation algorithm.

Acknowledgements

This work was supported by The Wellcome Trust. We thank Mark Neal and Royston Goodacre for helpful suggestions, and Herbert Sauro for his contribution to this project.

References

Aleksander, I. and Morton, H., 1990, An Introduction to Neural Computing. Chapman and Hall, London.

Amit, D.J., 1989, Modeling Brain Function. The World of Attractor Neural Networks. Cambridge University Press, Cambridge.

Anthony, M. and Biggs, N., 1992, Computational learning theory. Cambridge University Press, Cambridge.

Backman, K., O'Connor, M.J., Maruya, A., Rudd, E., McKay, D., Balakrishnan, R., Radjai, M., Dipasquantonio, V., Shoda, D., Hatch, R. and Venkatasubramanian, K., 1990, Genetic-engineering of metabolic pathways applied to the production of phenylalanine. Ann. NY Acad. Sci. 589, 16–24.

Bailey, J.E., 1991, Toward a science of metabolic engineering, Science 252, 1668–1675.

Beale, R. and Jackson, T., 1990, Neural Computing: An Introduction. Adam Hilger, Bristol.

Brouwn, G.G., Krijgsman, A.J., Verbruggen, H.B. and Bruijn, P.M., 1994, Single-layer networks for nonlinear system identification. Engng. Applic. Artif. Intell. 3, 227–243.

Cameron, D.C. and Tong, I., 1993, Cellular and metabolic engineering: an overview. Appl. Biochem. Biotechnol. 38, 105–140.

Chen, S. and Billings, S.A., 1992, Neural networks for nonlinear dynamic system modelling and identification. Int. J. Control 56, 319–346.

Cowan, J.D. and Sharp, D.H., 1988, Neural nets. Q. Rev. Biophys. 21, 365–427

Bailey, J.E., Birnbaum, S., Galazzo, J.L., Khosla, C. and Shanks, J.V. (1990) Strategies and challenges in metabolic engineering, Ann. NY Acad. Sci. 589, 1–15.

Cornish-Bowden, A. and Hofmeyr, J.-H.S., 1991, MetaModel: a program for modelling and control analysis of metabolic pathways on the IBM PC and compatibles. Comput. Applic. Biosci. 7, 89–93.

Cybenko, G., 1989, Approximation by superposition of a sigmoidal function. Mathemat. Control Signals Syst. 2, 303–314.

Eberhart, R.C. and Dobbins, R.W., 1990, Neural Network PC Tools. Academic Press, London.

Fell, D.A., 1992, Metabolic control analysis — a survey of its theoretical and experimental development. Biochem. J. 286, 313–330.

Freeman, J.A. and Skapura, D.M., 1991, Neural Networks. Addison-Wesley, Reading, Mass.

Gallant, S.I., 1993, Neural Network Learning. MIT Press, Cambridge, Mass.

Garey, M. and Johnson, D., 1979, Computers and intractability: a guide to the theory of NP-completeness. Freeman, San Francisco.

Goodacre, R., Edmonds, A.N. and Kell, D.B., 1993, Quantitative analysis of the pyrolysis-mass spectra of complex mixtures using artificial neural networks: application to casamino acids in glycogen. J. Anal. Appl. Pyrolysis 26, 93–114.

Goodacre, R., Neal, M.J. and Kell, D.B., 1994, Rapid and quantitative deconvolution of the pyrolysis mass spectra of complex binary and tertiary mixtures using multivariate calibration and artificial neural networks. Anal. Chem. 66, 1070–1085.

Goodacre, R., Trew, S., Wrigley-Jones, C., Saunders, G., Neal, M.J., Porter, N. and Kell, D.B., 1995, Rapid and quantitative analysis of metabolites in fermentor broths using pyrolysis mass spectrometry with supervised learning: application to the screening of *Penicillium chryosgenum* fermentations for the overproduction of penicillins. Anal. Chim. Acta, 313, 25–43.

Hayashi, K. and Sakamoto, N., 1986,, Dynamic analysis of enzyme systems. An introduction. Springer-Verlag , Berlin.

Hecht-Nielsen, R., 1990, Neurocomputing. Addison-Wesley, Massachusetts.

Heinrich, R. and Rapoport, T.A., 1974, A linear steady-state treatment of enzymatic chains. General properties, control and effector strength. Eur. J. Biochem. 42, 89–95.

Heinrich, R., Rapoport, S.M. and Rapoport, T.A., 1977, Metabolic regulation and mathematical models. Progr. Biophys. Molec. Biol. 32, 1–82.

Hertz, J., Krogh, A. and Palmer, R.G., 1991, Introduction to the Theory of Neural Computation. Addison-Wesley, Redwood City.

Holzhütter, H.G. and Colosimo, A., 1990, SIMFIT: a microcomputer software-toolkit for modelistic studies in biochemistry. Comput. Applic. Biosci. 6, 23–28.

Hornik, K., Stinchcombe, M. and White, H., 1989, Multilayer feedforward neural networks are universal function approximators. Neural Networks 2, 359–366.

Ikeda, M. and Katsumata, R., 1992, Metabolic engineering to produce tyrosine or phenylalanine in a tryptophan-producing *Corynebacterium glutamicum.* Appl. Env. Microbiol. 58, 781–785.

Jacobsson S.P. and Hagmann, A., 1993, Chemical-composition analysis of carrageenans by infrared-spectroscopy using partial least-squares and neural networks. Anal. Chim. Acta 284, 137–147.

Jordan, M.I., 1992, Constrained supervised learning. J. Math. Psych. 36, 396–425.

Kacser, H. and Burns, J.A., 1973, The Control of Flux. In Rate Control of Biological Processes, D.D. Davies (ed.) (Cambridge: Cambridge University Press), Symp. Soc. Exp. Biol. 27, 65–104.

Kacser, H. and Acerenza, L., 1993, A universal method for achieving increases in metabolite production. Eur. J. Biochem. 216, 361–367.

Katsumata, R. and Ikeda, M., 1993, Hyperproduction of tryptophan in *Corynebacterium glutamicum* by pathway engineering. Bio/Technology 11, 921–925.

Kell, D.B. and Westerhoff, H.V., 1986, Metabolic Control Theory: its role in microbiology and biotechnology. FEMS Microb. Rev. 39, 305–320.

Kell, D.B. and Davey, C.L., 1992, On fitting dielectric spectra using artificial neural networks. Bioelectrochem. Bioenerg. 28, 425–434.

Kell, D.B., Davey, C.L., Goodacre, R. and Sauro, H., 1993, When going backwards means progress: on the solution of biochemical inverse problems using artificial neural networks, in: Modern Trends in Biothermokinetics, S. Schuster, M. Rigoulet, R. Ouhabi and J.-P. Mazat, (eds.) (Plenum Press, New York and London), pp. 109–113.

Kohonen, T., 1989, Self-Organization and Associative Memory. 3rd Edition, Springer, Heidelberg.

Kuperstein, M., 1987, Adaptive visual-motor coordination in multijoint robots using parallel architecture. IEEE Conference on Robotics and Automation pp. 1595–1602.

Long, J.R., Gregoriou, V.G. and Gemperline, P.J., 1990, Spectroscopic calibration and quantitation using artificial neural networks. Anal. Chem. 62, 1791–1797.

Martens, H. and Naes, T., 1989, Multivariate calibration. Wiley, New York.

Masri, S.F., Chassiakos, A.G. and Caughey, T.K., 1992, Structure unknown non-linear dynamic systems: identification through neural networks. Smart Mater. Struct. 1, 45–56.

Masri, S.F., Chassiakos, A.G. and Caughey, T.K., 1993, Identification of nonlinear dynamic systems using neural networks. J. Appl. Mechan. 60, 123–133.

McClelland, J.L. and Rumelhart, D.E., 1988, Explorations in Parallel Distributed Processing; A Handbook of Models, Programs and Exercises. MIT Press, Cambridge, Massachustetts.

Mendes, P., 1993, GEPASI: a software package for modelling the dynamics, steady states and control of biochemical and other systems. Comput. Applic. Biosci. 9, 563–571.

Mendes, P. and Kell, D.B., 1994, Simultaneous estimates of single-enzyme kinetic parameters for multi-enzymetic systems using artificial neural networks, in: What is controlling Life?, E. Gnaiger, F.N. Gellerich and M. Wyss (eds.) (Innsbruck University Press, Innsbruck), Modern Trends in Biothermokinetics 3, 302–304.

Miller, W.T., 1987, Sensor-based control of robotic manipulators using a general learning algorithm. IEEE J. Robot. Autom. 3, 157–165.

Neal, M.J., Goodacre, R. and Kell, D.B., 1994, On the analysis of pyrolysis mass spectra using artificial neural networks. Individual input scaling leads to rapid learning. Proceedings of the World Congress on Neural Networks, 1994, San Diego, pp. I-318–I-323.

Niederberger, P., Prasad, R., Miozzari, G. and Kacser, H., 1982, A strategy for increasing an in vivo flux by genetic manipulations. The tryptophan system of yeast. Biochem. J. 287, 473–479.

Pao, Y-H., 1989, Adaptive Pattern Recognition and Neural Networks. Addison-Wesley, Reading, Massachusetts.

Peretto, P., 1992, An Introduction to the Modelling of Neural Networks. Cambridge University Press, Cambridge.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J., 1986, Learning internal representations by error propagation. in: D.E. Rumelhart, J.L. McClelland and the PDP Research Group (eds.) (MIT Press, Cambridge Mass), Parallel Distributed Processing: Explorations in the Microstructures of Cognition 1, 318–362.

Sauro, H.M. and Fell, D.A., 1991, SCAMP: a metabolic simulator and control analysis program. Math. Comput. Modelling 15 (12), 15–28.

Sauro, H.M. and Kell, D.B., 1992, Evaluation of steady-state kinetic parameters of metabolic pathways using neural networks. Binary 4, 189–190.

Simpson, P.K., 1990, Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations, Pergamon Press, New York.

Small, J.R. and Kacser, H., 1993, Responses of metabolic systems to large changes in enzyme activities and effectors. 2. The linear treatment of branched pathways and metabolite concentrations. Assessment of the general non-linear case. Eur. J. Biochem. 213, 625–640.

Stephanopoulos, G. and Vallino, J.J., 1991, Network rigidity and metabolic engineering in metabolite overproduction. Science 252, 1675–1681.

Tong, I-T., Liao, H.H. and Cameron, D.C., 1991, 1,3-propanediol production by Escherichia coli expressing genes from the Klebsiella pneumoniae DHA regulon. Appl. Env. Microbiol. 57, 3541–3546.

Veng-Pedersen, P. and Modi, N.B., 1993, Application of neural networks to pharmacodynamics. J. Pharm. Sci. 82, 918–926.

Wasserman, P.D., 1989, Neural Computing: Theory and Practice. Van Nostrand Reinhold, New York.

White, H., 1992, Artificial Neural Networks: Approximation and Learning Theory. Blackwell, Oxford.