LRP 430/91 July 1991

Invited paper presented at the
7th Conference REAL TIME '91
at KFA Jülich, FRG, 24-28 June 1991

J.B. Lister, H. Schnurrenberger, N. Staeheli,
N. Stockhammer, P.A. Duperrex and
J.-M. Moret

Neural Networks in Front-End Processing and Control

J.B. Lister, H. Schnurrenberger, N. Staeheli, N. Stockhammer, P.A. Duperrex and J.-M. Moret
Centre de Recherches en Physique des Plasmas
Association Euratom - Confédération Suisse
Ecole Polytechnique Fédérale de Lausanne
21, Av. des Bains - CH-1007 Lausanne - Switzerland

Abstract

Research into neural networks has gained a large following in recent years. In spite of the long term timescale of this Artificial Intelligence research, the tools which the community is developing can already find useful applications to real practical problems in experimental research. One of the main advantages of the parallel algorithms being developed in AI is the structural simplicity of the required hardware implementation, and the simple nature of the calculations involved. This makes these techniques ideal for problems in which both speed and data volume reduction are important, the case for most front-end processing tasks.

In this paper we illustrate the use of a particular neural network known as the Multi-Layer Perceptron as a method for solving several different tasks, all drawn from the field of Tokamak research. We also briefly discuss the use of the Multi-Layer Perceptron as a non-linear controller in a feedback loop.

We outline the type of problem which can be usefully addressed by these techniques, even before the large-scale parallel processing hardware currently under development becomes cheaply available. We also present some of the difficulties encountered in applying these networks.

I. INTRODUCTION

Although a computer can beat a human at arithmetic tasks, it has trouble distinguishing a Mercedes from a BMW. This implies that certain biological processes are superior in handling large quantities of information,.although they have intrinsically slower cycle times than present day computers. The quest for artificial systems which imitate some of the biological features which lead to this performance motivated even the earliest computer research, and this quest continues, driving the study of Artificial Neural Systems.

Current fields of research include vision, robotic control, speech recognition, handwriting analysis, sonar and radar analysis, time-series analysis. Existing commercial applications cover character recognition, credit risk assessment and production line quality control.

There is a correspondingly large number of classes of neural network, beyond the scope of this paper. The reader is referred to [1,2] for more information. Some networks operate with binary values, some produce an output which is a "recognized" version of the input (auto-associative) and some produce an independent "answer" (hetero-associative). In some, the information circulates, and in others it flows directly towards a set of outputs. In this short paper we only discuss one network, the Multi-Layer Perceptron (MLP). This is generally a hetero-associative network which we define in Section III. We introduce the ideas behind it by using simple analogies in Section II. We will show how MLP networks can solve practical front end processing tasks, by describing four simple but real examples, all taken from Tokamak research, Sections IV-VII. Another important and expanding domain of application of neural networks is non-linear dynamic control, which we introduce with a simple example in Section VIII. We briefly mention hardware implementation in Section IX, some practical difficulties in Section X and some conclusions in Section XI. There is no attempt to provide a review of the enormous bibliography, although a few papers are cited where appropriate.

II. NEURAL NETWORKS

As already stated, there is a large variety of neural networks being explored at present, and in this paper we choose to restrict ourselves to one, the MLP, which yields a multi-dimensional mapping. It accepts a large number of inputs and must provide one or more outputs whose values are continuous functions of the given input values. It possesses a forward-pass structure; this means that no information processed is re-used in a previous part of the processing. This class of network is particularly simple to understand and to use. It is a wide class and corresponds to most data analysis, data reduction, decisional and interpretative tasks.

Figure 1 shows four networks which are simple to understand and which perform obvious functions. We discuss these so that the use of the MLP subsequently appears a natural generalisation of some simple ideas. The first, Fig. 1(a), illustrates an averaging network. The inputs are A and B; the output is (A+B)/2. There are two weights (each 0.5) connecting the inputs to a summing processing element (PE), and a weight (1.0) connecting this PE to the output.

Figure 1(b) shows 4 input weights [1,1,-1,-1] into the summing PE which has an output value of A+B-C-D. This is passed through a non-linear PE which is a threshold function. The output θ(A+B-C-D) is a binary level indicating whether A+B>C+D.

Figure 1(c) shows two inputs into two logarithmic PEs, whose outputs are summed and passed through an exponentiation PE. The output is simply the product AB.

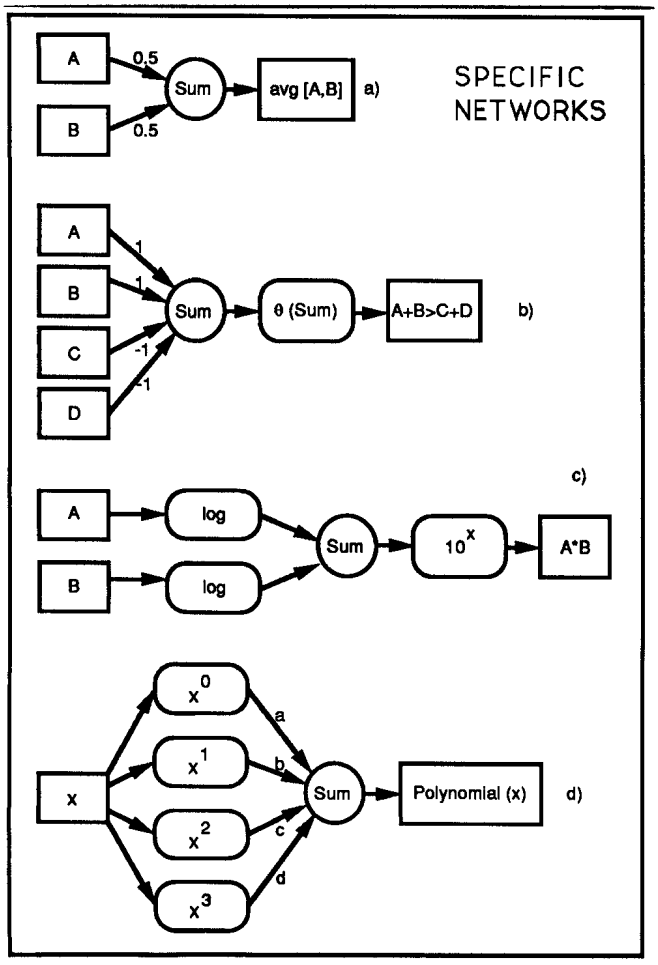

Fig. 1 Schematics of four arithmetic neural networks calculating : (a) the average of A and B, (b) a logical value representing A+B>C+D, (c) the product AB and (d) a polynomial expansion $a+bx+cx^2+dx^3$.

Figure 1(d) illustrates a single input fanned out with a unity weight to four non-linear PEs, each raising its input to a power 0-3. Their output weights and a summing PE define an output equal to $a+bx+cx^2+dx^3$.

Figure 2 shows a more complicated network with 100 inputs equal to a signal defined by a time-series. These are connected via weights equal to $\cos(n\omega t_i)$ (n=1...10) to 10 summing PEs. The outputs are the real part of the first ten harmonic Fourier components of the input time-series.

These five simple illustrations of interconnected forward-pass networks can be considered a priori as valid "neural" networks. They are all buildable; they are all useful; they are all simple; they are all correct; they are all based on well-established arithmetic; but they are all highly specific, and only apply to the problem they were designed for. As a given problem becomes larger and the operations become more complex, any conventional network will have a smaller ressemblance to any other existing network. These networks, although of extreme importance to the physicist or engineer, have strictly no interest as Artificial Neural Systems. Therefore it would be preferable to look for a more general network capable of being adapted to solve a larger class of tasks.

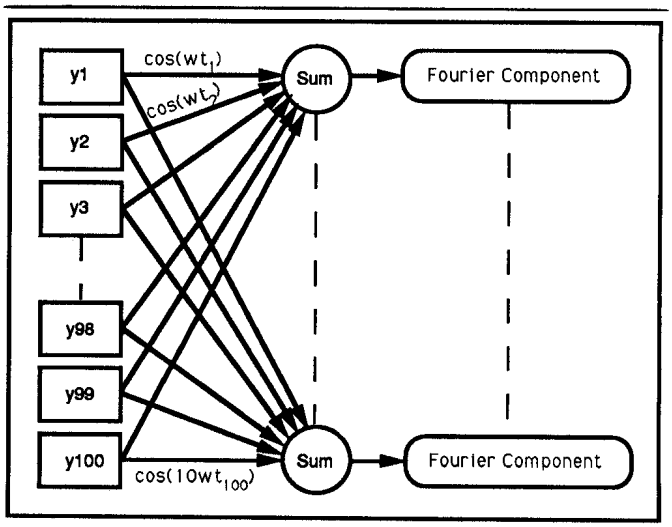

Fig. 2 A schematic of the Fourier components of a time-series.

The quest for a neural network which has some similitude with biological networks focusses on two main biological observations - a high multiplicity of similar cells and a large degree of connectivity. Early research into mathematical models of highly interconnected networks flourished in the 1960's to be hurriedly extinguished on the basis of a false alarm. The wounded discipline only took off again in the early 1980's. The false alarm had been the undeniable inability of a single matrix (known as the Perceptron) to emulate the exclusive-OR function of two binary inputs. The Perceptron was subsequently replaced by a more powerful but more complex structure known as the Multi-Layer Perceptron, described in the following section.

Both the Perceptron and the Multi-Layer Perceptron have the same goal - to reproduce a given input-output mapping using a multiplicity of simple but most importantly identical processing elements. The structures of Figs 1 and 2 must and can be replaced by a homogeneous neural network which can be made to be almost as good as the true functions they emulate, for a given range of input values. These structures can then be considered as generic transfer functions, or mappings, which can be adapted to solve any particular problem. The homogeneity of the solution offers a further advantage, namely highly parallel calculations, suited to future parallel processing.

These characteristic features - parallelism, homogeneity and generality - are what motivate the research

into neural networks, and which render them extremely attractive for many front-end processing tasks.

III. THE MULTI-LAYER PERCEPTRON

The MLP is an explicit non-linear and continuous mathematical relationship between a multi-variable input data vector (in our case experimentally measured signals) and a multi-variable output data vector (in our case parameters to be estimated on the basis of the available data). The MLP is often represented schematically as in Fig. 3. The input vector

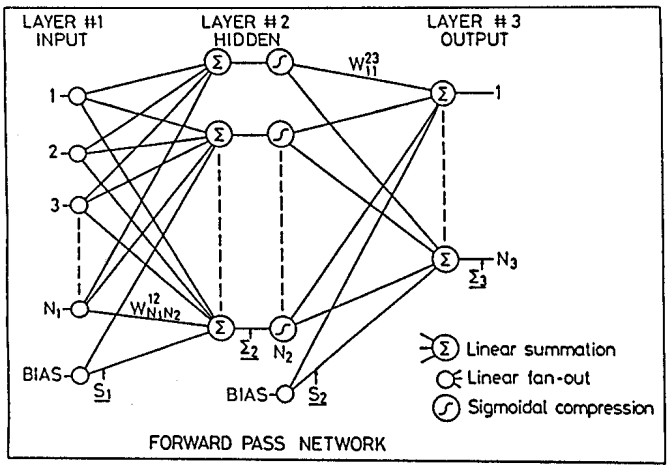

Fig. 3 The schematic description of a Multi Layer Perceptron

of dimension N_1, is linearly transformed by a matrix $\underline{W}^{12}$ into an intermediate vector of dimension N_2, known as the hidden layer. The size of this intermediate vector , which may be smaller or greater than the input vector, is the only design choice when specifying the MLP configuration. The hidden layer vector is then passed element by element through a compression function s, known as a sigmoidal function, which is continuous, differentiable and monotonically increasing. The sigmoidal function we use is symmetric and bounded (±1) :

$$s(x) = \frac{2}{(1 + e^{-x})} - 1 \qquad (1)$$

The compressed vector is then linearly transformed by a second matrix $\underline{W}^{23}$ into an output vector of dimension N_3. Such a configuration is referred to as a 1-hidden-layer MLP (an MLP-1); if we recompress and re-transform the output vector to produce a new output vector, we have a 2-hidden-layer MLP (an MLP-2) and so on. If we make a simple linear projection, we have the historically special case of an MLP-0.

The different MLP configurations share the property of transforming a vector of input data (S_i) into a vector of output data (x_k). This mapping can be written as $x_k = G_k(S)$. Any particular MLP-1 mapping G is defined by the values of the weights in the matrices $\underline{W}^{12}$ and $\underline{W}^{23}$. A wide range of literature can be found on the generality of MLP-1 and MLP-2 mappings [3,4]. It has been demonstrated that all bounded continuous functions can be approximated using the sigmoid (1) by some MLP-2 over a given volume in the space of S, given suitably large transformation matrices. A similar demonstration has been proposed for the case of the MLP-1. The work in this paper is restricted to MLP-1 mappings which are found to be adequate for the particular problems investigated.

IV. TASK 1 - OVERLAPPING GAUSSIANS

The first task we look at is the analysis of data from a spectrometer assumed to cover the range of two Gaussian spectral lines. These lines are of different amplitudes, central wavelengths and widths $A_{1,2}$, $f_{01,2}$, $df_{1,2}$, as illustrated in Fig. 4. The value of the signal at each of 20 sampling points (i) is given by

$$y(f_i ; A_{1,2}, f_{01,2}, df_{1,2}) = \sum_{j=1,2} A_j \exp(-(f_{0j}-f_i)^2/df_j^2) \qquad (2)$$

We are not interested in the raw data, but only need a real time estimate of the peak widths and centres, which

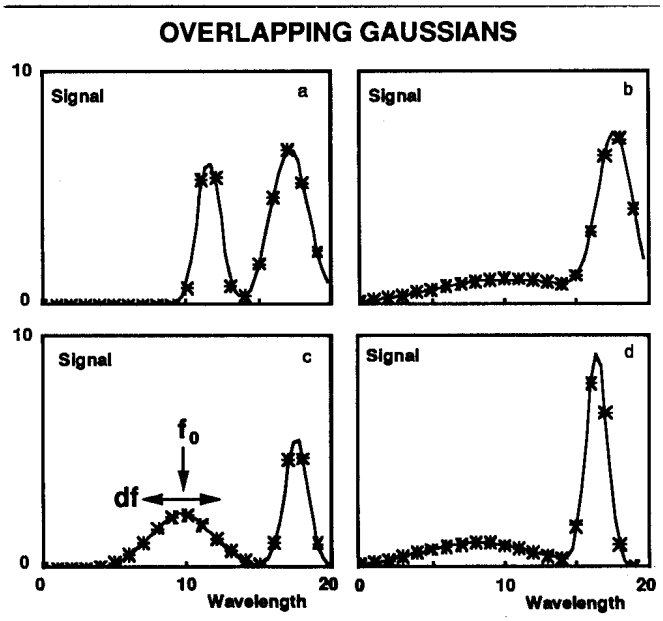

Fig. 4 The MLP fit of two overlapping Gaussian distributions to 20 spectral measurements, using a (20:10:6) MLP-1 network

might represent impurity ion temperatures and Doppler shift velocities in a Tokamak plasma. In order to estimate the six unknowns, we most commonly perform a least-squares fit, minimising the cost function

$$Q(\hat{A}_{1,2},\hat{f}_{01,2},\hat{df}_{1,2}) = \sum_i (S_i - y(f_i ; \hat{A}_{1,2}, \hat{f}_{01,2}, \hat{df}_{1,2}))^2 \qquad (3)$$

Except in simple cases when y is a linear function of the unknown variables (x_k), the minimisation of Q requires an iterative procedure which is time-consuming and not particularly suited to front-end processing

The resulting mapping, G_k, relating the data to the unknowns is clearly continuous and differentiable due to (2,3). However the iterative minimisation of the cost function does not make use of the fact that slightly different measurements are associated with only slightly different parameters. In principle, we could interpolate between neighbouring points, therefore a direct mapping G between the measured quantities $S_i(i=1,20)$ and the parameters $x_k(k=1,6)$ exists. Since there will rarely be a closed form for G, we are obliged to look for an approximation of each G_k, denoted $\hat{G}_k$, so that

$$x_k \approx \hat{x} = \hat{G}_k(S_i) \qquad (4)$$

The ability of the MLP to approximate any continuous mapping makes it an excellent candidate for providing the sought for representation of $\hat{G}_k(S_i)$. To find the specific MLP which can represent the G_k for this particular problem, we proceed as follows.

The range of x_k must first be defined, so that the required range of validity of $\hat{G}_k(S_i)$ can be known. For this example we chose the ranges:

$$A = [2,7] \text{ and } [5,10]$$
$$f_o = [8,12] \text{ and } [16,18]$$
$$df = [1,8] \text{ and } [1,2]$$
$$f_i = 1,...20$$

Examples of the $\{S_i, x_k\}$ pairs were created using (2) for 1000 values of the set of 6 parameters, randomly distributed within their given ranges. The first 500 examples of the $S_i \rightarrow x_k$ mapping were used to define an MLP-1 of architecture $(N_1:N_2:N_3)=(20:10:6)$. The MLP-1 is obtained by adjusting the weights in the two matrices so as to minimise the mean square residual between the example outputs and the MLP representation of these outputs. The methods used to perform this minimisation are usually modified gradient descent techniques. The end product of this procedure is the pair of weight matrices $\underline{W}^{12}$ and $\underline{W}^{23}$ (Fig. 3) which generates the required mapping to approximate G.

To test the applicability of the $\hat{G}_k$ mapping, we evaluated $\hat{x}_k = \hat{G}_k(S_i)$ for the second batch of 500 examples, and looked at the Root Mean Square (RMS) of the residuals $(\hat{x}_k - x_k)$ expressed as a percentage of the full scale of each variable (%FS). All variables were well represented, with an RMS residual of 1.6%FS. Figure 4 shows the measured points S_i (asterisks) and fitted curve $S_i(\hat{x}_k)$ for four very different widths and positions, illustrating the excellent quality of the $\hat{G}_k$ mapping provided by the MLP-1.

The forward pass calculation of $\hat{G}_k$ provides a set of estimates $\hat{x}_k$ which is of high quality, quick to implement and fast in execution.

V. TASK 2 - PLASMA EQUILIBRIUM

A challenging problem confronting research in Tokamaks is the precise definition of the plasma shape at any given instant. The estimation of this shape is essential for closed-loop feedback control, and must be carried out with a bandwidth of up to a few kilohertz. The result must also be precise since it ultimately limits the precision of the feedback control loop.

Figure 5 represents a simplified Tokamak. The small circles represent the positions of measurements of magnetic

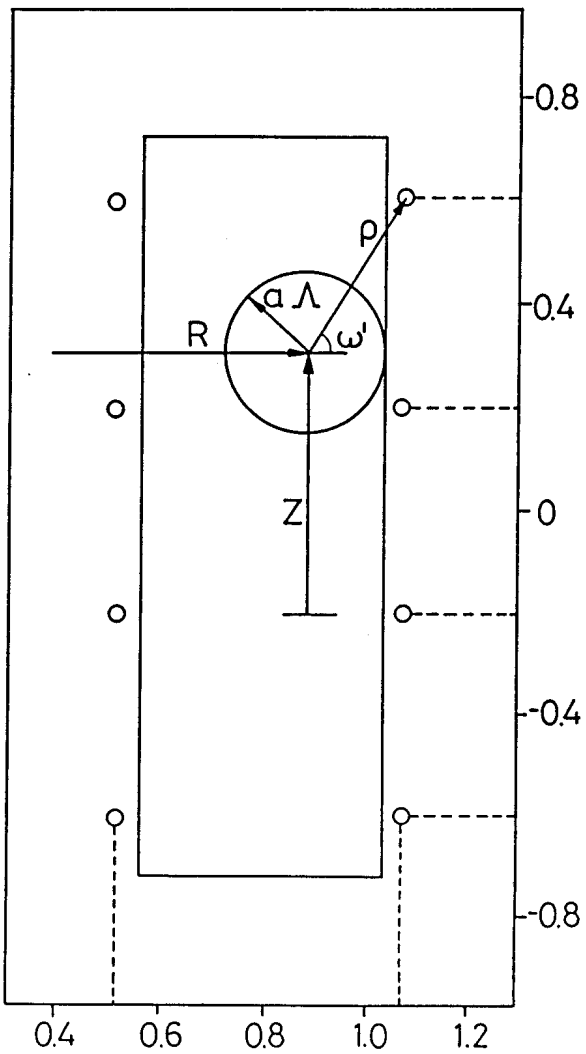

Fig. 5 Sketch of the tokamak equilibrium geometry. Variables are the major radius (R), vertical position (Z) and asymmetry factor (Λ)

flux. The large circle represents the plasma current with variable position (R,Z) and equilibrium asymmetry (Λ). The signals from the flux loops represent the experimental data (S_i) and the 3 unknowns [R,Z,Λ] represent the parameters to be estimated (x_k), continuing the notation of the previous section.

We must therefore develop a useful representation of the mapping between $S_i \rightarrow x_k$. We possess a description of the inverse mapping, which means that we can calculate the signals for a given configuration, but we cannot derive a closed form of the forward mapping. Thus we must construct an approximation $\hat{G}$, to be valid over the entire range of interest of the vector of measurements S_i.

For almost fixed circular plasmas, an approximate mapping can be obtained by linearising, and then inverting, the known physical mapping. As the shape of the plasma varies and the range of parameters increases, the linearised mapping becomes less and less reliable, especially for extreme values of plasma parameters, which are of most interest to us.

We consider the simplified test-case of a circular plasma within a 3:1 rectangular aperture as in Fig. 5. The flux distribution is known to be approximated by

$$\psi(R,Z,\Lambda,\rho,\omega') = \frac{\mu_0 I_p R}{2\pi}[\ln(8R/\rho)-2] - \frac{\mu_0 I_p}{4\pi}[\ln(\rho/a) + (\Lambda+0.5)(1-a^2/\rho^2)]\rho\cos\omega' \qquad (5)$$

where ρ, a, ω' are shown in the figure.

The values [R,Z,Λ] were varied randomly and uniformly between [0.7, 0.9], [-0.48, 0.48], [0.5, 4.0] respectively, to create a set of 1000 examples. As before, the first 500 examples were mapped by both a linear relationship obtained by Singular Value Decomposition techniques as well as by an MLP-1 with N_2=10.

To obtain a reasonable linear representation of $\psi \rightarrow$[R, Z, Λ] we need at least 15 flux-loops. On the other hand, the MLP-1 mapping provided a good fit even when the total number of flux loops was reduced to 6. An example of the quality of representation, fitted value vs. actual value, is shown in Fig. 6 with 6 input signals. The MLP-1 inherent

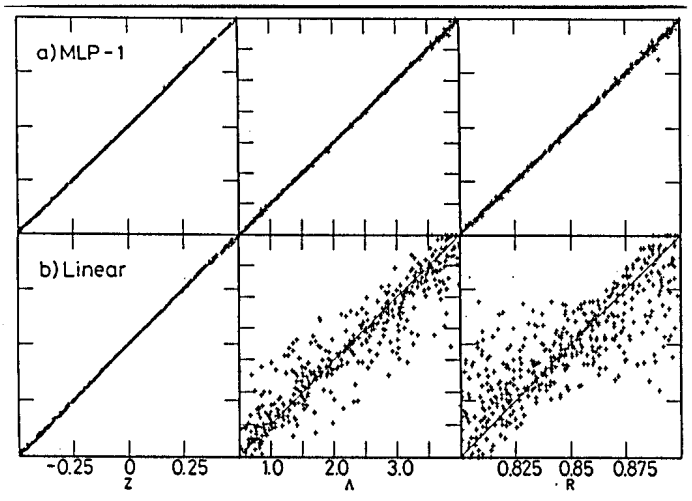

Fig. 6 Quality of the mapping for 6 flux loop differences; (a) MLP-1 mapping and (b) linear mapping

non-linearity is clearly well suited to reproduce the non-linear mapping problem represented by (5).

The range of R, [0.7 - 0.9], means that either the inner or outer wall defines the effective plasma aperture. This produces a bend in the mapping where the plasma surface touches both walls. Figure 7 shows the flux signals for 4 flux loops as the major radius R is varied for a vertically

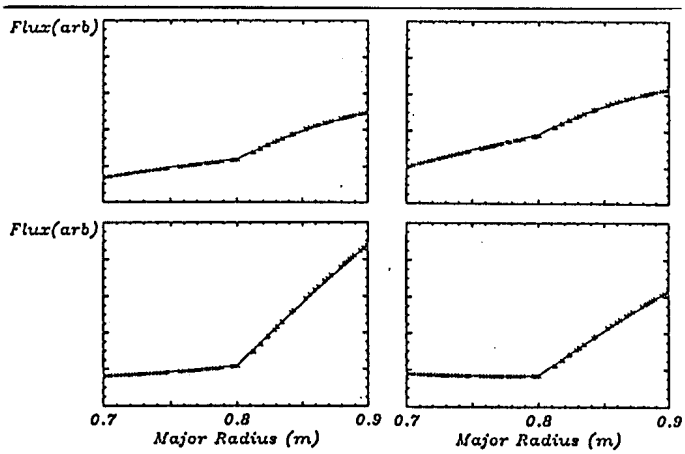

Fig. 7 Four flux signals with both inside and outside limited plasmas (Λ = 1.5, Z = 0.0, a = 0.14- 0.24)

centred and fixed Λ plasma, illustrating the structure of the mapping required. The crosses show the fitted data and the solid line shows the underlying function. This example with a gradient discontinuity was solved using a (9:10:3) MLP-1 network which gave a precision of 1.2%FS.

A more difficult problem was solved with more output variables to be estimated from more input data, using a (42:15:13) MLP-1, and training on real experimental data from the DIII-D Tokamak. The results were very encouraging [5] and provide an option for implementation on the TCV Tokamak.

VI. TASK 3 - SOFT X-RAY IMAGES

This section presents a new method of extracting crude global information from Tokamak soft X-ray imaging with a precision and speed which could render these diagnostics more useful for plasma control. The method relies on approximating a direct non-linear mapping between either tangential or poloidal soft X-ray images and the plasma parameters to be estimated.

Figure 8 illustrates two typical soft X-ray systems, using pinhole imaging, viewing either tangentially or poloidally. In both cases the images contain an enormous amount of information. The conventional approach to extracting some simple quantities such as the position of the peak, the elongation, and the vertical and horizontal half-widths of the emission profile involves a 2-step process. Since the pinhole camera provides a linear mapping between

the emission profile and the detector signals in the planes "P1", "P2", "T", the emission profile in the poloidal plane "P"

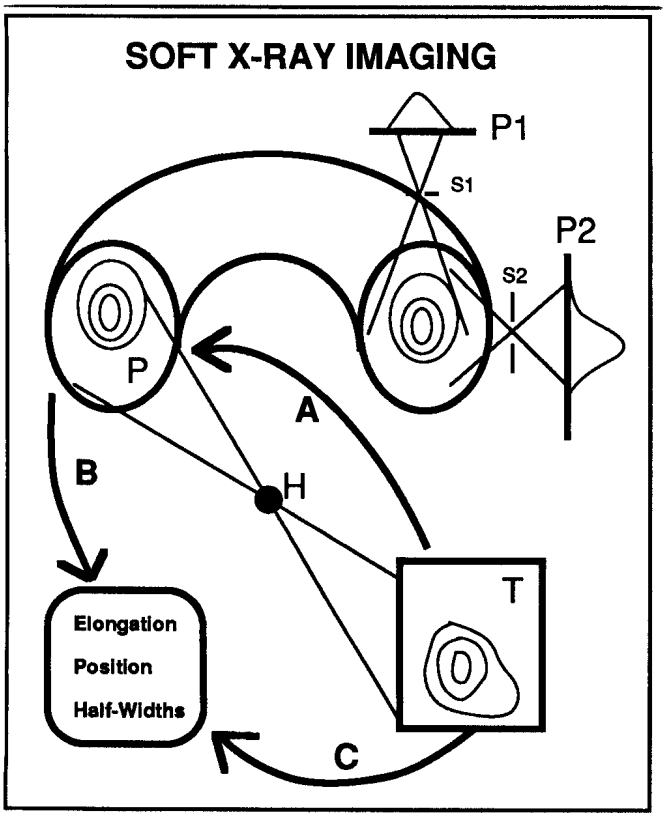

Fig. 8 A sketch of the soft X-ray imaging geometry. The 2-step analysis ("A","B") and one-step analysis using the neural network ("C") are indicated.

can be extracted by calculating and inverting this mapping, and applying it to the raw data, illustrated as the arrow "A" in Fig. 8. This operation is delicate due to the ill-conditioned nature of the pinhole projection matrix, and considerable effort has to be invested to perform this step.

In order to estimate the chosen basic parameters, we must subsequently fit some given form to the reconstructed emission profile. This is illustrated by the arrow "B" in Fig. 8. This 2-step process is unsatisfactory for our purposes for two reasons; we require a large dimension matrix multiply, and the estimation path is inevitably iterative for non-trivial parameterisation. The excessive computations required exclude real time estimates. The poloidal imaging does not require the inversion phase, but still requires the time consuming estimation phase.

We have investigated the use of an MLP-1 to provide the direct, non-linear mapping indicated as "C" in Fig. 8. We first showed that a (144:10:5) MLP-1 could accurately (~ 1%FS) map a 12x12 pixel array in the plane "T" to the amplitude, widths and positions of simple emission profiles in the plane "P". A more stringent challenge was to map a 6x6 pixel array to the central vertical elongation of varying shaped plasmas, as this information is useful for the control of the plasma, if available in real time. As previously, we generated 500 examples of varied plasma shapes, and trained a (36:10:1) MLP-1 network to map to the vertical elongation used to generate the examples. The results in Fig. 9, are encouraging, giving an error of 7%FS, corresponding to an

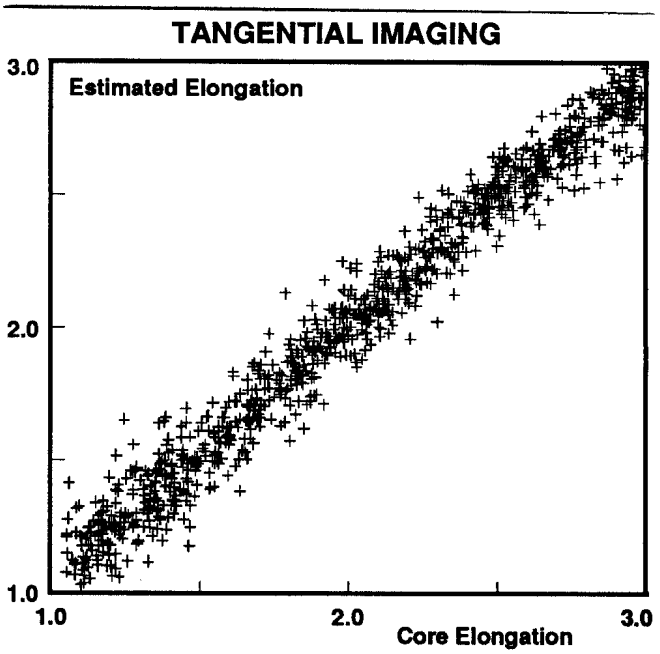

Fig. 9 Estimated values of elongation versus the true value for tangential viewing.

RMS error of 0.14 in terms of the absolute value of the plasma core elongation, which lay between 1.05 and 3.0 for the examples generated.

Tests on mapping the poloidal plane imaging through slits $S_{1,2}$ gave almost as good results as the tangential viewing, the mapping being learned from the same set of emission profile examples.

VII. TASK 4 - LANGMUIR PROBES

Tokamaks are presently equipped with many Langmuir probes. These are conducting surfaces immersed in the plasma and draw a current when electrically biased with a voltage V. The drawn current depends on the plasma density, the plasma electron temperature (T_e) and the floating potential (V_f) of the plasma, according to

$$S_i = I(V_i) = J_{sat} \times (1.0 - \exp(V_i - V_f)/T_e) \qquad (6)$$

where J_{sat} is itself a function of density. Figure 10 shows four such characteristic curves for different combinations of $[J_{sat}, T_e, V_f]$. Estimating the parameters from the data represents a similar problem of finding a mapping $S_i = I(V_i) \rightarrow [J_{sat}, T_e, V_f]$ where the inverse mapping (6) is explicit and the direct mapping is not.

We proceeded as before, generating 1000 examples of $\{S_i, x_k\}$, in the ranges J_{sat} = [0.33, 1.0], T_e = [33, 100], V_f = [$-2T_e$-20, -20]. A (30:5:3) MLP-1 was fitted with 500 examples, and the residuals were calculated for the remaining 500 examples obtaining 3.2%FS. Figure 10 shows four characteristic curves with the data (asterisks) and fitted

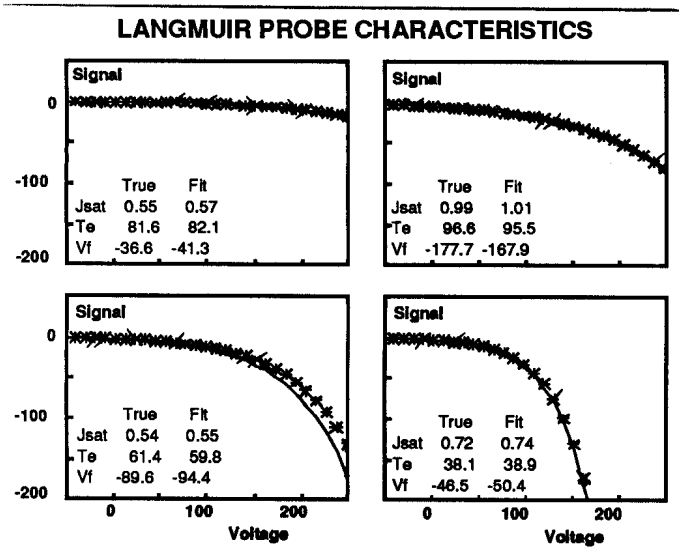

Fig. 10 The MLP fit of the Langmuir probe characteristics to 30 points in a voltage sweep, using a (30:5:3) MLP-1 network.

curves (line). Three are perfectly fitted, whereas the fourth is systematically wrong. The floating potential (V_f = −90 V) is at the edge of its range, where the quality of the mapping was systematically found to be poorer. Training an MLP to learn a given input-output mapping is prone to the same problems as any functional fitting; the quality of fit near the edge of the data volume is generally inferior, and extrapolation is generally dubious, in spite of some extravagant claims to the contrary. This is even particularly so for MLPs due to the rich surface structures they can generate.

In practice, multiple Langmuir probes are repetitively swept to measure the characteristic curve (6). The large quantity of data generated and the significant CPU time of the non-linear fitting analysis makes the MLP-1 an attractive solution to this problem, implementable in the front-end electronics. The accuracy of the mapping is at least as good as the physical reality of the underlying expression (6).

The four example tasks discussed possess similar properties. Input data are reduced to some output data, non-linearly but continuously related to the values of the inputs. The result is not necessarily exact, but can have an acceptable and useful precision. The simplicity of the calculations makes the data analysis extremely fast and suited to front-end processing. The analysed data can be used locally, or passed up to a higher level of processing, but with a much reduced volume.

VIII. DYNAMIC CONTROL

The vast majority of current applications of feedback control assume that the system to be controlled can be linearised, at least over a finite region of operating space. This assumption is often valid when the system is non-linear; a good example is the balancing of an inverted pendulum, for which a conventional proportional-differential (PD) control of the locally linearised system, Fig. 11(a), gives excellent results.

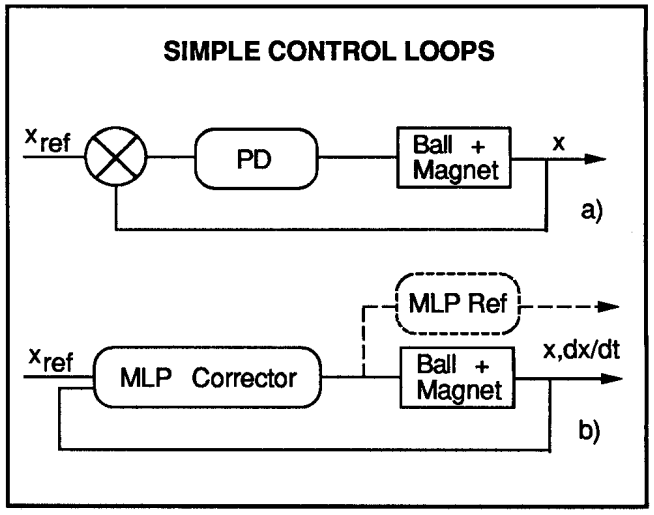

Fig. 11 Schematic of (a) a linear PD controller and (b) an MLP-1 driven non-linear controller showing the MLP reference model.

Other systems are more complicated, and are more fundamentally non-linear. This may be due to the underlying system, to an extended range of operation, or to complicated behaviour such as backlash, hysteresis, viscosity, all of which may evolve with time. The first approach to such problems is to maintain the linearised formalism but to adapt the feedback control, by automatic optimisation or by simply programming the feedback control parameters (gain scheduling).

An alternative approach to this type of problem is to leave the linearised world; the cost is high, however, as the powerful mathematical framework developed must also be abandoned. The generality of the MLP surface representation has aroused considerable interest as a candidate for a non-linear controller, Fig. 11(b). A number of groups are active in this field, and the work of one serves as an excellent example [6].

The system considered here is the levitation of a magnetisable ball using an electromagnet, Fig. 12. The non-linearity arises from the fact that the attractive force varies

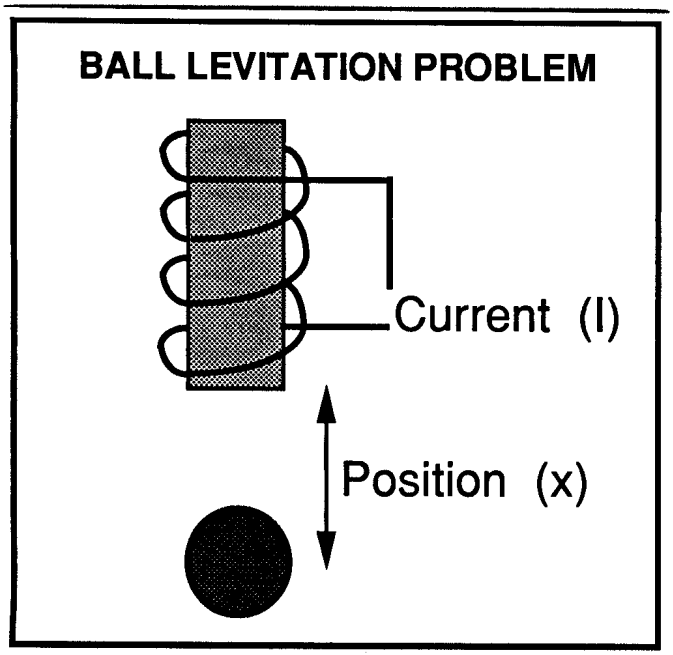

Fig. 12 The ball levitation problem.

as $F = k(I/x)^2$ where I is the magnetising current and x is the magnet-ball separation. The $1/x^2$ dependence gives rise to severe problems as x becomes small, and the I^2 dependence means that we cannot repel the ball with a negative current. The same problem would arise with a high thermal inertia DC current heater. The response of the system to the reference signal shown in Fig. 13 using a PD controller was difficult to optimise, and gave an inadequate result.

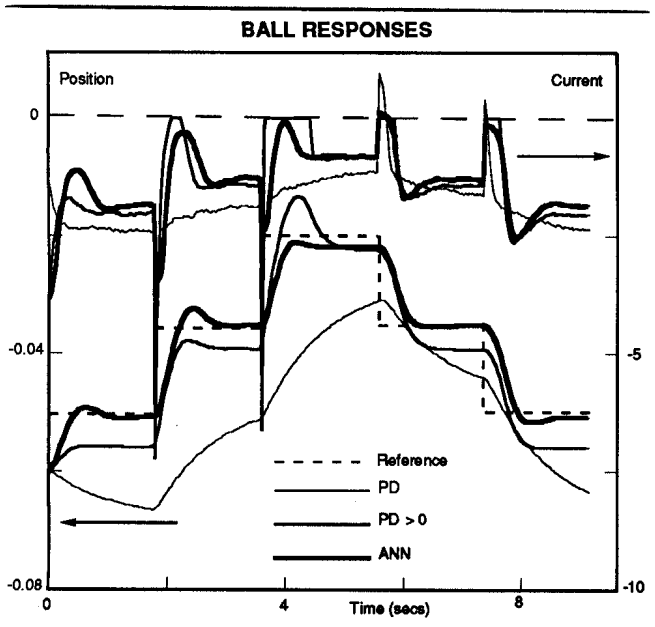

Fig. 13. Results of 3 controllers used to levitate the ball.

Rendering the controller non-linear by the intuitive modification, I = Max (I_{PD}, 0), improves the system response, Fig. 13, but this solution has already required a small input of understanding and intuition.

Using an MLP-1 in the feedback loop fed with the reference position, the actual position, the actual speed of the ball and an offset, Fig 11(b), allows us to obtain an improved control compared with that obtained with the PD controller and the modified PD controller. In Fig. 12 the lower group of curves shows the reference signal (dotted) and the system response with the 3 controllers. The upper group of curves shows the magnet currents. Using a simple (4:2:1) MLP-1 has produced a faster and less structured step response. It has most importantly acquired the property of not changing the sign of the current in the magnet, like the modified PD controller.

The optimisation of such a controller for an unknown process is not straightforward, and there have been many approaches proposed. One which appears encouraging is to use two neural networks, one to act as the non-linear controller, and one to act as the non-linear model reference, by analogy with the Model Reference Adaptive Systems (MRAS) of linearised systems (dotted system in Fig. 11(b)). The MLP emulating the non-linear system can be taught by example, identifying the system, and the controller MLP can be optimised on the predictive use of this reference model. Such studies are underway in several groups.

Although the levitated ball problem is not directly related to the main topics of this conference, the generality of this approach to non-linear control may find applications in our fields, where we have badly modelled, badly understood or simply non-linear systems to control. Neural controllers may well be more robust than conventional controllers and easier to adapt. The few particular studies in the literature do not yet allow us confirm this hope.

IX. IMPLEMENTATION

There have been many implementations of neural networks. Most of them only perform the forward pass calculation, the learning hardware being limited to parallel accelerators. The interconnected nature of the MLP makes it suitable for parallel digital processing using a small number of high speed processors. The interconnectivity is a major difficulty for all cellular implementations. Analogue implementations provide an attractive high speed option, but high density and high precision are still incompatible; our own implementation of a 3000 weight analogue system for task number 2 [7] represents several crates of electronics. The implementation of binary networks is simpler, but again there is no emerging best solution; optical devices are among the candidate solutions for such networks.

X. DIFFICULTIES

We have seen several data analysis tasks which can be successfully solved using the MLP-1 neural network. We can identify three difficulties when solving a new problem:

- choosing the number of hidden neurons (N_2 of Fig. 3)
- adjusting the MLP weights to fit the given examples
- judging whether the solution is acceptable.

The first difficulty is not particularly significant. Any fitted function (polynomial series, Fourier series) requires us to "guesstimate" the required number of free weights. The choice is somewhat subjective if exhaustive testing is to be avoided (due to the second difficulty). Finding a good architecture for these problems and others has required a little trial and effort and a little feeling.

The third difficulty is somewhat related. Unless we embark on exhaustive testing, how do we judge our solution. The use of a large example set for fitting as well as an unseen test example set is essential to avoid overfitting due to the specific ability of the MLP to generate highly structured mappings. We need to ensure that the number of free weights in the MLP is constrained by an adequate number of examples. We have avoided fitting fewer than double the number of example outputs as there are free weights.

The second and major difficulty is in finding the weights such that the MLP-1 furnishes a reasonably good $\hat{G}_k$. The term reasonably good is important, as there is no way, at present, of determining whether a given $\hat{G}_k$ is the best possible for the MLP-1 structure chosen. That is to say whether the fit is converging on a global minimum or lies within a given distance of a minimum. We must therefore judge a given $\hat{G}_k$ on its known merits rather than in absolute terms. The second part of this learning difficulty is in finding a given $\hat{G}_k$ within a reasonable time. The conventional gradient descent algorithm, known as back-propagation when applied to the MLP [1], is notoriously slow. A recent appraisal of this problem in terms of convergence eigenmodes has shown that these can be arbitrarily slow, even with optimally tuned gradient descent step-lengths. For this reason most users develop their own recipes for speeding up learning [7,8,9]. These recipes are difficult to compare, and are unanalysable. Most authors, however, end up with a satisfactory learning rate. The future will bring a new generation of parallel calculators for convergence, and hopefully this bottleneck will become less significant.

XI. CONCLUSIONS

The field of neural networks has very ambitious aims for large-scale problems of perception, signal processing and control. Such research projects are the domain of highly specialised groups. In this paper we have seen that even in a specific field, Tokamak research, there are interesting and useful applications already, using available techniques for creating and implementing the required networks. The continuous nature of the MLP mappings even makes them suitable for analogue techniques. Since many of these simple problems already require substantial front-end processing, the use of a simple estimation algorithm, even if slightly less accurate, can be of great interest.

If a particular front-end processing, or analysis task has some of the following properties:

- no direct mapping is explicitly known
- the direct mapping is delicate due to noise sensitivity
- the required mapping is functionally continuous
- the known direct mapping is computationally intensive
- the required mapping must be regularly adapted

then an MLP could be a candidate solution. The aim of this paper is to encourage other researchers with this type of data processing problem to test these powerful techniques.

ACKNOWLEDGEMENTS

It is a pleasure to acknowledge help from and informative and stimulating discussions with O. Bye, D. Diez. The work was partly funded by the Fonds national suisse de la recherche scientifique.

REFERENCES

[1] D.E. Rumelhart and J.L. McClelland, "Parallel Distributed Processing: Explorations of the Microstructures of Cognition", Vols I,II,III, MIT Press (1986)
[2] T. Kohonen, "Self-Organisation and Associative Memory", 3rd. Ed., Springer, New York, (1989)
[3] T. Poggio and F. Girosi, "Networks for Approximation and Learning", Proc. IEEE 78 (9) 1481 (1990)
[4] R. Hecht-Nielson, "Kolmorogov's Mapping Neural Network Existence Theorem", Proc. INNS Annual Meeting, San Diego, U.S.A., IEEE, Vol III 11 (1988)
[5] J.B. Lister and H. Schnurrenberger, "Fast Non-linear Extraction of Plasma Equilibrium Parameters Using a Neural Network Mapping", Lausanne Report LRP 397/90, to be published in Nuclear Fusion (1990)
[6] H. Bleuler, D. Diez, G. Lauber, U. Meyer and D. Zlatnik, "Nonlinear Neural Network Control with Application Example", Proc. INCC 1990 (Paris), IEEE 201 (1990)
[7] J.B.Lister, H. Schnurrenberger and Ph. Marmillod "Implementation of a Multi-Layer Perceptron for a Non-linear Control Problem", Lausanne Report LRP 398/90 (1990)
[8] T.P. Vogl, J.K. Mangis, A.K. Rigler, W.T. Zink and D.L. Alkon, "Accelerating the Convergence of the Back-Propagation Method", Biological Cybernetics 59 257 (1988)
[9] R. Battiti and F. Masulli, "The Bold Driver Method", Proc. INCC 1990 (Paris), IEEE 757 (1990)