Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Applied Soft Computing 2 (2003) 283–296

Applied Soft Computing

www.elsevier.com/locate/asoc

Improving neural network models of physical systems through dimensional analysis

David J. Gunaratnam*, Taivas Degroff, John S. Gero

School of Architecture, Design Science and Planning, University of Sydney, Sydney 2006, NSW, Australia

Received 12 April 2002; received in revised form 30 October 2002; accepted 31 October 2002

Abstract

The paper presents a technique for generating concise neural network models of physical systems. The neural network models are generated through a two-stage process. The first stage uses information embedded in the dimensions or units in which the data is represented. Dimensional analysis techniques are used initially to make this information explicit, and a limited search in the neural network architecture space is then conducted to determine dimensionless representations of variables/parameters that perform well for a given model complexity. The second stage uses information available in the numerical values of the data to search for high-level dimensionless variables/parameters, generated from simple combinations of dimensionless quantities generated in the first stage and which result in concise neural network models with improved performance characteristics. The search for these high-level dimensionless variables/parameters is conducted in an enhanced representation space using functional link networks with flat or near flat architectures. The use and effectiveness of the technique is demonstrated for three applications. The first is the design and analysis of reinforced concrete beams, which is representative of the class of problems associated with the design and analysis of composites. The second is the classical elastica problem, for predicting non-linear post-buckled behaviour of columns and the third, the analysis of a bent bar under a specified combination of loads.

Keywords: Dimensional analysis; Neural networks; Physical systems; Functional link; Concise models

1. Introduction

The need to develop a model for predicting behaviour of a system arises in a number of disciplines. In engineering, models of physical systems are required for solving prediction, control, diagnosis and design problems. The solutions to modelling problems require the determination of mapping functions between a set of input and output variables, and in physical systems the mapping functions are usually between the stimulus applied and the response of the system.

Conventional programming paradigms have been successfully used to generate these mapping functions, when domain knowledge is explicitly available to describe the models uniquely. Where the information available for constructing the model is only available in the form of data derived from observations or measurements, neural network techniques, based on the supervised learning paradigm, have been successfully used to generate the mapping function from the data. If the information available in the data is not adequate, search may have to be conducted in both the data representation and network architecture spaces to arrive

* Corresponding author. Tel.: +61-2-9351-5601; fax: +61-2-9351-3031.
E-mail address: davidg@arch.usyd.edu.au (D.J. Gunaratnam).

1568-4946/02/$ – see front matter
PII: S1568-4946(02)00061-3

at a mapping function that performs reasonably well on generalisation tasks.

In contrast, the design problems are inverse problems with one-to-many mapping, where a set of requirements can map on to a number of system descriptions. Thus, the solution to the design problem involves search in the design variables space to arrive at a system description that optimises a pre-defined objective function. The search for the optimum description can be conducted either by solving a series of direct problems—as in approaches based on mathematical programming techniques—or by using previous design and problem solving experiences—as in approaches based on AI and machine learning techniques.

In the latter approach, the previous design and problem solving experiences are used to prune the search space, to guide the search and to carry out the search. It is also possible to hierarchically decompose the design space into sub-spaces [1] and to conduct the search in these sub-spaces. Design relationships provide a basis for conducting the search in these design sub-spaces and for making design decisions. Thus representing and generating these relationships so that previous design and problem solving experiences can be encoded in a concise form, as high-level design relationships, without loss of information and generality, is of considerable interest [2–4]. Neural networks have also been used to learn design relationships from previous designs by generating the appropriate mapping functions.

In the present work, representation of data, in the form of high-level variables and parameters, is considered central to developing both concise neural network models with improved performance as well as high-level design relationships. This paper, however, focuses mainly on generating concise models, and high-level design relationships are considered only briefly within Section 6. A complete treatment of high-level design relationships requires resolving issues arising from the non-uniqueness of the mapping to be generated, and will be dealt with in a subsequent paper.

Domain independent techniques available for constructing neural network models of physical systems use only information available in the numerical values of the data. Improved performance of these models is usually achieved by combining dimensionality reduction with search in the network architecture space [5]. Dimensionality reduction is achieved by either discarding less significant variables or by combining variables linearly or non-linearly. There is usually some loss of information associated with these methods of dimensionality reduction, and there is no guarantee that the models generated would satisfy the principle of dimensional homogeneity.

In the present work, concise neural network models are generated through a two-stage process that combines information available in the data with search, mainly in the representation space. Information available in the dimensions or units in which the data is represented is made explicit in the first stage, through dimensional analysis, and is used to reduce dimensionality by combining variables and parameters to form a set of dimensionless products. The set of dimensionless products is not unique, and hence a limited search is carried out in the network architecture space to determine the set that leads to the best mapping function.

The set of dimensionless products selected at the end of the first stage is then combined further in the second stage, using information available in the numerical values of the data, to form high-level dimensionless variables/parameters for the system. The search for these high-level dimensionless variables/parameters is conducted in a series of enhanced representation spaces using functional link networks with flat or near flat architectures. The high-level variables/parameters are finally used to generate concise neural network models of the system.

In the sections that follow, we initially compare the domain independent methods for generating concise neural network models with the two-stage process used in the present work. Then the first stage of the process, of forming a set of dimensionless products by unlocking information embedded in the dimensions or units used in the representation of data, is considered. This is followed by the second stage of the process, where the bases for combining dimensionless products further to form high-level dimensionless variables/parameters, enhancing the representation space and then pruning the representation space based on model performance, are discussed. Finally, Section 6 demonstrates the effectiveness of the proposed two-stage process.

2. Improving performance of neural network models

Considerable work has been done in developing domain independent techniques for improving performance of neural network models [5]. These techniques are based on changing the representation of the data by some form of pre-processing transformation and/or searching the network architecture space, and use only the information available in the numerical values of the data.

Dimensionality reduction is usually the goal for determining data representations that lead to improved performance, and is achieved either by discarding less significant variables—feature selection—or by combining the variables to form a reduced set of input variables—feature extraction. The improvement in performance results from the less complex neural network model generated, with fewer adaptive parameters to be determined and hence less time for training [5]. Domain independent techniques for dimensionality reduction can either be based on information available in the input space only or use the mapping information available in the data. In the former case, dimensionality reduction is achieved by combining variables linearly using principal component analysis (PCA) networks or non-linearly using auto-associative networks. In the latter case, feature selection algorithms, based on techniques such as stepwise forward or backward feature selection and genetic algorithms, are used to identify the less significant variables based on a combined measure of performance that includes both the network prediction error and dimensionality of the input space [6].

There is usually some loss of information associated with the above approaches to dimensionality reduction. Further, the methods for combining variables are based only on information available in the input variable space. It is usually difficult to ascribe any meaningful physical interpretation to the combined variables, and there are no means of ensuring that the mapping functions generated satisfy the principle of dimensional homogeneity, or that the neural network models generated have the correct generalisation capability [7].

In contrast, domain dependent techniques based on prior or domain knowledge have been used to reduce the dimensionality and generate neural network models without loss of information and with enhanced generalisation capabilities [7–9]. Dimensional analysis [10,11] is used to unlock the information embedded in the dimensions in which the data are represented, and this additional information is made available to the network during training through dimensionless representations of data. Thus ensuring that the network models generated satisfy the principle of dimensional homogeneity, and are invariant with respect to the consistent set of units used in the measurement of data.

In the present work, further reduction in model dimensionality is achieved by generating data representations that are simple combinations of the dimensionless products arrived at by dimensional analysis—the high-level dimensionless variables/parameters. These high-level variables/parameters arise as a consequence of the laws and principles that constrain the behaviour of the system, and examples are given in a later section to illustrate the existence of these variables/parameters, and the degree to which further dimensionality reduction is possible. Since the high-level variables are simple combinations of the dimensionless products, it is possible to give physically meaningful interpretations to them. The second stage dimensionality reduction can also be achieved by domain independent techniques identified previously. The performances of the neural network models based on these alternative approaches, that satisfy the principle of dimensional homogeneity, are compared in Section 6.

The performance of the neural network model is also influenced by the parameters that define the neural network architecture. The search in the network architecture space can be conducted at the three levels of connection weights, network structure and activation function, in order to determine the best neural network model. In most applications, dealing with modelling of physical systems, the search is limited to the first two levels of connection weights and network structure, with the activation function being pre-defined and fixed. Genetic algorithm and other evolutionary computing techniques have been used to carry out the search at all three levels [12,13]. In the present work, however, the focus is on generating concise neural network models, and thus the search in the network architecture space is limited to less complex architectures.

3. Dimensionless representations

Dimensional analysis provides three constraints that the neural network models must satisfy, and these arise from the principle of dimensional homogeneity, product theorem and Buckingham's Pi theorem [10,11,14]. The Buckingham's Pi theorem provides the basis for generating a set of dimensionless products that satisfy these constraints, which in one form can be stated as follows: If a physical situation is characterised by n variables and r basic dimensions occur in the dimensional representations of these variables, then there are $n-r$ independent dimensionless products that are sufficient to describe the situation.

The dimensionless products, referred to as the π terms, can be arrived at by using a number of different methods, depending on the type of information available [11]. In the present work, the products are formed by partitioning the variable set into basis (r) and performance variables ($n-r$), and then forming products, in turn, of each of the performance variables with the basis variables. The exponents for the basis variables are selected to make these products dimensionless.

The above procedure is implemented in matrix form by first setting up a dimensional matrix, having n rows for the variables (the first r rows for the basis variables) and k columns for the representation of the dimensional exponents of the variables in the basic dimensions [7]. This matrix is manipulated using rank preserving operations to produce a diagonal form for the basis variable rows of the matrix. The required exponents for the dimensionless products are then determined from the elements in the rest of the rows of the modified matrix. Details of these operations are given in a later section of this paper.

Depending on the variables selected for the basis, different sets of π terms will be generated by this process. There are ${}^{n}\mathrm{C}_{r}$ possible sets of π terms, but they are not independent. It is possible to transform one set to another using simple operations. The basis variables are selected through a search process, and the search space is limited by using the basis selection heuristics suggested by Bhaskar and Nigam [14]. Performance of networks on the reduced π term sets is used to determine the final set of dimensionless variables to use. The training set size can be progressively reduced, while increasing the validation and test set, to discriminate between competing sets of π terms and determine the π term set—and hence the representation—that performs best.

The π terms selected in the first stage described above are then combined further to form higher order terms and achieve additional reduction in dimensionality. As the main interest is in developing simple models that best describe the data using variables that are amenable to physical interpretation, the search for this stage is limited to simple combinations of π terms, such as products, with exponents of individual terms limited to integers. Here too, the search space is pruned using domain knowledge or heuristics, where available.

In this paper variables are used for quantities, such as response of a system, which usually change within a given problem. Parameters are used for quantities required to describe the system model, and which are fixed within a problem, but allowed to change within a problem class. High-level variables and parameters are used to identify higher order π terms, which result in dimensionality reduction and are adequate to characterise the system. These high-level variables and parameters can thus be considered to be model features that provide the basis for predicting system responses or characteristics.

4. High-level variables and parameters

The high-level variables and parameters that result in dimensionality reduction arise as a consequence of the laws and principles that constrain the system being modelled. These can be determined either from the governing equations (and closed form solutions), if available, or from the numerical values of the data, and are either in the form of products of π terms raised to different exponents or the sums of such product terms. That these are the only forms possible can be established using the product theorem and homogeneity conditions. In this section, we provide two examples to illustrate how these high-level variables and parameters are formed and lead to, in some cases, a substantial reduction in the dimensionality of the model.

In the first example, we consider two modelling problems that are characterised by the same set of variables and parameters. The first modelling problem is to predict the linear response of a cantilever under a concentrated load at the free end. This is characterised

Table 1
Dimensional matrix for the cantilever and column buckling problems

Variables/parameters	[F]	[L]
E	1	−2
l	0	−1
P	1	2
δ	0	1
I	0	4

by the variables P and δ—load and deflection, respectively, at the free end—and the parameters l, E and I—length, Young's modulus and second moment of area of section, respectively, for the cantilever. The second is the well-known non-linear elastica problem, for predicting the lateral deflection δ of an elastic column subject to an axial load P.

Since both problems are characterised by the same set of variables and parameters, stage 1 of the dimensionality reduction process essentially leads to the same set of π terms. The five variables and parameters can be expressed in terms of two dimensions (F and L), hence only two of these can be in the basis. Selecting E and l to be in the basis, the resulting dimensional and modified dimensional matrices are as shown in Tables 1 and 2, respectively.

The modified dimensional matrix is obtained by transforming the dimensional matrix using rank preserving operations to produce the diagonal form for the basis parameters.

For both problems, π_2 represents the dimensionless response variable, π_1 the stimulus applied to the system and π_3 the dimensionless system parameter (Table 3). Hence, to model these two problems to predict response, the functional relationship should be of the form:

$$\pi_2 = F(\pi_1, \pi_3)$$

Table 2
Basis variables and matrix with upper diagonal form

Variables/parameters	[F]	[L]
E	1	0
l	0	1
P	2	2
δ	0	1
I	0	4

Table 3
Dimensionless π terms for cantilever and column buckling problems

π_1	P/El^2
π_2	δ/l
π_3	I/l^4

For each of the problems, it can be shown that the two π terms combine to form the high-level dimensionless variable $\pi_4 = \pi_1/\pi_3$, and the functional relationship representing the model is then given by

$$\pi_2 = F(\pi_4)$$

The existence of π_4 can be established for the two problems using moment curvature relationships, which provide the bases for deriving the governing equations. Additional domain information is, hence, required to arrive at the high-level dimensionless variables, and in the present work the information is derived from the numerical values of the data.

Closed form solutions are available for both problems, and when expressed in terms of π_2 and π_4, they take the following forms.

Cantilever problem:

$$\pi_2 = \tfrac{1}{3}\pi_4$$

Column buckling problem:

$$(\pi_4)^{1/2} = K(p)$$

where $p = \pi_2(\pi_4)^{1/2}/2$, and $K(p)$ is the complete elliptic integral of the first kind [15] and is given by

$$K(p) = \int_0^{\pi/2} \frac{\mathrm{d}\phi}{\sqrt{1 - p^2 \sin^2 \phi}}$$

Through the two-stage process, the dimensionality of the two problems has been reduced from 5 to 2. A concise neural network model for predicting response can thus be constructed with the dimensionless high-level variable π_4 as the input and the dimensionless response variable π_2 as the output.

The two problems discussed above result in quite different functions for the behavioural models—linear function for the cantilever and a complex non-linear function for the column. Despite this difference, the high-level variable π_4 is the same for both problems. Thus, in general, less information is required for establishing the existence of high-level variables, such

as π_4, than for learning the model function. Just two, properly selected, data points are adequate to establish the existence of terms such as π_1/π_3 or $\pi_1\pi_3$. In the present work, however, we assume that a set of data values exist and consider ways of establishing the existence of such terms based on information available in the numerical values of the given data.

For the column-buckling problem, the high-level variable π_4 has physical significance. The load P corresponding to $\pi_4 = \pi^2/4$ represents the buckling load, and hence defines the boundary between pre- and post-buckled states. The high-level variables thus not only lead to dimensionality reduction but also, in some situations, provide additional information for partitioning the input space into regions based on behaviour. This information can also be exploited in the training of the final network.

The second example selected is the transverse vibration of polar orthotropic circular plates with edges elastically restrained against rotation [4]. This example demonstrates that even though the π terms derived from dimensional analysis can be complex products of the basic variables that characterise the system, the high-level dimensionless variables can be remarkably simple products and sum of these π terms.

The variable and parameters that characterise the vibration problem are given by the plate radius and thickness (a and h, respectively), flexural rigidity in the radial and circumferential directions (D_r and D_θ, respectively), rotational stiffness of spring (k_{sp}), frequency (ω), density (ρ) and Poisson ratio ($\gamma_{\theta r}$). The eight parameters can be expressed in terms of three dimensions (F, L and T) and hence the problem can be formulated in terms of five dimensionless parameters. Selecting D_r, a and ω to be in the basis and following the procedure described above, the five dimensionless terms are:

$$\pi_1 = \frac{D_r}{D_\theta}, \quad \pi_2 = k_{sp}\frac{a}{D_r}, \quad \pi_3 = \frac{\rho a^5 \omega^2}{D_r},$$
$$\pi_4 = \frac{h}{a}, \quad \pi_5 = \gamma_{\theta r}$$

From the energy functional for the plate [4] it can be seen that these five π terms combine to form the following high-level dimensionless parameters:

$$\pi_1, \quad \pi_6 = \pi_3\pi_4, \quad \pi_7 = \pi_2 + \pi_5$$

The dimensionality of the problem can thus be reduced from 8 to 3, and the model for predicting the natural frequency of the plate can be expressed as:

$$\pi_6 = F(\pi_1, \pi_7)$$

The above examples demonstrate that by identifying the high-level variables and parameters for a system, it is possible to further reduce the dimensionality for the modelling problem and arrive at a simpler model for the system. These variables and parameters are formed mostly as products of π terms raised to different exponents (-1, 0, 1) and as sum of two π terms. The exponents can, however, assume other values in general. Identifying the high-level variables and parameters thus reduces, in most cases, to determining the reduction in dimensionality of the problem and the values of the exponents to achieve each stage of the reduction. This is achieved through a search process described in Section 5.

5. Generating and searching the enhanced representation space

The generation of the high-level variables and parameters from the original set of π terms can be posed as a search problem. The search is mainly for a set of values for the exponents, and is constrained by information available in the numerical values of the data and domain knowledge, where this is available. The search of the representation space, for the high-level variables and parameters, can be conducted using a genetic algorithm with a neural network providing the values for the evaluation function [16]. Genetic programming can also be used to generate both the variables and functions from data [17].

In the present work, an approach based on functional link networks [18] is used to carry out the search of the representation space. In the functional link networks the representation is enhanced, in a linearly independent manner, so that the mapping function can be learned more readily. Different functions can be used in the functional link to transform the original input pattern vector to arrive at the enhanced representations of the patterns. By selecting an appropriate functional-expansion model, it is possible to learn the mapping function with flat neural network architectures and improved learning rates [18].

Enhancing the representation in functional link networks can be considered to incorporate part of the non-linearity in the original mapping function into the enhanced input pattern. This reduces the non-linearity of the mapping function to be learned, and accounts for the resulting flat networks with less complex network architecture. In some problems it is possible to incorporate all of the non-linearity into the enhanced input pattern, and in this case a linear network would be adequate to learn the transformed mapping function.

There are thus two complementary decisions to be made in the design of functional link networks—the level of enhancements to be applied to the representation and the complexity of the network architecture. The original representation of the input pattern represents one end of the spectrum of the network design space, with all non-linearity incorporated into the network architecture. As the level of enhancement of the representation is progressively increased, the required mapping function and the architecture of the network become less complex. Hence at the other end of the spectrum, flat networks would be adequate. In all cases, however, the functional link networks result in increased dimensionality of the input space.

Our goal, however, is to generate high-level variables that would result in dimensionality reduction as well as leading to less complex networks. In order to achieve this, a functional-expansion model is selected that would include the high-level variables, in the form of products of π terms raised to different exponents, within the enhanced representation space. Rather than generate all possible high-level variables and deal with a very large representation space, we initially consider expansions that result in products and ratios of two π terms at a time. For a given set of dimensionless terms (π_i), the variables for the first functional-expansion model can be described as:

$$\left\{\pi_i, \quad \pi_i\pi_j\,(j > i), \quad \frac{\pi_i}{\pi_j}\,(i \neq j)\right\}$$

This enlarged representation space is then pruned using sensitivity information derived from networks with flat and near flat architectures, to arrive at a set of high-level variables with the desired reduction in dimensionality. This enhancement process can then be repeated to explore further possibilities for dimensionality reduction. If this process does not lead to the desired reduction in dimensionality, then the second expansion model, which introduces additional higher order terms and described below, can be applied:

$$\left\{\pi_i, \quad {\pi_i}^2, \quad \frac{{\pi_i}^2}{\pi_j}\,(i \neq j), \quad \frac{\pi_i}{{\pi_j}^2}\,(i \neq j)\right\}$$

Thus the high-level variables are generated through search, both in the representation space and the network architecture space. The search space at any stage, however, is limited by controlling the level of enhancements to the representation and the level of complexity of the network. Performance of the network is used to identify and terminate search along unproductive paths. Features used for successful termination of search include, reduction in dimensionality, all the original π terms are represented in the final set of high-level variables and improved generalisation capability of network.

6. Applications

Three applications are presented in this section to demonstrate the effectiveness of the proposed search method for determining concise neural network models with improved performance. The first application considers the analysis and design of reinforced concrete beams, and the method is used to develop both the analysis model and design relationship. In the analysis model, the non-linearity for the required function can be fully incorporated into the high-level variables and the function to be learned is then linear. Both the concrete design problem and the elastica problem, which forms the second application, have non-linearity incorporated into both the high-level variables and the function to be learned. The former includes a square root function and the latter a function that includes the complete elliptic integral of the first kind. The third application models the response of a bent beam subject to combined loads and provides another example where the non-linearity can be fully incorporated into the high-level variables. The search for the high-level variables, however, involves both the representation and network architecture spaces.

The performances of the concise network models constructed from high-level variables and parameters are compared to network models constructed from dimensionless variables and parameters at the end of

stage 1 of dimensionality reduction. The performance of concise model generated, for the concrete beam analysis problem, by the present approach is also compared to models generated by domain independent techniques such as principal component analysis and auto-associative networks, to demonstrate the effectiveness of the present approach.

6.1. Concrete beam analysis and design problems

The variables that characterise both problems are M_u (moment capacity), A_s (area of reinforcement), b (breadth), d (depth), f_c (concrete strength) and f_y (steel yield stress). In the analysis problem, the section characteristics of the beam are given and the moment capacity of the beam is required. Thus M_u would form the dependent variable and the other five variables, that define the section characteristics, would form the independent variables.

In the design problem, the section characteristics need to be determined to carry the design moment. The design problem, as posed, has no unique solution and a number of beam sections could be determined that have the required moment capacity. One way of enforcing uniqueness is to treat A_s, as the only variable and the others as parameters, which are pre-defined. The other is to pose the design problem as an optimisation problem. For the purposes of learning high-level design relationship, A_s is treated as the dependent variable in the present work.

There are only two dimensions—force and length—required to represent the six variables that characterise the analysis problem. There would thus be two variables in the basis, and hence ${}^6C_2$ possible ways of selecting the basis variables. This is reduced to four using the following basis selection heuristics: dimensionally rich and varied variables only to enter the basis; variables of interest to be excluded from the basis; and π terms to have the variables of interest in the numerator with integer exponents for all variables. The four basis variables sets are: f_y, b; f_y, d; f_c, b; and f_c, d.

The best performing π terms set, for the four possible basis variables sets, is determined by comparing the performance of the different sets for the same network architecture—and hence model complexity. The 31 data points given in Vanluchene and Sun [19] formed the data set, and the value of Pearson's coefficient of correlation (R) for the entire data set provides the performance measure of the neural network model. The performances of the different representations can also be compared as the model complexity changes and as the size of the training data set is reduced, while keeping the size of the total data set the same. By this process the π terms set that would lead to the least complex mapping function can be determined. Fig. 1 shows the performance of the networks for the different representations and network complexity (defined by the number of nodes in the hidden layer), with 11 data points in the training set. The best performing π terms set corresponds to f_y and d as the basis variables. The π terms in this set are given by

$$\pi_1 = \frac{A_s}{d^2}, \quad \pi_2 = \frac{f_c}{f_y}, \quad \pi_3 = \frac{b}{d}, \quad \pi_4 = \frac{M_u}{f_y d^3}$$

6.2. Analysis problem

For the analysis problem, π_1, π_2 and π_3 form the input variables for the neural network model and these are combined in the second stage of the process to enhance the representation and from which the high-level variables are to be selected. The first level of enhancement results in the following set of input variables:

$$\pi_1, \quad \pi_2, \quad \pi_3, \quad \pi_1\pi_2, \quad \pi_1\pi_3, \quad \frac{\pi_1}{\pi_2}, \quad \frac{\pi_1}{\pi_3}, \quad \pi_2\pi_3, \quad \frac{\pi_2}{\pi_3}, \quad \frac{\pi_3}{\pi_2}$$

based on π_1 and π_4 as the main variables of interest. As described in the previous section, a linear network is initially trained using the above 10 variables as the input and π_4 as the output. The coefficient of correlation for the entire data set is found to be 0.99981, indicating that almost all of the non-linearity in the function can be incorporated into the representation of the input variables. Dimensionality reduction of the input variable space is achieved by using sensitivity information to prune one variable at a time till only two most significant variables remain.

The two input variables that remain at the end of the above process are π_1 and $\pi_2\pi_3$. A linear network, with these two as the input variables, has a coefficient of correlation for the entire data set of 0.99983. The three criteria used for determining high-level variables are satisfied by these variables—input dimensionality

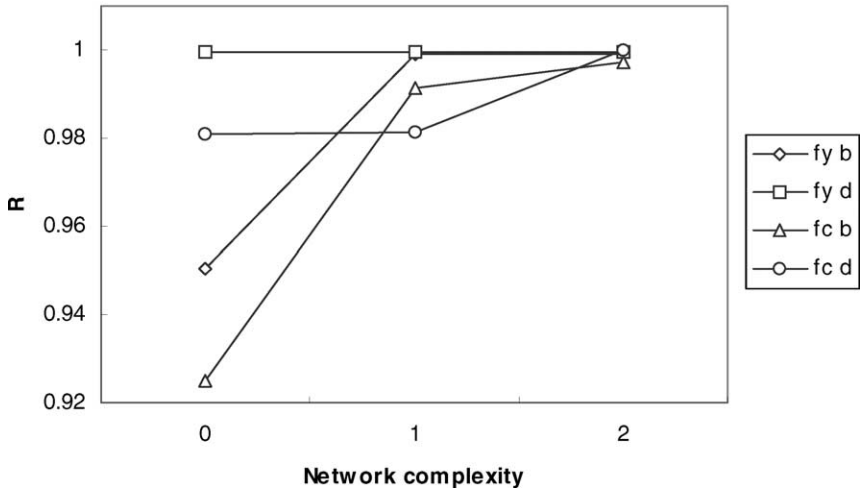

Fig. 1. Performance of networks for different basis variables for the concrete problem.

reduction (from 3 to 2), all the original π terms should be included and no significant drop in performance of network (there is actually a slight improvement mainly due to the improved generalisation capability as a consequence of dimensionality reduction). The final network model can be constructed using these two input variables and the best performing network is MLP 2-3-1 with coefficient of correlation 0.99999.

The improved performance of the MLP network indicates that it may be possible to incorporate more of the non-linearity into the input variables. Thus, a second level enhancement of the representation based on the input variables π_1 and $\pi_2\pi_3$ is carried out using the same functional-expansion model as before. The enhanced input variables are:

$$\pi_1, \quad \pi_2\pi_3, \quad \pi_1\pi_2\pi_3, \quad \frac{\pi_1}{\pi_2\pi_3}$$

Sensitivity information provided by the linear network is again used to prune the input variables. π_1 and $\pi_2\pi_3$ are again the two most significant variables. Functional-expansion model, that includes square of the input variables, is considered for the third level of enhancement. The new input variables are:

$$\pi_1, \quad \pi_2\pi_3, \quad \pi_1^2, \quad (\pi_2\pi_3)^2, \quad \frac{\pi_1^2}{\pi_2\pi_3}, \quad \frac{\pi_1}{(\pi_2\pi_3)^2}$$

Using the same pruning procedure as before results in π_1 and $\pi_1^2/\pi_2\pi_3$ being the two most significant input variables. The coefficient of correlation for a linear network with these two input variables is 0.99998, which is an improvement on the network model at the end of the first level of enhancement and subsequent pruning of the input space. The improved value of the coefficient of correlation also implies that more of the non-linearity in the mapping function is incorporated into the high-level variables, and accordingly a MLP 2-1-1 network is adequate to develop a model, with a correlation coefficient of 0.99999. The theoretical mapping function is linear in the two high-level variables and the equation can be recovered from the linear network. The lower value for the coefficient of correlation (0.99998) may be due to noise in the data.

6.3. Design problem

For the design problem, as posed previously, the input variables are π_2, π_3 and π_4, and the output variable is π_1. The search for the high-level variables also involves enhancement of representation followed by pruning of the input space to determine the two most significant input high-level variables. The first level of enhancement results in 10 input variables, similar to those for the analysis problem, and the coefficient of correlation for a linear network with these inputs is 0.99975. Again indicating that almost all of the non-linearity in the problem can be incorporated into the enhanced representation.

Table 4
Network architectures and performance indicators for five different representations for the concrete beam problem

Variables	Network	R_{tra}/S.D. ratio	R_{ver}/S.D. ratio	R_{tot}/S.D. ratio
Input: A_s, b, d, f_c and f_y Output: M_u	MLP 5-3-1	0.9999817 0.006077	0.999898 0.01521	0.9999302 0.0121937
Input: π_1, π_2, π_3 Output: π_4	MLP 3-3-1	0.9999828 0.005864	0.999973 0.00762	0.9999775 0.006805
Input: first two from PCA Output: π_4	MLP 2-3-1	0.9993156 0.037	0.9999787 0.00741	0.9996782 0.0255
Input: from auto-associative network Output: π_4	MLP 2-3-1	0.9944576 0.1051493	0.9957813 0.09754	0.9944596 0.1074748
Input: $\pi_1, \pi_1^2/\pi_2\pi_3$ Output: π_4	MLP 2-1-1	0.9999897 0.00454	0.9999864 0.005445	0.9999878 0.005021

Pruning the input space using sensitivity information and criteria identified previously for dimensionality reduction, results in π_4 and $\pi_2\pi_3$ as the two input variables at the end of the first level of enhancement. The coefficient of correlation for a linear network with these input variables is 0.99979, and the performance can be further improved to 0.99998 using a MLP 2-3-1 network. This again indicates that more non-linearity can be incorporated into the representation, and further enhancements followed by pruning of the input space results in the high-level variables π_4 and $\pi_4^2/\pi_2\pi_3$. The coefficient of correlation for a linear network with these two variables is 0.99997, and this value can be increased to 0.99999 with a MLP 2-3-1 network. Though the network learns the inverse function in the design problem, the high-level variables have the same form as the analysis problem, indicating that for both problems a good approximation for the required function can be obtained by expanding up to the second-order terms.

6.4. Effect of representation on performance

In order to demonstrate the effectiveness of the present approach, networks are constructed for the concrete beam analysis problem using a number of different representations. The first is based on the dimensional data, without any dimensionality reduction, and the network has five input variables and one output variable. The three π terms at the end of stage 1 of dimensionality reduction are used in the second. The dimensionality of the input variable space, at the end of stage 1, is further reduced using PCA and auto-associative networks for the third and fourth representations, respectively. Finally, the representation arrived at the end of stage 2 of the present approach.

The performances of the networks for the different representations are given in Table 4 for training, verification and total data sets. Performance indicators used for measuring performance are both coefficient of correlation and standard deviation ratio (error standard deviation/data standard deviation).

All networks are based on logistic activation function for the hidden layer nodes and linear activation function for the output nodes and use quasi-Newton training algorithm. The present approach provides the most concise network and the best performance indicators. The approaches based on domain independent techniques, for dimensionality reduction even from the end of stage 1, require a more complex network and result in a significant drop in performance. As indicated previously, the main weaknesses in the domain independent techniques are that only information available in the input space is used and there is some loss of information during the dimensionality reduction process. The next best performing network is based on dimensionless variables at the end of stage 1, but it has an increased dimensionality and consequently a more complex network.

6.5. Elastica problem

The variables that characterise the elastica large deflection column buckling problem are P (axial load), δ

(lateral deflection at the free end), E (Young's modulus), I (second moment of area of column section) and l (length of column). The variables of interest for the analysis problem are P and δ, and for the exponents of the variables in the π terms to be integers, the basis variables are uniquely determined as E and l. The π terms for these basis variables are:

$$\pi_1 = \frac{P}{El^2}, \quad \pi_2 = \frac{\delta}{l}, \quad \pi_3 = \frac{I}{l^4}$$

A network with π_1 and π_3 as input variables and π_2 as the output variable can be trained to model this problem. The dimensionality of the problem can be further reduced using the present approach by enhancing the input representation space followed by pruning using sensitivity information of the network for the enhanced representation. Thus, at the end of first level of enhancement, the input variables are:

$$\pi_1, \quad \pi_3, \quad \pi_1\pi_3, \quad \frac{\pi_1}{\pi_3}$$

based on π_1 as the main variable of interest. The data set, having 36 data points (28 training and 8 verification), is generated by solving the elliptic integral of the first kind, and defined previously, using Mathematica. A linear network is initially trained using the above four variables as the input and π_2 as the output. The coefficient of correlation for the entire data set is found to be 0.71510, indicating that the function to be learned is considerably non-linear and consequently a linear network is not adequate for stage 2 of the process. Next, a flat network with logistic activation function for the output node is considered, and the coefficient of correlation for the entire data set for this network is 0.93740. This is considerably better than the linear network and hence this network is adequate for carrying out the search for the high-level variable.

Using sensitivity information to prune the input variables progressively results in π_1/π_3 as the most significant variable. The correlation coefficient for the flat network with this input variable is 0.93313, and hence the criteria for identifying the high-level variable are satisfied. The final network with π_1/π_3 as the input variable is MLP 1-4-1 with correlation coefficient of 0.99899 on the full data set. Table 5 compares the performance of neural networks for the representations at the end of stages 1 and 2.

Table 5
Network architectures and performance indicators for dimensionless and high-level variables for the column buckling problem

Variables	Network	R_{tra}/S.D. ratio	R_{ver}/S.D. ratio	R_{tot}/S.D. ratio
Input: π_1, π_3	MLP 2-7-1	0.9990528	0.9992375	0.9990564
Output: π_2		0.04352	0.04138	0.04351
Input: π_1/π_3	MLP 1-4-1	0.9991001	0.9993051	0.998994
Output: π_2		0.04242	0.05532	0.0454153

It can be seen from Table 5 that for comparable network performance, the high-level variable, arrived at the end of stage 2, gives rise to a more concise model. The increased complexity of the network, compared to the concrete problem, reflects the increased non-linearity of the mapping function. Despite this increased non-linearity, it is possible to determine the high-level variable from sensitivity information provided by a flat network having logistic activation function for the output unit.

It is also possible to determine the high-level variable using near flat networks, but increasing the network complexity further can lead to the network forming the required function mainly from low-level variables, and the criteria for detecting high-level variables would not be satisfied. Hence the need to start the search with flat networks, so that mainly the high-level variables would be used in constructing the function required for minimising the errors.

6.6. Analysis of bent beam problem

In this example we consider the modelling of a bent beam, rigidly fixed at one end and subject to a concentrated load P at the free end and distributed load p on the free arm of the bent beam, to predict the deflection at the free end of the fixed arm of the beam [20]. The variables, in addition to the loads, that characterise the problem are the spans a and b of the arms of the bent beam, E (Young's modulus), I (second moment of area of beam section) and δ (deflection).

Two dimensions—force and length—are adequate to represent all the above variables. Based on the requirement that the variables of interest—loads and deflection—should only occur in the numerator and other criteria stated previously, the possible bases are:

$$E, a \quad \text{and} \quad E, b$$

The best performing network has E and b in the basis and the following π terms set:

$$\pi_1 = \frac{P}{Eb^2}, \quad \pi_2 = \frac{p}{Eb}, \quad \pi_3 = \frac{a}{b},$$
$$\pi_4 = \frac{I}{b^4}, \quad \pi_5 = \frac{\delta}{b}$$

The dimensionality of the problem is thus reduced from 7 to 5. These dimensionless variables, derived at the end of stage 1 of the process, can be used to develop a neural network model with π_1–π_4 as the input variables and π_5, the response variable, as the output. Further reduction in dimensionality is achieved, as before, by forming high-level variables from the π set, by enhancing the representation of the input space and then pruning it using sensitivity information provided by the network. The input variables at the end of first level of enhancement are:

$$\pi_1, \quad \pi_2, \quad \pi_3, \quad \pi_4, \quad \pi_1\pi_3, \quad \pi_1\pi_4, \quad \frac{\pi_1}{\pi_3},$$
$$\frac{\pi_1}{\pi_4}, \quad \pi_2\pi_3, \quad \pi_2\pi_4, \quad \frac{\pi_2}{\pi_3}, \quad \frac{\pi_2}{\pi_4}, \quad \pi_3\pi_4,$$
$$\frac{\pi_3}{\pi_4}, \quad \frac{\pi_4}{\pi_3}, \quad \frac{1}{\pi_3\pi_4}$$

The last two variables in the enhanced set may be discarded, as they are reciprocals of the previous two variables. Introducing them ensures that as much of the non-linearity as possible is incorporated in the representation. These two additional terms also help to establish the effectiveness of the pruning process in identifying the high-level variables, as they enlarge the search space.

The search process for the high-level variables was conducted using information available in 60 data points, with 40 for training, 10 for verification and 10 for testing. The testing set provides an objective means of comparing generalisation capabilities of the network models based on different representations, and demonstrates that dimensionality reduction by the present method enhances the generalisation capability of the network model.

The pruning process with linear network reduced the input variable set to:

$$\pi_2\pi_3, \quad \pi_3\pi_4, \quad \frac{\pi_2}{\pi_4}$$

This does not satisfy the requirements that all the original π terms should be included as well as satisfactory performance of network. For the next level of network complexity, a near flat network with a single node in the hidden layer gave the input variables as:

$$\pi_2\pi_3, \quad \frac{\pi_2}{\pi_4}, \quad \frac{\pi_3}{\pi_4}$$

This is also not acceptable, as it does not include π_1. The network complexity was progressively increased and the network with three nodes in the hidden layer gave an acceptable input set as:

$$\pi_1, \quad \pi_4, \quad \pi_2\pi_3$$

As expected, when the network complexity is low, all the input variables selected are either products or ratios of the π terms. Whereas, when the complexity increases, the input variables are a mix of the original and higher order terms, as part of the non-linearity in the representation of the variables is incorporated into the internal function representation of the network. Even though the less complex networks did not provide an acceptable set of input variables, the variables identified, such as $\pi_2\pi_3$ and π_2/π_4, have some of the features of the required high-level variables.

A neural network model can be developed with π_1, π_4 and $\pi_2\pi_3$ as the input variables and π_5 as the output variable. The best performing network was MLP 3-5-1 and had coefficients of correlation for training, verification and testing of 0.99982, 0.99987 and 0.99966, respectively. The performance of the network has improved, due to the reduction in dimensionality of the input space, in comparison to network with the original π terms set (see Table 6). Possibility for further reduction in dimensionality can be explored through another cycle of enhancement and pruning of the input space.

The second level of enhancement of the representation results in the following terms:

$$\pi_1, \quad \pi_4, \quad \pi_2\pi_3, \quad \pi_1\pi_4, \quad \frac{\pi_1}{\pi_4}, \quad \pi_1\pi_2\pi_3,$$
$$\pi_2\pi_3\pi_4, \quad \frac{\pi_2\pi_3}{\pi_4}$$

A linear network is adequate to determine the two most significant variables as:

$$\frac{\pi_1}{\pi_4}, \quad \frac{\pi_2\pi_3}{\pi_4}$$

Table 6
Network architectures and performance indicators for the different representations for the bent beam problem

Variables	Network	R_{tra}/S.D. ratio	R_{ver}/S.D. ratio	R_{tes}/S.D. ratio
Input: P, p, a, b, E, I	MLP 6-8-1	0.999966	0.9982125	0.998467
Output: δ		0.008245	0.06055	0.08032
Input: π_1, π_2, π_3, π_4	MLP 4-6-1	0.999353	0.9991565	0.9986674
Output: π_5		0.03597	0.04247	0.05855
Input: π_1, $\pi_2\pi_3$, π_4	MLP 3-5-1	0.9998227	0.9998692	0.9996562
Output: π_5		0.01883	0.017561	0.0279297
Input: π_1/π_4, $\pi_2\pi_3/\pi_4$	LIN 2-1	0.9999981	0.9999951	0.9999967
Output: π_5		0.001951	0.00328	0.002818

The sensitivity ratios for these two variables are very large compared to the other input variables, and hence the examination of the sensitivity information for the input variables of a single linear network is adequate to prune the input variable space to achieve the required dimensionality reduction. The values for the coefficient of correlation for the training, verification and testing sets are *0.9999981*, *0.9999951* and *0.9999967*, respectively, indicating the linear relationship that exists between the high-level input variables and the output variable.

It is clear from Table 6 that determining the high-level variables can not only reduce the complexity of the network, but also leads to improved performance of the network model. The proposed method has thus reduced the dimensionality of the problem from 7 to 3, and during this process identified a number of network models with improved performance characteristics to finally arrive at the most concise linear network model.

In order to study the effect of data set on the proposed method, a sub-set (30 training and 15 verification) of the original data set was used for stage 2 of the process. At the end of first level of enhancement of the representation and subsequent pruning of the input space, the linear and near flat networks (hidden layer with 1 and 2 nodes), identified $\pi_2\pi_3$, $\pi_3\pi_4$ and π_1/π_4 as the most significant variables. These, however, did not satisfy the criteria for network performance, even though they included all the π terms, and were discarded. On increasing the network complexity further, by introducing an additional node in the hidden layer, the most significant variables were identified as π_3, π_1/π_4 and π_2/π_4. This set satisfied all the criteria for acceptable higher order variables, and a network trained with these as input variables (*MLP 3-3-1*) with the original data set gave the performance indicators for the training, verification and testing sets as *0.9999956*/*0.002971*, *0.9999804*/*0.007257* and *0.9999956*/*0.003312*, respectively.

A second level enhancement of the representation for the above three higher order variables and subsequent pruning of the input variable space results in the same high-level variables, π_1/π_4 and $\pi_2\pi_3/\pi_4$, as before. Thus, using different data sets can result in the process searching along different paths to arrive finally at the unique set of high-level variables.

7. Conclusions

It has been demonstrated that a two-stage dimensionality reduction process can be applied to determine high-level variables required for developing a concise neural network model of physical systems. In general the network performance indicators—both the correlation coefficient and S.D. ratio—improve with dimensionality reduction. This is to be expected as dimensionality reduction by combining the original variables means that, through this process, each original data point can be used to generate infinite set of data points in the original representation space. Thus, dimensionality reduction at the end of both stages of the process enhances the information content in the original data set and accounts for the improved performance of the models. Neural networks can learn the underlying function even in the presence of noise in the data, and it is expected that the present approach

would also be able to detect the high-level variables in the presence of noise, provided the usual precautions are taken for preventing overfitting. Thus, the improved generalisation performance of the concise network model generated by this approach should also be realised with noisy data. Though the approach has been developed in the context of physical systems, this approach is also applicable to physical processes.

Acknowledgements

This research has been funded by a grant from the Australian Research Council.

References

[1] D.J. Gunaratnam, M.L. Maher, Case-based design tool for teaching structural design, in: Proceedings of the 31st Annual Conference of ANZAScA on Principles and Practice, Pictorial Press, Brisbane, Australia, 1998, pp. 177–184.
[2] J.S. Gero, A design method for cable network structures, in: Proceedings of the IASS Conference on Lightweight Shell and Space Structures, Mir, Moscow, 1977, pp. 84–94.
[3] D.J. Gunaratnam, Integrated design charts for reinforced concrete flexural sections, in: Proceedings of the Institution of Civil Engineers: Part 2, vol. 83, June 1987, pp. 443–451.
[4] D.J. Gunaratnam, A.P. Bhattacharya, Transverse vibrations and stability of polar orthotropic circular plates: high-level relationships, J. Sound Vib. 137 (3) (1989) 383–392.
[5] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, 1995.
[6] StatSoft, STATISTICA Neural Network, StatSoft Inc., USA, 1998.
[7] S. Rudolph, On topology size and generalisation of non-linear feed-forward neural networks, Neurocomputing 16 (1997) 1–22.
[8] D.J. Gunaratnam, J. Gero, Effect of representation on the performance of neural networks in structural engineering applications, Microcomputers Civil Eng. 9 (1994) 97–108.
[9] D.J. Gunaratnam, J.H. Garrett, Incorporating domain knowledge in neural network learning, in: C.H. Dagli (Ed.), Intelligent Engineering Systems Through Artificial Neural Networks, vol. 5, ASME (Mech. Engineers), New York, 1995, pp. 203–208.
[10] S.J. Kline, Similitude and Approximation Theory, McGraw-Hill, New York, 1965.
[11] R.C. Pankhurst, Dimensional Analysis and Scale Factors, Chapman & Hall, London, 1964.
[12] S. Rudolph, On a genetic algorithm for the selection of optimally generalizing neural network topologies, in: Proceedings of the Second International Conference on Adaptive Computing in Engineering Design and Control, Plymouth, 1996, pp. 79–86.
[13] E. Vonk, L.C. Jain, R.P. Johnson, Automatic Generation of Neural Network Architecture Using Evolutionary Computation, World Scientific, Singapore, 1997.
[14] R. Bhaskar, A. Nigam, Qualitative physics using dimensional analysis, Artif. Intell. 45 (1–2) (1990) 73–111.
[15] S.P. Timoshenko, J.M. Gere, Theory of Elastic Stability, McGraw-Hill, New York, 1961.
[16] D.J. Gunaratnam, T. Degroff, F. Fricke, Dimensionality reduction in neural network modelling using dimensional theory and genetic algorithm, in: C.H. Dagli (Ed.), Intelligent Engineering Systems Through Artificial Neural Networks, vol. 10, ASME (Mech. Engineers), New York, 2000, pp. 133–138.
[17] A. Ratle, M. Sebag, Grammar-guided genetic programming and dimensional consistency: application to non-parametric identification in mechanics, Appl. Soft Comput. 1 (2001) 105–118.
[18] Y. Pao, Adaptive Pattern Recognition and Neural Networks, Addison-Wesley, USA, 1989.
[19] R.D. Vanluchene, R. Sun, Neural networks in structural engineering, Microcomputers Civil Eng. 5 (1990) 207–215.
[20] T.H. Richards, Energy Methods in Stress Analysis, Ellis Horwood, England, 1977.