On the optimization of artificial neural networks for application to the approximation of chemical systems

By M. Ihme, A. L. Marsden AND H. Pitsch

1. Motivation and objectives

An artificial neural network (ANN) is a computational model for storing and retrieving acquired knowledge. ANNs consist of dense interconnected computing units that are simple models for complex neurons in biological systems. The knowledge is acquired during a learning process and is stored in the synaptic weights of the inter-nodal connections. The main advantage of neural networks is their ability to represent complex input/output relationships. They are well suited for use in data classification, function approximation, and signal processing, among others.

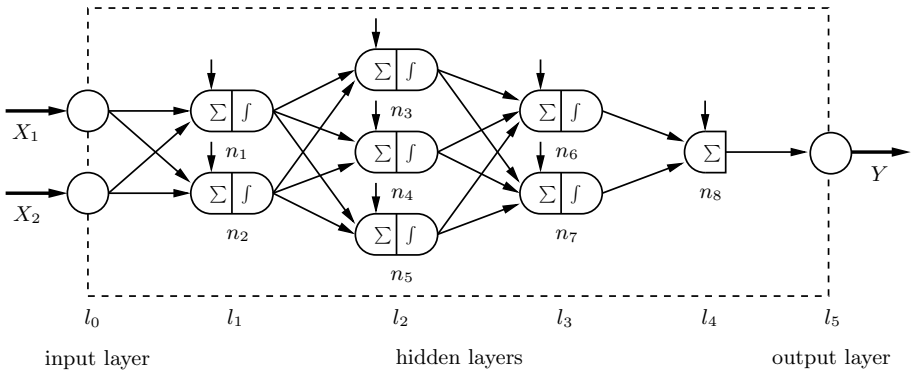
The performance, or fitness, of an ANN is often measured according to an error between target and actual output, training time, complexity of the ANN, or in terms of other properties important for the user. Although the elemental building blocks of a neural network, i.e., neurons, nodal connections, and the transfer functions of nodes are in themselves relatively simple, the various combinations can result in different topologies with similar or vastly different fitness characteristics. Therefore, the *a priori* design of a neural network with near-optimal fitness is not a trivial task and is usually guided by heuristics or trial-and-error. The architecture or topological structure of an ANN can be characterized by the arrangement of the layers and neurons, the nodal connectivity, and the nodal transfer functions. In this work, the class of multi-layer perceptrons (MLPs) is considered, which consists of an input layer with N_I input channels, N_L hidden layers, and an output layer with N_O output channels. The number of neurons in each hidden layer is denoted by $\underline{N}_\mathrm{N}$. Nodes in the first $N_\mathrm{L}-1$ hidden layers are characterized by a non-linear behavior; nodes in the last hidden layer are linear (Haykin 1994).

A multi-layer perceptron is shown in Fig. 1(a). Information in this 2-3-2-1 network propagates uni-directionally from the input to the output channel. Each neuron includes a threshold which is indicated by a vertical arrow in Fig. 1(a). The significance of the threshold is explained in Section 2. In the case of a symmetric network, in which all neurons in a particular layer are connected to the same nodes, the nodal connectivity matrix can be contracted to a layer connectivity matrix $\underline{\underline{L}}$. The matrix entry L_{ij} indicates the connectivity between layer j and layer i. The layer connectivity matrix corresponding to the ANN of Fig. 1(a) is shown in Fig. 1(b).

A particular topology of an ANN, to be denoted by $\mathcal{A}$, can formally be written as

$$\mathcal{A} = \mathcal{A}(N_\mathrm{L}, \underline{N}_\mathrm{N}, \underline{\underline{L}}, \underline{\psi} | N_\mathrm{I}, N_\mathrm{O}, Y, J) \ , \tag{1.1}$$

in which $\underline{\psi}$ is the vector of transfer functions for all nodes, Y is the approximated function, and J is a cost function characterizing the performance of the network. The last four arguments in Eq. (1.1) are constrained by the problem and the desired performance of the network. The design parameters N_L, $\underline{N}_\mathrm{N}$, $\underline{\underline{L}}$, and $\underline{\psi}$ determine the topology of the neural



(a) network architecture

$$\underline{\underline{L}} = \begin{array}{c} \\ l_0 \\ l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \end{array} \begin{array}{c} \begin{array}{cccccc} l_0 & l_1 & l_2 & l_3 & l_4 & l_5 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

(b) layer connectivity matrix

FIGURE 1. Multi-layer perceptron: (a) architecture of the MLP consisting of one input layer with two input channels X_1 and X_2, 4 hidden layers with, respectively, 2, 3, 2, and 1 neurons in each layer. The neurons in the hidden layers are denoted by n_i, with $i = 1, \ldots, 8$; (b) corresponding layer connectivity matrix.

network and have to be chosen in such a way that the ANN has an optimal performance characteristic. While small networks with only a few connections and synaptic weights are often limited in their capability to perform a certain task, large ANNs with many nonlinear nodes can result in poor generalizability, long training time, and computationally expensive knowledge retrieval (Yao 1999).

Over recent years considerable research has been conducted on the evolution of topological structures of networks using evolutionary algorithms (Koza & Rice 1991; Bornholdt & Graudenz 1992; Tang *et al.* 1995; Angeline *et al.* 1994; Miller *et al.* 1989; Husken *et al.* 2005; and references in Yao 1999). However, very little work has been carried out on evolving node-specific transfer functions or the simultaneous evolution of both nodal arrangement and transfer functions (Liu & Yao 1996; Hwang *et al.* 1997). This shortcoming is addressed in the present work by proposing a methodology that can be used to simultaneously optimize the nodal arrangement, connectivity, and transfer functions of a neural network using a generalized pattern search (GPS) method developed by Torczon (1997). The GPS method is complemented by a surrogate management framework (SMF), developed by Serafini (1998) and Booker *et al.* (1999), in order to increase the efficiency of pattern search methods for computationally expensive problems. In this method, the expensive cost function is approximated by a surrogate function based on Kriging (Koehler & Owen 1996). The GPS algorithm is a derivative-free method and provides robust con-

vergence properties. This method has been used previously by Booker *et al.* (1999) for rotorblade optimization and by Marsden *et al.* (2004) for trailing-edge airfoil optimization. Recently, Audet & Dennis (2000) extended the GPS method to problems with mixed design variables with bound constraints. This method, the so-called generalized mixed variable pattern search (GMVPS), was employed in the work of Kokkolaras *et al.* (2000) to design a thermal insulation system. Here, GMVPS will be use to optimize the nodal connectivities and transfer function types of each neuron.

The main objective of this work is to use automatically generated neural networks for to approximate non-linear functions, e.g., those encountered in representing chemically reactive systems (Christo *et al.* 1996*a*,*b*; Blasco *et al.* 1999*a*,*b*, 2000; Chen *et al.* 2000; Flemming *et al.* 2005). Chemical kinetic reaction mechanisms are often comprised of thousands of chemical reactions among hundreds of species. In numerical simulations of combustion systems, for example, the direct solution of transport equations for all these species is usually not feasible, partly because of the large range of chemical time scales. To reduce the number of independent species, dimensional reduction methods have been developed, in which a part of the chemical species are projected onto lower dimensional manifolds, parameterized by the remaining species. For instance, in the case of the steady flamelet model (Peters 1984), all chemical and thermodynamic quantities are represented using only two scalars. In the case of turbulent combustion and in cases where radiation and pollutant formation are important, accurate predictions might require the number of independent scalars to grow to five or even more (Peters 2000). The functional relation between chemical species and independent scalars is usually stored in conventional tables. The size of the chemistry table increases with the number of parameters and imposes a drastic restriction on the resolution for more than three independent parameters. ANNs, in contrast, have the potential to accurately approximate complex functions using modest amount of memory. Motivated by the chemistry application, test examples are performed using functions that closely resemble the chemical characteristics of combustion systems. Nevertheless, the proposed method for the optimization of the network architecture is general in nature and applicable to other problems of interest.

The remainder of the paper is organized as follows. Section 2 discusses the ANN model and describes the training process. The GPS and GMVPS methods are presented in Section 3. These algorithms are then applied in the optimization of neural networks. In the first step, only the number of neurons per layer are optimized and the connectivity matrix and type of transfer function are fixed. This optimization problem is solved using the surrogate management framework with standard generalized pattern search. In the following, an example of the approximation of a reactive system is presented in which the nodal arrangement, transfer function, and connectivity are parameters and optimal networks are generated using the GMVPS method. The performance of the automatically generated network is compared with results obtained using a conventional tabulation technique. The paper finishes with a discussion.

2. Artificial neural network

The general structure of an MLP is shown in Fig. 1(a). This particular network consists of $N_{\mathrm{I}} = 2$ input channels and $N_{\mathrm{O}} = 1$ output channels and $\mathcal{A} : \mathbb{R}^{N_{\mathrm{I}}} \rightarrow \mathbb{R}^{N_{\mathrm{O}}}$. Note that the input nodes do not represent neurons in the conventional sense. Here they merely perform the task of normalizing the input data so that the input signal feed into the network is in the interval $[-1, 1]^{N_{\mathrm{I}}}$. The output y_i of each neuron i in the hidden layers

is computed according to

$$y_i = \psi_i \left(\sum_{j=0}^{N_\mathrm{C}} C_{ij} \omega_{ij} x_{ij} \right) , \tag{2.1}$$

where x_{ij} is the input signal and N_C denotes the number of connections in the network. The element C_{ij} is unity if neurons i and j are connected and zero otherwise. The synaptic weights are denoted by ω_{ij}. Each neuron includes a threshold ω_{i0}, which is connected to a constant input signal $x_{i0} = -1$. The effect of the threshold is to lower the net input on the transfer function (Haykin 1994). A sigmoidal function

$$\psi(s) = a_1 \tanh(b_1 s) , \tag{T1}$$

cyclometric function

$$\psi(s) = a_2 \,\mathrm{atan}(b_2 s) , \tag{T2}$$

a linear function

$$\psi(s) = s \tag{T3}$$

or other monotonic functions are commonly used as transfer functions. The parameters a and b in the transfer functions (T1) and (T2) are adjustable by the user. Transfer functions (T1) and (T2) will be used in Section 4.

The synaptic weights in the network are adjusted during the training process, which in itself represents an optimization problem and can be written as

$$\min_{\omega \in \mathbb{R}^{N_\omega}} E(\omega) . \tag{2.2}$$

In the above problem, $E : \mathbb{R}^{N_\mathrm{O} \times N_\mathrm{t}} \rightarrow \mathbb{R}$ is the error between the actual and desired output of the network, which can be written as

$$E = \frac{1}{2\, N_\mathrm{t}} \sum_{j=1}^{N_\mathrm{t}} e^\mathrm{t}(j) \qquad \text{with} \qquad e^\mathrm{t}(j) = \sum_{i=1}^{N_\mathrm{O}} \left[Y_i(j) - Y_i^\mathrm{t}(j) \right]^2 \tag{2.3}$$

and $Y_i^\mathrm{t}(j)$ represents the j^th training sample of the output signal i. The number of training samples is denoted by N_t. The learning process, in which the weights are iteratively adjusted, is guided by the knowledge of the desired input/output examples and is called supervised learning. A Levenberg-Marquardt algorithm (Hagan *et al.* 1996) has been used for the optimization of the synaptic weights. This algorithm is a trust region Gauss-Newton method and its fast convergence makes it suitable for training neural networks.

The main advantage of ANNs is their generalizability, meaning their accurate performance on new data. In this respect ANNs are different from tabulation methods in which discrete values of a particular function of interest are memorized and stored in a table. A good generalization of a particular ANN is dependent on the complexity of the function, the architecture of the network, and the size and information content of the training set. Networks with only a few hidden neurons and synaptic weights are restricted in the approximation of complex functions, resulting in poor generalizability characterized by so-called under-fitting. On the other hand, a network that is too complex, meaning that there are considerably more synaptic weights than training samples, can result in possible over-fitting and non-smooth function approximation.

The appropriate size of the training set is crucial for the design of an optimal network, and is usually determined by two factors. First, the *a priori* knowledge of the function to be approximated allows for an assessment of the complexity and shape of the function.

Based on this information, the minimum size of the training set required for the description of the functional shape can be determined. A second constraint on N_t is imposed by the size of the network and the number of synaptic weights. The size of the training set for a fixed architecture with N_ω synaptic weights is given by

$$N_\mathrm{t} \geq \alpha N_\omega \ . \tag{2.4}$$

The value of α is chosen based on experience and analysis and typically ranges from approximately 30 for noisy training data down to $2-5$ for the approximation of analytical functions. The lower bound is chosen to avoid the possible occurrence of over-fitting, which can occur if N_t is significantly smaller than αN_ω.

The generalization potential of a trained ANN is assessed using test samples. These samples are used after training to evaluate the ability of the ANN to approximate untrained samples. In this process, the performance of the ANN can be evaluated by means of the following cost function

$$J(\mathcal{A}) = \log_{10}\left(\sqrt{\frac{1}{N_\mathrm{s}}\sum_{j=1}^{N_\mathrm{s}}(e^\mathrm{s}(j))^2}\right) \ , \tag{2.5}$$

where N_s is the number of test samples. The evaluation of the cost function requires that the synaptic weights in $\mathcal{A}$ are fully adjusted during a preceding training process. In order to allow for an objective comparison of the fitness between different network architectures, e^s is normalized to the interval $[0,1]$ so that $J(\mathcal{A})$ represents a relative error.

The resulting optimization problem may be formulated as

$$\begin{aligned} &\min_{N_\mathrm{L},\underline{N}_\mathrm{N},\underline{\underline{L}},\underline{\psi}} J(\mathcal{A}) \\ \text{subject to} \quad & N_\mathrm{L} = N_\mathrm{L}^{\max} \ , \\ & N_{\mathrm{N},i} \in \{0,1,\ldots,N_\mathrm{N}^{\max}\}, \qquad i = 1,2,\ldots,N_{\mathrm{L}-1} \ , \\ & N_{\mathrm{N},N_\mathrm{L}} = 1 \ , \\ & \underline{\underline{L}} \in \mathcal{L}, \quad \underline{\psi} \in \mathcal{P} \ , \end{aligned}$$

where $\mathcal{P}$ is the finite list of transfer functions and $\mathcal{L}$ denotes the finite set of possible layer connectivity matrices. The continuous variables are N_L and $\underline{N}_\mathrm{N}$ and the categorical ones are $\underline{\underline{L}}$ and $\underline{\psi}$.

3. GPS method and GMVPS algorithm

The GMVPS method is an extension of the GPS method (Torczon 1997), and has been developed by Audet & Dennis (2000) to handle mixed variable problems with bound constraints. In the present case, the optimization of a network architecture can be represented as a mixed variable problem in which the number of hidden layers and neurons are (continuous) integer-valued parameters and the transfer functions and connectivity matrix are of categorical type. Categorical variables must take on values from a predefined list or discrete set of elements. Optimization in this mixed variable space is performed to find a network architecture with optimal fitness, subject to certain constraints. In this context note that the pattern search method does not guarantee that the solution is a

global minimum. The limit point of the sequence of iterations, however, corresponds to a local optimal solution, which is defined with respect to a set of neighbors. The neighborhood is specified by the user and accounts for the variation of continuous and categorical variables. A rigorous definition of local optimality for a mixed variable problem is given by Audet & Dennis (2000).

The pattern search algorithm is a derivative-free mesh-based method. The algorithm generates a sequence of iterates, whose cost function is non-increasing (Audet & Dennis 2000). All points at which the cost function is evaluated are restricted to lie on a mesh. The algorithm proceeds in two stages: a search and a poll step. The search step allows for a mesh-restricted local and global exploration of the parameter space. In this step a finite number of search points are evaluated with the objective of identifying a region with a reduced cost function. For instance, random sampling of the parameter space using Latin hypercube sampling (LHS) (McKay *et al.* 1979) or a genetic algorithm can be employed. A considerable cost reduction of the search step can be achieved by employing a less expensive surrogate function. The surrogate function is often an approximation of the cost function. The shape of the function is continuously updated by incorporating all previously evaluated points of the cost function. This surrogate function is then used to identify a new point with a potentially lower cost function. However, the surrogate function can be used only in the continuous parameter space. For categorical variables, the above-mentioned techniques have to be employed.

In the present work, Kriging approximation is employed as the surrogate function. Kriging is a statistical method and is based on the use of spatial correlation functions. Its multi-dimensional extension makes this method attractive for optimization problems with several parameters. A summary of the Kriging approximation is given by Marsden *et al.* (2004). Details of implementation are described by Lophaven *et al.* (2002).

In the case of an unsuccessful search step, a poll step in the neighborhood of the incumbent point with the lowest cost function is executed. All continuous-valued points are restricted to lie on a mesh that is constrained to the parameter space. For a mixed variable problem, the mesh M at iteration k is defined to be the direct product of the categorical variable space Ω^c and the lattice in the continuous variable space

$$M_k = \Omega^c \times \left\{ \underline{\eta}_k + \Delta_k \underline{\underline{D}}\, \underline{\varsigma} : \underline{\varsigma} \in \mathbb{N}^{N^D} \right\} , \tag{3.1}$$

where $\Delta_k > 0$ is the mesh size parameter and $\underline{\underline{D}}$ is an $N^c \times N^D$ matrix whose columns form a positive spanning set. If $\underline{\underline{I}}$ denotes the identity matrix and $\underline{\iota}$ is the vector of ones, then $\underline{\underline{D}}$ is typically chosen as $\underline{\underline{D}} = [\underline{\underline{I}}, -\underline{\iota}]$ or $\underline{\underline{D}} = [\underline{\underline{I}}, -\underline{\underline{I}}]$ (Audet & Dennis 2000).

The poll step consists of a local search in the mesh neighborhood around the current best point and, in the case that the categorical variable space is not empty, also in the set of the categorical neighbors. Polling is conducted in three steps (Audet & Dennis 2000):

- polling with respect to the continuous variables and fixed categorical parameters,
- polling in the neighborhood of categorical variables and fixed continuous parameters,
- extended polling around the neighborhood of points whose cost function is close to the incumbent value.

The continuous poll step is an evaluation of adjacent mesh points forming a positive spanning set. This step is augmented with the evaluation of points in the discrete neighborhood. If these poll steps are unsuccessful extended polling is performed. Extended polling around promising points is triggered when the previous poll steps are unsuccess-

ful. A promising point ζ for the extended polling is defined as

$$J(\eta_k) < J(\zeta) \leq (1+\xi)J(\eta_k)\ , \tag{3.2}$$

where $\xi > 0$ is the poll trigger. If $J(\zeta)$ is close to $J(\eta_k)$ of the current best point, polling around the continuous neighborhood of ζ is performed with the expectation of finding a better point. If a point ζ^* is found with $J(\eta_k) < J(\zeta^*) < J(\zeta)$, extended polling around this point is continued. This continues until either a new incumbent is found or until all points in the extended poll set are inferior to the incumbent. If all poll steps are unsuccessful, the mesh is refined and a new iteration, starting with a search step, is performed. More details on the algorithm and convergence proofs can be found in Audet & Dennis (2000) and Abramson (2004).

4. Results

In this section, different aspects of network optimization are explored, including optimization of the nodal arrangement, type of transfer function, and connectivity.

In Section 4.1 the nodal arrangement of the ANN is optimized using the standard GPS together with surrogate approximation. In this case, the hidden layers and neurons of the network are free parameters. In Section 4.2, the optimization method is applied to the approximation of chemical systems with optimal ANNs. In this application the transfer function type and connectivity matrix are added as optimization parameters, and the network dependence on $N_\mathrm{L}, \underline{N}_\mathrm{N}$, and $\underline{\underline{C}}$ is optimized subject to the cost function constraint. The nodal transfer function and the connectivity are parameters of categorical type and therefore require that the GMVPS method be applied. Results for two different optimization processes are discussed.

4.1. *Optimal nodal arrangement*

In this section, the GPS method is applied to optimize the number of neurons in a fully connected MLP. The number of neurons in the first hidden layer is a free parameter and its bounds are $0 \leq N_{\mathrm{N},1} \leq 32 = N_\mathrm{N}^\mathrm{max}$. A hyperbolic tangent function with $a = 1.075$ and $b = 2$ is used as the transfer function. The number of test samples and training samples are equal and set to 350 so that the coefficient α in Eq. (2.4) is $\alpha \geq 2.7$. It can be expected that in the limit of small $N_{\mathrm{N},1}$, the network suffers from under-fitting and over-fitting might be observable for large $N_{\mathrm{N},1}$ due to small α-values. For the training of the individual networks during the GPS optimization, at most 200 iterations in the Levenberg-Marquardt algorithm are used.

To demonstrate the use of this method, a network with a scalar output, dependent on two input channels, is optimized. Mimicking a chemical reaction rate dependent on two input channels, the following analytical function is used:

$$Y = \exp\left\{ -\sum_{i=1}^{N_\mathrm{I}=2} \frac{(X_i - X_i^0)^2}{\sigma_i} \right\}\ , \tag{4.1}$$

with $\mathrm{X}^0 = (0.5, 0.5)^\intercal$, $\underline{\sigma} = (0.01, 0.1)^\intercal$, and $\underline{X} \in [0,1]^2$.

The evolution of the optimization algorithm is shown in Fig. 2. The surrogate function (solid line) is determined by the surrogate points (circles) and comprises all locations at which the cost function has previously been evaluated. Points evaluated during the search step, obtained from the minimization of the surrogate function, are denoted by square symbols. Locations that are evaluated during the poll step are denoted by diamonds

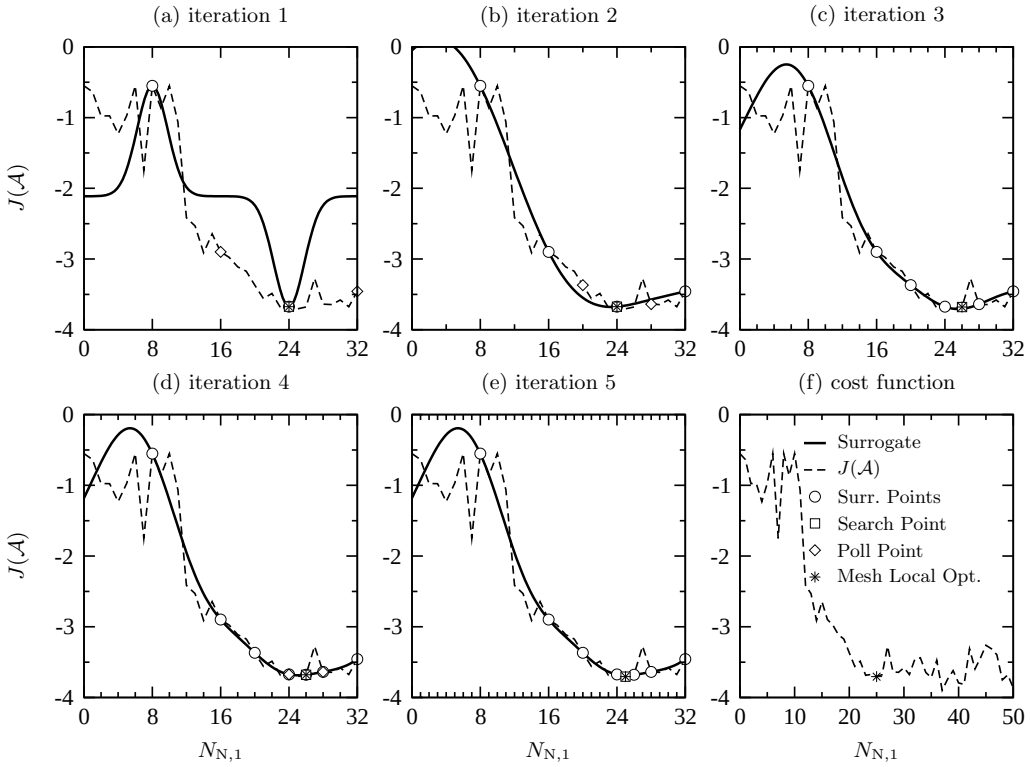
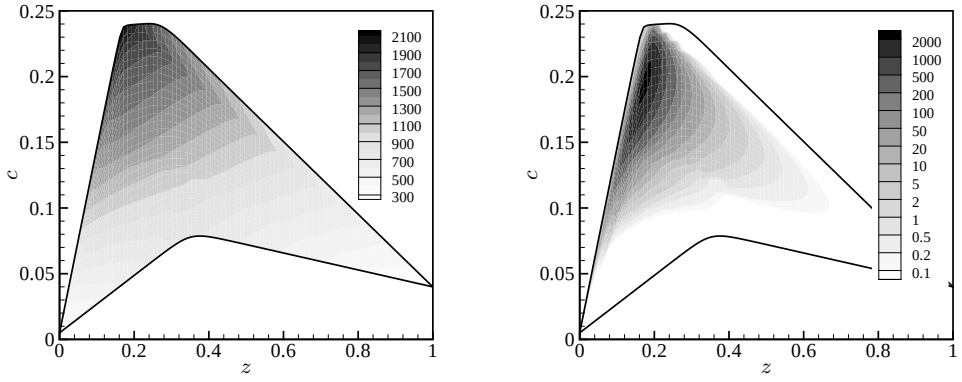


FIGURE 2. Evolution of the GPS optimization for a network with one non-linear hidden layer.

and the current best value for $N_{\mathrm{N},1}$ resulting in the best ANN performance is indicated by a star. In the first iteration (Fig. 2(a)) the minimum of the surrogate, evaluated during the search step, corresponds to the current best point. The search step is declared unsuccessful and polling around the previous optimal location is performed. The poll step does not result in a reduction of the cost function. The mesh size is reduced to $\Delta = 4$ in the succeeding iteration and the surrogate is updated (Fig. 2(b)). Neither the search nor the poll step are successful in the second iteration and the mesh size is further reduced to $\Delta = 2$ (Fig. 2(c)). The minimum of the newly updated surrogate function is computed and the cost function is evaluated, resulting in a reduction of $J(\mathcal{A})$ and therefore a better ANN structure. Both search and poll steps in the fourth iteration (Fig. 2(d)) are not successful and the mesh size is reduced to $\Delta = 1$ in iteration 5. From the updated surrogate function the search step determines a new minimum of the surrogate function for $N_{\mathrm{N},1} = 25$. The cost function for this location is evaluated, resulting in further improvement of the network fitness. The GMVPS algorithm identifies the mesh local optimizer at $N_{\mathrm{N},1} = 25$.

The GPS algorithm requires only eight function evaluations for the optimization of this particular network with one non-linear hidden layer. A function evaluation comprises both the training phase and fitness evaluation during the test process. It is also interesting to point out that two search steps (iterations 3 and 5) lead to reductions in the cost function and all poll steps were unsuccessful.

Figure 2(e) shows that the function evaluations are mainly clustered around the mesh local optimizer. This accumulation of points in the Kriging function can lead to a degra-



(a) Temperature [K] (b) Chemical source term [kg/(m^{3}s)]

FIGURE 3. Solution of the steady laminar flamelet equations as a function of mixture fraction z and progress variable c; (a) temperature and (b) chemical source term.

dation of the surrogate function (Marsden *et al.* 2004). This can be prevented by including space-filling points, which necessarily do not correspond to optimal points.

The evolution of the cost function for $0 \leq N_{\mathrm{N},1} \leq 50$ is shown in Fig. 2(f). For small $N_{\mathrm{N},1}$ the network suffers from under-fitting, resulting in poor generalization. By increasing the number of neurons and hence synaptic weights, $J(\mathcal{A})$ decreases and stays approximately constant for $N_{\mathrm{N},1} \geq 35$. A further increase in the number of synaptic weights results only in marginal improvement of the fitness. This can mainly be attributed to the limited complexity of the network structure (one hidden non-linear layer) and possible over-fitting for large $N_{\mathrm{N},1}$.

4.2. *Chemistry example*

In this section, optimal neural networks for the approximation of a chemical system are generated. In this application, all three optimization strategies, namely optimal nodal arrangement, transfer functions, and connectivity, are combined.

The chemical process under consideration describes methane/air combustion. The GRI 2.11 chemical mechanism (Bowman *et al.* 1997) containing 277 elementary chemical reactions among 49 species is used.

The steady laminar flamelet equations (Peters 1984) are often employed to describe the reaction-diffusion balance in non-premixed flames. The solutions to these equations provide temperature and mass fractions of all species in terms of two parameters. The mixture fraction z and the reaction progress variable c are used for this parameterization. Figure 3 shows the temperature and the chemical production rate of c, which can be viewed as a measure of the heat release. In the following, network optimization for the approximation of these two quantities as functions of these parameters is performed. The solid lines represent the system boundary; flame states outside this boundary are unaccessible. Note that the chemical source term is very localized around $z \approx 0.2$ and $0.15 \leq c \leq 0.23$.

The number of samples used during the training phase and testing process is 500. Sample data are obtained by applying an acception-rejection method (Rubinstein 1981);

Quantity	Neurons	Transfer Function	$\underline{\underline{L}}$	$J(\mathcal{A})$
Temperature	6-7-4-7-1	(T2)-(T2)-(T2)-(T2)-(T3)	$\begin{pmatrix} 0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0 \\ 1&1&0&0&0&0&0 \\ 1&0&1&0&0&0&0 \\ 1&0&0&1&0&0&0 \\ 1&0&0&0&1&0&0 \\ 0&0&0&0&0&1&0 \end{pmatrix}$	-2.780
Chem. Source Term	2-2-8-8-1	(T2)-(T2)-(T2)-(T2)-(T3)	$\begin{pmatrix} 0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0 \\ 0&0&1&0&0&0&0 \\ 0&0&0&1&0&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&0&1&0 \end{pmatrix}$	-3.124

TABLE 1. Optimal network architecture for the approximation of temperature and chemical source term. The transfer function and connectivity matrix are free categorical parameters.

this results in higher sampling density in regions where the functional value Y is large. This method consequently results in a better resolution of the important regions with high chemical source term and temperature.

By considering the optimization of the nodal transfer function and the connectivity in the network design, the categorical neighborhood requires an extension in order to identify a network with optimal fitness in the joint continuous and categorical space. In this example, the categorical neighborhood for the transfer function includes the following combinations: (i) The transfer functions for the neurons in all layers are identical, either of type (T1) or (T2); (ii) Except for the neurons in one layer, the transfer function for the neurons is identical. The set of neighbors for the connectivity contains the following combinations: (a) fully connected feed-forward network; (b) fully connected feed-forward network with $L_{i0} = 1$ where i represents any hidden layer; (c) fully connected feed-forward network with $L_{i0} = 1$ for all i hidden layers. The poll trigger is set to $\xi = 0.05$ and the maximum number of neurons per layer and the number of non-linear hidden layers are set to 8 and 5, respectively.

The optimal network for the approximation of the temperature consists of a 6-7-4-7-1 ANN. All neurons employ a cyclometric transfer function. The connectivity resembles a fully connected feed-forward network in which all neurons are also directly connected to the input layer. The cost function of this network is $J(\mathcal{A}) = -2.780$. The regression analysis for the network is shown in Fig. 4(a).

The network for the approximation of the chemical source term is considerably different from that of the temperature. This is mainly attributed to the different shapes of the functions for temperature and chemical source term. The architecture of the optimal network consists of a 2-2-8-8-1 fully connected feed-forward network. The cost function for this network is $J(\mathcal{A}) = -3.124$ and the regression analysis for this network is shown in Fig. 4(b). Both network characteristics are summarized in Table 1.

It is interesting to compare the network performance with a conventional tabulation technique. Here, a particular chemistry table will be denoted by $\mathcal{T}$ and the parameter space of z and c will be discretized using a Cartesian and equidistant grid. The function to be tabulated is linearly interpolated to grid points. The performance for the tabulation of temperature and chemical source term with increasing resolution is presented in Table 2.

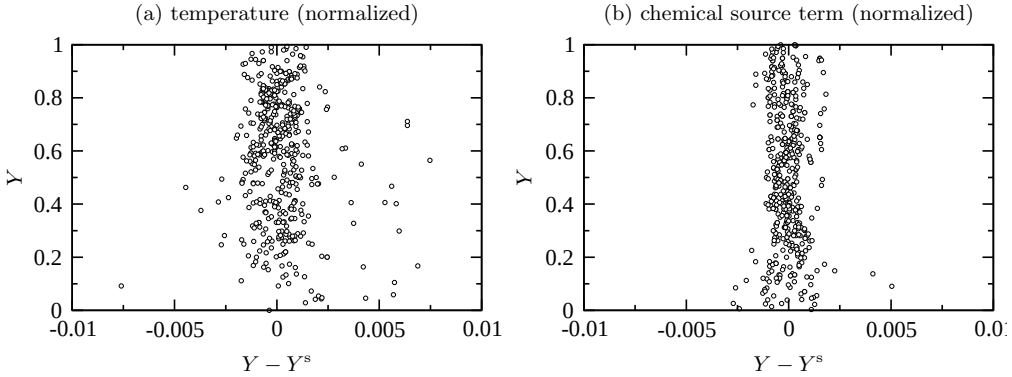


FIGURE 4. Regression analysis for approximation of (a) temperature and (b) chemical source with optimal network architecture. Note that the temperature and chemical source term are here shown in non-dimensional form.

Table size $N_z \times N_c$	Cost Function $J(\mathcal{T})$ Temperature	Chemical source term
100×100	-2.240	-1.677
200×200	-2.580	-2.193
300×300	-2.773	-2.492
400×400	-2.871	-2.621
500×500	-2.974	-2.791

TABLE 2. Comparison of the performance for the representation of temperature and chemical source term using conventional tabulation technique.

In order to allow for an objective comparison between the methods, the same set of test samples is used. Generally it can be observed that the performance of the tabulation method increases monotonically with finer resolution. The resolution required for an accurate chemistry representation grows rapidly and may impose a rather restrictive bound when the number of parameters is larger than three. It is interesting to point out that for an equivalent representation of the temperature using the optimal ANN described above, a table with about 300×300 grid points in z and c direction would be required. By comparing the fitness of the ANN and tabulation for the representation of the chemical source term (Tables 1 and 2), it can be concluded that more than 500×500 grid points are required for a table to achieve an accuracy comparable with the network approximation.

5. Discussion

Optimization of an ANN has been successfully carried out using standard pattern search methods in conjunction with a surrogate function and generalized mixed variable pattern search. These methods offer extensive flexibility in implementation and freedom in choice of parameters. Using a model problem, optimization for the number of neurons,

the transfer function, and the connectivity was explored. It was demonstrated that the use of categorical variables for the transfer function and connectivity resulted in identification of better performing ANNs compared with a trial-and-error approach.

Based on experience from the model problem, an ANN was optimized to model the chemistry of methane/air combustion for prediction of temperature and chemical source term. Regression analysis for the optimal ANN demonstrates satisfactory prediction of both quantities. It is interesting to note that the optimal ANNs for prediction of these two quantities are very different, which indicates that there are underlying discrepancies in the behavior of each of these two quantities. Network performance is compared with the conventional tabulation method. It is shown that the chemistry tabulation requires considerably more memory in order to obtain equivalent accuracy.

Acknowledgments

The authors gratefully acknowledge funding by the United States Department of Energy within the Advanced Simulation and Computing (ASC) program. Helpful discussions with Masoud Aryanpour and Andrew M. Bradley are gratefully acknowledged. The authors also wish to thank Mark Abramson and John E. Dennis, Jr. for guidance on the optimization methodology.

REFERENCES

Abramson, M. 2004 Mixed variable optimization of a load-bearing thermal insulation system using a filter pattern search algorithm. *Optimization and Engineering* **5** (2), 157–177.

Angeline, P., Saunders, G. & Pollack, J. 1994 Evolutionary algorithm that constructs recurrent neural networks. *IEEE Trans. Neural Networks* **5**, 54–65.

Audet, C. & Dennis, J. 2000 Pattern search algorithms for mixed variable programming. *SIAM Journal on Optimization* **11** (3), 573–594.

Blasco, J., Fueyo, N., Dopazo, C. & Ballester, J. 1999*a* Modelling the temporal evolution of a reduced combustion chemical system with an artificial neural network. *Combust. Flame* **113** (1-2), 38–52.

Blasco, J., Fueyo, N., Dopazo, C. & Chen, J. 2000 A self-organizing-map approach to chemistry representation in combustion applications. *Combust. Theory Modelling* **4** (1), 61–76.

Blasco, J., Fueyo, N., Larroya, J., Dopazo, C. & Chen, Y. 1999*b* Single-step time-integrator of a methane-air chemical system using artificial neural networks. *Computers and Chemical Engineering* **23** (9), 1127–1133.

Booker, A., Dennis, J., Frank, P., Serafini, D., Torczon, V. & Trosset, M. 1999 A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization* **17** (1), 1–13.

Bornholdt, S. & Graudenz, D. 1992 General asymmetric neural networks and structure design by genetic algorithms. *Neural Networks* **5** (2), 327–334.

Bowman, C., Hanson, R., Davidson, D., Gardiner, W., Lissianski, V., Smith, G., Golden, D., Frenklach, M. & Goldenberg, M. 1997 GRI-Mech 2.11. Available from `http://www.me.berkeley.edu/gri-mech/`.

Chen, J., Blasco, J., Fueyo, N. & Dopazo, C. 2000 An economical strategy for

storage of chemical kinetics: Fitting *in situ* adaptive tabulation with artificial neural networks. *Proc. Comb. Institute* **28**, 115–121.

Christo, F., Masri, A. & Nebot, E. 1996*a* Artificial neural network implementation of chemistry with PDF simulation of H_2/CO_2 flames. *Combust. Flame* **106** (4), 406–427.

Christo, F., Masri, A., Nebot, E. & Pope, S. 1996*b* An integrated PDF/neural network approach for simulating turbulent reacting systems. *Proc. Comb. Institute* **26**, 43–48.

Flemming, F., Sadiki, A. & Janicka, J. 2005 LES using artificial neural networks for chemistry representation. *Progress in Computational Fluid Dynamics* **5** (7), 375–385.

Hagan, M., Demuth, H. & Beale, M. 1996 *Neural Network Design*. Boston, Mass., USA: PWS Publishing Company.

Haykin, S. 1994 *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, N.J.: Prentice Hall.

Husken, M., Jin, Y. & Sendhoff, B. 2005 Structure optimization of neural networks for evolutionary design optimization. *Soft Computing* **9** (1), 21–28.

Hwang, M., Choi, J. & Park, J. 1997 Evolutionary projection neural networks. In *Proc. IEEE Conference on Evolutionary Computation (ICEC '97)*, pp. 667–671.

Koehler, J. & Owen, A. 1996 Computer experiments. In *Handbook of Statistics* (ed. S. Ghosh & C. Rao), pp. 261–308. New York: Elsevier Science.

Kokkolaras, M., Audet, C. & Dennis, Jr., J. 2000 Mixed variable optimization of the number and composition of heat intercepts in a thermal insulation system. *Tech. Rep.* TR00-21. Department of Computational & Applied Mathematics, Rice University, Houston.

Koza, J. & Rice, J. 1991 Genetic generation of both the weights and architecture for a neural network. In *Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN'91 Seattle)*, pp. 397–404.

Liu, Y. & Yao, X. 1996 Evolutionary design of artificial neural networks with different nodes. In *Proc. IEEE Conference on Evolutionary Computation (ICEC '96)*, pp. 670–675.

Lophaven, S., Nielsen, H. & Søndergaard, J. 2002 DACE: A MATLAB Kriging toolbox version 2.0. *Tech. Rep.* IMM-TR-2002-12. Technical University of Denmark, Copenhagen.

Marsden, A., Wang, M., Dennis, Jr., J. & Moin, P. 2004 Optimal aeroacoustic shape design using the surrogate management framework. *Optimization and Engineering* **5** (2), 235–262.

McKay, M., Beckman, R. & Conover, W. 1979 Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21** (2), 239–245.

Miller, G., Todd, P. & Hegde, S. 1989 Designing neural networks using genetic algorithms. In *Proc. 3rd Int. Conf. Genetic Algorithms and Their Applications* (ed. J. Schaffer), pp. 379–384.

Peters, N. 1984 Laminar diffusion flamelet models in non-premixed turbulent combustion. *Prog. Energy Combust. Sci.* **10** (3), 319–339.

Peters, N. 2000 *Turbulent Combustion*. Cambridge: Cambridge University Press.

Rubinstein, R. 1981 *Simulation and the Monte Carlo method*. John Wiley & Sons, Inc.

Serafini, D. 1998 A framework for managing models in nonlinear optimization of computationally expensive functions. PhD thesis, Rice University, Houston, Texas.

Tang, K., Chan, C., Man, K. & Kwong, S. 1995 Genetic structure for NN topology and weights optimization. In *Proc. 1st IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA'95)*, pp. 250–255.

Torczon, V. 1997 On the convergence of pattern search algorithms. *SIAM Journal on Optimization* **7** (1), 1–25.

Yao, X. 1999 Evolving artificial neural networks. *Proc. IEEE* **87** (9), 1423–1447.