Neural Network Based Response Surface Methods – a Comparative Study

Wolfram Beyer, Martin Liebscher, Michael Beer, Wolfgang Graf
TU Dresden, Germany

Abstract:

This paper deals with the application of the response surface method based on neural networks in the field of engineering. A multi-layer feedforward neural network is employed to replace the underlying structural analysis. It is trained on a data set from initial computations. The application of the backpropagation algorithm for training a neural network requires a decision for a particular training mode. For the examined examples it is shown that the incremental mode possesses different advantages compared to the batch mode. Furthermore, an approach to improve the approximation quality given by a neural network is introduced and demonstrated by means of a numerical example.

The authors would like to thank Dyna*more* GmbH, Stuttgart, Germany for supporting this study.

Keywords:

response surface method, neural networks, backpropagation

1 Introduction

The structural analysis of complex structures is usually solved by means of a computational model. The computational model is an abstraction of the structure and its environment including their interactions. Such a model is used to map structural parameters (e.g. material parameters, geometrical parameters, loads) onto structural responses (e.g. stresses, displacements, deformations). The complexity of the model increases the more information it has to contain. In many fields of engineering the Finite Element method is employed to perform tasks of arbitrary complexity. Structural analysis represents the basis for the structural design in a variety of approaches.

The values of the structural parameters are generally not predetermined but vary within specific ranges. This phenomenon is referred to as uncertainty and is accounted for with the aid of an uncertain structural analysis. For processing uncertain quantities of different characteristics associated numerical procedures are available. These primarily comprise stochastic analysis, fuzzy analysis, and fuzzy stochastic analysis. In any case, an uncertain structural analysis requires the repeated application of the underlying deterministic computational model. The more complex the model is the higher is the computational effort needed to perform the computation. This effort may be reduced by simplifying the computational model and reducing the number of calls of the deterministic computational model. In this context the use of an adequate approximation of the structural response represents the most effective measure to achieve a reasonable reduction of the numerical effort. Subsequently, an approximation by means of the response surface method based on neural networks is elucidated.

2 Neural Network Concept

2.1 Basic idea

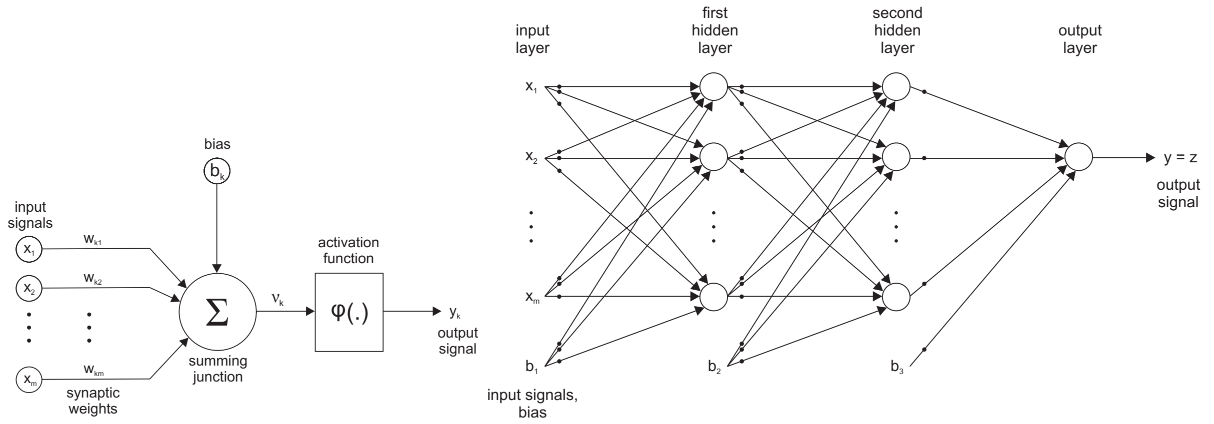
The idea of artificial neural networks is based on the design of the human brain. The human brain is constituted by information-processing units (so called neurons) that are connected by synapses, and it forms the kernel of the human nervous system. It is capable of processing input signals that are derived from the environment and of providing appropriate output signals (e.g. certain actions). The advantages of the human information processing system are complexity, nonlinearity, and parallelism. An artificial neural network resembles the human brain in many respects. It is constituted by neurons which are connected by synapses, it has the ability of mapping input signals onto output signals and to adapt to certain tasks during a training phase. The output produced by a neural network is called response surface. The idea of such a response surface in engineering is then to replace the deterministic computational model for structural analysis by a neural network. That is, the input signals comprise structural parameters such as loads, material parameters, and geometrical parameters and the network output provides the associated response surface in the form of stresses, displacements, or deformations. For this purpose, the network first needs to learn the features of the underlying deterministic computational model. This learning is based on initially performed structural analyses.

2.2 Network Architecture

There exist a variety of alternatives to design a neural network, e.g., feedforward neural networks, recurrent neural networks, radial basis function networks and committee machines. The focus of this study is set on feedforward neural networks which are already applied successfully in many fields of engineering. A specific network architecture is built by combining the two main components of a neural network, the neurons and the synapses. Each synapse connects two neurons with each other and possesses a synaptic weight w. It enables the signal flow from one neuron to the next one. A neuron represents an information processing unit that maps an input signal onto an output signal as illustrated in Fig. 1(a). It contains a summing junction v which lumps together the incoming signals x, each weighted by a specific synaptic weight w. The summation of input signals for neuron k (see Eq. (1)) also involves an external term called bias b.

$$v_k = \sum_{j=1}^{m} w_{kj} x_j + b_k \tag{1}$$

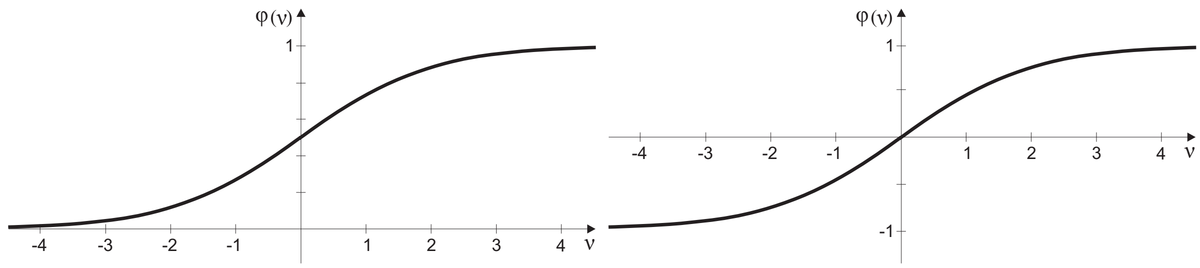
The weights w and the bias b allow the neuron to be adjusted to particular conditions. The summing junction v is used as input argument for the subsequently called activation function $\varphi(.)$, which produces the output y of the neuron. Different types of activation functions are available, such as threshold function, piecewise linear function and sigmoidal functions. It is often desired to have a differentiable activation function (see Sec. 3). In this study further consideration is only given to sigmoidal functions (see Fig. 2) as the most popular type of activation functions.



(a) Constitution of a neuron (b) Constitution of a feedforward neural network

Figure 1: Neural network

A multi-layer feedforward neural network (see Fig. 1(b)) permits signal flow exclusively in forward direction through the network. The neurons are organized in different layers: one input layer, one or more hidden layers, and one output layer. The produced output y of each neuron of a layer is transmitted to one or more neurons of the following layer by synaptic connections. Within a fully connected network every neuron is linked to all neurons of the following layer. Otherwise, if several connections are missing, the network is referred to as partially connected. Furthermore, it is possible to introduce shortcut connections between neurons of non-adjacent layers.



(a) Logistic sigmoid function (b) Sigmoid symmetric function

Figure 2: Sigmoidal activation functions

The output signals y of the neurons of the output layer constitute the response z of the whole neural network. The number of input and output signals determines the number of the neurons of the input and the output layer, respectively. The number of hidden layers and the dimensionality of these should be chosen problem-specific.

$$x \xrightarrow{neural\,network} y = z \tag{2}$$

Synaptic weights w are adjustable free parameters of a neural network and have the task to strengthen or weaken the signals transferred by the synaptic connections. These values are problem-specific adjusted during the training phase (see Sec. 3) of the neural network. The knowledge represented by a neural network after the training is stored in its synaptic weights. Usually the training phase is followed by a validation or testing phase in which the synaptic weights of the network are not changed anymore. The approximation-quality may be evaluated by calculating an error between the prediction of the neural network and the desired output.

3 Comparison of Training Algorithms

3.1 General Aspects of Backpropagation

The iterative process of adjustment of weights of a neural network is called training. The aim is to derive a neural network response that approximates the underlying data set with a maximum quality. There exist a variety of different methods to train a neural network. In this study the focus is on the backpropagation algorithm as one form of supervised error-correction learning. This algorithm requires training data in form of input-output pairs that are obtained by repeatedly applying the computational model for structural analysis. The backpropagation algorithm may be coarsely described with the following three steps, which have to be applied several times in an iteration.

1. Forward computation of input signal of training sample and determination of neural network response
2. Computation of an error between desired response and neural network response
3. Backward computation of the error and calculation of corrections to synaptic weights and biases

The initial values of the synaptic weights are obtained using random numbers. After the computation of the neural network response to an input signal the response error is determined and used to compute adequate changes of the synaptic weights with the aim of improving the quality of the neural network response in the next step. By applying these corrections to the weights it is attempted to minimize the error surface. Backpropagation is based on a standard gradient method. For the computation of the gradient the activation function needs to be differentiable. The correction of the synaptic weights may be expressed by the following Eq. (3) and is discussed in detail in [3, 1]. Within the backpropagation algorithm two different modes can be applied, which are called incremental and batch.

$$\begin{pmatrix} \text{weight} \\ \text{correction} \end{pmatrix} = \begin{pmatrix} \text{learning-rate} \\ \text{parameter} \end{pmatrix} * \begin{pmatrix} \text{local} \\ \text{gradient} \end{pmatrix} * \begin{pmatrix} \text{input} \\ \text{signal} \end{pmatrix} \tag{3}$$

Incremental Mode The incremental mode, also known as single, on-line or sequential mode, applies a weight correction after each presentation of one sample of training data. The training sample is chosen randomly out of the training data, leading to a stochastic nature of the search for the minimum of the error surface. This maximizes the probability of finding the global minimum. Furthermore, the incremental mode is resistant against redundant training data. As the weight correction is applied after each sample, the exact error of the specific sample is used.

Batch Mode The batch mode applies a weight correction only once after each epoch. During one epoch every sample of the training data is presented to the neural network. The weight correction according to each training sample needs to be stored and the weights are updated after the presentation of the whole training data using all stored weight changes. The training samples are not chosen randomly which assures a convergence to a (local) minimum. Further, redundant training data is disadvantageous, because it yields longer computation time.

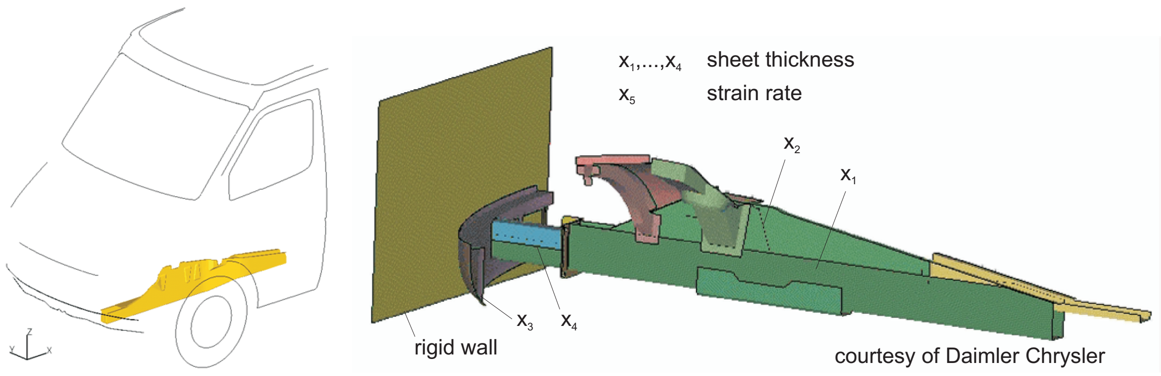
3.2 Numerical Comparison by Examples

For the comparison of the incremental mode and the batch mode calculations are performed under identical conditions. That means, the size of the used network, the training and testing data as well as the number of trained epochs are the same for the application of both modes. The training behavior varies in dependence on the randomly selected initial values for the synaptic weights and biases and on the training mode – incremental or batch. The evaluation of training results is performed by means of the root mean square error e_{rms} (see Eq. (4)) of the approximation. In Eq. (4) N is the number of training samples, d_i is the desired exact output of one training sample i, and y_i is the neural network output generated from training sample i.

$$e_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - y_i)^2} \tag{4}$$

During a testing phase the trained neural network is shown a data set containing samples not used during the training phase. The synaptic weights are not adjusted during this phase. The testing phase and the computation of a testing error gives information about generalization qualities of the neural network.

Example 1 The first example is taken from automobile industry (see Fig. 3(a)).



(a) Position of chassis beam in vehicle (b) Parameters of chassis beam

Figure 3: Example 1 – chassis beam [7]

The displayed chassis beam is an important component for the crash behavior of the vehicle. With the aid of a numerical structural analysis it is aimed at finding an appropriate design for this beam. The input parameters $x_i,\ i = 1,...,5$ which need to be adjusted in the design process are depicted in Fig. 3(b). The response which is approximated by means of a neural network is the stonewall force affecting the rigid wall. The development of the error, which is plotted over the number of epochs, is shown in Fig. 5(a).

Example 2 The second example deals with the planned bridge of Messina in Italy (see Fig. 4) which carries two railways as well as two highways.

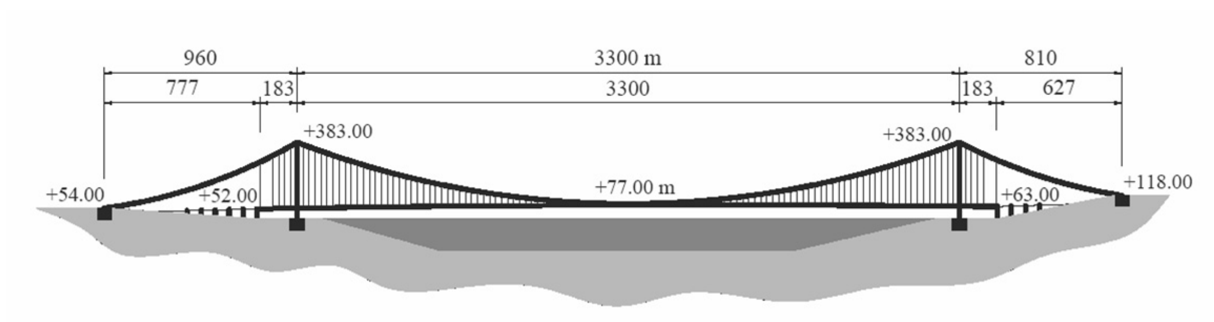
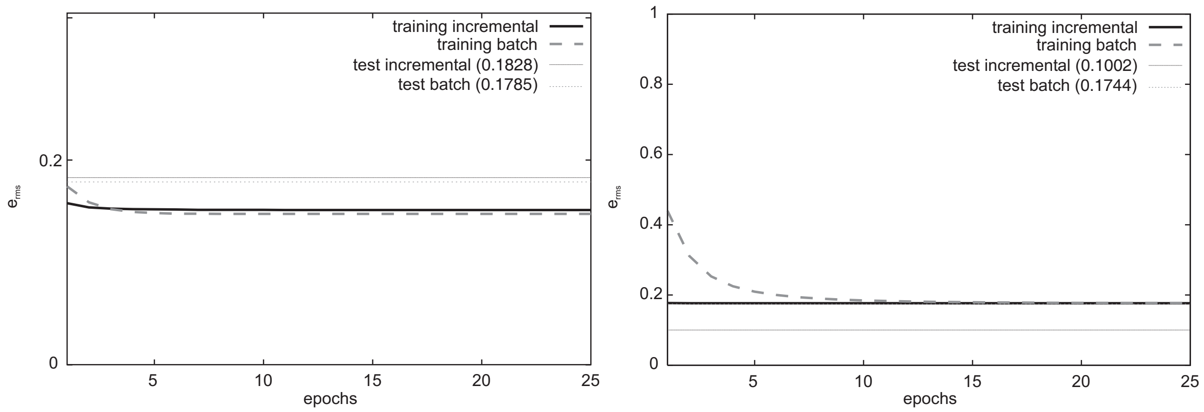


Figure 4: Example 2 – bridge of Messina [4, 5]

The region around the bridge is subject to seismic activity. Therefore, the bridge behavior under seismic loads needs to be examined. The input parameters are seismic loads at 11 different points. The structural response comprises displacements at 28 points of the bridge. As an example only one of these displacements is chosen to be approximated by means of a neural network. Training and testing results are shown in Fig. 5(b).



(a) Example 1 (b) Example 2

Figure 5: Comparison of the incremental mode and the batch mode

Interpretation For both examined examples it is obvious that both modes – incremental and batch – yield a similar remaining error of the approximation of the neural network with respect to the exact response. The incremental mode of backpropagation training, which updates the synaptic weights multiple times during one epoch, achieves a convergence within fewer epochs than the batch mode. For this reason the incremental mode needs less computation time for training and provides a better approximation quality after few epochs.

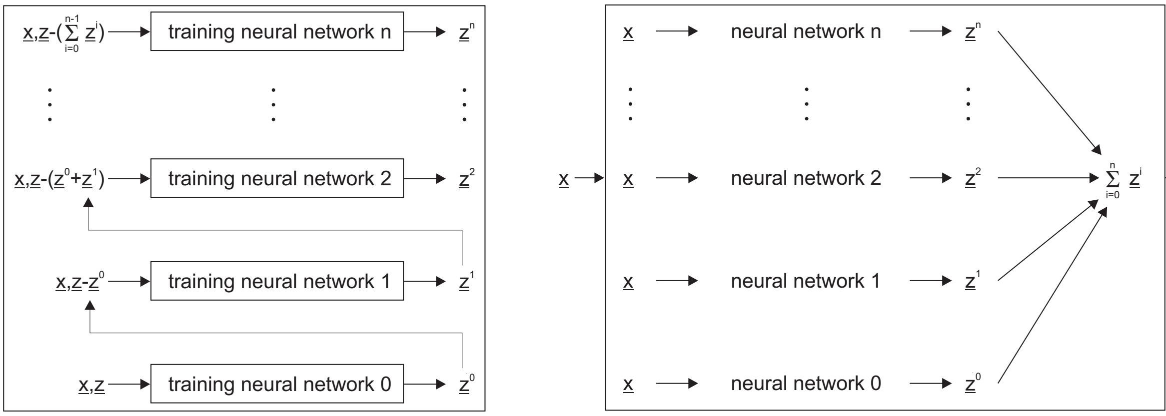
4 Improvement of Approximation Accuracy

In many cases the quality of a response surface based on a neural network – even after a fair amount of training epochs – still raises the desire for further improvement. Different approaches to improve the approximation quality are, e.g., an increase of complexity of the neural network by adding more hidden layers and / or hidden neurons and the use of more than one neural network for the response approximation. Multiple neural networks are used by committee machines (as implemented in LS-Opt), by section-wise approximations (e.g. using a clustering algorithm for subdividing), and by neural network composites. Subsequently, the specific use of a neural network composite is explained and examined in detail.

4.1 Selected Strategy

The term "composite" in connection with neural networks used in this study is associated with a perspective different from the definition given in [2]. The proposed approach is visualized in Fig. 6.

Herein, a first neural network is trained to approximate a certain response. The underlying task is very complex. Thus the neural network response is only capable of approximating certain features of the desired response. The idea is now to use the computed remaining error surface to train a second neural network. The overall response is then constituted by adding the responses of both networks. For further improvement of the response surface more neural networks may be added to approximate the remaining errors in this manner, respectively.

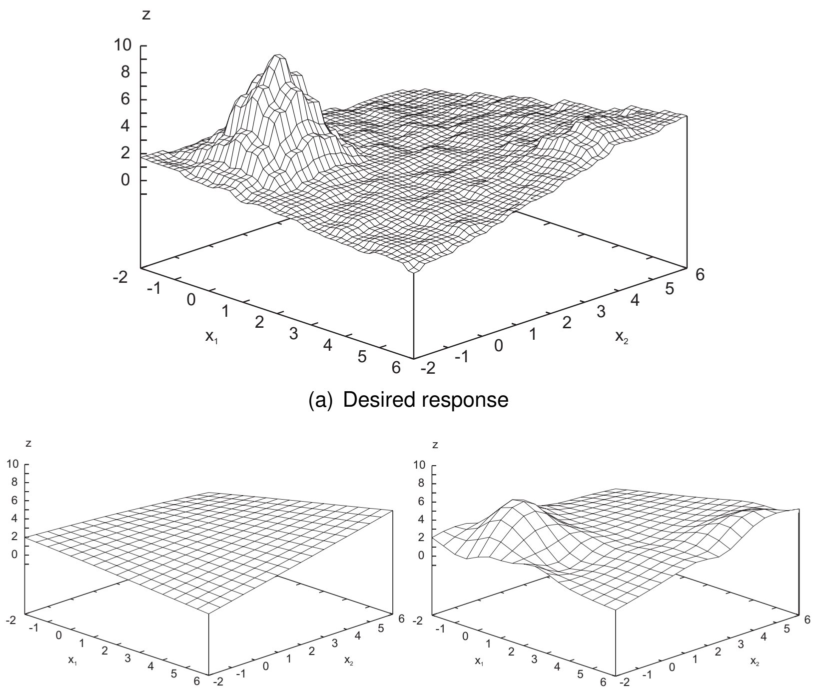


(a) Training (b) Evaluation

Figure 6: Neural network composite

4.2 Numerical Example

The use of a neural network composite is demonstrated by means of a numerical example. The response shown in Fig. 7(a) needs to be approximated.

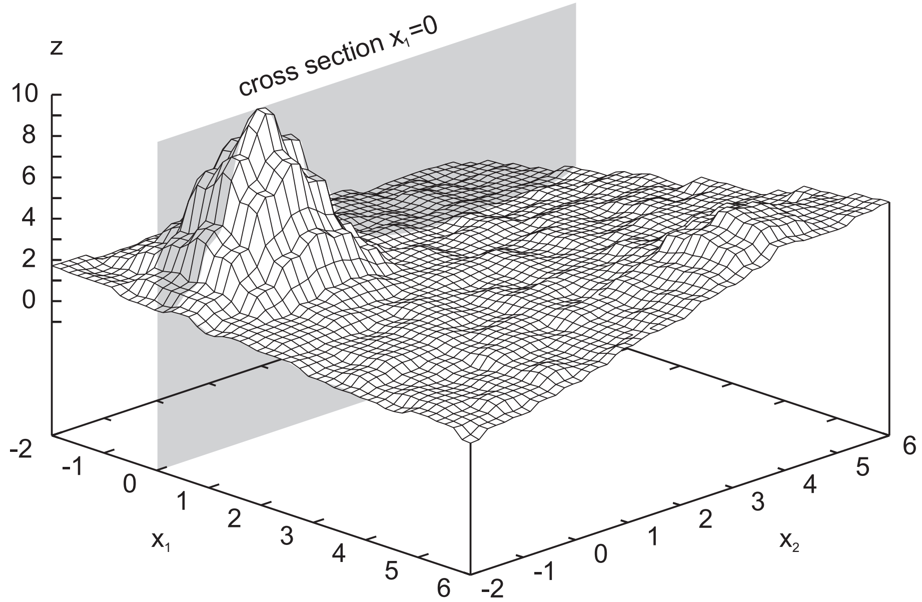


(a) Desired response

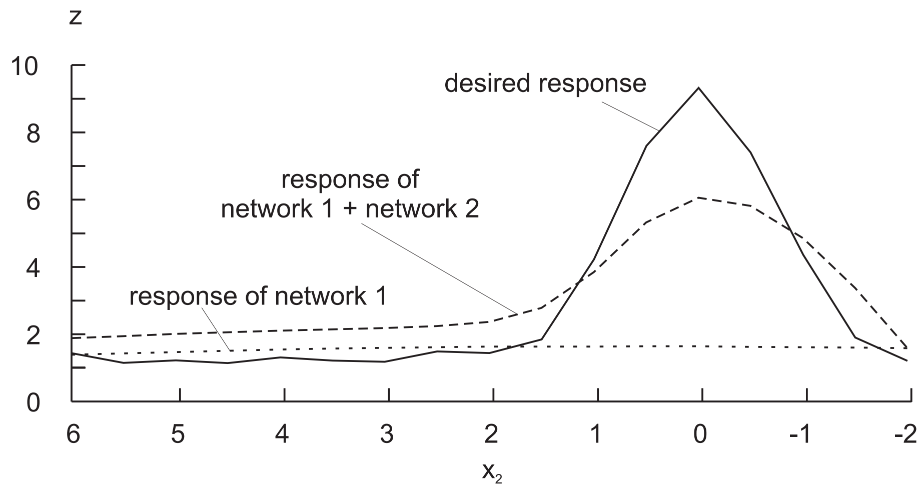
(b) Response of network 1 (c) Response of network 1 + 2

Figure 7: Responses

The response of the first trained network is plotted in Fig. 7(b). It is visible that the first neural network is only capable of approximating the global trend of the desired response. Local features such as extreme values are not reflected correctly. The second neural network is trained to approximate the error surface. The sum of the responses from both networks achieves a better approximation quality by giving respect to local features of the response (see Fig. 7(c)). The described procedure may be repeated to add more neural networks to eliminate the remaining errors. The quality improvement by adding the second network becomes particularly obvious in the two-dimensional plot at cross section $x_1 = 0$ in Fig. 8.



(a) Position of cross section $x_1 = 0$



(b) Cross section $x_1 = 0$

Figure 8: Comparison of responses

5 Conclusions

This study discusses two different issues of response surface approximation based on neural networks. First, the differences between the incremental mode and the batch mode of the backpropagation algorithm applied for training neural networks are investigated. For the examined examples it is found that the incremental mode possesses advantages such as faster convergence and a less computational effort compared to the batch mode. Consequently, an application of the incremental mode is proposed in approaches on this basis such as structural design and optimization. For example, an improvement of the successive response surface method (see [6]) used in LS-Opt may be achieved.

The second issue is the quality improvement of the response approximation by using more than one neural network. Different from the committee machine used by LS-Opt in this study a neural network composite is proposed. Such a network composite applies a stepwise reduction of the approximation error by using the error surface for network training. In the chosen example it is shown that a network composite is able to approximate several features of the desired response with a remarkable quality.

References

[1] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., 1997.

[2] G. Deco and D. Obradovic. *An Information Theoretic Approach to Neural Computing*. Springer, 1996.

[3] S. Haykin. *Neural Networks*. Tom Robbins / Prentice-Hall International, Inc., 1999.

[4] L. Scambi and F. Bontempi. A genetic algorithm approach for performance evaluation of long span suspension bridges. In *Proceedings of the Tenth International Conference on Civil, Structural and Environmental Engineering Computing*. Civil-Comp Press, Stirling, Scotland, 2005.

[5] L. Scambi and F. Bontempi. A fuzzy approach in the seismic analysis of long span suspension bridges. In *13th World Conference on Earthquake Engineering*, 2004.

[6] N. Stander, W. Roux, T. Eggleston, and K. Craig. *LS-Opt Manual*. Livermore Software Technology Corporation, March 2006.

[7] M. Thiele, M. Liebscher, and W. Graf. Fuzzy analysis as alternative to stochastic methods - a comparison by means of a crash analysis. In *4th German LS-DYNA Forum*, 2005.

