Optimal Training Sequences for Locally Recurrent Neural Networks

Krzysztof Patan and Maciej Patan

Institute of Control and Computation Engineering,
University of Zielona Góra
{k.patan,m.patan}@issi.uz.zgora.pl

Abstract. The problem of determining an optimal training schedule for a locally recurrent neural network is discussed. Specifically, the proper choice of the most informative measurement data guaranteeing the reliable prediction of the neural network response is considered. Based on a scalar measure of the performance defined on the Fisher information matrix related to the network parameters, the problem was formulated in terms of optimal experimental design. Then, its solution can be readily achieved via the adaptation of effective numerical algorithms based on the convex optimization theory. Finally, some illustrative experiments are provided to verify the presented approach.

1 Introduction

A training of neural network, being the dynamic data-driven process requires a proper selection of measurement data to provide satisfactory representation of the modelled system behaviour [1,2]. In practice, this is equivalent to determination of a limited number of observational units obtained from the experimental environment in such a way as to obtain the best quality of the system responses.

The importance of input data selection has already been recognized in many application domains [3]. One of the most stimulating practical examples is Fault Detection and Identification (FDI) of industrial systems [4]. A crucial issue among the fundamental tasks of failure protection systems is to provide reliable diagnosis of the expected system state. To produce such a forecast, however, an accurate model is necessary and its calibration requires parameter estimation. Preparation of experimental conditions in order to gather informative measurements can be very expensive or even impossible (e.g. for the faulty system states). On the other hand, the data form real-world system may be very noisy and using all the available data may lead to significant systematic modelling errors. In result, we are faced with the problem of optimal choice of the available training data in order to obtain the most accurate model.

Although it is well known that the training quality for neural networks heavily depends on the choice of input sequences, surprisingly, there have been relatively few contributions to experimental design for those systems [5,6] and, in addition, they focus mainly on the multi-layer perceptron class of networks. The applicability of such a static type of networks for the modelling of dynamic systems

C. Alippi et al. (Eds.): ICANN 2009, Part I, LNCS 5768, pp. 80–89, 2009.
© Springer-Verlag Berlin Heidelberg 2009

is rather limited. Recently, the problem of optimal selection of input sequences in the context of dynamic neural networks has been discussed by the authors in [7,4], where the problem is formulated in spirit of optimum experimental design theory for lumped systems [8]. However, the simulation results presented therein concern the training of the single dynamic neuron only. The contribution of this work is to extend this approach to the locally recurrent neural network with one hidden layer which can be applied in real-world systems. Moreover, to illustrate the delineated approach some experiments are performed using real process data.

2 Dynamic Neural Networks

The topology of the neural network considered is analogous to that of the multilayered feedforward one and the dynamics are reproduced by the so-called dynamic neuron models [9,10,4]. The state-space representation of the neuron is shown in Fig. 1. In this paper a discrete-time dynamic network with n time varying inputs and m outputs is discussed. The description of such kind of a dynamic network with v hidden dynamic neurons, each containing an r-th order IIR filter, is given by the following nonlinear system:

$$\begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{W}\boldsymbol{u}(k) \\ \boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{\sigma}(\boldsymbol{B}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k) - \boldsymbol{g})^T \end{cases}, \tag{1}$$

where $N = v \times r$ represents the number of model states, $\boldsymbol{x} \in \mathbb{R}^N$ is the state vector, $\boldsymbol{u} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$ are input and output vectors, respectively, $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is the block diagonal state matrix ($\mathrm{diag}(\boldsymbol{A}) = [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_v]$), $\boldsymbol{W} \in \mathbb{R}^{N \times n}$ ($\boldsymbol{W} = [\boldsymbol{w}_1 \boldsymbol{1}^T, \ldots, \boldsymbol{w}_v \boldsymbol{1}^T]^T$, where $\boldsymbol{w}_i$ is the input weight vector of the i-th hidden neuron), and $\boldsymbol{C} \in \mathbb{R}^{m \times v}$ are the input and output matrices, respectively, $\boldsymbol{B} \in \mathbb{R}^{v \times N}$ is a block diagonal matrix of feedforward filter parameters ($\mathrm{diag}(\boldsymbol{B}) = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_v]$), $\boldsymbol{D} \in \mathbb{R}^{v \times n}$ is the transfer matrix ($\boldsymbol{D} = [b_{0_1} \boldsymbol{w}_1^T, \ldots b_{0_v} \boldsymbol{w}_v^T]^T$), $\boldsymbol{g} = [g_1 \ldots g_v]^T$ denotes the vector of biases, and $\boldsymbol{\sigma} : \mathbb{R}^v \to \mathbb{R}^v$ is the nonlinear vector-valued function. The presented structure can be viewed as a network with a single hidden layer containing v dynamic neurons as processing elements and an output layer with linear static elements. For structural details of the network considered, the interested reader is referred to [4,11].

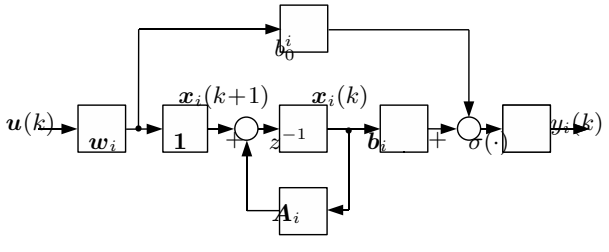

Fig. 1. State-space form of the i-th neuron with IIR filter

3 Optimal Sequence Selection Problem

3.1 Statistical Model

Let $\boldsymbol{y}^j = \boldsymbol{y}(\boldsymbol{u}^j;\boldsymbol{\theta}) = \{\boldsymbol{y}(k;\boldsymbol{\theta})\}_{k=0}^{L_j}$ denote the sequence of network responses for the sequence of inputs $\boldsymbol{u}^j = \{\boldsymbol{u}(k)\}_{k=0}^{L_j}$ related to the consecutive time instants $k = 0,\ldots,L_j < \infty$ and selected from among an *a priori* given set of input sequences $\mathcal{U}_d = \{\boldsymbol{u}^1,\ldots,\boldsymbol{u}^P\}$, where $\mathcal{U}_d \subset \mathcal{U}$. Here $\boldsymbol{\theta}$ represents a p-dimensional unknown network parameter vector which must be estimated using observations of the system (i.e. filter parameters, weights, slope and bias coefficients).

From the statistical point of view, the sequences of observations related to P input sequences may be considered as

$$\boldsymbol{z}^j(k) = \boldsymbol{y}^j(k;\boldsymbol{\theta}) + \boldsymbol{\varepsilon}^j(k), \quad k = 0,\ldots,L_j, \quad j = 1,\ldots,P, \tag{2}$$

where $\boldsymbol{z}^j(k)$ is the output and $\boldsymbol{\varepsilon}^j(k)$ denotes the measurement noise. It is customary to assume that the measurement noise is zero-mean, Gaussian and white, i.e.

$$E[\boldsymbol{\varepsilon}^i(k)\boldsymbol{\varepsilon}^j(k')] = v^2\delta_{ij}\delta_{kk'}, \tag{3}$$

where $v > 0$ is the standard deviation of the measurement noise, δ_{ij} and $\delta_{kk'}$ standing for the Kronecker delta functions.

An additional substantial assumption is that the training of the neural network, equivalent to the estimation of the unknown parameter vector $\boldsymbol{\theta}$, is performed via the minimization of the least-squares criterion

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\Theta_{\mathrm{ad}}} \sum_{j=1}^{P}\sum_{k=0}^{L_j} \|\boldsymbol{z}^j(k) - \boldsymbol{y}^j(k;\boldsymbol{\theta})\|^2, \tag{4}$$

where Θ_{ad} is the set of admissible parameters. It becomes clear that since $\boldsymbol{y}^j(k;\boldsymbol{\theta})$ strongly depends on the input sequences $\boldsymbol{u}^j$ it is possible to improve the training process through appropriate selection of input sequences.

3.2 Sequence Quality Measure and Experimental Design

In order to properly choose the input sequences which will be most informative for the training of the dynamic network, a quantitative measure of the goodness of parameter identification is required. A reasonable approach is to choose a performance measure defined on the Fisher Information Matrix (FIM), which is commonly used in optimum experimental design theory [12,8,13].

Sequences which guarantee the best accuracy of the least-squares estimates of $\boldsymbol{\theta}$ are then found by choosing $\boldsymbol{u}^j$, $j = 1,\ldots,P$ so as to minimize some scalar measure of performance Ψ defined on the *average Fisher information matrix* given by [14]:

$$\boldsymbol{M} = \frac{1}{PL_j}\sum_{j=1}^{P}\sum_{k=0}^{L_j} \boldsymbol{H}(\boldsymbol{u}^j,k)\boldsymbol{H}^T(\boldsymbol{u}^j,k), \tag{5}$$

where

$$\boldsymbol{H}(\boldsymbol{u},k)=\left(\frac{\partial \boldsymbol{y}(\boldsymbol{u},k;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)_{\boldsymbol{\theta}=\boldsymbol{\theta}^0} \tag{6}$$

stands for the so-called *sensitivity matrix*, $\boldsymbol{\theta}^0$ being a prior estimate to the unknown parameter vector $\boldsymbol{\theta}$ which can be obtained from previous experiments or alternatively some known nominal values can be used [15,16,14,17].

Such a formulation is generally accepted in optimum experimental design for nonlinear dynamic systems, since the inverse of the FIM constitutes, up to a constant multiplier, the Cramér-Rao lower bound on the covariance matrix of any unbiased estimator of $\boldsymbol{\theta}$ [13], i.e.

$$\operatorname{cov}\hat{\boldsymbol{\theta}} \succeq \boldsymbol{M}^{-1}. \tag{7}$$

Under somewhat mild assumptions [16,17], it is legitimate to assume that our estimator is *efficient* in the sense that the parameter covariance matrix achieves the lower bound.

As for criterion Ψ, various choices are proposed in the literature [13,8], but the most popular choice is so-called D-optimality (determinant) criterion:

$$\Psi(\boldsymbol{M}) = -\log\det \boldsymbol{M}; \tag{8}$$

which minimizes the volume of the uncertainty ellipsoid for the parameter estimates. The introduction of an optimality criterion renders it possible to formulate the sensor location problem as an optimization problem:

$$\Psi\big[\boldsymbol{M}(\boldsymbol{u}^1,\ldots,\boldsymbol{u}^P)\big] \longrightarrow \min \tag{9}$$

with respect to $\boldsymbol{u}^j$, $j=1,\ldots,P$ belonging to the admissible set $\mathcal{U}$.

The direct consequence of the assumption (3) is that we admit replicated input sequences, i.e. some $\boldsymbol{u}^j$s may appear several times in the optimal solution (because independent observations guarantee that every replication provides additional information). Consequently, it is sensible to reformulate the problem so as to operate only on the distinct sequences $\boldsymbol{u}^1,\ldots,\boldsymbol{u}^S$ instead of $\boldsymbol{u}^1,\ldots,\boldsymbol{u}^P$ by relabelling them suitably. To this end, we introduce $r_1,\ldots,r_S$ as the numbers of replicated measurements corresponding to the sequences $\boldsymbol{u}^1,\ldots,\boldsymbol{u}^S$. In this formulation, the $\boldsymbol{u}^i$s are said to be the *design* or *support* points, and $p_1,\ldots,p_S$ are called their weights. The collection of variables

$$\xi_P=\begin{Bmatrix}\boldsymbol{u}^1,\,\boldsymbol{u}^2,\,\ldots,\,\boldsymbol{u}^S\\ p_1,\;p_2,\;\ldots,\;p_S\end{Bmatrix}, \tag{10}$$

where $p_i = r_i/P$, $P=\sum_{i=1}^{S} r_i$, is called the *exact design* of the experiment. The proportion p_i of observations performed for $\boldsymbol{u}^i$ can be considered as the percentage of experimental effort spent at that sequence. Hence, we are able to rewrite the FIM in the form

$$\boldsymbol{M}(\xi_P)=\sum_{i=1}^{S} p_i \frac{1}{L_i}\sum_{k=0}^{L_i}\boldsymbol{H}^T(\boldsymbol{u}^i,k)\boldsymbol{H}(\boldsymbol{u}^i,k). \tag{11}$$

Here the p_is are rational numbers, since both r_is and P are integers. This leads to a discrete numerical analysis problem whose solution is difficult for standard optimization techniques, particularly when P is large. A potential remedy for this problem is to extend the definition of the design. This is achieved through the relaxation of constraints on weights, allowing the p_is to be considered as real numbers in the interval $[0, 1]$. This assumption will be also made in what follows. Obviously, we must have $\sum_{i=1}^{S} p_i = 1$, so we may think of the designs as probability distributions on $\mathcal{U}$. This leads to the so-called *continuous* designs, which constitute the basis of the modern theory of optimal experiments [8,13]. It turns out that such an approach drastically simplifies the design, and the existing rounding techniques [8] justify such an extension. Thus, we shall operate on designs of the form

$$\xi = \left\{ \begin{matrix} \boldsymbol{u}^1, \boldsymbol{u}^2, \ldots, \boldsymbol{u}^S; \\ p_1,\ p_2,\ \ldots,\ p_S \end{matrix} \quad \sum_{i=1}^{S} p_i = 1; \quad \forall i\ p_i \geqslant 0 \right\}, \tag{12}$$

which concentrates Pp_1 observational sequences for $\boldsymbol{u}^1$ (so we repeat approximately Pp_1 times the presentation of this sequence during the training of the network), Pp_2 for $\boldsymbol{u}_2$, and so on. Then we may redefine optimal design as a solution to the optimization problem

$$\xi^\star = \arg \min_{\xi \in \Xi(\mathcal{U}} \Psi[\boldsymbol{M}(\xi)], \tag{13}$$

where $\Xi(\mathcal{U})$ denotes the set of all probability distributions on $\mathcal{U}$.

3.3 Characterization of Optimal Solutions

In the remainder of this chapter we shall assume that $\boldsymbol{H} \in C(\mathcal{U}; \mathbb{R}^p)$. The following characterizations of the optimal design $\xi^\star$ can be derived in a rather straightforward manner from the general results given in [14] or [17].

Theorem 1. *An optimal design exists comprising no more than $p(p+1)/2$ support sequences. Moreover, the set of optimal designs is convex.*

The practical importance of this property cannot be underestimated since we can restrict our attention to the designs with limited number of sequences what significantly reduces the complexity of resulting optimization problem. But the next theorem is essential for the approach considered and provides a tool for checking the optimality of designs. It is usually called an *equivalence theorem* [18].

Theorem 2 (Equivalence theorem). *The following conditions are equivalent:*

(i) the design $\xi^\star$ minimizes $\Psi(M) = -\ln\det \boldsymbol{M}(\xi)$,
(ii) the design $\xi^\star$ minimizes $\max_{\boldsymbol{u}^i \in \mathcal{U}} \phi(\boldsymbol{u}^i, \xi)$, and
(iii) $\max_{\boldsymbol{u}^i \in \mathcal{U}} \phi(\boldsymbol{u}^i, \xi) = p$,

and the so-called *sensitivity function*

$$\phi(\boldsymbol{u}^i, \xi) = \operatorname{trace}\left(\frac{1}{L_i}\sum_{k=0}^{L_i} \boldsymbol{H}^T(\boldsymbol{u}^i, k)\boldsymbol{M}^{-1}\boldsymbol{H}(\boldsymbol{u}^i, k)\right)$$

is of paramount importance here as it can be interpreted in terms of average variance of the estimated system response being the natural measure for the quality of the training process. From the result above it comes immediately that suppressing the maximal level of the prediction variance is equivalent to the optimization of the D-optimality criterion. This paves the way to almost direct application of numerous efficient algorithms known from experimental design theory to the discussed problem. Since analytical determination of optimal designs is difficult or impossible even for very simple network structures, some iterative design procedures will be required. A simple computational scheme for that purpose is given in the next section.

4 Selection of Training Sequences

In the case considered in the paper, a computational algorithm can be derived based on the mapping $\mathcal{T} : \Xi(\mathcal{U}) \to \Xi(\mathcal{U})$ defined by

$$\mathcal{T}\xi = \begin{Bmatrix} \boldsymbol{u}^1, & \ldots, & \boldsymbol{u}^S \\ p_1\phi(\boldsymbol{u}^1, \xi)/p, & \ldots, & p_S\phi(\boldsymbol{u}^S, \xi)/p \end{Bmatrix}. \tag{14}$$

From Theorem 2 it follows that a design $\xi^\star$ is D-optimal if it is a fixed point of the mapping $\mathcal{T}$, i.e.

$$\mathcal{T}\xi^\star = \xi^\star. \tag{15}$$

Therefore, the following algorithm can be used as a generalization of that proposed in [19, p. 139] for the classical optimum experimental design problem consisting in iterative computation of a D-optimum design on a finite set:

Step 1. Guess a discrete starting design $\xi^{(0)}$ such that $p_i^{(0)} > 0$ for $i = 1, \ldots, S$. Choose some positive tolerance $\eta \ll 1$. Set $\ell = 0$.

Step 2. If the condition

$$\frac{\phi(\boldsymbol{u}^i, \xi^{(\ell)})}{p} < 1 + \eta, \quad i = 1, \ldots, S$$

is satisfied, then *STOP*.

Step 3. Construct the next design $\xi^{(k+1)}$ by determining its weights according to the rule

$$p_i^{(\ell+1)} = p_i^{(\ell)} \frac{\phi(\boldsymbol{u}^i, \xi^{(\ell)})}{m}, \quad i = 1, \ldots, S,$$

increment k by one and go to Step 2.

The convergence result of this scheme can be found in [17].

5 Illustrative Example

Simulation setting. All experiments were carried out using the AMIRA DR300 laboratory system. This laboratory system is used to control the rotational speed of a DC motor with a changing load [4]. A separately excited DC motor was modelled by using the dynamic neural network presented briefly in Section 2. The output signal was the rotational speed (T) measured by an analog tachometer. The input signal (U) was selected as a sum of sinusoids:

$$U(k) = 3\sin(2\pi 1.7k) + 3\sin(2\pi 1.1k - \pi/7) + 3\sin(2\pi 0.3k + \pi/3) \tag{16}$$

The structure of the neural network model (1) was selected arbitrarily and had the following structure: one input, three IIR neurons with second order filters and hyperbolic tangent activation functions, and one linear output neuron. Taking into account that a neural network is a redundant system, some of its parameters are not identifiable. In order to apply optimum experimental design to the neuron training, certain assumptions should be made. So, without loss of generality, let us assume that the feedforward filter parameter b_0 for each hidden neuron is fixed to the value of 1. This reduces the dimensionality of estimation and assures the identifiability of the rest of the parameters (i.e. it assures that the related FIM is non-singular).

At the beginning, the network was preliminarily trained in order to obtain the initial parameters estimates. Feeding the laboratory system with signal (16), a learning set containing 500 samples was formed, and then the training process was carried out off-line for 2000 steps using the Extended Dynamic Back-Propagation (EDBP) algorithm [7]. At the second stage of the training, the learning data were split into 20 time sequences, containing 150 consecutive samples each. The design purpose was to choose from this set of all learning patterns the most informative sequences (in the sense of D-optimality) and their presentation frequency (i.e. how often they should be repeated during the training). To determine the optimal design, a numerical routine from Section 4 was implemented in the form of the MATLAB program. All the admissible learning sequences taken with equal weights formed the initial design. The accuracy of the design algorithm was set to $\eta = 10^{-2}$.

Results. The network was preliminarily trained and the initial network parameters estimates are presented in the second column of Table 1. In this case the Sum of Squared Errors (SSE) calculated using the training set was equal to 5.7001. After that, the training of the network was continued in two ways. The first way was to use the optimal training sets selected during the optimum experimental design phase. The second way was to use random sequences as the training ones. The purpose of these experiments is to check the quality of parameter estimation. In the case considered here the optimal design consists of the sequences $5, 6, 8$ and 16. For a selected design, each distinct sequence was replicated proportionally to its weight in the design with total number of replications assumed to be $P = 10$. For example, if the optimal design consists of the four aforementioned

sequences with the weights 0.3, 0.1, 0.3 and 0.3, respectively, then during the training the 5-th, 8-th and 16-th sequences were used three times each, and the 6-th sequence only once (in random order). The training procedure was repeated 10 times using different measurement noise affecting the output of the system. The statistics are presented in Table 1. As we can see there, the application of training sets selected according to the optimal design leads to the better accuracies of parameter estimates than in the case of randomly selected training sets. It is observed that the standard deviation of each network parameter has lower value in the case of the optimal design what means the more reliable estimate of a given parameter.

The uncertainty of the network response prediction is examined based on the parameter estimates determined using the optimal and random designs. The testing phase of each of 10 realizations of locally recurrent network was performed using 1000 samples different from the training ones, and for each realization the quality measure in the form of the SSE was calculated. The results of testing are presented in Table 2. Looking at these results one can state that using random design it is possible to obtain a good generalization of the network, e.g. networks 9 and 10, but the results of training are not repetitive as in the case of optimal design when 9 of 10 training run give the similar good results. This fact, in connection with the plot of response prediction variance (Fig. 2) clearly shows

Table 1. Sample mean and the standard deviation of parameter estimates

para-	initial	sample mean		standard deviation	
meter	value	random design	optimal design	random design	optimal design
w_1	0.3232	0.2867	0.2894	0.0104	0.0028
w_2	0.9000	0.9105	0.9082	0.0034	0.0009
w_3	0.0758	0.0898	0.0789	0.0194	0.0027
b_{11}	0.8328	0.8187	0.8195	0.0040	0.0011
b_{21}	-0.6316	-0.6053	-0.6072	0.0078	0.0019
b_{31}	0.8558	0.8616	0.8581	0.0079	0.0011
b_{12}	0.7892	0.7742	0.7747	0.0042	0.0011
b_{22}	0.0631	0.0910	0.0897	0.0082	0.0019
b_{32}	0.5745	0.5808	0.5812	0.0076	0.0011
a_{11}	0.1258	0.1302	0.1301	0.0012	0.0003
a_{21}	0.0853	0.0807	0.0812	0.0015	0.0004
a_{31}	-0.4171	-0.4196	-0.4170	0.0055	0.0015
a_{12}	0.1656	0.1703	0.1703	0.0012	0.0003
a_{22}	0.0266	0.0217	0.0221	0.0016	0.0004
a_{32}	-0.5566	-0.5587	-0.5562	0.0052	0.0015
g_1	-0.3794	-0.4057	-0.4024	0.0132	0.0055
g_2	-0.3978	-0.3599	-0.3673	0.0206	0.0089
g_3	0.3187	0.3040	0.3136	0.0189	0.0008
c_1	-0.4908	-0.4905	-0.4893	0.0081	0.0032
c_2	0.7773	0.7708	0.7716	0.0078	0.0035
c_3	0.4540	0.4438	0.4408	0.0075	0.0006

Table 2. Results of network testing – SSE measure

Network realization	Random design	Optimal design
1	29.7367	31.0998
2	27.4287	26.5564
3	42.4463	26.4758
4	85.8052	26.6182
5	99.5833	26.4214
6	82.9475	26.4577
7	35.3615	26.6521
8	29.6130	26.5550
9	26.8438	26.5030
10	26.2403	26.2885

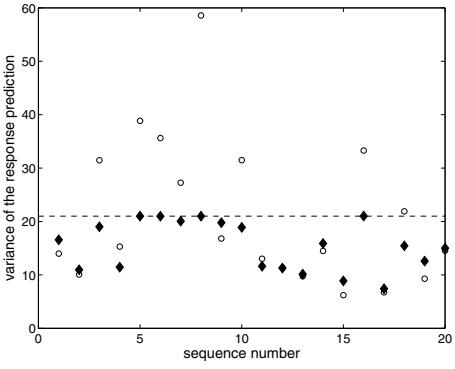

Fig. 2. Variances of the model response prediction for the optimum design (diamonds) and random design (circles)

that training based on optimal learning sequences leads to greater reliability of the network response as the maximal variance level can be significantly reduced.

Taking into account the computation burden, for the case considered here the selection of the optimal plan lasted 9 seconds. For comparison, the 500 training steps carried out off-line with the sequence of the length 500 lasted 117.18 seconds. Summarizing, the process of the optimal design selection by itself does not significantly prolongate the overall training procedure.

6 Conclusions

The results reported in this paper show that some well-known methods of optimum experimental design for linear regression models can be easily extended to the setting of the optimal training sequence selection problem for dynamic neural networks. The clear advantage of the proposed approach is that the quality of the training process measured in terms of the uncertainty of network response prediction can be significantly improved with the same effort spent on training or, alternatively, training process complexity can be reduced without degrading the network performance.

The proposed approach was also tested using other network structures. Experiments were carried out for a locally recurrent network with two hidden neurons as well as for a network with five hidden neurons. In each case considered, the results are similar to these presented in the paper taking into account the reliability of parameters estimates.

Future research will be focused on the application of other methods for determining optimal designs, namely methods which do not determine the presentation frequency, thus its practical application is easier.

References

1. Gupta, M.M., Jin, L., Homma, N.: Static and Dynamic Neural Networks. From Fundamentals to Advanced Theory. John Wiley & Sons, New Jersey (2003)
2. Korbicz, J., Kościelny, J., Kowalczuk, Z., Cholewa, W.: Fault Diagnosis. Models, Artificial Intelligence, Applications. Springer, Heidelberg (2004)
3. van de Wal, M., de Jager, B.: A review of methods for input/output selection. Automatica 37, 487–510 (2001)
4. Patan, K.: Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes. LNCIS. Springer, Berlin (2008)
5. Fukumizu, K.: Statistical active learning in multilayer perceptrons. IEEE Transactions on Neural Networks 11, 17–26 (2000)
6. Witczak, M.: Toward the training of feed-forward neural networks with the D-optimum input sequence. IEEE Transactions on Neural Networks 17, 357–373 (2006)
7. Patan, K., Patan, M.: Selection of training sequences for locally recurrent neural network training. In: Malinowski, K., Rutkowski, L. (eds.) Recent Advances in Control and Automation, pp. 252–262. Academic Publishing House, EXIT, Warsaw (2008)
8. Fedorov, V.V., Hackl, P.: Model-Oriented Design of Experiments. Lecture Notes in Statistics. Springer, New York (1997)
9. Tsoi, A.C., Back, A.D.: Locally recurrent globally feedforward networks: A critical review of architectures. IEEE Transactions on Neural Networks 5, 229–239 (1994)
10. Marcu, T., Mirea, L., Frank, P.M.: Development of dynamical neural networks with application to observer based fault detection and isolation. International Journal of Applied Mathematics and Computer Science 9(3), 547–570 (1999)
11. Patan, K.: Stability analysis and the stabilization of a class of discrete-time dynamic neural networks. IEEE Transactions on Neural Networks 18, 660–673 (2007)
12. Atkinson, A.C., Donev, A.N.: Optimum Experimental Designs. Clarendon Press, Oxford (1992)
13. Walter, E., Pronzato, L.: Identification of Parametric Models from Experimental Data. Springer, London (1997)
14. Patan, M.: Optimal Observation Strategies for Parameter Estimation of Distributed Systems. Lecture Notes in Control and Computer Science, vol. 5. Zielona Góra University Press, Zielona Góra (2004)
15. Uciński, D.: Optimal selection of measurement locations for parameter estimation in distributed processes. International Journal of Applied Mathematics and Computer Science 10(2), 357–379 (2000)
16. Rafajłowicz, E.: Optimum choice of moving sensor trajectories for distributed parameter system identification. International Journal of Control 43(5), 1441–1451 (1986)
17. Uciński, D.: Optimal Measurement Methods for Distributed Parameter System Identification. CRC Press, Boca Raton (2005)
18. Kiefer, J., Wolfowitz, J.: Optimum designs in regression problems. The Annals of Mathematical Statistics 30, 271–294 (1959)
19. Pázman, A.: Foundations of Optimum Experimental Design. Mathematics and Its Applications. D. Reidel Publishing Company, Dordrecht (1986)