ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

Multiple Local Models for System Identification Using Vector Quantization Algorithms

Luís Gustavo M. Souza and Guilherme A. Barreto

Federal University of Ceará, Department of Teleinformatics Engineering
Av. Mister Hull, S/N - Campus of Pici, CEP 60455-760, Fortaleza, Ceará, Brazil

Abstract. We introduce a novel method to build multiple local regression models based on the prototype vectors of the SOM network and other well-known vector quantization (VQ) algorithms. The resulting models are evaluated in the task of identifying the inverse dynamics of a heat exchanger data set. Additionally, we evaluate through statistical hypothesis testing the influence of the VQ algorithm on the performance of the local model. Simulation results demonstrate that the proposed method consistently outperforms previous MLP- and SOM-based approaches for system identification.

1 Introduction

Identification of nonlinear dynamic systems with neural networks was consistently evaluated by [1] who, under mild conditions, have shown that such NARX neural networks are able to solve difficult function approximation problems. Since then, neural-based system identification procedures have been dominated by standard supervised architectures, such as the Multilayer Perceptron (MLP) and the Radial Basis Functions (RBF) networks.

More recently, the self-organizing map (SOM) has emerged as a viable alternative to more traditional neural-based approaches to identification and control of nonlinear dynamical systems [2, 3, 4, 5, 6]. In this paper, multiple local models based on the SOM are developed and used to approximate the inverse dynamics of a heat exchanger data set. Local models have been a source of much interest because they have the ability to adhere to the local shape of an arbitrary surface, which is difficult especially in cases when the dynamical system characteristics vary considerably throughout the state space.

Several complex dynamical systems which can be described by the NARX model [7]: $y(t) = f[y(t-1), \ldots, y(t-p); u(t), u(t-1), \ldots, u(t-q+1)]$, where p and q are the memory orders of the dynamical model. This equation states that the system output y at time t depends on the past p output values and on the past q values of the input u. In many situations, it is also desirable to approximate the inverse mapping of a nonlinear plant: $u(t) = f^{-1}[\ u(t-1), \ldots, u(t-q); y(t-1), \ldots, y(t-p)]$, whose goal is to estimate the input of a given system based on previous values of the input and output variables.

The remainder of the paper is organized as follows. In Section 2 a novel method for the design of multiple local inverse NARX models based on the SOM is proposed. Computer simulations and discussions are presented in Section 3. The paper is concluded in Section 4.

ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

2 Local Models Based on Neighboring Prototype Vectors

The algorithm to be described is called *Multiple Models K-winners SOM* (M-KSOM), since it is based on the KSOM architecture. The KSOM itself is a local regression approach to solve system identification problems [8] that depends on the VQTAM model [6], which is an extension of the SOM algorithm for simultaneously performing vector quantization on the input and output spaces of a given nonlinear mapping.

In the VQTAM model, the input vector $\mathbf{x}(t)$ is composed of two parts. The first part, denoted $\mathbf{x}^{in}(t) \in \mathbb{R}^{p+q}$, carries data about the input of the dynamic mapping to be learned. The second part, denoted $x^{out}(t) \in \mathbb{R}$, contains data concerning the desired output of this mapping. The weight vector of neuron i, $\mathbf{w}_i(t)$, $i = 1, ..., N$, has its dimension increased accordingly. These changes are represented as

$$\mathbf{x}(t) = \begin{pmatrix} \mathbf{x}^{in}(t) \\ x^{out}(t) \end{pmatrix} \text{ and } \mathbf{w}_i(t) = \begin{pmatrix} \mathbf{w}_i^{in}(t) \\ w_i^{out}(t) \end{pmatrix}, \tag{1}$$

where $\mathbf{w}_i^{in}(t) \in \mathbb{R}^{p+q}$ and $w_i^{out}(t) \in \mathbb{R}$ are, respectively, the portions of the weight (prototype) vector which store information about the inputs and the outputs of the desired mapping.

Depending on the variables chosen to build the vector $\mathbf{x}^{in}(t)$ and scalar $x^{out}(t)$ one can use the SOM algorithm (or any other VQ algorithm) to learn the forward or the inverse mapping of a given dynamic system. For the inverse modeling task we are interested in, we define $\mathbf{x}^{in}(t) = [u(t-1), \ldots, u(t-q); y(t-1), \ldots, y(t-p)]^T$ and $x^{out}(t) = u(t)$.

The winning neuron i^* at time step t is determined based only on $\mathbf{x}^{in}(t)$:

$$i^*(t) = \arg\min_{\forall i}\{\|\mathbf{x}^{in}(t) - \mathbf{w}_i^{in}(t)\|\}. \tag{2}$$

For updating the weights, however, both $\mathbf{x}^{in}(t)$ and $x^{out}(t)$ are used:

$$\Delta\mathbf{w}_i^{in}(t) = \alpha(t)h(i^*, i; t)[\mathbf{x}^{in}(t) - \mathbf{w}_i^{in}(t)] \tag{3}$$

$$\Delta w_i^{out}(t) = \alpha(t)h(i^*, i; t)[x^{out}(t) - w_i^{out}(t)] \tag{4}$$

where $0 < \alpha(t) < 1$ is the learning rate, and $h(i^*, i; t)$ is a time-varying Gaussian neighborhood function. Both, $\alpha(t)$ and $h(i^*, i; t)$, should decay in time as in the usual SOM training.

The M-KSOM model building starts once VQTAM training is finished. For each neuron i in the VQTAM, we determine the K closest prototype vectors to the prototype vector $\mathbf{w}_i^{in}$:

$$\begin{aligned} j_1^{(i)} &= \arg\min_{\forall j \neq i}\{\|\mathbf{w}_i^{in} - \mathbf{w}_j^{in}\|\}; \\ &\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ j_K^{(i)} &= \arg\min_{\forall j \neq \{i, j_1, \ldots, j_{K-1}\}}\{\|\mathbf{w}_i^{in} - \mathbf{w}_j^{in}\|\}, \end{aligned} \tag{5}$$

where $\mathcal{J}_i = i \cup \{j_k^{(i)}\}_{k=1}^{K}$ is the set containing the indexes of the K prototype vectors which are closest to $\mathbf{w}_i^{in}$, including neuron i.

Once the set $\mathcal{J}_i$ is determined for each neuron i, we build N local regression models using the prototype vectors whose indexes belong to $\mathcal{J}_i$. Thus, associated to neuron i, we have a coefficient vector $\mathbf{a}_i \in \mathbb{R}^{p+q}$ computed using the least squares method:

$$\mathbf{a}_i = \left[\mathbf{R}_i^T \mathbf{R}_i + \lambda \mathbf{I}\right]^{-1} \mathbf{R}_i^T \mathbf{b}_i^{out}, \quad i = 1, ..., N \tag{6}$$

where $\mathbf{I}$ is a identity matrix of order K and $\lambda > 0$ (e.g. $\lambda = 0.001$) is a small regularization constant. The vector $\mathbf{b}_i^{out} \in \mathbb{R}^{K+1}$ is comprised of the output parts of the K prototype vectors whose indexes belong to $\mathcal{J}_i$:

$$\mathbf{b}_i^{out} = [w_i^{out} \;\; w_{j_1^{(i)}}^{out} \;\; \cdots \;\; w_{j_K^{(i)}}^{out}]^T, \tag{7}$$

while the matrix $\mathbf{R}_i$ is built as follows

$$\mathbf{R}_i = \begin{pmatrix} w_{i,1}^{in} & w_{i,2}^{in} & \cdots & w_{i,p+q}^{in} \\ w_{j_1^{(i)},1}^{in} & w_{j_1^{(i)},2}^{in} & \cdots & w_{j_1^{(i)},p+q}^{in} \\ \vdots & \vdots & \vdots & \vdots \\ w_{j_K^{(i)},1}^{in} & w_{j_K^{(i)},2}^{in} & \cdots & w_{j_K^{(i)},p+q}^{in} \end{pmatrix}_{(K+1)\times(p+q)} = \begin{pmatrix} \left(\mathbf{w}_i^{in}\right)^T \\ \left(\mathbf{w}_{j_1^{(i)}}^{in}\right)^T \\ \vdots \\ \left(\mathbf{w}_{j_K^{(i)}}^{in}\right)^T \end{pmatrix} \tag{8}$$

where the superscript T denotes the transpose vector/matrix.

Once the N local regression models are built, they can be used to approximate the output of the nonlinear mapping of interest. Note that we use one local model per input vector $\mathbf{x}(t)$. Which one to use at time t is defined by the index of the winning neuron, as shown in Eq. (2).

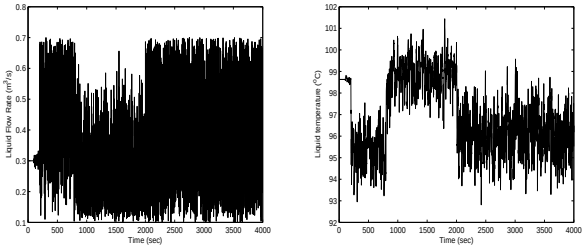
In this paper we use the M-KSOM method to approximate the inverse mapping of a nonlinear plant. Thus, the M-KSOM model estimate the current input $u(t)$ by means of the following equation:

$$\hat{u}(t) = \mathbf{a}_{i^*}^T(t)\mathbf{x}^{in}(t), \tag{9}$$

where the estimation error (residual) at time t is defined as $e(t) = u(t) - \hat{u}(t)$.

3 Computer Simulation and Discussion

All the models are evaluated via the statistics of the resulting normalized mean-squared error: $NMSE = \sum_{t=1}^{M} e^2(t)/M \cdot \hat{\sigma}_u^2$, where $\hat{\sigma}_u^2$ is the variance of the original time series $\{u(t)\}_{t=1}^{M}$ and M is the length of the sequence of residuals. Finally, hypotheses testing are carried out to analyze the influence of the VQ algorithm on the performance of the resulting local inverse model. The hypothesis testing is implemented through the Kolmogorov-Smirnov test on the estimation error distribution generated by a given model.



(a) (b)

Fig. 1: Measured values of (a) liquid flow rate and (b) outlet liquid temperature.

Table 1: Performance results for the heat exchanger data.

Neural Models	**NMSE**			
	mean	*min*	*max*	*variance*
MLP-1h	0.4292	0.4173	0.4501	4.0224e-005
M-KSOM	0.5640	0.4137	0.8118	0.0109
MLP-LM	0.5672	0.2140	0.6016	0.0048
KSOM	0.5841	0.4139	1.3877	0.0205
Linear	0.9257	0.9257	0.9257	5.2849e-014
MLP-2h	1.3003	1.2207	1.5466	0.0059

The heat exchanger data set comes from a liquid-satured steam heat exchanger [9], where water is heated by pressurized saturated steam through a copper tube. The motivation for the choice of the heat exchanger as a benchmark is that this plant is characterized by a non-minimum phase system. Figure 1 shows the measured values of the input time series ((a) $\{u(t)\}$, m^3/s) and the output time series ((b) $\{y(t)\}$, ^{o}C) and the sampling time is 1s.

Results on the NMSE criterion: All the models are trained using the first 3200 samples of the input/output time series (approx. 80% of the total) and tested with the remaining 800 samples. The best configuration found for the MLP-1h and MLP-LM models have 20 hidden neurons. For the MLP-2h, the number of hidden neurons in the second layer is heuristically set to half the number of neurons in the first hidden layer, respectively 10 and 20 neurons at each hidden layer. The MLP-based models were trained with constant learning rate equal to 0.1. All these nonlinear models are also compared with the linear inverse model, trained on-line through the plain LMS algorithm. The memory orders were set to $p = 6$ and $q = 3$, respectively.

For the KSOM and M-KSOM models, the number of neurons was set to $N = 30$. The initial and final learning rates are set to $\alpha_0 = 0.5$ and $\alpha_T = 0.001$, respectively. The neighborhood initial and final radii are $\sigma_0 = N/2$ and $\sigma_T = 0.001$, respectively. We found $K = 20$ by experimentation as the optimal value. The NMSE values were averaged over 100 training/testing runs, randomizing the initial weight values at each run. The obtained results are shown in Table 1.

Table 2: Performances of the M-KSOM model for different VQ algorithms.

M-KSOM Method				
VQ Algorithms	*NMSE*			
	mean	*min*	*max*	*variance*
SOM	0.5640	0.4137	0.8118	0.0109
WTA	0.6519	0.5620	0.7355	0.0016
K-means	0.8273	0.6594	0.8512	0.0008
FCL	1.2770	1.1522	1.4250	0.0037
FSCL	1.4013	1.1679	2.0755	0.0589

Table 3: KS-test on the M-KSOM performance for different VQ algorithms.

VQ Algorithm	**KS-test Results**
FCL	Reject Null Hypothesis
FSCL	Reject Null Hypothesis
WTA	Accept Null Hypothesis
K-means	Accept Null Hypothesis

The best performances were achieved by global models (MLP-1h and MLP-LM). However, the performance of the M-KSOM model is comparable to those resulting from MLP-1h and MLP-LM models.

The influence of the VQ algorithm on the performance of the M-KSOM approaches in the inverse system identification task are evaluated in Table 2. The M-KSOM method implemented using the following VQ algorithms: standard competitive learning (*winner-take-all*, WTA), K-means, frequency sensitive competitive learning (FSCL) and fuzzy competitive learning (FCL). By analyzing this table we can observe that the best local model was the one generated by the SOM algorithm, an indication that topology preservation is important for the proposed local regression approach.

Hypothesis Testing: The final experiment evaluates the degree of similarity among the sequence of residuals ($e(t) = u(t) - \hat{u}(t)$) generated by the M-KSOM method for different VQ algorithms. We use the Kolmogorov-Smirnov test (KS-test) [10] to measure the distance between the empirical cumulative distribution functions (CDF) of two sequences of residuals. The null hypothesis is that the sequences are drawn from the same distribution. If two M-KSOM models implemented using two different VQ algorithms generate statistically equivalent sequences of residuals (according to the KS-test), then the resulting local models are equivalent.

Tables 3 presents the results for the M-KSOM model. In this table, a rejection of the null hypothesis indicates that the CDF of the residuals generated by the original M-KSOM model[1] is different from the CDF of residuals generated by the M-KSOM implemented with a different VQ algorithm. The acceptance of the null hypothesis indicates that the CDF of the residuals generated by the original KSOM model is equivalent to the CDF of residuals generated by the KSOM implemented with a different VQ algorithm.

From Table 3 one can infer that the performance of the original M-KSOM

[1]The original M-KSOM model, as proposed in this paper, uses the SOM algorithm.

model is statistically equivalent to those obtained by implementing the M-KSOM with the WTA and K-means algorithms. This result is in accordance with the NMSE results shown in Table 2, since these M-KSOM models built using these alternative VQ algorithms generate the smallest values of NSME, in average. The original M-KSOM model produces the best NMSE results, however, if computational cost is an importante issue, the user can build the M-KSOM model using the WTA algorithm instead, with a slight degradation in performance.

4 Conclusion

In this paper we introduced a novel method to design local regression models using the prototype vectors of the Self-Organizing Map. The proposed approach, however, is general enough to be used with different vector quantization algorithms. We evaluate the performance of the proposed local regression approach for different VQ algorithms on the task of inverse system identification. All the resulting models were also analyzed statistically through the Kolmogorov-Smirnov test in order to study the influence of the VQ algorithms on the performances of the local models. The main general conclusion of the presented experiments is that the performances of VQ-based local inverse models are comparable to or better than those of global MLP-based models.

References

[1] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–27, 1990.

[2] J. Liu and D. Djurdjanovic. Topology preservation and cooperative learning in identification of multiple model systems. *IEEE Transactions on Neural Networks*, 19(2):2065–2072, 2008.

[3] J. Cho, J. Principe, D. Erdogmus, and M. Motter. Quasi-sliding mode control strategy based on multiple linear models. *Neurocomputing*, 70(4-6):962–974, 2007.

[4] G. A. Barreto and L. G. M. Souza. Adaptive filtering with the self-organizing maps: A performance comparison. *Neural Networks*, 19(6):785–798, 2006.

[5] J. Cho, J. Principe, D. Erdogmus, and M. Motter. Modeling and inverse controller design for an unmanned aerial vehicle based on the self-organizing map. *IEEE Transactions on Neural Networks*, 17(2):445–460, 2006.

[6] G. A. Barreto and A. F. R. Araújo. Identification and control of dynamical systems using the self-organizing map. *IEEE Transactions on Neural Networks*, 15(5):1244–1259, 2004.

[7] M. Norgaard, O. Ravn, N. K. Poulsen, and L. K. Hansen. *Neural Networks for Modelling and Control of Dynamic Systems*. Springer-Verlag, 2000.

[8] L. G. M. Souza and G. A. Barreto. Multiple local ARX modeling for system identification using the self-organizing map. In *Proceedings of II European Symposium on Time Series Prediction (ESTSP'08)*, pages 215–224, 2008.

[9] S. Bittanti and L. Piroddi. Nonlinear identification and control of a heat exchanger: A neural network approach. *Journal of the Franklin Institute*, 334(1):135–153, 1997.

[10] T. T. Soong. *Fundamentals of Probability and Statistics for Engineers*. John Wiley & Sons, West Sussex, England, 1st edition, 2004.