ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

Spline-based neuro-fuzzy Kolmogorov's network for time series prediction

Vitaliy Kolodyazhniy*

University of Basel - Institute for Psychology
Missionsstrasse 60/62, 4055 Basel - Switzerland

Abstract. A spline-based modification of the previously developed *Neuro-Fuzzy Kolmogorov's Network* (NFKN) is proposed. In order to improve the approximation accuracy, cubic B-splines are substituted for triangular membership functions. The network is trained with a hybrid learning rule combining least squares estimation for the output layer and gradient descent for the hidden layer. The initialization of the NFKN is deterministic and is based on the PCA procedure. The advantages of the modified NFKN are confirmed by long-range iterated predictions of two chaotic time series: an artificial data generated by the Mackey-Glass equation and a real data of laser intensity oscillations.

1 Introduction

According to the Kolmogorov's superposition theorem (KST) [1], any continuous function of d variables can be exactly represented by superposition of continuous functions of one variable and addition:

$$f(x_1,\ldots,x_d)=\sum_{l=1}^{2d+1} g_l\left[\sum_{i=1}^{d}\psi_{l,i}(x_i)\right],$$

where $x\in[x_1^{min},x_1^{max}]\times\ldots\times[x_d^{min},x_d^{max}]$, $g_l(\bullet)$ and $\psi_{l,i}(\bullet)$ are some continuous univariate functions, and $\psi_{l,i}(\bullet)$ are independent of f. Aside from exact representations [2], this theorem attracted the attention of many researchers as a basis for the construction of parsimonious universal approximators, e.g. in works [3–6]. In [4–6] it was shown that approximators in form of superposition of univariate functions and addition are especially advantageous for overcoming the curse of dimensionality. For the construction of the inner and outer functions in approximate Kolmogorov's representations, the authors of [5] and [6] used cubic splines due to their nice numerical properties.

A practical implementation of an interpretable KST-based universal approximator within a neuro-fuzzy modeling framework was proposed in [7]. This model called Fuzzy Kolmogorov's Network (FKN) has simple structure in form of a two-level fuzzy rule base according to the multi-resolution approach [4]. The training of the FKN is based on an alternating linear least squares technique.

In [8], a modification of the FKN called Neuro-Fuzzy Kolmogorov's Network (NFKN) was proposed. The NFKN is trained with a hybrid learning rule which is a

* This research was supported by the 6th Framework Project EUCLOCK (No. 018741).

combination of a gradient descent procedure and linear least squares method. Thus, the computational complexity of the training algorithm is reduced.

In this paper we consider how the approximation accuracy of the NFKN model can be further improved via the use of B-spline membership functions (MFs) [9] retaining at the same time the interpretability of the neuro-fuzzy model [7, 8].

2 Network Architecture

The NFKN architecture comprises two layers of neo-fuzzy neurons (NFNs) [10] and is described by the following equations:

$$\hat{f}(x_1,\ldots,x_d)=\sum_{l=1}^{n} f_l^{(2)}(o^{(1,l)}),\quad o^{(1,l)}=\sum_{i=1}^{d} f_i^{(1,l)}(x_i),\quad l=1,\ldots,n, \tag{1}$$

where n is the number of hidden layer neurons, $f_l^{(2)}(o^{(1,l)})$ is the l-th nonlinear synapse in the output layer, $o^{(1,l)}$ is the output of the l-th NFN in the hidden layer, $f_i^{(1,l)}(x_i)$ is the i-th nonlinear synapse of the l-th NFN in the hidden layer.

The nonlinear synapses are single input-single output fuzzy inference systems:

$$f_i^{(1,l)}(x_i)=\sum_{h=1}^{m_{1,i}}\mu_{i,h}^{(1)}(x_i)w_{i,h}^{(1,l)},\quad f_l^{(2)}(o^{(1,l)})=\sum_{j=1}^{m_{2,l}}\mu_{l,j}^{(2)}(o^{(1,l)})w_{l,j}^{(2)}, \tag{2}$$

$$l=1,\ldots,n,\quad i=1,\ldots,d,$$

where $m_{1,i}$ and $m_{2,l}$ is the number of MFs per synapse in the hidden and output layers respectively, $\mu_{i,h}^{(1)}(x_i)$ and $\mu_{l,j}^{(2)}(o^{(1,l)})$ are the MFs, $w_{i,h}^{(1,l)}$ and $w_{l,j}^{(2)}$ are tunable weights. Without loss of generality, we assume further that $m_{1,i}=m_1, i=1,\ldots,d$, and $m_{2,l}=m_2, l=1,\ldots,n$.

The description (1), (2) corresponds to the following two-stage fuzzy inference procedure

$$\hat{y}=\sum_{l=1}^{n}\sum_{j=1}^{m_2}\mu_{l,j}^{(2)}\left[\sum_{i=1}^{d}\sum_{h=1}^{m_1}\mu_{i,h}^{(1)}(x_i)w_{i,h}^{(1,l)}\right]w_{l,j}^{(2)} \tag{3}$$

and the following multi-resolution fuzzy rule base [7, 8]:

$$\text{IF } x_i \text{ IS } X_{i,h} \text{ THEN } o^{(1,1)}=w_{i,h}^{(1,1)}d \text{ AND}\ldots\text{AND } o^{(1,n)}=w_{i,h}^{(1,n)}d,$$

$$i=1,\ldots,d,\quad h=1,\ldots,m_1,$$

$$\text{IF } o^{(1,l)} \text{ IS } O_{l,j} \text{ THEN } \hat{y}=w_{l,j}^{(2)}n,\quad l=1,\ldots,n,\quad j=1,\ldots,m_2,$$

where $\hat{y}$ is the output of the neuro-fuzzy network, and $X_{i,h}$ and $O_{l,j}$ are the linguistic terms defined by MFs in the hidden and output layers, respectively. In NFKN, the MFs are triangular [7, 8]. In this paper, we generalize the NFKN via the use of B-spline basis functions for the MFs [9].

Given a sequence of ordered knots $\{c_1,\ldots,c_m\}$, the p-th B-spline basis function of order q is defined as

$$N_{p,q}(x)=\begin{cases}\begin{cases}1, & \text{for } c_p \le x < c_{p+1},\\ 0, & \text{otherwise,}\end{cases} & \text{if } q=1,\\ \dfrac{x-c_p}{c_{p+q-1}-c_p}N_{p,q-1}(x)+\dfrac{c_{p+q}-x}{c_{p+q}-c_{p+1}}N_{p+1,q-1}(x), & \text{if } q>1,\end{cases} \quad (4)$$

$$p=1,\ldots,m-q\,,\quad m\ge q\,.$$

Remarkably, one obtains triangular MFs letting $q=2$ in (4). Below we will consider only B-splines for $q=2$ and $q=4$, the latter case being cubic functions. For practical implementation, also quadratic B-splines for $q=3$ might be of interest.

For a spline-based NFKN given by (1)–(3), $m=m_1+q$ for the hidden layer, $m=m_2+q$ for the output layer, $\mu_{i,h}^{(1)}(x_i)=N_{h,q}(x_i)$, and $\mu_{l,j}^{(2)}(o^{(1,l)})=N_{j,q}(o^{(1,l)})$.

To maintain partition of unity throughout the universe of discourse of x, additional marginal functions should are added at both ends of the universe of discourse of x in (4) for $q>2$ (see Fig. 1) [9]. To further guarantee the partition of unity also for inputs outside the universe of discourse of x, we set the leftmost and rightmost knots to very large negative and positive values, respectively.

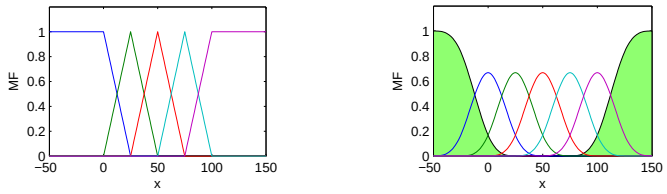

Fig. 1: B-spline membership functions of order 2 (*left*) and 4 (*right*) defined for variable $x\in[0,100]$ such that 5 membership function are defined over the universe of discourse of x. Shaded areas (*right*) correspond to marginal B-functions

As in [7, 8], we assume that the MFs are fixed and equidistantly spaced over the range of each NFN input. The parameters of the MFs (spline knots) are not tuned. The MFs in the NFKN at each input in the hidden layer are shared between all neurons.

3 Learning Algorithm

The weights of the NFKN are determined by means of a batch-training algorithm [8] briefly outlined below. The minimized error function is

$$E(t)=\frac{1}{2}\sum_{k=1}^{K}\left[y(k)-\hat{y}(t,k)\right]^2=\frac{1}{2}\left[Y-\hat{Y}(t)\right]^T\left[Y-\hat{Y}(t)\right], \quad (5)$$

where $Y=\left[y(1),\ldots,y(K)\right]^T$ is the vector of target values, and $\hat{Y}(t)=\left[\hat{y}(t,1),\ldots,\hat{y}(t,K)\right]^T$ is the vector of network outputs at epoch *t*, and *K* is the number of data points in the training data set.

Since the nonlinear synapses (2) are linear in parameters, the vector of the output layer weights $W^{(2)} = \left[w_{1,1}^{(2)}, w_{1,2}^{(2)}, \ldots, w_{n,m_2}^{(2)}\right]^T$ can be estimated with direct linear least squares (LS) optimization. We find a regularized LS solution as

$$W^{(2)}(t) = \left({\Phi^{(2)}}^T(t)\Phi^{(2)}(t) + \eta(t) I \right)^{-1} {\Phi^{(2)}}^T(t) Y^{(2)}(t), \tag{6}$$

$$\Phi^{(2)} = \left[\varphi^{(2)}(o^{(1)}(1)), \ldots, \varphi^{(2)}(o^{(1)}(M))\right]^T,$$

$$\varphi^{(2)}(o^{(1)}) = \left[\mu_{1,1}^{(2)}(o^{(1,1)}), \mu_{1,2}^{(2)}(o^{(1,1)}), \ldots, \mu_{n,m_2}^{(2)}(o^{(1,n)})\right]^T,$$

where $\eta(t)$ is an adjustable regularization term used to prevent rank deficiency.

The hidden layer weights $W^{(1)} = \left[w_{1,1}^{(1,1)}, w_{1,2}^{(1,1)}, \ldots, w_{d,m_1}^{(1,1)}, \ldots, w_{d,m_1}^{(1,n)}\right]^T$ are tuned using the following gradient descent-based rule:

$$W^{(1)}(t+1) = W^{(1)}(t) - \gamma \frac{\nabla_{W^{(1)}} E(t)}{\left\| \nabla_{W^{(1)}} E(t) \right\|}, \tag{7}$$

$$\nabla_{W^{(1)}} E(t) = -{\Phi^{(1)}}^T \left[Y - \hat{Y}(t) \right], \quad \Phi^{(1)} = \left[\varphi^{(1)}(x(1)), \ldots, \varphi^{(1)}(x(M))\right]^T,$$

$$\varphi^{(1)}(x) = \left[\varphi_{1,1}^{(1,1)}(x_1), \varphi_{1,2}^{(1,1)}(x_1), \ldots, \varphi_{d,m_1}^{(1,1)}(x_d), \ldots, \varphi_{d,m_1}^{(1,n)}(x_d)\right]^T,$$

$$\varphi_{i,h}^{(1,l)}(x_i) = \frac{df_l^{(2)}}{do^{(1,l)}} \mu_{i,h}^{(1,l)}(x_i), \tag{8}$$

where $0 < \gamma < 1$ is the learning rate constant, and normalization of the gradient in (7) is used to speed up the convergence. Derivatives $\frac{df_l^{(2)}}{do^{(1,l)}}$ in (8) can be computed noting that $\frac{dN_{p,q}(x)}{dx} = \frac{q-1}{c_{p+q-1} - c_p} N_{p,q-1}(x) + \frac{q-1}{c_{p+q} - c_{p+1}} N_{p+1,q-1}(x)$.

Each epoch of training starts with the optimization of the output layer weights according to (6), then the hidden layer weights are updated according to (7). In our implementation the training is stopped when either a pre-defined number of epochs are reached or the value of the error function (5) could not be improved for 50 subsequent iterations.

Since the NFKN does not have any bias weights and the outputs of neurons are linear w.r.t. their weights, the initialization of hidden layer weights $W^{(1)}$ can be performed deterministically via principal component analysis (PCA) as proposed in [11]. The hidden layer neurons are assigned weights from the first n loadings determined by PCA from the degrees of membership in the hidden layer computed on the training data set.

4 Experiments

In our experiments we tested the performance of models with membership functions of order $q = 2$ (triangular MFs, 'NFKN') and $q = 4$ (cubic B-spline MFs, 'SNFKN')

in iterated predictions of the Mackey-Glass (MG) time series [12] generated for $\tau = 17$, and the laser time series from Santa-Fe time series competition [13].

For the MG time series, we trained networks with 17 delayed inputs using 3000 data points from 118 to 3117 to predict the time series one step ahead. After training, the networks were used for iterated predictions of the next 500 points without 'knowing' the actual data of the time series for these 500 points. Networks with varying architectures for $m_1 = 4\ldots7$, $m_2 = 5\ldots9$, and $n = 1\ldots6$ were tried. The best SNFKN model provided a three times more accurate prediction than the best NFKN (see Fig. 2 and Table 1, where NMSE stands for normalized mean squared error).

Model	m_1	m_2	n	Parameters	Epochs	NMSE
SNFKN	5	7	6	552	149	0.0025
NFKN	5	9	6	564	189	0.0076

Table 1: Best results for MG time series prediction

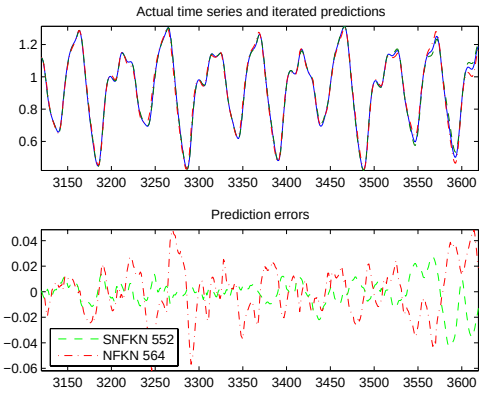

Fig. 2: Mackey-Glass time series (solid line in the upper plot), iterated predictions and errors (dashed line for SNFKN, dash-dotted line for NFKN)

For the laser time series, networks with 30 delayed inputs were trained using 970 data points from 31 to 1000, and the iterative predictions were performed for the next 100 points from 1001 to 1100. Networks with $m_1 = 4\ldots10$, $m_2 = 5\ldots13$, and $n = 1\ldots8$ were tried. As can be seen from Table 2 and Fig. 3, the best SNFKN model was almost 4 times more accurate than the best NFKN.

Model	m_1	m_2	n	Parameters	Epochs	NMSE	Rank
SNFKN	5	11	7	1127	465	0.0320	1
SNFKN	4	12	5	660	307	0.0767	6
NFKN	5	9	4	636	174	0.1194	1

Table 2: Best results for laser time series prediction

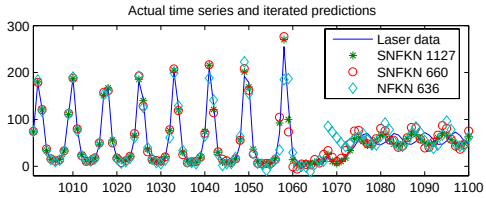

Fig. 3: Laser time series and iterated predictions

The best result in Table 2 with $NMSE = 0.032$ and 1127 parameters is very close to that reported in [13] for the winner of the Santa-Fe time series competition with $NMSE = 0.027$ and 1105 adjustable weights. However, the best forecasting model in [13] was based on a specialized multilayer architecture with FIR synapses and much more sophisticated and computationally intensive training algorithm.

References

[1] A. N. Kolmogorov, On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition, *Dokl. Akad. Nauk SSSR,* 114:953-956, 1957.

[2] R. Hecht-Nielsen, Kolmogorov's mapping neural network existence theorem. *Proceedings of the IEEE International Conference on Neural Networks,* San Diego, CA, Vol. 3, pages 11-14, 1987.

[3] V. Kůrková, Kolmogorov's theorem is relevant, *Neural Computation,* 3:617-622, 1991.

[4] Y. Yam, H. T. Nguyen and V. Kreinovich, Multi-resolution techniques in the rules-based intelligent control systems: a universal approximation result. *Proceedings of the 14th IEEE International Symposium on Intelligent Control/Intelligent Systems and Semiotics* (ISIC/ISAS'99), Cambridge, Massachusetts, September 15-17, pages 213-218, 1999.

[5] B. Igelnik, N. Parikh, Kolmogorov's spline network, *IEEE Transactions on Neural Networks,* 14:725-733, 2003.

[6] M. Coppejans, On Kolmogorov's representation of functions of several variables by functions of one variable, *Journal of Econometrics,* 123:1-31, 2004.

[7] V. Kolodyazhniy and Ye. Bodyanskiy, Fuzzy Kolmogorov's Network. In M.G. Negoita et al., editors, Lecture Notes in Computer Science 3214, pages 764-771, Springer-Verlag, 2004.

[8] Ye. Bodyanskiy, Ye. Gorshkov, V. Kolodyazhniy, and V. Poyedyntseva, Neuro-Fuzzy Kolmogorov's Network. In W. Duch et al., editors, Lecture Notes in Computer Science 3697, pages 1-6, Springer-Verlag, 2005.

[9] J. Zhang and A. Knoll, Constructing Fuzzy Controllers with B-Spline Models— Principles and Applications, *International Journal of Intelligent Systems,* 13: 257-285, 1998.

[10] T. Yamakawa, E. Uchino, T. Miki and H. Kusanagi, A neo fuzzy neuron and its applications to system identification and prediction of the system behavior. *Proceedings of the 2nd International Conference on Fuzzy Logic and Neural Networks* (IIZUKA-92), Iizuka, Japan, pages 477-483, 1992.

[11] V. Kolodyazhniy, F. Klawonn and K. Tschumitschew, A Neuro-Fuzzy Model for Dimensionality Reduction and Its Application, *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems*, 15:571-593, 2007.

[12] M. C. Mackey and L. Glass, Oscillation and chaos in physiological control systems, *Science,* 197: 287-289, 1977.

[13] A. S. Weigend and N. A. Gershenfeld, Results of the time series prediction competition at the Santa Fe Institute, *Proceedings of the IEEE International Conference on Neural Networks*, San Francisco, CA, 28 March-1 April 1993, Vol. 3, pages 1786-1793, 1993.