Representation of Neural Networks on Tensor-Product Basis

András Rövid[1], László Szeidl[2], Péter Várlaki[2]

[1]Óbuda University, John von Neumann Faculty of Informatics,
Bécsi út 96/b, 1034 Budapest Hungary, rovid.andras@nik.uni-obuda.hu
[2]Széchenyi István University System Theory Laboratory,
Egyetem tér 1, 9026 Győr Hungary, szeidl@sze.hu; varlaki@sze.hu

***Abstract**: The paper introduces a tensor-product-based alternative to approximate neural network models based on locally identified ones. The proposed approach may be a useful tool for solving many kind of black-box like identification problems. The main idea is based upon the approximation of a parameter varying system by locally identified neural network (NN) models in the parameter space on tensor-product form basis. The weights in the corresponding layers of the input local models are jointly expressed in tensor-product form such a way ensuring the efficient approximation. First the theoretical background of the higher order singular value decomposition and the tensor-product representation are introduced followed by the description of how this form can be applied for NN model approximation.*

***Keywords**: approximation; tensor product; neural network; system identification*

1 Introduction

Numerous methods have been proposed to deal with multi input, multi output systems, by the literature. As it is well known most real-life systems are to some extent nonlinear. There exist several types of nonlinear models, i.e. black box models, block structured models, neural networks, fuzzy models, etc. [12]. Approaches connecting the analytic and heuristic concepts may further improve their effectiveness and further extend their applicability. Linear parameter varying (LPV) structure is one by which non-linear systems can be controlled on the basis of linear control theories. As another frequently used approach to approximate dynamic systems the Takagi-Sugeno fuzzy modelling can be mentioned. This interest relies on the fact that dynamic T-S models are easily obtained by linearization of the nonlinear plant around different operating points [8]. Beyond these non-linear modelling techniques, the neural network-based approaches are highly welcome, as well, having the ability to learn sophisticated non-linear relationships [9][13]. Tensor product (TP) transformation is a numerical approach, which makes a connection between linear parameter varying models and higher order tensors ([5],[4]). The approach is strongly related to the generalized SVD the so called higher order singular value decomposition (HOSVD) [10], [11]. One of the most prominent property of the tensor product form is its complexity reduction and filtering support [6][7]. The proposed approach introduces a concept of how the joint representation of neural networks in tensor-product form can be performed and how this concept supports the efficient approximation of parameter varying systems on HOSVD basis via local neural nets in the parameter space. The paper is organized as follows: Section 2 gives a closer view on how to express a multidimensional function using polylinear functions on HOSVD basis, and how to reconstruct these polylinear functions, Section 3 shows how Neural Networks as local models can be expressed via HOSVD and finally future work and conclusions are reported.

2 Theoretical Background

Let us consider an *n*-variable smooth function

$f(\mathbf{x}), \mathbf{x} = (x_1, ..., x_N)^T, x_n \in [a_n, b_n], 1 \le n \le N,$

then we can approximate the function $f(\mathbf{x})$ with a series

$$f(\mathbf{x}) = \sum_{k_1=1}^{I_1} ... \sum_{k_N=1}^{I_N} \alpha_{k_1,...,k_n} p_{1,k_1}(x_1) \cdot ... \cdot p_{N,k_N}(x_N). \qquad (1)$$

where the system of orthonormal functions $p_{n,k_n}(x_n)$ can be chosen in classical way by orthonormal polynomials or trigonometric functions in separate variables and the numbers of functions I_n playing role in (1) are large enough. With the help of Higher Order Singular Value Decomposition (HOSVD) the approximation can be performed by a specially determined system of orthonormal functions depending on function $f(\mathbf{x})$. Assume that the function $f(\mathbf{x})$ can be given with some functions $\tilde{w}_{n,i}(x_n), x_n \in [a_n, b_n]$ in the form

$$f(\mathbf{x}) = \sum_{k_1=1}^{I_1} ... \sum_{k_N=1}^{I_N} \alpha_{k_1,...,k_n} \tilde{w}_{1,k_1}(x_1) \cdot ... \cdot \tilde{w}_{N,k_N}(x_N). \qquad (2)$$

Denote by $\mathcal{A} \in \mathbb{R}^{I_1 \times ... \times I_N}$ the *N*-dimensional tensor determined by the elements $\alpha_{i_1,...,i_N}, 1 \le i_n \le I_n, 1 \le n \le N$ and let us use the following notations (see [1]).

- $\mathcal{A} \boxtimes_n \mathbf{U}$: the n-mode tensor-matrix product,
- $\mathcal{A} \boxtimes_{n=1}^{N} \mathbf{U}_n$: the multiple product as $\mathcal{A} \boxtimes_1 \mathbf{U}_1 \boxtimes_2 \mathbf{U}_2 ... \boxtimes_N \mathbf{U}_N$.

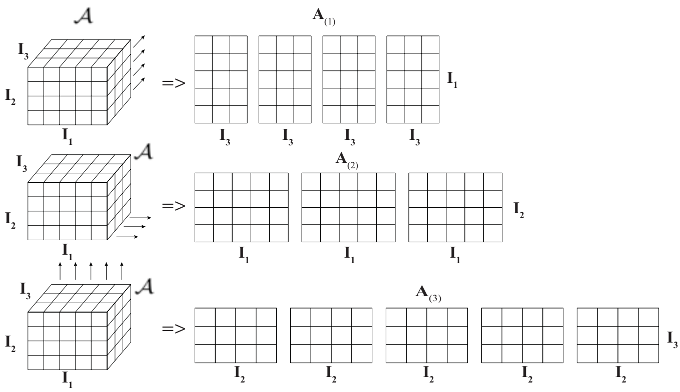

Figure 1
The three possible ways of expansions of a 3-dimensional array into matrices

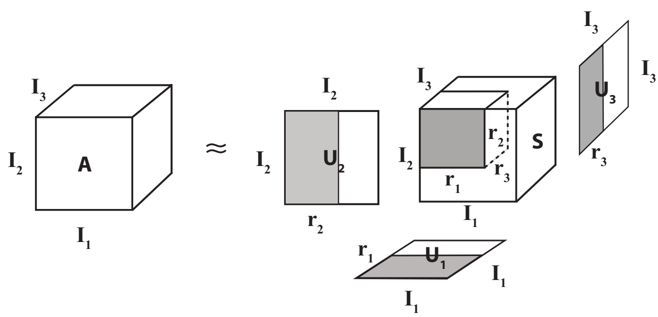

Figure 2
Illustration of the higher order singular value decomposition for a 3-dimensional array. Here S is the core tensor, the U_l -s are the l -mode singular matrices

The n-mode tensor-matrix product is defined by the following way. Let $\mathbf{U}$ be an $K_n \times M_n$ -matrix, then $\mathcal{A} \boxtimes_n \mathbf{U}$ is a $M_1 \times ... \times M_{n-1} \times K_n \times M_{n+1} \times ... \times M_N$ -tensor for which the relation

$$(\mathcal{A} \boxtimes_n \mathbf{U})_{m_1,...,m_{n-1},k_n,m_{n+1},...,m_N} \stackrel{def}{=} \sum_{1 \le m_n \le M_n} a_{m_1,...,m_n,...,m_N} U_{k_n,m_n}$$

holds.

Based on the HOSVD under mild conditions $f(\mathbf{x})$ can be represented in the form

$$f(\mathbf{x}) = \mathcal{D} \boxtimes_{n=1}^{N} \mathbf{w}_n(x_n), \tag{3}$$

where

- $\mathcal{D} \in \mathbb{R}^{r_1 \times \ldots \times r_N}$ is a special (so called core) tensor with the properties:

  (a) $r_n = rank_n(\mathcal{A})$ is the *n*-mode rank of the tensor $\mathcal{A}$, i.e. rank of the linear space spanned by the n-mode vectors of $\mathcal{A}$:

  $\{(a_{i_1,\ldots,i_{n-1},1,i_{n+1},\ldots,i_N},\ldots,a_{i_1,\ldots,i_{n-1},I_n,i_{n+1},\ldots,i_N})^T : 1 \le i_j \le I_n, 1 \le j \le N\}$,

  (b) all-orthogonality of tensor $\mathcal{D}$: two subtensors $\mathcal{D}_{i_n=\alpha}$ and $\mathcal{D}_{i_n=\beta}$ (the *n*th indices $i_n = \alpha$ and $i_n = \beta$ of the elements of the tensor $\mathcal{D}$ keeping fix) orthogonal for all possible values of n, α and $\beta : \left\langle \mathcal{D}_{i_n=\alpha}, \mathcal{D}_{i_n=\beta} \right\rangle = 0$ when $\alpha \neq \beta$. Here the scalar product $\left\langle \mathcal{D}_{i_n=\alpha}, \mathcal{D}_{i_n=\beta} \right\rangle$ denotes the sum of products of the appropriate elements of subtensors $\mathcal{D}_{i_n=\alpha}$ and $\mathcal{D}_{i_n=\beta}$,

  (c) ordering: $\left\|\mathcal{D}_{i_n=1}\right\| \ge \left\|\mathcal{D}_{i_n=2}\right\| \ge \cdots \ge \left\|\mathcal{D}_{i_n=r_n}\right\| > 0$ for all possible values of n ($\left\|\mathcal{D}_{i_n=\alpha}\right\| = \left\langle \mathcal{D}_{i_n=\alpha}, \mathcal{D}_{i_n=\alpha} \right\rangle$ denotes the Kronecker-norm of the tensor $\mathcal{D}_{i_n=\alpha}$).

- Components $w_{n,i}(x_n)$ of the vector valued functions

  $\mathbf{w}_n(x_n) = (w_{n,1}(x_n),\ldots,w_{n,r_n}(x_n))^T, 1 \le n \le N,$

  are orthonormal in L_2-sense on the interval $[a_n, b_n]$, i.e.

  $\forall n : \int_{a_n}^{b_n} w_{n,i_n}(x_n) w_{n,j_n}(x_n) dx = \delta_{i_n,j_n}, \quad 1 \le i_n, j_n \le r_n,$

  where $\delta_{i,j}$ is a Kronecker-function ($\delta_{i,j} = 1$, if $i = j$ and $\delta_{i,j} = 0$, if $i \neq j$).

For further details see [2][3][5]

2 HOSVD-based Representation of NNs

Let us consider a parameter varying system modelled by local neural networks representing local "linear time invariant (LTI) like" models in parameter space. Suppose that these local models are identical in structure, i.e. identical in the number of neurons for the certain layers and in shape of the transfer functions. The tuning of each local model is based on measurements corresponding to different parameter vector. In Fig. 4 a two parameter case can be followed. The architecture of local models is illustrated by Fig. 3. The output of such a local model can be written in matrix form as follows:

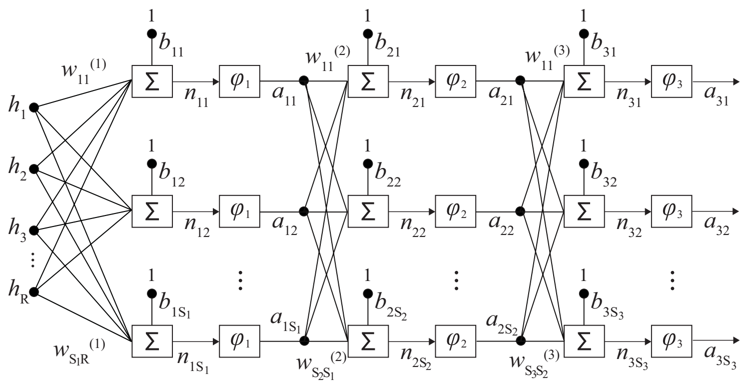

Figure 3

The architecture of the local neural network models. ($R = S_0$)

$$\mathbf{a}_3 = \varphi_3\left(\mathbf{W}^{(3)}\varphi_2\left(\mathbf{W}^{(2)}\varphi_1\left(\mathbf{W}^{(1)}\mathbf{h}\right)\right)\right)$$

where

$$\mathbf{W}^{(j)} = \begin{pmatrix} w_{11}^{(j)} & w_{12}^{(j)} & \cdots & w_{1S_{j-1}}^{(j)} & b_{j1} \\ w_{21}^{(j)} & w_{22}^{(j)} & & w_{2S_{j-1}}^{(j)} & b_{j2} \\ \vdots & & & & \\ w_{S_j1}^{(j)} & w_{S_j2}^{(j)} & & w_{S_jS_{j-1}}^{(j)} & b_{jS_j} \end{pmatrix}$$

where $j = 1..N_L$ and N_L stands for the number of layers which in our example is $N_L = 3$ (see Fig 3).

$$\mathbf{h} = \begin{pmatrix} h_1 & h_2 & \cdots & h_R & 1 \end{pmatrix}^T$$

stand for the input vector, while vector

$$\mathbf{a}_3 = \begin{pmatrix} a_{31} & a_{31} & \cdots & a_{3S_3} \end{pmatrix}^T$$

represents the output of the NN in Fig. 3.

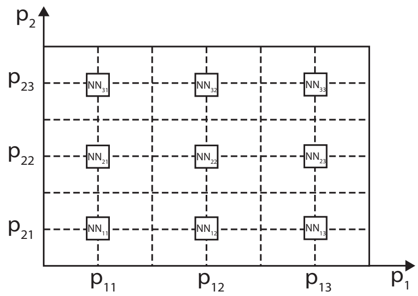

Figure 4
Example of a two dimensional parameter space, with identified neural networks as local models at equidistant parameter values

Let us assume that the behaviour of the system depends on parameter vector $\mathbf{p} = \begin{pmatrix} p_1 & p_2 & \cdots & p_N \end{pmatrix}^T$. Let $\mathbf{W}^{(j)}_{i_1 \cdots i_N}$ represent the matrix containing the weights for the *j*th layer of the local neural network model corresponding to parameter vector $p_{i_1, i_2, \ldots, i_N}$. Using the weights of the *j*th layer in all local models and the parameters p_i, where $i = 1..N$, an *N*+2 dimensional tensor $\mathbf{B} \in \Re^{I_1 \times I_2 \times \cdots \times I_N \times S_j \times (1+S_{j-1})}$ can be constructed, as follows:

$$\mathbf{W}^{(j)}_{i_1 \cdots i_N} = \left\{ \mathbf{B}_{i_1 \cdots i_N, \alpha, \beta}, 1 \le \alpha \le S_j, 1 \le \beta \le \left(1 + S_{j-1}\right) \right\}$$

$$\mathbf{W}^{(j)}_{i_1 \cdots i_N} \in \Re^{S_j \times (1+S_{j-1})}$$

By applying the HOSVD on the first *N* dimensions of tensor **B**, the core tensor $\mathcal{D}$ and for each dimension an *n*-mode singular matrix is obtained, which columns represent the discretized form of one-variable functions discussed in (1). Starting from the result of this decomposition the parameter varying model can be approximated with the help of the above mentioned local models, as follows. Tensor product (TP) transformation is a numerical approach, which can make connection between parameter varying models and higher order tensors. The weights corresponding to the *j*th layer of the parameter varying neural network model can be expressed in tensor product form, as follows:

$$\mathbf{W}^{(j)}(\mathbf{p}) = \mathcal{D} \boxtimes_{n=1}^{N} \mathbf{v}_n\left(p_n\right),$$

where $\mathcal{D}$ stands for the *N*+2 dimensional core tensor obtained after HOSVD and the elements of vector valued functions

$$\mathbf{v}_n\left(p_n\right)=\left(v_{n1}\left(p_n\right) \quad v_{n2}\left(p_n\right) \quad \cdots \quad v_{nI_n}\left(p_n\right)\right)$$

are the function values at parameter p_n of one-variable functions corresponding to the nth dimension of the core tensor $\mathcal{D}$. Finally, the output of the parameter varying model can be expressed via local neural network models illustrated in Fig. 3 in tensor product form as follows:

$$\mathbf{a}_3\left(\mathbf{p}\right)=\varphi_3\left(\mathbf{W}^{(3)}(\mathbf{p})\varphi_2\left(\mathbf{W}^{(2)}(\mathbf{p})\varphi_1\left(\mathbf{W}^{(1)}(\mathbf{p})\mathbf{h}\right)\right)\right),$$

where

$$\mathbf{W}^{(1)}\left(\mathbf{p}\right)=\mathcal{D}_1 \boxtimes_{n=1}^{N} \mathbf{v}_n^{(1)}\left(p_n\right),$$

$$\mathbf{W}^{(2)}\left(\mathbf{p}\right)=\mathcal{D}_2 \boxtimes_{n=1}^{N} \mathbf{v}_n^{(2)}\left(p_n\right),$$

$$\mathbf{W}^{(3)}\left(\mathbf{p}\right)=\mathcal{D}_3 \boxtimes_{n=1}^{N} \mathbf{v}_n^{(3)}\left(p_n\right).$$

By discarding the columns of the n-mode singular matrices corresponding to the smallest singular values model reduction can effectively be performed [7].

Conclusions

In the present paper a tensor-product based representation approach for neural networks has been proposed. By applying the HOSVD the parameter varying system can be expressed in tensor product form with the help of locally tuned neural network models. Our previous researches showed that the same concept can efficiently be applied to perform reduction in LPV systems [7]. Our next step is to analyse the impact of the reduction on the output of the system, how the approximation caused changes in weights of the NNs influence the output. We hope that it could be an efficient compromised modelling view using both the analytical and heuristical approaches.

Acknowledgement

The research was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences and by the Óbuda University

References

[1] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," SIAM *Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253-1278, 2000

[2] A. Rövid, L. Szeidl, P. Várlaki, "On Tensor-Product Model Based Representation of Neural Networks," *In Proc. of the 15th IEEE International Conference on Intelligent Engineering Systems*, Poprad, Slovakia, June 23–25, 2011, pp. 69-72

[3] L. Szeidl, P. Várlaki, " HOSVD Based Canonical Form for Polytopic Models of Dynamic Systems, " in *Journal of Advanced Computational Intelligence and Intelligent Informatics*, ISSN : 1343-0130, Vol. 13 No. 1, pp. 52-60, 2009

[4] S. Nagy, Z. Petres, and P. Baranyi, " TP Tool-a MATLAB Toolbox for TP Model Transformation " in *Proc. of 8th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics*, Budapest, Hungary, 2007, pp. 483-495

[5] L. Szeidl, P. Baranyi, Z. Petres, and P. Várlaki, "Numerical Reconstruction of the HOSVD Based Canonical Form of Polytopic Dynamic Models," in *3rd International Symposium on Computational Intelligence and Intelligent Informatics*, Agadir, Morocco, 2007, pp. 111-116

[6] M. Nickolaus, L. Yue, N. Do Minh, " Image interpolation using multiscale geometric representations, " in *Proceedings of the SPIE*, Volume 6498, pp. 1-11, 2007

[7] I. Harmati, A. Rövid, P. Várlaki, " Approximation of Force and Energy in Vehicle Crash Using LPV Type Description " in *WSEAS TRANSACTIONS on SYSTEMS*, Volume 9, Issue 7, pp. 734-743, 2010

[8] F. Khaber, K. Zehar, and A. Hamzaoui, " State Feedback Controller Design via Takagi-Sugeno Fuzzy Model : LMI Approach " in *International Journal of Information and Mathematical Sciences*, 2:3, ISBN:960-8457-10-6, pp. 148-153, 2006

[9] S. Chena, S. A. Billingsb, " Neural networks for nonlinear dynamic system modelling and identification " in *International Journal of Control*, Volume 56, Issue 2, pp. 319-346, 1992

[10] L. De Lathauwer, B. De Moor, and J. Vandewalle, " A Multilinear Singular Value Decomposition " in *SIAM Journal on Matrix Analysis and Applications*, 21(4), 2000, pp. 1253-1278

[11] N. E. Mastorakis, " The singular value decomposition (SVD) in tensors (multidimensional arrays) as an optimization problem. Solution via genetic algorithms and method of Nelder–Mead " in *SEAS Transactions on Systems*, 21(4), No. 1, Vol. 6, 2007, pp. 17-23

[12] Anne Van Mulders, Johan Schoukens, Marnix Volckaert, Moritz Diehl " Two Nonlinear Optimization Methods for Black Box Identification Compared " in *Preprints of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, July 6-8, 2009, pp. 1086-1091

[13] Babuska R., Verbruggen H. "Neuro-fuzzy methods for nonlinear system identification", *Elsevier, Annual Reviews in Control* " Volume 27, Number 1, 2003, pp. 73-85