ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Using self-organizing maps for regression: the importance of the output function

Thomas Hecht[1], Mathieu Lefort[1,2] and Alexander Gepperth[1,2] *

1- ENSTA ParisTech - UIIS division
858, boulevard des Maréchaux, 91762 Palaiseau, France
2- Inria FLOWERS - Inria Bordeaux Sud-Ouest
200 avenue de la Vieille Tour, 33405 Talence, France

Abstract. Self-organizing map (SOM) is a powerful paradigm that is extensively applied for clustering and visualization purpose. It is also used for regression learning, especially in robotics, thanks to its ability to provide a topological projection of high dimensional non linear data. In this case, data extracted from the SOM are usually restricted to the best matching unit (BMU), which is the usual way to use SOM for classification, where class labels are attached to individual neurons. In this article, we investigate the influence of considering more information from the SOM than just the BMU when performing regression. For this purpose, we quantitatively study several output functions for the SOM, when using these data as input of a linear regression, and find that the use of additional activities to the BMU can strongly improve regression performance. Thus, we propose an unified and generic framework that embraces a large spectrum of models from the traditional way to use SOM, with the best matching unit as output, to models related to the radial basis function network paradigm, when using local receptive field as output.

1 Introduction

Self-organizing map (SOM), especially the well known Kohonen model [1], is a powerful paradigm that is traditionally used for clustering, visualization and data analysis in various domains [2, 3], where each unit is usually associated with some label. It has also been used as an associative memory of motor and visual data to learn direct and inverse models for robotic control. These models exploit the SOM ability to cope with non linear redundant data in an adaptive way and to be well suited for planning [4, 5]. The best matching unit (BMU) alone is often used for computing the motor command to send in response to a visual goal. In order to improve precision, some authors propose to extend the Kohonen model by considering the motor commands associated to units close to the BMU with various interpolation methods (see [4, 5] for a review). Similar methods are also applied to time series prediction [6].

In addition to its use as an associative memory, the SOM paradigm is also used in various feedforward and recurrent architectures to represent one data flow before combining this representation with, for example, data from other

*Thomas Hecht gratefully acknowledges funding support by the "Direction Générale de l'Armement" (DGA) and École Polytechnique.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

modalities [7] or from previous time steps [8]. This separated and combined multimodal processing allows the system to have online and adaptability properties which are highly desirable, especially in robotics [9, 10]. In most cases, the BMU position is used as the relevant SOM output (see [11, 12] for example), even if a limited number of models proposes alternative functions [7, 8, 13, 14, 15].

Targeting the use of SOM-based models for online and adaptable regression learning, especially for multimodal and robotic purposes, this article quantitatively investigates the influence of the SOM output function on the system's performance. For that purpose, we use an unified framework combining a SOM which processes the inputs and whose output, after passing through the output function, feeds a linear regression step. In the next section, we introduce our architecture and the tested output functions and in section 3, we present the performance achieved with the different functions on a robotic control task.

2 Architecture

To study the influence of the output activity of the SOM when using it in an integrated model, we consider the architecture depicted in figure 1 (see next sections for equations and details of each component). This architecture is designed for online supervised learning of an input/output relationship. We already used this processing flow in a more global unsupervised system, with another data flow representation as the target signal, for the purpose of online and adaptable multimodal correlation learning [10]. This article targets to be a step towards the improvement of such models.

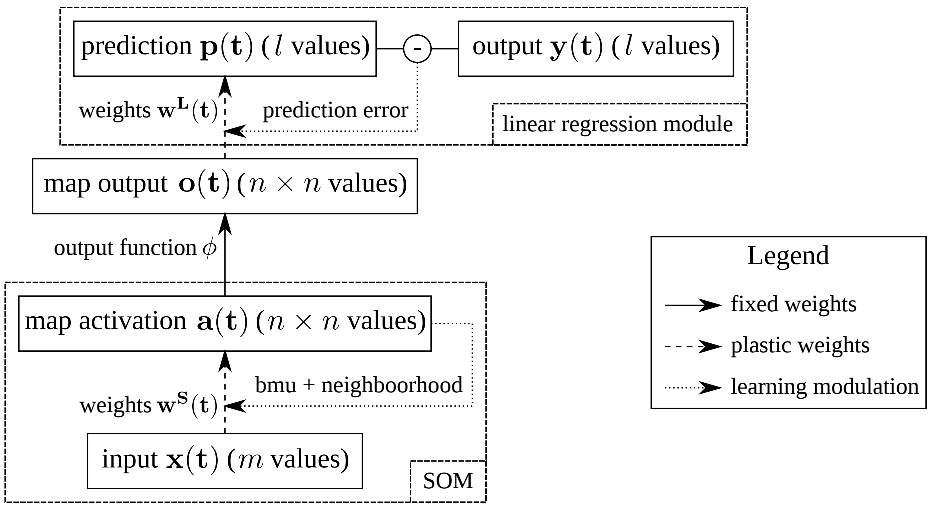


Fig. 1: Studied architecture that couples a SOM, representing the input with a generative learning procedure, with a learning regression step, learning the relationship between this SOM output representation and a given output target.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

2.1 Self-organizing map

In this article we use the Kohonen self-organizing map model. It is composed of $n \times n$ units organized on a two-dimensional grid lattice. Unit prototypes are updated with the learning rule proposed by Kohonen, with decreasing learning rate and Gaussian neighborhood [1]. The map activation $\mathbf{a(t)} = \{a_{ij}(t)\}_{(i,j)}$ is equal for each unit located at position (i, j) in the map to the euclidean distance between its prototype $\mathbf{w^S_{ij}(t)}$ and the current input $\mathbf{x(t)}$:

$$a_{ij}(t) = ||\mathbf{w^S_{ij}(t)} - \mathbf{x(t)}||$$

2.2 Output function

The SOM output activity $\mathbf{o(t)} = \{o_{ij}(t)\}_{(i,j)}$ is the activity seen from the other modules of an integrated architecture, here the linear regression. It is defined by applying an output function to the map activation, i.e. for a unit at (i, j):

$$o_{ij}(t) = \phi(\mathbf{a(t)}, \theta_{ij})$$

with θ_{ij} the set of (optional) parameters of the function ϕ. Note that the function ϕ is the same for all the units, even if its parameters can vary from unit to unit (although we do not make use of this possibility in this article).

We study various possible ϕ functions and present here three of them, inspired by the literature, one for each of three main types of possible output functions: global (section 2.2.1), local (section 2.2.2) and mixed local/global (section 2.2.3).

2.2.1 Global: Best matching unit (BMU)

The SOM output activity corresponds to the BMU, with coordinates (i^*, j^*), which is the usual way to define SOM representation (see [11, 12] e.g.):

$$o_{ij}(t) = \phi(\mathbf{a(t)}) = \begin{cases} 1 \text{ if } (i,j) = (i^*, j^*) \\ 0 \text{ otherwise} \end{cases}$$

2.2.2 Local: Gaussian similarity

The output activity of each unit depends only on the activation of this unit and is equal to a Gaussian similarity, as proposed in [8], with σ' its variance:

$$o_{ij}(t) = \phi(\mathbf{a(t)}, \sigma') = e^{-\frac{a_{ij}(t)^2}{2\sigma'^2}}$$

From a biological point of view, this output activity corresponds to overlapping receptive fields in the input space which are similar to what we can observe in visual areas. From a machine learning perspective, the complete proposed architecture corresponds to an online version of the radial basis function network (RBFN) paradigm [16] with Gaussian basis functions whose centers are placed with the SOM algorithm. RBFN is an universal approximator as proved in [17] but to our knowledge, the influence of the algorithm used for placing basis function centers on the model performance was not yet reviewed.

2.2.3 *Mixed local/global: Softmax*

The output activity takes into account local and global activation. It is equal to a softmax function on a similarity measure (such as in [14]), here chosen Gaussian, with σ' the Gaussian variance and p an integer power:

$$o_{ij}(t) = \phi(\mathbf{a(t)}, \sigma', p) = \frac{\left(e^{-\frac{a_{ij}(t)^2}{2\sigma'^2}}\right)^p}{\max\limits_{(i',j')} \left(e^{-\frac{a_{i'j'}(t)^2}{2\sigma'^2}}\right)^p}$$

Decreasing the softmax power parameter p leads to consider more units in the SOM output and provides a gradual change from a best matching unit representation ($p \to \infty$) to a Gaussian similarity one ($p = 1$), except for the normalization, closing the gap between the two previously described output functions.

2.3 Linear regression

Each output value of the linear regression module, that computes the prediction $\mathbf{p(t)} = \{p_k(t)\}_k$ made by the whole system, is equal to:

$$p_k(t) = \sum_{(i,j)} w^L_{kij}(t) o_{ij}(t)$$

with $w^L_{kij}(t)$ the weight between the unit at position (i, j) in the SOM and the k-th value of the prediction.

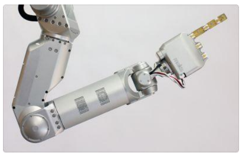
For an online architecture, the weights can be updated with an online version of linear regression. In practice, in this article, this linear regression is computed offline because this suppresses the stochasticity of the online version, thus providing a fair comparison between the tested output functions without having to perform a large amount of trials.

3 Experiments

3.1 Protocol

We test the influence of the output function in our architecture on the direct model learning of a simulated robotic arm (see figure 2). 100000 motor commands are generated uniformly in the motor space and current joint angles and corresponding spatial egocentric position of the effector were recorded every 0.5 seconds during the movements leading to 232462 data samples. The model is trained on a randomly chosen subset containing 90% of the data, the other 10% are used for testing its generalization ability.

We first train the 20×20 SOM, that we fix after checking that it is unfold, and then perform the offline linear regression on all learning examples with the different output functions. The variance of the Gaussian similarity is fixed to 200 for every output function tested. This value offers a good trade-off between the average system's performance and the spread of activity in the SOM.



Human size - Same joint limits as human arm
Joints (in degree) are in $[-80.2, 200.2] \times [-23.8, 150.1] \times [-85.1, 85.1] \times [0.4, 132.6] \times [-19.8, 199.8] \times [-58.8, 58.8] \times [-61, 61]$
Egocentric hand position (in m) is in $[-0.6, 0.6] \times [-0.8, 0.2] \times [-0.6, 0.6]$

Fig. 2: Seven degrees of freedom compliant arm (a2 from Meka Robotics).

3.2 Results

The system error is computed as the average euclidean distance between the predicted and the real spatial position of the effector over the 25830 testing examples. The average performance over 10 different SOMs, learned with random initialization, depending on the output function used are presented in figure 3.

	BMU	Gaussian similarity	Softmax			
			$p = 1$	$p = 10$	$p = 25$	$p = 100$
error (mm)	7.6 ± 1.5	1.04 ± 0.3	1.14 ± 0.3	2.09 ± 0.5	4.47 ± 1.3	7.08 ± 1

Fig. 3: Average error (in mm) depending on the SOM output function used.

We can clearly observe that the choice of the SOM output function has a high influence on the system's regression performance. Using all activities in the map (Gaussian similarity) leads to a 7 times lower prediction error than considering only the BMU. The softmax provides a large spectrum of performances interpolating between the two other functions when gradually changing the parameter p. This seems reasonable as softmax is close to the two other functions for extremal values of p (see section 2.2.3). However, it can be a worthwhile function for using SOM in an integrated architecture by providing various trade-offs between the system's performance and compression of output information.

4 Conclusion and perspectives

In this article, we study how the SOM output activity, when using it in a feed-forward architecture, influences the system's regression performance. We tested three SOM output functions, defined on the set of distance to prototypes of all units, in an unified architecture: the best matching unit (which is the traditional way to use SOM inherited from its application to classification problems), Gaussian similarity (leading to a system related to the radial basis function network paradigm) and softmax on Gaussian similarity (providing a spectrum of functions between the two first). Results obtained on the learning of a simulated robotic arm direct model clearly show that considering activities additional to the best matching unit significantly increases the system's regression performance, reducing the error by up to 7 times. These results indicate that the choice of the SOM output activity (which is independent from the prototype learning rule) is definitively an important point when using SOM in a feedforward architecture, which seems unfortunately underestimated in the literature.

This article is a first step towards the use of SOM-based architectures for regression learning, closing the gap between the traditional use of SOM for classification and algorithms from the machine learning field. Of course, our architecture can be used with other functions proposed in the artificial neural network field (see [18] for a review) and a deeper study will be necessary to determine the advantages and drawbacks of each function. Moreover, this work opens interesting perspectives to study how SOM-based algorithms can bring topological and adaptability properties to the machine learning field while keeping reasonable regression performances.

References

[1] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982.

[2] Arthur Flexer. On the use of self-organizing maps for clustering and visualization. *Intelligent Data Analysis*, 5(5):373–384, 2001.

[3] Teuvo Kohonen. Essentials of the self-organizing map. *Neural Networks*, 37:52–65, 2013.

[4] Guilherme De A. Barreto, Aluizio F.R. Araújo, and Helge J. Ritter. Self-organizing feature maps for modeling and control of robotic manipulators. *Journal of Intelligent and Robotic Systems*, 36(4):407–450, 2003.

[5] Helge Hülsen. *Self-organising locally interpolating maps in control engineering*. PhD thesis, Universität Oldenburg, 2007.

[6] Guilherme De A. Barreto. Time series prediction with the self-organizing map: A review. In *Perspectives of neural-symbolic integration*, pages 135–158. Springer, 2007.

[7] Mathieu Lefort, Yann Boniface, and Bernard Girau. Somma: Cortically inspired paradigms for multimodal processing. In *IJCNN*, 2013.

[8] Thomas Voegtlin. Recursive self-organizing maps. *Neural Networks*, 15(8):979–991, 2002.

[9] Alexander Gepperth. Efficient online bootstrapping of sensory representations. *Neural Networks*, 2012.

[10] Mathieu Lefort and Alexander Gepperth. Propre: Projection and prediction for multimodal correlations learning. an application to pedestrians visual data discrimination. In *IJCNN*, 2014.

[11] Guido Schillaci, Verena V. Hafner, and Bruno Lara. Online learning of visuo-motor coordination in a humanoid robot. a biologically inspired model. In *ICDL-Epirob*, 2014.

[12] Tamas Jantvik, Lenart Gustafsson, and Andrew P. Papliński. A self-organized artificial neural network architecture for sensory integration with applications to letter-phoneme integration. *Neural computation*, 23(8):2101–2139, 2011.

[13] Igor Farkaš and Matthew W. Crocker. Systematicity in sentence processing with a recursive self-organizing neural network. In *ESANN*, pages 49–54, 2007.

[14] Magnus Johnsson, Christian Balkenius, and Germund Hesslow. Associative self-organizing map. In *IJCCI*, pages 363–370, 2009.

[15] Jacob G. Martin, M. Alex Meredith, and Khurshid Ahmad. Modeling multisensory enhancement with self-organizing maps. *Frontiers in computational neuroscience*, 3, 2009.

[16] Martin D. Buhmann. Radial basis functions. *Acta Numerica 2000*, 9:1–38, 2000.

[17] Jooyoung Park and Irwin W. Sandberg. Universal approximation using radial-basis-function networks. *Neural computation*, 3(2):246–257, 1991.

[18] Wlodzislaw Duch and Norbert Jankowski. Survey of neural transfer functions. *Neural Computing Surveys*, 2(1):163–212, 1999.