Raport Badawczy

Research Report

RB/38/2016

Topological derivative and neural network for inverse problems of coupled models

M. Lipnicka, K. Szulc,
A. Żochowski

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences

POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 3810100

fax: (+48) (22) 3810105

Kierownik Zakładu zgłaszający pracę:
Prof. dr hab. inż. Antoni Żochowski

Warszawa 2016

SYSTEMS RESEARCH INSTITUTE
POLISH ACADEMY OF SCIENCES

Topological derivative and Neural Network for Inverse Problems of Coupled Models

Marta Lipnicka

University of Lodz, Faculty of Mathematics and Computer Science,
Banacha 22, 90-238 Lodz, Poland; marta@math.uni.lodz.pl

Katarzyna Szulc, Antoni Żochowski

Systems Research Institute of the Polish Academy of Sciences,
Newelska 6, 01-447 Warsaw, Poland e-mail: szulc@ibspan.waw.pl,
zochowsk@ibspan.waw.pl

Acknowledgement: This work is also partially supported by the National Science Centre under contract no. UMO-2014/15/B/ST1/00067

Warszawa 2016

Contents

Chapter 2

Neural networks

Artificial neural networks have several practical applications. One of them is the approximating some mapping [15], [4], [5]. We can generate such artificial neural network, which for some input vector calculates the appropriate output vector. The most commonly used networks to approximate some mappings are feedforward neural networks and Elman's neural networks. In this paper we use feedforward neural networks and Elman's neural networks to approximate some inverse mapping which for given input vector calculates the locations of hole.

2.1 Inverse mapping

Let us consider the interior sub-domain ω. In ω we have some fixed number of holes (1, 2 or 3). Each of holes is a circle with some small fixed radius. Let us consider, without loss of generality, that we have one hole in ω. We assume that the center of this hole is defined as (x_1, x_2). Our aim is to calculate the location of the hole inside ω based of the solution of the differential equation.

Let us consider mapping $g_1 : \mathbb{R}^2 \to \mathbb{R}^7$,

$$g_1(x_1, x_2) = [a_0, a_1, b_1, a_2, b_2, a_3, b_3],$$

where
$(x_1, x_2)-$ location of one hole in ω and
$a_0, a_1, b_1, a_2, b_2, a_3, b_3-$ coefficients in the Fourier series expansion function $u(\alpha)$, $\alpha \in [0, 2\pi)$.
Coefficients of the Fourier series depends on the solution of the differential equation and thus the location of the hole.

Using mapping g_1 we can calculate vector $(a_0, a_1, b_1, a_2, b_2, a_3, b_3)$ but we need to do the inverse procedure. It means that we must consider the mapping that for vector $(a_0, a_1, b_1, a_2, b_2, a_3, b_3)$ calculates the location of the hole (x_1, x_2). To this end we consider the inverse mapping.

Let us consider the inverse mapping defined as $f = g_1^{-1}$. It means that

$$f(a_0, a_1, b_1, a_2, b_2, a_3, b_3) = [x_1, x_2].$$

Mapping f for coefficients of the Fourier series calculates the location of the hole in ω.

From the mathematical point of view, the inverse mapping g_1^{-1} is difficult to evaluate. In this case we can use artificial neural networks to determine the inverse of mapping. We use the artificial neural networks to approximate mapping f. To solve this problem we use the feedforward and Elman's neural networks.

2.2 Learning set

To realize an approximation procedure of inverse mapping we must create an artificial neural network. This network has the following topology. The network has the input vector consisting of seven oeficients, each of which corresponds to one coefficient of the vector $a_0, a_1, b_1, a_2, b_2, a_3, b_3$. The network has a one hidden layer consisting of four neurons. In addition, the network has a two neurons in the output layer corresponding to the values x_1, x_2. Additionally, Elman's network has four neurons in the context layer. Due to kind of the problem we use sigmoidal activation function for each of the layers.

After generating the network we must relise a learning procedure. We use the Levenberg-Marquard learning procedure. To realise the learning procedure we need a learning set. Let N be a size of learning set $L = \{P, T\}$, where P is a set of patterns defined as

$$P = \{p_1, \ldots, p_N\}$$

such that

$$p_i = [p_i^1, \ldots, p_i^7] = [a_0, a_1, b_1, a_2, b_2, a_3, b_3]$$

for $i = 1, \ldots, N$ and T is a set of targets defined as

$$T = \{t_1, \ldots, t_N\}$$

such that

$$t_i = [t_i^1, t_i^2] = [x_1, x_2]$$

for $i = 1, \ldots, N$. For all $i = 1, \ldots, N$ p_i is a vector of coefficients of the Fourier series and t_i is location of hole.

In order to generate the learning set that contains N elements we generate a random hole in the ω N times. Next for each hole we consider the differential equation and then we solve them. We use the solution of the differential equation on $\partial\omega$ to generate Fourier series, whose the first seven coefficients are the values vector of mapping g_1. We define the values of mapping g_1 as arguments of mapping f and arguments of mapping g_1 as values of mapping f and we obtain elements of learning set.

For our learning set we realize learning procedure using the Levenberg-Marquard method. After that we obtain the learned neural network. This network for the vector of coefficients of the Fourier series calculates location of hole.

2.3 Numerical results

Here we present some results from numerical computations. We use Matlab environment. The size of learning set is 1000 elements and the size of testing set is 50 elements. In Fig.2.3 we present an architecture of the network given by Matlab. In Fig.2.3 and 2.3 we have the result of training performance and values of gradient, respectively. Fig.2.3 present the fitness of results to the target.

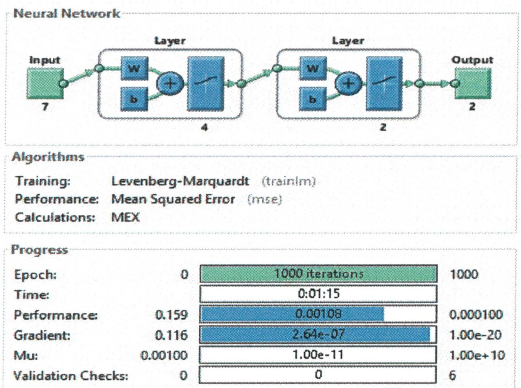

Figure 2.1: Architecture of the Matlab network

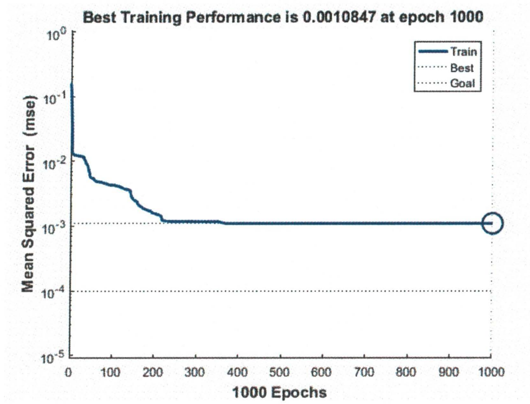

Figure 2.2: Training performance

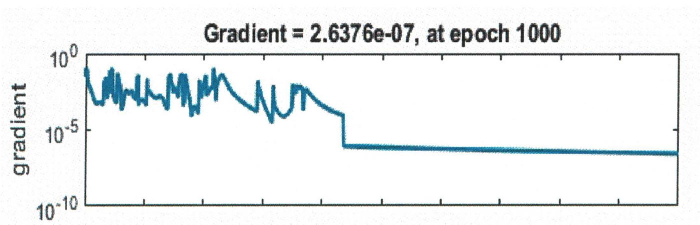

Figure 2.3: Values of the gradient

2.4 The case of several holes

Let us consider mapping $g_2 : \mathbb{R}^4 \to \mathbb{R}^7$,

$$g_2(x_1^1, x_2^1, x_1^2, x_2^2) = [a_0, a_1, b_1, a_2, b_2, a_3, b_3],$$

where
(x_1^1, x_2^1)– location of the first hole in ω,
(x_1^2, x_2^2)– location of the second hole in ω and
$a_0, a_1, b_1, a_2, b_2, a_3, b_3$– coefficients in the Fourier series expansion function $u(\alpha)$, $\alpha \in [0, 2\pi)$. Coefficients of the Fourier series depends on the solution of the differential equation and thus the location of two holes.

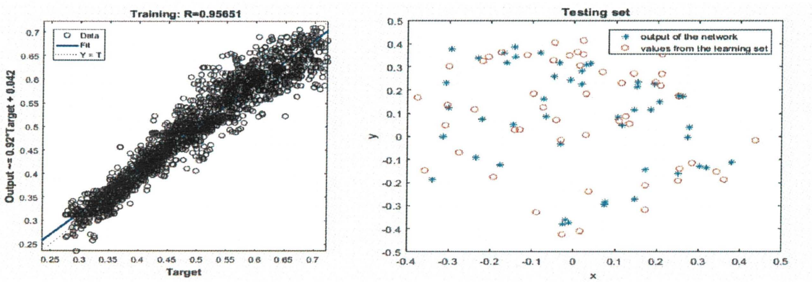

Figure 2.4: Fitness to target

Using mapping g_2 we can calculate vector $(a_0, a_1, b_1, a_2, b_2, a_3, b_3)$ but we need to do the inverse procedure. It means that we must consider the mapping that for vector $(a_0, a_1, b_1, a_2, b_2, a_3, b_3)$ calculates the location of two holes $(x_1^1, x_2^1, x_1^2, x_2^2)$. To this end we consider the inverse mapping.

Let us consider the inverse mapping defined as $f = g_2^{-1}$. It means that

$$f(a_0, a_1, b_1, a_2, b_2, a_3, b_3) = [x_1^1, x_2^1, x_1^2, x_2^2].$$

Mapping f for coefficients of the Fourier series calculates the location of two holes in ω.

In the same way we define the case of three holes.

2.5 Conclusion

In this paper we use artificial neural networks as an approximator of some mapping which for vector of coefficients of the Fourier series calculates location of one, two or three holes. The problems considered in the paper, especially those including several holes, are quite difficult because the voids which we want to localize are screened from the observation. They are contained in ω, while the goal functional is computed on Ω. As a result, the goal functionals are not very sensitive to the changes of the configuration and this impedes the use of algorithms based on gradients (shape derivatives).

Bibliography

[1] P. Fulmanski, A. Lauraine, J.-F. Scheid, J. Sokołowski, A level set method in shape and topology optimization for variational inequalities, Int. J. Appl. Math. Comput. Sci., Vol.17, No 3, p.413-430 (2007).

[2] S. Garreau, Ph. Guillaume, M. Masmoudi, The topological asymptotic for PDE systems: the elasticity case, SIAM J. Control Optim., vol. 39, No 6, pp. 1756-1778 (2001).

[3] S. M. Giusti, A. A. Novotny, C. Padra, Topological sensitivity analysis of inclusion in two-dimensional linear elasticity, Engineering Analysis With Boundary Elements, vol. 32, no. 11, pp. 926-935 (2008).

[4] M. Grzanek, A. Nowakowski, J. Sokolowski, Topological derivatives of eigenvalues and neural networks in identification of imperfections, Journal of Physics: Conference Series 135 (2008) 012046.

[5] M. Grzanek, A. Laurain, K. Szulc, Numerical algorithms for an inverse problem in shape optimization, Journal of Physics: Conference Series 135 (2008) 012047.

[6] M. Ignatowa, I. Kukavica, I. Lasiecka, A. Tuffaha, On well-posedness for a free boundary fluid-structure model, Journal of Mathematical Physics, 53 (2013).

[7] M. Iguernane, S.A. Nazarov, J.-R. Roche, J. Sokolowski, K. Szulc, Topological derivatives for semilinear elliptic equations, Int. J. Appl. Math. Comput. Sci., Vol.19, No.2, p.191-205 (2009).

[8] S.A. Nazarov, J. Sokołowski, Asymptotic analysis of shape functionals, Journal de Mathématiques pures et appliquées, Vol. 82, pp.125-196 (2003).

[9] J. Sokołowski, A. Żochowski, On the topological derivative in shape optimization, SIAM Journal on Control and Optimization. Vol. 37, No. 4, pp.1251–1272 (1999).

[10] J. Sokołowski, A. Żochowski, Optimality conditions for simultaneous topology and shape optimization, SIAM Journal on Control and Optimization, Vol. 42, No. 4, pp.1198–1221 (2003).

[11] J. Sokolowski, A. Zochowski, Modelling of topological derivatives for contact problems, Numerische Mathematik, Vol. 102, No. 1, pp.145 - 179 (2005).

[12] J. Sokolowski, A. Zochowski, Asymptotic analysis and topological derivatives for shape and topology optimization of elasticity problems in two spatial dimensions, Engineering Analysis with Boundary Elements. Vol. 32, pp.533-544 (2008).

[13] J. Sokołowski, J.-P. Zolésio, Introduction to shape optimization. Shape sensitivity analysis, Springer-Verlag, New York (1992).

[14] K. Szulc, A. Żochowski, Application of topological derivative to accelerate genetic algorithm in shape optimization of coupled models. Structural and Multidisciplinary Optimization: Volume 51, Issue 1 (2015), Page 183-192.

[15] H. White, Connectionist Nonparametric Regression: Multilayer Feedforward Networks Can Learn Arbitrary Mappings, Neural Networks, vol. 3 (1990), pp. 535-549.