ADAPTIVE STRATEGY FOR NEURAL NETWORK SYNTHESIS CONSTANT ESTIMATION

Pavel Vařacha

Tomas Bata University in Zlín, Faculty of Applied Informatic, nám. T. G. Masaryka 5555
760 01 Zlín, Czech Republic

Abstract

Neural Network Synthesis is a new innovative method for an artificial neural network learning and structural optimization. It is based on two other already very successful algorithms: Analytic Programming and Self-Organizing Migration Algorithm (SOMA). The method already recorded several theoretical as well as industrial application to prove itself as a useful tool of modelling and simulation. This paper explores promising possibility to farther improve the method by application of an adaptive strategy for SOMA. The new idea of adaptive strategy is explained here and tested on a theoretical experimental case for the first time. Obtained data are statistically evaluated and ability of adaptive strategy to improve neural network synthesis is proved in conclusion.

Keyword: neural; network; synthesis; SOMA; adaptive; strategy; structural; optimization

This Publication has to be referred as: Varacha, P[avel] (2016). Adaptive Strategy for Neural Network Synthesis Constant Estimation, Proceedings of the 26th DAAAM International Symposium, pp.0358-0364, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-07-5, ISSN 1726-9679, Vienna, Austria
DOI:10.2507/26th.daaam.proceedings.048

1. Introduction

Neural Network Synthesis (ANN synthesis) is a new innovative method for an artificial neural network learning and structural optimization. It is based on two other already very successful algorithms: Analytic Programming and Self-Organizing Migration Algorithm (SOMA). To explain all inner workings of the method would significantly exceed an intended scope of this paper. Nevertheless chapters 1.1 and 1.2 explain most important principles necessary for understanding of ideas considered in the paper. To understand Neural Network Synthesis in all details the reader is respectfully asked to explore some of other papers referenced at the end of this paper.

The main scope of the paper is to explore promising possibility to farther improve the method by application of an adaptive strategy for SOMA. The new idea of adaptive strategy is explained (chapter 2) and tested on a theoretical experimental case for the first time in chapter 3.

Obtained data are statistically evaluated and ability of adaptive strategy to improve neural network synthesis is proved in conclusion as can be seen in chapters 3.

1.1 Neural network synthesis

Development of the ANN synthesis as a successfully and effective method for the ANN designing is the main aim of my work. This chapter explains what can be understood under the term ANN synthesis and how the method works.

Clause: *Let there be a set of all neural networks with a forward running propagation* $\boldsymbol{ANN_{all}} = \{ANN_1, ANN_2, ..., ANN_i, ...\}$ *and a set of all functions* $\boldsymbol{F_{all}} = \{\boldsymbol{f_1}, \boldsymbol{f_2}, ..., \boldsymbol{f_k}, ...\}$. *Then for each* $ANNi \in \boldsymbol{ANN_{all}}$ *there exists a function* $\boldsymbol{f_k} \in \boldsymbol{F_{all}}$, *alternatively a set of functions* $\boldsymbol{F_k} \subset \boldsymbol{F_{all}}$ *such, that holds* $ANN_i \Leftrightarrow \boldsymbol{f_k}$, *alternatively* $ANN_i \Leftrightarrow \boldsymbol{F_k}$.

The Kolmogorov theorem further shows the validity of the inverse clause: *For every continuous function* $\boldsymbol{f_k} \in \boldsymbol{F_{all}}$ *there exists* $ANN_i \in \boldsymbol{ANN_{all}}$ *such, that holds* $\boldsymbol{f_k} \Leftrightarrow ANNi$.

Task: *Design an algorithm, which will by the means of the symbolic regression methods, evolutionarily scan a set* $\boldsymbol{F_{all}}$ *in order to find:*

a) $\mathbf{f_k} \Leftrightarrow \mathrm{ANN_i}$

b) $\mathbf{f_k}$, whose at least some subfunctions $\{\mathbf{f_1}, \mathbf{f_2}, ...\} \Leftrightarrow \{\mathrm{ANN_n}, \mathrm{ANN_m}, ...\}$

which solves the particular problem P with a global error $E_T < \xi$, *where* ξ *is the user defined biased tolerance threshold.*

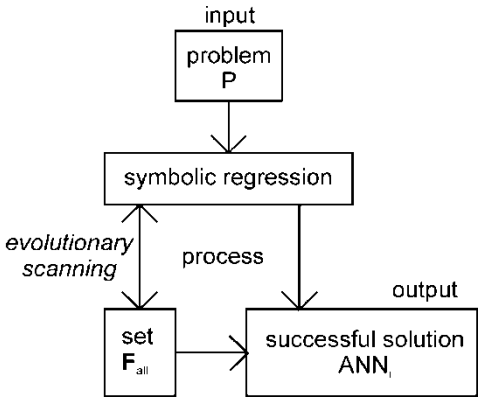

Fig. 1. Principle of the evolutionary scanning

AP can perform such evolutionary scanning above Fall set and provide the possibility to synthetize the ANN with an almost infinitely variable structure, complexity and scope. There is a very easy way of using AP for the ANN synthesis. (More information about this process can be found e.g. in [1].) The most important part is to define items of which the ANN will be composed. In this case the General Function Set (GFS) contains only three items. GFS is a set containing all possible elements from which resulted ANN can be build.

$$GFS_{all} = \{+, AN, K*x\} \quad (1)$$

Most important item of (1) is an Artificial Neuron (AN) (2) with a weighted hyperbolic tangent as a transfer function (3). The weight of output, steepness and thresholds are computed as K in AP (4) by evolutionary algorithm (see part 2.1).

GFS1 = {AN} (2)

$$AN(S) = w\frac{e^{2\,\lambda\,(S+\phi)}-1}{e^{2\,\lambda\,(S+\phi)}+1} \tag{3}$$

$$AN(S) = K_1\,\frac{e^{2\,K_2\,(S+K_3)}-1}{e^{2\,K_2\,(S+K_3)}+1} \tag{4}$$

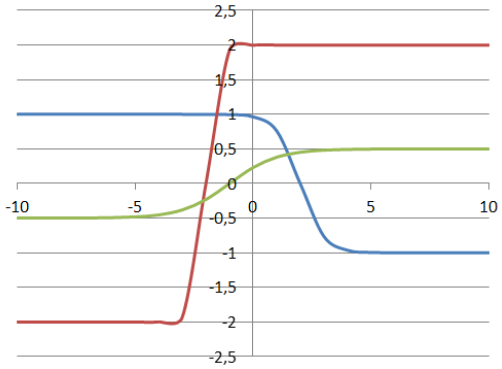

Fig. 2. AN transfer function for various w, λ, ϕ settings

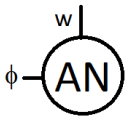

Fig. 3. Graphical example of AN

To allow more inputs into one ANN a simple plus operator (5) is used.

GFS_2 = {+} (5)

Fig. 4. Graphical example of plus operator

Finally, (6) represents the weighted input data.

*GFS_0 = K*x* (6)

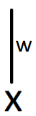

Fig. 5. Graphical example of weighted input

Under such circumstances, translation of an individual into the ANN can be easily grasped from Fig. 6. The whole process is cyclical. Individuals provided by the EA are translated into ANN. ANN are evaluated in accordance with a training data set and their global errors are used to set the fitness of these individuals. Consequently, a new generation is chosen and the whole process is repeated in the next migration loop.

T he introduced approach is not the only one possible. Different settings of the GFS were successfully used to synthetize the ANN performing classification. [2]

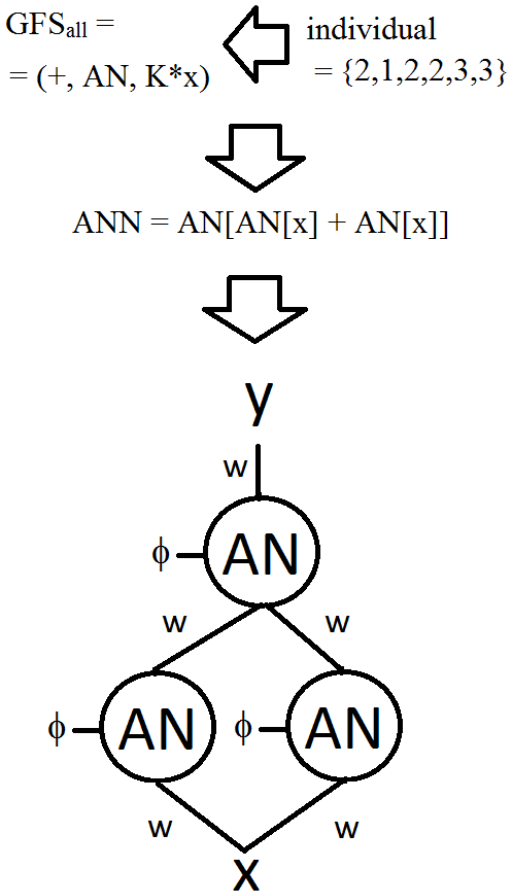

Fig. 6. Translation of an individual into ANN

1.2 Constant Processing

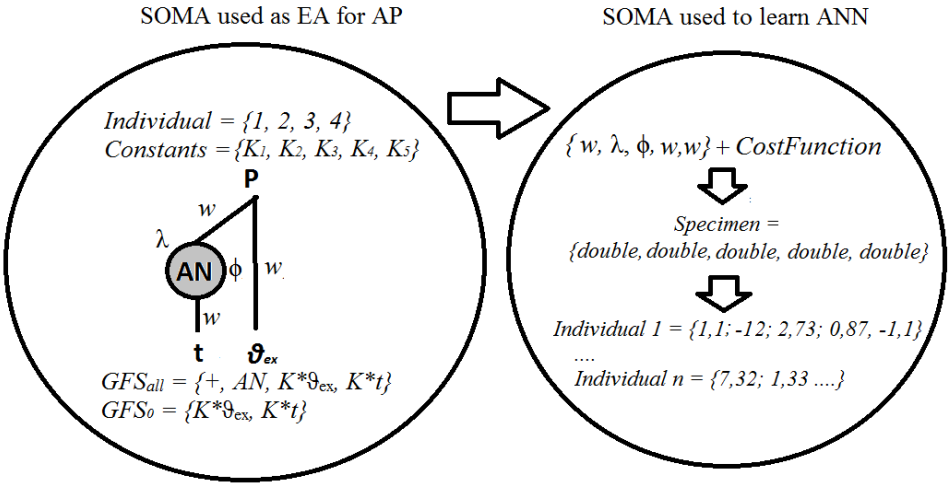

Fig. 7. Learning of a synthesized ANN

The synthesized ANN, programs or formulas may also contain constants "K", which can be defined in the GFS_0 or be a part of other functions included in the GFS_{all}. When the program is synthesized, all Ks are indexed, so K_1, K_2, …, K_n, are obtained and then all K_n are estimated. Several versions of AP exist in accordance with K_n estimation.

In this case, the asynchronous implementation of Self-Organizing Migration Algorithm (SOMA) (inside another SOMA, which operates AP) is used to estimate K_n. This is especially convenient for the ANN synthesis. K_n can be referred to as various weights and thresholds and their optimization by SOMA as ANN learning (see Fig. 7). [2] SOMA is commonly known evolutionary which is well described for example in [3].

1.3 Purpose of the study

As can be seen in previous chapter, number of constants which have to be estimated by SOMA differ with each individual. Nevertheless this fact is not considered in a standard version of Analytic Programming. Adaptive strategy proposed in this paper aim to influence control parameter of SOMA named PRT to improve SOMA's efficiency dealing with different number of constants to estimate. The main question of the study can be formulated as whether an improvement in the strategy of evolutionary algorithm used for ANN's learning can also improve overall results of ANN synthesis.

2. Adaptive Strategy

An idea of the adaptive strategy introduced in chapter 1.3 can be formulated into following hypothesis:

Hypothesis: Adaptive setting of control parameters of evolutionary algorithm based on number of estimated constants can positively influence ANN synthesis in accordance to its efficiency and speed.

For the purpose of this study, adaptive strategy is reduce to one control parameter of SOMA named PRT. For more information about PRT and its importance for SOMA, please refer [4].

Technically speaking Commonly, SOMA is set on PRT = 0.1. In contradiction this paper proposes adaptive strategy so PRT will differ from individual to individual PRT = 1 / number of K_n.

3. Experiment

In order to statistically evaluate the hypothesis stated in chapter 2. the function approximation problem was chosen as an aim of the experiment. The function (7) proposed by [5] as an appropriate approximation benchmark was chosen to be approximated by the ANN.

$$y = x_i^5 - 2x_i^3 + x_i \quad \text{where } x_i \in \langle -1, \text{by the step } 0.04, 1 \rangle \tag{7}$$

Fig. 8 (automatically generated by ANN synthesis software) shows an example of synthetized ANN approximating (7). The difference between the ANN and (7) is depicted as a red area which could be minimized by the process of synthesis.

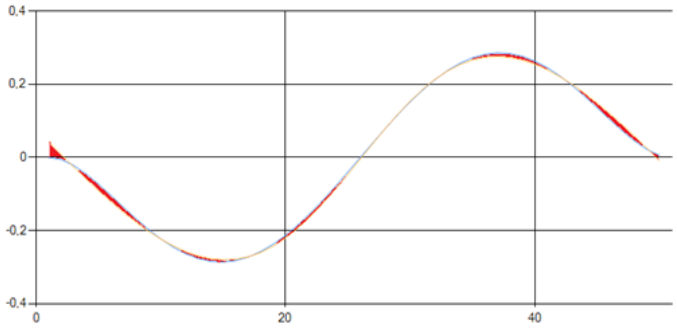

Fig. 8. Approximation of (7) by synthetized ANN

AP was executed 100 times (physically on 8 cores of the Super Micro Server) to produce an ANN with the RMSD < 0.005. The main intention was to find such an ANN which met this condition and which simultaneously used as few AN as possible. The setting of Asynchronous SOMA used as the EA for AP can be seen in Table 1. and SOMA setting used for ANN learning in Table 2.

Number of Individuals	48
Individual Parameters	100
Low	0
High	3
PathLength	3
Step	0,11
PRT	1/ depth
Divergence	0.01
Period	1

Table 1. Setting of SOMA used as EA for AP

Number of Individuals	number of Kn * 0.5 (at least 10)
Individual Parameters	100
Low	-10
High	10
PathLength	3
Step	0,11
PRT	**Based on experiment**
Divergence	0.01
Period	6

Table 2. Setting of SOMA used to optimize Kn

Based on experiment setting PRT which SOMA used to optimize Kn is set either conservatively to PRT = 0.1 or adaptively PRT = 1 / number of K_n.

4. Results

The adaptive strategy for K_n estimation consists in replacement of the static PRT value by the value which inversely depends on K_n dimension.

	PRT = 1 / number of K_n	PRT = 0.1
Average time needed for synthesis	194 s	505 s
Average number of used AN	9	15

Table 3. PRT strategy for Kn estimation

A total of 672,779 evaluations of AP individual fitness were completed during 100 AP executions and the separate SOMA run was performed for all of them to set their K_n value while PRT was set to 0.1. However, under such conditions, in 7 cases AP was not able to find a sufficient ANN at all. This results can be clearly interpreted as a confirmation of the examined hypothesis. Nevertheless, there are limited by the fact that our study aim to explore adaptive strategy of only one parameter of one specifically selected evolutionary algorithm.

5. Conclusion

The Neural Network Synthesis was developed on the basis of AP and SOMA algorithms. The method was successfully tested on the real life problems [6] – [9] (as can be seen in extended electronic version of this paper) as well as on widely recognized benchmark functions [10], [11] with respect to the function approximation, prediction and problems. This applications make contribution of this research significant as it help the ANN synthesis prove itself as a useful and efficient tool for nonlinear modelling in comparison with competing methods [12] – [17].

Obtained results of experiment considered in this paper proves an ability of proposed adaptive strategy to further improve ANN synthesis. The experiment was based on theoretical example. This open a possibility applied such approach also on some more time demanding practical industrial case. Such exploration will be subjected by our future research in combination with our recent results considering an evolutionary control [18] and [19].

6. Acknowledgements

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Programme project No. LO1303 (MSMT-7778/2014) and by the Technology Agency of the Czech Republic as a part of the project called TA03010724 AV and EV LED liminaire with a higher degree of protection.

7. References

[1] Zelinka I., Vařacha P., Oplatková Z., Volná, E. Structural Synthesis of Neural Network by Means of Analytic Programmig. In 12th International Conference on Soft Computing. Czech Republic, VUT Brno, 2006, p. 25-30.

[2] Vařacha P. Neural Network Synthesis Dealing with Classification Problem. In Recent Researches in Automatic Control. Montreux : WSEAS Press, 2011, p. 377-382.

[3] Šenkerik R., Zelinka I., Davendra D., Oplatkova Z. Utilization of SOMA and differential evolution for robust stabilization of chaotic Logistic equation, Computers & Mathematics with Applications, Volume 60, Issue 4, 2010, Pages 1026-1037, ISSN 0898-1221, doi 10.1016/j.camwa.2010.03.059.

[4] Šenkeřík R. Oplatkova Z., Zelinka I, Davendra D., Synthesis of feedback controller for three selected chaotic systems by means of evolutionary techniques: Analytic programming, Mathematical and Computer Modelling, Available online 27 May 2011, ISSN 0895-7177, 10.1016/j.mcm.2011.05.030.

[5] Vařacha P., Zelinka I. Analytic Programming Powered by Distributed Self-Organizing Migrating Algorithm Application. In IEEE Proceedings 7th International Conference Computer Information Systems and Industrial Management Applications. Ostrava : IEEE Computer Society, 2008, p. 99-100.
[6] Prechelt L., Proben1—A Set of Neural Network Benchmark Problems and Benchmarking Rules, Universität Karlsruhe, 1994, Germany
[7] Mangarianm O.L., Wolberg W.H., Cancer diagnosis via linear programming, SIAM News, Volume 23, Number 5, 1990, p. 1-18.
[8] Král E., Dolinay V., Vašek L., Vařacha P. Usage of PSO Algorithm for Parameters Identification of District Heating Network Simulation Model. In 14th WSEAS International Conference on Systems. Latest Trands on Systems.Volume II, Rhodes, WSEAS Press (GR) , 2010. p. 657-659
[9] CHRAMCOV, Bronislav. Identification of time series model of heat demand using Mathematica environment. In Recent Researches in Automatic Control. Montreux : WSEAS Press, 2011, s. 346-351.
[10] Zelinka I., Studies in Fuzziness and Soft Computing, New York : Springer-Verlag, 2004.
[11] Koza J. R., Genetic Programming, MIT Press, 1998, ISBN 0-262-11189-6
[12] Jui-Yu W., MIMO CMAC neural network classifier for solving classification problems, Applied Soft Computing, Volume 11, Issue 2, The Impact of Soft Computing for the Progress of Artificial Intelligence, 2011, p. 2326-2333,
[13] Falco D.I., Cioppa E., Tarantino, Discovering interesting classification rules with genetic programming, Applied Soft Computing 1, 2002 p. 257–269.
[14] Brameier M., Banzhaf W., A comparison of linear genetic programming and neural networks in medical data mining, IEEE Transactions on Evolutionary.
[15] Turner et al., Grammatical Evolution of Neural Networks for Discovering Epistasis among Quantitative Trait Loci Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics Book Series Title: Lecture Notes in Computer Science 2010 Publisher: Springer Berlin / Heidelberg, p: 86 – 97.
[16] Vonk E., Jain L.C., Johnson R.P., Automatic Generation of Neural Network Architecture Using Evolutionary Computation, Advances in Fuzzy Systems – Applications and Theory, Volume 14, 1997, World Science, ISBN: 981-02-3106-7.
[17] Koza J. R. et al. Genetic Programming III; Darwinian Invention and problem Solving, Morgan Kaufmann Publisher, 1999.
[18] Šenkeřík R., Zelinka I., Pluháček M., Davendra D., Komínková Oplatková Z., Chaos Enhanced Differential Evolution in the Task of Evolutionary Control of Selected Set of Discrete Chaotic Systems. The Scintific World Journal, 2014, n. 836484, p. 1-13. ISSN 2356-6140.
[19] Šenkeřík R., Zelinka I., Pluháček M., Komínková Oplatková Z., Evolutionary Control of Chaotic Burgers Map by means of Chaos Enhanced Differential Evolution. International Journal of Mathematics and Computers in Simulations, 2014, n. 8, p. 39-45. ISSN 1998-0159.