Application of Particle Swarm Optimization Algorithm to Neural Network Training Process in the Localization of the Mobile Terminal

Jan Karwowski[1], Michał Okulewicz[1], and Jarosław Legierski[2]

[1] Warsaw University of Technology, Faculty of Mathematics and Information Science, Koszykowa 75, 00-662 Warsaw, Poland
M.Okulewicz@mini.pw.edu.pl
[2] Orange Labs Poland, Obrzeżna 7, 02-691 Warszawa, Poland
Jaroslaw.Legierski@orange.com

Abstract. In this paper we apply Particle Swarm Optimization (PSO) algorithm to the training process of a Multilayer Perceptron (MLP) on the problem of localizing a mobile GSM network terminal inside a building.

The localization data includes the information about the average GSM and WiFi signals in each of the given (x,y,floor) coordinates from more than two thousand points inside a five story building.

We show that the PSO algorithm could be with success applied as an initial training algorithm for the MLP for both classification and regression problems.

Keywords: Particle Swarm Optimization, Neural Network training, Mobile terminal localization.

1 Introduction

In recent years, biologically inspired computational intelligence algorithms have grown a lot of popularity. Examples of such algorithms, based on the idea of swarm intelligence, are PSO and bird flocking algorithm. Those algorithms have been used in the real world applications in the area of computer graphics and animation [17], solving hard optimization tasks [14] or document clustering [7].

PSO has not been widely tested in a high dimensional optimization problems but has already been applied in the process of training a neural network [11]. Error backpropagation (BP) is a well known algorithm for learning multilayer perceptron. In our paper we use stochastic gradient descent backpropagation algorithm proposed in [5], Algorithm (4). In experiments we used implementation of BP from [1].

In recent years on telecommunication market we can observe a large number of mobile application and services based on mobile terminal location. Unfortunately very popular location method used by most of them, is based on the Global Positioning System (GPS). GPS does not work inside the buildings, because the

L. Iliadis, H. Papadopoulos, and C. Jayne (Eds.): EANN 2013, Part I, CCIS 383, pp. 122–131, 2013.

GPS signal is to weak to be propagated indoors. Another mobile terminal location method is based on the Location Based Services (LBS) in communication service providers networks. LBS are characterized by significant location error e.g. in Poland location error in mobile networks reaches values from 170 to 400 meters in urban area [18]. Therefore an easy for implementation, low from cost calculation point of view and fast algorithms to locate the mobile phones inside the buildings are an urgent business need in order to create new and innovative applications and services.

In this article the authors show that PSO algorithm could be with success applied as an initial algorithm for training MLP with two and more hidden layers and that the success of the algorithm does not depend much on the choice of the parameters for PSO or the MLP architecture (which is not true for BP as shown in [10] [22] [23]).

The algorithms were tested in the application of fingerprinting technique on localizing mobile terminal in the building. The localization is based on the GSM and WiFi signals.

The remainder of the paper is organized as follows: First, in section 2 we briefly summarize the PSO algorithm. In section 3 the problem of localizing the mobile terminal is presented. Application of the PSO algorithm and the experimental setup and results are presented in sections 4. The last section summarizes the experimental findings and concludes the paper.

2 Particle Swarm Optimization Algorithm

PSO algorithm is an iterative optimization method proposed in 1995 by Kennedy and Eberhart [12] and further studied and developed by many other researchers, e.g., [20], [19], [6]. In short, PSO implements the idea of swarm intelligence to solving hard optimization tasks.

In the PSO algorithm, the optimization is performed by the set of particles which are communicating with each other. Each particle has its location and velocity. In every step t a location of particle i, x_t^i is updated based on particle's velocity v_t^i:

$$x_{t+1}^i = x_t^i + v_t^i. \tag{1}$$

In our implementation of PSO (based on [2] and [20]) in $t+1$ iteration ith particle's velocity v_{t+1}^i is calculated according to the following rules:

1. a weighted center c_t^i of $x_{best}^{neighbours_i}$, x_{best}^i and x_t^i points is computed:

$$c_t^i = \frac{g x_{best}^{neighbours_i} + l x_{best}^i + x_t^i}{3} \tag{2}$$

2. a new velocity is computed on the basis of current particle location x_t^i, a weighted center c_t^i and current particle's velocity v_t^i,

$$v_{t+1}^i = u^{(u-ball)} \left\| c_t^i - x_t^i \right\| + (c_t^i - x_t^i) + a v_t^i, \tag{3}$$

where

$x_{best}^{neighbours_i}$ represents the best location in terms of optimization, found hitherto by the neighbourhood of the ith particle,

x_{best}^{i} represents the best location in terms of optimization, found hitherto by the particle i,

g is a neighbourhood attraction factor,

l is a local attraction factor,

a is an inertia coefficient,

$u^{(u-ball)}$ is a random vector with uniform distribution over a unit size n–dimensional ball.

In our case the value of the fitness function for the PSO algorithm is the sum of values of the errors on whole training set and the vector in a search space is a vector of weights of the neural network.

As already mentioned, Jing et al. [11] applied PSO to training MLP. It was used to train a much smaller MLP than we are using. Jing et al. network has layers consisting of 3, 6 and 1 neurons for input, hidden and output layer, respectively. Our networks have several neurons in each of the hidden layers. Moreover, in our approach PSO algorithm is used to find initial solution which is later used as starting point for BP algorithm.

3 Localization of a Mobile Terminal in a Building

The problem of localizing a terminal of a mobile network in a building with a usage of fingerprinting technique has been already presented in the literature [3,4,13,15].

The task is to predict location of mobile terminal – triple of a floor and x, y coordinates in the floor plane. In the fingerprinting technique the model on which the predictions are done is constructed on the basis of previously recorded WiFi or GSM signals in the known locations in a building.

It was also shown that the WiFi signals based localization methods are more precise than the GSM signals based, while on the other hand they might not be always available (f.e. during the loss of electricity in the building or simply lack of enough WiFi access points). We will show the difference in that precision found in our research.

The dataset and the problem discussed in this article are the same one as presented in the article showing the basis for predicting credibility of floor predictions [9]. In this article we present a comparison of predictions based on WiFi and GSM signals and also a comparison between MLP initialized with random weights and initially trained with PSO algorithm.

The dataset consists of a 1199 training and validation points gathered in a 1.5 x 1.5m or 3.0 x 3.0m grid at different dates (two series of measurements) and 1092 test points gathered at another day in the grid shifted by half of the resolution of the original grid (one series of measurements). The data comes from all the floors (including ground floor) of a five story building of Faculty of Mathematics and Information Science of Warsaw University of Technology. The data was gathered in halls, corridors, laboratories and lecture rooms.

Each vector of the data consists of the average Received Signal Strength (RSS) from the Base Transceiver Stations (BTS) of the GSM system and RSS from the Access Points (AP) of the WiFi network. Each vector is labeled with x, y and floor coordinates defined for each of the points in which the data was gathered.

Therefore the task of localizing a mobile terminal in a building may be looked upon as a regression task for x and y coordinates and classification task for floor coordinate, making a neural network a universal tool for both of the tasks.

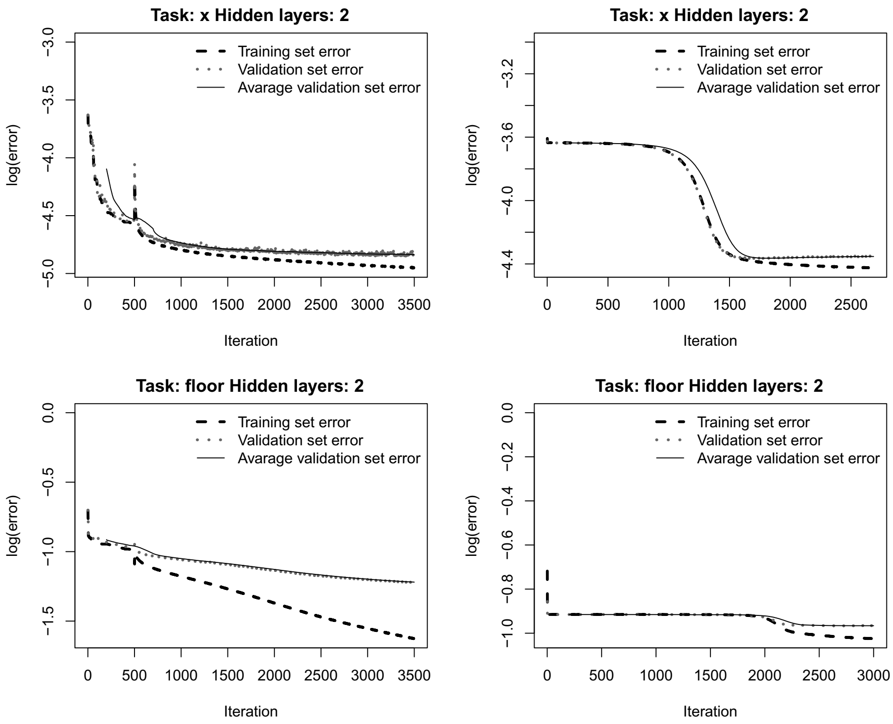

Fig. 1. The example training process of a neural network with 2 hidden layers for floor classification and regression in one of the directions. The left column presents the runtime where first 500 iterations were done by the PSO algorithm and the rest by the BP algorithm. The right column presents runtime of the BP algorithm.

4 Tests and Results

Datasets for all the tests were composed in a following way:

– one series of measurements was chosen as a training set,
– another series of measurements (in the same grid as training set, but gathered on a different day) was used as a validation set,
– a series of measurements gathered on a different day in a shifted grid was used as a test set.

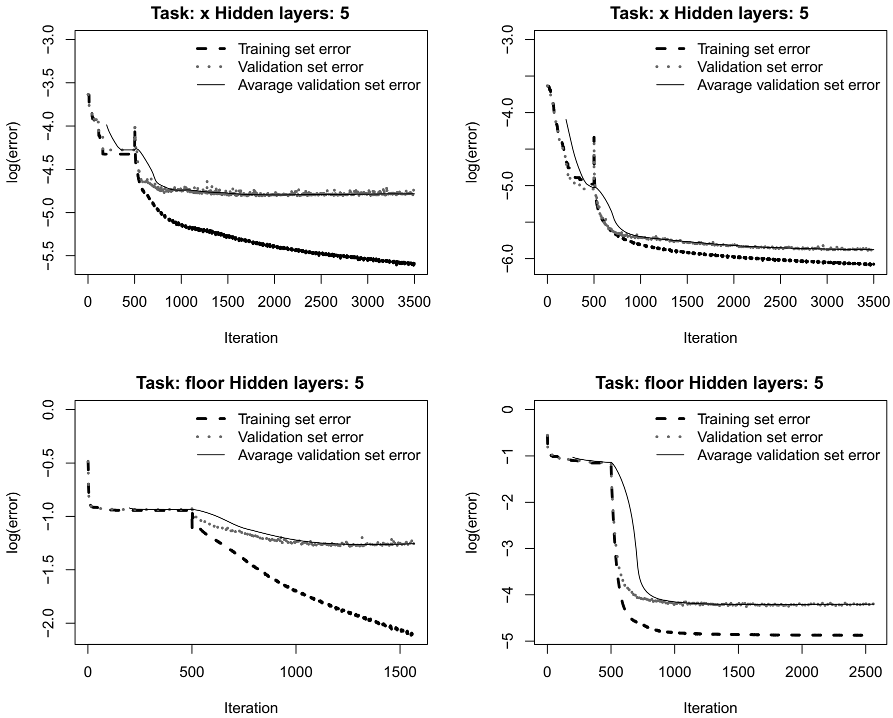

Fig. 2. The example training process of a neural network with 5 hidden layers for floor classification and regression in one of the directions. First 500 iterations were done by the PSO algorithm and the rest by the BP algorithm. Left column shows the runtime for the GSM data and the right one for the WiFi data.

All the tests were run in the following scenario:

1. A neural network was trained in a batch mode by the PSO algorithm.
2. An initially trained network was used as a starting point and trained by the online BP algorithm.
3. Separately a network was trained for comparison from a random point by the online BP with the same parameters.
4. In both cases the training process by online BP was stopped when the error on validation set began to rise.

The baseline of all experiments was performance of the network initialized by PSO with 500 iterations and 40 particles and further trained by the BP for at most 3000 iterations. The parameters for the baseline PSO were set as follows $g = 1.4$, $l = 1.4$, $a = 0.63$, $P(Particle_i \text{ is neighbor of } Particle_j) = 0.5$. For the baseline BP the learning rate for each example was set to 0.0008, momentum was set to 0.3 and value of average error on validation set was counted over 200 iteration. The comparison of the baseline experiment against experiments with

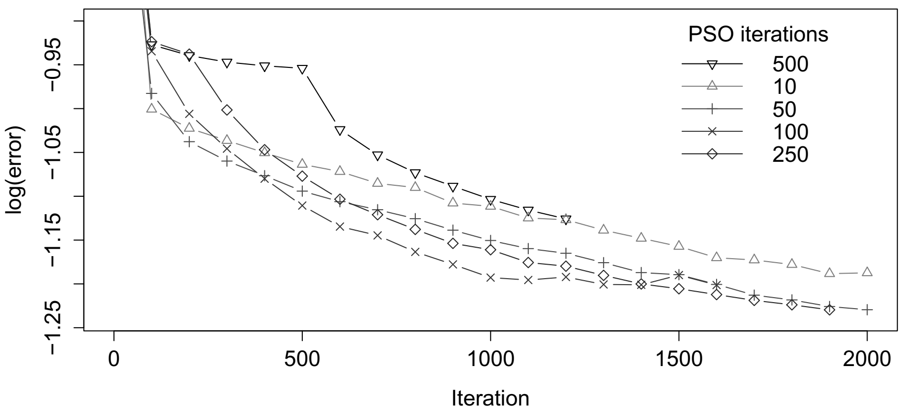

Fig. 3. Comparison of the average network convergence on the validation set for different number of initial PSO iterations

different parameters are shown in the Tables 1-4. In each comparison only the mentioned parameters are changed while the rest remains the same as in baseline experiment. The comparison has been done on the basis of classification accuracy (i.e. the ratio of the properly classified records from the test set) for the floor, and on the basis of 0.9 quantile of absolute error given in meters for prediction of the X and Y coordinates. The result of the baseline is boldfaced in each of the tables.

The PSO was additionally run with the following parameters:

– 5, 10, 25, 50, 100, 150, 200, 250 iterations,
– $g = 2.2$ and $l = 0.6$, $g = 0.6$ and $l = 2.2$.

The results of those tests are presented in the Tables 1 and 2
The BP was additionally run with the following parameters:

– learning rate = 0.0002, 0.0004, 0.0006, 0.0010.

The results of those tests are presented in the Table 3.

The following neural networks with following number of neurons in hidden layers were tested (all with full connections between subsequent layers):

– 60 and 60,
– 60, 40 and 20,
– 60, 40, 40 and 20,
– 60, 40, 40, 30 and 20.

The results of those tests are presented in the Table 4.

The networks had 26 inputs for GSM data and 107 inputs when used for WiFi data. Each input represented a strength of the given BTS or AP signal present anywhere in the building. For regression tasks the networks had one output

Table 1. Comparison for different number of PSO iterations for GSM and WiFi data. The results of baseline experiment are boldfaced.

PSO iterations	Classification accuracy		0.9 quantile of $\lvert e_X\rvert$		0.9 quantile of $\lvert e_Y\rvert$	
	GSM	WiFi	GSM	WiFi	GSM	WiFi
5	54%	98%	8.66m	5.93m	11.98m	6.55m
10	53%	98%	8.91m	5.31m	11.08m	6.03m
25	53%	98%	8.37m	5.35m	11.40m	6.29m
50	53%	98%	8.80m	4.96m	12.87m	6.14m
100	52%	98%	8.61m	5.33m	11.76m	6.94m
150	56%	97%	9.13m	5.45m	12.47m	6.47m
200	54%	97%	8.90m	5.01m	11.92m	6.48m
250	53%	98%	8.77m	5.46m	11.55m	6.42m
500	**54%**	**98%**	**9.11m**	**5.18m**	**11.66m**	**6.48m**

Table 2. Comparison for different PSO local and global attraction factor parameters for GSM and WiFi data. The results of baseline experiment are boldfaced.

PSO parameters	Classification accuracy		0.9 quantile of $\lvert e_X\rvert$		0.9 quantile of $\lvert e_Y\rvert$	
	GSM	WiFi	GSM	WiFi	GSM	WiFi
0.6 2.2	53%	98%	8.86m	5.47m	11.86m	6.74m
1.4 1.4	**54%**	**98%**	**9.11m**	**5.18m**	**11.66m**	**6.48m**
2.2 0.6	54%	98%	11.38m	5.54m	12.12m	6.40m

Table 3. Comparison for different values of a BP learning rate for GSM and WiFi data. The results of baseline experiment are boldfaced.

Learning rate	Classification accuracy		0.9 quantile of $\lvert e_X\rvert$		0.9 quantile of $\lvert e_Y\rvert$	
	GSM	WiFi	GSM	WiFi	GSM	WiFi
0.0002	50%	98%	9.53m	6.21m	12.29m	7.35m
0.0004	52%	98%	9.00m	5.75m	11.96m	6.66m
0.0006	53%	98%	8.80m	5.48m	11.89m	6.68m
0.0008	**54%**	**98%**	**9.11m**	**5.18m**	**11.66m**	**6.48m**
0.0010	54%	98%	8.77m	5.56m	11.73m	6.10m

Table 4. Comparison for different number of neural network hidden layers for GSM and WiFi data. The results of baseline experiment are boldfaced.

Hidden layers	Classification accuracy		0.9 quantile of $\lvert e_X\rvert$		0.9 quantile of $\lvert e_Y\rvert$	
	GSM	WiFi	GSM	WiFi	GSM	WiFi
2	52%	98%	9.06m	5.54m	11.93m	6.79m
3	**54%**	**98%**	**9.11m**	**5.18m**	**11.66m**	**6.48m**
4	52%	98%	8.67m	5.37m	11.73m	5.82m
5	53%	98%	8.92m	5.27m	12.65m	5.87m

representing the predicted location for X or Y coordinate in the building. For a classification the networks had six outputs, for each of the six classes representing the floor of the building.

The example training processes of BP and PSO+BP algorithm for a network with two hidden layers are presented on the Fig. 1. For the larger number of hidden layer BP starting from random point was not able to achieve in 3000 iterations better results then classifying all test data as the most frequently occurring floor and regression models predicted the weighted average in both directions.

Figure 2 shows the training processes of PSO+BP algorithm of the neural network with 5 hidden layers for GSM and WiFi data.

5 Discussion and Conclusions

As can be seen from results a good starting point for online BP algorithm is very important. When choosing random weights for BP error on the training set was not decreasing at all or was constant for a large number of iterations and network started to converge only after several hundreds of iterations. Popular approach to this problem is selecting learning rate and network topology individually for every problem or using adaptive learning rate. [22,21]

Our approach is to start learning network with PSO for a small number of iterations and then set the network weights found by this method as a starting point for BP. The results show that convergence of BP is much faster and stable than in experiments where BP was started from random point.

Our results show that the PSO algorithm is useful in the process of training MLP to solve the problem of localizing a mobile terminal. Hybrid learning method has given better results than plain BP. The important advantage of the PSO (being a method of global optimization) is possibility for leaving local minima, while BP has been reported to stuck in them [8].

It is important to notice, that one iteration of the PSO algorithm is much slower than one iteration of BP algorithm (approximately by a factor of half a number of particles). On the other hand it is possible to efficiently implement a parallel version of the PSO and even using a very small number of PSO iterations (e.g. 10) is sufficient for finding a good starting point for BP (although the convergence of BP was faster for a larger number of iterations as presented in Fig. 3).

Our results also confirm that WiFi based localization gives a more accurate location of the mobile terminal, especially in the problem of floor classification. Floor classification based on GSM signals was accurate in about 55% of observations and based on WiFi signals in about 98%.

Further research on the hybrid PSO+BP method should include comparison with another methods (e.g. BP algorithm with adaptive learning rate, batch BP algorithm). Research on the problem of localizing a mobile terminal with the usage of neural networks should take into account learning and testing on not aggregated data and also observations about the credibility of predictions for the GSM data should be considered [9].

Acknowledgements. Study was partially supported by research fellowship within "Information technologies: Research and their interdisciplinary applications" agreement number POKL.04.01.01-00-051/10-00 and partially financed from the funds of National Science Centre granted on the basis of decision DEC–2012/07/B/ST6/01527.

The analyses of the results and the plots have been done with the usage of R[16].

References

1. Neuroph Java Neural Network Framework (2012), http://neuroph.sourceforge.net/
2. Standard PSO 2011 (2012), http://www.particleswarm.info/
3. Benikovsky, J., Brida, P., Machaj, J.: Localization in Real GSM Network with Fingerprinting Utilization. In: Chatzimisios, P., Verikoukis, C., Santamaría, I., Laddomada, M., Hoffmann, O. (eds.) Mobile Lightweight Wireless Systems. LNICST, vol. 45, pp. 699–709. Springer, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-16644-0_60
4. Bento, C., Soares, T., Veloso, M., Baptista, B.: A Study on the Suitability of GSM Signatures for Indoor Location. In: Schiele, B., Dey, A.K., Gellersen, H., de Ruyter, B., Tscheligi, M., Wichert, R., Aarts, E., Buchmann, A.P. (eds.) AmI 2007. LNCS, vol. 4794, pp. 108–123. Springer, Heidelberg (2007), http://dx.doi.org/10.1007/978-3-540-76652-0_7
5. Bottou, L.: Stochastic gradient learning in neural networks. In: Proceedings of Neuro-Nımes 1991, vol. 8 (1991)
6. Cristian, I.T.: The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters 85(6), 317–325 (2003)
7. Cui, X., Potok, T., Palathingal, P.: Document clustering using particle swarm optimization. In: Proceedings 2005 IEEE Swarm Intelligence Symposium, SIS 2005, pp. 185–191 (June 2005)
8. Gori, M., Tesi, A.: On the problem of local minima in backpropagation. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(1), 76–86 (1992)
9. Grzenda, M.: On the prediction of floor identification credibility in RSS-based positioning techniques. In: Ali, M., Bosse, T., Hindriks, K.V., Hoogendoorn, M., Jonker, C.M., Treur, J. (eds.) IEA/AIE 2013. LNCS, vol. 7906, pp. 610–619. Springer, Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-38577-3_63
10. Jacobs, R.A.: Increased rates of convergence through learning rate adaptation. Neural Networks 1(4), 295–307 (1988)
11. Jing, Y.W., Ren, T., Zhou, Y.C.: Neural network training using pso algorithm in atm traffic control. In: Huang, D.S., Li, K., Irwin, G. (eds.) Intelligent Control and Automation. LNCIS, vol. 344, pp. 341–350. Springer, Heidelberg (2006), http://dx.doi.org/10.1007/978-3-540-37256-1_41
12. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks. IV, pp. 1942–1948 (1995)
13. Lakmali, B., Dias, D.: Database Correlation for GSM Location in Outdoor and Indoor Environments. In: 4th International Conference on Information and Automation for Sustainability, pp. 42–47 (2008)

14. Okulewicz, M., Mańdziuk, J.: Application of Particle Swarm Optimization Algorithm to Dynamic Vehicle Routing Problem. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2013, Part II. LNCS, vol. 7895, pp. 547–558. Springer, Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-38610-7_50
15. Otsason, V., Varshavsky, A., LaMarca, A., de Lara, E.: Accurate GSM Indoor Localization. In: Beigl, M., Intille, S.S., Rekimoto, J., Tokuda, H. (eds.) UbiComp 2005. LNCS, vol. 3660, pp. 141–158. Springer, Heidelberg (2005), http://dx.doi.org/10.1007/11551201_9
16. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2012) ISBN 3-900051-07-0, http://www.R-project.org/
17. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. SIGGRAPH Comput. Graph. 21(4), 25–34 (1987), http://doi.acm.org/10.1145/37402.37406
18. Sabak, G.: api.orange.pl tutorial for building localization applications. Tech. rep., Orange (2012) (in polish), http://telco21.pl/orange-celli/
19. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of IEEE International Conference on Evolutionary Computation, pp. 69–73 (1998)
20. Shi, Y., Eberhart, R.: Parameter selection in particle swarm optimization. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, pp. 591–600. Springer, Heidelberg (1998)
21. Vogl, T.P., Mangis, J., Rigler, A., Zink, W., Alkon, D.: Accelerating the convergence of the back-propagation method. Biological Cybernetics 59(4-5), 257–263 (1988)
22. Wilson, D.R., Martinez, T.R.: The need for small learning rates on large problems. In: Proceedings of the International Joint Conference on Neural Networks, IJCNN 2001, vol. 1, pp. 115–119 (2001)
23. Yu, X.H., Chen, G.A.: Efficient backpropagation learning using optimal learning rate and momentum. Neural Networks 10(3), 517–527 (1997)