Basis Functions Comparative Analysis in Consecutive Data Smoothing Algorithms

F.D. Tarasenko and D.A. Tarkhov(✉)

Saint-Petersburg Polytechnic University of Peter the Great,
Saint-Petersburg, Russia
oudi@mail.ru, dtarkhov@gmail.com

Abstract. In this paper, we investigate algorithms for constructing experimental data dependence based on sequential processing of the points one by one. Four algorithms are reviewed, comparative analysis for different basis functions, a level of noise and other options is made. In addition to static data, there was an investigation of dynamic data case. The sine with variable frequency is used as an approximative function. Numerical experiments led to the conclusions about the comparative efficiency of algorithms and basis functions. The recommendations for the use of the algorithms are given.

Keywords: Consecutive algorithms · Data processing · RBF network · Spline

1 Introduction

Problems of high neural network training costs stimulate the search for ways to accelerate this process. The main directions of the acceleration are

(1) Finding a good initial approximation to the required weights of the neural network, which allows to significantly accelerate global optimization algorithms for functional errors.
(2) Parallelization of the learning process that allows you to accelerate it, using multiprocessor computer systems and graphics cards.
(3) The use of specialized processors.

The work deals with algorithms to realize the first two approaches. If we solve the problem of constructing RBF-network on data set, you can use the algorithms discussed below to quickly build a good approximation, using equidistant from the fixed centers and the wide range of Gaussian basis functions. Further, it is possible to clarify the position of the centers, and the width of the basis functions are weighting coefficients using any global optimization algorithm. Less smooth functions can be used in the first step. In the next step, using a pre-built approximation of basis functions and scaling operations can be obtained neural network approximation with the required activation functions - Gaussians, sigmoid, etc.

Algorithms considered in the work allow for effective parallelization. Thus, procedures for distribution of the points for intervals and updates the weights of the basis functions can be separated. Wherein, the interval, into which the new point may be

L. Cheng et al. (Eds.): ISNN 2016, LNCS 9719, pp. 482–489, 2016.
DOI: 10.1007/978-3-319-40663-3_55

determined simultaneously with the specification of the weights caused by the previous point. Since the basis functions are non-zero only in a small neighborhood center, it is possible to update the weight of several functions simultaneously and in parallel. Another option is to parallelize the simultaneous use of the considered algorithms and evolutionary algorithm of neural network training. This updated with the help of another portion of the data processing network includes an evolving population.

Separately mention studied the possibility of applying the algorithms we have considered building a model for dynamic data. The data obtained from the function, which varies with time. This situation is common to observe the object whose behavior changes in the process of building the model. The possibility of restructuring the model in accordance with these changes.

These algorithms can be used in hybrid neural network procedures for constructing approximate solutions of differential equations. In the first phase of operation of such procedures by the classical method of finite difference constructed approximation on points, the second - with the help of the algorithms considered in this paper the neural network approximation is built, on the third - it is specified by the methods discussed in [8–10].

Algorithms of consecutive data smoothing were studied in several works [11, 12].

In this paper, we review the methods [1–3] of finding dependence $y = f(x)$ on experimental data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$ in the situation when the points (x_i, y_i) are received and processed one by one, which may be associated with the need to process data in real-time. Let the required dependence provide in view if $y = \sum_{j=1}^{n} c_j \varphi_j(x)$, where $\varphi_j = \varphi(\alpha_j |x - z_j|)$ or $\varphi_j = \varphi(\alpha_j (x - z_j)^2)$.

Such types of functions are called RBF-nets [3–5]. Finding parameters c_j, α_j and z_j is called network learning. In this paper considers the case, where only the parameter c_j is finding. We set other parameters ourselves.

As it known, every piecewise linear function can be represented in the sum form $\sum_{j=1}^{n} c_j \varphi(\alpha_j (x - z_j))$, if we chose triangular cap as basis function (Fig. 1).

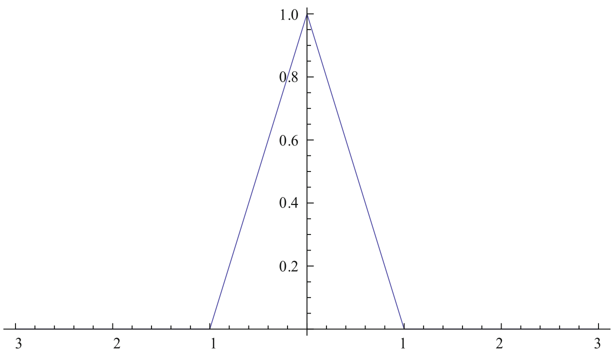

Fig. 1. The plot of the triangular cap.

$$y = \varphi(x) = (1 - |x|)_+ = \begin{cases} 1 - |x|, & |x| \le 1 \\ 0, & |x| > 1 \end{cases}$$

In a such way any spline can be decomposed in the sum of basis functions, which match to its degree of smoothness [6].

In the numerical experiments we used following basis splines [6]: parabola, cubic parabola, gaussian.

We studied several algorithms for data smoothing:

(1) Processing points one by one with the adaptation of the weights of the basis functions with the nearest center.
(2) same, but with the adaptation of the weights of the two nearest functions.
(3) same as (2) but assumes a linear dependence of the speed of adaptation of the weights from the distance of the abscissa of a point to the center of the basis function [1].
(4) Finding the optimal coefficients of decomposition in basis functions the solution of the linear system.

For approaches 1 and 3 was also studied the variation of the algorithm with a previously determined law of variation of step.

In approaches 1–3 as the basis for adaptation of the coefficients is the minimization of a quadratic functional of the error.

2 Approaches

2.1 Approach 1

The initial values of coefficients $c_j = 0$. In the step from N-1 point to N the changing of weight coefficient of the basis function with the nearest center, written as $\Delta_k(N)$, is equal

$$\Delta_k(N) = \frac{Q_k}{S_k}, \tag{1}$$

where

$$S_k = \sum\nolimits_{i=1}^{N} \varphi_k^2(x_i), Q_k = \sum\nolimits_{i=1}^{N} \varphi_k(x_i)\delta_i(N) \tag{2}$$

Moving to the next step we need to recalculate the error according to the formula:

$$\delta_i(N+1) = \delta_i(N) - \Delta_k \varphi_k(x_i). \tag{3}$$

2.2 Approach 2

Unlike the first approach, we select two basis functions between which centers is the received point.

The changing of weights is made according to the formulas

$$\Delta_k = \frac{Q_k S_{k+1} - Q_{k+1} P_k}{S_k S_{k+1} - P_k^2}, \Delta_{k+1} = \frac{Q_{k+1} S_k - Q_k P_k}{S_k S_{k+1} - P_k^2} \tag{4}$$

where

$$P_k = \sum\nolimits_{i=1}^{N} \varphi_k(x_i)\varphi_{k+1}(x_i). \tag{5}$$

When the denominator of (2) turns to the zero we use formulas (7) and (8). Moving to the next step the terms in the sum (2) and (5) are added for all basis functions, for which $\varphi_k(x_i) \neq 0$.

Error δ_i when moving to the next step you need to recalculate according to the formula:

$$\delta_i(N+1) = \delta_i(N) - \Delta_k \varphi_k(x_i) - \Delta_{k+1}\varphi_{k+1}(x_i) \tag{6}$$

2.3 Approach 3

Let x_N is between the centers of the functions with numbers k and k + 1. In previous notation we use the formula [1]:

$$\Delta_k(N) = \Delta(N)\lambda, \Delta_{k+1}(N) = \Delta(N)(1-\lambda), \tag{7}$$

where $\lambda = \frac{z_{k+1} - x_N}{z_{k+1} - z_k}$ [1].

We find the number $\Delta_k(N)$ by the minimization of the error function and so it is equal

$$\Delta(N) = \frac{\lambda Q_k + (1-\lambda)Q_{k+1}}{\lambda^2 S_k + 2\lambda(1-\lambda)P_k + (1-\lambda)^2 S_{k+1}}. \tag{8}$$

Moving to the next step we need to recalculate an error according to the formula Error when moving to the next step you need to recalculate according to the formula

$$\delta_i(N+1) = \delta_i(N) - \Delta(N)\left(\lambda\varphi_k(x_i) + (1-\lambda)\varphi_{k+1}(x_i)\right). \tag{9}$$

For approaches 1 and 3 decreasing by a given law $\Delta_k(N)$ can be used instead of formulas (1), (7) and (8), for example, $\Delta(N) = \left(1 - \frac{2}{T}\right)^N \delta_k(N)$, where $T = 5n + N_{max}$ [1]. This significantly reduces the amount of computation.

2.4 Approach 4

This approach involves the construction of linear regression and involves solving systems of linear equations, which can be obtained by a recurrent way when we get a new point [3, 4, 7].

3 Results of Numerical Experiments

Comparative testing of algorithms of the approaches 1–4 for the above basis functions for different values of the error dimensions, the number of points and number of basis functions was done. Also, the cases of dynamic (variable in time) data were analyzed at the different sets of parameters and the choices of basis functions. The interval of the argument [0;1]. Below are illustrations of the algorithms results, some of the most characteristic data of numerical experiments and conclusions obtained in the analysis of the results of these experiments. The first picture shows the result of processing experimental dependence, obtained by generate points around the function sin(πx) + 0.1 sin(10πx), using the first algorithm and the formulas from [1] to 20 basis functions, a 400-hundred experimental points, the error of "measurement" – 0.1 and 10 runs of the algorithm.

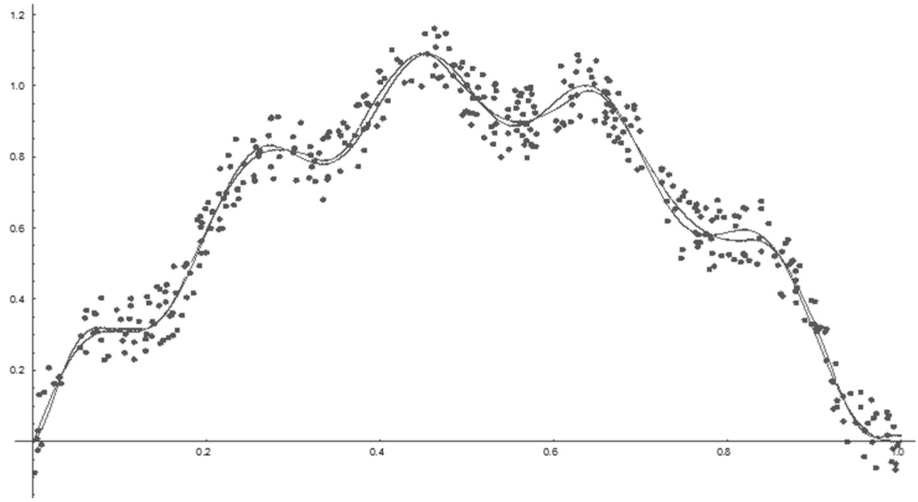

Fig. 2. The result of the first algorithm for static data.

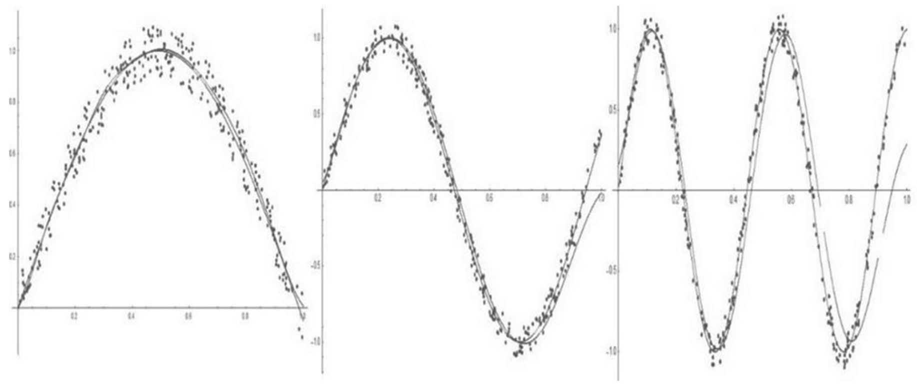

Fig. 3. Results of the second approach for dynamic data.

Figure 3 shows the results of applying this second approach to dynamic data. We approximate the function sin(wπx), where w varies over the computational time from 1 to 4.5. Basis function - cubic parabola. 300 experimental points. 10 basis functions. Measurement error - 0,1. There are three moments in which the original function has the forms, respectively: sin(πx), sin(2,25 πx), sin(πx 4,5) (Fig. 3, Table 2).

Table 1. Mathematical expectation of error for the eight tests of the algorithm with 50 points. Static data. Approximative function - sin(πx). The number of basis functions - 10. The number of experimental points - 50.

ε	*m*	Approach	Triangular cap	Parabola	Cubic parabola	Gaussian
0.01	10	1 E.	0.0045	0.0359	0,0251	0,0337
1	10		0.2880	0.3115	0,0404	0,0308
0.01	1		0.0247	0.0363	0,0536	0,0209
0.01	10	1 A.	0,0047	0,0050	0,0056	0,3342
1	10		0,0320	0,0406	0,0361	0,0355
0.01	1		0,1888	0,1126	0,1130	0,1313
0.01	10	2	0,0039	0,0039	0,0054	0,0038
1	10		0,0331	0,0330	0,0281	0,0343
0.01	1		0,0057	0,0150	0,0195	0,0146
0.01	10	3 E.	0,0132	0,0125	0,0238	0,0217
1	10		0,0292	0,0378	0,0387	0,0398
0.01	1		0,0464	0,0403	0,0365	0,0487
0.01	10	3 A.	0,0044	0,0044	0,0053	0,0422
1	10		0,0278	0,0342	0,0358	0,0579
0.01	1		0,1016	0,0975	0,0945	0,1041
0.01		4	0,0031	0,0035	0,0040	0,0080
1			0,2838	0,2542	0,2633	0,2901

Here and next, ε - error of observations, m - the number of iterations of an algorithm, n - the number of points, A. is for adjustable step in accordance with the following formula from [1], E. is for exact step in accordance with the formulas given above for each algorithm.

Table 2. Mathematical expectation of error for the four tests of the algorithm. Dynamic data. Approximative function - sin(wπx), where w varies over the computational time from 1 to 2 and up to 4.5. The number of basis functions is 10. Measurement error is 0.1.

m	*n*	*w*	Approach	Triangular cap	Parabola	Cubic parabola	Gaussian
100	300	4,5	1	0,4252	0,4084	0,4060	0,4360
50	100	2		0,1908	0,1910	0,1755	0,1811
100	300	4,5	2	0,2880	0,2624	0,2481	0,2348
50	100	2		0,1454	0,1361	0,1321	0,1233
100	300	4,5	3	0,4176	0,4296	0,4422	0,4767
50	100	2		0,1854	0,1815	0,1779	0,1817

4 Conclusions

(1) Reviewed algorithms and basis functions showed good performance in the considered task.
(2) None of the algorithms does not have a decisive advantage over others. This allows us to recommend the simplest of them – the first.
(3) The advantages of exact computation of step over the formula from [1] could not compensate for a substantial increase in computational complexity, if the number of points is large enough, or duplicated as necessary.
(4) For large errors and a small number of experimental points, all the methods are unsatisfactory. Some better than others, the results of smoothing using cubic basis function for a large number of points and the gaussian for small.
(5) The results we got using different basis functions approximately the same. The choice of specific function is dictated by conditions on the smoothness. If such conditions exist, then the preferred is a triangular function because of the minimality of computational complexity. If these conditions are not known in advance or sufficiently rigid, it is preferable gaussian, as having infinite smoothness.
(6) The best of the considered algorithms for dynamic data smoothing is approach 2, as it handles much better the end of the interval.

The paper is based on research carried out with the financial support of the grant of the Russian Science Foundation (Project No. 14-38-00009, the program-targeted management of the Russian Arctic zone development). Peter the Great St. Petersburg Polytechnic University.

References

1. Hakimov B.V.: Modeling Correlation Dependences with Splines on the Examples in Geology and Ecology, p. 144. Moscow State University, Neva, St. Petersburg (2003). (in Russian)
2. Hakimov, B.V., Mikheev, M.I.: Nonlinear model of a neuron - multi-dimensional spline. Neurocomputers Dev. Appl. **7**, 36–40 (2012). (in Russian)
3. Tarkhov, D.A.: Consecutive algorithms for data smoothing. Neurocomputers Dev. Appl. **3**, 11–18 (2015). (in Russian)
4. Tarkhov, D.A.: Neural network models and algorithms. In: Radio Engineering, Moscow, p. 352 (2014). (in Russian)
5. Haykin, S.: Neural Networks: A Comprehensive Foundation, p. 823. Prentice Hall, Upper Saddle River (1999)
6. Svinyin, S.F.: Basis splines in the theory of samples of signals. Science 118 (2003). St. Petersburg (in Russian)
7. Albert, A.: Regression and the Moor-Penrose Pseudoinverse, p. 179. Academic Press, New York (1972)
8. Vasilyev, A.N., Tarkhov, D.A.: Mathematical models of complex systems on the basis of artificial neural networks. Nonlinear Phenom. Complex Syst. **17**, 327–335 (2014)

9. Kainov, N.U., Tarkhov, D.A., Shemyakina, T.A.: Application of neural network modeling to identification and prediction in ecology data analysis for metallurgy and welding industry. Nonlinear Phenom. Complex Syst. **17**, 57–63 (2014)
10. Lazovskaya, T.V., Tarkhov, D.A.: Fresh approaches to the construction of parameterized neural network solutions of a stiff differential equation. St. Petersburg Polytechnical Univ. J. Phys. Math. **1**, 192–198 (2015)
11. Ueno, T., Kawanabe, M., Mori, T., Maeda, S., Ishii, S.: A semiparametric statistical approach to model-free policy evaluation. In: Proceedings of the 25th International Conference on Machine Learning, pp. 1072–1079 (2008)
12. Bottou, L., Lecun, Y.: On-line learning for very large datasets. Appl. Stoch. Models Bus. Ind. **21**, 137–151 (2005)