Applied Mathematical Modelling 37 (2013) 5027–5038

Contents lists available at SciVerse ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Utilizing feed-back neural network approach for solving linear Fredholm integral equations system

A. Jafarian *, S. Measoomy Nia

Department of Mathematics, Urmia Branch, Islamic Azad University, Urmia, Iran

ARTICLE INFO

Article history:
Received 26 November 2011
Received in revised form 3 August 2012
Accepted 10 September 2012
Available online 20 September 2012

Keywords:
Fredholm integral equation systems
Feed-back neural networks (FNN)
Cost function
Approximate solution
Learning algorithm

ABSTRACT

This paper intended to offer an architecture of artificial neural networks (NNs) for finding approximate solution of a second kind linear Fredholm integral equations system. For this purpose, first we substitute the N-th truncation of the Taylor expansion for unknown functions in the origin system. By applying the suggested neural network for adjusting the real coefficients of given expansions in resulting system. The proposed NN is a two-layer feed-back neural network such that it can get a initial vector and then calculates it's corresponding output vector. In continuance, a cost function is defined by using output vector and the target outputs. Consequently, the reported NN using a learning algorithm that based on the gradient descent method, will adjust the coefficients in given Taylor series. Eventually, we have showed this method in comparison with existing numerical methods such as trapezoidal quadrature rule provides solutions with good generalization and high accuracy. The proposed method is illustrated by several examples with computer simulations.

1. Introduction

Integral equation methods are very useful for solving many problems in several applied fields like mathematical economics and optimal control theory. Because these problems are often reduced to integral equations. Since these equations usually can not be solved explicitly, so it is necessary to get different numerical techniques. There are numerous advanced and efficient methods which have been focusing on the solution of integral equations. First time, Taylor expansion approach was presented for solution of integral equations by Kanwal and Liu in [1] and then has been extended in [2–9]. Also Variational iteration method [10], homotopy analysis method [11–17], Legendre polynomial method [18,19] and Adomian decomposition method [20] are efficacious techniques. Additionally, some other numerical methods applied to solve the integral equations in [21–26]. In the most of the introduced approaches the integral equation is transformed to a linear system. But unfortunately in some cases linear system corresponding to the integral equation problem is ill-conditioned problem. Hence we have to use other valid methods.

In this paper we want to propose a new numerical approach to approximate solution of a linear Fredholm integral equations system of the second kind. For this aim, first the N-th truncation of the Taylor expansion for unknown function $F_i(t)$ is substituted instead of the unknown function in the given integral equation system. Then we use a two-layer feed-back neural network, where it can get a input vector and calculates its corresponding output vector. The outputs from this neural network are numerically compared with target outputs. Next a cost function is defined that measures the difference between the target output and corresponding actual output. Then the suggested neural net by using a learning algorithm that based on the gradient descent method adjusts the real connection weights to any desired degree of accuracy. Here is an outline of

* Corresponding author. Tel.: +98 04413662872; fax: +98 04413453371.
E-mail addresses: jafarian5594@yahoo.com (A. Jafarian), measoomy@yahoo.com (S. Measoomy Nia).

0307-904X/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.apm.2012.09.029

the paper. In Section 2, the basic notations and definitions of integral equation and Taylor polynomial method are briefly presented. Section 3 describes how to find a approximate solution of the given Fredholm integral equations system by using NN. Finally, the reliability of the FNN method versus trapezoidal quadrature rule is checked to the integral equations system in Section 4.

2. Preliminaries

In this section we give a detailed study of integral equations and Taylor expansion which are used in the next sections.

2.1. Integral equation

The basic definition of integral equation is given in [27,28].

Definition 1. The Fredholm integral equation of the second kind is

$$F(t) = f(t) + \lambda(ku)(t), \tag{1}$$

where

$$(ku)(t) = \int_a^b k(s,t)F(s)ds, \quad a \leqslant t \leqslant b.$$

In Eq. (1), $k(s,t)$ is an arbitrary kernel function over the square $a \leqslant s, t \leqslant b$ and $f(t)$ is a function of $t : a \leqslant t \leqslant b$. If the kernel function satisfies $k(s,t) = 0$, $s > t$, we obtain the Volterra integral equation

$$F(t) = f(t) + \lambda \int_a^t k(s,t)F(s)ds. \tag{2}$$

2.2. Linear Fredholm integral equations system

A system of linear Fredholm type integral equations of the second kind is in the following general form

$$\begin{cases} \sum_{i=1}^{n} A_{1i}(t)F_i(t) = f_1(t) + \int_a^b \left(\sum_{i=1}^{n} k_{1i}(s,t) \cdot F_i(s)\right) ds \\ \vdots, \\ \sum_{i=1}^{n} A_{ji}(t)F_i(t) = f_j(t) + \int_a^b \left(\sum_{i=1}^{n} k_{ji}(s,t) \cdot F_i(s)\right) ds \\ \vdots, \\ \sum_{i=1}^{n} A_{ni}(t)F_i(t) = f_n(t) + \int_a^b \left(\sum_{i=1}^{n} k_{ni}(s,t) \cdot F_i(s)\right) ds, \end{cases} \tag{3}$$

where $t, s \in [a,b]$ and $A_{pi}(t), k_{pi}(s,t)$ are real functions. Moreover, in Eq. (3) the function $A_{pi}(t)$ and the kernel $k_{pi}(s,t)$ are given and assumed to be sufficiently differentiable with respect to all their arguments on the interval $a \leqslant t, s \leqslant b$. Also, $F(t) = [F_1(t), \ldots, F_n(t)]$ is the vector solution where to be determined.

2.3. Taylor series

Let us first recall the basic principles of the Taylor polynomial method for solving Fredholm integral equations [1]. Since these results are the key for our problems therefore we explain them. For simplify, first we assume that $n = 2$. In this special case the Eq. (3) can be summarized as following form

$$\begin{cases} A_{11}(t)F_1(t) + A_{12}(t)F_2(t) = f_1(t) + \int_a^b (k_{11}(s,t)F_1(s) + k_{12}(s,t)F_2(s))ds, \\ A_{21}(t)F_1(t) + A_{22}(t)F_2(t) = f_2(t) + \int_a^b (k_{21}(s,t)F_1(s) + k_{22}(s,t)F_2(s))ds, \end{cases} \tag{4}$$

where the function $f_p(t)$ and the kernel $k_{pi}(s,t)$ have been given and $F(t) = [F_1(t), F_(t)]$ has to be evaluated. To obtain the solution of the given system in the form of

$$F_{pN}(t) = \sum_{i=0}^{N} \left(\frac{1}{i!} \cdot F_p^{(i)}(c) \cdot (t-c)^i\right), \; a \leqslant t, c \leqslant b, \; p = 1, 2, \tag{5}$$

which is the Taylor polynomial of degree N at $t = c$, we first differentiate each equation of system (4) N times with respect to t and get

$$\sum_{i=0}^{j}\left(\binom{j}{i}.A_{p1}^{(j-i)}(t)\cdot F_1^{(i)}(t)\right)+\sum_{i=0}^{j}\left(\binom{j}{i}\cdot A_{p2}^{(j-i)}(t)\cdot F_2^{(i)}(t)\right)=f_p^{(j)}(t)+\int_a^b\left(\frac{\partial^{(j)}k_{p1}(s,t)}{\partial t^j}.F_1(s)+\frac{\partial^{(j)}k_{p2}(s,t)}{\partial t^j}.F_2(s)\right)ds,\ p=1,2;j=0,\dots,N. \tag{6}$$

The aim of this paper is determining of the coefficients $F_p^{(i)}(c), (p = 1, 2;\ i = 0, \dots, N)$ in Eq. (5). For this aim, we expanded $F_p(s)$ in Taylor series at $c = 0$ and substituted its N-th truncation in (6). Without loss generality we assume that in system (3), $a = 0$. Because in the otherwise with doing change variable x = s-a the lower bound of the integral equations in (3) transformed to 0. Now we can write:

$$\sum_{i=0}^{j}(T_{p1}(j,i)\cdot F_1^{(i)}(0))+\sum_{i=0}^{j}(T_{p2}(j,i)\cdot F_2^{(i)}(0))=f_p^{(j)}(0)+\sum_{i=0}^{N}(R_{p1}(j,i)\cdot F_1^{(i)}(0))+\sum_{i=0}^{N}(R_{p2}(j,i)\cdot F_2^{(i)}(0)), \tag{7}$$

where

$$T_{p1}(j,i)=\frac{j!}{i!(j-i)!}.A_{p1}^{(j-i)}\Big|_{t=0},$$

$$T_{p2}(j,i)=\frac{j!}{i!(j-i)!}.A_{p2}^{(j-i)}\Big|_{t=0},$$

$$R_{p1}(j,i)=\frac{1}{i!}\cdot\int_a^b\left(\frac{\partial^{(j)}k_{p1}(s,t)}{\partial t^j}\Big|_{t=0}.s^i\right)ds,$$

and

$$R_{p2}(j,i)=\frac{1}{i!}\cdot\int_a^b\left(\frac{\partial^{(j)}k_{p2}(s,t)}{\partial t^j}\Big|_{t=0}.s^i\right)ds,\ \ p=1,2;\ j=0,\dots,N.$$

The applied approach for determining the coefficient $F_p^{(i)}(0)$ in Eq. (7) will be described in section 3.

3. Learning of neural network

This section, first gives a short review on feed-back neural networks (FNNs) and then will suggest a learning algorithm to obtain an approximate solution of the given integral equations system.

3.1. Input–output relations of each unit

Now consider a two-layer FNN with $(2N + 2)$ input units and $(N + 1)$ output units such that all input–output signals and connection weights be real numbers. Let us assume that two vectors $X_1 = (X_{10}, X_{11}, \dots, X_{1N})$ and $X_2 = (X_{20}, X_{21}, \dots, X_{2N})$ be initial values for the unknown coefficients $F_1^{(i)}(0)$ and $F_2^{(i)}(0)$ $(i = 0, \dots, N)$ in (7), respectively. Now by presenting these values to our FNN, the input–output relation of each unit can be written as follows:

Input units:

The input neurons make no change in their inputs, so:

$$o_{ri} = X_{ri},\ \ r = 1, 2;\ \ i = 0, 2, \dots, N. \tag{8}$$

Output units:

$$Y_j = f(net_j),$$

$$net_j=\sum_{i=0}^{N}w_{ji}.o_{1i}+\sum_{i=0}^{N}w'_{ji}\cdot o_{2i},\quad j=0,1,\dots,N.$$

where X_{ri} is real number. In above equation w_{ji} (w'_{ji}) denotes the connection weight from the input signal X_{1i} (X_{2i}) to the jth output unit and $f(x)$ is identical activation function corresponding to output nodes (see Fig. 1).

3.2. Cost function

Let us define the matrices X_{ri}, w_{ji}, w'_{ji} and target vector $B_p = (B_{po}, \dots, B_{pN})$ as following:

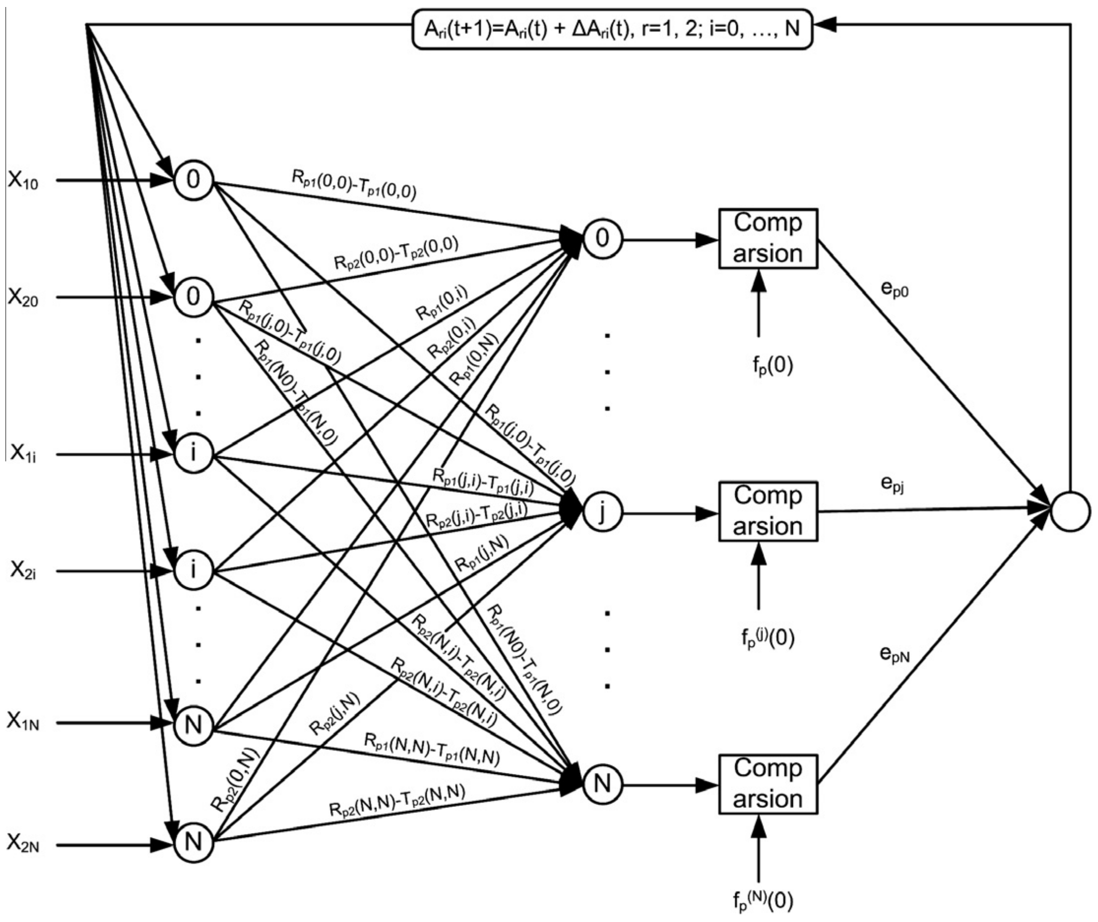

Fig. 1. The proposed FNN.

$$w_{ji} = \begin{cases} R_{p1}(j,i), & i > j, \\ R_{p1}(j,i) - T_{p1}(j,i), & i \leqslant j, \end{cases} \quad w'_{ji} = \begin{cases} R_{p2}(j,i), & i > j, \\ R_{p2}(j,i) - T_{p2}(j,i), & i \leqslant j, \end{cases}$$

and

$$X_{ri} = F_r^{(i)}(0), \;\; B_{pj} = -f_p^{(j)}(0), \;\; p, r = 1,2; \; j, i = 0, \ldots, N.$$

Suppose that $Y_p = (Y_{p0}, \ldots, Y_{pN})$ be the output vector corresponding to the input vectors X_1 and X_2. Now we want to deduce a learning algorithm with using of the cost function which has been defined for the calculated output vector Y_p and the target vector B_p, as follows:

$$e_{pj} = \frac{(B_{pj} - Y_{pj})^2}{2}, \; j = 0, \ldots, N. \tag{9}$$

In general, the cost function for the given FNN is obtained as:

$$e_p = \sum_{j=0}^{N} e_{pj}. \tag{10}$$

3.3. Learning algorithm of the FNN

Let real quantities X_{ri} $(r = 1,2; \; i = 0, \ldots, N)$ are initialized at random values for input signals. We want to update the crisp parameter X_{ri} such that $X_{ri} = F_r^{(i)}(0)$. At first we calculate the matrices w, w' and Y_p $(p = 1,2)$ by using the relations that were described at first of this section. For crisp parameter X_{ri} adjustment rule can be written as follows:

$$X_{ri}(n+1) = X_{ri}(n) + \Delta X_{ri}(n), \tag{11}$$

$$\Delta X_{ri}(n) = -\eta.\frac{\partial e_p}{\partial X_{ri}} + \alpha.\Delta X_{ri}(n-1), \ r = 1,2; \ i = 0,\ldots,N, \tag{12}$$

where n is the number of adjustments, η is the learning rate and α is the momentum term constant. Thus our problem is to calculate the derivative $\frac{\partial e_p}{\partial X_{ri}}$ in (12). The derivative $\frac{\partial e_p}{\partial X_{ri}}$ can be calculated from the cost function e_p in (10). We calculated $\frac{\partial e_p}{\partial X_{ri}}$ as follows:

$$\frac{\partial e_p}{\partial X_{ri}} = \frac{\partial e_{p1}}{\partial X_{ri}} + \cdots + \frac{\partial e_{pN}}{\partial X_{ri}}, \ r = 1,2; \ i = 0,\ldots,N. \tag{13}$$

In other hand

$$\frac{\partial e_{pj}}{\partial X_{ri}} = \frac{\partial e_{pj}}{\partial Y_{pj}} \cdot \frac{\partial Y_{pj}}{\partial net_{pj}} \cdot \frac{\partial net_{pj}j}{\partial X_{ri}} = (f_p^{(j)}(0) + Y_{pj}) \cdot \frac{\partial net_{pj}}{\partial X_{ri}}, \ j = 0,\ldots,N,$$

where

$$\frac{\partial net_{pj}}{\partial X_{ri}} = \begin{cases} R_{pr}(j,i), & i > j, \\ R_{pr}(j,i) - T_{pr}(j,i), & i \leqslant j. \end{cases}$$

Now the learning algorithm can be summarized as following:

Learning algorithm

Step 1: $\eta > 0$, $\alpha > 0$ and $Emax > 0$ are chosen. Then crisp quantities X_{ri} $(r = 1,2; \ i = 0,\ldots,N)$ are initialized at random values.

Step 2: Let $n := 0$ where n is the number of iterations of the learning algorithm. Then the running error E is set to 0.

Step 3: Let $n := n + 1$. Repeat below procedure for $p = 1,2$:

(i) Forward calculation: Calculate the output vector Y_p by presenting the input vectors X_1 and X_2.
(ii) Back-propagation: Adjust crisp parameter X_{ri} using the cost function (10) and Eqs. (11) and (12).

Step 4: Cumulative cycle error is computed by adding the present error to E.

Step 5: The training cycle is completed. For $E < Emax$ terminate the training session. If $E > Emax$ then E is set to 0 and we initiate a new training cycle by going back to *Step 3*.

4. Numerical examples

This section contains three examples of linear Fredholm integral equations system of second kind. In these examples, we illustrate the use of the FNN technique to approximate solutions of the present systems. For each example, the computed values of the approximate solution are calculated over a number of iterations and the cost function is plotted. Also the present method is compared with trapezoidal quadrature rule (TQR). In the following simulations, we use the specifications as follows:

1. Learning rate $\eta = 0.01$.
2. Momentum constant $\alpha = 0.003$.
3. Stoping conditions: $Emax < 0.001$.

Example 4.1. Consider the following integral equations system:

$$\begin{cases} 2F_1(t) + 3F_2(t) = f_1(t) + \int_0^1 \ (t+s)F_1(s)ds + \int_0^1 \ tsF_2(s)ds \\ 3F_1(t) - 4F_2(t) = f_2(t) + \int_0^1 \ (2ts-1)F_1(s)ds + \int_0^1 \ (t-s)F_2(s)ds \end{cases}$$

with

$$f_1(t) = 2e^t - \frac{3}{t-2} - t(e-1) - t(ln^4 - 1) - 1,$$

$$f_2(t) = e - 2t + ln^4 + 3e^t + \frac{4}{t-2} - t(ln^2) - 2.$$

where the exact solution is $F_1(t) = e^t$ and $F_2(t) = \frac{1}{2-t}$. In this example, we illustrate the use of the FNN technique to approximate the solution of this integral equations system. For this case, we used a polynomial of degree 3. Before starting calculations we assumed that $F_r^{(i)}(0) = 0.5$ $(r = 1,2; \ i = 0,\ldots,3)$. Then we have:

$$F_1(t) = \frac{1}{12}t^3 + \frac{1}{4}t^2 + \frac{1}{2}t + \frac{1}{2},$$

Table 1
The approximated solutions with error analysis for Example 4.1.

n	$F_{1n}(0)$	$F_{2n}(0)$	e_n
1	(0.587 0.541 0.581 0.573)	(0.420 0.418 0.406 0.437)	9.6915
2	(0.654 0.572 0.645 0.634)	(0.369 0.360 0.340 0.394)	5.7070
3	(0.705 0.597 0.697 0.684)	(0.339 0.318 0.295 0.365)	3.5497
4	(0.745 0.616 0.739 0.726)	(0.322 0.288 0.264 0.347)	2.3458
5	(0.777 0.632 0.774 0.761)	(0.314 0.266 0.243 0.336)	1.6471
⋮	⋮	⋮	⋮
49	(0.998 0.979 0.996 0.998)	(0.490 0.247 0.246 0.373)	0.01392
50	(0.998 0.980 0.997 0.998)	(0.491 0.247 0.247 0.373)	0.01287
51	(0.999 0.980 0.997 0.999)	(0.491 0.248 0.249 0.374)	0.01189
52	(0.999 0.980 0.998 0.999)	(0.492 0.248 0.249 0.374)	0.01098
53	(1.000 0.980 0.998 0.999)	(0.492 0.248 0.249 0.374)	0.01015
54	(1.000 0.981 0.999 0.999)	(0.492 0.248 0.249 0.374)	0.00938

$$F_2(t) = \frac{1}{12}t^3 + \frac{1}{4}t^2 + \frac{1}{2}t + \frac{1}{2}.$$

After 54 iterations, the approximate functions $F_1(t)$ and $F_2(t)$ estimated as:

$$F_1(t) = 0.16636t^3 + 0.49756t^2 + 0.93039t + 0.98445,$$

$$F_2(t) = 0.06235t^3 + 0.12381t^2 + 0.23793t + 0.46931.$$

Numerical result can be found in Table 1. Figs. 2 and 3 show the convergence behaviors for computed values of the vectors $F_{1n}^{(i)}(0)$ and $F_{2n}^{(i)}(0)$ where n is index of iterations. The exact solution and the approximated solution are compared in Fig. 4.

Trapezoidal rule: Moreover, this example is going to show difference between the FNN approach and the trapezoidal quadrature rule. Consider again example 1, the trapezoidal quadrature procedure is as follows:

Let the region of integration is subdivided into two equal intervals of width $h = \frac{1}{2}$, three integration nodes $t_i = \frac{(i-1)}{2}$ and $f_i = f(t_i)$ (for $i = 1, \ldots, 3$). After using the TQR for this integral equation, following relations are derived:

$$\begin{cases} 2F_1(t_i) + 3F_2(t_i) = f_1(t_i) \\ \quad + \frac{1}{4}\left(t_i.F_1(0) + (2t_i+1)F_1(\frac{1}{2}) + (t_i+1)F_1(1)\right) + \frac{1}{4}\left(t_i.F_2(\frac{1}{2}) + t_i.F_2(1)\right) \\ 3F_1(t_i) - 4F_2(t_i) = f_2(t_i) + \frac{1}{4}\left(-F_1(0) + (2t_i-2)F_1(\frac{1}{2}) + (2t_i-1)F_1(1)\right) \\ \quad + \frac{1}{4}\left(t_i.F_2(0) + (2t_i-1)F_2(\frac{1}{2}) + (t_i-1)F_2(1)\right) \end{cases}.$$

These relations are transformed to system of linear equations

$$\begin{bmatrix} 2 & -\frac{1}{4} & -\frac{1}{4} & 3 & 0 & 0 \\ -\frac{1}{8} & \frac{3}{2} & -\frac{3}{8} & 0 & \frac{23}{8} & -\frac{1}{8} \\ -\frac{1}{4} & -\frac{3}{4} & \frac{3}{2} & 0 & -\frac{1}{4} & \frac{11}{4} \\ \frac{13}{4} & \frac{1}{2} & \frac{1}{4} & -4 & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{13}{4} & 0 & -\frac{1}{8} & -4 & \frac{1}{8} \\ \frac{1}{4} & 0 & \frac{11}{4} & -\frac{1}{4} & -\frac{1}{4} & -4 \end{bmatrix} \begin{bmatrix} F_1(0) \\ F_1(\frac{1}{2}) \\ F_1(1) \\ F_2(0) \\ F_2(\frac{1}{2}) \\ F_2(1) \end{bmatrix} = \begin{bmatrix} \frac{5}{2} \\ \frac{2197}{677} \\ \frac{1333}{250} \\ \frac{742}{239} \\ \frac{243}{80} \\ \frac{1291}{362} \end{bmatrix}.$$

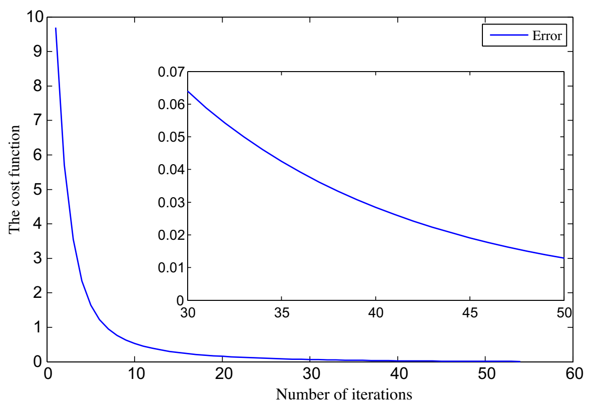

Fig. 2. The cost function for Example 4.1 on the number of iterations.

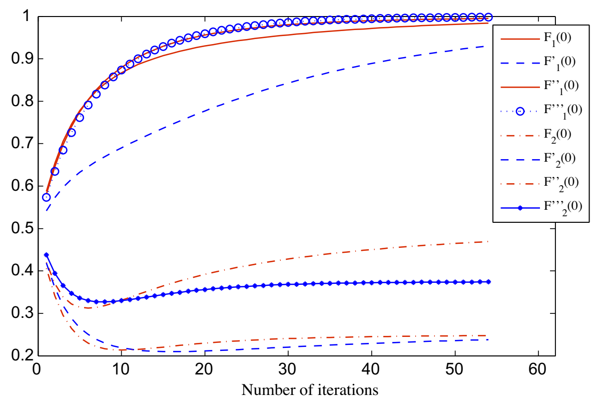

Fig. 3. Convergence of the approximated solution for Example 4.1.

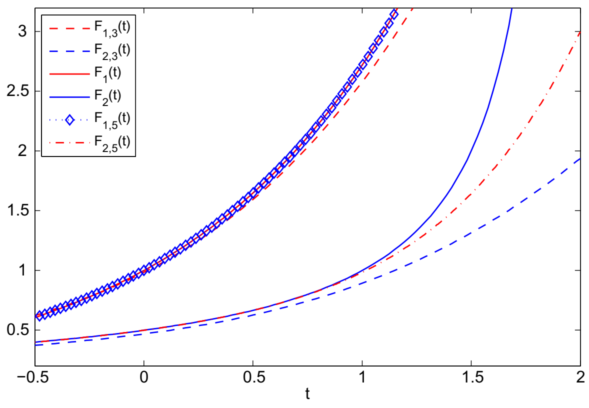

Fig. 4. Comparison between the numerical position solution and exact one for Example 4.1.

Table 2
Comparison the exact and approximate solutions for Example 4.1.

t_j	Exact	FNN	TQR	t_j	Exact	FNN	TQR
$F_1(t)$				$F_2(t)$			
0	1.0000	1.0010	1.0070	0	0.5000	0.4922	0.53744
0.5	1.6487	1.6364	1.6973	0.5	0.6667	0.6557	0.6978
1	2.7183	2.6476	2.8084	1	1.0000	0.9284	1.0249

The solution of above system can be calculated by using a standard method, therefore we can write:

$$\begin{bmatrix} F_1(0) \\ F_1(\frac{1}{2}) \\ F_1(1) \\ F_2(0) \\ F_2(\frac{1}{2}) \\ F_2(1) \end{bmatrix} = \begin{bmatrix} \frac{1007}{1000} \\ \frac{3269}{1926} \\ \frac{1070}{381} \\ \frac{445}{828} \\ \frac{1924}{2757} \\ \frac{1029}{1004} \end{bmatrix}.$$

Comparison the exact solution and the approximate solutions at nodes t_i are collected in Table 2.

It is clear that to get the best approximating solutions for unknown functions, the truncation limit N must be chosen large enough.

Example 4.2. Consider the following system of Fredholm integral equations:

Table 3
The approximated solutions with error analysis for Example 4.2.

n	$F_{1n}(0)$	$F_{2n}(0)$	e_n
1	(2.9337 2.0515 0.1038)	(4.5337 1.9393 0.3333)	144.500
2	(2.9663 2.1908 0.0732)	(4.6141 1.9603 0.3242)	2.09265
3	(2.9701 2.1952 0.0704)	(4.6555 2.0008 0.3283)	0.38395
4	(2.9707 2.1905 0.0683)	(4.6673 2.0216 0.3300)	0.36510
5	(2.9714 2.1859 0.0654)	(4.6770 2.0424 0.3312)	0.34826
⋮	⋮	⋮	⋮
132	(2.9459 2.0472 0.0388)	(5.0523 2.9077 −0.03967)	0.00010
133	(2.9461 2.0476 0.0389)	(5.0522 2.9080 −0.03963)	0.00010
134	(2.9463 2.0481 0.0390)	(5.0522 2.9083 −0.03959)	0.00010
135	(2.9465 2.0485 0.0391)	(5.0521 2.9086 −0.03955)	0.00010
136	(2.9467 2.049 0.0391)	(5.0520 2.9089 −0.03952)	0.00010
137	(2.9468 2.0494 0.0392)	(5.0520 2.9092 −0.03948)	0.00009

$$\begin{cases} F_1(t)+F_2(t)=f_1(t)+\int_0^1((t-2s)F_1(s)+(t-ts)F_2(s))ds, \\ F_1(t)+F_2(t)=f_2(t)+\int_0^1((2t+s)F_1(s)+(t^2-s^2)F_2(s))ds, \end{cases}$$

where

$$f_1(t)=\frac{9t}{2}+\frac{35}{6},$$

$$f_2(t)=-\frac{26t^2}{4}-3t+\frac{33}{4},$$

and the exact solution is $F_1(t)=2t+3$ and $F_2(t)=3t+5$. In this example, we illustrate the use of FNN technique to approximate the solution of this integral equations system. For this case, we used a polynomial of degree 2. We trained FNN with 6 input units and 3 output units. Taking $F_1^{(0)}(0)=2$, $F_1^{(1)}(0)=1$, $F_1^{(2)}(0)=0.5$, $F_2^{(0)}(0)=4$, $F_2^{(1)}(0)=2$ and $F_2^{(2)}(0)=0.5$ we have:

$$F_1(t)=2+t+\frac{1}{4}t^2 \text{ and } F_2(t)=4+2t+\frac{1}{4}t^2.$$

After 137 iterations the approximate functions $F_1(t)$ and $F_2(t)$ transformed to follow functions:

$$F_1(t)=-0.0581t^2+2.0119t+3.0119,$$
$$F_2(t)=0.17811t^2+2.9675t+4.9768.$$

Table 3 shows the approximated solution over a number of iterations and Figs. 5–7 show the convergence behaviors for computed values of the vectors $F_{1n}^{(i)}(0)$ and $F_{2n}^{(i)}(0)$ for different numbers of iterations.

From Table 3 and Fig. 7 we can conclude that the exact and approximate solutions are equal in interval $[0,1]$.

Trapezoidal rule: For this example the trapezoidal rule with three integration nodes can be written as follows:

$$\begin{cases} F_1(t_j)+F_2(t_j)=f_1(t_j)+\frac{1}{4}\big(t_j.F_1(0)+2(t_j-1)F_1(\frac{1}{2})+(t_j-2)F_1(1)\big)+\frac{1}{4}(t_j.F_2(0)+t_j.F_2(\frac{1}{2})) \\ F_1(t_j)+F_2(t_j)=f_2(t_j)+\frac{1}{4}(2t_j.F_1(0)+2(2t_j+\frac{1}{2})F_1(\frac{1}{2})+(2t_j+1)F_1(1)) \\ \quad +\frac{1}{4}(t_j^2.F_2(0)+2(t_j^2-\frac{1}{4})F_2(\frac{1}{2})+(t_j^2-1)F_2(1)), \quad j=1,\ldots,3. \end{cases}$$

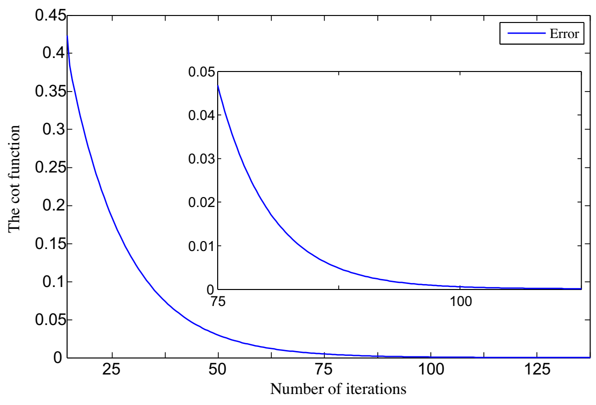

Fig. 5. The cost function for Example 4.2 on the number of iterations.

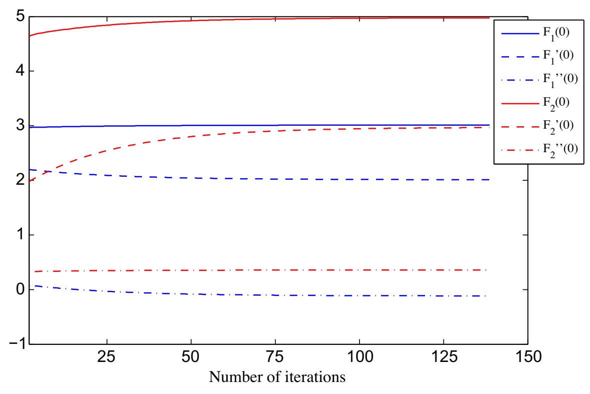

Fig. 6. Convergence of the approximated solution for Example 4.2.

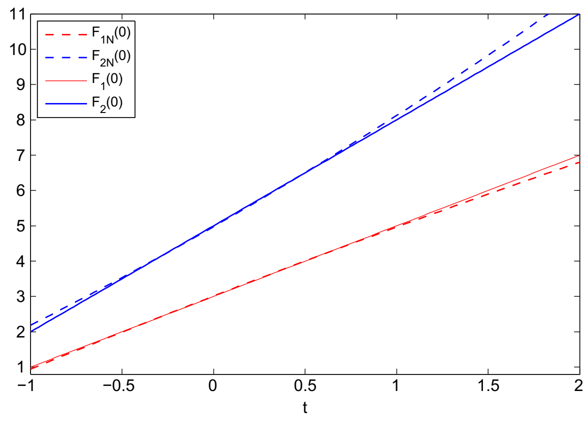

Fig. 7. Comparison between the numerical position solution and exact one for Example 4.2.

Table 4
Comparison the exact and approximate solutions for Example 4.2.

t_j	*Exact*	*FNN*	*TQR*	t_j	*Exact*	*FNN*	*TQR*
$F_1(t)$				$F_2(t)$			
0	3.0000	2.9468	2.7625	0	5.0000	5.0520	5.1274
0.5	4.0000	3.9764	3.6946	0.5	6.5000	6.5017	6.3255
1	5.5000	5.0158	5.2524	1	8.0000	7.9415	7.8711

Comparison the exact solution and the approximate solutions at nodes ti are collected in Table 4.

Example 4.3. Let system of integral equations

$$\begin{cases} (2t+8)F_1(t)+(5t^2-4)F_3(t)=f_1(t)+\int_0^2(2t+4s)F_1(s)ds+\int_0^2(s-t^2)F_2(s)ds+\int_0^2 2tsF_3(s)ds \\ (t^2+4)F_1(t)+tF_2(t)=f_2(t)+\int_0^2(2t^2+3s)F_1(s)ds+\int_0^2(s^2-5t)F_2(s)ds+\int_0^2 5tsF_3(s)ds \\ 2tF_1(t)+(4t-3)F_2(t)+(t^2-5t)F_3(t)=f_3(t)+\int_0^2(5ts+4t^2)F_1(s)ds+\int_0^2(s+t)F_3(s)ds, \end{cases}$$

with

$$f_1(t)=\frac{1}{3}(30t^4+3t^3+156t^2-92t-154),$$

$$f_2(t)=\frac{1}{3}(9t^4+39t^2+49t-168),$$

$$f_3(t)=\frac{1}{3}(6t^4-15t^3+87t^2-187t-67).$$

Table 5
The approximated solutions with error analysis for Example 4.3.

n	$F_{1n}(0)$	$F_{2n}(0)$	$F_{3n}(0)$	e_n
1	$(-8.68\ \ -2.71\ \ 0.62)$	$(2.84\ \ 3.88\ \ 1.62)$	$(5.21\ \ 0.14\ \ 1.07)$	9430
2	$(-4.64\ \ -1.40\ \ 2.21)$	$(2.54\ \ 4.47\ \ 0.28)$	$(3.29\ \ -1.84\ \ 0.70)$	2963
3	$(-3.68\ \ -1.62\ \ 3.34)$	$(1.70\ \ 4.45\ \ -0.65)$	$(4.80\ \ -0.76\ \ 1.67)$	926.8
4	$(-2.63\ \ -0.65\ \ 4.45)$	$(2.34\ \ 5.14\ \ -0.40)$	$(4.56\ \ -1.28\ \ 1.21)$	280.1
5	$(-2.82\ \ -0.86\ \ 4.65)$	$(2.04\ \ 5.02\ \ -0.56)$	$(5.23\ \ -0.44\ \ 1.81)$	185.0
⋮	⋮	⋮	⋮	⋮
183	$(-4.14\ \ 0.14\ \ 5.96)$	$(3.05\ \ 5.15\ \ -0.57)$	$(3.82\ \ -0.63\ \ 3.55)$	0.010
184	$(-4.16\ \ 0.15\ \ 5.97)$	$(3.06\ \ 5.16\ \ -0.56)$	$(3.83\ \ -0.63\ \ 3.56)$	0.010
185	$(-4.17\ \ 0.16\ \ 5.98)$	$(3.06\ \ 5.16\ \ -0.55)$	$(3.84\ \ -0.62\ \ 3.57)$	0.010
186	$(-4.18\ \ 0.17\ \ 5.98)$	$(3.07\ \ 5.17\ \ -0.55)$	$(3.84\ \ -0.62\ \ 3.57)$	0.009
187	$(-4.19\ \ 0.17\ \ 5.99)$	$(3.07\ \ 5.17\ \ -0.54)$	$(3.85\ \ -0.61\ \ 3.58)$	0.009
188	$(-4.19\ \ 0.17\ \ 5.99)$	$(3.08\ \ 5.18\ \ -0.54)$	$(3.85\ \ -0.61\ \ 3.58)$	0.008

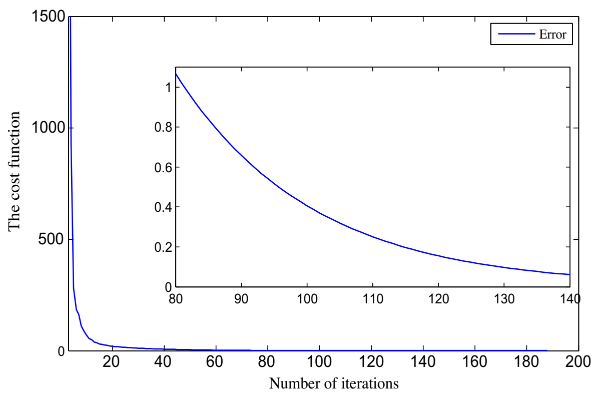

Fig. 8. The cost function for Example 4.3 on the number of iterations.

In addition the exact solution of the above system is, $F_1(t) = 3t^2 - 4, F_2(t) = 5t + 3$ and $F_3(t) = 2t^2 - t + 4$. We trained the neural network as described in last example. Before starting calculations, we assumed that the preliminary functions are:

$$F_1(t) = t^2 + t - 3,\ \ F_2(t) = \frac{3}{2}t^2 + t + 4 \ \text{ and } \ F_3(t) = 2\ t.$$

After 188 iterations the above approximate functions transformed to follow functions:

$$F_1(t) = 2.9907t^2 + 0.17879t - 4.1916,$$
$$F_2(t) = -0.2899t^2 + 5.1595t - 3.0609,$$
$$F_3(t) = 1.8868t^2 - 0.82262t - 3.8363.$$

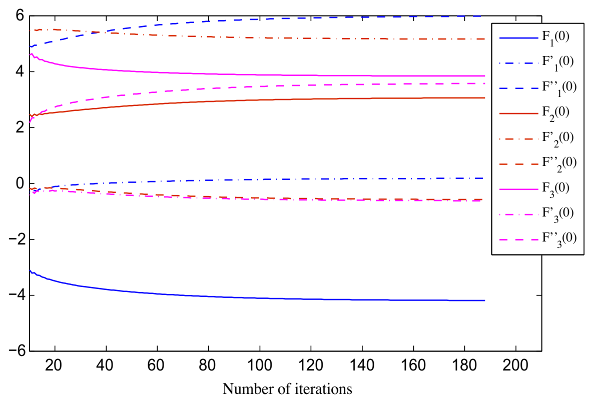

Fig. 9. Convergence of the approximated solution for Example 4.3.

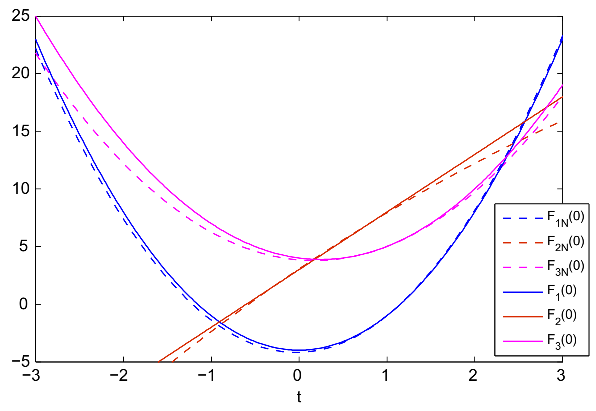

Fig. 10. The comparison between numerical position solution and exact one for Example 4.3.

Table 6
Comparison the exact and approximate solutions for Example 4.3.

t_j	*Exact*	*FNN*	*TQR*	t_j	*Exact*	*FNN*	*TQR*	t_j	*Exact*	*FNN*	*TQR*
$F_1(t)$				$F_2(t)$				$F_3(t)$			
0	−4.0000	−4.1900	−3.7822	0	3.0000	3.0800	3.1213	0	4.0000	3.8500	3.7923
1	−1.0000	−1.0250	−1.1527	1	8.0000	7.9900	8.1128	1	5.0000	5.0300	4.7486
2	8.0000	8.1300	8.1827	2	13.0000	12.6600	13.4500	2	10.0000	9.7900	10.4641

Numerical result can be found in Table 5. Similarly Figs. 8–10 show the accuracy of the solution or the convergence behaviors for computed values of the vectors $F^{(i)}_{1n}(0)$, $F^{(i)}_{2n}(0)$ and $F^{(i)}_{3n}(0)$ where n is index of iterations.

From the numerical examples we can conclude that with using of this method we can find the analytical solution for this kind of equations system, if the exact solution of given problem be a polynomial.

Trapezoidal rule: Similarly, the trapezoidal rule with 3 integration nodes has been applied for the present example (Tables 6). After solving the result linear system from the TQR approach, following numerical results have been acquired.

5. Conclusions

In this paper, an architecture of feed-back neural networks has been proposed for approximating solution of the Fredholm integral equations system. The presented FNN in this study is a method for computing the coefficients in the Taylor expansions of linear integral equations solutions. To get the best approximating solution of the equation, the truncation limit N must be chosen large enough. An interesting feature of this method is finding the analytical solution for the integral equation, if the exact solution of this equation was a polynomial of degree N or less than N. With the availability of this methodology, now it will be possible to investigate the approximate solution of other kinds of integral equations. Additionally, the FNN method has been compared with trapezoidal quadrature rule. The analyzed examples illustrate the ability and reliability of the present method versus the TQR approach. The obtained solutions, in comparison with exact solutions admit a remarkable accuracy.

References

[1] R.P. Kanwal, K.C. Liu, A Taylor expansion approach for solving integral equations, Int. J. Math. Educ. Sci. Technol. 20 (1989) 411–414.
[2] M. Gulsu, M. Sezer, The approximate solution of high order linear difference equation with variable coefficients in terms of Taylor polynomials, Appl. Math. Comput. 168 (2005) 76–88.
[3] K. Maleknejad, N. Aghazadeh, Numerical solution of Volterra integral equations of the second kind with convolution kernel by using Taylor-series expansion method, Appl. Math. Comput. 161 (2005) 915–922.
[4] S. Nas, S. Yalnba, M. Sezer, A Taylor polynomial approach for solving high-order linear Fredholm integrodifferential equations, Int. J. Math. Educ. Sci. Technol. 31 (2000) 213–225.
[5] N. Sezer, Taylor polynomial solution of Volterra integral equations, Int. J. Math. Educ. Sci. Technol. 25 (1994) 625–633.
[6] M. Sezer, A method for approximate solution of the second order linear differential equations in terms of Taylor polynomials, Int. J. Math. Educ. Sci. Technol. 27 (1996) 821–834.
[7] M. Sezer, M. Gulsu, A new polynomial approach for solving difference and Fredholm integro-difference equations with mixed argument, Appl. Math. Comput. 171 (2004) 332–344.
[8] S. Yalçinbaş, Taylor polynomial solutions of nonlinear VolterraFredholm integral equations, Appl. Math. Comput. 127 (2002) 195–206.
[9] S. Yalçinbaş, M. Sezer, The approximate solution of high-order linear Volterra Fredholm integro-differential equations in terms of Taylor polynomials, Appl. Math. Comput. 112 (2000) 291–308.

[10] X. Lan, Variational iteration method for solving integral equations, Comput. Math. Appl. 54 (2007) 1071–1078.
[11] M. El-Shahed, Application of He's homotopy perturbation method to Volterra's integro-differential equation, Int. J. Non. Sci. Num. Simul. 6 (2005) 163–168.
[12] M. Ghasemi, M. Tavassoli Kajani, E. Bobolian, Numerical solutions of the nonlinear VolterraFredholm integral equations by using homotopy perturbation method, Appl. Math. Comput. 188 (2007) 446–449.
[13] A. Golbabai, B. Keramati, Modified homotopy perturbation method for solving Fredholm integral equations, Chaos Soliton Fract. (2006), http://dx.doi.org/10.1016/j.chaos.2006.10.037.
[14] S.J. Liao, Beyond Perturbation: Introduction to the Homotopy Analysis Method, Chapman Hall/CRC Press, Boca Raton, 2003.
[15] S.J. Liao, On the homotopy analysis method for nonlinear problems, Appl. Math. Comput. 147 (2004) 499–513.
[16] S.J. Liao, Notes on the homotopy analysis method: some definitions and theorems, Communications in Nonlinear Science and Numerical Simulation (2008), http://dx.doi.org/10.1016/j.cnsns.2008.04-013.
[17] S.J. Liao, Y. Tan, A general approach to obtain series solutions of nonlinear differential equations, Stud. Appl. Math. 119 (2007) 297–355.
[18] S. Yalçinbaş, M. Sezer, Sezer, H. Hilmi Sorkun, Legendre polynomial solutions of high-order linear Fredholm integro-differential equations, Appl. Math. Comput. 210 (2009) 334–349.
[19] N. Bildik, A. Konuralp, S. Yalçinbaş, Comparison of Legendre polynomial approximation and variational iteration method for the solutions of general linear Fredholm integro-differential equations, Comput. Math. Appl. 59 (2010) 1909–1917.
[20] S. Abbasbandy, Numerical solution of integral equation: Homotopy perturbation method and Adomian's decomposition method, Appl. Math. Comput. 173 (2006) 493–500.
[21] C.T.H. Baker, A perspective on the numerical treatment of Volterra equations, J. Comput. Appl. Math. 125 (2000) 217–249.
[22] K. Holmaker, Global asymptotic stability for a stationary solution of a system of integro-differential equations describing the formation of liver zones, SIAM J. Math. Anal. 24 (1993) 116–128.
[23] K. Maleknejad, F. Mirzae, S. Abbasbandy, Solving linear integro-differential equations system by using rationalized Haar functions method, Appl. Math. Comput. 155 (2004) 317–328.
[24] K. Maleknejad, M. Tavassoli Kajani, Solving linear integro-differential equation system by Galerkin methods with hybrid functions, Appl. Math. Comput. 159 (2004) 603–612.
[25] K. Maleknejad, M. Shahrezaee, H. Khatami, Numerical solution of integral equation system of the second kind by Block-pulse function, Appl. Math. Comput. 1 (2005) 11–24.
[26] A. Tahmasbi, O.S. Fard, Numerical solution of linear Volterra integral equations system of the second kind, Appl. Math. Comput. 201 (2008) 547–552.
[27] L.M. Delves, J.L. Mohamed, Computational methods for integral equations, Cambridge University Press, Cambridge, 1988.
[28] S. Effati, R. Buzhabadi, A neural network approach for solving Fredholm integral equations of the second kind, Neural Comput. Appl. (2010), http://dx.doi.org/10.1007/s00521-010-0489-y.339-355.