Ibn Al-Haitham Journal for Pure and Applied Science | مجلة إبن الهيثم للعلوم الصرفة و التطبيقية
No. 3 Vol. 25 Year 2012 | 2012 السنة 25 المجلد 3 العدد

Design Feed Forward Neural Network to Determine Doses of the Decongestant for Cold Pills

L. N. M. Tawfiq and Y. R. B. Al-Mayouf*
Department of Mathematics, College of Education Ibn-Al-Haitham, University of Baghdad
***Department of computer science, College of Education Ibn-Al-Haitham, University of Baghdad**

Received in: 30 May 2012 Accepted in: 24 September 2012

Abstract

The aim of this paper is to design feed forward neural network to determine the effects of cold pills and cascades from simulation the problem to system of first order initial value problem. This problem is typical of the many models of the passage of medication throughout the body. Designer model is an important part of the process by which dosage levels are set. A critical factor is the need to keep the levels of medication high enough to be effective, but not so high that they are dangerous.

Keyword: Ann**,** feed forward neural network, propagation training algorithm.

Introduction

The study of cold pills based on the work of the contemporary applied mathematician, Edward Spitznagel Professor of Mathematics at Washington University, St. Louis, Missouri [1]. In this paper we design feed forward neural network(FFNN) to determine the doses of decongestant since FFNNs are parallel computational models comprised of densely interconnected, simple, adaptive processing units, characterized by an inherent propensity for storing experiential knowledge and rendering it available for use, FFNNs resemble the human brain in two fundamental respects; Firstly, knowledge is acquired by the network from its environment through a learning process, and secondly, interneuron connection strengths, known as synaptic weights are employed to store the acquired knowledge[2]. In the proposed approach the model function is expressed as the sum of two terms: the first term satisfies the initial conditions (IC) and contains no adjustable parameters. The second term can be found by using feed forward neural network(FFNN) which is trained so as to satisfy the differential equation and such technique is called collocation neural network. Since it is known that a multilayer FFNN with one hidden layer can approximate any function to arbitrary accuracy [3],[4] thus our FFNN contains one hidden layer. Now, we illustrate how our approach can be used to find the approximate solution of the system of first order differential equation:

$y_i'(x) = F_i(x, y_1, y_2)$, $i = 1,2$, (1)

where a subject to certain IC's and $x \in R$, $D \subset R^n$ denotes the domain and y(x) is the solution to be computed.

If $y_t(x, p)$ denotes a trial solution with adjustable parameters p, the problem is transformed to a discretize form:

$$\text{Min}_p \sum_{\vec{x}_i \in \hat{D}} F(x_i, y_{t1}(x_i, p), y_{t2}(x_i, p)) \quad , \quad (2)$$

subject to the constraints imposed by the IC's.

In the proposed approach, the trial solution y_t employs a FFNN and the parameters p correspond to the weights and biases of the neural architecture. We choose a form for the trial

function $y_t(x)$ such that it satisfies the IC's. This is achieved by writing it as a sum of two terms:

$$y_t(x_i , p) = A(x) + G(x, N(x, p)) \quad , \quad (3)$$

where N(x, p) is a single output FFNN with parameters p and n input units fed with the input vector x. The term A(x) contains no adjustable parameters and satisfies the IC's. The second term G is constructed so as not to contribute to the IC's, since $y_t(x)$ satisfy them. This term can be formed by using a FFNN whose weights and biases are to be adjusted in order to deal with the minimization problem.

Computation of the Gradient

An efficient minimization of (2) can be considered as a procedure of training the FFNN, where the error corresponding to each input vector x_i is the value E (x_i) which has to force near zero. Computation of this error value involves not only the FFNN output but also the derivatives of the output with respect to any of its inputs. Therefore, in computing the gradient of the error with respect to the network weights consider a multi layer FFNN with n input units (where n is the dimensions of the domain) one hidden layer with H sigmoid units and a linear output unit .

For a given input vector $x = (x_1, x_2, \ldots, x_n)$ the output of the FFNN is :

$$N = \sum_{i=1}^{H} v_i \sigma(z_i), \text{ where } z_i = \sum_{j=1}^{n} w_{ij} x_j + b_i$$

w_{ij} denotes the weight connecting the input unit j to the hidden unit i
v_i denotes the weight connecting the hidden unit i to the out put unit ,
b_i denotes the bias of hidden unit i , and
σ (z) is the sigmoid transfer function (tanhsig.).
The gradient of FFNN, with respect to the parameters of the FFNN can be easily obtained as:

$$\frac{\partial N}{\partial v_i} = \sigma (z_i) \quad , \quad (4)$$

$$\frac{\partial N}{\partial b_i} = v_i \sigma'(z_i) \quad , \quad (5)$$

$$\frac{\partial N}{\partial w_{ij}} = v_i \sigma'(z_i)\, x_j \quad , \quad (6)$$

Once the derivative of the error with respect to the network parameters has been defined, then it is a straight forward to employ any minimization technique**.** It must also be noted, the batch mode of weight updates may be employed.

Cold Pills and Cascades

At the first sign of a cold many of us start to take cold pills. These pills usually contain a decongestant to relieve stuffiness. The pill dissolves in the gastrointestinal tract, and the two medications diffuse into the bloodstream. The bloodstream takes the medications to the sites where they have therapeutic effect. Both medications are eventually removed from the blood by the kidneys and the liver. The dynamics of the rising and falling levels of the medications in the GI tract and in the bloodstream may be modeled by system of first-order linear differential equations. Pharmaceutical companies do extensive testing to determine the movement medications through the body. This process is modeled by treating various components of the human body as compartments and following the medication as it enters and leaves these compartments. Examples of body compartments are the GI tract, the bloodstream, the tissues and the excretory system. Testing has shown that a typical cold medication leaves one compartment (e.g., the GI tract) and moves into another (e.g., the bloodstream) at a rate proportional to the amount present in the first compartment. The

coefficient of proportionality depends upon the specific medication, the compartments involved, and the age and general health of the individual. In this section we look at three models based on different dosage strategies suppose that a single dose of a fast-dissolving cold pill is taken. The pill dissolves "instantaneously" in the GI tract and releases A milligrams of decongestant. Each medication independently diffuses into the bloodstream.

One Instantly Dissolving Dose: The Model IVP

Let $y_1(t)$be the amount of medicine of decongestant in the GI tract at time t after the pill has dissolved. We shall use the Balance Law to compute the rate of change of the amount in each compartment. Because the medication moves out of the GI tract and into the blood-stream at a rate proportional to the amount in the GI tract, and because nothing is coming in, we have that:

$$\frac{dy_1(t)}{dt} = -k_1 y_1(t), \quad y_1(0) = A$$

where $k_1 > 0$ is the coefficient of proportionality, and A is the initial amount the units for the rate constant k_1 are reciprocals of the units chosen to measure the time t. For example the units of k_1 are $(\text{hours})^{-1}$ if time is given in hours.

The level of medication in the bloodstream will build up from zero (initially, there are no cold medications in the blood), but then will fall as the kidneys and liver do their job of removing "foreign" substances from the blood. If $y_2(t)$ denotes the amount of medication in the bloodstream at time t, then the Balance Law implies that :

$$\frac{dy_2(t)}{dt} = k_1 y_1(t) - k_2 y_2(t), \qquad y_2(0) = 0$$

The first term on the right side of the rate equation models the observed fact that the medication leaving the GI tract goes directly into the bloodstream ; the second rate term models the clearance of medication from the blood and into the excretory system.

Consequently, the system of first-order ODEs and flow of medication is given by the IVP :

$$\frac{dy_1}{dt} = -k_1 y_1 \quad , \quad y_1(0) = A$$

$$\frac{dy_2}{dt} = k_1 y_1 - k_2 y_2, \qquad y_2(0) = 0 \qquad \ldots\ldots\ldots (7)$$

where the (positive) coefficient k_2 is typically smaller than k_1.

System (7) models the flow of decongestant. However, the flow rates differ because the experimentally determined values of the rate constants k_1 and k_2 are quite different for the two medications (see Table 1). These differences lead to very different levels of the two medications in the bloodstream.

Observe from table (1) how much smaller k_2 is than k_1. Consequently, the medication will stay in the bloodstream longer than in the GI tract.

Solving The Model IVP

Professor Edward Spitznagel in [1] solved this problem analytically and obtain the following solution for $k_1 \neq k_2$:

$$y_1(t) = Ae^{-k_1 t} \quad , \quad y_2(t) = \frac{k_1 A}{k_1 - k_2}(e^{-k_2 t} - e^{-k_1 t}) \quad \ldots\ldots \quad (8)$$

where time t is measured forward from the instant the initial dose of A units of medication is released into the GI tract.

From the solution set (8) see that, as expected, the levels of the medication in the GI tract and the bloodstream tend to zero as time increases. Since, the medication is initially at the zero level in the bloodstream, these levels reach a maximum value at some later time.

Now, we solve system(7) for A=1 where k_1 = 1.386, k_2 = 0.1386 using three layers feed forward neural network with back propagation training algorithm having one input unit, five hidden units with tanhsig. transfer function and one linear output unit. Table (2) gives the analytic and neural solutions with different training algorithm such as : Levenberg – Marquardt (trainlm) [5], and quasi – Newton (trainbfg)[6] and its errors given in table (3), table(4) gives the weight and bias of the designer network and table(5) gives the performance of the train with epoch and time

Figures (1) show the levels of decongestant in the GI tract and the blood over a six-hour period if the values of k_1 and k_2 given in Table (1) are used and A = 1.

The clearance coefficient k_2 of medication from the blood-stream is often much lower for old and sick people than it is for the young and healthy. This means that for some people the medication levels in the blood may become dangerously high even with a "standard" dosage.

Few of us stop with taking one cold pill. The models that follow heat the more realistic setting of taking medication over a period of time.

Continuous Doses

If a medication is embedded in resins that dissolve at varying rates, a fixed flow of medication can be assured. Tiny beads of the mixture are packed into a capsule, and the medications are released at a constant rate into the CI I tract over a period of hours. One capsule of continuous-acting medication such as CONTAC (Smith-Kline Beecham) may need to be taken only twice a day. The mathematical model for this situation is slightly different from IVP (7). There is no medication in the GI tract at the start of the continuous dosage regimen, but instead there is a constant inflow rate I due to the gradually dissolving beads.

Continuous-Acting Capsules: Model IVP and its Solution

The model IVP for the amounts $y_1(t)$and $y_2(t)$ of continuous-acting medication in the GI tract and bloodstream, respectively, is :

$$\frac{dy_1}{dt} = I - k_1 y_1 \quad , \quad y_1(0) = 0$$

$$\frac{dy_2}{dt} = k_1 y_1 - k_2 y_2 \quad , \quad y_2(0) = 0 \qquad \ldots\ldots \quad (9)$$

Where I is a positive constant modeling the rate of release of each medication into the GI tract .

The system of IVP (9) solved analytically by Professor Edward Spitznagel in [1] and he found the following solution :

$$y_1(t) = \frac{I}{k_1}(1 - e^{-k_1 t})$$

$$y_2(t) = \frac{I}{k_2}\left[1 + \frac{1}{k_1 - k_2}(k_2 e^{-k_1 t} - k_1 e^{-k_2 t})\right] \quad \ldots\ldots\ldots \quad (10)$$

Observe that as time increases, the level of medication in the GI tract tends to I / k_1 and that in the bloodstream to I / k_2 . The solution formulas (10) are valid for any positive values of the coefficients k_1 and k_2 , $k_1 \neq k_2$.
Now, we solve system (9) where $k_1 = 1.386$, $k_2 = 0.1386$ and $I = 12$, using FFNN .
Table (6) gives the analytic and neural solutions with different training algorithm such as: Levenberg–Marquardt (trainlm), and quasi–Newton (trainbfg) and its errors given in table (7), table(8) gives the weight and bias of the designer network and table(9) gives the performance of the train with epoch and time .

Suppose that the input rate I is taken to be one unit of each medication per hour. Thus, a total dose of twenty-four units is administered continuously over a period of one day. Figures (2) show that, if the values of k_1 and k_2 from Table (1) are used, the decongestant builds up quite rapidly to its equilibrium level in the blood.

Repeated Doses of Fast-Dissolving Pills

Many cold pills dissolve fairly quickly in the GI tract, releasing their medications over a period of no more than half an hour. Repeated doses must then be taken every four to six hours in order to keep up the medication levels in the blood. In this case, the input rate I into the GI tract is not constant, but operates at a high level for a short time, then cuts off entirely until the next dose is taken four to six hours later.

One Dose livery Six Hours: The Model IVP

In contrast to a continuous acting medication, other forms of medications dissolve very rapidly in the GI tract. This means that the medication is delivered at a high constant rate but only over a short time. Suppose that the rate is 12 units / hour, but delivered in half an hour and then repeated at six-hour intervals. The model IVP is given below:

$$\frac{dy_1}{dt} = I(t) - k_1 y_1, \qquad y_1(0) = 0$$

$$\frac{dy_2}{dt} = k_1 y_1 - k_2 y_2, \qquad y_2(0) = 0$$

I(t) = 12 sqw (t , 12 / 100 , 6)
where sqw (t , 12/100, 6) denotes the pulse function with period 6 hours, which is "on" for half an hour at the start of each period and " off " otherwise (1/2 hour = 12/100 % of the 6-hour period). See [7] for a description of sqw and other piecewise continuous functions.

The total dosage of 24 units in one day is the same as in the continuous- acting model .Figures 2 display the medication levels corresponding to this repeated on-off dosage pattern over a forty – eight hour period if values of k_1 and k_2 are taken from Table 1, using FFNN. Medication levels rise during the half hour the cold pill is dissolving in the GI tract, but then fall until the next dose. The amount of decongestant in the blood quickly reaches and then oscillates around an equilibrium level.

References

1. Robert L. B. and Courtney, S. C. (1996) Differential Equations A Modeling perspective; United States of America.
2. Galushkin I. A.(2007) "Neural Networks Theory", Berlin Heidelberg.

3. Tawfiq, L.N.M. and Naoum, R.S. (2007) " Density and approximation by using feed forward Artificial neural networks ", Ibn Al-Haitham Journal for Pure & Applied Sciences, Vol. 20 (1).
4. Jabber A. K. (2009)," On Training Feed Forward Neural Networks for Approximation Problem ", MSc Thesis, Baghdad University, College of Education (Ibn Al-Haitham).
5. Tawfiq, L. N. M. and Oraibi, Y. A.(2012) Improve Levenberg-Marquardt Training Algorithm For Feed Forward Neural Networks, Second Conference on Computer and Information Technology/College of Computer – University of Anbar.
6. Stanevski, N. and Tsvetkov, D. (2004) "On the Quasi-Newton Training Method for Feed-Forward Neural Networks", International Conference on Computer System and Technologies.
7. Abd-Al-Razak H. A. (2010) Solution of a System of Ordinary Initial Value Problems using Semi-Analytic Method, MSc Thesis, Baghdad University, College of Education (Ibn Al - Haitham).

Tabel (1) : Rate Coefficients of Decongestant for Cold Medications

k_1 $1.386\ hour^{-1}$
k_2 $0.1386\ hour^{-1}$

Table (2):The analytic & neural result of the problem (7)

x	Analytic (y_{1a}) solution	Neural solution (Trainlm) (y_{1t1})	Neural solution (Trainbfg)(y_{1t2})
0	1	1	0.999837751002654
0.1	0.870576189297161	0.870645179547517	0.870576189297933
0.2	0.757902901371166	0.757902901371166	0.757902901371998
0.3	0.659812219732971	0.659812219732971	0.659810076684370
0.4	0.574416807906831	0.574424576246922	0.574416254623237
0.5	0.500073595695768	0.500073595695768	0.500073595698388
0.6	0.435352165308950	0.435344727835639	0.435352165307395
0.7	0.379007229076934	0.379002100324034	0.379007285987416
0.8	0.329954669205873	0.329954669205873	0.329954669210264
0.9	0.287250678558054	0.287250678558054	0.287250678555912
1	0.250073601112094	0.250073601112094	0.250073601113749

x	Analytic solution(y_{2a})	Neural solution (Trainlm) (y_{2t1})	Neural solution (Trainbfg) (y_{2t2})
0	0	1.530880578316805e-16	-3.059617923823967e-12
0.1	0.128510464762339	0.128419804302230	0.128510464758401
0.2	0.238619746993297	0.238604066374236	0.238627462671385
0.3	0.332733745257234	0.332733745257234	0.332737486328395
0.4	0.412946642430978	0.412945914088550	0.412946642432535
0.5	0.481081253748791	0.481081253748791	0.481081253754288
0.6	0.538724152786095	0.538724152786095	0.538724152783951
0.7	0.587256251241918	0.587256251241918	0.587254349184960
0.8	0.627879420907607	0.627880263180820	0.627876612156372
0.9	0.661639670057994	0.661639670057993	0.661639670057443
1	0.689447320205629	0.689447320205629	0.689447320203182

Table (3) : A comparison between analytic and neural solutions

$\lvert y_{2a} - y_{2t2} \rvert$	$\lvert y_{1a} - y_{1t2} \rvert$	$\lvert y_{2a} - y_{2t1} \rvert$	$\lvert y_{1a} - y_{1t1} \rvert$
3.059617923823967e-12	1.622489973456265e-04	1.530880578316805e-16	0
3.937739023740505e-12	7.728262474415715e-13	9.066046010927642e-05	6.899025035600470e-05
7.715678088132671e-06	8.324452238639424e-13	1.568061906176421e-05	0
3.741071161966758e-06	2.143048601244146e-06	5.551115123125783e-17	0
1.557254325490476e-12	5.532835937183123e-07	7.283424278692330e-07	7.768340090974490e-06
5.497102772977769e-12	2.620459405022757e-12	0	1.110223024625157e-16
2.144506794365952e-12	1.555866546709694e-12	1.110223024625157e-16	7.437473311000531e-06
1.902056958269505e-06	5.691048232669971e-08	2.220446049250313e-16	5.128752899485889e-06
2.808751235261475e-06	4.391098595846188e-12	8.422732126200927e-07	0
5.505595979116151e-13	2.141509192199465e-12	1.110223024625157e-16	0
2.447597680088620e-12	1.654731907052565e-12	1.110223024625157e-16	5.551115123125783e-17

Table (4): Weight and bias of the network for different training algorithm

Bias for trainlm of (y_1)	**Weight for trainlm of (y_1)**	**Weight for trainlm of (y_1)**
Net.B{1}	Net. LW{2,1}	Net.IW{1,1}
0.252859672695871	0.017611616453924	0.241569248284151
0.024385651630710	0.681708279346507	0.351195035228651
0.530723581584766	0.880833084532577	0.126738675949935
0.934854674058909	0.139821021807629	0.739408639260099
0.769328503579249	0.810853784598876	0.445679823858157

bias for trainbfg of (y_1)	**Weight for trainbfg of (y_1)**	**Weight for trainbfg of (y_1)**
Net.B{1}	Net. LW{2,1}	Net.IW{1,1}
0.413824842108441	0.670860771332138	0.715931252353248
0.180845024472233	0.982859184594668	0.248146352936429
0.995631187232624	0.936821328628590	0.531900325168972
0.520386194349656	0.576272354062037	0.382209682936579
0.885282695498373	0.080186187030087	0.801761242973036

Bias for trainbfg of (y_2)	**Weight for trainbfg of (y_2)**	**Weight for trainbfg of (y_2)**
Net.B{1}	Net. LW{2,1}	Net.IW{1,1}
0.431049305282942	0.688725578355128	0.869492229395656
0.190738233247638	0.596021891944303	0.947906559807061
0.327775459738820	0.214739790766330	0.254304842122024
0.937657078181142	0.535169282982425	0.048491632422091
0.847367010662909	0.055755557082901	0.831838695092381

Weight for trainlm of (y_2)	**Weight for trainlm of (y_2)**	**bias for trainlm of (y_2)**
Net.IW{1,1}	Net. LW{2,1}	Net.B{1}
0.101305687973029	0.618786526718974	0.241305720471971
0.468935089858187	0.699150113317535	0.594729968864710
0.710646362418379	0.314828670017582	0.868232914442518
0.601346407503426	0.804788645898101	0.818668373595483
0.204916976156494	0.178022430412338	0.590084132290161

Table(5) : The performance of the train with epoch and time

Train code	Performance for MSE	Time	Epoch	Performance
Trainlm of y_1	4.010417891598336e-10	0:00:04	295	2.20e-33
Trainlm of y_2	6.927087884925925e-10	0:00:06	468	3.96e-33
Trainbfg of y_1	2.154243203877806e-09	0:00:28	1268	5.07e-28
Trainbfg of y_2	6.957344125042582e-12	0:00:12	543	2.26e-25

Table (6) : The analytic & neural result of the problem (9)

x	Analytic solution (y_{a1})	Neural (Trainlm) (y_{1t1}) solution	Neural (Trainbfg)(y_{1t2}) solution
0	0	0	0.008076966501195
0.1	1.120552473617658	1.119074768489855	1.121576309950494
0.2	2.096078776007225	2.096078776007224	2.096078776010794
0.3	2.945348746900685	2.945519665978265	2.945329004027621
0.4	3.684702961845622	3.684702961845623	3.684702961851884
0.5	4.328367136833180	4.328367136833180	4.328367136844937
0.6	4.888725841480949	4.888755428434839	4.888725841491276
0.7	5.376560787212696	5.376560787212696	5.376560787208372
0.8	5.801258275273828	5.801258275273830	5.801258275281827
0.9	6.170989796034165	6.171052575520006	6.170989796037555
1	6.492869254440744	6.492869254440744	6.492853256454569

x	Analytic solution(y_{a2})	Neural (Trainlm) (y_{2t1}) solution	Neural (Trainbfg) (y_{2t2}) solution
0	0	0	6.970449786967482e-08
0.1	0.079077570606103	0.083579667139451	0.079077614840971
0.2	0.301069405674204	0.301069405674204	0.296617057945247
0.3	0.645370996519114	0.645370996519114	0.643383165800754
0.4	1.094073563826115	1.094073563826114	1.094073602030272
0.5	1.631614766704876	1.629619175798611	1.631757503744028
0.6	2.244474623805575	2.242721178439358	2.244474633309778
0.7	2.920910794904590	2.920910794904590	2.920910793699243
0.8	3.650728128703034	3.650322736313918	3.650728106463634
0.9	4.425078041900662	4.425078041900662	4.425158477200711
1	5.236283868595361	5.236283868595361	5.236283818105677

Table (7) : A comparison between analytic and neural solutions

$\lvert y_{2a} - y_{2t2} \rvert$	$\lvert y_{1a} - y_{1t2} \rvert$	$\lvert y_{2a} - y_{2t1} \rvert$	$\lvert y_{1a} - y_{1t1} \rvert$
6.970449786967482e-08	**0.008076966501195**	**0**	**0**
4.423486724136616e-08	**0.001023836332836**	**0.004502096533347**	**0.001477705127803**
0.004452347728957	**3.569144979564953e-12**	**6.106226635438361e-16**	**4.440892098500626e-16**
0.001987830718360	**1.974287306349964e-05**	**0**	**1.709190775800451e-04**
3.820415694910651e-08	**6.261213769676033e-12**	**6.661338147750939e-16**	**4.440892098500626e-16**
1.427370391517968e-04	**1.175681774157056e-11**	**0.001995590906265**	**0**
9.504203646315545e-09	**1.032685048585336e-11**	**0.001753445366216**	**2.958695388954880e-05**
1.205346933375040e-09	**4.324540725519910e-12**	**0**	**8.881784197001252e-16**
2.223940009926650e-08	**7.998046669399628e-12**	**4.053923891160771e-04**	**1.776356839400251e-15**
8.043530004897548e-05	**3.389288849575678e-12**	**0**	**6.277948584099136e-05**
5.048968443333024e-08	**1.599798617490933e-05**	**0**	**0**

Table(8): Weight and bias of the network for different training algorithm

Bias for trainlm of (y_1)	Weight for trainlm of (y_1)	Weight for trainlm of (y_1)
Net.B{1}	Net. LW{2,1}	Net.IW{1,1}
0.134776995255468	0.792456860497627	0.505692448683919
0.569914349014358	0.837900449095441	0.004625760582352
0.574691506274377	0.687358399403697	0.188468748383506
0.016102750791896	0.081434100056796	0.779639443833603
0.934828315776462	0.806140569013718	0.610825949687488

Weight for trainlm of (y_2)	Weight for trainlm of (y_2)	bias for trainlm of (y_2)
Net.IW{1,1}	Net. LW{2,1}	Net.B{1{
0.478376283649921	0.922429224841606	0.912782147048597
0.124994240255344	0.934308461414860	0.534170292387813
0.064155985584096	0.824811938202454	0.451461598784967
0.386664880532908	0.183582125058882	0.285596533472468
0.558245691279696	0.393108093448545	0.622774097441083

Bias for trainbfg of (y_2)	Weight for trainbfg of (y_2)	Weight for trainbfg of (y_2)
Net.B{1}	Net. LW{2,1}	Net.IW{1,1}
0.612551084945355	0.002291815245818	0.018010467945131
0.133072433875949	0.772712775811915	0.503631782070622
0.283353859003173	0.661798871106240	0.126991991291519
0.837400333016416	0.397966331038198	0.305174566418702
0.865931228918039	0.501051178487892	0.256058502513567

Bias for trainbfg of (y_1)	Weight for trainbfg of (y_1)	Weight for trainbfg of (y_1)
Net.B{1}	Net. LW{2,1}	Net.IW{1,1}
0.828924078623420	0.100480242633400	0.735194614493405
0.724963349991644	0.522668514129408	0.879697065220770
0.337293776914633	0.058688252904172	0.851423388162803
0.211463983876867	0.813646068464573	0.672029350928386
0.339840122932540	0.345685331495202	0.798496214678467

Table (9) : The performance of the train with epoch and time

Performance for MSE	Time	Epoch	Performance	Train code
1.814434713408252e-07	0:00:05	302	2.82e-32	Trainlm of y1
2.249195719271235e-06	0:00:02	102	4.83e-32	Trainlm of y2
5.423422452091749e-06	0:00:35	1461	2.49e-26	Trainbfg of y1
1.947413043462885e-06	0:00:05	243	2.20e-23	Trainbfg of y2

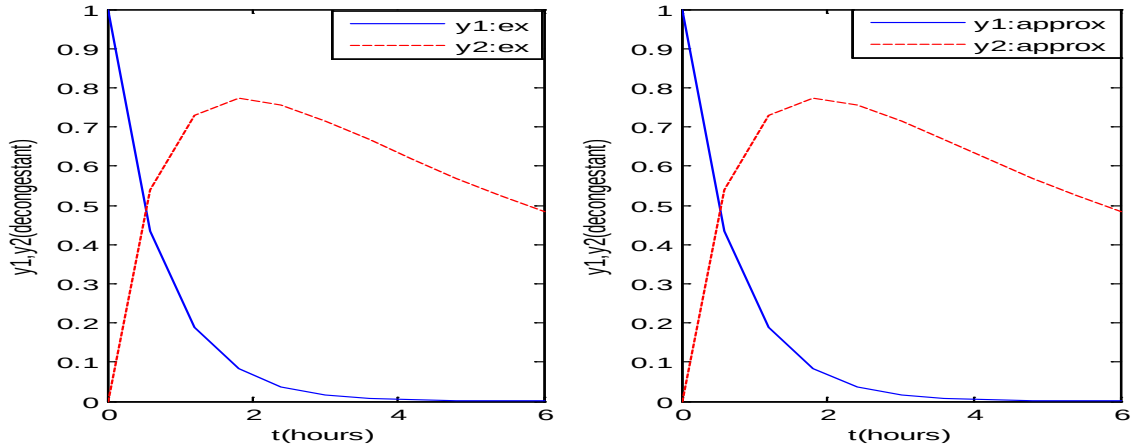


Fig. (1): decongestant in GI tract(solid)and bloodstream(dashed):single unit dose (k_1 = 1.386, k_2 = 0.1386)

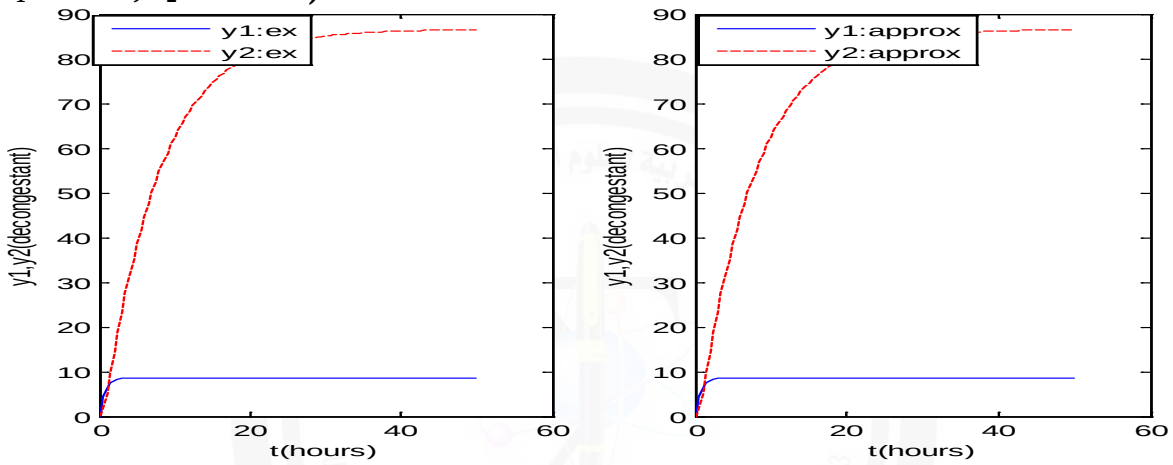
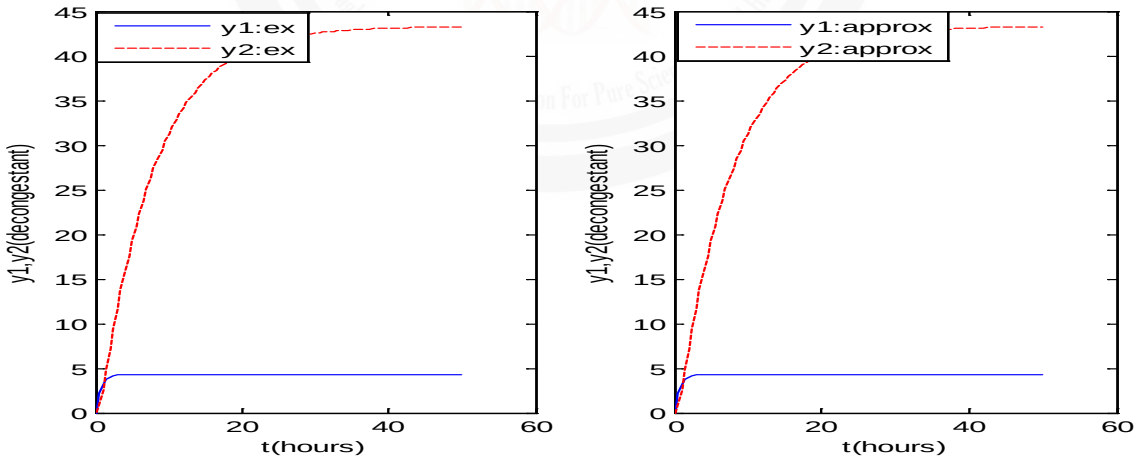


Fig. (2): Decongestant in GI tract (solid) and bloodstream (dashed) : continuous acting capsules (k_1 = 1.386, k_2 = 0.1386) , I = 12 .



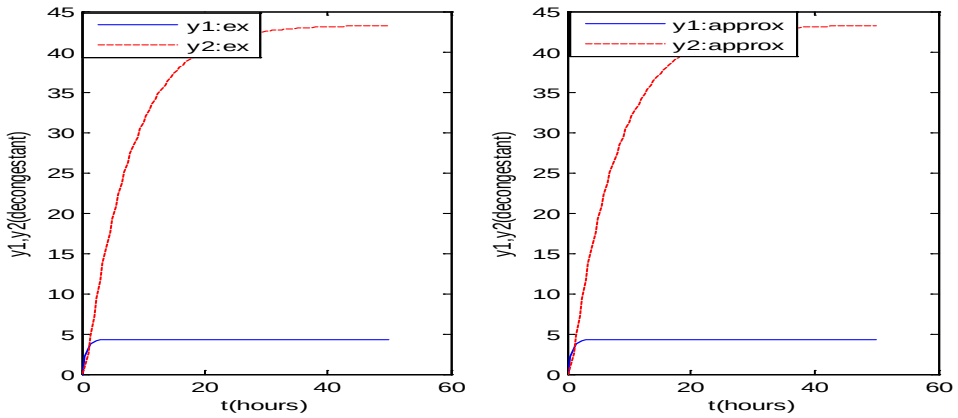


Fig. (3): Decongestant in GI tract (solid) and bloodstream (dashed): continuous acting capsules ($k_1 = 1.386$, $k_2 = 0.1386$) , I = 6

Ibn Al-Haitham Journal for Pure and Applied Science
مجلة إبن الهيثم للعلوم الصرفة و التطبيقية
No. 3 Vol. 25 Year 2012
العدد 3 المجلد 25 السنة 2012

تصميم شبكة عصبية صناعية ذو تغذية تقدمية لتحديد جرعات دواء الديكونجيستد

لمى ناجي محمد توفيق و يسر رافد بحر المعيوف
قسم الرياضيات، كلية التربية- ابن الهيثم، جامعة بغداد
استلم البحث في:30 ايار 2012 قبل البحث في : 24 ايلول 2012

الخلاصة

الهدف من هذا البحث هو تصميم شبكة عصبية صناعية ذو تغذية تقدمية لتحديد تأثير حبوب البرد و تعاقبها من خلال استخدام النمذجة و تحويل المشكلة إلى منظومة معادلات مسائل القيم الابتدائية من الرتبة الأولى .هذه المسألة مثال لعدد من النماذج التي تمثل مرور العلاج خلال الجاسم. الانموذج المصمم هو جزء مهم في العلاج والذي يحدد مستوى الجرعات و معدلها حيث ان التحليل الحرج هو الحاجة لبقاء مستوى العلاج عال و فعال لكن ليس العلو الذي يجعله خطر
الكلمات المفتاحية:. الشبكة العصبية، الشبكة العصبية التقدمية،استخدام النمذجة .