REPRESENTATION OF FIRST ORDER DYNAMICAL SYSTEMS USING NEURAL NETWORKS

J.-A. LUZARDO *
Universidad Simón Bolívar. Dpto. de PyS.
Caracas 1081A. Venezuela
jluzardo@usb.ve
A. CHASSIAKOS
California State University,Long Beach. ET Dept.
Long Beach. CA 90840
achassk@engr.csulb.edu
A. RUMBOS
Pomona College. Math Dept.
Claremont. CA 91711
arumbos@pomona.edu

ABSTRACT

The approximation of Dynamical Systems (DSs) using Neural Networks (NNs) is considered in this paper in a broader sense than the mere trajectory approximation for finite time. The object of this study is to try to determine the capabilities of NNs to reproduce structural properties of DSs in order to achieve approximation for all trajectories that remain in a closed region of the state space as t tends to infinity. This is a new approach to approximating DSs using NNs, which we will call the representation of DSs rather than an approximation of trajectories. The problem so stated is under current research, and the preliminary results concerning first order dynamical systems are presented here.

1. Introduction

A natural extension of the well known capabilities of NNs (Neural Networks) to approximate static maps (Cybenko [2], Funahashi [3], Hornik et al [7]) is the approximation of dynamical systems (DSs). In this sense, we can refer to Li [9], and Funahashi and Nakamura [4], who proved that trajectories of a dynamical system in an interval $[t_0, T]$ can be approximated by a continuous time recurrent NN. In both works, the n-dimensional dynamical system is embeded into a higher dimensional one corresponding to a recurrent NN. Chen and Chen [1] give a different approach by approximating the output of a dynamical system at any fixed time since such a output can be expressed as a continuous functional defined on $\boldsymbol{C}[a, b]$. Olurotimi [11] proposes a method in which the parameters of a recurrent neural network are determined by training the parameters of an associated feedforward NN. The goal of the training is to approximate the state variables of a given dynamical system during a finite time.

The results mentioned above are restricted to the approximation of trajectories (also called orbits) for finite time intervals or fixed time. These results show limited capabilities of NNs to approximate dynamical sytems and do not tell us if NNs can actually reproduce the broad behavioral complexity that most DSs present and which is characterized by their structural properties: equilibrium points, limit cycles, closed orbits, attractors, strange attractors, etc. We introduce the concept of *representation* of DSs in order to point out that a complete dynamical system approximation must involve two aspects: the trajectory approximation itself and the duplication of the orbit structure (structural properties). To be more specific, we describe our general problem as follows: given a dynamical system S_1, the representation problem consists in finding another dynamical system S_2 which is *topologically equivalent* (Hale and Koçak [6], Wiggins [12]) to S_1, and whose orbits approximate the orbits of S_1 with any desired degree of accuracy in a compact subset of $\boldsymbol{R}^n$. Note that, for the representation problem, time does not need to be finite since the approximation remains valid as long as the orbits do not leave the compact subset of $\boldsymbol{R}^n$.

The problem so stated is highly complex. In this paper, we begin by considering the simple case

* This work is part of the author's current PhD dissertation in the Engineering Mathematics program at The Claremont Graduate School, CA 91711, USA.

0-7803-2768-3/95/$4.00

of scalar dynamical systems. Special emphasis is placed on the representation of *structurally unstable* systems ([6], Chap 2). For this kind of system, very small perturbations in the vector field can produce a completely different orbit structure, and because of this, the approximation will eventually fail. This problem can be solved by using NN capabilities to approximate static functions as shown in this paper.

2. Flow Equivalence of Scalar Differential Equations

Consider two scalar differental equations (first order systems):

$$\dot{x} = f(x) \qquad (1)$$

and

$$\dot{x} = g(x), \qquad (2)$$

where $x \in \boldsymbol{R}$ and $f(x)$, $g(x)$ are $\boldsymbol{C}^r$ $(r \geq 1)$.

Intuitevely, the two flows generated by f(x) and g(x) are considered to be qualitatively equivalent if they have the same orbit structure, i.e., if they have equal number of orbits and the directions of the flows on the corresponding orbits are the same (Hale and Koçak [6], Chap. 2). In the case where $g(x) = f(x) + \epsilon(x)$ (the system described by (2) is a perturbed system with respect to the original one (1)), the flow equivalence is guaranteed if the flow generated by f(x) is structurally stable. Thus, the feasible approximations g(x) we are dealing with, are those which satisfy the ε-neighborhood of f(x) in the C^1 topology. When this is the case, we shall say that g belongs to the ε-$\boldsymbol{C}^1$ neighborhood of f, i.e.,

$$\begin{aligned} \|g(x) - f(x)\|_1 &= \max_{x \in I}\{\ |g(x) - f(x)|, \\ &\qquad |g'(x) - f'(x)|\ \} < \varepsilon. \end{aligned}$$

By requiring the functions as well as their derivatives to be close enough at all points in the closed interval I, the perturbed system g will keep the hyperbolic equilibria of f in number and nature.

Theorem 2.1. A scalar differential equation $\dot{x} = f(x)$ with a finite number of equilibrium points is structurally stable if and only if all its equilbrium points are hyperbolic.

Proof. This theorem is a particular case of the Peixoto's Theorem about structural stability for vector fields on two-dimensional manifolds (see Guckenheimer and Holmes [5] and Wiggins [12] for more details).□

Structurally stable systems have robust properties in the sense that, if a system is structurally stable, then any sufficiently close system has the same qualitative behavior (orbit structure). The situation is different for a structurally unstable system; in this case, there are certain sort of perturbations that do produce structural changes in the vector field no matter how small they are. Bifurcation Theory deals with the effects caused by such perturbations, whereas this study focuses on the reliable representation of scalar dynamical systems using neural networks. Therefore, we are interested, as a first step, in determining those perturbations which preserve the orbit structure of structurally unstable systems.

Proposition 2.1. Suppose that f(x) in (1) is a $\boldsymbol{C}^{r+1}$ $(r \geq 1)$ function and has a finite number of equilibrium points. Assume, moreover, that f(x) has only one non-hyperbolic equilibrium point located (without loss of generality) at $\bar{x} = 0$, and that the values of the first r derivatives of f(x) at $\bar{x} = 0$ are identically zero. that is,

$$f'(0) = f^{(2)}(0) = \cdots = f^{(r)}(0) = 0.$$

Let g(x) be a $\boldsymbol{C}^{r+1}$ function which belongs to the ε-$\boldsymbol{C}^1$ neighborhood of f. Then, a sufficient condition for the perturbed system $\dot{x} = g(x)$ to be topologically equivalent to $\dot{x} = f(x)$ is

$$g(0) = g'(0) = g^{(2)}(0) = \cdots = g^{(r)}(0) = 0. \qquad (3)$$

Proof. Since g(x) is in the ε-$\boldsymbol{C}^1$ neighborhood of f, we can write:

$$g(x) = f(x) + \varepsilon\, h(x),$$

where $|h(x)| < 1$ in the open interval $I \in \boldsymbol{R}$. By the smoothness of f(x) and g(x), we can Taylor expand both functions and conclude that the power series of h(x) is similar to that one of g(x) or f(x). Rewriting g(x) as a power expansion, we obtain:

$$\begin{aligned} g(x) &= (a_{r+1}\, x^{r+1} + O_f(r+1)) \\ &\quad + \varepsilon\, (b_{r+1}\, x^{r+1} + O_h(r+1)) \\ &= (a_{r+1} + \varepsilon\, b_{r+1})\, x^{r+1} \\ &\quad + (O_f(r+1) + \varepsilon\, O_h(r+1)). \end{aligned}$$

The expressions $O_f(r+1)$ and $O_h(r+1)$ represent the higher order terms, and can be made as small as necessary by taking an appropiate η such that $|x| < \eta$. Therefore, the local behavior of g(x) around $\bar{x} = 0$ is determined by the term $(a_{r+1} + \varepsilon\, b_{r+1})\, x^{r+1}$. If ε remains sufficiently small for $|x| < \eta$, then the sign of g(x) is equal to the sign of f(x) in a neigborhood about $\bar{x} = 0$. This results in the local equivalence of both vector fields. The extension of this result to the entire interval I follows from the fact that g(x) is in the ε-$\boldsymbol{C}^1$ neighborhood of f(x) and the remaining equilibrium points are hyperbolic.□

Condition (3) is an attempt to reproduce the local behavior around the non-hyperbolic fixed points. Notice that the condition (3) should be satisfied at every non-hyperbolic fixed point of f(x) in I as a way to obtain topological equivalence.

Although condition (3) of Proposition 2.1 is strong, we will show in Section 3 that neural networks can actually satisfy it. A neural network can be considered as a perturbed system described by (2). By adjusting the parameters of the net, it is possible to obtain NNs which are equivalent to scalar structurally unstable systems with finite number of fixed points.

3. Dynamical neural networks and its capabilities to represent dynamical systems

The sort of dynamical neural network (DNN) considered in this study is described by a three layer NN whose outputs are integrated in time and fed back. The equation corresponding to the k-th state variable of a DNN with n state variables and m hidden units is:

$$\dot{x}_k = \sum_{j=1}^{n} c_{kj}\, \sigma(\sum_{i=1}^{m} w_{ji} x_i + \theta_j)\,.$$

In the particular case of a first-order DNN, the previous equation is reduced to:

$$\dot{x} = N(x) = \sum_{i=1}^{k} c_i\, \sigma(w_i x + \theta_i)\,, \qquad (4)$$

where $\sigma(\cdot)$ is a sigmoid function given by:

$$\sigma(z) = \frac{1}{1+e^{-z}}\,. \qquad (5)$$

We remark that the sort of DNN considered in this paper, equation (4), are conceptually different from the continuous time recurrent neural networks (RNNs) widely treated in the literature. In our case, neurons of DNNs in the hidden layer do not have dynmics, whereas, in the case of RNNs, each neuron of the hidden layer has dynamics. Therefore, the order of a DNN remains constant no matter the number of neurons in the hidden layer.

In order to understand the dynamical capabilities of the system (4) to represent first order dynamical systems described by (1), we first approach the static approximation of f(x) through the three-layer feedforward NN N(x). It is clear that a static approximation in the sense of the C^0 topology (Cybenko [2], Funahashi [3], Hornik et al [7]) is not enough to guarantee the dynamical approximation. The following theorem shows that a C^m topology can be obtained:

Theorem 3.1. Let $\zeta^{(m)}(x)$ be the m-th derivative of the sigmoid function $\zeta(x) = 1/(1+\exp(-x))$, and $m \geq 1$. Let K be a compact subset of $\boldsymbol{R}^n$, and $f(\boldsymbol{x})$ be a real valued continuous function on K. Then for any $\epsilon > 0$, there exists an integer k and real constants c_i, θ_i $(i = 1, \ldots, k), w_{ij}$ $(i = 1, \ldots, k, j = 1, \ldots, n)$ such that

$$N_{\zeta^{(m)}}(x_1, \ldots, x_n) = \sum_{i=1}^{k} c_i \zeta^{(m)} \left(\sum_{j=1}^{n} w_{ij} x_j + \theta_i \right)$$

satisfies

$$\max_{x \in K} |f(x_1, \ldots, x_n) - N_{\zeta^{(m)}}(x_1, \ldots, x_n)| < \epsilon\,,$$

for any $m \geq 1$.

Proof. The proof is similar to that given by Funahashi in [3] since $\zeta^{(m)}(x) \in L^1(\boldsymbol{R})$ $(m \geq 1)$ and its Fourier transform $(\Sigma^{(m)}(\xi))$ is non-zero at $\xi = 1$.□

Remark 1. Theorem 3.1 tells us that the derivatives of the sigmoid function can be used as activation functions in a NN. In the scalar case,the approximation in the C^m topology can be obtained using the fact that $|f^{(m)}(x) - N^{(m)}(x)| < \epsilon$. Extra conditions are needed to reduce the effects of the integration constants. These conditions have to do with the *closeness* between f(x) and N(x) at isolated points of a closed interval I; for instance, the conditions $N(a_0) = f(a_0), N'(a_1) = f'(a_1), \ldots, N^{(m-1)}(a_{m-1}) = f^{(m-1)}(a_{m-1})$ $(a_i \in I,\ i = 0, \ldots, m-1)$ eliminate the effects of the integration constants.

Hornik et al [8] proved for the general case that NNs can approximate any C^m non-linear map in the C^m topology. In conclusion, these results about static approximation directly are applied to obtain dynamical approximation as long as equilibria are not included in the interval I where the dynamical approximation is carried out. In such a case, by increasing the number of neurons and/or improving the parameter training, a specific approximation goal can be achieved as the trajectory remains in the interval. On the contrary, when equilibrium points are present in the interval I, we need the NN to match the function f(x) and its derivatives at those equilibrium points as stated in Proposition 2.1. In this way, the DNN (4) represents the dynamical system (1) as long as the trajectories do not leave the interval I, i.e., the trajectories generated by any initial condition $x(0) \in I$ in both systems have the same qualitative behavior and are approximated to each other with any desired degree of accuracy as $x(t) \in I$. The following theorem illustrates the capabilities of NNs to match functions at isolated points:

Theorem 3.2. Assume that $N(x)$ is a NN given by

$$N(x) = \sum_{i=0}^{n} c_i\,\sigma(w\,x + \theta_i)\,,$$

where $x \in \boldsymbol{R}$, $w \neq 0$, and $\sigma(z) = 1/(1+\exp(-z))$. Let $f(x)$ be a $\boldsymbol{C}^n[a,b]$ function and F a $n \times 1$ constant vector given by

$$F = (f(\bar{x})\ f'(\bar{x}) \cdots f^{(n-1)}(\bar{x}))^T\,,$$

where $\bar{x} \in [a,b]$ and T denotes transpose. Then the linear equation system on c_i

$$\Gamma\,C = F\,, \tag{6}$$

where:

$$\Gamma = \begin{bmatrix} y_1(\bar{x}) & y_2(\bar{x}) & \cdots & y_n(\bar{x}) \\ y_1'(\bar{x}) & y_2'(\bar{x}) & \cdots & y_n'(\bar{x}) \\ \vdots & \vdots & \cdots & \vdots \\ y_1^{(n-1)}(\bar{x}) & y_2^{(n-1)}(\bar{x}) & \cdots & y_n^{(n-1)}(\bar{x}) \end{bmatrix},$$

$$\begin{aligned} y_i^{(j)}(\bar{x}) &= \sigma^{(j)}(w\,\bar{x} + \theta_i)\,, \\ \forall i &= 1, \ldots, n\,, \\ \forall j &= 0, \ldots, n-1\,, \\ C &= (c_1\,c_2 \ldots c_n)^T\,, \end{aligned}$$

has a unique solution $C = \Gamma^{-1} F$ if and only if the sigmoidal functions of the NN are all distint, that is, $\theta_i \neq \theta_j$, $\forall\, i \neq j$.

Proof. The proof of the theorem consists in proving that $\det(\Gamma) \neq 0$ if and only if $y_i(\bar{x}) \neq y_j(\bar{x})$ for $i \neq j$. In [10] a generic expression for the derivatives of the sigmoide was found:

$$y^{(n)}(x) = \sigma^{(n)}(w\,x + \theta) = w^n \sum_{k=1}^{n+1} (-1)^{k-1} K_k^{(n)} y^k\,, \tag{7}$$

where $K_k^{(n)} = (k-1)!\,S_{n+1,k}$, $S_{n,k}$ are the Stirling numbers of the second kind (see [10] and references therein), and $K_k^{(n)} = 0\ \forall n < 0, k < 1, k > n+1$. Substituting (7) in Γ yields:

$$\Gamma(\bar{x}) = K\,V_y(\bar{x})\,D_y(\bar{x})\,,$$

where $K = [k_{ij}]$ $(i = 1, \ldots, n;\ j = 1, \ldots, n)$,

$$k_{ij} = \begin{cases} 0 & \text{for } j > i, \\ (-1)^{j-1} w^{i-1} K_j^{i-1} & \text{for } j \leq i, \end{cases}$$

$$V_y(\bar{x}) = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ y_1(\bar{x}) & y_2(\bar{x}) & \cdots & y_n(\bar{x}) \\ y_1^2(\bar{x}) & y_2^2(\bar{x}) & \cdots & y_n^2(\bar{x}) \\ \vdots & \vdots & \ddots & \vdots \\ y_1^{n-1}(\bar{x}) & y_2^{n-1}(\bar{x}) & \cdots & y_n^{n-1}(\bar{x}) \end{bmatrix},$$

$$D_y(\bar{x}) = \begin{bmatrix} y_1(\bar{x}) & 0 & 0 & \cdots & 0 \\ 0 & y_2(\bar{x}) & 0 & \cdots & \vdots \\ 0 & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & y_n(\bar{x}) \end{bmatrix}.$$

Since no coefficient in the diagonals of K and D_y is zero, the only possibilty for Γ to be singular is that the Vandermonde matrix V_y is singular, and this occurs if and only if $y_i(\bar{x}) = y_j(\bar{x})$, for any $i \neq j$.□

Remark 1. The theorem establishes that NNs can locally approximate a function up to the desired derivative order by having the appropriate number of neurons (one more than the derivative order) and matching the function and its derivatives at a specific point. The approximation is valid in a certain neighborhood of that point.

Remark 2. Theorem 3.2 points out that a NN can satisfy the condition required in Proposition 2.1. Intuitively, assigning different values of w to each sigmoid would ease the approximation process, however Theorem 3.2 does not determine the conditions under which Γ would be singular for such a case. Ideally, those conditions must define a set $\aleph \in \boldsymbol{R}$ with measure zero in order to assure that only isolated points lead to the singularities. For instance, consider the case of two different sigmoids ($w_1 \neq w_2$, $\theta_1 \neq \theta_2$) where Γ is described by:

$$\Gamma = \begin{bmatrix} y_1(x) & y_2(x) \\ w_1\,y_1(x)(1-y_1(x)) & w_2\,y_2(x)(1-y_2(x)) \end{bmatrix}.$$

Γ is singular if $w_1(1-y_1(x)) = w_2(1-y_2(x))$. By differentiating both sides of this last equation, we obtain that Γ is singular only at isolated points.

Remark 3. By having more sigmoids than necessary we increase the degrees of freedom. We can use these extra degrees of freedom to make the necessary adjustments at the critical points as required by Proposition 2.1 without deteriorating too much the approximation in the rest of the interval. Notice that only acting upon the parameters c_i, corresponding to the linear combination of the sigmoids, we can obtain this.

4. A simulation example

Let us consider the scalar system:

$$\dot{x} = f(x) = \frac{1 + 3x - 4x^3}{10},$$

which has two equilibrium points: $\bar{x}_1 = -0.5$ and $\bar{x}_2 = 1$. The first one is non-hyperbolic with quadratic degeneracy since $f(\bar{x}_1) = 0$, $f'(\bar{x}_1) = 0$ and $f''(\bar{x}_1) \neq 0$; therefore, the system is structurally unstable. A neural network with twelve neurons and parameter vectors W, θ and C (see table 1) was trained to approximate the function f(x), and the error $\max_{x \in [-1.5,1.5]} |f(x) - N(x)| < 3.5 \times 10^{-4}$ was obtained. Fig. 1 depicts both functions f(x) and N(x) around the non-hyperbolic equilibrium point $\bar{x}_1 = -0.5$.

Neural Network Parameters						
i	1	2	3	4	5	6
w_i	3.00	2.40	1.80	1.20	0.60	-0.60
θ_i	3.75	2.40	1.35	0.60	0.15	0.15
c_i	1.0839	-1.1017	5.7833	14.5648	-36.3556	36.3556
$c_{new(i)}$	4.0765	-9.2915	28.2436	-14.1857	-36.3556	34.8971
i	7	8	9	10	11	12
w_i	-1.20	-1.80	-2.40	-3.00	1.00	-1.00
θ_i	0.60	1.35	2.40	3.75	3.00	-3.00
c_i	-14.5648	-5.7833	1.1017	-1.0839	0.1000	0.1000
$c_{new(i)}$	-14.5648	-14.7519	3.8677	-1.5299	11.4252	7.1691

Table 1: Parameters of N(x) and $N_{new}(x)$

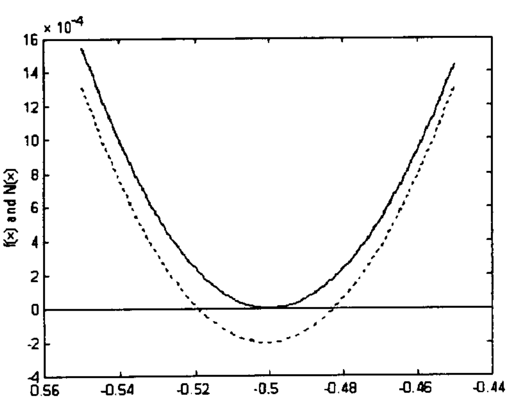

Fig. 1: Neural network N(x) (dotted) and $f(x) = (1 + 3x - 4x^3)/10$ (solid) around $\bar{x}_1 = -0.5$.

Note that the flows of the scalar systems $\dot{x} = f(x)$ and $\dot{x} = N(x)$ are not topologically equivalent though the static approximation is considered to be good. There are trajectories in the interval $I_o = [-1.5, 1.5]$ that can not be approximated at all. For instance, consider the initial condition $x(0) = -0.49$ that originates the time responses illustrated in fig. 2.

In order to achieve the topological equivalence between both flows, the conditions of Proposition

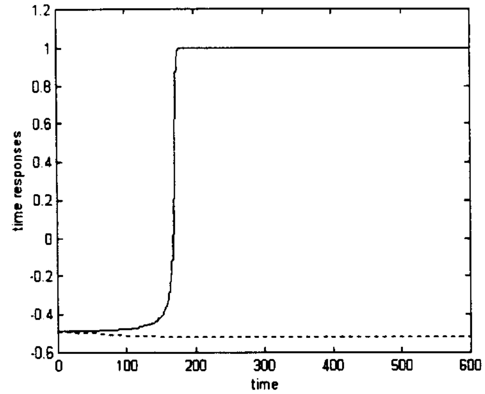

Fig. 2: Time responses for the initial condition $x(0) = -0.49$ of $\dot{x} = N(x)$ (dotted) and $\dot{x} = f(x)$ (solid).

2.1 must be met. In this particular example, we need only to modify the parameters c_i of the previous NN (see Theorem 3.2) to match f(x) and its derivative at the critical point; however, the topological equivalence does not guarantee per se a good approximation of all the trajectories in the interval I_o but their qualitative behavior. Therefore, in order not to lose the static approximation which will improve the dynamical one, we chose five different points (including $\bar{x}_1$) to match both functions. In other words, the new NN will satisfy: $N_{new}(x_i) = f(x_i)$ and $N'_{new}(x_i) = f'(x_i)$, where $x_1 = -1.5$; $x_2 = -0.5$; $x_3 = 0.5$; $x_4 = 1$, and $x_5 = 1.5$. The new coefficients are defined by: $C_{new} = C + \Delta C$, where ΔC is found by solving the linear equation system:

$$\begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_5 \end{bmatrix} \Delta C = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_5 \end{bmatrix} - \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_5 \end{bmatrix} C,$$

where

$$\Gamma_k = \begin{bmatrix} y_1(x_k) & y_2(x_k) & \cdots & y_{12}(x_k) \\ y'_1(x_k) & y'_2(x_k) & \cdots & y'_{12}(x_k) \end{bmatrix},$$

$$y_i^{(j)}(x_k) = \sigma^{(j)}(w_i x_k + \theta_i),$$

$$F_k = (f(x_k)\, f'(x_k))^T,$$

and $k = 1, \ldots, 5$; $i = 1, \ldots, 12$; $j = 0, 1$. Notice that the matrix $\Gamma = [\Gamma_1^T \ldots \Gamma_5^T]^T$ has rank 10 (two neurons are needed for each point as the NN matches the values of $f(\cdot)$ and $f'(\cdot)$, see Theorem 3.2 and its corresponding remarks), therefore the solution exists and there are two extra degrees of freedom. These extra degrees of freedom can be used in the process of achieving the minumum $\|\Delta C\|$

and the subsequent static approximation along I_o. Fig. 3 shows the approximation static error between $N_{new}(x)$ and f(x), and Fig. 4 shows the time responses for the initial condition $x(0) = -4.9$ for both dynamical systems $\dot{x} = f(x)$ and $\dot{x} = N_{new}(x)$. These systems are totally equivalent and approximate each other for any initial condition in the interval $I_o = [-1.5, 1.5]$.

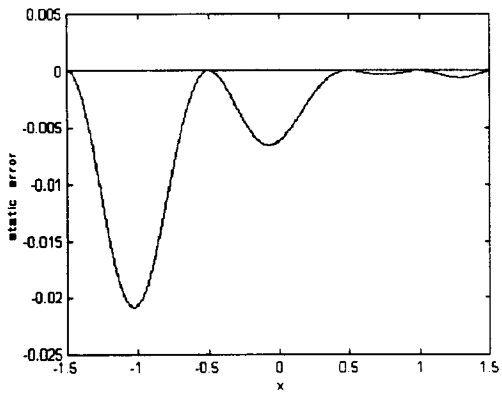

Fig. 3: Static approximation error: $f(x) - N_{new}(x)$

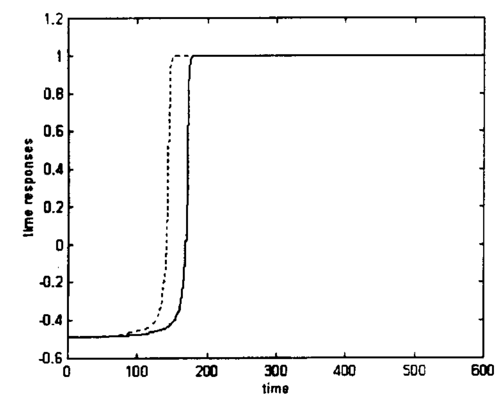

Fig. 4: Time responses for the initial condition $x(0) = -0.49$ of $\dot{x} = N_{new}(x)$ (dotted) and $\dot{x} = f(x)$ (solid).

5. Conclusions

We have shown the feasibility for DNNs to represent first order dynamical systems with finite number of eqilibria in a closed interval $I \in \boldsymbol{R}$. The representation implies that, in an interval $I \in \boldsymbol{R}$, the trajectories of the DNN duplicate the qualitative structure of the original system's flow and approximate the corresponding trajectories. This has been done by reproducing the structural properties of the original system, that is, its equilibrium points and their respective characteristics. The excellent properties of NNs to approximate functions allow us to achieve this particular requirement for scalar systems. Unfortunately, the reproduction of equilibrium points and the local characteristics around them is not a sufficient condition to achieve the representation of higher order DSs. New constraints are necessary to obtain that goal, and the determination of such constraints is the subject of current investigation, as well as the capabilities of DNNs to satisfy them.

References

[1] Chen,T., & Chen, H. "Approximations of continuous functionals by neural networks with applications to dynamic systems". *IEEE Trans. on Neural Networks*, Vol. 4, No 6, pp. 910-918, 1993.

[2] Cybenko, G. "Aproximation by superpositions of a sigmoidal function". *Mathematics of Control, Signals and Systems*, Vol. 2, pp. 303-314, 1989.

[3] Funahashi, K. "On the approximate realization of continuous mappings by neural networks". *Neural Networks*, Vol. 2, pp. 183-192, 1989.

[4] Funahashi, K.I., & Nakamura, Y. "Approximation of dynamical systems by continuous time recurrent neural networks". *Neural Networks*, Vol. 6, pp. 801-806, 1993.

[5] Guckenheimer, J., & Holmes, P. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields.* Springer-Verlag, New York, 1983.

[6] Hale, J. K., & Koçak, H. *Dynamics and bifurcations.* Springer-Verlag, New York, 1991.

[7] Hornik, K., Stinchcombe, M., & White, H. "Multilayer feedforward networks are universal approximators". *Neural Networks*, Vol. 2, pp. 359-366, 1989.

[8] Hornik, K., Stinchcombe, M., & White, H. "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks". *Neural Networks*, Vol. 3, pp. 551-560, 1990.

[9] Li, L. K. "Aproximation of dynamical systems by continuous time recurrent neural networks" *IEEE International Conference on Neural Networks ICNN'92*, Vol.2, pp.266-271, 1992.

[10] Minai, A. A., & Williams, R. D. "On the derivatives of the sigmoid". *Neural Networks*, Vol. 6, pp. 845-853, 1993.

[11] Olurotimi, O. "Recurrent neural network training with feedforward complexity". *IEEE Trans. on Neural Networks*, Vol. 5, No 2, pp. 910-918, 1994.

[12] Wiggins, S. *Introduction to applied nonlinear dynamical systems and chaos.* Springer-Verlag, New York, 1990.