Comment on "Recurrent Neural Networks: a Constructive Algorithm, and its Properties"

Léon PERSONNAZ, Gérard DREYFUS
ESPCI, Laboratoire d'Électronique
10, rue Vauquelin, 75005 Paris, France
Gerard.Dreyfus@espci.fr

Abstract

In their paper [1], Tsoi and Tan present what they call a "canonical form", which they claim to be identical to that proposed in Nerrand et al [2]. They also claim that the algorithm which they present can be applied to any recurrent neural network. In the present comment, we disprove both claims.

Back in 1993, Nerrand et al. [2] proposed a general approach to the training of recurrent networks, either adaptively (on-line) or non-adaptively (off-line). One of the main points of that paper was the introduction of the minimal state-space form, or canonical form, defined in relations (4) and (4a) of their paper as:
$\underline{z}(n+1) = \varphi\,[\underline{z}(n), \underline{u}(n)]$ (state equation)
$\underline{y}(n+1) = \psi[\underline{z}(n+1), \underline{u}(n+1)]$ (output equation)
where $\underline{z}(n)$ is a state vector, i.e. a *minimal* set of variables necessary at time n for computing the future output vector $\underline{y}(n+1)$, the external input vector $\underline{u}(n+1)$ (control inputs, measured disturbances, ...) being known. A graphic representation of the canonical form is shown in Figure 1, assuming that functions φ and ψ are computed by a single neural network.

In appendix (1) of their paper, Nerrand et al. showed how to compute the order of any recurrent network, and, in appendix (2) they derived the canonical form of various recurrent network architectures, which had been proposed by other authors.
Since then, researchers of the same group made use of the canonical form in various circumstances ([3], [4], [5]). They derived a general proof of existence of the canonical form for a class of nonlinear discrete-time models including neural networks, and they provided a systematic procedure for deriving their canonical form, which was presented and published on various occasions ([6], [7]).

In [1], Tsoi and Tan provide, as the first figure of the paper, a drawing of the canonical form which looks like Figure 1 of [2], except for the fact that the name of the feedback variables is omitted, which makes the meaning of the drawing ambiguous. Indeed, they claim that the canonical network may be written as an input-output form:
$y(n) = f\,(y(n - 1), y(n - 2), ..., y(n - N), u(n), u(n - 1), ..., u(n - M))$.
This is not correct. Counterexamples, together with an in-depth investigation of the problems related to the derivation of a nonlinear input-output model, can be found in [8].

In addition, Tsoi and Tan argue that the canonical form, and the training algorithms, presented by Nerrand et al. in [2] are restricted to sigmoidal nonlinearities; again, this is definitely not correct. The canonical form has nothing to do with the nonlinearities of the neurons, and the training algorithms presented in [2] can be applied to networks of neurons with arbitrary (differentiable) nonlinearities (it is not even required that all neurons have the same non linearities).

In their conclusion, Tsoi and Tan argue that it was proved in [9] that any RNN architecture can be put in what *they* call a canonical form, which is actually an input-output form. We have shown above that this is not the case. Therefore, their claim that any RNN can be trained by the algorithm that they propose is disproved.

Since there is a lot of confusion, both in [1] and in [9], about the canonical form, we would like to insist that the following results were proved and published:
(i) *any recurrent neural network* can be transformed into a canonical form, as defined in [2] (which is nothing but the minimal state-space form); by *any recurrent network,* we mean discrete-time networks with any graph of connections, any delays between neurons (provided that there is no cycle whose sum of delays is equal to zero), and any nonlinearities for individual neurons (and possibly different nonlinearities for different neurons);
(ii) contrarily to the claim of Tsoi and Tan, there is a systematic procedure that produces the canonical form of any recurrent neural network [7].

Readers who are interested in the problems of recurrent network architectures are invited to read our related comment [10] on the related paper by Tsoi and Back [9] published in the same issue of Neurocomputing as [1].

Literature References

[1] A.C. Tsoi, S. Tan, Recurrent Neural Networks: a Constructive Algorithm, and its Properties, Neurocomputing 15 (1997), 309-326.

[2] O. Nerrand, P. Roussel-Ragot, L. Personnaz, G. Dreyfus, Neural Comput. 5 (1993) 165-199.

[3] I. Rivals, L. Personnaz, G. Dreyfus, J.L. Ploix, Modélisation, classification et commande par réseaux de neurones : principes fondamentaux, méthodologie de conception et illustrations industrielles, *in* Les réseaux de neurones pour la modélisation et la commande de procédés, J.P. Corriou, ed. (Lavoisier Tec & Doc, 1995).

[4] J.L. Ploix, G. Dreyfus, Knowledge-based Neural Modeling: Principles and Industrial Applications, *in* Industrial Applications of Neural Networks, F. Fogelman-Soulié, P. Gallinari, eds. (World Scientific, 1997).

[5] I. Rivals, L. Personnaz, Black-Box Modeling with State-Space Neural Networks, *in* Neural Adaptive Control Technology, R. Zbikowski, K.J. Hunt, eds. (World Scientific, 1996).

[6] G. Dreyfus, Y. Idan, Discrete-time Neural Networks and their Canonical Form, Machines that Learn Conference, Snowbird, Utah (1997).

[7] G. Dreyfus, Y. Idan, The Canonical Form of Discrete-Time Nonlinear Models, Neural Comput. 10 (1998) 133-164.

[8] I.J. Leontaritis, S.A. Billings, Input-output Parametric Models for Nonlinear Systems, parts I and II, Int. J. Control 41 (1985) 303-344.

[9] A.C. Tsoi, A.D. Back, Discrete Time Recurrent Neural Network Architecture, a Unifying Review, Neurocomputing 15 (1997) 183 - 223.

[10] L. Personnaz, G. Dreyfus, A Comment on "Discrete Time Recurrent Neural Network Architectures: a Unifying Review", Neurocomputing, submitted.

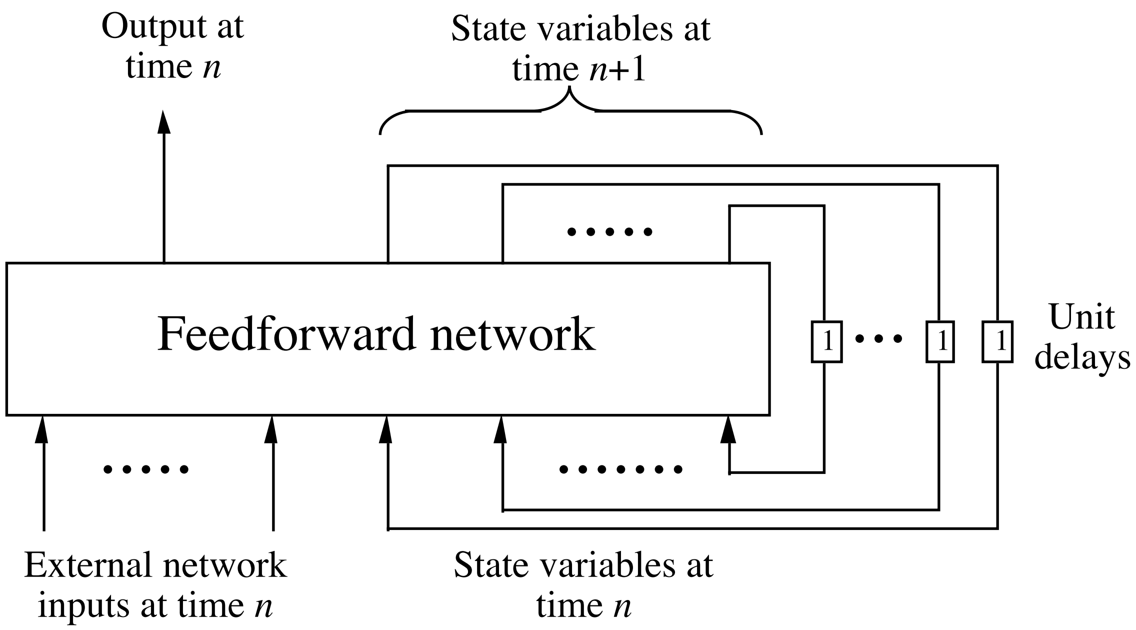
Output at
time n
State variables at
time n+1
Feedforward network
1
1
1
Unit
delays
External network
inputs at time n
State variables at
time n

Figure 1

Figure captions

Figure 1 : the genuine canonical form introduced by Nerrand et al.