Identification of Finite Automata Using Recurrent Neural Networks

*Sung Hwan Won, Cheol Hoon Park
School of Electrical Engineering and Computer Science
KAIST
e-mail : SungHwanWon@hotmail.com

Abstract

This paper demonstrates that the recurrent neural networks can be used successfully for the identification of finite automata (FAs). A new type of recurrent neural network (RNN) is proposed and the offline training algorithm, regulated Levenberg-Marquadt (LM) algorithm, for the network is developed. Simulation result shows that the identification and the extraction of FAs are practically achievable.

I. Introduction

The study of natural systems begins and ends with the specification of observables belonging to such a system, and a characterization of the manner in which they are linked [1]. Most of the systems in nature are nonlinear and dynamical, which means that by applying linear and static models, it is not probable to identify the systems in the real world. For decades, recurrent neural networks (RNNs) are successfully utilized to complete this kind of task and it was proven that RNNs can approximate any kind of nonlinear dynamical systems [2].

In case of identifying a nonlinear dynamical system of which time and state space are both discrete, we call the system as an automaton, and normally we deal with finite automata (FAs) which have a finite number of states. Systems with infinite number of states is not practical.

This paper deals with identifying FAs. We propose a new type of RNN that is adequate to identify FAs and a training algorithm. In the end of the paper, the simulation results are shown to prove that the proposed method works.

II. Proposed Network

Fig. 1(a) shows the Moore version of the network and Fig. 1(b) shows the Mealy version of it. $\mathbf{u}[t]$ is the input and $\mathbf{y}[t]$ is the output of the network at time t, and we specify $\mathbf{x}[t]$ as the state of the system at time t.

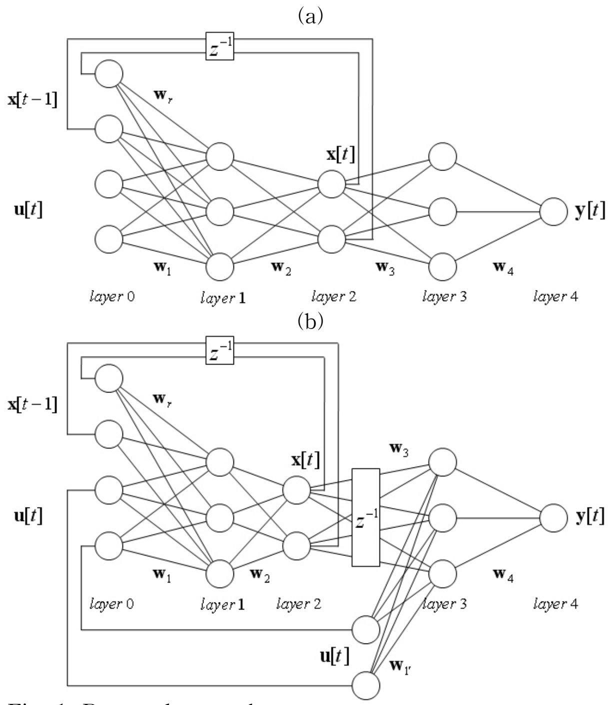

Fig. 1. Proposed network.

The output of the Moore version is

$$\mathbf{y}[t]=\varphi\{\mathbf{w}_4\varphi(\mathbf{w}_3\mathbf{x}[t]+\mathbf{b}_3)+\mathbf{b}_4\} \quad (1)$$

and the output of the Mealy version is

$$\mathbf{y}[t]=\varphi\{\mathbf{w}_4\varphi(\mathbf{w}_1\mathbf{u}[t]+\mathbf{w}_3\mathbf{x}[t-1]+\mathbf{b}_3)+\mathbf{b}_4\}, \quad (2)$$

where

$$\mathbf{x}[t]=\varphi\{\mathbf{w}_2\varphi(\mathbf{w}_1\mathbf{u}[t]+\mathbf{w}_r\mathbf{x}[t-1]+\mathbf{b}_1)+\mathbf{b}_2\}, \quad (3)$$

and φ is the transfer function of the neurons. The transfer function can be chosen properly. It can be linear, logistic, or hard-limited depending on the purpose. Here, we set it logistic.

Ⅲ. Training Algorithm

The Levenberg-Marquadt (LM) algorithm is known to be very effective algorithm on training neural networks including RNNs [3]. However, we need to change a little bit on the cost function because we need to have finite states. Therefore, the cost function should be

$$E=\sum_{t=0}^{T}\left(\|\mathbf{t}[t]-\mathbf{y}[t]\|^2+\|\mathbf{1}-\mathbf{x}^2[t]\|^2\right), \quad (4)$$

where $\mathbf{t}[t]$ is the target at time t. Thus, the Jacobian of the modified cost function is

$$J=-\left[\frac{\partial\mathbf{y}[t]}{\partial\mathbf{w}}\;2\frac{\partial\mathbf{x}[t]}{\partial\mathbf{w}}\mathbf{x}[t]\right]. \quad (5)$$

Ⅳ. Simulation Result

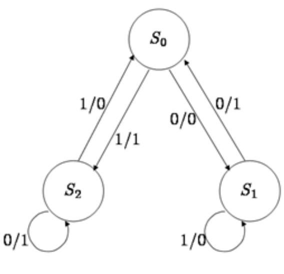

Fig. 2. A FA to be identified.

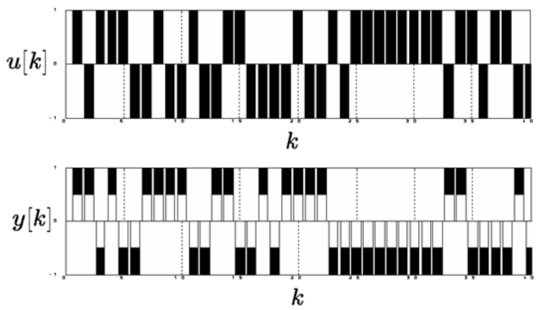

Fig. 3. The observed data and the actual output of a RNN.

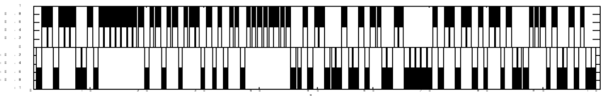

Fig. 4. The generalization performance test

The observed data from a FA in Fig. 2 is shown in Fig. 3. The observed input and output is depicted as black bars. The half of the output of the network is depicted as white bars. It can be seen that they are perfectly correspond.

Fig. 4 shows the generalization test of the network. We can see that the desired output (black bars) and the actual network output (white bars) correspond here, too. Therefore, we can say that the network identified the FA completely.

Ⅴ. Conclusion

We proposed the general method to identify FAs using RNNs. We also proposed the regulated LM algorithm instead of using heuristic algorithm which makes the training time long by applying the additional term and modified Jacobian. The simulation result proves that the proposed method is practically achievable.

References

[1] J. Casti, *Alternate Realities: Mathematical Models of Nature and Man*, New York: Wiley, 1989.

[2] E. D. Sontag, "Neural nets as systems models and controllers," in *Proceedings of 7th Yale Workshop Adaptive Learning Systems*, pp.73-79, 1992.

[3] M. T. Hagan and M. B. Menhaj, "Training Feedforward Networks with the Marquadt Algorithm," *IEEE Trans. on Neural Netw.*, vol. 5, no. 6, pp. 989-993, Nov. 1994.