Adaptive vs. Accommodative Neural Networks for Adaptive System Identification: Part II

James T. Lo
Department of Mathematics and Statistics
University of Maryland Baltimore County
Baltimore, MD 21250, U.S.A.
e-mail: jameslo@umbc.edu

Devasis Bassu
Telcordia Technologies
445 South Street,
Morristown, NJ 07960.
e-mail: dbassu@research.telcordia.com

***Abstract*— Adaptive neural networks (i.e. NNs with long- and short-term memories), and accommodative neural networks, which are recurrent NNs with fixed weights, are perhaps the most effective paradigms for general and systematic adaptive series-parallel system identification. Adaptive NNs involve less online computation, no poor local minima to fall into, and much more timely and better adaptation than neural networks with all their weights adjusted online. Accommodative NNs do not require online weight adjustment.**

Part I of this sequence of papers presented in IJCNN'01 reported that adaptive NNs have much better generalization ability than accommodative NNs in two numerical examples. In this Part, more comparison of the two paradigms is made for series-parallel identification of both deterministic and stochastic plants. Numerical examples show that although adaptive NNs consistently outperform accommodative NNs for generalization, the accommodative NNs have satisfactory generalization performances. However, in an example involving bifurcation and chaos, while the adaptive NN trained on periodic trajectories of the logistic dynamical system tracks accurately its chaotic trajectories, the accommodative NN trained better on the same data fails totally. As reflected in all the examples studied, the variability of the plant outputs seems to directly affect the generalization ability of the accommodative NN. Then an open question is: "How do we measure the variability of the plant outputs to determine whether an accommodative NN has adequate generalization ability for a given application?"

I. Introduction

Adaptive neural networks (NNs) (i.e. NNs with long- and short-term memories) were proposed for adaptive processing by the first-named author at ICNN'96 and ICNN'97. They were expected to have the online benefits of less computation, no convergence to poor local minima, and shorter transients as compared with the MLPs with all their weights adjusted online, which have been confirmed in [1]. On the other hand, properly trained recurrent NNs with fixed weights have been known to be able to adapt to an uncertain environment. To distinguish this ability with that of an adaptive system that adjusts any of its parameters for adaptation, the former ability is called accommodative ability. Intuitively speaking, the accommodative ability of an accommodative NN is a manifestation of its estimating implicitly the uncertain environmental process or a function thereof.

This work was supported in part by the National Science Foundation under Grant No.ECS-0114619 and the Army Research Office under Contract DAAD19-99-C-0031. The contents of this paper do not necessarily reflect the position or the policy of the Government.

The accommodative ability for adaptive filtering is a corollary of a main theorem for recursive neural filtering [2]. This ability was discussed in [3], [4]. The accommodative ability of recurrent NNs was also studied for engine idle speed control and time series prediction in [5]. In a companion paper also presented in IJCNN'03, a rigorous mathematical proof of the accommodative ability for series-parallel system identification is given.

This paper is a continuation of the paper [6] presented in IJCNN'01, in which adaptive NNs were shown to have superior generalization ability than accommodative NNs in identifying a linear deterministic plant with a cubic actuator and a deterministic exponential autoregressive plant in the series-parallel formulation of system identification. In fact, in these two examples, the accommodative NNs failed to generalize. In this paper, adaptive and accommodative NNs are compared on not only deterministic but also stochastic plants. The variability of the environmental parameters in these plants are smaller than those in the preceding paper [6]. The purpose is to see whether accommodative NNs can generalize at all in identifying deterministic as well as stochastic plants, for which adaptive recurrent NNs have to be used instead of adaptive multilayer perceptrons as the adaptive identifiers.

A deterministic linear plant with cubic actuation and a deterministic bilinear plant were each identified with an adaptive multilayer perceptrons and an accommodative recurrent multilayer perceptron. The environmental parameters were restricted to small ranges to hold the variability of the plant trajectories small. Numerical results show that adaptive NNs used as series-parallel identifiers have consistently better generalization ability than accommodative NNs. Nevertheless, the accommodative NNs are able to generalize reasonably well and track the plant outputs online rather closely.

A stochastic linear plant with cubic actuation and a stochastic bilinear plant, obtained by including random driving in the foregoing plants, were also identified with adaptive and accommodative NNs. The series-parallel identification involved is actually an adaptive neural filtering problem. Hence, adaptive recurrent NNs (with long- and short-term memories) have to be used instead of adaptive multilayer perceptrons as identifiers. Again, such adaptive identifiers are shown to consistently have better generalization ability than accommodative NNs for identifying both of the stochastic plants, and the accommodative

NNs have reasonably well online generalization performances.

It has been suspected that the variability of the plant outputs rather than the range of the environmental process affects generalization performances directly. The well-known logistic difference equation is identified with an adaptive NN and an accommodative NN. The logistic difference equation has asymptotically stable, periodic and chaotic trajectories over a small range $[1,4]$ of its parameter. The adaptive NN and the accommodative NN were trained on asymptotically stable data. The former generalizes successfully to track periodic and chaotic trajectories, but the latter fails completely.

The numerical examples in this paper as well as those in the preceding paper [6] prompt one open question: How do we measure the variability of the plant outputs to determine whether an accommodative NN has adequate generalization ability for a given application? More analytical understanding such as that presented in another paper by the first-named author in IJCNN'03 is desirable.

II. Identification of Deterministic Plants

A. Example 1. Linear system with a cubic actuator

Consider a simple first order autoregressive process given by

$$y_{t+1} = \theta_1 y_t + (u_t - \theta_1)\, u_t\, (u_t + \theta_2) \quad (1)$$

where u_t is the driver sequence and (θ_1, θ_2) is the environmental parameter of interest.

For the a priori training data, six values for the environmental parameter (θ_1, θ_2) given by $\Theta = \{(0.1,-1), (0.2,-1), (0.3,-1), (0.1,-1.5), (0.2,-1.5), (0.3,-1.5)\}$ are selected. Offline training is performed using an MLP with LASTMs with 2:12:7:1 architecture. The final offline RMSE value is noted to be 1.3962e-01. Next, an accomodative MLPWIN with 2:10:10:1 architecture is trained using the same a priori training data set. The final offline RMSE value for the accomodative MLPWIN is 1.3896e-01 (slightly lower than the final value of the trained MLP with LASTMs).

Online testing is performed using the values $\{(0.25,-0.9), (0.15,-1.25), (0.05,-1.6)\}$ for θ on both the ANNs. The results of this test are illustrated in figures 1, 2 and 3.

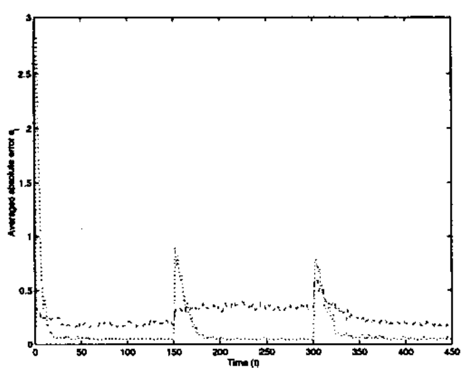

Fig. 1. Example 1: Average absolute error for accomodative MLPWIN and MLP with LASTMs

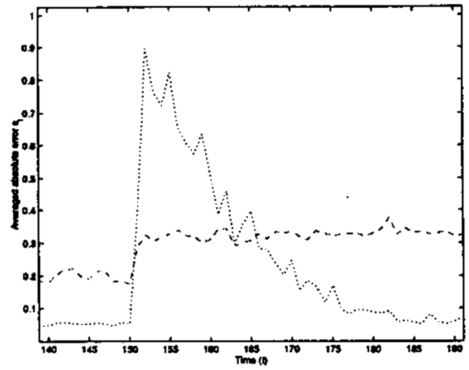

Fig. 2. Example 1: Average absolute error for accomodative MLPWIN and MLP with LASTMs (at time t=150)

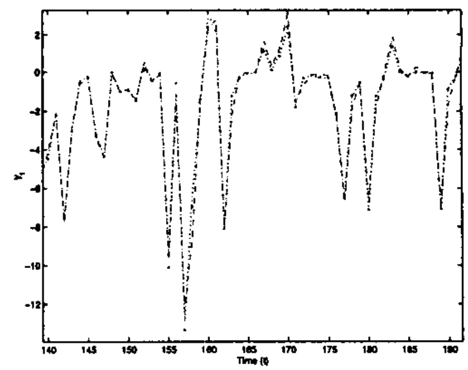

Fig. 3. Example 1: Change-point at time t=150

Figure 1 shows the online absolute errors for the accomodative MLPWIN and the MLP with LASTMs. The three peaks indicate the change-points at times = 1, 151, and 301 respectively. Figure 2 shows figure 1 zoomed at time t=150. Observe that is takes about 30 time points for the network to detect and adapt to the new value of θ. Figure 3 shows a typical realization of the output of the accomodative MLPWIN and the output of the MLP with LASTMs against the system output around the change-point at time t=150.

B. Example 2. Bilinear system

Consider the bilinear dynamic system given by

$$y_{t+1} = \theta_1 y_t + \theta_2 y_t u_t + u_t \quad (2)$$

where u_t is is the driver sequence and (θ_1, θ_2) is the environmental parameter of interest.

Six values of the environmental parameter (θ_1, θ_2), given by $\Theta = \{(0.1,0.25), (0.2,0.25), (0.3,0.25), (0.1,0.5), (0.2,0.5), (0.3,0.5)\}$ are selected for the a priori training data. Offline training is performed using an MLP with LASTMs with 2:10:7:1 architecture. The final offline RMSE value is 8.7842e-02. Next, an accomodative MLPWIN with 2:10:10:1 architecture is trained using the same a priori training data set. The final offline RMSE value for the accomodative MLPWIN is 8.3197e-02 (note that this is lower than the final value of the trained MLP with LASTMs).

Online testing is carried out using the values $\{(0.15, 0.3), (0.075, 0.35), (0.35, 0.2)\}$ for θ. The results of this test are illustrated in figures 4, 5 and 6.

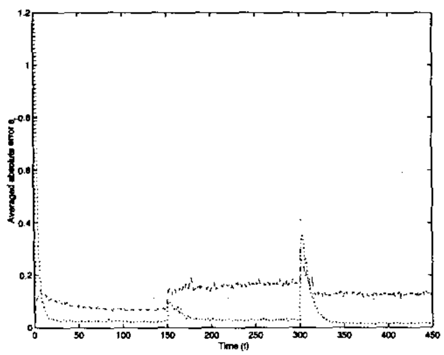

Fig. 4. Example 2: Average absolute error for accomodative MLPWIN and MLP with LASTMs

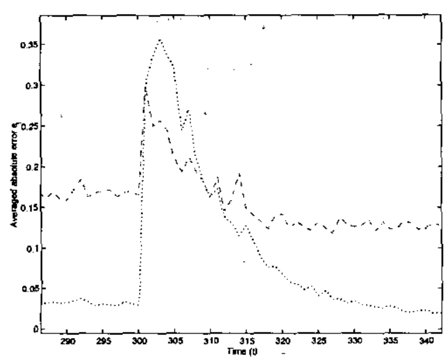

Fig. 5. Example 2: Average absolute error for accomodative MLPWIN and MLP with LASTMs (at time t=150)

Figure 4 shows the online absolute errors for the accomodative MLPWIN and the MLP with LASTMs. The three peaks indicate the change points at times = 1, 151, and 301 respectively. Figure 5 shows figure 4 at time t=150. In this case, the ANN take about 40 time steps to adjust to the new value of θ. Figure 6 shows a typical realization of the output of the accomodative MLPWIN and the output of the MLP with LASTMs against the system output at time t=150.

III. Identification of Deterministic Plants

A. Example 3. Series-parallel identification of a stochastic linear system with a cubic actuator

Consider the first order autoregressive dynamic system (1) with noise given by

$$y_{t+1} = \theta y_t + (u_t - \theta)\, u_t\, (u_t + 1) + \epsilon_t \qquad (3)$$

where ϵ_t are independent Gaussian noises with 0.4 standard deviation (approximately 10% of the signal), u_t is the driver input and θ is the parameter of interest.

Three exemplary values of the environmental parameter θ given by $\Theta = \{0.1, 0.3, 0.5\}$ are selected for forming

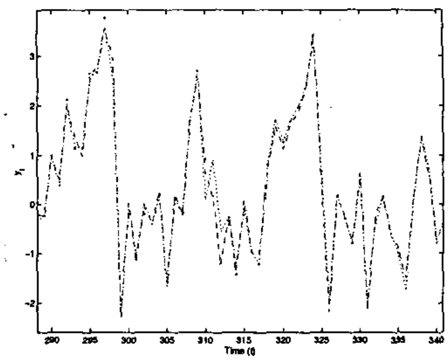

Fig. 6. Example 2: Change-point at time t=150

the a priori training data set. An MLPWIN with LASTMs with 2:5:5:1 architecture is used for offline training. The final RMSE value is recorded to be 3.690107e-03. Another accommodative MLPWIN with 2:8:8:1 architecture is trained using the same data set. The final RMSE value for the accommodative MLPWIN is 1.486838e-02 (further training was not possible without overfitting).

Online testing is performed using the values $\{0.2, 0.4, 0.6\}$ for θ. The results of this test are illustrated in figures 7 and 8.

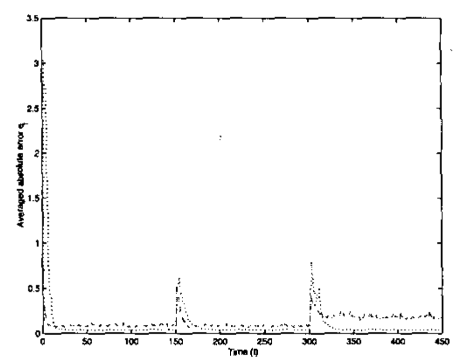

Fig. 7. Example 3: Average absolute error for accommodative MLPWIN and MLPWIN with LASTMs

Fig. 7 shows the online absolute errors for the accommodative MLPWIN and the MLPWIN with LASTMs. The changes points occur at times = 1, 151 and 301 respectively. Observe that the RMSE for the accomodative MLPWIN goes up by a significant amount as θ is pushed to 0.6 to test the generalization ability of the ANNs. This can be clearly seen in Fig. 8 which shows Fig. 7 around time $t = 300$.

B. Example 4. Series-parallel identification of a stochastic bilinear system

Consider the bilinear system (2) with noise given by

$$y_{t+1} = \theta_1 y_t + \theta_2 y_t u_t + u_t + \epsilon_t \qquad (4)$$

where u_t is the driver input, $\theta = (\theta_1, \theta_2)$ is the parameter of interest and ϵ_t are independent Gaussian noises with 0.135

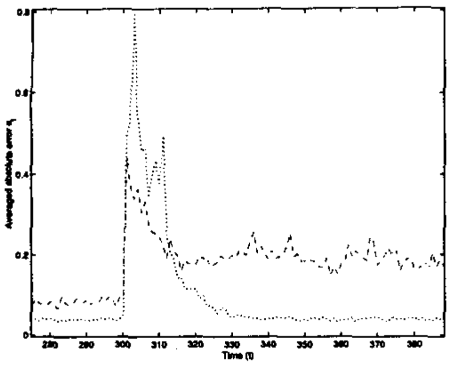

Fig. 8. Average absolute error for accommodative MLPWIN and MLPWIN with LASTMs (at time t= 300)

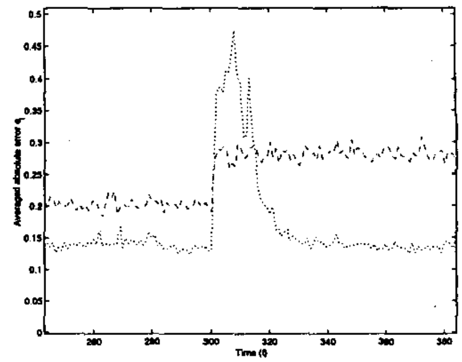

Fig. 10. Example 4: Average absolute error for accommodative MLPWIN and MLPWIN with LASTMs (at time t= 300)

standard deviation (approximately 10% of the standard deviation of the signal).

Six values of θ given by $\Theta = \{(0.1, 0.25), (0.2, 0.25), (0.3, 0.25), (0.1, 0.5), (0.2, 0.5), (0.3, 0.5)\}$ were selected for obtaining the a priori training data. An MLPWIN with LASTMs with 2:6:1 architecture is trained offline using this data set. The final RMSE value is recorded to be 1.425569e-02. Next, an accommodative MLPWIN with 2:10:10:1 architecture is trained using the same data set. The final RMSE value for the MLPWIN is 1.752183e-02 (beyond which the training started to overfit the data).

Online testing is carried out using the values $\{(0.25, 0.3), (0.15, 0.4), (0.35, 0.2)\}$ for θ. The results of this test are illustrated in Fig. 9 and Fig. 10.

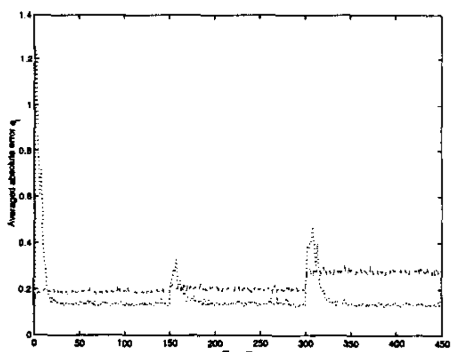

Fig. 9. Example 4: Average absolute error for accommodative MLPWIN and MLPWIN with LASTMs

Fig. 9 shows the online absolute errors for the accommodative MLPWIN and the MLPWIN with LASTMs. The changes points occur at times = 1, 151 and 301 respectively. Observe that the RMSE for the accomodative MLPWIN goes up by a significant amount as θ is pushed to $(0.35, 0.2)$ to test the generalization ability of the ANNs. This can be seen in Fig. 10 which shows Fig. 9 around time $t = 300$.

The following examples look at systems where the environmental parameter *theta* has high variability as described at the beginning of this section.

IV. Identifying the Logistic System

A. *Example 5. Series-parallel identification of the logistic equation*

Consider the logistic equation given by

$$y_{t+1} = \theta y_t (1 - y_t) \tag{5}$$

where $0 \le \theta \le 4$ and the initial condition $y_0 \in (0, 1)$. In this case, there is no driver sequence u_t. The system exhibits different dynamics based on the value of θ as listed below.

1) $0 \le \theta \le 1$: $y = 0$ is the asymptotically stable point
2) $1 \le \theta \le 3$: $y = \frac{\theta - 1}{\theta}$ is asymptotically stable point
3) $\theta = 3$: Transition from stability to oscillation
4) $3 < \theta \le 3.449$: Oscillation with period 2
5) $3.449 < \theta \le 3.5699$: Oscillation with period doubling as θ increases (bifurcation)
6) $\theta > 3.5699$: Chaotic

The MLPWIN with LASTMs is not capable of handling chaotic systems. Further, for $0 \le \theta < 3$, the solution is asymptotically stable and stabilizes very fast for nearly all values in the specified region. It would be no problem for the short term memory to adapt to this asymptotic value (using the bias term only and setting all other weights to zero). For such reasons, the system (5) is studied for values of $3 < \theta \le 3.5699$.

Three values of θ given by $\Theta = \{0.8, 1.5, 2.8\}$ are choosen for a priori training. The training data set comprises of 200 sequences, each containing 43 consecutive I/O pairs corresponding to one of the three θ values. The initial condition y_0 is selected uniformly from (0.0001, 0.9999). Offline training is performed using a MLP with LASTMs with 1:4:1 architecture. The final RMSE is recorded to be 9.020597e-03. Next, an accomodative MLPWIN with 1:10:1 architecture is trained using the same training data set. Several values of 3, 5, 10 and 15 are used for priming. The final RMSE (corresponding to a priming length of 5) is 8.828213e-03.

Online testing is performed using the values $\{1.5, 3.5, 2.7, 3.2, 3.65\}$ for θ. Each test sequence is 500 long consisting of 100 consecutive I/O pairs for each θ value in the given order. The initial condition y_0 is selected uniformly from [0.05,0.5]. The results of this test are illustrated in figures 11, 12, 13 and 14.

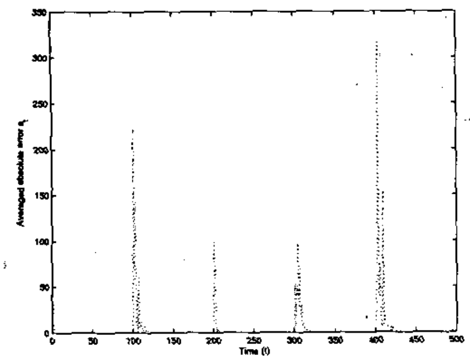

Fig. 11. Example 5: Average absolute error for MLP with LASTMs

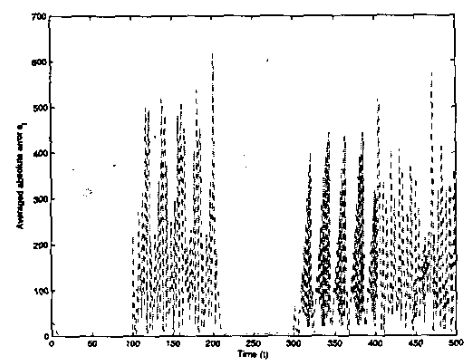

Fig. 12. Example 5: Average absolute error for accomodative MLPWIN

Figures 11 and 12 show the absolute error during online testing for the MLP with LASTMs and the accomodative MLPWIN respectively, averaged over 500 test sequences. Fig. 13 is blown-up version of Fig. 11 at time t=400. The MLP with LASTMs takes about 20 steps to adjust itself to the new value of θ. Fig. 14 shows a typical realization of the MLP with LASTMs against the plant output. The output for the accomodative MLPWIN is not included because of its poor performance during online testing.

V. Conclusion

Numerical feasibility and comparison of adaptive and accommodative NNs for series-parallel identification of dynamical systems have been studied and the results are reported in this paper and its preceding Part I. The results show that both types of NN are satisfactory series-parallel identifiers of both deterministic and stochastic plants with small variabilities caused by uncertain environmental parameters. Nevertheless, adaptive NNs have consistently better generalization capability. Moreover, when the variability of a plant under identification is large, accommodative NNs usually fail to generalize.

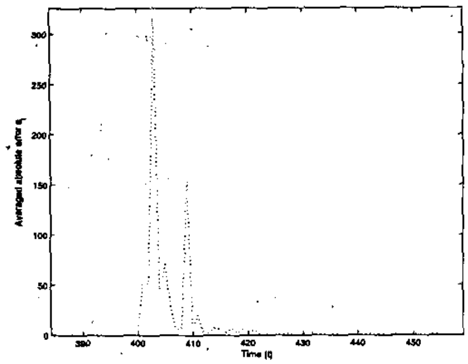

Fig. 13. Example 5: Average absolute error for MLP with LASTMs (at time t=150)

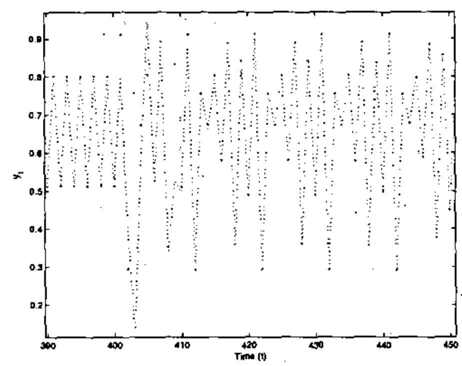

Fig. 14. Example 5: Change-point at time t=150

However, accommodative NNs do not require online weight adjustment. If online computation must be minimized and especially if there is no data for online weight adjustment, accommodative NNs are a good or only choice. We conclude that both adaptive and accommodative NNs are viable paradigms for series-parallel identification of both deterministic and stochastic dynamical system. Selection between them for an application depends on the variability of the plant involved, generalization performance required, amount of online computation allowed, and training data available online.

References

[1] J. T. Lo and D. Bassu, "Adaptive multilayer perceptrons with long- and short-term memories," *IEEE Transactions on Neural Networks*, vol. 13-1, pp. 22–33, January 2002.

[2] J. T. Lo, "Synthetic approach to optimal filtering," *IEEE Transactions on Neural Networks*, vol. 5, pp. 803–811, September 1994.

[3] ——, "Neural network approach to optimal filtering," Rome Laboratory, Air Force Material Command, Tech. Rep. RL-TR-94-197, 1994.

[4] J. T. Lo and L. Yu, "Adaptive neural filtering by using the innovations process," *Proceedings of the 1995 World Congress on Neural Networks*, vol. II, pp. 29–35, July 1995.

[5] L. A. Feldkamp and G. V. Puskorious, "Training of robust neural controllers," in *Proceedings of the 33rd Conference on Decision and Control*, Lake Buena Vista, Florida, December 1994, pp. 2754–2759.

[6] J. T. Lo and D. Bassu, "Adaptive versus accommodative neural networks for adaptive identification of dynamic systems," in *Proceedings of the 2001 International Joint Conference on Neural Networks*, Washington, D.C., July, 2001.