An N-Limit Cycle Artificial Neural Network for Multivalued Logic

T. Yang, N. Pimchaipong, R. W. Newcomb, and R. deFigueiredo
Microsystems Laboratory
Electrical Engineering Department
University of Maryland
College Park, MD 20740, USA
email:newcomb@eng.umd.edu

Abstract

A neuron consisting of a second order system with n limit cycles is introduced. With different initial conditions this neuron can produce n different outputs allowing the neuron to be used for multivalued logic such as an n-limit cycle artificial neural network. After a review of the n-limit cycle oscillator the system configuration is presented. Data is inserted via weights as initial conditions on the second order oscillator giving the neuron. For m neurons the neural network can store m to the nth power states in an n-level logic.

1 Introduction

Although for the past decades artificial neural network research has been prosperous, the elements of artificial neural networks, namely neuron models, are still weakly linked with realistic ones, not only the models of neurons but also the still-limited knowledge of our own neural systems [1, 2]. Among those approaches, Hopfield neural networks are promising ones.

Hopfield neural networks [3, 4] are dynamical systems which process the initial condition information over time while moving through a sequence of states. In such systems neurons with either a hard-limit activation function produce binary outputs or with a continuous activation function provide analog values. The binary output provides limited information about the neural network, but the analog ones are sensitive to noise. A multivalued logic, a hybrid of binary logic and the analog signals, retains both advantages [5, 6].

After reviewing the n-limit cycle oscillator of [7], a new neuron with n states which can produce multivalued outputs is presented. The neuron uses a second order oscillator with n limit cycles. With different initial conditions the second order oscillator can produce n different limit cycles which may be utilized as n different outputs for the neuron. These n different outputs can be applied to multivalued logic and classification in artificial neural networks.

In the following section we first introduce the neuron and its basic structure. In section 3, we prove that the new neuron has the ability of classification with its multivalued output by using SIMULINK for simulation. Conclusions are made in the final section.

2 Basic Structure of the Neuron with n Limit Cycles

In this section we introduce the basic structure for our neuron which includes the n-limit cycle oscillator, an input stage with weights which sets initial conditions, a peak value detector, and an output classifier.

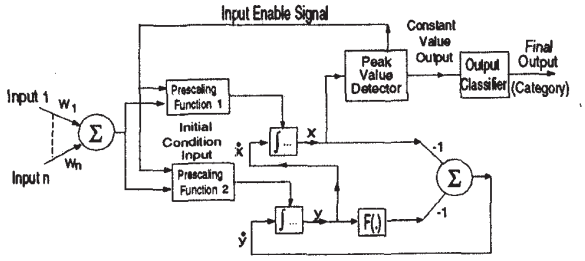

Figure 1: The basic structure of the new neuron

2.1 The Neuron

The new neuron, as shown in figure 1, includes an n-limit cycle oscillator with input weight stage, a peak value detector, and an output classifier. In this figure, the mathematical expression for the oscillator will be presented and explained in details in next subsection.

0-7803-3073-0/96/$5.00

The neuron works as follows. The n-limit cycle oscillator receives its initial conditions from its weight stage first. After the peak value detector detects the periodic oscillation peak value, the peak value detector passes the peak value to the output classifier. After classifying the peak values, the output classifier outputs the final real number value of the neuron.

This neuron can produce n different outputs because of the n peak values of the oscillator. According to different initial conditions from the input state the n-limit cycle oscillator can switch its oscillations among these limit cycles. With the switch ability among n limit cycles the neuron can produce n different outputs which are applicable for multivalued logic and signal classification.

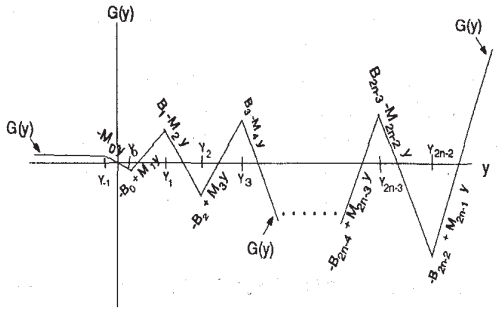

Figure 2: A piecewise linear function $G(y)$

2.2 The N-limit Cycle Oscillator

The core of the new neuron is the n-limit cycle oscillator which results from two nonlinear differential equations. According to deFigueiredo [8], the Liénard nonlinear differential equations,

$$\frac{dx}{dt} = y \tag{1-a}$$

$$\frac{dy}{dt} = -F(y) - x, \tag{1-b}$$

have n periodic solutions, where $F(.)$ is a real-valued, continuous, and locally Lipschitzian function. In [7], we designed a two-limit cycle oscillator by adapting the theory in [8] to the case of piecewise linear nonlinearities, $G(.)$, for which the same design criteria hold.

If we design equation (1) to have n limit cycles with the case of the origin unstable, the criteria for the piecewise linear function $G(y)$ which is shown in figure 2 would be described as follows:

1. Choose $0 < M_0 < min\{2, M_k\}$, $0 < M_k < 2$, $Y_{-1} < 0$ and $Y_0 > 0$, where $k = 1, \ldots, 2n-1$. Choose $G(Y_{-1}) > 0$ and for $y \leq Y_{-1}$, the constant value
$$G(y) = G(Y_{-1}) \tag{2}$$
Calculate $N_k(M_0, M_k)$ according to the appendix of [7].
2. For $Y_{-1} < y \leq Y_0$, $G(y)$ passes the origin by the choice
$$G(y) = -M_0 y \tag{3}$$
3. Decide B_0 by
$$B_0 = (M_0 + M_1)Y_0 \tag{4}$$
4. For each $i = 1, \ldots, 2n-2$, in sequence choose Y_i such that
$$Y_i \geq N_i B_{i-1} > Y_{i-1} \tag{5}$$
Since $Y_i = y_i$ for $i > 0$ this decides the y_k along the y axis. To optimize the design, Y_i can be set equal to $N_i B_{i-1}$. Calculate B_i according to
$$B_i = -B_{i-1} + (M_i + M_{i+1})Y_i \tag{6}$$
and repeat for the next i.

Three limit cycles with their piecewise linear function $G(y)$ are shown in figure 3.

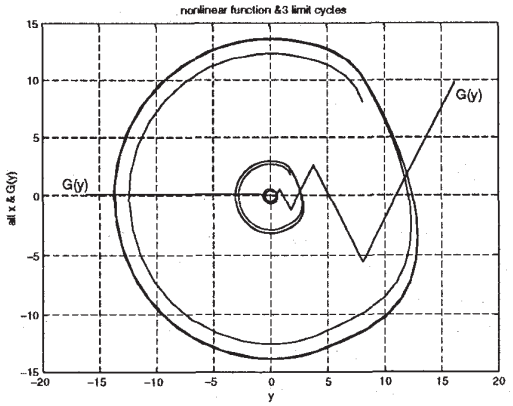

Figure 3: An example of three limit cycles

2.3 Input Weight Stage

There are two stages for the input. First the neuron receives the input signals via the the weights. Then by a pre-scaling function the input values are mapped into amplitudes consistent with desired initial conditions for the oscillator. The weights can be found by the standard weight setting techniques used for neural networks. A training rule needs to be used for the purpose.

2.4 Peak Value Detector

The purposes of the peak detector are to filter the false peaks from the initial oscillation, produce different constant output according to the values of peaks, and signal the input stage of the neuron to accept the new input.

The algorithm for the peak detector is described below, where $signal(t)$ is from the oscillator.

1. $signal(t_0) = 0$ and $peak(t_0) = 0$, where t_0 is the initial time.

2. If $signal(t) > 0$, go to 3.

3. Compare if $signal(t) > signal(t-1)$,
   true, then $peak(t) = signal(t)$ and repeat the comparison.
   false, $peak(t)$ found and go to 4.

4. Check $|peak(t) - peak(t-1)| \leq$ tolerant error
   true, $signal(t)$ oscillates in periodic,
   pass the peak(t) to output classifier,
   set $peak(t) = 0$ and $signal(t) = 0$,
   neuron receives new input.
   false, repeat 2.

After the peak value is found, the value can be useful for the specific output. Alternatively, the limit cycles themselves can be used as outputs.

2.5 Output Classifier

The property of the n-limit cycle oscillator to produce n different outputs can be used to classify n different things. For m neurons, n^m things can be classified. After receiving the peak values from the peak value detector, output classifier can assign these peak values by different categories for special purposes.

3 Example

In this section the new neuron with 5 limit cycles is used to classify distinct pairs, (a, b), of four numbers within a set,

$$S = \{0, 1, 2, 3\}. \tag{7}$$

The neuron, fed arbitrarily pairs of different elements in S, can classify the pair by outputting the summation of each pair, $sum = a + b$. The output is also the category which the pair belongs to. Pairs of S need to be classified into the following five categories:

$(0,1)$ for category 1, output 1
$(0,2)$ for category 2, output 2
$(0,3), (1,2)$ for category 3, output 3
$(1,3)$ for category 4, output 4
$(2,3)$ for category 5, output 5
output 0 for unknown state.

In the above case, an n-limit cycle oscillator with five different limit cycles is designed according to subsection 2.2. With $n = 5$, the oscillator is produced with $M_0 = 0.1$, $M_k = 1.9$, $Y_{-1} = -0.5$, and $Y_0 = 0.5$. There are two inputs which are $Input_1 = a$ and $Input_2 = b$. The weights are set to 1. The same pre-scaling function is used here for two initial conditions of the neuron.

The pre-scaling function is roughly mapped by possible pair summation which is sum into different regions for different limit cycles. If $sum = 1$, then map to 0.8, the region for the first limit cycle. If $sum = 2$, then map to 3.5, the region for the second limit cycle. If $sum = 3$ which are $(0,3)$ and $(1,2)$, then map to 14, the region for the third limit cycle. If $sum = 4$, then map to 60, the region for the fourth limit cycle. If $sum = 5$, then map to 280, the region for the fifth limit cycle. To meet the above requirements, the pre-scaling function $f(.)$, a piecewise linear function, is presented as follows:

$$f(sum) = \begin{cases} 0.2, & sum \leq 0 \\ 0.6sum + 0.2, & 0 < sum \leq 1 \\ 2.7sum - 1.9, & 1 < sum \leq 2 \\ 10.5sum - 17.5, & 2 < sum \leq 3 \\ 46sum - 124, & 3 < sum \leq 4 \\ 220sum - 820, & 4 < sum \leq 5 \\ 280, & sum > 5 \end{cases} \tag{8}$$

Zero is set for the initial output when the simulation initially starts, and no peak is found. The output classifier specifies the categories according to the peak values.

The simulation was run by SIMULINK. The input pairs which are $(1,0)$, $(2,0)$, $(3,0)$, $(3,1)$, $(2,1)$, and $(2,3)$ in sequence are repeatedly shown in figure 4, and all possible pairs are fed into the neuron. Figure 5 shows the sequence of the outputs in the heavier, dark line with the inputs from figure 4 in the dotted or lighter lines. The output is initially set to zero and holds its value untill the peak detector finds the peak; afterwards, the output shows a new value and the neuron allows new inputs. Therefore, there is always a period of time delay for the output to show the result of the inputs. The output values are the input pairs' categories and maintain their previous values untill new output.

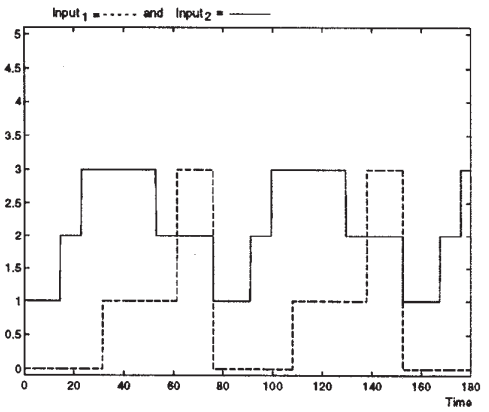

Figure 4: The $Input_1(t)$ and $Input_2(t)$

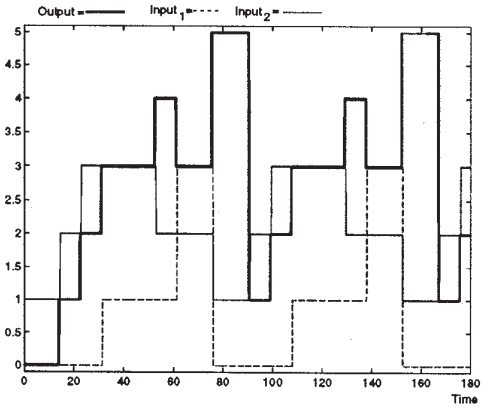

Figure 5: The outputs from the neuron for classification with inputs

The results indicate that the neuron shows its ability of classification and also multivalued output.

4 Conclusions

In this paper we present a new neuron with n limit cycles. The basic structure is introduced. With the n-limit cycle oscillator, this neuron can produce n different values which can be used for multivalued logic or classification in artificial neural networks. As shown in the example, the neuron has the ability of classification and multivalued output.

References

[1] J. M. Zurada, *Introduction to Artificial Neural Systems*, West Publishing Company, St. Paul, 1992.

[2] J. Dayhoff, *Neural Network Architectures – An Introduction*, Van Nostrand Reinhold International Company Limited, New York, 1990.

[3] D. W. Tank and J. J. Hopfield, "Simple "Neural" Optimization Networks: an A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit," *IEEE Transactions on Circuits and Systems*, vol. CAS-33, no. 5, pp. 533–541, May 1986.

[4] J. J. Hopfield and D. W. Tank, ""Neural" Computation of Decision in Optimization Problems," *Biological Cybernetics*, vol. 52, no. 3, pp. 141–152, July 1985.

[5] K. C. Smith, "The Prospects for Multiple-Valued Logic: A Technology and Applications View," *IEEE Transactions on Computer*, vol. C-30, no. 9, pp. 619–632, September 1981.

[6] K. W. Current, "Current-Mode CMOS Multiple-Valued Logic Circuits," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 2, pp. 95–107, February 1994.

[7] T. Yang, S. Hadjipanteli, R. W. Newcomb, and R. deFigueiredo, "Two-Limit-Cycle Piecewise Linear Oscillator," *Proceedings of the IEEE Singapore International Conferences on Intelligent Control and Instrumentation*, Singapore, June 1995, pp. 232–236.

[8] R. J. P. deFigueiredo, "On the Existence of N Periodic Solutions of Liénard's Equation," *Nonlinear Analysis: Theory, Methods and Applications*, vol. 7, no. 5, pp. 483–499, May 1983.