第21卷第12期 半 导 体 学 报 Vol. 21, No. 12
2000年12月 CHINESE JOURNAL OF SEMICONDUCTORS Dec., 2000

Circuit Design of On-Chip BP Learning Neural Network with Programmable Neuron Characteristics*

LU Chun(卢 纯), SHI Bing-xue(石秉学) and CHEN Lu(陈 卢)

(*Institute of Microelectronics, Tsinghua University, Beijing 100084, China*)

Abstract: A circuit system of on-chip BP(Back-Propagation) learning neural network with programmable neurons has been designed, which comprises a feedforward network, an error back-propagation network and a weight updating circuit. It has the merits of simplicity, programmability, speediness, low power-consumption and high density. A novel neuron circuit with programmable parameters has been proposed. It generates not only the sigmoidal function but also its derivative. HSPICE simulations are done to a neuron circuit with level 47 transistor models as a standard 1.2μm CMOS process. The results show that both functions are matched with their respective ideal functions very well. The non-linear partition problem is used to verify the operation of the network. The simulation result shows the superior performance of this BP neural network with on-chip learning.

Key words: hardware implementation of neural networks; CMOS analogue integrated circuits; programmability

EEACC: 1295; 1285; 2570D

CLC number: TN492 **Document code**: A **Article ID**: 0253-4177(2000)12-1164-06

1 Introduction

Artificial neural networks with a Back-Propagation (BP) algorithm present a practical approach to various problems. The hardware implementation is very necessary and essential because of the normal requirements of many applications. Hardware implementatin of BP neural networks can be achieved in several ways, including off-chip learning, chip-in-the-loop learning and on-chip learning. Which one to be chosen is not always clear-cut in practice, and the answer depends on not only the application, but also the topology of a net-

* Project Supported by National Natural Science Foundation of China (Under Grant No. 69636030).

LU Chun (卢 纯), Ph. D. candidate. Her work focuses on the analog IC design and VLSI implementation of artificial neural networks. E-mail: luch@dns.ime.tsinghua.edu.cn

SHI Bing-xue(石秉学), Professor. His present research interests include digital/analog IC design, artificial neural networks, fuzzy logic system and their VLSI implementation.

CHEN Lu(陈 卢), Ph. D. candidate. His work focuses on DC-DC converter and VLSI implementation of artificial neural networks.

Received 18 June 2000, revised manuscript received 1 August 2000

work and its different constrains as well. On-chip learning is imperative if the system meets the following requirements[1—5]: (1) high speed, (2) autonomous operation in an unknown and changing environment, (3) small volume, (4) reduced weight.

One of the neural networks' important components is the neuron, whose performance and complexity greatly affect the whole net. In many literatures, its activation function is found to be sigmoid. In the on-chip back-propagation learning, the non-linear function and its derivative are both required. A simple neuron circuit[6], which can realize both the neuron activation function and its derivative, has been proposed in this paper. Having current inputs and voltage outputs, the neuron is built with strong-inversion biased transistors. Furthermore, the circuit enables the threshold and the gain factor to be adjustable.

2 Circuit System Architecture

The BP network includes the input layer, the hidden layer(s) and the output layer. Each layer has several neurons. The transfer function of each neuron is always a sigmoid one as expressed in Eq. 1,

$$f(\boldsymbol{S}) = \frac{1}{1 + \exp(-\alpha(\boldsymbol{S}) + \theta)} \tag{1}$$

where $\boldsymbol{S} = \boldsymbol{X} \cdot \boldsymbol{W}$, $\boldsymbol{X}$ is the input matrix, $\boldsymbol{W}$ is the weight matrix, α is the gain factor and θ is the threshold vector. It is supposed that R is the number of the training set elements, w_{ij}^{l} is the weight between the ith ($0 \leqslant i < n$) neuron of the $(l-1)$th layer and the jth neuron of the lth ($l = 1, 2, \cdots, L$) layer, and θ_j^l is the threshold of the jth neuron of the lth layer. For the sake of convenience, let $\theta_j^l = w_{nj}^{l}$ and $x_n^{l-1} = 1$. To a certain training sample r ($r = 1, 2, \cdots, R$), $x_{i,r}^{l-1}$ is the output of the ith neuron of the $(l-1)$th layer; $x_{j,r}^{l}$ is that of the jth neuron of the lth layer; $t_{j,r}$ is the target value when $l = L$; $s_{j,r}^{l}$ is the weighted sum from the ith neurons of the $(l-1)$th layer to the jth neuron of the lth layer. The feedforward calculation can be expressed as follows,

$$x_{j,r}^{l}(k) = f(s_{j,r}^{l}(k)) = f\left[\sum_{i=0}^{n} w_{ij}^{l}(k) x_{i,r}^{l-1}(k)\right] \tag{2}$$

To describe the error back-propagation process, several definition should be made first. The neuron error is defined as,

$$\epsilon_{ij,r}^{l}(k) = \begin{cases} t_{j,r} - x_{j,r}^{l}(k), & l = L \\ \sum_j w_{ij}^{l+1}(k)\,\delta_{ij,r}^{l+1}(k), & 1 \leqslant l < L \end{cases} \tag{3}$$

where the weight error is defined as,

$$\delta_{ij,r}^{l}(k) = f'(s_{i,r}^{l}(k))\,\epsilon_{ij,r}^{l}(k) \tag{4}$$

Then the weight updating rule can be expressed as Equation (5),

$$w_{ij}^{l}(k+1) = w_{ij}^{l}(k) + \eta \sum_{r=1}^{R} \delta_{ij,r}^{l}(k)\, x_{j,r}^{l}(k) \tag{5}$$

where η is the learning rate, $\Delta w_{ij}^{l}(k+1) = \sum_{r=1}^{R} \delta_{ij,r}^{l}(k)\, x_{j,r}^{l}(k)$ is the weight change.

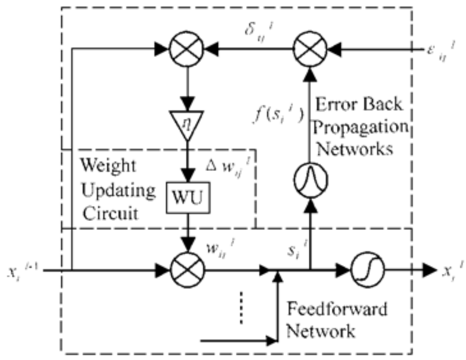

FIG. 1 Diagram of Circuit System

The circuit system is designed according to the algorithm above. It comprises a feedforward network, an error back-propagation network and a weight updating circuit, as shown in Figure 1. In the feedforward network, the synapse is realized by the Gilbert multiplier, as is simple and area-economic. The nonlinear I-V transfer function is achieved by the neuron. Using the forward difference method, the neuron generates a sigmoidal functin with its derivative. The latter is used in the error back-propagation network. The Weight Unit (WU) implements the weight update operation, with the diagram shown in Fig. 2. A 7-bit ADC is used to convert the analog weight change signal to a digital one and is added to the 12-bit weight. The new weight is converted to an analog signal by a DAC for the next feedforward calculation and stored in the RAM for the next weight updating.

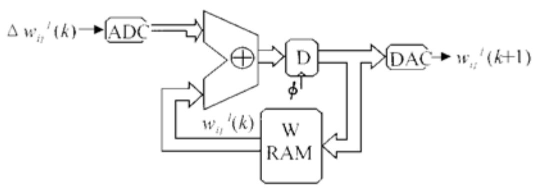

FIG. 2 Diagram of Weight Updating Unit

3 Neuron Circuit

Figure 3(a) shows the schematic diagram of the proposed neuron circuit. V_{dd} is the 3.3V-voltage source. V_{out1} outputs the sigmoidal activation function. The approximate derivative can be obtained via ($V_{out1} - V_{out2}$). In the dash frame, the fixed voltage V_{ref1} is carefully chosen so that both the transistors M1 and M2 work in their respective linear ranges. The formed linear resistor R_{AB} can be controlled by the gate voltage of both transistors V_N and V_P. In the dash dot frame, a simple differential pair composed of identical transistors and the active loads makes the actual sigmoidal non-linear. One port of the differential pair is connected to point B and the other to a fixed voltage V_{ref2} or $V_{ref2} - \Delta V$, where ΔV is a small fixed voltage. I_{ref1} and I_{ref2} are the fixed current sources.

Assuming that M3, M4 operate in saturation and follow the ideal square law, we have

$$I_{ref} = \frac{\beta}{2}[(V_B - V_C)^2 + (V_D - V_C)^2] \quad (6)$$

where β is the transconductance parameter for the transistors M3 and M4. The voltage of point C can be obtained from Eq. 7.

$$V_C = \frac{(V_B + V_D) - \sqrt{\frac{4I_{ref2}}{\beta} - (V_B - V_D)^2}}{2} \quad (7)$$

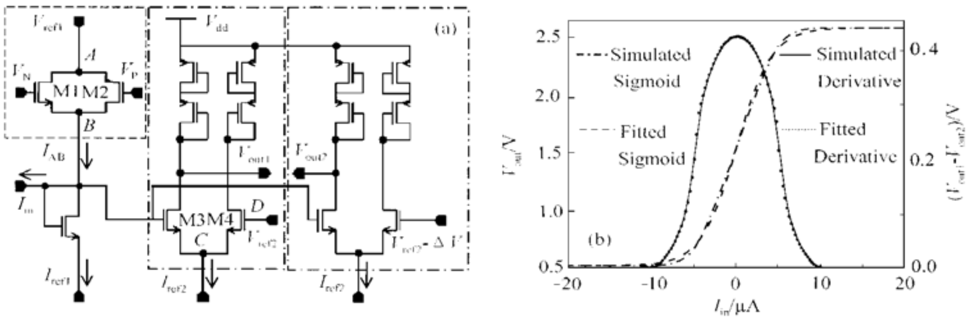

FIG. 3 Neuron

(a) Circuit, (b) Simulated Curves Compared with Fitted Ones

Assuming that V_d is the input differential voltage, i. e. $V_d = V_B - V_D$, then

$$I_{d3}(V_d) = \frac{\beta}{2}(V_B - V_C)^2 = \frac{I_{ref2}}{2} + \frac{\beta V_d \sqrt{\frac{4I_{ref2}}{\beta} - V_d^2}}{4} \tag{8}$$

$$I_{AB} = I_{in} + I_{ref1}$$

When I_{in} is small, $V_d > \sqrt{\frac{2I_{ref2}}{\beta}}$, V_{out1} remains the low saturation voltage. As I_{in} is increasing, V_B descends tardily and V_{out1} increases slowly. When $V_d < -\sqrt{\frac{2I_{ref2}}{\beta}}$, V_{out} reaches the high saturation level and remains.

The dash dot line in Fig. 3(b) shows the HSPICE simulation result of the neuron activation function, with level 47 transistor models being a standard 1.2μm CMOS process. The fitted sigmoid curve is also shown in Fig. 3(b) by the dash line. The relative error is not more than 3%.

Assuming that $V_{out} = V_{out}(I_{in})$, which is the generated neuron activation function, with the forward difference method, the approximate derivative voltage V_{deriv} can be achieved by subtracting V_{out2} from V_{out1}, as below,

$$\begin{aligned} V'_{out}(I_{in}) &= V'_{out}(V_d) \times V'_d(I_{in}) \\ &\cong -\frac{V_{out}(V_B - V_{ref2} + \Delta V) - V_{out}(V_B - V_{ref2})}{\Delta V} \times R_{AB} \end{aligned} \tag{9}$$

$$\begin{aligned} V_{deriv}(I_{in}) &\equiv \frac{\Delta V}{R_{AB}} \times V'_{out}(I_{in}) \\ &\cong -(V_{out}(V_B - (V_{ref2} - \Delta V)) - V_{out}(V_B - V_{ref2})) \\ &= V_{out1} - V_{out2} \end{aligned} \tag{10}$$

The derivative observed from simulation and that from the simulated neuron activation function are shown as the solid line and the dot line in Fig. 3(b), respectively. The relative error between them is less than 5%.

The great power of an artificial neural network derives from its adaptability to the unknown and changing environment. Therefore, good programmability is of the fundamental importance[8,9]. Different application need different gain factor α and threshold vector θ, which can be obtained by varying I_{ref1}, V_N and V_P.

The threshold vector θ can be adjusted by changing the reference current I_{ref1}. When I_{ref1} increases, the current I_{in} will decrease, which is needed to satisfy that $V_B - V_{ref2} > \sqrt{2I_{ref2}/\beta}$ so that the activation curve will shift to the left. Otherwise, it will shift to the right. Figure 4(a) shows the simulated neuron transfer functions with different thresholds.

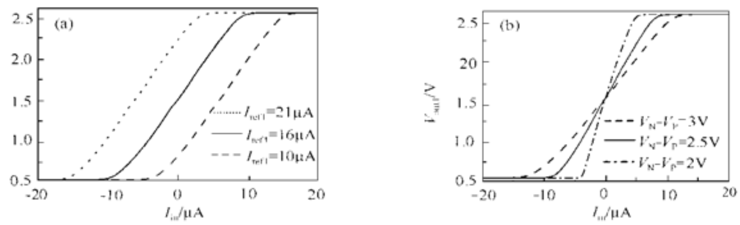

FIG. 4 Programmability of Neuron

(a) Transfer Curve with Different Thresholds, (b) Transfer Curves with Different Gain Factors

The gain factor α can be varied by changing the control voltages V_N and V_P. When both transistors M1 and M2 are working in their respective linear ranges and their sizes are chosen as $\beta_1 = \beta_2$, the relation between I_{AB} and V_{AB} can be written as

$$I_{AB} = I_1 + I_2 = \beta_1 V_{AB}[(V_N - V_{T1}) - (V_P + |V_{T2}|)] \tag{11}$$

so the equivalent linear resistor R_{AB} is written as

$$R_{AB} = \frac{1}{\beta_1[(V_N - V_P) - (V_{T1} + |V_{T2}|)]} \tag{12}$$

Equation (12) shows that the bigger $(V_N - V_P)$ is, the less R_{AB} would be, i. e. the less the slope of V_B versus I_{in} would be. Should the V_{out1} increase more slowly, the gain factor would be smaller. Different activation functions with various gain factors are shown in Fig. 4(b). Note that the saturation levels of the sigmoid remain constant for different gain values, in contrast to most of the implementations reported in literatures[7]. This ensures that for different gain values, the input linear range of the synapse in subsequent layer can be fully used.

4 Experiment Results

The non-linear partition problem is used to verify the operation of the proposed circuit system with 2-1 configuration. Figure 5 illustrates the transient output of the training. Considered that the low output voltage of the neuron is 0.52V, the high output voltage is 2.59V and the middle voltage is 1.56V, the experiment can be described as follows: if the two inputs are both lower than 1.56V or both greater than 1.56V, the output is 2.59V; otherwise, it is 1.56V. The corresponding inputs of linear *A*, *B*, *C* and *D* in Fig. 4 are (1V, 1V), (1V, 2V), (2V, 1V), (2V, 2V) respectively and the corresponding targets are 0.52, 2.59, 2.59 and 0.52V respectively. It can be seen from Fig. 5 that the circuit system converges within 1ms.

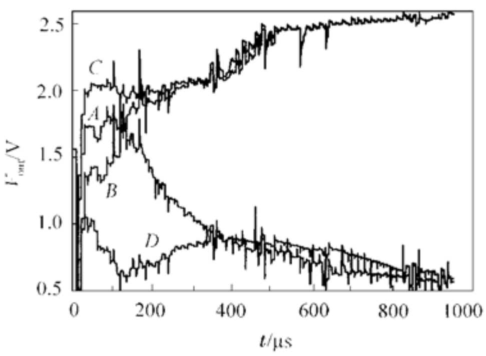

FIG.5 Transient Output Curves of Non-Linear Partition Problem

5 Conclusions

A circuit system of programmable BP on-chip learning neural network has been designed with analog circuits except the weight storage unit. It has the merits of simplicity, speediness, low power-consumption and high density. The whole system comprises a feedforward network, an error back-propagation network and a weight updating circuit. A novel programmable neuron has been proposed, which generates the sigmoidal function and its derivative using the forward differential method. With level 47 transistor models as a standard 1.2μm CMOS process, HSPICE simulations are done to the neuron. The results show that the relative error between the generated neuron activation function and its fitted sigmoid function is less than 3% and that between the derivatives observed from the simulation and the simulated neuron activation function is less than 5%. Moreover, the threshold and the gain factor of the neuron can be easily programmed according to different requirements. The simulation of the non-linear partition problem verifies the superior performance of this BP neural network with on-chip learning.

References

[1] Y. Berg, Analog Integrated Circuits and Signal Processing, 1996, **9**: 65—75.

[2] T. Morie and Y Amemiya, IEEE Journal of Solid State Circuits, 1994, **29**(9): 1086—1093.

[3] G. Cauwenberghs, IEEE Trans. Neural Networks, 1996, **7**(2): 346—361.

[4] M. Valle, Analog Integrated Circuits and Signal Processing, 1996, **9**: 231—245.

[5] B. K. Dolenko and H. C. Card, IEEE Trans. Neural Networks, 1995, **6**(5): 1045—1052.

[6] Chun Lu and Bingxue Shi, Electronics Letters, 2000, **36**(6): 553—555.

[7] S. T. Lee and K. T. Lau, Electronics Letters, 1995, **31**: 1618—1619.

[8] 林谷, 石秉学, 半导体学报, 2000, **21**(1): 69—75[LIN Gu and SHI Bingxue, Chinese Journal of Semiconductors, 2000, **21**(1): 69—75(in Chinese)].

[9] 粟国星, 石秉学, 路伟, 半导体学报, 1988, **19**(7): 532—537[LI Guoxing, SHI Bingxue and LU Wei, Chinese Journal of Semiconductors, 1988, **19**(7): 532—537(in Chinese)].