Proceedings of the V Brazilian Conference on Neural Networks - V Congresso Brasileiro de Redes Neurais
pp. 43-46, April 2-5, 2001 – Rio de Janeiro - RJ - Brazil

A Capacitive Adaptive Memory for Neural Applications

Volnei A. Pedroni, Ph.D.
Departamento de Engenharia Eletrônica – CEFET/PR
pedroni@daeln.cefetpr.br

Abstract

Due to their simplicity and relatively small silicon area, hardware realizations of neural networks (NN) often rely on capacitors for storing the synaptic weights. The weights, however, must be constantly changed (adapted) during the learning process. Circuits capable of incrementing-decrementing these stored (analog) values on-board are generally complex, resulting in large-area MOS implementations. For this reason, learning is often done off-board, being the weights loaded onto the NN later. This paper addresses a capacitive circuit which implements an incrementing-decrementing scheme directly on the storage medium. The circuit makes use of a second (small) capacitor, which transfers charge to or removes charge from the main (storage) capacitor, hence increasing or decreasing its voltage. The variation of the stored voltage is exponential, changing faster in the beginning of the adaptation procedure and slower at the end.

1. Introdution

Even very simple neural networks require a substantial number of synaptic weights to perform a given function. The most frequent way used to store these weights in analog implementations of neural circuits is by means of small, local capacitors [1]-[9]. One of the difficulties, however, is that the weights must be changed (*adapted*) constantly during the *learning* process. Circuits capable of incrementing-decrementing these stored (analog) values on-board are invariably complex, requiring therefore large silicon areas.

This paper addresses a capacitive circuit which implements an increment-decrement scheme directly on the storage medium, requiring for that very little additional silicon space. The circuit makes use of a second (small) capacitor, which transfers charge to or removes charge from the main (storage) capacitor, hence increasing or decreasing its voltage (*synaptic weight*), according with requirements dictated by the neuron during the learning phase. As will be shown, the stored voltage varies exponentially, that is, the size of ΔV (the voltage increment or decrement) is initially bigger and then decreases as further adaptation cycles are applied.

2. Capacitive increment-decrement storage circuit

Fig. 1 illustrates the basic capacitive increment-decrement principle. If capacitors C_1 and C_2 are initially charged with voltages V_1 and V_2, respectively, then after the switch is closed the common voltage will be $V=(C_1V_1+C_2V_2)/(C_1+C_2)$. This implies that voltage V_2, for example, will suffer an in(de)crement equal to $\Delta V=(V_1-V_2)C_1/(C_1+C_2)$. Therefore, by designing the ratio $\lambda \equiv C_1/(C_1+C_2)$ properly, one can make the in(de)crements as small (or as large) as desired.

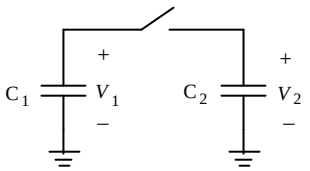


Figure 1: Basic capacitive increment-decrement principle. Upon closing the switch, voltage V_2 will suffer an in(de)crement $\Delta V=\lambda(V_1-V_2)$, where λ is a *design* parameter given by $\lambda \equiv C_1/(C_1+C_2)$.

Fig. 2 summarizes the main circuit (shown later). In it, two fixed voltages, V_H (high) and V_L (low) are used to pre-charge the auxiliary (small) capacitor C'. Initially the neuron defines whether the synaptic weight (represented by voltage V, stored on the main capacitor, C) should be increased (INC switch closed) or decreased (DEC switch closed). Then, with SHARE open, the LOAD switch is pulsed closed, thus pre-charging C' with either voltage V_H or V_L. If now, with LOAD open, the SHARE switch is closed, charge will be redistributed between the two capacitors, causing the voltage V on the main capacitor to increase or decrease, respectively, by the amount

$$\Delta V=\lambda(V_H - V) \quad \text{or} \quad \Delta V=\lambda(V_L - V), \qquad (1)$$

where $\lambda \equiv C'/(C+C')$.

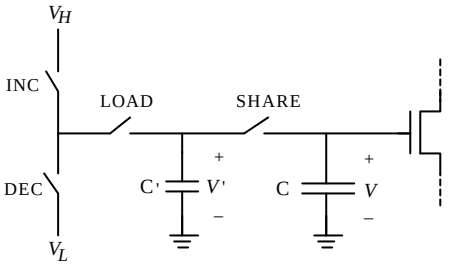


Figure 2: Summary of capacitive inc-dec circuit.

It is easy to verify that, if we start with a voltage V (on C), and apply a series of n successive INC cycles, V will grow exponentially and, after n cycles, its value will be

$$V[n] = V_H - (1-\lambda)^n (V_H - V) \qquad (2)$$

Eq. (2) is illustrated on the left-hand side of Fig. 3, starting with $V=V_L$. Likewise, if we start with V and apply n successive DEC cycles, we obtain

$$V[n] = (1-\lambda)^n V\,, \qquad (3)$$

which is illustrated on the right-hand side of Fig. 3, starting with $V=V_H$.

The complete circuit is shown in Fig. 4. As can be seen, two full CMOS switches, controlled by two global signals, LOAD and SHARE, are used to perform the steps described above, while a local switch, controlled by the DIR (direction) signal, originated by the local neuron, determines which voltage (V_H or V_L) should be used to pre-charge C'. Notice in the inset that the control clocks are non-overlapping.

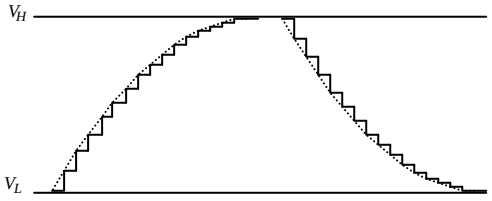


Fig. 3: Exponential behavior of inc-dec circuit (Eq. (2) on the left, Eq. (3) on right).

3. System resolution and convergence

One thing to be noticed is that the auxiliary capacitor, C', takes up very little silicon area. Indeed, the smaller λ the higher the system resolution. However, if the resolution is too high, it might affect the system convergence time. Both aspects are discussed below.

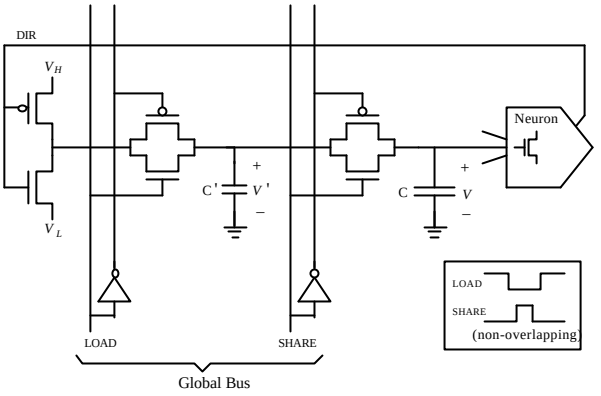


Figure 4: Complete diagram of the capacitive inc-dec circuit. Two full CMOS switches, controlled by the two global signals (common to all neurons) LOAD and SHARE, are used to perform the pre-charge and charge-sharing functions. A locally generated signal, DIR (direction), defines the pre-charge voltage (V_H or V_L). The control signals are depicted in the inset.

3.1 System resolution

We define the system resolution as the ratio between the circuit's dynamic range, $V_H - V_L$, and a single iteration voltage step, ΔV. So it is easy to verify, from Eq. (2) or (3), or in Fig. 3, that the resolution is minimal when V is near either V_H or V_L and must be changed (adapted) toward the other extreme, for in this case ΔV is maximum (using n=1 in Eq. (2), for example, one obtains $\Delta V_{max}=\lambda(V_H - V_L)$). For our purposes, we will consider V at the midpoint of the dynamic range, in which case $\Delta V=\lambda(V_H - V_L)/2$.

If we want the system resolution to be b bits, then

$$\frac{V_H - V_L}{\Delta V} = 2^b \;\rightarrow\; \boldsymbol{l} \equiv \frac{C'}{C + C'} = 2^{-(b-1)} \qquad (4)$$

Eq. (4) serves to determine the design parameter λ. For instance, for an 8-bit resolution, the design of the capacitors must obey C=127C'.

Still regarding the system resolution, it is important to notice that the resolution discussed above corresponds to a *single* iteration. Clearly, if repeated adaptation cycles are applied, basically any voltage value can be obtained on C.

It is also noteworthy that this kind of approach (of storing weights on capacitors), though relatively robust, for disturbance errors during learning are corrected automatically, does not allow very high resolutions, so the use of a large spread of weight values is not possible. For practical reasons, the resolution is seldom above 8 bits.

3.2 System convergence

Another aspect of relevance relates to the convergence time. For instance, the dynamic range of the weights should be smaller than that of the inc-dec circuit, for it would take an infinite amount of time (an infinite number of iterations) if V were to reach V_L or V_H. This is clear in Fig. 3.

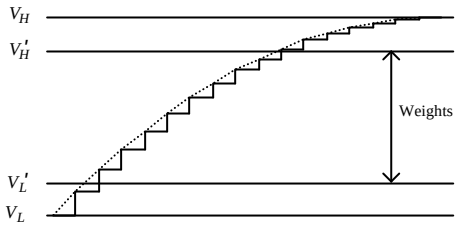


Figure 5: Distinct dynamic ranges used to improve convergence.

Fig. 5 illustrates the use of distinct dynamic ranges, where V_L and V_H are used for the inc-dec circuit, while the weight values are restricted to the range from V'_L to V'_H. In this case, making use of Eq. (2), we verify that the maximum number of iterations (worst case) needed for V to go from the lower to the upper extreme (that is, from V'_L to V'_H) is given by

$$n = \frac{\log \dfrac{V_H - V_H'}{V_H - V_L}}{\log(1-\lambda)} \qquad (5)$$

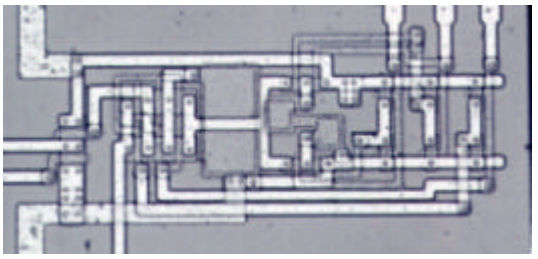
Likewise, using Eq. (3), we verify that moving from V'_H to V'_L requires

$$n = \frac{\log \dfrac{V_L' - V_L}{V_H - V_L}}{\log(1-\lambda)} \qquad (6)$$

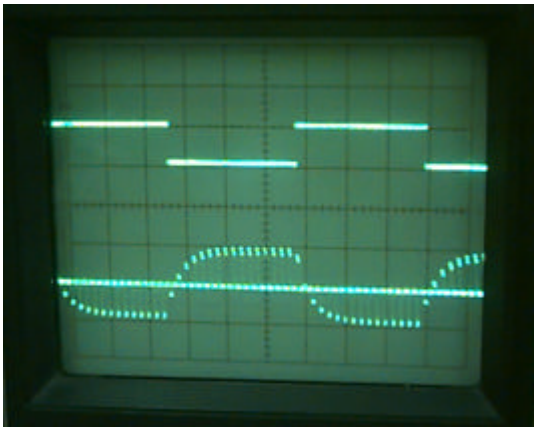
iterations. For example, if the voltages in Fig. 5 are 0V, 1V, 4V, and 5V, and λ=1/32 (b=6 bits), then n=51 iterations are needed to go from one extreme to the other of the neuron's dynamic range.

4. Experimental results

A 16-cell increment-decrement chip was fabricated through MOSIS, using 1.2μm CMOS technology. The transmission-gate transistors were all of size W/L=12/2, and the capacitors were designed using double-poly, being the area of C 800μm^2. For C' two options were employed, one with 40μm^2 ($\lambda \cong 0.05$) and the other with 260μm^2 ($\lambda \cong 0.25$). A microphotograph of an isolated cell appears in Fig. 6(a), while one set of measurements (respective to the option with the larger value of C') is presented in Fig. 6(b). The upper trace of Fig. 6(b) shows the DIR signal, which causes a succession of INC cycles, followed by a succession of DEC cycles. The corresponding voltage (weight) on capacitor C is shown in the lower trace of the scope. The measurements resemble Fig. 3, with the voltage increments-decrements in good agreement with Eqs. (2) and (3).



(a)



(b)

Figure 6: (a) Inc-dec cell microphotograph and (b) experimental results.

5. Conclusion

We have presented a very simple, area-efficient VLSI implementation of an analog storage circuit, which allows the stored values to be changed directly on board. The circuit operates based on the principle of charge-redistribution between two capacitors. The implementation is appropriate for neural networks and other highly parallel analog signal processing systems. A theoretical analysis was presented, along with experimental results obtained from a 1.2μm CMOS chip.

References

[1] M. H. Cohen, A. G. Andreou, “Analog CMOS integration and experimentation with an auto-adaptive independent component analyzer,” IEEE Trans. on Circuits and Systems II, pp. 65-77, Feb. 1995.

[2] J. G. Elias, D. P. M. Northmore, “Switched-capacitor neuromorphs with wide-range variable dynamics,” IEEE Trans. on Neural Networks, pp. 1542-1548, Nov. 1995.

[3] G. Cauwenberghs, “An analog VLSI recurrent neural network learning a continuous-time trajectory,” IEEE Trans. on Neural Networks, pp. 346-361, March 1996.

[4] S. L. Adler, G. Bhanot, J. D. Weckel, “Algorithmic aspects of a neuron coherent wave synapse realizations,” IEEE Trans. on Neural Networks, pp. 1262-1271, Sep. 1996.

[5] L. W. Buchan, ª F. Murray, H. M. Reekie, “Floating gate memories for pulse-stream neural networks,” Electronics Letters, pp. 397-399, Feb. 1997.

[6] A. J. Montalvo, R. S. Gyurcsik, J. J. Paulus, “An analog VLSI neural network with on-chip perturbation learning,” IEEE Journal of Solid State Circuits, pp. 535-543, April 1997.

[7] E. I. El-Masry, H. Yang, M. A. Yakout, “Implementation of artificial neural networks using current-mode pulse width modulation technique,” IEEE Trans. on Neural Networks, pp. 532-547, May 1997.

[8] A. Schmid, Y. Leblebici, D. Mlynek, “Compact charge-based 4 bit flash ADC circuit architecture for ANN applications,” Electronics Letters, pp. 784-786, April 1998.

[9] A. Rantala, S. Franssila, K. Kaski, L. Lampinen, M. Aberg, P. Kuivalainen, “High-precision neuron MOSFET structures,” Electronics Letters, pp. 155-157, Jan. 1999.