Auto-Programmable Impulse Neural Circuits

D. Watula and J. Meador
Department of Electrical Engineering and Computer Science
Washington State University
Pullman, WA 99164-2752

Abstract–

Impulse neural networks use pulse trains to communicate neuron activation levels. Impulse neural circuits emulate natural neurons at a more detailed level than that typically employed by contemporary neural network implementation methods. An impulse neural circuit which realizes short term memory dynamics is presented. The operation of that circuit is then characterized in terms of pulse frequency modulated signals. Both fixed and programmable synapse circuits for realizing long term memory are also described. The implementation of a simple and useful unsupervised learning law is then presented. The implementation of a differential Hebbian learning rule for a specific mean-frequency signal interpretation is shown to have a straightforward implementation using digital combinational logic with a variation of a previously developed programmable synapse circuit. This circuit is expected to be exploited for simple and straightforward implementation of future auto-adaptive neural circuits.

1 Introduction

Natural neurons compute by performing spacial and temporal integration of incoming impulse trains. In this paper, we describe circuits which implement artificial neural networks that operate in a similar fashion–by integrating trains of discrete pulses. Artificial neurons which work in this manner are desirable not only because they more closely mimic natural neuron functionality than other neural-network paradigms, but also because pulse-based communication inherits positive aspects of both digital and analog communication systems. Data can be represented without quantization effects by encoding information in the time period between identical digital pulses. This combination of analog signaling with digital waveforms is robust; even in a noisy environment, reliable detection of the presence of a pulse is simple. These advantages may have contributed to the natural selection of pulse-based communication in natural neural networks.

2 Information Representation and Interpretation

The impulse neural circuits we are working with communicate via the use of pulse frequency modulation (PFM). There are two readily apparent ways of interpreting the information

6.3.2

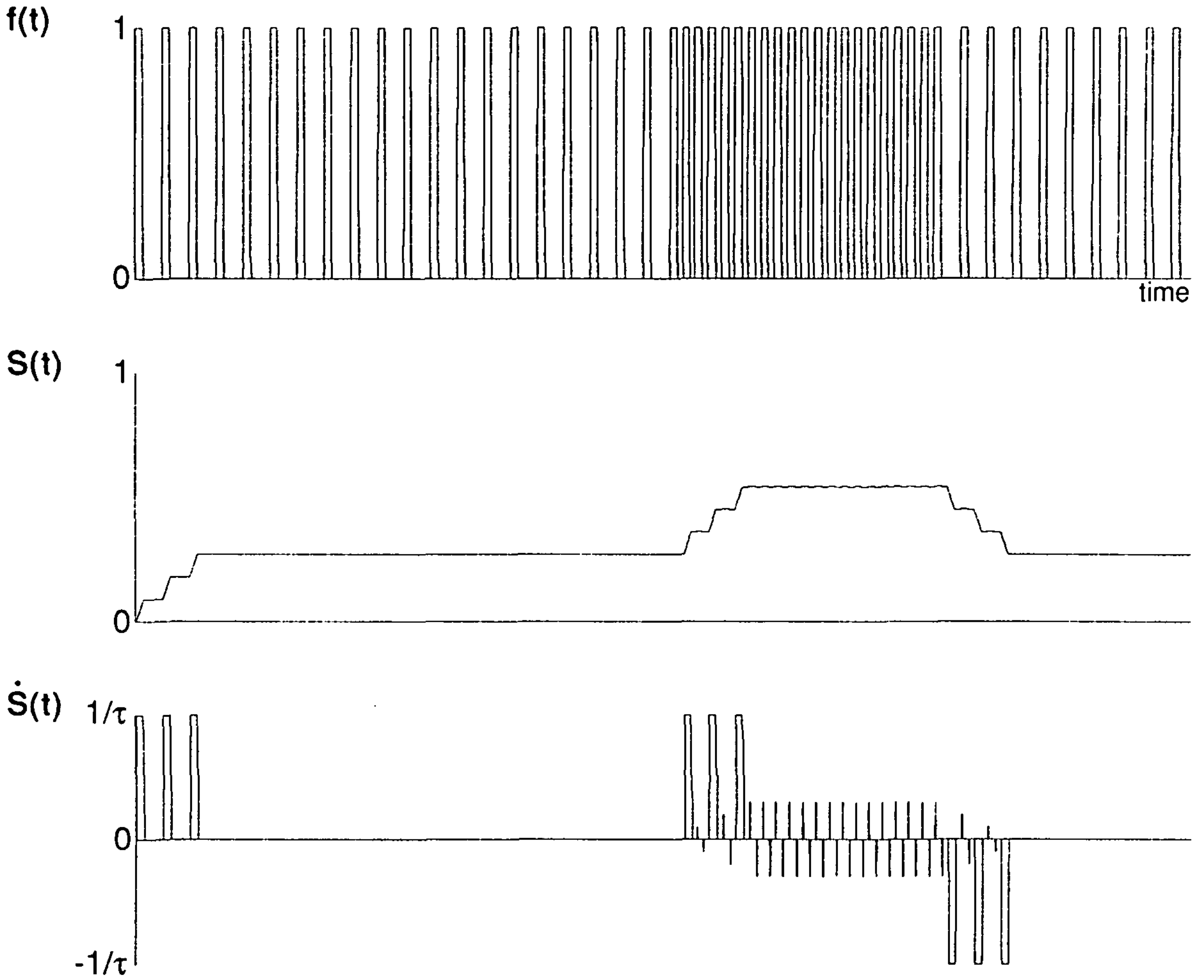

Figure 1: A pulse train f(t), its estimated mean frequency signal S(t), and signal velocity. Waveform f(t), consisting of uniform 10us pulses with 1us rise and fall times, changes from 30kHz to 60kHz and back. The observation period is $\tau = .0001s$. S(t) and its derivative were evaluated numerically

content of a PFM waveform. Information may be perceived as being contained in the time interval between pulses or in the average of these intervals: equivalently, it is contained in either the instantaneous frequency of the pulse train or in its mean frequency. One advantage of the mean frequency interpretation is that it is easily estimated. The signal value is equivalent to the average frequency as estimated by the reciprocal of the observed average firing period, which is in turn proportional to the number of pulses which occur over an observation period.

Investigations into the nature of PFM communication in neurons often emphasize the mean-frequency interpretation. The integral-pulse-frequency-modulated (IPFM) signal model for natural neural systems provides a case in point. The spectrum of an IPFM signal contains a DC-referenced component proportional to the modulating signal that can be easily recovered by low-pass filtering or averaging. This signal model suggests these operations are the most likely methods employed by natural neurons to recover signal information, as they are the simplest and otherwise the only ways in which distortion-free demodulation can be achieved [1].

The integrate-and-fire neuron model used to generate IPFM signals accurately describes the operation of the impulse neural circuits we are investigating. This makes the time-averaged signal interpretation useful for analyzing impulse neural circuits.

In the remainder of this paper we will assume a time-averaged signal interpretation over some arbitrary period τ. Under the assumption of a continuous pulse waveform, the signal $S(t)$ represents the average pulse firing rate:

$$S(t) = \frac{1}{\tau}\int_{t-\tau}^{t} f(\sigma)d\sigma \qquad (1)$$

Since the signal is represented by the time average of the pulse train, it has been noted that the pulse train itself should contain information regarding the time-differential (or velocity) of the signal [Gluc88]. In fact, simple application of a fundamental theorem of calculus yields:

$$\dot{S}(t) = \frac{1}{\tau}(f(t) - f(t-\tau)) \qquad (2)$$

This well-known formulation proves useful in the analysis of the impulse neural circuit to be described as well as in the development of a Hebbian learning rule which is particularly well suited to VLSI circuit implementation. Figure 1 illustrates a PFM coded signal and its derivative. A similar formulation using a low pass filter signal representation also allows for stable numerical computation of the instantaneous mean frequency and its derivative [3]. The application of the low pass signal representation to VLSI implementation is beyond the scope of this paper, but may have future utility.

3 Impulse Neural Circuits

The impulse neural circuits we have been developing incorporate a blend of analog and digital circuit implementation styles [8]. There are two principle subcircuits utilized, namely

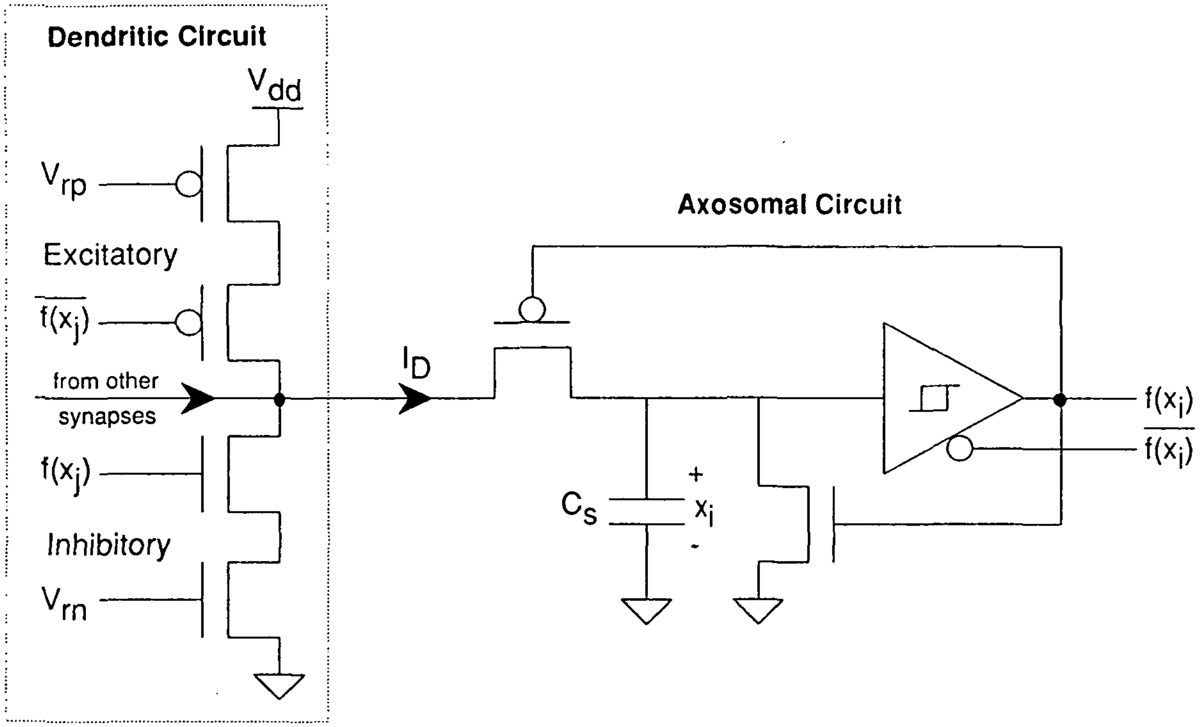

Figure 2: Axoxomal and fixed-synpase circuits

axosomal and synapse circuits. The axosomal circuits compute a nonlinear summation of input signals whose magnitudes are weighted by the synapse circuits. Together they implement an integrate and fire mechanism which corresponds to one of several commonly used models of natural neuron function [6],[2],[1].

3.1 Axosomal Circuit

The axosomal circuit (Figure 2) is a simple current controlled relaxation oscillator. The circuit has two discrete states of operation, the input integration state and action potential generation. The Schmitt trigger establishes two threshold potentials which determine when the neuron moves between these states. Feedback to two FETs, one in series with and one shunting capacitor C_s, determines whether it is being charged by net input excitation during the integration state or being discharged during an action potential. Instantaneous membrane potential is represented by voltage x_i across the capacitor. The upper Schmitt trigger threshold voltage (V_{th}) emulates the natural neuron action-potential threshold while the lower (V_{tl}) emulates the membrane potential at which the voltage-gated ion channels close.

During the integration state, $x_i < V_{th}$ and the series PFET conducts charge to or from C_s at a rate dependent upon the net result of excitatory and inhibitory stimuli (expressed by I_D in Figure 2). Over this period, the shunting NFET conducts no current. When x_i reaches V_{th}, $f(x_i)$ switches from 0 to 1, switching off the series PFET and driving the shunting NFET into the active operating region. This is the action potential state of the circuit. The capacitor discharges toward ground at a rate determined by C_s, V_{th}, and the

NFET characteristics. When $x_i = V_{tl}$, the circuit switches back to the accumulation state and one firing cycle is complete. Circuit operation continues in this manner at a rate which is dependent upon the net excitation from the dendritic current.

3.2 Synapse Circuits

Two dendritic circuits have been developed, a fixed synapse and a programmable synapse. Using a parallel-connected combination of these synaptic cells, networks with any combination of inhibitory and excitatory connections can be constructed. Additionally, these circuits can emulate additive or shunting connections.

Figure 2 depicts the inhibitory and excitatory fixed synapse circuits. In each two-transistor circuit, one FET behaves as a voltage-controlled switch for converting uniform pulses into weighted, discrete charge packets that are conducted to or from an axosomal summing capacitor. A second FET determines the synaptic weight, or size of the charge packet. Depending on the choice of reference voltage (V_{rp} or V_{rn}), the pair acts as either a switched current source in an additive synapse or a switched resistor in a shunting connection. Further control over synaptic efficacy is attained by customizing FET W/L ratios. Thus, fixed networks may be implemented simply by tailoring device geometries and selecting appropriate bias voltages.

Unfortunately, experimental work has shown that it is difficult to select these parameters based on weight values obtained from conventional neural-network training techniques. More traditional paradigms, such as error backpropagation, employ abstract models which do not exhibit the nonlinear response or process variations found in these electronic circuits. One solution to this problem is the application of auto-adaptive circuitry which compensates for circuit imperfections.

The programmable synapse constitutes a portion of that auto-adaptive circuitry. Specifically, it stores the current adaptive state (connection weight) in a non-volatile fashion as the threshold voltage (V_{th}) of a floating-gate FET. The programmable synapse circuit (Figure 3) consists of five transistors, two of which share a common floating gate. In combination with reference FET Q_1, the circuit acts as a switchable current source controlled by reference voltage V_{rn} and the impulse gating signal $f(x_j)$. With V_{rn} adjusted such that Q_4 operates in the subthreshold region, a small shift of V_{th} induces a significant change in the Q_4 source current. With Q_1 properly biased, this current controls the amount of current flowing into the dendritic summing node (I_D). With V_{rn} fixed, variations in V_{th} for Q_4 will vary I_D over several orders of magnitude when Q_3 conducts.

The circuit is programmed by the application of voltage pulses V_{pp} to the control gate of Q_5, causing a high electric field to form in the dielectric between the gate and the substrate. This will induce charge tunneling between the substrate and the floating gate, where the carriers will become trapped. Any trapped carriers will redistribute themselves between Q_4 and Q_5 since they share the conductive polysilicon floating gate. As a result, the V_{th} of both transistors will be shifted slightly. A positive V_{pp} causes negatively charged electrons to become trapped in the floating gate, which in turn causes I_{ds} to drop below I_{ref}. There will then be an increase in the current flowing into the summing node. The

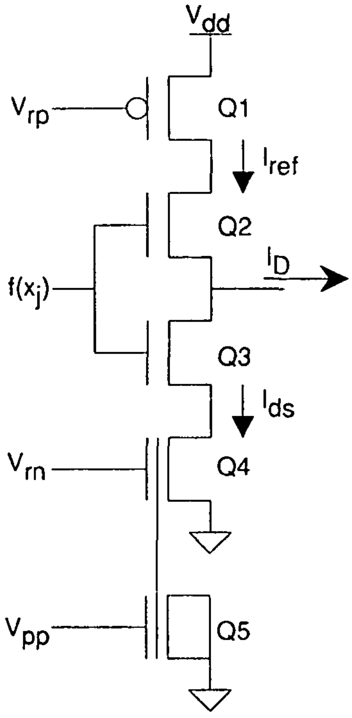

Figure 3: Programmable synapse circuit

opposite happens for a negative V_{pp}. Note that, unlike the fixed synaptic cells, a single programmable synapse may change from excitatory to inhibitory or vice versa.

3.3 Signal transfer function

The transfer function computed by an impulse neural circuit can be derived under the assumption that the chief information carrying component of the impulse train is its average value. The impulse neural circuit which we have described can be shown to use time-averaging to demodulate input signals. It can also be shown that the average output signal is a nonlinear function of a weighted summation of averaged input signals.

The averaged output of an impulse circuit which integrates input signals during both the integration and action potential state is given by:

$$S_o(t) = \frac{1}{\tau}\int_{t-\tau}^{t} f(\sigma)d\sigma = \frac{AT_0}{C(V_{th}-V_{tl})}\sum_{j=1}^{N_I}\beta_j\frac{1}{\tau}\int_{t-\tau}^{t} h(f_j)d\sigma = \frac{AT_0}{C(V_{th}-V_{tl})}\sum_{j=1}^{N_I}\beta_j S_j \qquad (3)$$

for observation interval τ. A, T_0 are the output pulse amplitude and width, C is the integration capacitance, h is the nonlinear transconductance relation associated with synapse j, β_j is the transconductance factor of synapse j, and V_{th}, V_{tl} are as defined previously. N_I is the total number of inputs.

This general result shows that the output mean firing rate is a linear weighted summation of input mean firing rates for a circuit which integrates inputs over the entire observation period *tau*, regardless of the circuit state. Such is not the case, however, in the actual neuron circuit. Input signals are not integrated during the action potential state. This means that the circuit output will eventually saturate with increasing net excitation. A nonlinear activation function can be derived for the circuit using a special case of (3) where the observation interval is $\tau = T_0 + T_1$, with T_1 representing the integration period for a single output pulse:

$$S_o(t) = \frac{AT_0}{C(V_{th} - V_{tl})} \sum_{j=1}^{N_I} \beta_j \frac{1}{T_0 + T_1} \left(\int_{t-T_0-T_1}^{t-T_1} h(f_j) d\sigma + \int_{t-T_1}^{t} h(f_j) d\sigma \right) \tag{4}$$

Since no input signals register an effect during the action potential state, the first integral term evaluates to zero. Algebraic manipulation of the resulting equation yields the following expression for T_1:

$$T_1 = \frac{C(V_{th} - V_{tl})}{NET} \tag{5}$$

where

$$NET = \textstyle\sum_{j=1}^{N_I} \beta_j \hat{S}_j$$

where $\hat{S}_j$ represents an approximation of the input signal over the (shorter) observation interval T_1 and NET is the net input excitation (weighted sum of input signal approximations). The actual input/output relationship computed by the circuit becomes:

$$S_0 = \frac{1}{T_0 + T_1} = \frac{1}{T_0 + \frac{C(V_{th} - V_{tl})}{NET}} \tag{6}$$

Here, S_0 expresses the instantaneous firing rate of the neuron output signal as a nonlinear weighted sum of mean firing rates observed over the integration period. This function is sigmoidal, saturating as NET approaches infinity and dropping to zero with $NET = 0$. This result agrees precisely with that obtained via the more specific analysis reported in [8], and concurs with circuit simulation results.

A series of SPICE simulations (Figure 4) of an axosomal circuit extracted from a 2μ CMOS layout shows very close agreement with the form of (6) over at least six orders of magnitude of input current. Preliminary measurements from a prototype of the axosomal circuit are also presented (Figure 5). Although the data exhibits the same exponential slope and saturation characteristic as the simulation, it also manifests an awkward transition into saturation that has not yet been explained. The large offset in the data before the saturation region is attributable to the input capacitance of the chip, which is significant since the extracted value of $Csubs$ is less than $.2pF$.

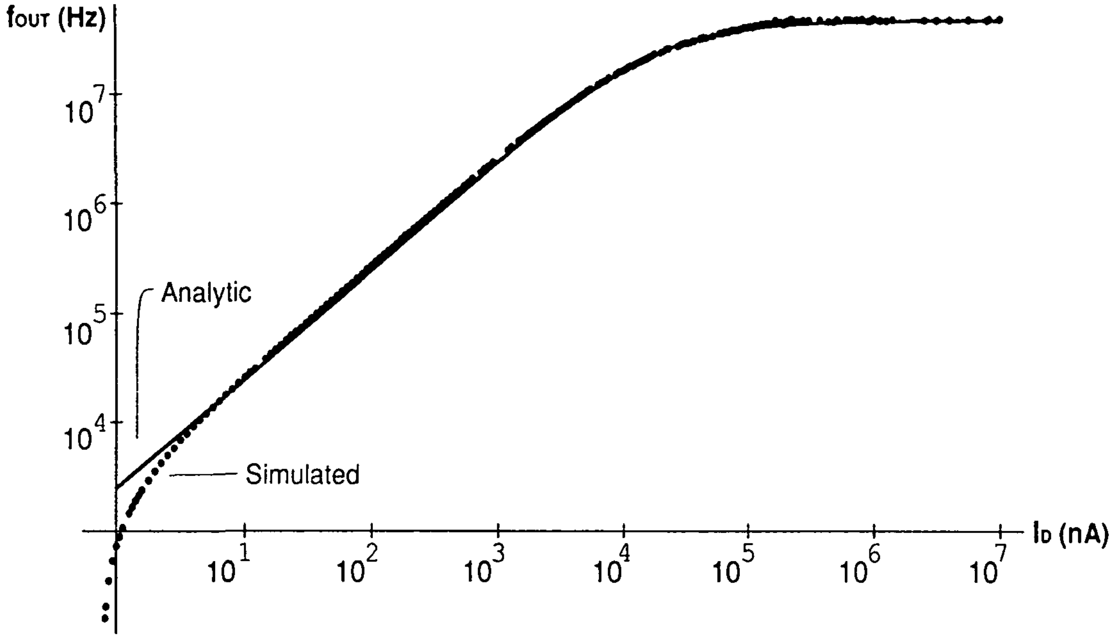

Figure 4: Comparison of simulated transfer characteristic with Equation 6 for $C = .18pF, T_0 = 2.2 \times 10^{-8}s$, and $V_{th} - V_{tl} = 2.2v$

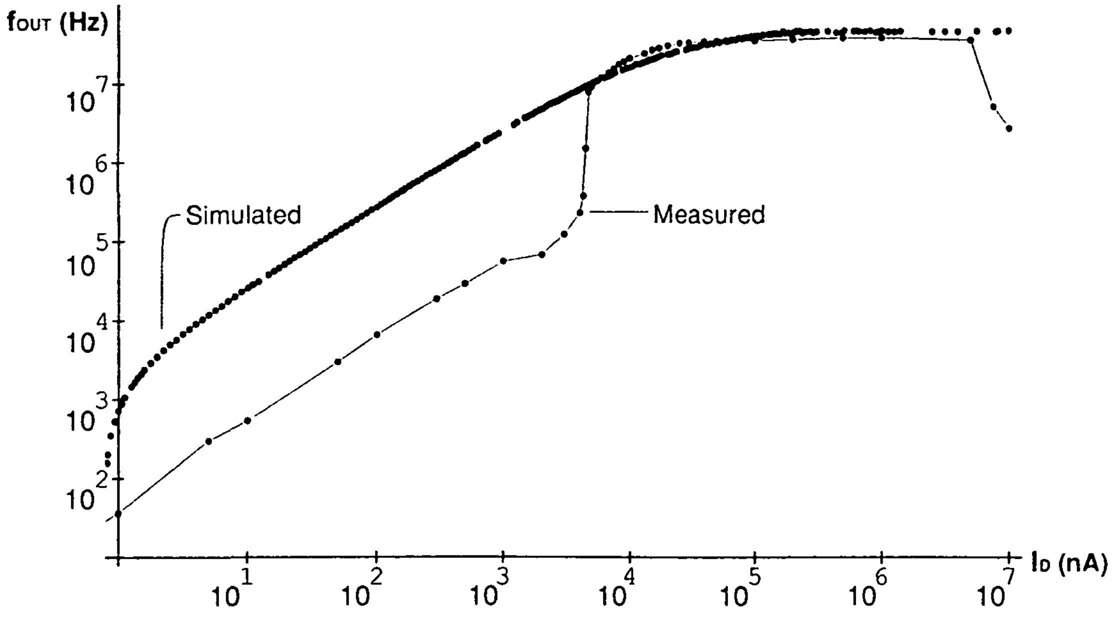

Figure 5: Comparison of simulated transfer characteristic with measurements from a prototype neuron

4 Auto-Adaptation in an Impulse Neural Circuit

There exist three general neural network adaptation strategies. Neural networks can be trained using supervised, unsupervised, or reinforcement methods. Any algorithms which update connection weights as a function of some error between an output and target vector are supervised. Error backpropagation is one of the best known examples of a supervised learning algorithm. Supervised adaptation not only updates the connections when errors occur, but also gives some indication about how to update the connections to reduce error. Unsupervised training methods use no error feedback whatsoever to adjust connection weights. Their chief advantage is the exclusive use of local information from pre- and post-synaptic neurons. The reliance on only local information helps simplify VLSI implementation of the learning rule. Reinforcement methods utilize output feedback to update connection weights, but without specific error gradient information. Reinforcement learning can use either global or local information, and provides a good model for classical conditioning phenomena which have been observed in natural neurons.

Natural neural systems adapt to the environment while processing trains of pulses in the apparent absence of direct performance feedback, so unsupervised learning provides one model for adaptation in such systems. It is our goal to implement auto-adaptive VLSI neural circuits, so a natural alternative is to begin with algorithms that use locally available information to update connection strengths. Unsupervised learning is a logical candidate for this.

At least four types of unsupervised learning have been described, including signal Hebbian, differential Hebbian, competitive, and differential competitive learning. This algorithm class exhibits a simplicity and locality not present in more complex training algorithms such as backpropagation and ART. Our work thus far has focused upon Hebbian rules exclusively, even though extensions based on competitive learning may also be possible. Hebbian style adaptation rules are found in adaptive bidirectional associative memories (ABAMs) [Kosk89], and in reinforcement generation for classical conditioning models. Hebbian rules have also been used in principle component analysis for feature extraction in lieu of computationally complex matrix algebra [9],[7],[5].

Signal Hebbian adaptation is expressed as a correlation of pre-synaptic and post-synaptic neuron firing rates:

$$\dot{w}_{ij} = \beta S_i(t) S_j(t) \tag{7}$$

where $S_i(t)$ represents a mean firing rate of neuron i at time t and β is an adaptation rate constant. It is important to note that this expression is independent of the type of frequency averaging used, so its meaning can vary with the signal interpretation. Even though the signal Hebbian rule can account for learning, it cannot account for "unlearning", since connection strengths monotonically increase with signal correlations.

The differential Hebbian adaptation rule can be expressed as the correlation between the rate-of-change (velocity) of pre- and post-synaptic signals:

$$\dot{w}_{ij} = \beta \dot{S}_i \dot{S}_j \tag{8}$$

When used for reinforcement generation, differential Hebbian has been shown to account for several aspects of classical conditioning behavior which go beyond the simple acquisition accounted for by signal Hebbian learning.

A differential Hebbian learning rule for the mean frequency signal representation expressed in (1) can be implemented using simple digital logic in conjunction with a variation of the adaptive synapse circuit of Figure 3. Given the relation described in (2) it can be shown that the product of two signal velocities can be computed using the relation:

$$\tau^2 \dot{S}_1(t)\dot{S}_2(t) = f_1(t)f_2(t) - f_1(t)f_2(t-\tau) - f_2(t)f_1(t-\tau) + f_1(t-\tau)f_2(t-\tau) \tag{9}$$

Since all $f_i(t)$ are binary signals having the value "1" when an action potential is occurring at time t, much of the computation can be carried out by simple combinational logic. This representation is naturally compatible with impulse neural circuits, as the binary pulse train signal representation allows for the reduction of the multiply operations to simple logical AND functions.

Using (1) to establish a signal interpretation based upon inverse mean period allows for the design of a simple auto-adaptive synapse circuit. Figure 6 diagrams an implementation of an auto-adaptive differential Hebbian synapse cell which is currently being developed. The binary products are computed by the AND functions while their arithmetic combination is computed by charge summation in the floating gate circuit. Delayed versions of the input signals are derived from simple binary delay cells. Only one delay cell is required per neuron circuit. The configuration shown will exhibit some features of classical conditioning if f_1 is the output from the post-synaptic neuron and f_2 is any input signal. As such, the circuit computes reinforcement signals which contribute to both connection acquisition and extinction.

5 Conclusion

We have shown that a differential Hebbian adaptation rule can be implemented in a simple and straightforward manner in an impulse neural circuit. The results of this work are expected to contribute to the implementation of classically conditioned neural circuits and adaptive bidirectional associative memories as well as any other network training algorithm based upon a differential Hebbian learning rule.

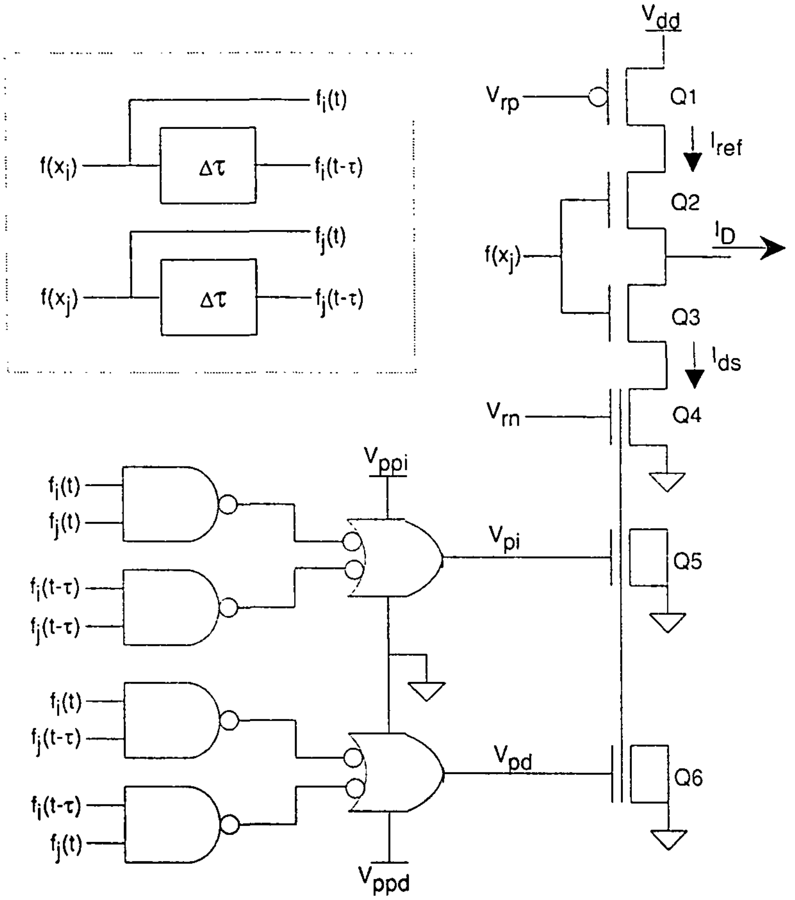

Figure 6: An autoadaptive differential Hebbian synapse cell representing the weight from neuron j to neuron i

References

[1] Bayly, E., "Spectral Analysis of Pulse Frequency Modulation in the Nervous Systems," IEEE Trans. on Bio-Medical Engineering, Vol. BME-15, No. 4, pp. 257-265, 1968.

[2] Gestri, G., "Pulse frequency modulation in neural systems: a random model," Biophys J., Vol. 11, pp. 98-109, 1971.

[3] Gluck,M.A., D.B. Parker, E. Reifsnider, "Some Biological Implications of a Differential Hebbian Learning Rule," Psychobiology, Vol. 6 No. 3, pp. 298-302, 1988.

[4] Kosko, B., "Unsupervised Learning in Noise," Proc. IJCNN, Washington D.C., pp. I-7,I-17, 1989.

[5] Kung, S.Y., "Orthogonal Learning Network for Constrained Principal Component Problem," Proc. IJCNN, San Diego CA., 1990.

[6] Hoppensteadt, F.C., An Introduction to the Mathematics of Neurons, Cambridge University Press, Cambridge, 1986.

[7] Leen, T., M. Rudnick, and D. Hammerstrom, "Hebbian Feature Discovery Improves Classifier Efficiency," Proc. IJCNN, San Diego CA., 1990.

[8] Meador, J, et al., "Programmable Impulse Neural Circuits," IEEE Trans. on Neural Networks (to appear).

[9] Oja, E., "A simplified neuron model as a principal component analyzer," J. Math. Biology, Vol. 15, pp 267-273, 1982.