A.1 Background

Nonvolatile semiconductor memory devices have been widely used in digital applications [93]. Also they have shown promise for diverse analog applications including analog adaptive filters [94], trimming of analog circuits [95, 96], and neural networks [36, 42, 69]. In particular, the present poor state of modeling of submicron devices for analog applications has prompted the idea of using on-chip trimming for compensation of unwanted or unpredictable effects in such submicron designs. In some neural-network implementations, researchers have begun to take advantage of the long-term storage capabilities of non-volatile semiconductor memories. In particular, electrically-erasable programmable read-only memories (EEPROMs) have been used for their on-board writable-readable properties. Such EEPROM devices are of two kinds, charge-trapping and floating-gate devices [97]. Here we will briefly review each type.

A.2 Device Review

A.2.1 Charge-Trapping Devices

In charge-trapping devices, the information is stored in the form of electrons trapped at the interface of particular layers in a multi-layer gate structure. Amongst such devices, one can identify the metal-nitride-oxide-semiconductor (MNOS) device [98], a simple drawing of which is shown in Fig. A.1. In this device, the gate is formed of a sandwich of nitride ($Si_3N_4$) between oxide ($SiO_2$) and metal ($Al$).
By applying a positive high voltage to the metal layer of the gate, electrons move from substrate to the oxide-nitride interface, causing the threshold voltage of the device to increase. Conversely, application of a negative high voltage to the metal layer of the gate, removes electrons from the traps and returns them back to the substrate, thereby reducing the threshold voltage of the device. This shift of the threshold voltage can be sensed by other circuitry to detect the state of the stored information.

A.2.2 Floating-Gate Devices

In these devices, controllable charge is stored on a conducting or semi-conducting layer surrounded by insulators. Normally these devices have two gates arranged vertically. The outer gate is connected to the external circuitry. However, the inner gate is not electrically connected to outside, but is isolated between insulator layers, which are normally silicon oxide [99]. The existence of a floating layer surrounded by good insulators gives good long-term charge retention to these devices.

Charging and discharging of the floating gate is performed by a physical phenomenon called Fowler-Nordheim tunneling [100, 101, 102]. For a normal
polysilicon/oxide/silicon MOS structure, there is a 3.2eV energy barrier that prevents electrons from moving between silicon and polysilicon through the oxide layer. At a given temperature, electrons have a certain probability of tunneling a certain distance into the oxide layer. For example, at room temperature, the tunneling distance is about 5nm. The Fowler-Nordheim phenomenon predicts that if the electric field within the high-tunneling-probability distance of the oxide layer is strong enough (that is greater than 3.2V in 5nm, about $6.4 \times 10^6 V/cm$), then the electrons that tunnel into $SiO_2$ can surpass the energy barrier and enter the conduction band of oxide. In the conduction band of oxide, they are quite mobile and can move with the field toward the more positive electrode (which is either silicon or poly, depending on which one is connected to the higher potential).

Various techniques used to create a significantly high field within the high tunneling-probability distance have led to several different EEPROM structures. Fig. A.2 illustrates one of these floating-gate devices with a thin oxide layer (of thickness

![Fig. A-2: A simple representation of a floating-gate device with a thin-oxide layer on its drain area.](image-url)
around 10 nm) over its drain area [103]. In this structure, if a positive high-voltage pulse is applied to the outer gate, while the drain, source and substrate are grounded, electrons tunnel from the drain through the thin oxide onto the floating gate. As a result, the threshold voltage as seen from the outer gate will shift to a higher value. If a positive high-voltage pulse is applied to the drain, while the gate and substrate are grounded, electrons will be removed from the floating gate and the threshold voltage will shift to a lower value. Although this device works well enough for use in a digital circuit, the resulting structure shows strong dependence between drain current and voltage [52] which is not appropriate for some analog applications such as an adjustable current source. As well, fabrication of the thin-oxide requires an extra processing step and increases the chance of device failure. Also the very thin oxide layer is found to degrade the charge-retention performance.

To solve the above-mentioned problems, textured-poly floating-gate devices have been fabricated. In these devices, tunneling occurs between poly layers separated by a thicker oxide layer (around 50 to 80 nm). As well, many small bump-like areas on the surface of the poly layers help to build up fields 4 to 5 times larger than the average applied field [104]. However, these devices require special processing to create the textured-surface poly silicon.

Several other charge-injection structures have been proposed and tested using conventional CMOS technologies [95, 96]. One of these involves injectors which employ cornered-poly diffusion tunneling, where the local field is increased by appropriately shaping of the geometry of the electron-emitting surface. This is due to the fact that corners introduced by lithographic features in the polysilicon layer can enhance the electric field by a factor of 2 to 4 [95]. In Northern Telecom standard 1.2μm CMOS technology (CMOS4S), several tunneling injectors of this type, have been fabricated and tested [96].

In [105, 106, 107], the evidence of tunneling between polysilicon layers in ordinary CMOS processes has been reported. Fig. A.3 shows one typical poly1-poly2 charge injector. This charge injector uses the field enhancement produced by sharp edges associated with oxide-separated poly-1-poly-2 layers. The results in [107] show that while for tunneling between diffusion and poly, pulses with amplitude around 25V are required, this structure reaches the same level of charge injection with pulses which are 10V lower in amplitude.

Because they do not involve drain-region shaping, these devices do not suffer from a high dependency between their drain current and voltage. Especially, the interpoly-injection design looks appealing for our application. Moreover, their charge-retention capabilities are also better than their thin-oxide counterparts. However, all of these devices continue to suffer from cycling problems: after many programming cycles, data retention is affected by electron trapping at defects in the oxide layer and at the silicon-oxide interface. The trapped electrons raise the threshold voltage permanently and reduce the programming range. Ultimately, this makes the
Fig. A-3: (a) An inter-poly charge injector can be implemented using a standard double-poly CMOS process. (b) Cross-sectional view of the charge injector.

device useless. However, this situation is very likely to improve, since the solution for this problem is the growth of high-quality silicon oxide.
A.3 Conclusion

For the reasons described above and because of the essential role of reprogrammability in neural-network applications, we have had to look for another alternative in our present implementation, one capable of an almost unlimited number of programming cycles, as discussed in Chapter 5. However, we believe strongly that in the near future using increasingly-available high-quality-oxide technologies, the synaptic computational and weight-storage functions can be merged into a single floating-gate SyMOS device. Moreover, the availability of such a floating-gate device also serves to simplify dramatically the polarity-control function as identified in Figure 8-1.
Appendix

Scaling Effects

B.1 The Effect of the Scaling of CMOS Technology on SANNs

As CMOS technology advances in the direction of higher integration levels, the dimensions of the devices are shrinking dramatically. One of the most important effects of scaling appears in the operating equation of a MOS device. In particular, it has been shown that short-channel devices tend to provide a current that is linearly proportional to the difference of the gate-source and threshold voltages, rather than following the usual quadratic relation [151]. In view of the possibility of the construction of very-high-density neural systems at smaller and smaller feature sizes, we have considered the system-level implications of linearly-operating devices on the pattern-classification property of SANNs. To this end, we have repeated some of our simulations with devices with an operating equation of the form

$$I_D = \begin{cases} 
K(V_{GS}-V_{th})^n & \text{if } (V_{GS}>V_{th}) \\
0 & \text{otherwise}
\end{cases} \quad (2-1)$$

where $1 \leq n \leq 2$.

Figure B-1 compares the decision boundaries constructed using quadratic ($n = 2$) and linear ($n = 1$) devices. As can be seen, the general form of the decision boundaries is preserved. While for quadratic devices, part of the boundary is formed by a quadratic relation and other parts with linear boundaries, for linear devices,
Fig. B-1: The effect of linear operation of MOS devices in an SANN. (a) Synapses have positive signs and the radius term is negative. (b) and (c) One of the synaptic terms is negative while the bias term is positive (continued on subsequent pages). $W_1=W_2=2$ and $R=1$.

different linear segments continuously joined together form the discriminating boundary. Moreover, as Figures B-1-(b) and (c) verify, switch control of the sign of synapses allows discriminating boundaries to be formed with various pieces of line segments.

Our conclusion is that quadratic behavior of the SyMOS devices is of less importance than the continuous-boundary property they provide, and that SANNs remain equally useful when scaling leads to linear operation.
Fig. B-1. Continued. The effect of linear operation of MOS devices in an SANN. (a) Synapses have positive signs and the radius term is negative. (b) and (c) One of the synaptic terms is negative while the bias term is positive (continued on subsequent page). W1=W2=2 and R=1.
Fig. B-1. Continued. The effect of linear operation of MOS devices in an SANN. (a) Synapses have positive signs and the radius term is negative. (b) and (c) One of the synaptic terms is negative while the bias term is positive. $W_1=W_2=2$ and $R=1$. 

(-1,1,1)
In this work, we have introduced several system-level modifications to the design process of ANNs. These are intended to facilitate hardware implementation through the use of the intrinsic characteristic of a saturated MOS transistor in the synaptic operation. Also, we have shown that in corresponding networks, three minimum-size-floating-gate transistors can implement each synaptic block used in the definition of a fully-functional neuron. Here, we want to introduce and quantify some of the practical advantages of this approach. In our evaluation, we use three basic criteria, namely speed, power, and chip area.

C.1 Speed

The saturation region is where a MOS transistor can provide the highest driving current for a given gate voltage. Correspondingly, a saturated device will function at a higher speed in comparison to one operating in the triode or sub-threshold region. This property suggests that when a neural-network processing engine is expected to handle a huge amount of serial time-varying input data, the saturation-mode approach will demonstrate the best speed performance. A good example of the need for high-speed real-time operation is in hand-written-character-recognition applications, such as for postal-code recognition, where the requirement is to increase the processing rate and the throughput of associated networks to the highest possible value. In such applications, saturated devices operating in parallel networks are logically expected to provide the fastest possible solution.

We note that here, as in other similar situations, there is a trade-off between speed and power consumption, or in other words, between the number of processing...
elements operating in parallel and the maximum throughput, or processing rate. For example, if we consider the charging time of a capacitive load, and assume that a transistor with current $I_1$ will charge it in $t_1$ seconds, then a device with the current $10 \times I_1$ will charge it in $t_1/10$ seconds, but with a power consumption which is 10 times greater for the same supply voltage. Hence, with a constant power budget, a designer must decide whether to obtain a faster and smaller network with a high throughput, or a larger but slower one. Note, in another words, that we can dissipate our power budget in time by employing fewer faster devices, or in space by spreading a network over a larger chip area having more elements, each operating at a lower power level.

C.2 Power Consumption

As noted above, SyMOS networks can be used in two distinct types of environments:

i) Applications with low-rate inputs that require parallel interaction of input data with a large amount of stored data at a relatively low frequency of operation (for example, for image-processing hardware, the input rate is expected to be in range of kilo-Hertz or lower, but the number of pixels can be extremely large). In such a case, each synapse should be designed to consume the lowest-possible power. For example, for devices that consume at most $5 \mu W$ of power, an integration level of up to 200,000 synapses per Watt can be obtained. Thus, this technique applies in general to networks with up to 2,000,000 synapses in a chip dissipating $12-15 W$ maximum power.

ii) Applications with high-rate inputs that require fast synaptic units to provide a higher throughput. In this case, each synaptic unit may have a power dissipation ranging from $50 \mu W$ to $200 \mu W$ (in different designs) giving an integration-level of 20,000 to 5,000 synapses per Watt. In general, chips with 200,000 to 50,000 synapses dissipating up to $15 W$ are possible. Further, we project that operating speeds up to $50 MHz$ are obtainable with such devices. In that case, the computational power is projected to be in range of 250 billion connections per second per Watt.

In order to compare these estimates with commercially-available products, let us consider one of Intel’s neural-network chips: ETANN is an analog-neural-network chip that uses an advanced floating-gate MOS technology. It has 10,240 synapses and 64 neurons and consumes $1.5 W$ power, which on average gives 6827 synapses per Watt. The processing time for this chip is $3 \mu s$, which translates to a $334 kHz$ input rate. The computational power of this chip is announced to be 2 billion connections per second.
C.2.1 Detailed Calculation of Power Dissipation

As we have noted, the average power dissipation of our networks depends on the statistics of the distribution of input data. This is a result of the fact that in pattern-classification applications, the location of the input pattern in the input space can be distinguished while some of the discriminating transistors are in their off state, thereby reducing the average power dissipation. However, when we are designing the chip, the width of power-supply lines and cooling considerations must be based on the worst case with the highest possible power dissipation, both locally and totally. Such detailed calculations for our current chips, which are designed on the assumption of \(50\mu W\) average power dissipation per synapse (for operation at a moderate speed range), are shown below:

(i) Power consumption in each synapse:

(i-1) In the SyMOS device:

\[
P = I_D \cdot V_{DS}.
\]

\(I_D \sim 17\mu A\), and \(V_{DS} \leq 3V\), therefore

\[
P \sim 50\mu W \Rightarrow \sim 20,000 \text{ synapses /Watt}.
\]

(i-2) For refreshing of the \(1\mu F\) analog-storage capacitor at a 100 Hz refresh rate and after less than 1% decay using a 5V supply:

\[
P = f \cdot C \cdot (\Delta V)^2 = 10^2 \times 1 \times 10^{-12} \times (0.01V_{DD})^2,
\]

\[
= 25 \times 10^{-14} W = 2.5fW.
\]

(i-3) Power consumption at the input capacitance of the SyMOS, assumed to be equivalent to a \(50fF\) capacitance operating with a 4V swing at 20MHz:

\[
P = f_I \cdot C \cdot V_{Ss}^2 = 20 \times 10^6 \times 50 \times 10^{-15} \times 16 = 16\mu W
\]

(i-4) Power consumption in the sign-of-synapse switch, assuming a 0.1 V drop at the \(17\mu A\) bias current:
\[ P = V_{\text{drop}} \cdot I_{\text{through}} = 0.1 \text{V} \times 17\mu\text{A} = 1.7\mu\text{W} \]

(i-5) Static power dissipation in the SRAM cell:

\[ P_{\text{SRAM, static}} = 0. \]

(i-6) Total average power dissipation in each synapse is:

\[ P_{\text{synapse}} = 50 + 16 + 1.7 = 67.7\mu\text{W} \]

(ii) Power consumption in each neuron:

(ii-1) Current mirrors:

PMOS-mirrors:

\[ P = 4 \cdot V_{DS} \cdot I_{DS} = 4 \times 2\text{V} \times 100\mu\text{A} = 800\mu\text{W}, \]

NMOS-mirrors:

\[ P_N = 2 \cdot V_{DS} \cdot I_{DS} = 2 \times 3\text{V} \times 100\mu\text{A} = 600\mu\text{W} \]

Total:

\[ P_{\text{mirrors}} = 800\mu\text{W} + 600\mu\text{W} = 1400\mu\text{W} = 1.4\text{mW}. \]

(ii-2) Sigmoid unit: Let us assume the output unit is a voltage source driving a load combination of \( R_{\text{equivalent}} = 20\text{k}\Omega \) (this is an average resistance substituted to approximate the whole range) and \( C = 0.5\text{pF} \) operating at \( 20\text{MHz} \) switching frequency between 1V and 4V voltage levels (\( V_{\text{difference}} = 3\text{V} \)).

\[ P_R = \frac{1}{2 \times 10^4} + \frac{16}{2 \times 10^4} = 425\mu\text{W}, \]

170
\[ P_C = 20 \times 10^6 \times 0.5 \times 10^{-12} \times 9 = 90 \mu W \ , \]

\[ P_{\text{sigmoid}} = 425 + 90 = 515 \mu W. \]

(ii-3) The total power consumption of the neural-signal-collecting part will be:

\[ P_{\text{neuron}} = 515 \mu W + 1400 \mu W = 1915 \mu W = 1.92 mW. \]

(iii) Power consumption in the addressing backbone:

According to HSPICE simulations, the power dissipation for addressing will be less than 20mW for a 20MHz clock speed.

\[ P_{\text{addressing}} = 20 mW. \]

(iv) The total power consumption of a chip with \( S \) synapses and \( N \) neurons operating at 20MHz from a 5V supply then will become:

\[ P_{\text{total}} = S \cdot P_{\text{synapse}} + N \cdot P_{\text{neuron}} + P_{\text{addressing}}, \]

where \( S \) is the number of synapses, and \( N \) is the number of neurons on the chip. If \( S = 10,000 \) and \( N = 200 \), then:

\[ P_{\text{total}} = 10,000 \times 67.7 \mu W + 200 \times 1915 \mu W + 20 mW = 1.080 W. \]

The estimated computational power of such a chip with 10,000 synapses and 200 neurons is 200 billion connections per second.

C.3 Area

As discussed in Chapter 6, in our current implementation, a synaptic unit including all of the related addressing circuitry occupies a 80\( \mu m \times 80\mu m \) area. Almost half of which is used by analog-storage and input-coupling capacitors.
Another 20% is used for a 1-bit SRAM cell. However, in a floating-gate implementation, the core circuitry of a synapse can be reduced to one floating-gate SyMOS (there being no need for threshold-controlling or analog-storage capacitors) and two floating-gate switches, leading to an estimated $10\mu m \times 10\mu m$ area in a 1.2\mu m technology. This corresponds to integration of up to 10,000 synapses per square millimeter, or a million in a square centimeter.

We note that the area of a synapse in the floating-gate design of Intel is $2009 (\mu m)^2$, in a 1\mu m EEPROM technology [15]. The minimum area of a synaptic unit in the reported fabricated chips [15] is 560 (\mu m)^2, which is quite larger than what can be obtained by following our proposed approach.

Even now, if we consider the implementation of a 10,000-synapse 200-neuron chip using the available 1.2\mu m CMOS4S technology, and employing our currently-available building blocks, the required area is $68.4 (mm)^2$, where $2 \times 600,000 (\mu m)^2$ area is for the addressing backbone. This is quite manageable with the currently-available technologies. As noted, its performance would be expected to be in the range of 200 billion connections/s/W.

C.4 Overall Performance

We conclude that the approach presented in this book, by merit of its employment of the intrinsic characteristic equation of a single transistor as the synaptic operation, can dramatically reduce the area of a synaptic unit while providing a range of possibilities for an appropriate balance between speed and power dissipation.
References


[40] E. A. Vittoz, "Analog VLSI signal processing: why, where, and how?"  
Analog Integrated Circuits and Signal Processing 6, pp. 27-44, Kluwer  

network processor with programmable topology," IEEE J. Solid-State  

1989.

implemented using collective computation," Chapel Hill Conf. on VLSI, pp.  

[44] H. P. Graf, L. D. Jackel, R. E. Howard, B. Straughn, J. S. Denker, W. Hubbard,  
D. M. Tennant, and D. Schwartz, "VLSI implementation of neural network  
memory with several hundred neurons," in Neural Networks for Computing,  
182-87, 1986.

modified Hopfield network," IEEE J. of Solid-State Circuits, vol. 24, no. 4,  


[47] M. Ismail, "Four transistor continuous-time MOS transconductor," Electronics  

[48] N. Khachab, and M. Ismail, "Novel continuous-time all MOS four quadrant  
PA, May 1987.


for wafer-scale neuromorphic systems," Proc. IEEE First Int. Conf. Neural  


319, April 1989.


Index 1

A
accuracy 6, 15
accurate analog computation 26
activity level of a neuron 12
adaptable 32
adaptation 26
addressing 100
adaptation schemes 29
adjustable interconnections 27
adjustment of
the structure 5, 19
the weights 5, 20
algorithm 4
analog cellular neural network [152] 183
analog floating-gate synapses [106, 107] 180
analog multiplexer 110
analog neural networks 6, 14, 15
analog neural network arithmetic and
logic unit (ANNA) 15
analog neural network processors [73, 118, 123] 178
analog on-chip storage 11, 96
Analog Quadratic Neural Networks
(AQNNs) 11, 125, 129
analog quadratic neurons (AQNs) 143
analog synaptic weight array 15
analog vector-matrix multipliers [49] 176
analog VLSI neural systems [55, 56, 65, 66, 70] 177

analogy to computer evolution 57
analysis of operation 6
analytical explanation 23
ANN implementations 8
ANNA 15
ANNs 7, 41, 53, 86, 126, 151
AQNNs 136, 137
architecture design 41, 56, 126, 152
architectures
quadratic synaptic 36
linear synaptic 146
area 3, 171
array processors for NNs [18, 19] 174
artificial neural networks 4
associative memory 25
AT&T Bell Laboratories 15, 23
AT-MIO-16X 115
autonomous driving [5] 173
autonomous systems 1
available resources 88
B
backpropagation 30, 32, 44, 46
backpropagation [4, 86] 179
bandwidth 62
basis functions 29
batch training 34, 35
Bell Laboratories 15
Bell Northern Research (BNR) 93
Bellcore 14
benchmarks 36
biasing circuit 139
BiCMOS 21
BiCMOS analog neural network [117] 180
binary-logic examples
bio-compatibility 3
biological 126
neural networks 85
neural systems 14
neurons 53
systems 12
biologically-inspired models 3, 88
bipolar process 28
Boltzmann Machine [128] 14, 114, 181
bounded weights [89] 44, 179
BP 111, 112, 137
bumps 145, 149
bump-like functions 148
bump-like areas 160
C
CAD 101
California Scientific Software 15
Caltech group 15
Canadian Microelectronics Corporation (CMC) 93, 115, 137, 152
CASS 103, 106, 139, 140
capacitive storage 73, 95
CCD 20
chain-rule differentiation 44, 137
character recognition 126
canonic
input-output 61
of a MOS transistor 30
quadratic 31
characterize the processing hardware 26
classification problem 128, 142
clock feedthrough 99
clock-feedthrough-induced offset 98
closed-boundary discriminating surfaces 49
clusters 127, 128
CMOS process 3, 12, 15, 74, 132
CMOS4S 75, 77, 82, 98, 115, 137, 152
CMOS-compatible ANNs [149, 154] 183
CNAPS 13
comparison of existing systems 23
comparison of new systems 69
competitive-learning 29
complementary discriminating functions 119
complementary output areas 60
compromises 88
computational block 74
computational performance 2
computer architecture 57
computing with neural nets [87, 88] 179
Connection Machine [21] 13, 174
connectionist learning procedures [116] 180
constrained weight 36
continuous-time filters 16
contributions 153
convergence 35, 61, 70, 114, 129
convergence speed 155
convolution 147
counter-propagation networks 144
CSI 61
CSIA neuron 61
current summation 88, 132
current-accumulating unit 91
Current-Source-Inhibited Architecture (CSIA) 59, 68, 69, 70
current-to-voltage conversion 137
D
DASA 67, 69, 70
data clustering 125, 126
decision boundaries 118, 163
decoder 99, 108
dedicated hardware 5
design objectives 57
design philosophy 26
device-level design 152
difference-squaring operations 132
differentiable output units 28
digital implementations 12, 13, [135] 182
digital multiplier 12
digital storage 17, 73
digital switches 68, 70, 88, 94
digital transistors 59, 67, 91
Digital-Analog Switchable-Sign-Synapse Architecture (DASA) 67
 discontinuous synaptic relation 70
discriminating circle 47, 51, 55
discriminating functions 41, 64, 118, 123
discriminating hyper-plane 44
discriminating hyper-sphere 106
discriminating lines 106
discriminating surfaces 53, 56, 71, 141
 closed 46
 closed-boundary 58, 125, 126, 127
e lliptical 131
 non-planar 46
 quadratic-constrained 86
 spherical 41
discriminating transistors 169
distributed
 neuron-synapse 20
 parallel networks 3
 parallel processors 3
doughnut shape 109
dynamic behavior 87
dynamic MOS RAM [93, 121] 181
dynamic on-chip capacitive storage 73
dynamic range 3, 141
dynamic storage of the weights 18
dynamic-capacitive storage 96
 E
EEPROM 74, 157, 159, 172
EEPROM-based weight functions [69]
 178
effects of quantization [129] 181
efficient CMOS implementation 6
Elmasry M. I. 81
elastic-ring method 144
embedded knowledge 26
embedded parallelism 2
energy function 14
energy per connection 24
equi-value curves 64, 117
equi-value surfaces 53, 63-66, 119
error
 mean-squared 31, 113
error backpropagation 29, 44, 45, 47,
 67, 111, 137
error measure 29
ETANN neural network chip 15, 168,
 [42] 176
Euclidean distance 29, 126, 128, 131,
 145
example
 pattern classification 38
 pattern recognition 38
 excitation 58, 61
 excitatory input 89
 experimental chip 116
 external training 155
 F
fabrication 81
fanin/fanout 130
fault tolerance 130
feature space 28, 41
features
 binary 30
 continuous-valued 30
feedback processes 5
feedforward networks 28, 41, 49, 153
feed-through cells 102
field-programmable analog arrays 20,
 [59] 177
figure of merit 107
flexibility 10
floating-gate analog MOS memory [95,
 96] 179
floating-gate devices 74-74, 79, 80, 153,
 154, 158, 159
floating-gate EEPROM cells [102] 180
four-quadrant multipliers [48] 176
Fourier analysis 29
Fowler-Nordheim tunneling 158
function approximation 6, 11, 44, 132,
 145, 149, 150, 153
functional characteristic equation 27
functions
 discriminating 30
inhibitory input 89
input patterns 28
input-output transformations 29
Intel 14, 107, 168, 172
Intel’s Flash-memory process 14
intelligent machines 2
intelligent MOS transistors 53
interleaved transistors 139
iWARP systolic computer 13, [25, 26] 174
J
Japanese Ministry of International Trade and Industry (MITI) 24
K
Keithley source and measurement units 82, 140
Kohonen feature mapping 144
labeled training set 128
leakage currents 98
learning 2, 25, 113
learning machines 1
learning rate 114
learning vector quantization 144
learning weight 20
linear MOS resistances 16
linear summer 28
linearly operating MOS devices 166
linear synapses 41-46, 127-128
linear-synapse model 7
localized storage 27
localized storage of parameters 25
logic functions 44
Lont et al 29
low power consumption 26
low-level silicon building blocks 5, 88
low-level silicon devices 6, 22
low-level silicon devices 22
M
many-transistor block 3
margin 69, 70
MasPar MP-1 architecture 13, [23, 24] 174
massively-parallel collective processing 14
matching of transistors 139
MATLAB 53, 65, 117
McCulloch and Pitts 27, [75] 178
MDAC (multiplying D/A) type of synapses 18
mean-squared error 31, 113
measurements 119
memory devices 157
metal-nitride-oxide-semiconductor (MNOS) 20, 157
microprobes 82
microprocessors 57
MIMD 13
minority-carrier injection 98
mismatch 104
MIT J-Machine 13
MIT Lincoln Laboratory 18
MITI 24
Mitsubishi Electric Corporation 13, 14
mixing function 27
MNOS 20, 158
modal expansion 29
modal-analysis solution 29
model-free distributed learning [91] 113
modeling 73
modularity 89
MOS capacitor 79
MOS implementation technology 5, 28
MOS-transistor-like processing elements 6
motivation 2
multi-gate CMOS transistors 74
multi-layer networks 29, 142, 143
multiplexers 108, 110
multiplication 67, 127
analog 15
nonlinear 23
non-perfect 22
multipliers 128
multiplying D/A converters 18
N
NAND decoder 99, 108, 109
NAND problem 49, 56
National Instruments 107, 115
nature-like approach 53
Nestor Inc. 14
networks
  bounded-weight 44
  quadratic 29, 41
  semi-quadratic 39
neural
  algorithms 2
  application 5
  approach 3
  architecture 2
  computation 11
  hardware 2, 7
  machine 2, 3
  solution 3, 5
  solutions 5
  system 3
neural computing
  electro-optic 11
neural networks 2, 126
  analog CMOS 20
  current-mode 19
  hardware-compatible 23
  linear-synapse 41, 127
  pulse-signal 12
  quadratic 23, 29
  reconfigurable 20
  silicon-friendly 56
  sub-threshold design 19
  switched-capacitor 18
  wafer-scale 12
neural-algorithm implementation 5
neural-based A/D converter [45] 176
neural-hardware architecture 57
neurocomputation 2, 11, 29, 73
neuro-computation in electromagnetics [79] 178
neuroMOS 74
neuron MOSFET 21, 22, 74, [67] 177
neuronal output functions 28
neurons 73
  linear-synapse 41
Ng W. T. 76
Ni1000 Recognition Accelerator chip 14
noise 2, 96-97
noise and offset reduction 26
noise-corrupted data 29
noise immunity 19, 130
noise in training 114
noise margin 38, 69, 70
noise suppression 29
nonlinear programming problems 18
nonlinear synapses 22
nonvolatile 157
nonvolatile analog memory 18, 157
Northern Telecom 75, 98, 115, 137, 152
O
offset-cancellation unit 110
offset reduction 115
on-chip area 3
on-chip backpropagation learning [146] 182
on-chip learning 155, [29] 175
on-chip resources 14
on-chip weight-refresh circuitry 155
operational amplifier 15
optical character recognition (OCR) 13, 142
optoelectronic devices 20
optoelectronic future computers 11
optoelectronic neuron [64] 177
organic materials 53
output unit
  saturating nonlinear 28
  sigmoid 47
P
parallel
  distributed networks 5
  machines 2
  neural architectures 2
  neural networks 2
parallel array machines 12
parallel gradient descent method [131] 181
parallelism 2
parallelism at

190
a training-set level 12
all levels 12
Partial Random Search Algorithm [130]
114
pattern classification 11, 30, 37, 46, 57, 169
pattern recognition 14, 28, 126, 142, 152
pattern-classification 30
pattern-recognition applications 14
perceptrons 28, 30, 44, 73, 86, 128
performance 38, 39
performance limits of computers [10] 173
performance-flexibility chart 11
peripheral circuitry 107
philosophy of design 26
photomicrograph 137
pipelining 12
pixel array 38
poly-1 to poly-2 capacitors 82
postal-code recognition 30
power consumption 3
power dissipation 53, 55, 59, 70, 88, 100, 130, 169
precision 96
Probabilistic Neural Networks 14, 129
problem-solving procedure 4
processing element
  single-transistor 53
processing elements 5, 59, 88, 127
programmable architecture 20
pronouncing English text [7] 173
prototype vectors 30
prototypical grouping 30
pseudo-bumps 145
pulse-coded neural networks [12] 174
Q
quadratic
  ANNs 86
  features 127
  forms 30, 129
  function 5
  networks 29, 30, 129
  neurons 126, 149
  representation 30
  surfaces 86, 130
  synapses 6, 137
  synaptic relation 25
quadratic grouping 30
quadratic neural networks 129, [82, 83, 84, 85, 138, 149] 178
R
radial basis functions 14, 129, 143
radius 106
RBFs 129
RCE 129
real-time hardware 112
real-world information processing 2
recognition accelerator [36] 175
reconfigurable neural networks 19, [51] 176, [58] 177
Reduced-Instruction-Set Computer (RISC) 12
redundancy 58
reference vectors 29, 128, 144
refresh rate 169
refreshing analog values 18
refreshing period 83, 84
registers 115
regularity 106
regularity and locality of data 12
relaxation networks [33] 175
reliability issues 83
resistive synaptic weights 15
resistive weights
  adjustable 16
  fixed 15
resolution 84
Restricted Coulomb Energy (RCE) 14, 129
retina 38
reuse resources 89
ridges 145
RISC 12, 58, 87
RISC processor array for ANN [17] 174
Rosenblatt group 28
rotated quadratic forms 30
sampled-noisy-measured data 29
SANNs 6, 11, 85, 153, 163
scaling of MOS technology 163, 164, [151] 183
semi-quadratic synapses 30, 39
separating function 142
serial computers 2
Shibata et al 21, 74
shifted impulses 147
short-channel devices 81, 163
sigmoid 7, 21, 43, 53, 90, 104, 106, 108, 137, 149
sign-of-synapse register 62
sign-select block 93
silicon 28
silicon-friendly implementations 53
SIMD 13
simple networks (SNs) 50
simple processing elements 3
Simple SyMOS Networks (SSNs) 55
simulated annealing 114, [127] 181
simulation program 6, 36
simulator 36, 67
single-instruction multiple-data 13
single-assignment language 12
single-transistor network 6
SNAP 13
SNs 50, 53
special-purpose design 89
speech-to-speech translation 2, [6] 173
speed 24, 167
squaring operation 29
SRAM 91, 93, 94
SRAM cell 87, 93, 100, 107
SSNs 53, 55
SSSA 59, 70, 71, 87, 89, 118, 123, 152
Stanford 14
static storage of analog voltages 18
stochastic gradient 34, 35
stochastic solution 58
stochastic search 114
storage element 74
storage function 27
submicron CMOS technologies 6, 157
subthreshold conduction 19, 98, [56] 177
subtracting-squaring unit 132
subtracting-squaring unit (SSU) 138
Switchable-Sign-Synapse Architecture (SSA) 59, 62
switched-capacitor filters [111, 119, 120] 181
switched-capacitor neural networks 18, [53, 54] 176
symbolic versus connectionist [2] 173
SyMOS 6, 53, 55, 57, 62, 68, 73, 74, 77, 86, 91, 92, 115, 152
floating-gate 162
limitations 55
networks 56
synapse-MOS 6, 53, 73
Synapse-MOS [108] 180
Synapse-MOS Artificial Neural Networks (SANNs) 11
synapse-MOS device 73
synapses
  fully-quadratic 152
  general non-linear 22
  linear 41, 44, 45, 73
  linear multiplicative 8
  MOS-compatible semi-quadratic 30
  quadratic 46, 62, 86
  quadratic CMOS 126
  semi-quadratic 30, 31
  short-channel linear 164
  simple MOS-compatible 25
  single-transistor 41, 50, 54, 55
  single-transistor quadratic 39
two single-transistor 51
synaptic
  combining blocks 28
  controllable parameters 28
  element 6
  function 5
  information 28
  memories 28
  mixing function 27
  operation 172
output 27
relations 25, 35
weight 27, 64
system-level performance 88
systolic 13
T

taxonomic trees
electrical signals 8
flow of information 8
generality of usage 10
implementation technology 9
performance versus flexibility 10
physical implementation media 9
training methods 8
types of synaptic units 9
types of synaptic-weight storage 10
technologies other than CMOS 20
technology
BiCMOS 20
CCD 20
CMOS 18
GaAs 20
gate-array 12
high-quality-oxide 162
silicon MOS 30
technology-compatible blocks 57
test problems 6, 68, 115
testability 108
the elastic ring method 144
thin oxide 154
three-layer networks 31, 32
threshold gradient 103
threshold voltage 52, 77, 80, 86, 104
externally-controllable 153
externally-controllable 31, 74, 75
training 2, 25
algorithm 7, 32, 36, 62, 67, 107, 113, 114
algorithms 2, 155
cases 7
coefficients 35
effect on offset 115
equations 25, 32, 68
mode 34
parameters 5
phase 58
process 7, 31, 44, 70, 130
rule 44
set 3, 5, 34, 128
speed improvements 30, 112
time 58, 69, 125, 155
update rate 113
training algorithm 28
transistor 3, 5, 39, 49, 50, 56, 59, 64, 71, 74, 89, 94, 140
transistor capacitances 79, 80, 92
transistor switch 94
transmission gates 91, 94, 98
trimming 110
trimming of analog circuits 157
tunnelling 160
U
UCL 143, 144
universal approximators [142] 182
unsupervised competitive learning 6, 125, 131, 143, 152
unsupervised classification 128
unsupervised data clustering 25
updates of coefficients 35
V
vector-matrix multiplication 15, 127, 132
VLSI analog hardware 14
VLSI analog memories [97] 179
VLSI architectures for neural networks [13,15,16] 174
VLSI implementation 6, 23, 89, 101, 151
of SSSA 123
voltage-controlled resistors 16
W
wasted resources 89
way to go 70
weak-inversion operation 19
weight perturbation 36, 113, 114, 115, [126] 181
weight values 5, 107
weighted sum 74, 147, 149
weighted-sum device 53
weights
  adjustable 20
  bounded 25, 46, 59
  constrained 44
  learning 20
  on-chip storage of 73
  reconfigurable 19
  unbounded 46, 59
weight-storage 3, 17
  techniques 10, 17, 18
winner-take-all 144
wired-sum 90, 136
WP 114, 115
write-only memory 108
X
X-decoder 99, 100, 108
XNOR 67, 68
XOR 44, 45, 49, 68, 69, 86
XOR-problem solution 47, 129
x-select 101
Y
Y-decoder 99, 108
Y-select 101