ADAPTIVE PATTERN RECOGNITION BY MINI-MAX NEURAL NETWORKS as a part of an intelligent processor

Harold H. Szu
NRL, Code 5756, Washington, D. C. 20375. Tel. (202) 767-1493

Abstract- In this decade and progressing into 21st Century, NASA will have missions including Space Station and the Earth related Planet Sciences. To support these missions, a high degree of sophistication in machine automation and an increasing amount of data processing throughput rate are necessary. Meeting these challenges requires intelligent machines, designed to support the necessary automations in a remote space and hazardous environment. There are two approaches to designing these intelligent machines. One of these is the knowledge-based expert system approach, namely AI. The other is a non-rule approach based on parallel and distributed computing for adaptive fault-tolerances, namely Neural or Natural Intelligence (NI). The union of AI and NI is the solution to the problem stated above.

The NI segment of this unit extracts features automatically by applying Cauchy simulated annealing [Phys. Lett. A122, p.157; Proc. IEEE, V.75,p.1538] to a mini-max cost energy function. The feature discovered by NI can then be passed to the AI system for future processing, and vice versa. This passing increases reliability, for AI can follow the NI formulated algorithm exactly, and can provide the context knowledge base as the constraints of neurocomputing. Such integration is exemplified by the pattern recognition Human Visual Systems; tracking of gray scaled objects for instance. Consequently, both AI and NI can work together to solve the same problem by unifying into an intelligent processor.

The mini-max cost function that solves the unknown feature can furthermore give us a top-down architectural design of neural networks by means of Taylor series expansion of the cost function. A typical mini-max cost function consists of (1) the sample variance of each class in the numerator, and (2) separation of the center of each class in the denominator. Thus, when the total cost energy is minimized, the conflicting goals of intraclass clustering and interclass segregation are achieved simultaneously. This Taylor expansion variable is a neuronic vector representation which traces along a Peano's curve. A selective space-filling capability exists when a more detailed spatial resolution becomes desirable at the picture where an interesting change occurs [IJCNN-90, D.C., p. II-76].

INTRODUCTION

Research and operations that support NASA's missions have experienced an increasing volume of data that requires automated information processing, among others (e.g. Discovery shuttle between the space station and the earth shown in Fig. 1 Top). One necessity is the next generation smart sensors. They are needed for two reasons. First, they are needed to perform multisensor data auto-fusion (between thematic mapper spectral band imageries and high spatial resolution imageries) in order to improve the picture resolution beyond the geometrical corrections and proper registrations. They are also needed to extract features to identify space rocket boosters shown in Fig. 1 center (provided by courtesy of T Dworetzky). From left to right, these are Goddard (1941), V-2 (German, 1944), Redstone (1961), Atlas Centaur (1962), Delta 3920 (1982), Titan 34D (1982), Saturn V (1967), Ariane (European 1979), Energia (Soviet 1987), and Conestoga II (Future). Automated feature extraction can also be useful to update maps as well as to help manage earth's resources. For example, an extra road through the palm forest was discovered by Environmental Research Institute Michigan (ERIM) in Fig. 1 (Bottom) D.

The trend in the modern telecommunications is toward multi-media, higher-speed and increased intelligence (Fig. 2 (a)). Thus, another application of intelligent machines is, according to NTT Review (Vol. 1, No.1 May 1989), a Broad-Band Integrated Service Digital Network (B-ISDN) that has been proposed and will probably undergo construction around 1995 (see Fig. 2 (b), used with permission). B-ISDN will have the capability of processing voice, images, and text, simultaneously based on neurocomputing.

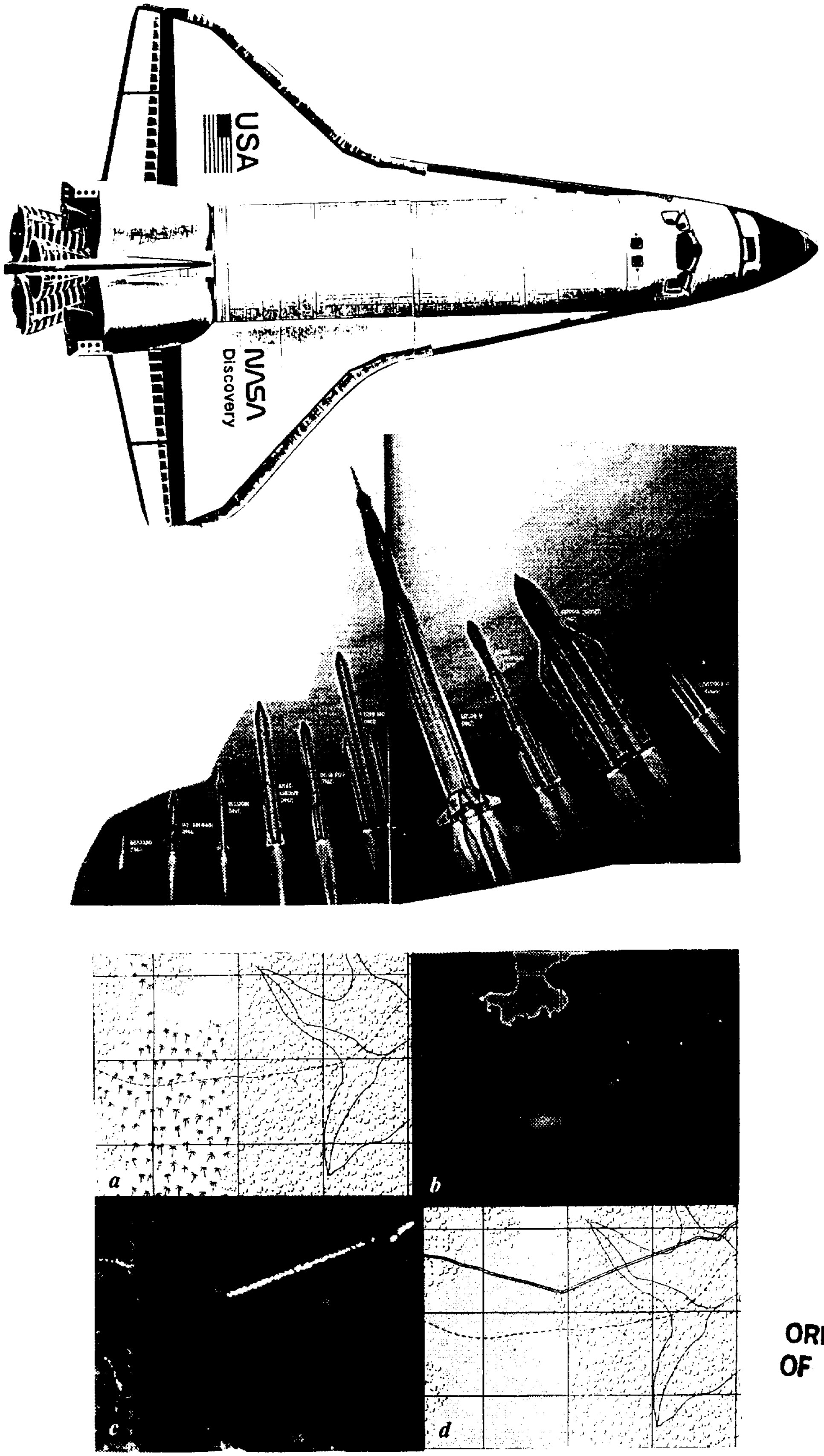


Figure 1. (Top) NASA's Space Shuttle Discovery. (Center) Feature extraction to identify various space rocket boosters. (Bottom) Automatic feature extraction to update maps and to help manage earth's resources.

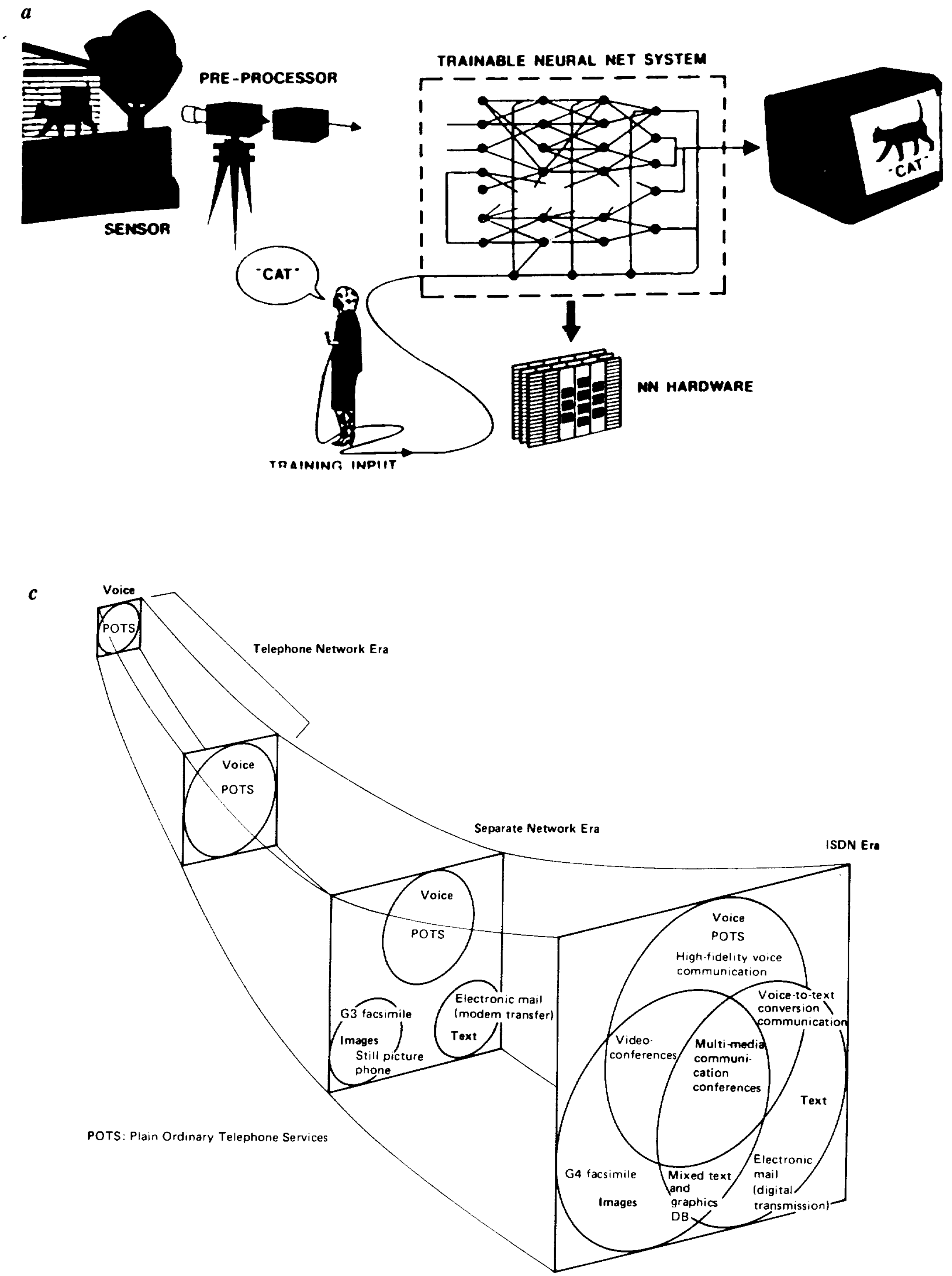

a
PRE-PROCESSOR
TRAINABLE NEURAL NET SYSTEM
SENSOR
"CAT"
"CAT"
NN HARDWARE
TRAINING INPUT
c
Voice
POTS
Telephone Network Era
Voice
POTS
Separate Network Era
ISDN Era
Voice
POTS
G3 facsimile
Images
Still picture phone
Electronic mail (modem transfer)
Text
Voice
POTS
High-fidelity voice communication
Voice-to-text conversion communication
Video-conferences
Multi-media communi-cation conferences
Text
G4 facsimile
Images
Mixed text and graphics DB
Electronic mail (digital transmission)
POTS: Plain Ordinary Telephone Services

Figure 2. (a) The trend towards Multi-media Communication. (c) Mixing of voice and image spectral components by an associative memory.

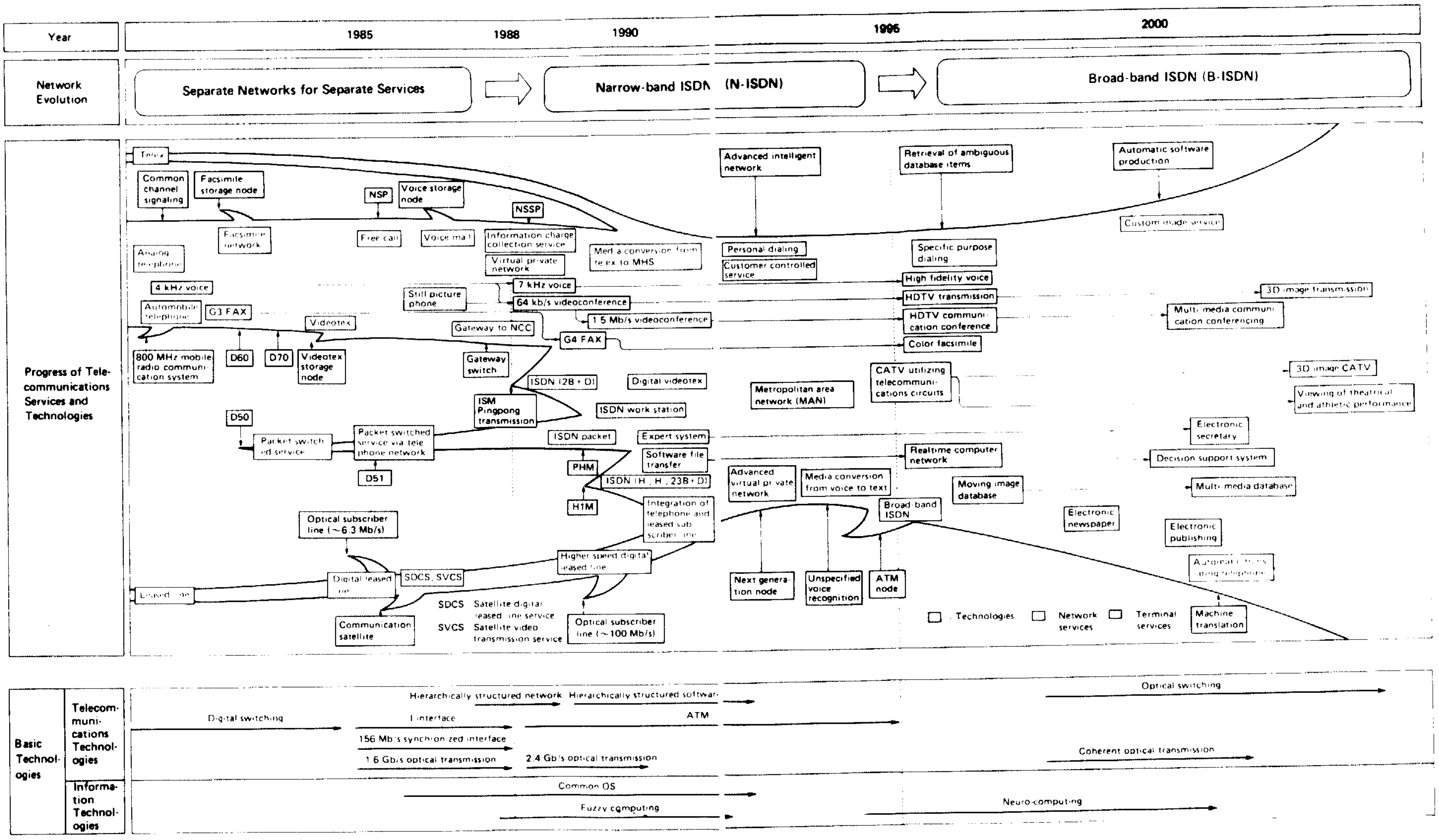

Year
1985
1988
1990
1995
2000
Network Evolution
Separate Networks for Separate Services
Narrow-band ISDN (N-ISDN)
Broad-band ISDN (B-ISDN)
Progress of Telecommunications Services and Technologies
Telex
Common channel signaling
Facsimile storage node
NSP
Voice storage node
NSSP
Facsimile network
Free call
Voice mail
Information charge collection service
Virtual private network
Media conversion from telex to MHS
Analog telephone
4 kHz voice
Automobile telephone
G3 FAX
Videotex
Still picture phone
7 kHz voice
64 kb/s videoconference
1.5 Mb/s videoconference
Gateway to NCC
G4 FAX
800 MHz mobile radio communication system
D60
D70
Videotex storage node
Gateway switch
ISDN (2B + D)
Digital videotex
ISM Pingpong transmission
ISDN work station
D50
Packet switched service
Packet switched service via telephone network
ISDN packet
Expert system
Software file transfer
D51
PHM
ISDN (H, H, 23B + D)
H1M
Integration of telephone and leased subscriber line
Optical subscriber line (~6.3 Mb/s)
Higher speed digital leased line
Leased line
Digital leased line
SDCS, SVCS
Communication satellite
SDCS Satellite digital leased line service
SVCS Satellite video transmission service
Optical subscriber line (~100 Mb/s)
Advanced intelligent network
Personal dialing
Customer controlled service
Metropolitan area network (MAN)
Advanced virtual private network
Media conversion from voice to text
Next generation node
Unspecified voice recognition
ATM node
Retrieval of ambiguous database items
Specific purpose dialing
High fidelity voice
HDTV transmission
HDTV communication conference
Color facsimile
CATV utilizing telecommunications circuits
Realtime computer network
Moving image database
Broad-band ISDN
Automatic software production
Custom made service
3D image transmission
Multi media communication conferencing
3D image CATV
Viewing of theatrical and athletic performance
Electronic secretary
Decision support system
Multi media database
Electronic newspaper
Electronic publishing
Automatic interpreting telephone
Machine translation
□ Technologies
□ Network services
□ Terminal services
Basic Technologies
Telecommunications Technologies
Information Technologies
Digital switching
Hierarchically structured network
Hierarchically structured software
I interface
ATM
156 Mb/s synchronized interface
1.6 Gb/s optical transmission
2.4 Gb/s optical transmission
Optical switching
Coherent optical transmission
Common OS
Fuzzy computing
Neuro-computing

Figure 2. (b) The NTT Review forecasts a Broad-Band Integrated Service Digital Network (B-ISDN) by 1995 using neurocomputing.

Associative memory can mix the voice and image spectral components by vector outer products (shown in DARPA's Report as a dotted matrix array in Fig. 2 (c)). This cross correlation information can be processed at the data rate of about 3 Gb/s. Usually such a high data rate requires optical computing based on optical switching and coherent optical transmissions. However, neurocomputing's debut in telecommunication is predicted by NTT to be five years earlier than optical computing, despite extensive research efforts in optical computing by AT&T & others.

REVIEW OF NEURAL NETWORK LEARNING ALGEBRA

Neural network computing is a nonlinear system that satisfies 4 none-principles with the fifth none-principle remains to be worked out. These are: (1) a none-linear threshold logic of neurons, (2) a none-local associative memory, (3) a none-stationary neurodynamics, and (4) a none-convex system energy, meaning more than one extremum in the energy landscape. The first one is known to us 30 years ago, when the Rosenblatt's perceptron was proposed to be a random collection of neurons. It had been shown by Minsky and Papert to be insufficient for the natural intelligence, and thus giving the need to the birth to AI. These 4 none-principle can be approximated by (1) piecewise-linear, namely binary neurons, (2) piecewise-local, namely the rank-1 vector outer product, (3) piecewise stationary, namely iterative revisions, and (4) piecewise convex, namely local gradient descents. In these controlled approximations, these interwoven complex principles become decoupled and amenable to powerful computer simulations. Since then NI has been coming a long way, there remains a missing fifth none-principle. Such a none-programming learning principle has been claimed by some, but the hidden teachers/programmers remain to be unraveled to most of us for pedagogical reasons. This is the state of the art of neurocomputing theory.

Neurocomputing learning algebra are based on the variants of Hebbian ideas. Giving two random inputs of two neuron firing rates about 100 Hz, $\mathbf{u}_i\ \mathbf{u}_j$, there are limited algebraic structures that one can manipulate with to extract meaningful information. If the change of synaptic weight at the ith and the jth interconnection, ΔW_{ij}, should be related to the inputs as follows:

- **Correlation Learning:** $\Delta W_{ij} \approx u_i\, u_j$

  (maximum information-exchange rule between a pair of random firing rates)
- **Gradient Learning :** $\Delta W_{ij} \approx (D_i - v_i)\, u_j$

  (Error correction by a pre-set output goal D_i that decides when the change of actual output v_i stops: the delta rule)
- **Competitive Learning** $\Delta W_{ij} \approx u_i\, (u_j - W_{ij})$

  (any change must balance against the old cluster establishment w_{ij})
- **Differential Learning** $\Delta W_{ij} \approx (du_i/dt)\,(du_j/dt)$

  (Only time rate changes, derived by Taylor series expansion of $\mathbf{u}_i(t)$, matters)

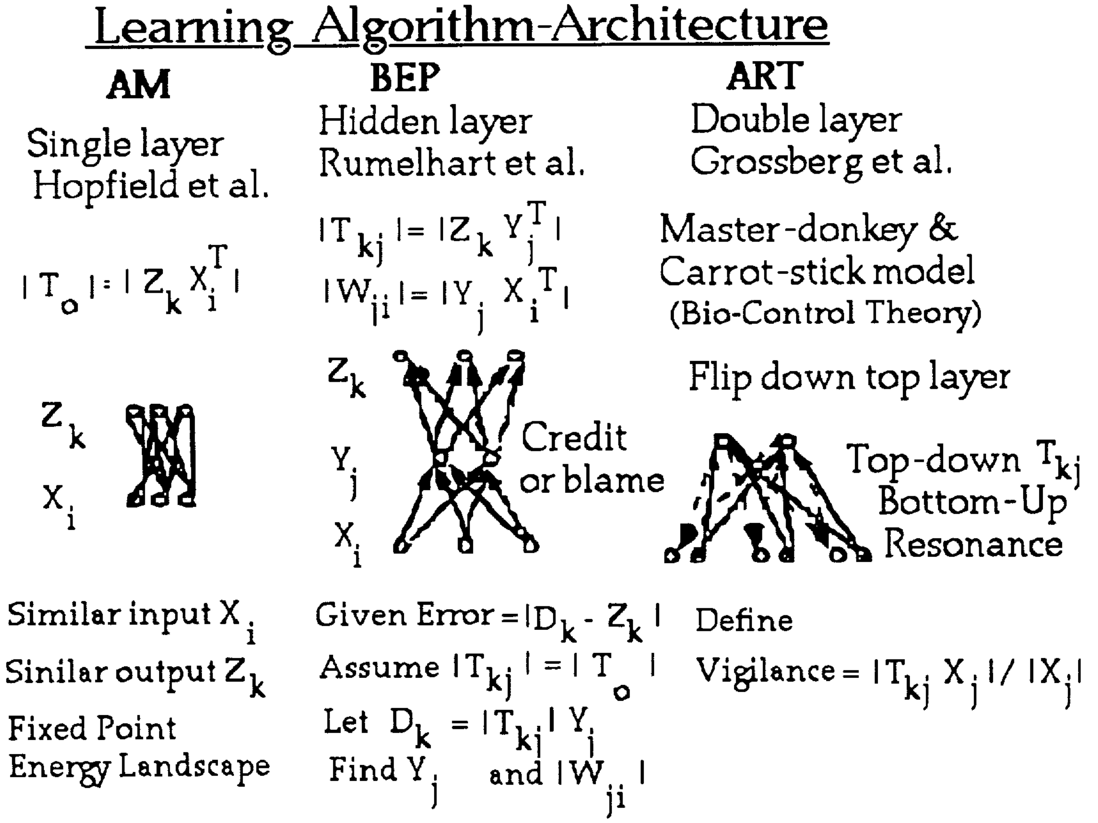
REVIEW OF NEURAL NETWORK ARCHITECTURES

Neural network architectures are important for parallel and distributed computing. There are: one layer of Hopfield's Associative Memory (AM), two layers of Grossberg's Adaptive Resonance Theory (ART), and three layers of Rumelhart's Back Error Propagation (BEP), as shown schematically in Fig. 3, Learning Algorithm-Architecture as follows.

• In the left hand column of Fig. 3, similar inputs X_i are mapped into similar outputs Z_k in a feature space. Such a (hetero-associative) matrix memory is formed by the vector outer product forming a matrix denoted as $|Z_k X_i^T|$, where the superscript T stands for the transpose of the column vector X (indexed with the component i) and the column vector becomes a row vector. Matrix memory is a static version of Hopfield neural networks, because of the fixed point coding between the input and the output requires no learning. By a fixed point coding we mean that "write-by-outer-product" and "read-by-inner-product" and using the matrix-vector operation without iterations.

• In the middle column of Fig. 3, when the similar inputs produce the surprising outputs, an extra layer is introduced to interpolate these abnormal results by means of supervised training. The difference $|D - Z|$ from the output Z with respect to the desired output D is considered to be the error propagates backward by means of a local gradient descent methodology. The system can have the potential for the generalization. There are several theories about the size of the so-called hidden layer and the ability to do the abstraction (with more neurons than that of input nodes) or the generalization (with a fewer neurons). The degree of freedom must match the number of sample classes to be classified based on the orthogonal feature space min-max concept described in the following section. In such a quasi-orthogonal storage, this rule seems to be reasonable in assigning credit-or-blame.

- When the desired output D is not yet known, Grossberg model of Adaptive Resonance Theory (ART) becomes handy. It might be thought as to flip down the unknown output layer in order to compare the unknown input directly as shown in the righthand column of Fig. 3. The master has its own top-down wires **T** $_{jk}$ (shown by dotted lines), while the donkey has its own bottom-up wires b $_{ij}$. In order to carry out automatically the clustering technique by following the leader, the top layer master puts his feet into donkey's input x_j to test his own normalized prediction $|S<x_j|T_{jk}|x_k>|/|S<x_j|x_k>|$ with a predetermined parameter, called the vigilance parameter between 0.5 to 0.9. Therefore, the difference between the traditional control theory with the negative feedback and the neural network is that both the incentive/carrot and the punishment/stick are used in the biological model having both the excitation and the inhibition exerted at different parts of the self-organized system.



Learning Algorithm-Architecture

AM

Single layer
Hopfield et al.

$|T_o| = |Z_k X_i^T|$

Z_k
X_i

Similar input X_i
Sinilar output Z_k
Fixed Point
Energy Landscape

BEP

Hidden layer
Rumelhart et al.

$|T_{kj}| = |Z_k Y_j^T|$
$|W_{ji}| = |Y_j X_i^T|$

Z_k
Y_j
X_i

Credit
or blame

Given Error = $|D_k - Z_k|$
Assume $|T_{kj}| = |T_o|$
Let $D_k = |T_{kj}| Y_j$
Find Y_j and $|W_{ji}|$

ART

Double layer
Grossberg et al.

Master-donkey &
Carrot-stick model
(Bio-Control Theory)

Flip down top layer

Top-down T_{kj}
Bottom-Up
Resonance

Define
Vigilance = $|T_{kj} X_j| / |X_j|$

Figure 3. Review of Learning Algorithm-Architecture.

An interesting taxonomy dilemma about counting of layers is due to the ambiguity of counting about layers of neurons or about layers of interconnects. The single-layer Hopfield architecture seems to have two layers of neurons, with respect to the three layers of Rumelhart architecture. On the other hand, the Hopfield architecture is considered to be a single layer on a VLSI design. This dilemma may be resolved by asking: What is more important in counting, the layer of interconnect synaptic weights, or the layer of neurons ? Since the synaptic weights contain the important memory information, then Hopfield's network should be counted as one layer.

Designs of Energy Cost Functions in A Neuronic Vectorial Representation

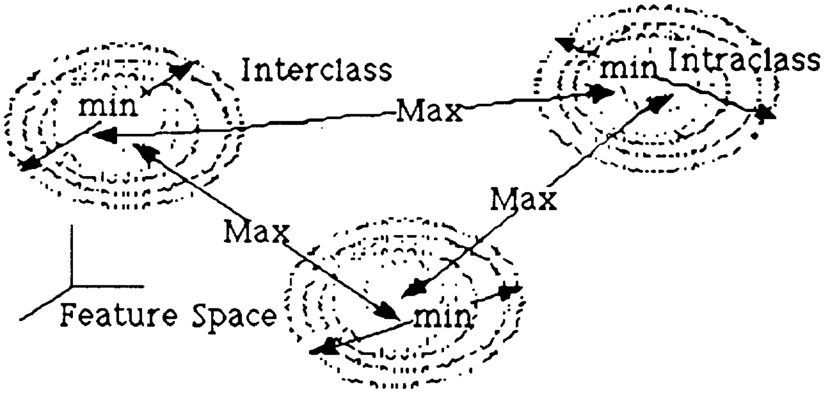
An important question for practical applications is how to speed up the training process and to insure a fast convergence of weight adjustment? We have suggested a general procedure of Taylor series expansion of the clustering-declustering mini-max energy to estimate the synaptic weights. Here, we extend the procedure by a self-consistently variational technique to make the truncated higher order terms of the Taylor series negligible.

A top-town design of a hard-wired neural network algorithm has been initiated by Hopfield, et al, for constrained optimizations. We consider a supervised top-down design goes beyond Hopfield's attempt. The minimum clustering of the alike and the maximum declustering of the disalike seems to be two contradicting goals. A tradeoff can be mathematically constructed by the linear combination of those pairs alike in the numerator and the pairs of disalikes in the denominator of a mini-max energy formalism (schematically shown in the cost energy expression of Fig.4).

Top Down Design of Hard-Wired Neural Networks

Mini - Max Energy Principle

$$\text{Energy} = \sum \frac{1}{|\text{Interclass Distance}|} + \sum |\text{Intraclass Distance}|$$



Taylor Expansion to derive multiple layer interconnects

$$\text{Energy} = -\frac{1}{2}\sum_{i,j} T_{i,j} V_i V_j - \frac{1}{6}\sum_{i,j,k} T_{i,j,k} V_i V_j V_k - \cdots$$

Figure 4. A top-down design of Neural Networks.

Let us consider some application of pattern classifications. The class of physically different objects {o, O, p, P, q, Q} need to be cleverly pre-processing by a smart sensor mimicking our eyes or by ourselves and then endow our wisdom about how we classify the set with a functional mapping into a feature space { $o(V_i)$, $O(V_i)$, $p(V_i)$, $P(V_i)$, $q(V_i)$, $Q(V_i)$} spanned by a sufficient set of neurons V_i mimicking the human visual system of the brain (Szu & Scheff 1989). The first term of the energy in the denominator is similar to the Coulomb energy of repulsive electric charges (reduced Coulomb energy model of Cooper, et al., and Lorentz forces of Sayeh, et al), and the second term in the numerator is similar to the least square method(when in an arbitrary power becomes Kohonen's kth norm clustering method).

While the first order derivative is reserved for the aforementioned neurodynamics equations, the second order derivative when it is evaluated at the equilibrium value: $V_i=V^{(o)}_i$, $V_j=V^{(o)}_j$ becomes the Taylor's coefficient

$$T^{(o)}_{i,j} = (\partial^2 E/\partial V_i \partial V_j)|_{V_i=V^{(o)}_i, V_j=V^{(o)}_j}$$

Then, the Hopfield-like hard-wired interconnect $T^{(o)}_{i,j}$ become soft-wired $T_{i,j}$ by means of the Hebbian learning that make the cubic order negligible.

$$T_{i,j} = T^{(o)}_{i,j} + \varepsilon\, \delta V_i\, \delta V_j$$

$$T_{i,j,k}|_{V_i=V^{(o)}_i+\delta V_i, V_j=V^{(o)}_j+\delta V_j} << T^{(o)}_{i,j}$$

Similarly, the procedure can be analogously extended to the three layers:

$$T_{i,j,k}=T^{(o)}_{i,j,k} + \varepsilon\, f(\delta\, V_i \delta V_j \delta V_k)$$

which makes the next fourth order derivative negligible. The case of hidden layer architecture means that $T_{i,j,k}$ is a block-diagonalized tensor of which the input ith layer can not communicate with the kth layer output layer without going through the jth hidden layer of neurons.

We can show that a single layer of a fully interconnected Hopfield network of five neurons of 25 interconnects can be reduced by the use-it-or-lose-it principle to 6 interconnects. Without actual physical rearrangement, it becomes topologically equivalent to a three layer of Minsky nets by clamping 2 input neurons and 1 output neuron to be trained repeatedly with the "exclusive OR" input-output relationship. This illustrates the second computing principle that can not only be used to determine the learning algorithm but also used to derive the neural network architectural change consistently.

Experimental aspect of the unified learning theory has been demonstrated by NTT scientists using several life neurons, extracted from the hippocampus of chicken brain. In delayed video recording they have shown that neuronic hair fibers T_{ij} grow for seeking out the nutrition and other neurons, in a competitive learning fashion. The winning hair fiber has grown fatter into a mature axonic interconnect, while the other loser shrinks off, on an electronic chip substrate covered with the life sustaining liquid. The present unified theory is possible to explain such a growing synapse because of the extended McCulloch-Pitts neuron model with two transfer functions for two independent degrees of freedom, namely the sigmoidal firing rate transfer function and the synaptical transfer function. Such a model has been coined with a name of the hairy neuron neural networks (Szu, 1989).

NEUROCOMPUTING IS MORE THAN PARALLEL COMPUTING

The famous von Neumann bottleneck, 10^9 operations per second (ops), for a sequential computer has been circumvented by parallel computing models which require lock steps among multiple processors controlled by a precision clock cycle that has unfortunately created the second bottleneck, 10^{12} ops, (that I wish to call) **the five W bottleneck**, namely "who should do what, when, where and how" bottleneck, due to the necessary trade off between the actual execution and the communication for timing and assigning jobs among multiple pipe lines. Therefore, the following asynchronous neurocomputers are fundamentally important and can make possible a cheaper VLSI fabrication of neurocomputers. Although the fabrication advantages without the demand of timing accuracy is conceivable, but without neuronic processor timing the dynamics about when and how the collective computing is finished requires mathematical insurance. Thus, we will prove three theorems for three neurocomputing learning mechanisms with hard-wired, soft-wired, and brittle-wired interconnects. Our purpose is to point out the possibility of allowing the system to determine its own topological structure, by means of a dynamically reconfigurable hairy neuron model described below. In order to minimize the overall energy, dynamically reconnected neurofilaments T_{ij} (located at the protein-mediated output axons) can play an equally important role as the synaptic junction W_{ik} weight adjustments (located at the ion-mediated input dendrite tree). The extra degree of freedom of the hairy neurons is the synaptic transfer function having a nonnegative slope

$$T_{ij}=f(W_{ik}); \qquad (dT_{ij}/dW_{ik}) \geq 0$$

while the McCulloch-Pitts neuron model has one internal degree of freedom prescribing the firing rate transfer squash function

$V_i = g(U_i)$; $(dV_i/dU_i) \geq 0$

The following three convergence theorems all depend on the mathematical truth that (d (any real quantity)/d $t_i)^2 \geq 0$ with respect to any time axis:

$t_i = t_i^{(o)} + \varepsilon_i\, t,$

where the information arrival time has an arbitrary initial time $t_i^{(o)}$ and a positive time scale factor $\varepsilon_i > 0$ with respect to a collective or universal time axis t.

(1) Hopfield-like Asynchronous Computing by Hard-Wired Nets $E_1(V)$

We consider first a system of a hard-wired neural networks. We assume a network activity energy $E_1(V)$ in terms of the output firing rate vector **V** with the components V_i whose i index runs from one neuron to a million, e.g. the mega-Cray. We can use either $E_1(V_i)$ or $E_1(V)$. The input firing rate to the ith neuron is wired according to the McCulloch-Pitts model with the bias Θ_i:

$$U_i = \Sigma_j W_{ij} V_j + \Theta_i. \qquad (1)$$

The synaptic weight W_{ij} at the jth junction of the ith neuron input dendrite has a physical gap, analogous to the spark plug, through which the ion-mediated firing rates from other outputs V_j are collected. Then, Hebbian learning would mean the changing of the spark plug gap for tuning up the car engine firing rates. Due to the diffusion of discrete ions through those synaptic junctions, the firing rate fluctuates like a discrete time series at the molecular time scale t in the order of one millisecond. The information flow with a reduced fluctuation of the neurotransmitters plays an important annealing role for the global convergence of the neurodynamics.

Each neuron can be operated at its own time axis:

$$t_i = t_i^{(o)} + \varepsilon_i\, t, \qquad (2)$$

where the information arrival time has an arbitrary initial time $t_i^{(o)}$ and a positive time scale factor $\varepsilon_i > 0$ with respect to a collective or universal time axis t. This asynchronicity is essential to account for different information flow rates due to the biological inhomogeneity at neuronic level.

The total input is instantaneously mapped to the output by a nonlinear transfer function g,

$$V_i = g(U_i) \qquad (3a)$$

A squash function known in biology as a sigmoidal function is often used

$$g(x)=1/(\,1 + \exp(-x)) \qquad (3b)$$

for the simplicity of the analytic slope:

$$dg/dx = g(1-g) \geq 0, \qquad (3c)$$

which vanishes at g=0 when the neuronic decision means no, or at g=1 meaning yes. This set of Eq. (3a, b, c) describes an analog model of McCulloch-Pitts neurons. The original proof of convergence by Hopfield uses explicitly a quadratic energy expression among neurons for easy analog VLSI implementation. An independent proof has been given by Cohen and Grossberg that does not require the symmetry property of interconnects.

Each fine grained processor has been modeled in this paper by a different propagation speed governed by the first order equation:

$$(dU_i/dt_i) = -\,(\partial E_1(V)\,/\partial V_i), \qquad (4)$$

driven by a local energy gradient.

The collective answer should emerge at (dE/dt)=0 when the seemingly random computing without the lock-clock synchronizations. With respect to the collective time, the following macroscopically irreversibility: $(dE/dt) \leq 0$ will be guaranteed.

Theorem I: Asynchronous Convergence based on $(dE_1(V)/dt) \leq 0$.

If the neural network energy $E_1(V)$ depends only on the set **V** of all output firing rates V_i, and if and only if an arbitrary transfer function, $V_i=g(U_i)$ has a non-negative slope: $(dV_i/dU_i) \geq 0$, then the change of each neuron input U_i governed by its own time axis, through the first order dynamics: $(dU_i/dt_i) = -\,(\partial E_1(V)/\partial V_i)$ where $t_i = t_i^{(o)} + \varepsilon_i\, t$ with $\varepsilon_i > 0$, will guarantee the monotonic convergence $(dE_1/dt) \leq 0$.

Proof: The differential increment of in time must maintain the direction of the time flow, Eq. (3b) implying a positive characteristic factor,

$dt_i = \varepsilon_i\, dt$,or, $(dt_i/dt) = \varepsilon_i\ > 0$

The energy-gradient is so-to-speak the force upon the axonic output that changes the firing rate of the total dendritic input Eq. (1). Nonetheless, the global energy converges with respect to the collective or universal time t.

$$(dE_1(V_i)/dt) = \Sigma_i\, (\partial E_1/\partial V_i)\, (dV_i/dt_i)(dt_i/dt) \qquad (5a)$$
$$= -\Sigma_i\, \varepsilon_i\, (dU_i/dt_i)\ \ (dV_i/dt_i) \qquad (5b)$$
$$= -\Sigma_\iota\, \varepsilon_i\, (dU_i/dt_i)^2\, (dV_i/dU_i) \qquad (5c)$$
$$\leq 0\,. \qquad (5d)$$

Eq. (5a) is obtained by the chain rule of differentiation; in Eq. (5b), use is made of Newtonian Eq. (4) to eliminate the the energy slope ; Eq. (5c) is merely the identity $(dV_i/dt_i)=(dV_i/dU_i)(dU_i/dt_i)$ used to produce the second power of (dU_i/dt_i) in Eq. (5c). The last inequality Eq. (5d) is based on the mathematical truth that the square of arbitrary real number

$$(dU_i/dt_i)^2 = (\text{Real Numbers})^2 \geq 0$$

must be nonnegative in any time scale.

In the general convergence proof for arbitrary time axis t_i with $\varepsilon_i > 0$, we require no detail structure of the energy function, other than once differentiable. Thus, we have indeed verified the intuition that nothing changes $(dE_1/dt) = 0$ at the moment of convergence. This theorem may be called the first asynchronous neurocomputing principle that predicts the macroscopic irreversibility $(dE_1/dt) \leq 0$ from the microscopic reversible but time-asynchronous neurodynamics Eq. (4). The irreversibility is due to the necessary and sufficient condition Eq. (2) of the nonlinear transfer function g (that is equivalent to the stosszahl Ansatz of the binary collision transfer function in the Boltzmann Transport Equation). Although the proof similar to the Lyaponov theorem in the standard control theory, the learning mechanism in bio-control theory has been left unanswered.

(2) Rumelhart-like Weight-Adjustment Learning: Soft-Wired $E_2(W_{ij})$

Due to the biological inhomogeneity, the energy gradient descent methodology may be slightly generalized to a time-asynchronous learning algorithm that each neuron could have its own time axis

$$(dW_{ij}/dt_i) = -(\partial E_2(W_{ij})/\partial W_{ij}), \qquad (6a)$$
$$dt_i = \varepsilon_i\, dt\,, \qquad (6b)$$

Rumelhart, et al., has applied Eq. (6) to a feed forward and fixed layer architecture, within the synchronized layer of neurons: $\varepsilon_i = 1$. A slightly generalized convergence proof of time-asynchronous neurocomputing is given as follows:

Theorem II: Synaptic Adjustment Convergence:

$$(dE_2(W_{ij})/dt) = \Sigma_i\, (\partial E_2/\partial W_{ij})\, (dW_{ij}/dt) \qquad (7a)$$
$$= -\Sigma_i\, (dW_{ij}/dt_i)\, (dW_{ij}/dt_i)(dt_i/dt) \qquad (7b)$$
$$= -\Sigma_i\, \varepsilon_i\, (dW_{ij}/dt_i)^2 \qquad (7c)$$
$$\leq 0 \qquad (7d)$$

The adjustment of the synaptic weights W_{ij} can be derived implicitly in terms of the square error of the desired output **D** from the actual output **V**, when a given input **U** is fed into the layered network. Such a methodology is known as the **backward-error-propagation** resulting in a delta learning rule to assign the credit or the blame to other layered neurons behind them. To illustrate both energy functions $E_1(V_i)$ and $E_2(W_{ij})$, we assume

$$E_2(V_i(W_{ij})) = (1/2)\ \Sigma\, i\ (\,D_i\ -\ V_i)^{\,2} \qquad (8)$$

to be the square error of the desired response D_i from the actual output V_i, which, in terms of the analytical transfer function g of the input $U_i = \Sigma_j\, W_{ij}\, V'_j\ +\ \Theta_i$ Eq. (1), are the upward link synaptic weights. We denote the set of (input, actual output, desired output) respectively as $(U_i\,,\, V_i\,,\, D_i)$. If there is no error: $(V_i\ -\ D_i) = 0$, no learning takes place. The upward link weights W_{ij} are adjusted to reduce the difference, by multiplying the time-dependent factor ε_i to both hand sides of Eq. (6a).

$$\varepsilon_i(dW_{ij}/dt_i) \equiv \Delta W_{ij} = -\varepsilon_i(\partial E/\partial W_{ij})$$
$$= -\varepsilon_i\{(\partial E/\partial V_i)(dV_i/dU_i)\}(\partial U_i/\partial W_{ij})$$
$$= -\varepsilon_i\{(V_i - D_i)V_i(1 - V_i)\}\ V'_i \qquad (9a)$$

where the straightforward differentiation has produced the result.

The delta learning formula is the input energy change: $-\{(\partial E/\partial V_i)(dV_i/dU_i)\} = -(\partial E/\partial U_i) \equiv \delta_i$ with respect to the top layer input: $U_i=\Sigma_j W_{ij} V'_j + \Theta_i$ in terms of the upward synaptic links W_{ij}. Such an energy change at the top layer input is propagated downward to the the input energy change with respect to the hidden layer input: $U'_k = \Sigma_m W'_{km} V''_m + \Theta'_k$, in terms of the downward synaptic links W'_{kj}

$$\delta_i \equiv -(\partial E/\partial U_i) = -\Sigma_k (\partial E/\partial U'_k)(\partial U'_k/\partial U_i)$$
$$= \Sigma_k \delta'_k \Sigma_m(\partial U'_k/\partial V''_m)(\partial V''_m/\partial V_i)(dV_i/dU_i)$$
$$\cong (dV_i/dU_i)\ \Sigma_k \delta'_k W'_{ki} \qquad (9b)$$

where the approximation equality sign $\cong$ is due to the replacement of the unknown top layer input V_i with the known bottom layer input V''_m. Thus, the delta learning rule remains to be approximately independent of neuronic time axes.

$$\delta_j = V_j(1 - V_j)\ \Sigma_k \delta'_k W'_{kj} \qquad (9c)$$

(3) Morphology Convergence for Hairy Neurons with Brittle-wired $E3(V_i;T_{ij})$

In this section, we wish to formulate a set of neurodynamics equations which can settle itself into an appropriate network architecture, e.g. one layer of Hopfield, three layers of Rumelhart, or two layers of Grossberg. Neurophysiological experiments have recently shown that an active neuron can grow hairy neurofilaments, denoted as T_{ij}, in competing for nutritions and networking partnership against other neurons, and has been called a hairy neuron model(Szu 1989). The distinction between input synaptic weight W_{ik} from the output axonic neurofilament T_{ij} is necessary because of the recent neurophysiological experiments: (1) the use-it or lose-it synaptic pruning in one eye jack of a new born kitten, and (2) the actin protein generating the growth of neurofilaments. These neurofilament hair lines are competing for food and partnership. The winner grows fatter, while the loser shrinks thinner. The active growth of neurofilament T_{ij} reaches out and touches other neuron, and becomes eventually matured and retracts itself in forming a physical gap, the synaptic junction W_{ik}, for better resistive control of the ion diffusion potential without the initial direct contact. In order to take into account the possibility of the pruning of synapses W_{ik} (1), and the active growth of neurofilaments T_{ij} (2), the synaptic weights W_{ik} at the ith neuronic dendrite tree and kth junctions are assumed to be dormant variables, while the neurofilaments T_{ij} located at the ith axonic output can grow into the jth neuron with the active tread-mill microtube assembly mechanism. Thus, we have extended the classical McCulloch-Pitts neuron model, Eq. (2), to include one more degree of freedom, such as the synaptic transfer function

$$T_{ij}=f(W_{ij}) \qquad (10a)$$

between the axonic filaments T_{ij} (protein actin-driven for dynamic growing/pruning) and the dendrite synapses W_{ij} (positive ion-driven firing rates). The biological survival principle, use it or lose it, can be applied to the neuron level to explain the observed fact of a reduced synaptic gap density by a pruning mechanism in the one eye jack experiment on a new born kitten. In this experiment, a patch was place over the eye of a new born kitten. The post-natal development of its brain had no optical inputs and the optical processing neural networks died off leaving the kitten normal eye function blind. It will take a life long training to regain the binocular vision. The synaptic transfer function Eq. (7a) becomes, in the new born or high gain limit, a binary step function of the threshold b and the step size a.

$$f(x) = a\ \mathrm{step}(x-b) \qquad (10b)$$

in the first pass a blurred template which had a correct statistics of image pieces through the straightforward pointing-and-tracking summation of many frames (about 16 distorted fields) according to the centroid of the whole frame (Szu & Blodgett 1982) (c.f Distorted Fields, Object, Long Term Average, Centroid Correction). This effect had demonstrated the need of a smart sensor concept such as the eye which can see a weak star during an "instance of good seeing" (Szu et al. 1980) through the turbulent sky. On the contrary, the undiscriminating and dumb telescope camera can only produce a blurred picture of the weak star in the over exposed picture by the whole frame summation based on the straightforward pointing-and-tracking gimbal without any adaptive phase for turbulence medium phase correction mechanism.

Recently, a sequence of distorted imagery that consists of a training set of 15 samples of hand-written characters (each has 4 by 4 pixels, only trained to recognize 3 classes) has demonstrated the ability of generalization: recognize a new class of letter (Szu&Scheff 1989). This was done by means of critical feature extraction using the "mini-max concept" to discover by itself a new class of 5 more hand-written characters by analyzing the "intra-interclass clustering property" on the self-constructed feature space (c.f. Fig. 6 for 20 samples of 4 classes). This example used a table top computer, because the Gram-Schmidt orthogonal feature extraction was based on the associative memory employing the Fixed-Point Cycle Two Theorem (Szu, Scheff 1989). Such a procedure of parallel Gram-Schmidt constrained orthogonalization could be exceedingly usefully for a covert communication constrained by call signs and known scrambling instruction, because feature extraction by means of the straightforward projection is not permitted to obliterate critical portion of the signal. However, any practical construction of large set of orthogonal feature vectors could be subject to a realtime processing bottleneck. In this paper, the Fast Simulated Annealing (FSA) technique is adopted to alleviate the bottleneck problem.

Image processing by annealing techniques have been attempted (Geman & Geman,1984) (Smith et al. 1983) mainly for noise/distortion reduction. Neural networks have been recently applied to pattern recognition by Kohonen, Fukushima, Grossberg, Hopfield, etc.. White noise annealing and neural networks are combined through the Boltzmann Machine (Hinton, Sejnowski, Ackley, 1984) of which colored noise variant has been referred to as Cauchy Machine (Szu 1987) (Scheff &Szu 1987) (Takefuji & Szu 1989)

SPATIO-TEMPORAL IMAGERIES

A useful clutter rejection hypothesis is that man-made vehicles are designed to minimize the hydrodynamic drag via streamlined shapes and wheels while the natural environment of tree trunks is mainly vertical against the gravity (unpublished work of J. Landa, H.Szu). Thus, a sequence of imagery of land vehicles passing by bushes is considered, Fig. 7 (a). When a land vehicle moves by a tree, the partial occlusion of the vehicle by the tree trunk can be easily overcome by a properly pointing tracking, zooming, imaging on the moving vehicle. The image sequence can be averaged and threshold to get rid of the relative motion between the tree and the vehicle, Fig. 7 (b), together with the 9 by 9 scanning Peano curve. The centroid pointing and tracking of the vehicle is assumed to produce the averaged gray-scaled image $< I_c(x,y) >$

$$< I_c(x,y) > = \Sigma_j \; I_{j}(x+x_c,y+y_c)/ \text{ frames} \qquad (13)$$

where (x_c, y_c) is a vehicle local centroid coordinate. After a certain threshold, the obscuring effect of the tree and bush will be minimized. Fig.7 (describe the templates)

$$L_c(x,y) = \text{Threshold}(< I_c(x,y) >) \qquad (14)$$

Let the critical feature of the template class-c be denoted as $f_c(x,y)$. Then, the performance criterion is the minimum distance between the template of the c-class=1,2 together with the direction cosine in the numerator, and the maximum difference between feature vectors in the denominator. Thus, the mini-max filter energy is

$$E(f_c)= a \, \Sigma_{c \neq c'} (< f_c | f_{c'} >) \; + b \, \Sigma_{c=1,2,...} \, | f_c - \underline{I}_c |^2 + \Sigma_{c \neq c'} \; d \, / \, | f_c - f_{c'} |^2 \qquad (15)$$

where the coefficient of the direction cosine via the inner product < | > may be heavily weighted, e.g. by setting a = 10 (relative to b = 1, c=1, and d=10). The change of energy is defined as $\Delta E = E_{new} - E_{old}$.

CAUCHY MACHINE

The image space is 2-D; but the search space can be 1-D, provided that space-filling scanning technique is adopted here for mapping 2-D imagery space to 1-D search space and yet preserving the local neighborhood

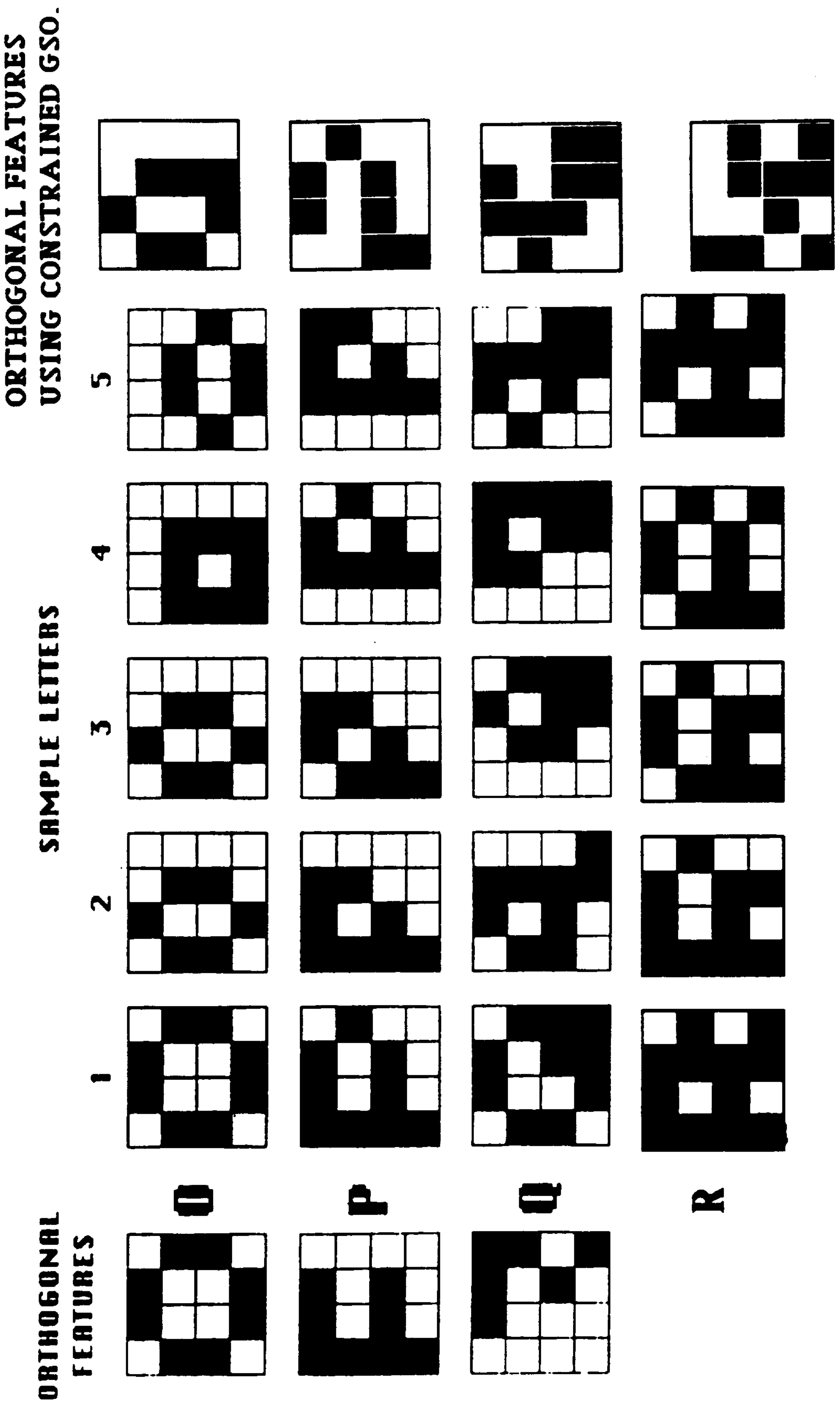


Figure 6. Hand-written character recognition by orthogonal feature extraction using constrained Gram-Schmidt orthogonalization (GSO) procedure.

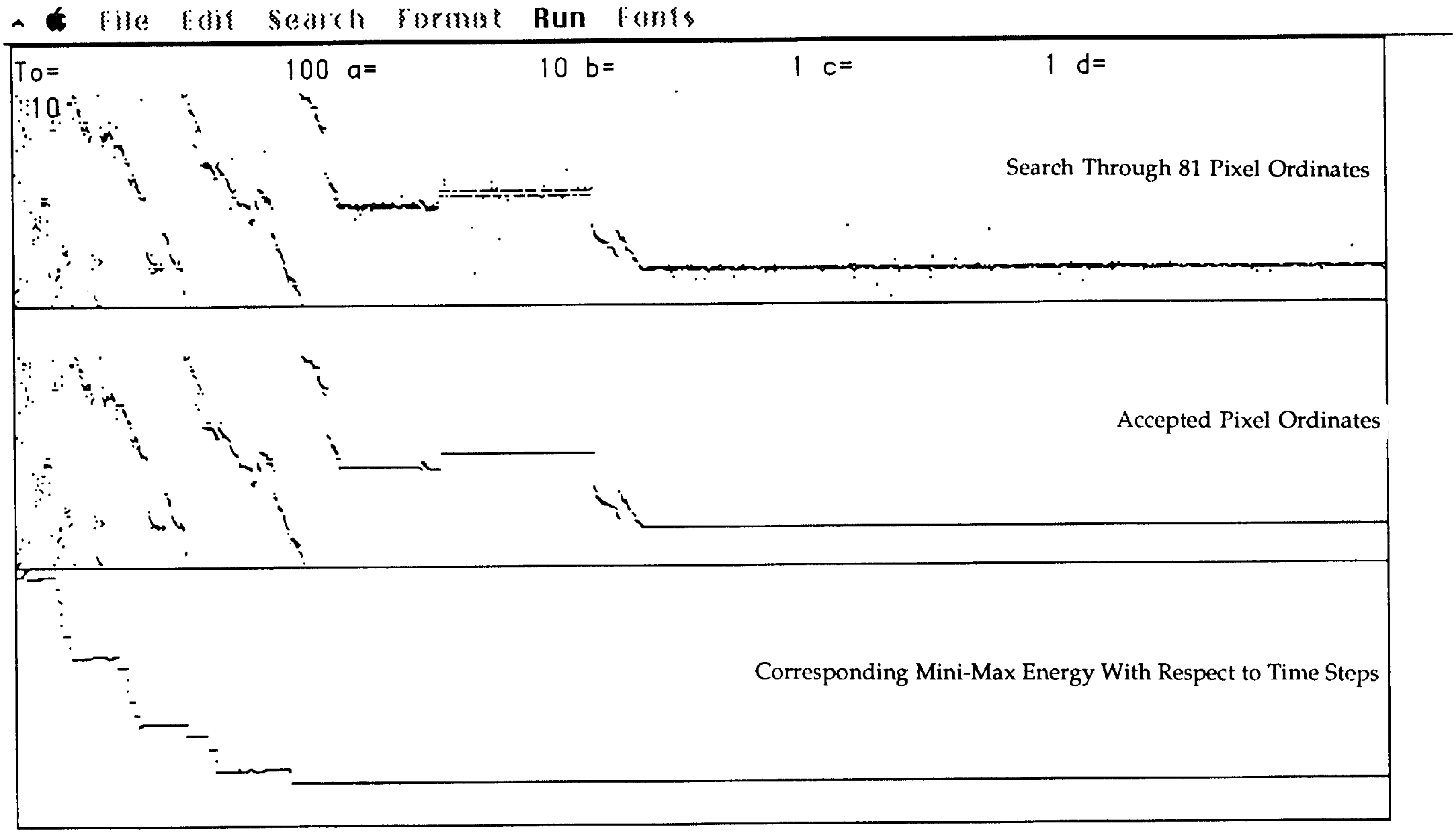

File
Edit
Search
Format
Run
Fonts
To=
10
100 a=
10 b=
1 c=
1 d=
Search Through 81 Pixel Ordinates
Accepted Pixel Ordinates
Corresponding Mini-Max Energy With Respect to Time Steps

Figure 8. Cauchy simulated annealing search for the mini-max global minimum energy. Three segments of the ordinate (top segment: searching 9x9 states, middle segment: accepted 9x9 states, and bottom segment: the energy of the visited state) are plotted with respect to the abscissa of 2000 time points in three minutes CPU time on a Macintosh II.

Hinton, G.E., Sejnowski, T.J., & Ackley, D.H. (1984). Boltzmann Machines: Constrained Satisfaction Networks that Learn. *CMU-CS-84-119, Carnegie Mellon Univ. May, 1984. "Parallel Distributed Processing, Vol. I , Vol.II Edited by J. McCelland, D. Rumelhart, PDP Group, MIT Press, 1986*

Hopfield, J.J. (1982). Neural Networks and Physical Systems with Emergent Collective Properties Like those of Two-State Neurons. Proc. Natl. Acad. Sci. USA Vol.79, pp. 2554-2558.

Kohonen, T (1984). Self-Organization and Associative Memory. Springer-Verlag, Berlin

McCulloch W.S. & Pitts, W. (1943). A logical Calculus of the Ideas Imminent in Nervous Activity. Bulletin of Mathematical Biophysics, 5, pp 115-133

Scheff, K., & Szu, H. (1987). 1-D Optical Cauchy Machine Infinite Film Spectrum Search. *Int. Conf.Neural Networks-87, P. III-673, San Diego*

Smith, W.E., Barrett, H.H., & Paxman, R.G. (1983). Reconstruction of objects from coded images by simulated annealing. *Optics Letters, Vol.8, pp 199-201*

Szu, H., Blodgett, J., & Sica, L. (1980). Local Instances of Good Seeing. *Optical Comm., Vol. 35, pp. 317- 322*

Szu, H., & Blodgett, J (1982). Self-reference Spatiotemporal Image-Restoration Technique.*J.Opt.Soc.Am., Vol.72, pp.1666-1669*

Szu, H., & Messner, R. (1986). Adaptive Invariant Novelty Filters. *Proc. IEEE, V.74, p.519*

Szu, H.,& Scheff, K. (1989). Gram-Schmidt Orthogonalization Neural Nets for Optical Character Recognition. *Int Joint Conference on Neural Networks, Vol. I, pp. 547-555, Washington D.C., June 18-22*

Szu, H. (1987). Fast Simulated Annealing. *In: "Neural Networks for Computing," AIP Conf. Vol. 15, pp. 420-425, Edited by J. Denker, Snow Bird U.T., 1987; Also, Phys. Letters A 122,p.157, Jun 8, 1987; Proc.IEEE, V. 75, p.1538.*

Tarkefuji, Y., & Szu,H. (1989). Parallel Distributed Cauchy Machine. *Int.Joint Conf. Neural Networks-89, p. I-529, Washington D.C. June 18-22*

Szu, H. (1989). Reconfigurable neural nets by Energy Convergence Learning Principle based on extended McCulloch and Pitts Neurons and Synapses. *Int.Joint Conf. Neural Networks-89, p. I-485, Washington D.C. June 18-22*

Szu, H. & Scheff, K (1990). Simulated Annealing Feature Extraction from Occluded and Cluttered Objects. *Int. Joint Conf.Neural Networks-90, p. II-76, Washington D.C. Jan.15-18,* Problem. *Int. Joint Conf.Neural Networks-90, p. I-317, Washington D.C. Jan.15-18*

Appendix A : Fast Simulated Annealing Algorithm (TRUE_BASIC Version)

```
DATA 4,5,8,9,11,14,15,16,17,38,41,44,46,47,50,51,52,53,56,5758,59,67,69,70,71,72,78,79      !input 81 Peano-scanning pixel#
DATA 4,5,8,9,12,13,14,15,16,17,30,31,37,42,43,46,47,50,51,52,53,56,57,58,59,62,63,69,70    !1= black feature Eq. (13)
DIM f1(81),f2(81),ave1(81),ave2(81),ft1(81),ft2(81)
MAT ave2 =0                                          ! True_Basic  Matrix Operation
FOR n=1 to 29                                        ! read an object into ave1, namely I1, Eq. (13)
  READ k
  LET ave1(k)=1
NEXT n
FOR m = 30 to 58                                     ! read another object  into ave2, namely I2, Eq. (13)
  READ J
  LET ave2(J)=1
NEXT m
RANDOM                                               ! random number rnd generated [0,1]
FOR t=1 to tmax                                      ! after initialize the display
  LET temp=To/(1+t)                                  ! Fast Simulated Annealing cooling schedule
  LET theta=(rnd-.5)*Pi                              ! uniform theta using the radian angle option
   LET dx=int(temp*tan(theta))                       ! new pixel by T tan(theta), Eq. (15)
  LET xnew=mod(x+dx,82)                              ! module for 81 scan pixels
  IF xnew=0 then LET xnew=81
  IF f2(xnew)=0 THEN
    LET ft2(xnew)=ave2(xnew)
    LET ft1(xnew)=0
  ELSE
    LET ft2(xnew)=0
    LET ft1(xnew)=ave1(xnew)
  END IF
  LET enew= 0
  LET denominator=0
  LET ef1=0
  LET ef2=0
  FOR n=1 to 81
     LET ef1=ef1+(ft1(n)-ave1(n))*(ft1(n)-ave1(n))
     LET ef2=ef2+(ft2(n)-ave2(n))*(ft2(n)-ave2(n))
      LET denominator=denominator+(ft1(n)-ft2(n))*(ft1(n)-ft2(n))
     LET enew = enew + ft1(n)*ft2(n)
  NEXT n
  LET enew= a*enew + b*ef1 + c*ef2 + (d/denominator)     ! constants are typed into the code at run time
  IF enew<eold then
    MAT f2=ft2
    MAT f1=ft1
    LET eold=enew
    LET x=xnew
  END IF
  IF enew>=eold then
    IF (rnd*0.5)<(1/(1+exp((enew-eold)/temp))) then     !hill climbing Eq. (16)
      MAT f2=ft2
      MAT f1=ft1
      LET eold=enew
      LET x=xnew
    END IF
  END IF
  PLOT POINTS :t,xnew+200
  PLOT POINTS :t,x+100
  PLOT POINTS :t,eold/2
 NEXT t
```