14 Competitive Networks

14.1 Hamming Network

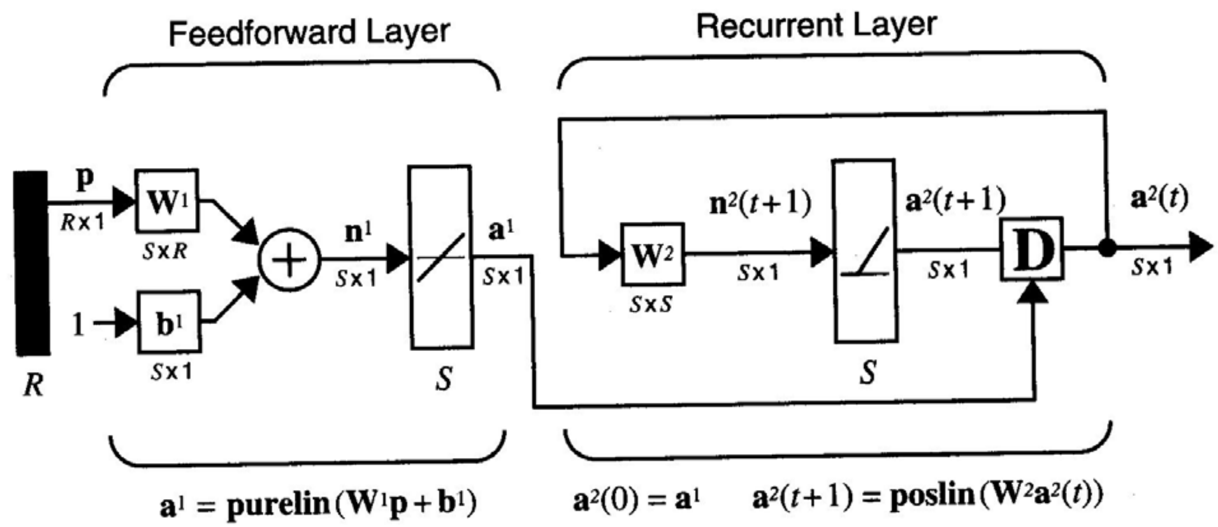


Figure 14.1-1 Hamming Network

- The first layer (which is a layer of instars) performs a correlation between the input vector and the prototype vectors.
- The second layer performs a competition to determine which of the prototype vectors is closest to the input vector.

Layer 1

- Multiple instars: Multiple pattern recognition

Q Prototype vectors, *R*: Number of input

$$\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_Q\} \tag{14.1-1}$$

$$\mathbf{W}^1 = \begin{bmatrix} {}_1\mathbf{w}^T \\ {}_2\mathbf{w}^T \\ \vdots \\ {}_S\mathbf{w}^T \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \vdots \\ \mathbf{p}_Q^T \end{bmatrix}, \mathbf{b}^1 = \begin{bmatrix} R \\ R \\ \vdots \\ R \end{bmatrix} \tag{14.1-2}$$

$$\mathbf{a}^1 = \mathbf{W}^1\mathbf{p} + \mathbf{b}^1 = \begin{bmatrix} \mathbf{p}_1^T\mathbf{p} + R \\ \mathbf{p}_2^T\mathbf{p} + R \\ \vdots \\ \mathbf{p}_Q^T\mathbf{p} + R \end{bmatrix} \tag{14.1-3}$$

Layer 2

- Layer 2: a competitive layer
- Initialized with the outputs of the feedforward layer
- Finally, only one neuron with nonzero output: winning neuron: recognized pattern: a winner-take-all competition

$$\mathbf{a}^2(0) = \mathbf{a}^1 \quad (14.1\text{-}4)$$

$$\mathbf{a}^2(t+1) = \mathbf{poslin}(\mathbf{W}^2\mathbf{a}^2(t)) \quad (14.1\text{-}5)$$

$$w_{ij}^2 = \begin{cases} 1, & if\,(i=j) \\ -\varepsilon, & otherwise \end{cases}, \text{ where } 0 < \varepsilon < \frac{1}{S-1} \quad (14.1\text{-}6)$$

$$\mathbf{W} = \begin{bmatrix} 1 & -\varepsilon & \cdots & -\varepsilon \\ -\varepsilon & 1 & \cdots & -\varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ -\varepsilon & -\varepsilon & \cdots & 1 \end{bmatrix} \quad (14.1\text{-}7)$$

$$a_i^2(t+1) = poslin\left(a_i^2(t) - \varepsilon \sum_{j \neq i} a_j^2(t) \right) \quad (14.1\text{-}7)$$

- The output of the neuron with the largest initial condition decreases more slowly than the outputs of the other neurons.

14.2 Competitive Layer

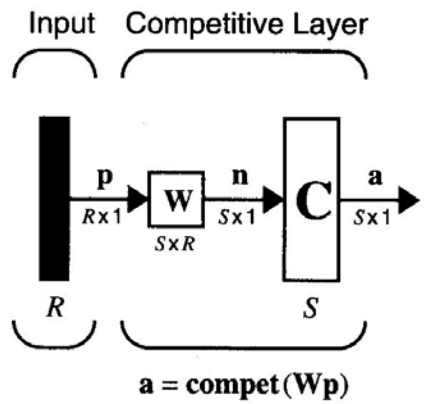


Figure 14.2-1 Competitive Layer

$$\mathbf{a} = \mathbf{compet}(\mathbf{n}) = \mathbf{compet}(\mathbf{Wp}) \quad (14.2\text{-}1)$$

$$a_i = \begin{cases} 1, i = i^* \\ 0, i \neq i^* \end{cases}, \text{ where } n_{i^*} \geq n_i, \forall i, \text{ and } i^* \leq i, \forall n_i = n_{i^*} \quad (14.2\text{-}2)$$

$$\mathbf{n} = \mathbf{Wp} = \begin{bmatrix} {}_1\mathbf{w}^T \\ {}_2\mathbf{w}^T \\ \vdots \\ {}_S\mathbf{w}^T \end{bmatrix} \mathbf{p} = \begin{bmatrix} {}_1\mathbf{w}^T\mathbf{p} \\ {}_2\mathbf{w}^T\mathbf{p} \\ \vdots \\ {}_S\mathbf{w}^T\mathbf{p} \end{bmatrix} = \begin{bmatrix} L^2 \cos\theta_1 \\ L^2 \cos\theta_2 \\ \vdots \\ L^2 \cos\theta_S \end{bmatrix} \quad (14.2\text{-}3)$$

14.2.1 Competitive Learning

Instar rule,

$$ {}_i\mathbf{w}(q) = {}_i\mathbf{w}(q-1) + \alpha a_i(q)(\mathbf{p}(q) - {}_i\mathbf{w}(q-1)) \quad (14.2.1\text{-}1)$$

Kohonen rule,

$$ {}_i\mathbf{w}(q) = {}_i\mathbf{w}(q-1) + \alpha(\mathbf{p}(q) - {}_i\mathbf{w}(q-1)) = (1-\alpha)\,{}_i\mathbf{w}(q-1) + \alpha\mathbf{p}(q) \quad (14.2.1\text{-}2)$$

$$ {}_i\mathbf{w}(q) = {}_i\mathbf{w}(q-1), \quad i \neq i^* \quad (14.2.1\text{-}3)$$

- The row of the weight matrix that is closest to the input vector moves toward the input vector.
- It moves along a line between the old row of the weight matrix and the input vector.

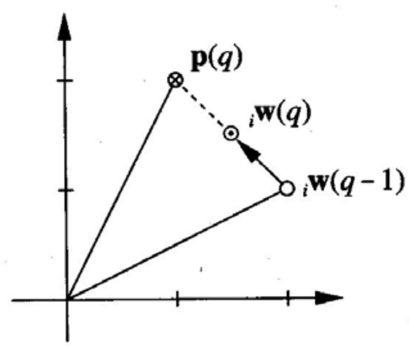


Figure 14.2.1-1 Graphical Representation of the Kohonen Rule

$$\mathbf{p}_1 = \begin{bmatrix} -0.1961 \\ 0.9806 \end{bmatrix}, \mathbf{p}_2 = \begin{bmatrix} 0.1961 \\ 0.9806 \end{bmatrix}, \mathbf{p}_3 = \begin{bmatrix} 0.9806 \\ 0.1961 \end{bmatrix}, \mathbf{p}_4 = \begin{bmatrix} 0.9806 \\ -0.1961 \end{bmatrix}, \mathbf{p}_5 = \begin{bmatrix} -0.5812 \\ -0.8137 \end{bmatrix}, \mathbf{p}_6 = \begin{bmatrix} -0.8137 \\ -0.5812 \end{bmatrix} \quad (14.2.1\text{-}4)$$

$$_1\mathbf{w} = \begin{bmatrix} 0.7071 \\ -0.7071 \end{bmatrix}, {}_2\mathbf{w} = \begin{bmatrix} 0.7071 \\ 0.7071 \end{bmatrix}, {}_3\mathbf{w} = \begin{bmatrix} -1.0000 \\ 0.0000 \end{bmatrix}, \mathbf{W} = \begin{bmatrix} {}_1\mathbf{w} \\ {}_2\mathbf{w} \\ {}_3\mathbf{w} \end{bmatrix} \quad (14.2.1\text{-}5)$$

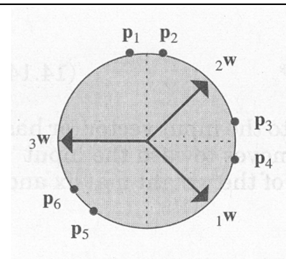


Figure 14.2.1-2 Input and Initial Random Pattern Vectors

Presenting $\mathbf{p}_2$,

$$\mathbf{a} = \mathbf{compet}(\mathbf{W}\mathbf{p}_2) = \mathbf{compet}\left(\begin{bmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \\ -1.0000 & 0.0000 \end{bmatrix} \begin{bmatrix} 0.1961 \\ 0.9806 \end{bmatrix} \right) = \mathbf{compet}\left(\begin{bmatrix} -0.5547 \\ 0.8321 \\ -0.1961 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (14.2.1\text{-}6)$$

$\alpha = 0.5$,

$$ {}_2\mathbf{w}^{new} = {}_2\mathbf{w}^{old} + \alpha(\mathbf{p}_2 - {}_2\mathbf{w}^{old}) = \begin{bmatrix} 0.7071 \\ 0.7071 \end{bmatrix} + 0.5\left(\begin{bmatrix} 0.1961 \\ 0.9806 \end{bmatrix} - \begin{bmatrix} 0.7071 \\ 0.7071 \end{bmatrix}\right) = \begin{bmatrix} 0.4516 \\ 0.8438 \end{bmatrix} \quad (14.2.1\text{-}7)$$

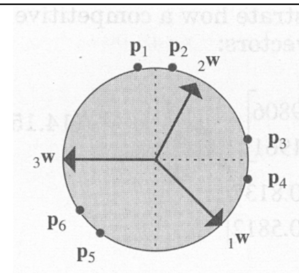


Figure 14.2.1-3 Weight Modification

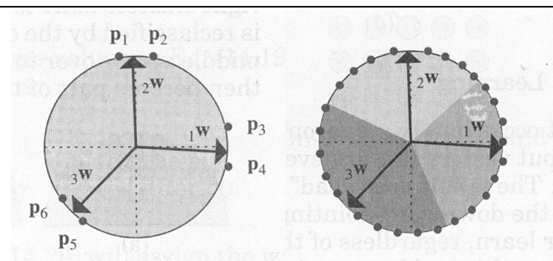


Figure 14.2.1-4 Final Weight Modification and Region Classified by the Final Weight Modification

14.2.2 Problems with Competitive Layers

- The choice of learning rate forces a trade-off between the speed of learning and the stability of the final weight vectors.
- When clusters are close together, a weight vector forming a prototype of one cluster may "invade" the territory of another weight vector.
- A neuron's initial weight vector is located so far from any input vectors that it never wins the competition.
- A competitive layer must have as many classes as it has neurons. When the number of clusters is not known in advance.
- Competitive layers cannot form classes with nonconvex regions or classes that are the union of unconnected regions.

14.3 Self Organizing Feature Maps

Kohonen rule,

$$ {}_i\mathbf{w}(q) = {}_i\mathbf{w}(q-1) + \alpha(\mathbf{p}(q) - {}_i\mathbf{w}(q-1)) = (1-\alpha)\,{}_i\mathbf{w}(q-1) + \alpha\mathbf{p}(q) \quad i \in N_{i^*}(d) \tag{14.3-1} $$

Neighborhood,

$$ N_i(d) = \{ j, d_{ij} \le d \} \tag{14.3-2} $$

- When a vector **p** is presented, the weights of the winning neuron and its neighbors will move toward **p**.

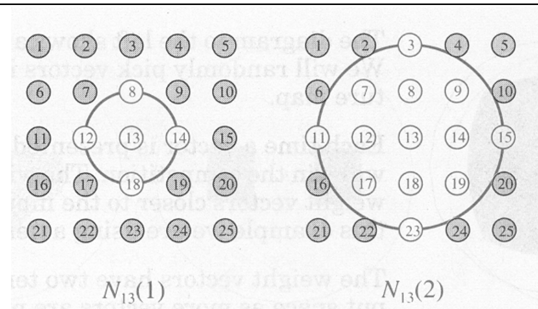


Figure 14.3-1 Neighborhoods

$$ N_{13}(1) = \{8,12,13,14,18\} \tag{14.3-3} $$

$$ N_{13}(2) = \{3,7,8,9,11,12,13,14,15,17,18,19,23\} \tag{14.3-4} $$

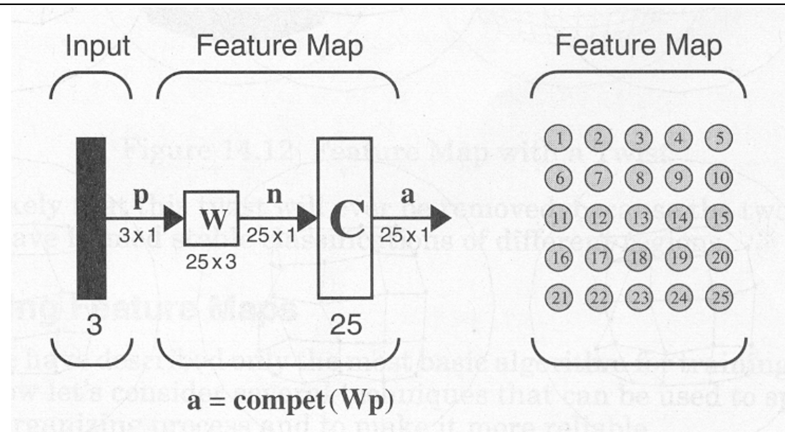


Figure 14.3-2 Self-Organizing Feature Map

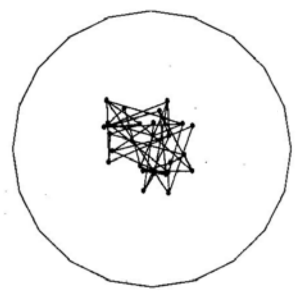


Figure 14.3-3 Initial Weight Vectors

- Each three-element weight vector is represented by a dot on the sphere.
- The weights are normalized, therefore they are on the surface of a sphere.
- Dots of neighboring neurons are connected by lines to see how the physical topology of the network is arranged in the input space.
- Each time a vector is presented, the neuron with the closest weight vector wins the competition.
- The winning neuron and its neighbors move their weight vectors closer to the input vector (and therefore to each other).
- The weight vectors spread out over the input space as more vectors are presented.
- The weight vectors move toward the weight vectors of neighboring neurons.

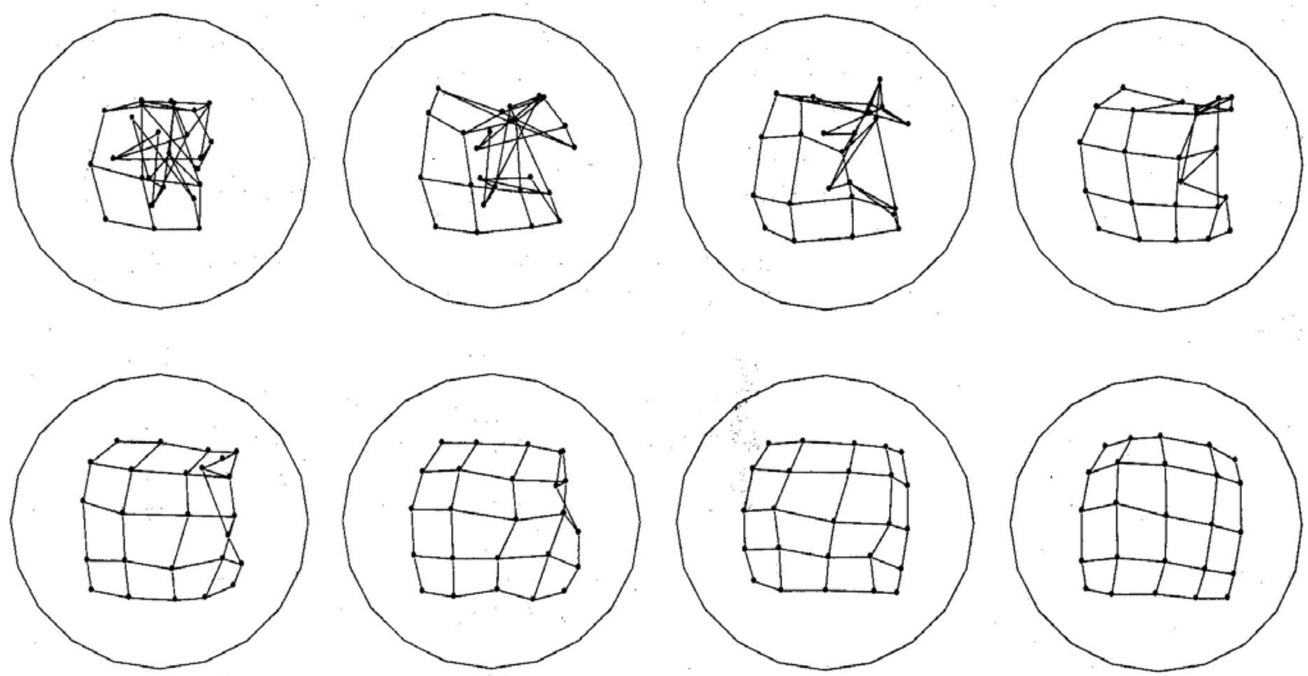


Figure 14.3-4 Self-Organization, 250 Iterations per Diagram

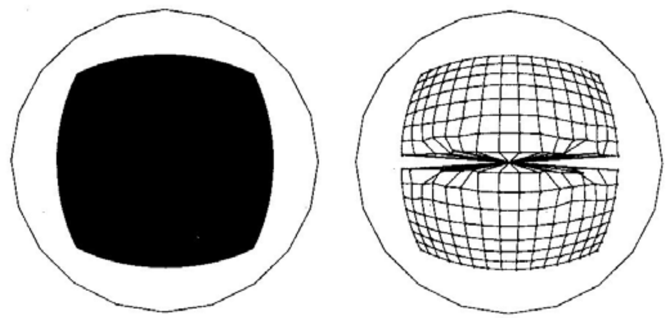


Figure 14.3-5 Feature Map with a Twist

14.3.1 Improving Feature Maps

- Varying the size of the neighborhoods during training.
- Varying the learning rate over time.
- Using larger learning rate for the winning neuron than the neighboring neurons
- Using distance between the input vector and the prototype vectors instead of the inner product

14.4 Learning Vector Quantization

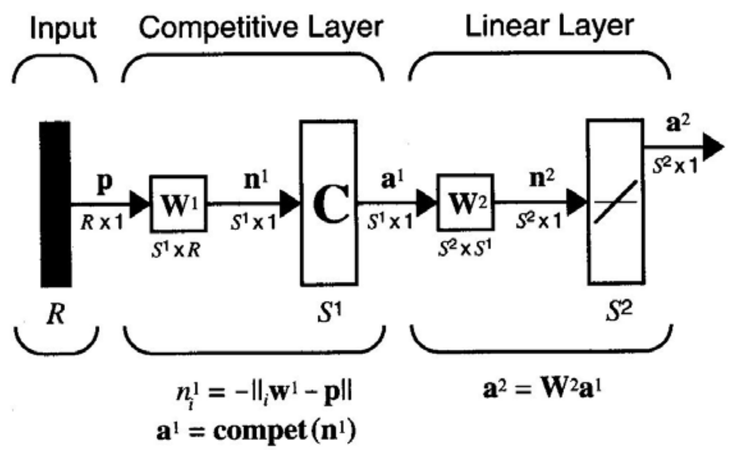


Figure 14.4-1 LVQ Network

$$n_i^1 = -\left\| {}_i\mathbf{w}^1 - \mathbf{p} \right\| \tag{14.4-1}$$

$$\mathbf{n}^1 = -\begin{bmatrix} \left\| {}_1\mathbf{w}^1 - \mathbf{p} \right\| \\ \left\| {}_2\mathbf{w}^1 - \mathbf{p} \right\| \\ \vdots \\ \left\| {}_{S^1}\mathbf{w}^1 - \mathbf{p} \right\| \end{bmatrix} \tag{14.4-2}$$

$$\mathbf{a}^1 = \mathbf{compet}(\mathbf{n}^1) \tag{14.4-3}$$

- In the first layer, the winning neuron indicates a subclass.
- The second layer is used to combine subclasses into a single class.

$$(w_{ki}^2 = 1) \Rightarrow \text{ subclass } i \text{ is a part of class } k. \quad (14.4\text{-}4)$$

- The columns of $\mathbf{W}^2$ represent subclasses, the rows represent classes.

14.4.1 LVQ Learning

Supervised learning,

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \ldots, \{\mathbf{p}_Q, \mathbf{t}_Q\} \quad (14.4.1\text{-}1)$$

If hidden neuron i is to be assigned to class k, then set $w_{ki}^2 = 1$ (14.4.1-2)

- Once $\mathbf{W}^2$ is defined, it will never be altered.
- The hidden weights $\mathbf{W}^1$ are trained with a variation of the Kohonen rule.

$${}_{i^*}\mathbf{w}^1(q) = {}_{i^*}\mathbf{w}^1(q-1) + \alpha(\mathbf{p}(q) - {}_{i^*}\mathbf{w}^1(q-1)), \text{ if } a_{k^*}^2 = t_{k^*} = 1 \quad (14.4.1\text{-}3)$$

$${}_{i^*}\mathbf{w}^1(q) = {}_{i^*}\mathbf{w}^1(q-1) - \alpha(\mathbf{p}(q) - {}_{i^*}\mathbf{w}^1(q-1)), \text{ if } a_{k^*}^2 = 1 \neq t_{k^*} = 0 \quad (14.4.1\text{-}4)$$

Example:

$$\text{Class 1:}\left\{\mathbf{p}_1=\begin{bmatrix}-1\\-1\end{bmatrix},\mathbf{p}_2=\begin{bmatrix}1\\1\end{bmatrix}\right\},\quad \text{class 2:}\left\{\mathbf{p}_3=\begin{bmatrix}1\\-1\end{bmatrix},\mathbf{p}_4=\begin{bmatrix}-1\\1\end{bmatrix}\right\} \tag{14.4.1-5}$$

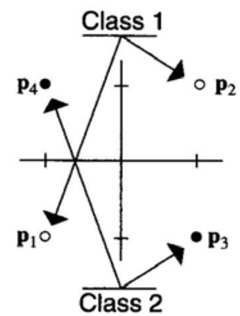


Figure 14.4.1-1 Members of Classes

$$\left\{\mathbf{p}_1=\begin{bmatrix}-1\\-1\end{bmatrix},\mathbf{t}_1=\begin{bmatrix}1\\0\end{bmatrix}\right\},\left\{\mathbf{p}_2=\begin{bmatrix}1\\1\end{bmatrix},\mathbf{t}_2=\begin{bmatrix}1\\0\end{bmatrix}\right\},\left\{\mathbf{p}_3=\begin{bmatrix}1\\-1\end{bmatrix},\mathbf{t}_3=\begin{bmatrix}0\\1\end{bmatrix}\right\},\left\{\mathbf{p}_4=\begin{bmatrix}-1\\1\end{bmatrix},\mathbf{t}_4=\begin{bmatrix}0\\1\end{bmatrix}\right\} \tag{14.4.1-7}$$

$$\mathbf{W}^2=\begin{bmatrix}1&1&0&0\\0&0&1&1\end{bmatrix} \tag{14.4.1-8}$$

$${}_1\mathbf{w}^1=\begin{bmatrix}-0.543\\0.840\end{bmatrix},{}_2\mathbf{w}^1=\begin{bmatrix}-0.969\\-0.249\end{bmatrix},{}_3\mathbf{w}^1=\begin{bmatrix}0.997\\0.094\end{bmatrix},{}_4\mathbf{w}^1=\begin{bmatrix}0.456\\0.954\end{bmatrix} \tag{14.4.1-9}$$

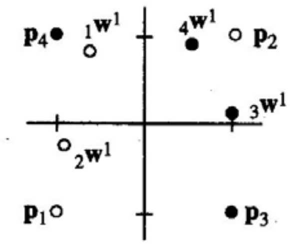


Figure 14.4.1-2 Initial Weight

Presenting $\mathbf{p}_3$,

$$\mathbf{a}^1 = \mathbf{compet}(\mathbf{n}^1) = \mathbf{compet}\left(\begin{bmatrix} -\|{}_1\mathbf{w}^1 - \mathbf{p}_3\| \\ -\|{}_2\mathbf{w}^1 - \mathbf{p}_3\| \\ -\|{}_3\mathbf{w}^1 - \mathbf{p}_3\| \\ -\|{}_4\mathbf{w}^1 - \mathbf{p}_3\| \end{bmatrix}\right) = \mathbf{compet}\left(\begin{bmatrix} -2.40 \\ -2.11 \\ -1.09 \\ -2.03 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{14.4.1-10}$$

$$\mathbf{a}^2 = \mathbf{W}^2\mathbf{a}^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{14.4.1-11}$$

$${}_3\mathbf{w}^1(1) = {}_3\mathbf{w}^1(0) + \alpha(\mathbf{p}_3 - {}_3\mathbf{w}^1(0)) = \begin{bmatrix} 0.997 \\ 0.094 \end{bmatrix} + 0.5\left(\begin{bmatrix} 1 \\ -1 \end{bmatrix} - \begin{bmatrix} 0.997 \\ 0.094 \end{bmatrix}\right) = \begin{bmatrix} 0.998 \\ -0.453 \end{bmatrix} \tag{14.4.1-12}$$

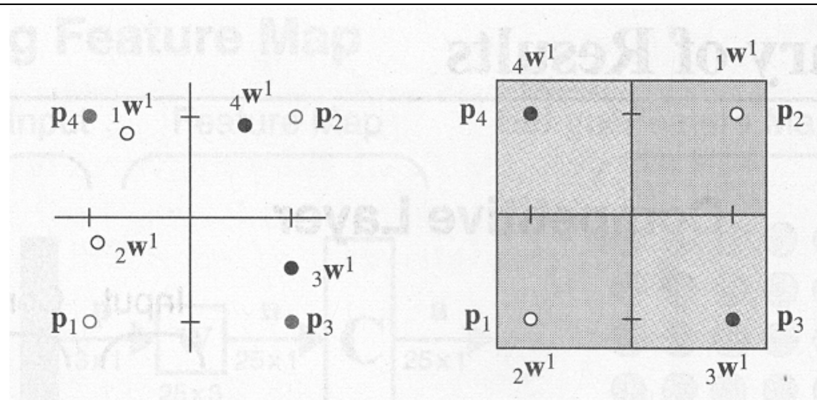


Figure 14.4.1-3 After First and Many Iterations

15 Grossberg Network

15.1 Basic Nonlinear Model

Leaky integrator,

$$\varepsilon \frac{dn(t)}{dt} = -n(t) + p(t) \qquad (15.1\text{-}1)$$

where ε: the system time constant,

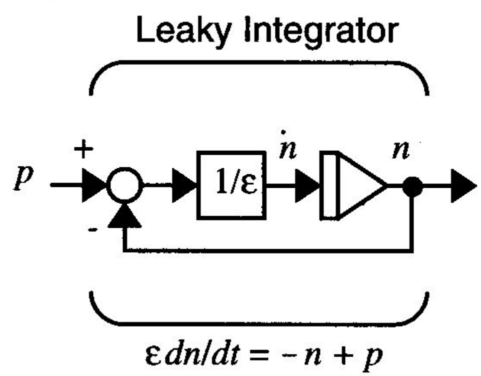


Figure 15.1-1 Leaky Integrator

The response of the leaky integrator to an arbitrary input $p(t)$,

$$n(t) = e^{-t/\varepsilon} n(0) + \frac{1}{\varepsilon}\int_0^t e^{-(t-\tau)/\varepsilon} p(t-\tau)\, d\tau \qquad (15.1\text{-}2)$$

The input $p(t)$: constant and the initial condition $n(0)$ = zero,

$$n(t) = p(1 - e^{-t/\varepsilon}) \qquad (15.1\text{-}3)$$

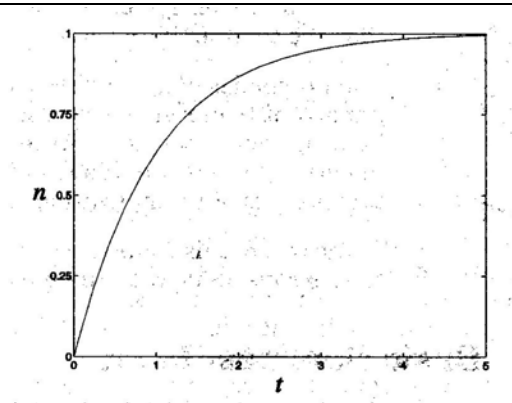


Figure 15.1-2 Leaky Integrator Response, p = 1 and ε = 1

- If the input p is scaled, then the response $n(t)$ will be scaled by the same amount.
- The speed of response of the leaky integrator is determined by the time constant ε.

Shunting model,

$$\varepsilon \frac{dn(t)}{dt} = -n(t) + (b^{+} - n(t))p^{+} - (n(t) + b^{-})p^{-} \qquad (15.1\text{-}4)$$

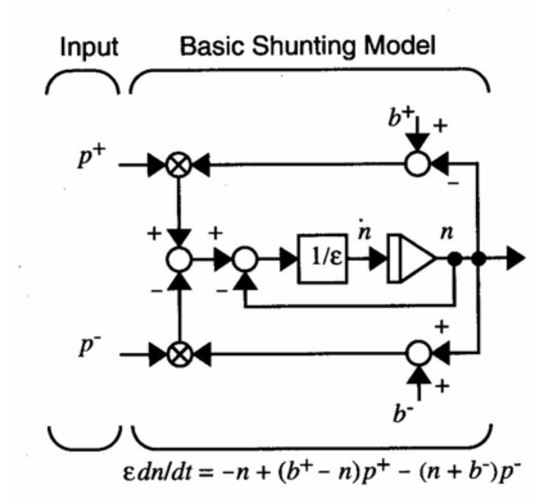


Figure 15.1-3 Shunting Model

- p^+: the excitatory input, a nonnegative value
- p^-: the inhibitory input, a nonnegative value
- b^+ and b^-: the upper and lower limits on the neuron response, nonnegative constants

Shunting model,

- The first term, $-n(t)$, a linear decay term
- The second term, $(b^+ - n(t))p^+$, nonlinear gain control, used to set an upper limit on $n(t)$ of b^+
- The third term, $-(n(t) + b^-)p^-$, nonlinear gain control. Used to set a lower limit on $n(t)$ of b^-

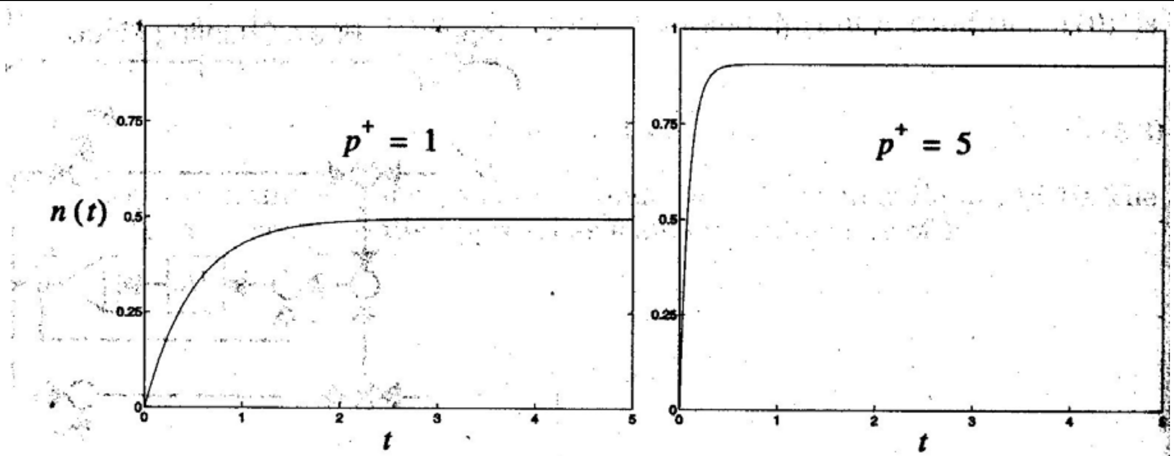


Figure 15.1-4 Shunting Network Response

- If $n(0)$ falls between b^+ and $-b^-$, then $n(t)$ will remain between these limits.

15.2 Two-Layer Competitive Network

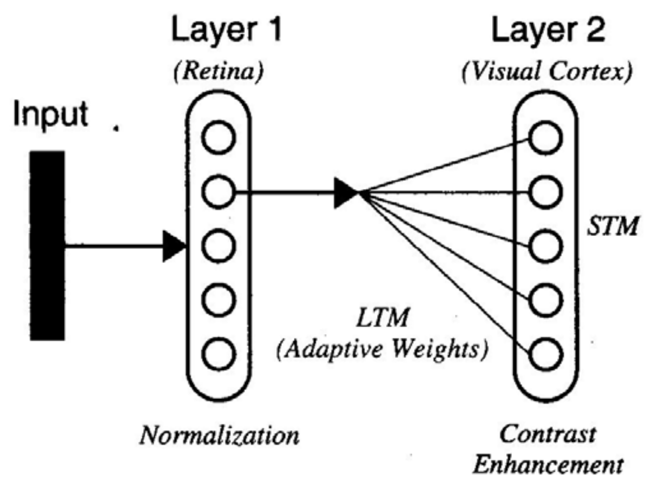


Figure 15.2-1 Grossberg Competitive Network

Grossberg network:

1. Layer 1
2. Layer 2
3. Adaptive weights

Layer 1

- Layer 1 of the Grossberg network receives external inputs and normalizes the intensity of the input pattern.

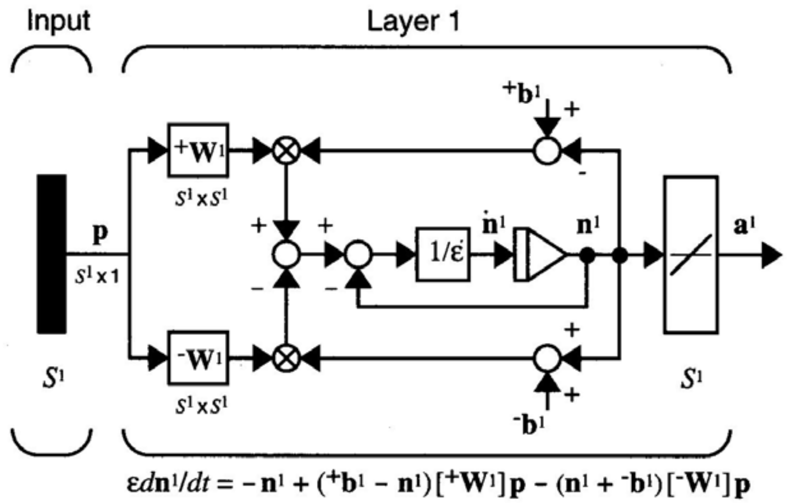


Figure 15.2-2 Layer 1 of Grossberg Network

$$\varepsilon\frac{d\mathbf{n}^1(t)}{dt} = -\mathbf{n}^1(t) + ({}^{+}\mathbf{b}^1 - \mathbf{n}^1(t))[{}^{+}\mathbf{W}^1]\mathbf{p} - (\mathbf{n}^1(t) + {}^{-}\mathbf{b}^1)[{}^{-}\mathbf{W}^1]\mathbf{p} \qquad (15.2\text{-}1)$$

- The parameter ε determines the speed of response. It is chosen so that the neuron responses will be much faster than the changes in the adaptive weights.

The excitatory input: $[{}^{+}\mathbf{W}^{1}]\mathbf{p}$, where ${}^{+}\mathbf{W}^{1}$: an on-center weight matrix

$${}^{+}\mathbf{W}^{1} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{15.2-2}$$

The inhibitory input: $[{}^{-}\mathbf{W}^{1}]\mathbf{p}$, where ${}^{-}\mathbf{W}^{1}$: an off-surround weight matrix

$${}^{-}\mathbf{W}^{1} = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{bmatrix} \tag{15.2-3}$$

The inhibitory bias ${}^{-}\mathbf{b}^{1}$ is set to zero (the lower limit of the shunting model is set to zero).

The excitatory bias ${}^{+}\mathbf{b}^{1}$ is set to the same value (the upper limit for all neurons is the same).

$${}^{+}b_i^1 = {}^{+}b^1, i = 1,2,\ldots,S^1 \tag{15.2-4}$$

Response of neuron *i*,

$$\varepsilon \frac{dn_i^1(t)}{dt} = -n_i^1(t) + ({}^{+}b^1 - n_i^1(t))p_i - n_i^1(t)\sum_{j \neq i} p_j \tag{15.2-5}$$

In the steady state ($dn_i^1(t)/dt = 0$),

$$0 = -n_i^1 + ({}^{+}b^1 - n_i^1)p_i - n_i^1 \sum_{j \neq i} p_j \tag{15.2-6}$$

$$n_i^1 = \frac{{}^{+}b^1 p_i}{1 + \sum_{j=1}^{S^1} p_j} \tag{15.2-7}$$

Defining relative intensity of input *i*,

$$\bar{p}_i = \frac{p_i}{P} \text{ where } P = \sum_{j=1}^{S^1} p_j \tag{15.2-8}$$

The steady state neuron,

$$n_i^1 = \left(\frac{{}^{+}b^1 P}{1 + P} \right) \bar{p}_i \tag{15.2-9}$$

- n_i^1 is proportional to the relative intensity $\bar{p}_i$, regardless of the magnitude of the total input *P*.
- The total neuron activity is bounded,

$$\sum_{j=1}^{S^1} n_j^1 = \sum_{j=1}^{S^1} \left(\frac{{}^{+}b^1 P}{1 + P} \right) \bar{p}_j = \left(\frac{{}^{+}b^1 P}{1 + P} \right) \leq {}^{+}b^1 \tag{15.2-10}$$

- The input vector is normalized so that the total activity is less than ${}^{+}b^1$, while the relative intensities of the individual elements of the input vector are maintained.

Example: Two neurons, with ${}^{+}b^{1}=1$, $\varepsilon = 0.1$,

$$0.1\frac{dn_1^1(t)}{dt} = -n_1^1(t) + (1 - n_1^1(t))p_1 - n_1^1(t)p_2 \qquad (15.2\text{-}11)$$

$$0.1\frac{dn_2^1(t)}{dt} = -n_2^1(t) + (1 - n_2^1(t))p_2 - n_2^1(t)p_1 \qquad (15.2\text{-}12)$$

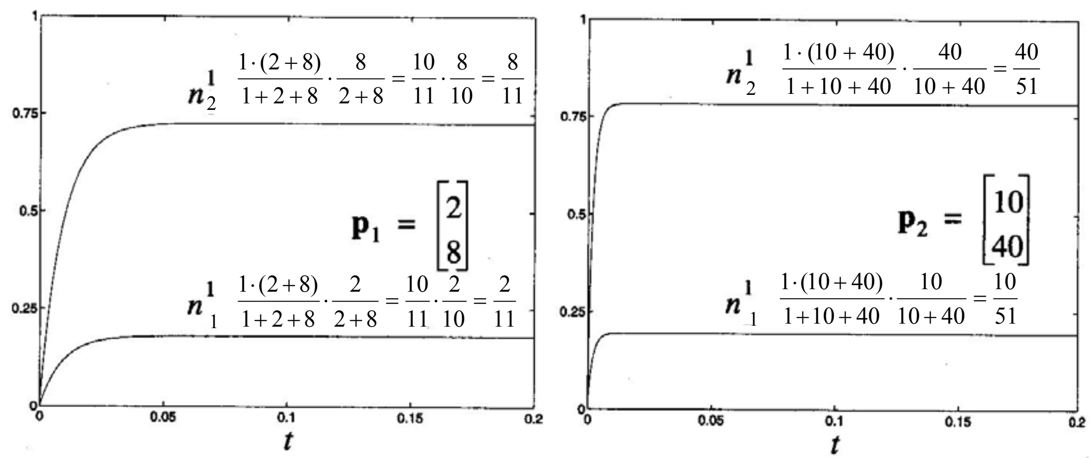
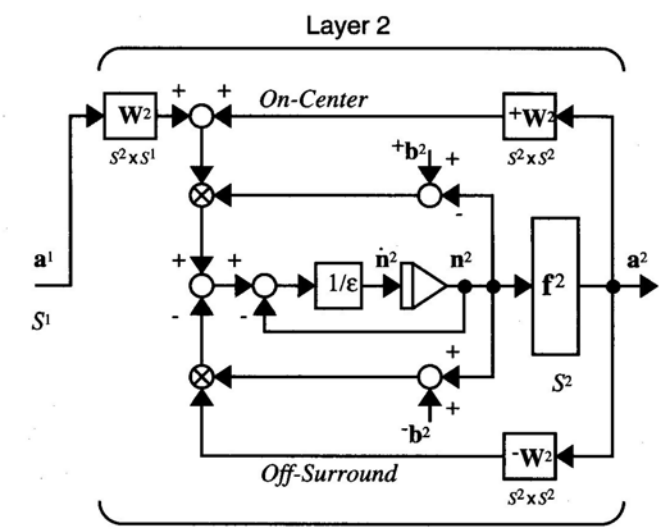


Figure 15.2-3 Responses from Layer 1

Layer 2

1. Normalizing total activity in the layer
2. Contrast enhancing the patterns, so that the neuron that receives the largest input will dominate the response.
3. Operating as a short-term memory (STM) by storing the contrast-enhanced pattern.



$$\varepsilon d\mathbf{n}^2/dt = -\mathbf{n}^2 + ({}^{+}\mathbf{b}^2 - \mathbf{n}^2)\{[{}^{+}\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2) + \mathbf{W}^2\mathbf{a}^1\} - (\mathbf{n}^2 + {}^{-}\mathbf{b}^2)[{}^{-}\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2)$$

Figure 15.2-4 Layer 2 of Grossberg Network

$$\varepsilon \frac{d\mathbf{n}^2(t)}{dt} = -\mathbf{n}^2(t) + ({}^+\mathbf{b}^2 - \mathbf{n}^2(t))\{[{}^+\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2(t)) + \mathbf{W}^2\mathbf{a}^1\} - (\mathbf{n}^2(t) + {}^-\mathbf{b}^2)[{}^-\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2(t)) \quad (15.2\text{-}13)$$

- The excitatory input $\{[{}^+\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2(t)) + \mathbf{W}^2\mathbf{a}^1\}$, where ${}^+\mathbf{W}^2 = {}^+\mathbf{W}^1$ provides on-center feedback connections.
- The inhibitory input, $[{}^-\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2(t))$, where ${}^-\mathbf{W}^2 = {}^-\mathbf{W}^1$ provides off-surround feedback connections.
- The rows of $\mathbf{W}^2$, adaptive weight, after training, will represent the prototype patterns.

Example: A two-neuron layer

$$\varepsilon = 0.1, {}^+\mathbf{b}^2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, {}^-\mathbf{b}^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathbf{W}^2 = \begin{bmatrix} ({}_1\mathbf{w}^2)^T \\ ({}_2\mathbf{w}^2)^T \end{bmatrix} = \begin{bmatrix} 0.9 & 0.45 \\ 0.45 & 0.9 \end{bmatrix} \quad (15.2\text{-}14)$$

$$f^2(n) = \frac{10(n)^2}{1 + (n)^2} \quad (15.2\text{-}15)$$

$$(0.1)\frac{dn_1^2(t)}{dt} = -n_1^2(t) + (1 - n_1^2(t))\{f^2(n_1^2(t)) + ({}_1\mathbf{w}^2)^T\mathbf{a}^1\} - n_1^2(t)f^2(n_2^2(t)) \quad (15.2\text{-}16)$$

$$(0.1)\frac{dn_2^2(t)}{dt} = -n_2^2(t) + (1 - n_2^2(t))\{f^2(n_2^2(t)) + ({}_2\mathbf{w}^2)^T\mathbf{a}^1\} - n_2^2(t)f^2(n_1^2(t)) \quad (15.2\text{-}17)$$

- The inputs to Layer 2 are the inner products between the prototype patterns and the output of Layer 1.
- Layer 2 performs a competition between the neurons, which tends to contrast enhance the output pattern.
- In the Grossberg network, the competition maintains large values and attenuates small values, but does not necessarily drive all small values to zero.

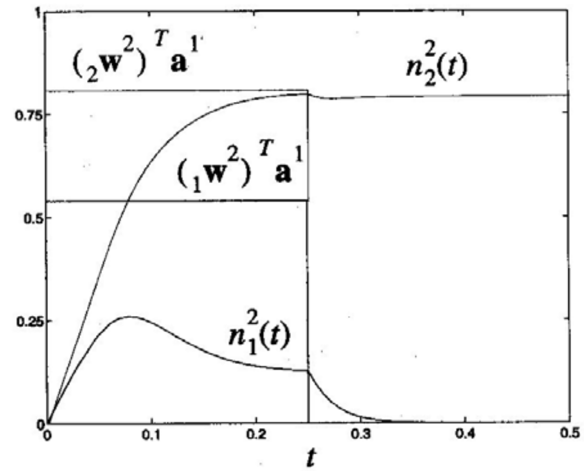


Figure 15.2-5 Response of Layer 2, Applying $\mathbf{a}_1 = [0.2 \quad 0.8]^T$ for 0.25 Seconds

$$({}_1\mathbf{w}^2)^T\mathbf{a}^1 = [0.9 \quad 0.45]\begin{bmatrix}0.2\\0.8\end{bmatrix} = 0.54 \tag{15.2-18}$$

$$({}_2\mathbf{w}^2)^T\mathbf{a}^1 = [0.45 \quad 0.9]\begin{bmatrix}0.2\\0.8\end{bmatrix} = 0.81 \tag{15.2-19}$$

- After the input has been set to zero, the network further enhances the contrast and stores the pattern.

15.2.1 Choice of Transfer Function

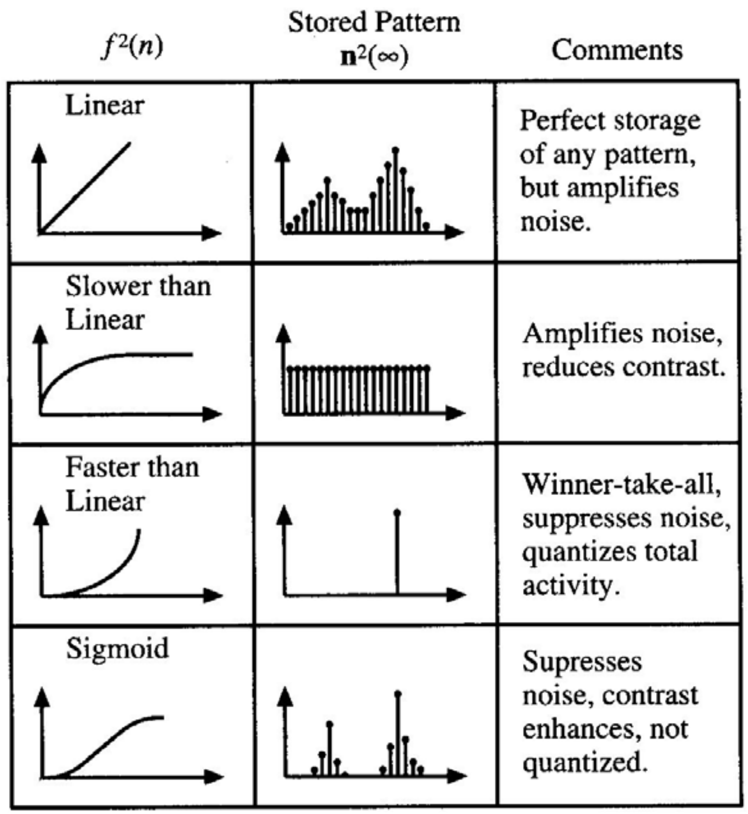


Figure 15.2.1-1 Effect of Transfer Function $f^2(n)$

15.2.2 Learning Law

- Grossberg calls these adaptive weights, $\mathbf{W}^2$, the long-term memory (LTM).
- The rows of $\mathbf{W}^2$ represent patterns that have been stored and that the network is be able to recognize.

One learning law for $\mathbf{W}^2$,

$$\frac{dw_{i,j}^2(t)}{dt} = \alpha\{-w_{i,j}^2(t) + n_i^2(t)n_j^1(t)\} \tag{15.2.2-1}$$

- The first term is a passive decay term.
- The second term implements a Hebbian-type learning.
- Together, this learning implement the Hebb rule with decay.

Turn off learning when $n_i^2(t)$ is not active,

$$\frac{dw_{i,j}^2(t)}{dt} = \alpha n_i^2(t)\{-w_{i,j}^2(t) + n_j^1(t)\} \tag{15.2.2-2}$$

$$\frac{d[{}_i\mathbf{w}^2(t)]}{dt} = \alpha n_i^2(t)\{-[{}_i\mathbf{w}^2(t)] + \mathbf{n}^1(t)\} \tag{15.2.2-3}$$

- This is the continuous-time implementation of the instar learning rule.

Example: A network with two neurons in each layer, $\alpha = 1$,

$$\frac{dw_{1,1}^2(t)}{dt} = n_1^2(t)\{-w_{1,1}^2(t) + n_1^1(t)\} \quad (15.2.2\text{-}4)$$

$$\frac{dw_{1,2}^2(t)}{dt} = n_1^2(t)\{-w_{1,2}^2(t) + n_2^1(t)\} \quad (15.2.2\text{-}5)$$

$$\frac{dw_{2,1}^2(t)}{dt} = n_2^2(t)\{-w_{2,1}^2(t) + n_1^1(t)\} \quad (15.2.2\text{-}6)$$

$$\frac{dw_{2,2}^2(t)}{dt} = n_2^2(t)\{-w_{2,2}^2(t) + n_2^1(t)\} \quad (15.2.2\text{-}7)$$

- Two different input patterns are alternately presented to the network for periods of 0.2 seconds at a time.
- Layer 1 and Layer 2 converge very quickly, in comparison with the convergence of the weights,

$$\text{for pattern 1: } \mathbf{n}^1 = \begin{bmatrix} 0.9 \\ 0.45 \end{bmatrix}, \mathbf{n}^2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (15.2.2\text{-}8)$$

$$\text{for pattern 2: } \mathbf{n}^1 = \begin{bmatrix} 0.45 \\ 0.9 \end{bmatrix}, \mathbf{n}^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (15.2.2\text{-}9)$$

- Pattern 1 is coded by the first neuron in Layer 2.
- Pattern 2 is coded by the second neuron in Layer 2.

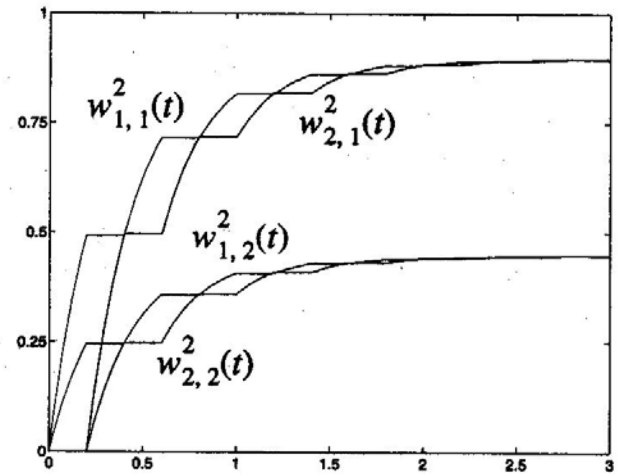


Figure 15.2.2-1 Response of the Adaptive Weights

Three major differences between the Grossberg and the basic Kohonen competitive network

1. The Grossberg network is a continuous-time network (satisfies a set of nonlinear differential equations).
2. Layer 1 of the Grossberg network automatically normalizes the input vectors.
3. Layer 2 of the Grossberg network can perform a "soft" competition, rather than the winner-take-all competition of the Kohonen network.

16 Adaptive Resonance Theory

16.1 Overview of Adaptive Resonance

- Layer 1, Layer 2, Orienting Subsystem
- L1-L2: Instar, L2-L1: Outstar

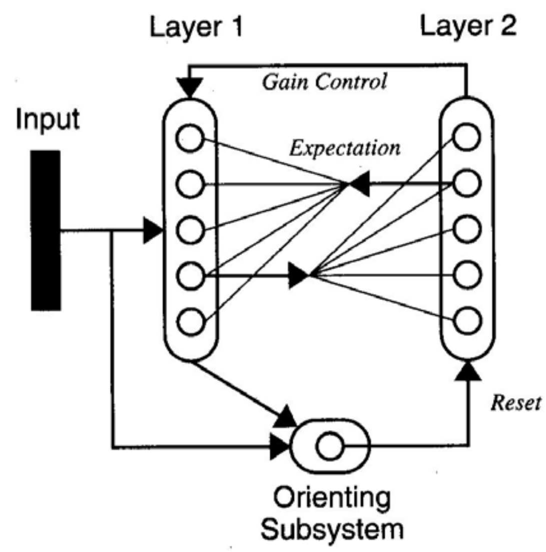
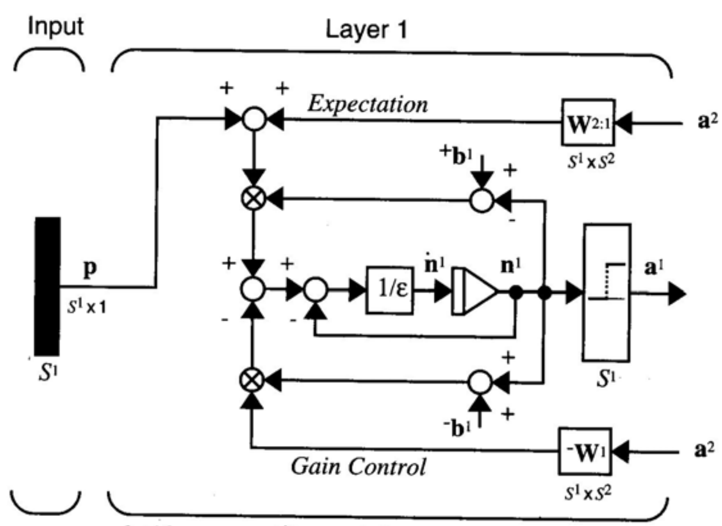


Figure 16.1-1 Basic ART Architecture

16.2 Layer 1

- The main purpose of Layer 1 is to compare the input pattern with the expectation pattern from Layer 2. (Both patterns are binary in ART1.)
- If the patterns are not closely matched, the orienting subsystem will cause a reset in Layer 2.
- If the patterns are close enough, Layer 1 combines the expectation and the input to form a new prototype pattern.



$$\varepsilon\, d\mathbf{n}^1/dt = -\mathbf{n}^1 + ({}^{+}\mathbf{b}^1 - \mathbf{n}^1)\{\mathbf{p} + \mathbf{W}^{2:1}\mathbf{a}^2\} - (\mathbf{n}^1 + {}^{-}\mathbf{b}^1)[{}^{-}\mathbf{W}^1]\mathbf{a}^2$$

Figure 16.2-1 Layer 1 of the ART1 Network

$$\varepsilon\frac{d\mathbf{n}^1(t)}{dt} = -\mathbf{n}^1(t) + ({}^{+}\mathbf{b}^1 - \mathbf{n}^1(t))\{\mathbf{p} + \mathbf{W}^{2:1}\mathbf{a}^2(t)\} - (\mathbf{n}^1(t) + {}^{-}\mathbf{b}^1)[{}^{-}\mathbf{W}^1]\mathbf{a}^2(t) \tag{16.2-1}$$

$$\mathbf{a}^1 = \mathbf{hardlim}^{+}(\mathbf{n}^1) \tag{16.2-2}$$

$$hardlim^{+}(n) = \begin{cases} 1, & n > 0 \\ 0, & n \le 0 \end{cases} \tag{16.2-3}$$

- The excitatory input: $\mathbf{p} + \mathbf{W}^{2:1}\mathbf{a}^2(t)$,

If the *j*th neuron in Layer 2 wins the competition,

$$\mathbf{W}^{2:1}\mathbf{a}^2 = \begin{bmatrix}\mathbf{w}_1^{2:1} & \mathbf{w}_2^{2:1} & \cdots & \mathbf{w}_j^{2:1} & \cdots & \mathbf{w}_{S^2}^{2:1}\end{bmatrix}\begin{bmatrix}0\\0\\\vdots\\1\\\vdots\end{bmatrix} = \mathbf{w}_j^{2:1} \tag{16.2-4}$$

$$\mathbf{p} + \mathbf{W}^{2:1}\mathbf{a}^2 = \mathbf{p} + \mathbf{w}_j^{2:1} \tag{16.2-5}$$

- The inhibitory input: $[{}^{-}\mathbf{W}^1]\mathbf{a}^2(t)$, the gain control term, where

$${}^{-}\mathbf{W}^1 = \begin{bmatrix}1 & 1 & \cdots & 1\\1 & 1 & \cdots & 1\\\vdots & \vdots & \ddots & \vdots\\1 & 1 & \cdots & 1\end{bmatrix} \tag{16.2-6}$$

- The gain control input to Layer 1 will be one when Layer 2 is active, and zero when Layer 2 is inactive.

16.2.1 Steady State Analysis

$$\varepsilon \frac{dn_i^1}{dt} = -n_i^1 + ({}^{+}b^1 - n_i^1)\left\{ p_i + \sum_{j=1}^{S^2} w_{i,j}^{2:1} a_j^2 \right\} - (n_i^1 + {}^{-}b^1) \sum_{j=1}^{S^2} a_j^2 \qquad (16.2.1\text{-}1)$$

where $\varepsilon << 1$

If Layer 2 is inactive, each $a_j^2 = 0$,

$$\varepsilon \frac{dn_i^1}{dt} = -n_i^1 + ({}^{+}b^1 - n_i^1)\{ p_i \} \qquad (16.2.1\text{-}2)$$

$$0 = -n_i^1 + ({}^{+}b^1 - n_i^1) p_i = -(1 + p_i) n_i^1 + {}^{+}b^1 p_i \qquad (16.2.1\text{-}3)$$

$$n_i^1 = \frac{{}^{+}b^1 p_i}{1 + p_i} \qquad (16.2.1\text{-}4)$$

- If $p_i = 0$ then $n_i^1 = 0$, and if $p_i = 1$ then $n_i^1 = {}^{+}b^1 / 2 > 0$.

By using *hardlim*$^{+}$ function,

$$\mathbf{a}^1 = \mathbf{p} \qquad (16.2.1\text{-}5)$$

- When Layer 2 is inactive, the output of Layer 1 is the same as the input pattern.

If Layer 2 is active and neuron *j* is the winning neuron in Layer 2, $a_j^2 = 1$ and $a_k^2 = 0$ for $k \neq j$,

$$\varepsilon \frac{dn_i^1}{dt} = -n_i^1 + ({}^{+}b^1 - n_i^1)\left\{p_i + w_{i,j}^{2:1}\right\} - (n_i^1 + {}^{-}b^1) \tag{16.2.1-6}$$

$$0 = -n_i^1 + ({}^{+}b^1 - n_i^1)\left\{p_i + w_{i,j}^{2:1}\right\} - (n_i^1 + {}^{-}b^1) = -(1 + p_i + w_{i,j}^{2:1} + 1)n_i^1 + ({}^{+}b^1(p_i + w_{i,j}^{2:1}) - {}^{-}b^1) \tag{16.2.1-7}$$

$$n_i^1 = \frac{{}^{+}b^1(p_i + w_{i,j}^{2:1}) - {}^{-}b^1}{2 + p_i + w_{i,j}^{2:1}} \tag{16.2.1-8}$$

To obtain AND operation,

$${}^{+}b^1(p_i + w_{i,j}^{2:1}) - {}^{-}b^1 \tag{16.2.1-9}$$

$${}^{+}b^1(2) - {}^{-}b > 0 \tag{16.2.1-10}$$

$${}^{+}b^1 - {}^{-}b < 0 \tag{16.2.1-11}$$

$${}^{+}b^1(2) > {}^{-}b > {}^{+}b^1 \tag{16.2.1-12}$$

For example, ${}^{+}b^1 = 1$ and ${}^{-}b^1 = 1.5$,

$$\mathbf{a}^1 = \mathbf{p} \cap \mathbf{w}_j^{2:1} \tag{16.2.1-13}$$

Summary of the steady state operation of Layer 1,

If Layer 2 is not active (i.e., each $a_j^2 = 0$),

$$\mathbf{a}^1 = \mathbf{p} \tag{16.2.1-14}$$

If Layer 2 is active (i.e., one $a_j^2 = 1$),

$$\mathbf{a}^1 = \mathbf{p} \cap \mathbf{w}_j^{2:1} \tag{16.2.1-15}$$

Example:

$$\varepsilon = 0.1, {}^{+}b^1 = 1, {}^{-}b^1 = 1.5 \tag{16.2.1-16}$$

$$\mathbf{W}^{2:1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \mathbf{p} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{16.2.1-17}$$

If Layer 2 is active, and neuron 2 of Layer 2 wins the competition,

$$0.1\frac{dn_1^1}{dt} = -n_1^1 + (1-n_1^1)\left\{p_1 + w_{1,2}^{2:1}\right\} - (n_1^1 + 1.5) = -n_1^1 + (1-n_1^1)\left\{0+1\right\} - (n_1^1 + 1.5) = -3n_1^1 - 0.5 \tag{16.2.1-18}$$

$$0.1\frac{dn_2^1}{dt} = -n_2^1 + (1-n_2^1)\left\{p_2 + w_{2,2}^{2:1}\right\} - (n_2^1 + 1.5) = -n_2^1 + (1-n_2^1)\left\{1+1\right\} - (n_2^1 + 1.5) = -4n_2^1 + 0.5 \tag{16.2.1-19}$$

$$\frac{dn_1^1}{dt} = -30n_1^1 - 5 \tag{16.2.1-20}$$

$$\frac{dn_2^1}{dt} = -40n_2^1 + 5 \tag{16.2.1-21}$$

$$n_1^1(t) = -\frac{1}{6}[1 - e^{-30t}] \tag{16.2.1-22}$$

$$n_2^1(t) = \frac{1}{8}[1 - e^{-40t}] \tag{16.2.1-23}$$

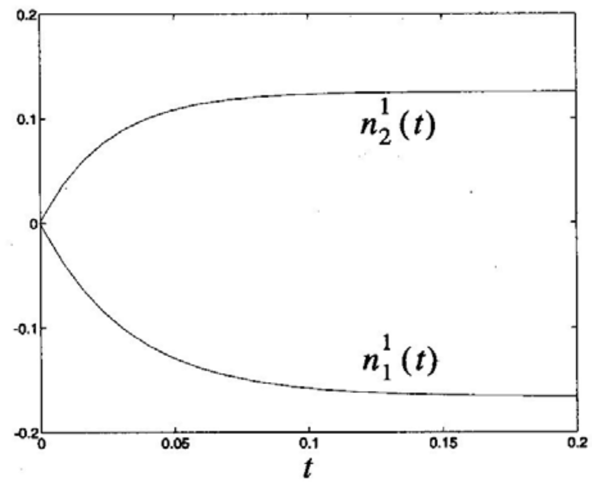
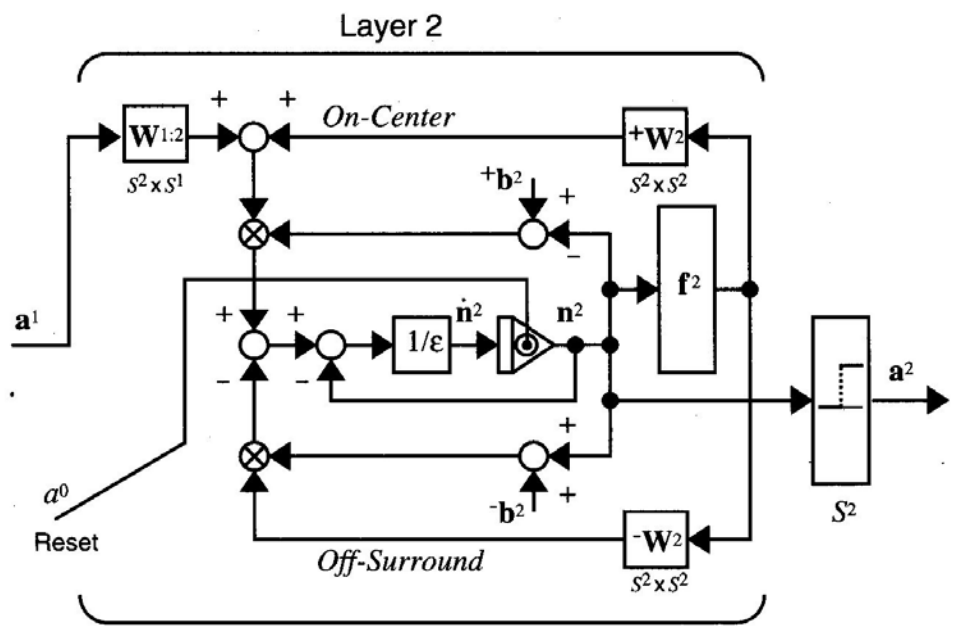


Figure 16.2.1-1 Response of Layer 1

$$\mathbf{p} \cap \mathbf{w}_2^{2:1} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cap \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \mathbf{a}^1 \qquad (16.2.1\text{-}24)$$

16.3 Layer 2

- Layer 2 of the ART1 network is almost identical to Layer 2 of the Grossberg network.
- Its main purpose is to contrast enhance its output pattern, winner-take-all competition.
- The integrator of ART can be reset.
- In this type of integrator any positive outputs are reset to zero whenever the a^0 signal becomes positive.
- The outputs that are reset remain inhibited for a long period of time until an adequate match has occurred and the weights have been updated.
- The reset signal, a^0, is the output of the orienting subsystem. It generates a reset whenever there is a mismatch at Layer 1 between the input signal and the L2-L1 expectation.
- Two transfer functions are used in ART1.
- The transfer function $\mathbf{f}^2(\mathbf{n}^2)$ is used for the on-center/off-surround feedback connections.
- The output of Layer 2 is computed as $\mathbf{a}^2 = \mathbf{hardlim}^+(\mathbf{n}^2)$.



$$\varepsilon\, d\mathbf{n}^2/dt = -\mathbf{n}^2 + ({}^{+}\mathbf{b}^2 - \mathbf{n}^2)\{[{}^{+}\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2) + \mathbf{W}^{1:2}\mathbf{a}^1\} - (\mathbf{n}^2 + {}^{-}\mathbf{b}^2)[{}^{-}\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2)$$

Figure 16.3-1 Layer 2 of the ART1 Network

$$\varepsilon \frac{d\mathbf{n}^2(t)}{dt} = -\mathbf{n}^2(t) + ({}^{+}\mathbf{b}^2 - \mathbf{n}^2(t))\{[{}^{+}\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2(t)) + \mathbf{W}^{1:2}\mathbf{a}^1\} - (\mathbf{n}^2(t) + {}^{-}\mathbf{b}^2)[{}^{-}\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2(t)) \qquad (16.3\text{-}1)$$

The excitatory input: $\{[{}^{+}\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2(t)) + \mathbf{W}^{1:2}\mathbf{a}^1\}$

${}^{+}\mathbf{W}^2$: on-center feedback connections

$\mathbf{W}^{1:2}$: adaptive weights, trained according to an instar rule

The inhibitory input: $[{}^{-}\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2(t))$

${}^{-}\mathbf{W}^2$: off-surround feedback connections

Example: A two-neuron layer,

$$\varepsilon = 0.1, {}^{+}\mathbf{b}^2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, {}^{-}\mathbf{b}^2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{W}^{1:2} = \begin{bmatrix} ({}_1\mathbf{w}^{1:2})^T \\ ({}_2\mathbf{w}^{1:2})^T \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 1 & 0 \end{bmatrix} \tag{16.3-2}$$

$$f^2(n) = \begin{cases} 10(n)^2, & n \ge 0 \\ 0, & n < 0 \end{cases} \tag{16.3-3}$$

$$(0.1)\frac{dn_1^2(t)}{dt} = -n_1^2(t) + (1 - n_1^2(t))\{f^2(n_1^2(t)) + ({}_1\mathbf{w}^{1:2})^T\mathbf{a}^1\} - (n_1^2(t) + 1) f^2(n_2^2(t)) \tag{16.3-4}$$

$$(0.1)\frac{dn_2^2(t)}{dt} = -n_2^2(t) + (1 - n_2^2(t))\{f^2(n_2^2(t)) + ({}_2\mathbf{w}^{1:2})^T\mathbf{a}^1\} - (n_2^2(t) + 1) f^2(n_1^2(t)) \tag{16.3-5}$$

- The inputs to Layer 2 are the inner products of the prototype patterns with the output of Layer 1.
- Layer 2 then performs a competition between the neurons.
- The transfer function $f^2(n)$ is chosen to be a faster-than-linear transfer function.
- After the competition, one neuron output will be 1, and the other neuron output will be zero.

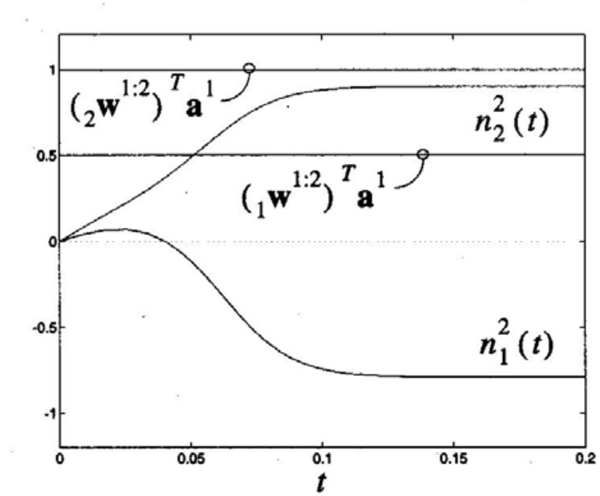


Figure 16.3-2 Response of Layer 2, $\mathbf{a}^1 = [1 \;\; 0]^T$

The steady state Layer 2 output,

$$\mathbf{a}^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{16.3-6}$$

The steady state operation of Layer 2,

$$a_i^2 = \begin{cases} 1, & if \; ((_i\mathbf{w}^{1:2})^T \mathbf{a}^1 = \max[(_j\mathbf{w}^{1:2})^T \mathbf{a}^1]) \\ 0, & otherwise \end{cases} \tag{16.3-7}$$

16.4 Orienting Subsystem

- The purpose of Orienting Subsystem is to determine if there is a sufficient match between the L2-L1 expectation and the input pattern.

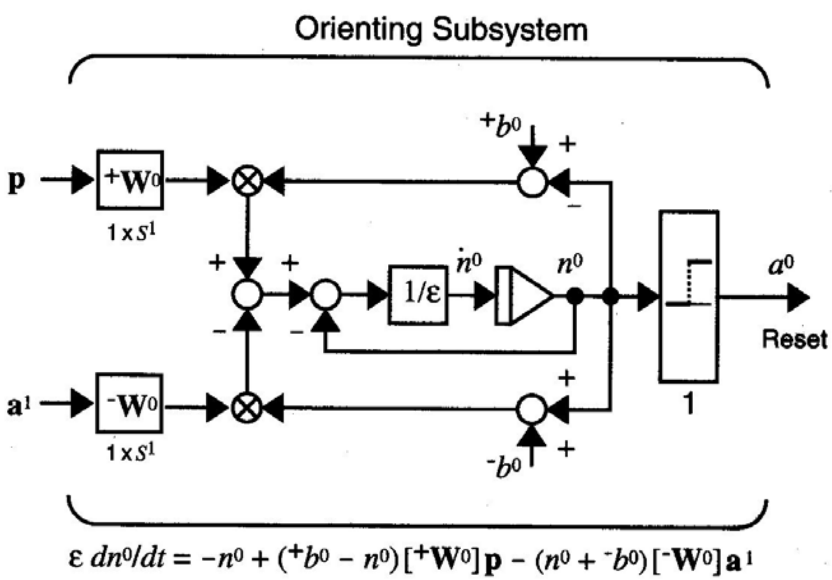


Figure 16.4-1 Orienting Subsystem of the ART1 Network

$$\varepsilon\frac{dn^0(t)}{dt} = -n^0(t) + ({}^{+}b^0 - n^0(t))\{{}^{+}\mathbf{W}^0\mathbf{p}\} - (n^0(t) + {}^{-}b^0)\{{}^{-}\mathbf{W}^0\mathbf{a}^1\} \tag{16.4-1}$$

The excitatory input: ${}^{+}\mathbf{W}^0\mathbf{p}$, where

$${}^{+}\mathbf{W}^0 = \begin{bmatrix}\alpha & \alpha & \cdots & \alpha\end{bmatrix} \tag{16.4-2}$$

$${}^{+}\mathbf{W}^0\mathbf{p} = \begin{bmatrix}\alpha & \alpha & \cdots & \alpha\end{bmatrix}\mathbf{p} = \alpha\sum_{j=1}^{S^1} p_j = \alpha\|\mathbf{p}\|^2 \tag{16.4-3}$$

The inhibitory input: ${}^{-}\mathbf{W}^0\mathbf{a}^1$, where

$${}^{-}\mathbf{W}^0 = \begin{bmatrix}\beta & \beta & \cdots & \beta\end{bmatrix} \tag{16.4-4}$$

$${}^{-}\mathbf{W}^0\mathbf{a}^1 = \begin{bmatrix}\beta & \beta & \cdots & \beta\end{bmatrix}\mathbf{a}^1 = \beta\sum_{j=1}^{S^1} a_j^1(t) = \beta\|\mathbf{a}^1\|^2 \tag{16.4-5}$$

$$0 = -n^0 + ({}^{+}b^0 - n^0)\{\alpha\|\mathbf{p}\|^2\} - (n^0 + {}^{-}b^0)\{\beta\|\mathbf{a}^1\|^2\} = -(1 + \alpha\|\mathbf{p}\|^2 + \beta\|\mathbf{a}^1\|^2)n^0 + {}^{+}b^0(\alpha\|\mathbf{p}\|^2) - {}^{-}b^0(\beta\|\mathbf{a}^1\|^2) \tag{16.4-6}$$

$$n^0 = \frac{{}^{+}b^0(\alpha\|\mathbf{p}\|^2) - {}^{-}b^0(\beta\|\mathbf{a}^1\|^2)}{(1 + \alpha\|\mathbf{p}\|^2 + \beta\|\mathbf{a}^1\|^2)n^0} \tag{16.4-7}$$

${}^{+}b^0 = {}^{-}b^0 = 1$, then $n^0 > 0$ if $\alpha\|\mathbf{p}\|^2 - \beta\|\mathbf{a}^1\|^2 > 0$,

$$n^0 > 0, if\ \frac{\|\mathbf{a}^1\|^2}{\|\mathbf{p}\|^2} < \frac{\alpha}{\beta} = \rho \tag{16.4-8}$$

ρ: vigilance parameter, $0 < \rho < 1$

If the vigilance is close to 1, a reset will occur unless $\mathbf{a}^1$ is close to $\mathbf{p}$.

If the vigilance is close to 0, $\mathbf{a}^1$ need not be close to $\mathbf{p}$ to prevent a reset.

Eample:

$$\varepsilon = 0.1, \alpha = 3, \beta = 4, or(\rho = 0.75) \tag{16.4-9}$$

$$\mathbf{p} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{a}^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{16.4-10}$$

$$(0.1)\frac{dn^0(t)}{dt} = -n^0(t) + (1 - n^0(t))\{3(p_1 + p_2)\} - (n^0(t) + 1)\{4(a_1^1 + a_2^1)\} \tag{16.4-11}$$

$$\frac{dn^0(t)}{dt} = -110n^0(t) + 20 \tag{16.4-12}$$

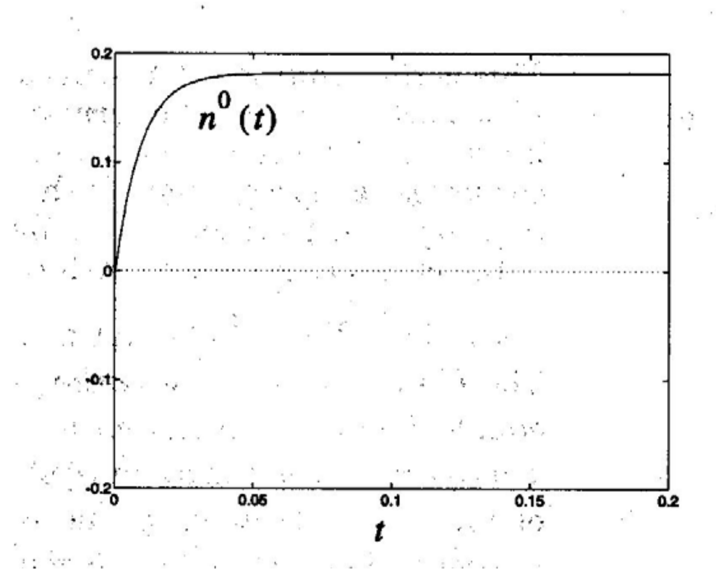


Figure 16.4-2 Response of the Orienting Subsystem

The steady state operation of the Orienting Subsystem,

$$a^0 = \begin{cases} 1, & if\,[\|\mathbf{a}^1\|^2 / \|\mathbf{p}\|^2 < \rho] \\ 0, & otherwise \end{cases} \qquad (16.4\text{-}13)$$

16.5 Learning Law: Ll-L2

16.5.2 Learning Law: L1-L2

Learning law for $\mathbf{W}^{1:2}$,

$$\frac{d[{}_i\mathbf{w}^{1:2}(t)]}{dt} = a_i^2(t)\left[\{{}^+\mathbf{b} - {}_i\mathbf{w}^{1:2}(t)\}\zeta[{}^+\mathbf{W}]\mathbf{a}^1(t) - \{{}_i\mathbf{w}^{1:2}(t) + {}^-\mathbf{b}\}[{}^-\mathbf{W}]\mathbf{a}^1(t)\right] \quad (16.5.2\text{-}1)$$

$${}^+\mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, {}^-\mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, {}^+\mathbf{W} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, {}^-\mathbf{W} = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{bmatrix} \quad (16.5.2\text{-}2)$$

- When neuron i of Layer 2 is active, the ith row of $\mathbf{W}^{1:2}$, ${}_i\mathbf{w}^{1:2}$, is moved in the direction of $\mathbf{a}^1$.
- ${}_i\mathbf{w}^{1:2}$ is normalized.
- The excitatory bias is ${}^+\mathbf{b} = \mathbf{1}$ (a vector of l's), and the inhibitory bias is ${}^-\mathbf{b} = \mathbf{0}$ to ensure ${}_i\mathbf{w}^{1:2}$ remain between 0 and 1.

$$\frac{d[w_{i,j}^{1:2}(t)]}{dt} = a_i^2(t)\left[(1 - w_{i,j}^{1:2}(t))\zeta a_j^1(t) - w_{i,j}^{1:2}(t)\sum_{k \neq j} a_k^1(t)\right] \quad (16.5.2\text{-}3)$$

Neuron i is active in Layer 2 ($a_i^2(t) = 1$),

$$0 = \left[(1 - w_{i,j}^{1:2})\zeta a_j^1 - w_{i,j}^{1:2}\sum_{k \neq j} a_k^1\right] \quad (16.5.2\text{-}4)$$

When $a_j^1 = 1$,

$$0 = (1 - w_{i,j}^{1:2})\zeta - w_{i,j}^{1:2}(\left\|\mathbf{a}^1\right\|^2 - 1) = -(\zeta + \left\|\mathbf{a}^1\right\|^2 - 1)w_{i,j}^{1:2} + \zeta \tag{16.5.2-5}$$

$$w_{i,j}^{1:2} = \frac{\zeta}{\zeta + \left\|\mathbf{a}^1\right\|^2 - 1} \tag{16.5.2-6}$$

When $a_j^1 = 0$,

$$0 = -w_{i,j}^{1:2}\left\|\mathbf{a}^1\right\|^2 \tag{16.5.2-7}$$

$$w_{i,j}^{1:2} = 0 \tag{16.5.2-8}$$

$$_i\mathbf{w}^{1:2} = \frac{\zeta\mathbf{a}^1}{\zeta + \left\|\mathbf{a}^1\right\|^2 - 1} \tag{16.5.2-9}$$

where $\zeta > 1$,

- The prototype patterns is normalized.

16.5.3 Learning Law: L2-L1

Learning law for $\mathbf{W}^{2:1}$,

$$\frac{d[\mathbf{w}_j^{2:1}(t)]}{dt} = a_j^2(t)[-\mathbf{w}_j^{2:1}(t) + \mathbf{a}^1(t)] \quad (16.5.3\text{-}1)$$

- When neuron *j* in Layer 2 is active, column *j* of $\mathbf{W}^{2:1}$ is moved toward the $\mathbf{a}^1$.

$$\mathbf{0} = -\mathbf{w}_j^{2:1} + \mathbf{a}^1, \mathbf{w}_j^{2:1} = \mathbf{a}^1 \quad (16.5.3\text{-}2)$$

- $\mathbf{W}^{1:2}$ and $\mathbf{W}^{2:1}$ are updated at the same time.
- When neuron *j* of Layer 2 is active and there is a sufficient match between the expectation and the input pattern (which indicates a resonance condition), then row *j* of $\mathbf{W}^{1:2}$ and column *j* of $\mathbf{W}^{2:1}$ are adapted.
- In fast learning, column *j* of $\mathbf{W}^{2:1}$ is set to $\mathbf{a}^1$, while row *j* of $\mathbf{W}^{1:2}$ is set to a normalized version of $\mathbf{a}^1$.

16.6 ART1 Algorithm Summary

Initialization

- $\mathbf{W}^{2:1}$ matrix is set to all 1's.
- $\mathbf{W}^{1:2}$ matrix is set to $\zeta/(\zeta + S^1 - 1)$.

Algorithm

1. First, we present an input pattern to the network. Since Layer 2 is not active on initialization (i.e., each $a_j^2 = 0$), the output of Layer 1 is

$$\mathbf{a}^1 = \mathbf{p} \tag{16.6-1}$$

2. Next, we compute the input to Layer 2,

$$\mathbf{W}^{1:2}\mathbf{a}^1 \tag{16.6-2}$$

and activate the neuron in Layer 2 with the largest input;

$$a_i^2 = \begin{cases} 1, & if\ (({}_i\mathbf{w}^{1:2})^T\mathbf{a}^1 = \max[({}_k\mathbf{w}^{1:2})^T\mathbf{a}^1]) \\ 0, & otherwise \end{cases} \tag{16.6-3}$$

In case of a tie, the neuron with the smallest index is declared the winner.

3. We then compute the L2-L1 expectation (where we assume neuron *j* of Layer 2 is activated):

$$\mathbf{W}^{2:1}\mathbf{a}^2 = \mathbf{w}_j^{2:1} \tag{16.6-4}$$

4. Now that Layer 2 is active, we adjust the Layer 1 output to include the L2-Ll expectation:

$$\mathbf{a}^1 = \mathbf{p} \cap \mathbf{w}_j^{2:1} = \mathbf{p} \tag{16.6-5}$$

5. Next, the Orienting Subsystem determines the degree of match between the expectation and the input pattern:

$$a^0 = \begin{cases} 1, & if\, [\|\mathbf{a}^1\|^2 / \|\mathbf{p}\|^2 < \rho] \\ 0, & otherwise \end{cases} \tag{16.6-6}$$

6. If a^0= 1, then we set $a_j^2 = 0$, inhibit it until an adequate match occurs (resonance), and return to step 1. If a^0= 0, we continue with step 7.

7. Resonance has occurred. Therefore we update row j of $\mathbf{W}^{1:2}$:

$$_j\mathbf{w}^{1:2} = \frac{\zeta \mathbf{a}^1}{\zeta + \|\mathbf{a}^1\|^2 - 1} \tag{16.6-7}$$

8. We now update column j of $\mathbf{W}^{2:1}$:

$$\mathbf{w}_j^{2:1} = \mathbf{a}^1 \tag{16.6-8}$$

9. We remove the input pattern, restore all inhibited neurons in Layer 2, and return to step 1 with a new input pattern.

- The input patterns continue to be applied to the network until the weights stabilize (do not change).
- ART1 algorithm always forms stable clusters for any set of input patterns.

17 Stability

17.1 Recurrent Networks

- For feedforward networks, the output is constant (for a fixed input) and is a function only of the network input.
- For recurrent networks, the output of the network is a function of time.

By nonlinear differential equations of the form,

$$\frac{d}{dt}\mathbf{a}(t) = \mathbf{g}(\mathbf{a}(t),\mathbf{p}(t),t) \tag{17.1-1}$$

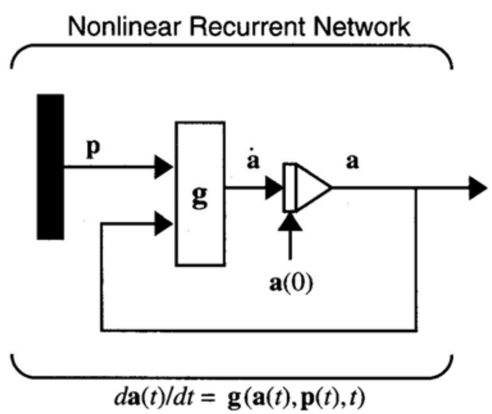


Figure 17.1-1 Nonlinear, Continuous-Time, Recurrent Network

17. 2 Stability Concepts

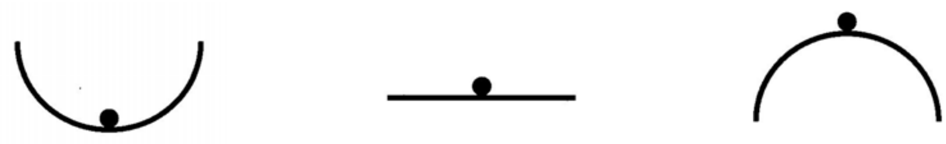


Figure 17.2-1 Three Ball Bearing Systems with Dissipative Friction in a Gravity Field

Asymptotically stable point: If we move the bearing to a different position, it will eventually settle back to the bottom of the trough.

Stable in the sense of Lyapunov point: If we place the bearing in a different position, it will not move back but will not roll farther away from the center point.

Unstable equilibrium point: If the bearing is given the slightest disturbance, it will roll down the hill.

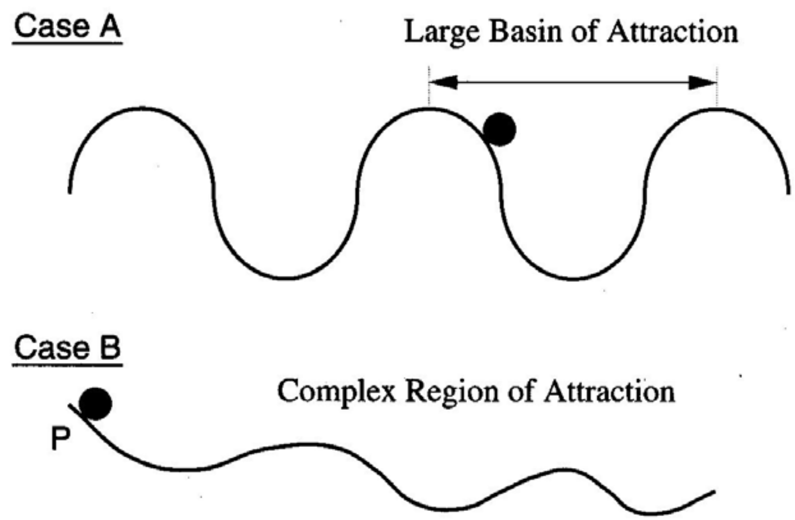


Figure 17.2-2 Basins of Attraction

17.2.1 Definitions

An equilibrium point; a point **a*** where the derivative is zero.

Definition 1: **Stability (in the sense of Lyapunov)**

The origin, $\mathbf{a}^* = \mathbf{0}$, is a stable equilibrium point if for any given value $\varepsilon > 0$ there exists a number $\delta(\varepsilon) > 0$ such that if $\|\mathbf{a}(0)\| < \delta$, then the resulting motion $\mathbf{a}(t)$ satisfies $\|\mathbf{a}(t)\| < \varepsilon$ for $t > 0$.

Figure 17.2.1-1 Stable (with Friction) and Unstable (without Friction)

Definition 2: **Asymptotic Stability**

The origin is an asymptotically stable equilibrium point if there exists a number $\delta > 0$ such that whenever $\|\mathbf{a}(0)\| < \delta$ the resulting motion satisfies $\|\mathbf{a}(t)\| \to 0$ as $t \to \infty$.

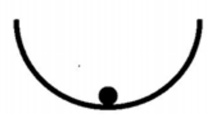


Figure 17.2.1-2 Asymptotic Stable (with Friction), Stable (without Friction)

Definition 3: **Positive Definite**

A scalar function $V(\mathbf{a})$ is positive definite if $V(\mathbf{0}) = 0$ and $V(\mathbf{a}) > 0$ for $\mathbf{a} \neq 0$.

Definition 4: **Positive Semidefinite**

A scalar function $V(\mathbf{a})$ is positive semidefinite if $V(\mathbf{a}) \geq 0$ for all $\mathbf{a}$.

17.3 Lyapunov Stability Theorem

Consider the autonomous (unforced, no explicit time dependence) system,

$$\frac{d\mathbf{a}}{dt} = \mathbf{g}(\mathbf{a}) \tag{17.3-1}$$

Theorem 1: **Lyapunov Stability Theorem**

If a positive definite function $V(\mathbf{a})$ can be found such that $dV(\mathbf{a})/dt$ is negative semidefinite, then the origin ($\mathbf{a} = \mathbf{0}$) is stable for the system of (17.3-1). If a positive definite function $V(\mathbf{a})$ can be found such that $dV(\mathbf{a})/dt$ is negative definite, then the origin ($\mathbf{a} = \mathbf{0}$) is asymptotically stable. In each case, V is called a Lyapunov function of the system.

17.4 Pendulum Example

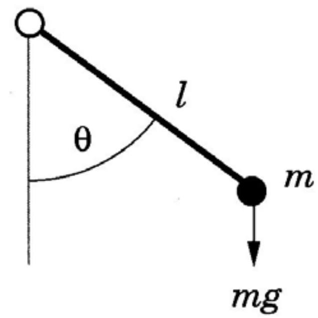


Figure 17.4-1 Pendulum

Using Newton's second law ($F = ma$),

$$ml\frac{d^2}{dt^2}(\theta) = -c\frac{d\theta}{dt} - mg\sin(\theta) \quad (17.4\text{-}1)$$

$$ml\frac{d^2\theta}{dt^2} + c\frac{d\theta}{dt} + mg\sin(\theta) = 0 \quad (17.4\text{-}2)$$

where θ: the angle of the pendulum, *m*: the mass of the pendulum, *l*: the length of the pendulum, *c*: the damping coefficient, and *g*: the gravitational constant.

State variables,

$$a_1 = \theta \text{ and } a_2 = \frac{d\theta}{dt} \tag{17.4-3}$$

$$\frac{da_1}{dt} = a_2 \tag{17.4-4}$$

$$\frac{da_2}{dt} = -\frac{g}{l}\sin(a_1) - \frac{c}{ml}a_2 \tag{17.4-5}$$

Consider the stability of the origin (**a = 0**),

$$\frac{da_1}{dt} = a_2 = 0 \tag{17.4-6}$$

$$\frac{da_2}{dt} = -\frac{g}{l}\sin(a_1) - \frac{c}{ml}a_2 = -\frac{g}{l}\sin(0) - \frac{c}{ml}(0) = 0 \tag{17.4-7}$$

- The origin is an equilibrium point.

Total (kinetic and potential) energy of the pendulum

$$V(\mathbf{a}) = \frac{1}{2}ml^2(a_2)^2 + mgl(1-\cos(a_1)) \tag{17.4-8}$$

$$\frac{d}{dt}V(\mathbf{a}) = [\nabla V(\mathbf{a})]^T g(\mathbf{a}) = \frac{\partial V}{\partial a_1}\left(\frac{da_1}{dt}\right) + \frac{\partial V}{\partial a_2}\left(\frac{da_2}{dt}\right) \tag{17.4-9}$$

$$\frac{d}{dt}V(\mathbf{a}) = (mgl\sin(a_1))a_2 + (ml^2a_2)\left(-\frac{g}{l}\sin(a_1) - \frac{c}{ml}a_2\right) = -cl(a_2)^2 \le 0 \tag{17.4-10}$$

- $dV(\mathbf{a})/dt$ is negative semidefinite.
- From Lyapunov's theorem, then, we know that the origin is a stable point.
- However, we cannot say, from the theorem and this Lyapunov function, that the origin is asymptotically stable.
- As long as the pendulum has friction, it will eventually settle in a vertical position, the origin is asymptotically stable.

When $g = 9.8$, $m = 1$, $1 = 9.8$, $c = 1.96$,

$$\frac{da_1}{dt} = a_2 \quad (17.4\text{-}11)$$

$$\frac{da_2}{dt} = -\sin(a_1) - 0.2a_2 \quad (17.4\text{-}12)$$

$$V = (9.8)^2\left[\frac{1}{2}(a_2)^2 + (1 - \cos(a_1))\right] \quad (17.4\text{-}13)$$

$$\frac{dV}{dt} = -(19.208)(a_2)^2 \quad (17.4\text{-}14)$$

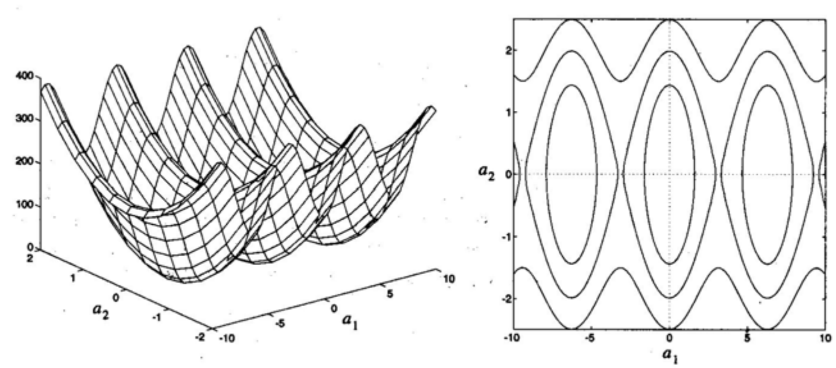


Figure 17.4-2 Pendulum Energy Surface, Three Possible Minimum Points of the Energy Surface at 0 and ±2π

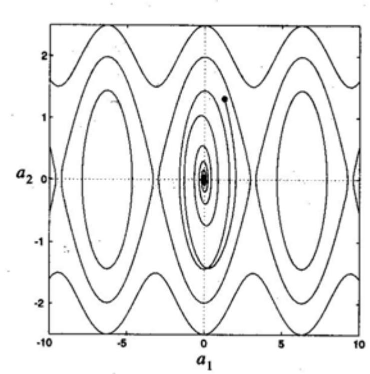


Figure 17.4-3 Pendulum Response on State Variable Plane, $a_1(0)$ = 1.3 radians (74°), $a_2(0)$ = 1.3 radians per second

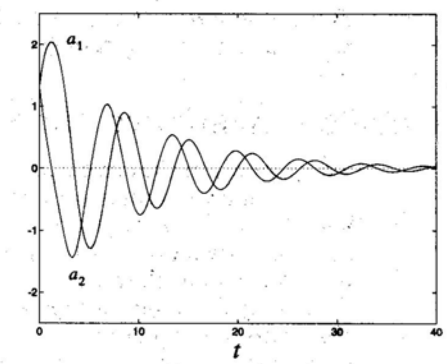


Figure 17.4-4 State Variables vs. Time

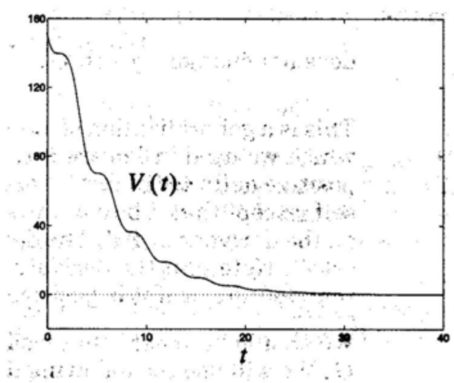


Figure 17.4-5 Pendulum Lyapunov Function (Energy) vs. Time

17.5 Lasalle's Invariance Theorem

Definition 5: **Lyapunov Function**

Let V be a continuously differentiable function from R^n to R. If G is any subset of R^n, we say that V is a Lyapunov function on G for the system $d\mathbf{a}/dt = \mathbf{g}(\mathbf{a})$ if

$$\frac{dV(\mathbf{a})}{dt} = (\nabla V(\mathbf{a}))^T \mathbf{g}(\mathbf{a}) \qquad (17.5\text{-}1)$$

does not change sign on G.

Definition 6: **Set *Z***

$$Z = \{\mathbf{a} : dV(\mathbf{a})/dt = 0, \mathbf{a}\_in\_the\_closure\_of\_G\} \qquad (17.5\text{-}2)$$

Definition 7: **Invariant Set**

A set of points in R^n is invariant with respect to $d\mathbf{a}/dt = \mathbf{g}(\mathbf{a})$ if every solution of $d\mathbf{a}/dt = \mathbf{g}(\mathbf{a})$ starting in that set remains in the set for all time.

Definition 8: **Set *L***

L is defined as the largest invariant set in Z.

- If L has only one stable point, then that point is asymptotically stable.

Theorem 2: **Lasalle's Invariance Theorem**

If V is a Lyapunov function on G for $d\mathbf{a}/dt = \mathbf{g}(\mathbf{a})$, then each solution $\mathbf{a}(t)$ that remains in G for all $t > 0$ approaches $L^\circ = L \cap G$ as $t \to \infty$. (G is a basin of attraction for L, which has all of the stable points.) If all trajectories are bounded, then $\mathbf{a}(t) \to L$ as $t \to \infty$.

- If a trajectory stays in G, then it will either converge to L, or it will go to infinity. If all trajectories are bounded, then all trajectories will converge to L.

Corollary 1: **Lasalle's Corollary**

Let G be a component (one connected subset) of

$$\Omega_\eta = \{\mathbf{a} : V(\mathbf{a}) < \eta\} \quad (17.5\text{-}3)$$

- Assume that G is bounded, $dV(\mathbf{a})/dt \le 0$ on the set G, and let the set $L^\circ = closure(L \cap G)$ be a subset of G. Then L° is an attractor, and G is in its region of attraction.

17.5.1 Example

$\eta = 100$,

$$\Omega_{100} = \{\mathbf{a} : V(\mathbf{a}) \le 100\} \tag{17.5.1-1}$$

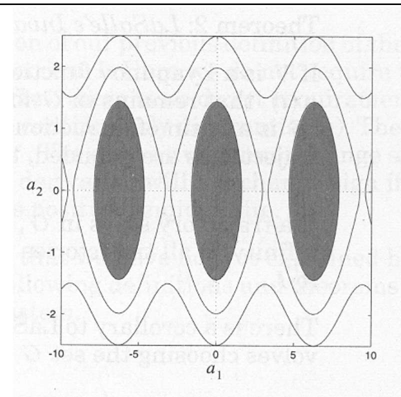


Figure 17.5.1-1 Illustration of the Set Ω_{100}

By choosing the component of Ω_{100} that contains $\mathbf{a} = \mathbf{0}$.

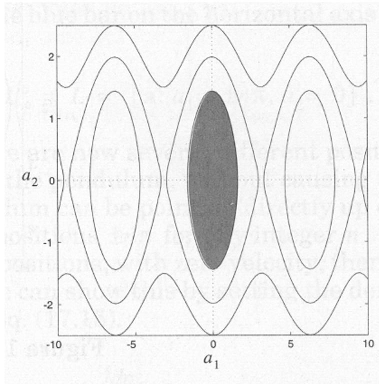


Figure 17.5.1-2 Illustration of the Set G

$$Z = \{\mathbf{a} : dV(\mathbf{a})/dt = 0, \mathbf{a}\_in\_the\_closure\_of\_G\} = \{\mathbf{a} : a_2 = 0, \mathbf{a}\_in\_the\_closure\_of\_G\} \quad (17.5.1\text{-}2)$$

$$Z = \{\mathbf{a} : a_2 = 0, -1.6 \le a_1 \le 1.6\} \quad (17.5.1\text{-}3)$$

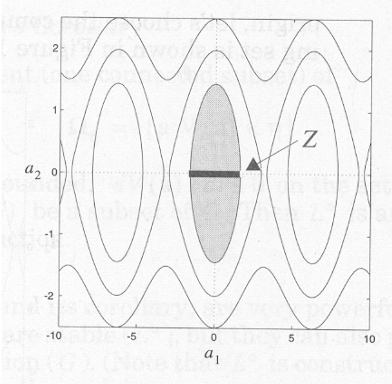


Figure 17.5.1-3 Illustration of the Set Z

$$L = \{\mathbf{a} : \mathbf{a} = 0\} \quad (17.5.1\text{-}4)$$

$$L^\circ = closure(L \cap G) = L = \{\mathbf{a} : \mathbf{a} = 0\}\,. \quad (17.5.1\text{-}5)$$

- L° is an attractor (asymptotically stable point) in G, its region of attraction.
- Any trajectory that starts in G will decay to the origin.

$$\Omega_{300} = \{\mathbf{a} : V(\mathbf{a}) \le 300\} \tag{17.5.1-6}$$

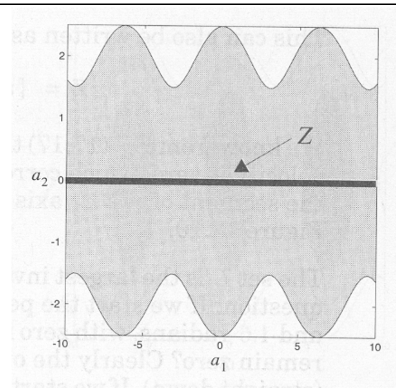


Figure 17.5.1-4 Illustration of $G = \Omega_{300}$ and Z

$G = \Omega_{300}$, since Ω_{300} has only one component.

$$Z = \{\mathbf{a} : a_2 = 0\} \tag{17.5.1-7}$$

$$L^{\circ} = L = \{\mathbf{a} : a_1 = \pm n\pi, a_2 = 0\} \tag{17.5.1-8}$$

$$\frac{da_1}{dt} = a_2 = 0 \tag{17.5.1-9}$$

$$\left(\frac{da_2}{dt} = -\sin(a_1) - 0.2a_2 = -\sin(a_1)\right) \Rightarrow (a_1 = \pm n\pi) \tag{17.5.1-10}$$

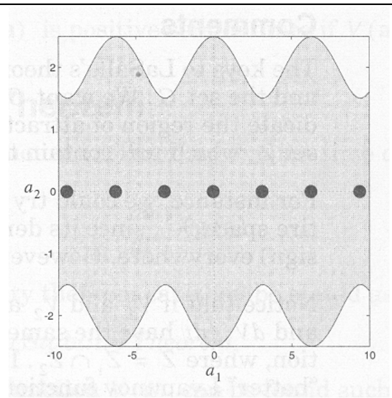


Figure 17.5.1-5 The Set L°

18 Hopfield Network

18.1 Hopfield Model

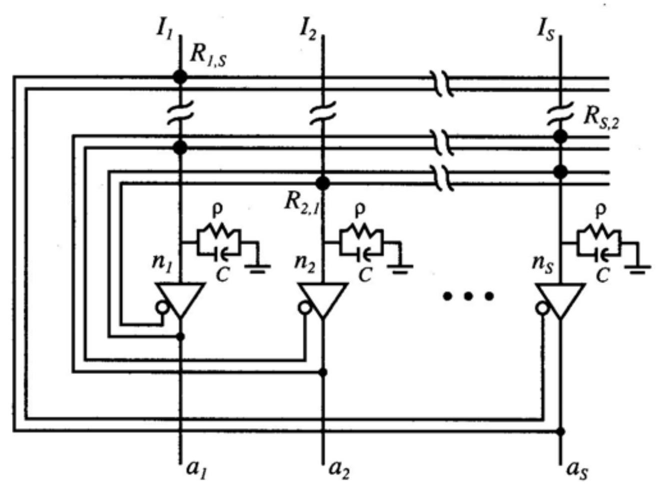


Figure 18.1-1 Hopfield Model

$$C\frac{dn_i(t)}{dt} = \sum_{j=1}^{S}\frac{a_j(t)-n_i}{R_{ij}} - \frac{n_i(t)}{\rho} + I_i = \sum_{j=1}^{S} T_{i,j}a_j(t) - \frac{n_i(t)}{R_i} + I_i \tag{18.1-1}$$

$$\left|T_{i,j}\right| = \frac{1}{R_{i,j}}, \frac{1}{R_i} = \frac{1}{\rho} + \sum_{j=1}^{S}\frac{1}{R_{i,j}}, n_i = f^{-1}(a_i) \tag{18.1-2}$$

By multiplying with R_i,
$$R_i C \frac{dn_i(t)}{dt} = \sum_{j=1}^{S} R_i T_{i,j} a_j(t) - n_i(t) + R_i I_i \tag{18.1-3}$$

Defining,
$$\varepsilon = R_i C, w_{i,j} = R_i T_{i,j}, b_i = R_i I_i \tag{18.1-4}$$

$$\varepsilon \frac{dn_i(t)}{dt} = -n_i(t) + \sum_{j=1}^{S} w_{i,j} a_j(t) + b_i \tag{18.1-5}$$

$$\varepsilon \frac{d\mathbf{n}(t)}{dt} = -\mathbf{n}(t) + \mathbf{W}\mathbf{a}(t) + \mathbf{b} \tag{18.1-6}$$

$$\mathbf{a}(t) = \mathbf{f}(\mathbf{n}(t)) \tag{18.1-7}$$

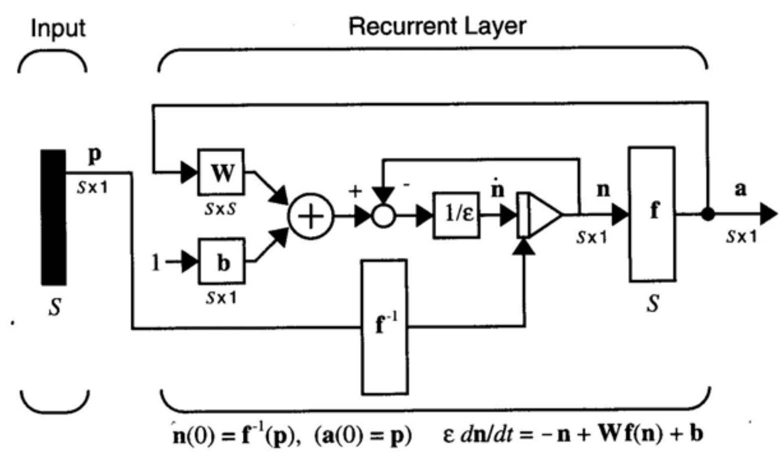


Figure 18.1-2 Hopfield Network

18.2 Lyapunov Function

Lyapunov function in Lasalle's theorem of Hopfield network,

$$V(\mathbf{a}) = -\frac{1}{2}\mathbf{a}^T\mathbf{W}\mathbf{a} + \sum_{i=1}^{S}\left\{\int_0^{a_i} f^{-1}(u)du\right\} - \mathbf{b}^T\mathbf{a} \tag{18.2-1}$$

$$\frac{d}{dt}\{-\frac{1}{2}\mathbf{a}^T\mathbf{W}\mathbf{a}\} = -\frac{1}{2}\nabla[\mathbf{a}^T\mathbf{W}\mathbf{a}]^T\frac{d\mathbf{a}}{dt} = -[\mathbf{W}\mathbf{a}]^T\frac{d\mathbf{a}}{dt} = -\mathbf{a}^T\mathbf{W}\frac{d\mathbf{a}}{dt} \tag{18.2-2}$$

$$\frac{d}{dt}\left\{\int_0^{a_i} f^{-1}(u)du\right\} = \frac{d}{da_i}\left\{\int_0^{a_i} f^{-1}(u)du\right\}\frac{da_i}{dt} = f^{-1}(a_i)\frac{da_i}{dt} = n_i\frac{da_i}{dt} \tag{18.2-3}$$

$$\frac{d}{dt}\left[\sum_{i=1}^{S}\left\{\int_0^{a_i} f^{-1}(u)du\right\}\right] = \mathbf{n}^T\frac{d\mathbf{a}}{dt} \tag{18.2-4}$$

$$\frac{d}{dt}\{-\mathbf{b}^T\mathbf{a}\} = -\nabla[\mathbf{b}^T\mathbf{a}]^T\frac{d\mathbf{a}}{dt} = -\mathbf{b}^T\frac{d\mathbf{a}}{dt} \tag{18.2-5}$$

$$\frac{d}{dt}V(\mathbf{a}) = -\mathbf{a}^T\mathbf{W}\frac{d\mathbf{a}}{dt} + \mathbf{n}^T\frac{d\mathbf{a}}{dt} - \mathbf{b}^T\frac{d\mathbf{a}}{dt} = [-\mathbf{a}^T\mathbf{W} + \mathbf{n}^T - \mathbf{b}^T]\frac{d\mathbf{a}}{dt} \tag{18.2-6}$$

$$[-\mathbf{a}^T\mathbf{W} + \mathbf{n}^T - \mathbf{b}^T] = -\varepsilon\left[\frac{d\mathbf{n}(t)}{dt}\right]^T \tag{18.2-7}$$

$$\frac{d}{dt}V(\mathbf{a}) = -\varepsilon\left[\frac{d\mathbf{n}(t)}{dt}\right]^T\frac{d\mathbf{a}}{dt} = -\varepsilon\sum_{i=1}^{S}\left(\frac{dn_i}{dt}\right)\left(\frac{da_i}{dt}\right) \tag{18.2-8}$$

$$\frac{dn_i}{dt} = \frac{d}{dt}[f^{-1}(a_i)] = \frac{d}{da_i}[f^{-1}(a_i)]\frac{da_i}{dt} \tag{18.2-9}$$

$$\frac{d}{dt}V(\mathbf{a}) = -\varepsilon\sum_{i=1}^{S}\left(\frac{dn_i}{dt}\right)\left(\frac{da_i}{dt}\right) = -\varepsilon\sum_{i=1}^{S}\left(\frac{d}{da_i}[f^{-1}(a_i)]\right)\left(\frac{da_i}{dt}\right)^2 \tag{18.2-10}$$

If $f^{-1}(a_i)$ is an increasing function,

$$\frac{d}{da_i}[f^{-1}(a_i)] > 0 \tag{18.2-11}$$

$$\frac{d}{dt}V(\mathbf{a}) \le 0 \tag{18.2-12}$$

- If $f^{-1}(a_i)$ is an increasing function, $dV(\mathbf{a})/dt$ is a negative semidefinite function. $V(\mathbf{a})$ is a valid Lyapunov function.

18.2.1 Invariant Sets

$$G \subset R^S \tag{18.2.1-1}$$

$$Z = \{\mathbf{a} : dV(\mathbf{a})/dt = 0, \mathbf{a}\_in\_the\_closure\_of\_G\} \quad (18.2.1\text{-}2)$$

$$Z = \{\mathbf{a} : \frac{d\mathbf{a}}{dt} = \mathbf{0}\}\text{, which is set of equilibrium points} \quad (18.2.1\text{-}2)$$

$$L = Z \tag{18.2.1-3}$$

- All points in Z are potential attractors.

18.2.2 Example

$$a = f(n) = \frac{2}{\pi}\tan^{-1}\left(\frac{\gamma\pi n}{2}\right) \tag{18.2.2-1}$$

$$R_{1,1} = R_{2,2} = \rho_1 = \rho_2 = 0, R_{1,2} = R_{2,1} = 1, C_1 = C_2 = 1 \tag{18.2.2-2}$$

$$\mathbf{W} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{18.2.2-3}$$

$$\varepsilon = R_i C = 1 \tag{18.2.2-4}$$

With $\gamma = 1.4$ and $I_1 = I_2 = 0$,

$$\mathbf{b} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{18.2.2-5}$$

The Lyapunov function,

$$V(\mathbf{a}) = -\frac{1}{2}\mathbf{a}^T\mathbf{W}\mathbf{a} + \sum_{i=1}^{S}\left\{\int_0^{a_i} f^{-1}(u)du\right\} - \mathbf{b}^T\mathbf{a} \tag{18.2.2-6}$$

$$-\frac{1}{2}\mathbf{a}^T\mathbf{W}\mathbf{a} = -\frac{1}{2}\begin{bmatrix} a_1 & a_2 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = -a_1 a_2 \tag{18.2.2-7}$$

$$\int_0^{a_i} f^{-1}(u)du = \frac{2}{\gamma\pi}\int_0^{a_i}\tan\left(\frac{\pi}{2}u\right)du = \frac{2}{\gamma\pi}\left[-\log\left[\cos\left(\frac{\pi}{2}u\right)\right]\frac{2}{\pi}\right]_0^{a_i} \tag{18.2.2-8}$$

$$\int_0^{a_i} f^{-1}(u)du = -\frac{4}{\gamma\pi^2}\log\left[\cos\left(\frac{\pi}{2}a_i\right)\right] \tag{18.2.2-9}$$

$$V(\mathbf{a}) = -a_1a_2 - \frac{4}{1.4\pi^2}\left[\log\{\cos\frac{\pi}{2}a_1\} + \log\{\cos\frac{\pi}{2}a_2\}\right] \tag{18.2.2-10}$$

$$\frac{d\mathbf{n}}{dt} = -\mathbf{n} + \mathbf{Wf}(\mathbf{n}) = -\mathbf{n} + \mathbf{Wa} \tag{18.2.2-11}$$

$$\frac{dn_1}{dt} = a_2 - n_1 \tag{18.2.2-12}$$

$$\frac{dn_2}{dt} = a_1 - n_2 \tag{18.2.2-13}$$

$$a_1 = \frac{2}{\pi}\tan^{-1}\left(\frac{1.4\pi}{2}n_1\right) \tag{18.2.2-14}$$

$$a_2 = \frac{2}{\pi}\tan^{-1}\left(\frac{1.4\pi}{2}n_2\right) \tag{18.2.2-15}$$

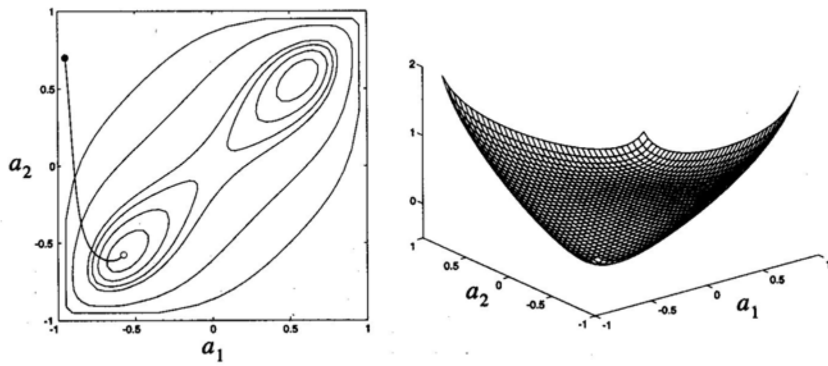


Figure 18.2.2-1 Hopfield Example Lyapunov Function and Trajectory

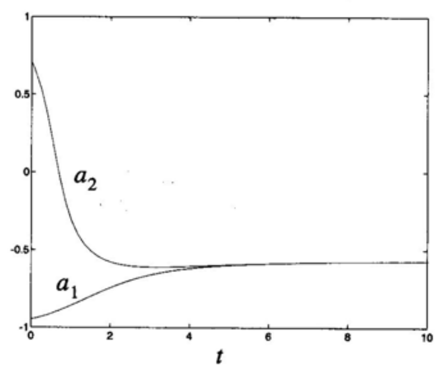


Figure 18.2.2-2 Hopfield Example Time Response

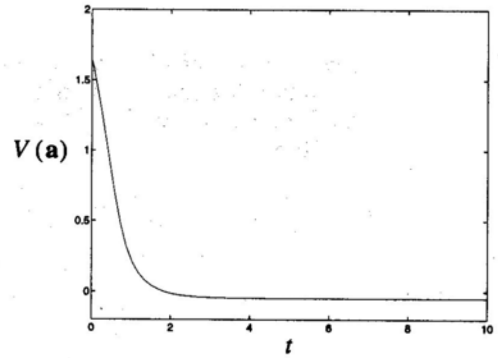


Figure 18.2.2-3 Lyapunov Function Response

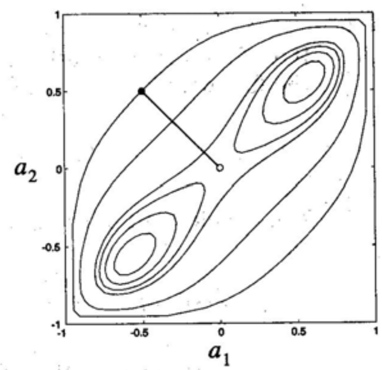


Figure 18.2.2-4 Hopfield Convergence to a Saddle Point

18.2.3 Hopfield Attractors

The potential attractors of the Hopfield network are equilibrium points.

$$\frac{d\mathbf{a}}{dt} = \mathbf{0} \tag{18.2.3-1}$$

The minima of a function must be stationary points. The stationary points of $V(\mathbf{a})$,

$$\nabla V = \left[\frac{\partial V}{\partial a_1} \quad \frac{\partial V}{\partial a_2} \quad \cdots \quad \frac{\partial V}{\partial a_S} \right] = \mathbf{0} \tag{18.2.3-2}$$

$$V(\mathbf{a}) = -\frac{1}{2}\mathbf{a}^T \mathbf{W} \mathbf{a} + \sum_{i=1}^{S} \left\{ \int_0^{a_i} f^{-1}(u) du \right\} - \mathbf{b}^T \mathbf{a} \tag{18.2.3-3}$$

$$\nabla V(\mathbf{a}) = [-\mathbf{W}\mathbf{a} + \mathbf{n} - \mathbf{b}] = -\varepsilon \left[\frac{d\mathbf{n}(t)}{dt} \right] \tag{18.2.3-4}$$

$$\frac{\partial}{\partial a_i} V(\mathbf{a}) = -\varepsilon \frac{dn_i}{dt} = -\varepsilon \frac{d}{dt}([f^{-1}(a_i)]) = -\varepsilon \frac{d}{da_i}[f^{-1}(a_i)] \frac{da_i}{dt} \tag{18.2.3-5}$$

If $f^{-1}(a)$ is linear,

$$\frac{d\mathbf{a}}{dt} = -\alpha \nabla V(\mathbf{a}) \tag{18.2.3-6}$$

- The response of the Hopfield network is steepest descent.
- If you are in a region where $f^{-1}(a)$ is approximately linear, the network solution approximates steepest descent.

For an increasing function,

$$\frac{d}{dt}[f^{-1}(a_i)] > 0 \tag{18.2.3-7}$$

The points for which

$$\frac{d\mathbf{a}(t)}{dt} = \mathbf{0} \tag{18.2.3-8}$$

are also the points where

$$\nabla V(\mathbf{a}) = \mathbf{0} \tag{18.2.3-9}$$

- The attractors, which are members of the set L, will also be stationary points of the Lyapunov function $V(\mathbf{a})$.

18.3 Effect of Gain

$$a = f(n) = \frac{2}{\pi}\tan^{-1}\left(\frac{\gamma\pi n}{2}\right) \quad (18.3\text{-}1)$$

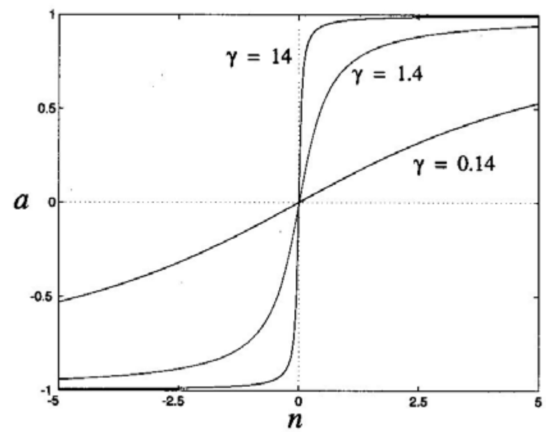


Figure 18.3-1 Inverse Tangent Amplifier Characteristic

- As γ goes to infinity, $f(n)$ approaches a signum (step, hardlims) function.

The general Lyapunov function,

$$V(\mathbf{a}) = -\frac{1}{2}\mathbf{a}^T\mathbf{W}\mathbf{a} + \sum_{i=1}^{S}\left\{\int_0^{a_i} f^{-1}(u)du\right\} - \mathbf{b}^T\mathbf{a} \quad (18.3\text{-}2)$$

$$f^{-1}(u) = \frac{2}{\gamma\pi}\tan\left(\frac{\pi u}{2}\right) \tag{18.3-3}$$

$$\int_0^{a_i} f^{-1}(u)du = \frac{2}{\gamma\pi}\left[\frac{2}{\pi}\log\left(\cos\left(\frac{\pi a_i}{2}\right)\right)\right] = -\frac{4}{\gamma\pi^2}\log\left(\cos\left(\frac{\pi a_i}{2}\right)\right) \tag{18.3-4}$$

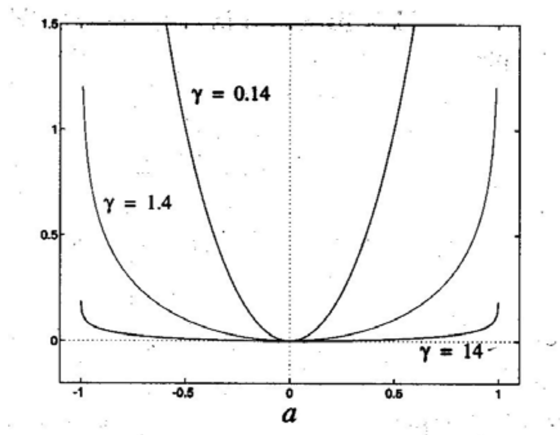


Figure 18.3-2 Second Term in the Lyapunov Function

- As γ increases the function flattens and is close to 0 most of the time.
- As the gain γ goes to infinity, the integral in the second term of the Lyapunov function will be close to zero in the range -1 < a_i < 1.

The high-gain Lyapunov function,

$$V(\mathbf{a}) = -\frac{1}{2}\mathbf{a}^T\mathbf{W}\mathbf{a} - \mathbf{b}^T\mathbf{a} \tag{18.3-5}$$

$$V(\mathbf{a}) = -\frac{1}{2}\mathbf{a}^T\mathbf{W}\mathbf{a} - \mathbf{b}^T\mathbf{a} = \frac{1}{2}\mathbf{a}^T\mathbf{A}\mathbf{a} + \mathbf{d}^T\mathbf{a} + c \tag{18.3-6}$$

$$\nabla^2 V(\mathbf{a}) = \mathbf{A} = -\mathbf{W}, \mathbf{d} = -\mathbf{b}, c = 0 \tag{18.3-7}$$

$$\nabla^2 V(\mathbf{a}) = -\mathbf{W} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \tag{18.3-8}$$

$$\left|\nabla^2 V(\mathbf{a}) - \lambda I\right| = \begin{vmatrix} -\lambda & -1 \\ -1 & -\lambda \end{vmatrix} = \lambda^2 - 1 = (\lambda + 1)(\lambda - 1) \tag{18.3-9}$$

The eigenvalues are $\lambda_1 = -1$ and $\lambda_2 = 1$. The eigenvectors are

$$\mathbf{z}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{z}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{18.3-10}$$

There will be constrained minima at the two corners of the hypercube

$$\mathbf{a} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \text{ and } \mathbf{a} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \tag{18.3-11}$$

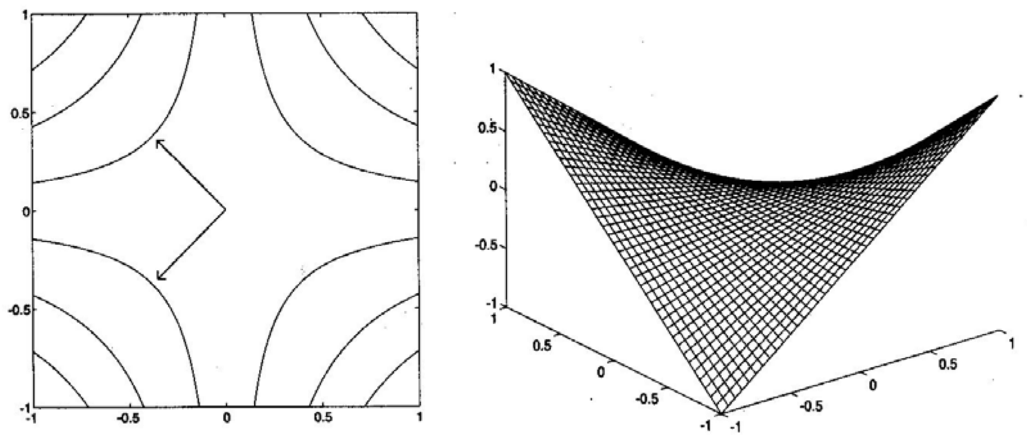


Figure 18.3-3 Example High Gain Lyapunov Function

- When the gain is very small, there is a single minimum at the origin. As the gain is increased, two minima move out from the origin toward the two corners given.

18.4 Hopfield Design

- The Hopfield network does not have a learning law associated with it. It is not trained, nor does it learn on its own.
- A design procedure based on the Lyapunov function is used to determine the weight matrix.

The high-gain Lyapunov function,

$$V(\mathbf{a}) = -\frac{1}{2}\mathbf{a}^T\mathbf{W}\mathbf{a} - \mathbf{b}^T\mathbf{a} \quad (18.4\text{-}1)$$

- The Hopfield design technique is to choose the weight matrix **W** and the bias vector **b** so that *V* takes on the form of a function to be minimized.

18.4.1 Content-Addressable Memory

- When an input pattern is presented to the network, the network should produce the stored pattern that most closely resembles the input pattern.
- The initial network output is assigned to the input pattern. The network output should then converge to the prototype pattern closest to the input pattern.

The prototype patterns,

$$\{\mathbf{p}_1, \mathbf{p}_2, \ldots \mathbf{p}_Q\} \quad (18.4.1\text{-}1)$$

- Each of these vectors consists of S elements, having the values 1 or -1.
- $Q << S$.

Quadratic performance index,

$$J(\mathbf{a}) = -\frac{1}{2}\sum_{q=1}^{Q}([\mathbf{p}_q]^T\mathbf{a})^2 \tag{18.4.1-2}$$

- If the elements of the vectors **a** are restricted to be ±1, this function is minimized at the prototype patterns.

When the prototype patterns are orthogonal, the performance index at one of the prototype patterns,

$$\mathbf{J}(\mathbf{p}_j) = -\frac{1}{2}\sum_{q=1}^{Q}([\mathbf{p}_q]^T\mathbf{p}_j)^2 = -\frac{1}{2}([\mathbf{p}_j]^T\mathbf{p}_j)^2 = -\frac{S^2}{2} \tag{18.4.1-3}$$

- $J(\mathbf{a})$ will be largest (least negative) when **a** is not close to any prototype pattern, and will be smallest (most negative) when **a** is equal to any one of the prototype patterns.

When the weight matrix applies the supervised Hebb rule (with target patterns being the same as input patterns) as

$$\mathbf{W} = \sum_{q=1}^{Q}\mathbf{p}_q(\mathbf{p}_q)^T \tag{18.4.1-4}$$

$$\mathbf{b} = \mathbf{0} \tag{18.4.1-5}$$

The Lyapunov function,

$$V(\mathbf{a}) = -\frac{1}{2}\mathbf{a}^T\mathbf{W}\mathbf{a} = -\frac{1}{2}\mathbf{a}^T\left[\sum_{q=1}^{Q}\mathbf{p}_q(\mathbf{p}_q)^T\right]\mathbf{a} = -\frac{1}{2}\sum_{q=1}^{Q}\mathbf{a}^T\mathbf{p}_q(\mathbf{p}_q)^T\mathbf{a} \tag{18.4.1-6}$$

$$V(\mathbf{a}) = -\frac{1}{2}\sum_{q=1}^{Q}([\mathbf{p}_q]^T \mathbf{a})^2 = J(\mathbf{a}) \qquad (18.4.1\text{-}7)$$

- The Lyapunov function is indeed equal to the quadratic performance index for the content-addressable memory problem.
- The Hopfield network output will tend to converge to the stored prototype patterns.
- If there is significant correlation between the prototype patterns, the supervised Hebb rule does not work well. In that case the pseudoinverse technique has been suggested.
- In the best situation, where the prototype patterns are orthogonal, every prototype pattern will be an equilibrium point of the network. However, there will be many other equilibrium points as well. The network may well converge to a pattern that is not one of the prototype patterns.
- A general rule is that, when using the Hebb rule, the number of stored patterns can be no more than 15% of the number of neurons.

18.4.2 Hebb Rule

When the Hebb rule is used to compute the weight matrix in Hopfield network and the prototype patterns are orthogonal,

$$\mathbf{W} = \sum_{q=1}^{Q} \mathbf{p}_q (\mathbf{p}_q)^T \tag{18.4.2-1}$$

When a prototype vector $\mathbf{p}_j$ is applied to the network,

$$\mathbf{W}\mathbf{p}_j = \sum_{q=1}^{Q} \mathbf{p}_q (\mathbf{p}_q)^T \mathbf{p}_j = \mathbf{p}_j (\mathbf{p}_j)^T \mathbf{p}_j = S\mathbf{p}_j \tag{18.4.2-2}$$

$$\mathbf{W}\mathbf{p}_j = \lambda \mathbf{p}_j \tag{18.4.2-3}$$

- Each prototype vector is an eigenvector of the weight matrix and they have a common eigenvalue of $\lambda = S$.

The eigenspace X for the eigenvalue $\lambda = S$,

$$X = \text{span}\{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_Q\} \tag{18.4.2-4}$$

- The vector, **a**, that is a linear combination of the prototype vectors is an eigenvector.

$$\begin{aligned}
\mathbf{W}\mathbf{a} &= \mathbf{W}\{\alpha_1 \mathbf{p}_1 + \alpha_2 \mathbf{p}_2 + \ldots + \alpha_Q \mathbf{p}_Q\} \\
&= \{\alpha_1 \mathbf{W}\mathbf{p}_1 + \alpha_2 \mathbf{W}\mathbf{p}_2 + \ldots + \alpha_Q \mathbf{W}\mathbf{p}_Q\} \\
&= \{\alpha_1 S\mathbf{p}_1 + \alpha_2 S\mathbf{p}_2 + \ldots + \alpha_Q S\mathbf{p}_Q\} \\
&= S\{\alpha_1 \mathbf{p}_1 + \alpha_2 \mathbf{p}_2 + \ldots + \alpha_Q \mathbf{p}_Q\} = S\mathbf{a}
\end{aligned} \tag{18.4.2-5}$$

The entire space R^S can be divided into two disjoint sets,

$$R^S = X \cup X^{\perp} \qquad (18.4.2\text{-}6)$$

where $X^{\perp}$: the orthogonal complement of X.

- Every vector in $X^{\perp}$ is orthogonal to every vector in X.

$\mathbf{a} \in X^{\perp}$,

$$(\mathbf{p}_q)^T \mathbf{a} = 0, q = 1,2,\ldots,Q \qquad (18.4.2\text{-}7)$$

$$\mathbf{Wa} = \sum_{q=1}^{Q} \mathbf{p}_q (\mathbf{p}_q)^T \mathbf{a} = \sum_{q=1}^{Q} (\mathbf{p}_q \cdot 0) = \mathbf{0} = 0 \cdot \mathbf{a} \qquad (18.4.2\text{-}8)$$

- $X^{\perp}$ defines an eigenspace for the repeated eigenvalue $\lambda = 0$.

Summary

- The weight matrix has two eigenvalues, S and 0.
- The eigenspace for the eigenvalue S is the space spanned by the prototype vectors.
- The eigenspace for the eigenvalue 0 is the orthogonal complement of the space spanned by the prototype vectors.

Hessian matrix for the high-gain Lyapunov function V,

$$\nabla^2 V = -\mathbf{W} \qquad (18.4.2\text{-}9)$$

- The eigenvalues for $\nabla^2 V$ will be -S and 0.

- The high-gain Lyapunov function is a quadratic function.
- The first eigenvalue is negative, V will have negative curvature in X.
- The second eigenvalue is zero, V will have zero curvature in $X^{\perp}$.
- Because V has negative curvature in X, the trajectories of the Hopfield network will tend to fall into the corners of the hypercube $\{\mathbf{a}:-1<a_i< 1\}$ that are contained in X.
- By using the Hebb rule, there will be at least two minima of the Lyapunov function for each prototype vector.
- If $\mathbf{p}_q$ is a prototype vector, then $-\mathbf{p}_q$ will also be in the space spanned by the prototype vectors, X.
- The negative of each prototype vector will be one of the comers of the hypercube $\{\mathbf{a}:-1<a_i< 1\}$ that are contained in X. There will also be a number of other minima of the Lyapunov function that do not correspond to prototype patterns.
- The minima of V are in the corners of the hypercube $\{\mathbf{a}:-1<a_i< 1\}$ that are contained in X.
- These corners will include the prototype patterns, and also some linear combinations of the prototype patterns.
- The minima that are not prototype patterns are often referred to as spurious patterns.

Example:

When $R_{1,1} = R_{2,2} = \rho_1 = \rho_2 = 0, R_{1,2} = R_{2,1} = 1, C_1 = C_2 = 1$,

$$\mathbf{W}' = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{18.4.2-10}$$

An attractor by using high gain Lyapunov function,

$$\mathbf{p}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{18.4.2-11}$$

By using the Hebb rule with one prototype pattern at the attractor,

$$\mathbf{W} = \mathbf{p}_1(\mathbf{p}_1)^T = \begin{bmatrix} 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{18.4.2-12}$$

$$\mathbf{W}' = \mathbf{W} - \mathbf{I} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{18.4.2-13}$$

The high-gain Lyapunov function,

$$V(\mathbf{a}) = -\frac{1}{2}\mathbf{a}^T\mathbf{W}\mathbf{a} = -\frac{1}{2}\mathbf{a}^T\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\mathbf{a} \tag{18.4.2-14}$$

$$\nabla^2 V(\mathbf{a}) = -\mathbf{W} = \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix} \tag{18.4.2-15}$$

Its eigenvalues,

$$\lambda_1 = -S = -2 \text{ and } \lambda_2 = 0 \tag{18.4.2-16}$$

The corresponding eigenvectors,

$$\mathbf{z}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \text{ and } \mathbf{z}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{18.4.2-17}$$

The first eigenvector, corresponding to the eigenvalue -*S*, represents the space spanned by the prototype vector:

$$X = \{\mathbf{a} : a_1 = a_2\} \tag{18.4.2-18}$$

The second eigenvector, corresponding to the eigenvalue 0, represents the orthogonal complement of the first eigenvector:

$$X^{\perp} = \{\mathbf{a} : a_1 = -a_2\} \tag{18.4.2-19}$$

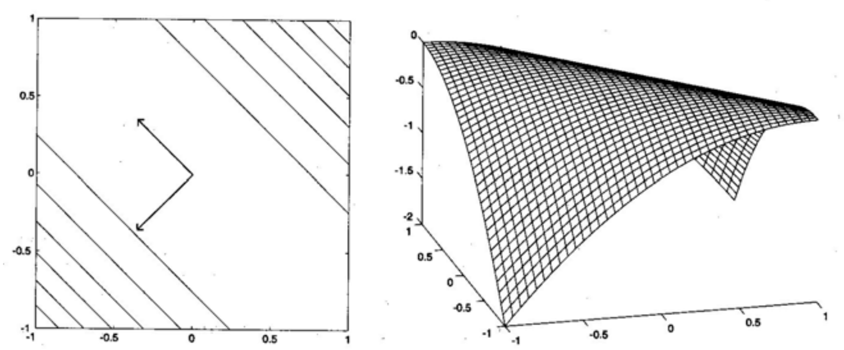


Figure 18.4.2-1 Example Lyapunov Function

18.4.3 Lyapunov Surface

- For the content-addressable memory network, all of the diagonal elements of the weight matrix will be equal to Q (the number of prototype patterns).

The diagonal is zero by subtracting Q times the identity matrix,

$$\mathbf{W}' = \mathbf{W} - Q\mathbf{I} \tag{18.4.3-1}$$

$$\mathbf{W}'\mathbf{p}_q = [\mathbf{W} - Q\mathbf{I}]\mathbf{p}_q = S\mathbf{p}_q - Q\mathbf{p}_q = (S - Q)\mathbf{p}_q \tag{18.4.3-2}$$

- (S-Q) is an eigenvalue of **W**', and the corresponding eigenspace is X, the space spanned by the prototype vectors.

$\mathbf{a} \in X^{\perp}$,

$$\mathbf{W}'\mathbf{a} = [\mathbf{W} - Q\mathbf{I}]\mathbf{a} = \mathbf{0} - Q\mathbf{a} = -Q\mathbf{a} \tag{18.4.3-3}$$

- -Q is an eigenvalue of **W**', and the corresponding eigenspace is $X^{\perp}$.

Summary

- The eigenvectors of **W**' are the same as the eigenvectors of **W**, but the eigenvalues are (S-Q) and -Q, instead of S and 0.
- The eigenvalues of the Hessian matrix of the modified Lyapunov function, $\nabla^2 V'(\mathbf{a}) = -\mathbf{W}'$, are -($S$-$Q$) and Q.
- The Lyapunov function surface will have negative curvature in X and positive curvature in $X^{\perp}$, in contrast with the original Lyapunov function, which had negative curvature in X and zero curvature in $X^{\perp}$.