Chapter 11

Adaptive Resonance Theory

Neural Networks: A Classroom Approach
Satish Kumar
Department of Physics & Computer Science
Dayalbagh Educational Institute (Deemed University)

Competition and Cooperation

- Ubiquitous in the real world
- Ensemble based neuronal cooperation and competition
  - Neurons race to enhance their activation.
- Neurons within an ensemble compete with one another to maximize their activation.
- Complex feedback paths within and between such ensembles
- Quickly lead to very complicated dynamics.
- Primary aim:
  - Constrain the network architecture and neuron signal functions sufficiently enough to extract useful functionality from the system

Noise vs Saturation

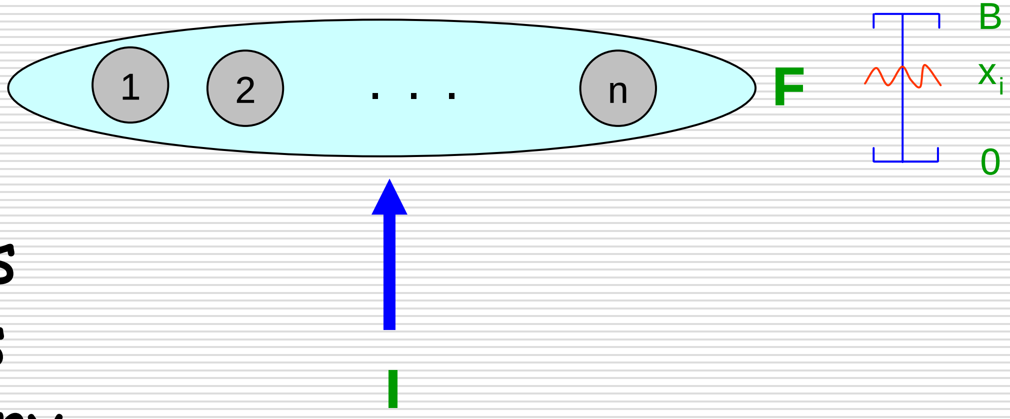
- Noise:
  - Operating levels of neuronal signals are very small
  - Small signals can be easily lost as noise
- Saturation:
  - All neurons have finite operating levels
  - Strong signals can cause neurons to saturate, leading effectively to a loss of information

Noise-Saturation Dilemma

- If the activities of neurons are sensitive to small inputs, how does one avoid having them saturate for large inputs?
- If the activities are sensitive to large inputs, how does one avoid small inputs getting lost as noise?

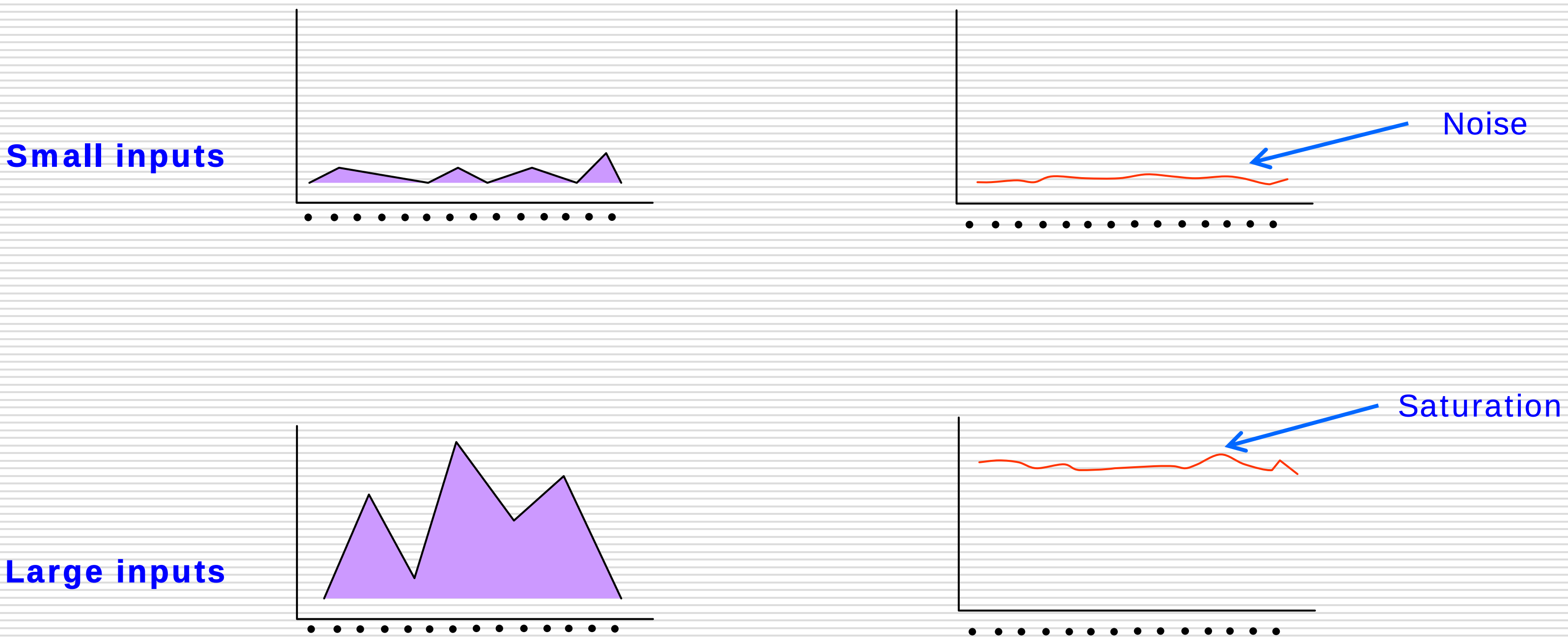
An Interpretation of Activation

- Each node comprises a population of B excitable cells
- Measure of the activity of a node is the number of cells that are firing at any point of time
- Range of neuron activation is 0 to B



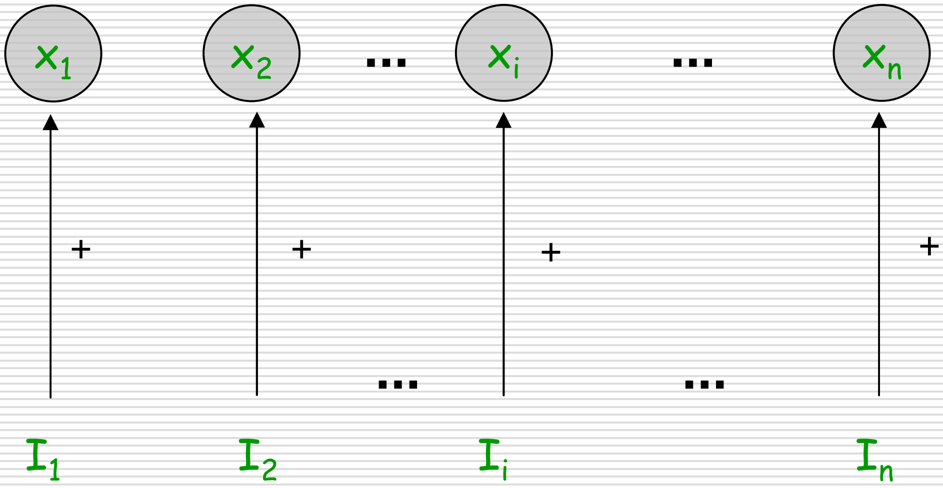
The layer F comprises n nodes which can receive an external input I

Functional Unit is a Spatial Pattern: Noise vs Saturation



Input Spatial pattern **Network Activation Pattern**

Shunting Net: No Interactions



$$\dot{x}_i = -Ax_i + (B - x_i)I_i$$

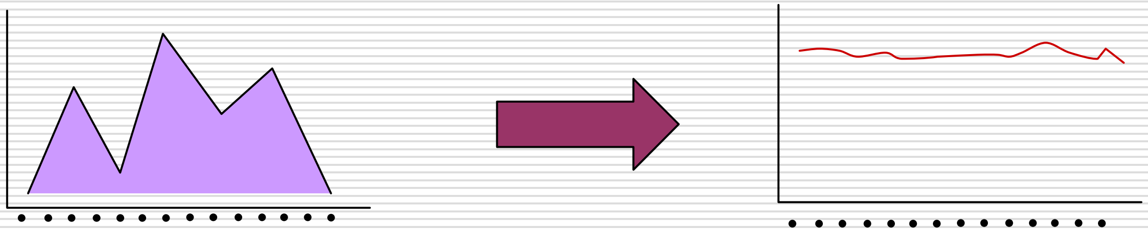
Shunting Network with no Interactions

- Assume a spatial pattern
  - $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)^T$
- where $\theta_i = I_i / \tilde{I}$
- In equilibrium

$$\hat{x}_i = \frac{B I_i}{A + I_i} = \frac{B\,\theta_i\,\tilde{I}}{A + \theta_i\,\tilde{I}} = \frac{B}{(\frac{A}{\theta_i \tilde{I}}) + 1}$$

$$\rightarrow B \text{ as } \tilde{I} \rightarrow \infty$$

Shunting Network with no Interactions



As the input intensity increases, activities saturate in a shunting network with no interactions

Shunting Networks with Lateral Interactions

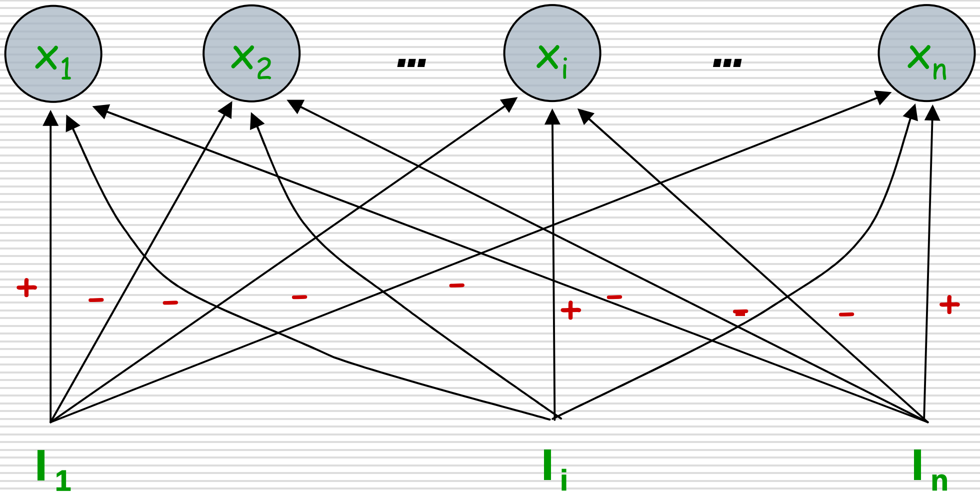
- Preserve the sensitivity of the network to θ_i even as the total input intensity increases
- Note

$$\theta_i = \frac{I_i}{I_i + \sum_{j \neq i} I_j}$$

Cooperation

Competition

On-Center Off-Surround Network



Introducing lateral interactions of external inputs to the shunting network can help solve the noise-saturation dilemma

Multiplicative or Shunting Interactions

- Introduce a *multiplicative* or *shunting* feedback term

Cooperation Competition

$$\dot{x}_i = -Ax_i + (B - x_i)\,I_i - x_i \sum_{j \neq i} I_j$$

Product terms

Shunting Nets Don't Saturate!

- Steady state analysis reveals

$$\hat{x}_i = \frac{B\tilde{I}}{A+\tilde{I}}\theta_i = \frac{B\theta_i}{(\frac{A}{\tilde{I}})+1}$$

$$\rightarrow B\theta_i \text{ as } \tilde{I} \rightarrow \infty$$

  - System activities no longer saturate as the input intensity increases
  - Pattern information is preserved for an infinite operating range!
  - Simple competitive feedback solves the noise-saturation dilemma.

Automatic Gain Control

- Factor x_i which multiplies ΣI_j is an automatic gain control
  - Large activity values: x_i increases towards B and the effective inhibitory feedback increases which tends to restore the activity towards the zero state resting level
  - Low activity values: inhibition becomes small allowing the dynamics to be governed to a large extent by the excitation
  - Automatic gain control helps to maintain the system's sensitivity to pattern information at varying levels of input intensity

Simple Shunting Networks: Shift Invariance

- System maintains its sensitivity for different off-surrounds
- A larger off-surround requires a larger on-center at which the system demonstrates that sensitivity
- System automatically adapts its range of sensitivity depending on the value of the present off-surround level

$$\hat{x}_i = \frac{B I_i}{A + \tilde{I}} = \frac{B I_i}{A + I_i + \sum_{j \neq i} I_j}$$

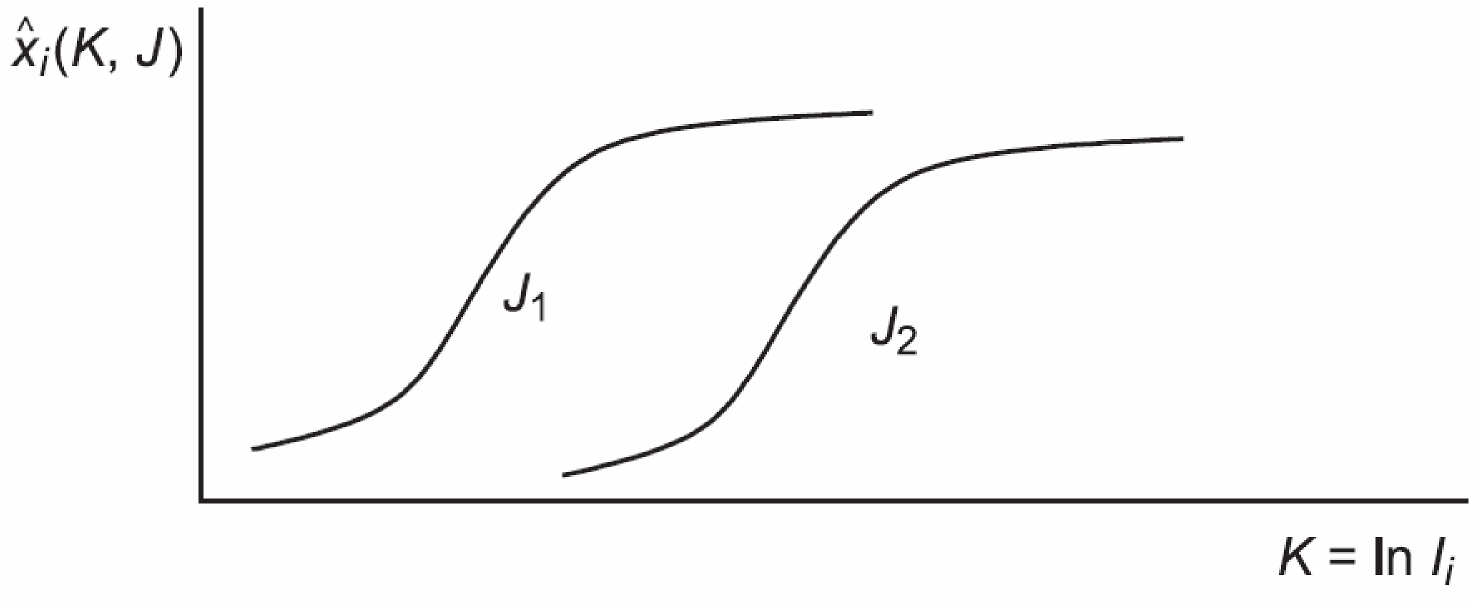
$$K = \ln I_i$$

$$J = \sum_{j \neq i} I_j$$

$$\hat{x}_i(K, J) = \frac{B e^K}{A + e^K + J}$$

Simple Shunting Networks: Shift Invariance

$$\hat{x}_i(K + S, J_1) = \hat{x}(K, J_2)$$

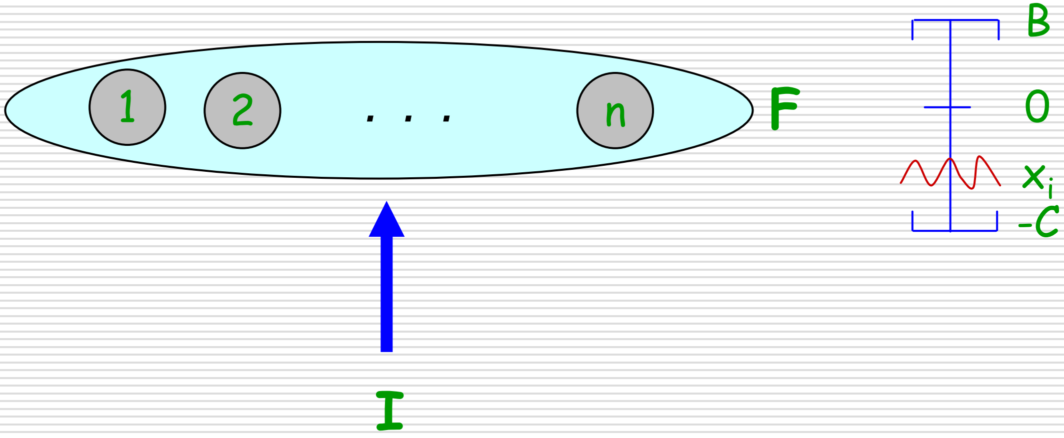


Shunting networks shift their range of sensitivity as a function of the net off-surround

Simple Shunting Networks: Shift Invariance

- Make the shunting model more biological
  - *Activity levels go negative as they do in vivo*
  - Range from [0,B] to [−C,B]

$$\dot{x}_i = -Ax_i + (B - x_i)I_i - (x_i + C)\sum_{j \neq i} I_j$$



Simple Shunting Networks: Noise Suppression

- In steady state →

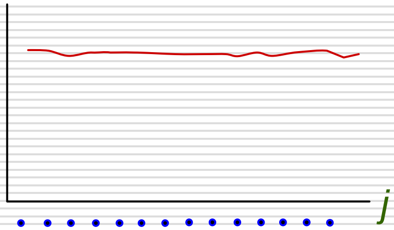
$$\hat{x}_i = \frac{(B+C)\tilde{I}}{A+\tilde{I}}\left[\theta_i - \frac{C}{B+C}\right]$$

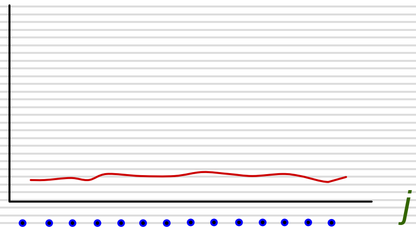
- $C/(C+B)$ is called the *adaptation level*
- A threshold that θ_i must exceed in order to produce a positive activity level
- Such a network can suppress noise: *zero spatial frequency patterns*

Simple Shunting Networks: Noise Suppression

- Suppose the ratio of C/B matches the ratio of nodes excited by I_i to those inhibited by I_i
- In other words, we set $C/B = 1/(n-1)$ or $C/(C+B) = 1/n$

$$\hat{x}_i = \frac{(B+C)\tilde{I}}{A+\tilde{I}}\left[\theta_i - \frac{1}{n}\right]$$



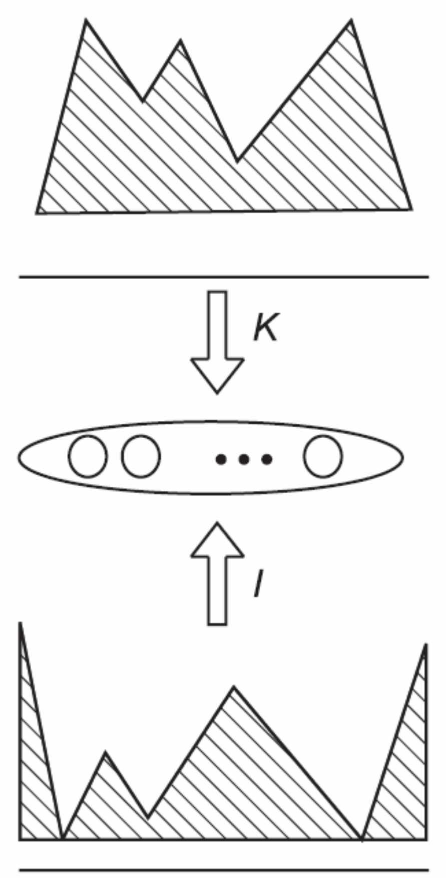


Shunting networks with a hyperpolarizing term suppress noise

Noise Suppression Facilitates Pattern Matching

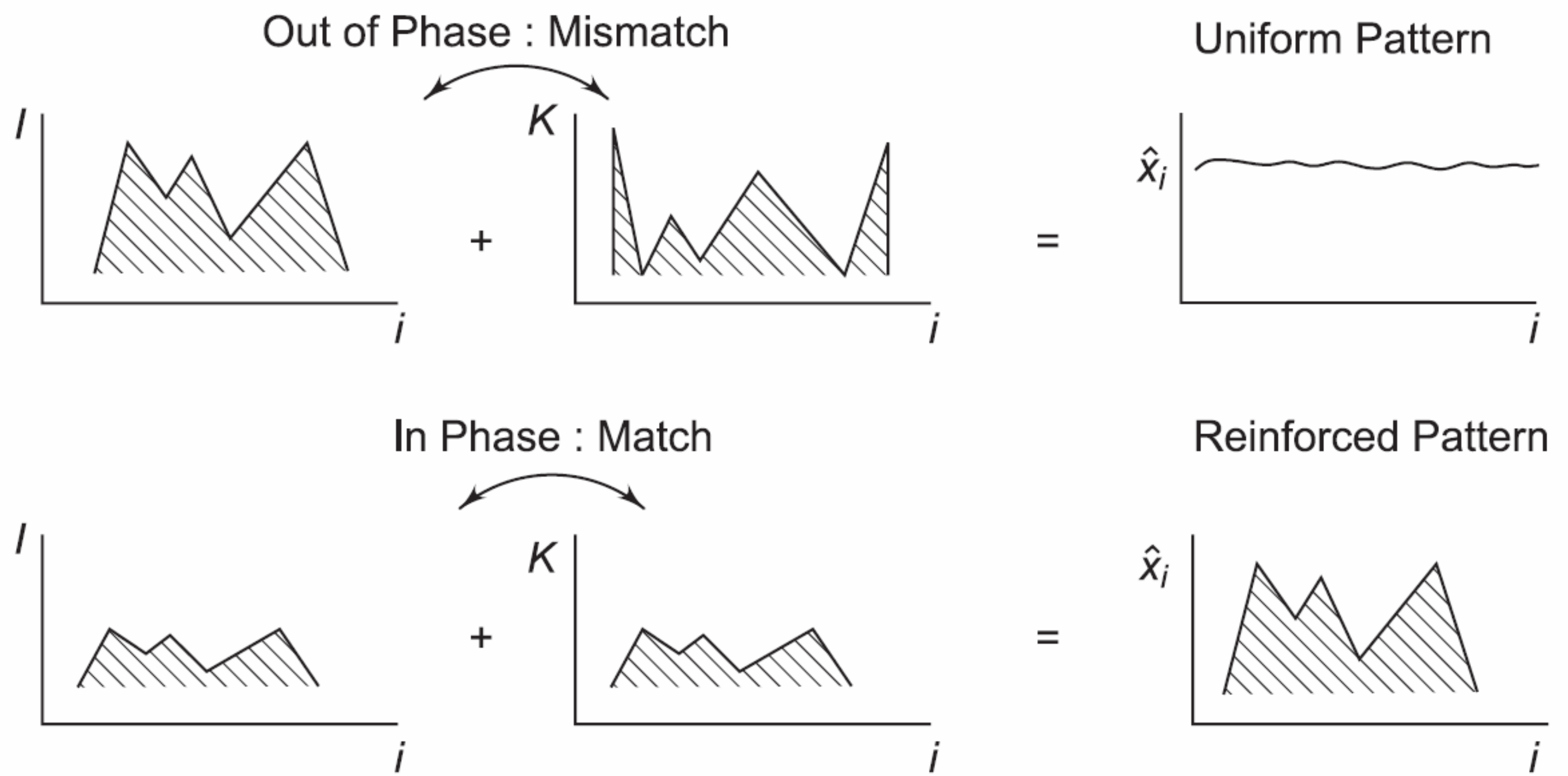
- When two spatial patterns impinge simultaneously on an on-center-off-surround network, the network computes the extent to which they match.

$$\hat{x}_i = \frac{(B+C)(1+\alpha)\tilde{I}}{A+(1+\alpha)\tilde{I}}\left(\theta_i - \frac{C}{B+C}\right)$$



Noise Suppression Facilitates Pattern Matching

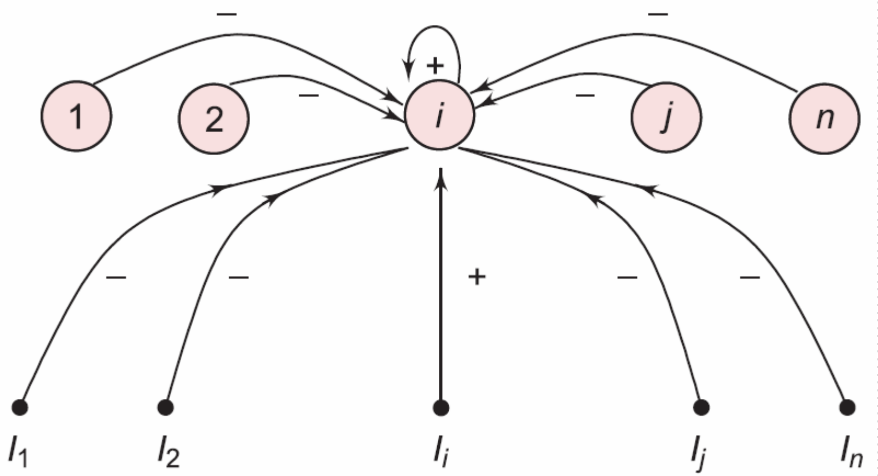
Out of phase patterns lead to somewhat uniform activity patterns which are suppressed



In-phase patterns reinforce one another and are amplified

Recurrent On-Center—Off-Surround Networks

- Include inhibitory and excitatory intra-field signal feedback



$$\dot{x}_i = -Ax_i + (B - x_i)\big(I_i + \mathcal{S}(x_i)\big) - x_i \left(J_i + \sum_{j \neq i} \mathcal{S}(x_j) \right)$$

Generalized On-Center Off-Surround Systems

- The noise suppression property when generalized to systems that have *distance dependent interactions*
- Endows them with the capability to detect edges in a spatial input
- Connectivity in networks with distance dependent interactions is usually governed by kernel functions such as the Gaussian function
- Also form the basis of Mexican hat networks we study in Chapter 12

Shunting Model has Cohen-Grossberg Dynamics

- Consider, the general class of on-center-off-surround shunting models

$$\dot{x}_i = -A_i x_i + (B_i - x_i)\left[I_i + \mathcal{S}(x_i)\right] - (x_i + C_i)J_i + \sum_{j=1}^{n} w_{ji}\mathcal{S}(x_j)$$

- Set $y_i = x_i + C_i$

$$\dot{y}_i = a_i(y_i)\left(b_i(y_i) - \sum_{j=1}^{n} c_{ji} d_j(y_j)\right)$$

Shunting Model has Cohen-Grossberg Dynamics

- With the substitutions:

$$a_i(y_i) = y_i$$

$$b_i(y_i) = \frac{1}{y_i}\{A_iC_i - (A_i + J_i)y_i + (B_i + C_i - y_i)[I_i + \mathcal{S}(y_i - C_i)]\}$$

$$c_{ji} = w_{ji}$$

$$d_j(y_j) = \mathcal{S}(y_j - C_j)$$

Transformation to Pattern Variable Form

- See algebra in text
- Pattern variable →

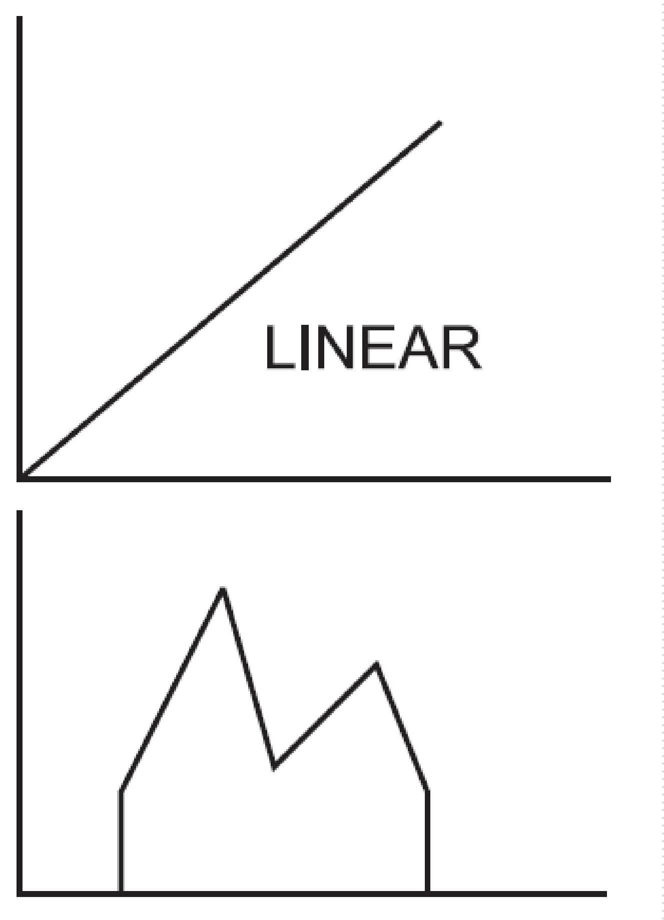
$$\overline{\underline{X}}_i = \frac{x_i}{\sum_j x_j} = \frac{x_i}{X}$$

$$\dot{\overline{\underline{X}}}_i = B\overline{\underline{X}}_i \sum_k \overline{\underline{X}}_k (g_i - g_k)$$

Signal to activation ratio

Choice of the Signal Function Determines Network Behaviour

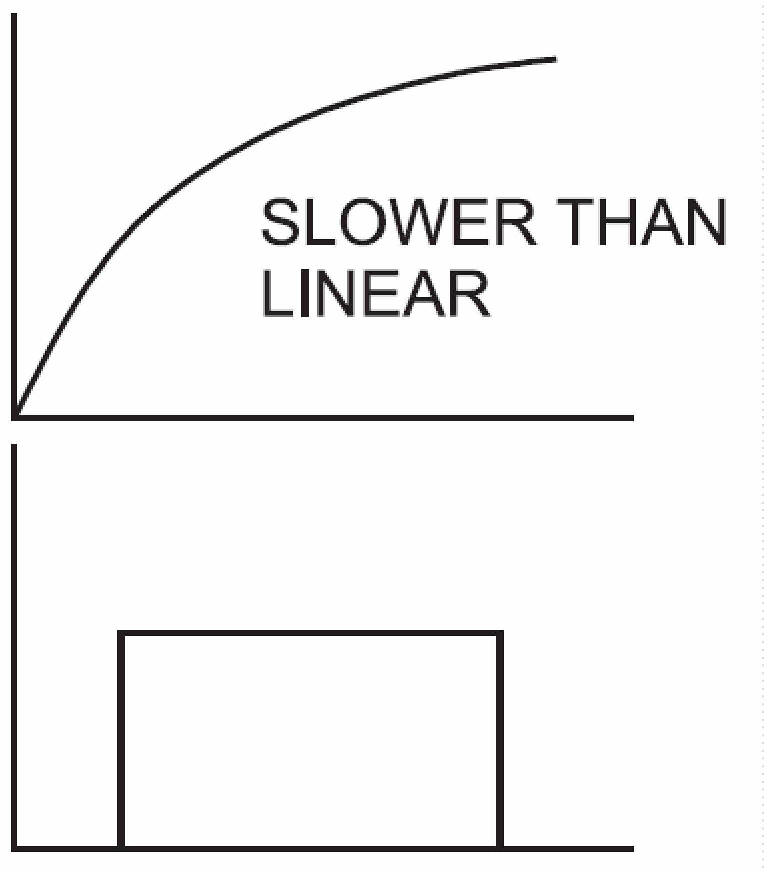
- ***Case 1:*** ***Linear Signal Function***
- Stores patterns, amplifies noise



Choice of the Signal Function Determines Network Behaviour

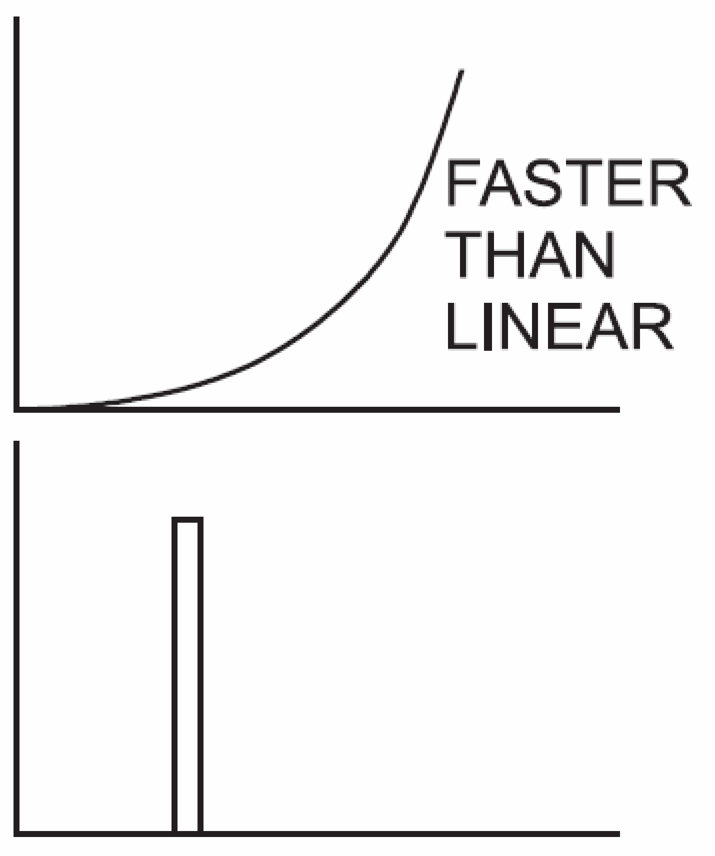
- *Case 2: Slower-than-linear Signal Function*

- Amplifies noise, and experiences seizure.



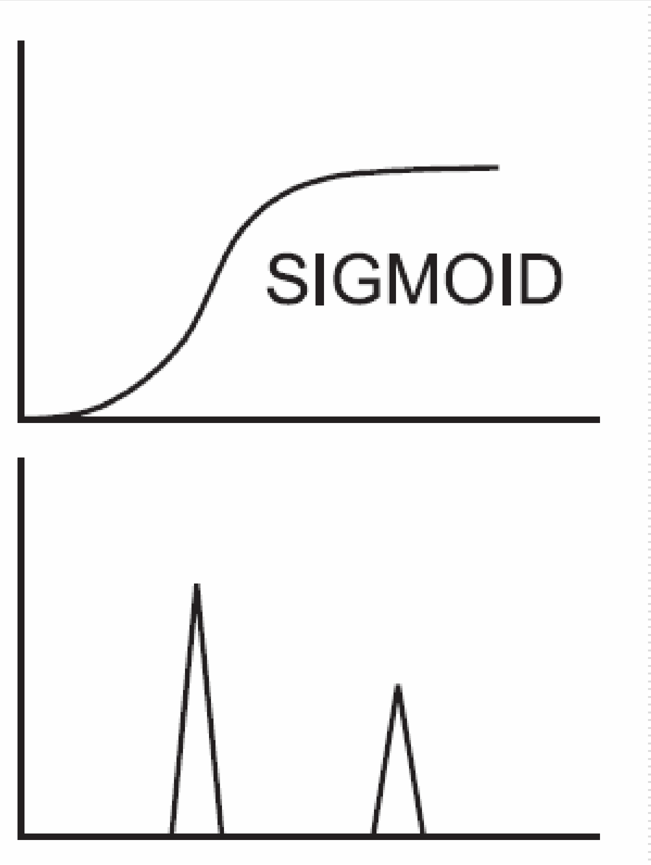
Choice of the Signal Function Determines Network Behaviour

- ***Case 3:*** ***Faster-than-linear Signal Function***
- Quenches noise, and exhibits winner-take-all behaviour.



Choice of the Signal Function Determines Network Behaviour

- ***Case 4: Combining the Three Cases:*** ***Sigmoidal Signal Function***

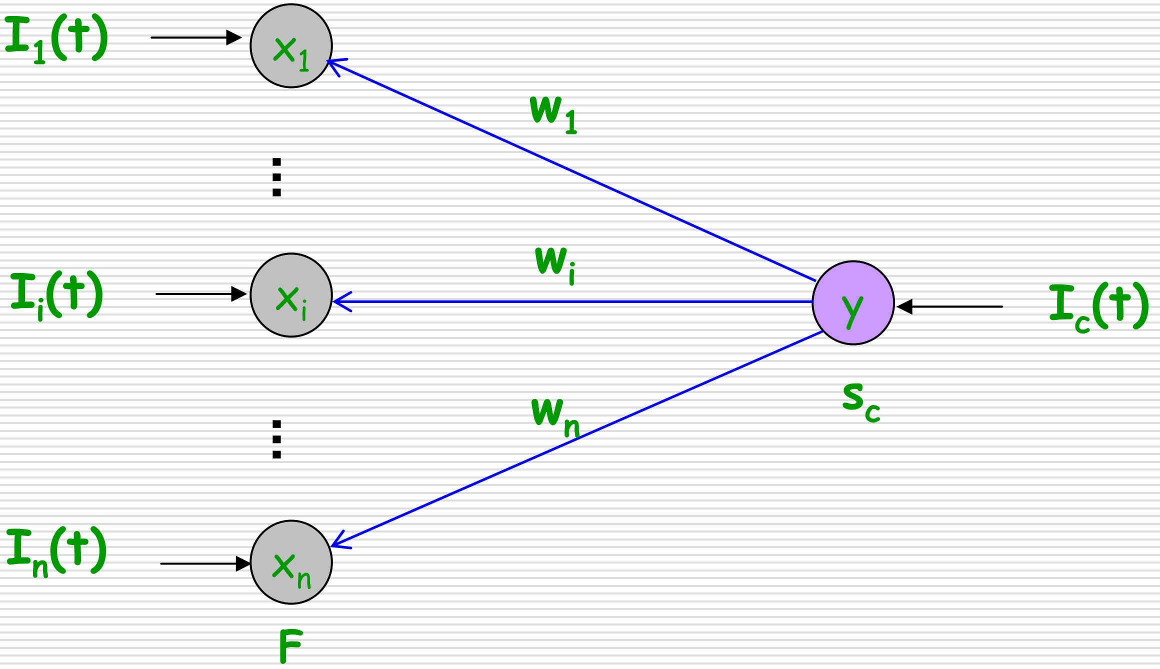


- Combining the three cases: faster-than-linear, linear, slower than-linear, quenches noise and enhances the signal.

Building Blocks of Adaptive Resonance

- Study specialized architectures of neuronal systems that integrate both short-term memory and long-term memory dynamics
- Perform powerful functions of storing, recalling and recognizing *spatial* patterns of neuronal activity
- Simple building blocks when put together in a systematic way, result in the adaptive resonance architecture
- Two models required are
  - *outstars*
  - *instars*

Outstar: Architecture



$$\dot{x}_i = -ax_i + s_c w_i + I_i \qquad i = 1, \ldots, n$$

$$\dot{w}_i = -bw_i + s_c x_i \qquad i = 1, \ldots, n$$

Outstar: Analysis

- Total activity equations

$$\dot{\overline{X}}_i = C(W_i - \overline{X}_i) + D(\theta_i - \overline{X}_i)$$

LTM values read into STM activities

spatial pattern read into STM

$$\dot{W}_i = E(\overline{X}_i - W_i)$$

STM downloads into LTM

Outstar: Analysis

- In the absence of the command neuron signal, $s_c = 0$
  - External inputs set up neuronal activities quickly to extract pattern space information
  - No learning can take place unless the command neuron switches on.
- In the absence of any input when $\theta_i = 0$
  - LTMs are read out into STMs when external inputs are absent and the command neuron switches on

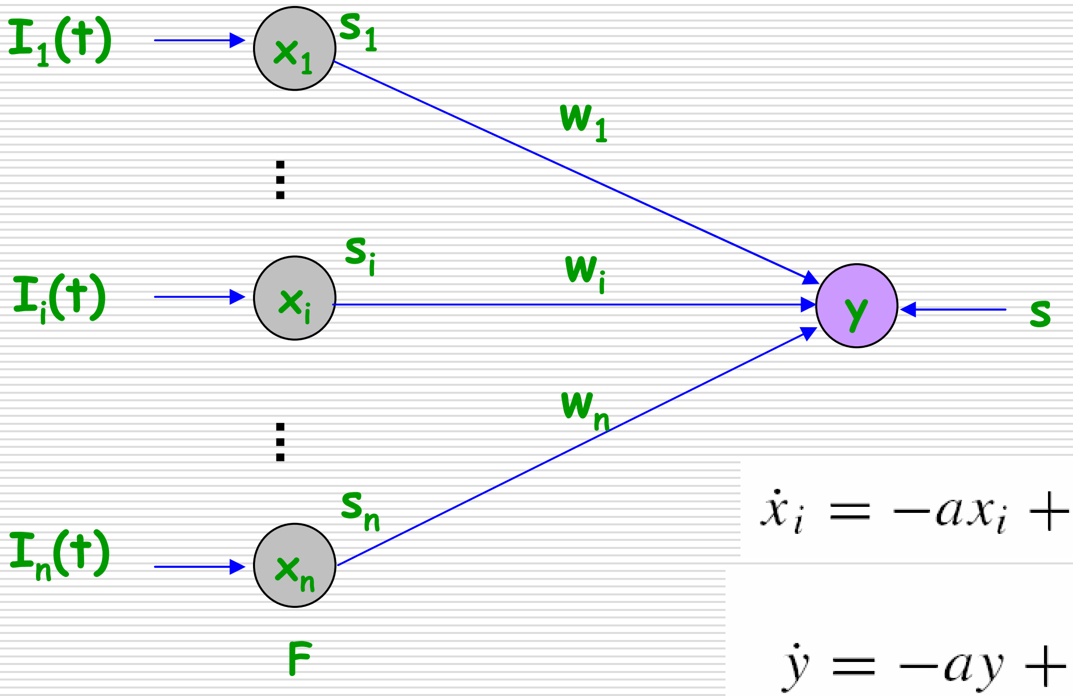
$$\dot{\underline{\overline{X}}}_i = D(\theta_i - \underline{\overline{X}}_i)$$

$$\dot{W}_i = 0$$

$$\dot{\underline{\overline{X}}}_i = C(W_i - \underline{\overline{X}}_i) \quad \longleftarrow \textbf{Fast}$$

$$\dot{W}_i = E(\underline{\overline{X}}_i - W_i) \quad \longleftarrow \textbf{Slow}$$

Instar: Architecture



$$\dot{x}_i = -ax_i + I_i(t) \qquad i = 1, \ldots, n$$

$$\dot{y} = -ay + \sum_{i=1}^{n} w_i s_i$$

$$\dot{w}_i = -bw_i + s_i y \qquad i = 1, \ldots, n$$

Instar: Analysis

- In the steady state,

$$x_i(\infty) = \frac{\tilde{I}}{a}\theta_i \qquad i = 1, \ldots, n$$

- Assume for analysis that $s_i = K\theta_i$
- Therefore

$$\dot{w}_i = -bw_i + K\theta_i y \qquad i = 1, \ldots, n$$

$$\dot{W}_i = G(\theta_i - W_i)$$

Instar: Analysis

- Pattern information is downloaded into LTMs
- Learning is "gated" by the postsynaptic neuron activation y

$$\dot{W}_i = G(\theta_i - W_i)$$

Ky/W

$$\dot{y} = -ay + \sum_{i=1}^{n} w_i s_i = -ay + K \sum_{i=1}^{n} w_i \theta_i$$

$$y(\infty) = \frac{K}{a} \sum_{i=1}^{n} w_i \theta_i$$

Instar: Analysis

- Special Case: Learning has Signal Hebbian Form

$$\dot{w}_i = -bw_i + s_i s = -bw_i + K\theta_i s$$

- The form of the signal s is binary threshold

$$\dot{W}_i = \frac{K s}{W}\left(\theta_i - W_i\right)$$

- Learning takes place only if $s = 1$

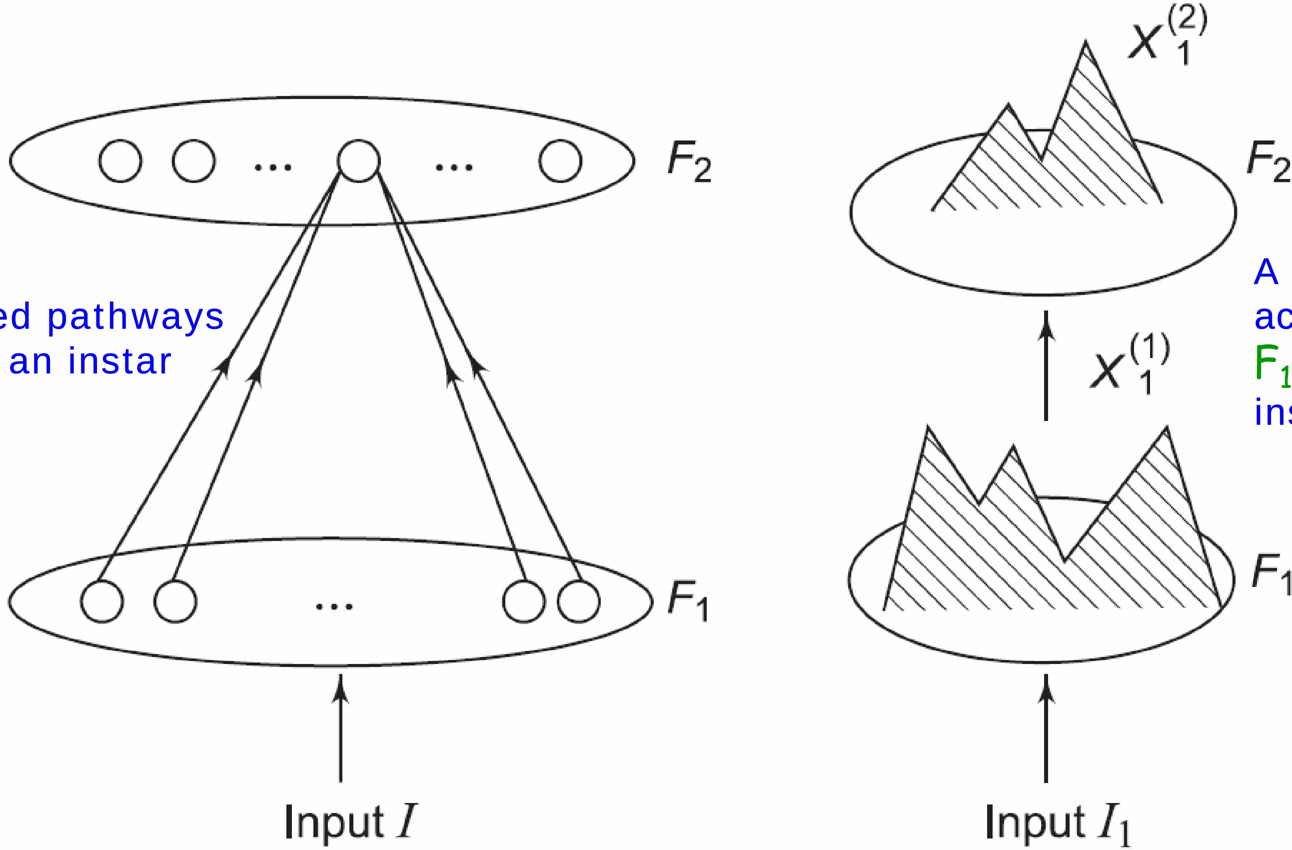
Fundamental Questions

- "How do internal representations of the environment develop through experience?"
  - How is there a consistency between such internal models?
  - How do errors in internal codes get corrected?
  - How does the system adapt to a constantly changing environment?

Helmholtz Doctrine of Perception

- Internal cognitive factors dramatically influence our perception of the environment
  - When external sensory data impinges on a system, it sets up an internal feedback process
  - This elicits a feedback expectancy or learned prototype
  - This modifies the external sensory data patterns recorded.
  - It is only when there is a consensus or "resonance" between an impinging pattern and a learnt feedback expectancy prototype that true perception takes place

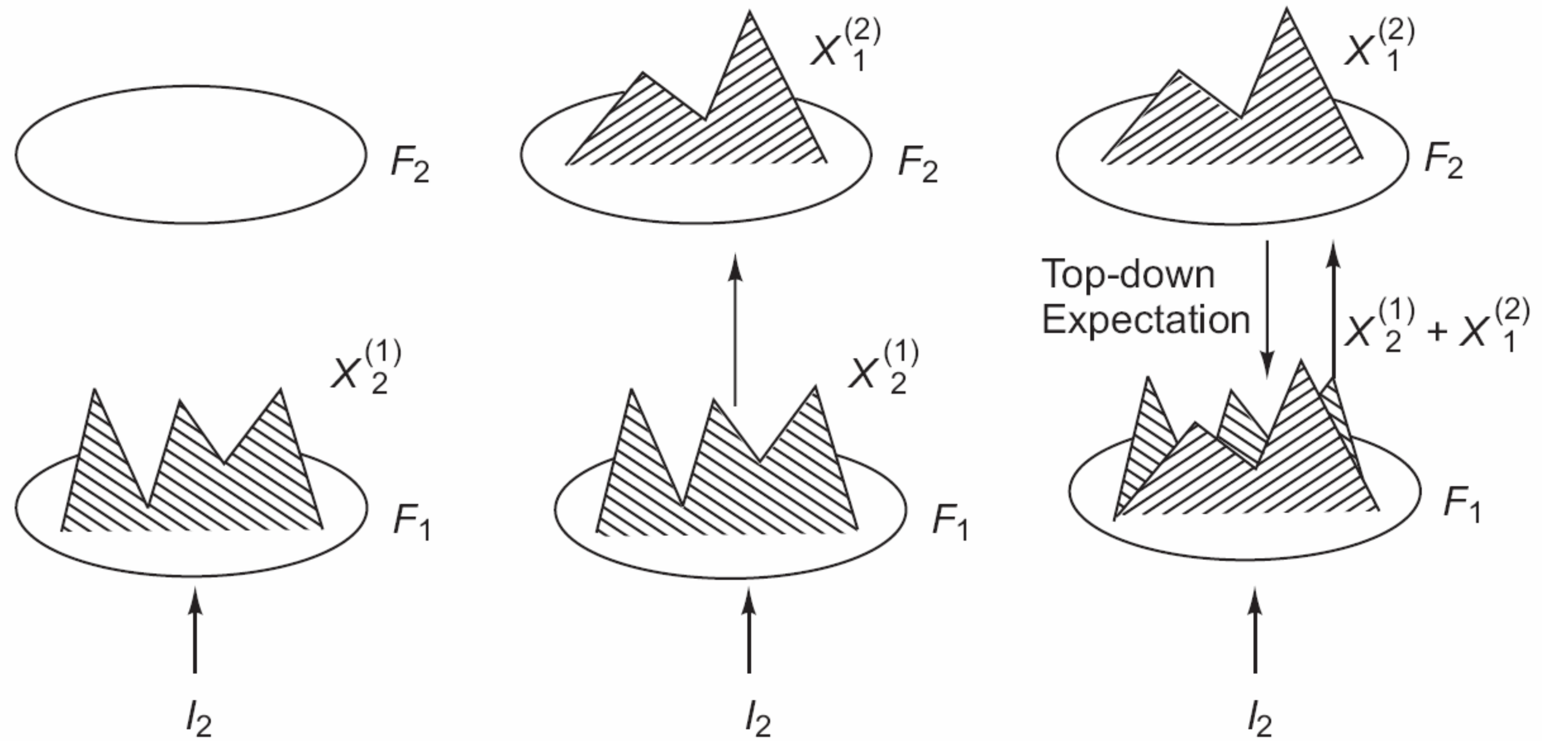
Substrate of Resonance (1)



Weighted pathways form an instar

A spatial pattern of activity filters from F_1 to F_2 through the instars

Substrate of Resonance (2)



A bottom-up input elicits a learned feedback expectancy in the form of a top-down response.

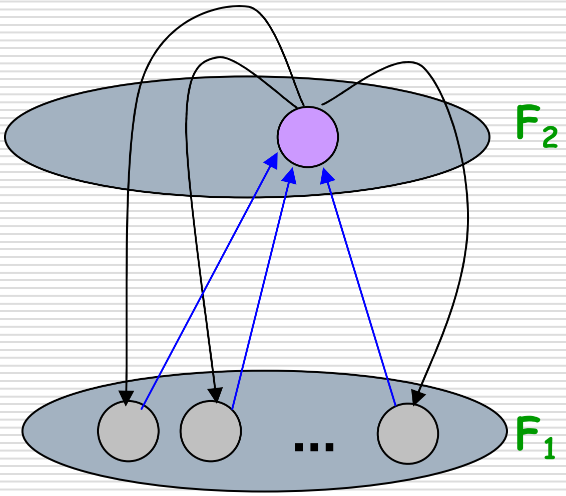
Superimposition of patterns in F_1 can then either lead to pattern reinforcement or suppression

Substrate of Resonance (3)

- Pattern matching represents a *resonance* between the input and what is expected to be seen as input
- This cycle of resonance should persist between layers F_1 and F_2 as long as the input is held active.
- Pattern mismatch represents a condition of *dissonance*
  - Signals a coding error
  - Pattern of uniform/near uniform activities should be suppressed along with the elicited pattern X
  - Paves the way for the external input to elicit another expectancy pattern across F_2 which might match the present input to a greater extent

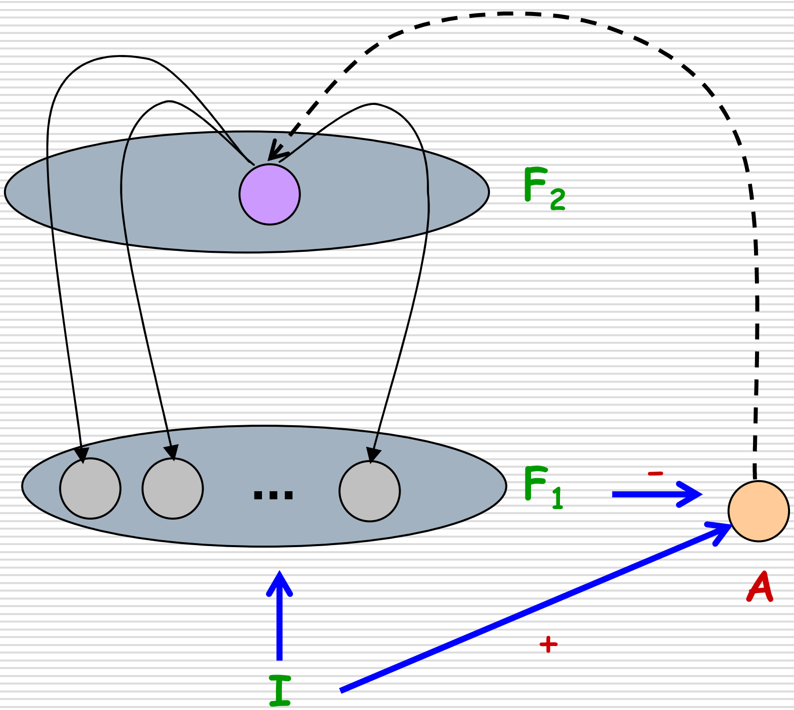
Structural Details of the Resonance Model

- Bottom-up inputs filter from F_1 to F_2 through instar weights
- Learned feedback expectancy elicited from F_2 is fed back to F_1 through an outstar



Structural Details of the Resonance Model: Attentional Vigilance

- An external *reset* node that samples the net activity of the input pattern and the activity of the pattern that develops across F_1 after top-down feedback is superimposed
- Attentional vigilance
  Facilitates long-term suppression of the F_2 node that fires in case of an error



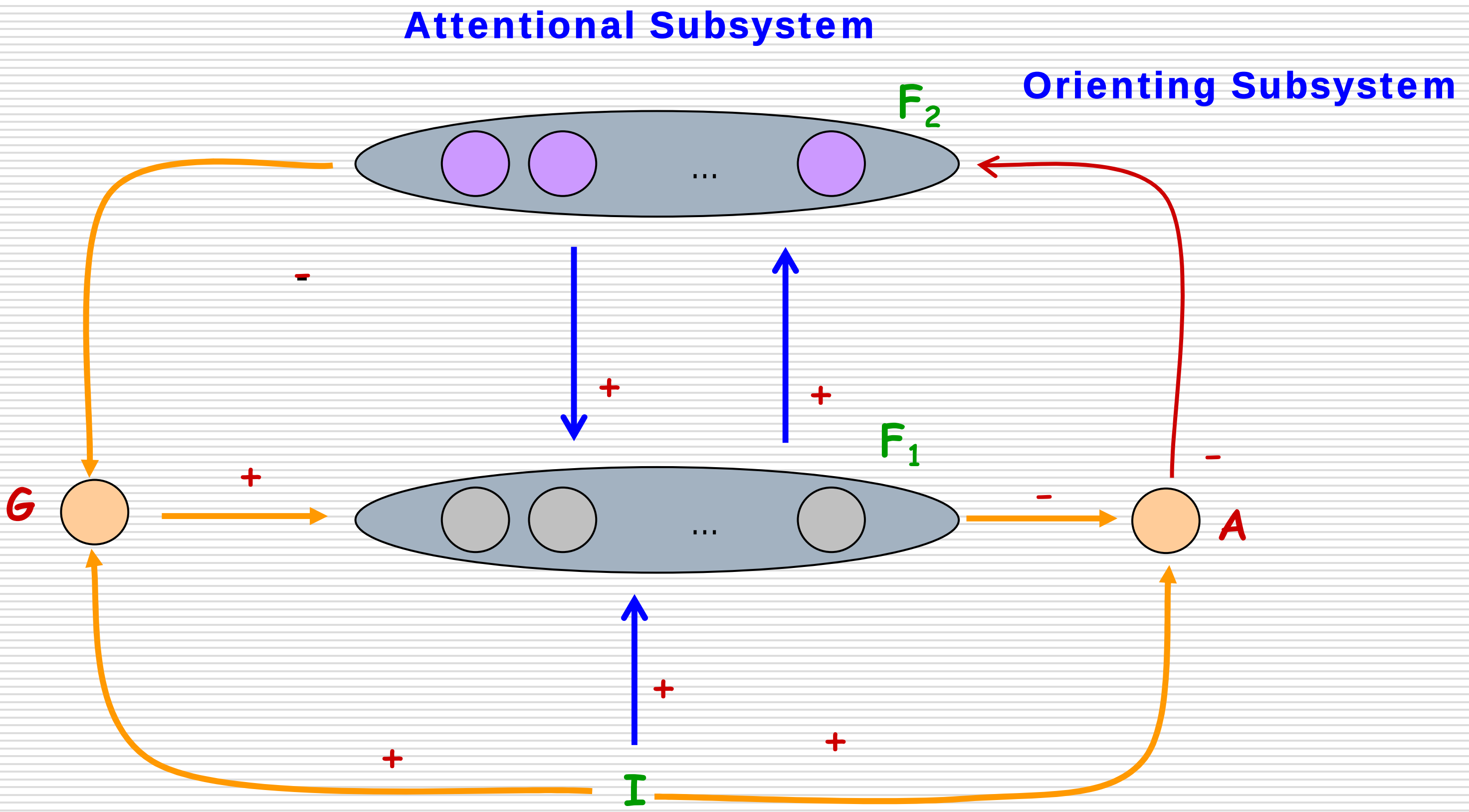
Search Cycle using Attentional Vigilance

- If the feedback reinforces the spatial pattern across F_1
  - Activity levels of F_1 nodes are amplified
  - Increases the inhibition to *A* further
  - increasingly difficult for *A* to fire
- If the feedback pattern mismatches the pattern presently set up across F_1
  - Leads to a suppression of activities of F_1 nodes.
  - Decreases the net inhibition to *A*
  - Results in the net activation of node *A* going above the threshold.
  - Causes *A* to fire
  - When *A* fires, its signal provides long lasting inhibition to the currently firing node of F_2
  - This suppresses the feedback
  - Causes the original spatial pattern that elicited the feedback to be reinstated across F_1
  - The competitive processes in F_2 select a new winner which elicits a new feedback and a new search cycle is initiated.

Adaptive Resonance Theory 1 (ART 1)

- ART 1 is a binary classification model.
- Various other versions of the model have evolved from ART 1
- Pointers to these can be found in the bibliographic remarks
- The main network comprises the layers F_1, F_2 and the attentional gain control as the *attentional subsystem*
- The attentional vigilance node forms the *orienting subsystem*

ART 1: Architecture



ART 1: STM Dynamics

- neuronal activations are confined to intervals
  - $[-B_1/C_1, 1/A_1]$
  - $[-B_2/C_2, 1/A_2]$

$$\epsilon \dot{x}_i = -x_i + (1 - A_1 x_i) J_i^+ - (B_1 + C_1 x_i) J_i^- \qquad i = 1, \ldots, n$$

$$\epsilon \dot{y}_j = -y_j + (1 - A_2 y_j) J_j^+ - (B_2 + C_2 y_j) J_j^- \qquad j = 1, \ldots, m$$

  - For details see text

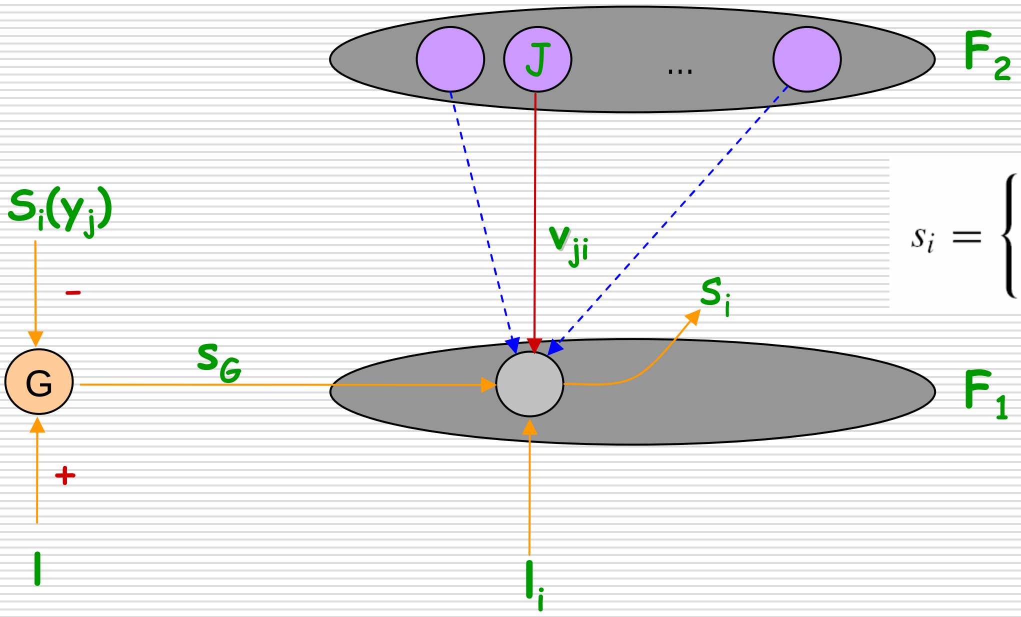
ART 1: Binary Switching Scenario

- F_2 behaves like a binary choice network

$$\mathcal{S}_j(x_j) = \begin{cases} 1 & \text{if } U_j = \max\ \{U_k\}_{k=1}^{m} \\ 0 & \text{otherwise} \end{cases}$$

- In this binary switching scenario then,
  - $V_i = D_1 v_{Ji}$ where node J of F_2 is active
- Top down feedback vector V_J is a scaled version of the outstar weights that emanate from the only active node J :
  - $V_J = D_1(v_{J1}, \ldots, v_{Jn})^T$

ART 1: 2/3 Rule



$$s_i = \begin{cases} 1 & \text{if any two of three inputs are active} \\ 0 & \text{otherwise} \end{cases}$$

Three kinds of inputs to each F_1 neuron decide when the neuron fires ⇒

- External input I_i
- Top-down feedback through outstar weights v_{ji}
- Gain control signal s_G

ART 1: 2/3 Rule

- The gain control signal $s_G = 1$ if I is presented and all neurons in F_2 are inactive
- s_G is nonspecific
- When the input is initially presented to the system, $s_G = 1$ → $s_i = I_i \wedge s_G = I_i \wedge 1 = I_i$
- As soon as a node J in F_2 fires as a result of competition, $s_G = 0$ → $s_i = I_i \wedge v_{Ji}$

Long-term Memory Dynamics

- Bottom-up connections,

$$\dot{w}_{ij} = K_1\, \mathcal{S}_j(y_j)\Big(-E_{ij} w_{ij} + \mathcal{S}_i(x_i)\Big)$$

- Top-down connections,

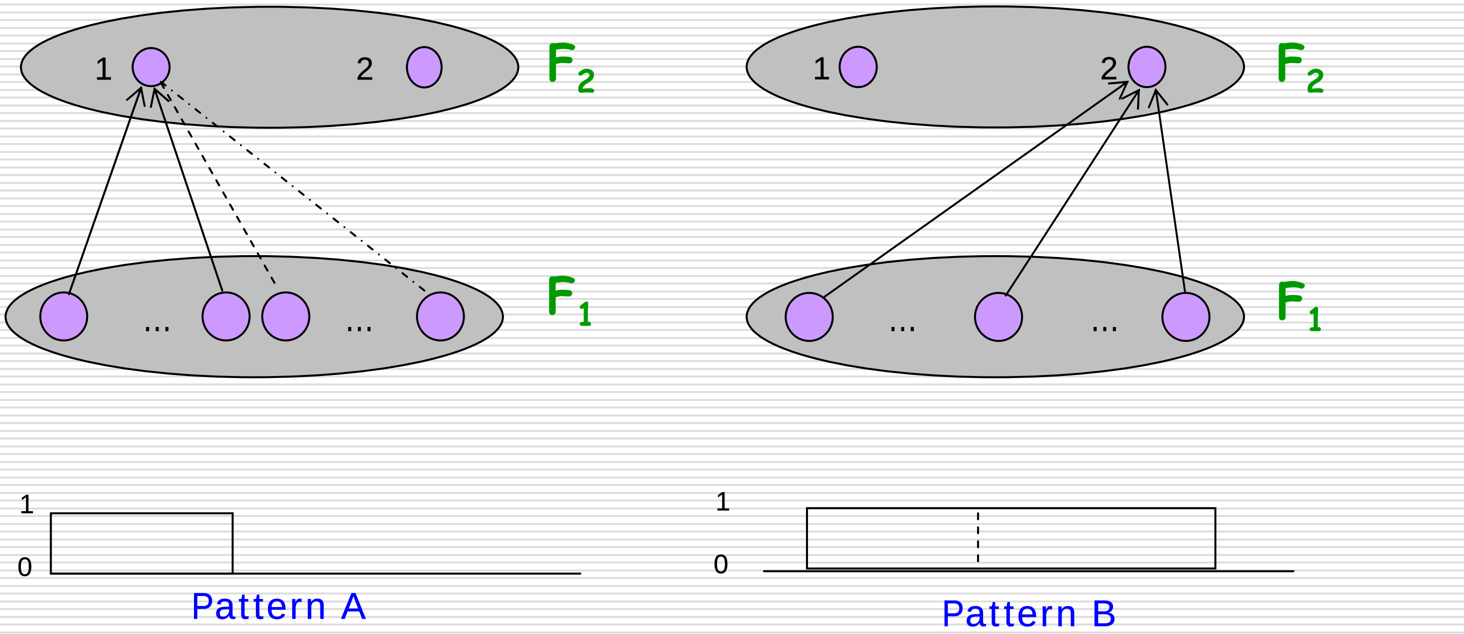
$$\dot{v}_{ji} = K_2\, \mathcal{S}_j(y_j)\Big(-E_{ji} v_{ji} + \mathcal{S}_i(x_i)\Big)$$

Weber Law Rule

- Basic idea
  - Values of weights learnt during presentation of a pattern A with a smaller number of active nodes should be larger than weights learnt during presentation of another pattern B with a larger number of active nodes
- Mathematically
  - As instar learning proceeds, the connection strength between active F_1 and F_2 nodes asymptotically approaches a value

$$w_{ij}(\infty) = \frac{\alpha}{\beta + |I|}$$

Weber Law Rule: Encoding Instars



Design of LTM Coefficient for Weber Law Rule

- Return to the LTM dynamics equation: $$\dot{w}_{ij} = K_1 \mathcal{S}_j(y_j)\Big(-E_{ij}w_{ij} + \mathcal{S}_i(x_i)\Big)$$
- Choose $$E_{ij} = \mathcal{S}_i(x_i) + \frac{1}{L}\sum_{k\neq i}\mathcal{S}_k(x_k) \qquad K_1 = KL$$
- Then, $$\dot{w}_{ij} = K\mathcal{S}_j(y_j)\Big((1-w_{ij})L\mathcal{S}_i(x_i) - w_{ij}\sum_{k\neq i}\mathcal{S}_k(x_k)\Big)$$
- Straightforward to verify that this embodies Weber Law Form

Final LTM Equations

- Instar

$$\dot{w}_{ij} = \begin{cases} K\big((1 - w_{ij})L - w_{ij}(|I| - 1)\big) & \text{if nodes } i \text{ and } j \text{ are active} \\ -K|I|w_{ij} & \text{if node } i \text{ is inactive, node } j \text{ is active} \\ 0 & \text{if node } j \text{ is inactive} \end{cases}$$

- Outstar

$$\dot{v}_{ji} = \begin{cases} -v_{ji} + 1 & \text{if nodes } i \text{ and } j \text{ are active} \\ -v_{ji} & \text{if node } i \text{ is inactive, node } j \text{ is active} \\ 0 & \text{if node } j \text{ is inactive} \end{cases}$$

Vigilance Parameter

- When V_J impinges on F_1
  - the activity of the signal vector can either remain the same if $V_J = I$ or
  - can decrease if V_J differs from I at some positions
- Degree of match measured by

$$M = \frac{|V_J \wedge I|}{|I|} = \frac{|\mathcal{S}(X)|}{|I|}$$

- Set a threshold ρ, called the vigilance parameter which defines the degree of match necessary to admit pattern I to a class (or cluster) J

Resonance

$$M = \frac{|\mathbb{S}(\mathbf{X})|}{|\mathbf{I}|} \geqslant \rho$$

- Resonance occurs when an external input filters through to F_2 and causes a category node J to fire which in turn sends back an expectation that matches the input to a degree greater than the vigilance parameter.
- The modified signal vector (after feedback) must subsequently cause the *same* F_2 node to fire in order for resonance to occur.
- The degree of match is acceptable
  - System goes into the resonant state
  - Admits the present input to category J
  - *Fast learning* takes place: learning is assumed to be immediate

$$w_{iJ} = \begin{cases} \frac{L}{L-1+|\mathbb{S}(X)|} & \text{if } \mathbb{S}_i(x_i) = 1 \\ 0 & \text{if } \mathbb{S}_i(x_i) = 0 \end{cases}$$

$$v_{Ji} = \begin{cases} 1 & \text{if } \mathbb{S}_i(x_i) = 1 \\ 0 & \text{if } \mathbb{S}_i(x_i) = 0 \end{cases}$$

STM Reset

$$M < \rho$$

- The degree of match is less than the minimum acceptable value as defined by ρ
- The system cannot admit the current input pattern I into presently firing category J of F_2
- Node A immediately fires an STM reset that provides a long lasting suppression of the presently active node of F_2
- This switches off the top-down outstar feedback and restores s_G and $S(X) = I$

Search

- Competition takes place between the remaining $m - 1$ nodes
  - A new winner emerges
  - A new outstar readout takes place.
- The new feedback vector is compared with the input I
  - $S(X)$ is possibly modified
- $|S(X)|/|I|$ is recomputed and compared with ρ
  - If it is greater than ρ, then resonance
  - If it is less than ρ, then an STM reset fires again and suppresses this second node of F_2, and the search repeats
  - If all nodes are exhausted the ART system adds a new F_2 node and admits the current input directly.

ART 1: Operational Summary

Given	A set of Q binary patterns $\mathcal{T} = \{I_k\}_{k=1}^{Q} \quad I_k \in \mathbb{B}^n$ to be classified by an ART 1 system.
Assume	↬ n nodes in F_1; m nodes in F_2. ↬ Let $\mathcal{J}$ define the set of indices of F_2 neurons that have not yet been reset in the current search cycle.
Initialize	↬ Set up fields F_1 and F_2 with n and m neurons respectively. $w_{ij} = \frac{2}{1+n} \quad \forall\, i, j$ $v_{ji} = 1 \quad \forall\, i, j$ $L = 2$ Set up ρ as specified.

ART 1: Operational Summary

Iterate ⟳Repeat

{

⇝ Select a pattern I_k and set $\mathcal{J} = \{1, \ldots, m\}$.

⟳Repeat

{

⇝ Compute F_1 signals:

Set $\mathcal{S}(X) = I_k$.

Since $\frac{|\mathcal{S}(X)|}{|I_k|} = \frac{|I_k|}{|I_k|} = 1.0 > \rho$ no reset.

⇝ Compute F_2 activations and choose the winner:

$u_j = \sum_{i=1}^{n} w_{ij} \mathcal{S}(x_i) \qquad \forall j \in \mathcal{J}$

Choose the winner index J such that $u_J = \max_{j \in \mathcal{J}} \{u_j\}$.

⇝ Recompute F_1 signals in the presence of outstar feedback:

$\mathcal{S}(X) = V_J \wedge I_k$ where $V_J = (v_{J1}, \ldots, v_{Jn})$.

ART 1: Operational Summary

$\rightsquigarrow$ Check for reset:

a. $M = |\mathcal{S}(X)|/|I_k|$

b. If $M \quad \rho$ do fast learning as specified in the next step.

c. If $M \geq \rho$ then reset node $J : \mathcal{J} = \mathcal{J} \setminus \{J\}$

c1. If $\mathcal{J} \neq \phi$, repeat search cycle from inner loop.

c2. Else add a new node to $F_2 : m = m + 1;\ \mathcal{J} = \{m\}$.

Set the winner $J = m$ and $V_J = I_k$.

Set $w_{iJ} = \frac{2}{1+|I_k|} \quad i = 1, \ldots, n$.

Break and take the next pattern.

$\rightsquigarrow$ Fast learning in winning instar/outstar

$w_{iJ} = \frac{2}{1+|\mathcal{S}(X)|} \qquad i = 1, \ldots, n$

$V_J = \mathcal{S}(X)$

}(until I_k leads to resonance on a node.)

}(until each pattern $I_k \in \mathcal{T}$ directly leads to resonance.)

Hand-worked Example

- Cluster the vectors 11100, 11000, 00001, 00011
- Low vigilance: 0.3
- High vigilance: 0.7

Hand-worked Example: ρ = 0.3

k	I_k	$\mathbf{W}$	U	J	V_J	$\mathcal{S}(X)$	M	$M > \rho$	Fast Learning
1	11100	$\begin{pmatrix} .33 & .33 \\ .33 & .33 \\ .33 & .33 \\ .33 & .33 \\ .33 & .33 \end{pmatrix}$	(.99 .99)	1	11111	11100	1.0	Yes	$\mathbf{W} = \begin{pmatrix} .5 & .33 \\ .5 & .33 \\ .5 & .33 \\ 0 & .33 \\ 0 & .33 \end{pmatrix}$ $\mathbf{V} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$
2	11000	$\begin{pmatrix} .5 & .33 \\ .5 & .33 \\ .5 & .33 \\ 0 & .33 \\ 0 & .33 \end{pmatrix}$	(1.0 .66)	1	11100	11000	1.0	Yes	$\mathbf{W} = \begin{pmatrix} .66 & .33 \\ .66 & .33 \\ 0 & .33 \\ 0 & .33 \\ 0 & .33 \end{pmatrix}$ $\mathbf{V} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$
3	00001	$\begin{pmatrix} .66 & .33 \\ .66 & .33 \\ 0 & .33 \\ 0 & .33 \\ 0 & .33 \end{pmatrix}$	(0 .33)	2	11111	00001	1.0	Yes	$\mathbf{W} = \begin{pmatrix} .66 & 0 \\ .66 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$ $\mathbf{V} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$
4	00011	$\begin{pmatrix} .66 & 0 \\ .66 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$	(0 1)	2	00001	00001	0.5	Yes	$\mathbf{W} = \begin{pmatrix} .66 & 0 \\ .66 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$ $\mathbf{V} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Hand-worked Example: $\rho = 0.7$

k	I_k	$\mathbf{W}$	U	J	V_J	$\mathcal{S}(X)$	M	$M > \rho$	Fast Learning
1	11100	$\begin{pmatrix} .33 & .33 \\ .33 & .33 \\ .33 & .33 \\ .33 & .33 \\ .33 & .33 \end{pmatrix}$	(.99 .99)	1	11111	11100	1.0	Yes	$\mathbf{W} = \begin{pmatrix} .5 & .33 \\ .5 & .33 \\ .5 & .33 \\ 0 & .33 \\ 0 & .33 \end{pmatrix}$ $\mathbf{V} = \begin{pmatrix} 1\,1\,1\,0\,0 \\ 1\,1\,1\,1\,1 \end{pmatrix}$
2	11000	$\begin{pmatrix} .5 & .33 \\ .5 & .33 \\ .5 & .33 \\ 0 & .33 \\ 0 & .33 \end{pmatrix}$	(1.0 .66)	1	11100	11000	1.0	Yes	$\mathbf{W} = \begin{pmatrix} .66 & .33 \\ .66 & .33 \\ 0 & .33 \\ 0 & .33 \\ 0 & .33 \end{pmatrix}$ $\mathbf{V} = \begin{pmatrix} 1\,1\,0\,0\,0 \\ 1\,1\,1\,1\,1 \end{pmatrix}$
3	00001	$\begin{pmatrix} .66 & .33 \\ .66 & .33 \\ 0 & .33 \\ 0 & .33 \\ 0 & .33 \end{pmatrix}$	(0 .33)	2	11111	00001	1.0	Yes	$\mathbf{W} = \begin{pmatrix} .66 & 0 \\ .66 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$ $\mathbf{V} = \begin{pmatrix} 1\,1\,0\,0\,0 \\ 0\,0\,0\,0\,1 \end{pmatrix}$
4	00011	$\begin{pmatrix} .66 & 0 \\ .66 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$	(0 1) (0) (−)	2 1 3	00001 11000 11111	00001 00000 00011	0.5 0.0 1.0	No No Yes	$\mathbf{W} = \begin{pmatrix} .66 & 0 & 0 \\ .66 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & .66 \\ 0 & 1 & .66 \end{pmatrix}$ $\mathbf{V} = \begin{pmatrix} 1\,1\,0\,0\,0 \\ 0\,0\,0\,0\,1 \\ 0\,0\,0\,1\,1 \end{pmatrix}$

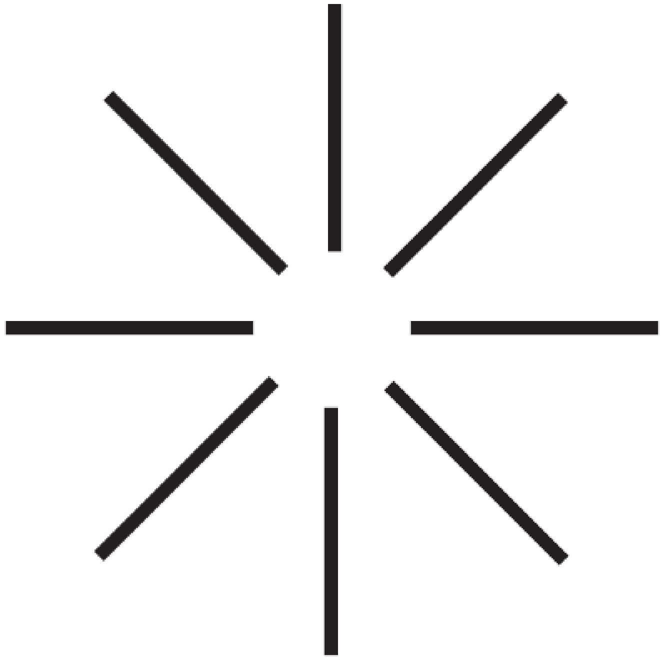
ART 1: MATLAB Code

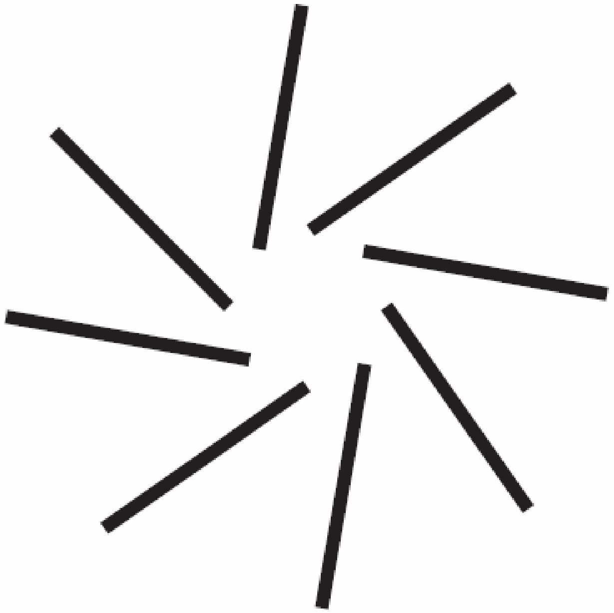
```
n = 5 %Initialize n,m, and rho
m = 1
rho = 0.3
out = ones(m,n) %Outstar initialized to 1
inwt = 2/(1+n) %Initial value of each instar weight
in = (inwt*out)' %Instar set up

p=[1 1 1 0 0 %Store the patterns
1 1 0 0 0
0 0 0 0 1
0 0 0 1 1]

q = 4 %q patterns to be classified

newnodeflag = 1 %Get into the loop
while(newnodeflag==1) % New node added to F2...
newnodeflag =0 % reset the newnodeflag...
for k = 1:q % for each pattern
  outstarlearn = 0 % reset the outstar learnt flag
  index = ones(1,m) % all nodes can compete
  y = p(k,:)*in % compute F2 activations
  while(sum(index) ~= 0) % some nodes not reset
    [maxy,windex] = max(y) % find index of winner
    s = out(windex,:).*p(k,:) % compute F1 signals
  M = sum(s)/sum(p(k,:)) % Find ratio of signals, M
  if M > rho % check ratio with vigilance
    out(windex,:) = s % if ok then learn
    in(:,windex) = (2/(1+sum(s)))*s'
    outstarlearn = 1 % set learnt flag
    break % exit
  else
    index(windex) = 0 % else reset the index entry
    y(windex) = -1 % suppress activity of M
  end % (they can never be negative!)
end

% No node could classify

if (~(outstarlearn) & sum(index) == 0)
  m = m+1 % add a new node
  out(m,:)=p(k,:) % Learn
  in(:,m)=p(k,:)'*(2/(1+sum(p(k,:))))
  newnodeflag = 1 % set the new node added flag
 end
end
end
```

Neurophysiological Evidence for ART Mechanisms

- The attentional subsystem of an ART network has been used to model aspects of the *inferotemporal cortex*
- Orienting subsystem has been used to model a part of the hippocampal system, which is known to contribute to memory functions
- The feedback prevalent in an ART network can help focus attention in models of visual object recognition

Ehrenstein Pattern Explained by ART !





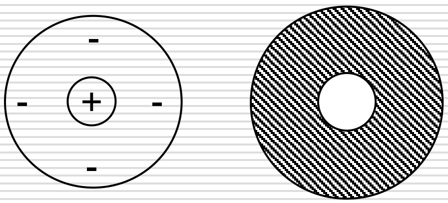
Generates a circular illusory contour – a circular disc of enhanced brightness

The bright disc disappears when the alignment of the dark lines is disturbed!

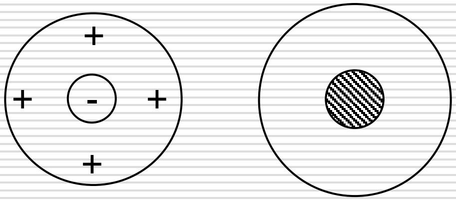
Other Neurophysiological Evidence

- Adam Sillito [University College, London]
  - Cortical feedback in a cat tunes cells in its LGN to respond best to lines of a specific length.
- Chris Redie [MPI Entwicklungsbiologie, Germany]
  - Found that some visual cells in a cat's LGN and cortex respond best at line ends— more strongly to line ends than line sides.
- Sillito et al. [University College, London]
  - Provide neurophysiological data suggesting that the cortico-geniculate feedback closely resembles the matching and resonance of an ART network.
  - Cortical feedback has been found to change the output of specific LGN cells, increasing the gain of the input for feature linked events that are detected by the cortex.

On Cells and Off Cells



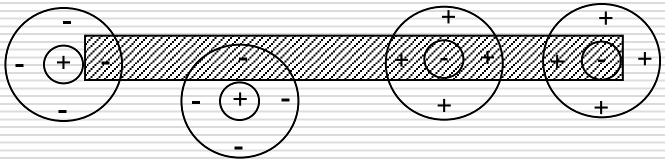
(a) On Cell



(b) Off Cell

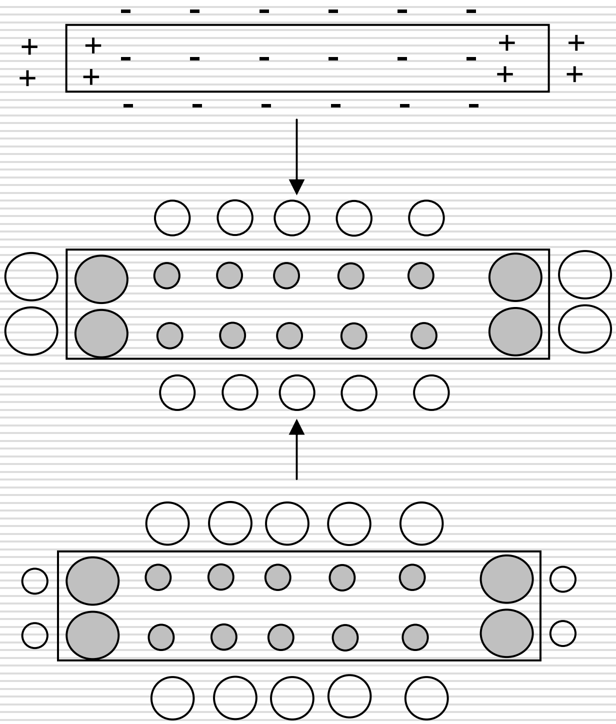
V1 – LGN Feedback

- Responses of ON-OFF cells **along the ends and edges of a dark line**



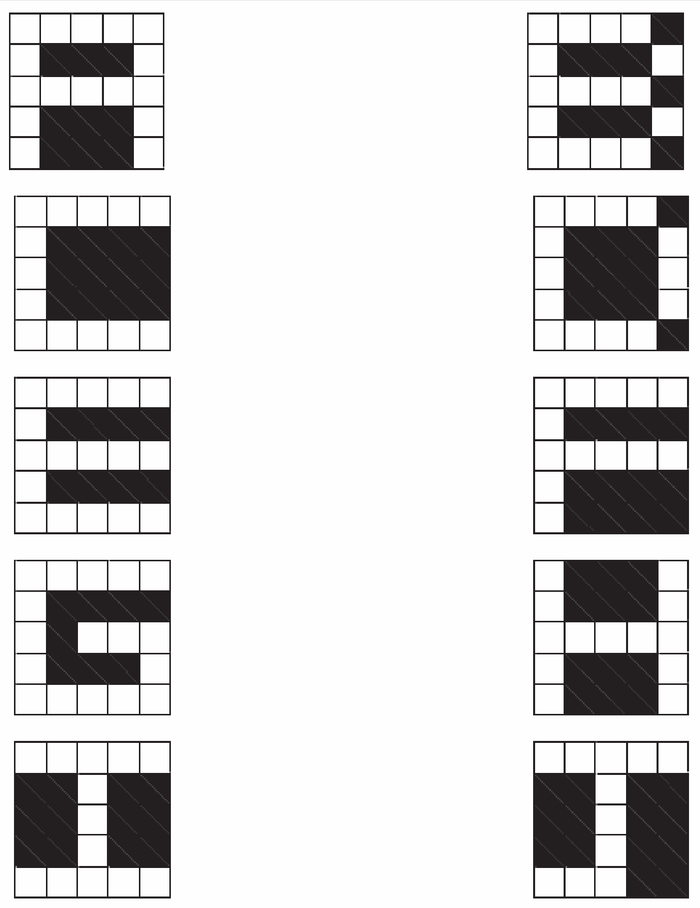
- Modifies ON-OFF cell activities to create brightness buttons

V1– LGN feedback pattern



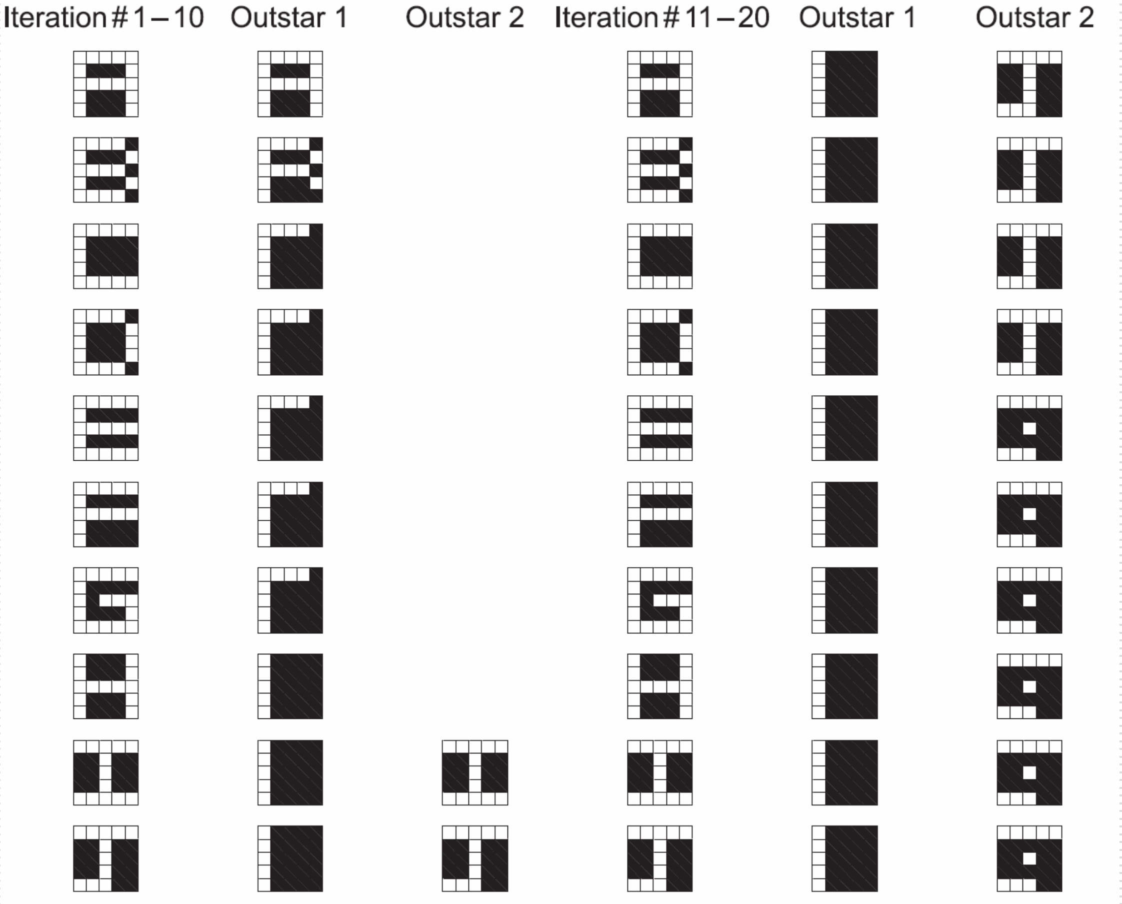
ART 1: Clustering Application

- Clustering pixel based alphabet images



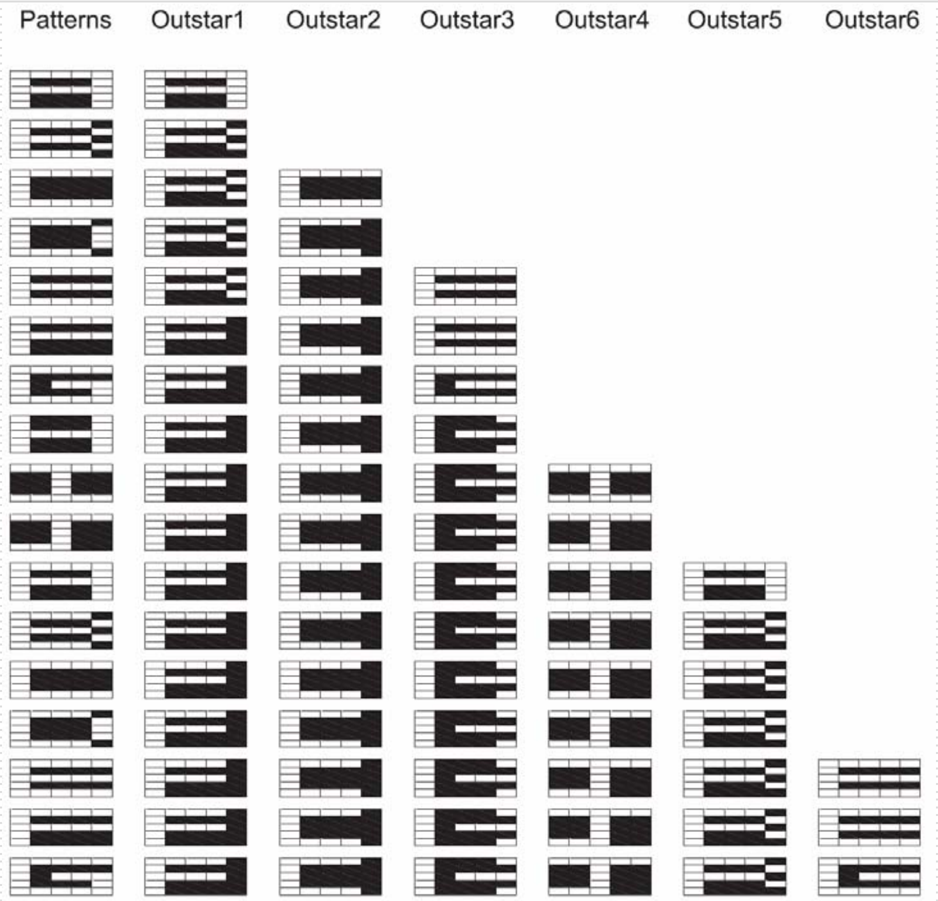
ART 1: Clustering Application

- $\rho = 0.3$



ART 1: Clustering Application

- $\rho = 0.7$



Other Applications

- Aircraft Part Design Classification System.
  - See text for details.