What can you do with your Brain-Inspired Computer now that you've built it?

Presented by Jim Anderson

Department of Cognitive and Linguistic Sciences

Brown University

Providence, RI 02912

Ersatz Core Participants

James A Anderson, Brown University

James_Anderson@brown.edu

Paul Allopenna, Aptima, Inc. and Brown University

pallopenna@aptima.com

John Santini, Jr., Alion, Inc. and Brown University

John_Santini@brown.edu

Gerald Guralnik, Brown University.

Gerry@het.brown.edu

Our Motto:

We want to build a first-rate, second-rate brain.

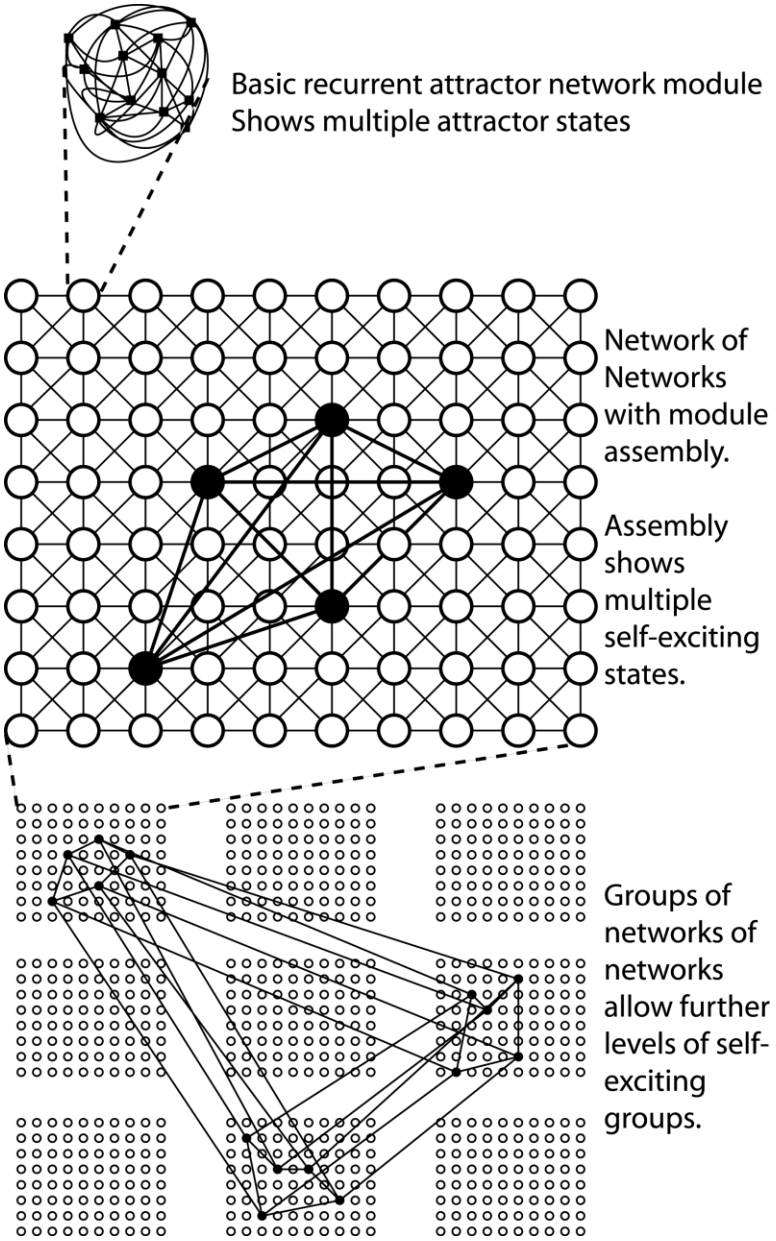
Acknowledgements

This work was supported by:

A seed money grant from the Office of the Vice President for Research, Brown University.

Phase I and Phase II SBIRs, "The Ersatz Brain Project," to **Aptima, Inc.** (Woburn MA), Dr. Paul Allopenna, Project Manager. Funding from the **Air Force Research Laboratory**, Rome, NY

Phase I STTR from DARPA to Aptima and Brown.

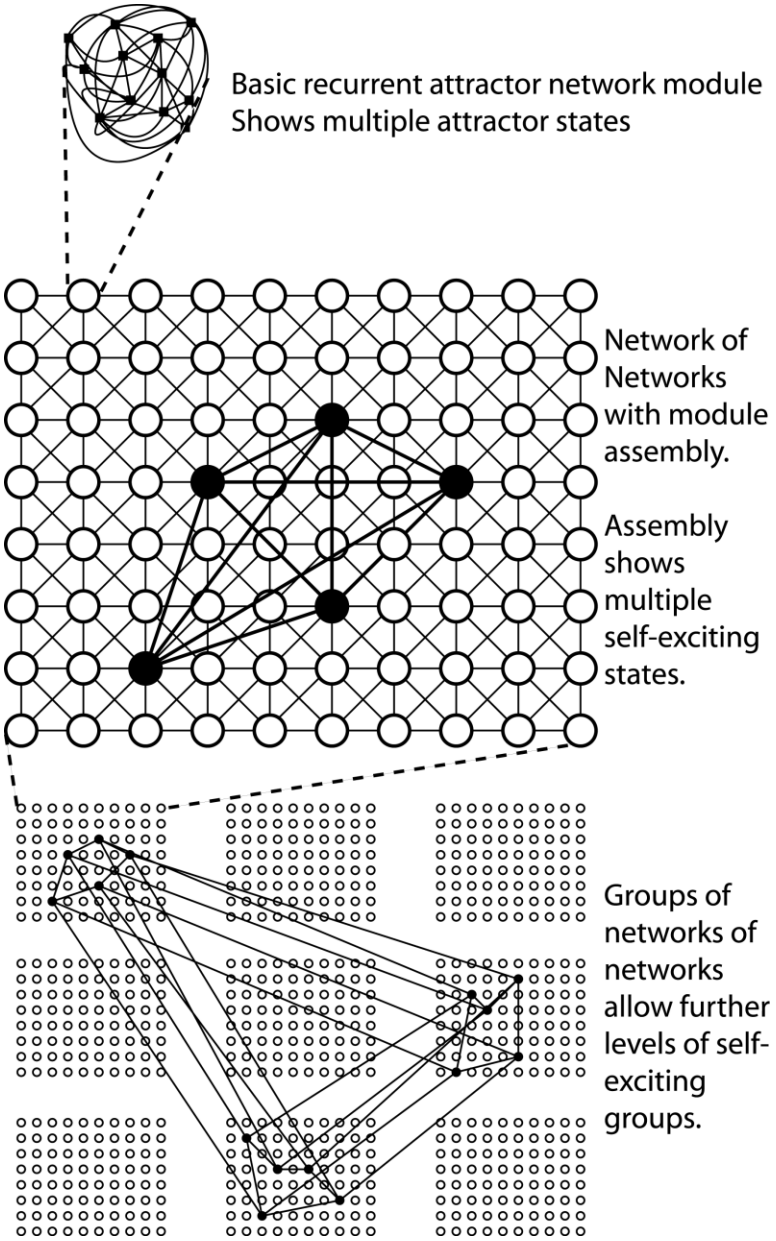


Summary (1)

The Ersatz Brain is constructed from many **small attractor networks** (modules, based on cortical columns) sparsely interconnected into a **Networks of Networks.**

It is capable of performing **specific cognitive tasks**.

It has both a **discrete** and a **continuous** set of control structures capable of **performing these specific tasks through explicit, teachable cognitive programs.**

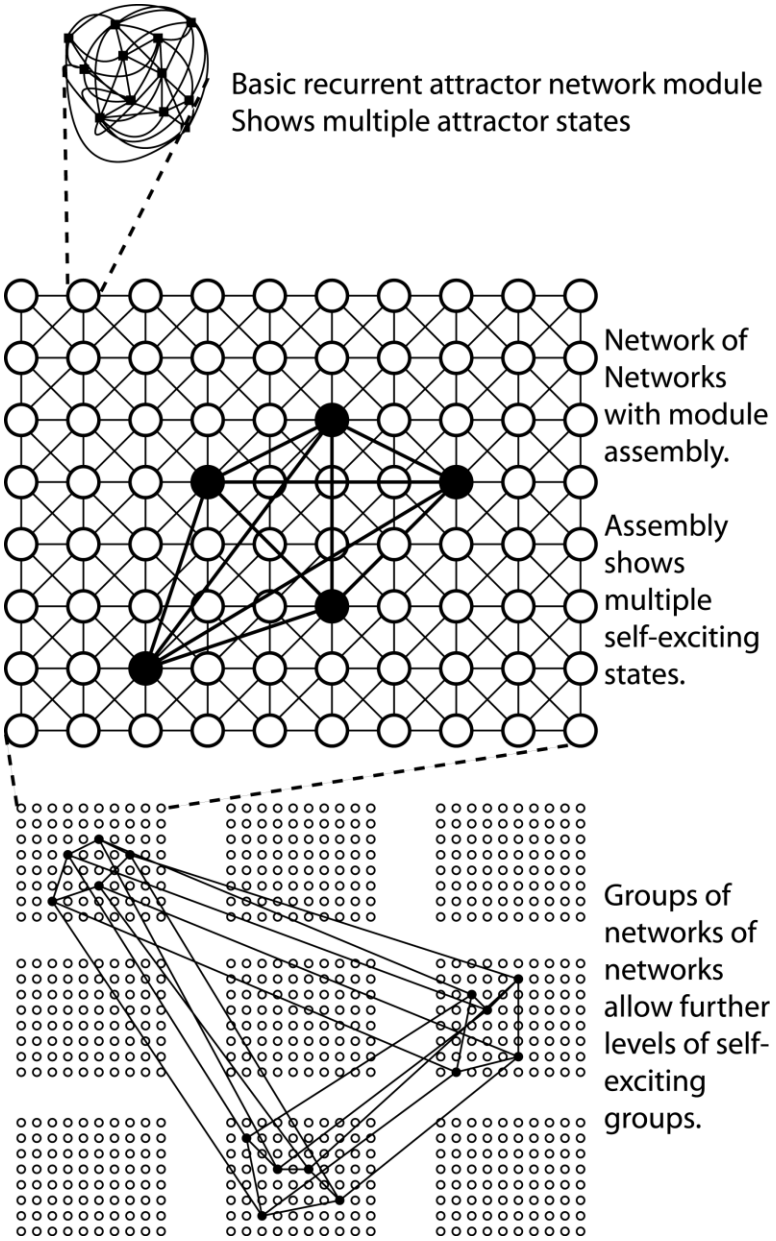


Summary (2)

Processing moves in scale from:

- **Inflexible but highly optimized, sensors**
- Through **sparse module assemblies** formed by learning in a single array of modules.
- To **flexible programs, involving learning and large dynamical systems** working at **multiple spatial scales**.

The resulting hierarchical computation is **fast, parallel, and highly integrative.**



Summary (3)

Will discuss examples of problems and computational techniques well suited to a brain-like computer:

In A Single Array:

- Identity and Symmetry
- Vowel Formant Invariant Representation

In Multiple Arrays:

Memory Filter Applications:

- List Scanning
- Disambiguation
- Anomaly Detection

The Ersatz Brain Approximation: The Network of Networks.

Conventional wisdom says **neurons are the basic computational units of the brain.**

The Ersatz Brain Project is based on a different approximation.

The Network of Networks model was developed in collaboration with **Jeff Sutton** then at Harvard medical school, now at NSBRI.

Cerebral cortex contains **intermediate level structure,** between neurons and an entire cortical region.

Intermediate level brain structures are **hard to study experimentally** because they require **recording from many cells simultaneously.**

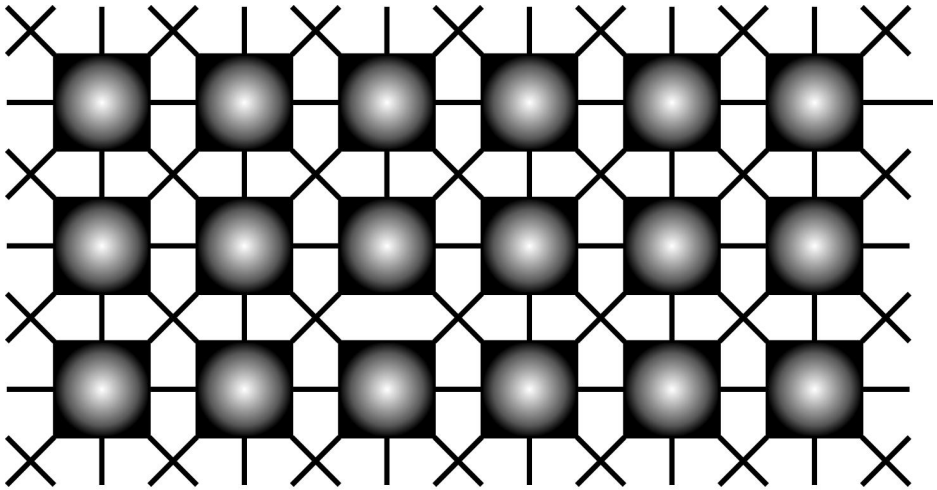
Network of Networks Approximation

We use the **Network of Networks [NofN] approximation** to structure the hardware and to reduce the number of connections.

The **basic computing units** are **not neurons**, but small (10^4 neurons) **attractor networks**.

Basic Network of Networks Hardware Architecture:

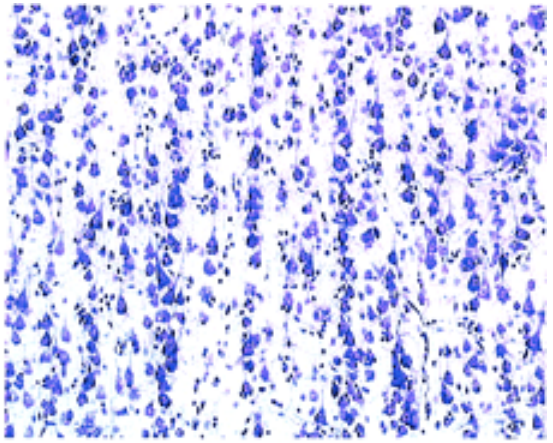
- **2 Dimensional arrays of modules (dynamical systems)**
- **Locally connected to neighbors**



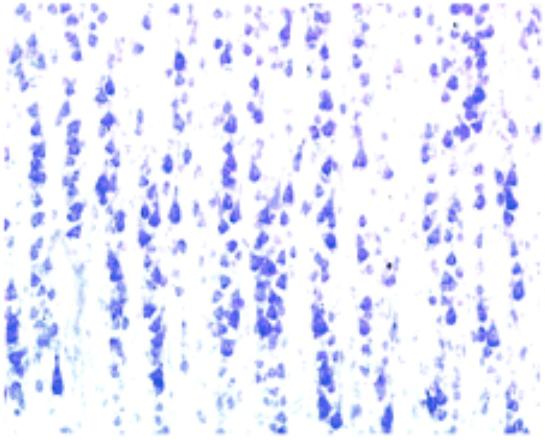
Network of Networks Modular Architecture

Cortical Columns: Minicolumns

"The basic unit of cortical operation is the *minicolumn* … It contains of the order of 80-100 neurons except in the primate striate cortex, where the number is more than doubled. The minicolumn measures of the order of 40-50 μm in transverse diameter, separated from adjacent minicolumns by vertical, cell-sparse zones … The minicolumn is produced by the iterative division of a small number of progenitor cells in the neuroepithelium." (Mountcastle, p. 2)



9 year old human.



67 year old human.

VB Mountcastle (2003). Introduction [to a special issue of *Cerebral Cortex* on columns]. *Cerebral Cortex*, **13**, 2-4.

Figure: Nissl stain of cortex in *planum temporale*.

Columns: Functional

Groupings of minicolumns seem to form the physiologically observed **functional columns.** Best known example is orientation columns in V1.

They are significantly bigger than minicolumns, typically around **0.3-0.5 mm.**

Mountcastle's summation:

"Cortical columns are formed by the binding together of many minicolumns by common input and short range horizontal connections. … The number of minicolumns per column varies … between 50 and 80. Long range intracortical projections link columns with similar functional properties." (p. 3)

Cells in a column ~ (80)(100) = 8000

Module Behavior

Elementary Modules

The activity of the non-linear attractor networks (**modules**) is dominated by their **attractor states.**

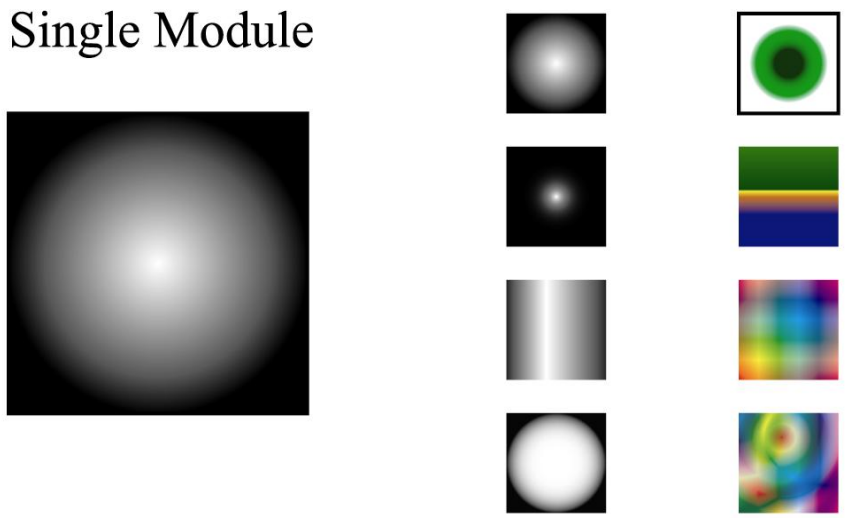
Attractor states may be **built in** or **acquired through learning.**

We **approximate** the **activity of a module** as a weighted sum of attractor states.That is: it forms an **adequate set of basis functions.**

Activity of Module:

$$\mathbf{x} = \Sigma\ c_i \mathbf{a}_i$$

where the $\mathbf{a}_i$ are the attractor states.

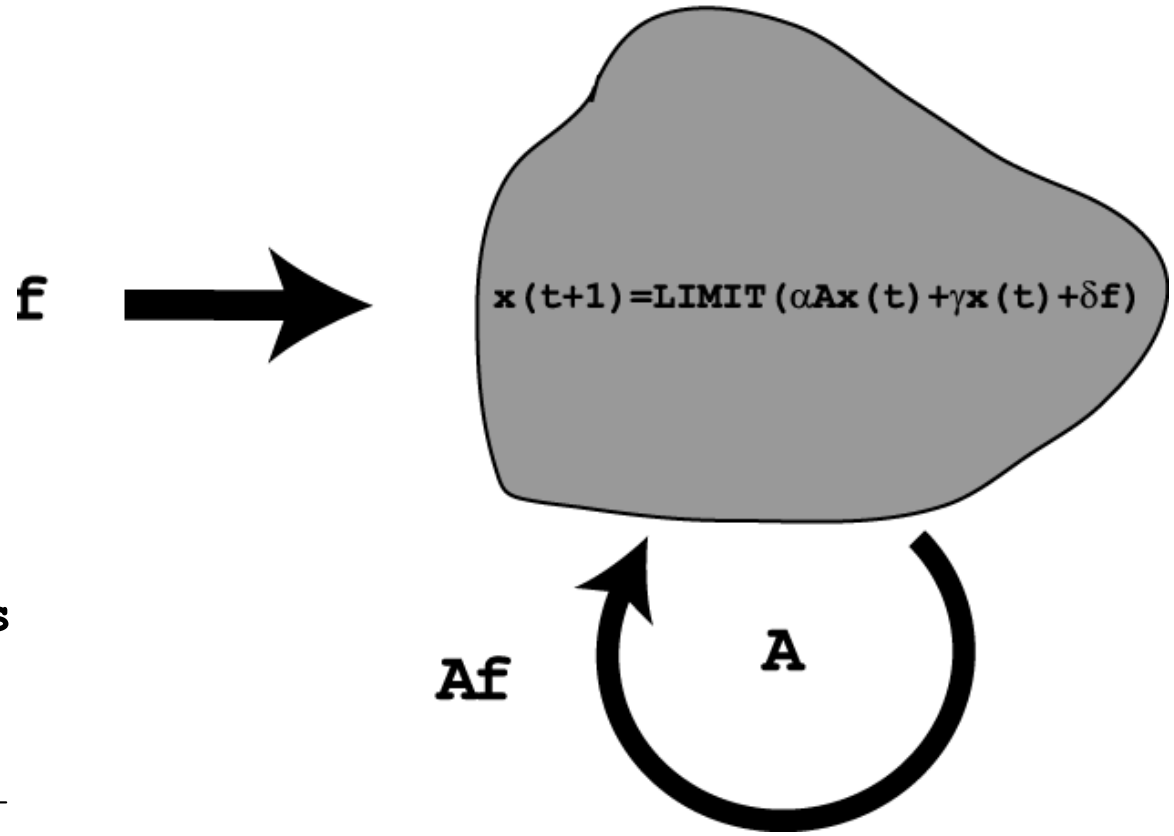


A Single Module Can Have Multiple Stable States

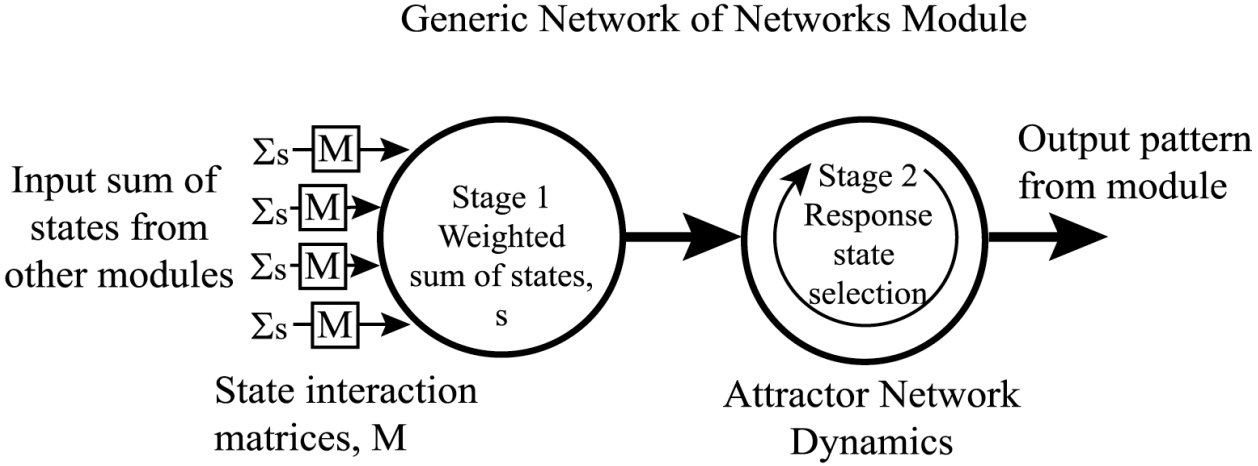
The Single Module: BSB

The attractor network we often use for the individual modules is the **BSB network** (Anderson, 1993).

It can be analyzed using the **eigenvectors** and **eigenvalues** of its local connections.



Interactions between Modules



Interactions between modules are described by **state interaction matrices, $\mathbf{M_i}$.**

The state interaction matrix elements give the **contribution** of an attractor state, $\mathbf{s_i}$, in one module to the **activity** of an attractor state in a connected module.

In the BSB **linear** region

$$\mathbf{x}(t+1) = \mathbf{\Sigma}\ \mathbf{M_i s_i} + \mathbf{f}(t) + \gamma\,\mathbf{x}(t) + \mathbf{gain}$$

weighted sum from other modules — **input** — **ongoing activity** — **control**

Computational Constraints

Sparse Connectivity

The brain is **sparsely connected.** (Unlike many abstract neural nets.)

A neuron in cortex may have on the order of **100,000** synapses. There are more than $\mathbf{10^{10}}$ neurons in the brain. Fractional connectivity is very low: **0.001%.**

May be a **major computational limitation** of the brain.

Implications:

- Connections are **expensive** biologically since they take space, use energy, and are hard to wire up correctly.
- Connections are **valuable.**
- The **pattern of connection** is under tight control.
- **Short local connections** are cheaper than **long** ones.

Ersatz approximation makes extensive use of local connections for computation.

Binding Module

Patterns Together.

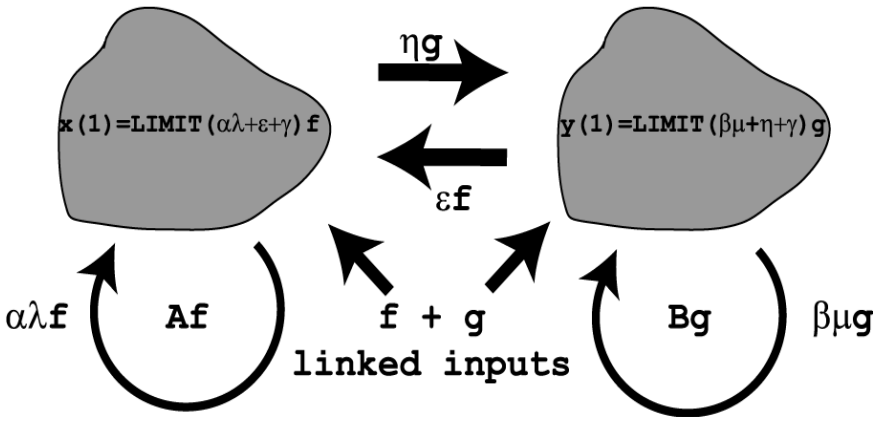
An associative **Hebbian learning event** will tend to link **f** with **g** through the local connections.

There is a speculative connection to the important **binding problem** of cognitive science and neuroscience.

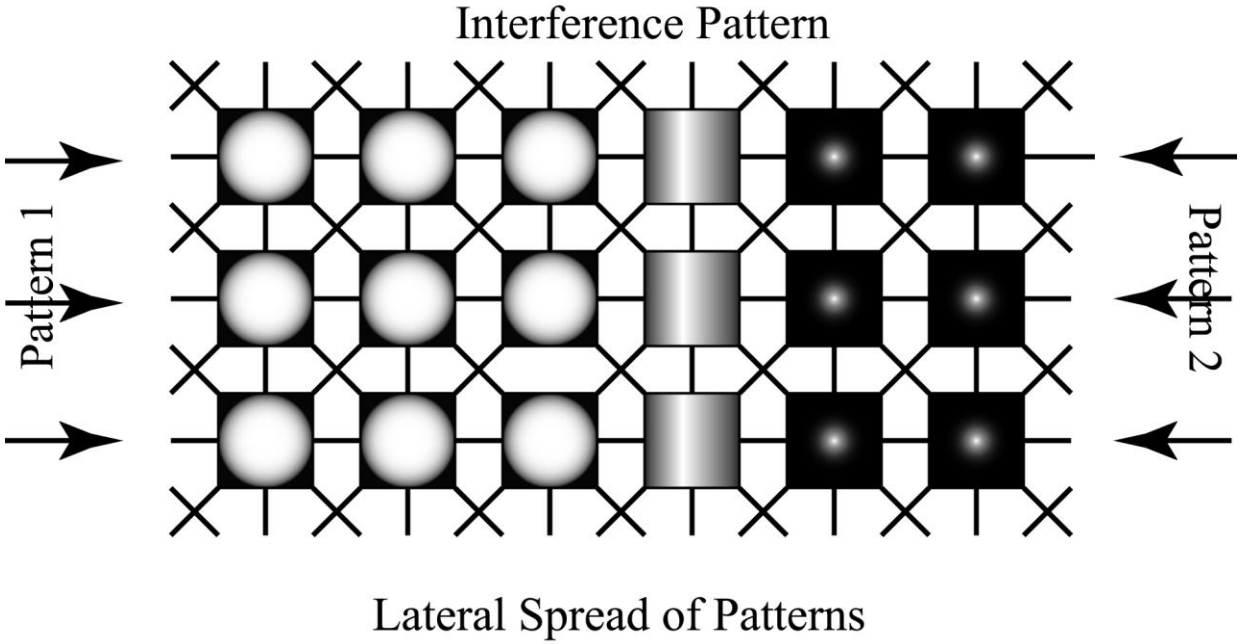
The larger groupings will act like a unit.

Responses will be stronger to the pair **f,g** than to either **f** or **g** by itself.

Two adjacent modules interacting. Hebbian learning will tend to bind responses of modules together if **f** and **g** frequently co-occur.



Interference Patterns



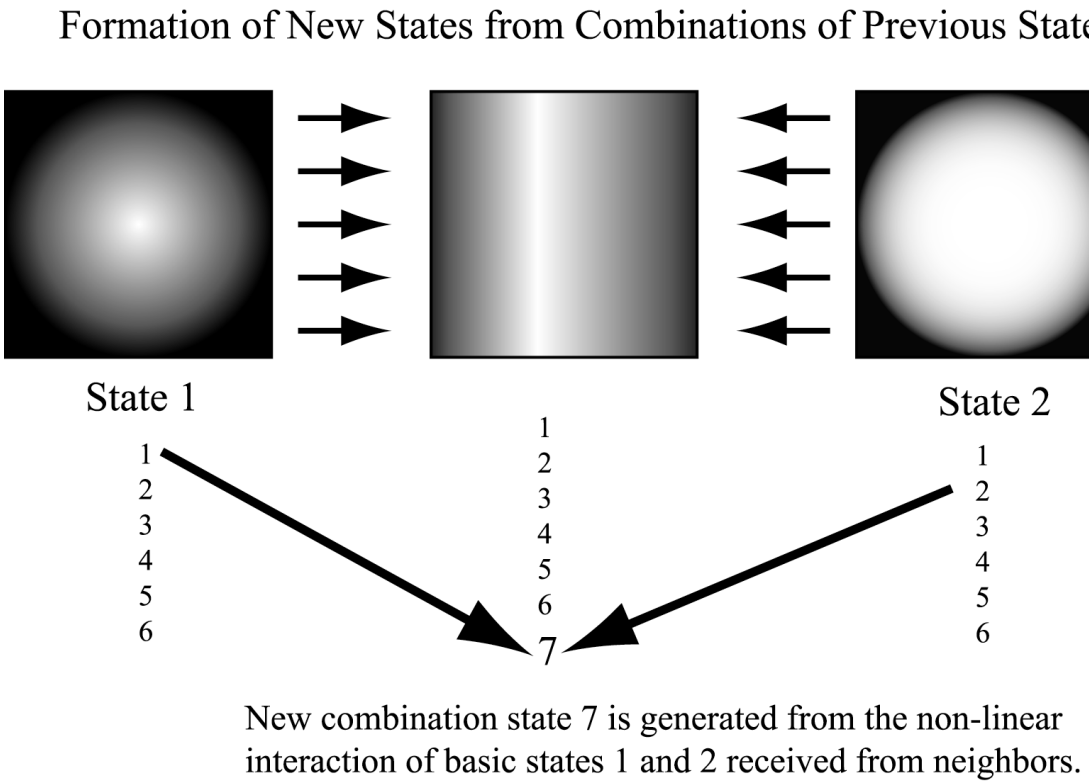
We are using **local transmission of (vector) patterns**, not **scalar activity**.

Patterns good, scalars bad: patterns allow selectivity

We have the potential for **traveling pattern waves** using the local connections.

Lateral information flow has the potential for formation of **feature combinations** in the **interference patterns** where two patterns collide.

Learning the Interference Pattern



The individual modules are **nonlinear learning networks.**

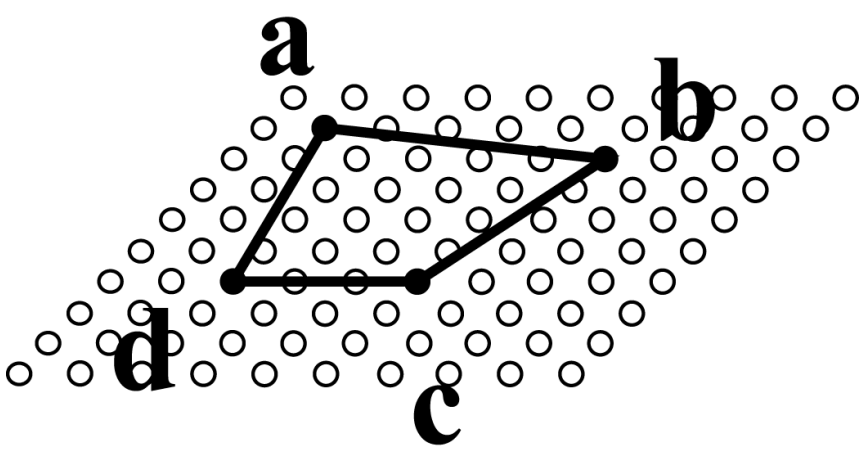
We can form **new attractor states** when an interference pattern forms when two different patterns meet at a module.

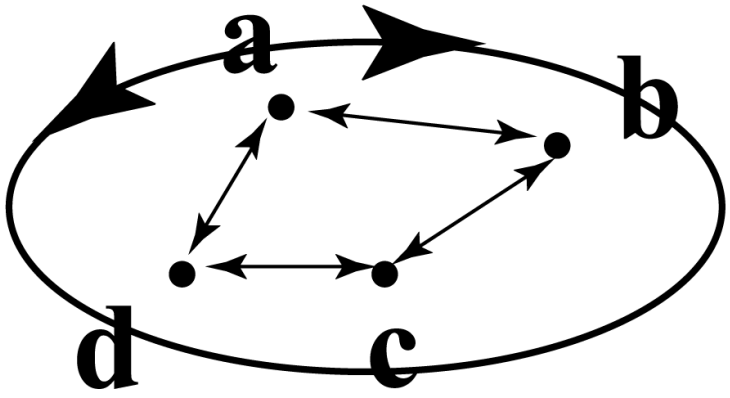
Module Assemblies

Small numbers of active selective modules in a network of networks array became **associatively linked through learning**.

Module assemblies can form through learning **self-exciting groupings** that can represent integrated information.

Form a useful intermediate level emergent entity.





Module Dynamics: Deflation Technique

Module behavior (for us) is characterized by eigenvectors, eigenvalues of an internal connection matrix **A**.

Recurrent feedback enhances amplitude of dominant eigenvector. (Power method.)

Suppose e_i is dominant module pattern with eigenvalue λ_i. of internal connection matrix **A.**

Get estimates of e_i, λ_i.

Form new matrix,

$A' = A - \lambda\, e_i\, e_i^{\,T}$ (**Form is that of Hebbian Anti-learning**)

The eigenvalue spectrum has now changed to decrease the previously dominant eigenvector which will now have a small eigenvalue.

Retrieved attractor will be different.

Can **manipulate module properties in a controlled way.**

Software

Generality: Developing software is harder and slower than building hardware!

A feature of the Ersatz EPU structure.

- Information transmission both local and long range can be **slow.**
- It takes **multiple steps** (a long time) to move data to distant modules.
- **This is can be a feature, not a bug.**

Implications

Must pay attention to the

Temporal aspects of module behavior

- **Communication times**
- **Module temporal dynamics**
- **The details of spatial arrangement of data affects communication times.**

Consistent with cortical neuroscience.

Implication: One way to "program" the array is to use these "analog" spatial and temporal properties to control computation.

Module Evolution

Module evolution with learning:

- From an **initial repertoire** of basic attractor states in the modules (as dynamical systems)

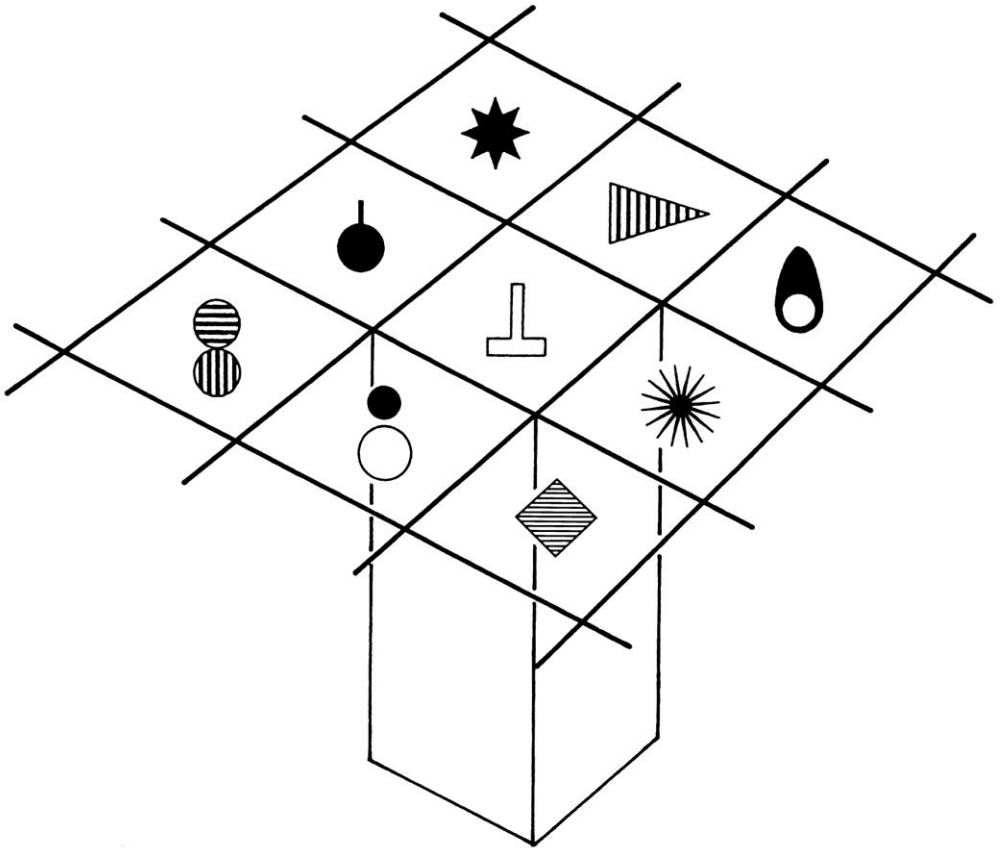
- to the development through slow **lateral pattern movement** and **associative learning** of **specialized pattern combination** states **unique** to each module.

Biological Evidence

Biological Evidence: Columnar Organization in Inferotemporal Cortex

Tanaka (2003) suggests a columnar organization of different response classes in primate **inferotemporal cortex.**

There may be internal structure in these regions: for example, spatial representation of the orientation of the image in the column.

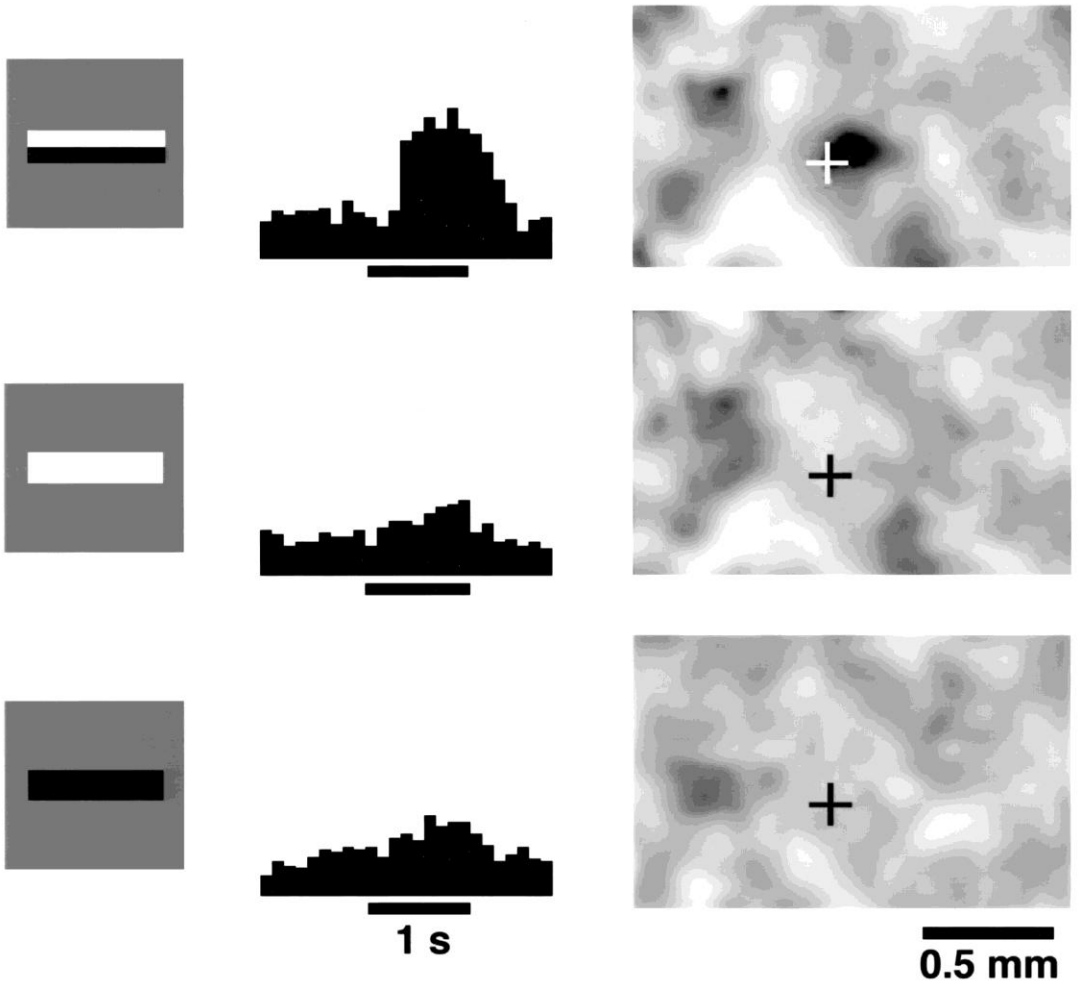


IT Response Clusters: Imaging

Tanaka (2003) used intrinsic visual imaging of cortex. Train video camera on exposed cortex, cell activity can be picked up.

At least a factor of twenty higher resolution than fMRI.

Size of response is around the size of functional columns seen elsewhere: **300-400 microns.**



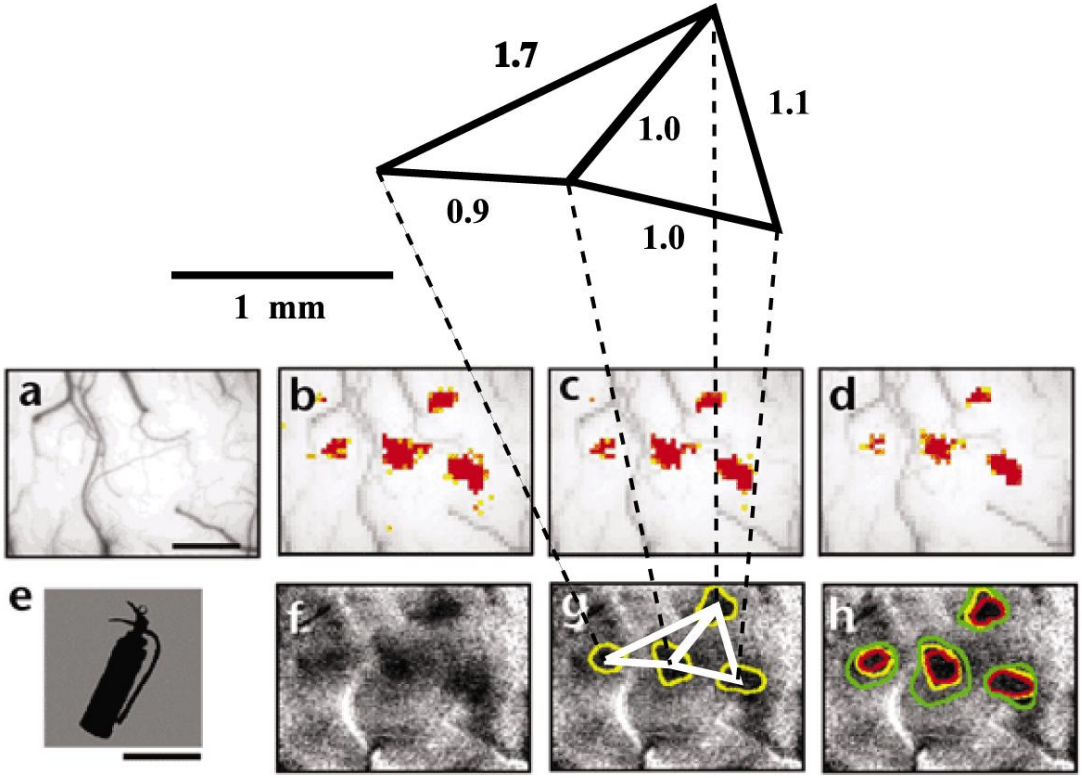
Columns: Inferotemporal Cortex

Responses of a region of IT to complex images involve **discrete columns.**

The response to a picture of a fire extinguisher shows how regions of activity are determined.

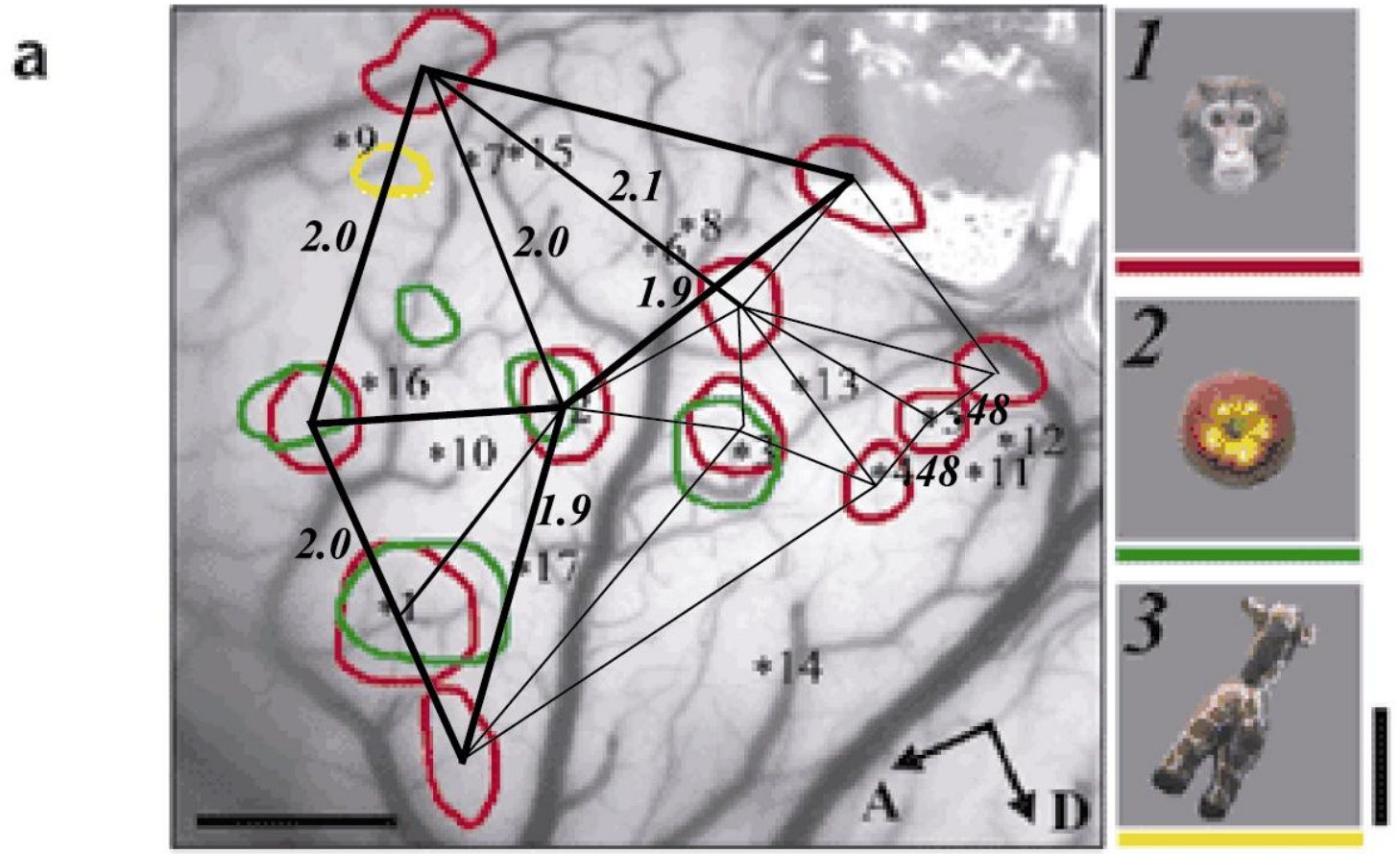
Boundaries are where the activity falls by a half.

Note: some spots are roughly equally spaced.



K Tsunoda, Y Yamane, M Nishizaki, and M Tanifuji (2001). Complex objects are represented in macaque inferotemporal cortex by the combination of feature columns. *Nature Neuroscience*, **4**, 832-838.

Active IT Regions for a Complex Stimulus



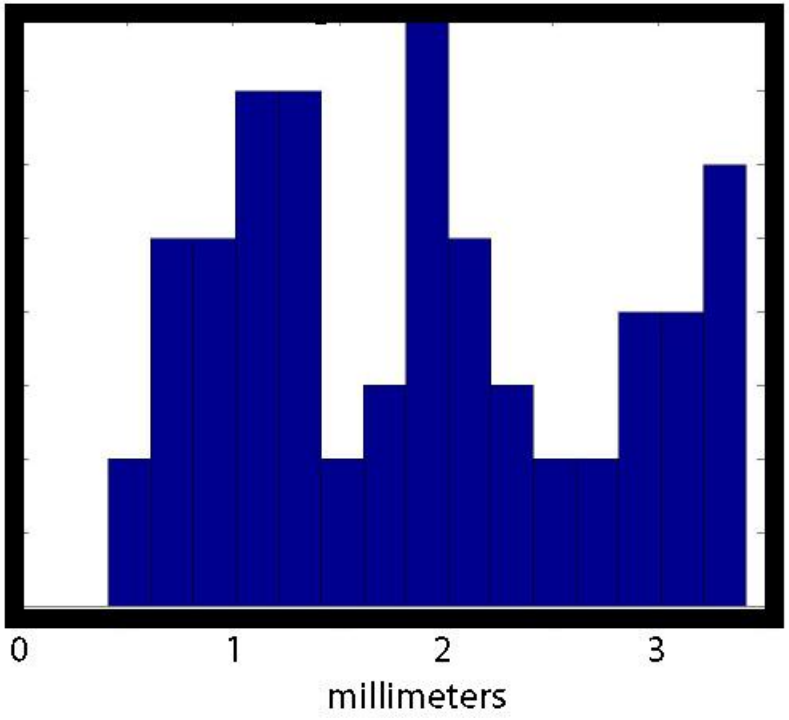
K Tsunoda, Y Yamane, M Nishizaki, and M Tanifuji Nature Neuroscience, 4, 832-838

Note the large number of roughly equally distant spots (2 mm) for a familiar complex image.

Histogram of Distances

We plotted histograms of distances in published IT intrinsic images of complex figures.

Distances computed from data in previous figure (Dimitriadis)



Back-of-the-Envelope Engineering Considerations

Engineering Hardware Considerations

We feel that there is a size, connectivity, and computational power **sweet spot** at the level of the parameters of the network of network model.

If an **elementary attractor network** has **10^4 actual neurons**, that network might have **50 attractor** states. Each elementary network might **connect to 50 others** through **state connection matrices**.

A brain-sized system might consist of **10^6 elementary units** with about **10^{11} (0.1-1 terabyte)** numbers specifying the connections.

If **100 to 1000 elementary units** on a chip gives a total of **1,000 to 10,000 chips** in a cortex sized system. Well within the upper bounds of current technology.

Modules
(Ersatz Processing Units:EPUs)

Function of EPU Modules:

- **Simulate local integration:** Addition of inputs from outside, from other modules.
- **Simulate local network dynamics.**
- **Communications Controller:** Handle long range (i.e. not neighboring) interactions.

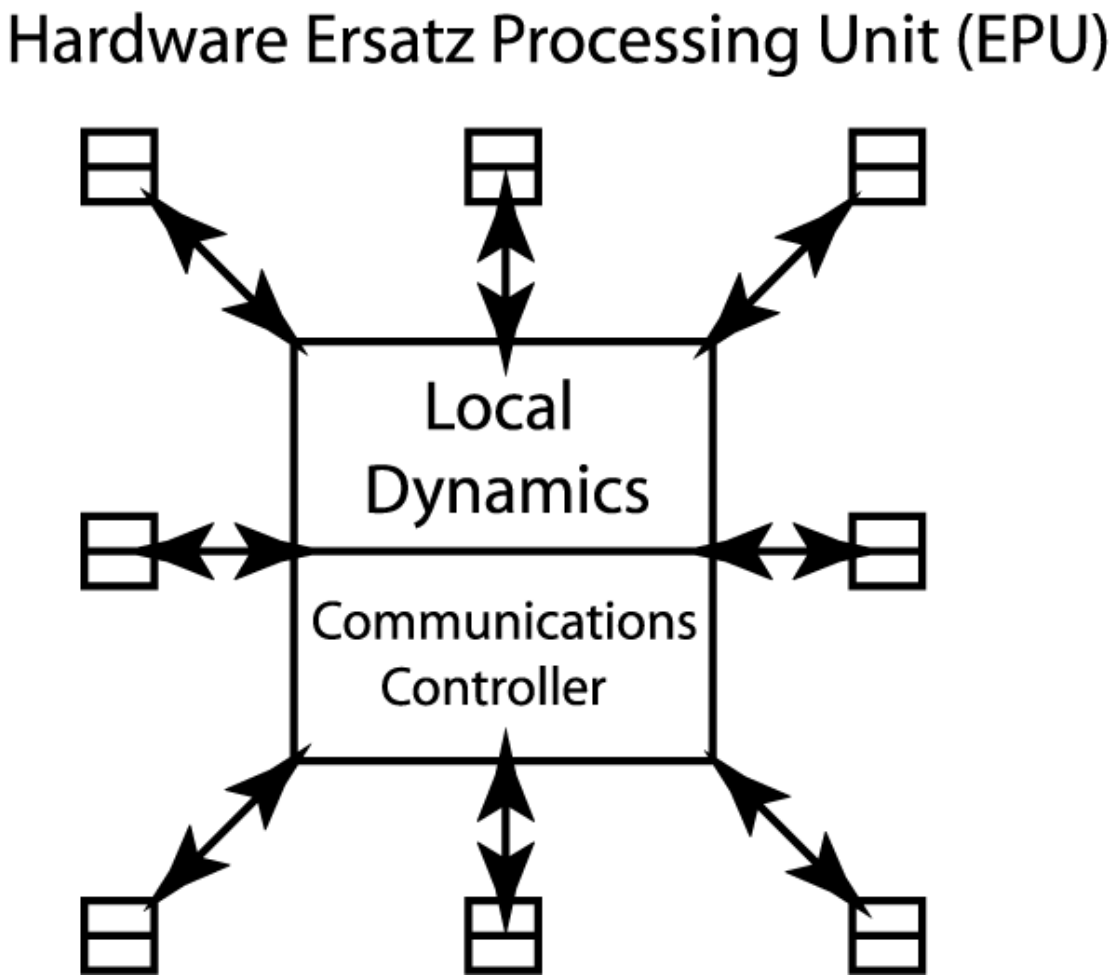
Simpler approximations are possible:

- "Cellular automaton". (Ignore local dynamics.)
- Approximations to local dynamics.

Physical (Hardware) Module

We assume only local connections for the physical hardware.

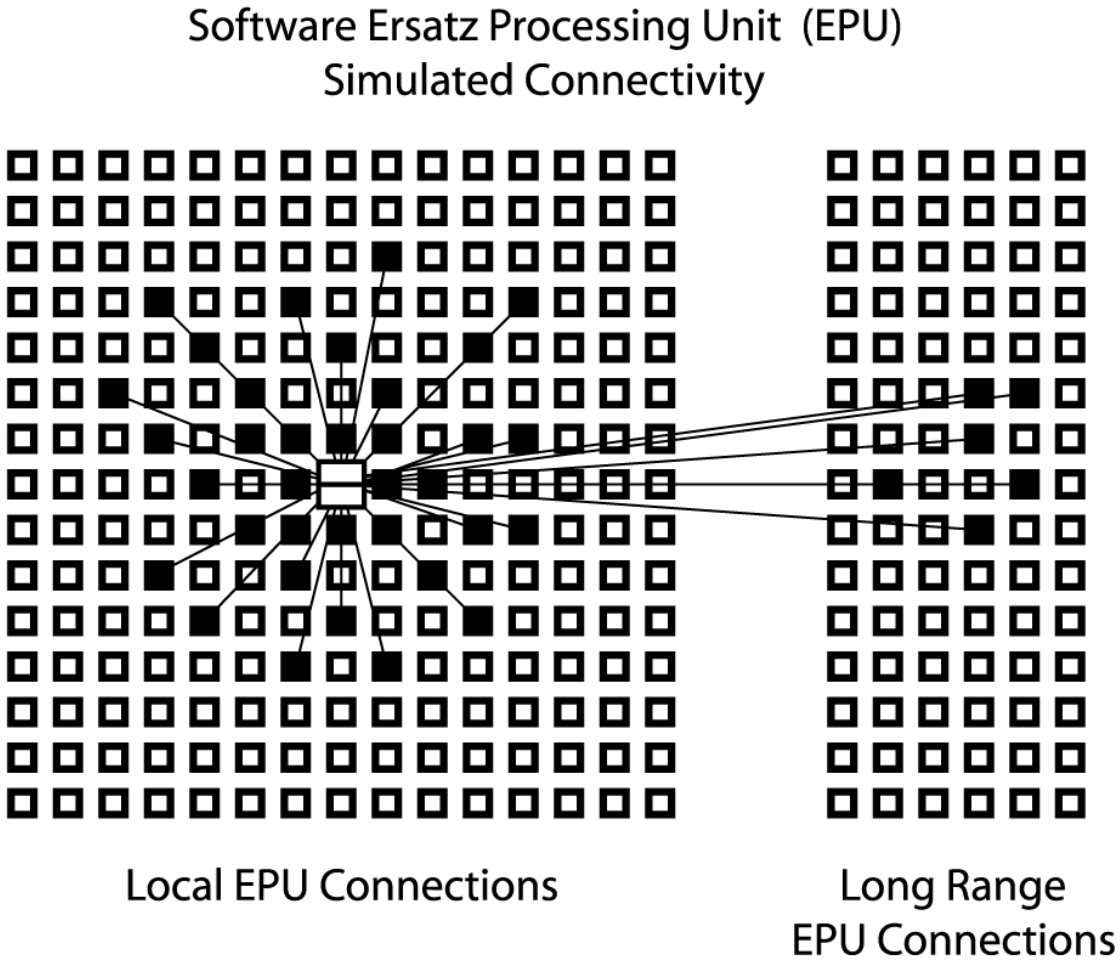
Reason:
Flexible, easy to build, easy to work with.



Software Based Connectivity

In cortex,more connections than just nearest neighbors exist.

Can simulate these with EPU module software, using the Communications Controller.



Ersatz Brain Software

Problems and Techniques (1)

Brain software can use mechanisms seen in cognitive science.

Cognitive science suggests brain-like computers should use unfamiliar techniques.

Primary "Mind" Computational mechanisms:

- Development of **useful approximations.**
- Highly effective **destructive data compression**
- Storage of **gist** as opposed to storage of **detail**
- One cognitive mechanism: **concepts, categories**
- Use of **analogy, metaphor, anecdote.**

Humans use these approximation techniques very effectively.

Problems and Techniques (2)

Limited Computational Resources also require skillful **allocation of processing resources**.

Examples:

- **Segmentation (Figure-ground)**
- **Memory based filters**
- **Adaptation to a known environment**

Continuous Control Mechanisms

- **Arousal** (Analog gain parameters)
- **Hebbian learning** (strengthen links)
- **Hebbian anti-learning** (adaptation, deflation, scanning)

Ersatz Program Discussion

Specific programs we will discuss:

Use and development of data representations

- **Computation of "identity"** and "**symmetry"**
- Speech: **Learning formant ratio invariances for vowels.**

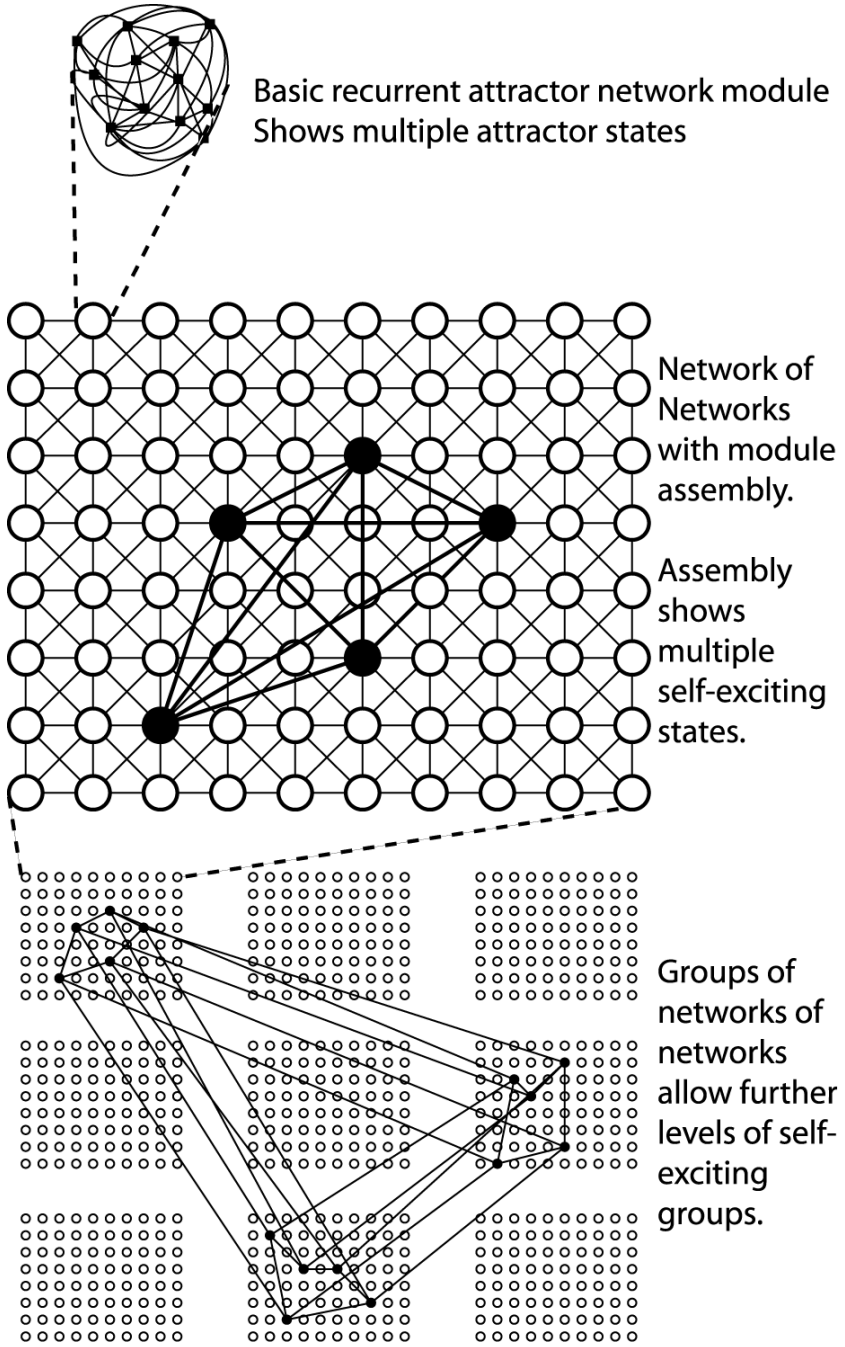
Memory Access**:**

- **Computing with a short learned list.**
- **Disambiguation**
- **Speculative extensions to anomaly detection.**

For a programmable system we need to work at multiple scales.

Module assemblies in multiple arrays work together in self exciting **“arrays of arrays.”**

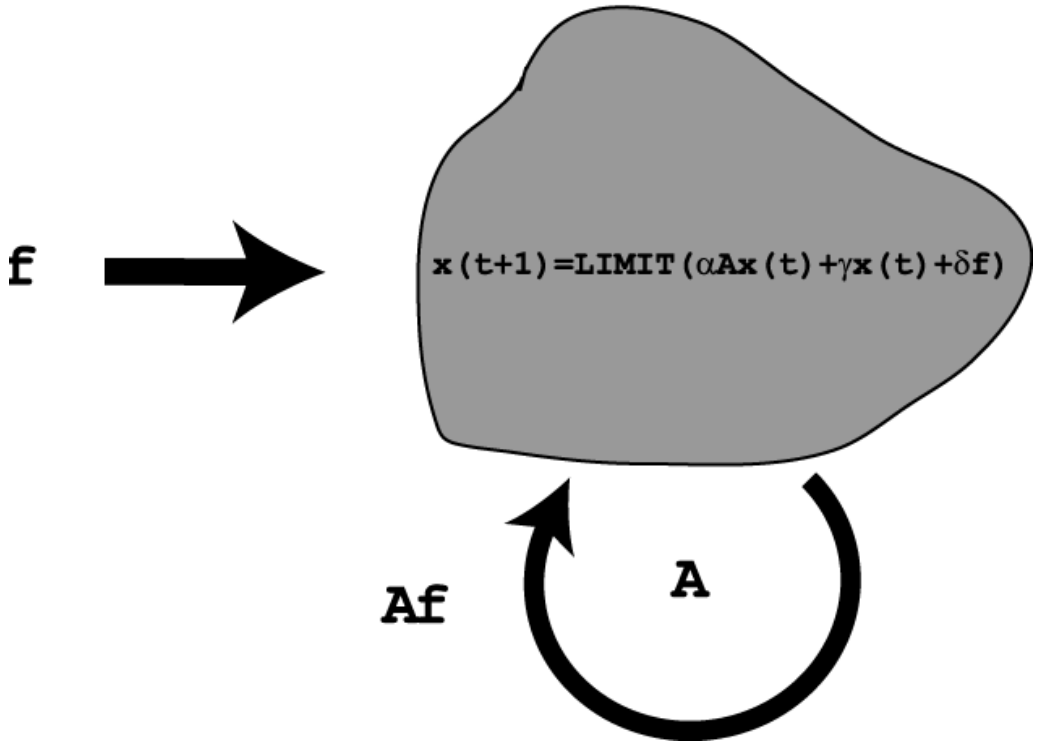
These levels are dynamical systems at different scales.



Modules

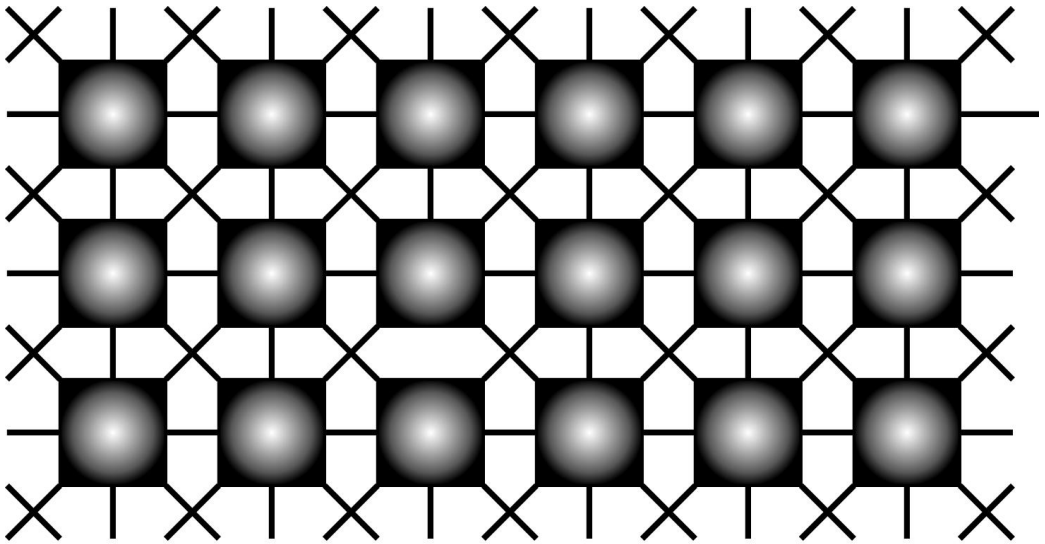
The Ersatz Brain has a natural set of functional scaling mechanisms.

Single cortical columns (**modules,** 10,000 units) are modeled as a dynamical system.



Arrays of Modules

Large two dimensional arrays of these modules formed a **network of networks.**



Network of Networks Modular Architecture

Arrays of Modules:

Topographic Computing
Identity and Symmetry

Identity

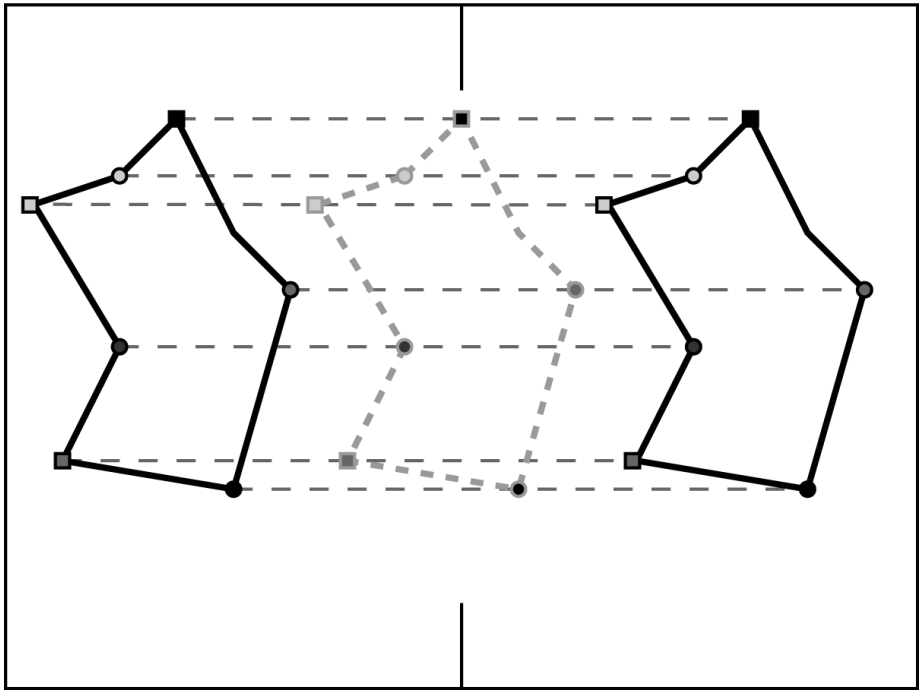
Consider simple **identity**.

Two sets of (sparsely coded) features are on a Network of Networks array **in any location**.

Are the two sets the same?

Note: The "**identity**" operation really mean to "very high degree of similarity" **not logical identity**.

Identical Figures



Pattern Spread

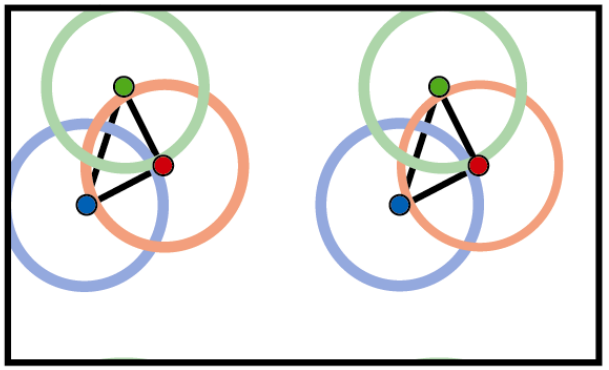
Patterns **spread laterally** from their origin.

When two of the same patterns **meet they add**.

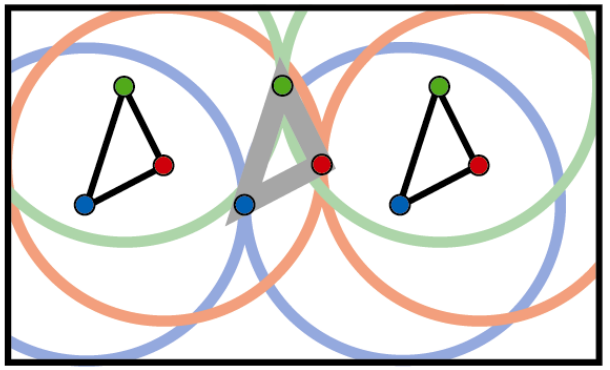
The modules at the **interference pattern** show a large response.

These spatio-temporal interactions are like those in **optics**, or **surface acoustic wave filters** [SAW].)

Identity Computation
Using Lateral Pattern Activity



Two Time Steps

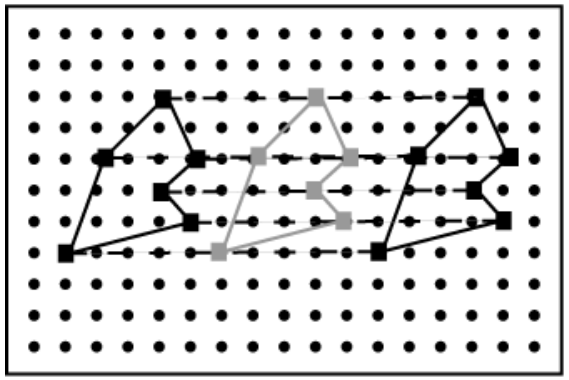


Four Time Steps.

When activity superimposes at a module, it becomes active.

When the initial patterns are identical, the active modules become active at the same instant.

Ersatz Identity Program



Step 2. Objects are topographically arranged features.

Identical objects have identical features, translated.

Lateral pattern transmission leads to interference patterns at a location half way between the original patterns.

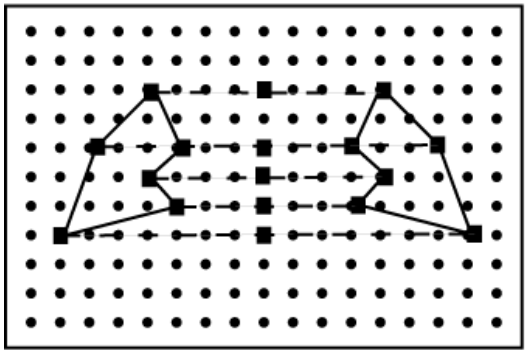
NOTE: All the interference pattern locations are activated simultaneously.

Symmetry

Use a variant of the identity program to compute symmetry

Use lateral pattern spread with different weighting of size of spatial regions and arrival times.

Ersatz Symmetry: Step 2



Step 2. Objects are topographically arranged features.
Symmetrical objects have identical features, mirror imaged.

Lateral pattern transmission leads to interference patterns at the location half way between the patterns.
Interference pattern locations are activated over a range of times.

Essence of Computation

Identity Activity:

- High amplitude, high degree of temporal localization
- Diffuse spatial location

Symmetry Activity:

- Diffuse, spread out time activation
- High degree of spatial localization between objects

Identity is also **translation independent**, symmetry is **not**.

We are using **analog filter operations** to compute a "high-level" cognitive function.

Typical of Ersatz cognitive applications: **they use both discrete and analog filter operations.**

Single Network of Networks Arrays: Vowel Formant Invariance

Another Test Problem: Speech Formant Invariances

- We want to develop a useful **data representation** inspired by the the perceptual **invariances** seen in human speech.

- Look at analyzing **vowels in a speech signal** as an example of a common class of natural signals and their transformations.

Speech Signal Basics

Vowels are long duration and often stable in their structure over time.

- Problems: **high variability of examples, different speakers, accents, dipthongs, acoustic similarity between vowels, context effects, age, gender.**
- The acoustic signals from a vowel are dominated by the **resonances** of the vocal tract, called **formants.**

Vowel Processing

Vocal tracts come in **different sizes**: men, women, children.

The resonant peaks (formants) **change their frequency** as a function of **vocal tract length**.

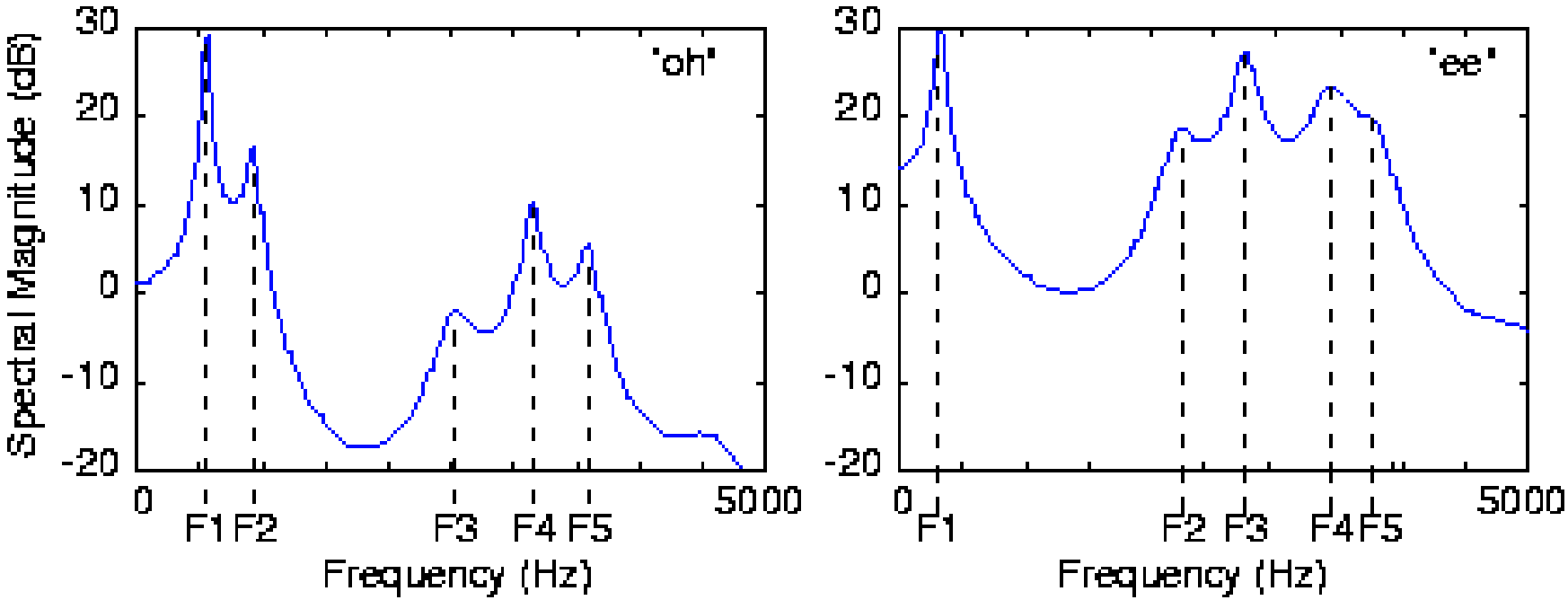
The **frequency shifts can be substantial**.

Shifts cause no problem for human speech perception.

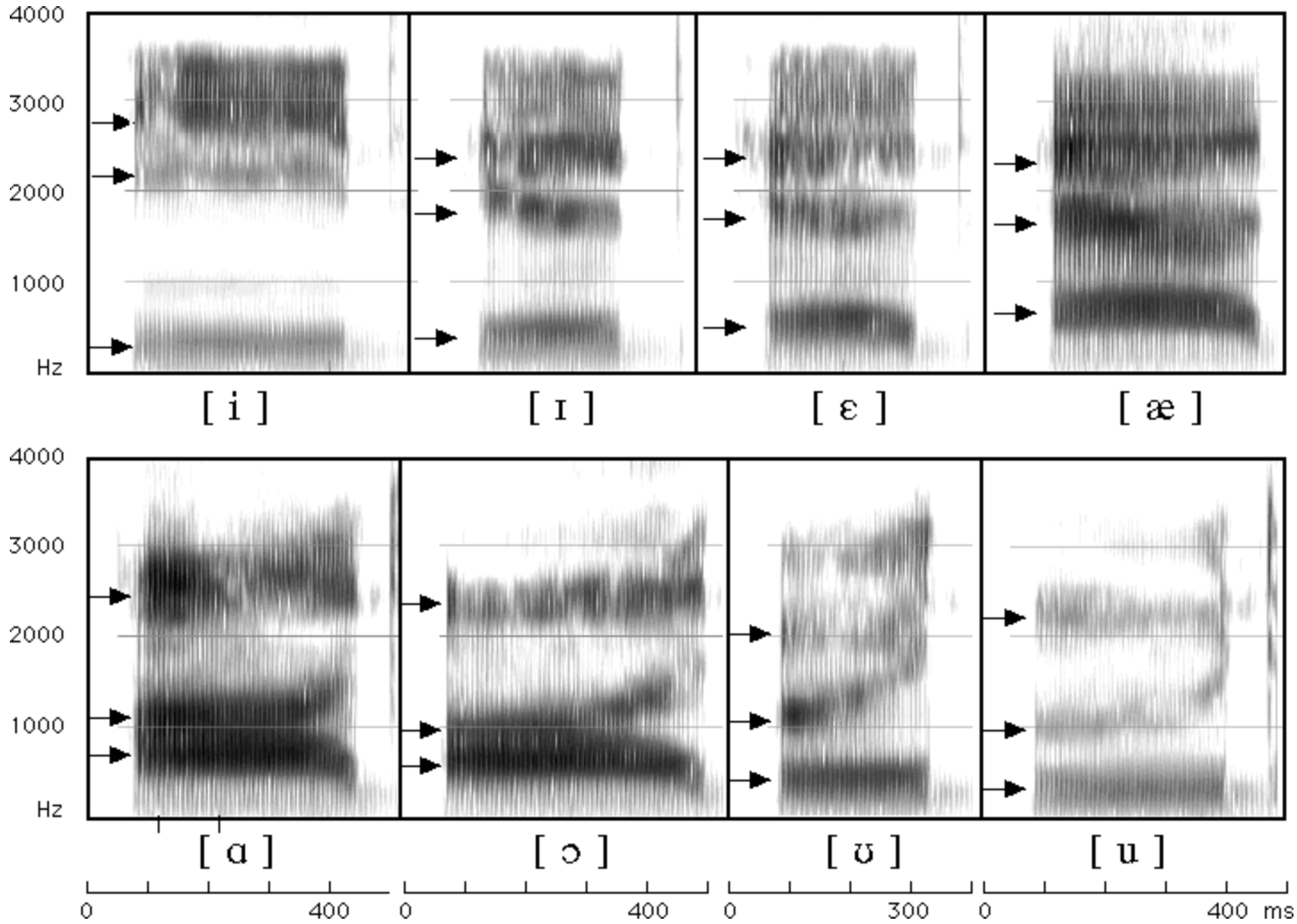
An important perceptual feature for phoneme recognition is the **ratios** between the formant frequencies, **not absolute values of frequency**.

Can we make the Ersatz system respond to ratios?

Power Spectrum of a Steady State Vowel



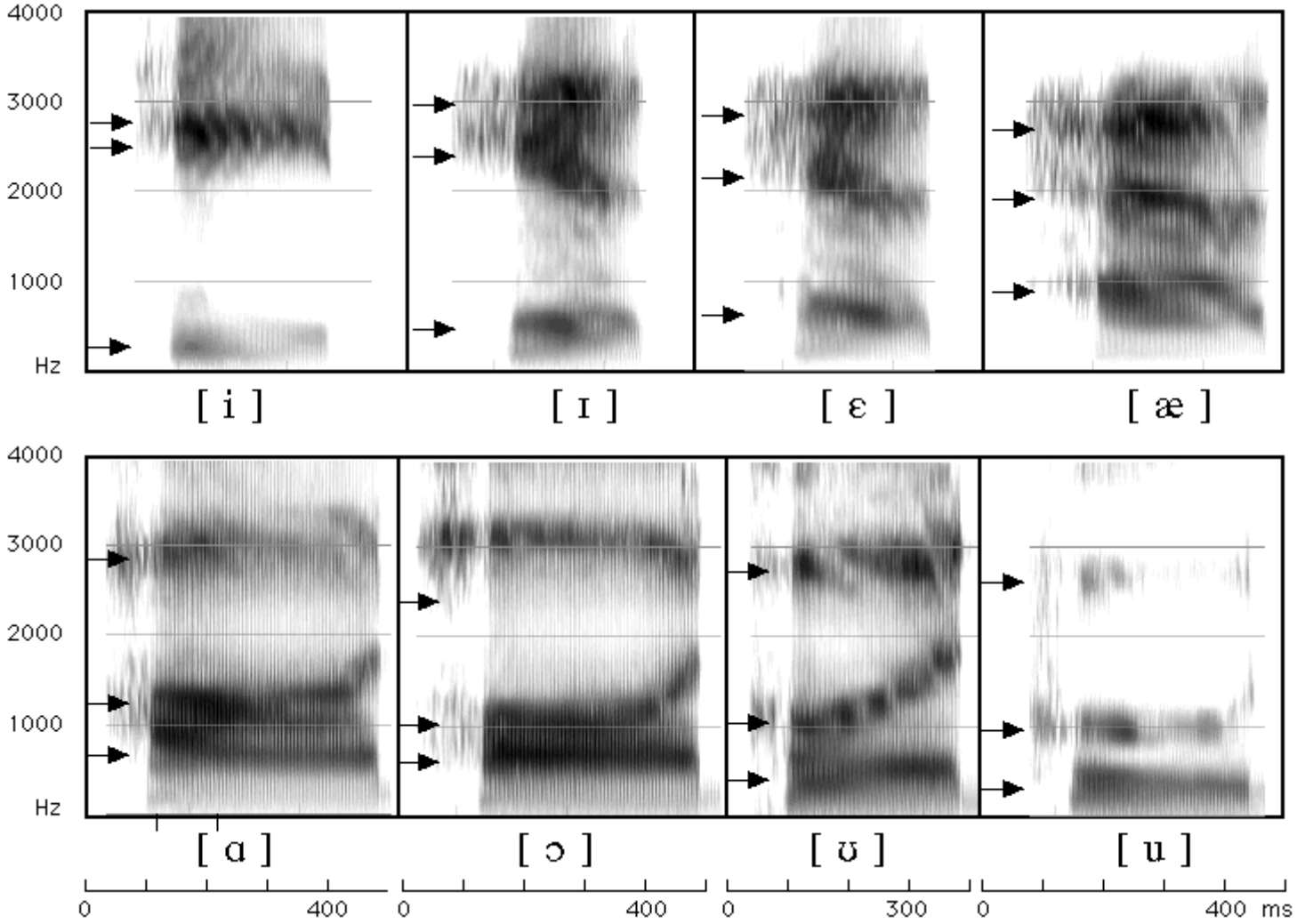
Sound Spectrogram: Male American



Words: heed, hid, head, had, hod, hawed, hood, who'd

From: P Ladefoged (2000), A Course in Phonetics, 4th Edition, Henle

Sound Spectrogram: Female American



Words: heed, hid, head, had, hod, hawed, hood, who'd

From: P Ladefoged (2000), A Course in Phonetics, 4th Edition, Henle

Average Formant Frequencies for Men, Women and Children.

		[i]		[æ]		[u]	
Men	F1	267	(0.86)	664	(0.77)	307	(0.81)
Women	F1	310	(1.00)	863	(1.00)	378	(1.00)
Children	F1	360	(1.16)	1017	(1.18)	432	(1.14)
Men	F2	2294	(0.82)	1727	(0.84)	876	(0.91)
Women	F2	2783	(1.00)	2049	(1.00)	961	(1.00)
Children	F2	3178	(1.14)	2334	(1.14)	1193	(1.24)
Men	F3	2937	(0.89)	2420	(0.85)	2239	(0.84)
Women	F3	3312	(1.00)	2832	(1.00)	2666	(1.00)
Children	F3	3763	(1.14)	3336	(1.18)	3250	(1.21)

Small sample of data set from Watrous (1991) derived originally from **Peterson and Barney** (1952).

Ratios Between Formant Frequencies (Hz) for Men, Women and Children.

		[i]	[æ]	[u]
Men	F1/F2	0.12	0.38	0.35
Women	F1/F2	0.11	0.42	0.39
Children	F1/F2	0.11	0.43	0.36
Men	F2/F3	0.78	0.71	0.39
Women	F2/F3	0.84	0.72	0.36
Children	F2/F3	0.84	0.70	0.37

Small sample of data set taken from Watrous (1991) derived originally from **Peterson and Barney** (1952).

Other Representation Issues

There is a roughly **logarithmic spatial mapping** of **frequency** onto the surface of auditory cortex.

Maps of sound frequency with location are called **tonotopic** maps.

A logarithmic spatial coding has the effect of **translating** the parameters multiplied by the constant **the same distance.**

Log transformations are common in sensory systems.

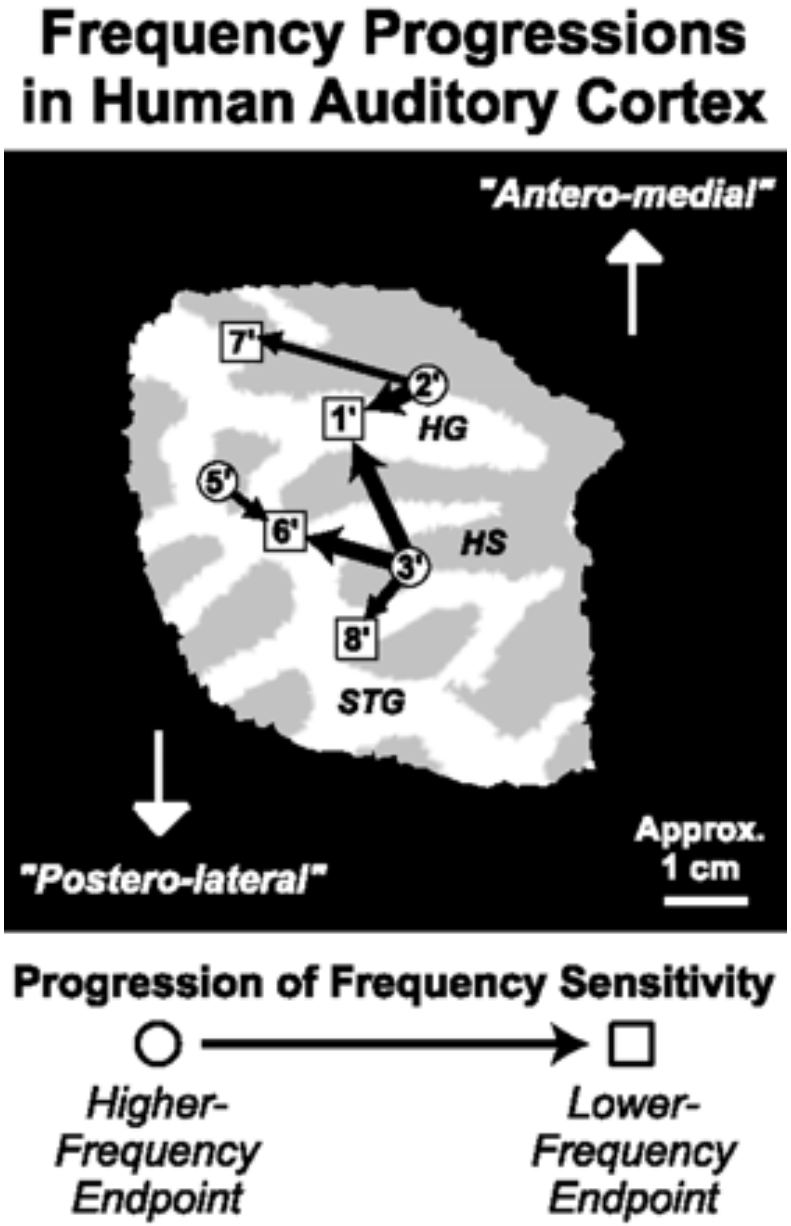
Multiple Maps

Human fMRI derived tonotopic maps in human auditory cortex.

Note at least **five**, probably **six** maps.

Some joined at high frequency end and some at low frequency end.

(Figure 6 from Talavage, et al., p. 1290)



Ratio Filtering

Our computational goal:

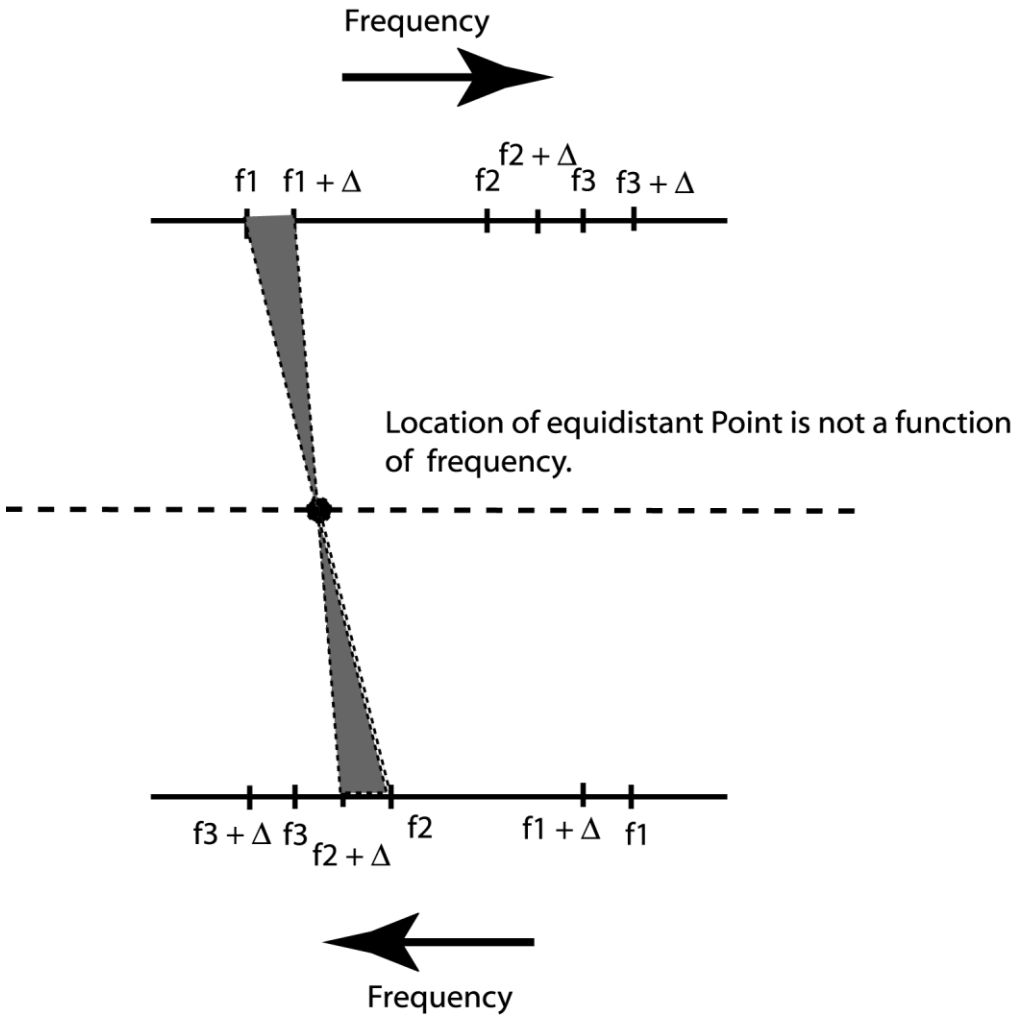
Enhance the representation of **ratios** between formant frequencies

De-emphasize the exact values of those frequencies.

We wish to make a filter using a data representation that responds to one aspect of the input data.

Simple Topographic System To Represent Relationships

- Simplest system: Two opposing maps of frequency.
- Look at points equally distant between f1 on one map and f2 on the other.
- These are points of equal transmission times.)
- Shift frequency by constant amount, Δ.
- The point of equal distance between new frequencies (f1+Δ) and (f2+Δ) does not move.

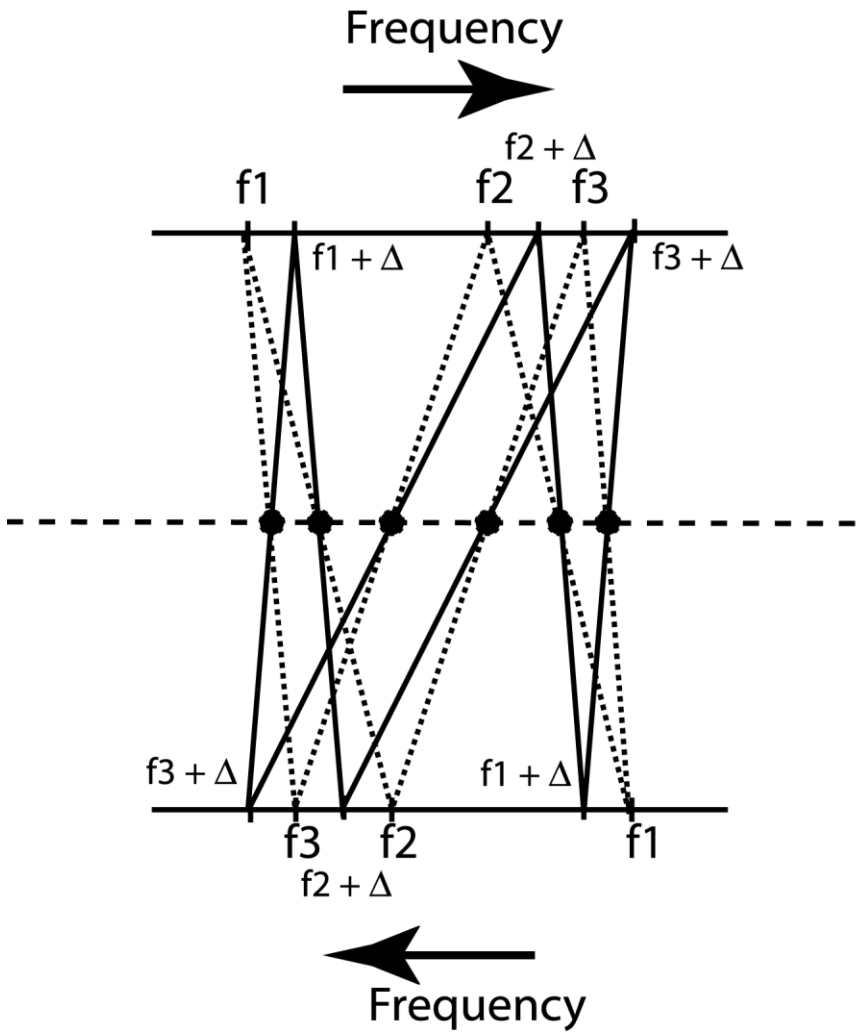


Multiple Formants

Multiple formants will have multiple relations represented topographically.

Note: multiple modules in a distributed representation.

Note shift leaves invariant points at center line.



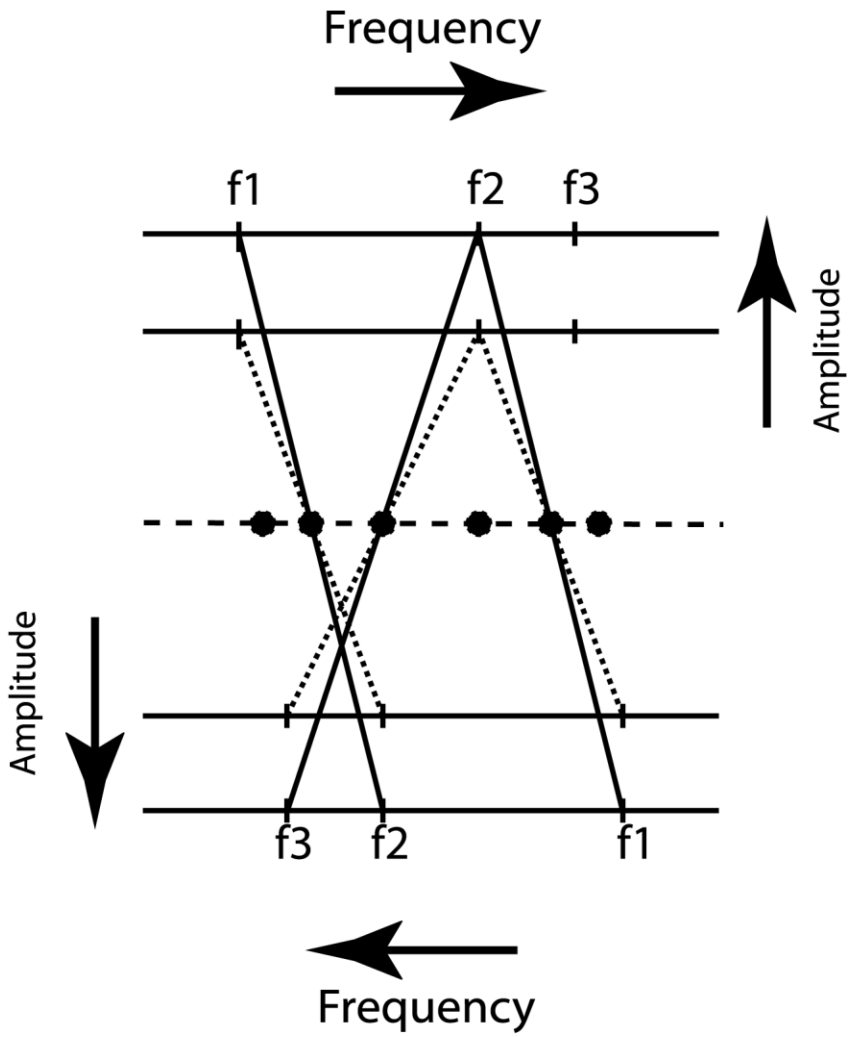
Another Acoustic Invariant Parameter: Loudness

- Auditory coding of amplitude is not as simple as it might seem.
- Auditory cortex cells respond to a **region of amplitude and frequency.**
- They respond to a **small range of frequencies (Q~12).**
- But they also respond to **amplitude over a limited (10-30 db) range.**
- **Cells turn off if the signal is too soft, too high, too loud, or too low.**
- Suppose we assume **amplitude maps** perpendicular to frequency maps.

Multiple Parameters

We can apply our representation trick to two parameters simultaneously.

Two maps moving in opposite directions do not change the invariant points.

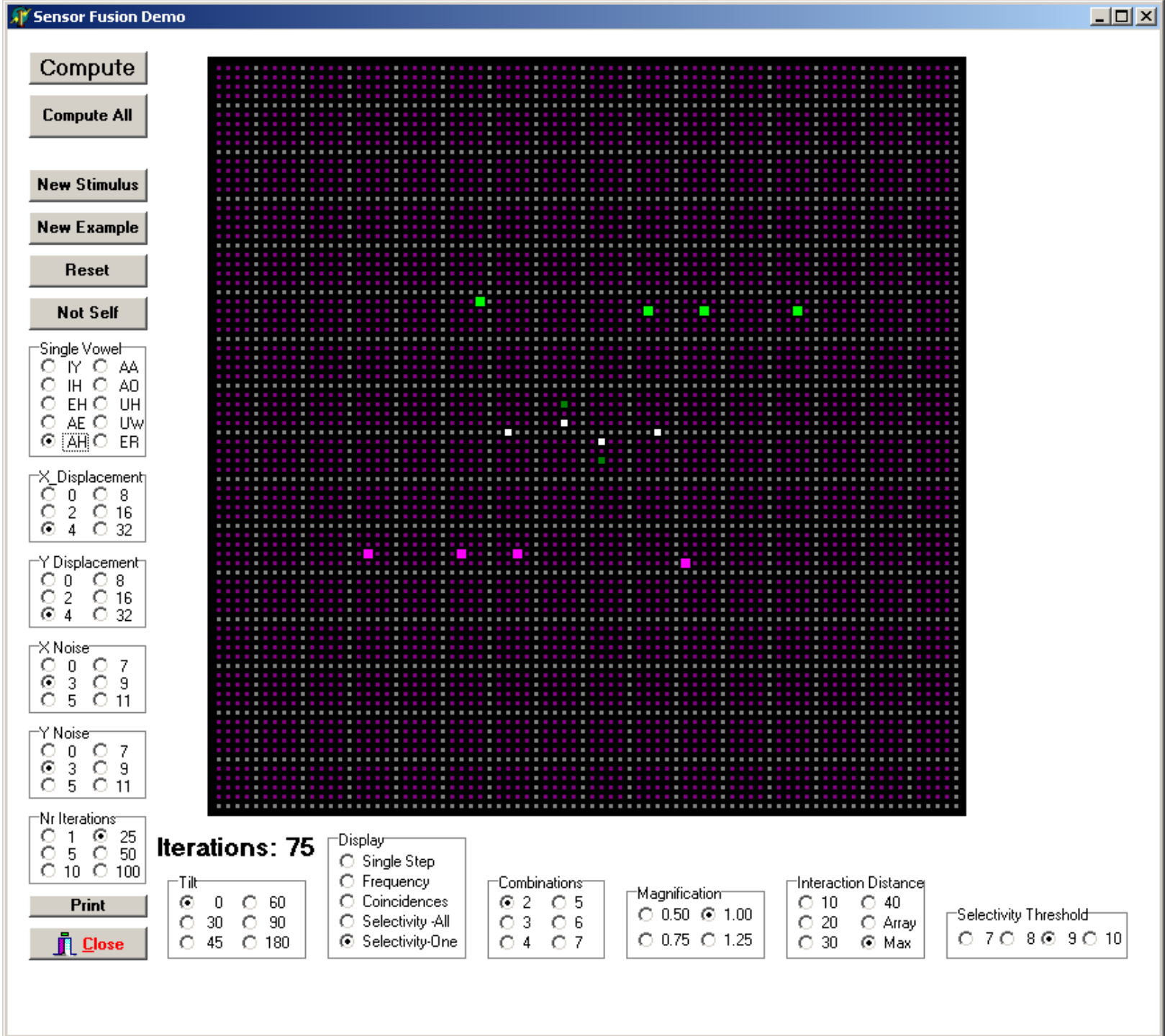


Peterson and Barney Data Simulation

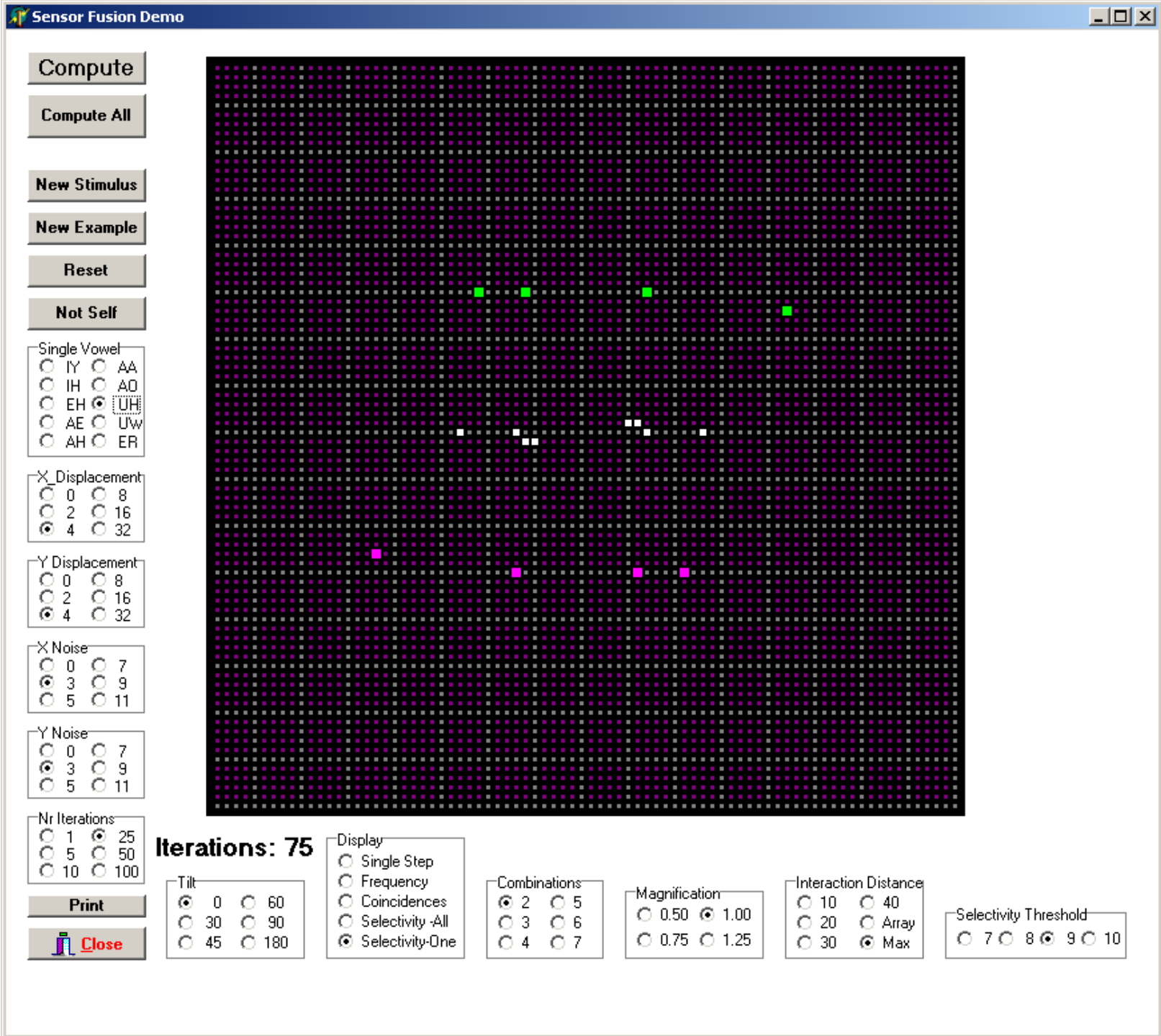
- Simulation learned 10 vowels, with examples from the “male”, “female” and “child” categories.
- Total of 75 examples.
- Included added shifts corresponding to both ***x*** and ***y*** translations. (I.e. noise).
- White squares are **invariant** modules: modules responding strongly to only one vowel category.
- Develops a somewhat **sparse** representation.
- **Invariant** modules arise primarily as a result of the initial topography of the data representation.

Vowel "AH"

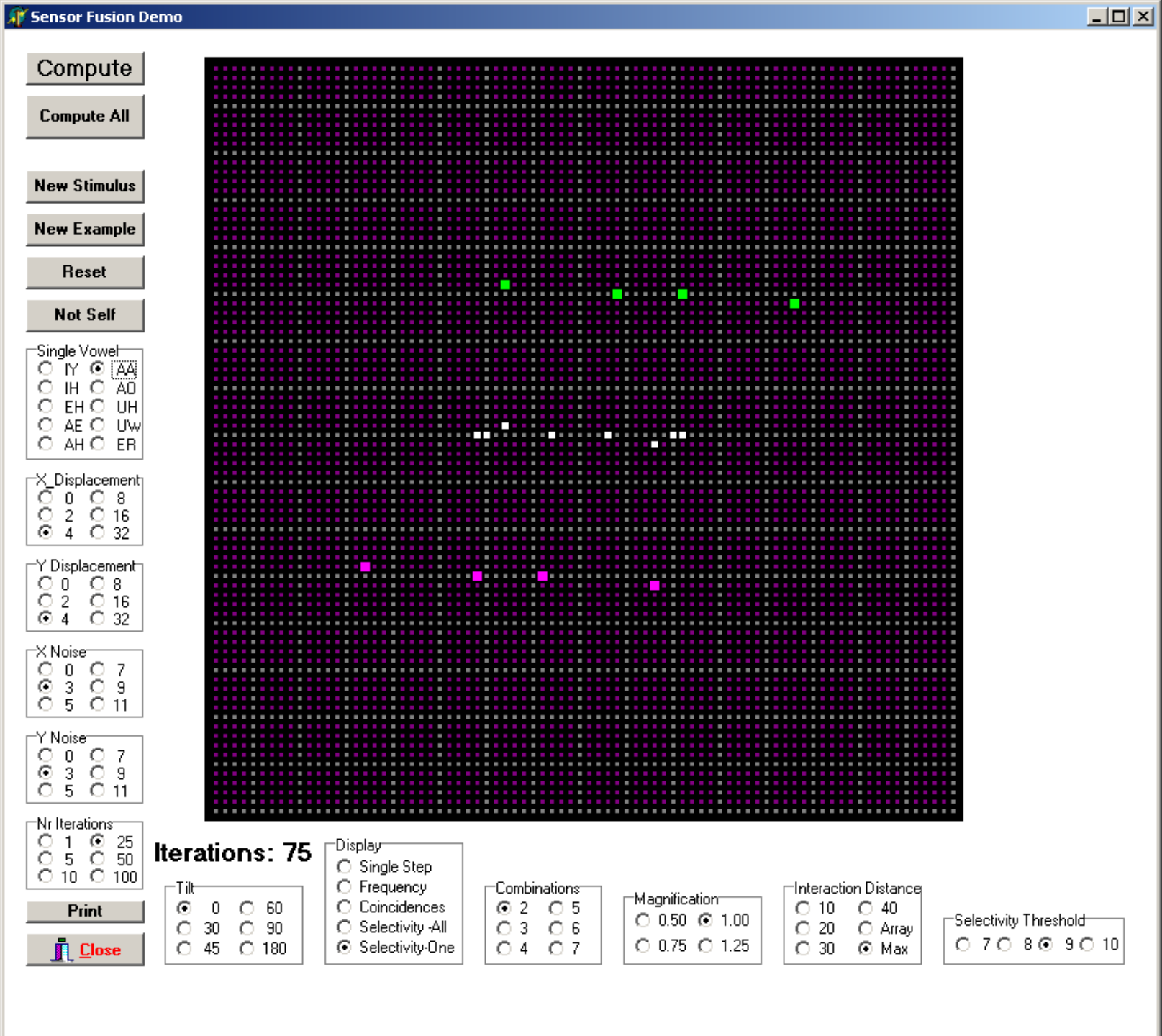
The four green and four purple squares are an example of one input pattern for that vowel category.



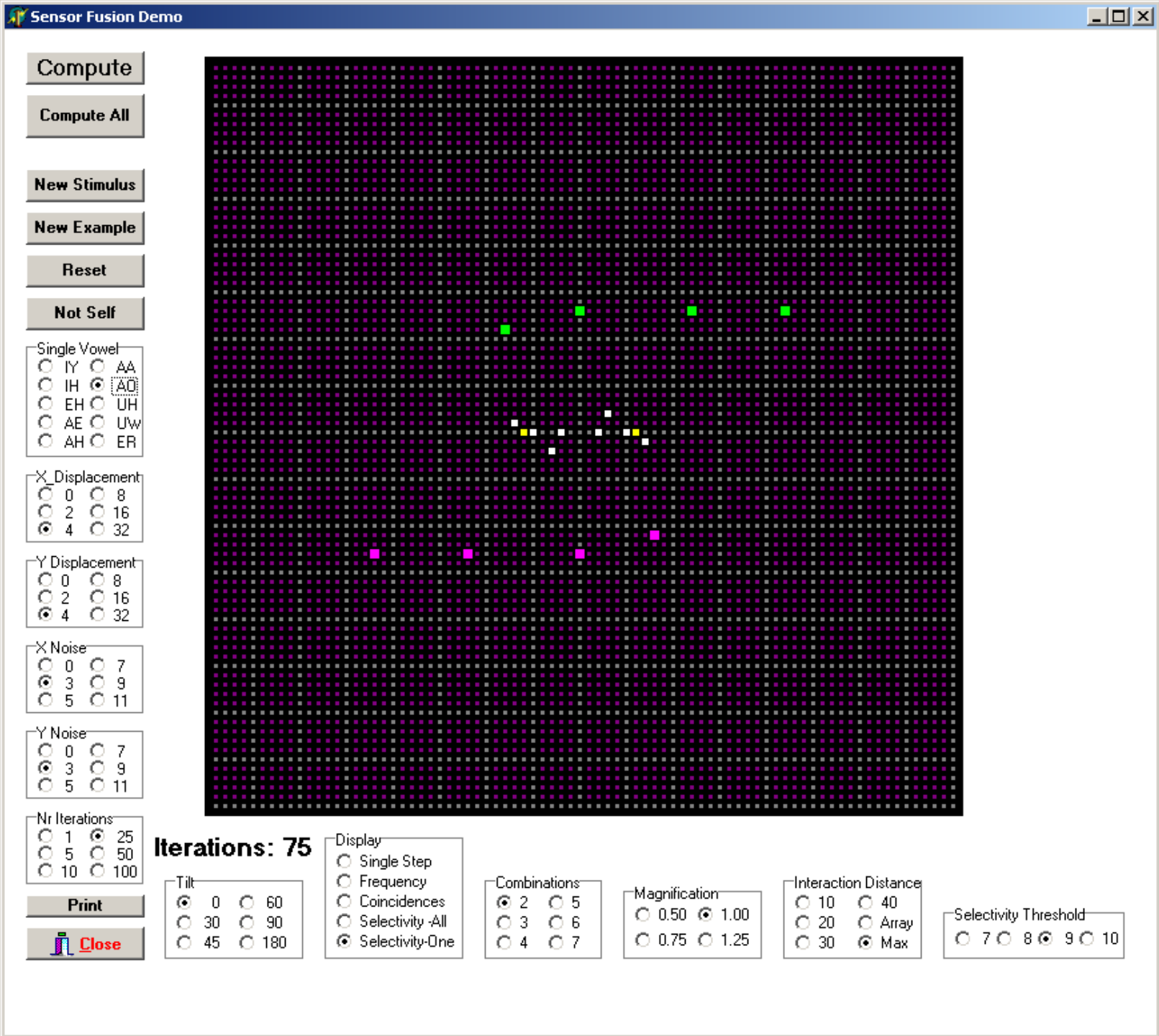
Vowel
"UH"


Sensor Fusion Demo
Compute
Compute All
New Stimulus
New Example
Reset
Not Self
Single Vowel
IY AA
IH AO
EH UH
AE UW
AH ER
X_Displacement
0 8
2 16
4 32
Y Displacement
0 8
2 16
4 32
X Noise
0 7
3 9
5 11
Y Noise
0 7
3 9
5 11
Nr Iterations
1 25
5 50
10 100
Print
Close
Iterations: 75
Tilt
0 60
30 90
45 180
Display
Single Step
Frequency
Coincidences
Selectivity -All
Selectivity-One
Combinations
2 5
3 6
4 7
Magnification
0.50 1.00
0.75 1.25
Interaction Distance
10 40
20 Array
30 Max
Selectivity Threshold
7 8 9 10

Vowel
“AA”


Sensor Fusion Demo
Compute
Compute All
New Stimulus
New Example
Reset
Not Self
Single Vowel
IY AA
IH AO
EH UH
AE UW
AH ER
X_Displacement
0 8
2 16
4 32
Y Displacement
0 8
2 16
4 32
X Noise
0 7
3 9
5 11
Y Noise
0 7
3 9
5 11
Nr Iterations
1 25
5 50
10 100
Print
Close
Iterations: 75
Tilt
0 60
30 90
45 180
Display
Single Step
Frequency
Coincidences
Selectivity -All
Selectivity-One
Combinations
2 5
3 6
4 7
Magnification
0.50 1.00
0.75 1.25
Interaction Distance
10 40
20 Array
30 Max
Selectivity Threshold
7 8 9 10

Vowel
"AO"


Sensor Fusion Demo
Compute
Compute All
New Stimulus
New Example
Reset
Not Self
Single Vowel
IY AA
IH AO
EH UH
AE UW
AH ER
X_Displacement
0 8
2 16
4 32
Y Displacement
0 8
2 16
4 32
X Noise
0 7
3 9
5 11
Y Noise
0 7
3 9
5 11
Nr Iterations
1 25
5 50
10 100
Print
Close
Iterations: 75
Tilt
0 60
30 90
45 180
Display
Single Step
Frequency
Coincidences
Selectivity -All
Selectivity-One
Combinations
2 5
3 6
4 7
Magnification
0.50 1.00
0.75 1.25
Interaction Distance
10 40
20 Array
30 Max
Selectivity Threshold
7 8 9 10

Arrays of
(Arrays of Modules)

Applications to Cognition:
Memory Filters
(Beyond Skinner)

Multiple Arrays

Next, consider complex dynamical systems involving **multiple NofN arrays and multiple module assemblies**.

Focus on memory use and access.

Consider next two ways to look at the operation of memory in Ersatz systems.

First is **simple pattern recognition**. Useful but inflexible.

- Call it "**passive memory**."

The second is **task based memory access**. Can be programmed and is flexible and powerful.

- Call it **"active memory."**

Passive Memory

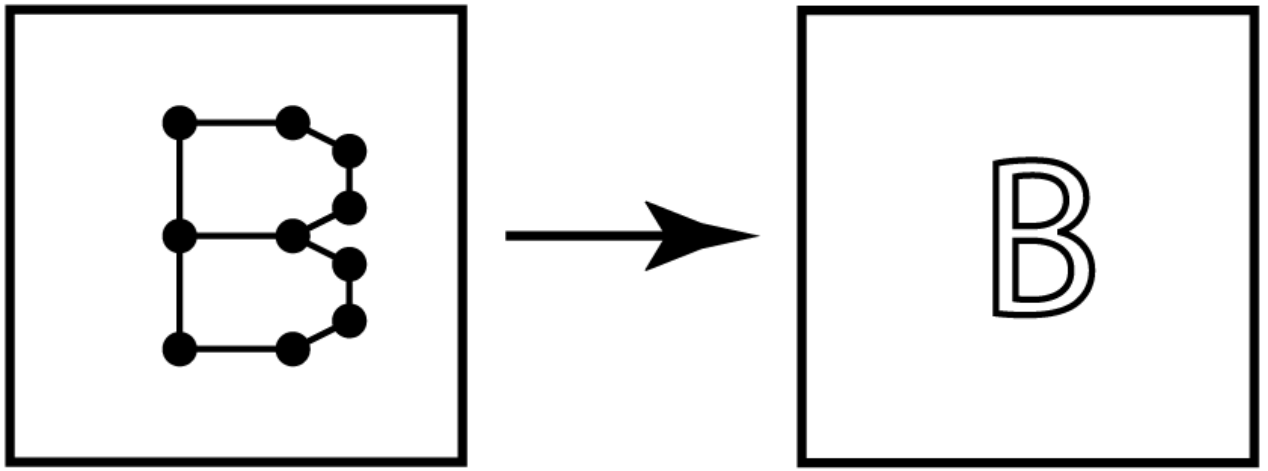
Many pattern recognition (and neural network models) STILL assume a **stimulus-response** (**S-R**) structure for learning and behavior.

Based on 1920's behaviorism.

Psychology spent 50 years showing **S-R** psychology is inadequate to explain human behavior. (Resulting in the "cognitive revolution" of the 1970's.)

Passive Memory

Simple Associative Link



Input 'B' Feature Set

'B' Module Assembly

Simple Association: Basic neural net feed-forward associative link).
Input: Set of topographically organized features.
Output: Module assembly activation

Version of **classic pattern recognition** structure. **Inflexible**, with some limited cognitive computational power. **We can do better**.

More Flexible: Active Memory

In **active memory**, memory and sensory data from **different NofN arrays** can mix together at another array.

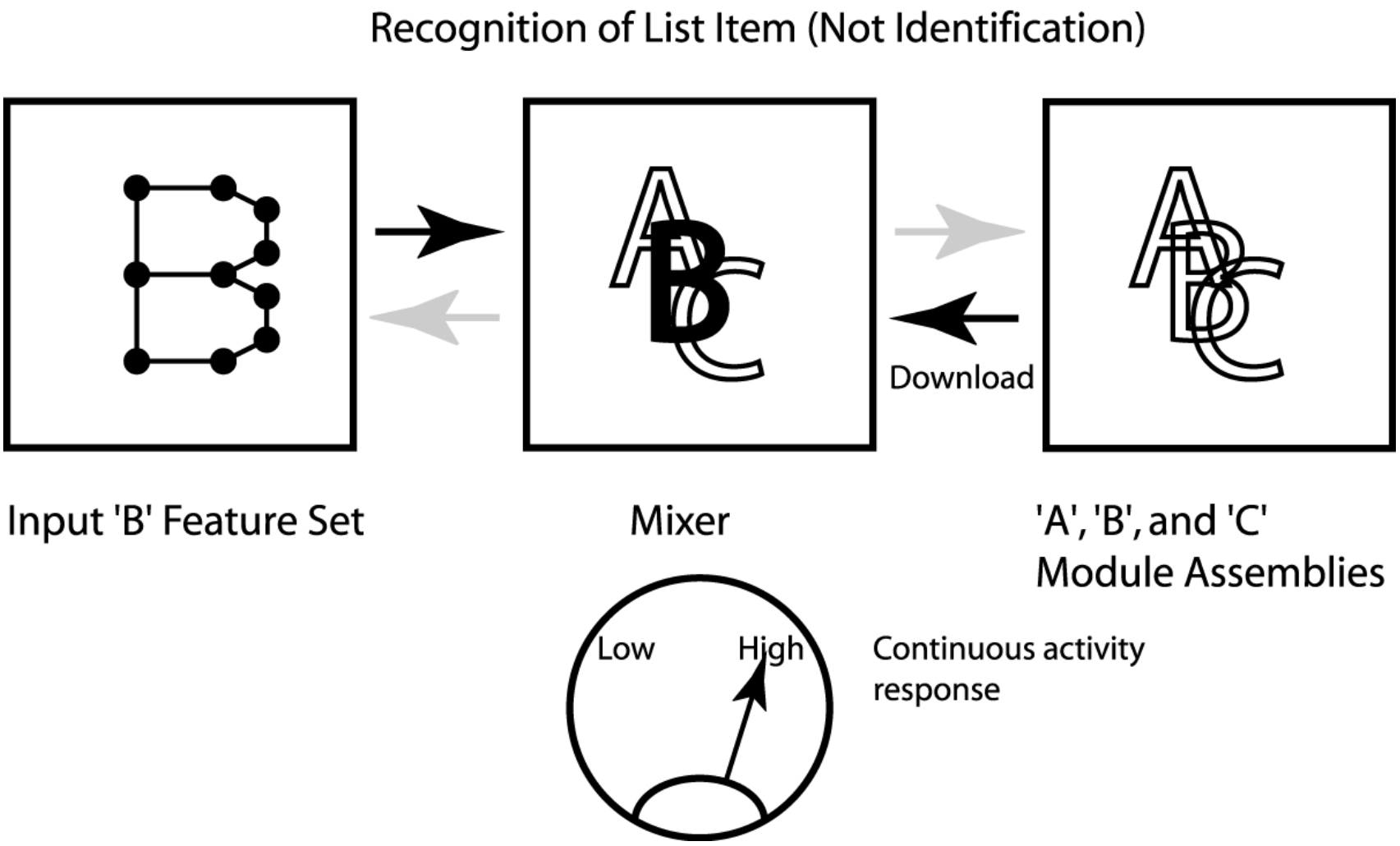
We call this array the **mixer.**

The **activity level at the mixer** can provides the result of the computation.

Such an architecture can be:

- Flexible.
- Task dependent.
- Programmable.

Active Memory: Three NofN Arrays



Fast Decision Systems

One goal of the Ersatz project is to make an artificial system with these desirable properties:

- Be **fast** (work in a very small number of steps. No tree searches.)
- Be able to **integrate information from many sources**.
- Example: **combine information** from the **senses** and from **perceptual analysis** with **past information stored in memory** in a **fast, efficient, programmable manner**.
- Be right most of the time.
- (As in Malcolm Gladwell book: ***Blink***)
- First application: **Sternberg List Experiments**

The Sternberg Experiment

We outline next a program to perform the famous (among cognitive scientists!) **Sternberg List Scanning Experiments**.

Task:

- Memorize a short list of items (2 to 5 items).
- Decide quickly whether a new item was on the list or not.

The Task

Assume we learn a short list of letters.

A, B, G, K, L

K is presented.

Press the **positive response** key.

M is presented.

Press the **negative response** key.

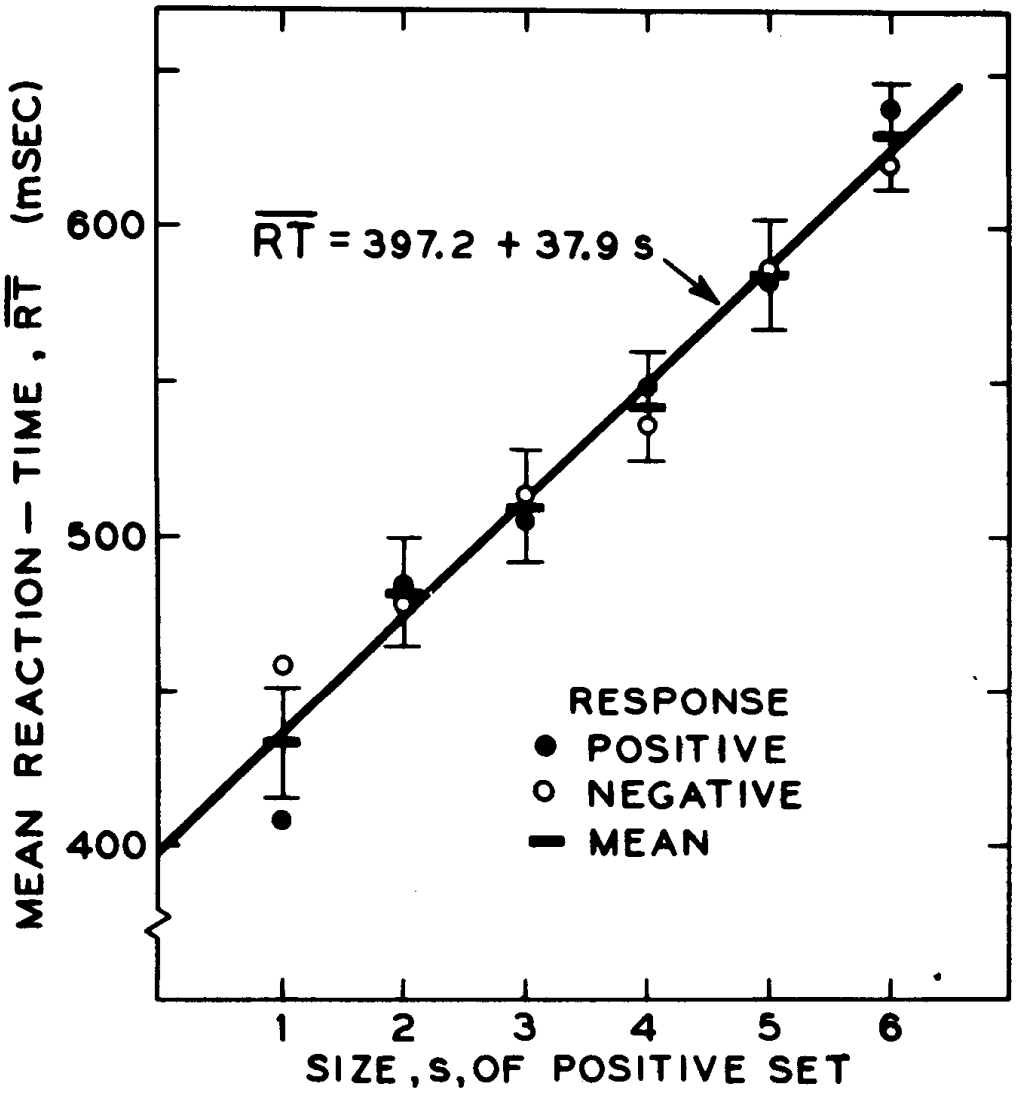
Record the time it takes to respond.

Sternberg Response Time Data

Plot average response time against list length.

Key finding:

Positive (present) and **negative** (absent) responses have the same slope.



Implications of the Data

If we scan the items sequentially, we will scan **half the list items** on average for a positive response.

But we must scan **all the items** for a negative response.

The slope of the **negative responses** should grow **twice as fast** as the **positive responses.**

Must **scan more items for negative responses.**

Not found.

Not Trivial!

This task is not trivial.

Similar to many real world tasks.

Parallel Search through huge data sets:

Is MANTY a word?

Need less than one second to answer.

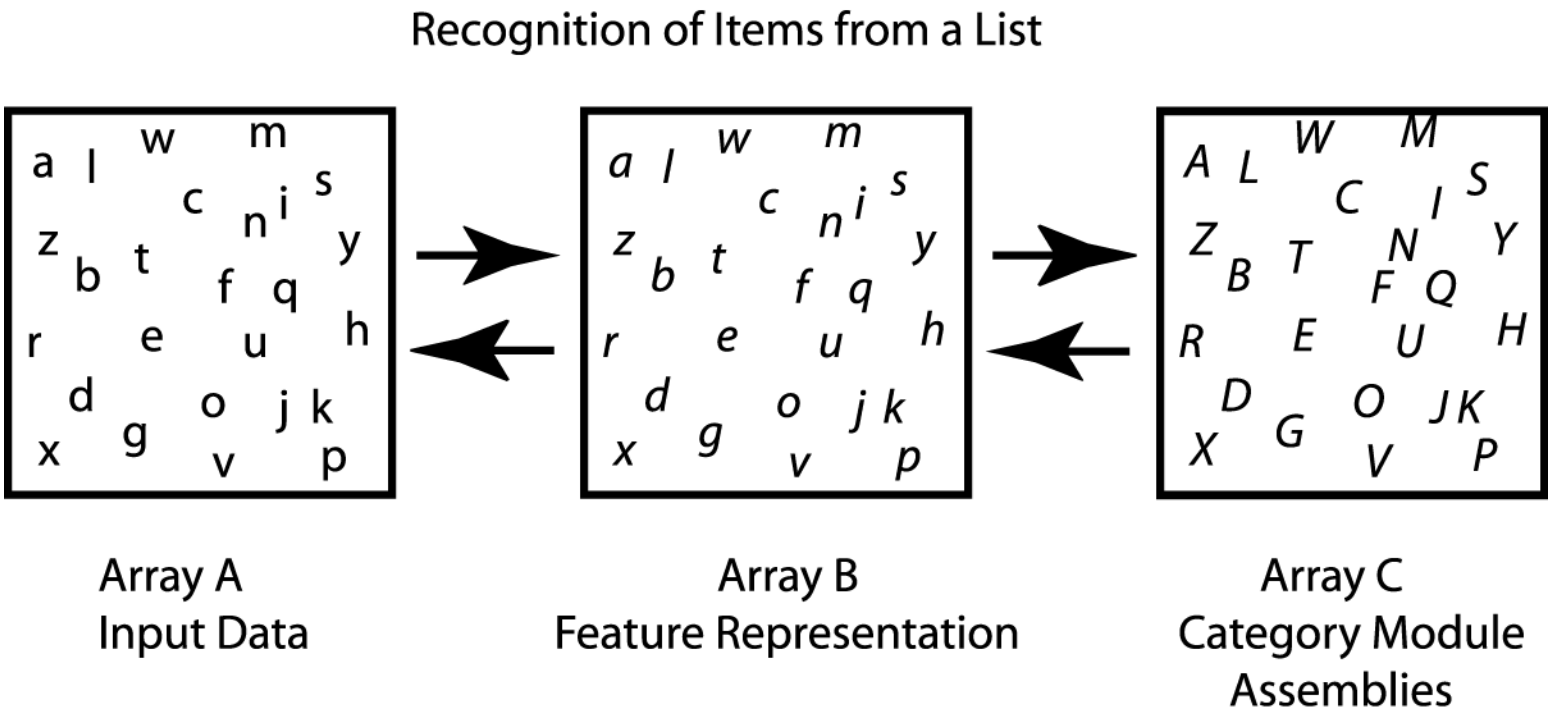
Extension to "**imprecise**" and "**poorly formed**" queries to data bases are of great interest to many.

Humans are exceedingly good at such tasks.

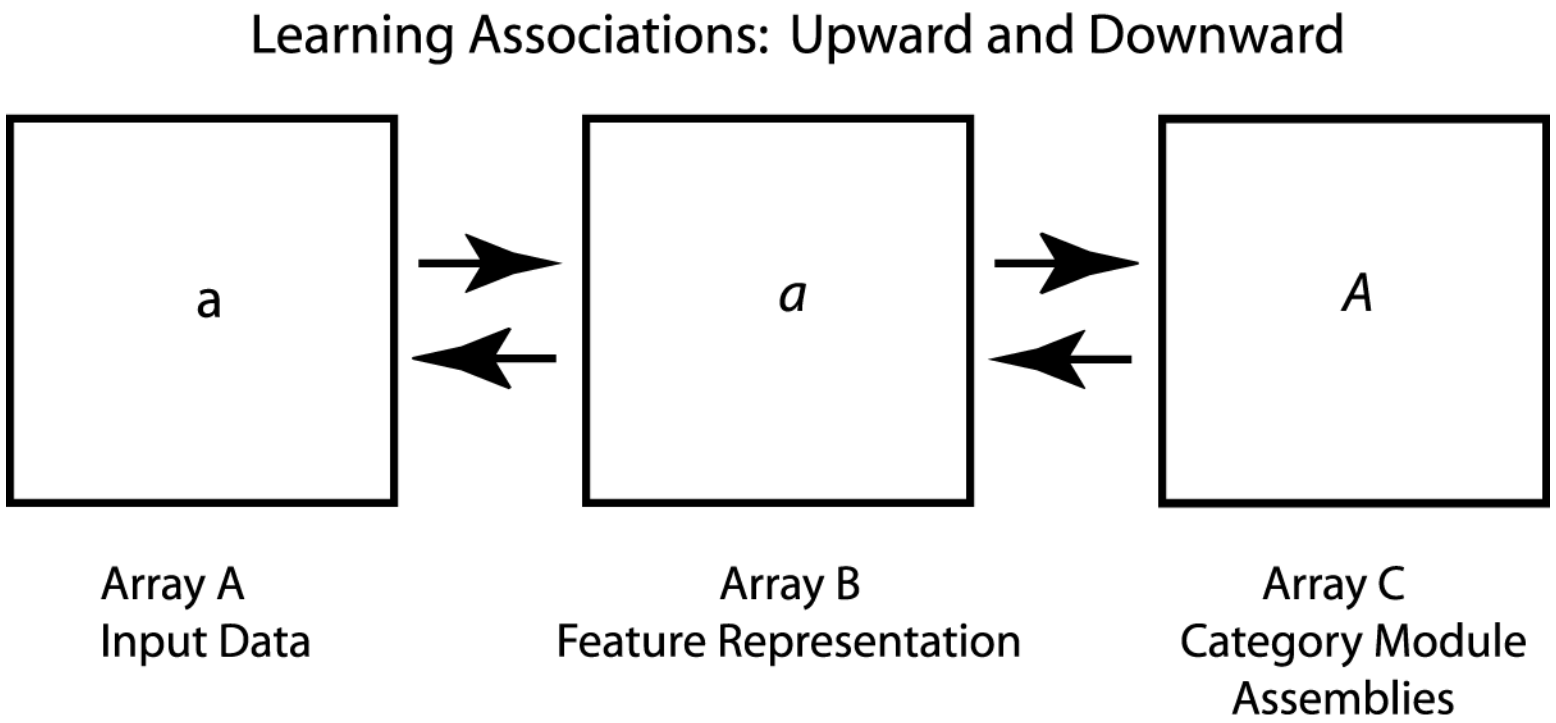
Notoriously hard for computers.

Sternberg Program 1: Letters

We assume 26 letters represent the categories.



Sternberg Program 2: Associations



Sternberg Program 2: Associations

When the categories are forming, we also learn the **reciprocal associations**:

ay	**<=**	***a***	**<=**	***A***
bee	**<=**	***b***	**<=**	***B***
cee	**<=**	***c***	**<=**	***C***
…				
zee	**<=**	***z***	**<=**	***Z***

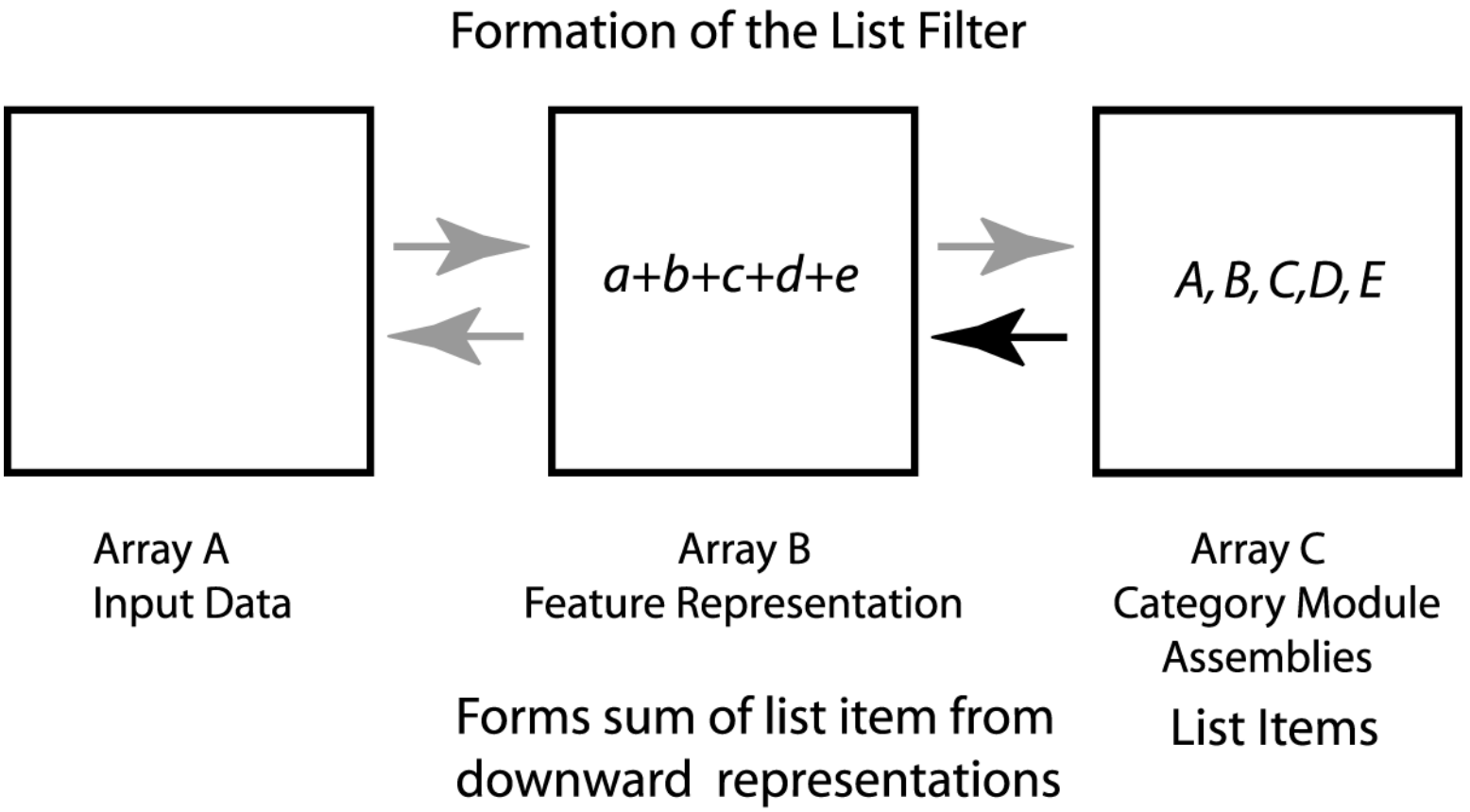
Sternberg Program 3: Formation of the Attentional List Filter

Step 1. Form list representation by sequential presentation:

Array A	**Array B**	**<=**	**Array C**
	a	***<=***	***Present list item A***
	a+b	***<=***	***Present list item B***
	a+b+c	***<=***	***Present list item C***
	a+b+c+d	***<=***	***Present list item D***
	a+b+c+d+e	***<=***	***Present list item E***

Assume activity in **Array B** sums patterns excited by downward projections from **Array C**.

Formation of the Attentional List Filter



Sternberg Program 4: List Interactions

Assume the Mixer (**Array B**) computes the **inner product between the test item** and the **list filter.**

The activity at **Array B** when input pattern **a** is presented to input **Array A** is then

Scalar Activity in **B** = ***[a,a] + [a,b] + [a,c] + [a,d] + [a,e]***

= [a,a] + noise.

This technique realizes a matched filter for pattern *a*. *The matched filter is the optimal linear filter.*

Example: If patterns ***[a,b,c … z]*** are orthogonal:

Positive activity value corresponds to **presence** of the test input in the list. **Zero activity** corresponds to **absence** from the list.

An Unintuitive Prediction

Prediction: Detecting the presence of a item on the list does **not** require **identification** of the test item.

The system **does not** respond,

"I see a C.

C is on the list.

Therefore I will make a positive response."

Instead it responds,

"The item is some item on the list.

Therefore I will make a positive response.

With additional effort I conclude the item is also a C."

Efficient Search

This curious prediction is experimentally correct.

It also **makes good engineering sense**.

To perform a task, we should only work at the necessary level of detail.

Sternberg Example: We use an **uncategorized** cluster of letter features for the list scanning match.

Low level matches are **programmed** by **task-based high level information**.

A **"just-good-enough"** strategy makes sense for complex memory based searches with slow, limited computational resources.

Arrays of (Arrays of Modules)

Applications to Cognition: Memory Filters and Disambiguation

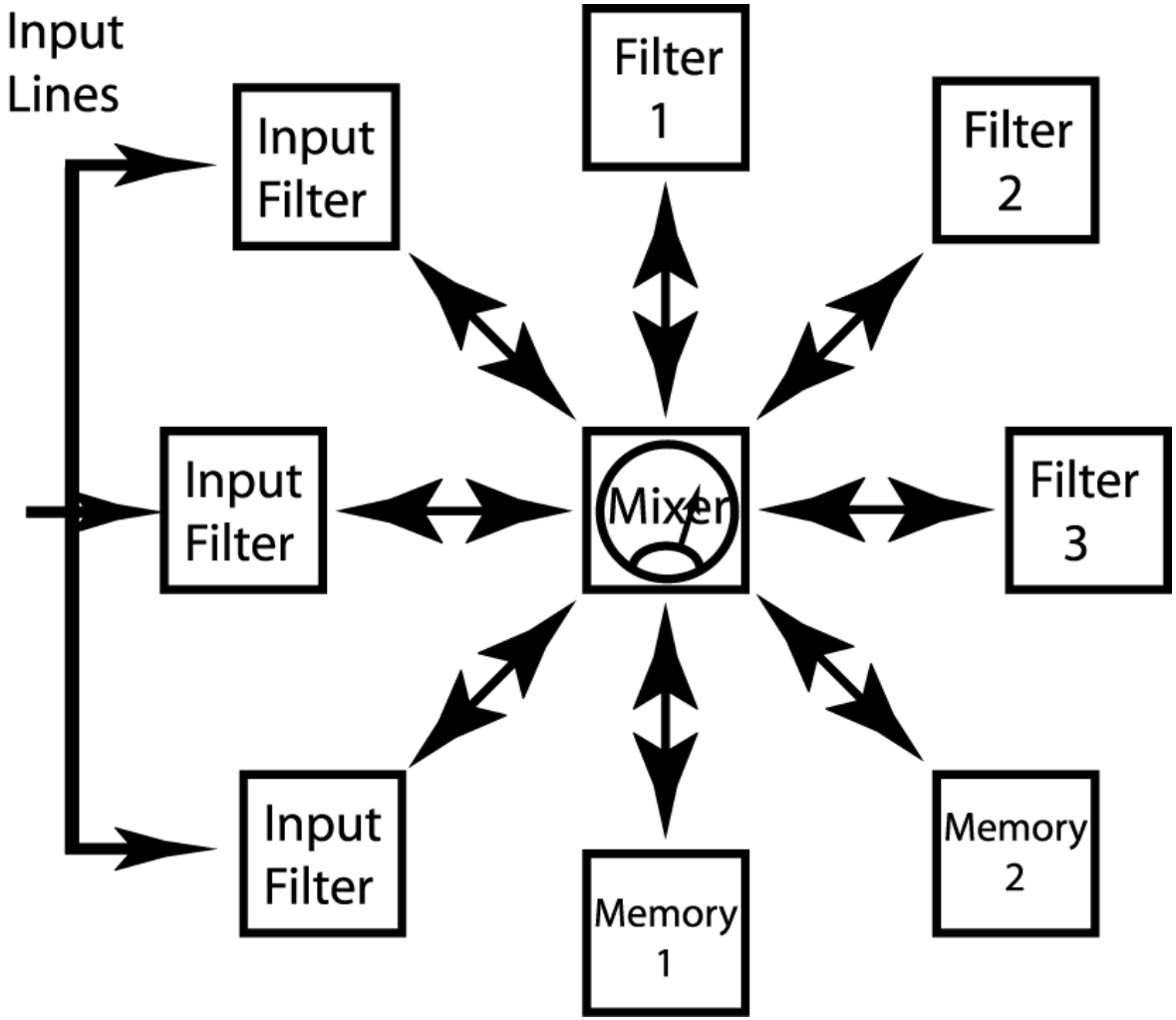
High Level Cognition

The most important applications of the Ersatz system will be to high level cognition.

For example, we can extend these ideas to work with contextual and conceptual information.

We may now have entries both into language and into cognitive signal processing with the same approach.

Extensions: Generalized Mixer



Simple Filter Disambiguation

We can make simple context neighborhood from **integrating multiple data sources**.

Integrating information easily is something we are supposed to be good at. Useful, too.

Suppose previous input words hang around and are associatively linked to their associates.

Consider a set of four sentences on a common topic:

I needed caffeine.

I walked to Starbucks.

"A vente," I said to the barrista.

"Ah, java", I exclaimed

Disambiguation

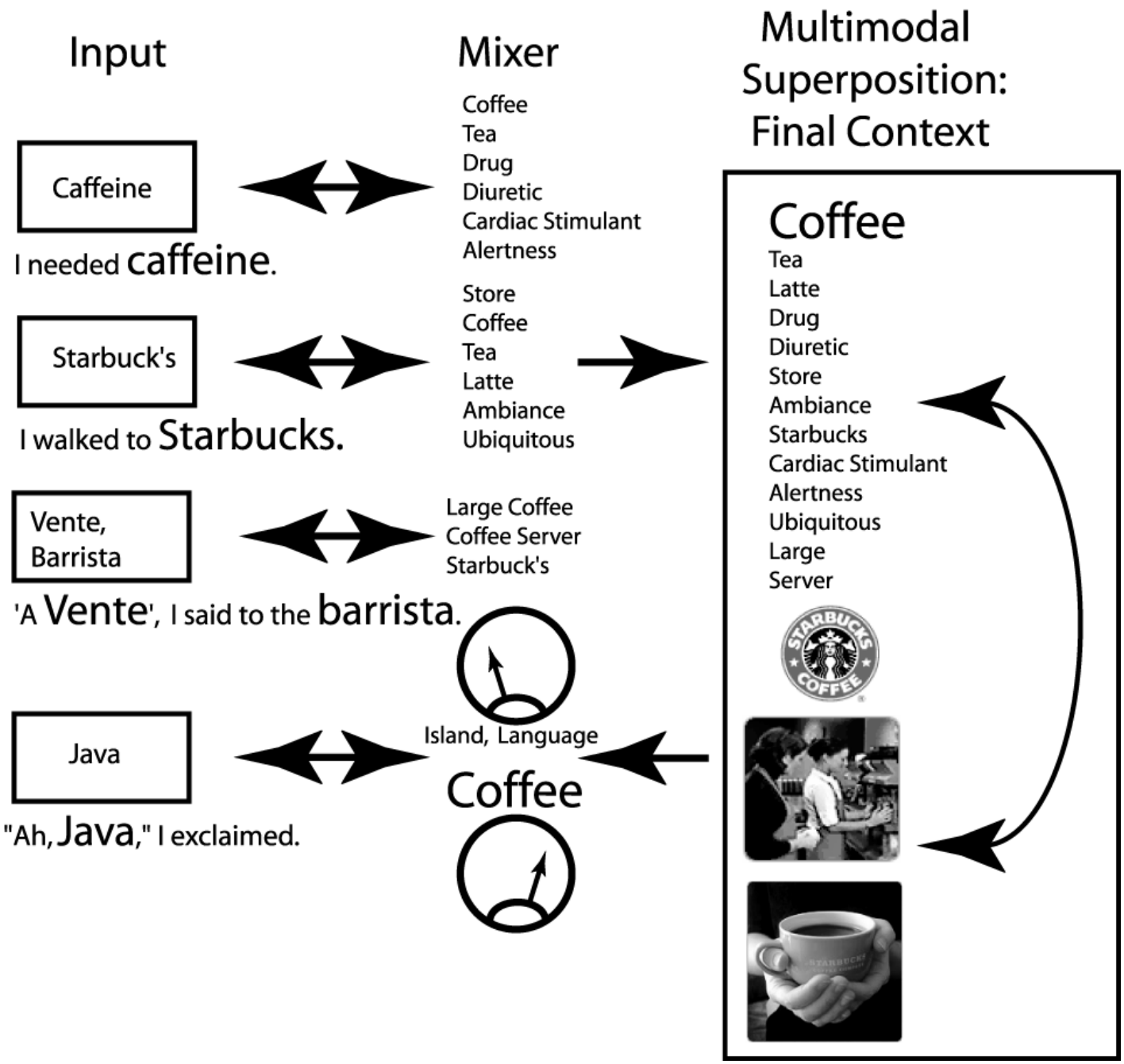
Assume that **word patterns from multiple sentences superimpose** like the Sternberg lists. (Long time-constant integration based upon context.)

What is the appropriate meaning of '**java**' in the sequence?

The string '**java**' can have three meanings:

1. A slang term for **coffee.**
2. An island in **Indonesia**.
3. A well-known **programming language**.

Superposition of the associations of the words gives rise to a context that disambiguates '**java**.'


Input
Mixer
Multimodal Superposition: Final Context
Caffeine
I needed caffeine.
Coffee
Tea
Drug
Diuretic
Cardiac Stimulant
Alertness
Starbuck's
I walked to Starbucks.
Store
Coffee
Tea
Latte
Ambiance
Ubiquitous
Vente, Barrista
'A Vente', I said to the barrista.
Large Coffee
Coffee Server
Starbuck's
Java
"Ah, Java," I exclaimed.
Island, Language
Coffee
Coffee
Tea
Latte
Drug
Diuretic
Store
Ambiance
Starbucks
Cardiac Stimulant
Alertness
Ubiquitous
Large
Server
STARBUCKS COFFEE

Arrays of
(Arrays of Modules)

Applications to Cognition:
Novelty and Anomaly

Signal Processing Problem

Detect very rare but important event in a noisy environment

Example from military acoustics:

Detect torpedo noises. Very rare, but you really want to know when they happen.

Noisy environment: shrimp, fish, whales, boats

Good filters already exist for the torpedo sounds.

But in practice, the false alarm rate is very high

Humans can do better but have other duties on shipboard.

Module Dynamics: Deflation Technique

Module behavior (for us) is characterized by eigenvectors, eigenvalues of an internal connection matrix **A**.

Recurrent feedback enhances amplitude of dominant eigenvector. (Power method.)

Suppose e_i is dominant module pattern with eigenvalue λ_i. of internal connection matrix **A.**

Get estimates of e_i, λ_i.

Form new matrix,

$A' = A - \lambda\, e_i\, e_i^{\,T}$ (**Form is that of Hebbian Anti-learning**)

The eigenvalue spectrum has now changed to decrease the previously dominant eigenvector which will now have a small eigenvalue.

Retrieved attractor will be different.

Adapt the Environment

For rare event detection: use a classic signal processing technique with an Ersatz twist.

Assume most of the time a stable safe environment.

Use deflation at the module level to adapt multi-module patterns corresponding to environment.

Deflation is Hebbian anti-learning. (one form of **adaptation**)

Such adaptation can be invoked at array **all levels** from low level sensory processing to high level cognitive analysis.

Result is multi-stage selective nulling of the stable environment based on memory and experience.

Highly Selective Multi-level Novelty Filter

Nulling:

Once the (safe) long term environment is **nulled out**, any significant new activity will be “novel”.

This novel activity captures **attention** and can be analyzed more carefully with standard (or Ersatz) techniques.

Produces a highly selective null.

Ersatz nulling can/will occur at all “cognitive” levels, both high and low.

Explicit Simple Control

Use same idea for **simple system control:**

Control the system response sensitivity (up or down) to a particular pattern or pattern class.

- **Hebbian adaptation: nulling (deflation)**
- **Hebbian enhancement (learning)**

Use **backward projections** to control lower level sensitivity excited by specific high level nodes. ("categories to watch for.")

Or use **forward projections** to control higher level sensitivities to respond to particular low level features ("features to watch for")

Analog/discrete processing across levels of organization is typical of Ersatz computation.

Overview of Ersatz Approach: Cognitive Analysis

Assemblies of module assemblies

- Multiple **programmable attentional filters**.
- **Teachable attentional filters**.
- **Flexible control** at several time scales.
- Use of long term learning, intermediate term and short term adaptation.
- Interactions between data from **memory** and from **perception**.
- **Cognitive analysis can be based on running multi-step cognitive "software."**

Cognitive Programming: High Level Teaching

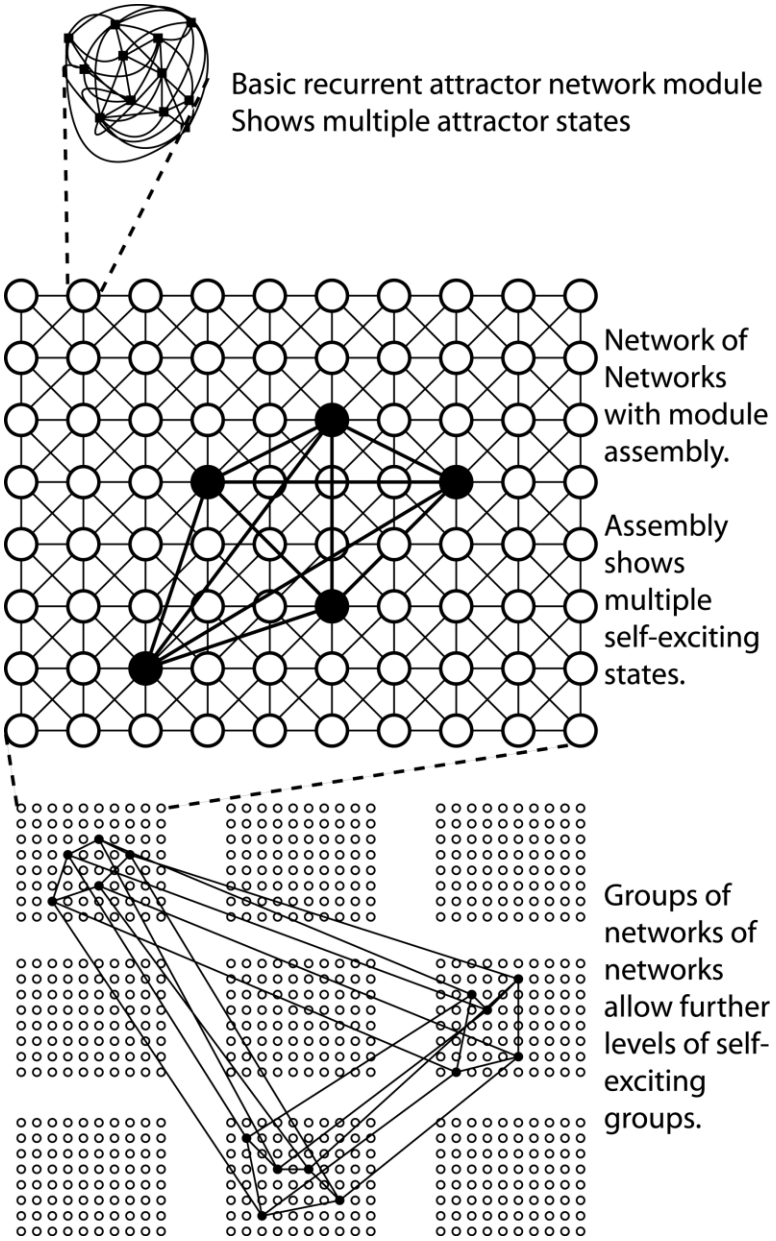
We can **explicitly teach** the system at a high level to perform tasks by being told and then learning the operation sequence used for the task.

Discussed simple module response enhancement using Hebbian adaptation, learning.

Programs **can be a memorized series of steps controlling dynamical system parameters and attentional filters.**

Unfortunately, **crisp logical program definitions may not be possible in general for Ersatz systems.**

But the **robust analog properties** of the Ersatz system may be **tolerant of logical imprecision.**

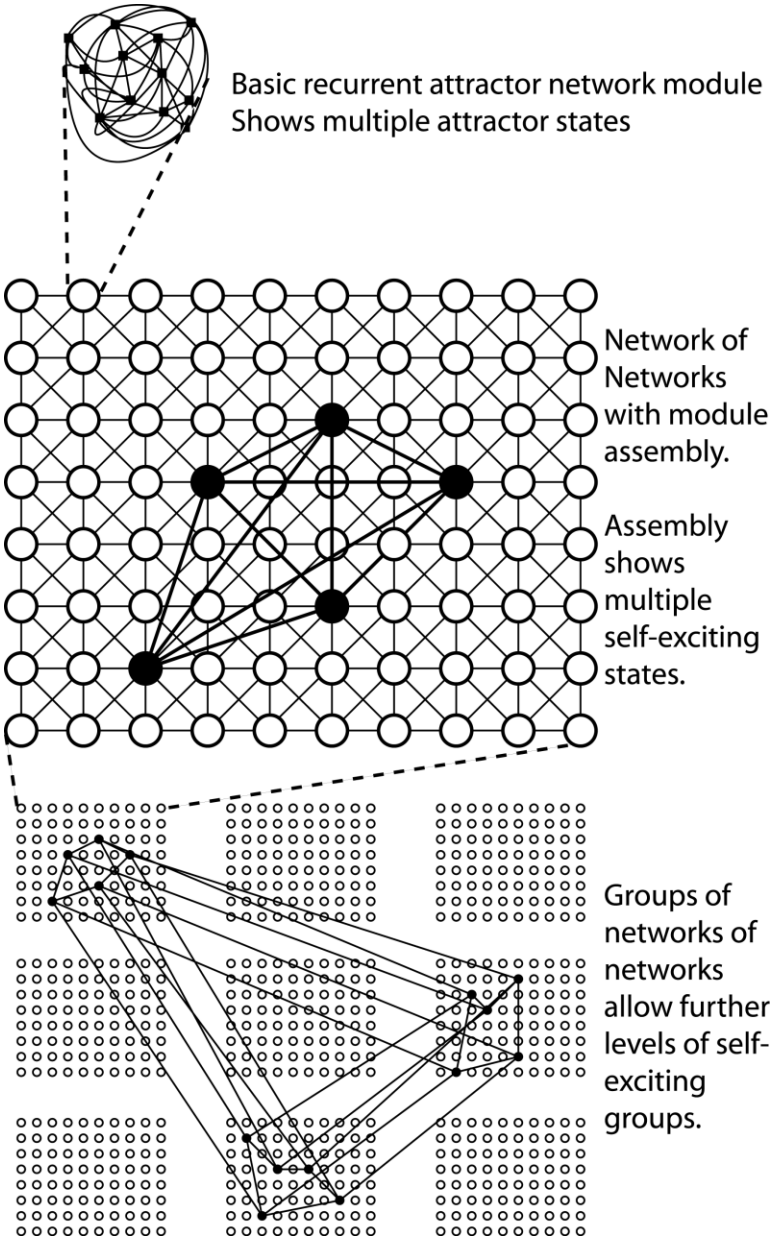


Summary (1)

The Ersatz Brain is constructed from many **small attractor networks** (modules, based on cortical columns) sparsely interconnected into a **Networks of Networks.**

It is capable of performing **specific cognitive tasks**.

It has both a **discrete** and a **continuous** set of control structures capable of **performing these specific tasks through explicit, teachable cognitive programs.**

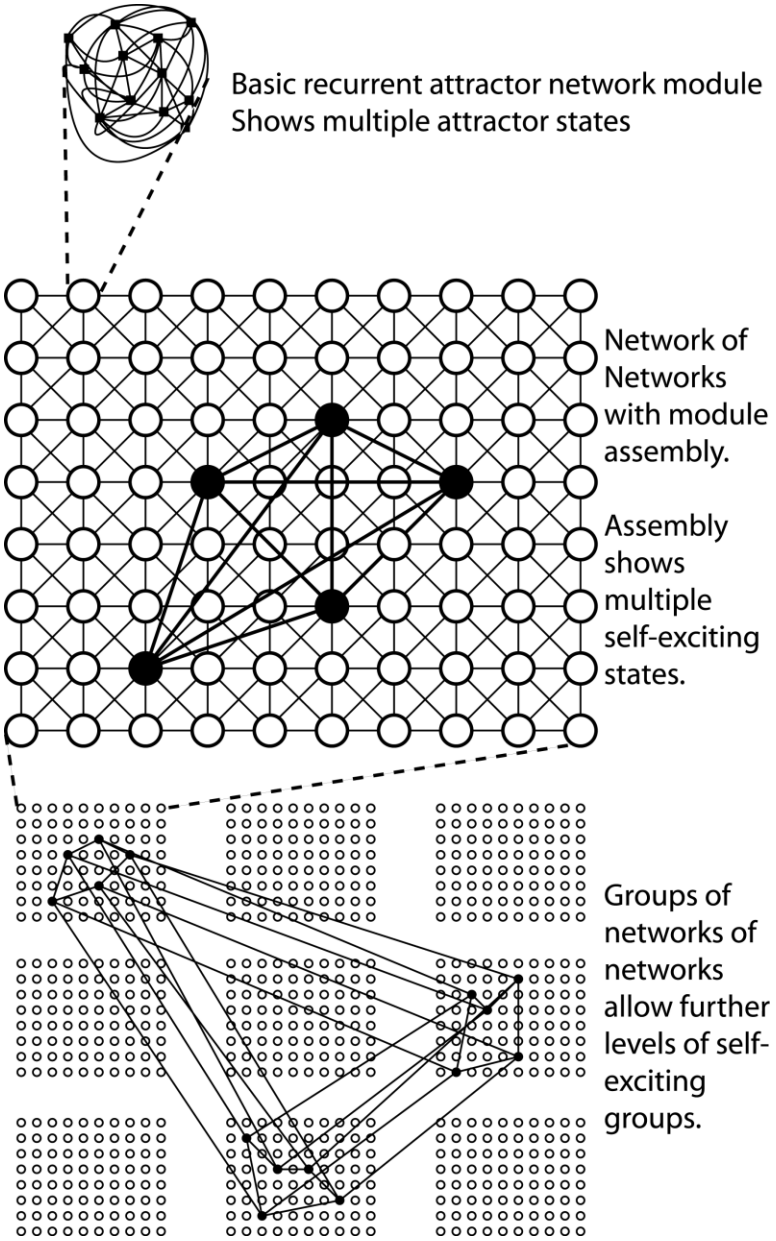


Summary (2)

Processing moves in scale from:

- **Inflexible but highly optimized, sensors**
- Through **sparse module assemblies** formed by learning in a single array of modules.
- To **flexible programs, involving learning and large dynamical systems** working at **multiple spatial scales**.

The resulting hierarchical computation is **fast, parallel, and highly integrative.**



Summary (3)

Discussed examples of problems and computational techniques well suited to a brain-like computer:

In A Single Array:

- Identity and Symmetry
- Vowel Formant Invariant Representation

In Multiple Arrays:

Memory Filter Applications:

- List Scanning
- Disambiguation
- Anomaly Detection