CN550: Neural and Computational Models of Recognition, Memory, and Attention

Overview, History, Philosophy

Lecture 1

Instructors: Anatoli Gorchetchnikov <anatoli@bu.edu>
Heather Ames <starfly@cns.bu.edu>
Teaching fellow: Karthik Srinivasan <skarthik@bu.edu>

Course Goals

To analyze various neural and statistical pattern recognition models, and their historical development and applications

To develop mathematical techniques and definitions to support fluent access to the neural network and pattern recognition literature

To relate modeling to experimental data from cognitive psychology, neuropsychology, and neurophysiology of normal and abnormal individuals

Course work emphasizes skill development, including writing, mathematics, computational analysis, teamwork, and oral communication

Grading Policy

Grades are determined by performance on:

- Blog writing: 10%
  - 10 blog entries based on lecture readings or other relevant readings
- Computational Workshop Participation: 10%
- Project Pre-proposals: 5%
  - Students will choose two benchmarks and present them to the class
  - Students will make the benchmark dataset and information available on class wiki
  - Algorithm sign-up: each student chooses two algorithms for implementation (from two different families)

Grading Policy

Grades are determined by performance on:

- Midterm: 20%
- Final: 20%
- Class Project Presentations: 15%
  - Day 1: individual presentation of own algorithms
  - Day 2: group presentations of benchmark comparison for different algorithms
- Class Project write-up: 20%
  - Form of scientific paper
  - Use of own results in Results section and use other students’ results for comparison with other results in Discussion section

Readings

Required: Duda, Richard O., Hart, Peter E., & Stork, David (2001) *Pattern Classification*. Second Edition. New York: Wiley.

Recommended:

- Schacter, Daniel L. (1996) *Searching for Memory: The Brain, the Mind, and the Past.* New York: Basic Books.
- Kandel, E., Schwartz, J.H., and Jessell, T.M. (2000). *Principles of Neural Science*, 4th Edition. New York: McGraw-Hill.
- Levine, D.S. (2000). *Introduction to Neural and Cognitive Modeling*, 2nd Edition. Hillsdale, NJ: Erlbaum.
- Strunk, William, Jr., & White E.B. (1959-2000) *The Elements of Style*, Fourth Edition. Needham Heights, MA: Allyn & Bacon.

Class Project

Comparative studies of supervised learning systems of your choice

Class teams (3-4 people each) will analyze a set of different algorithms (2-3 per person depending on complexity), with common benchmark problems and system evaluation criteria

At the end of the semester you will integrate findings, draw conclusions, present results in class and write a final essay summarizing your contributions and experiences

Class Project: Group Tasks

Split into groups

Individuals research the existing benchmarks and propose them to the group

The group selects two of these benchmarks and presents them to class

After discussion one of these benchmarks per group will be selected

Both groups will use both selected benchmarks, but each group will be responsible for presenting cumulative results for their benchmark at the end

Groups coordinate selection of algorithms for each member to implement during the project

Class Project: Individual Tasks

Propose benchmarks

Select the algorithms you want to implement and coordinate them with the group

Implement the algorithms and collect data

Share the data with your group and with the other group

Prepare individual presentation of your data and participate in group presentation of the benchmark data

Prepare the final write-up of your results in the form of scientific paper

What Is Learning?

Webster's dictionary defines learning as "gaining knowledge, understanding, or skill by study or experience"

How can we determine the quality of learning?

No direct access, so performance change is the only measure

Learning is a process underlying the change in performance in the certain task/environment based on experience with this task/environment

Machine learning literature suggests that the only way to estimate the quality of learning is to compare the performance on a certain test before and after learning

In academic setting the initial test is often omitted

What Is Memory?

The ability to store experiences for later recall and use

Two components:

- Storage, Encoding
- Recall, Retrieval

Each of the two can be impaired, careful experiments are designed to distinguish these impairments

What is the relationship between memory and learning?

- Quality of learning evaluates cumulative effect of both encoding and retrieval
- Previous experiences (memories) can affect the speed of learning of new experiences

What Is Memory?

In general we would like to say that learning only affects encoding, but since retrieval affects performance…

Retrieval shall be efficient enough to allow quick access to all stored memory episodes and efficient search through them

For that reason biology seem to use content-addressable memory: memory that is accessed by providing it with a part of the contents you are looking for

An example of content-addressable memory is an autoassociative memory that recreates a complete memorized pattern from its fragment

What Is Memory?

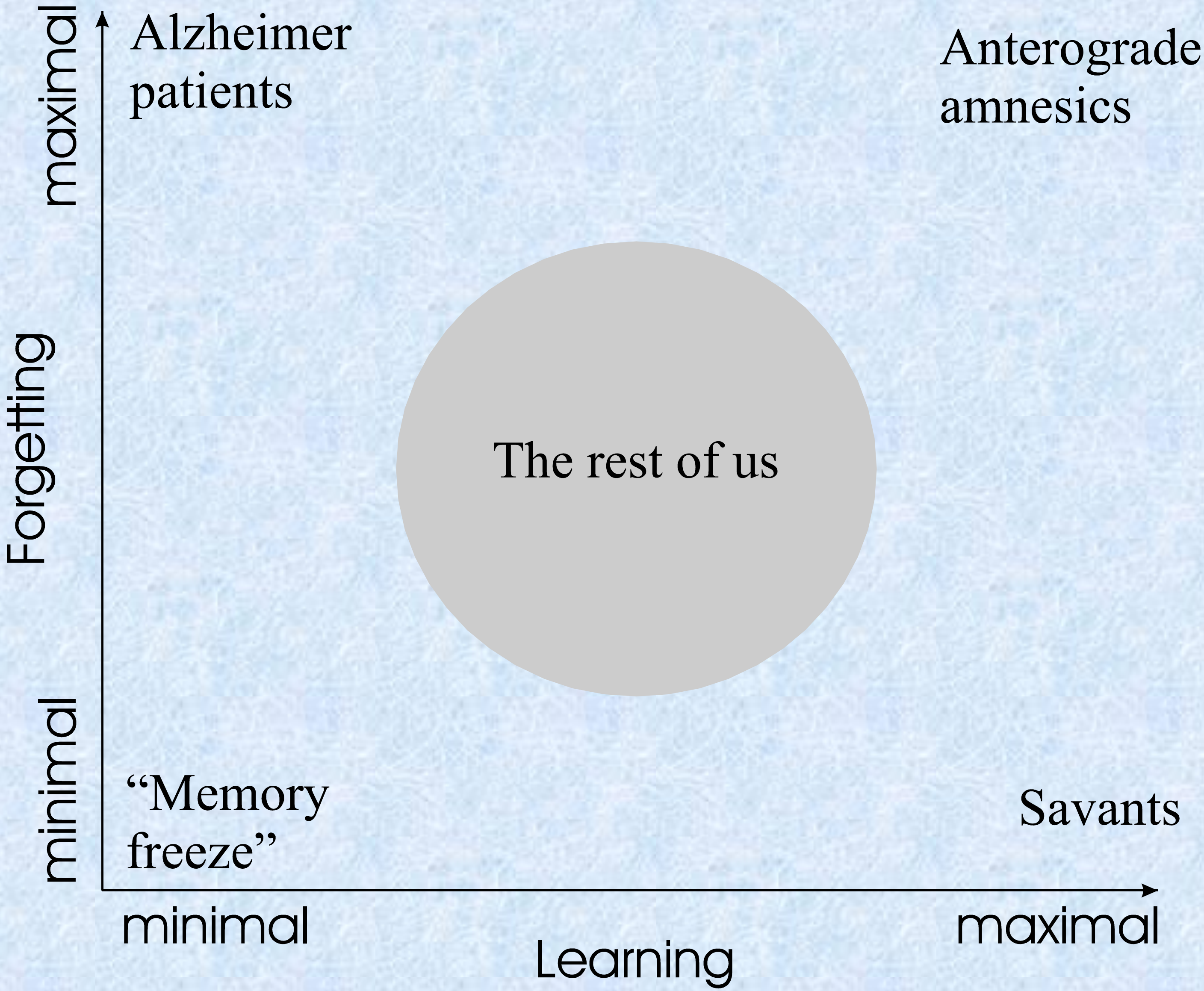
Another important property of memory is a rate of forgetting

Two reasons to increase this rate:

- Quite often excessive details are irrelevant or even harmful for a good performance
- Memory capacity is likely limited

Reasons to decrease this rate are obvious

Learning and Memory



Alzheimer patients

Anterograde amnesics

The rest of us

"Memory freeze"

Savants

Forgetting: minimal – maximal

Learning: minimal – maximal

Memory Extremes

Daugman, John G. (1990) Brain metaphor and brain theory. In Eric Schwartz (Ed.) *Computational Neuroscience.* Cambridge, Mass.: MIT Press. Chapter 2: pp. 9-18.

Borges, Jorge Luis (1942) Funes, the Memorious. In: *Ficciones* (translation), New York: Grove Press (1962), pp. 107-115 http://en.wikipedia.org/wiki/Borges

Henig, Robin Marantz (2004) The quest to forget. *The New York Times Magazine*, April 4, 2004, pp. 32-37.

Treffert, Darold A., and Christensen, Daniel D. (2005) Inside the mind of a savant. *Scientific American*, Dec., pp. 108-113.

Audio & video

- http://www.npr.org/templates/story/story.php?storyId=5352811 Unique memory lets woman replay life like a movie
- Clive Wearing: Living without memory YouTube (BBC – The Mind)

Does Perfect Memory Mean Good Learning?

It appears that the best memory system for learning should be capable of

- Fast memorizing of individual facts or events
- Generalizing the facts and events that are similar
- Forgetting the irrelevant specific details of these generalized experiences

In this case we keep the important information and disregard the unimportant pieces

Furthermore, if new event falls within previously generalized framework, we can perform right away and consider details later – improvement of performance

Classification

Our learning system should be able to classify the situations that it encounters and respond accordingly to this classification

First step is to classify the situation, object, or task

Incorrect classification will lead to poor performance no matter the skills

Classes can be more general or more specific, depending on the task at hand a different level of specificity is needed

Specificity of a class determines how much additional information about its members we know without investigating individual members

Classification

Sometimes classes are called categories and grouping items into classes is called clustering

How can we define a certain class or category?

Robert Sheckley. (1971) The Cruel Equations. In: *Can You Feel Anything When I Do This?* ISBN-10: 0-385-03495-4 ISBN-13: 978-0-385-03495-1, Doubleday & Company, Inc., Garden City, New York

Sentient being that knows the password – allow into the camp

Sentient being that does not know the password – keep away from the camp

Class Definitions and Pattern Recognition

Common set of properties determines the class

Giving a prototypical object can also define a class

For each new instance we need to compare properties with a prototype or with a set of class defining properties using some metrics

Pattern recognition is a part of the classification process that applies these metrics to determine the class of each pattern

Properties

Properties can be

- Binary (features)
- Continuous (dimensions)
- Enumerable

Some properties have more value than others, this can also be included in the metrics

The remainder of the course is concerned with establishing these metrics in different systems as well as with formalizing the sets of properties

Neural Network Architectures

Adaptive filters
  - McCulloch-Pitts neuron (1943)
  - Pitts & McCulloch (1947)

Perceptrons
  - Original perceptron
  - Adaline / Madaline & LMS
  - Back propagation

Hebbian learning

Learning Matrix/Crossbar

Linear Associative Memory

Outstar / Instar Networks

Adaptive Resonance Theory
  - ART 1, ART 2, …
  - ARTMAP, for associative memory

Cognitron / Neocognitron

Self-organizing Maps

Simulated annealing/ Energy models

McCulloch-Pitts Neuron (1943)

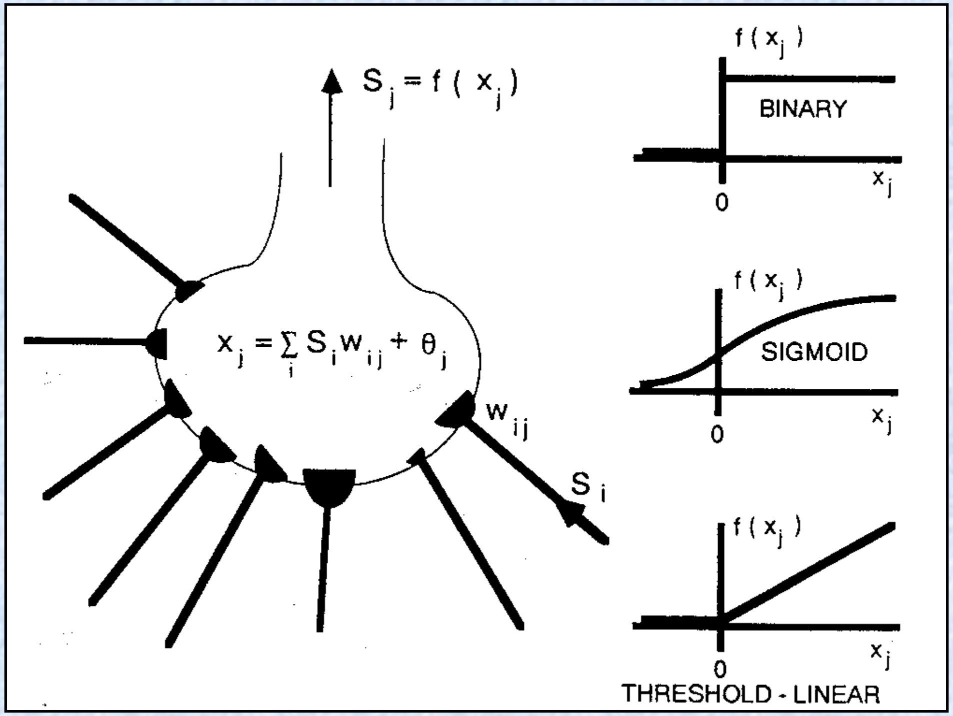
“A logical calculus of the ideas immanent in nervous activity”

Networks can be configured to perform arbitrary logical functions

Proposed as a computer architecture

Lost to von Neumann computer architecture

Aside: DARPA SyNAPSE program seeking to replace von Neumann



$$\sum_i S_i w_{ij} = \vec{S} \cdot \vec{w} =$$

$$= \underset{\text{Energy}}{\underset{\uparrow}{\left\|\vec{S}\right\| \cdot \left\|\vec{w}_j\right\|}} \underset{\text{Pattern}}{\underset{\uparrow}{\cos\left(\vec{S}, \vec{w}_j\right)}}$$

W. Pitts & W. McCulloch (1947)

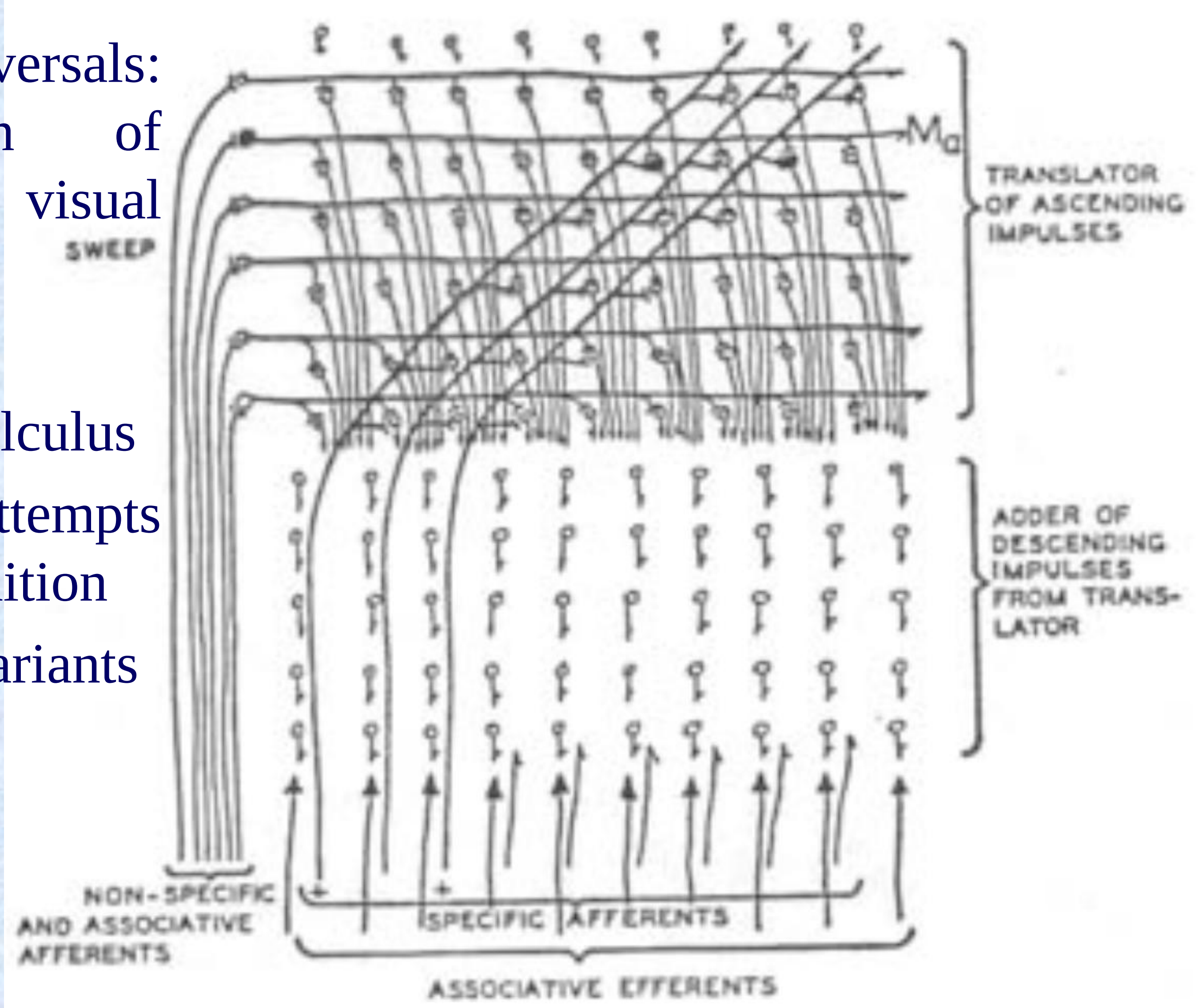
“How we know universals: the perception of auditory and visual forms”

Similar to logical calculus

One of the first attempts on pattern recognition

Computation of invariants

Still no learning


SWEEP
Ma
TRANSLATOR OF ASCENDING IMPULSES
ADDER OF DESCENDING IMPULSES FROM TRANSLATOR
NON-SPECIFIC AND ASSOCIATIVE AFFERENTS
+
+
SPECIFIC AFFERENTS
ASSOCIATIVE EFFERENTS

Frank Rosenblatt – Perceptron (1957)

A large class of neural models

Also, Eduardo Caianiello (1961)

McCulloch-Pitts based binary classifier plus learning

Supervised, error-based

Proven to converge for a linearly separable set of data

Weight update rule:

$$w_i(t+1) = w_i(t) + \eta\left(d_j - y(t)\right)x_{ij}$$

Or in continuous form

$$\frac{dw_i}{dt} = \eta\left(d(t) - y(t)\right)x_i(t)$$

If we want to cast it back in CN510 terms it is anti-Hebb with presynaptically gated teaching signal

Sometimes is called delta rule:

$$\frac{dw_i}{dt} = \eta\delta(t)x_i(t)$$

Adaline / Madaline (B. Widrow, 1960)

Adaptive Linear Element

Differs from perceptron as postsynaptic signal used in the delta rule before applying signal function to it (in perceptron it is used after signal function is applied)

Adaline had one postsynaptic neuron, Madaline had many

Widrow-Hoff delta rule:

$$\frac{dw_i}{dt} = \eta\delta\left(t\right)\frac{x_i\left(t\right)}{\|x\|}$$

Leads to the minimization of the least mean square error averaged over all inputs

Used in adaptive equalizers, modems, antennae, echo cancellation …

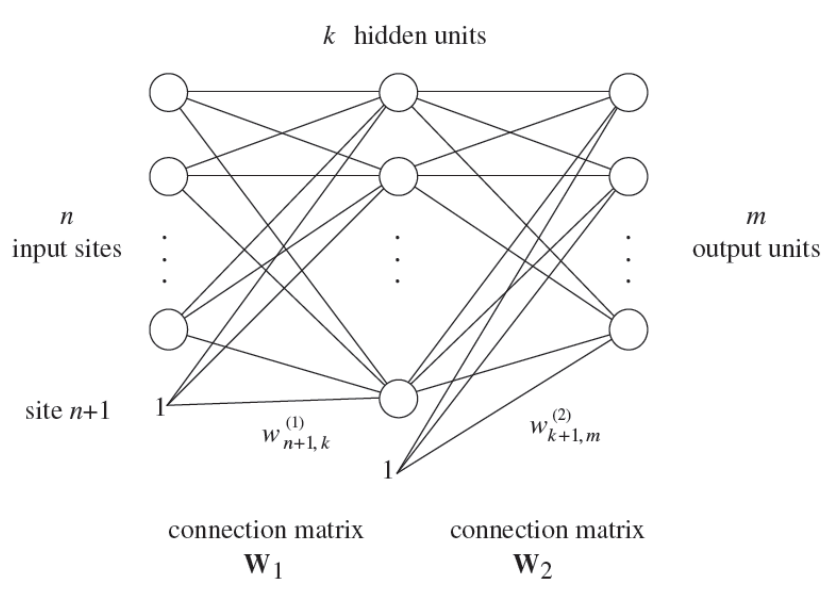
Frank Rosenblatt – Multi-Layer Perceptron (1962)

Principles of Neurodynamics:

- Section 13.3 – Back-propagating error correction procedures,
- Section 13.4 Summary of the new architecture

Similar architectures proposed independently by

- Arthur Earl Bryson, Yu-Chi Ho (1969)
- Paul Werbos (1974)
- D. Parker (1982)



Algorithm requires differentiable signal function

Convergence is very slow and not guaranteed

Can converge to local minimum

Hebbian Learning

"*When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes place in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.*"

Donald Hebb (1949)

Usually is represented as weight dependency on correlation between input and output

$$\dot{w}_{ij} = \eta x_i y_j$$

Variation of Hebb's rule with different gating of decay are considered separately

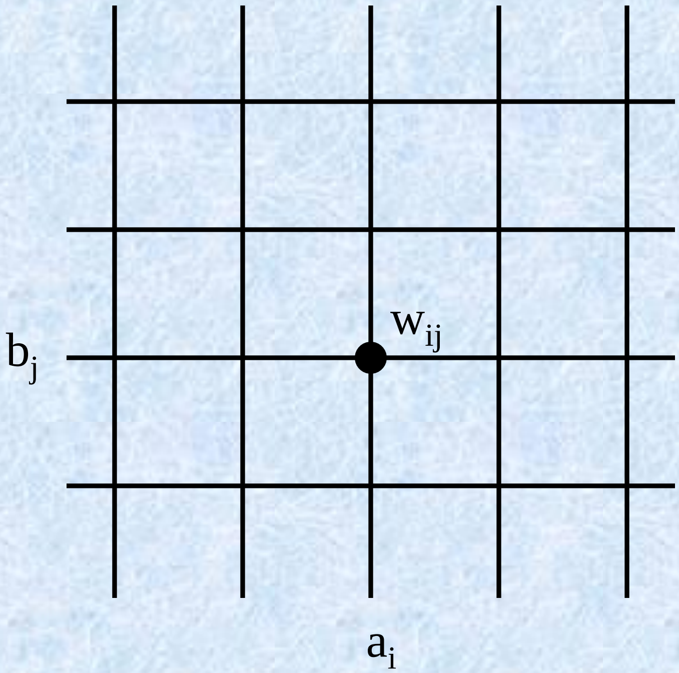
Learning Matrix (Crossbar)

Karl Steinbuch (1961)

Learning: set correct category *b*

$$\Delta w_{ij} = \begin{cases} 0 & if \ b_j = 0 \\ \eta & if \ b_j = 1 \, and \ a_i = 1 \\ -\eta & if \ b_j = 1 \, and \ a_i = 0 \end{cases}$$

Performance: compare *a* to all vectors *w*, the best match gives the category



$$b_j = \begin{cases} 1 & if \ \|\vec{a}\| \cdot \|\vec{w}_j\| \cos\left(\vec{a}, \vec{w}_j\right) = \max\left(\vec{a} \cdot \vec{w}\right) \\ 0 & otherwise \end{cases}$$

Learning Matrix (Crossbar)

Learning is technically Hebbian:

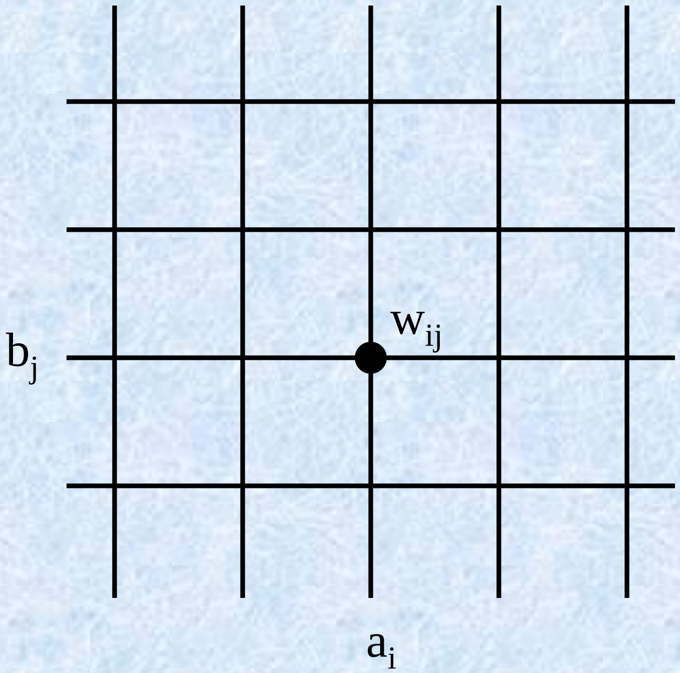
$$\Delta w_{ij} = \begin{cases} 0 & if\ b_j = 0 \\ \eta & if\ b_j = 1\, and\ a_i = 1 \\ -\eta & if\ b_j = 1\, and\ a_i = 0 \end{cases}$$

is equivalent to

$$\dot{w}_{ij} = \eta x_i y_j$$

if $x_i = 2a_i - 1;\ y_j = b_j$

Performance: breaks without complement coding



Aside: might be revived based on memristive nanodevices that are currently developed

Complement Coding

For binary features:

Take an original vector of features (e.g. 10011001)

Build a complement vector

01100110

Combine both vectors into a new vector and use it as input

For continuous attributes:

Similar, but use a_{imax}-a_i where a_{imax} is the maximal possible value of an attribute i

For example for normalized values of a_i vector (0.3, 0.5, 0.8) turns into (0.3, 0.5, 0.8, 0.7, 0.5, 0.2)

The “physical” meaning of the complement coding is that it allows to code not only a measure of property presence but also a measure of its absence

Linear Associative Memory (LAM)

J. Anderson (1969)

T. Kohonen (1970)

K. Nakano (1972)

Many variations exist

S.-I. Amari (1972) – stability analysis

OLAM (Optimal LAM, Kohonen & Ruohonen, 1973) uses Moore-Penrose pseudoinverse to compute weights

Basically same correlation weight matrix with a twist: input vectors are mutually orthogonal

Linear Associative Memory (LAM)

Learning: $w_{ij} = \sum_{p} x_i^{(p)} y_j^{(p)}$

Recall: $y_j = \vec{x}^{(t)} \vec{w}_j = \sum_i x_i^{(t)} w_{ij} = \sum_i x_i^{(t)} \sum_{p} x_i^{(p)} y_j^{(p)} =$

$$= \sum_i \sum_{p} x_i^{(t)} x_i^{(p)} y_j^{(p)} = \sum_{p} \left(\vec{x}^{(t)} \cdot \vec{x}^{(p)} \right) y_j^{(p)} = \left\| \vec{x}^{(t)} \right\|^2 y_j^{(p=t)}$$

Mutual orthogonality of inputs drastically reduces the capacity of the network: with 8 inputs there are 256 possible input vectors but only 8 mutually orthogonal vectors

External control of system dynamics plays a very important role in all of the above networks

Steve Grossberg (1961) – Embedding Fields

Critical difference that there is no external control: learning is not disabled during recall, same equations govern both

$$\dot{x}_i = -Ax_i + [excitation] - [inhibition]$$

$$\dot{w}_{ij} = \eta(t)\, f\left(x_i, y_j, w_{ij}\right)$$

Note that this is what Steve & Co call "real time"

Variations of activation: additive

$$\dot{x}_i = -Ax_i + \sum I_k - \sum I_j$$

shunting

$$\dot{x}_i = -Ax_i + (B - x_i)\sum I_k - (C + x_i)\sum I_j$$

Variations of Learning

Hebb with passive decay $\dot{w}_{ij} = \eta x_i y_j - \alpha w_{ij}$

Postsynaptically gated decay (Instar) $\dot{w}_{ij} = \eta x_i y_j - \alpha y_j w_{ij}$

Presynaptically gated decay (Outstar) $\dot{w}_{ij} = \eta x_i y_j - \alpha x_i w_{ij}$

Note that both instar and outstar lead to asymmetric weights

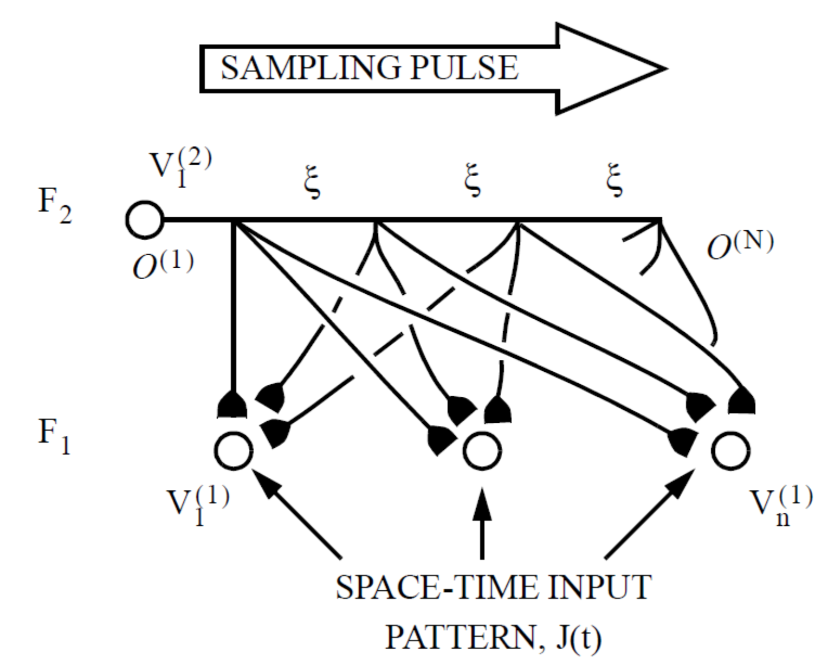
$$w_{ij} \neq w_{ji}$$

Some can claim that this is non-hebbian, but only if hebbian is nothing but correlation

Outstar theorems prove convergence (Grossberg, 1967-1972)

Avalanche (Grossberg 1969)

- Combination of multiple outstars at consecutive branching points
- Samples the spatial patterns through time, learns spatio-temporal sequence
- Modified to have multiple sequential outstars with a common "go" signal
  - Binary – go-nogo
  - Continuous – performance speed



Contrast Enhancement

Feedforward competition:

$$\frac{dx_i}{dt} = -Ax_i + \left(B - x_i\right)I_i - \left(C + x_i\right)\sum_{j \neq i} I_j$$

Gives automatic gain control

Recurrent competition

$$\frac{dx_i}{dt} = -Ax_i + \left(B - x_i\right)\left(f(x_i) + I_i \right) - \left(C + x_i\right)\sum_{j \neq i} f(x_j)$$

Allows to manipulate the shape of output with different signal functions *f()*

Contrast enhancement taken to the extreme (WTA) can serve as a category selection

Combining Instar and RCF gives a system similar to the learning matrix

Fukushima (1975) – Cognitron/Neocognitron

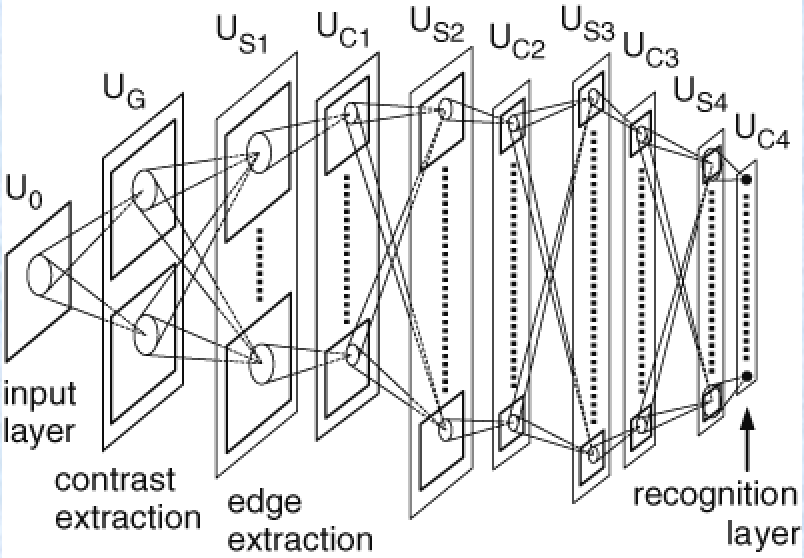
Based on Hubel & Wiesel cascading network model

Alternating populations of simple and complex cells

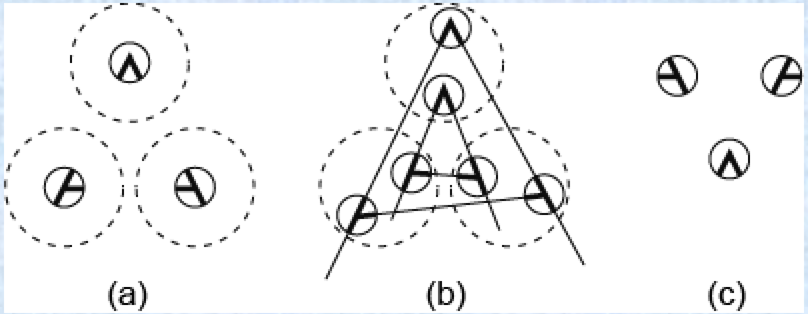
Translation, rotation, and scale invariant pattern recognition

Competitive learning on the local scale

Teaching signal on each level



Basic principle: extract features and tolerate distortions



Catastrophic Forgetting

Variations to cope with “catastrophic forgetting” – learning of new patterns erodes the previous memory

- Slow learning through multiple presentations of input patterns, interleaved learning of new patterns
- Input restrictions: orthogonal inputs have non-overlapping weight sets, so no interference
- Output restrictions: WTA categories also have non-overlapping weight sets
- Slow down or stop learning for the established weights
- Learning is conditional, it only happens when certain conditions are met (e.g. disabled during recall, requires resonance, etc.)

Adaptive Resonance Theory

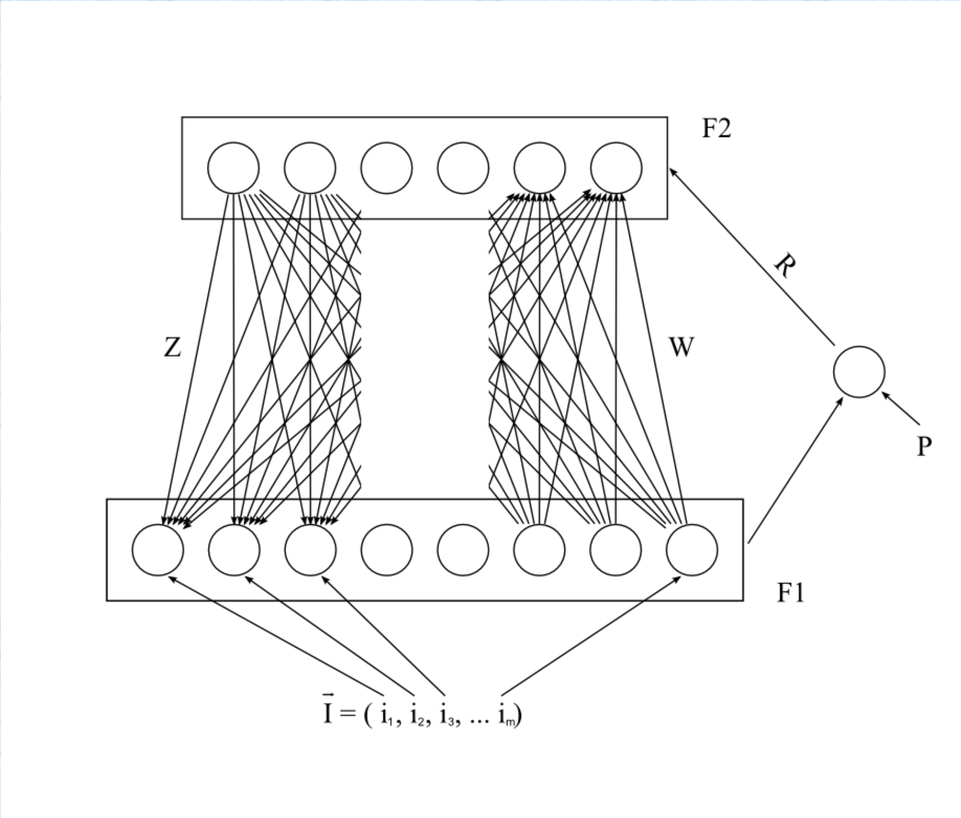
Interaction of bottom-up and top-down learning

Bottom-up weights learn which category the input pattern shall activate

Top-down weights learn which pattern each category is supposed to be activated by

Top-down weighs hold the prototype for a category

Automatic self-correction if TD and BU do not match

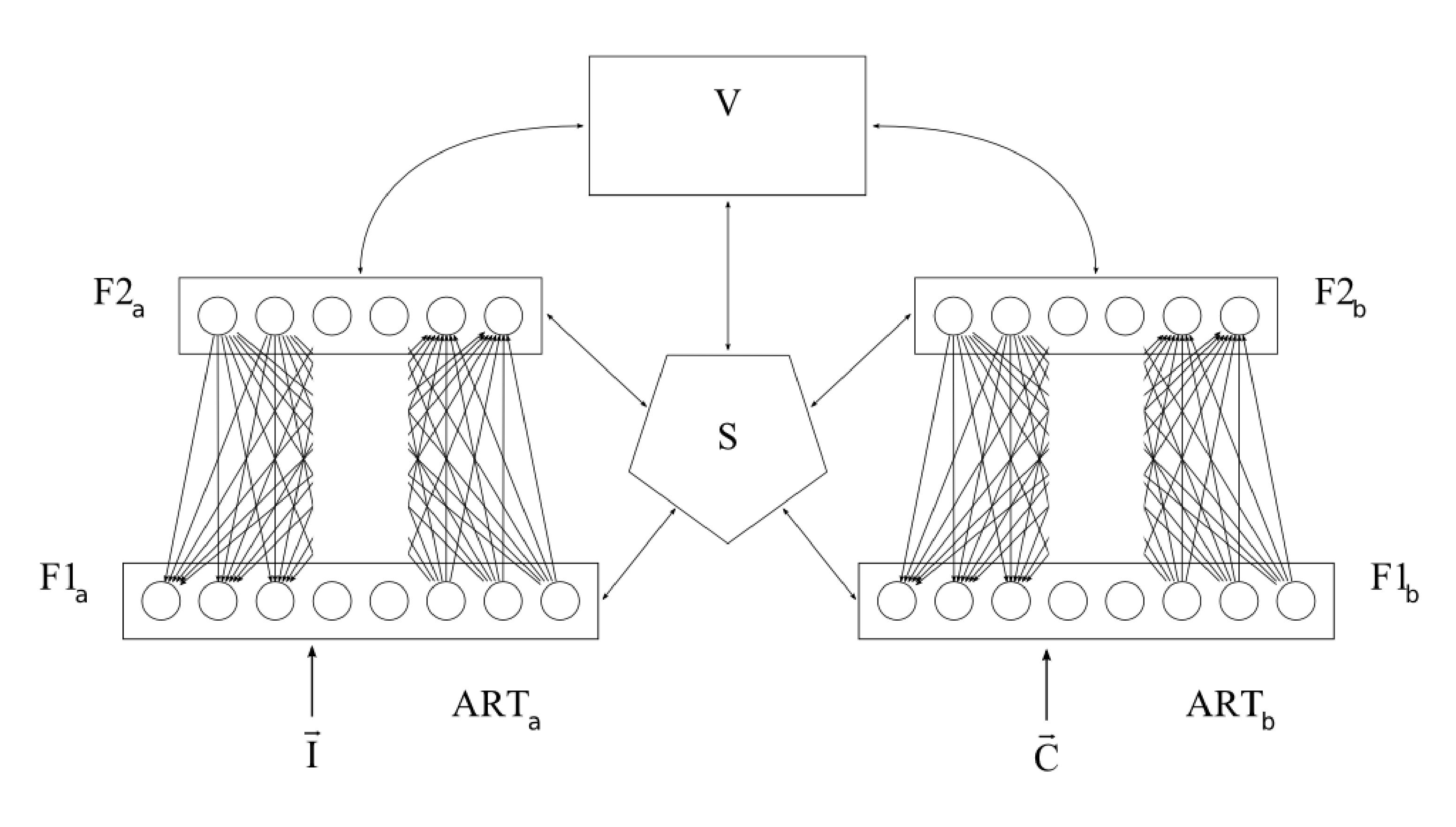


Variable degree of what match is: controlled through vigilance parameter

ARTMAP

Designed for supervised learning

Includes internal control of vigilance matching criterion



Simulated Annealing

Probabilistic weight change laws to cope with gradient descent tendency to get stuck in local minima

Represent weights as an energy function

The goal is to bring the system to a state with the minimum possible energy

High temperature – large jumps in solution space, low temperature – small jumps

Temperature is gradually reduced to 0

For any given finite problem, the probability to reach the global optimal solution approaches 1 as the annealing schedule is extended

The time required to ensure success will usually exceed the time required for a complete search of the solution space

Boltzmann Machine

Stochastic recurrent neural network by Hinton and Sejnowski (1983)

Uses simulated annealing to learn

Due to the slowness of the process is in general impractical

With constrained connectivity can still be used effectively

Easily parallelizable

Theoretically interesting

Technically, Helmholz was musing with the idea of using simulated annealing for inference long before modern neural networks

Quasi-neural and Non-neural Pattern Recognition

Deterministic:

- Support vector machines

Probabilistic:

- Expectation maximization
- Maximum likelihood estimation
- Bayesian estimation

Genetic algorithms