Final Exam

- **Thursday, December 11, 8:00am – 10:00am**
- **rooms: pending…**

- No books, no questions, work alone, everything seen in class.

Artificial Neural Networks and AI

Artificial Neural Networks provide...

- A new computing paradigm
- A technique for developing trainable classifiers, memories, dimension-reducing mappings, etc
- A tool to study brain function

Converging Frameworks

- **Artificial intelligence (AI):** build a "packet of intelligence" into a machine
- **Cognitive psychology:** explain human behavior by interacting processes (schemas) "in the head" but not localized in the brain
- **Brain Theory:** interactions of components of the brain -

  - computational neuroscience

  - neurologically constrained-models

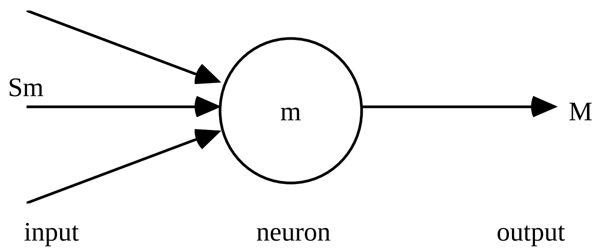
- and abstracting from them as both **Artificial intelligence** and **Cognitive psychology:**

  - connectionism: networks of trainable "quasi-neurons" to provide "parallel distributed models" little constrained by neurophysiology

  - abstract (computer program or control system) information processing models

Vision, AI and ANNs

- **1940s: beginning of Artificial Neural Networks**



McCullogh & Pitts, 1942

$\Sigma_i \; w_i x_i \; \geq \; \theta$

Perceptron learning rule (Rosenblatt, 1962)

Backpropagation

Hopfield networks (1982)

Kohonen self-organizing maps

...

Vision, AI and ANNs

1950s: beginning of computer vision

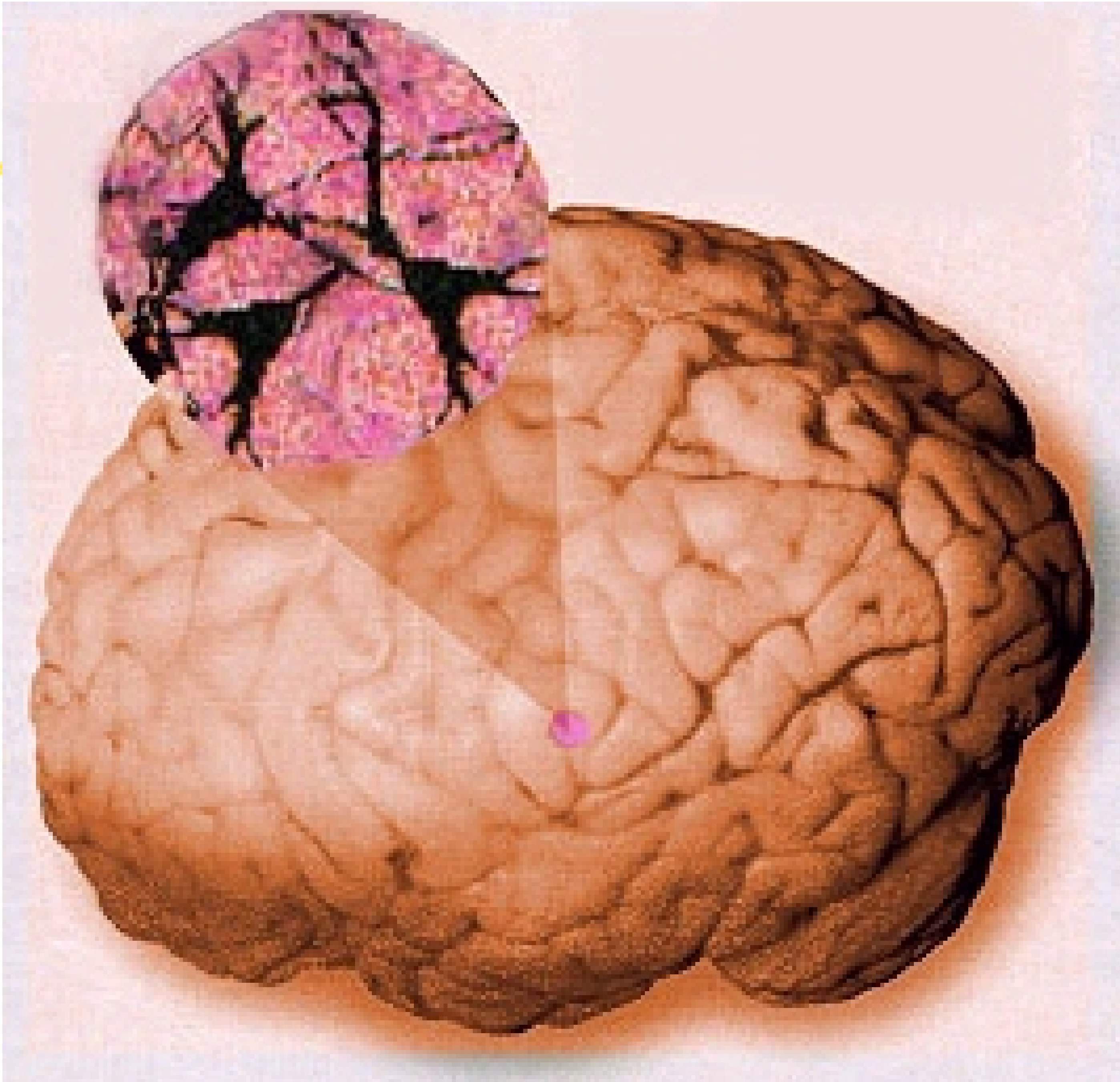
Aim: *give to machines same or better vision capability as ours*

Drive: AI, robotics applications and factory automation

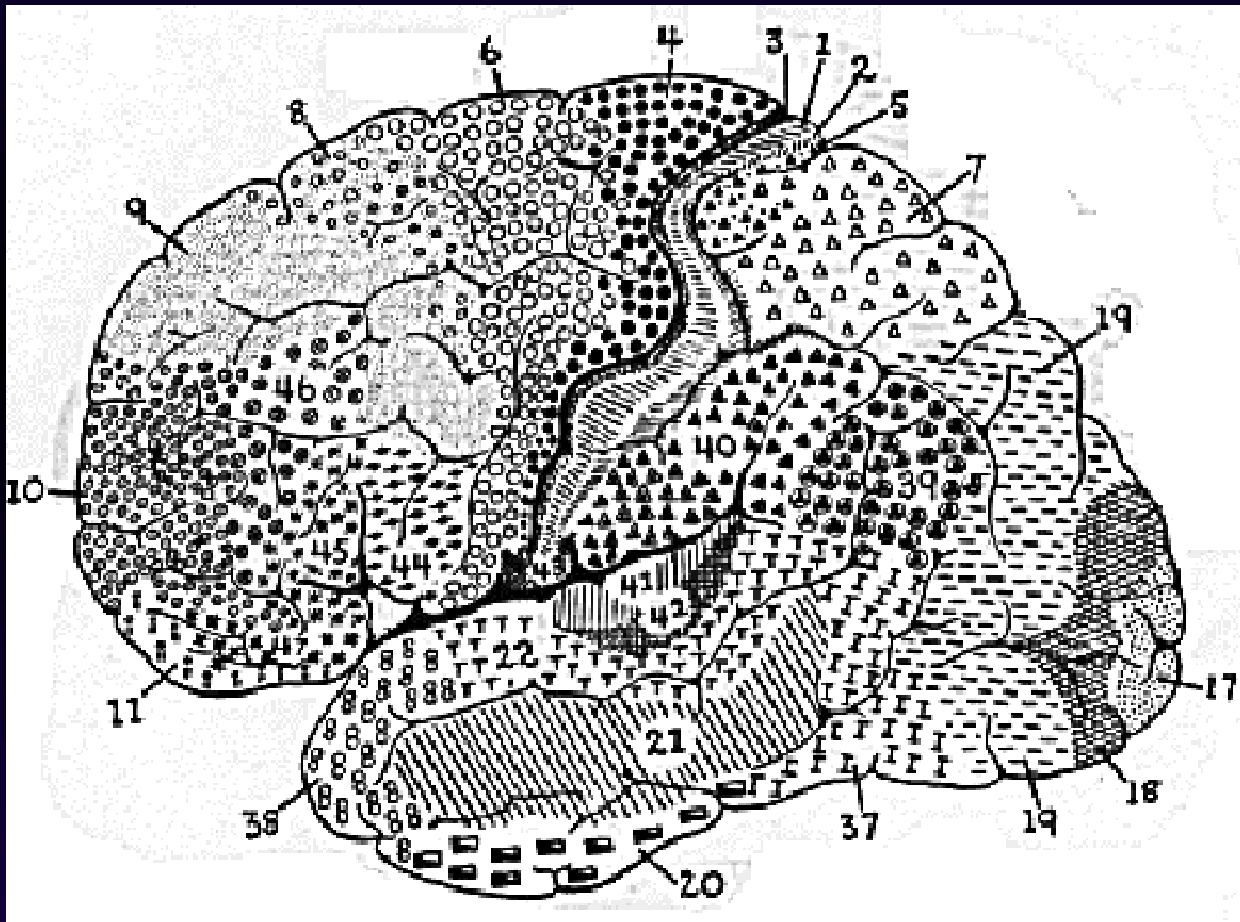
Initially: passive, feedforward, layered and hierarchical process that was just going to provide input to higher reasoning processes (from AI)

But soon: realized that could not handle real images

1980s: Active vision: make the system more robust by allowing the vision to adapt with the ongoing recognition/interpretation



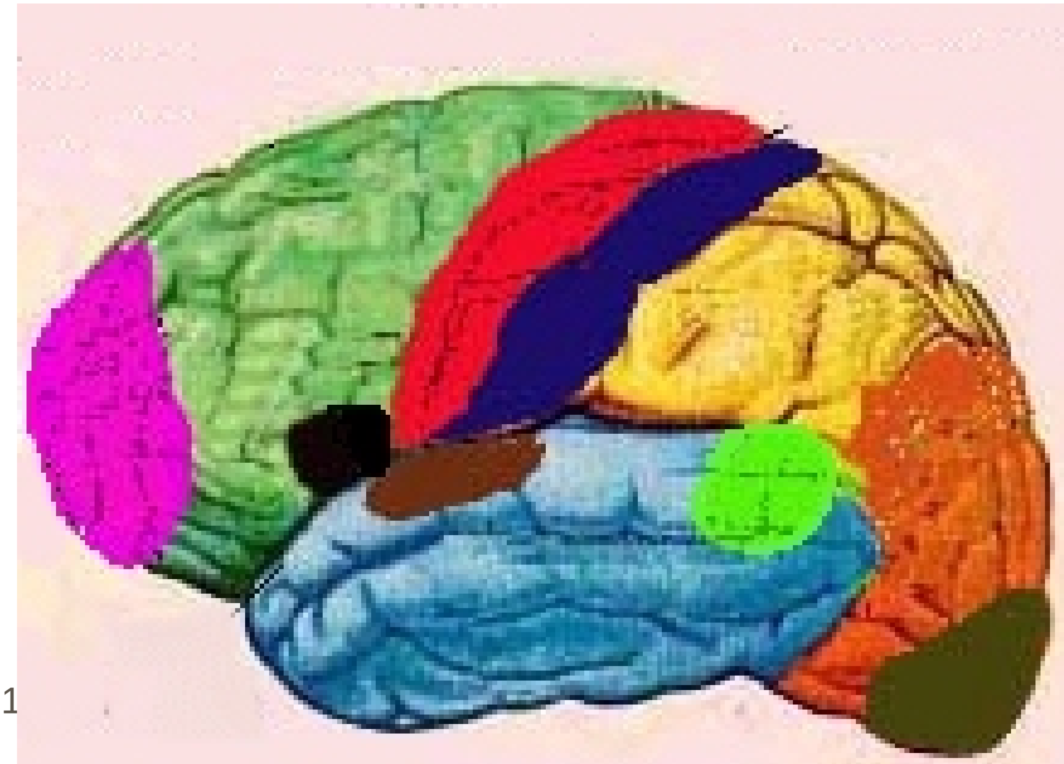
Brodmann's cytoarchitectural map of Cortical Areas



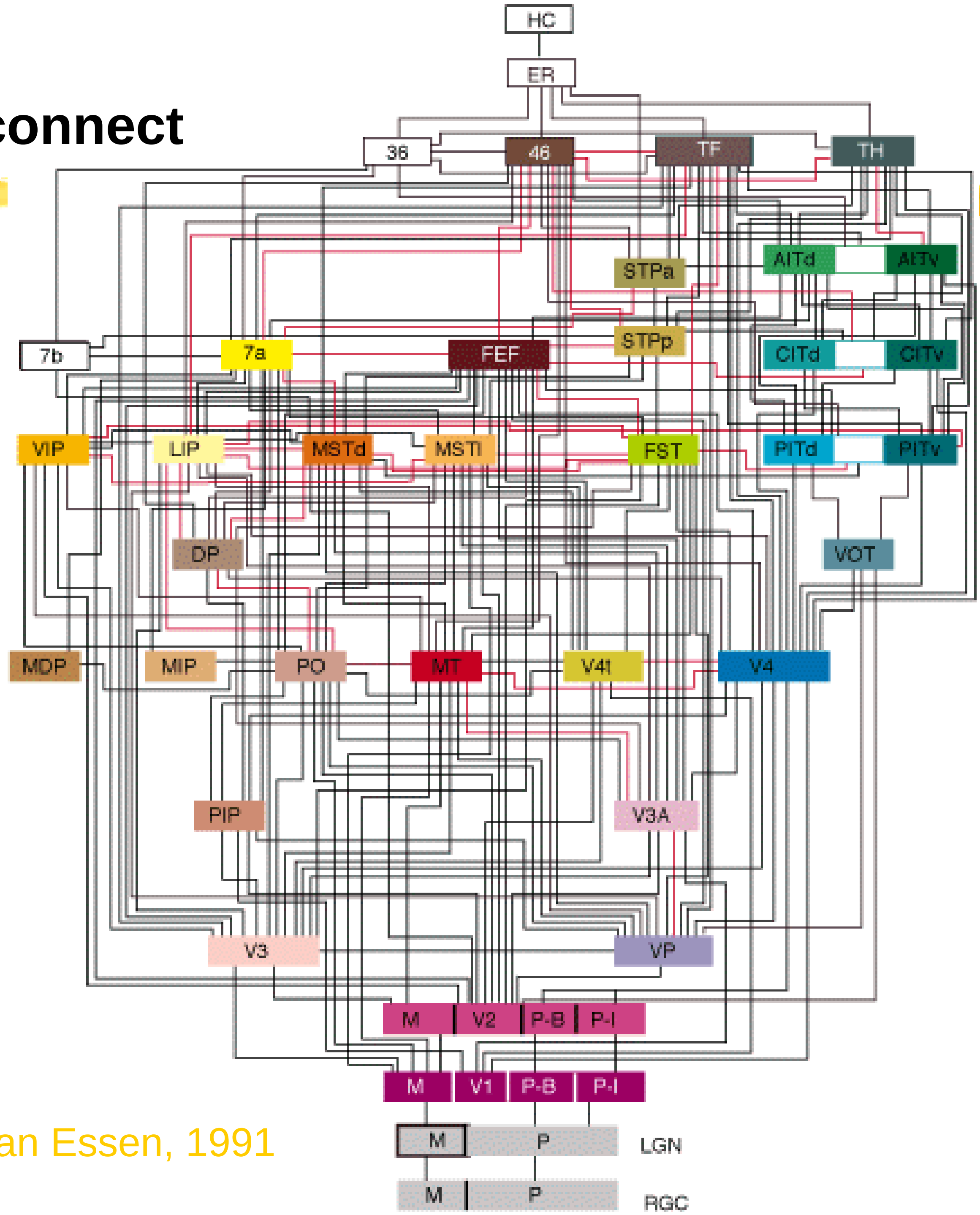
Lateral View

Major Functional Areas

- Primary motor: voluntary movement
- Primary somatosensory: tactile, pain, pressure, position, temp., mvt.
- Motor association: coordination of complex movements
- Sensory association: processing of multisensorial information
- Prefrontal: planning, emotion, judgement
- Speech center (Broca's area): speech production and articulation
- Wernicke's area: comprehen-
- sion of speech
- Auditory: hearing
- Auditory association: complex
- auditory processing
- Visual: low-level vision
- Visual association: higher-level
- vision



Interconnect



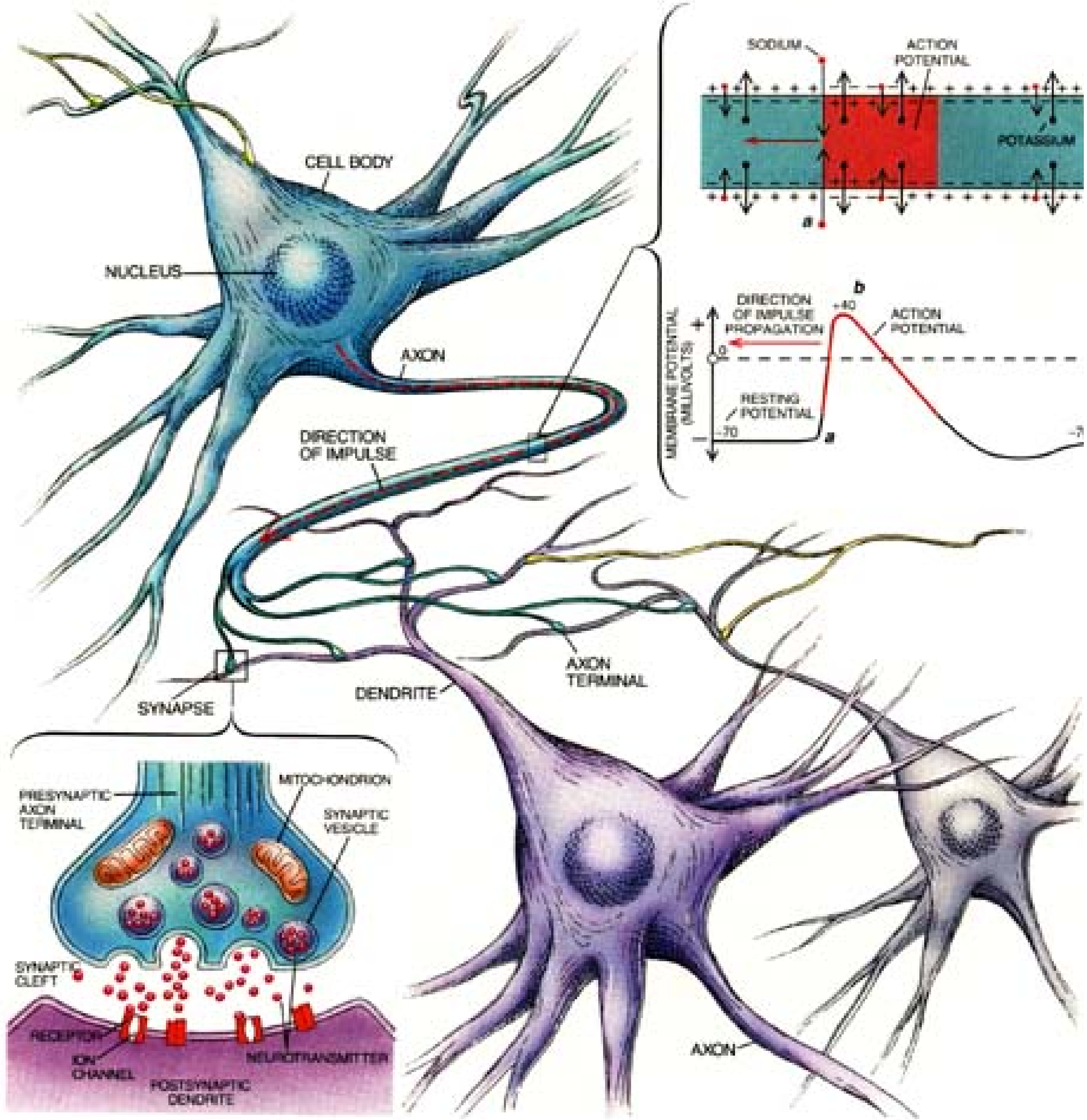
Felleman & Van Essen, 1991

More on Connectivity

Which brain area is connected to which other one,
And in which directions?

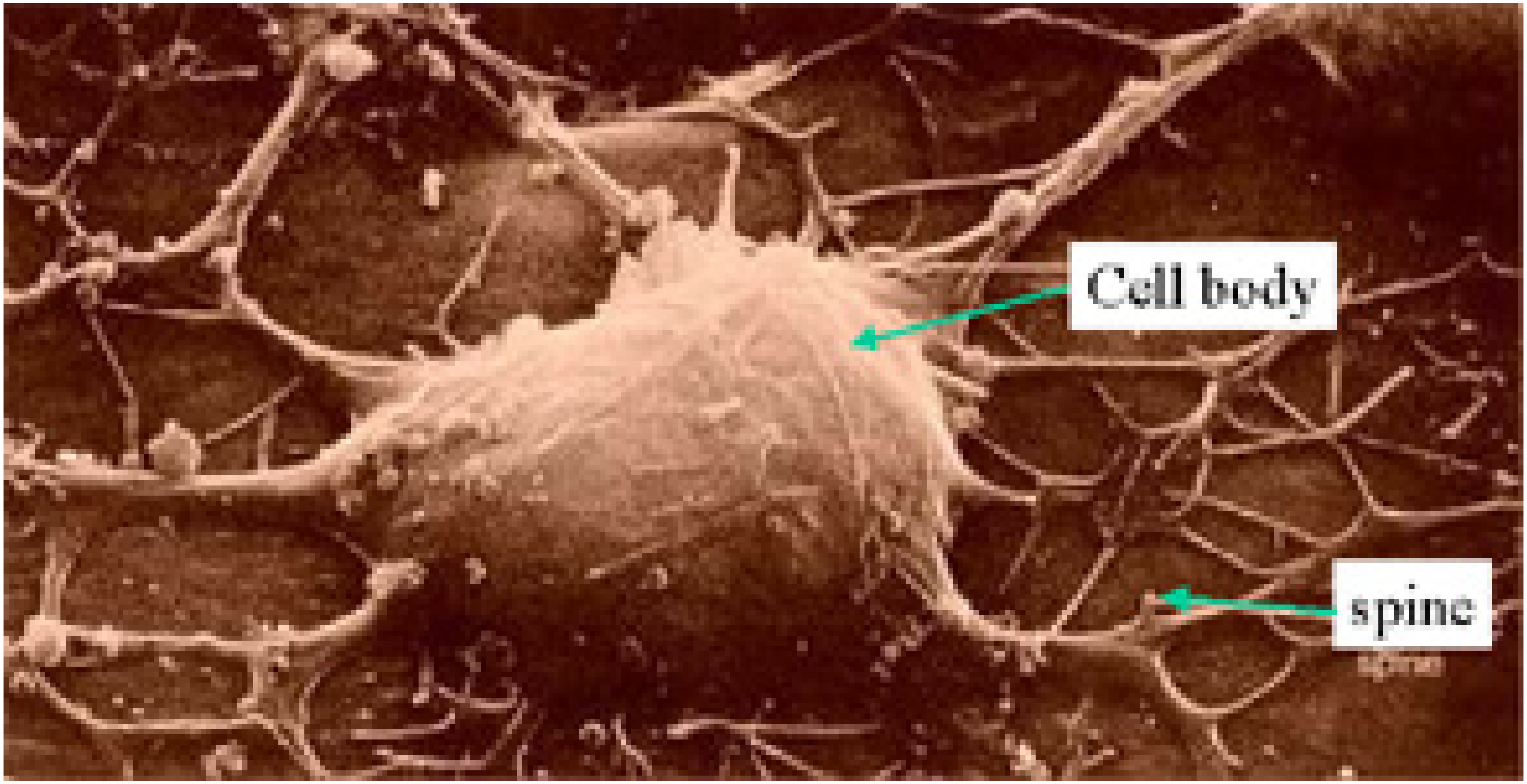
	V1	V2	VP	V3	PIP	V3A	PO	V4	MT	V4t	VOT	DP	LIP	VIP	MSTd	MSTl	PITd	PITv	7a	STPp	CITd	CITv	STPa	AITv	FEF	TF	46	FST	TH	AITd
V1	/	<	Ø	<	<	<	<	<	<	<	Ø	Ø	Ø	Ø	Ø	<?	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
V2	>	/	<	<		<	<	<	<	<		Ø	Ø		<	<	<	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	<	Ø	Ø	Ø
VP	Ø	>	/		<	≤	<	<	<	Ø	<	Ø	<	<	<	Ø	<	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	<	<	Ø	Ø	Ø
V3	>	>		/	<	<	<	<	<	<	Ø	Ø	<	<	<	Ø	<	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	<	<	Ø	Ø	Ø
PIP	>		>	>	/		<	<				<					Ø	Ø		Ø	Ø	Ø	Ø	Ø						Ø
V3A	>	>	≥	>		/	Ø	<	≤			<	<		<	<	<	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	<	Ø	Ø	Ø
PO	>	>	>	>	>		/		≤			≤	Ø	Ø	<	<	Ø	Ø	Ø	<	Ø	Ø	Ø	Ø	<	Ø		Ø	Ø	Ø
V4	>	>	>	>	>	>	Ø	/	≤	≤	<	<	<	Ø	Ø	Ø	<	<	<	Ø	Ø	<	<	Ø	<	<	<?	<	<	Ø
MT	>	>	>	>		≥	≥	≥	/	=		Ø	<	<	<	<	<	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	<	<	Ø	Ø
V4t	>	>	Ø	>				≥	=	/					<?	<?	<										<			
VOT	Ø	Ø	>	Ø				>			/							<	<											
DP	Ø	Ø	Ø	Ø	>	>	≥	>	Ø			/	≤		≤	<?	Ø	Ø	Ø	<	Ø	Ø	Ø	Ø		Ø	<		Ø	Ø
LIP	Ø		>	>		>		>	>			≥	/		≤		Ø	=	<		Ø	Ø		Ø	<	≤	<	≤	Ø	Ø
VIP	Ø		>	>					>					/	=	≤	Ø	Ø			Ø	Ø		Ø	<			≤		Ø
MSTd	Ø	>	>	>		>	>	Ø	>	>?		≥	≥	=	/		<?	<?	≤	<	Ø	Ø		Ø	<	<		≥		Ø
MSTl	>?	>	Ø	Ø	Ø	>	>	Ø	>	>?		>?		≥		/	Ø	Ø		<	Ø	Ø		Ø	<			≤		Ø
PITd	Ø	Ø	Ø	Ø	Ø	Ø	Ø	>	Ø		>		Ø	Ø	>?	Ø	/		Ø			<?		<?	<		<	≤		<?
PITv	Ø	Ø	Ø	Ø	Ø	Ø	Ø	>	Ø		>		=	Ø	>?	Ø		/	Ø		<?	<?		<?	<	<?	<	≤	<	<?
7a	Ø	Ø	Ø	Ø		Ø	>	Ø	Ø			>	>		≥		Ø	Ø	/		Ø	Ø			≤	<	≤?		<	<
STPp	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø			Ø		Ø	>	>				/	=	=	<			<		>	<	Ø
CITd	Ø	Ø	Ø	Ø	Ø	Ø	Ø	>	Ø			Ø		Ø	Ø	Ø		>	Ø	=	/			<?					<	<?
CITv	Ø	Ø	Ø	Ø	Ø	Ø	Ø	>	Ø			Ø		Ø	Ø	Ø	>	>	Ø	=		/		<?		<			<	<?
STPa	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø			Ø		Ø						>			/			<	<		<	
AITv	Ø	Ø	Ø	Ø	Ø	Ø	Ø	>	Ø			Ø		Ø	Ø	Ø	>?	>?			>?	>?	Ø	/		<		Ø	<	
FEF	Ø	>	>	>		>	>	>?	>	>			>	>	>	>			≥						/		≤			<
TF	Ø	Ø	>	>	Ø	Ø	Ø	>	Ø				≥	Ø	>			>	>	>		>	>	>		/	≤	≤		
46	Ø	Ø	Ø	Ø		Ø		>	>			>	>		Ø		>	>	≥?				>		≥	≥	/	Ø	≤	<
FST	Ø	>	>	>	Ø	>	Ø	>	>	>			≥	≥	≤	≥	≥	≥		<	Ø	Ø		Ø		≥	Ø	/		Ø
TH	Ø	Ø	Ø	Ø	Ø	Ø	Ø	>	Ø			Ø		Ø	Ø	Ø		>	>	>	>	>	>	>			≥		/	
AITd	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø			Ø		Ø	Ø	Ø	>?	>?	>		>?	>?			>		>	Ø		/

Remember?
Neurons & synapses



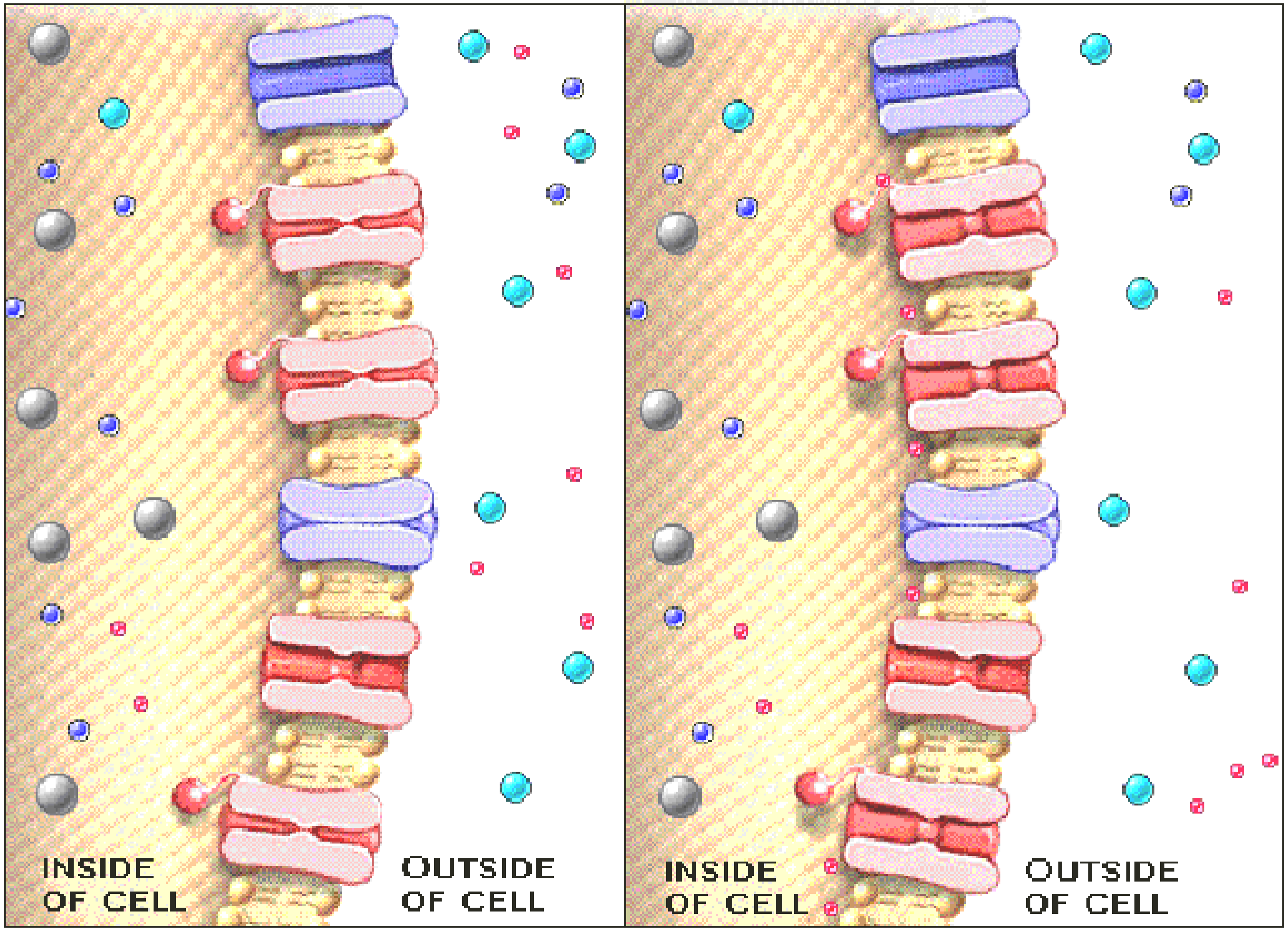
Key terms:
Axon
Dendrites
Synapses
Soma (cell body)

Electron Micrograph of a Real Neuron



Remember? Transmenbrane Ionic Transport

- ***Ion channels*** act as gates that allow or block the flow of specific ions into and out of the cell.



Approaches to neural modeling

- Biologically-realistic, detailed models
  - E.g., cable equation, multi-compartment models
  - The Hodgkin-Huxley model
  - Simulators like NEURON (Yale) or GENESIS (Caltech)

- More abstract models, still keeping realism in mind
  - E.g., integrate & fire model, simple and low detail but preserves spiking behavior

- Highly abstract models, neurons as operators
  - E.g., McCulloch & Pitts model
  - Classical "neural nets" modeling

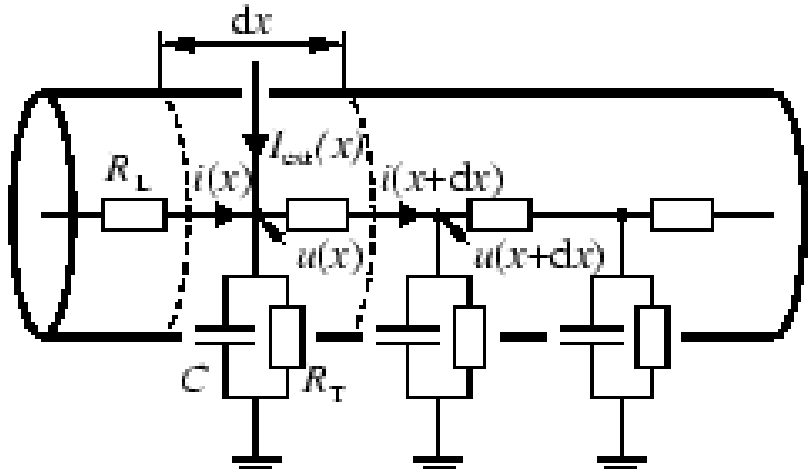
The Cable Equation

- See

http://diwww.epfl.ch/~gerstner/SPNM/SPNM.html

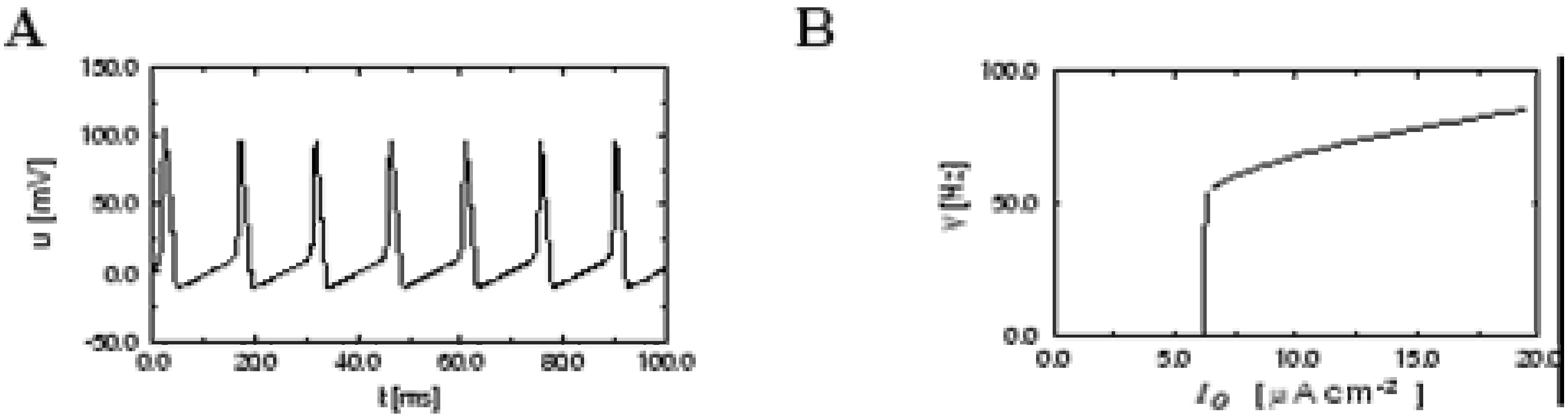
for excellent additional material (some reproduced here).

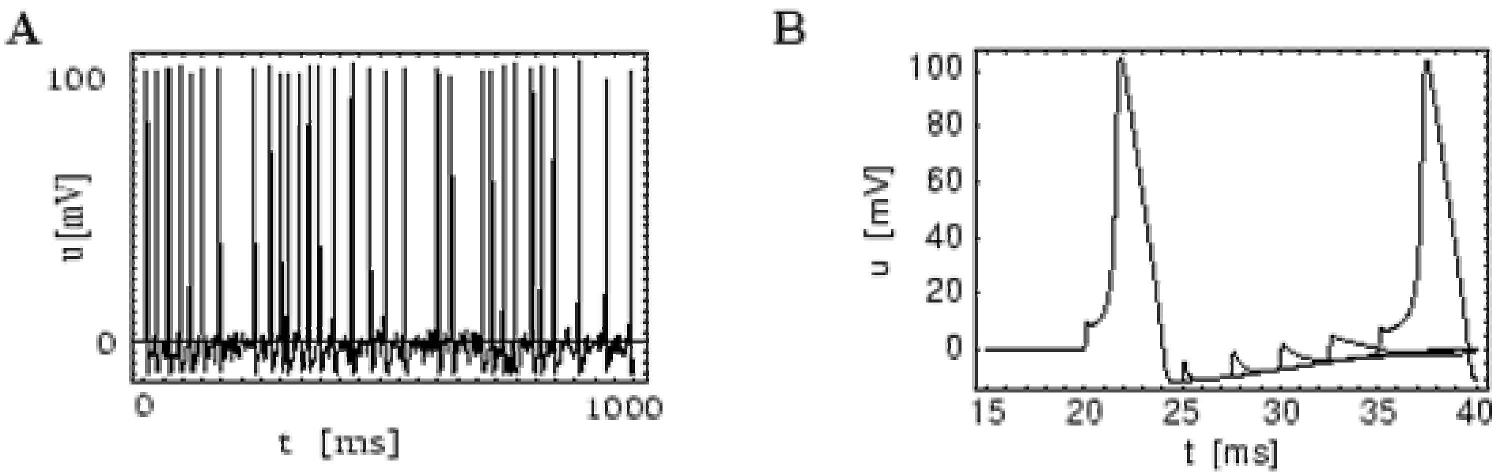
- Just a piece of passive dendrite can yield complicated differential equations which have been extensively studied by electronicians in the context of the study of coaxial cables (TV antenna cable):



The Hodgkin-Huxley Model

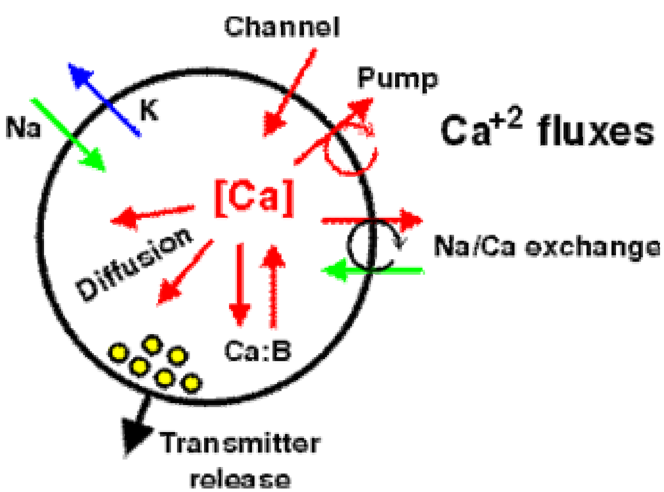
Example spike trains obtained...

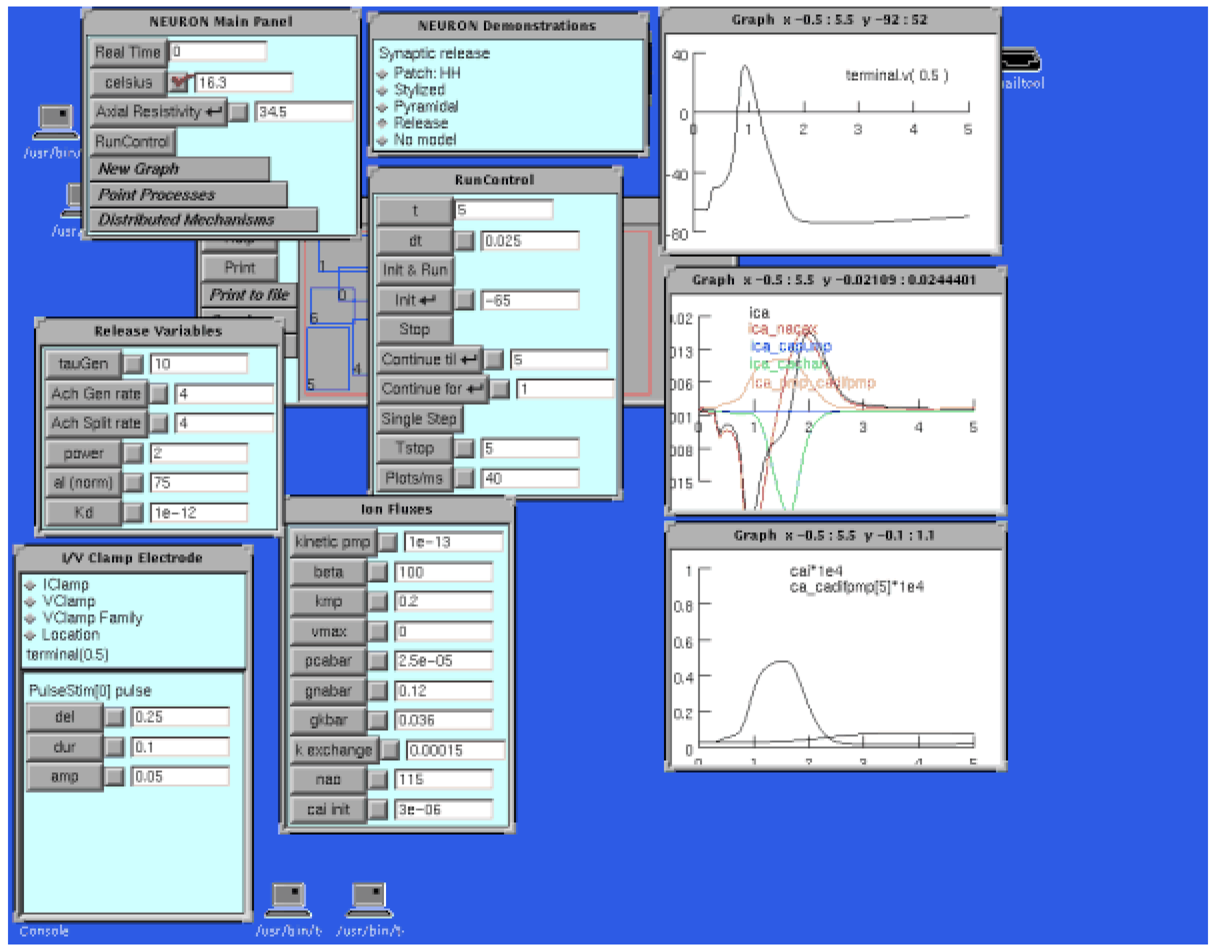




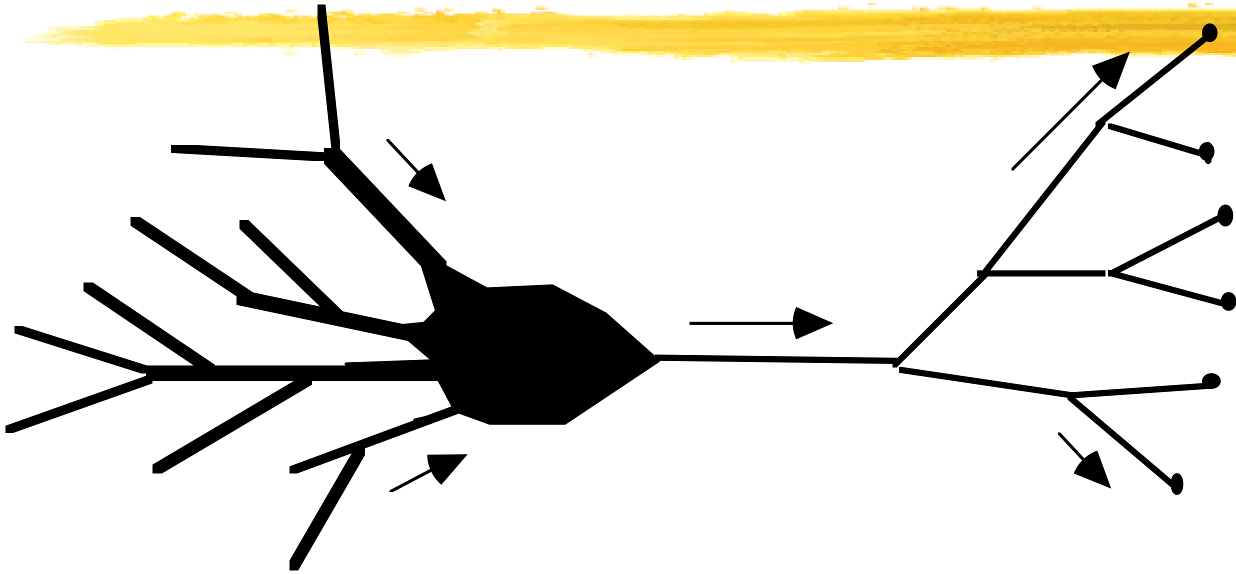
Detailed Neural Modeling

A simulator, called “Neuron” has been developed at Yale to simulate the Hodgkin-Huxley equations, as well as other membranes/channels/etc. See http://www.neuron.yale.edu/





The "basic" biological neuron

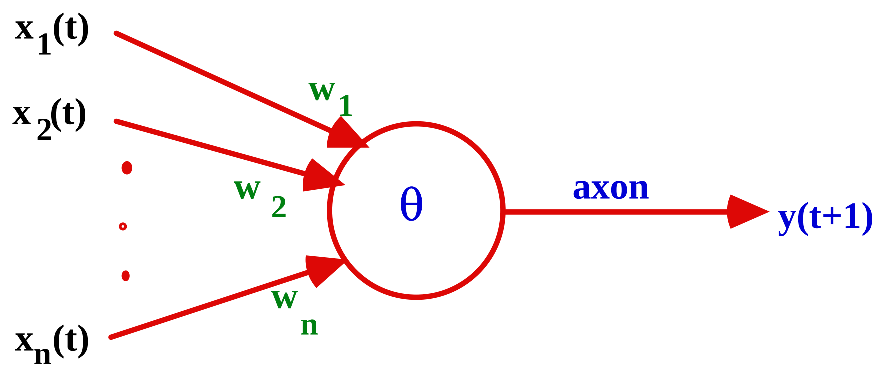


Dendritees Soma Axon with branches and synaptic terminals

- The soma and dendrites act as the input surface; the axon carries the outputs.
- The tips of the branches of the axon form synapses upon other neurons or upon effectors (though synapses may occur along the branches of an axon as well as the ends). The arrows indicate the direction of "typical" information flow from inputs to outputs.

Warren McCulloch and Walter Pitts (1943)

- **A** McCulloch-Pitts neuron operates on a discrete time-scale, t = 0,1,2,3, ...   with time tick equal to one refractory period



- At each time step, an input or output is

  on or *off* — 1 or 0, respectively.

- Each connection or synapse from the output of one neuron to the input of another, has an attached weight.

Excitatory and Inhibitory Synapses

- We call a synapse

  excitatory if $w_i > 0$, and

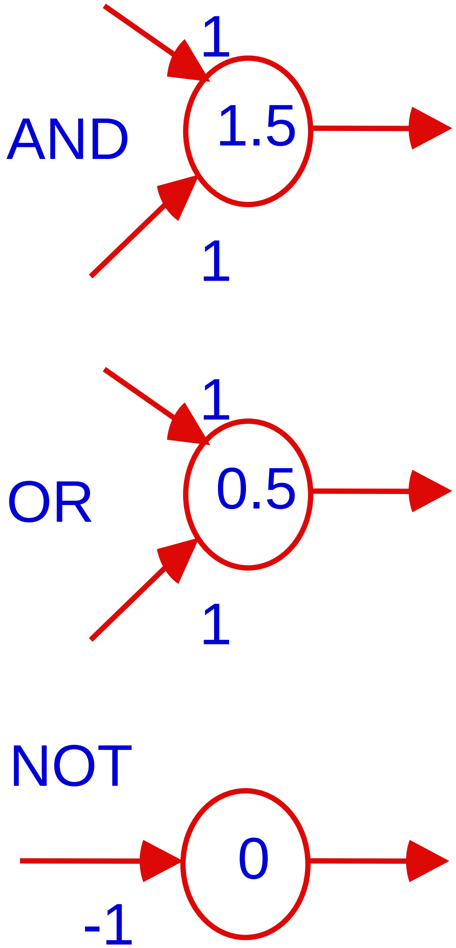
  inhibitory if $w_i < 0$.

- We also associate a threshold θ with each neuron

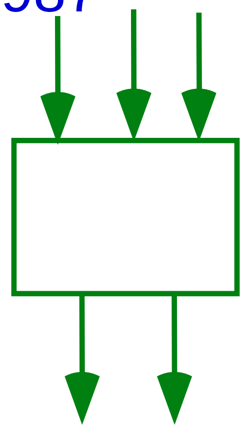
- A neuron fires (i.e., has value 1 on its output line) at time t+1 if the weighted sum of inputs at t reaches or passes θ:

$$\mathbf{y(t+1) = 1 \quad \text{if and only if} \quad \sum w_i x_i(t) \geq \theta}$$

From Logical Neurons to Finite Automata

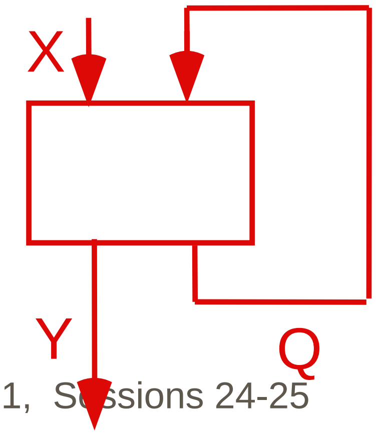


Brains, Machines, and Mathematics, 2nd Edition, 1987



Boolean Net

X → Y



Finite Automaton

Increasing the Realism of Neuron Models

- The McCulloch-Pitts neuron of 1943 is important

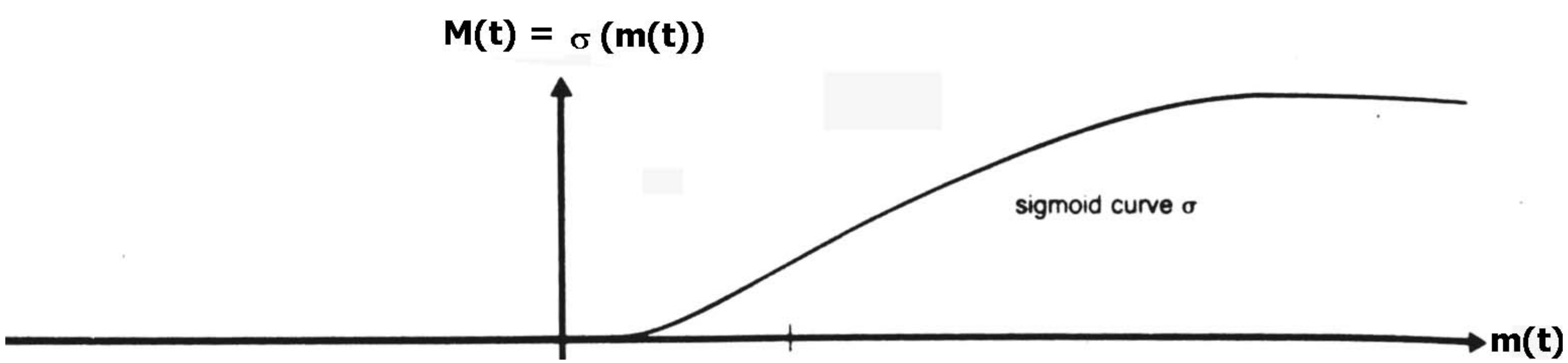
as a basis for

- logical analysis of the neurally computable, and
- current design of some neural devices (especially when augmented by **learning rules** to adjust synaptic weights).

- However, it is no longer considered a useful model for making contact with neurophysiological data concerning real neurons.

Leaky Integrator Neuron

- The simplest "realistic" neuron model is a continuous time model based on using the firing rate (e.g., the number of spikes traversing the axon in the most recent 20 msec.) as a continuously varying measure of the cell's activity
- The state of the neuron is described by a single variable, the membrane potential.
- The firing rate is approximated by a sigmoid, function of membrane potential.



Leaky Integrator Model

$$\tau \dot{m}(t) = -m(t) + h$$

has solution $m(t) = e^{-t/\tau} m(0) + (1 - e^{-t/\tau})h$

$\rightarrow h$ for time constant $\tau > 0$.

- We now add synaptic inputs to get the

Leaky Integrator Model:

$$\tau \dot{m}(t) = -m(t) + \Sigma_i w_i X_i(t) + h$$

where $X_i(t)$ is the firing rate at the i^{th} input.

- Excitatory input ($w_i > 0$) will increase $\dot{m}(t)$
- Inhibitory input ($w_i < 0$) will have the opposite effect.
- $X(t) = g(m(t))$ with g() a sigmoid relates output to membrane potential

Hopfield Networks

- A paper by John Hopfield in 1982 was the catalyst in attracting the attention of many physicists to "Neural Networks".

- In a network of McCulloch-Pitts neurons whose output is 1 iff $\Sigma w_{ij} s_j \geq \theta_i$ and is otherwise 0, neurons are updated synchronously: every neuron processes its inputs at each time step to determine a new output.

Hopfield Networks

- A Hopfield net (Hopfield 1982) is a net of such units subject to the asynchronous rule for updating one neuron at a time:

  "Pick a unit i at random.

  If $\Sigma w_{ij}\, s_j \geq \theta_i$, turn it on.

  Otherwise turn it off."

- Moreover, Hopfield assumes symmetric weights:

  $w_{ij} = w_{ji}$

“Energy” of a Neural Network

- Hopfield defined the “energy”:

$$E = - \tfrac{1}{2} \sum_{ij} s_i s_j w_{ij} + \sum_i s_i \theta_i$$

- If we pick unit i and the firing rule (previous slide) does not change its s_i, it will not change E.

s_i: 0 to 1 transition

- If s_i initially equals 0, and $\Sigma\, w_{ij}s_j \geq \theta_i$

then s_i goes from 0 to 1 with all other s_j constant, and the "energy gap", or change in E, is given by

$$\Delta E = -\tfrac{1}{2}\,\Sigma_j\,(w_{ij}s_j + w_{ji}s_j) + \theta_i$$

$$= -(\Sigma_j\, w_{ij}s_j - \theta_i) \qquad \text{(by symmetry)}$$

$$\leq 0.$$

s_i: 1 to 0 transition

- If s_i initially equals 1, and $\Sigma\ w_{ij}s_j < \theta_i$

then s_i goes from 1 to 0 with all other s_j constant

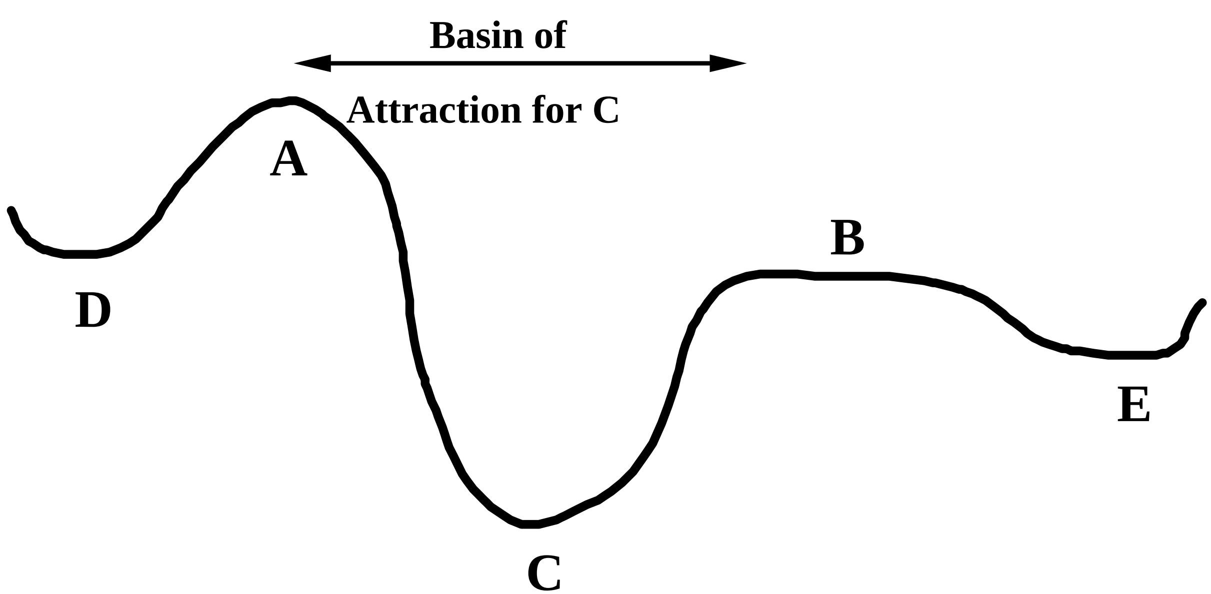
The "energy gap," or change in E, is given, for symmetric w_{ij}, by:

$$\Delta E = \Sigma_j\ w_{ij}s_j - \theta_i < 0$$

- ***On every updating we have $\Delta E \leq 0$***

Minimizing Energy

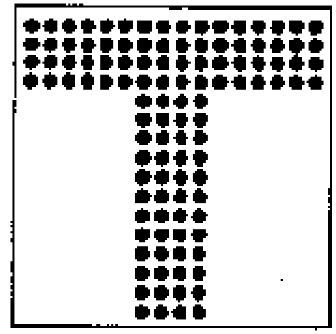
- On every updating we have $\Delta E \leq 0$
- Hence the dynamics of the net tends to move E toward a minimum.
- We stress that there may be different such states — they are *local* minima. Global minimization is not guaranteed.



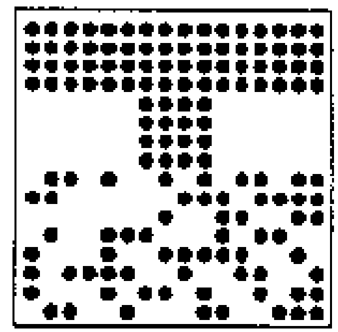
Associative Memories

- http://www.shef.ac.uk/psychology/gurney/notes/l5/l5.html

- Idea: store:

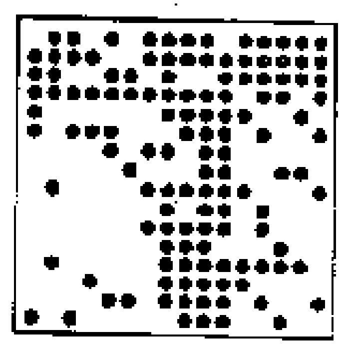


Original 'T'



half of image corrupted by noise

So that we can recover it if presented with corrupted data such as:



20% corrupted by noise (whole image)

Associative memory with Hopfield nets

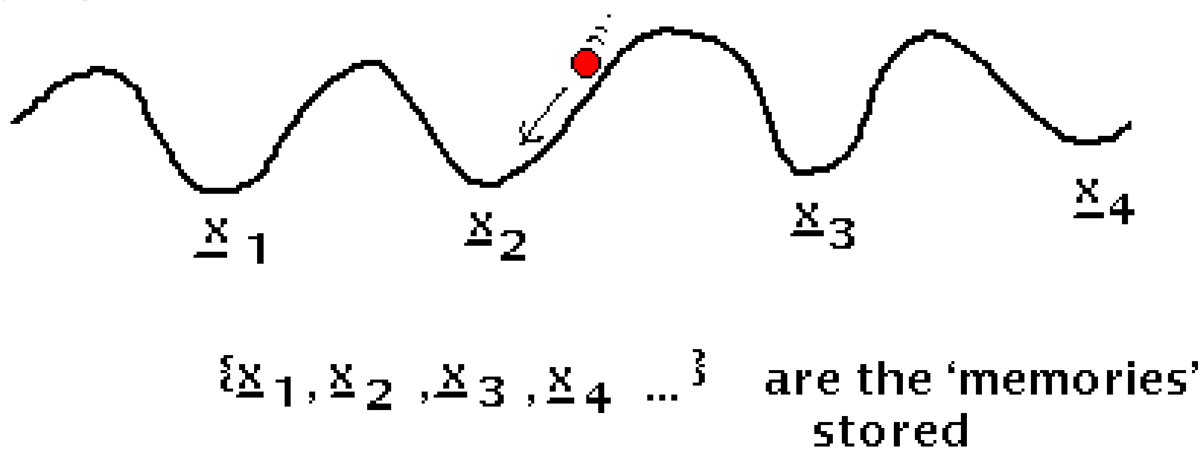
- Setup a Hopfield net such that local minima correspond
to the stored patterns.

- Issues:

  - because of weight symmetry, anti-patterns (binary reverse) are stored as well as the original patterns (also spurious local minima are created when many patterns are stored)

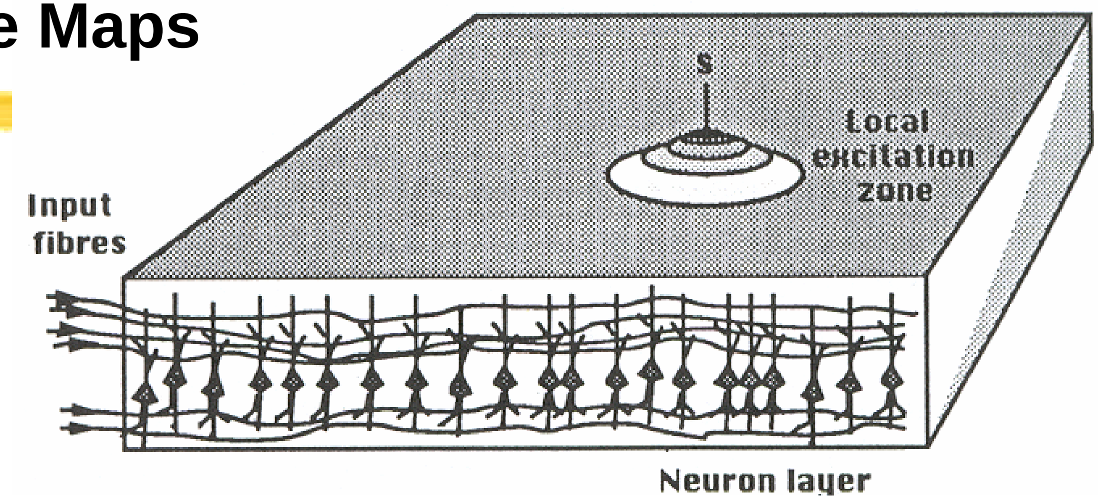
  - if one tries to store more than about **0.14*(number of neurons)** patterns, the network exhibits unstable behavior

  - works well only if patterns are uncorrelated



Self-Organizing Feature Maps

- The neural sheet is represented in a discretized form by a (usually) 2-D lattice A of formal neurons.



- The input pattern is a vector x from some pattern space V. Input vectors are normalized to unit length.
- The responsiveness of a neuron at a site r in A is measured by

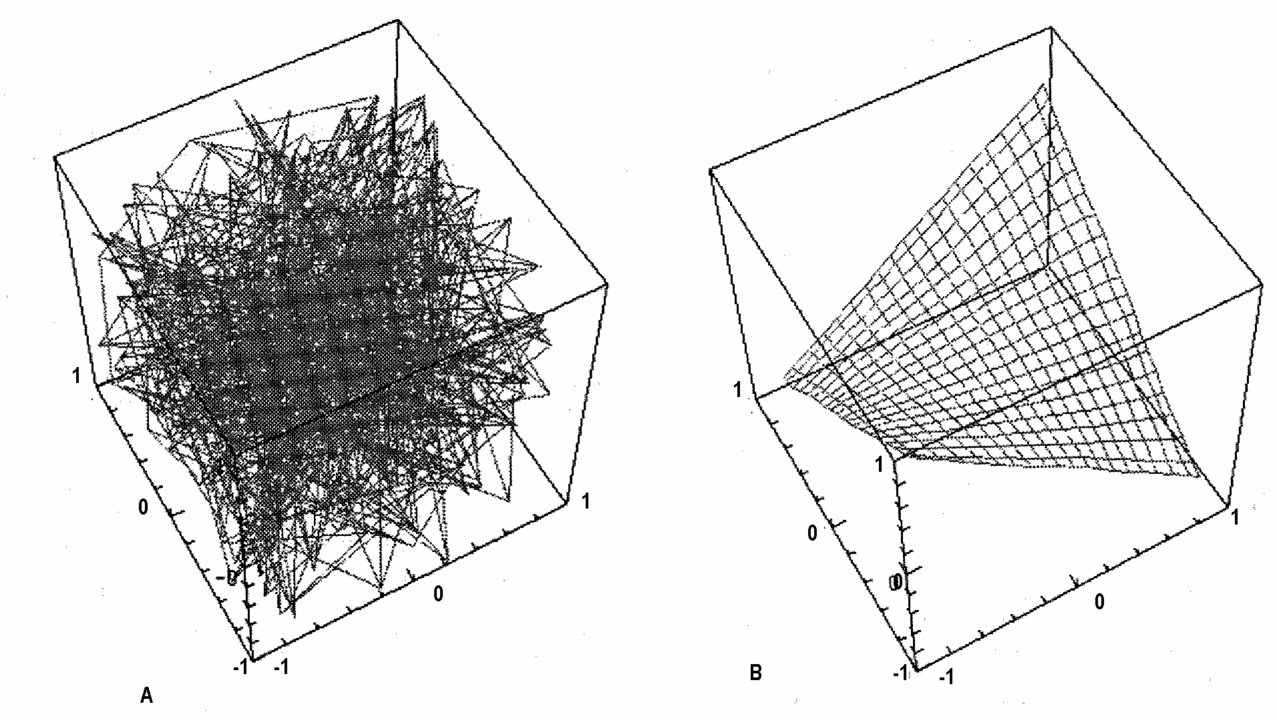
$$x.w_r = \Sigma_i x_i w_{ri}$$

where w_r is the vector of the neuron's synaptic efficacies.

- The "image" of an external event is regarded as the unit with the maximal response to it

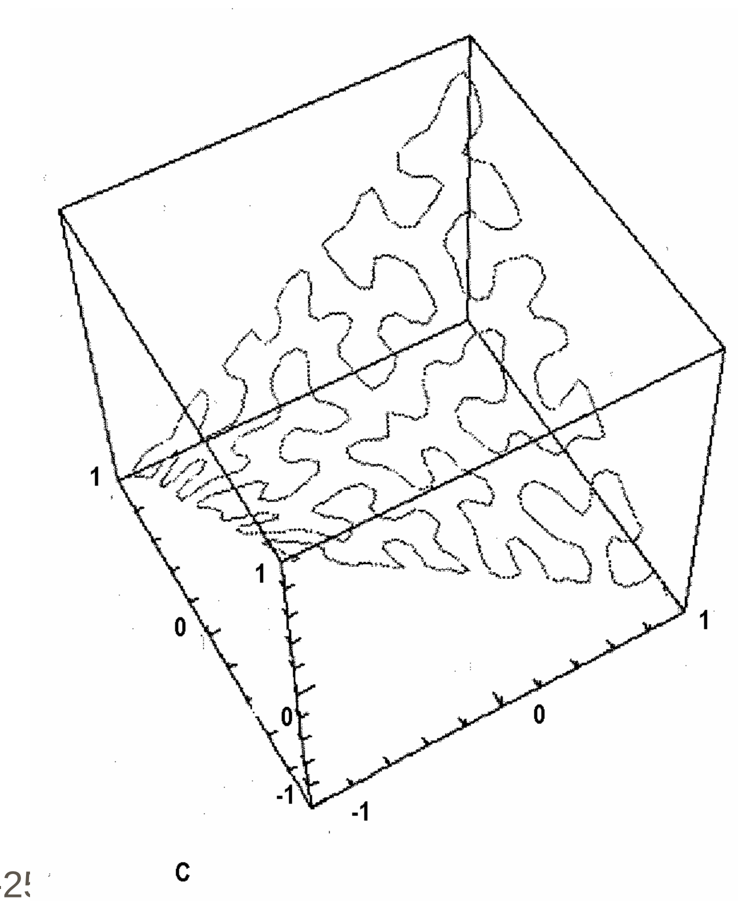
Self-Organizing Feature Maps

- Typical graphical representation: plot the weights (wr) as vertices and draw links between neurons that are nearest neighbors in A.



Self-Organizing Feature Maps

- These maps are typically useful to achieve some dimensionality-reducing mapping between inputs and outputs.



Applications: Classification

Business
- Credit rating and risk assessment
- Insurance risk evaluation
- Fraud detection
- Insider dealing detection
- Marketing analysis
- Mailshot profiling
- Signature verification
- Inventory control

Engineering
- Machinery defect diagnosis
- Signal processing
- Character recognition
- Process supervision
- Process fault analysis
- Speech recognition
- Machine vision
- Speech recognition
- Radar signal classification

Security
- Face recognition
- Speaker verification
- Fingerprint analysis

Medicine
- General diagnosis
- Detection of heart defects

Science
- Recognising genes
- Botanical classification
- Bacteria identification

Applications: Modelling

Business

- Prediction of share and commodity prices
- Prediction of economic indicators
- Insider dealing detection
- Marketing analysis
- Mailshot profiling
- Signature verification

Engineering

- Transducer linerisation
- Colour discrimination
- Robot control and navigation
- Process control
- Aircraft landing control
- Car active suspension control
- Printed Circuit auto routing
- Integrated circuit layout
- Image compression

Science

- Prediction of the performance of drugs from the molecular structure
- Weather prediction
- Sunspot prediction

Medicine

- . Medical imaging and image processing

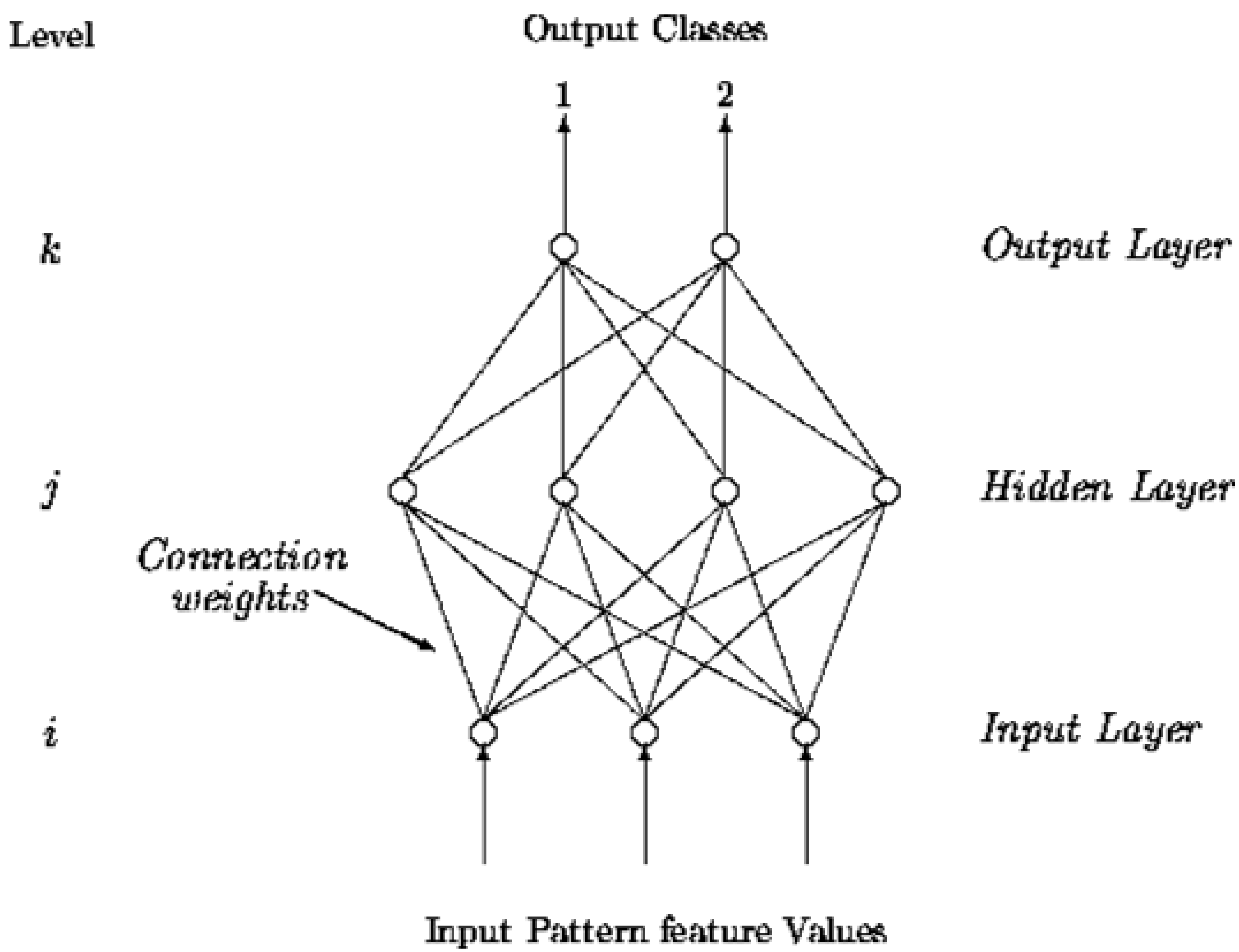
Applications: Forecasting

- Future sales
- Production Requirements
- Market Performance
- Economic Indicators
- Energy Requirements
- Time Based Variables

Applications: Novelty Detection

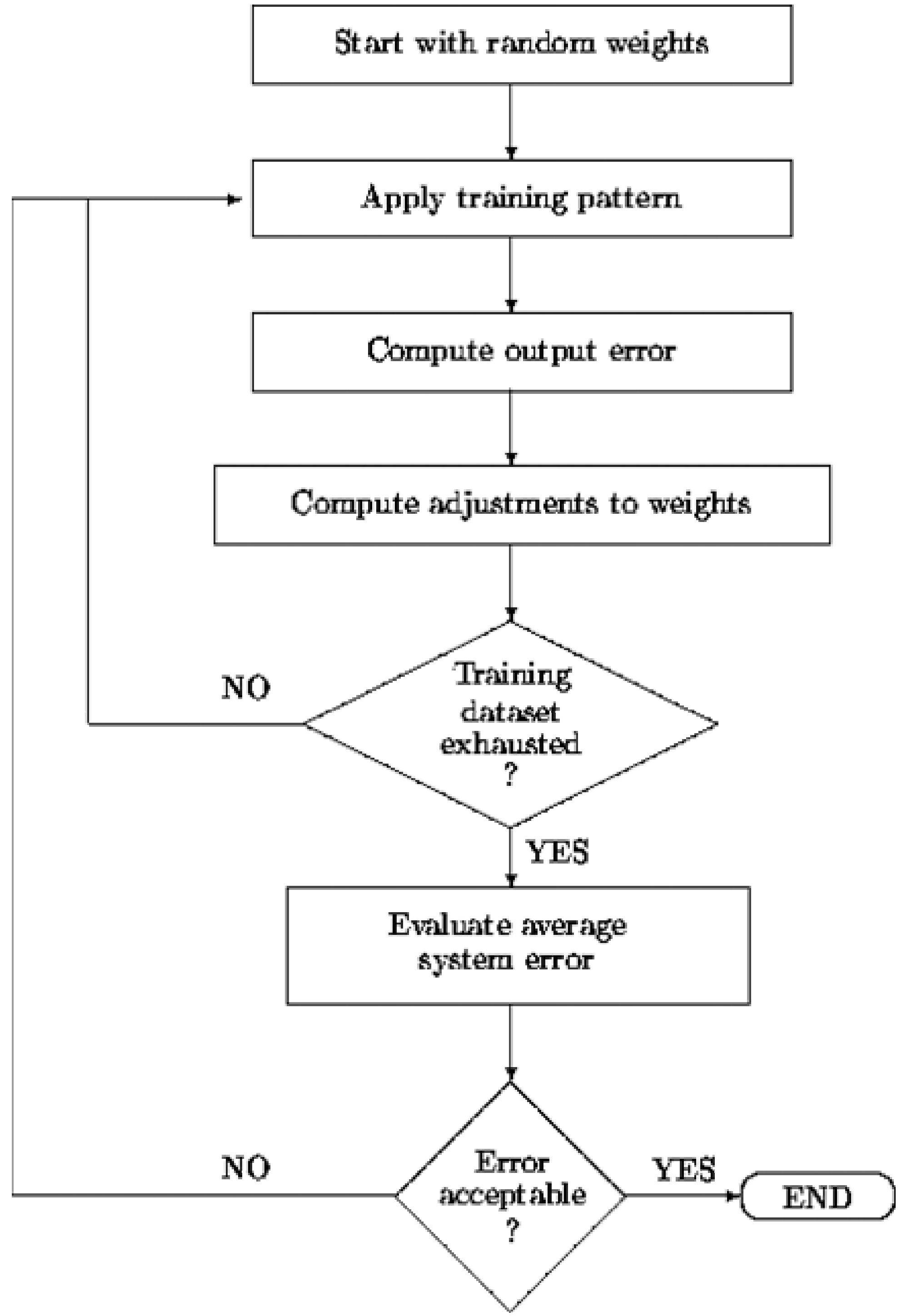
- Fault Monitoring
- Performance Monitoring
- Fraud Detection
- Detecting Rate Features
- Different Cases

Multi-layer Perceptron Classifier



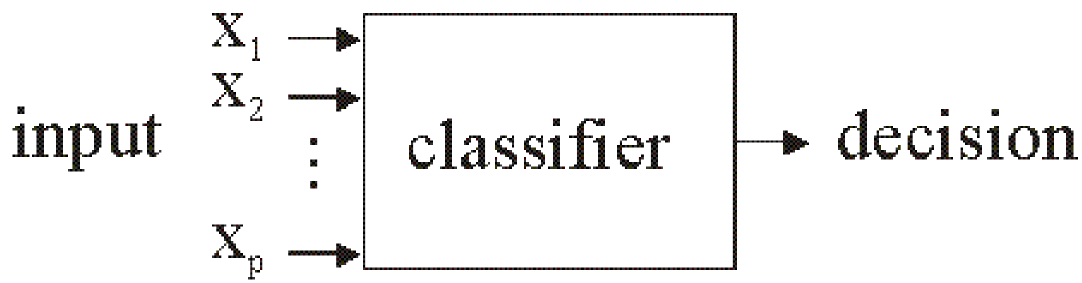
Multi-layer Perceptron Classifier

http://ams.egeo.sai.jrc.it/eurostat/Lot16-SUPCOM95/node7.html

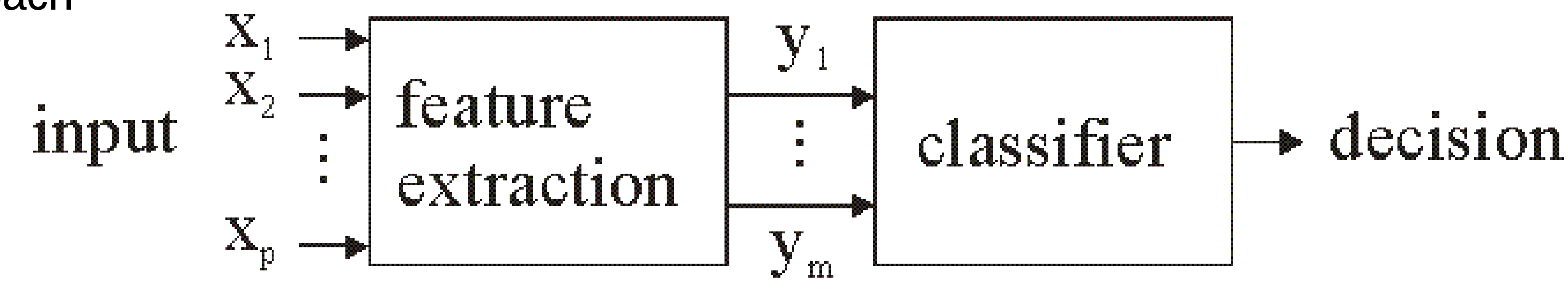


Classifiers

- http://www.electronicsletters.com/papers/2001/0020/paper.asp
- 1-stage approach

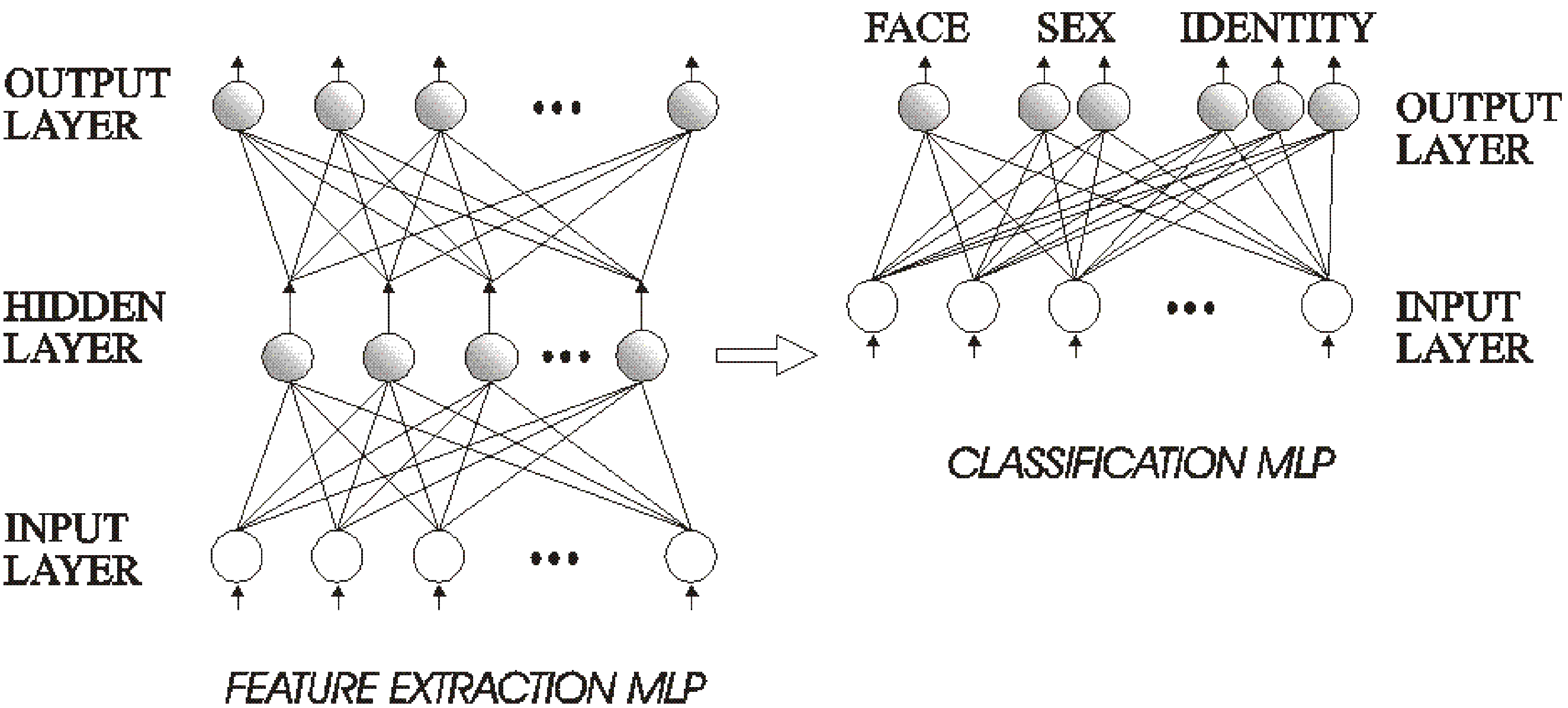


- 2-stage approach



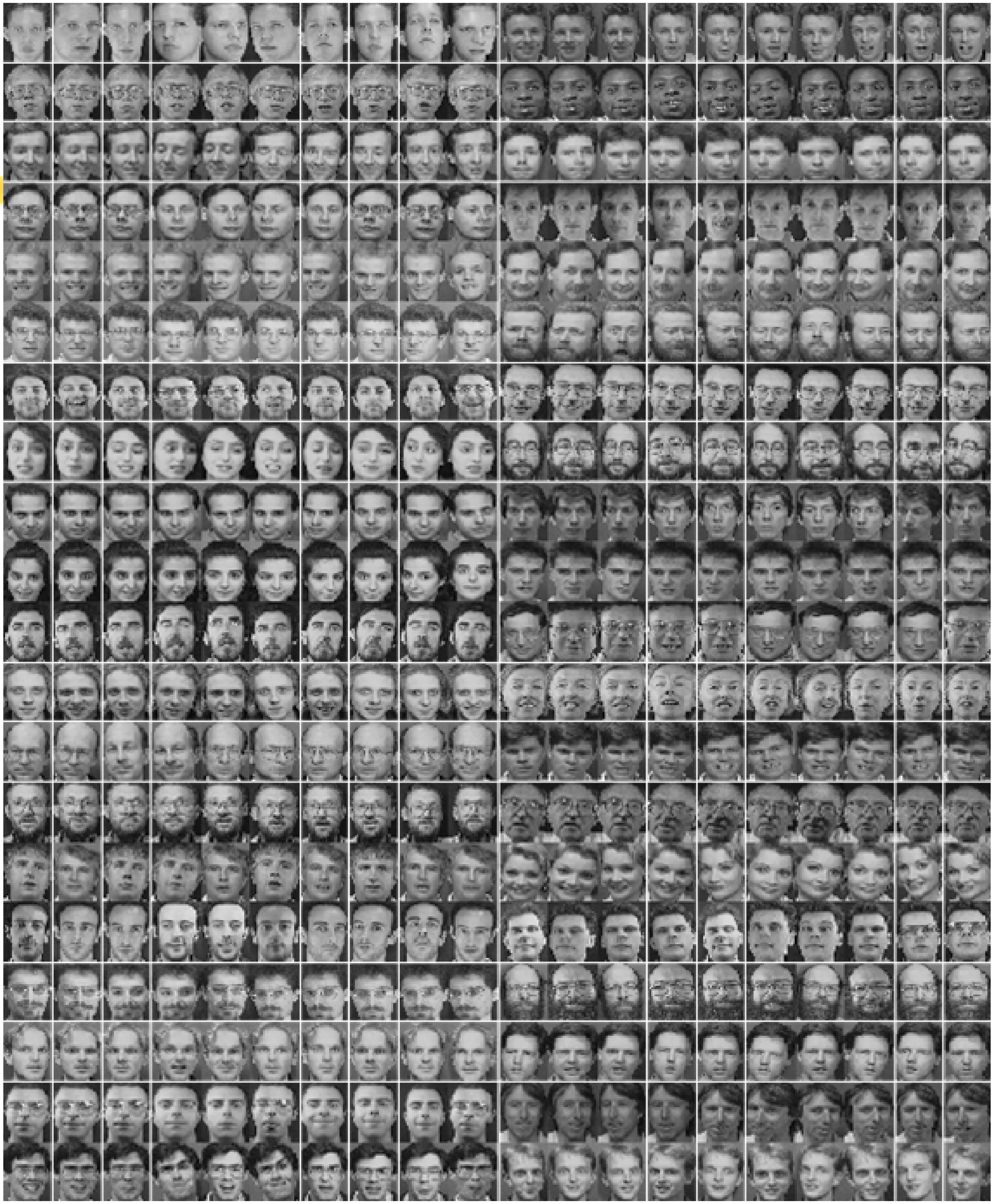
Example: face recognition

- Here using the 2-stage approach:

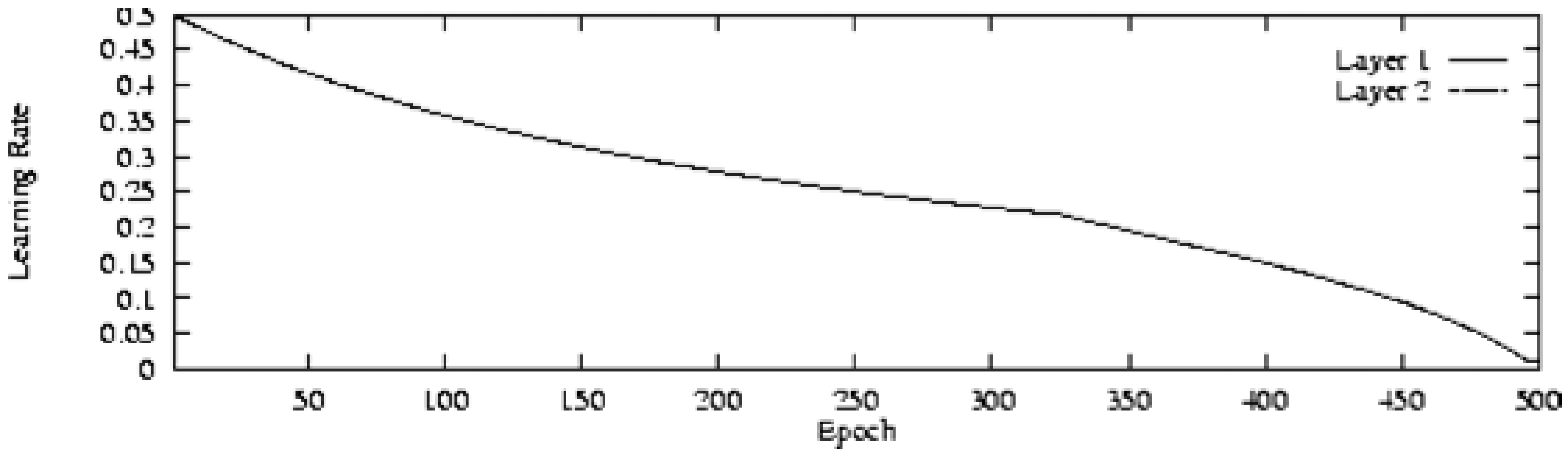


Training

- http://www.neci.nec.com/homepages/lawrence/papers/face-tr96/latex.html

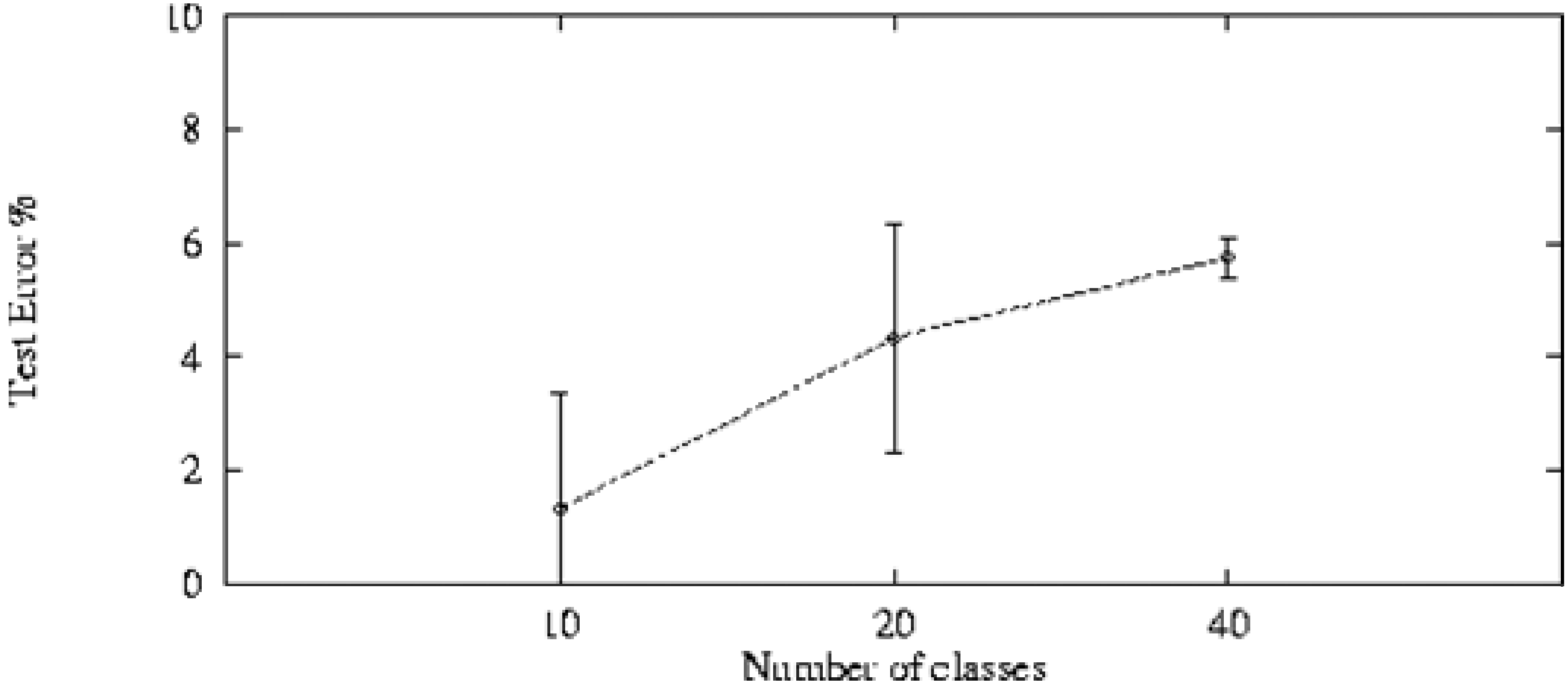


Learning rate



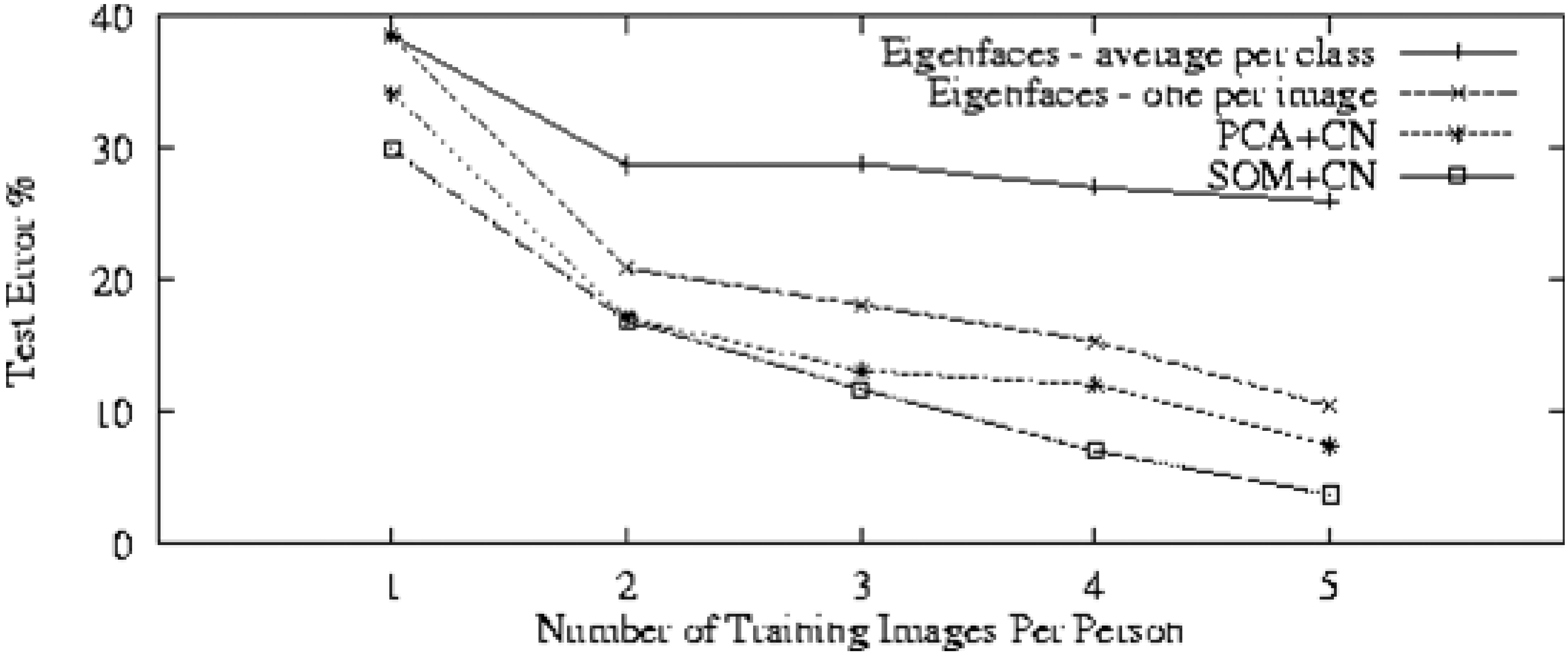
Testing / Evaluation

- Look at performance as a function of network complexity



Testing / Evaluation

- Comparison with other known techniques



Capabilities and Limitations of Layered Networks

- Issues:

- what can given networks do?
- What can they learn to do?
- How many layers required for given task?
- How many units per layer?
- When will a network generalize?
- What do we mean by generalize?
- ...

Capabilities and Limitations of Layered Networks

- What about boolean functions?

- Single-layer perceptrons are very limited:
  - XOR problem
  - etc.

- But what about multilayer perceptrons?

We can represent any boolean function with a network with just one hidden layer.

How??

Capabilities and Limitations of Layered Networks

To approximate a set of functions of the inputs by a layered network with continuous-valued units and sigmoidal activation function...

Cybenko, 1988: ... at most two hidden layers are necessary, with arbitrary accuracy attainable by adding more hidden units.

Cybenko, 1989: one hidden layer is enough to approximate any continuous function.

Intuition of proof: decompose function to be approximated into a sum of localized "bumps." The bumps can be constructed with two hidden layers.

Similar in spirit to Fourier decomposition. Bumps = radial basis functions.

Optimal Network Architectures

How can we determine the number of hidden units?

- genetic algorithms: evaluate variations of the network, using a metric that combines its performance and its complexity. Then apply various mutations to the network (change number of hidden units) until the best one is found.

- Pruning and weight decay:
  - apply weight decay (remember reinforcement learning) during training
  - eliminate connections with weight below threshold
  - re-train

- How about eliminating units? For example, eliminate units with total synaptic input weight smaller than threshold.

For further information

- See

Hertz, Krogh & Palmer: Introduction to the theory of neural computation (Addison Wesley)

In particular, the end of chapters 2 and 6.