Machine Learning

Susanne Still
University of Hawaii at Manoa

Learning and Adaptation

- Most intelligent systems show signs of learning.
- Most biological, “alive” systems utilize adaptation.

Challenges:

- Understand the principles of learning
- Build learning machines

- Machine learning is crucial in robotics and AI:

- Often easier to build a learning system than to hand-code a program that works.
- Example: a walking robot on the moon. Many DOFs; changing environment.

- Typical tasks that require learning:
  - Speech
  - Handwriting and object recognition
  - Intelligent user interfaces
  - Motor behavior

- A true AI requires learning on many levels!

- Machine learning crucial for data analysis:

- Huge and very complex data sets *too large to analyze by hand*; for example:
  - CERN
  - data from complex systems, e.g. ecological

- High frequency task, *too fast to analyze by hand*; e.g. stock price prediction from trading data.

- Human can solve task, but can not explain how; e.g. character recognition.

- No human expert; e.g. DNA analysis.

Application Areas:	Examples:
Physics	➡ particle physics
Bioinformatics	➡ microarray data
Computer Vision	➡ object recognition
Robotics	➡ decision making
Graphics	➡ realistic simulations
Speech	➡ recognition, identification
Financial analysis	➡ option pricing
E-commerce	➡ data mining
Medicine	➡ diagnostics, drug design
Computer games	➡ adaptive opponents
Multimedia	➡ retrieval across databases

Machine learning is one of the 21 century's core technologies.

Organizational items

- Ask questions! There is no such thing as a dumb question.
- Break long lectures into 2 parts to improve learning and retention of material.
- Lectures are being recorded.
- Reserve 10 minutes at the end for questions and informal discussion (cameras off).
- Homework problems, and exam at the end of the course.

Outline

1. Overview of some history and core ideas

- Physical limits to information processing
   2. Equilibrium thermodynamics applied to information processing
   3. Information processing far from thermodynamic equilibrium, a very brief introduction, *selected topics*.

4. From physical limits to information theory...
5. ...to unsupervised learning...
6. ...and cluster analysis.
7. Supervised learning: neural nets (mostly feed-forward)
8. Introduction to main ideas in statistical learning theory; Support vector machines, Kernel trick.
9. Bayesian Inference
10. Brief overview of selected "fashionable" ML techniques.

Machine learning for physicists!

1. Overview of some history and core ideas

- Physical limits to information processing

   2. Equilibrium thermodynamics applied to information processing
   3. Information processing far from thermodynamic equilibrium, a very brief introduction, *selected topics*.

4. From physical limits to information theory...
5. ...to unsupervised learning...
6. ...and cluster analysis.
7. Supervised learning: neural nets
8. Introduction to main ideas in statistical learning theory; Support vector machines, Kernel trick.
9. Bayesian Inference
10. Brief overview of selected "fashionable" ML techniques.

contact me

- don’t be shy, you can talk to me any time!
- email: sstill @ hawaii . edu
- my website needs updating, sorry www2.hawaii.edu/~sstill/

Learning machines

An introduction and some historical context

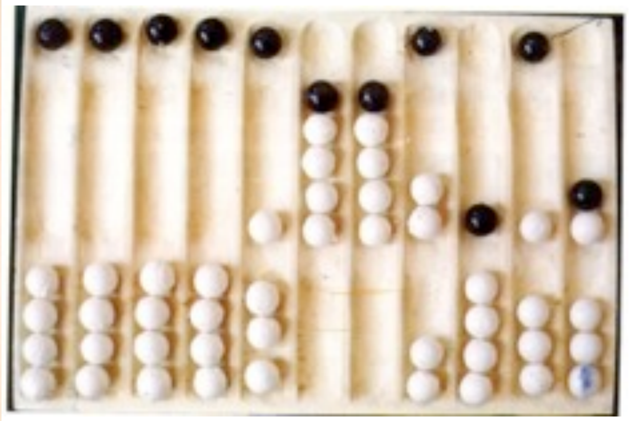
Machines that learn?

- Animals learn - how?
- What is learning?
- What is inference?
- What is a machine?
- Machines that can think?
- Machines that can do math?

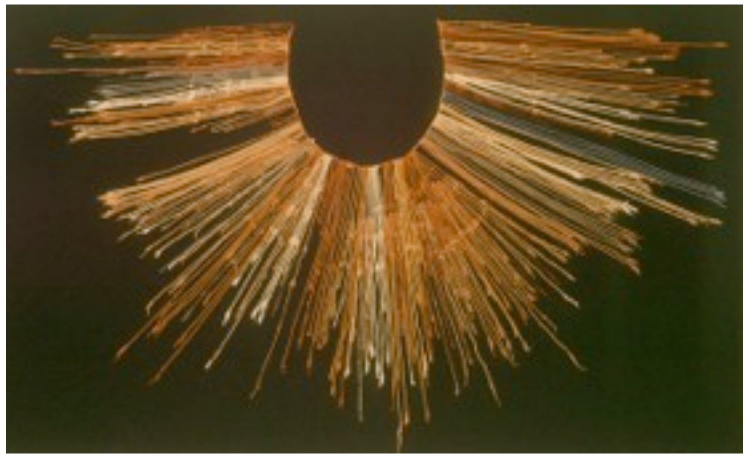
Machines that compute

- Devices that help perform simple calculations go back to ancient civilizations!

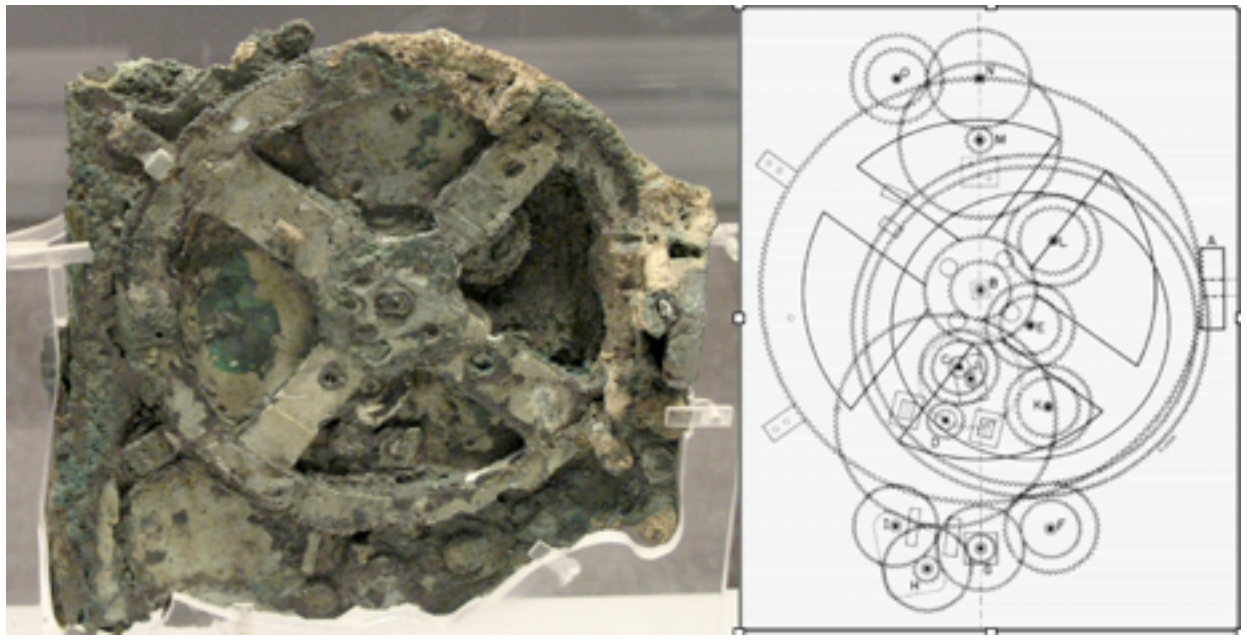
- Mesopotamia (ca. 2700BC): Abacus. Pebbles (latin: calculi)



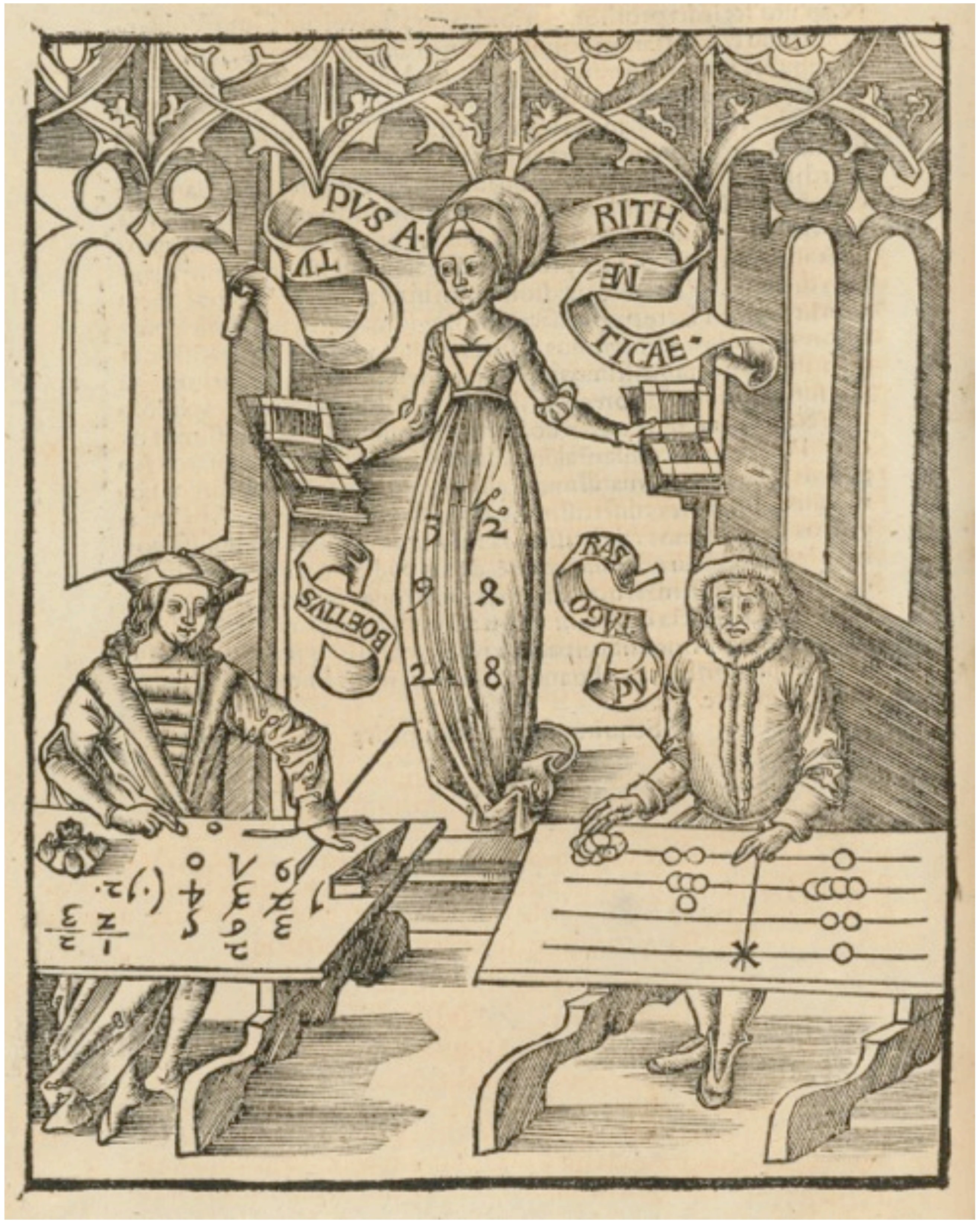
- South America (ca.2600BC): Quipu



- Predicting eclipses, "Antikythera mechanism" (ca. 200BC, 30 gears), "analog computation".



Abacus vs pen and paper



a competition, 1503

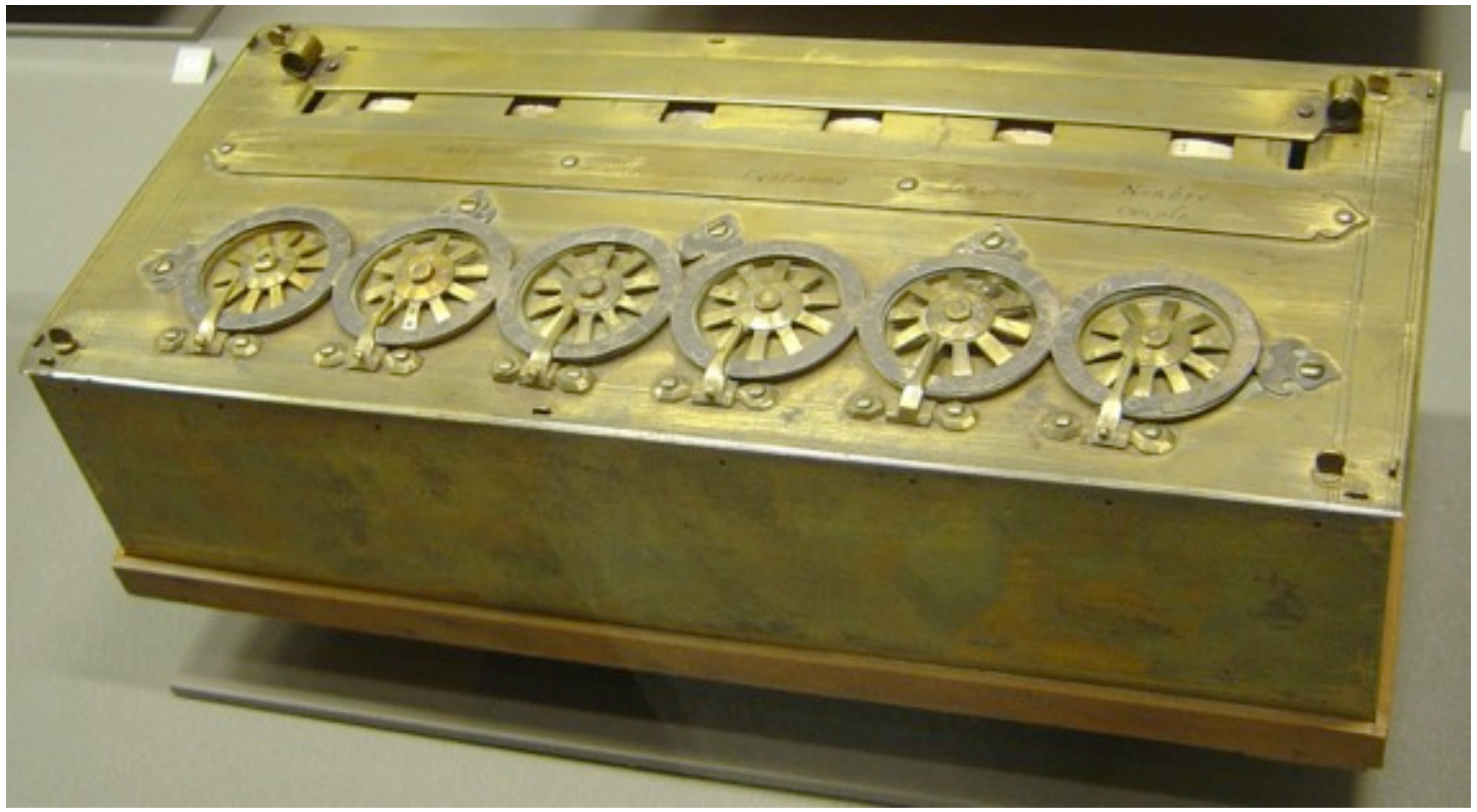
“abacist vs algorist”

abacus faster, but no written record

(introduction of arabic numerals in Europe)

Adding machine

- People make mistakes, arithmetic is tedious, so, let's build a machine!
- Pascal (1642)



Analog computing machines

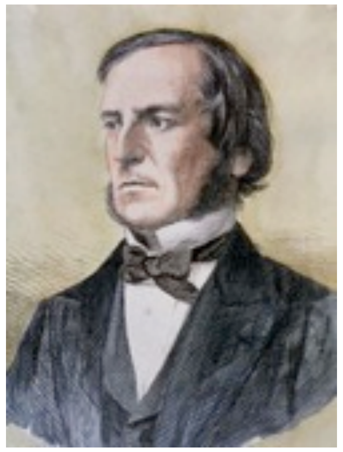
- Mechanical device for tide prediction based on Fourier analysis

- Lord Kelvin (1872)
  10 components

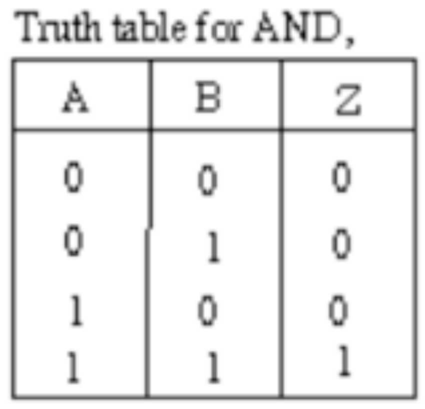
- 62 components,
  Germany (1935)
  (in use until 1968!)

“Modern” digital computer

- Based on boolean logic (George Boole 1847)

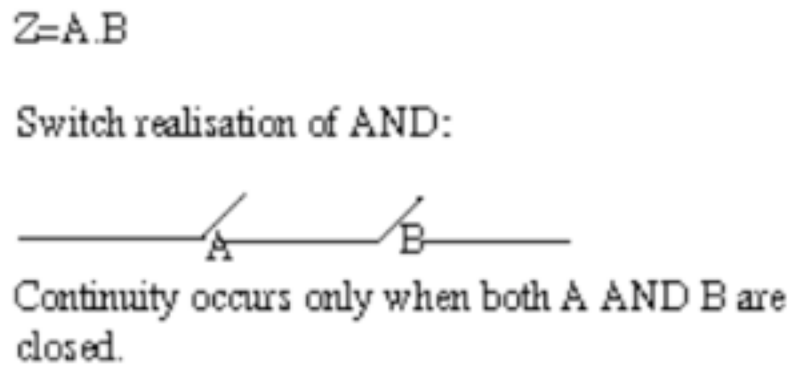


- Implemented basically via switches



Truth table for AND,

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1



Z=A.B

Switch realisation of AND:

Continuity occurs only when both A AND B are closed.

- Programmable

- Charles Babbage envisioned mechanical computers, but never completed one. (25,000 metal parts, 15 tons, precision issues.)

- Ada Lovelace wrote first “program” for his machine.

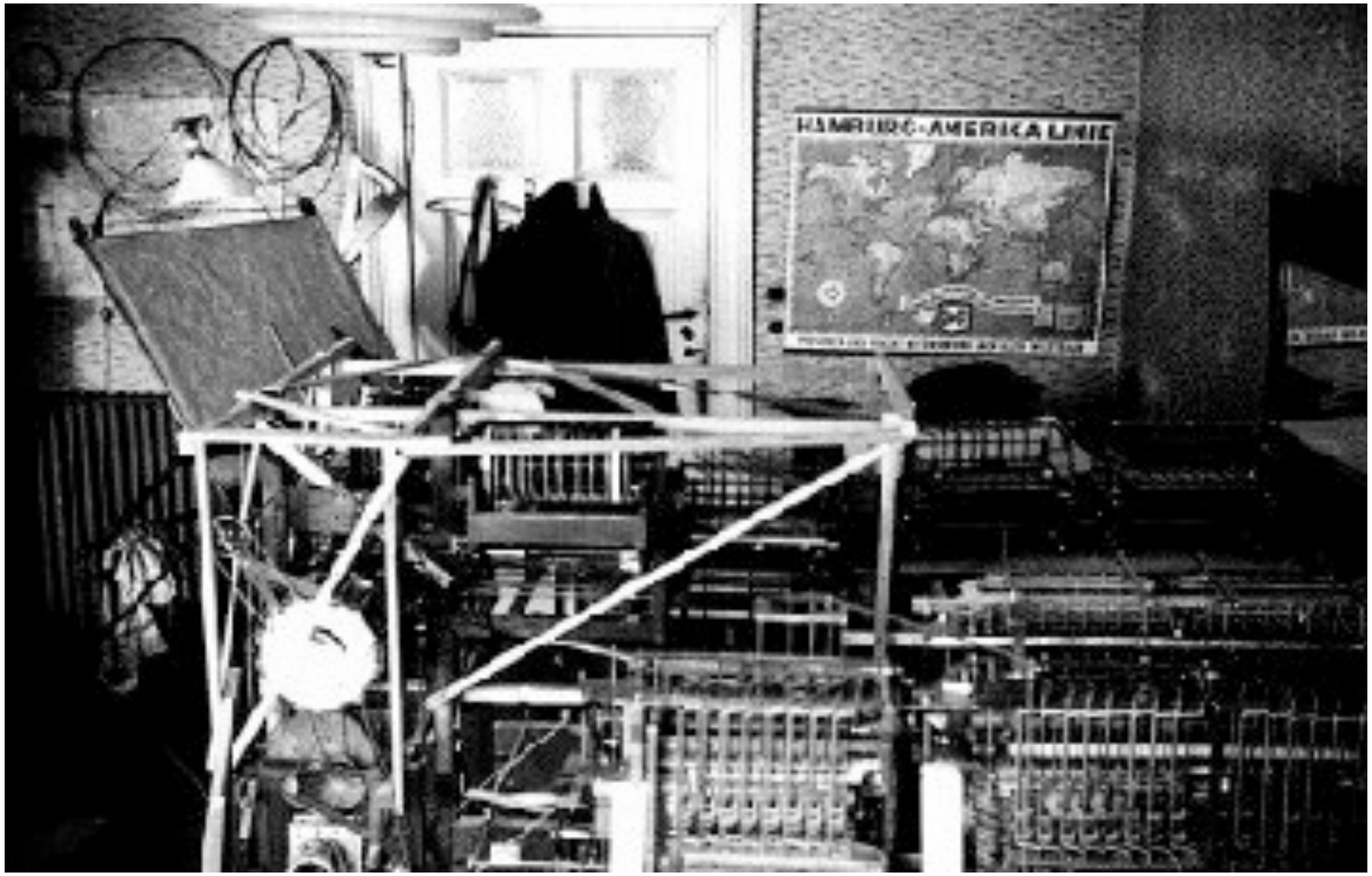
Probably the first built...

(Konrad Zuse 1938-41)

Z1-Z3, built in his parent's apartment in Berlin...

- 30,000 metal parts (reliability problems!)

Destroyed in 1944 by WW2 bombing



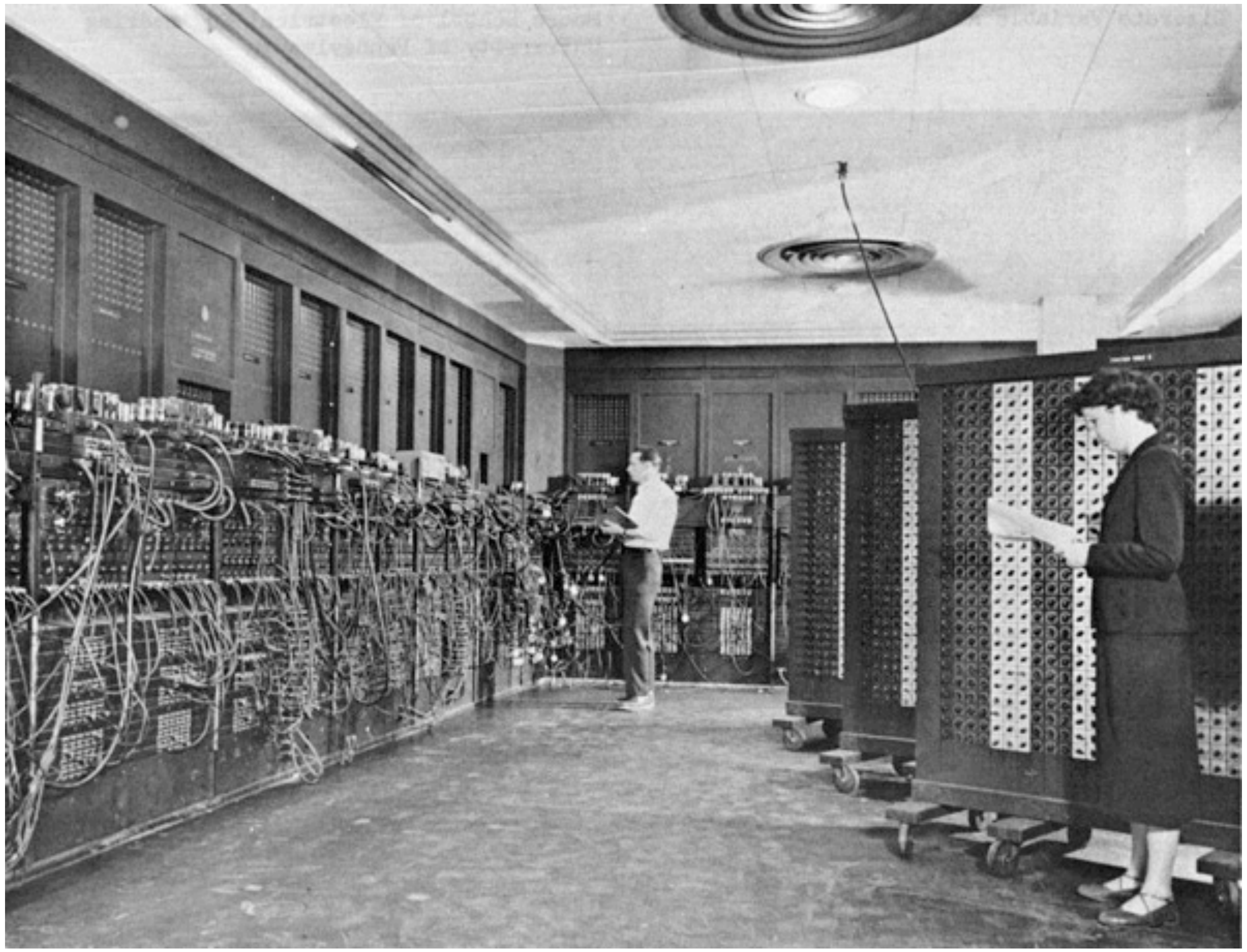
... the Z4 (1950)

First computer sold (that actually worked)
sold to ETH Zurich, Switzerland

- 2500 electro-mechanical relays
- ***256 byte memory*** (64 32-bit floating point words)
- ***one multiply every 3 seconds*** (modern laptops billions of times faster)

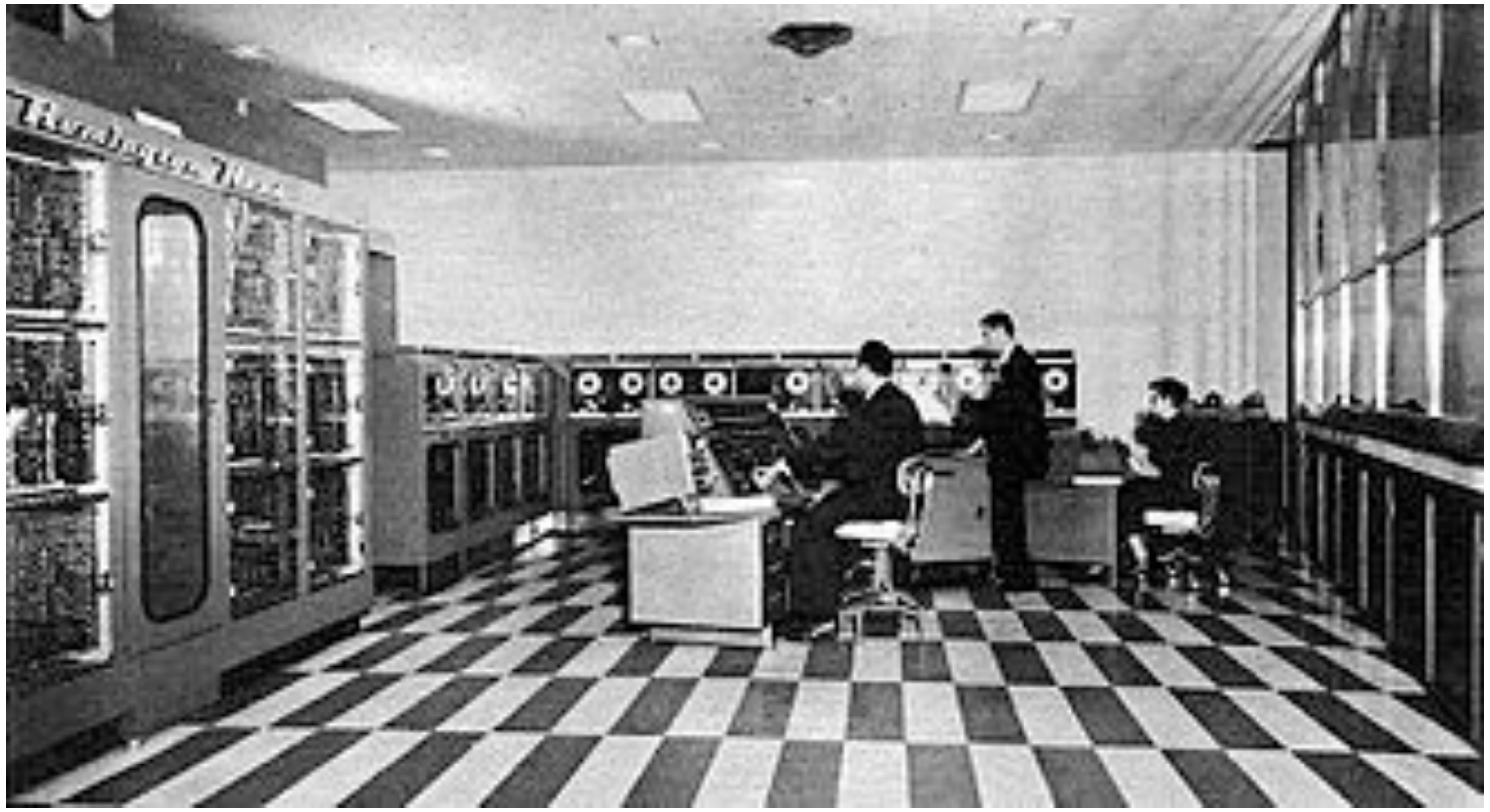
Another historical example: american Electronic Numerical Integrator And Computer (ENIAC, 1945)

- electronic (1000x faster than electromechanical; ***356 multiply/sec***)
- > 18,000 vacuum tubes
- programmed by plugboard and switches



UNIVAC

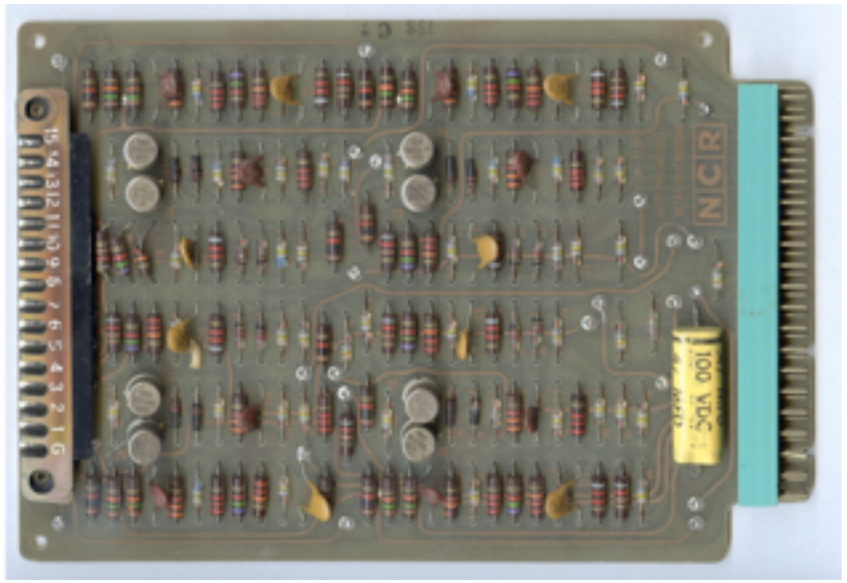
- first commercial computer produced in the US
- The first UNIVAC was accepted by the US Census Bureau on March 31, 1951



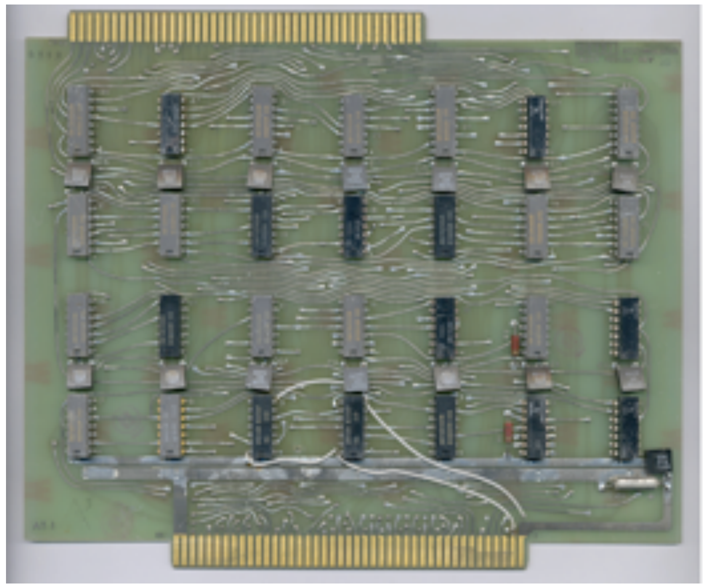
- 5,200 vacuum tubes, 29,000 pounds (13 metric tons), consumed 125 kW
- ***about 1,900 operations per second***

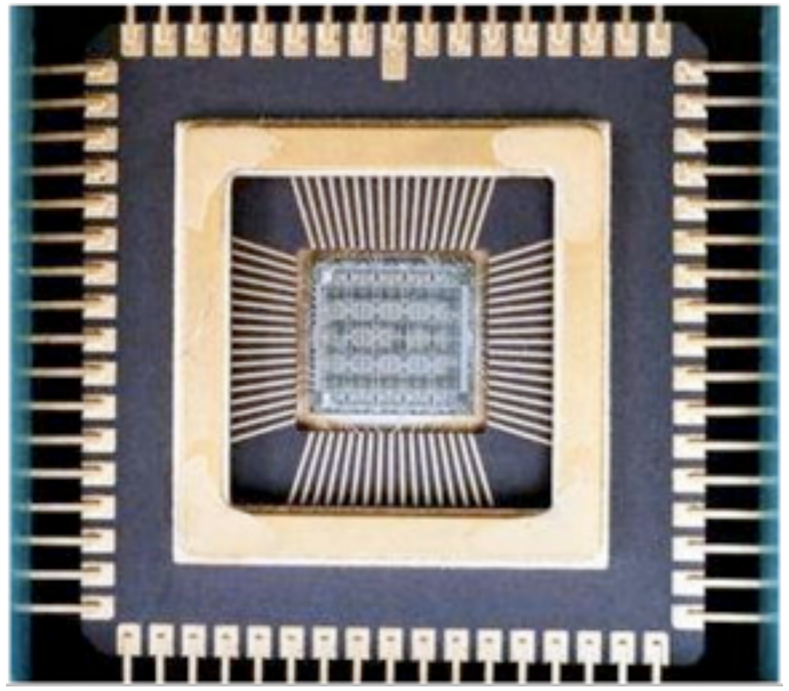
From vacuum tube computers, to transistor computers, integrated circuit computers, **but...**

tubes (4 logic gates, 1950s)



8 logic gates (1960s)


ICs hundreds (1970s)


ICs millions (1990s)

… basically

all digital computers are the same

The Turing-Church thesis (very informal)
any "computable function" that one computer can do,
any other digital computer can also do;
they are all are equivalent to Turing's machine.
(speeds may vary enormously)

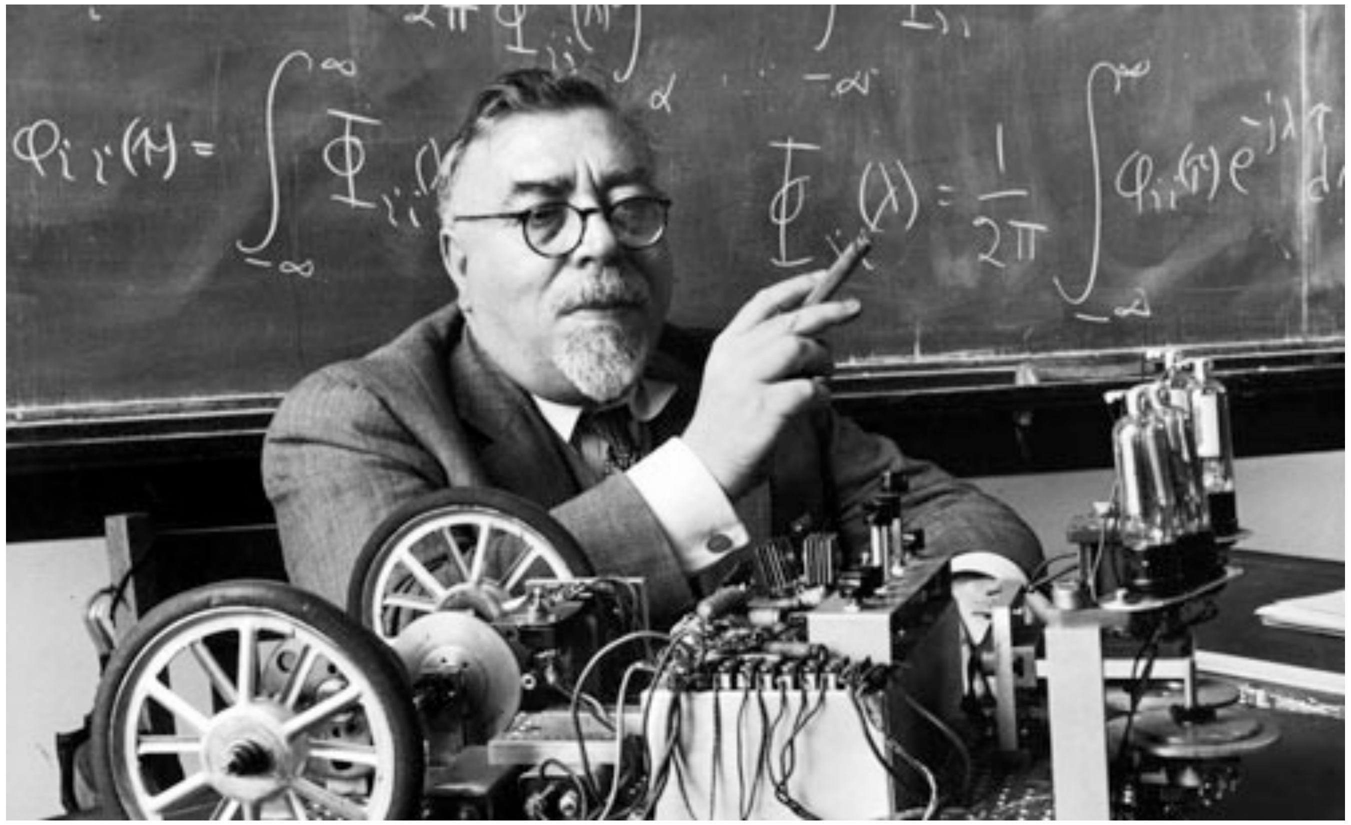
Origins of “modern” computer science (1930s)

- Foundations were laid by some breakthroughs starting in the 1930s with the work of Alan Turing, Alonzo Church Kurt Goedel, John v. Neumann, Emil L. Post, and others.
- ***Alan Turing (1936) introduced abstract model of a general purpose computer.***

Ambition: understand how living systems think; understand intelligence!

- Motivation driving the great minds that started this "new discipline", which is now Computer Science.
- Goes back to Alan Turing, John von Neumann, Norbert Wiener (cybernetics), and others, starting in the1930s
- *Physicists and mathematicians working together with neuroscientists*
- First models of neurons. (Pitts, McCulloch)
  => neural nets => machine learning

Cybernetics



- Information processing, control and (self-) regulation in the animal and the machine
- 10 Macy Conferences (1946-53); eclectic group of interdisciplinary participants.

Visionary field spun/influenced a many areas, for example:

- Control theory and *Information theory*
- Computational neuroscience
- *Neural networks (machine learning)*
- Neuromorphic engineering
- Theory of self-organization of living systems

Information theory

- Basis for communication (wired and wireless), e.g. transmitting TV signals, internet, phone.
- At the core lies the question:

What is information?
How to measure it?

- Original intellectual motivation: move from an energy based description of the world to an information based one.

Claude Shannon

- While Wiener attacked the hard problem of continuous information processing,
- Shannon made progress by first considering discrete symbols, showed that to fulfill simple assumptions, information is best measured by log(1/p)
- Average $$-\sum_x p(x)\log[p(x)]$$ is directly related to Gibbs entropy (e.g. E.T. Jaynes)
- Measures uncertainty, and the maximum (average) amount of information that can be gained by measuring the outcome of the random variable x

- **Information** is uncertainty reduction

$$\boxed{I[X,Y] = H[X] - H[X|Y]}$$

$$H[X] = -\sum_x p(x)\log[p(x)]$$

$$H[X|Y] = -\sum_{x,y} p(x,y)\log[p(x|y)]$$

- **Channel capacity** is the maximally transmittable (rate of) information $\max_{p(x)} I[x,y]$
  maximum over all information sources
  (x is input, y is output of the information channel)

- A **continuous signal** has infinite information rate.
- But infinite resolution is irrelevant for most applications, some level of distortion is tolerable.
- The achievable rate of a continuous information source, if transmitted to finite resolution, i.e. for fixed average distortion is:

$$R(D) := \min_{p(s|x)} I[s,x]$$

$$\text{s.t.} \langle d(s,x) \rangle_{p(s,x)} = D$$

- Represent original signal, x, by encoded signal, s. Given: distortion function d(s,x); information source p(x)
- (units: convert between information in bits per symbol, and rate in bits per second: multiply by a constant - symbols per second)

Computational neuroscience to neural networks

- Computational neuroscience produces mathematical models of neurons of varying degree of complexity
- One of the first was the McCulloch-Pitts model, pioneered by Pitts in the early 1940s
- Simple model leads to simplest "learning machine", the "Perceptron" (F. Rosenblatt 1957)
- From perceptron to multi-layer neural nets to the "Neocognitron", to "deep learning"...

...we will hear more about this after the break...

Neuromorphic engineering

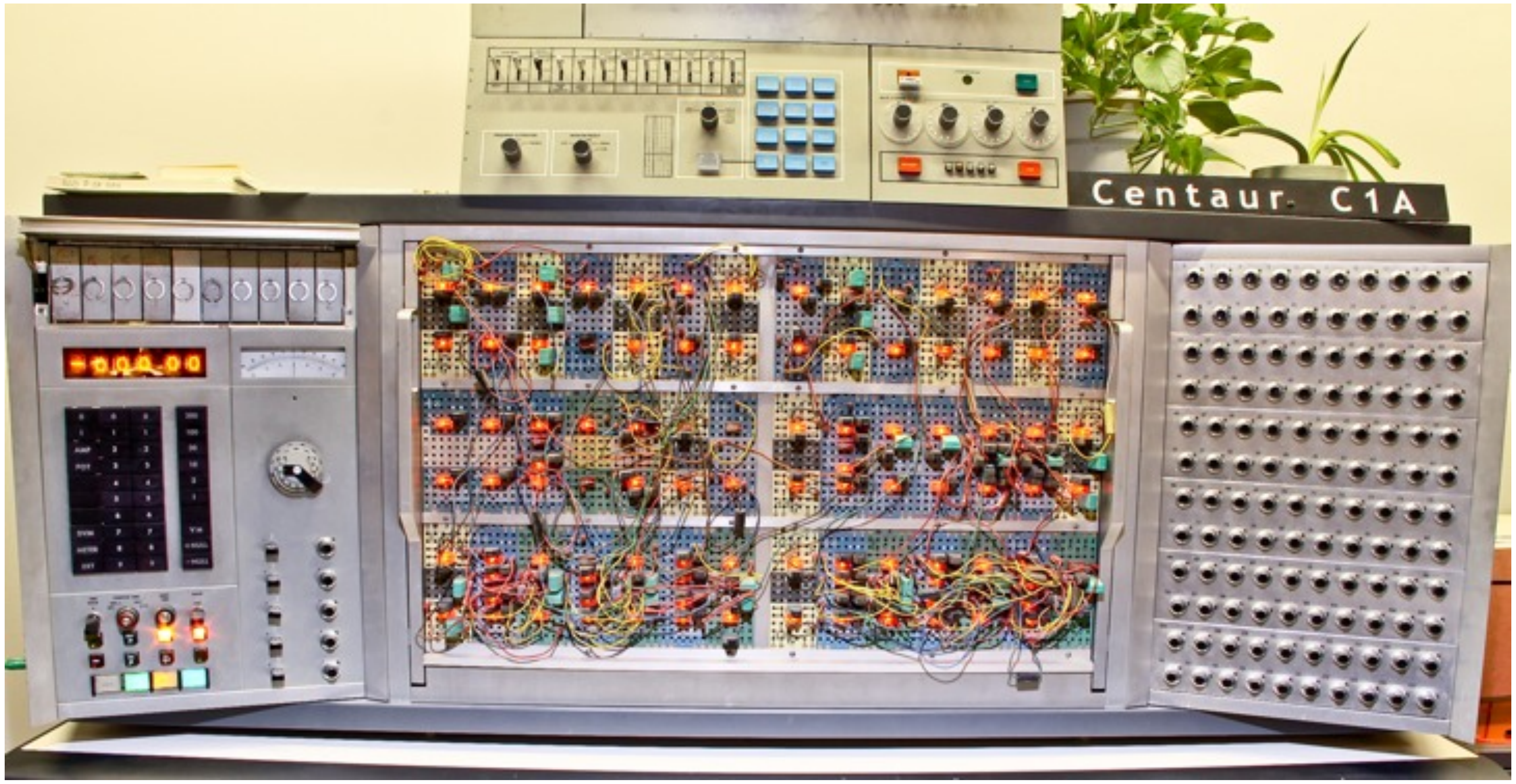
- Interesting observation (Mead, late 1980s) Transistors in the sub-threshold regime: current depends approximately exponentially on gate voltage. Similar current-voltage curve in ion channels (building blocks of cell membranes and neurons in the brain).
- Allows for a biology-inspired approach to computing, adaptive and able to learn
- *analog* VLSI

Neuromorphic engineering

- Achievements include:
- Silicon retina (Mahowald 1988)
- Silicon cochlea (Lyon and Mead 1988)
- Silicon neuron (Mahowald 1991)
- Silicon synapse (Dorio et al 1996)
- Cognitive systems (Indiveri et al. 2013-present)

- Applications include:
- Prosthetics
- Low power devices

More historical relics-electronic *analog* programmable computers



- used e.g. for research in Chaos theory in the 1970s, e.g. Rössler (1976), Shaw (1978).

Lorenz attractor

- Mathematical model for unpredictability and Chaos, e.g. in weather (1963)
  - originally run on digital computer
  - took all afternoon to get a few orbits!

analog was faster at the time

https://www.youtube.com/watch?v=mbJpRAVZZuU

digital now:

real-time simulation run on this laptop
50,000 points at once!
(courtesy Rob Shaw)

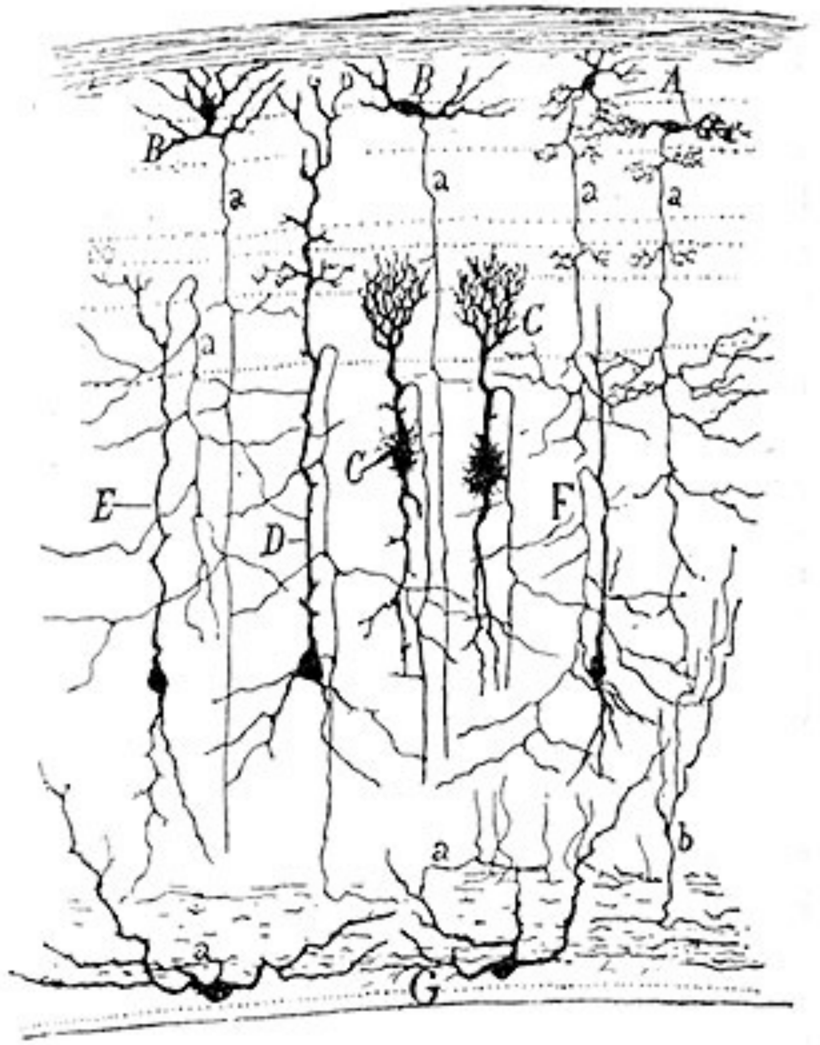
Back to the main track...

- **Origins of neural networks** come from mathematical descriptions of neurons (brain cells)
- *Physicists and mathematicians working together with neuroscientists*
- From neuron models to neural nets to machine learning...
  ...after (short) coffee break (10 min).

Break (10 min)

From brains to learning machines

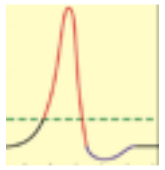
- ***Neurons:*** central nervous system (CNS) has ca.10^11. They are:
- separate entities (Ramon y Cajal, Nobel Price 1906)
- connected by synapses (Sherington and Adrian, Nobel Price, 1932)
- many different types of neurons with different functionality



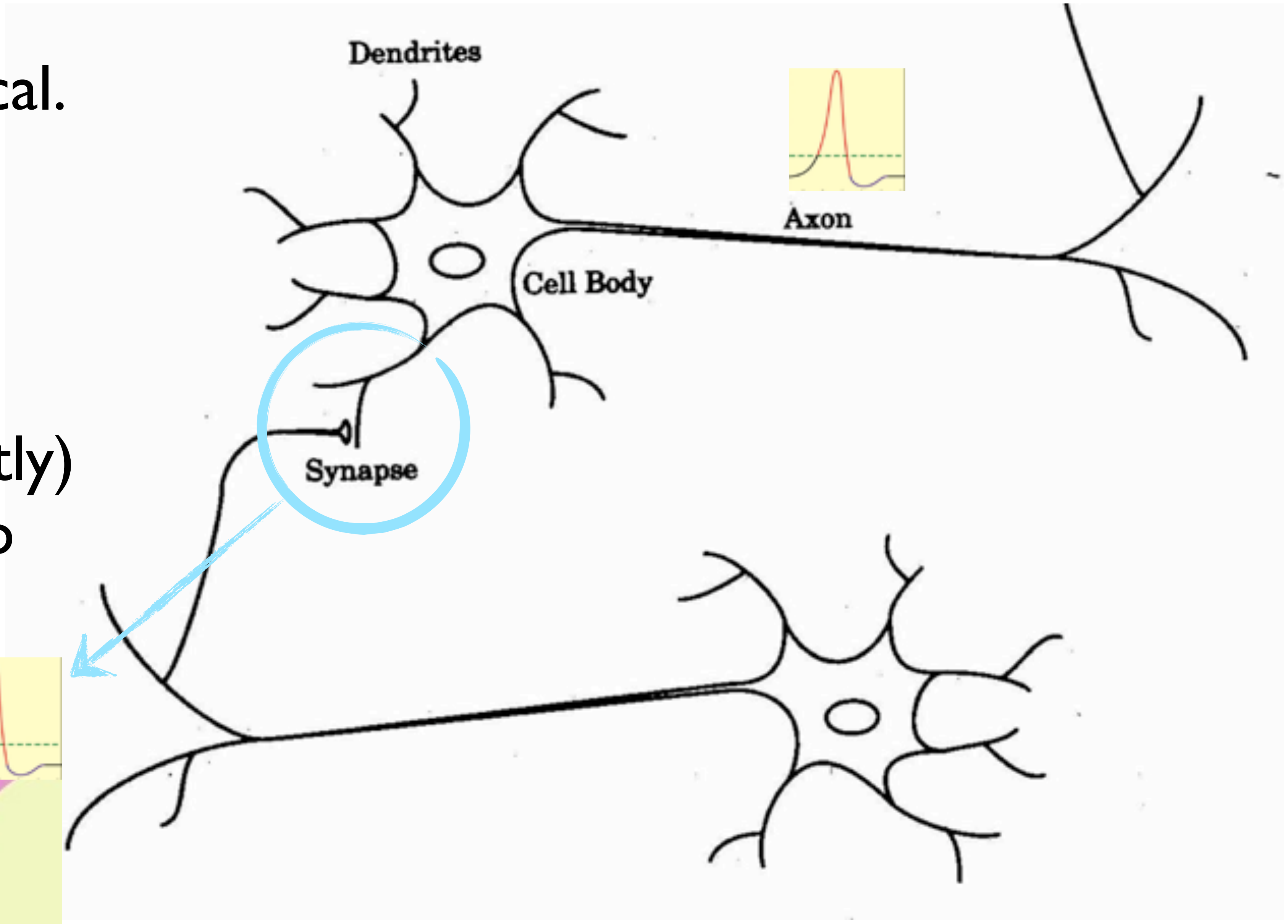
Optic tectum (sparrow), drawing by Cajal

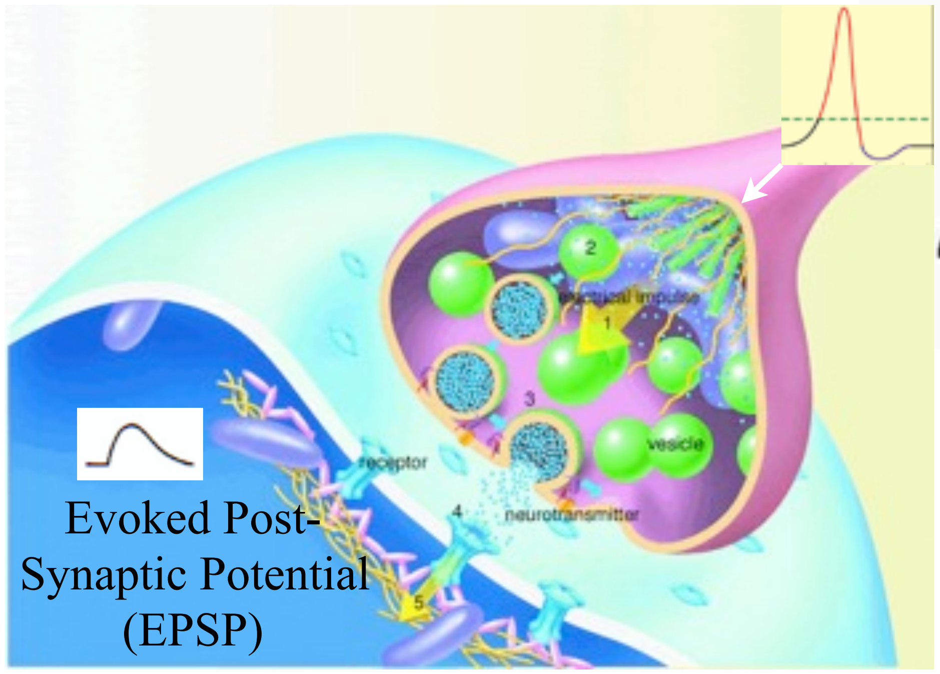
Neurons communicate

- Information processing within a neuron: electrical. Action potential ("spike").



- Information processing between neurons: (mostly) chemical (exception: gap junctions)

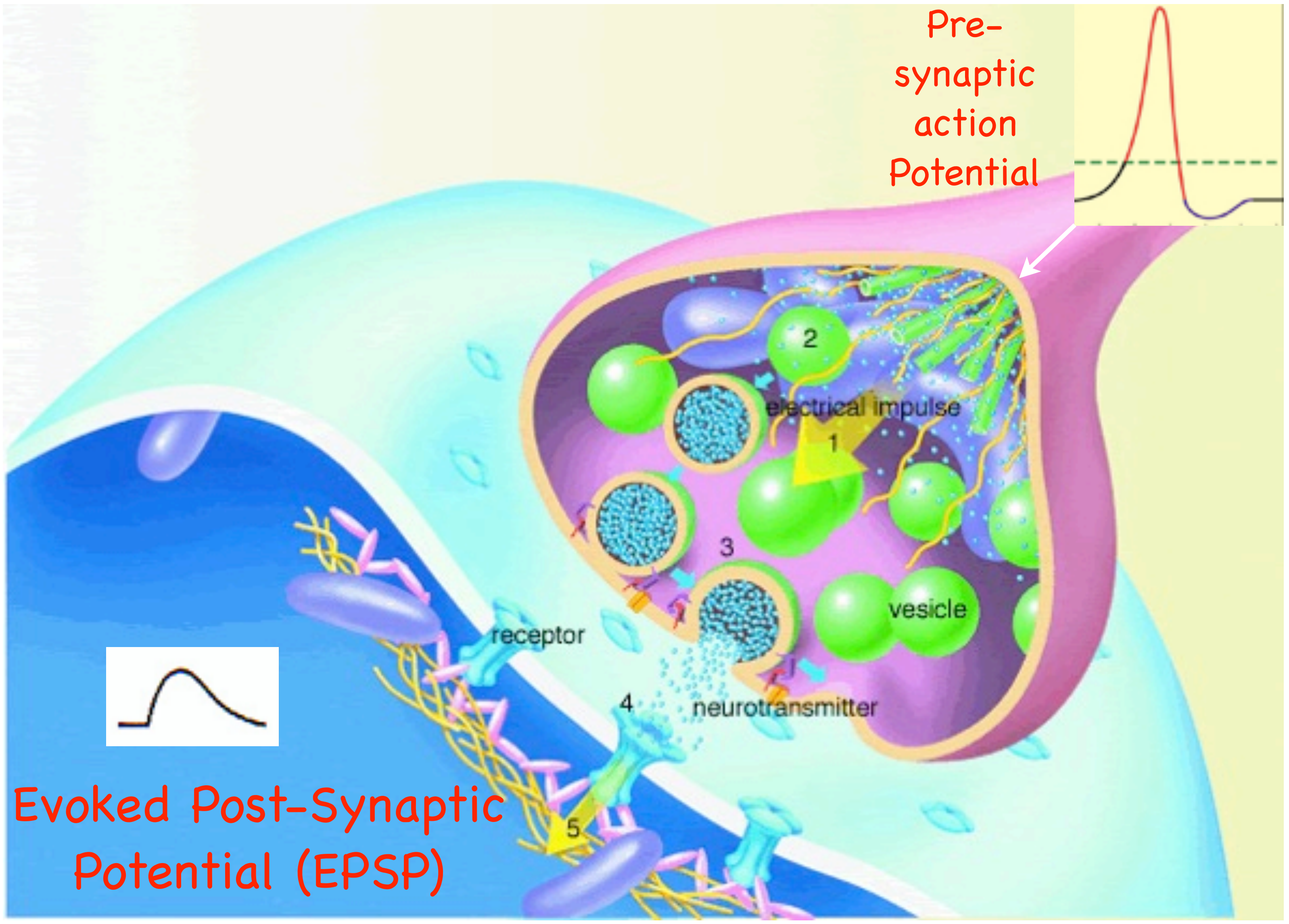




http://ffden-2.phys.uaf.edu/212_fall2003.web.dir/Keith_Palchikoff/biological%20neuron.JPG

http://www.brain.riken.go.jp/english/g_braaw/images/g5/synapse.gif

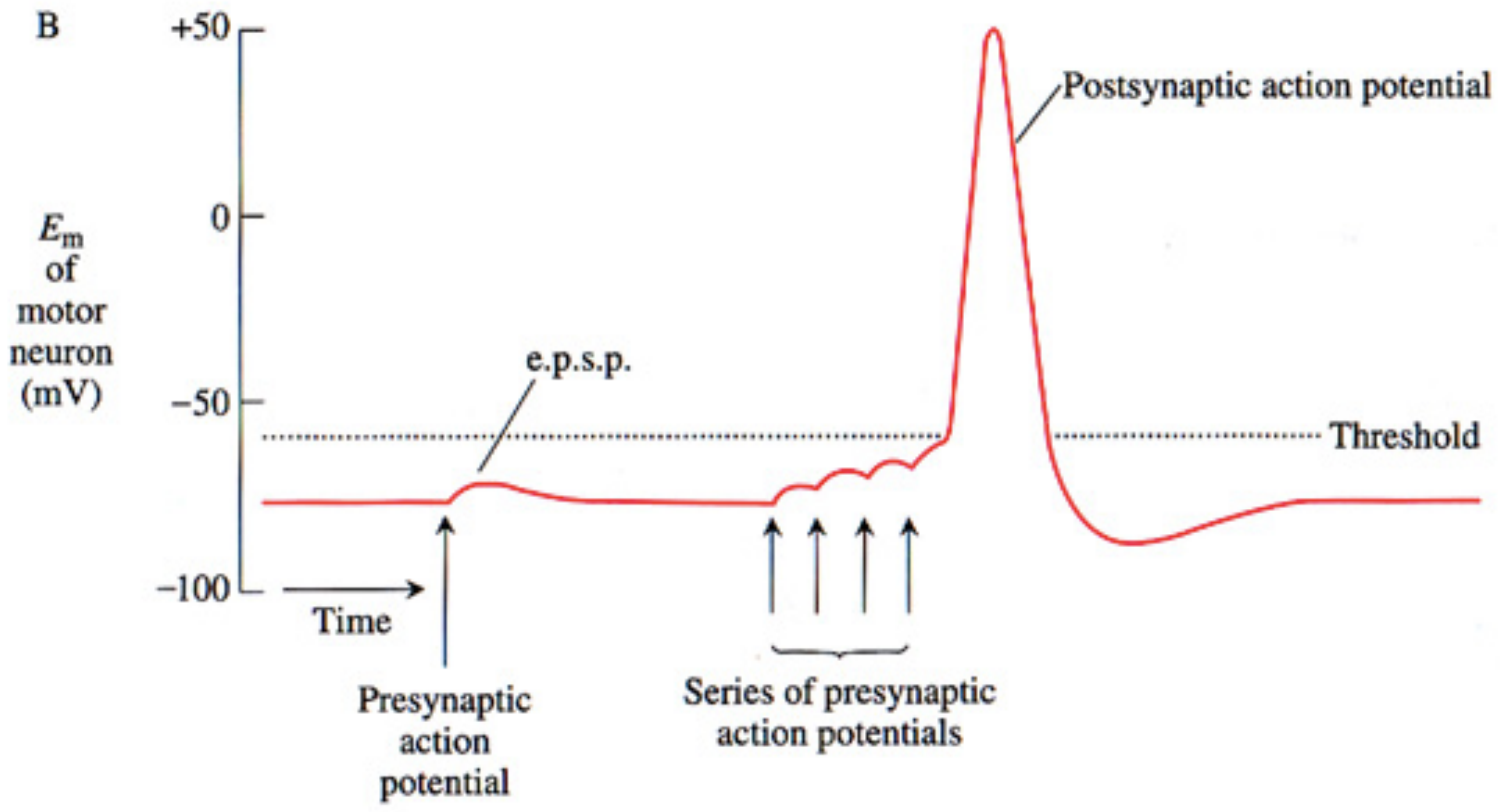
- Pre-synaptic action potential depolarizes synaptic terminal
- Voltage-sensitive calcium ion channels open => calcium enters
- Release of neurotransmitter; diffusion through synaptic cleft.
- Neurotransmitter binds to post-synaptic receptors
- Ions enter post-synaptic cell => EPSP


Pre-synaptic action Potential
2
electrical impulse
1
3
vesicle
receptor
4
neurotransmitter
5
Evoked Post-Synaptic Potential (EPSP)

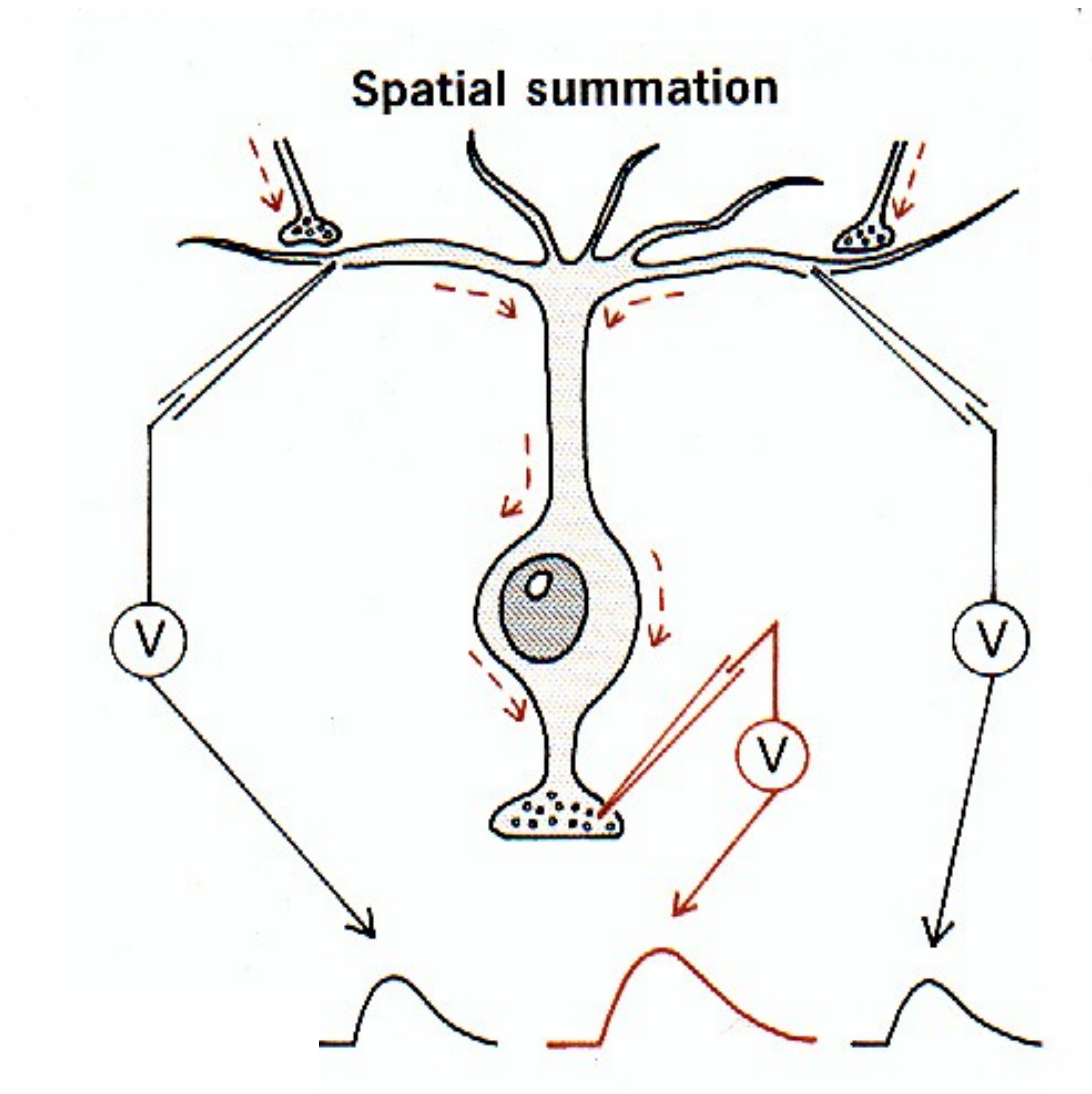
http://www.brain.riken.go.jp/english/g_braaw/images/g5/synapse.gif

(Temporal Integration of EPSPs)

(B) Responses of the postsynaptic motor neuron to action potentials in the presynaptic sensory neuron. At the upward arrows, action potentials are triggered in the presynaptic neuron by an electrical stimulus.

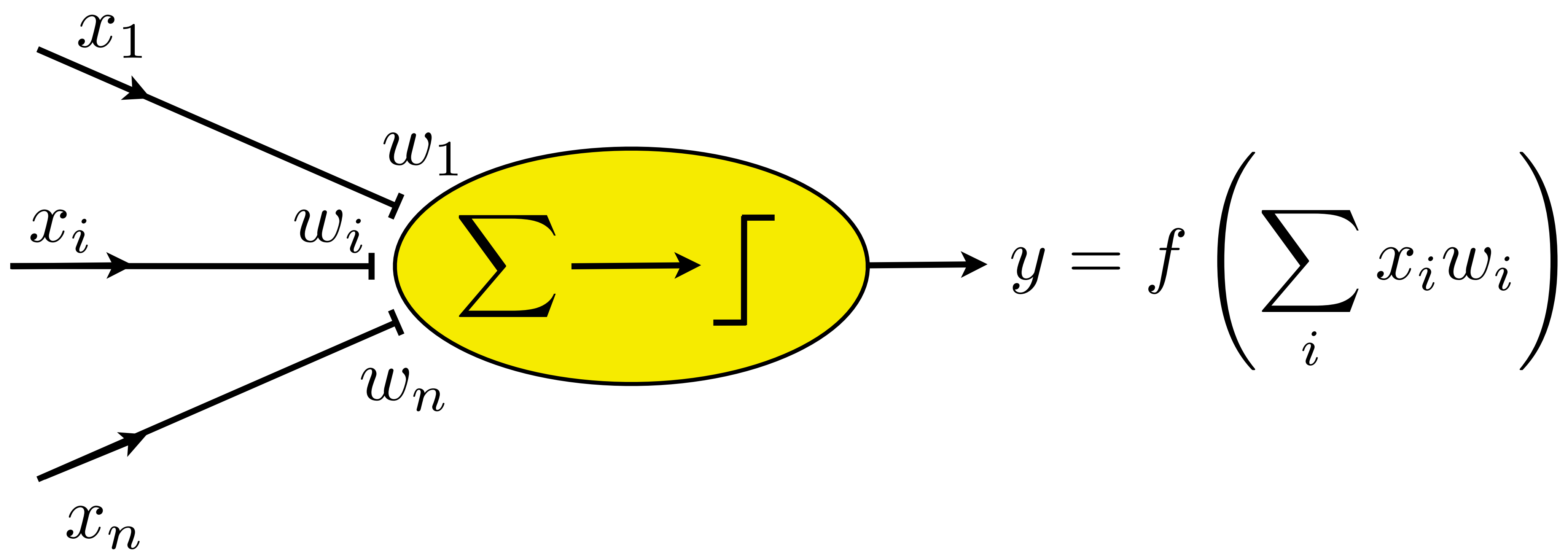


Spatial Integration of EPSPs



Mathematical simplification

- Neurons are either on or off: represented by binary value.

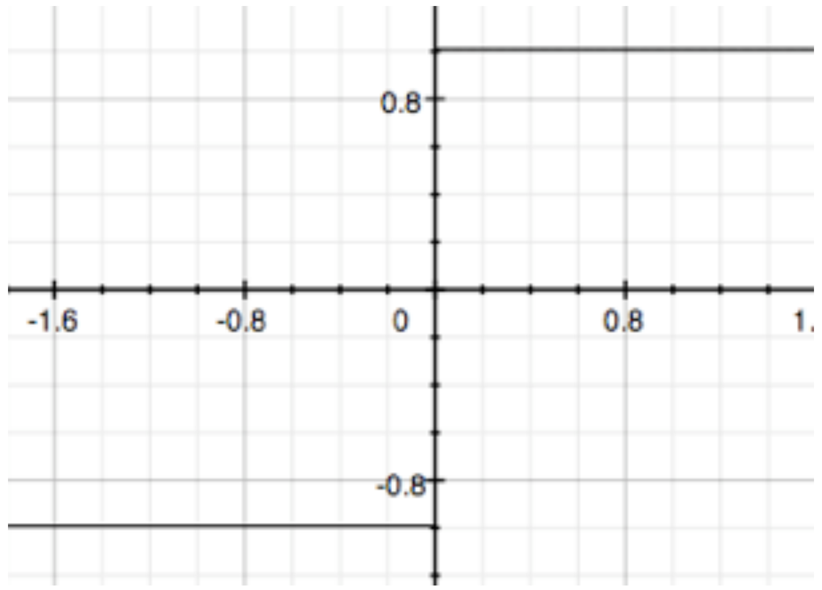


- Inputs form n neurons: x_i
- get multiplied by weights at "synapses": $x_i w_i$
- then added $\sum_i x_i w_i$
- if above threshold, then neuron is "on".

- Remark: add x_0 to the input vector, in order to write the bias as $b = x_0 w_0$ to get more compact form **wx**, rather than **wx** + b

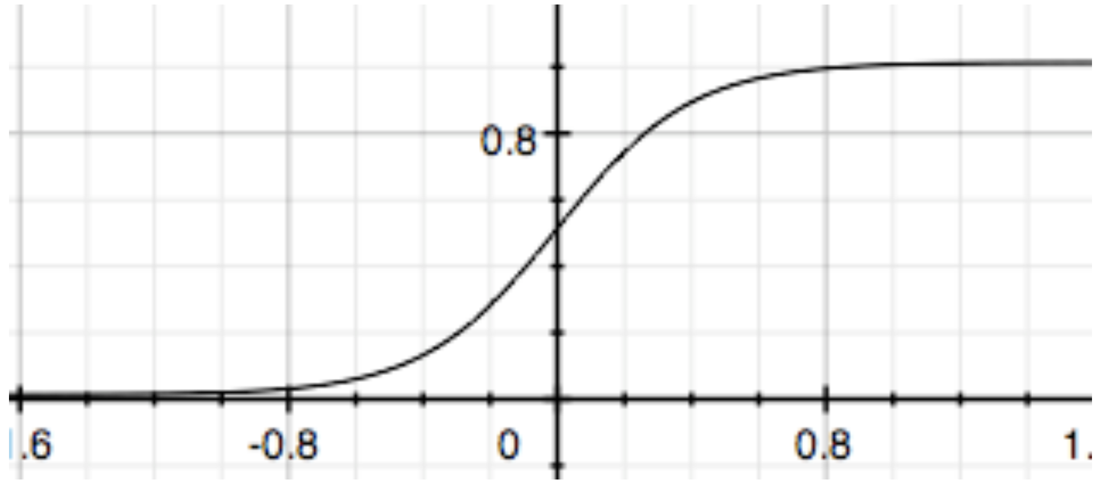
Transfer function

- Step function:



- Used by Pitts & McCulloch (1942), and in Rosenblatt's Perceptron (1957)

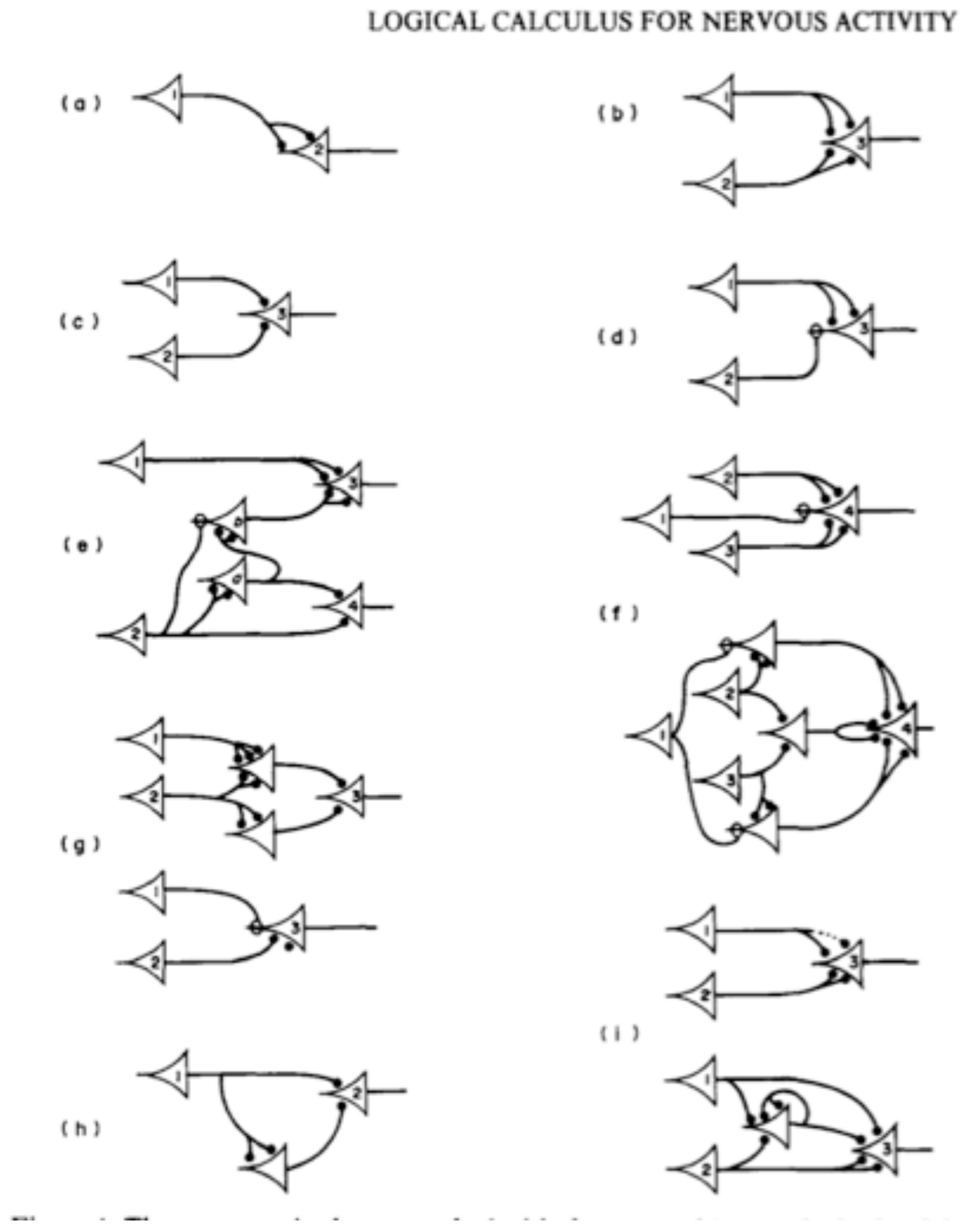
- Sigmoid (used in many "modern" feed forward neural nets):



- Sigmoidal function is differentiable (good for deriving gradient decent learning rule; Backpropagation algorithm, Werbos 1974)

Neural networks (1940s/50s)

- Pitts and McCulloch (1942/3): “Formal” (mathematical) neurons thought of as processing units
- *Networks* can emulate any logical function.
- D. Hebb: Connections between neurons can change. “Learning rule”: strengthen proportional to correlation between activity of pre- and post synaptic neuron.



Perceptron (Rosenblatt, 1957)

- Probably the first "learning machine". Artificial neuron that solved a classification task (supervised learning):

- Given: N input vectors **x** together with labels l (this is the "teaching" signal).

- Goal: for any given input, the output of the classifier should be the same as the label.

- Assume that the labels are binary, either -1 or +1.

➡ *Does binary classification*.

Perceptron learning

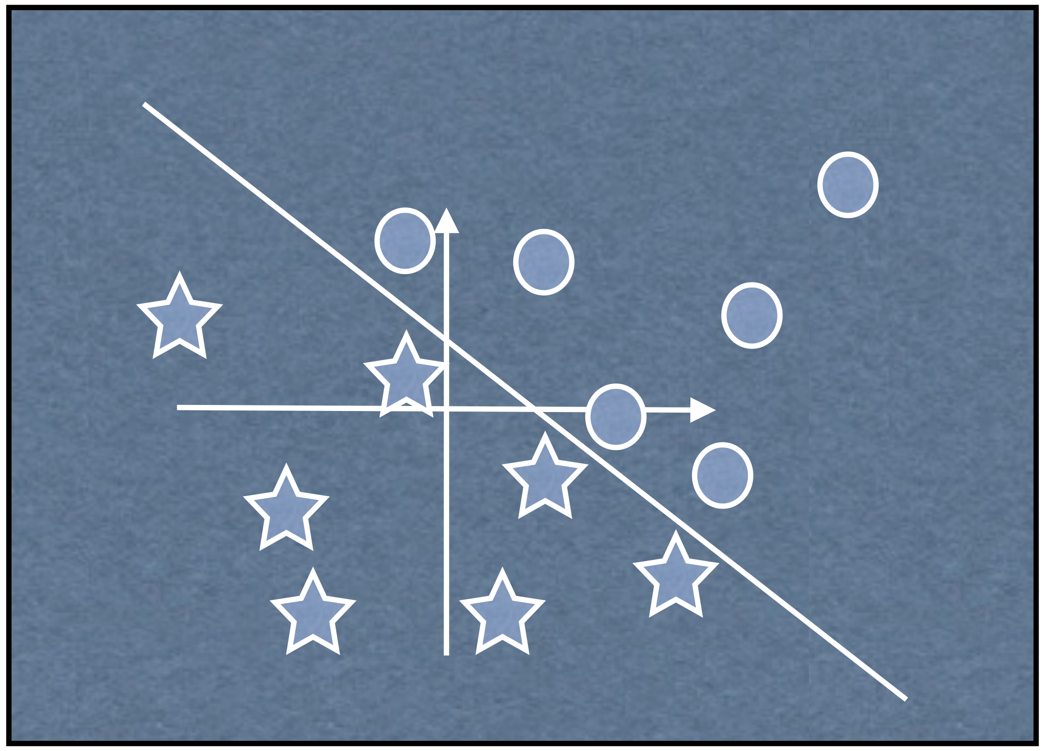
- *adjust weights according to the correctness of the output* until all input data in training set are classified correctly.
- **error measure**: compare output of the neuron (y) to desired output (= label, l): both are either -1 or 1, so:
    - y*l = 1: correct classification, then: $\boxed{l - y = 0}$
    - y*l = -1: incorrect classification, then: $\boxed{l - y = 2l}$
- adjust weight vector **w** *for each misclassified* input **x**, by adding l***x**. This turns **w** towards/away from **x** if l=1/-1
- can do $\boxed{\mathbf{w} \longleftarrow \mathbf{w} + c(l-y)\mathbf{x}}$ for all training examples **x**
- c: step size parameter called “learning rate”; $0 < c < 1$

Representational power

- the decision boundary of the perceptron is a line:

  wx + b

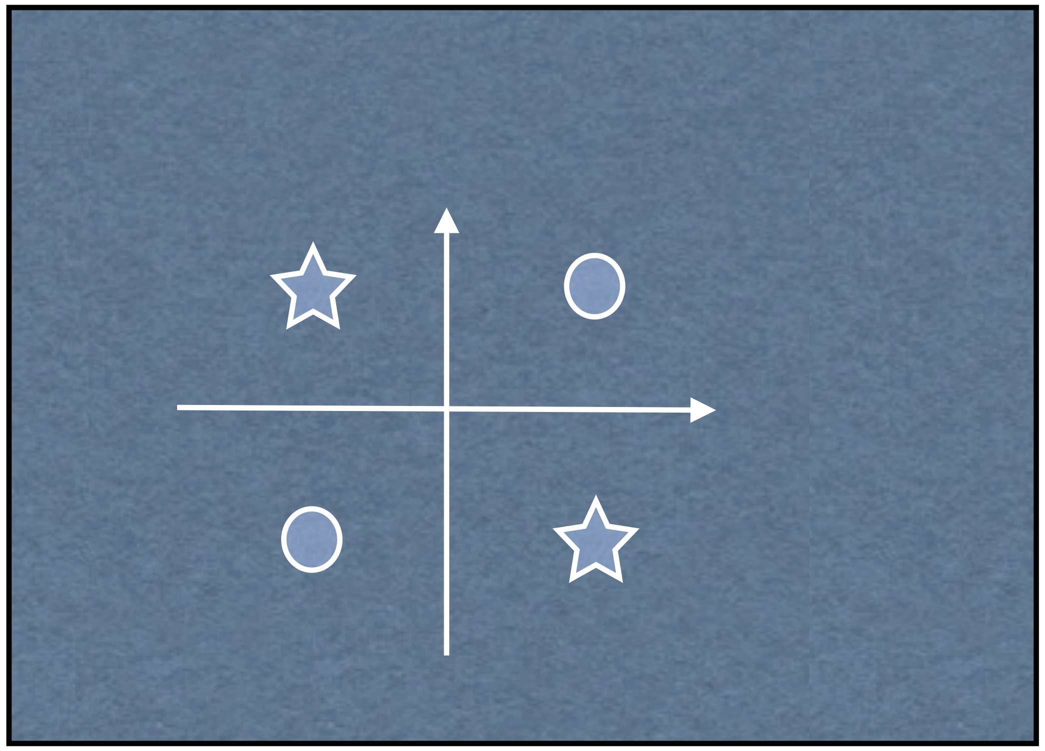
- the perceptron learning algorithm *converges* if input data are **linearly separable.**



- Novikoff (1962): Perceptron Convergence Theorem; first *margin-based error bound* (in a way the "dawn" of statistical learning theory)

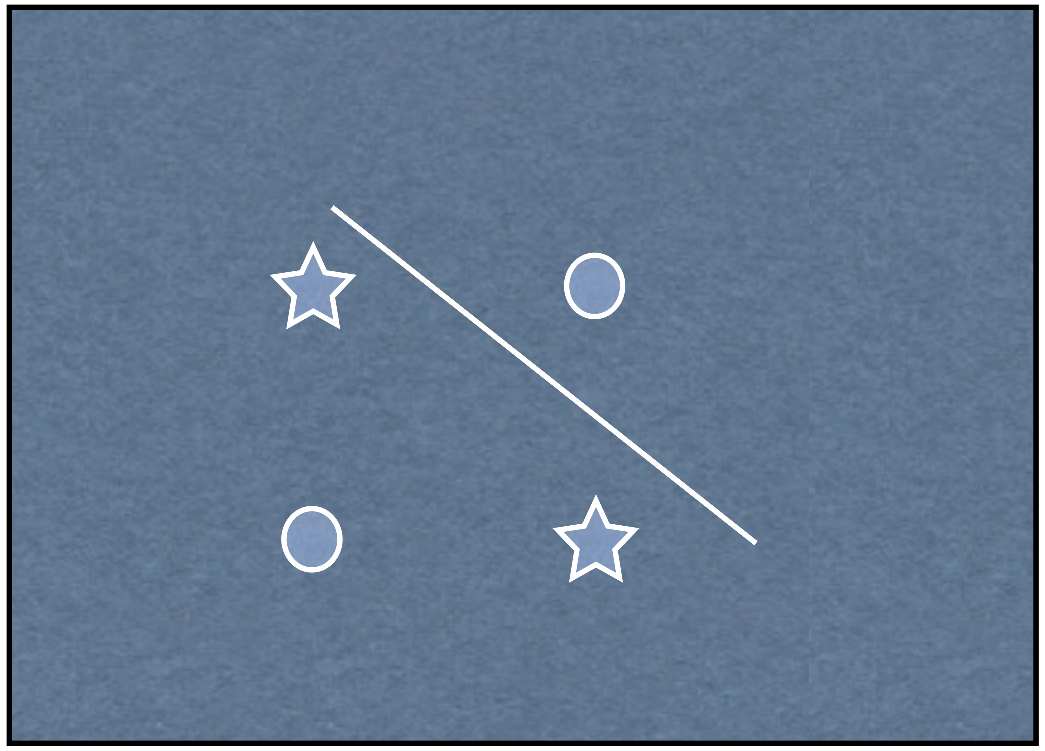
XOR problem

- Perceptron algorithm can not classify input data which can not be separated by a line.
- For example XOR



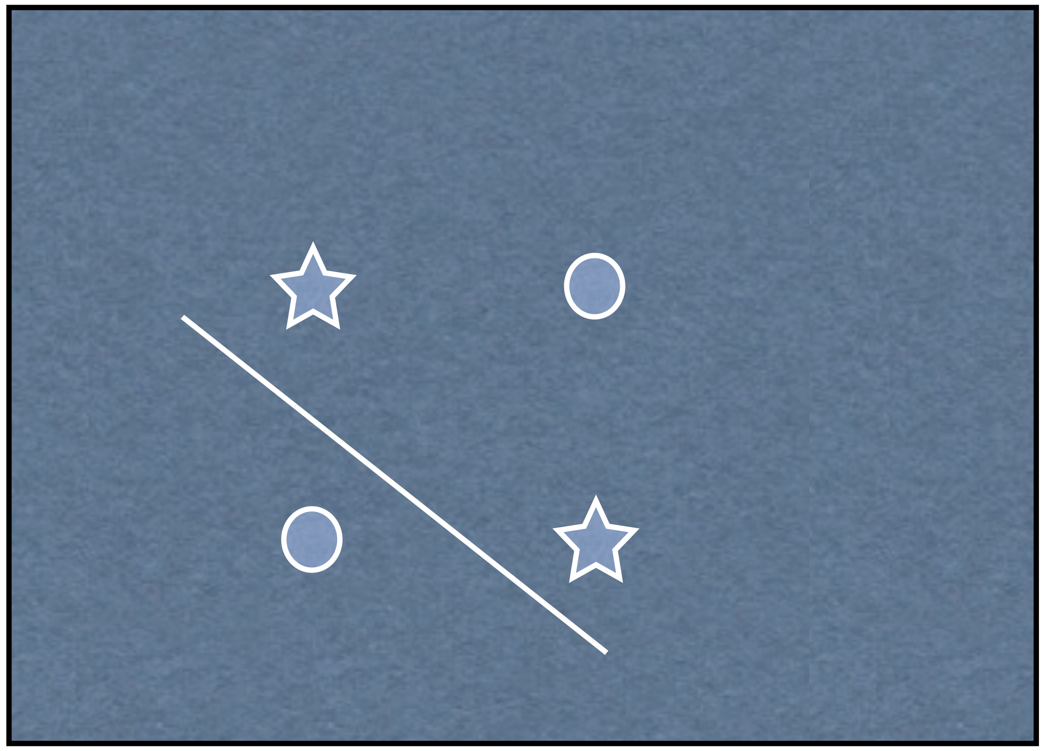
XOR problem

- Perceptron algorithm can not classify input data which can not be separated by a line.
- For example XOR



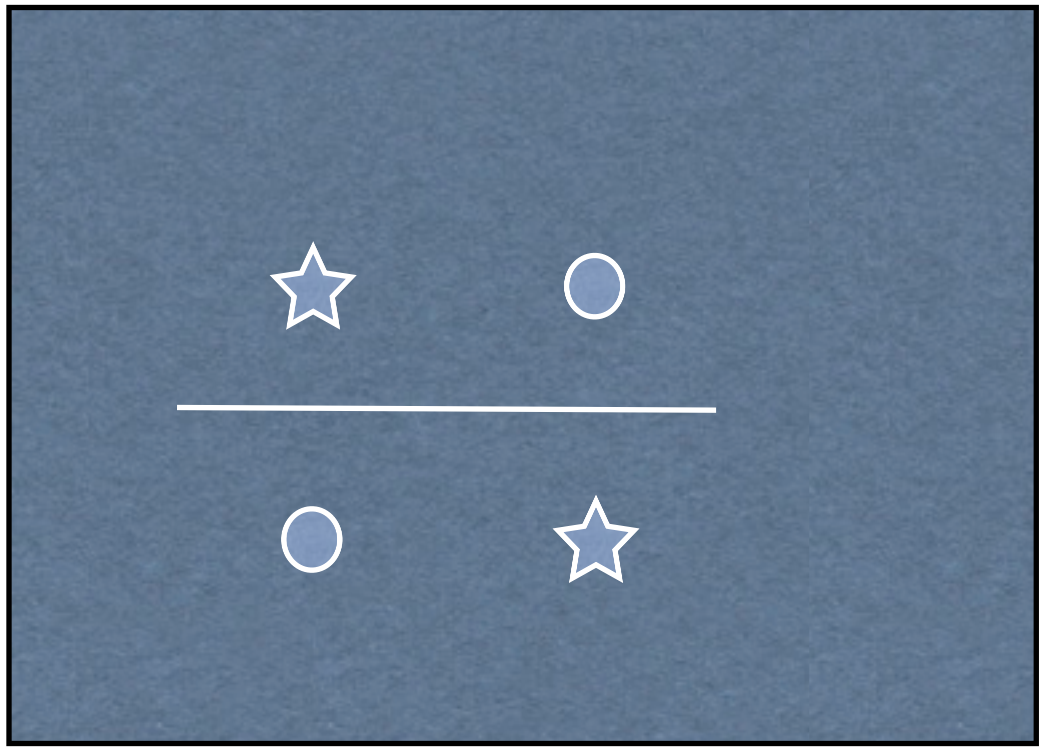
XOR problem

- Perceptron algorithm can not classify input data which can not be separated by a line.
- For example XOR



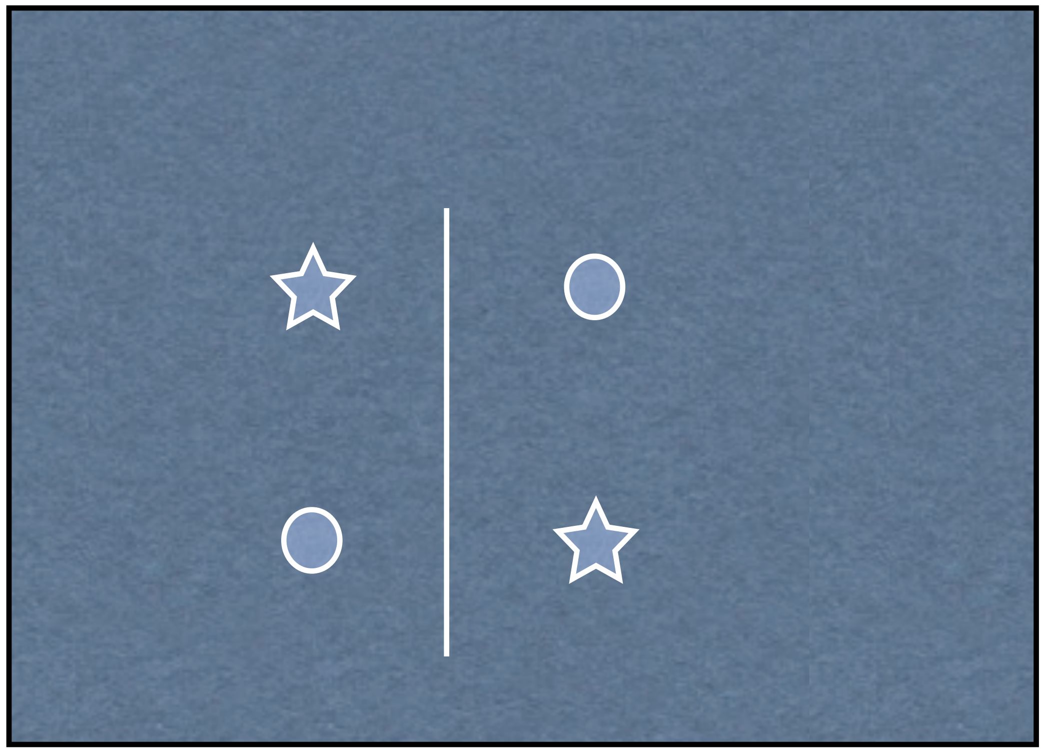
XOR problem

- Perceptron algorithm can not classify input data which can not be separated by a line.
- For example XOR



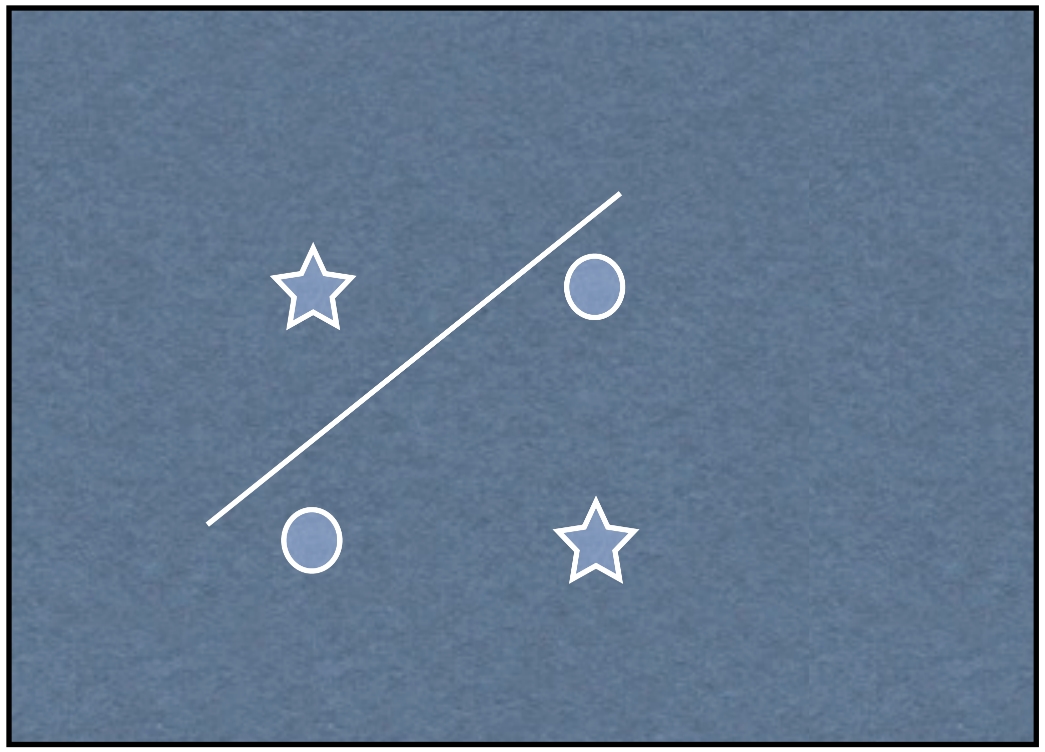
XOR problem

- Perceptron algorithm can not classify input data which can not be separated by a line.
- For example XOR



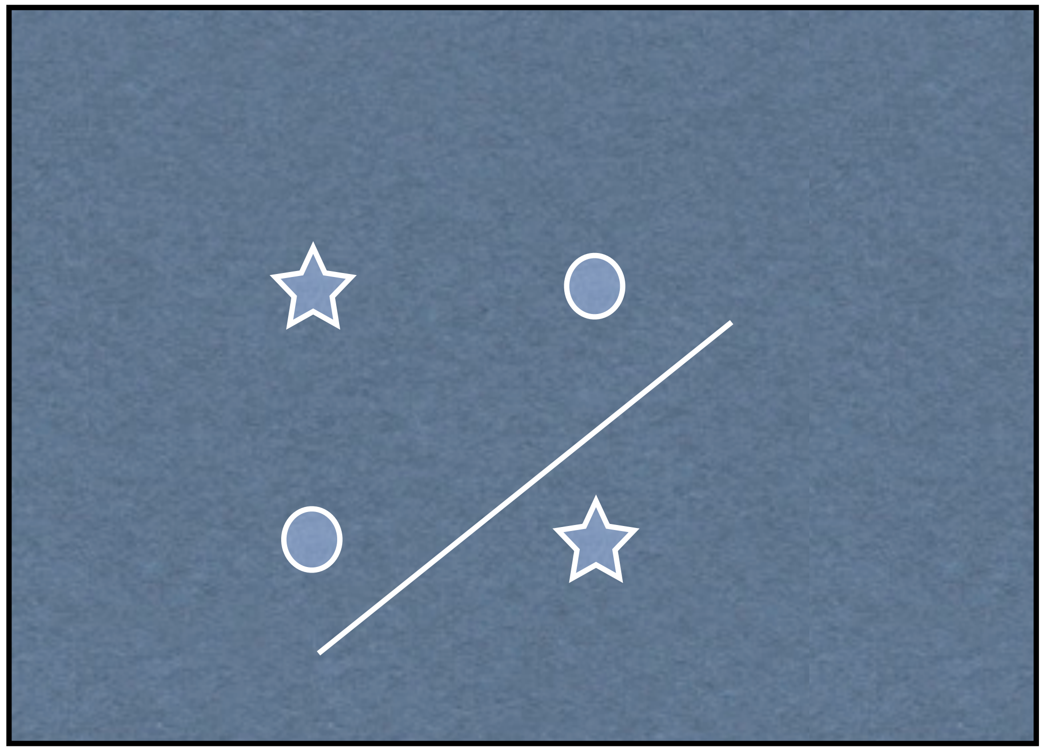
XOR problem

- Perceptron algorithm can not classify input data which can not be separated by a line.
- For example XOR



XOR problem

- Perceptron algorithm can not classify input data which can not be separated by a line.
- For example XOR



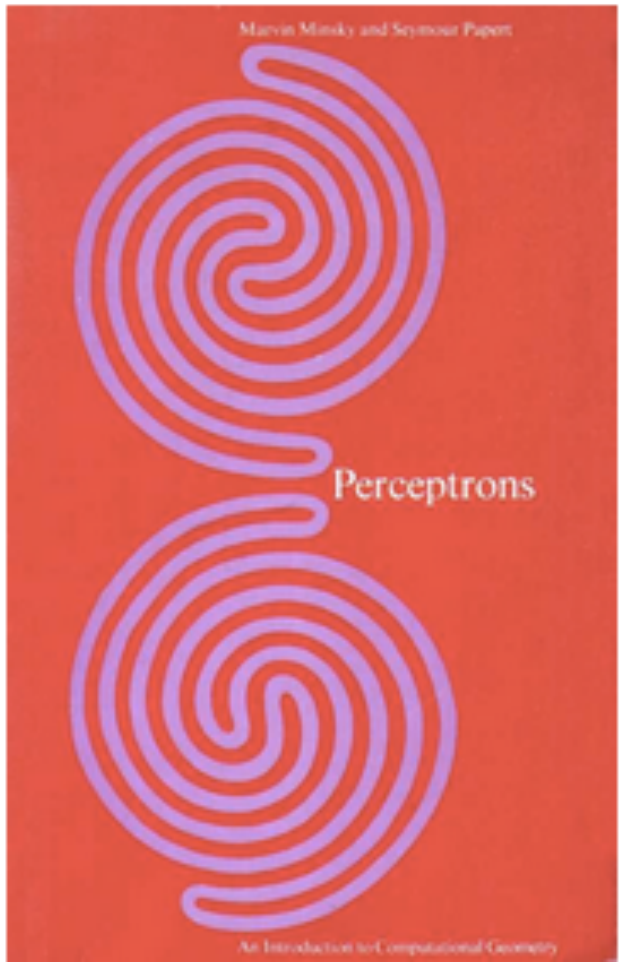
- A ***single artificial neuron*** can express the boolean functions AND, OR, and NOT, **but not XOR**.

Minsky and Papert (1969)

- Shortcomings of perceptron were pointed out in a fierce manner by proponents of **symbolic AI** in what became a very influential book: "Perceptrons".

- Their intuition was that ***neural nets would not be useful in practical applications.***

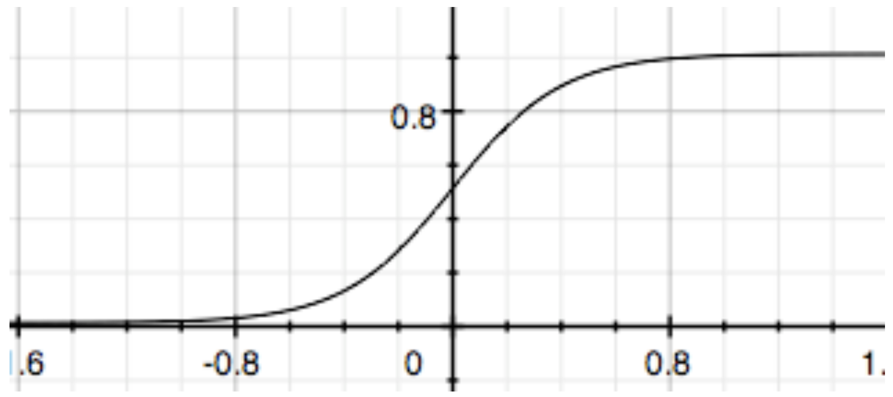
- This put a stop to neural network funding, and research was significantly diminished.


Marvin Minsky and Seymour Papert
Perceptrons
An Introduction to Computational Geometry

Come-back of neural nets and beginnings of machine learning

- Backpropagation algorithm (Werbos, 1974).
- Neocognitron (Fukushima, late1970s).
- Backprop re-discovered, used to train multi-layer networks in the 1980s: first big successes in pattern recognition.
  - ➡ Training convolutional NNs, eventually "deep learning".
- 1960s-1980s: Vapnik with Chervonenkis and others looked carefully at the classification problem and developed statistical learning theory, addressing the problem of *model complexity control.* Foundation for:
  - ➡ Support Vector machines (SVM, 1963)
  - ➡ Kernel machines (1990s)

Backpropagation



- Use differentiable transfer function
- Gradient descent over all units in network: use chain rule to compute gradient.
    - Forward pass: compute outputs of all units starting from input layer, ending with output layer
    - Backward pass: compute gradient starting from output layer.
    - Update the weights.
- Stochastic gradient descent - helps with computational cost and thus speed.

Kunihiko Fukushima

- Built the first electronic retina out of discrete components (1970, with Y. Yamaguchi, M. Yasuda, S. Nagata)
- Neocognitron, the “grandfather” of convolutional neural nets (1979/80)

Neocognitron

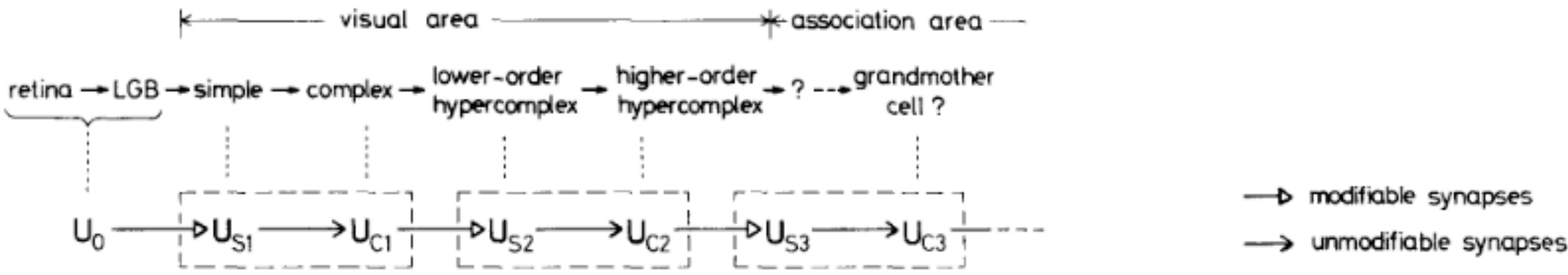

visual area
association area
retina → LGB → simple → complex → lower-order hypercomplex → higher-order hypercomplex → ? --→ grandmother cell ?
U_0 U_{S1} U_{C1} U_{S2} U_{C2} U_{S3} U_{C3}
modifiable synapses
unmodifiable synapses

Fig. 1. Correspondence between the hierarchy model by Hubel and Wiesel, and the neural network of the neocognitron

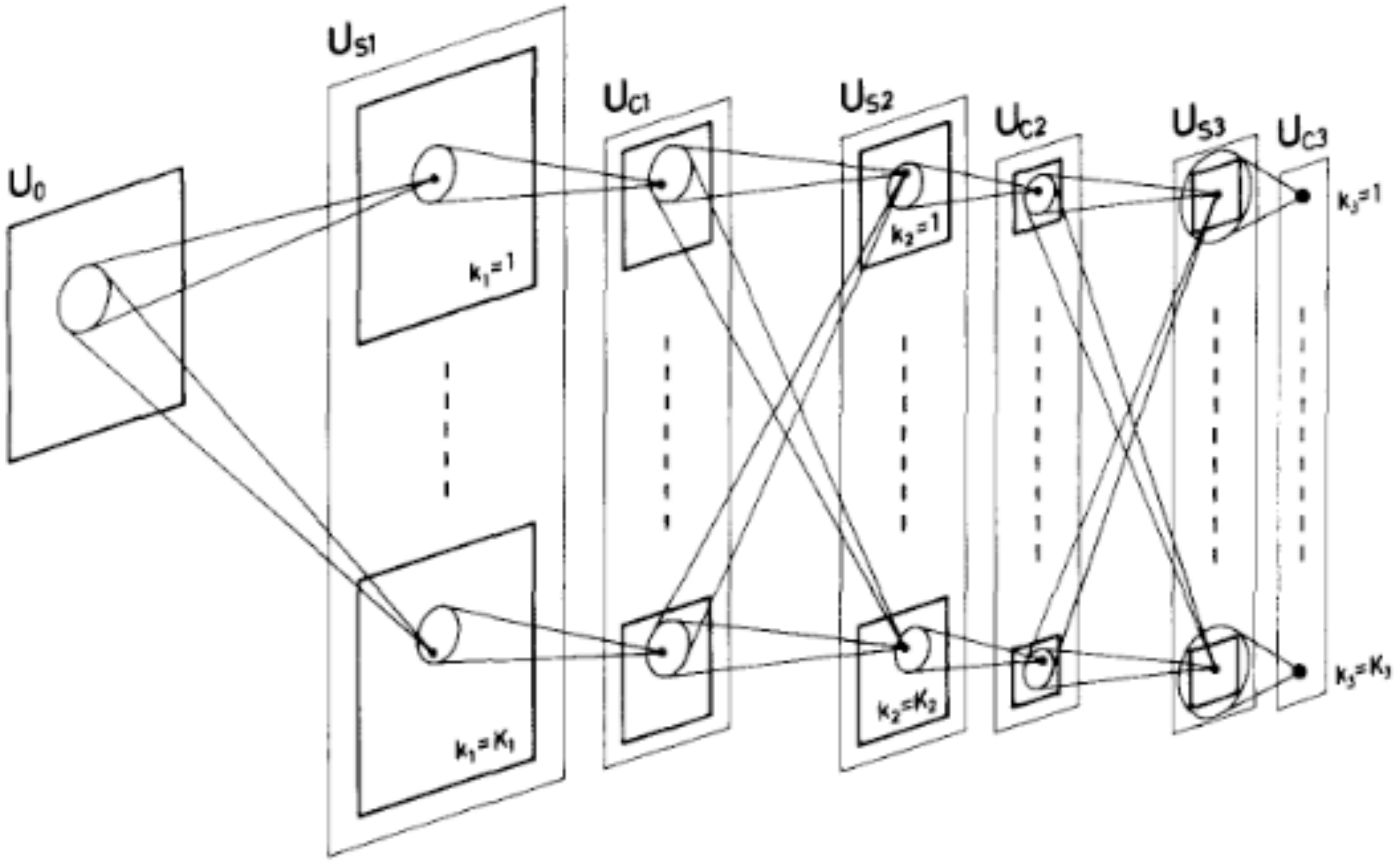


Fig. 2. Schematic diagram illustrating the interconnections between layers in the neocognitron

ca.1980s-mid 90s: Neural nets

- Recurrent NNs as Ising models (Hopfield)
- Boltzmann machine (Hinton, Sejnowski)
- Convolutional network trained by backprop alg. for *hand written digit recognition* (LeCun and others).

1990s: SVMs, Kernels and Bayes

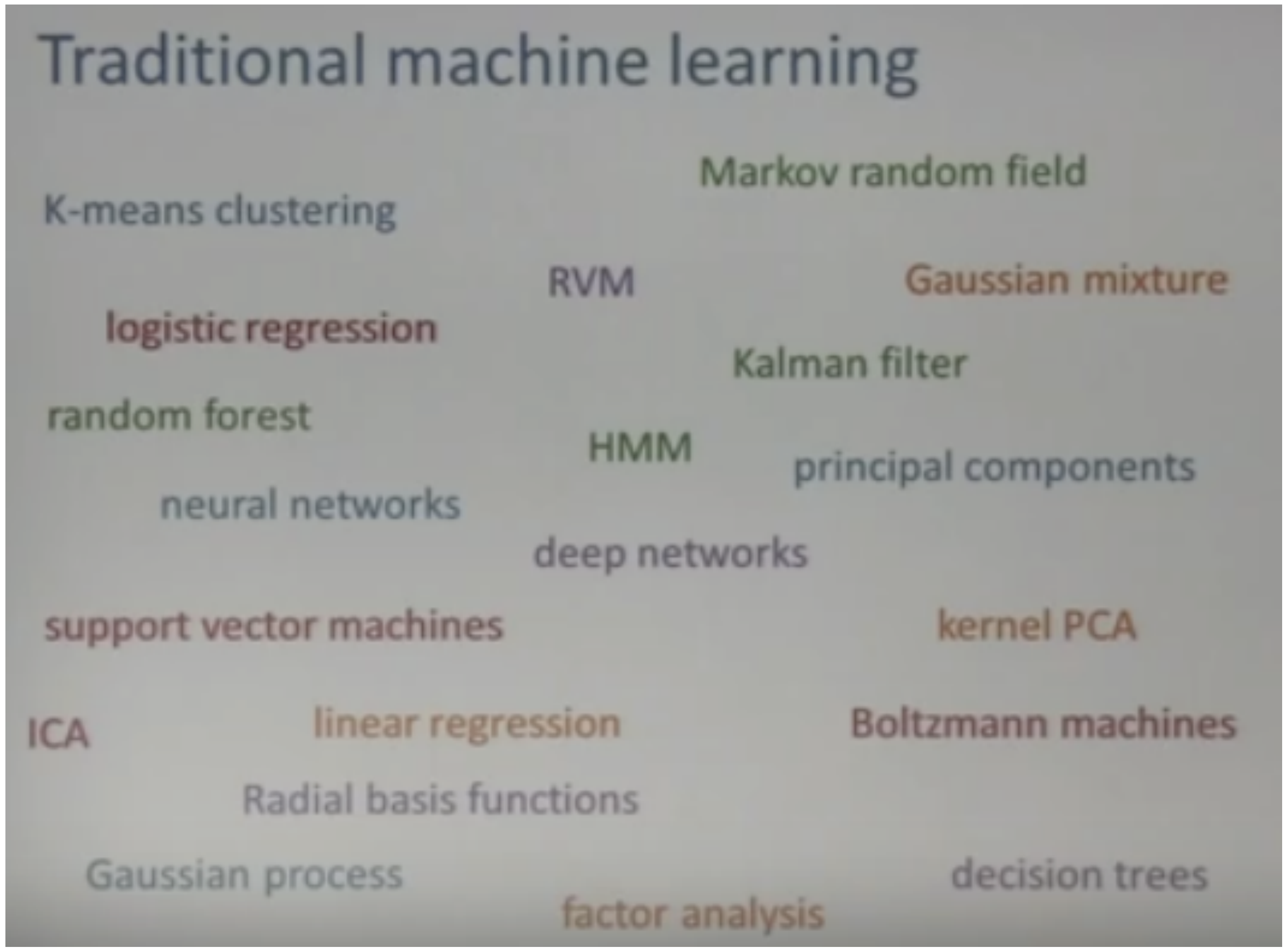
- Support vector machine alg. (Vapnik&Chervonenkis 1963)
- Nonlinear classifiers via Kernel trick (Vladimir Vapnik with Isabelle Guyon, Bernard Moser, 1992)
- Soft margin SVM implements *empirical risk minimization* (Vapnik and Corinna Cortes, 1995)
- More Kernel machines (Kernel PCA etc., Bernhard Schoelkopf, A. Smolla, and many others)
- Bayesian Inference and probabilistic foundations of ML (MacKay, Neal, Hinton, Bishop, Jordan,...)
- Increasing amount of information theory applied to machine learning, and to neuroscience (Bialek and others)

New Millenium

- Proliferation of ML methods!
- Many new application areas
- “Big data”
- NNs big comeback in “deep learning”
- New hardware:
  - GPUs
  - Quantum computers... ?!

ML techniques

- This is just a selection...
- NIPS conference proceedings (Neural Information Processing Systems)
- Journal of Machine learning research
- Other conferences, such as UAI (Uncertainty in Artificial Intelligence), ICML(Int. Conf. Machine Learning)
- Obscure "old" books



Traditional machine learning

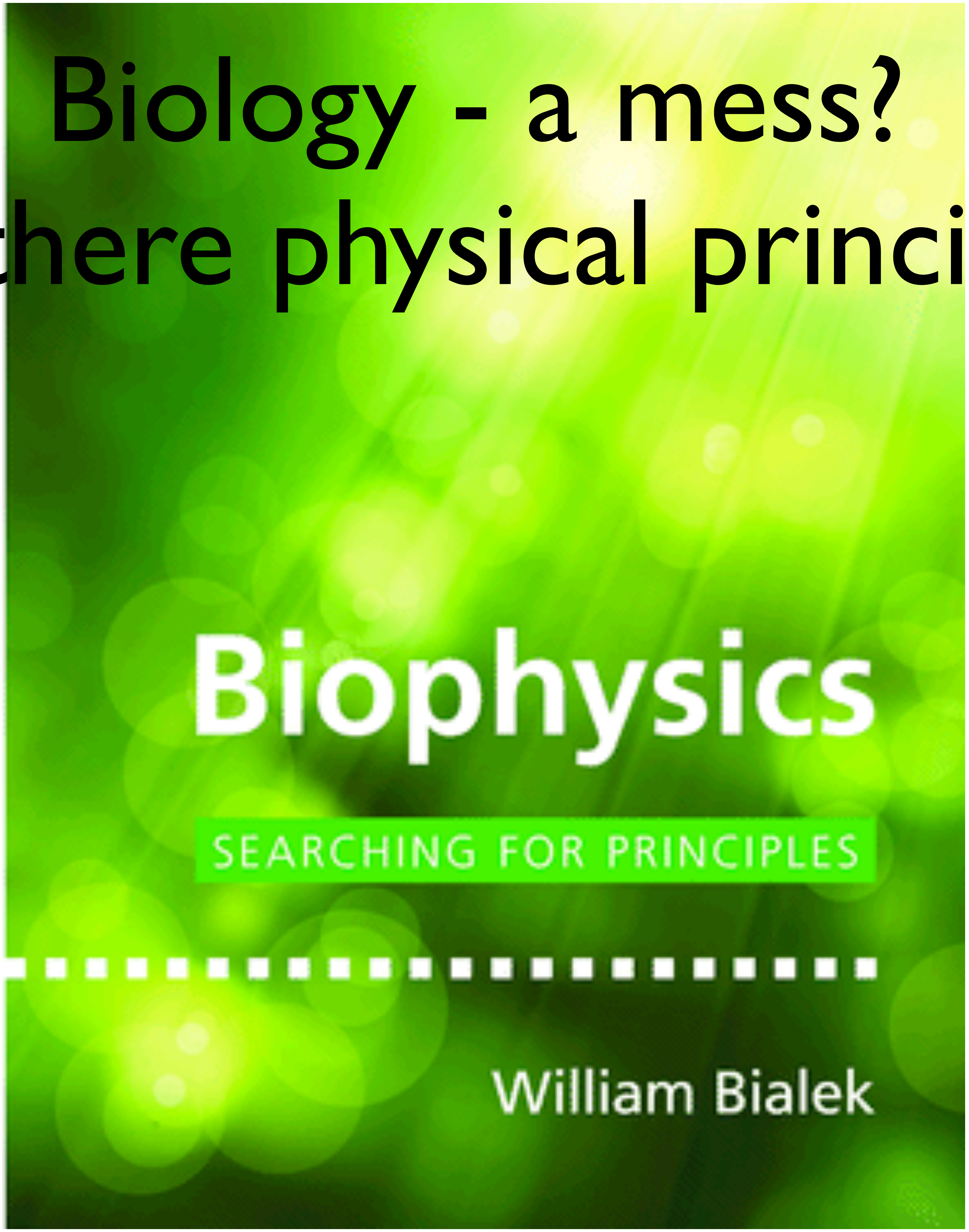
Markov random field
K-means clustering
RVM
Gaussian mixture
logistic regression
Kalman filter
random forest
HMM
principal components
neural networks
deep networks
support vector machines
kernel PCA
ICA
linear regression
Boltzmann machines
Radial basis functions
Gaussian process
decision trees
factor analysis

Image stolen from Chris Bishop's lecture at MLSS 2013, MPI Tuebingen

Ack! Too much “stuff”

- Are there some simple principles?
- Is there physics behind all of this?
    - Presumably living systems *are physical*, so then, can we have physical principles guide us in the jungle of data processing and learning algorithms?
- Study physical limits of information processing...
- Hope to get to building principles!

Biology - a mess?
Are there physical principles?



Biophysics

SEARCHING FOR PRINCIPLES

William Bialek

Machine learning for physicists!

✓ Overview of some history and core ideas

● Physical limits to information processing

   2. Equilibrium thermodynamics applied to information processing

   3. Introduction to information processing far from thermodynamic equilibrium, *selected topics*.

4. From physical limits to information theory...

5. ...to unsupervised learning...

6. ...and cluster analysis.

7. Supervised learning: neural nets

8. Introduction to main ideas in statistical learning theory; Support vector machines, Kernel trick.

9. Bayesian Inference

10. Brief overview of selected "fashionable" ML techniques.

Homework

- Implement perceptron learning algorithm: N Inputs $\mathbf{x}_j$ & labels l_j
  - Initialize weight vector $\mathbf{w}$
  - While there exist misclassified examples:
    - Compute output $y_j = \theta(\mathbf{w}\mathbf{x}_j)$
    - For each example, update the weights: $\mathbf{w} \mathrel{+}= c(l_j - y_j)\mathbf{x}_j$
- Play around with the parameter (learning rate) and the input data, and verify for yourself what the Perceptron can and can not do
  - Make a movie of Perceptron converging, and one of Perceptron failing on the XOR.
- What else do you notice?
  - Is every solution the same? If not, are some "better" than others in some sense?