

Connections and symbols

AT1

Emmanuel Dupoux

- bref historique du connexionnisme et de l'IA symbolique
- les années 80: la confrontation ideologique: faux débat versus vraies questions
- Les nouvelles pistes
 - compositionnalité et produit tensoriel
 - récursivité et systeme dynamique
 - (gradualité et modèles bayesiens)

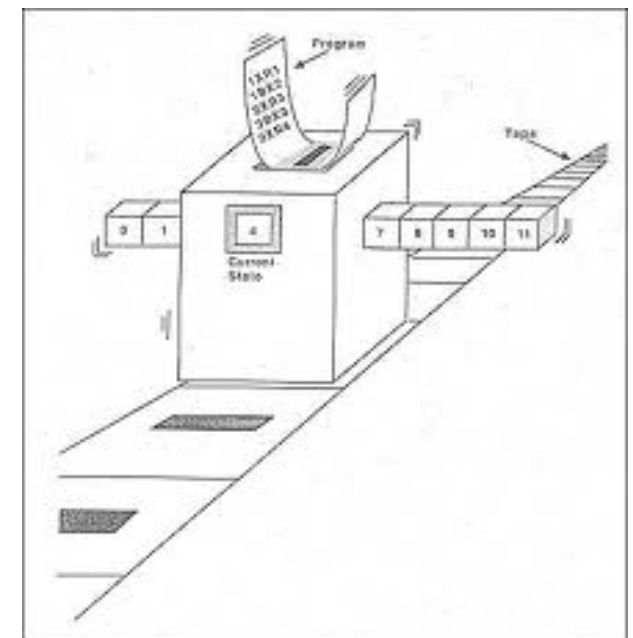
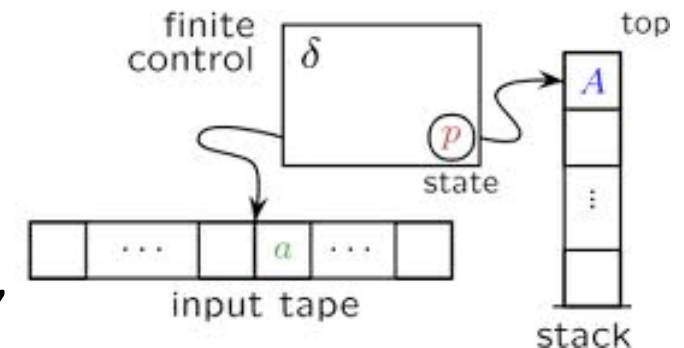
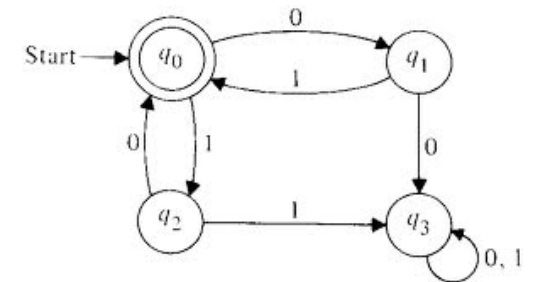
Symbols

Cognition and symbols

- reasoning = computation in a formal calculus
 - Euclides, Leibniz (universal language + reasoning calculus), Boole (formal system for logical and set theoretic reasoning), Frege (logisict programme for mathematics), Peano, Russell, etc
- Formal system:
 - A finite set of symbols (i.e. the [alphabet](#)), that can be used for constructing formulas (i.e. finite strings of symbols).
 - A [grammar](#), which tells how [well-formed formulas](#) (abbreviated *wff*) are constructed out of the symbols in the alphabet. It is usually required that there be a decision procedure for deciding whether a formula is well formed or not.
 - A set of axioms or [axiom schemata](#): each axiom must be a wff.
 - A set of [inference rules](#) (going from wff to wff)
- Examples
 - first order propositional logic (A, B, C, \rightarrow)
 - second order logic ($A, B, C, \forall \alpha, \rightarrow$)
 - predicate logic ($P(x)$)

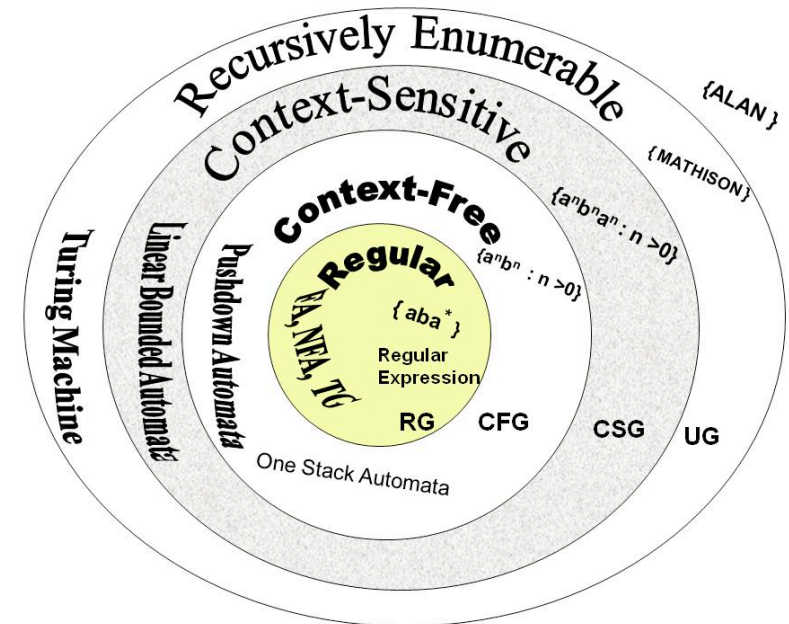
Cognition and symbols (II)

- Thinking=computation over symbols
 - computability theory (1930-) Godel, Post, Kleene, Church, Turing, Markov
- Physical symbolic systems
 - Finite state automaton, pushdown automaton, Turing machines computational power
 - Power of automata:
 - Finite state automata
 - Pushdown automata
 - Turing machines with one tape, Turing machines with several tapes, lambda calculus (eg lisp), string rewriting system (eg production), recursive functions
 - Church thesis:
 - Turing machines are on the top of the hierarchy



Cognition and symbols (III)

- Language
 - Miller & Chomsky (1963)
 - to each automaton, a class of languages
 - human languages are between context free and recursively enumerable
 - Symbolic AI
 - prolog, expert systems
 - SOAR, ACT*



symbolic processing

- Newell (1980) articulated the role of the mathematical theory of *symbolic* processing.
 - Cognition involves the manipulation of symbols – analogous to words, concepts, schema, etc.
 - What are symbols?
 - Definition is hard to pin down.
 - Roughly, it's like the values of a categorical variable (male, female, red, blue, dog, cat).
 - Operators on those symbols would then be things like “is-a” “a-kind-of” “purpose” “shape” “part-of” “object”

- E.g. recognize a red apple

input = symbol(s) -> algorithms who work on input -> output = more symbol(s)

Input:

Red(X)
Round(X) ...

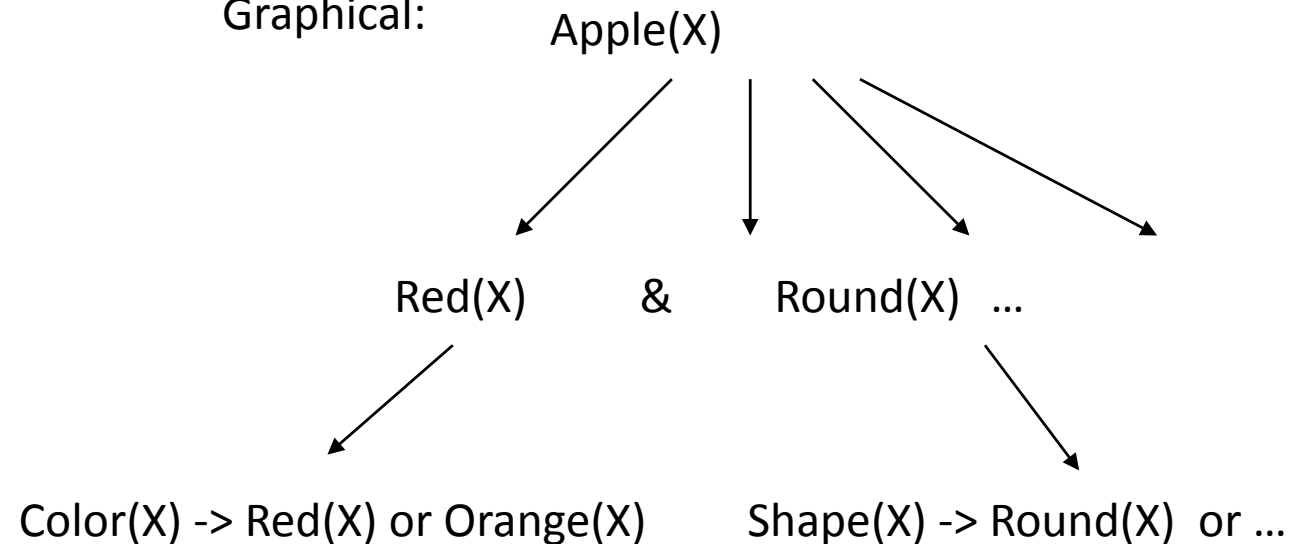
Program:

if (Orange(X) & Round(X) ...) then Orange(X)
if (Red(X) & Round(X) ...) then Apple(X)
...

Output:

Apple(X)

Graphical:



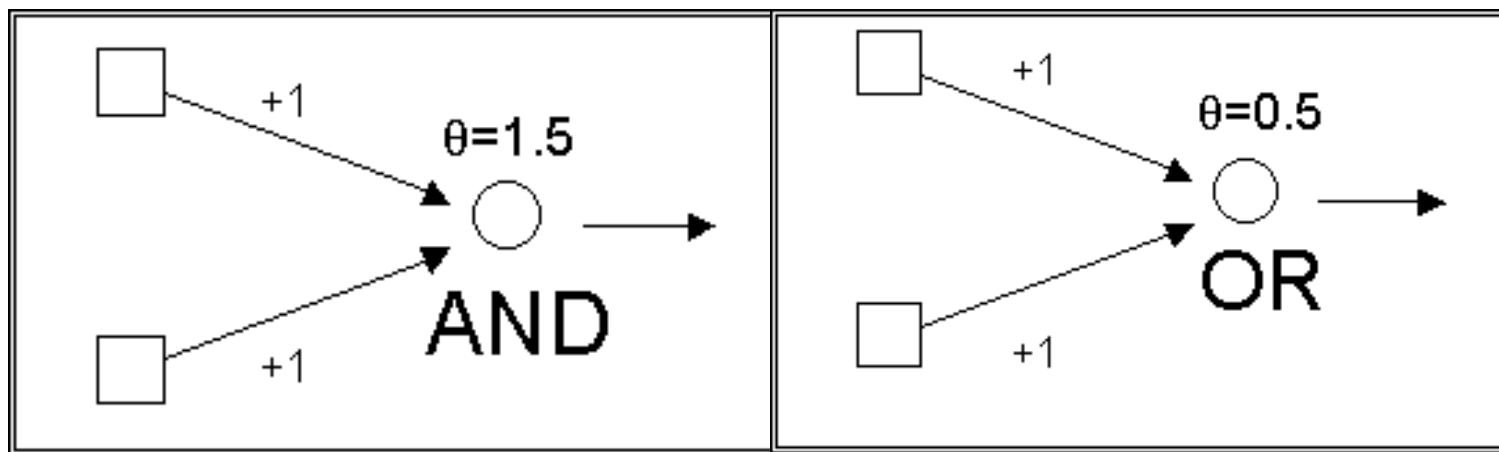
Connections

McCulloch & Pitts (1943)

- **Neural networks as computing devices**
 - What logical operations could neurons compute?
- **Five assumptions based on then-current knowledge of neurons**
 - 1. The activity of a neuron is “all-or-none” (binary coding)
 - 2. Each neuron has a fixed threshold on the required number of synapses that need to be excited before the neuron itself will be excited. Weights are identical.
 - 3. Synaptic action causes a time delay before firing.
 - 4. Inhibition is absolute.
 - 5. The physical structure of a network of neurons doesn't change with time; connections and their strengths are static.

McCulloch/Pitts neurons

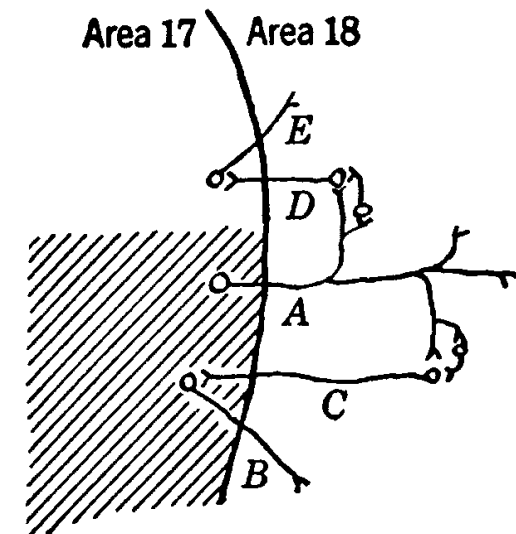
- McCulloch/Pitts neurons can then be used to compute any (finite) logical function



- BUT, McCulloch/Pitts networks can't learn.

Hebb (1949)

- The first rule for self-organized *learning*
- Hebb recognized the existence of feedforward, long range lateral, and feedback connections
- These cortical circuits admit self-sustaining activity that reverberate in « cell assemblies »
- synapses are the fundamental computational and learning unit
- *activity-dependent synaptic plasticity* as a basic operation



Hebb, D. (1949). Organization of Behavior: A Neuropsychological Theory (New York: John Wiley and Sons).

Learning in a Hebbian network

- *“When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part 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.”*

LT Potentiation (Bliss & Lomo, 1973; Kelso et al, 1986)

LT Depression (Markram et al. 1997)

The “Hebb rule”

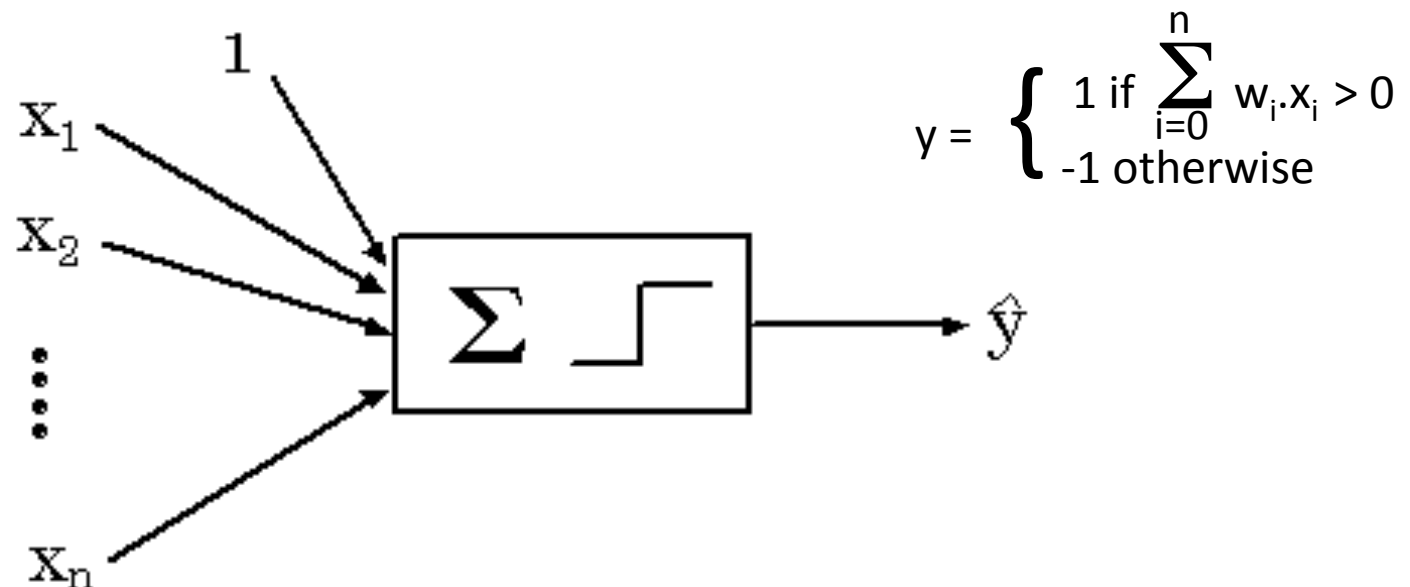


- $\Delta w_{ij} = \eta a_i a_j$
 - where a 's are activation values (-1 or 1), and η is a learning rate parameter.
 - Equation is applied until weights “saturate” (typically at 1) and do not keep increasing as inputs are presented.
- Think of Hebbian learning as picking up on correlations between features in the environment
 - Features that co-occur will grow strong positive weights, those that do not occur together will have grow negative weights, random pairing produces zero weights

The perceptron

(Rosenblatt, 1958, 1962)

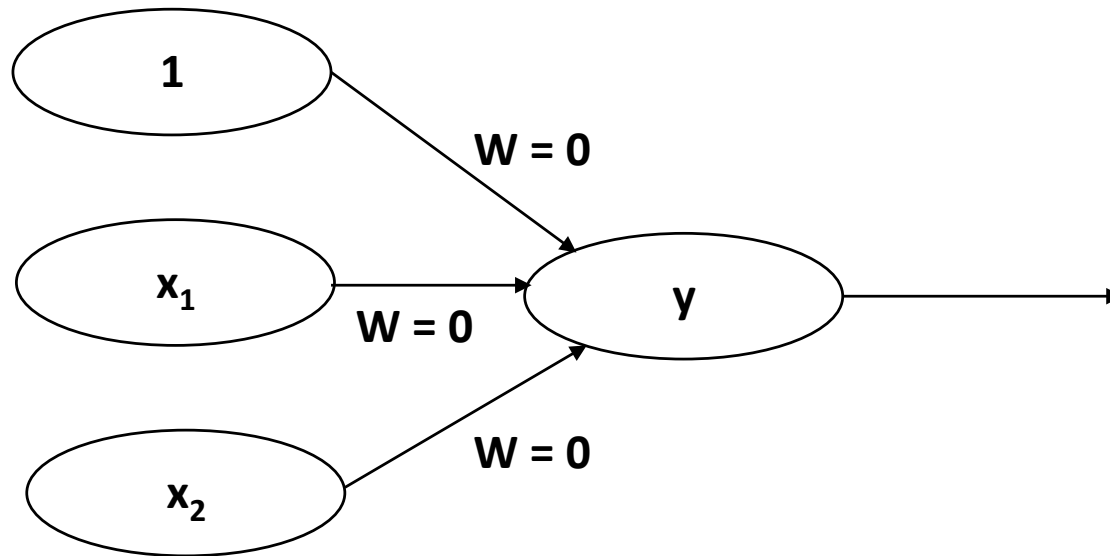
- First model for learning with a teacher (supervised learning)
- McCulloch-Pitts neurons (linear-threshold) with connections that can be modified by learning



Perceptron Learning Rule

- Start with random connections w_i
- Error-driven learning rule (delta rule):
 - $\Delta w_i = \eta (t - y) x_i$
 - t is the target value (given by the teacher)
 - y is the perceptron output
 - η is a small constant (e.g. 0.1) called *learning rate*
- If the output is correct ($t=y$) the weights w_i are not changed
- If the output is incorrect ($t \neq y$) the weights w_i are changed such that the output of the perceptron for the new weights is *closer* to t (error decreases).
- The algorithm converges to the correct classification
 - if the training data is linearly separable
 - and η is sufficiently small

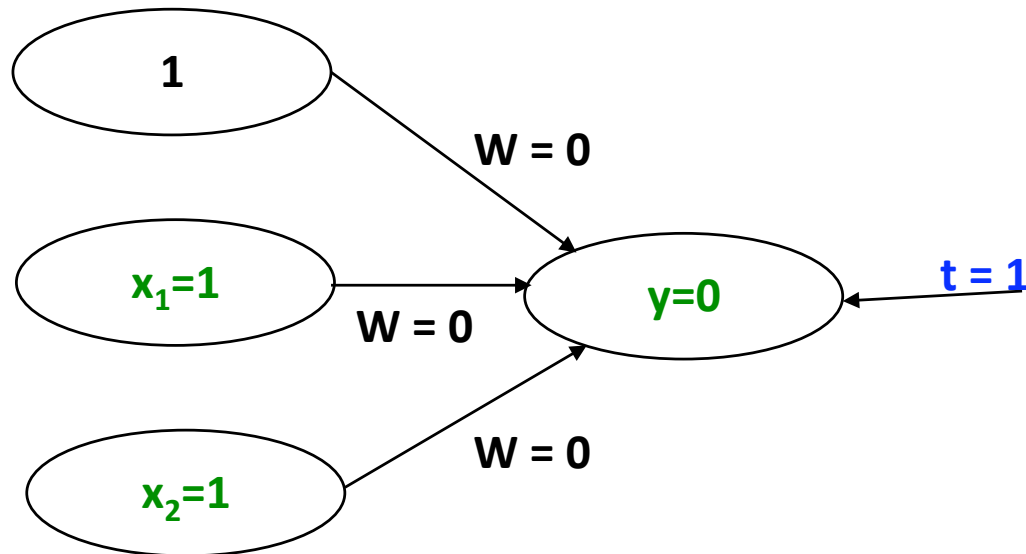
Example: learning the AND



AND		
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

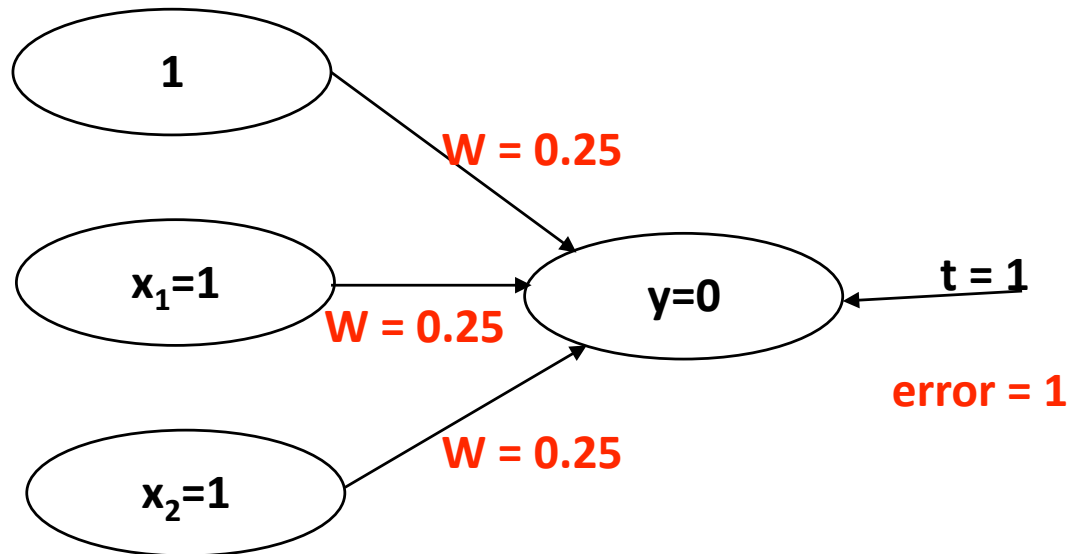
- Initial weights: 0, $n=0.25$

Example: learning the AND



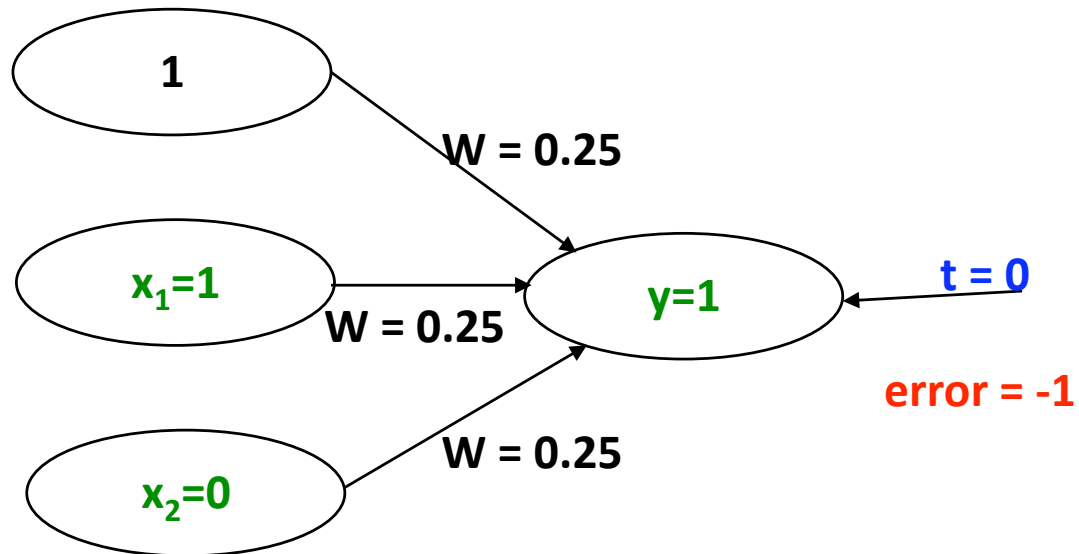
AND		
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Example: learning the AND



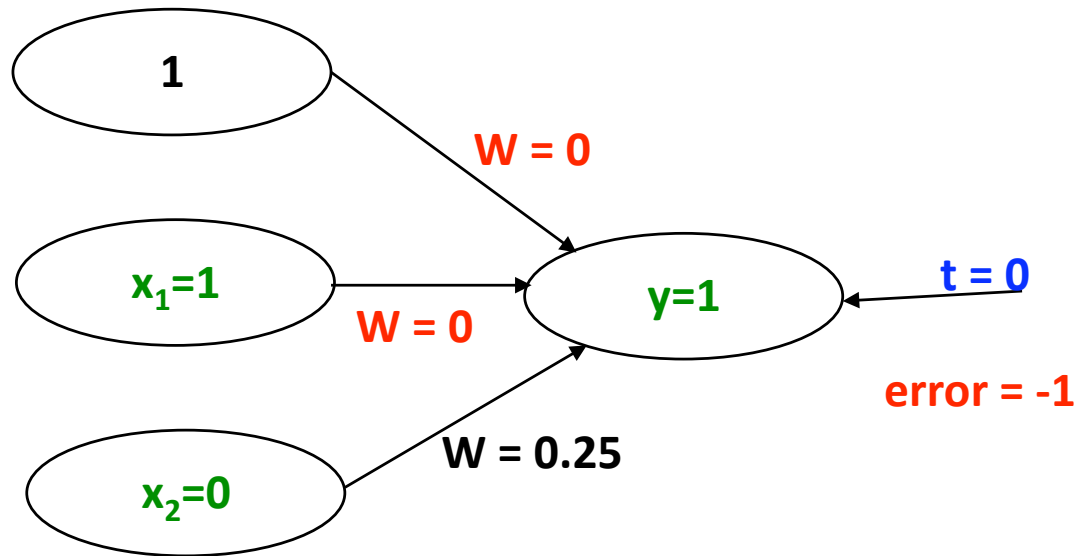
AND		
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Example: learning the AND



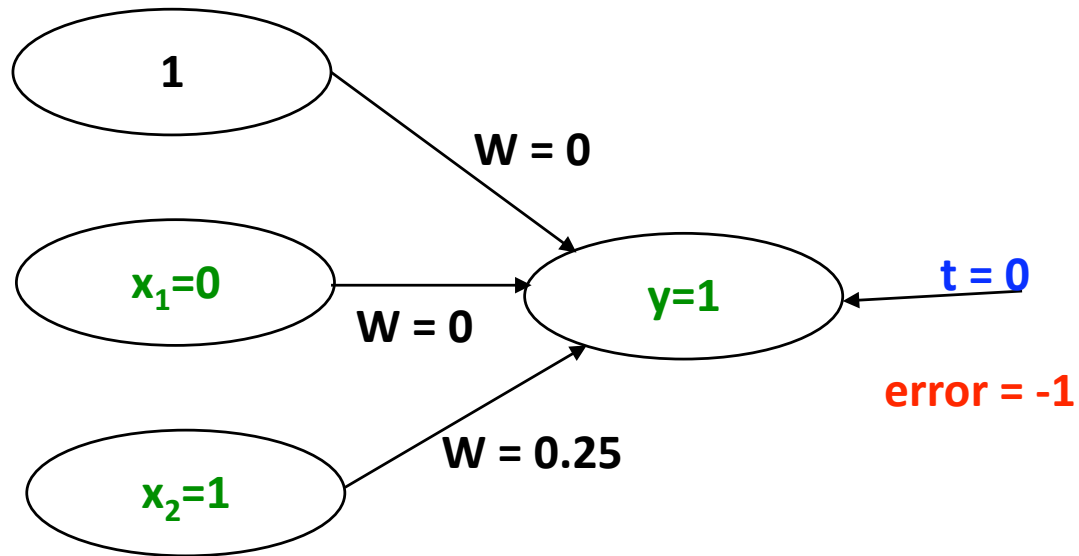
AND		
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Example: learning the AND



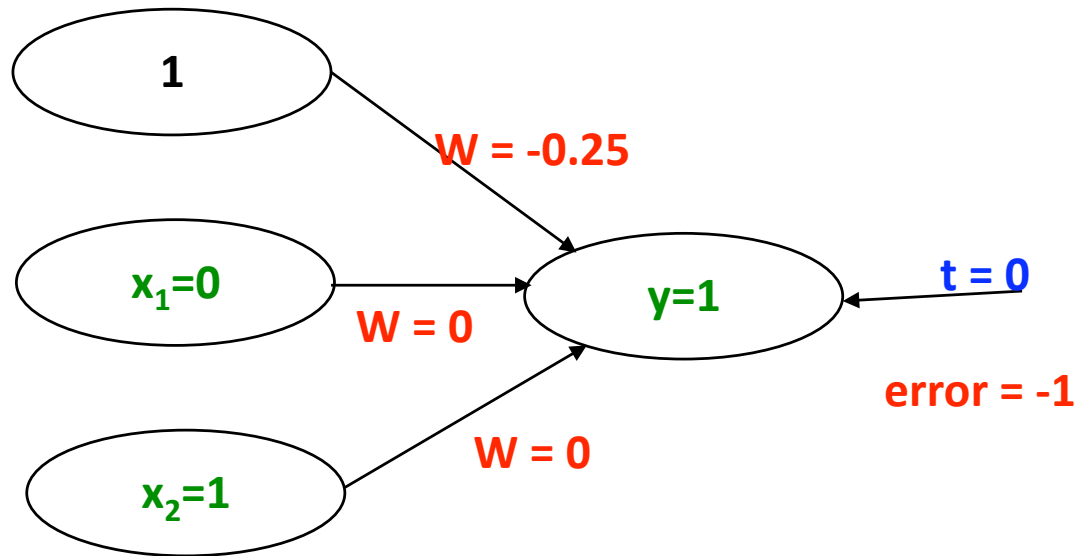
AND		
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Example: learning the AND



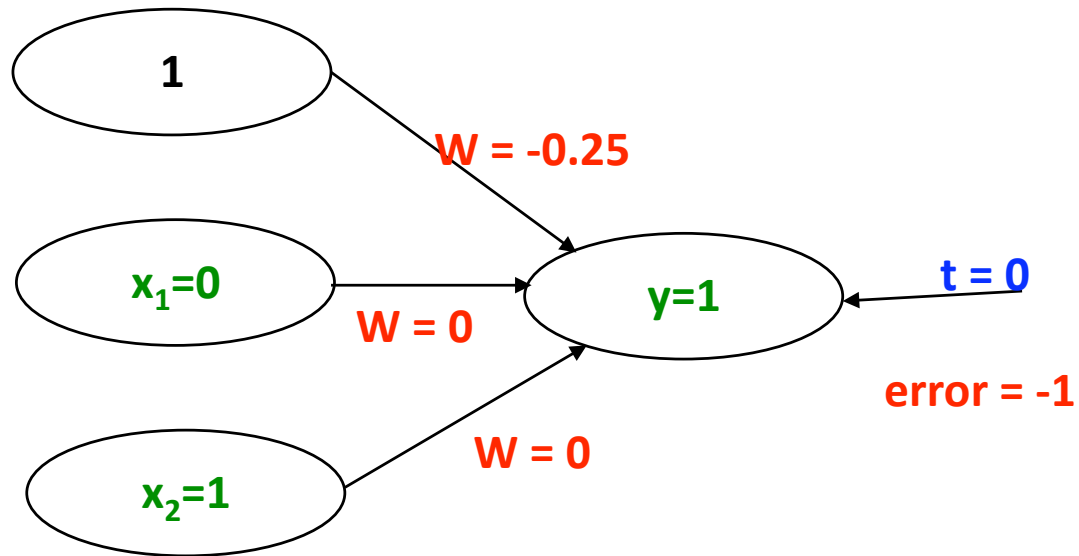
AND		
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Example: learning the AND



AND		
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

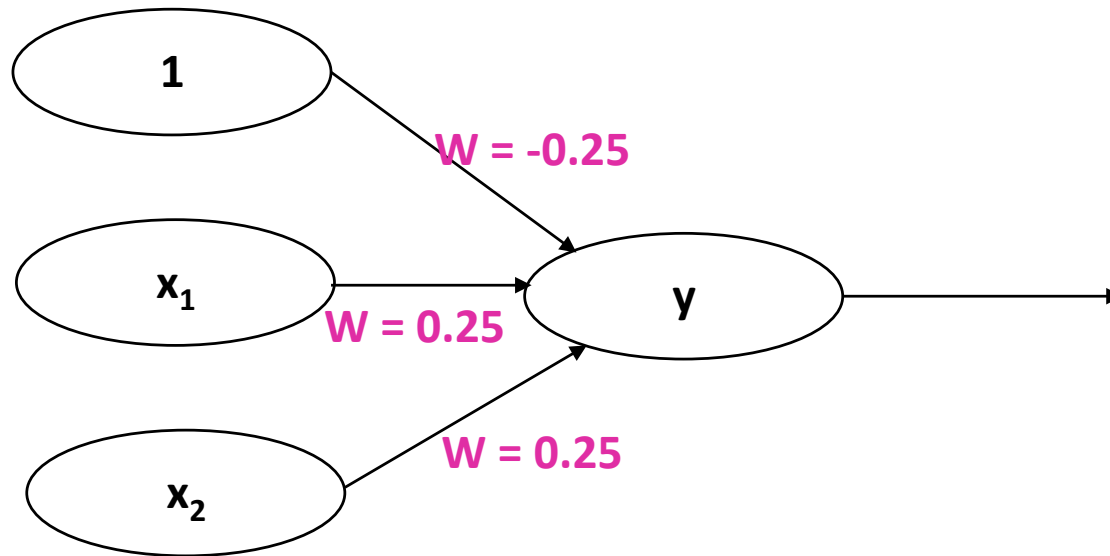
Example: learning the AND



AND		
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

... and so on and so forth ...

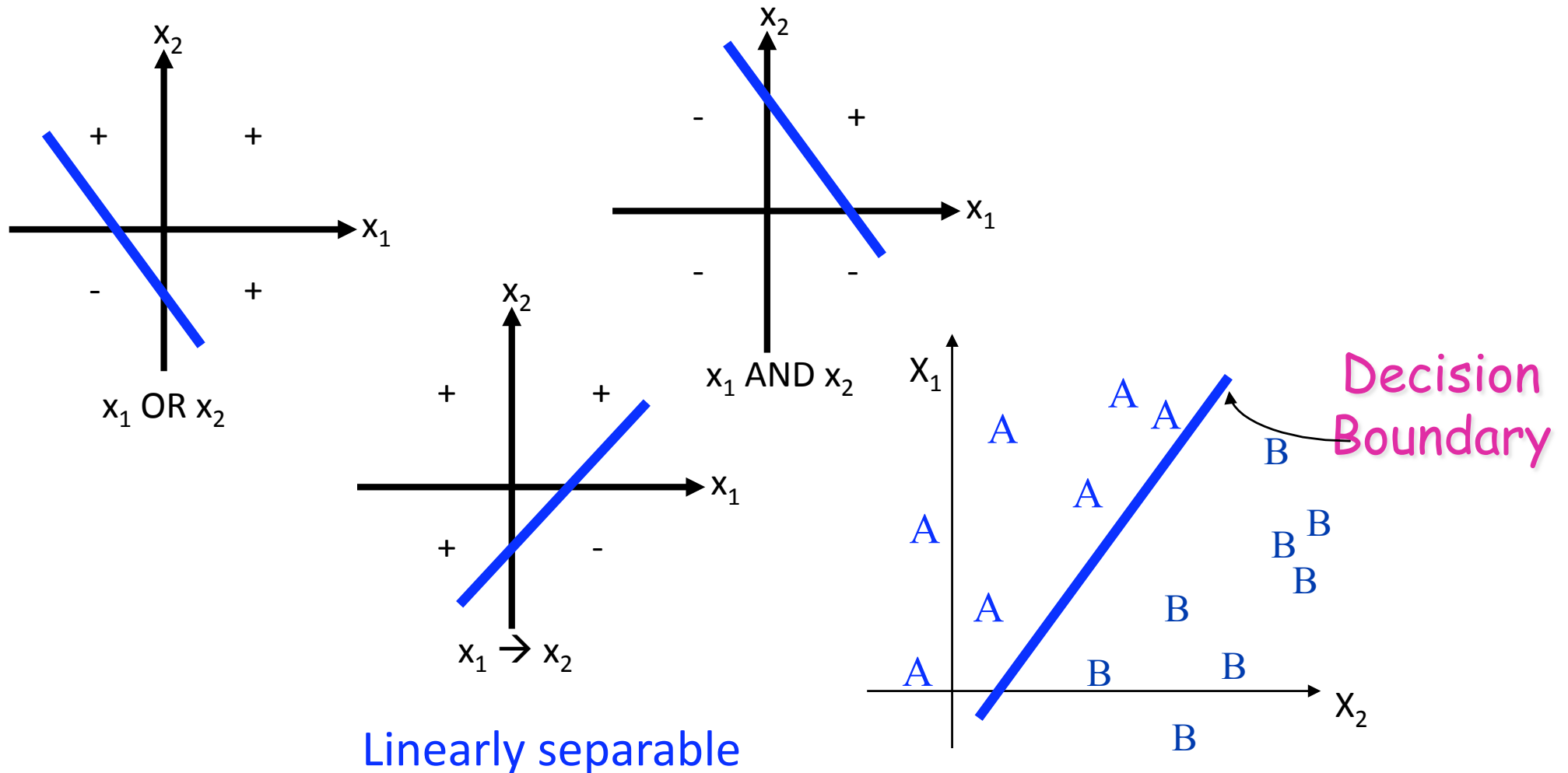
Example: learning the AND



AND		
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

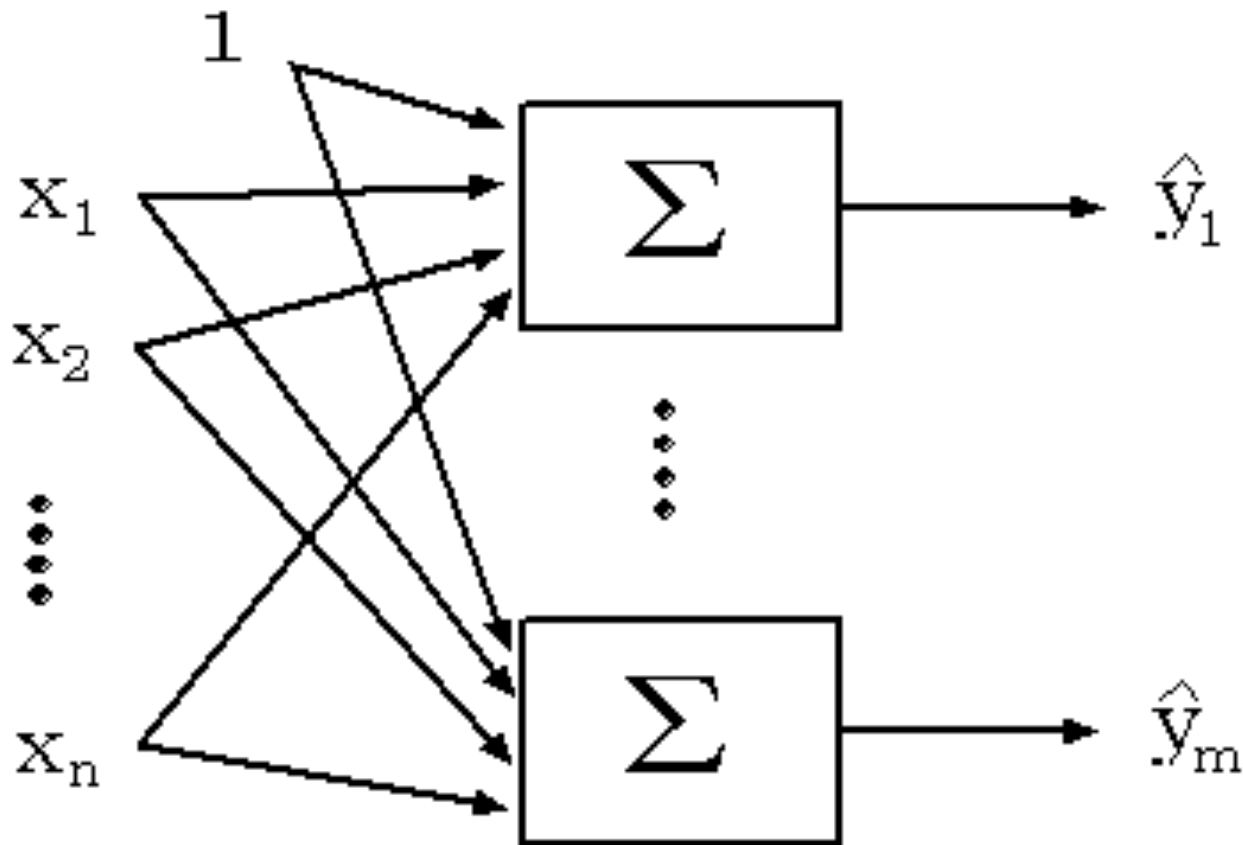
Final weights

Decision Boundary of a Perceptron



- Perceptron is doing something similar to linear discriminant analysis, binomial regression (or Bayes classification when the distributions are gaussian)
- Decision boundary is a hyperplane when more than 2 inputs

Generalization: Multiple outputs

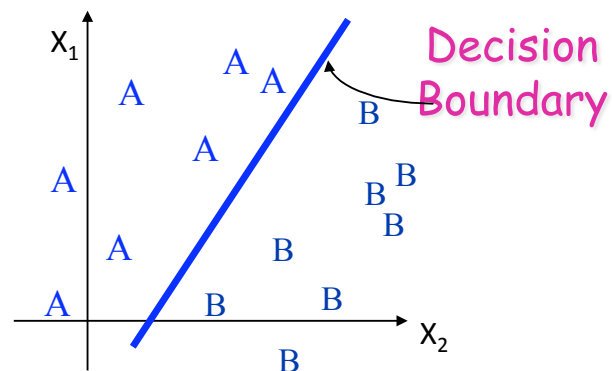
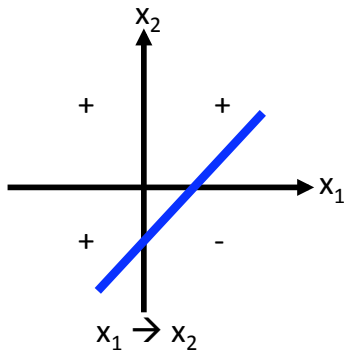
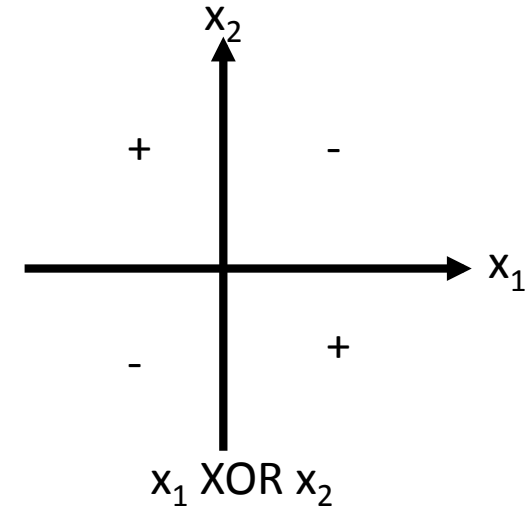
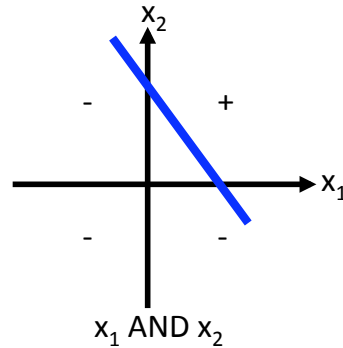
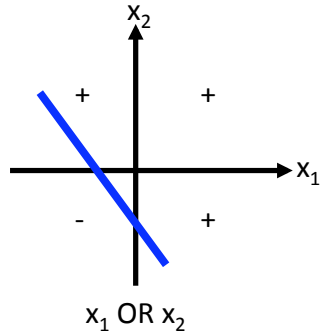


- similar multinomial LDA or multinomial regression

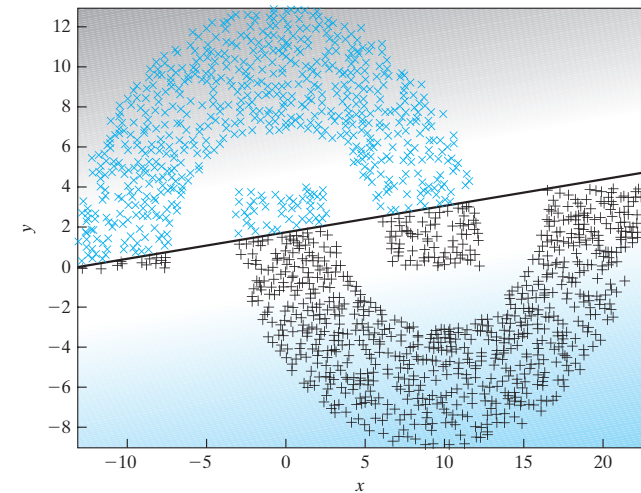
Minsky & Papert (1969)

- Presented a formal analysis of the properties of perceptrons and revealed several fundamental limitations.
- Limitations
 - Can't learn nonlinearly separable problems like the XOR
 - *More...*

Decision Boundary of a Perceptron



Linearly separable



Non-Linearly separable

Minsky & Papert cont.

- Limitations
 - So....can't learn nonlinearly separable problems like the XOR
 - Although including “hidden layers” allows one to hand-design a network that can represent XOR and related problems, they showed that the perceptron learning rule can't *learn* the required weights.
 - They also showed that even those functions that *can* be learned by perceptron rule learning may require huge amounts of learning time

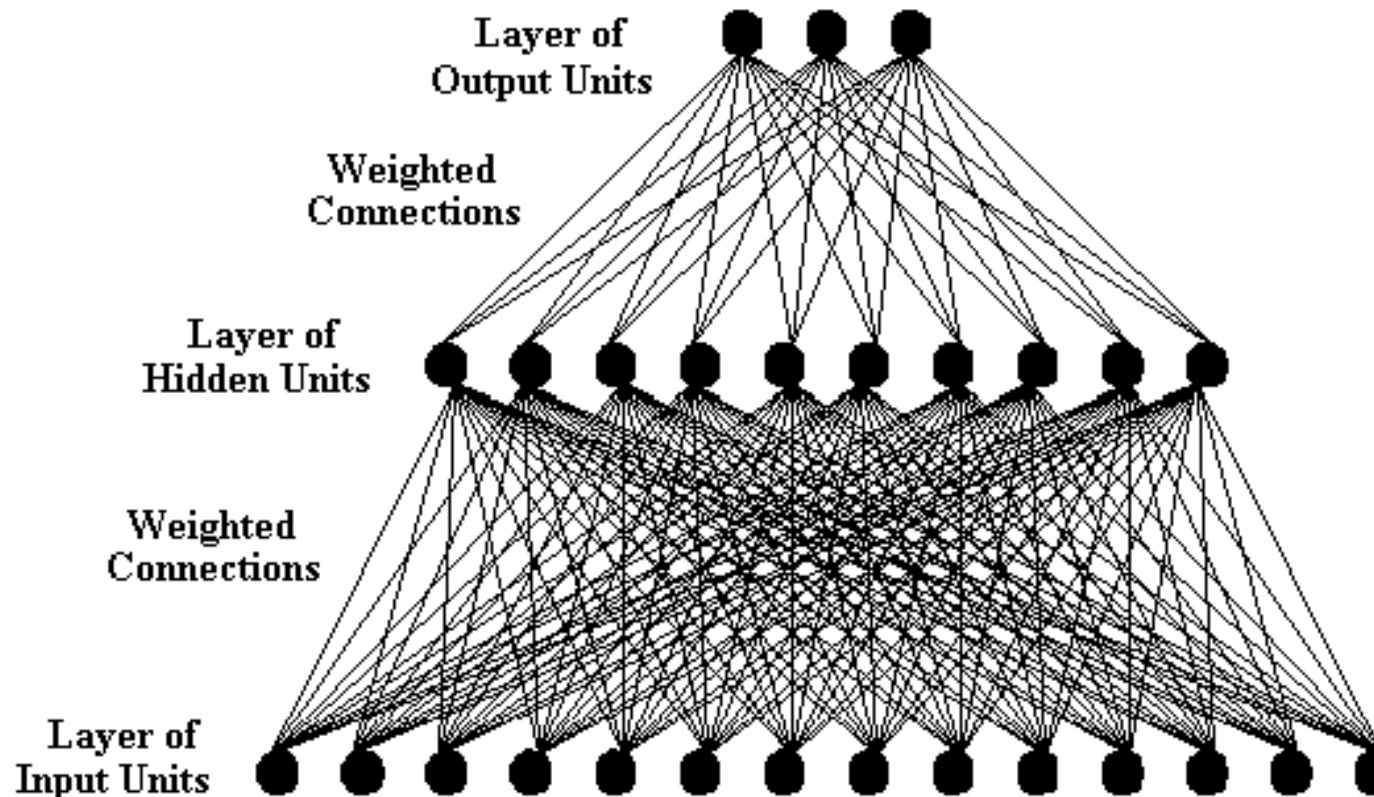
Fallout of Minsky & Papert's analysis

- This paper was nearly the death of this budding field.
- Subsequent research was largely done in “garages”.
 - i.e., only in obscure academic circles.

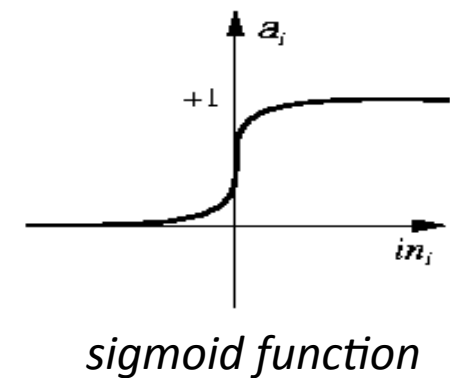
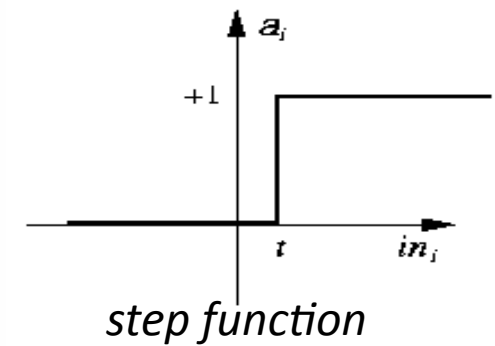
The revival: the 80s

- multilayered feedforward networks
 - Generalization of the Delta Rule: backpropagation of error
 - learning of distributed representations
- Symmetric recurrent networks
 - winner take all
 - autoassociation

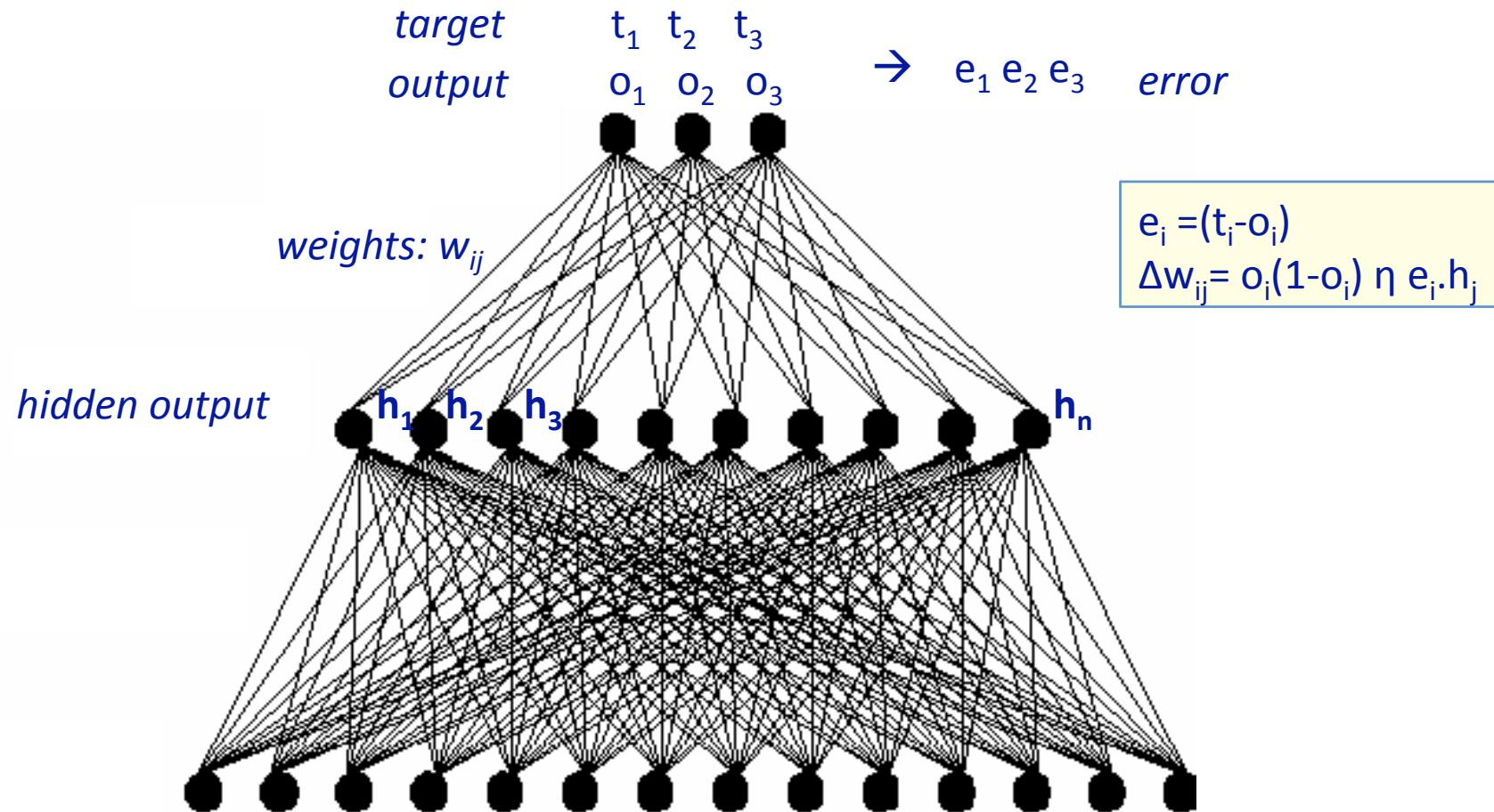
The revival I: the Multi-Layer Perceptron



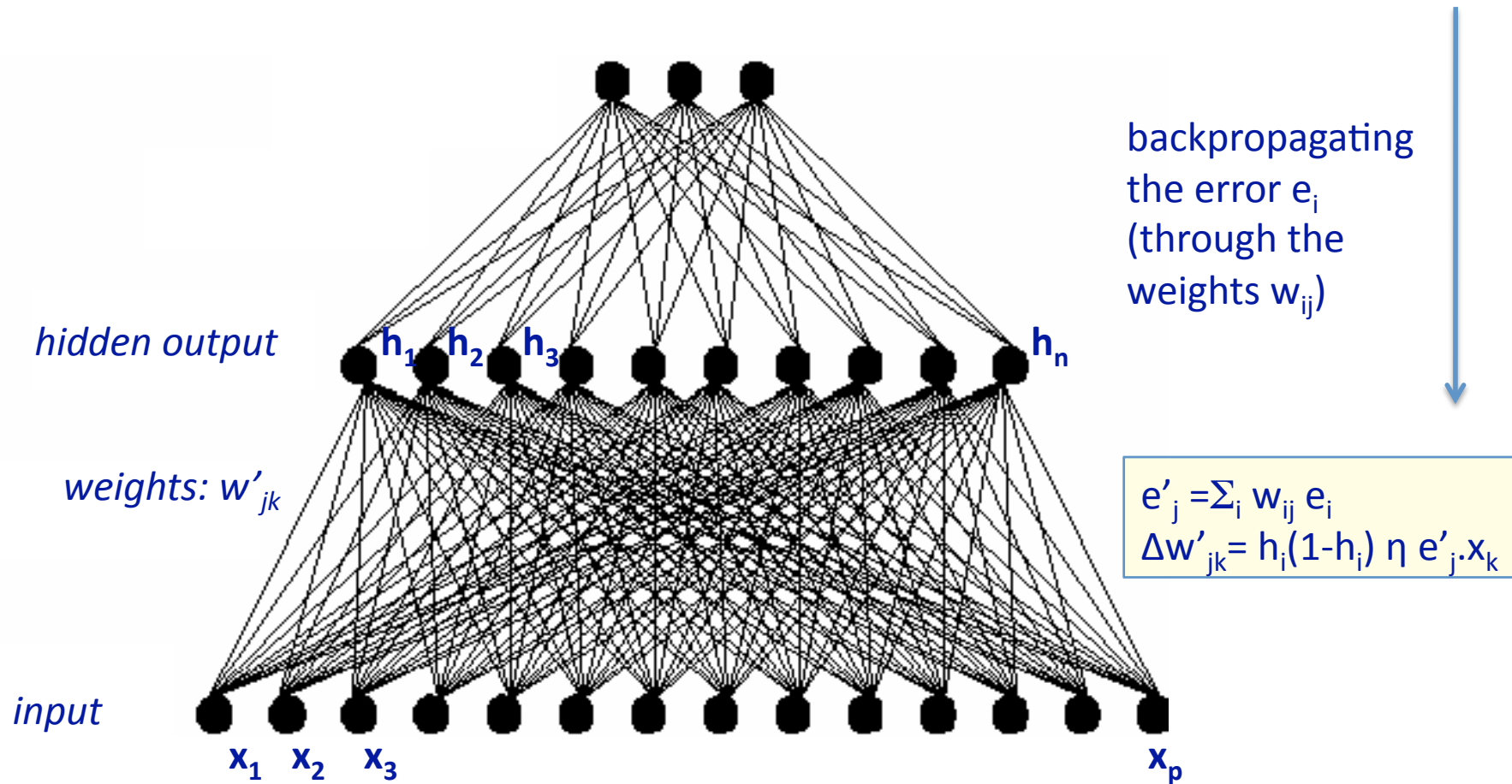
Activation functions



The backpropagation learning rule (I)



The backpropagation learning rule (II)



Expressive Capabilities of MLP

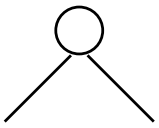
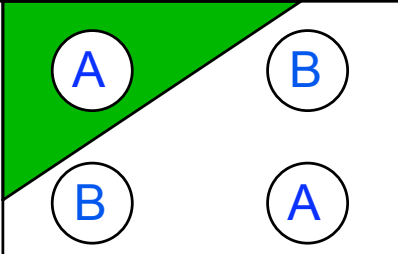
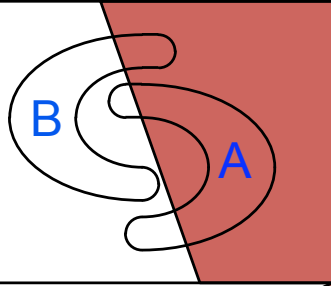
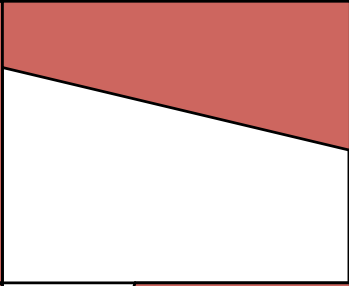
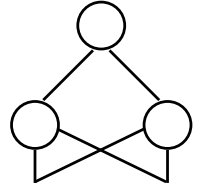
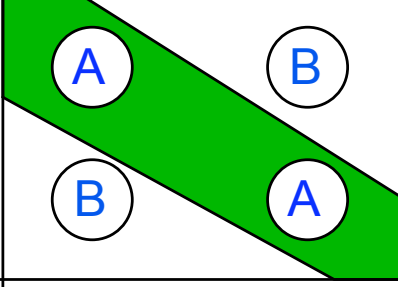
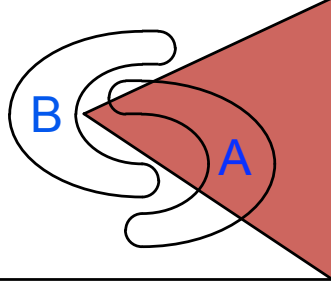
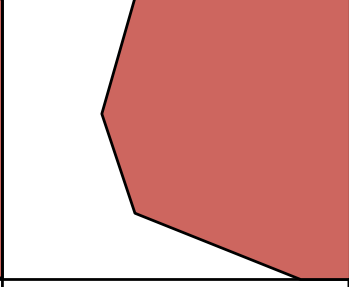
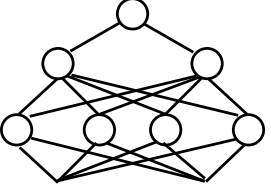
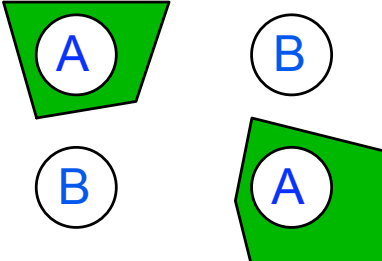
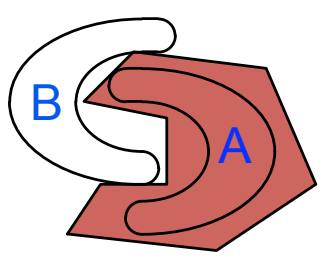
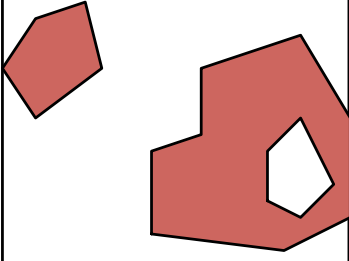
Boolean functions

- Every boolean function can be represented by network with single hidden layer
- But might require exponential (in number of inputs) hidden units

Continuous functions

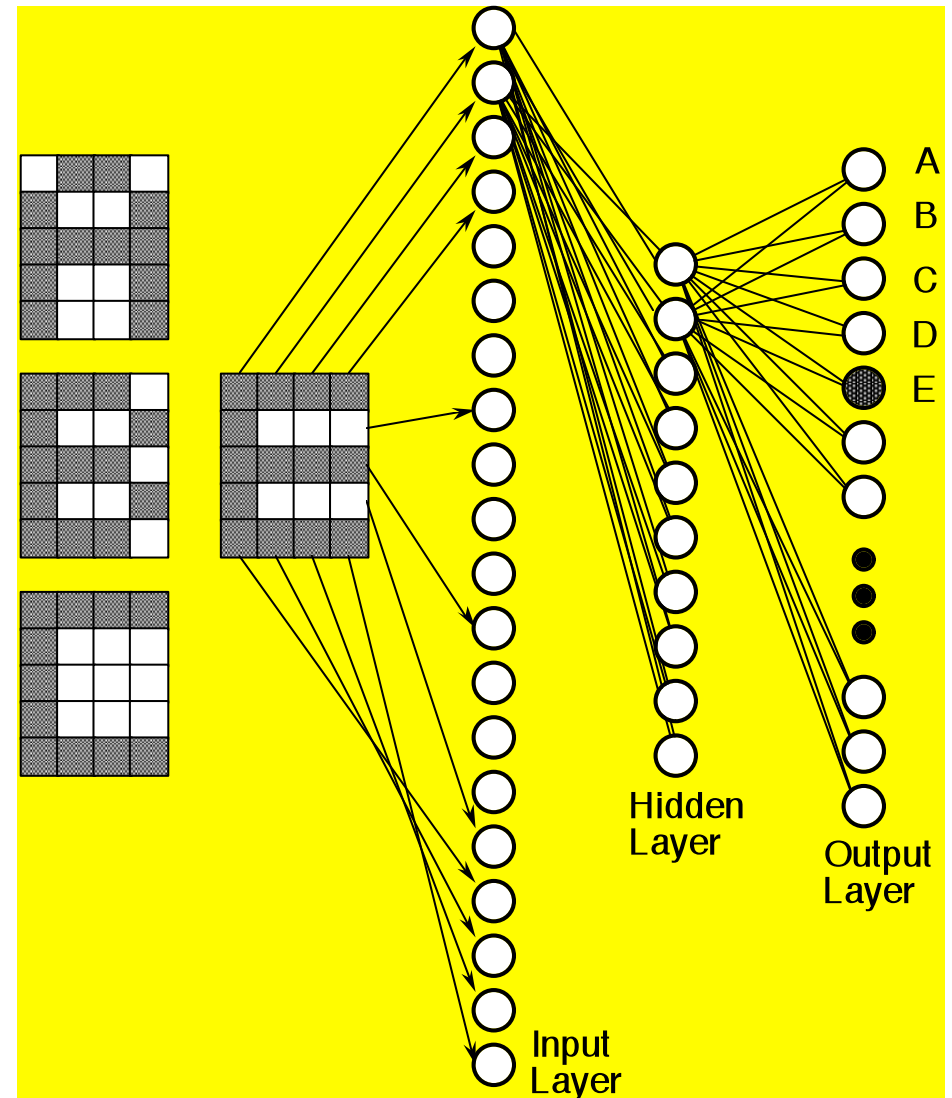
- Every bounded continuous function can be approximated with arbitrarily small error, by network with one hidden layer [Cybenko 1989, Hornik 1989]
- Any function can be approximated to arbitrary accuracy by a network with two hidden layers [Cybenko 1988]

Different Non-Linearly Separable Problems

<i>Structure</i>	<i>Types of Decision Regions</i>	<i>Exclusive-OR Problem</i>	<i>Classes with Meshed regions</i>	<i>Most General Region Shapes</i>
<i>Single-Layer</i> 	<i>Half Plane Bounded By Hyperplane</i>			
<i>Two-Layer</i> 	<i>Convex Open Or Closed Regions</i>			
<i>Three-Layer</i> 	<i>Arbitrary (Complexity Limited by No. of Nodes)</i>			

Example: Neural network for OCR

- feedforward network
- trained using Back-propagation



Example: ALVINN

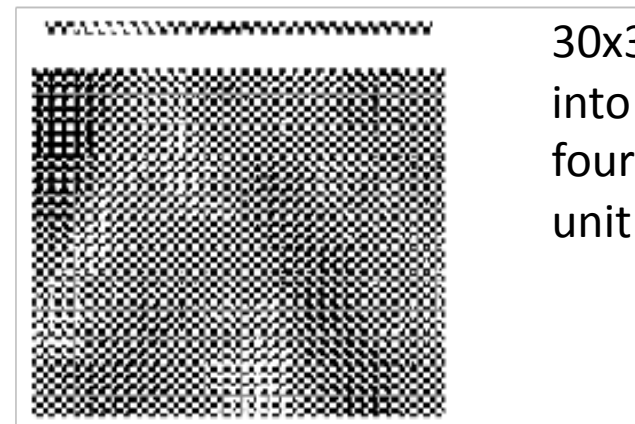
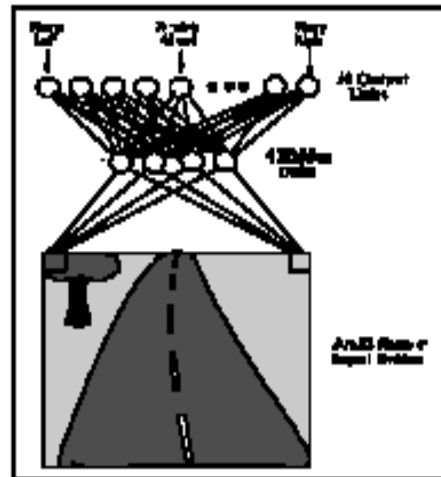
Drives 70 mph on a public highway



30 outputs
for steering

4 hidden
units

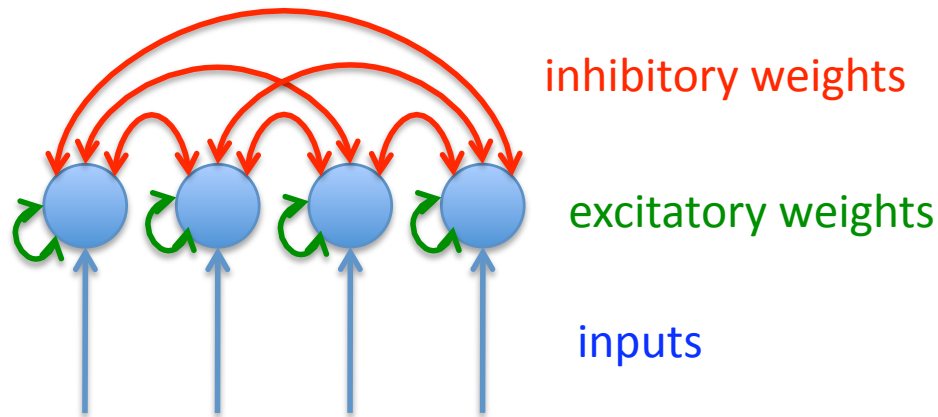
30x32 pixels
as inputs



30x32 weights
into one out of
four hidden
unit

The revival II: Symmetric dynamical networks

- A very simple example: a winner-take all network



- at each step:
 - compute the total input for all units
 - compute the output for all units
 - iterate

The revival II: Symmetric dynamical networks

- Boltzman machine (Hinton & Sejnowsky, 1983)
 - neurons: binary (-1, 1)
 - network: symmetric weights
 - update: stochastic (1 with probability $1 / (1 + e^{-\text{sum}(\text{inputs})})$)
 - dynamics: global energy minimization, basins of attraction
 - learning rule: tries to reproduce the distribution of its inputs (stochastically learns the basins of attraction)
- Hopfield network (Hopfield 1982)
 - deterministic variant ('temperature'=0)
 - learning rule becomes Hebb Rule.
 - performs pattern completion, content-addressable memory

The revival III: McClelland & Rumelhart's (1986)

Parallel Distributed Processing

- Basic mechanisms
 - feature discovery and competitive learning
 - dynamical system and harmony theory
 - learning in Boltzmann machines
 - internal representation through backpropagation
- Formal analysis
 - linear algebra
 - activation functions
 - delta rule
- Psychological Applications
 - schemata and sequential through processes
 - speech perception (TRACE)
 - blackboard model of reading
 - learning and memory
 - learning past tense of english verbs
 - sentence processing: assigning roles to constituents
- Biological mechanisms
 - anatomy of cortex
 - place recognition and goal location
 - neural plasticity and critical period
 - amnesia and distributed memory

An example: early models of lexical access

- Morton (1969) Logogen theory
- Forster (1976) Serial search

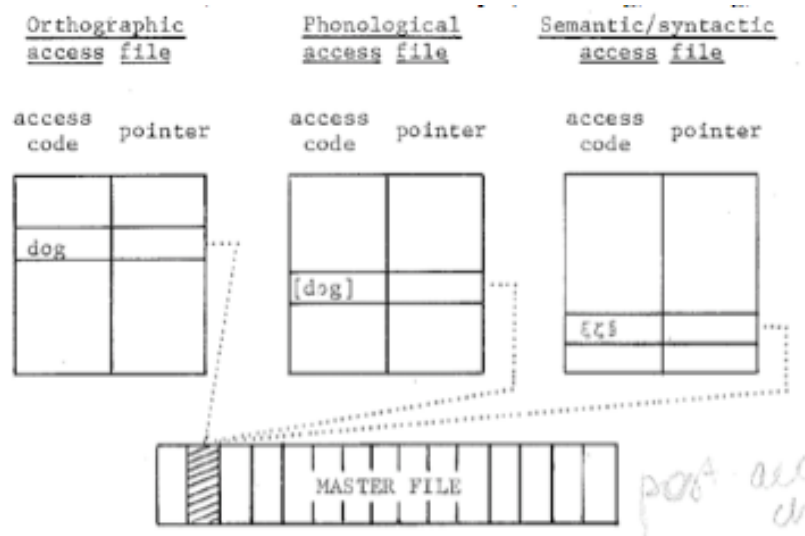
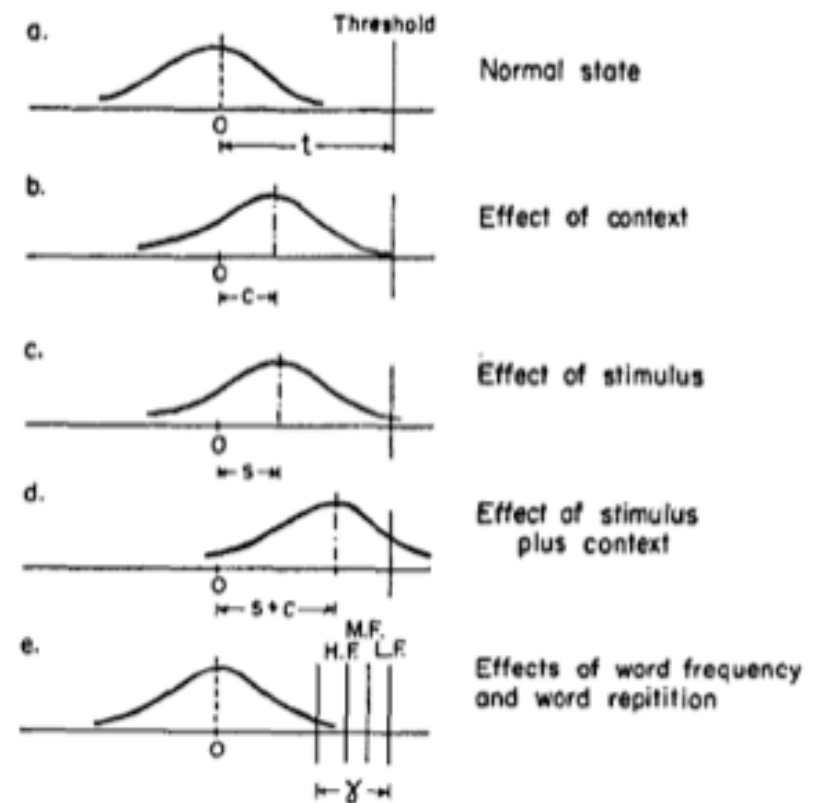


Figure 4. Organization of peripheral access files and master file.

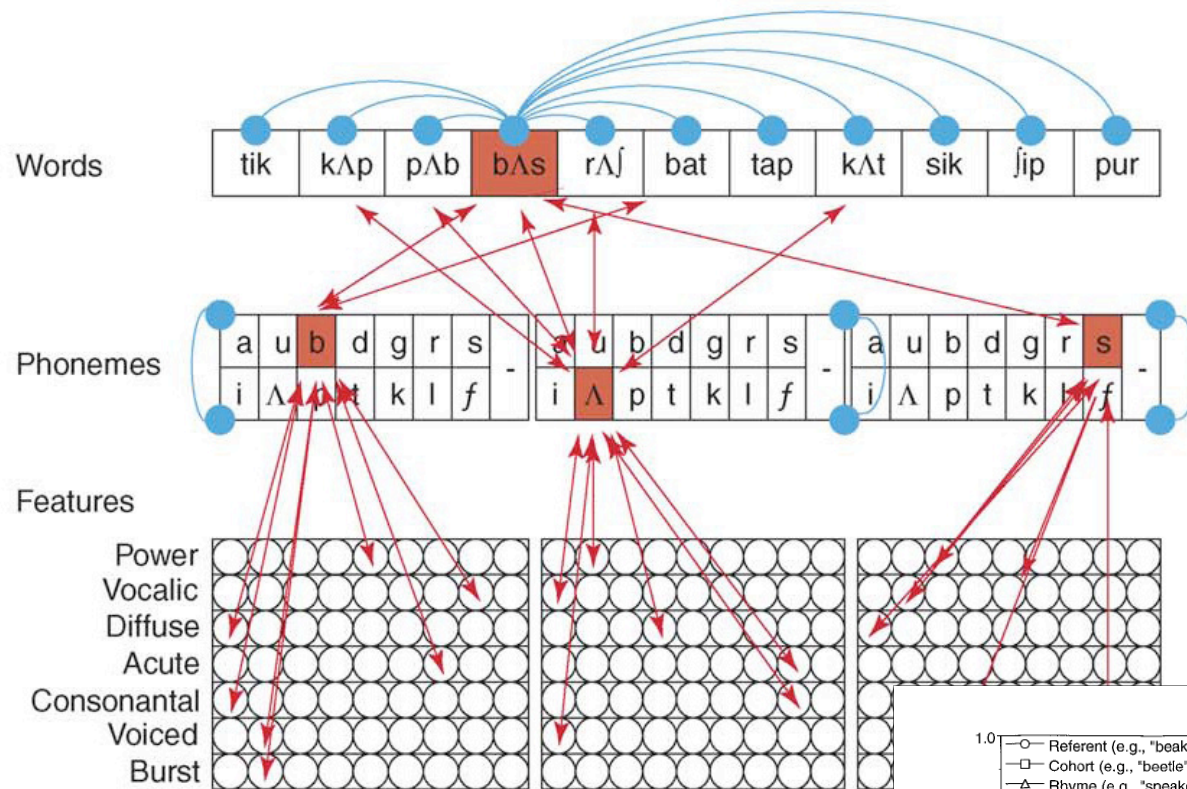


Forster, K. I. (1976). Accessing the mental lexicon. In , *New approaches to language mechanisms*. Amsterdam: North-Holland.

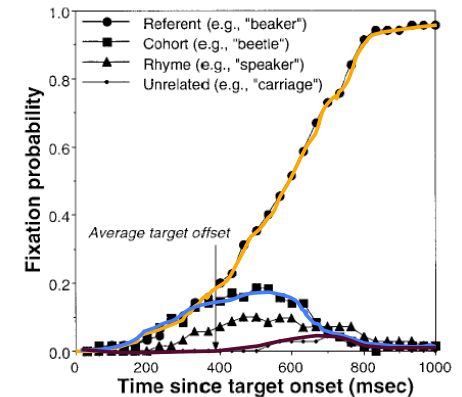
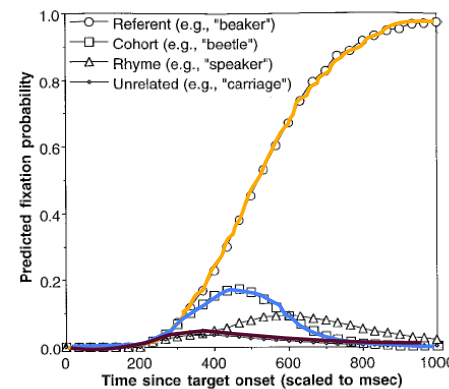
Morton, J. (1969). Interaction of information in word recognition. *Psychological Review* 76(2). 165-178.

a new PDP model

- Elman & McClelland (1986)



Allopena et al (1998)



The revival III: McClelland & Rumelhart's (1986)

Parallel Distributed Processing

- Cognition involves the spreading of activation, relaxation, statistical correlation.
- Represents a method for how symbolic systems might be implemented
 - Hypothesized that apparently symbolic processing is an emergent property of subsymbolic operations.
- Advantages
 - Fault tolerance & graceful degradation
 - Can be used to model learning
 - More naturally capture nonlinear relationships
 - Fuzzy information retrieval
 - bridges the gap with real neural processing

The **critique**: Pinker & Mehler (1988)

- Lachter & Bever: connectionist theories are a return to associationism (Chomsky vs Skinner revisited)
- Pinker & Prince: connectionist models of the capacity to derive the past tense of English verbs is inadequate
 - rules: wug → wugged
 - exceptions: put -> put, go->went, dig->dug
- Fodor & Pylyshyn: connectionist theories are inadequate models of language and thought

Fodor & Pylyshyn

- Position of the problem: classical theories vs connectionism
 - Agree:
 - both classical theories & connectionism are *representationalists* (they assign some 'meaning' to the elements – symbols or nodes)
 - Disagree
 - classical theory encode structural relationships and processes (eg, constituents, variables, rules)
 - connectionists only encode causal relationships and processes (x causes y to fire)
 - Arguments against connectionist systems: mental representation and processes are structure sensitive
 - combinatorial semantics
 - semantics of « J. loves M. » derived from semantics of « J. », « loves » and « M. »
 - productivity
 - the list of thoughts/sentences is not finite (I can construct new thoughts with old ones)
 - systematicity
 - I construct them in a systematic way
 - eg: « x loves M. » (where x can be any proper noun)
 - eg: If I can think « J. loves M. », I can think « M. loves J. »
 - recursivity & constituent structure:
 - If I can think « P. thinks that M. is nice » I can think « J thinks that P thinks that M is nice »
- > connectionist systems have none of the above properties

Fodor & Pylyshyn (cont)

- Objections to symbolic/classical systems
 - rapidity of cognitive processes/neural speed
 - difficulty of pattern recognition/content based retrieval in conventional architectures
 - committed to rule vs exception dichotomy
 - inadequate for intuitive /nonverbal behavior
 - acutely sensitive to damage/noise (vs graceful degradation)
 - storage in classical systems is passive
 - inadequate account of gradual/frequency based application of rules
 - inadequate account of nondeterminism
 - no account of neuroscience
 - none of these arguments are valid or relevant
- CONCLUSIONS
 1. current connectionist theories are inadequate
 2. if they were to be made adequate they would be mere implementation of classical architecture

Questions

- les arguments de Fodor contre les modèles connectionnistes sont ils valides
- les réponses de Fodor aux arguments des connectionnistes sont elles pertinentes
- que penser de la première conclusion (les modèles connectionnistes sont inadéquats comme modèles de la pensée et du langage)
- que penser de la seconde conclusion (les modèles connectionnistes qui sont adéquats ne sont que des implémentations des modèles classiques)

basic biblio

- Rumelhart, D.E., J.L. McClelland and the PDP Research Group (1986). Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations, Cambridge, MA: MIT Press
- McClelland, J.L., D.E. Rumelhart and the PDP Research Group (1986). Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 2: Psychological and Biological Models, Cambridge, MA: MIT Press
- Pinker, Steven and Mehler, Jacques (1988). Connections and Symbols, Cambridge MA: MIT Press.
- Jeffrey L. Elman, Elizabeth A. Bates, Mark H. Johnson, Annette Karmiloff-Smith, Domenico Parisi, Kim Plunkett (1996). Rethinking Innateness: A connectionist perspective on development, Cambridge MA: MIT Press.
- Marcus, Gary F. (2001). The Algebraic Mind: Integrating Connectionism and Cognitive Science (Learning, Development, and Conceptual Change), Cambridge, MA: MIT Press