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
  - compositionalité et produit tensoriel
  - recursivité et systeme dynamique
  - (gradualité et modèles bayesiens)

- What is thinking ? what kind of object can think?
  - thinking is manipulation of discrete **symbols**
  - **computers** (could be programmed to) think
  - thinking is performing **parallel, distributed computations**
  - **neural networks** (could be constructed to) think

Symbols

Cognition and symbols

- reasoning = computation in a formal calculus
  - Aristote, Leibniz (universal language + reasoning calculus), Boole (formal system for logical and set theoretic reasoning), Frege (logisict program 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, ->)
  - second order logic (A,B, C, Vα, ->)
  - predicate logic (P(x))

Symboles: *x, y, z, … = - ( )*

Grammaire: *énoncé : expression = expression*

expression: (expression)

| expression – expression

| variable

variable : x | y | z …

Axiomes

$$(A) \quad x = x - (y - y)$$

$$(B) \quad x - (y - z) = z - (y - x)$$

Règles d'inférences

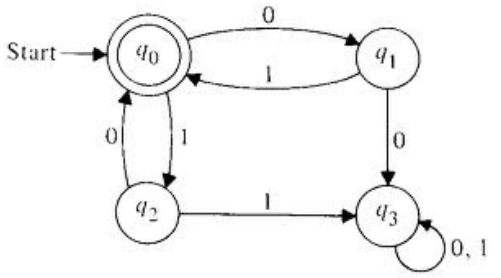
- substitution (d'une expression quelconque pour toute occurrence d'une variable)
- remplacement de a par b partout si a=b

Quine (1934). A method for generating part of arithmetic without intuitive logic

Théorèmes

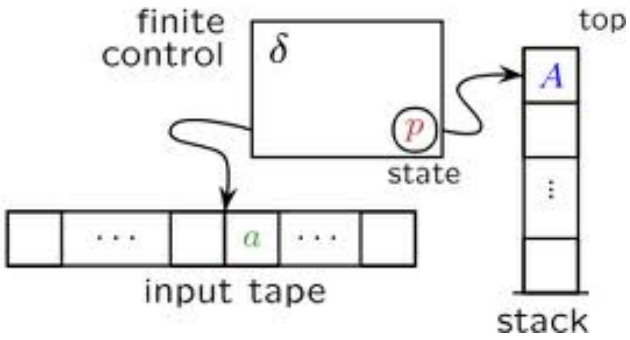
(1)	x	=	x	(A),(A)
(2)	x	=	$x-(z-z)$	(A)
(3)	$y-(x-z)$	=	$z-(x-y)$	(B)
(4)	$y-(x-(z-z))$	=	$z-z-(x-y)$	(B)
(5)	$z-(z-z-(x-y))$	=	$x-y-(z-z-z)$	(B)
(6)	$y-y-(x-x-x)$	=	$y-y-(x-x-x)$	(1)
(7)	$y-x$	=	$z-z-(x-y)$	(2),(4)
(8)	$x-z-y$	=	$w-w-(y-(x-z))$	(7)
(9)	$x-y-z$	=	$w-w-(z-(x-y))$	(7)
(10)	$z-(y-x)$	=	$x-y-(z-z-z)$	(7),(5)
(11)	$x-(y-y)$	=	$y-y-(x-x-x)$	(10)
(12)	$x-y-z$	=	$w-w-(y-(x-z))$	(3),(9)
(13)	$x-(y-z)$	=	$x-y-(z-z-z)$	(B),(10)
(14)	$x-(y-y)$	=	$x-y-(y-y-y)$	(13)
(15)	$x-y-z$	=	$x-z-y$	(8),(12)
(16)	$z-z-(x-y)$	=	$z-(x-y)-z$	(15)
(17)	$y-x$	=	$z-(x-y)-z$	(7),(16)
(18)	$y-x$	=	$y-(x-z)-z$	(3),(17)
(19)	x	=	$x-y-(y-y-y)$	(A),(14)
(20)	x	=	$y-y-(x-x-x)$	(A),(11)
(21)	$y-y-(x-x-x)$	=	x	(20),(6)

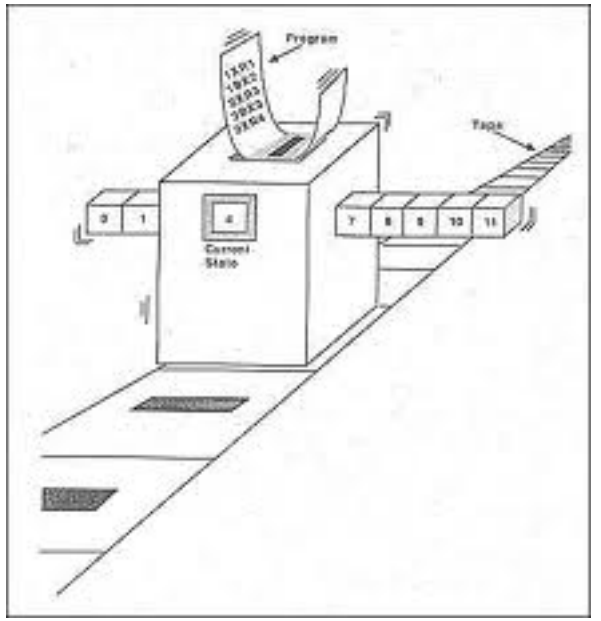
Cognition and symbols (II)



- Thinking=computation over symbols
  - computability theory (1930-) Godel, Post, Kleene, Church, Turing, Markov

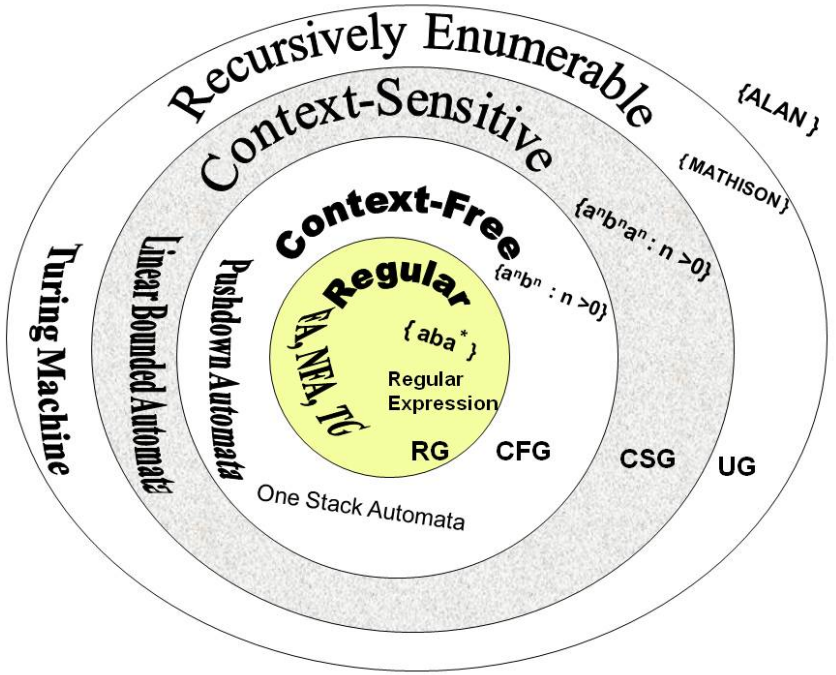
- Physical symbolic systems
  - Finite state automaton, pushdown automaton, Turing machines computational power
  - Power of automata:
    - Finite state automata
    - Pushdown automata
    - Turing machines with one tape,Turing machines with several tapes, lambda calculus (eg lisp), string rewriting system (eg production), recursive functions
  - Church thesis:
    - Turing machines are on the top of the hierarchy





theoretical background

- Formal linguistics
  - Miller & Chomsky (1963)
    - to each automaton, a class of languages
    - human languages are between context free and recursively enumerable

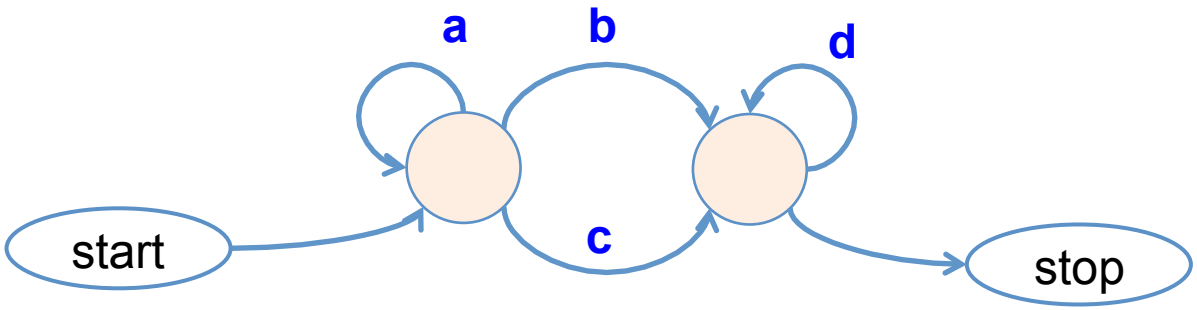


définitions

- langage formel
  - alphabet: ensemble fini de symboles: Σ={a,b, c, …}
    (ou {chien, chat, le, une, chasse, etc})
  - énoncé: séquence finie de symboles: eg: aacba
    (ou le.chien.chasse.le.chat)
  - ensemble de tous les énonces: Σ* (infini)
  - langage: L sous-ensemble (potentiellement infini) de Σ*

Alphabet $\Sigma=\{a, b, c, d\}$

Automate à états finis

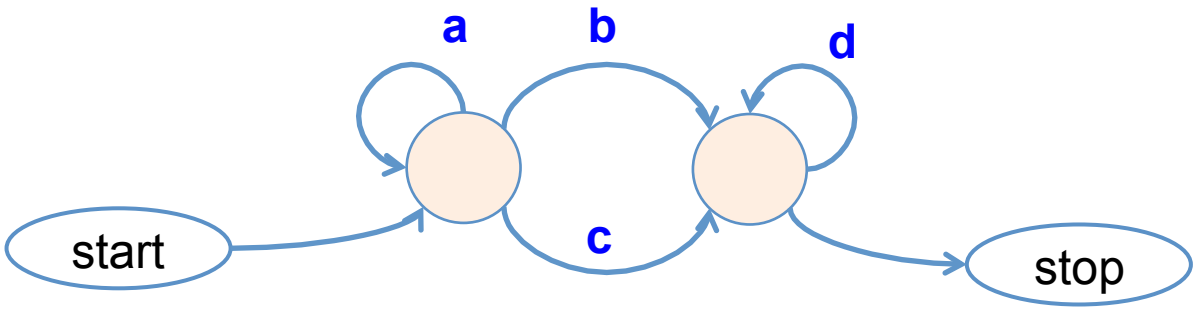


Exemples a, b, c, ab, ba, bd, ac, bddddd,cd, abd, acd,aabd, aaaaaaabddddddd, … $\in$ L ?

Alphabet

$\Sigma = \{a, b, c, d\}$

Automate à états finis



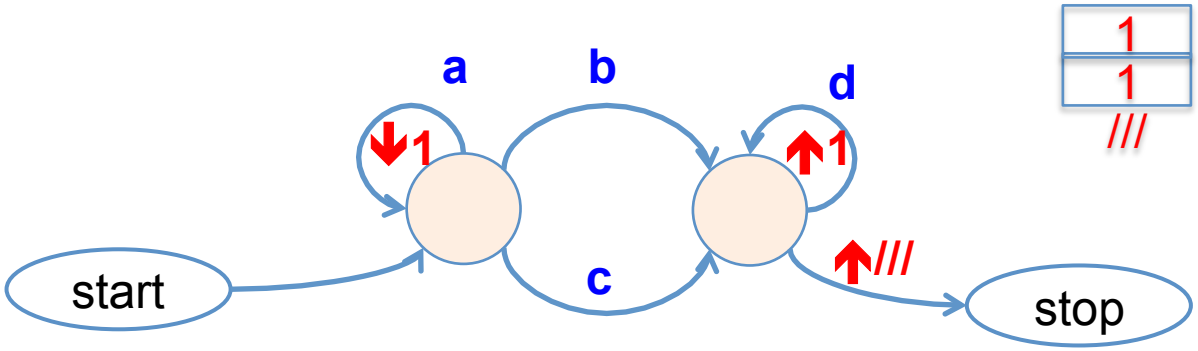
Exemples

~~a~~, b, c, ab, ~~ba~~, bd, ac, bddddd,cd, abd, acd,aabd, aaaaaaabddddddd, … ∈ L ?

Expression régulière

a*[bc]d*

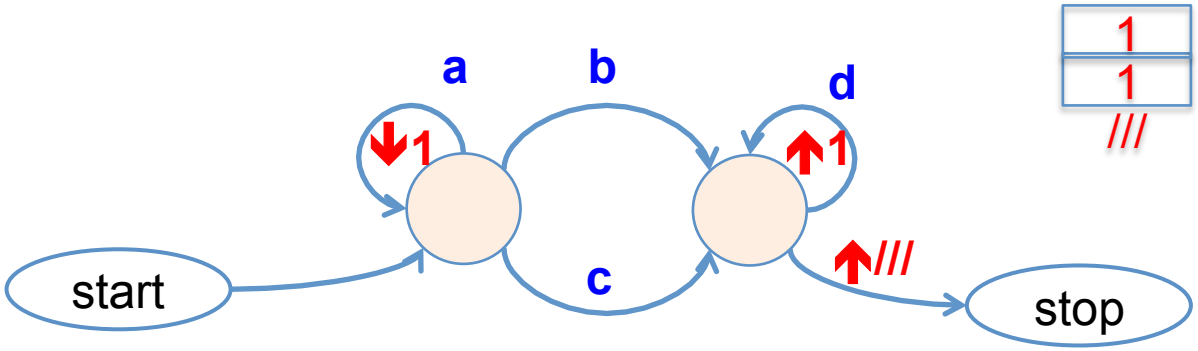
Automate
à pile



Exemples

b, c, ab, bd, bddddd,cd, abd,
acd,aabd, aaaaaaabddddddd, etc

Automate
à pile



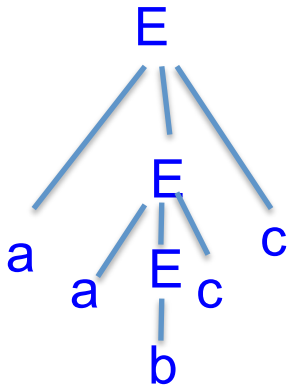
Exemples

b, c, ab, bd, bddddd,cd, abd,
acd,aabd, aaaaaaabddddddd, etc

Grammaire
algébrique
(context free)

$a^n[bc]d^n$ n>=0

$E \rightarrow aEd \mid b \mid c$



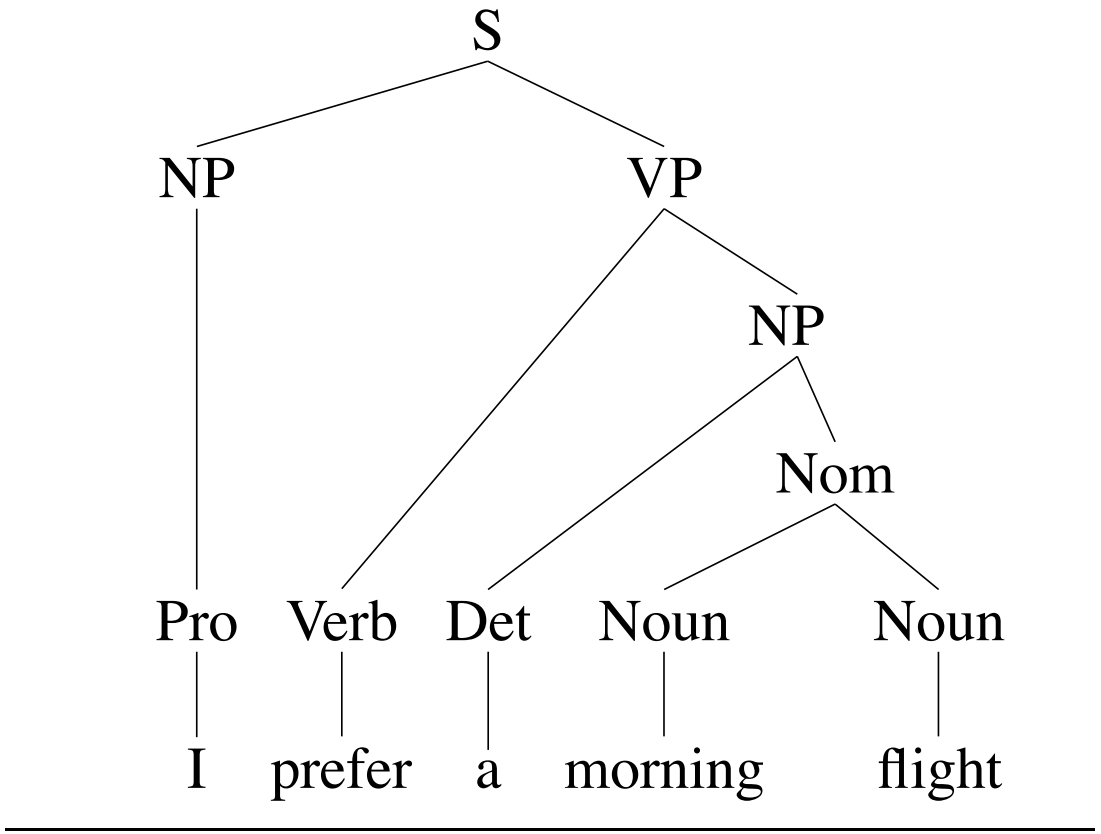
parsing: a small grammar of English

Grammar

S	$\rightarrow$	*NP VP*	I + want a morning flight
NP	$\rightarrow$	*Pronoun*	I
	|	*Proper-Noun*	Los Angeles
	|	*Det Nominal*	a + flight
Nominal	$\rightarrow$	*Noun Nominal*	morning + flight
	|	*Noun*	flights
VP	$\rightarrow$	*Verb*	do
	|	*Verb NP*	want + a flight
	|	*Verb NP PP*	leave + Boston + in the morning
	|	*Verb PP*	leaving + on Thursday
PP	$\rightarrow$	*Preposition NP*	from + Los Angeles

Lexicon

Noun	$\rightarrow$	*flights* | *breeze* | *trip* | *morning* | ...
Verb	$\rightarrow$	*is* | *prefer* | *like* | *need* | *want* | *fly*
Adjective	$\rightarrow$	*cheapest* | *non−stop* | *first* | *latest* | *other* | *direct* | ...
Pronoun	$\rightarrow$	*me* | *I* | *you* | *it* | ...
Proper-Noun	$\rightarrow$	*Alaska* | *Baltimore* | *Los Angeles* | *Chicago* | *United* | *American* | ...
Determiner	$\rightarrow$	*the* | *a* | *an* | *this* | *these* | *that* | ...
Preposition	$\rightarrow$	*from* | *to* | *on* | *near* | ...
Conjunction	$\rightarrow$	*and* | *or* | *but* | ...


S
NP
VP
NP
Nom
Pro
Verb
Det
Noun
Noun
I
prefer
a
morning
flight

- note on dependencies in human languages

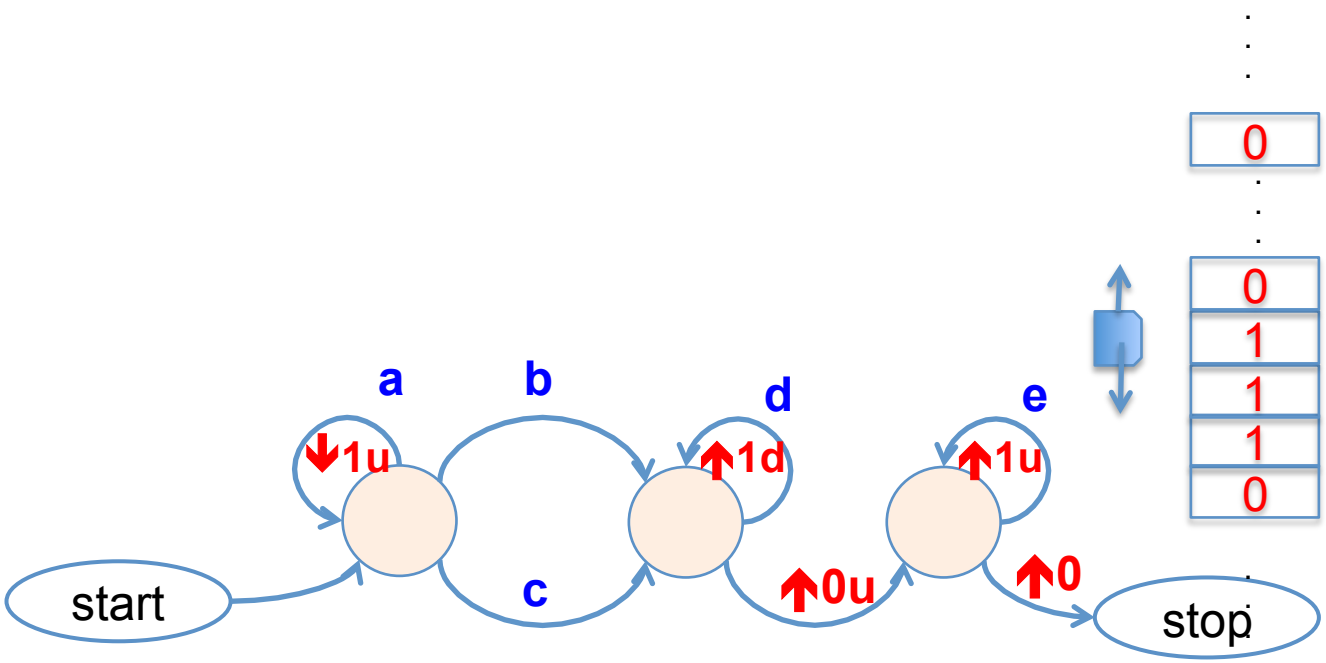
a. that we$_1$ let$_1$ the children$_2$ help$_2$ Hans$_3$ paint$_3$ the house$_3$.

b. daß wir$_1$ die Kinder$_2$ dem Hans$_3$ das Haus$_3$ streichen$_3$ helfen$_2$ lassen$_1$.

c. das mer$_1$ d'chind$_2$ em Hans$_3$ es huus$_3$ lönd$_1$ hälfe$_2$ aastriiche$_3$.

- a and b can be handled by a push down automaton, but not c

Machine
de Turing



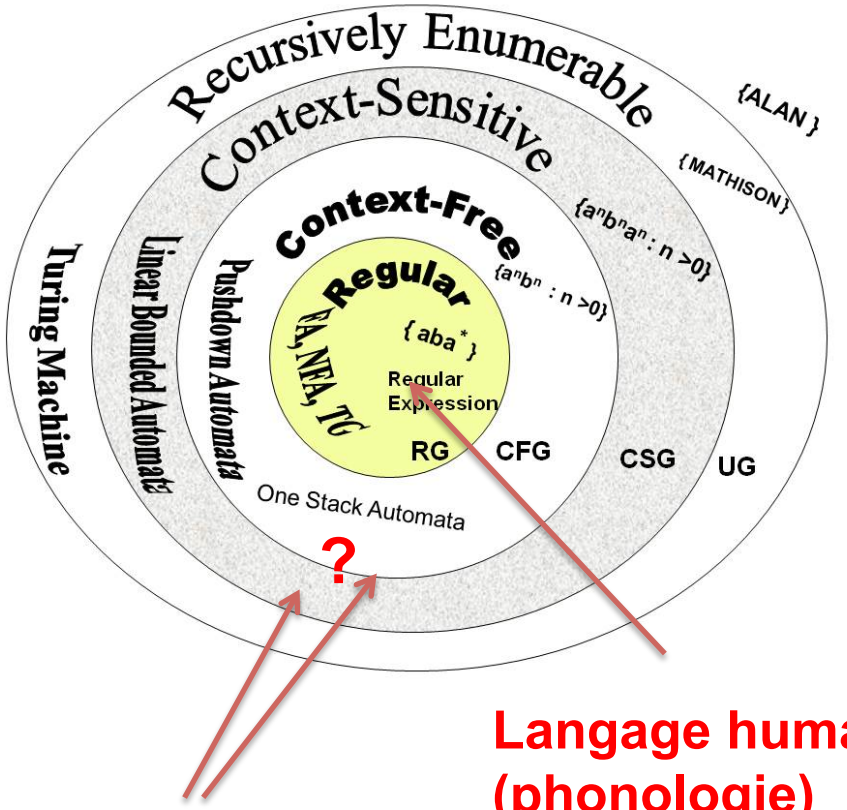
Grammaire
récursivement
énumérable
Système de
production

$a^n[bc]d^ne^n \quad n>=0$

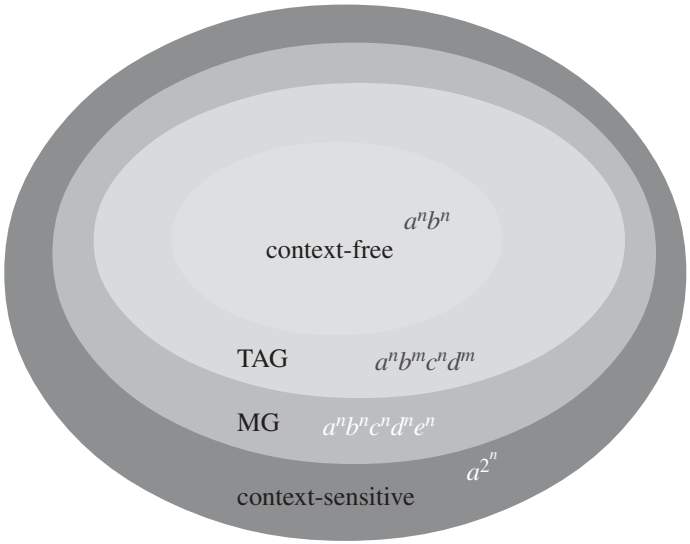
$(N \cup \Sigma)^* N (N \cup \Sigma)^* \rightarrow (N \cup \Sigma)^*$

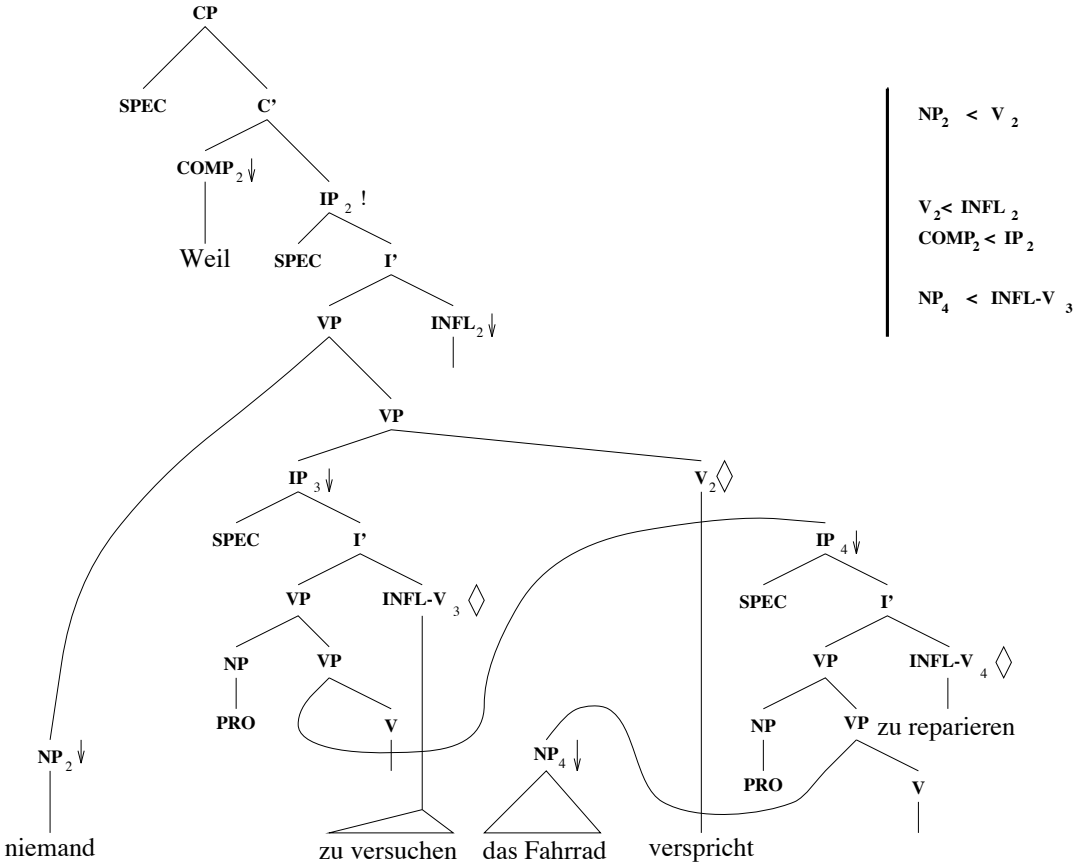
quelle classe d'automate pour les langues humaines?

– Miller & Chomsky (1963)
  - à chaque classe d'automates, une classe de langages
  - les langages humains sont entre context free et récursivement énumérables (en ce qui concerne la syntaxe) et régulières ou sous
    -régulières en ce qui concerne la phonologie



Minimalist grammarts,
Dependency grammars,
Tree adjoining grammars,
categorical grammars
… etc…

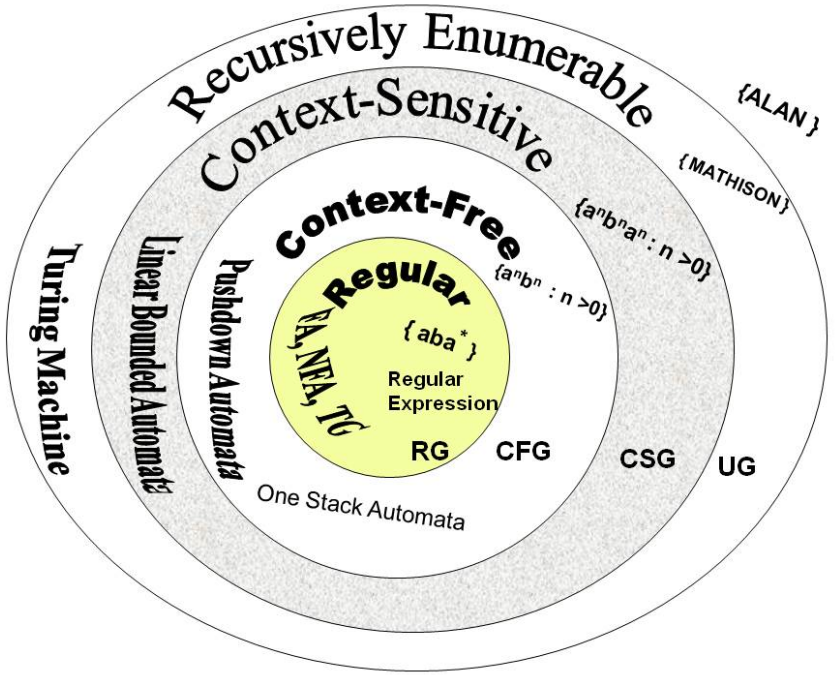




Jager et al (2012). Refining the Chomsky Hierarchy. Phil Tran Roy Soc B

Cognition and symbols (III)

- Formal linguistics
  - Miller & Chomsky (1963)
    - to each automaton, a class of languages
    - human languages are between context free and recursively enumerable



  - Symbolic AI (1956)
    - 1980: 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?
    - 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)
…

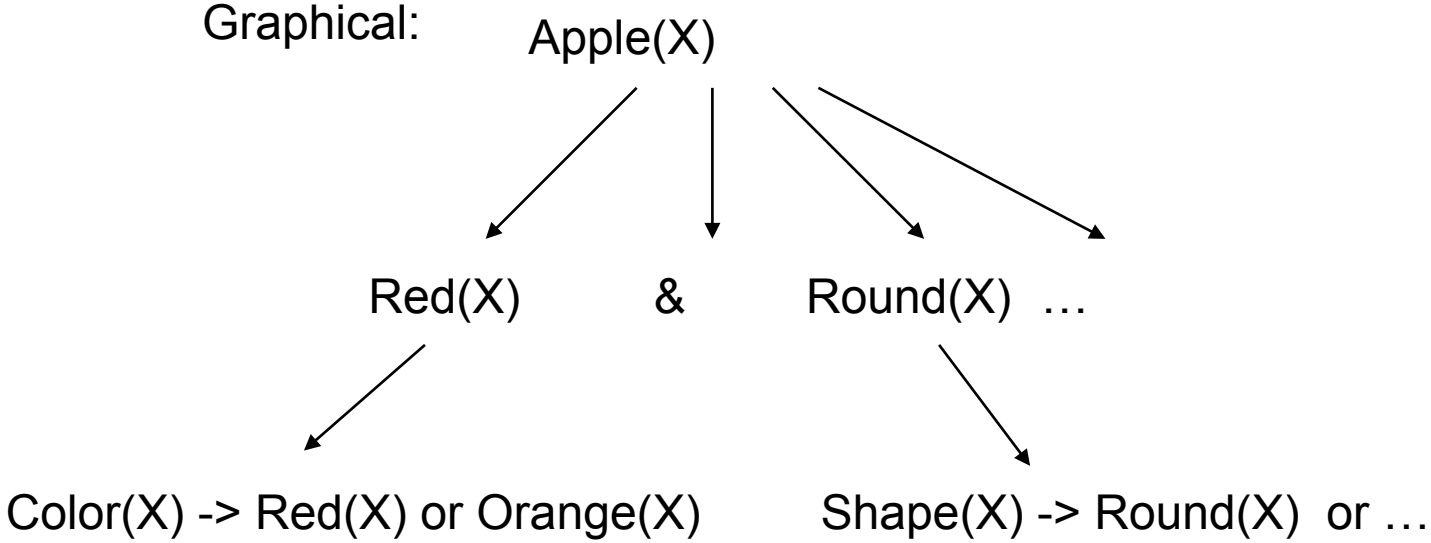
Program:

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

Output:

Apple(X)

Graphical:



- ACT-R (Anderson 2004)
- Cyc (openCyc)
  - 200k terms, 2m beliefs
- Never Ending Language Learning
  - since 2010 50m candidate beliefs

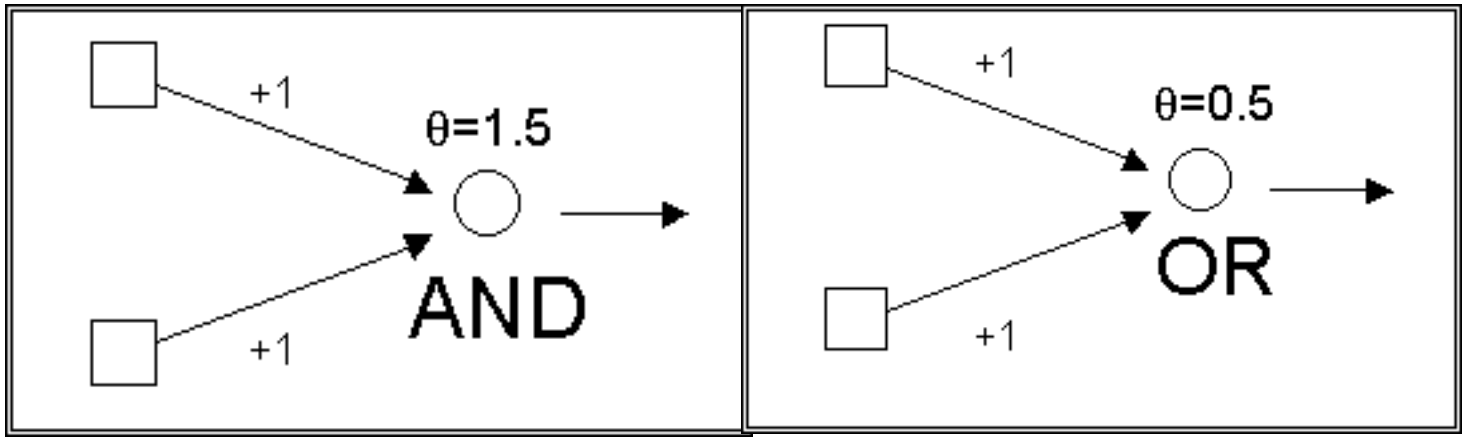
Connections

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

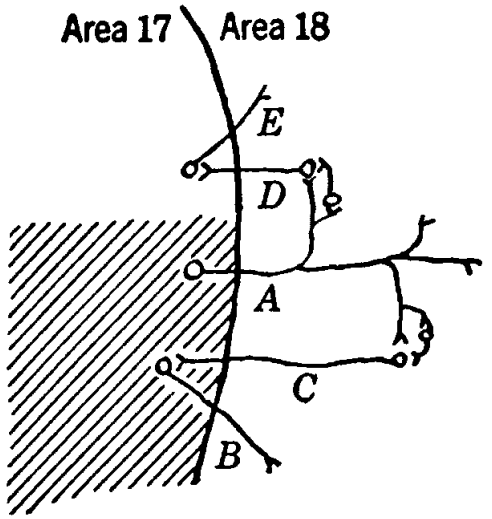
- McCulloch/Pitts neurons can then be used to compute any (finite) logical function



- 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 grown 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)

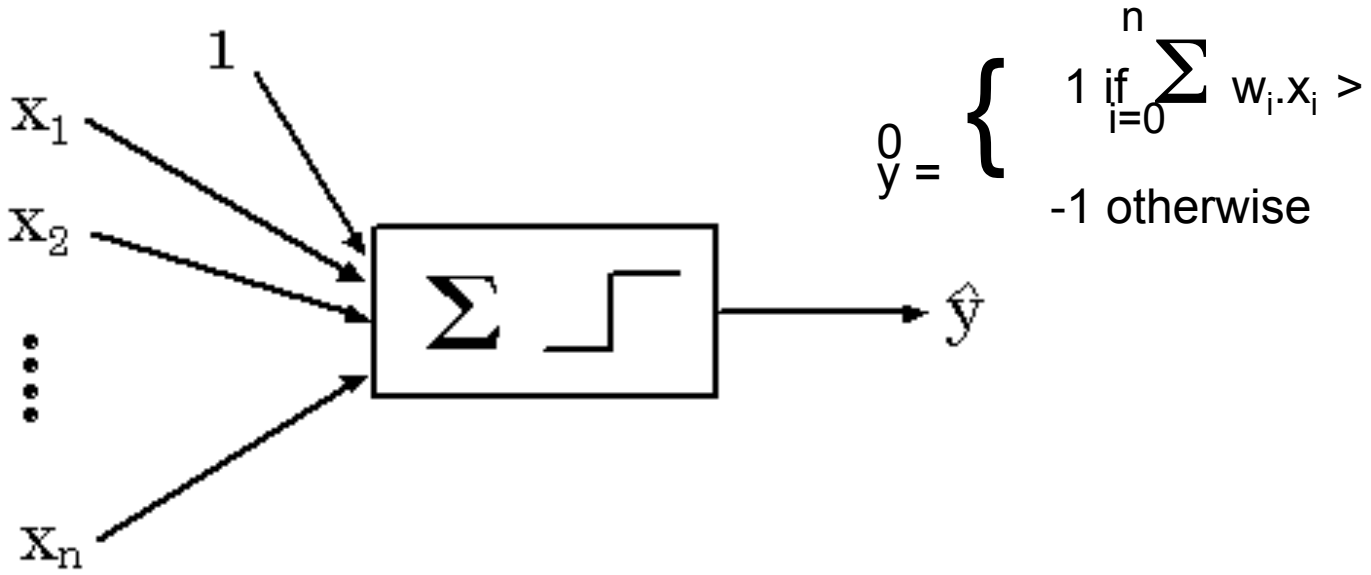
The "Hebb rule"

- $\Delta w_{ij} = \eta a_i a_j$
  - 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.
- Hebbian learning finds 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

-> *what is the associated computation?*

The perceptron
(Rosenblatt, 1958, 1962)

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



Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*. **65(6)**, 386-408.

Perceptron Learning Rule

- Start with random connections w_i
- Error-driven learning rule (delta rule):

  $\Delta w_i = \eta\ (t - y)\ x_i$

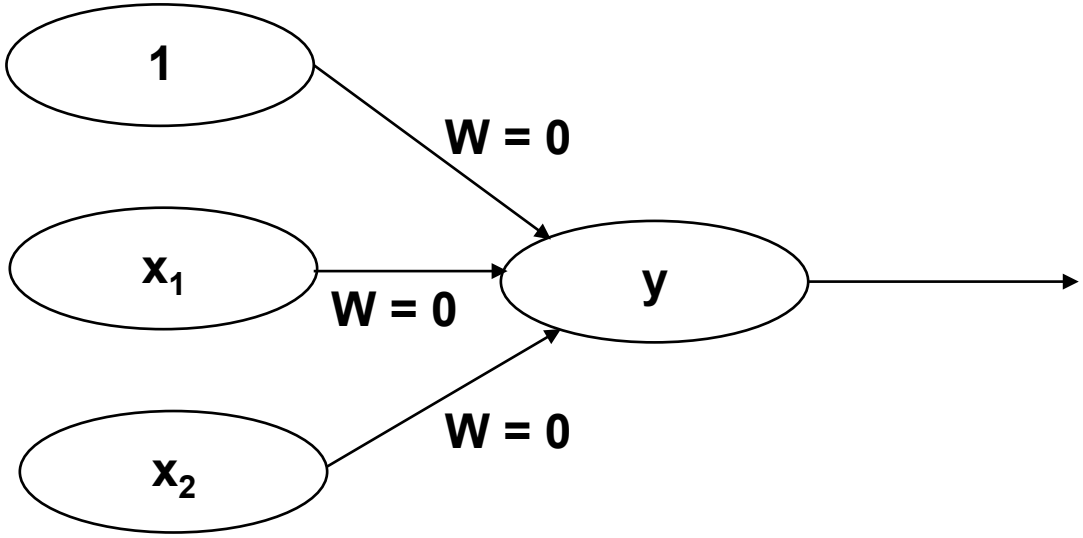
  t is the target value (given by the teacher)

  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

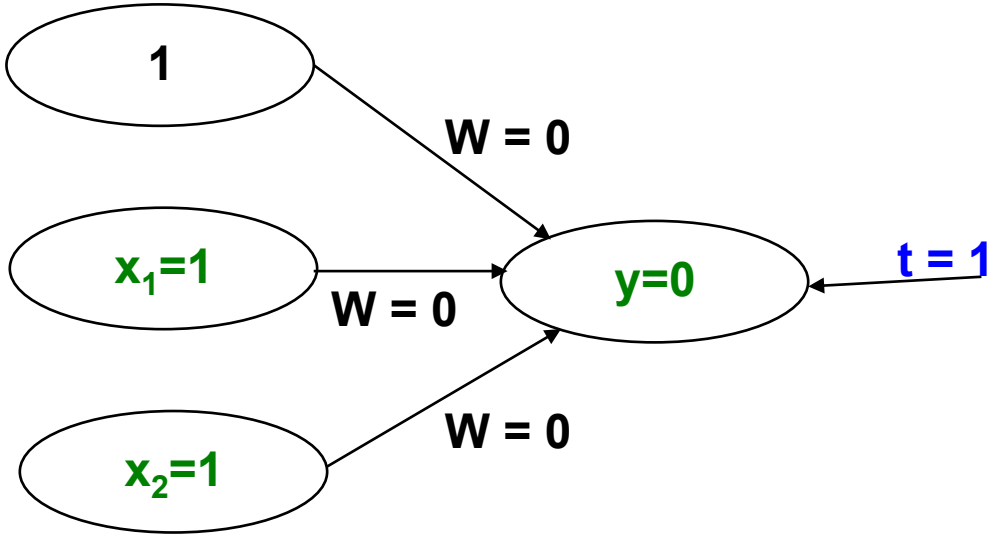


AND

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

- Initial weights: 0, n=0.25

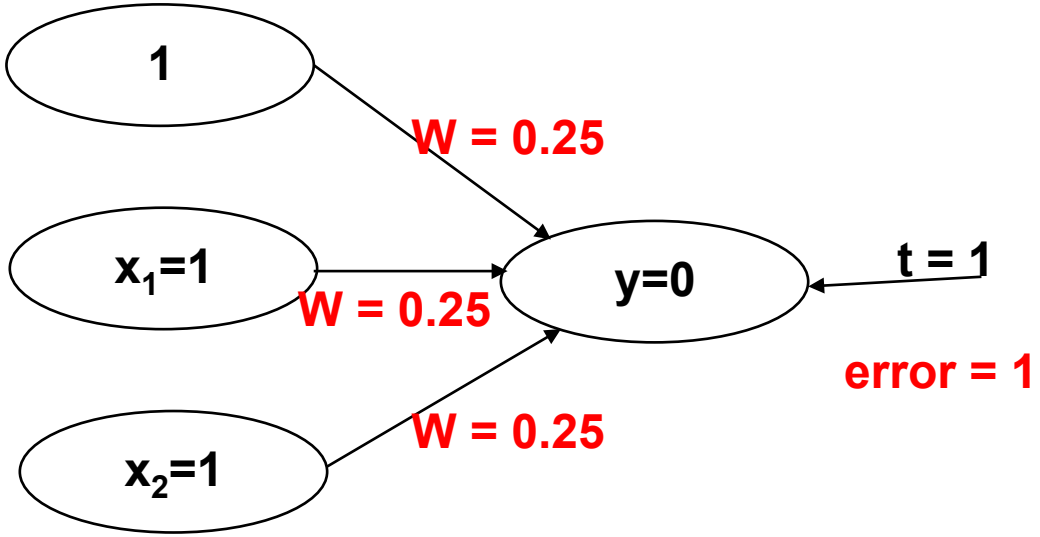
Example: learning the AND



AND

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

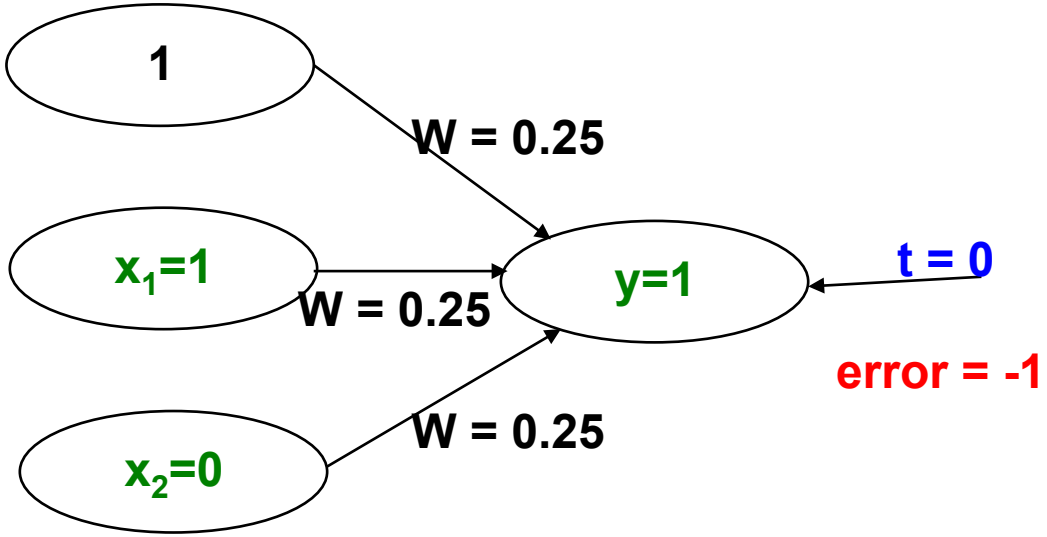
Example: learning the AND



AND

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

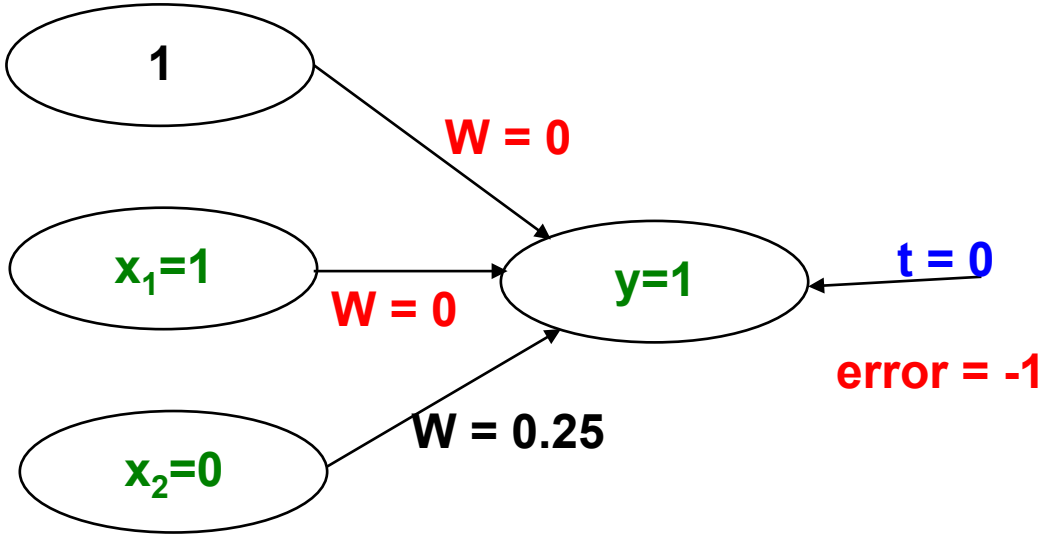
Example: learning the AND



AND

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

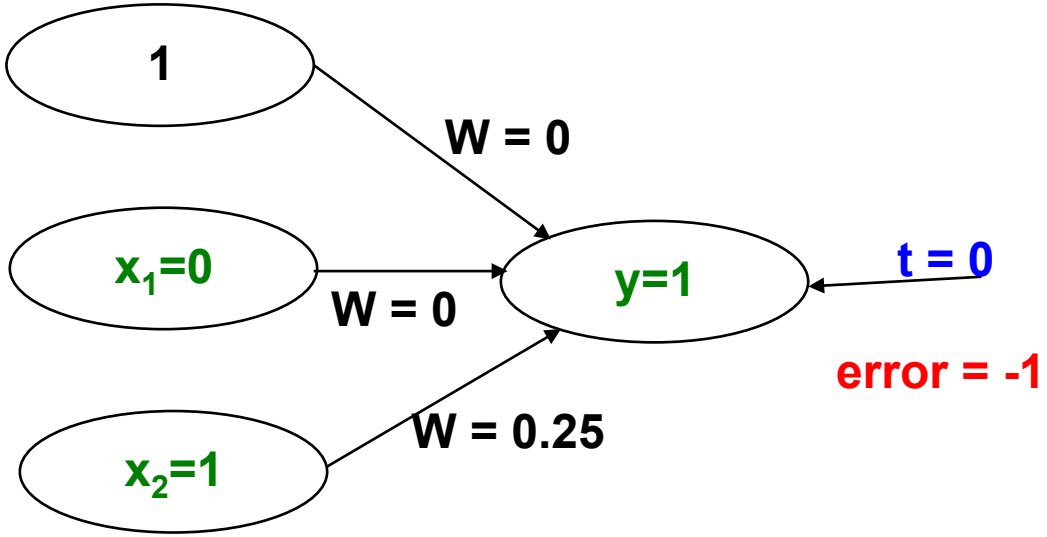
Example: learning the AND



AND

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

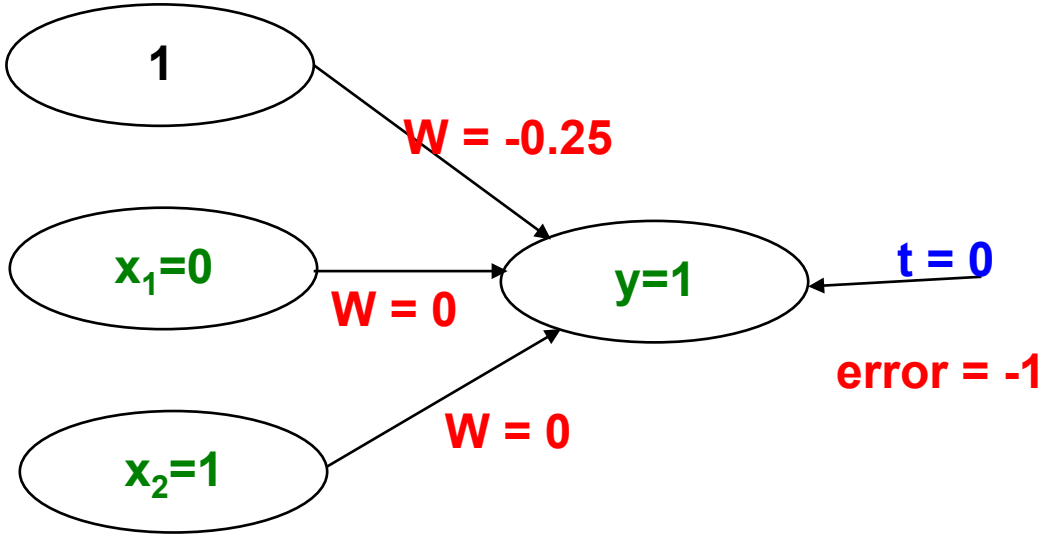
Example: learning the AND



AND

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

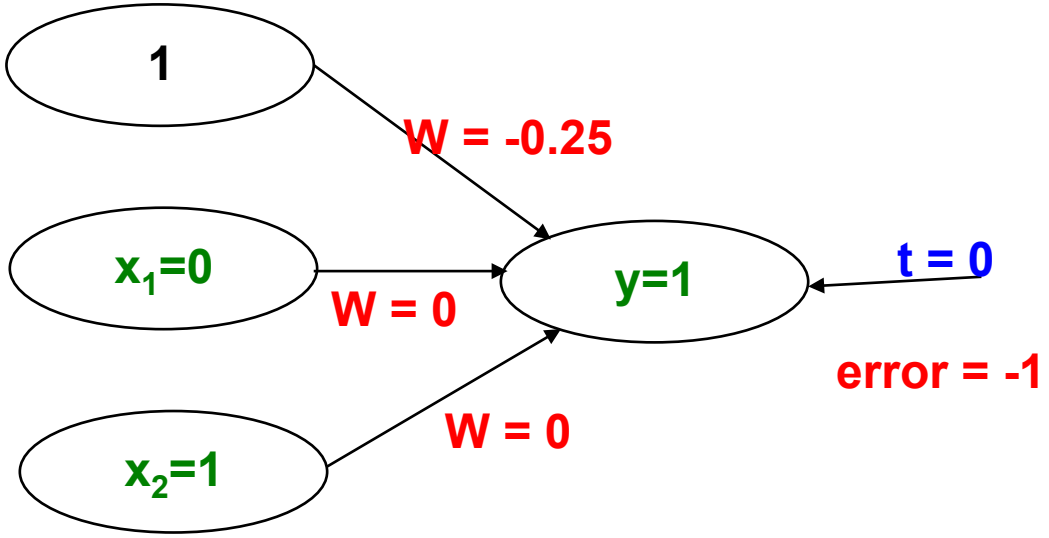
Example: learning the AND



AND

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

Example: learning the AND

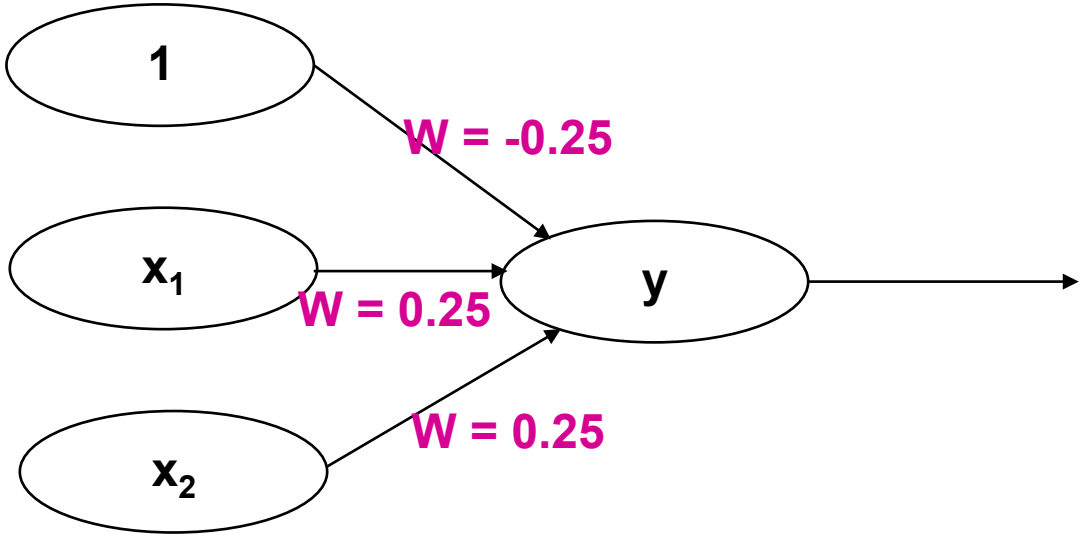


AND

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

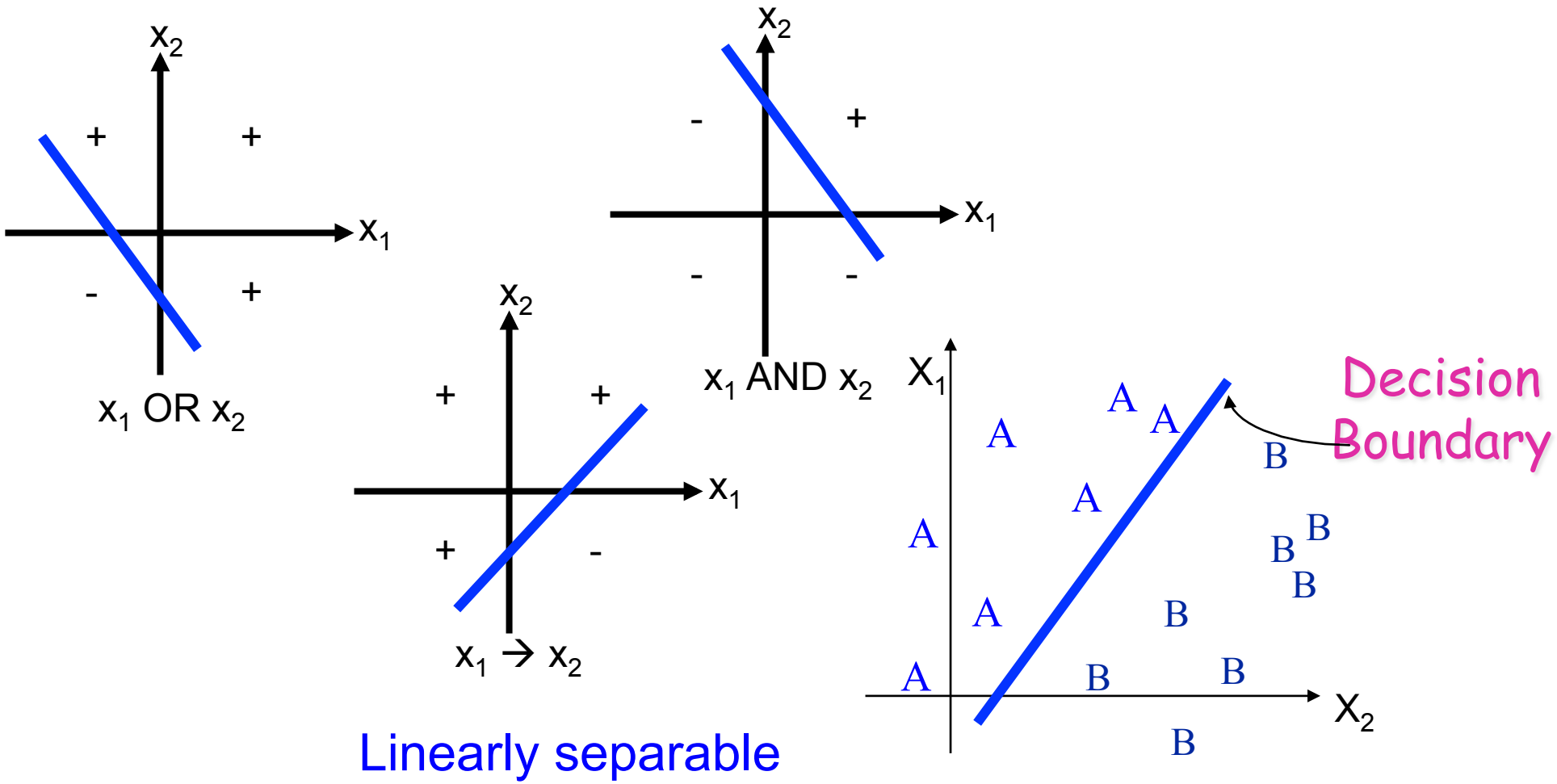


AND

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

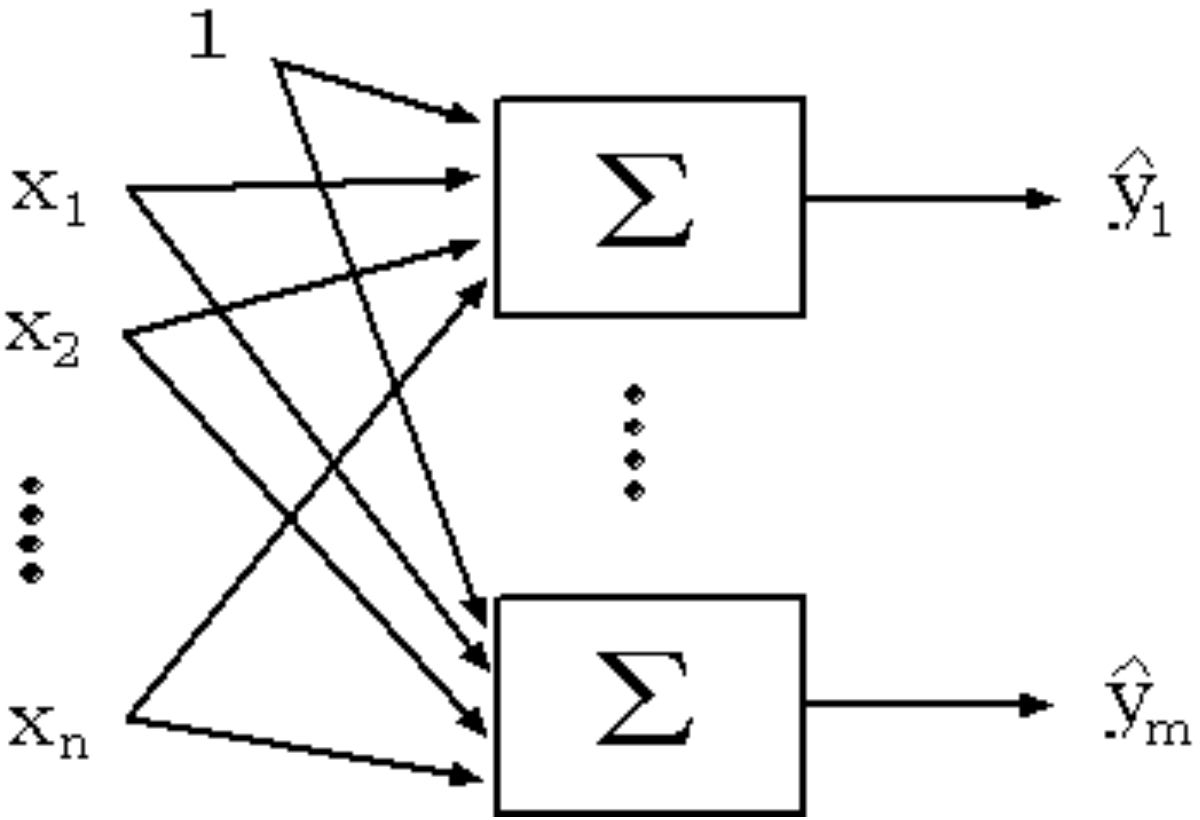
Final weights

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

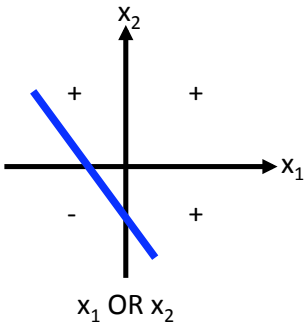


- similar multinomial LDA or multinomial regression

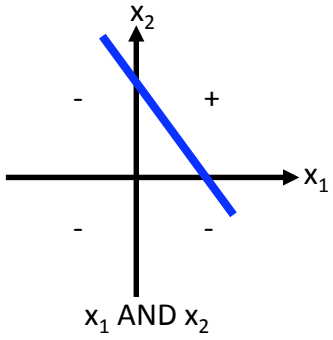
Minsky & Papert (1969)

- 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…*

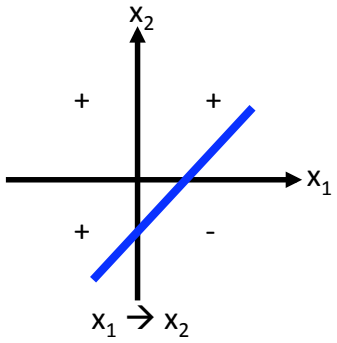
Decision Boundary of a Perceptron



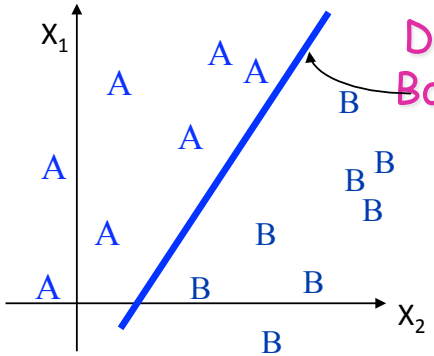
x_1 OR x_2



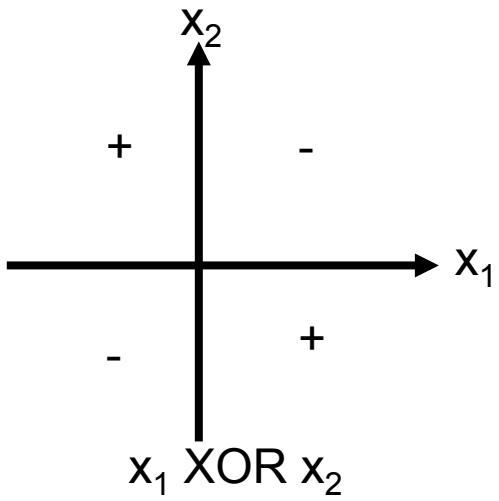
x_1 AND x_2



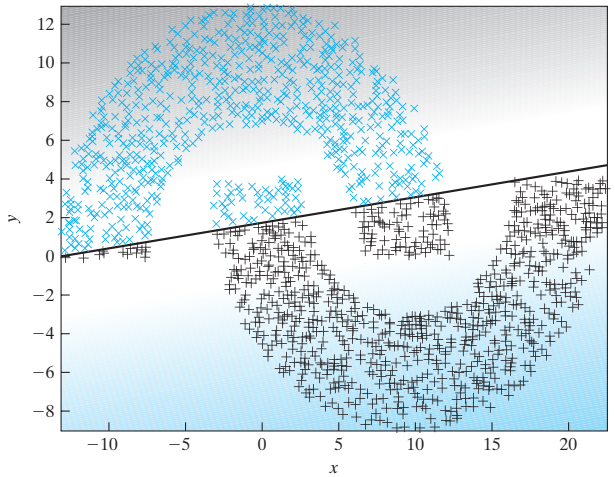
$x_1 \rightarrow x_2$



Linearly separable



x_1 XOR x_2



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

Consequences of Minsky & Papert's analysis

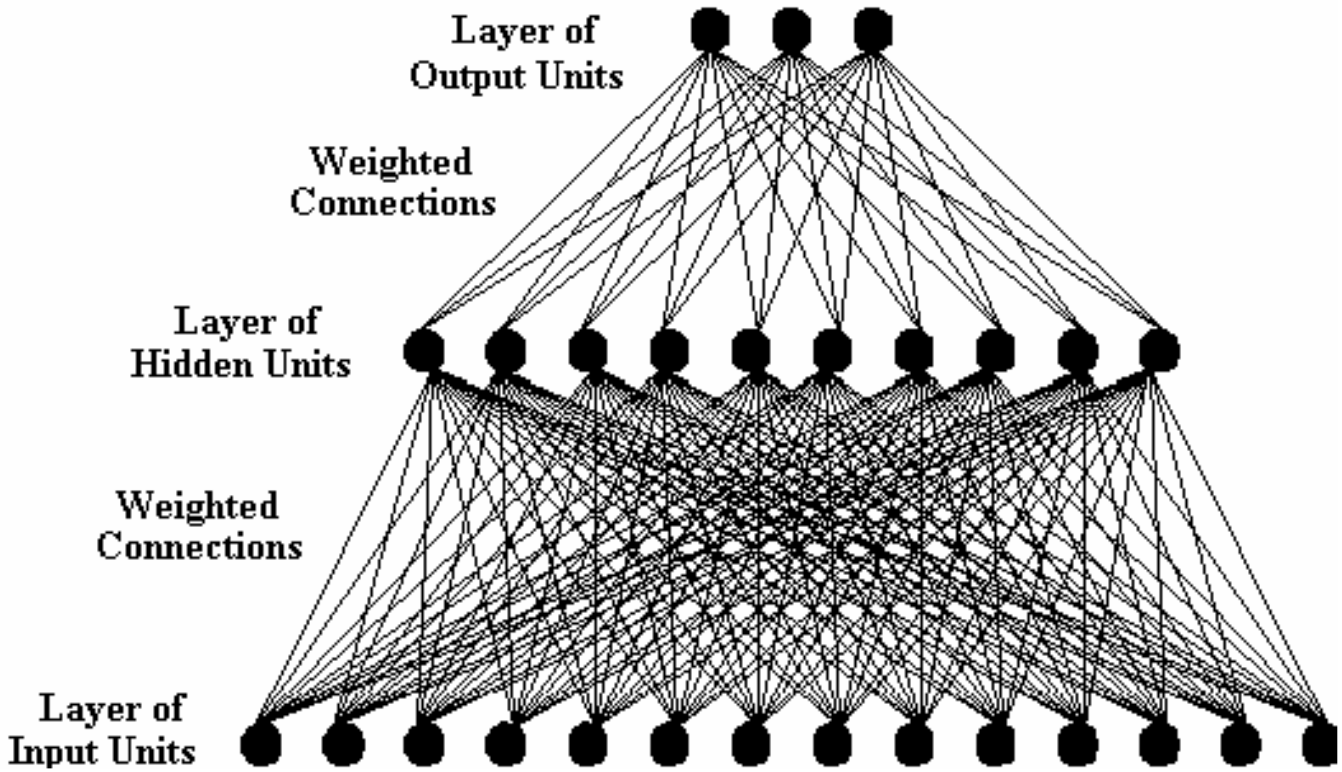
- nearly caused the death of this field.
- Subsequent research was not mainstream

The revival: the 80s

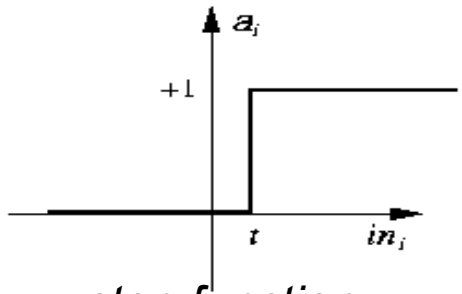
- multilayered feedforward networks
  - Generalization of the Delta Rule: backpropagation of error
  - learning of distributed representations
- Symmetric recurrent networks
  - winner take all
  - autoassociation

Rumelhart, David E.; Hinton, Geoffrey E., Williams, Ronald J. (1986). "Learning representations by back-propagating errors". Nature 323.
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

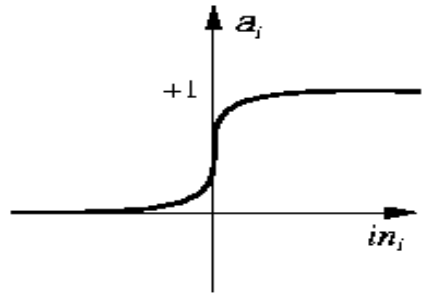
The revival I: the Multi-Layer Perceptron



Activation functions

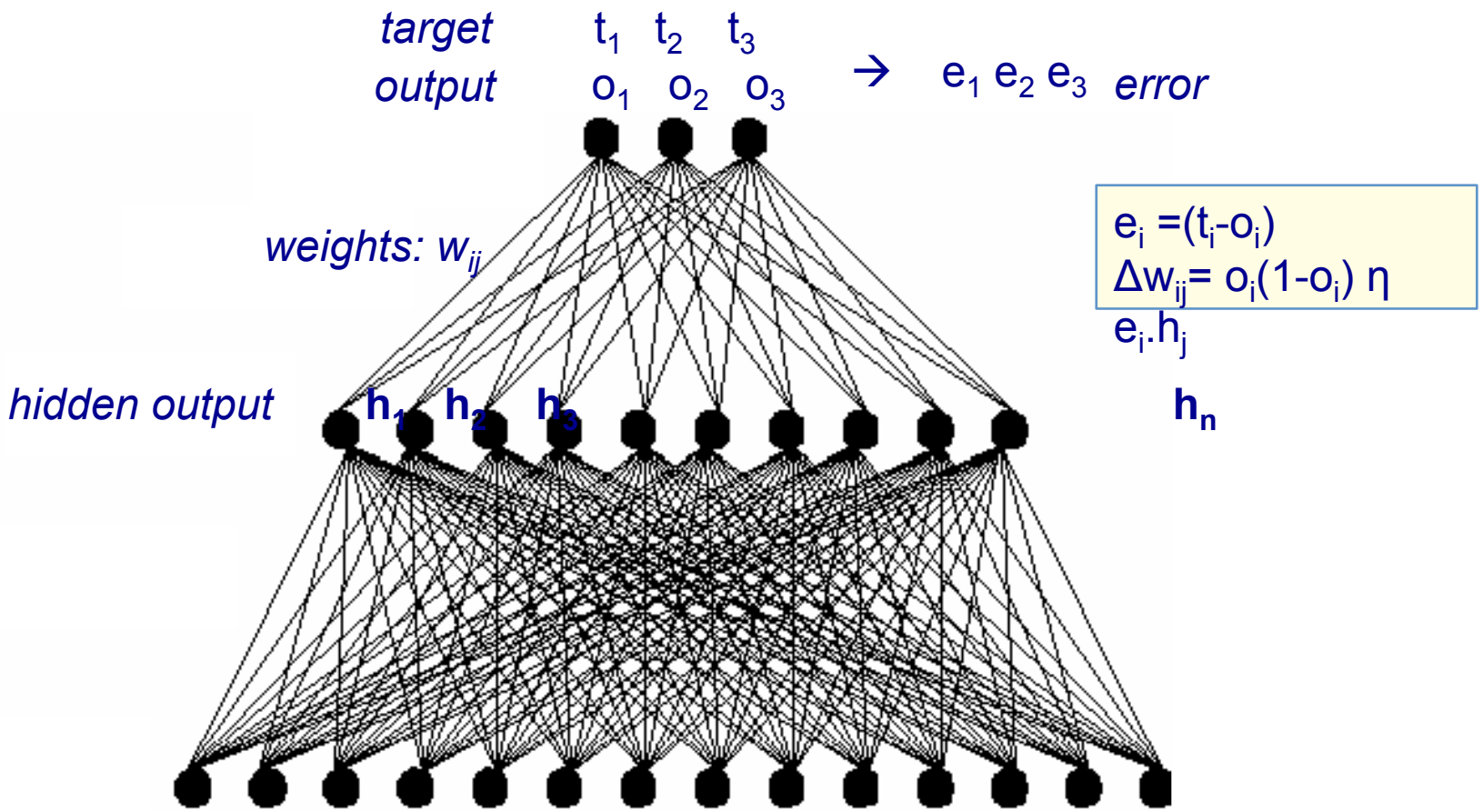


step function



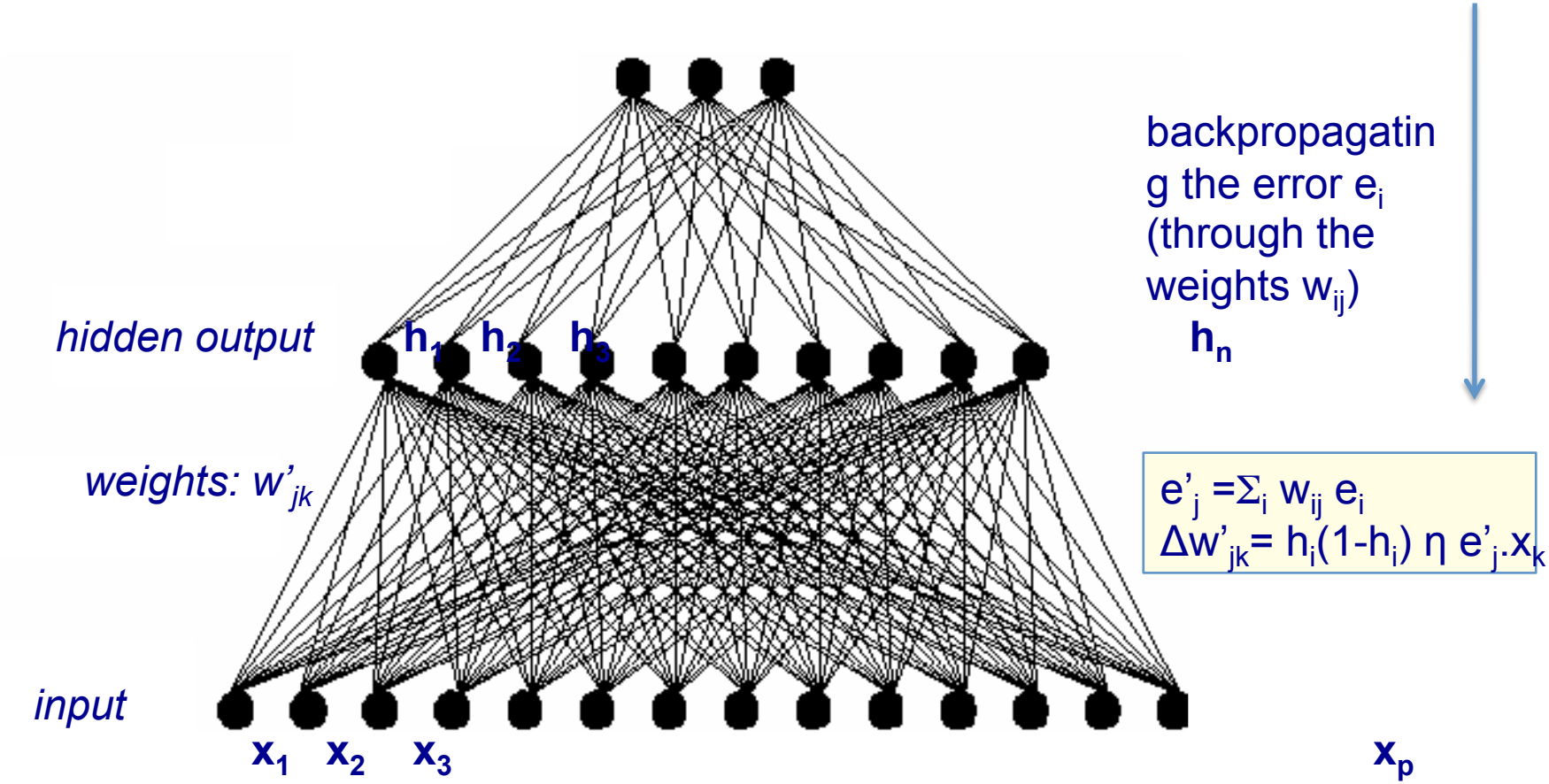
sigmoid function

The backpropagation learning rule (I)



Bryson,A.E, Ho, Y.-C. (1969). Applied optimal control: optimization, estimation, and control.

The backpropagation learning rule (II)



Expressive Capabilities of MLP

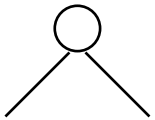
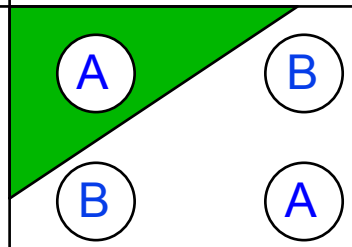
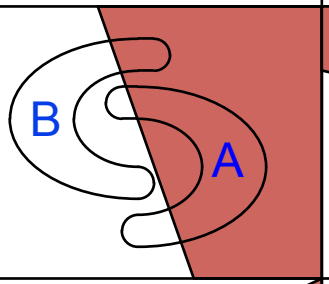
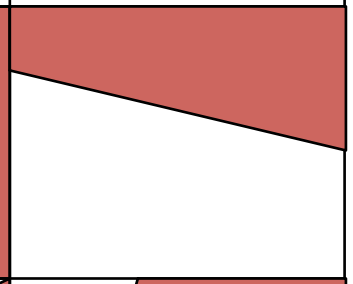
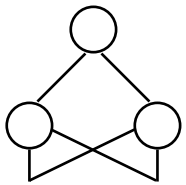
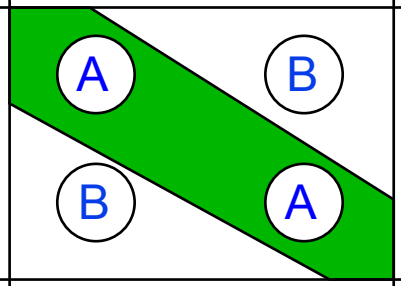
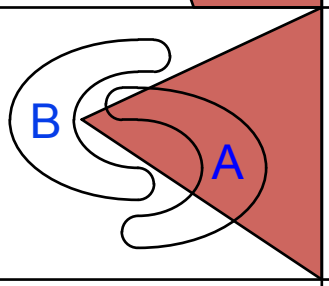
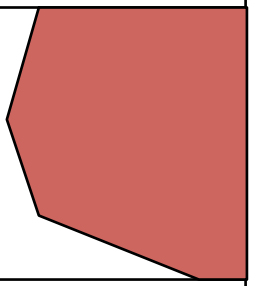
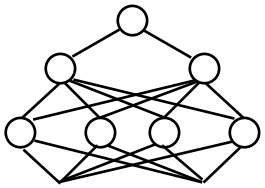
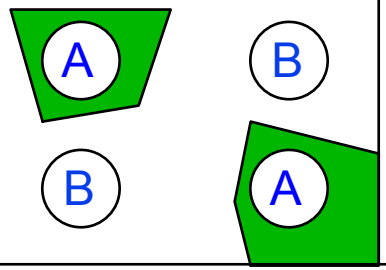
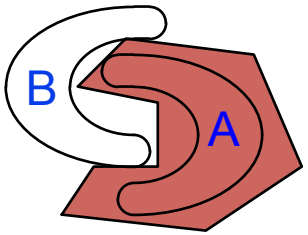
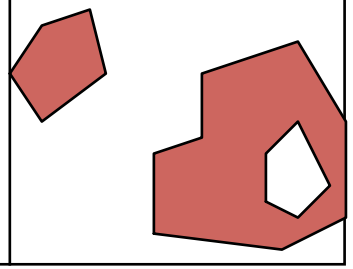
Boolean functions

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

Continuous functions

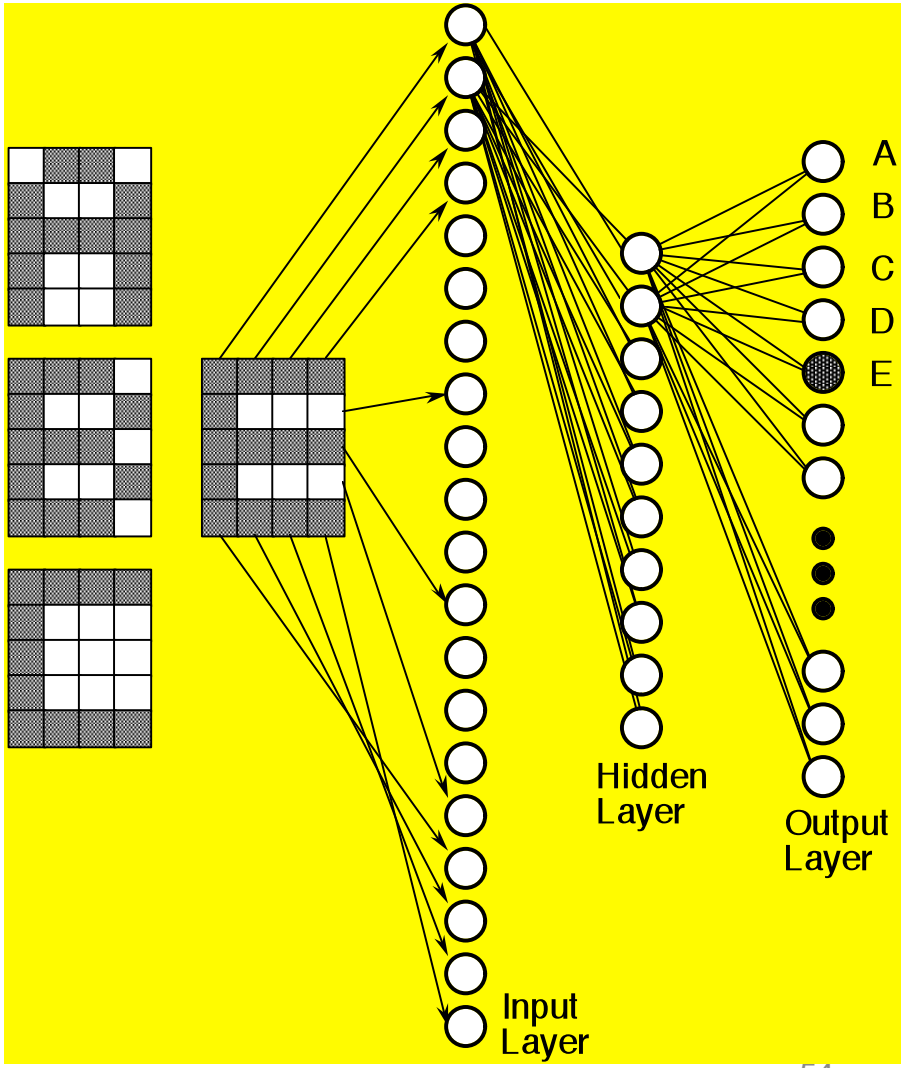
- Every bounded continuous function can be approximated with arbitrarily small error, by network with one hidden layer [Cybenko 1989, Hornik 1989]
- Any function can be approximated to arbitrary accuracy by a network with two hidden layers [Cybenko 1988]

Different Non-Linearly Separable Problems

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

Example: Neural network for OCR

- feedforward network
- trained using Back- propagation



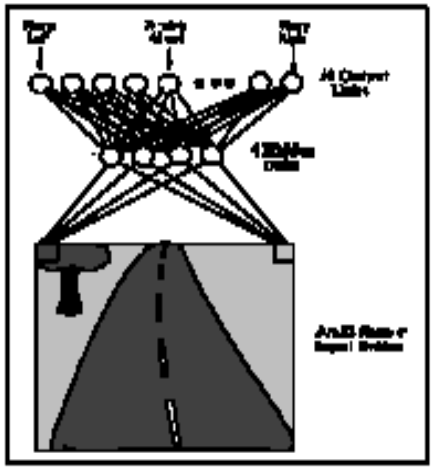
Example: ALVINN

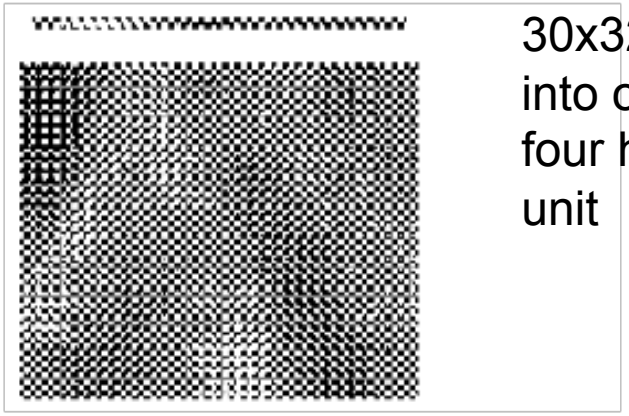
Drives 70 mph on a public highway

30 outputs for steering

4 hidden units

30x32 pixels as inputs

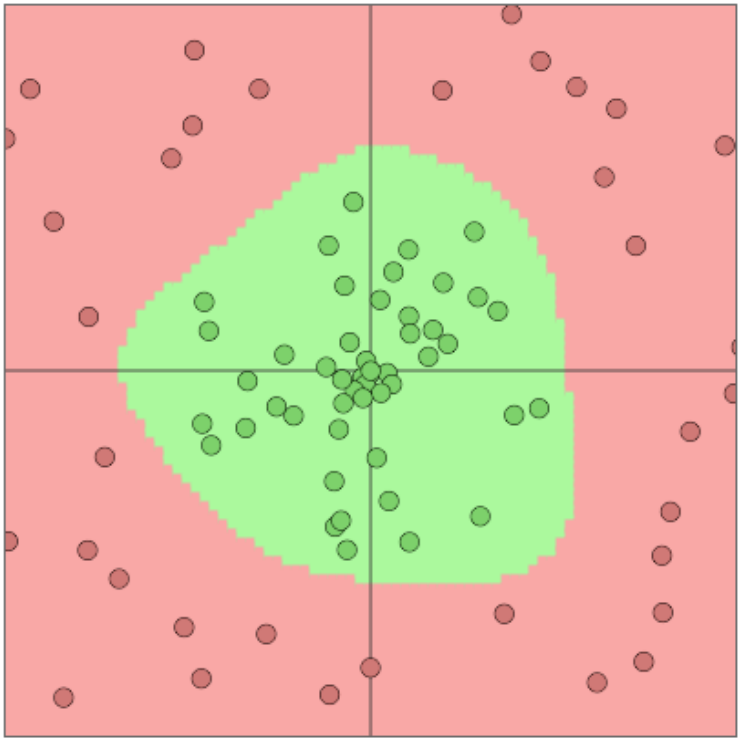


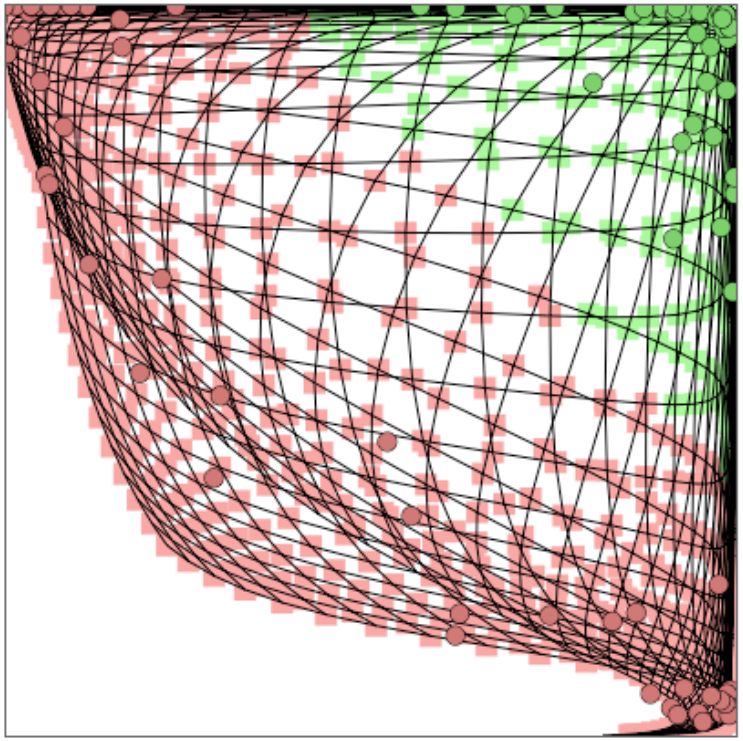


30x32 weights into one out of four hidden unit

https://www.youtube.com/watch?v=Rq_kBl0UZmY

http://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html



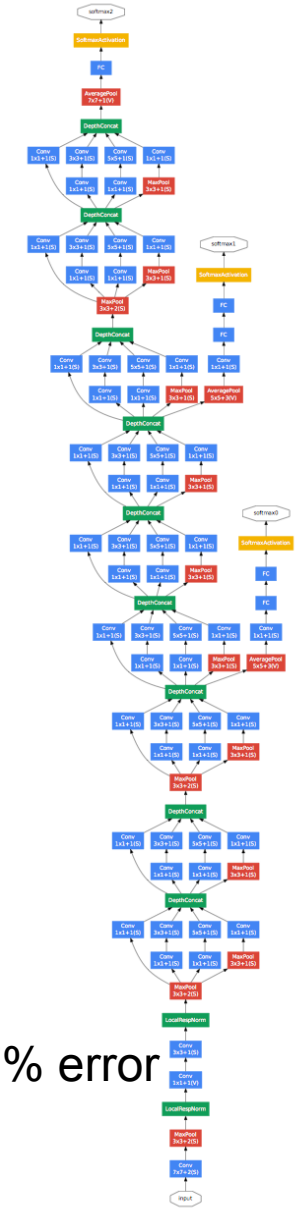


Current trends

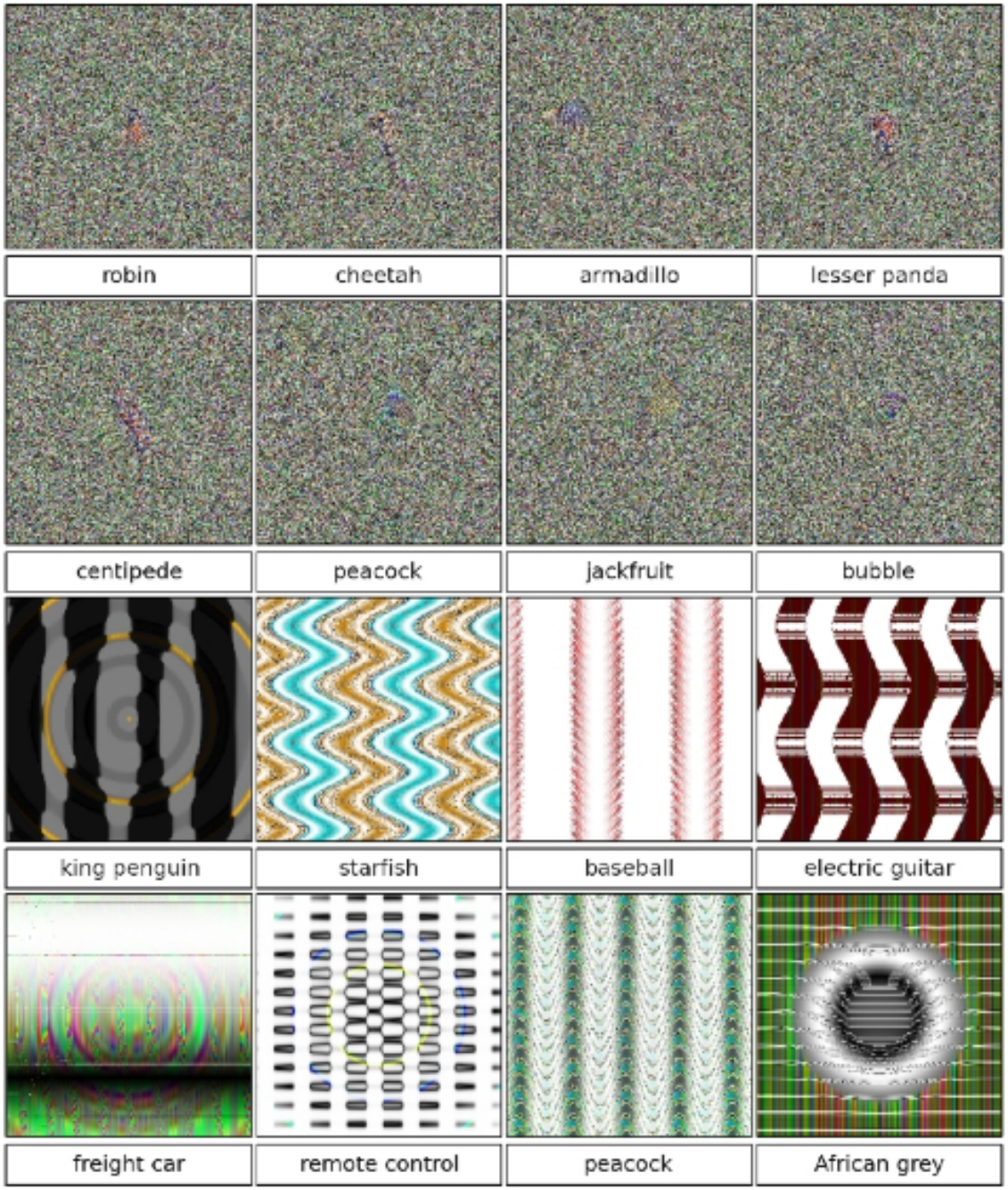
- very deep hierarchical networks
  - trained in a heavily supervised fashion
  - but, seem to correlate with human performance and monkey IT neurons

Yamins et al (2014) *Proc. Natl. Acad. Sci. U.S.A.*

1000 categories
100.000 test
1M training
Google net: 6.67% error
Humans: 5%



Google net,
Szegedy et al, 2014

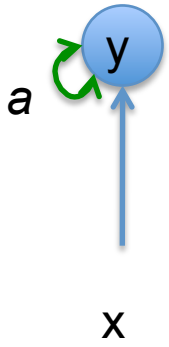

robin
cheetah
armadillo
lesser panda
centipede
peacock
jackfruit
bubble
king penguin
starfish
baseball
electric guitar
freight car
remote control
peacock
African grey

The revival II: Symmetric dynamical networks

discrete dynamical system: $Y_{t+1} = F(Y_t)$

where Y_t is the state of the system at time t

eg: a single neuron



$$y_{t+1} = a.y_t + x_t$$

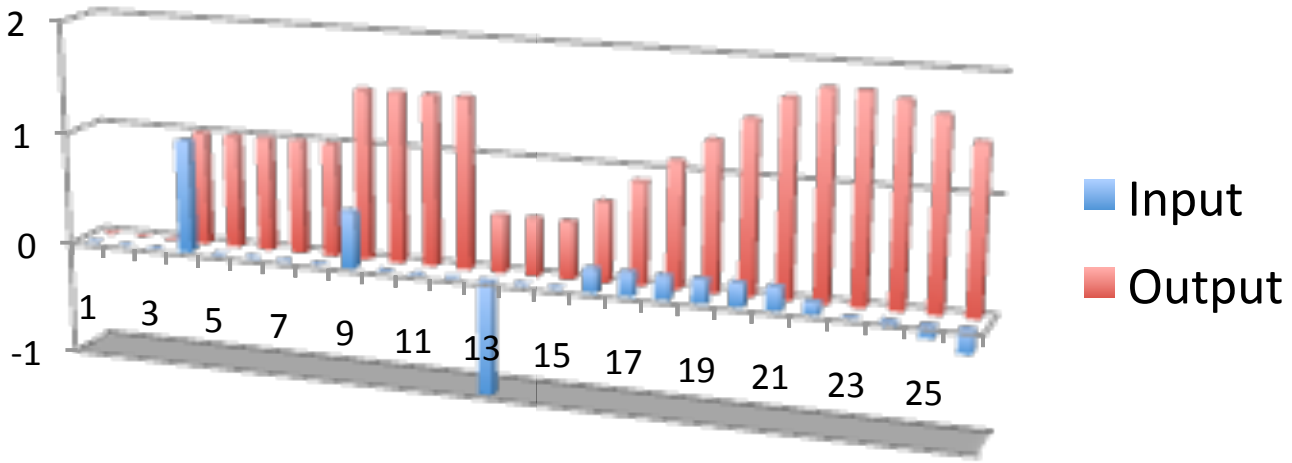
very different behaviors as a function of the value of a

- a=1: memory
- 0<a<1: leaky integration
- a=-1: oscillator
- -1<a<0: damped oscillator

$$y_{t+1}=a.y_t+x_t$$

a=1

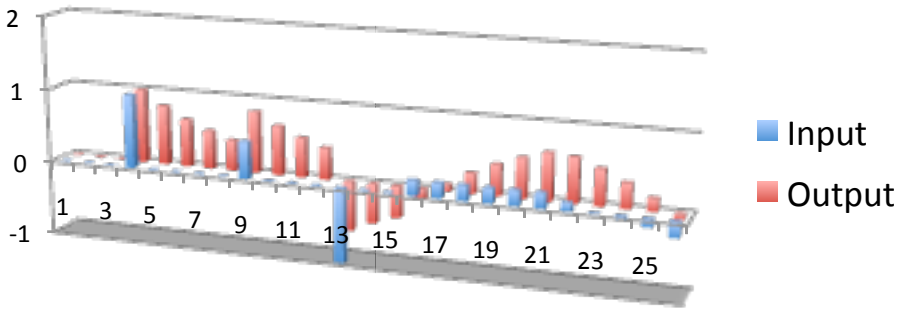
t	Input (X_t)	Output (Y_{t+1})
1	0	0
2	0	0
3	0	0
4	1	1
5	0	1
6	0	1
7	0	1
8	0	1
9	0.5	1.5
10	0	1.5
11	0	1.5



→ memory, integrator (counter)

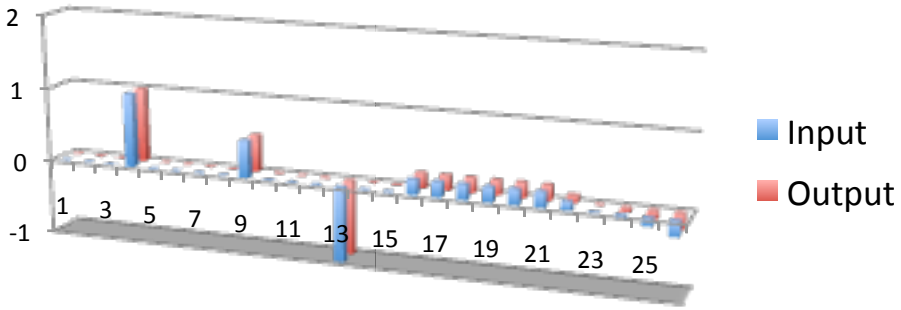
$$y_{t+1}=a.y_t+x_t$$

a=.7



→leaky integrator
→attractors (depending on input x/(1-a))

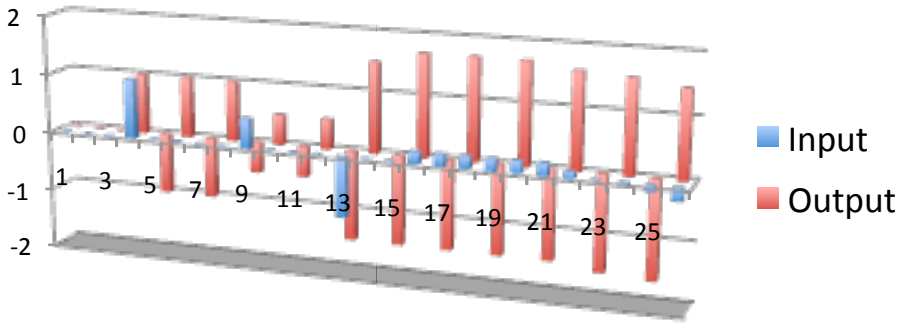
a=0



→copy (identity) operation

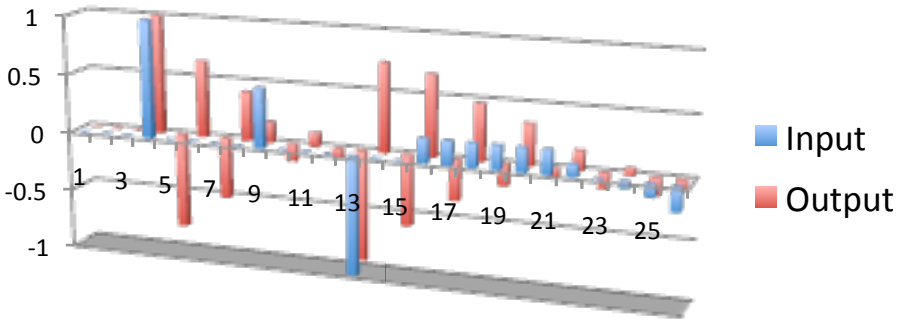
$$y_{t+1}=a.y_t+x_t$$

a=-1



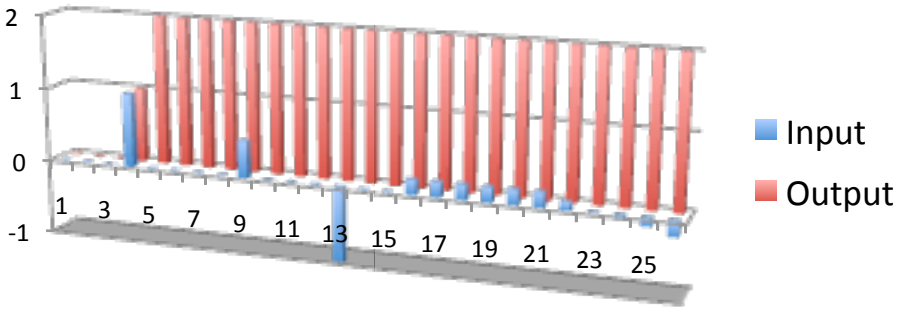
→oscillator

a=-.8



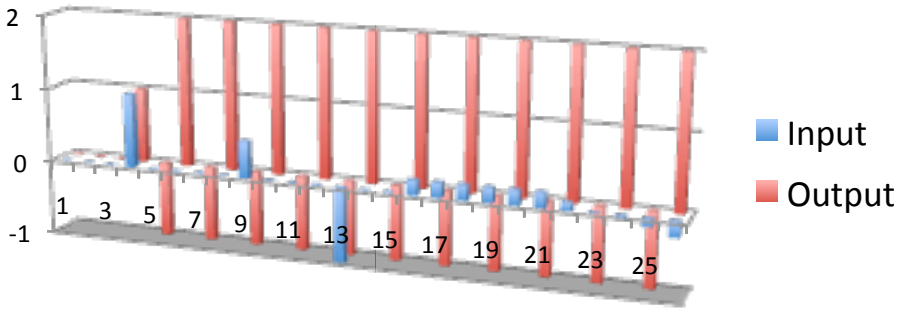
→damped oscillator

a=2



→saturating memory

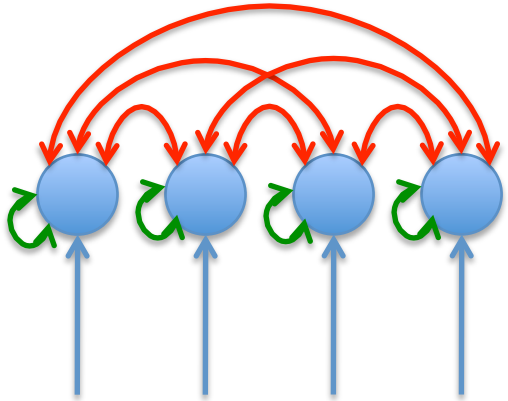
a=-2



→epileptic oscillator

The revival II: Symmetric dynamical networks

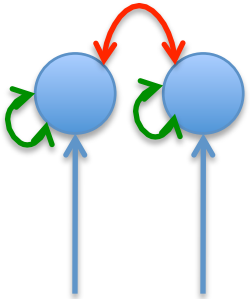
- A very simple example: a winner-take all network



inhibitory weights

excitatory weights

inputs



behaviors:
memorise
compute the max

- 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 & Sejnovsky, 1983)
  - neurons: binary (-1, 1)
  - network: symmetric weights
  - update: stochastic (1 with probability $1/ (1+e^{-sum(inputs)})$
  - 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 Bolzmann 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

Applications: early models of lexical access

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

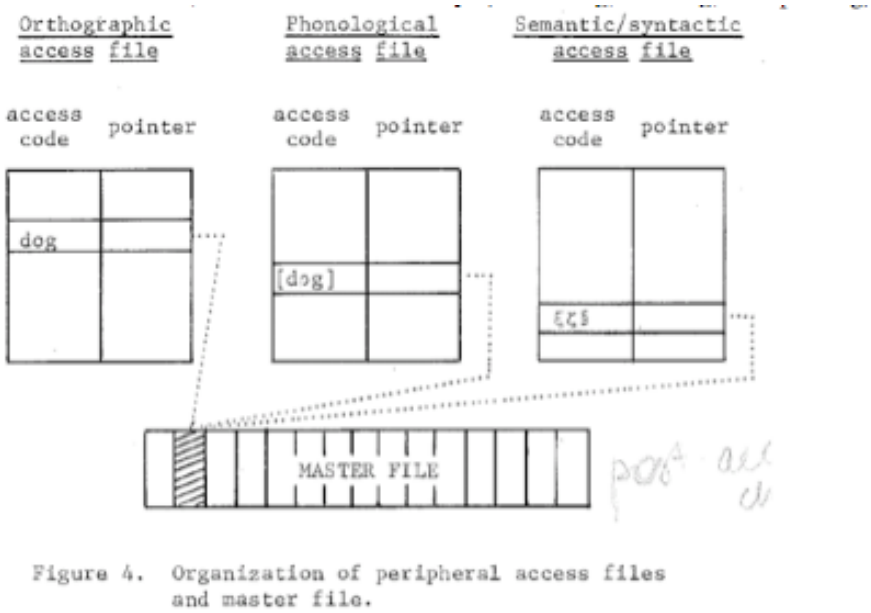
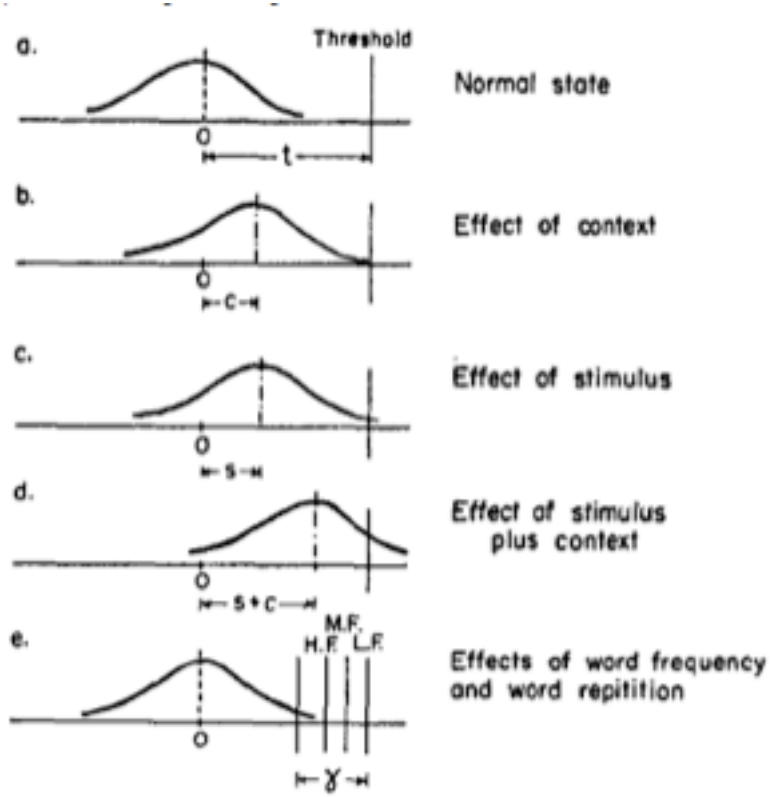


Figure 4. Organization of peripheral access files and master file.

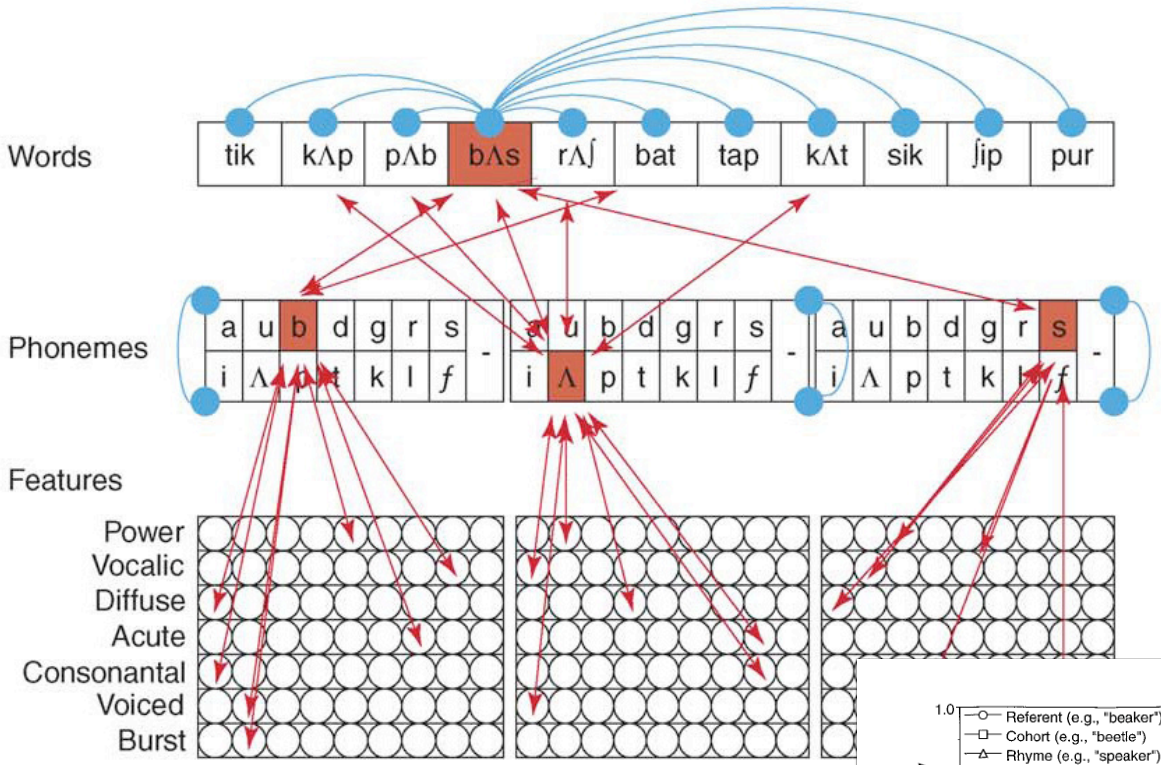


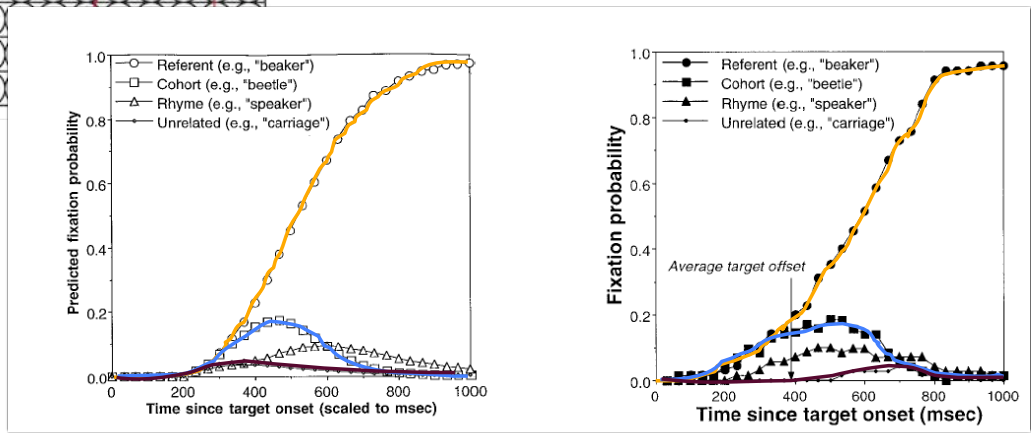
Forster, K. I. (1976). Accessing the mental lexicon. In , *New approaches to language mechanisms. Amsterdam: North-Holland.*

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

a new PDP model

- Elman & McClelland (1986)





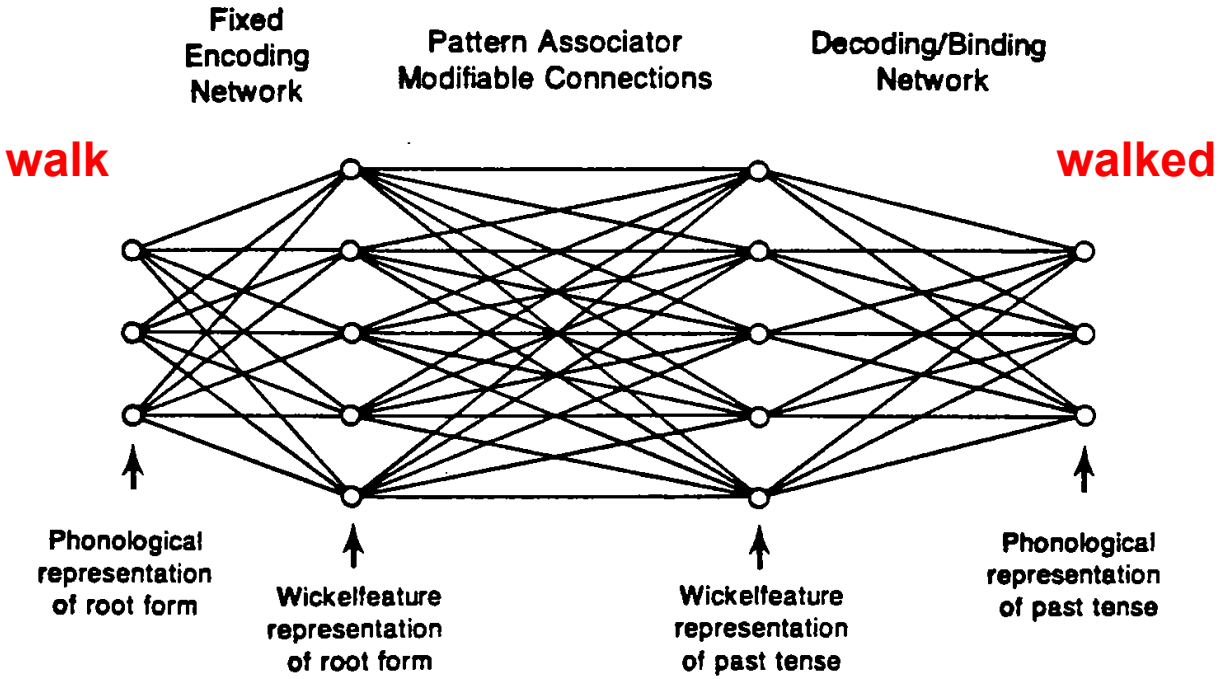
Allopena et al (1998)

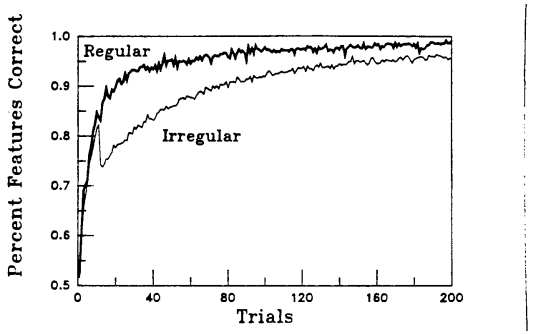
The revival III: McClelland & Rumelhart's (1986) Parallel Distributed Processing

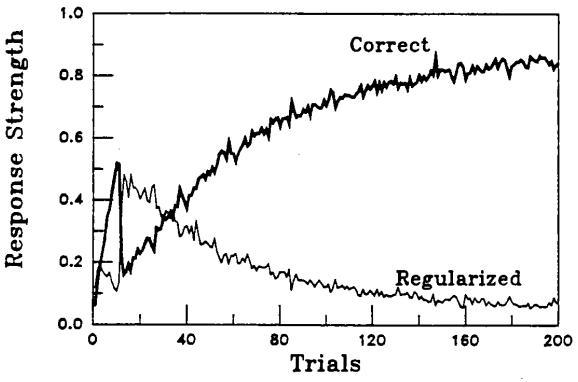
- Cognition involves the spreading of activation, relaxation, statistical correlation.
- Represents a method for how symbolic systems might be implemented
  - Hypothesized that apparently symbolic processing is an emergent property of subsymbolic operations.
- Advantages
  - Fault tolerance & graceful 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: connectionnist theories are inadequate models of language and thought







Rumelhart & McClelland (1986)

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: connectionnist theories are inadequate models of language and thought

Fodor & Pylyshyn

- Position of the problem: classical theories vs connectionnism
  - Agree:
    - both classical theories & connectionism are *representationalist*s (they assign some 'meaning' to the elements – symbols or nodes)
  - Disagree
    - classical theory encode structural relationships and processes (eg, constituents, variables, rules)
    - connectionnists only encore causal relationships and processes (x causes y to fire)
- Arguments against connectionnist 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
  - inadaquate for intuitive /nonverbal behavior
  - acutely sensitive to damage/noise (vs graceful degradation)
  - storage in classical systems is passive
  - inadaquate 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

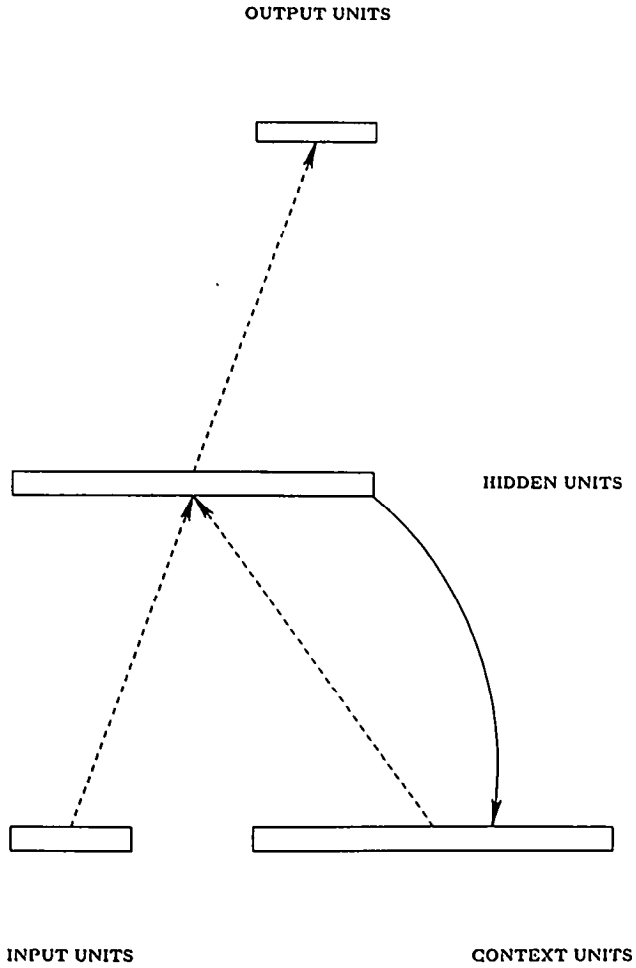
in brief

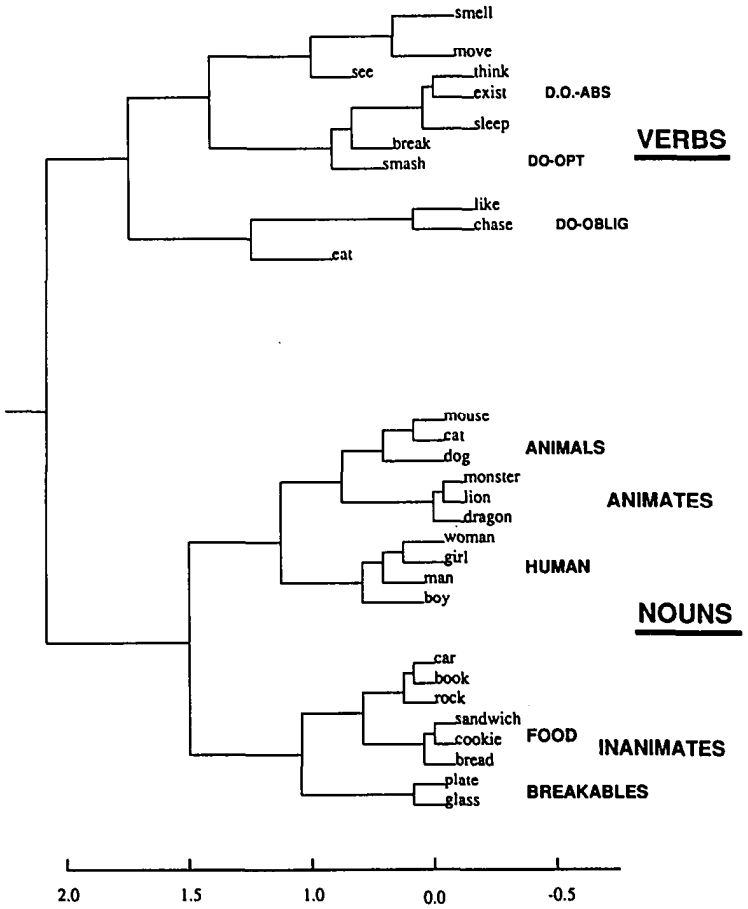
- The Fodor & Pylyshyn challenge:
  - (current) connectionist architectures fail to capture complex behaviors
  - (future) connectionist architectures are ‘mere’ implementation of symbolic architectures

Elman

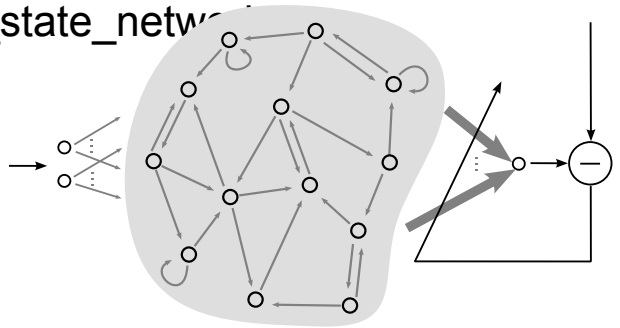
- structure of the paper
  - representing time
  - SRN architecture
  - xor through time
  - badiiguuu
  - word segmentation (15 words)
  - part of speech (13 categories, 29 words, 15 sentence templates)

Backprop applied





- extensions of Elman's SRN
  - computational capacity of SRN
    - Servan-Schreiber, D., Cleeremans, A., & McClelland, J.L. (1988). Encoding sequential structure in simple recurrent networks (CMU Tech. Rep. No. CMU-CS-88-183). Pittsburgh, PA: Carnegie-Mellon University, Computer Science Department.
    - Lawrence, S., Giles, C. L., & Fong, S. (2000). Natural language grammatical inference with recurrent neural networks. IEEE Transactions on Knowledge and Data Engineering , 12(1), 126–140.
    - Pollack, J. B. (1991). The induction of dynamical recognizers. Machine Learning , 7(2–3), 227–252. R
    - Rodriguez, P. (2001). Simple recurrent networks learn context-free and context-sensitive languages by counting. Neural Computation, 13(9).
  - reservoir computing
    - http://reservoir-computing.org
    - Jaeger H (2007) Echo state network. Scholarpedia 2(9): 2330. http://www.scholarpedia.org/article/Echo_state_netw

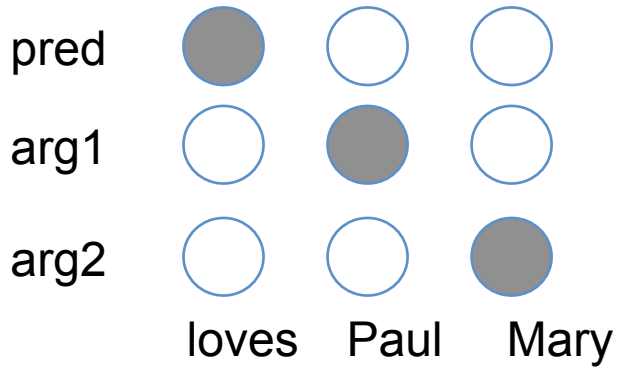


- structure of Smolensky
  - representing structures by fillers and roles
    - examples: trees, lists, etc
  - tensor products and filler/role binding (definition)
    - local, semilocal and distributed
  - unbinding (exact and selfadressed)
  - capacity and graceful saturation
  - continuous and infinite structures
  - binding and unbinding networks
  - analogy between binding units and hebb weights
  - example of a stack
  - structured roles

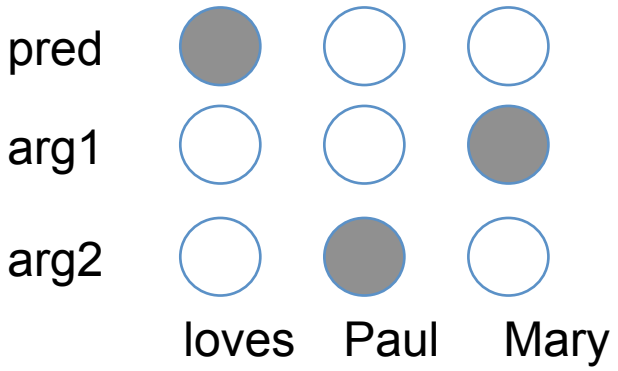
the Smolensky response: tensor products

Paul loves Mary-> loves(Paul,Mary)
pred=loves, arg1=Paul, arg2=Mary
pred*loves+arg1*Paul+arg2*Mary

Paul loves Mary
Mary is loved by Paul



Mary loves Paul
Paul is loved by Mary



→Problem of tensor product representations: exponential with sentence complexity

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structure in connectionnist systems. Artificial Intelligence, 46, 159-216.

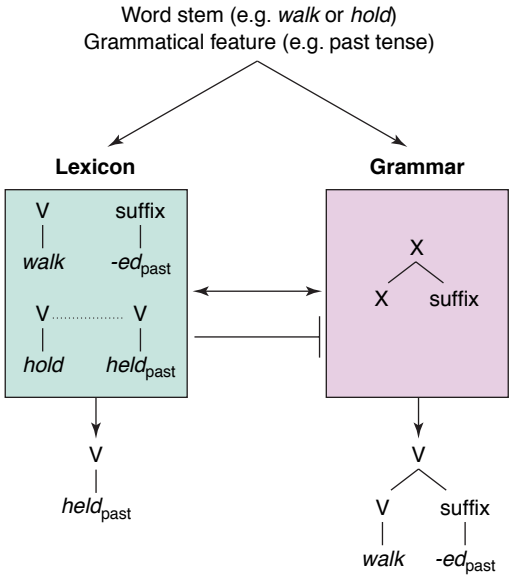
- extensions:
  - implementation of a phonological theory (Optimality Theory) in a tensor product network with energy relaxation
    - see the Harmonic Mind (Smolensky & Legendre)
  - Escaping the explosion in nb of neurons: holographic reduced representations
    - define A * B as an operation that preserves the dimensions (eg xor, circular convolution)

Conclusions

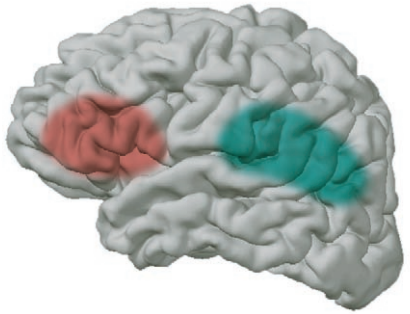
- What about the F&P Challenge?
  - tensor products are an interesting implementation/alternative to symbolic systems
  - recurrent networks could also be an alternative, but much less understood
- The hidden debate
  - innate vs learner structures (to be continued…)

Conclusions

- empirical impact of the debate
  - past tense in English
    - rule: play->played, fax->faxed
    - exceptions: sing->sang, put->put
    - Pinker & Prince (1988)
    - procedural vs declarative memory (Ullman et al, 1997; Pinker & Ullman, 2002)



	Lexicon	Grammar
Used for:	roots, idioms, irregulars, some regulars	phrases, sentences, any regular form
Form of computation:	lookup, association	combination, unification
Subdivision of:	declarative memory	procedural system
Associated with:	words, facts	rules, skills
Principal substrate:	temporo-parietal cortex	frontal cortex, basal ganglia



Pinker, S. & Prince, A. (1988) On language and connectionism *Cognition*, 28, 73-193.

Ullman MT, Corkin S, et al. (1997). A neural dissociation within language: *Journal of Cognitive Neuroscience*, 9: 266–276.

Pinker, S. & Ullman, M. (2002) The past and future of the past tense. *Trends in Cognitive Science*, 6, 456-463.

Conclusions

- empirical impact of the debate (cont)
  - statistical learning vs algebraic learning in infants
    - Saffran et al, (1996), Marcus et al, (1999), Pena et al (2002)

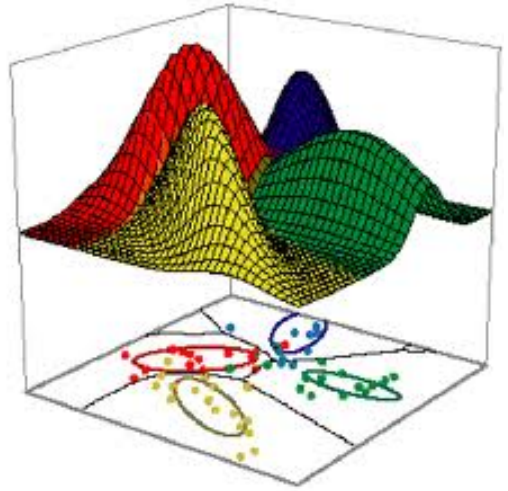
Saffran, J., Aslin,R., Newport, E. (1996). *Science*, 274,1926.
Marcus, G.F., Vijayan, S., Bandi Rao, S., Vishton, P. M. (1999). *Science*, 283, 77
Peña, M., Bonatti, L., Nespor, M., Mehler J. (2002). Signal-Driven Computations in Speech Processing, Science, 298, pp. 604-607.

  - exemplar-based versus abstract representations
    - object recognition (Biederman & Gerhardstein, 1993), face recognition, speech recognition (eg Goldinger, 1988; Johnson 1997, Pierrehumbert 2001)

Biederman, I. & Gerhardstein, P. C. (1993). Recognizing depth-rotated objects: Evidence and conditions for 3D viewpoint invariance. Journal of Experimental Psychology: Human Perception and Performance, 19, 1162-1182.
Johnson, K. (1997). Speech perception without speaker normalization: An exemplar model. In K. Johnson & J.W. Mullennix (eds.), Talker Variability in Speech Processing, pp. 145-165. San Diego: Academic Press.
Pierrehumbert, J. (2001). Exemplar dynamics: Word frequency, lenition and contrast. In J. Bybee and P. Hopper (eds.), Frequency and the Emergence of Linguistic Structure, pp. 137-157. Amsterdam: Benjamins.
Goldinger, S.D. (1998). Echoes of echoes? an episodic theory of lexical access. *Psychological Review*

Extensions

- computational reduction: finding the right architecture
- other connectionnist architectures
  - Kohonen's maps (competitive learning) (Kohonen, 1982)
  - Adaptive Resonance Theory (Grossberg, 1976)
  - Reinforcement learning (Barto, Sutton, Anderson. 1983)
- other computational frameworks
  - Probabilistic/Bayesian frameworks
  - Predictive Coding/Free Energy



Papiers

- Fodor & Pylyshyn (1988) Connectionism and cognitive architecture: a critical analysis
- Elman (1990) Finding structure in time
- Smolensky (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems

- Marcus (1998). Rethinking eliminative connectionism
- McClelland (2009). The place of modeling in Cognitive Science

Fodor & Pylyshyn

- Position of the problem: classical theories vs connectionnism
  - Agree:
    - both classical theories & connectionism are *representationalist*s (they assign some 'meaning' to the elements – symbols or nodes)
  - Disagree
    - classical theory encode structural relationships and processes (eg, constituents, variables, rules)
    - connectionnists only encore causal relationships and processes (x causes y to fire)
- Arguments against connectionnist 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
  - inadaquate for intuitive /nonverbal behavior
  - acutely sensitive to damage/noise (vs graceful degradation)
  - storage in classical systems is passive
  - inadaquate 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