Psych 215: Language Sciences (Language Acquisition)

Lecture 15

Poverty of the Stimulus II

Seidenberg (1997): Innate Biases ≠ Grammatical Knowledge

The Standard Theory, according to Chomsky

Big Questions of Language Acquisition:

What constitutes knowledge of language?

How is this knowledge acquired?

How is this knowledge used?

Seidenberg (1997): Innate Biases ≠ Grammatical Knowledge

Knowledge of language, according to Chomsky

Knowledge of language = grammar

Grammar = complex set of rules and constraints that gives speakers intuitions that some sentences belong in the language while others do not

Competence Hypothesis: Grammar is separate from “performance factors”, like dysfluencies (she said…um..wrote that), errors (I bringed it), memory capacity (The boy that the dog that the cat chased bit ran home.), and statistical properties of language (frequency of transitive (Sarah ate the peach) vs. intransitive use (Sarah ate))

“I think we are forced to conclude that…probabilistic models give no particular insight into some of the basic problems of syntactic structure” - Chomsky, 1957

Seidenberg (1997): Innate Biases ≠ Grammatical Knowledge

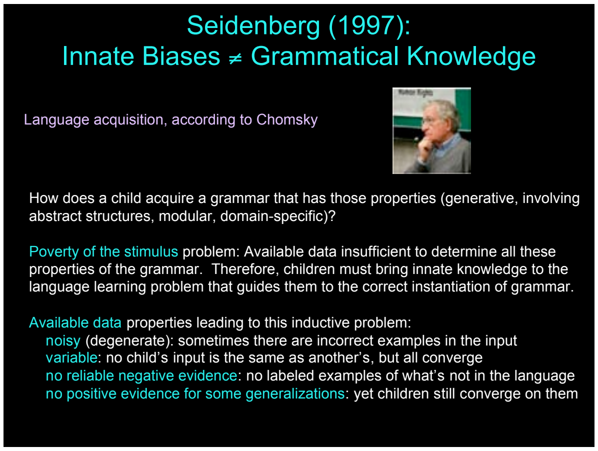
Properties of language, according to Chomsky

Grammar is generative: it can be used to produce and comprehend an infinite number of sentences

Grammar involves abstract structures: information that speakers unconsciously used is not overtly available in the observable data

Grammar is modular: there are separate components with different types of representations governed by different principles

Grammar is domain-specific: language exhibits properties not seen in other areas of cognition, so it cannot be the product of our general ability to think and learn



Seidenberg (1997): Innate Biases ≠ Grammatical Knowledge

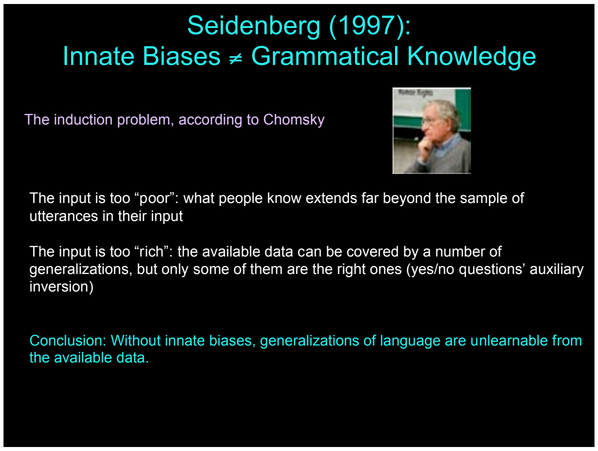
Language acquisition, according to Chomsky

How does a child acquire a grammar that has those properties (generative, involving abstract structures, modular, domain-specific)?

Poverty of the stimulus problem: Available data insufficient to determine all these properties of the grammar. Therefore, children must bring innate knowledge to the language learning problem that guides them to the correct instantiation of grammar.

Available data properties leading to this inductive problem:

- noisy (degenerate): sometimes there are incorrect examples in the input
- variable: no child's input is the same as another's, but all converge
- no reliable negative evidence: no labeled examples of what's not in the language
- no positive evidence for some generalizations: yet children still converge on them



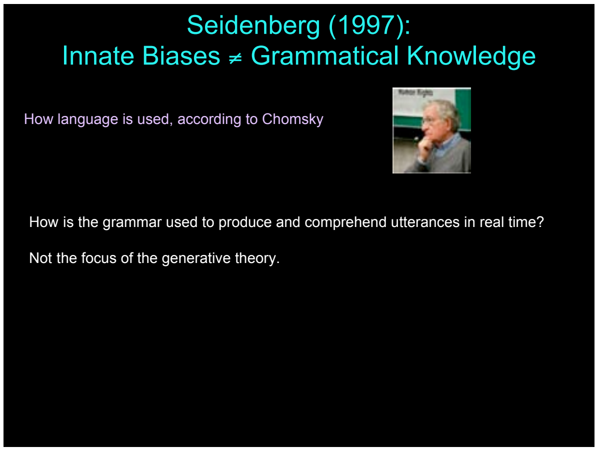
Seidenberg (1997): Innate Biases ≠ Grammatical Knowledge

The induction problem, according to Chomsky

The input is too "poor": what people know extends far beyond the sample of utterances in their input

The input is too "rich": the available data can be covered by a number of generalizations, but only some of them are the right ones (yes/no questions' auxiliary inversion)

Conclusion: Without innate biases, generalizations of language are unlearnable from the available data.

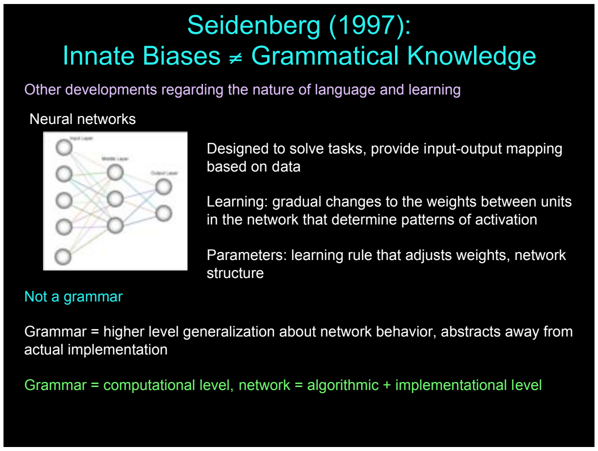


Seidenberg (1997): Innate Biases ≠ Grammatical Knowledge

How language is used, according to Chomsky

How is the grammar used to produce and comprehend utterances in real time?

Not the focus of the generative theory.



Seidenberg (1997): Innate Biases ≠ Grammatical Knowledge

Other developments regarding the nature of language and learning

Neural networks

Input Layer
Middle Layer
Output Layer

Designed to solve tasks, provide input-output mapping based on data

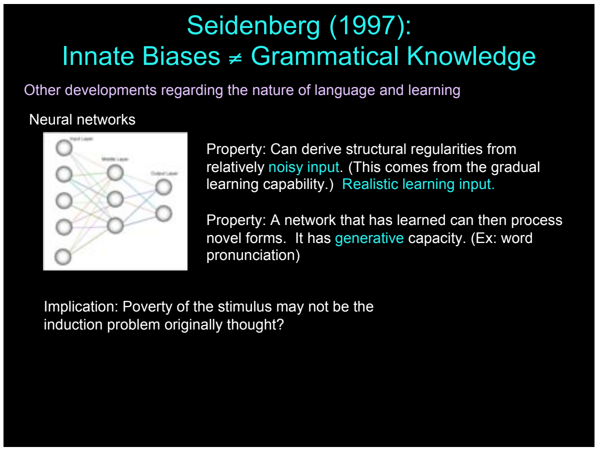
Learning: gradual changes to the weights between units in the network that determine patterns of activation

Parameters: learning rule that adjusts weights, network structure

Not a grammar

Grammar = higher level generalization about network behavior, abstracts away from actual implementation

Grammar = computational level, network = algorithmic + implementational level


Input Layer
Middle Layer
Output Layer

Seidenberg (1997): Innate Biases ≠ Grammatical Knowledge

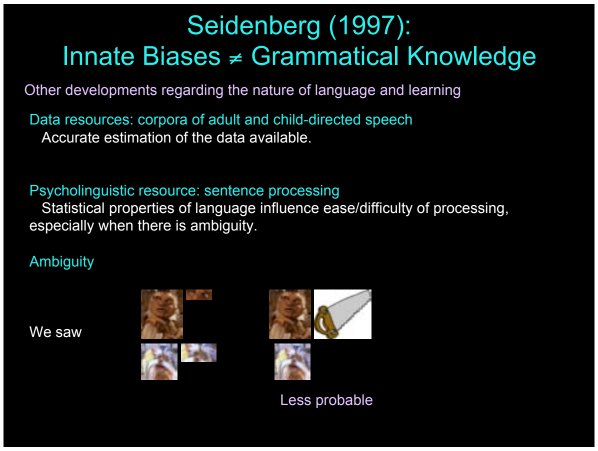
Other developments regarding the nature of language and learning

Neural networks

Property: Can derive structural regularities from relatively noisy input. (This comes from the gradual learning capability.) Realistic learning input.

Property: A network that has learned can then process novel forms. It has generative capacity. (Ex: word pronunciation)

Implication: Poverty of the stimulus may not be the induction problem originally thought?



Seidenberg (1997): Innate Biases ≠ Grammatical Knowledge

Other developments regarding the nature of language and learning

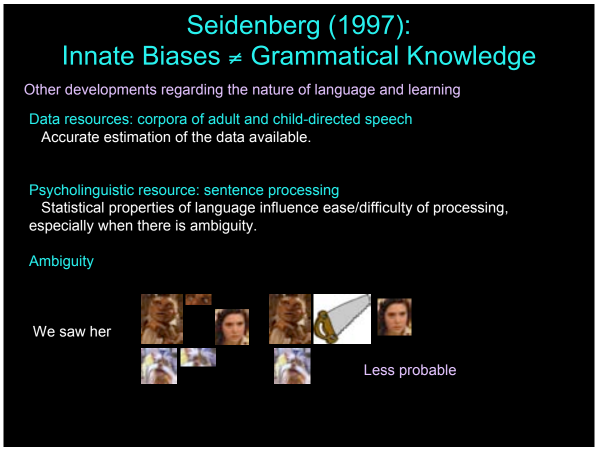
Data resources: corpora of adult and child-directed speech
Accurate estimation of the data available.

Psycholinguistic resource: sentence processing
Statistical properties of language influence ease/difficulty of processing, especially when there is ambiguity.

Ambiguity

We saw

Less probable



Seidenberg (1997): Innate Biases ≠ Grammatical Knowledge

Other developments regarding the nature of language and learning

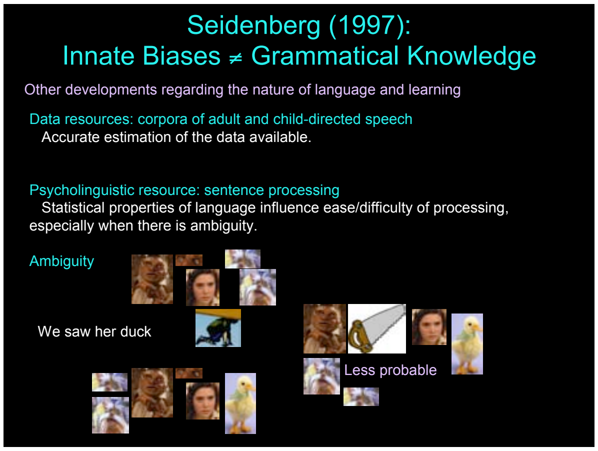
Data resources: corpora of adult and child-directed speech
Accurate estimation of the data available.

Psycholinguistic resource: sentence processing
Statistical properties of language influence ease/difficulty of processing, especially when there is ambiguity.

Ambiguity

We saw her

Less probable



Seidenberg (1997): Innate Biases ≠ Grammatical Knowledge

Other developments regarding the nature of language and learning

Data resources: corpora of adult and child-directed speech
Accurate estimation of the data available.

Psycholinguistic resource: sentence processing
Statistical properties of language influence ease/difficulty of processing, especially when there is ambiguity.

Ambiguity

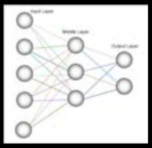
We saw her duck

Less probable

Seidenberg (1997):
Innate Biases ≠ Grammatical Knowledge

Other developments regarding the nature of language and learning

Seidenberg's point: Statistical properties determine language use and neural nets provide a way to explicitly encode, acquire, and exploit this information.



Seidenberg (1997):
Innate Biases ≠ Grammatical Knowledge

Other developments regarding the nature of language and learning

Children can encode statistical properties of language (Jusczyk 1997 = properties of sounds, Saffran et al. 1996 = transitional probabilities of syllables)

Seidenberg's point: Acquisition is about learning to use the language, which means paying attention to its statistical properties and learning from them.

Another point: Connectionist networks formalize the implementation of bootstrapping - extracting regularity from the data (used for word segmentation, word meaning, grammatical category, syntactic constructions)

Seidenberg (1997):
Innate Biases ≠ Grammatical Knowledge

Big point of Seidenberg:

"…[connectionism] attempts to explain language in terms of how is it acquired and used rather than an idealized competence grammar. The idea is not merely that competence grammar needs to incorporate statistical and probabilistic information; rather it is that the nature of language is determined by how it is acquired and used and therefore needs to be explained in terms of these functions and the brain mechanisms that support them. Such performance theories are not merely the competence theory plus some additional assumptions about acquisition and processing; the approaches begin with different goals and end up with different explanations for why languages have the properties they have."

Seidenberg (1997):
Innate Biases ≠ Grammatical Knowledge

Connectionism in Action: An example where it could help
Correlations between verb meaning and verb usage

Hoggle loaded jewels into his bag.
Hoggle loaded his bag with jewels.

Hoggle poured jewels into his bag.
*Hoggle poured his bag with jewels.

*Hoggle filled the jewels into his bag.
Hoggle filled his bag with jewels.

Input is irregular - children do not get explicit examples of all of these, yet somehow come to know this paradigm.

Seidenberg (1997): Innate Biases ≠ Grammatical Knowledge

Clue

clusters of verbs with similar properties (if children realize this, learning is easier)

load, pile, cram, spray, scatter
pour, drip, slop, slosh
fill, blanket, cover, coat

Problem: How would the child know to cluster these verbs together if they never hear all the verbs in all the possible syntactic frames? Semantically, they're very similar.

However…

This is a constraint satisfaction problem, which neural nets are really good at solving.

Seidenberg (1997): Innate Biases ≠ Grammatical Knowledge

Information available on groupings

load, pile, cram, spray, scatter
pour, drip, slop, slosh
fill, blanket, cover, coat

1) How much the semantics of each verb overlaps with any other verb
2) Correlations between syntactic frames verbs appear in and the exact semantics of the verb
3) Item-specific idiosyncracies (due to language change)

Connectionist net can learn the right subgroups (Allen 1997) from this information

…and then much easier to notice that there are syntactic usage generalizations for the groups. Therefore, this can be learned. Which is good, since it's a language-specific property.

Seidenberg (1997): Innate Biases ≠ Grammatical Knowledge

But what about learning more abstract things (like syntax) and language-independent things that are hard (or impossible) to observe?

…future work for connectionist models.

And innate knowledge?

"Innate capacities may take the form of biases or sensitivities toward particular types of information inherent in environmental events such as language, rather than a priori knowledge of grammar itself."

"Brain organization therefore constrains how language is learned, but the principles that govern the acquisition, representation, and use of language are not specific to this type of knowledge"

Marcus (2003): Symbols

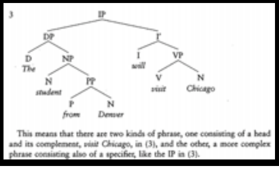
The Mind & Symbols

Big question: Is the mind something that manipulates symbols? Or is the basic unit of cognition something that has nothing to do with the "sentences and propositions" of symbol-manipulation (Churchland, 1995)?

Symbol-manipulating models: typically described in terms of production rules & hierarchical trees

Production rule: If precondition 1 is true, do action 1
"If surface is hot, remove hand"

Hierarchical tree:

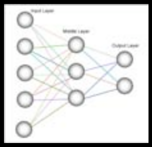


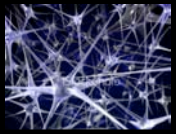
This means that there are two kinds of phrase, one consisting of a head and its complement, *visit Chicago*, in (3), and the other, a more complex phrase consisting also of a specifier, like the IP in (3).

Marcus (2003): Symbols

The Mind & Symbols

Connectionist models: tend to be “neurally-inspired”, described in terms of neuron-like units and synapse-like connections





Important point: Just because something is connectionist doesn’t mean it can’t also manipulate symbols (connectionist = implementational level, symbols = computational level)

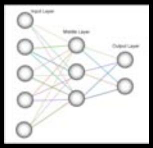
Marcus (2003): Symbols

The Symbolicists vs. the Non-Symbolic Connectionists

Symbolicist assumption: circuits in the brain correspond in some way to the basic devices necessary for symbol manipulation (e.g. some circuit supports representation of a rule)

Non-Symbolic Connectionist assumption: there will not be any brain circuits like this (rules are epiphenomena of regularity in patterns of activation)

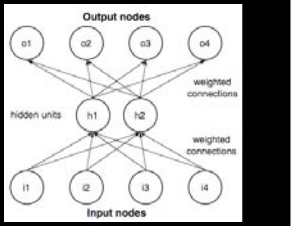
Non-symbolic connectionists tend to focus on multilayer perceptrons as a model of cognition, and this is the model in general that’s brought up whenever symbols (or no symbols) are. (This is because it’s an explicitly-formed model.)



Marcus (2003): Symbols

Multilayer Perceptrons

Nodes: have activation values (0.5, 1.0)
- input/output: have meaning associated with them (+ed, *walk*, …)
- meaning affects what things are considered alike (c onset (cat ~ cab) vs. +animal (cat ~ dog))



Activation values: numbers assigned to nodes, based on input

Ex: +furriness is set to 1.0 is input is furry, 0.0 otherwise

Furriness node = 1.0

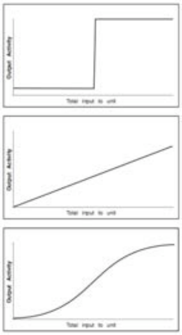
Marcus (2003): Symbols

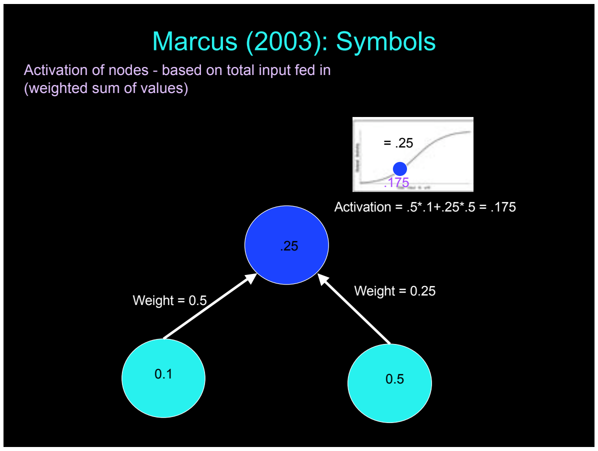
Activation of nodes - based on total input fed in (weighted sum of values)

Step or binary threshold function - either on or off, based on threshold

Linear function - activation scales linearly with input

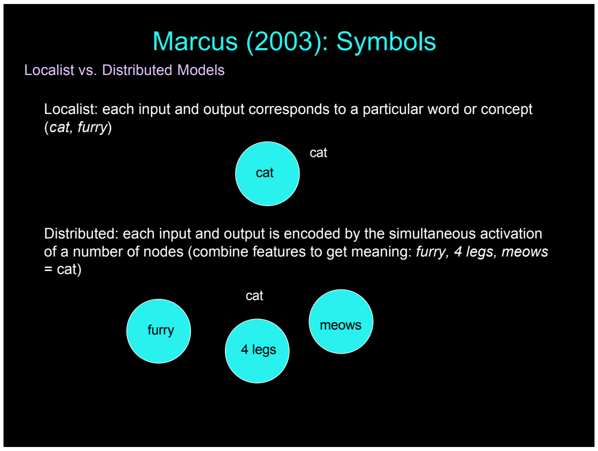
Sigmoid function - activation scales curvily with input (models with hidden units tend to use this kind)





Marcus (2003): Symbols

Activation of nodes - based on total input fed in (weighted sum of values)

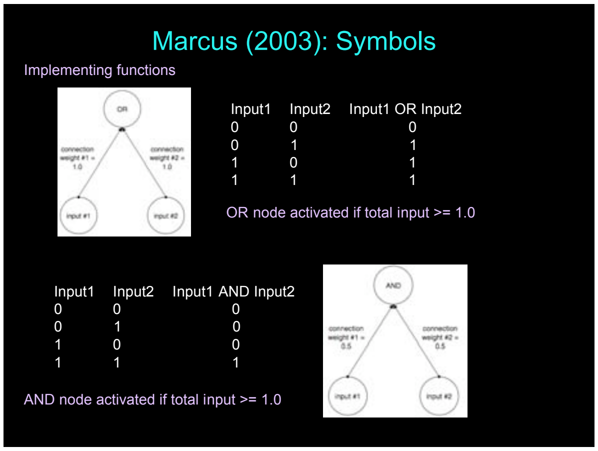


Marcus (2003): Symbols

Localist vs. Distributed Models

Localist: each input and output corresponds to a particular word or concept (*cat, furry*)

Distributed: each input and output is encoded by the simultaneous activation of a number of nodes (combine features to get meaning: *furry, 4 legs, meows* = cat)



Marcus (2003): Symbols

Implementing functions

OR
connection weight #1 = 1.0
connection weight #2 = 1.0
input #1
input #2

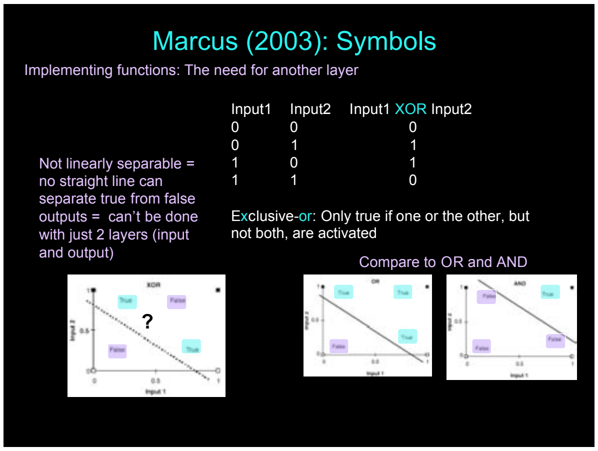
Input1	Input2	Input1 OR Input2
0	0	0
0	1	1
1	0	1
1	1	1

OR node activated if total input >= 1.0

Input1	Input2	Input1 AND Input2
0	0	0
0	1	0
1	0	0
1	1	1

AND node activated if total input >= 1.0

AND
connection weight #1 = 0.5
connection weight #2 = 0.5
input #1
input #2



Marcus (2003): Symbols

Implementing functions: The need for another layer

Input1	Input2	Input1 XOR Input2
0	0	0
0	1	1
1	0	1
1	1	0

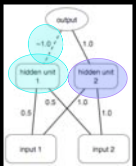
Not linearly separable = no straight line can separate true from false outputs = can't be done with just 2 layers (input and output)

Exclusive-or: Only true if one or the other, but not both, are activated

Compare to OR and AND

Marcus (2003): Symbols

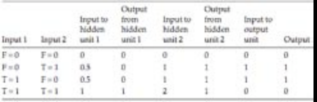
Implementing functions: The need for another layer



Input1	Input2	Input1 XOR Input2
0	0	0
0	1	1
1	0	1
1	1	0

Exclusive-or: Only true if one or the other, but not both, are activated

OR and not AND



Input 1	Input 2	Input to hidden unit 1	Output from hidden unit 1	Input to hidden unit 2	Output from hidden unit 2	Input to output unit	Output
F = 0	F = 0	0	0	0	0	0	0
F = 0	T = 1	0.5	0	1	1	1	1
T = 1	F = 0	0.5	0	1	1	1	1
T = 1	T = 1	1	1	2	1	0	0

Marcus (2003): Symbols

About hidden layers

Sometimes thought of as recoding the input (ex: XOR hidden layer has OR and AND in it) - similar to internal representations of input

About learning with multiple layers: initially, connection weights are random and need to be adjusted

One way: Hebbian learning
"Cells that fire together wire together" - strengthen connection weight between input node and output node every time they are active simultaneously

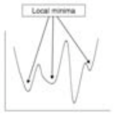
Another way: Delta ("difference") rule learning
Change weight of connection between input and output node, based on activation of input node multiplied by difference between what output node should have done and what output node actually did (involves parameter = learning rate = how much adjustment)
For hidden layers, use back propagation variant that estimates what hidden layer input and output activations should be. (Not believed to be psychologically plausible by some researchers, though.)

Marcus (2003): Symbols

The nice thing about back propagation

If learning rate is small, back propagation is a gradient descent algorithm - gradually getting closer and closer to a right answer (set of weights), which is at a metaphorical "valley" on the answer "landscape"

One pitfall: local minima



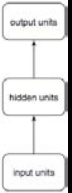
Bonus: Small learning rate = gradual learning (which is what children seem to do)

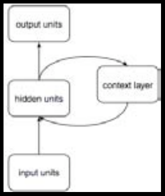
But these algorithms require supervision - need to know what the right output activation should have been. Where does this come from? One answer: The data to the learner. (Need to verify this for each learning problem, though.)

Example: past tense model
Scenario: past tense of *run*
Model predicts: *run+ed* …therefore, adjust weights
Data = *ran*

Marcus (2003): Symbols

Feedforward vs. Simple Recurrent Networks





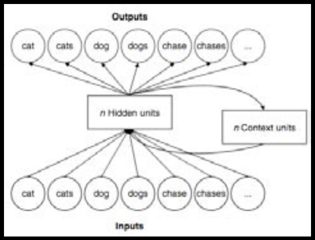
Simple recurrent networks (SRNs) used for learning linguistic inflection, grammatical knowledge, object permanence, categorization, logical deduction…

SRNs can learn something about the sequence of elements presented over time.

Marcus (2003): Symbols

Elman (1990, 1993) sentence prediction model

26 input, 26 output nodes



Context layer records copy of activation pattern of hidden layer = gives 1 time step worth of memory

Learning via back propagation

Task: predict next word in sentence
Training: sentences from toy grammar with 23 words and a variety of grammatical dependencies (subject-verb agreement)

Marcus (2003): Symbols

Elman (1990, 1993) sentence prediction model

Learned complicated strings like
cats chase dogs
boys who chase dogs see girls

…without grammatical rules explicitly built in (singular subject has verb with singular ending, match main clause subject with main clause verb).

Taken as strong evidence against the need for grammatical rules in language.

Anderson & Hinton (1981)

"…the symbol-processing metaphor may be an inappropriate way of thinking about computational processes that underlie abilities like learning, perception, and motor skills…alternative models that appear to be more appropriate for machines like the brain."

Marcus (2003): Symbols

Rumelhart & McClelland (1986): past tense acquisition model

Two-layer perceptron (no hidden layer)
"provides a distinct alternative…to [rules] in any explicit sense"

But what are the real arguments against symbol manipulation?

One argument: multilayer perceptrons are more compatible with what we know about the way the brain is laid out

Counter-argument: perceptron "neurons" and "synapses" are only loosely based on real brain neurons and synapses - they have properties real ones don't; also, how is back propagation instantiated?

Marcus (2003): Symbols

More arguments against symbol-manipulation models

Argument: Connectionist models have been show to degrade gracefully (when part of the network is knocked out, it can still function)

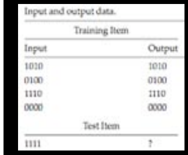
Counter-argument: Symbol-manipulating models can do this, too (error-correction algorithms, partial feature-matching algorithms)

Another argument: they have been show mathematically to be able to represent a large class of functions (universal function approximator, generalization ability)

Counter-argument: mathematical proofs don't have real world considerations (like non-infinite data or realistic distributions of data); also, class of representable functions may not be all the ones needed for language (partial recursive functions would be necessary (Hadley (2000))

Marcus (2003): Generalization

A sample function



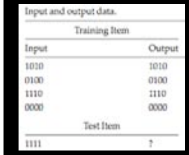
Input and output data.

Input	Output
Training Item	
1010	1010
0100	0100
1110	1110
0000	0000
Test Item	
1111	?

What do you think the answer to the test item is?

Marcus (2003): Generalization

A sample function



Input and output data.

Input	Output
Training Item	
1010	1010
0100	0100
1110	1110
0000	0000
Test Item	
1111	?

Humans: 1111

Multilayer perceptrons: 1110

Why? 4th position is always 0 in training set. Local mathematically sound generalization from training data: 4th position is always 0.

But this doesn't seem to be what humans do…

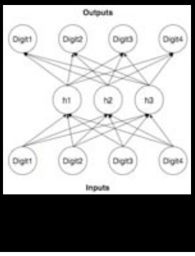
An algebraic system that uses an identity function (f(x) = x) would be able to capture this behavior.

Marcus (2003): Generalization

Learning freely generalizable functions

Multilayer perceptrons can learn universally quantified one-to-one mapping functions only if they see an example of each input-output combination. This is because each output is independent of the others.

Multilayer perceptron model for learning identity function with 4 digits



Model has no way to tell all digits should be treated uniformly

Point: For situations where humans freely generalize, need a model of cognition that is also capable of free generalization

Language = many places where humans seem to freely generalize from restricted data

Marcus (2003): Types of Questions

Good: Does the mind have rules in anything more than the descriptive sense?

Not so good: Can we build a connectionist model that does [insert something humans seem to do, ex: the past tense formation in English]?

Just because the model is implemented in a connectionist style does not mean it doesn't have rules built in.

Better: What design features must a connectionist model that does [insert something humans seem to do] need to include?

Past tense case: The connectionist models that come closest to implementing rules & memory do the best approximation to human performance.

Implication: Rules (operations over variables) are an important part of human cognition.