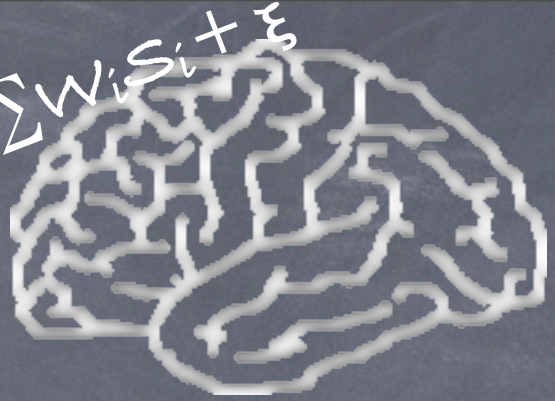
ACT-R: modeling the brain

A Cognitive Architecture

ACT-R

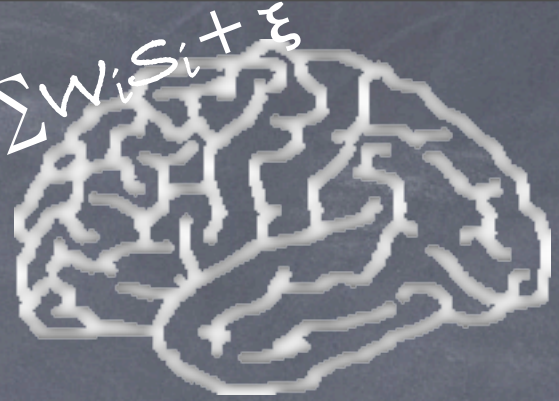
$$A_i = B_i + \sum W_i S_i + \xi$$



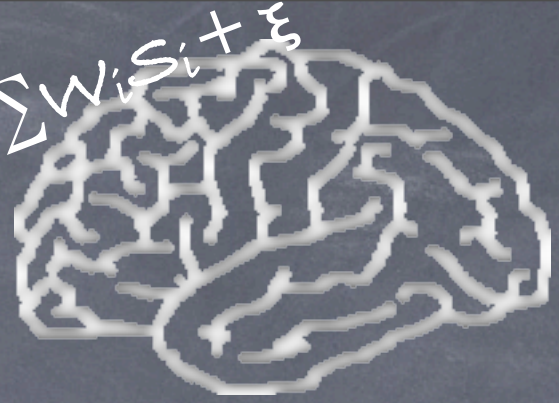
"If the brain were so simple we could understand it,
we would be so simple we couldn't."

Lyall Watson

Overview



- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary

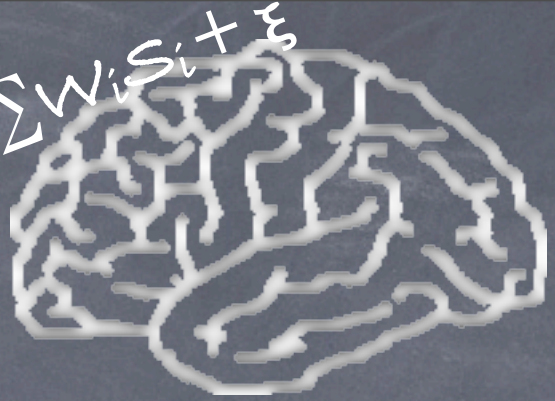


ACT-R:

- stands for Adaptive Control of Thought - Rational
- developed by Anderson at CMU in 1993
- it's a theory about how the mind works, and also a runnable program, written in lisp
- can be used for general problem solving
- but most useful for modeling aspects of human behaviour

ACT-R

$$A_i = B_i + \sum W_i S_i + \xi$$

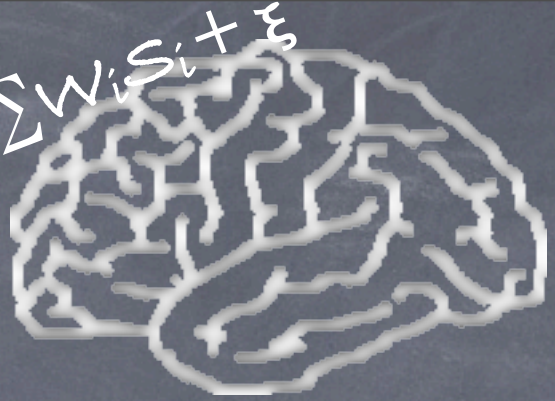


- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary

ACT-R

the architecture

$$A_i = B_i + \sum W_i S_i + \xi$$



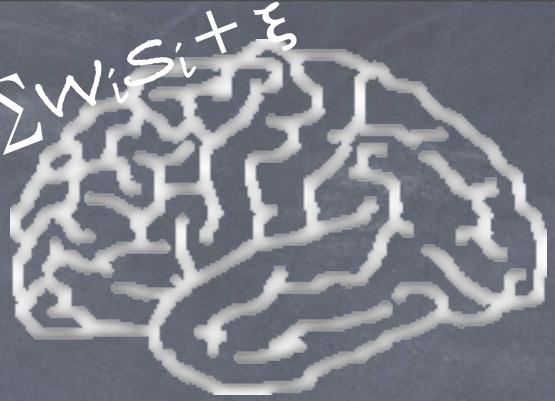
- act-r is a hybrid cognitive architecture
- symbolic level
  - production system
  - facts, rules, goals
- subsymbolic level
  - learning without making it explicit with production rules
  - determines which facts are retrieved, which rules are used, how long the retrieval takes

"best of both worlds"

ACT-R

the architecture

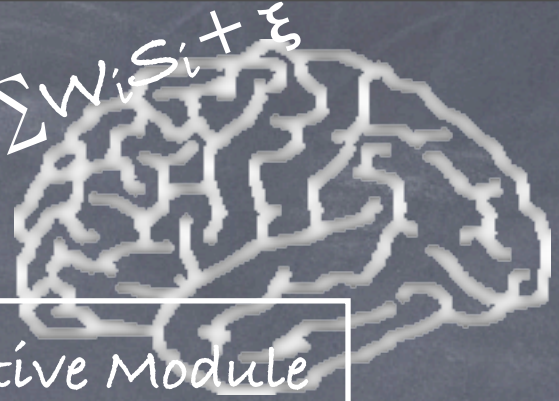
$$A_i = B_i + \sum W_i S_i + \xi$$

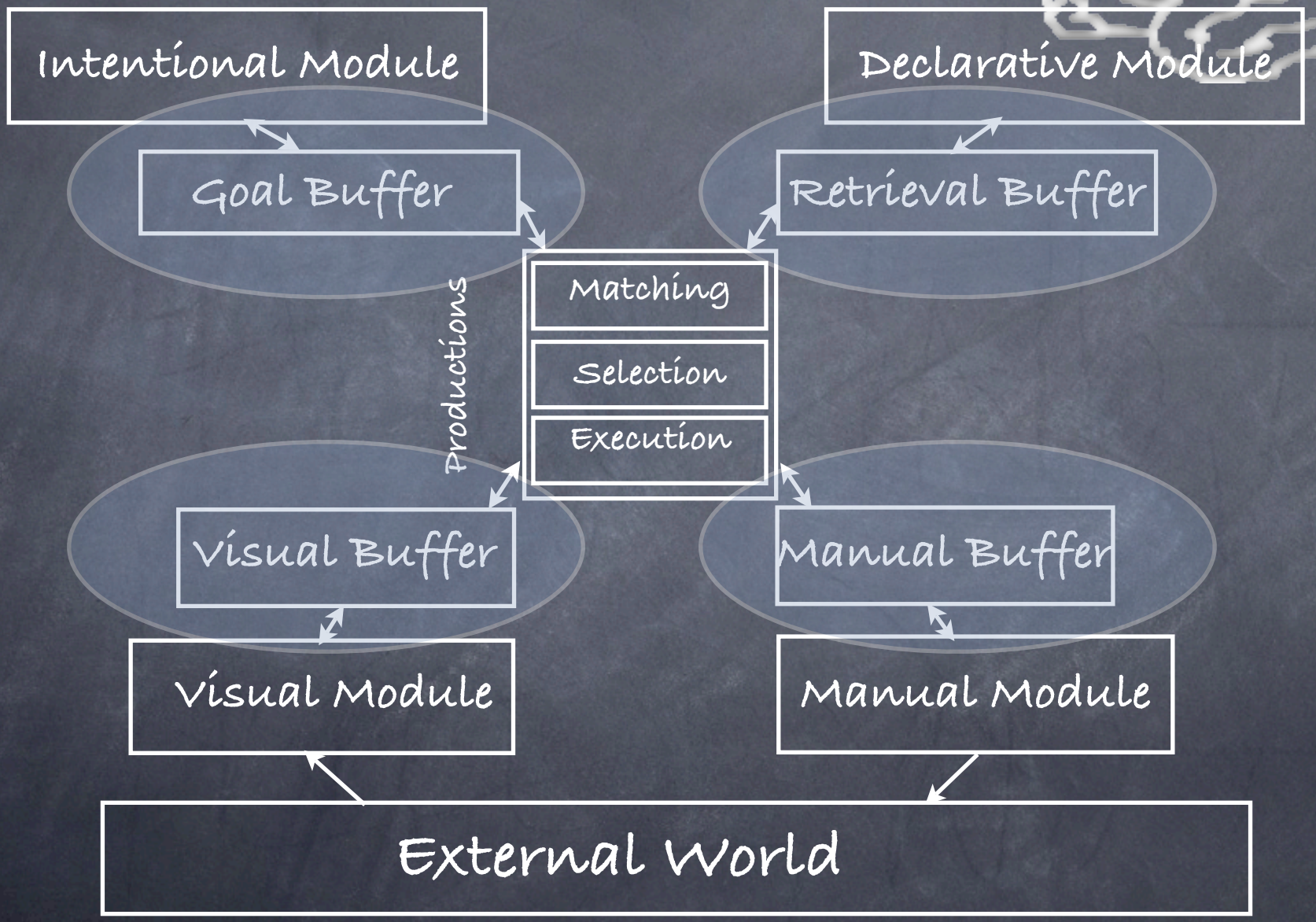


- various parts ("modules") of the brain is involved in cognition
- input from our eyes, movement of our arms, remembering facts stored in our long term memory etc.
- not all of this information available at all time
- buffers corresponding to the modules reflect this, for instance we don't know all our knowledge at once, or attend to all objects in our visual field at once

ACT-R the architecture

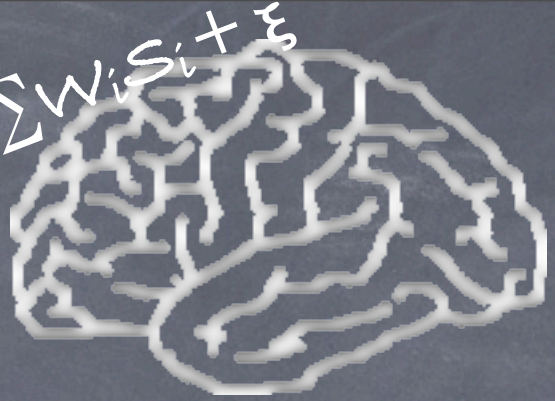
$$A_i = B_i + \sum W_i S_i + \xi$$



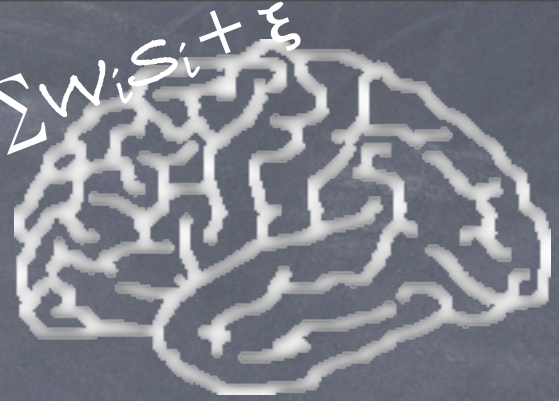


ACT-R

$$A_i = B_i + \sum W_i S_i + \xi$$



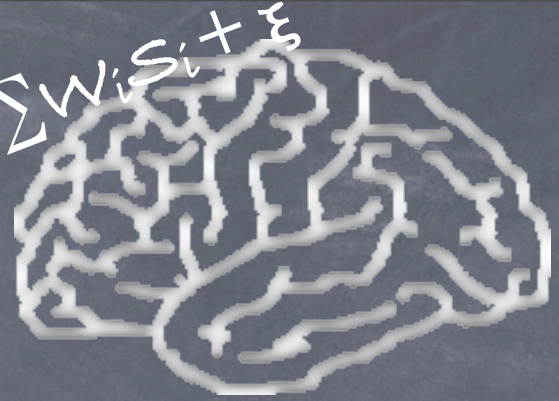
- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary



the building blocks

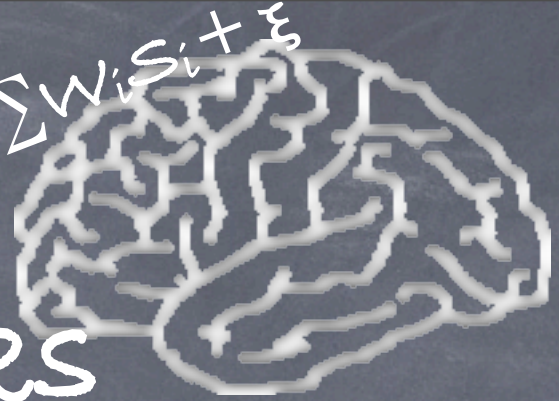
chunks

- declarative memory contains facts, the smallest units of knowledge, in ACT-R this is called a chunk
- a chunk type is defined with the command (chunk-type chunk-name slot1 slot2 ...)
- an instance of a chunk is added to declarative memory with the add-dm command



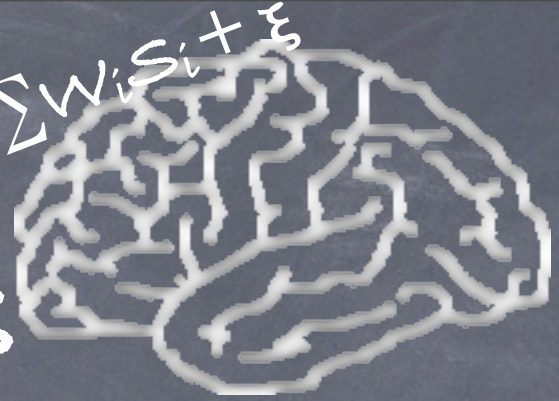
the building blocks chunks

- (chunk-type bird feet flies)
- (ostrich isa bird feet 2 flies no)
- the first one is the template or class, and the second is an instance, a single fact, or in act-r terminology; a chunk



the building blocks production rules

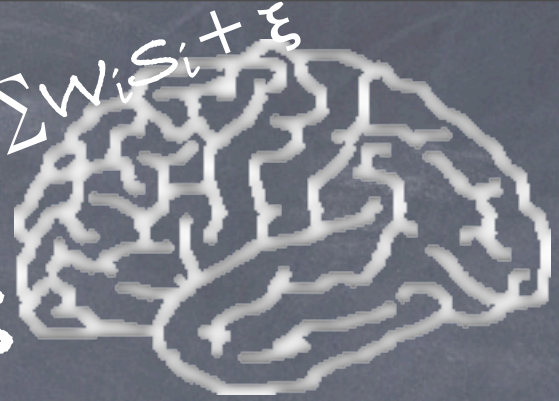
- the central production system contains rules, which makes up the procedural knowledge
- these rules are called productions
- matched against the chunks available in the buffers
- but only one production get to fire in each cycle



the building blocks

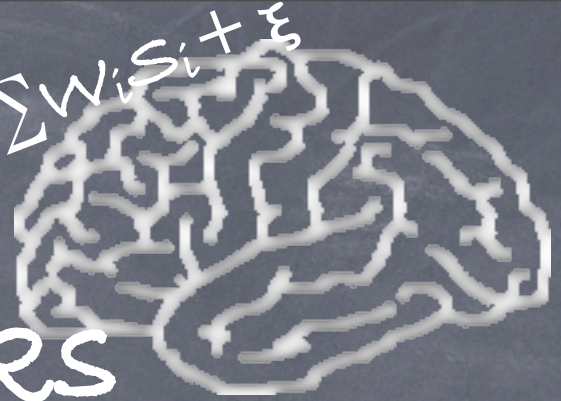
production - syntax

- a production consists of a LHS and a RHS
- the LHS must access the goal buffer
- a production is written as
  - (p LHS ==> RHS)
- LHS and RHS access buffers and matches chunks with conditions



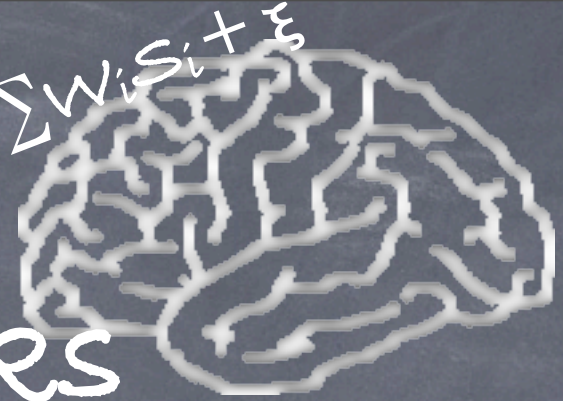
the building blocks
production execution

- the production system has a cycle that executes the matching, selection and execution every simulated time unit, typically 50 ms
- a production is thought of as a simple step of cognition
- it's how act-r models the smallest units of cognitive processing



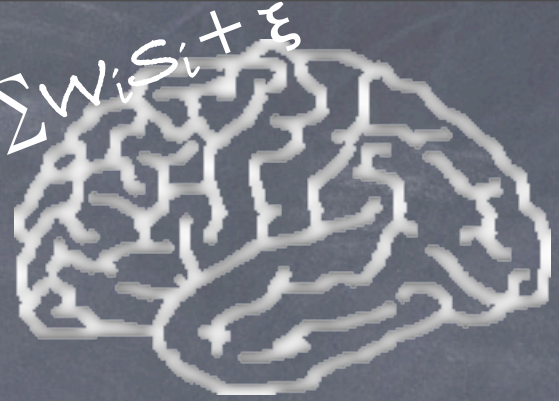
the building blocks
buffers

- goal, retrieval, visual, manual most common
  - buffers correspond to modules, and more modules can be added
- is accessed through =buffer-name> (modifies), +buffer-name> (request and indirect purge) or -buffer-name> (purge)
- the only way to access chunks; retrieval from buffers to the central production system



the building blocks
buffers

- communication with the "outside world" (simulated or real through experiments) happen in the perceptual-motor modules
- current object of attention from these modules are also made available in buffers



processing

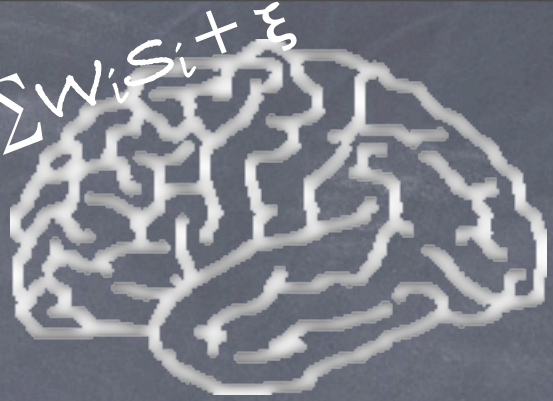
- the productions all exists in the central matching-selection-execution system
- the different buffers can contain information
- sequentiality from
  - only one chunk per buffer at a certain time
  - only one production gets to fire per production cycle

variables

- variables are written like =var-name
- =goal and =retrieval are also variables, bound to whatever is in the goal and retrieval buffers
- the first time a variable is used in a production it gets assigned (bound to) the values of the specified slot from the chunk in the buffer
- if the slot does not have a value, then the pattern does not match

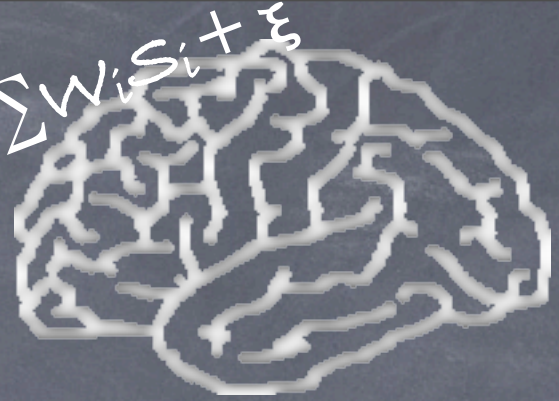
ACT-R

$$A_i = B_i + \sum W_i S_i + \xi$$



- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary

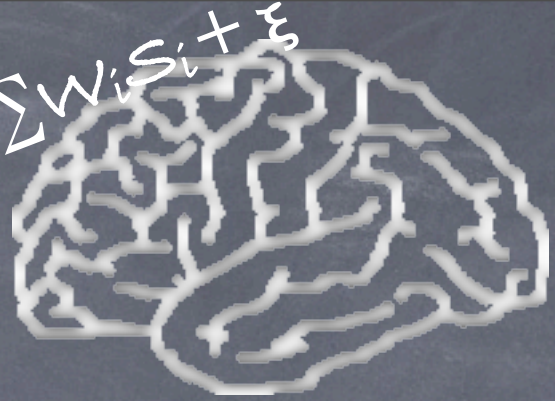
ACT-R example: count




```
(chunk-type count-order first second)
(chunk-type count-from start end count)
(add-dm
 (b ISA count-order first 1 second 2)
 ...
 (f ISA count-order first 5 second 6)
 (first-goal ISA count-from start 2 end 4)
 )
(p start
   =goal>
      ISA         count-from
      start       =num1
      count       nil
   ==>
   =goal>
      count       =num1
   +retrieval>
      ISA         count-order
      first       =num1
)
```

ACT-R example: count

$A_i = B_i + \sum W_i S_i + \xi$

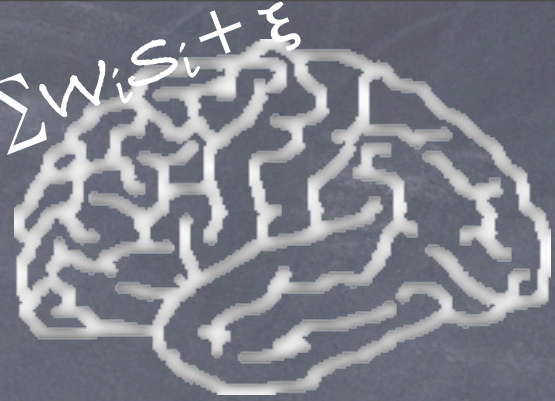


```
(P increment
   =goal>
      ISA         count-from
      count       =num1
    - end         =num1
   =retrieval>
      ISA         count-order
      first       =num1
      second      =num2
==>
   =goal>
      count       =num2
   +retrieval>
      ISA         count-order
      first       =num2
   !output!       (=num1)
)
```

```
(P stop
   =goal>
      ISA         count-from
      count       =num
      end         =num
==>
   -goal>
   !output!       (=num)

)
```

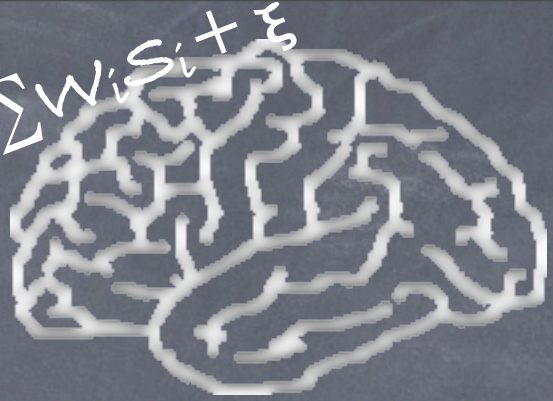
$A_i = B_i + \sum W_i S_i + \xi$



DEMO

ACT-R

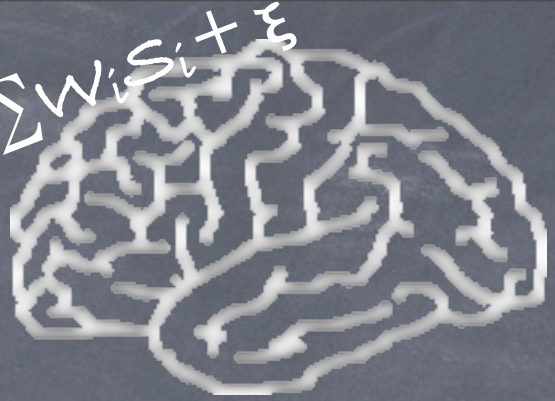
$$A_i = B_i + \sum W_i S_i + \xi$$



- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary

subsymbolic features

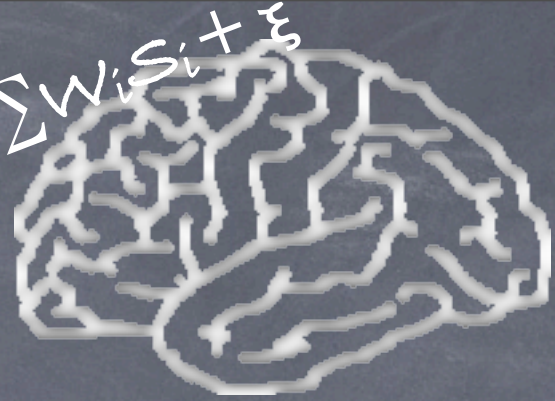
$$A_i = B_i + \sum W_i S_i + \xi$$



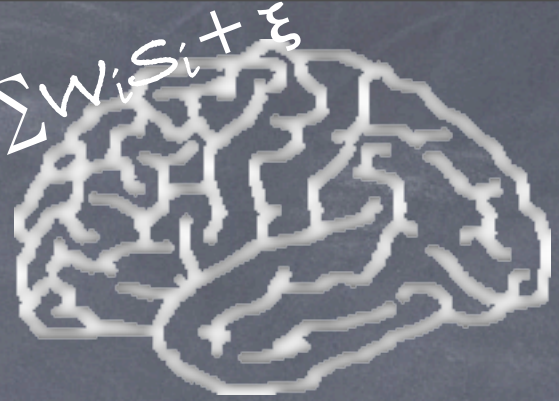
- learning of probability of goal achievement
- learning of relevance of knowledge at hand
- conflict resolution: which chunks to retrieve, which production to fire
- possible to implement the subsymbolic functionality using production rules, but gets cumbersome and expensive

subsymbolic features

$$A_i = B_i + \sum W_i S_i + \xi$$



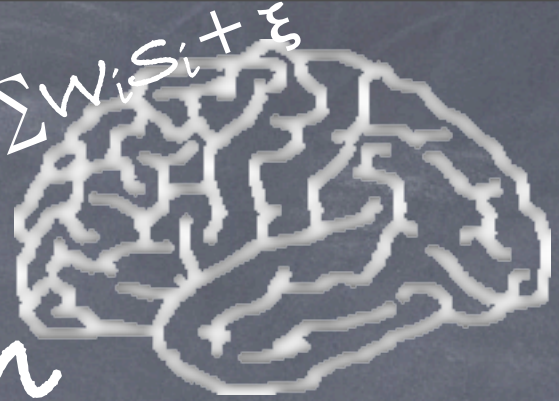
- parameters determines if they are used, and how, but ACT-R is supposed to be a parameter free model, not fitting the model to the data
  - act-r often criticized for involving too much parameter tweaking
- the goal is a adaptive architecture with fixed parameters that are able to predict data from psychological experiments



Chunk selection

activation (A)

- Activation = recency of use + usefulness + noise
- a chunk is selected on basis of its recency of use and usefulness in similar situations
- the noise component adds some randomness to the selection process

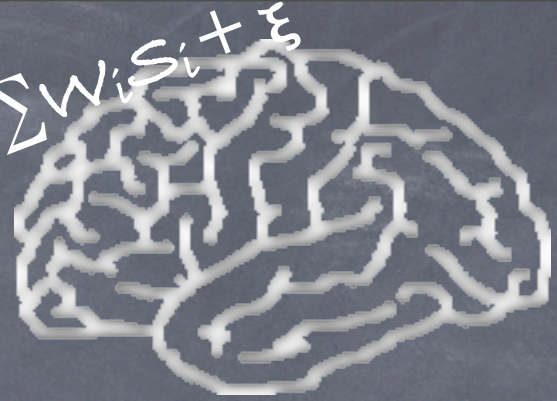


production selection

utility

- $U = P^*G - C$ + noise
- a production is selected on basis of its likelihood (P) to lead to the goal minus the cost (C) of execution
  - $P = \frac{\text{Successes}}{\text{Successes} + \text{Failures}}$
- The goal value (G) is a global parameter, cost and probability of success is learned

$$A_i = B_i + \sum W_i S_i + \xi$$



how it fits together

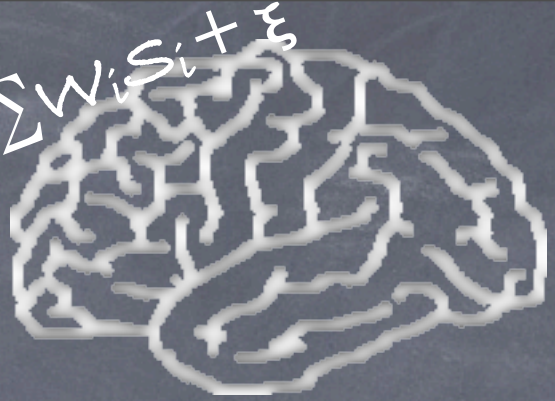
	Declarative	Procedural
Symbolic	retrieval of chunks	application of production rules
Subsymbolic	activation	utility

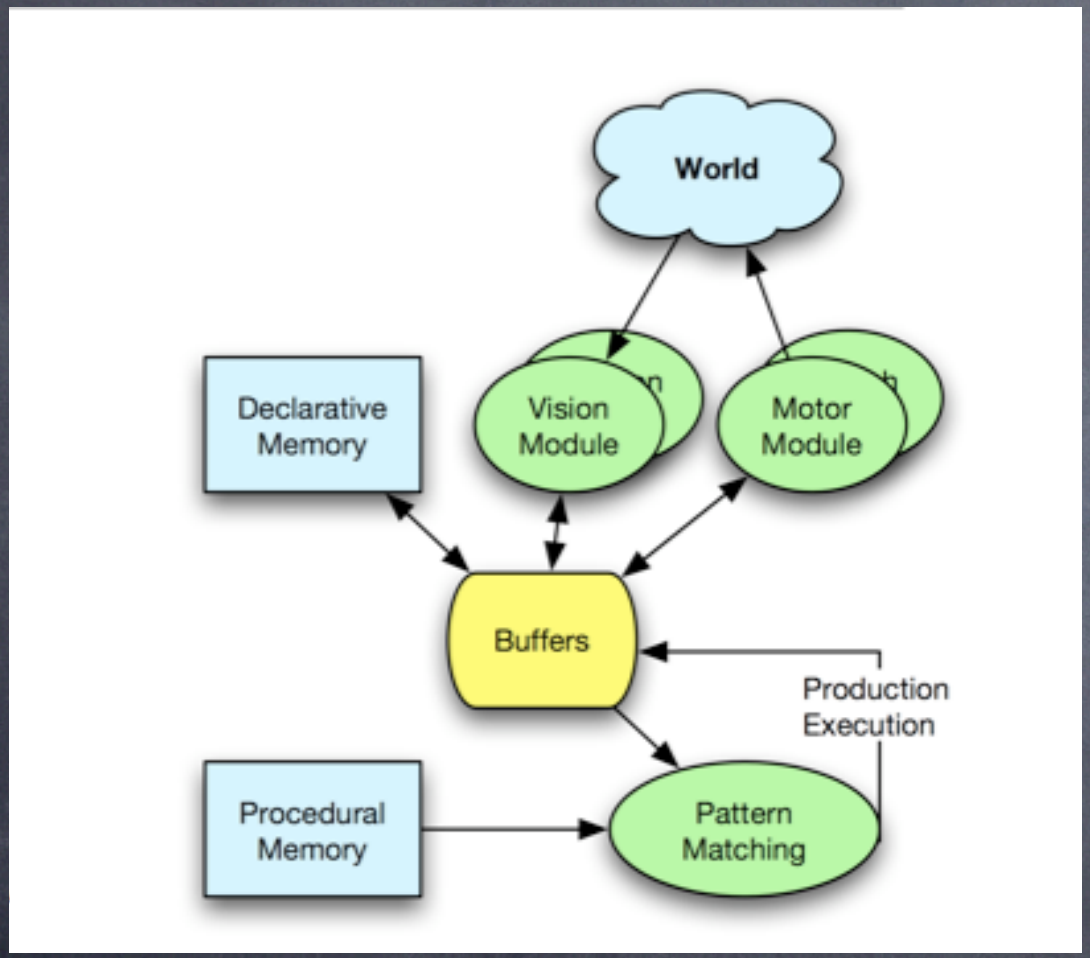
Overview

- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary

ACT-R

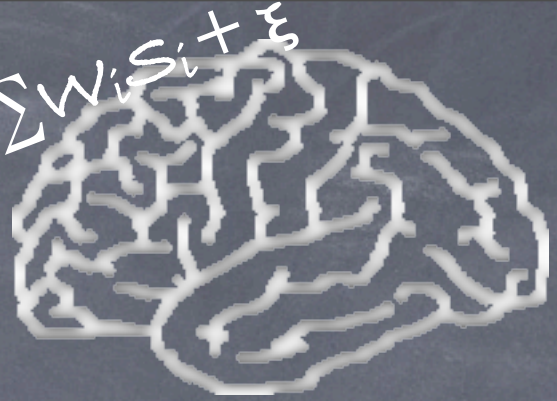
$$A_i = B_i + \sum W_i S_i + \xi$$





well this looks like a standard production system, what makes act-r so special?

$A_i = B_i + \sum W_i S_i + \xi$

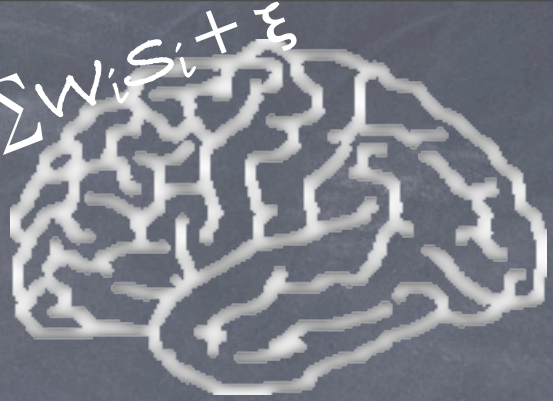


unified theory of cognition

recap:

- Proposed by Allen Newell in 1987
- Newell argued for the need of a set of general assumptions for cognitive models that account for all of cognition
- presents a unified way of viewing and modeling all cognitive processes
- act-r proposed as a candidate for UTC

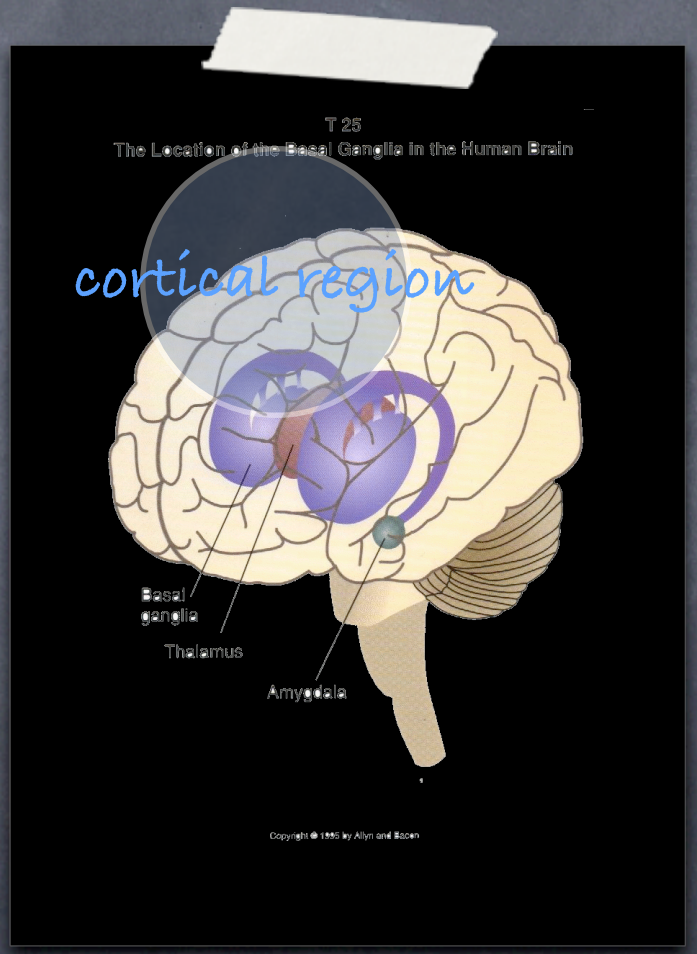
$$A_i = B_i + \sum W_i S_i + \xi$$



The brain:

- the retrieval from the declarative memory and the processing of chunks in the production system are roughly equivalent to the flow from the cortex to the basal ganglia and back

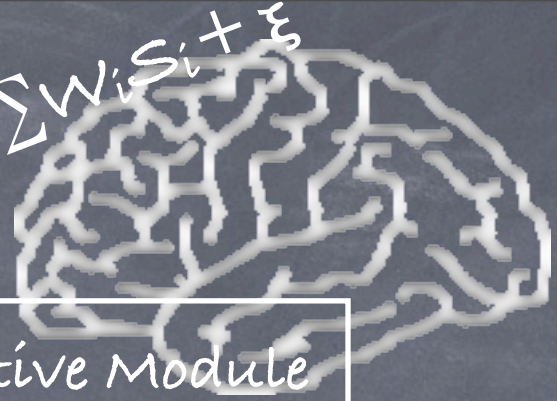
- the basal ganglia are thought to be where we acquire new knowledge



ACT-R

it's thought to be a mapping to how the brain actually works...

$$A_i = B_i + \sum W_i S_i + \xi$$



Intentional Module

Declarative Module

Goal Buffer

Retrieval Buffer

DL cortex

VL cortex

Productions

Matching

Selection

Execution

striatum

basal ganglia

pallidum

thalamus

parietal

Visual Buffer

Manual Buffer

motor

occipital

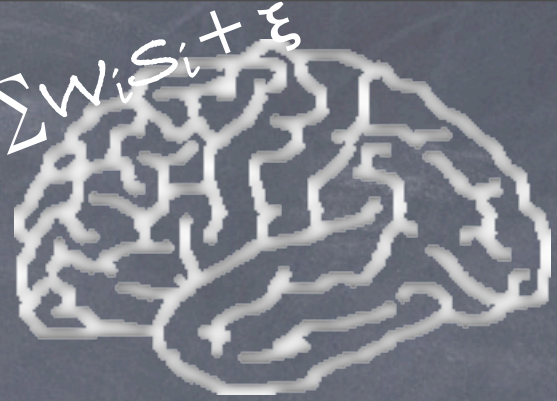
motor/cerebellum

Visual Module

Manual Module

External World

$$A_i = B_i + \sum W_i S_i + \xi$$



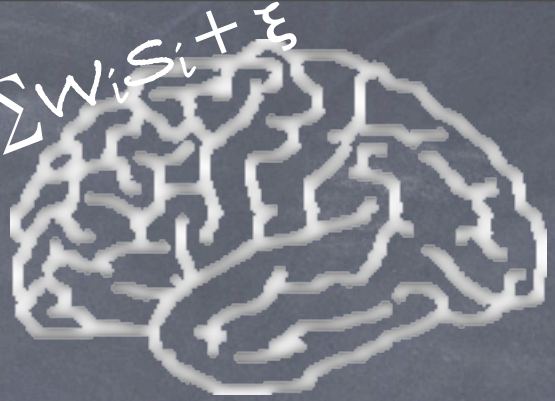
but...

- of course it's a simplification
- direct links between brain regions exist
- brain functions not fully understood
- not a general agreement in the cognitive psychology community that the human cognitive processes can be modeled as a production system

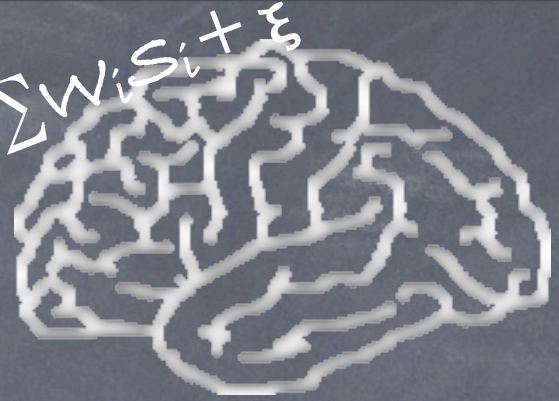
!

ACT-R

$$A_i = B_i + \sum W_i S_i + \xi$$



- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary



learning

new chunks

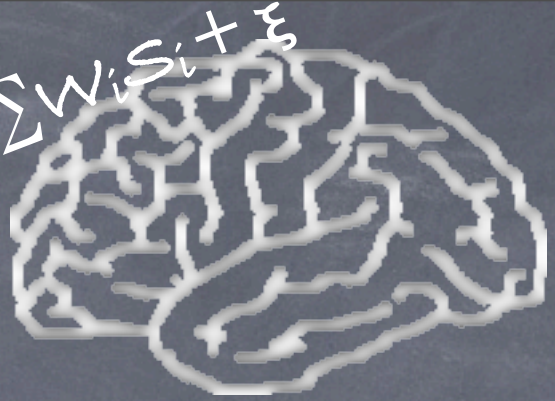
- chunks can be aquired from outside world by using for instance the perceptual-motor modules

production compilation

- if two productions are fired in sequence, we can combine them into a new production
- if the new production is better, it will eventually replace the old ones

ACT-R

$$A_i = B_i + \sum W_i S_i + \xi$$



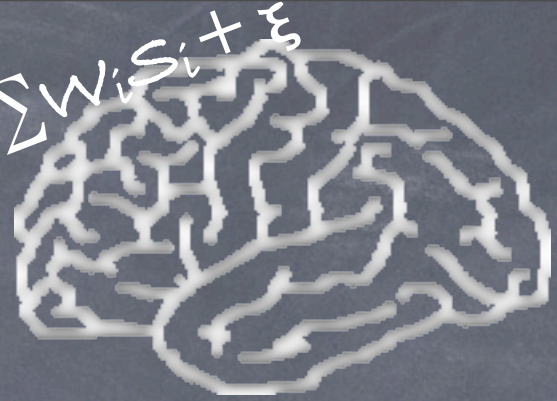
- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary

uses

- Human-Computer Interaction to produce user models that can assess different computer interfaces
- education (cognitive tutoring system) to "guess" the difficulties that students may have and provide focused help
  - CMU's cognitive math tutor a commercial success
- neuropsychology, to interpret brain scan data

act-r and emotion

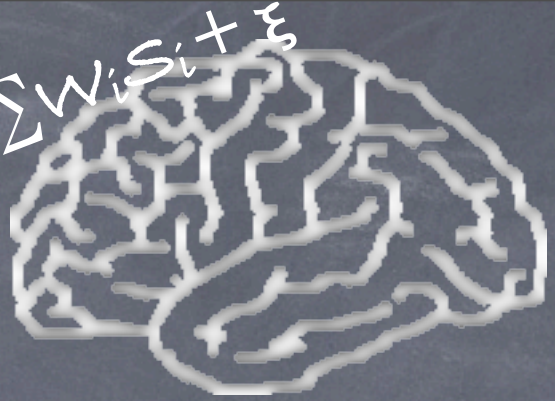
$$A_i = B_i + \sum W_i S_i + \xi$$



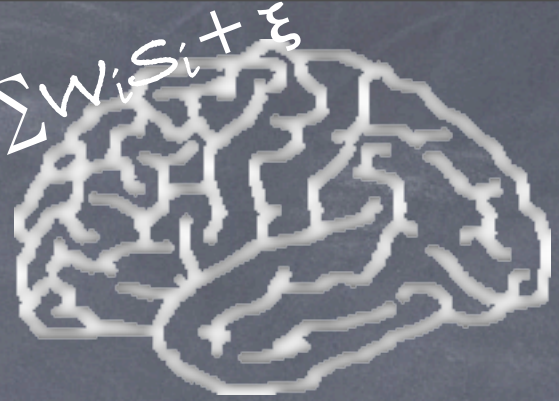
- emotion play a role in problem solving
- optimism makes us pursue a certain strategy, pessimism gets us out of local minima
- act-r can be used to model how emotions affect our problem solving strategies (Belavkin, 2002)
- for instance, we can model optimist as going from a low goal value and much noise, to a high goal value and less noise, optimism makes us focus on current solving strategy

ACT-R

$$A_i = B_i + \sum W_i S_i + \xi$$



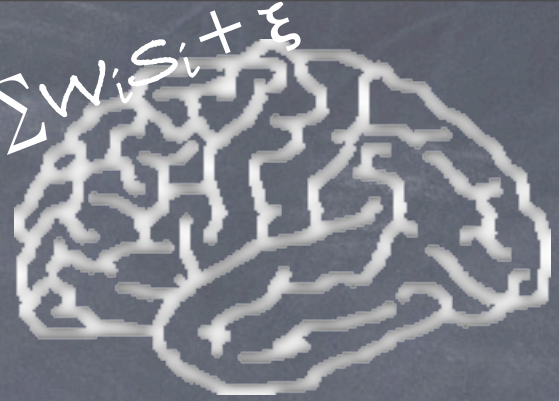
- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary



challenges

- hard to write plausible models
  - one has to be both skilled in writing computer programs and in modeling cognitive processes
  - thus confined to a small set of academics
- knowledge of lisp almost required
- documentation is not very good, and the different versions of act-r are not backwards compatible

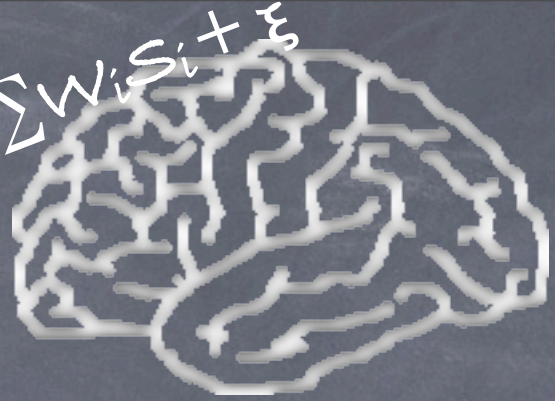
$$A_i = B_i + \sum W_i S_i + \xi$$



summary

- act-r introduces a way to think about the brain and the parts of it involved in learning and planning
- it is proposed as a unified theory of cognition
- act-r assumes that the cognitive processes in the brain can be modeled as a production system
- hard to write good models
- not a complete solution to cognitive modeling, but hopes to bring some insight

$A_i = B_i + \sum W_i S_i + \xi$



Questions?