CISC452/CMPE452/COGS 400

NN – Why?

Farhana Zulkernine

Why Cognitive Modeling?

1. To understand cognitive processes better
2. To computationally implement a cognitive process
   - Create a model of the mind
     - Can we build an actual mind some day?
3. Compare and evaluate the various explanations of the cognitive processes
4. Predict outcomes of cognitive processes

Modeling Tools

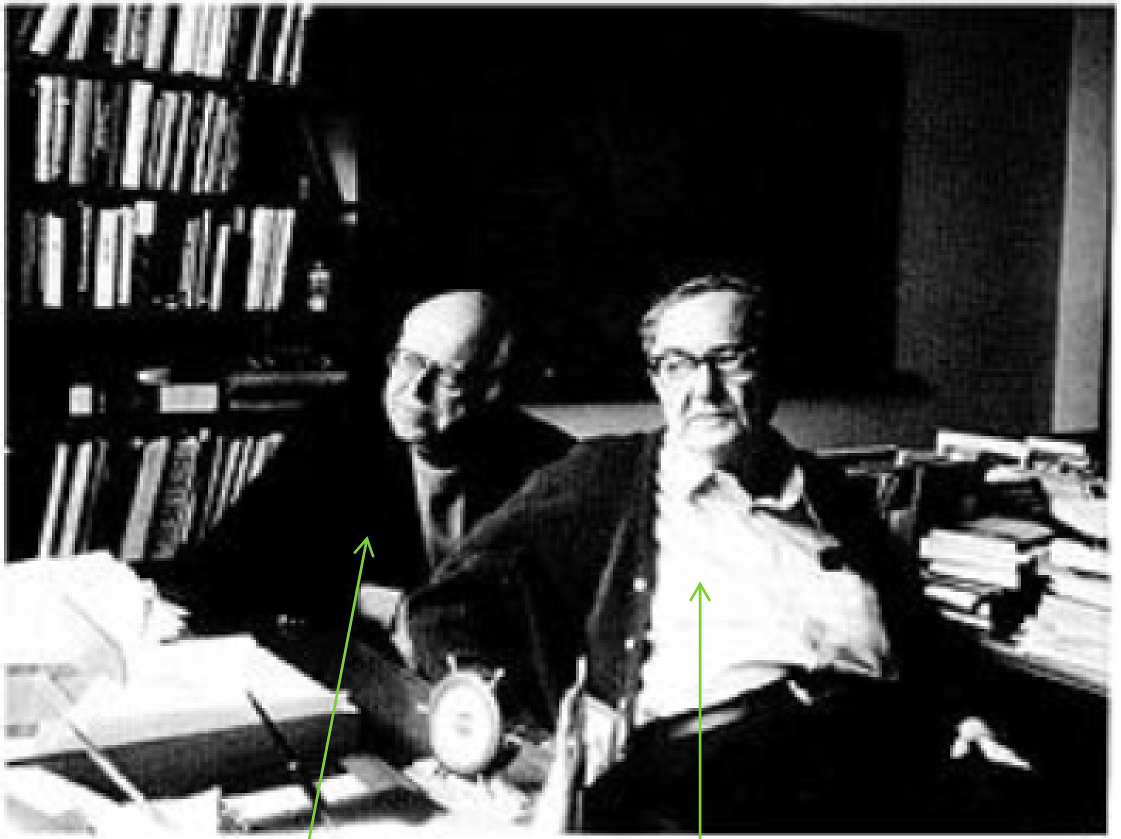
- Many different programming languages such as rule based languages
  - Prolog
  - LISP
  - Other high level programming languages
- Tools
  - Matlab
  - Weka
  - R and other machine learning tools

Cognitive Modeling Techniques

- Two main approaches
  - Physical Symbol System (PSS): The traditional modeling approach using symbols for representation of information and rules for processing those information – E.g. a decision tree
  - Artificial Neural Network (ANN): There is no symbol only connections which as a whole represent a model. An internal node in the network does not represent any information – nodes represent very simple computation but no information
  - Also a hybrid approach of the above two

PSS

- The Physical Symbol System model was introduced by Allen Newell and Herb Simon in 1956



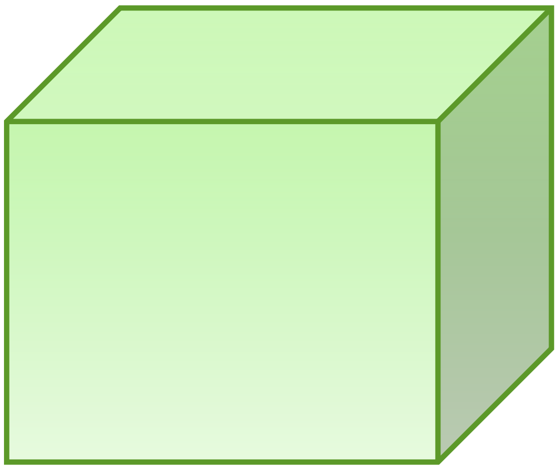
Allen Newell Herb Simon

Physical Symbol System (PSS)

- Input is coded in ***symbolic form.***
- ***Operation is carried out*** on them**.**
- Operations are defined by ***rules.***
- Operations are ***serial; only one at a time but rapidly and repeatedly if needed.***

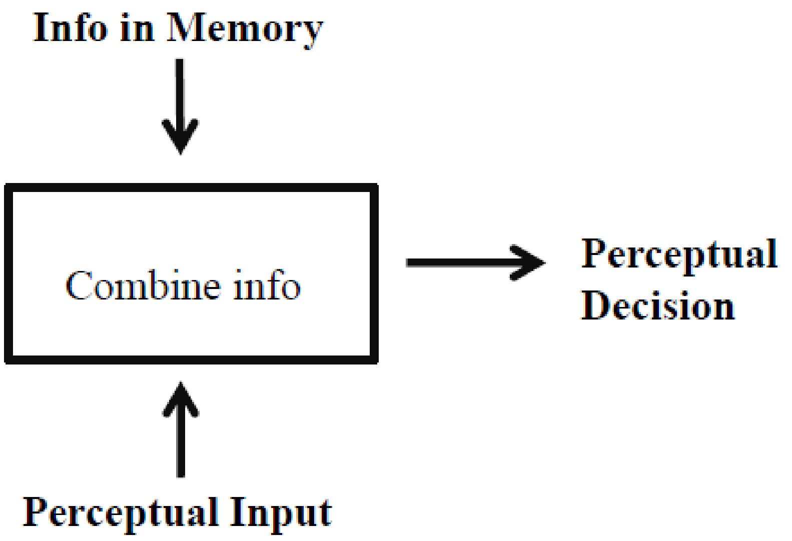
Perception is totally beyond a PSS

- Requires information about multiple perspectives to interpret an image
- Also system requires massive set of rules to be able to put things together again….
- Rules are kind of static – does not typically apply *incremental learning*



Why?

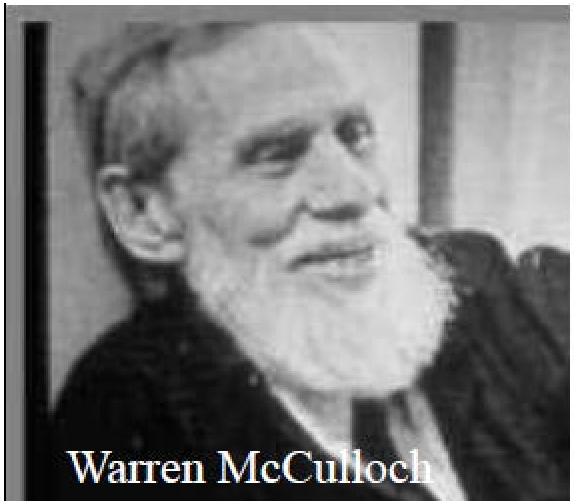
- It is very difficult to get a PSS to "*see*" because of the trouble combining top-down with bottom-up information where information is processed at both levels in parallel …….
  - Feedback loop



Limits of PSS

- Some psychological processes are quite automatic and unconscious – so we have no insight into how they might occur.
- It is very difficult to specify rules for such behaviour.
- We need a different *representation* and *algorithm for dealing with these processes*.

The Alternative Approach…



- Warren McCulloch (1899-1969) was an army psychiatrist who turned into neuropsychologist
- Assembled research group with Walter Pitts, Jerry Lettvin
- Pursued cybernetic model of the brain at MIT

The Foundational Paper

- McCulloch and Pitts, (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5:115-133*.
  - In this paper McCulloch and Pitts tried to understand how the brain could produce **highly complex patterns** by using many basic cells that are connected together.
  - These basic brain cells are called neurons.

Artificial Neuron Model

- McCulloch and Pitts (1943) described a relatively simple unit with the following properties:

1. The unit is either in a *resting state* or *fires*, i.e., binary states.
2. It receives activation from prior units and sends activation to following units.
3. It sums all incoming activation.
4. It fires when a threshold is reached.

Today IBM's SyNAPSE Chip

- Computer chip that is modelled on biological brains
- Not great in what traditional computers excel at.
- But it is far better suited to pattern recognition and processing images, sound and other sensory data.
- Can be used on smart phone-sized circuit board with connectors to hook up cameras or other devices
- Very low power consumption
- Very high parallel computation

Modeling Cognitive Processes

- Distributed parallel processing is required to model human cognitive processes.
  - Perception
  - Learning
  - Language processing
- But for modeling it is necessary to understand the problem.

The Tri-level Hypothesis

- David Marr (1982), neuroscientist, proposed the idea that information processing systems must be understood by three distinct, complementary levels of analysis:

Conceptual Framework

1. *Computational level*: A description of what problems does the system solve and why. E.g. the game of chess.

Cognitive Model

2. *Representation and Algorithmic level*: How does the system solve the problem, specifically what representations are used and what processes are applied to the representations. E.g. how is chess board and pieces and goals represented and what are the rules?

Neural Model

3. *Hardware or Implementation level*: How is the system implemented. E.g. computer program for chess or the effector neurons and muscles in the brain for vision.

Example – Learning

- How do we learn?
  - Gradually, make mistakes, get feedback and correct
  - Learn from reward that doing something can be rewarding
  - Skill learning by doing something repeatedly
  - Use previous knowledge from memory to create analogies and decide about similar situations
- In our brain
  - Initially connections change
  - Then the strength of signals passed through the connections change

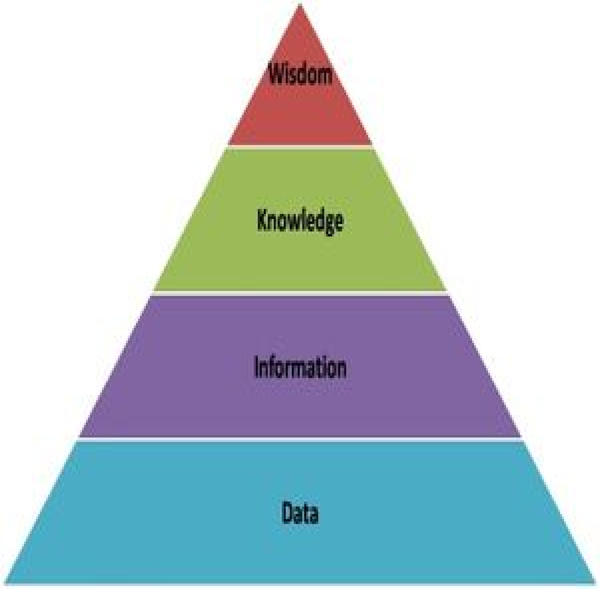
Learning in ANN

- Reorganize or restructure ANN
- Increase connection strength by
  - increasing associated weight
- ANN has 3 main types of learning
  - Error correction learning
    - Learn from mistakes
  - Reinforcement/Correlation/Hebbian learning
    - Learn if rewarded – Named after Donald Heb
  - Competitive learning
    - Variation of Hebbian learning

How to model learning?

The Pyramid of Wisdom

- The Era of Big Data – Data Collection and Integration
- Information – filtered data
- Knowledge – correlated data
- Wisdom based on prediction



Why Learning ?

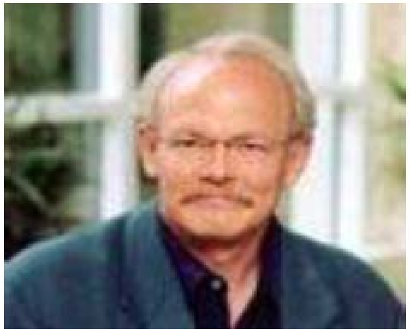
- Not all data are available at once.
- Data represent a state that changes.
- A state may be dependent on many factors – not possible to store models of all different combinations of these factors.
- ANNs represent weighted connections of all these factors where learning mainly changes the weights and sometimes connections.

The First Famous ANN

- The first ANN to achieve prominence, however, was hardwired in advance.
- This is the Interactive Activation Model, presented by McClelland and Rumelhart (1980; Rumelhart & McClelland, 1981)
- ANNs implement Parallel Distributed Processesing (PDP) where processing is done in a distributed network in parallel.

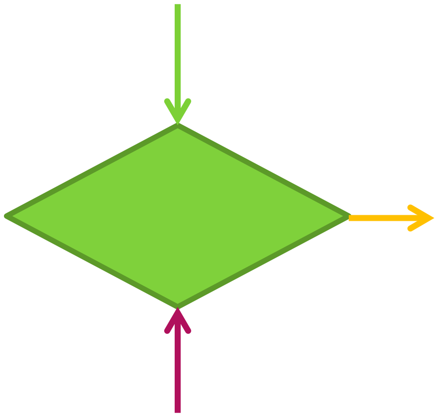
Parallel Distributed Processing (PDP)

- PDP also explains how *top-down and bottom-up information is combined in a perceptual task, like word recognition.*
- Example: Interactive Activation Model



Jim McClelland

Dave Rumelhart



Learning ⟷ Perception

- Perception is modified by learning
  - Two effects show that perception is modified by what you learn as you learn to read….
    - *Word Apprehension Effect (WAE):* A word is easier to report than an equal number of unrelated letters.
    - *Word Superiority Effect (WSE):* A four letter word is easier to perceive than a single letter.
- Perception is assisted by *context;* we use what *we know to improve what we see*.
  - *Top-down is combined with bottom-up.*

IAM → WAE and WSE

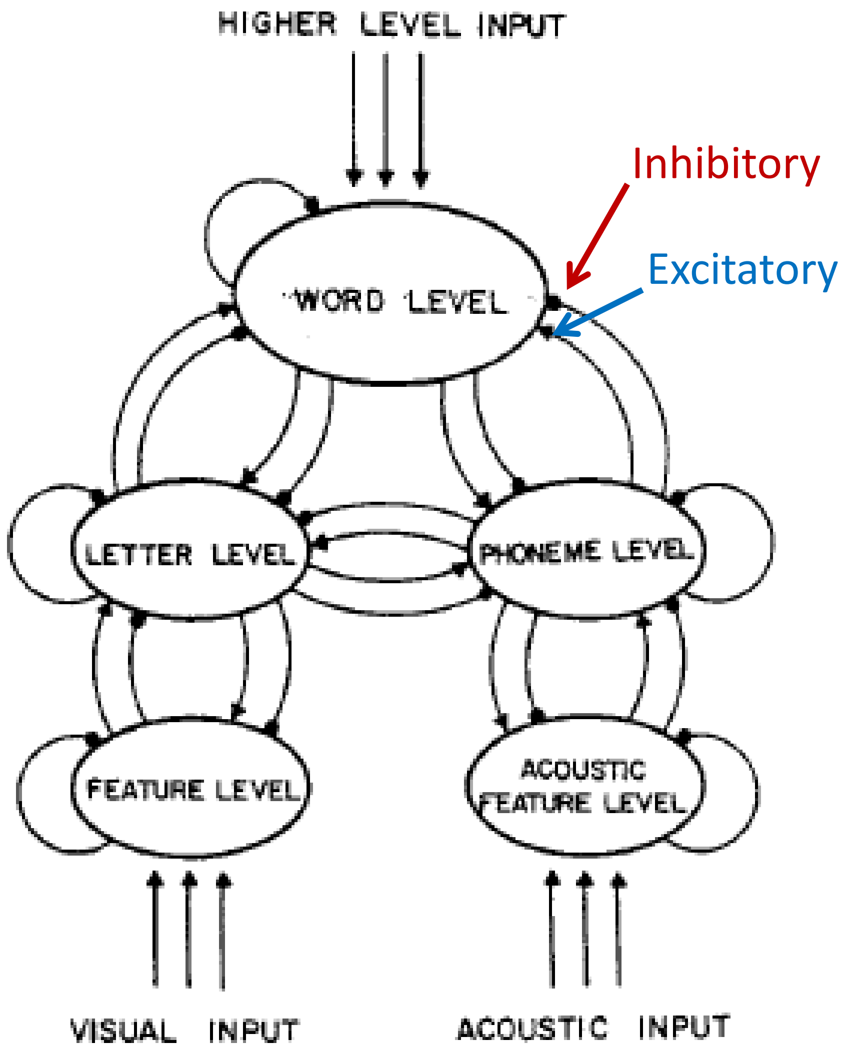
- IAM was designed to account for the data collected by Reicher (1969), that is, it should produce a *WSE* when a mask is used.

  E.g., 

- It was not designed to account for the *Word Apprehension Effect (WAE)* but it should explain that just as well if the model reflects human processes…

Word Perception

- Some of the processing levels involved in visual and auditory word perception with interconnection (McClelland and Rumelhart, 1981).
- Assumptions: 3 levels where top and bottom level gets input from higher level and visual and acoustic inputs accordingly.



Input Features for Letters

- Simplified feature analysis of input font
- Limited lexicon
- Input is which of the visual feature detectors are on for each letter

Font and feature analysis process is from Rumelhart 1970 and from Rumelhart and Siple 1974.

IAM – Context Effects in Letter Perception

- IAM is a model of context effects in perception of letters
- Perception results from excitatory and inhibitory interactions of detectors of visual features, letters, and words.

1. Visual input excites detectors for visual features in the display
2. Active features *inhibit other feat*ures and
   - Activate those letters which contain the features and
   - Inhibit letters which do not contain the features.
3. Active letters *inhibit other letters* and
   - Activate words which contain the letters and
   - Inhibit words which do not contain the letters.

Feedback Cycles in the Model

4. Active word detectors mutually *inhibit* each other then
   – Send feedback to letter level to activate the letters they contain and inhibit letters they do not contain
   – Feedback strengthens perception of constituent letters.
5. Active letter detectors
   – Send feedback to feature level to activate the features they contain and inhibit features they do not contain
   – Feedback strengthens perception of constituent features.
6. And thus it cycles between the feature, letter and word…

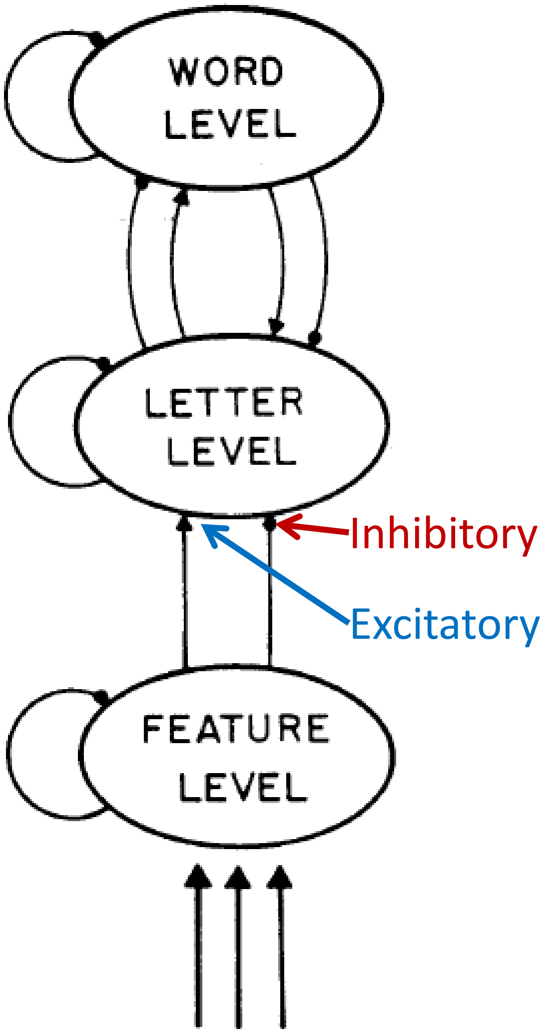
- Letters in words are more perceptible because they get the reinforcement activation from word level

Assumptions in IAM

- Multi-level processing – word, letter, feature
- Visual perception involves parallel processing
- Perception is an *interactive* process – top-down (conceptually driven) and bottom-up (data-driven) works simultaneously where interfering constraints determine the final result of perception.
- Only consider *excitatory* and *inhibitory* activations.

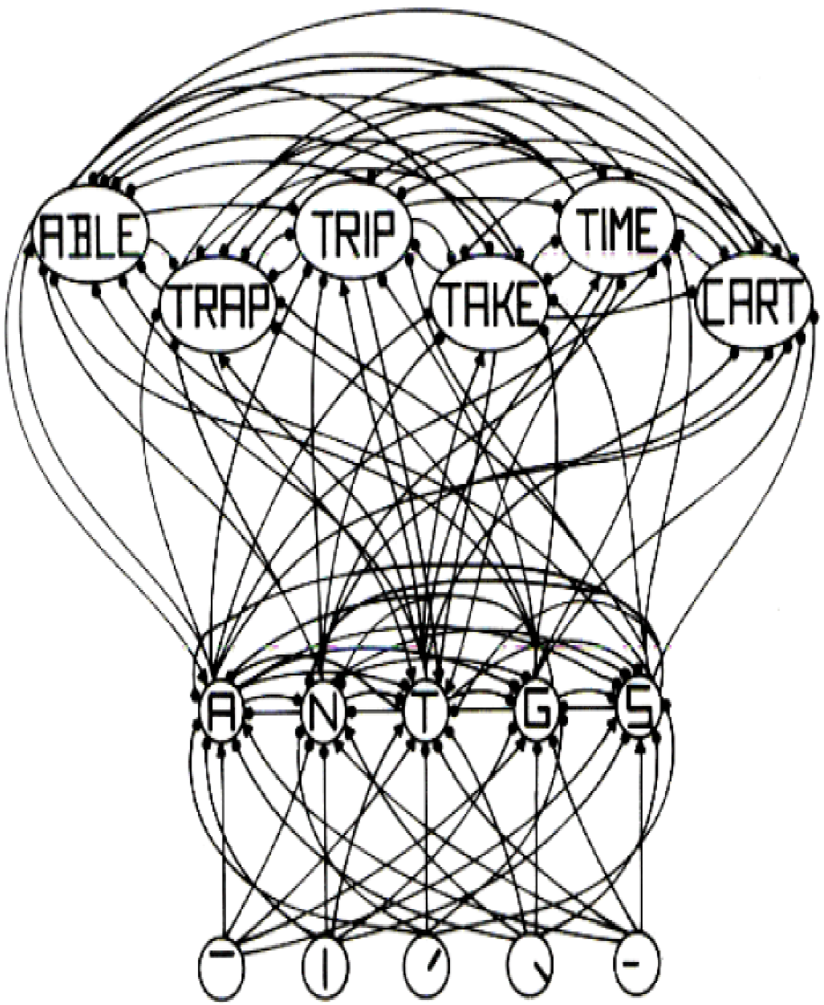
IAM – Simple Model – Part 1

- To make it simple and reduce interactions, initial model (part 1) considers
  - Reciprocity of activation between word and letter in paper on part 1.
  - Ignores phonological processes
- Features are given as inputs.
- Discrete rather than continuous time.
- A word and letter level node connects with neighbours at the same or adjacent levels word or letter node.



IAM Model – Example for ‘T’

- Only consider a subset of the neighbours of the letter *t*.
- Activation is a positive real value at time *t*, given by $a_i(t)$
- In absence of activation, a previously activated node decays back at a rate θ_i to inactive state with a resting state $a <= 0$.
- Nodes for high frequency words have resting levels higher than low frequency words but varies by r_i.



How to model this using PSS?