Day 21: RAAM: Recursive Auto-Associative Memory

> 💡 This is the Professor's specialty, so it will *definitely* be on the exam.

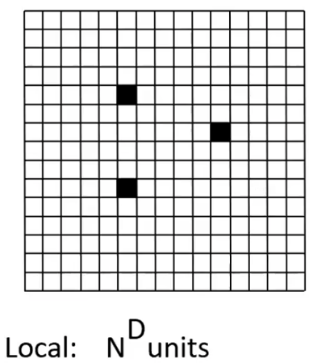
- Major breakthrough in Neural Networks about how to store things
- Minsky & Papert's General Principle for Machine Learning (and against Perceptrons), 1969:

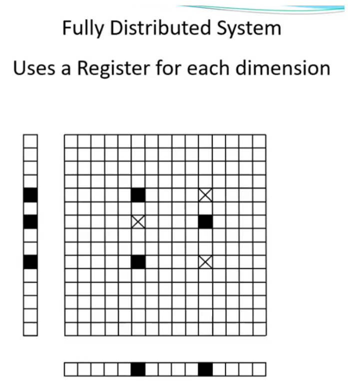
  > No machine can learn to recognize X unless it possesses, at least potentially, some scheme for representing X.
  >
  > Perceptrons had no way to represent knowledge.

  - This is not strictly speaking true, but close enough
- Fodor & Pylyshyn (1988), philosophers, came up with the following:

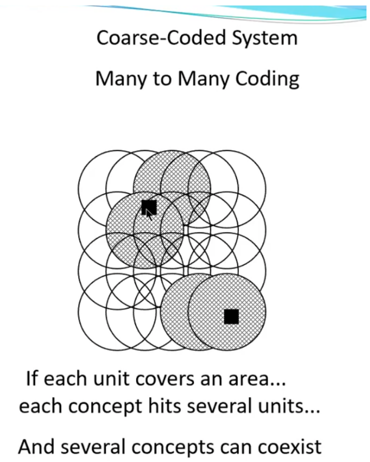
  > The Language of Thought is necessarily compositional, combinatorial, and systematic. The only known adequate systems are rule- or syntax- based. Therefore, Connectionism is a mere implementation detail, or wrong.

- Distributed Representations: A representation a horse should include a representation of a horse's leg.
  - However, the network can't be infinitely big, which implies an activation pattern per concept instead of a neuron per concept
  - Different types of representations:





Fully Distributed produces phantom activations, shown by the `x`s.



- Coarse Coding for NLP: tuples of letters

Represent a sequence by overlap:

#STORE#

is represented by collection:

#ST + STO + TOR + ORE + RE#

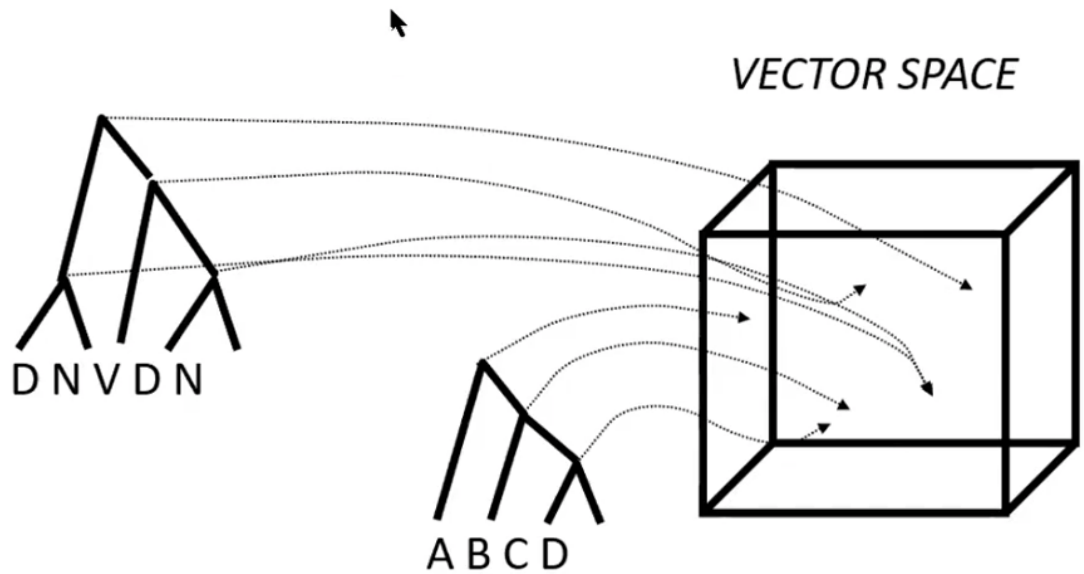
Problem: Can't represent `BANANA`, because ANA can only be activated once. (There are counter example for any tuples, quadruples, etc.)

The Professor's Solution: RAAM

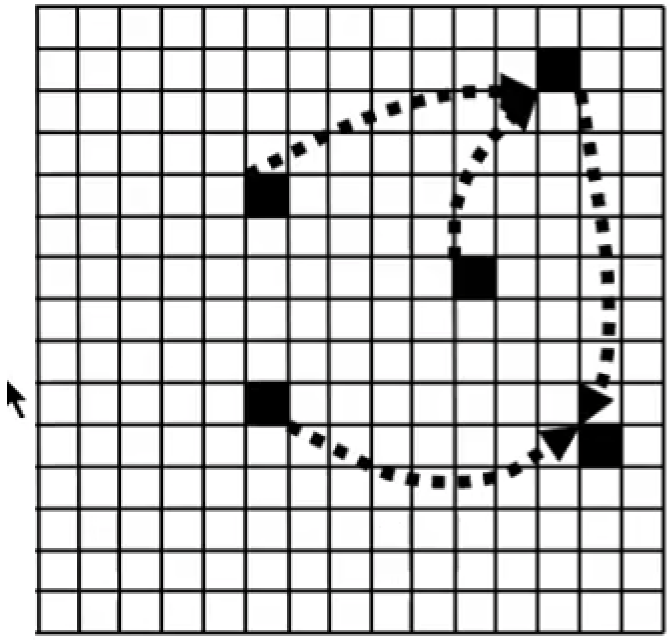
- Automatic way to develop representations for sequences and trees

- Also develops access mechanisms

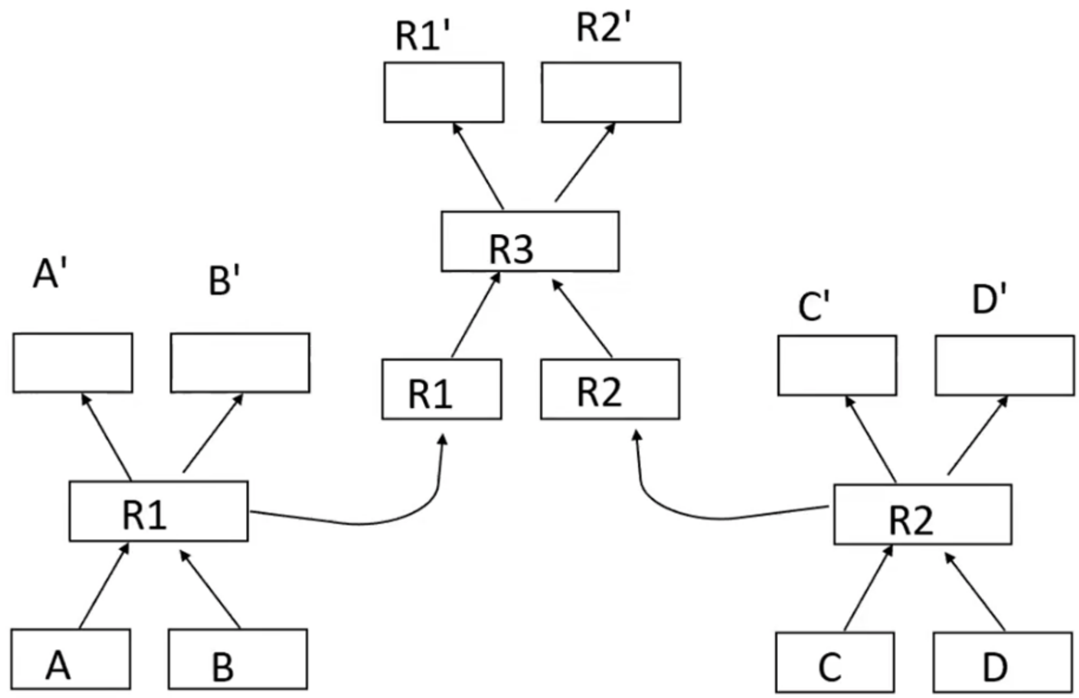
- Built on top of BP's ability to learn with hidden units
- Packs symbolic structure representations (parse trees, etc.) into vector space:



- Each leaf has a specific vector representation, and therefore so does each non-leaf node. (Store the leafs, compute the hidden nodes.)



- To do this, conceptualize two machines, an Encoder/Constructor and a Decoder/Reconstructor. The encoder combines many partial representations into a whole one, while the decoder does the reverse.
    - To properly deconstruct, you also need a test that checks if a value is a terminal node or a hidden node
        - This was done by having each input value be either 0, 1. A node was considered terminal if all outputs were within 0.2 of either 0 or 1.
    - To do this, he used an auto associator (where the first half is the encoder and the bottom half is the decoder)
    - With infinite training, eventually this converges to being correct
    - Example for training on the `A, B, C, D` (which form a tree)



- Exploring RAAMs:
  - Bottom-up search to see how much it can represent (try all pairs, all triples, etc.)
- RAAM produced lots of philosophical papers
- Challenges of RAAMs:
  - Scaling to more than six levels (requires either very tight convergence or a different design methodology)
  - Can theoretically process infinite language, with infinitely-precise rationals
  - Need a killer application that proves RAAMs are the best:
    - Speech, handwriting, vision, etc.
  - How to learn from statistics of data instead of full data