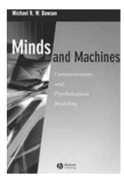
Psychology 452
Week 9: PDP Interpretation: Coarse Coding

- Coarse coding
- Coarse coding in the balance scale problem
- Coarse coding in the kinship problem
- Allocentric coarse coding: From a PDP network to the hippocampus

Course Structure

When	What
Weeks 1, 2, 3	Connectionist Building Blocks
Weeks 4, 5, 6	Case Studies of Connectionism
Week 7	Midterm Exam
Weeks 8, 9, 10	**Interpreting Connectionist Networks**
Weeks 11, 12	Deep Learning Basics
Week 13	Final Exam

Chapter 5 Discussion

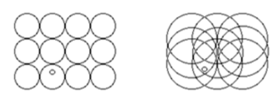


- Questions?
- Important Terms
  - Computer simulation
  - Production system
  - Cryptarithmetic
  - Input unit
  - Hidden unit
  - Output unit
  - Net input function
  - Activation function

Detecting Spatial Properties

- Consider the two coding schemes below
- Note how it is possible to obtain fine spatial resolution by combining the responses of poor spatial detectors

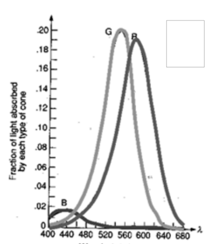
Representation



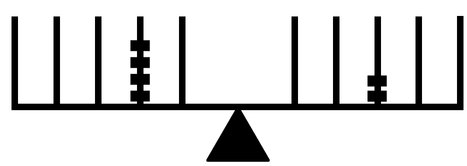
Larger receptive fields provide better resolution (when whole pattern is considered)

Coarse Coding

- Coarse coding requires that a property be encoded by a set of detectors
- Usually the detectors will have overlapping sensitivities
- Many examples of this type of coding are found in the human visual system
  - Colour detection
- Do PDP networks coarse code as well? Is this where "distributed" comes from?
- Let's explore these questions by considering an example network



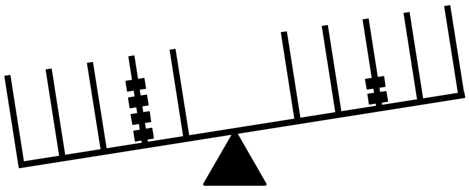
The Balance Scale Task



- Which way will the scale tip?
- Or will it balance?
- This task was originally described by Inhelder and Piaget in 1958

Piaget & Inhelder

The Balance Scale Task



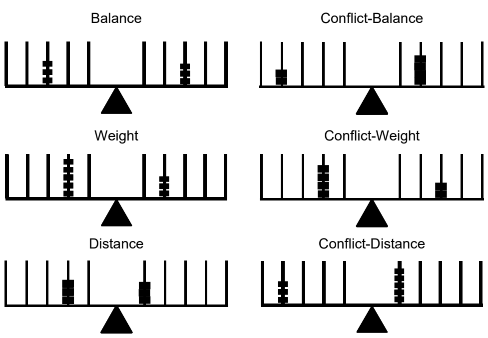
- This example tips to the left
- How do children reason about this kind of problem?

Development Of Reasoning

- The balance scale task has been studied by developmental psychologists for half a century
- Siegler provided a typology of balance scale problems
- Performance on the different problem types varies with age

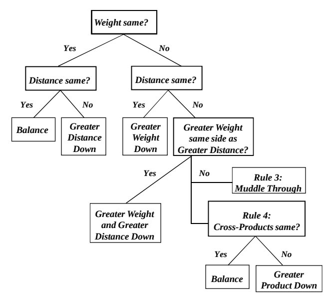
Robert Siegler

Siegler Typology



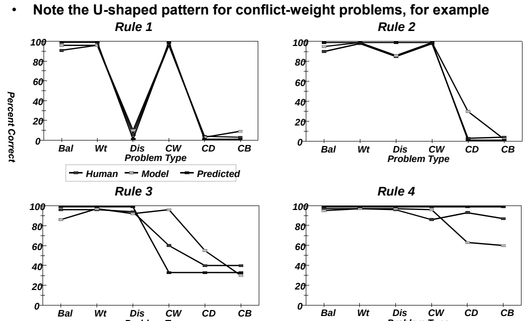
Rule Assessment Model

- Siegler proposed a decision tree model of rules that was an attempt to describe the developmental sequence observed with this task
- Performance on the task depended on what level (rule) was being used



Developmental Trends

- Performance changes as new rules are exploited
- Note the U-shaped pattern for conflict-weight problems, for example



Modeling

- Many researchers have developed classical and symbolic models of this task
- These models have not been synthetic!
  - Fit to data
  - Rule assessment methodology
  - Match Siegler's predictions
- "Regardless of the learning algorithm that one adopts (connectionist or symbolic), the choice of attributes to use is crucial *if the model's output is to match the human data*" (Schmidt & Ling, 1996, p. 211, emphasis added).

Bill Schmidt

Charles Ling

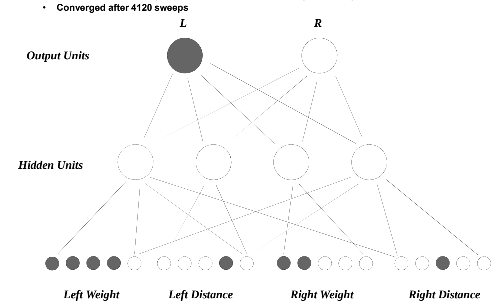
Synthetic Approach

- There are lots of reasons to suspect the Siegler approach
- Why not distance ourselves from it, then?
- Dawson and Zimmerman (2003) explored a synthetic approach to the balance scale problem
  - Build a network
  - Don't fit data
  - Interpret the network
  - What new things do you learn about the balance scale task?

Corrine Zimmerman

Balance Scale Network

- A network of value units was trained to solve the balance scale task
  - Thermometer encoding of weight, local encoding of peg location (distance)
  - 625 problems, learning rate 0.005, biases start at 0, weights in range ±0.1
  - Converged after 4120 sweeps



An Additive Rule

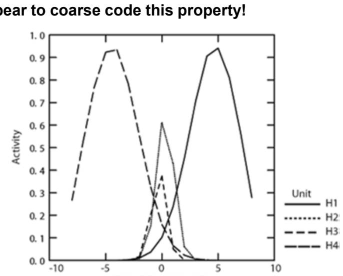
- Optimal rule for the task is the torque rule
  - (LW x LD) – (RW x RD)
- This rule can't be a primitive because of the multiplication of inputs
- A plausible alternative is an additive rule
  - (RW + RD) – (LW + LD)
- The two rules are highly correlated

Sensitivity To Torque Or Additivity

- Do the hidden units serve as tools that compute torque or additivity?
- We correlated hidden unit activity with both of these measures for all 625 patterns
  - Torque rule correlations:
    - H1:0.92, H2: 0.92, H3: -0.87, and H4: –0.92
  - Additive rule correlations:
    - H1:0.97, H2: 0.97, H3: -0.92, and H4: –0.97
- Hidden units are most sensitive to the additive rule
- Additive rule is a good approximation to the torque rule that can be computed by hidden units
- But why are 4 hidden units required?

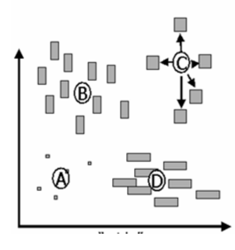
Coarse Coding: Additive Rule

- Hidden units are not uniformly sensitive to the additive rule
- They appear to coarse code this property!



Clustering Approach

- When we find coarse coding in networks, we want to find regularities distributed across hidden units
- To do this, we perform cluster analysis on hidden unit activities
  - K-means
  - Assign patterns to clusters
  - Make assignments to minimize distances within cluster
- We then can examine the properties of patterns that fall into the same cluster, looking for commonalities
- This is very similar to our approach to making local interpretations of value unit bands

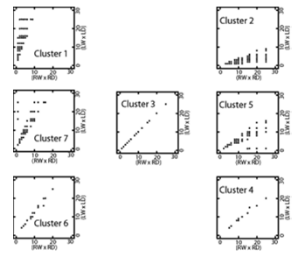


How Many Clusters?

- With clustering, we need a stopping rule to determine how many clusters
- With our networks, we use the following heuristic:
  - Choose a value for k (starting small)
  - Perform the cluster analysis
  - Examine cluster membership
    - If each cluster is "pure", so that all members yield the same network response, then stop
    - If clusters are not "pure", then k is too small – increase k, and repeat the procedure

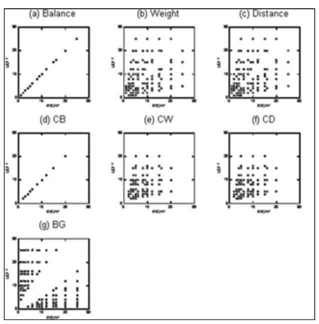
Clustering Hidden Unit Activities

- Using this rule with the balance scale network, we obtained 7 different clusters
- These clusters were amazingly regular when plotted in a two-dimensional additive rule space



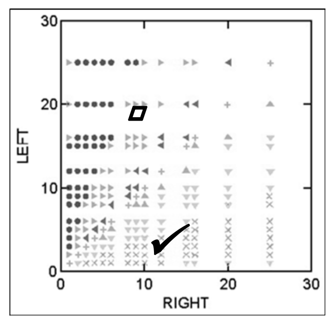
Clustering Siegler Patterns

- The Siegler classification of patterns is much less systematic when plotted in this same pattern space
- The network has delivered a new, sensible typology!



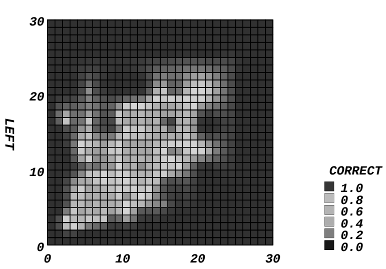
Making Experimental Predictions

- The cluster analysis can be used to make predictions about the type of problems that should affect accuracy and time for human subjects
- RT and error should be affected by position of the problem in the space
- Poorer performance nearer the diagonal of the space is predicted



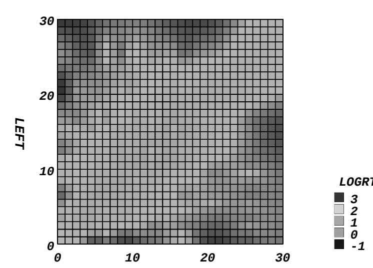
Accuracy Results

- Plotting proportion correct as a function of problem position in the space supports the prediction



Latency Results

- Plotting reaction time as a function of problem position in the space also supports the prediction

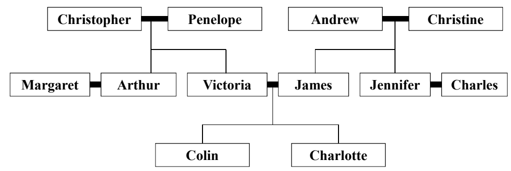


Implications

- Synthetic approach did not fit data
- The network provided in a new typology of problems
- The network provided a new rule for solving the problem
- The network revealed how hidden units could solve the problem via coarse coding
- The network generated new experimental predictions

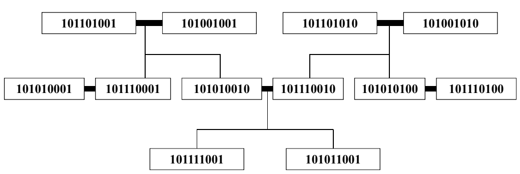
Coarse Coding: Example 2

- Hinton's kinship problem
- Ask a network about a name and a relation
- Network outputs a name
- "Who is James' father?" "Andrew"

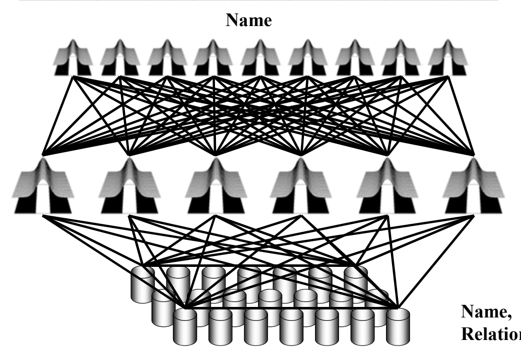


Network Representation

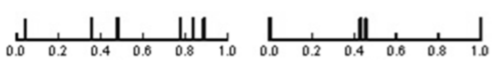
- 21 inputs, 6 hidden, 9 output
- 9 bit code for name
  - (family, gender, generation, person)
- 12 bit unary code for relation
  - (nephew, niece, aunt, uncle, brother, sister, father, mother, daughter, son, wife, husband
- 6 families, 52 queries per family, 312 patterns



The Kinship Network

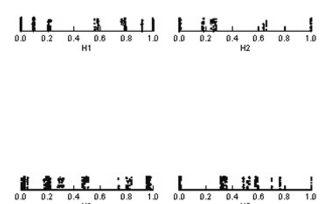


Family Detectors



In each of these units, every band represents a single family. But note that identification of "family" requires both hidden units!

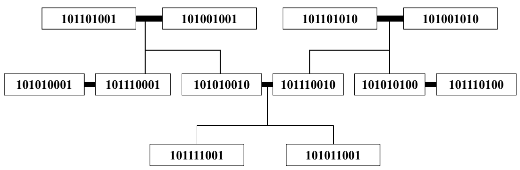
Tree Regularity Detectors



In each of these units, bands represent groups of individuals within a family tree. However, local interpretations of any band does not identify an individual!

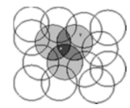
Example Band

Hidden Unit 3, Band D, N = 24
wife or husband of person 010 in generation 1,
or
father or mother of person 010 in generation 2

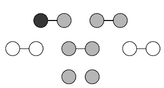


Coarse Coding

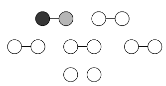
- How are these broad categories of individuals used by the network?
- Individuals are represented by coarse coding!
- One person falls out of the intersection of different bands in different hidden units



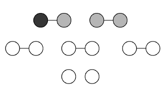
Example Intersection



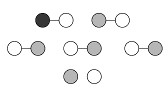
H1 Band A



H2 Band B



H3 Band D



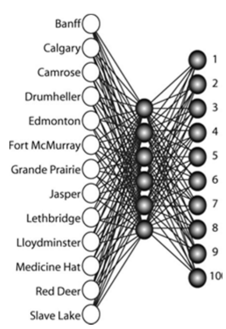
H5 Band A

Coarse Coding: Example 3

- We were interested in representations of space
- We wanted a psychologically relevant task
- We decided to train a network to rate the distances between cities in Alberta

The Alberta Network

- We used the value unit architecture
- Local coding of input cities and output ratings
- Smallest network that worked used 6 hidden units
- 169 training patterns

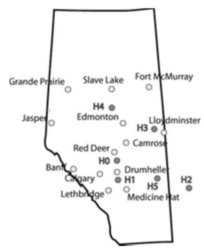


Network Analysis

- Traditional network analyses did not work very well with this network
- We decided to explore the relationships between hidden unit properties (activities and weights) and map distances
- Much of this analysis required us to use optimization tools to locate hidden units on a map in order to maximize the relationship between the map and the network
- Lots of details are provided in Dawson, Boechler, and Valsangkar-Smyth (2000), which is available from my lab web site

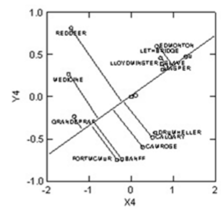
Hidden Units On The Map

- Hidden units could be placed on the map
- Position maximized correlation between weight and distance
- Near perfect correlations when map was "distorted"



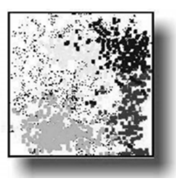
Hidden Unit Properties

- Hidden units were metric
- Individual hidden units, though, had a very inaccurate internal map
- Accuracy of space came from *coarse allocentric coding*!



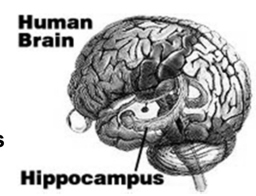
Place Cells

- Researcher's have argued that the hippocampus instantiates Tolman's cognitive map
- Place cells fire only when an animal's head is at a certain position in the environment



Is The Hippocampus A Map?

- Place cells are not topographically organized
- Place cells are at best locally metric
- Hippocampus does not seem very "maplike"!



Place Cells And Coarse Allocentric Coding

- Our network is not maplike either, but has internalized a map of Alberta
- Hidden units are like place cells
- Perhaps the hippocampus is a PDP map, using coarse allocentric coding