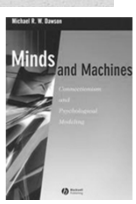
Psychology 452
Week 8: Local Interpretation Of Networks

- **Network Interpretation**
- **Examining Connection Weights**
- **Local Analysis Of Bands**

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 4 Discussion



- **Questions?**
- **Important Terms**
  - **Mathematical model**
  - **Pavlovian conditioning**
  - **Classical conditioning**
  - **Blocking**
  - **Rescorla-Wagner model**
  - **Recursive equation**
  - **Extinction**

PDP Models Are Hard To Understand

- **This is because they are nonlinear, large, messy, and often unstructured**
- **"One thing that connectionist models have in common with brains is that when you open them up and peer inside, all you can see is a big pile of goo" (Mozer & Smolensky, 1989)**
- **Problems of network interpretation might limit connectionist contributions to cognitive science**

Michael Mozer

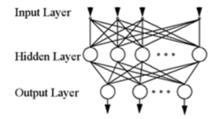
Paul Smolensky

Responding To McCloskey

- How do you interpret networks?
- Statistical analyses of network connectivity
    - Hanson & Burr 1990
    - Dawson 2003
- Map out the network as we would the brain
    - Moorhead, Haig & Clement 1989
    - Dawson, Kremer & Gannon 1994
    - Berkeley, Dawson, Medler, Schopflocher & Hornsby 1995

Strategy 1: Analyze Weights

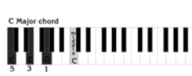
- A trained network has very few things to look at:
    - Processor weights and biases
    - Processor responses to stimuli
- What can be learned about the nature of a network by focusing our attention on the properties of its weights?



The Music Chord Problem

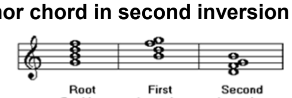
- One important task in music theory training and piano technical training is chord identification
- Example: read a chord
    - What general type of chord is it?
    - What is its key?
    - What is its inversion?

- Example: listen to a chord
    - What general type of chord is it?
    - Independent of key
    - Independent of inversion



Major And Minor Chords

- The training set used 12 different major keys
    - Major chord in root position
    - Major chord in first inversion
    - Major chord in second inversion
- The training set used 12 different minor keys
    - Minor chord in root position
    - Minor chord in first inversion
    - Minor chord in second inversion



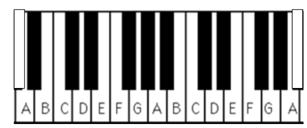
Dominant And Diminished 7ths

- The training set used 12 different major keys
  - Dominant 7[th] chord for each key
  - Root position and all inversions

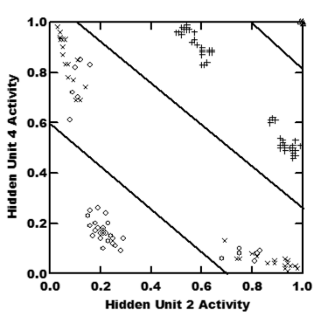
- The training set used 12 different minor keys
  - Diminished 7[th] chord for each key
  - Root position and all inversions

The Music Chord Network

- 4 output processors (value units)
  - Major chord
  - Minor chord
  - Dominant chord
  - Diminished chord
- 4 hidden processors (value units)
- 24 input units
  - Piano keyboard
  - Two octaves
  - Starting note is A
- 192 training patterns
- Dawson/Schopflocher rule
  - Learning rate of 0.005
  - Weight start ±0.01
  - Biases start at 0.00
- Converged after 5392 epochs



Preliminary Results



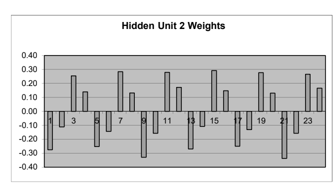
- Preliminary analyses are used to focus later interpretation
- Discriminant analysis indicated that hidden units 2 and 4 could solve 94% of the problem.
- Only made mistakes with 2[nd] inversion of major chords
- How so good?
- Why the problem?

Chord Type

- △ Diminished
- × Dominant
- + Minor
- ○ Major

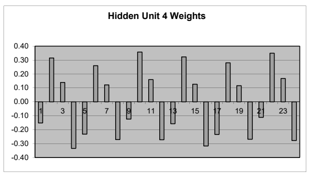
Hidden Unit 2 Connections

- An examination of hidden unit 2 weights indicated repeated use of the same values
- If connection weights represented note names, then this unit used 4 instead of 12!



Hidden Unit 4 Connections

- Hidden unit 4 represents notes with a very similar scheme to that used by hidden unit 2!



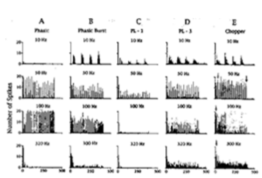
New Theory Of Music

- The connection weights grouped notes into the same category
- No three notes in a group would ever co-occur
- They are equally spaced on the keyboard
- The sum of note “names” identifies chord type
- Misses are cleaned up by the remaining two hidden units

Note Name	Hidden Unit 2	Hidden Unit 4
A, C#, F	-0.29	-0.17
A#, D, F#	-0.13	0.31
B, D#, G	0.28	0.14
C, E, G#	0.15	-0.29

Strategy 2: Examine Unit Responses

- In many cases, individual connection weights will not be very useful
- What may be more useful is examining the effect of many weights combined
- That is, wiretap the units and look at responses



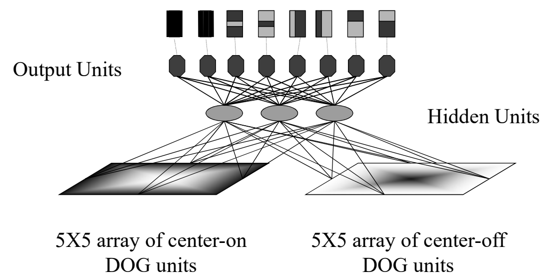
Moorhead, Haig, Clement

- Could a multilayer perceptron be trained to carry out some of the functions of the early visual system?
- If so, then what would its internal representations be like?
- What would be the relationship between this network and the biological visual system?

General Method

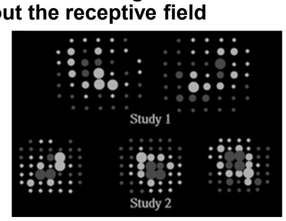
- **Train a PDP network to detect oriented edges and lines**
  - **Input units = retinal ganglia**
  - **Hidden units = parvocellular LGN neurons**
  - **Output units = simple cells**
- **Key issue: do hidden units adopt center-surround receptive fields?**

The Network



Brain-like Treatment

- **Moorhead, Haig and Clement treated the network like the brain when they examined its internal representations**
- **They spotmapped the receptive fields of the hidden units, by measuring the unit's response as a small stimulus "light" was moved throughout the receptive field**

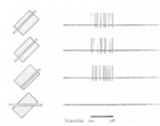


Conclusions

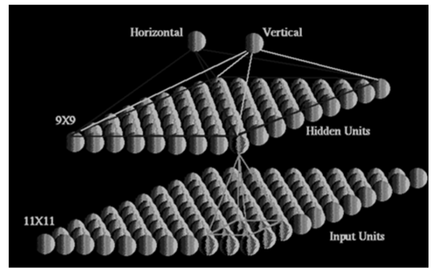
- **No center-surround receptive fields found**
- **"There is no direct equivalence between the retinogeniculate striate pathway and a neural network that has been trained to respond in a manner similar to simple cells" (p. 802).**
- **But … lots of potential problems:**
  - **Why prefilter images?**
  - **Why so few hidden units?**
  - **Why violate limited order constraint?**
  - **Why pass the stimulus through the center all the time?**
- **Dawson, Kremer & Gannon (1994) tried to fix these problems, and use a different interpretative strategy too**

A New Approach

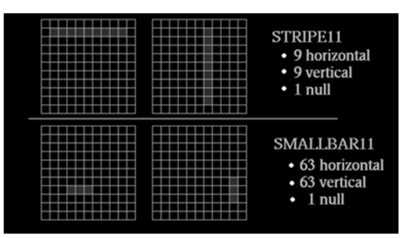
- Train output units as complex cells -- sensitive to orientation anywhere on the retina
- Do hidden units develop simple cell receptive fields?
- Impose the limited order constraint



A New Network



Example Stimuli

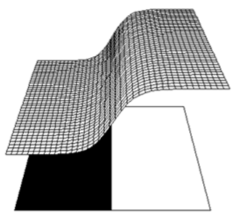


Trigger Feature

- According to Barlow (1972), a trigger feature is the pattern (presented to a cell's receptive field) that produces a maximum response in a cell
- Barlow's neuron doctrine called for a search for trigger features
- What is the trigger feature for an integration device?

Horace Barlow

Integration Device



- •Monotonic
- •Therefore only one trigger feature
- •Maximum input for positive connections
- •Minimum input for negative connections

The Kremer Rule

- For an integration device, find the pattern that has the maximum input through every positive weight, and the minimum input through every negative weight
- This is the trigger feature for the unit

-0.12	2.12	-0.34
-0.56	3.15	-0.25
-1.13	1.13	-0.89

-0.12	2.12	-0.34
-0.56	3.15	-0.25
-1.13	1.13	-0.89

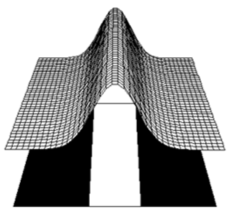
Results

- We would only expect by chance 2 simple cell receptive fields
- In our two studies we found 13 and 27 such hidden units -- highly significant -- but only when the limited order constraint was imposed

The Trouble With Triggers

- By definition, a cell should only have one trigger feature
- But doesn't describing a cell in this way throw lots of information away?
- Isn't it possible that a family of patterns might serve as triggers for a unit, or that distributions of activities of many patterns are important for interpretation?

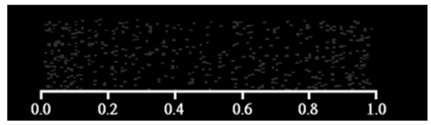
Triggers For A Value Unit



- With a mean of 0, any net input lying in the plane orthogonal to the input weights is a trigger feature
- Value units require considering families of inputs!

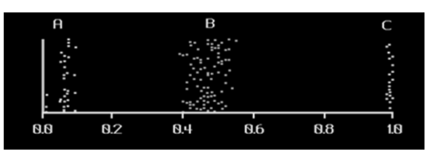
Jittered Density Plot

- One plot per hidden unit
- One point per pattern
- Horizontal location = activity
- Random vertical location prevents overlapping points



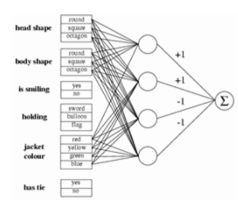
Banded Density Plots

- The jittered density plot for a value unit often reveals distinct, interpretable bands
- Patterns that fall in the same band share definite features



The Monks Problems

- Standard benchmark in machine learning literature
- Classify "monks" on basis of some general charcteristics
- Important because it is one of the few problem types that researchers have used to compare different architectures.

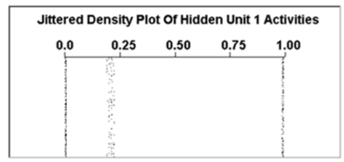


First Monks Problem Network

- One output value unit
- Two hidden units
- 15 input units representing monk characteristics
- 432 training patterns
- Dawson/Schopflocher rule
  - Learning rate of 0.01
  - Weight start ±0.1
  - Biases start at 0.00
- Converged after 22 epochs

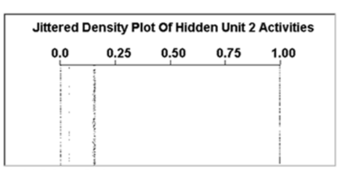
Wiretaps Of Hidden Unit 1

- Hidden unit 1 was wiretapped
- A jittered density plot revealed 3 bands



Wiretaps Of Hidden Unit 2

- Hidden unit 2 was also wiretapped
- It had a similar banded structure in its jittered density plot



Definite Features

- Definite features were revealed in the bands
- These features could be used to solve the first Monks problem

Unit	Band	Definite Feature	Interpretation	Implication
H1	A	Input 3 = Input 6	Eh?	Eh?
	B	Input 11 = 1 Inputs 12, 13, 14 = 0	Jacket red	In target class
	C	Input 11 = 0 Input 3 ≠ Input 6	Jacket not red Different body and head shapes	Not in target class
H2	A	Input 11 = 1 Inputs 12, 13, 14 = 0	Jacket red	In target class
	B	Input 2 = Input 5	Eh?	Eh?
	C	Input 11 = 0 Input 2 ≠ Input 5	Jacket not red Different body and head shapes	Not in target class

Distributed Features?

- Are local features enough?
- Some of the bands seem distributed
- Network response involves both hidden units considered at the same time
- Dealing with this situation is the topic of next week's lecture