Psychology 452

Week 1: Connectionism and Association

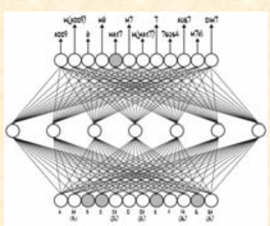
Course Overview
Properties Of Connectionism
Building Associations Into Networks
The Hebb Rule
The Delta Rule

Michael R.W. Dawson

- PhD from University of Western Ontario
- Research interests in foundations of cognitive science, artificial neural networks, embodied cognitive science
- Research methods include computer simulation and LEGO robot fabrication
- For details about my research, go to my home web page

Course Objectives

- Explore the foundations of connectionist cognitive science
- Provide "hands on" experience with various artificial neural networks
- Use this experience to consider the role of connectionism in cognitive science

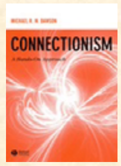


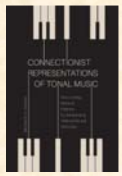
Course Trajectory

When	What
Weeks 1-3	Basics of three architectures (DAM, perceptron, MLP)
Weeks 4-6	Cognitive science of DAMs and perceptrons
Week 7	Connectionism and Cognitive Psychology
Weeks 8-10	Interpreting MLPs
Weeks 11-13	Case studies (interpretations, applications, architectures)

Course Evaluation

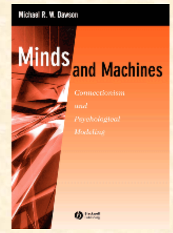
- Scaffolded Network Project (50%)
  - Report of simulation study in which you train a network to do something you are interested in
  - If you prefer, could be a review paper relating networks to some topic that you are interested in
  - Handed in in four stages
- Text Assignments (40%)
  - Handed in weekly
- Class Participation (10%)
- Texts:
  - Dawson, M.R.W. (2004) *Minds and Machines*. Oxford: Blackwell (on website)
  - Dawson, M.R.W. (2005) *Connectionism: A Hands-On Approach.* Oxford: Blackwell (on website)
  - Dawson, M.R.W. (2018) *Connectionist Representations of Tonal Music* Edmonton: AU Press





Meeting Plan

- Start with lecture
- Quick break
- Move on to activity/discussion
  - Starting on Week 4 this will involve going through Dawson (2018)
  - Case studies of using the networks that we learn about
  - Hands-on demonstrations of network training and network interpretation

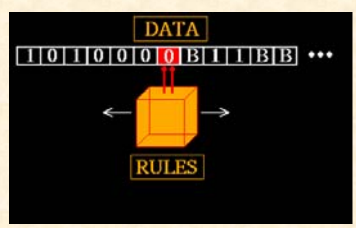


Course WWW Support

- Lots of WWW support
- Lectures, additional readings, information about assignments, pointers to other sites of relevance
- Software, training sets etc for assignments only available on the web
- http://www.bcp.psych.ualberta.ca/~mike/Pearl_Street/PSYCO452/
- We will be exposed to this website in more detail during our hands-on activity later this evening

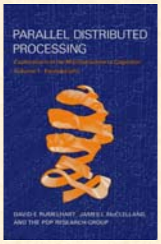
The Classical Approach

The Classical approach adopts a strict "structure/rule" distinction in its view of information processing. It views cognition as the rule-governed manipulation of symbols.



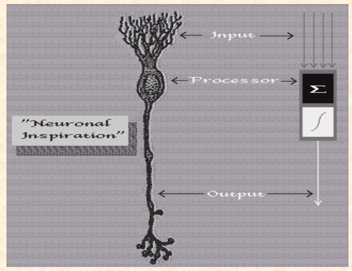
Connectionism

- Since the 1980s there has been an explosion of interest in parallel distributed processing (PDP) or connectionist architectures
- These architectures have been developed to solve a number of possible problems with classical accounts of cognitive science



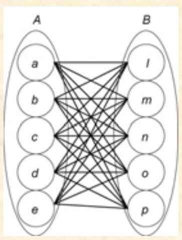
Neuronal Inspiration

- PDP modelers pay more attention to the brain than do Classical researchers
- A PDP processor can be viewed as an abstract, simplified description of a neuron



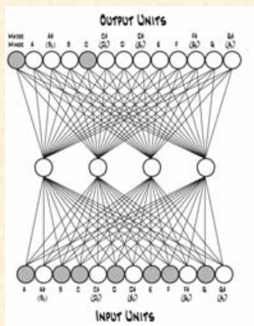
Parallel Processing

- PDP models are networks of simple processors that operate simultaneously
- This causes fast computation, even if components are slow
- This is intended to fix the speed limitation of Classical models



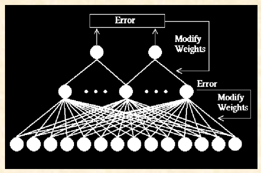
Distributed Representations

- A PDP network's knowledge is stored as a pattern of weighted connections between processors
- These connections are analogous to a Classical program
- This knowledge is very distributed, providing damage resistance and graceful degradation



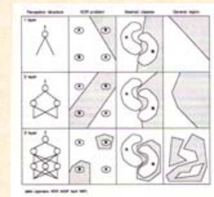
Networks Learn

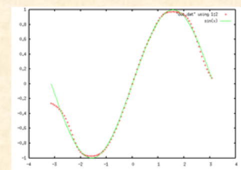
- Artificial neural networks are rarely “programmed”
- Instead, they learn from experience
- Most of the networks that we will encounter learn from their mistakes
- The root of this learning is a basic law of association, the law of contiguity



What Can Networks Do?

- What kinds of tasks can modern networks perform?
- Networks are often used to classify patterns
- Networks are also capable of approximating functions
- We will see examples of both of these abilities throughout the course
- Let us consider how they might <u>learn</u> to do this!





First Building Block: Association

- James’ <u>law of contiguity</u> for associating two ideas together, where each idea is represented as a pattern of neural activity:
- “When two elementary brain-processes have been active together or in immediate succession, one of them, on reoccurring, tends to propagate its excitement into the other” (James, 1890)

William James

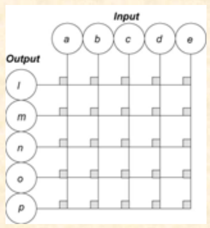
Hebb And Association

- “When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes place in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased” (Hebb, 1949)
- **Principle of contiguity!**
- Let us use this principle to create a <u>basic learning rule</u> for a simple connectionist network!

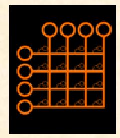
Donald Hebb

Distributed Associative Memory

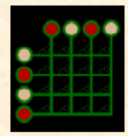
- Simple connectionist architecture:
  - Distributed associative memory
  - Standard pattern associator
- One set of input units
- One set of output units
- Modifiable connections
- Task
  - Learn associations between input and output patterns
  - Later, present one pattern, and have the system recall the other associated pattern



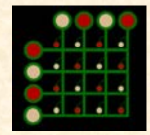
Hebb Learning



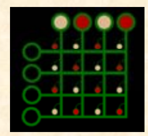
Tabula Rasa



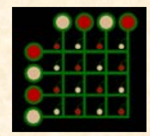
Activate Two Patterns



Associate Patterns



Activate Input



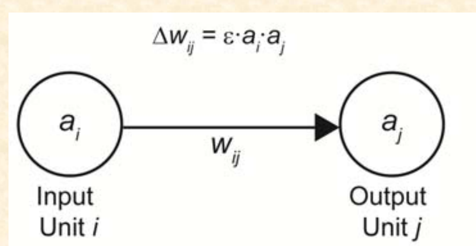
Retrieve Output

Desired Weight Changes

- Qualitatively, we can look at the activity of an input unit and an output unit to see how the weight between them should change to learn the association
- The relationship between them is multiplicative

Activity Of Input Unit	Activity Of Output Unit	Direction Of Desired Weight Change
Positive	Positive	Positive
Negative	Negative	Positive
Negative	Positive	Negative
Positive	Negative	Negative

Simplest Hebb



Algebra of Learning

- Mathematically Hebb learning can be described as using the outer product of two vectors to create a matrix of weight changes
- These are added to existing connection weights
- This is associative learning!

$$\eta \cdot c \cdot d^T = \Delta_{t+1}$$

$$0.1 \bullet \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \bullet \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 & 0.3 & 0.4 \\ 0.2 & 0.4 & 0.6 & 0.8 \\ 0.3 & 0.6 & 0.9 & 1.2 \\ 0.4 & 0.8 & 1.2 & 1.6 \end{bmatrix}$$

Trial (t+1)	Operation	Equation Describing Weight Values
0	Start with the ***0*** matrix	$W_0 = 0$
1	Associate ***a*** with ***b***	$W_1 = W_0 + \Delta_1$ $= 0 + (b \bullet a^T)$ $= (b \bullet a^T)$
2	Associate ***c*** with ***d***	$W_2 = W_1 + \Delta_2$ $= (b \bullet a^T) + \Delta_2$ $= (b \bullet a^T) + (d \bullet c^T)$

Algebra Of Recall

- Recall from the memory involves sending an input signal through existing weights to produce output unit activity
- Algebraically this involves multiplying the matrix of existing connection weights through the vector of input unit activities
- ***r = Wc***

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 10 \\ 20 \\ 30 \\ 40 \end{bmatrix}$$

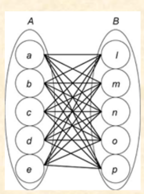
(A)

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 10 \\ 20 \\ 30 \\ 40 \end{bmatrix}$$

(B)

Limitations Of Hebb Rule

- Hebb rule has many problems, which you will observe first hand from your exercises:
  - Only learns orthogonal patterns
  - Produces error when overtraining
  - Unable to deal with linear dependence
- We would like to develop a new kind of Hebb learning rule
  - This rule would permit the network to correctly recall correlated patterns
  - This rule would also allow the network to improve its performance with repeated presentations of patterns

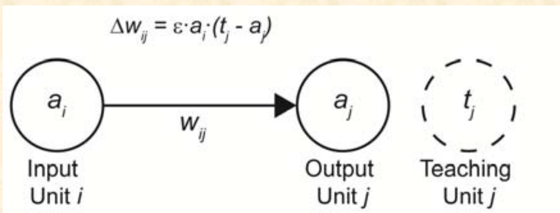


Error And Weight Change

- What would happen if we computed output unit error before we used the Hebb rule to learn associations?
- How would weights change if error was taken into account?

Activity Of Input Unit	T - O	Implication	Operation To Reduce Error	Direction Of Desired Weight Change
Positive	Positive	T > O	↑O	Positive
Positive	Negative	T < O	↓O	Negative
Positive	Zero	T = O	None	Zero
Negative	Positive	T > O	↑O	Negative
Negative	Negative	T < O	↓O	Positive
Negative	Zero	T = O	None	Zero

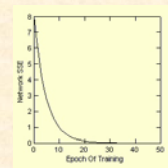
Simplest Delta Rule



The Delta Rule

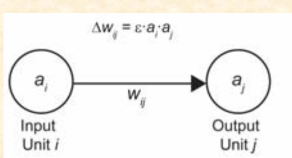
- The delta rule can be viewed as a Hebb-style association between an input vector and an (output) error vector
- Repeated applications will reduce error
- The amount of learning depends on the amount of error
- The delta rule can be written as:

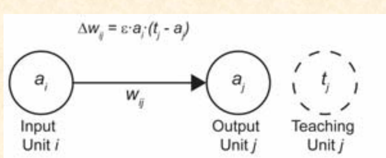
$$\Delta_{t+1} = \eta\,((t - o) \bullet c^T)$$



Comparing The Rules

- The delta rule is a minor variation on the Hebb rule
- The delta rule is based on Hebb learning, and is the basis for other learning rules that we will encounter





Two More Building Blocks

- Association is the first of three key building blocks for connectionist networks
- We still need to add <u>nonlinearities</u> into the processing units, letting them make decisions
- We still need to add some methods by which <u>layers</u> of these nonlinearities can be coordinated together
- These will be our topics in later lectures