PSYCO 452

Week 3: Sequences Of Decisions

- •Building Associations
- •Making Decisions
- •Sequences Of Decisions
- •Multilayer Perceptron
- •Credit Assignment Problem
- •Backpropagation Of Error
  - –Integration Device Network
  - –Networks Of Value Units

Course Trajectory

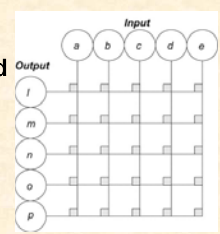
When	What
Weeks 1-3	Basics of three architectures (DAM, perceptron, MLP)
Weeks 4-6	Cognitive science of DAMs and perceptrons
Week 7	Connectionism and Cognitive Psychology
Weeks 8-10	Interpreting MLPs
Weeks 11-13	Case studies (interpretations, applications, architectures)

Chapter 10 Discussion

- Questions?
- Important Terms
  - Nonlinearity
  - All-or-none law
  - Activation function
  - Perceptron
  - Logistic equation
  - Integration device
  - Gaussian equation
  - Value unit
  - Gradient descent rule
  - Linear nonseparability

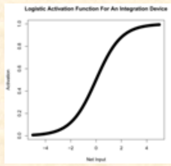
Association: The DAM

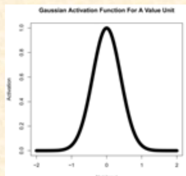
- Modern views of neural association have produced the distributed associative memory
- This memory model has many interesting properties
- But we know that it is severely limited in its processing power



Nonlinearity: The Perceptron

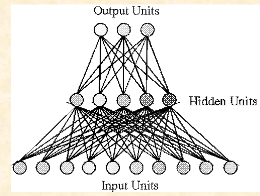
- Our first step in dealing with this issue was developing a DAM with nonlinear activation functions in its output units – i.e. the perceptron
- Nonlinearity permits output units to be interpreted as making decisions





Adding Hidden Units

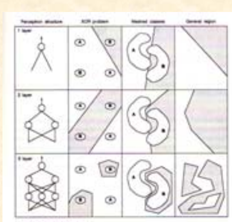
- In this lecture we consider another step to make networks even more powerful than perceptrons
- With nonlinear activation functions, intermediate layers of processors add power
- To make powerful, modern networks, we can train multilayer perceptrons that include at least one layer of hidden units



Classification: How Powerful?

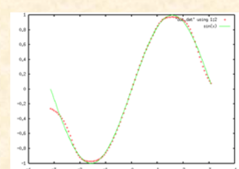
- Lippmann (1987) proved that a network with only two layers of hidden units can be an ***arbitrary pattern classifier***
- “No more than three layers [of connections] are required in perceptron-like feedforward nets” (p. 16).

Richard P. Lippmann

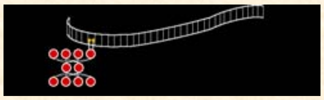


Function Fitting: How Powerful?

It has been proven that networks can be ***universal function approximators***. “If we have the right connections from the input units to a large enough set of hidden units, we can always find a representation that will form any mapping from input to output” (Rumelhart, Hinton, & Williams, 1986).



Computation: How Powerful?



- McCulloch and Pitts proved that one could build a UTM tape head from a network.
- "To psychology, however defined, specification of the net would contribute all that could be achieved in that field" (1943).

Warren McCulloch

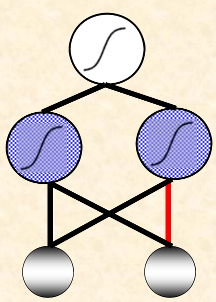
Walter Pitts

Teaching Perceptrons

- With nonlinear activation functions, we considered variations of the delta rule for training perceptrons
  - Delta rule for threshold function
    - $W_{ij(new)} = W_{ij(old)} + \eta(t_j - o_j)a_i$
  - Gradient descent rule for logistic function
    - $W_{ij(new)} = W_{ij(old)} + \eta(t_j - o_j)f'(net)a_i$
  - Gradient descent rule for Gaussian function
    - $W_{ij(new)} = W_{ij(old)} + \eta(t_j - o_j)G'(net)a_i + \eta(t_j * net)G'(net)a_i$

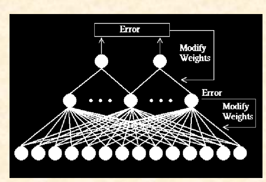
Teaching Multilayer Perceptrons?

- Hidden units have no desired values associated with them
- So how do you compute their error?
- How do you give each hidden unit proper "credit" for its contribution to overall network error?
- Use calculus to determine how much the overall error rate is affected by manipulating one of the weights that goes into a hidden unit



Backpropagation of Error

- "Generalized delta rule"
- Originally discovered by Paul Werbos in 1975
- Re-discovered (and popularized) by Rumelhart, Hinton, & Williams (1986)



David Rumelhart

Geoff Hinton

Ronald J. Williams

The Logistic Equation

- The logistic activation is nonlinear, which is crucial
- More importantly it is continuous
- It has a derivative
- It permits the use of calculus to derive a learning rule

$$a_{pi} = \frac{1}{1+e^{(-net_{pi})}}$$

Gradient Descent

- $E = \Sigma E_p = \Sigma\Sigma (t_{pi} - o_{pi})^2$
  - "Total error = sum of squared differences between desired and observed activity, summing over all patterns and all output units"
- We want to go downhill in E

  $\Delta w_{ij} = -k\ \delta E_p / \delta w_{ij}$
- Calculus says: $\Delta w_{ij} = \varepsilon\ \delta_{pi}\ a_{pj}$
  - Weight change = learning rate * error * input activity

What Is δpi?

- The derivative of the logistic equation with respect to net input is

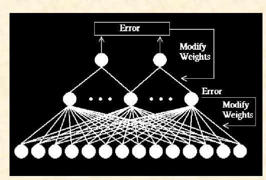
  $a_{pi}\ (1 - a_{pi})$
- So, for an output unit:

  $\delta_{pi} = (t_{pi} - a_{pi})\ a_{pi}\ (1 - a_{pi})$
- And for a hidden unit:

  $\delta_{pi} = a_{pi}\ (1 - a_{pi})\ \Sigma\ \delta_{pk}\ w_{ki}$

From Math To Algorithm

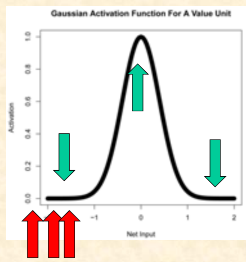
– Present a pattern to a network
  - Calculate output unit error
  - Use gradient descent rule to modify output unit weights
  - Send error backwards as "net input" to hidden units to determine their error
  - Use gradient descent rule to modify hidden unit weights



– Repeat for next pattern, until network converges on a solution

Value Units And GDR

- Inserting the Gaussian into the standard backprop rule does not lead to efficient learning
- Instead, the network usually falls into a local minimum in which it has learned to turn off to all problems
- Dawson and Schopflocher (1992) had to derive a new version of GDR to solve this problem



An Elaborated Error Term

- Standard error term in GDR

$$E = \sum E_p = \sum\sum (t_{pi} - o_{pi})^2$$

- Dawson & Schoplocher error term

$$E = \sum E_p = \sum\sum (t_{pi} - o_{pi})^2 + \sum\sum t_{pi}(net_{pi} - \mu_i)^2$$

- This second term keeps some of the patterns in the middle of the activation function!

Michael R.W. Dawson

Don Schopflocher

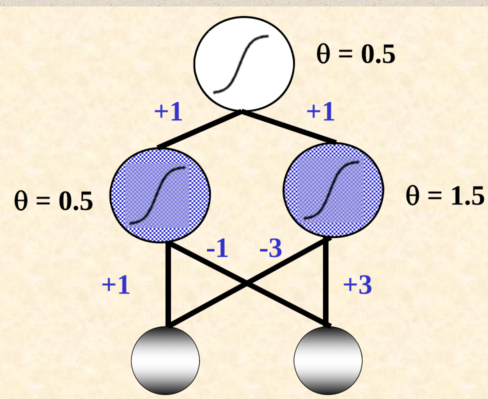
A New Learning Rule

- Using the Gaussian, and the Rumelhart Hinton & Williams chain rule procedure, one can derive a learning rule for value units:

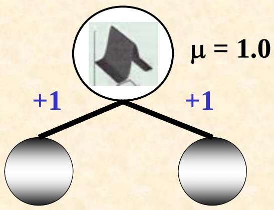
$$\Delta w_{ij} = \eta(\delta_{pi} - \varepsilon_{pi})\, a_{pj}$$

- Essentially the same as GDR, with the exception of an elaborated (two component) error term

An XOR Network

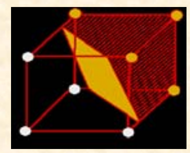


Another XOR Network



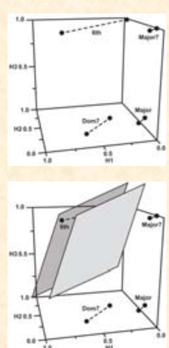
Pattern Recognition

- Networks are frequently used to classify patterns
- They carve a pattern space into decision regions
- Patterns are classified according to these decision regions
- Each unit with a logistic activation function makes a single straight cut through a pattern space



Value Unit Carving

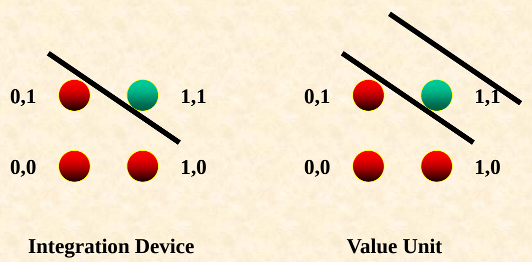
- A value unit carves a pattern space in a different fashion
- It behaves as if it has two thresholds
- It makes two parallel straight cuts through a pattern space
- The two cuts are very close together in the space



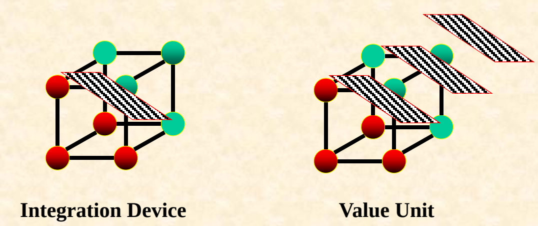
Comparing Network Types

- By looking at activation functions, and how they carve pattern spaces, you can predict when networks will have problems
- There should be network type by problem type interactions:
- Value units
  - Good for linearly nonseparable
  - Bad for linearly separable
- Integration devices
  - Good for linearly separable
  - Bad for linearly nonseparable

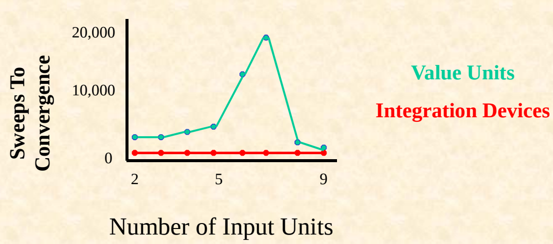
2 Majority Partitioning



3 Majority Partitioning



Speed For Majority



Speed For Parity