

1

Self-organizing Learning

- What kind of human learning might be explained through this self-organizing learning process?
- What kind of learning can't be explained in this way, and might require additional learning mechanisms?

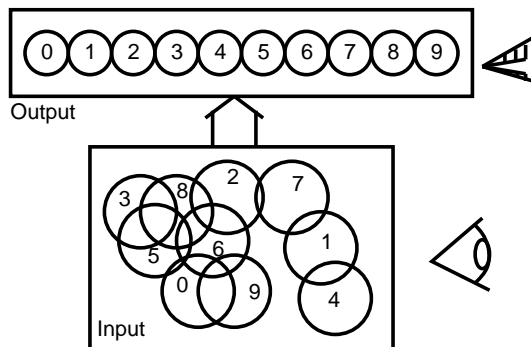
2

Learning

- Tuning detectors locally to achieve global results.
- Two main types:
 - Learning internal **model** of environment (Ch 4)
 - Learning to solve a **task** (produce output from input) (Ch 5)
 - Doing both at the same time (Ch 6)

3

Task Learning: Input-Output Mappings



Make appropriate response given stimulus.

Make appropriate interpretation, expectation, plan...

Sims.

4

Task Learning: Good News/Bad News

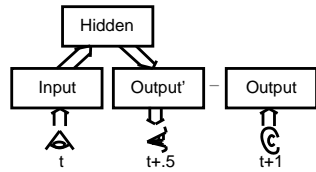
Bad News: Hebbian can't learn arbitrary input/output mappings.

Good News: Error-driven learning rule can, using discrepancy between actual and target outputs (error).

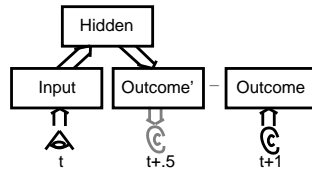
5

Where the Target Comes From

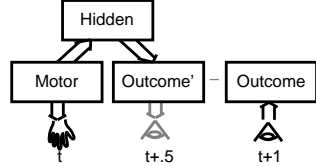
a) Explicit Teacher



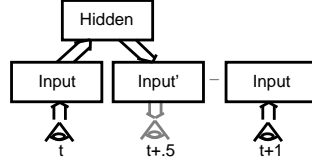
b) Implicit Expectation



c) Implicit Motor Expectation

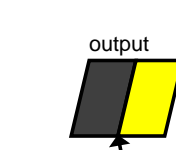
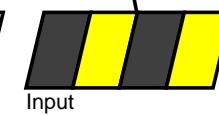
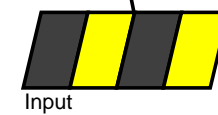
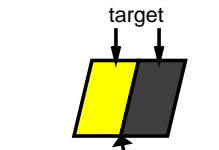


d) Implicit Reconstruction



6

What is Target? Activation Phases

a) Minus Phase
(expectation)b) Plus Phase
(outcome)

$$\Delta w_{ik} = \epsilon(a_k^+ - a_k^-)a_i \quad (1)$$

7

Credit/Blame Assignment

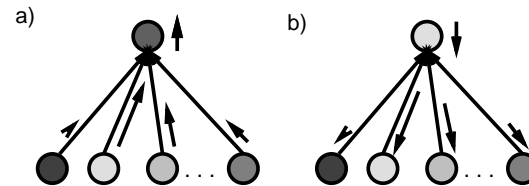
Input activation = caloric content of food.

Connection weight = how much you eat of that food. If your weight is too high, what do you do?
(eat less of the high calorie inputs!)

If your weight is too low, what do you do?
(eat more of the high calorie inputs!)

8

Credit/Blame Assignment



Weights reflect strongest *solution*
(vs. strongest *correlation* in Hebbian).

Sims.

9

Task Learning: Good News/Bad News

Bad News: Hebbian can't learn arbitrary input/output mappings.

Good News: Error-driven learning rule can, using discrepancy between actual and target outputs (error).

Lots of good news/bad news in this domain...

– Follow chronological progression to see challenges and how they were addressed?

– Or cut to final, best algorithm first and then hear about how we got there?