

1

So far...

- **Units:** conductance, current, membrane potential, equilibrium potential, firing.
- **Networks:** transformations, amplifications, attractors, basic building blocks of cognition.

How do networks ever come to do interesting things?

- **Learning**

2

Learning

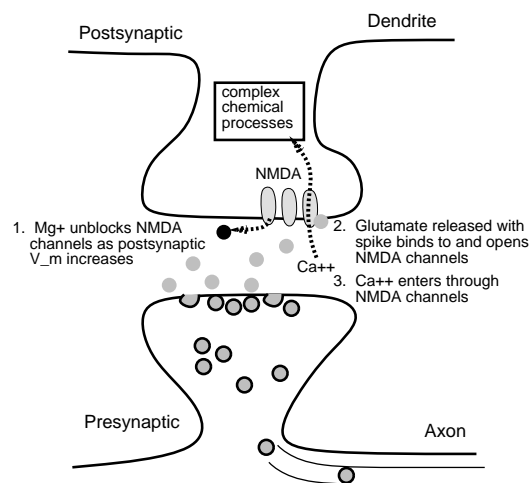
Tuning the detectors locally with the overall network in mind.

Two main types: learning internal **model** of environment, & learning to solve a **task** (produce appropriate output from input).

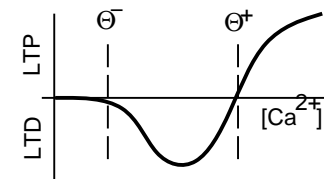
Biology suggests associative or Hebbian learning: LTP and LTD.  
("Units that fire together wire together!")

3

Biology: NMDA-mediated LTP/D



Strong activity = LTP, weak = LTD:



#### 4 Model Learning (aka self-organizing learning, Ch 4)

Based directly on Hebbian mechanisms.

Does principal components analysis (PCA) on input correlations.

#### Task Learning (Ch 5)

Hebbian often can't learn to produce specific output for given input.

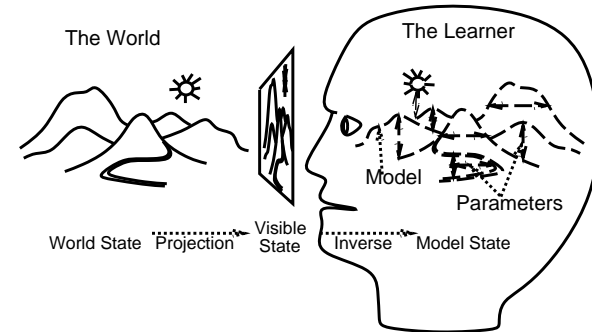
Error-driven task learning can, using discrepancy between actual and target outputs (error).

Biology does not support error passing.

Error can instead be computed as difference of two *activation* states (consistent w/biology of LTP/D).

#### 5

#### Model Learning



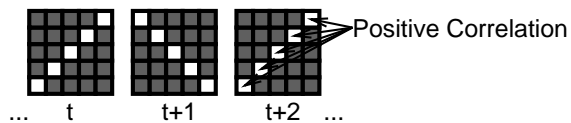
Get a lot of poor quality information.

Need biases to augment, structure this info.

One good bias is to pick up on correlations.

#### 6

#### Correlational Learning



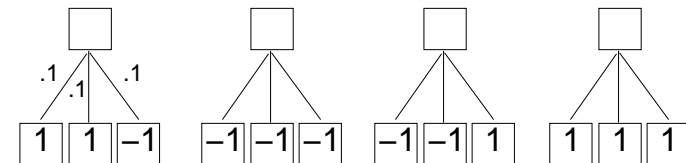
Represent the strongest (principal) correlations in the environment.

(whether for pixels in visual images, emotions and people, behaviors, etc.)

Simulation.

#### 7

#### Correlation Demo



Linear Activation:

$$y_j = \sum_i x_i w_{ij} \quad (1)$$

Simple Hebb rule:

$$\Delta w_{ij} = \epsilon x_i y_j \quad (2)$$

*Weights get stronger for the two units that are correlated, don't get stronger for uncorrelated unit!*

8

## Problems

Weights will grow infinitely large w/ simple Hebb rule.  
 – Normalize by subtracting off weight.

$$\Delta w_{ij} = \epsilon y_j (x_i - w_{ij}) \quad (3)$$

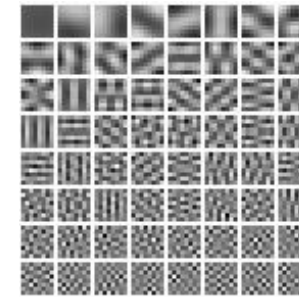
What happens w/ additional receiving units in this simple system?  
 They will all represent the same correlations.  
 – Want different units to pick up on different correlations.  
 SPCA or CPCA allows units to specialize.  
 kWTA allows competition.

9

## Sequential PCA (SPCA)

Compute correlations across all inputs for one unit (principal component), then for the next unit (next strongest component), etc.

SPCA of natural visual scenes:



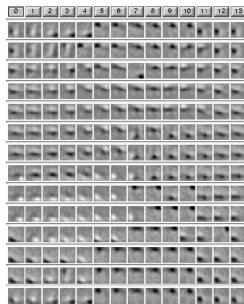
If you average over everything, all correlations disappear!  
 Problem: Assumes world is *hierarchy*.

10

## Conditional PCA (CPCA)

Compute correlations conditional on only subset of inputs (i.e., where particular feature present).

CPCA of natural visual scenes:



Captures world as *heterarchy* – no overall principal component, just a bunch of smaller ones.

11

## CPCA Equations

$$\Delta w_{ij} = \epsilon y_j (x_i - w_{ij}) \quad (4)$$

Weight moves towards  $x_i$ , conditional on  $y_j$ .

Achieves conditional probability goal:

$$\begin{aligned} w_{ij} &= P(x_i = 1 | y_j = 1) \\ &= P(x_i | y_j) \end{aligned} \quad (5)$$

Weight reflects probability that the sender  $x_i$  is active given that the receiver  $y_j$  is.

12

## Biological Implementation

$$\Delta w_{ij} = \epsilon y_j (x_i - w_{ij})$$

When both  $y_j$  and  $x_i$  large: LTP, lots of  $Ca^{++}$

When  $y_j$  large but  $x_i$  small: LTD, some  $Ca^{++}$

When  $y_j$  small: Nothing,  $Mg^+$  blocking NMDA channels.

Simulations.

13

## Corrections

- CPCA weights tend to not have much *dynamic range* or *selectivity*.
- Correct for by *renormalizing weights* and *contrast enhancement*.
- Quantitative adjustments – retain qualitative features of CPCA motivated by bio.

14

## Renormalization

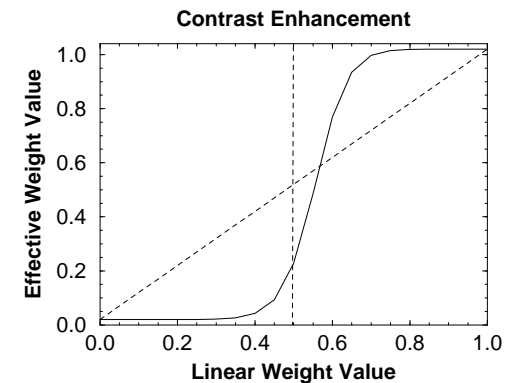
Renormalization: keep a weight value of .5 for “uncorrelated” inputs even if inputs are sparsely active:

via a parameter `savg_cor`

15

## Contrast Enhancement

Between strongest and weaker correlations, via sigmoidal function:



Slope via *gain*  $\gamma$

Offset (movable threshold)  $\theta$

