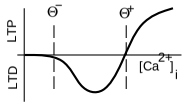
1 Combined Model & Task Learning

1. Pros and Cons: Use Both.
2. Inhibition is also an Important Bias.

2 Biology Says: Both

	Plus Phase					
	$x_i^+, y_j^+ \approx 0$			$x_i^+, y_j^+ \approx 1$		
Minus Phase	Err	Hebb	Combo	Err	Hebb	Combo
$x_i^-, y_j^- \approx 0$	0	0	**0**	+	+	**+**
$x_i^-, y_j^- \approx 1$	–	0	**–**	0	+	**+**

No Ca^{2+} → no learning
Mod Ca^{2+} → LTD
High Ca^{2+} → LTP

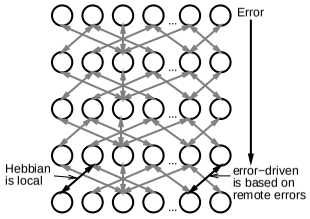


Only case disagreeing w/ Hebb is bottom left quadrant (LTD).
→ Low Ca^{2+} (just minus phase) produces LTD
→ same mechanism as Hebb LTD for y_j=1, x_i=0.

3 Other mechanisms for Error-driven Learning

- Neuromodulatory signals: Dopamine, Acetylcholine, etc.
- "Phasic" signals elicited by brain systems computing 'expected reward' and deviations from this expectation
- Resulting signals, when combined with target information (*what should have been expected*) in subsequent state, can enhance **contrast** between two succeeding attractor states
- Lots of evidence that LTP, LTD under neuromodulatory control
- Hebbian learning always occurs locally, in every synapse (model learning, statistics)
- Brain regions innervated by DA, ACh have enhanced weight changes during errors, leading to contrastive Hebbian learning (approximated by delta rule)

4 Functional: Pros and Cons



	Pro	Con
Hebbian (local)	autonomous, reliable	myopic, greedy
Error-driven (remote)	task-driven, cooperative	co-dependent, lazy

Error-driven = Left-wing, Hebbian = Right-wing (?!)

5 Combining Error-driven + Hebbian

Get benefits of both:

$$\Delta w_{ij} \approx \Delta_{hebb} + \Delta_{err} \quad (1)$$

$\Delta_{hebb} = \epsilon a_j (a_i - w_{ij})$

$\Delta_{err} = \epsilon[(a_i^+ a_j^+) - (a_i^- a_j^-)]$

$$\Delta w_{ij} = (k_{hebb})\Delta_{hebb} + (1 - k_{hebb})(\Delta_{err}) \quad (2)$$

Hebbian bias helps so that weights are constrained to smaller set of solutions (otherwise too interdependent in err-driven)

6 Inhibitory Competition as a Bias

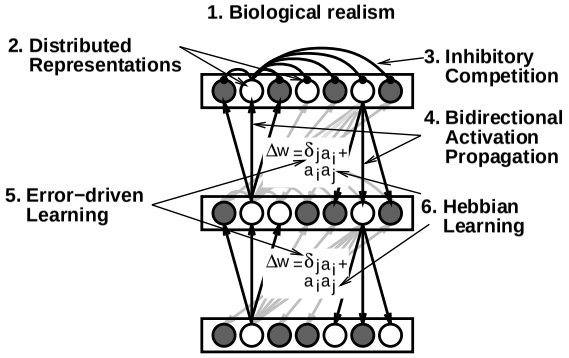
Inhibition:

- Causes sparse, distributed representations (many alternatives, only a few relevant at any time).
- Competition and specialization: survival of fittest.
- Self-organizing learning.

(Often more important than Hebbian bias)

7

The Whole Enchilada



8

Generalization

How well do we deal with things we've never seen before?

nust

each time you walk into class, each social interaction, each sentence you hear, etc.

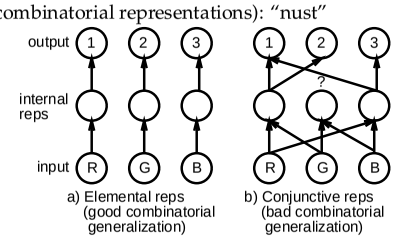
We're constantly faced with new situations, and generalize reasonably well to them.

How do we do it?

9

Generalization

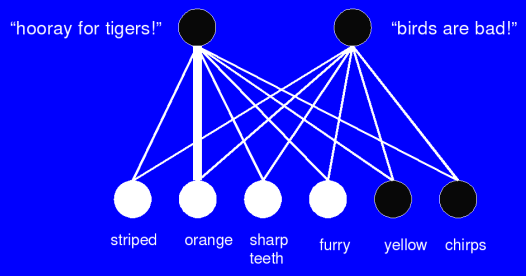
Distributed reps: novel items are novel combinations of existing features (combinatorial representations): "nust"



Hebbian & inhibition: produce elemental, combinatorial reps.

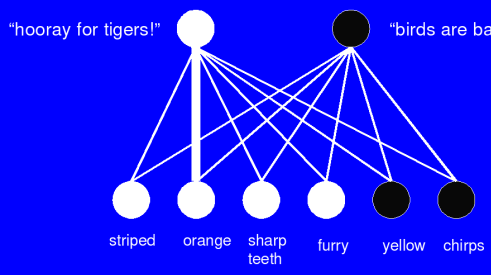
10

The Lazy Tiger Detector



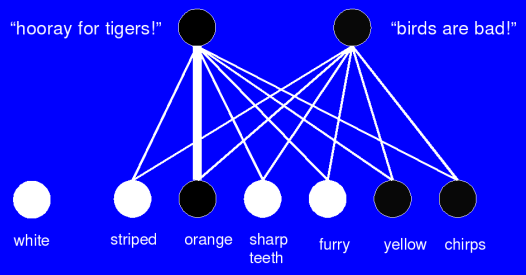
11

The Lazy Tiger Detector

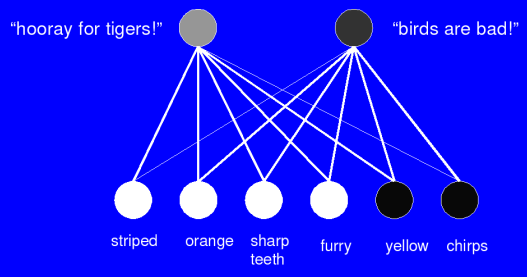


12

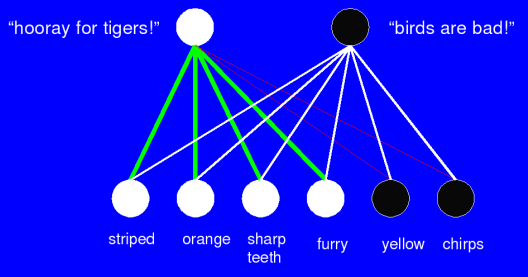
The Lazy Tiger Detector



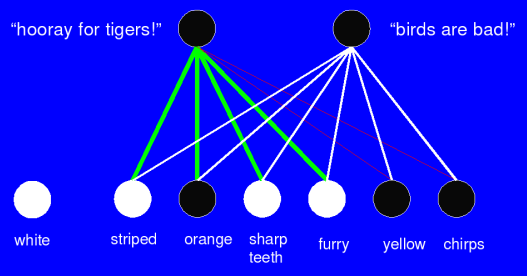
13 Hebb Shows More Appropriate Generalization



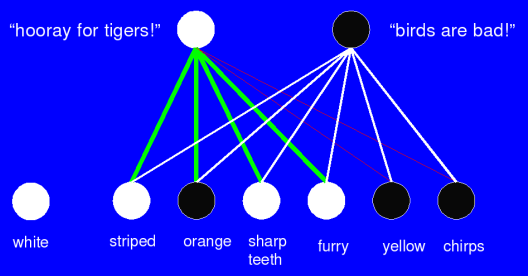
14 Hebb Shows More Appropriate Generalization



15 Hebb Shows More Appropriate Generalization



16 Hebb Shows More Appropriate Generalization



17 Sims: [model_and_task.proj]

Hebb:

- Sometimes fails to learn the training set
- Represents meaningful "things" in the world (correlations)
- Shows good generalization

Error (GeneRec):

- Always learns the training set
- Representations are "mushy"
- Can show poor generalization

Error + Hebb:

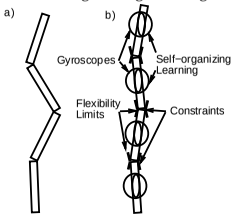
- Learns the training set (more quickly than error alone)
- Represents meaningful features
- Shows good generalization!

18 Deep Networks

Need many hidden layers to achieve many stages of transformations (dramatically re-representing the problem).

But then the error signals are very remote & weak.

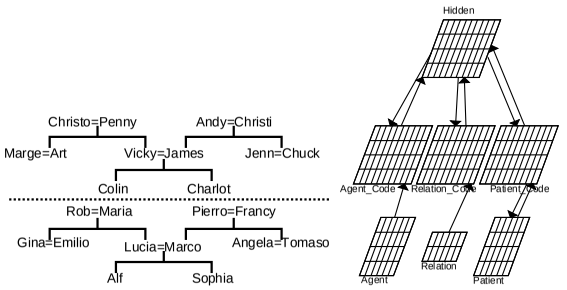
Need to add constraints and self-organizing learning:



- Hebb gives each layer *local* guidance on representations
- Inhib competition restricts flexibility (only certain states are valid)
- Combined model + task → fewer degrees of freedom to adapt

19

Example: Family Trees (Hinton, 1986)



24 people, 12 relationships (brother, mother, granddaughter, etc)

Who is Alf's grandmother?
Who is Lucia's daughter?

20

[family_trees.proj]