Error-driven learning and XCAL

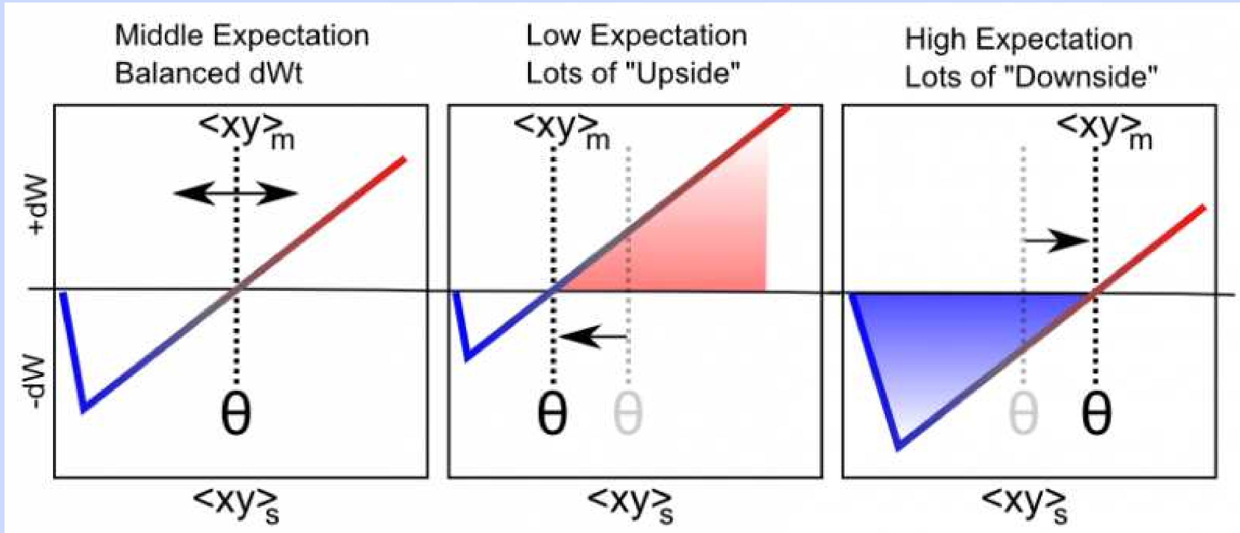
- Self-organizing Hebbian learning arises from biologically-motivated XCAL rule, adjusting weights as a function of synaptic activity $x_i y_j$ over the short-term, relative to long-term values (floating threshold).

- The XCAL rule can easily be adapted to support error-driven learning, by using the ’medium-term’ synaptic activity as the comparison.

- Medium-term activity: average of both recent outcome and earlier expectation - just averages synaptic activity over a longer horizon. Short-term activity reflects outcome preferentially.

- So, the difference between them will largely reflect differences between expectation and outcome - the *delta*.

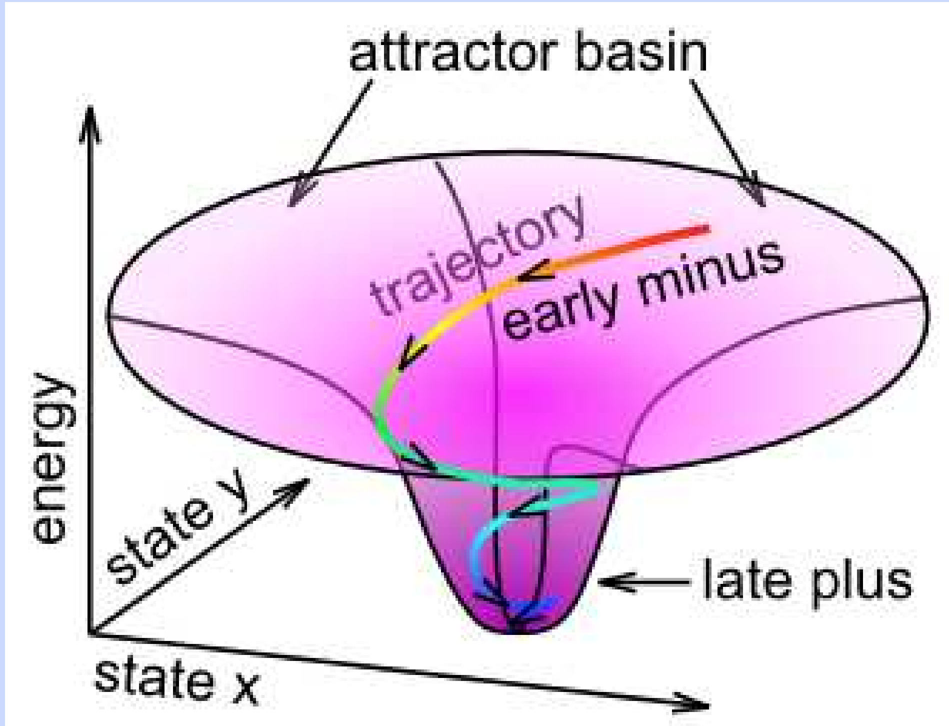
medium term activity as expectation in XCAL



$$\Delta w \approx x_s y_s - x_m y_m$$

This relationship is true for most of the XCAL function (the linear part).
Then goes back to zero for low $x_s y_s$ – no learning for no activity (Ca^{++}).

This is why it is called Contrastive Attractor Learning



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

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

Other mechanisms for Error-driven Learning

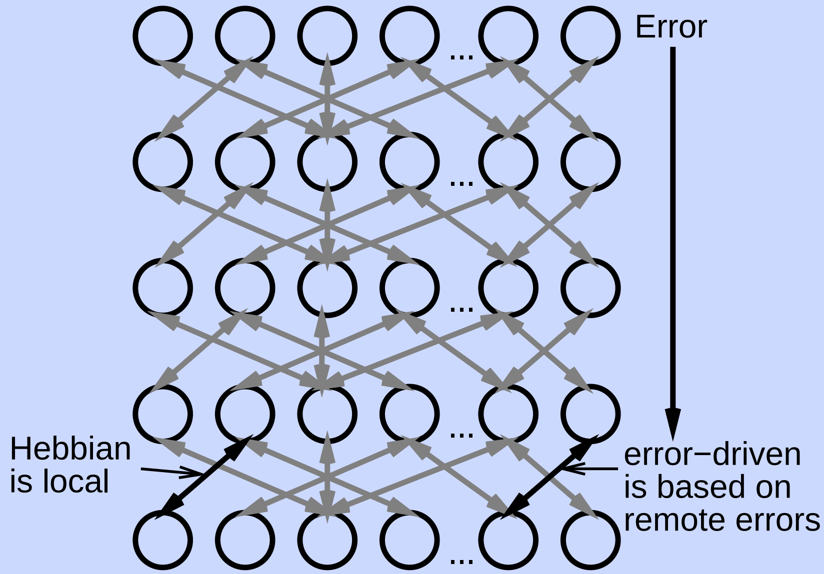
- Neuromodulatory signals: Dopamine, Acetylcholine, etc.
- “Phasic” signals elicited by brain systems computing ’expected reward’ and deviations from this expectation
- Resulting signals, when combined with target information (*what should have been expected*) in subsequent state, can enhance **contrast** 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 attractor learning (approximated by delta rule)

Combined Model & Task Learning

1. Pros and Cons: Use Both.

2. Inhibition is also an Important Bias.

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 (?!)

Combining Error-driven + Hebbian in single XCAL rule

Get benefits of both:

$$\Delta w_{ij} \approx \Delta_{hebb} + \Delta_{err}$$

Combining Error-driven + Hebbian in single XCAL rule

Get benefits of both:

$$\Delta w_{ij} \approx \Delta_{hebb} + \Delta_{err}$$

$$\Theta_p = \lambda y_l + (1 - \lambda) x_m y_m$$

$$\Delta w_{ij} = \lambda_l f_{xcal}(x_s y_s, y_l) + \lambda_m f_{xcal}(x_s y_s, x_m y_m)$$

- λ ("`thr_l_mix`" in simulator) is a parameter affecting degree to which XCAL threshold is determined by y_l or y_m.
- can differ between brain systems, or even be modulated dynamically (e.g. by neuromodulators)

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

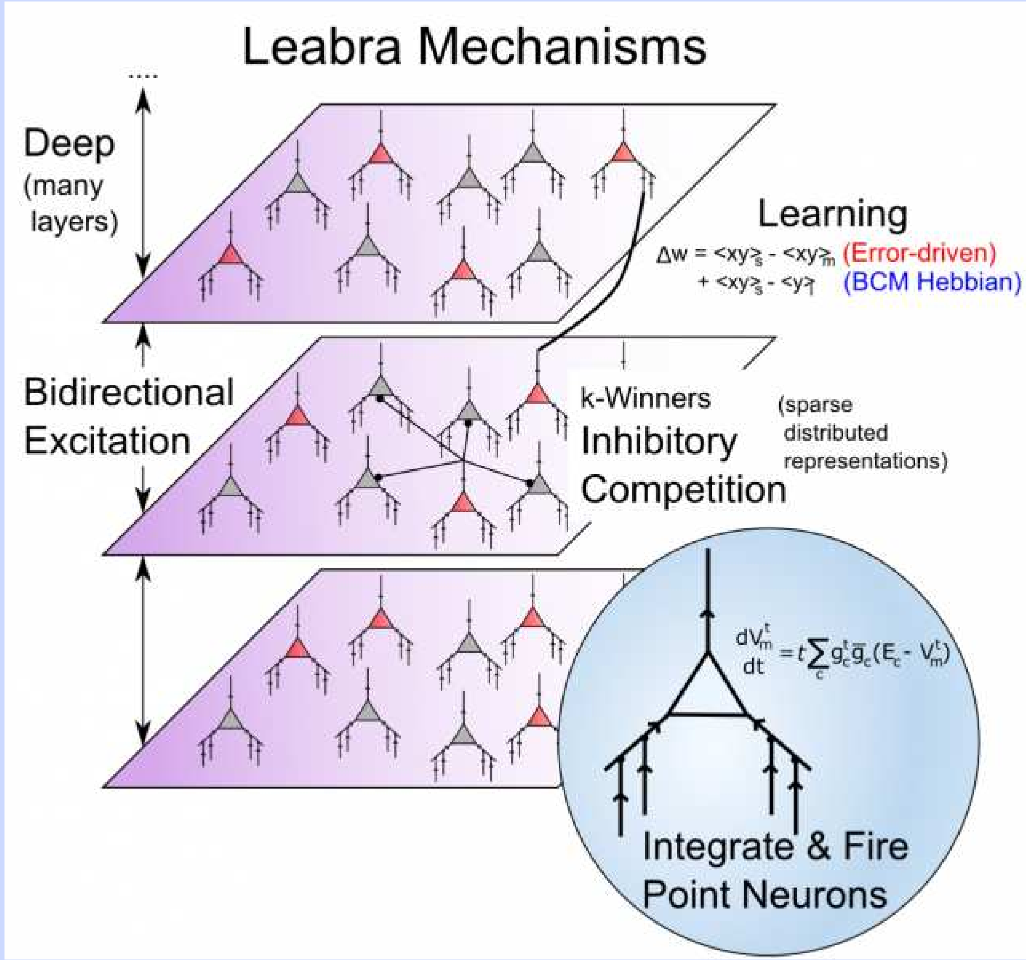
Inhibitory Competition as a Bias

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)

The Whole Enchilada



Generalization

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

Generalization

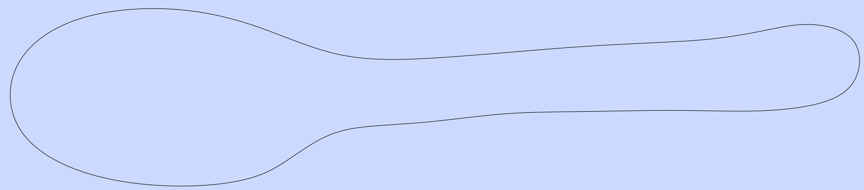
How well do we deal with things we've never seen before?

nust

Generalization

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

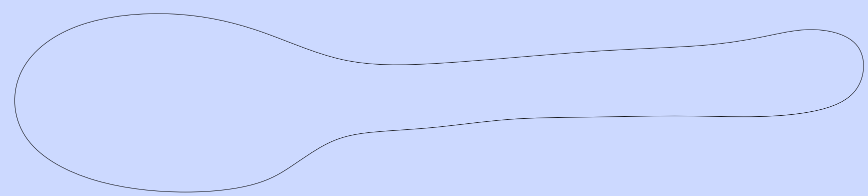
nust



Generalization

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

nust

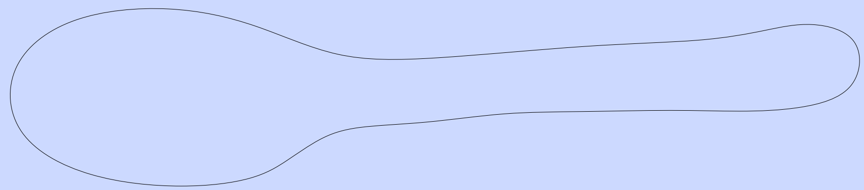


each time you walk into class, each social interaction, each sentence you hear, etc.

Generalization

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

nust



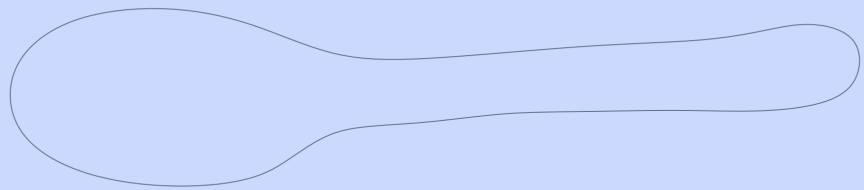
each time you walk into class, each social interaction, each sentence you hear, etc.

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

Generalization

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

nust

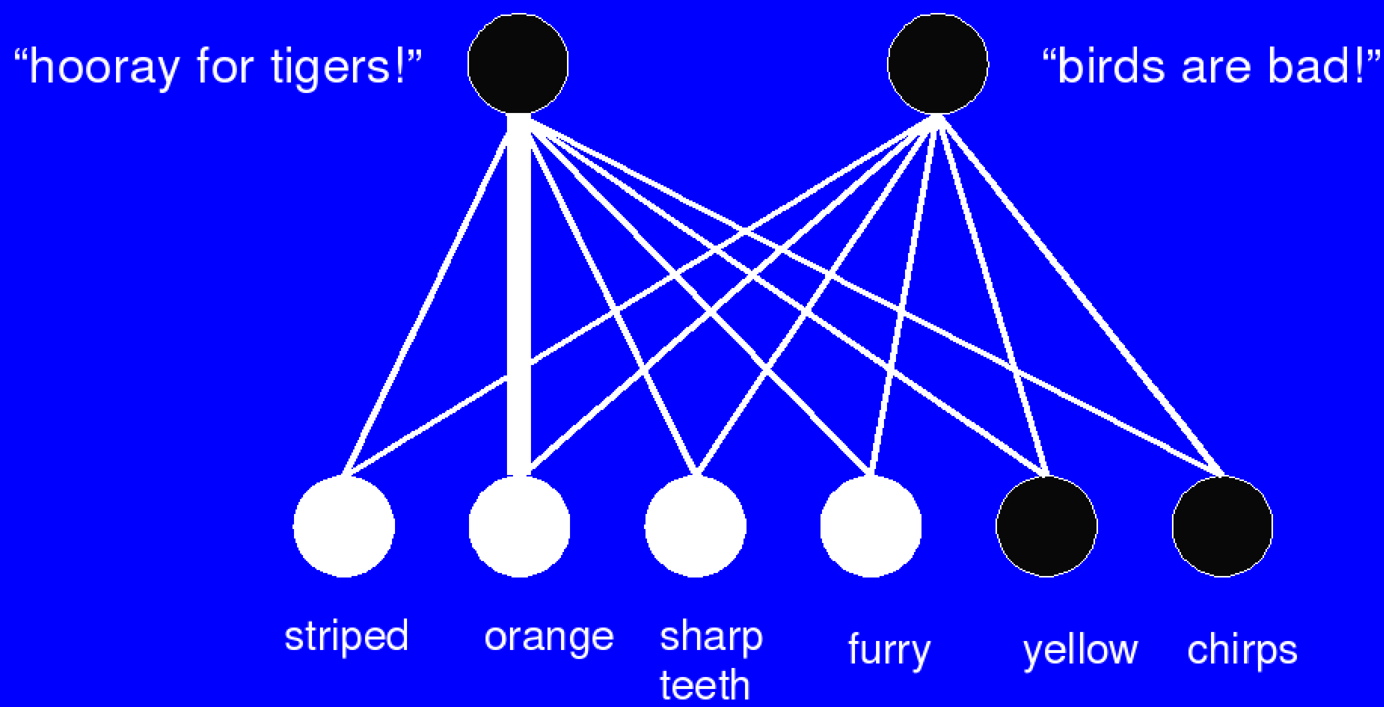


each time you walk into class, each social interaction, each sentence you hear, etc.

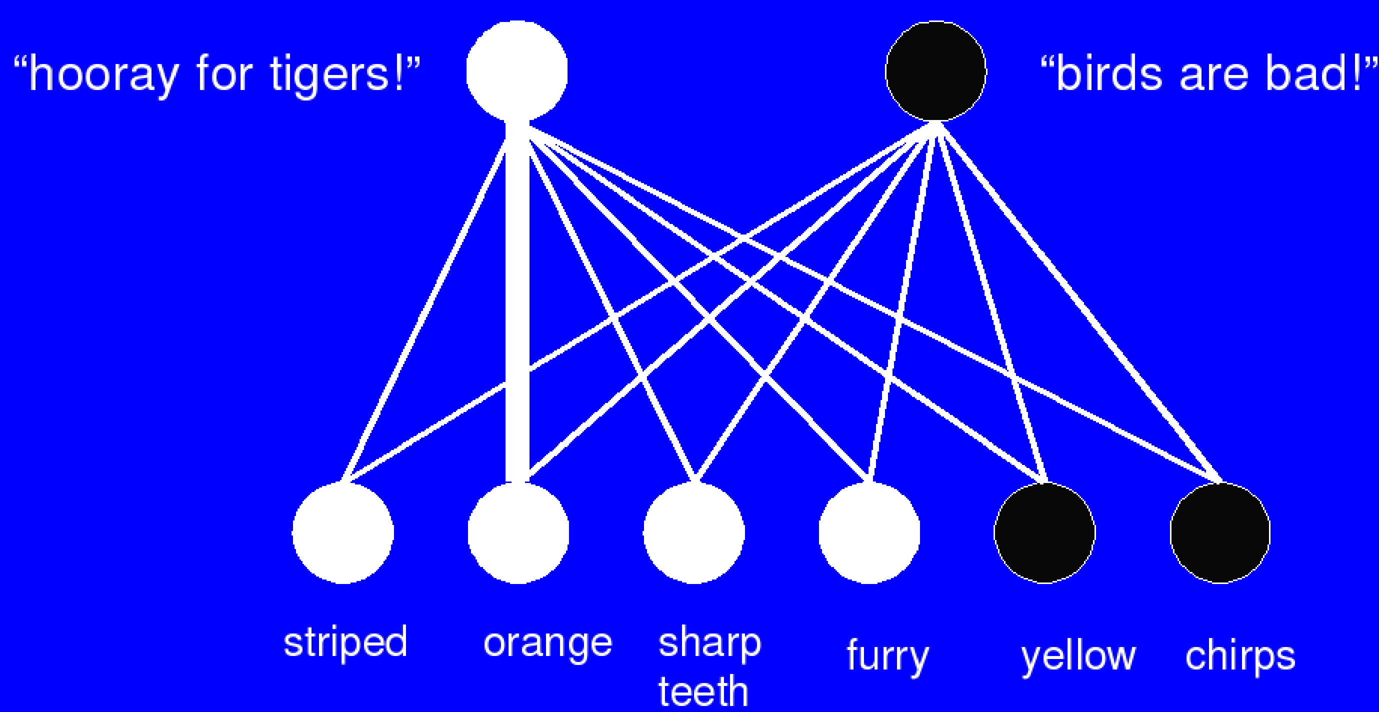
We're constantly faced with new situations, and generalize reasonably well to them.

How do we do it?

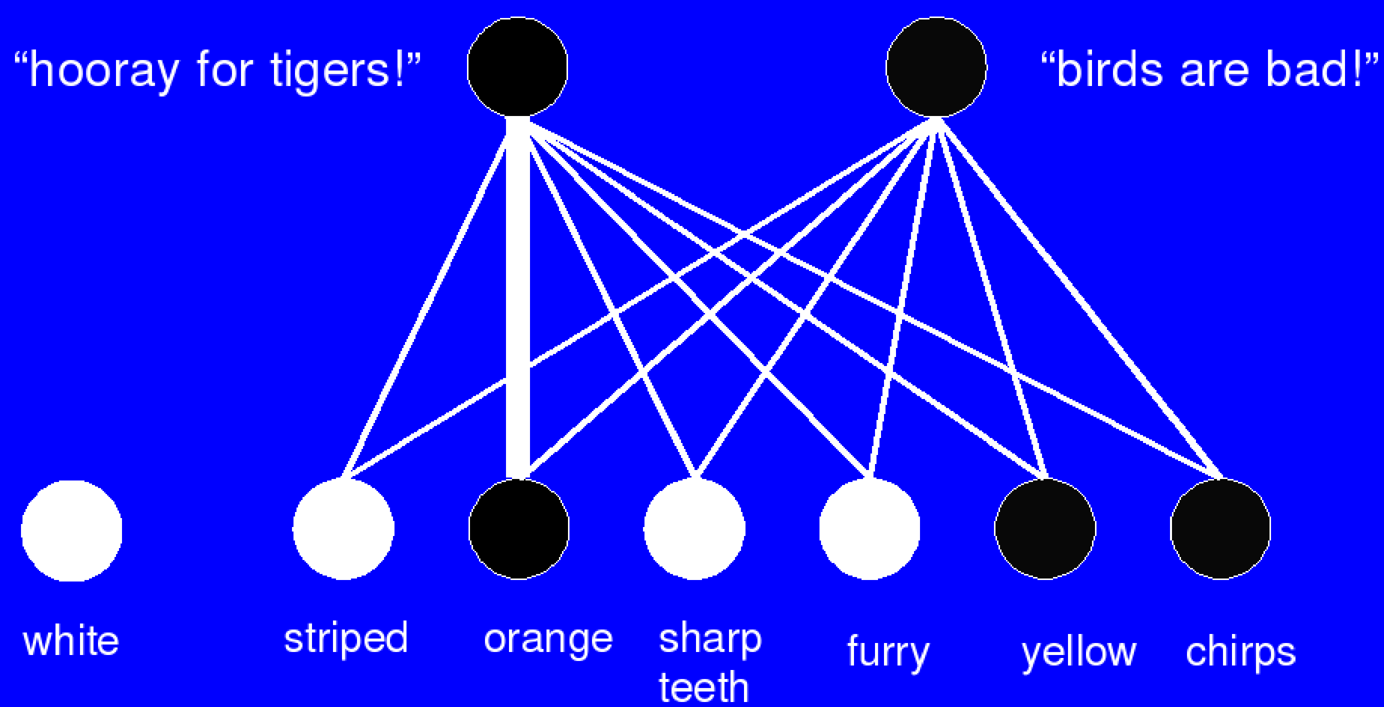
The Lazy Tiger Detector



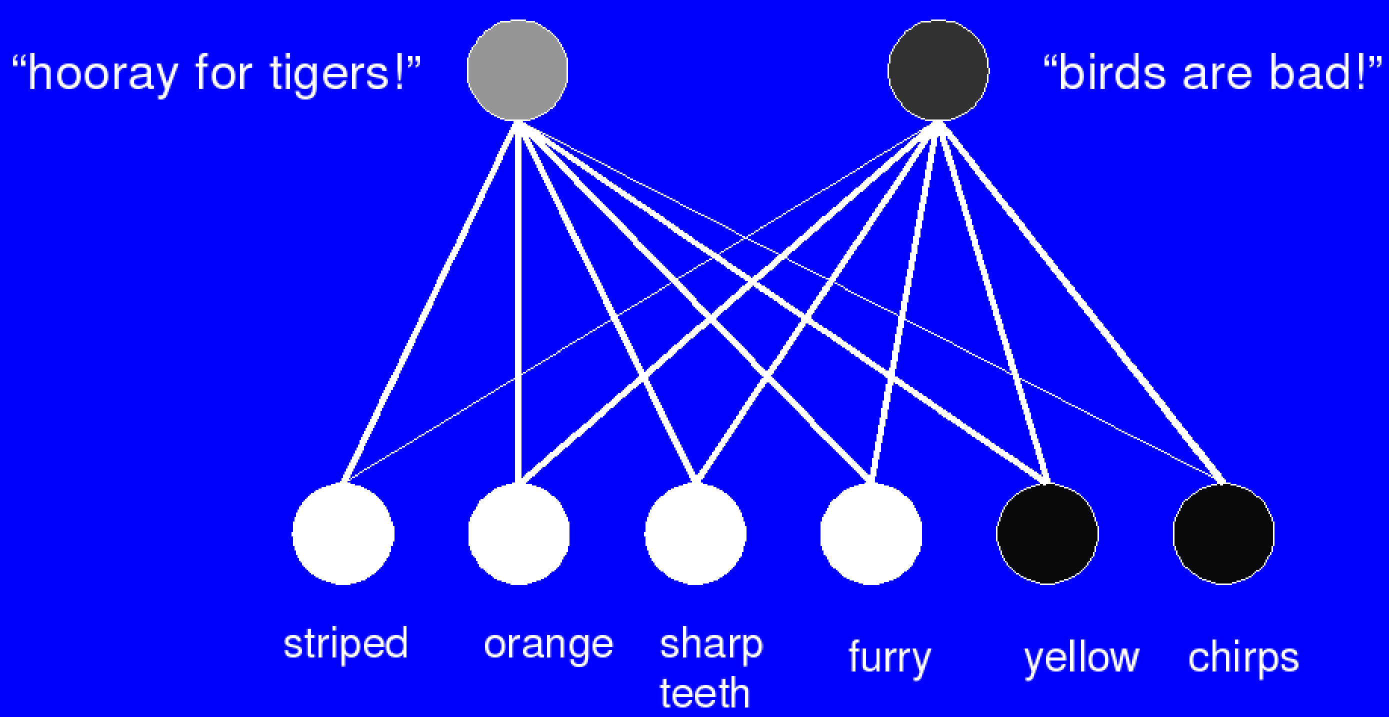
The Lazy Tiger Detector



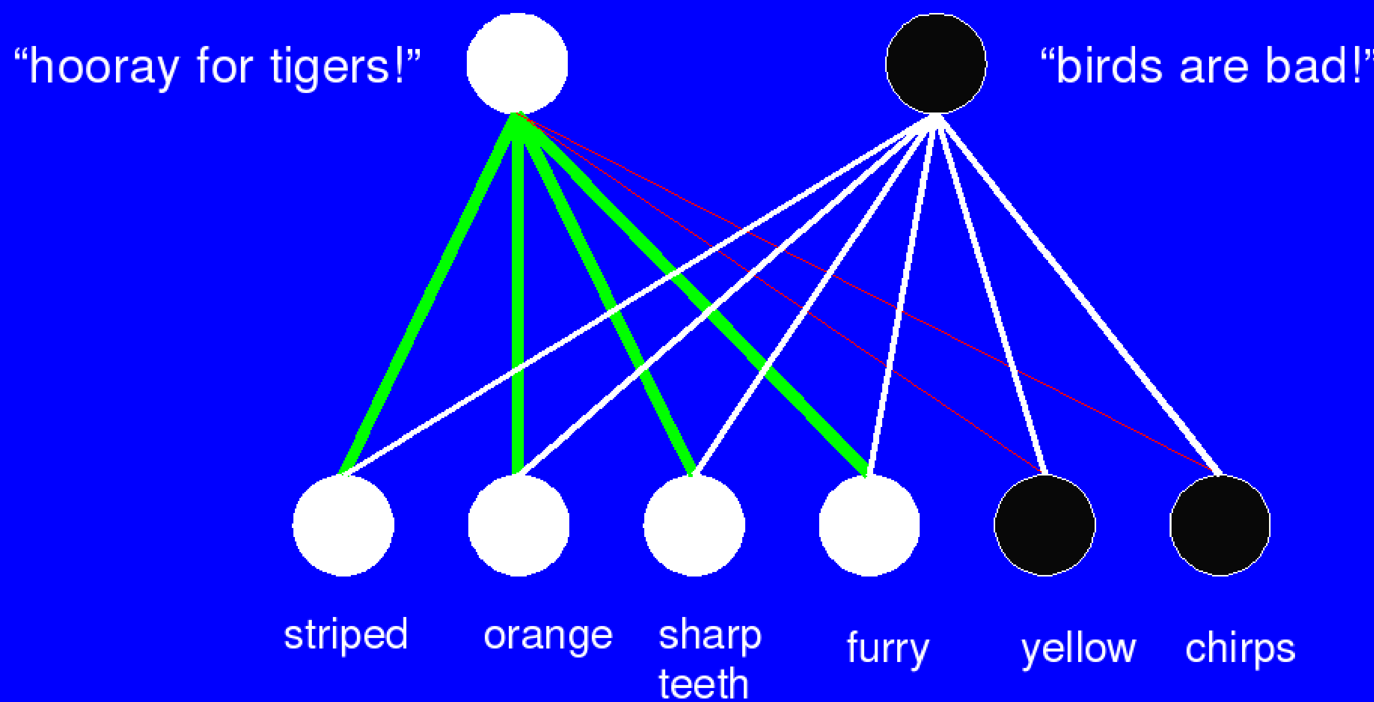
The Lazy Tiger Detector



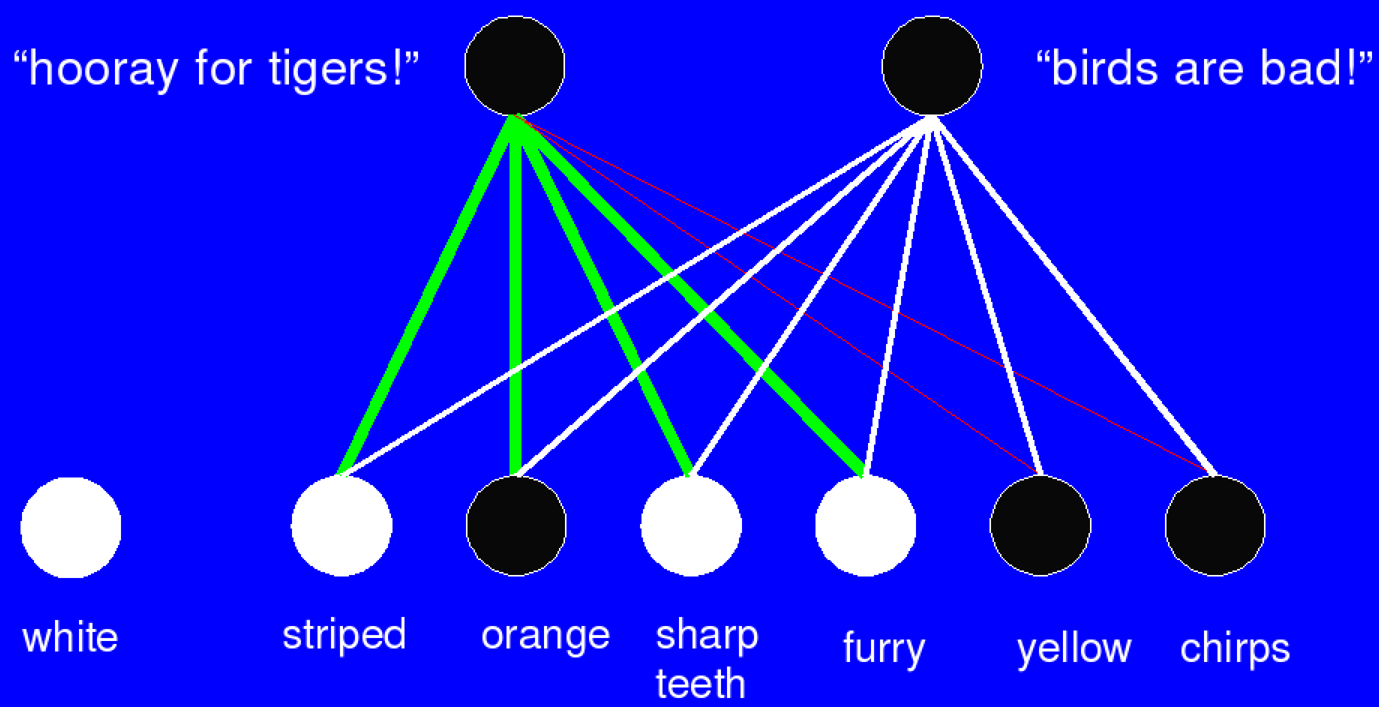
Hebb Shows More Appropriate Generalization



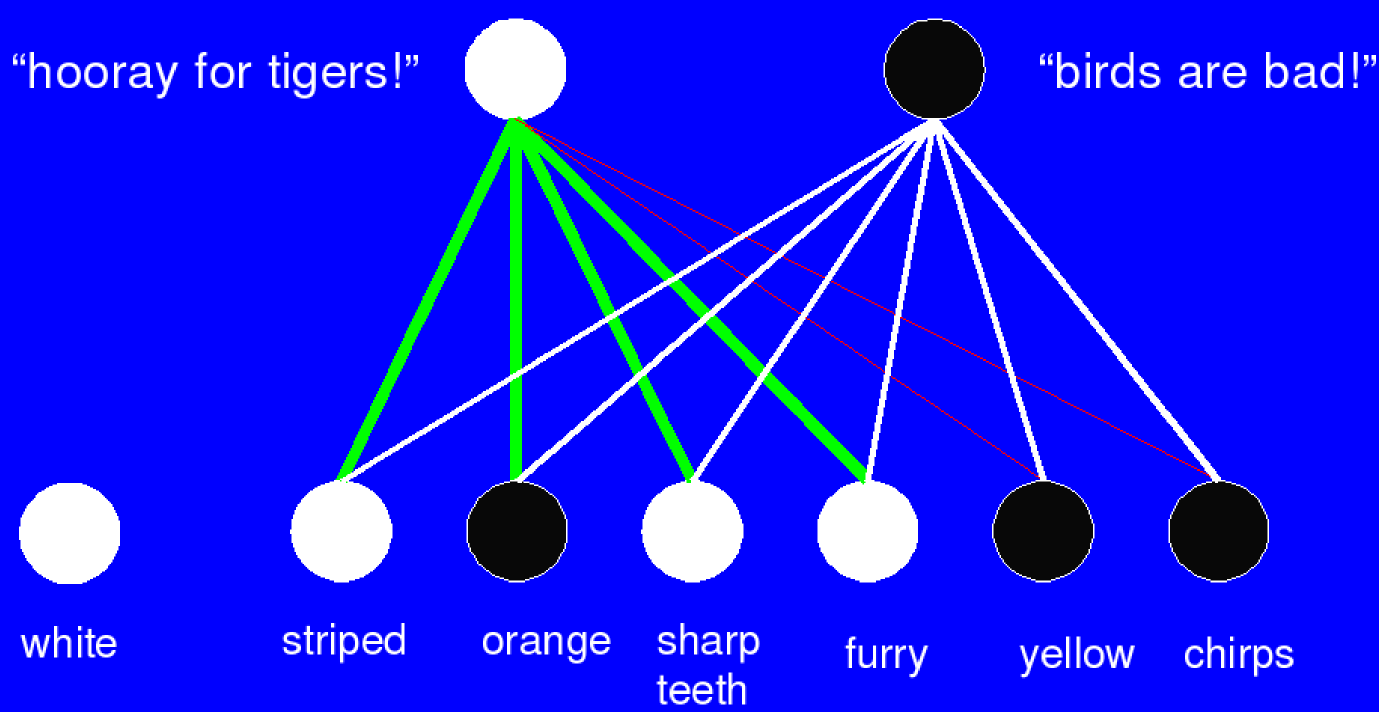
Hebb Shows More Appropriate Generalization



Hebb Shows More Appropriate Generalization



Hebb Shows More Appropriate Generalization



Hebb:

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

Error (GeneRec / XCAL with xm ym):

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

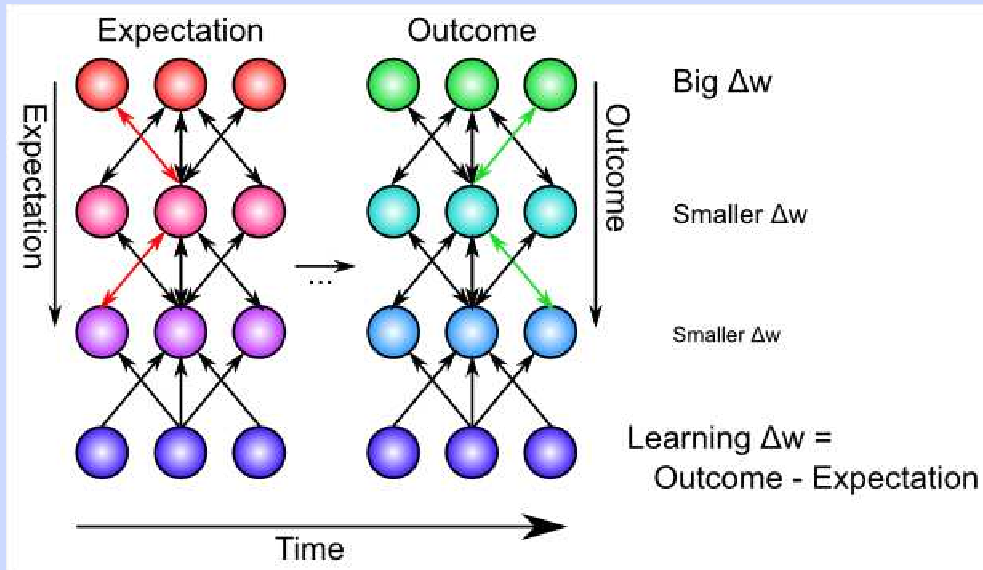
Error + Hebb:

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

Deep Networks

Need many hidden layers to achieve many stages of transformations (dramatically re-representing the problem). (cf. recent surge in interest in "deep learning" in machine learning for speech recognition, Google, etc)

But then the error signals are very remote & weak.

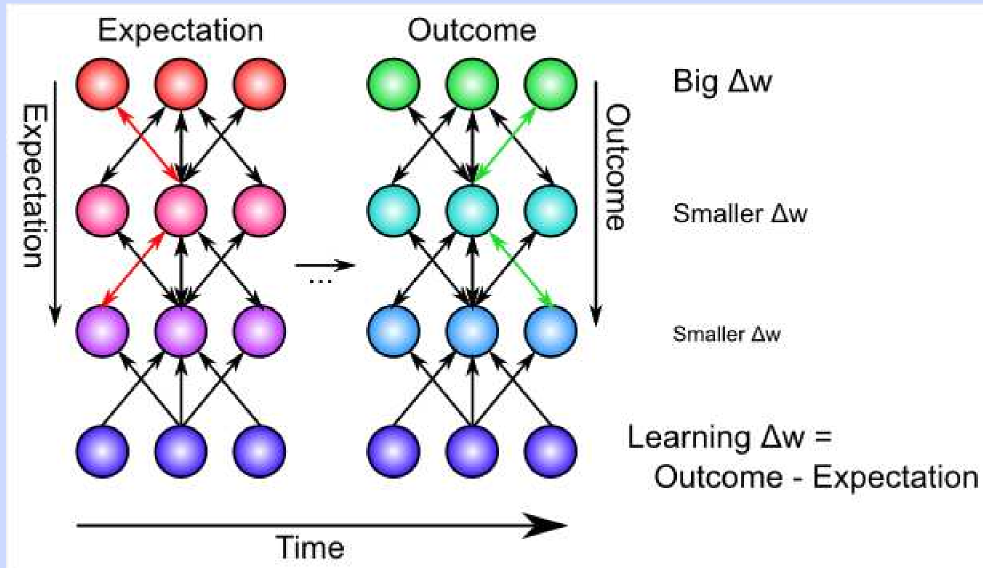


Need to add constraints and self-organizing learning:

Deep Networks

Need many hidden layers to achieve many stages of transformations (dramatically re-representing the problem). (cf. recent surge in interest in "deep learning" in machine learning for speech recognition, Google, etc)

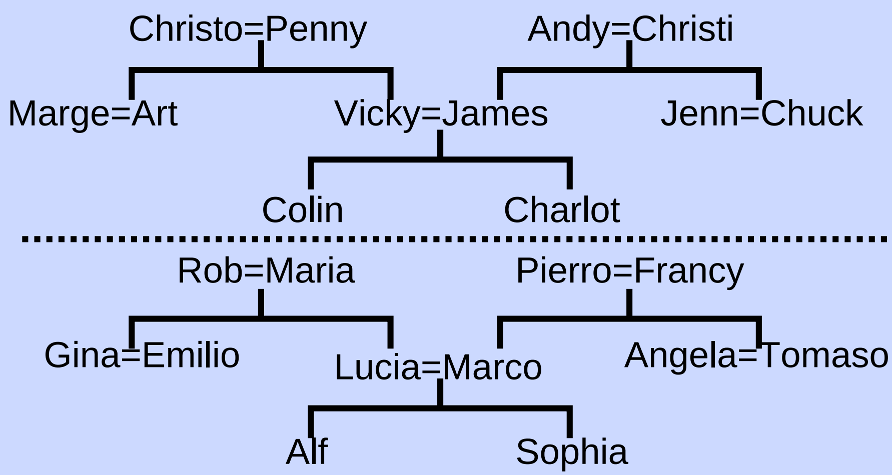
But then the error signals are very remote & weak.

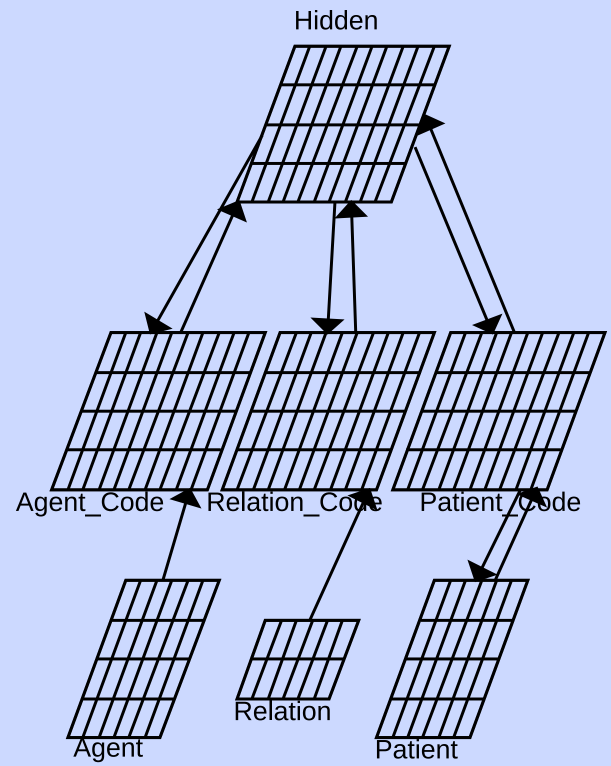


Need to add constraints and self-organizing learning:

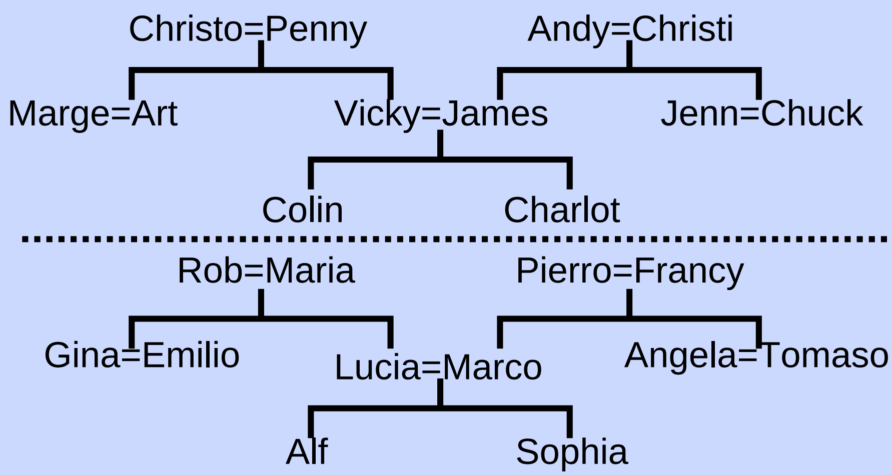
- Hebb gives each layer *local* guidance on representations
- Inhib competition restricts flexibility (only certain states are valid)
- Combined hebb + err $\rightarrow$ fewer degrees of freedom to adapt

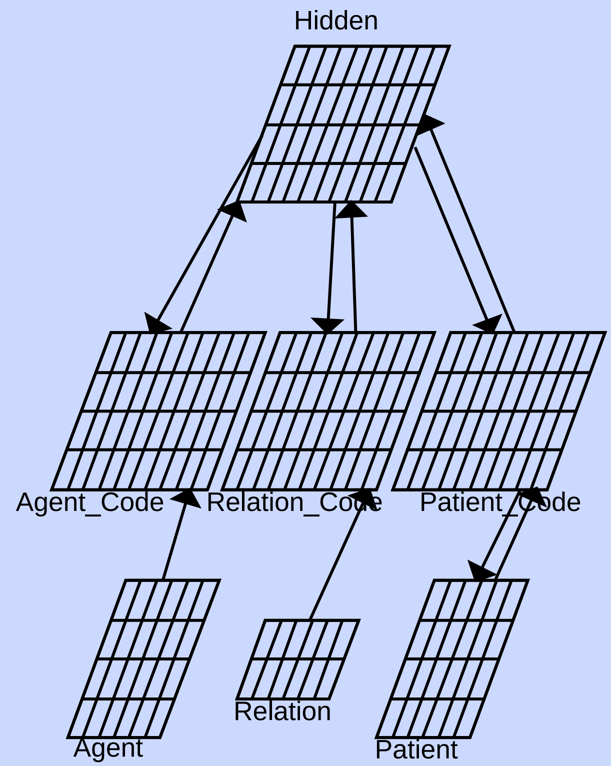
Example: Family Trees (Hinton, 1986)





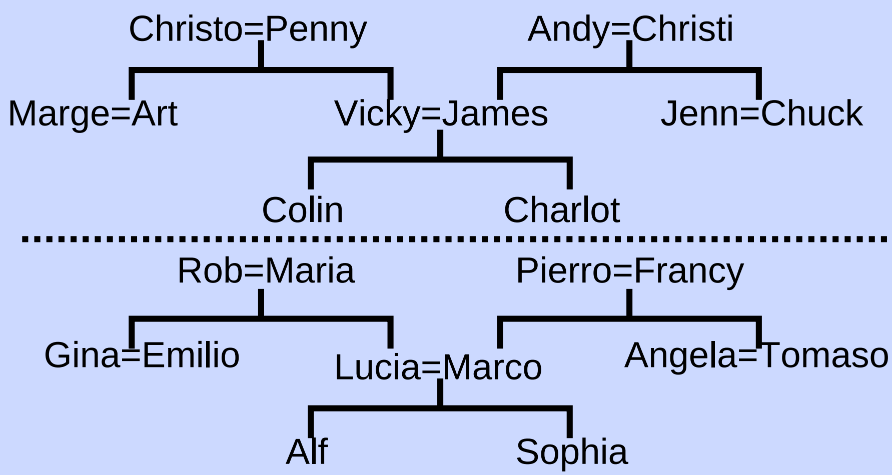
Example: Family Trees (Hinton, 1986)

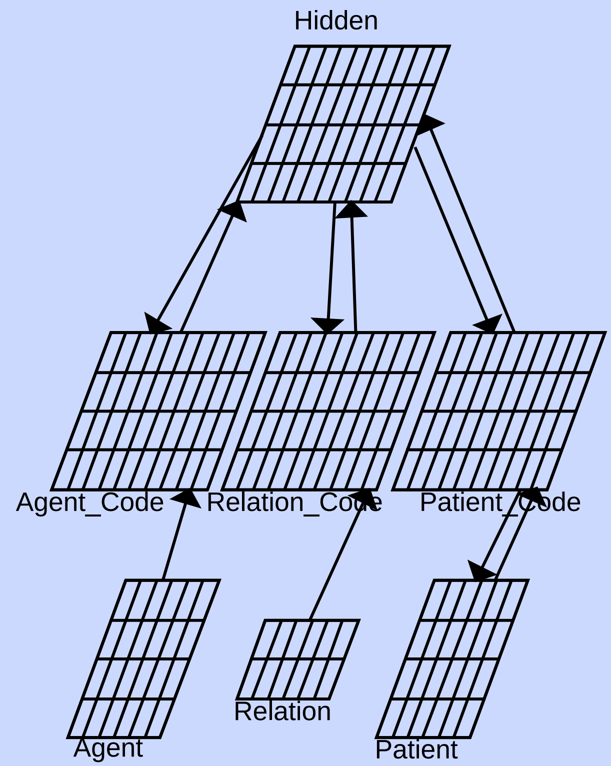




24 people, 12 relationships (brother, mother, granddaughter, etc)

Example: Family Trees (Hinton, 1986)





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

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

[family_trees.proj]