Hierarchies of RBM's

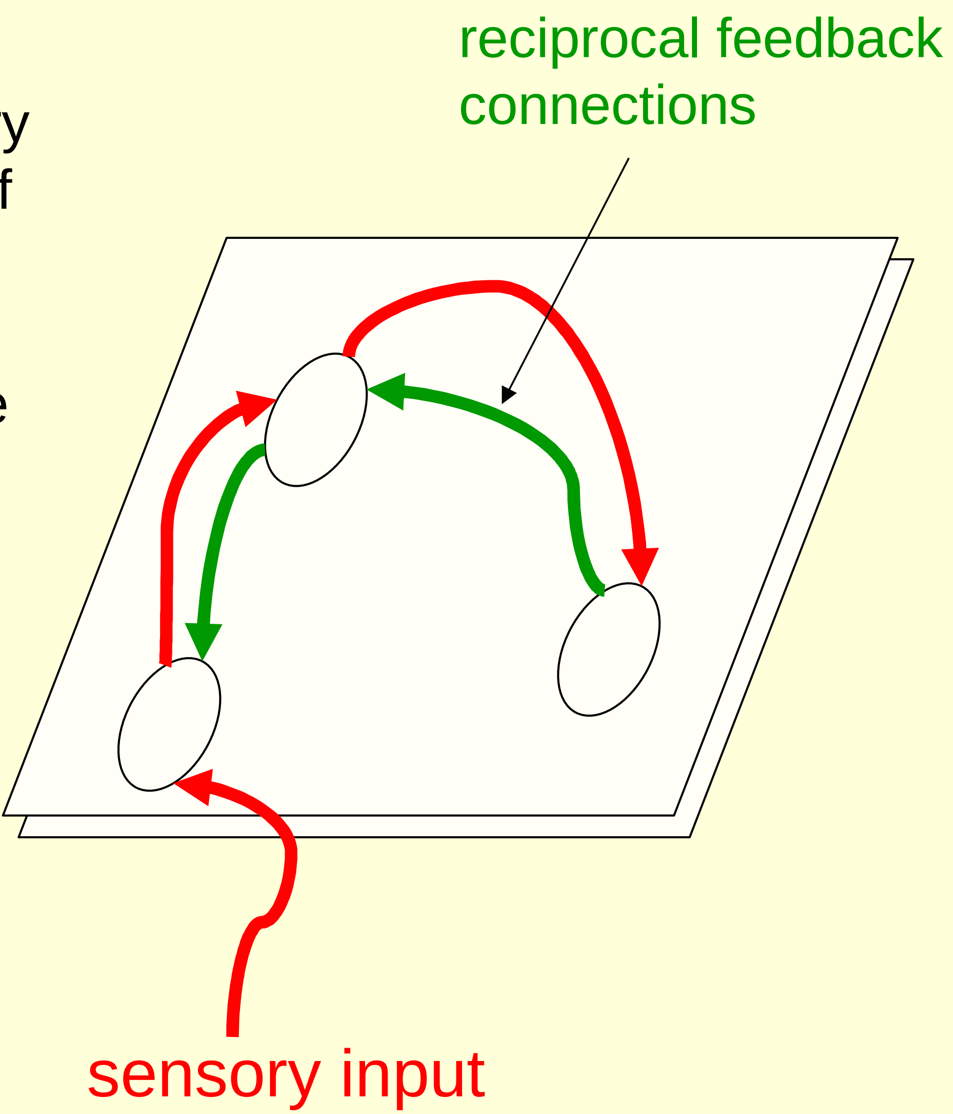
Geoffrey Hinton

CIfAR &

University of Toronto

The structure of cortex

- Cortex is a big sheet that has a very similar anatomical structure in all of the different cortical areas.
- It looks as if evolution has found a good, general-purpose architecture that gets turned into special-purpose cortical areas.
- The special purpose areas are created by three factors:
  - A general-purpose learning algorithm.
  - Connection pathways that are genetically specified.
  - Highly structured and very rich sensory input.



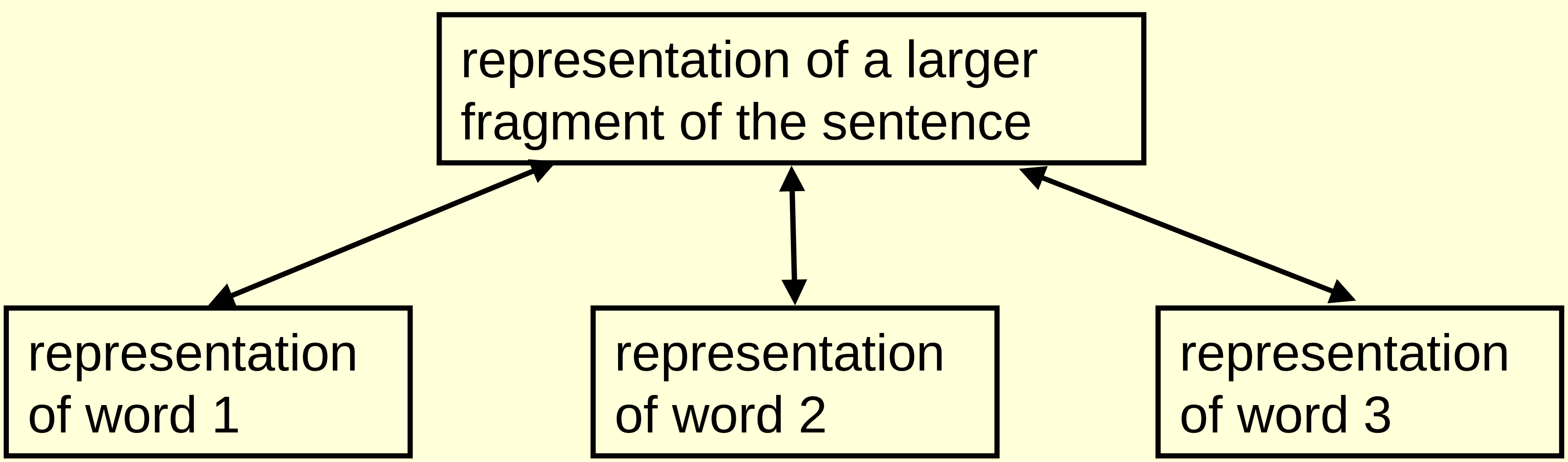
The reciprocal feedback connections

- Whenever one cortical area makes connections to a higher area there are always reciprocal connections coming back.
  - These “top-down” connections seem to have weaker effects than the bottom-up ones.

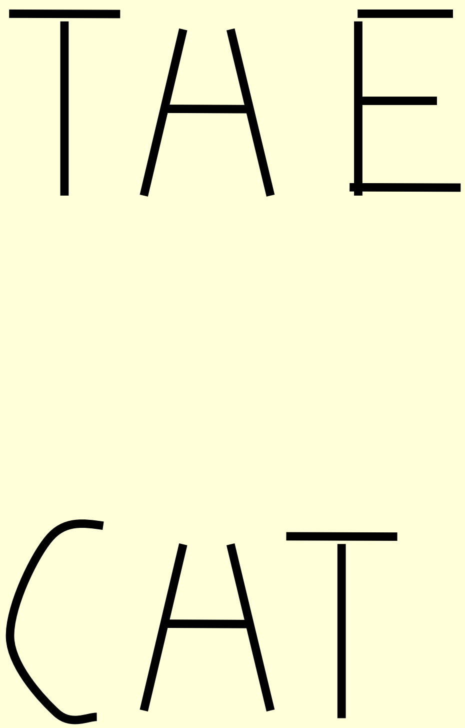
- Many functions have been suggested for the reciprocal connections:
  - Top-down effects in perception
  - A supervisory signal to facilitate learning
  - A way to enhance the object of attention and to suppress the background.

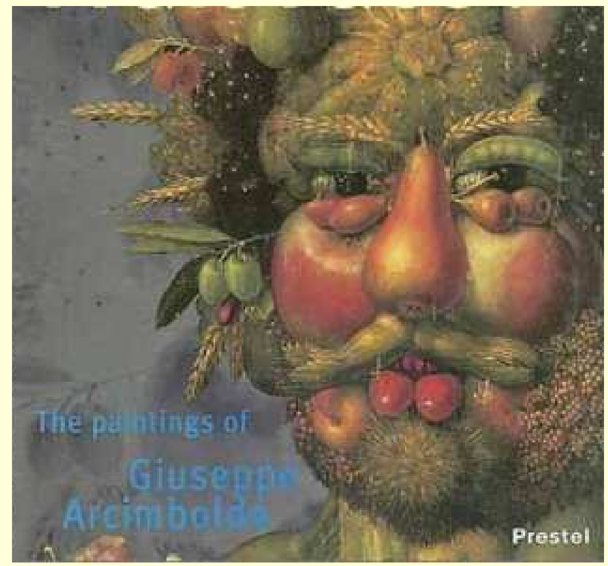
Some top-down effects in perception

- Consider the sentence:
  "She scromed him with the frying pan."
- You have a pretty good idea what "scromed" means. The context provided by the whole sentence makes strong predictions about the meaning of the word that occupies that role.



A whole influences the perception of its parts

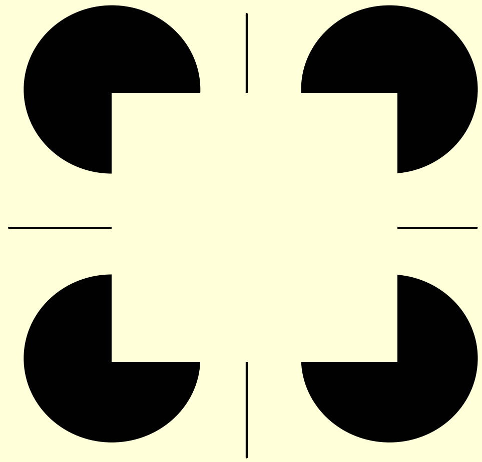


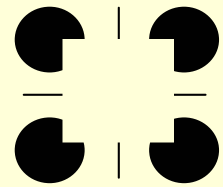


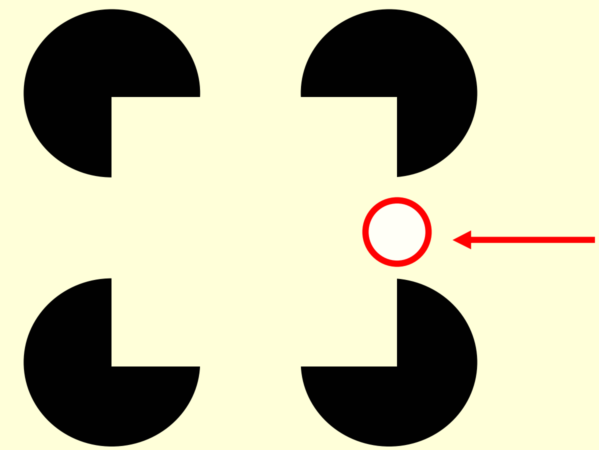
But does this happen during the formation of the first percept or during the subsequent formation of the percept for a part?

We see contours that are not really there

(and so do low-level neurons in a monkey's visual system)





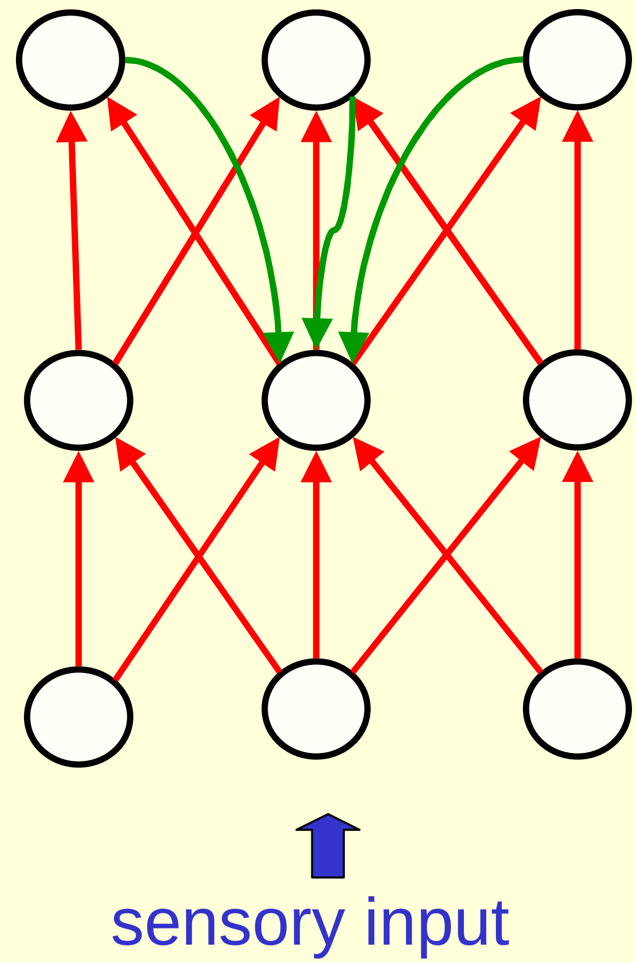


A neuron that detects a vertical line in this region will fire, but it fires much later than normal (Tai Sing Lee)

Generative models and perception

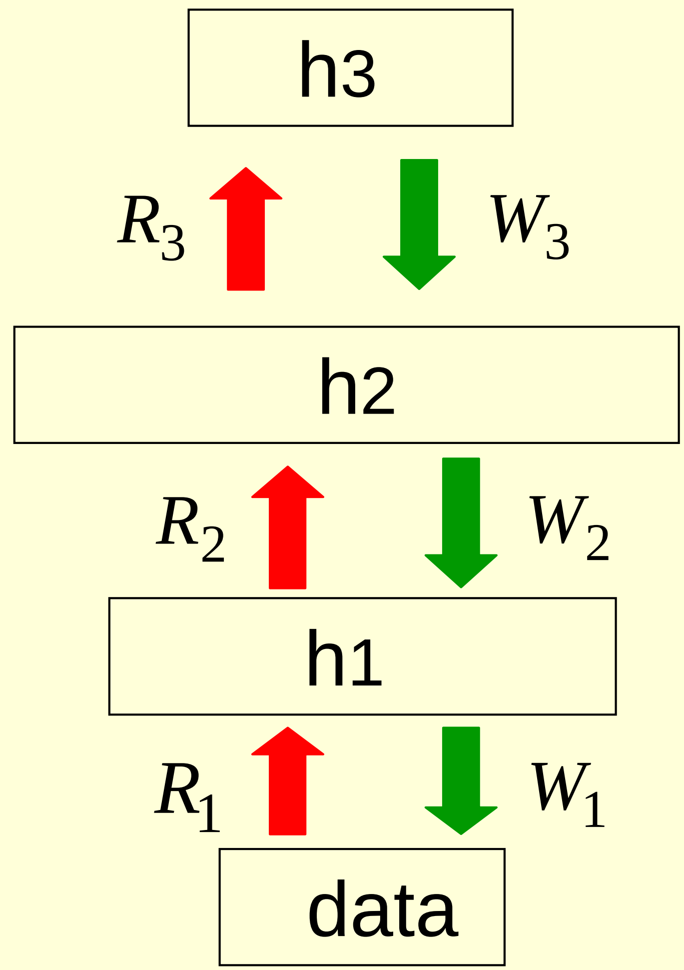
- Suppose that the top-down connections learn a generative model of the sensory input.
  - For visual input this would be like learning to do computer graphics.
  - Computer graphics converts a high-level representation into an image.

- Now we have to learn the top-down connections as well as the bottom-up ones.
  - This does not seem like progress!
  - But maybe the two sets of connections can train each other.



The wake-sleep algorithm

- Wake phase: Use the recognition weights to perform a bottom-up pass.
  - Train the generative weights to reconstruct activities in each layer from the layer above.
- Sleep phase: Use the generative weights to generate samples from the model.
  - Train the recognition weights to reconstruct activities in each layer from the layer below.

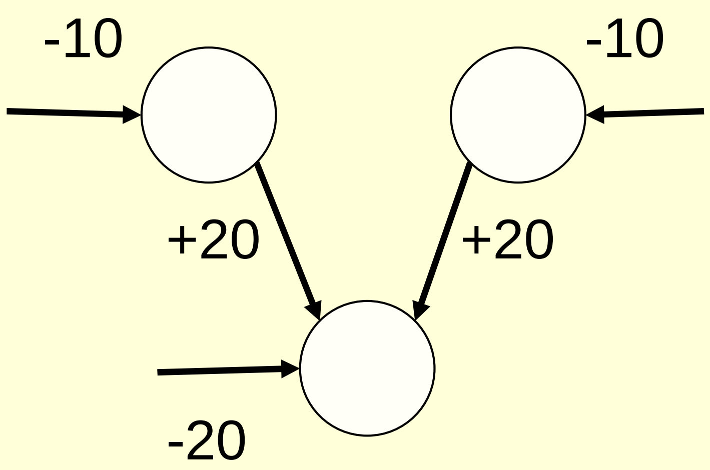


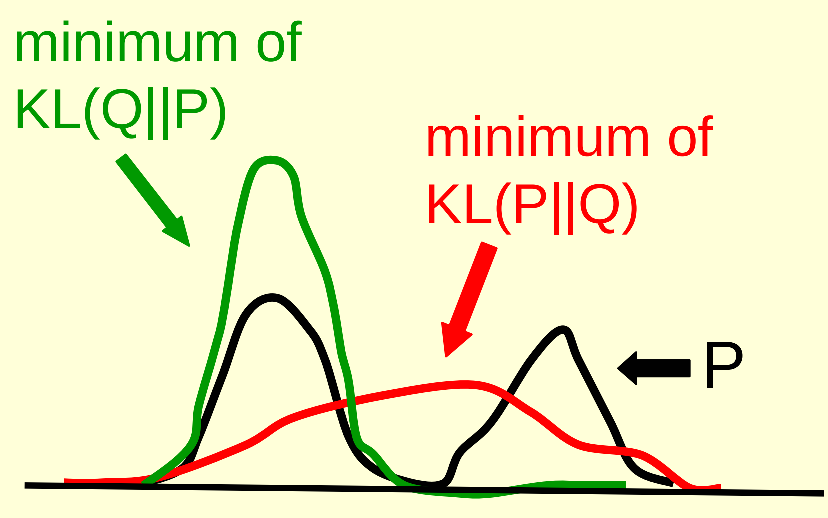
The flaws in the wake-sleep algorithm

- The recognition weights are trained to invert the generative model in parts of the space where there is no data.
  - This is wasteful.
- The recognition weights follow the gradient of the wrong divergence. They minimize KL(P||Q) but the variational bound requires minimization of KL(Q||P).
  - This leads to incorrect mode-averaging
- The posterior over the top hidden layer is very far from independent because the independent prior cannot eliminate explaining away effects.

Mode averaging

- If we generate from the model, half the instances of a 1 at the data layer will be caused by a (1,0) at the hidden layer and half will be caused by a (0,1).
  - So the recognition weights will learn to produce (0.5,0.5)
  - This represents a distribution that puts half its mass on very improbable hidden configurations.
- Its much better to just pick one mode and pay one bit.





The contrastive version of wake-sleep

- Replace the top layer of the DAG by an RBM
  - This eliminates bad variational approximations caused by top-level units that are independent in the prior.
  - It is nice to have an associative memory at the top.
- Replace the ancestral pass in the sleep phase by a top-down pass starting with the state of the RBM produced by the wake phase.
  - This makes sure the recognition weights are trained in the vicinity of the data.
  - It also reduces mode averaging. If the recognition weights prefer one mode, they will stick with that mode even if the generative weights like some other mode just as much.

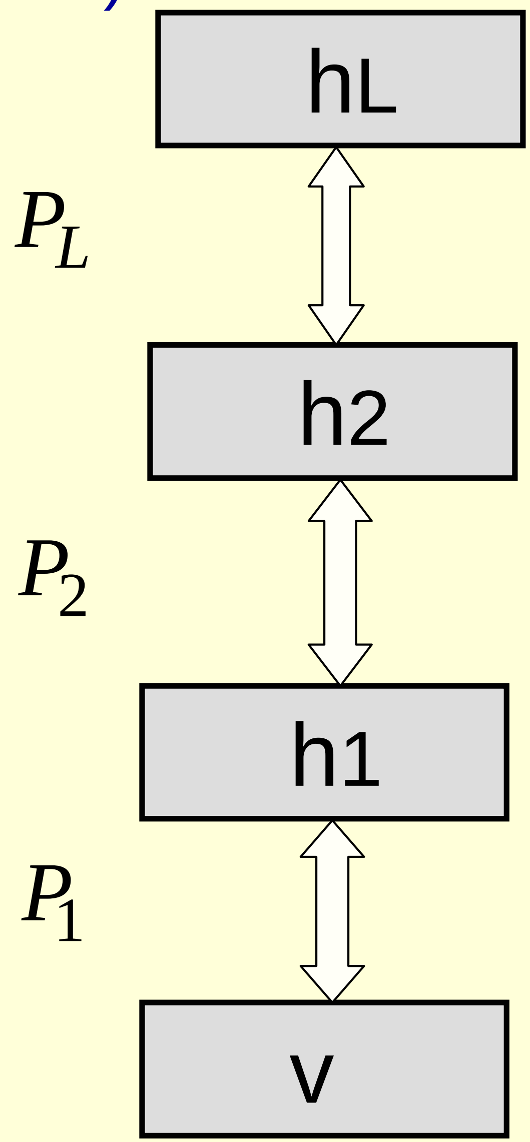
A stack of RBM's
(Yee-Whye Teh's picture)

- Each RBM has the same subscript as its hidden layer.
- Each RBM defines its own distribution over its visible vectors

$$P_l(h_{l-1}) = \frac{\sum_{h_l} \exp(-E(h_{l-1}, h_l))}{Z_l}$$

- Each RBM defines its own distribution over its hidden vectors

$$P_l(h_l) = \frac{\sum_{h_{l-1}} \exp(-E(h_{l-1}, h_l))}{Z_l}$$



The variational bound

Each time we replace the prior over the hidden units by a better prior, we win by the difference in the probability assigned

$$\log p(v) \geq \log P_1(v) + \sum_{l=1}^{l=L-1} \left\langle \log P_{l+1}(h_l) - \log P_l(h_l) \right\rangle_{Q(h_l|v)}$$

Now we cancel out all of the partition functions except the top one and replace log probabilities by goodnesses using the fact that:

$$\log P_l(x) = G_l(x) - \log Z_l$$

$$G(v) = \log \sum_h \exp(-E(v,h))$$

$$G(h) = \log \sum_v \exp(-E(v,h))$$

$$\log p(v) \geq G_1(v) + \sum_{l=1}^{l=L-1} \left\langle G_{l+1}(h_l) - G_l(h_l) \right\rangle_{Q(h_l|v)} - \log Z_L$$

This has simple derivatives that give a more justifiable

fine-tuning algorithm than contrastive wake-sleep.

Two expressions for G(v)

$$G(v) = \log \sum_h \exp(-E(v,h))$$

$$G(v) = \left\langle v_i h_j w_{ij} \right\rangle_{Q(h|v)} + entropy\big(Q(h \mid v)\big)$$

The conditional
distribution of h given v

Differentiating the bound

$$\log p(v) \geq G_1(v) + \sum_{l=1}^{l=L-1} \left\langle G_{l+1}(h_l) - G_l(h_l) \right\rangle_{Q(h_l|v)} - \log Z_L$$

- The derivatives of the bound with respect to a weight come from derivatives of G and derivatives of Q.
- The derivatives of G are simple.
- The derivatives via Q are trickier and require an approximation.

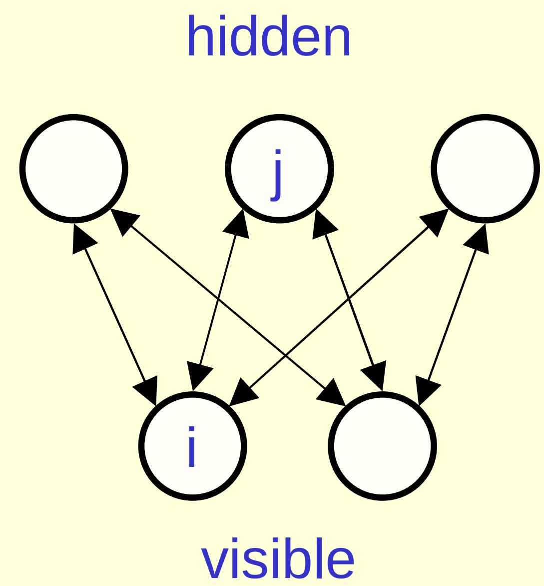
Derivatives of G

- With v fixed, we get a conditional distribution Q(h|v).
- If we then change w_ij by epsilon, two things happen:
  - The expected goodness (with Q held constant) changes by

$$\varepsilon \, v_i \left\langle h_j \right\rangle_{Q(h|v)}$$

  - The conditional distribution of h changes. But this has no effect on G because Q was chosen to minimize G

$$at\ Q(h\,|\,v), \quad \frac{\partial G(v)}{\partial Q(h)} = 0$$



I use * to mean the recognition distribution obtained on the first up-pass. v and h are the visible and hidden units of whatever RBM we are thinking about

$$\frac{\partial G(v^*)}{\partial w_{ij}} = v_i^* \langle h_j \rangle_{Q(h|v^*)}$$

$$\frac{\partial G(h^*)}{\partial w_{ij}} = h_j^* \langle v_i \rangle_{Q(v|h^*)}$$

$$\frac{\partial G(v^*)}{\partial w_{ij}} - \frac{\partial G(h^*)}{\partial w_{ij}} = h_j^* \left(v_i^* - \langle v_i \rangle_{Q(v|h^*)} \right)$$

in expectation

The derivatives via Q

- We need to know how G(v) changes when the probability of turning on v_j changes.
- What we really want is:

$$G(v \mid v_i = 1) - G(v \mid v_i = 0)$$

- But this would require sampling h twice for each visible unit.
- What if we assume that all the weights from v_j are small?
  - Flipping the binary state of v_j will only cause a small change in h so, to first order, we can ignore the change in h because

$$at\ Q(h \mid v), \quad \frac{\partial G(v)}{\partial Q(h)} = 0$$

Expected changes in energy caused by changing the probability of turning on a unit

- For the RBM in which the unit is visible

$$\frac{\partial G(v)}{\partial \langle v_i \rangle} = entropyterm + \sum_j w_{ij} \langle h_j \rangle_{Q(h|v^*)}$$

- By symmetry, for the RBM below

$$\frac{\partial G(h)}{\partial \langle h_i \rangle} = entropyterm + \sum_j w_{ij} \langle v_j \rangle_{Q(v|h)}$$

Combining the via Q derivatives from the higher and lower RBM's

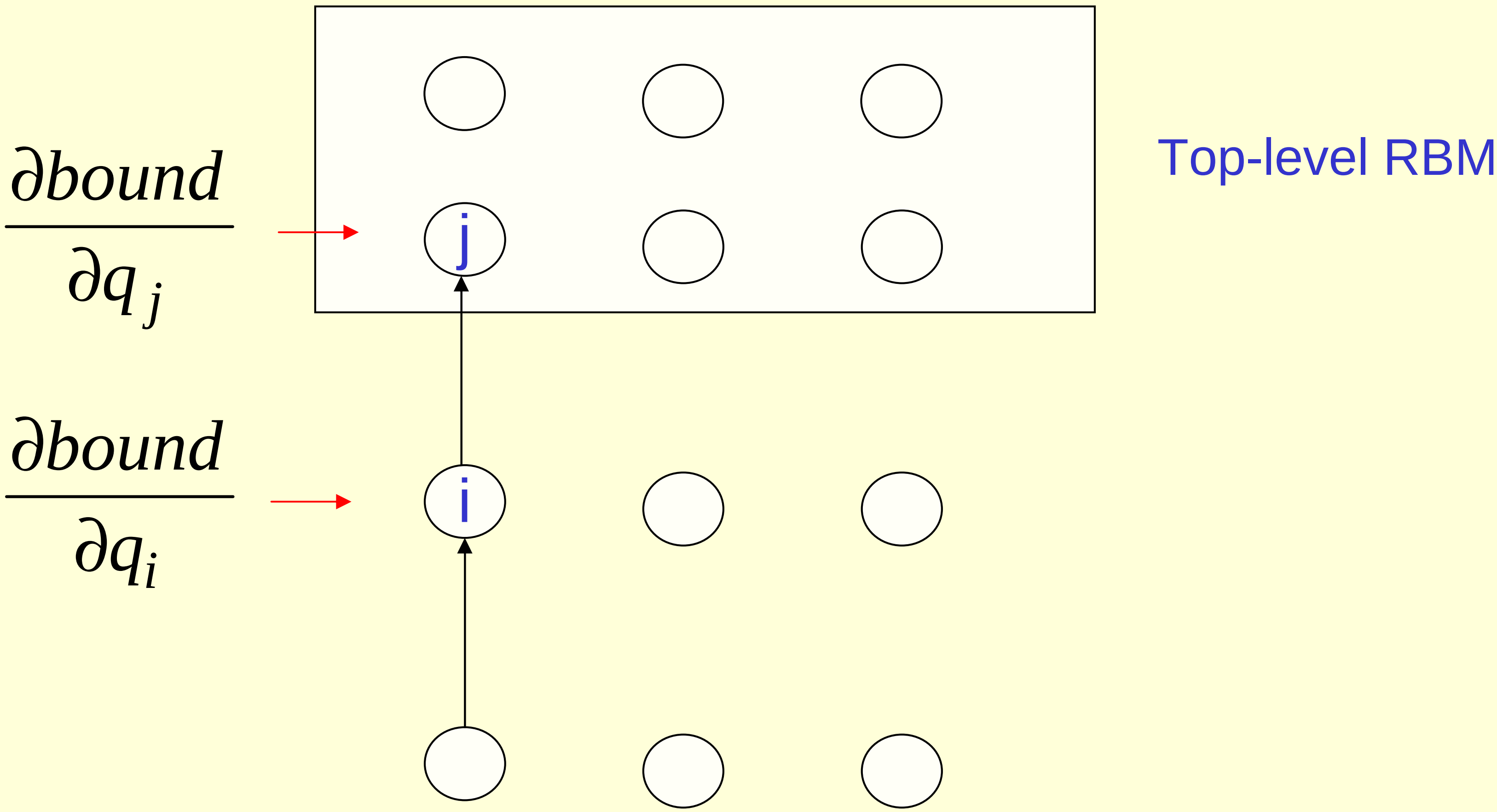
$$\frac{\partial bound}{\partial \langle q_i \rangle} = \sum_j w_{ij} \langle h_j \rangle_{Q(h|v^*)} - \sum_j w_{ij} \langle v_j \rangle_{Q(v|h^*)}$$

top-down input

bottom-up input from a one-step, top-down reconstruction

If we use mean field inference in which the hidden units have real valued activities, this derivative is not approximate.

Back-propagating the derivatives that come from changing Q



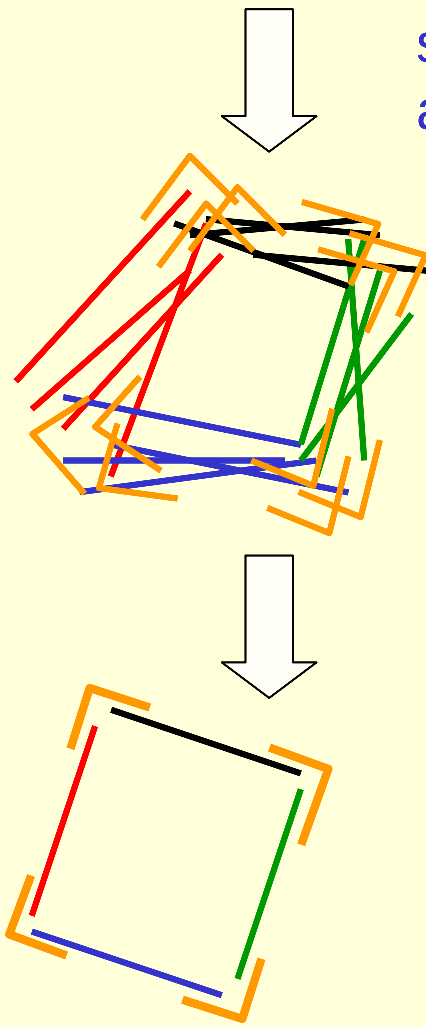
Start with the visible units of the top-level RBM and back-propagate, adding in the derivative of the bound at each level.

Change of topic

Generating the parts of an object

- One way to maintain the constraints between the parts is to generate each part very accurately
  - But this would require a lot of communication bandwidth.
- Sloppy top-down specification of the parts is less demanding
  - but it messes up relationships between features
  - so use redundant features and use lateral interactions to clean up the mess.
- Each transformed feature helps to locate the others
  - This allows a noisy channel

“square” + pose parameters



sloppy top-down activation of parts

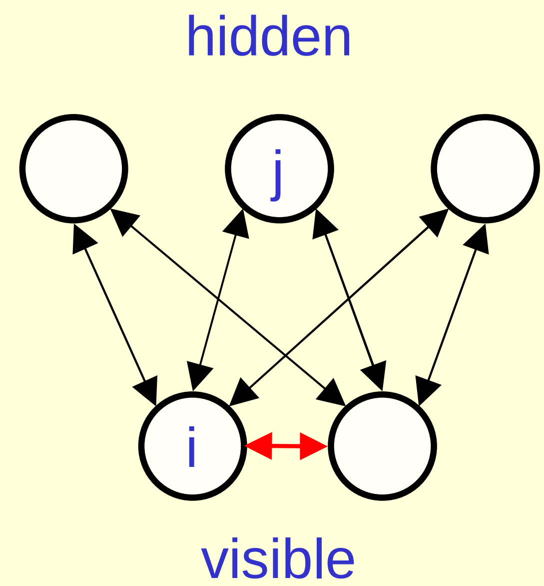
features with top-down support

clean-up using known interactions

Its like soldiers on a parade ground

Semi-restricted Boltzmann Machines

- We restrict the connectivity to make learning easier.
- Contrastive divergence learning requires the hidden units to be in conditional equilibrium with the visibles.
  – But it does not require the visible units to be in conditional equilibrium with the hiddens.
  – All we require is that the visible units are closer to equilibrium in the reconstructions than in the data.
- So we can allow connections between the visibles.



Learning in SRBM's

- Method 1: To form a reconstruction, cycle through the visible units updating each in turn using the top-down input from the hiddens plus the lateral input from the other visibles.
- Method 2: Use "mean field" visible units that have real values. Update them all in parallel.
  - Use damping to prevent oscillations

$$p_i^{t+1} = \lambda p_i^t + (1-\lambda)\,\sigma(x_i)$$

damping

total input to i

Show results in paper

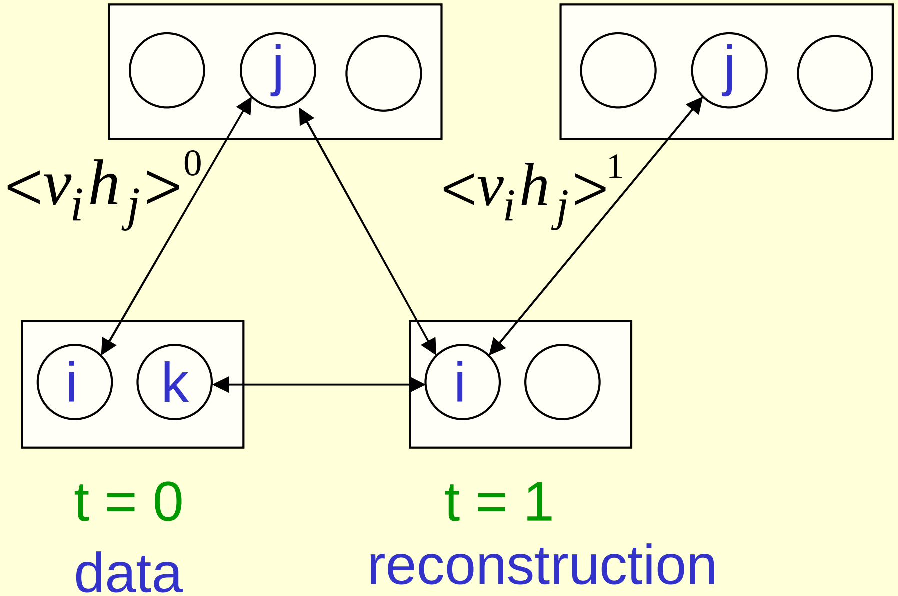
Why do we whiten data?

- Images typically have strong pair-wise correlations.
- Learning higher order statistics is difficult when there are strong pair-wise correlations.
  - Small changes in parameter values that improve the modeling of higher order statistics may be rejected because they form a slightly worse model of the much stronger pair-wise statistics.

Whitening the learning signal instead of the data

- Contrastive divergence learning can remove the effects of the second-order statistics on the learning without actually changing the data.
  - The lateral connections model the second order statistics
  - If a pixel can be reconstructed correctly using second order statistics, its will be the same in the reconstruction as in the data.
  - The hidden units can then focus on modeling high-order structure that cannot be predicted by the lateral connections.

learning an SRBM



Start with a training vector on the visible units.

Update all the hidden units in parallel

Update the all the visible units in parallel to get a “reconstruction”.

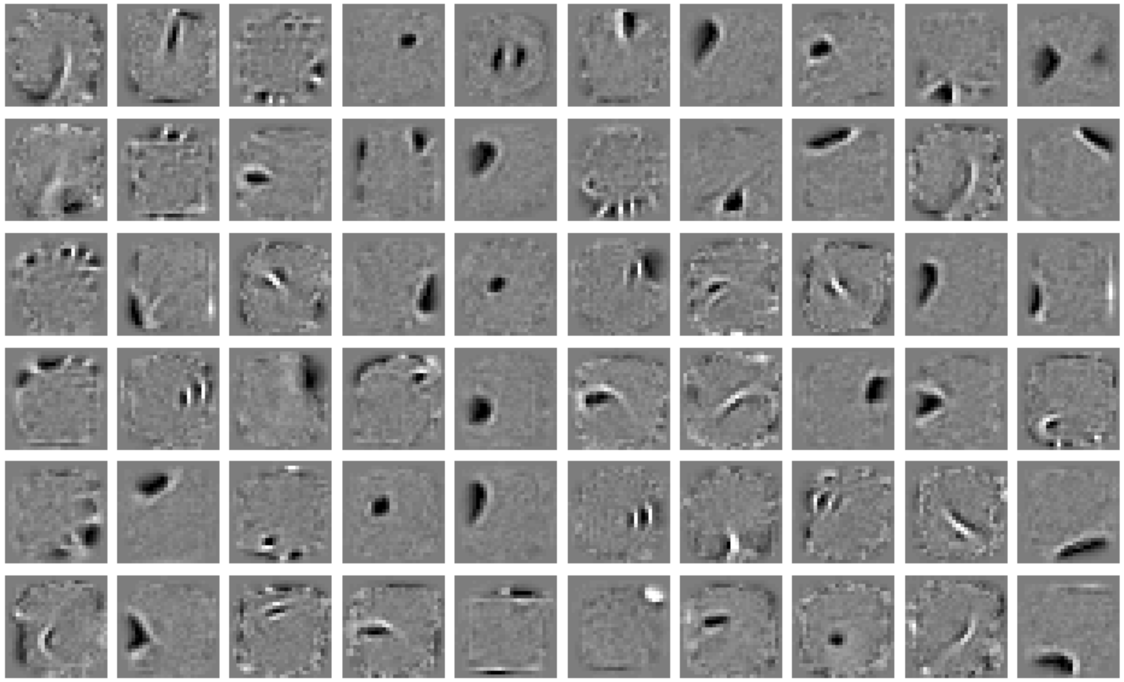
Update the hidden units again.

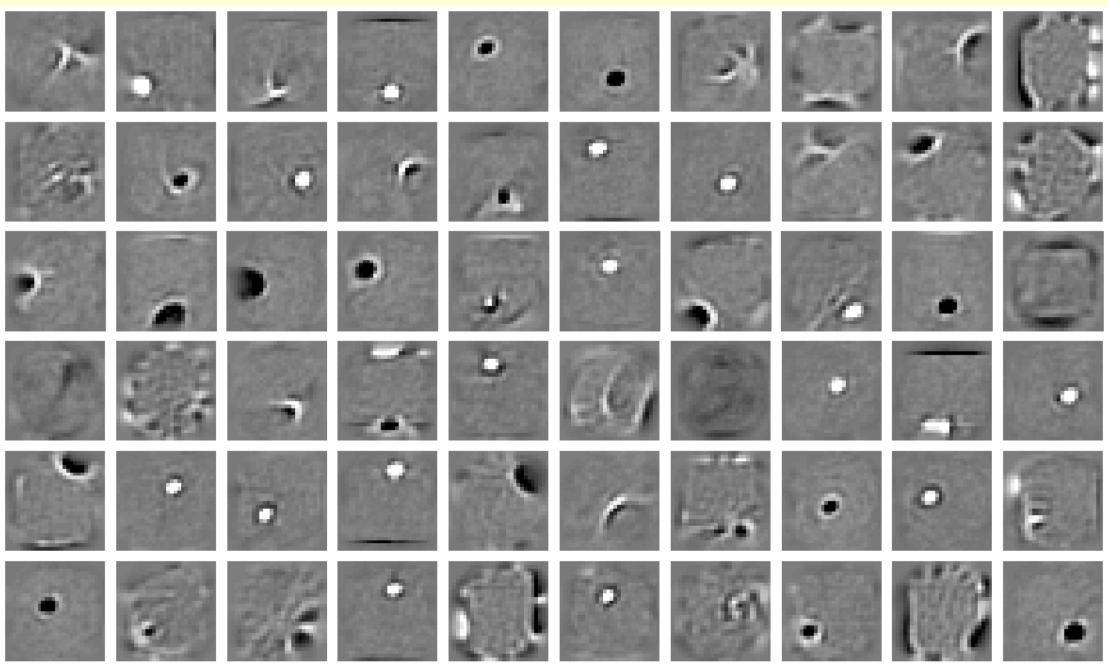
$$\Delta w_{ij} = \varepsilon \left(\langle v_i h_j \rangle^0 - \langle v_i h_j \rangle^1 \right)$$

$$\Delta l_{ik} = \varepsilon \left(\langle v_i v_k \rangle^0 - \langle v_i v_k \rangle^1 \right)$$

A funny way to use an MRF

- The lateral connections form an MRF.
- The MRF is used during learning and generation.
- The MRF is not used for inference.
  - This is a novel idea so vision researchers don't like it.
- The MRF enforces constraints. During inference, constraints do not need to be enforced because the data obeys them.
  - The constraints only need to be enforced during generation.
- Unobserved hidden units cannot enforce constraints.
  - This requires lateral connections or observed descendants.





Hidden fields on mnist digits.

One model uses laterals between the visibles and the other doesn't.

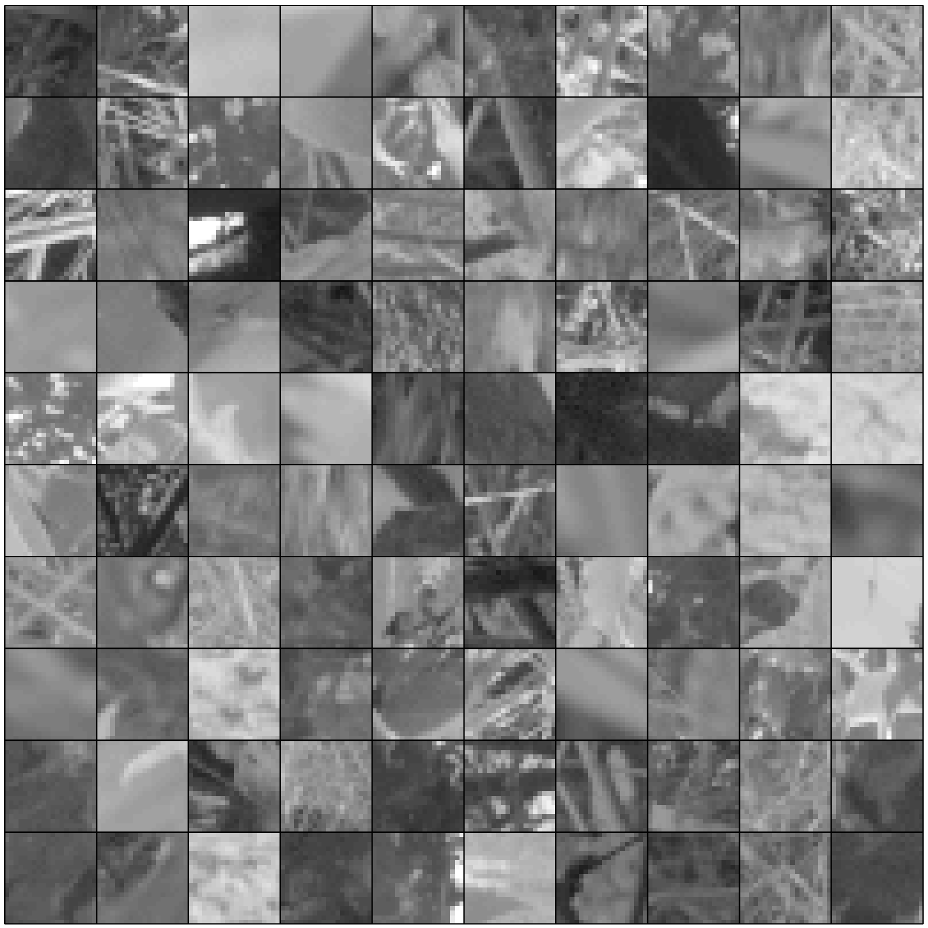
Which is which?

Results on modeling natural image patches using a stack of RBM's (Osindero and Hinton)

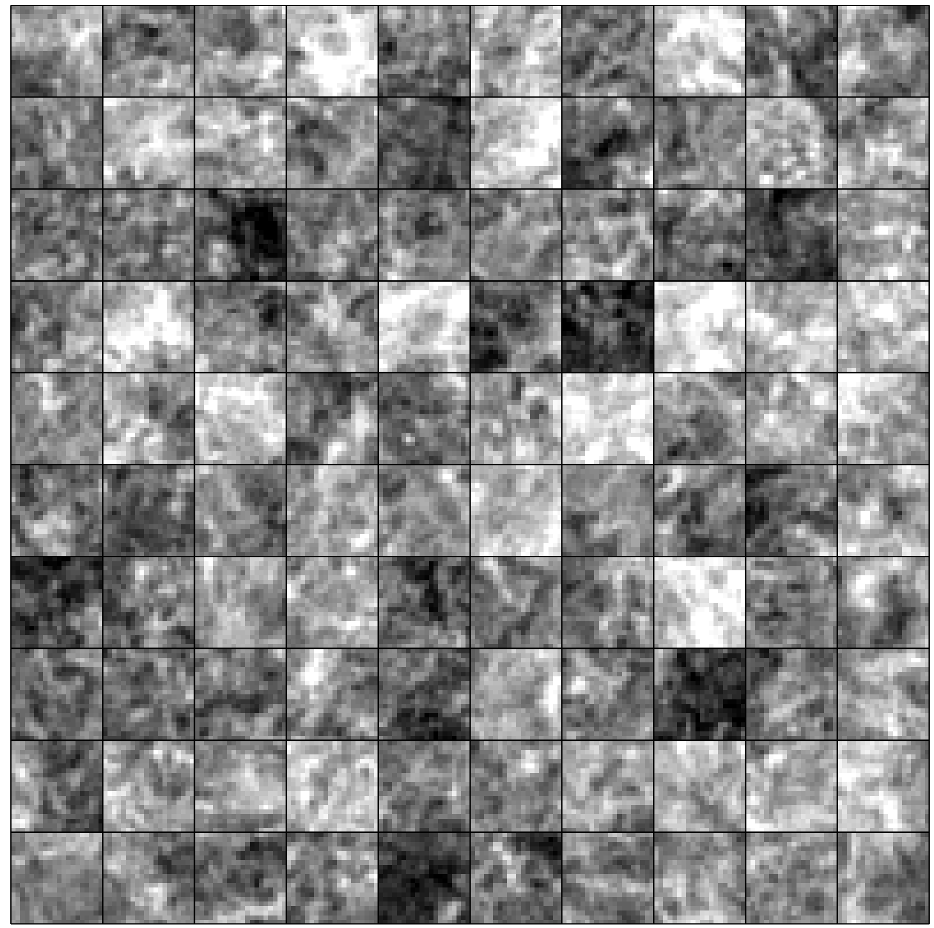
- 100,000 Van Hateren image patches, each 20x20
- Stack of RBM's learned one at a time.
- 400 Gaussian visible units that see whitened image patches.
- 400→2000→500→1000
- Hidden units are all binary with learned lateral connections when they are the visible units of their RBM.
- Generation involves letting the visible units of each RBM settle using mean field with the already decided states in the level above determining the effective biases.

Without lateral connections

real data

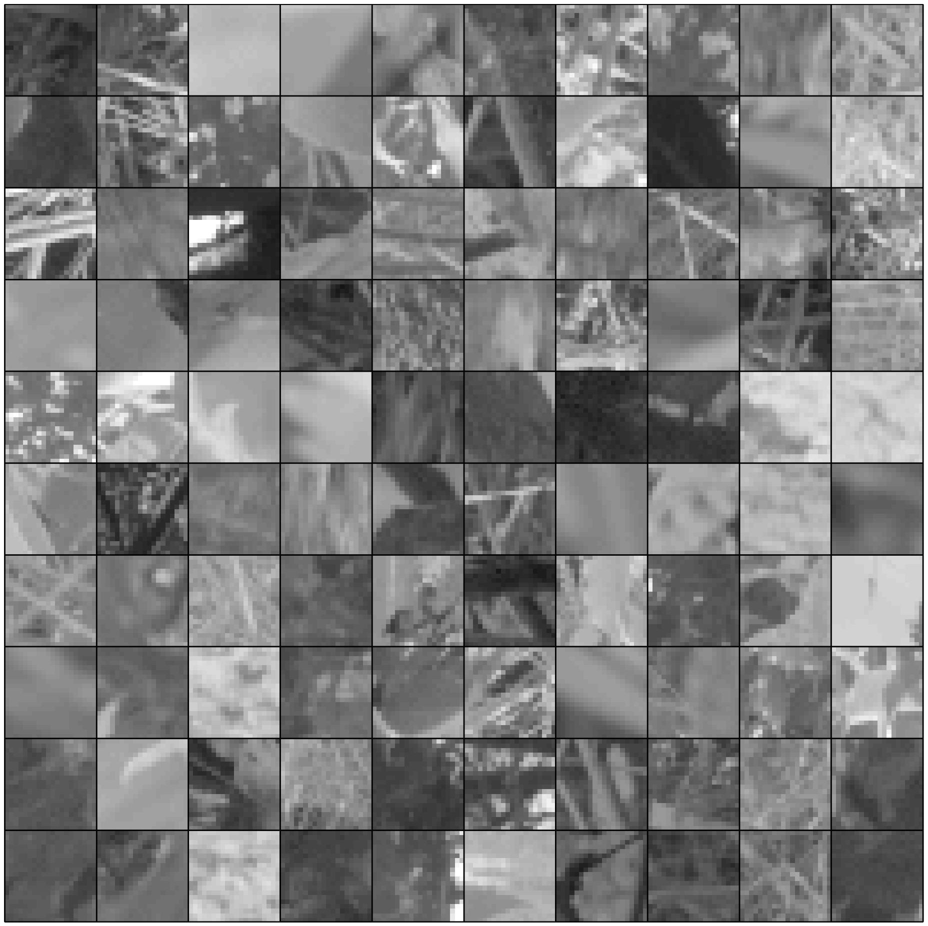


samples from model

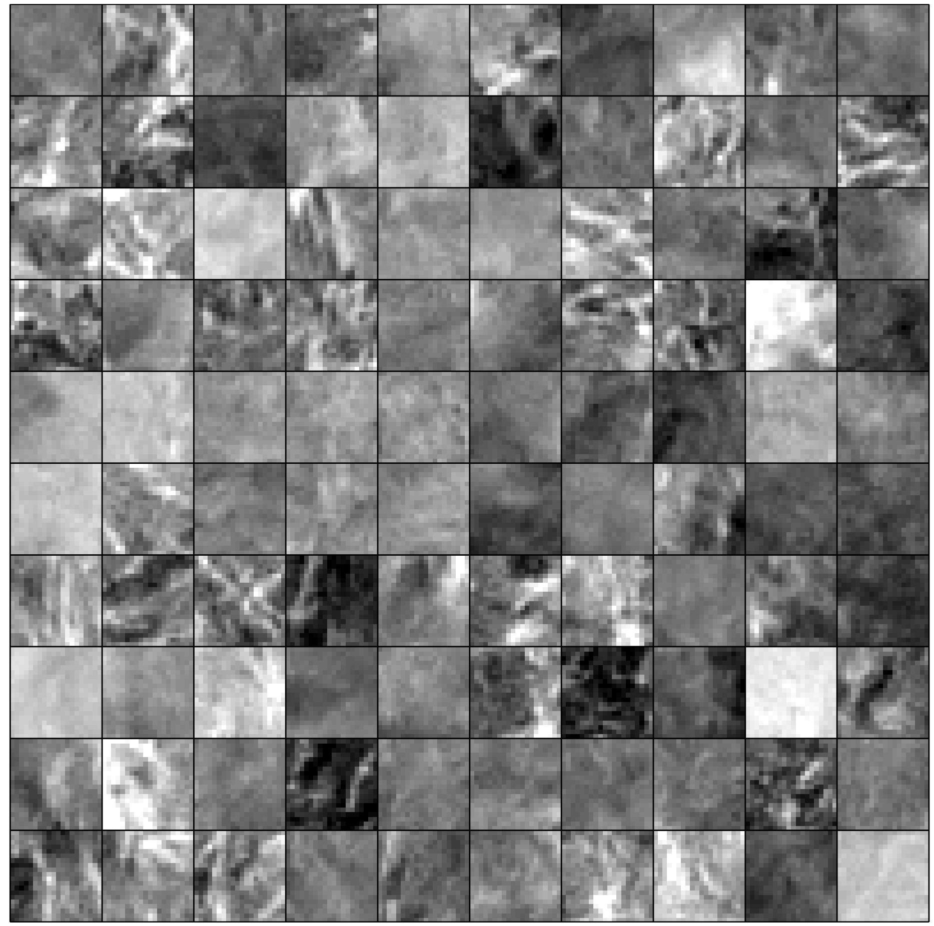


With lateral connections

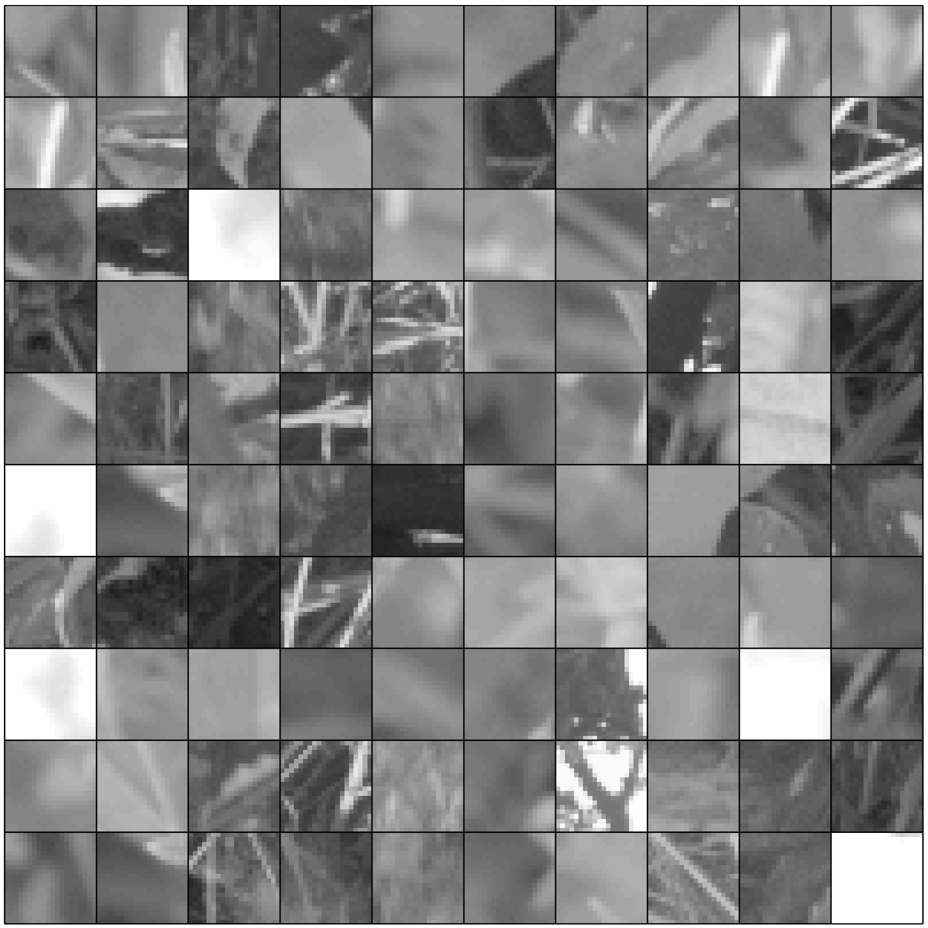
real data

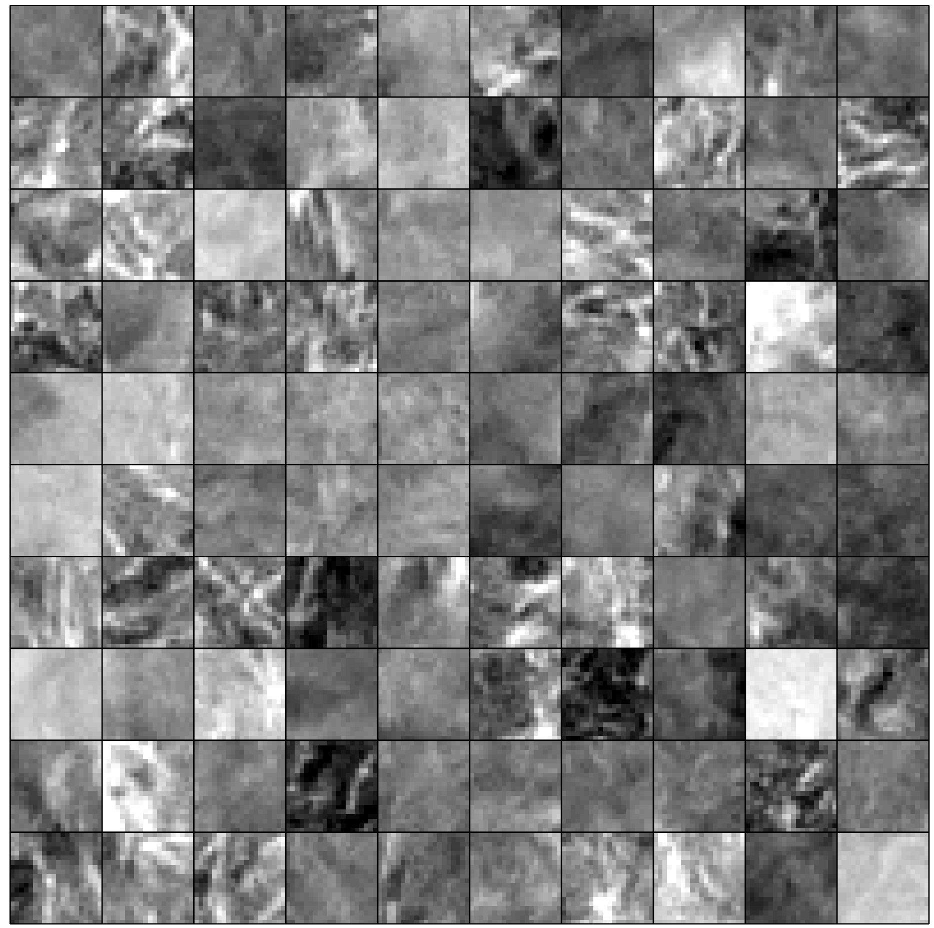


samples from model

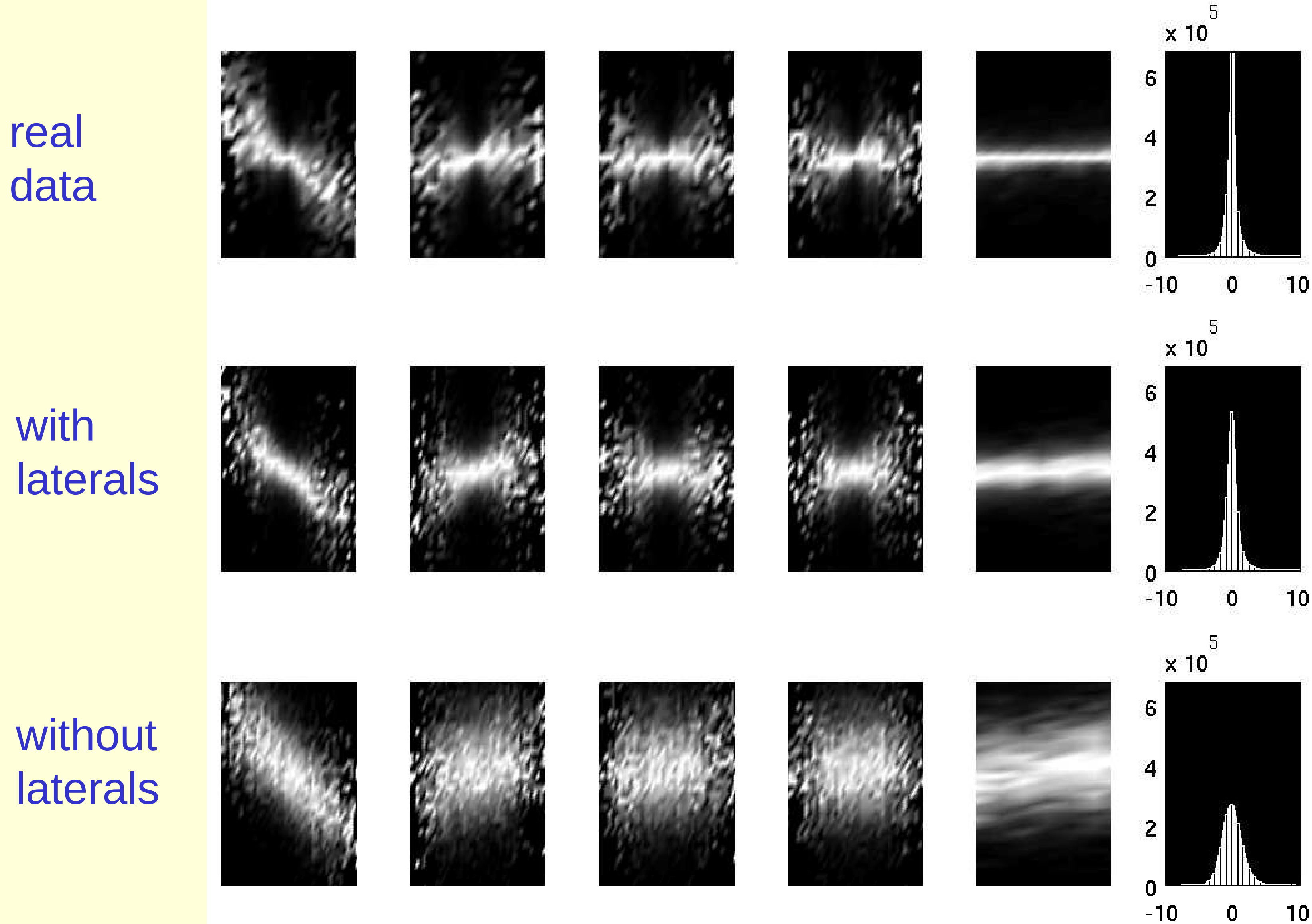


Closest images in training set



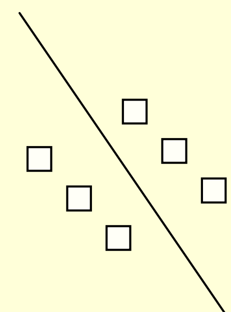


Statistics of filter outputs



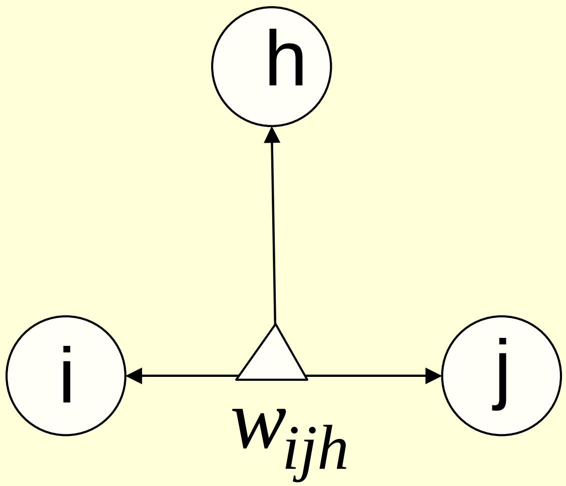
What is an edge?

- Its hard to get a robust definition because what we really mean by an edge is a breakdown in the correlational structure of the image.
  - You cannot predict pixels across an occluding edge.



Higher-order RBM's are CRF's

(see article in Scholarpedia on Boltzmann machines)



$$E = -\sum_{ijh} s_i s_j (s_h w_{ijh})$$