Unsupervised Learning

Peter Dayan
GCNU

Dayan & Abbott
Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems.
MIT Press, 2001.

Statistics and Development

activity-dependent wiring

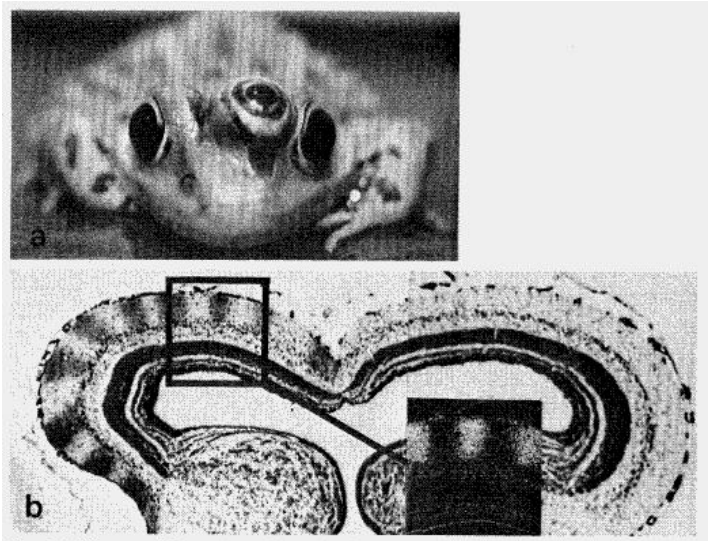
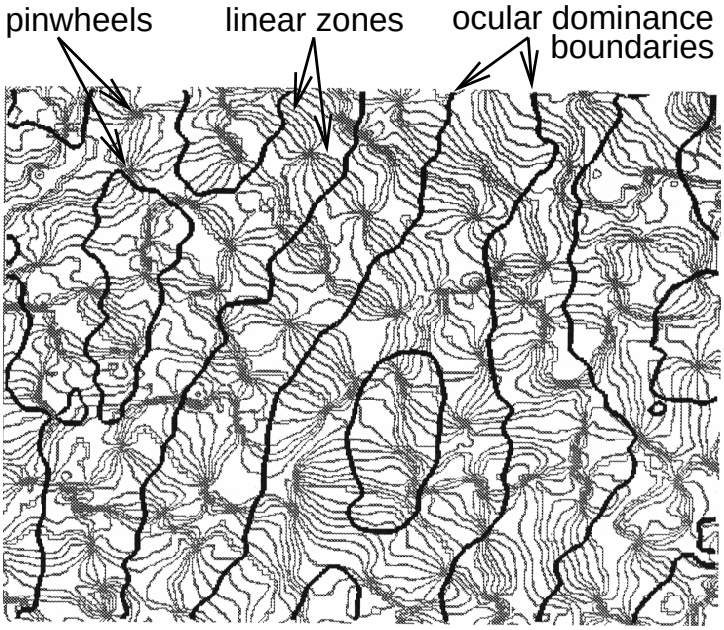
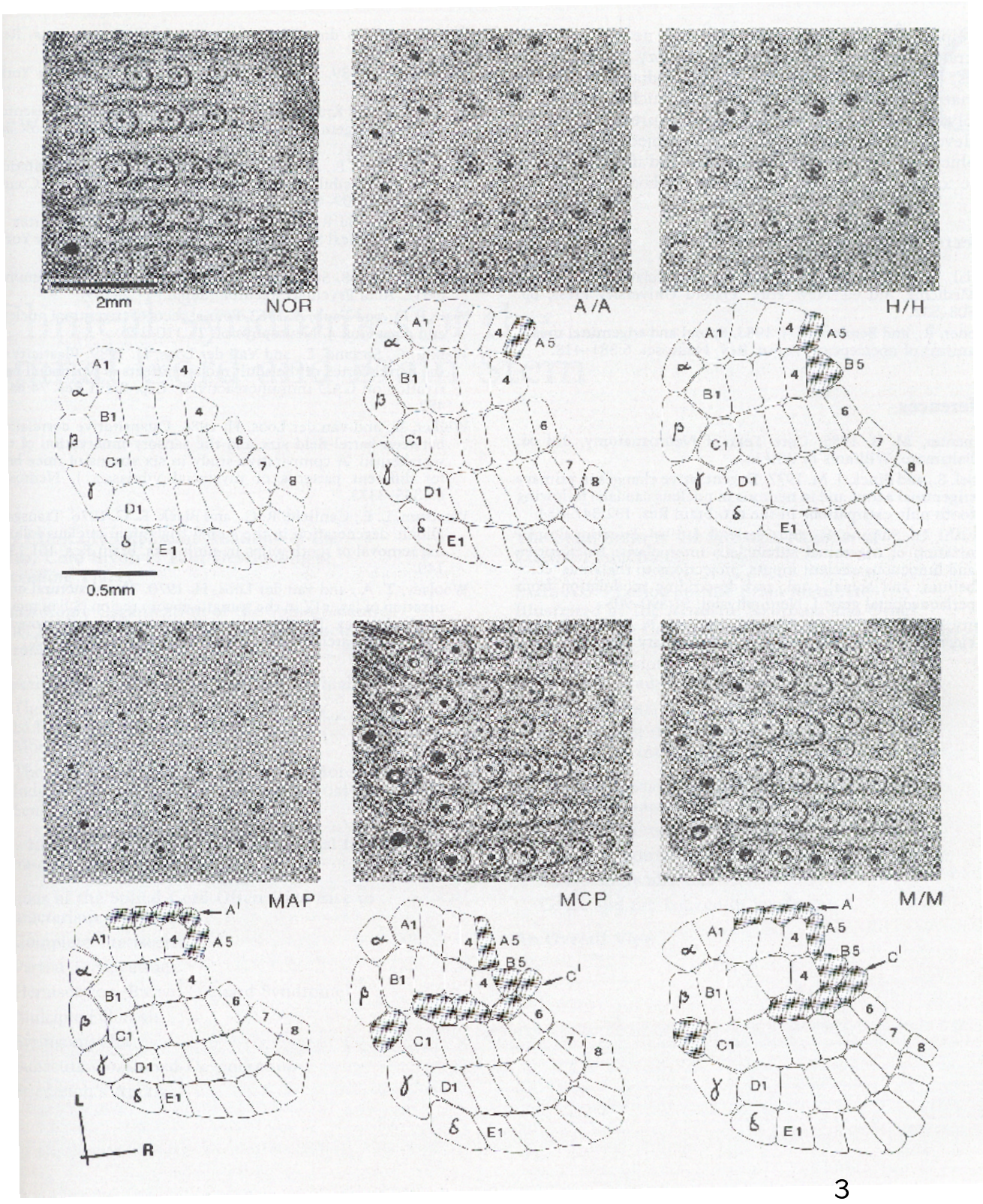


Fig. 1. (a) Three-eyed *Rana pipiens* 8 months after metamorphosis. The central eye primordium was implanted at Shumway stage 17 from a similarly staged donor. The supernumerary eye has externally normal dimensions, but lacks a pupillary response. (b) Autoradiographic distributions of grain densities in the optic tectum of a 3-month postmetamorphic three-eyed frog after injection of 10 μCi of [^{3}H]proline into the vitreous body of the normal eye. (Inset) Dark-field enlargment showing the pronounced segregation of labeled and unlabeled regions of the tectal neuropil.



Barrel Cortex



Modeling Development

Two strategies:

mathematical understand the selectivities and the patterns of selectivities from the perspective of pattern formation:

- *reaction diffusion equations*
- *symmetry breaking*

based on underlying mechanisms of plasticity such as Hebbian learning

computational understand the *selectivities* from basic principles of processing:

- *extraction*
- *representation*

of statistical structure.

Understand *patterns* using other principles, *eg* minimal wiring volume

Unsupervised Learning

Work with:

$$P_I[\mathbf{x}] = \frac{1}{M}\sum_{\mu=1}^{M}\delta(\mathbf{x}-\mathbf{x}^{\mu})$$

or continuous version, and *prior* information about what is important.

Way of quantifying quality of representation:

- good reconstruction
- cheapness/brevity
- independence
- sparsity

maybe no general answer?

Two Classes of Method

density estimation attempt to *fit* $P_I[\mathbf{x}]$ using a model with hidden structure or *causes*:

$$P[\mathbf{x}|\mathbf{y};\mathcal{G}]$$

leading to:

$$P_I[\mathbf{x}] \sim P[\mathbf{x};\mathbf{G}] = \sum_{\mathbf{y}} P[\mathbf{x}^\mu, \mathbf{y};\mathcal{G}].$$

too:

stringent texture

lax lookup table

FA; MoG; sparse coding; ICA; Boltzmann machine; Helmholtz machine; HMM; Kalman filter; graphical models

structure search look for unusual structure (projection pursuit); particular regularities (stereo)

too *unsystematic*.

ML Density Estimation

Make:

$$P_I[\mathbf{x}] = P[\mathbf{x}; \mathcal{G}] = \sum_{\mathbf{y}} P[\mathbf{x}, \mathbf{y}; \mathcal{G}]$$

to model how $\mathbf{x}$ might have been *generated* or *caused*. **Synthetic** model.

Key quantity is the **analytical** model:

$$P[\mathbf{y}|\mathbf{x}; \mathcal{G}] = \frac{P[\mathbf{x}, \mathbf{y}; \mathcal{G}]}{\sum_{\mathbf{y}'} P[\mathbf{x}, \mathbf{y}'; \mathcal{G}]}$$

learning $\mathcal{G}$ on the basis of examples captures the overall statistical structure in the collection of patterns

representing $\mathbf{x}$ **using** $P[\mathbf{y}|\mathbf{x}; \mathcal{G}]$ indicates the possible generators of $\mathbf{x}$.

strong assumption

Fitting $\mathcal{G}$

Minimise $\quad \text{argmin}_{\mathcal{G}} KL\left[P_I[\mathbf{x}], P[\mathbf{x};\mathcal{G}]\right]$

$$= \text{argmin}_{\mathcal{G}} \sum_{\mathbf{x}} P_I[\mathbf{x}] \log \frac{P_I[\mathbf{x}]}{P[\mathbf{x};\mathcal{G}]}$$

$$= \text{argmax}_{\mathcal{G}} \sum_{\mathbf{x}} P_I[\mathbf{x}] \log P[\mathbf{x};\mathcal{G}]$$

$$= \text{argmax}_{\mathcal{G}} \sum_{\mu} \log P[\mathbf{x}^\mu;\mathcal{G}]$$

$$= \text{argmax}_{\mathcal{G}} \prod_{\mu} P[\mathbf{x}^\mu;\mathcal{G}]$$

is maximum likelihood density estimation.

Analysis Optimises Synthesis

If we can calculate or sample from the exact analytical distribution $P[\mathbf{y}|\mathbf{x};\mathcal{G}]$, then we have:

$$\begin{aligned}
\nabla_{\mathcal{G}} \log P[\mathbf{x}^{\mu};\mathcal{G}] &= \nabla_{\mathcal{G}} \log \sum_{\mathbf{y}} P[\mathbf{x},\mathbf{y};\mathcal{G}] \\
&= \sum_{\mathbf{y}} \frac{1}{\sum_{\mathbf{y}'} P[\mathbf{x},\mathbf{y}';\mathcal{G}]} \nabla_{\mathcal{G}} P[\mathbf{x},\mathbf{y};\mathcal{G}] \\
&= \sum_{\mathbf{y}} \frac{P[\mathbf{x},\mathbf{y};\mathcal{G}]}{\sum_{\mathbf{y}'} P[\mathbf{x},\mathbf{y}';\mathcal{G}]} \nabla_{\mathcal{G}} \log P[\mathbf{x},\mathbf{y};\mathcal{G}] \\
&= \sum_{\mathbf{y}} P[\mathbf{y}|\mathbf{x};\mathcal{G}] \nabla_{\mathcal{G}} \log P[\mathbf{x},\mathbf{y};\mathcal{G}]
\end{aligned}$$

a form that is handy for *exponential family distributions:* and underlies the Boltzmann machine

Helmholtz Free Energy

Is there a better way, or one involving only an approximate

$$\mathcal{Q}[\mathbf{y};\mathbf{x}^\mu] \simeq P[\mathbf{y}|\mathbf{x};\mathcal{G}] \ ?$$

Jensen's inequality:

$$\begin{aligned}\log P[\mathbf{x}^\mu;\mathcal{G}] &= \log \sum_{\mathbf{y}} P[\mathbf{x}^\mu,\mathbf{y};\mathcal{G}] \\ &\geq \sum_{\mathbf{y}} \mathcal{Q}[\mathbf{y};\mathbf{x}^\mu] \log \frac{P[\mathbf{x}^\mu,\mathbf{y};\mathcal{G}]}{\mathcal{Q}[\mathbf{y};\mathbf{x}^\mu]}\end{aligned}$$

with equality iff $\mathcal{Q}[\mathbf{y};\mathbf{x}^\mu] \propto P[\mathbf{x}^\mu,\mathbf{y};\mathcal{G}]$, *ie*

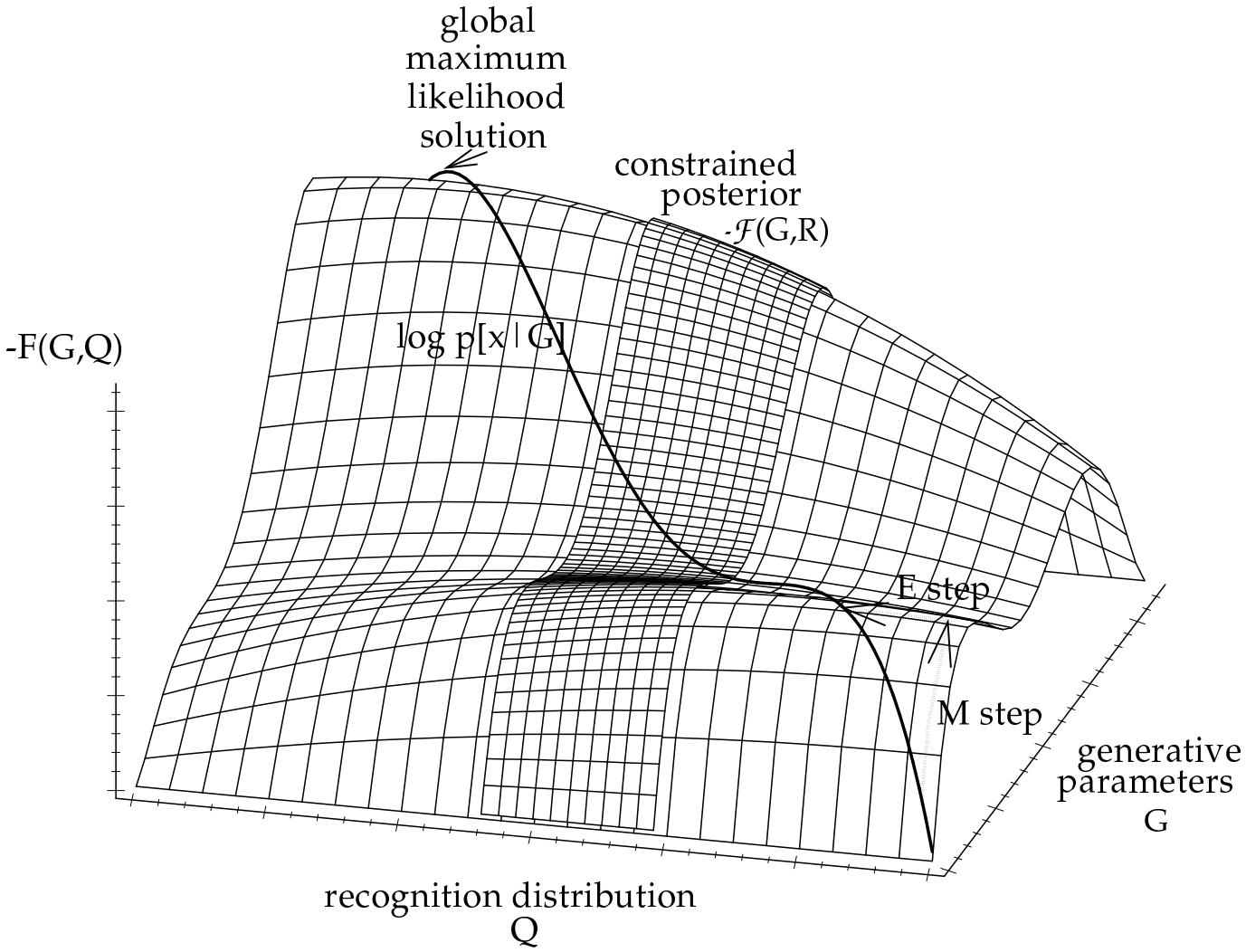
$$\mathcal{Q}[\mathbf{y};\mathbf{x}^\mu] = \frac{P[\mathbf{x}^\mu,\mathbf{y};\mathcal{G}]}{\sum_{\mathbf{y}'} P[\mathbf{x}^\mu,\mathbf{y}';\mathcal{G}]} = P[\mathbf{y}|\mathbf{x}^\mu;\mathcal{G}].$$

Minimise in $\mathcal{Q}$ and $\mathcal{G}$, $\mathcal{F}[\mathbf{x}^\mu,\mathcal{Q};\mathcal{G}] =$

$$-\sum_{\mathbf{y}} \mathcal{Q}[\mathbf{y};\mathbf{x}^\mu] \log P[\mathbf{x}^\mu,\mathbf{y};\mathcal{G}] + \sum_{\mathbf{y}} \mathcal{Q}[\mathbf{y};\mathbf{x}^\mu] \log \mathcal{Q}[\mathbf{y};\mathbf{x}^\mu]$$

EM is coordinatewise descent in this.

Graphical Interpretation



ML Method

Stochastic grad desc on $\sum_{\mu=1}^{\Omega} \mathcal{F}[\mathbf{x}^\mu, \mathcal{Q}; \mathcal{G}]$:

$$\Delta\mathcal{G} \propto -\nabla_{\mathcal{G}}\mathcal{F}[\mathbf{x}^\mu, \mathcal{Q}; \mathcal{G}].$$

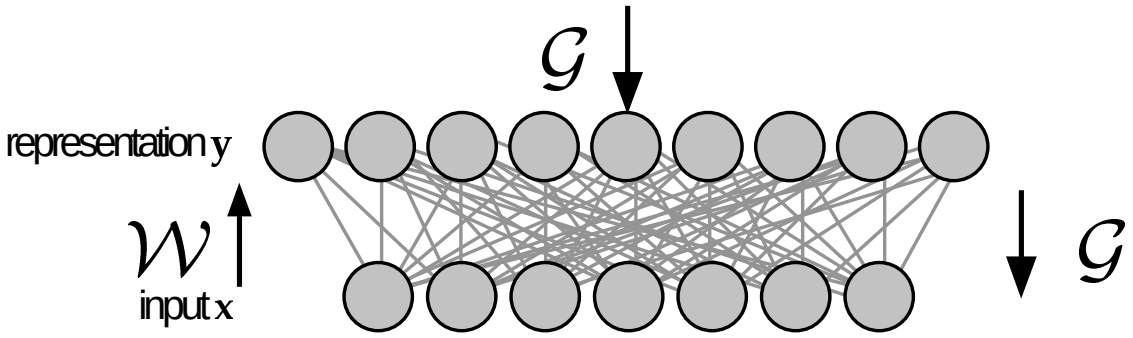
with $\mathcal{Q} =$:

exact MoG, FA

learned HM

iterative approximation mean field, Olshausen & Field, ICA

stochastic wake-sleep



where $\mathcal{W}$ parameterises the analytical model.

Factor Analysis

- PCA: minimise $\sum_{\mathbf{x}^\mu} |\mathbf{x}^\mu - \mathcal{G}\mathbf{y}^\mu|^2$ implies that $\mathcal{G}$ contains the first principal components of the covariance matrix of the $\mathbf{x}^\mu$, and the $\mathbf{y}^\mu$ are the 'loadings' of those PCs for example μ. Cares about *variance* and lacks a generative model of the data.

- FA: fit a model:

$$\begin{array}{rlll} \mathbf{y} \sim \mathcal{N}[\mathbf{0}, \sigma^2\mathbf{I}] & P[\mathbf{y};\mathcal{G}] & \propto & e^{-|\mathbf{y}|^2/2\sigma^2} \\ \mathbf{x}|\mathbf{y} \sim \mathcal{N}[\mathbf{G}\mathbf{y}, \Psi] & P[\mathbf{x}|\mathbf{y};\mathcal{G}] & \propto & e^{-(\mathbf{x}-\mathbf{G}\mathbf{y})^T\Psi^{-1}(\mathbf{x}-\mathbf{G}\mathbf{y})/2} \end{array}$$

  with $\Psi = \text{diag}\,(\tau_1^2, \tau_2^2, \ldots, \tau_n^2)$.

  - only cares about *covariance*
  - has a generative model of the data
  - has *a priori* independent factors $\mathbf{y}$

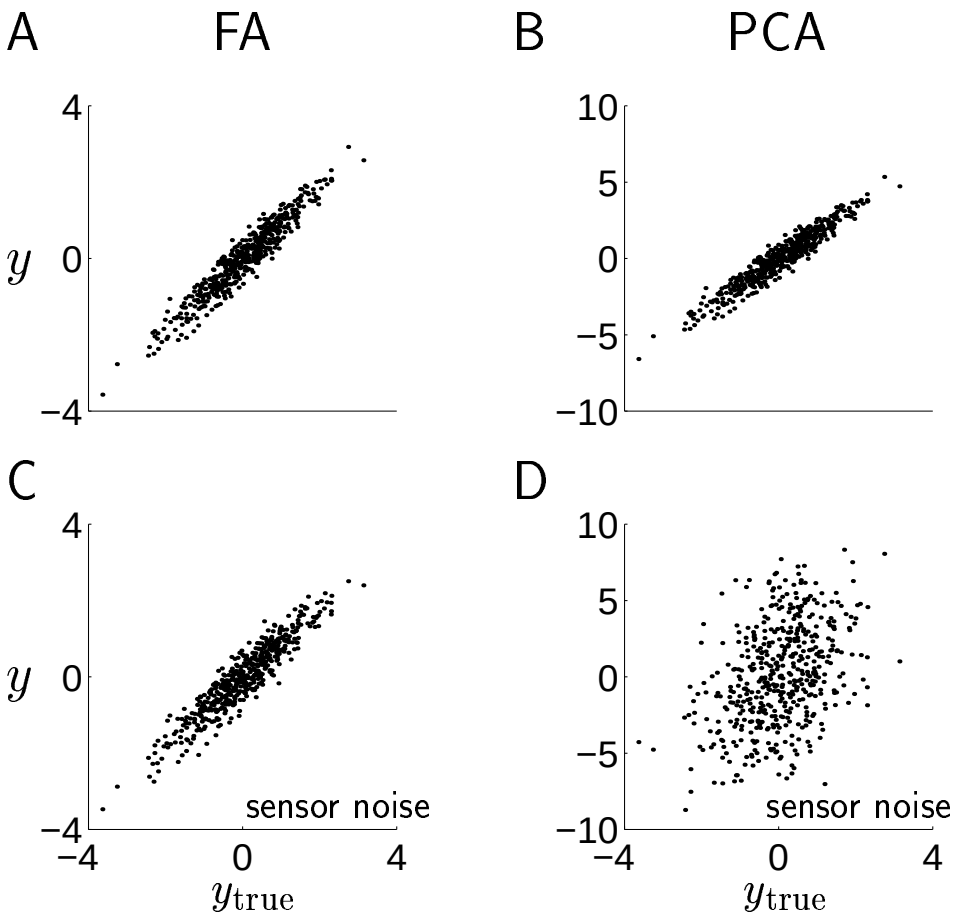
Working through:

$$P[\mathbf{x}|\mathcal{G}] \sim \mathcal{N}\left[\mathbf{0}, \sigma^2\mathbf{G}\mathbf{G}^T + \Psi\right]$$

and standard techniques do ML on Wishart distribution wrt $\mathbf{C}$.

PCA vs FA

$x_i = y + \epsilon_i$: alter $\mathcal{V}(\epsilon_i)$:



FA ignores the extra variance in the component and seeks out shared structure.

Recognition Model

Linear, Gaussian, generative model $\Rightarrow$ closed form analytical model:

$$\mathbf{W}^* = \left(\frac{\mathbf{I}}{\sigma^2} + \mathbf{G}^T\boldsymbol{\Psi}^{-1}\mathbf{G}\right)^{-1}\mathbf{G}^T\boldsymbol{\Psi}^{-1}$$

$$\boldsymbol{\Sigma}^* = \left(\frac{\mathbf{I}}{\sigma^2} + \mathbf{G}^T\boldsymbol{\Psi}^{-1}\mathbf{G}\right)^{-1}$$

weighted average of prior; posterior. $\boldsymbol{\Sigma}^*$ is *independent* of $\mathbf{x}$ – representationally handy.

Note $-\log P[\mathbf{x}^\mu, \mathbf{y}; \mathcal{G}] =$

$$\tfrac{1}{2\sigma^2}|\mathbf{y}|^2 + \tfrac{1}{2}(\mathbf{x}^\mu - \mathbf{G}\mathbf{y})^T\boldsymbol{\Psi}^{-1}(\mathbf{x}^\mu - \mathbf{G}\mathbf{y}) +$$

$$\tfrac{1}{2}\left(\log\sigma^2 + \textstyle\sum_{i=1}^n \log\tau_i^2\right) + \mathcal{K}$$

prior; reconstruction terms.

rotational invariance:

$$\mathbf{G}\mathbf{y} = (\mathbf{G}\mathbf{U})(\mathbf{U}^T\mathbf{y})$$

$$\langle(\mathbf{U}^T\mathbf{y})(\mathbf{U}^T\mathbf{y})^T\rangle = \mathbf{U}^T\langle\mathbf{y}\mathbf{y}^T\rangle\mathbf{U} = \mathbf{U}^T\mathbf{U} = \mathbf{I}$$

PCA

Either $\Psi = \tau^2 \mathbf{I}$ which forces rotational symmetry in $\mathbf{x}$.

Or deterministic:

$$\mathcal{Q}[\mathbf{y}; \mathbf{x}^\mu, \mathbf{w}] = \delta(y(\mathbf{x}^\mu) - \mathbf{w}\mathbf{x}^\mu)$$

which makes

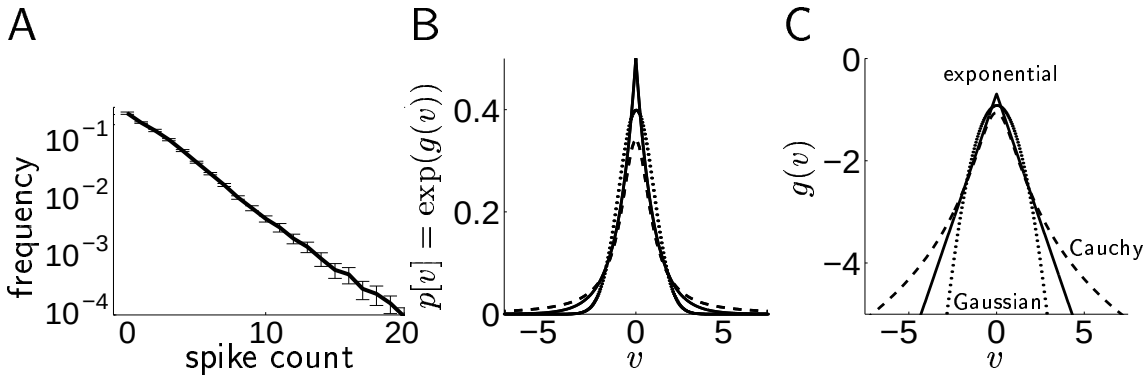
$$\begin{aligned} \mathcal{F}[\mathbf{g}, \mathbf{w}] &\equiv -\sum_{\mu=1}^{M} \log P[\mathbf{x}^\mu, y(\mathbf{x}^\mu); \mathcal{G}] \\ &= \mathcal{K} + \frac{1}{2\tau^2} \sum_{\mu=1}^{M} |\mathbf{x}^\mu - \mathbf{g} y(\mathbf{x}^\mu)|^2 \end{aligned}$$

as before.

$\times$ only second order statistics

Clustering

Unlike FA/PCA, cortical representations are *sparse:*



MoG is ultimately sparse: $P[\mathbf{y}] = 0$ unless only $y_a = 1$, and

$$\begin{aligned} P[y_a = 1] &= \pi_a \\ P[\mathbf{x}|y_a = 1; \mathcal{G}] &\sim \mathcal{N}[\mathbf{g}^a, \mathbf{I}] \end{aligned}$$

EM for MoG

Tractable E phase: $P[y_a{=}1|\mathbf{x}^\mu;\mathcal{G}] =$

$$\frac{\pi_a P[\mathbf{x}^\mu|y_a{=}1;\mathbf{G}]}{\sum_{a'=1}^{n_y}\pi_{a'}P[\mathbf{x}|y_{a'}{=}1;\mathbf{G}]} = \frac{\pi_a e^{-|\mathbf{x}^\mu-\mathbf{g}^a|^2/2}}{\sum_{a'=1}^{n_y}\pi_{a'}e^{-|\mathbf{x}^\mu-\mathbf{g}^{a'}|^2/2}}$$

and then

$$\mathcal{F}[\mathbf{x}^\mu,\mathcal{Q};\mathcal{G}] = \sum_{a=1}^{n_y} P[y_a=1|\mathbf{x}^\mu;\mathcal{G}]\left(\frac{1}{2}|\mathbf{x}^\mu-\mathbf{g}^a|^2 - \log\pi_a\right).$$

Stochastic grad desc M phase:

$$\begin{aligned} g_b^a(t{+}1) &= g_b^a(t) - \epsilon\nabla_{g_b^a}\mathcal{F}[\mathbf{x}^\mu,\mathcal{Q};\mathcal{G}(t)] \\ &= g_b^a(t) + \epsilon P[y_a=1|\mathbf{x}^\mu;\mathcal{G}(t)]\,[\mathbf{x}^\mu-\mathbf{g}^a(t)]_b \\ \rho_a(t{+}1) &= \rho_a(t) + \epsilon(P[y_a=1|\mathbf{x}^\mu;\mathcal{G}]-\pi_a) \end{aligned}$$

where $\pi_a = \frac{e^{\rho_a}}{\sum_{a'}e^{\rho_{a'}}}$. *Delta* rules.

(elastic net is this, plus a prior over $\mathbf{g}^a$)

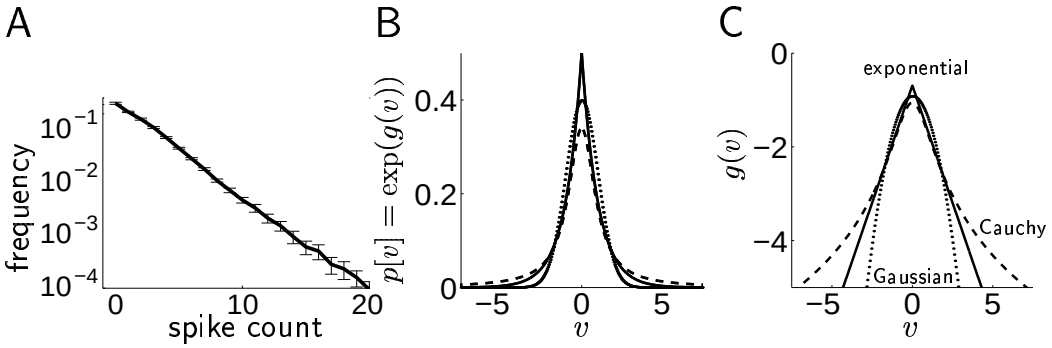
Nonlinear FA

Encourage sparse representation through a prior over $\mathbf{y}$:

$$P[\mathbf{y};\mathcal{G}] = \prod_a e^{-f(y_a)}$$

where $f(y)$ is a function such as

$$\alpha \log(y_0^2 + y^2)$$



Exact analysis is intractable; O&F used deterministic:

$$\mathcal{Q}[\mathbf{y};\mathbf{x}^\mu,\tilde{\mathbf{y}}] = \delta(\mathbf{y}-\tilde{\mathbf{y}})$$

where $\tilde{\mathbf{y}}$ comes from minimising:

$$\mathcal{F}[\mathbf{x}^\mu,\mathcal{Q};\mathcal{G}] = \frac{1}{\tau^2}\left|\mathbf{x}-\mathbf{G}\tilde{\mathbf{y}}\right|^2 + \sum_{a=1}^{n_y} f(\tilde{y}_a)$$

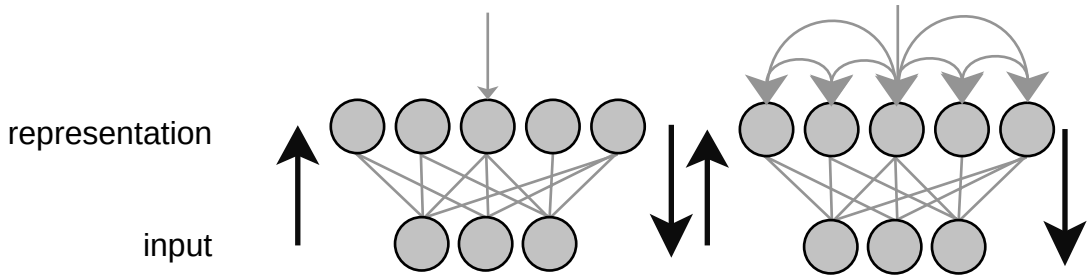
Recognition

This gives:

$$\begin{aligned}\dot{\tilde{\mathbf{y}}} &= -\epsilon \nabla_{\tilde{\mathbf{y}}} \mathcal{F}[\mathbf{x}^\mu, \tilde{\mathbf{y}}(\tau); \mathcal{G}] \\ &= \epsilon \left[\mathbf{G}^T \left(\mathbf{x}^\mu - \mathbf{G}\tilde{\mathbf{y}}(\tau) \right) - \mathbf{f}'(\tilde{\mathbf{y}}(\tau)) \right] \\ &= \epsilon \mathbf{G}^T \mathbf{x}^\mu - \epsilon \left(\mathbf{G}^T \mathbf{G} \tilde{\mathbf{y}}(\tau) + \mathbf{f}'(\tilde{\mathbf{y}}(\tau)) \right)\end{aligned}$$

(non-convex; so not unique)

Two different architectures:



with roles for recurrent weights.

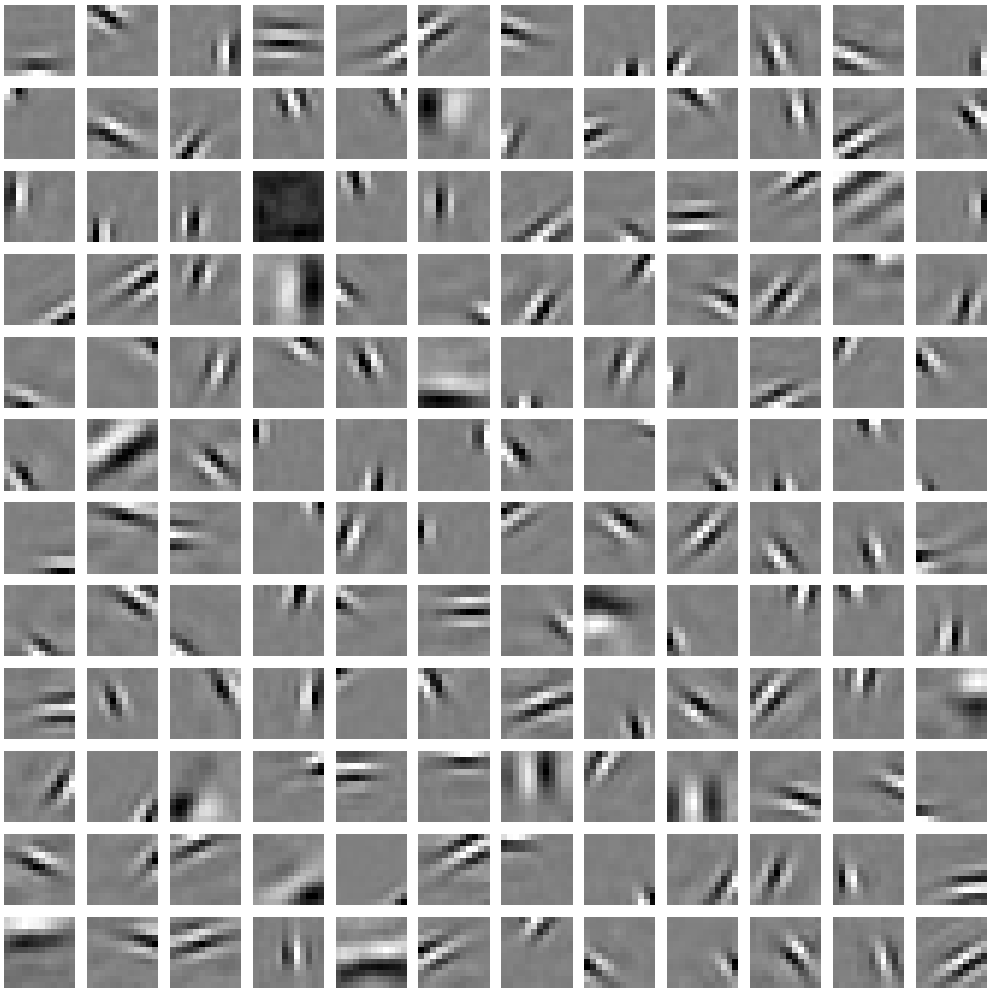
Learning rule for $\mathbf{G}$:

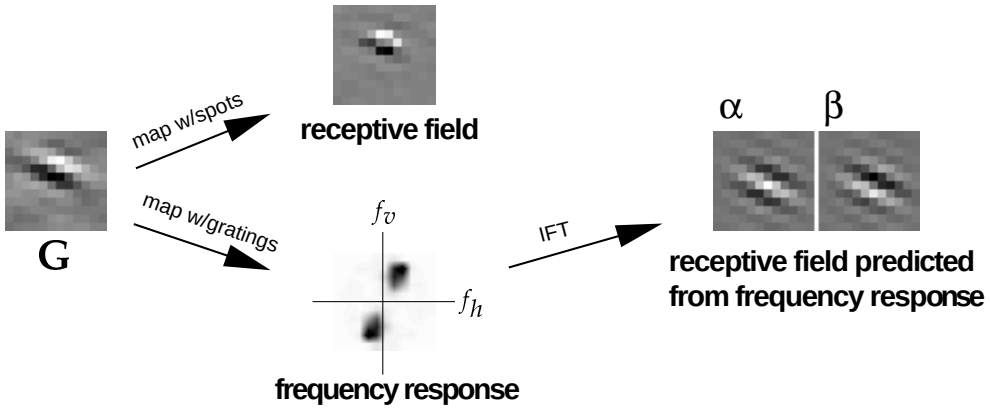
$$\begin{aligned}\mathbf{G}_{ba}(t+1) &= \mathbf{G}_{ba}(t) - \epsilon \nabla_{\mathbf{G}_{ba}} \mathcal{F}[\mathbf{x}^{\mu}, \tilde{\mathbf{y}}^{*}; \mathbf{G}(t)] \\ &= \mathbf{G}_{ba}(t) + \epsilon \left[\mathbf{x}^{\mu} - \mathbf{G}(t)\tilde{\mathbf{y}}^{*}\right]_{b} \tilde{y}^{*}_{a}.\end{aligned}$$

normalisation constraints so $|\mathbf{G}| \not\to \infty$.

- prior of independence and sparsity controls learning – PCA would give non-localised patches
- *posterior* over $\mathbf{y}$ is a form of distributed population code for scenes – with mutual dependence we study below
- therefore, prior interferes with natural requirement for posterior – unfortunate
- scenes aren't really caused by Gabor patches – need to build a hierarchical version of the model

Olshausen and Field





ICA

Linear and non-linear FA have *independent* factor prior:

$$P[\mathbf{x}, \mathbf{y}; \mathcal{G}] \propto \left[\prod_{a=1}^{M} e^{-f_a(y_a)} \right] e^{-|\mathbf{x}-\mathbf{G}\mathbf{y}|^2/2\tau^2}$$

do:

- $\mathbf{G}$ invertible with $\mathbf{W} = \mathbf{G}^{-1}$ and $M = N$
- $\tau^2 = 0$
- $P_a[y_a; \mathcal{G}] = p_a(y_a) = e^{-f_a(y_a)}$

Then since $\int_x \delta(x - gy)p(y)dy = \frac{1}{g}p\left(\frac{x}{g}\right)$:

$$\begin{aligned}
\log P[\mathbf{x}|\mathcal{G}] &= \log \int_{\mathbf{y}} \left[\prod_{a=1}^{m} p_a(y_a) \right] \delta(\mathbf{x} - \mathbf{G}\mathbf{y}) d\mathbf{y} \\
&= \log \frac{1}{|\mathbf{G}|} \prod_{a=1}^{m} p_a\left([\mathbf{W}\mathbf{x}]_a\right) \\
&= \log |\mathbf{W}| + \sum_{a=1}^{m} \log p_a\left([\mathbf{W}\mathbf{x}]_a\right)
\end{aligned}$$

where $|\mathbf{W}| = 1/|\mathbf{G}|$ is the determinant of $\mathbf{W}$.

ICA

Do direct gradient ascent in $\mathbf{W}$:

$$\begin{aligned}\Delta \mathbf{W}_{ab} &\propto \frac{\partial}{\partial \mathbf{W}_{ab}} \log P[\mathbf{x}|\mathbf{W}^{-1}] \\ &= \left[\mathbf{W}^T\right]^{-1}_{ab} + z_a x_b\end{aligned}$$

where $z_a \equiv \frac{\partial}{\partial y_a} \log p_a(y_a)$.

If $p_a(y_a) \propto 1/\cosh(y_a)$, then $z_a = -\tanh(y_a)$ then

$$\Delta \mathbf{W}_{ab} \propto \left[\mathbf{W}^T\right]^{-1}_{ab} - \tanh(y_a) x_b$$

is anti-weight decay and anti-Hebb.

Multiply by $\mathbf{W}^T\mathbf{W}$:

$$\Delta \mathbf{W}_{ab} \mathbf{W}^T\mathbf{W} \propto \mathbf{W}_{ab} - \tanh(y_a) \left[\mathbf{y}^T\mathbf{W}\right]_b$$

leads to very similar results as O&F.

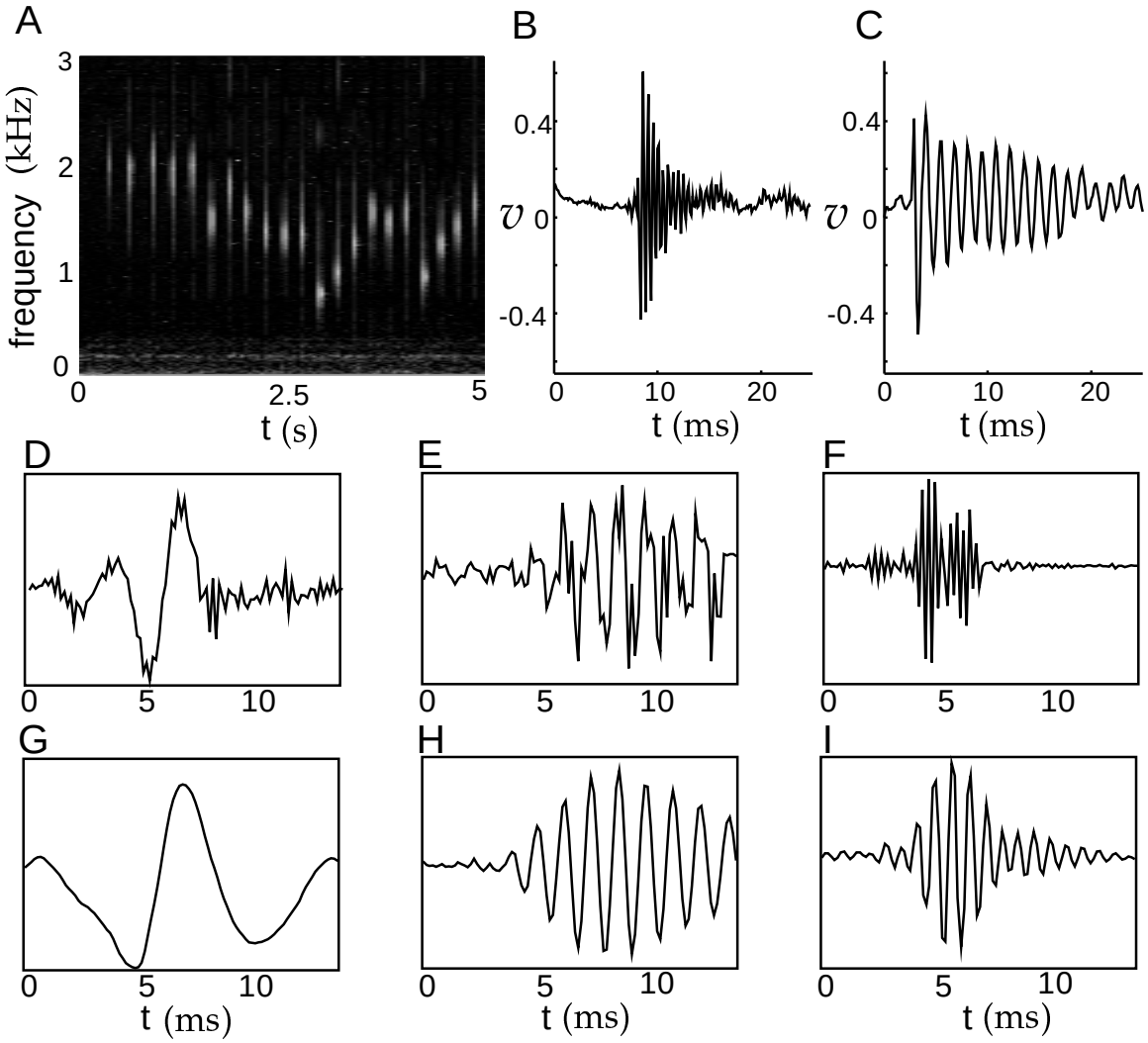
ICA

- independent *sources* underlie $\mathbf{x}$
- through a linear mixing process
- unscramble by searching for statistical independence
- for temporal sources, use fourier components
- not clear how to make hierarchical – too strong an initial constraint?
- equivalent to infomax in a deterministic network:
  - $p_{\mathbf{Y}}(\mathbf{y}) = \frac{p_{\mathbf{X}}(\mathbf{f}^{-1}(\mathbf{y}))}{|\mathbf{J}(\mathbf{f}^{-1}(\mathbf{y}))|}$
  - $\mathbf{J}(\mathbf{x})_{ab} = \partial y_a(\mathbf{x})/\partial x_b$ with $|\mathbf{J}(\mathbf{x})| = |\mathbf{W}| \prod_{a=1}^{m} |y'_a(\mathbf{W}\mathbf{x})|$
  - infomax requires $\text{argmax}_{\mathbf{W}} \mathcal{H}[\mathbf{y}]$ and

    $\langle \log |\mathbf{J}(\mathbf{x})| \rangle_{P_I[\mathbf{x}]} = \log |\mathbf{W}| + \langle \log |y'_a(\mathbf{W}\mathbf{x})| \rangle$
  - so if $y'_a(u) = p_a(u)$ then same as before

ICA Applied to Sounds

Tooth-tapping:



Independent components of tooth-tapping sounds. B;C) waveforms of taps for high and low frequency notes respectively. Three independent components calculated on the basis of 100 samples (1/80s); receptive fields $\mathbf{W}_{i*}$ for three output units. D is reported as being sensitive to the sound of the air-conditioner; E;F extract tooth taps of two different frequencies. G;H;I) are the associated rows of $\mathbf{G} = \mathbf{W}^{-1}$, showing what activity in the input is caused by activity in D;E;F. Adapted from Bell & Sejnowski (1996).

Multiresolution Analysis

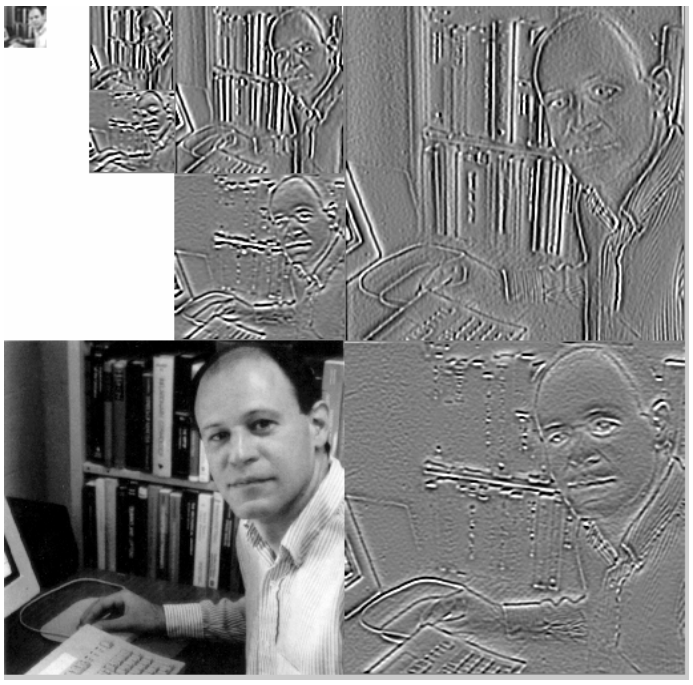
Output of sparse coding, ICA, Hebbian models are reminiscent of multiresolution, wavelet analysis:

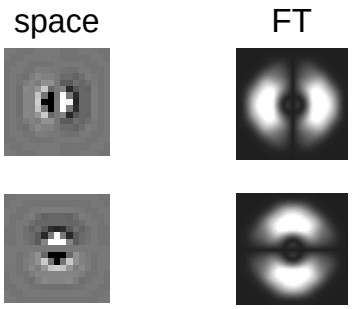
- systematic decompositions with
- populations of units
- tuned to different spatial frequencies
- distributed across space in a
- $\sim$ translation invariant way

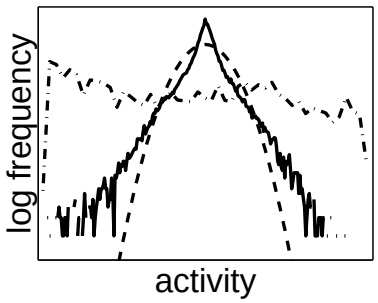
cf wavelets, pyramid codes, quadrature mirror filters, *etc*

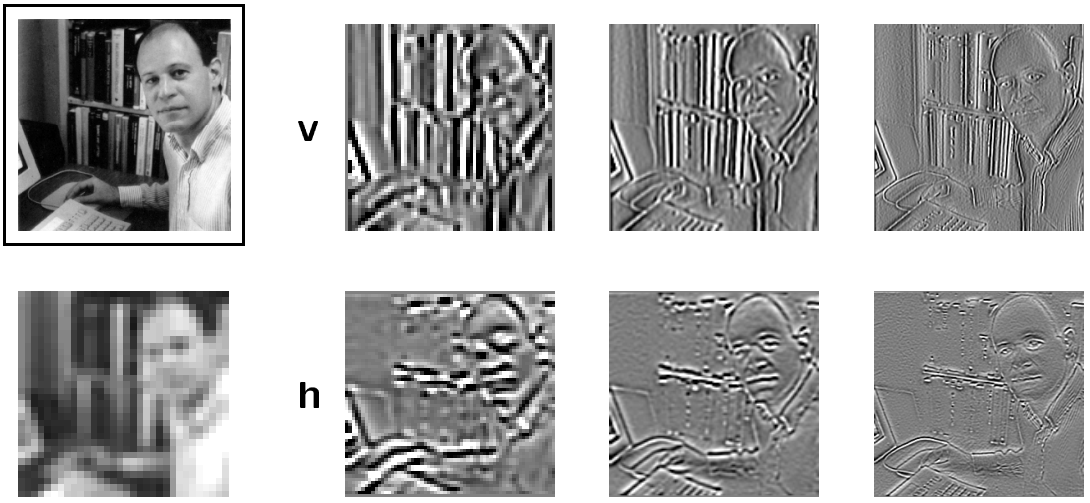
evidence from adaptation studies

Wavelet Decomposition





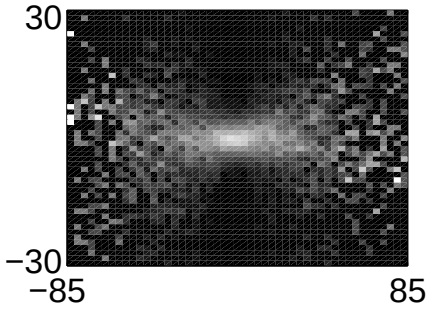


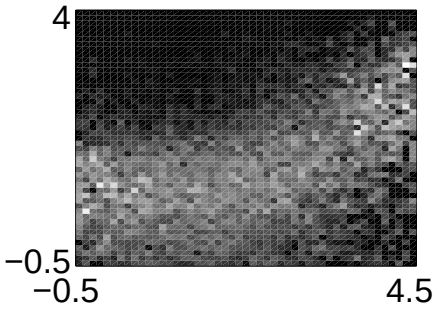


Wavelet Decomposition

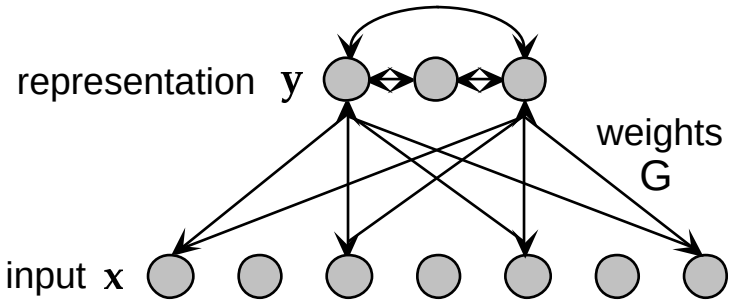
Write $\mathbf{y} = \mathbf{H}\mathbf{x}$; filters in *rows* of $\mathbf{H}$

- $\mathbf{H}$ square, $\mathbf{H}\mathbf{H}^T = \mathbf{I}$, then $\mathbf{x} = \mathbf{H}^T\mathbf{y}$ is self-inverting − tight frame if $\mathbf{H}$ not square; $\mathbf{H}^T\mathbf{H} = \mathbf{I}$
- *sparse* − *eg* double exponential
- good lossless image *codes*
- and lossy − since can zero coefficients near to 0 and send high frequency coefficients more coarsely
- remaining non-independence (QMFs) $\mathcal{E}[C|P] = 0$, but $\mathcal{E}[\log|C||\log|P|] \neq 0$ − *coordination* across different spatial scales used in zerotree codes; $\times$'ly normalize





The Boltzmann Machine



$\mathbf{z} = \{\mathbf{x}, \mathbf{y}\}$: derived from the Hopfield net
$\mathcal{G}_{ij} \sim$ readiness for $z_i = z_j$

Energy $E(\mathbf{z}, \mathbf{G}) = -0.5\mathbf{z}^T\mathbf{G}\mathbf{z}$ with
$\mathbf{G}_{ij} = \mathbf{G}_{ji}, \mathbf{G}_{ii} = 0$. Boltzmann distribution:

$$P[\mathbf{z}; \mathbf{G}] = \frac{e^{-E(\mathbf{z})/T}}{\sum_{\mathbf{z}'} e^{-E(\mathbf{z}')/T}}$$

where T is a computational 'temperature'.

Can't tractably *calculate* $\sum_{\mathbf{z}'} e^{-E(\mathbf{z}')/T}$, but can define a Markov chain whose stationary distribution is $P[\mathbf{z}; \mathbf{G}]$.

If $\mathbf{x}$ is clamped:

$$P[\mathbf{y}|\mathbf{x}; \mathbf{G}] = \frac{e^{-E(\mathbf{x},\mathbf{y})/T}}{\sum_{\mathbf{y}'} e^{-E(\mathbf{x},\mathbf{y}')/T}}$$

Learning Rules

Since:

$$\nabla_{\mathbf{G}_{ij}} P[\mathbf{z};\mathbf{G}] = P[\mathbf{z};\mathbf{G}]\left(z_i z_j - \sum_{\mathbf{z}'} P[\mathbf{z}';\mathbf{G}] z'_i z'_j\right)$$

we get $\nabla_{\mathbf{G}_{ij}} \log P[\mathbf{x}^\mu;\mathbf{G}] =$

$$\begin{aligned}\frac{\nabla_{\mathbf{G}_{ij}} P[\mathbf{x}^\mu;\mathbf{G}]}{\log P[\mathbf{x}^\mu;\mathbf{G}]} &= \frac{\sum_{\mathbf{y}} \nabla_{\mathbf{G}_{ij}} P[\mathbf{z}^\mu;\mathbf{G}]}{P[\mathbf{x}^\mu;\mathbf{G}]} \\ &= \sum_{\mathbf{y}} P[\mathbf{y}|\mathbf{x}^\mu;\mathbf{G}] z_i^\mu z_j^\mu - \sum_{\mathbf{z}'} P[\mathbf{z}';\mathbf{G}] z'_i z'_j\end{aligned}$$

two averages.

Generate *samples*:

wake clamp $\mathbf{x}^\mu$ sample $\mathbf{y}|\mathbf{x}^\mu$ and approximate:

$$\sum_{\mathbf{y}} P[\mathbf{y}|\mathbf{x}^\mu;\mathbf{G}] z_i^\mu z_j^\mu \sim \frac{1}{K}\sum_{\nu=1}^{K} z_i^{\mu,\nu} z_j^{\mu,\nu} \equiv \langle z_i^\mu z_j^\mu \rangle_+.$$

sleep sample $\mathbf{z}'$ from unconditional distribution and use

$$\sum_{\mathbf{z}'} P[\mathbf{z}';\mathbf{G}] z'_i z'_j \sim \frac{1}{K}\sum_{\nu=1}^{K} z_i'^{\nu} z_j'^{\nu} \equiv \langle z'_i z'_j \rangle_-.$$

Gibbs Sampling

If $\mathcal{T}_{\mathbf{z}\mathbf{z}'}$ is the transition matrix of a Markov chain such that *detailed balance*:

$$P[\mathbf{z}]\mathcal{T}_{\mathbf{z}\mathbf{z}'} = P[\mathbf{z}']\mathcal{T}_{\mathbf{z}'\mathbf{z}}$$

holds for $P[\mathbf{z}]$, then application of $\mathcal{T}$ to any starting $\mathbf{z}(0)$ will lead ultimately to unbiased sampling from $P[\mathbf{z}]$.

If ΔE_k is the *energy gap:*

$$\Delta E_k = E(\mathbf{z})|_{z_k=1} - E(\mathbf{z})|_{z_k=0} = -\sum_j \mathcal{G}_{kj} z_j$$

between unit k being on or off, then set $z_k = 1$ with probability:

$$p_k = \sigma(-\Delta E_k / T)$$

forces detailed balance with $P[\mathbf{z}]$.

This is just the probabilistic version of the Hopfield rule.

Mean Field Methods

Consider approximate

$$\mathcal{Q}[\mathbf{y};\mathbf{q}] \sim P[\mathbf{y}|\mathbf{x}^\mu;\mathcal{G}] = \frac{1}{Z}e^{-E(\mathbf{z})}$$

from which we can easily calculate

$$\sum_{\mathbf{y}} \mathcal{Q}[\mathbf{y};\mathbf{q}] z_i^\mu z_j^\mu = \mathcal{E}_\mathcal{Q}[z_i^\mu z_j^\mu]$$

Approximate using *factorised:*

$$\mathcal{Q}[\mathbf{y};\mathbf{q}] = \prod_k q_k^{y_k}(1-q_k)^{1-y_k}$$

and determine *variational* parameters $\mathbf{q}$ by minimising

$$\begin{aligned} KL[\mathcal{Q},P] &= \sum_{\mathbf{y}} \mathcal{Q}[\mathbf{y};\mathbf{q}] \log \mathcal{Q}[\mathbf{y};\mathbf{q}]/P[\mathbf{y}|\mathbf{x};\mathcal{G}] \\ &= \mathcal{E}_\mathcal{Q}[E(\mathbf{z})] - \sum_k \mathcal{H}[q_k] + \log Z \\ &= -\frac{1}{2}\sum_{ij} q_i \mathcal{G}_{ij} q_j - \sum_k \mathcal{H}[q_k] + \log Z \end{aligned}$$

which is the same as maximising the free energy $\mathcal{F}$ with respect to $\mathcal{Q}$

Mean Field Methods

Consider coordinate-wise descent for q_i:

$$q_i' = \sigma(\sum_j \mathcal{G}_{ij} q_j)$$

is just a deterministic version of stochastic Gibbs sampling.

Compute $\mathcal{Q}_i[\mathbf{x}, \mathbf{y}] \sim P[\mathbf{x}, \mathbf{y}; \mathcal{G}]$ similarly for the negative phase, and use

$$\Delta \mathcal{G}_{ij} \propto \mathcal{E}_{\mathcal{Q}_+}[z_i^\mu z_j^\mu] - \mathcal{E}_{\mathcal{Q}_-}[z_i z_j]$$

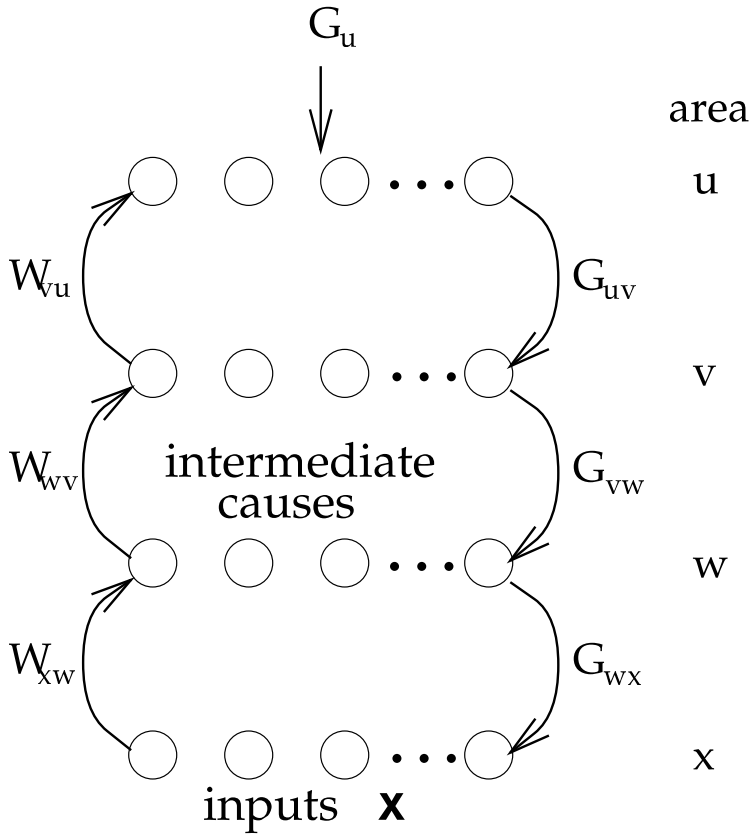
the trouble is that the mean field distribution is a poor approximation for the unconstrained negative phase

- MF is a general strategy
- can look for *upper* bounds on the log probability as well as lower bounds
- can be *exact* in certain limits
- can mix exact and MF methods together

The Boltzmann Machine

- learning during wake; unlearning during sleep
- learning stops when network's distribution over $\mathbf{x}$ equals world's distribution over $\mathbf{x}$
- works for arbitrary architectures; can learn anything that architecture can express
- relax symmetry constraint using weight decay
- stochastic recognition model
- first application of ML density estimation
- stochasticity implemented by spikes?
- Gibbs sampling is awful
- learning is based on difference of two noisy quantities
- can use *mean field* methods instead.

The Helmholtz Machine

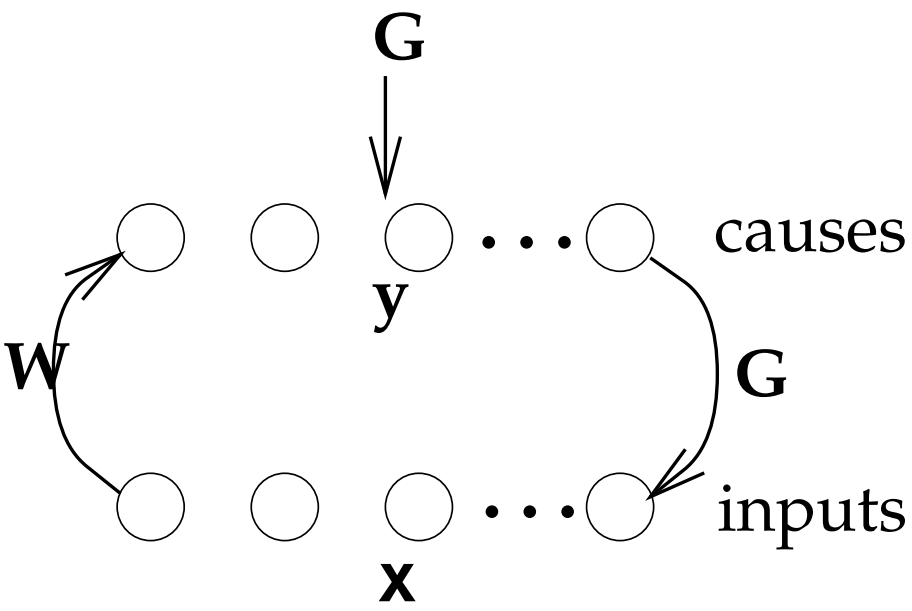


- hierarchical feedback generative sigmoid model $\mathbf{G}$
- hierarchical feedforward recognition model $\mathbf{W}$ (responsible for receptive fields)
- learned analysis by learned synthesis
- simple wake-sleep algorithm

Recognition

Use another set of (recognition) parameters $\mathbf{W}$ to produce an estimate of the inverse to the generative distribution:

$$\mathcal{Q}[\mathbf{y}; \mathbf{x}, \mathbf{W}] \simeq P[\mathbf{y}|\mathbf{x}; \mathcal{G}],$$



and make $\mathcal{Q}$ unreasonably simple.

The Binary HM

Consider the simple Bernoulli model:

$$P[\mathbf{y};\mathcal{G}] = \prod_a [\sigma(g_a^y)]^{y_a} [1-\sigma(g_a^y)]^{1-y_a}$$

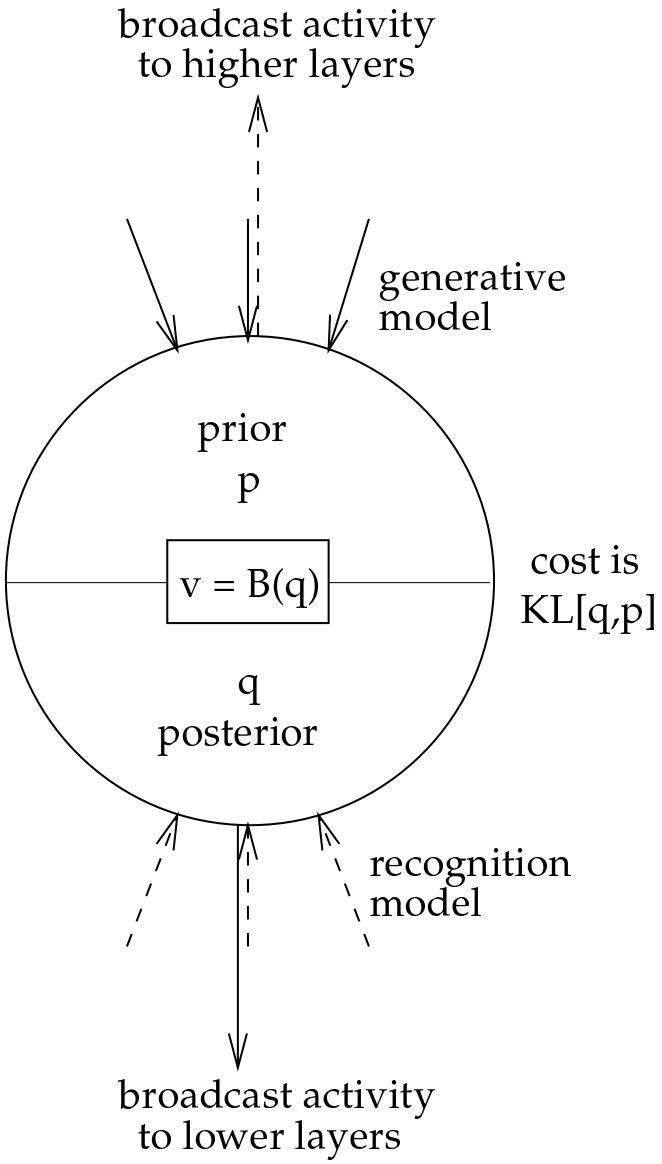
$$P[\mathbf{x}|\mathbf{y};\mathcal{G}] = \prod_b [\sigma([\mathbf{G}\mathbf{y}]_b)]^{x_b} [1-\sigma([\mathbf{G}\mathbf{y}]_b)]^{1-x_b}.$$

and use the same for the recognition model:

$\mathbf{y}$	probability		
$\{0,0,0\}$	$(1-q_1)$	$(1-q_2)$	$(1-q_3)$
$\{0,0,1\}$	$(1-q_1)$	$(1-q_2)$	q_3
$\{0,1,0\}$	$(1-q_1)$	q_2	$(1-q_3)$
$\{0,1,1\}$	$(1-q_1)$	q_2	q_3
$\{1,0,0\}$	q_1	$(1-q_2)$	$(1-q_3)$
$\{1,0,1\}$	q_1	$(1-q_2)$	q_3
$\{1,1,0\}$	q_1	q_2	$(1-q_3)$
$\{1,1,1\}$	q_1	q_2	q_3

and only need 3 probabilities/activities/firing rates.

A Single Unit in the Binary HM



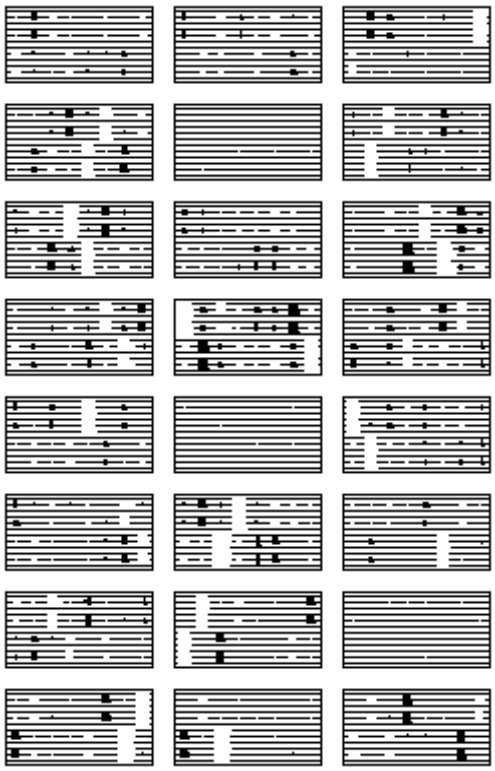
The Shifter Problem

Recognition

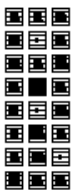
2–3 : 3.7



1–2 : 11.7

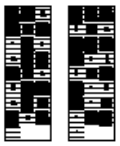


Biases to 2 : 38.4

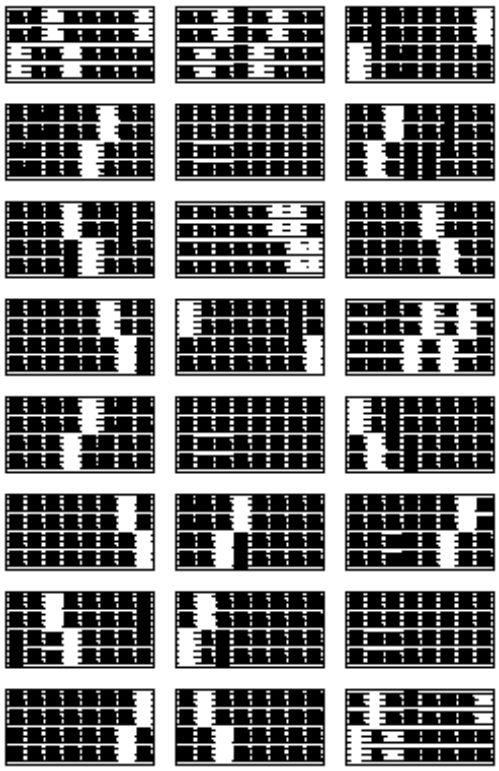


Generative

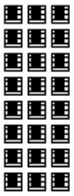
3–2 : 13.3



2–1 : 13.3



Biases to 2 : 3.0



The Wake-Sleep Algorithm

In the given form this requires back-prop. Emulate the BM by searching for a stochastic approximation form:

$$\frac{\partial}{\partial \mathbf{G}_{ij}}\mathcal{F} = \frac{\partial}{\partial \mathbf{G}_{ij}} \sum_{\mathbf{y}} \mathcal{Q}(\mathbf{y};\mathbf{x},\mathbf{W}) \log P[\mathbf{x},\mathbf{y};\mathbf{G}]$$

since $\mathcal{H}[\mathcal{Q}]$ doesn't depend on $\mathbf{G}$.

Here the terms:

$$P[\mathbf{x},\mathbf{y};\mathbf{G}] = \prod \left[\sigma(\textstyle\sum_j s_j \mathbf{G}_{ji})\right]^{s_i} \left[1-\sigma(\textstyle\sum_j s_j \mathbf{G}_{ji})\right]^{1-s_i}$$

and since

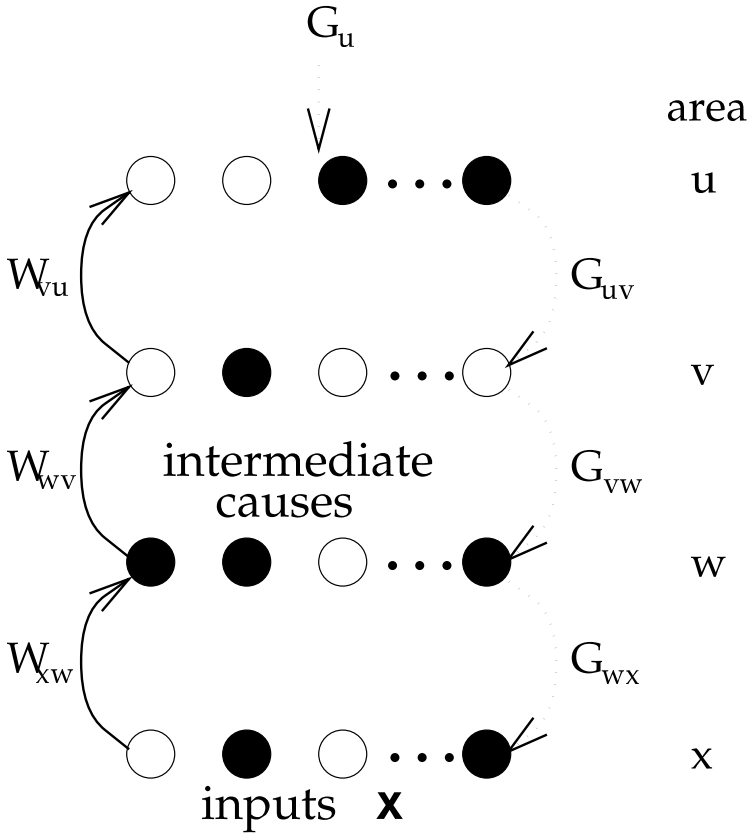
$$\frac{\partial}{\partial y} \log \sigma(y)^x (1-\sigma(y))^{1-x} = x - \sigma(y)$$

we get

$$\frac{\partial}{\partial \mathbf{G}_{ij}}\mathcal{F} = \left\langle \left(s_i - \sigma(\textstyle\sum_j s_j \mathbf{G}_{ji})\right) s_j \right\rangle_+$$

averaging over $\mathcal{Q}$ which is just the delta rule.

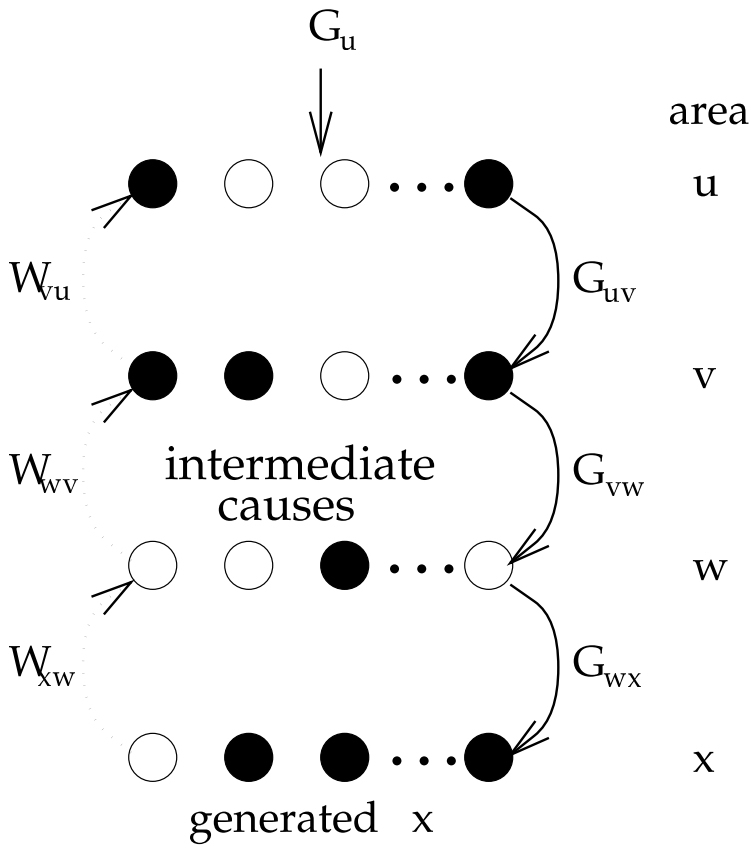
Wake



- Clamp $\mathbf{x}$
- Sample $w_j = \sigma(\sum_i \mathbf{W}^{ij}_{xw} x_i)$
- Train $\Delta \mathbf{G}^{ji}_{wx} \propto w_j \left[x_i - \sigma(\sum_k \mathbf{G}^{ki}_{wx} y_k) \right]$

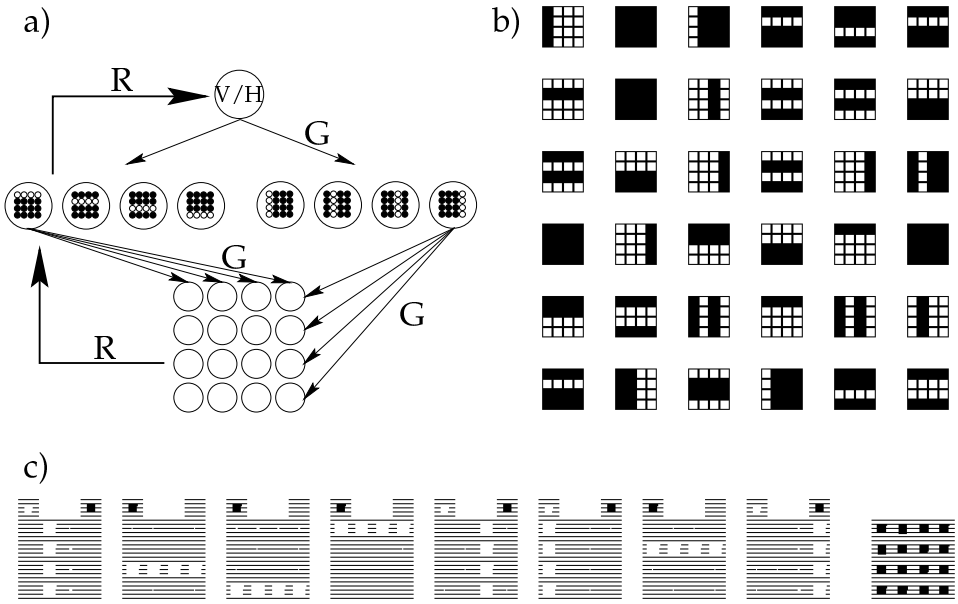
Sleep

Training $\mathcal{R}$ is not so simple, since both terms in $\mathcal{F}$ depend on it. Treat recognition like inverse generation: $KL[P, Q]$ not $KL[Q, P]$



- Sample $\mathbf{u}$
- Sample $x_i = \sigma(\sum_j \mathbf{G}^{ji}_{wx} w_j)$
- Train $\Delta \mathbf{W}^{ij}_{xw} \propto x_i \left[w_j - \sigma(\sum_k \mathbf{W}^{kj}_{xw} x_k) \right]$

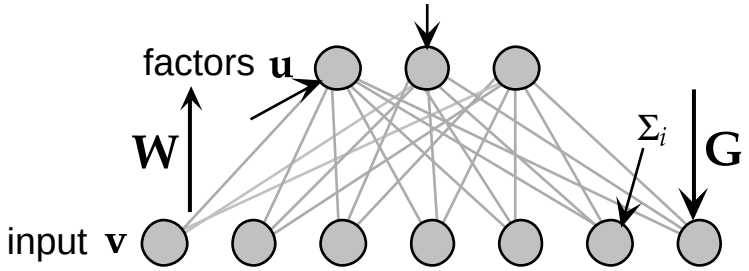
The Bars



The Digits (3hl)



Factor Analysis



Invent recognition weights R:

$$\mathcal{Q}[\mathbf{y};\mathbf{x},\mathbf{W},\Sigma] \sim \mathcal{N}[\mathbf{W}\mathbf{x},\Sigma]$$

Then:

wake observe $\mathbf{x}$, sample $\mathbf{y}$, change:

$$\Delta\mathbf{G}_{ji} \propto y_j \left[x_i - \textstyle\sum_j \mathbf{G}_{ji} y_j\right]$$

$$\Delta\tau_b^2 \propto \tfrac{1}{\tau_b^4}\left([\mathbf{x}^\mu - \mathbf{G}\mathbf{y}]_b^2 - \tau_b^2\right).$$

sleep sample $\mathbf{y}$, sample $\mathbf{x}$, change:

$$\Delta\mathbf{W}_{ij} \propto x_i \left[y_j - \textstyle\sum_i \mathbf{W}_{ij} x_i\right]$$

The delta rule and FA instead of the Hebb rule and PCA.

What Might we Learn?

Insight into:

- statistically appropriate hierarchies
- control mechanisms – activity dependent *vs* independent processes:
  a) intrinsic properties of cells
  b) specific feedforward/feedback connections
  c) intracortical interactions
- experimental manipulations to inputs or cortical activities and their likely structural effects

Autoencoders:

- factorial – forces $\mathbf{y}_i \perp \mathbf{y}_j$ which makes subsequent inference easier
- sparse – encourages independence
- topographic – $\mathbf{y}_i \not\perp \mathbf{y}_j$ forces them to capture related quantities
- non-redundant

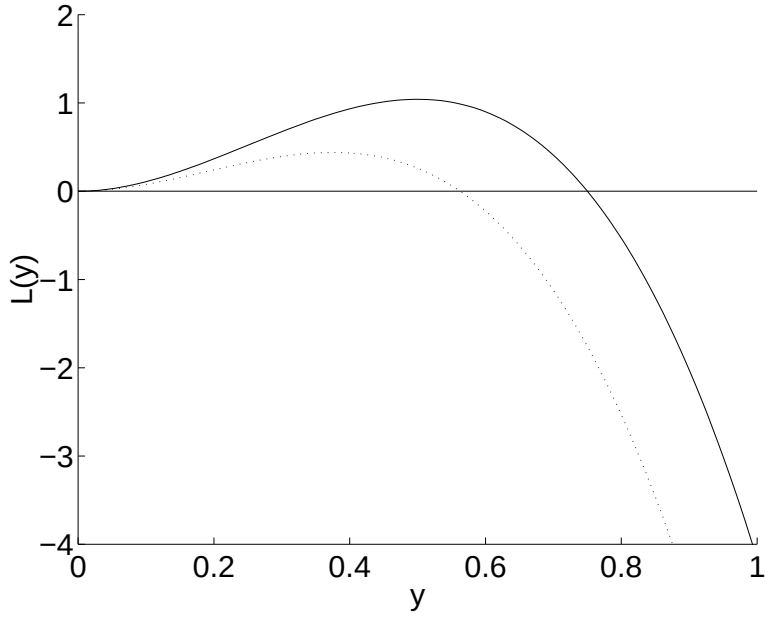
Projection Pursuit

Feature-based models are less holistic

- central limit theory $\rightarrow$ most low dimensional projections $y(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$ of high-dimensional data are Gaussian
- $\Rightarrow$ $\mathbf{w}$ that maximises non-Gaussianity makes y unusually 'interesting'.
- examples are kurtosis, multi-modality, *etc*

Let

$$\theta(\mathbf{w}) = \left\langle y^2(\mathbf{x}) \right\rangle_{P_I[\mathbf{x}]} \qquad L(y(\mathbf{x})) \propto -y^2(\mathbf{x}) \left(\frac{y(\mathbf{x})}{3} - \frac{\theta(\mathbf{w})}{4} \right)$$



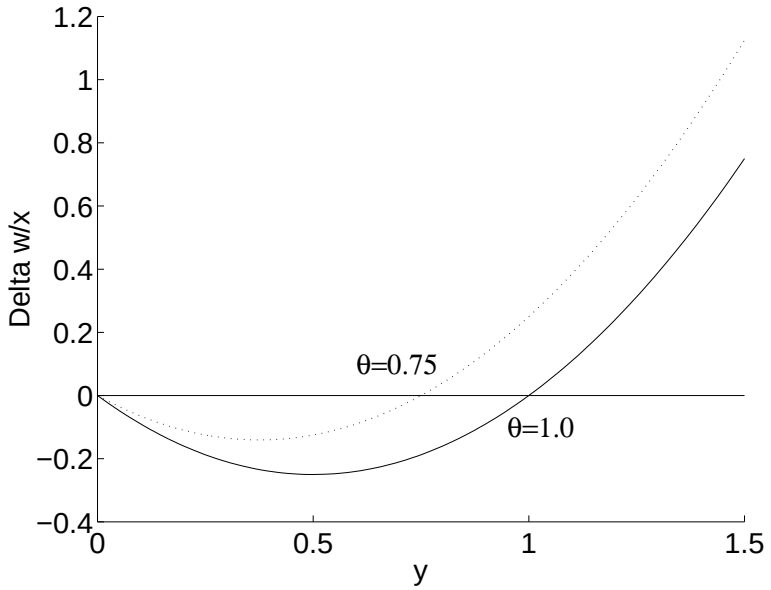
The BCM Rule

Changing weights by gradient descent to minimise loss:

$$\begin{aligned}\Delta w_a &\propto \nabla_{w_a} \left\langle L(y(\mathbf{x})) \right\rangle_{P_I[\mathbf{x}]} \\ &\propto \left\langle y^2(\mathbf{x}) x_a - \theta(\mathbf{w}) y(\mathbf{x}) x_a \right\rangle_{P_I[\mathbf{x}]}\end{aligned}$$

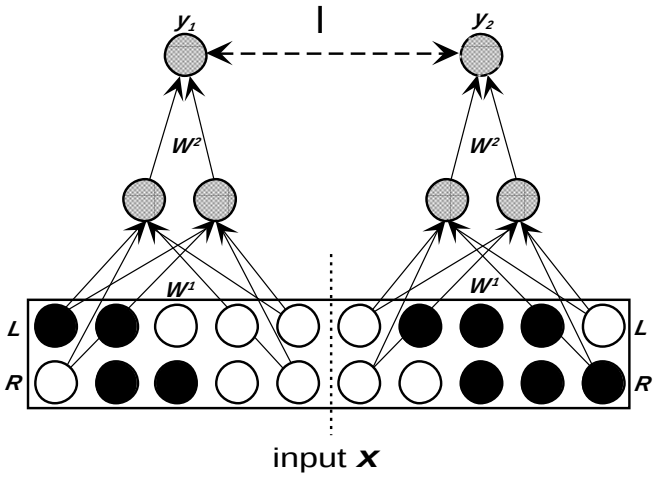
If $\theta(\mathbf{w})$ adapts slowly, use samples $\mathbf{x}^i$ and

$$\Delta w_a \propto \left(y^2(\mathbf{x}^\mu) - \theta(\mathbf{w}) y(\mathbf{x}^\mu) \right) x_a^\mu.$$



Becker & Hinton: IMAX

maximising mutual information between y^1 and y^2:



$\sigma(y^i)$ is *probability* that unit i fires:

$$\mathcal{I}(y_1; y_2) = \mathcal{H}[y_1] + \mathcal{H}[y_2] - \mathcal{H}[y_1; y_2]$$

with $\mathcal{H}[y_1] = -p_1 \log p_1 - p_{\bar{1}} \log p_{\bar{1}};\ p_i = \langle y_i \rangle_{\mathbf{x}}$.

$$\mathcal{H}[y_1; y_2] = -p_{12} \log p_{12} - p_{1\bar{2}} \log p_{1\bar{2}} - \ldots$$

$$\frac{\partial \mathcal{I}(y_1; y_2)}{\partial y_1(\mathbf{x})} = -\left[\log \frac{p_1}{p_{\bar{1}}} - y_2(\mathbf{x}) \log \frac{p_{12}}{p_{\bar{1}2}} - y_{\bar{2}}(\mathbf{x}) \log \frac{p_{1\bar{2}}}{p_{\bar{1}\bar{2}}}\right]$$

increase $\mathcal{H}[y_1]$ or *decrease* $\mathcal{H}[y_1; y_2]$.
Neurobiologically objectionable function.

Representational Learning

ML techniques:

- PCA (approximate)
- FA, mixtures of Gaussians (exact)
- non-linear extensions to FA
- intractability of inverse – mean field, sampling, Helmholtz machine
- modeling, δ-rule based, automatic adjustment to numbers of hidden units

Non-ML techniques:

- interesting projections (non-Gaussian, high variance, *etc*
- information preservation, mutual information maximisation,
- can be Hebbian based — complex adjustment for more hidden units

Generative Models

BM
Boltzmann Machine

SBN
Sigmoid Belief Net

NOTATION
mix = mixture
reg = regularize
switch = switching dynamics
dynam = linear dynamics
distr = distributed hidden state

MVB
MultiVariate Binomial

MMVB
Mixture of MVB's

GAUSS
Gaussian

MOG
Mixture of Gaussians

G-HMM
Gaussian Hidden Markov Model

DS-HMM
Distributed State HMM

FA
Factor Analysis

MFA
Mixture of FA's

FA-HMM
Factor-Analyser HMM

PCA
Principal Components Analysis

MSSM
Mixture of SSM's

SM-SSM
Switching Mixture of SSM's

ICA
Independent Components Analysis

SSM
State-Space Models

NICA
Noisy ICA

HCE
Hierarchical Community of Experts

References

Barlow, HB (1989) Unsupervised learning. *Neural Computation* **1**:295-311.

Becker, S & Plumbley, M (1996) Unsupervised neural network learning procedures for feature extraction and classification. *International Journal of Applied Intelligence* **6**:185-203.

Dayan, P, Hinton, GE, Neal, RM & Zemel, RS (1995) The Helmholtz machine. *Neural Computation* **7**:889-904.

Chapter 10 of Dayan, P & Abbott, LF (2001). *Theoretical Neuroscience.* Cambridge, MA: MIT Press.
`people.brandeis.edu/~abbott/book/`

Hinton, GE (1989) Connectionist learning procedures. *Artificial Intelligence* **40**:185-234.

Hinton, GE, Dayan, P, Frey, BJ & Neal, RM (1995) The wake-sleep algorithm for unsupervised neural networks. *Science* **268**:1158-1160.

MI Jordan, editor (1999). *Learning in Graphical Models.* Cambridge, MA: MIT Press.

Olshausen, BA & Field, DJ (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381**:607-609.

Zemel, RS (1994) *A Minimum Description Length Framework for Unsupervised Learning*. PhD Dissertation, Computer Science, University of Toronto.

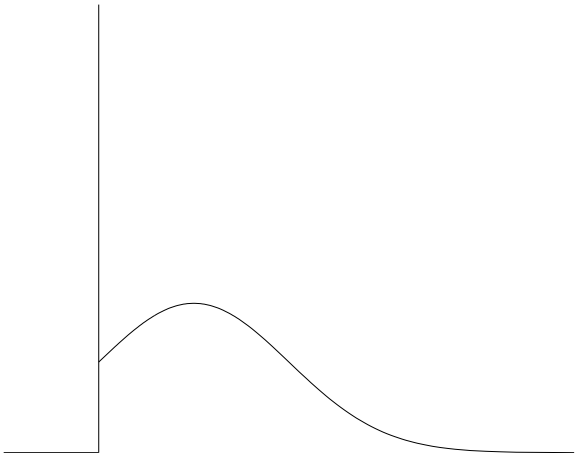
Rectified Gaussian Belief Nets

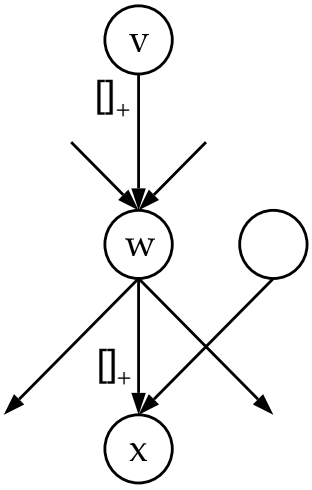
Change the generative model, generate a new Gibbs sampler, generate a new learning algorithm.

Probabilistic activation function for the units:

$$w_i \sim \mathcal{N}[\bar{w}_i \equiv g_0 + \sum_j \mathcal{G}_{ij}[v_j]_+, \sigma_i^2]$$

has two types of variable: unrectified w_i and rectified $[w_i]_+$:



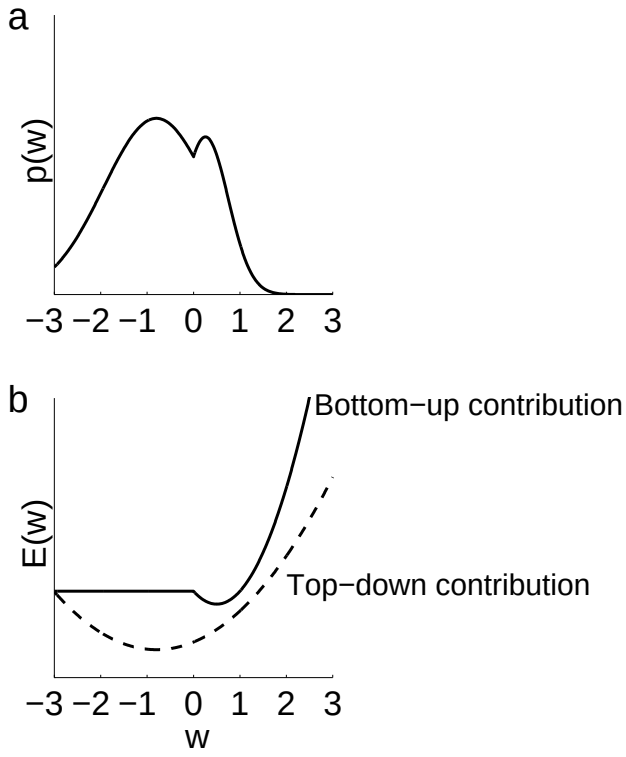


Gibbs Sampling

Consider performing Gibbs sampling to set the state of w_i *given* fixed values of the states of all the other units:

$$E(w_i) = \frac{1}{2\sigma_i^2}(w_i - \bar{w}_i)^2 + \sum_k \frac{1}{2\sigma_k^2}(x_k - \bar{x}_k)^2$$

where $\bar{x}_k$ depends on $[w_i]^+$:

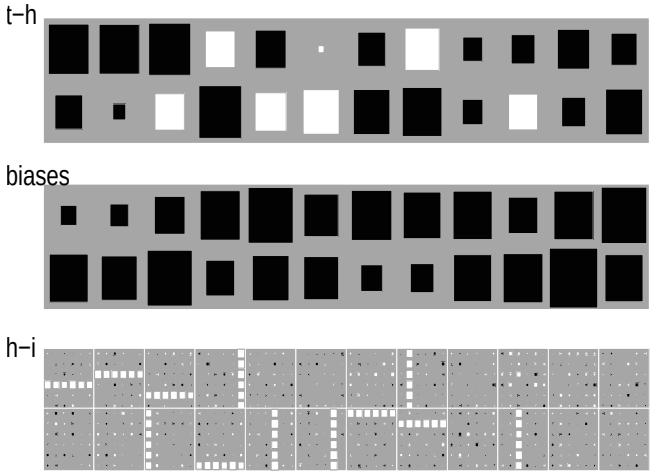


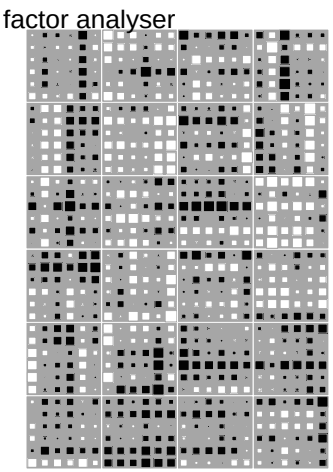
and use *eg* rejection sampling to determine a sample

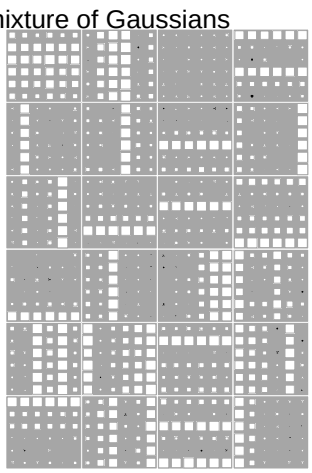
Learning rule for the top-down weights $\mathcal{G}_{ij}$ is the usual delta rule:

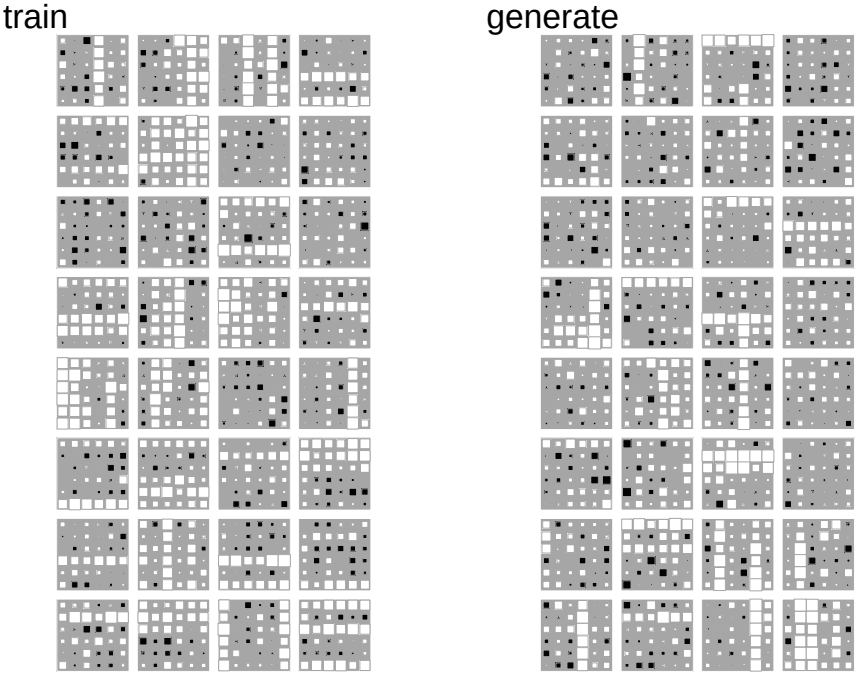
$$\Delta \mathcal{G}_{ij} \propto (w_i - \bar{w}_i)[v_j]_+$$

based on samples at equilibrium









Shifter Problem

