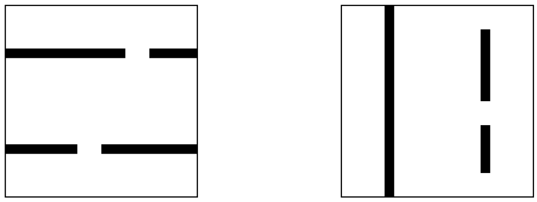
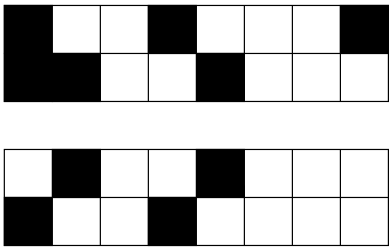
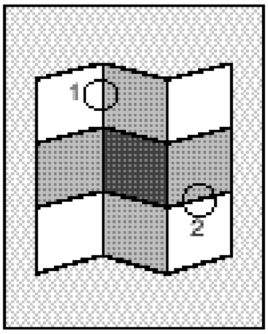
The Helmholtz Machine
The Wake Sleep Algorithm

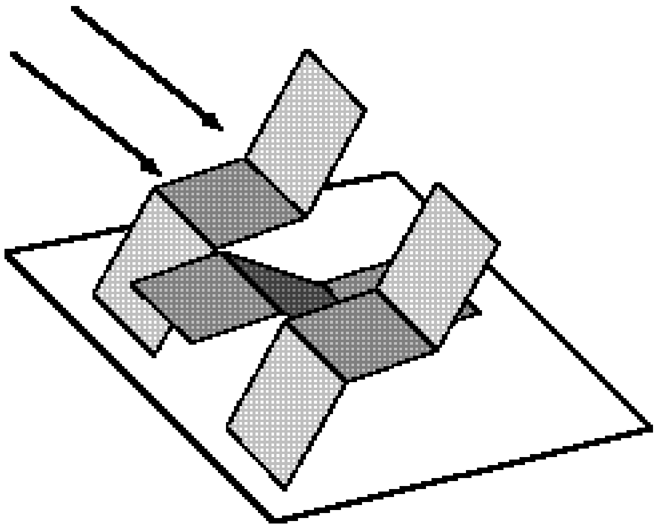
Peter Dayan

Data









Re-representational Learning

Independent inputs $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{N_x}$:

$$\mathcal{P}_I[\mathbf{x}] \sim \frac{1}{N_x} \sum_{i=1}^{N_x} \delta(\mathbf{x} - \mathbf{x}^i)$$

distal **causes** underlie $\mathcal{P}_I[\mathbf{x}]$:

- somewhat directly *detectable*
- generally *useful*

learn collection of causes from $\mathbf{x}^1, \ldots$,
represent new $\mathbf{x}$ in their terms.

Two basic methods:

1. build a *synthetic* model $\mathcal{P}[\mathbf{x}; \mathcal{G}]$ of $\mathcal{P}_I[\mathbf{x}]$
   ML density estimation
2. search $\mathcal{P}_I[\mathbf{x}]$ for particular *features*
   clustering   projection pursuit
   IMAX   invariances

   activity-dependent development

ML Methods

If hidden *causes* $\mathbf{y}$ underlying $\mathbf{x}$ are important:

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

makes

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

prior information in:

structure of the model (table-lookup)

distributions over the parameters $\mathcal{G}$

ML Tasks

a) make $\mathcal{P}[\mathbf{x}^i;\mathcal{G}]$ close to $\mathcal{P}_I[\mathbf{x}^i]$.

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

$$\sim\ \text{argmin}_{\mathcal{G}}\ \sum_i \mathcal{P}_I[\mathbf{x}^i] \log \frac{\mathcal{P}_I[\mathbf{x}^i]}{\mathcal{P}[\mathbf{x}^i;\mathcal{G}]}$$

is the same as

$$\text{argmax}_{\mathcal{G}}\ \prod_i \mathcal{P}[\mathbf{x}^i;\mathcal{G}]$$

b) Represent $\mathbf{x}^i$ in terms of causes:

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

analysis by synthesis

Generalising E & M

Posterior $\mathcal{P}[\mathbf{y}|\mathbf{x}^i;\mathcal{G}]$ can be computationally intractable – use cheaper alternative $\mathcal{Q}[\mathbf{y};\mathbf{x}^i]$:

Jordan's lemma:

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

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

$$\mathcal{Q}[\mathbf{y};\mathbf{x}^i] = \mathcal{P}[\mathbf{y}|\mathbf{x}^i;\mathcal{G}]$$

so (Neal & Hinton) minimise $\mathcal{F}[\mathbf{x}^i,\mathcal{Q};\mathcal{G}]$:

E phase   minimise wrt $\mathcal{Q}$
M phase   minimise wrt $\mathcal{G}$

Factor Analysis

Two level generative model:

$$\begin{aligned} \mathbf{y} &\sim \mathcal{N}[\mathbf{0}, \sigma^2 \mathsf{I}] & \mathcal{P}[\mathbf{y}; \mathcal{G}] &\propto e^{-|\mathbf{y}|^2/2\sigma^2} \\ \mathbf{x}|\mathbf{y} &\sim \mathcal{N}[\mathsf{G}\mathbf{y}, \tau^2 \mathsf{I}] & \mathcal{P}[\mathbf{x}|\mathbf{y}; \mathcal{G}] &\propto e^{-|\mathbf{x}-\mathsf{G}\mathbf{y}|^2/2\tau^2} \end{aligned}$$

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

or diag(τ_i^2) rather than τ^2I.

and $\mathcal{P}[\mathbf{y}|\mathbf{x}; \mathcal{G}] \sim \mathcal{N}[\mathsf{W}^*\mathbf{x}, \Sigma^*]$ where

$$\begin{aligned} \mathsf{W}^* &= \left(\frac{\mathsf{I}}{\sigma^2} + \frac{\mathsf{G}^T\mathsf{G}}{\tau^2}\right)^{-1} \frac{\mathsf{G}^T\mathbf{x}}{\tau^2} \\ \Sigma^* &= \left(\frac{\mathsf{I}}{\sigma^2} + \frac{\mathsf{G}^T\mathsf{G}}{\tau^2}\right)^{-1} \end{aligned}$$

Bottom-up weights *include* prior.

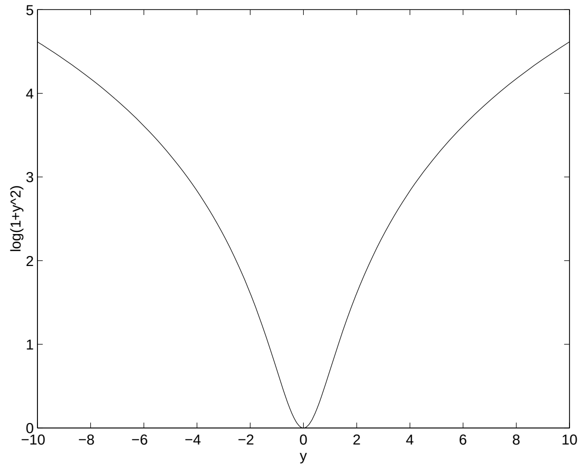
Or *exclude* it and do dynamics.

Sparse Coding

Neural reasons for sparsity.

$$\mathcal{P}[\mathbf{y};\mathcal{G}] = \prod_a e^{-f(y_a)} \qquad \mathcal{P}[\mathbf{x}|\mathbf{y};\mathcal{G}] \sim \mathcal{N}[\mathsf{G}\mathbf{y}, \tau^2 \mathsf{I}]$$

with $f(y) = \alpha \log(y_0^2 + y^2)$



Olshausen & Field used deterministic:

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

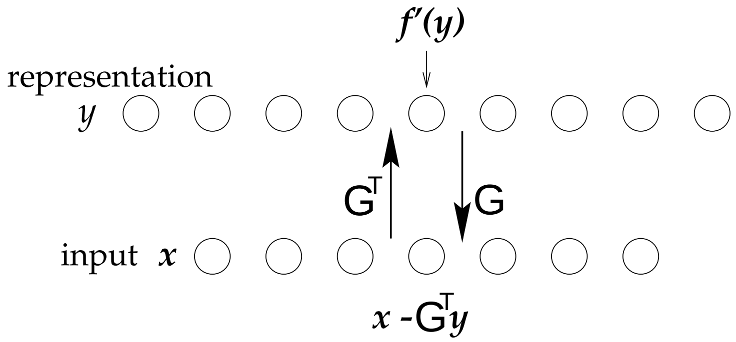
and minimised wrt $\tilde{\mathbf{y}}$ (*cf* mean field):

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

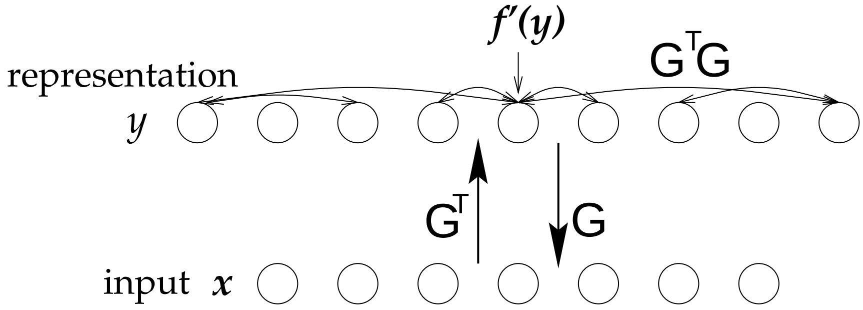
E Phase

To implement:
$\tilde{\mathbf{y}}(\tau+1) = \tilde{\mathbf{y}}(\tau) - \epsilon \nabla_{\tilde{\mathbf{y}}} \mathcal{F}[\mathbf{x}^i, \tilde{\mathbf{y}}(\tau); \mathcal{G}]$:

$$\tilde{\mathbf{y}}(\tau) + \epsilon \left[\mathsf{G}^T \left(\mathbf{x}^i - \mathsf{G}\tilde{\mathbf{y}}(\tau) \right) - \mathbf{f}'(\tilde{\mathbf{y}}(\tau)) \right]$$



$$\tilde{\mathbf{y}}(\tau) + \epsilon \mathsf{G}^T \mathbf{x}^i - \epsilon \left(\mathsf{G}^T \mathsf{G} \tilde{\mathbf{y}}(\tau) + \mathbf{f}'(\tilde{\mathbf{y}}(\tau)) \right)$$

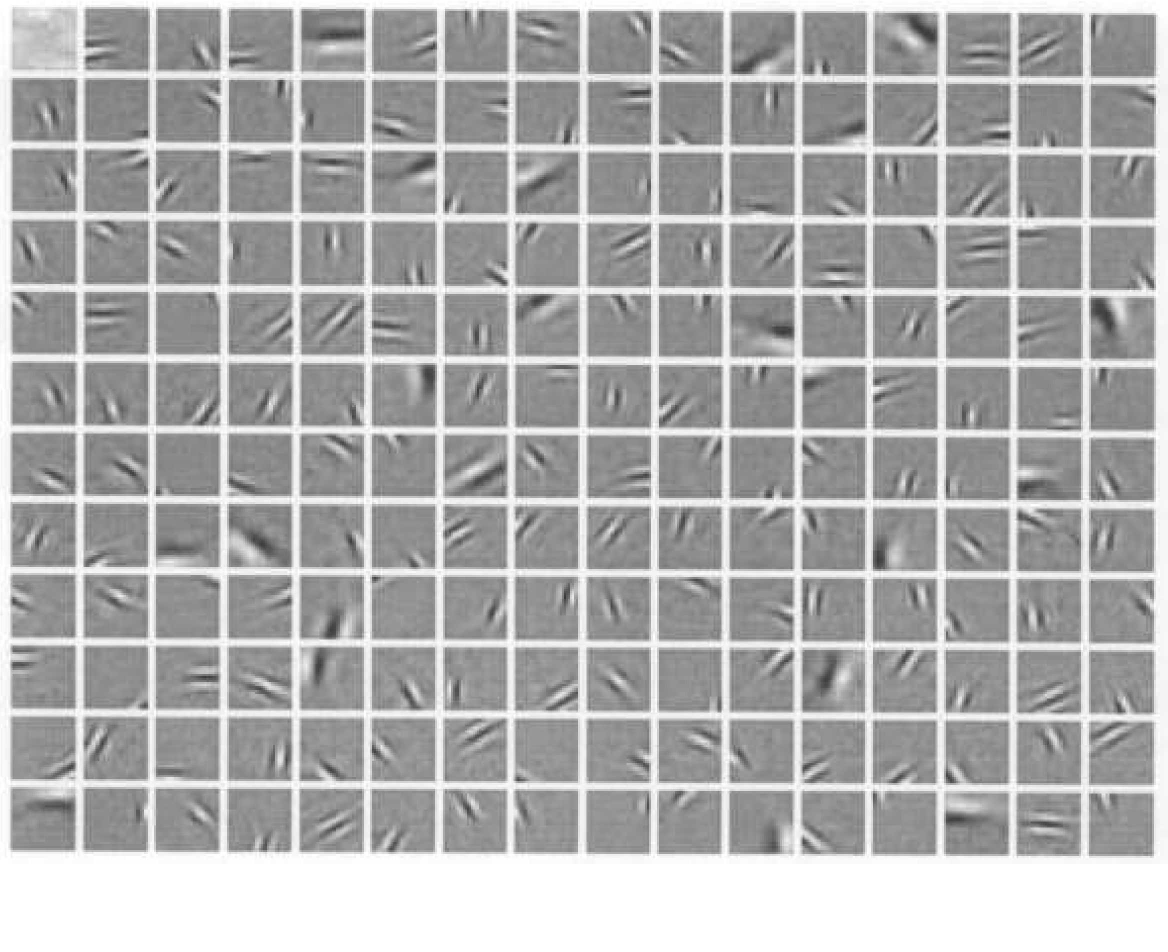


M Phase

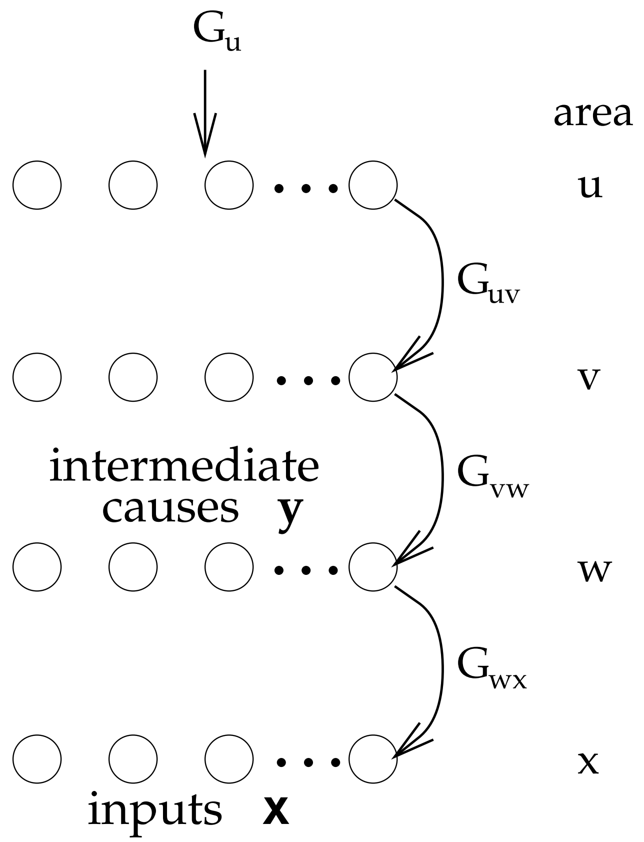
Just uses the delta rule:

$$\begin{aligned} \mathsf{G}_{ba}(t+1) &= \mathsf{G}_{ba}(t) - \alpha \nabla_{\mathsf{G}_{ba}} \mathcal{F}[\mathbf{x}^i, \tilde{\mathbf{y}}^*; \mathsf{G}(t)] \\ &= \mathsf{G}_{ba}(t) + \alpha \left[\mathbf{x}^i - \mathsf{G}(t)\tilde{\mathbf{y}}^*\right]_b \tilde{y}_a^*. \end{aligned}$$

plus normalisation *etc*:



The Helmholtz Machine

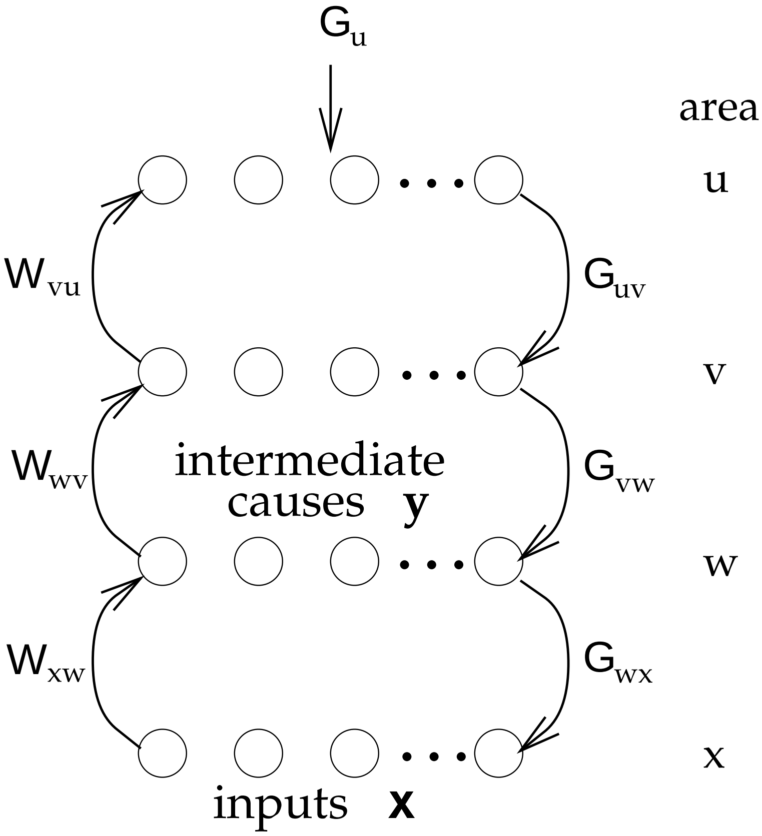


$$\mathcal{P}[\mathbf{x},\mathbf{w},\mathbf{v},\mathbf{u};\mathcal{G}] = \mathcal{P}[\mathbf{u};\mathsf{G}_u]\mathcal{P}[\mathbf{v}|\mathbf{u};\mathsf{G}_{uv}] \times \mathcal{P}[\mathbf{w}|\mathbf{v};\mathsf{G}_{vw}]\mathcal{P}[\mathbf{x}|\mathbf{w};\mathsf{G}_{wx}]$$

Belief Net Task: learn $\mathcal{G}_*$ to fit $\mathcal{P}[\mathbf{x};\mathcal{G}_*]$

But what is $\mathcal{P}[\mathbf{w}|\mathbf{x};\mathcal{G}_*]$?

The Suggestion



Use another set of (recognition) parameters W to produce an estimate of the inverse to the generative distribution:

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

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

The Wake-Sleep Algorithm

For linear factor analysis, we saw:

$$\mathcal{P}[\mathbf{y}|\mathbf{x};\mathcal{G}] \sim \mathcal{N}\left[\mathsf{W}^*\mathbf{x}, \mathbf{\Sigma}^*\right]$$

For the Helmholtz machine specify:

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

With:

$$\mathcal{F}[\mathbf{x}^i,\mathcal{Q};\mathcal{G}] = -\sum_{\mathbf{y}} \mathcal{Q}[\mathbf{y};\mathbf{x}^i] \log \frac{\mathcal{P}[\mathbf{x}^i,\mathbf{y};\mathcal{G}]}{\mathcal{Q}[\mathbf{y};\mathbf{x}^i]}$$

Two phases:

wake sample $\mathcal{Q}[\mathbf{y};\mathbf{x}^i]$, delta rule:

$$\nabla_{\mathsf{G}_{ba}} \log \mathcal{P}[\mathbf{x}^i,\mathbf{y};\mathsf{G}] = \frac{1}{\tau_b^2}\left[\mathbf{x}^i - \mathsf{G}\mathbf{y}\right]_b y_a$$

sleep make $\mathcal{Q}[\mathbf{y};\mathbf{x},\mathsf{W},\mathbf{\Sigma}] \sim \mathcal{P}[\mathbf{y}|\mathbf{x};\mathcal{G}]$

The Sleep Phase

Minimise:

$$\int_{\mathbf{x}} d\mathbf{x}\mathcal{P}[\mathbf{x}]KL\left[\mathcal{P}[\mathbf{y}|\mathbf{x}],\mathcal{Q}[\mathbf{y};\mathbf{x}]\right]$$

rather than

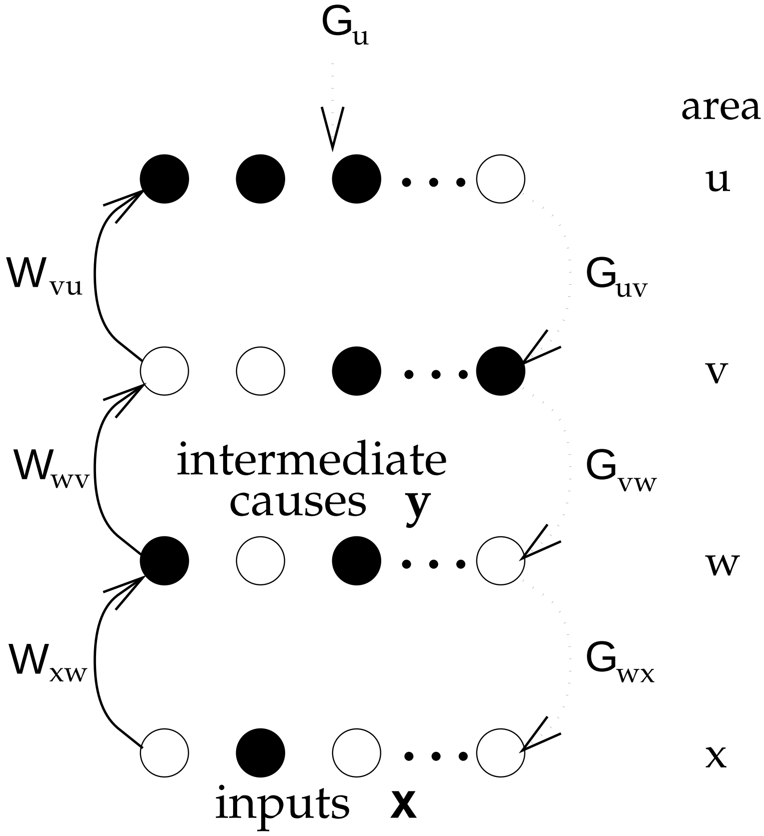
$$\int_{\mathbf{x}} d\mathbf{x}\mathcal{P}[\mathbf{x}]KL\left[\mathcal{Q}[\mathbf{y};\mathbf{x}],\mathcal{P}[\mathbf{y}|\mathbf{x}]\right]$$

leads to learning rules such as:

$$\frac{\partial}{\partial \mathsf{W}_{ab}}\left\{-\log\mathcal{Q}[\mathbf{y}^\circ;\mathbf{x}^\circ]\right\} = \left[\Sigma^{-1}\left(\mathbf{y}^\circ - \mathsf{W}\mathbf{x}^\circ\right)\right]_a x_b.$$

based on samples from $\mathcal{P}[\mathbf{x},\mathbf{y}:\mathcal{G}]$.

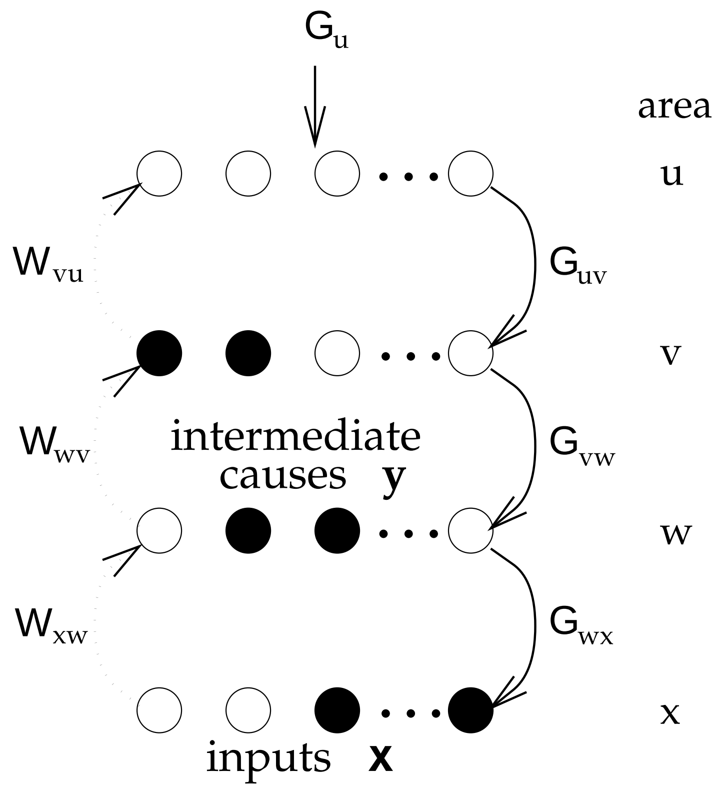
Wake



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

Sleep

Training W is not so simple, since both terms in $\mathcal{F}$ depend on it. Treat recognition like inverse generation:



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

Discussion

- ML density estimation with hidden *causes* $\mathbf{y}$
- causes are used as internal representations
- only a heuristic for unsupervised learning
- employed architectural priors of independence, sparsity, *etc*
- computational intractability leads to approximations in the probability distributions
- rate-based deterministic version using free energy
- relationship to many neural suggestions

Graphical Interpretation