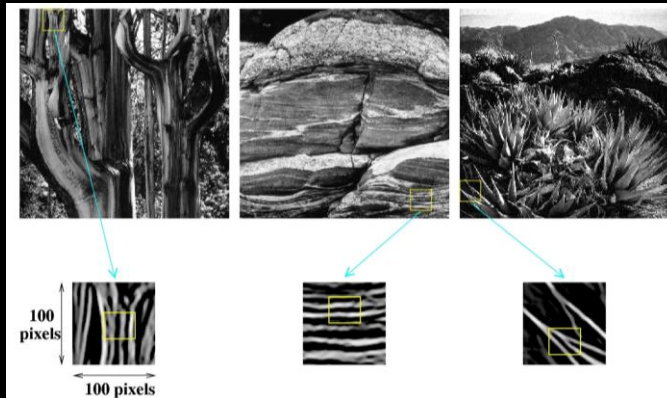


Sparse Coding and Predictive Coding



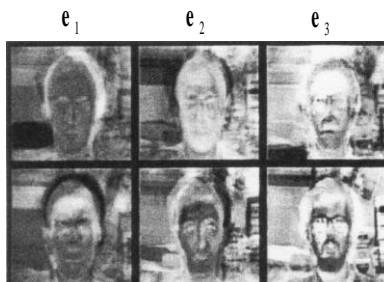
Can we learn a good representation of natural images?
What does the brain do?

Eigenvectors strike again?

Input data \mathbf{u} :
Face images
(N pixels total)



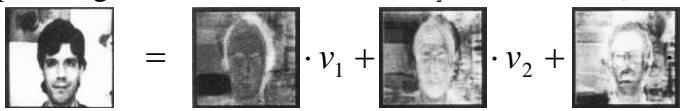
Eigenvectors
of the Input
Covariance
Matrix



“Eigenfaces”

A Linear Model of Images using Eigenvectors

Input image \mathbf{u}



$$\mathbf{u} = \sum_{i=1}^N \mathbf{e}_i v_i$$

Suppose you use only the first M principal eigenvectors:

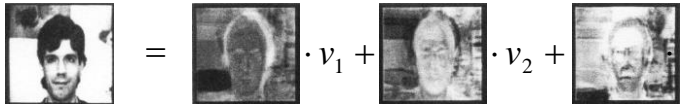
$$\mathbf{u} = \sum_{i=1}^M \mathbf{e}_i v_i + \text{noise} \quad (M < N)$$

Handwritten notes: $M=10$, $N=10^6$, 1000×1000

3

Not so fast, Eigenvectors!

Input image \mathbf{u}




$$\mathbf{u} = \sum_{i=1}^M \mathbf{e}_i v_i + \text{noise} \quad (M < N)$$

Eigenvector representation is good for compression but not so good if you want to extract the local components (or parts) of an image (e.g., parts of a face, local edges in a scene, etc.)

4

A Linear Model for Natural Images

Input image \mathbf{u}



$$= \boxed{?} \cdot v_1 + \boxed{?} \cdot v_2 + \boxed{?} \cdot v_3 + \dots$$

Handwritten annotations: \mathbf{g}_1 , \mathbf{g}_2 , and \mathbf{g}_3 are circled in red above the boxes. Red arrows point from the \mathbf{g}_i terms in the equations below to these boxes.

$$\rightarrow \mathbf{u} = \sum_{i=1}^M \mathbf{g}_i v_i + \text{noise}$$

(Note: M can be larger than N)

$$= \mathbf{G}\mathbf{v} + \text{noise}$$

Handwritten annotations: Red arrows point from the \mathbf{G} and \mathbf{v} in the second equation to the \mathbf{g}_i and v_i in the first equation.

\mathbf{G} = matrix whose columns are \mathbf{g}_i
 \mathbf{v} = vector whose elements are v_i

Defining the Generative Model: Likelihood

Causes \mathbf{v} Prior $p[\mathbf{v}]$

Generative model

Input Data \mathbf{u} Likelihood $p[\mathbf{u} | \mathbf{v}; G]$

Linear model:

$$\mathbf{u} = \mathbf{G}\mathbf{v} + \text{noise}$$

Assume *noise* is Gaussian white noise:

$$p[\mathbf{u} | \mathbf{v}; G] = \text{Gaussian}(\mathbf{u}; \mathbf{G}\mathbf{v}, I)$$

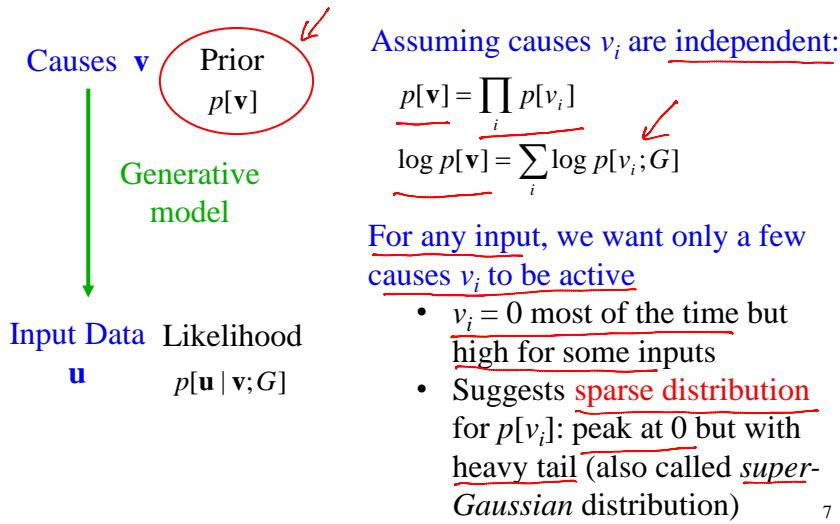
$$\propto \exp\left(-\frac{1}{2}\|\mathbf{u} - \mathbf{G}\mathbf{v}\|^2\right)$$

Log likelihood:

$$\log p[\mathbf{u} | \mathbf{v}; G] = -\frac{1}{2}\|\mathbf{u} - \mathbf{G}\mathbf{v}\|^2 + C$$

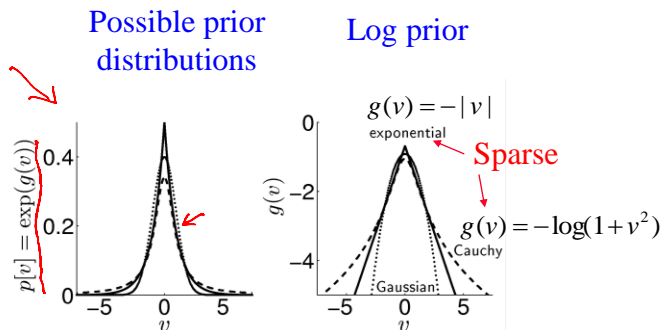
Handwritten annotations: Red arrows point from the \mathbf{v} in the linear model to the Prior and from the \mathbf{u} in the linear model to the Input Data. Red circles highlight the Likelihood and the $\mathbf{u} - \mathbf{G}\mathbf{v}$ term in the log likelihood equation. Red underlines are present under \mathbf{g}_i , v_i , $\mathbf{G}\mathbf{v}$, and \mathbf{u} in various equations.

Defining the Generative Model: Prior



7

Examples of Sparse Prior Distributions



$$\rightarrow p[\mathbf{v}] = c \cdot \prod_i \exp(g(v_i))$$

$$\rightarrow \log p[\mathbf{v}] = \sum_i g(v_i) + c$$

8

Bayesian approach to finding \mathbf{v} and learning G

- Find \mathbf{v} and G that maximize posterior probability of causes:

$$\rightarrow p[\mathbf{v} | \mathbf{u}; G] = k \cdot p[\mathbf{u} | \mathbf{v}; G] p[\mathbf{v}; G]$$

- Equivalently, maximize log posterior:

$$\begin{aligned} \rightarrow F(\mathbf{v}, G) &= \log p[\mathbf{u} | \mathbf{v}; G] + \log p[\mathbf{v}; G] + \log k \\ &= -\frac{1}{2} \|\mathbf{u} - G\mathbf{v}\|^2 + \sum_i g(v_i) + K \end{aligned}$$

Alternate between two steps

(similar to EM algorithm):

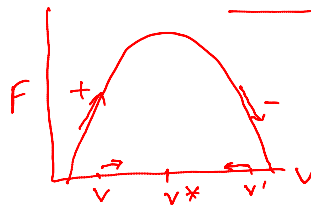
\rightarrow Maximize F with respect to \mathbf{v} ,
keeping G fixed

How? \leftarrow

\rightarrow Maximize F with respect to G ,
given the \mathbf{v} from above

How? \leftarrow

Gradient ascent $\frac{d\mathbf{v}}{dt} \propto \frac{dF}{d\mathbf{v}}$



Finding \mathbf{v} for a given input

$$\text{Gradient ascent} \quad \frac{d\mathbf{v}}{dt} \propto \frac{dF}{d\mathbf{v}} = G^T (\mathbf{u} - G\mathbf{v}) + g'(\mathbf{v})$$

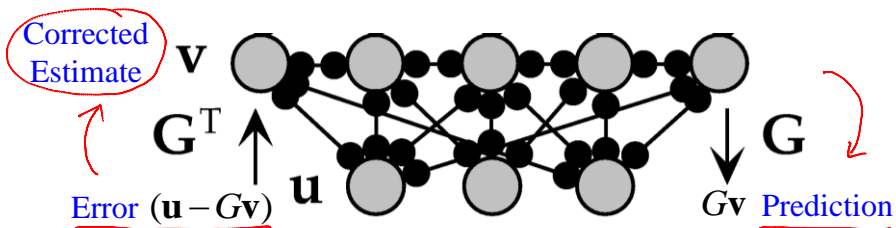
$$\tau \frac{d\mathbf{v}}{dt} = G^T (\mathbf{u} - G\mathbf{v}) + g'(\mathbf{v})$$

Error
Reconstruction (prediction) of \mathbf{u}
Firing rate dynamics (Recurrent network)

Sparseness constraint

Recurrent Network Implementation of Sparse Coding

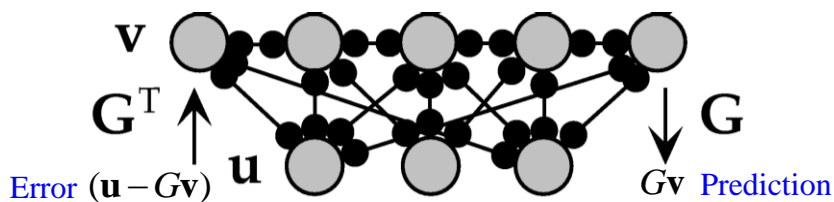
$$\tau \frac{d\mathbf{v}}{dt} = \mathbf{G}^T (\mathbf{u} - \mathbf{G}\mathbf{v}) + g'(\mathbf{v})$$



11

Image Source: Dayan & Abbott textbook

Learning the Synaptic Weights \mathbf{G}



Gradient ascent

$$\frac{dG}{dt} \propto \frac{dF}{dG} = (\mathbf{u} - \mathbf{Gv})\mathbf{v}^T$$

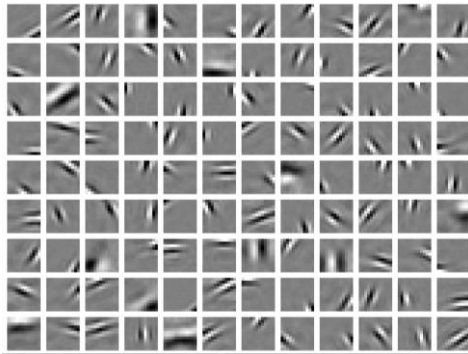
Learning rule

$$\tau_G \frac{dG}{dt} = (\mathbf{u} - \mathbf{Gv})\mathbf{v}^T$$

Hebbian!
(similar to Oja's rule)

12

Result: Learning G for Natural Images

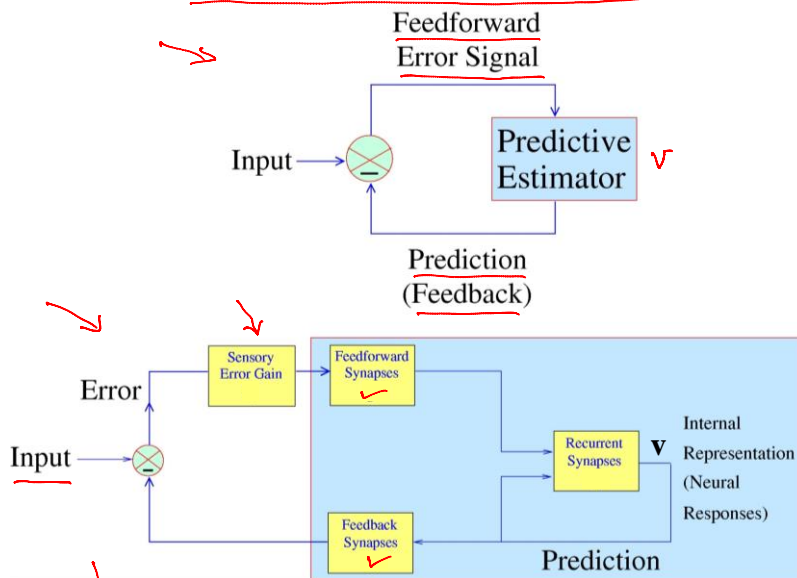


(Olshausen & Field, 1996)

Each square is a column \mathbf{g}_i of \mathbf{G} (obtained by collapsing rows of the square into a vector)

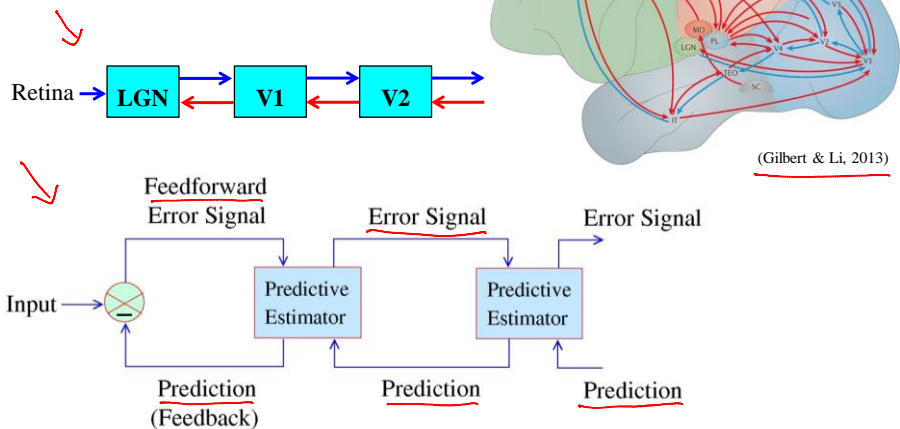
The \mathbf{g}_i look like local edge or bar features similar to receptive fields in primary visual cortex (V1)

Sparse Coding Network is a special case of Predictive Coding Networks



See Supplementary Materials: (Rao, *Vision Research*, 1999; Rao & Ballard, *Nature Neurosci.*, 1999)

Predictive Coding Model of the Visual Cortex



See Supplementary Materials: (Rao & Ballard, *Nature Neurosci.*, 1999)

Computational Neuroscience

Next Week:
Neurons as Classifiers and
Reinforcement Learning