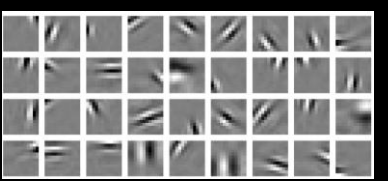
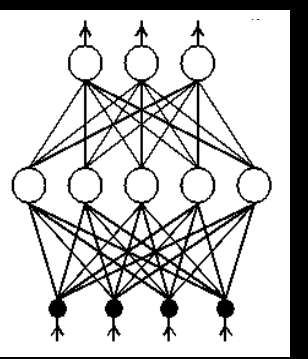
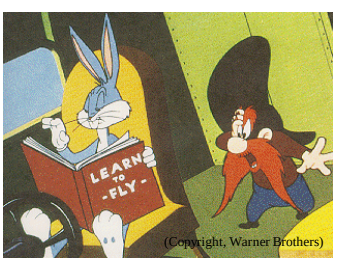
CSE/NB 528
Lecture 12: Unsupervised Learning
(Chapters 8 & 10)





Today's Agenda: Learning about Learning

- Hebbian learning and its variants (Covariance, Oja rule)
  - Relation to Principal Component Analysis (PCA)
- Unsupervised Learning
  - Generative Models



(Copyright, Warner Brothers)

Flashback: Hebbian Learning

- Linear neuron: $v = \mathbf{w}^T\mathbf{u} = \mathbf{u}^T\mathbf{w}$
- Basic Hebb Rule: $\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{u}v$ (or $\mathbf{w} \rightarrow \mathbf{w} + \varepsilon \cdot \mathbf{u}v$)
- What is the average effect of this rule?

$$\tau_w \frac{d\mathbf{w}}{dt} = \langle \mathbf{u}v \rangle_{\mathbf{u}} = \langle \mathbf{u}\mathbf{u}^T\mathbf{w} \rangle_{\mathbf{u}} = \langle \mathbf{u}\mathbf{u}^T \rangle_{\mathbf{u}} \mathbf{w} = Q\mathbf{w}$$

- Q is the input correlation matrix: $Q = \langle \mathbf{u}\mathbf{u}^T \rangle$

Variants of Hebb's Rule

- Hebb:

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{u}v$$ Unstable

- Covariance rule:

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{u}(v - \langle v \rangle)$$ Unstable

- Oja's rule:

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{u}v - \alpha v^2 \mathbf{w}$$ Stable $\|\mathbf{w}\| \rightarrow \frac{1}{\sqrt{\alpha}}$

What does the Hebb rule do anyway?

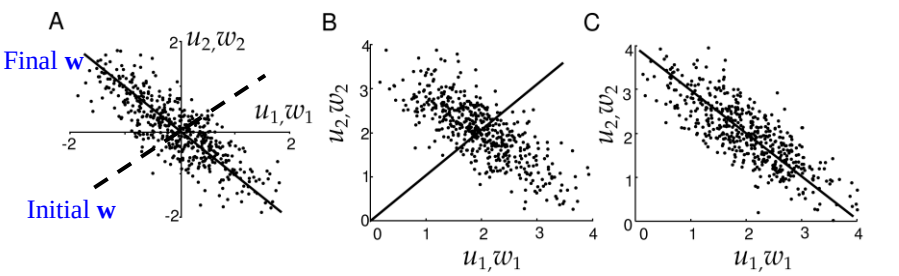
Eigenvector analysis of Hebb rule…

Hebb Rule implements Principal Component Analysis (PCA)!

Pure Hebb
Input mean = (0,0)

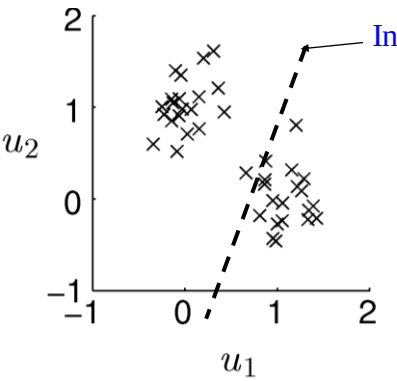
Pure Hebb
Input mean = (2,2)

Covariance Rule
Input mean = (2,2)



Hebb rule *rotates* weight vector to align with principal eigenvector of input correlation/covariance matrix (i.e. direction of maximum variance)

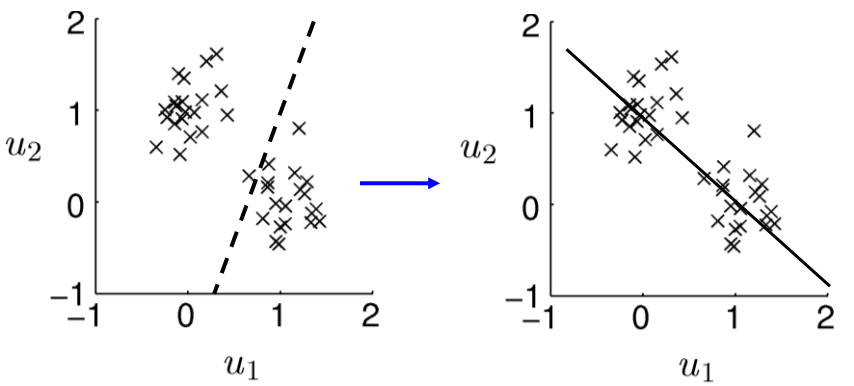
What about this data?



?

What does the covariance rule learn?

PCA does not correctly describe the data



Input data is made up of two clusters (Gaussians)
→ two "causes"

The Goal of Unsupervised Learning

$p[\mathbf{v}]$ (prior) — Causes **v** — $p[\mathbf{v}\,|\,\mathbf{u};G]$ (posterior)

Generative model

$p[\mathbf{u}\,|\,\mathbf{v};G]$ (likelihood) — Data **u**

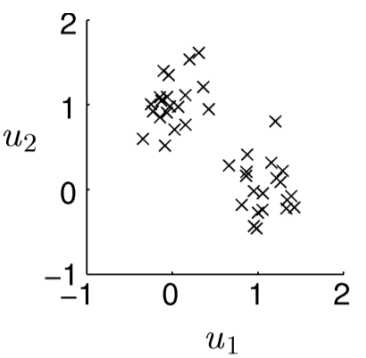
- **Goal:** Learn a "Generative Model" for the data you are seeing
  - Mimic the data generation process
- **General Approach:** Given data **u**, solve two sub-problems:
  - Estimate causes **v** (compute posterior)
  - Learn parameters G

Example 1

$p[\mathbf{v}]$ (prior) — Causes **v**

Generative model

$p[\mathbf{u}\,|\,\mathbf{v};G]$ (likelihood) — Data **u**



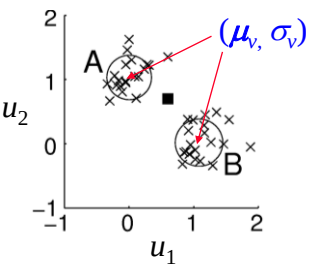
What is a possible generative model for this data?

Example 1: Mixture of Gaussians

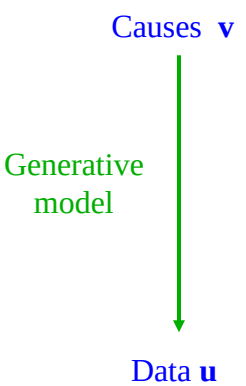
- Generative Model: Assume data was generated by a mixture of Gaussians
- Goal: Learn means and variances of Gaussians as well as priors $p[v]$
- Neural Implementation: Two neurons A and B that learn the means and variances from data
  - Related to competitive learning, learning vector quantization (LVQ), self-organizing map (SOM), K-means, and EM algorithm in machine learning

$$p[\mathbf{u}] = \sum_v p[\mathbf{u} \mid v] p[v]$$

Each $p[\mathbf{u} \mid v]$ is a Gaussian.



Example 2: Linear Generative Model



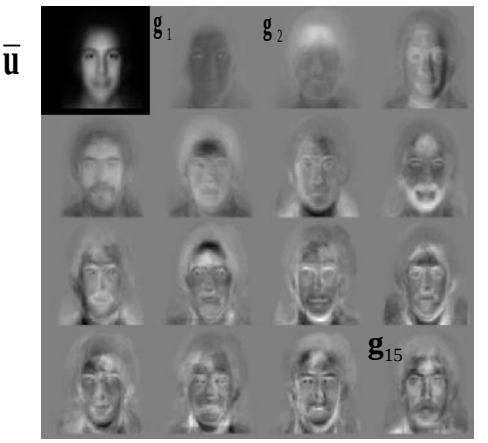
Example: Suppose input **u** was generated by a linear superposition of causes v_1, v_2, …, v_k with basis vectors (or "features") $\mathbf{g}_i$

$$\mathbf{u} = \sum_i \mathbf{g}_i v_i + noise$$

(e.g., an image composed of several features, or audio containing several voices)

Example: "Eigenfaces"

- Suppose your basis vectors or "features" $\mathbf{g}_i$ are the eigenvectors of input covariance matrix (e.g., face images)



Linear combination of eigenfaces

$\mathbf{g}_1 v_1$ $\mathbf{g}_2 v_2$ $\cdots$ $\mathbf{g}_8 v_8$

$$\bar{\mathbf{u}} + \sum_i \mathbf{g}_i v_i$$

Image

Linear Generative Model

- Suppose input **u** was generated by linear superposition of causes $v_1, v_2, \ldots, v_k$ and basis vectors or "features" $\mathbf{g}_i$:

$$\mathbf{u} = \sum_i \mathbf{g}_i v_i + noise = G\mathbf{v} + noise$$

- Problem: For a set of inputs **u**, estimate causes v_i for each **u** and learn feature vectors $\mathbf{g}_i$
  - Suppose number of causes is much lesser than size of input
- Idea: Find **v** and G that minimize reconstruction errors:

$$E = \frac{1}{2}\left\|\mathbf{u} - \sum_i \mathbf{g}_i v_i\right\|^2 = \frac{1}{2}(\mathbf{u} - G\mathbf{v})^T(\mathbf{u} - G\mathbf{v})$$

Probabilistic Interpretation

- E is the same as the *negative log likelihood* of data: Likelihood = Gaussian with mean $G\mathbf{v}$ and identity covariance matrix I

$$p[\mathbf{u} \mid \mathbf{v}; G] = N(\mathbf{u}; G\mathbf{v}, I)$$

$$E = -\log p[\mathbf{u} \mid \mathbf{v}; G] = \frac{1}{2}(\mathbf{u} - G\mathbf{v})^T(\mathbf{u} - G\mathbf{v}) + C$$

Minimizing error function E is the same as maximizing log likelihood of the data

Bayesian approach

- Find **v** and G that maximize posterior:

$$p[\mathbf{v}\,|\,\mathbf{u};G] \propto p[\mathbf{u}\,|\,\mathbf{v};G]\,p[\mathbf{v};G]$$

- Equivalently, find **v** and G that maximize:

$$F(\mathbf{v},G) = \left\langle \log p[\mathbf{u}\,|\,\mathbf{v};G] + \log p[\mathbf{v};G] \right\rangle$$

Prior for causes (what should this be?)

What do we know about the causes **v**?

- We would like the causes to be *independent*
  - If cause A and cause B always occur together, then perhaps they should be treated as a single cause AB?
- Examples:
  - Image: Composed of several independent edges
  - Sound: Composed of independent spectral components
  - Objects: Composed of several independent parts

What do we know about the causes **v**?

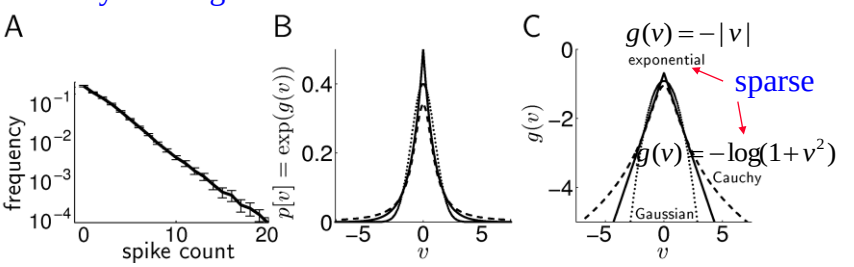
- We would like the causes to be *independent*
- Idea 1: We would like: $p[\mathbf{v};G] = \prod_a p[v_a;G]$
- Idea 2: If causes are independent, only a few of them will be active for any input
  - v_a will be 0 most of the time but high for a few inputs
  - Suggests a sparse distribution for the prior $p[v_a;G]$

Prior Distributions for Causes

Spikes in area IT in monkey viewing TV

Possible prior distributions

Log prior



$$p[\mathbf{v};G] \propto \prod_a \exp(g(v_a))$$

Next Class:
Predictive Coding
Supervised Learning
Reinforcement Learning

To Do:

Homework #3

Group project