Machine Learning 10-701

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University

March 29, 2011

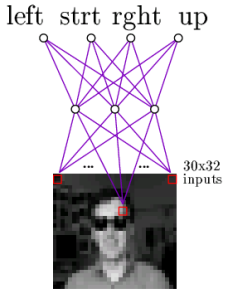
Today:
Learning representations II

- Artificial neural networks
- PCA
- ICA
- CCA

Readings:

- Bishop Ch. 12 through 12.1
- "A Tutorial on PCA," J. Schlens
- Wall et al., 2003

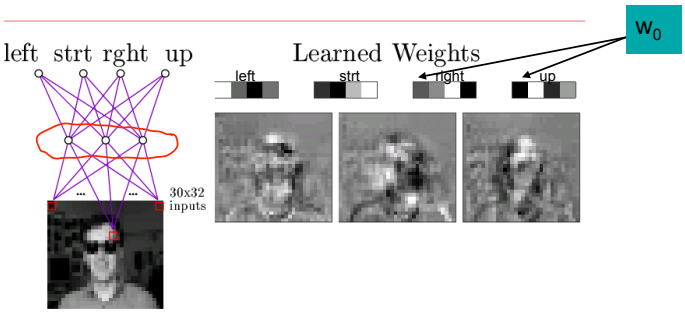
Neural Nets for Face Recognition



Typical input images

90% accurate learning head pose, and recognizing 1-of-20 faces

Learned Hidden Unit Weights

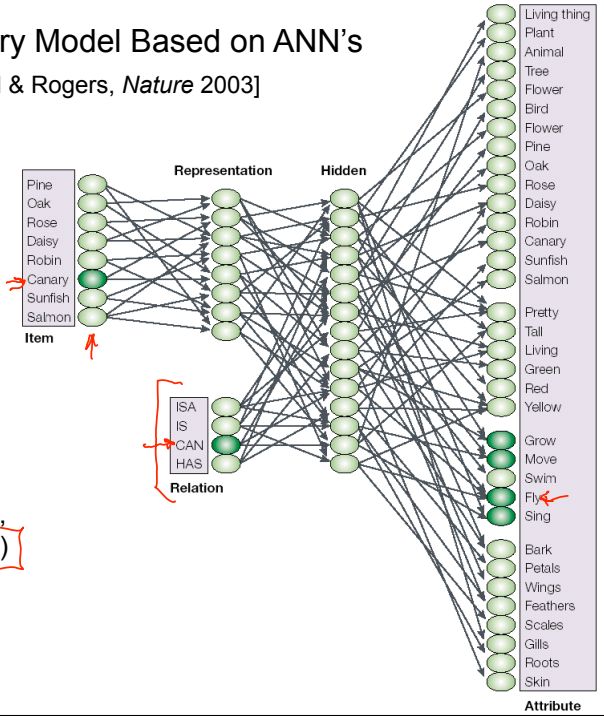


Typical input images

http://www.cs.cmu.edu/~tom/faces.html

Semantic Memory Model Based on ANN's

[McClelland & Rogers, *Nature* 2003]

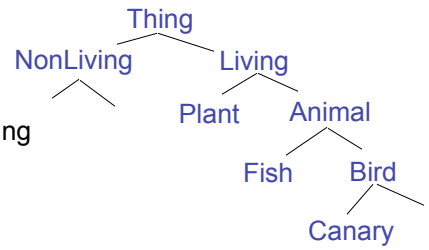


Train with assertions,
e.g., Can(Canary,Fly)

Humans act as though they have a hierarchical memory organization

1. Victims of Semantic Dementia progressively lose knowledge of objects
   But they lose specific details first, general properties later, suggesting hierarchical memory organization

2. Children appear to learn general categories and properties first, following the same hierarchy, top down*.



Question: What learning mechanism could produce this emergent hierarchy?

* some debate remains on this.

Memory deterioration follows semantic hierarchy

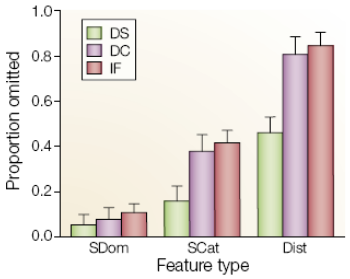
[McClelland & Rogers, *Nature* 2003]

a

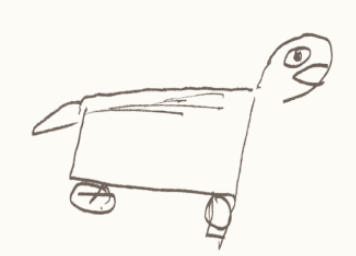
Picture naming responses for JL

Item	Sept. 91	March 92	March 93
Bird	+	+	Animal
Chicken	+	+	Animal
Duck	+	Bird	Dog
Swan	+	Bird	Animal
Eagle	Duck	Bird	Horse
Ostrich	Swan	Bird	Animal
Peacock	Duck	Bird	Vehicle
Penguin	Duck	Bird	Part of animal
Rooster	Chicken	Chicken	Dog

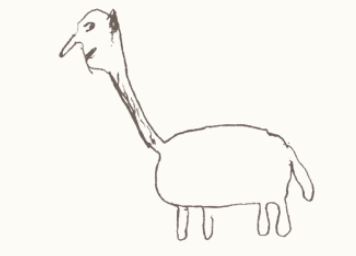
b



c IF's delayed copy of a camel



d DC's delayed copy of a swan



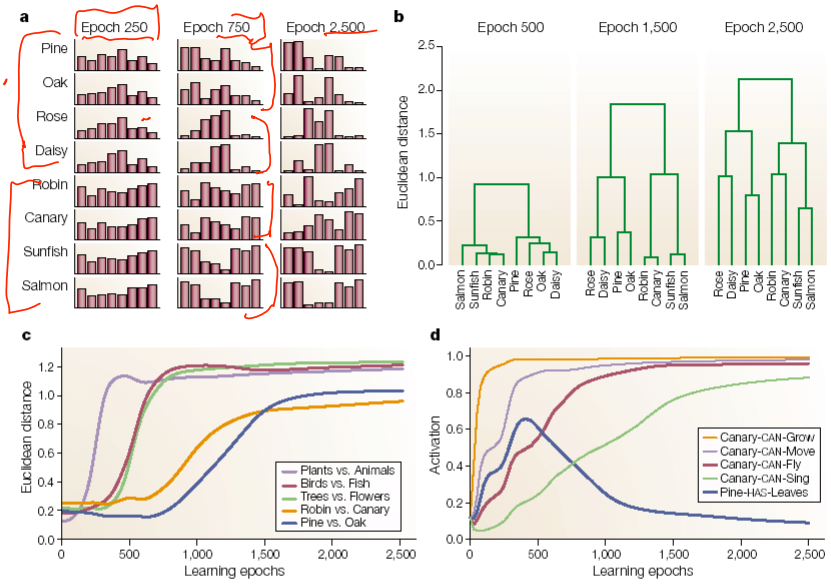
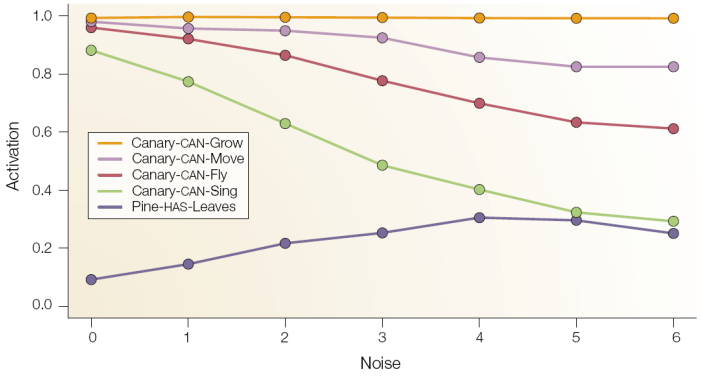


Figure 4 | **The process of differentiation of conceptual representations.** The representations are those seen in the feedforward network model shown in FIG. 3. **a** | Acquired patterns of activation that represent the eight objects in the training set at three points in the learning process (epochs 250, 750 and 2,500). Early in learning, the patterns are undifferentiated; the first difference to appear is between plants and animals. Later, the patterns show clear differentiation at both the superordinate (plant–animal) and intermediate (bird–fish/tree–flower) levels. Finally, the individual concepts are differentiated, but the overall hierarchical organization of the similarity structure remains. **b** | A standard hierarchical clustering analysis program has been used to visualize the similarity structure in the

ANN Also Models Progressive Deterioration

[McClelland & Rogers, *Nature* 2003]



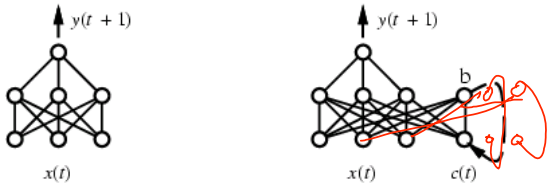
average effect of noise in inputs to hidden layers

Training Networks on Time Series

- Suppose we want to predict next state of world
  - and it depends on history of unknown length
  - e.g., robot with forward-facing sensors trying to predict next sensor reading as it moves and turns

Training Networks on Time Series

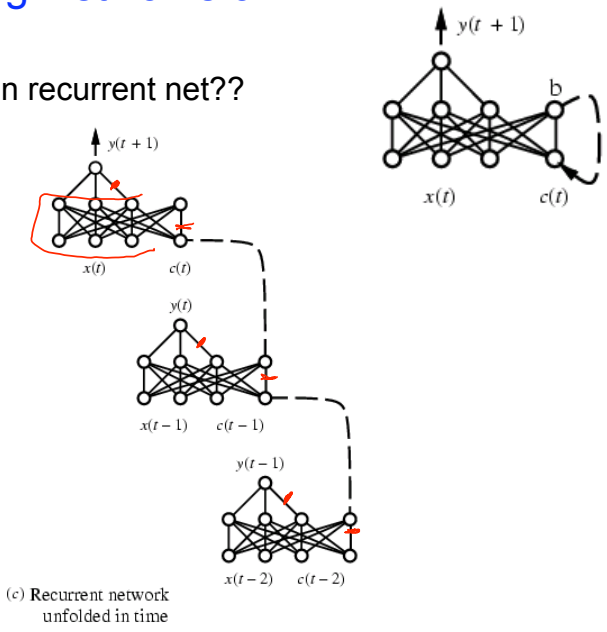
- Suppose we want to predict next state of world
  - and it depends on history of unknown length (non-Markovian)
  - e.g., robot with forward-facing sensors trying to predict next sensor reading as it moves and turns
- Idea: use hidden layer in network to capture state history



(a) Feedforward network (b) Recurrent network

Training Networks on Time Series

How can we train recurrent net??



(c) Recurrent network unfolded in time

Summary: Neural Networks

- Represent highly non-linear decision surfaces
- Learn f: X → Y, where Y is vector (e.g., image)
- Hidden layer represents re-representation of input
  - to optimize prediction accuracy (minimize sum sq error)
- Role in modeling human cognition

- Local minimum problems solving for MLE/MAP parameters using gradient descent

Learning Lower Dimensional Representations

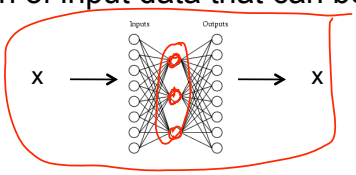
- Supervised learning of lower dimension representation
  - Hidden layers in Neural Networks
  - Fisher linear discriminant
- Unsupervised learning of lower dimension representation
  - Principle Components Analysis (PCA)
  - Independent components analysis (ICA)
  - Canonical correlation analysis (CCA)

Principle Components Analysis

- Idea:
  - Given data points in d-dimensional space, project into lower dimensional space while preserving as much information as possible
    - E.g., find best planar approximation to 3D data
    - E.g., find best planar approximation to 10^4 D data
  - In particular, choose projection that minimizes the squared error in reconstructing original data

Principle Components Analysis

- Like auto-encoding neural networks, learn re-representation of input data that can best reconstruct it

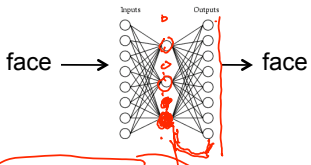


PCA:

- learned encoding is *linear* function of inputs (not *logistic)*
- No local minimum problems when training!
- Given d-dimensional data X, learns d-dimensional representation, where
  - the dimensions are orthogonal
  - top k dimensions are the k-dimensional linear re-representation that minimizes reconstruction error (sum of squared errors)

PCA Example

$$face_i = \Sigma_k\ c_{ik}\ eigenface_k$$



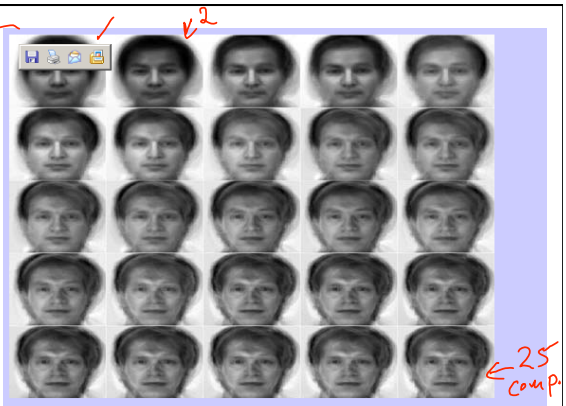
faces

eigenfaces

Thanks to Christopher DeCoro
see http://www.cs.princeton.edu/~cdecoro/eigenfaces/

Reconstructing a face from the first N components (eigenfaces)

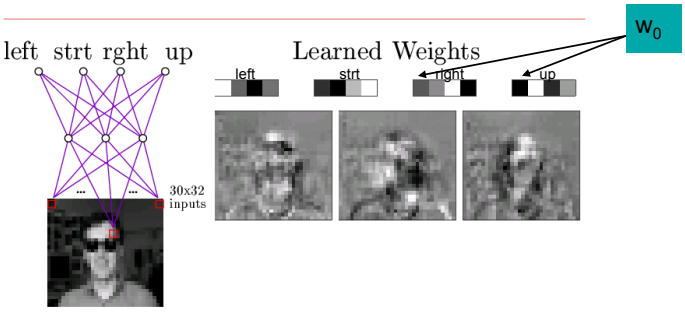
Adding 1 additional PCA component at each step



In this next image, we show a similar picture, but with each additional face representing an additional 8 principle components. You can see that it takes a rather large number of images before the picture looks totally correct.

Adding 8 additional PCA components at each step

Learned Hidden Unit Weights



Typical input images

http://www.cs.cmu.edu/~tom/faces.html

PCA: Find Projections to Minimize Reconstruction Error

Assume data is set of d-dimensional vectors, where nth vector is

$\mathbf{x}^n = \langle x_1^n \ldots x_d^n \rangle$

We can represent these in terms of any d orthogonal vectors $u_1 \ldots u_d$

$$\mathbf{x}^n = \sum_{i=1}^{d} z_i^n \mathbf{u}_i; \quad \mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}$$

PCA: given M<d. Find $\langle \mathbf{u}_1 \ldots \mathbf{u}_M \rangle$

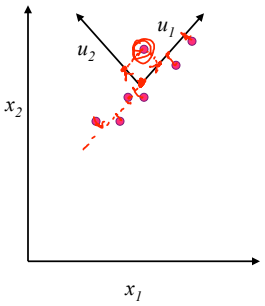
that minimizes $E_M \equiv \sum_{n=1}^{N} ||\mathbf{x}^n - \hat{\mathbf{x}}^n||^2$

where $\hat{\mathbf{x}}^n = \bar{\mathbf{x}} + \sum_{i=1}^{M} z_i^n \mathbf{u}_i$

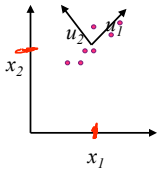
Mean

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}^n$$

original, reconstructed



PCA



PCA: given M<d. Find $\langle \mathbf{u}_1 \ldots \mathbf{u}_M \rangle$

that minimizes $E_M \equiv \sum_{n=1}^{N} ||\mathbf{x}^n - \hat{\mathbf{x}}^n||^2$

where $\hat{\mathbf{x}}^n = \bar{\mathbf{x}} + \sum_{i=1}^{M} z_i^n \mathbf{u}_i$

dimensions to encode x's

dimension of x

Note we get zero error if M=d, so all error is due to missing components.

Therefore,

$$E_M = \sum_{i=M+1}^{d} \sum_{n=1}^{N} [\mathbf{u}_i^T (\mathbf{x}^n - \bar{\mathbf{x}})]^2$$

$$= \sum_{i=M+1}^{d} \mathbf{u}_i^T \Sigma \, \mathbf{u}_i$$

This minimized when $\boldsymbol{u}_i$ is eigenvector of Σ, the covariance matrix of X. i.e., minimized when:

$$\Sigma \mathbf{u}_i = \lambda_i \mathbf{u}_i$$

Covariance matrix: $\Sigma = \sum (\mathbf{x}^n - \bar{\mathbf{x}})(\mathbf{x}^n - \bar{\mathbf{x}})^T$

$$\Sigma_{ij} = \sum_{n=1}^{N} (x_i^n - \bar{x}_i)(x_j^n - \bar{x}_j)$$

PCA

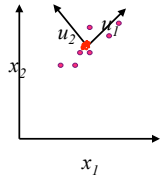
Minimize $E_M = \sum_{i=M+1}^{d} \mathbf{u}_i^T \Sigma \, \mathbf{u}_i$

$\rightarrow \quad \Sigma \mathbf{u}_i = \lambda_i \mathbf{u}_i$

Eigenvector of Σ

Eigenvalue (scalar)

$\rightarrow \quad E_M = \sum_{i=M+1}^{d} \lambda_i$

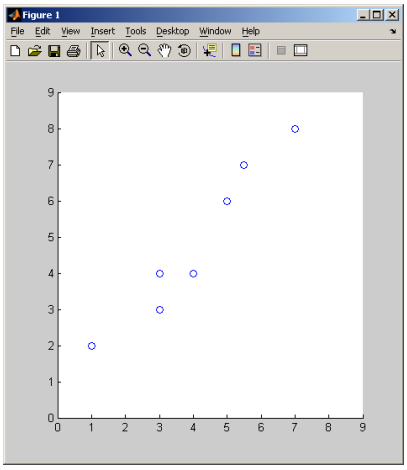


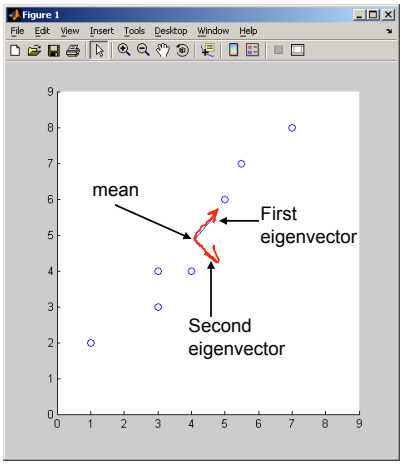
PCA algorithm 1:

1. X ← Create N x d data matrix, with one row vector x^n per data point
2. X ← subtract mean $\bar{x}$ from each row vector x^n in X
3. Σ ← covariance matrix of X
4. Find eigenvectors and eigenvalues of Σ
5. PC's ← the M eigenvectors with largest eigenvalues

PCA Example

$$\hat{\mathbf{x}}^n = \bar{\mathbf{x}} + \sum_{i=1}^{M} z_i^n \mathbf{u}_i$$

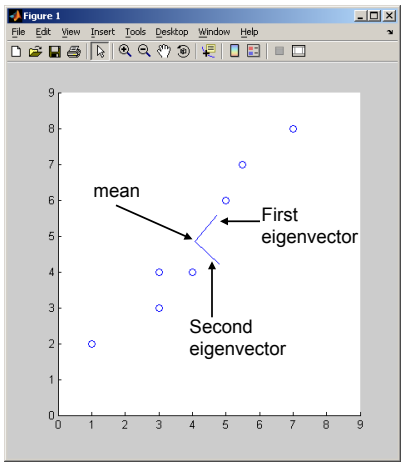


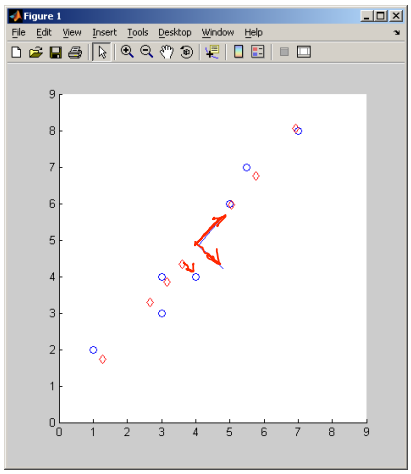


PCA Example

$$\hat{\mathbf{x}}^n = \bar{\mathbf{x}} + \sum_{i=1}^{M} z_i^n \mathbf{u}_i$$

Reconstructed data using only first eigenvector (M=1)





Very Nice When Initial Dimension Not Too Big

What if very large dimensional data?

- e.g., Images (d ¸ 10^4)

Problem:

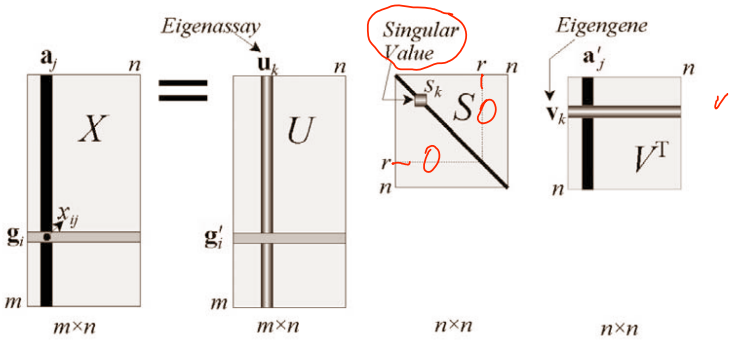
- Covariance matrix Σ is size (d x d)
- d=10^4 → $|\Sigma| = 10^8$

Singular Value Decomposition (SVD) to the rescue!

- pretty efficient algs available, including Matlab SVD
- some implementations find just top N eigenvectors

SVD

$$X = USV^T$$



Data X, one row per data point

US gives coordinates of rows of X in the space of principle components

S is diagonal, $S_k > S_{k+1}$, S_k^2 is kth largest eigenvalue

Rows of V^T are unit length eigenvectors of X^TX

If cols of X have zero mean, then $X^TX = c\,\Sigma$ and eigenvects are the Principle Components

[from Wall et al., 2003]

Singular Value Decomposition

To generate principle components:

- Subtract mean $\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x^n$ from each data point, to create zero-centered data
- Create matrix X with one row vector per (zero centered) data point
- Solve SVD: $X = USV^T$
- Output Principle components: columns of V (= rows of V^T)
  - Eigenvectors in V are sorted from largest to smallest eigenvalues
  - S is diagonal, with s_k^2 giving eigenvalue for kth eigenvector

Singular Value Decomposition

To project a point (column vector x) into PC coordinates:

$V^T x$

$X = U S V^T$

If x_i is i[th] row of data matrix X, then

- (i[th] row of US) $= V^T x_i^T$
- $(US)^T = V^T X^T$

To project a column vector x to M dim Principle Components subspace, take just the first M coordinates of $V^T x$

Independent Components Analysis (ICA)

- PCA seeks orthogonal directions <Y_1 ... Y_M> in feature space X that minimize reconstruction error

- ICA seeks directions <Y_1 ... Y_M> that are most *statistically independent*. I.e., that minimize $I(Y)$, the mutual information between the Y_i :

$$I(Y) = \left[\sum_{j=1}^{J} H(Y_j)\right] - H(Y)$$