University of California
Berkeley

College of Engineering
Department of Electrical Engineering
and Computer Sciences

Professors : N.Morgan / B.Gold

EE225D

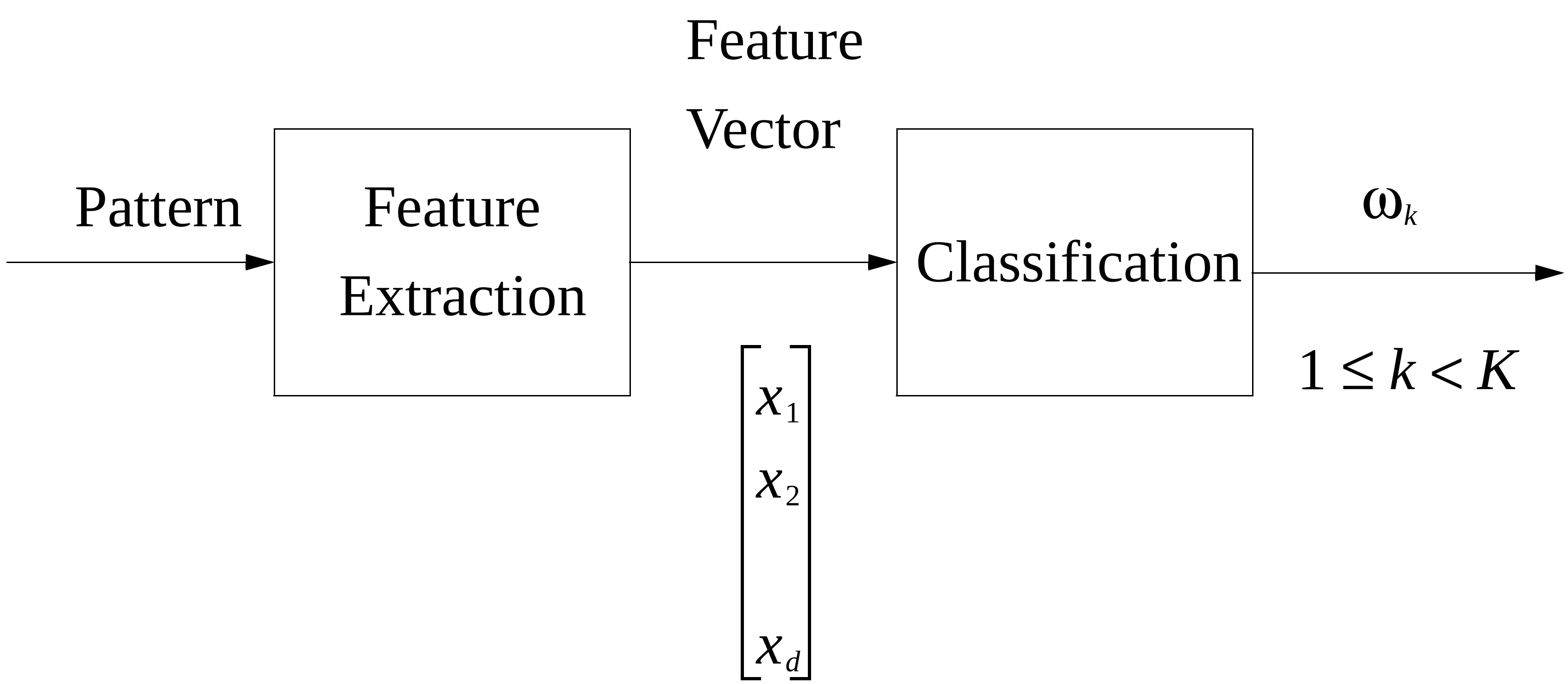
Spring,1999

Pattern Classification

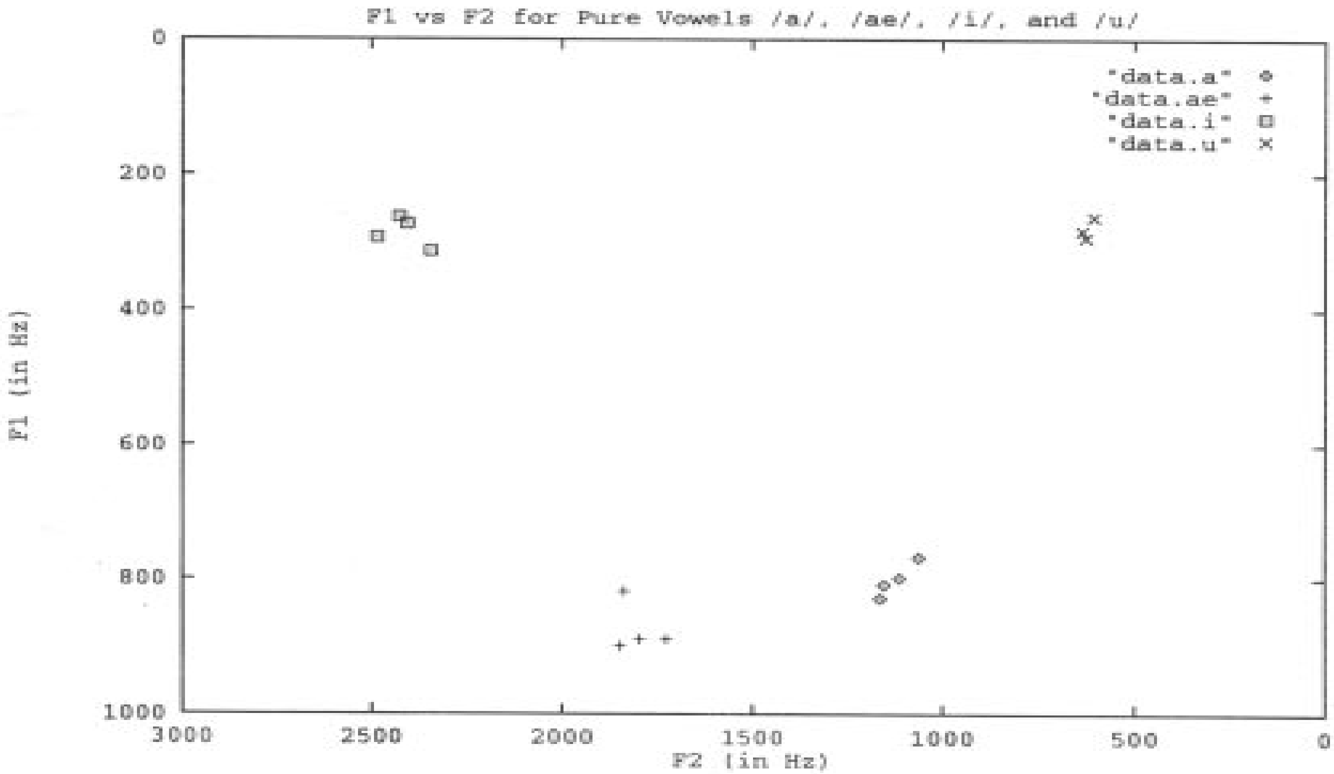
Lecture 8

Speech Pattern Recognition

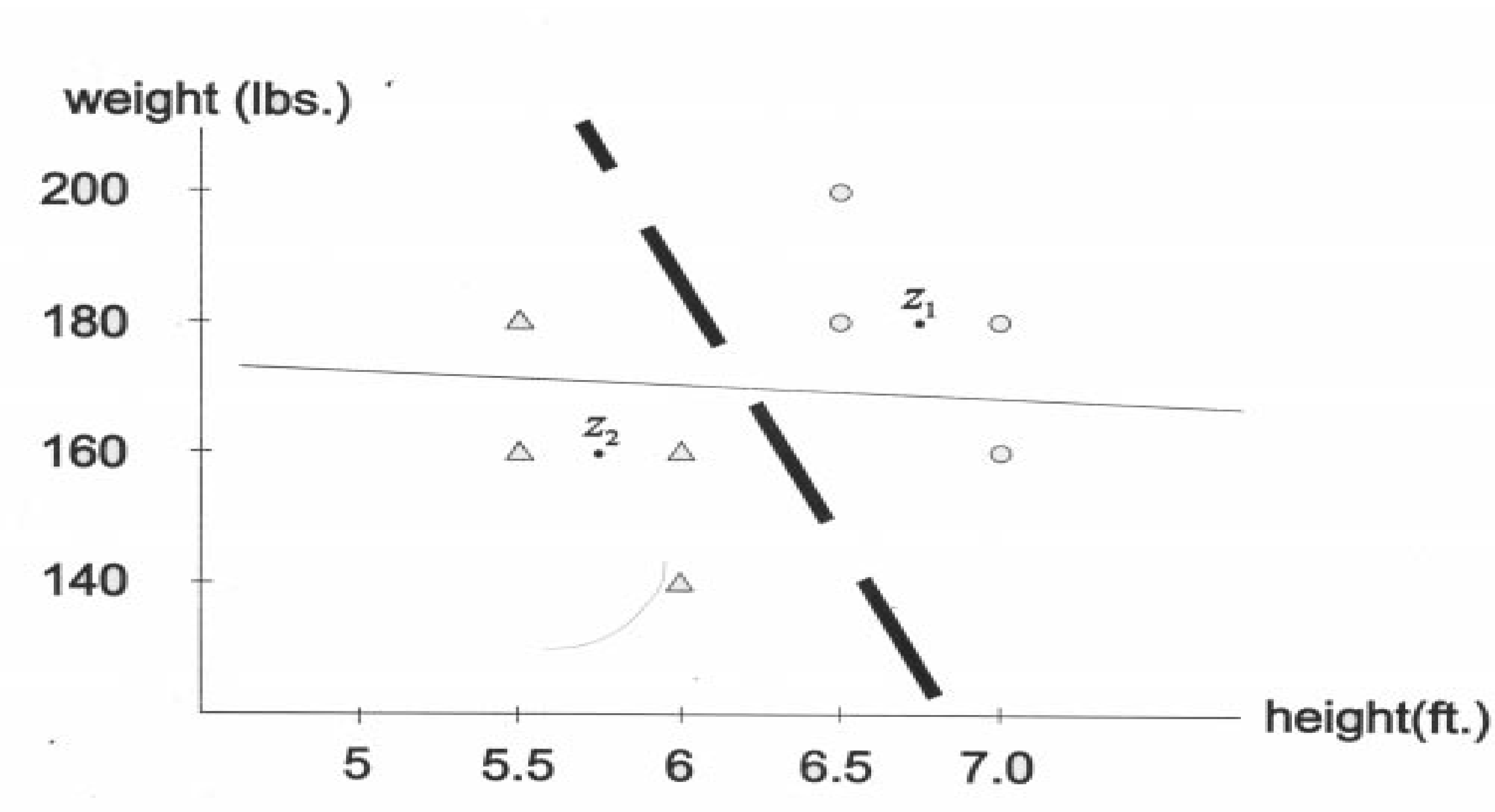
- Soft pattern classification plus temporal sequence integration
- Supervised pattern classification: class labels used in training
- Unsupervised pattern classification: class labels not available or used


Pattern
Feature Extraction
Feature Vector
$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$
Classification
ω_k
$1 \le k < K$

- Training: learning parameters of classifier
- Testing: classify independent test set, compare with labels and score

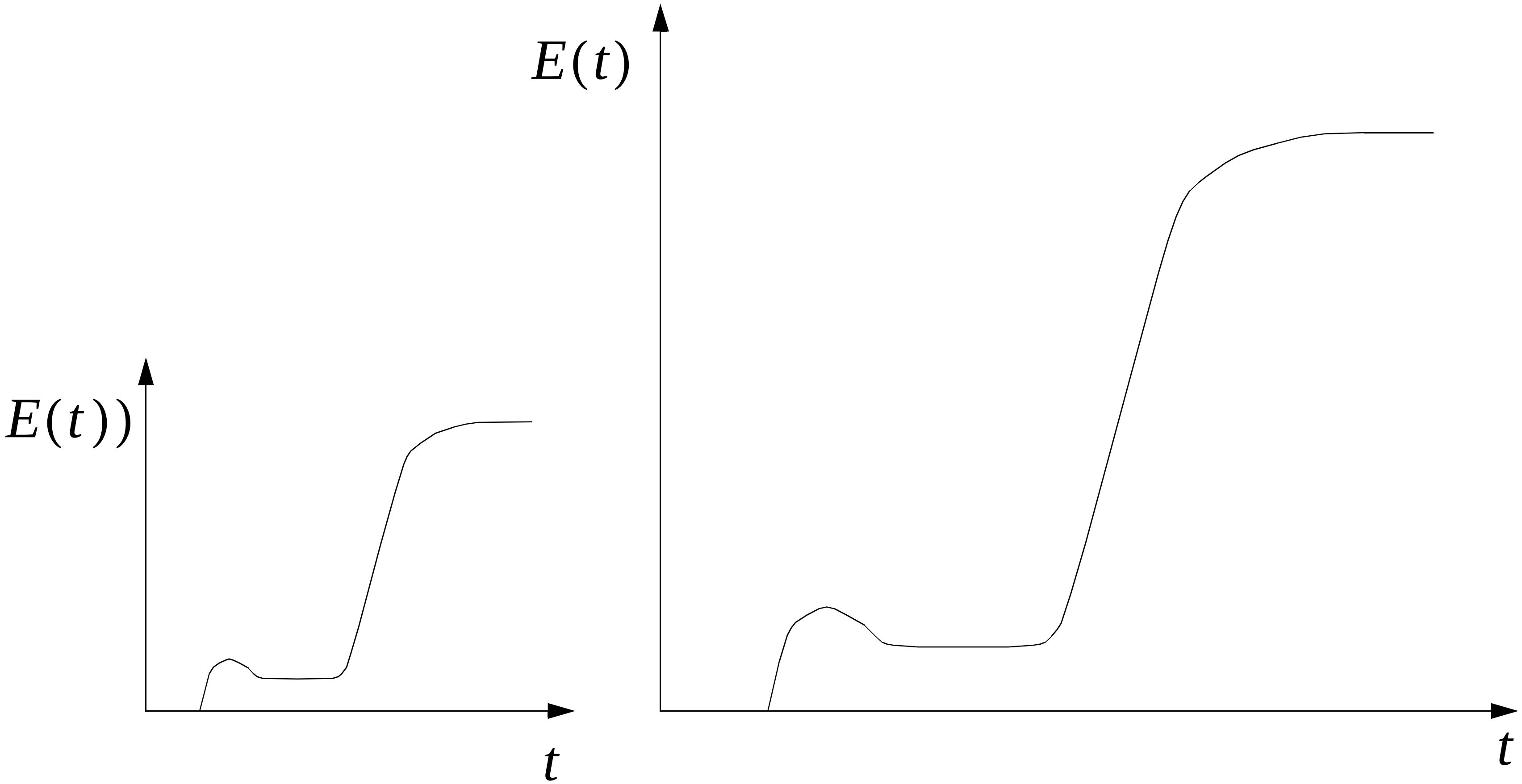


F1 vs F2 for Pure Vowels /a/, /ae/, /i/, and /u/
"data.a"
"data.ae"
"data.i"
"data.u"
F1 (in Hz)
0
200
400
600
800
1000
3000
2500
2000
1500
1000
500
0
F2 (in Hz)


weight (lbs.)
200
180
160
140
z_1
z_2
height(ft.)
5
5.5
6
6.5
7.0

Feature Extraction Criteria

- Class discrimination
- Generalization
- Parsimony (efficiency)



plosive + vowel energies for 2 different gains

$$\frac{\partial}{\partial t}\log CE(t) = \frac{\partial}{\partial t}(\log C + \log E(t))$$

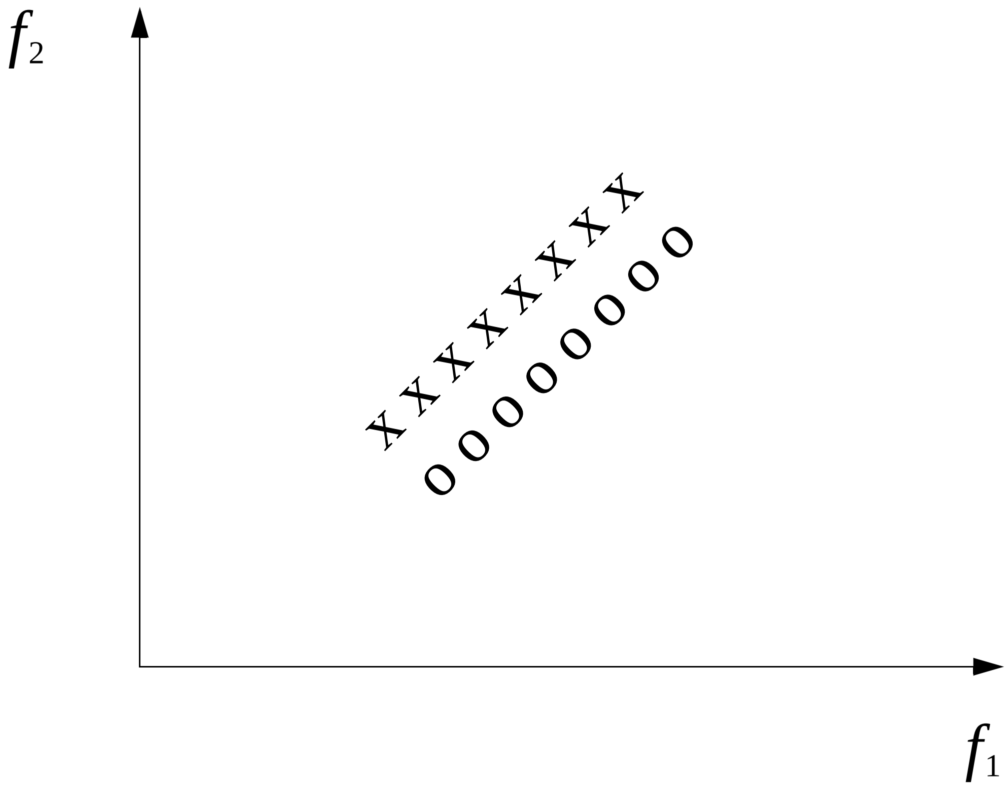
$$= \frac{\partial}{\partial t}\log E(t)$$

Feature Vector Size

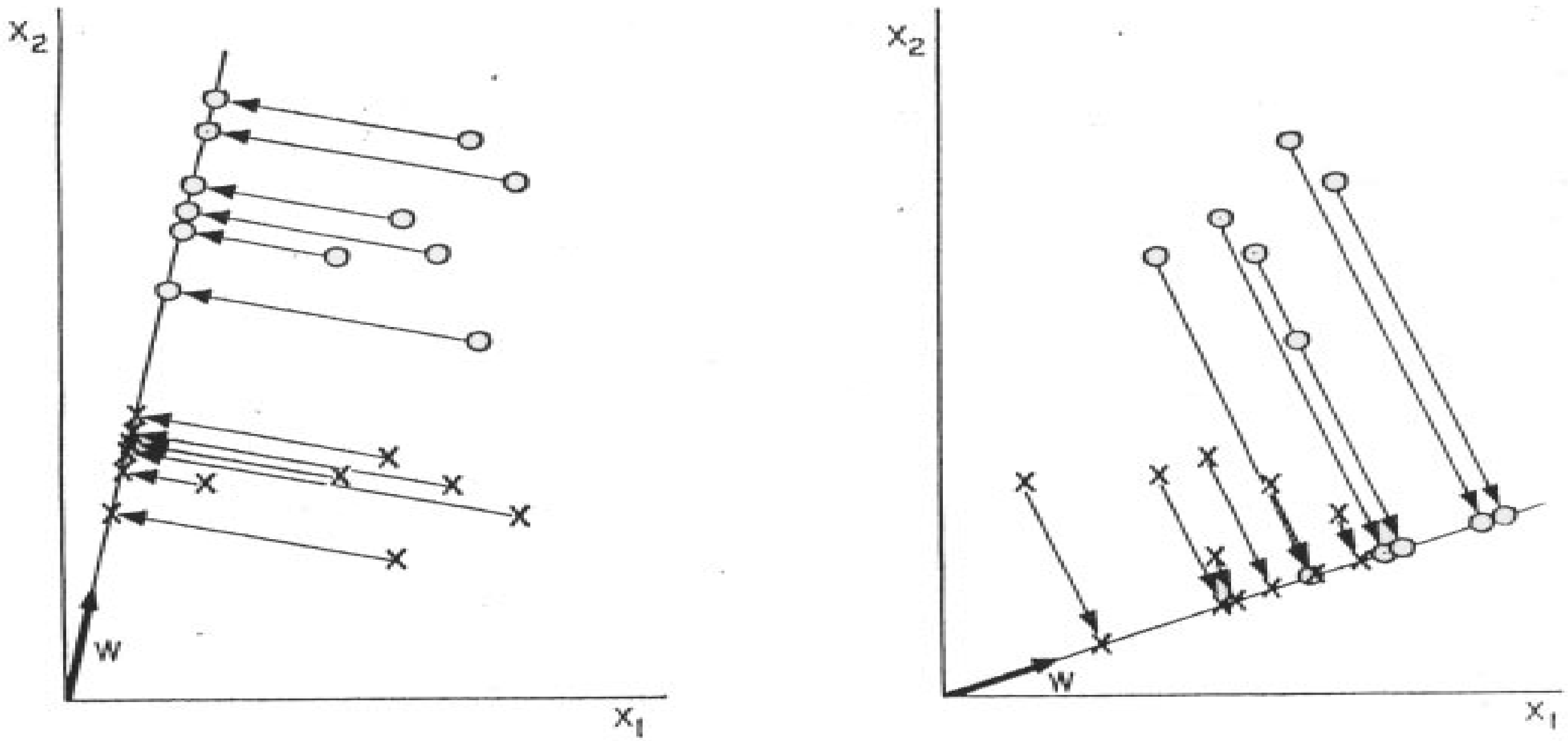
- Best representations for discrimination on training set are large (highly dimensioned)
- Best representations for generalization to test set are (typically) succinct)

Dimensionality Reduction

- Principal components (i.e., SVD, KL transform, eigenanalysis ...)
- Linear Discriminant Analysis (LDA)
- Application-specific knowledge
- Feature Selection via PR Evaluation


f_2
x x x x x x x x x
o o o o o o o o o
f_1

FISHER'S LINEAR DISCRIMINANT



Projection of samples onto a line.

PR Methods

- Minimum Distance
- Discriminant Functions
- Linear Discriminant
- Nonlinear Discriminant

  (e.g, quadratic, neural networks)
- Statistical Discriminant Functions

Minimum Distance

- Vector or matrix representing element
- Define a distance function
- Choose the class of stored element closest to new input
- Choice of distance equivalent to implicit statistical assumptions
- For speech, temporal variability complicates this

z_i = template vector (prototype)

x = input vector

Choose i to minimize distance

$$\arg_i \min \sqrt{(x - z_i)^T (x - z_i)} = \arg_i \min (x - z_i)^T (x - z_i) = \arg_i \min (x^T x + z_i^T z_i - 2x^T z_i)$$

$$\arg_i \max \left(\frac{z_i^T z_i - 2x^T z_i}{-2} \right) = \arg_i \max \left(x^T z_i - \frac{1}{2} z_i^T z_i \right)$$

$$\text{If } z_i^T z_i = 1 \text{ for all } i \Rightarrow \arg_i \max (x^T z_i)$$

Problems with Min Distance

- Proper scaling of dimensions (size, discrimination)
- For high dim, sparsely sampled space

Decision Rule for Min Distance

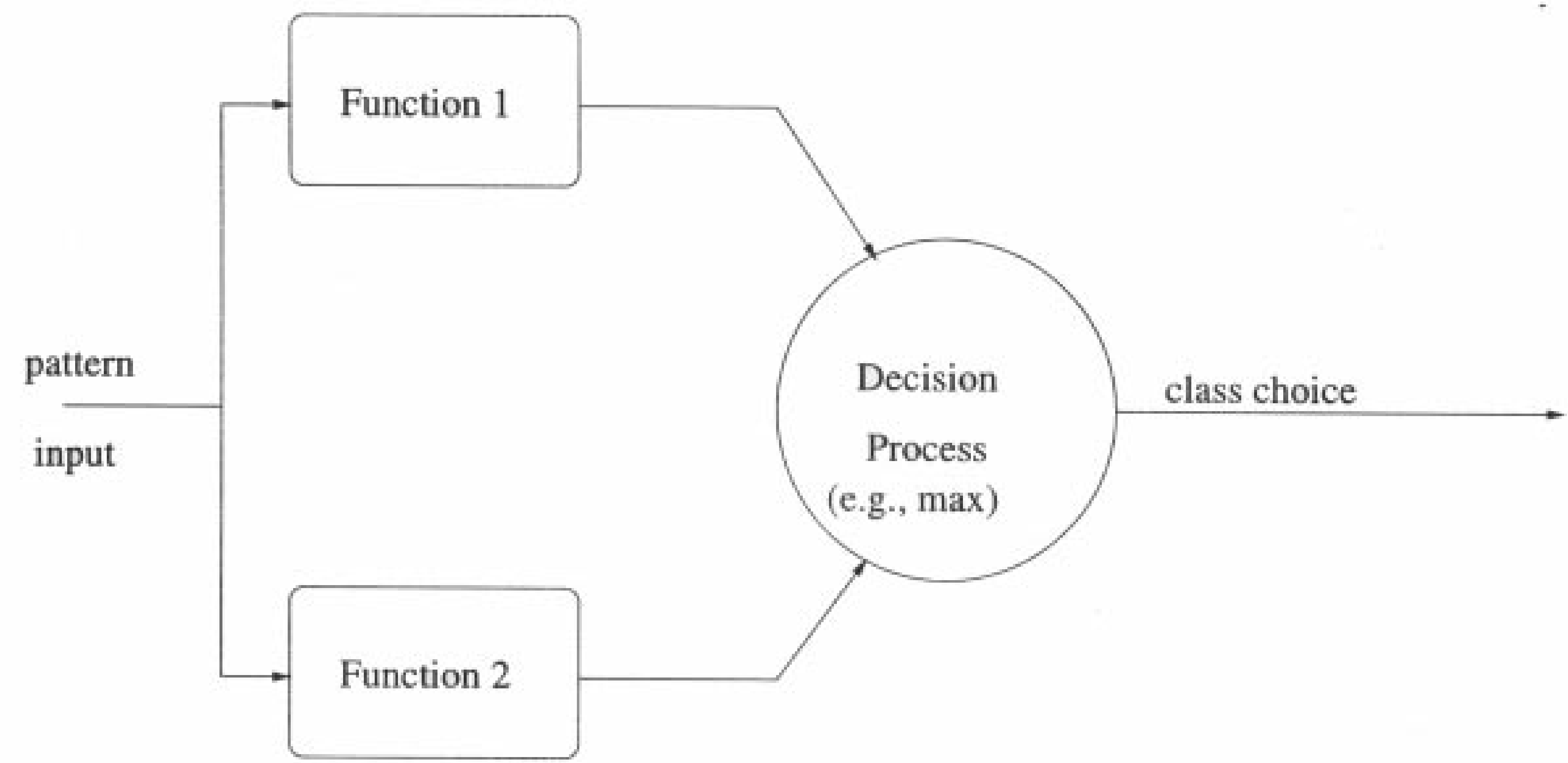
- Nearest Neighbor (NN) - in the limit of infinite samples, at most twice the error of optimum classifier
- k-Nearest Neighbor (kNN)
- Lots of storage for large problems; potentially large searches

Some Opinions

- Better to throw away bad data than to reduce its weight
- Dimensionality-reduction based on variance often a bad choice for supervised pattern recognition

Discriminant Analysis

- Discriminant functions max for correct class, min for others
- Decision surface between classes
- Linear decision surface for 2-dim is line, for 3 is plane; generally called hyperplane
- For 2 classes, surface at $\omega^T \boldsymbol{x} + \omega_0 = 0$
- 2-class quadratic case, surface at $\boldsymbol{x}^T \boldsymbol{W} \boldsymbol{x} + \omega^T \boldsymbol{x} + \omega_0 = 0$


Function 1
pattern
input
Decision
Process
(e.g., max)
class choice
Function 2

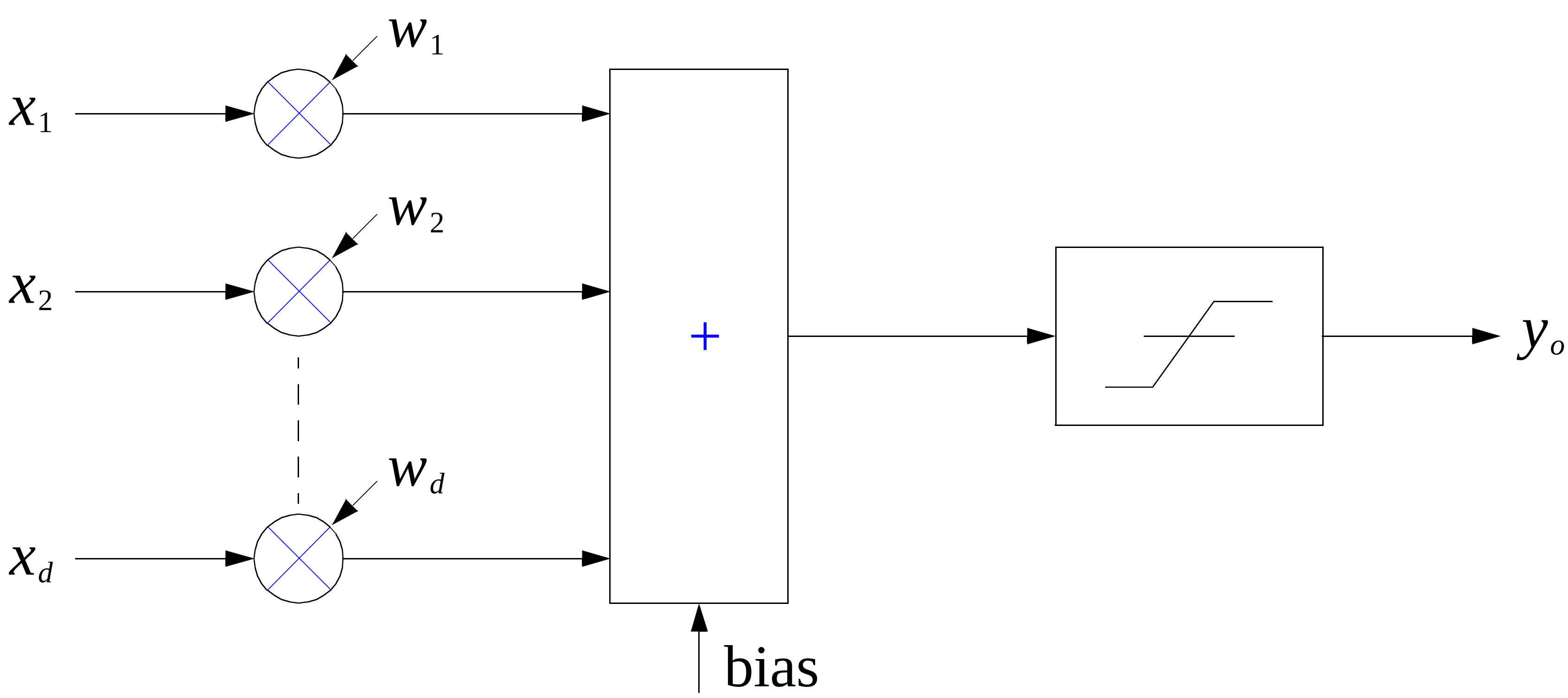
Training Discriminant Functions

- Minimum distance
- Fisher linear discriminant
- Gradient learning

Generalized Discriminators - ANNs

- McCulloch Pitts neural model
- Rosenblatt Perceptron
- Multilayer Systems

The Perceptron



McCulloch-Pitts Neuron - Rosenblatt Perceptron

Perceptron Convergence

If classes are linearly separable the following rule will converge in a finite number of steps :

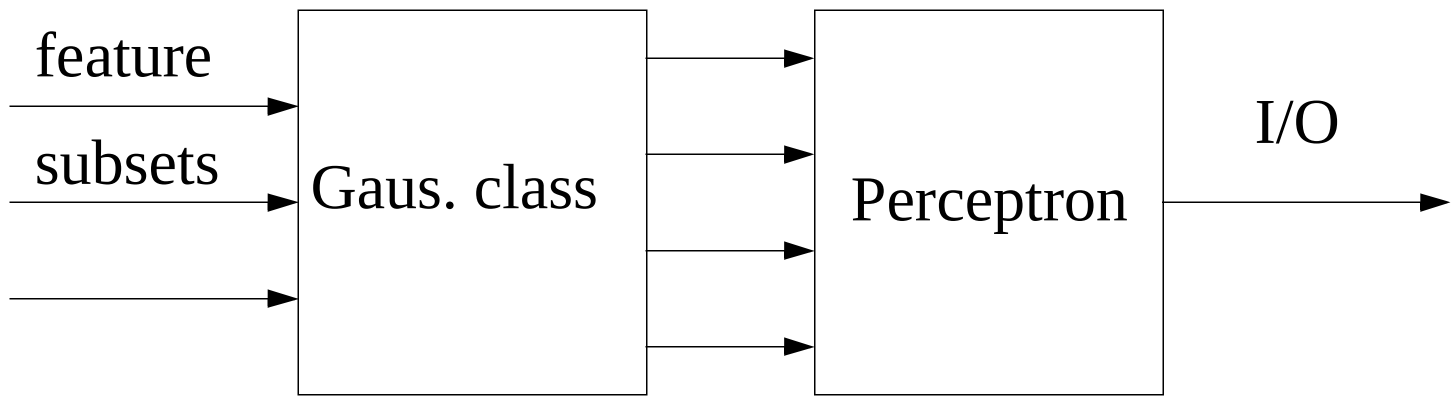
For each pattern *x* at time step *k*;

$$\text{if}\begin{cases} x(k) \in \text{ class 1, } \omega^T(k)x(k) \le 0 \\ \qquad \Rightarrow \omega(k+1) = \omega(k) + cx(k) \\ x(k) \in \text{ class 2, } \omega^T(k)x(k) \ge 0 \\ \qquad \Rightarrow \omega(k+1) = \omega(k) - cx(k) \end{cases}$$

$$\text{else}\begin{cases} \omega(k+1) = ""\omega(k) \end{cases}$$

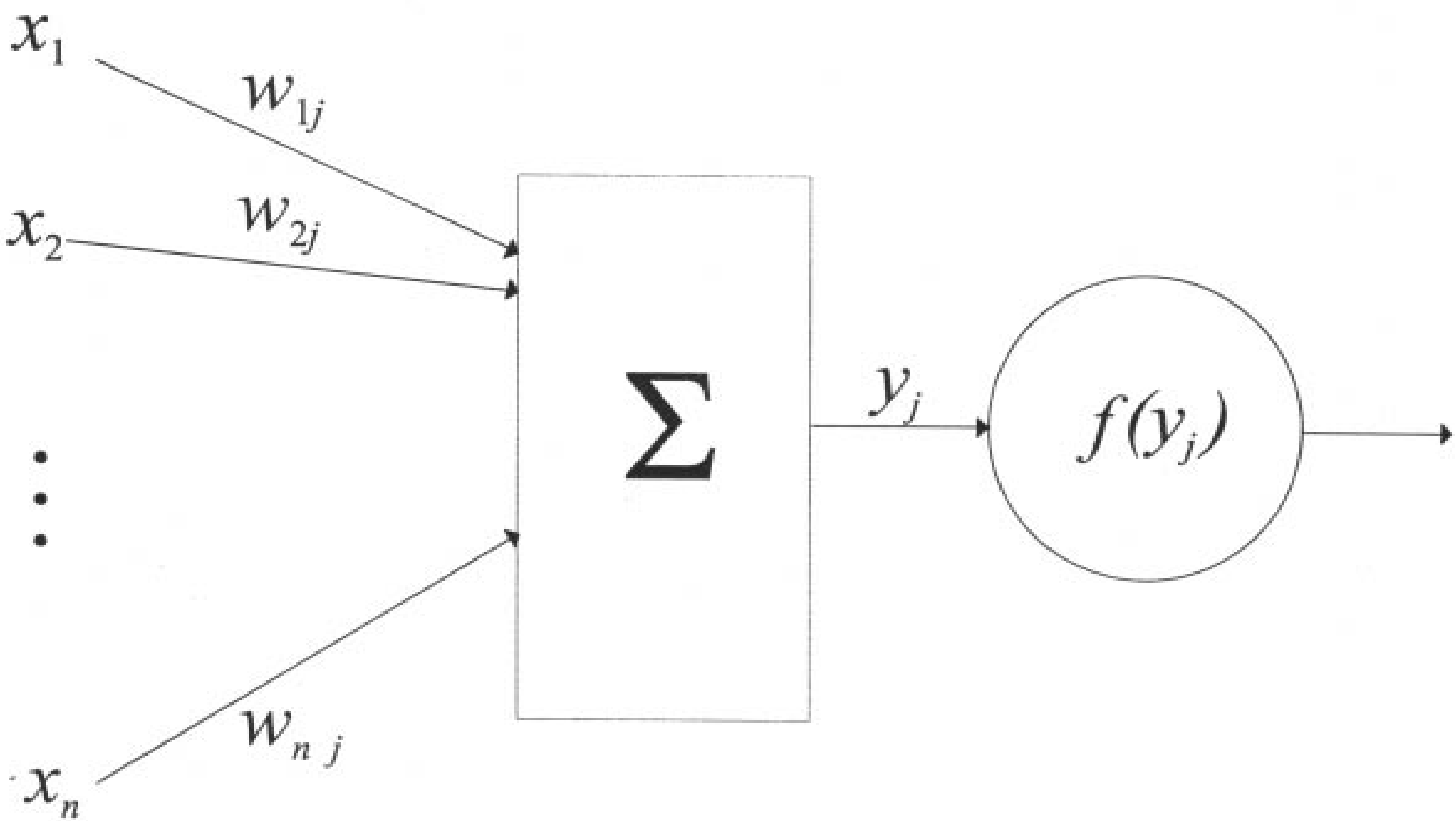
Multilayer Perceptrons

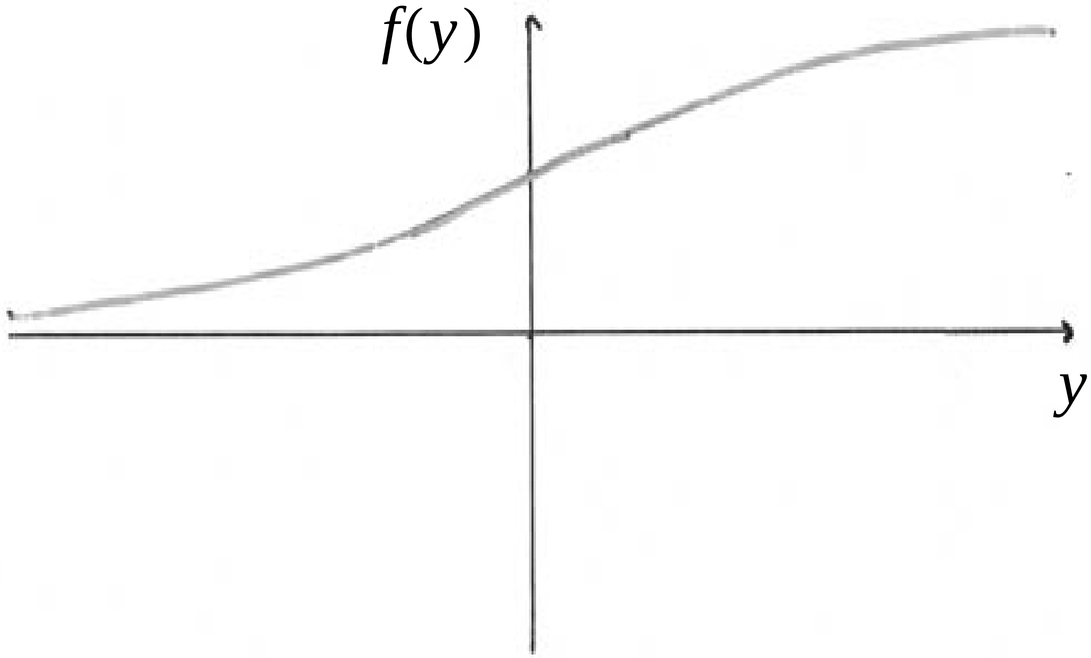
•Heterogeneous, “hard” nonlinearity :(DAID, 1961)



•Homogeneous, “soft” nonlinearity

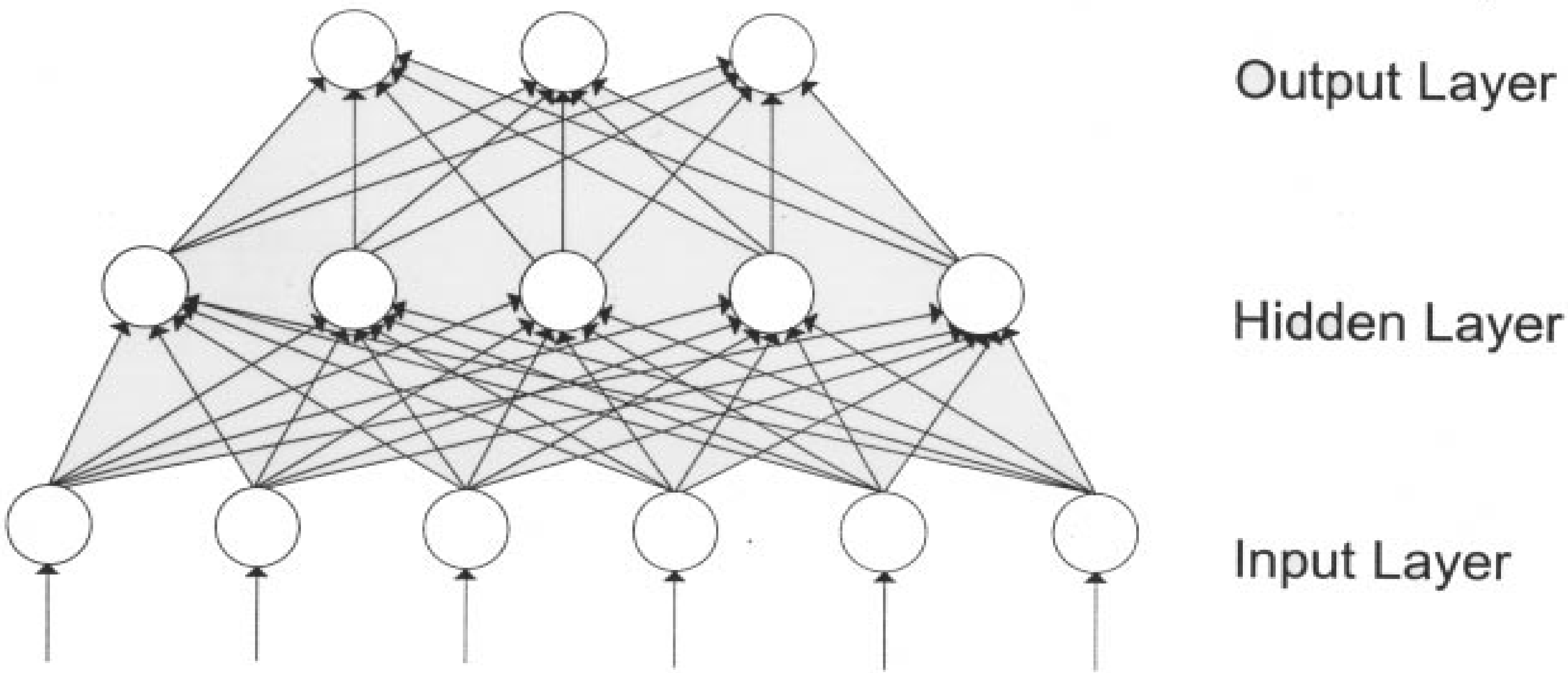
(“modern” MLP)


x_1
w_{1j}
x_2
w_{2j}
x_n
w_{nj}
Σ
y_j
$f(y_j)$



$$f(y) = \frac{1}{1 + e^{-y}} \quad \text{(sigmoid)}$$

$$0 < f(y) < 1$$


Output Layer
Hidden Layer
Input Layer

Some PR Issues

- Testing on the training set
- Training on the test set
- No. parameters vs no. training examples: overfitting and overtraining