Digital Image Processing
Chapter 12:
Object Recognition

Patterns and Pattern Classes

- A *pattern* is an arrangement of descriptors
- A *feature* : a descriptor of a pattern for recognition
- A *pattern class* : a family of patterns that share some common properties, denoted as $\omega_1, \omega_2, \ldots, \omega_W$, where W is the number of classes.
- Pattern recognition : assigning patterns to their respective classes.
- Pattern arrangements :
  1. *vectors*(for quantitative descriptions),
  2. *strings* or
  3. *trees*(for structural descriptions)

Object recognition

- Recognition of individual regions, which we call ***objects*** or ***patterns***.

1. Decision-theoretic approach
   - deals with patterns described using quantitative descriptors such as length, area and texture

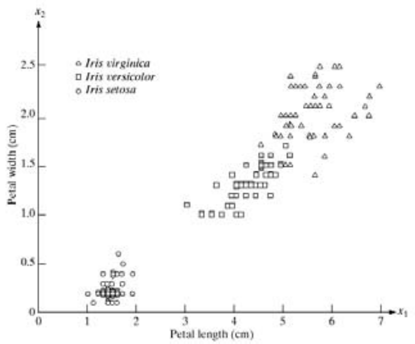
2. Structural approach
   - deals with patterns best- described by qualitative descriptors such as the relational descriptors

- Concept of "***learning***" from sample patterns.

Discriminant analysis[1936, Fisher]

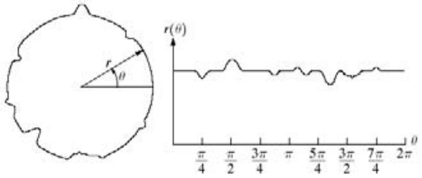
- To recognize 3 types of *iris* flowers by measuring the widths and lengths of their petals.
- Each flower is described by two measurements in 2-D pattern vector of $\mathbf{x} = [\, x_1, x_2 \,]^T$

FIGURE 12.1 Three types of iris flowers described by two measurements.



Feature Selection Problem

- The degree of *class separability* depends strongly on the choice of descriptors selected for an application.
- Fig. 12.2 : Different types of noisy shapes
  - Quantitative information
  - Represent each object by its signature to obtain 1-D signals
    - $x_1=r_1(\theta_1),\ x_2=r_2(\theta_2),\ \ldots,\ x_n=r_n(\theta_n)$
  - Or use the first ***n*** statistical moments as descriptors of each pattern vector.

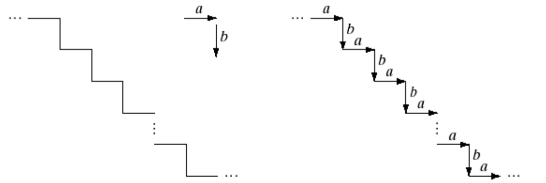


a b

FIGURE 12.2 A noisy object and its corresponding signature.

Pattern characteristics by structural relationships

String descriptions adequately generate patterns of objects and other entities whose structure is based on relatively simple connectivity of primitives, usually associated with boundary shape.



a b

FIGURE 12.3 (a) Staircase structure. (b) Structure coded in terms of the primitives *a* and *b* to yield the string description . . . *ababab* . . . .

Pattern characteristics by structural relationships

- Fingerprint recognition is based on the interrelationships of *print features* called, *minutiae.*
  1. relative sizes and locations,
  2. primitive components that describe fingerprint ridge properties, such as
     - abrupt endings, branching, merging, and disconnected segments.

Usually, minutiae. precise details; small or trifling matters: the minutiae of his craft.

Tree descriptions

Most hierarchical ordering schemes lead to tree structures

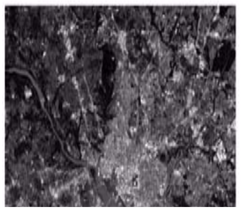


FIGURE 12.4 Satellite image of a heavily built downtown area (Washington, D.C.) and surrounding residential areas. (Courtesy of NASA.)

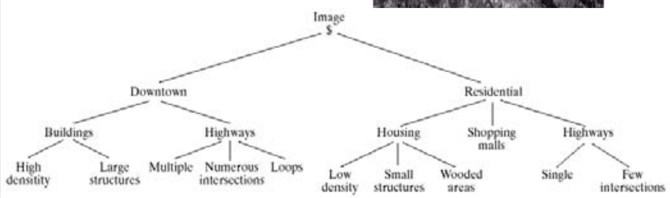


FIGURE 12.5 A tree description of the image in Fig. 12.4.

Recognition based on decision-theoretic methods

Based on the use of *decision* (or *discriminant*) *functions* :
Let $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ of an n-dimensional pattern vector.

For W pattern classes $\omega_1, \omega_2, \ldots, \omega_W$, the basic problem in decision-theoretic pattern recognition is to find W *decision functions* $d_1(\mathbf{x}), d_2(\mathbf{x}), \ldots, d_W(\mathbf{x})$, with the property that, if a pattern $\mathbf{x}$ belongs to class ω_i then,

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \quad j = 1, 2, \ldots, W, j \neq i$$

The *decision boundary* separating class ω_i, from ω_j is given by values of $\mathbf{x}$ for which $d_i(\mathbf{x}) = d_j(\mathbf{x})$

Develop various approaches for finding decision functions as shown above.

Minimum Distance Classifier

The *prototype* of each pattern : mean vector of the patterns of that class

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in \omega_j} \mathbf{x}$$

Euclidean distance $D_j(\mathbf{x}) = \|\mathbf{x} - \mathbf{m}_j\|$

Assign $\mathbf{x}$ to class ω_i if $D_i(\mathbf{x})$ is the smallest distance.

Selecting the smallest distance is equivalent to evaluating the functions

$$d_j(\mathbf{x}) = \mathbf{x}^T \mathbf{m}_j - ½\, \mathbf{m}_j^T \mathbf{m}_j$$

And assigning $\mathbf{x}$ to class ω_i if $d_i(\mathbf{x})$ yields the largest numerical value.

Matching

Matching represent each class by a *prototype pattern vector*.

An unknown pattern is assigned to the class to which it is closest in terms of a predefined metric.

1. Minimum distance classifier
2. Approaches based on correlation

Decision Boundary

$$d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$$

The surface given by this eq. is the *perpendicular bisector* of the line segment joining $\mathbf{m}_i$ and $\mathbf{m}_j$.

For n = 2, the perpendicular bisector is a *line*,
for n = 3 it is a *plane*, and
for n > 3 it is called a *hyperplane*.

Iris Classification by Matching

$\mathbf{m}_1 = (4.3, 1.3)^T$, $\mathbf{m}_2 = (1.5, 0.3)^T$
$d_1(\mathbf{x}) = \mathbf{x}^T \mathbf{m}_1 - ½\, \mathbf{m}_1^T \mathbf{m}_1 = 4.3x_1 + 1.3x_2 - 10.1$
$d_2(\mathbf{x}) = \mathbf{x}^T \mathbf{m}_2 - ½\, \mathbf{m}_2^T \mathbf{m}_2 = 1.5x_1 + 0.3x_2 - 1.17$

The equation of boundary is

$$d_{12}(\mathbf{x}) = d_1(\mathbf{x}) - d_2(\mathbf{x}) = 2.8x_1 + 1.0x_2 - 1.17$$

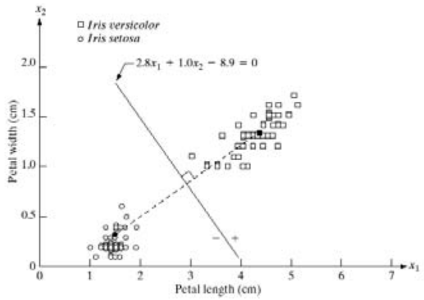


FIGURE 12.6 Decision boundary of minimum distance classifier for the classes of *Iris versicolor* and *Iris setosa*. The dark dot and square are the means.

Minimum Distance Classifier

The segmentation problem is solved by artificially *highlighting* the *key characteristics* of each character.
The design of the font ensures that the waveform of each character is *distinct* from that of all others.

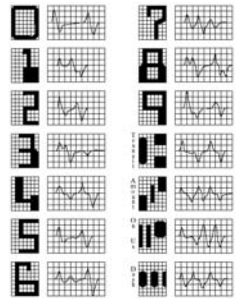


FIGURE 12.7 American Bankers Association E-13B font character set and corresponding waveforms.

Minimum Distance Classifier

Works well when *the distance between means* is large compared to the spread or randomness of each class with respect to its mean.

The minimum distance classifier yields ***optimum*** performance (in terms of minimizing *the average loss of misclassification*) when the distribution of each class about its mean is in the form of a spherical "***hypercloud***" in *n*-dimensional pattern space.

The simultaneous occurrence of *large mean separations* and *relatively small class spread* occur seldomly in practice unless the system designer controls the nature of the input.

Matching by correlation

The correlation between *f(x,y)* and a subimage *w(x,y)* is

$$c(x,y) = \sum_s \sum_t f(s,t)\, w(x+s, y+t)$$

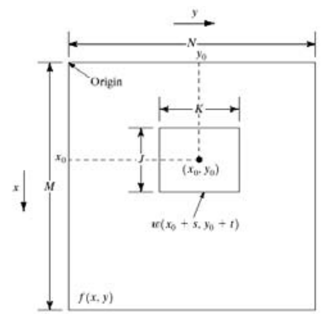
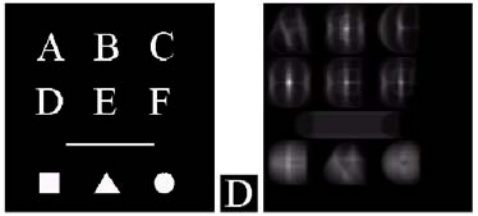


FIGURE 12.8 Arrangement for obtaining the correlation of f and w at point (x_0, y_0).

Matching by correlation

- Normalization for *amplitude changes* : use correlation coefficient
- Normalization for changes in *size* and *rotation* is difficult

- Correlation in frequency domain is possible.

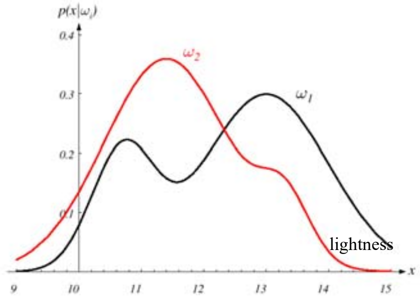


a b c
FIGURE 12.9
(a) Image.
(b) Subimage.
(c) Correlation coefficient of (a) and (b). Note that the highest (brighter) point in (c) occurs when subimage (b) is coincident with the letter "D" in (a).

Class Conditional Probabilities

$P(x \mid \omega_2)$: Class conditional probability for salmon
Likelihood: Given that salmon has been observed, what is the probability of this salmon's lightness is between 11 and 12?

ω_1: Sea bass
ω_2: Salmon



Optimum Statistical Classifier

A probabilistic approach to recognition.
$p_i(\boldsymbol{\omega}_i/\mathbf{x})$: the probability that a particular pattern $\mathbf{x}$ came from class $\boldsymbol{\omega}_i$.
If the pattern classifier decides that $\mathbf{x}$ came from class $\boldsymbol{\omega}_j$, when it actually came from class $\boldsymbol{\omega}_i$, it incurs a loss, denoted $\boldsymbol{L_{ij}}$.
An average loss incurred in assigning $\mathbf{x}$ to $\boldsymbol{\omega}_j$ is

$$r_j(\mathbf{x}) = \sum_{k=1}^{W} L_{kj}\, p(\omega_k / \mathbf{x})$$

, the *conditional average risk* or *loss*.

Since $p(A/B) = [p(A)p(B/A)]/p(B)$,

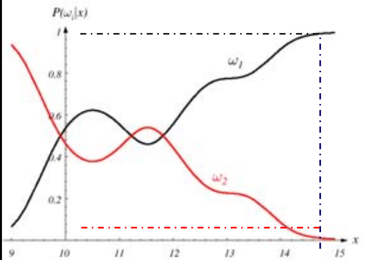
$$r_j(\mathbf{x}) = \frac{1}{p(\mathbf{x})} \sum_{k=1}^{W} L_{kj}\, p(\mathbf{x}/\omega_k) P(\omega_k)$$

Definitions & Bayes Decision Rule

$$P(\omega_j \mid x) = \frac{p(x \mid \omega_j) \cdot P(\omega_j)}{\sum_{x \in X} p(x \mid \omega_j) \cdot P(\omega_j)} = \frac{p(x \mid \omega_j) \cdot P(\omega_j)}{p(x)}$$

Posterior Probabilities

- Bayes rule allows us to compute the *posterior* probability (difficult to determine) from *prior* probabilities, likelihood and the evidence (easier to determine).



Posterior probabilities for *priors* $P(\omega_1) = 2/3$ and $P(\omega_2) = 1/3$.

For example, given that a pattern is measured to have feature value $x = 14$,

the probability it is in category ω_2 is roughly 0.08, and

that it is in ω_1 is 0.92.

At every x, the *posteriors* sum to 1.0.

Bayes Classifier

The "loss" for a correct decision generally is assigned a value of ***zero***, and the loss for any incorrect decision usually is assigned the same ***nonzero*** value (say, 1). Then the loss function is

$$L_{ij} = 1 - \delta_{ij}, \text{ where } \delta_{ij} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases}$$

$$r_j(\mathbf{x}) = \sum_{k=1}^{W} (1 - \delta_{kj}) p(\mathbf{x}/\omega_k) P(\omega_k) = p(\mathbf{x}) - p(\mathbf{x}/\omega_j) P(\omega_j)$$

The Bayes classifier then assigns a pattern $\mathbf{x}$ to class $\boldsymbol{\omega}_i$ if $p(\mathbf{x}/\omega_i)P(\omega_i) > p(\mathbf{x}/\omega_j)P(\omega_j)$

Bayes Classifier

For the relative order only, $1/p(\mathbf{x})$ can be dropped as

$$r_j(\mathbf{x}) = \sum_{k=1}^{W} L_{kj}\, p(\mathbf{x}/\omega_k) P(\omega_k)$$

If a classifer computes $r_1(\mathbf{x})$, $r_2(\mathbf{x})$, …, $r_W(\mathbf{x})$ for each pattern $\mathbf{x}$ and assigns the pattern to the class with the smallest loss, the total average loss with respect to all decisions will be minimum. The classifier that minimizes the total average loss is called the ***Bayes classifier***.

Thus the Bayes classifier assigns an unknown pattern $\mathbf{x}$ to class $\boldsymbol{\omega}_i$ if $r_i(\mathbf{x}) < r_j(\mathbf{x})$, for all $j \neq i$.

Assumptions for Bayes Classifier

The Bayes classifier is *0-1 loss function*, where a pattern vector $\mathbf{x}$ is assigned to the class whose decision function yields the largest numerical value of $d_j(\mathbf{x}) = p(\mathbf{x}/\omega_j)P(\omega_j)$

- $p(\mathbf{x} \mid \omega_j)$: The probability density functions of the patterns in each class
- $P(\omega_j)$: The probability of occurrence of each class.

If the pattern vectors, $\mathbf{x}$, are n dimensional, then $p(\mathbf{x}/\boldsymbol{\omega}_i)$ is a function of n variables, which, if its form is not known, requires methods from multivariate probability theory for its estimation.

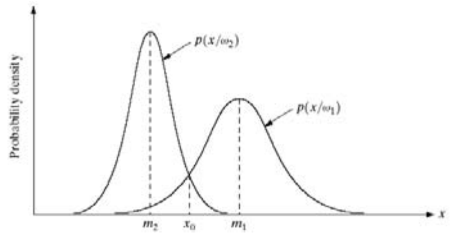
Assume an analytic expression for the various density functions and an estimation of the necessary parameters from sample patterns from each class.

Bayes Classifier for Gaussian Pattern Classes

The Bayes decision function is of the form,

$$d_j(\mathbf{x}) = p(\mathbf{x}/\omega_j)P(\omega_j)$$
$$= \frac{1}{\sqrt{2\pi}\sigma_j} e^{-gfds\frac{(x-m_j)^2}{2\sigma_j^2}} P(\omega_j),\ \ j=1,2$$

FIGURE 12.10 Probability density functions for two 1-D pattern classes. The point x_0 shown is the decision boundary if the two classes are equally likely to occur.



N-dimensional Case

The *Bayes decision function* is of the form, where each density is specified completely by its mean vector $\mathbf{m}_j$ and covariance matrix $\mathbf{C}_j$, which are defined as

$$\mathbf{m}_j = E_j\{\mathbf{x}\} \text{ and } \mathbf{C}_j = E_j\{(\mathbf{x}-\mathbf{m}_j)(\mathbf{x}-\mathbf{m}_j)^T\},$$

Where E_j {} denotes rhe expected value of the argument over the patterns of class ω_j.

$$p(\mathbf{x}/\omega_j)$$
$$= \frac{1}{(2\pi)^{n/2}|\mathbf{C}_j|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_j)^T\mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)}$$

N-dimensional Case

Approximating the expected value E_j by the average value of the quantities in question yields

$$\mathbf{m}_j = \frac{1}{N_j}\sum_{\mathbf{x}\in\omega_j}\mathbf{x} \quad \text{and} \quad \mathbf{C}_j = \frac{1}{N_j}\sum_{\mathbf{x}\in\omega_j}\mathbf{x}\mathbf{x}^T - \mathbf{m}_j\mathbf{m}_j^T$$

From $d_j(\mathbf{x}) = p(\mathbf{x}/\omega_j)\ P(\omega_j)$, we can use the term,

$d_j(\mathbf{x}) \approx \ln[p(\mathbf{x}/\omega_j)\ P(\omega_j)]$
$= \ln[p(\mathbf{x}/\omega_j)]+\ln[P(\omega_j)]$
$= \ln[P(\omega_j)] - n/2\ \ln[2\pi] - 1/2\ \ln|\mathbf{C}_j| - 1/2[(\mathbf{x}-\mathbf{m}_j)^T\mathbf{C}^{-1}_j(\mathbf{x}-\mathbf{m}_j)]$

The *Bayes decision functions* for Gaussian pattern classes under the condition of a *0-1 loss function* : **hyperquadrics** (*quadratic functions* in *n*-dimensional space), because no terms higher than the second degree in the components of **x** appear in the equation.

N-dimensional Case

If $\mathbf{C} = \mathbf{I}$, and $P(\omega_j)=1/W$, then

$$d_j(\mathbf{x}) = \mathbf{x}^T\ \mathbf{m}_j| - 1/2\ \mathbf{m}_j^T\mathbf{m}_j$$

These are the decision functions for a minimum distance classifier.

Thus the minimum distance classifier is **optimum** in the *Bayes sense* if

1. the pattern classes arc *Gaussian*,
2. all $\mathbf{C} = \mathbf{I}$
3. $P(\omega_j)=1/W$

Hyperspheres

• Gaussian pattern classes satisfying these conditions are *spherical clouds* of identical shape in *n*-dimensions(called *hyperspheres*).

• The *minimum distance classifier* established a *hyperplane* between every pair of classes, with the property that the *hyperplane* is the *perpendicular bisector* of the line segment joining the center of the pair of hyperspheres.

• In 2-D, the classes constitute *circular regions* and the boundaries become *lines* that bisect the line segment joining the center of every pair of such circles.

Remotely Sensed Imagery

The classification of *remotely sensed imagery* generated by multispectral scanners aboard aircraft, satellites, or space stations.

The *voluminous image data* generated by these platforms make automatic image classification and analysis a task of considerable interest in remote sensing.

The applications of remote sensing are varied and include land use, crop inventory, crop disease detection, forestry, air and water quality monitoring, geological studies, weather prediction, and a score of other applications having environmental significance.

Two Pattern Classes in 3-D

$$\mathbf{C}_1 = \mathbf{C}_2 = \frac{1}{16}\begin{bmatrix} 3 & 1 & 1 \\ 1 & 3 & -1 \\ 1 & -1 & 3 \end{bmatrix}$$

$$\mathbf{C}^{-1} = \begin{bmatrix} 8 & -4 & -4 \\ -4 & 8 & 4 \\ -4 & 4 & 8 \end{bmatrix}$$

$d_j(\mathbf{x}) = \mathbf{x}^T \mathbf{m}_j - 1/2\, \mathbf{m}_j^T \mathbf{m}_j$

$d_1(\mathbf{x}) = 4x_1 - 1.5$

$d_2(\mathbf{x}) = -4x_1 + 8x_2 + 8x_3 - 5.5$

$d_1(\mathbf{x}) - d_2(\mathbf{x}) = 8x_1 - 8x_2 - 8x_3 + 4 = 0$

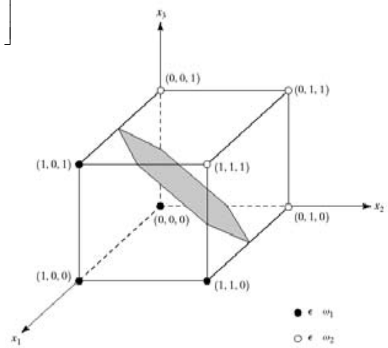
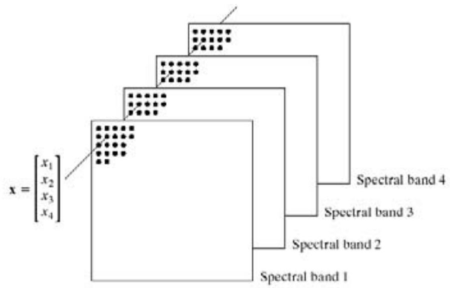


FIGURE 12.11 Two simple pattern classes and their Bayes decision boundary (shown shaded).

Multispectral Image : violet, green, red, infrared

If the images are of size 512 X 512 pixels, each stack of four multispectral images can he represented by 262,144 4-D pattern vectors.

FIGURE 12.12 Formation of a pattern vector from registered pixels of four digital images generated by a multispectral scanner.



Bayes Classifier for remote sensing applications

Bayes classifier for *Gaussian patterns* requires estimation of the mean vector and covariance matrix for each class.

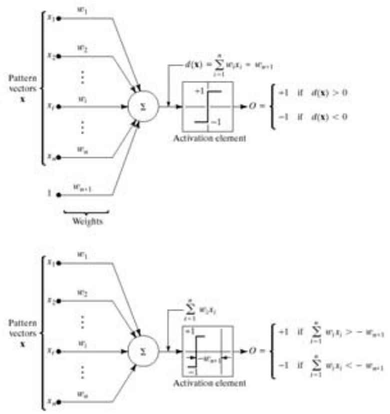
In remote sensing applications, these are obtained by collecting multispectral data for each region of interest and then using these samples.

Pixel-by-pixel classification of an image actually segments the image into various classes.

This approach is like segmentation by thresholding with several variables,

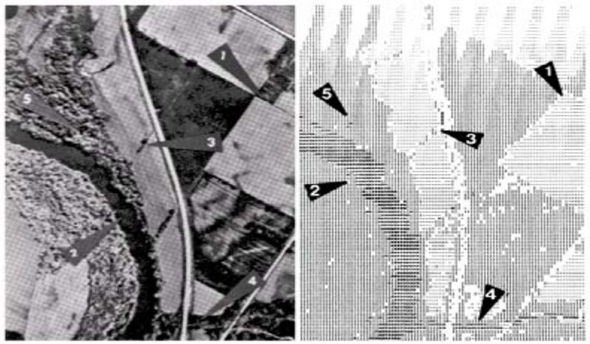
Neural Networks : Perceptron for 2 pattern classes

a
b
FIGURE 12.14 Two equivalent representations of the perceptron model for two pattern classes.



Remote sensing applications

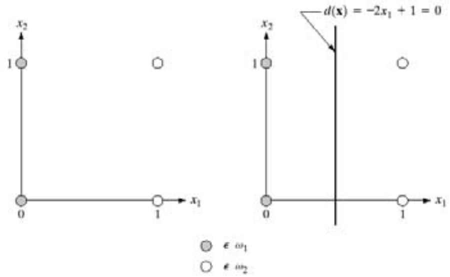
Classify areas such as vegetation, water, and bare soil.



a b
FIGURE 12.13 (a) Multispectral image. (b) Printout of machine classification results using a Bayes classifier. (Courtesy of the Laboratory for Applications of Remote Sensing, Purdue University.)

Training Algorithms

a b
FIGURE 12.15 (a) Patterns belonging to two classes. (b) Decision boundary determined by training.



Multilayer Feedforward Neural Networks

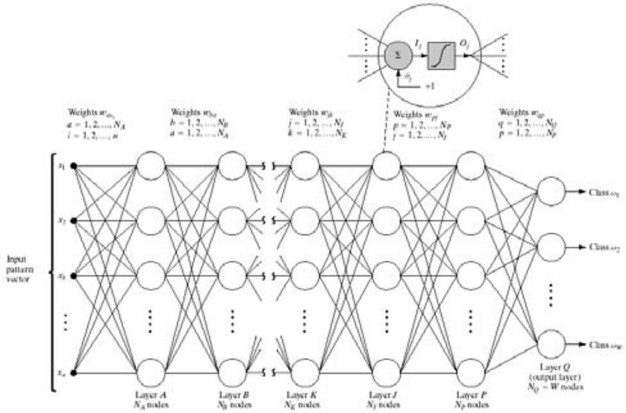
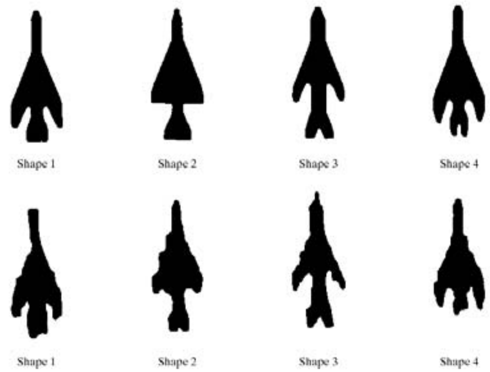


FIGURE 12.16 Multilayer feedforward neural network model. The blowup shows the basic structure of each neuron element throughout the network. The offset, θ_j, is treated as just another weight.

An application

a
b

FIGURE 12.18
(a) Reference shapes and
(b) typical noisy shapes used in training the neural network of Fig. 12.19.
(Courtesy of Dr. Lalit Gupta, ECE Department, Southern Illinois University.)



The Sigmoid Activation Function

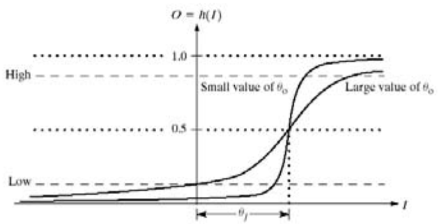


FIGURE 12.17 The sigmoidal activation function of Eq. (12.2-47).

3-Layer Neural Network for Recognizing Shapes

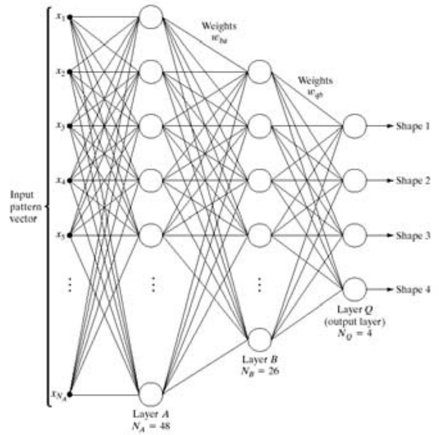
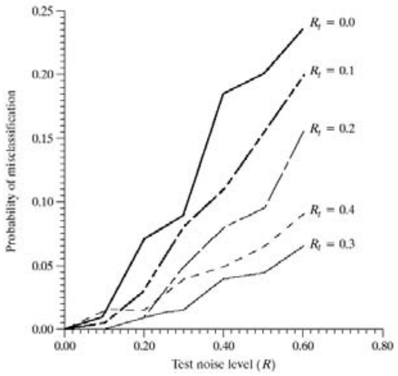


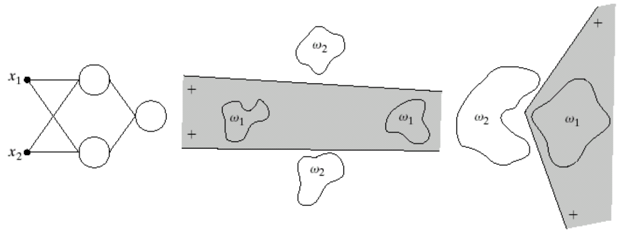
FIGURE 12.19 Three-layer neural network used to recognize the shapes in Fig. 12.18. (Courtesy of Dr. Lalit Gupta, ECE Department, Southern Illinois University.)

Performance

FIGURE 12.20 Performance of the neural network as a function of noise level. (Courtesy of Dr. Lalit Gupta, ECE Department, Southern Illinois University.)



Decision Boundaries



a b c

FIGURE 12.22 (a) A two-input, two-layer, feedforward neural network. (b) and (c) Examples of decision boundaries that can be implemented with this network.

Improved Performance with more training patterns

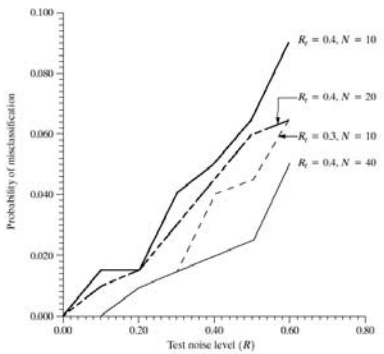


FIGURE 12.21 Improvement in performance for $R_t = 0.4$ by increasing the number of training patterns (the curve for $R_t = 0.3$ is shown for reference). (Courtesy of Dr. Lalit Gupta, ECE Department, Southern Illinois University.)

Decision Regions

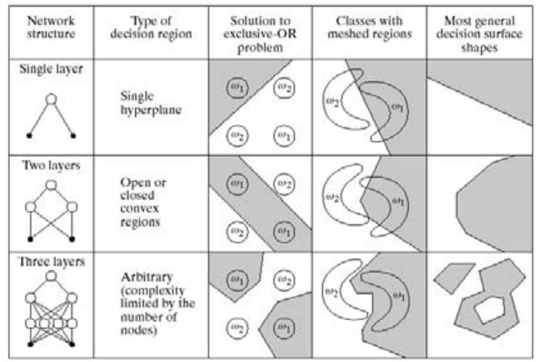
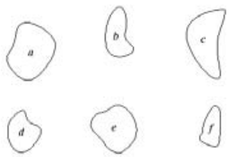
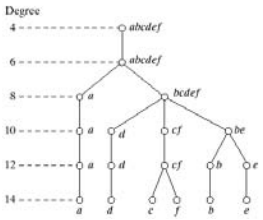


FIGURE 12.23 Types of decision regions that can be formed by single- and multilayer feed-forward networks with one and two layers of hidden units and two inputs. (Lippman)

Structural Methods



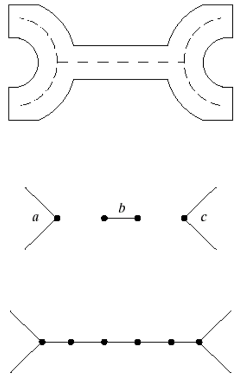


	a	b	c	d	e	f
a	–	6	6	6	6	6
b		–	8	8	10	8
c			–	8	8	12
d				–	8	8
e					–	8
f						–

a
b c

FIGURE 12.24 (a) Shapes. (b) Hypothetical similarity tree. (c) Similarity matrix. (Bribiesca and Guzman.)

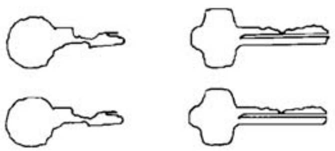
Syntactic Recognition of Strings



a
b
c

FIGURE 12.26 (a) Object represented by its (pruned) skeleton. (b) Primitives. (c) Structure generated by using a regular string grammar.

String Matching



R	1.a	1.b	1.c	1.d	1.e	1.f
1.a	–					
1.b	16.0	–				
1.c	9.6	26.3	–			
1.d	5.1	8.1	10.3	–		
1.e	4.7	7.2	10.3	14.2	–	
1.f	4.7	7.2	10.3	8.4	23.7	–

R	2.a	2.b	2.c	2.d	2.e	2.f
2.a	–					
2.b	33.5	–				
2.c	4.8	5.8	–			
2.d	3.6	4.2	19.3	–		
2.e	2.8	3.3	9.2	18.3	–	
2.f	2.6	3.0	7.7	13.5	27.0	–

R	1.a	1.b	1.c	1.d	1.e	1.f
2.a	1.24	1.50	1.32	1.47	1.55	1.48
2.b	1.18	1.43	1.32	1.47	1.55	1.48
2.c	1.02	1.18	1.19	1.32	1.39	1.48
2.d	1.02	1.18	1.19	1.32	1.29	1.40
2.e	0.93	1.07	1.08	1.19	1.24	1.25
2.f	0.89	1.02	1.02	1.24	1.22	1.18

a b
c d
e f
g

FIGURE 12.25 (a) and (b) Sample boundaries of two different object classes; (c) and (d) their corresponding polygonal approximations; (e)–(g) tabulations of *R*. (Sze and Yang.)

Production rules and semantic information

TABLE 12.1 Example of semantic information attached to production rules.

Production	Semantic Information
$S \rightarrow aA$	Connections to a are made only at the dot. The direction of a, denoted θ, is given by the direction of the perpendicular bisector of the line joining the end points of the two undotted segments. The line segments are 3 cm each.
$A \rightarrow bA$	Connections to b are made only at the dots. No multiple connections are allowed. The direction of b must be the same as the direction of a. The length of b is 0.25 cm. This production cannot be applied more than 10 times.
$A \rightarrow bB$	The direction of a and b must be the same. Connections must be simple and made only at the dots.
$B \rightarrow c$	The direction of c and a must be the same. Connections must be simple and made only at the dots.

Finite Automaton

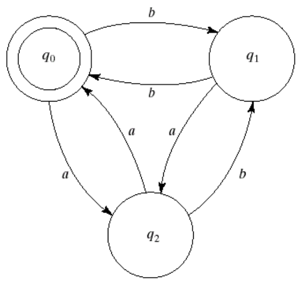
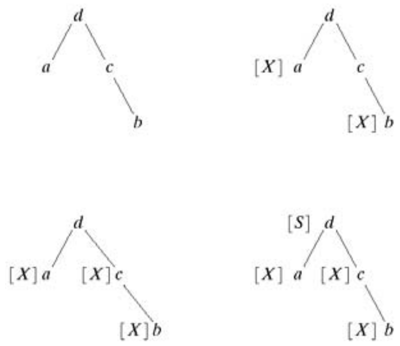


FIGURE 12.27 A finite automaton.

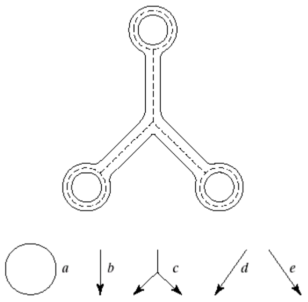
Tree Automaton



a b
c d

FIGURE 12.29 Processing stages of a frontier-to-root tree automaton: (a) Input tree. (b) State assignment to frontier nodes. (c) State assignment to intermediate nodes. (d) State assignment to root node.

Skeleton by a tree grammar

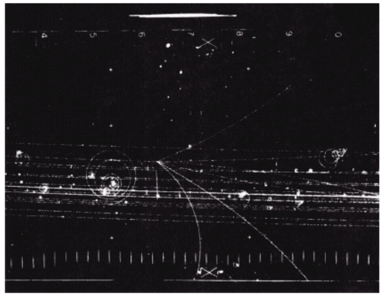


a
b

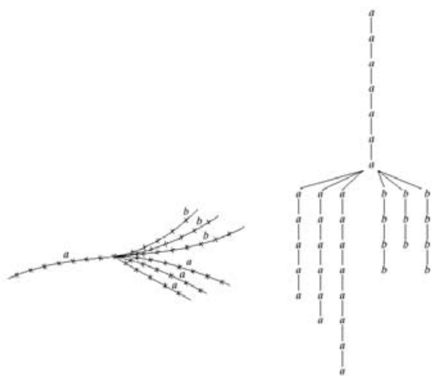
FIGURE 12.28 (a) An object and (b) primitives used for representing the skeleton by means of a tree grammar.

Bubble Chamber

FIGURE 12.30 A bubble chamber photograph. (Fu and Bhargava.)



Coded Event

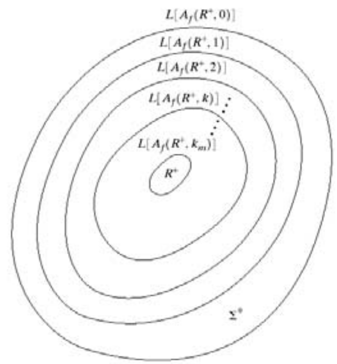


a b

FIGURE 12.31 (a) Coded event from Fig. 12.30. (b) Corresponding tree representation. (Fu and Bhargava.)

Relationships between Language Classes

FIGURE 12.33 Relationship between $L[A_f(R^+, k)]$ and k. The value of k_m is such that $k_m \geq$ (length of the longest string in R^+).



Finite Automaton

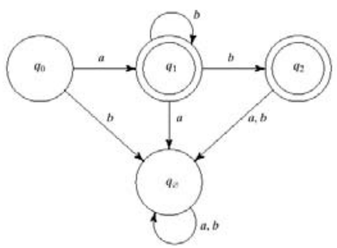


FIGURE 12.32 State diagram for the finite automaton inferred from the sample set $R^+ = \{a, ab, abb\}$.

State Diagram

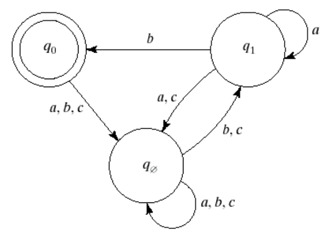


FIGURE 12.34 State diagram for the automaton $A_f(R^+, 1)$ inferred from the sample set $R^+ = \{caaab, bbaab, caab, bbab, cab, bbb, cb\}$.

Final Assignment – due 6/19

- Put the solution file into the class web hard.

1. Follow instructions for Fig. 12.2 in the textbook. What happens if you use 1, 2, or 3 samples instead of 4 samples? Summarize what happened in the procedure and analyze the result.

2. Find out sample images which contain human faces(multiple if available) and apply *Bayes classification* to find out the face regions in the picture. You have to try different image representations including
   - RGB
   - HSV
   - YCbCr

3. Repeat problem 2 for text images. Try to segment text regions with different representations including RGB, HSV, YCbCr. You should present results for at least a simple one and a somewhat nontrivial one.