Chap 7

Image Segmentation

Image Segmentation

- Edge-Based Segmentation
  - The edge information is used to determine boundaries of objects
- Pixel-based direct classification methods
  - The estimation methods derived from the histogram statistics of the image are used
- Region-based methods
  - Pixels are analyzed directly for a region growing process based on a pre-defined similarity criterion

Edge-based image segmentation
-Edge Detection Operations

- The gradient magnitude and directional information can be obtained by convolving the edge detection masks with the image

$Sobel\ operation$

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$M = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$$

$Second-order\ \ gradient\ \ (Laplacian)$

$$G_{L(4)} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$G_{L(8)} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

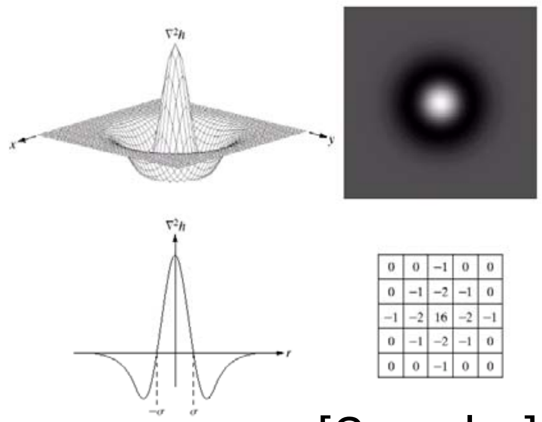
Edge-based image segmentation
-Edge Detection Operations

- Laplacian is very sensitive to noise
  - It is beneficial to apply a smoothing filter(Gaussian) first before taking a Laplacian of the image
    - Laplacian of Gaussian (LOG)
    - The image obtained by convolving the LOG mask with the original image is analyzed for zero-crossing to detect edges.

$$h(x,y) = \nabla^2\left[g(x,y) \otimes f(x,y)\right]$$

$$= \nabla^2\left[g(x,y)\right] \otimes f(x,y)$$

$$= \left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4}\right) e^{-\frac{\left(x^2+y^2\right)}{2\sigma^2}} \otimes f(x,y)$$



0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

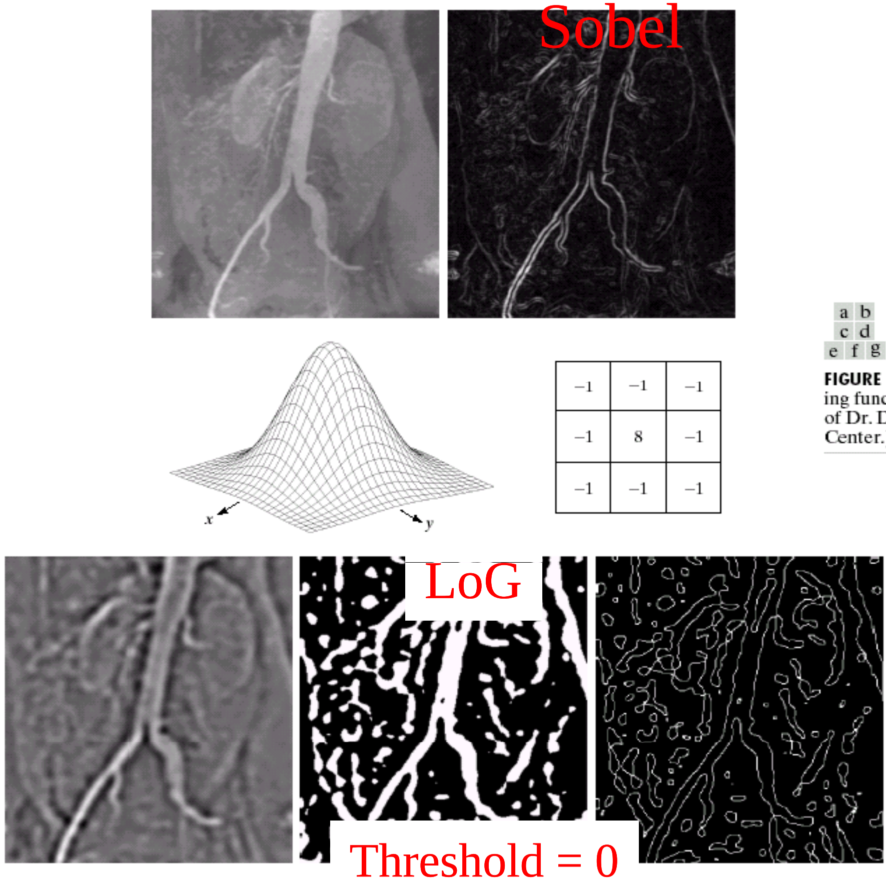
a b
c d

FIGURE 10.14 Laplacian of a Gaussian (LoG). (a) 3-D plot. (b) Image (black is negative, gray is the zero plane, and white is positive). (c) Cross section showing zero crossings. (d) 5 × 5 mask approximation to the shape of (a).

[Gonzalez]

Edge-based image segmentation
-Edge Detection Operations

- LOG operation



a b
c d
e f g

FIGURE 10.15 (a) Original image. (b) Sobel gradient (shown for comparison). (c) Spatial Gaussian smoothing function. (d) Laplacian mask. (e) LoG. (f) Thresholded LoG. (g) Zero crossings. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

[Gonzalez]

Edge-based image segmentation
-Boundary Tracking

- The closed regions can be formed by the edge-linking procedures following the edge-detection

The neighborhood b_{j+1} of edge point b_j can be found to be a boundary point if it satisfied the following conditions.

$$|e(b_j)| > T_1$$

$$|e(b_{j+1})| > T_1$$

$$|e(b_j) - e(b_{j+1})| < T_2$$

$$|\phi(b_j) - \phi(b_{j+1})| \bmod 2\pi < T_3$$

Where e is the magnitude and ϕ the edge orientation

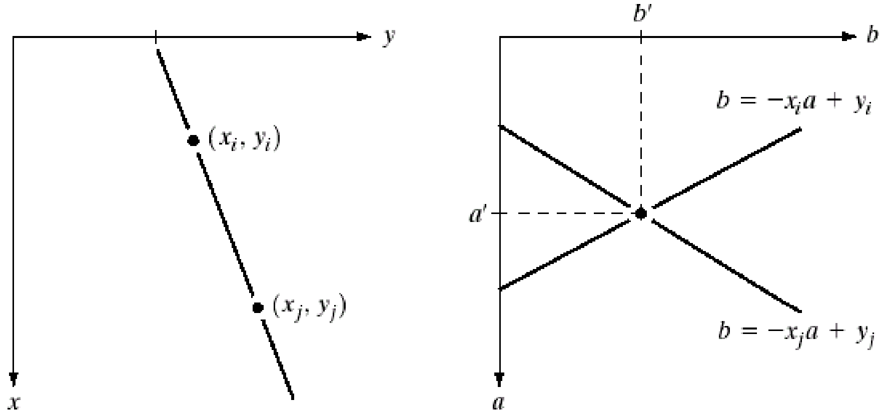
Edge-based image segmentation

Hough Transform

Haugh Transform

x - y space parameter space

$$y_i = ax_i + b \quad \Rightarrow \quad b = -x_i a + y_i$$



a b

FIGURE 10.17 (a) xy-plane. (b) Parameter space.

[Gonzalez]

Edge-based image segmentation

Hough Transform

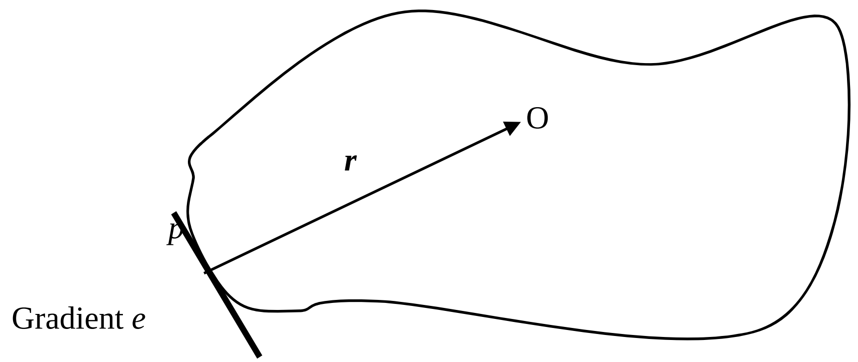
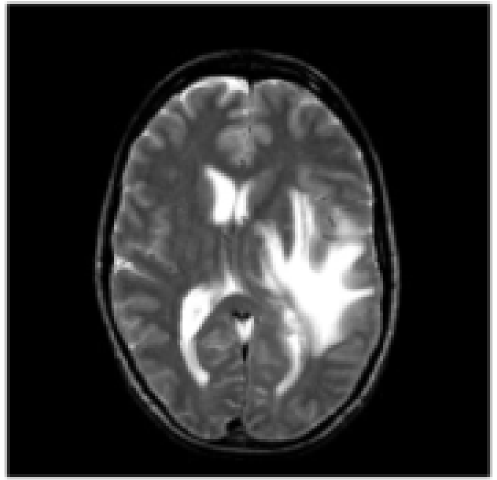
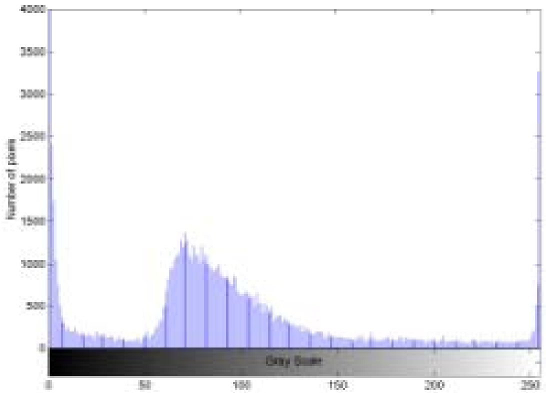


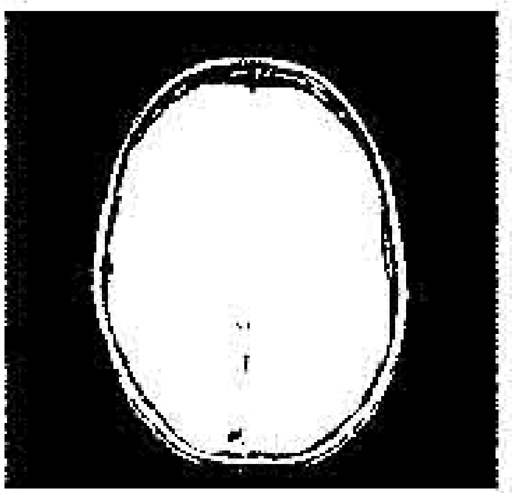
Figure 7.2. A model of the object shape to be detected in the image using Hough transform. The vector ***r*** connects the Centroid and a tangent point *p*. The magnitude and angle of the vector ***r*** are stored in the R-table at a location indexed by the gradient of the tangent point *p*.

Gray-Level Thesholding

$$g(x,y)=\begin{cases}1 & \text{if } f(x,y)>T\\ 0 & \text{if } f(x,y)\le T\end{cases}$$



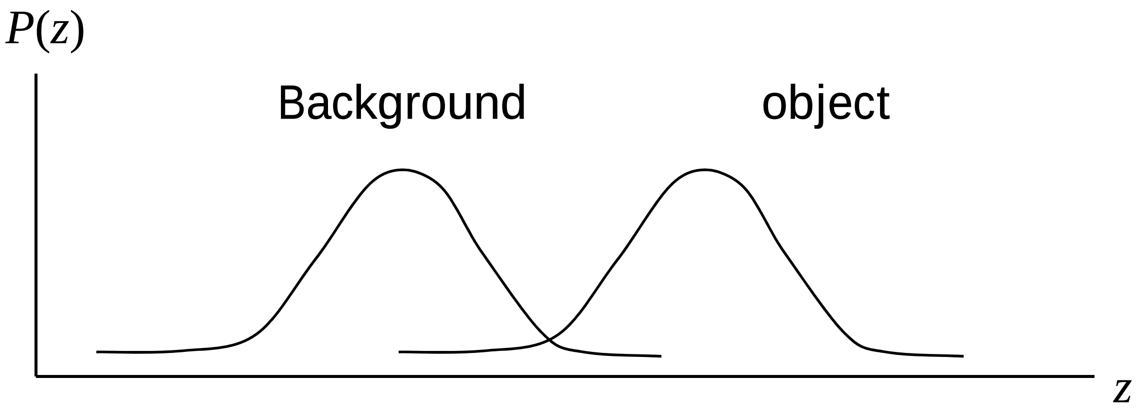




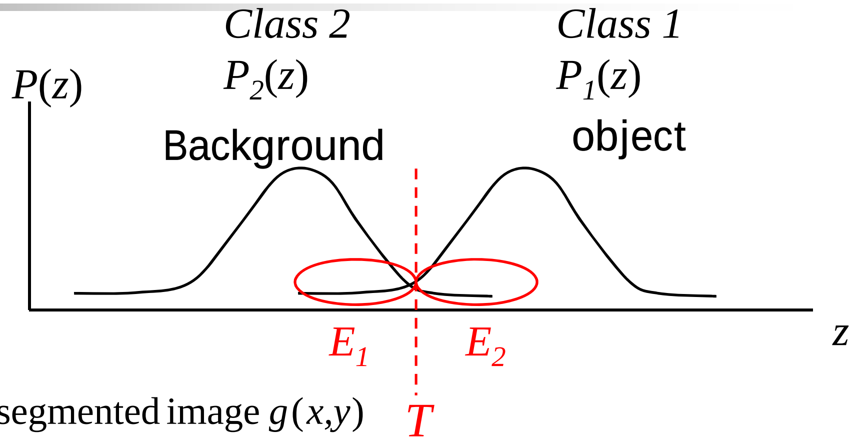
Optimal Global Thresholding

- The parametric distribution based methods can be applied to the histogram of an image
  - Assume the histogram has two Gaussian distributions

$$p(z) = P_1 p_1(z) + P_2 p_2(z)$$



Optimal Global Thresholding



$p(z) = P_1 p_1(z) + P_2 p_2(z)$

$p_1(z)$: The Gaussian distribution of class 1

$p_2(z)$: The Gaussian distribution of class 2

$P_1 + P_2 = 1$

a pixel $f(x,y)$ can be classified to *class* 1 or *class* 2 of the segmented image $g(x,y)$

$$g(x,y) = \begin{cases} Class\ 1 \quad (object) & \text{if } f(x,y) > T \\ Class\ 2 \quad (background) & \text{if } f(x,y) < T \end{cases}$$

The error probabilities of misclassifying a pixel

$E_1(T) = \int_{-\infty}^{T} p_1(z)dz$: The probability of erroneously classifying a *class* 1 pixel to *class* 2

$E_2(T) = \int_{T}^{\infty} p_2(z)dz$: The probability of erroneously classifying a *class* 2 pixel to *class* 1

The overall probability of error in pixel classification using the threshold T

$E(T) = P_2 E_1(T) + P_1 E_2(T)$ $\leftarrow$ $\cancel{E(T) = P_1(T)E_1(T) + P_2(T)E_2(T)}$ Mistake in textbook

Optimal Global Thresholding

$$E(T) = P_2 E_1(T) + P_1 E_2(T) \qquad (7.14)$$

- For image segmentation, the objective is to find an optimal threshold T that minimizes the overall probability of error in pixel classification.
  - The process requires the parameterization of the probability density distributions to find the μ_1, σ_1, and μ_2, σ_2

$$p(z) = \frac{P_1}{\sqrt{2\pi}\sigma_1} e^{-(z-\mu_1)^2/2\sigma_1^2} + \frac{P_2}{\sqrt{2\pi}\sigma_2} e^{-(z-\mu_2)^2/2\sigma_2^2} \qquad (7.15)$$

  - The optimal global threshold T can be determined by finding a general solution the minimizes the equation (7.14) with mixture distribution in Equation (7.15)

Optimal Global Thresholding

- (7.14) thus satisfies the following quadratic expression

$$AT^2 + BT + C = 0$$

$$\textit{where}$$

$$A = \sigma_1^2 - \sigma_2^2$$

$$B = 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2)$$

$$C = \sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + 2\sigma_1^2\sigma_2^2 \ln(\sigma_2 P_1 / \sigma_1 P_2)$$

- If the variances of both classes can be assumed to be equal to σ^2

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln\left(\frac{P_2}{P_1}\right)$$

The determined threshold

Pixel Classification Through Clustering

- The approach is useful when images with pixels representing a feature vector consisting of multiple parameters of interest are to be segmented.
  - Parameters: Gray value, RGB components, contrast and local texture measurements for each pixel
  - Clustering may produce disjoint regions with holes or regions with a single pixel
    - Post-processing algorithm is usually applied
    - region growing, pixel connectivity or rule-based algorithm

Pixel Classification Through Clustering

k-means Clustering

- Partition *d*-dimentional data into *k* clusters
- The objective function providing the desired properties of the distribution of feature vectors of clusters

- The algorithm is quite sensitive to the initial cluster assignment and the choice of the distance measure. (See the following algorithm for more)

Pixel Classification Through Clustering

k-means Clustering

Algorithm

1. Select the number of clusters k with initial cluster centroids v_i ; $i=1,2,\ldots,k$
2. Partition the input data points into k clusters by assigning each data point x_j to the closest cluster centroid v_i .(ex. Euclidean distance)

$$d_{ij} = \left\| x_j - v_i \right\|$$

3. Compute a cluster assignment matrix $\mathbf{U}$ representing the partition of the data points with the binary membership value of the jth data point to the ith cluster such that $\mathbf{U}=[u_{ij}]$, where

$$\begin{array}{cc} u_{ij} & \text{points}(n) \\ \text{clusters}(k) & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$u_{ij} \in \{0,1\} \quad \text{for all } i,\ j$$

$$\sum_{i=1}^{k} u_{ij} = 1 \quad \text{for all } j \text{ and } 0 < \sum_{j=1}^{n} u_{ij} < n \text{ for all } i$$

Pixel Classification Through Clustering

k-means Clustering

Algorithm (cont.)

4. Re-computer the centroids using the membership values as

$$v_i = \frac{\sum_{j=1}^{n} u_{ij} x_j}{\sum_{j=1}^{n} u_{ij}} \quad \text{for all } i$$

u_{ij}

$$\begin{array}{cc} & \text{points}(n) \\ \text{clusters}(k) & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

5. If cluster centroids does not change, stop; otherwise go to step 2

The k-means clustering method optimizes the sum-of-squared-error based objective function $J_w(\mathbf{U},\mathrm{v})$

$$J_w(U,v) = \sum_{i=1}^{k} \sum_{j=1}^{n} \left\| x_j - v_i \right\|^2$$

Pixel Classification Through Clustering

Fuzzy k-means Clustering

- Utilize an adaptable membership value (u_{ij}) that can be updated

$$J_w(U,v) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m d_{ij}^2 = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m \left\| x_j - v_i \right\|^2$$

c : the number of clusters

n : the number of data vectors

u_{ij} : the fuzzy membership

m : the fuzziness index

$$0 \le u_{ij} \le 1 \text{ for all } i,j$$

$$\sum_{i=1}^{c} u_{ij} = 1 \quad \text{for all } j \quad \text{and} \quad 0 < \sum_{j=1}^{n} u_{ij} < n \quad \text{for all } i$$

Region Growing

- Examine pixels in the neighborhood based on a pre-defined similarity criterion
- The neighborhood pixels with similar properties are merged to form closed regions for segmentation
- Guarantee the segmented regions of connected pixels

Region Growing

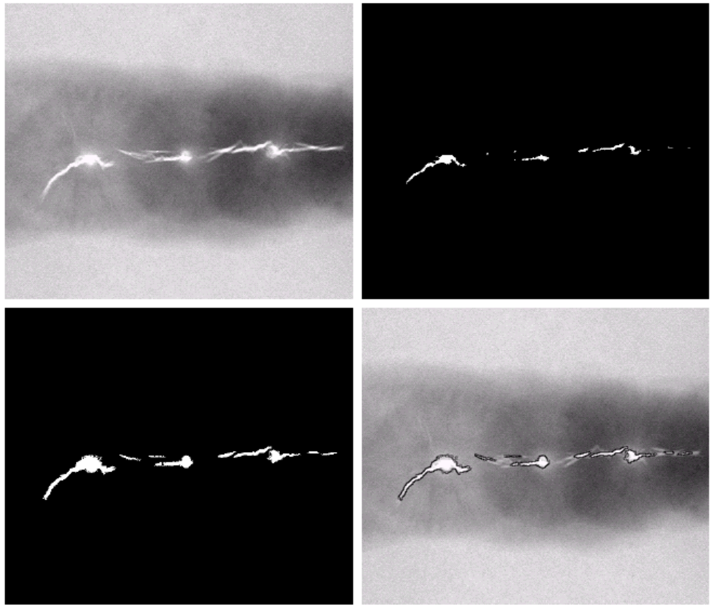
- Two criteria
  - A similarity criterion that defines the basis for inclusion of pixels in the growth of the region
  - A stopping criterion stops the growth of the region

1. Determine the seed for each region
   (the pixels with gray 255 are selected to be seeds in this case)
2. Choose criteria for region growing
   - The absolute gray-level difference between any pixels and seed
   - The detected pixel had to be 8-connected to at least one pixel in that region

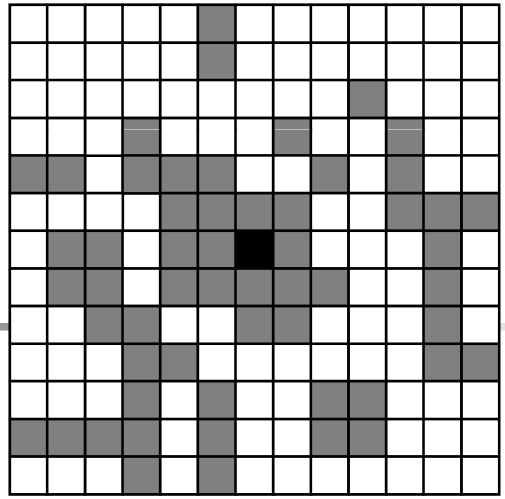
[Gonzalez]

a b
c d

FIGURE 10.40 (a) Image showing defective welds. (b) Seed points. (c) Result of region growing. (d) Boundaries of segmented defective welds (in black). (Original image courtesy of X-TEK Systems, Ltd.).



Region Growing



Center Pixel

Pixels satisfying the similarity criterion

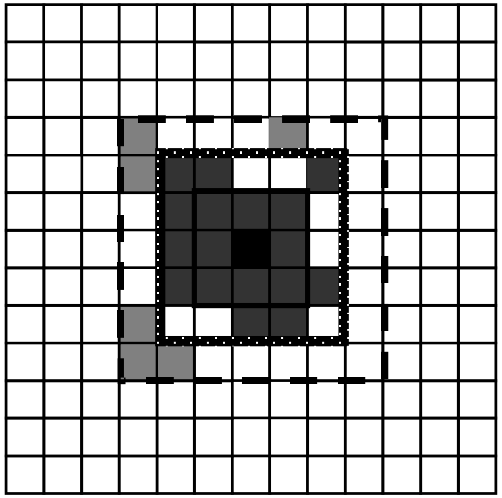
Pixels not satisfying similarity criterion

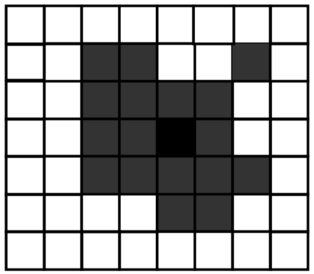
1st iteration — 3x3 neighborhood, 100% (9/9) growth satisfied

2nd iteration — 5x5 neighborhood , 56%(9/16) growth satisfied

3rd iteration — 7x7 neighborhood, %(6/24) growth satisfied

The stopping criterion : the minimum percentage is 30%

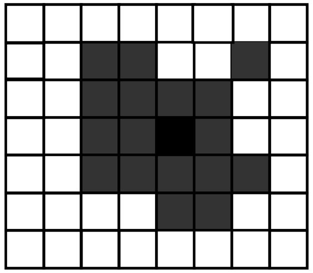




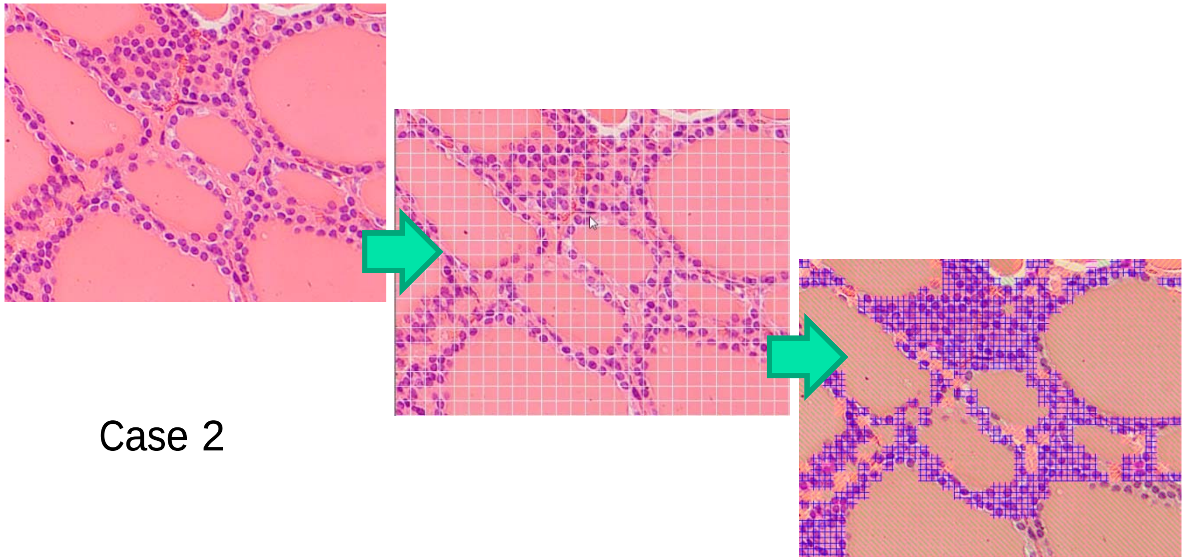
Segmented region

Region -Merging

- An image is partitioned into a large number of potential homogeneous regions
  - Be examined for homogeneity of pre-defined property as gray values, contrast, texture...
  - Two neighborhood regions with similar property are merged



Case 1



Case 2

Region -Splitting

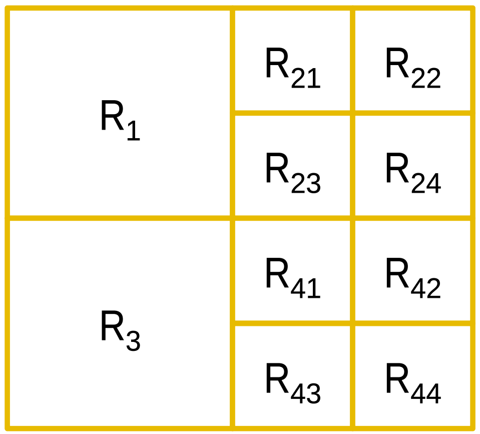
- Examine the heterogeneity of a predefined property of the entire region in terms of its distribution and the mean, variance, minimum and maximum values.
  - Heterogeneous region R is split into two or more regions R_1, R_2, ..., R_n
  - Rule-based splitting and quad-tree based splitting

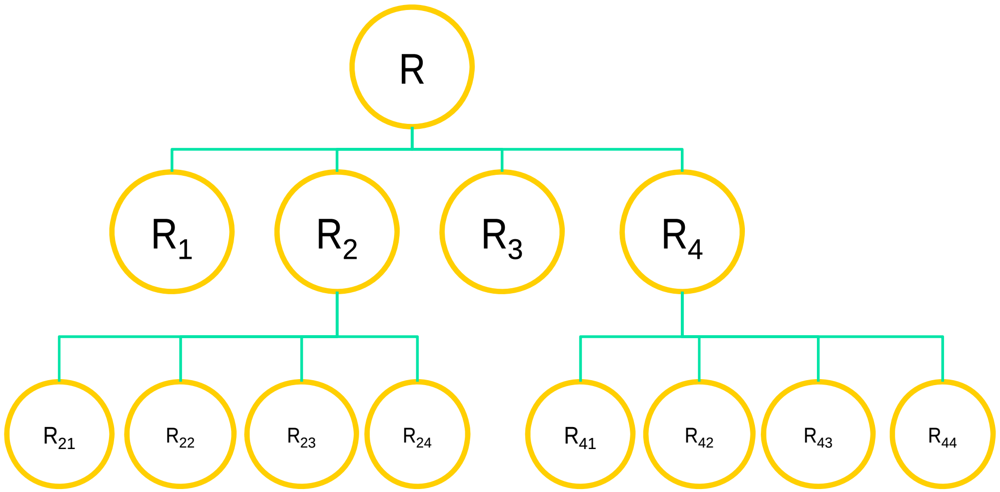
1. Each region, $R_i; i = 1,2,\ldots,n$ is connected.
2. $\bigcup_{i=1}^{n} R_i = R$.
3. $R_i \cap R_j = O$ for all $i, j; i \neq j$.
4. $H(R_i) = TRUE$ for $i = 1,2,\ldots,n$.
5. $H(R_i \cup R_j) = FALSE$ for $i \neq j$,

where $H(R_i)$ is a logical predicate for the homogeneity criterion on the region R_i.

Region -Splitting

- An image with quad region-splitting process





Quad-tree structure

Recent advances in segmentation

The large variability in anatomical structures needs of a reliable, accurate, and diagnostically useful segmentation

- Model-based estimation methods
- Rule-based methods

Estimation-Model Based Adaptive Segmentation
-Multi-level Adaptive Segmentation (MAS)

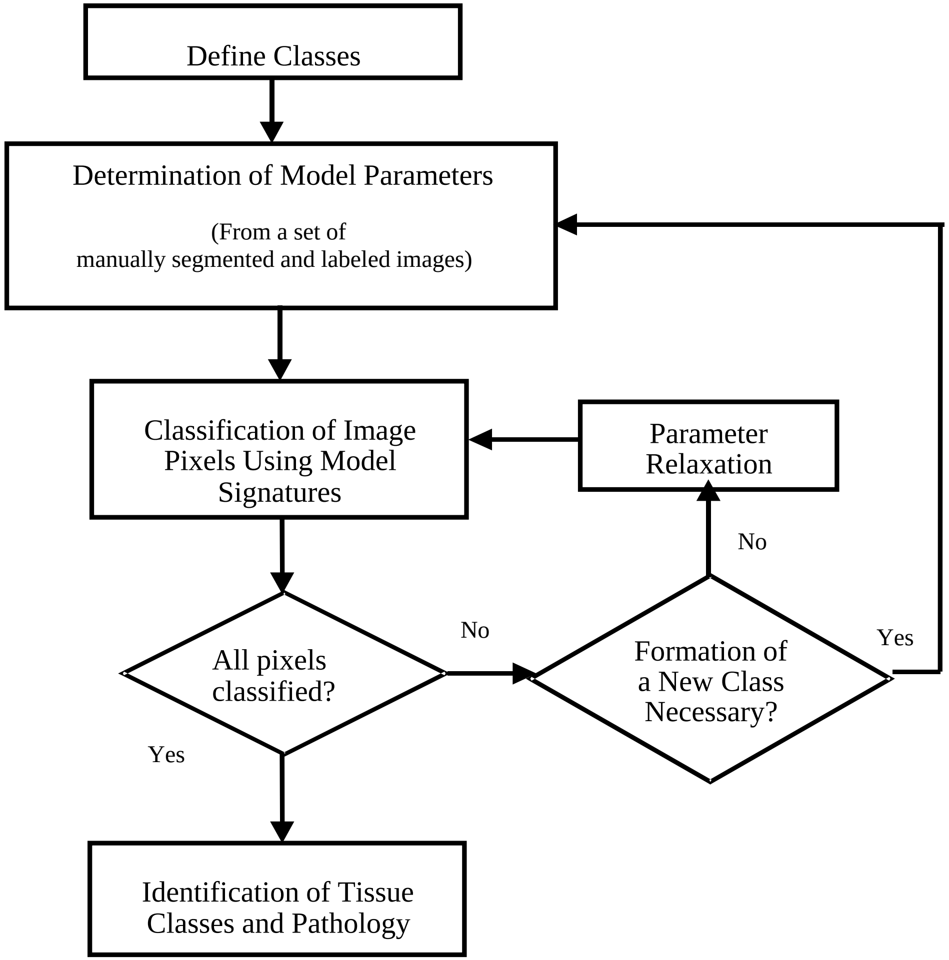


Image Segmentation Using Neural Networks
-Neural Network

- Neural Network do not require underlying class probability distribution for accurate classification

- The decision boundaries for pixel classification are adapted through an iterative training process (supervised neural network)

- Neural Network paradigms
  - Backpropagation, Radial Basis Function, and Self-Organizing Feature Maps

Image Segmentation Using Neural Networks
-Neural Network

- Important works for neural network
  - Selection of a useful set of features as input data for classification
  - Selection of the training examples
  - The structure of the network

Artificial Neural Network: Backpropagation

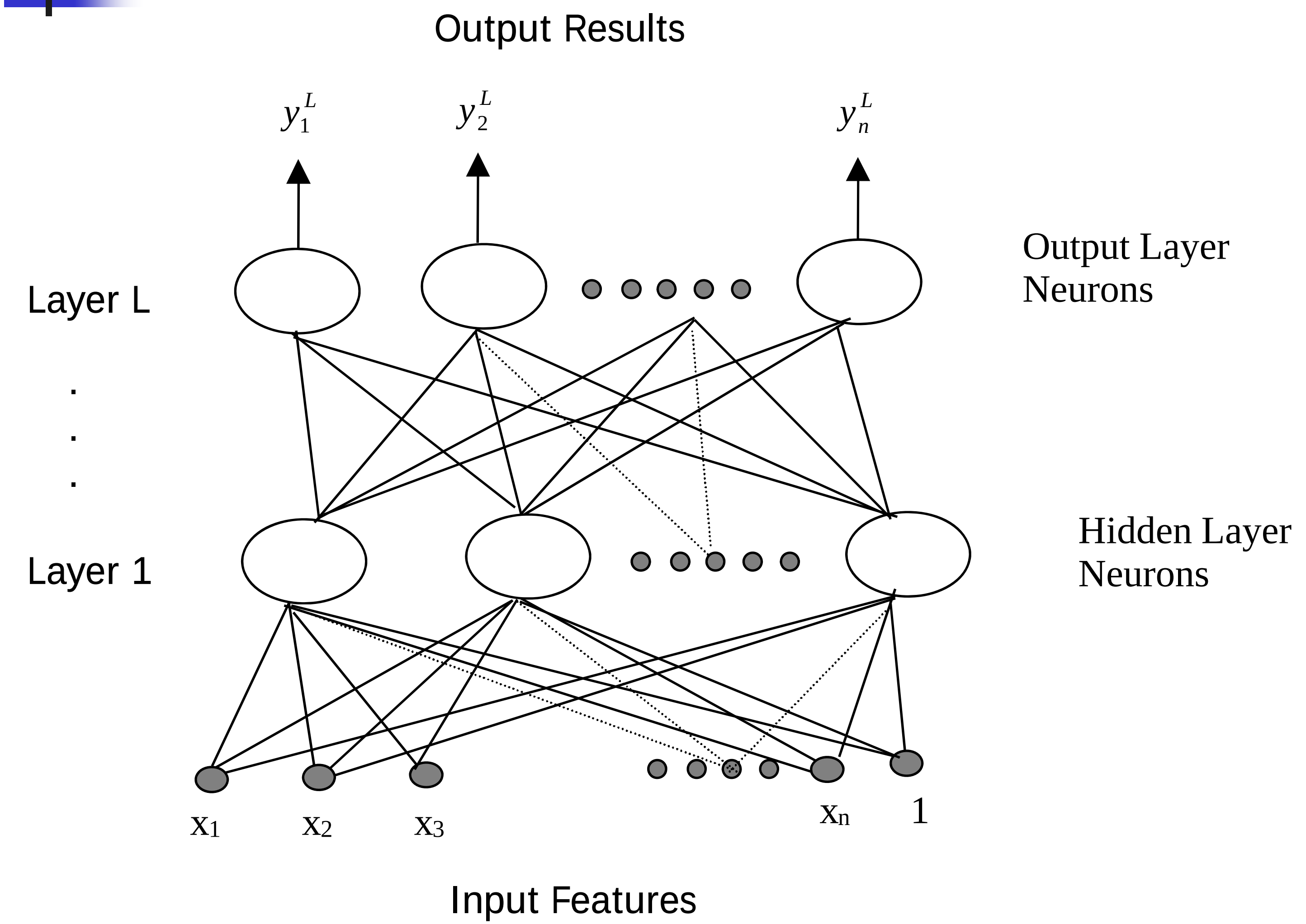
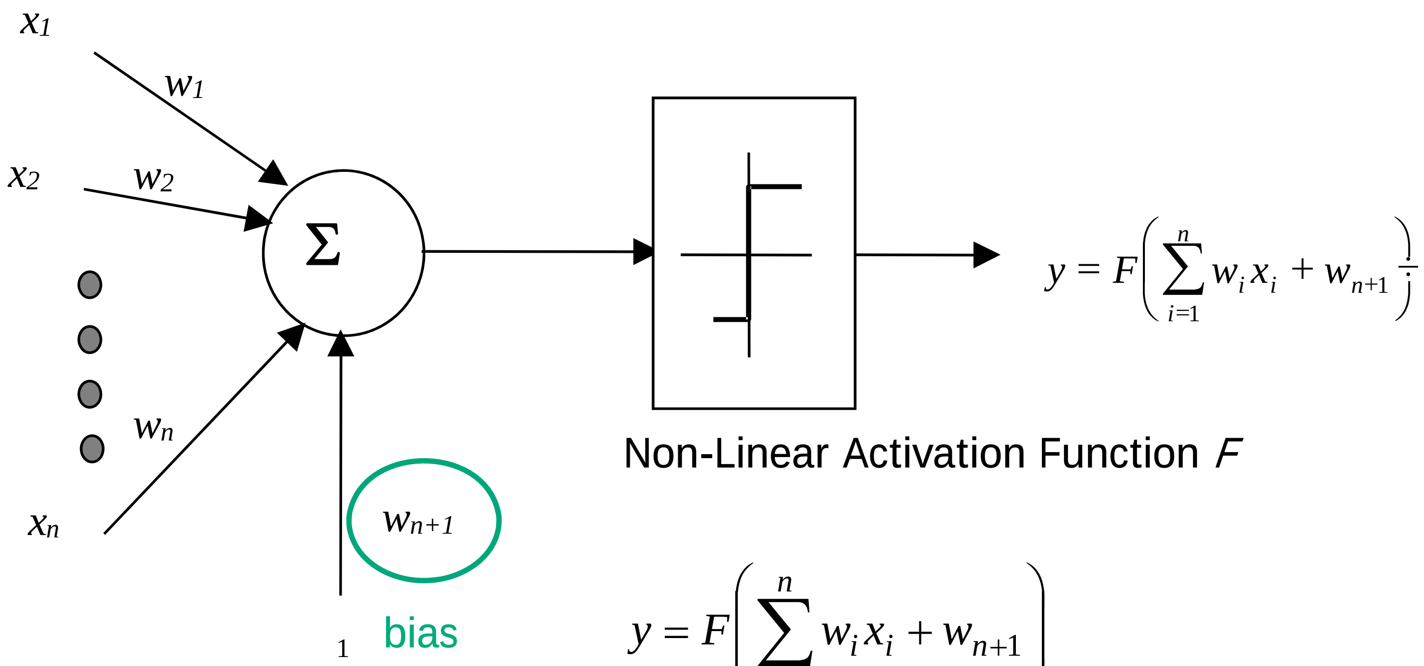


Image Segmentation Using Neural Networks
-Backpropropagation Neural Network



$$y = F\left(\sum_{i=1}^{n} w_i x_i + w_{n+1}\right)$$

$F(y)$ can be defined as a sigmoidal function

$$F(y) = \frac{1}{1+e^{-y}}$$

Figure 7.11. A basic computational neural element or Perceptron for classification

Image Segmentation Using Neural Networks
-Backpropropagation Neural Network

$$y^{(k)} = F\left(W^k y^{(k-1)}\right) \text{ for } k = 1,2,\ldots L$$

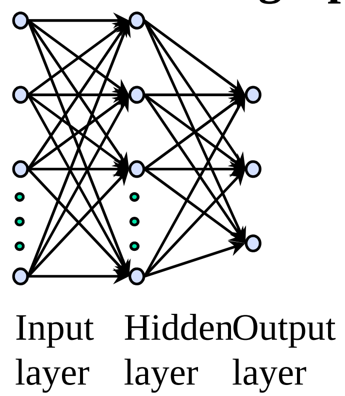
where $y^{(k)}$ is the output of the kth layer

$$\mathrm{y}^{(0)} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_n \\ 1 \end{bmatrix}; \quad \mathrm{y}^{(k)} = \begin{bmatrix} y_1^{(k)} \\ y_2^{(k)} \\ . \\ y_n^{(k)} \\ 1 \end{bmatrix}$$

and

$$W^{(k)} = \begin{bmatrix} w_{11}^{(k)} & w_{12}^{(k)} & . & w_{1n}^{(k)} & w_{1(n+1)}^{(k)} \\ w_{21}^{(k)} & w_{22}^{(k)} & . & w_{2n}^{(k)} & w_{2(n+1)}^{(k)} \\ . & . & . & . & . \\ w_{n1}^{(k)} & w_{n2}^{(k)} & . & w_{nn}^{(k)} & w_{n(n+1)}^{(k)} \\ & & . & & \end{bmatrix}$$

Architectural graph



Signal-flow graph

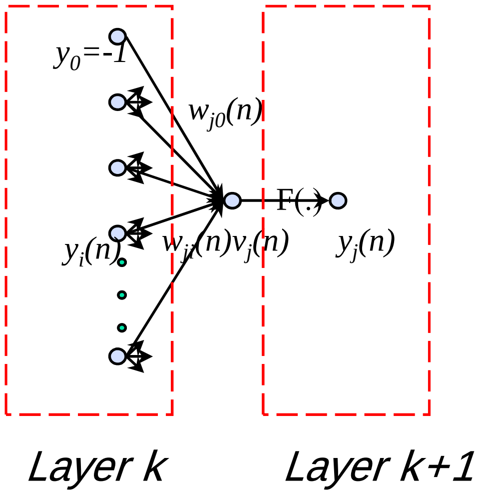
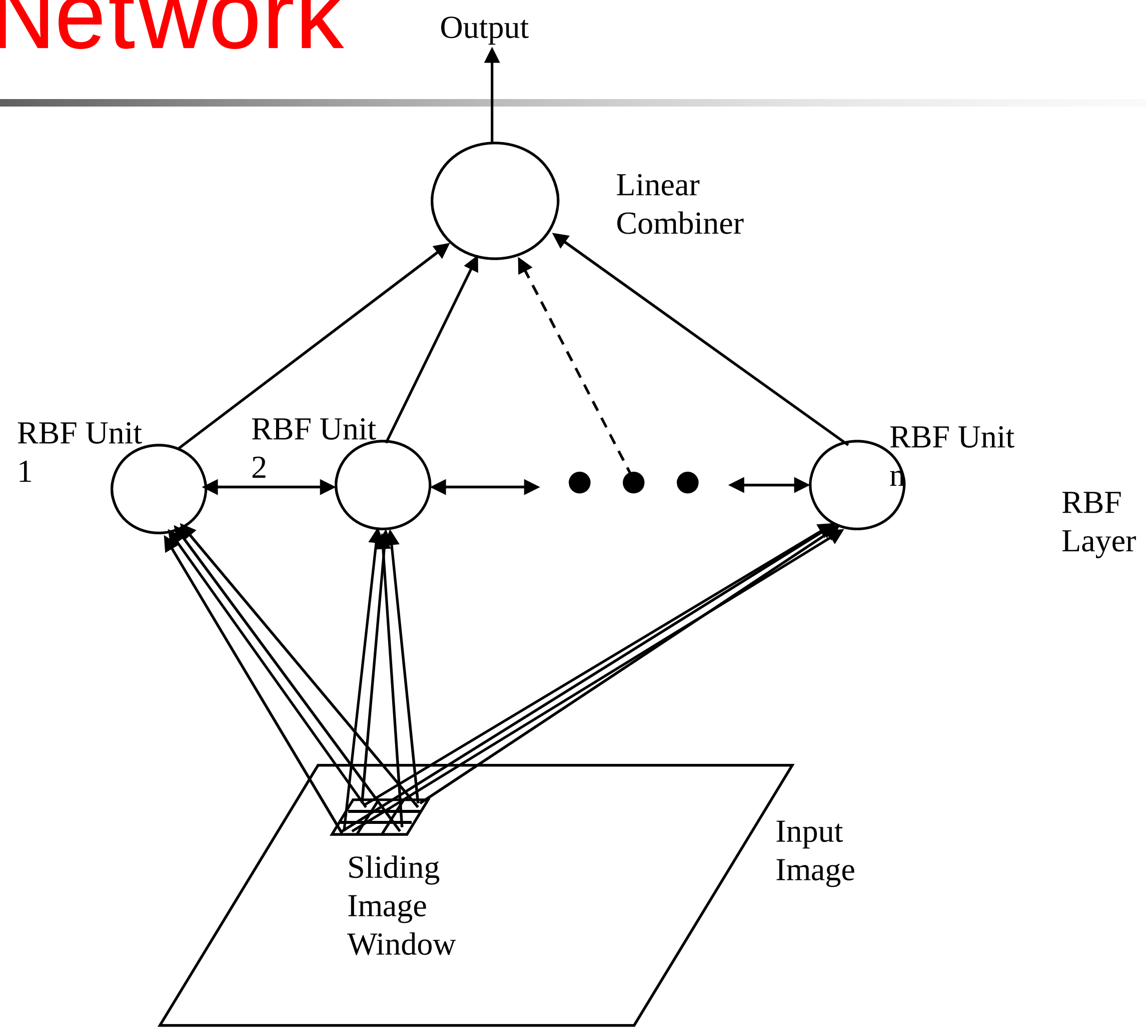


Image Segmentation Using Neural Networks
-Backpropropagation Neural Network

- Least Mean Squared(LMS) error algorithm for training the neural network

  1. Assign random weights in the range of [-1,+1] to all weights w_{ij}^{k}
  2. For each classified pattern pair $\{\boldsymbol{y}^{(0)}, \boldsymbol{y}^{L}\}$ in the train set, do the following steps:
     a. Compute the output values of each neural element using the current weight matrix.
     b. Find the error $\boldsymbol{e}^{(k)}$ between the computed output vector and the desired out vector.
     c. Adjust the weight matrix using the change $d\boldsymbol{W}^{(k)}$ computed as $d\boldsymbol{W}^{(k)} = \alpha \boldsymbol{e}^{(k)}[\boldsymbol{y}^{(k-1)}]$ for all layers =1,…,L, where α is the learning rate that can be set between 0 and 1.
  3. Repeat Step 2 for all classified pattern pairs in the training set until the error vector for each training example is sufficiently low or zero.

RBF Network



Output
Linear
Combiner
RBF Unit
1
RBF Unit
2
RBF Unit
n
RBF
Layer
Sliding
Image
Window
Input
Image

RBF Network

- The final output of the network $f(x)$

$$f(x) = \sum_{i=1}^{k} w_i \Phi(\mathbf{x}, i)$$

$$\textbf{with}\ \ \Phi(\mathbf{x}, i) = \frac{\exp\left(-\frac{\|x - c_i\|}{2\sigma_i}\right)}{\sum_{j=1}^{K} \exp\left(-\frac{\|x - c_j\|}{2\sigma_j}\right)}$$

c_i : the K centers
w_i : the K weighted connections from the hidden layer units to the output unit.

RBF Network

- Within the basic topology, the system can be represented in state-space from y=(F,w), where F is the matrix of activation functions

$$F = \begin{bmatrix} \Phi(x_1,1) & \ldots & \Phi(x_1,K) \\ \ldots & \ldots & \ldots \\ \Phi(x_N,1) & \ldots & \Phi(x_N,K) \end{bmatrix}$$

The weight van be calculated from the Moore-Penrose inverse

$$w = \left(F^T \cdot F + \alpha \cdot I\right)^{-1} \cdot F^T \cdot y$$

where α is a small value such as 0.1

RBF NN Based Segmentation