Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

MIN-Fakultät
Fachbereich Informatik
Arbeitsbereich SAV/BV (KOGS)

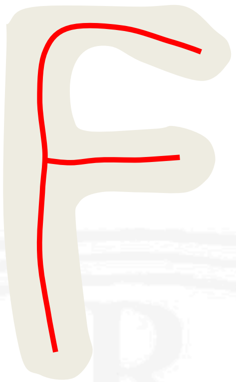
Image Processing 1 (IP1)
Bildverarbeitung 1

Lecture 14 – Skeletonization and Matching

Winter Semester 2015/16

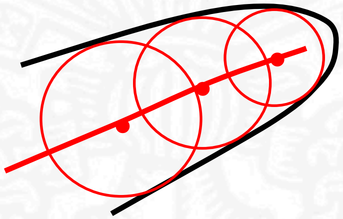
Slides: Prof. Bernd Neumann
Slightly revised by: Dr. Benjamin Seppke & Prof. Siegfried Stiehl

Skeletons

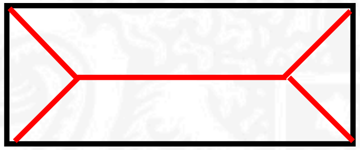


The skeleton of a region is a line structure which represents "the essence" of the shape of the region, i.e. follows elongated parts.

- Useful e.g. for character recognition
- Medial Axis Transform (MAT) is one way to define a skeleton:
  The MAT of a region R consists of all pixels of R which have more than one closest boundary point.
- MAT skeleton consists of centers of circles which touch boundary at more than one point

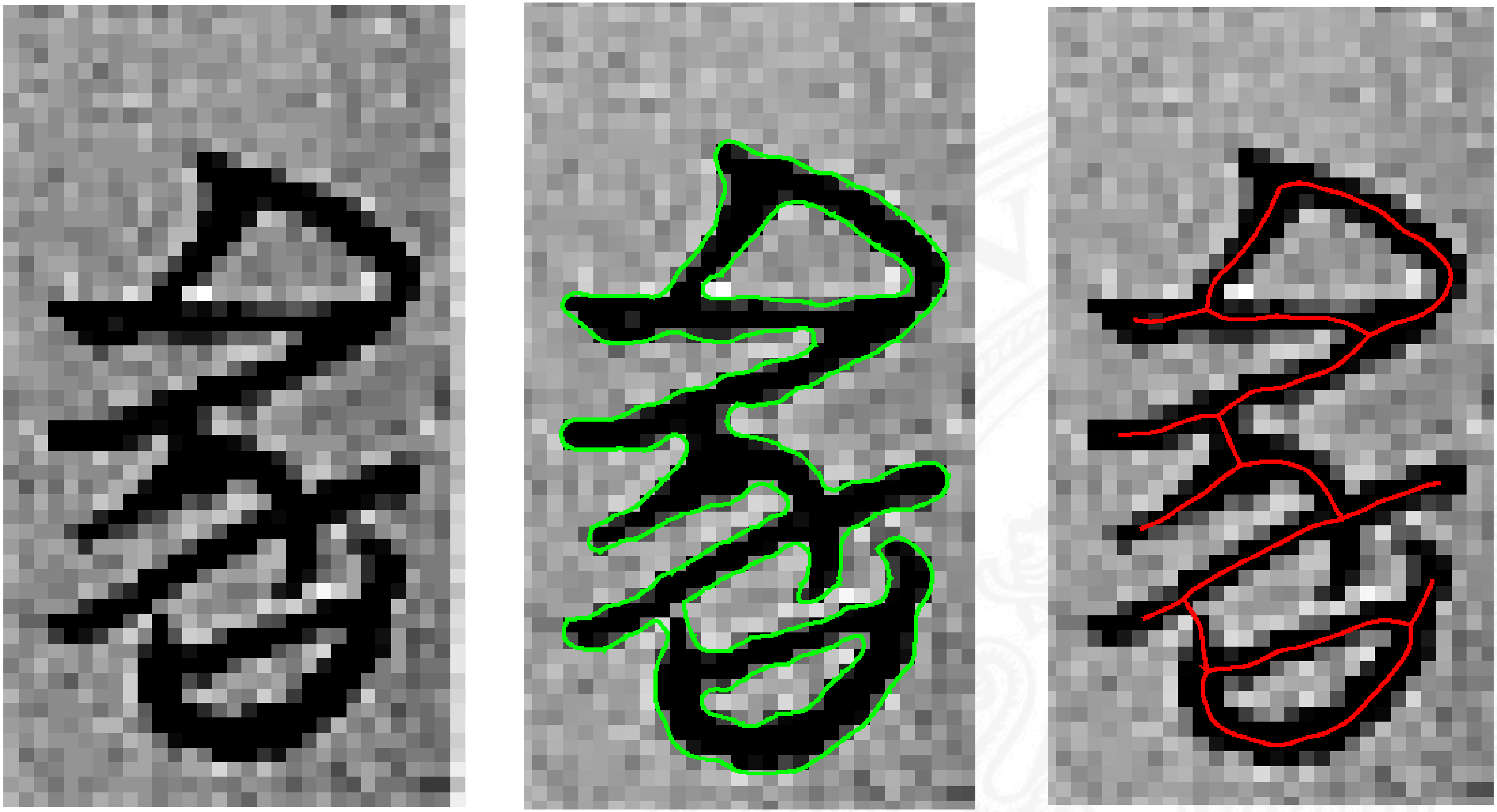


- MAT skeleton of a rectangle shows problems:



Note that "closest boundary point" depends on digital metric!

Skeleton Extraction for Chinese Character Description

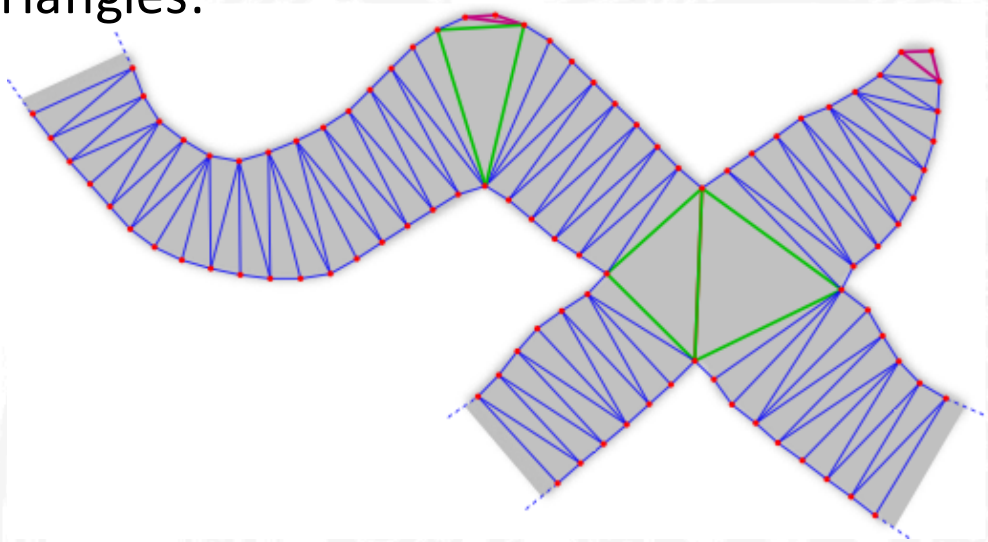


Stroke Analysis by Triangulation

Constrained Delaunay Triangulation (CDT) connects contour points to triangles such that the circumference of a triangle contains no other points.

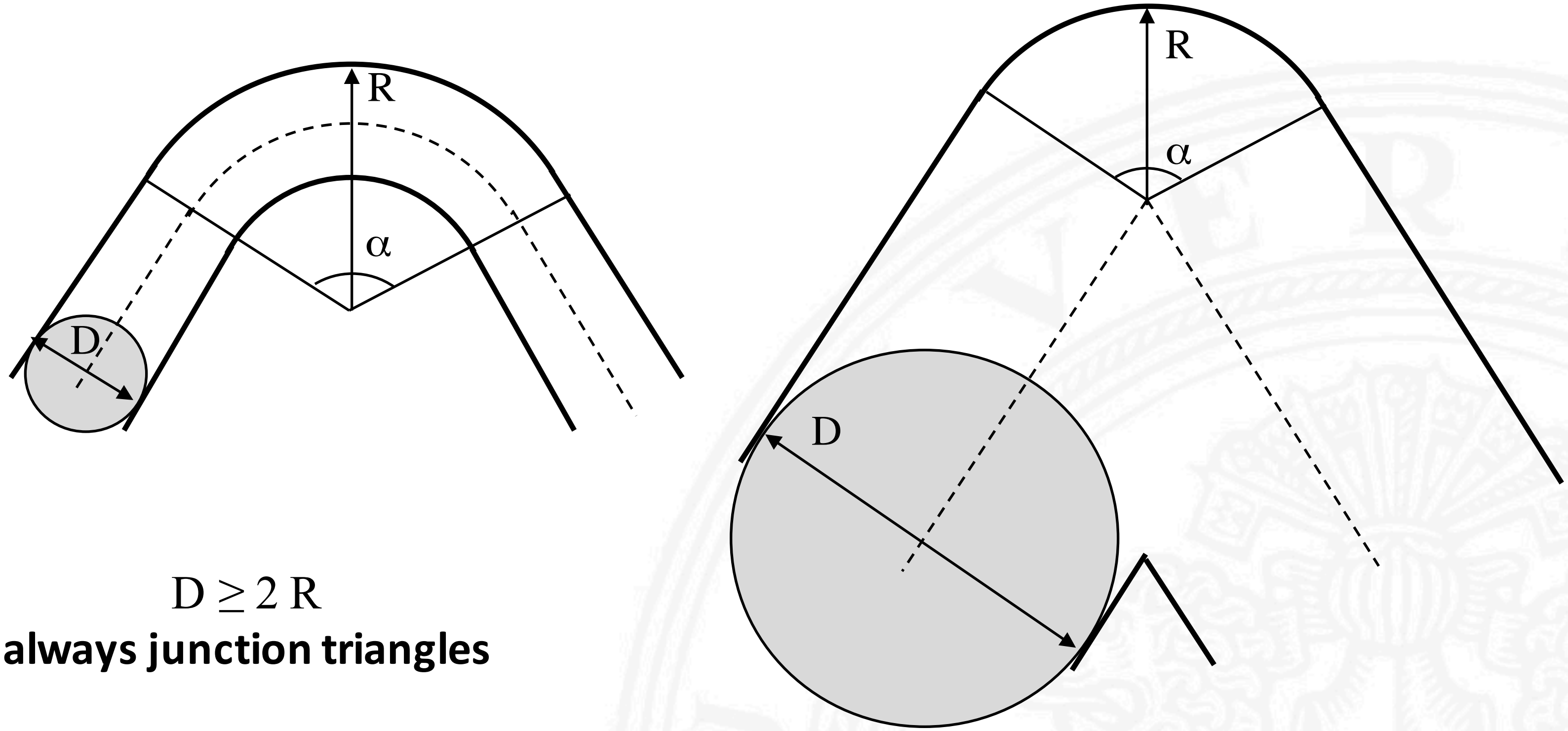
CDT generates three types of triangles:

- junction triangles (green)
- none of the triangle sides coincides with the contour
- sleeve triangles (blue)
- terminal triangles (red)



Junction triangles indicate stroke intersections or sharp stroke corners

Conditions for Junction Triangles



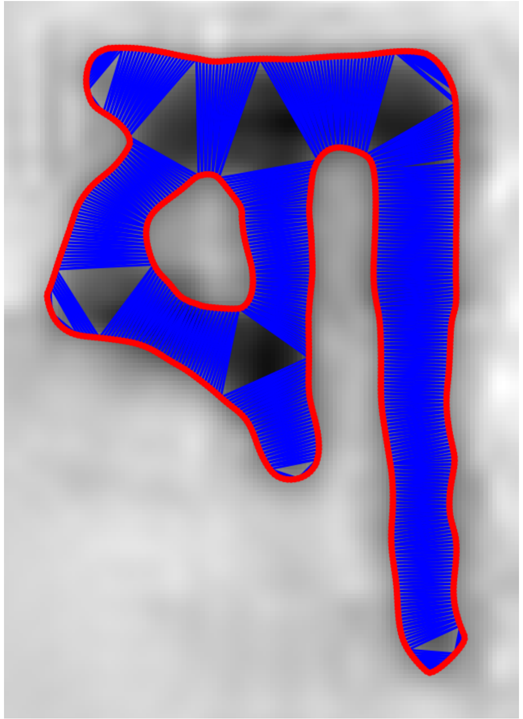
$D \geq 2R$

always junction triangles

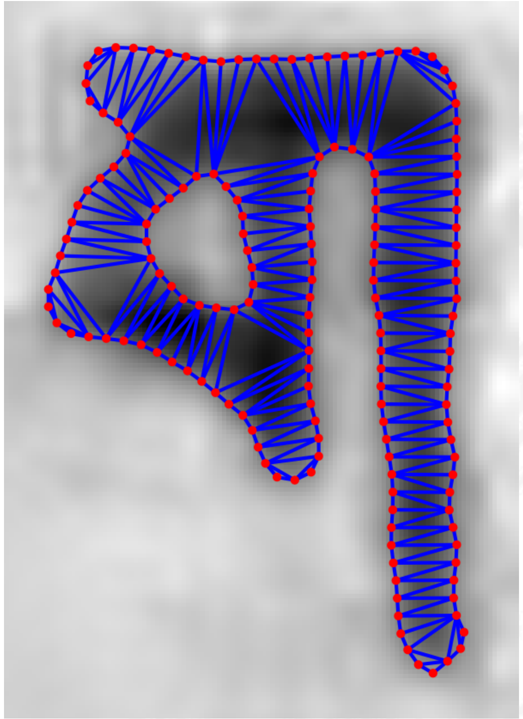
A curved line with angle α and outer contour radius R, drawn with a stylus of diameter D, will generate a junction triangle if

$$D > R\,(1 + \cos \alpha/2)$$

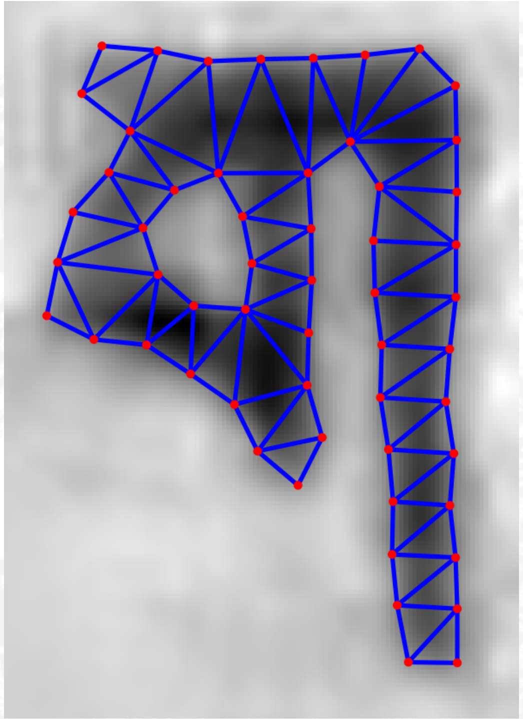
Weak Influence of Contour Point Spacing



dense spacing



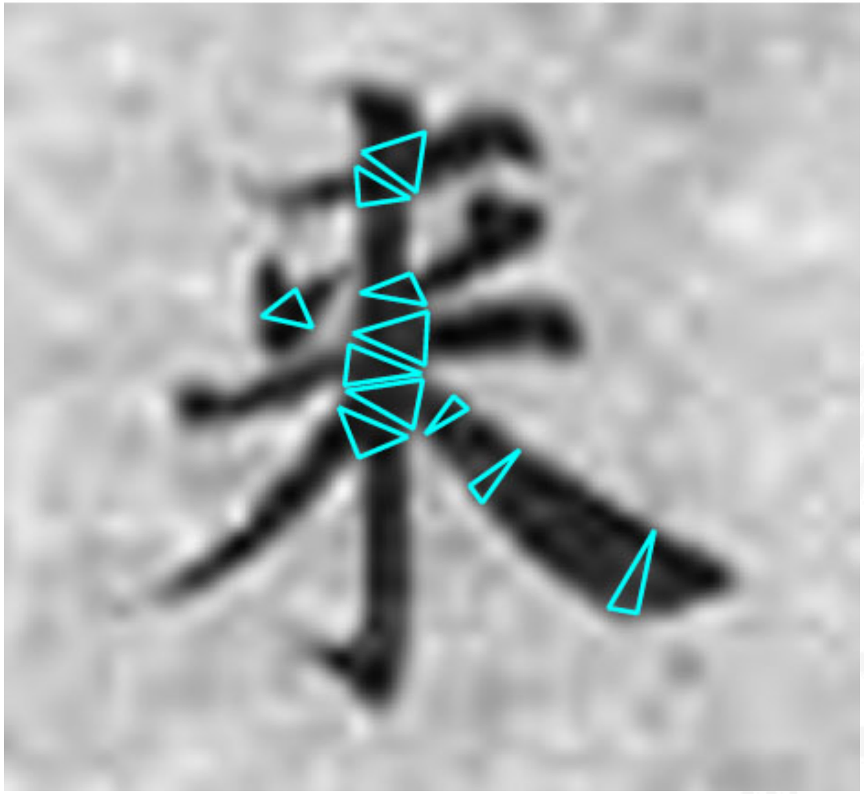
medium spacing



coarse spacing

no junction triangles if corners are cut

Stroke Segment Merging



- Segments meeting at a junction may be merged if they are compatible regarding orientation and stroke width
- Segments between two neighbouring junction triangles may be intersections with irregular direction and stroke width
- Global criteria and knowledge of the writing system must be invoked to resolve ambiguities

Results of Stroke Analysis I

Results of Stroke Analysis II

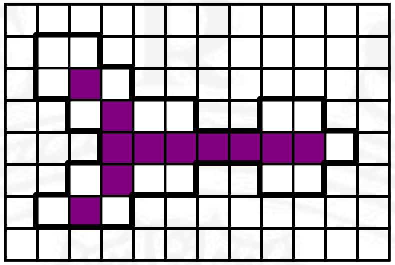
Thinning Algorithm

Thinning algorithm by Zhang and Suen 1987
(from Gonzalez and Wintz: "Digital Image Processing")

Repeat A to D until no more changes:

A Flag all contour points which satisfy conditions (1) to (4)
B Delete flagged points
C Flag all contour points which satisfy conditions (5) to (8)
D Delete flagged points

Example:



Assumptions:
- region pixels = 1
- background pixels = 0
- contour pixels 8-neighbours of background

Neighbourhood labels:

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

Conditions:

(1) $2 \leq N(p_1) \leq 6$
(2) $S(p_1) = 1$
(3) $p_2 \times p_4 \times p_6 = 0$
(4) $p_4 \times p_6 \times p_8 = 0$

(5) $2 \leq N(p_1) \leq 6$
(6) $S(p_1) = 1$
(7) $p_2 \times p_4 \times p_8 = 0$
(8) $p_2 \times p_6 \times p_8 = 0$

$N(p_1)$ = number of nonzero neighbours of p_1

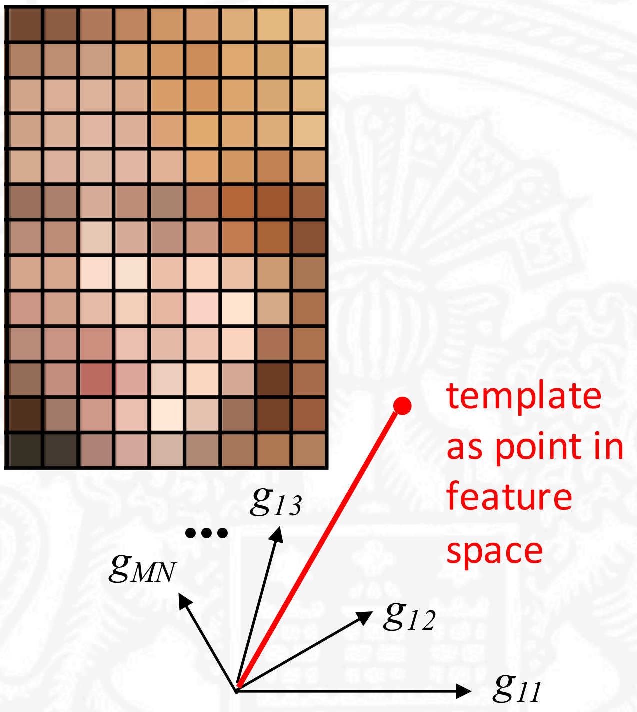
$S(p_1)$ = number of 0 - 1 transitions in ordered sequence $p_2, p_3, \ldots$

Templates

A template is a translation-, rotation- and scale-variant shape desription. It may be used for object recognition in a fixed, reoccurring pose.

- A M-by-N template may be treated as a vector in MN-dimensional feature space
- Unknown objects may be compared with templates by their distance in feature space

Example:
Template for face recognition



Distance measures:

g_{mn} pixels of image

t_{mn} pixels of template

$$d_e^2 = \sum_{mn} (g_{mn} - t_{mn})^2$$ squared Euclidean distance

$$d_a = \sum_{mn} | g_{mn} - t_{mn} |$$ absolute distance

$$d_b = \max_{mn} \left| g_{mn} - t_{mn} \right|$$ maximal absolute distance

Cross-correlation

$$r = \sum_{mn} g_{mn} t_{mn}$$ cross-correlation between image g_{mn} and template t_{mn}

Compare with squared Euclidean distance d_e^2:

$$d_e^2 = \sum_{mn} (g_{mn} - t_{mn})^2 = \sum_{mn} g_{mn}^2 + \sum_{mn} t_{mn}^2 - 2r$$

Image "energy" Σg_{mn}^2 and template "energy" Σt_{mn}^2 correspond to length of feature vectors.

$$r' = \frac{\sum_{mn} g_{mn} t_{mn}}{\sqrt{\sum_{mn} g_{mn}^2 \sum_{mn} t_{mn}^2}}$$

Normalized cross-correlation is independent of image and template energy. It measures the cosine of the angle between the feature vectors in MN-space.

Cauchy-Schwartz Inequality:

$|r'| \le 1$ with equality iff $g_{mn} = c\, t_{mn}$, all mn

Fast Normalized Cross-Correlation I

- Normalized Cross-correlation should be preferred w.r.t. cross-correlation:
  - Illumination invariant
  - Comparable resulting value range [-1, ..., 1]
- Problem:
  - (non-normalized) cross-correlation can be computed efficiently Remember Convolution Theorem, Fourier-Transform & FFT
  - Normalization is not computable using FFT!
  - Computation time is very high!
- Solution by Lewis 95: Optimize the Normalized Cross-Correlation by means of FFT and caching strategies

Fast Normalized Cross Correlation II

Recall the basic equation:

$$r' = \frac{\sum_{mn} g_{mn} t_{mn}}{\sqrt{\sum_{mn} g_{mn}^2} \sqrt{\sum_{mn} t_{mn}^2}}$$

Non-nomalized cross-correlation:
may be computed fast using the FFT, since:

$$\sum_{mn} g_{mn} t_{mn} = FT^{-1}\left(FT(g) \cdot conj\left(FT(t)\right)\right)$$

Sum under the template:
constant for all nm,
can be precomputed

Sum under the image:
Not constant!
Changes for each position of the template!

Idea of Lewis: Use the integral image to compute the non-constant term efficiently

Fast Normalized Cross Correlation III

Creation of (squared) integral images $s(u,v)$ and $s2(u,v)$:

$$s(u, v) = g(u, v) + s(u-1, v) + s(u, v-1) - s(u-1, v-1)$$

$$s^2(u, v) = g^2(u; v) + s^2(u-1, v) + s^2(u, v-1) - s^2(u-1, v-1)$$

Extraction of the sums for a window (size $M \times N$) at position (u,v):

$$e_f = s(u+N-1, v+N-1) - s(u-1, v+N-1) - s(u+N-1, v-1) + s(u-1, v-1)$$

$$e_f^2 = s^2(u+N-1, v+N-1) - s^2(u-1, v+N-1) - s^2(u+N-1, v-1) + s^2(u-1, v-1)$$

Complexity analysis:

- Table creation needs approx. 3M×N operations
- Less than explicitly computed window sums!

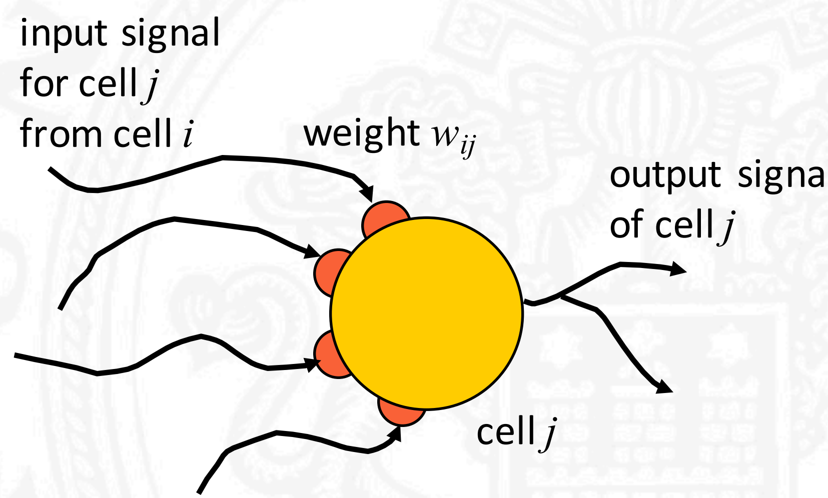
In praxis: Acceleration of factors 1000 and more w.r.t. the naive implementation!

Artificial Neural Nets

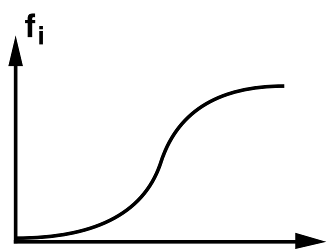
Information processing in biological systems is based on neurons with roughly the following properties:

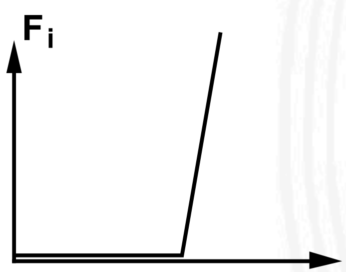
- the degree of activation is determined by incoming signals
- the outgoing signal is a function of the activation
- incoming signals are mediated by weights
- weights may be modified by learning

net input for cell j	$\Sigma\, w_{ij}\, o_i(t)$
activation	$a_j(t) = f_j\,(a_j, \Sigma\, w_{ij}\, o_i(t))$
output signal	$o_j(t) = F_j\,(a_j)$



Typical shapes of f_i and F_i:

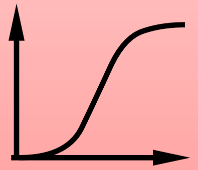


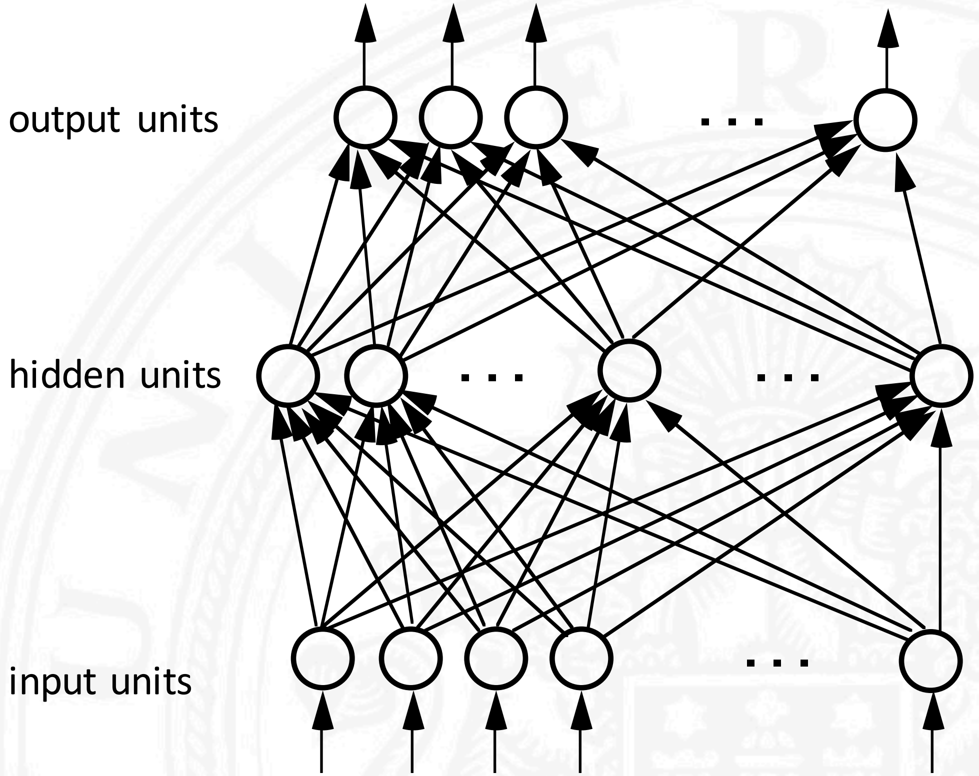


Multilayer Feed-forward Nets

Example: 3-layer net

- each unit of a layer is connected to each unit of the layer below
- units within a layer are not connected
- activation function f is differentiable (for learning)

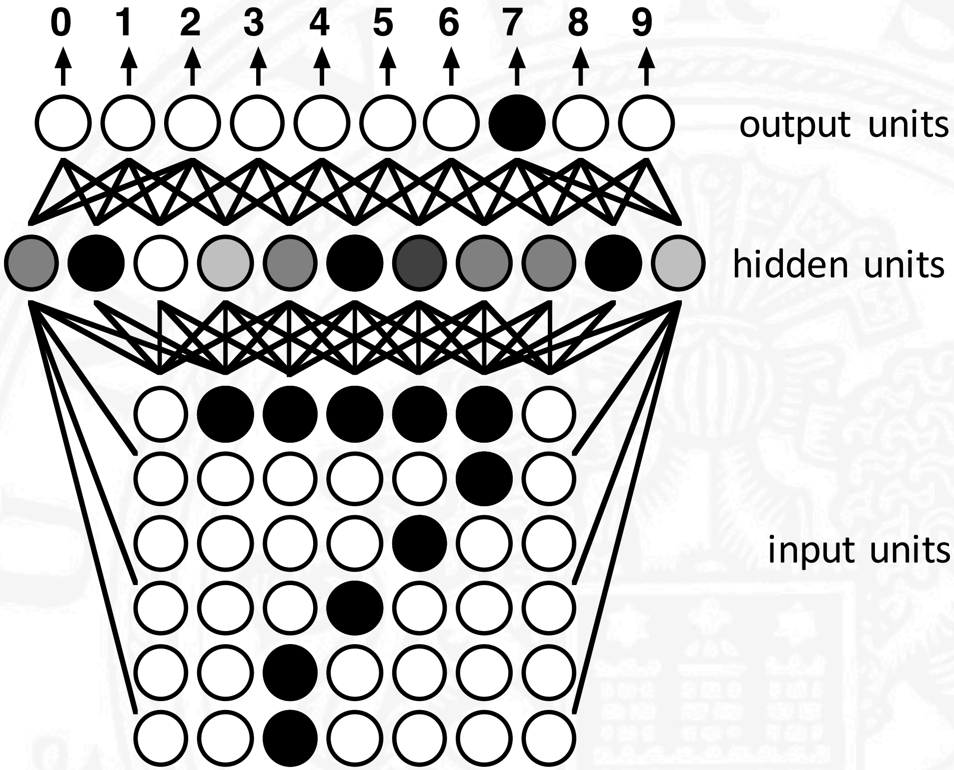




Character Recognition with a Neural Net

Schematic drawing shows 3-layer feed-forward net:

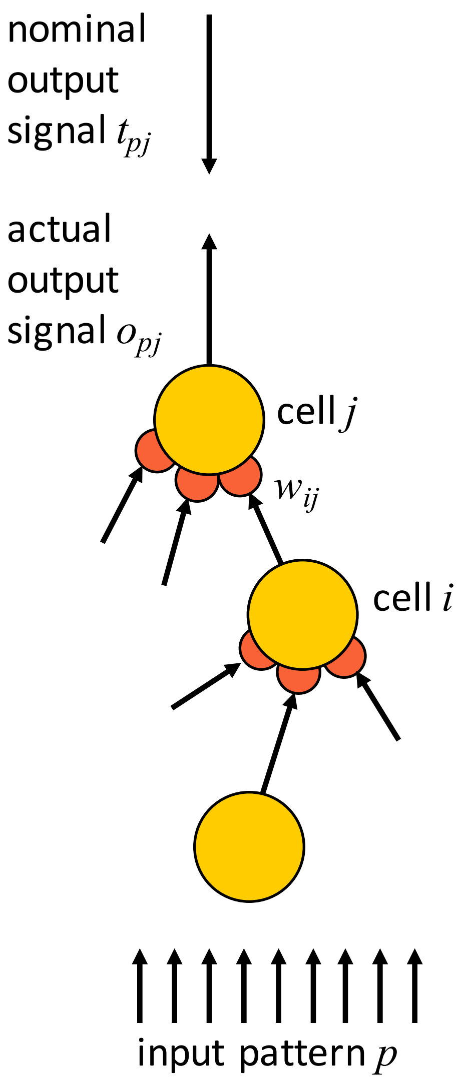
- input units are activated by sensors and feed hidden units
- hidden units feed output units
- each unit receives weighted sum of incoming signals



Supervised learning

Weights are adjusted iteratively until prototypes are classified correctly
(-> backpropagation)

Learning by Backpropagation



Supervised learning procedure:

- **present example and determine output error signals**
- **adjust weights which contribute to errors**

Adjusting weights:

- Error signal of output cell j for pattern p is

  $\delta_{pj} = (t_{pj} - o_{pj})\, f_j´(net_{pj})$

  $f_j´()$ is the derivative of the activation function $f()$

- Determine error signal δ_{pi} for internal cell i recursively from error signals of all cells k to which cell i contributes.

  $\delta_{pi} = f_i´(net_{pi})\ \Sigma_k\, \delta_{pk} w_{ik}$

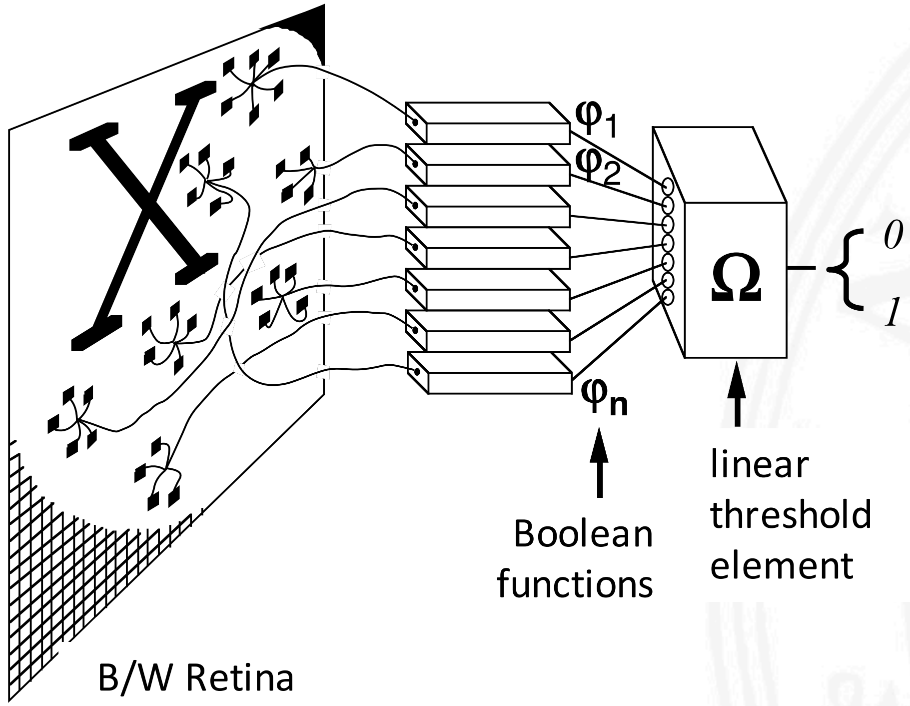
- Modify all weights: $\Delta_p w_{ij} = \eta \delta_{pj} o_{pi}$ h is a positive constant

The procedure must be repeated many times until the weights are "optimally" adjusted. There is no general convergence guarantee.

Perceptrons I

Which shape properties can be determined by combining the outputs of local operators?

A perceptron is a simple computational model for combining local Boolean operations. (Minsky and Papert, Perceptrons, 69)



φ_i Boolean functions with local support in the retina:
- limited diameter
- limited number of cells

output is 0 or 1

Ω compares weighted sum of the φ_i with fixed threshold θ:

$$\Omega = \begin{cases} 1 & \text{if } \sum w_i \varphi_i > \theta \\ 0 & \text{otherwise} \end{cases}$$

Perceptrons II

A limited-diameter perceptron cannot determine connectedness

Assume perceptron with maximal diameter d for the support of each φ_i.
Consider 4 shapes as below with $a <$ d and b $>>$ d.

Boolean operators may distinguish 5 local situations:

φ_1 φ_2 φ_3 φ_4 φ_5

φ_5 is clearly irrelevant for distinguishing between the 2 connected and the 2 disconnected shapes

For Ω to exist, we must have:

$$w_1 \varphi_1 + w_4 \varphi_4 < \theta \qquad w_2 \varphi_2 + w_4 \varphi_4 > \theta$$

$$w_2 \varphi_2 + w_3 \varphi_3 < \theta \qquad w_1 \varphi_1 + w_3 \varphi_3 > \theta$$

⇒ $\Sigma\, w_i \varphi_i < 2\theta$ ⇒ $\Sigma\, w_i \varphi_i > 2\theta$

⇒ contradiction, hence Ω cannot exist