12 From Logic Expressions to Logic Networks

Fuzzy Systems Engineering
Toward Human-Centric Computing

Contents

12.7 From fuzzy logic networks to Boolean functions and their minimization through learning

12.8 Interfacing the fuzzy neural network

12.9 Interpretation aspects – a refinement of induced RBS

12.10 Reconciliation of perception of information granules and granular mappings

12.1 Introduction

Neural networks

- Neural networks
  - – nonlinear processing elements
  - – highly plastic
  - – capable of learning
  - – universal approximation
- Learning strategies
  - – supervised (e.g. backpropagation)
  - – unsupervised (e.g. self-organizing maps)

Neural networks

- Learning methods
  - parametric learning (e.g. gradient-based)
  - structural learning (e.g. genetic algorithms)
- Highly distributed processing
- Black box nature
- Encode a description of data
  - difficult to interpret
  - lack of transparency

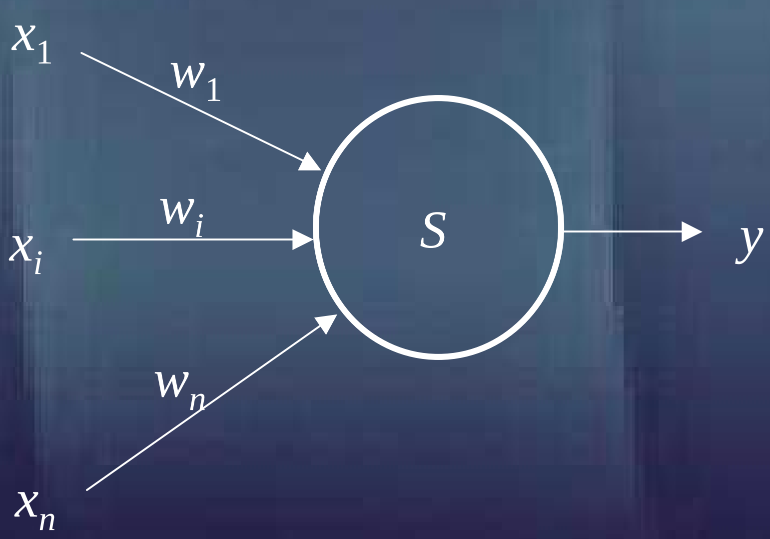
Fuzzy neurons and networks

- Highly distributed processing
- Adds transparency
- Uses ***and*** and ***or*** generic logic operations
- Encode a collection of logic statements (rules)

12.2 Main categories of fuzzy neurons

Aggregative neurons

or **neuron**

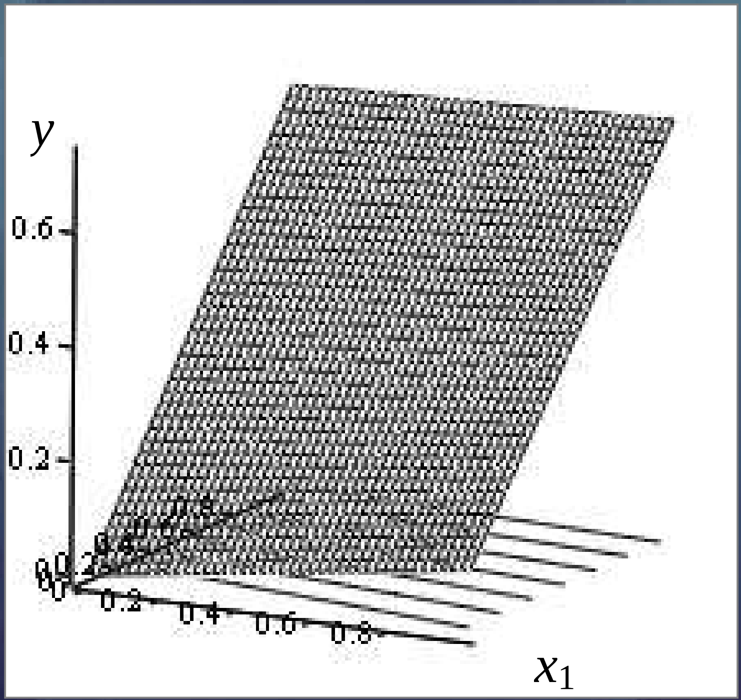


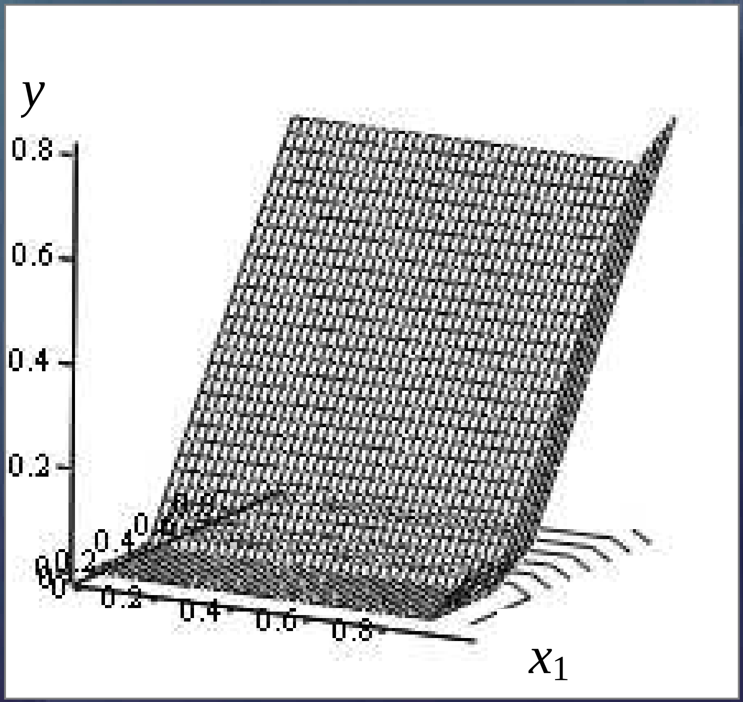
S: $[0, 1]^n \rightarrow [0, 1]$

$$y = \mathop{S}_{i=1}^{n} (x_i \; t \; w_i)$$

$y = \mathrm{OR}(\mathbf{x};\mathbf{w})$

or neuron





t = product
s = probabilistic sum

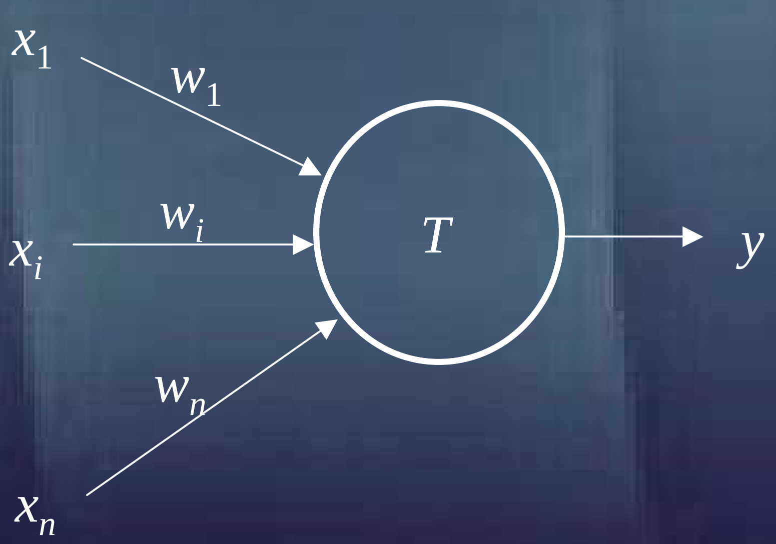
$\mathbf{w}$ = [0.1, 0.7]

t = Lukasiewicz
s = Lukasiewicz

$\mathbf{w}$ = [0.1, 0.7]

and neuron

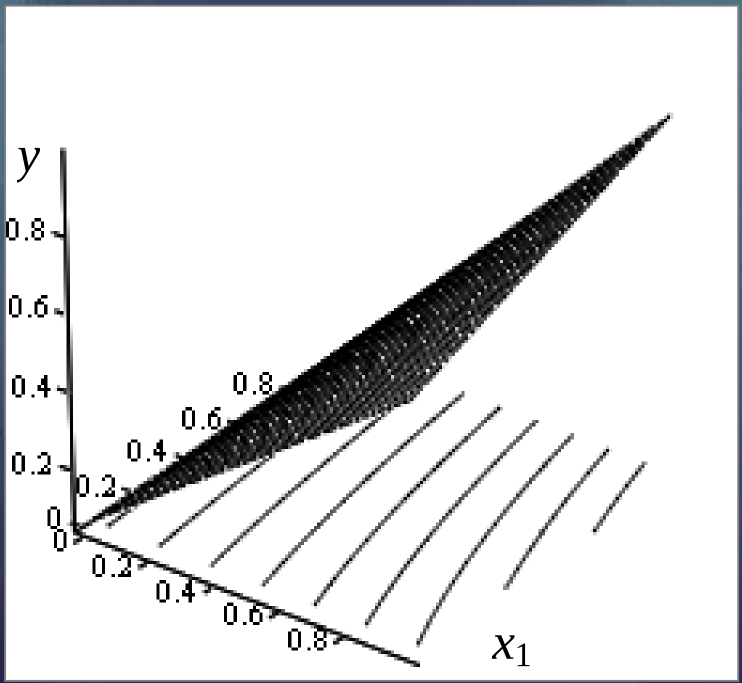
$T: [0, 1]^n \rightarrow [0, 1]$



$$y = \underset{i=1}{\overset{n}{\mathrm{T}}} (x_i \; s \; w_i)$$

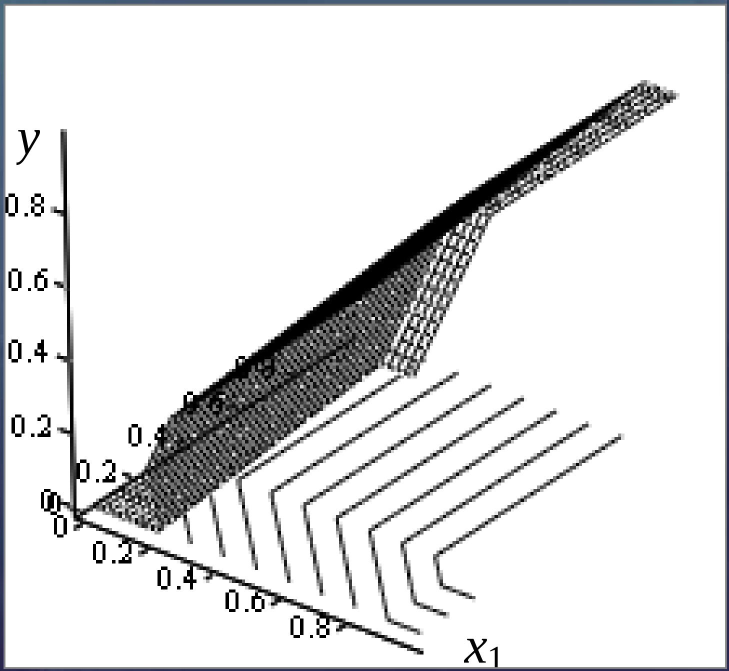
$y = \text{AND}(\mathbf{x;w})$

and neuron



t = product
s = probabilistic sum

w = [0.1, 0.7]

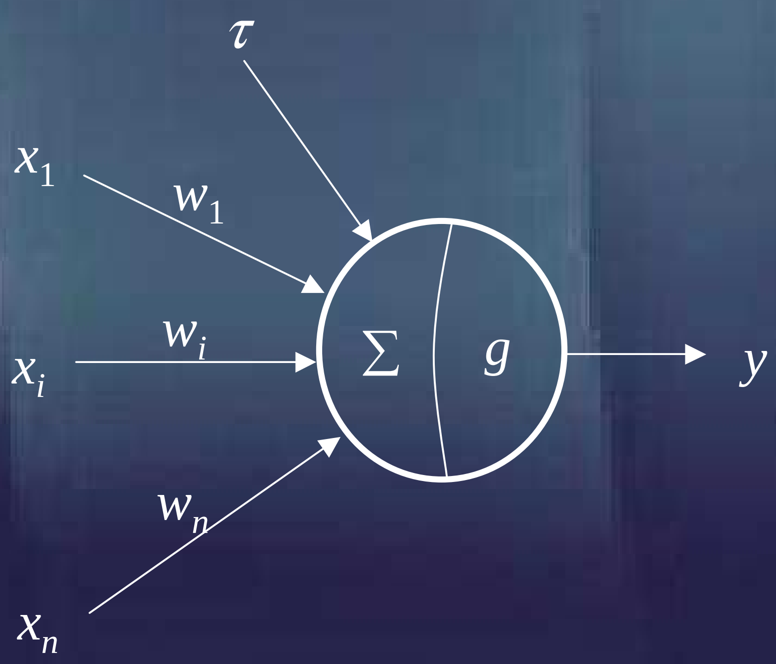


t = Lukasiewicz
s = Lukasiewicz

w = [0.1, 0.7]

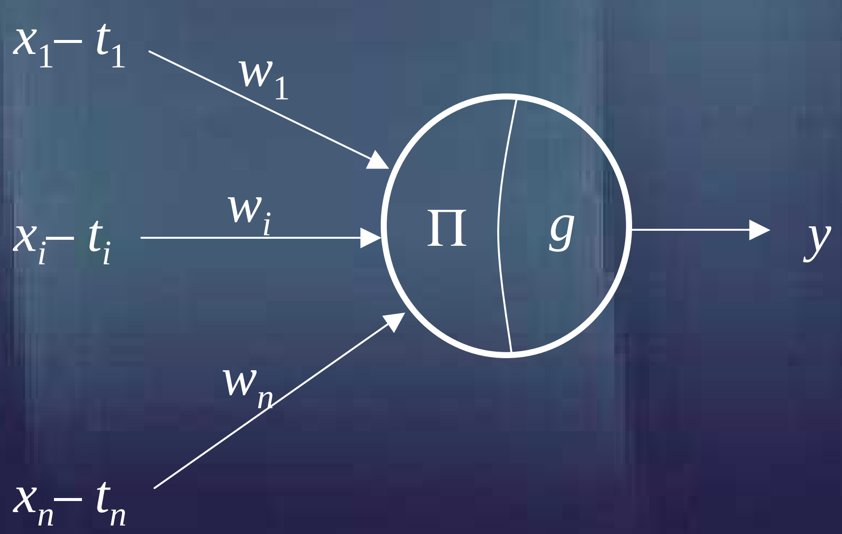
Standard neurons

$\sum$ neuron (additive)



$$y = g\left[\sum_{i=1}^{n} (x_i \, w_i + \tau)\right]$$

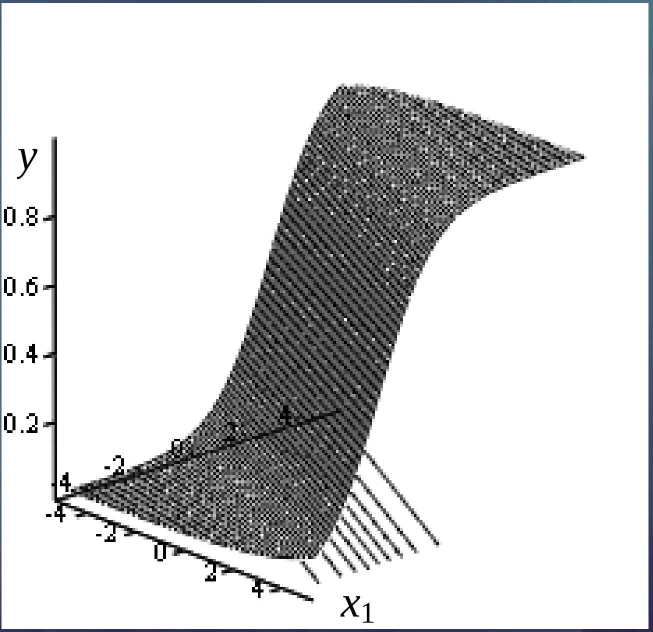
Π neuron (multiplicative)



$$y = g\left(\prod_{i=1}^{n} (x_i - t_i)^{w_i}\right)$$

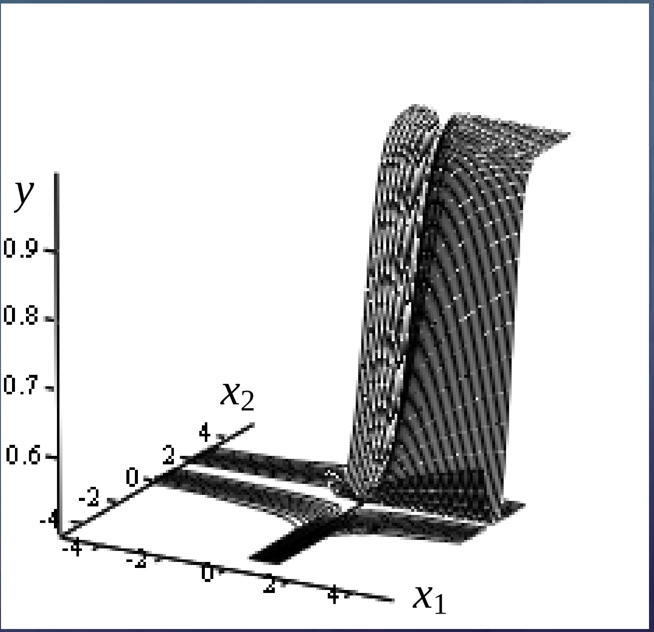
Characteristics of the standard neurons

Additive



$\tau = 0.2, w_1 = 1.0, w_2 = 2.0$

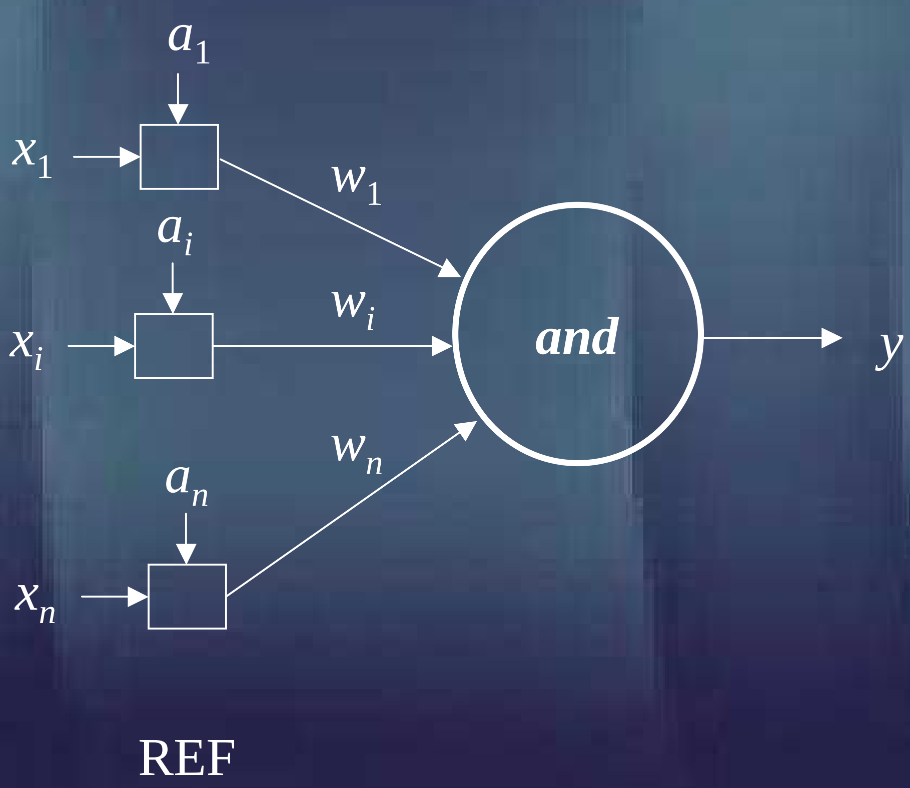
Multiplicative



$t_1 = 1.0, t_2 = 0.7, w_1 = 0.5, w_2 = 2.0$

$$g(u) = 1/(1 + \exp(-u))$$

Reference neurons



$$y = \mathop{\mathrm{T}}_{i=1}^{n} (\mathrm{REF}(x_i, a_i)\, s\, w_i)$$

$$\mathrm{REF}(x, a) = \begin{cases} \mathrm{INCL}(x, a) \\ \mathrm{DOM}(x, a) \\ \mathrm{SIM}(x, a) = \mathrm{INCL}(x, a)\ t\ \mathrm{DOM}(x, a) \end{cases}$$

$$\mathrm{INCL}(x, a) \equiv x \Rightarrow a$$

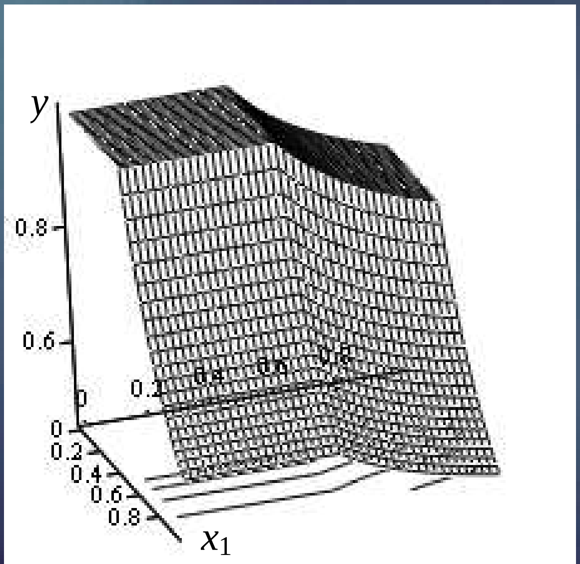
$$\mathrm{DOM}(x, a) \equiv a \Rightarrow x$$

$$\mathrm{SIM}(x, a) \equiv (x \Rightarrow a)\ t\ (a \Rightarrow x)$$

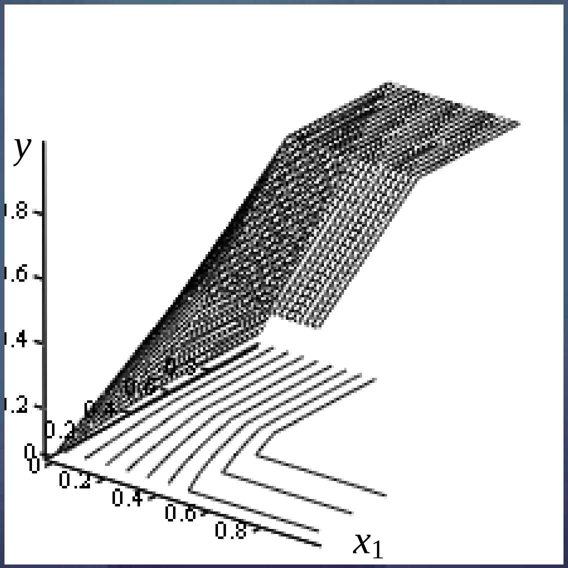
$$a \Rightarrow b = \sup\{ c \in [0,1] \mid a\ t\ \mathrm{c} \leq b\} \quad a,b \in [0, 1]$$

Characteristics of the reference neurons

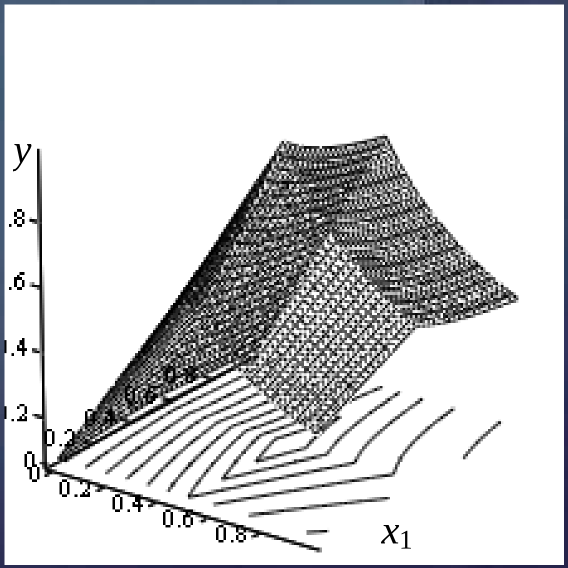
Inclusion



Dominance



Similarity



t-norm: product
s-norm: probabilistic sum

$w_1 = 0.1, \quad w_2 = 0.7$
$t_1 = 0.5, \quad t_2 = 0.5$

12.3 Uninorm-based fuzzy neurons

Main classes of unineurons

- ***and*** unineurons

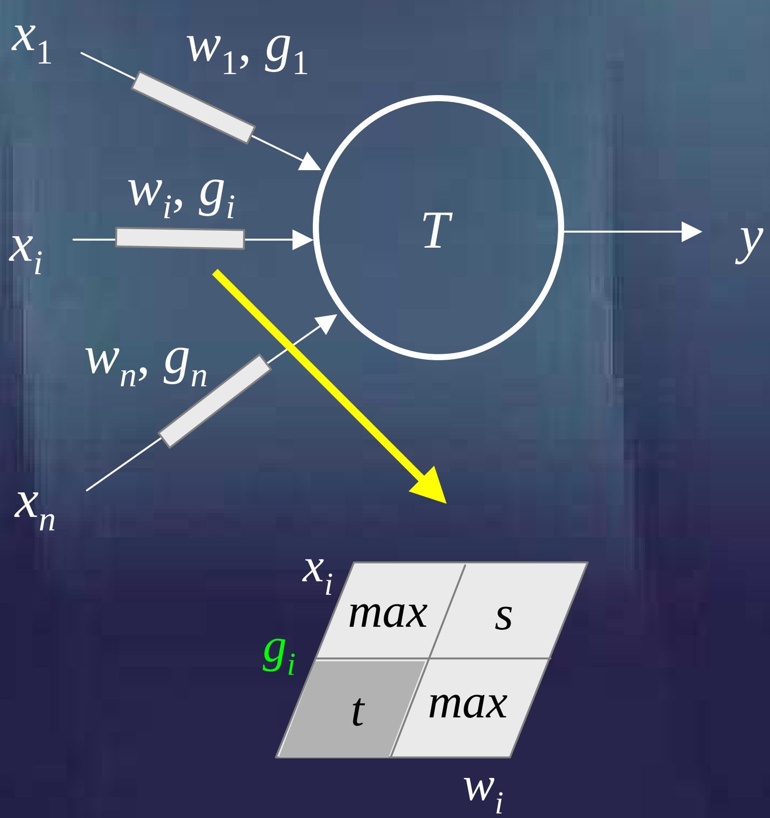
  and_U

- ***or*** unineurons

  or_U

and Unineurons

y = AND_U(**x;w,g**)

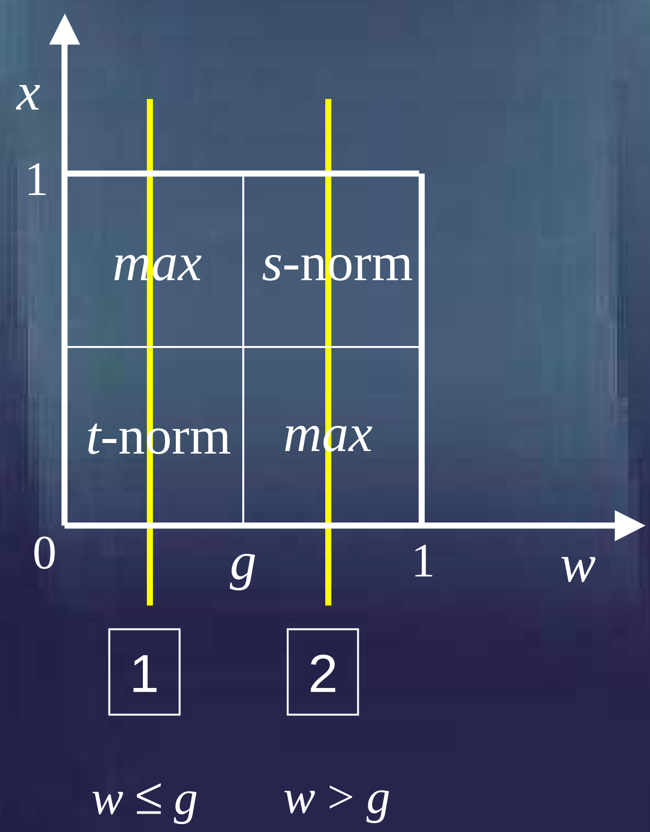


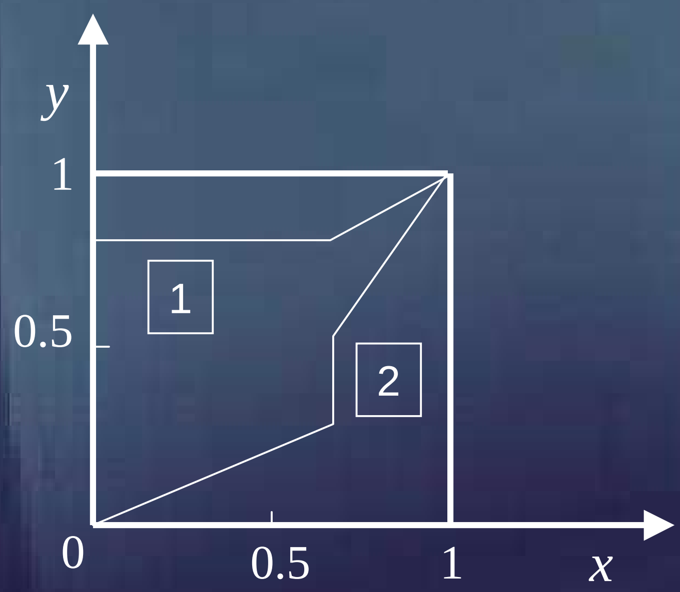
$$y = \mathop{\mathrm{T}}_{i=1}^{n} (u(x_i, w_i, g_i))$$

Pedrycz and Gomide, FSE 2007

Properties and characteristics of unineurons

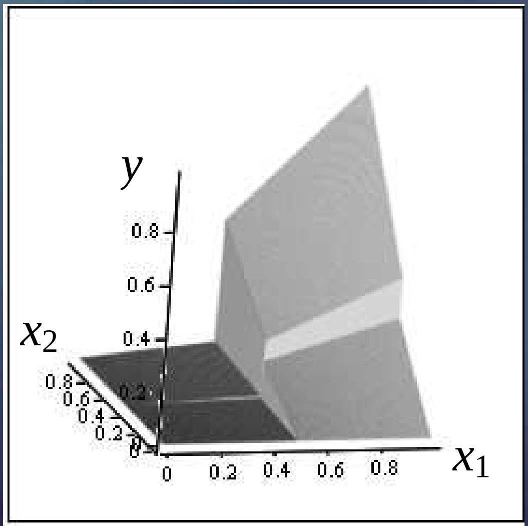
- Processing at the level of individual inputs: ***and*** neuron



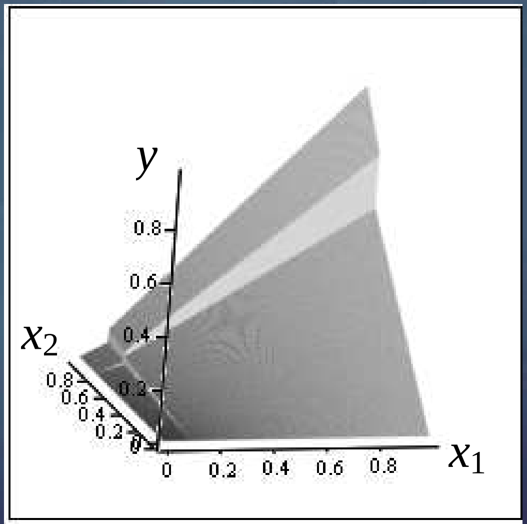


I/O characteristics of ***and*** unineurons

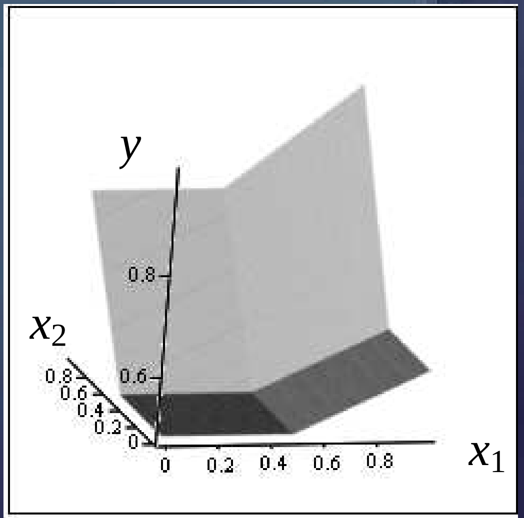
t-norm: product,   s-norm: probabilistic sum



w = [0.05, 0.30]
g = [0.50, 0.45]



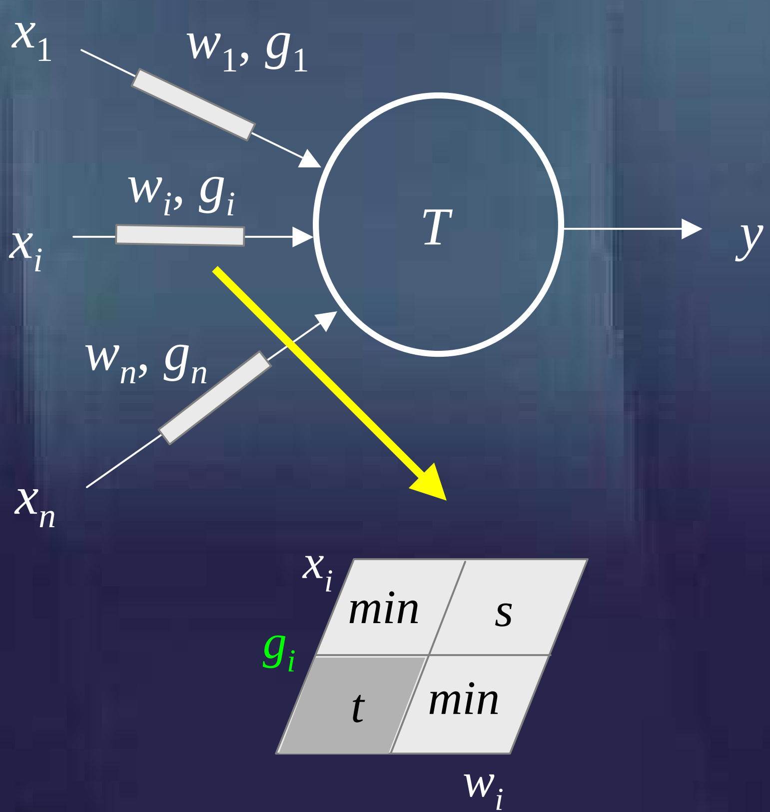
w = [0.05, 0.50]
g = [0.10, 0.80]



w = [0.80, 0.60]
g = [0.50, 0.50]

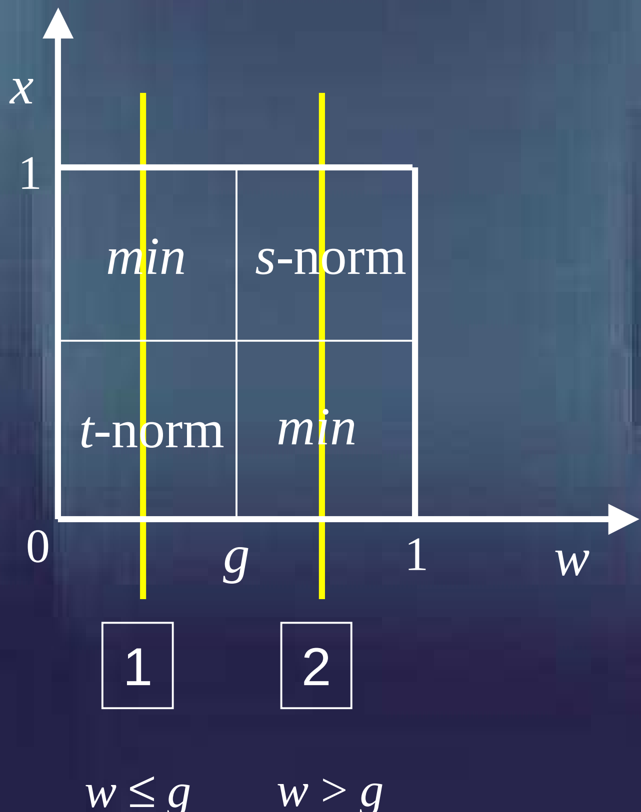
or Unineurons

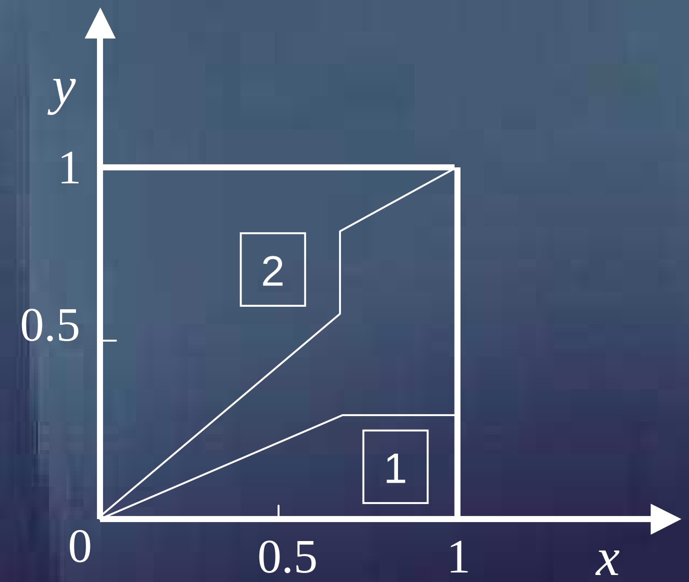
y = OR_U(**x;w,g**)



$$y = \overset{n}{\underset{i=1}{\mathrm{S}}} (u(x_i, w_i, g_i))$$

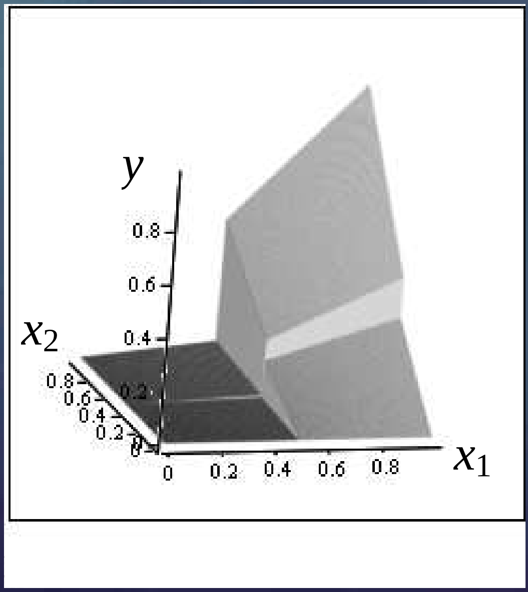
- Processing at the level of individual inputs: ***or*** neuron



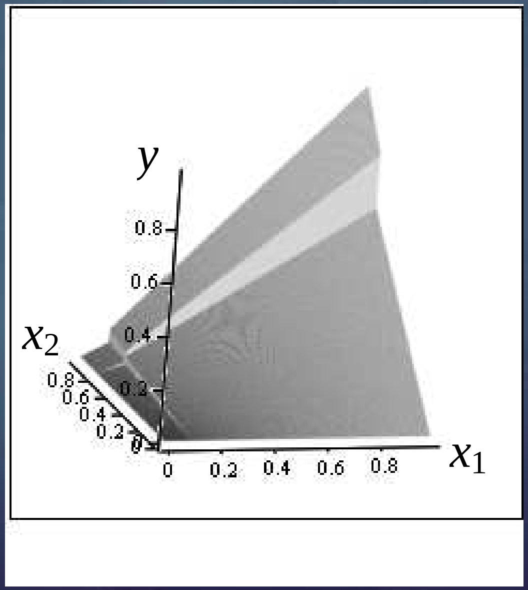


I/O characteristics of ***or*** unineurons

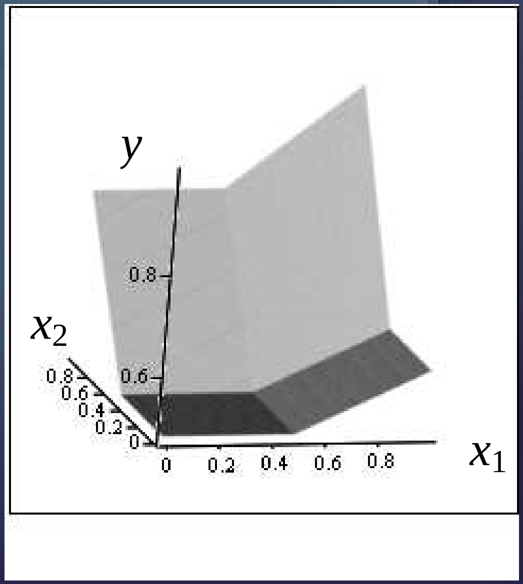
t-norm and s-norm: Lukasiewicz



w = [0.80, 0.50]
g = [0.30, 0.05]



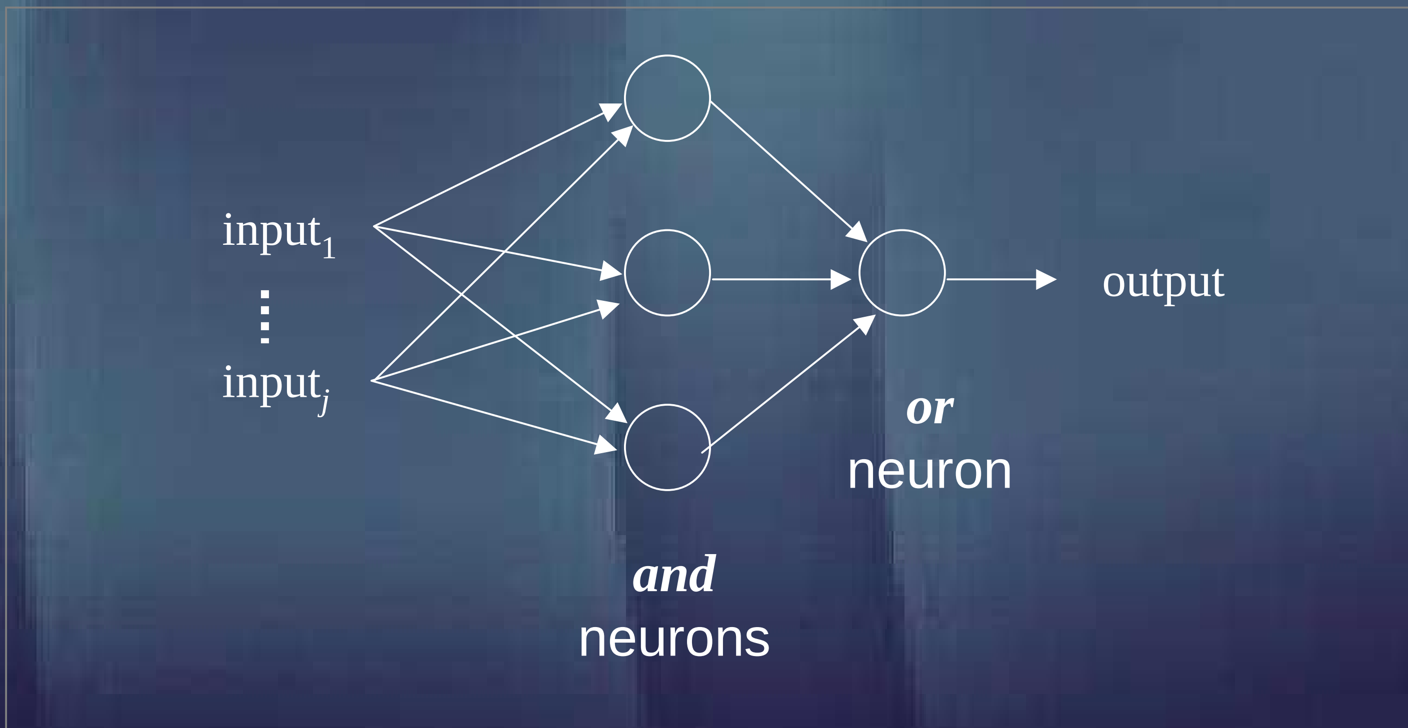
w = [0.20, 0.60]
g = [0.50, 0.40]



w = [0.60, 0.80]
g = [0.40, 0.10]

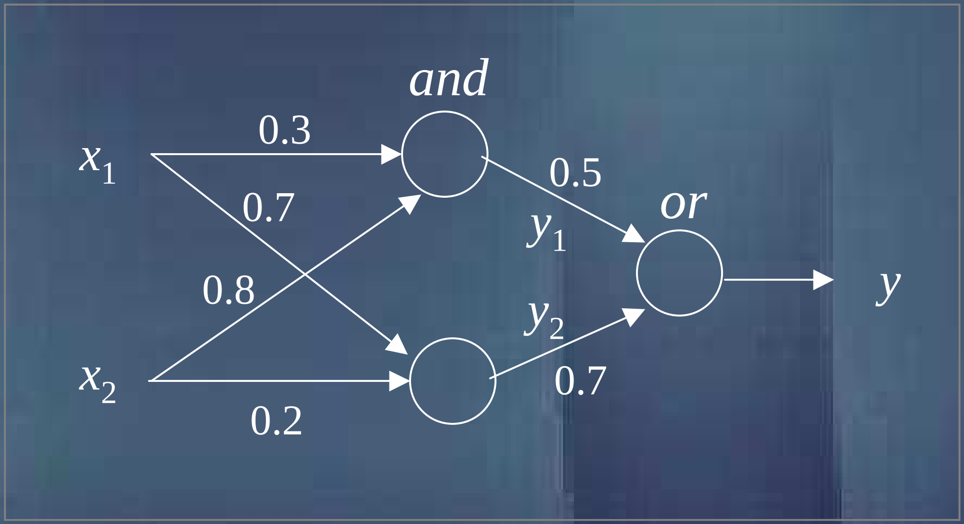
12.4 Architectures of logic networks

Logic processor: A canonical realization



if ($input_1$ *and* .....*and* $input_j$) *or* ($input_d$ *and* ....*and* $input_f$) **then** output

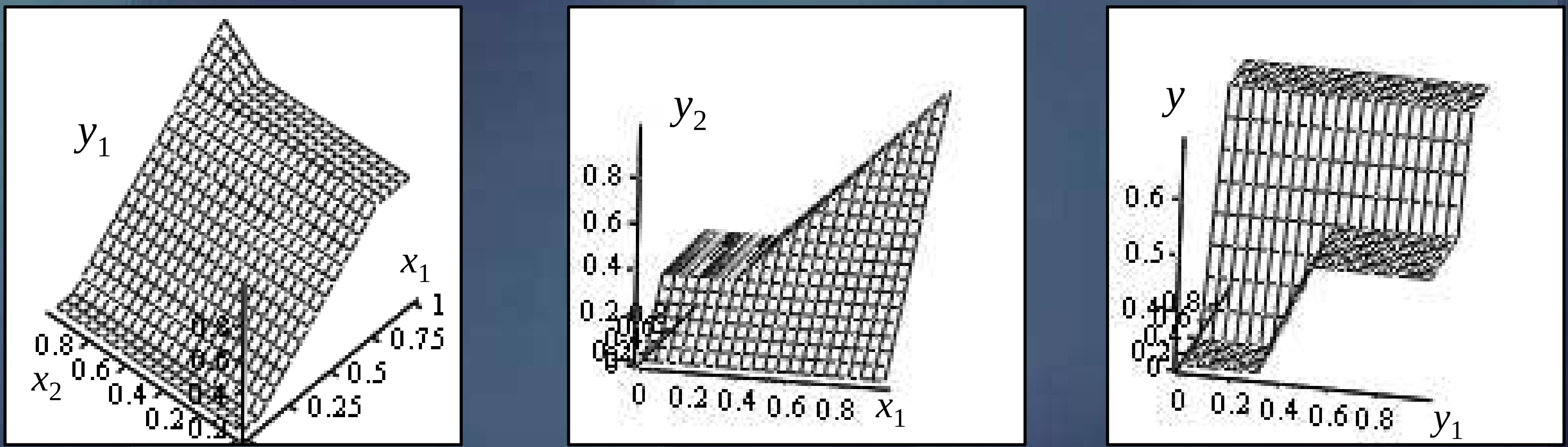
Examples



Logic processor

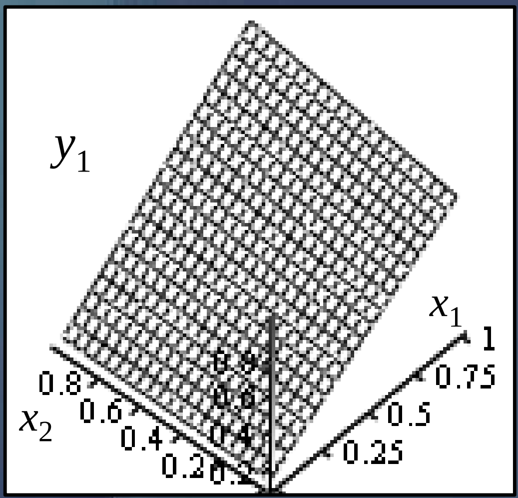
Outputs (a)

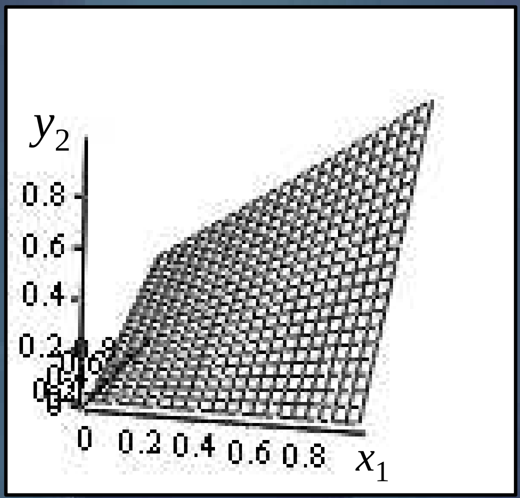
-

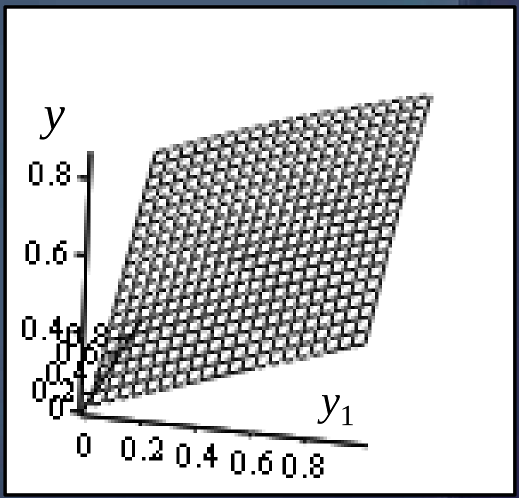


and = min, *or* = max

Outputs (b)



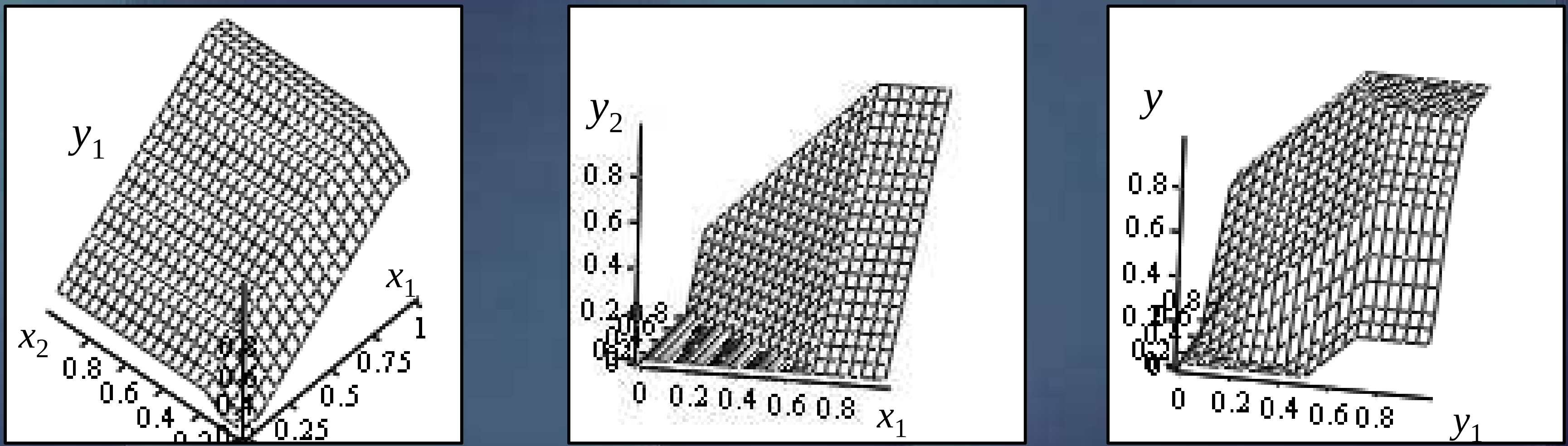




and = product, *or* = probabilistic sum

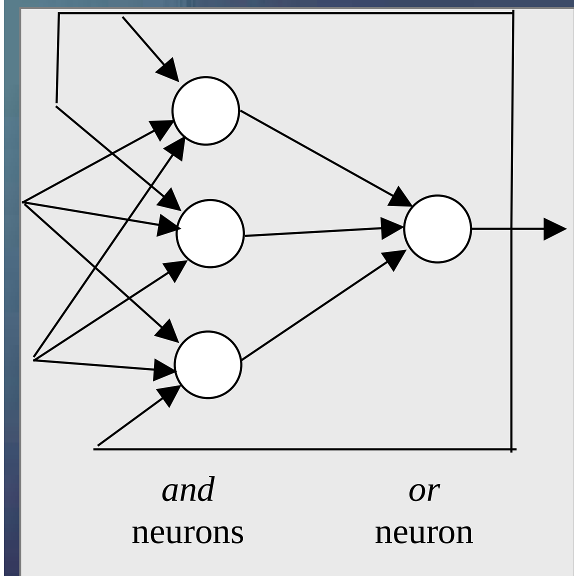
Outputs (c)

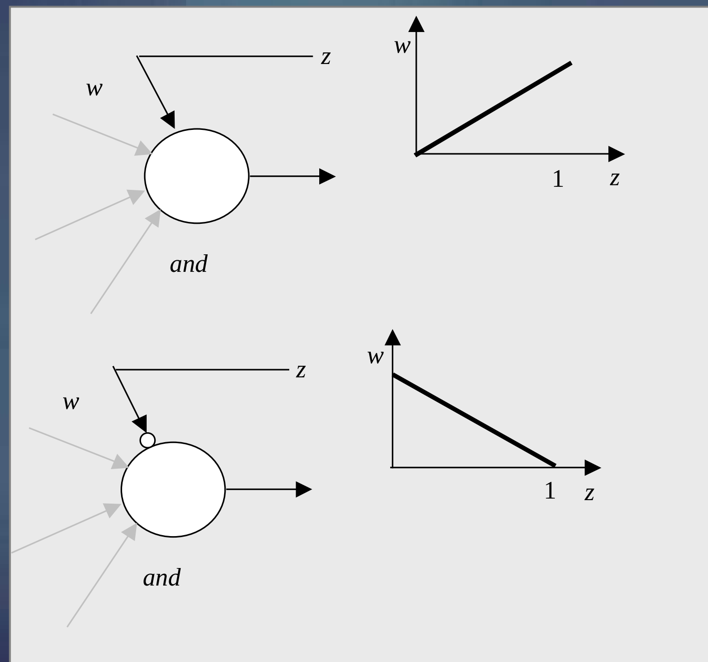
-

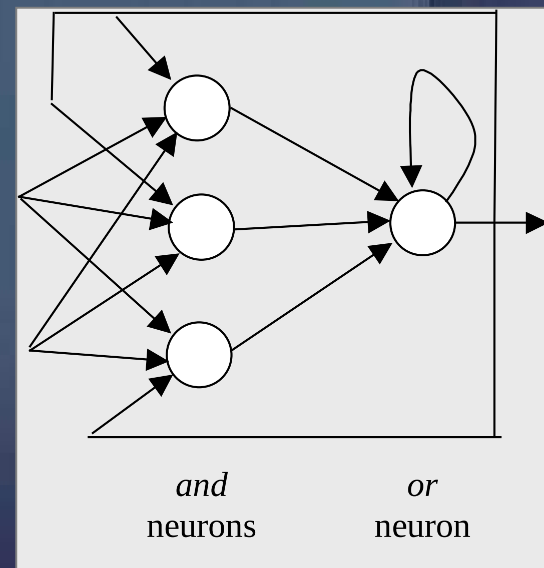


and = Lukasiewcz, *or* = Lukasiewcz

Fuzzy neural networks with feedback loops

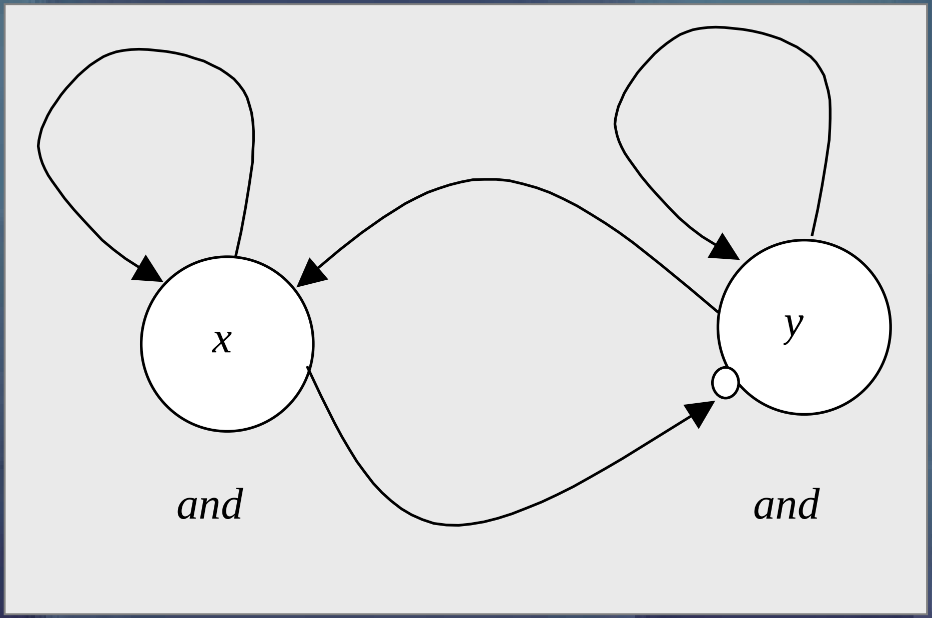






Excitatory and inhibitory connections

Example



$$x(k) = \text{And}(\mathbf{w},[x(k-1), y(k-1)])$$

$$y(k) = \text{And}(v,[y(k-1), y(k-2), \bar{x}(k-1)])$$

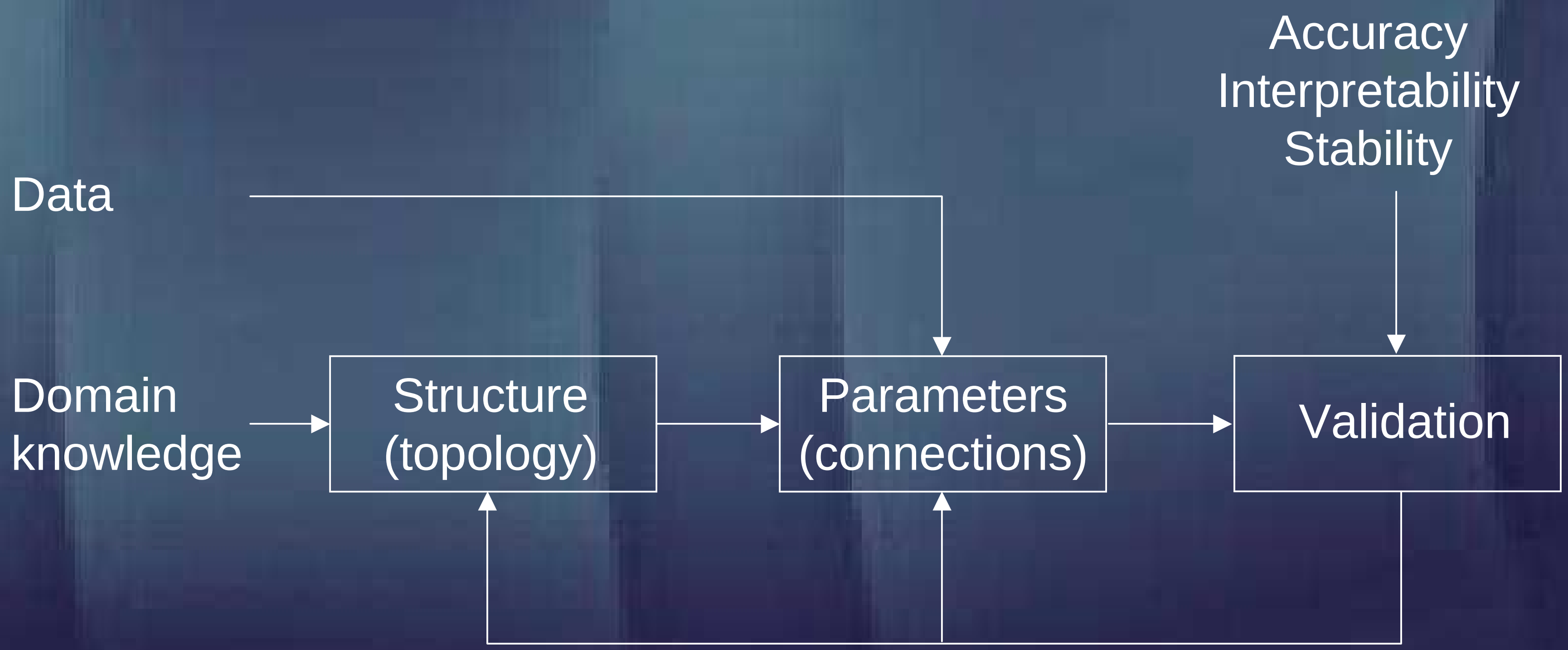
$$\bar{x}(k-1) = 1 - x(k-1)$$

11.5 The development mechanisms of the fuzzy neural networks

Development facets

- Structural learning
  - architecture (topology)
  - t norms
  - s norms
- Parametric learning
  - numeric values of connections

Key design phases



Gradient-based learning schemes for the networks

- Training data: input/output pairs $\{\mathbf{x}(k), target(k)\}$, $k = 1,2,\ldots, N$
- $\mathbf{x}(k) \in [0, 1]^n$
- $target(k) \in [0, 1]^m$
- Q is a performance index

$$\boxed{connection\,(\mathrm{iter}+1) = connection\,(\mathrm{iter}) - \alpha\nabla_{connection(\mathrm{iter})}Q}$$

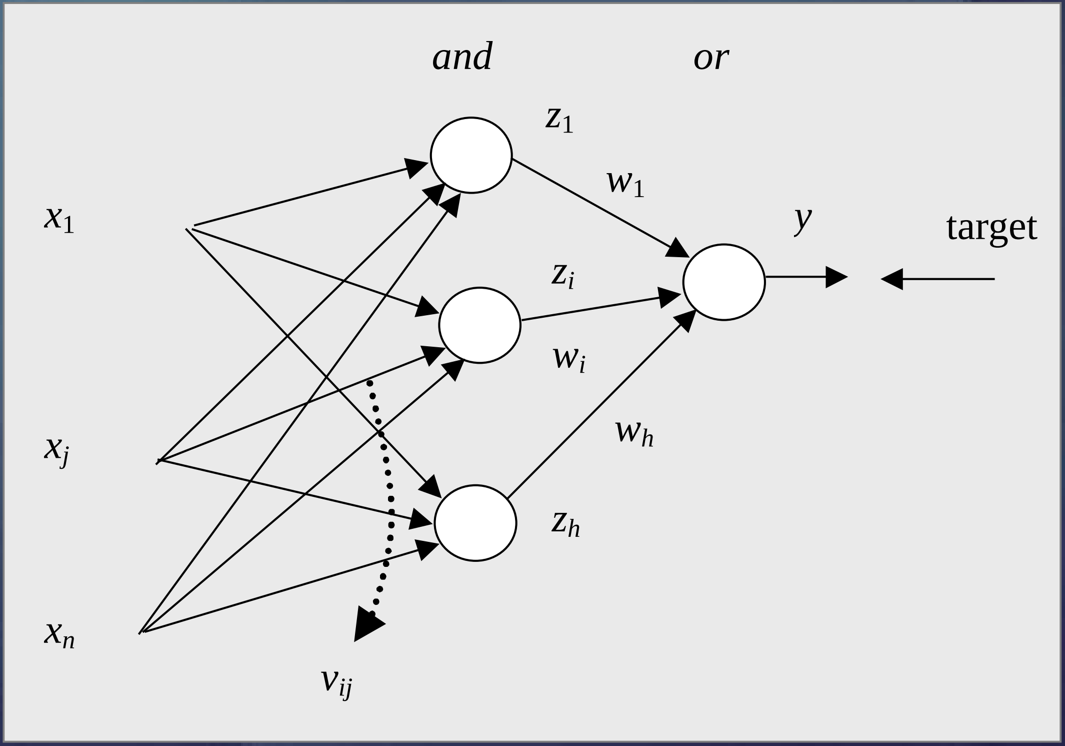
Basic scheme

Learning as an optimization problem

$$\min_{\mathbf{w},\mathbf{v}} Q = \sum_{k=1}^{N} (y(k) - \text{target}(k))^2$$

$$s.t. \quad \mathbf{w} \in [0,1]^{nh}$$

$$\mathbf{v} \in [0,1]^{h}$$



$$< connection\,(\text{iter} + 1) = connection\,(\text{iter}) - \alpha \nabla_{connection(\text{iter})} Q >$$

$< >$ is a truncation operation

$$\frac{\partial Q}{\partial w_i} = (y - \text{target})\frac{\partial y}{\partial w_i}$$

$$\frac{\partial y}{\partial w_i} = \frac{\partial}{\partial w_i}\left(\underset{j=1}{\overset{h}{S}}(w_j t z_j)\right) = \frac{\partial}{\partial w_i}[A_i s(w_i t z_i)]$$

$$A_i = \underset{\substack{j=1 \\ j \neq i}}{\overset{h}{S}}(w_j t z_j)$$

$$\min(x,w) = \begin{cases} w \text{ if } x \leq w \\ x \text{ if } x > w \end{cases}$$

$$\frac{\partial \min(x,w)}{\partial w} = \begin{cases} 1 \text{ if } x \leq w \\ 0 \text{ if } x > w \end{cases}$$

Particular min/max cases

$$\frac{\partial \min(x,w)}{\partial w} = ||w \subset x|| = w \Rightarrow x$$

Generalization min/max

$$\frac{\partial \max(x,w)}{\partial w} = ||x \subset w|| = x \Rightarrow w$$

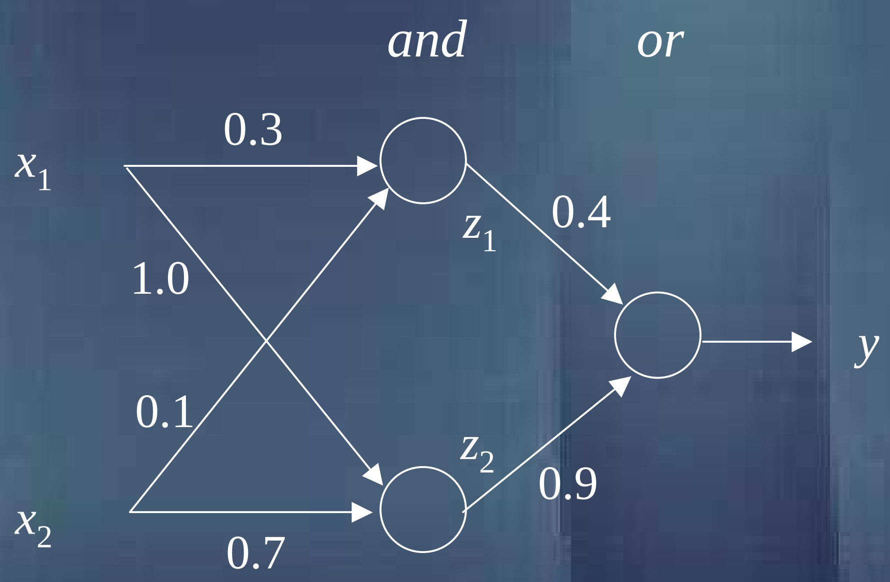
Development modes

- Successive expansions
  – increase the size of the network
- Successive reductions
  – prune “weakest” connections

12.6 Interpretation of fuzzy neural networks

- *or* neurons
  - – weighted *or* combination of the inputs
  - – high value of the connection ⇒ higher influence of the corresponding input
- *and* neurons
  - – weighted and combination of the inputs
  - – low value of the connection ⇒ higher influence of the corresponding input
- Rule generation
  - – Step 1: start with highest value of the *or* connection
  - – Step 2: translate *and* neuron into and combination of the inputs

Example



Step 1: if $z_{2/0.9}$ or $z_{1/0.4}$ then y

Step 2: if $[x_{2/0.7}]_{0.9}$ *or* $[x_{2/0.1}$ *and* $x_{1/0.3}]_{0.4}$ then y

Retention of the most significant connections

- Reducing weakest connections to 0 or 1

  – define a threshold mapping ϕ_θ: $[0,1] \rightarrow [\theta\ , 1] \cup \{0\}$
  – thresholds λ and μ

- *or* neurons $\theta = \lambda$

$$\phi_\lambda(w) = \begin{cases} w & \text{if} \quad w \geq \lambda \\ 0 & \text{if} \quad w < \lambda \end{cases}$$

- *and* neurons $\theta = \mu$

$$\phi_\mu(w) = \begin{cases} 1 & \text{if} \quad w > \mu \\ w & \text{if} \quad w \leq \mu \end{cases}$$

Conversion of the fuzzy network to the Boolean version

- *or* neurons $\phi_\lambda : [0,1] \rightarrow \{0, 1\}$

$$\phi_\lambda(w) = \begin{cases} 1 & \text{if} \quad w \geq \lambda \\ 0 & \text{if} \quad w < \lambda \end{cases}$$

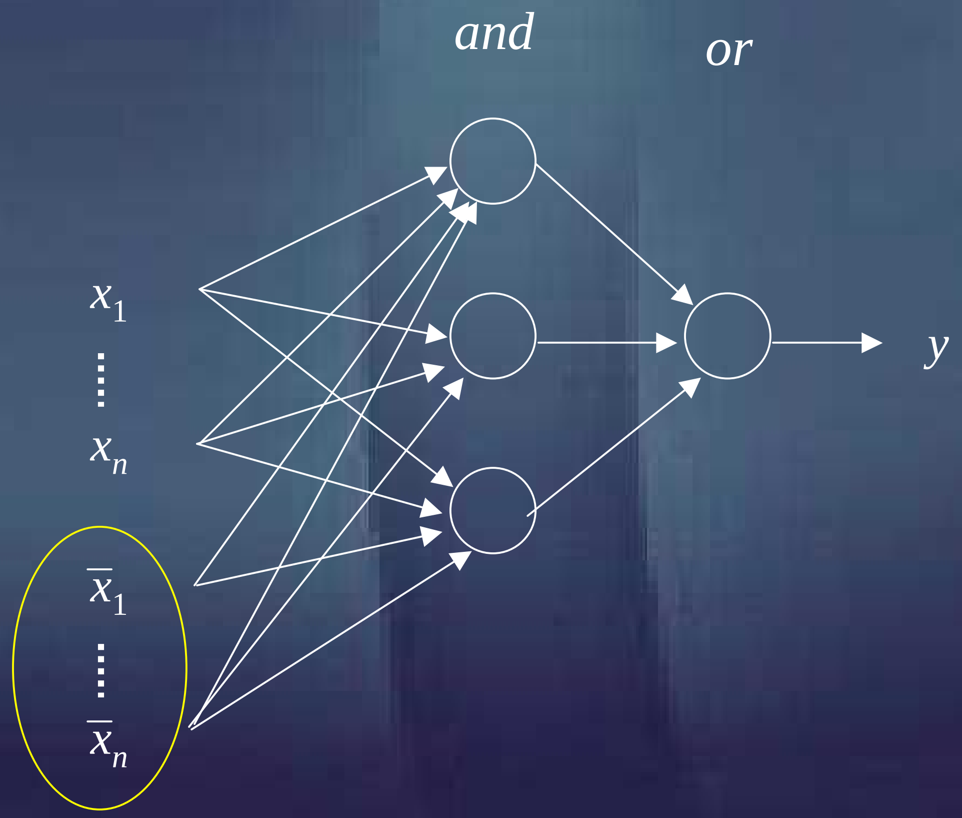
- *and* neurons $\phi_\mu : [0,1] \rightarrow \{0, 1\}$

$$\phi_\mu(w) = \begin{cases} 1 & \text{if} \quad w > \mu \\ 0 & \text{if} \quad w \leq \mu \end{cases}$$

12.7 From fuzzy logic networks to Boolean functions and their minimization through learning

- *and* and *or* neurons generalize (subsume) *and* and *or* logic gates
- Logic functions are encoded by fuzzy logic networks
- Logic functions may involve complements of the original variables
- After reducing connections to Boolean versions
  - – simplification of Boolean functions usually with Karnaugh maps
  - – networks simplifies using learning instead

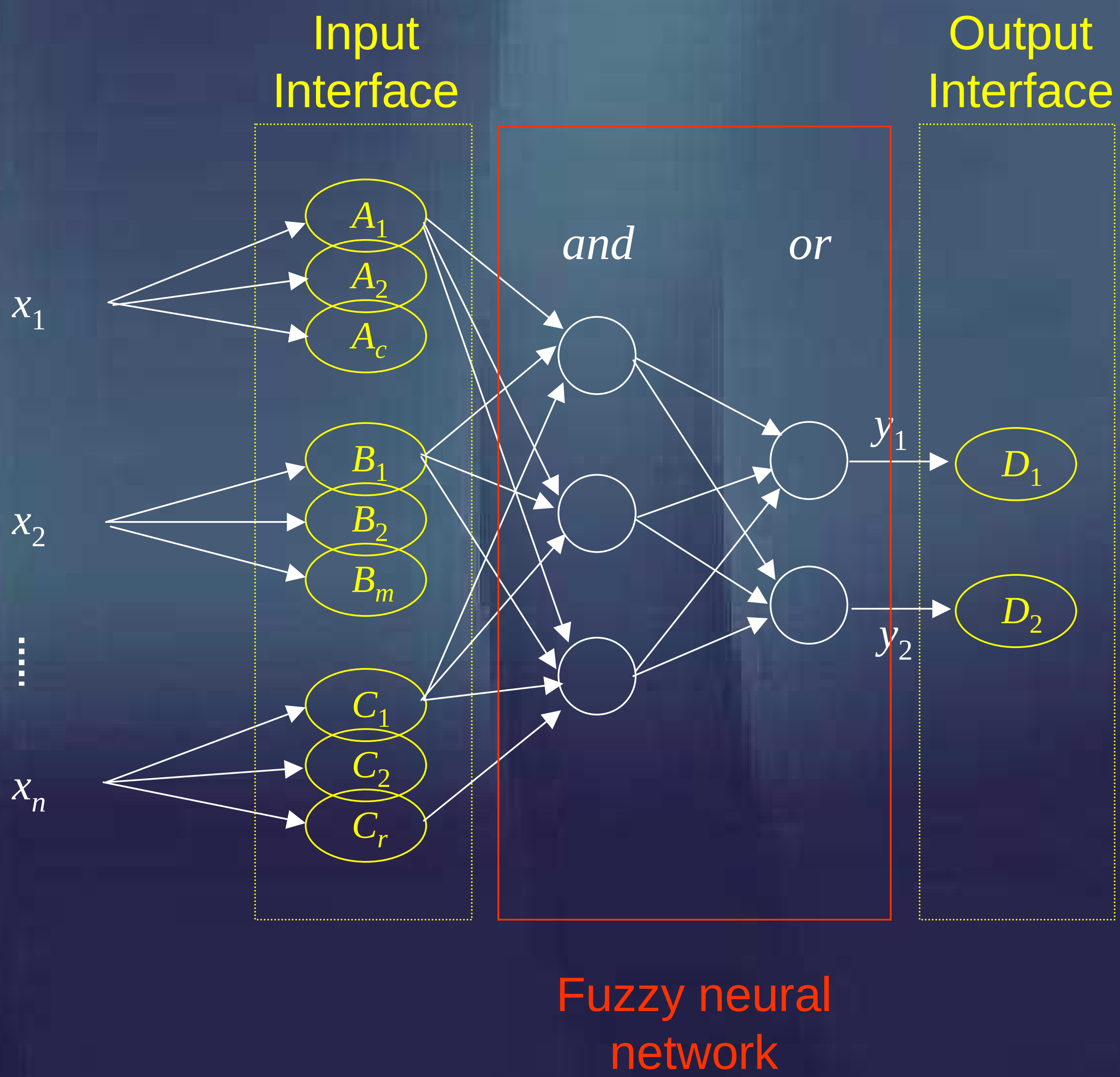
Fuzzy logic networking learning Boolean function



Complemented inputs

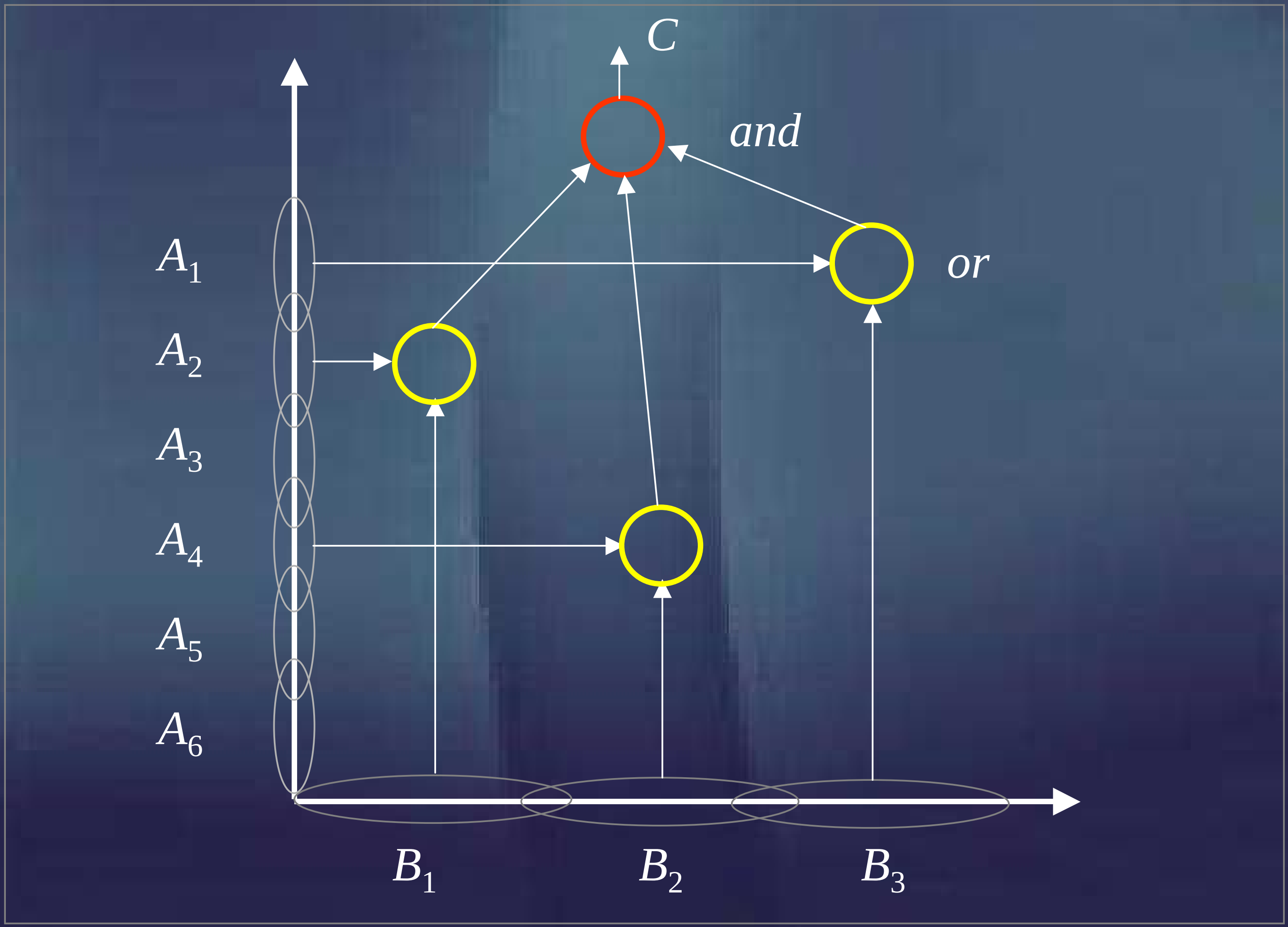
12.8 Interfacing the fuzzy neural network

Overall architecture



Pedrycz and Gomide, FSE 2007

Geometry of rules supported by fuzzy neural nets

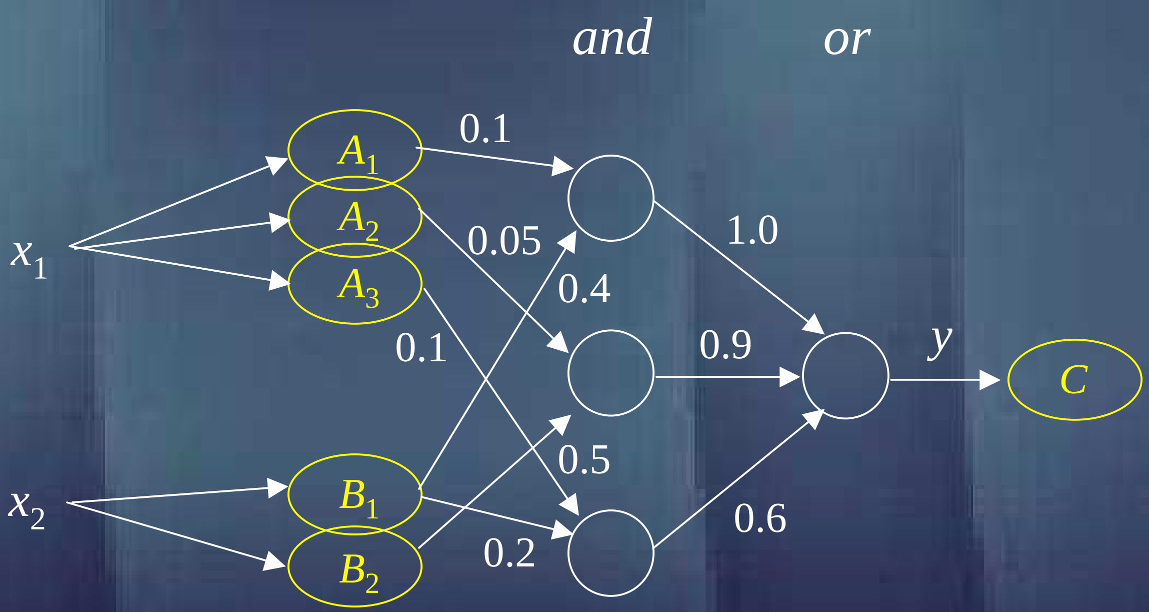


if (A_4 and B_2) or (A_2 and B_1) or (A_1 and B_3) then C

Pedrycz and Gomide, FSE 2007

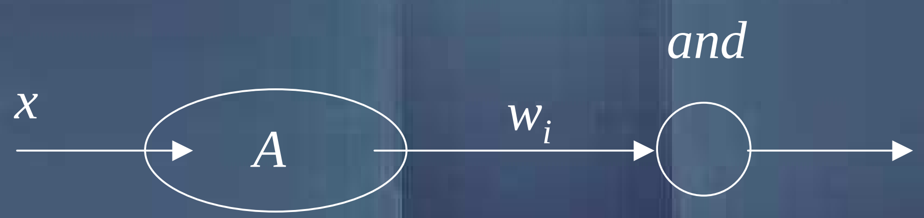
12.9 Interpretation aspects: A refinement of induced rule-based system

Example



If $[A_{1/0.1}$ and $B_{1/0.4}]_{1.0}$
or
$[A_{2/0.05}$ and $B_{2/0.5}]_{0.9}$
or
$[A_{3/0.1}$ and $B_{1/0.2}]_{0.6}$
then
C

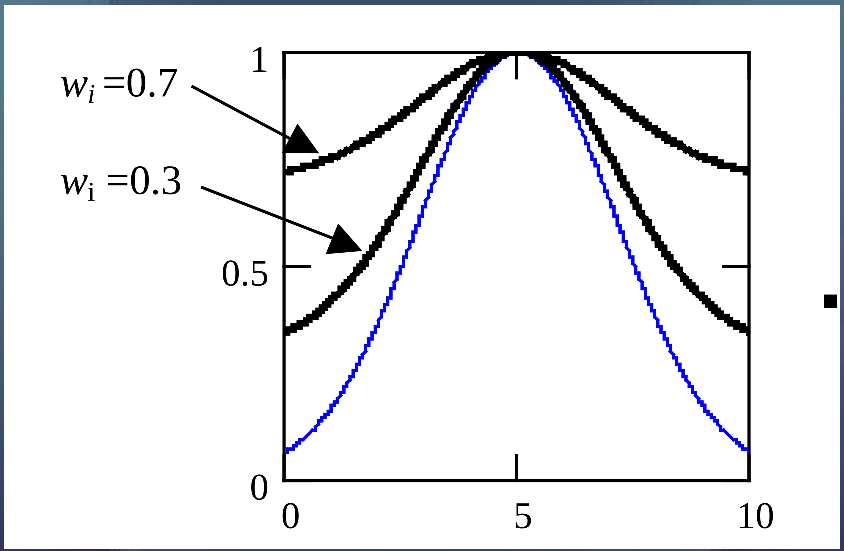
- Transformation of fuzzy set A of interface through w_i
  - leads to $\tilde{A}$
  - higher values of w_i make $\tilde{A}$ close to one



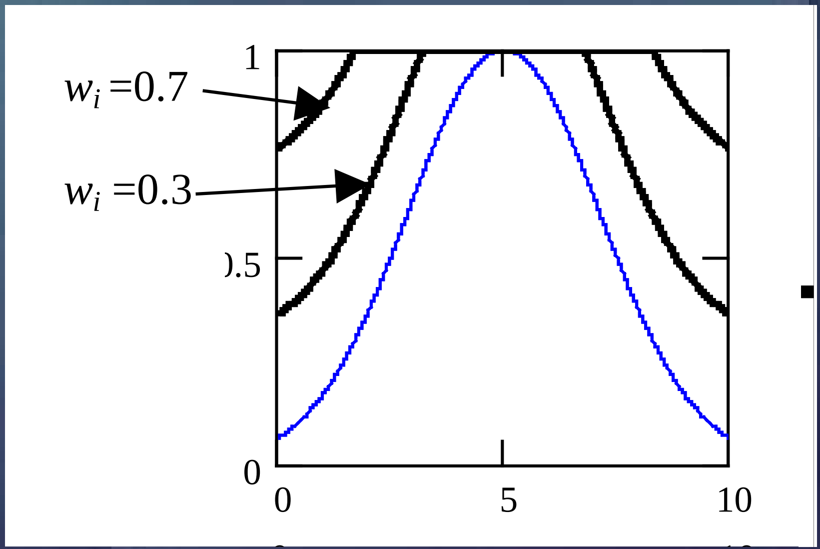
$$x_i = A(x)$$

$$\tilde{A}(x) = x_i \; s \; w_i = A(x) \; s \; w_i$$

Original fuzzy set of the interface: Gaussian

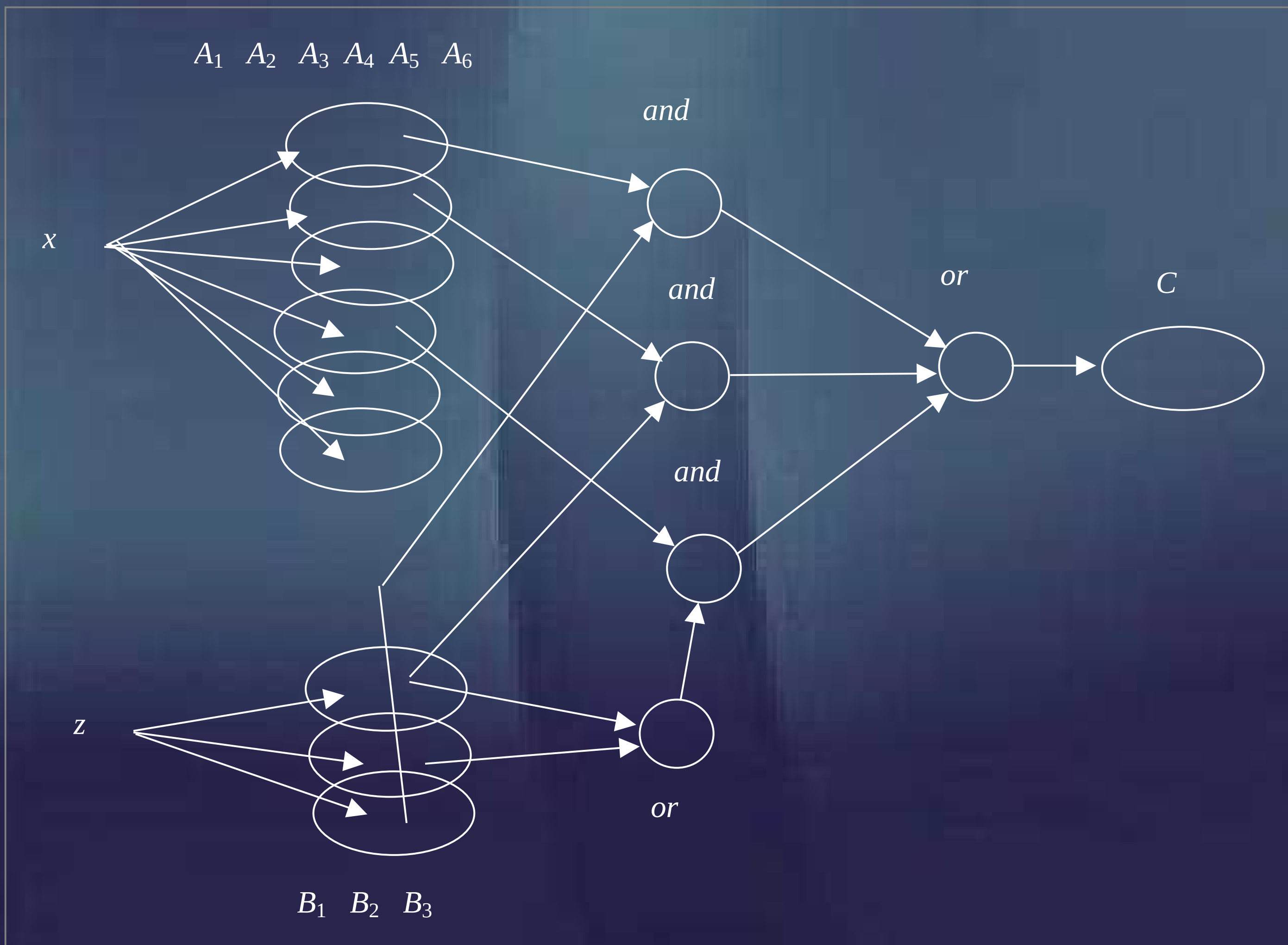


(a) t-conorm: probabilistic sum

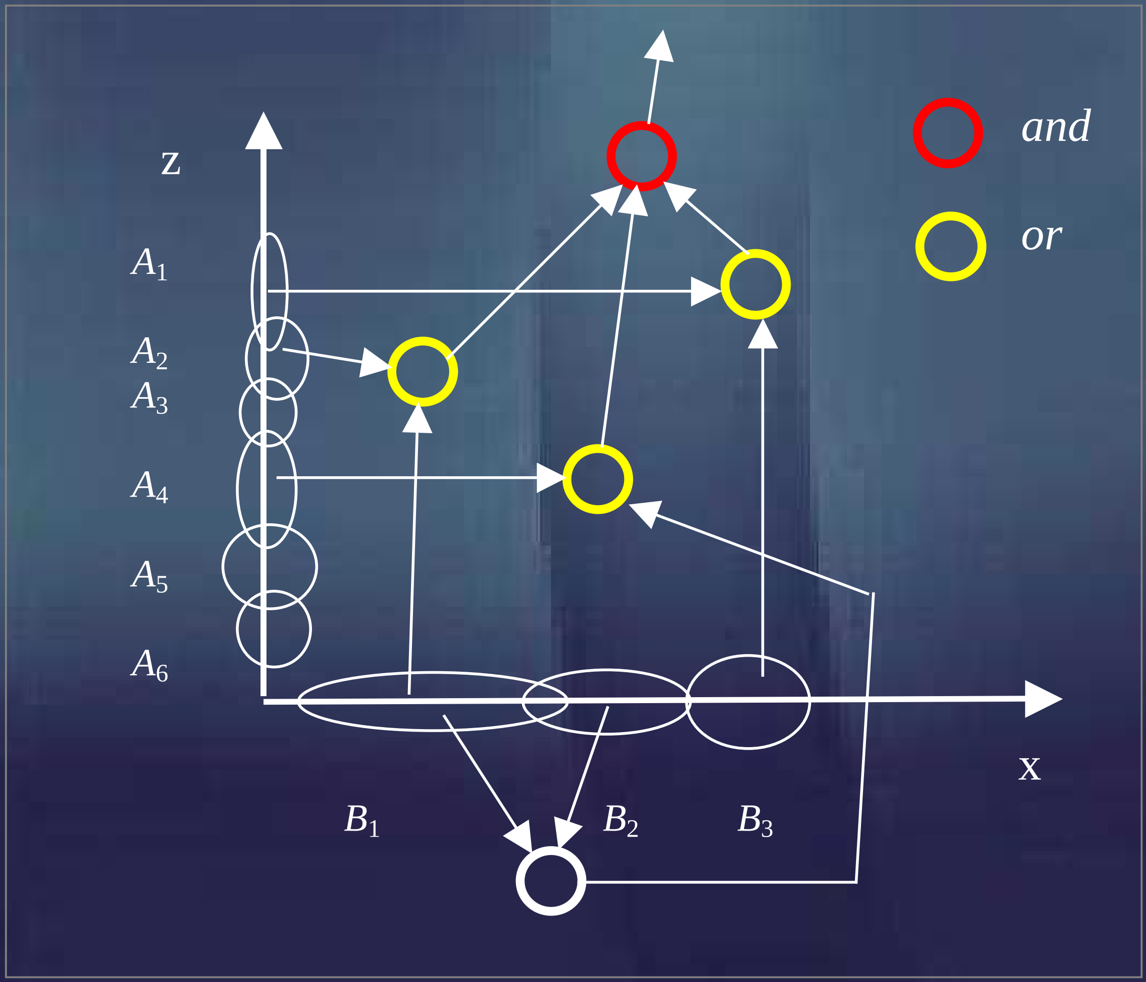


(b) t-conorm: Lukasiewicz

Example of augmented fuzzy neural network



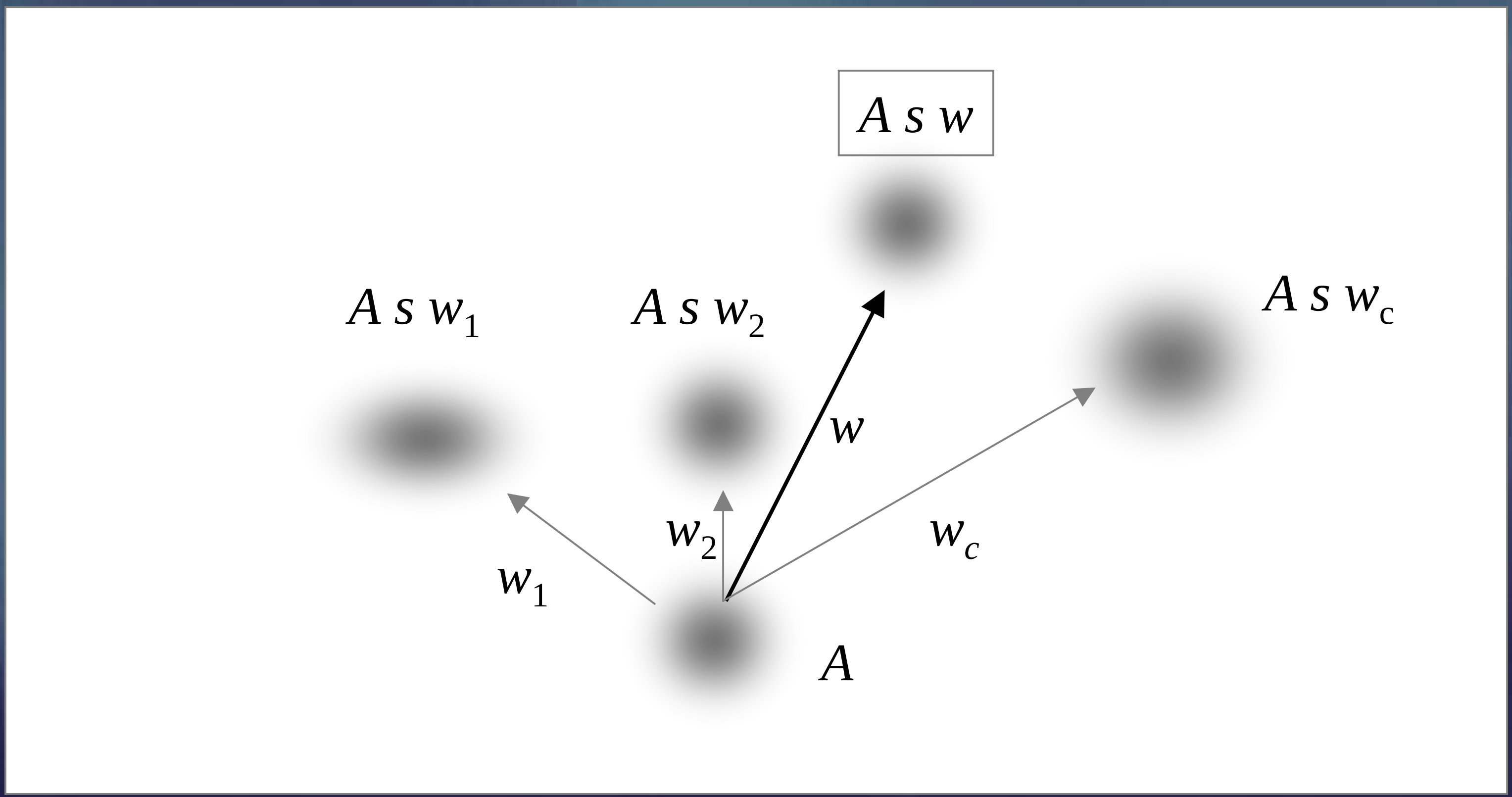
The underlying geometry



12.10 Reconciliation of perception of information granules and granular mappings

- Information granules
  - can be perceived in different ways
  - perception depends on the context
- Modeling perception
  - logic oriented transformation of fuzzy sets
  - mechanism of reconciliation
- Reconciliation of various perceptions viewed as an optimization

Reconciliation of perception of information granule



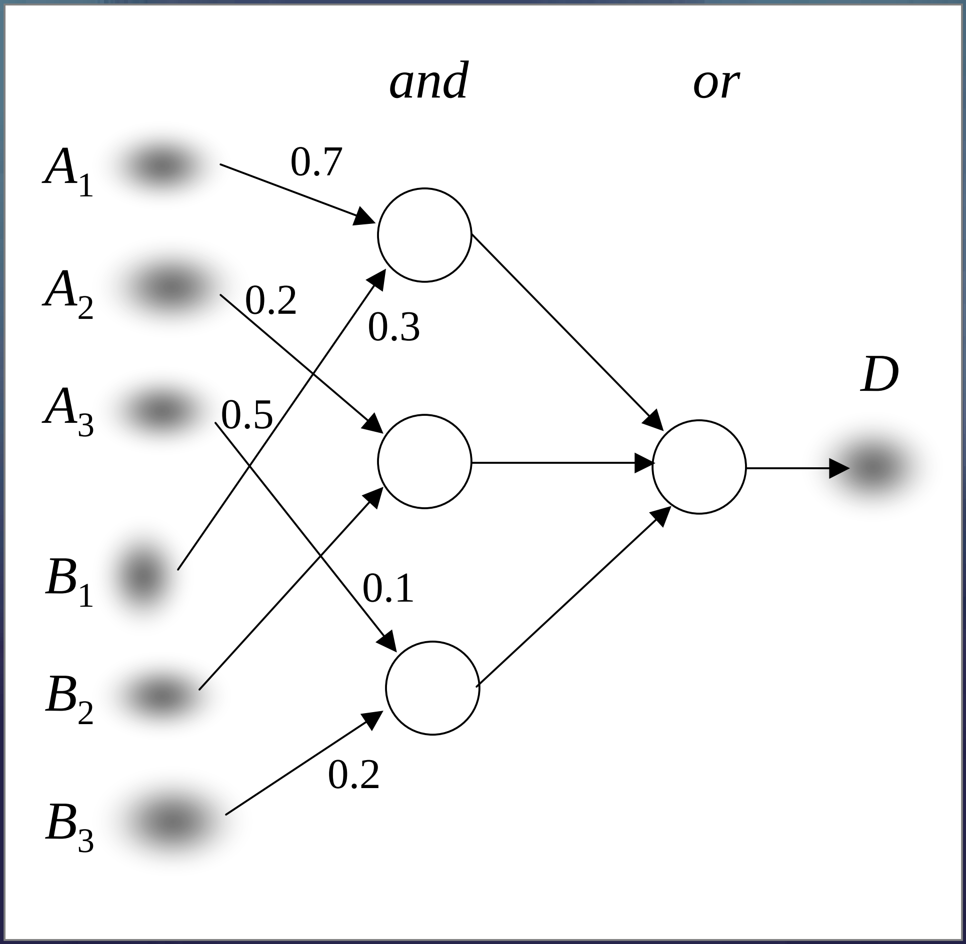
$$\tilde{A}(x) = A(x)\ s\ w, \qquad w \in [0,1]$$

The optimization process

$$\min_{\mathbf{w}} Q = \sum_{i=1}^{c} \int_X [A(x)sw_i - A(x)sw]^2 \, dx$$

$$s.t. \quad \mathbf{w} \in [0,1]$$

An application of the perception mechanism to fuzzy rule-based systems



If

{[(A_1 *or* 0.7) and (B_1 *or* 0.33)] *and* 0.9}

or

{[(A_2 *or* 0.2) and (B_2 *or* 0.50)] *and* 0.7}

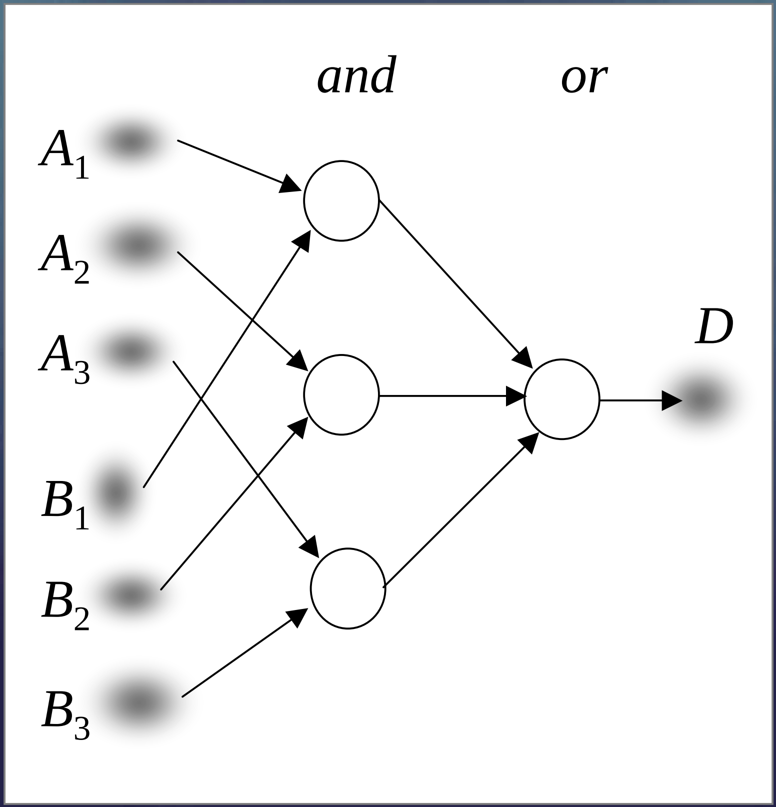
or

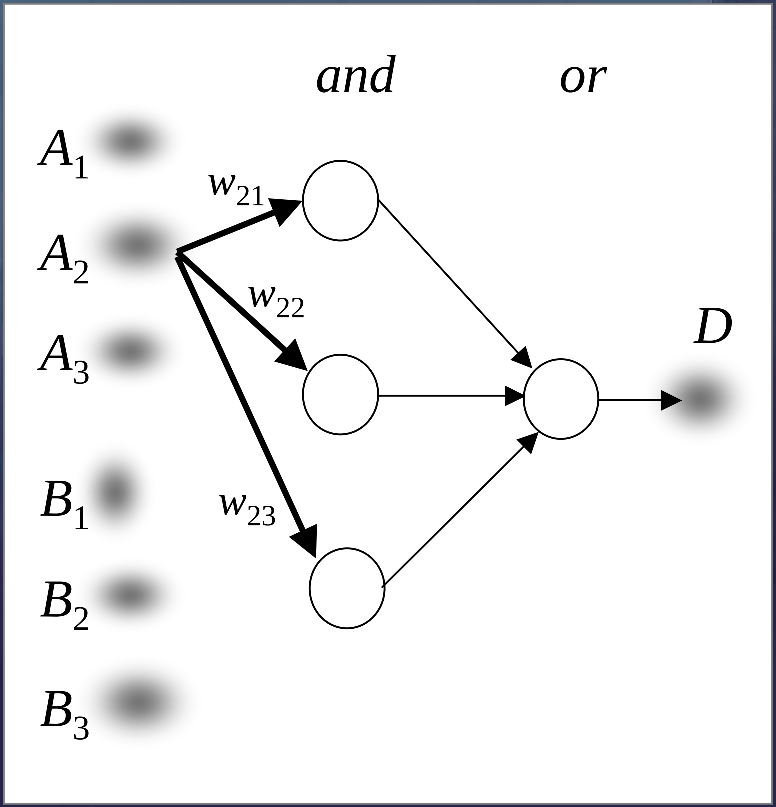
{[(A_3 *or* 0.1) and (B_3 *or* 0.20)] *and* 1.0}

then

D

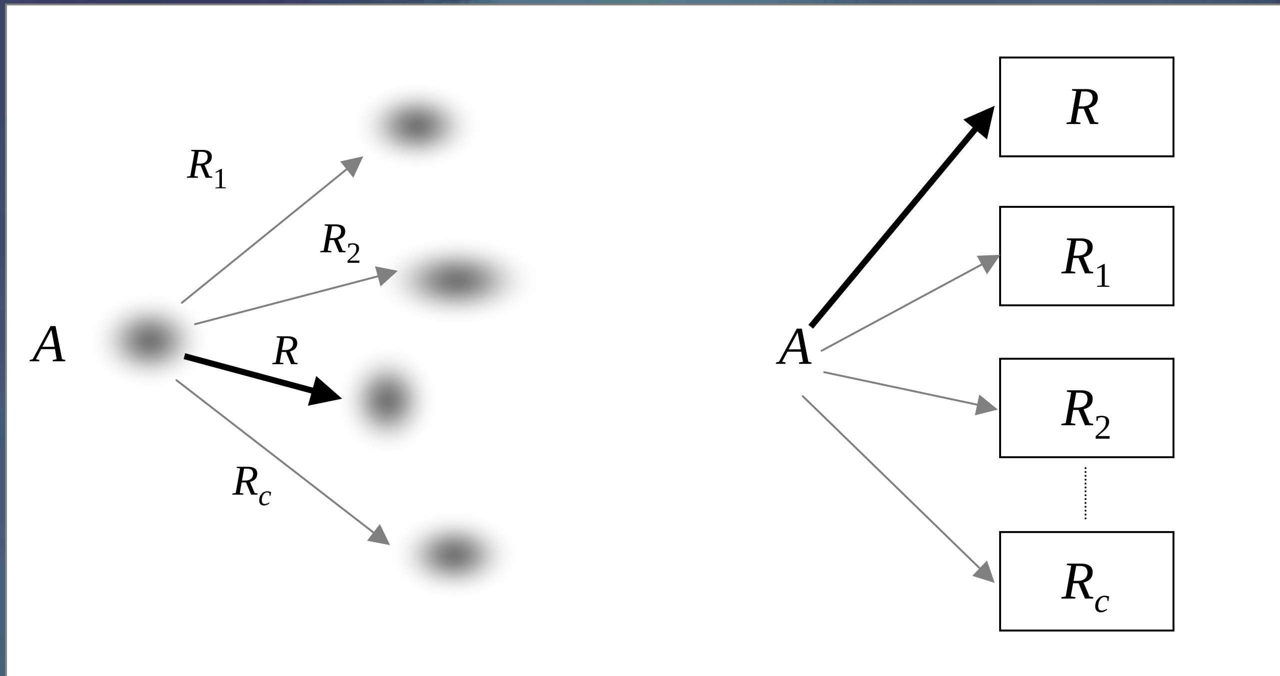
Reconciliation of impact of the input on individual *and* nodes through optimization





Reconciliation of granular mappings

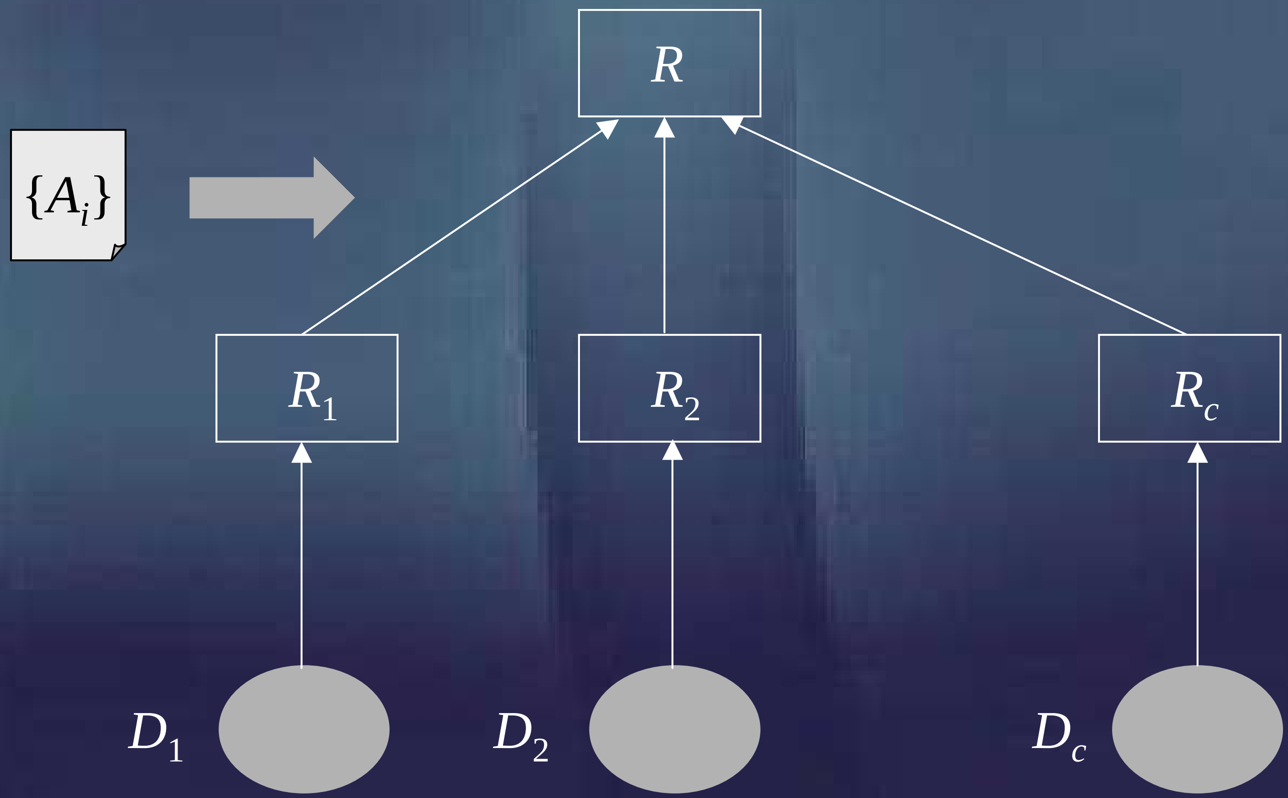
- Problem
  - – R_i: $\mathbf{X} \rightarrow \mathbf{Y}$, $i = 1,\ldots,c$ are given
  - – R_i are relational mappings
  - – determine R such that it forms a reconciliation with R_i´s
- Reconciliation involves a family of fuzzy sets $A_1, A_2,\ldots,A_N$



$$Q=\sum_{i=1}^{c}\| A\circ R_i - A\circ R\|^2$$

$$Q=\sum_{l=1}^{N}\sum_{k=1}^{c}\| A_l\circ R_k - A_l\circ R\|^2$$

Reconciliation of fuzzy models with granular probes $\{A_i\}$



$$Q = \sum_{i=1}^{c} \| A \circ R_i - A \circ R \|^2$$

$$= \sum_{l=1}^{N} \sum_{k=1}^{c} \sum_{j=1}^{m} \left(\underset{i=1}{\overset{n}{S}} (A_l(x_i) t R_k(x_i, x_j)) - \underset{i=1}{\overset{n}{S}} (A_l(x_i) t R(x_i, x_j)) \right)^2$$

$$R(\text{iter}+1) = R(\text{iter}) - \alpha \nabla_R Q$$