LOGIC NEURAL NETWORKS AS A FRAMEWORK FOR USER-CENTRIC COMPUTING

Witold PEDRYCZ

Dept. of Electrical & Computer Eng.
University of Alberta,Edmonton, Canada
&
Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

March 2005

Agenda

Introductory Comments: Neurofuzzy models

General architecture of neurofuzzy models

Granular Computing as a modeling framework

Logic Processing

Generic fuzzy neurons

Architectures of fuzzy logic processing and their development

Combinational and sequential fuzzy models

Neurofuzzy systems: a general view

FUZZY SETS

NEURAL NETWORKS

Available data

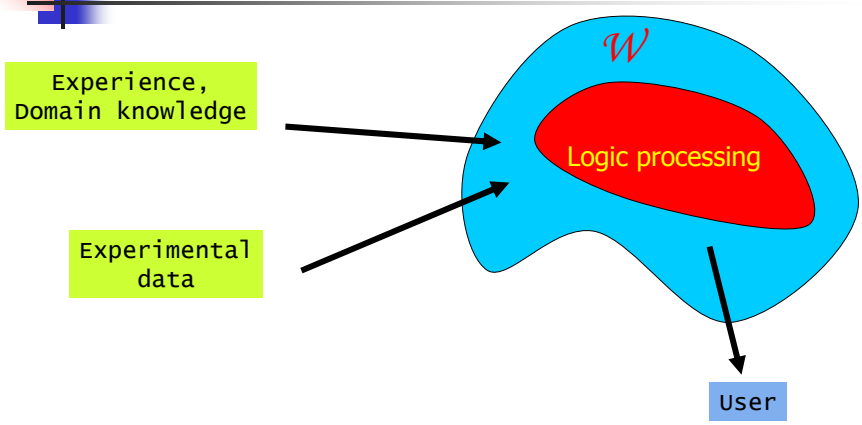
Design objectives

Architectures
Learning schemes

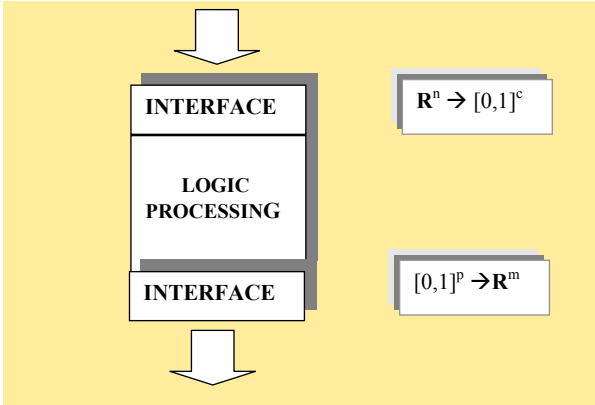
accuracy
interpretability

Rule-based systems
Structural and parametric learning

Fuzzy Models: General Architecture



General Topology



Fuzzy and Fuzzy Models: Development Guidelines

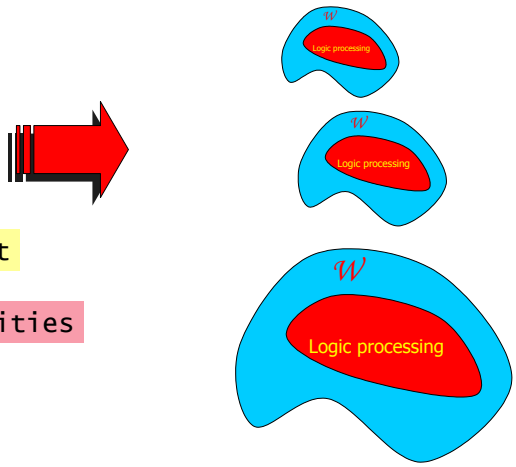
Interpretability

Nature of data

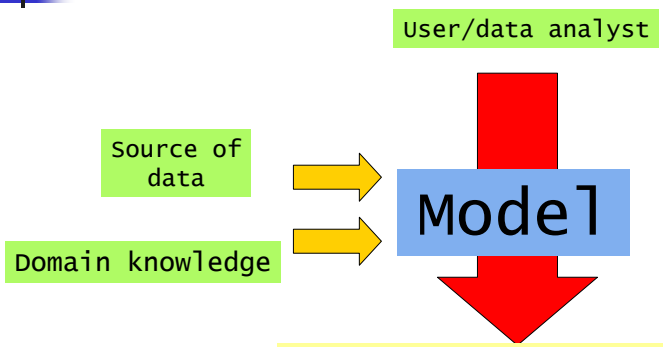
Accuracy

Computational effort

Generalization abilities



Fuzzy Modeling: Rationale



- User's preferences
- Transparency of model
- "What-if" analysis available
- Accuracy-interpretability
- Relevance analysis (generalization..)

Logic Processing

- Interpretability and transparency of results
- Ability to accommodate and calibrate prior domain knowledge
- Transparency – Plasticity (learning) tradeoff

Logic and Computing

… we have seen how this leads to a lower level of arithmetical precision but to a higher level of logical reliability: a deterioration in arithmetic has been traded for an improvement in logic…

John von Neumann, 1958

Logic networks

S.A. Kauffman, Metabolic stability and epigenesis in randomly connected nets, J. Theor. Biol. 22 (1969) 437-467.

Qualitative modeling

Logic expressions

Logic expressions
- **variables** → **Information granules**
- **operators** → **(Logic) connectives**

$$W = L(\underbrace{A,\ B,\ C\ldots}_{\text{Information granules}},\ \text{connectives},\ \underbrace{a}_{\text{parameters}})$$

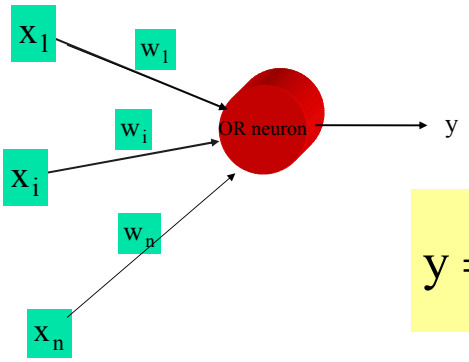
Triangular norms (t- and s-norms)

Generic models of logic operations in fuzzy logic

t-norm: AND operation ↓ **s-norm: OR operation** ↓

Correspondence principle
if [0,1] replaced by {0,1}
→ logic operators of two-valued logic

OR Neuron

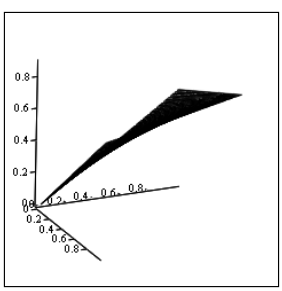


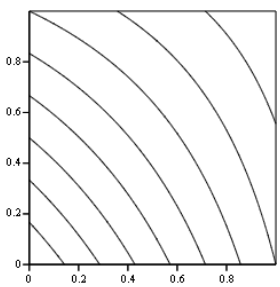
$$y = \underset{i=1}{\overset{n}{S}}[w_i t x_i]$$

OR Neuron- characteristics

$$y = \underset{i=1}{\overset{n}{S}}[w_i t x_i]$$

weights





w=[0.7 0.6]

OR Neuron- boundary cases

$$y = \underset{i=1}{\overset{n}{S}}[w_i t x_i]$$

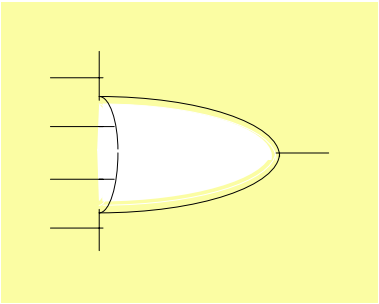
All weights are equal to 1 ➔ $y = OR(x_1, x_2, \ldots, x_n)$

Some weights are equal to 0 ➔ elimination of respective variables

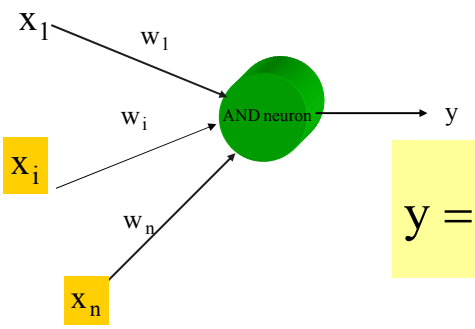
From OR gates to OR neurons

$w_i = 1$

$x_i = 1$ or 0



AND Neuron

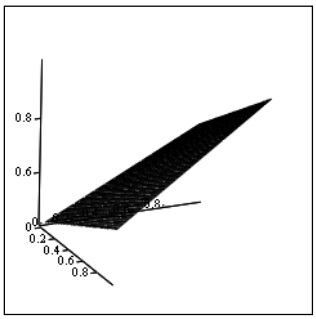


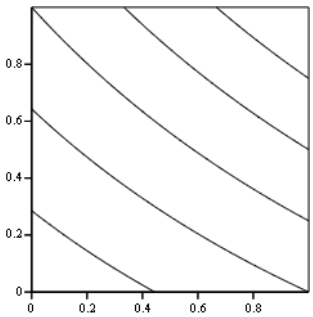
$$y = \underset{i=1}{\overset{n}{T}}[w_i s x_i]$$

AND Neuron- characteristics

$$y = \underset{i=1}{\overset{n}{T}}[w_i s x_i]$$

w=[0.7 0.6]





AND Neuron- boundary cases

$$y = \underset{i=1}{\overset{n}{T}}[w_i s x_i]$$

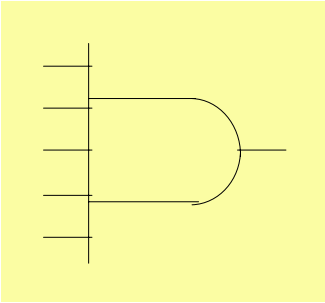
All weights are equal to 0 ➔ $y = AND(x_1, x_2, \ldots, x_n)$

Some weights are equal to 1 ➔ elimination of respective variables

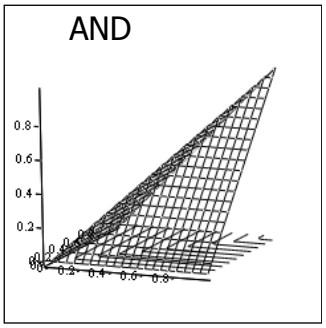
From AND gates to AND neurons

$w_i = 0$

$x_i = 1$ or 0

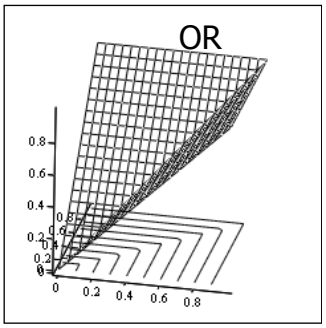


Fuzzy Neurons and Digital Systems



and

w=[0 0]



or, or

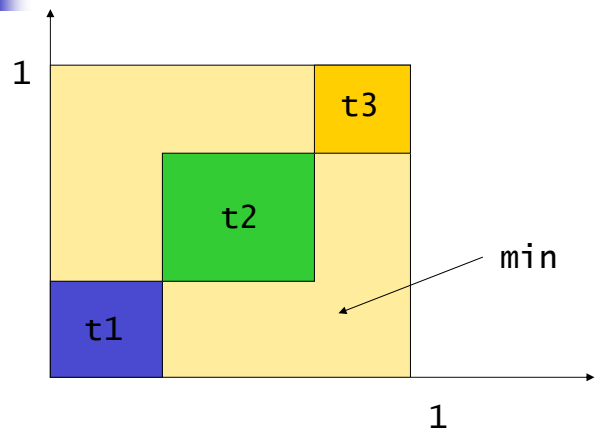
w=[1 1]

Generalizations of fuzzy neurons

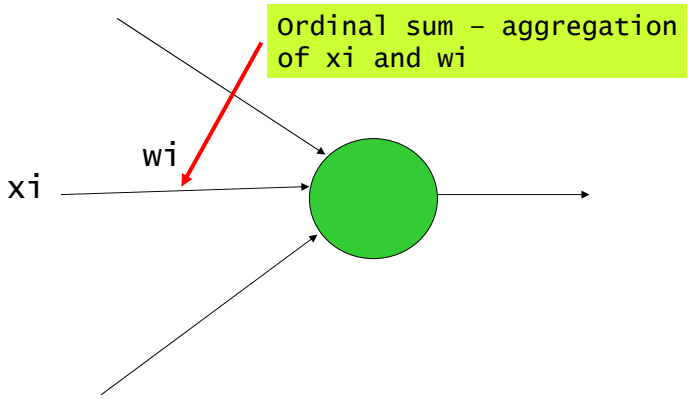
Augmented logic operators

- ordinal sum
- uninorm
- OR/AND neurons

Ordinal sum



Generalized OR neuron



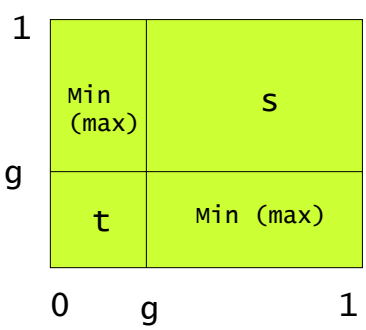
Uninorm

U: $[0,1]^2 \rightarrow [0,1]$

(a)Commutativity
(b)Monotonicity
(c)Associativity
(d)Identity g in[0,1] such that u(x,g)=x

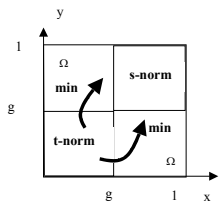
Uninorm

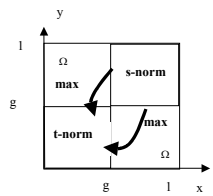
U: $[0,1]^2 \rightarrow [0,1]$



$$u(x,y)=\begin{cases} gt(\frac{x}{g},\frac{y}{g}), \text{ if } x,y\le g \\ g+(1-g)s(\frac{x-g}{1-g},\frac{y-g}{1-g}), \text{if } x,y\in[g,1] \\ \min(x,y) \text{ or } \max(x,y), \text{otherwise} \end{cases}$$

Uninorm: two developments



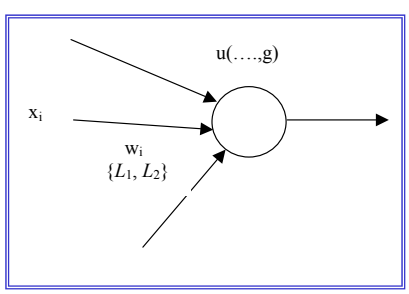


***and*-dominated** ***or*-dominated**

Unineurons

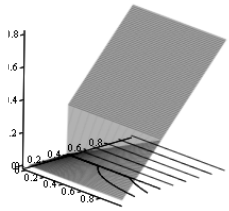
$y = u(or(x_1,w_1),\ or(x_2,w_2),\ldots,or(x_n,w_n),g)$

$y = u(and(x_1,w_1),\ and(x_2,w_2),\ldots,and(x_n,w_n),g)$

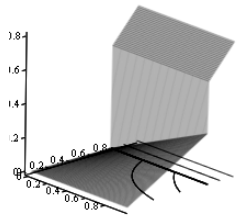


Unineuron [UNI-AND]

$y = u(and(x_1,w_1), and(x_2,w_2),....,and(x_n,w_n),g)$

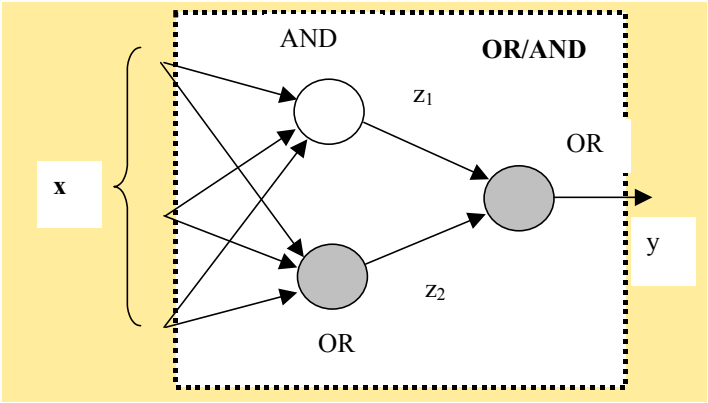


g=0.3



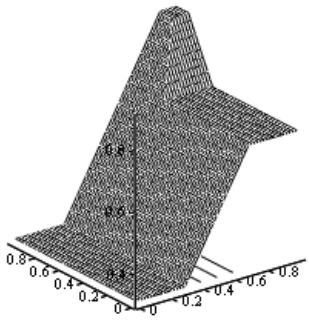
g=0.6

OR/AND neuron



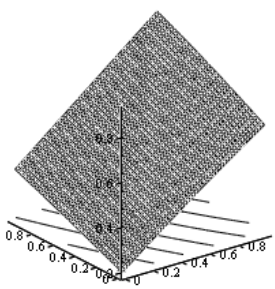
OR/AND neuron-characteristics

t: min, s: max



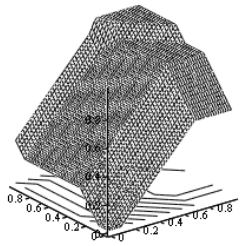
OR/AND neuron-characteristics

t: product, s: probabilistic sum



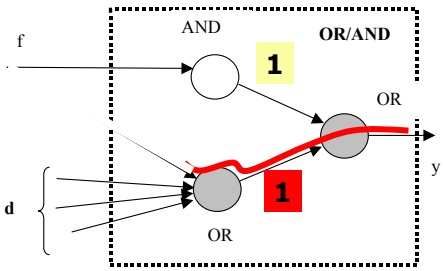
OR/AND neuron-characteristics

t, s: Lukasiewicz connectives



OR/AND neuron: concept generalization and refinement

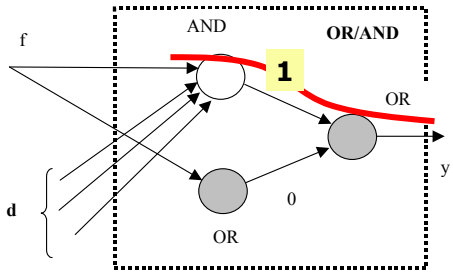
Concept "f" *generalized* by "d"



generalization

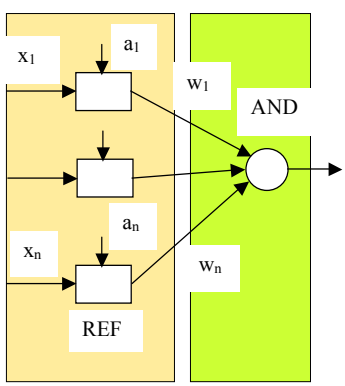
OR/AND neuron: concept generalization and refinement

Concept "f" *specialized* by "d"



Specialization (refinement)

Referential Computing



Referential logic predicates

- "included in"

  x is *included* in a

"dominates"

  x *dominates* a

- "similar to"

  x is *similar* to a

Referential logic predicates: motivation

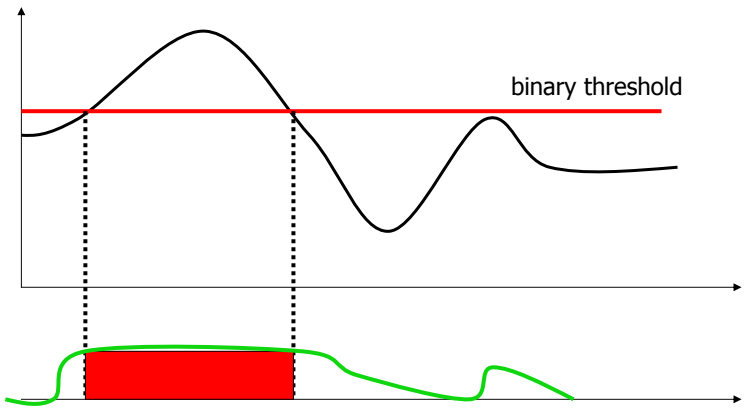
Binary Inclusion → threshold operation

acceptance decision (in classification)
warning signal (sensors)
.....

Continuous predicate of inclusion → fuzzy threshold

partial acceptance
continuous warning signal

Binary and fuzzy thresholding



Inclusion

x included in a, x and a in [0,1]

Implication (→)

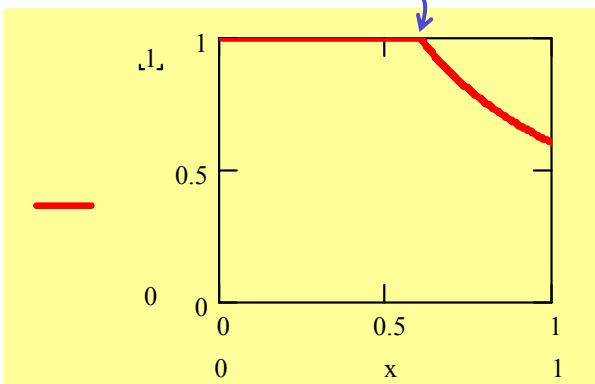
$$a \to b = \sup\{c \in [0,1] \mid atc \le b\},\ a, b \in [0,1]$$

Example (t-norm: product)

$$x \to a = x \subset a = \begin{cases} 1, \text{if } x \le a \\ a/x, \text{if } x > a \end{cases}$$

Inclusion

a=0.6



Inclusion neuron

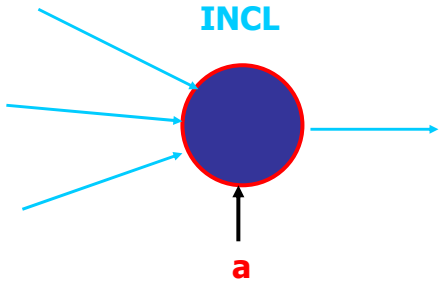
$$y = \mathrm{INCL}(\mathbf{x};\mathbf{a},\mathbf{w}) = \prod_{i=1}^{n}[(x_i \rightarrow a_i)sw_i]$$

w- weights (connections)

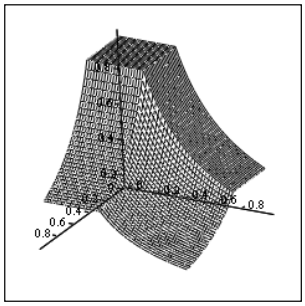
a- reference point

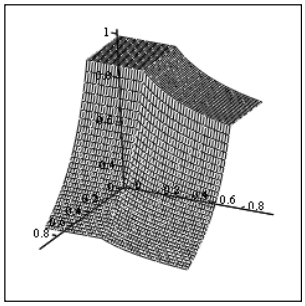
Inclusion neuron

$$p(x;\mathbf{a}) = \prod_{i=1}^{n} x_i^{a_i}$$



Inclusion neuron





w=[0 0]; **a**=[0.4 0.4]

w=[0 0.7]; **a**=[0.4 0.4]

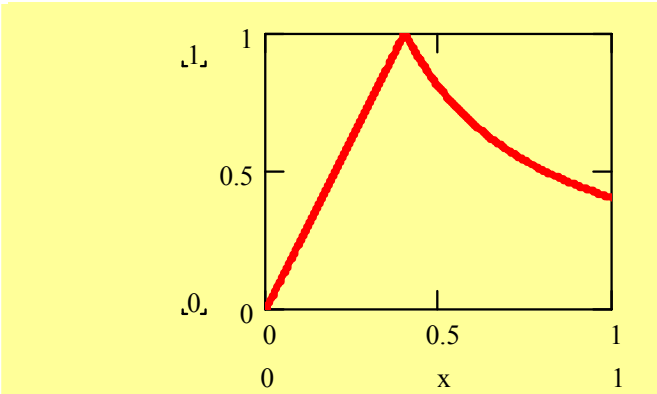
Similarity

x similar to a, x and a in [0,1]

Implication (→)

$$x \equiv a = (x \rightarrow a)t(a \rightarrow x)$$

Similarity

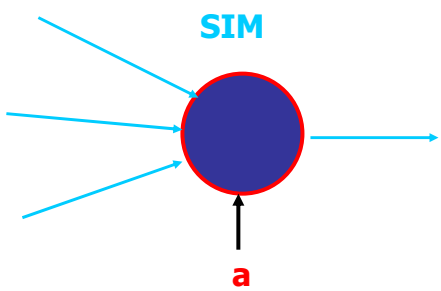


Similarity neuron

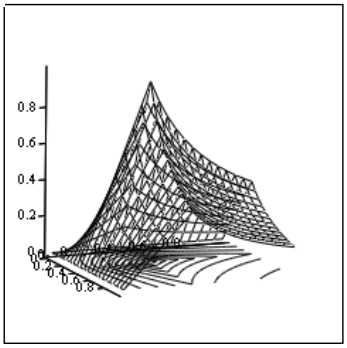
$$y = \mathrm{SIM}(\mathbf{x};\mathbf{a},\mathbf{w}) = \prod_{i=1}^{n}[(x_i \equiv a_i)sw_i]$$

Similarity neuron

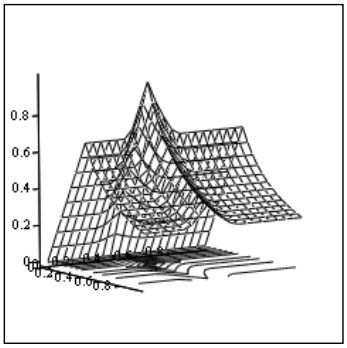
$$y = \mathrm{SIM}(\mathbf{x};\mathbf{a},\mathbf{w}) = \prod_{i=1}^{n}[(x_i \equiv a_i)sw_i]$$



Similarity neuron



w=[0 0]; **a**=[0.4 0.4]



w=[0 0.7]; **a**=[0.4 0.4]

Tolerance neuron

Tolerate changes of signal if those are within some range

TOL(x1,x2,a,b,c,d,w1,w2)=
[tol(x1,a,b) s w1] t [tol(x2,c,d) s w2]

s: s-norm, t: t-norm

Shannon expansion

Boolean function f: {0,1} → {0,1}

$$y = f(x) = (\overline{x} \text{ AND } f(0)) \text{ OR } (x \text{ AND } f(1))$$

f: $\{0,1\}^2 \rightarrow \{0,1\}$

$$\begin{aligned} y = f(x_1, x_2) = & (\overline{x}_1 \text{ AND } \overline{x}_2 \text{ AND } f(0,0)) \\ & \text{OR } (\overline{x}_1 \text{ AND } x_2 \text{ AND } f(0,1)) \\ & \text{OR } (x_1 \text{ AND } \overline{x}_2 \text{ AND } f(1,0)) \\ & \text{OR } (x_1 \text{ AND } x_2 \text{ AND } f(1,1)) \end{aligned}$$

Shannon expansion

Boolean function f: $\{0,1\}^n \rightarrow \{0,1\}$

$$y = f(x_1, x_2, \ldots, x_n) = \text{SUM_OF_MINTERMS}$$

minterms

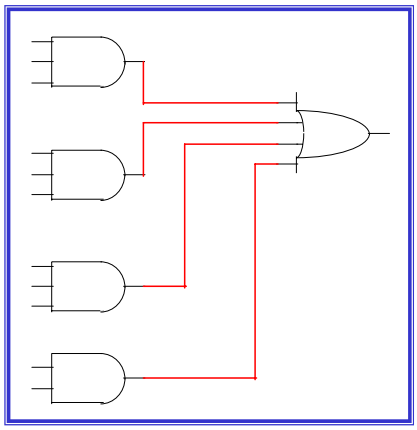
$M_0 = \overline{x}_1 \text{ AND } \overline{x}_2 \text{ AND} \ldots \text{AND } \overline{x}_n$

....

$M_{2^n-1} = x_1 \text{ AND } x_2 \text{ AND} \ldots \text{AND } x_n$

Realization of Boolean functions (combinational systems)

$x_i = 1$ or 0



Shannon expansion - fuzzy function

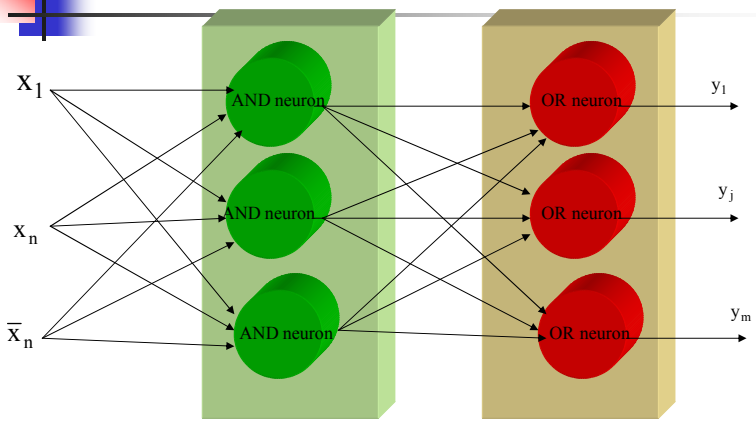
f: $[0,1]^n \rightarrow [0,1]$

$$y \approx (M_0 \ t \ w_0) \, s \, (M_1 \ t \ w_1) \, s \ldots (M_{2^n-1} \ t \ w_{2^n-1})$$

AND neurons

Canonical representation

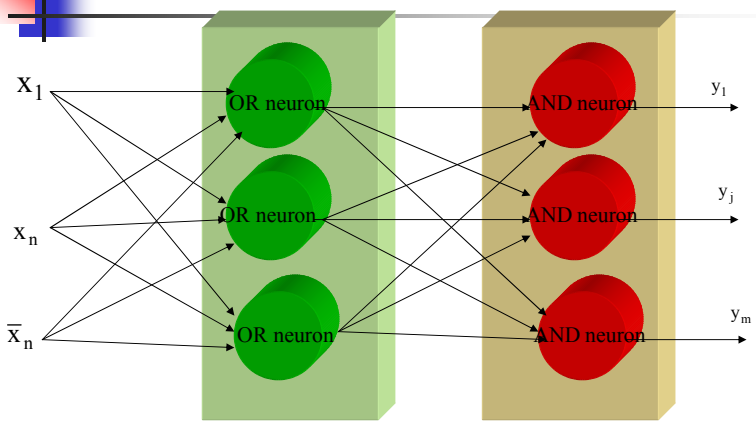
Logic Processor



generalized minterms

logic sum

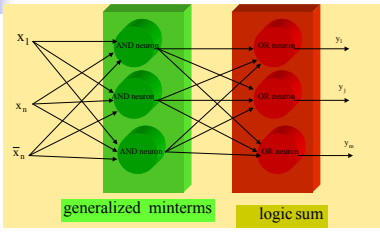
Logic Processor-dual structure



generalized maxterms

logic and - aggregation

Logic Processor- learning



Typically: *gradient-based* learning

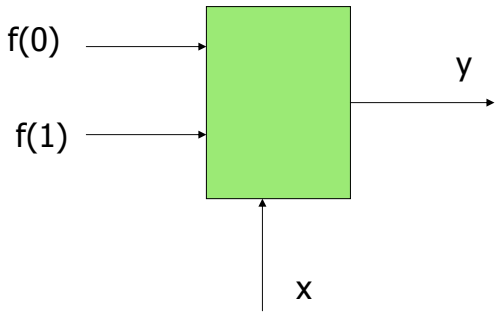
update of the connections of the neurons
backward propagation of error

Structural optimization:

Number of nodes (AND neurons) in hidden layer
Specific model of t- and s-norm

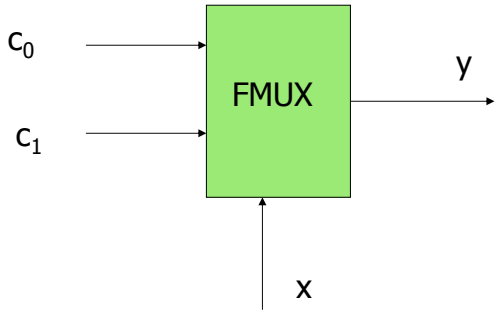
Shannon expansion – multiplexer realization

$$y = f(x) = (\overline{x} \text{ AND } f(0)) \text{ OR } (x \text{ AND } f(1))$$



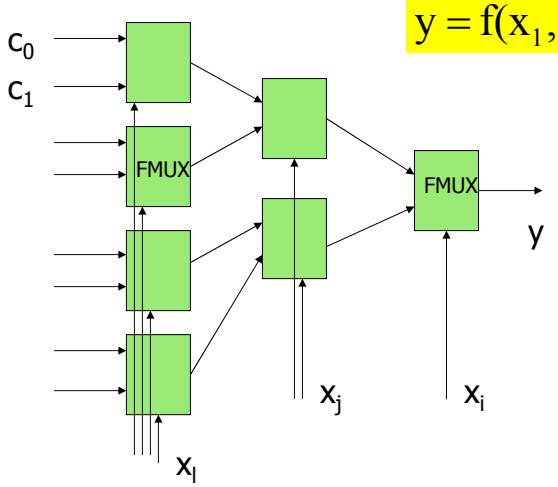
Fuzzy multiplexer

$$y = (\bar{x}\ t\ c_0)\ s\ (x\ t\ c_1)$$



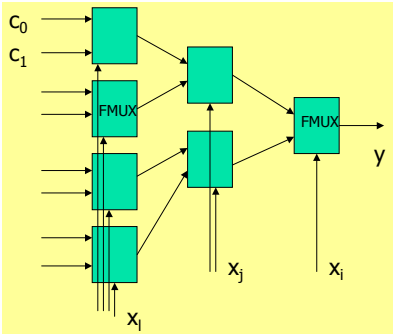
Network of fuzzy multiplexers

$$y = f(x_1, x_2, \ldots, x_n)$$



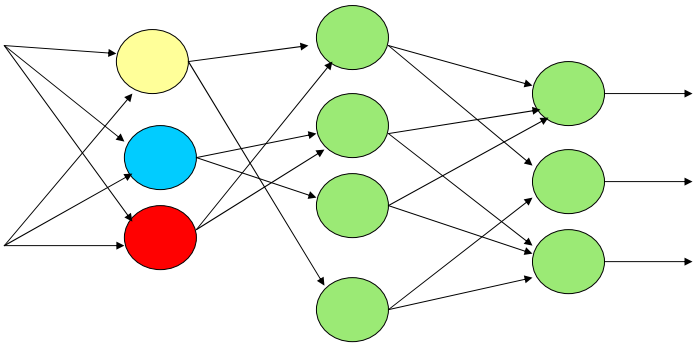
Network of fuzzy multiplexers: design

Optimization of constants (c_0, c_1, ...)



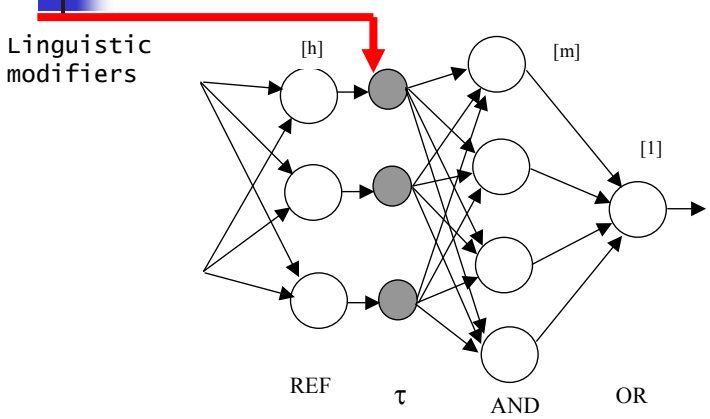
Selection of variables-dimensionality reduction

Heterogeneous logic network



Referential neurons **AND neurons** **OR neurons**

Heterogeneous logic network

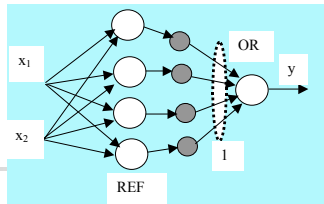


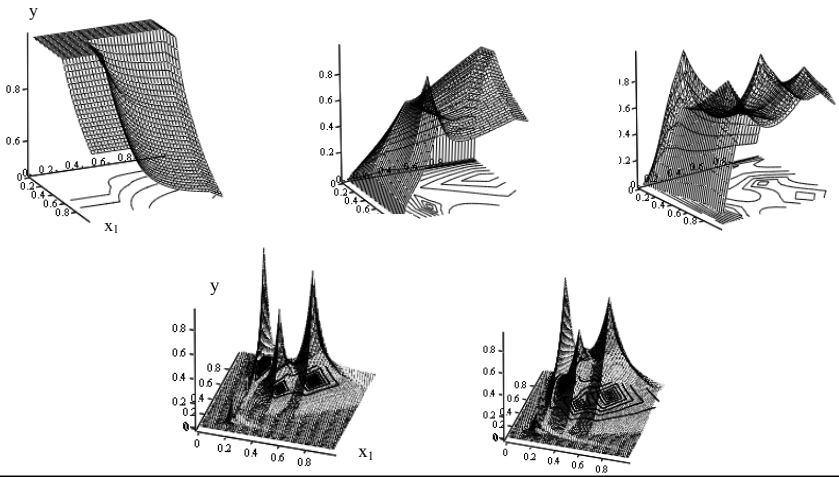
Referential neurons **AND neurons** **OR neurons**

Linguistic modifiers

½ - *more or less* (dilution effect)
0 - unknown
1 – true (neutral)
2 - *very* (concentration effect)
4 - *very* (*very*) = *very* ^{2} (strong concentration effect)

Examples





Combinational & Sequential Fuzzy Systems

Static **dependencies → fuzzy neurons→**

→Fuzzy logic logic network

Dynamic **dependencies → fuzzy neurons and fuzzy flip-flops→**

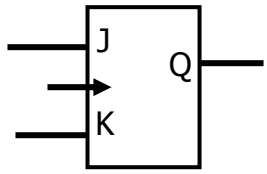
→Fuzzy sequential logic logic network

- **FUZZY FLIP-FLOPS**
- **DESIGN**

Fuzzy JK flip-flop

Next state equation

$$Q^{+} = J\overline{Q} + \overline{K}Q$$



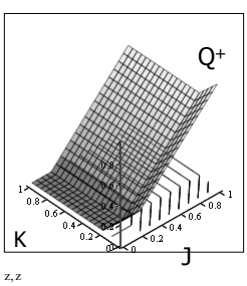
Generalization:

inputs (J,K) and state (Q,Q^{+}) in [0,1]
use of t- and s-norms

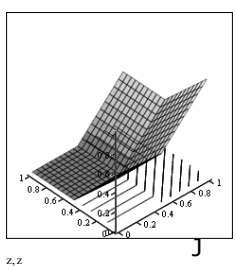
$$Q^{+} = (J t \overline{Q}) s (\overline{K} t Q)$$

Fuzzy JK flip-flop: characteristics

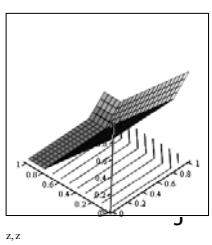
t-norm, s-norm: Lukasiewicz connectives



Q=0.2

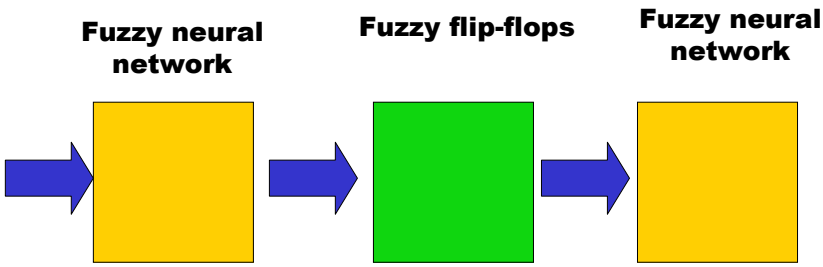


Q=0.5



Q=0.8

Fuzzy sequential model



Fuzzy State Machines

Concluding Comments

Fuzzy logic networks as structures of knowledge representation and learning

Transparency of networks, diversity of logic operations, and their plasticity

Learning involving mechanisms of neurocomputing and evolutionary optimization