NATURAL COMPUTING

……. INTELIGENTA COMPUTATIONALA .. INTEGRATA

Arhitecturi + algoritmi – mimeaza inteligenta naturala

Aplicatii:

-Recunoastere biometrica

-Recunoastere voce

-Asistenti pentru persoane cu dizabilitati

-Recunoastere de imagini

- Dispozitive introducere date (semnaturi, scris etc.)

- “Smart sensors”

- Robotica …

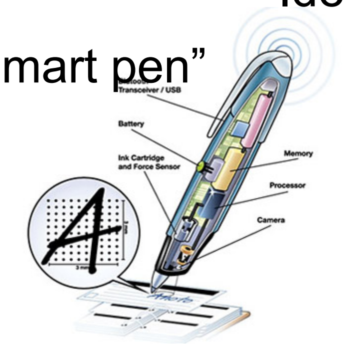
- Idei noi ??

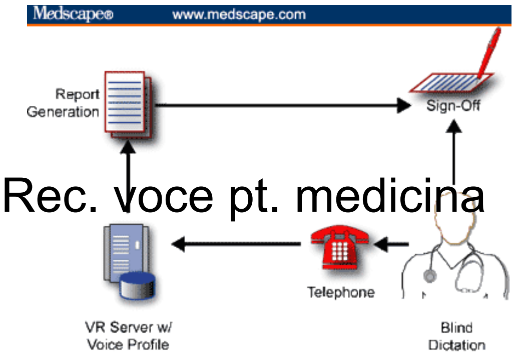
Algoritmi si arhitecturi cu **complexitate redusa** si implicit usor integrabili in tehnologii (VLSI, FPGA, microcontrollere)

“LOW POWER”

vOICe for Android

“smart pen”





Rec. voce pt. medicina

Intelegere imagini -> conversie mesaje sonore

TRANSFER DE CUNOSTINTE

Inteligența este facultatea de a descoperi proprietățile obiectelor și fenomenelor înconjurătoare, cât și a relațiilor dintre acestea, dublată de posibilitatea de a rezolva probleme noi.

Inteligența unui sistem nu este definită de modul în care este el alcătuit, ci prin modul în care se comportă.

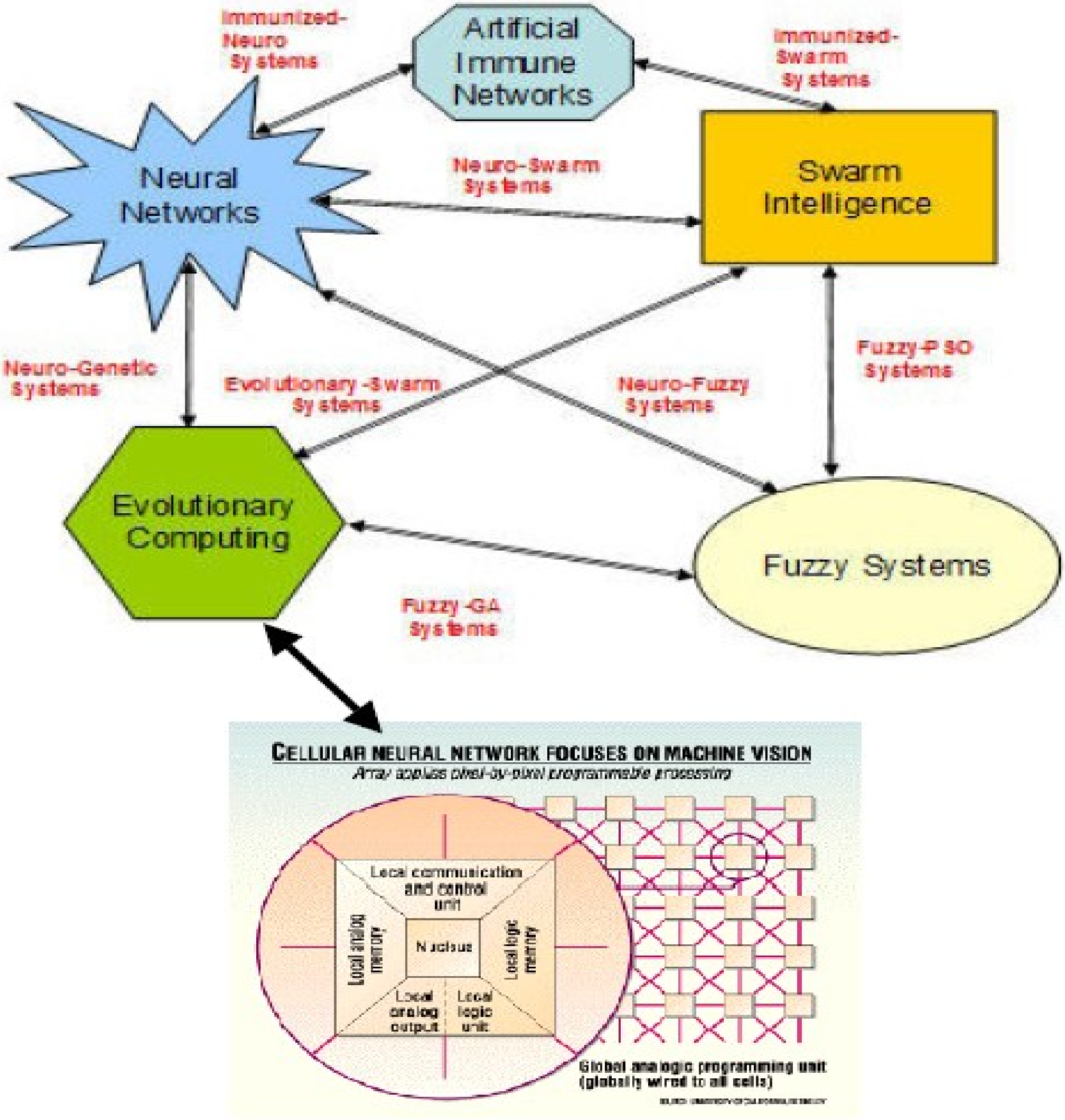
STRUCTURA (ARHITECTURA)

MEDIUL DE STIMULARE

CAPACIATATE DE GENERALIZARE

Putem avea structuri supradimensionate (over-fitting) sau sub-dimensionate (under-fitting; acestea nu pot suporta o “inteligenta” prea evoluata

Necesitatea de INTEGRARE presupune optimizarea (reducerea complexitatii, adecvare la tehnologie) STRUCTURII (ARHITECTURII) si a algoritmului de invatare (TRANSFER DE CUNOSTINTE)



DIFERITE SUBDOMENII ALE INTELIEGNTEI COMPUTATIONALE

ARTIFICIAL INTELLIGENCE VS. COMPUTATIONAL INTELLIGENCE

Se bazeaza pe reguli de inferenta (sisteme expert)

Desi a fost o mare speranta in anii ’60 –’70 nu a putut rezolva multe probleme practice

Se bazeaza pe codificarea si prelucrarea numerica (computing) a informatiei.

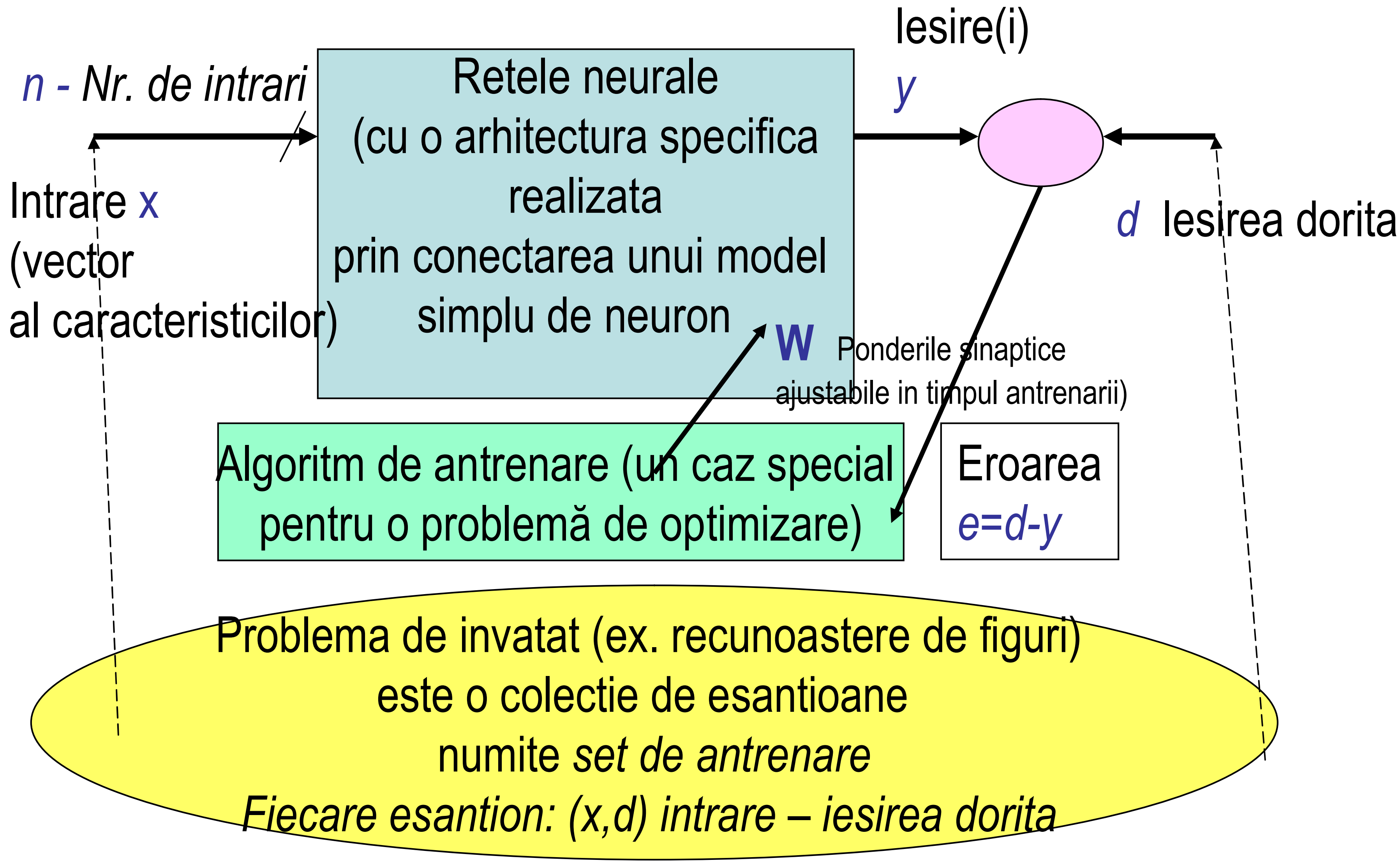
Arhitecturi computationale de inferenta:
-Neurale
- Fuzzy

http://design.open.ac.uk/ecidII/docs/Bitterman.pdf

Approaches based on **Classical AI are inferior compared to approaches based on CI** regarding the treatment of most complexity issues in design. In particular this concerns dealing with vagueness, multi-objectivity and large amount of possible solutions. Therefore, application of classical AI is limited to problems that minimally involve these issues. As design tasks are generally characterized by these issues, application of the classical AI approach for such tasks is questionable in general.

Neuro

Sisteme “Feed-forward” (fara bucla de reactie)
Folosit în cele mai multe cazuri practice



Fuzzy

Sisteme "Feed-forward" cu logica fuzzy

n Nr. de intrari

Iesirea y

Intrare x
(vector
al caracteristicilor)

Sisteme cu logica fuzzy

Arhitectura este definită astfel incat se poate cuantifica folosind reguli"fuzzy"
Sisteme expert ca:
IF temperatura (x) este mare
THEN controlul caldurii (y) este mic

G parametri dati de expertul uman

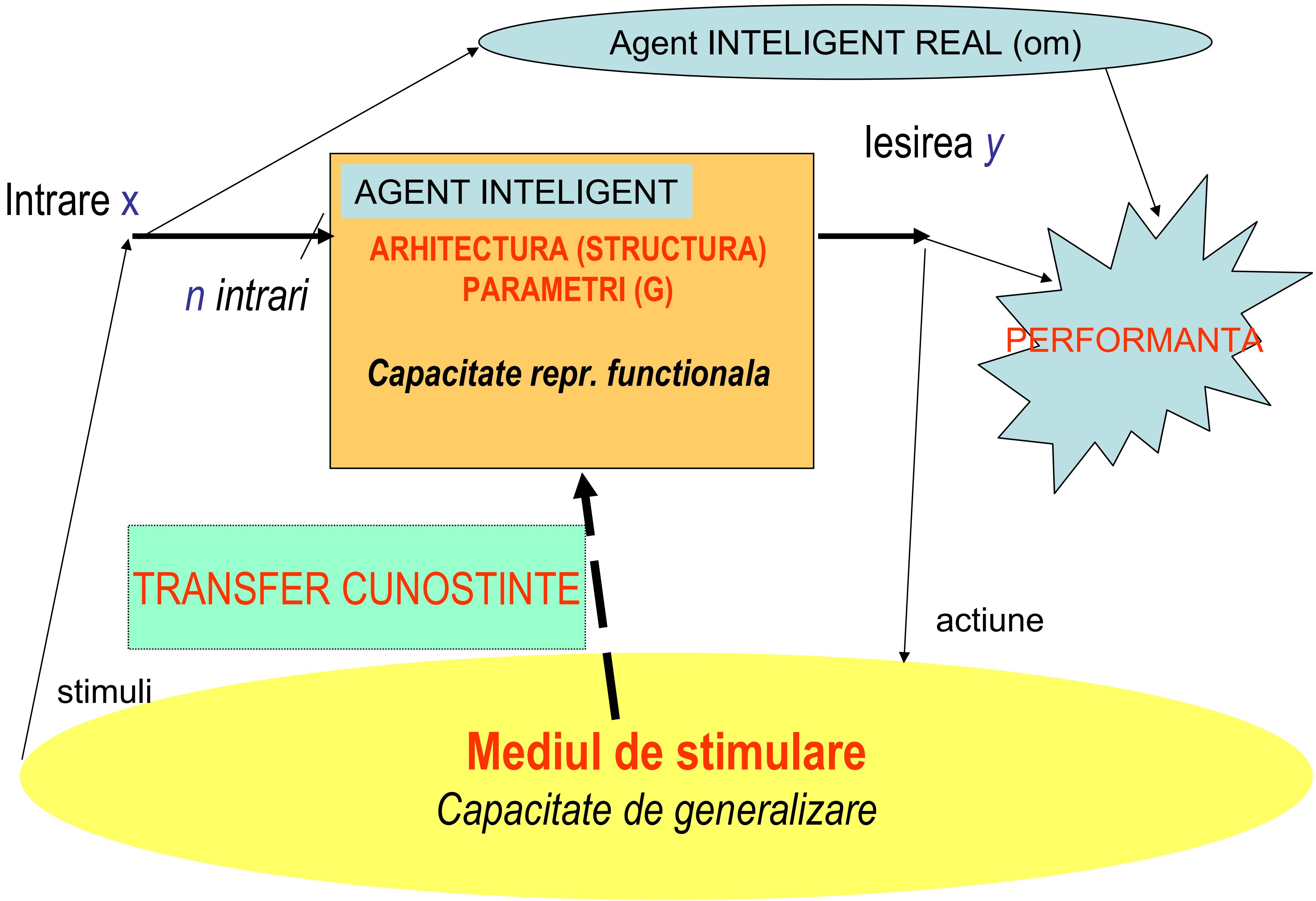
Algoritmul de antrenare nu mai este necesar

Dar există şi sistemele neuro-fuzzy unde antrenarea se face pentru un reglaj fin

Problema de invatat (ex. controlarea temperaturii) este o colecţie de reguli, exprimate ca mai sus

Nu este nevoie de antrenarea datelor – avantaj important

Dar functioneaza numai pe baza cunostintelor umane care pot fi exprimate ca reguli


Agent INTELIGENT REAL (om)
Iesirea y
Intrare x
AGENT INTELIGENT
ARHITECTURA (STRUCTURA)
PARAMETRI (G)
n intrari
Capacitate repr. functionala
PERFORMANTA
TRANSFER CUNOSTINTE
actiune
stimuli
Mediul de stimulare
Capacitate de generalizare

Studiu de caz – optimizarea pentru integrare a arhitecturii

a) Neuron cu sinapse multiplicative:

$$syn(x,w) = xw\,; \quad v = w_0 + \sum_{i=1}^{n} syn(x_i, w_i)\,; \;\; y = sign(v)$$

b) Neuron cu sinapse comparative (dupa cum s-a aratat in [1] prin utilizarea unor astfel de neuroni in sisteme neuro-fuzzy) performantele sistemului raman neschimbate, in schimb complexitatate de implementare este mult redusa.

$$syn(x,w) = \tfrac{1}{2}\left(|x+w| - |x-w|\right); \;\; v = w_0 + \sum_{i=1}^{n} syn(x_i, w_i)\,; \quad y = sign(v),$$

Pentru implementarea efectiva a sinapsei se va utiliza formula echialenta [1] care permite calulul separat al bitilor de marime si respectiv al bitului de semn:

$$abs(syn(w,x)) = \min(|x|,|w|)$$

$$sign(syn(w,x)) = sign(x)\,sign(w)$$

	n=6	n=16	
Multiplicativa	9 Slices	64 slices	$O(n) = (n/2)^2$
Comparartiva	6 Slices	16 slices	$O(n) = n$

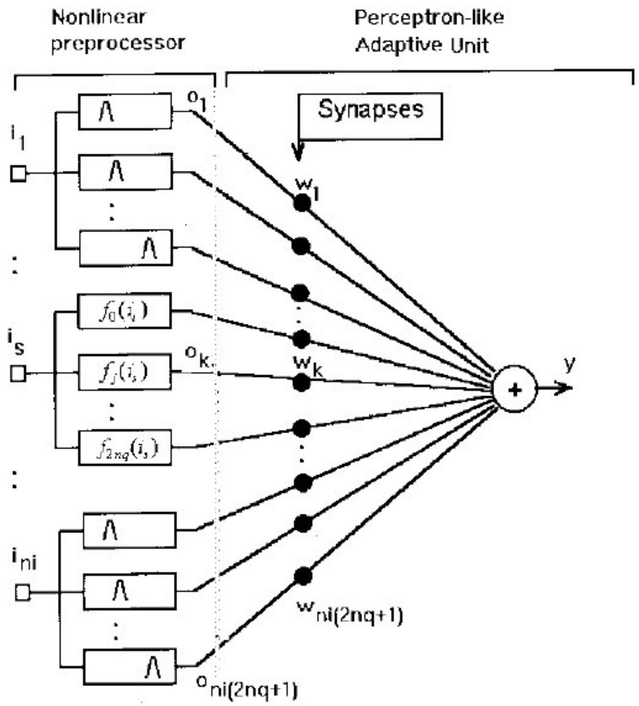


Fig. 1. The fuzzy perceptron. A nonlinear preprocessor is followed by an adaptive perceptron-like structure.

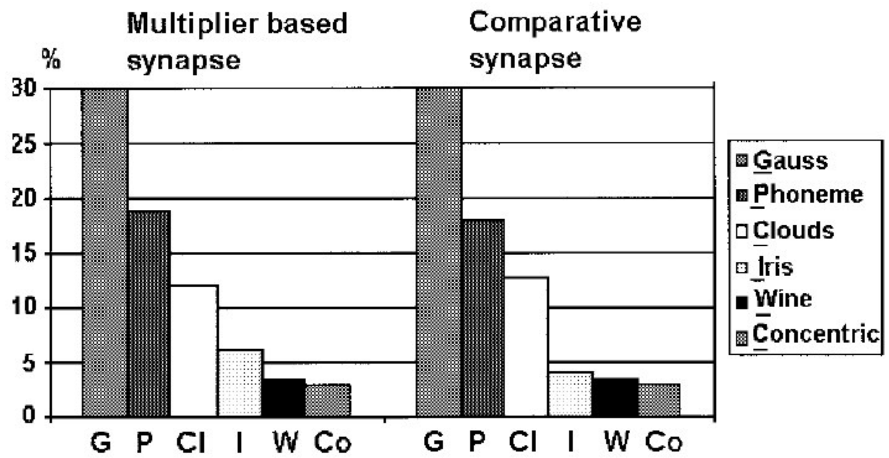


Fig. 6. A comparison of the classification performance for both synaptic models. The label below each bar indicates the classification problem. The bar height indicates the best misclassification error obtained in each case for its associated optimal neuro-fuzzy structure.

[1] Radu Dogaru and Leon O. Chua, „The Comparative Synapse: A Multiplication Free Approach to Neuro-Fuzzy Classifiers“, in IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: FUNDAMENTAL THEORY AND APPLICATIONS, VOL. 46, NO. 11, NOVEMBER 1999, pp. 1366-1371.

ALTA ABORDARE SIMILARA

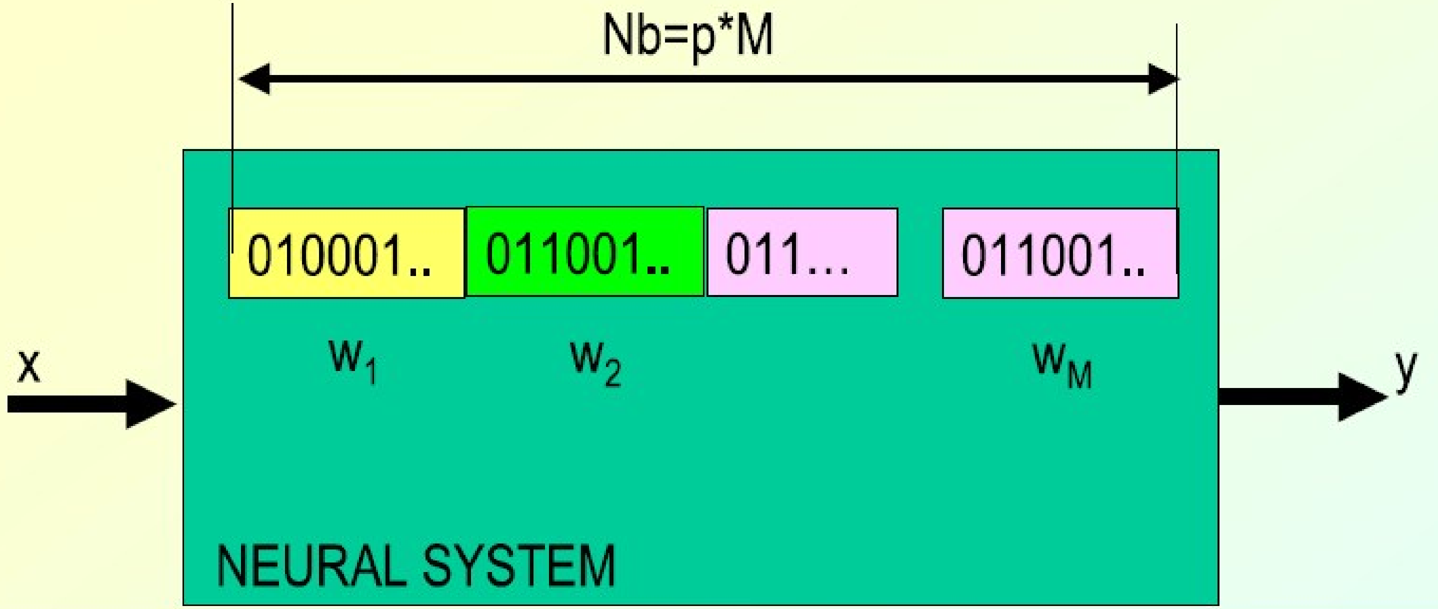
Utilizarea unei precizii finite pentru reprezentarea parametrilor

An efficient finite precision RBF-M neural network architecture using support vectors

Dogaru, R.; Dogaru, I.;
Dept. of Appl. Electron. & Inf. Eng., Univ. Politeh. of Bucharest, Bucharest, Romania

This paper appears in: Neural Network Applications in Electrical Engineering (NEUREL), 2010 10th Symposium on
Issue Date: 23-25 Sept. 2010
On page(s): 127 - 130
Location: Belgrade

Se pune problema optimizarii numarului efectiv de biti pentru reprezentarea parametrilor



In FLP *M* is optimized for best generalization (if too small, no functional representation capability; if too large – poor generalization)

In FP and FPBW a finer optimization is possible by properly choosing the representation size *p* → FP and FPBW in some cases improve the classification performance while making the realization more efficient.

r - parametru structural unic (poate fi optimizat relativ usor)

Performance of FPBW implementation of RBF-M (Problem: Iris)

Precision p (bits)	2	3	4	6	7	10
m	8	18	16	15	15	15
PIC (%)	8	2	0	0	0	0
r	0.32	0.27	0.31	0.31	0.31	0.31

A finite precision representation is not necessary bad for the overall performance, by carefully tuning it can achieve a performance that is unmatched by any other floating point model (for instance, SVM achieves best PIC=4% for this problem).

An explanation stands in the Occam's razor: A floating point representation may lead to an excess in the required bits to store the knowledge and ensure generalization.

Complexity of the best solutions for IRIS problem

FLP case: 5 support vectors with 4 components each;
6 Adaline weights (32 bits FLP) => Nb=26x32=832 bits

Best recognition error = 2%

FP case: 5 support vectors with 4 components each;
6 Adaline weights (4 bits FP) => Nb=26x4=104 bits

Best recognition error = 0%

FPBW case: 16 support vectors with 4 components each (4 bits FP);
17 Adaline weights (1 bit/synapse) => Nb=64x4+17=273 bits

Best recognition error = 0%

No multiplier is needed, this architecture may have the most efficient FPGA implementation

Application – handwritten character recognition

Percentage of misclassification

	FLP	FP (p=6 bits)	FPBW (p=5 bits)	SVM
PIC (%)	4.3	4.3	5.37	4.3
m (units)	31	32	42	-
r	0.515	0.515	0.42	-

As in the previous cases, after a proper tuning of the FP precision it is possible to get the best performance for this 10-class problem (in this case 4.3%)

A wider palette of problems

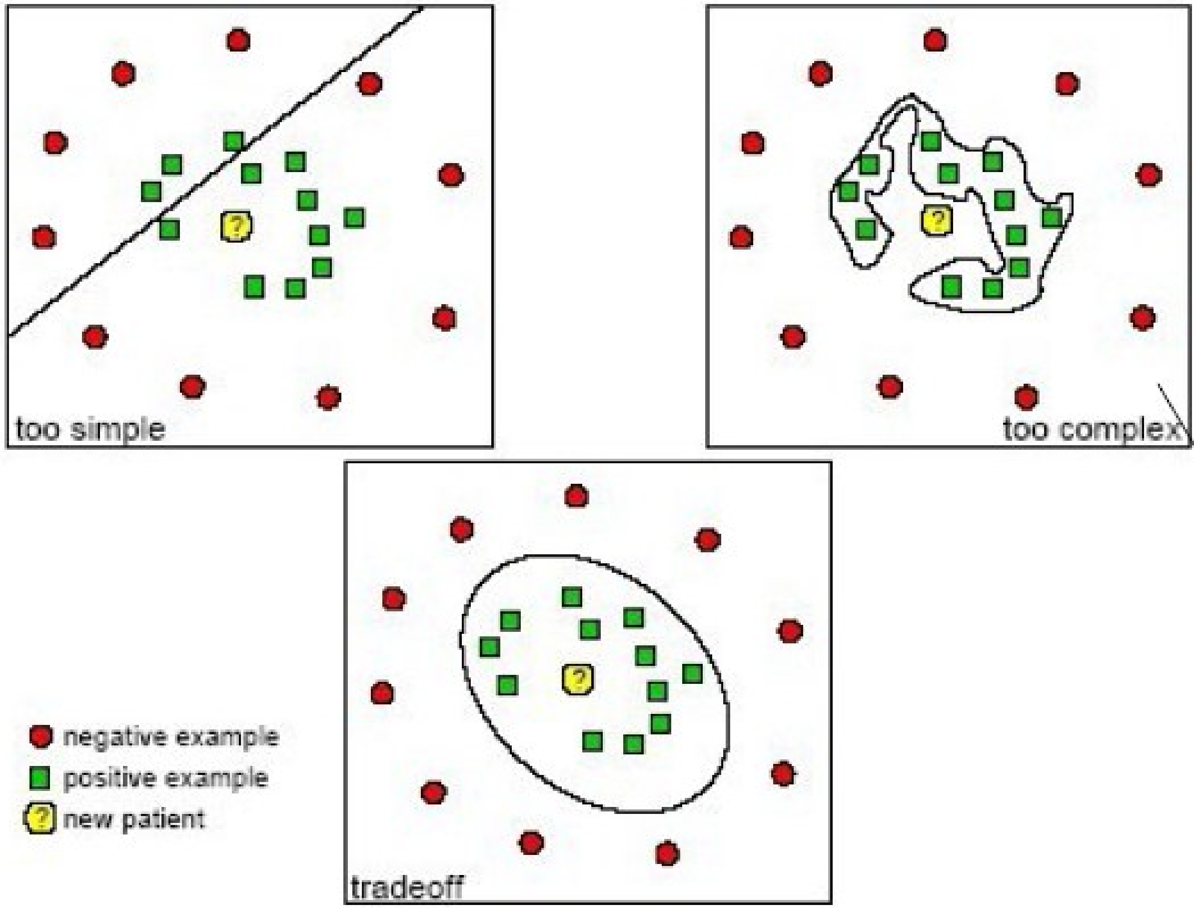
Note than a properly tuned fixed point VLSI friendly RBF-M outperforms SVM.

For certain problems fixed point Adaline is a better choice.

Problem	RBF-M	ADALINE	SVM (FLP)
IRIS [7]	PIC=0% FP *p=4* m = 5		PIC=4%
WINE [7]	PIC=0% FP *p=5* m = 49	PIC=0% FP *p=3* (m=0)	PIC=1.13%
HEART	PIC=19.1% FP *p=8* m=18	PIC=19.35% FP *p=12* (m=0)	PIC=19.1%
WAVE (version 1) [7]	PIC=18 % FP *p=5* m = 17	PCIC=15% FP *p=12* (m=0)	PIC=15%
PHONEME [8]	PIC=21% FPBW *p=4* m = 64		PIC=22.5%
DIABETE (Pima Indians) [7]	PIC=23.4% FPBW *p=7* m = 19		PIC=24.5%

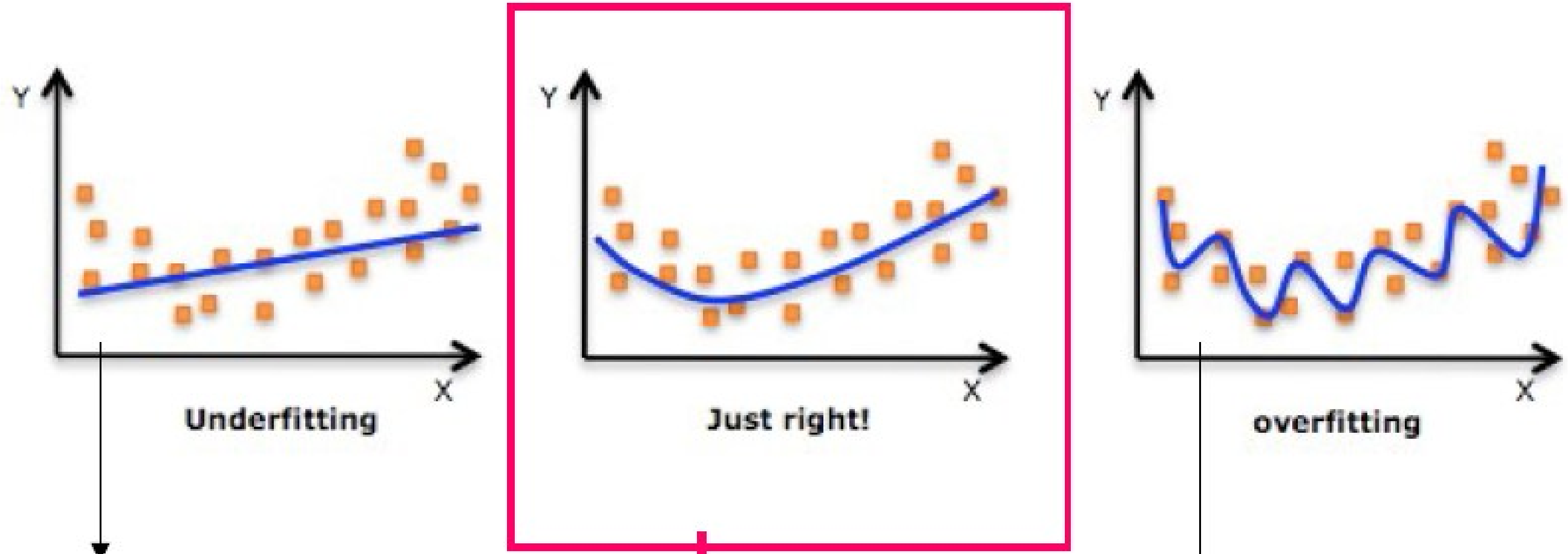
CU PRIVIRE LA PARAMETRI

Underfitting and Overfitting



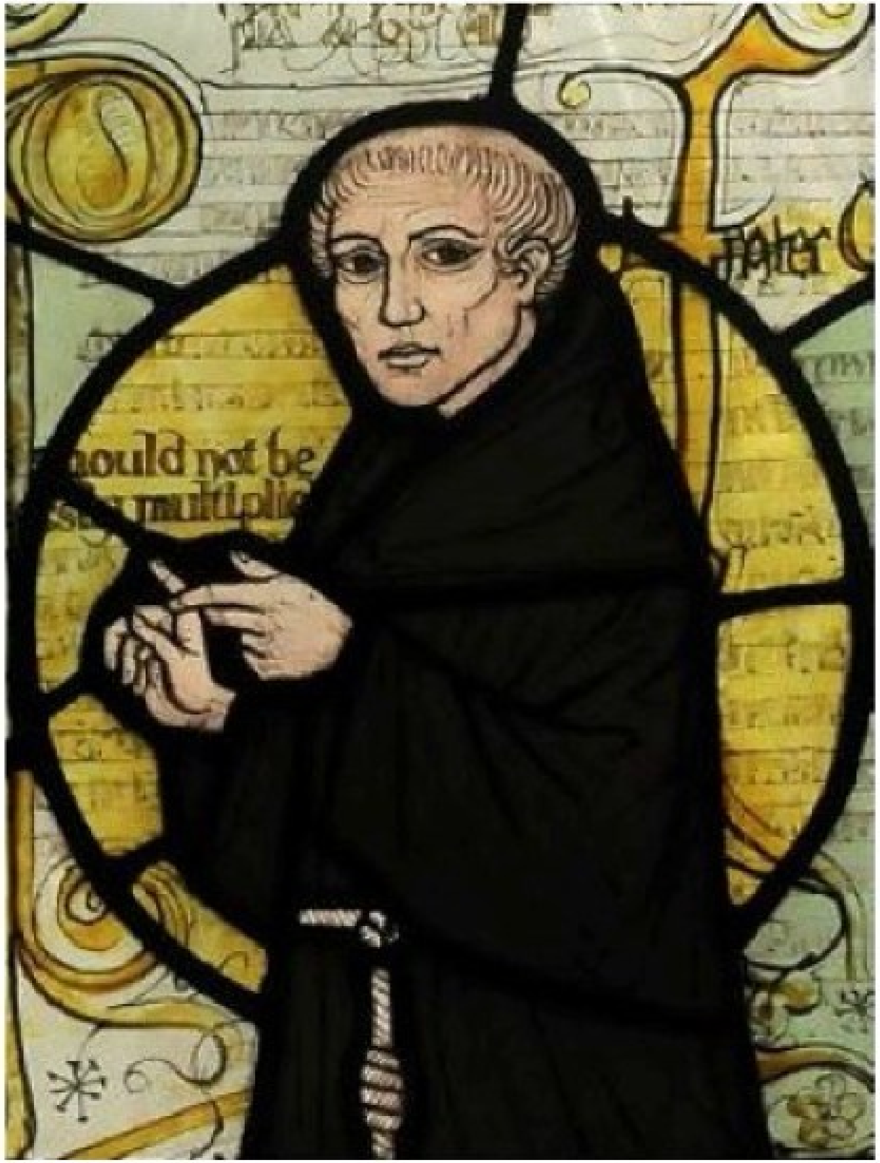
Prea putini parametri

Prea multi parametri



Capacitate de reprezentare
functionala redusa !

Memorizare excesiva
(overfitting) – generalizare slaba



LEGEA PARSIMONIEI
- Aplicabila si in alte privinte

Occam's Razor

often expressed in Latin as the *lex parsimoniae*, translating to **law of parsimony**, **law of economy** or **law of succinctness**,

William of Ockham

1288 – 1348