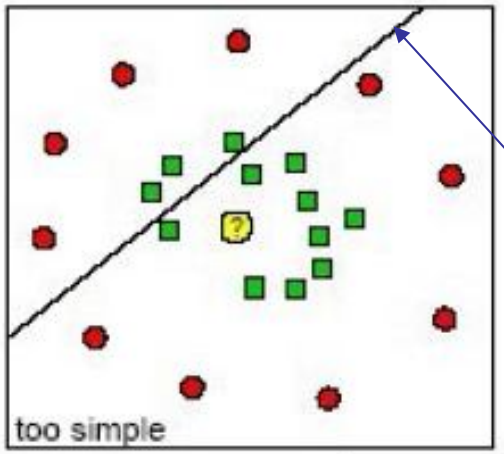
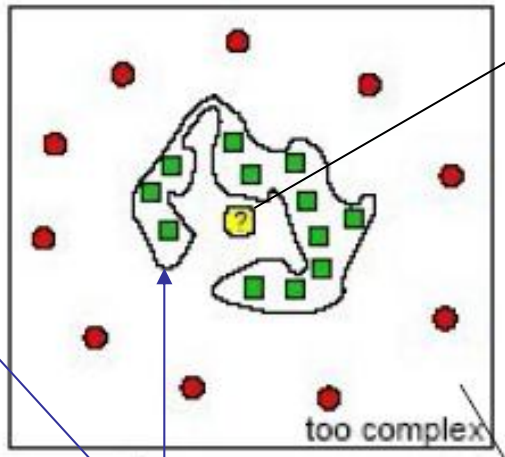
CURS 2

Alegerea nr. de parametri si a structurii unui sistem inteligent:

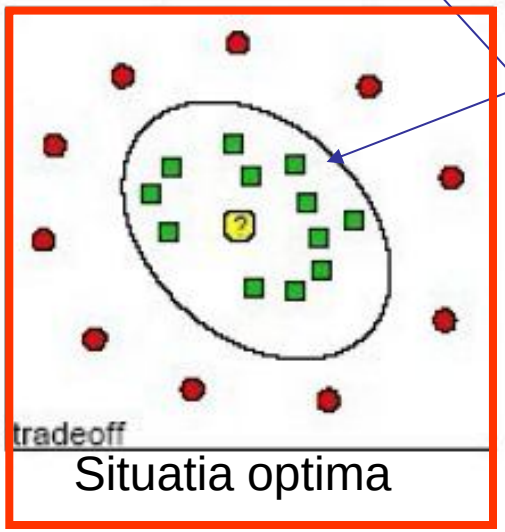
- Overfitting (supra-dimensionare) si underfitting (sub-dimensionare),
- Capacitate de reprezentare functionala (CRF).
  Universalitate, aproximatori functionali universali;
- Probleme liniar si neliniar separabile;
- Metode de extindere a CRF inclusiv din perspectiva unei complexitati cat mai reduse
-Principiul lui Occam;
Exemplificare: neuron "m-nest“
- Memorizare vs. generalizare, rolul datelor de antrenament

Underfitting and Overfitting





Un nou vector de intrare (nu a fost prezent la antrenare) este clasificat gresit



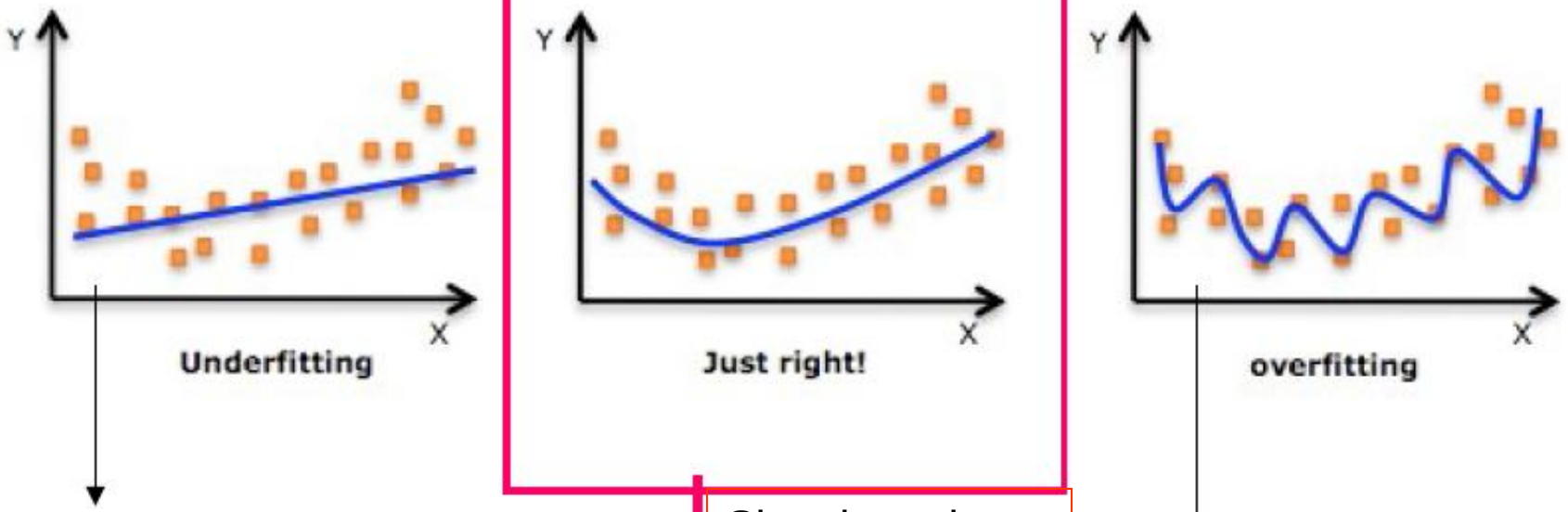
Situatia optima

negative example
positive example
new patient

Frontierele sunt date de functia “sablon” y=f(x,w) asociata sistemului inteligent;
Parametrii w se ajusteaza (antrenare) astfel incat frontiera sa se “muleze” pe date.
Functia “sablon” = structura sistemului inteligent

Prea putini parametri

Prea multi parametri



Capacitate de reprezentare functionala redusa !

Situatia optima
Numar optim de parametri si structura (functia sablon) optima

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

"Make everything as simple as possible, but no simpler"

Einstein

MODEL SIMPLIFICAT DE NEURON (NEURON LINIAR STANDARD)

Timpul este ignorat
in modele simple de neuron

$f(x) = \mathrm{sgn}(x)$

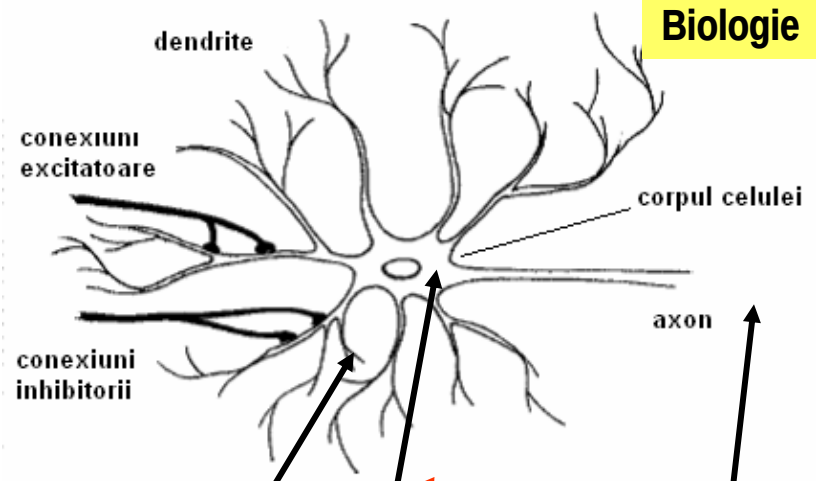
1943, McCulloch &Pitts

↓

Model continuu

$f(x) = \tanh(\lambda x)$

Biologie



Informatia este stocata
in ponderile sinaptice
care urmeaza sa fie
optimizate in ceea ce
priveste functia obiectiv

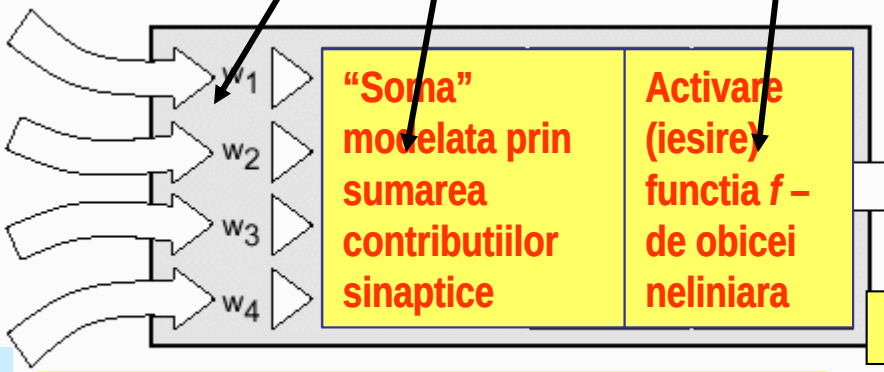
$$y = f\left(\sum_{i=1}^{n} w_i x_i\right)$$

Vector de intrare (stimul)

x_1

x_i

x_4



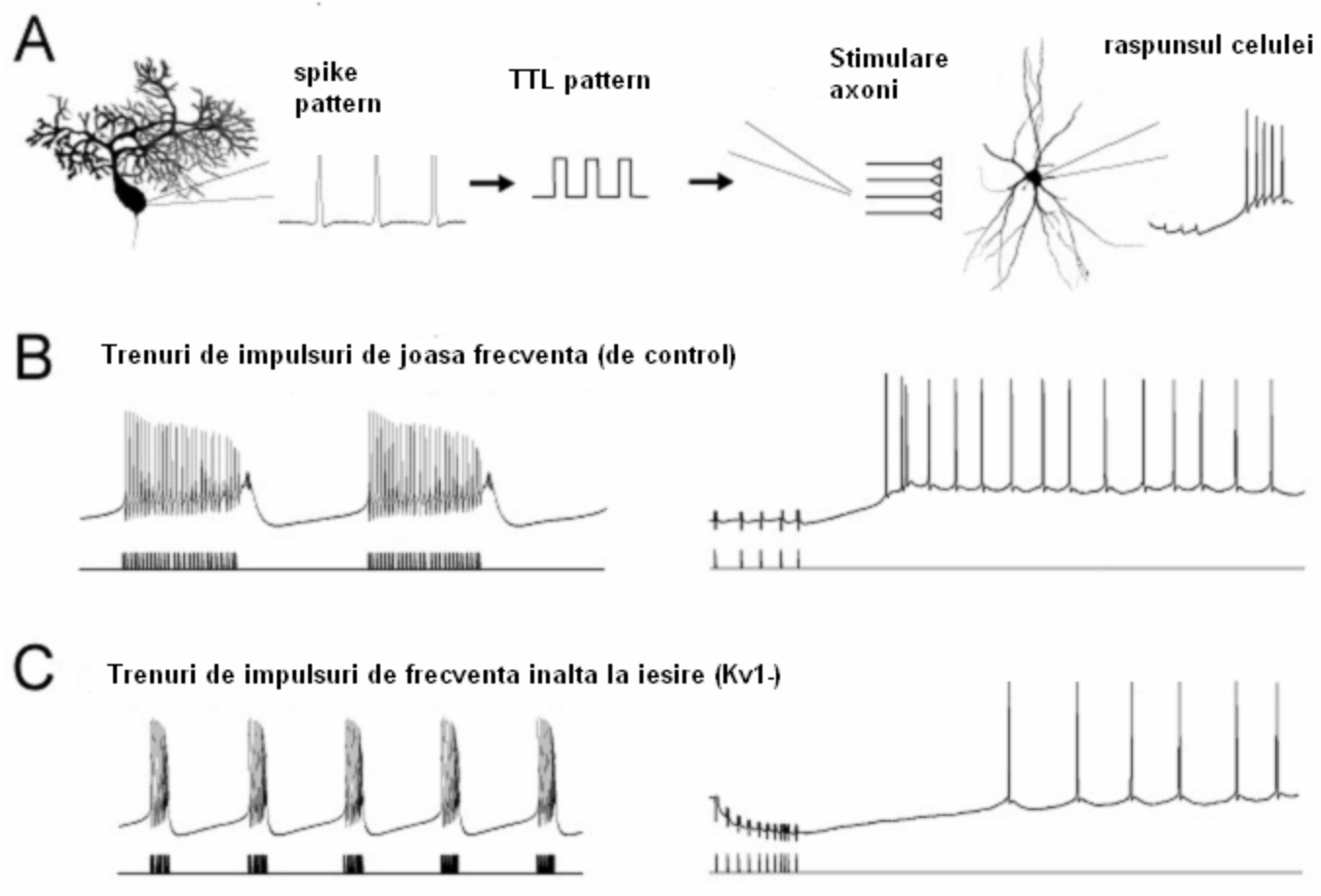
Iesire (binara/continua)

Modelul neuronului artificial

Intrari (**vectorul caracteristicilor, sau stimului**)

Modul de lucru al neuronilor reali (biologici)

Frecventa spike-urilor proportionala cu nivelul de excitatie in “corpul celulei”



Aceasta relatie neliniara este modelata prin utilizarea functiei *f*

Corpul celulei este de obicei modelat prin insumarea functiilor (model liniar)

Capacitate de Reprezentare Functionala (CRF),

Aproximatori Functionali Universali (AFU)

Cum se poate imbunatati CRF→ AFU

Exercitiu: Antrenarea unui neuron cu 2-intrari pentru invatarea functiei AND

Un singur neuron poate invata mai multe functii Boole? Pe toate?

Reprezentarea grafica a spatiului de intrare poate fi foarte utila:
-Frontiera asociata neuronului liniar este o dreapta si pozitia ei este stabilita prin modificarea ponderilor.
- Setul de antrenare (ex. tabela de adevar a functiei doritepoate fi reprezentat prin puncte marcate (marcajul corespunde etichetei d asociate unui vector stimul x)
- Antrenarea este procesul de gasire a pozitiei dreptei astfel incat de o parte a ei sa avem puncte dintr-o singura categorie (eticheta)

Neuronul liniar standard (perceptron liniar) poate doar invata numai problemele liniar separabile.
Functia AND este una dintre acestea. XOR nu este.

Liniar Separabilitate – vs. Nonliniar Separabilitate

Problema XOR

Intrebare: cum se poate rezolva ?

Experiment: - Implementati un neuron“standard” (McCulloch & Pitts, sau ADALINE) si listati toate functiile Booleane cu *n* intrari. Ponderile trebuie alese aleator pana cand nici o alta noua functie Booleana nu mai este gasita.

Intrari *n*	Functii liniar separabile	Toate functiile posibile
1	4	4
2	14	16
3	104	256
4	1882	65536
5	94572	4.3×10^{9}
6	5028134	1.845×10^{19}
7	?	3.4×10^{38}

Odata cu crestera numarului de intrari, procentul de functii logice liniar separabile scade !

APROXIMATORI FUNCTIONALI UNIVERSALI (AFU)

Neuronul liniar are capacitate de reprezentare functionala (CRF) limitata: prin ajustarea parametrilor poate invata numai o functie logica linear separabila.

Se doreste construirea unor structuri (sisteme inteligente) care sunt universale in sensul ca pot reprezenta orice functie dintr-o anumită categorie (de exemplu: functii Boole cu *n*-intrari). Aceasta înseamnă că o structură predefinită există, astfel încât oricare dintre funcțiile dorite în categorie (de exemplu, orice funcție booleena) poate fi învățata prin ajustarea setului de parametri. 2^{2^n}

O astfel de structura are proprietatea de a fi un aproximator functional universal (AFU). Ea garantează că erorile la antrenare sunt doar rezultatul definirii gresite a datelor de antrenare.
Altfel, (sisteme cu CRF redusa) erorile de învățare pot fi mari din cauza capacitatii functionale limitate indiferent de cat de bine s-au selectat datele de antrenament

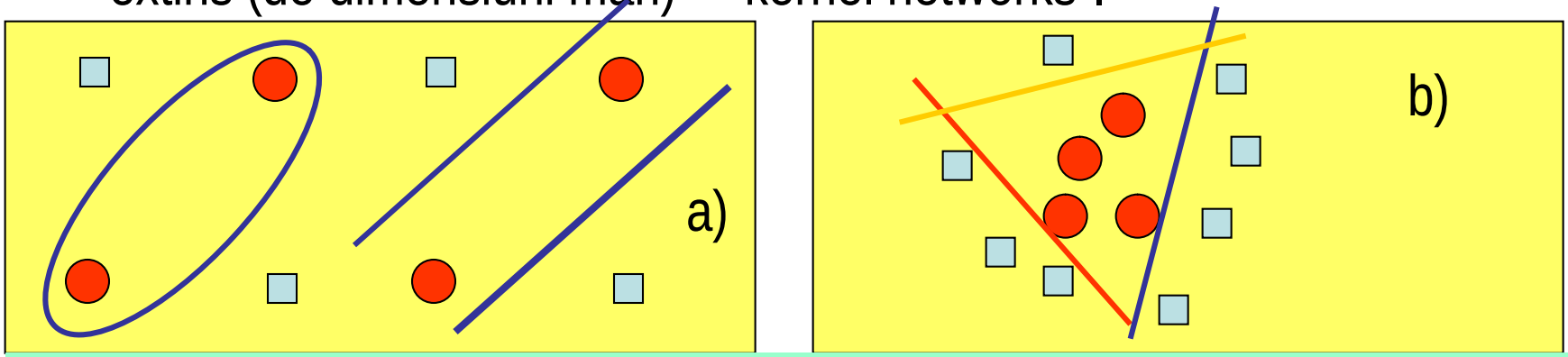
Modalitati de a construi aproximatorul universal:

a) Cresterea complexitatii modelului de neuron – ex. polinomial si "multi-nested" (imbricare multipla) Referinte pentru "Multi-nested":

UNIVERSAL CNN CELLS RADU DOGARU; LEON O. CHUA,
International Journal of Bifurcation and Chaos (IJBC) Volume: 9 No: 1 Year: 1999 pp. 1-48

b) Retele cu neuroni "standard": Perceptronul multi-strat (MLP)

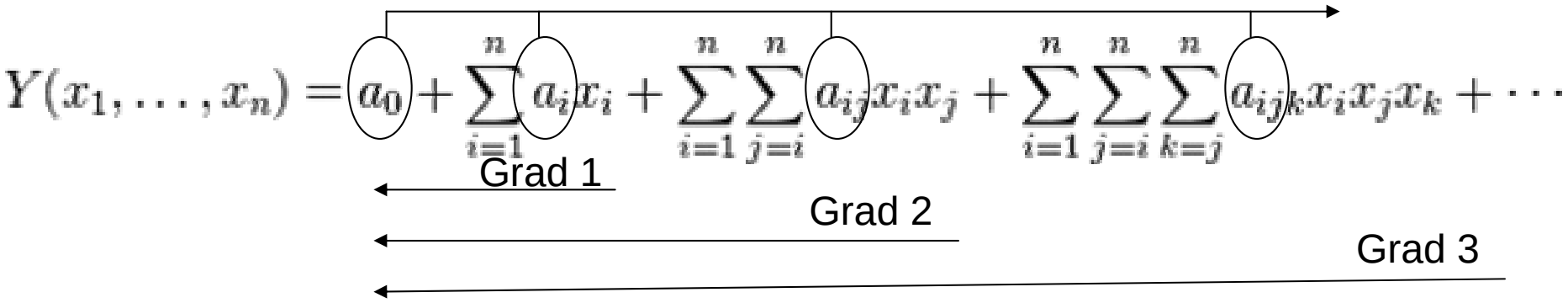
c) Adaline ("neuron standard") antrenat intr-un spatiu de intrare neliniar extins (de dimensiuni mari) – "kernel networks".



Explicatie grafica pentru fiecare metoda imbunatatita – vizualizarea problemei XOR

a) Doua exemple cu complexitati de implementare diferite:

1. Modelul polinomial al functiei (la modelul liniar se adauga termeni pana la gradul *k* inclusiv. S-a demonstrat ca in acest caz se pot reprezenta TOATE functiile Boole cu *k* intrari (AFU). De exemplu pentru functiile cu 2 intrari – modelul este polinomial de gradul 2.



http://www.pnnsoft.com/research-polynomial-neural-network.html

Dezavantaj: Numar foarte mare de parametri, multi operatori “ne-economici” (inmultire)

O versiune mai compacta prin inlocuirea functiei polinomiala cu o functie “nested” utilizand numai operatori “economici” (adunare, functie modul) →

2. Celula cu neliniariate imbricata multiplu (gen "Matryoshka")

Se exploateaza principiul "imbricarii" pentru a obtine structuri adaptive compacte (convenabile in aplicatii integrate) .

Echivalent cu un polinom de grad 8 !!

$$g(x) = |x|$$

Valoarea absoluta "modul de baza"

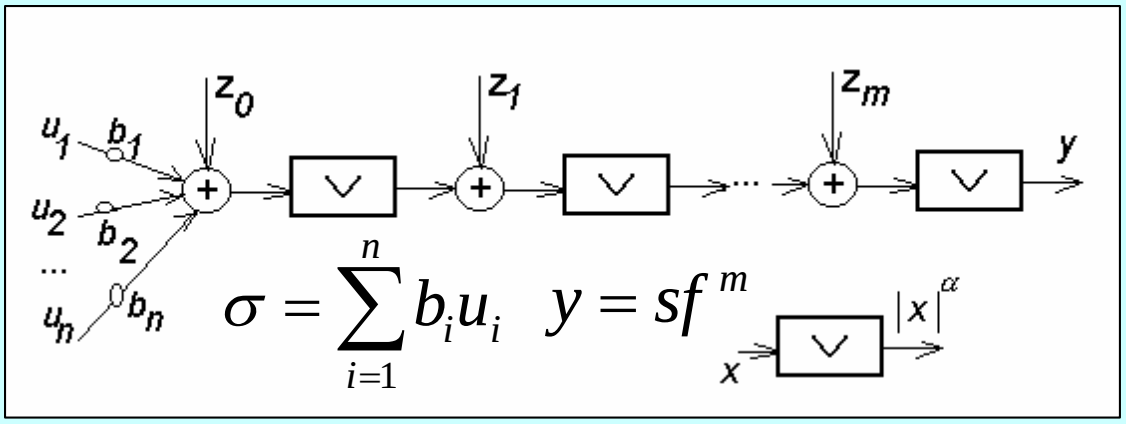
$$f(x) = z_3 + \Big|z_2 + \Big|z_1 + \Big|z_0 + x\Big|\Big|\Big|$$

3 valori absolute "imbricate"

2 segmente monotone

2^3 =8 segmente monotone

Sistem compact folosind modelul cu imbricare multipla ca functie de iesire (adugata unui model de neuron liniar)



Arhitectura celulei "multi-nested"

Se observa ca intrarile nu sunt restranse la valori binare. Iesirea este libera sa varieze intr-un domeniu continuu;

Prin aplicarea suplimentara a functiei sign(y) se obtine un model care poate reprezenta functii Boole (sau probleme de clasificare)

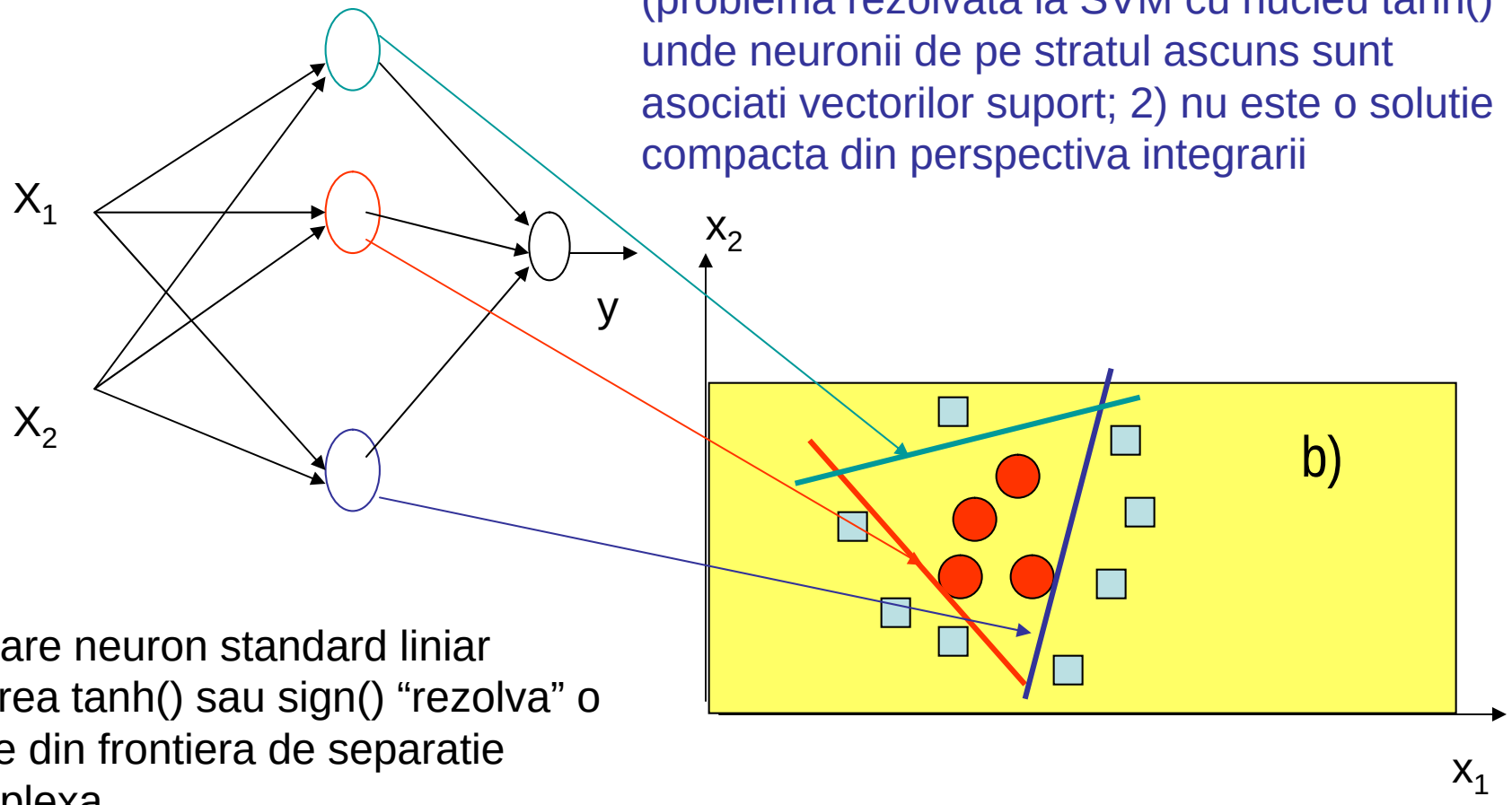
Poate fi antrenat folosind algoritmi de optimizare globala ("Simulated Annealing", Alopex, Algoritmi Genetici (GA) etc.)

Exemplu:

PARITY-8: $\{\mathbf{b} = [1,1,1,1,1,1,1,1,1], s = -1, \mathbf{z} = [0,-4,-2,-1]\}$,

b) Retele cu neuroni “standard”: Perceptronul multi-strat (MLP)

Dezavantaje: 1) determinarea numarului optim de neuroni se face prin tatonari (problema rezolvata la SVM cu nucleu tanh() unde neuronii de pe stratul ascuns sunt asociati vectorilor suport; 2) nu este o solutie compacta din perspectiva integrarii



Fiecare neuron standard liniar (iesirea tanh() sau sign() “rezolva” o parte din frontiera de separatie complexa

c)

Retele Neurale cu functii nucleu

Formule generale de definire a neuronului cu functii nucleu

$$y = \sum_{k=0}^{m} o_k w_k, \quad \text{unde } o_k = \varphi_k(\mathbf{x})$$

Adaline

Alegerea potrivita a functiei neliniare(denumita si “nucleu”)

Daca nucleele si parametrii lor sunt alesi independenti, invatarea se reduce la antrenarea LMS(Adaline).

De asemenea se mai poate numi si neuron “ascuns”

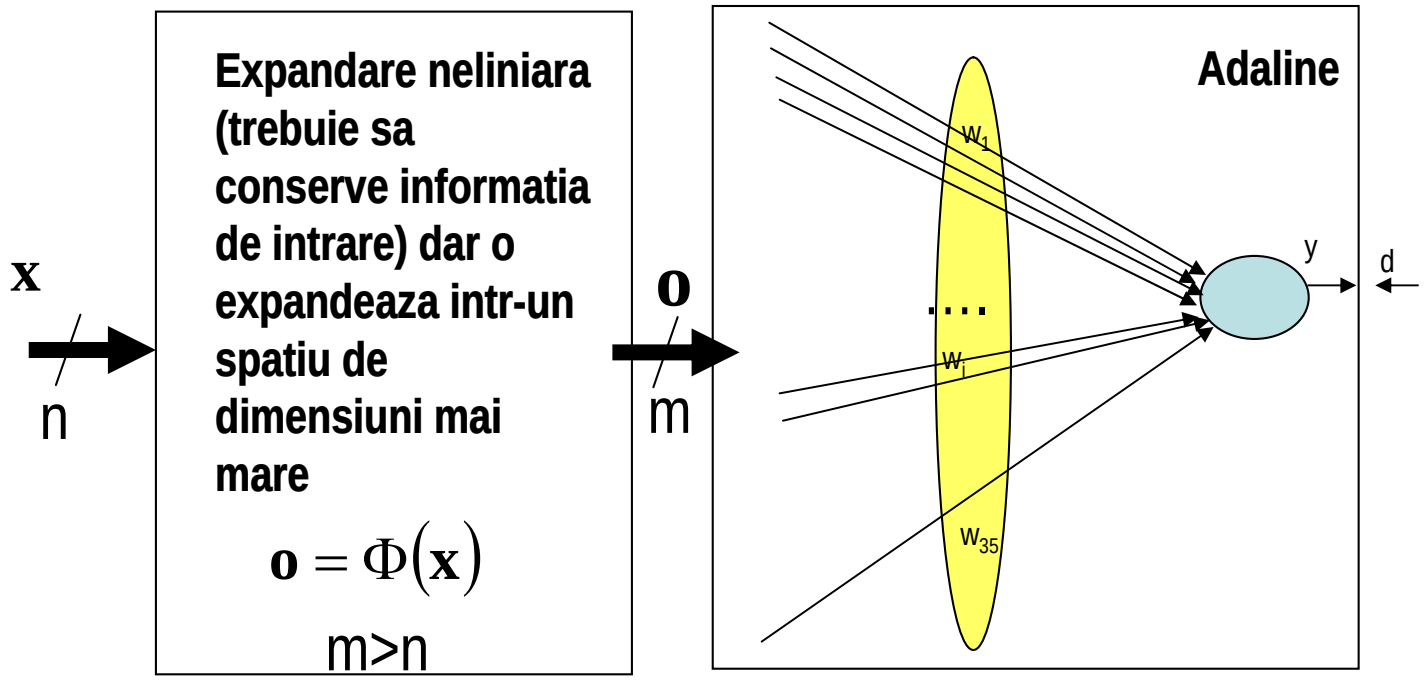
Avantaje:

-Antrenarea se efectureaza numai la nivelul Adaline (neuron liniar + algoritm LMS) si este garantat convergenta, simpla (complexitate de integrare), si rapida.

- Functiile nucleu se pot alege astfel incat sa optimizeze pe langa criteriul de performanta si pe cel al unei complexitati de integrare minime !

- De regula se prefera ca tot ansamblul functiilor nucleu sa fie controlat de un parametru generic unic (de exemplu o raza la o structura de tip RBF) care influenteaza dimensiunea spatiului de iesire “m” si prin urmare permite un acord al numarului de parametri (din Adaline) in sensul respectarii principiului Ockham → obtinerea unui maxim de performanta la generalizare

Retele neurale cu functii nucleu sunt de tip Adaline si opereaza intr-un spatiu de intrare neliniar expandat



Teorema Cover (1965): Daca m>>n o problema care nu este liniar separabila in spatiul de intrare x, poate deveni liniar separabila (si deci poate fi invatata de Adaline) in spatiul expandat o. Necesita doar o alegere potrivita a functiilor nucleu: $\Phi = \left[\varphi_1(\mathbf{x}),...\varphi_j(\mathbf{x}),...,\varphi_m(\mathbf{x})\right]$

Necesitatea unui set de test: Generalizare vs. Memorizare

Cu scopul de a imbunatati generalizarea (ex. capacitatea unui sistem de a raspunde corect la stimuli care nu au fost folositi in procesul de antrenare) se impart datele in doua seturi: TR (setul de antrenare) si TS (setul de testare).

O buna performanta utilizand exclusiv TR indica doar faptul ca sistemul inteligent are o buna CRF (capacitate de reprezentare functionala). Este insa posibil sa fie supradimensionat (overfitting). De obicei, performanta (ex. procentul de decizii eronate) se imbunatateste (scade) odata cu cresterea numarului de parametri.

Dar aceasta reprezinta numai o buna capacitate de a memoriza si nu este in mod necesar si o garantie pentru o buna capacitate de a generaliza (inteligenta).

Trebuie in acest caz folosit si un set de test aditional pentru a masura performantele de generalizare si pentru optmizarea numarului de parametri astfel incat sa evitam over/under fitting (supra/sub – dimensionarea)

$$TS=\{ ...(x^p,d^p) ..\}_{p=1..,ts}$$

$$TR=\{ ...(x^p,d^p) ..\}_{p=1..,tr}$$

TS si TR sunt disjuncte (nu contin elemente comune)