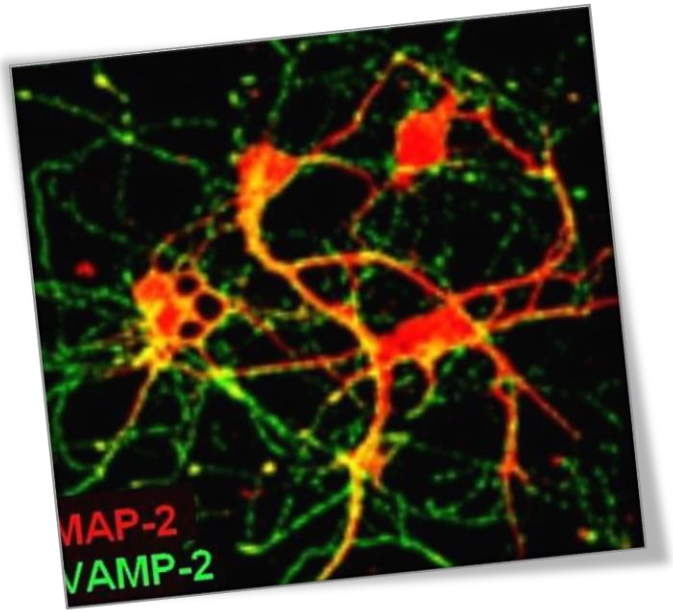
PROCESAREA CUNOŞTINȚELOR ŞI CALCUL INTELIGENT - PCCI

Reţele neuronale artificiale

- Introducere şi istoric
- Neuronul elementar

Cuprins:

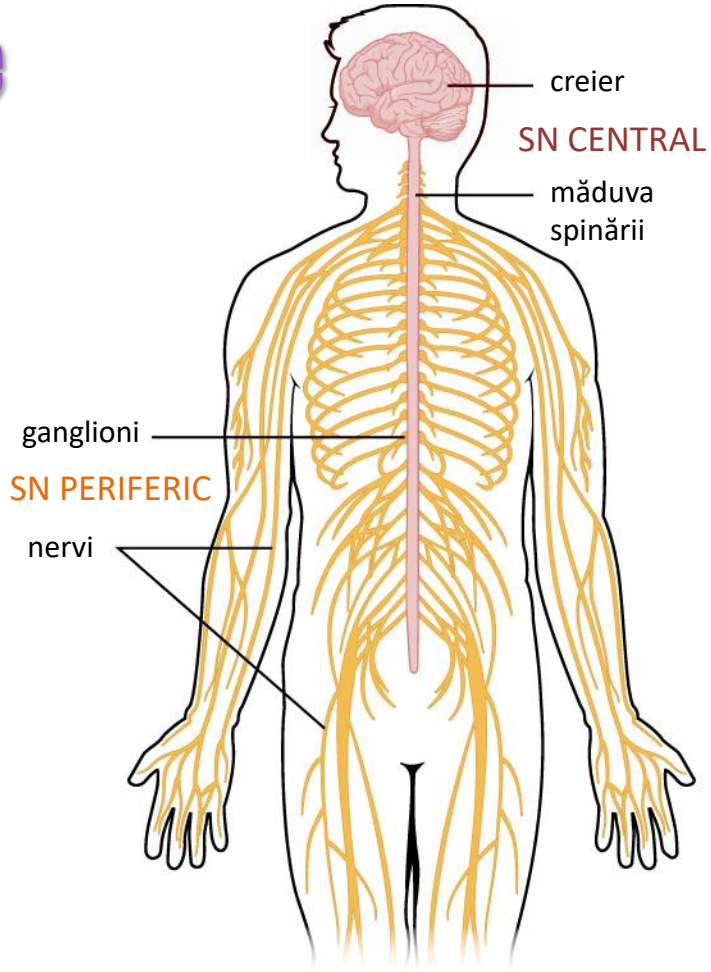
- Reţele neuronale şi inspiraţia lor biologică
- Neuronul matematic elementar
  - Modelarea neuronului biologic ca neuron matematic
  - Convenţii matematice de notare, vectorizarea calculelor
  - Istoric
  - Tipuri de neuroni artificiali folosiţi de RNA
    - Liniar
    - Sigmoid logistic
- Antrenarea neuronului elementar
  - Regresia liniară
  - Algoritmul descreşterii gradientului pentru antrenarea neuronului liniar şi logistic



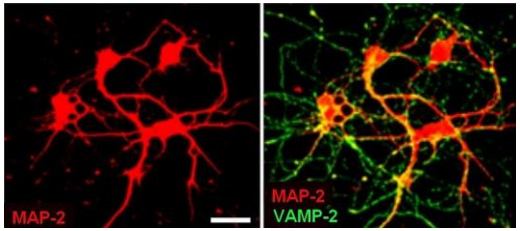
Rețele neuronale biologice şi artificiale

Rețele neuronale biologice

- Sistemul nervos îndeplineşte mai multe funcţii:
  - Receptarea stimulilor externi
  - Transmiterea informaţiei
  - Procesarea informaţiei, senzorială şi cognitivă, în centrii nervoşi din creier şi coloana vertebrală
  - Transmiterea unor reacţii



[OpenStax College, Wikipedia]



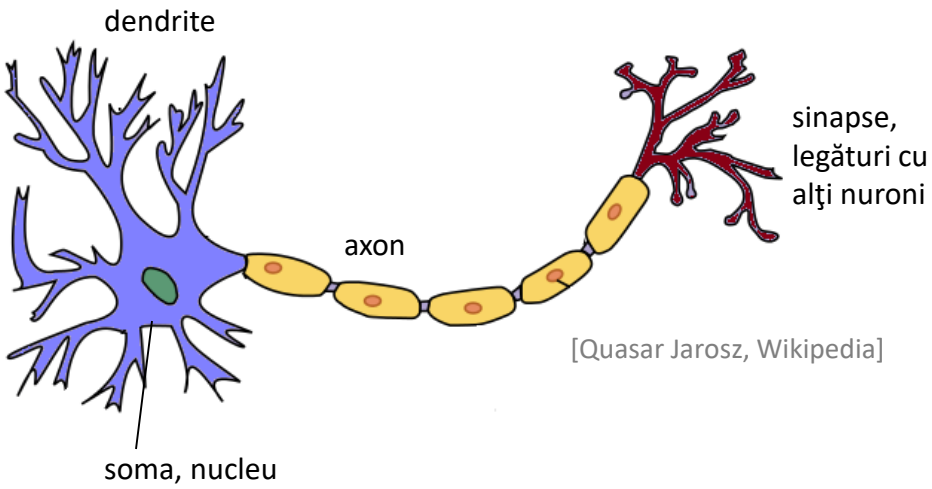
[Noemi Tonna et al - Wikipedia]

- Senzorii, căile de transmisie şi centrii nervoşi sunt alcătuite din neuroni interconectaţi, care formează reţele neuronale

Neuronul biologic

- Unitatea funcţională elementară a sistemului nervos este neuronul.

- Alcătuire:
  - intrări: dendrite excitatorii şi inhibitorii
  - corp procesor – soma
  - ieşire: axon şi sinapse de legătură cu alţi neuroni



[Quasar Jarosz, Wikipedia]

- Funcţionare: când suma tuturor impulsurilor primite prin dendrite (inhibitorii + excitatorii) depăşeşte un prag de activare, neuronul trimite un impuls prin axon

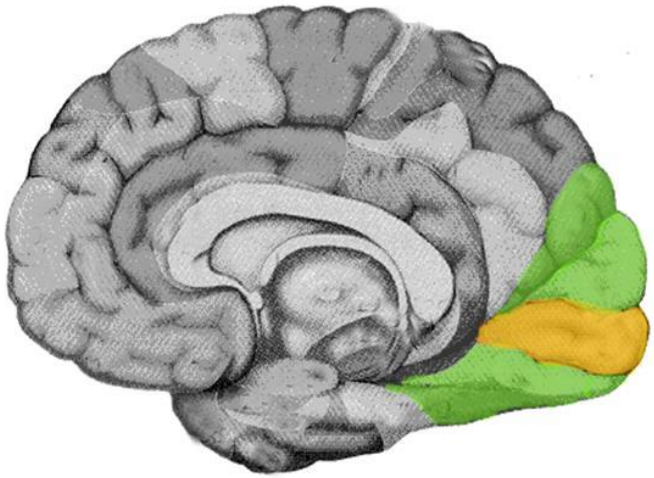
Creierul biologic

- Biologic, învăţarea sau deprinderea se realizează prin întărirea anumitor sinapse (legături) dintre anumiţi neuroni, formând căi sinaptice. Uitarea este sinonimă cu distrugerea acestor căi.

- Încă nu cunoaştem mecanismele recunoaşterii sau ale asocierii.

- Creierul copilului este o pânză albă; reţelele de neuroni din anumite zone ale cortexului formează centrii nervoşi prin antrenare (exersare) şi specializare, în copilăria timpurie, şi pot fi “reprogramate”.

- Cortexul arată ca o pânză întinsă pe suprafaţa creierului



[Wikipedia]

Centrul nervos vizual din creier

Rețele neuronale artificiale

- Elementul de bază al oricărei RNA este neuronul elementar.

- Mai mulţi neuroni artificiali interconectaţi formează o reţea neuronală artificială (RNA)

- Diversele tipuri de RNA se diferenţiază prin:
  - Arhitectură (complexitate, modul de interconectare al neuronilor)
  - Funcţia îndeplinită (de aproximare, clasificare etc.)

- RNA învaţă prin antrenare.

Rețele neuronale artificiale (RNA)

- RNA sunt algoritme informatice inspirate din funcţionarea sistemului nervos al fiinţelor vii, pe care o modelează prin calcule matematice şi instrucţiuni de program.

- RNA actuale nu pot simula funcţionarea sistemului nervos ca un tot unitar, deosebit de complexă şi în mare parte încă necunoscută, ci doar funcţii specifice ale acestuia (memorarea, recunoaşterea).

- Mecanismele de inspiraţie biologică sunt aplicate pentru rezolvarea unor probleme specifice din diverse domenii ale ingineriei: prognoze, recunoaşterea formelor, clasificare.

De ce RNA în loc de rezolvare analitică ?

- Deoarece problemele practice depăşesc adesea limitele minţii umane, prin complexitate, volum imens de date, caracteristici importante greu de identificat într-un noian de parametri posibili. Uneori nu cunoaştem foarte bine problema, însă avem la dispoziţie un volum mare de date. RNA învaţă problema.
  - Calculatorul rezolvă rapid un număr imens de calcule secvențiale, în vreme ce mintea umană este imbatabilă la procesarea paralelă a informaţiei.

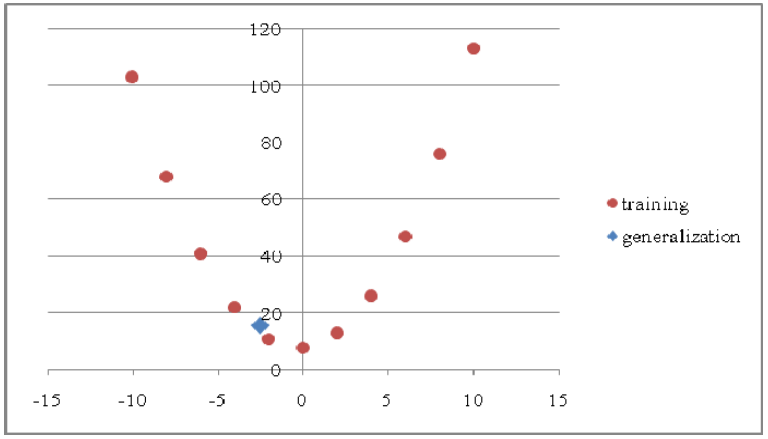
- Deoarece este de preferat identificarea unui model matematic general aplicabil unei categorii de probleme şi aceleiaşi probleme atunci când se actualizează datele de intrare.
  - Modelarea analitică rezolvă de obicei o problemă particulară, pe datele existente la un moment dat.

Învățarea RNA

- Tipuri de învăţare:
  - Supravegheată (supervised learning)
  - Nesupravegheată (unsupervised learning)
  - Recompensă-pedeapsă (reinforcement learning)

Învățarea supravegheată

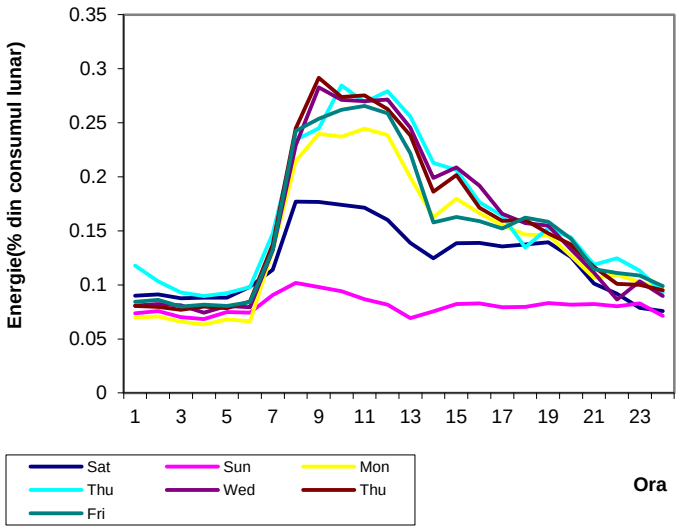
- Se cunosc perechi de intrare-ieşire, excitaţie-răspuns, datele sunt “etichetate”.
- Reţeaua învaţă datele cunoscute şi apoi generalizează pe baza învăţării pentru date necunoscute.
- Utilizări: regresii, prognoze, clasificări.
- Cea mai cunoscută RNA cu învăţare supravegheată: perceptronul multistrat (multilayer perceptron - MLP)

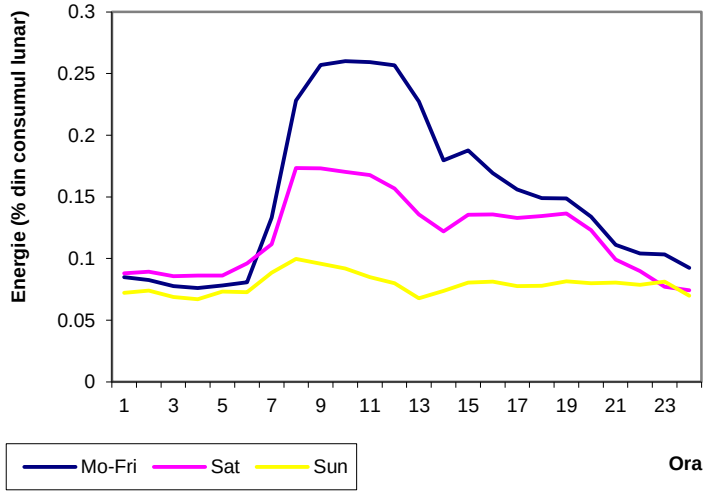


	x	y	f(x,y)
antrenare	-10	3	103
	-8	4	68
	-6	5	41
	-4	6	22
	-2	7	11
	0	8	8
	2	9	13
	4	10	26
	6	11	47
	8	12	76
	10	13	113
gen.	-2.5	9.5	15.75

Învățarea nesupravegheată

- Datele sunt “neetichetate”, RNA le sortează şi le grupează pe categorii, învaţă diferenţe şi caracteristici comune.
- Utilizare: clasificare, grupare pe categorii (clustering)
- Cea mai cunoscută RNA cu învăţare nesupravegheată: hărţile cu autoorganizare Kohonen (Self Organizing Feature Maps – SOM, SOFM)

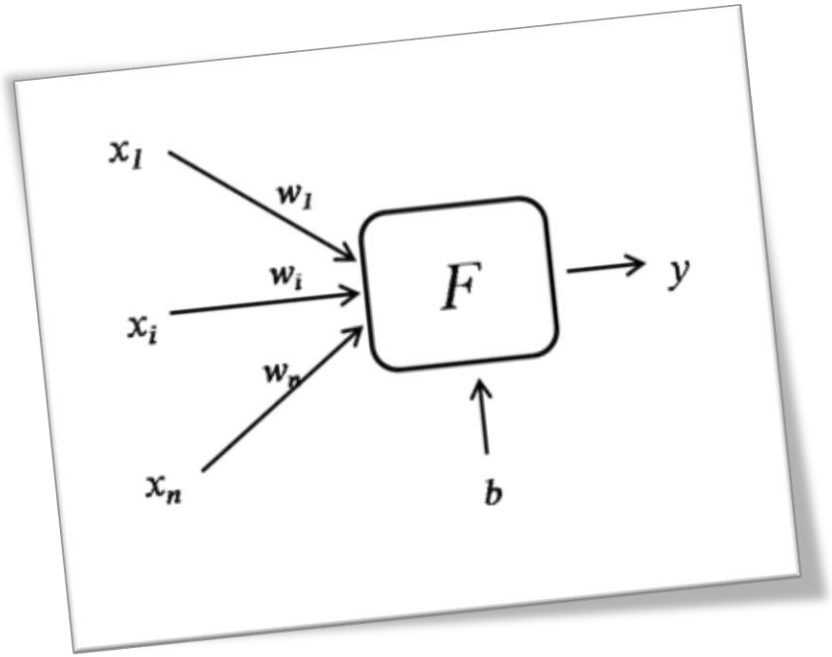




Clasificarea curbelor de sarcină ale consumatorilor (profilarea)

Învățarea prin recompensă

- Încurajarea neuronilor cu performanţe ridicate, care contribuie mai mult la rezolvarea problemei, şi descurajarea (penalizarea) celor cu performanţe slabe.



Neuronul elementar

Modele de principiu ale proceselor

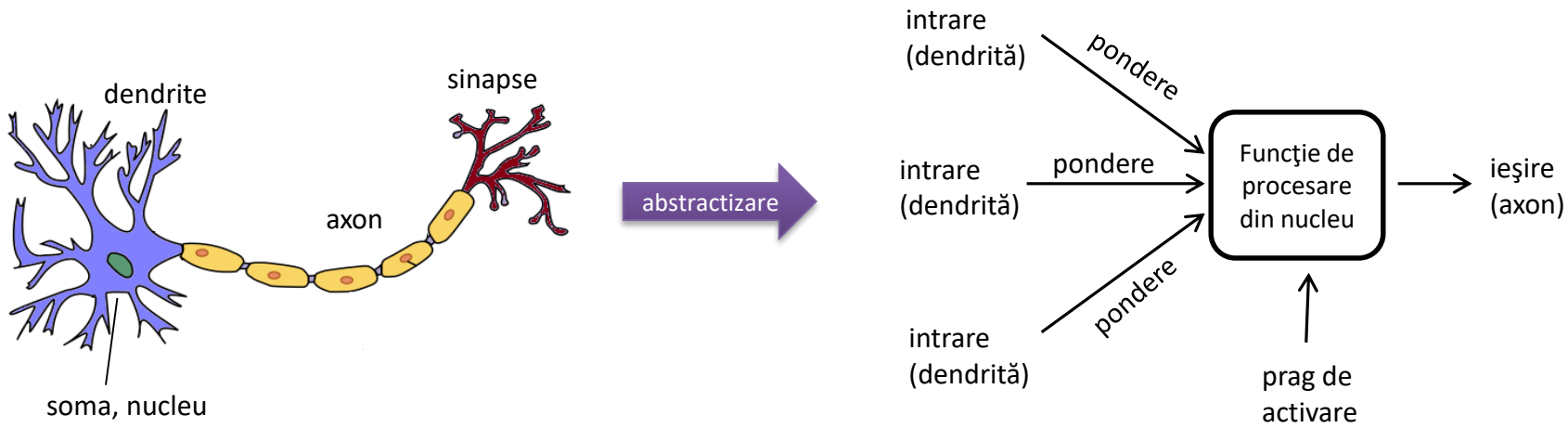
- În ştiinţă, se preferă reprezentarea proceselor reale prin modele matematice simplificate, care
  - ❖ permit înţelegerea mai bună a principiilor simple din spatele unui fenomen complex,
  - ❖ simplifică calculele matematice.

- Gradul de complexitate al modelului poate fi crescut ulterior, pe măsura înţelegerii lui, pentru a se apropia de realitate.

- Exemplu: calculul de regim permanent al reţelelor electrice:
  - o Construcţie simetrică pe cele trei faze,
  - o Sarcină simetrică pe cele trei faze, constantă pe un interval de timp considerat.

Neuronul elementar matematic

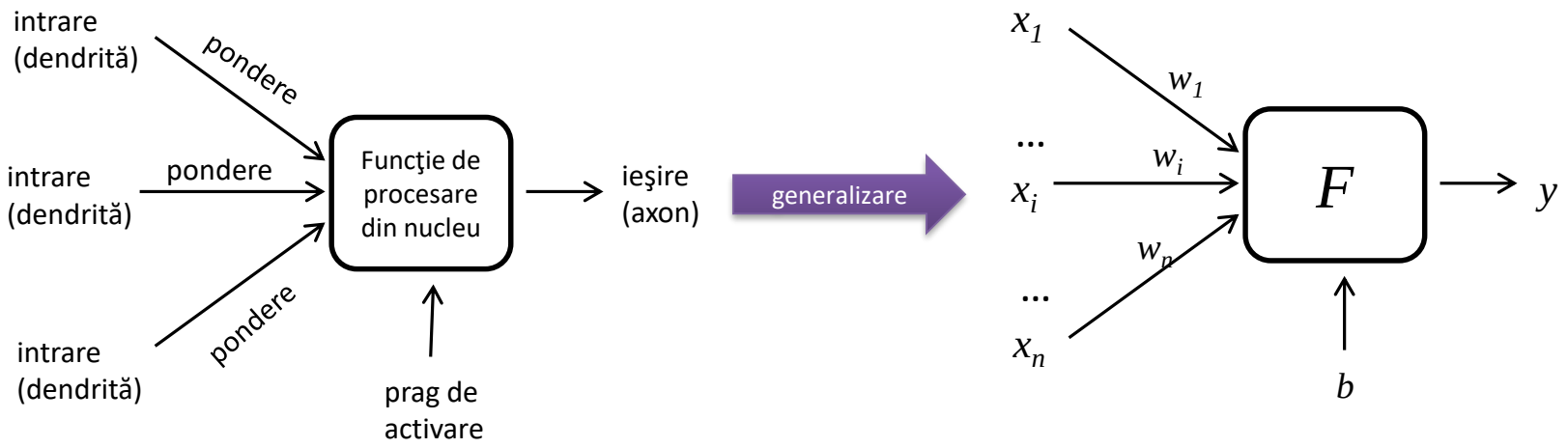
- Neuronul artificial este un model simplificat şi abstract al neuronului biologic.



- Neuronul biologic: fiecare intrare este atenuată (ponderată), deoarece un neuron primeşte pe intrare o fracţiune din impulsul distribuit prin sinapsele neuronilor anteriori din lanţ sau o fracţiune din impulsul senzorial.
- Spre deosebire de neuronul biologic, ponderile neuronului artificial devin frecvent supraunitare.

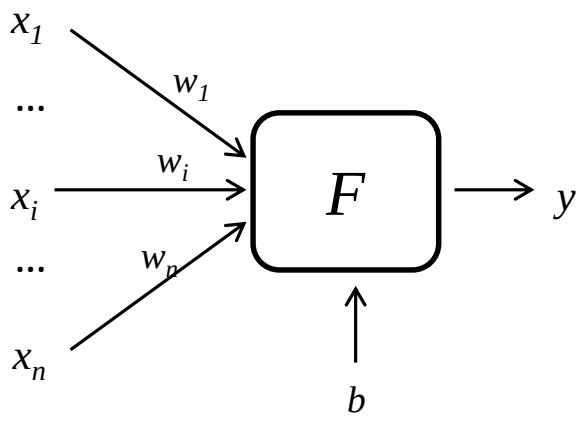
Neuronul elementar matematic

Matematica permite modelarea abstractizată şi generalizată a realităţii.



- Notaţii:
  - F – funcţie de activare a neuronului (activation function)
  - x_i – parametru (feature) din modelul de intrare $X = [x_1, \ldots, x_i, \ldots x_n]$
  - w_i – pondere (weight) asociată parametrului x_i

Neuronul elementar matematic



- Scriere desfăşurată (cu ecuaţii matematice):

$$y = b + F(x_1 \cdot w_1 + \ldots + x_i \cdot w_i + \ldots + x_n \cdot w_n)$$

unde:

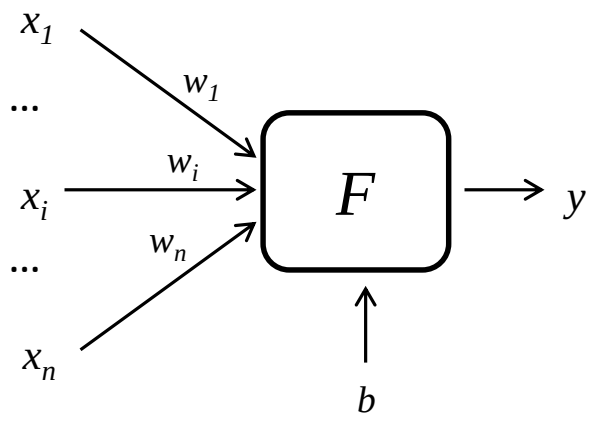
$$b + x_1 \cdot w_1 + \ldots + x_i \cdot w_i + \ldots + x_n \cdot w_n \overset{not}{=} net$$

$$y = F(net)$$

sau

$$y = F\left(b + \sum_{i=1}^{n} w_i \cdot x_i\right)$$

Neuronul elementar matematic



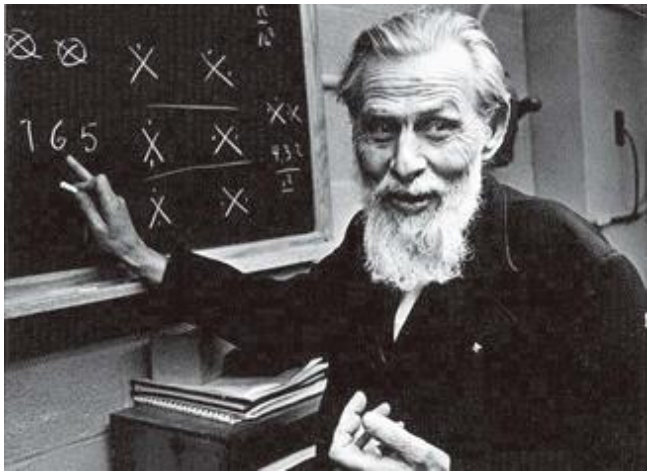
- Scriere compactă (cu matrice şi vectori), pentru programatori

$$net = b + \begin{bmatrix} x_1 \\ \dots \\ x_i \\ \dots \\ x_n \end{bmatrix} \cdot \begin{bmatrix} w_1 & \dots & w_i & \dots & w \end{bmatrix} = b + [X] \cdot [W]^T$$

$$y = F(net) = F(b + [X] \cdot [W]^T)$$

Istoric

- Primul model de neuron artificial



Warren Sturgis McCulloch, neurofiziolog şi Walter Pitts, logician

Lucrarea “A Logical Calculus of the Ideas Immanent in Nervous Activity”

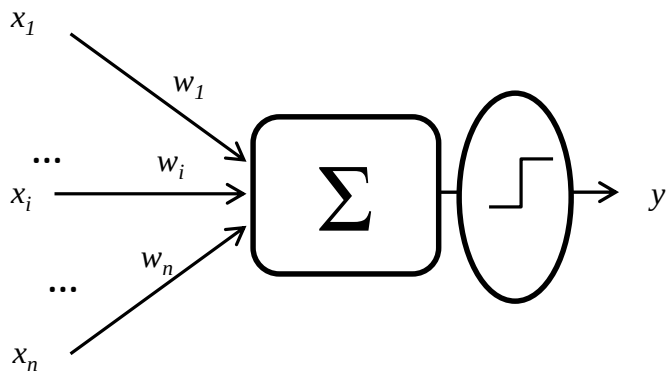
[http://web.csulb.edu/~cwallis/artificialn/walter_pitts.html]

Neuronul McCulloch-Pitts

- Este de tip logic, intrările şi ieşirile pot avea valori 0 sau 1
- Ponderile au aceeaşi valoare
- Pe o pondere se pot primi mai multe intrări
- O intrare este inhibitoare, cu caracter absolut (de veto)

- Funcţionare: dacă suma intrărilor ponderate este mai mare decât un prag definit şi intrarea inhibitoare este 1, ieşirea are valoarea 1; în caz contrar, 0

- Utilizabil pentru simularea porţilor logice



$$net = \sum_{i=1}^{n} w_i \cdot x_i$$

$$y = \begin{cases} 1, & daca \quad net > prag, \quad x_{inhib} = 0 \\ 0 & in\ caz\ contrar \end{cases}$$

Istoric

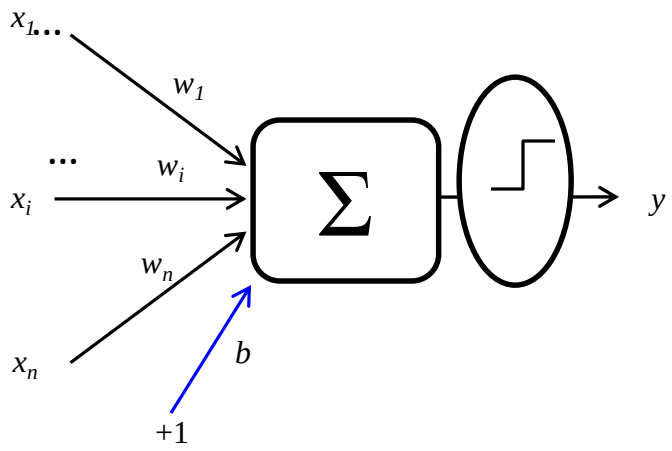
- Perceptronul

[library24.library.cornell.edu]

Frank Rosenblatt, psiholog,
cercetător la Universitatea Cornell, SUA

Neuronul de tip perceptron

- Intrările şi ponderile au valori reale, pozitive sau negative
- Funcţionare: dacă suma intrărilor ponderate este mai mare decât un prag definit, ieşirea are valoarea 1; în caz contrar, ieşirea are valoarea 0
- Pragul b (bias) este considerat ca o intrare cu valoare 1 a cărei pondere este variabilă
- Utilizare: clasificator



$$net = -b + \sum_{i=1}^{n} w_i \cdot x_i$$

$$y = \begin{cases} 1, & daca \quad net > 0 \\ 0 & in\ caz\ contrar \end{cases}$$

Algoritmul de învățare al perceptronului

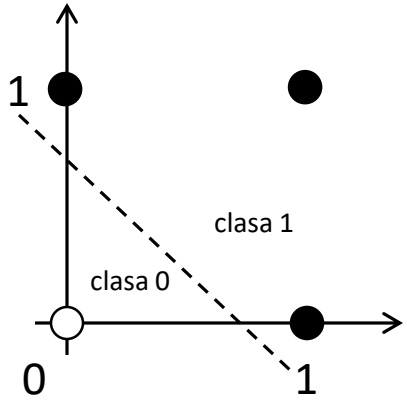
- Pentru toate modelele de intrare cunoscute, identifică valoarea corectă a ponderilor, aceeaşi pentru toate modelele, astfel încât neuronul să identifice corect dacă un model de intrare aparţine (1) sau nu (0) unei clase, după următorul algoritm [Hinton]:

- Pentru fiecare model de intrare:
  - Dacă neuronul identifică clasa 0 în loc de 1, adună modelul de intrare la ponderile curente
  - Dacă neuronul identifică clasa 1 în loc de 0, scade modelul de intrare din ponderile curente
  - Dacă neuronul identifică corect clasa din care face parte modelul, nu modifică ponderile

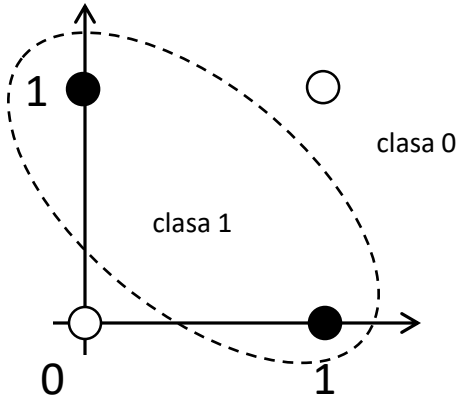
- Algoritmul găseşte garantat ponderile corecte după un număr de paşi, dacă ele există

Limitările perceptronului

- În 1969, Marvin Minsky şi Seymour Papert demonstrează în cartea *Perceptrons* că un singur perceptron nu poate învăţa decât funcţii liniar separabile (“the XOR debate”).



SAU, OR – funcţie liniar separabilă



SAUE, XOR – funcţia nu este liniar separabilă

- Cercetătorii vremii au extins implicit această concluzie şi la reţelele formate din mai mulţi neuroni, ceea ce a dus la diminuarea drastică a interesului pentru RNA.

Istoric

- John Hopfield - reţele neuronale asociative

- David Everett Rumelhart & James McClelland

"Parallel Distributed Processing: Explorations in the Microstructure of Cognition"

- Teoria calculului conexionist: reprezentarea informaţiei într-un sistem cognitiv se realizează în conexiunile dintre unităţi elementare mici (neuroni)
- Primele modele de simulare pe calculator a percepţiei umane

Neuronul elementar după 1968

- Perceptronul are funcţia de activare treaptă (0 şi 1)

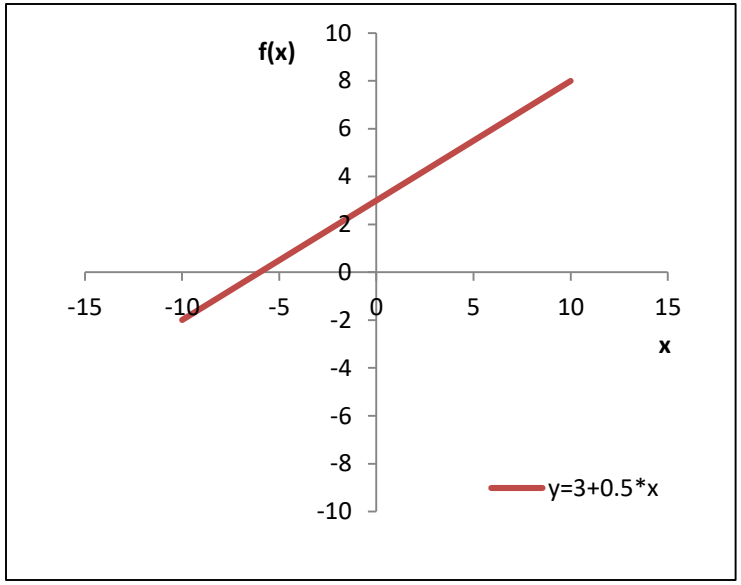
- Cercetările din anii ‘80 au propus noi tipuri de funcţii de activare pentru neuroni şi algoritme pentru reţele neuronale.

- Cele mai frecvent utilizate funcţii de activare:
  - Funcţia liniară
  - Funcţia sigmoid logistic
- Altele: tangent hiperbolic, clopotul gaussian

Funcții de activare

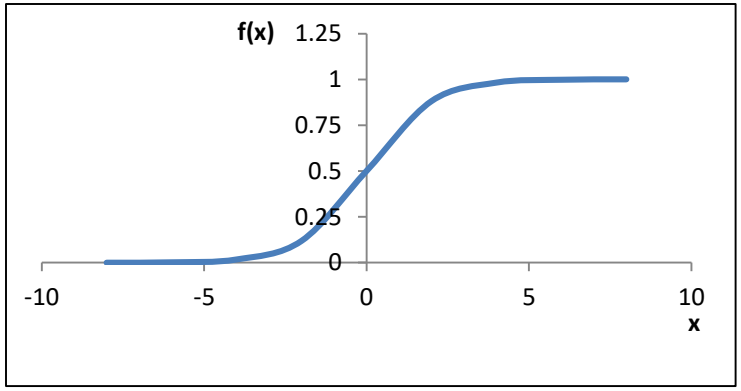
- Liniară:

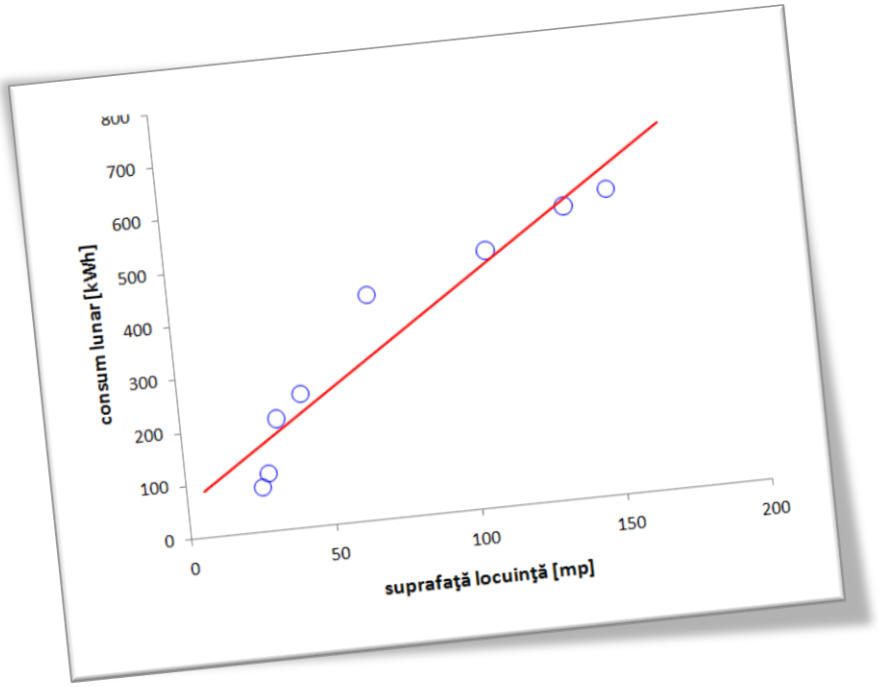
$$f(x) = a_0 + a_1 \cdot x$$



- Sigmoid logistic:

$$f(x) = \frac{1}{1+e^{-x}}$$





ÎNVĂȚAREA SUPRAVEGHEATĂ
Regresia liniară şi algoritmul de descreştere a gradientului pentru neuronul elementar cu funcție de activare liniară

Exemplu de problemă:

- **Estimarea consumului pentru un client nou al unui furnizor, în vederea negocierii unui tarif reciproc avantajos**
- Date de intrare: preluate din baza de date a furnizorului pentru consumatorii aflaţi deja sub contract


EPT energie pentru tine

CONTRACT DE FURNIZARE A ENERGIEI ELECTRICE

Încheiat astăzi, **31 februarie 2085** între

(1) Consumator: **Vasile Popescu**, CNP **12307310877531**

şi

(2) Furnizor: SC EPT Energy S.A.

pentru loc consum tip **casnic**, suprafaţa **253** mp, situat la **adresa Str. Viitorului nr. 1**, oraş **Iaşi**, jud. **Iaşi**

tip tarif **CRII02,** pe perioada **2 ani,** având ca obiect furnizarea energiei electrice în imobil, în condiţiile asigurării nivelului de calitate al energiei şi puterii minime detaliate conform negocierilor purtate între reprezentanţii Consumatorului şi ai Furnizorului.

reprezentanţii Consumatorului şi ai Furnizorului.

- Neuronul învaţă să prezică în viitor consumul unui client nou (nr. 10) pe baza consumurilor şi altor caracteristici ale clienţilor folosiţi în datele de intrare.

Nr.	Titular contract	Persoane	Suprafaţă locuinţă [mp]	Studii	Venit lunar [lei]	Consum mediu [kWh /lună]	Putere instalată [kW]
1	Nicolae Popa	3	110	superioare	3350	482	10
2	Vasile Ionescu	5	33	liceu	3100	208	8
3	Maria Popovici	2	42	superioare	5480	250	15
4	Adriana Elisei	3	138	superioare	4450	550	20
5	Gabriela Dăscălescu	4	68	superioare	2300	422	23
6	Aurel Degeratu	5	26	liceu	3200	83	6
7	Paul Irimciuc	4	28.5	liceu	2500	108	6
9	Florin Mihalcea	2	153	doctorat	6600	573	12

Exemplu de problemă

- Pentru învăţare, se reţin din datele cunoscute doar cele relevante şi se exprimă matematic sub forma unei funcţii;

Nr.	Titular contract	Persoane	Suprafaţă locuinţă [mp]	Studii	Venit lunar [lei]	Consum mediu [kWh /lună]	Putere instalată [kW]
			x_1		x_2	y	x_3
1	Nicolae Popa	3	110	superioare	3350	482	10
2	Vasile Ionescu	5	33	liceu	3100	208	8
3	Maria Popovici	2	42	superioare	5480	250	15
4	Adriana Elisei	3	138	superioare	4450	550	20
-	...	...	...	...	...	...	...
7	Paul Irimciuc	4	28.5	liceu	2500	108	6
9	Florin Mihalcea	2	153	doctorat	6600	573	12

Presupunem: consumul mediu lunar depinde de suprafaţa casei, venitul lunar al familiei, puterea electrică a aparatelor din casă.

Matematic, parametrizat: $y=f(x_1, x_2, x_3)$

- x_1, x_2, x_3 – caracteristici sau parametri de intrare (features)
- $f(x_1, x_2, x_3)$ – funcţie sau transformare, relaţie între intrări şi rezultat

Exemplu de problemă

Nr.	Titular contract	Persoane	Suprafaţă locuinţă [mp]	Studii	Venit lunar [lei]	Consum mediu [kWh /lună]	Putere instalată [kW]
1	Nicolae Popa	3	110	superioare	3350	482	10
2	Vasile Ionescu	5	33	liceu	3100	208	8
3	Maria Popovici	2	42	superioare	5480	250	15
4	Adriana Elisei	3	138	superioare	4450	550	20
5	Gabriela Dăscălescu	4	68	superioare	2300	422	23
6	Aurel Degeratu	5	26	liceu	3200	83	6
7	Paul Irimciuc	4	28.5	liceu	2500	108	6
9	Florin Mihalcea	2	153	doctorat	6600	573	12

- O coloană din tabel = caracteristică sau parametru (feature).
- O linie din tabel = un caz rezolvat al problemei sau un model de antrenare.

- Cunoaştem doar datele din tabel, nu şi funcţia $y=f(x_1, x_2, x_3)$, care poate avea o expresie matematică complicată ($y=0.385\cdot x_1+\exp(x_2)/5\cdot x_3$).

- Vom aproxima funcţia *f* reală cu alta, mai simplă - procedeul se numeşte regresie.

Regresia

- Tipul de regresie ales trebuie să îndeplinească două condiţii:
  - Să aproximeze cât mai aproape funcţia căutată (să identifice cât mai bine funcţia originală necunoscută)
  - Să aibă gradul cât mai mic (să fie uşor de calculat).

Regresia

- Regresia polinomială - funcţia este un polinom

$$f(x) = a_0 + a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \ldots + a_1 \cdot x + a_0$$

  - Liniară - funcţia este un polinom de gradul I - cea mai mare putere la care e ridicat x este 1

$$f(x) = a_0 + a_1 \cdot x \ + a_2 \cdot x_2$$

  - Pătratică: polinom de gradul II - cea mai mare putere la care e ridicat x este 2

$$f(x) = a_0 + a_1 \cdot x_1^2 + a_2 \cdot x_2$$

  - Cubică (grad III), quadratică (grad IV)…

Aproximarea prin regresie polinomială

consum mediu= f (suprafaţa casei, venit, putere instalată),

*poate consum=5+0.5*suprafaţă+0.35*venit+0.15*putere*

$$y = f(x_1, x_2, x_3) = 5 + 0.5 \cdot x_1 + 0.35 \cdot x_2 + 0.15 \cdot x_3$$

*sau poate consum =0.43*suprafaţă+0.12*venit²*

$$y = 0.43 \cdot x_1 + 0.12 \cdot x_2^2 + 0 \cdot x_3$$

Căutăm să determinăm valoarea cea mai bună a coeficienţilor numerici, care dau funcția ce trece cel mai aproape de punctele cunoscute, (care aproximează cel mai bine tendinţa).

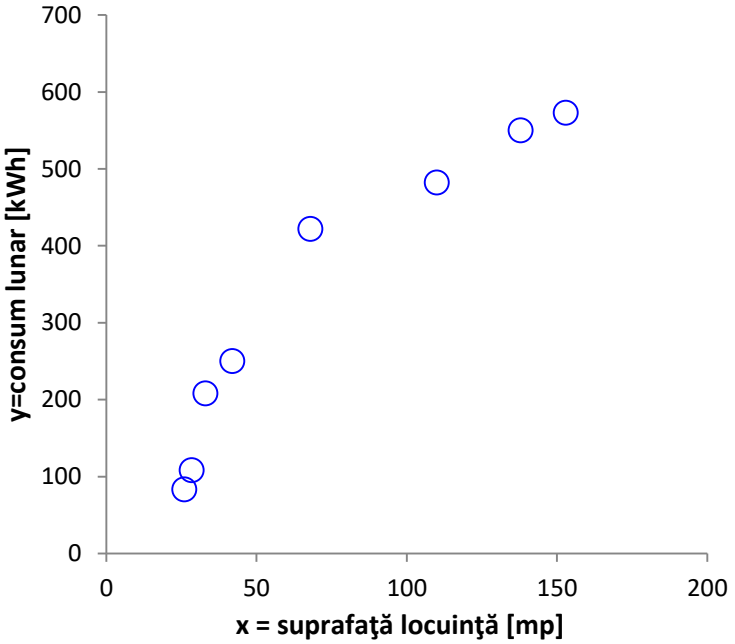
Problema noastră...

- Iniţial, pentru simplificare, păstrăm pentru funcţia y o singură variabilă, suprafaţa casei:

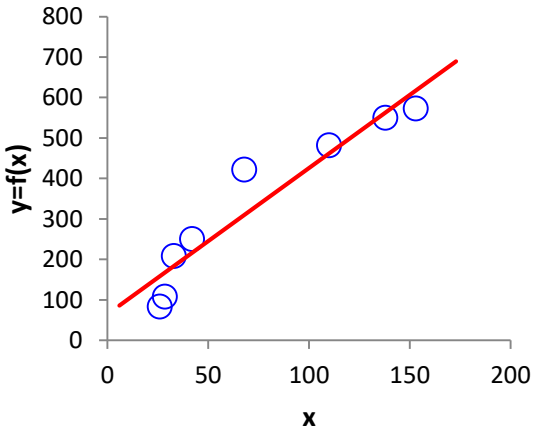
y=f (suprafaţa locuită, venit lunar, putere instalată) devine *y= f(suprafaţa locuită)*

$y=f(x_1, x_2, x_3)$ devine $y= f(x)$

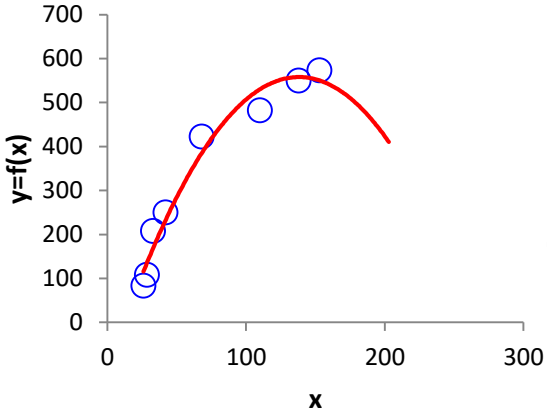
Model	Suprafaţă locuită [mp]	Consum mediu [kWh /lună]
	x	y
$(x^{(1)},y^{(1)})$	110	482
$(x^{(2)},y^{(2)})$	33	208
$(x^{(3)},y^{(3)})$	42	250
$(x^{(4)},y^{(4)})$	138	550
$(x^{(5)},y^{(5)})$	68	422
$(x^{(6)},y^{(6)})$	26	83
$(x^{(7)},y^{(7)})$	28.5	108
$(x^{(8)},y^{(8)})$	153	573



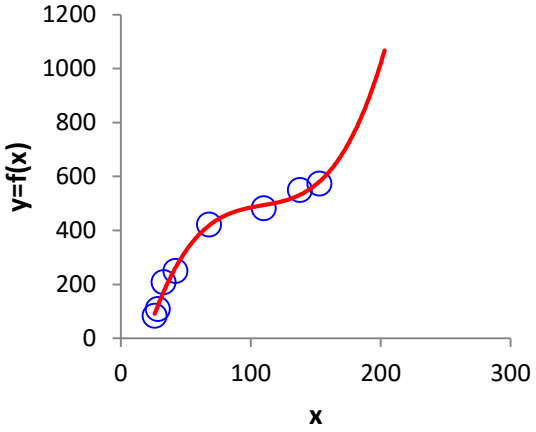
Regresie liniară vs grad superior



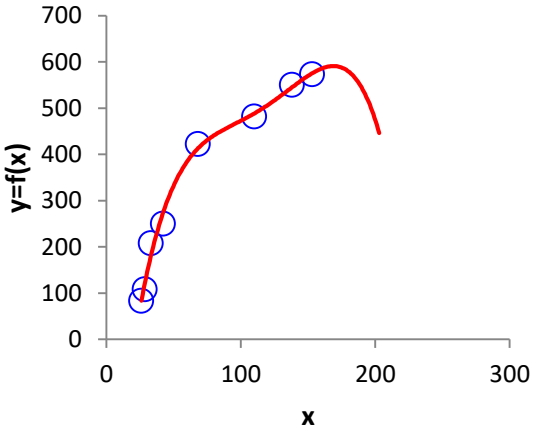
liniară
(grad I)



pătratică
(grad II)



cubică
(grad III)

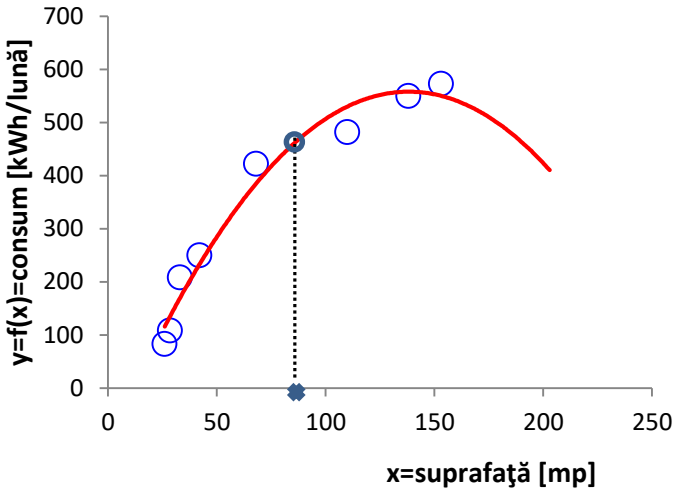


quadratică
(grad IV)

Sursa: datele din tabelul anterior + Excel

Aproximarea prin regresie

- La ce ne ajută asta?

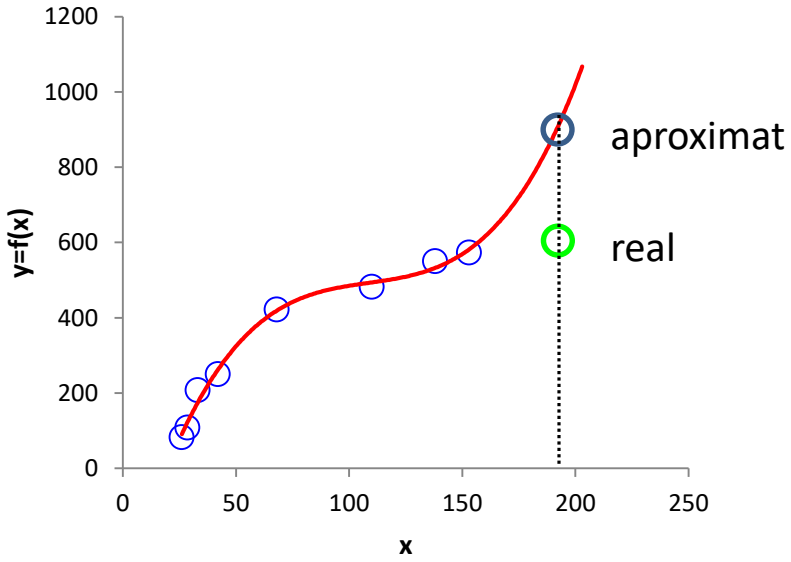


Putem determina valoarea aproximativă a funcţiei pentru oricare alt punct intermediar, necunoscut (GENERALIZARE).

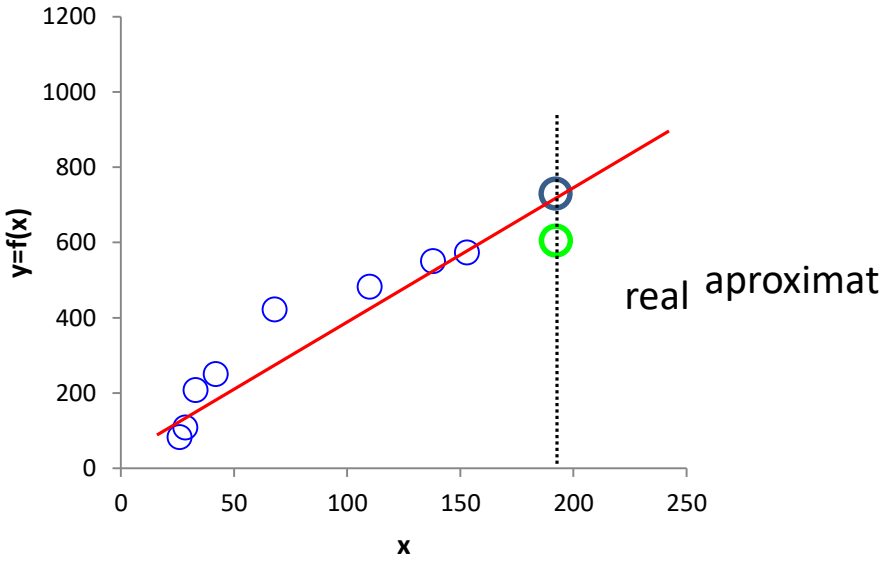
În cazul problemei noastre, dacă furnizorul primeşte un nou client, îi poate aproxima consumul aşteptat pe baza suprafeţei casei, venitului şi puterii electrice totale a consumatorilor instalaţi în casă.

Regresie liniară vs grad superior

- O aproximare bună pe datele iniţiale nu garantează o generalizare bună.



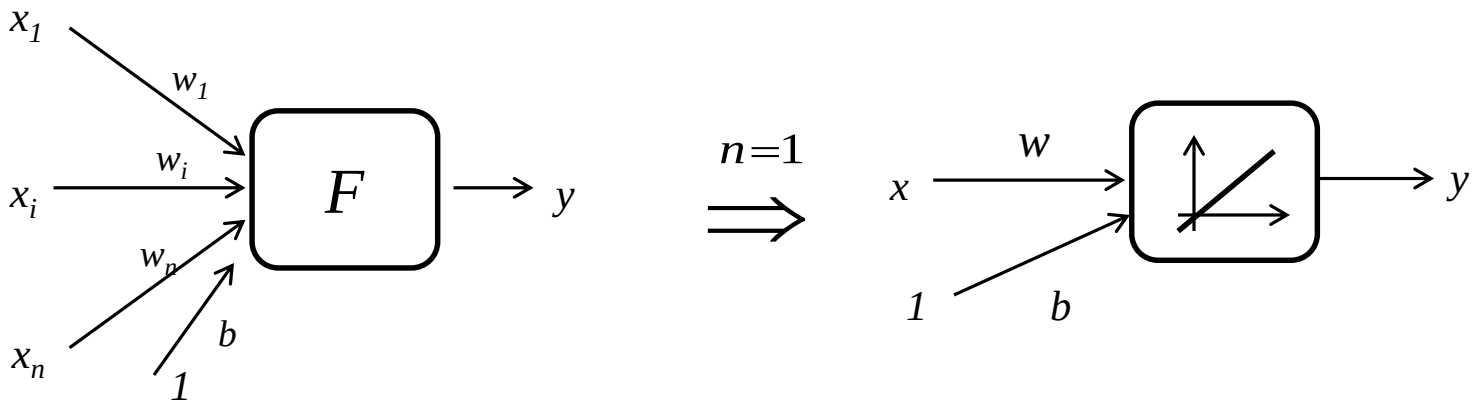
Regresie cubică (grad III)



Regresie liniară (grad I)

Neuronul elementar şi regresia liniară

- Se poate constata că ecuaţia neuronului elementar liniar cu o singură intrare este cea a unei drepte:



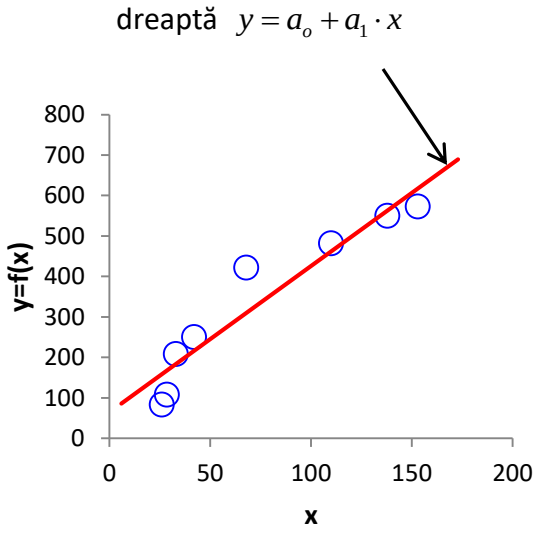
$$y = F\left(b + \sum_{i=1}^{n} w_i \cdot x_i\right)$$

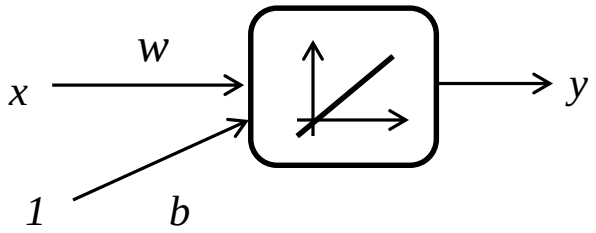
$$y = b + w \cdot x$$

- Pentru a determina ponderea şi pragul optim al neuronului, putem aplica regresia liniară cu o singură variabilă.

Neuronul cu o intrare şi regresia liniară

- Trebuie identificate valorile optime ale coeficienţilor w şi b astfel încât neuronul să facă corespondenţa $x \rightarrow y(x)$ cu eroare minimă (E) pentru toate modelele de intrare cunoscute.





$$y = b + w \cdot x$$

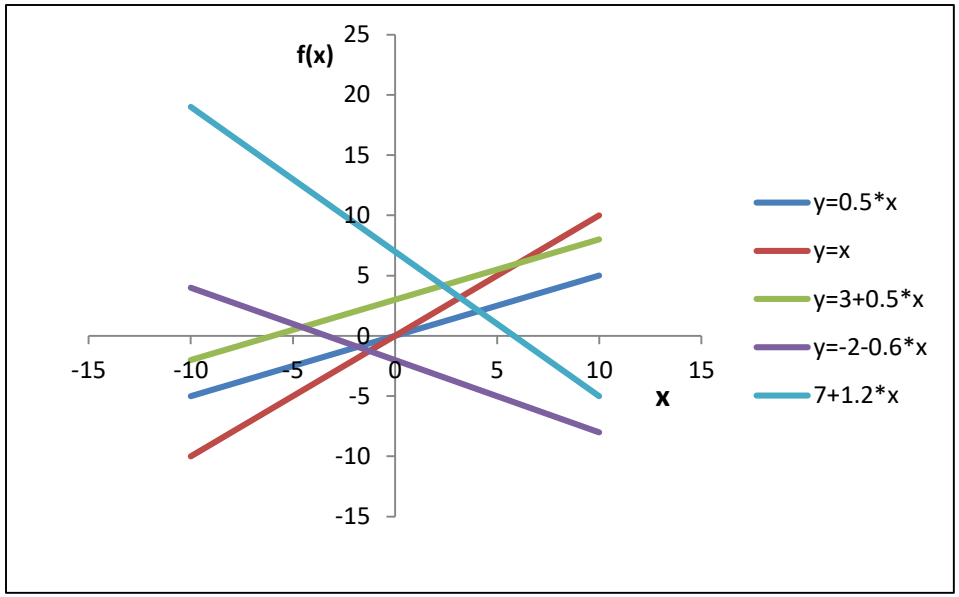
- este o problemă de optimizare: $\min_{w,b} E(w,b)$

Matematică – funcția liniară

- Funcţia se numeşte liniară pentru că graficul ei este o linie dreaptă
- expresia analitică generală pentru o singură variabilă:

$$f(x) = a_0 + a_1 \cdot x$$

- Variind coeficienţii a_0 şi a_1, se modifică graficul funcţiei
- Dreptele cu $a_0=0$ trec prin origine

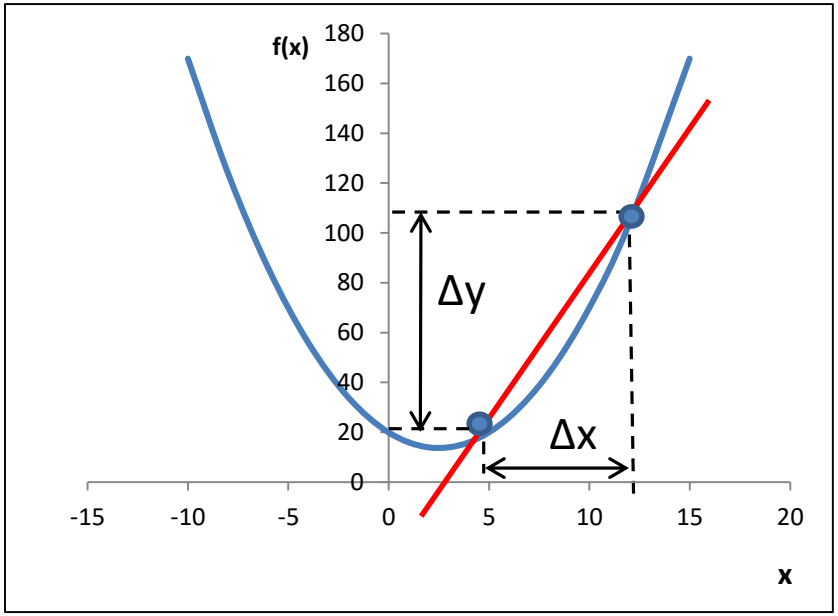


Graficele unor funcţii liniare de o variabilă pentru x în intervalul [-10, 10]

Matematică – panta unei drepte

- Panta dreptei – exprimă cu cât creşte y atunci când x creşte cu o unitate

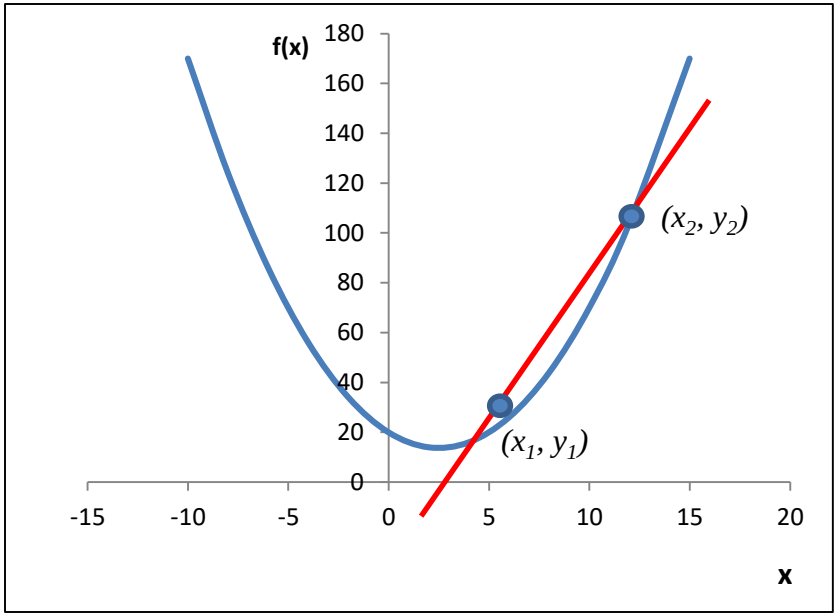
$$m = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{y_2 - y_1}{x_2 - x_1} = \frac{\Delta y}{\Delta x}$$



- este pozitivă (m>0) când x şi y cresc sau scad simultan
- este negativă: (m<0) când x scade , y creşte / când x creşte, y scade
- este 0 pentru o dreaptă orizontală
- este nedefinită pentru o dreaptă verticală

Matematică – panta unei drepte

- Cunoscând $dx=x_2 - x_1$, y_2 şi panta, se poate afla y_1.
- Cunoscând x_1,x_2, y_1 şi y_2, se poate afla panta.

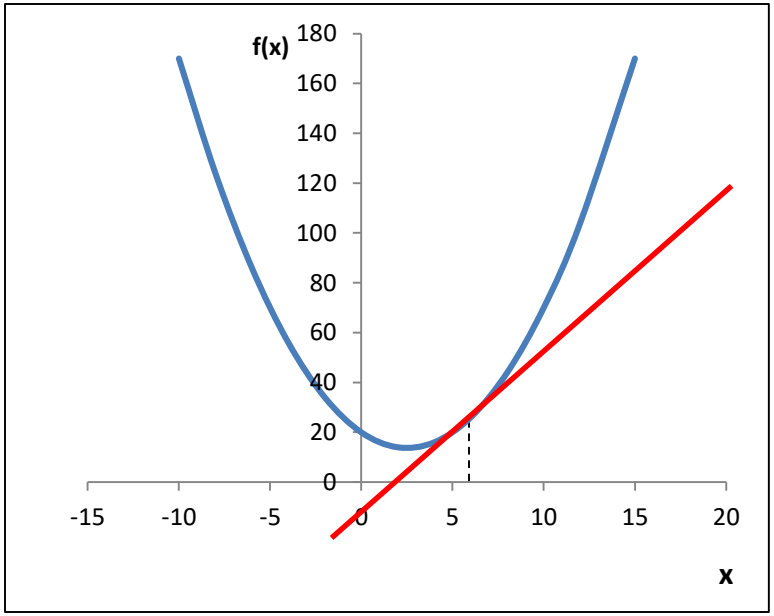


Matematică – derivate

- Derivata unei funcţii în punctul x este tangenta la curba funcţiei în punctul respectiv

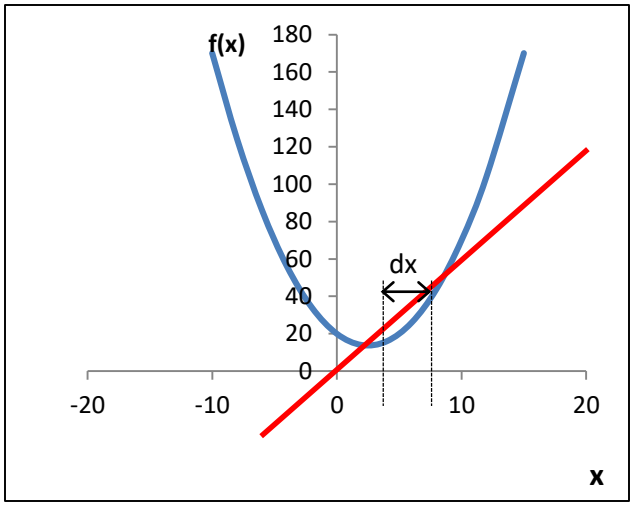
$$f'(x) = \lim_{x_2 - x_1 \to 0} \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{dy}{dx}$$

  - d înseamnă variaţie foarte mică



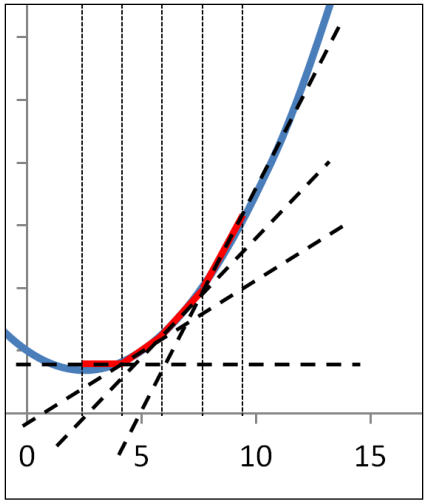
Matematică – linearizarea

- Când intervalul dx este foarte mic, eroarea de aproximare a curbei cu o dreaptă este acceptabilă – linearizare



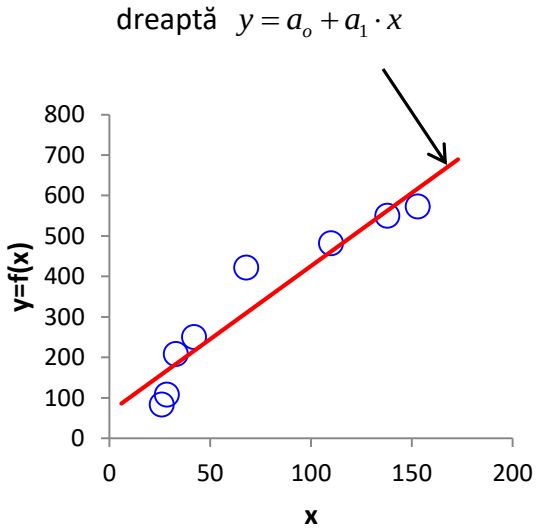
- Linearizarea simplifică calculele

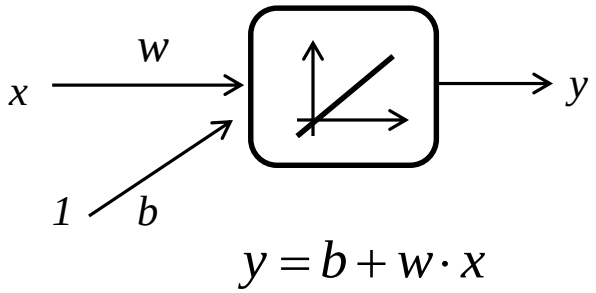
- Panta tangentei se reduce treptat, tinzând la 0 atunci când y tinde la minim



Neuronul cu o intrare şi regresia liniară

- Trebuie identificate valorile optime ale coeficienţilor w şi b astfel încât neuronul să facă corespondenţa $x \rightarrow y(x)$ cu eroare minimă (E) pentru toate modelele de intrare cunoscute.

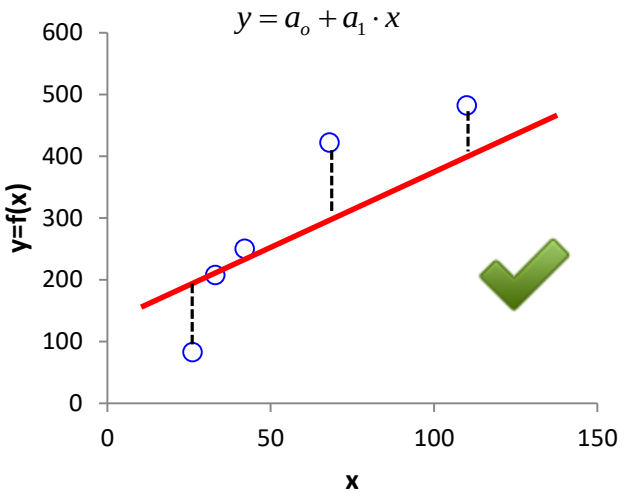


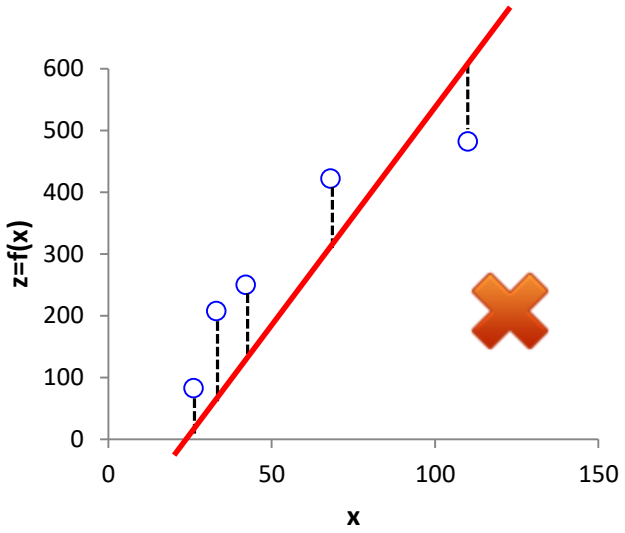


- este o problemă de optimizare: $\min_{w,b} E(w,b)$ - "găseşte acele valori ale lui w şi b pentru care E are valoarea cea mai mică (minimă)"

Eroarea regresiei liniare

- Suma distanţelor dintre punctele – modele de intrare (valorile dorite d) şi dreapta de regresie (valorile obţinute o) trebuie să fie minimă.





$$E = \sum_{m=1}^{M} \left\| d^{(m)} - o^{(m)} \right\|^2 = \sum_{m=1}^{M} \left(d^{(m)} - o^{(m)} \right)^2$$

Abaterea pătratică totală (cost)

M - numărul total de modele din setul de antrenare

‖ … ‖ - norma euclideană (distanţa)

Eroarea folosită în calcule

- Pentru simplificarea calculelor ulterioare, se preferă formula

$$E = \frac{1}{2} \cdot \sum_{m=1}^{M} \left(d^{(m)} - o^{(m)} \right)^2$$

- Se mai poate folosi abaterea pătratică medie totală

$$E = \frac{1}{2 \cdot M} \cdot \sum_{m=1}^{M} \left(d^{(m)} - o^{(m)} \right)^2$$

- M este un coeficient scalar, deci cele două formule au acelaşi efect, diferă doar valorile numerice obţinute.

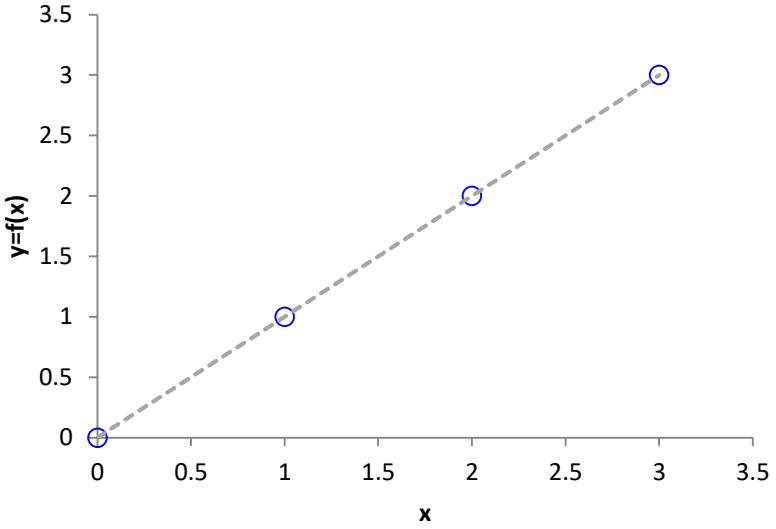
Reprezentarea grafică a erorii E

- Se consideră o problemă de regresie liniară cu o singură variabilă, simplificată prin renunţarea temporară la coeficientul a_0, $y=a\cdot x$

- Dreptele de tipul $y=a\cdot x$ trec prin origine

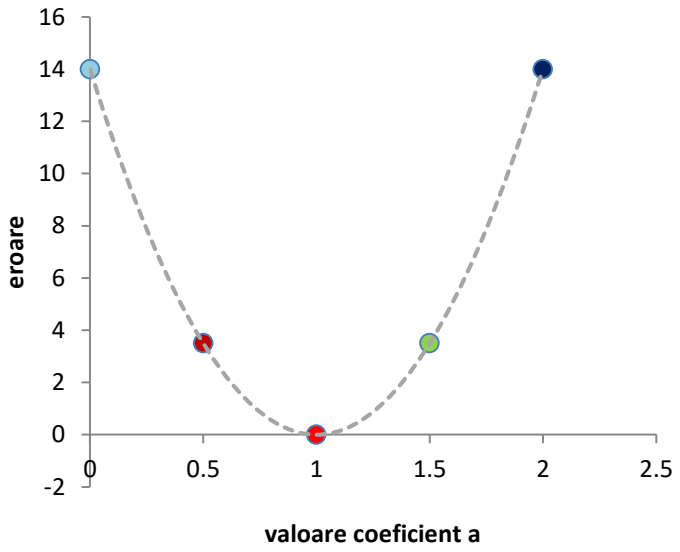
- Se presupune că se cunosc perechi de intrare-ieşire care modelează exact funcţia $y=x$

- Dreapta de regresie cu eroare minimă ar fi chiar $y=x$.



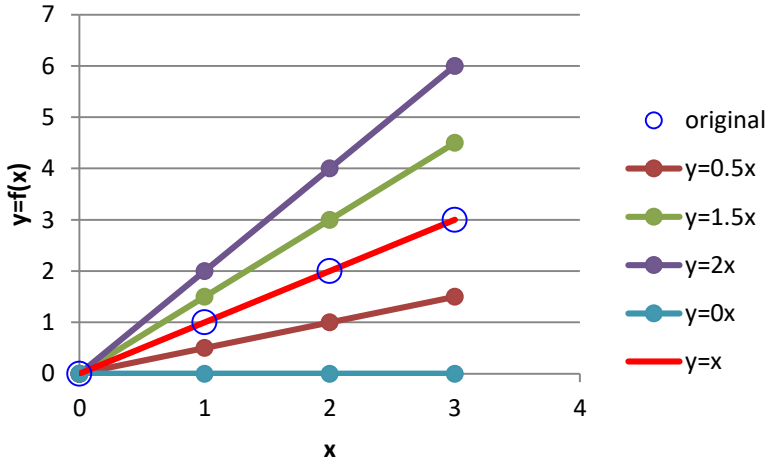
Reprezentarea grafică a erorii E

- Dacă se calculează eroarea pentru diverse valori ale coeficientului a, adică diverse pante ale dreptei de regresie $y=a\cdot x$ şi se trasează graficul funcţiei $E=f(a)$, se obţine o parabolă.
- Grafic 2-D
- În grafic e reprezentat E/4, abaterea medie pătratică totală



Graficul erorii pentru regresia liniară y=a·x

		Valori obţinute pentru drepte de regresie y=a·x				
x	valori dorite (y=1·x)	y=1·x a=1	y=0.5·x a=0.5	y=1.5·x a=1.5	y=2·x a=2	y=0·x a=0
0	0	0	0	0	0	0
1	1	1	0.5	1.5	2	0
2	2	2	1	3	4	0
3	3	3	1.5	4.5	6	0
Eroare:		0	3.5	3.5	14	14



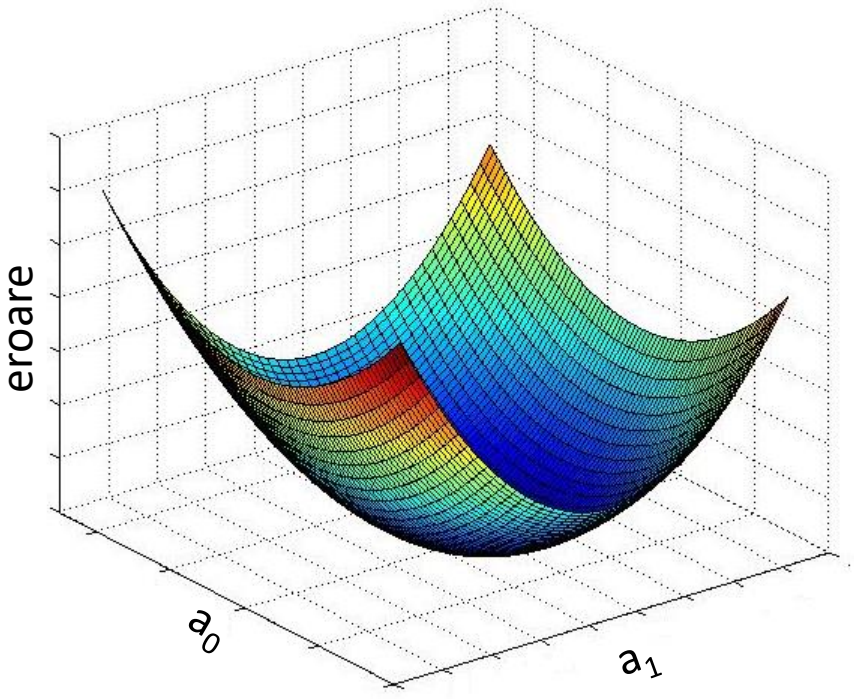
Calculul erorii şi reprezentare grafică pentru diverse drepte de regresie y=a·x

Reprezentarea grafică a erorii E

- Pentru regresia $y=a_0+a_1 \cdot x$, funcţia $E=f(a_1,a_2)$ va fi o suprafaţă 3-D, alcătuită din mai multe parabole, câte una pentru fiecare valoare a lui a_0.

- Un punct pe această suprafaţă va reprezenta un set de coordonate (a_0,a_1), care definesc o dreaptă de regresie $y=a_0+a_1 \cdot x$

- Suprafaţa erorii este convexă.



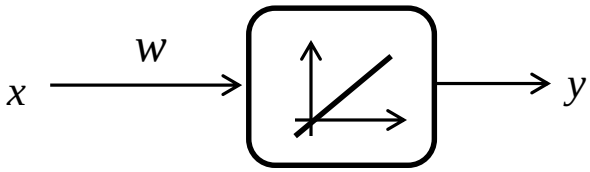
Graficul erorii pentru regresia liniară $y=a_0+a_1 \cdot x$

Recapitulare

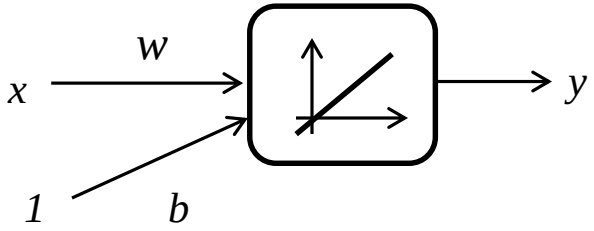
- $y=a_1 \cdot x$ sau $y=w \cdot x$ modelează neuronul cu o singură intrare fără prag
- Valorile erorii pentru toţi w posibili într-un interval stabilit o parabolă (de obicei nu are minimul în $y=0$), deoarece în cele mai multe cazuri o dreaptă nu poate aproxima perfect datele de intrare

- $y=a_0+a_1 \cdot x$ sau $y=b+w \cdot x$ modelează neuronul cu o singură intrare şi prag
- Valorile erorii pentru toţi b şi w posibili formează o parabolă cu minimul în acele valori w şi b pentru care eroarea de regresie este minimă

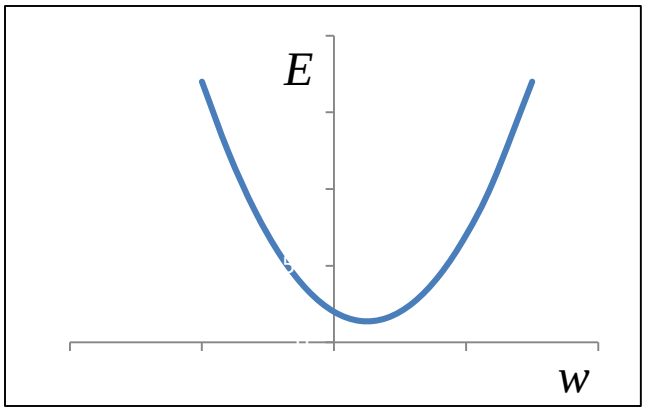
Recapitulare

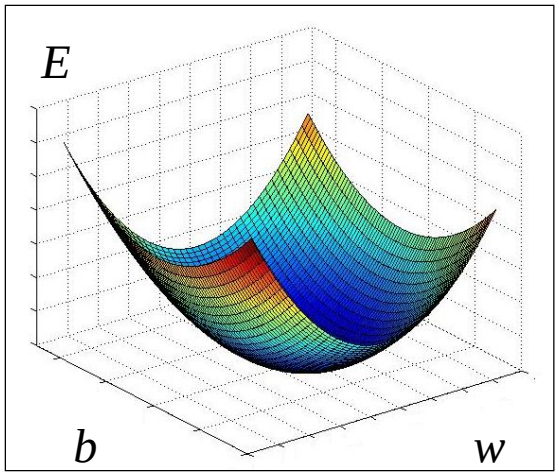


$y = w \cdot x$



$y = b + w \cdot x = w_0 + w_1 \cdot x$

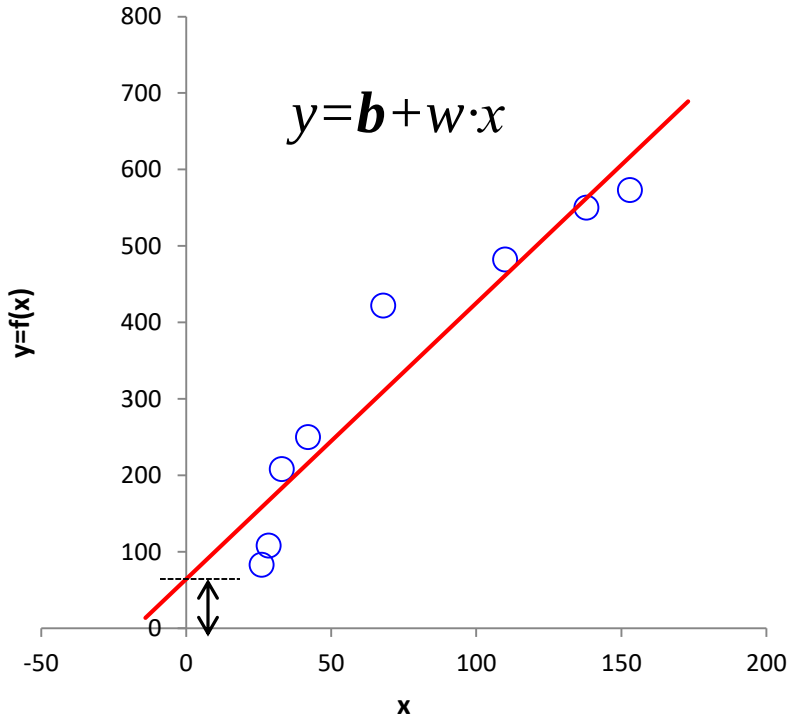




Variaţia erorii regresiei liniare: $E = \frac{1}{2 \cdot M} \cdot \sum_{m=1}^{M} \left(d^{(m)} - y^{(m)} \right)^2$

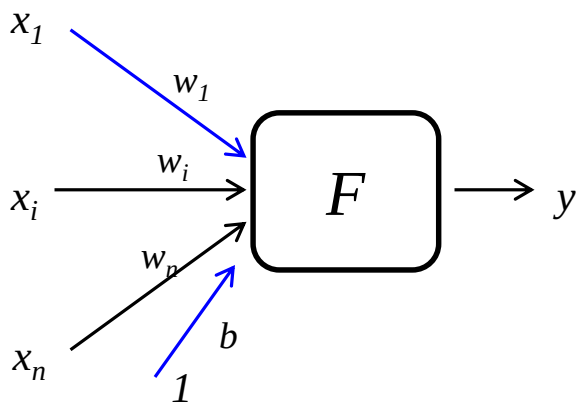
Avem nevoie de prag !

- Dreapta optimă de regresie nu trece întotdeauna prin origine !



Regresia liniară multiplă

- Neuronul cu maxim o intrare şi un prag este un caz particular al neuronului cu n intrări şi un prag:



- Regresia liniară cu o singură variabilă este un caz particular al regresiei liniare multiple.
- Cu un singur neuron liniar, putem rezolva doar probleme de regresie liniară - un singur neuron este limitat.

Regresia liniară multiplă

- Regresia cu o singură variabilă:

$$y = b + w \cdot x$$

- Regresia multiplă (RM)

$$y = b + w_1 \cdot x_1 + \ldots + w_i \cdot x_i + \ldots + w_n \cdot x_n =$$

$$= w_0 \cdot 1 + w_1 \cdot x_1 + \ldots + w_i \cdot x_i + \ldots + w_n \cdot x_n = \sum_{i=0}^{n} w_i \cdot x_i,\ x_0 = 1$$

- RM scrisă vectorial

$$y = \begin{bmatrix} x_0 \\ \ldots \\ x_i \\ \ldots \\ x_n \end{bmatrix} \cdot \begin{bmatrix} w_1 & \ldots & w_i & \ldots & w_n \end{bmatrix} = [X] \cdot [W]^T$$

Regresia liniară multiplă

Nr.	Titular contract	Persoane	Suprafaţă locuinţă [mp]	Studii	Venit lunar [lei]	Consum mediu [kWh /lună]	Putere instalată [kW]
			x_1		x_2	y	x_3
1	Nicolae Popa	3	110	superioare	3350	482	10
2	Vasile Ionescu	5	33	liceu	3100	208	8
3	Maria Popovici	2	42	superioare	5480	250	15
4	Adriana Elisei	3	138	superioare	4450	550	20
-	...	...	...	...	...	...	...
7	Paul Irimciuc	4	28.5	liceu	2500	108	6
9	Florin Mihalcea	2	153	doctorat	6600	573	12

- *consum=f(suprafaţă, prag), y=f(x,b)* - regresie cu o variabilă
  consum=f(suprafaţă,, venit, putere instalată, prag), y=f(x_1, x_2, x_3, b)– regresie multiplă
  - În multe cazuri, regresia cu o singură variabilă nu este suficientă pentru a obţine rezultate optime.
  - Însă prea multe variabile pot cauza specializarea.
  - Este necesar un compromis, trebuie făcute încercări preliminare.

Antrenarea unui neuron

- Este necesară o metodă generală, care să realizeze minimizarea erorii pentru orice problemă de regresie liniară pentru care se cunosc perechi de intrare –ieşire *([X]-y)*

- Problema $\min_{w,b}(E)$, unde $$E = \frac{1}{2} \cdot \sum_{m=1}^{M} \left(d^{(m)} - o^{(m)} \right)^2$$

poate fi rezolvată pe două căi:

- Analitic
- Printr-o metodă de aproximaţii succesive, iterativă

Rezolvarea analitică

problema:

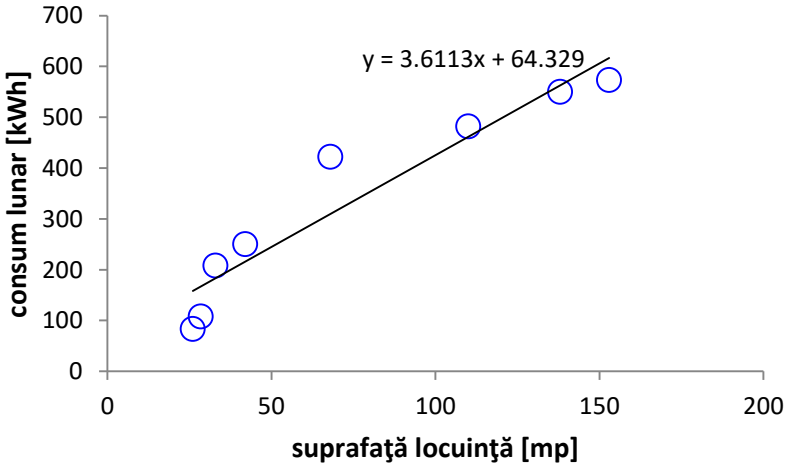
Model	Suprafaţă locuinţă [mp]	Consum mediu [kWh /lună]
	x	y
$(x^{(1)},y^{(1)})$	110	482
$(x^{(2)},y^{(2)})$	33	208
$(x^{(3)},y^{(3)})$	42	250
$(x^{(4)},y^{(4)})$	138	550
$(x^{(5)},y^{(5)})$	68	422
$(x^{(6)},y^{(6)})$	26	83
$(x^{(7)},y^{(7)})$	28.5	108
$(x^{(8)},y^{(8)})$	153	573

prag x

$$X = \begin{bmatrix} 1 & 110 \\ 1 & 33 \\ 1 & 42 \\ 1 & 138 \\ 1 & 68 \\ 1 & 26 \\ 1 & 28.5 \\ 1 & 153 \end{bmatrix} \qquad y = \begin{bmatrix} 482 \\ 208 \\ 250 \\ 550 \\ 422 \\ 83 \\ 108 \\ 573 \end{bmatrix}$$

$$[b \quad w] = inv(X \cdot X^T) \cdot X^T \cdot y$$

$$b = 64.3286,\ w = 3.6113$$



verificare în Excel

- Metoda funcţionează pentru oricâte variabile de intrare (creşte numărul de coloane din matricea $[X]$ şi numărul de w calculaţi).

Rezolvarea analitică

- Problemele rezolvate de RNA au de obicei un număr uriaş de modele de intrare (mii, milioane de linii în matricea X) şi mulţi parametri de intrare (coloane în matricea X).

- Metoda analitică necesită inversarea unei matrice.

- Inversarea matricelor este un efort de calcul mare chiar şi pentru calculatoarele de azi.

- De obicei, se preferă o metodă iterativă.

Metode iterative

- Pleacă de la o aproximaţie iniţială a soluţiei căutate (foarte probabil greşită)
- Corectează în mod repetat aproximaţia iniţială, în mai mulţi paşi sau iteraţii, după o formulă sau un set de formule matematice
- Ultima aproximaţie calculată este soluţia căutată

- Criteriul de oprire poate fi:
  - S-a efectuat un număr prestabilit de iteraţii
  - Corecţia nu mai aduce îmbunătăţiri relevante soluţiei găsite.

- Exemplu: metodele Seidel-Gauss sau Newton-Raphson de calcul al regimului permanent al reţelelor electrice.

Algoritmul general al unei metode iterative:

- Stabileşte aproximaţia iniţială a soluţiei: $s=s^{(0)}$,
- Porneşte contor iteraţii $t=0$
- repetă
  - Incrementează contor iteraţii $t=t+1$
  - Calculează corecţia soluţiei: Δs
  - Corectează soluţia cunoscută în iteraţia curentă $s^{(t+1)}=s^{(t)}+\Delta s$
  - Verifică criteriul de oprire
- Dacă s-a îndeplinit criteriul de oprire, soluţia căutată este $s^{(t+1)}$

Metode iterative

- O metodă iterativă va calcula de cele mai multe ori o valoare aproximativă, nu pe cea exactă a soluţiei căutate.

- Metoda este
  - Convergentă – dacă soluţia calculată tinde spre soluţia căutată
  - Divergentă – dacă soluţia calculată se îndepărtează de soluţia căutată sau oscilează în jurul ei.
  - Oscilantă – dacă soluţia calculată se apropie şi se îndepărtează de soluţia exactă, fără a avea o direcţie clară

Metodă iterativă generală pentru un neuron

- Ipoteză: neuron liniar cu o intrare şi un prag, $y = b + w \cdot x$
- Problema de rezolvat: găseşte valorile optime ale w şi b astfel încât E să fie minim:

$$\min_{w,b} E(w,b)$$

- Pragul b este tratat ca ponderea unei intrări cu valoarea 1, i se aplică aceeaşi formulă de corecţie ca şi ponderilor.

- Iniţializează w şi b cu valori aleatorii, mici, $w^{(0)}$ şi $b^{(0)}$
- Iniţializează contor iteraţii $t=0$.
- Repetă:
  - Incrementează contor iteraţii
  - Calculează corecţii ponderi Δw şi Δb
  - Corectează w şi b:
    - $w^{(t+1)} + w^{(t)} + \Delta w$
    - $b^{(t+1)} + b^{(t)} + \Delta b$
  - Calculează eroarea obţinută cu noii $w^{(t+1)}$ şi $b^{(t+1)}$
  - Verifică condiţia de oprire
- Dacă s-a îndeplinit condiţia de oprire, w şi b sunt (aproximativ) cei căutaţi.

Metodă iterativă generală pentru un neuron

- Dacă eroarea E scade, înseamnă că w şi b se apropie de valorile care definesc dreapta optimă de regresie şi metoda este convergentă.

- Metoda trebuie să funcţioneze pentru oricâte ponderi w, adică neuroni cu oricâte intrări, adică modele de intrare cu oricâţi parametri.

Metoda gradientului

- Cea mai cunoscută şi simplă metodă iterativă de antrenare pentru neuroni şi reţele neuronale este metoda gradientului sau metoda descreşterii gradientului (gradient descent)

- Pentru minimizarea funcţiei eroare E în raport cu ponderile w_0 (pragul b), $w_1, \ldots w_i, \ldots w_n$, $\min_{\substack{w_0, w_1, \ldots, \\ w_i, \ldots, w_n}} E(w_0, w_1, \ldots, w_i, \ldots, w_n)$

- În fiecare iteraţie t, toate ponderile w_i se corectează cu următoarea formulă:

$$w_i^{(t+1)} = w_i^{(t)} - \eta \cdot \frac{\partial E}{\partial w_i^{(t)}}\bigg|_{w=w_i^{(t)}}, \quad i = 0 \ldots n$$

- η – rată de învăţare, număr subunitar, 0.1 - 0.2.

Metoda gradientului

- Formula erorii $$E = \frac{1}{2} \cdot \sum_{m=1}^{M} \left(d^{(m)} - o^{(m)} \right)^2$$

$o^{(m)}$ este valoarea obţinută pentru modelul de intrare m, $(x_0, x_1, \ldots x_i, \ldots x_n)^{(m)}$, cu funcţia de regresie definită de coeficenţii $w_0, w_1, \ldots w_i, \ldots w_n$, adică ieşirea neuronului, deci eroarea depinde de ieşirea neuronului

$$o^{(m,t)} = w_0^{(t)} + w_1^{(t)} x_1^{(m)} + \ldots + w_i^{(t)} x_i^{(m)} + \ldots + w_n^{(t)} x_n^{(m)}$$

- Folosind regula derivării în lanţ, eroarea se poate scrie:

$$\left. \frac{\partial E}{\partial w_i} \right|_{w=w_i^{(t)}} = \frac{1}{2} \cdot \sum_{m=1}^{M} \frac{\partial o^{(m)}}{\partial w_i} \cdot \left. \frac{\partial E}{\partial o^{(m)}} \right|_{w=w_i^{(t)}} = -\sum_{m=1}^{M} x_i^{(m)} \cdot (d^{(m)} - o^{(m)})$$

$$w_i^{(t+1)} = w_i^{(t)} - \eta \cdot \left. \frac{\partial E}{\partial w_i^{(t)}} \right|_{w=w_i^{(t)}} = w_i^{(t)} + \eta \cdot \sum_{m=1}^{M} x_i^{(m)} \cdot (d^{(m)} - o^{(m,t)}), \quad i = 0 \ldots n$$

Suma derivatelor este egală cu derivata sumei şi viceversa

$$\frac{\partial E}{\partial w_i}\Big|_{w=w_i^{(t)}} = \frac{\partial}{\partial w_i}\left(\frac{1}{2}\cdot\sum_{m=1}^{M}\left(d^{(m)}-o^{(m)}\right)^2\right) = \frac{1}{2}\cdot\sum_{m=1}^{M}\frac{\partial}{\partial w_i}\left(d^{(m)}-o^{(m)}\right)^2$$

$$= \frac{1}{2}\cdot\sum_{m=1}^{M}\frac{\partial o^{(m)}}{\partial w_i}\cdot\frac{\partial E}{\partial o^{(m)}}\Big|_{w=w_i^{(t)}} = -\sum_{m=1}^{M}x_i^{(m)}\cdot\left(d^{(m)}-o^{(m)}\right)$$

$$o^{(m,t)} = w_0^{(t)} + w_1^{(t)}x_1^{(m)} + \ldots + w_i^{(t)}x_i^{(m)} + \ldots + w_n^{(t)}x_n^{(m)}$$

- derivata parţială faţă de w_i este derivata funcţiei atunci când restul w sunt constante;
- derivata unei constante e 0,
- $d(w\cdot x)/dw = x$

$$\frac{\partial o^{(m)}}{\partial w_i}\Big|_{w=w_i^{(t)}} = x_i^{(m)}$$

$$\frac{\partial E}{\partial o^{(m)}}\Big|_{w=w_i^{(t)}} = \frac{d}{do^{(m)}}\left(\left(d^{(m)}-o^{(m)}\right)^2\right)$$

$$(u')^n = n\cdot(u)^{n-1}\cdot u'$$

(regula derivării funcţiilor compuse)

$$\frac{\partial E}{\partial o^{(m)}}\Big|_{w=w_i^{(t)}} = 2\cdot\left(d^{(m)}-o^{(m)}\right)\cdot(-1)$$
$$= -2\cdot\left(d^{(m)}-o^{(m)}\right)$$

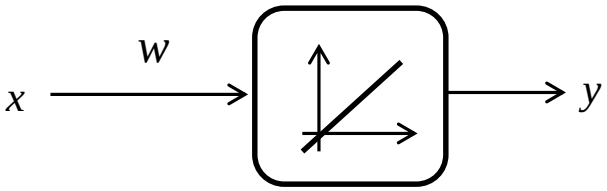
Metoda gradientului:

- Forma algoritmului pentru regresia multiplă:

- Iniţializează aleatoriu $w_0, w_1, \ldots w_i, \ldots w_n$
- Iniţializează contor iteraţii t=0
- Repetă
  - Incrementează iteraţie: t=t+1
  - calculează corecţii ponderi $\Delta w_i^{(t+1)} - \eta \cdot \frac{\partial E}{\partial w_i^{(t)}}|_{w_i}, \quad i = 0 \ldots n$
  - Corectează ponderi: $w_0^{(t+1)} = w_i^{(t)} + \Delta w_i^{(t+1)}, \quad i = 0 \ldots n$
  - Recalculează eroarea E
  - Verifică criteriul de oprire (nr. iteraţii sau progres eroare)
- Dacă s-a îndeplinit condiţia de oprire, antrenarea s-a încheiat.

Interpretare grafică

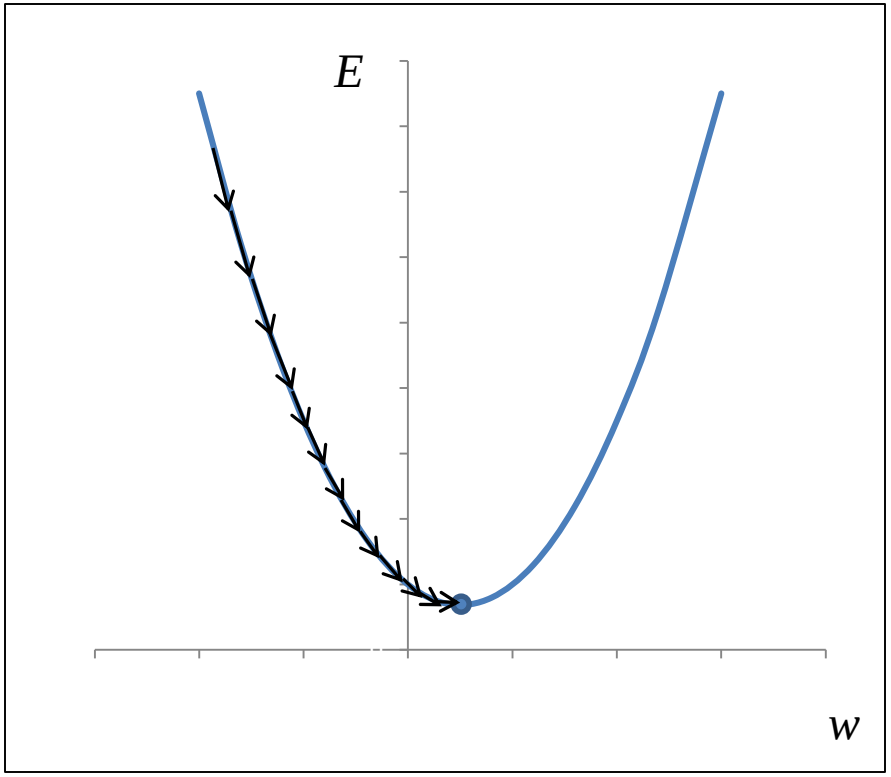
- Regresie cu o variabilă fără prag



$$y = w \cdot x$$

$$w^{(t+1)} = w^{(t)} - \eta \cdot \frac{\partial E}{\partial w^{(t)}}\bigg|_{w=w_i^{(t)}}$$

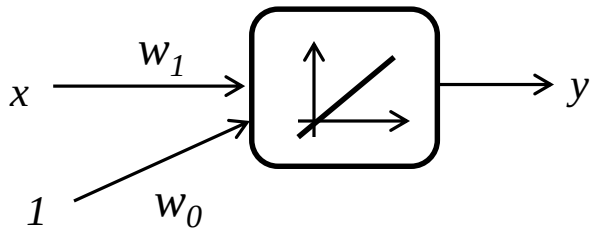
- Pe măsură ce eroarea se apropie de minim, corecţiile scad
- Algoritmul converge întotdeauna către minimul global, dacă η este ales corect



- Paşii sunt mai mici dacă η este mai mic

Interpretare grafică

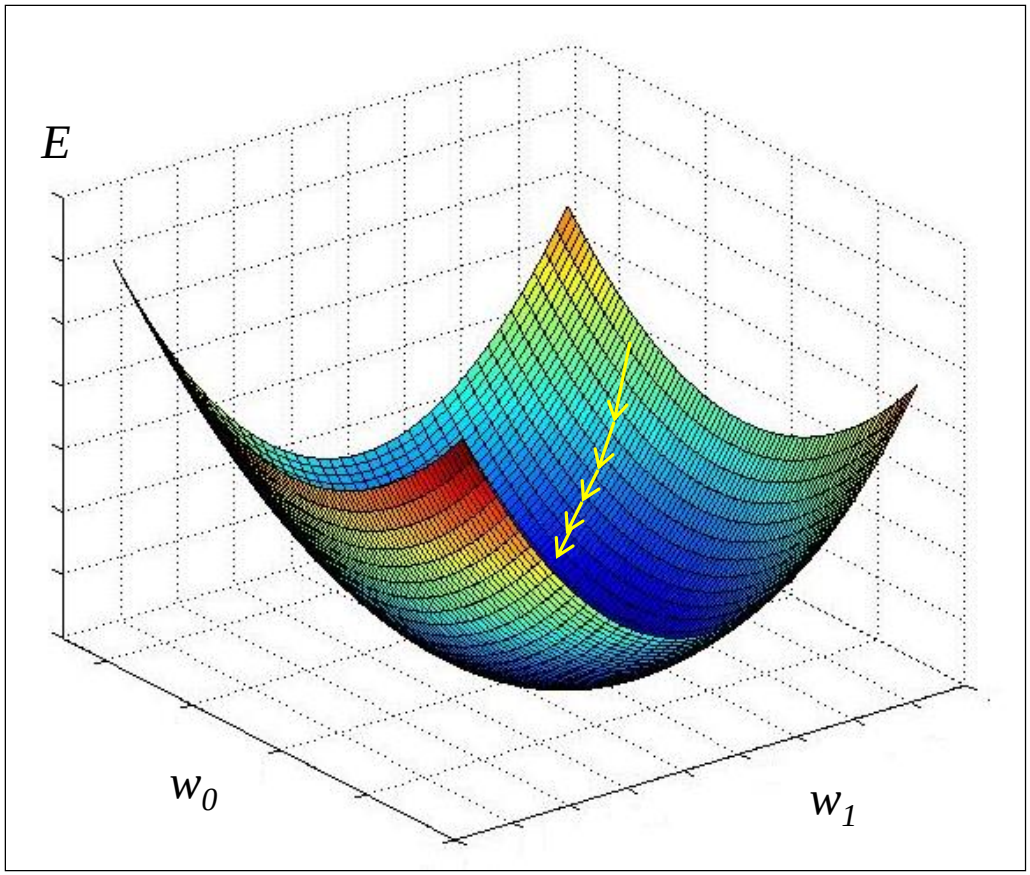
- Regresie cu o variabilă şi prag



$$y = b + w \cdot x = w_0 + w_1 \cdot x$$

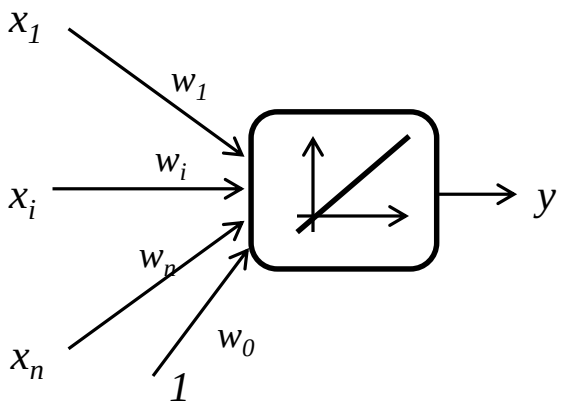
$$w^{(t+1)} = w^{(t)} - \eta \cdot \frac{\partial E}{\partial w^{(t)}}\bigg|_{w=w_i^{(t)}}$$

- Suprafaţa erorii e convexă şi algoritmul converge întotdeauna spre minimul global, dacă η este ales corect

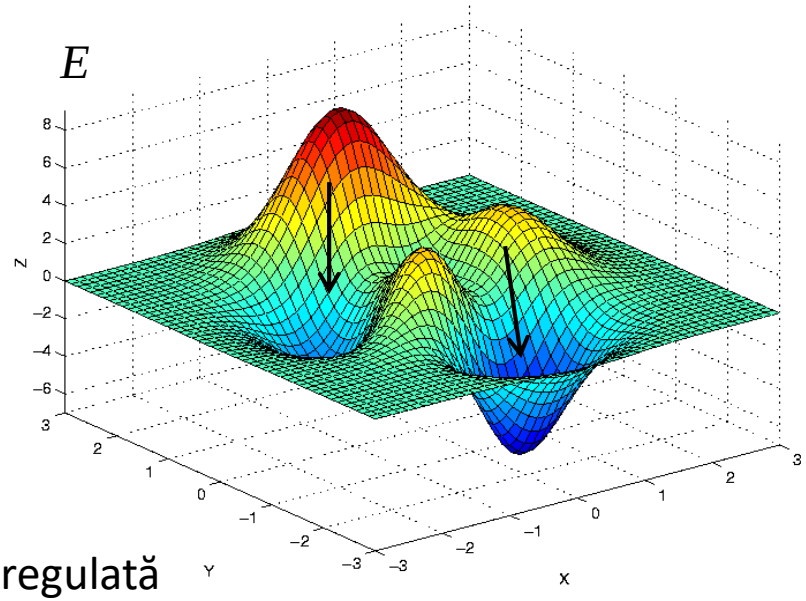


Interpretare grafică

- Regresie multiplă



$$y = w_0 + w_1 \cdot x_1 + \ldots + w_n \cdot x_n$$

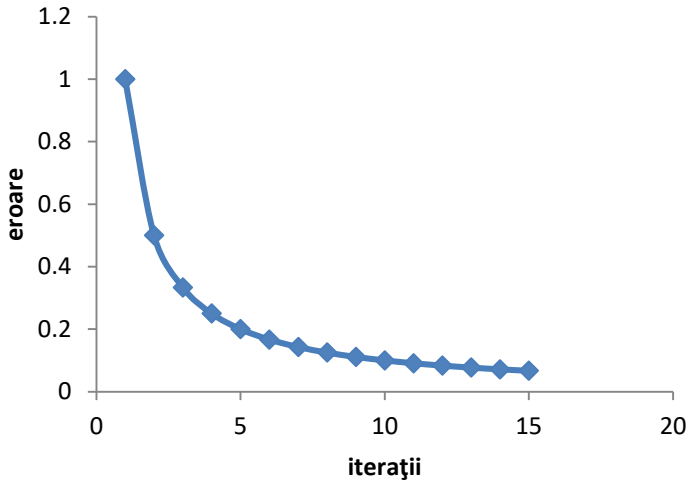


- Suprafaţa erorii nu mai e convexă, ci neregulată

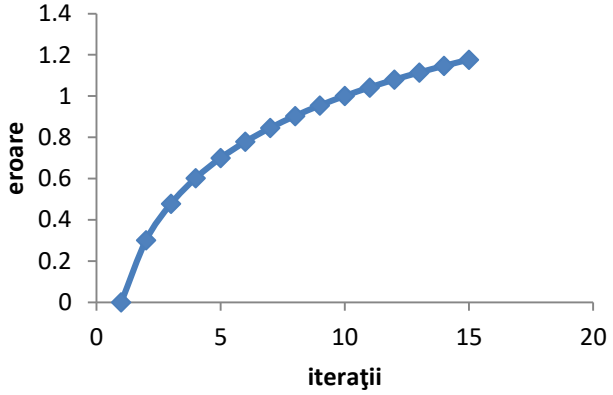
- În funcţie de punctul de iniţializare al ponderilor, eroarea se poate opri într-un minim local.

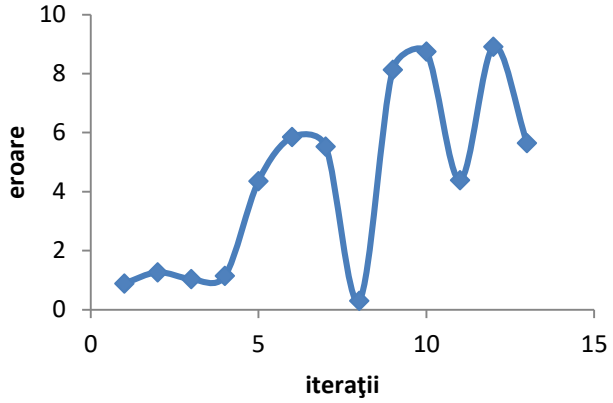
Convergența algoritmului gradientului

- Convergenţă: eroarea scade neliniar



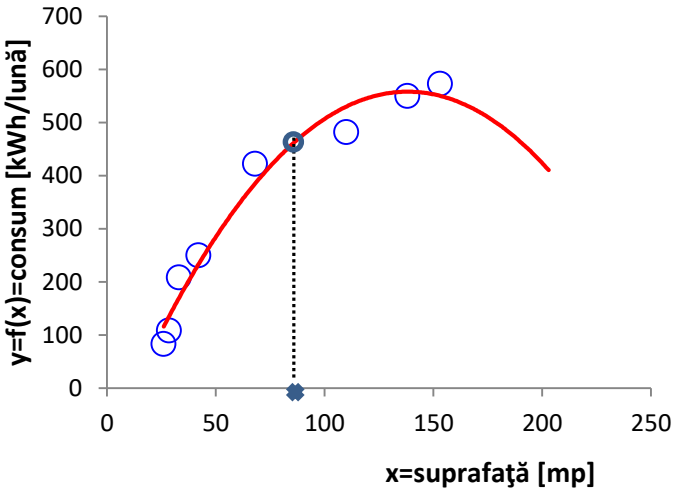
- Divergenţă / comportament oscilant: eroarea creşte sau oscilează



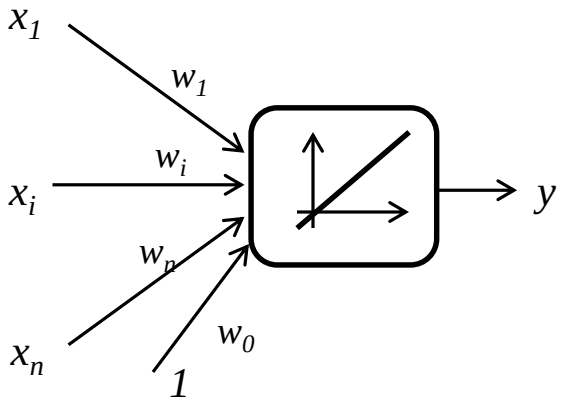


Recapitulare

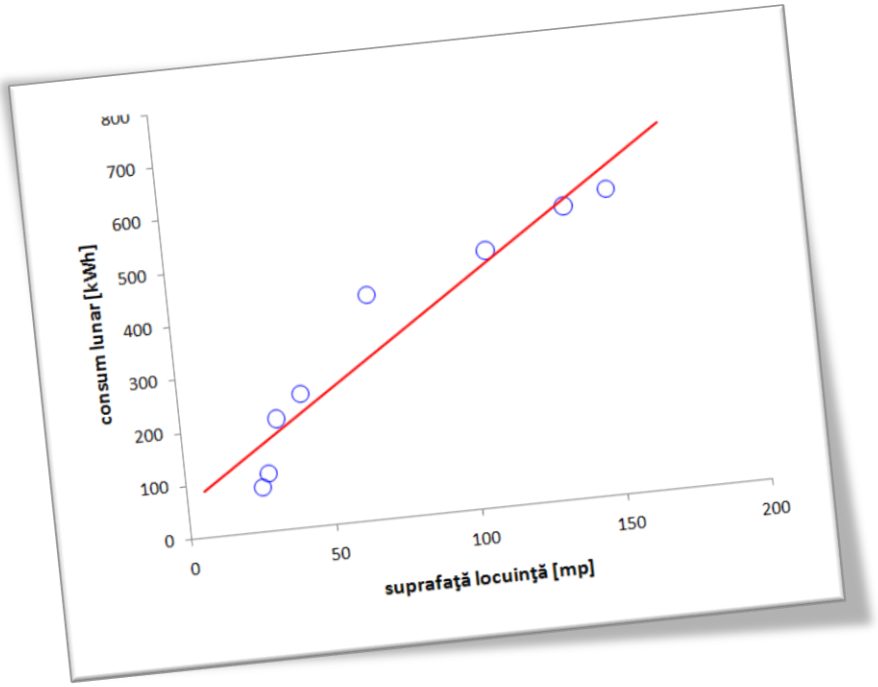
- Prin regresie, se pot aproxima valorile pe care o funcţie le poate lua într-un interval, fără a cunoaşte expresia funcţiei, ci doar eşantioane discrete.



- Neuronul elementar cu funcţie de activare liniară poate fi antrenat pentru probleme de regresie liniară folosind algoritmul gradientului.



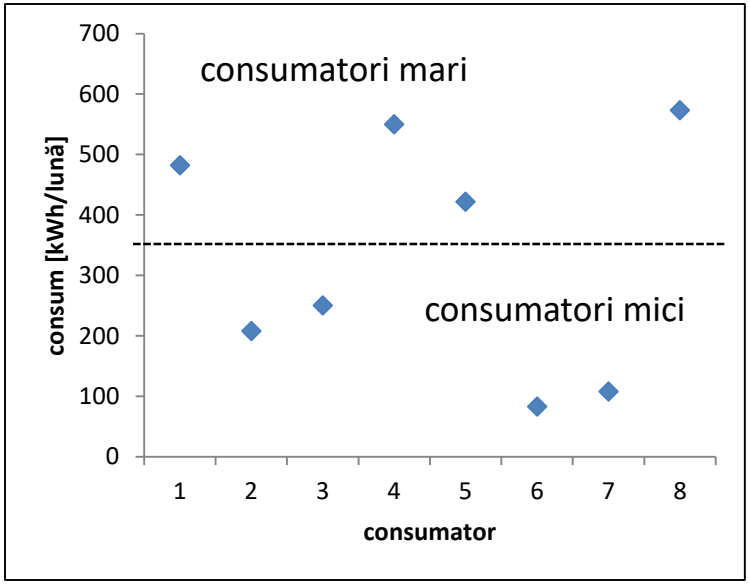
$$y = w_0 + w_1 \cdot x_1 + \ldots + w_n \cdot x_n$$



ÎNVĂȚAREA SUPRAVEGHEATĂ
Probleme de clasificare. Neuronul elementar cu funcție de activare sigmoid logistic

Clasificare

- Gruparea datelor pe categorii (clase)
  - Clienţi cu consum >350 kWh/lună – mari – clasa 1 (adevărat)
  - Clienţi cu consum <=350 kWh/lună – mici – clasa 0 (fals)

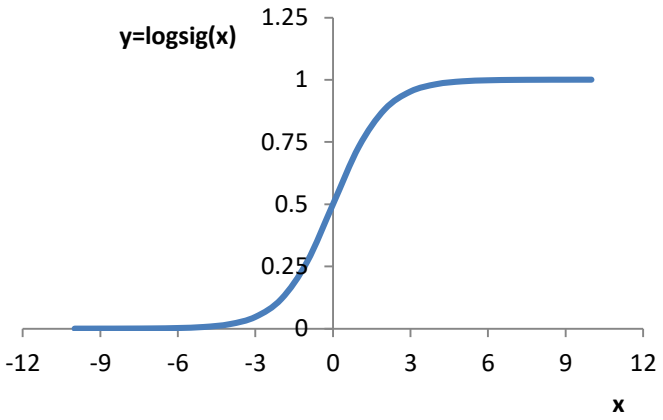


Nr.	Titular contract	Consum mediu [kWh/lună]	Clasă
1	Nicolae Popa	482	Mare
2	Vasile Ionescu	208	Mic
3	Maria Popovici	250	Mic
4	Adriana Elisei	550	Mare
5	Gabriela Dăscălescu	422	Mare
6	Aurel Degeratu	83	Mic
7	Paul Irimciuc	108	Mic
9	Florin Mihalcea	573	Mare

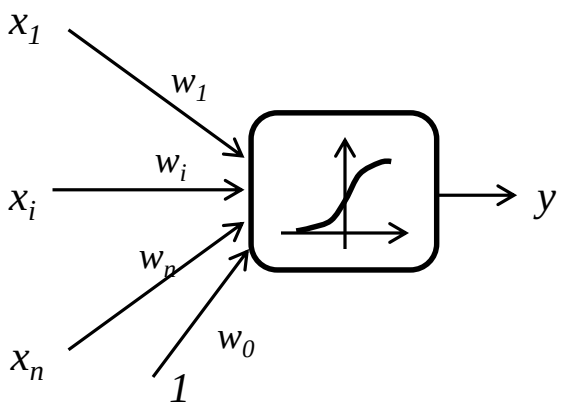
Neuronul logistic

- Funcţia sigmoid logistic:

$$y = \frac{1}{1+e^{-x}} = logsig(x)$$



- Neuronul logistic:



Intrarea netă şi ieşirea neuronului logistic:

$$net = w_1 \cdot x_1 + \ldots + w_i \cdot x_i + \ldots + w_n \cdot x_n$$

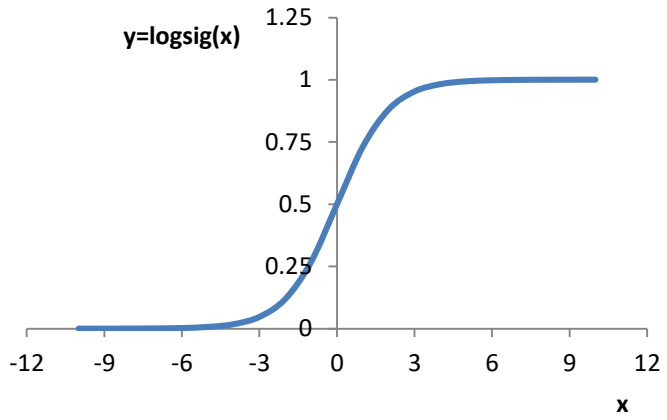
$$y = \frac{1}{1+e^{-net}}$$

Funcția sigmoid logistic

- Funcţia sigmoid logistic:

$$y = \frac{1}{1+e^{-x}} = logsig(x)$$

  - ia valori între 0 şi 1
  - se intersectează cu axa $y=0$ la $x=0.5$

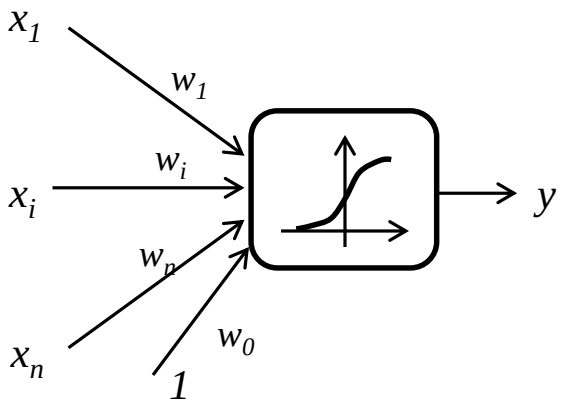


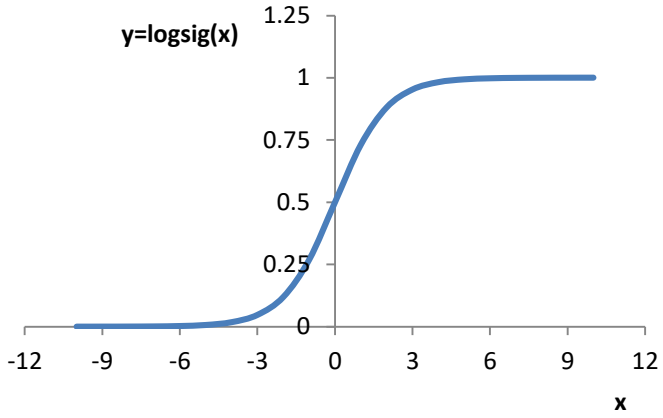
- Într-o problemă de clasificare de tip 0/1, valoarea funcţiei sigmoid logistic este probabilitatea ca modelul de intrare X să aparţină clasei 1.
  - Când $y>=0.5$, x aparţine clasei 1 (sau, în general, aparţine unei anumite clase)
  - Când $y<0.5$, x aparţine clasei 0 (nu aparţine unei anumite clase)

  - $logsig\ (x)>0.5$ atunci când $x>0$
  - $logsig\ (x)<=0.5$ atunci când $x<=0$

Funcționarea neuronului logistic

- Neuronul logistic

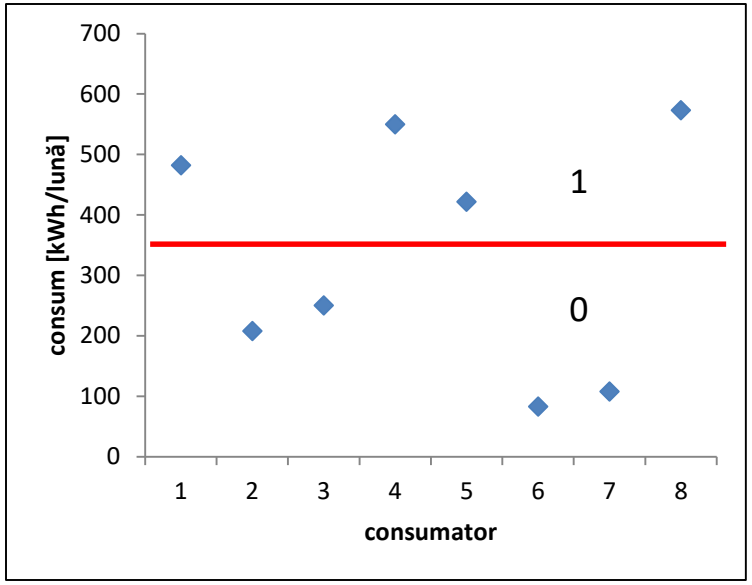


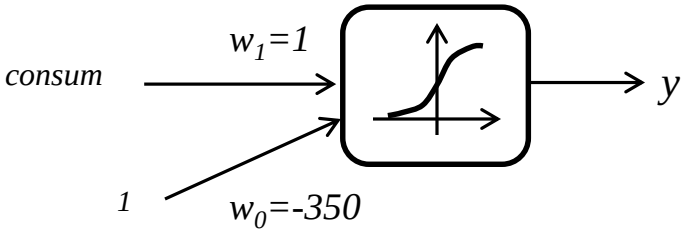


- Pentru un neuron logistic: modelul $X=(x_1, \ldots x_i, \ldots, x_n)$ *şi* $net(X)=w_1 \cdot x_1 + \ldots + w_i \cdot x_i + \ldots + w_n \cdot x_n$,
- dacă $y=logsig(net)=0.83$, atunci modelul X aparţine cu probabilitate 0.83 clasei 1 şi cu probabilitate 0.17 clasei 0.

  - Neuronul va clasifica un model în clasa 1 atunci când intrarea sa netă va avea valori 0 sau pozitive, $net(X)=w_1 \cdot x_1 + \ldots + w_i \cdot x_i + \ldots + w_n \cdot x_n > 0$;
  - Neuronul va clasifica un model în clasa 0 atunci când intrarea sa netă va avea valori negative, $net(X)=w_1 \cdot x_1 + \ldots + w_i \cdot x_i + \ldots + w_n \cdot x_n <= 0$;

Clasificarea cu neuronul logistic





$$net = 1\cdot(-350) + consum\cdot 1 = -350 + consum$$

$$consum >= 350 \Leftrightarrow net >= 0, logsig(net) >= 0.5$$
$$consum < 350 \Leftrightarrow net < 0, logsig(net) < 0.5$$

- Problema antrenării neuronului rămâne cea de determinare a ponderilor optime w_i, astfel încât, pentru toate modelele de intrare, recunoaşterea să fie cât mai exactă (modelele 0 să fie recunoscute în clasa 0, modelele 1 să fie recunoscute în clasa 1)
- Dreapta de separaţie se numeşte frontieră (decision boundary)

Metoda gradientului pentru neuronul logistic

- Actualizarea ponderilor:

$$w_i^{(t+1)} = w_i^{(t)} - \eta \cdot \frac{\partial E}{\partial w_i^{(t)}}\Big|_{w=w_i^{(t)}}, \quad i = 0...n$$

$$E = \frac{1}{2} \cdot \sum_{m=1}^{M} \left(d^{(m)} - o^{(m)} \right)^2$$

- Pentru calculul derivatelor erorii, se aplică regula derivării în lanţ:

$$\frac{\partial E}{\partial w_i^{(t)}} = \frac{1}{2} \cdot \sum_{m=1}^{M} \frac{\partial o^{(m)}}{\partial w_i} \cdot \frac{\partial E}{\partial o^{(m)}} = \frac{1}{2} \cdot \sum_{m=1}^{M} \frac{\partial net^{(m)}}{\partial w_i} \cdot \frac{\partial o^{(m)}}{\partial net^{(m)}} \cdot \frac{\partial E}{\partial o^{(m)}}$$

$$o^{(m)} = \frac{1}{1 + e^{net^{(m)}}}$$

$$net^{(m)} = w_1 \cdot x_1^{(m)} + ... + w_i \cdot x_i^{(m)} + ... + w_n \cdot x_n^{(m)}$$

$$\frac{\partial E}{\partial w_i^{(t)}} = -\sum_{m=1}^{M} x_i^{(m)} \cdot o^{(m)} \cdot (1 - o^{(m)}) \cdot \left(d^{(m)} - o^{(m)} \right)$$

$$\frac{\partial net^{(m)}}{\partial w_i} = x_i^{(m)}$$

$$\frac{\partial o^{(m)}}{\partial net^{(m)}} = o^{(m)} \cdot (1 - o^{(m)})$$

$$\frac{\partial E}{\partial o^{(m)}}\Big|_{w=w_i^{(t)}} = -2 \cdot \left(d^{(m)} - o^{(m)} \right)$$

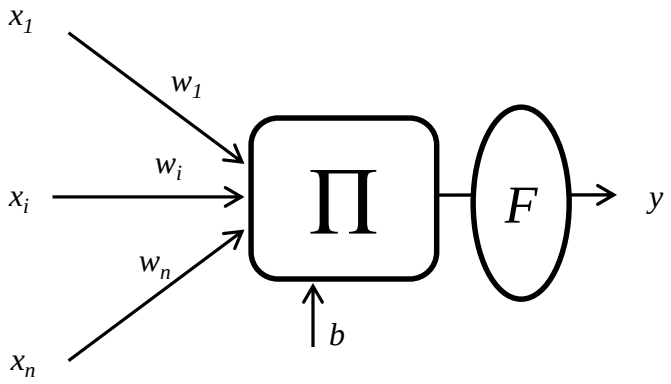
Metoda gradientului pentru neuronul logistic

- Formularea este aceeaşi cu cea pentru regresia multiplă, se schimbă doar modul de calcul al derivatelor erorii:

- Iniţializează aleatoriu $w_0, w_1, \ldots w_i, \ldots w_n$
- Iniţializează contor iteraţii t=0
- repetă
  - ❖ Incrementează iteraţie: t=t+1
  - ❖ calculează corecţii ponderi $\Delta w_i^{(t+1)} - \eta \cdot \frac{\partial E}{\partial w_i^{(t)}}|_{w_i}, \quad i = 0 \ldots n$
  - ❖ Corectează ponderi: $w_0^{(t+1)} = w_i^{(t)} + \Delta w_i^{(t+1)}, \quad i = 0 \ldots n$
  - ❖ Recalculează eroarea E
  - ❖ Verifică criteriul de oprire (nr. iteraţii sau progres eroare)
- Dacă s-a îndeplinit condiţia de oprire, antrenarea s-a încheiat.

Alte tipuri de neuroni

- Neuronul produs



$$y = F(b + w_1 \cdot x_1 \cdot \ldots \cdot w_i \cdot x_i \cdot \ldots \cdot w_n \cdot x_n) =$$

$$= F\left(b + \prod_{i=1}^{n} w_i \cdot x_i\right)$$

- Spiking neurons

  - Se activează doar când ating un potenţial minim, calculat separat de funcţia de activare.

Sunt mai greu de antrenat

va urma...

- Reţele neuronale artificiale pentru învăţare supravegheată