

Curs 9:

Modele de regresie

Structura

- Motivație
- Corelații, coeficient de corelație
- Regresie liniară
- Modele neliniare
 - Arbori de regresie
 - Rețele RBF

Motivație

Problema: Pornind de la caracteristici cunoscute ale unei mașini (e.g. Nr cilindri, cai putere, greutate, model etc) se dorește estimarea consumului de combustibil (e.g. exprimat prin “miles per gallon”)

Exemplu [autoMpg.arff de la <http://archive.ics.uci.edu/ml/datasets.html>]

```
@relation autoMpg
```

```
@attribute cylinders { 8, 4, 6, 3, 5} @attribute displacement real
```

```
@attribute horsepower real @attribute weight real @attribute acceleration real
```

```
@attribute model { 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82}
```

```
@attribute origin { 1, 3, 2}
```

```
@attribute class real
```

```
@data
```

```
8,307,130,3504,12,70,1,18
```

```
8,350,165,3693,11.5,70,1,15
```

```
4,113,95,2372,15,70,3,24
```

```
6,198,95,2833,15.5,70,1,22
```

```
6,199,97,2774,15.5,70,1,18
```

Motivation

Problema: Pornind de la caracteristici cunoscute ale unei mașini (e.g. Nr cilindri, cai putere, greutate, model etc) se dorește estimarea consumului de combustibil (e.g. exprimat prin “miles per gallon”)

Exemplu [autoMpg.arff de la <http://archive.ics.uci.edu/ml/datasets.html>]

```
@relation autoMpg
```

```
@attribute cylinders { 8, 4, 6, 3, 5} @attribute displacement real
```

```
@attribute horsepower real @attribute weight real @attribute acceleration real
```

```
@attribute model { 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82}
```

```
@attribute origin { 1, 3, 2}
```

```
@attribute class real
```

```
@data
```

```
8,307,130,3504,12,70,1,18
```

```
8,350,165,3693,11.5,70,1,15
```

```
4,113,95,2372,15,70,3,24
```

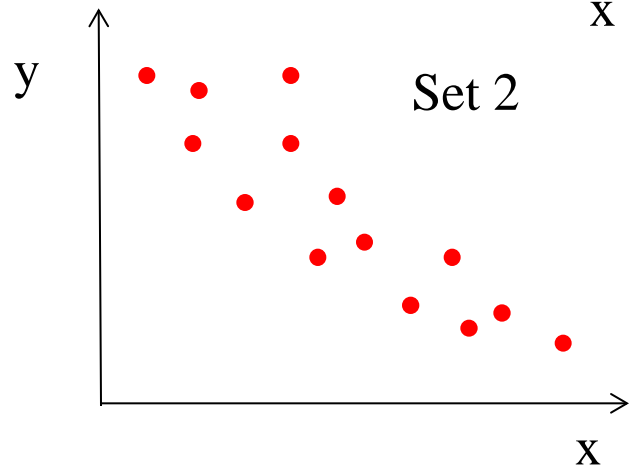
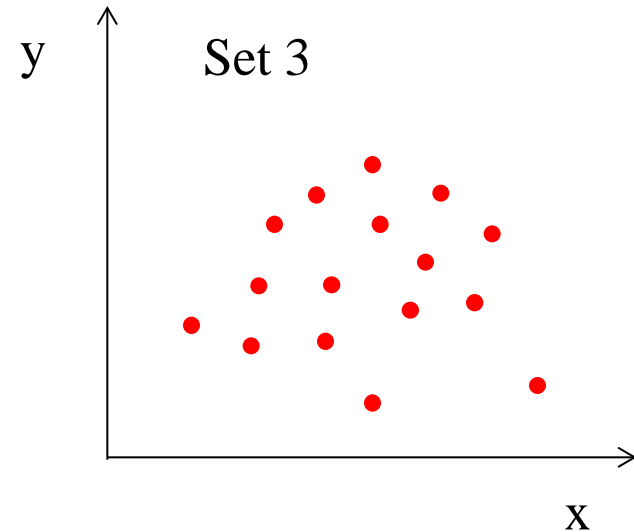
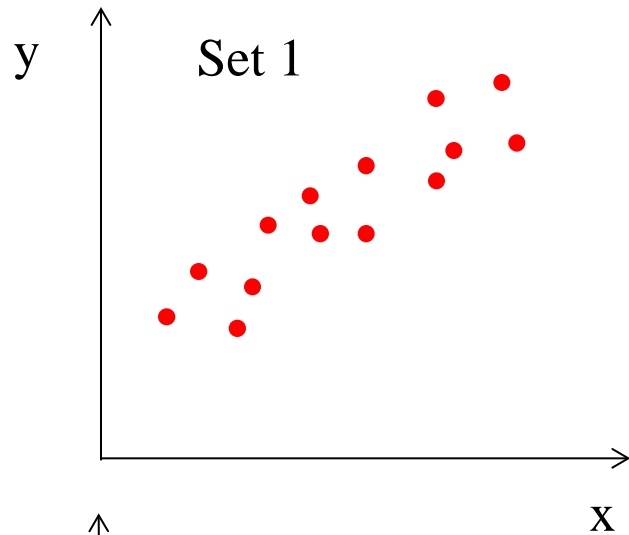
```
6,198,95,2833,15.5,70,1,22
```

```
6,199,97,2774,15.5,70,1,18
```

Se caută o relație care să descrie dependența dintre consumul de combustibil (atributul class în setul de date) și caracteristicile mașinii (primele 7 atribute din setul de date)

Un exemplu mai simplu

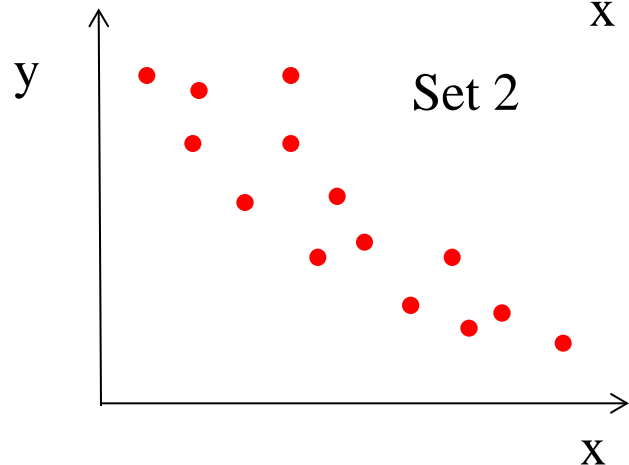
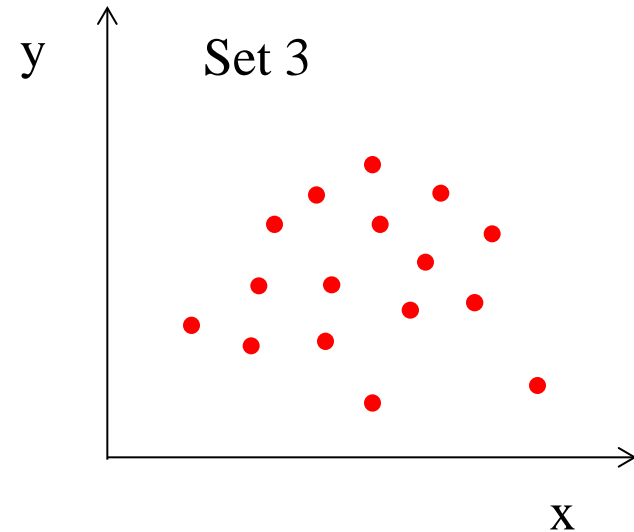
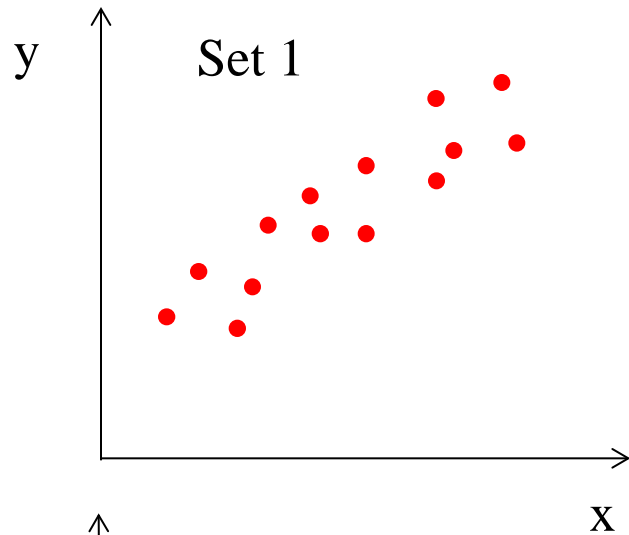
Câteva seturi de date generate artificial



Ce se poate spune despre datele din fiecare set?

Un exemplu mai simplu

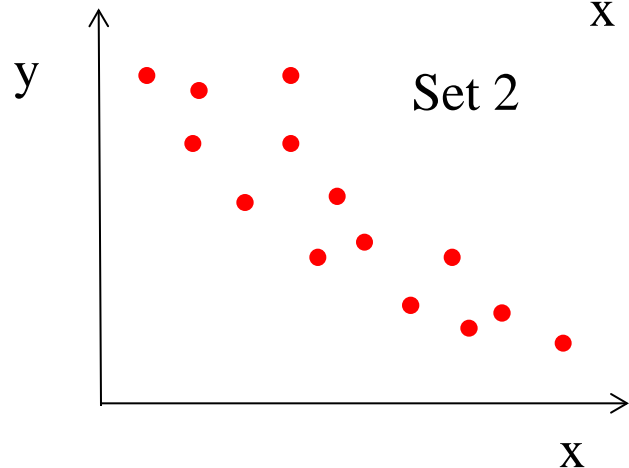
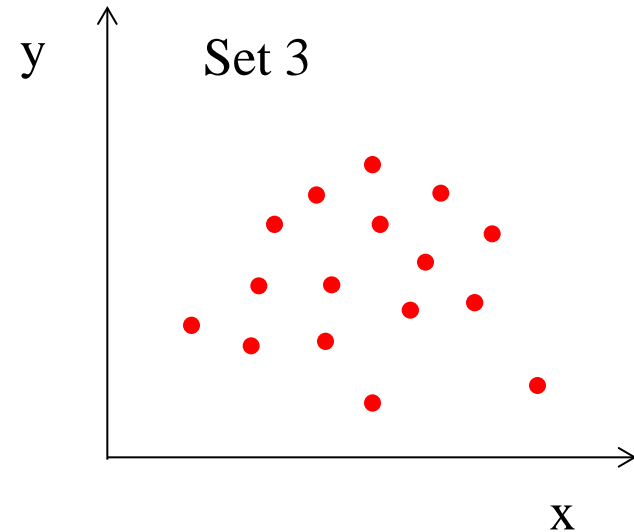
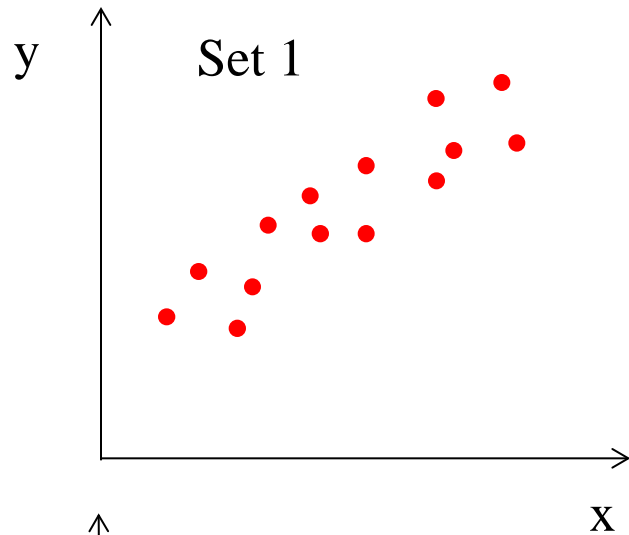
Câteva seturi de date generate artificial



Set 1: datele par să fie corelate pozitiv
= dacă x crește atunci și y crește

Un exemplu mai simplu

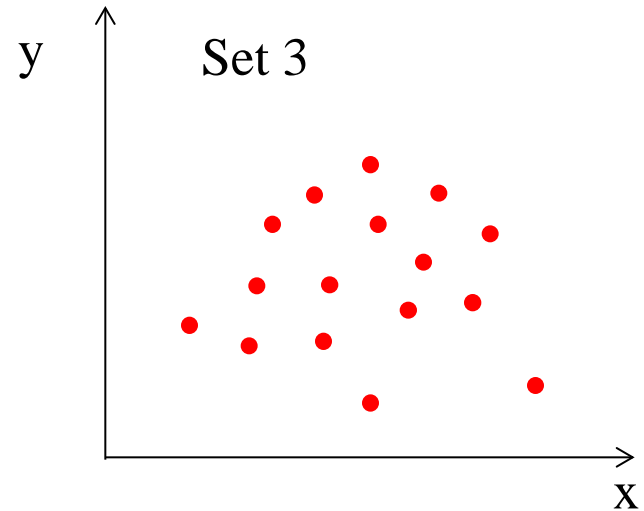
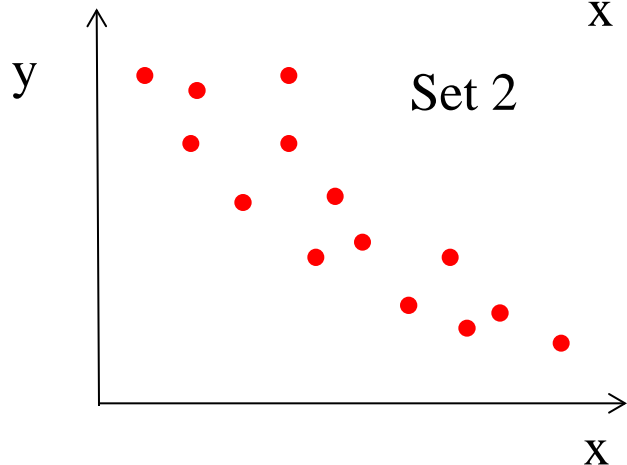
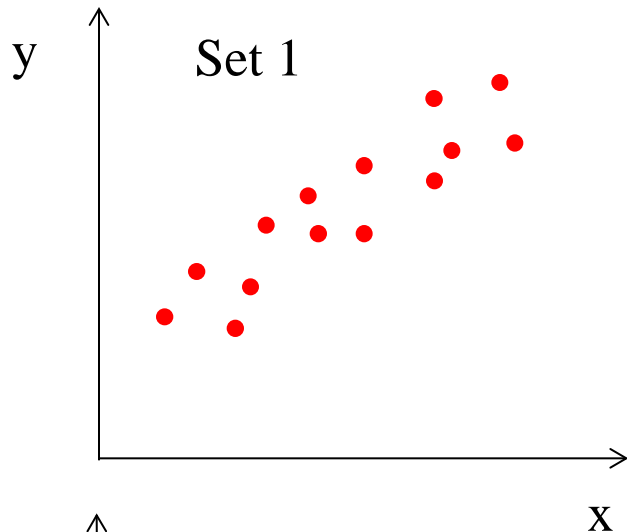
Câteva seturi de date generate artificial



Set 2: datele par să fie corelate negativ
= dacă x crește atunci y scade

Un exemplu mai simplu

Câteva seturi de date generate artificial



Set 3: datele par să nu fie corelate (e doar un nor de puncte)

Intrebari:

- Cum poate fi masurat gradul de corelație?
- Ce tip de corelație există?

Coeficient de corelație

Cum poate fi masurat gradul de corelație?

[reminder – Probabilități și Statistică]

- De exemplu folosind **coeficientul de correlation Pearson** – exprimă **gradul de corelație liniară** dintre cele două variabile

$$R(X, Y) = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \text{avg}(X))(y_i - \text{avg}(Y))}{\text{stdev}(X)\text{stdev}(Y)}$$

Obs: $-1 \leq R(X, Y) \leq 1$

$$\text{stdev}(X) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \text{avg}(X))^2}$$

$$\text{stdev}(Y) = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \text{avg}(Y))^2}$$

$$\text{avg}(X) = \frac{1}{n} \sum_{i=1}^n x_i, \quad \text{avg}(Y) = \frac{1}{n} \sum_{i=1}^n y_i$$

- $R(X, Y)$ apropiat 1: corelație liniară pozitivă
- $R(X, Y)$ apropiat to -1: corelație liniară negativă
- $R(X, Y)$ apropiat to 0: nu sunt corelate liniar (poate exista corelație neliniară între X și Z)

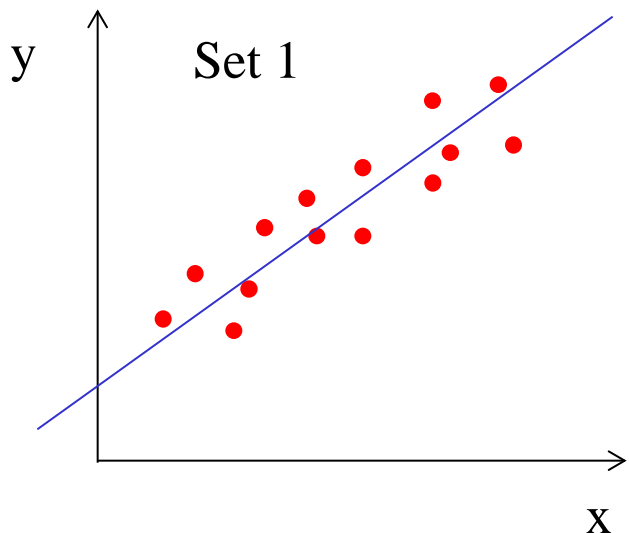
Regresie liniară

Ce tip de corelație ? [reminder – Statistică]

Cazul cel mai simplu: **Dependența liniară** dintre două variabile: $Y=w_1X+w_0$

- X = variabila predictor (independentă, intrare, explicativă)
- Y = variabila prezisă (dependentă, răspuns, explicată)

Scopul regresiei liniare: estimarea parametrilor w_1 și w_0 a.î. valorile asociate variabilelor X (i.e. x_1, x_2, \dots, x_n) și Y (i.e. y_1, y_2, \dots, y_n) sunt bine explicate de către funcția liniară, i.e. Suma pătratelor erorilor este minimizată



$$SSE(w_1, w_0) = \sum_{i=1}^n (y_i - (w_1 x_i + w_0))^2$$

$$= \sum_{i=1}^n (y_i - \overline{w} x_i)^2$$

$$(\overline{w} = (w_1, w_0), \overline{x}_i = (x_i, 1)^T)$$

Vector linie

Vector coloană

Regresie liniară simplă

Reminder: algebra liniară

$$w = (w_1, w_0), D = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ 1 & 1 & \dots & 1 \end{pmatrix}^T, y = (y_1, y_2, \dots, y_n)^T$$

$$\begin{aligned} SSE(w) &= \|y - Dw^T\|^2 = (y - Dw^T)^T (y - Dw^T) \\ &= y^T y - 2wD^T y + wD^T Dw^T \end{aligned}$$

Determinarea vectorului w care minimizează $SSE(w)$ este echivalentă cu determinarea punctului critic al lui SSE , adică rezolvarea următoarelor ecuații în raport cu w :

$$D^T Dw^T = D^T y \Rightarrow w^T = (D^T D)^{-1} D^T y = D^+ y$$

$$D^+ = (D^T D)^{-1} D^T \text{ este pseudoinversa lui } D$$

Regresie liniară multiplă

Obs: abordarea poate fi extinsă în cazul mai multor variabile predictor (e.g. Setul autoMPG)

$$w = (w_1, w_2, \dots, w_d, w_0), D = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{d1} & x_{d2} & \dots & x_{dn} \\ 1 & 1 & \dots & 1 \end{pmatrix}^T, y = (y_1, y_2, \dots, y_n)^T$$

$$\begin{aligned} SSE(w) &= \|y - Dw^T\|^2 = (y - Dw^T)^T (y - Dw^T) \\ &= y^T y - 2wD^T y + wD^T Dw^T \end{aligned}$$

$$D^T Dw^T = D^T y \Rightarrow w^T = (D^T D)^{-1} D^T y$$

Regresie liniară - regularizare

Obs: dacă matricea $D^T D$ este singulară (inversa nu poate fi calculată) atunci funcția obiectiv (SSE) este modificată prin adăugarea unui termen de **regularizare** care va modifica matricea în așa fel încât să se obțină o matrice inversabilă).

Exemple:

- Regularizare Tikhonov (ridge regression)

$$SSE'(w) = SSE(w) + \lambda \|w\|^2$$

$$w = (D^T D + \lambda I)^{-1} D^T y$$

$I = (d + 1) \times (d + 1)$ matrice identitate

Obs:

- Termenul de penalizare “descurajează” valorile mari ale parametrilor
- Parametrul termenului de regularizare (**lambda**) poate fi ales în manieră adaptivă folosind validare încrucișată

Regresie liniară - regularizare

Obs: dacă matricea $D^T D$ este singulară (inversa nu poate fi calculată) atunci funcția obiectiv (SSE) este modificată prin adăugarea unui termen de **regularizare** care va modifica matricea în așa fel încât să se obțină o matrice inversabilă).

Exemple:

- Regularizare Lasso

$$SSE'(w) = SSE(w) + \lambda \sum_{i=1}^d |w_i|$$

Obs:

- În acest caz problema de optimizare se rezolvă folosind metode numerice
- Este utilă în cazul problemelor cu multe variabile dintre care o mare parte sunt irelevante (specific pt “sparse models”)

Modele liniare generalizate

Idee: în loc de $y_i = w_1 x_i + w_0$ ieșirea (y_i) este modelată printr-o variabilă aleatoare care are media $f(w_1 x_i + w_0)$

Principalele elemente ale unui model GLM (generalized linear model):

- Funcția de medie (mean function): f
- Funcția de legătură (link function): f^{-1}
- Distribuția de probabilitate (probability distribution)

Mean function	Link function	Distribution
$f(u) = u$	identity	normal
$f(u) = -1/u$	inverse	exponential, gamma
$f(u) = \exp(u)$	Log	Poisson
$f(u) = 1/(1 + \exp(-u))$	Logit	Bernoulli

Modele liniare generalizate

Idee: în loc de $y_i = w_1 x_i + w_0$ ieșirea (y_i) este modelată printr-o variabilă aleatoare care are media $f(w_1 x_i + w_0)$

Principalele elemente ale unui model GLM (generalized linear model):

- Funcția de medie (mean function): f
- Funcția de legătură (link function): f^{-1}
- Distribuția de probabilitate (probability distribution)

Mean function	Link function	Distribution
$f(u) = u$	identity	normal
$f(u) = -1/u$	inverse	exponential, gamma
$f(u) = \exp(u)$	Log	Poisson
$f(u) = 1/(1 + \exp(-u))$	Logit	Bernoulli

Regresie
clasică
(metoda celor
mai mici
pătrate)

Modele liniare generalizate

Idee: în loc de $y_i = w_1 x_i + w_0$ ieșirea (y_i) este modelată printr-o variabilă aleatoare care are media $f(w_1 x_i + w_0)$

Principalele elemente ale unui model GLM (generalized linear model):

- Funcția de medie (mean function): f
- Funcția de legătură (link function): f^{-1}
- Distribuția de probabilitate (probability distribution)

Mean function	Link function	Distribution
$f(u) = u$	identity	normal
$f(u) = -1/u$	inverse	exponential, gamma
$f(u) = \exp(u)$	Log	Poisson
$f(u) = 1/(1 + \exp(-u))$	Logit	Bernoulli

Regresie
logistică

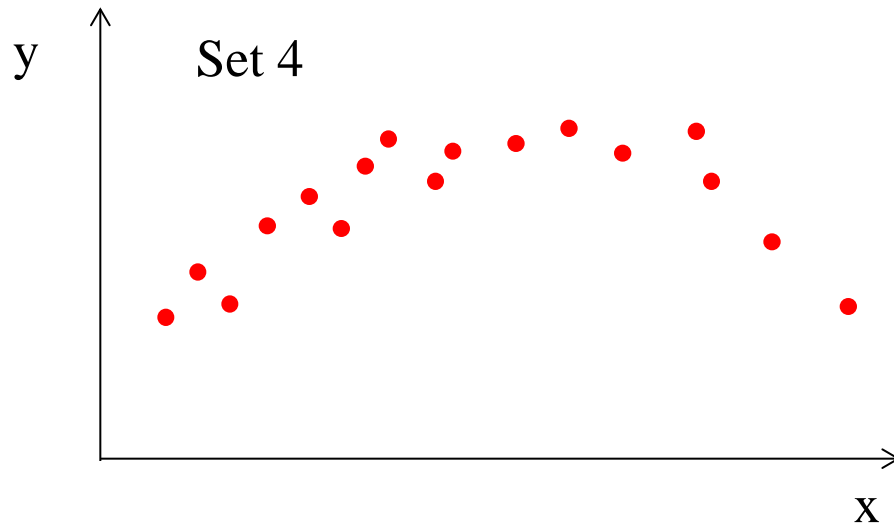
Regresie neliniară

Cum se abordează cazul în care dependența dintre variabila prezisă și cele predictor nu este liniară?

Sunt necesare alte modele

Exemple:

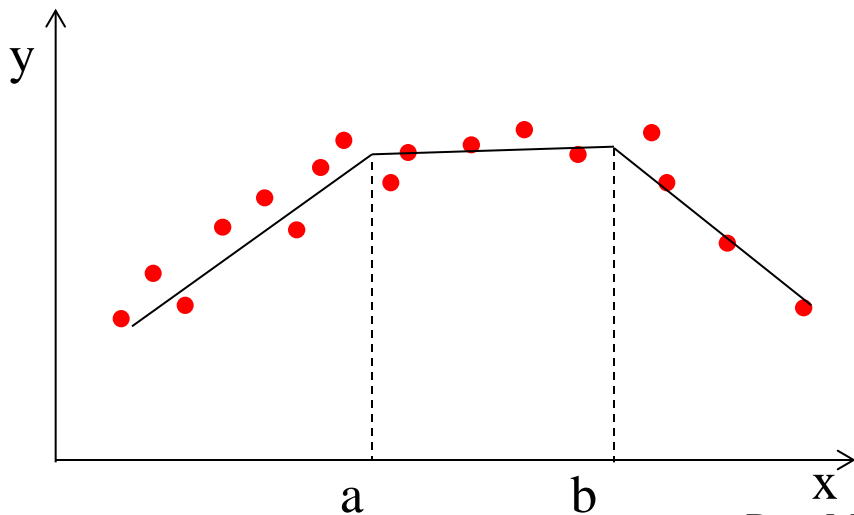
- Arbori de regresie
- Rețele neuronale



Regresie neliniară

Ideea principală:

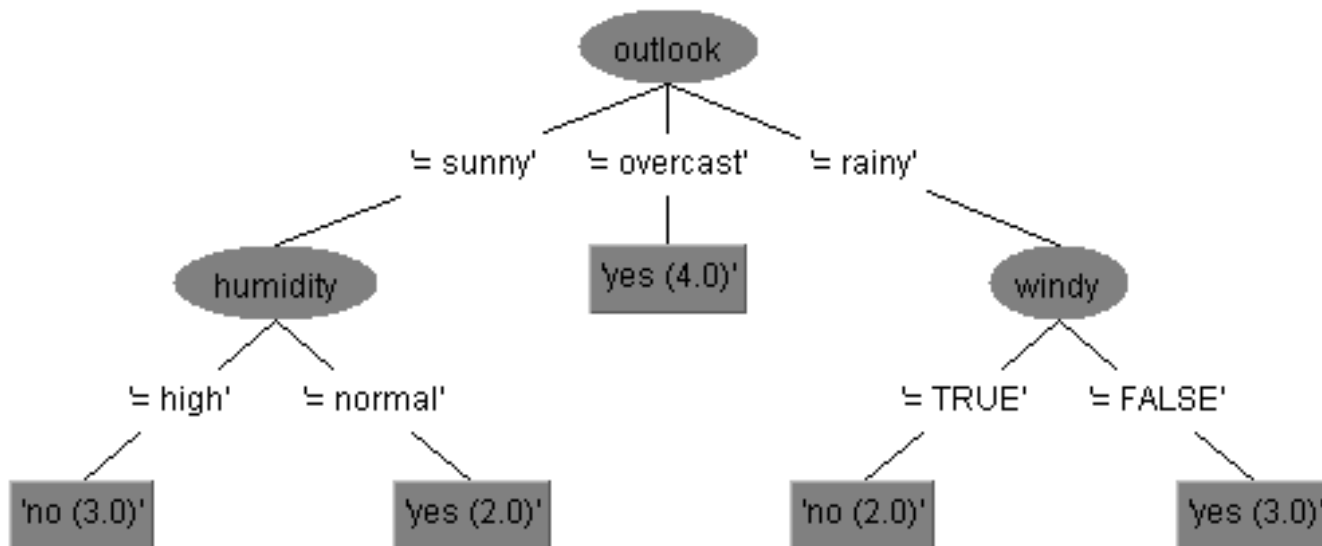
- O dependență neliniară poate fi modelată prin mai multe funcții liniare (câte una pentru fiecare regiune)
- Procesul de regresie constă din două etape:
 - Identificarea regiunilor prin partiționarea spațiului variabilelor predictor
 - Identificarea modelului de regresie (liniar) pt fiecare dintre regiuni



Arbori de regresie

Reminder:

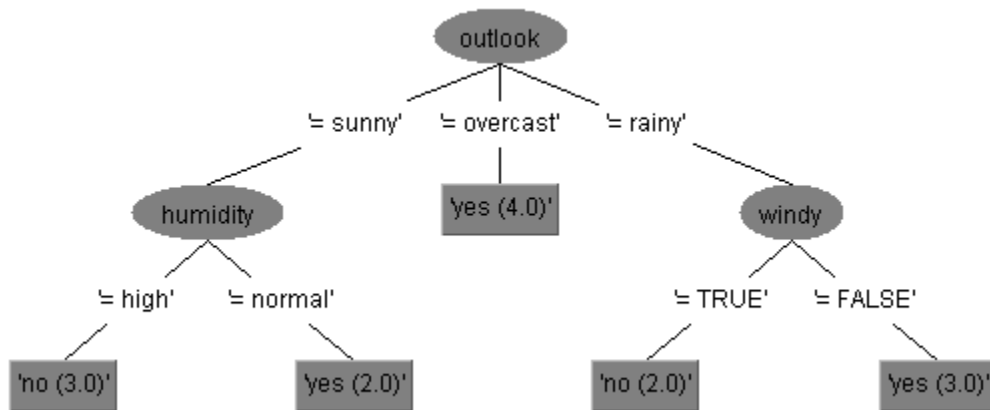
Arbori de decizie= arbore în care nodurile interne conține condiții referitoare la variabilele predictor iar cele frunză sunt informații privind variabila predictor (în cazul arborilor de clasificare variabila prezisă este discretă și nodurile frunză conțin indicatori de clasă)



Arbori de regresie

Reminder:

Arbori de decizie = arbore în care nodurile interne conține condiții referitoare la variabilele predictor iar cele frunză sunt informații privind variabila predictor (în cazul arborilor de clasificare variabila prezisă este discretă și nodurile frunză conțin indicatori de clasă)



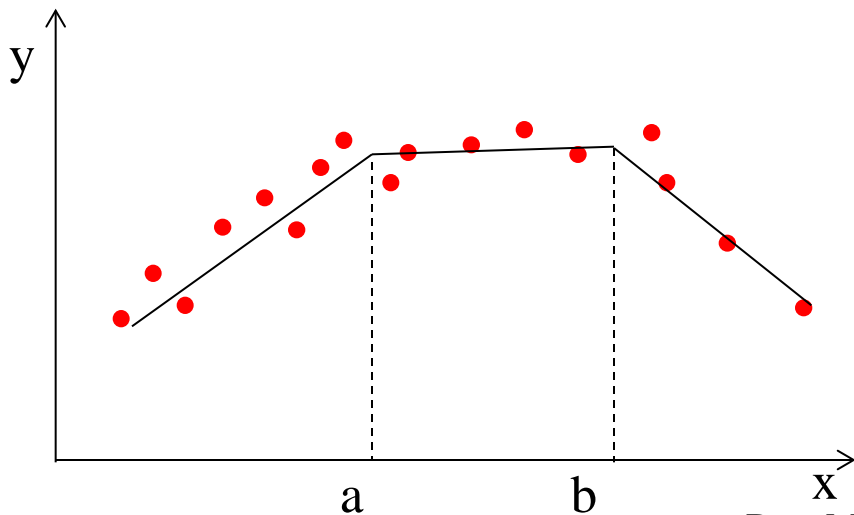
Intrebare:

- Dar dacă variabila prezisă este continuă? (ex: în locul unui răspuns de tipul da/nu în cazul problemei “weather-play” ar fi o valoare $[0,1]$ care ar exprima un nivel de decizie între 0 (nu) și 1 (da))

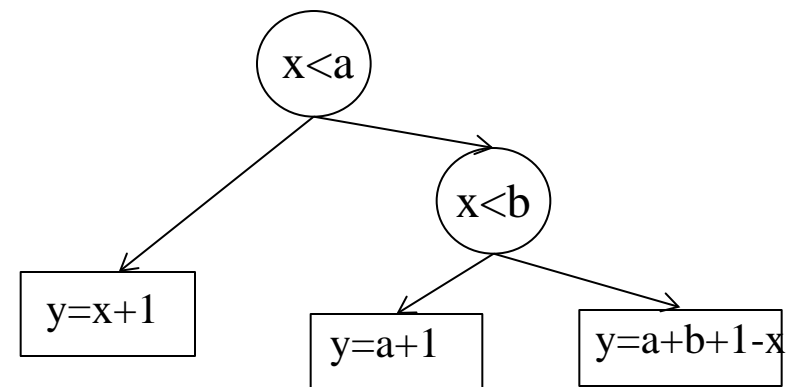
Arbori de regresie

Ideea principală:

- Se utilizează un proces similar de partiționare a spațiului de decizie ca și în cazul arborilor de clasificare
- Pt variabile predictor continue condiția de ramificare este: **variabila < valoare** sau **variabila > valoare** sau **variabila in [min,max]**
- Se deduce un model de regresie (de exemplu liniar) pt fiecare dintre regiunile identificate prin procedura de ramificare



Exemplu foarte simplu -> model liniar pe porțiuni



Regresie neliniară

Dincolo de modelele de regresie liniară pe porțiuni:

- Se extinde modelul clasic de regresie liniară considerând attribute transformate prin intermediul unor funcții

$$y = w_0 + w_1 h_1(x) + w_2 h_2(x) + \dots + w_m h_m(x)$$

(x e un vector h_i e o funcție ce asociază un scalar sau un vector argumentului său)

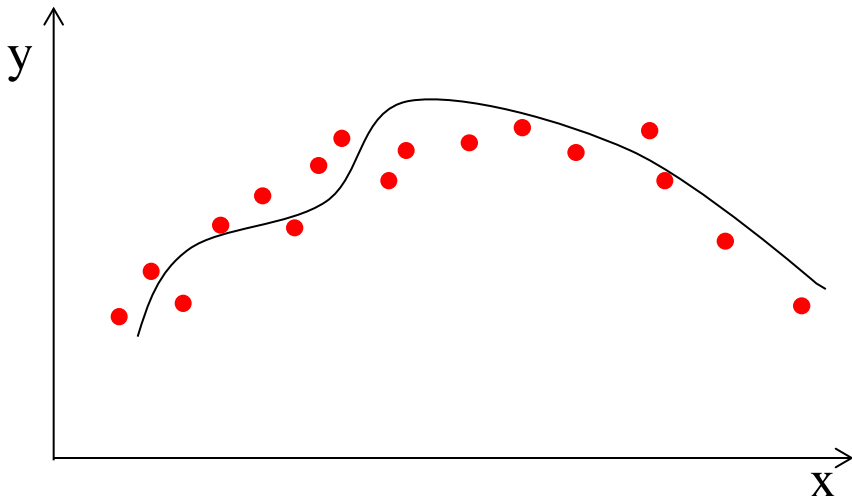
Caz particular 1. Modele polinomiale: $y = w_0 + w_1 x + w_2 x^2 + \dots + w_m x^m$

(x este un scalar)

Caz particular 2.

Modele bazate pe funcții nucleu (kernel functions): h_i sunt funcții care iau valori semnificative doar pr regiuni limitate din spațiul variabilelor predictor

- Dacă aceste funcții au simetrie radială (de exemplu funcții gaussiene) se ajunge la rețelele de tip RBF (un caz particular de rețele neuronale)

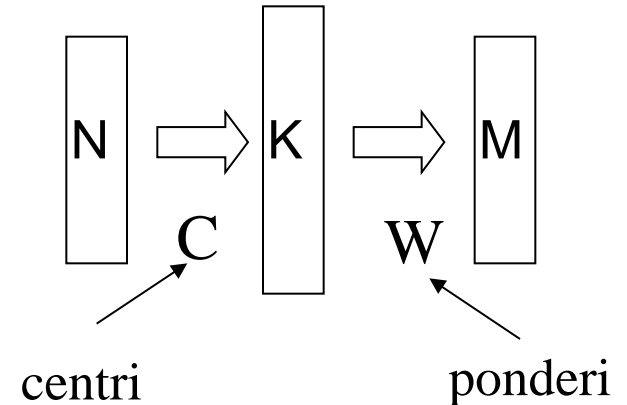


Rețele cu funcții radiale

- RBF - “Radial Basis Function”:

- Arhitectura:

- Două nivele de unități funcționale



- Funcții de agregare:

- Unități ascunse: distanța dintre vectorul de intrare și cel al ponderilor corespunzătoare unității ascunse
- Unități de ieșire: suma ponderată

$$G(X, C^k) = \|X - C^k\| = \sqrt{\sum_{i=1}^N (x_i - c_i^k)^2}$$

Funcții de transfer (activare):

- nivelul ascuns: funcții cu simetrie radială
- nivelul de ieșire: funcții liniare

Rețele cu funcții radiale

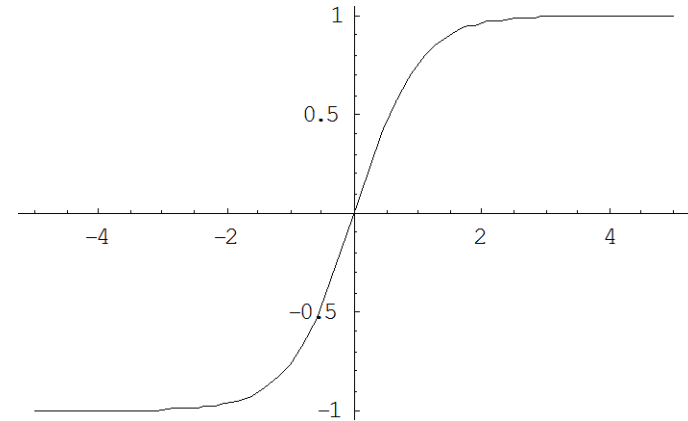
Diferența față de rețelele feedforward clasice:

Funcții de transfer:

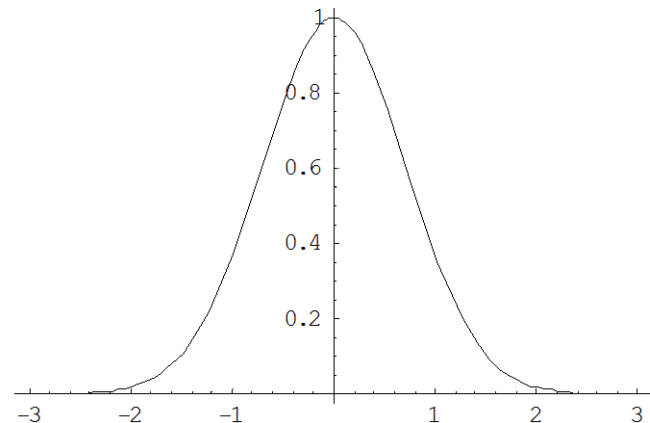
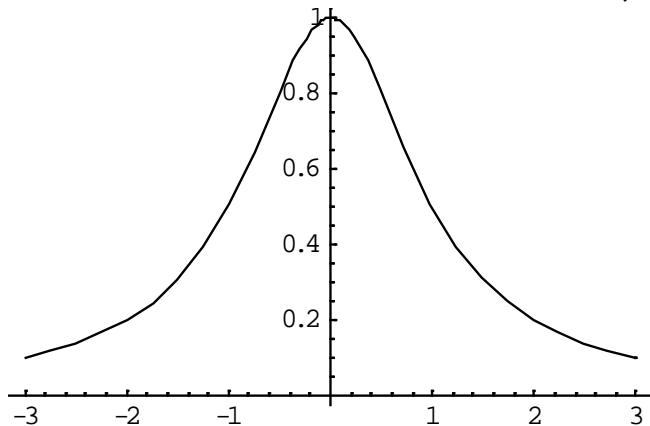
FF: funcții sigmoidale

RBF: funcții cu simetrie radială

Funcție sigmoidală



Funcții cu simetrie radială

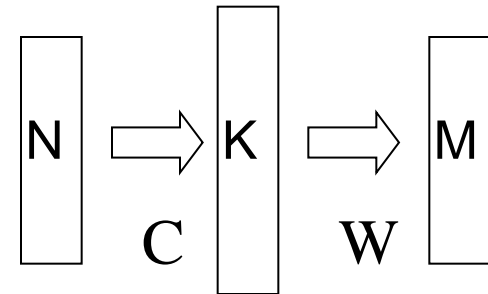


Rețele cu funcții radiale

Funcționare:

$$y_i = \sum_{k=1}^K w_{ik} g(\|X - C^k\|) - w_{i0}, \quad i = \overline{1, M}$$

$$y_i = \sum_{k=1}^K w_{ik} z_k - w_{i0}, \quad z_k = g(\|X - C^k\|)$$



Matrice centri

Matrice ponderi

Parametrii C^k pot fi interpretați ca **prototipuri (centri)** asociați unităților ascunse: vectorii de intrare X apropiați lui C^k vor conduce la o valoare de ieșire semnificativă pe când cei îndepărtați vor conduce la o valoare de ieșire nesemnificativă; la construirea răspunsului rețelei vor contribui doar unitățile a căror centri sunt suficient de similari cu data de intrare

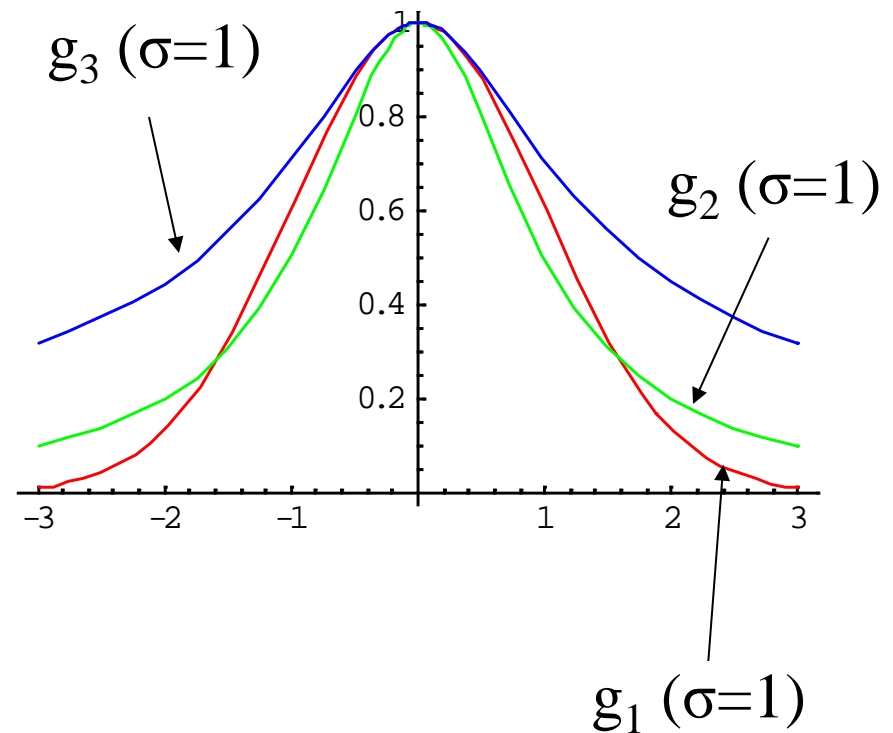
Rețele cu funcții radiale

Exemple de funcții radiale:

$$g_1(u) = \exp(-u^2 / (2\sigma^2))$$

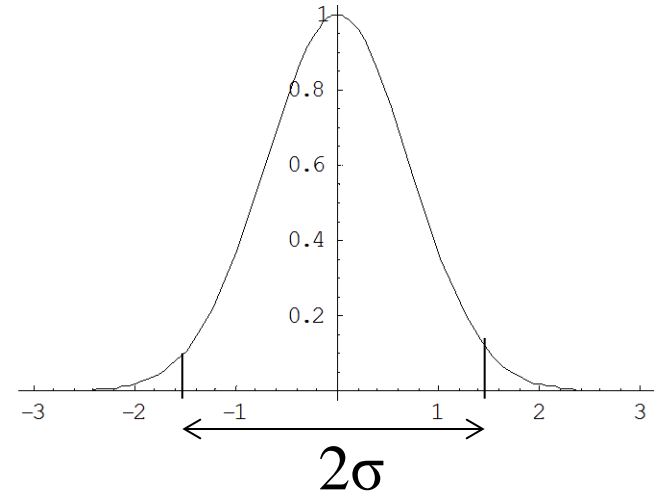
$$g_2(u) = 1/(u^2 + \sigma^2)$$

$$g_3(u) = 1/\sqrt{u^2 + \sigma^2}$$

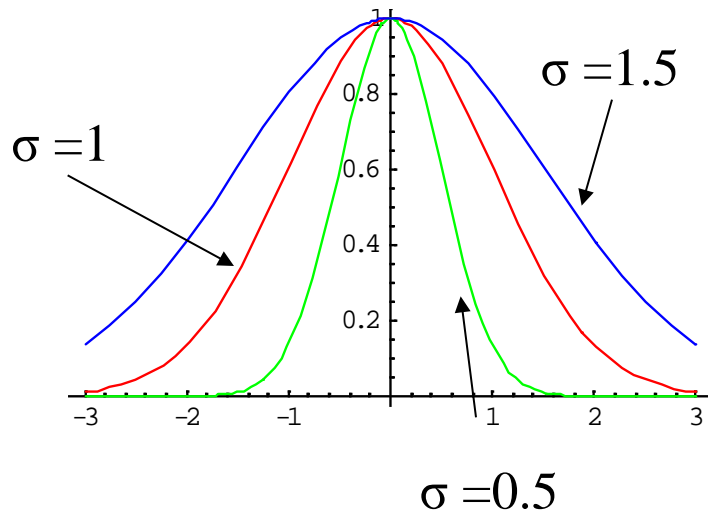


Rețele cu funcții radiale

- Fiecare unitate ascunsă este “sensibilă” la semnalele de intrare provenite dintr-o regiune a spațiului de intrare aflată în vecinătatea centrului. Aceasta regiune este denumită **câmp receptiv**
- Dimensiunea câmpului receptiv depinde de σ

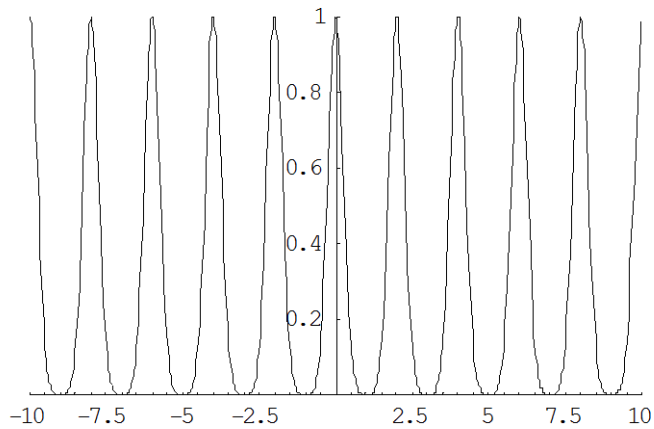
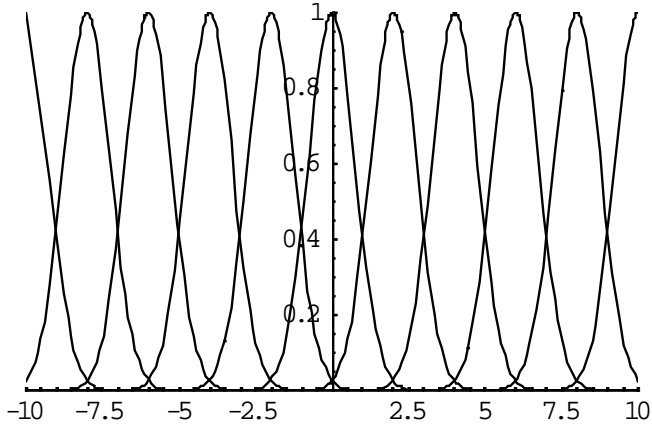


$$g(u) = \exp\left(-\frac{u^2}{2\sigma^2}\right)$$

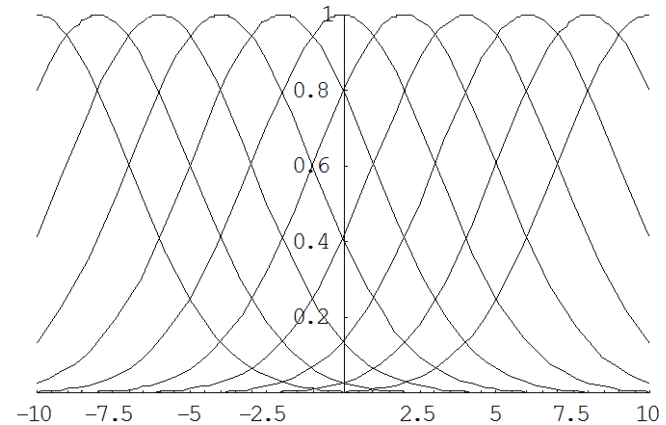
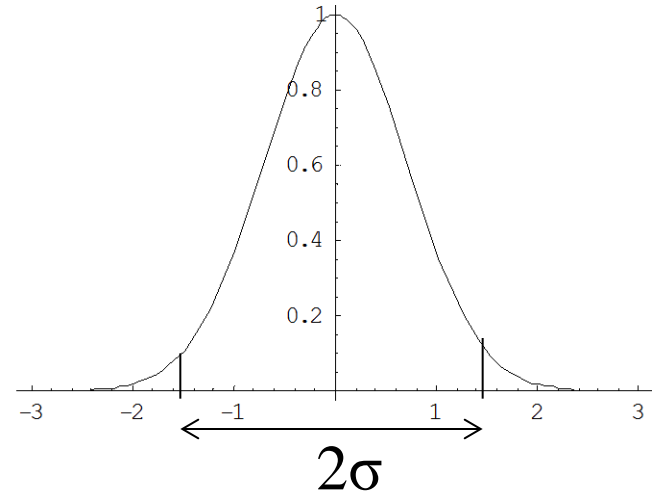


Rețele cu funcții radiale

Influența lui σ : $g(u) = \exp\left(-\frac{u^2}{2\sigma^2}\right)$



subacoperire

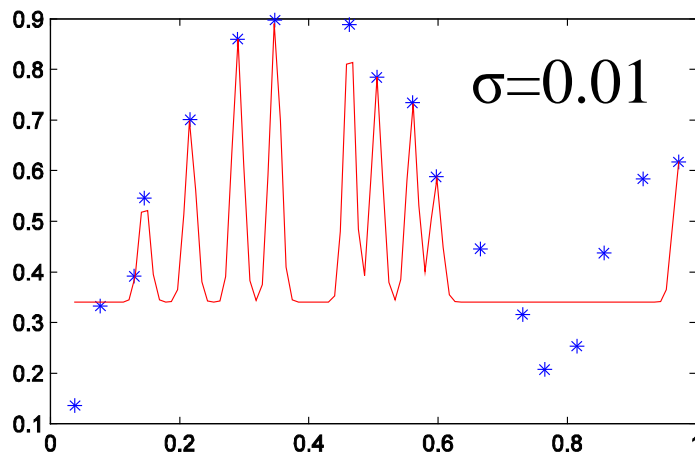
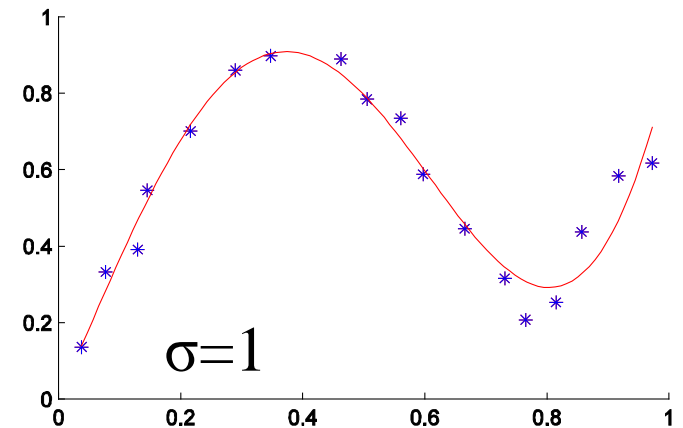


supraacoperire

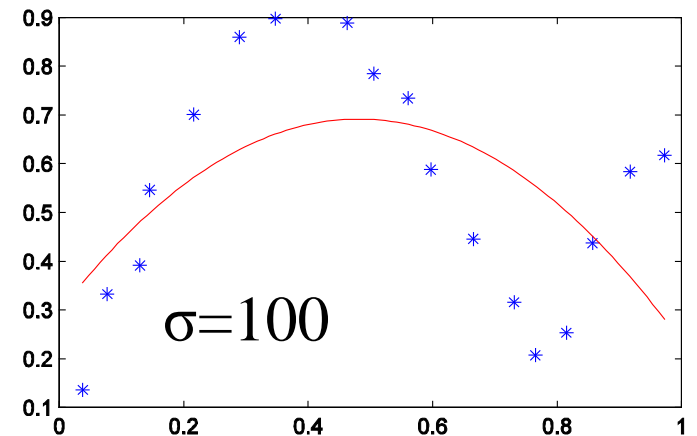
Rețele cu funcții radiale

- O bună acoperire a domeniului datelor de intrare de către câmpurile receptive ale funcțiilor radiale de transfer este esențială pentru calitatea aproximării
- Valori prea mici conduc la incapacitatea de a produce rezultate pentru întreg domeniul datelor
- Valori prea mari nu surprind variabilitatea datelor

Acoperire adecvată



Subacoperire

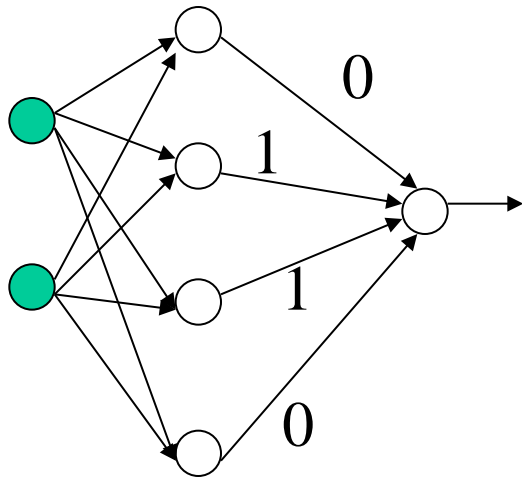


Supraacoperire

Rețele cu funcții radiale

Exemplu (caz particular) : rețea RBF pentru reprezentarea lui XOR

- 2 unități de intrare
- 4 unități ascunse
- 1 unitate de ieșire



Centrii:

u.a. 1: (0,0)

u.a. 2: (1,0)

u.a. 3: (0,1)

u.a. 4: (1,1)

Ponderi:

w1: 0

w2: 1

w3: 1

w4: 0

Funcție de activare:

$g(u)=1$ if $u=0$

$g(u)=0$ if $u \neq 0$

Aceasta abordare nu poate fi aplicată pentru probleme generale de aproximare

Rețele cu funcții radiale

Invățare:

Set de antrenare: $\{(x^1, d^1), \dots, (x^L, d^L)\}$

Etape:

- (a) Stabilirea parametrilor corespunzatori nivelului ascuns: centrul C și parametrii σ
- (b) Determinarea parametrilor W (problemă de optimizare liniară)

Obs: Invățarea de tip RBF elimină o parte dintre dezavantajele algoritmului BP: convergența lentă, blocarea în minime locale (întrucât se ajunge la rezolvarea unei probleme mai simple de optimizare) etc.

Rețele cu funcții radiale

Invățare:

Set de antrenare: $\{(x^1, d^1), \dots, (x^L, d^L)\}$

(a) Stabilirea parametrilor corespunzători nivelului ascuns: centrii C și parametrii σ

(a) $K=L$ (nr centri = nr exemple), $C^k=x^k$

(b) $K<L$: centrii se stabilesc

(a) prin selecție aleatoare dintre exemplele din setul de antrenare

(b) prin selecție sistematică dintre exemplele din setul de antrenare (Orthogonal Least Squares)

(c) prin utilizarea unui algoritm de grupare (poate permite și estimarea numărului de centri) – în acest caz centrii nu vor face neapărat parte din setul de antrenare

Rețele cu funcții radiale

Orthogonal Least Squares:

- Selecție incrementală a centrilor astfel încât eroarea să fie micșorată cât mai mult
- Noul centru este ales astfel încât să fie ortogonal pe spațiul generat de către centrii deja selectați (procesul este bazat pe metoda de ortogonalizare Gram-Schmidt)
- Abordarea este corelată cu regresia de tip “ridge”

Rețele cu funcții radiale

Grupare (clustering):

- Se urmărește identificarea a K clase în setul de date de antrenare $\{X_1, \dots, X_L\}$ astfel încât datele din fiecare clasă să fie suficient de similare pe când datele din clase diferite să fie suficient de diferite
- Fiecare clasă va avea un reprezentant (e.g. media datelor din clasă) care va fi considerat centrul clasei
- Algoritmii pentru determinarea reprezentanților clasei sunt cunoscuți sub numele de algoritmi partiționali (realizează o partiționare a spațiului de intrare)

Algoritm clasic: **K-means**

Rețele cu funcții radiale

Varianta incrementală:

- Se pornește cu un număr mic de centri inițializați aleator
- Se parcurge setul de antrenare:
 - Dacă există un centru suficient de similar cu data de intrare atunci componentele centrului respectiv se modifică pentru a asigura asimilarea datei de intrare în clasa aferentă centrului.
 - Dacă data de intrare este diferită semnificativ de toți centrii atunci este adăugat un nou centru (echivalent cu adăugarea unei noi unități ascunse) care este inițializat chiar cu data de intrare analizată

Obs: necesită definirea unor valori prag care să permită cuantificarea pt suficient de similar/diferit

Rețele cu funcții radiale

Antrenare incrementală pentru rețele RBF

$$K = K_0$$

$$C_i^k = \text{select}(\{X_1^i, \dots, X_L^i\}), i = 1..N; k = 1..K$$

$$t = 0$$

REPEAT

FOR $l = 1, L$ DO

determina $k^* \in \{1, \dots, K\}$ astfel incat $d(X^l, C^{k^*}) \leq d(X^l, C^k)$ pt orice k

IF $d(X^l, C^{k^*}) < \delta$ THEN $C^{k^*} := C^{k^*} + \eta \cdot (X^l - C^{k^*})$

ELSE $K = K + 1; C^K = X^l$

$$t = t + 1$$

$$\eta = \eta_0 t^{-\alpha}$$

UNTIL $t > t_{\max}$ OR $\eta < \varepsilon$

Rețele cu funcții radiale

Estimarea lărgimilor câmpurilor receptive.

- Reguli euristice:

$$\sigma = \frac{d_{\max}}{\sqrt{2K}}, \quad d_{\max} = \text{distanța maximă dintre centri}$$

$$\sigma_k = \gamma d(C^k, C^j), \quad C^j = \text{centrul cel mai apropiat de } C^k, \gamma \in [0.5, 1]$$

$$\sigma_k = \frac{1}{m} \sum_{j=1}^m d(C^k, C^j), \quad C^1, \dots, C^m : \text{cei mai apropiați } m \text{ centri de } C^k$$

- Proces iterativ intercalat:
 - Fixează valorile σ și optimizează valorile centrilor
 - Fixează valorile centrilor și optimizează valorile σ

Rețele cu funcții radiale

Determinarea ponderilor conexiunilor dintre nivelul ascuns și nivelul de ieșire:

- Problema este similară cu cea a antrenării unei rețele cu un singur nivel de unități cu funcții liniare de activare sau cu cea a estimării parametrilor unui model liniar de regresie

$$E(W) = \frac{1}{2} \sum_{l=1}^L \sum_{i=1}^M \left(d_i^l - \sum_{k=1}^K w_{ik} g_k^l \right)^2, \quad g_k^l = g(\|x^l - C^k\|)$$

$$E(W) = \frac{1}{2} \sum_{l=1}^L \|d^l - Wg^l\|^2 = \frac{1}{2} \sum_{l=1}^L (d^l - Wg^l)^T (d^l - Wg^l)$$

$$\nabla E(W) = - \sum_{l=1}^L (g^l)^T (d^l - Wg^l) = 0$$

$$G^T G W = G^T d$$

$$W = (G^T G)^{-1} G^T d$$

Rețele cu funcții radiale

Determinarea ponderilor conexiunilor dintre nivelul ascuns și nivelul de ieșire:

- Problema este similară cu cea a antrenării unei rețele cu un singur nivel de unități cu funcții liniare de activare
- Algoritm: Widrow-Hoff (caz particular al algoritmului BackPropagation)

□ Inicializare:

$w_{ij}(0) := \text{rand}(-1, 1)$ (ponderile sunt inițializate aleator în $[-1, 1]$),
 $p := 0$ (contor de iterații)

□ Proces iterativ

REPEAT

FOR $l := 1, L$ DO

 Calculează $y_i(l)$ și $\delta_i(l) = d_i(l) - y_i(l)$, $i = 1, M$

 Ajustează ponderile: $w_{ik} := w_{ik} + \text{eta} * \delta_i(l) * z_k(l)$

ENDFOR

 Calculează $E(W)$ pentru noile valori ale ponderilor

$p := p + 1$

UNTIL $E(W) < E^*$ OR $p > p_{\text{max}}$

Comparație: RBF vs. BP

Rețele de tip RBF:

- 1 nivel ascuns
- Funcții de agregare bazate pe distanțe (pt. nivelul ascuns)
- Funcții de activare cu simetrie radială (pt. nivelul ascuns)
- Unități de ieșire cu funcție liniară
- Antrenare separată a parametrilor adaptivi
- Similare cu tehnicile de aproximare locală

Rețele de tip BackPropagation(BP):

- Mai multe nivele ascunse
- Funcții de agregare bazate pe suma ponderată
- Funcții de activare sigmoide (pt. nivelul ascuns)
- Unități de ieșire liniare sau neliniare
- Antrenare simultană a parametrilor adaptivi
- Similare cu tehnicile de aproximare globală