Regresszió

Csorba János

Nagyméretű adathalmazok kezelése

2010. március 31.

A feladat

- X magyarázó attribútumok halmaza
- Y magyarázandó attribútum(ok)
- Kérdés: f : X -> Y
- a kapcsolat pár „tanítópontban ismert”

- Regresszió: Y folytonos
- Osztályozás: Y diszkrét

Tervezési szempontok

- Gyorsaság:
  - modell előállítás és használat
- Robosztusság
  - Hiányzó, kilógó (outlier) adatok
- Skálázhatóság
  - Sok adat esetén is használható
- Skála-invariancia
  - $X_i := X_i * a$ ($a>0$) esetén az eredmény nem változik (mértékegységváltás)
- Értelmezhetőség
  - Értelmezhető-e a modellből kinyerhető tudás az ember számára (Neurális hálókkal probléma…)

Teljesítmény mérés

- U hasznosság függvény
- f() optimális, ha E[U(Y,f(X))] maximális
- Általában „inverz hasznosság függvény” (Hibafüggv.) : L
- f() optimális, ha VOH = E[L(Y,f(X))] minimális
- VOH: Várható Osztályozási Hiba
- X,Y eloszlása nem ismert, így E[] helyett gyakorlatban átlagszámítás

Tipikus hiba függvények

- Négyzetes hiba: $L(Y,Y') = (Y-Y')^2$
- $f_{opt} = E[Y|X=x]$
- Abszolút hiba: $L(Y,Y') = |Y-Y'|$
- $f_{opt} = \text{median}(Y|X=x)$
- f_{opt}: elméleti regressziós görbe
- X,Y eloszlások ismeretében tehát volna optimális megoldás, ezt az eloszlást azonban általában nem ismerjük

Legközelebbi k szomszéd módszer

- Motiváció: közeli pontokhoz hasonló érték tartozik.
- Univerzális módszer (osztályozásra és regresszióra egyaránt használható)
- f(X) : Vegyük X-hez a k db legközelebbi tanítópontot. Y : A k pont Y értékeinek átlaga.
  (Osztályozás esetén többségi szavazás)
- „legközelebbi”: X többdimenziós. Pontok távolságát definiálni kell. Általában Euklideszi, de néha módosítani kell…

Módszer hatékonysága

- Optimum: f(x) = E[Y|X=x]
- A módszer ennek „közelítése” két szempontból
  - X=x helyett a közeli pontok, tehát X~=x
  - Várható érték helyett átlag
- A módszer univerzális approximátor, ha a pontok sűrűsége az egész térben elég nagy.

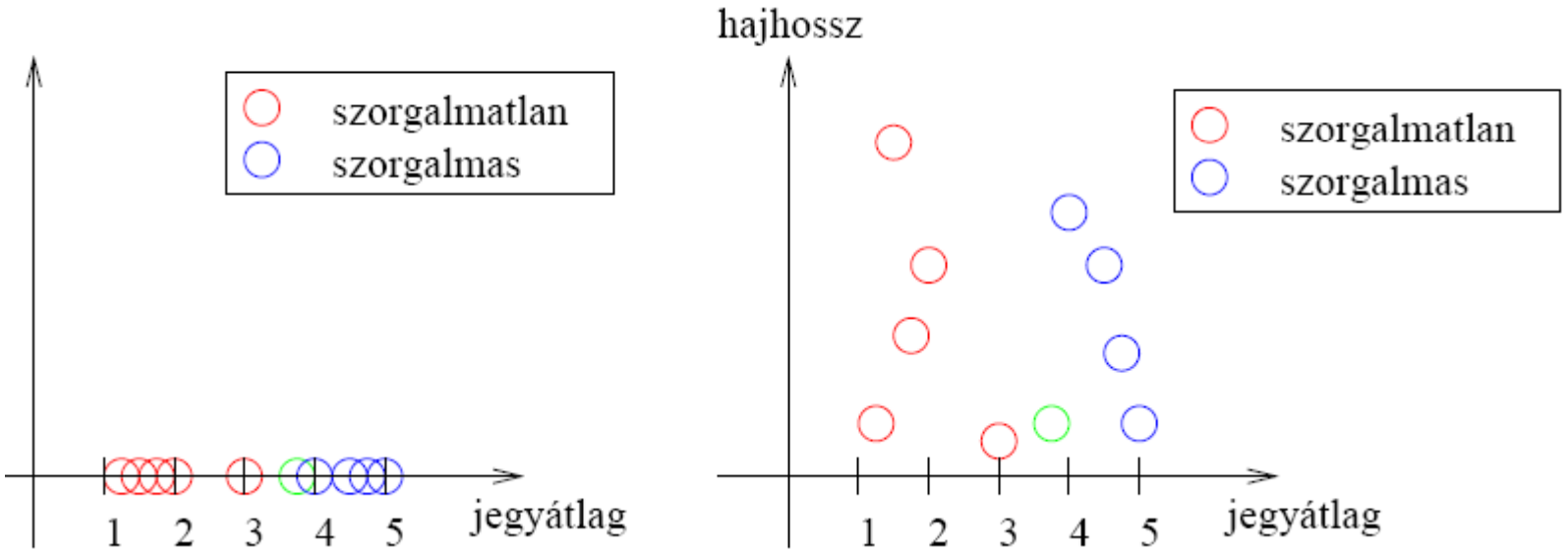
  Sok tanítópontra van szükség.

„Dimenzióátok”

- Probléma:
  - Tetszőleges környezetben elég pont kell a jó működéshez
  - Rögzített sűrűség mellett a szükséges tanítópontok száma exponenciálisan nő a dimenzióval -> Túl sok pont kéne
- Megoldás:
  - Korlátozni kell a felhasznált dimenziókat
  - Ki kell szűrni az eredményt nem, vagy nem jelentősen befolyásoló attribútumokat (független attribútumok kiszűrése)

Probléma/2

- Független attribútumokra való érzékenység:

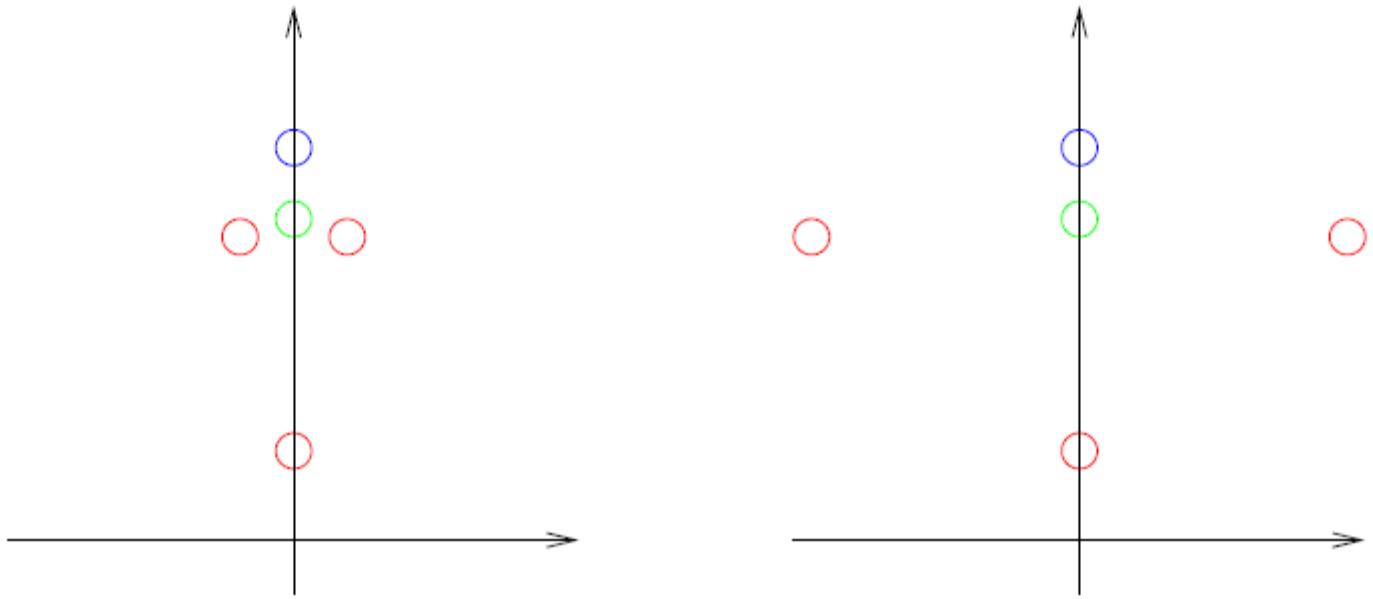


Megoldás/2

- Több tanítópont alkalmazása
- Szakértők bevonása (megmondják mik a független az adott attribútum esetén)
- Statisztikai tesztek függetlenség eldöntésére
- Ha az attribútumok száma nem túl nagy: az összes részhalmazra elvégezni a regressziót, majd a legjobbat kiválasztani
- Ha túl sok attribútum van: bővítés, csökkentés módszere (addig veszünk/hagyunk el 1 attribútumot, amíg javul a teljesítmény)

Probléma/3

- Mértékegységre való érzékenység (nincs skála-invariancia)
- Mértékegységeket rögzíteni kell.
- Távolságfüggvény módosításával elérhető az attribútumok közötti „fontossági sorrend” kialakítása



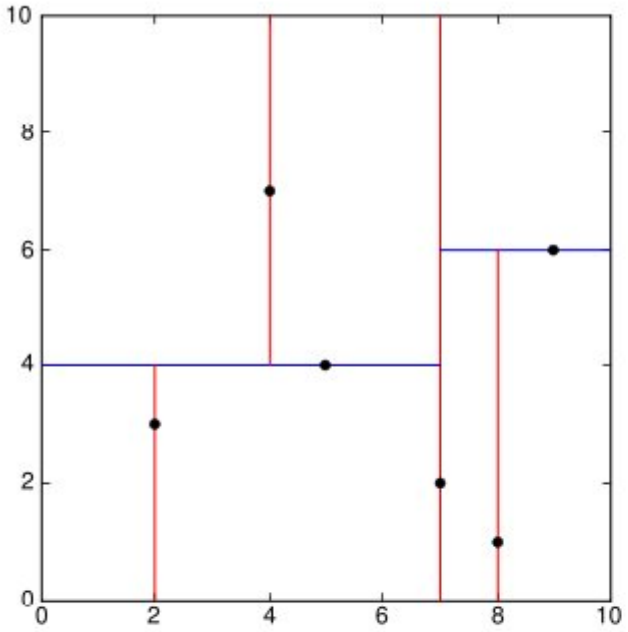
Algoritmus futás ideje

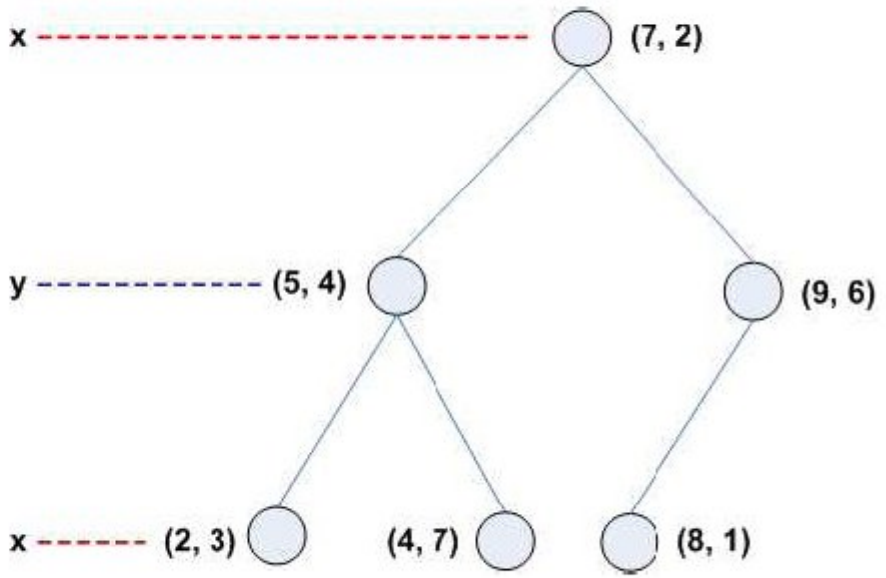
- Legyen k<<n, így a legközelebbi k pont keresés nem tart sokkal tovább, mint a legközelebbi.
- Lineáris keresés legközelebbi pontra:
  - N tanítópont esetén O(N) lépés.
- Rendezett pontok esetén:
  - Bináris keresés: O(log(N)) lépés pontonként
    (A rendezés az elején O(N*log(N)) lépés)
  - Csak egy dimenziós attribútum halmaz esetén működik
- Több dimenzió esetén:
  - KD fával

KD fa

- Minden lépésben valamelyik tengellyel párhuzamosan kettéosztjuk a ponthalmazt. A fa leveleiben tároljuk az adott osztályban található pontokat
- 2 ellentétes cél:
  - Minden lépésben közel felezzük a ponthalmazt
    - Legközelebbi pont lehet távol, de a pontok egyenletesen helyezkednek el
  - Minden kialakult hipertéglatest egy kocka legyen
    - A legközelebbi pont vagy a kockában, vagy egy szomszédos kockában van

KD-fa példa

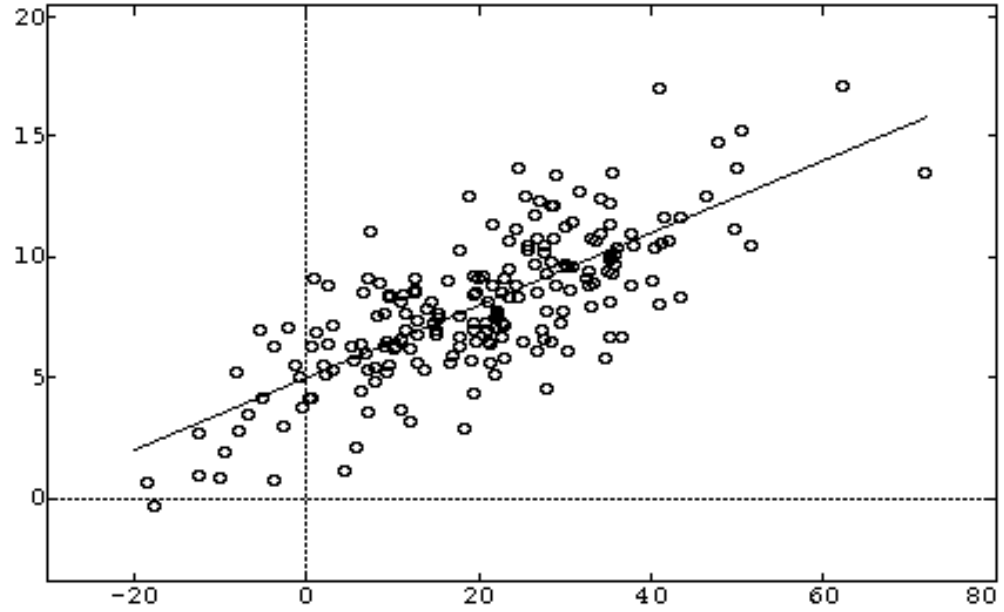




Tartományok meghatározása KD-fa építésénél (bal oldali ábra) és a KD-fa (jobb oldali ábra)

Lineáris regresszió/1

- Alapfeladat (1 dimenzió)
- Adottak pontok a síkon, keressük a hozzá legjobban „illő” egyenest. (Ahol a pontok négyzetes hibája minimális)



Lineáris Regresszió/2

- Általánosan (dim(X) > 1)
- Tegyük fel, hogy X_i és Y között lineáris kapcsolat van:

$$\widehat{Y} = \widehat{w}_0 + \sum_{j=1}^{n} X_j \widehat{w}_j$$

- Ez vektoros formában is felirható (X_0 := 1):

$$\widehat{Y} = X^T \widehat{w},$$

- A négyzetes hiba, amit minimalizálni akarunk:

$$\sum_{i=1}^{|\mathcal{T}|} (y_i - x_i^T w)^2.$$

Lineáris Regresszió/3

- A hiba felírható az alábbi formában:

$$(\mathbf{y} - \mathbf{X}w)^T(\mathbf{y} - \mathbf{X}w)$$

- Ennek w szerinti deriváltja:

$$-2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}w$$

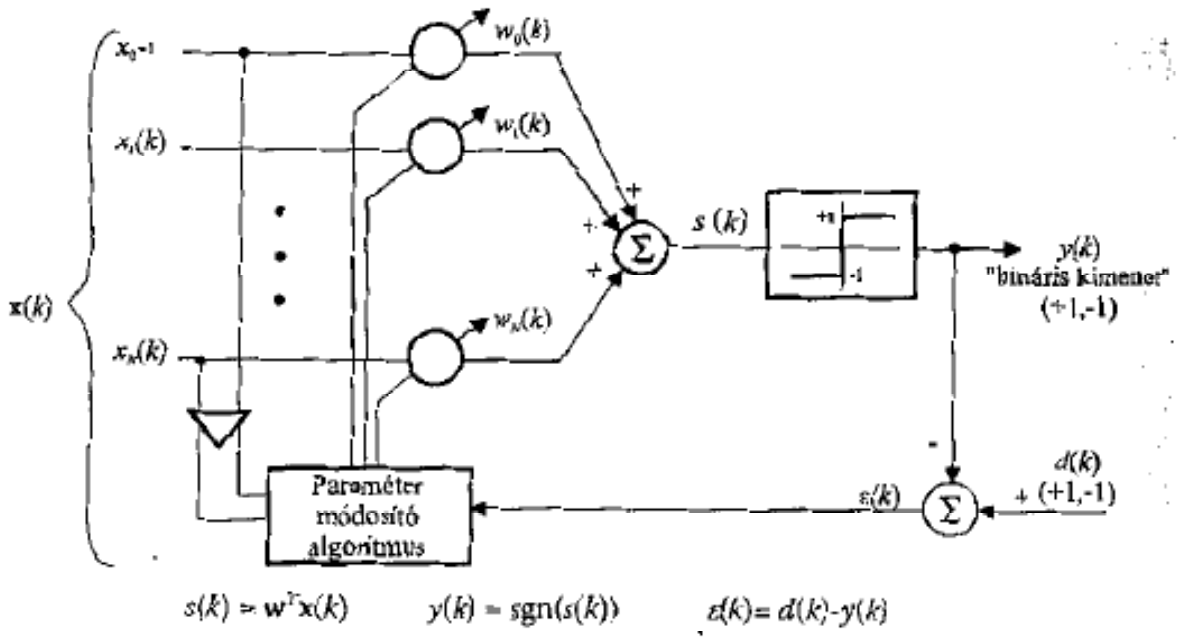
- Ahol a derivált 0, ott minimum hely van (maximum nem lehet), azaz a hiba minimális, ha:

$$\widehat{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

(feltételeztük, hogy X^TX nem szinguláris)

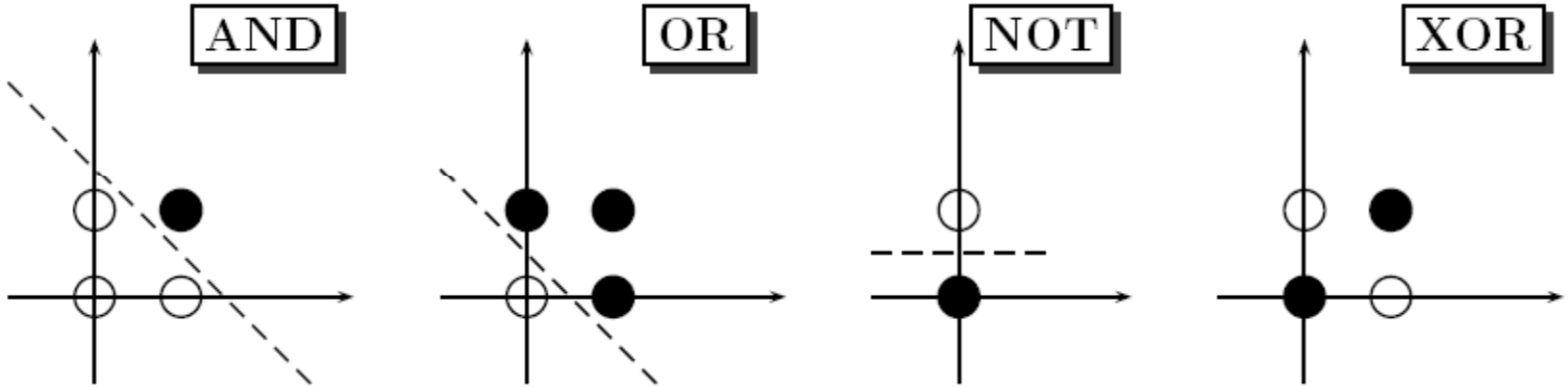
Perceptron

- Tetszőleges lineáris szeparációt meg lehet valósítani
- Tanítása: iteratívan, „hiba visszaterjesztéssel”
- Felépítése:



Lineáris szeparálhatóság

- És, vagy, tagadás logikai függvények lineárisan szeparálhatók, de XOR már nem:



- Tetszőleges logikai függvény kifejezhető a tagadás, vagy (illetve és) függvényekkel.

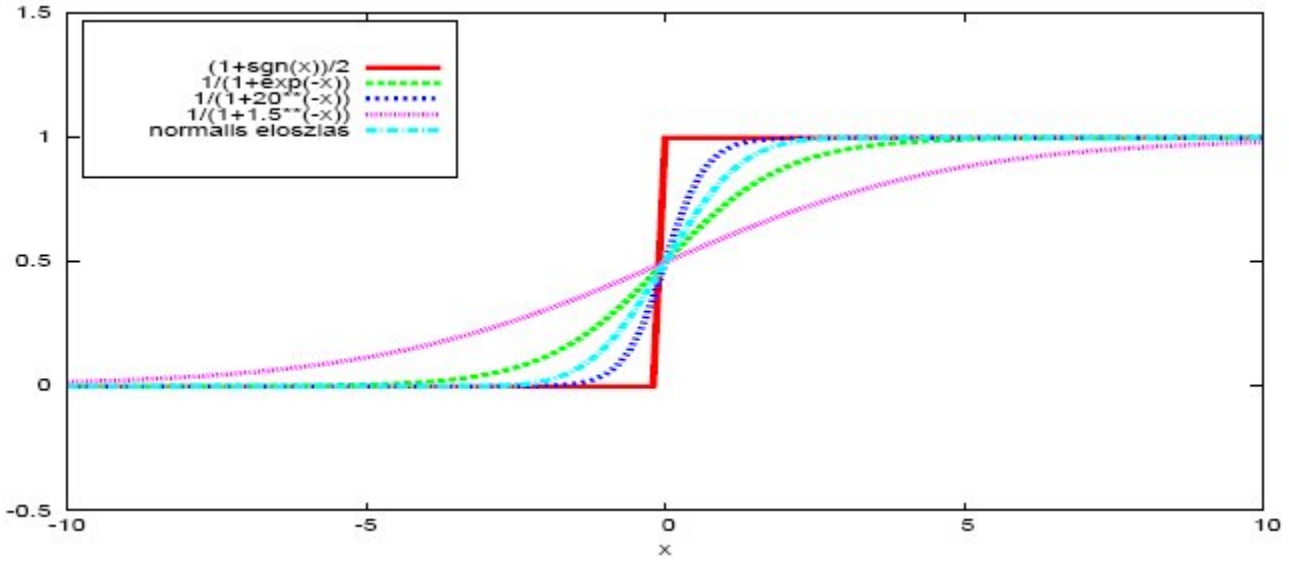
  Például: X_1 XOR X_2 == $(x_1 \vee x_2) \wedge \overline{(x_1 \wedge x_2)}$.

Többrétegű hálózatok

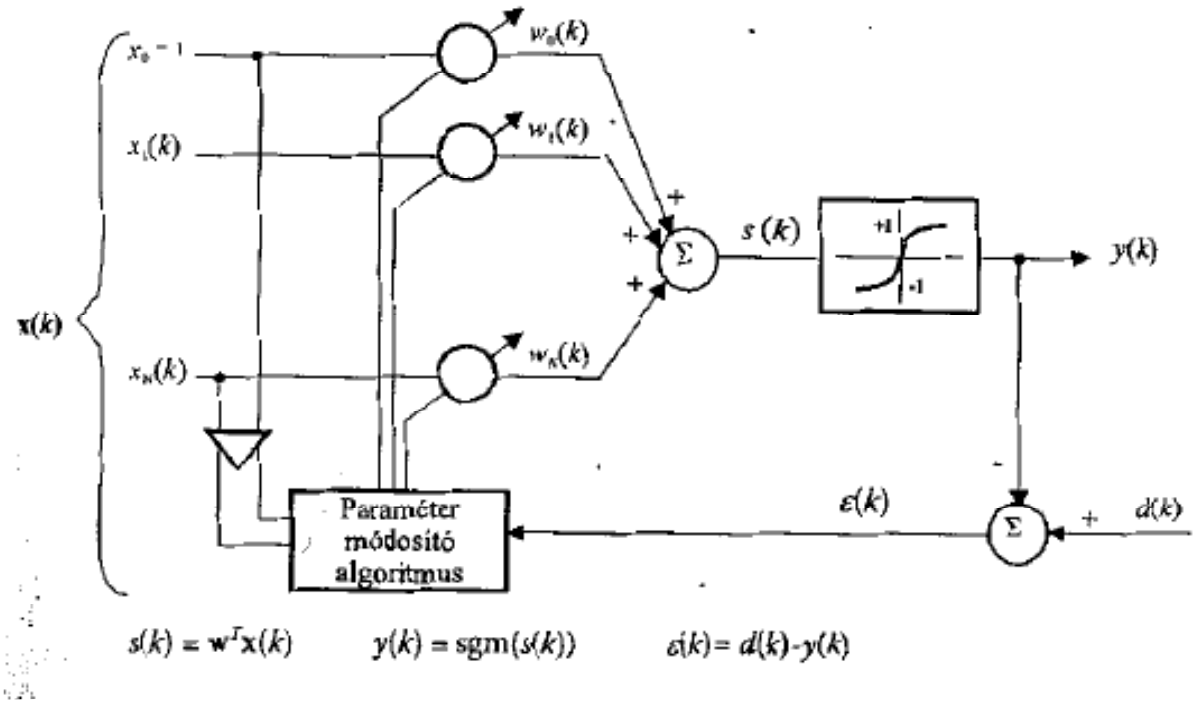
- Több rétegű perceptron tehát tetszőleges logikai függvényt képes leírni.
- Olyan módszer kell, amivel a súlyok megfelelően beállíthatók (taníthatók) a tanítópontok felhasználásával.
- Szignum függvényt használva nem ismert ilyen eljárás
- Szignum helyett „szigmoid” függvényeket használva létezik hatékony módszer a súlyok beállítására (tanulásra)

Szigmoid függvény

- Folytonos, deriválható, monoton, szimmetrikus (f(-x) = 1 - f(x))
- + végtelenben 1-hez, - végtelenben 0-hoz tart.



Elemi neuron felépítése



Elemi neuron tanítása

- Gradiens alapú eljárással
- Hiba:

$$\varepsilon(k) = d(k) - y(k) = d(k) - sgm(s(k)) = d(k) - sgm(w^T(k)x(k))$$

- Négyzetes hiba gradiense:

$$\frac{\partial \varepsilon^2}{\partial w} = -2\varepsilon sgm'(s)x$$

- Súlymódosítás a gradiens eljárás szerint:

$$w(k+1) = w(k) + 2\mu(k)\varepsilon(k)sgm'(s(k))x(k)$$

Neurális hálózatok

- Elemi neutronok által alkotott többrétegű hálózatok
- Belátható, hogy az így kapott hálózat univerzális approximátor
- A súlytényezők beállíthatók a negatív gradiens módszert használva. (Az eljárás gyakorlati problémák esetén konvergens)