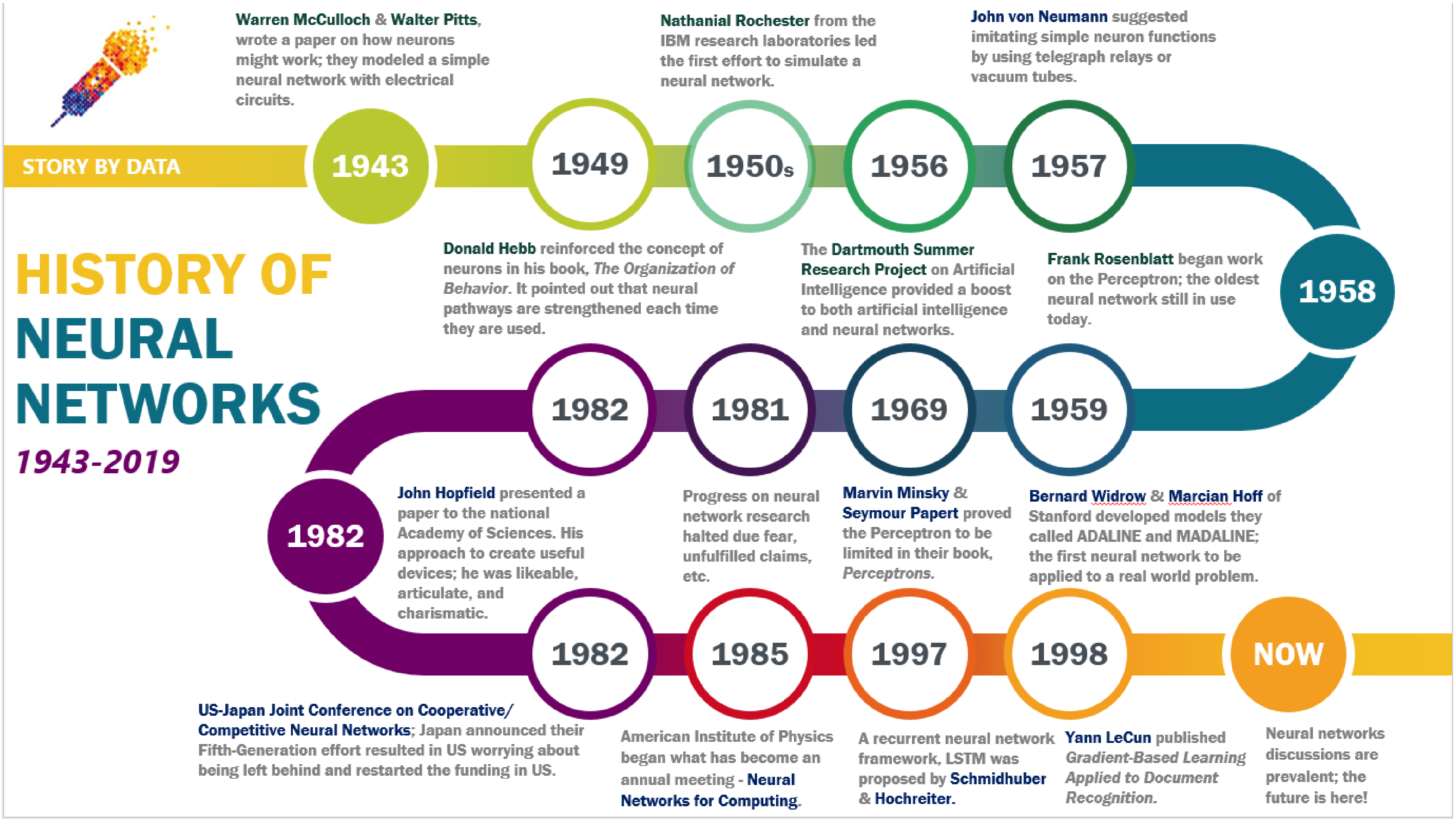
Mesterséges neurális hálózatok

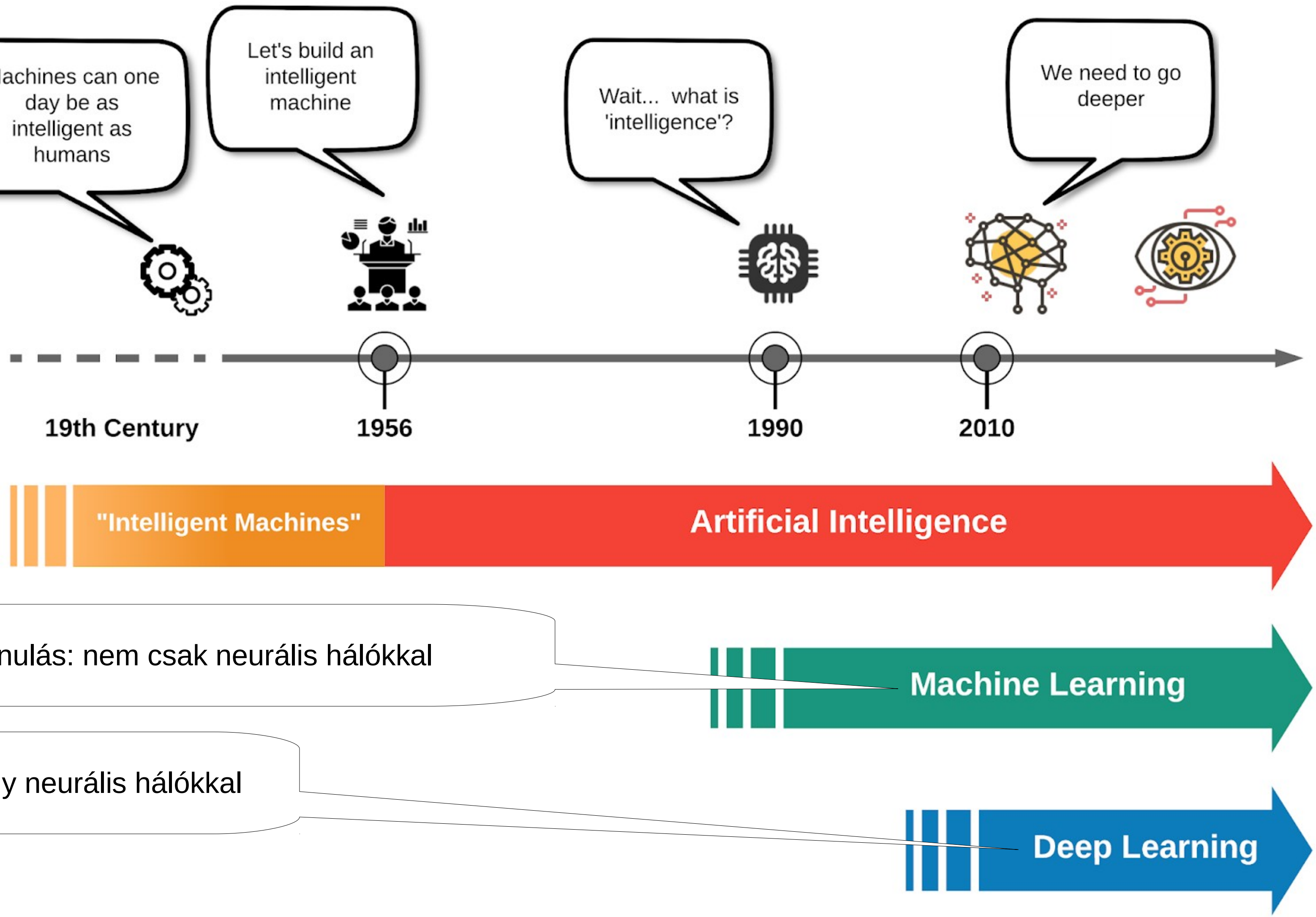
Benyó Balázs, BME
bbenyo@iit.bme.hu

Mesterséges neurális hálózatok:
Neurális hálózatok története

Történeti áttekintés a kezdetektől



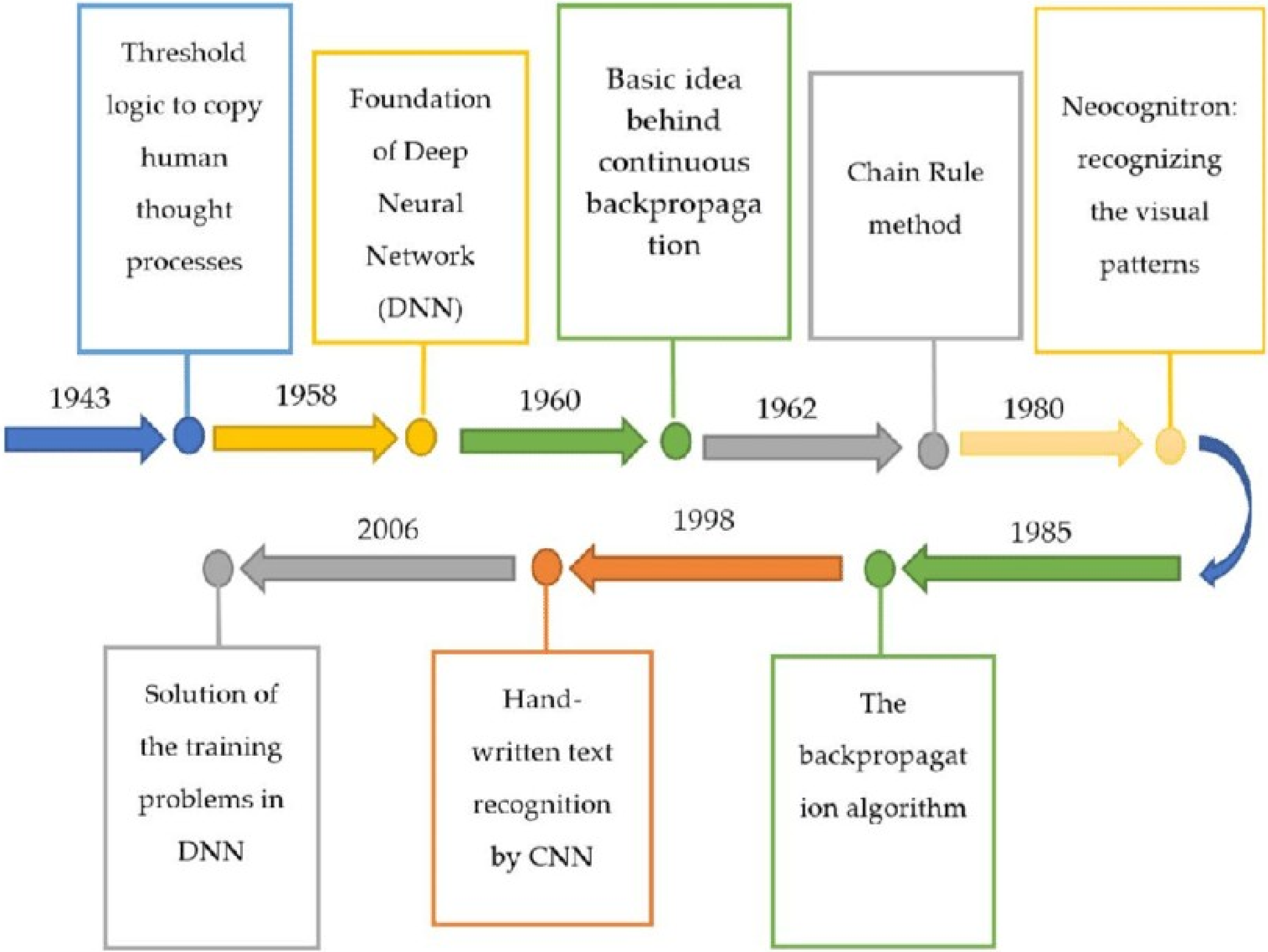
Fontos fogalmak



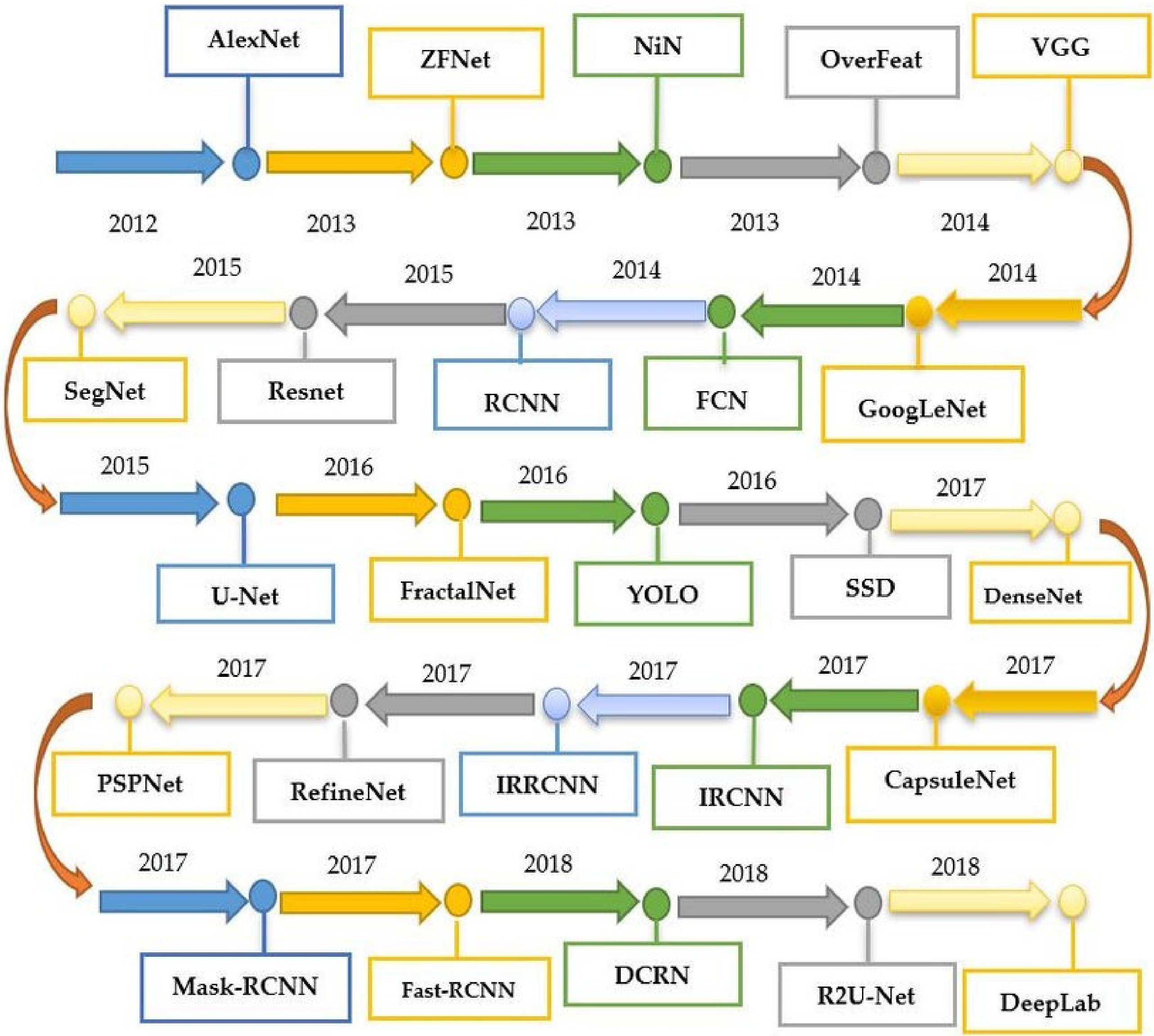
Gépi tanulás: nem csak neurális hálókkal

Tanulás mély neurális hálókkal

Fontos módszerek neurális hálózatok alkalmazásához



Konkrét neurális háló típusok megjelenési ideje a közelmúltban

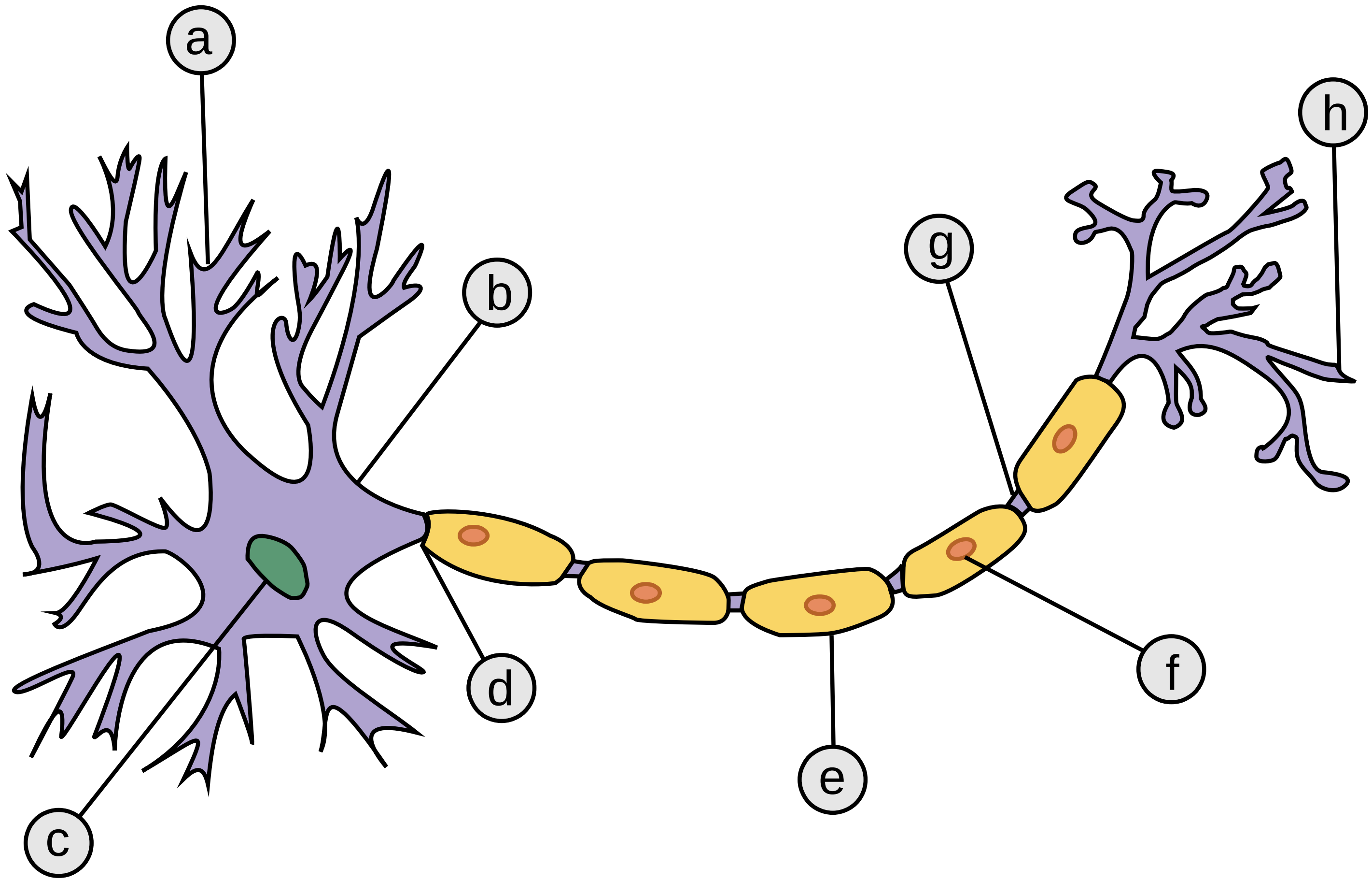


Mesterséges neurális hálózatok: **Fiziológiai analógia**

Fiziológiai analógia

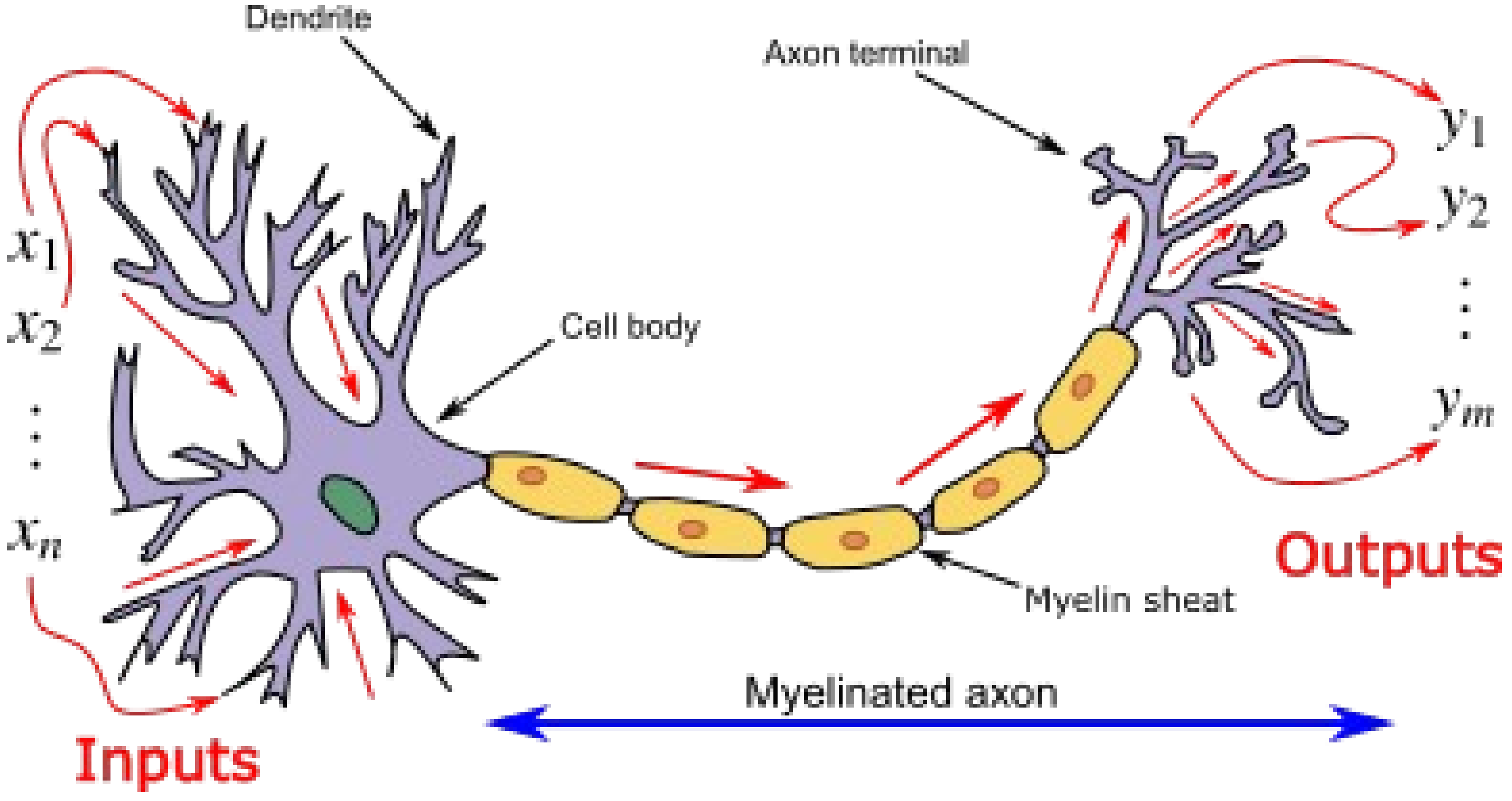
- **Analógia alapja:**
  - **Idegsejtek** hálózata
  - **Idegsejt** (neuron) sematikus **működése**:
    - Képes **ingerület átvitelre**, ill. továbbítására
    - Idegrendszer: agy, gerincvelő, idegdúcok
    - **Idegsejt** felépítése/működése:
      - Sejtmag, sejttest
      - Neuronok **kapcsolódása**:
        - **szinapszis** (synapse) keresztül (kémiai/elektromos)
        - Központi idegrendszer: nagyszámú kapcsolat (kimenet: ~10^3, bemenet: ~10^4)
      - **Dendrit**: beérkező ingerületet a sejttest felé továbbítja
      - **Axon**: hosszú nyúlvány
        - Ingerület továbbítása a környező neuronok felé
      - **Ingerület**: szétterjedő potenciál változás (akciós potenciál) a sejtmembránon

Gerincvelői mozgató neuron vázaltos képe

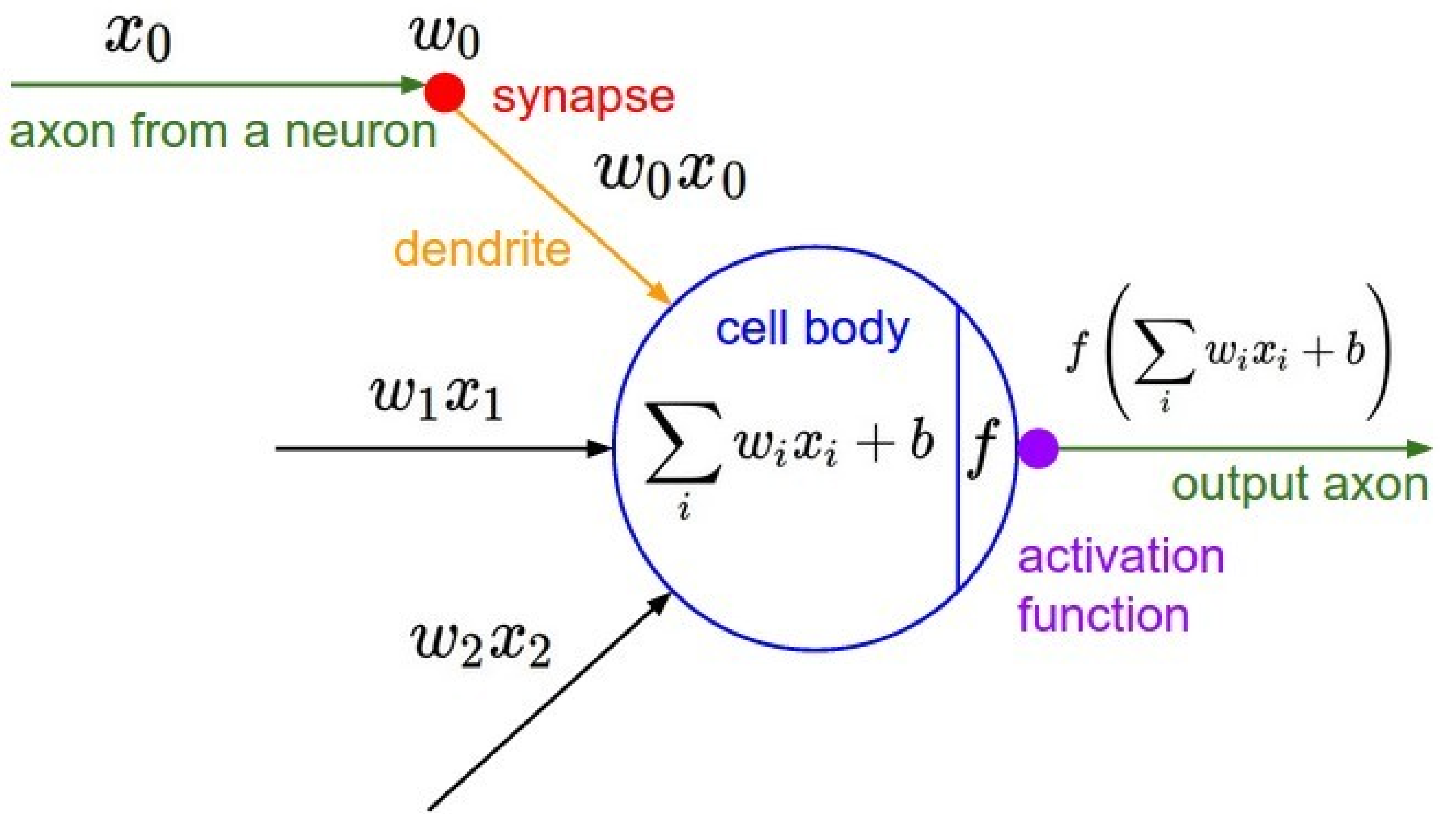


a. dendrit, b. sejttest, c. sejtmag, d. axondomb, e. Swann-hüvely, f. Swann-sejt (mag), g. Ranvier-féle befűződés, h. végfácska (telodendrion) elágazódás

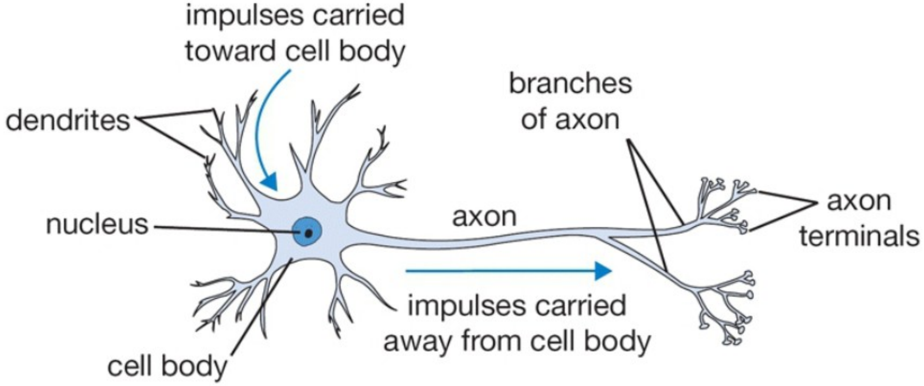
Idegsejt működésének analógiája

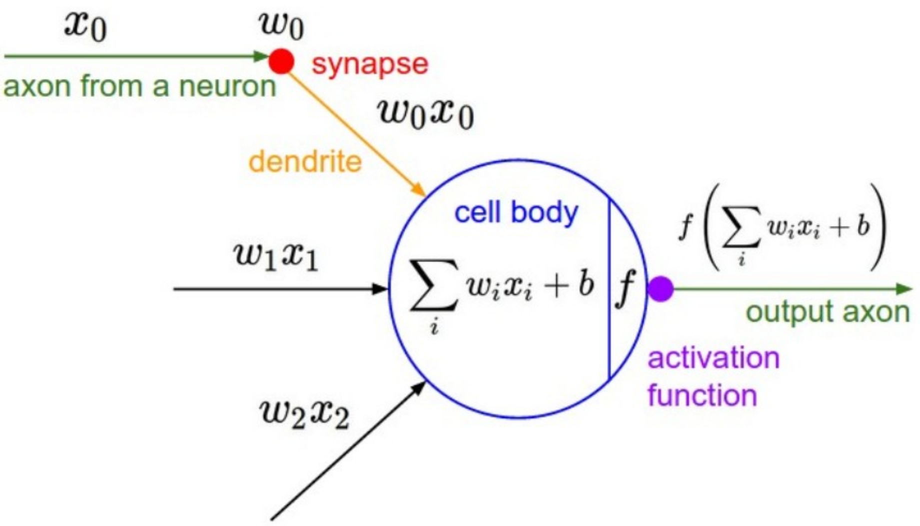


Neurális hálózat egy csomópontjának működése



Neurális hálózat egy csomópontja





https://cs231n.github.io/neural-networks-1/

Mesterséges neurális hálózatok: **Alapfogalmak**

Neurális hálózat felépítése

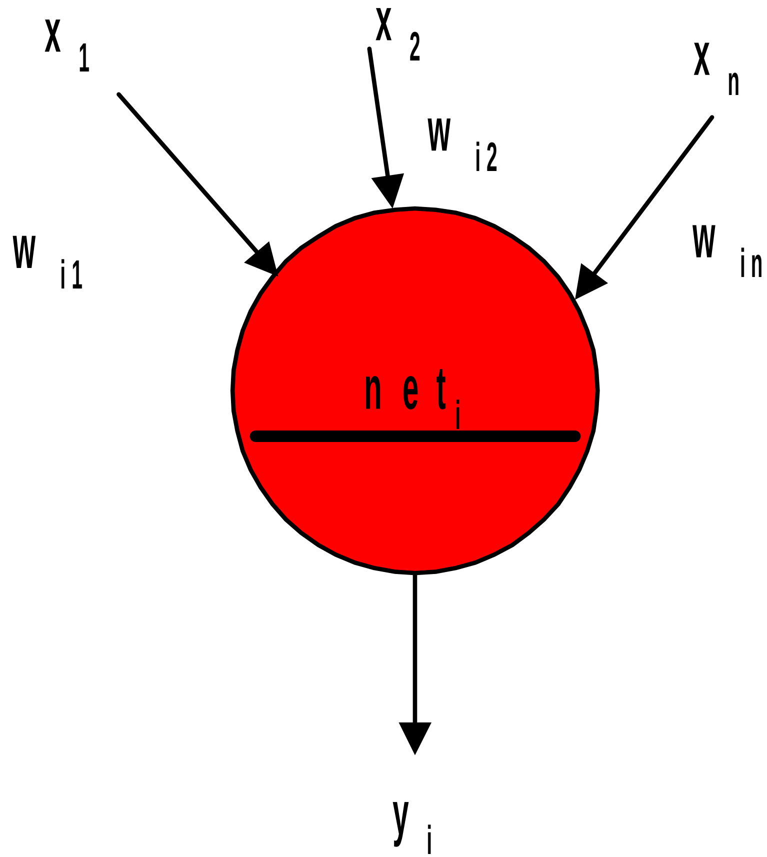
- **Gráf modell:** *irányított, súlyozott gráf*
  - **Csomópontok** (~*idegsejtek*)
  - **Irányított élek** (~*idegsejtek kapcsolatai, jelátvivő csatornák*)
  - **Súlyok** *(~neuronok kapcsolatának erősségére*)

Egy általános csomópont jelátvitele

- **Bejövő jelek** erőssége: x_j
- **Súlyok**: w_{ij}
- Csomópontba **belépő jel**: net_i
  - súlyozott összeg

$$net_i = \sum_{j=1}^{n} w_{ij} x_j$$

i-dik csomópont jelátvitele



Formális leírás

- **Csomópontba belépő jel**
  - Súlyok (w_i) és a bemenetek (x) vektorainak skalár szorzata:

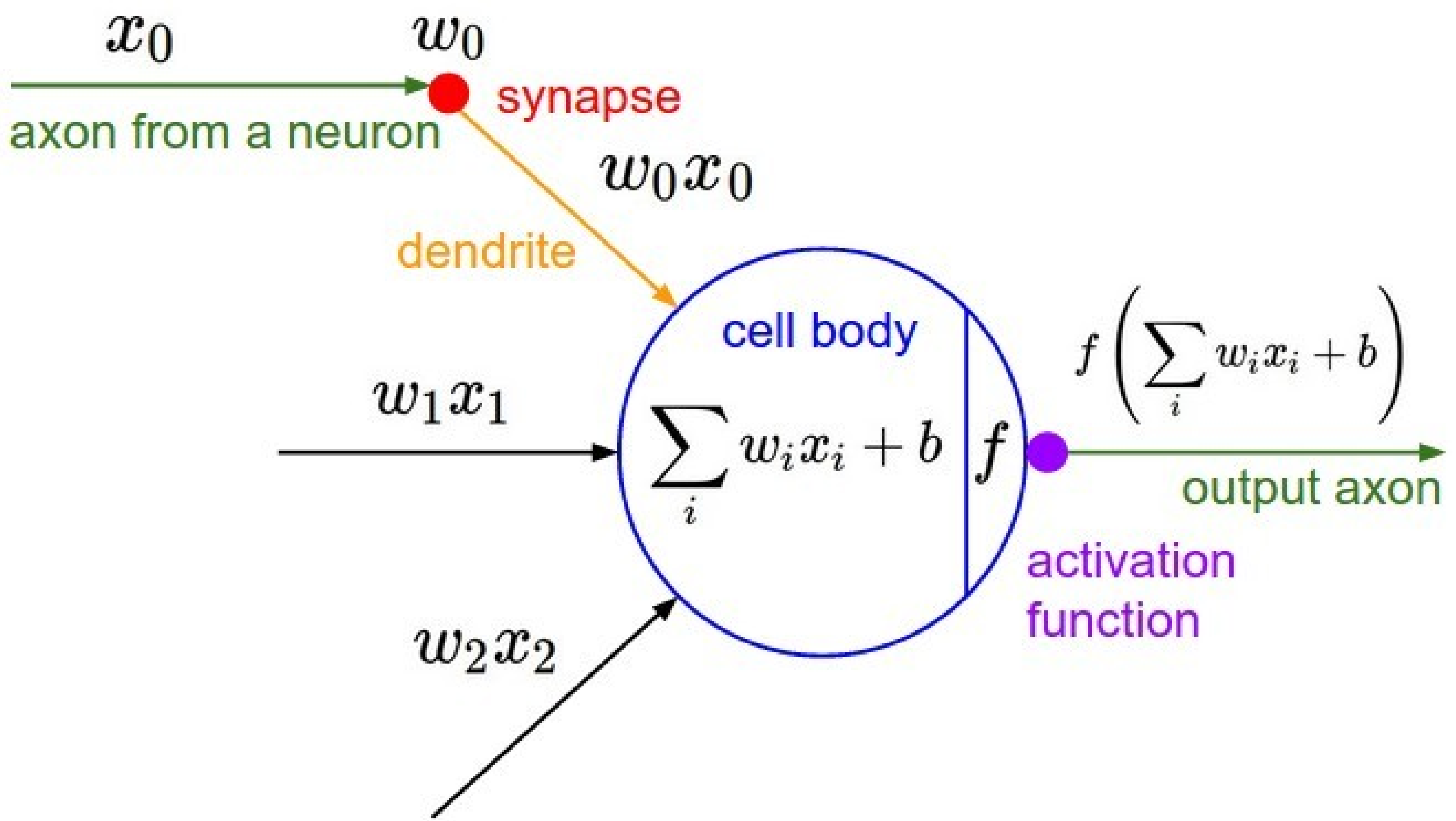
$$net_i = w_i^T x$$

- **Csomópontból kilépő jel:**

$$y_i = f(net_i) = f\left(w_i^T x\right)$$

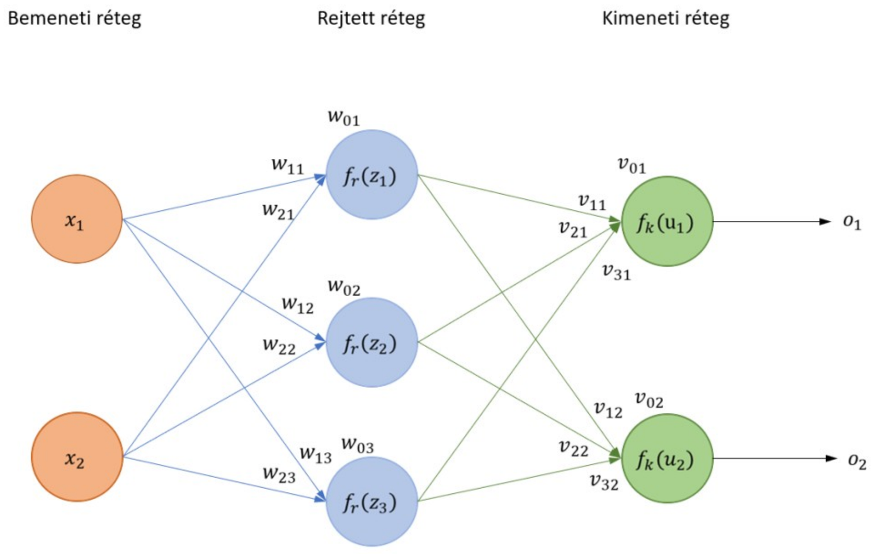
  - **Aktivációs függvény:** *f()*
- *A hálózat az* ***x*** *vektort* ***leképzi*** *az* ***y*** *vektorba*

Neurális hálózat csomópontjának általános működése

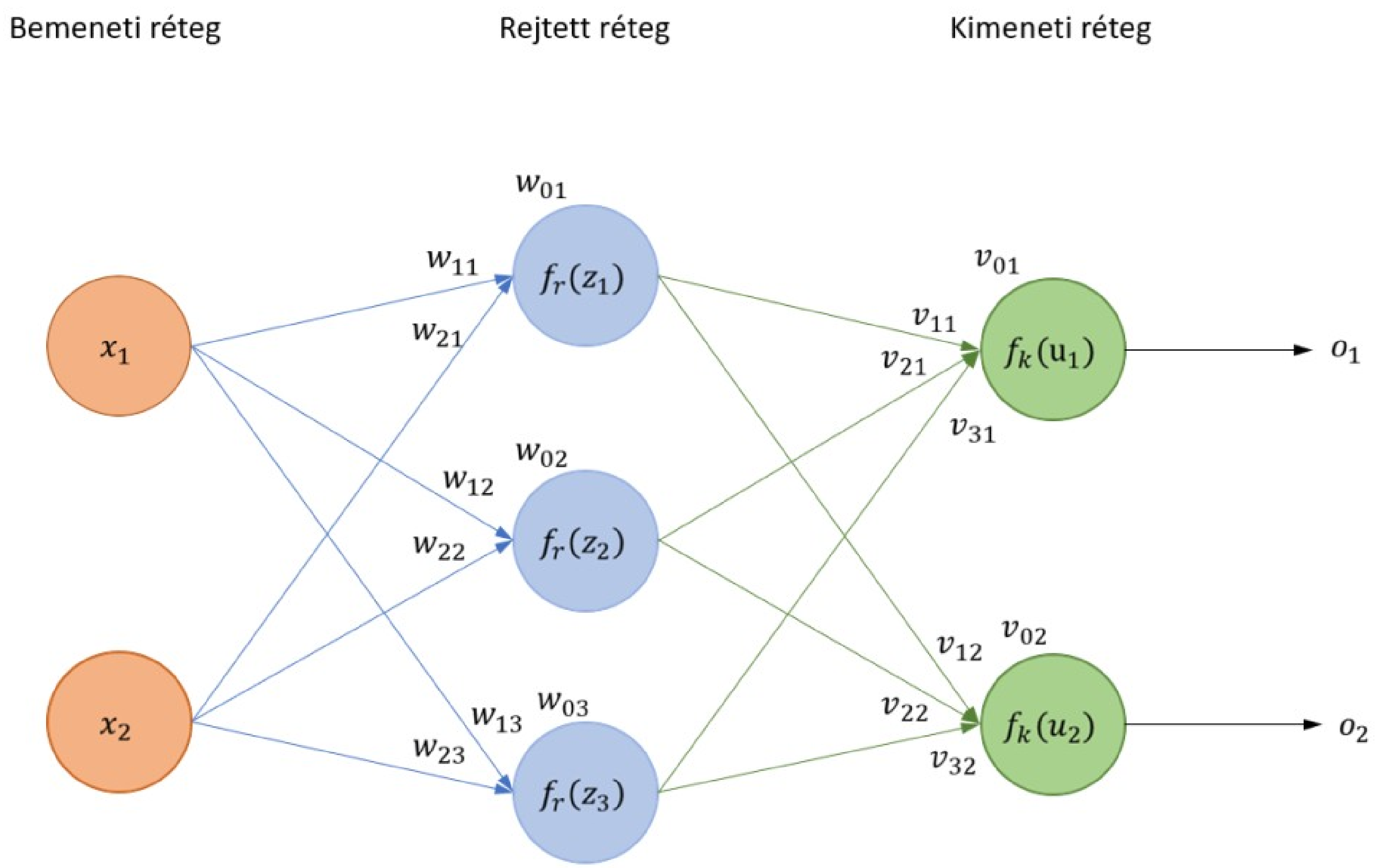


Neurális hálózat felépítése

- **Gráf modell:** *irányított, súlyozott gráf*
  - Csomópontok
  - Irányított élek (*~jelátvivő csatornák*)
  - Súlyok *(~neuronok kapcsolatának erősségére*)
- **Hálózati topológia – csomópontok rétegekbe rendezettek:**
  - Összeköttetés (*általában*) csak a **szomszédos rétegekben** levő csomópontok között (*feed-forward network*).
  - Számos más megoldás is létezik.
- **Rétegek:**
  - *Szükséges rétegek:*
    - *bemeneti réteg,*
    - *kimeneti réteg,*
  - *Opcionális réteg:*
    - *rejtett-réteg(ek)*



Három rétegű hálózat

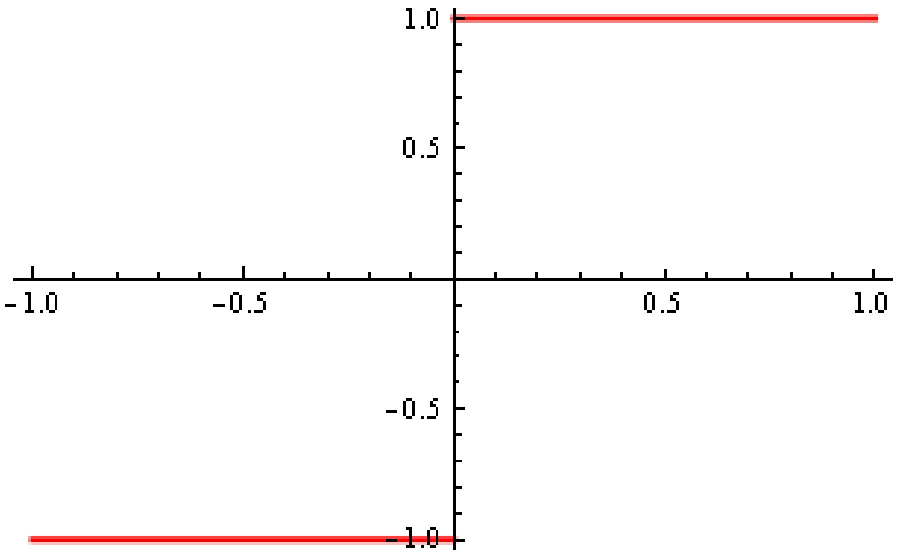


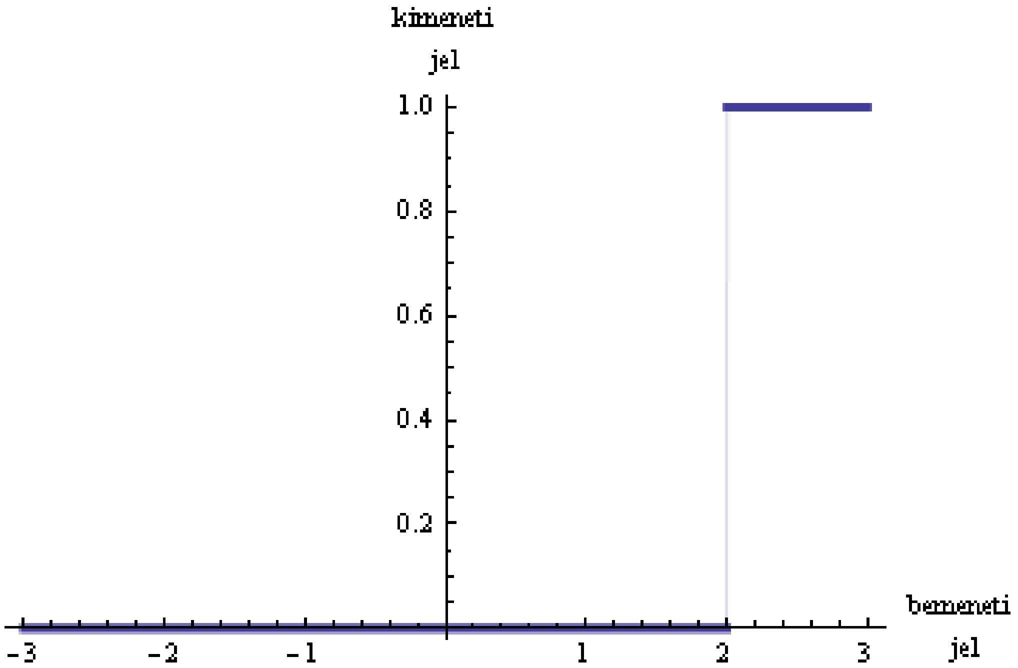
Aktivációs függvény (*f()*)

- A bemenet és a kimenet **kapcsolatának** leírása:
  - **aktivációs függvény**
  - **Nem lineáris** viselkedés meghatározása
- Egyszerű aktivációs függvény:
  - Heaviside vagy **egységugrás függvény**
- Számos más aktivációs függvényt használunk
  - Neurális háló típusától függően
- Változatok a kimeneti **értékkészletre**:
  - {-1, +1} („bipoláris kimenet”)
  - {0, 1} (bináris kimenet)

Példák aktivációs függvényre

- **Signum** függvény
  - Értékkészlet: {-1, 1}
  - Eltolás nélkül
- **Egységugrás** fv.
  - Értékkészlet: {0, 1}
  - Eltolás = 2

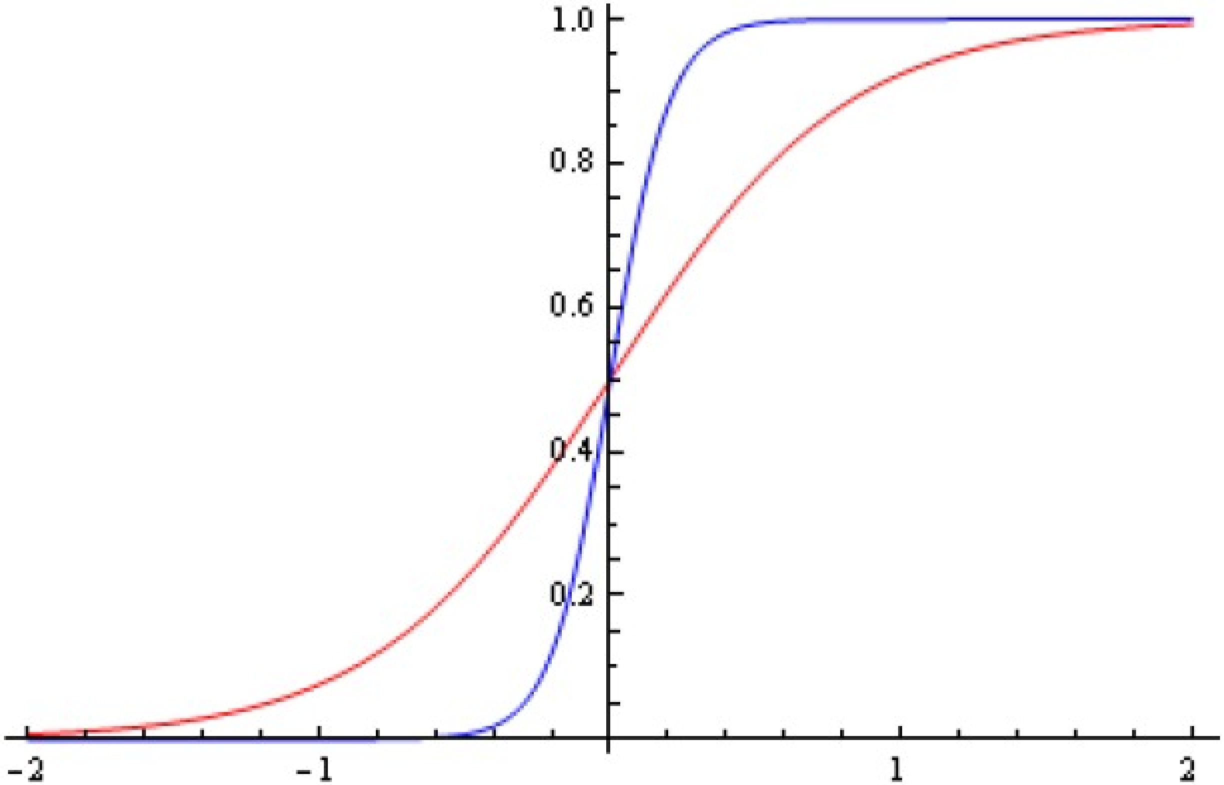




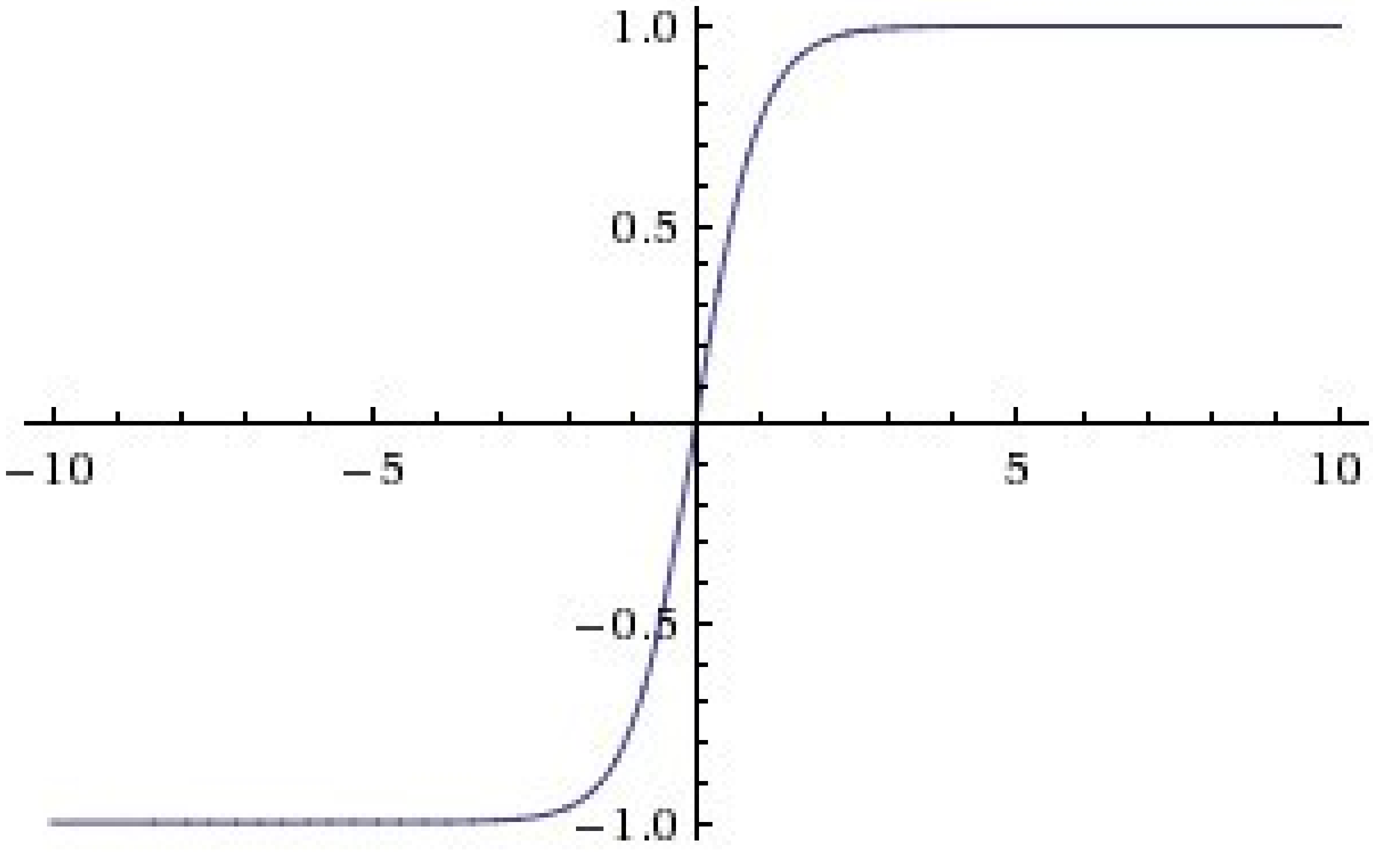
Példa folytonos aktivációs függvényre

- ***Szigmoid** függvény*
- → Folytonos kimeneti értékek előállítása

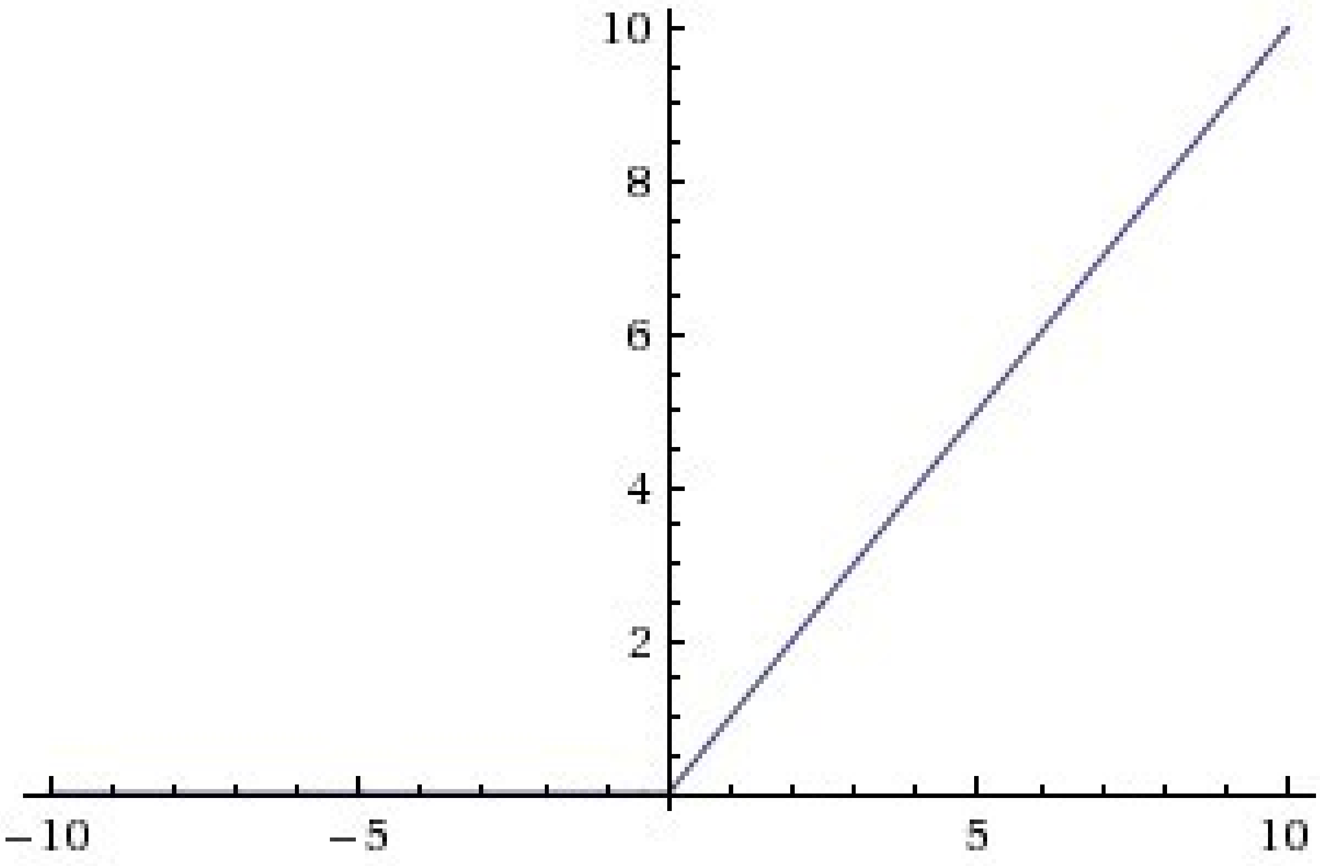
$$\sigma(x, \alpha) = \frac{1}{1 + e^{-\alpha x}}.$$



tanh()

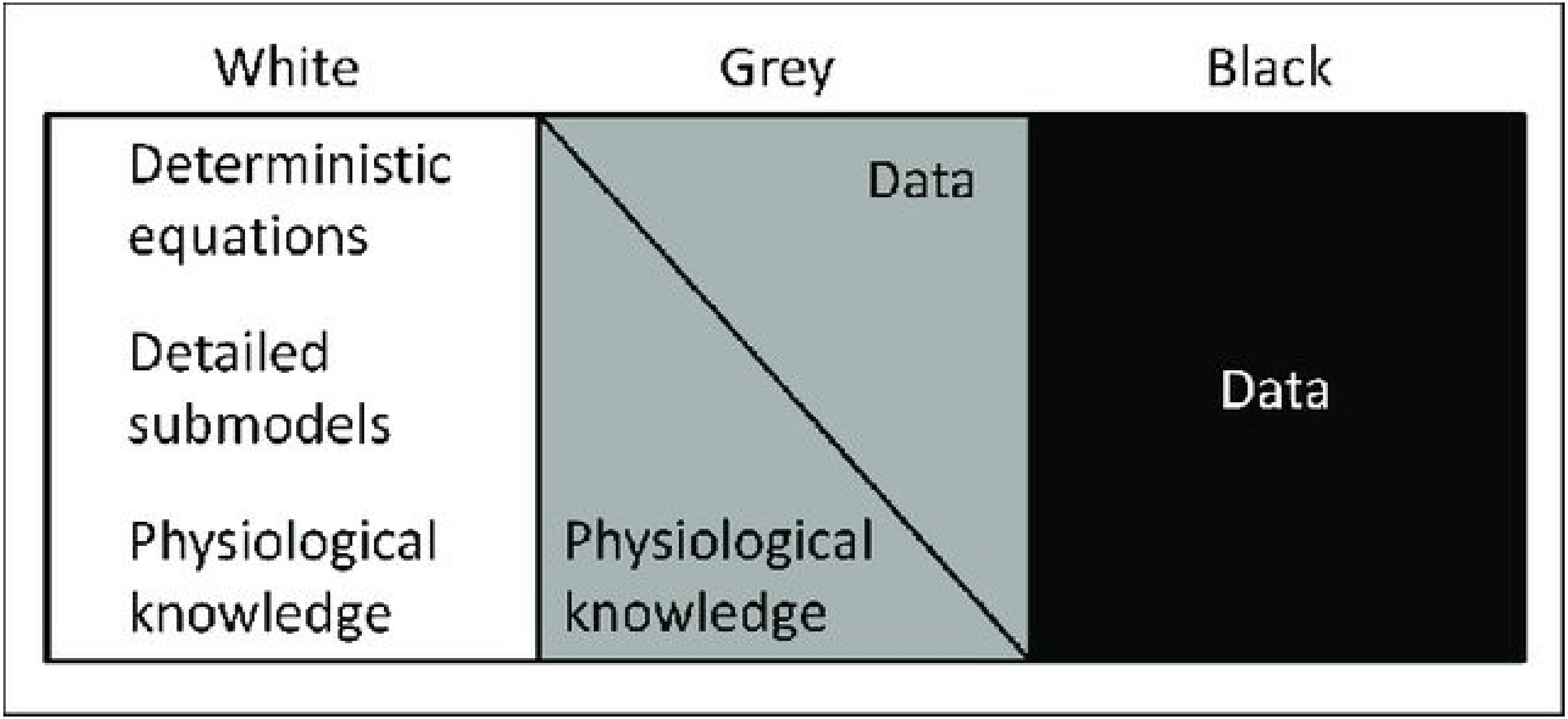


Relu

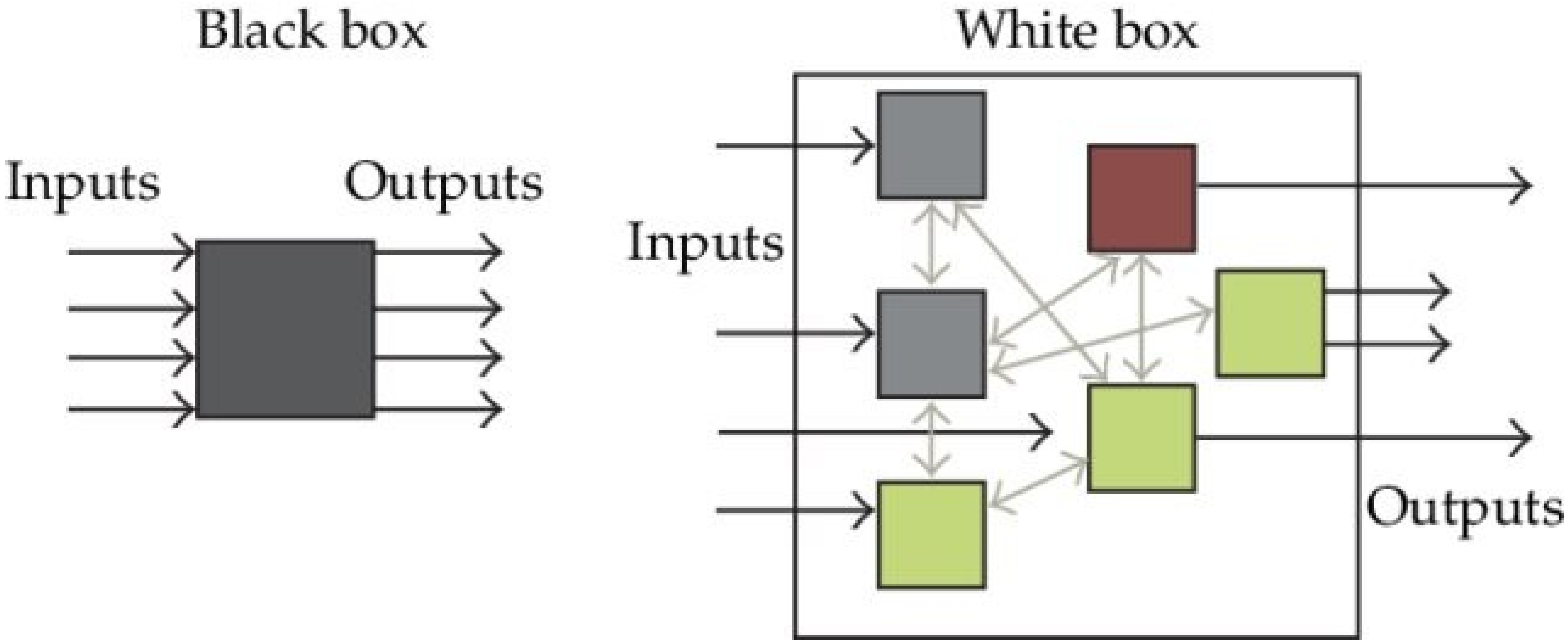


Mesterséges neurális hálózatok:
Modell típusokról általában

Modell típusokról általában



Black-box modell vs. white-box modell

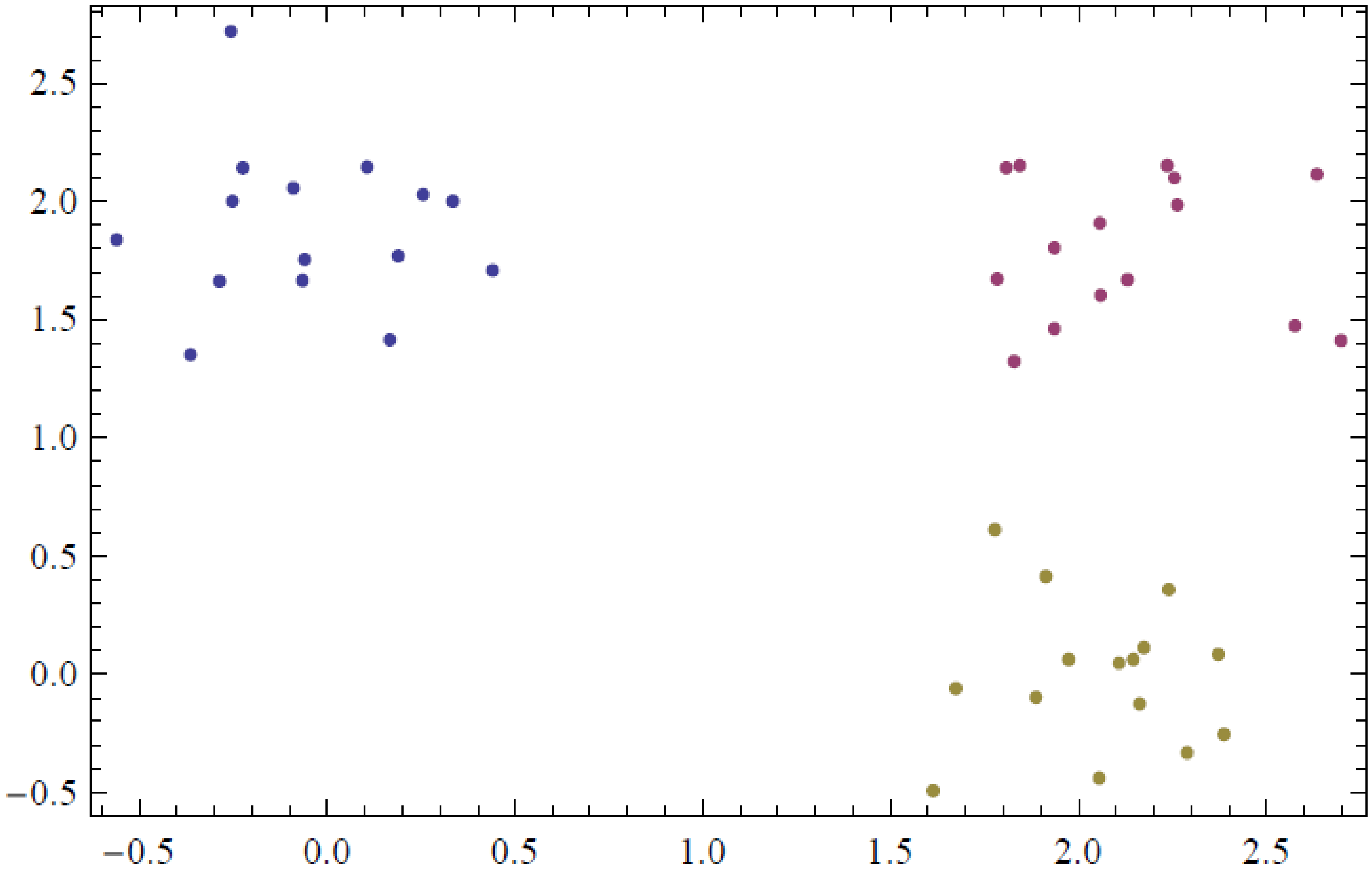


Black-box modell vs. white-box modell

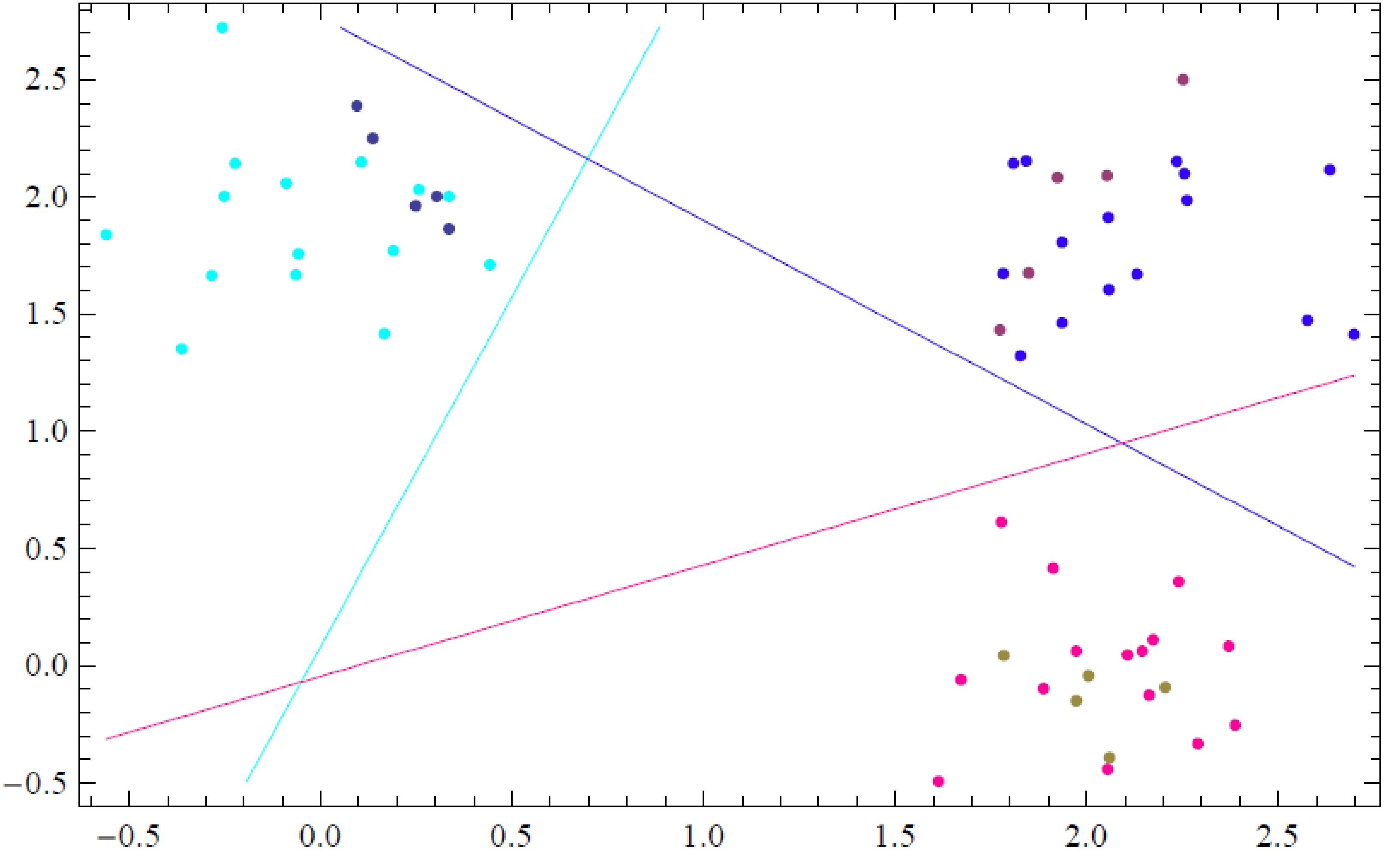
- **Black-box modell**
  - A rendszer **bemenetei** és **kimenetei** alapján határozzuk meg a modellt.
  - A modell interpretációja, a belső működés, a részrendszerek leírása **nem cél**.
  - Az esetleges modell hibák értelmezése, a rendszerre vonatkoztatható következtetések levonása, a hibák elemzése, javítása **nehéz**.
- **White-box modell**
  - A **rendszerben érvényes törvényszerűségek**, és részrendszerek **működésének ismeretében** határozzuk meg.
  - A modell elemei és a valóságos rendszer elemei közötti **összerendelés megadható**.
  - Az esetleges modell hibák értelmezése, a **rendszerre vonatkoztatható következtetések** levonása, a **hibák elemzése**, **javítása** általában **egyszerű**.

Mesterséges neurális hálózatok: **Tipikus neurális hálóval megoldható feladatok**

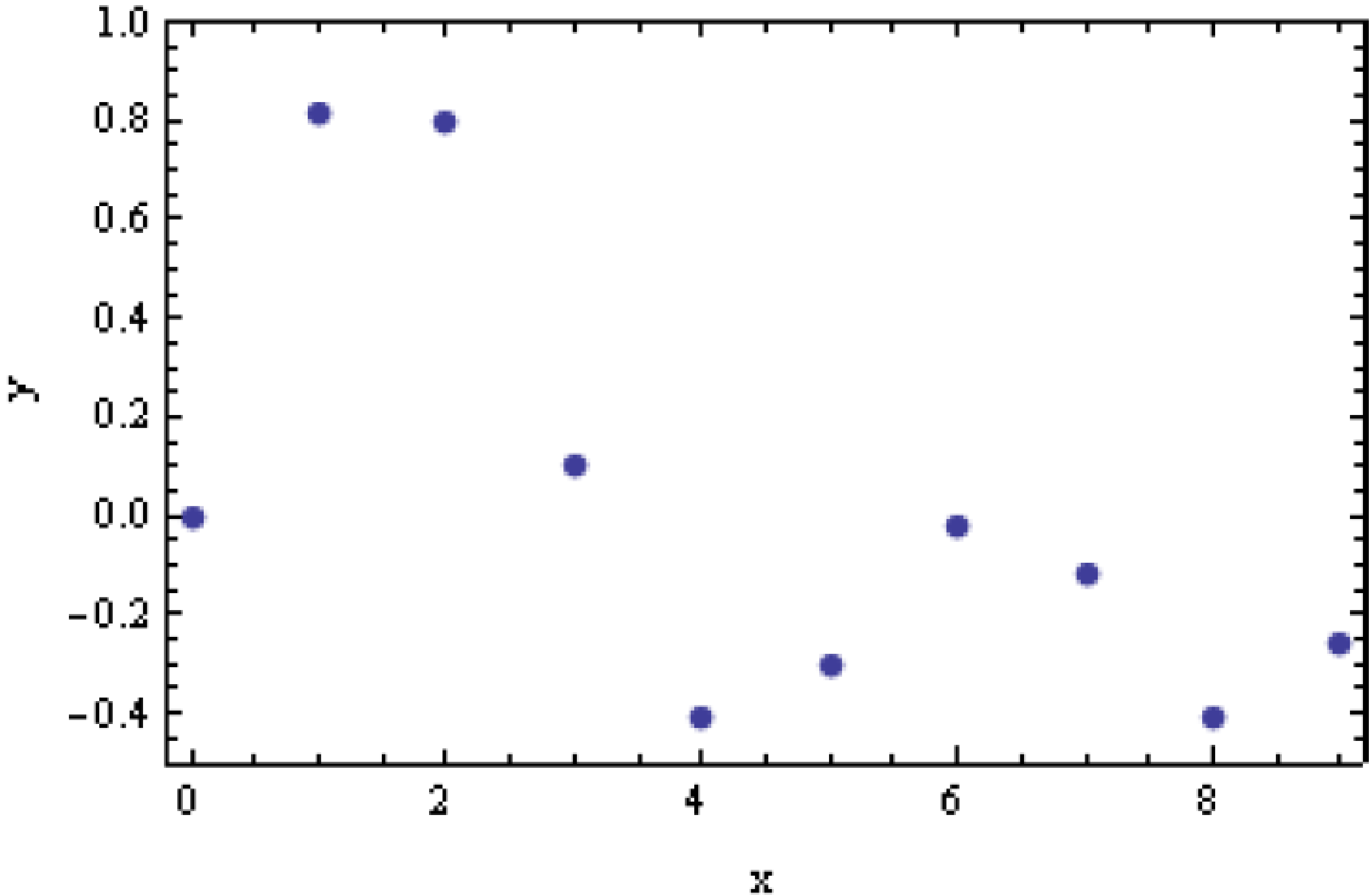
Osztályozás:
adatok osztályokba sorolása



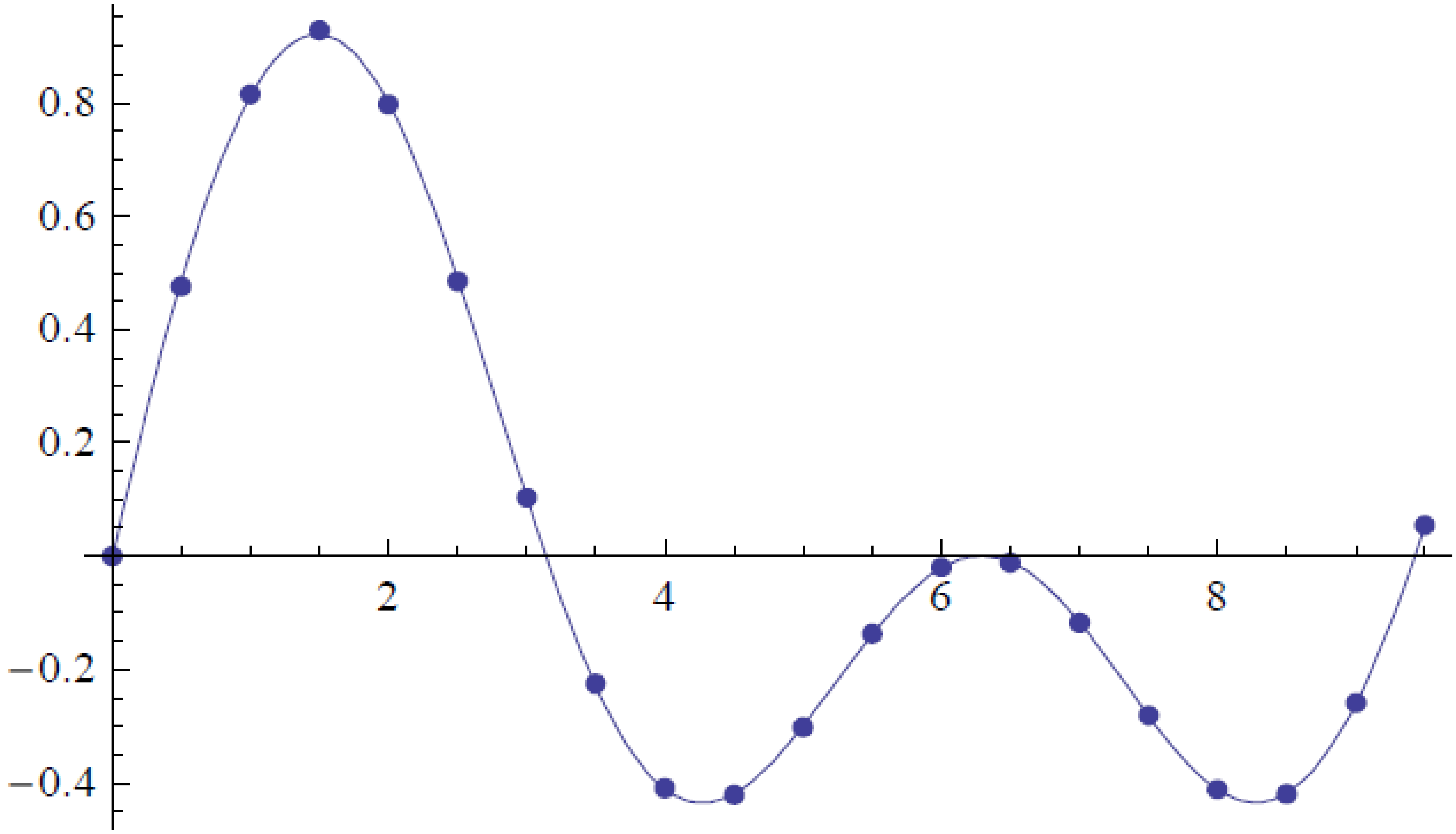
Osztályozás:
adatok osztályokba sorolása



Függvényközelítés



Függvényközelítés

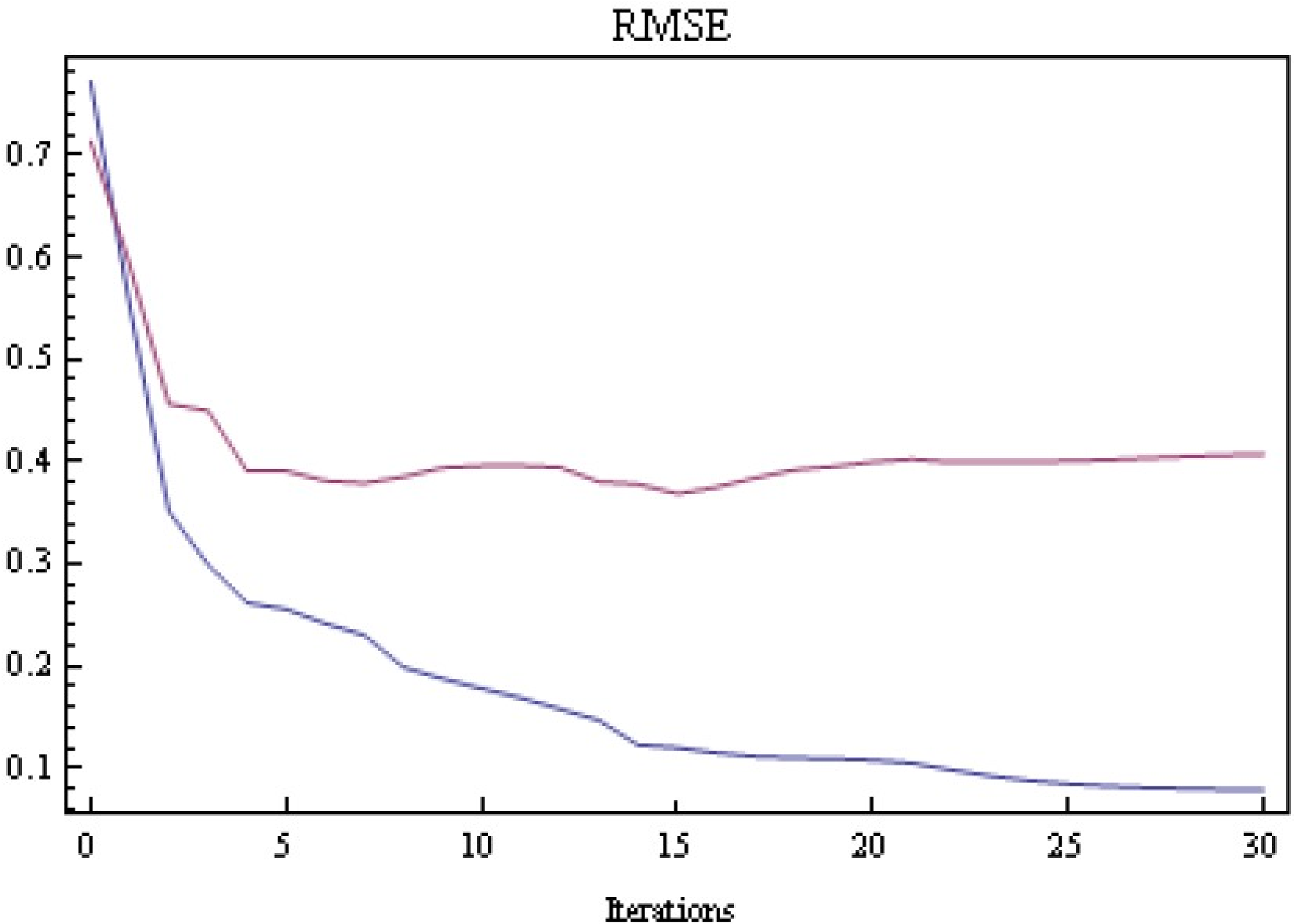


Mesterséges neurális hálózatok: **Hálózat tanítása**

Neurális hálózat tanítása

- Tanítás: **súlyok meghatározása**
  - Nagyon gyakran **iteratív** folyamat
  - Cél: teljesüljön a **kívánt bemenet/kimenet leképezés**
  - **Mintahalmaz**:
    - **Ismert** input-output párok
    - Számuk **lényegesen nagyobb**, mint a súlyok száma
    - Tanítás ~ többváltozós függvény minimalizálása
  - **Hibafüggvény:** az ismert bemenetekre az aktuális hálózat által adott válaszok és az elvárt kimenetek közötti eltérés jellemzése (mértéke):
    - Leggyakrabban alkalmazott **négyzetes hibafüggvény**:
      - a hálózat által adott és az ismert (helyes) kimenetek közötti értékek négyzetösszege
  - Tanítás elsődleges célja: a **hibafüggvény minimalizálása**

Hibafüggvény változása a tanító és a validációs halmazon - példa

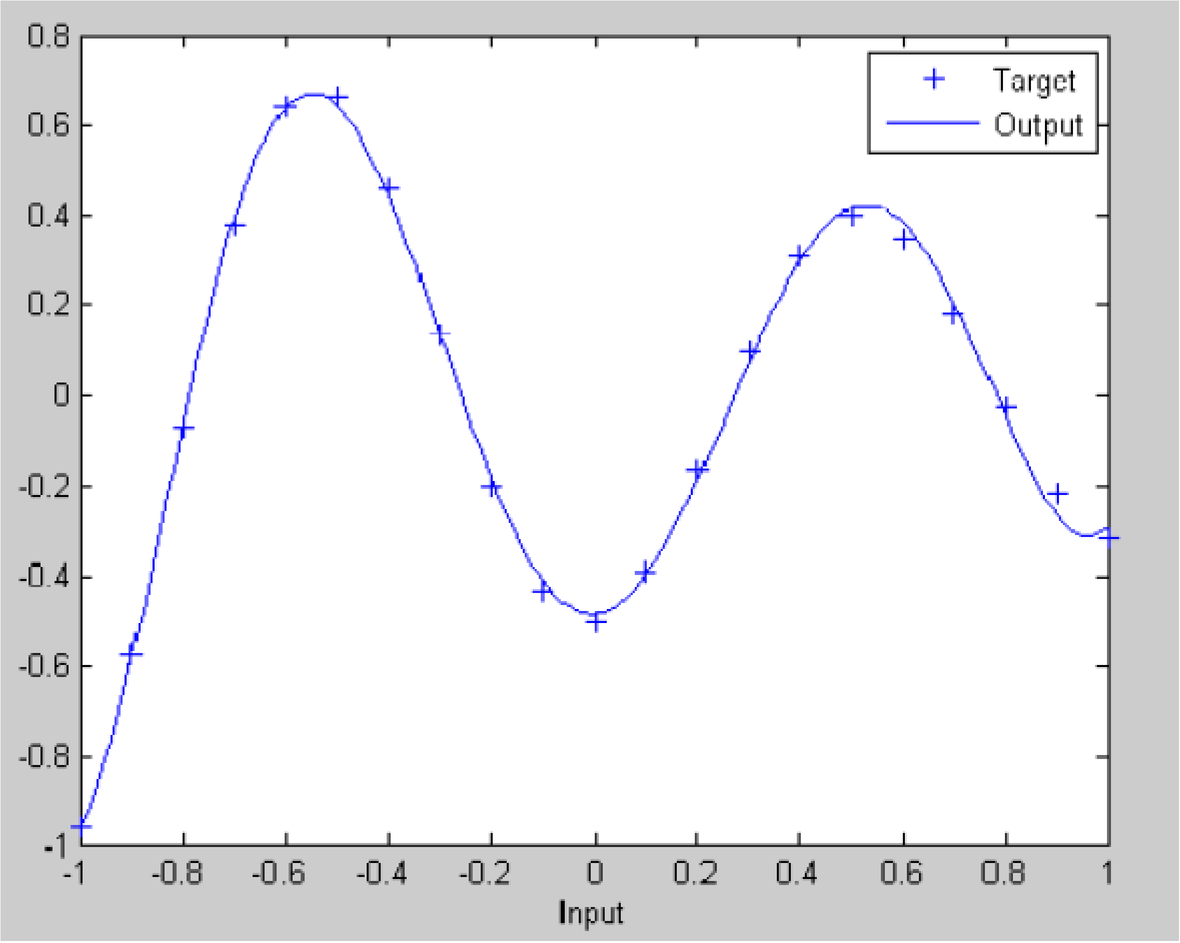


Mesterséges neurális hálózatok:
Hálózat tanításának általános problémái

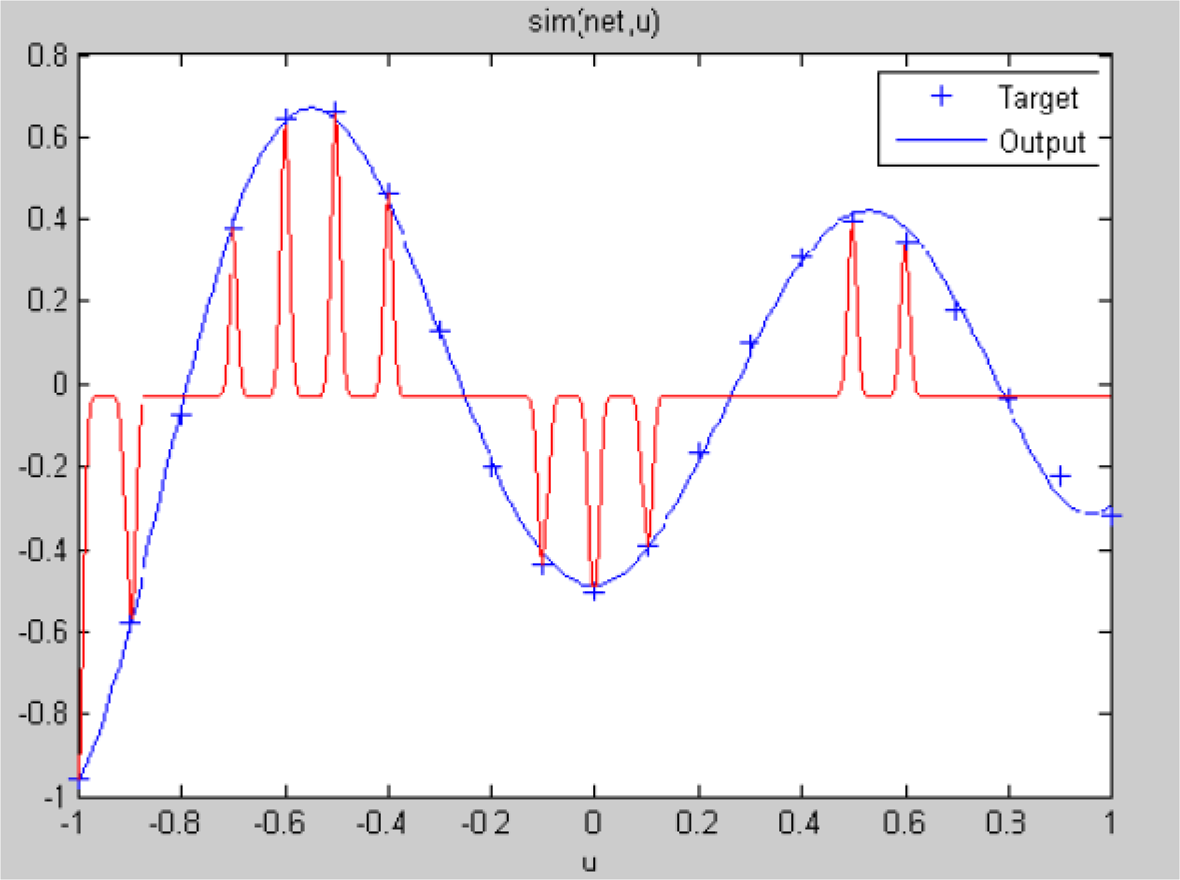
Hálózat tanításának jellemzése

- Hálózat **általánosító képessége**
  - Tanulásban **részt nem vett** bemenetekre milyen „mértékben” ad helyes kimenetet
- Tanítás lehetséges gyakori hibái
  - **Túltanulás**: az általánosító képesség ugyan csökken a tanítás során, de a hálózat „megtanulja” a tanító halmaz elemeit
  - **Alultanulás**: a hálózat által biztosított leképzés hibája további tanulással csökkenthető

Példa: Közelítendő függvény

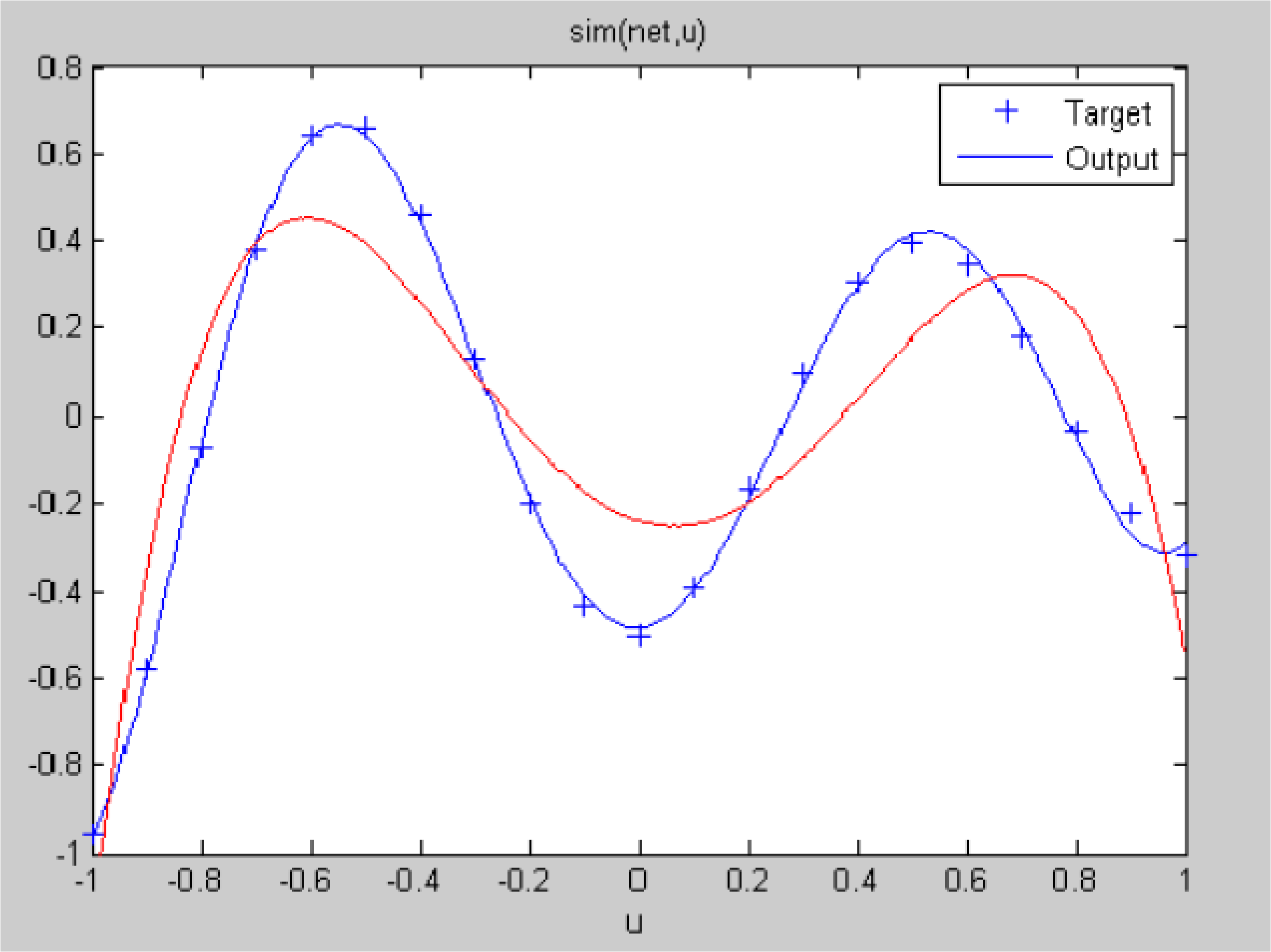


„Túltanulás” esete



Ugyan a mért hiba kicsi a tanító pontokon számolva, más pontokon vizsgálva viszont lényegesen nagyobb.
A hálózat általánosító képessége alacsony.

„Alultanulás” esete

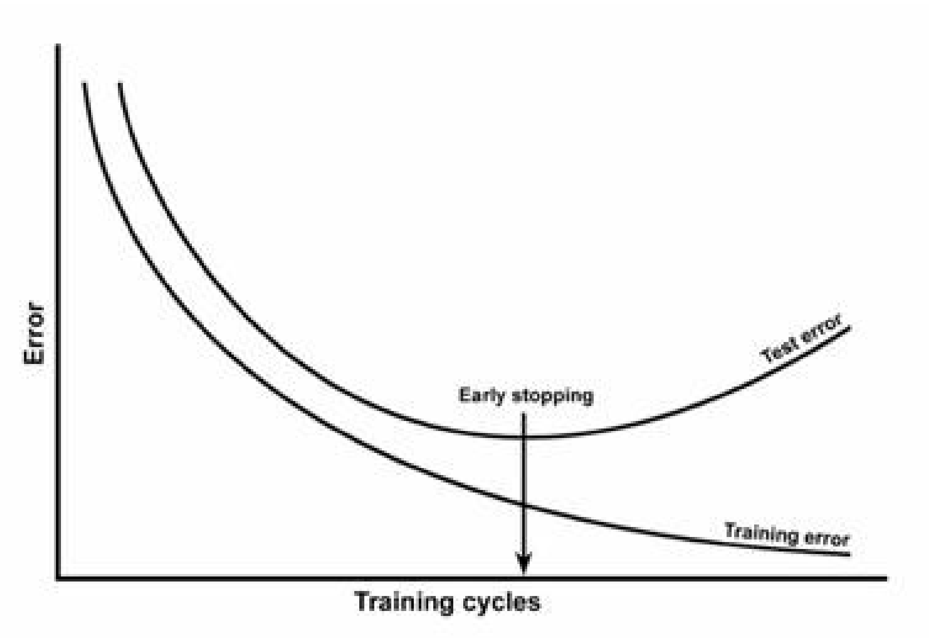


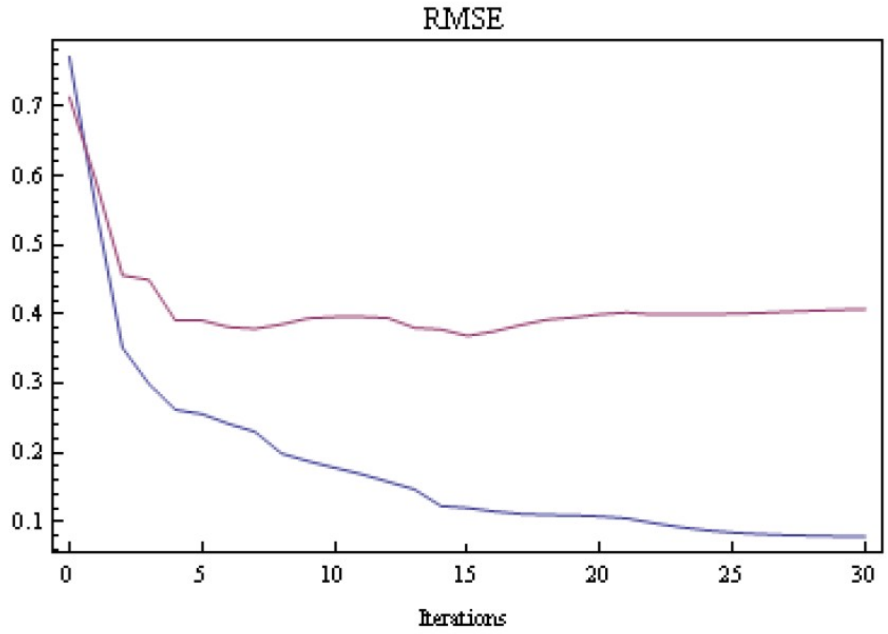
A mért hiba a tanító pontokon számolva, ill. más pontokon vizsgálva hasonló, de viszonylag nagy, a tanítás folytatásával csökken.

Hálózat tanításának jellemzése

- Hálózat **általánosító képessége**
  - Tanulásban **részt nem vett** bemenetekre milyen „mértékben” ad helyes kimenetet
- Általánosító képesség **meghatározása**:
  - Ismert input-output párok felosztása:
    - **Minta halmaz** / **validációs halmaz**
  - Általánosító képesség mérése:
    - **Hibafüggvény** segítségével, kiszámoljuk:
      - a minta halmazon ($\mathbf{H_m}$) és a
      - validációs ($\mathbf{H_v}$) halmazon.
  - Helyes mértékű tanítás:
    - A hiba a minta és a validációs halmazon **azonos nagyságrendben** van, ill. **együtt csökken** a tanítás folytatásával ($H_m \sim H_v$)
    - **Túltanulás:** H_m-et csökkenteni tudom, de H_v-t már nem a tanítás folytatásával
    - **Alultanulás**: a tanítás folytatásával H_m még lehetne csökkenteni úgy, hogy H_v is csökkenjen

Hiba változása a tanítás során



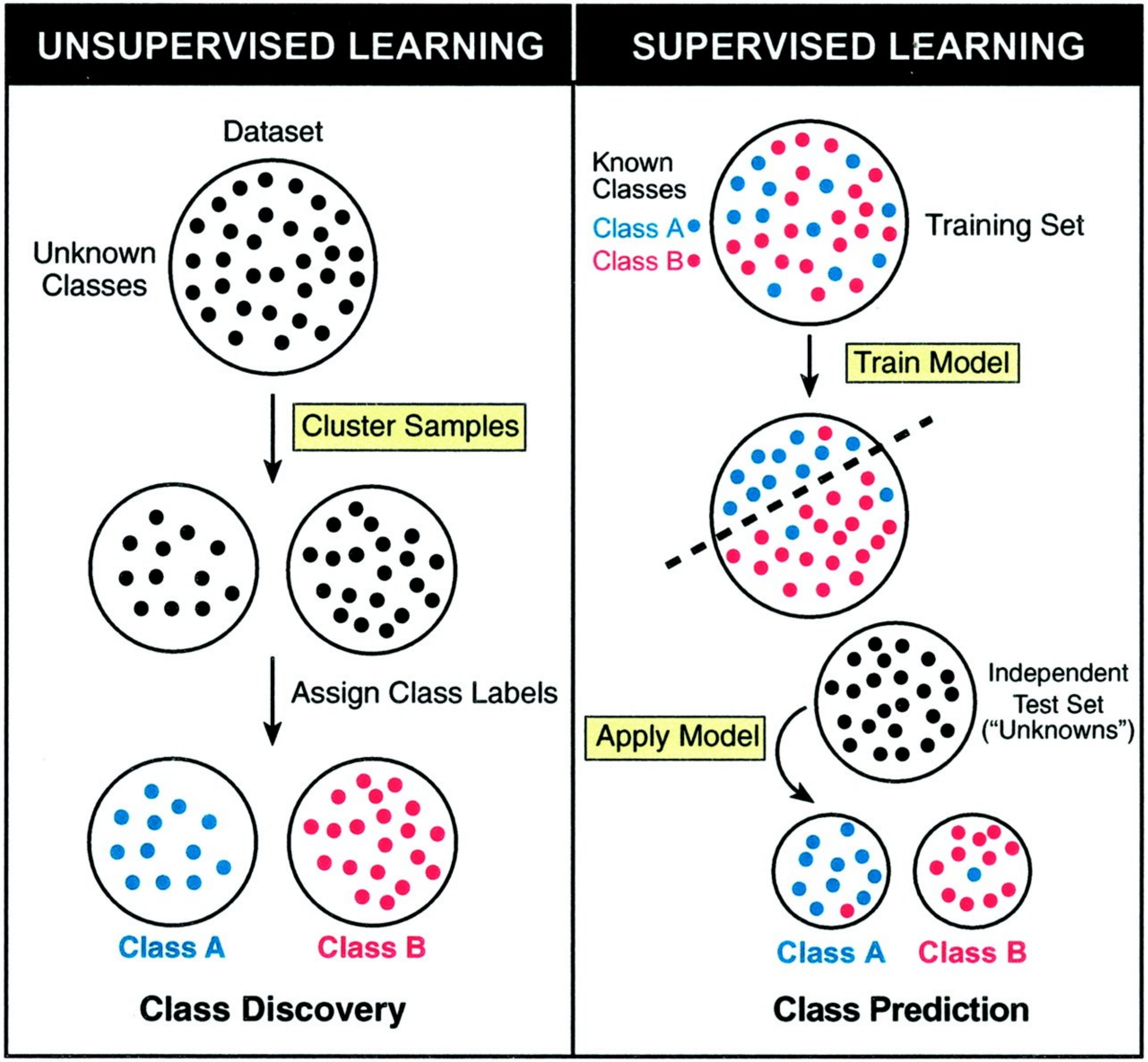


Mesterséges neurális hálózatok:
Hálózatok tanításának típusai

Hálózatok tanításának típusai

- Tanítás módszerének típusa:
  - **Felügyelt tanítás**
    - Input-output adatpárok használata a súlyok meghatározására a tanítás során
    - Jellemző megoldandó feladattípus: **függvény közelítés, adat-osztályozás**
    - Hibafüggvény: az input-output adatpárok alapján számoljuk
  - **Felügyelet nélküli tanítás**
    - Csak a bemeneti (input) adatok használata a súlyok meghatározására a tanítás során
    - Jellemző megoldandó feladattípus: **adat-klasszifikáció**
    - Hibafüggvény: az eredményre vonatkozó valamilyen a priori ismeretek alapján a kimeneti leképzés, hálózati súlyok stb. alapján számoljuk
      - pl. az osztályozás „jósága” alapján

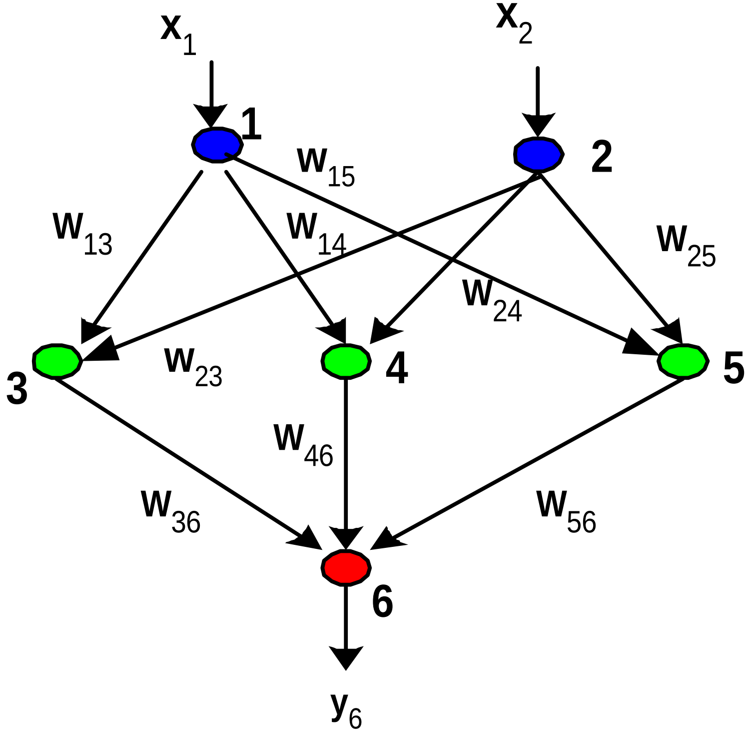
Felügyelt és felügyelet nélküli tanítás osztályozási feladat esetén



Mesterséges neurális hálózatok:
Hálózatok szimulációja

Neurális hálózatok szimulációja

- **Hálózatok szimulációja:**
  - Input-output leképzés megadása különböző bemenetek esetén



$$\begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} w_{13} & w_{23} \\ w_{14} & w_{24} \\ w_{15} & w_{25} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

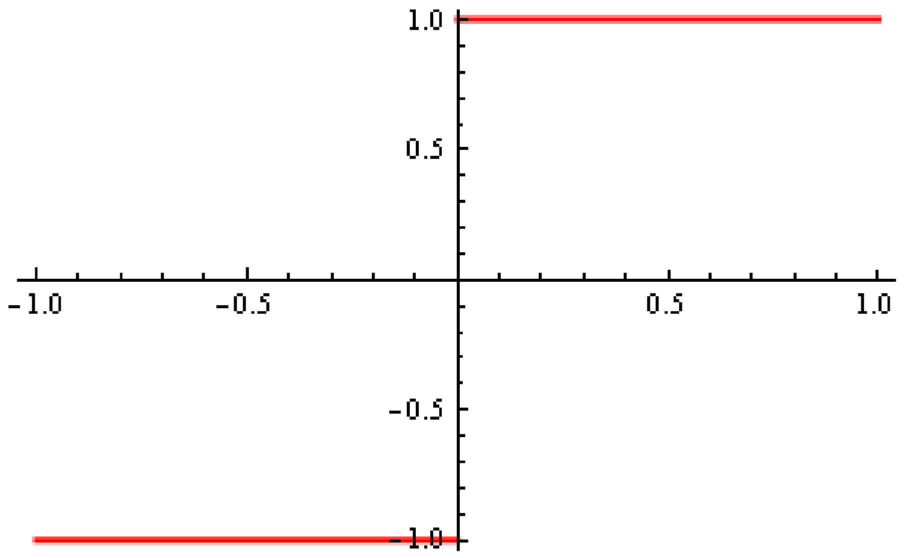
$$x_6 = \begin{pmatrix} w_{36} & w_{46} & w_{56} \end{pmatrix} \begin{pmatrix} f(x_3) \\ f(x_4) \\ f(x_5) \end{pmatrix}$$

$$y_6 = f(x_6)$$

Mesterséges neurális hálózatok:
Perceptron hálózat

Egyrétegű perceptron hálózat

- **Egyrétegű perceptron hálózat:**
  Rejtett réteg nélküli neurális hálózat
- Átviteli függvény:
  - **Signum függvény**
- Alkalmazási terület:
  - Osztályozás, alakfelismerés
- Létezik bináris változata
  - 0,1 leképzés
  - Egyszerűbb tanítás
- Csak **lineárisan szeparábilis** problémák megoldására alkalmazható (l. később)
- Rosenblatt (1962)

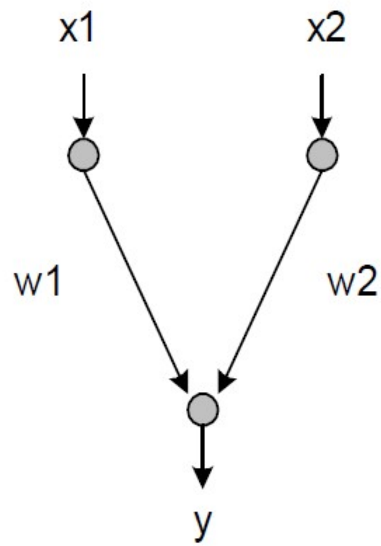


A eltolás (bias) alkalmazása a küszöbérték helyett

- Példa:
  - AND logikai függvény megvalósítása
  - **X1**, **x2** bemenetek; **w1**, **w2**, súlyok; $\boldsymbol{\tau}$ eltolás

A eltolás (bias) alkalmazása a küszöbérték helyett

- Példa:
  - AND logikai függvény megvalósítása
  - **x1**,**x2** bemenetek; **w1**, **w2**, súlyok; **τ** eltolás

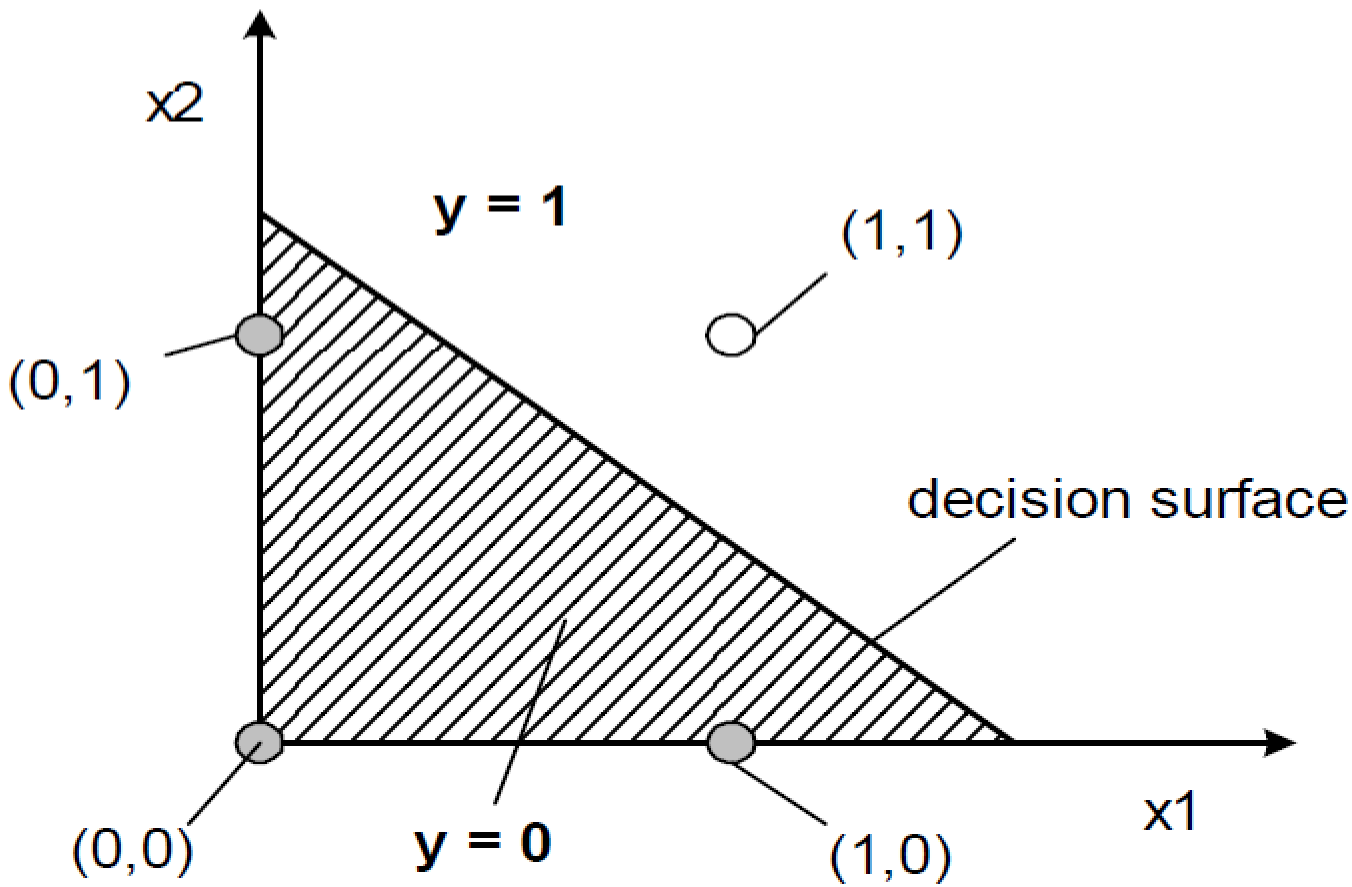


Az AND függvény input - output táblázata

x1	x2	y
0	0	0
0	1	0
1	0	0
1	1	1

$$y(x1,x2) = f\,(w1\;x1 + w2\;x2 - \tau)$$

AND döntési felülete



Tanítás és a döntési felület

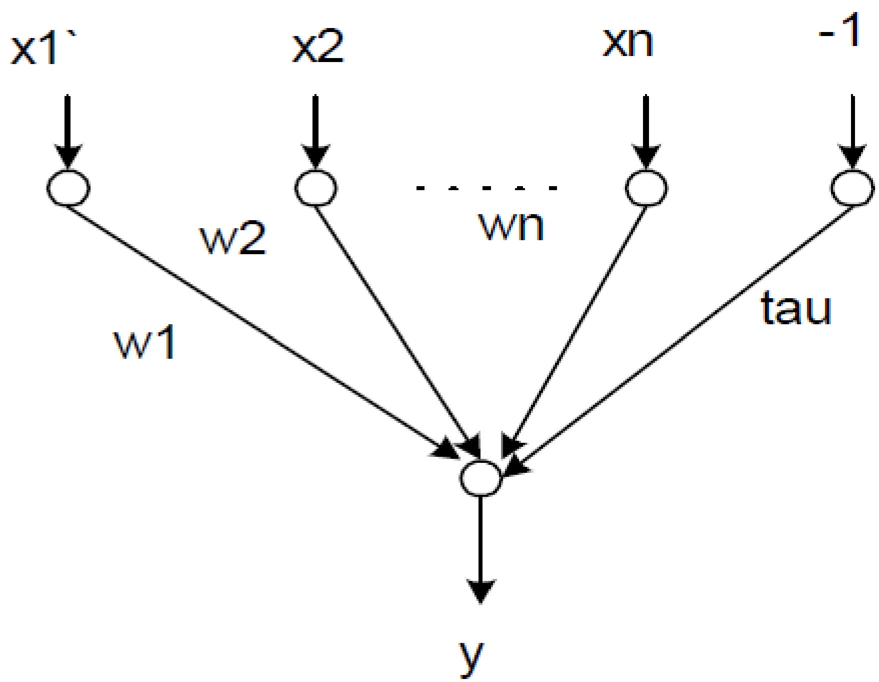
- **Döntési felület** osztályozási feladat esetén: az a felület, ami elválasztja a különböző kimenetet eredményező bemeneti értékeket tartalmazó **bemeneti tartományokat**
- Döntési felület az egyrétegű **perceptron hálózat** esetén egy **n-1 dimenziós felület**, ahol n a probléma bemeneti változóinak száma
- A **tanítás során** implicit módon **a döntési felületet határozzuk meg**, azt „mozgatjuk" a súlyok változtatásával

- **AND** esetén a döntési felület alapján a súlyok meghatározása:
  $x2 = -a \cdot x1 + b$ ← *döntési felület egyenlete*

  $0 = w2 \cdot x2 + w1 \cdot x1 - \tau$ ←*neurális háló itt „vált" (l. háló egyenlet+aktivációs fv.)*
  $x2 = -w1/w2 \cdot x1 + \tau/w2$
- Döntési felület paraméterei és a súlyok közötti összefüggések:
  $a = w1/w2$, $b = \tau/w2$ ←→ $w2=\tau/b$, $w1= \tau \cdot a/b$

Hálózat megvalósítása bias alkalmazásával

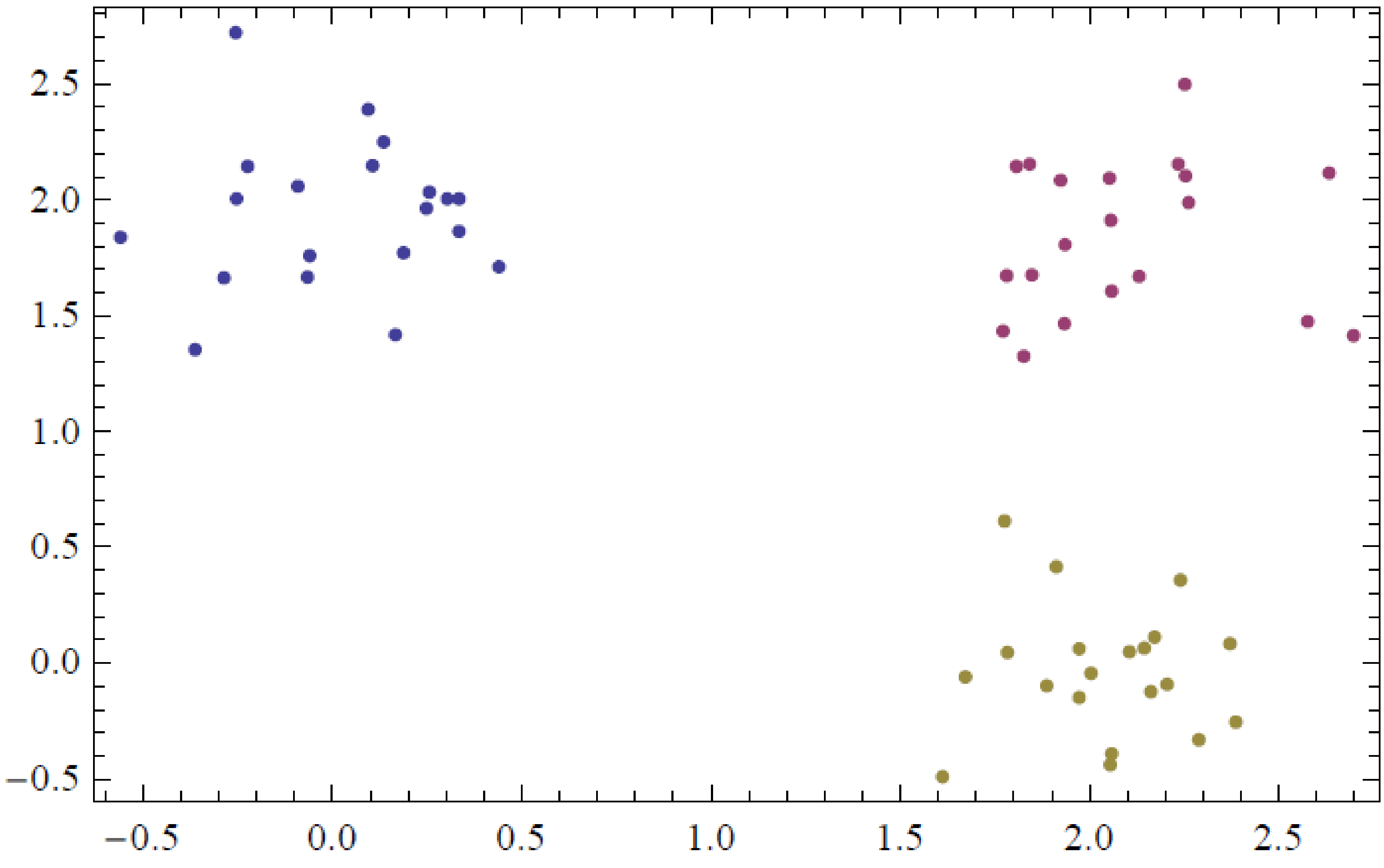
- Aktivációs függvény küszöbértéke: 0, az un. **bias („eltolás”) csomópont** súlyával (τ) állítható a küszöbérték



Perceptron hálózat: példa

- Van három csoportba sorolható ponthalmazom a kétdimenziós térben:
  - Koordináták: X1, X2

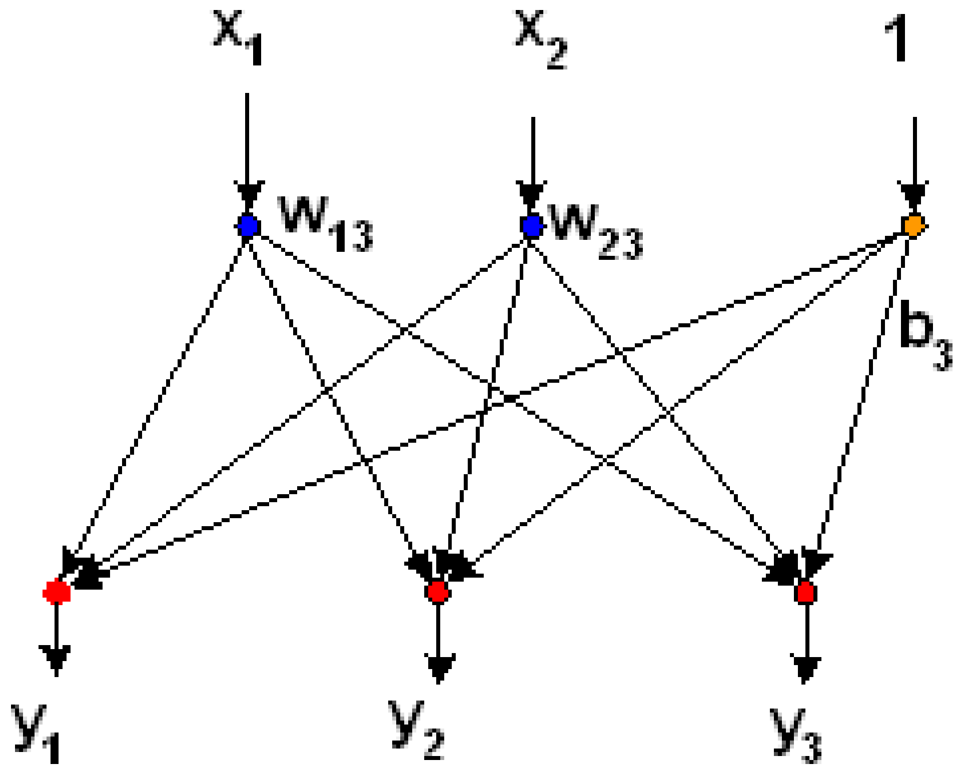
Perceptron hálózat: input pont halmaz



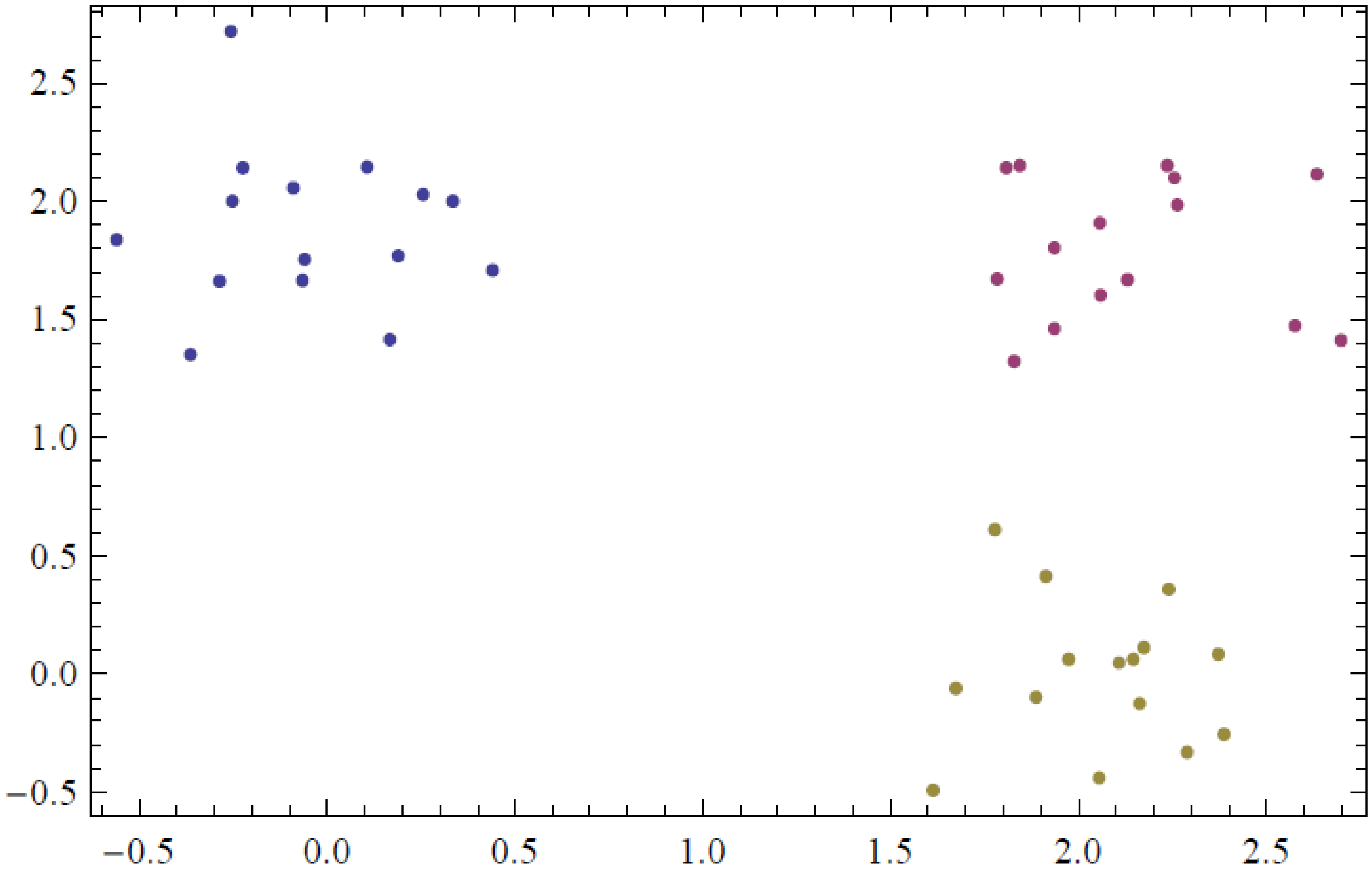
Perceptron hálózat: példa

- Van három csoportba sorolható ponthalmazom a kétdimenziós térben:
  - Koordináták: X1, X2
- A hálózat **kimeneteit** az egyes **csomópontokhoz** rendeljük:
  - A csoporthoz rendelt kimeneteten az 1 érték jelezze az adott kimenethez tartozó pontot
  - Kimenetek: y1, y2, y3
- Perceptron hálózat: nincs rejtett réteg

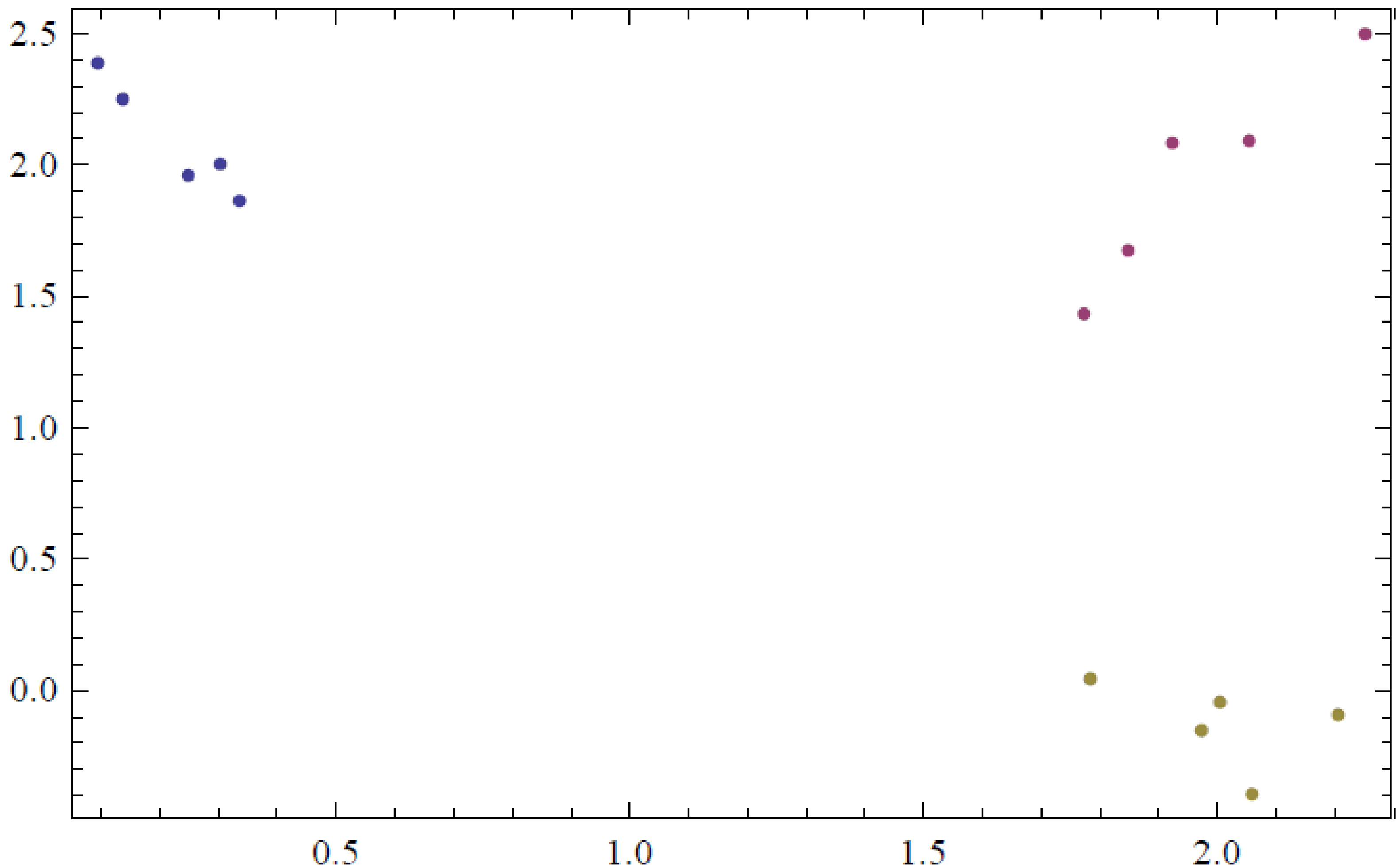
Perceptron hálózat: hálózati topológia



Perceptron hálózat: tanító halmaz



Perceptron hálózat: validációs halmaz



Perceptron hálózat tanítása

- **Tanuló** halmaz:
  - ***m*** darab $\{\boldsymbol{x}^{(k)}, \boldsymbol{d}^{(k)}\}$, **k = 1, 2,…,m** input-output pár
  - $\boldsymbol{x}^{(k)}$, $\boldsymbol{d}^{(k)}$ vektorok
  - $\boldsymbol{x}^{(k)}$ k-adik inputhoz a $\boldsymbol{d}^{(k)}$ output tartozik
- Feltesszük, hogy **m > n**,
  ekkor a következő egyenlet **túlhatározott**:
  (***m*** egyenlet, ***n*** ismeretlen)

$$d^{(k)} = sign\left(\sum_{i=1}^{n} x_i^{(k)} w_i\right) \quad k = 1,2,\ldots,m$$

Tanítás lépései

1) Kezdeti súlyok adása: pl. véletlen súlyok [0,1] (**w(n):=rand();**)
   k:=1 *(*k: *lépés szám)*
2) $\mathbf{x^{(k)}} \rightarrow \mathbf{y^{(k)}}$ kiszámítása (k-dik bemenetre adott válasz a w(n) alapján)
3) Ha $\mathbf{y^{(k)} =! d^{(k)} \rightarrow w(n+1) = w(n) + \Delta w}$

   $\mathbf{\Delta w = \nu \cdot d^{(k)} \cdot x^{(k)}}$, $\mathbf{0< \nu <1}$

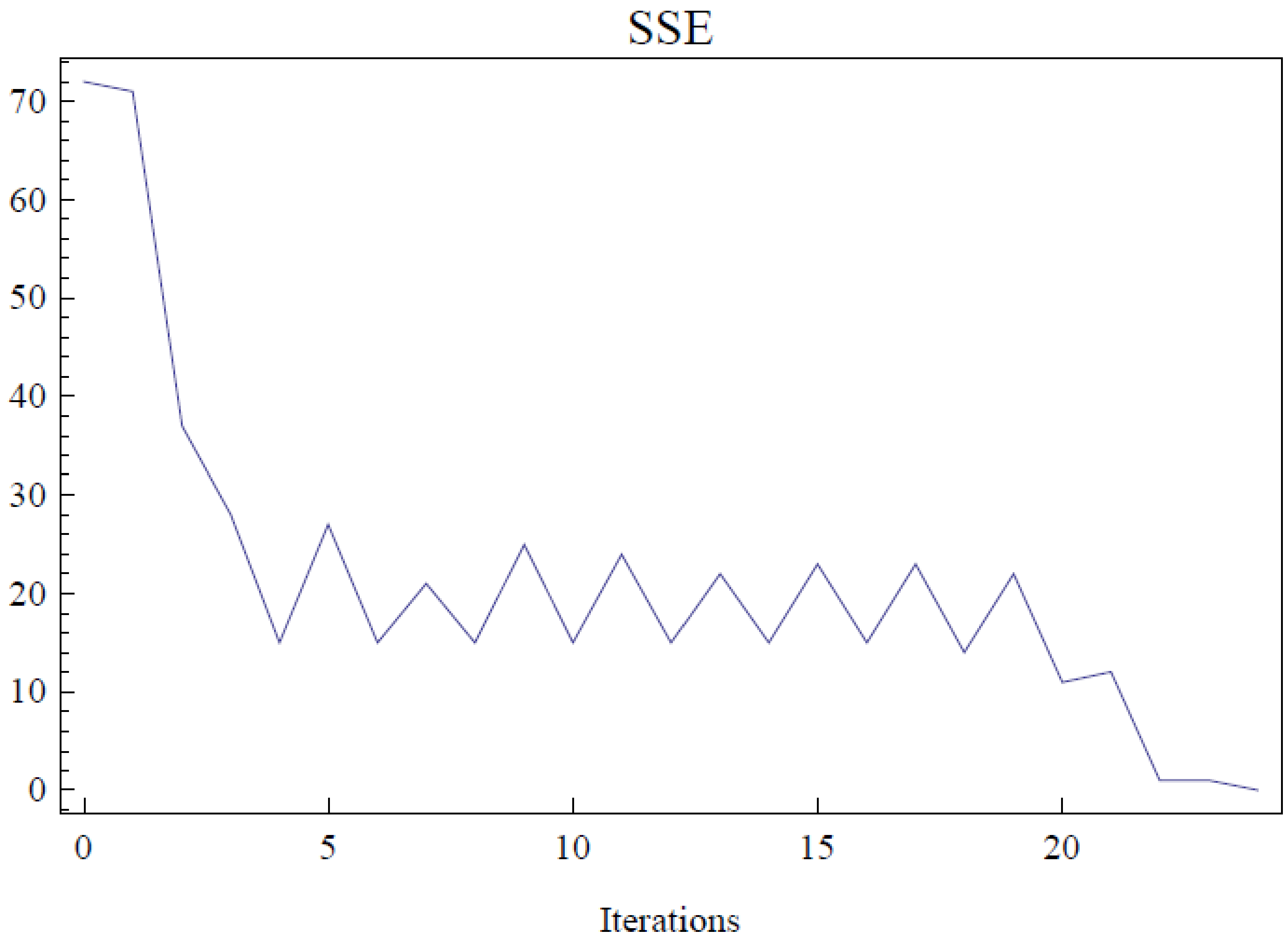
   ($\boldsymbol{\nu}$*: tanulási paraméter,* $\mathbf{d^{(k)}}$*={-1;+1}, a súlyokat az* $x^{(k)}$ *irányába mozgatjuk)*

   Ha $\mathbf{y^{(k)} = d^{(k)} \rightarrow w(n+1) = w(n)}$
4) **k:=k+1,** GOTO 2) (egymás után vesszük a bemenet/kimenet párokat)

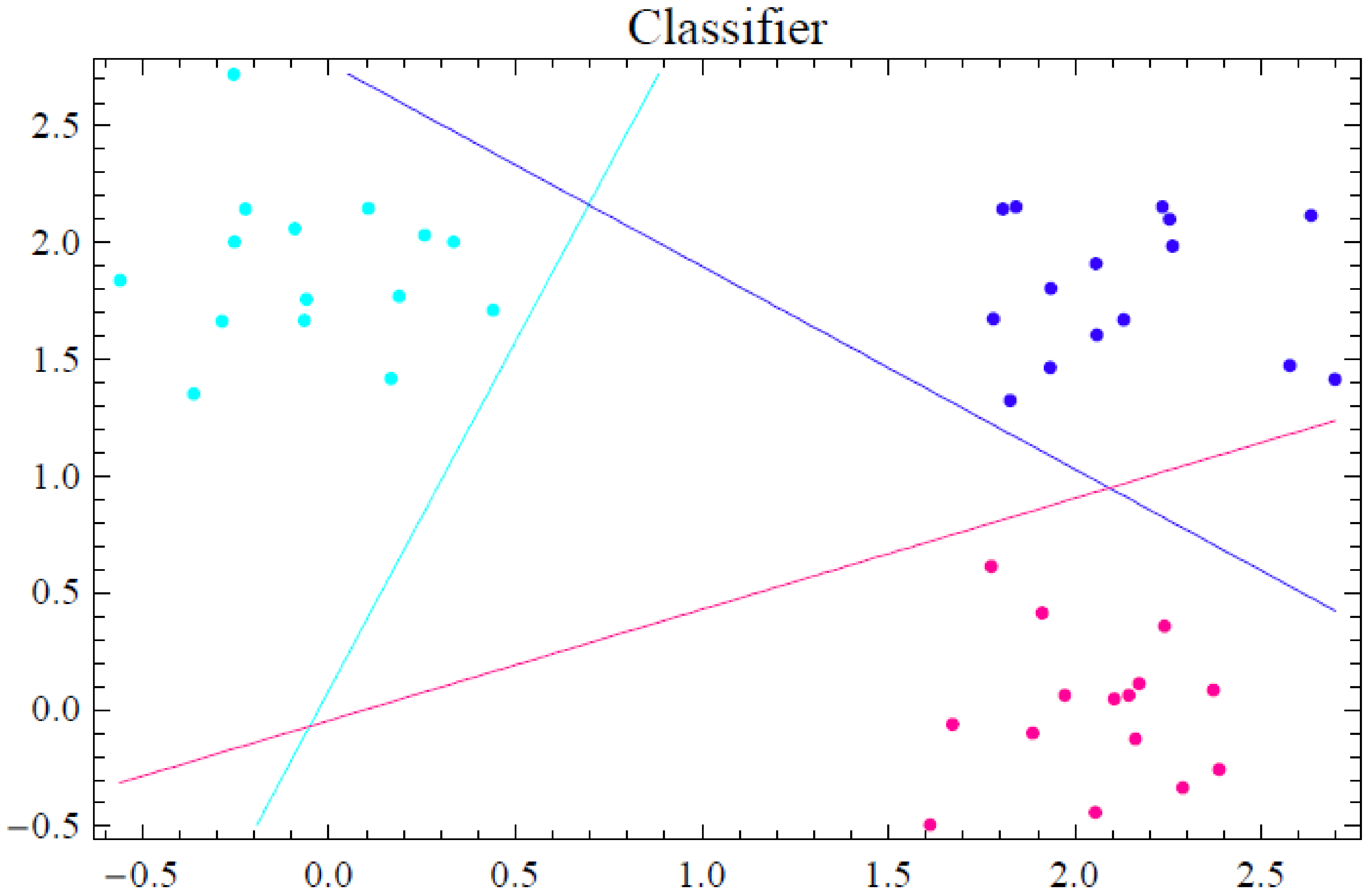
   Megállási feltétel: egyik input esetén sem kellett a súlyokat változtatni!

   (*Rosenblatt és Novikoff tétel: változtatások száma korlátos* → *véges lépés)*

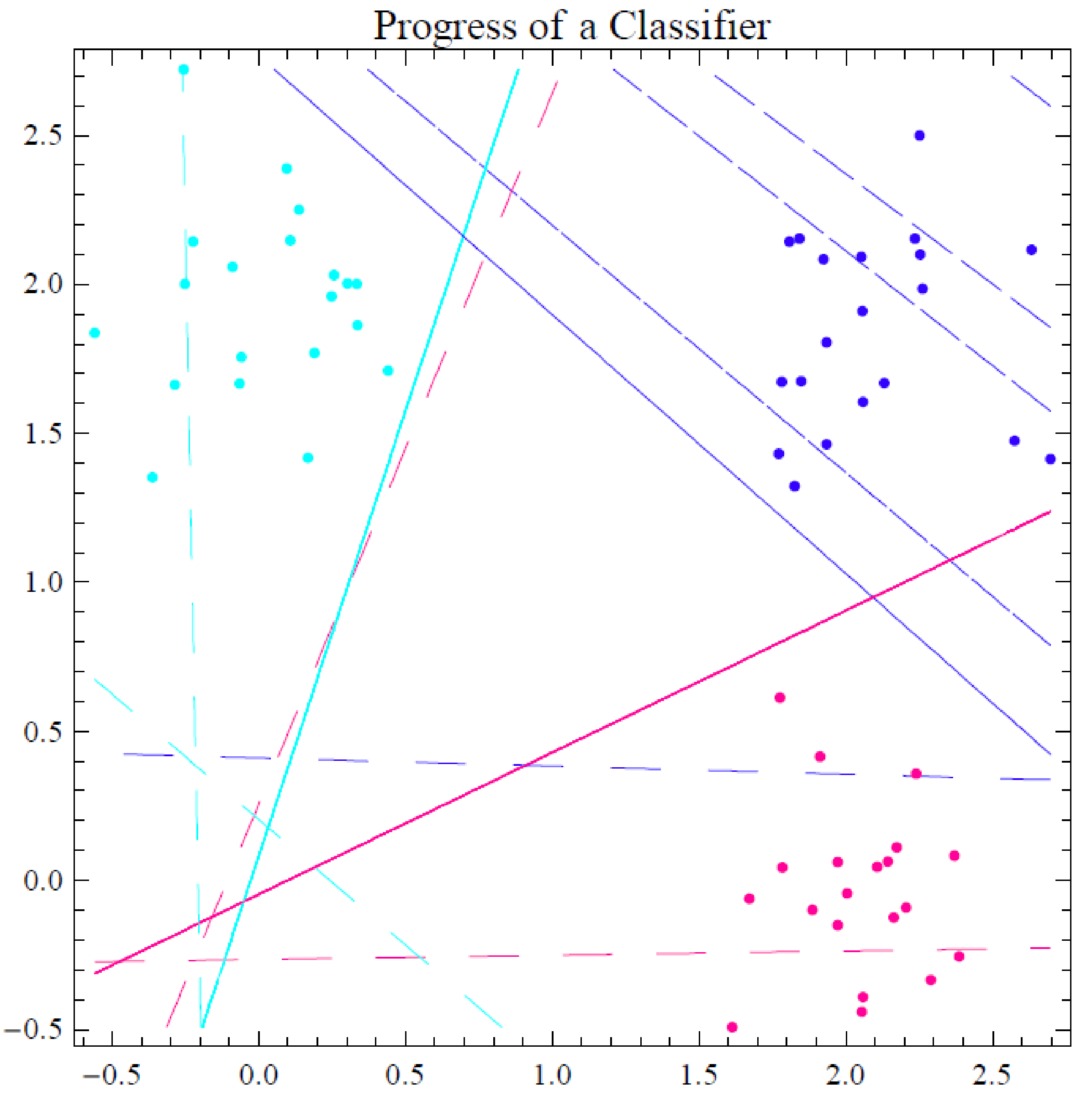
Iteratív tanítás: hibafüggvény változása iterációnként



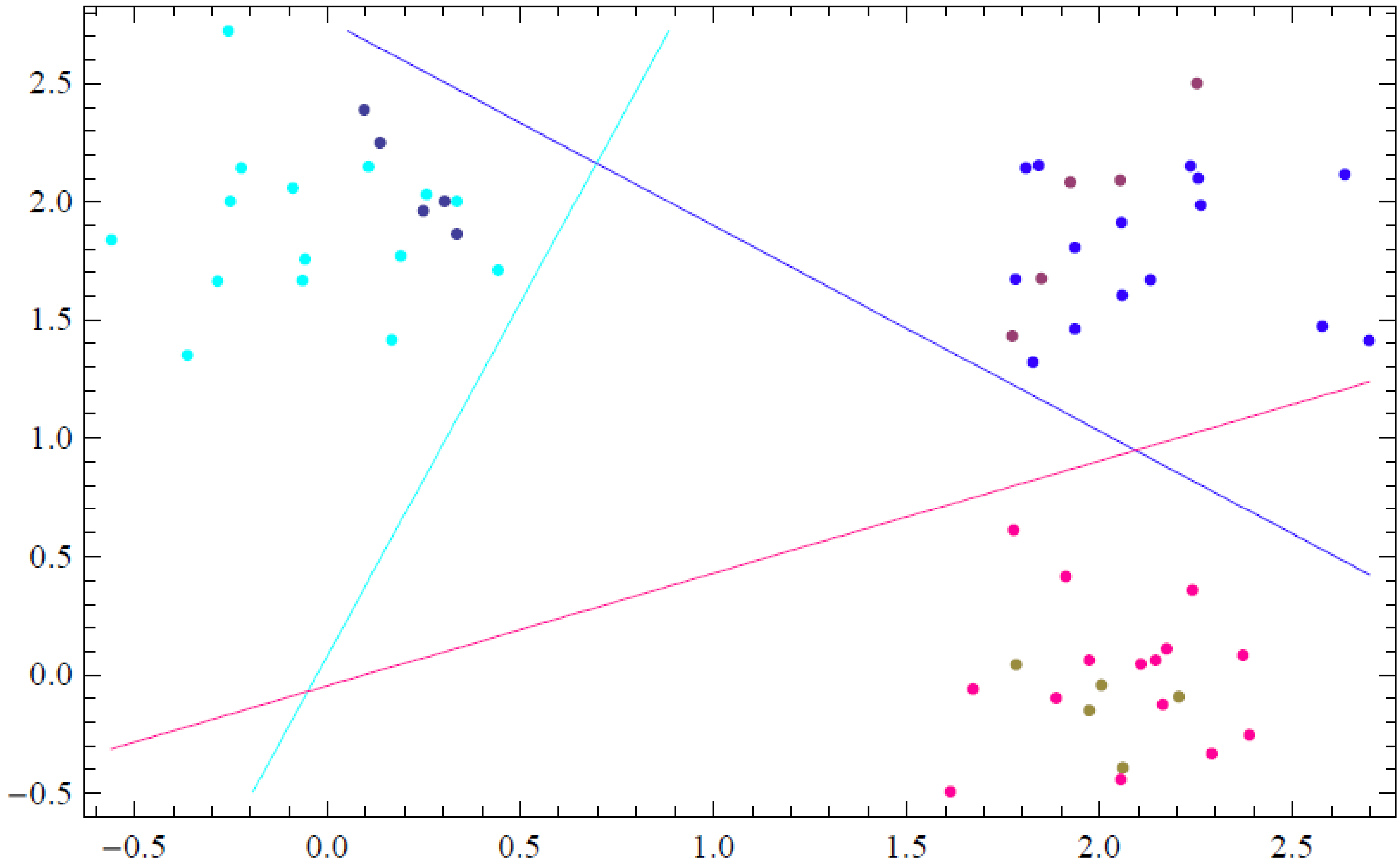
Az eredményül kapott vágási síkok



Vágási síkok változása az iterációk során



Tanító és validációs pontok a vágási síkokkal

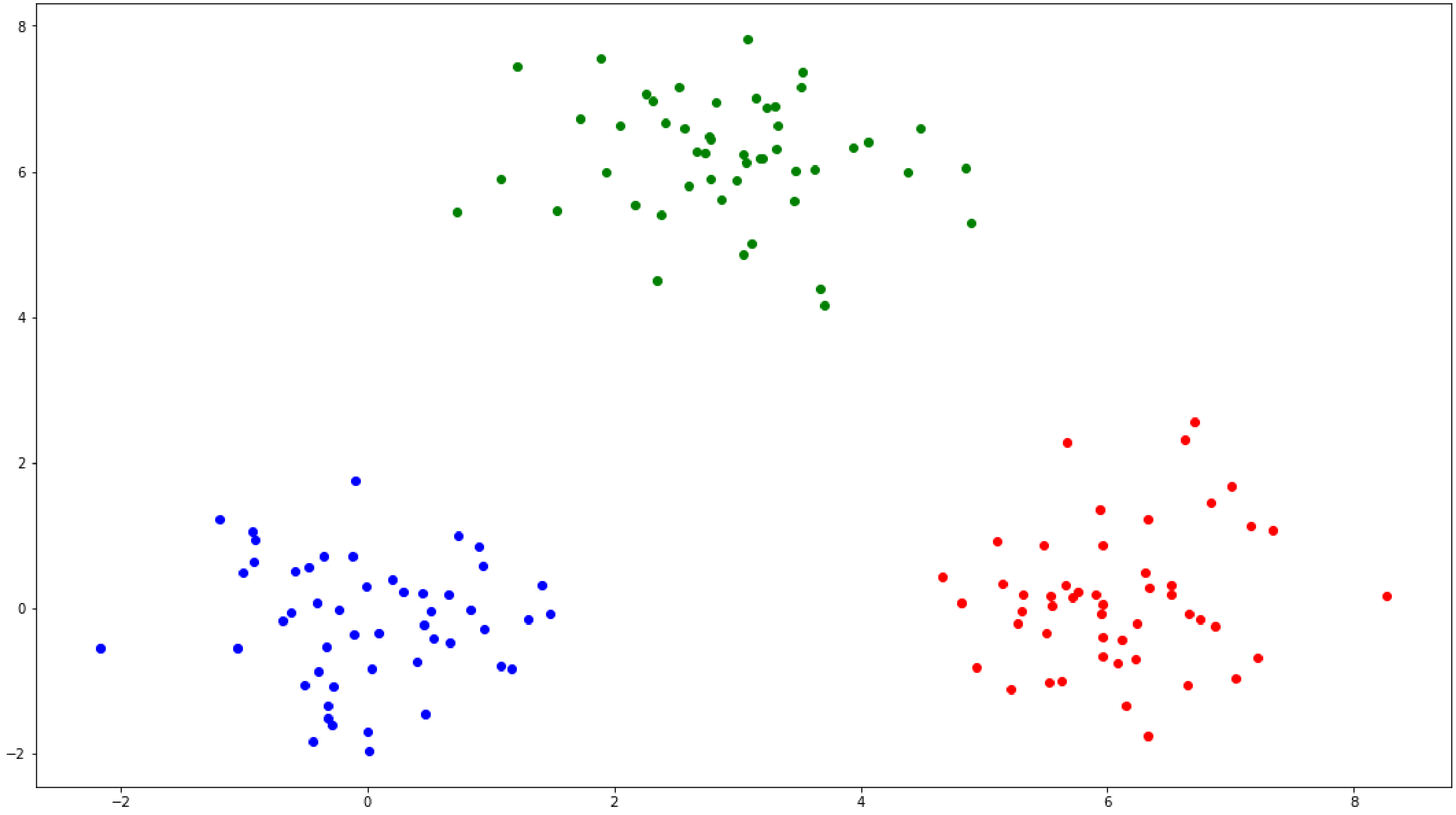


Perceptron hálózat értékelése

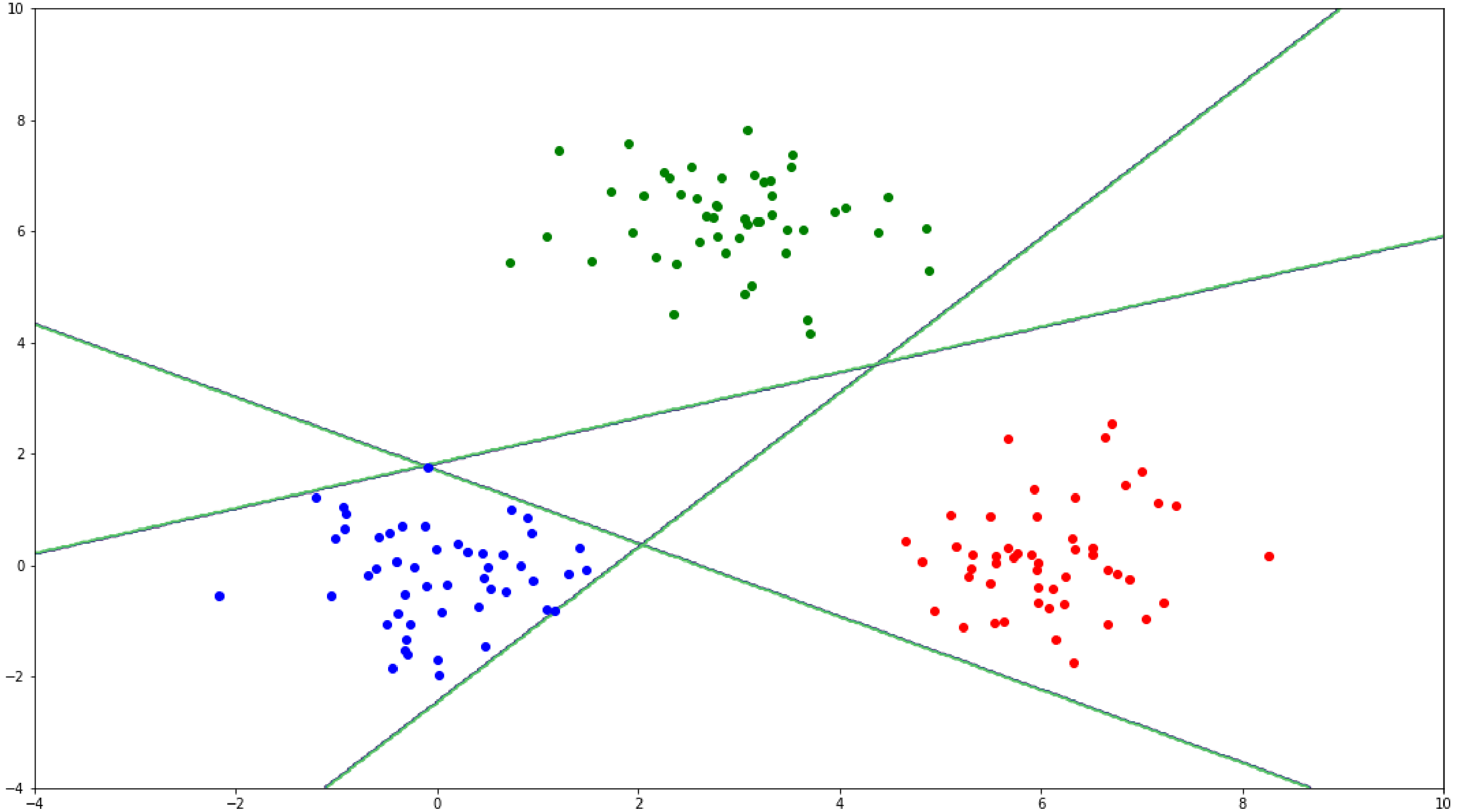
- **Sikeres** osztályozás:
  - Jól definiált tartományok az egyes osztályokhoz
- **Hátrány**: A paraméter térben (síkon) bizonyos tartományok **nem tartoznak egyértelműen** a megkülönböztetett osztályokhoz
- Csak **lineárisan szeparábilis** (szeparálható) problémák megoldására alkalmazható

Mesterséges neurális hálózatok:
Perceptron hálózat demó

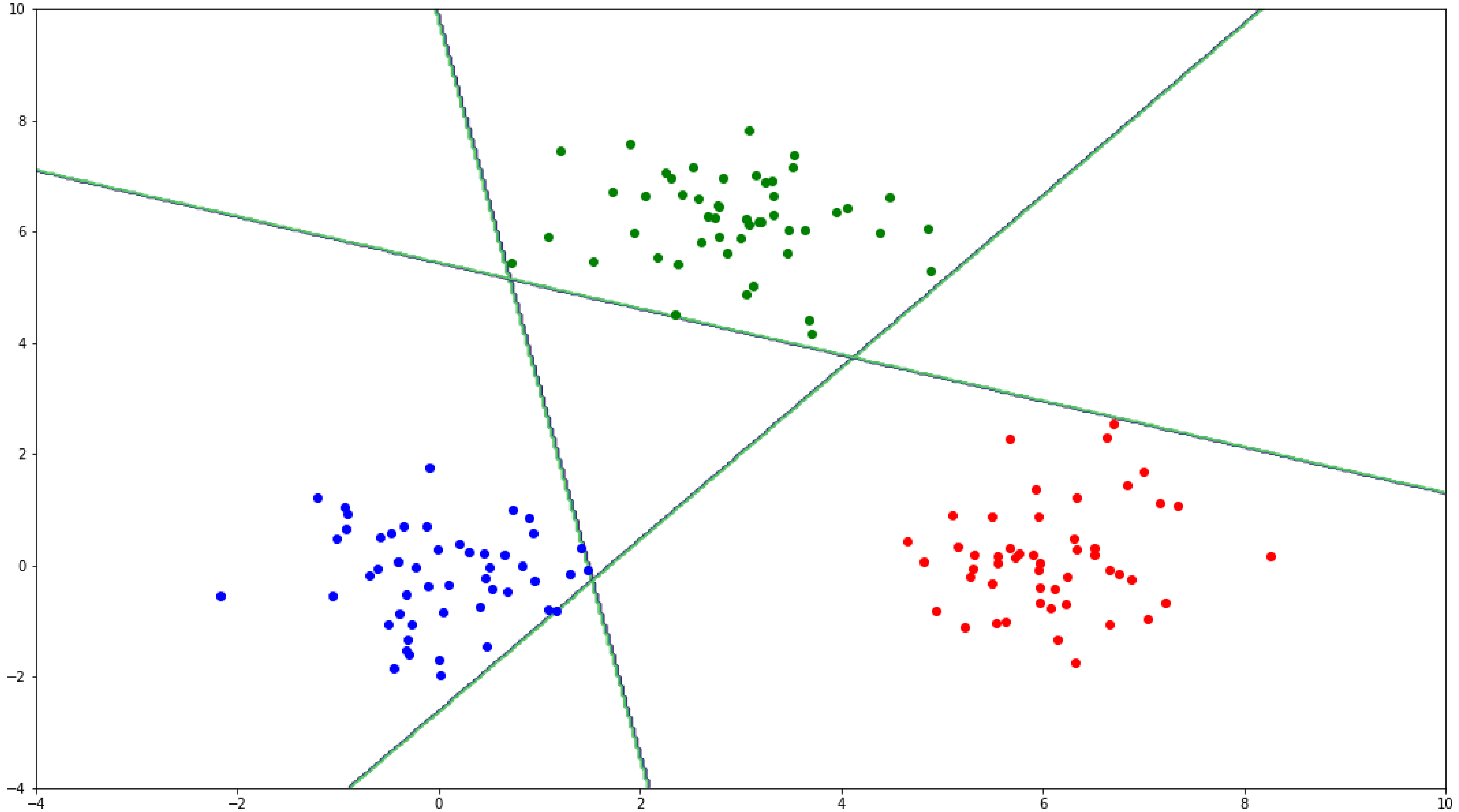
Perceptron példa három osztályra: bemeneti adathalmaz



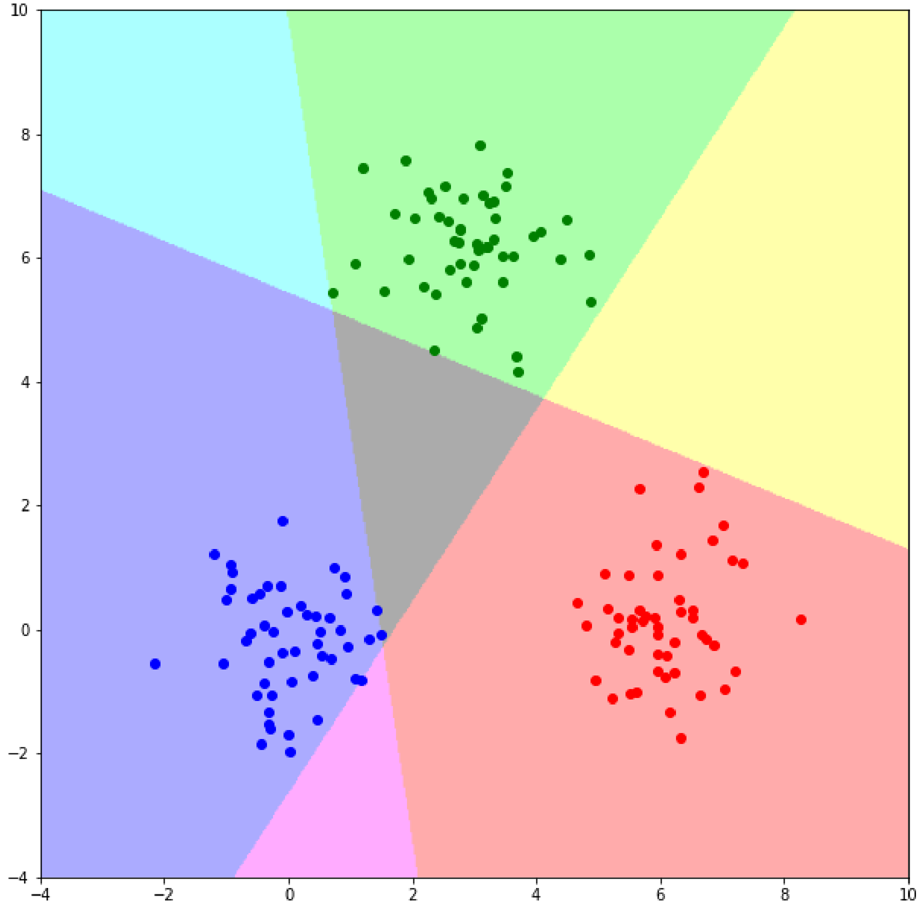
Perceptron példa három osztályra: eredmény - szeparációs hipersíkok



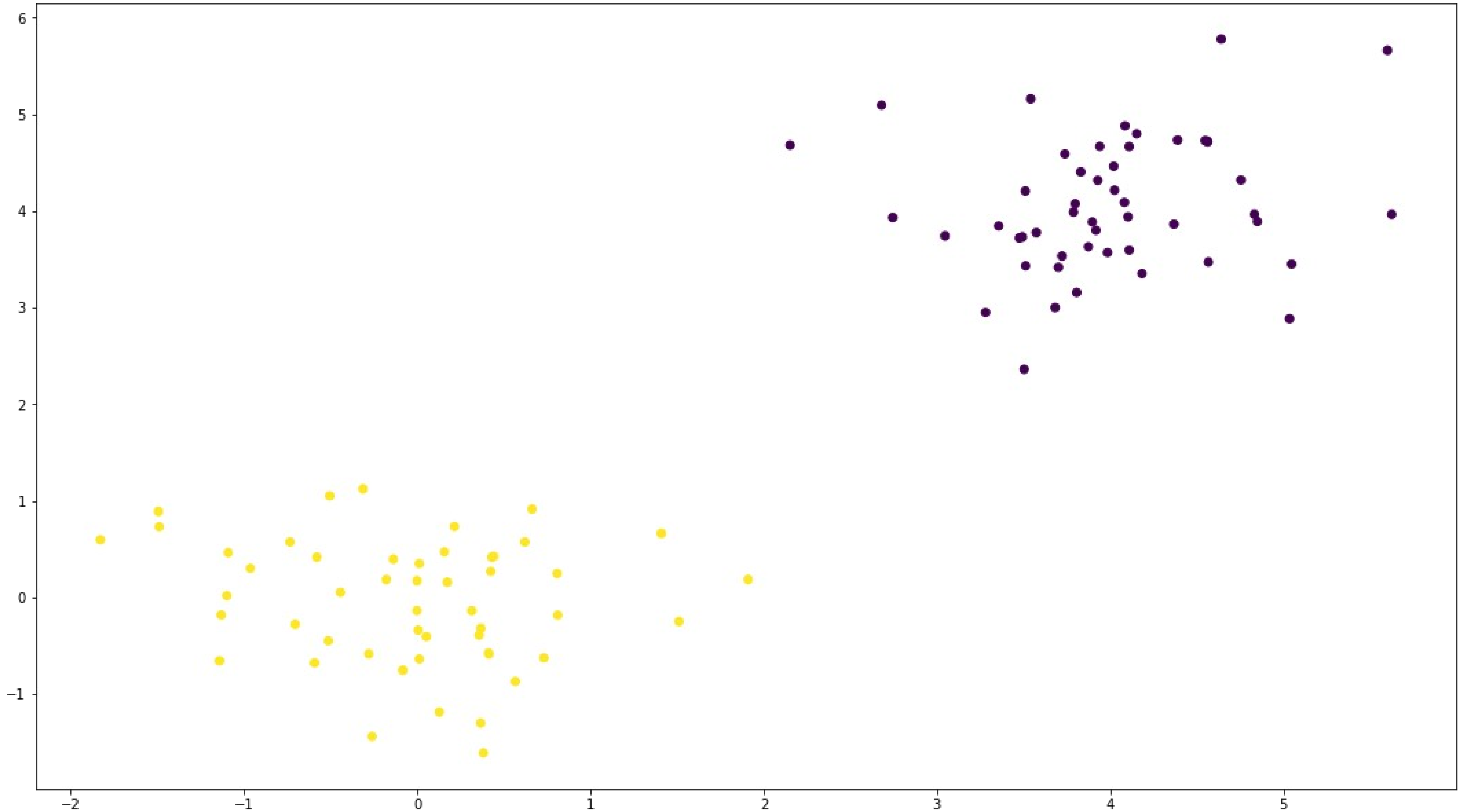
Perceptron példa három osztályra: eredmény - szeparációs hipersíkok



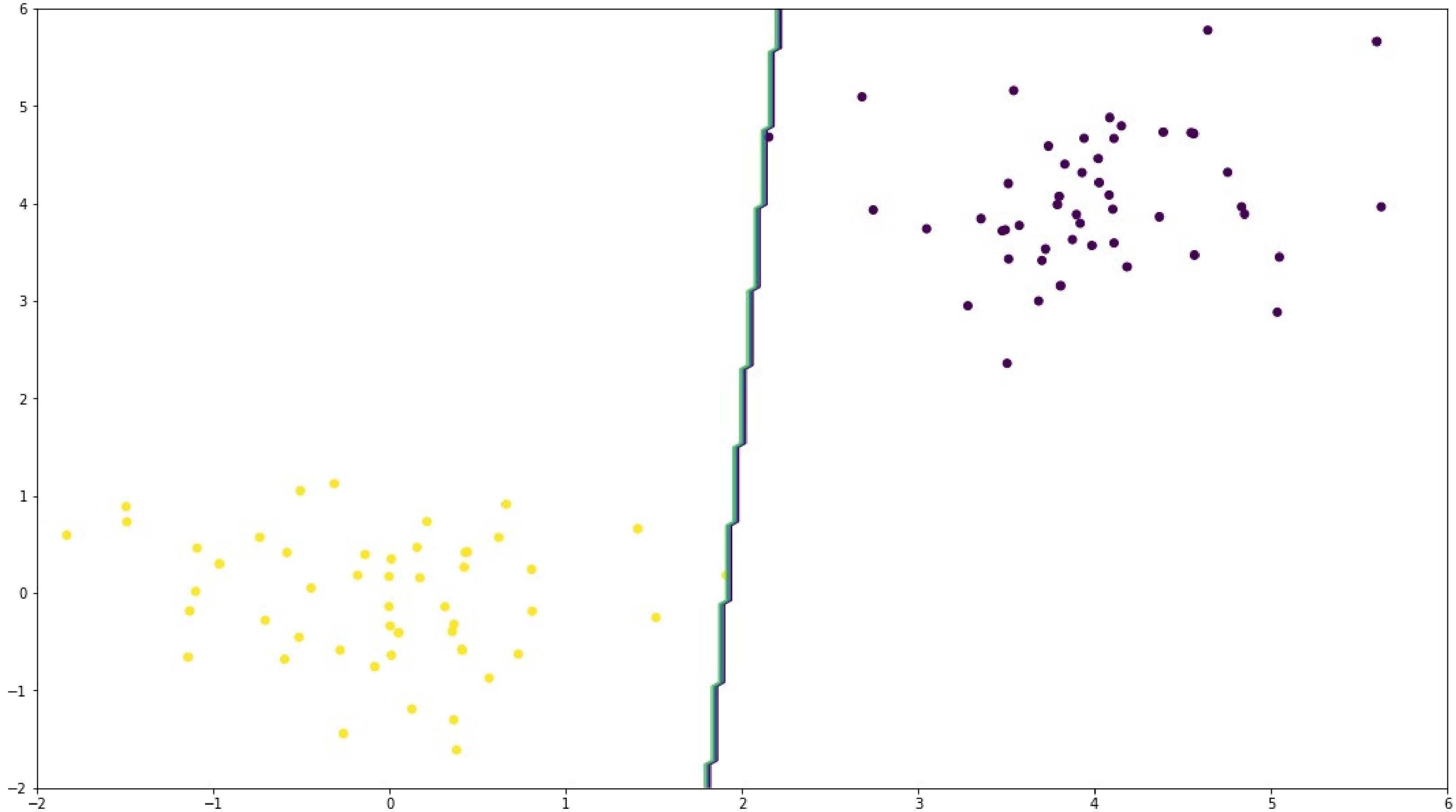
Perceptron példa három osztályra: eredmény - tartományok



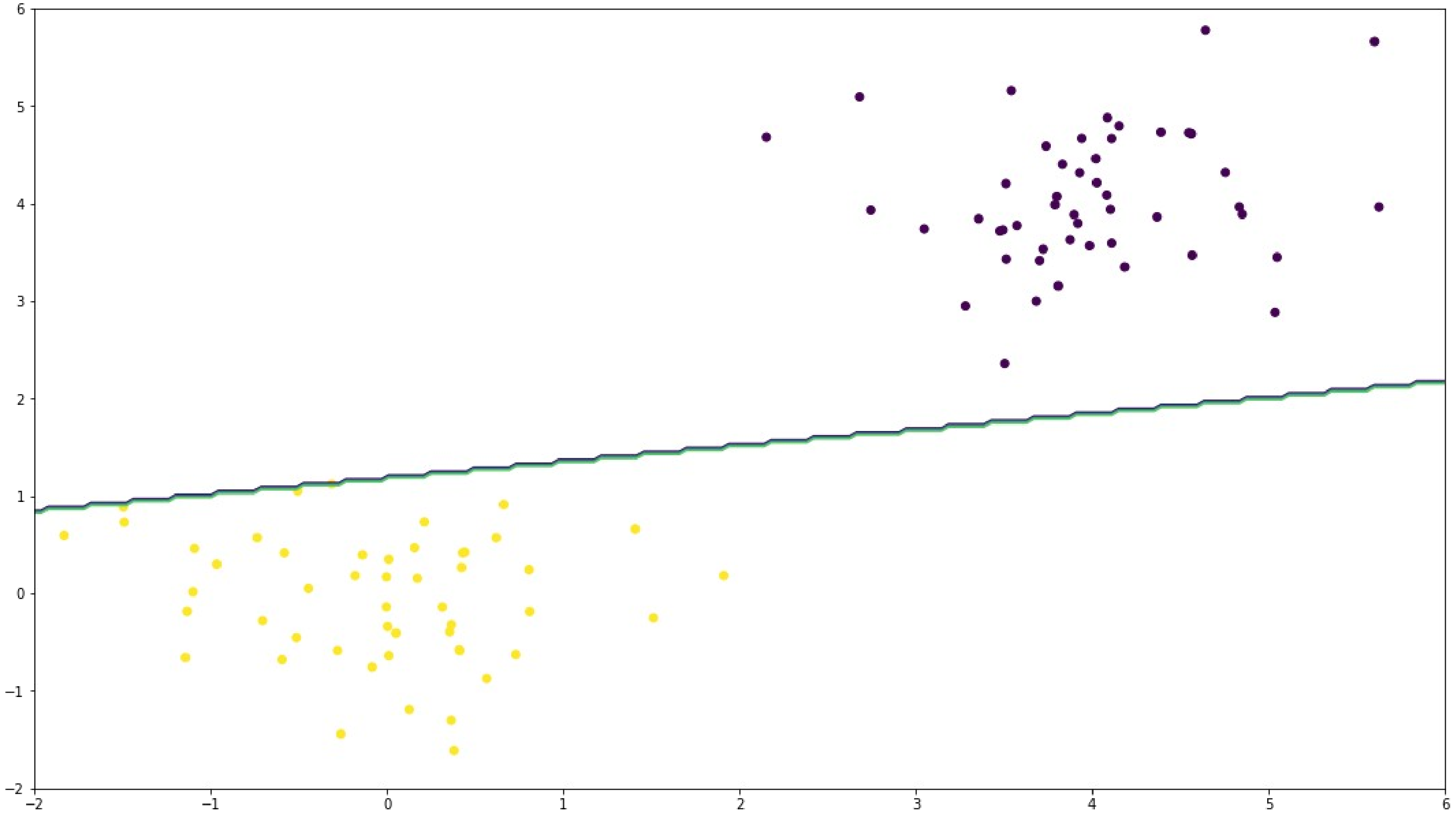
Perceptron példa két osztályra: bemeneti adathalmaz



Perceptron példa két osztályra: osztályozás függése a kezdeti értékektől



Perceptron példa két osztályra: osztályozás függése a kezdeti értékektől

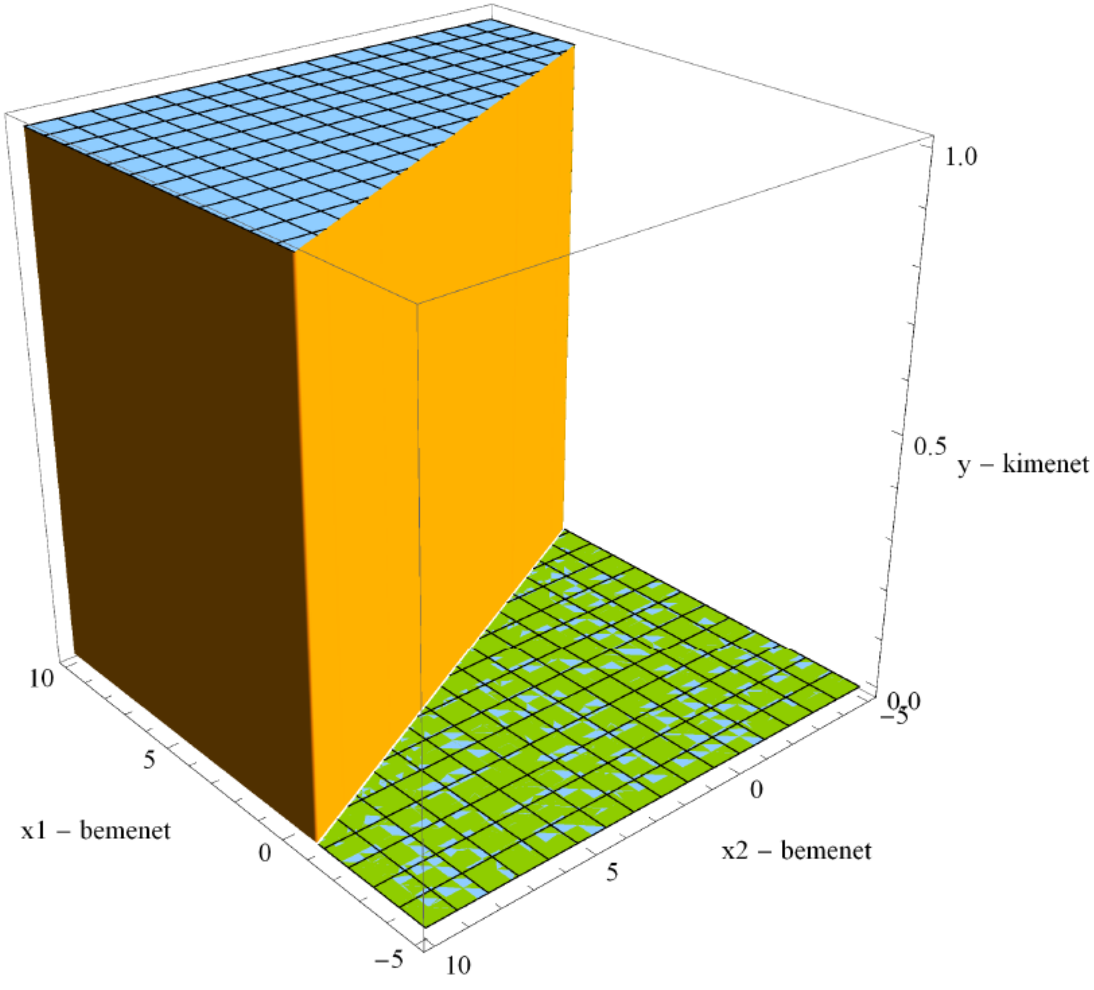


Mesterséges neurális hálózatok:
Döntési felület

Döntési sík: egy egyszerű példán keresztül

- Osztályozási probléma
- Bemenetek:
  - x1, x2
- Kimenet:
  - y
- Perceptron hálózat, súlyok:
  - w1=0,5
  - w2=0,25
- Eltolás:
  - Z=2
- {x1, x2} síkon két kimeneti értéktartomány:
  - y=0
  - y=1
- Elválasztó felület:
  - Egyenes
- Magasabb dimenzióban:
  - **Döntési felület:**

    Hálózat kimeneti értékei alapján meghatározható bemeneti értéktartományokat elválasztó felület



Lineárisan szeparálható probléma

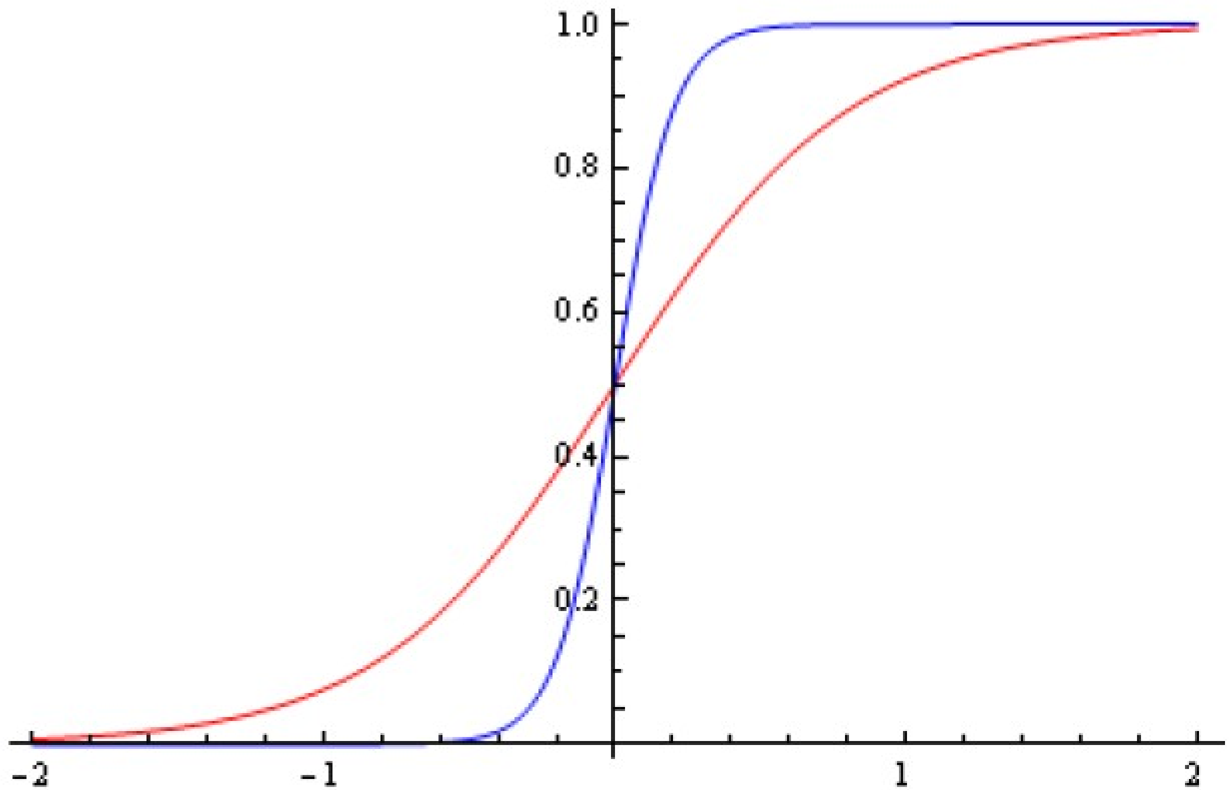
- Adott:
  - egy **n** dimenziós térben,
  - különböző osztályokba tartozó pontok.
- **Lineárisan szeparálható** az adott probléma:
  - Az egyes osztályok elkülönítése lehetséges egy **n–1** dimenziós hipersíkkal.

Backpropagáció (backpropagation) hálózatok

Backpropagation hálózatok jellemzői

- Folytonos kimeneti értékek előállítása:
  - Aktivációs függvény általában: ***szigmoid függvény***
  - Általában *alfa = 1*

$$\sigma(x,\alpha) = \frac{1}{1+e^{-\alpha x}}.$$



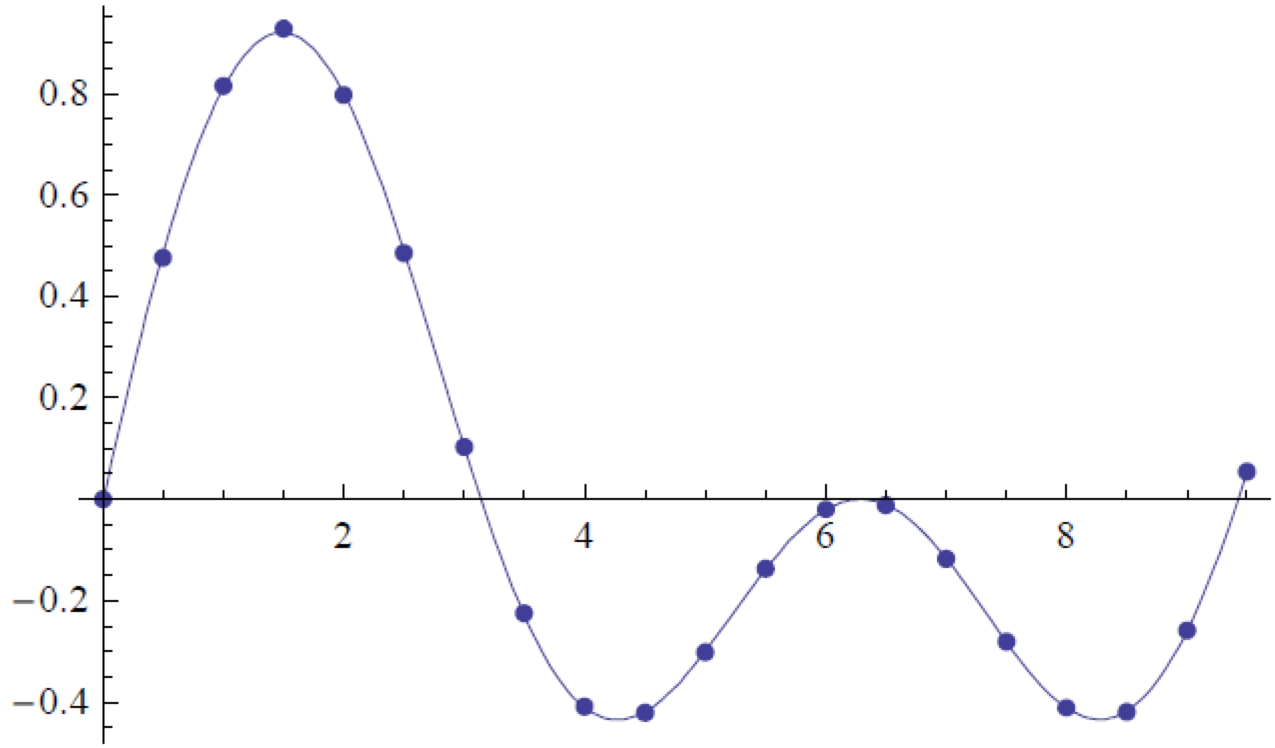
Backpropagation hálózatok jellemzői

- **Rejtett rétegek** száma:
  - Egy vagy több
  - Erősen **nemlineáris** viselkedés
- Gyakori alkalmazásai:
  - Nemlineáris folytonos **függvények közelítése**
  - Nemlineáris **osztályozási feladatok**

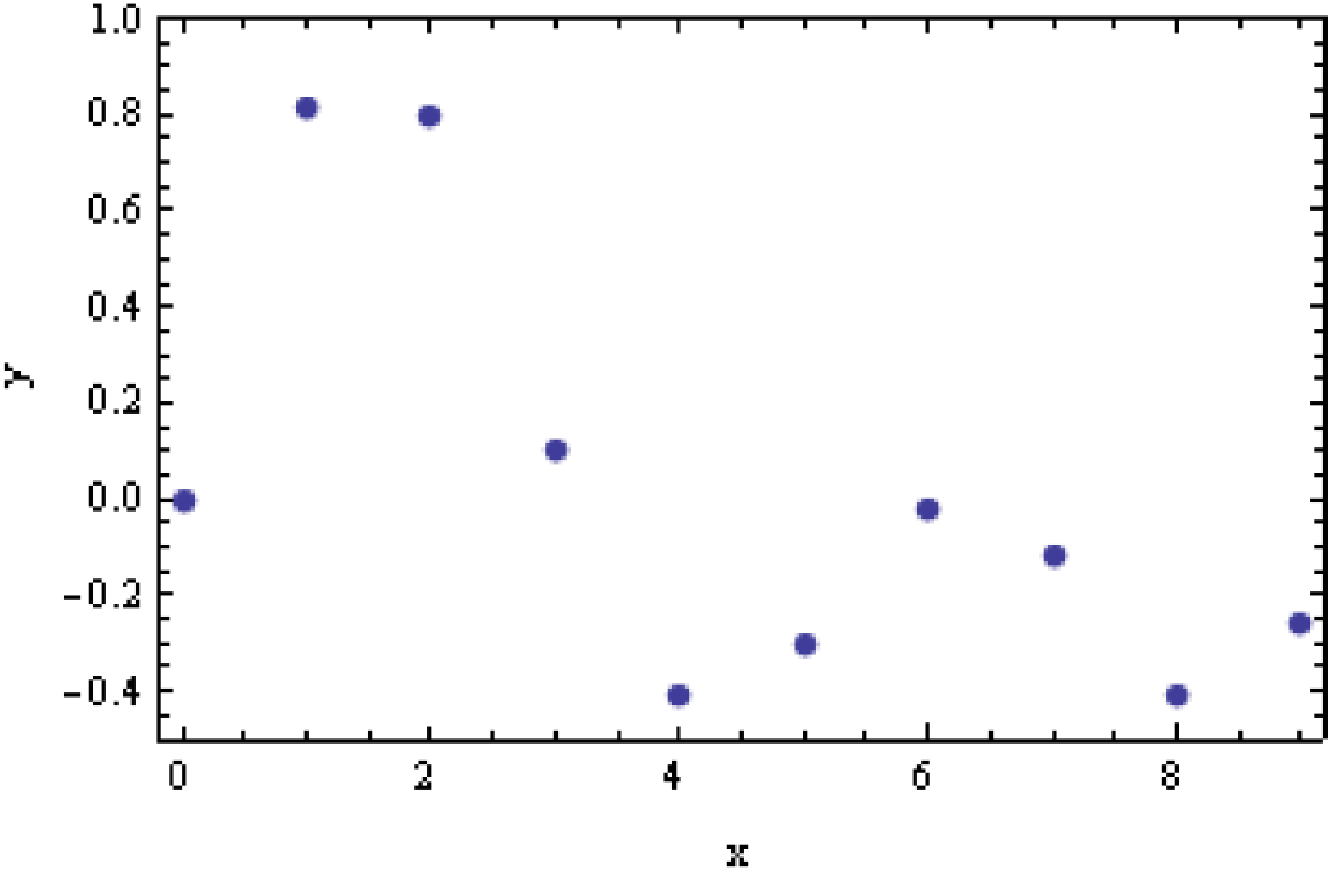
Példa: Függvény közelítése diszkrét pontok alapján

- Függvény:

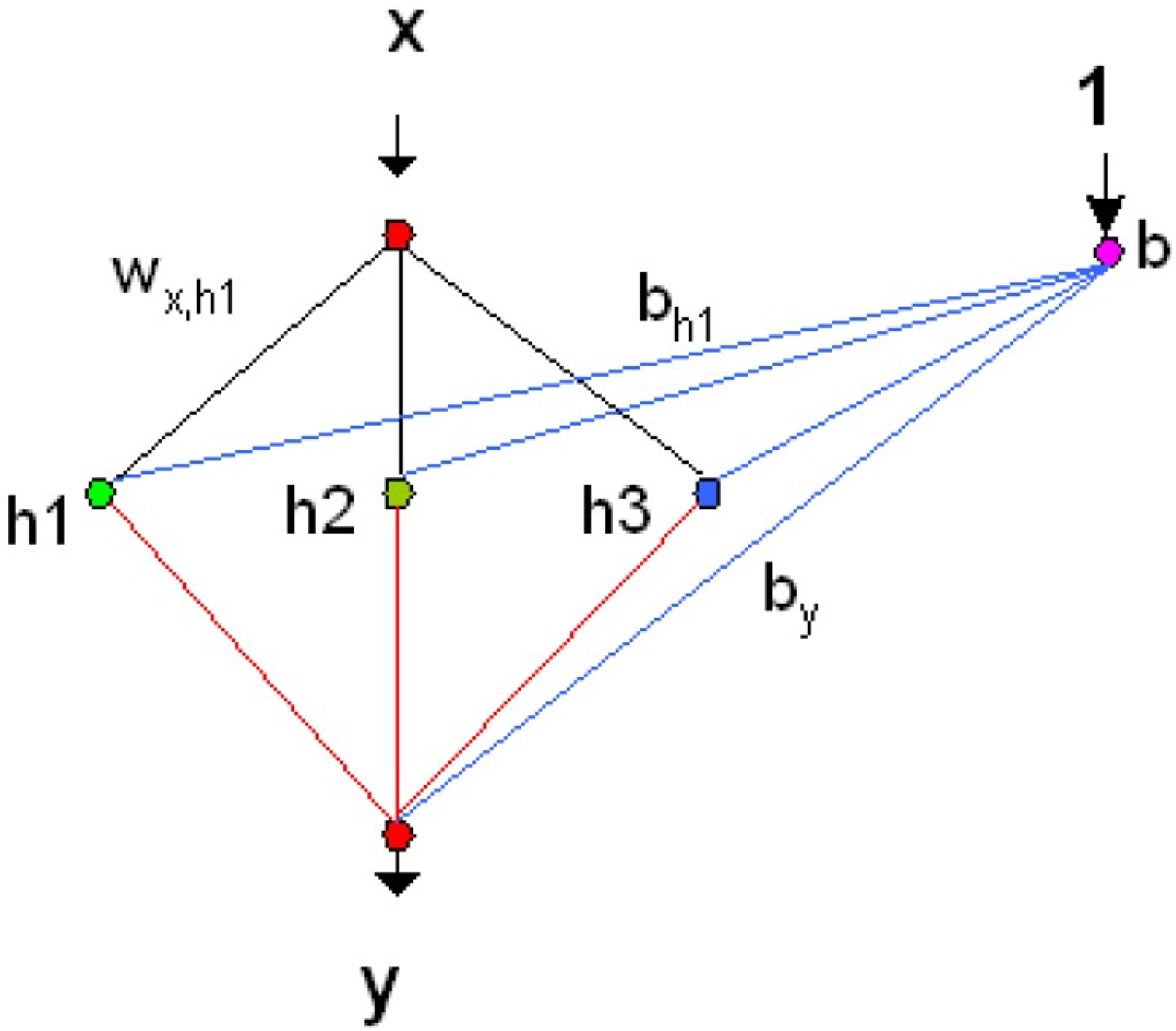
$$y(x) = \sin(x)\cos\left(\frac{x}{4}\right)$$



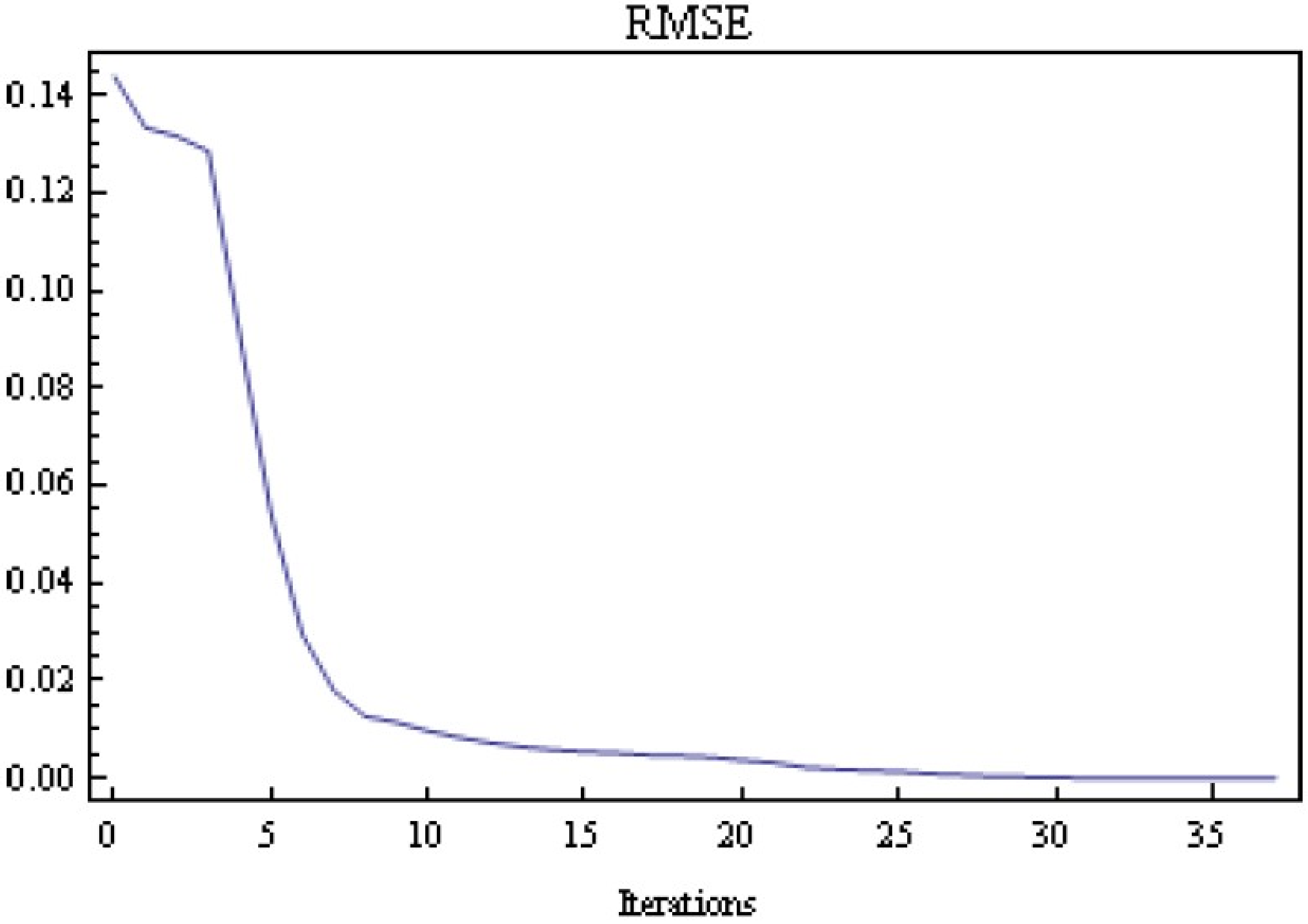
Tanításhoz használt pontok



Hálózati topológia



Hibafüggvény a tanítás során

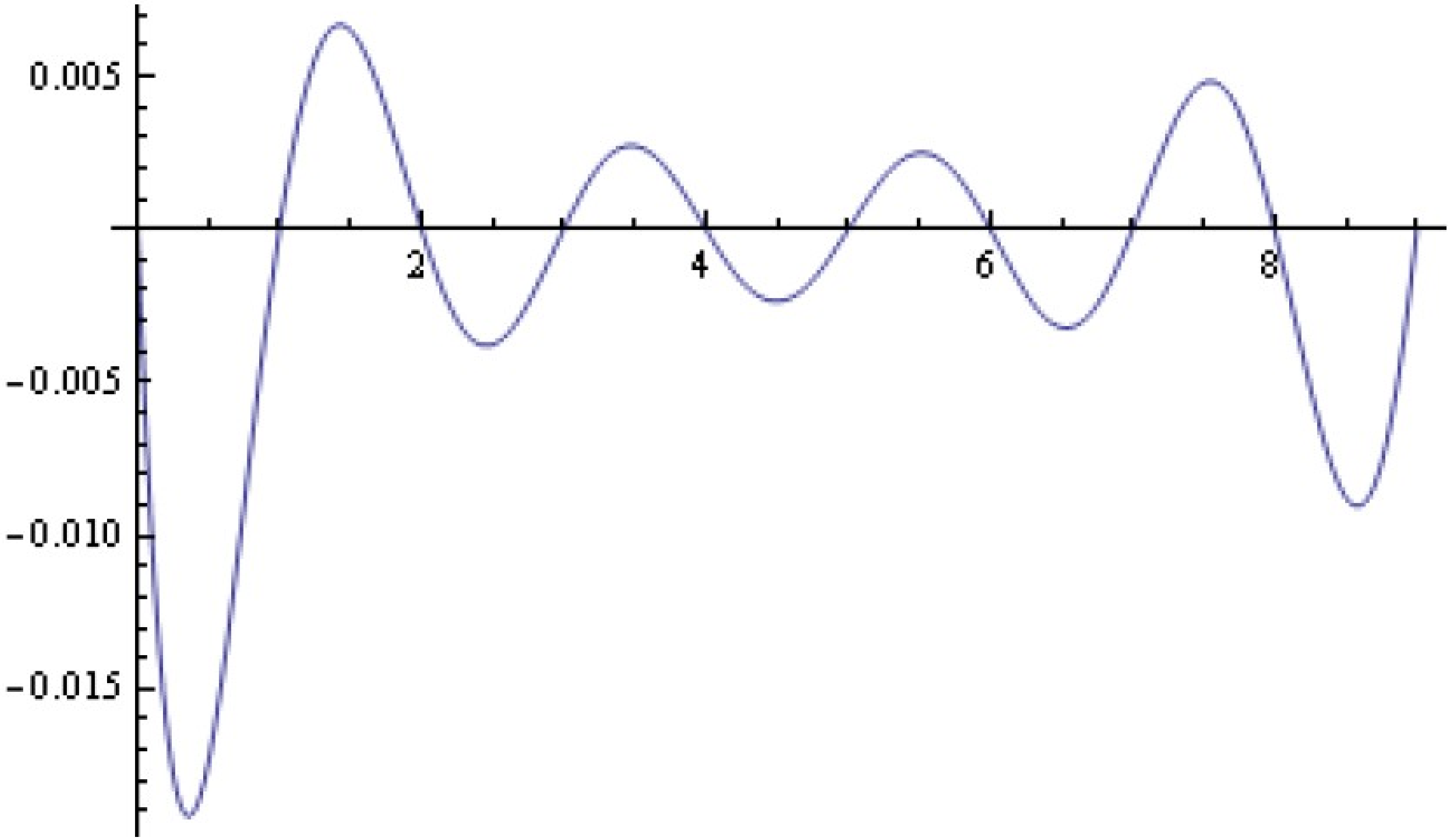


A függvény közelítése

- A tanítás utáni **súlyok**:
  - Bemeneti, kimeneti súlyok, bias
- **Aktivációs függvény** behelyettesítése:
  - hálózat leképzése
- A **hálózat leképzése** adja a **függvény közelítés** analitikus alakját:

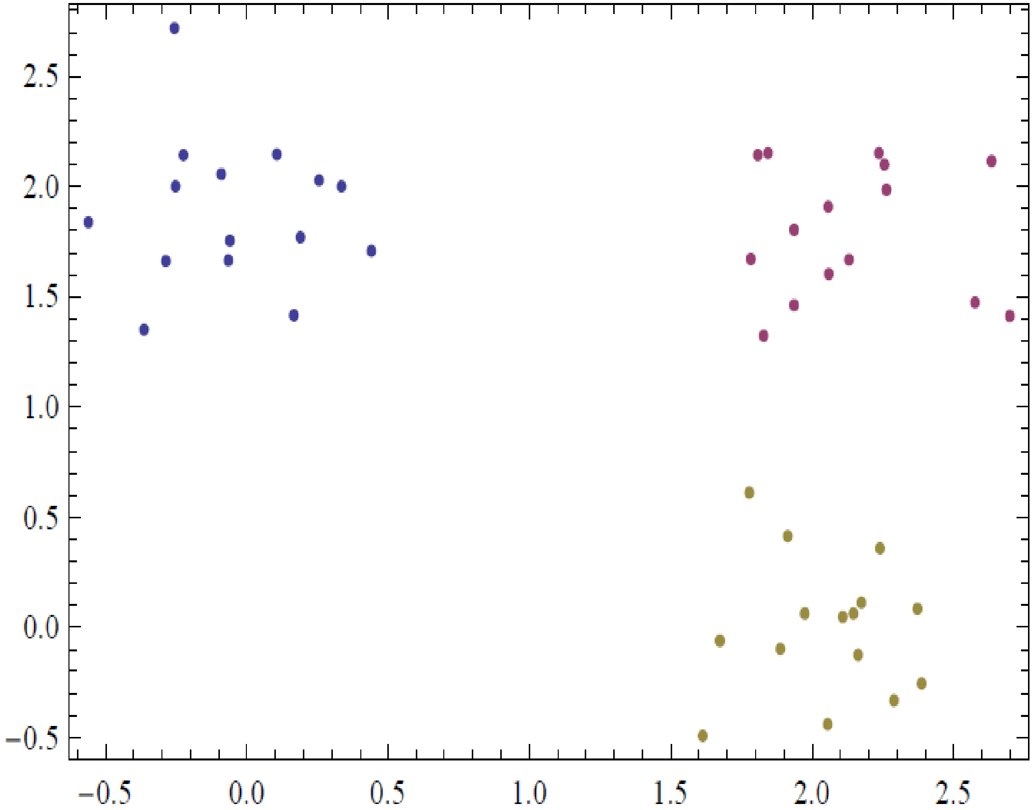
$$y = 24.7998 - \frac{5.88779}{1 + e^{8.10076 - 1.10901x}} - \frac{9.85332}{1 + e^{2.69175 - 0.990433x}} - \frac{44.0739}{1 + e^{-0.194443 + 0.156156x}}$$

Lokális hiba változása

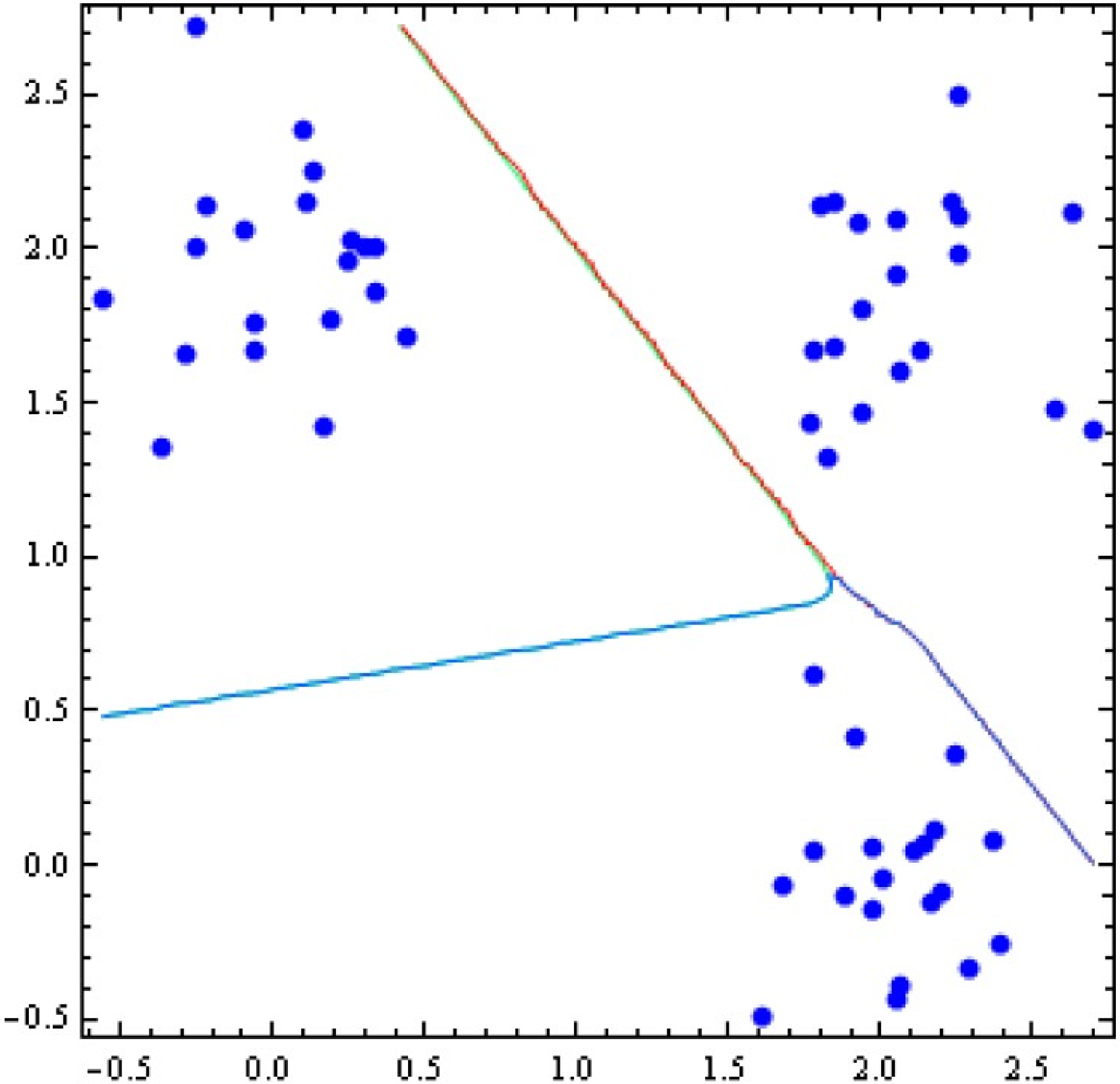


Nemlineáris osztályozási példa

- Hálózat: *Backpropagation* hálózat *sigmoid* aktivációs függvénnyel
- Topológia:
  - 2 rejtett réteg
  - 1 réteg: 3 neuron
  - 2. réteg: 5 neuron
- Osztályozandó pontok:



Tanítás eredménye



Backpropagation hálózat tanítása

- **Backpropagation** módszer
  - Hibafüggvény minimumkeresése
    - Legmeredekebb csökkenés módszere
    - Második derivált konstans
  - Szigmoid aktivizálós függvény:
    - Deriváltja kifejezhető magával a függvénnyel

Tanítás módszere: rejtett réteg nélküli hálózat esetén

- Delta szabály (rejtett réteg nélküli hálózat esetén)

$$R(w) = \left(y - \sigma(wx)\right)^2$$

$$w_{i+1} = w_i - \left(\frac{dR(w)}{dw}\right)_{w=w_i} \xi$$

$$w_{i+1} = w_i + 2\left(y - \sigma(w_i x)\right)\left(1 - \sigma(w_i x)\right)\sigma(w_i x) x \xi$$

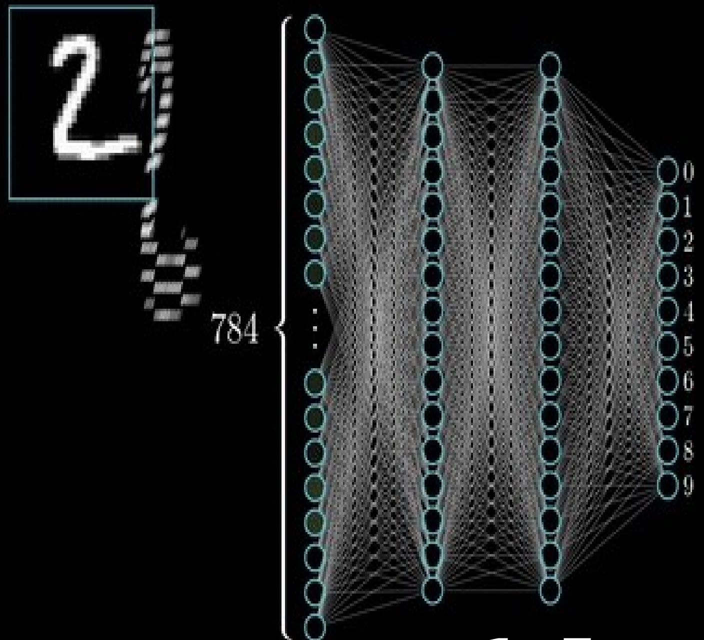
$$\Rightarrow$$

$$w_{i+1} = w_i + \eta \delta_i x$$

$$\delta_i = \left(y - y_i\right)\left(1 - y_i\right) y_i$$

$$y_i = \sigma\left(w_i x\right)$$

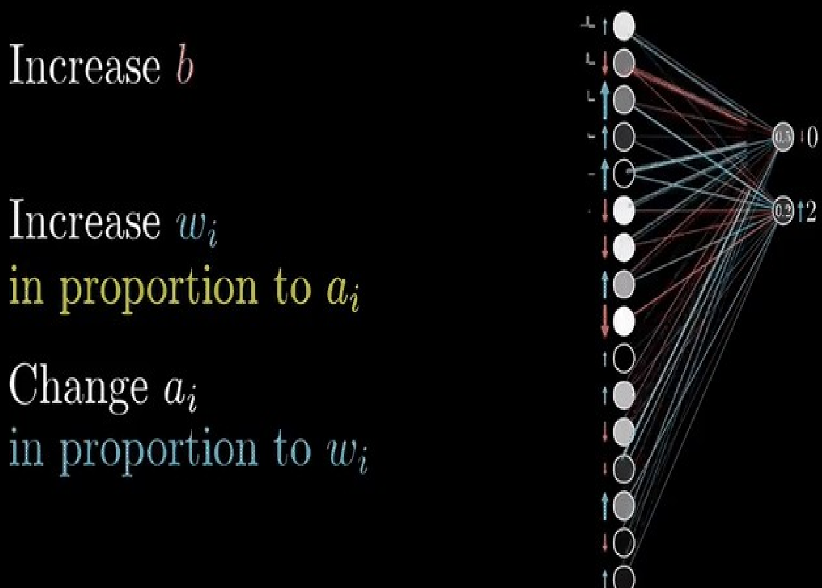
- R(w): négyzetes hibafüggvény (*ezt szeretnénk iteratív algoritmussal minimalizálni*)
- W(i+1): alkalmazandó súly az (i+1)-dik iterációban
  - legmeredekebb csökkentés módszerével számoljuk, második deriváltat konstansnak feltételezve
- W(i): súly az (i)-dik iterációban ($\eta = 2\xi$)
- y(i): az i-dik iterációban számolt súllyal a kimenet
- y az elvárt kimenet x bemenet esetén



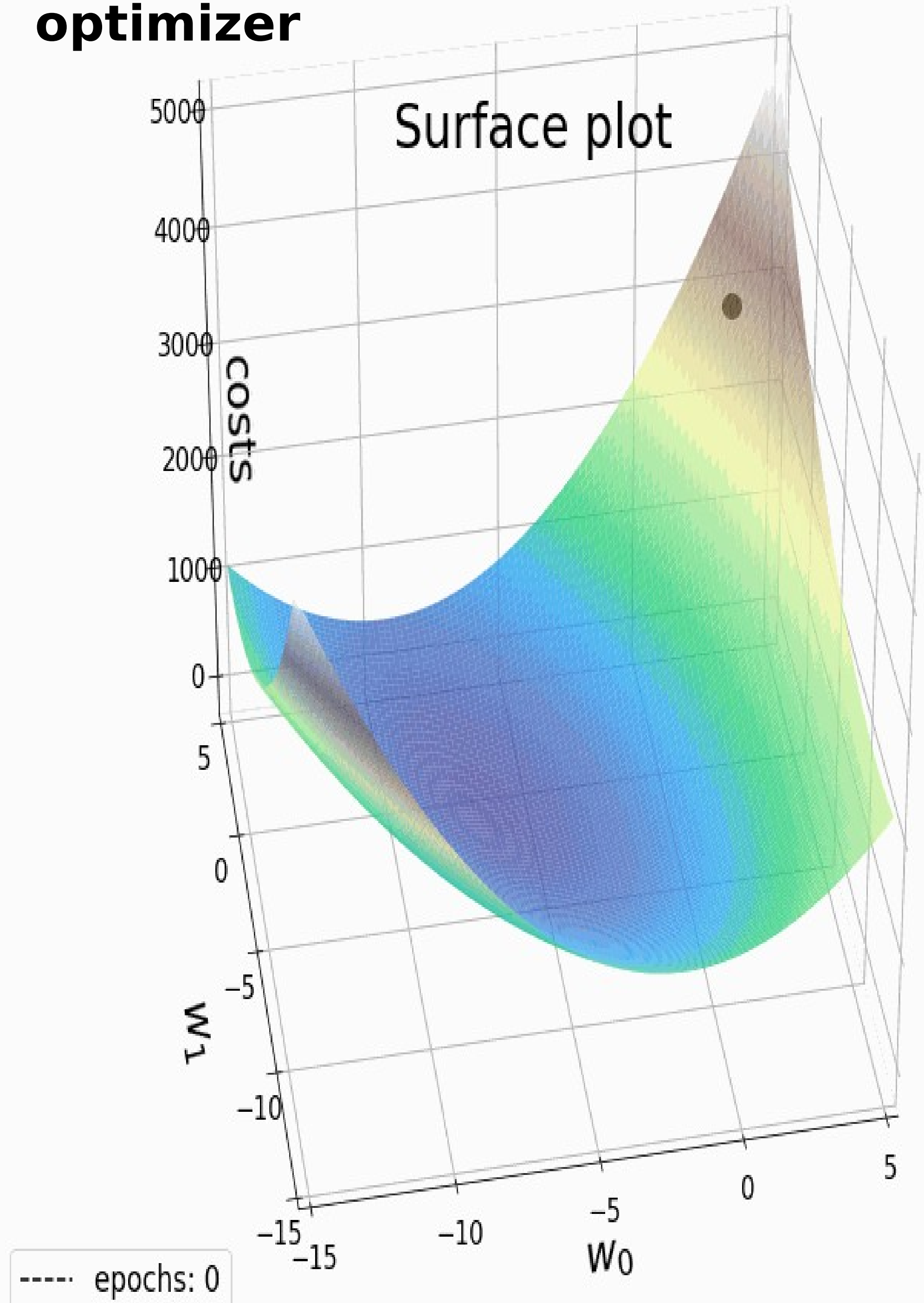
1. Forward

2. Backpropagation

Taking the derivative of the loss with respect to each weight parameters in the network.



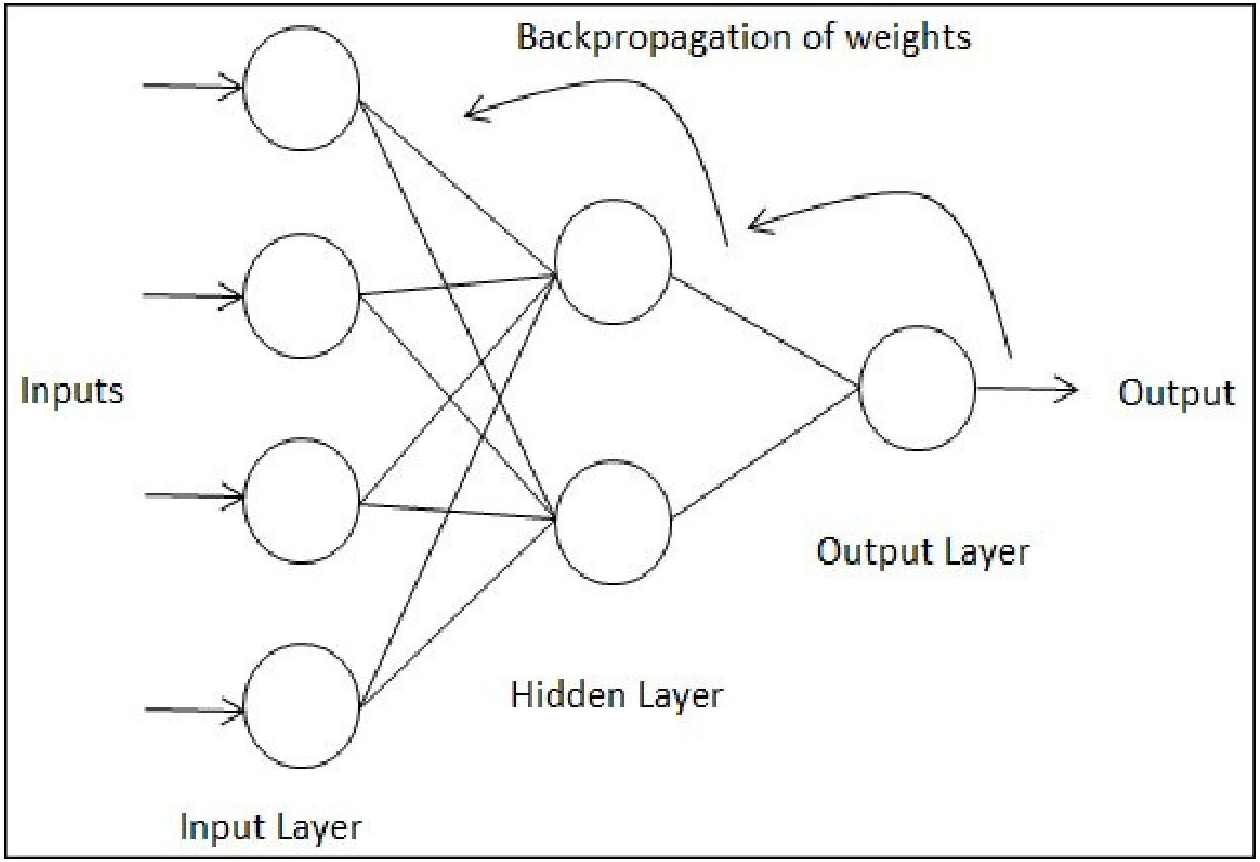
3. Stepping the optimizer



Backpropagation hálózat tanítása: általános eset

- **Backpropagation** módszer
  - Általánosított tanuló eljárás
- **Delta szabály** alapján
  - Szigmoid aktivizálós függvényt alkalmazó hálózatok esetén
- Súlyok meghatározása akár nagyméretű nemlineáris optimalizációs problémák esetén
- **Lépések:**
  - **A) Propagation**: bemenet x(i) → w(i) → kimenet: y(i)
  - **B) Backpropagation**: w(i+1) ← w(i) ← (y-y(i))
  - Több réteg esetén a sorrend:
    - output rétegtől az input réteg felé

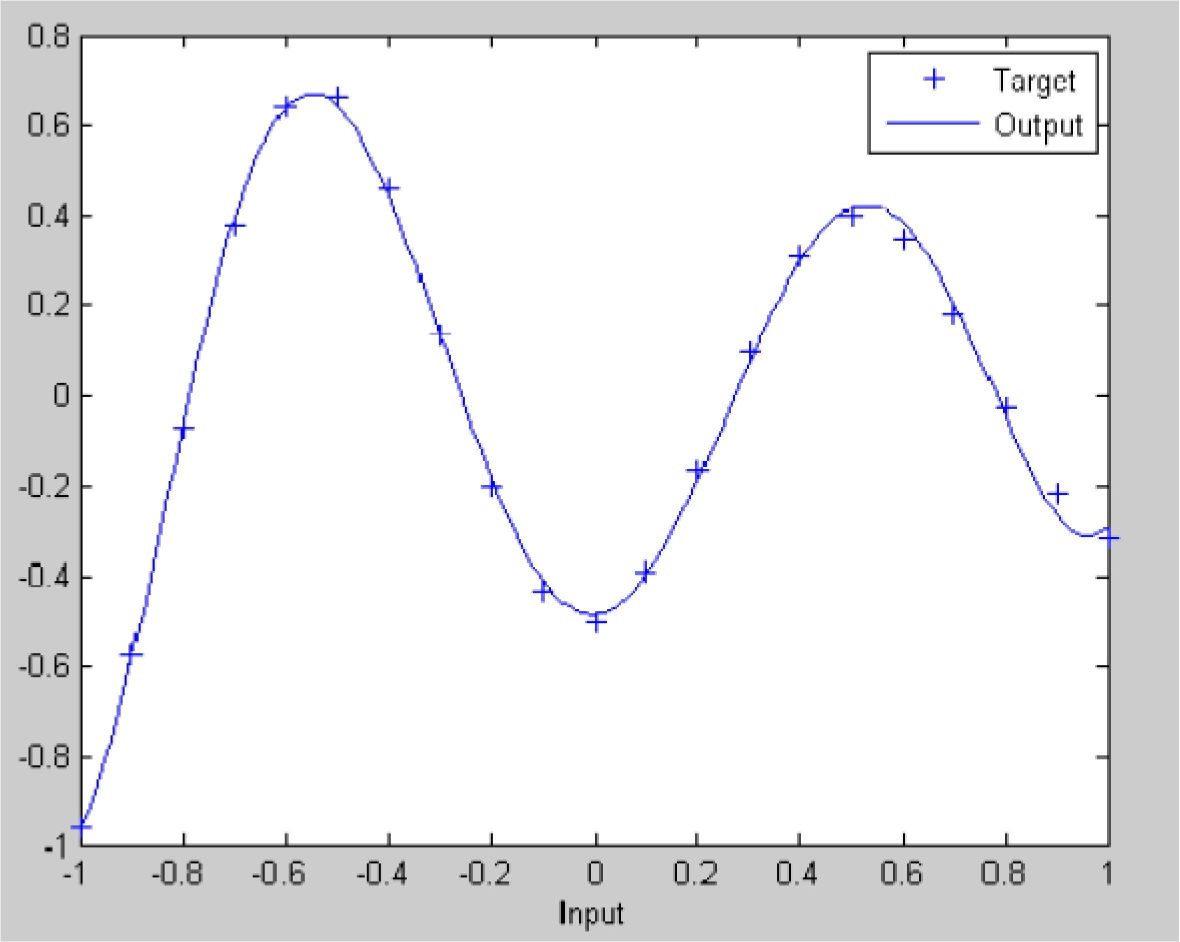
Backpropagation alkalmazása több réteg esetén



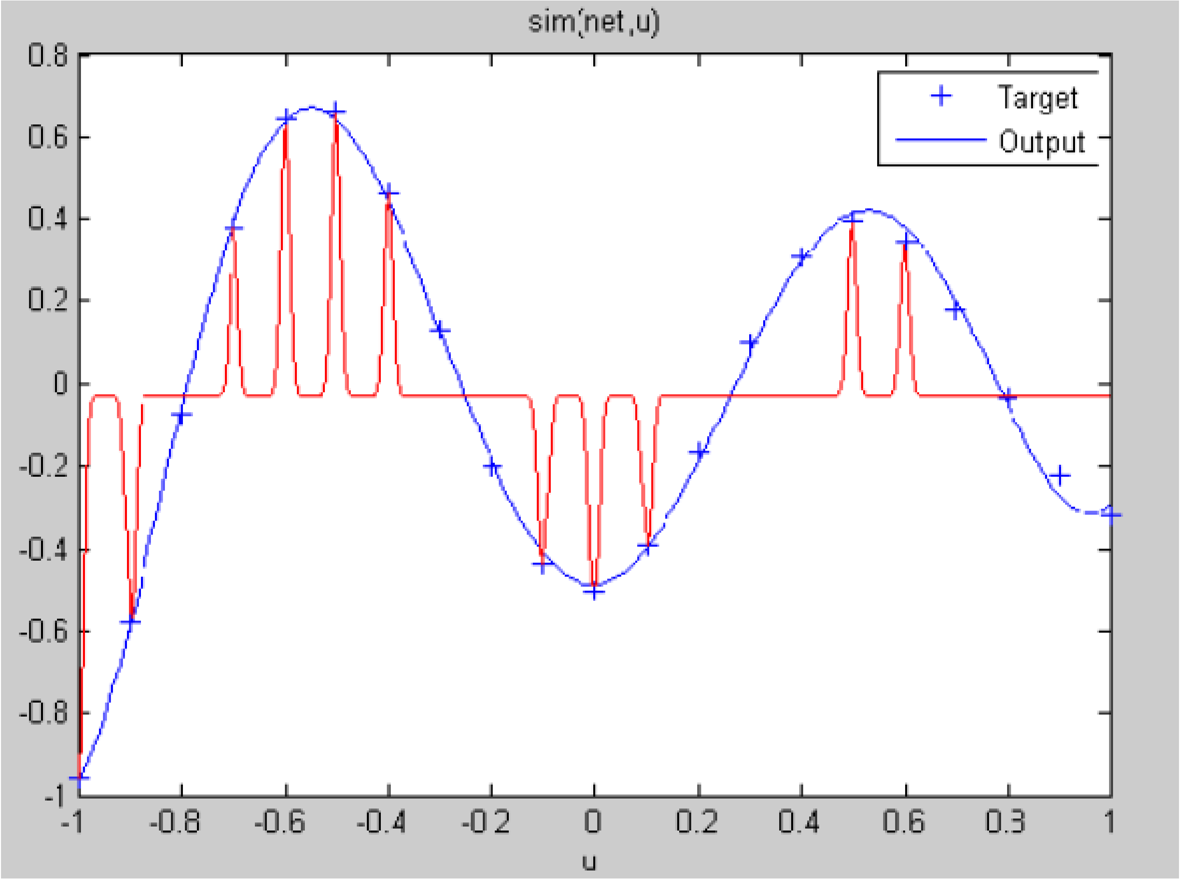
Mesterséges neurális hálózatok:

Hálózat tanításának általános problémái

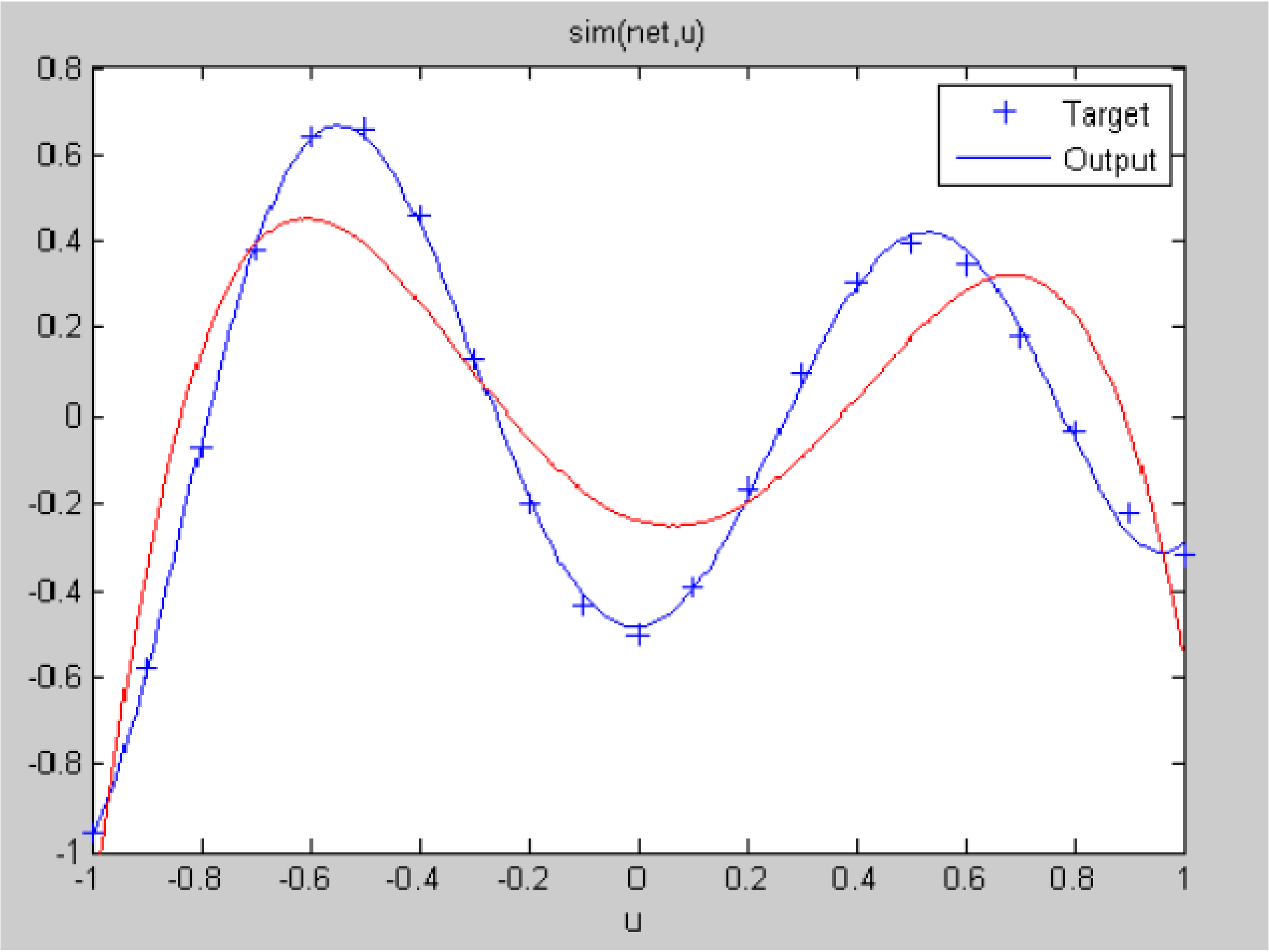
Közelítendő függvény



„Túltanulás” esete



„Alultanulás” esete



Megfelelő tanulás biztosítása: módszerek I.

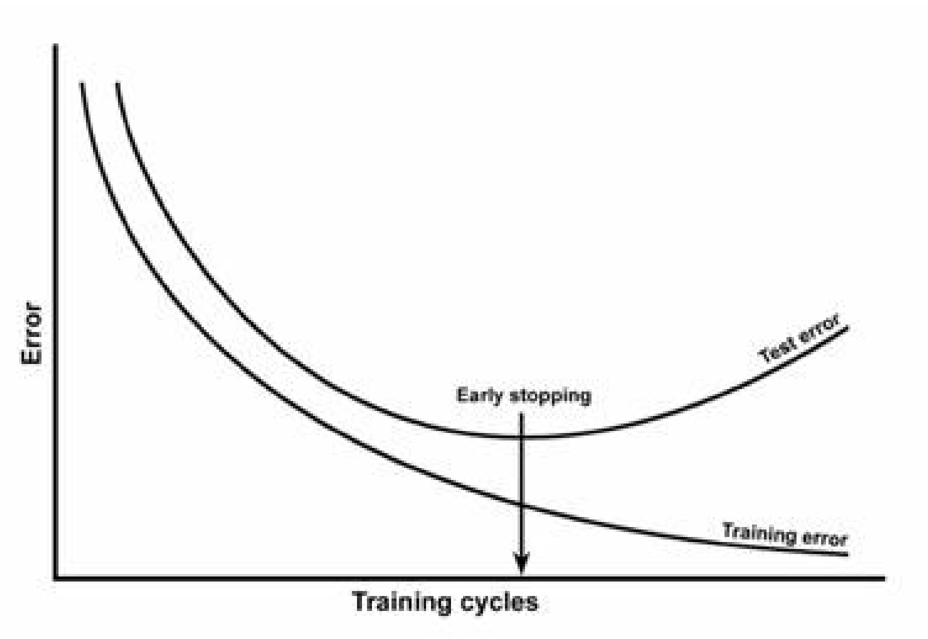
A) Alul-, ill. túltanulás elkerülése **megfelelő hálózati topológia** választásával:

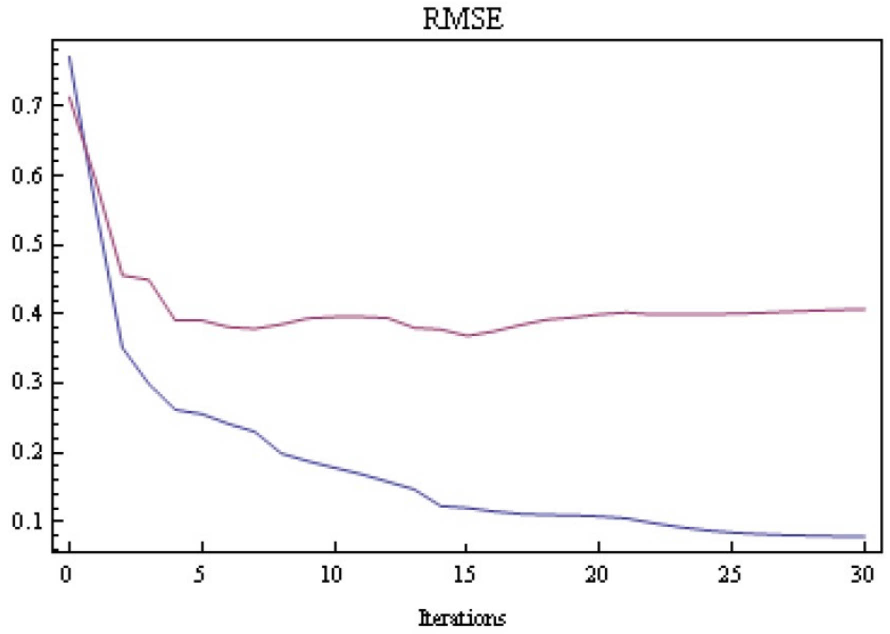
- megfelelően választott rejtett rétegbeli csomópont-szám;
- tanulás után validáció.

B) Korai leállás módszere

- **tanuló-** és egy **validációs** halmaz,
- addig tanítunk, amíg a validációs halmazon **is csökken** a hiba.

Korai leállás módszere: hiba a tanító és a validációs halmazon





Megfelelő tanulás biztosítása: módszerek II.

C) Regularizáció a túltanulás elkerülésére

- Túltanulás esetén a leképzés gradiense általában nagy
- A hibafüggvény megváltoztatása
  - **Cél 1)** hálózat **hibájának** csökkentése

$$R(w)=\sum_{j=1}^{n}\left(y_j-\sigma\left(w^T x_j+b\right)\right)^2$$

  - **Cél 2)** hálózat által megvalósított függvény **gradiensének minimalizálása**

$$R(w)=\sum_{j=1}^{n}\left(y_j-\sigma\left(w^T x_j+b\right)\right)^2+\delta_w w^T w$$

Radiál Bázis Függvény (RBF) hálózatok

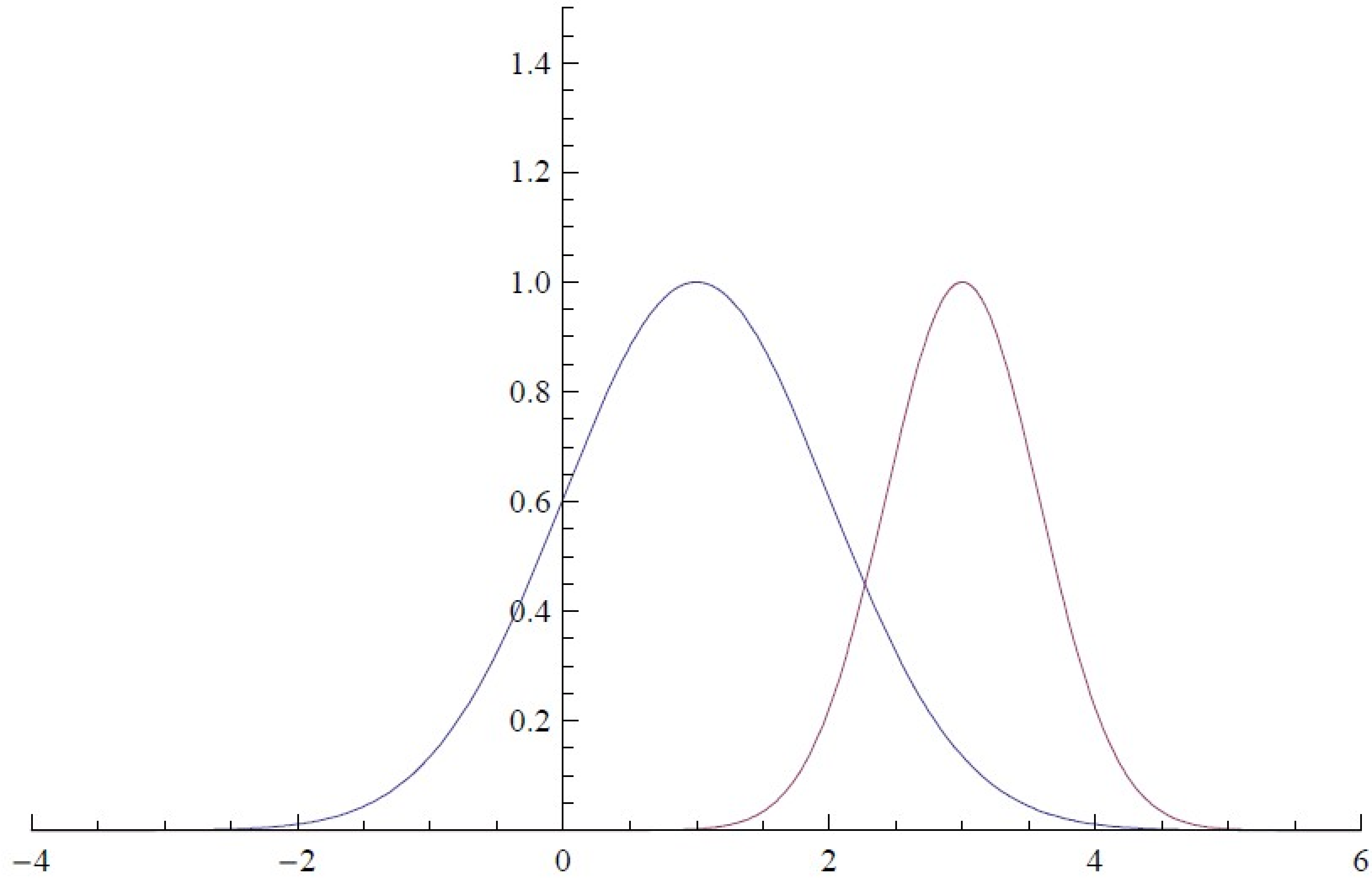
RBF hálózat

- Aktivációs függvény
  - **Radiál Bázis Függvény**

$$y(x) = e^{-\lambda(x-c)^2}$$

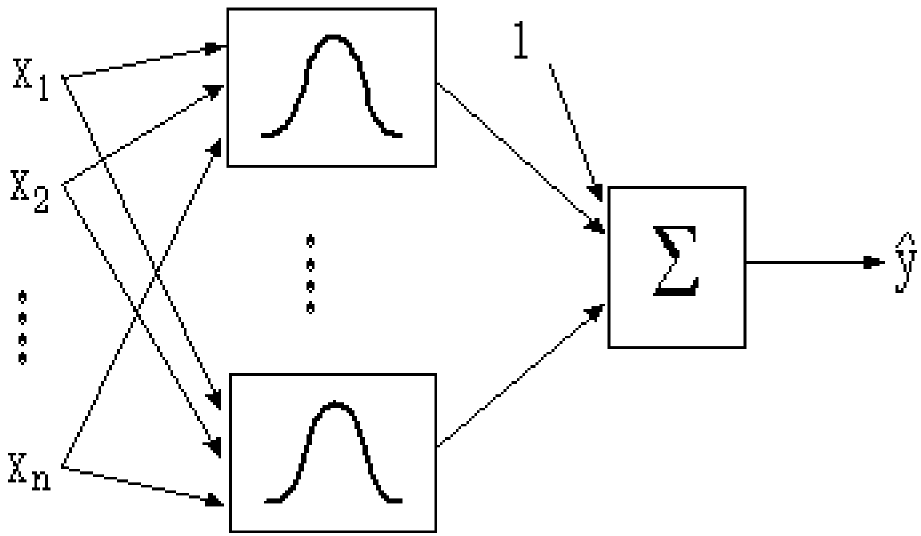
  - Harang alakú görbe
  - Paraméterek:
    - λ harang „nyílásszöge”, c középpontja (~eltolás)
  - Jól alkalmazható hálózat **függvény közelítésre**.

Példák RBF aktivációs függvényre



Általános RBF hálózat felépítése

- **Nincs rejtett** réteg
- Rétegek:
  - **Bemeneti réteg:**
    - *n* db csomópont (n bemenet)
  - **Aktív réteg:**
    - *nb* db csomópont
- Bias csomópont
  - Súlyozható eltolás a kimeneten
- Egy kimeneti csomópont
  - Az aktív réteg kimenetek (és a bias) összege
- Lehetséges lineáris tag



RBF példa: kétváltozós függvény közelítése

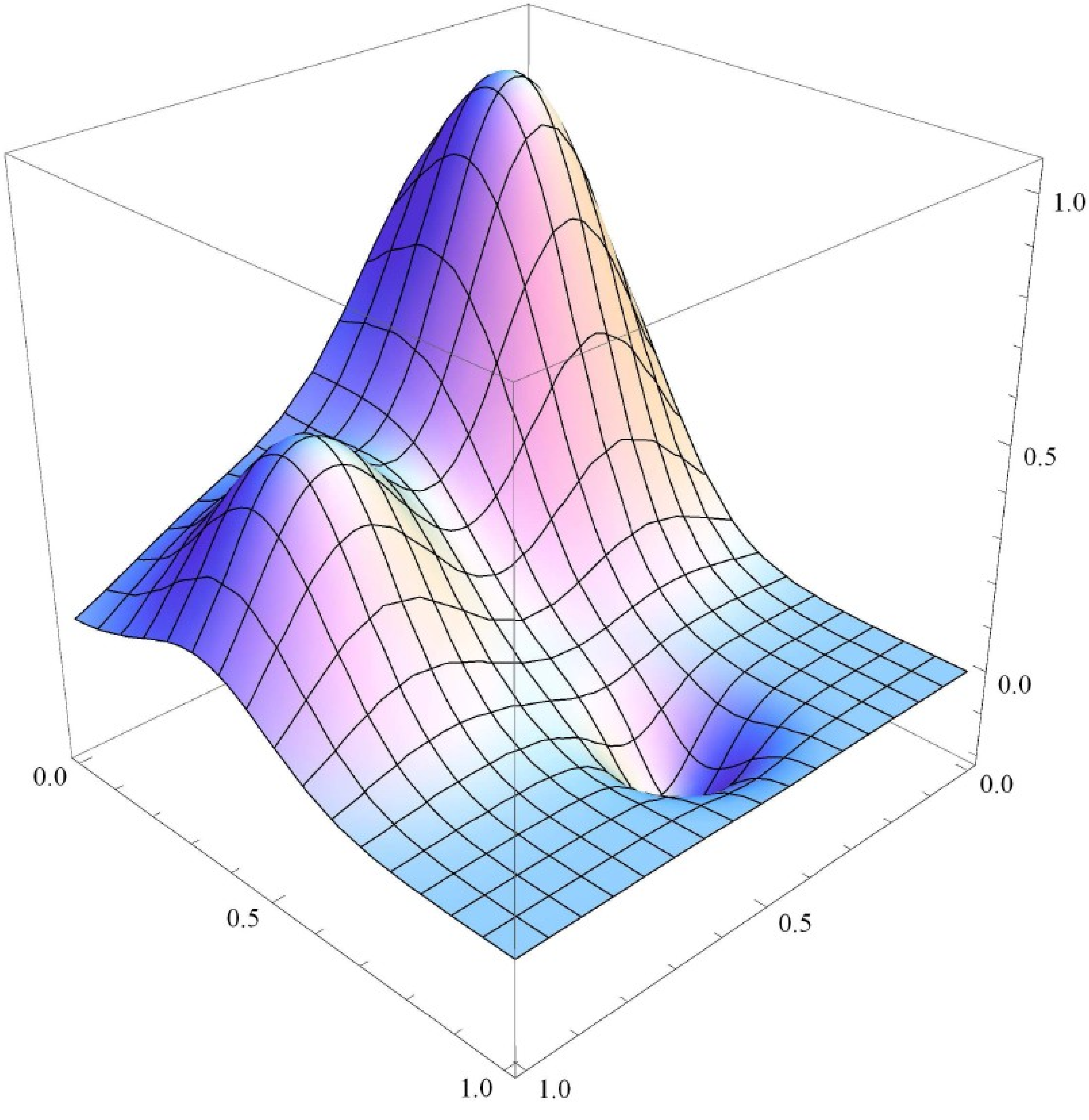
Kétváltozós függvény közelítése

- Közelítendő függvény: *Franke teszt függvény*

$$f(x, y) = \frac{3}{4} e^{-\frac{1}{4}\left((9x-2)^2+(9y-2)^2\right)} + \frac{3}{4} e^{-\frac{1}{49}(9x+1)^2-\frac{1}{10}(9y+1)^2} +$$

$$\frac{1}{2} e^{-\frac{1}{4}\left((9x-7)^2+(9y-3)^2\right)} - \frac{1}{5} e^{-(9x-4)^2-(9y-7)^2}$$

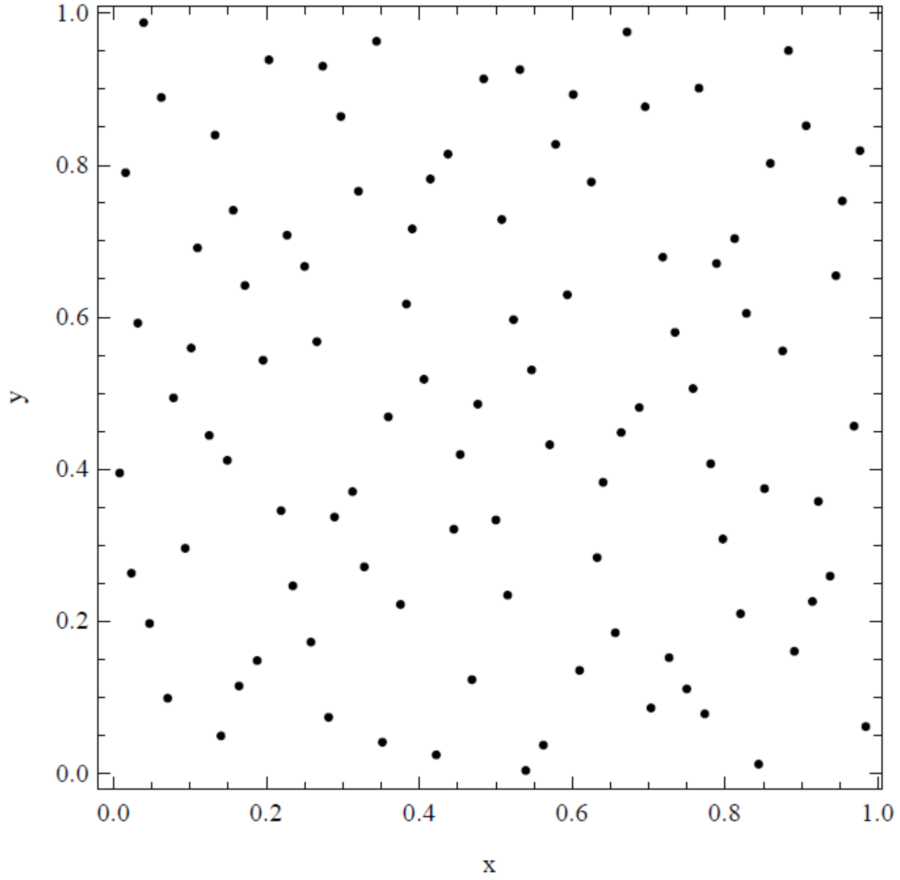
Közelítendő függvény



Mintavételi pontok választása

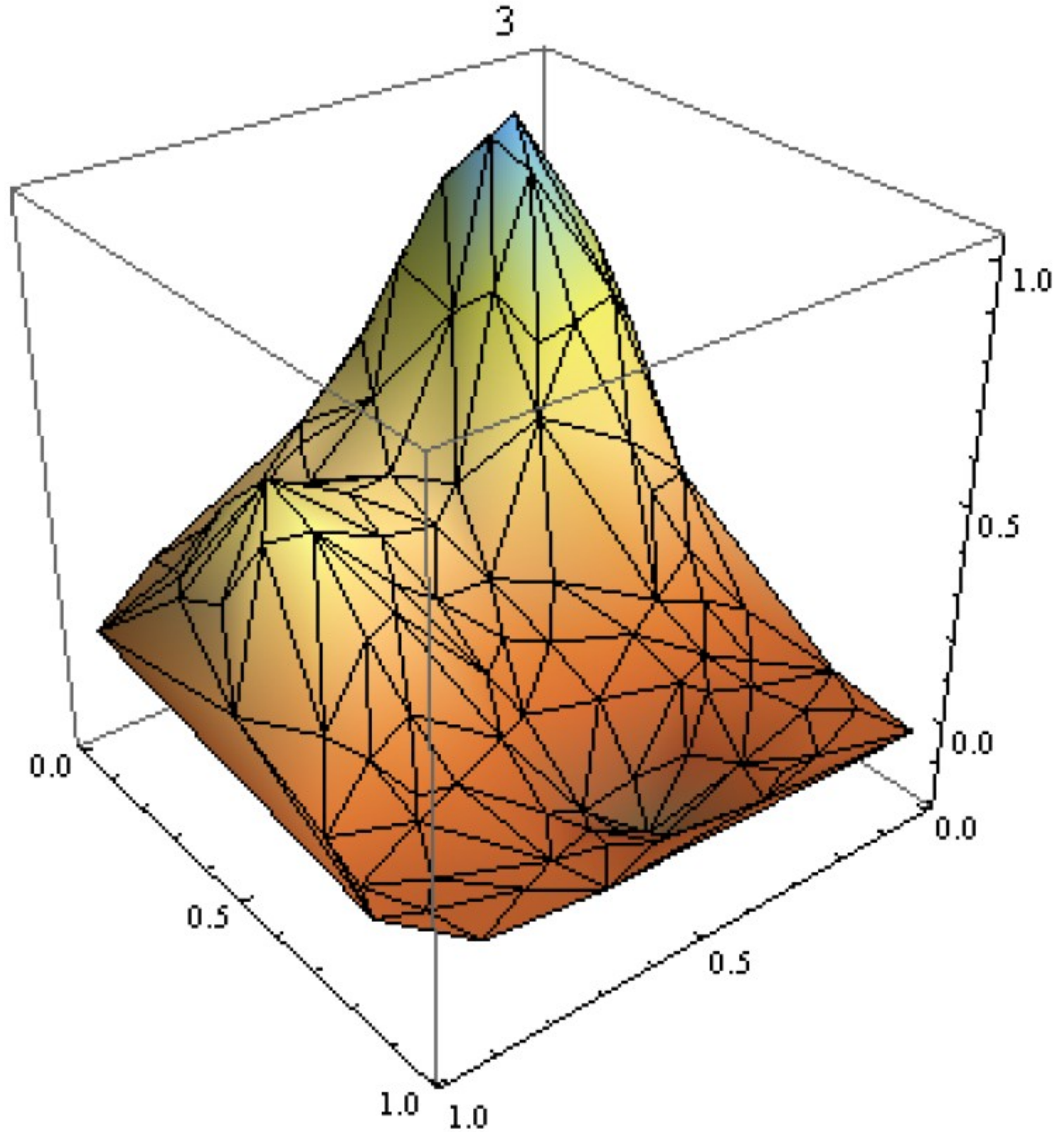
- Valóság: általában **nem szabályosan mintavételezett** ismert mintáim vannak
- Cél:
  - „egyenletes” mintavételezés
- **Pszeudo-random ponthalmaz**
  - Gyakran vannak „csomósodási” pontok
- ***Halton-féle* pontok**:
  - egyenletesen elosztott pontok egy adott tartományban

$$x, y \in [0, 1]$$

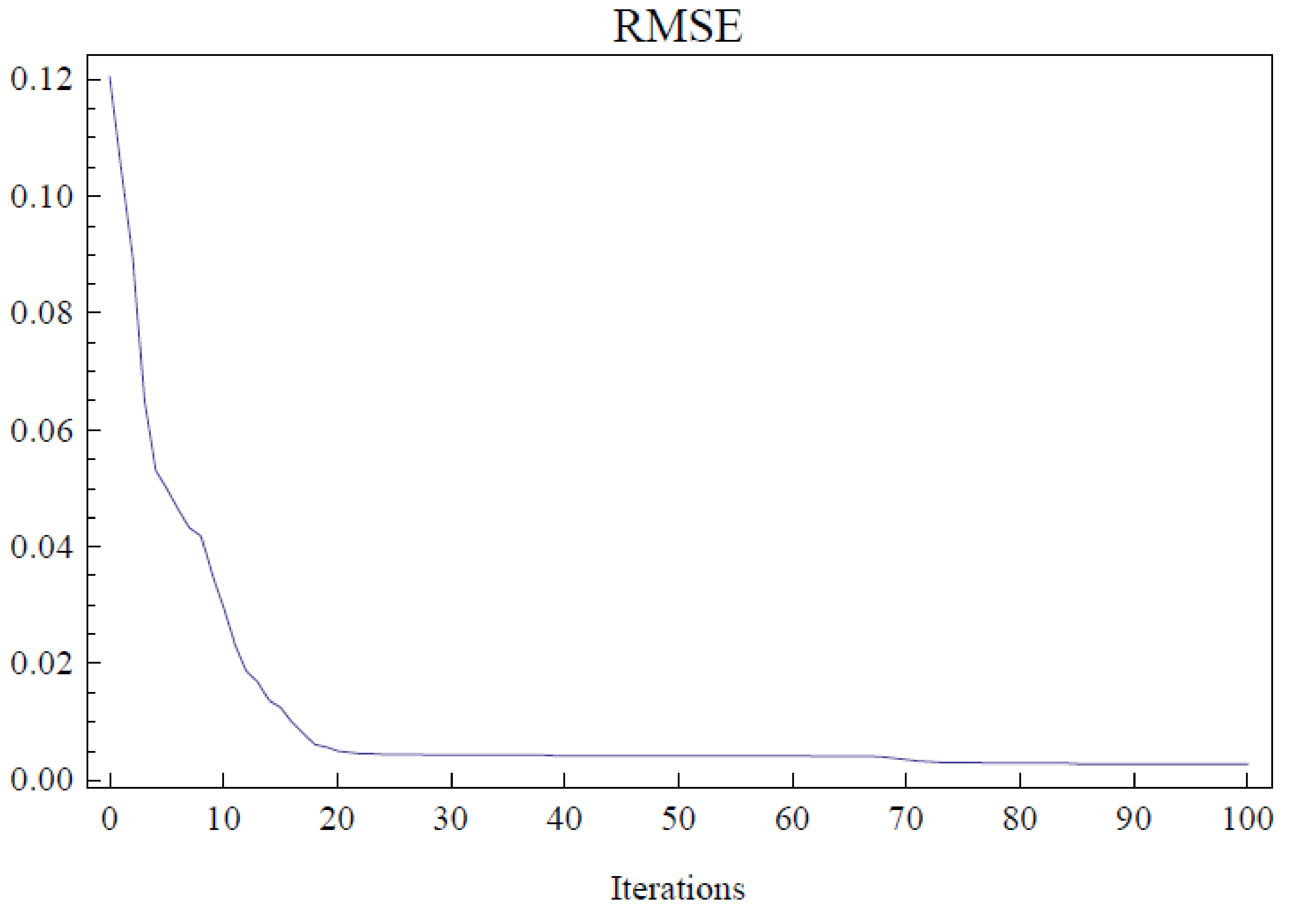


Háromszögfelbontáson alapuló spline függvény közelítés eredménye

$x, y \in [0, 1]$



A hiba alakulása a tanítás során

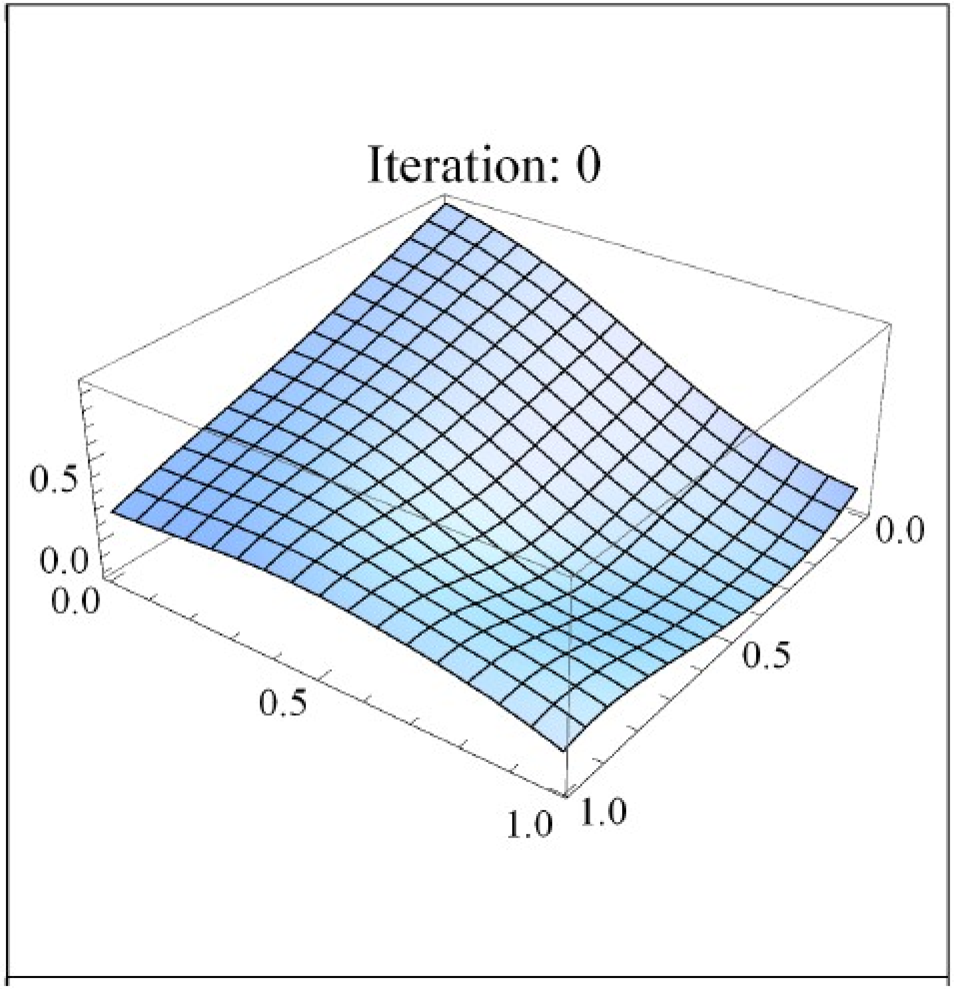


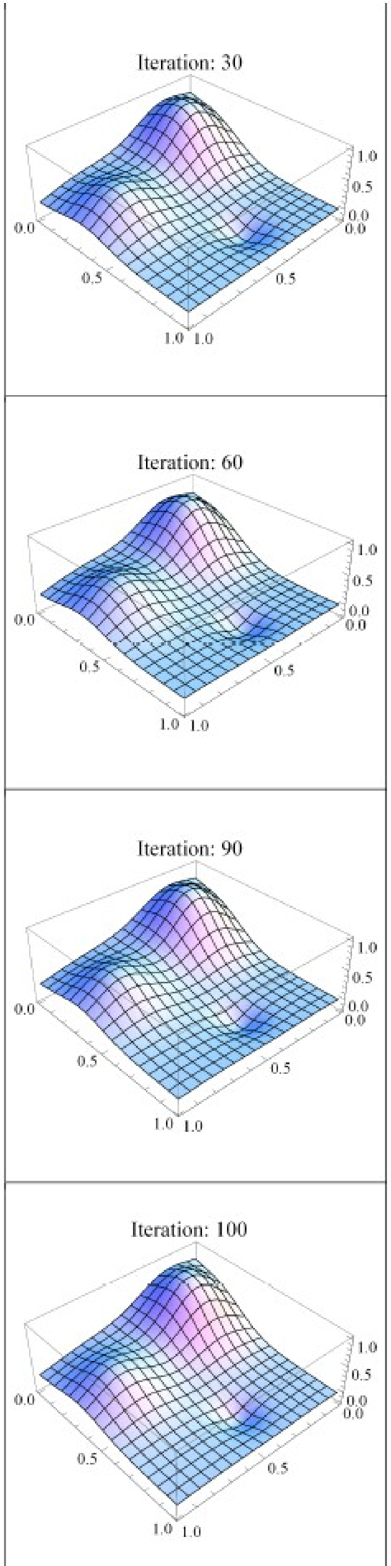
A tanítás eredményeként kapott leképzés

- Hálózati topológia:
  - 6 csomópont
- Leképzés:

$$
\begin{aligned}
&-15.3348-392.858\,e^{-1.81497\,(0.412253+x)^2-1.81497\,(-0.795555+y)^2}+\\
&388.71\,e^{-1.82951\,(0.409158+x)^2-1.82951\,(-0.795431+y)^2}-\\
&0.187153\,e^{-94.7028\,(-0.444808+x)^2-94.7028\,(-0.775338+y)^2}+\\
&0.446418\,e^{-22.3538\,(-0.77959+x)^2-22.3538\,(-0.335169+y)^2}+\\
&0.770331\,e^{-20.2089\,(-0.216997+x)^2-20.2089\,(-0.212042+y)^2}+\\
&1237.5\,e^{-0.00746198\,(-23.8598+x)^2-0.00746198\,(0.822984+y)^2}-9.40744\,x+0.51159\,y
\end{aligned}
$$

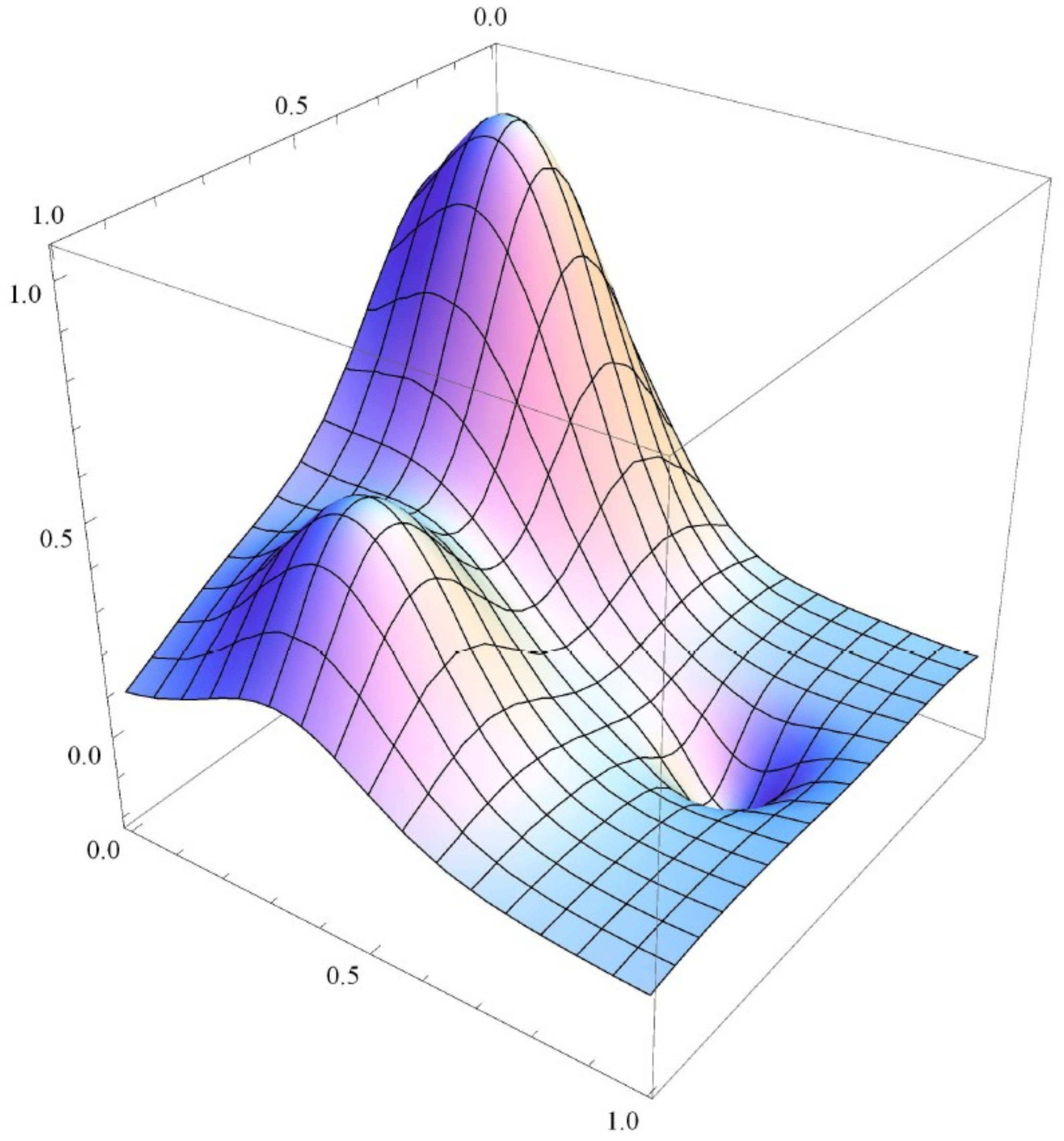
Kezdeti leképzés tanítás előtt



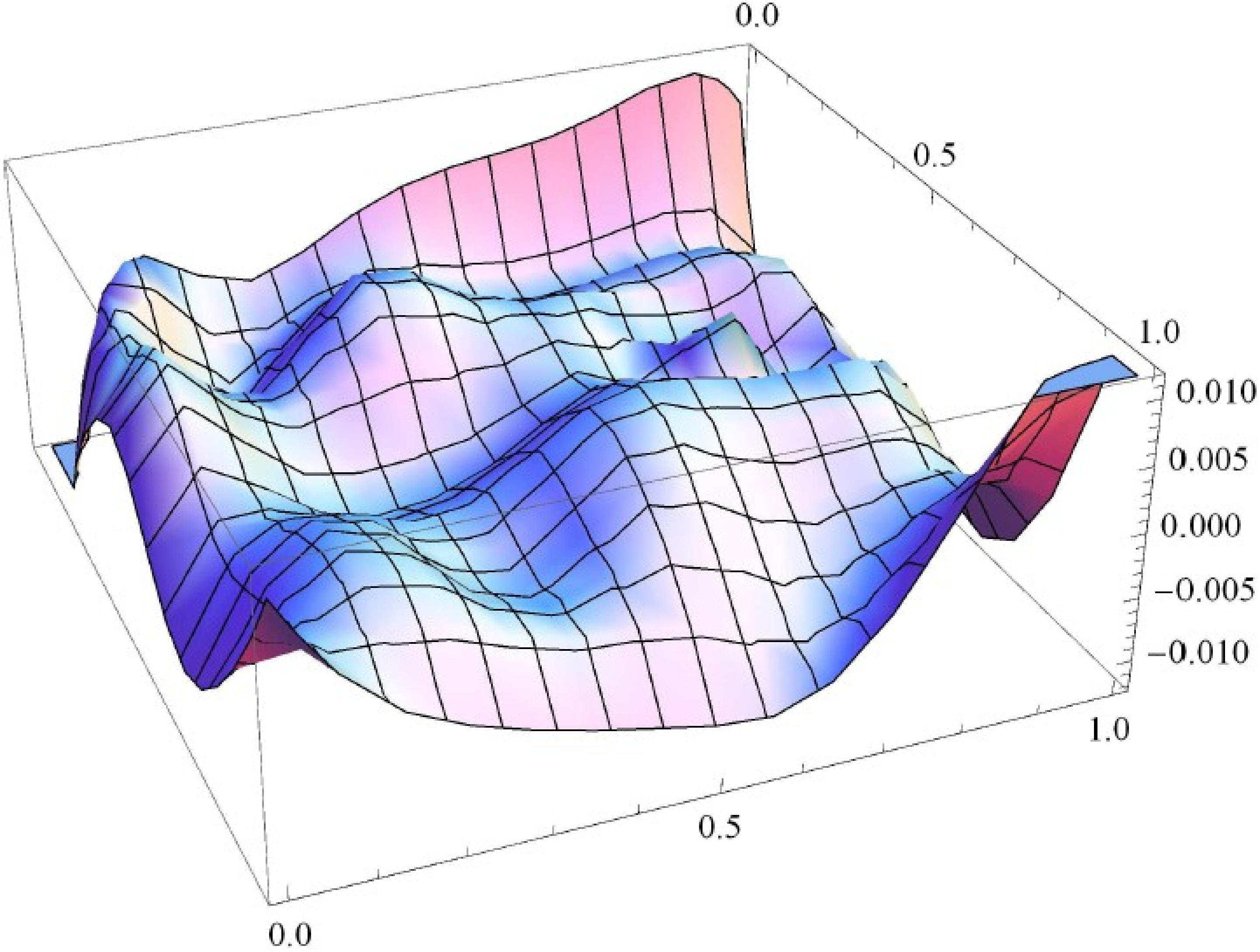


A leképzés a tanítás egyes lépései után

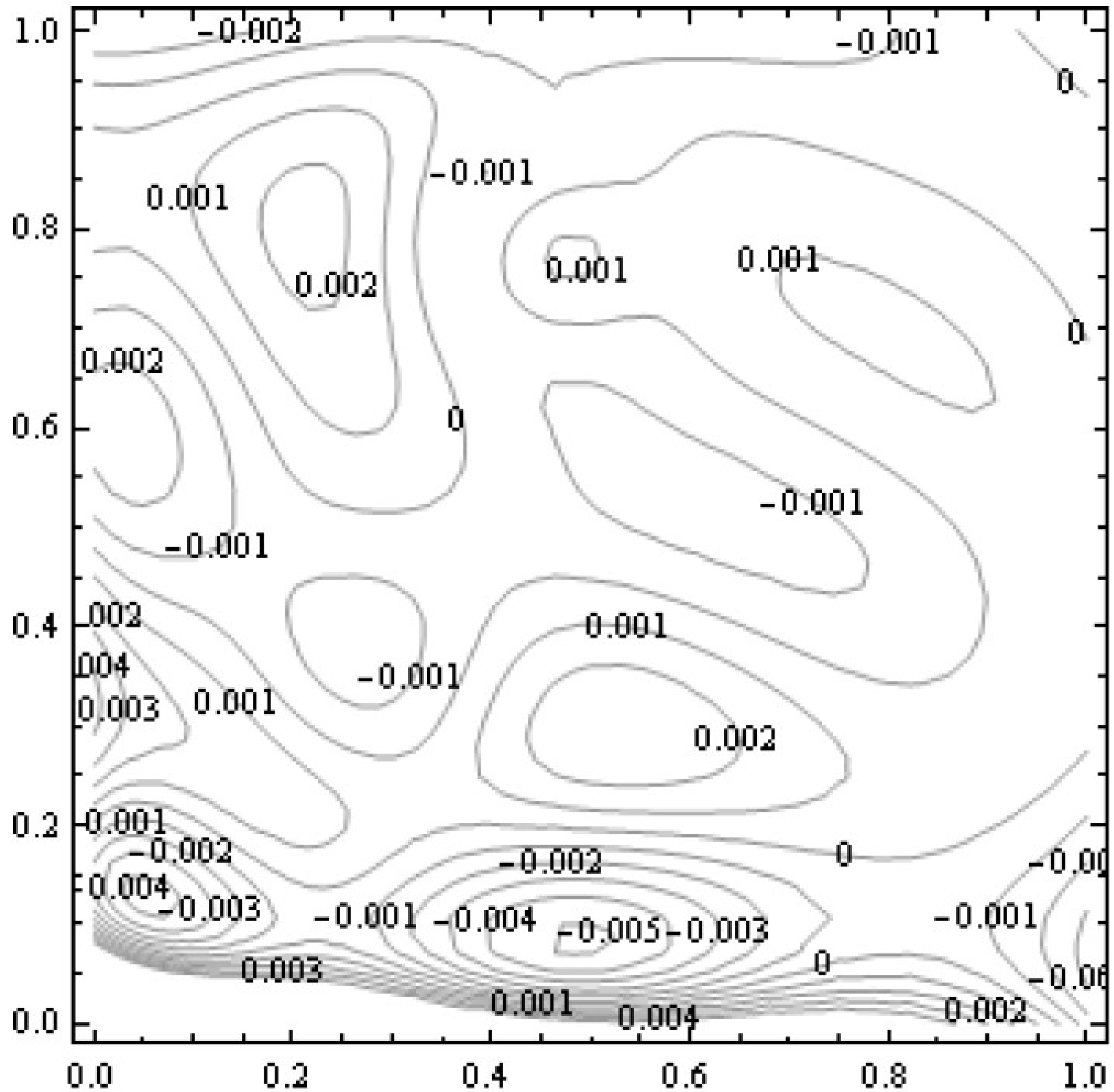
Az eredményül kapott leképzés



A leképzés hibája



A leképzés hibája



Összefoglalás

- **RBF hálózatok**
  - Jól használhatók **folytonos függvények** közelítésére
    - RBF hálózatok univerzális approximátorok
    - Megfelelő számú – és eloszlású – tanító elem esetén a hiba leszorítható
  - Hálózat tanítása:
    - Viszonylag egyszerű tanítás
    - Tanító pontokat egyszer alkalmazzuk