

Intelligens Rendszerek Elmélete

5



A mesterséges neurális hálózatok alapfogalmai és meghatározó elemei

Mesterséges neurális hálózatok felügyelt és önszervező tanítása

<http://uni-obuda.hu/users/kutor/>

TED^x Danubia 2011

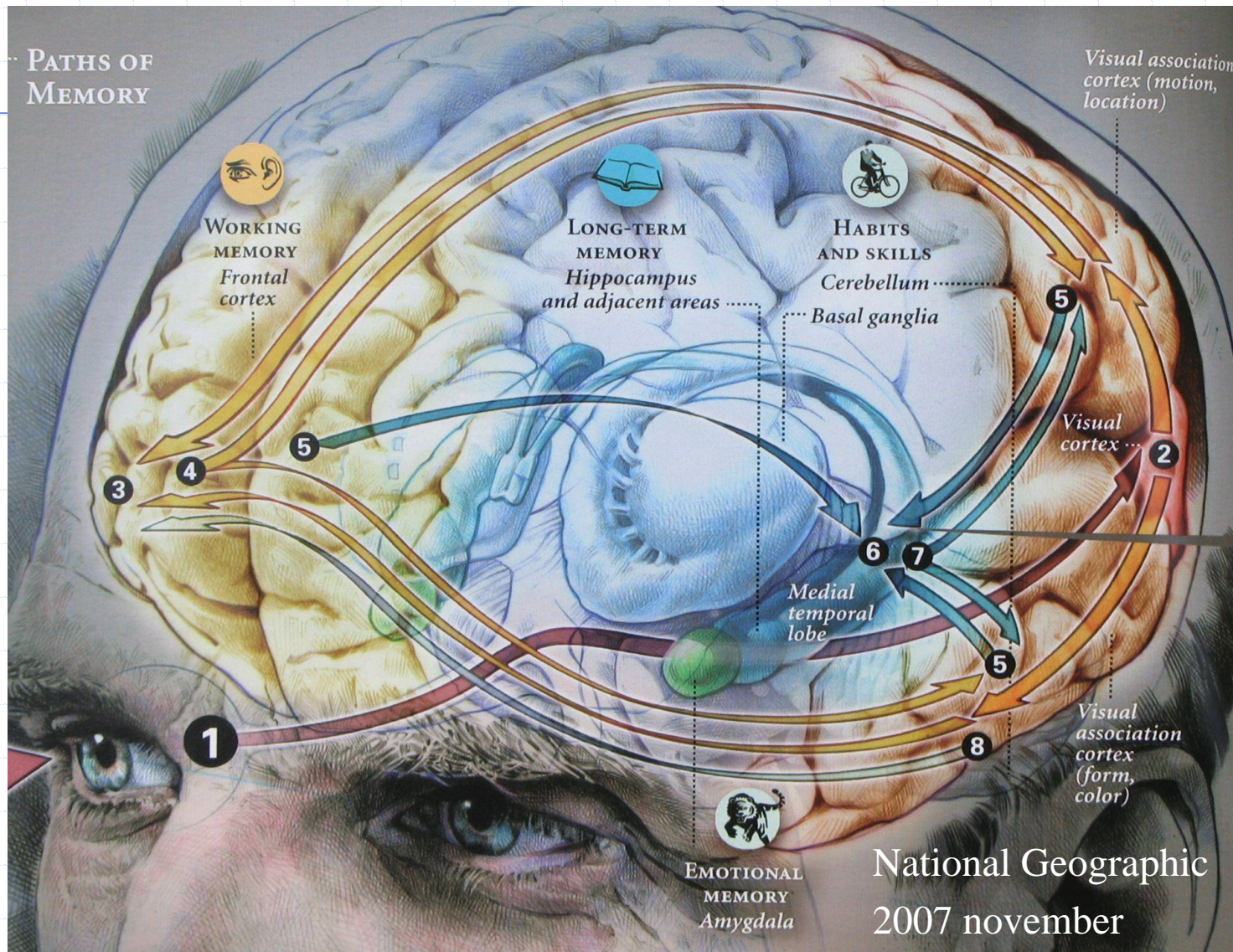
http://www.youtube.com/watch?v=6ug2OIEr1l8&feature=youtube_gdata_player



TAMÁS FREUND



Az emlékezés kapcsolatrendszer





<http://www.mindmachineproject.org/>

2016.

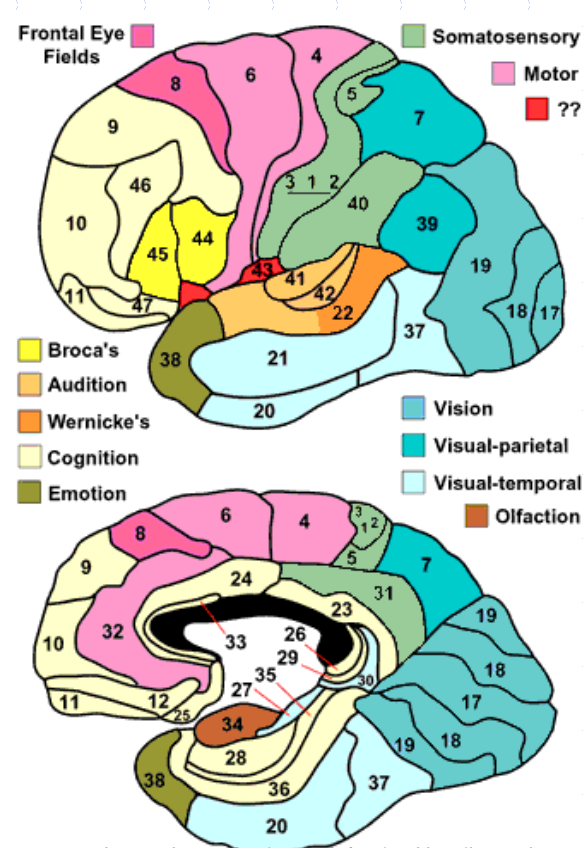
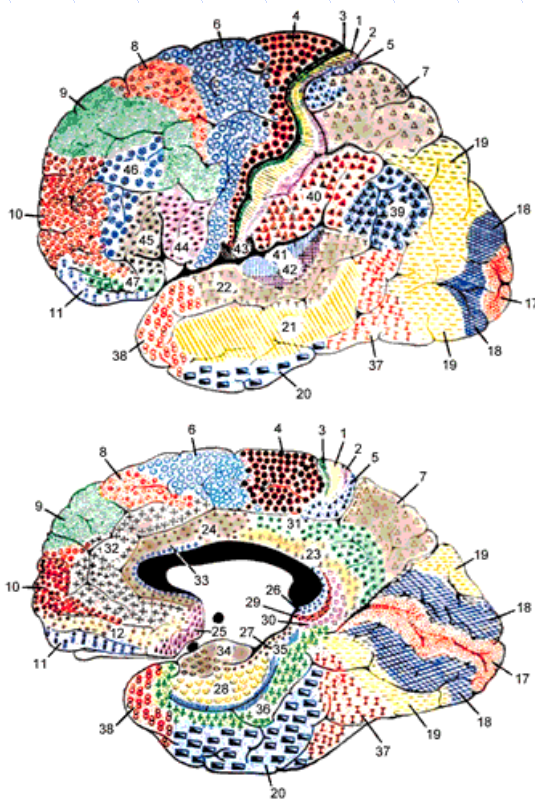
Óbudai Egyetem, NIK Dr. Kutor László

IRE 5/64/5

A természetes idegi hálózatok tanulságai

A feldolgozás nem univerzális! (A hálózat típusa határozza meg a működést!)

A működés párhuzamos és hierarchikus (hagymahéj model)



„Brodmann” agyterületek

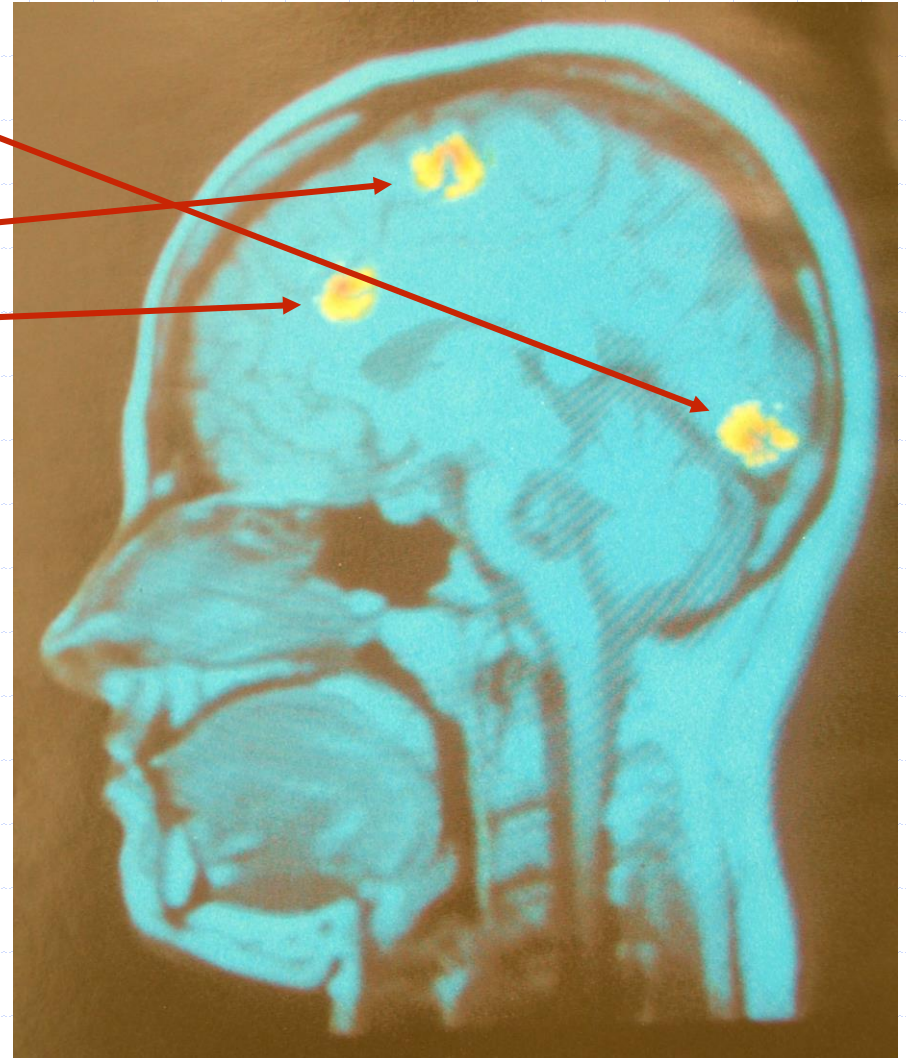
<http://spot.colorado.edu/~dubin/talks/brodmann/brodmann.html>

Az agyterületek működésének MRI + PET képe

Írott szöveg olvasása

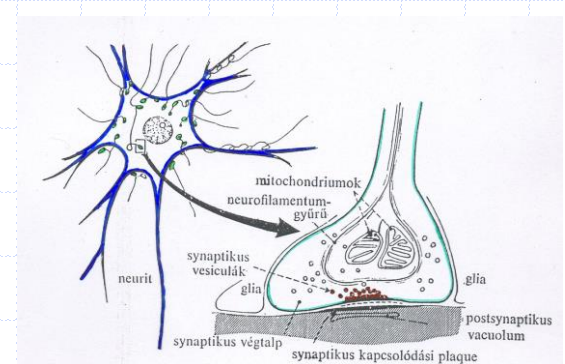
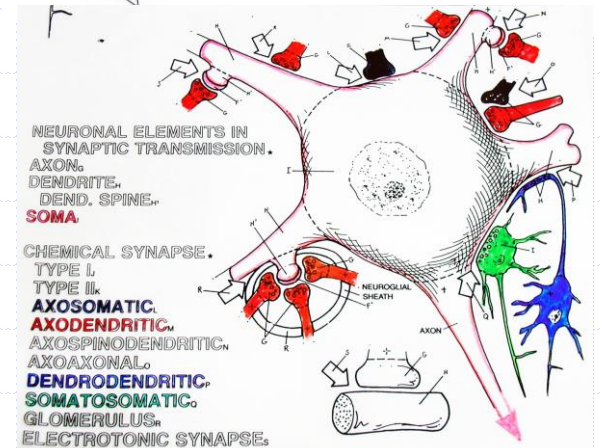
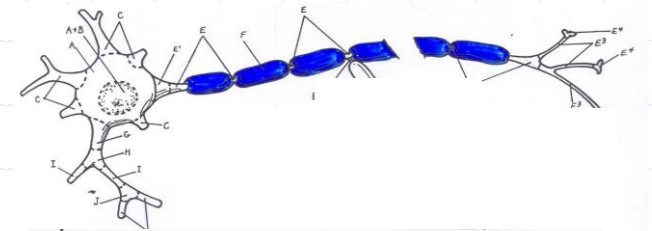
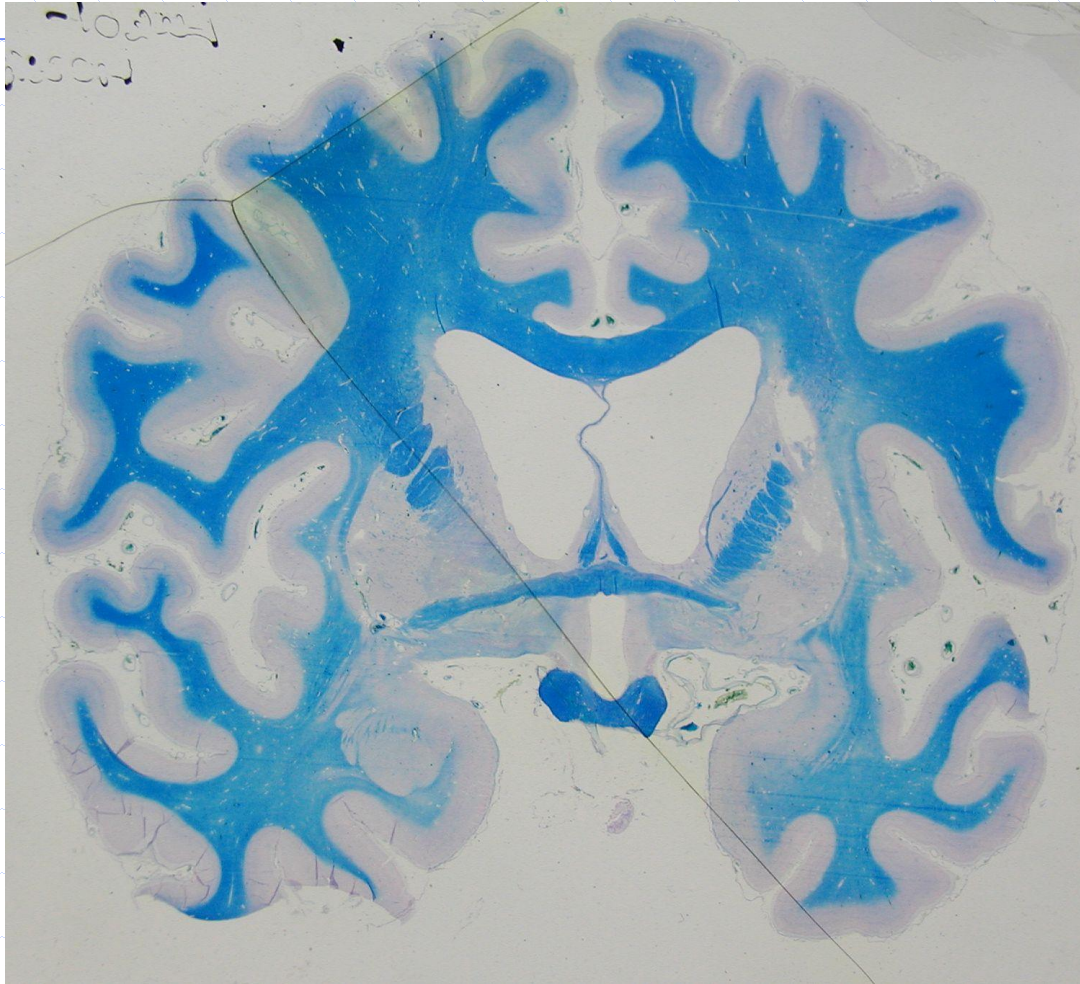
Szöveg kimondása

Szöveg értelmezése

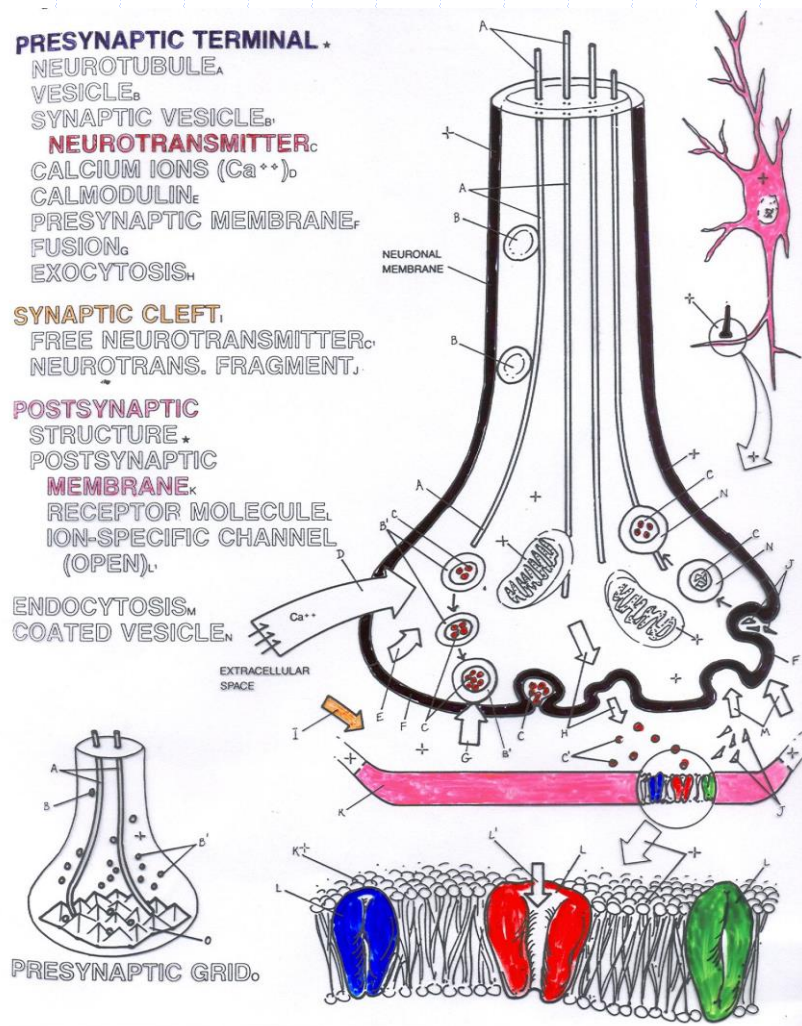
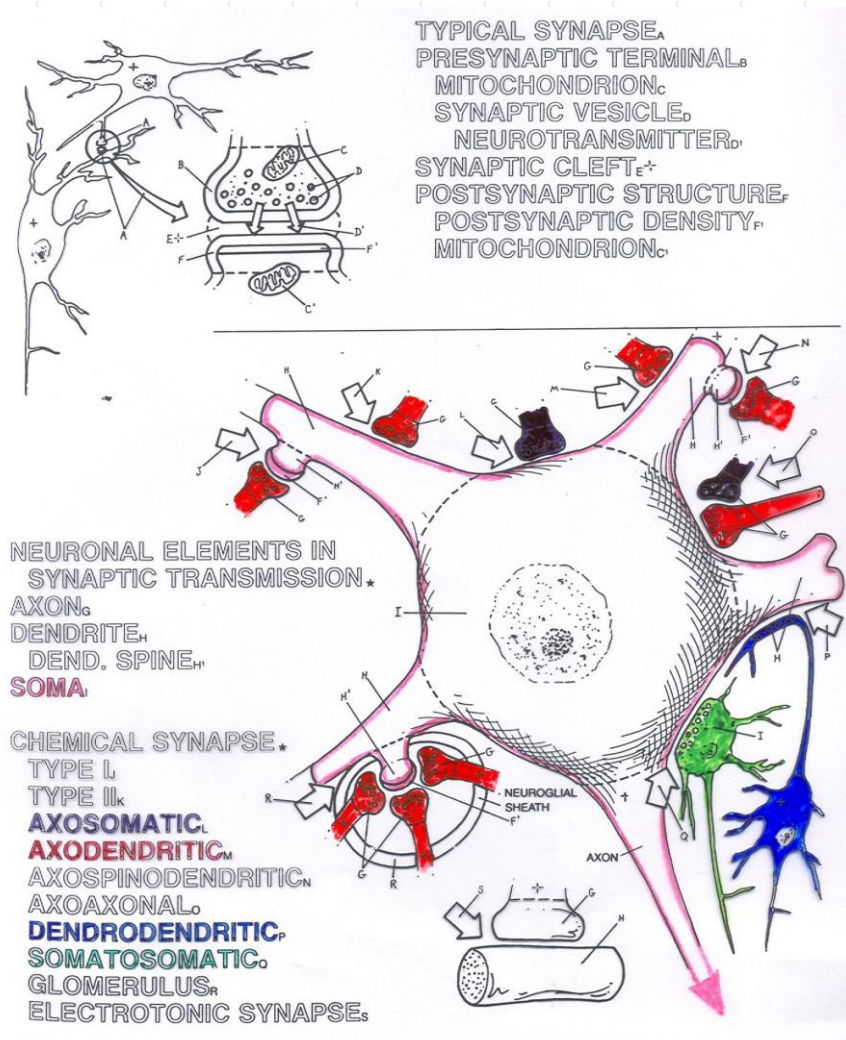


Mark Dubin, U. of Colorado
Principles of Neural Science
E. Kandel, J. Schwartz, T. Jessel

Természetes idegi hálózatok kapcsolódása



Az ingerület keletkezése



Neurális hálózatok

- ◆ Az információ feldolgozás új (?) paradigmája
- ◆ A biológiai inspirációjú információ feldolgozás, a „soft computing” egyik területe, ahol modellként az idegrendszer struktúráját és működését vesszük alapul.
- ◆ A tudományterület a kezdeti stádiumban van, mégis számos alkalmazási területen az egyszerűsített modellekkel is jobb eredmények érhetőek el mint a „hagyományos” algoritmikus megoldásokkal.

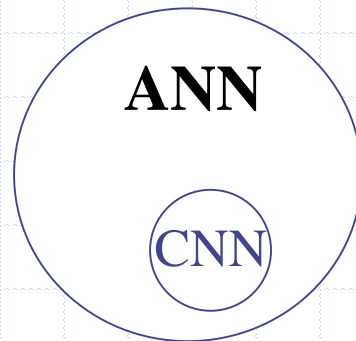
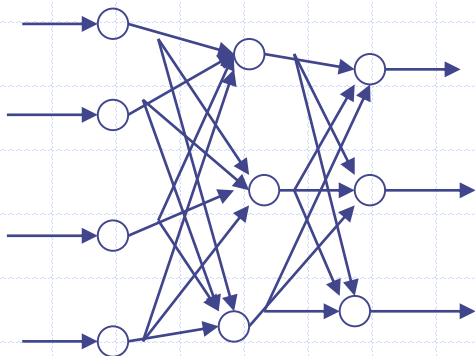
A neurális hálózatok általános jellemzői

1. A neurális hálózatok nagyon egyszerű processzorokból, az un. neuronokból épülnek fel. A processzorok változtatható súlytényezőjű összeköttetések hálózatán át kommunikálnak egymással.
2. A neurális hálózatokat nem programozzuk, hanem tanítjuk.
3. A tárolt információk a hálózatban elosztottan, a súlytényezők közvetítésével ábrázolódnak.
4. A neurális hálózatok hibatűrők. Az elosztott párhuzamos tudásreprezentáció miatt a súlytényezők egy részének jelentős megváltozása sem befolyásolja alapvetően a hálózat működését.

Neurális hálózatok alapfogalmai

Gyakori elnevezések:

Neural Networks	Neurális hálózatok	NN
Artificial Neural Networks	Mesterséges neurális hálózatok	ANN
Artificial Neural Systems	Mesterséges neurális rendszerek	ANS
Connectionist Models	Konnekcionista modellek	
Parallel Distributed Processing	Párhuzamos elosztott feldolgozás	PDP
Neural Computers	Neuronszámítógépek	

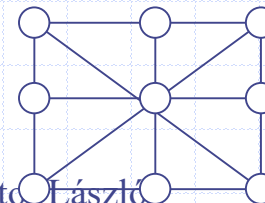


Cellular Neural Network

L.O. Chua, L. Yang, T. Roska 1988

Lokális kapcsolatok

Analóg áramkörök



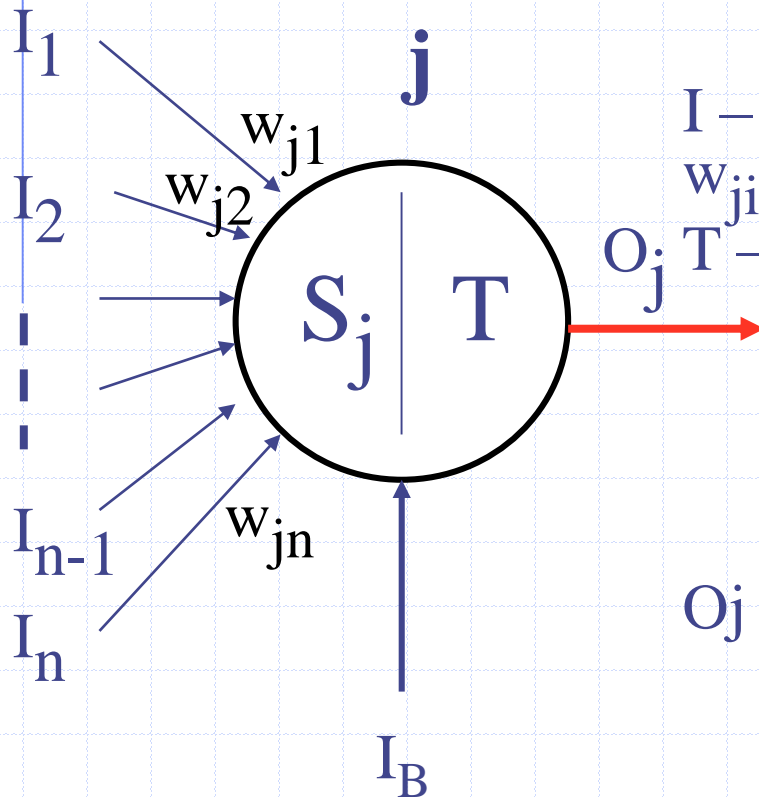
Alapfeltevések a formális neuron megfogalmazásakor

- ◆ Az idegsejt működése „minden vagy semmi jellegű”
- ◆ Az idegsejt ingerületbe hozásához bizonyos időn belül néhány (**legalább 2(!?)**) bemenetet ingerelni kell
- ◆ Az idegrendszerben az egyetlen jelentős késleltetés a szinapszisoknál jön létre
- ◆ Bármely gátló szinapszis működése teljesen megakadályozza az idegsejt ingerületbe kerülését **!?**
- ◆ Az idegrendszer összeköttetési hálózata az időben nem változik

A McCulloch és Pitts formális neuron

W. Mc Culloch és W. Pitts (1943)

„Először tekintették az agyat számításokat végző szervnek”



I – ingerfelvevők (bemenet)

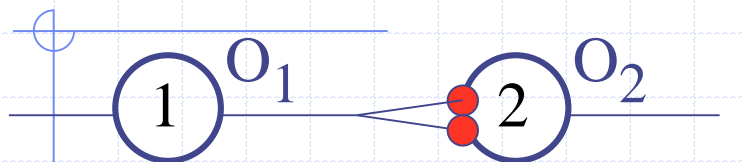
w_{ji} súlytényezők

O_j T – Árviteli (Transzfer) függvény

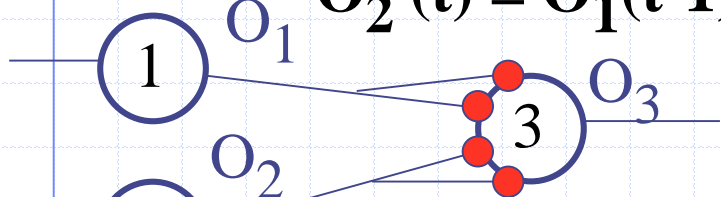
$$S_j = \sum_{i=1}^n w_{ji} I_i$$

$$O_j = 0 \text{ ha } S_j \leq 0 \quad O_j = +1 \text{ ha } S_j > 0$$

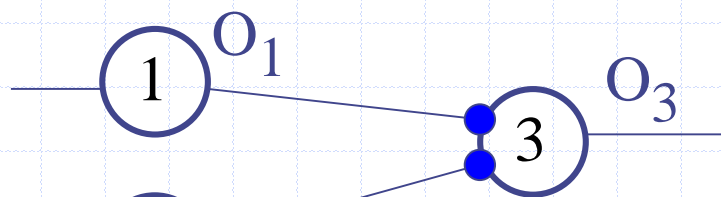
Logikai műveletek McCulloch-Pitts neuronokkal



$$O_2(t) = O_1(t-1)$$



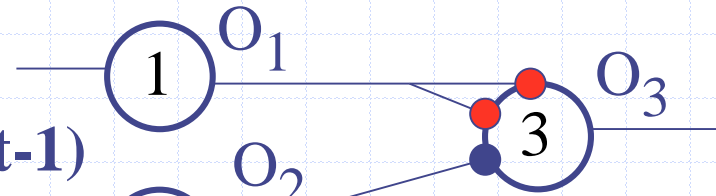
$$O_3(t) = O_1(t-1) + O_2(t-1)$$



$$O_3(t) = \overline{O_1(t-1)} \cdot \overline{O_2(t-1)}$$



$$O_3(t) = O_1(t-1) \cdot O_2(t-1)$$



$$O_3(t) = O_1(t-1) \cdot \overline{O_2(t-1)}$$

● serkentés ● gátlás

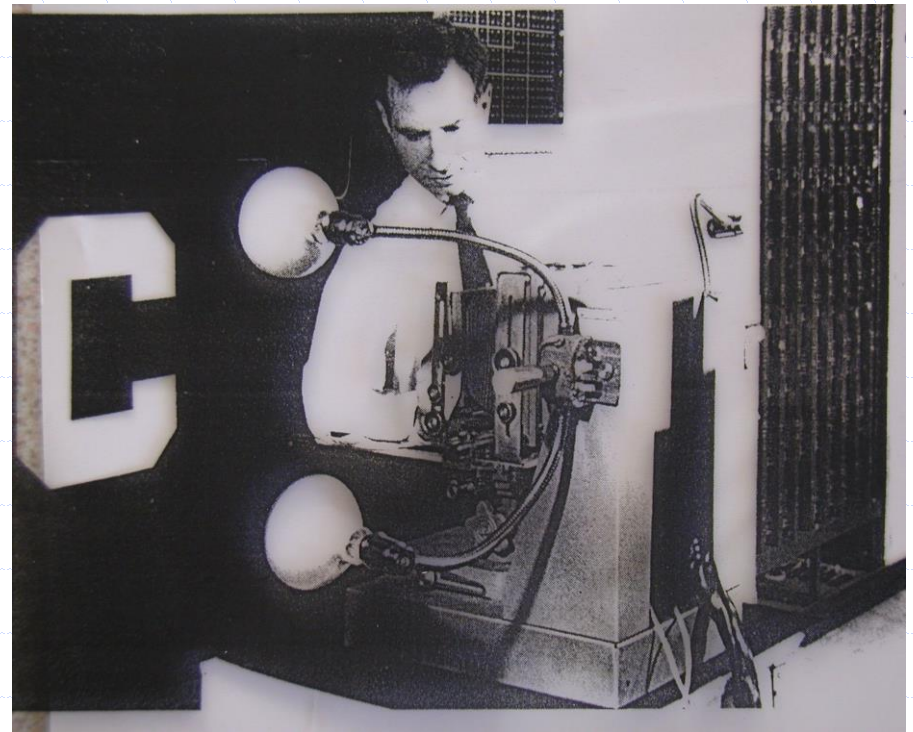
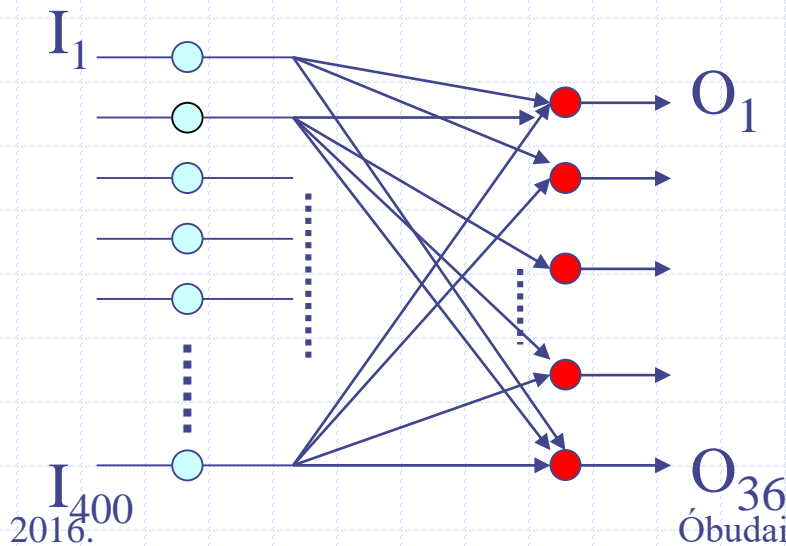
De Morgan !!!

Az első mesterséges neurális hálózat: Perceptron

Frank Rosenblatt (1957)

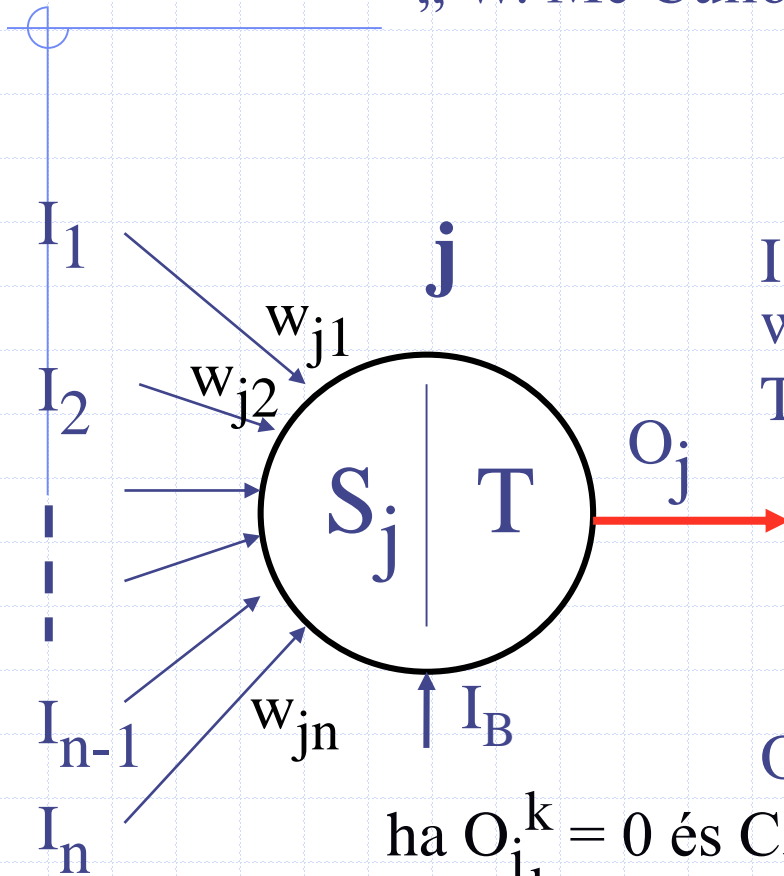
Vetített nyomtatott betűk felismerése tanítás alapján

- 20 x 20 fotóérzékelő
- Mc. Culloch-Pitts neuronok
- Előrecsatolt egyrétegű hálózat



A perceptron processzora

„ W. Mc Culloch és W. Pitts „)



I – ingerfelvők (bemenet)

w_{ji} súlytényezők

T – Árviteli (Transzfer) függvény

$$S_j = \sum_{i=1}^n w_{ji} I_i$$

$O_j = 0$ ha $S_j \leq 0$ $O_j = +1$ ha $S_j > 0$

ha $O_j^k = 0$ és $C_j^k = 1$ akkor $w_{ji}(t+1) = w_{ji}(t) + I_i^k(t)$
ha $O_j^k = 1$ és $C_j^k = 0$ akkor $w_{ji}(t+1) = w_{ji}(t) - I_i^k(t)$

A felügyeletes tanítás lényege, algorithmusa

Mottó:

Addig változtatjuk a súlytényezőket, amíg a bemenő mintákra a hálózat a megfelelő-, előre kiszámított válaszokat nem adja.

Algorithmusa:

1. Kezdeti súlytényezők beállítása
2. A tanítóminta bemeneti értéke alapján a hálózat kimeneti értékének kiszámítása.
3. A tanítóminta célértékének összehasonlítása a hálózat célértékével.
4. Szükség esetén a hálózat súlytényezőinek módosítása.
5. A tanítás folytatása mindaddig, amíg a hálózat az összes tanítómintára – egy előre rögzített hibahatárnál kisebb hibával a célértéknek megfelelő kimeneti értéket nem tudja előállítani.

A perceptron tanító algoritmus

Kezdeti súlytényezők beállítása (random !?)

Tanítás **iter** amíg a hiba el nem éri a hibahatárt ($H_i \leq H_h$)

$k = 0, H_i = 0$

Minták **iter** amíg $k = p$ nem teljesül (ahol p = tanító minták száma)

A k -ik tanítóminta bemeneti vektora alapján processzoronként az aktiváció kiszámítása $S_{jk} = \sum I_j^k * W_{ji}$

A közbő átviteli függvény alapján processzoronként a kimeneti értékek kiszámítása. (O_j)

A hálózat súlytényezőinek módosítása.

ha $O_j^k = C_j^k$ akkor $w_{ji}(t+1) = w_{ji}(t)$

ha $O_j^k = 0$ és $C_j^k = 1$ akkor $w_{ji}(t+1) = w_{ji}(t) + I_i^k(t)$

ha $O_j^k = 1$ és $C_j^k = 0$ akkor $w_{ji}(t+1) = w_{ji}(t) - I_i^k(t)$

A hálózat hibájának kiszámítása $H_j^k = C_j^k - O_j^k$

A hibák összesítése $H_i = H_i + H_j$

$k := k + 1$

Minták **end**

Tanítás **end**

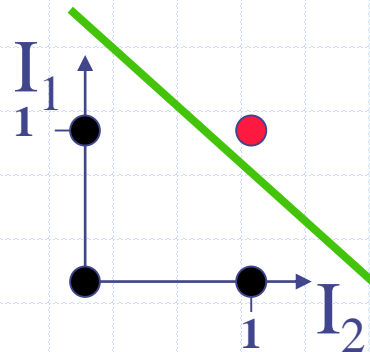
A perceptron jellemzői

(„Perceptrons” M. Minsky – S. Pappert, 1969)

1. Csak egy réteg tanítását teszi lehetővé
2. Csak lineárisan elválasztható csoportokat tud osztályozni.

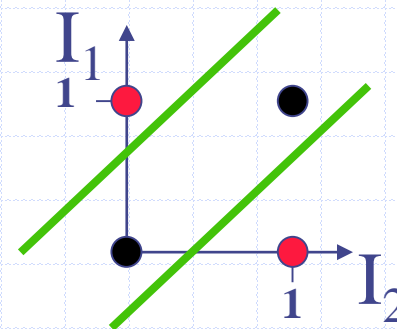
$$I_1 * I_2 \quad O$$

0	0	0
0	1	0
1	0	0
1	1	1



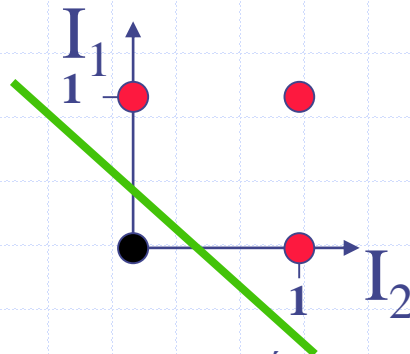
$$I_1 \oplus I_2 \quad O$$

0	0	0
0	1	1
1	0	1
1	1	0



$$I_1 + I_2 \quad O$$

0	0	0
0	1	1
1	0	1
1	1	1



A perceptron értékelése



I do not come to the discussion of connectionism as a neutral observer. In fact, the standard version of its history assigns me a role in a romantic story whose fairytale resonances surely contribute at least a little to connectionism's aura of excitement.

Once upon a time two daughter sciences were born to the new science of cybernetics. One sister was natural, with features inherited from the study of the brain, from the way nature does things. The other was artificial, related from the beginning to the use of computers. Each of the sister sciences tried to build models of intelligence, but from very different materials. The natural sister built models (called neural networks) out of mathematically purified neurones. The artificial sister built her models out of computer programs.

In their first bloom of youth the two were equally successful and equally pursued by suitors from other fields of knowledge. They got on very well together. Their relationship changed in the early sixties when a new monarch appeared, one with the largest coffers ever seen in the kingdom of the sciences: Lord DARPA, the Defense Department's Advanced Research Projects Agency. The artificial sister grew jealous and was determined to keep for herself the access to Lord DARPA's research funds. The natural sister would have to be slain.

The bloody work was attempted by two staunch followers of the artificial sister, Marvin Minsky and Seymour Papert, cast in the role of the huntsman sent to slay Snow White and bring back her heart as proof of the deed. Their weapon was not the dagger but the mightier pen, from which came a book—*Perceptrons*—purporting to prove that neural nets could never fill their promise of building models of mind: *only computer programs could do this*. Victory seemed assured for the artificial sister. And indeed, for the next decade all the rewards of the kingdom came to her progeny, of which the family of expert systems did best in fame and fortune.

But Snow White was not dead. What Minsky and Papert had shown the world as proof was not the heart of the princess; it was the heart of a pig.

Seymour Papert, 1988

A neurális hálózatok legfontosabb meghatározó tényezői

1. A neuronok (processzorok)

(neuron, artificial neuron, node, unit, cell)

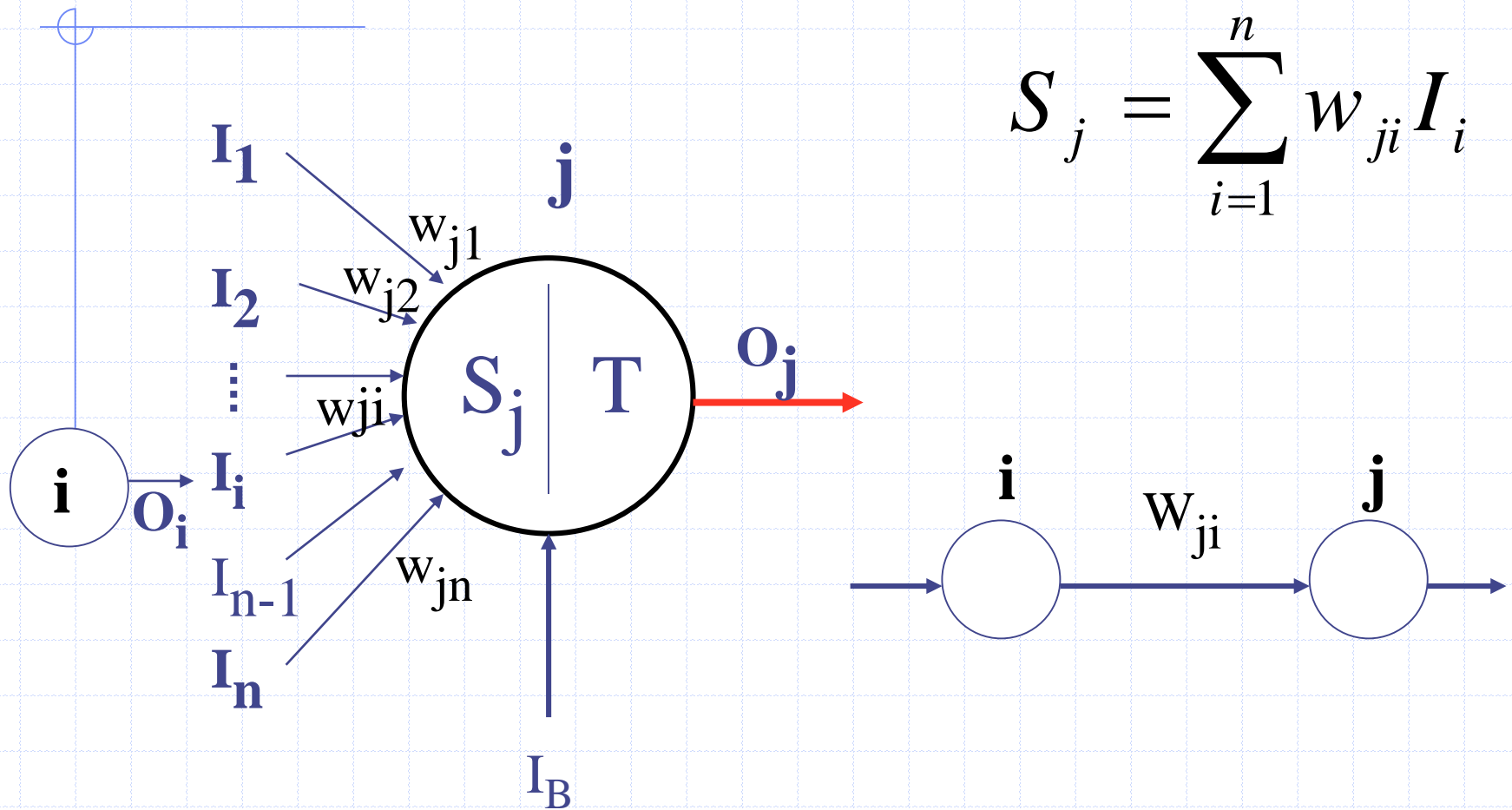
2. A hálózat összeköttetési sémája (topológiája)

„mit mivel kötünk össze”, (súlytényező mátrix)

3. A tanító szabályokat alkalmazó algoritmus

(a súlytényezők beállítása, „hangolása”)

Az alap „neuron” (processzor) felépítése

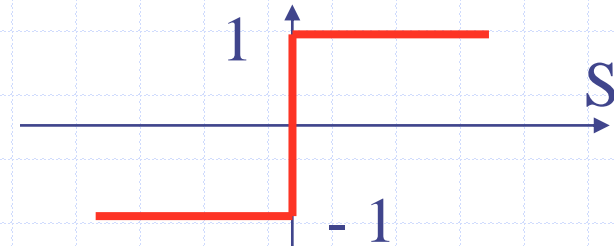
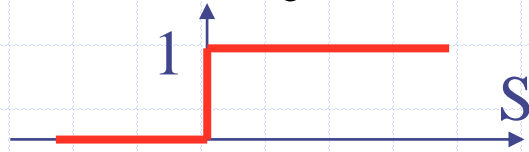


$$S_j = \sum_{i=1}^n w_{ji} I_i$$

I – bemenet), w_{ji} súlytényezők, T – Átviteli (Transzfer) függvény

Leggyakrabban használt átviteli függvények

1. Ugrás függvény: $O_j = 0$ vagy -1 , ha $S \leq 0$, $O_j = 1$ ha $S > 0$

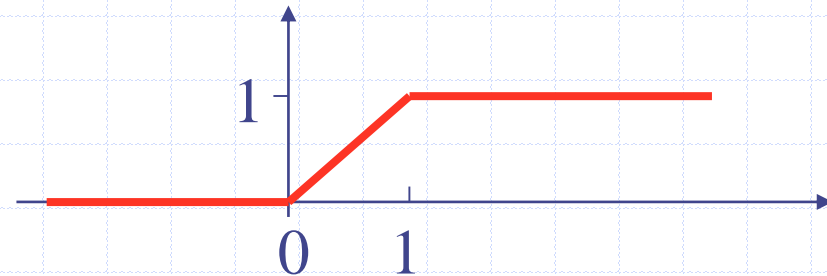


2. Korlátozott lineáris függvény

$O_j = 0$, ha $S \leq 0$,

$O_j = S$ ha $0 \leq S < 1$

$O_j = 1$ ha $S > 1$

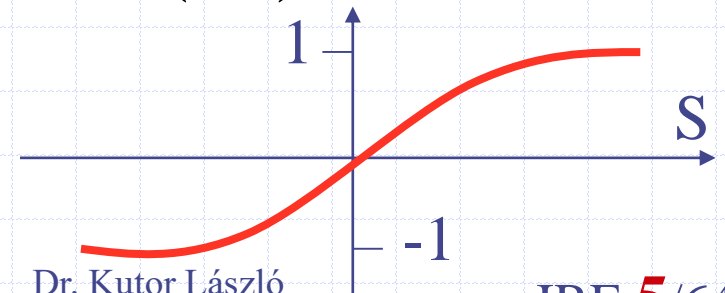
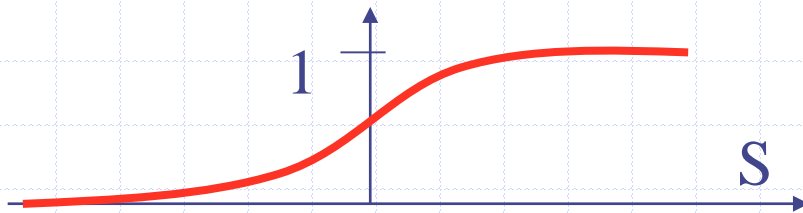


3. Szimoid függvény

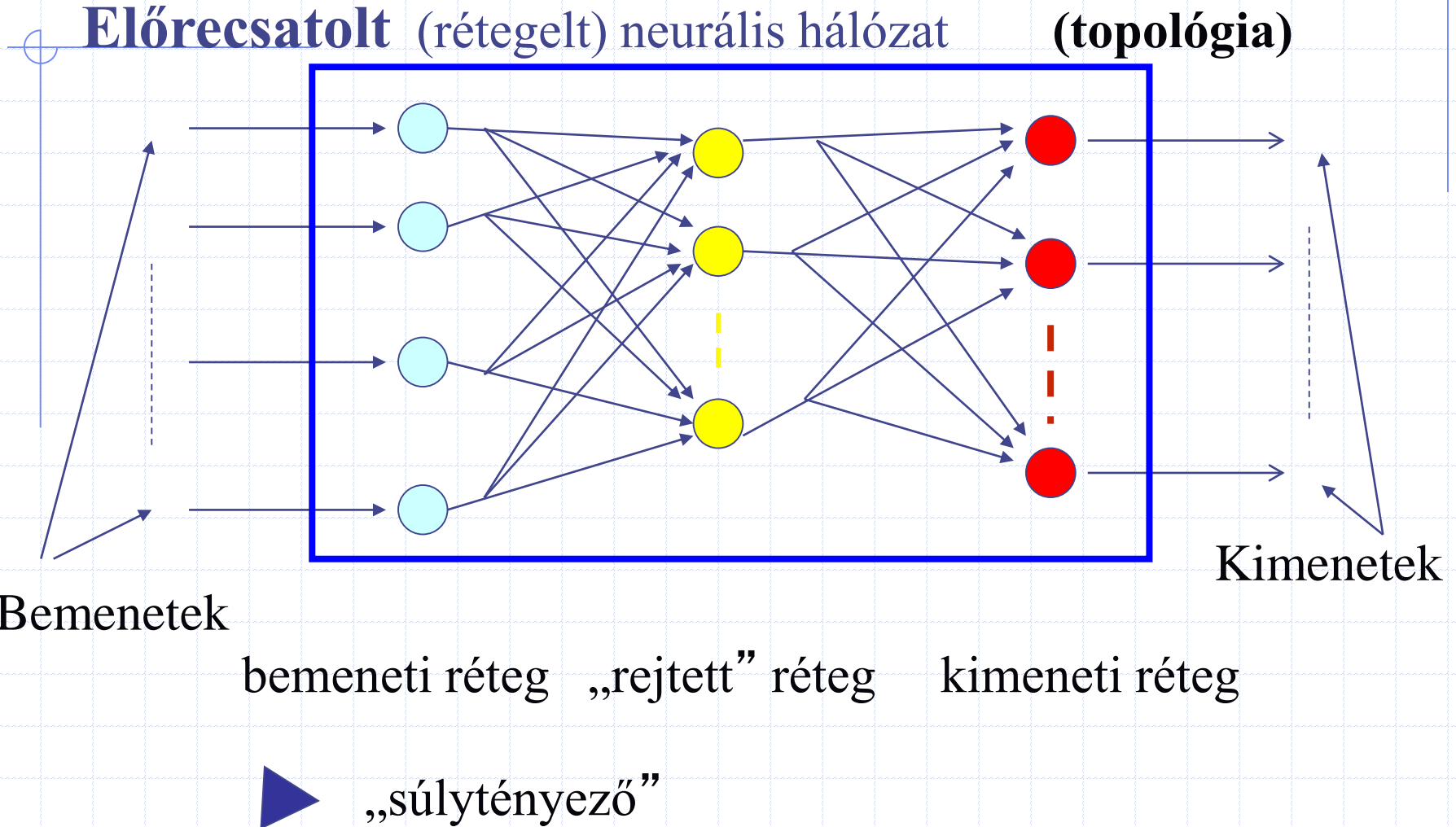
$O_j = 1/(1+e^{-Sj})$

$O_j = 1 - 1/(1+S)$ ha $S \geq 0$

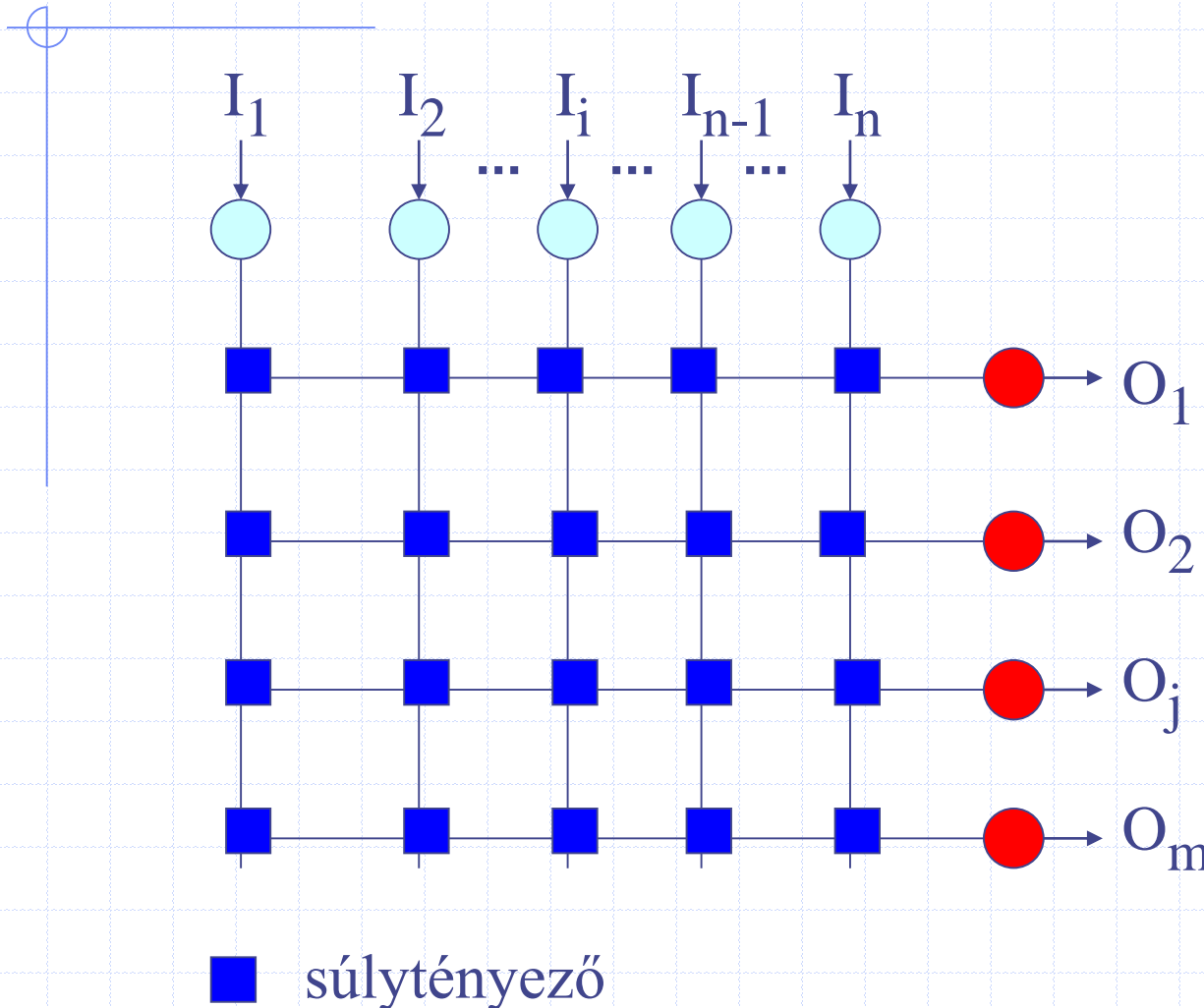
$O_j = -1 + 1/(1-S)$ ha $S < 0$



Tipikus neurális hálózat összeköttetések 1.



Az előrecsatolt hálózatok alternatív ábrázolása



Súlymátrix

$$\begin{matrix}
 w_{11} & w_{12} & w_{1i} & w_{1n} \\
 w_{21} & w_{22} & w_{2i} & w_{2n} \\
 w_{j1} & w_{j2} & w_{ji} & w_{jn} \\
 w_{m1} & w_{m2} & w_{mi} & w_{mn}
 \end{matrix}$$

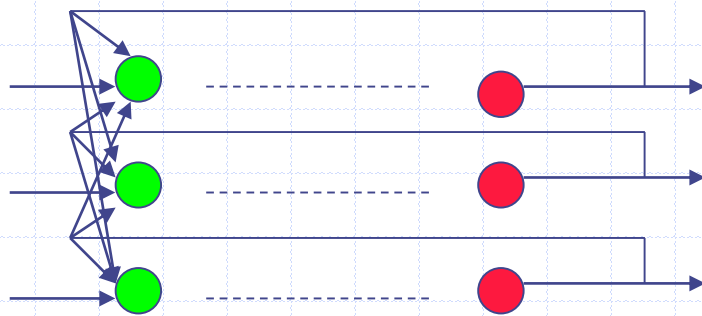
$$O = f(S)$$

$$S = I * W$$

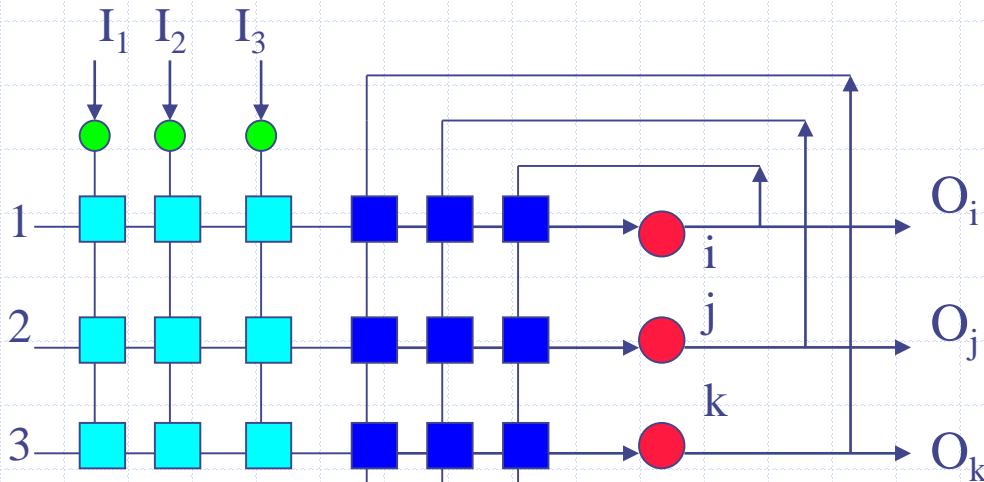
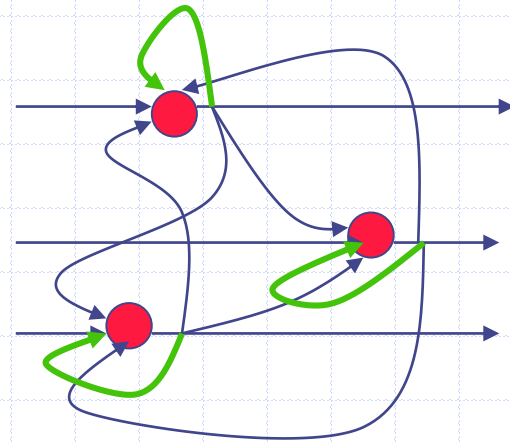
Mátrix műveletek !

Visszacsatolt neurális hálózat

Rétegzelt

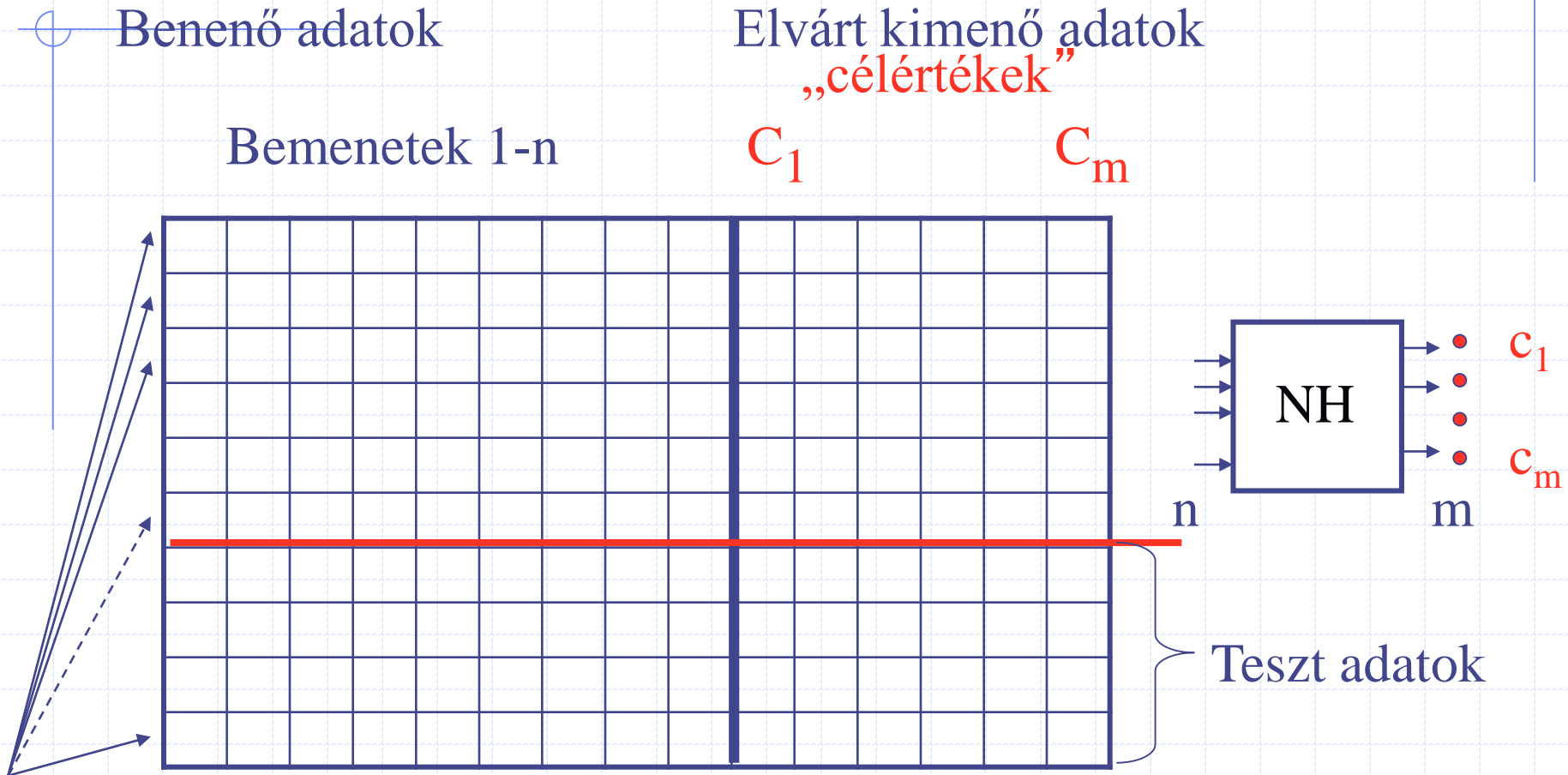


Teljesen összekötött



$$\mathbf{O} = \mathbf{I} \times \mathbf{W1} + \mathbf{O} \times \mathbf{W2}$$

A tanító adatok szerkezete



Tanító minták 1-k

Tanítási szabályok

Tanítás = súlytényezők (kis lépésekkel (?) való) beállítása

Tanítási típusok:

1. Felügyelt (felügyeletes) tanítás
2. Felügyelet nélküli (önszerveződő) tanítás

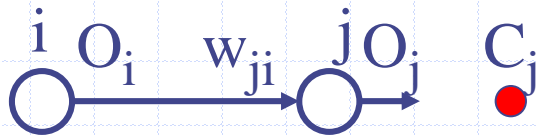
Alap tanítási szabályok:



Hebb szabály (Donald O. Hebb)

$$w_{ji}(t+1) = w_{ji}(t) + \alpha * O_i * O_j$$

ahol α = tanítási tényező, $0 \leq \alpha \leq 1$

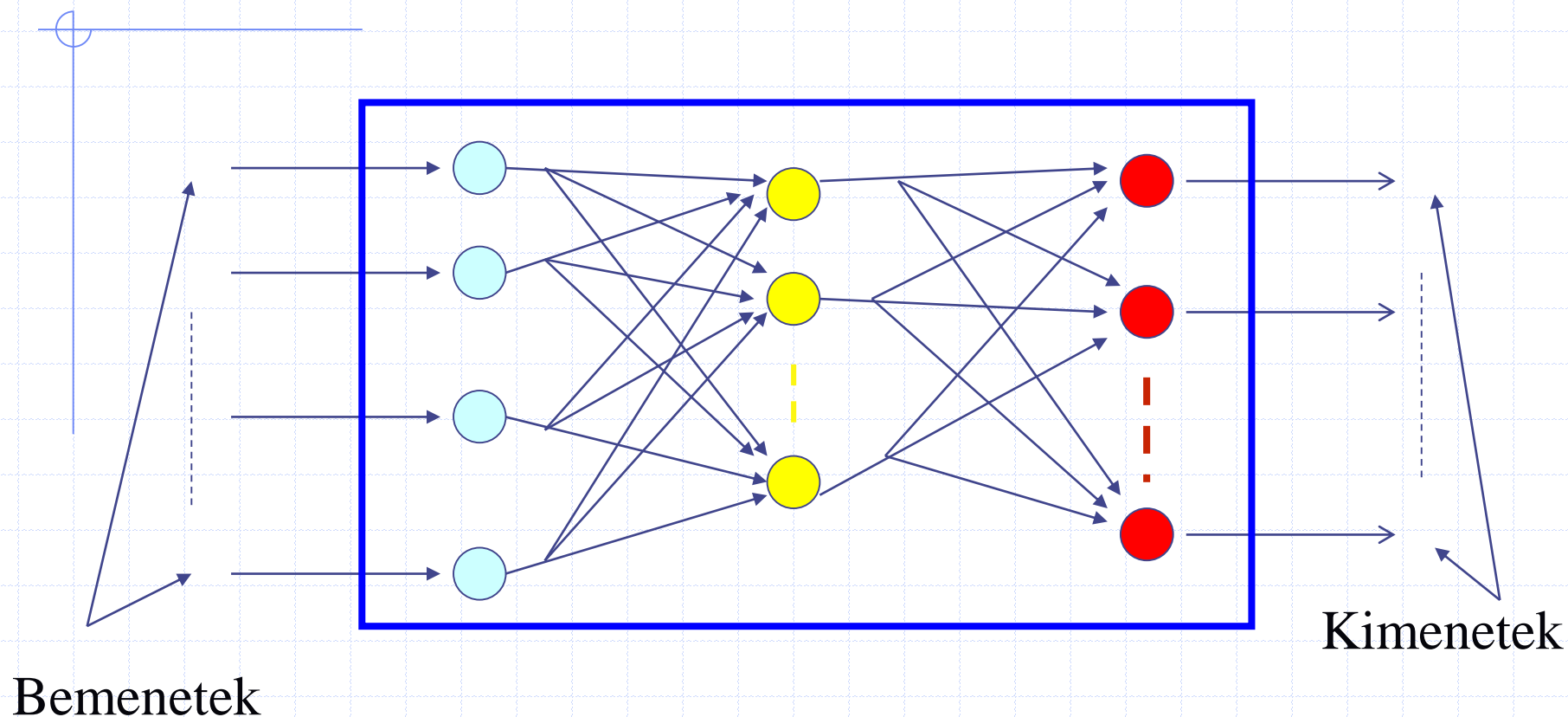


Delta szabály (Widrow- Hoff)

$$w_{ji}(t+1) = w_{ji}(t) + \alpha * O_i * (C_j - O_j)$$

ahol $C_j - O_j = \Delta_j$

Réteges szerkezetű hálózat



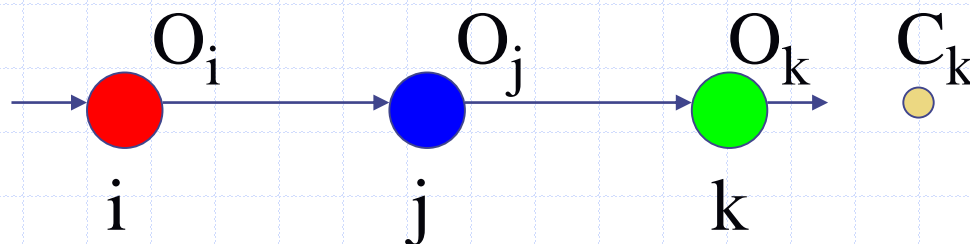
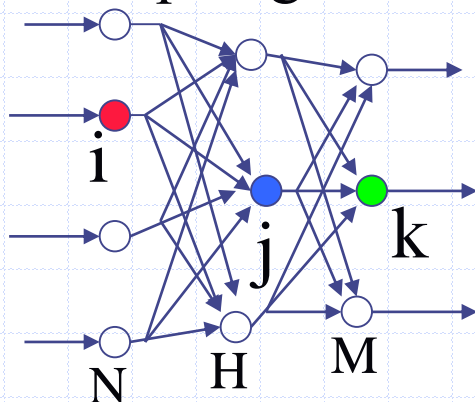
Megoldandó probléma: hogyan lehet a hálózat belsejében lévő összeköttetéseket tanítani?

Többrétegű neurális hálózatok tanítása

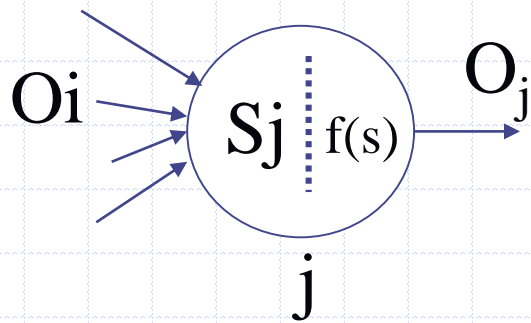
(D.E Rummelhart, G.E. Hinton, R.J. Williams, 1986)

(David Parker, Yann Le Cun) +.....

Hálózat topológia

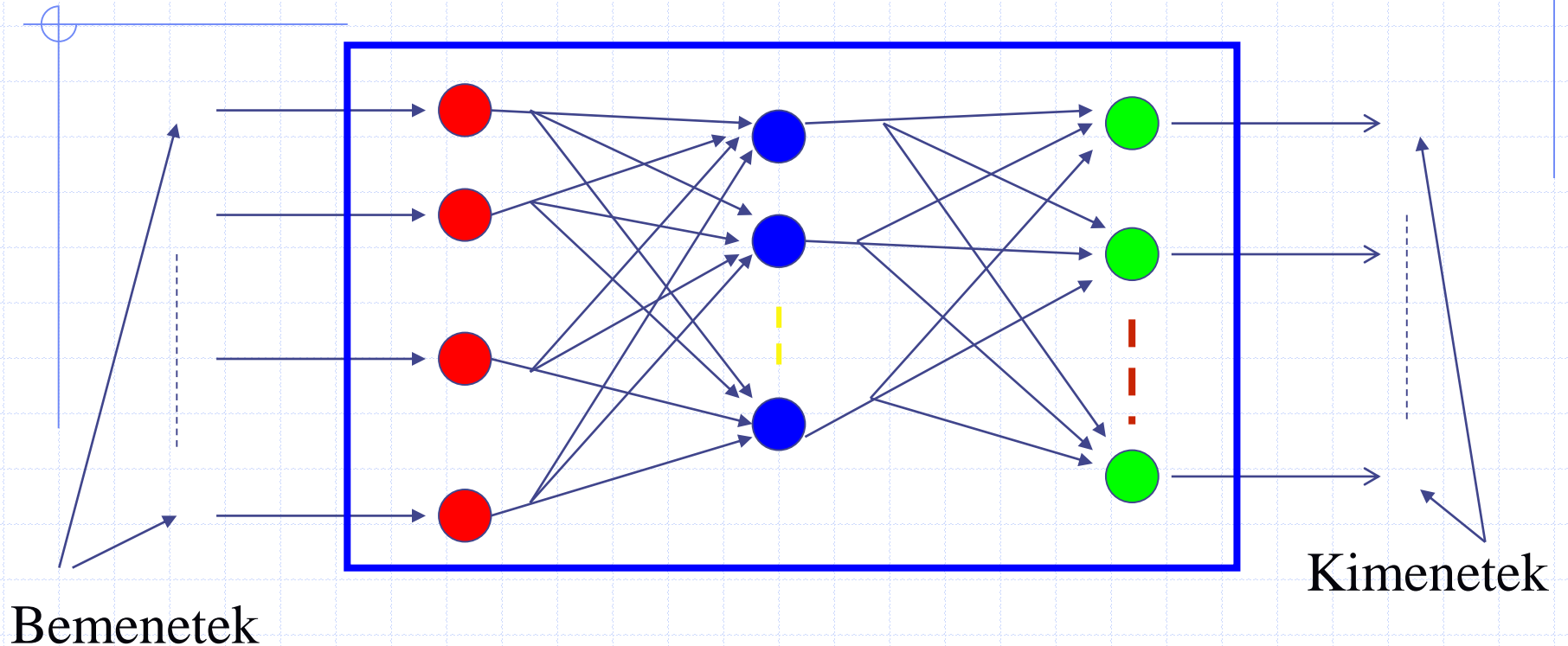


Processzor:



$$S_j = \sum_{i=1}^N w_{ji} O_i \quad O_j = f(s) = 1/(1+e^{-s})$$

Két (tanítható) rétegű előrecsatolt hálózat



bemeneti réteg „rejtett” réteg kimeneti réteg

▶ „súlytényező”

A hálózat hibája felügyelt tanítás esetén

Egy tanító mintánál: $E = \frac{1}{2} \sum_k (C_k - O_k)^2$

A teljes tanító mintára: $E = \frac{1}{2} \sum_p \sum_k (C_k - O_k)^2$

A tanítás (súlytényező változtatás) mottója:

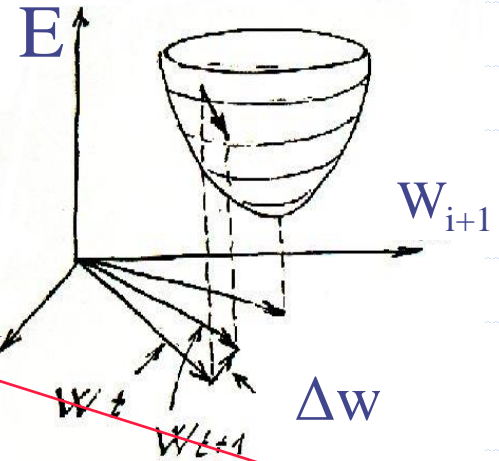
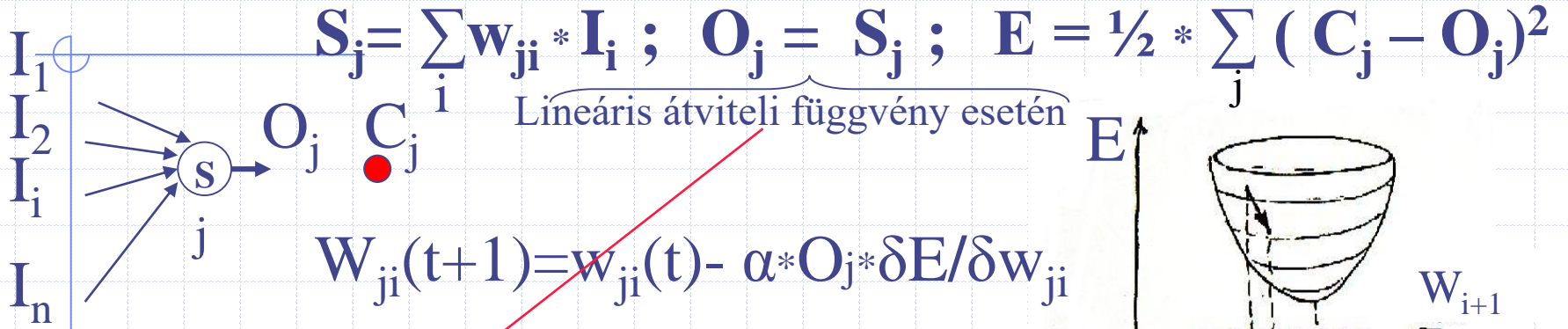
„hiba visszaterjesztés = Back error Propagation”

„A hálózat súlytényezőit a hiba létrehozásában játszott szerepükkel arányosan változtatjuk.” =

A súlytényezőket a hibafüggvény parciális deriváltja szerint változtatjuk!

(Grádiens módszer!!!)

A delta szabály matematikai értelmezése

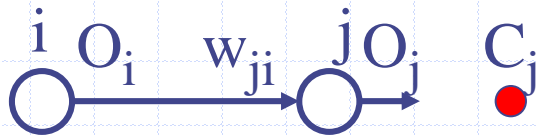


$$\delta E / \delta w_{ji} = \delta E / \delta O_j * \delta O_j / \delta S_j * \delta S_j / \delta w_j$$

$$\delta E / \delta O_j = \frac{1}{2} * 2 * (C_j - O_j) * -1 = -(C_j - O_j)$$

$$\delta O_j / \delta S_j = 1$$

$$\delta S_{ji} / \delta w_{ji} = \delta (w_{j1} * I_1 + \dots + w_{ji} * I_i + \dots + w_{jn} * I_n) = I_i$$



Delta szabály

$$w_{ji}(t+1) = w_{ji}(t) + \alpha * I_i * \Delta w_{ji}$$

ahol α = tanítási tényező, $0 \leq \alpha \leq 1$,

A tanítást leíró összefüggések többrétegű hálózatoknál (általánosított delta szabállyal)

Általánosított delta szabály: (deriválás a lánc szabály alapján)

$$\delta E / \delta w_{kj} = \delta E / \delta O_k * \delta O_k / \delta S_k * \delta S_k / \delta w_{kj}$$

Súlytényező változtatás a kimeneti rétegben

$$W_{kj}(t+1) = W_{kj}(t) + \alpha \Delta_k O_j =$$

$$W_{kj}(t) + \alpha * \underbrace{(C_k - O_k)}_{\Delta k} * f(S_k) * (1 - f(S_k)) * O_j$$

Súlytényező változtatás a „rejtett” rétegben

$$W_{ji}(t) + \alpha * \sum_{k=1}^M \underbrace{(\Delta k * W_{kj})}_{\Delta j} * f(S_j) * (1 - f(S_j)) * O_i$$

(A levezetés a javasolt olvasmányok között található)

Neurális hálózatok alkalmazásának menete

- A tanító adatok összeállítása.
- Feladatspecifikus neurális hálózat (paradigma) kiválasztása.
- A hálózat jellemzőinek (a processzorok átviteli függvényének, a processzorok számának, a tanítási módszereknek és paramétereinek, valamint a kezdeti súlymátrix értékeinek) kiválasztása.
- Teljesítmény mérő módszer kiválasztása
- Tanítás és tesztelés, amíg a hálózat a kívánt viselkedést nem mutatja.

Az egyik nagysikerű neurális hálózat alkalmazás a NetTalk

Célja: Angol betűk kiejtésének kiszámolása

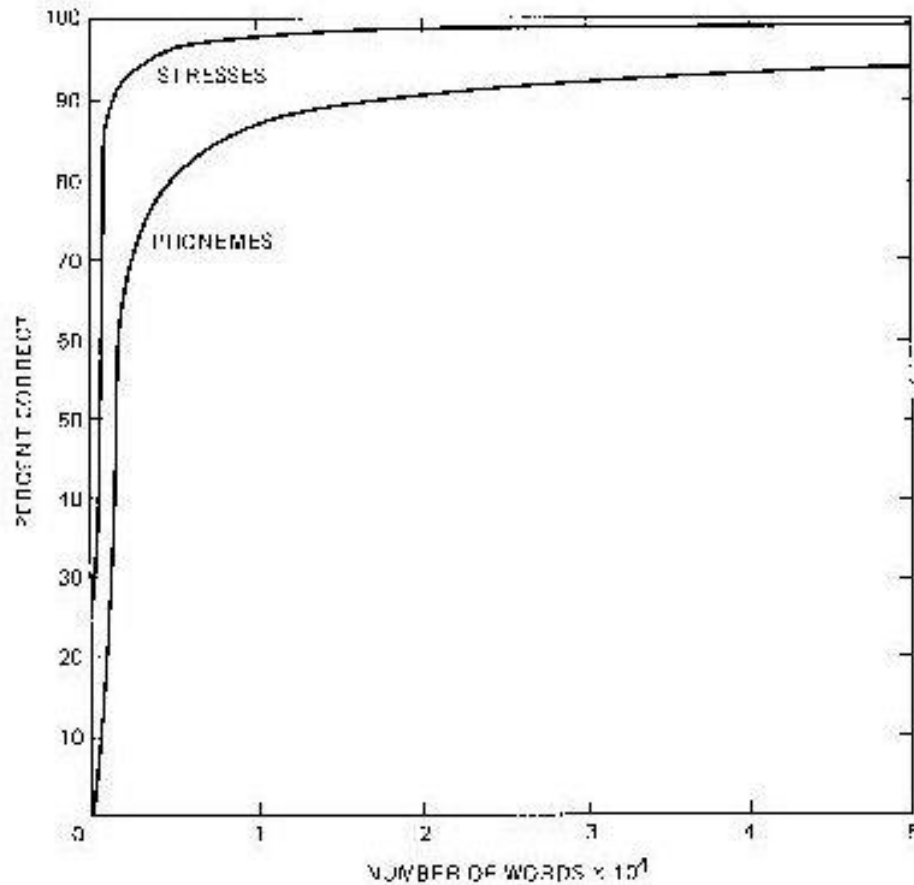
Módszere: 2 tanítható rétegű, előrecsatolt neurális hálózat

7 egymást követő betű közül a középsőt kellett kiejteni

Eredménye: Minden korábbi megoldásnál jobb eredményt mutatott

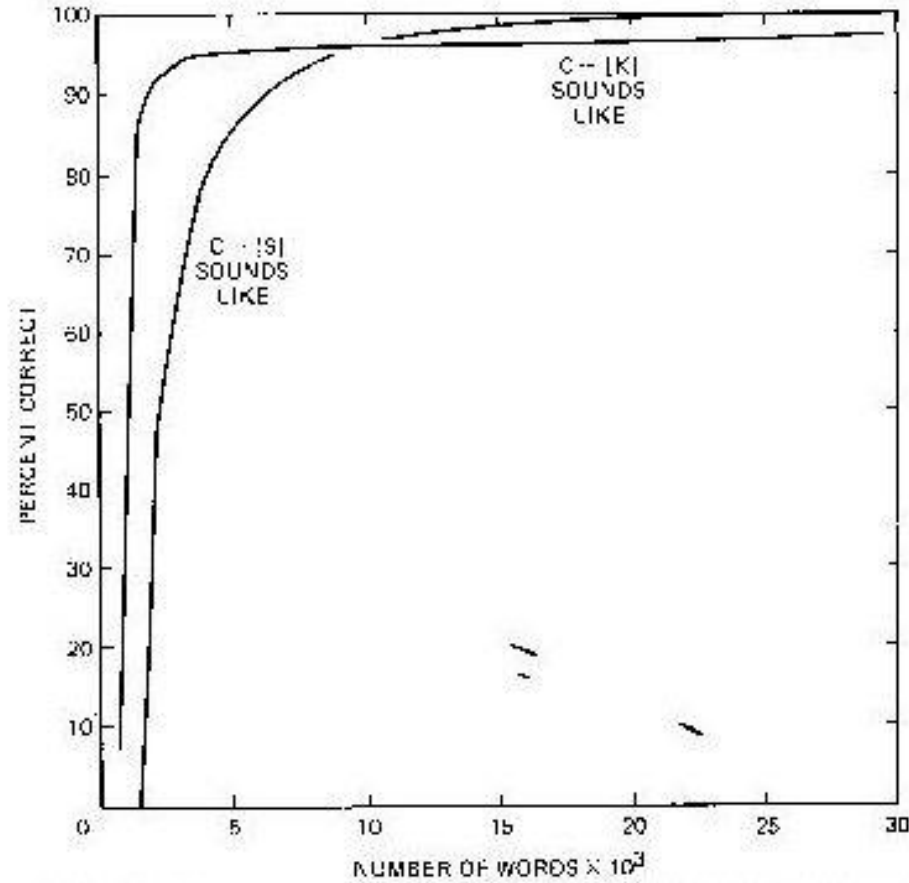
A NETTALK tapasztalatai 1.

Fonémák, hangsúlyok tanulása



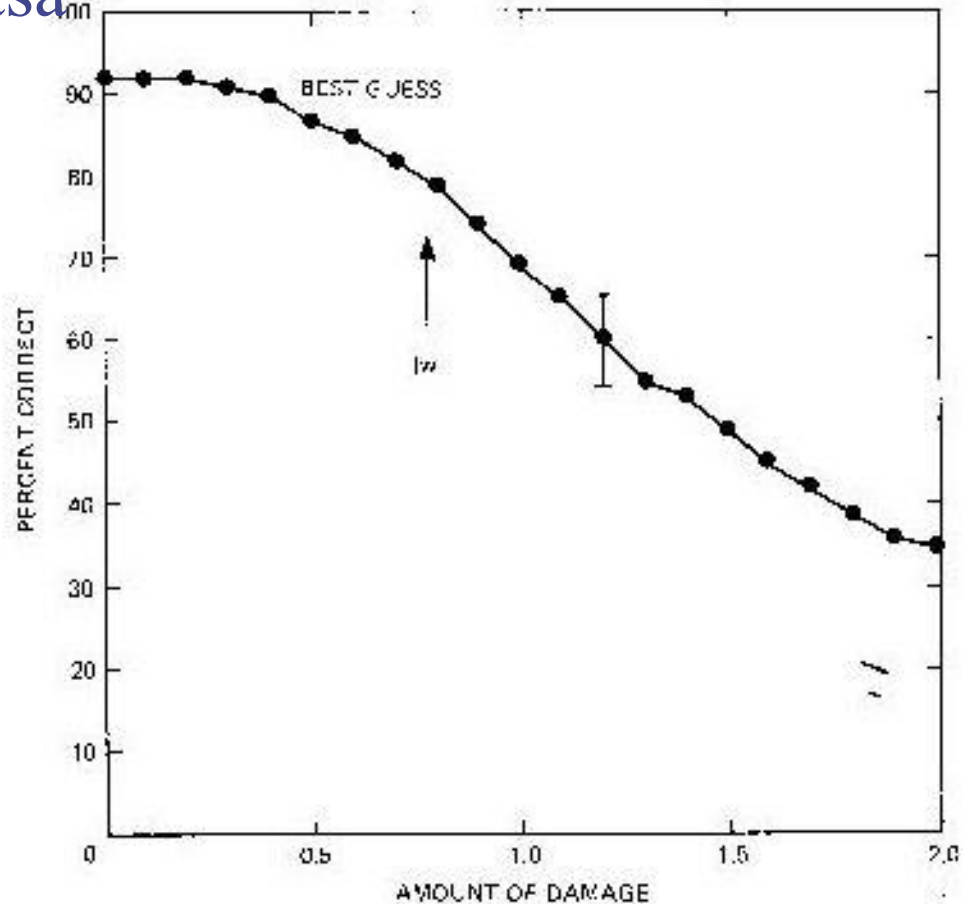
A NETTALK tapasztalatai 2.

C és K tanulása



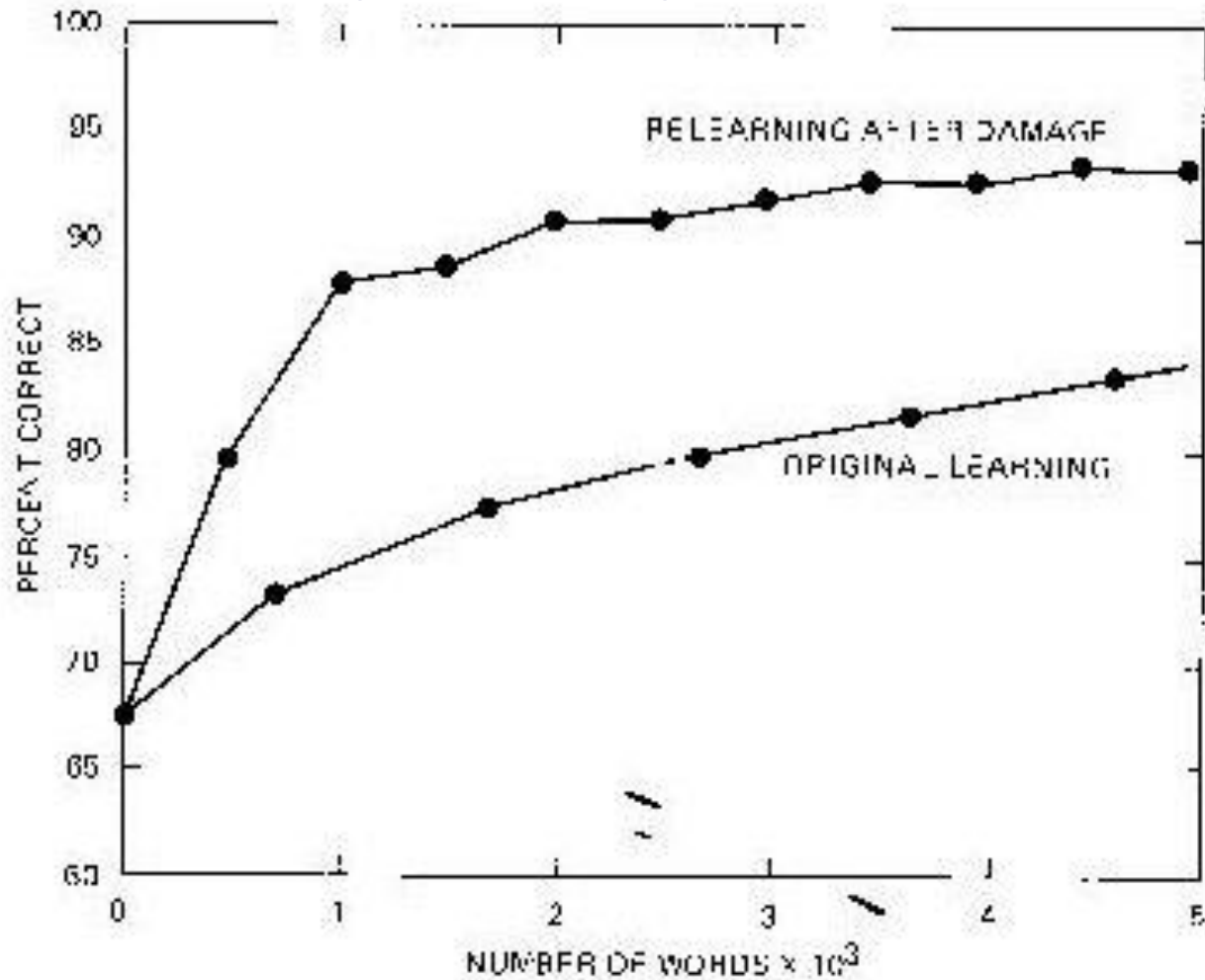
A NETTALK tapasztalatai 3.

A súlytényezők megváltoztatásának hatása



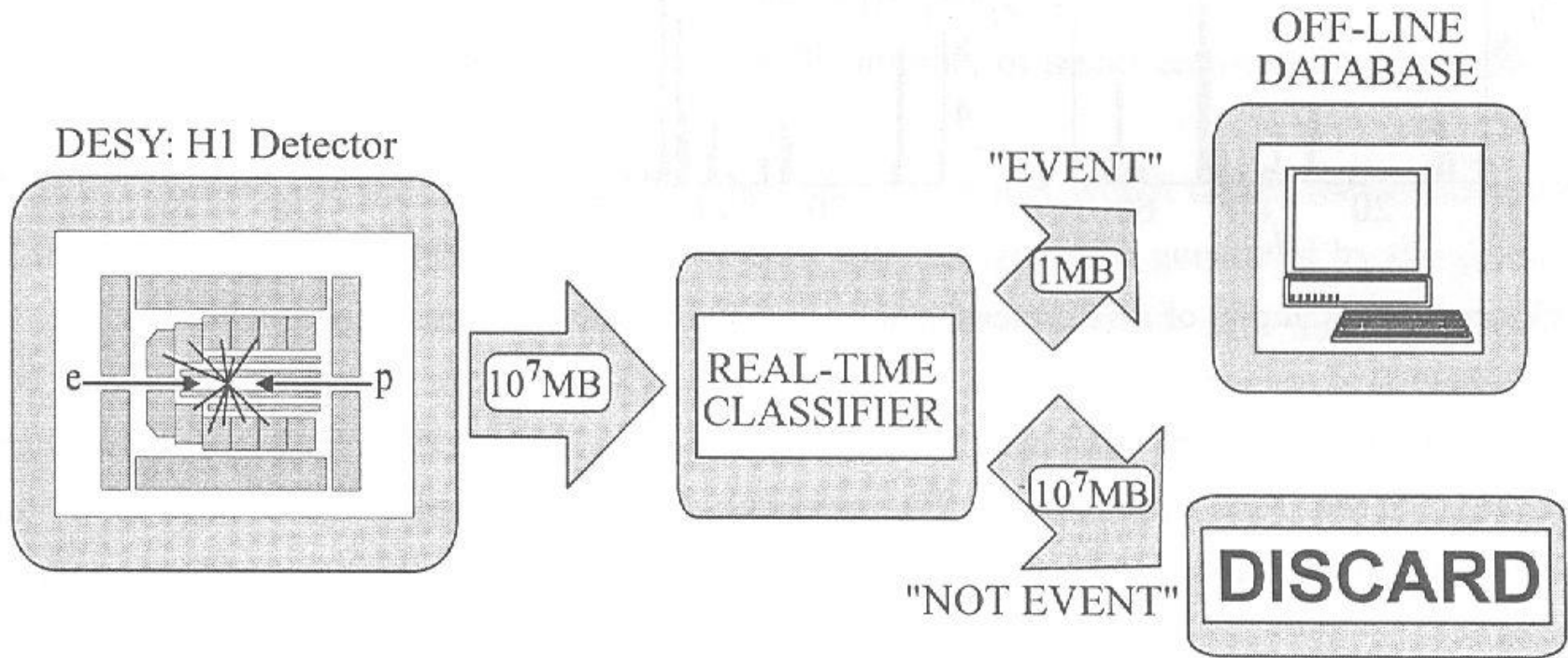
A NETTALK tapasztalatai 4.

Az újratanulás jellemzői



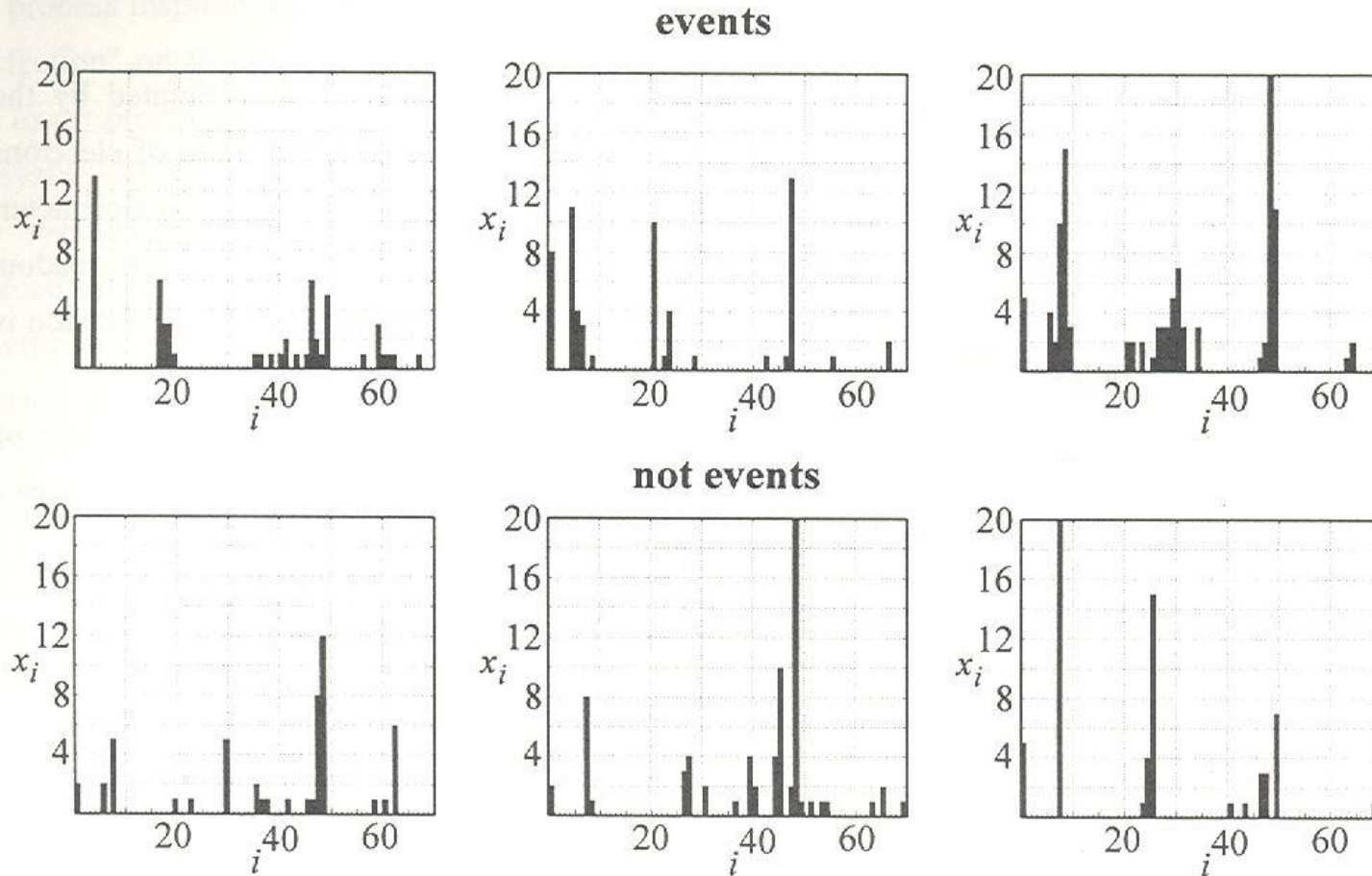
Alkalmazási példa 2/1.

Masa Péter Cenr 199x



Alkalmazási példa 2/2.

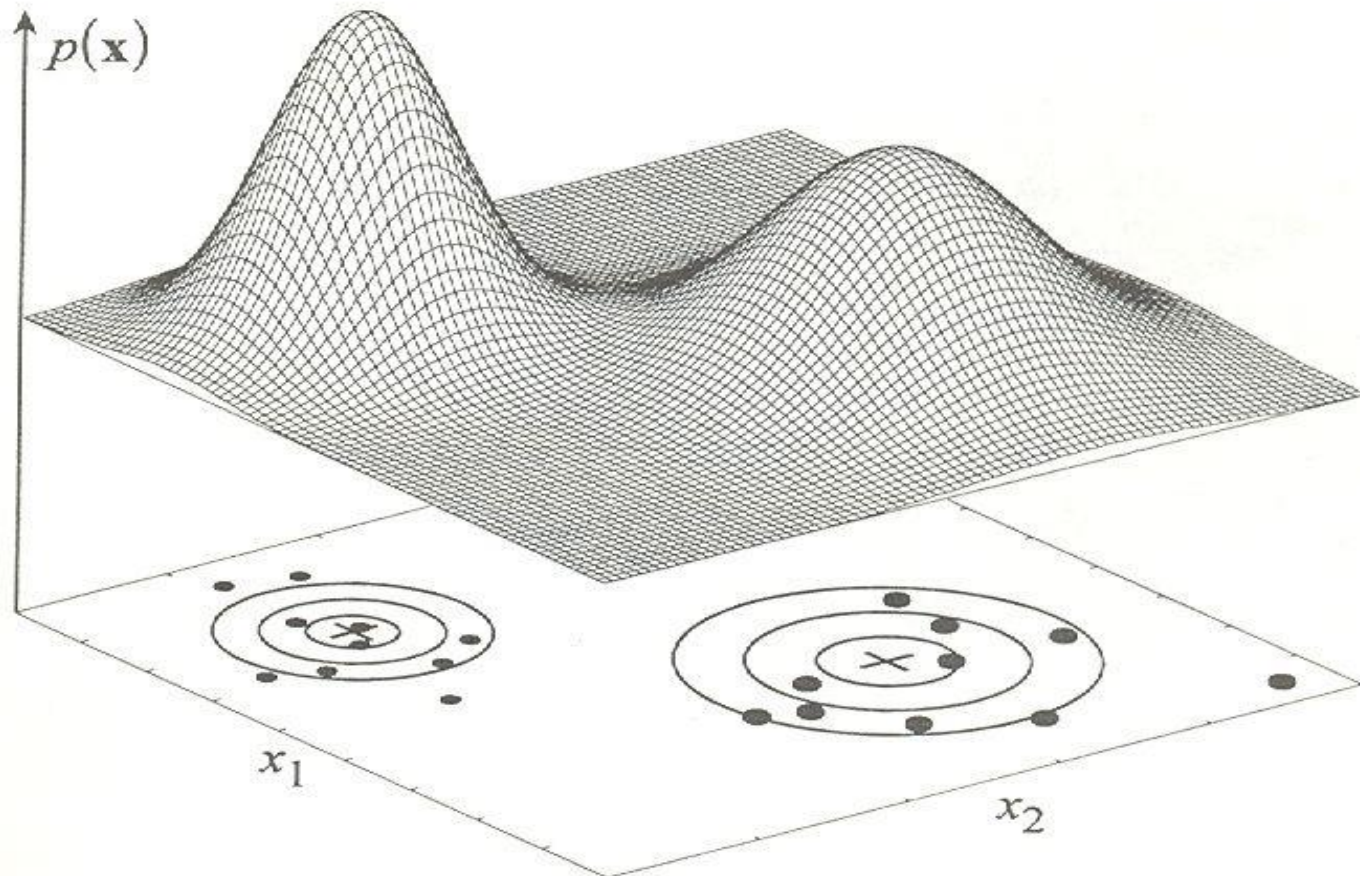
Osztályozandó minták:



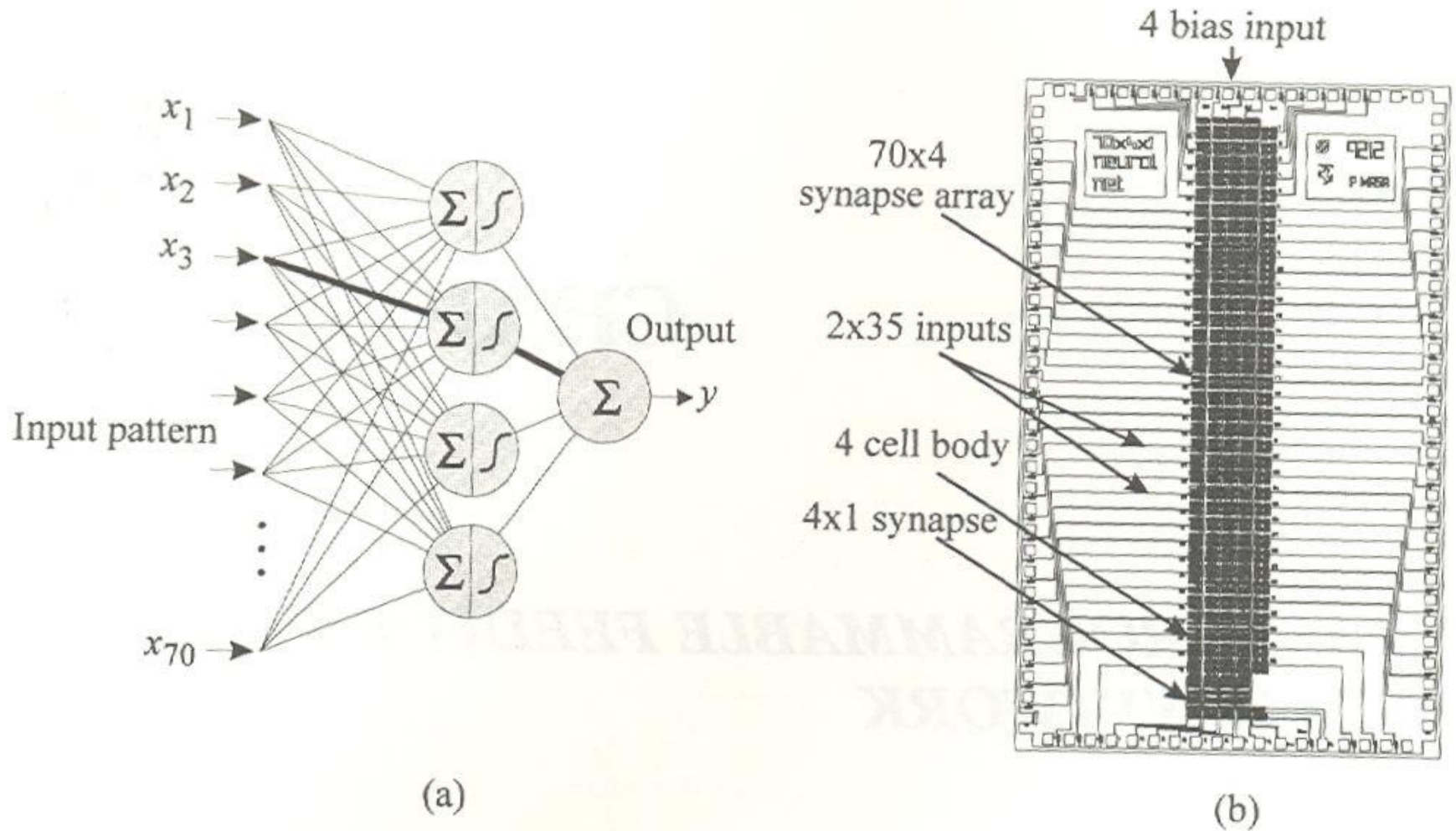
Examples of $i=70$ dimensional "event" and "not event" patterns, x

Alkalmazási példa 2/3.

Megkülönböztetendő minták 3 dimenzió esetén

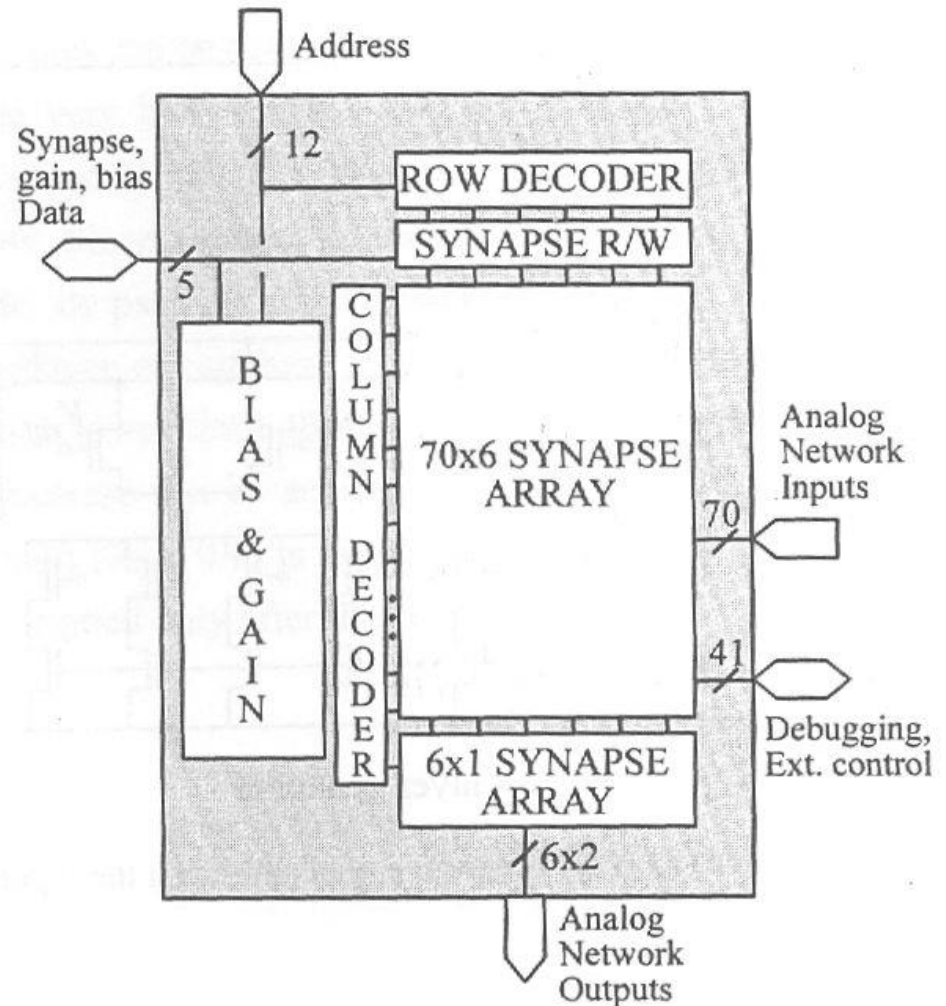
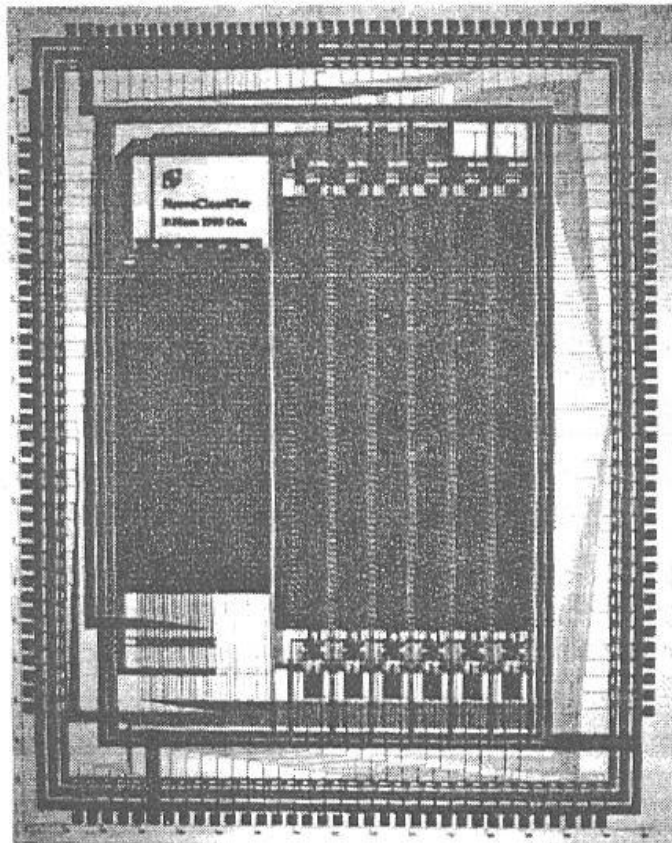


Alkalmazási példa 2/4.



Alkalmazási példa 2/5.

A megvalósított áramkör



Alkalmazási példa 2/6.

A neurális megoldás teljesítmény mutatói

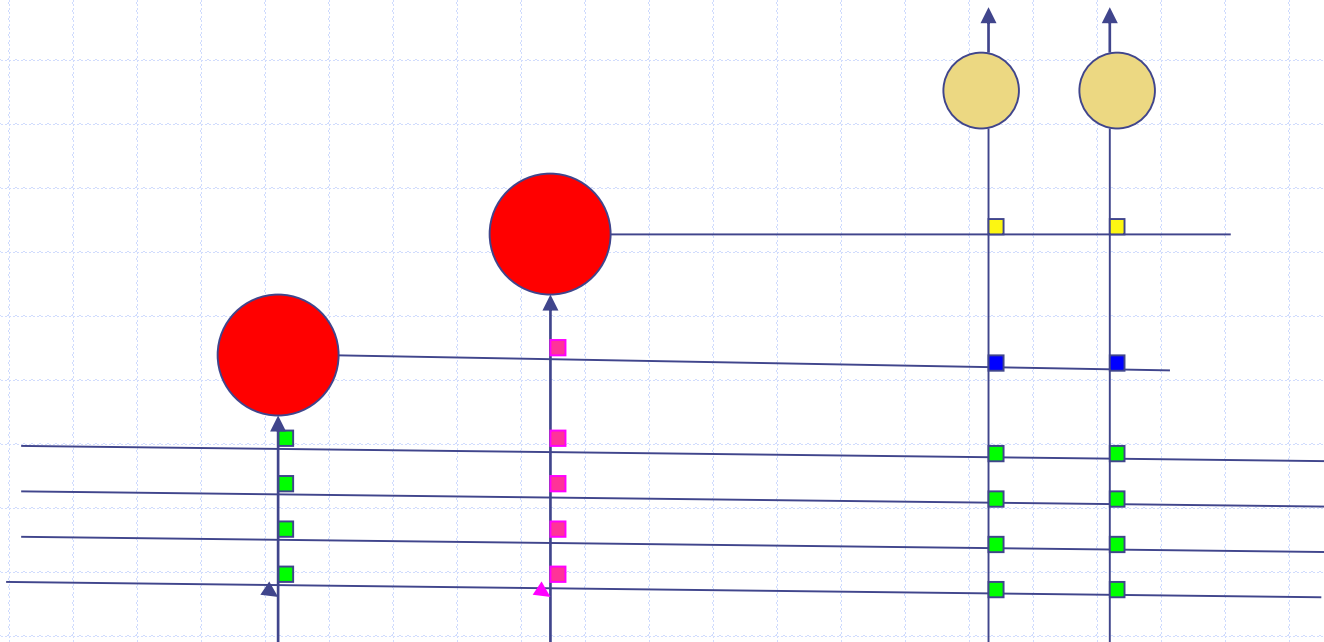
Total processing delay:	20 nanoseconds
Computation speed:	20 billion multiplications and additions per second
Equivalent input bandwidth	4 GBytes/second
Number, resolution of synapses	426, 5 bits (4 bits + sign)
Synapse size	400x70 μm^2
Number of transistors:	40 000
PGA package:	144 pins
Chip size	10mmx9mm (1.5 μm DLM CMOS, ES2, EUROCHIP)
On-chip static RAM	3750 bits
Power dissipation:	P < 0.6W

Felügyelt tanító algoritmusok gyorsító megoldásai

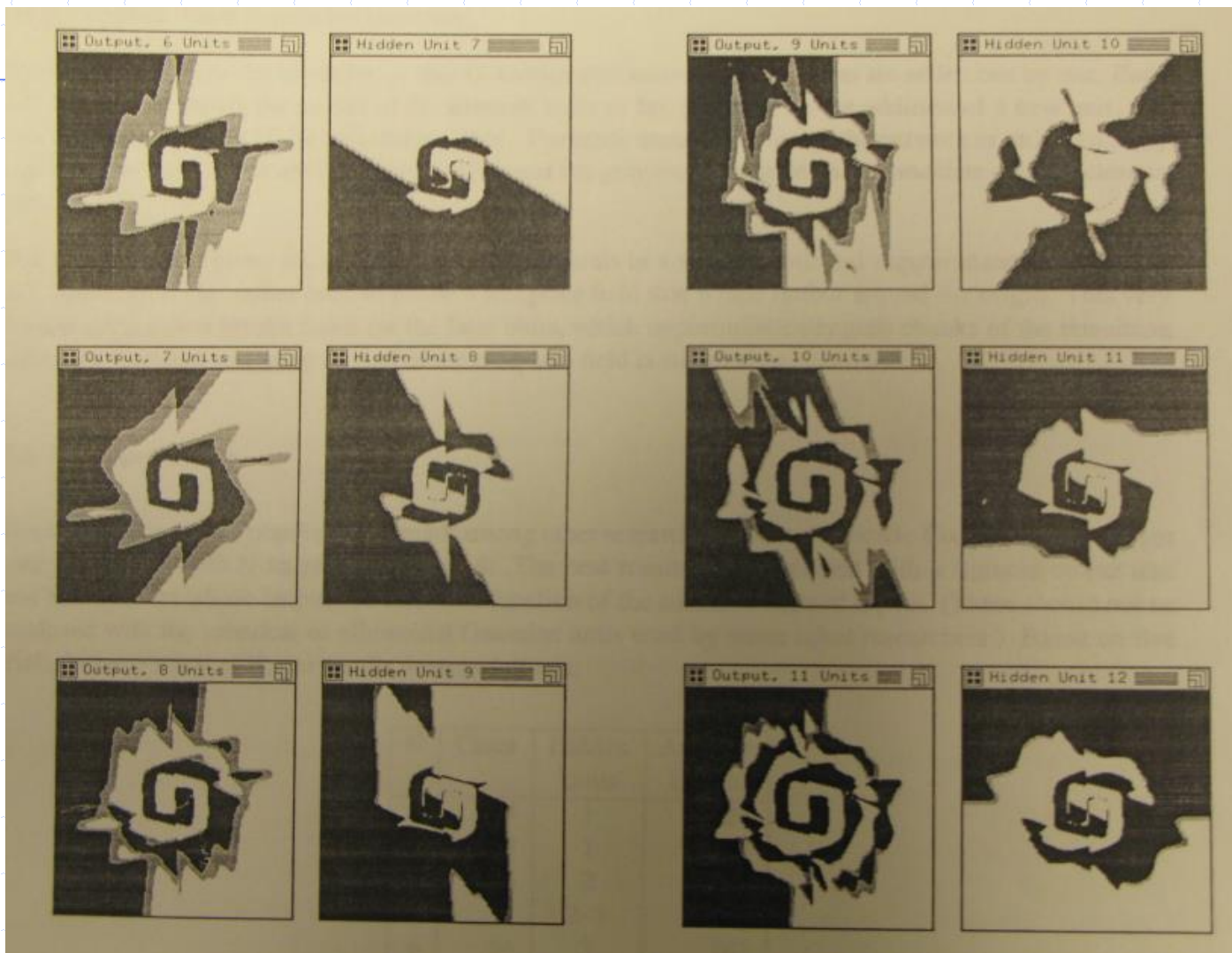
1. Momentum
2. Csökkenő hibahatár („descending epsilon”) módszere
3. Kis súlytényezők kiszűrése („metszés”)
4. Inkrementális hálózat építés „kaszád korreláció”

Lépcsőről-lépcsőre épülő hálózat

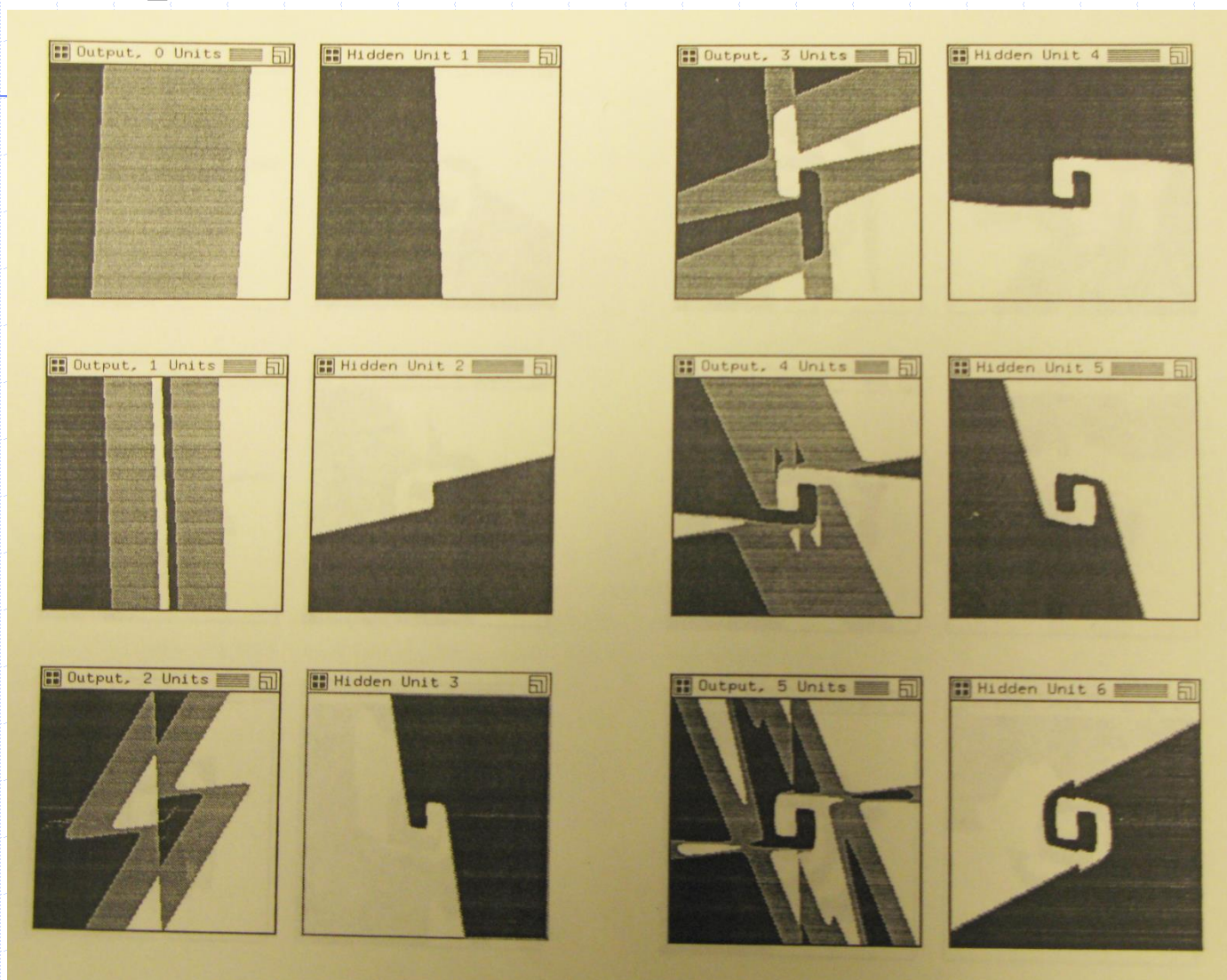
„Kaskád korreláció” (KK) Scott E. Fahlman Christian LeBierre



Kettős spirál tanulása „KK” algoritmussal 2.

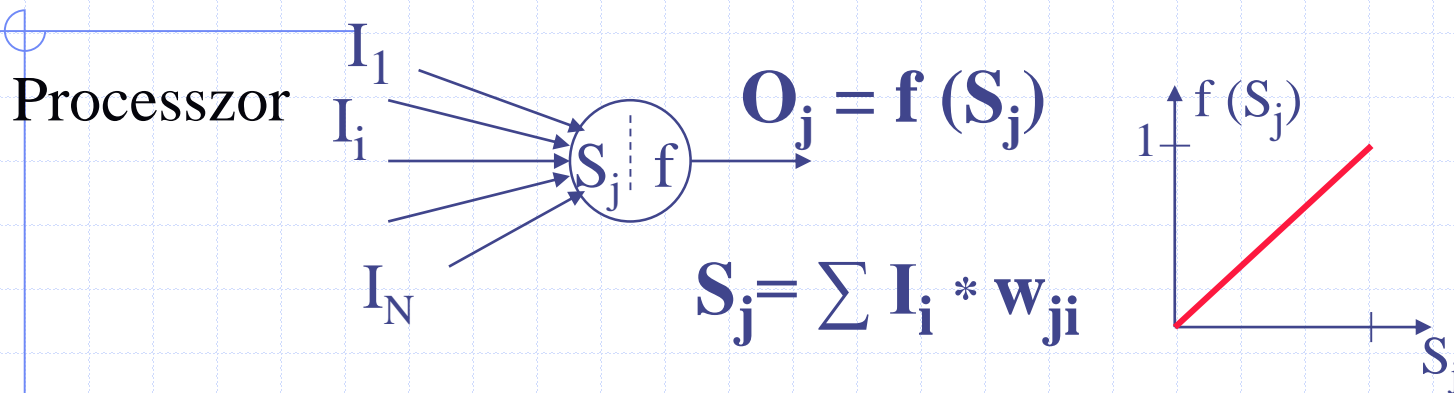


Kettős spirál tanulása „KK” algoritmussal 2.

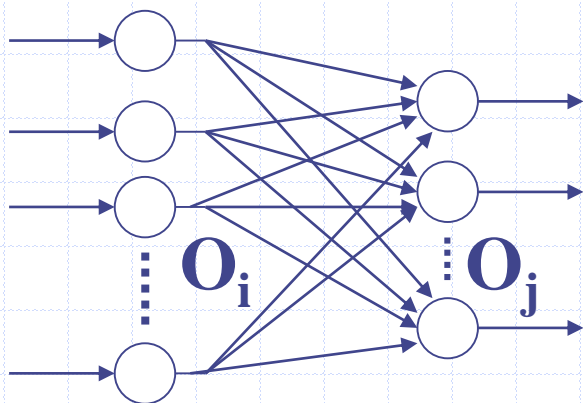


Versengéses (competitive) tanulás

Carpenter, Grossberg 1988



Topológia: egy rétegű előrecsatolt, teljesen összekötött



Megkötések:

- 1.) $\sum w_{ji} = 1$
- 2.) Súly értékek: $0 < w_{ji} < 1$
- 3.) A bemenő vektor bináris

A versengéses tanító algoritmus (Grossberg)

Mottó:

A győztes visz mindent

1. Kezdeti súlytényezők beállítása (inicializálás, véletlenszerű) $0 < W_j < 1$
2. A tanítóminta i -ik értéke (vektora) alapján, a processzorok kimeneti értékeinek kiszámítása. $S_j = \sum O_i * w_{ji}, \quad O_j = f(S_j)$
3. A legnagyobb kimeneti értékű processzor kiválasztása. A győztes visz mindent elv alapján, a győztes kimeneti értéket 1-re, az összes többi kimeneti értéket 0-ra változtatjuk
3. A győztes elem súlytényezőit megváltoztatjuk (csak azokat!)

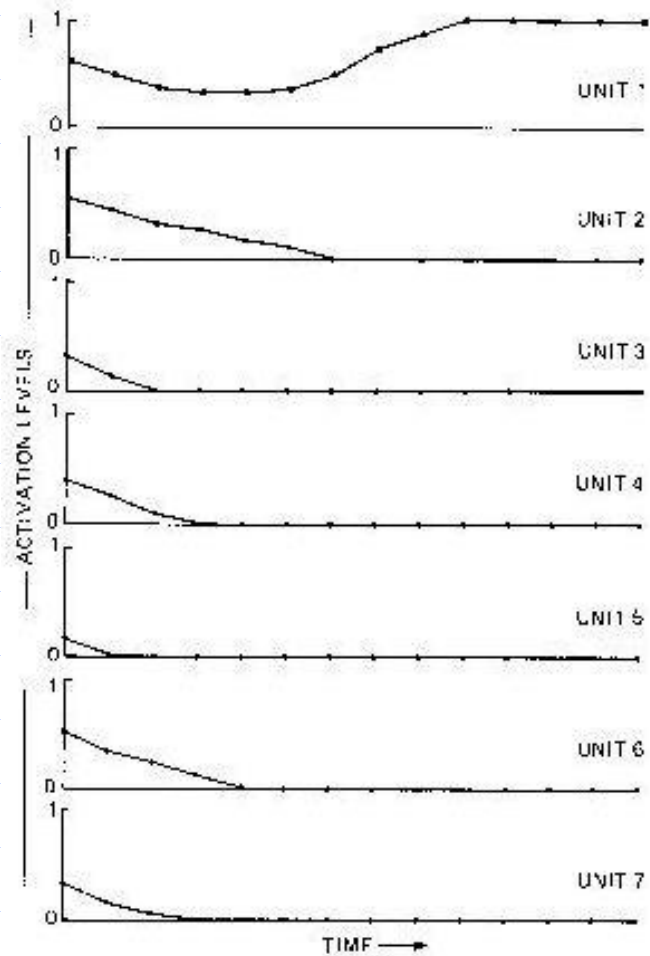
$$\Delta W_{ji}(t+1) = W_{ji}(t) + \Delta w_{ji}, \quad \Delta w_{ji} = \alpha (O_i/m - w_{ji}(t))$$

ahol α = tanulási együttható, $0 < \alpha \ll 1$ (tipikusan 0.01-0.3)

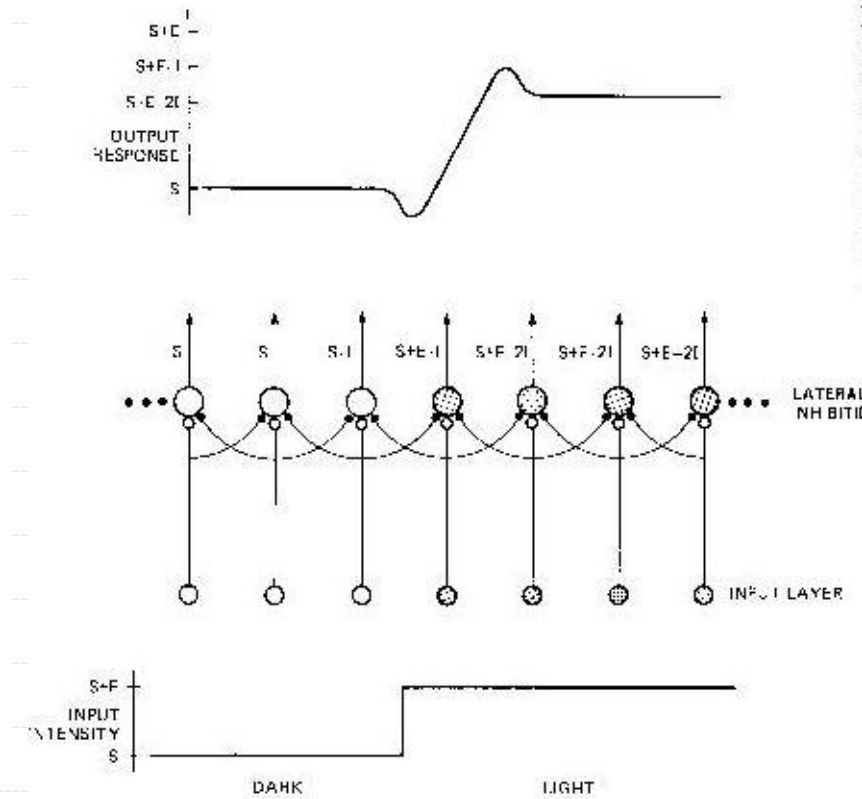
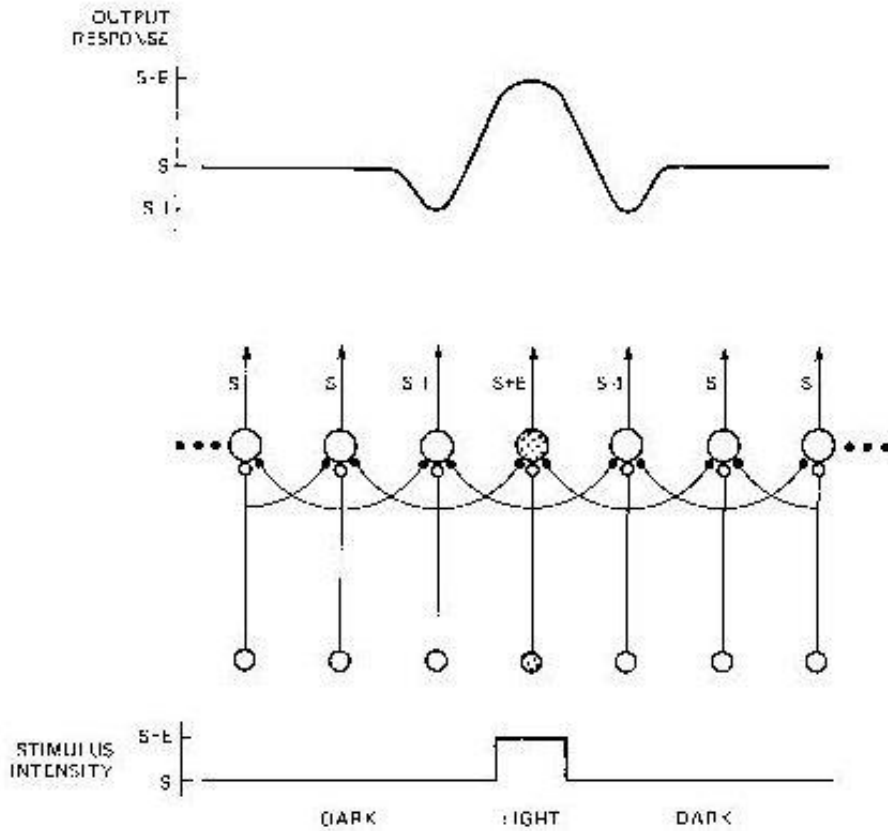
m = az aktív bemenetek száma

5. A 2. 3. 4. pont ismétlése amíg a kimenetek két egymást követő tanítási ciklus során nem változnak.

Versengéses tanulás folyamata

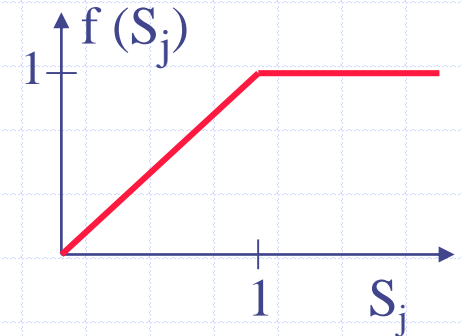
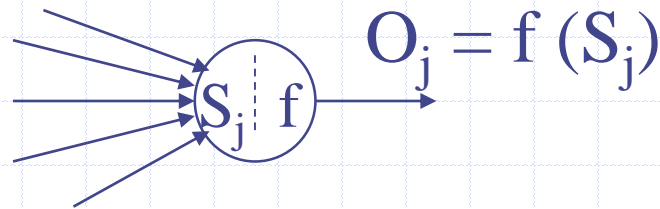


Versengés oldalirányú gátlással



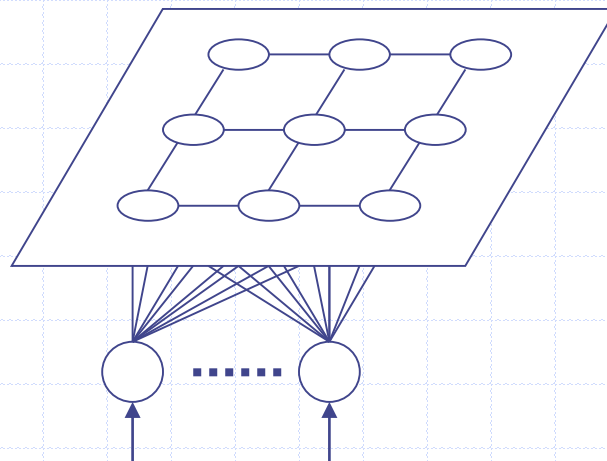
A Kohonen hálózat processzora és topológiája

Processzor:



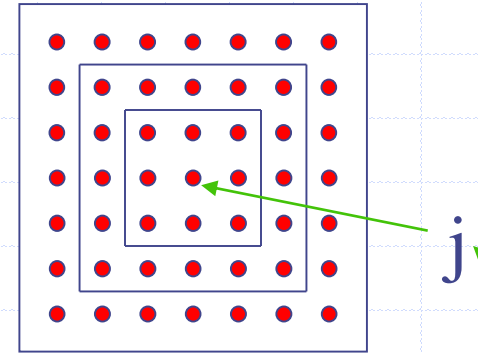
$$S_j = \sum I_i * w_{ji} + \text{társ processzorok aktivációja}$$

Hálózat topológia: egy rétegű, teljesen összekötött, előrecsatolt



A Kohonen tanító algoritmus

- 1.) Kezdeti súlytényezők beállítása
Kezdeti környezet beállítása
- 2.) A bemeneti vektor (tanító minta) rákapcsolása a bemenetekre
- 3.) Minden processzor elemnél a bemenő vektor és a súlyvektor egyezésének (távolságának) kiszámítása



$$d_j = \|I - W_j\| = \sqrt{\sum (I_i - W_{ji})^2}$$

ahol

N = a bemeneti vektor elemeinek száma

I_i = a bemeneti vektor (I) i -ik elemének értéke

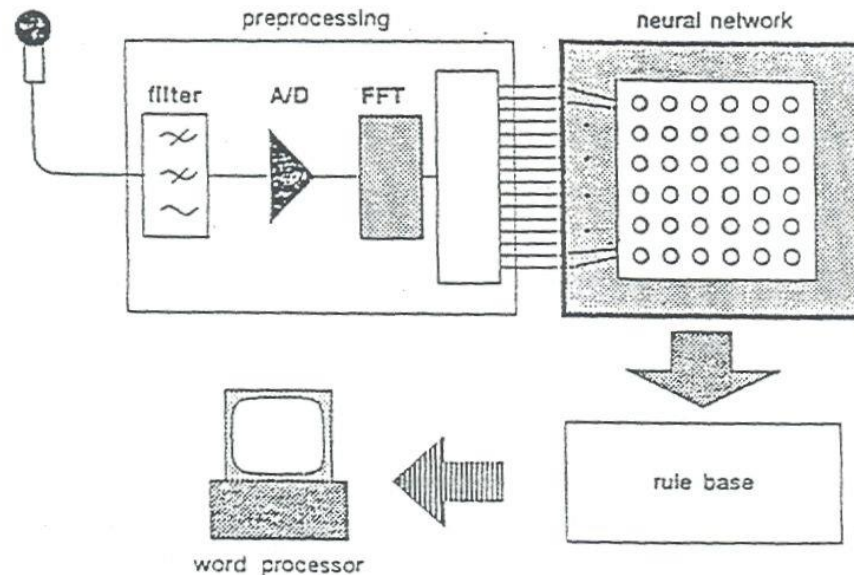
W_{ji} = a j –ik processzor elemhez tartozó, az i -ik bemenettől érkező összeköttetés súlytényezője

- 4.) A legkisebb eltérést mutató processzor kiválasztása (pl. j)
- 5.) A kiválasztott elem (j) környezetében (N_j) a súlytényezők módosítása

- 6.) A 2., 3., 4., 5.-ik lépés ismétlése amíg a kimenetek nem változnak

Kohonen fonetikus írógépe

(Teuvo Kohonen, 1982)



Jellemzői:

5.4 KHz aluláteresztő szűrő, 12 bit A/D, 13.03 KHz mintavétel,
256 pontos Fourier transzformáció (FFT)

Fonémák kézi azonosítása a tanításhoz,

Szabály alapú következtetés (15-20 ezer szabály)

TMS 32010 digitális processzor

Közel folyamatos beszéd feldolgozás 92-97%-os pontosság

A súlytényező megváltoztatása a Kohonen tanuló algoritmusban

$$W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji}(t)$$

$$\text{Ahol } \Delta W_{ji}(t) = \alpha(I_i - W_{ji})$$

$$\alpha(t) = \alpha(0)(1 - t/T), \quad t = \text{az adott tanulási iteráció száma}$$

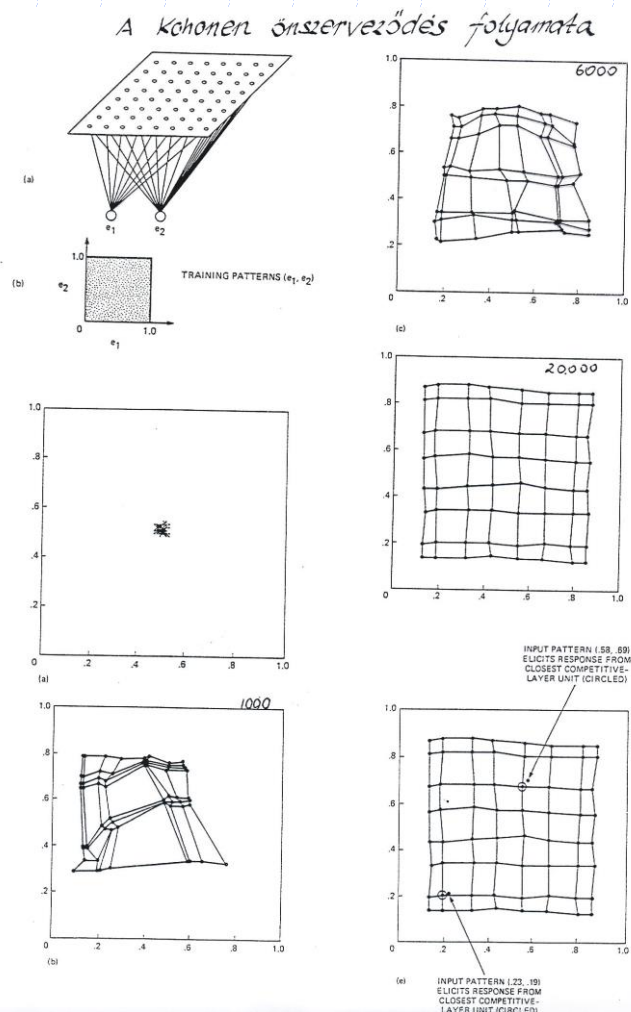
$T = \text{a teljes tanulási ciklusok száma}$

A tanulás során módosított környezet nagysága csökken!

$$N_j(t) = N(0)(1 - t/T)$$

Az „önszerveződés” folyamata a Kohonen hálózatban

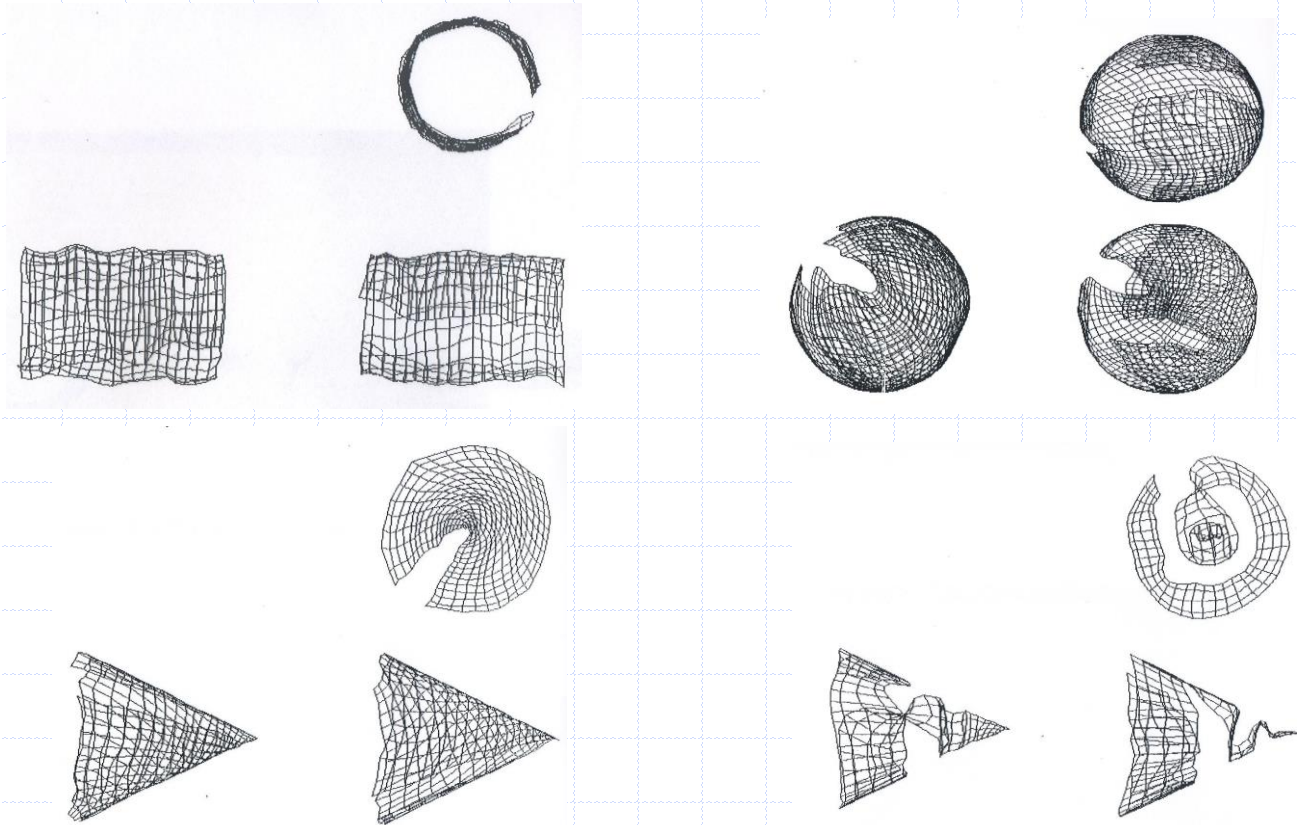
A véletlenszerűen beállított súlytényezők a tanulás során egyre inkább felveszik a tanítóminta statisztikai eloszlását.



Példák 3D-s tárgyak leképezésére 2D-be

Bemenetek száma: 3
Tanítóminta: 1000

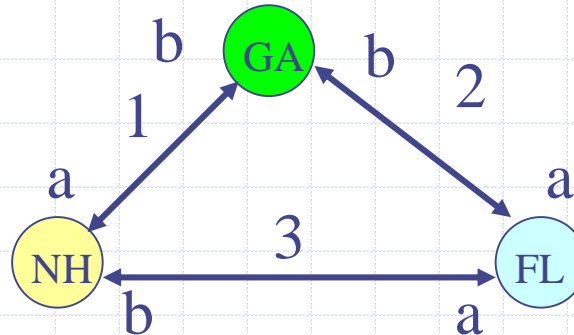
Kimenetek száma: 20
Tanítási ciklus: 15-30



Mikor célszerű neurális hálózatokat alkalmazni?

- ◆ A megoldandó problémával kapcsolatban gazdag adathalmaz áll rendelkezésre
- ◆ A megoldáshoz szükséges szabályok ismeretlenek
- ◆ A rendelkezésre álló adathalmaz nem teljes, hibás adatokat is tartalmazhat
- ◆ Sok összefüggő bemenő adat-, összefüggő kimeneti paraméter áll rendelkezésre

Lágy számítási modellek kapcsolata



- 1 a.) a neurális hálózat tanítása (súlykeresés), topológia megkeresése
b.) az egyed rátermettségének változtatása a tesztelés során
- 2 a.) Fuzzy változók tagsági függvényeinek meghatározása,
Fuzzy szabályok keresése
b.) Fuzzy kiértékelő módszerek alkalmazása az egyedek rátermettségének meghatározására
- 3 a.) a neurális hálózatok adaptív tulajdonságainak bevitele a Fuzzy logikát alkalmazó rendszerekbe.
b.) szabályok automatikus feltárása tapasztalati adatokból

Kérdések

- Miért nem tekintjük az idegrendszert univerzális számítógépnek?
- Az idegsejtek viszonylag lassú működése ellenére miért képes nagy számítási kapacitásra az agy?
- Mivel magyarázható az idegrendszer nagyfokú hibatűrése?
- Miért nem vezet garantáltan jó megoldáshoz a hiba visszaterjesztő algoritmus, és hogyan lehet megkerülni a problémát?