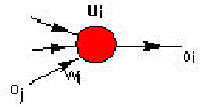
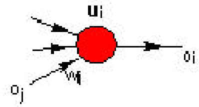


Elõrecsatolt mesterséges neuronháló modellek

1. 1943. *McCulloch - Pitts*: Bináris háló
2. 1958. *Rosenblatt*: Perceptron
3. 1972. *Anderson, ill. Kohonen*: Lineáris asszociátor
   (egymástól függetlenül)
4. 1981. *McClelland és Rumelhart*: Versengés és együtt-
   mûködés
5. 1980, 1983. *Fukushima*: Neocognitron
6. ~1983. *Kohonen és Grossberg*: Szembeterjedéses háló
   (Counter propagation)
7. 1985-86. *Rumelhart + Hinton + Williams*: Back propagation
   Le Cun; Parker
8. 1988. *Kohonen*: Önszervezõ háló



McCulloch - Pitts: Bináris háló

A mesterséges neuron elsõ számítási modellje (1943).

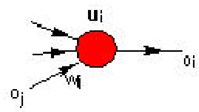
Warren S. McCulloch és *Walter Pitts*, ``A logical calculus of the ideas immanent in nervous activity'', *Bulletin of Mathematical Biophysics*, 5: 115-133.

Az akkori neurobiológiai eredményeken alapul, melynek elõfeltevései:

- a neuron "minden - vagy semmi" alapon mûködik
- bizonyos adott számú szinapszisnak kell lennie gerjesztve az összegzési idõben azért, hogy a neuron egyáltalán kisüljön és ez a szám független a neuron megelõzõ aktivitásától és állapotától
- az egyetlen jelentõs késleltetés az idegrendszerben a szinaptikus késleltetés
- bármely tiltó szinapszis aktivitása egyedül elegendõ a neuron gerjesztésének megakadályozásához
- a hálózat struktúrája idõben nem változik.

A cél az ideghálózat olyan matematikai modelljének létrehozása volt, amely biológiailag pontos és magasszintû kognitív képességeket mutat.

Forrás: http://hebb.cis.uoguelph.ca/~skremer/Teaching/27642/BP/BP.html



McCulloch - Pitts: Bináris háló ..

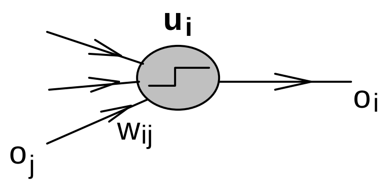
A neuron formális matematikai modellje:

$$a_i(t) = \sum_{j=1}^{n} w_{ij} * o_j(t)$$

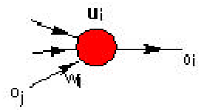
$$o_i(t+1) = H(a_i(t))$$

ahol $o_j(t)$ az u_i neuron j. bemenete, w_{ij} *a bemenet súlya,*
$o_i(t)$ az u_i neuron kimenete a t. idõciklusban,
$a_i(t)$ az aktivációs potenciál az u_i neuronban a t. idõciklusban,
H(a) pedig a *Q* küszöbbel jellemezheto egységugrás függvény:

$$H(a) = \begin{cases} 1,\_ha\_a \geq q, \\ 0\_egyébként. \end{cases}$$



Forrás: http://hebb.cis.uoguelph.ca/~skremer/Teaching/27642/BP/BP.html

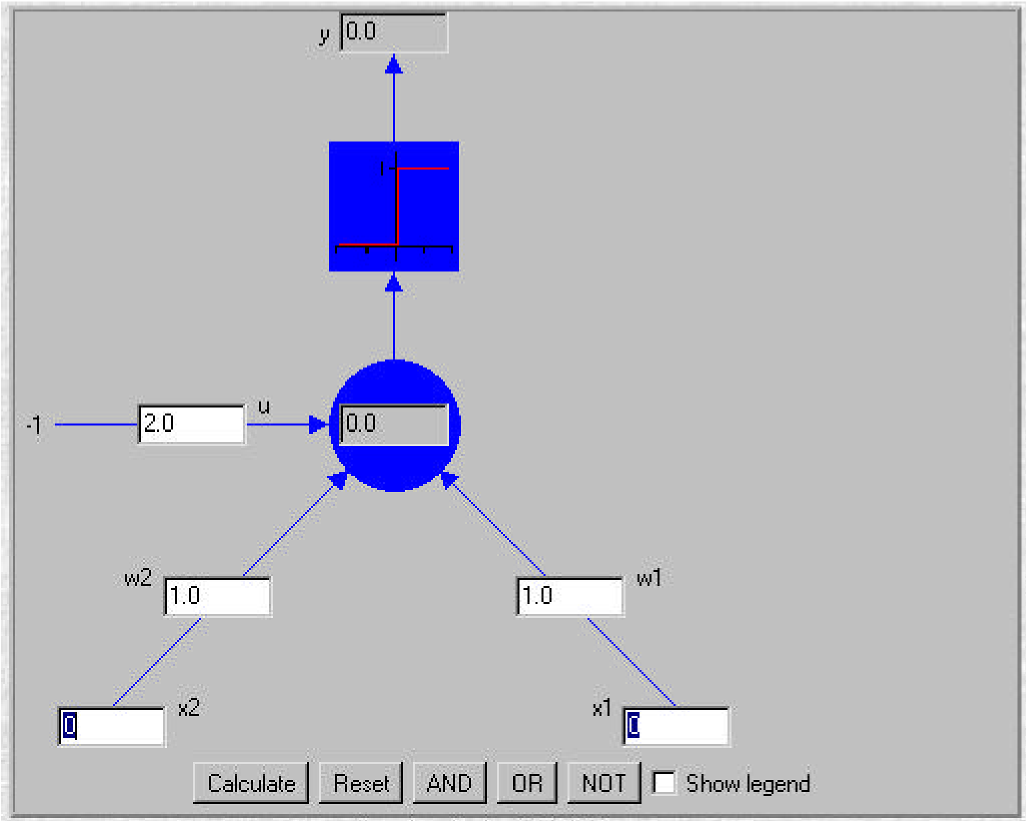


McCulloch - Pitts: Bináris háló ..

Jellemzõi: • Nem tanítható

• Rögzített súlyok.

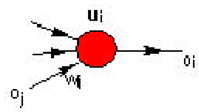
Bemutató Java applet: http://home.cc.umanitoba.ca/~umcorbe9/neuron.html#Inputs



Applet 1: Artificial Neuron

Logikai operátor	*w1*	*w2*	*q*
AND	1	1	2
OR	1	1	1
NOT	-1		0

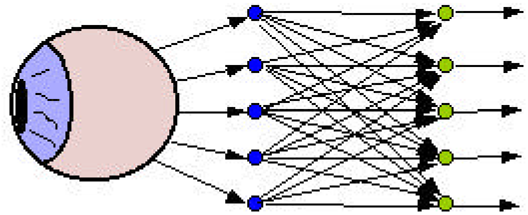
AND, OR, NOT logikai mûveletekkel bármely logikai háló felépíthetõ.



Rosenblatt: Perceptron, 1957

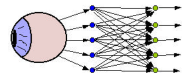
Rosenblatt: Principles of Neurodynamics c. könyv, 1962.

A Perceptron egy jellemzõfelismerõ. Az input bemenetek által reprezentált mintát képes besorolni a neki betanított mintaosztályok valamelyikébe.



Jellemzõk:

- A látást, a retinát modellezte
- A retina-modell mátrix alakban elrendezett fényszenzorai képezték az inputot
- A szenzorkimeneteket mesterséges neuronok csoportjához vezették, mely csoport elemei különbözõ mintákat ismertek fel
- Az output réteg neuronja addig nem tüzel, amíg egy adott gerjesztési szint, azaz egy adott típusú inputmintázat nem jelentkezik.
- Ez a modell azon a tapasztalati megfigyelésen alapult, miszerint a hasonló összetettségû alakzatok közül egyesek könnyebben megtanulhatók és felidézhetõk, mint mások.



Rosenblatt: Perceptron ..

Jellemzõk..:

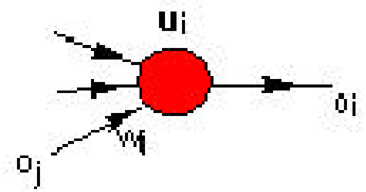
- Képes megtanulni egyszerû minták felismerését
- Kétrétegû (ahol az input réteg csak elosztó szerepû, azaz valójában a Perceptronok egy rétegben vannak), heteroasszociatív, legközelebbi szomszéd - elvû mintafelismerõ
- Folyamatos értékû és bináris inputtal egyaránt mûködhet
- Offline módon tanul, idõciklusokban mûködik és hibajavító módszerrel tanulja meg az *(X_k, Y_k), k=1..m* mintapárokat. A *k.* mintapár input vektora $X_k = (x_1^k, x_2^k, \dots, x_n^k)$ analóg értékekkel, az $Y_k = (y_1^k, y_2^k, \dots, y_p^k)$ kimeneti vektor bipoláris [-1;+1] értékekkel bír.
- Egy Perceptron azt dönti el, hogy egy input minta két osztály közül melyikhez tartozik.
- Felügyelt tanulást végez.

Rosenblatt: Perceptron ..

A Perceptron betanítása (egy kimenõ rétegbeli neuronra mutatva)

- Inicializáljuk a *wij* súlyokat és a *Q* küszöböt kicsi véletlen értékekre.
- Adjuk meg t. inputot és a hozzátartozó elvárt outputot: $x_1(t), x_2(t), \ldots, x_n(t)$, valamint $d(t)$.
- Számítsuk az inputhoz tartozó kimenetet:

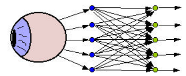
$$y(t) = H\left(\sum_{j=1}^{n} w_j(t) x_j(t) - q\right)$$



ahol $H(x) = \begin{cases} 1, \; ha \; x \geq 0, \\ -1 \; egyébként. \end{cases}$

- Módosítsuk a súlyokat:
  ha $d(t) = y(t)$ $\quad w_j(t+1) = w_j(t)$ $\quad$ változatlan,
  ha $y(t) = -1$ a kimenet, de $d(t) = 1$ az elvárt: $\quad w_j(t+1) = w_j(t) + x_j(t)$,
  ha $y(t) = +1$ a kimenet, de $d(t) = -1$ az elvárt: $\quad w_j(t+1) = w_j(t) - x_j(t)$,

- Amennyiben az utolsó lépésben a súlyok változtak: ugrás a 2. pontra, egyébként a betanítás kész.



Rosenblatt: Perceptron ..

Widrow – Hoff delta szabály

Widrow és Hoff az algoritmus módosítását javasolták. Errorként, hibaként értelmezték az elvárt kimenet és a számított kimenet eltérését és ennek nagyságával arányos súlymódosítást javasoltak:

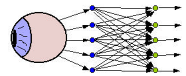
- Módosítsuk a súlyokat:

  $w_j(t+1)= w_j(t) + h*[d(t) – y(t)]* x_j(t)$

  ahol h a tanulási együttható $(0< h <1)$

A módosított Perceptront az ADALINE-ban és a sok ilyen neuron összekapcsolásával kapott MADALINE-ban alkalmazták. (MultiAdaline)

Az h tanulási együttható szabályozza a konvergencia fokát. Célszeru menet közben értékét változtatni, tuningolni, hogy eleget tudjon tenni az input eloszlásában meglevo nagy változásokhoz való gyors alkalmazkodásnak és az algoritmus végén a stabil súlyállapot eléréséhez szükséges input-átlagolásnak.



Rosenblatt: Perceptron ..

Az algoritmus konvergenciája

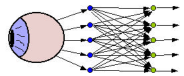
- Rosenblatt bebizonyította, hogy ha egy osztályozási feladatnak van megoldása, akkor a Perceptron betanítása **konvergens** és a korrekt osztályozás **véges** ido alatt elérheto.

- Ez azt mutatja, hogy a Percepron betanulása csak a reprezentációs képessége által korlátozott.

- A reprezentációs kapacitás szembenállása a adaptációs képességgel egy fontos kérdése a mesterséges neurális hálóknak.

- A bizonyított konvergencia és a számítástechnika eredményei iránt az iparban, a katonai kutatásokban és az alkalmazások modellezésében megnyilvánuló növekvo érdeklodés a kutatás és a pénzügyi támogatás eroteljes növekedését eredményezte.

- A teljes betanítóminta halmazra értelmezett hiba: $E = \frac{1}{2}\sum_{i=1}^{p}\left|y_i - d_i\right|^2$
  A hibának nullához kell konvergálnia.



Rosenblatt: Perceptron ..

A Perceptron hibája

A Perceptron **csak lineárisan szétválasztható** minták osztályozására képes.

Marvin Minsky és Seymour Papert 1969-ben jelentette meg a **Perceptrons** címu könyvet,melyben elemzik a Perceptron adatreprezentációs képességeit.

Lineáris függetlenség

A vizsgálathoz vegyük a Perceptron egy egyszeru, kétbemenetes esetét.

$$y(t) = H\left(\sum_{j=1}^{n} w_j(t)x_j(t) - q\right)$$

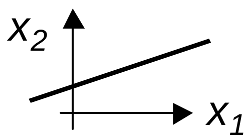
A H(x) függvény 0 bemeno értékénél van az ugrás, mely mentén a függvényértékek két értékre szeparálódnak:

$$\sum_{j=1}^{n} w_j(t)x_j(t) - q = 0$$

Csak két bemenetes esetre konkretizálva és kibontva:

$w_1(t) * x_1(t) + w_2(t) * x_2(t) - q = 0$, melyet átrendezve egy **egyenes** adódik:

$$x_2(t) = \frac{-w_1(t)}{w_2(t)} x_1(t) + \frac{q}{w_2(t)}$$

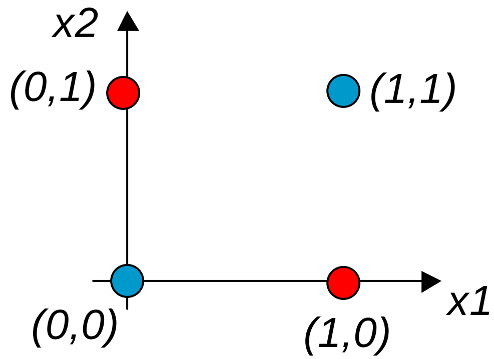


dr.Dudás László

Rosenblatt: Perceptron ..

A Perceptron hibája ..

A feladatok egy része olyan pontok szétválasztását igényli, melyeket lineáris szeparáló objektummal (egyenes, sík, hipersík) nem lehet elkülöníteni, az osztályokat nem lehet létrehozni. Ezek legnevezetesebbike a **XOR** logikai muvelet.

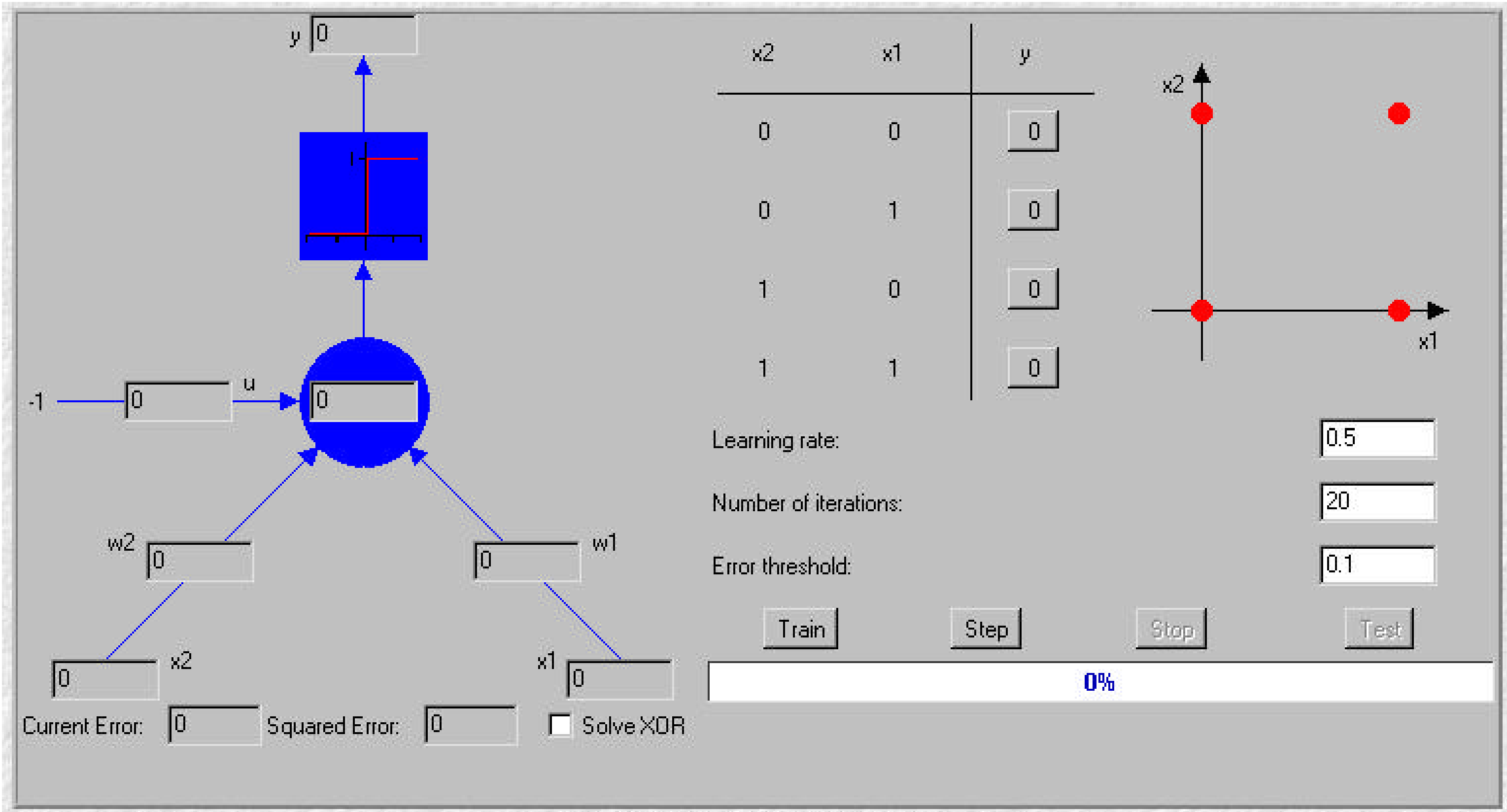


XOR		***x1***	
		0	1
	0	0	1
x2			
	1	1	0

dr.Dudás László

Rosenblatt: Perceptron ..

A Perceptron muködésének demonstrálása:
http://home.cc.umanitoba.ca/~umcorbe9/neuron.html#Inputs
Tanulmányozható a betanítás és a mûködés egyaránt.



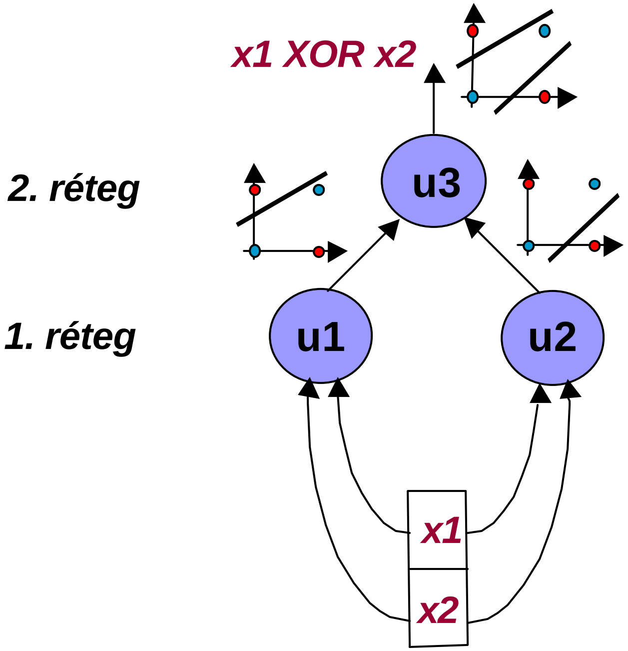
Applet 2: Perceptron Learning

Többrétegû Perceptron (Back propagation)

Cél: a lineárisan szeparálhatatlan inputok megkülönböztethetõségének elérése.

Megoldás: Perceptronok kettõ, vagy több rétegben

Az elsõ réteg neuronjai alkalmasak az inputminta kisebb, egyenessel elszeparálható részeinek megkülönböztetésére, a második réteg neuronja pedig alkalmasak az elsõ réteg neuronjaitól jövõ kétféle osztályba tartozó jelek megkülönböztetésére. Például a XOR függvény megoldása:



Az u1 neuron észleli a (0,1) bemenetet, az u2 pedig az (1,0) bemenet esetén ad 1-et. Az u3 neuronnak csak egy VAGY függvényt kell realizálnia.

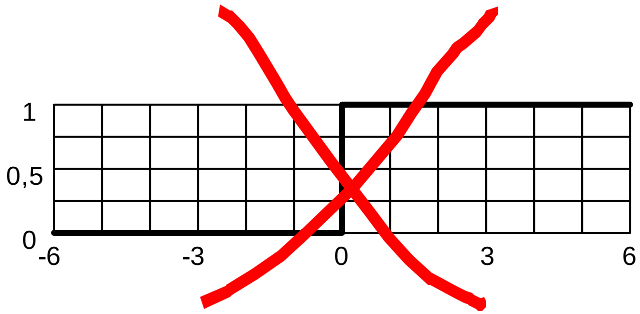
dr.Dudás László

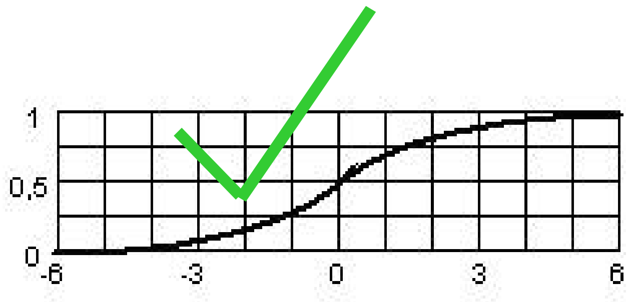
Többrétegû Perceptron ..

Probléma: a súlyok csak megadással állíthatók be, **a háló nem képes tanulni**.

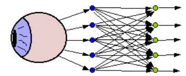
Oka a kredit hozzárendelési probléma: a lépcsõs átviteli függvény, mely miatt a második rétegre nem jut el információ az x1, x2 bemenetek tényleges értékérõl. Mivel a tanulás megfelel az aktív inputok és az aktív neuronok közötti kapcsolat erõsítésének, lehetetlen a háló megfelelõ részét erõsíteni, mivel az aktuális inputok el vannak vágva az outputtól a közbensõ réteg által. A kétállapotú neuron, lévén „tüzel”, vagy „nem tüzel” állapotban, nem ad semmi jelzést a súlyok módosításának mértékérõl. A küszöbhöz közeli, a neuront éppen csak tüzelésre késztetõ inputok súlyait nem kellene ugyanolyan mértékben módosítani, mint azokét, amelyek hatására a neuron aktiváltsága messze esik a küszöbtõl. A bûnös az ugrásfüggvény.

Megoldás: alkalmazzunk a küszöb közelében **átmenettel bíró átviteli függvényt**, mint pl. a lineáris küszöb, vagy méginkább a szigmoid függvényt.





dr.Dudás László



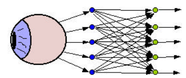
Többrétegû Perceptron ..

A többrétegû Perceptronban megjelenik egy, vagy több **rejtett** réteg, mely kapcsolatot teremt a bemeneti réteg és a kimeneti réteg között. A rejtett réteg és a kimeneti réteg minden neuronja egy-egy Perceptron, de ferdeátmenetes átviteli függvénnyel. Az input réteg neuronjainak csak bemenetijel-szétosztó szerepük van.

A többrétegû Perceptron összetett minták megtanulására képes, de ehhez **új tanulási szabályra** van szükség: ez az **általánosított delta szabály**, vagy **hibavisszaterjesztés (backpropagation).** Kidolgozóik **Rumelhart, McClelland és Williams (1986),** korábbiak **Parker 1982, Verbos 1974.**

A tématerület legfontosabb irodalma:
Rumelhart-McClelland: Parallel Distributed Processing c. könyv.

A tanítás alapgondolata: az elvárt és a számított kimenetek eltérését, mint a háló súlyaitól függõ hibát értelmezzük és ezen, a súlyok terében értelmezett hibafüggvényen hajtunk végre egy minimális pont keresést.



A többrétegû Perceptron tanítása

- Jelölések: E_k a k mintapárhoz számított hiba, d_{ki} az elvárt kimenet a k minta esetén az *i.* neuronnál, o_{ki} az aktuális kimenet az *i*. neuronnál, w_{ji} a súly a j. neurontól az i. felé haladó kapcsolatnál.
- A hiba az összes mintapárra:

$$E_k = \frac{1}{2}\sum_{i=1}^{n}\left(d_{ki} - o_{ki}\right)^2$$

- Az aktivációs potenciál minden egyes *i*. neuronra a *k*. mintapárnál:

$$net_{ki} = \sum_{j=1}^{n} w_{ji} o_{kj}$$

- Az *i.* neuron kimenetét származtassuk az aktivációs potenciálból az f_i szigmoid átviteli függvénnyel (bármilyen monoton növekvõ folytonosan differenciálható függvény jó):

$$o_{ki} = f_i(\ net_{ki})$$

- A hibafelület w_{ij} súly irányába esõ deriváltja:

$$\frac{\partial E_k}{\partial w_{ji}} = \frac{\partial E_k}{\partial net_{ki}} \frac{\partial net_{ki}}{\partial w_{ji}}$$

A többrétegû Perceptron tanítása ..

- Az irányszerinti derivált átrendezett alakjába behelyettesítve az aktivációs potenciált:

$$\frac{\partial net_{ki}}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}} \sum_{l=1}^{n} w_{li} o_{kl} = \sum_{l=1}^{n} \frac{\partial w_{li}}{\partial w_{ji}} o_{kl} = o_{kj}$$

Mivel $\frac{\partial w_{li}}{\partial w_{ji}} = 0$, kivéve amikor $l=j$, amikoris 1.

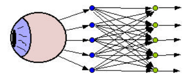
- A hiba változását definiálhatjuk, mint a függvényét a háló bemenetei változásának egy neuronhoz viszonyítva:

$$-\frac{\partial E_k}{\partial net_{ki}} = d_{ki}$$

- Ezzel az w_{ji} súly szerinti deriváltja a hibának:

$$-\frac{\partial E_k}{\partial w_{ji}} = d_{ki} o_{kj}$$

- E_k értékének csökkentése ezért a $d_{ki}o_{kj}$ értékével arányos súlymódosításokat jelent: $D_k w_{ji} = h\ d_{ki} o_{kj}$.



A többrétegû Perceptron tanítása ..

- Mostmár csak minden egyes neuronra kell ismernünk d_{ki} értékét, hogy csökkenthessük az E_k hibát.
- A hibaváltozást megadó korábbi képletünket tovább alakítva kapjuk:

$$d_{ki} = -\frac{\partial E_k}{\partial net_{ki}} = -\frac{\partial E_k}{\partial o_{ki}} \frac{\partial o_{ki}}{\partial net_{ki}} \qquad (1)$$

- A második egyenloségbol a szigmoid függvénnyel számított kimenet alkalmazásával:

$$\frac{\partial o_{ki}}{\partial net_{ki}} = f_i'(net_{ki})$$

- Ugyanonnan, de tekintsük most az elso egyenloséget! A hiba képletébol differenciálhatjuk Ek-t oki szerint:

$$\frac{\partial E_k}{\partial o_{ki}} = -(d_{ki} - o_{ki})$$

- Ilymódon $d_{ki}=f'_i(net_{ki})(d_{ki}-o_{ki})$. Ez hasznos a neuronok outputjának meghatározásához, mivel az elvárt érték és a számított kimenet egyaránt rendelkezésre áll, azonban nem a rejtett neuronokra, mivel azok elvárt értéke nem ismert. Rejtett neuronokra a következot tehetjük:

A többrétegû Perceptron tanítása ..

- Ha az i. neuron nem kimeneti neuron, a láncszabályt tovább alkalmazva a következõt írhatjuk:

$$\frac{\partial E_k}{\partial o_{ki}} = \sum_{l=1}^{p} \frac{\partial E_k}{\partial net_{kl}} \frac{\partial net_{kl}}{\partial o_{ki}} = \sum_{l=1}^{p} \frac{\partial E_k}{\partial net_{kl}} \frac{\partial}{\partial o_{ki}} \sum_{j=1}^{n} w_{jl} o_{kj} = -\sum_{l=1}^{n} d_{kl} w_{il}$$

ehhez az akciós potenciált és a hiba változását megadó kifejezéseket használva és figyelembe véve, hogy az összeg kiesik, mivel a parciális derivált csak egy értékre nem nulla. Végül a fenti kifejezést az (1)-be helyettesítve végül ezt kapjuk:

$$d_{ki} = f_i'(net_{ki}) \sum_{l=1}^{p} d_{kl} w_{il}$$

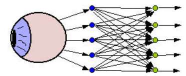
Ez az egyenlet megadja a hibafüggvény változását a hálózat súlyainak függvényében. Módot ad a hibafüggvény megváltoztatására, csökkentésére. Elõbb a kimeneti neuronokra számítjuk a hibát, majd visszafelé terjesztjük a megelõzõ rétegekre hogy azok is beállíthassák súlyaikat.Innen ered a betanítási módszer **backpropagation** elnevezése.

A többrétegû Perceptron tanítása ..

- A nemlineáris **szigmoid** függvény alkalmazásának elõnye, hogy egészen hasonlít a lépcsõs függvényre és hasonló viselkedést produkál.
- Alakja $f(net)= 1/(1+e^{-k\ net})$, értéke 0 és 1 közé esik. A k konstans szabályozza a meredekségét, $k= \infty$ esetén a lépcsosfüggvényt adja.
- Alkalmazásának elõnye, hogy egyszerû a deriválása:

$$f'(net) = k^*e^{-k^*net}/(1+e^{-k^*net})^2 = k^*f(net)(1-f(net)) = k^*o_{ki}(1-o_{ki})$$

A deriváltja ezek szerint a kimenetek egyszerû függvénye.



A többrétegû Perceptron tanulási algoritmusa

- A súlyok és a küszöbök beállítása kisértékû véletlenszámokra.
- A bemeneti minta és az elvárt kimenet megadása.
  Az input: $X_k = x_1, x_2, \ldots, x_n$, az elvárt kimenet: $D_k = d_1, d_2, \ldots, d_m$, ahol n a bemeneti rétegbeli neuronok, m a kimeneti rétegbeli neuronok száma.
  Állítsuk be w_1 értékét $-Q$ értékre, és legyen x_1 mindig 1.
- Számítsuk az aktuális bemeneti mintával a kimenetet.
  Mindegyik réteg számítja a saját

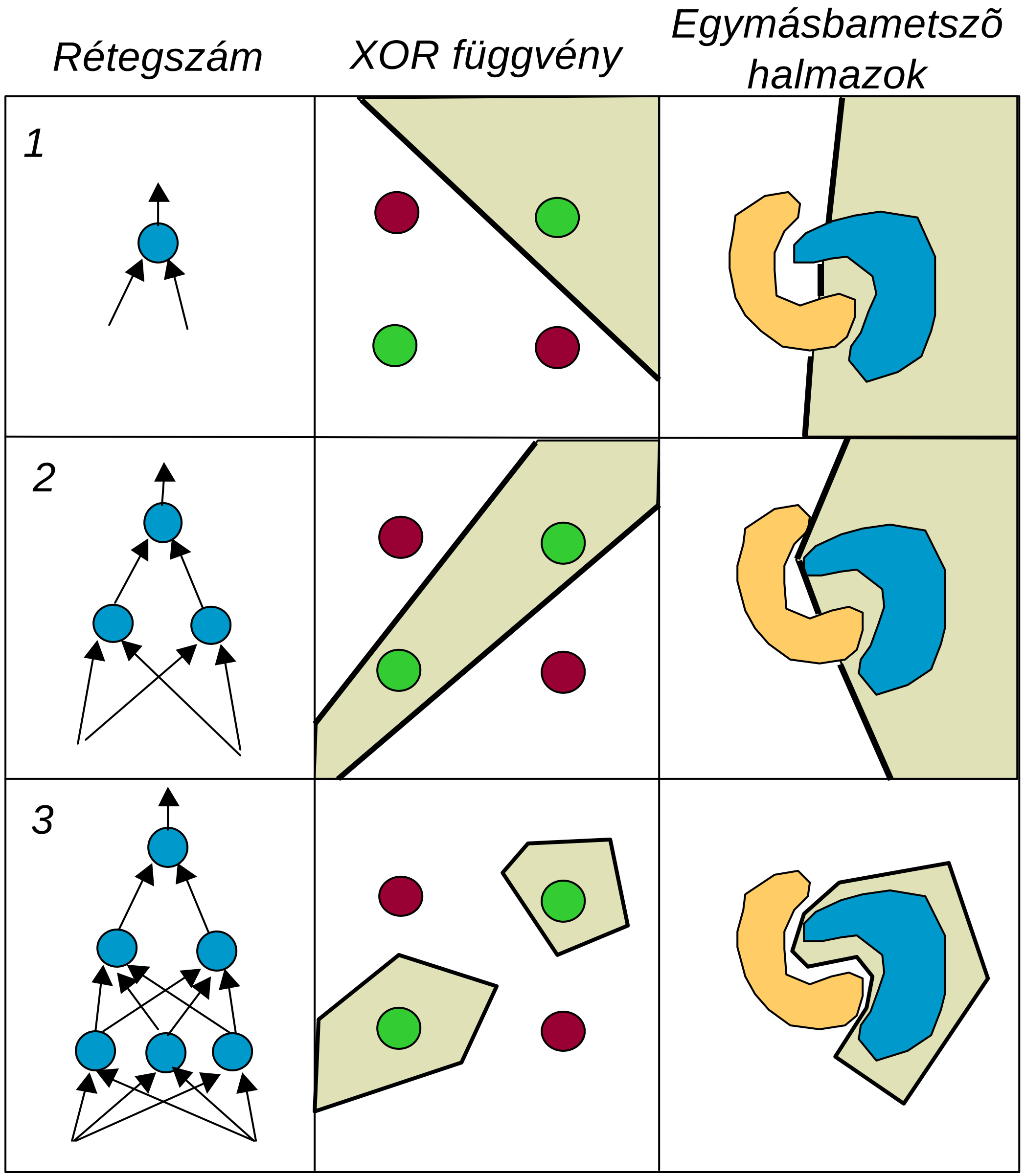
  $$y_{ki} = f\left[\sum_{j=1}^{n} w_{ij} x_j\right]$$

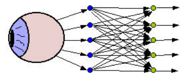
  kimenetét és továbbadja a következo rétegnek. A kimeneti rétegen megjelennek a számított értékek: o_{ki}
- Módosítsuk a súlyokat
  Induljunk a kimeneti rétegtol és haladjunk visszafelé!

  $$w_{ji}(t+1) = w_{ji}(t) + h d_{ki} o_{ki}$$

  h a tanulási együttható, d_{ki} a hiba az $i.$ neuron kimenetén a $k.$ minta bemutatásakor. A kimeneti réteg neuronjaira:
  $d_{ki} = l * o_{ki} * (1 - o_{ki})(d_{ki} - o_{ki})$
  A rejtett réteg neuronjaira: $d_{ki} = l \cdot o_{ki}(1 - o_{ki}) \sum_{l=1}^{p} d_{kl} w_{il}$, l fut az i. neuron feletti réteg neuronjaira.

A többrétegû Perceptron osztályozó képessége





A többrétegû Perceptron jellemzõi

- Többrétegû Perceptron – perceptronszerû neuronokból álló rétegek
- Elõrecsatolt mûködés
- Hiba-hátraterjesztéses, felügyelt tanulás
- Folyamatosan differenciálható átviteli függvény, többnyire szigmoid
- Három, perceptronokból álló réteg bármilyen mintaosztályozásra alkalmas
- A háló az input szerkezetének belsõ leképzését hozza létre.
- A betanuláshoz a mintapárok többszöri bemutatása szükséges
- Energiafelülettel szemléltethetõ
- A tanulás nem mindig konvergens
- Léteznek a tanulási problémákat megoldó továbbfejlesztései
- A radial basis függvényre épülõ változat hiperellipszoidokat alkalmaz és garantáltan konvergens
- Változatos valós alkalmazásokban használták.

dr.Dudás László

Anderson, ill. Kohonen: Lineáris asszociátor, 1972

Jellemzoi:

- a neuron kisülésekor a tüzelés frekvenciája változott az input függvényében
- nem egyszeru küszöb átviteli függvény
- tudta a *XOR* függvényt is
- tanulás a Delta - szabállyal :

  $\Delta w_{ij} = h(\, T_i(t) - a_i(t))\, o_j(t)$, ahol:

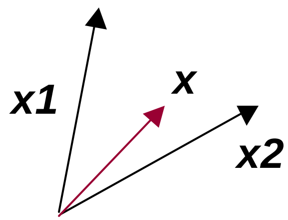
$\Delta\, w_{ij}$	súlyváltozás az ij kapcsolatnál (tanulás)
h	tanulási tényezo
T_i	az output minta értéke helyes válasz esetén ez lenne (elvárt érték)
a_i	a vizsgált *i*. neuron aktivációs potenciálja
o_j	bemenet az *i.* neuronnál a *j.* neuron kimenetérol (, vagy a *j*. inputról)

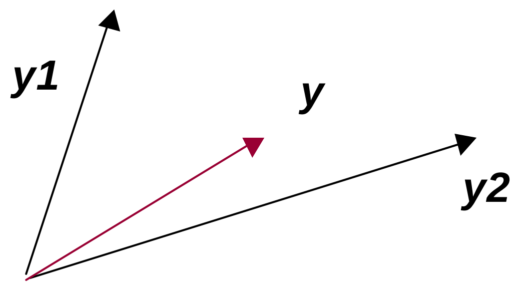
- ha a minták lineárisan függetlenek, akkor többszöri bemutatással betaníthatók

Anderson, ill. Kohonen: Lineáris asszociátor ..

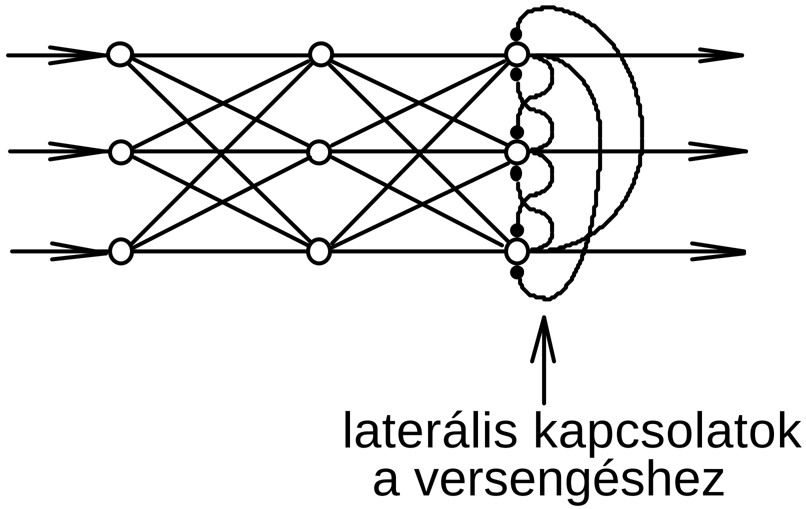
Kohonen által bevezetett fogalmak:

- **Autoasszociativitás:** a minta egy részének megadása elegendõ a teljes minta elõhívásához.
  Adott a betanítás: $\Phi(\mathbf{x}_j)= \mathbf{x}_i$, akkor ha $\mathbf{x}$ input közel van $\mathbf{x}_i$-hez, akkor a kimenet $\mathbf{x}_i$ lesz.
- **Heteroasszociativitás:** a minta egy részének megadása elegendõ a teljes inputhoz társított output elõhívásához.
  Adott a betanítás: $\Phi(\mathbf{x}_j)= \mathbf{y}_i$, akkor ha $\mathbf{x}$ input közel van $\mathbf{x}_i$-hez, akkor a kimenet $\mathbf{y}_i$ lesz.
- **Interpolatív asszociativitás:**
  Adott a betanítás: $\Phi(\mathbf{x}_j)= \mathbf{y}_i$, ekkor ha $\mathbf{x}$ input az $\mathbf{x}_i$-k (i=1,2,…,n) kombinációja, akkor a kimenet $\mathbf{y}$ az $\mathbf{y}_i$-k (i=1,2,…,n) kombinációja lesz.





McClelland és Rumelhart: Versengés és együttmuködés 1981



A legnagyobb kimeneti értéku neuron a laterális kapcsolatok segítségével elnyomja a többi kimenetét, o nyer.

McClelland és Rumelhart a Szófelismero modellükben a szavak hasonlóságát a neuronok közötti izgató és tiltó egymásrahatásokon keresztül mutatták be:

- egyik réteg: betuk,
- másik réteg: szavak.

A szóban eloforduló betuket reprezentáló neuronok között a betu-rétegben **együttmuködés**, egymás hatásának erosítése van.

A szavak azonos pozíciójában lévo betuk között **versengés** van (<u>T</u>AKE, <u>M</u>AKE).

McClelland és Rumelhart: Versengés és együttmuködés ..

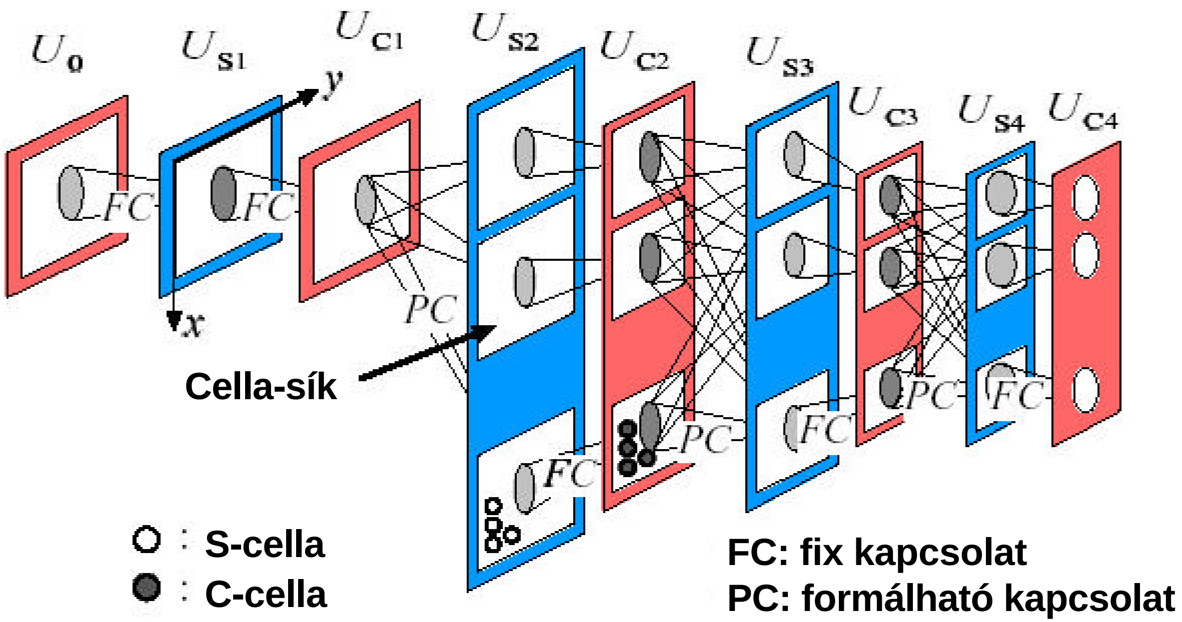
- **Versengés:** a legnagyobb kimenetû neuron a laterális kapcsolatok segítségével elnyomja a többi kimenetét, õ nyer.
- **Együttmûködés:** az együttmûködõ neuronok erõsítik egymást, hogy együtt nyerjenek.

	T	*M*	*A*	*E*	*K*
Pozíció 1	+	+			
Pozíció 2			+		
Pozíció 3				+	
Pozíció 4					+

Versengés az elsõ pozícióért. A T betû neuronja és az 1. pozíciót jelentõ neuron erõsítik egymást

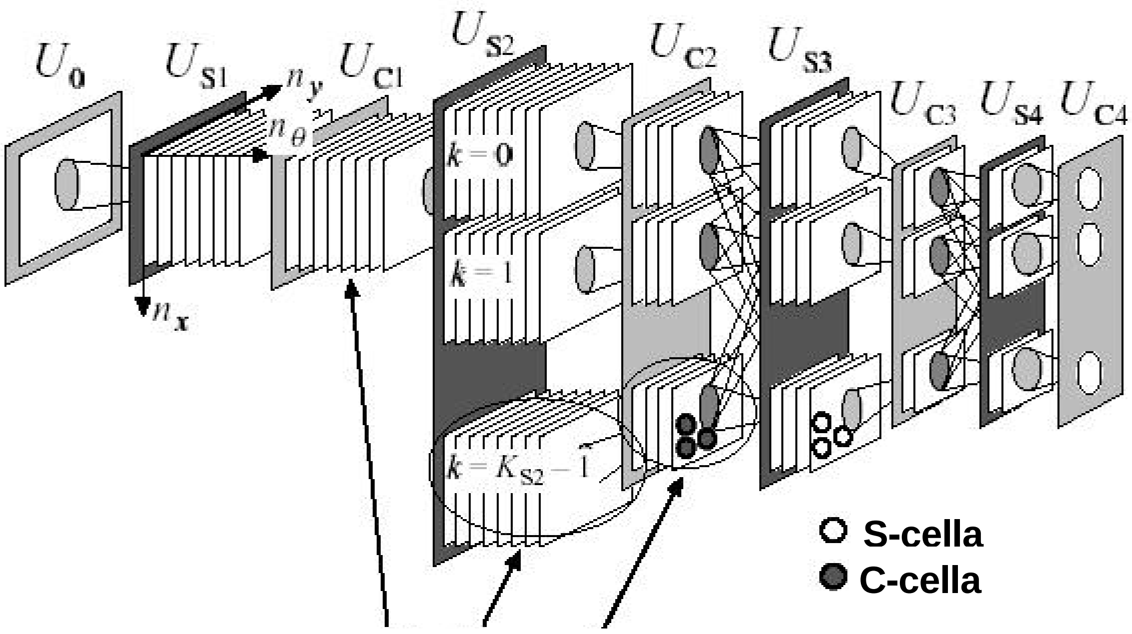
Fukushima: Neocognitron 1983

- Az eredeti modell (Cognitron, 1980) felügyelet nélküli, az újabb felügyelt tanulást valósít meg. 7, vagy 9 réteg, analóg típusú neuronok. Az eredeti modellben csak a maximális kimenetu neuronok erosíthették meg saját input kapcsolataikat. A cél: kézzel írt számok felismerése volt.
- Felépítés: hierarchikus szerkezet, 9 réteg, beleértve a retina réteget is. A kidolgozott háló nem volt érzékeny a karakter pozíciójára és méretére sem. A rendszer egyszerû cellákból álló rétegekbõl állt, ahol mindegyik réteg jellemzõfelismerõ, tulajdonságfelismerõ rendszerként mûködött.



Forrás: Satoh-Kuroiwa-Aso-Miyake: Recognition of rotated patterns using neocognitron

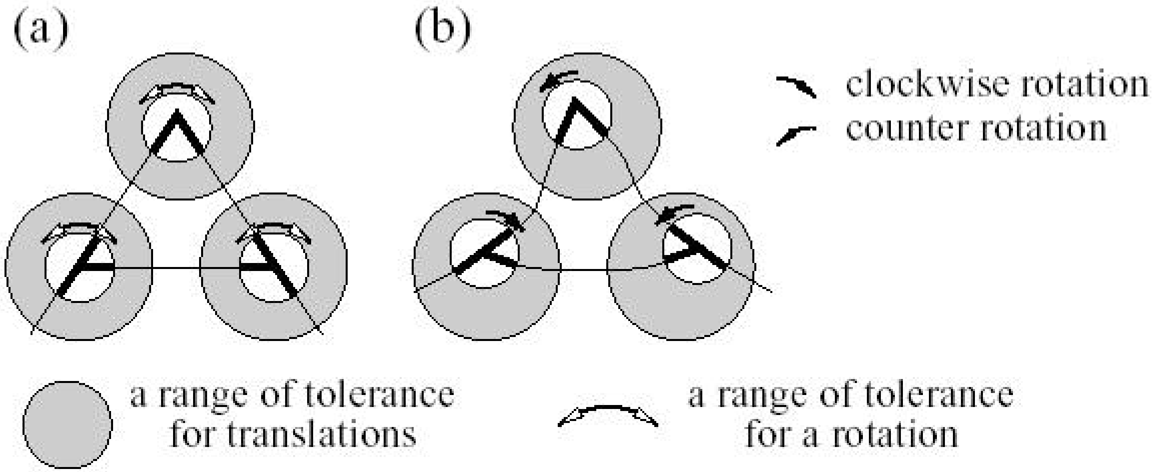
■ Satoh-Kuroiwa-Aso-Miyake: **Betûelforgatásra érzéketlen Neocognitron változat**



Forrás: Satoh-Kuroiwa-Aso-Miyake: Recognition of rotated patterns using neocognitron

- Satoh-Kuroiwa-Aso-Miyake: **Betûelforgatásra érzéketlen Neocognitron változat ..**

Az **A** betû jellemzõ betûrészleteinek felismerése érzéketlen az eltolódásra és az elfordulásra:

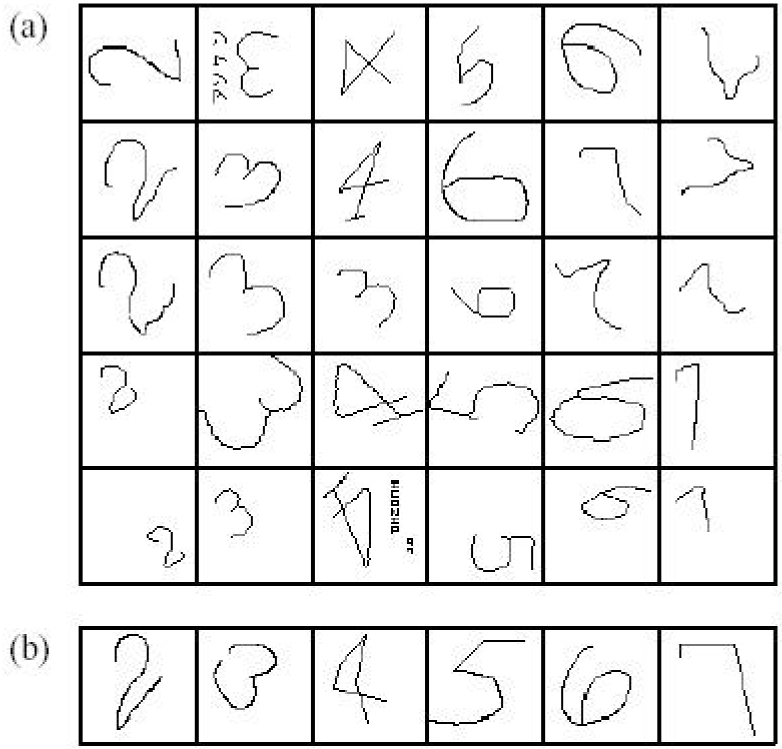


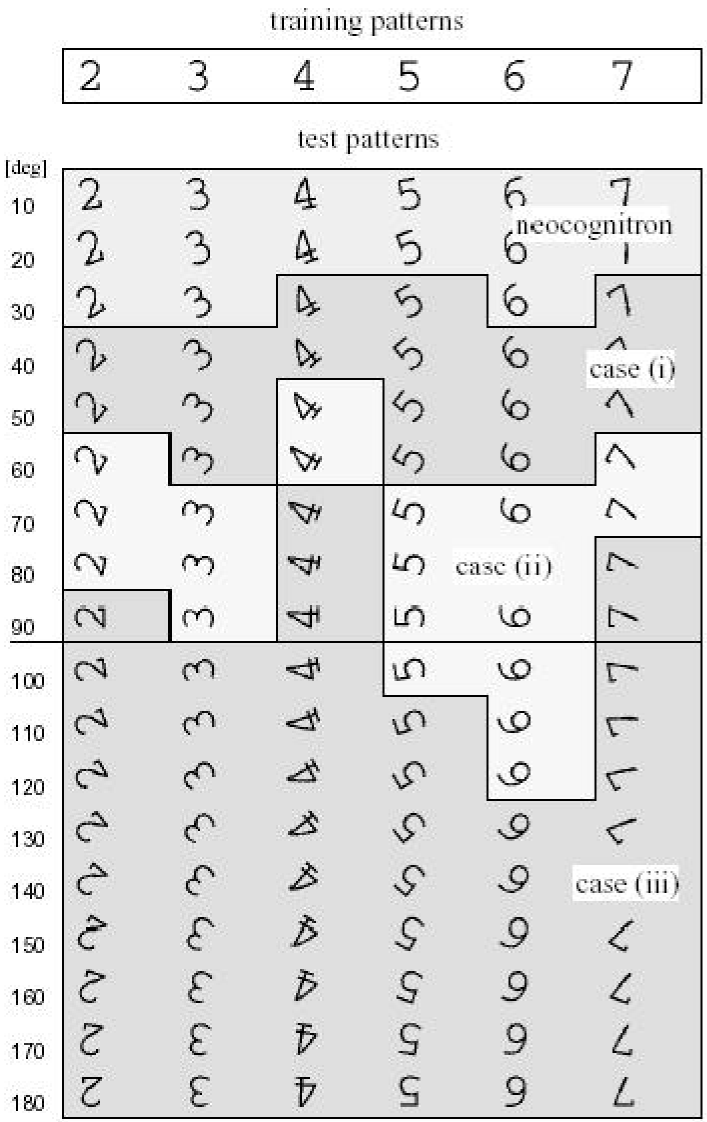
Forrás: Satoh-Kuroiwa-Aso-Miyake: Recognition of rotated patterns using neocognitron

M e s t e r s é g e s i n t e l l i g e n c i a

- Satoh-Kuroiwa-Aso-Miyake: **Betûelforgatásra érzéketlen Neocognitron változat ..**

A betanítéshoz és teszteléshez használt minták:
Alul: sikeresen felismert kézírásminták.



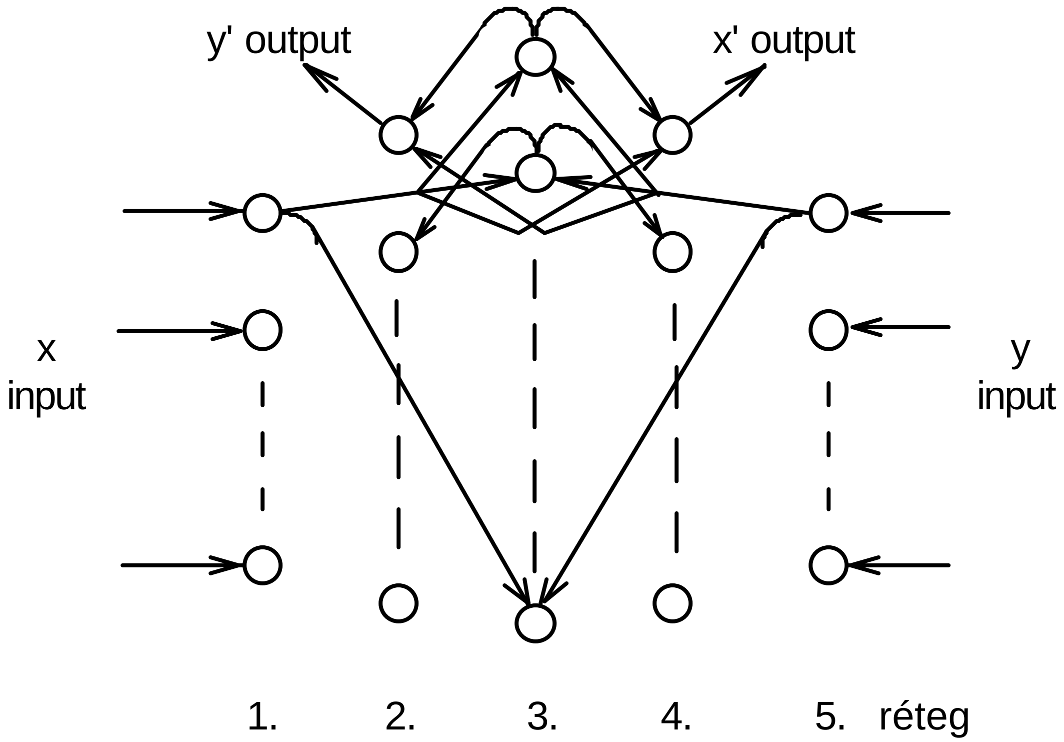


Forrás: Satoh-Kuroiwa-Aso-Miyake: Recognition of rotated patterns using neocognitron

dr.Dudás László

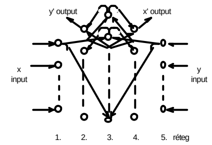
Hecht Nielsen: Szembeterjedéses háló (Counter propagation) Kohonen és Grossberg modulok egyesítése, ~1983

- A Kohonen és a Grossberg féle tanulási elv kombinálásával egy új típusú neurális háló keletkezett, mely 5 rétegbõl áll:



- 2., 4.: átlagtanuló Grossberg modulok, 3. Versengõ Kohonen réteg

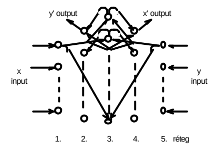
■ Hecht Nielsen: Szembeterjedéses háló ..



Mûködése:

- Inputjelek bemenete: 1. és 5. réteg.
- Az inputjelek átterjednek a 3., Kohonen réteg minden neuronjára. Az átterjedés szembe történik, innen ered az elnevezése (counterpropagation= szembeterjesztés). Valamint átterjednek a 4. és 5. réteg megfelelõ neuronjára.
- A 3. réteg minden egyes neuronja küld jelet a 2. és 4. réteg minden egyes neuronjának. Ezek a kimeneti rétegek jelentik a Grossberg modulokat. Ide a megfelelõ neuronokra közvetlenül is eljutnak az **x** és **y** jelek.
- A 3. Réteg neuronjai között versengés van. Az a neuron nyer (kimenete=1) amelynek a súlyai a legjobb megfelelést jelentik az x és y minták között. A többi kimenete 0. Csak a nyertes neuron tudja a súlyait beállítani, módosítani az input mintáknak megfelelõen a betanítás alatt. A 3. Réteg neuronjainak súlyai úgy állnak be, hogy optimális halmazát alkossák az x és y input minták közötti viszonyoknak, hasonlóságnak.
- A 2. És negyedik réteg neuronjai megtanulják az x és y bemenetek értékeinek azon átlagát, amelyek akkor adódnak, amikor a 3. Réteg minden egyes neuronja nyer a mintaközelségi versenyben. Ez az átlagtanuló struktúra a Grossberg találmány. (Outstar).

■ Hecht Nielsen: Szembeterjedéses háló ..



Mûködése ..:

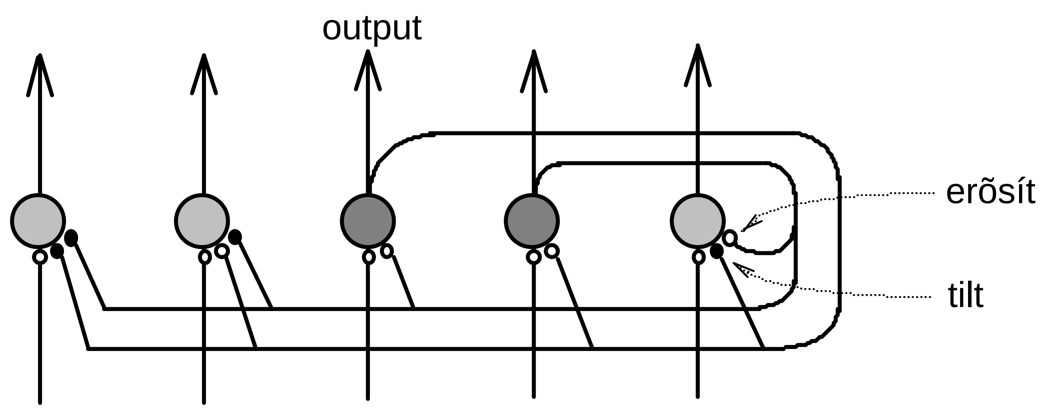
- Azután, hogy a rétegek elérték az egyensúlyt az **x** és **y** inputpár kiváltja egy olyan output mintapár kibocsátását, amelyek legjobban illeszkednek a 3. réteg súlymintázatpárjával.
- Ha valamelyik az **x** és **y** közül 0, a kimenet egy olyan pár lesz, amely a betanítottak közül a legjobban egyezik az ismert inputtal. Ha az input részei hiányoznak, akkor a háló kitölti a hiányt a legközelebbi mintával.
- A szembeterjedéses háló újdonsága volt, hogy létezõ hálótípusokból hozott létre új mûködést. Ezenfelül kétféle tanuló algoritmust is tartalmazott.
- A tanulás kétfázisú. Elõbb a Kohonen réteg súlyait állítja, majd a Grossberg rétegét.
  A tanulás konvergenciája elérhetõ.

Be	Ki
~**x**	**x’** = teljes **x**
~**y**	**y’** = teljes **y**

Be	Ki
~**x**	**x**’ = teljes **x**
y=0	**y**’ = teljes **y**, **x**-tõl eltérõ

dr.Dudás László

■ Kohonen: Önszervezo háló, 1988

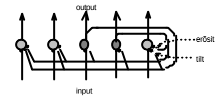


A kapcsolatok csak két neuron esetére vannak megrajzolva, de mindre hasonló

Jellemzõi:

- **Felügyelt tanítás nélküli**, nem igényel elvárt output mintákat!
- Alaprendszer: egy, vagy kétdimenziós, küszöb típusú egységekbõl álló tömb, **laterális kapcsolatokkal**.
- **Általánosító képesség**, az egyedekre vonatkozó extrém információk elvesztése nélkül.
- Mûködés: a rendszer úgy módosítsa önmagát, hogy az egymáshoz közeli neuronok hasonlóan válaszoljanak. A neuronok **versenyeznek** egymással, a „gyõztesé minden” módon. A gyõztes itt egy fizikailag közelálló **csoport**.
- A megismert eseménytér összefüggései **leképezõdnek** egy ugyanolyan topológiájú belsõ reprezentációra.

■ Kohonen: Önszervezo háló ..



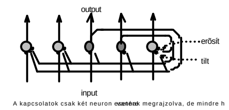
Jellemzõi..:

- Az önszervezõ hálók képesek felismerni az input adatokban meglévõ hasonlóságokat anélkül, hogy elõzetesen osztályokat jelölnénk ki. Fogalomalkotásra képesek!

A Kohonen-féle háló önszervezõ tulajdonság-térkép, leképezés

- Az agyban meglévõ funkcióspecifikus tagolódás kialakulását modellezi. Ezen régiókban a neuronok szervezõdése, topológiája megfelel az oda ingert küldõ érzékszervek topológiájának. Pl. különféle frekvenciájú hangok észlelésére a fonotopic map jött létre az agyban. Ezen területek lokalozált válasza az érzékszervektõl jövõ ingerekre, úgy tûnik, hogy egy speciális típusú laterális átcsatolással érhetõ el. (Mexikói kalap)
- Ezen egyrétegû háló idõfüggõ, idõben változó laterális kapcsolatokkal bír.
- A felügyelet nélküli (unsupervised) tanulást a neuronok belsõ adaptációs szabálya biztosítja. A tanulás a szomszéd neuronoknál is végbemegy, nem csak a válaszadó neuron, hanem a környezete is megnöveli a válaszát az adott input mintára. A szenzorikus szomszédok agyi szomszédok lesznek.

■ Kohonen: Önszervezo háló ..

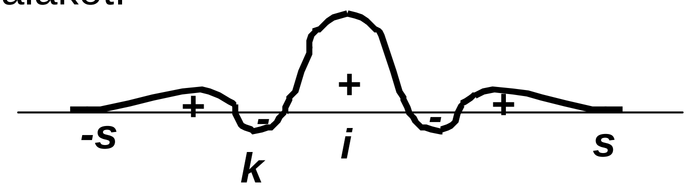


A Tulajdonság térképek kialakulása

- Az önszervezés kritikus pontja a lokális válasz létrehozása a laterális kapcsolatok által. Egy adott **x** input vektor esetén a nyertes neuron környezetére jutó átcsatolás erõsségét és elõjelét egy mexikói kalap alakkal jellemezhetjük. Matematikai alakja:

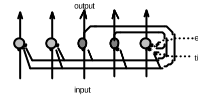
$$y_i(t) = r\left[x_i(t) + \sum_{k=-s}^{s} g_k y_{i+k}(t-1)\right]$$

- Az i. neuron kimenete yi függ a bemenetétõl xi és az s távolságú szomszédos neuronok elõzõ idõpontbeli kimenetétõl. ρ a szigmoid átviteli függvény, γk pedig a [-s;s] szomszédossági tartományban fejezi ki a laterális kapcsolatok erosségét, a mexikói kalap alakot.



- A háló klusterekbe (1D,2D), vagy buborékokba (3D) szervezi a neuronokat: a kezdeti aktivitáseloszlás többé-kevésbé véletlenszeru, de idovel klusterekbe, buborékokba csoportosul.
- A számítás lépései:
  - Az aktivitás maximumának megtalálása (nyertes)
  - A környezet neuronjainak megadása (megerõsítés).

■ Kohonen: Önszervezo háló ..



Tanulás

- Inger, input hatására a laterális kapcsolatoknak köszönhetõen a háló létrehozza a **klustert**, vagy **buborékot**, mely aktív. Azaz a legerõsebb aktivitással bíró neuron és környezõ neuronjai adnak pozitív kimenetet, miközben a többiek nulla kimenetûek. Az inputból eredõ aktiválás az u_c neuronnál:

$$x_c = \sum_j w_{jc} \cdot a_j$$

  azaz az inputok súlyozott összege.

- A **kluster átmérõje** függ a pozitív és negatív súlyok arányától. A kluster méret meghatározása után lezajlik a tanulás ezekben a neuronokban: minden beleesõ neuron közelebb viszi a súlyait egy bizonyos értékkel az A_k inputmintához. A kívülesõ neuronoknál nincs tanulás, változás.