

*Bevezetés a lágy számítás módszereibe*

***Neurális hálózatok***  
***Neurális hálózatok tanulása***

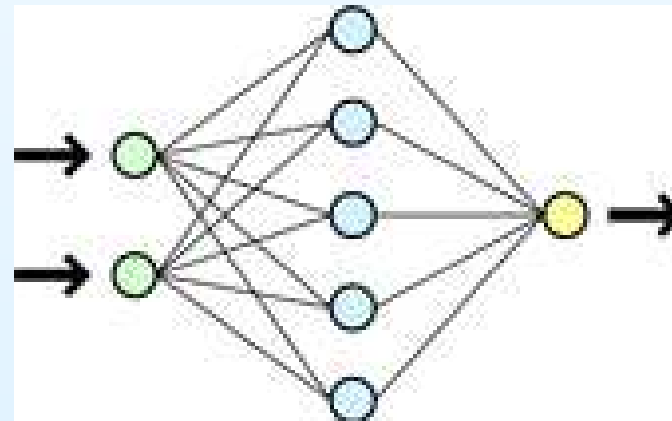
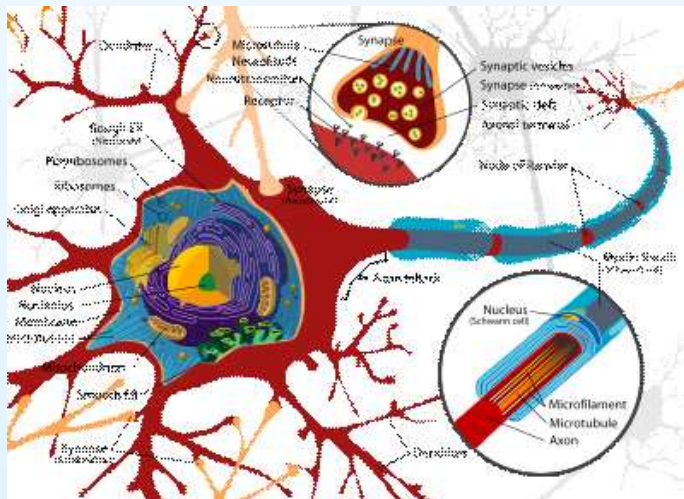
Werner Ágnes

*Villamosmérnöki és Információs Rendszerek Tanszék*

e-mail: [werner.agnes@virt.uni-pannon.hu](mailto:werner.agnes@virt.uni-pannon.hu)

# Neurális hálózat

- A neurális hálózat biológiai neuronok összekapcsolt csoportja.
- Hálózatok alapelve, hogy a számolásokat egymással összekapcsolt kis feldolgozóegységek, neuronok végzik.
- A neurális hálózat egyszerű egységekből áll, abban az értelemben, hogy belső állapotai leírhatók számokkal, ezek az aktivációs értékek.
- Mindegyik egység generál egy aktiválási értéktől függő kimeneti értéket (jelet).
- Minden egység kiküldi a kimeneti értékét az összes többi egységnek, amelyekkel kimenő kapcsolatban vannak.



# Neurális hálózatok tervezésének menete

---

- Reprezentatív tapasztalati adat gyűjtése (bemeneti adatok és (elvárt válaszok))
- Megfelelő feladat specifikus neurális paradigma kiválasztása
- Rendszer paraméterek (a neuronok átviteli függvényének, a neuronok számának, a tanítási módszernek és paramétereinek, valamint a kezdeti súlymátrix értékeinek) kiválasztása
- Teljesítmény mérő módszer kiválasztása
- A rendszer tanítása és tesztelése

# Tervezéskor felmerülő kérdések

---

## Neuron szinten:

- a neuron átviteli függvénye
- a bemenő jelek típusa

## Hálózat szinten:

- a hálózat topológiája
- rétegelt struktúránál a rétegek száma
- neuronok száma a különböző rétegekben
- neuronok típusa a különböző rétegekben

## Tanítási szinten

- tanító algoritmus
- tanítási paraméterek (pl.:  $\alpha$ ,  $\beta$ , ...)
- a tanítás megállásának feltételei

## Célszerű neurális hálózatot 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

# ***Visszacsatolt hálózat***

## Visszacsatolt hálózat

- Kimenete az idő függvénye
- Először bemeneti mintát helyezünk a hálózatra
- Eltávolítjuk az "indító" bemenetet
- Az inicializált kimenet a visszacsatolás révén kényszeríti az új módosított bemenetet a hálózatra
- Ez a módosított bemenet hozza létre a következő módosított kimenetet és így tovább
- Leképezés:  $\mathbf{o}(t + \Delta) = F[\mathbf{W}\mathbf{o}(t)]$
- $\Delta$  idő egységnyi:  $\mathbf{o}(k + 1) = F[\mathbf{W}\mathbf{o}(k)]$ 
  - $\mathbf{o}(1) = F[\mathbf{W}\mathbf{x}(0)]$
  - $\mathbf{o}(2) = F[\mathbf{W}F[\mathbf{W}\mathbf{x}(0)]]$
  - $\vdots$
  - $\mathbf{o}(k + 1) = F[\mathbf{W}F[\dots F[\mathbf{W}\mathbf{x}(0)]]]$

A hálózat állapot-változásainak a sorát írják le a  $k = 1, 2, \dots$  pillanatokban.

# Egyrétegű visszacsatolt hálózat

---

## Minta-asszociáció

### 1. Autoasszociáció

- a hálózatban a bemenetekre kapcsolt **minták egy készletét** tárolják
- később, ha a hálózatra olyan mintát helyeznek, amelyik hasonló a minták tárolt készletének egyik tagjához, az asszociálja a bemenetet a legközelebbi tárolt mintával

### 2. heteroasszociáció

- a hálózat összetartozó **mintapárokat** tárol
- a hálózatnak egy megadott - esetleg eltorzult, zajos - bemeneti mintához az ahhoz tartozó kimenetet kell válaszként előállítania



# Egyrétegű visszacsatolt hálózat

---

## Osztályozási feladatoknál

1. Ismert, hogy a bemeneti minták készlete **hány osztályba** van szétosztva
  - a hálózat egy adott bemeneti minta esetén válaszként meg kell, hogy adja a minta osztálytagságára vonatkozóan milyen információk érvényesek
  - az osztályokat **diszkrét értékű kimeneti vektorokkal** fejezik ki, az osztályozók kimeneti neuronjainak **aktivizációs függvényei binárisak**
2. Az osztályok száma nem ismert
  - csak a bemeneti minták állnak rendelkezésre
  - a hálózatnak kell megtalálnia az osztályok számát és a bemeneti minták besorolását azokba
  - a bemeneti mintatérben **klasztereket** alkotnak

# *Neurális hálózatok tanulása*

# Tanulás

---

- Az **ember** tanulásakor csak a kialakított viselkedés változását észleljük, annak lefolyását közvetlen módon nem.
- A **neurális hálózat** tanulása során a folyamat minden lépésében megfigyelhetjük az ok-okozat hatást.
- A **hálózat tanulásán** olyan folyamatot értünk, amely arra kényszerít egy hálózatot, hogy súlyait és/vagy felépítését úgy változtassa, hogy **egy bemeneti mintához egy kimeneti mintát szolgáltasson**.
- A tanulás csak akkor szükséges, ha a hálózat előzetes megtervezése, kialakítása nem lehetséges.

## Tanulási módok

---

- **Struktúra tanulás:** a neuronok, rétegek összeköttetési típusának ill. számának változása;
- **Paraméterek tanulása:** a súlyok, küszöbök változása;
- A tanulás befejeződött, ha a kívánt leképezés, a kívánt kimeneti vektorok reprodukálása kielégítő pontossággal megtörtént.

## Tanulás lefolyása

---

- A tanulás véghezvitele ismert mintapárok alapján az időben egymást követő **tanulási ciklusokkal** történik, miközben a minták prezentálásakor ezek korrekcióját a **szabad paraméterek módosításával** végzik.
- Ha a korrekciót minden lépésben, minden egyes tréning minta után elvégzik, akkor **on-line tanulásról** van szó.
- Az **off-line** (kötegetelt) **tanulásnál** több mintát a hálózatra helyeznek egymás után, és csak az összegzett hibajeleket használják a korrekcióra, tanulásra.

## Felügyelt tanulás

---

- Adott bemeneti mintákhoz tartozó helyes kimenetek rendelkezésre állnak vagy osztály besorolások ismertek.
- A helyes vagy célkimenet és az aktuális kimenet különbségét, a **hibajele**t valamilyen módon pl. negatív gradiens irányú változtatással csökkenti, felügyelten **minimalizálja**.

## Nem felügyelt tanulás

---

- Induláskor **nem ismerjük a helyes eredményeket** (a tanítót a tanulás során maga a hálózat alakítja ki).
- Mivel csak a bemeneteket ismerjük, azokból kell valamilyen módon a **tanulást megszervezni** (pl. a bemeneti mintákat valamilyen jellegzetességük szerint önkényesen osztályokba sorolja).
- Használat:
  - Hasonlóságok megállapítása a bemeneti minták között;
  - Csoportok, klaszterek kialakítása a bementi mintatérben;
  - Adattömörítés.

## Az általános tanulási szabály

A  $\mathbf{w}_i$  súlyvektor növekedése egy tanulási lépésben arányos az  $\mathbf{x}(t)$  bemeneti vektor és az  $r$  tanulási jel szorzatával:

$$\Delta \mathbf{w}_i(t) = \eta * r * \mathbf{x}(t)$$

ahol  $\eta$  egy pozitív szám, a **tanulási ráta** (tanulási állandó), amely meghatározza a tanulás mértékét.

Az  $r$  **tanulási (tanuló) jel** szabályonként más és általában  $\mathbf{w}_i$  és  $\mathbf{x}$ -nek valamilyen függvénye:

$$r := r(\mathbf{w}_i, \mathbf{x}, d_i)$$



## Az általános tanulási szabály felhasználása

- Az általános tanulási szabály felhasználásával többféle felügyelt és nem felügyelt tanulási szabály nyerhető a megfelelő tanuló jel alkalmazásával
- Ezzel az általános tanulási szabállyal a  $\mathbf{w}_i$  súlyvektornak a  $t$  időpontbeli tanuló lépés által létesített növekedése:

$$\Delta \mathbf{w}_i(t) = \eta * r * [\mathbf{w}_i(t), \mathbf{x}(t), d_i(t)] * \mathbf{x}(t)$$

- Azaz a  $t$  időpontban adaptált, módosított súlyvektor a következő  $(t + 1)$  időpontban vagy tanulási lépésben az alábbira módosul:

$$\mathbf{w}_i(t + 1) = \mathbf{w}_i(t) + \eta * r * [\mathbf{w}_i(t), \mathbf{x}(t), d_i(t)] * \mathbf{x}(t)$$

## Hebb tanulási szabály

- Nem felügyelt tanulás esetén használjuk. Ilyenkor nem állnak rendelkezésre az egyes bemenetekhez tartozó helyes kimenetek.
- A Hebb tanulás egyszerű alakjában a tanuló jel egyenlő a neuron kimenetével:

$$r := f(\mathbf{w}_i^t \mathbf{x}) = o_i$$

- Így a súlyvektor változás az általános tanulási szabály értelmében

$$\Delta \mathbf{w}_i = \eta f(\mathbf{w}_i^t \mathbf{x}) \mathbf{x} = \eta o_i \mathbf{x}$$

és a súlyvektor összetevőinek adaptációs szabálya a bemeneti réteg egy súlyára:

$$\Delta \mathbf{w}_{ij} = \eta o_i x_j \quad j = 1, \dots, n$$

Két réteg között, ahol a második réteg bemenete az első kimenete, így

$$\Delta \mathbf{w}_{ij} = \eta o_i o_j \text{ érvényes.}$$

## A perceptron tanulási szabály

- A tanuló jel a  $d_i$  kívánt és az  $o_i$  valóságos kimenet különbsége, vagyis a hiba:  $r := d_i - o_i$

- A tanulás tehát felügyelt.

- A bináris változatban az aktuális kimenet:

$$o_i = \text{sgn}(\mathbf{w}_i^t \mathbf{x})$$

és ezzel a súlyvektor módosítási szabálya:

$$\Delta \mathbf{w}_i = \eta [d_i - \text{sgn}(\mathbf{w}_i^t \mathbf{x})] \mathbf{x}$$

- Egy súly változtatása pedig:

$$\Delta \mathbf{w}_{ij} = \eta [d_i - \text{sgn}(\mathbf{w}_i^t \mathbf{x})] x_j \quad j = 1, \dots, n$$

## A perceptron tanulási szabály

- A bipoláris bináris aktivizációs függvénynek megfelelően a súlyok akkor és csak akkor kerülnek módosításra, ha az  $o_i$  aktuális kimenet nem helyes.
- Mivel a kívánt válasz csak 1 vagy  $-1$  lehet, a kifejezés az alábbi alakra egyszerűsödik:

$$\Delta \mathbf{w}_i = +(-)2\eta \mathbf{x}$$

A  $+$  előjel tartozik a  $d_i = 1$  és  $\text{sgn}(\mathbf{w}^t \mathbf{x}) = -1$ , a negatív a  $d_i = -1$  és  $\text{sgn}(\mathbf{w}^t \mathbf{x}) = 1$  értékhez, míg a  $d_i = \text{sgn}(\mathbf{w}^t \mathbf{x})$  esetében a formula nem használható (ha nincs hiba).

# ***Genetikus algoritmusok és neurális hálózatok integrálása***

# A XOR probléma 1.

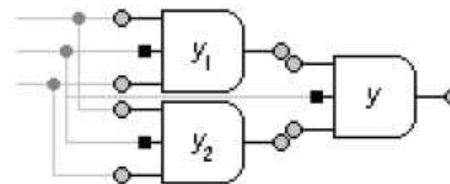
Az XOR probléma megoldásához minimálisan 5 darab perceptronra van szükség, amelyek közül 2 a rejtett rétegben helyezkedik el.

Ez azért van így, mert a kizáró vagy operátort lebontottuk részfeladatokra.

A XOR az  $y = (x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2) = (x_1 \vee x_2) \wedge \neg(x_1 \wedge x_2)$  összefüggéssel írható le.

Ha a  $y_1 = x_1 \vee x_2$  és  $y_2 = \neg(x_1 \wedge x_2)$  jelöléseket hajtjuk végre, akkor következik, hogy az előbbi egyenlőség a következőképpen néz ki:  $y = y_1 \wedge y_2$ . Az XOR igazságtáblázata:

$x_1$	$x_2$	XOR	$x_1 \wedge \neg x_2$	$\neg x_1 \wedge x_2$	$y_1 = x_1 \vee x_2$
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	0	0	0



A XOR problémát megoldó perceptronok összekapcsolása

## A XOR probléma 2.

- Genetikus algoritmussal egy neuronháló struktúráját kell megkeresni
- A legelterjedtebb neuronháló struktúra a XOR feladat számára: 3 rétegű, 5 neuron
- A szerkezetet irányított szomszédsági mátrix segítségével tudjuk megadni:

	1	2	3	4	5	súlyeltolás
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	1	1	0	0	0	1
4	1	1	0	0	0	1
5	0	0	1	1	0	1

Kromoszóma reprezentáció:

|000000|000000|110001|110001|001101|

Ezzel a reprezentációval különböző neuronhálózat struktúrákat tudunk létrehozni.