

Üzleti Intelligencia

7. Előadás: Mesterséges mélytanulás

Kuknyó Dániel
Budapesti Gazdasági Egyetem

2023/24
1.félév

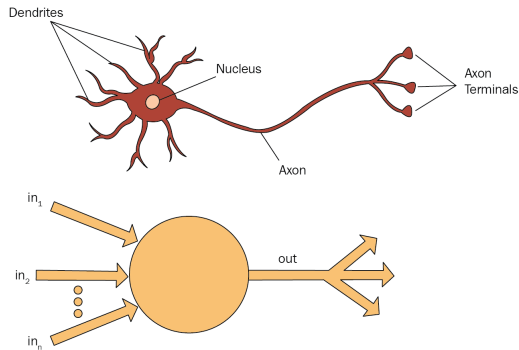
- 1 Bevezetés
- 2 Tanítás
- 3 Konvolúciós hálózatok
- 4 Mélytanulási problémák

- 1 Bevezetés
- 2 Tanítás
- 3 Konvolúciós hálózatok
- 4 Mélytanulási problémák

Biológiai és mesterséges neuronok

Az ember találmányait mindig a természet ihlette. A repülőgépeket a madarak mintájára, a gépkocsit a lovak inspirálták. A természetes lépés ezután az volt, hogy az emberi agyat is modellezzik.

Az első perceptron modellt Warren McCulloh és Walter Pitts hozta létre, először pedig Frank Rosenblatt épített egy perceptron gépet. Ez egy képfelismerő gép volt, 400 véletlenszerűen kapcsolt fotocella volt az érzékelője. A súlyokat potenciométerek implementálták, és a súlyok frissítését elektromotorok hajtották végre.



A neuron

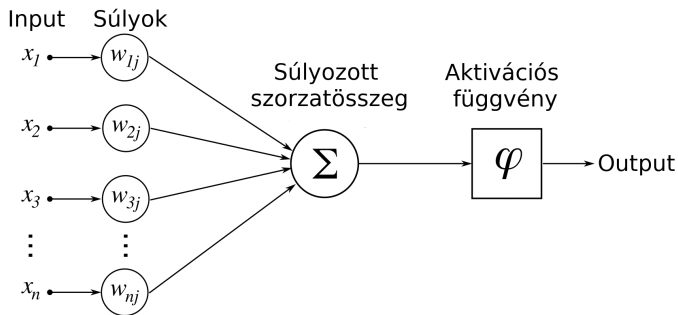
Az egyes neuronok egyszerű elemi műveleteket végeznek. A neuronnak több x inputja (kapcsolata) van, és mindegyikhez egy w **súly** tartozik.

A neuron kiszámítja az inputjainak a súlyozott összegét

$$z = x_1w_1 + x_2w_2 + \dots + x_nw_n$$

majd ezt az értéket behelyettesíti egy aktivációs függvénybe:

$$h = \varphi(z)$$

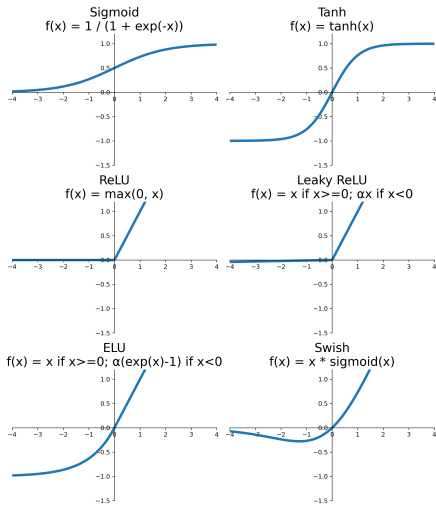


Gyakori aktivációs függvények

A neuron által kiszámolt súlyozott szorzatösszeg egy aktivációs függvénybe kerül behelyettesítésre.

A neurális hálózatok az aktivációs függvények segítségével **sajátítanak el komplex mintázatokat**. Az aktivációs függvény vezeti be a neurális hálózatokba a **nemlineáris transzformációt**, enélkül csak egy lineáris transzformáció lenne.

A különböző alkalmazásokra külön aktivációs függvények használatosak.



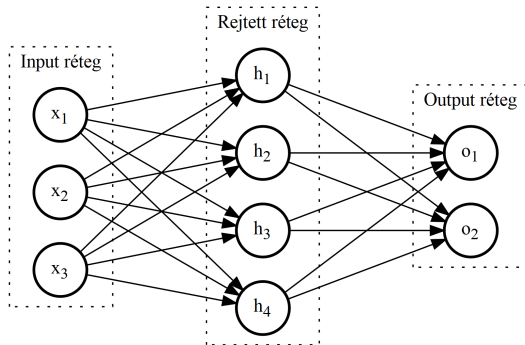
Többrétegű hálózatok

Ebben az esetben a neuronok rétegekben foglalnak helyet. A kapcsolataik az előző réteg kimeneteivel állnak összeköttetésben. A legelső réteg neuronjai a bemeneti adattal állnak összeköttetésben. Minden bemeneti jellemzőhöz egy neuron tartozik.

Teljesen becsatolt neuronréteg kimenete

$$h_{W,b}(X) = \varphi(XW + b)$$

- X : Input jellemzők mátrixa
- W : Kapcsolati súlyok mátrixa
- b : Torzítások vektora
- φ : Aktivációs függvény

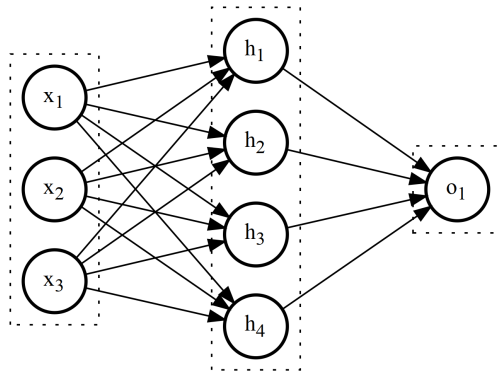


Hálózati architektúrák

A regressziós problémák esetén a neurális hálónak **egyetlen output neuronja** van.

A regresszió **tárgya egy folytonos** változó. Ebben az esetben a neuron output értéke a neurális hálózat predikciója a célváltozóra vonatkozóan.

Például: hány fok lesz holnap este?
- 35.

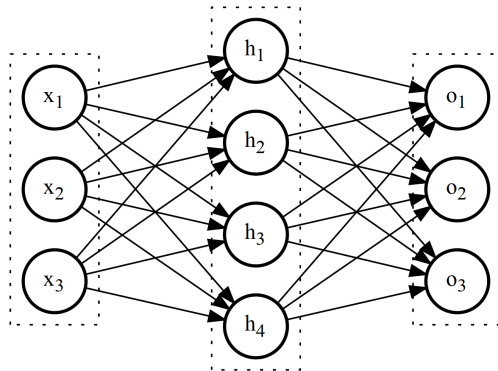


Hálózati architektúrák

Az osztályozási problémák tárgya **egy diszkrét változó**, amely különálló kategóriákra osztható.

Az osztályozó hálózatnak **annyi output neuronja van, ahány kategória lehetséges osztályozás esetén**. A predikció a mintaegyed adott osztályba esésének valószínűségét adja.

Például: meleg vagy hideg lesz az idő holnap este?
- Hideg.



1 Bevezetés

2 Tanítás

3 Konvolúciós hálózatok

4 Mélytanulási problémák

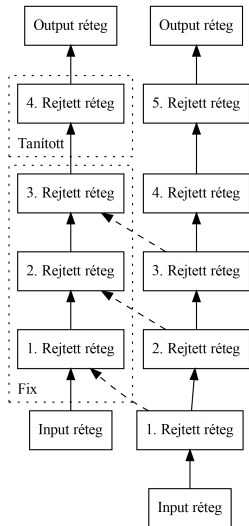
Hiba-visszaáramoltató algoritmus neurális hálózat súlyok tanítására

- ❶ **Inicializáció:** súlyok és torzítások kezdőértékeinek véletlenszerű megadása
- ❷ **Előreáramoltatás:** rétegek output értékének kiszámítása:
 - Rejtett réteg: $h_h = \varphi(W_h \cdot x + b_h)$
 - Output réteg: $h_o = \varphi(W_o \cdot h_h + b_o)$
- ❸ **Költség kiszámítása:** regresszió esetén pl.: $MSE = \sum (h_o - y)$
- ❹ **Visszaáramoltatás:**
 - Rejtett réteg hibája: $\delta_h = (W_o \cdot \delta_o) \odot \varphi'(z_h)$
 - Output réteg hibája: $\delta_o = (h_o - y) \odot \varphi'(z_o)$
- ❺ **Gradiens ereszkedés a súlyokon és torzításokon:**
 - Rejtett réteg: $W_h \leftarrow W_h - \alpha \cdot \delta_h \cdot x^\top$
 - Rejtett réteg: $b_h \leftarrow b_h - \alpha \cdot \delta_h$
 - Output réteg: $W_o \leftarrow W_o - \alpha \cdot \delta_o \cdot h_o^\top$
 - Output réteg: $b_o \leftarrow b_o - \alpha \cdot \delta_o$
- ❻ **Ismétlés a meghatározott lépésszámgig**

Transzfertanulás

Neurális hálózatok esetén lehetséges más neurális hálózatok **előretanított súlyainak felhasználása**. Ebben az esetben adott rétegek kimaradnak a tanításból.

Jellemzően az alsó rétegek lesznek rögzítettek, és a felső rétegek pedig tanítottak. Ez lehetőséget ad a hálózatnak **új mintázatokat megismerni hasonló feladatok** esetén.

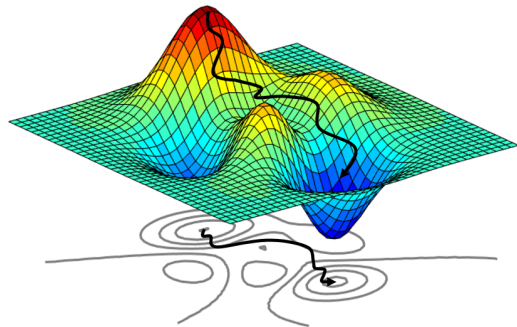


- 1 Bevezetés
- 2 Tanítás
- 3 Konvolúciós hálózatok**
- 4 Mélytanulási problémák

Az optimalizáló algoritmusok célja

A matematikában az optimalizációs algoritmusok melyeknek célja megtalálni a legjobb megoldást egy olyan problémára, ahol valamilyen korlátozásokkal kell **egy célfüggvényt minimalizálni**.

Mivel a neurális hálók által reprezentált nemlineáris függvények minimum helyei nem számíthatók ki zárt formájú függvénnyel, az optimalizáló algoritmusok iteratívak.



- 1 Bevezetés
- 2 Tanítás
- 3 Konvolúciós hálózatok
- 4 Mélytanulási problémák

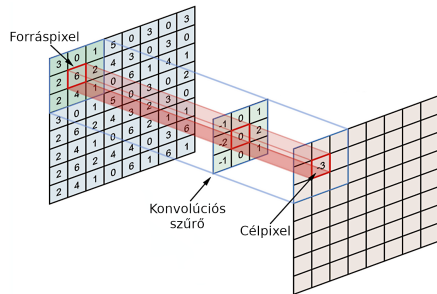
A konvolúció művelete

A konvolúció egy matematikai művelet, ami két függvényt kombinál egy harmadikká. Egy **jelfüggvényt** valamilyen **magfüggvénnyel** vegyít egy outputtá, ami megadja, hogy **a mag mennyiben befolyásolja a jelet**.

Konvolúció (2D, diszkrét)

A diszkrét konvolúció két 2D tömbön értelmezett, és jellemzők kivonatolására, szűrők alkalmazására használatos. Konvolúció f jelen és g magon, i, j képp koordinátákra:

$$(f * g)[i, j] = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} g[i - m, j - n] f[m, n]$$

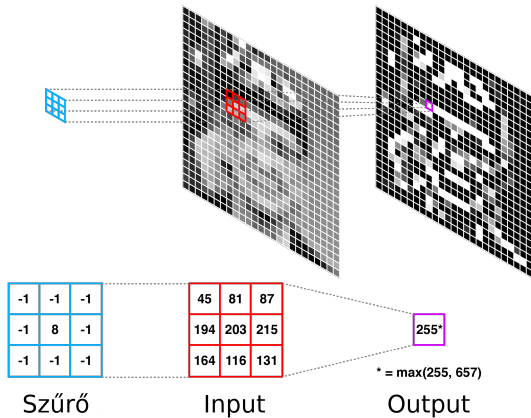


$$\begin{aligned} &(-1 \cdot 3) + (0 \cdot 0) + (1 \cdot 1) + \\ &(-2 \cdot 2) + (0 \cdot 6) + (2 \cdot 2) + \\ &(-1 \cdot 2) + (0 \cdot 4) + (1 \cdot 1) = -3 \end{aligned}$$

A konvolúciós réteg

A neurális hálózatok konvolúciós rétegeiben a **tanítható súlyok** az egyes **forráspixelekhez tartozó konvolúciós szűrőkben elhelyezkedő értékek**. Ez minden forráspixelhez egy egyedi szűrőt jelent, de lehetséges tetszőleges számú szűrő is.

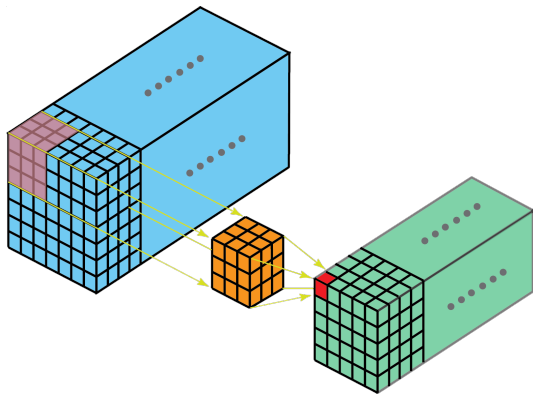
A konvolúciós magfüggvények szűrőként is alkalmazhatók. Például a képen egy élkereső szűrő látható, ami az outputra az input képen található élek másolódnak át.



Konvolúció több dimenzióban

A konvolúció általánosítható több dimenzióra is. A gyakorlatban 3D konvolúció megy végbe a neurális hálózatokban, hiszen a színes képek 3D mátrixokban tárolódnak el, ahol az egyes dimenziók a *magasság*, *hosszúság* és *csatorna*. A csatorna három elemű és az R,G,B értékeket kódolja.

A konvolúció tetszőleges dimenzióra általánosítható művelet.



Pooling réteg

A pooling vagy lazítás egy almintázási művelet, ami csökkenti a **bemenet térbeli dimenzióit**, miközben a fontos információt megtartja. A konvolúciós hálózatokban gyakran használatos a számítási igény csökkentésére, túltanulás elkerülésére és a hálózat segítésére abban, hogy fontos mintázatokot tanuljon meg.

Pooling neuron

A vele kapcsolatban álló **neuronok output értékeit aggregálja valamilyen függvény szerint**. Leggyakrabban ez az **átlagolás** vagy a **maximum** kiválasztás.

-2	-4	-4	5
14	-11	-2	10
14	11	-11	-3
13	14	-9	-9



14	10
14	-3

Max pooling

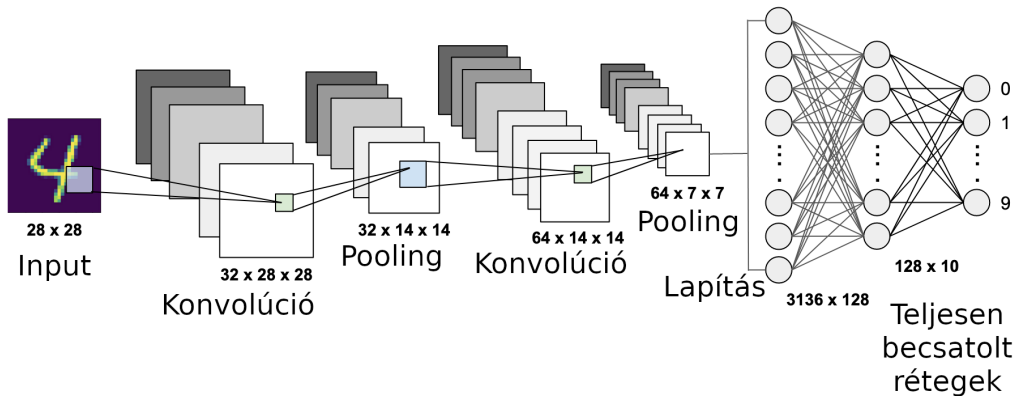


-1	2
13	-8

Avg pooling

$$\frac{1}{4}(-2 - 4 + 14 - 11) = -1$$
$$\max\{-2, -4, 14, -11\} = 14$$

Teljes konvolúciós architektúra



- 1 Bevezetés
- 2 Tanítás
- 3 Konvolúciós hálózatok
- 4 Mélytanulási problémák

Bináris osztályozás