

MESTERSÉGES NEURONHÁLÓ

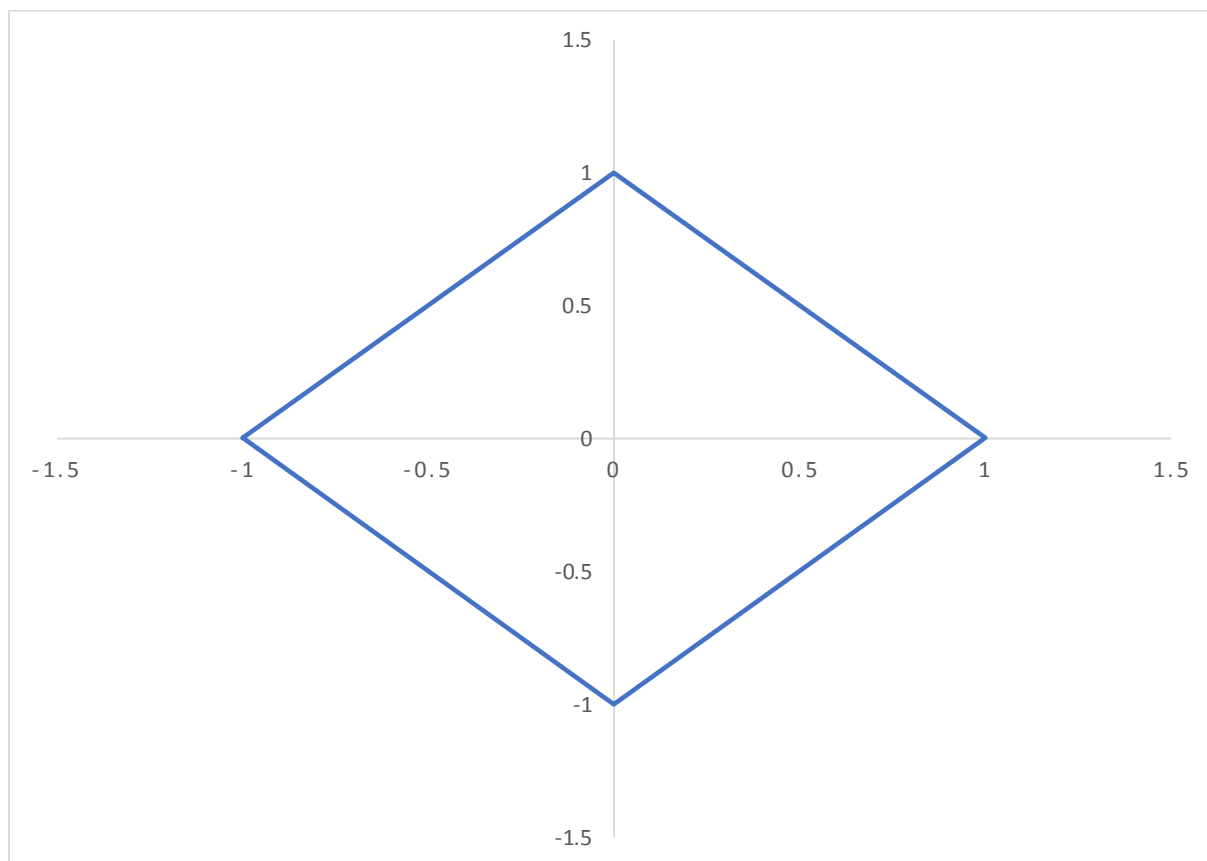
Beadandó

MESTERSÉGES INTELLIGENCIA

Burian Sándor - AWCYHE

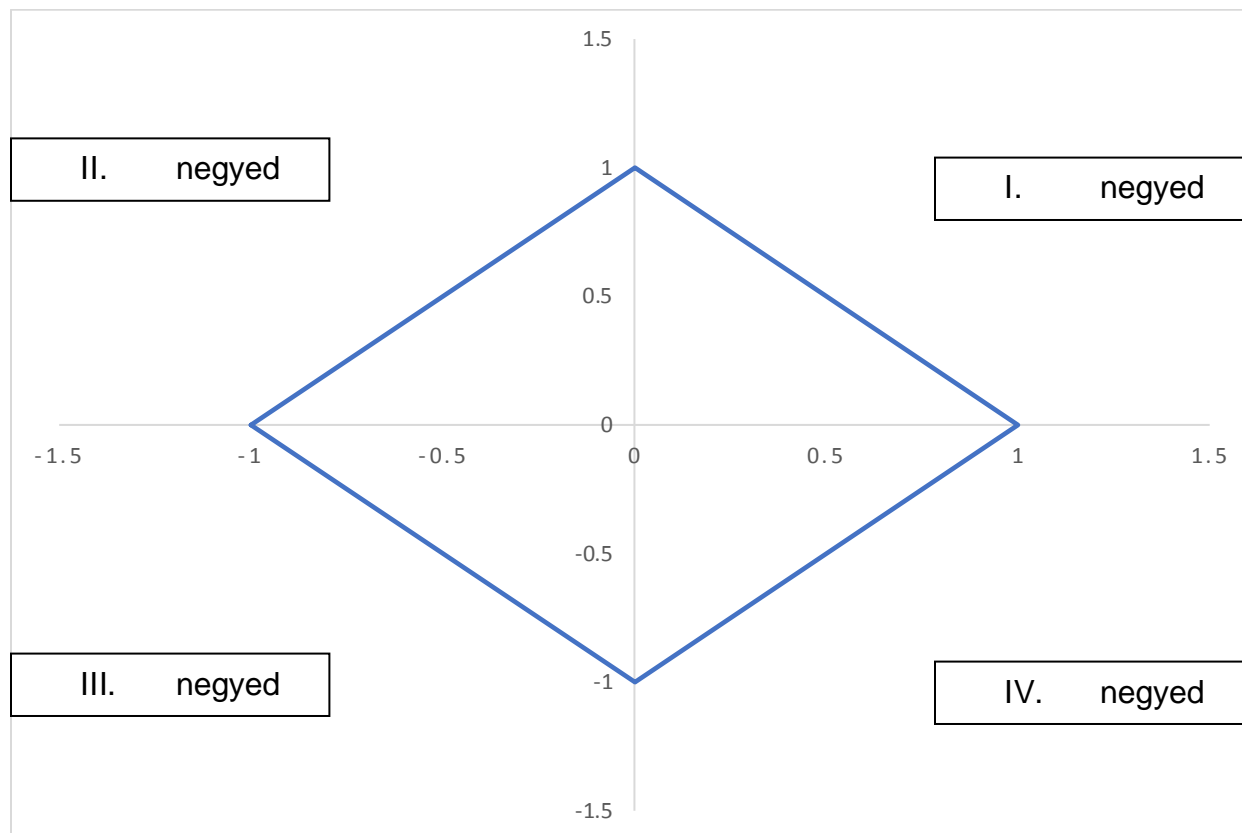
Feladat leírás

Tervezzen egy olyan neuronhálózatot, amely síkbeli koordináta rendszer pontjairól el tudja dönteni, hogy azok beleesnek-e azon origó középpontú rombuszba, amelynek átlói a koordináta tengelyekre esnek (origóban metszik egymást) és 2 egység hosszúak. Használjon Rosenblatt-féle perceptronokat. (A perceptronok súly-tényezőit "kézzel" kell kiszámolnia.)



Bevezető

A feladatot többretegű perceptronnal oldhatjuk meg. Vezessük be a következő jelölést:



Adott pont koordinátái (X_a, X_b) – a bemenet. Ez adott pont $X_a = X$, $X_b = Y$ koordinátája. Ezek súlyai rendre w_0, w_1, w_2 . Négy perceptronnal teszteljük. A négy számolási képlethez a bemenet a koordináták és a súlyok mindhez ugyanaz, a kimenet 0 vagy 1. Ezek meghatározásához ismerjük az egyenes egyenletét¹:

$n(n_1, n_2)$ normálvektorú $P(x_0, y_0)$ egyenes egyenlete: $n_1 \cdot x + n_2 \cdot y = n_1 \cdot x_0 + n_2 \cdot y_0$

ahol x, y az egyenes pontjainak koordinátái. Ezt a tanult képlettel összevetve²:

$$0 = w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 \text{ (továbbiakban [*])}$$

¹

https://hu.wikipedia.org/wiki/Koordin%C3%A1tageometria#A_norm%C3%A1lvektorb%C3%B3l_fel%C3%ADrhat%C3%B3_egyenes_egyenlete hozzáférés 2020 május 23

² <https://people.inf.elte.hu/gt/mi/12.neuronhalok.pdf#page=13> hozzáférés 2020 május 23

Az alábbi egyenletrendszert kapjuk:
$$\begin{cases} n_1 \cdot x + n_2 \cdot y - (n_1 \cdot x_0 + n_2 \cdot y_0) = 0 \\ w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 = 0 \end{cases}$$
 ahol a keresett súlyok megfeleltethetők a fenti képlethez az alábbiak szerint:

$$w_1 = n_1$$

$$w_2 = n_2$$

$$w_0 = -(n_1 \cdot x_0 + n_2 \cdot y_0)$$

A fentieket felhasználva az alábbi helyettesítési értékekkel dolgozva megkaphatjuk mindegyik negyedre (a rombusz oldalaira) a megfelelő értékeket:

- I. Akkor 1 a kimenet, ha az első negyedben húzott $(1,0)(0,1)$ egyenes alatt van a pont. *(a számolt érték ≤ 0)*. Ehhez ha az $n(0.5,0.5)^3$ és $P(1.5,-0.5)$ értékeket választjuk akkor $w_1 = 0.5$; $w_2 = 0.5$; $w_0 = -(0.5 \cdot 1.5 - 0.5 \cdot 0.5) = -0.5$ súlyokat kapjuk.
- II. Akkor 1 az output, ha a második negyedben húzott $(-1,0)(0,1)$ egyenes alatt van a pont. *(a számolt érték ≤ 0)* Ehhez az $n(-0.5,0.5)^4$ és $P(0.5,1.5)$ -re számított súlyok: $w_1 = -0.5$; $w_2 = 0.5$; $w_0 = -(0.5 \cdot 0.5 + 0.5 \cdot 1.5) = -0.5$ súlyokat kapjuk.
- III. Akkor 1 a kimenet, ha a harmadik negyedben húzott $(-1,0)(0,-1)$ egyenes fölött van a pont. *(a számolt érték ≤ 0)* Ehhez az $n(-0.5,-0.5)^5$ és $P(-1.5,1.5)$ -re számított súlyok: $w_1 = -0.5$; $w_2 = -0.5$; $w_0 = -(-0.5 \cdot (-1.5) + (-0.5) \cdot 0.5) = -0.5$ súlyokat kapjuk.
- IV. Akkor 1 az output, ha a negyedik negyedben húzott $(0,-1)(1,0)$ egyenes fölött van a pont. *(a számolt érték ≤ 0)* Ehhez az $n(0.5,-0.5)^6$ és $P(1.5,0.5)$ -re számított súlyok: $w_1 = 0.5$; $w_2 = -0.5$; $w_0 = -(0.5 \cdot 1.5 + (-0.5) \cdot 0.5) = -0.5$ súlyokat kapjuk.

A kimeneti jeleket (4 darab $\{0,1\}$) bemenetként kapja a következő (V.) perceptron ami azt teszteli van-e köztük hamis (0). Ha van akkor az adott bemeneti pont a rombuszon

³ A $(0.5,0.5)$ értékek azért fontosak mert a merőleges ami az origóból húzható az egyenesünkre itt metszi az egyenest, tehát ez az egyeneshez tartozó normálvektor.

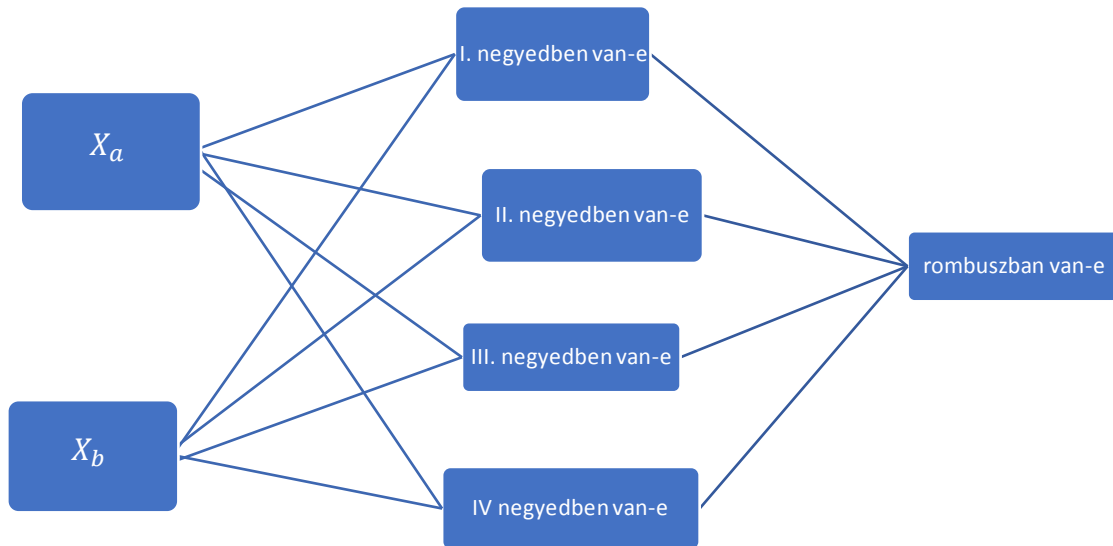
⁴ Az egyenes-hez tartozó origóból induló normálvektor metszéspontja az egyenssel hasonlatosan az előzőhöz.

⁵ Az egyenes-hez tartozó origóból induló normálvektor metszéspontja az egyenssel hasonlatosan az előzőhöz.

⁶ Az egyenes-hez tartozó origóból induló normálvektor metszéspontja az egyenssel hasonlatosan az előzőhöz.

kívülre esik. Ehhez a négy bemeneti értéket ($\{0,1\}$) összeszorozzuk. Ha az eredmény 0 akkor *hamis* ad, vagyis a pont a rombuszun kívülre esett.

A hálózat topológiája



A feladatot egy táblázattal tudjuk modellezni a legegyszerűbben, az alábbi módon:

Megjegyzés: A feladat megoldása során azokat a pontokat is megfelelőnek tekinthetjük, amelyek a rombusz oldalaira esnek, ez a "tanítás" folyamatát nem befolyásolja, csupán, egy konzekvencia kérdése.

bemenet			várt	súlyok			összegzett bemenet	szá- mított	hi ba	köv.sorhoz +			stim- mel	talá- lati a- rány
x0	x1	x2	y	w0	w1	w2	$I(x1,x2)=w0 + w1*x1 + w2*x2$	O	e	Dw0	Dw1	Dw2		
1	1	1	0	-0.5	0.5	0.5	0.5	1	-1	-0.1	-0.1	-0.1	0	
1	0.5	0.3	1	-0.6	0.4	0.4	-0.28	0	1	0.1	0.05	0.03	0	
1	0.7	0.9	0	-0.5	0.5	0.4	0.202	1	-1	-0.1	-0.1	-0.09	0	
1	0.9	0.8	0	-0.6	0.4	0.3	0.014	1	-1	-0.1	-0.1	-0.08	0	

Amiben az első három oszlop a bemenet: $x_0 - t$ az x_0 jelöli, aminek a kezdőértéke 1, $x_1 - t$ az x_1 jelöli, aminek a kezdőértéke 1, $x_2 - t$ az x_2 jelöli, aminek a kezdőértéke 1 - re a várt eredményt az y jelöli, a negyedik oszlopban, jelen esetben az értéke 0. A következő három oszlop az előzőekben kiszámított súlyok, jelen példarészletben az első negyed súlyai. Jelölésük $w_0 - nak$ w_0 , $w_1 - nek$ w_1 , $w_2 - nek$ w_2 . Kezdőértékük rendre: -0.5, 0.5, 0.5. A következő sorban az értékük a számítási képlet szerint változik, erről később részletesen.

A következő (8.) oszlop a számított összegzett bemenet a [*] képlet szerint a jelen táblázat beli jelölések alapján: $I(x_1, x_2) = w_0 + w_1 * x_1 + w_2 * x_2$ számított értéke a bemeneti adatok alapján 0,5.

A következő oszlop a számított output⁷ a Step függvény segítségével

$$\begin{cases} o = 1, & \text{ha az előző oszlop - beli elem } I(x_1, x_2) > 0 \\ o = 0, & \text{ha az előző oszlop - beli elem } I(x_1, x_2) \leq 0 \end{cases}$$

A következő oszlop a hibát tartalmazza, az előadás jelöléseivel $e = y - o$ képlet alapján.

A súlyok értékének meghatározásához az alábbi képlet⁸ szolgál segítségül: $\Delta w_i = \eta \cdot x_i \cdot (y - o)$ aminek megfelelően a táblázatban az alábbi jelöléseket használjuk:

$$\begin{aligned} \Delta w_0 &= Dw_0 = 0.1 * x_0 * e \\ \Delta w_1 &= Dw_1 = 0.1 * x_1 * e \\ \Delta w_2 &= Dw_2 = 0.1 * x_2 * e \end{aligned}$$

Így a kezdeti w_0 , w_1 , w_2 (a súlyok amik az egyenest határozták meg eredetileg) az alábbiak szerint módosulnak a Δw_i használatával:

$$\begin{aligned} w_0 &:= w_0 + Dw_0 \\ w_1 &:= w_1 + Dw_1 \\ w_2 &:= w_2 + Dw_2 \end{aligned}$$

Ehhez a táblázat következő sorában a w_0 , w_1 , w_2 oszlopot rendre az előző sorban levő adat és a Dw_0 , Dw_1 , Dw_2 oszlop előző sorában számított értékek összege tölti fel.

Hogy látványosabb legyen az eredmény, a következő, "stimme!" című oszlopba "TRUE" érték kerül, amennyiben a számított output azonos a várt eredménnyel, és "FALSE", hogyha nem. Ez gyakorlatilag a „hiba”⁹ oszlopban számított hiba lebutított kiírása, amikor nem érdekel, milyen irányban téved, csak maga a pusztta tény, hogy helyes-e a számítás, pusztán a látványosabb eredmény miatt.

Kezdetnek 25 pont koordinátáit adjuk meg, a hozzájuk tartozó várt eredménnyel, és a fentebb leírt képletekkel a megfelelő oszlopokban.

A táblázat "találási arány"¹⁰ oszlopában az "arány" szám a helyes válaszok százalékát mutatja, az utolsó méréstől következő adathalmazra számítva

⁷ <https://people.inf.elte.hu/gt/mi/12.neuronhalok.pdf#page=11> hozzáférés 2020 május 23

⁸ <https://people.inf.elte.hu/gt/mi/12.neuronhalok.pdf#page=12> hozzáférés 2020 május 23

⁹ táblázatban "J" oszlopként jelölt

¹⁰ táblázatban "O" oszlopként jelölt

$\frac{\text{helyes válaszok száma}}{\text{az összes minta száma}}^{11}$ – ezáltal követhető, hogy az ismételt tanítás során valóban tanul-e a gép.

Az első 25 adat összesített találati aránya 48%. Az előbb beadott adatokkal újra végezve a számításokat látványos, hogy a gép tanult, már 84% a jó válaszok aránya, még egyszer átfuttatva ugyanazokat az adatokat már 84%. Ezután újra beadva ugyanezeket a számokat 92%, majd 100%. Még egy ismétlés után már az új adatokra sem ad hibát.

A többi perceptort hasonlóan adhatjuk meg a táblázat további oszlopaiban, a fenti függvényeket használva rendre a megfelelő oszlopokban. Ahhoz, hogy a végén ne csupán a négy egyenes egyik avagy másik oldalára való kerülést ellenőrizzük ugyanazokat az adatokat kell megadnunk, ügyelve, hogy a várt eredmény mindig megfelelő legyen az adott egyenesre(perceptorra) nézve.

Ezt az utolsó perceptor követi ami megállapítja, hogy a pont hol helyezkedik el, a rombuszban vagy azon kívül?

A perceptorok az alábbi százalékban adnak helyes válszokat etaponként:

	I. perceptor	II.perceptor	III. perceptor	IV.perceptor	V.perceptor
1. etap	48	44	72	56	72
2. etap	84	64	96	88	84
3. etap	84	64	100	100	92
4. etap	92	84	100	100	92
5. etap	100	80	100	100	100
6. etap	100	76	100	100	100
Új adatok	100	100	96 ¹²	100	100

¹¹ Csak etaponként, tanításonként, nem a teljes adatbázisra értendő!

¹² False negatív érték miatt, elfogadható az arány, a végső v. perceptornál

A csatolt táblázatok szín magyarázat:

mindegyik perceptrónál a várt eredmény oszlopa	első etap
II. Perceptor	második etap
III. Perceptor	harmadik etap
IV. Perceptor	negyedik etap
V. Perceptor	ötödik etap
	hatodik etap
	új adat

A táblázat még számos segéd sort tartalmaz, mint ahogy azt fentebb láttuk. A bementi adatoktól nagyban függ a módszer gyorsasága, ha megfelelő adatokat választunk akár jelentősen kevesebb futásból(etapból) is elérhető lehet ez az eredmény.