

Časť A – Implementácia logických funkcií AND a OR pre dvojrozmerný vstup pomocou binárneho perceptrónu

Simuláciou sme ukázali, že pomocou binárneho perceptrónu je možné realizovať logické funkcie AND a OR pre dvojrozmerný vstup.

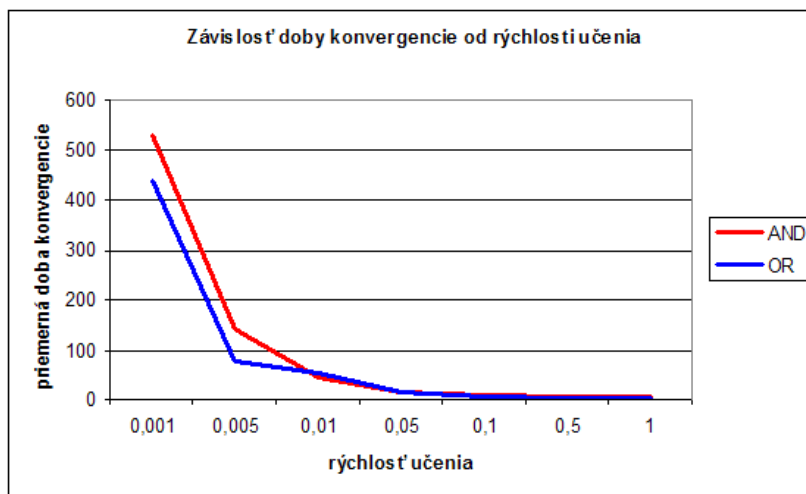
a. Vplyv rýchlosti učenia na dobu konvergenencie

Dobu konvergenencie sme testovali na oboch funkciách pre α z množiny $\{0,001; 0,005; 0,01; 0,05; 0,1; 0,5; 1\}$. Pre každú hodnotu α sme učenie spustili 15 krát (vždy s inak inicializovanými váhami na začiatku) a sledovali dobu konvergenencie. Nakoľko sa vo výsledkoch vyskytovali anomálie (napr. vyskytla sa situácia, kedy sa váhy na začiatku inicializovali tak, že perceptrón nerobil žiadne chyby a doba konvergenencie bola 1), do priemeru sme nebrali 3 najkratšie a 3 najdlhšie doby konvergenencie.

Zistili sme, že pokiaľ je rýchlosť učenia malá ($< 0,01$), doba konvergenencie klesá s rastúcou rýchlosťou učenia. Pri väčšej rýchlosti (blízkej 1) učenia však už na rýchlosť konvergenencie nemá pozorovateľný vplyv - dobu konvergenencie viac ovplyvňuje náhoda (ako sa inicializujú počiatočné váhy).

α	AND	OR
0,001	529,0	438,3
0,005	141,6	76,7
0,01	42,8	52,6
0,05	14,4	11,9
0,1	8,6	6,1
0,5	5,4	3,0
1	6,9	3,6

tabuľka 1: priemerná doba konvergenencie v závislosti od rýchlosti učenia



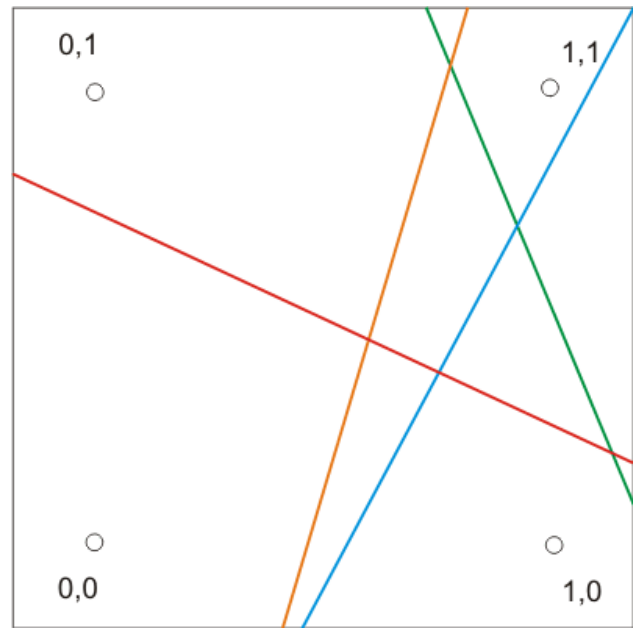
graf 1: závislosť doby konvergenencie od rýchlosti učenia

b. Možný rozsah rýchlosti učenia α

Aby učiace pravidlo vôbec fungovalo, musí platiť $\alpha > 0$. Pokiaľ by sa $\alpha = 0$ - váhy by sa vôbec nemenili, pokiaľ by $\alpha < 0$, k cieľovým váham by sme sa nepribližovali ale sa od nich vzdľávali. Simuláciami sme zistili, že rýchlosť učenia α môže byť nastavená aj nezmyselne vysoká (testované pre „extrémne“ $\alpha = 10^9$) a binárny perceptrón je stále schopný sa úlohu naučiť.

c. Priebeh učenia logického AND

ep.	x1	x2	w1	w2	w3	y	d	net	err
1	0	0	0,3121	-0,4602	-0,5899	1	0	0,5899	0,5
	0	1	0,3121	-0,4602	-0,0899	0	0	-0,3703	0,5
	1	0	0,3121	-0,4602	-0,0899	1	0	0,4020	1
	1	1	-0,1879	-0,4602	0,4101	0	1	-1,0582	1,5
2	0	0	0,3121	0,0398	-0,0899	1	0	0,0899	0,5
	0	1	0,3121	0,0398	0,4101	0	0	-0,3703	0,5
	1	0	0,3121	0,0398	0,4101	0	0	-0,0980	0,5
	1	1	0,3121	0,0398	0,4101	0	1	-0,0582	1
3	0	0	0,8121	0,5398	-0,0899	1	0	0,0899	0,5
	0	1	0,8121	0,5398	0,4101	1	0	0,1297	1
	1	0	0,8121	0,0398	0,9101	0	0	-0,0980	1
	1	1	0,8121	0,0398	0,9101	0	1	-0,0582	1,5
4	0	0	1,3121	0,5398	0,4101	0	0	-0,4101	0
	0	1	1,3121	0,5398	0,4101	1	0	0,1297	0,5
	1	0	1,3121	0,0398	0,9101	1	0	0,4020	1
	1	1	0,8121	0,0398	1,4101	0	1	-0,5582	1,5
5	0	0	1,3121	0,5398	0,9101	0	0	-0,9101	0
	0	1	1,3121	0,5398	0,9101	0	0	-0,3703	0
	1	0	1,3121	0,5398	0,9101	1	0	0,4020	0,5
	1	1	0,8121	0,5398	1,4101	0	1	-0,0582	1
6	0	0	1,3121	1,0398	0,9101	0	0	-0,9101	0
	0	1	1,3121	1,0398	0,9101	1	0	0,1297	0,5
	1	0	1,3121	0,5398	1,4101	0	0	-0,0980	0,5
	1	1	1,3121	0,5398	1,4101	1	1	0,4418	0,5
7	0	0	1,3121	0,5398	1,4101	0	0	-1,4101	0
	0	1	1,3121	0,5398	1,4101	0	0	-0,8703	0
	1	0	1,3121	0,5398	1,4101	0	0	-0,0980	0
	1	1	1,3121	0,5398	1,4101	1	1	0,4418	0



tabuľka 2: stav siete pri učení sa logického AND (rýchlosť učenia $\alpha=0.5$)

graf 2: vývoj hraničnej priamky po epochách
červená – epocha 1, oranžová – epocha 3, modrá – epocha 5, zelená – epocha 7 (cieľový stav)

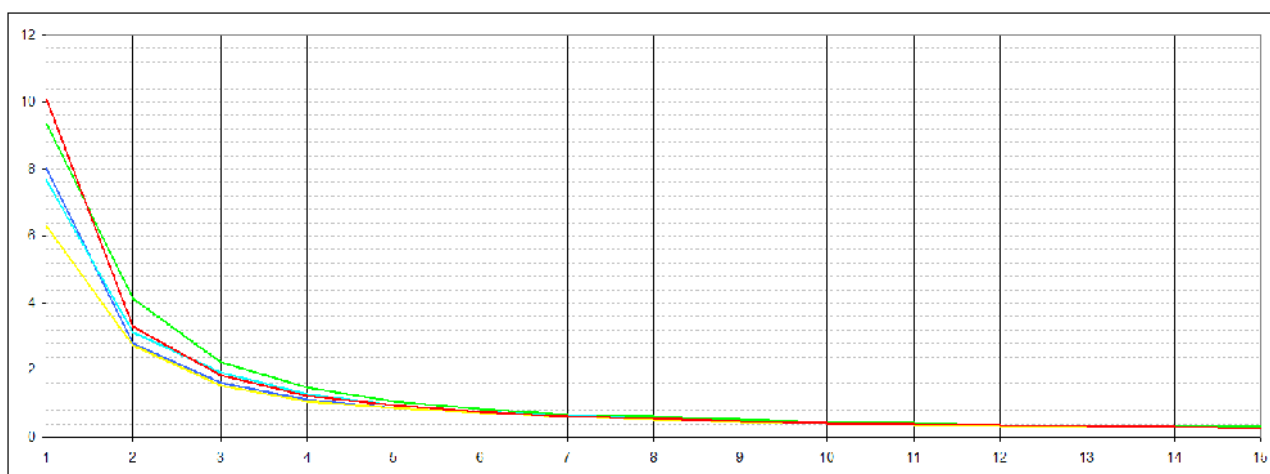
d. Použitie spojitého perceptrónu na túto úlohu

Na riešenie tejto úlohy by bolo možné použiť aj spojitý perceptrón. Hlavné rozdiely by spočívali v tom, že výstup by sme nedostávali z množiny $\{0; 1\}$ ale z intervalu $\langle 0; 1 \rangle$. Takisto perceptrón by sme nemohli učiť do okamihu, kým chyba dosiahne 0, ale do okamihu, kedy chyba klesne pod nejakú nami stanovenú toleranciu. Zaujímavý je tiež fakt, že spojitý perceptrón nie je „odolný“ voči nezmyselne vysokým rýchlostiam učenia ako binárny perceptrón – a pre rýchlosti učenia $\gg 1$ nie je schopný sa túto úlohu naučiť.

Časť B – spojitý perceptrón ako separátor dvoch tried obrazcov

Simuláciou sme overili, že spojitý perceptrón je možné použiť ako separátor dvoch tried kompaktných obrazcov.

Perceptrón sme testovali s rýchlosťou učenia $\alpha = 0,8$ a zastavovacou podmienkou chyba $< 0,1$. Simuláciu sme spustili 5 krát – priemerná doba konvergence bola 38 epoch. Graf 3 znázorňuje vývoj chyby v jednotlivých simuláciách (pre prehľadnosť je zobrazených len prvých 15 epoch). Je vidieť, že chyby sa výrazne líšia len v niekoľkých prvých epochách, čo je zjavne spôsobené náhodným inicializovaním váh. Od cca 7 epochy bola chyba v jednotlivých simuláciách prakticky rovnaká.



graf 3: vývoj chyby pre 5 simulácií s rovnakými parametrami

Zistili sme, že perceptrón nebol 100% úspešný. V troch simuláciách nedokázal zaradiť správne jeden (vždy ten istý) obrazec z testovacej množiny. Za úspešné zaradenie sme považovali, ak pre obrazec s požadovaným výstupom 0 vrátil číslo v intervale $<0; 0,1>$ a pre obrazec s požadovaným výstupom 1 vrátil číslo v intervale $<0,9; 1>$:

Nesprávne zaradený obrazec:		výsledok:	správnosť klasif.:
0 0 1 1 0	simulácia 1:	0,901223 (OK)	100%
0 0 1 1 0	simulácia 2:	0,896838	99%
0 0 1 1 0	simulácia 3:	0,919558 (OK)	100%
0 0 0 0 0	simulácia 4:	0,897208	99%
0 0 0 0 0	simulácia 5:	0,898591	99%
požadovaný výstup: 1			