

# 1. Semestrální práce, 2013

## NEU

---

NEURONOVÁ SÍŤ

MARKÉTA JEDLIČKOVÁ

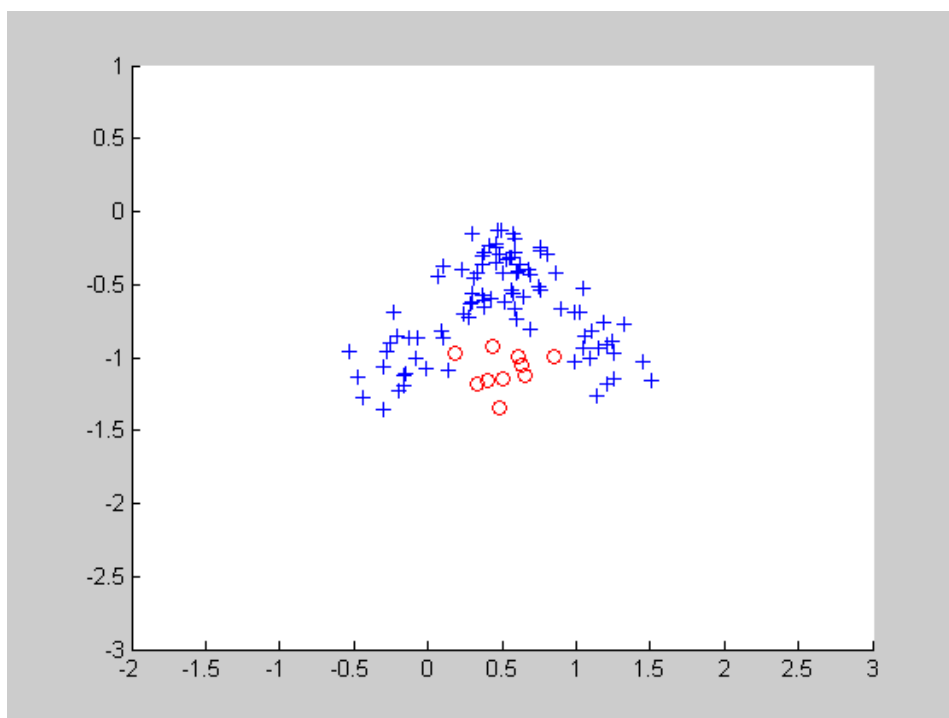
25. PROSINCE 2013



KATEDRA KYBERNETIKY  
ZÁPADOČESKÁ UNIVERZITA V PLZNI

## NEU - Zadání semestrální práce č.1

1. Pro trénovací množinu znázorněnou na obrázku<sup>1</sup> navrhnete takovou neuronovou síť, která po natrénování dokáže oddělit červené body od modrých.



2. Vypracujte referát, kde uvedete, s jakými problémy jste se při řešení úlohy setkali a jak jste je řešili. Konkrétně tedy vždy uveďte, jak jste při trénování měli nastavené nastavitelné parametry sítě a jak se v takovém případě síť chovala. Pokud se Vám síť s navrženými parametry nepodařilo natrénovat, uveďte, co jste změnili a proč a jakých výsledků jste se změněnými parametry dosáhli.
3. Funkční program i referát odevzdejte vyučujícímu.

Poznámka: V případě zájmu si např. pomocí skriptů `generuj_trenovaci_mnozину.m` a `generuj_shluk.m` můžete vygenerovat ještě jinou další trénovací množinu a pokusit se pro ni natrénovat vhodnou neuronovou síť.

25.10.2013, Vlasta Radová

---

<sup>1</sup> Trénovací množina je uložena v textovém souboru tak, že každá řádka odpovídá jedné trénovací dvojici. Na každém řádku jsou 3 položky, které jsou od sebe odděleny mezerou. První dvě položky jsou vstupy neuronové sítě, třetí položka je odpovídající požadovaný výstup sítě.

Příklad výpisu trénovací množiny ( $x_1$ ,  $x_2$  a  $u$  je uvedeno pouze pro názornost, v textovém souboru není.):

$x_1$	$x_2$	$u$
6.4408467155225013e-001	-5.8024135339391125e-001	1.0000000000000000e+000
4.5475041308657149e-001	-2.4644945484678327e-001	1.0000000000000000e+000
4.0818292046910754e-001	-2.3662677299640716e-001	1.0000000000000000e+000
5.0025222696548344e-001	-1.1492978773866835e+000	-1.0000000000000000e+000
4.4059138143769083e-001	-9.2549117329521979e-001	-1.0000000000000000e+000
6.8270827428183134e-001	-3.9871744069688964e-001	1.0000000000000000e+000
4.2414962804646078e-001	-5.9217085479074094e-001	1.0000000000000000e+000

---

# Obsah

---

<b>1</b>	<b>Řešení</b>	<b>1</b>
1.1	Volba sítě . . . . .	1
1.2	Trénování sítě . . . . .	2
1.3	Natrénovaná neuronová síť . . . . .	4
<b>2</b>	<b>Závěr</b>	<b>6</b>

---

# Seznam obrázků

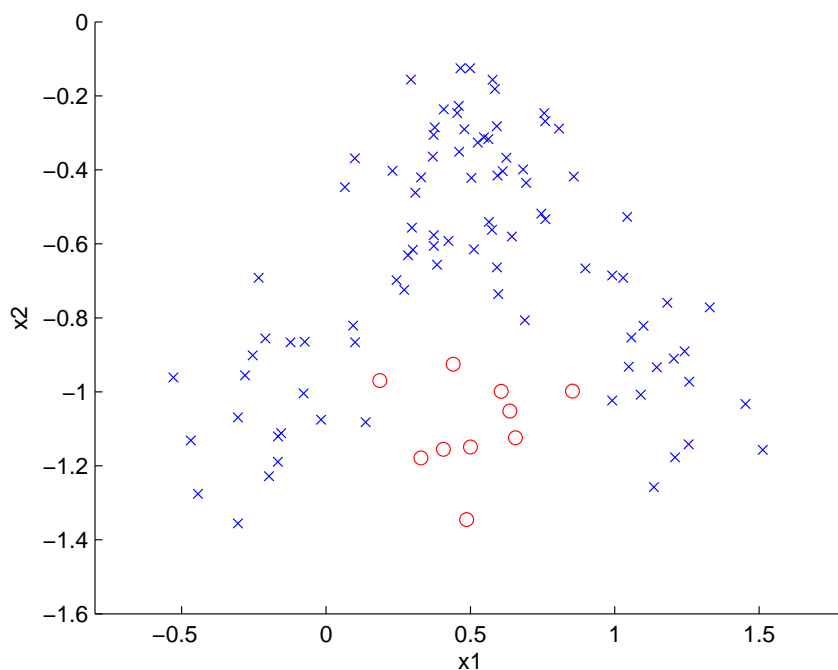
---

1.1	Trénovací množina . . . . .	1
1.2	Schéma neuronové sítě . . . . .	2
1.3	Chyba trénování . . . . .	3
1.4	Porovnání výstupu $u$ a $y$ . . . . .	4
1.5	Výsledné rozdělení shluků . . . . .	5

## Řešení

### 1.1 Volba sítě

Pro natrénování dané množiny dat je potřeba k oddělení shluků použít alespoň dvouvrstvou neuronovou síť. Síť má ze zadání jeden výstup. Trénovací množinu dat je možné vidět na obr. 1.1



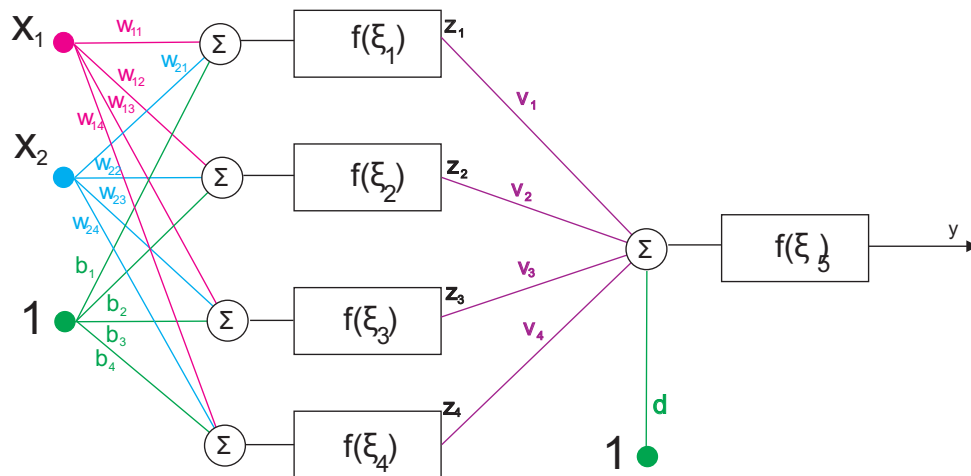
Obrázek 1.1: Trénovací množina

Požadovaný výstup obsahuje pouze hodnoty 1 a  $-1$ , byla tedy zvolena bipolární spojitá aktivační funkce. Známe i její derivaci, kterou využijeme v algoritmu zpětné propagace. Bipolární funkce a její derivace jsou definovány jako:

$$f(\xi) = \frac{2}{1 + e^{\lambda \xi}} - 1 \quad (1.1)$$

$$f'(\xi) = \frac{\lambda}{2}(1 - f^2) \quad (1.2)$$

Zvolenou strukturu neuronové sítě je možné vidět na obr. 1.2, kde  $w_{ij}, v_i$  jsou váhy jednotlivých vrstev a  $b_i, d$  jsou prahy jednotlivých vrstev. Vstupem neuronové sítě je  $x_1$  a  $x_2$ , výstupem první vrstvy  $z_i$  a výstupem druhé vrstvy  $y$ .



Obrázek 1.2: Schéma neuronové sítě

## 1.2 Trénování sítě

Pro trénování neuronové sítě je zvolena metoda zpětné propagace, která byla probrána na přednáškách. Tato metoda je vhodná pro učení nelineárních neuronových sítí za pomoci trénovací množiny.

Pro algoritmizaci metody v programovém prostředí Matlab bylo nejdůležitější otázkou správné nastavení parametrů pro učení sítě. V algoritmu se setkáme s těmito parametry:

- $c$  - *konstanta učení* - malá hodnota zpomaluje proces učení, pro velkou hodnotu hrozí nebezpečí nestability procesu učení
- $\lambda$  - *konstanta strmosti* - obdobně jako u konstanty učení

Testování bylo prováděno pro tři neurony obsažené ve skryté vrstvě. Zaznamenávána byla celková chyba v posledním trénovacím cyklu. Výpočet této chyby je možné vidět v rovnici 1.3. Kde  $k$  je počet vzorků v jednom trénovacím cyklu,  $u$  je požadovaný výstup a  $y$  je výstup z modelu neuronové sítě. Požadavkem bylo aby se tato chyba s přibývajícími cykly snižovala.

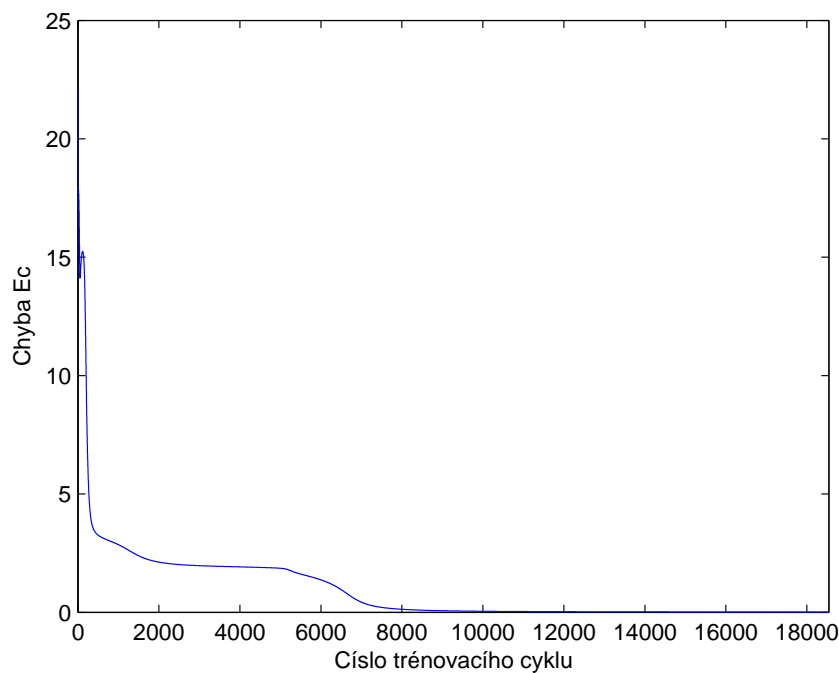
$$E_k = E_{k-1} + \frac{1}{2}(u_k - y_k)'(u_k - y_k) \quad (1.3)$$

Pro nastavení správné hodnoty těchto parametrů byl nejdříve zvolen menší počet trénovacích cyklů ( $p = 500$ ). Pro tento počet cyklů bylo prováděno testování parametrů. Výsledky testu můžeme pozorovat v tabulce níže.

$\lambda$	$c = 0.5$	1	1.5	2	5
<b>0.1</b>	17.942	16.3	15.15	15	3.1696
<b>0.2</b>	7.7168	3.7539	3.0388	3.146	3.9196
<b>0.3</b>	3.0683	3.249	3.54	3.809	5.6478

Pro menší počet cyklů  $p = 500$  proběhla simulace optimálně. Dle testování byly pro parametry zvoleny hodnoty  $\lambda = 1.5$  a  $c = 0.2$ . Při zvýšení počtu trénovacích cyklů na 30000 a nastavení chyby na  $E_{max} = 0.01$  simulace nedoběhla, především díky stagnující chybě. Počet neuronů ve vnitřní vrstvě byl tedy zvýšen na 4.

Pro čtyři neurony bylo provedeno jen krátké testování s počtem vzorků 500, kterým bylo ukázáno, že průběh simulace je přibližně stejný. Neuronová síť byla tedy spuštěna pro parametry  $\lambda = 1.5$  a  $c = 0.2$  a počet trénovacích cyklů 30000. Síť dosáhla chyby  $E_{max} = 0.01$  po 18542 trénovacích cyklech. Průběh chyby můžeme pozorovat na obr. 1.3.

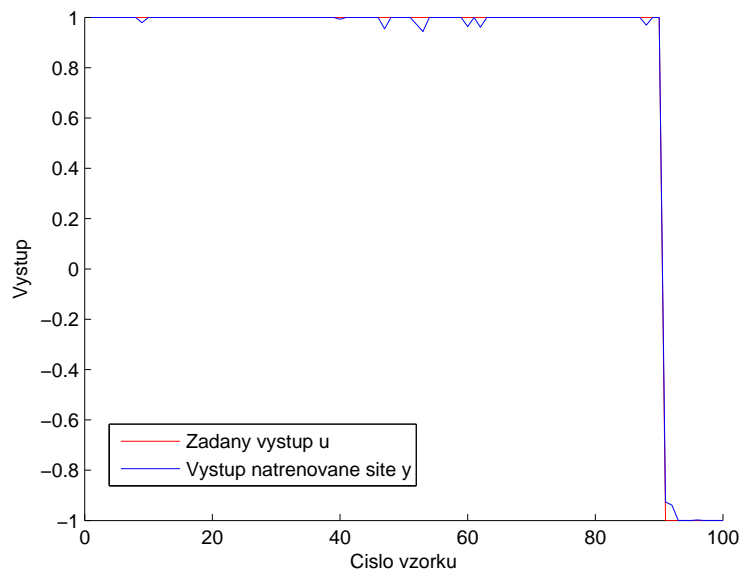


Obrázek 1.3: Chyba trénování

S přibývajícím počtem trénovacích cyklů by chyba  $E_c$  měla konvergovat k nule. Experimentálně tedy bylo vyzkoušeno trénování neuronové sítě pro 200000 trénovacích cyklů. Síť dosáhla chyby  $E_c = 0.000333$ .

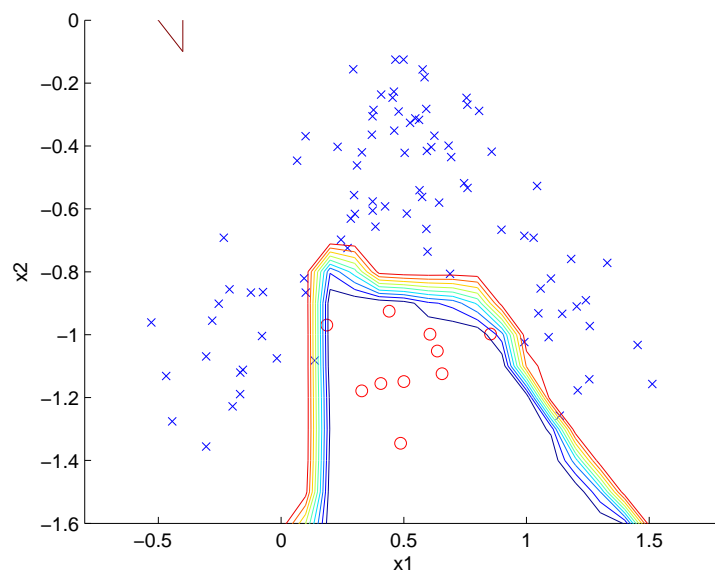
### 1.3 Natrénovaná neuronová síť

Síť jsme po vhodném nastavení parametrů natrénovali a na obr. 1.4 je dobře patrné porovnání požadovaného výstupu  $\mathbf{u}$  a výstupu získaného po natrénování neuronové sítě  $\mathbf{y}$ .



Obrázek 1.4: Porovnání výstupu  $\mathbf{u}$  a  $\mathbf{y}$

Z důvodu omezené kapacity počítače nebylo možné provést přesnější trénování sítě. Průběh chyby je exponenciální, takže s přibývajícím časem se stále pomaleji blíží k nule. Pro naše účely byla získaná chyba  $E_{max} = 0.01$  dostačující. Výsledné rozdělení shluků natrénované neuronové sítě můžeme vidět na obr. 1.5.



Obrázek 1.5: Výsledné rozdělení shluků



# Závěr

---

V semestrální práci byla provedena aplikace návrhu neuronové sítě oddělující dva shluky. Prvním krokem bylo zvolení vhodné struktury a aktivační funkce. Funkce byla zvolena jako nelineární, bipolární a spojitá. Díky této volbě byla při následném trénování použita metoda zpětné propagace. Tato metoda se hodí pro nelineární neuronové sítě. Během trénování jsme testovali vhodné nastavení parametrů  $\lambda$  a  $c$ .

Na závěr byla zvolena maximální požadovaná chyba  $E_{max} = 0.01$ , která byla pro naše účely dostačující a bylo provedeno porovnání výstupu požadovaného a výstupu získaného pomocí modelu neuronové sítě. Výsledky byly uspokojivé.

Negativem této práce byla především časová náročnost při trénování sítě, pro chybu  $E_{max} = 0.01$  trval výpočet cca 40 minut.