ANALÝZA A KLASIFIKACE BIOMEDICÍNSKÝCH DAT

prof. Ing. Jiří Holčík, CSc.

IV. NEURONOVÉ SÍTĚ II.

MU IBA

TYPY NEURONOVÝCH SÍTÍ

- existuje celá řada neuronových sítí, např.
  - vícevrstvá perceptonová síť (MLP)
  - Hopfieldova síť
  - Kohonenovy samoorganizující se mapy
  - síť RBF (Radial Basis Function – radiální bázická funkce)
  - ...
- každá neuronová síť je vhodná pro jiné třídy úloh;
- některé neuronové sítě se mohou vzájemně doplňovat;
- základními úlohami neuronových sítí jsou klasifikace a regrese (aproximace)
- podle přítomnosti „učitele" můžeme neuronové sítě dělit na sítě s učitelem a bez učitele

NÁVRH NEURONOVÉ SÍTĚ

- ☑ pro řešení každé úlohy musí být navržena jedinečná neuronová síť
- ☑ otázka vhodného výběru sítě
- ☑ výběr struktury sítě, tj. počet vstupů, výstupů, vrstev, skrytých neuronů, typ aktivačních funkcí, atd.
- ☑ výběr trénovacího algoritmu
- ☑ problém *over-sizing*, *over-learning (over-fitting)*

PERCEPTRONY

- ☑ neuronové sítě se strukturou s jednosměrným šířením zpracovávaného signálu

⇩

nejsou problémy se stabilitou, všechny výpočty se provedou v jednom taktu.

- ☑ pomalá konvergence a obtížné vyrovnávání se s lokálními minimy

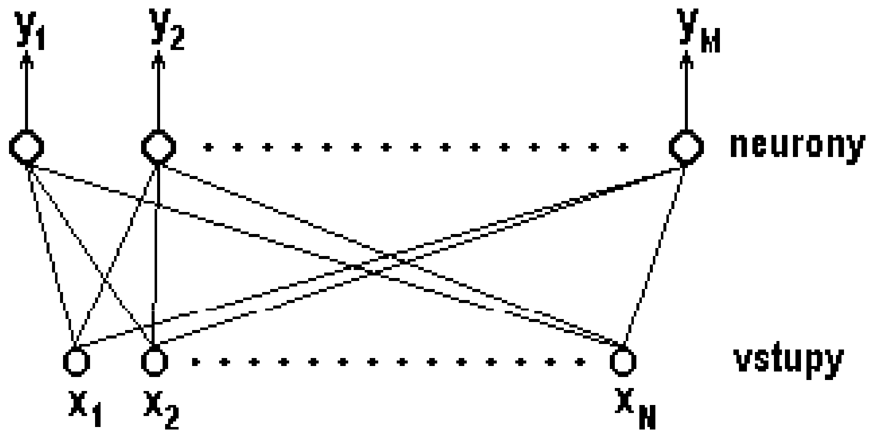
PERCEPTRONY

- ☑ neuron – vnitřní potenciál je počítán jako vážený počet vstupů;
- ☑ aktivační funkce je sigmoida (je spojitá, lze ji derivovat) –

$$y_i = f(net_i) = (1 + \exp(-\lambda.net_i))^{-1}$$

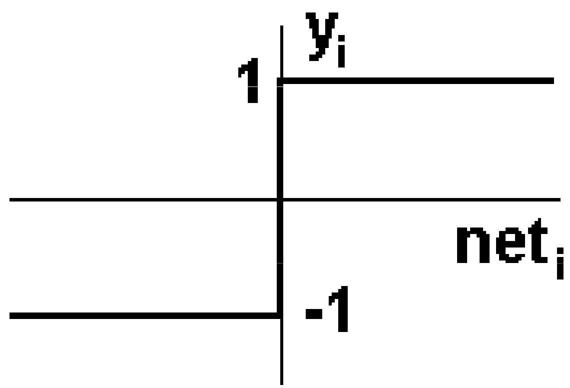
JEDNOVRSTVÝ N,M PERCEPTRON

- N,M perceptron má N vstupů a M výstupů, tj. M separátně pracujících neuronů.

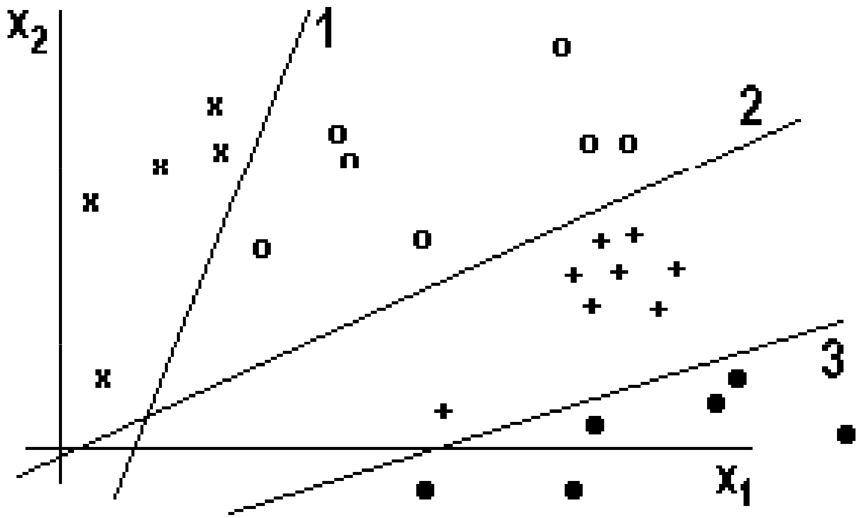


JEDNOVRSTVÝ N,M PERCEPTRON

Předpokládejme, že převodní charakteristiku i-tého neuronu zvolíme ve tvaru $y_i = f_i(net_i) = sign(net_i)$, přičemž aktivační hodnota $net_i = \sum_j w_{ij}.x_j$, kde x_j je hodnota přicházející z j-tého aktivačního zdroje na vstup i-tého neuronu s vahou w_{ij}.



JEDNOVRSTVÝ N,M PERCEPTRON



	x	o	+	•
y_1	1	-1	-1	-1
y_2	1	1	-1	-1
y_3	1	1	1	-1

JEDNOVRSTVÝ N,M PERCEPTRON

Učení N,M perceptronu spočívá v nastavení vah každého z použitých M neuronů.

Obecné požadavky na způsob úpravy vah:

- ☑ konvergence - rychle a monotónně;
- ☑ algoritmická formulace.

JEDNOVRSTVÝ N,M PERCEPTRON

Učení jednovrstvého perceptronu se skládá z následujících fází:

- ☑ náhodná počáteční volba vektorů vah;
- ☑ učení
  - ➔ na vstup perceptronu jsou přiváděny jednotlivé vzory z učební množiny a výsledky po zpracování perceptronem jsou srovnávány s požadovaným výstupem;
  - ➔ v případě, že se požadovaný a neuronovou sítí spočítaný výstup liší více než je povoleno, pak se modifikují hodnoty váhovacích koeficientů tak, aby se rozdíl mezi oběma hodnotami výstupů minimalizoval;
  - ➔ pokud je celková chyba zpracování větší než je povoleno, je třeba pokračovat v učení; v opačném případě lze perceptron použít.

JEDNOVRSTVÝ N,M PERCEPTRON

UČENÍ PODLE ROSENBLATTOVA δ PRAVIDLA

1. krok - počáteční nastavení vah a prahů (je výhodné, jsou-li náhodně nastaveny na malé hodnoty);
2. krok - vložení nového vzoru a zjištění hodnoty požadovaného výstupu d(t);
3. krok - výpočet výstupu

$$y(t) = f\left(\sum_{i=0}^{N-1} w_i(t)x_i(t) - \Theta\right)$$

JEDNOVRSTVÝ N,M PERCEPTRON

UČENÍ PODLE ROSENBLATTOVA δ PRAVIDLA

4. krok - adaptace vah

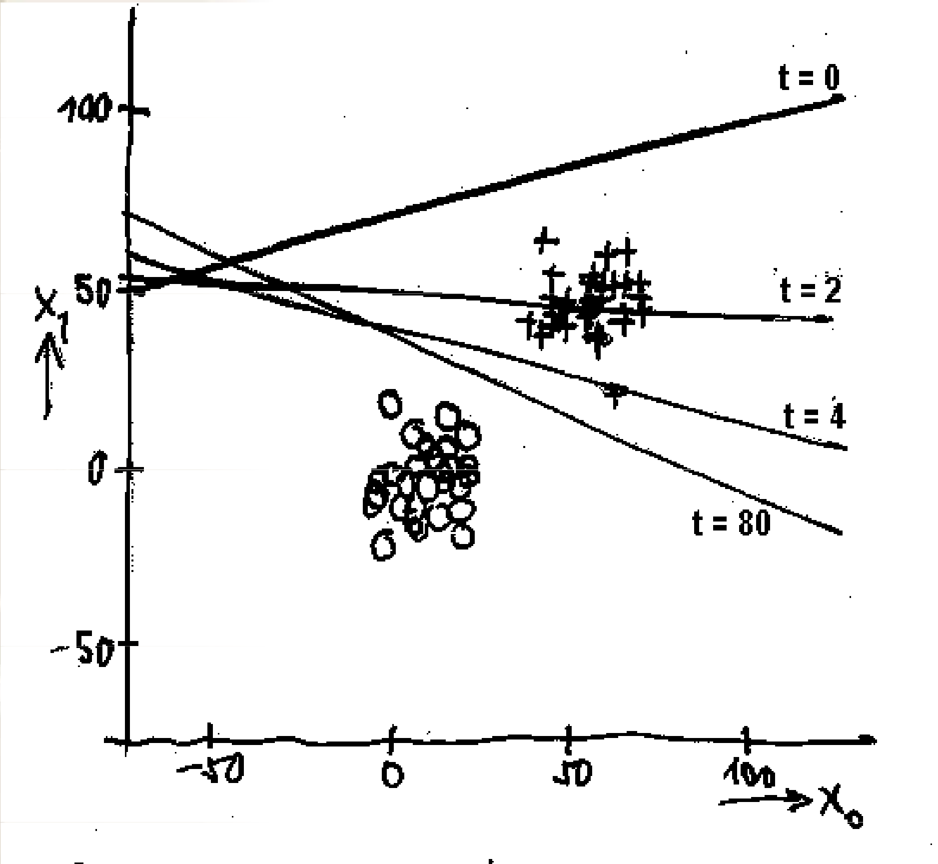
$$w_i(t+1) = w_i(t) + \eta.[d(t) - y(t)].x_i(t), \qquad 0 \leq i \leq N;$$

$$\delta = d(t) - y(t)$$

$\eta > 0$, $\eta < 1$ je konstanta ovlivňující rychlost konvergence - kompromis mezi rychlou adaptací a zohledněním dřívějších zkušeností;

5. krok - skok na krok 2;

JEDNOVRSTVÝ N,M PERCEPTRON



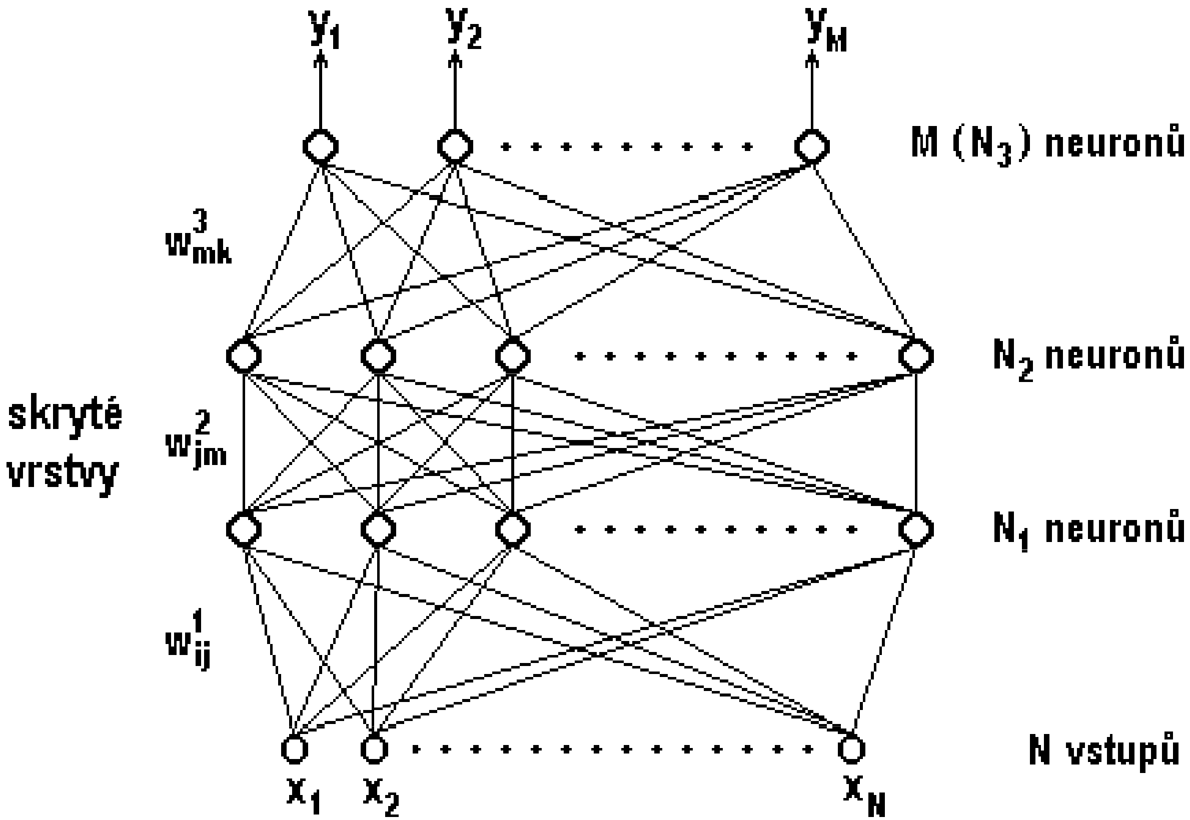
je dokázáno, že:

- jsou-li klasifikační třídy lineárně separabilní, procedura učení konverguje a hranice leží mezi oběma množinami;
- nejsou-li třídy lineárně separabilní, potom poloha hranice osciluje;

možné modifikace:

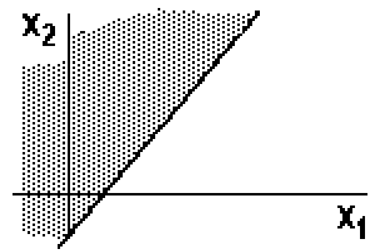
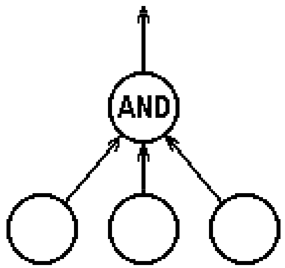
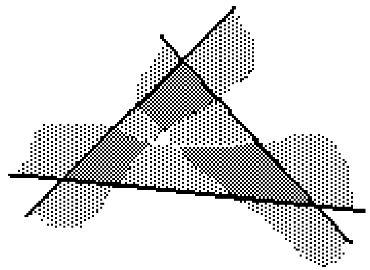
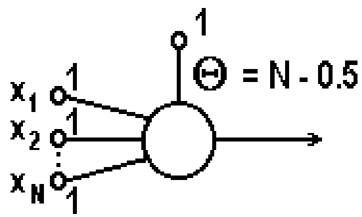
- optimalizace metodou nejmenších čtverců;
- využití gradientních metod.

VÍCEVRSTVÝ PERCEPTRON

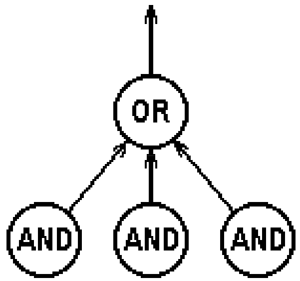
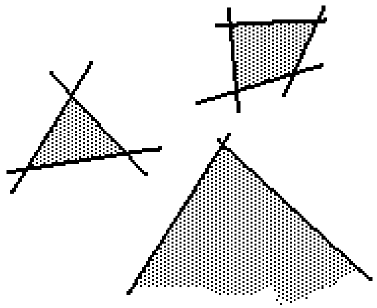
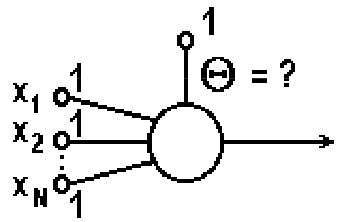
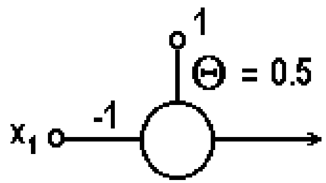


☑ Na rozdíl od jednovrstvého perceptronu může být vícevrstvý perceptron použit i pro řešení nelineárně separabilních problémů.

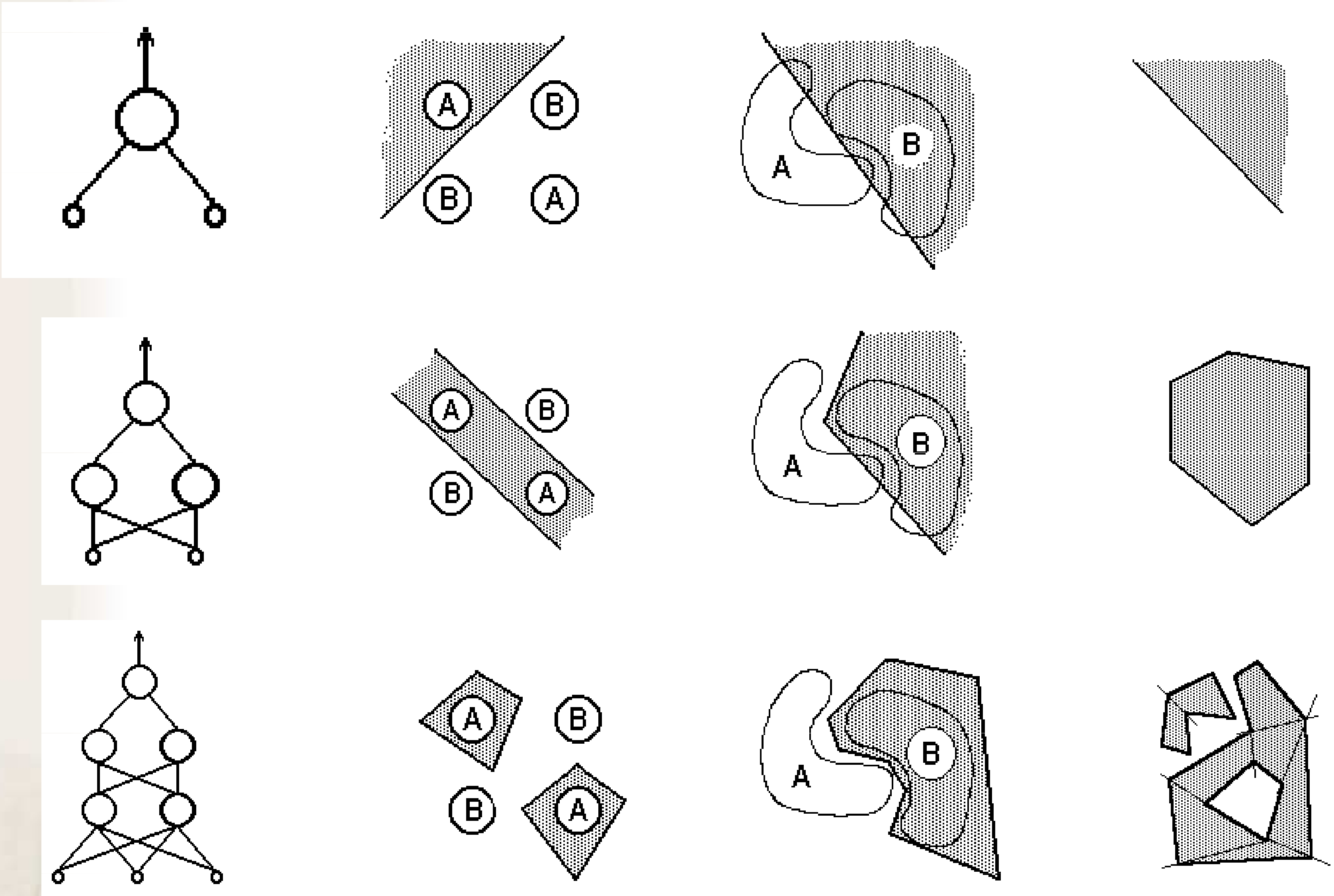
KOLIK POTŘEBUJEME V PERCEPTRONU VRSTEV?

1. *vrstva* - má-li neuron na výstupu ostrou nelinearitu, tj. výstup nabývá pouze dvou hodnot 0 a 1, resp. -1 a 1, dokáže prostor rozdělit pomocí lineární hranice na dvě poloroviny			
2. *vrstva* - realizuje logickou funkci „AND“			***neuronová realizace operátoru AND*** 

KOLIK POTŘEBUJEME V PERCEPTRONU VRSTEV?

<table>
<tr>
<td>3. vrstva - realizuje logickou vrstvu „OR“</td>
<td></td>
<td></td>
<td>neuronová realizace operátoru OR
</td>
</tr>
<tr>
<td colspan="3"></td>
<td>realizace operátoru negace
</td>
</tr>
</table>

ROZDĚLENÍ OBRAZOVÉHO PROSTORU PŘI RŮZNÉM POČTU VRSTEV PERCEPTRONU



KOLIK POTŘEBUJEME V KAŽDÉ VRSTVĚ UZLŮ?

- je-li počet neuronů malý, síť nedokáže postihnout všechny závislosti v trénovacích datech;
- je-li počet neuronů velký, zvyšuje se doba učení a navíc vlivem nadměrného počtu trénovacích dat síť špatně generalizuje vlivem tzv. přeučení (overfitting – zdůrazňování náhodných fluktuací pro učení sítě)

KOLIK POTŘEBUJEME V KAŽDÉ VRSTVĚ UZLŮ?

TEORIE

dokázáno:

- ☑ pro dokonalé oddělení dvou tříd (máme-li k dispozici N vzorů) je potřeba nejvýše N-1 neuronů;
- ☑ pro dokonalé oddělení dvou tříd stačí N/n neuronů, kde n je dimenze vstupního prostoru;
- ☑ počet skytých neuronů musí růst s velikostí trénovací množiny;
- ☑ daná funkce může být aproximována s libovolnou přesností perceptronovou sítí se dvěma skrytými vrstvami, přičemž v první skryté vrstvě má síť $nm(m+1)$ neuronů a v druhé vrstvě $m^2(m+1)^n$ neuronů, kde $m \geq 2n+1$

KOLIK POTŘEBUJEME V KAŽDÉ VRSTVĚ UZLŮ?

HEURISTIKA

- N_1 … dáno počtem potřebných lineárních hranic
- N_2 … = 1, pokud je oblast definované třídy konvexní;

  > 1, pokud oblast není konvexní; v tom případě je N_2 rovno počtu konvexních oblastí (ve většině případů je $N_2 \ll N_1$);
- N_3 … počet tříd;

KOLIK POTŘEBUJEME V KAŽDÉ VRSTVĚ UZLŮ?

JINÁ HEURISTIKA

pro třívrstvý perceptron (dvě skyté vrstvy a jedna výstupní)

první vrstva: o něco více neuronů, než je vstupů;

druhá vrstva: počet nueronů daný aritmetickým průměrem počtu neuronů v první vrstvě a počtu výstupů

KOLIK POTŘEBUJEME V KAŽDÉ VRSTVĚ UZLŮ?

JEŠTĚ JINÁ HEURISTIKA

první vrstva: dvojnásobek počtu vstupů a výstupů;

druhá vrstva: polovic neuronů v první vrstvě