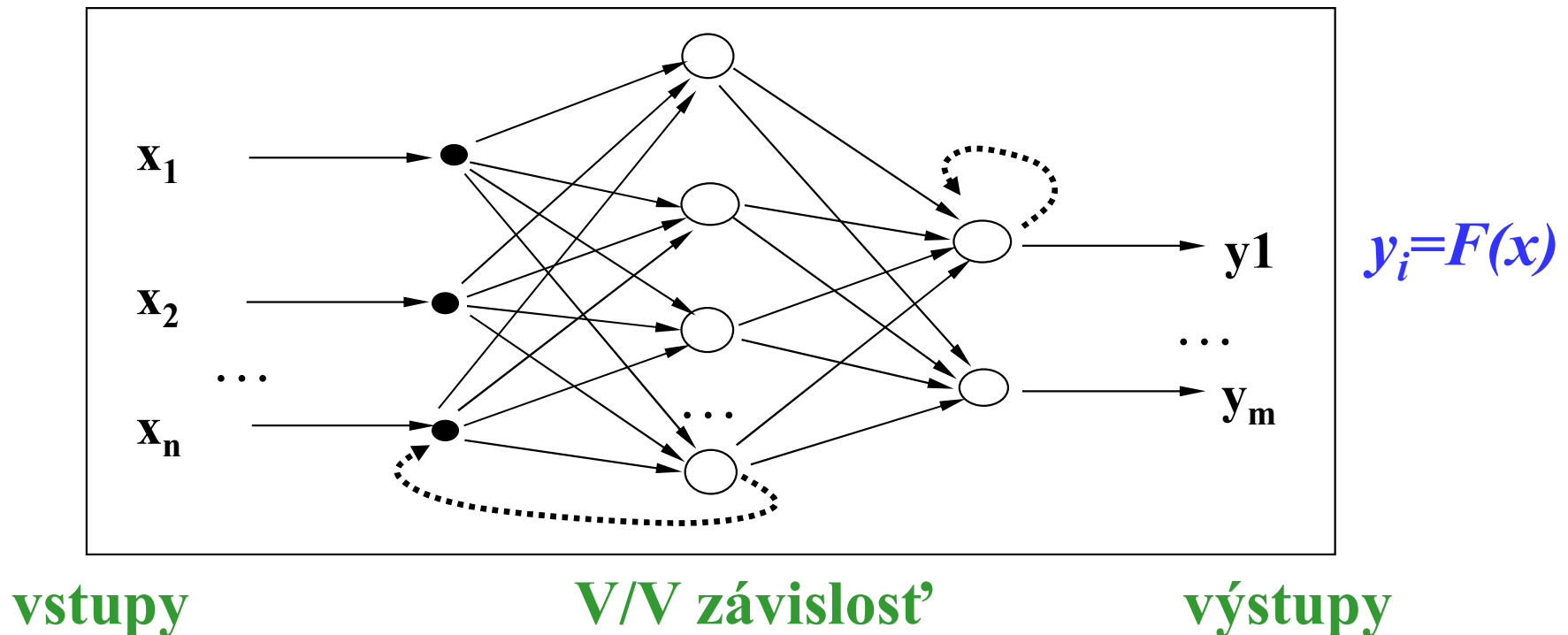


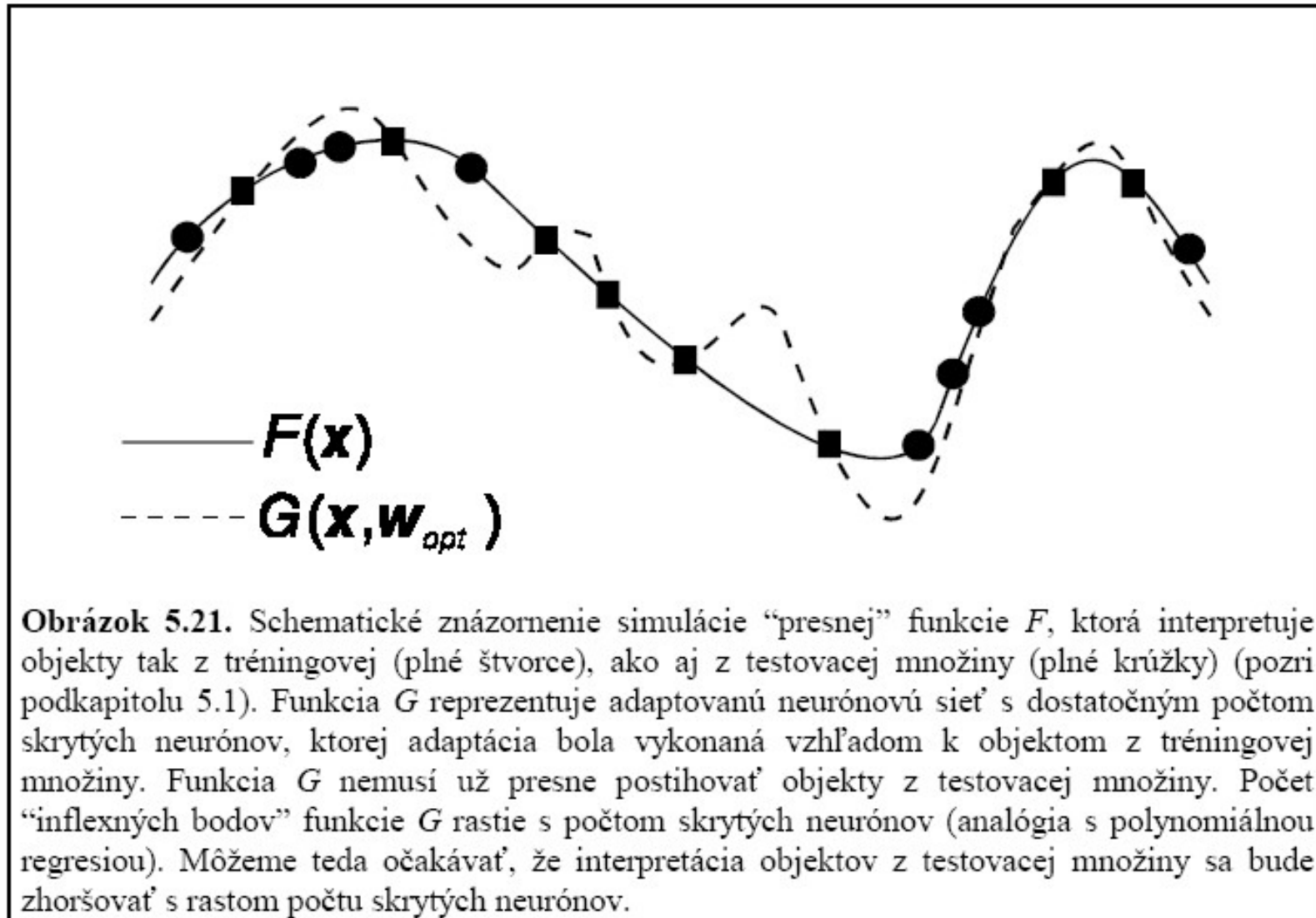
Aplikácie umelých neurónových sietí

UNS – univerzálny aproximátor (nelineárnych) vstupno/výstupných zobrazení



- Modelovanie zobrazní $R^n \rightarrow R^m$
- Aproximácia bezčasových (statických) závislostí
- Aproximácia časových (dynamických) závislostí (čas, oneskorenia, zotrvačnosti, signály)
- Parametrizácia (trénovanie) na základe existujúcich V/V (historických) údajov - vzorov

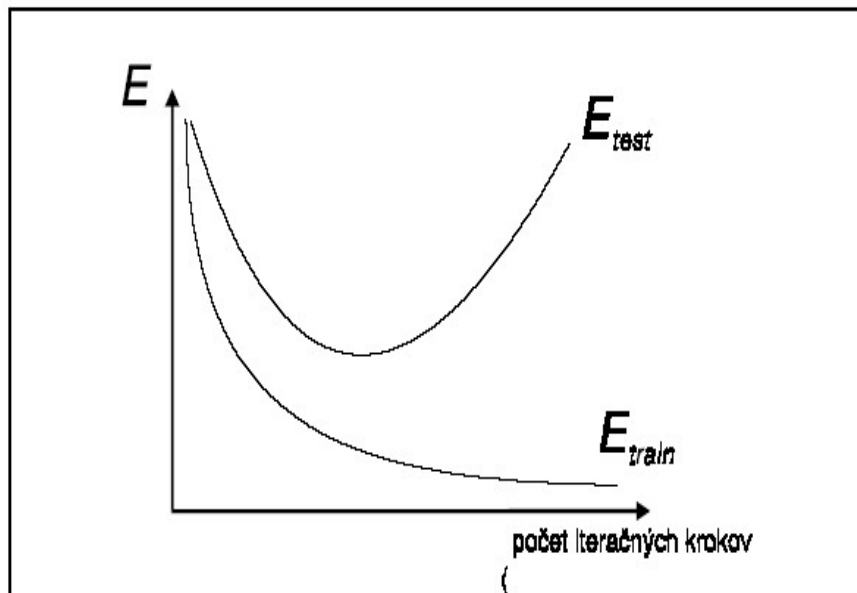
Architektúra NS a počet adaptačných krokov pri aproximácii nelineárnej funkcie $y=f(x)$



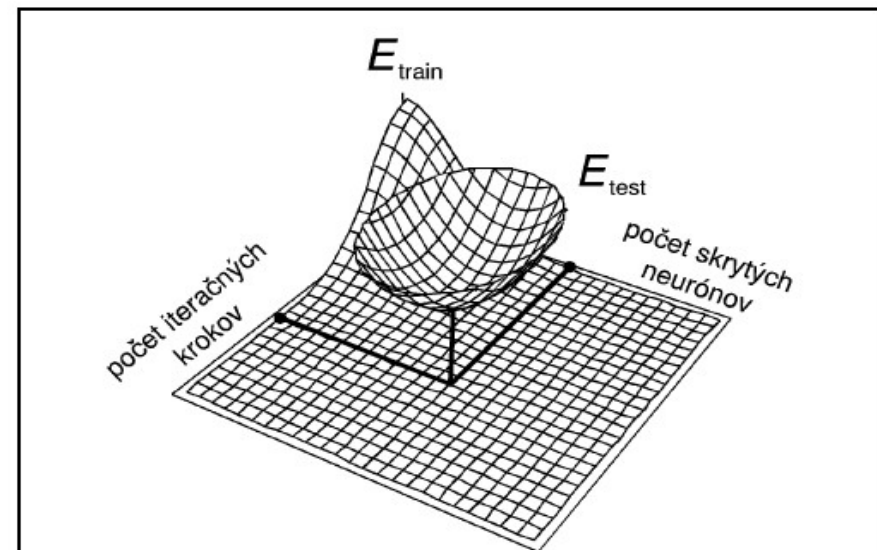
Architektúra NS a počet adaptačných krokov pri aproximácii nelineárnej funkcie $y=f(x)$

$$E_{train} = \frac{1}{2} \sum_{i=1}^{A_{train}} (G(w, x_i) - \hat{x}_i)^2$$

$$E_{test} = \frac{1}{2} \sum_{i=1}^{A_{test}} (G(w, x_i) - \hat{x}_i)^2$$



Obrázok 5.22. Pribeh účelových funkcií pre tréningovú a testovaciu množinu v závislosti na počte iteračných krokov adaptácie (resp. na počte skrytých neurónov) neurónovej siete. Účelová funkcia E_{train} klesá s rastom počtu iteračných krokov (resp. skrytých neurónov), zatiaľ čo účelová funkcia E_{test} vykazuje minimum.



Obrázok 5.23. Schematické znázornenie priebehu účelových funkcií E_{train} a E_{test} vzhľadom k počtu skrytých neurónov a počtu iteračných krokov. Na rozdiel od účelovej funkcie E_{train} , ktorá monotónne klesá s rastúcim počtom skrytých neurónov a s rastúcim počtom iteračných krokov, účelová funkcia E_{test} má minimum pre optimálny počet skrytých neurónov a iteračných krokov (pozri hrubé čiary na grafe).

Typy aplikácií UNS v praxi

- **Klasifikácie do tried**
- **Asociácia, rozpoznávanie**
- **Predikcia, modelovanie**
- **Konceptualizácia**
- **Filtrácia**
- **iné**

Klasifikácia do tried

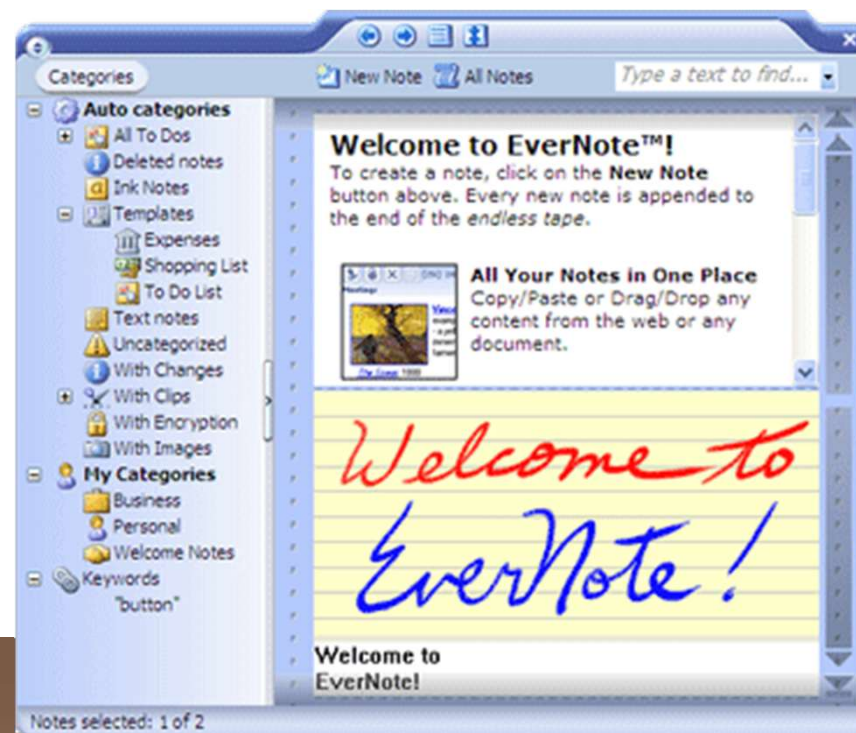
- Rozdeľovanie objektov, javov do tried podľa definovaných kritérií
- - Počet vstupov závisí od počtu uvažovaných kritérií (ukazovateľov).
 - Počet výstupov závisí od počtu tried.
 - Počet neurónov v skrytých vrstvách závisí od miery zložitosti problému.
- Typy sietí :
 - Kohonenova sieť
 - Viacvrstvá perceptrónová sieť
 - Counter-propagation network
 - Pravdepodobnostné n.s.
 - ...
- Príklady :
 - rozdeľovanie jablák a pomarančov
 - rozdeľovanie objektov podľa veľkosti a hmotnosti
 - rozdeľovanie výrobkov podľa kvality, oddelovanie nepodarkov

Asociácia, rozpoznávanie

- **Rozpoznávanie naučených vzorov podľa predlôh (autoasociácia)**
- **Schopnosť rozpoznať vzory, ktorých informácia je čiastočne porušená, chýbajúca, neúplná**
- **Typy sietí :**
 - **Hopfieldova sieť**
 - **Boltzmannov stroj**
 - **Hammingova sieť**
 - **Bidirekcionálna asociatívna pamäť**
 - **Spatio-temporal pattern recognition**

Príklady rozpoznávania

- Rozpoznanie písmen, čísel, obrazcov, obrysov
- Rozpoznávanie textu
 - tlačенý text
 - rukou písaný text
- Rozpoznávanie obrazu
 - orientácia robota v priestore, ...
 - kamera, sonar, radar, IR, ultrazvuk
 - identifikácia bojových cieľov



DOKÁŽETE PREČÍTAŤ NASLEDUJÚCU VETU ?

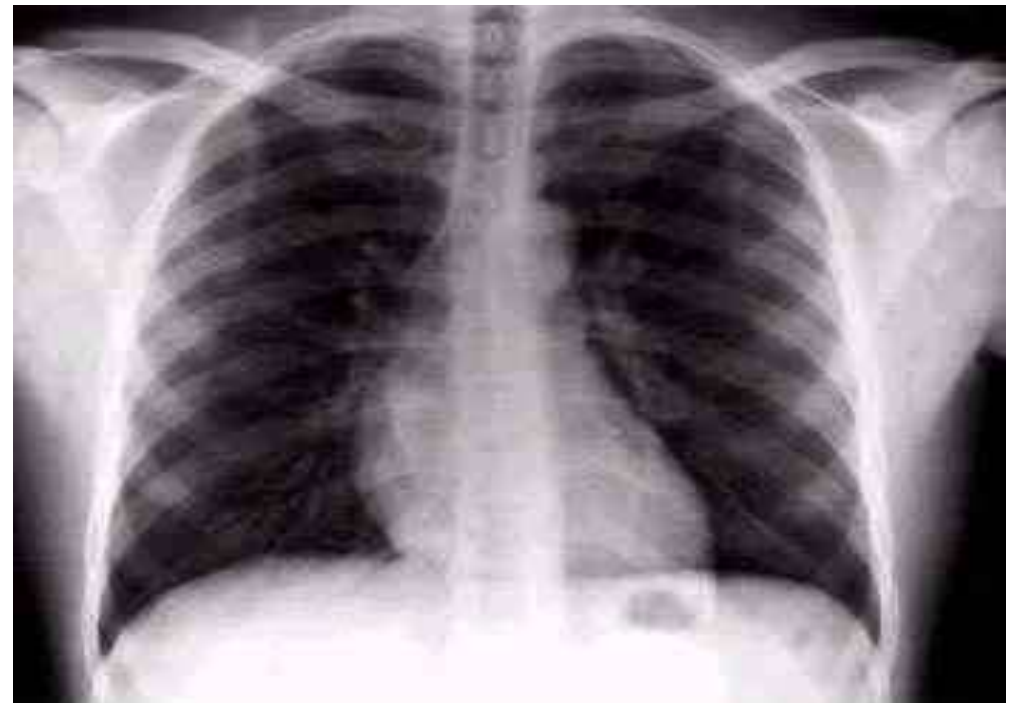
**V SÚIVSOLTSI S VSÝKUMOM NA CMABRIDGE
UINERV TISY**

**VLŠYO NJAVA O, ŽE NZEÁELŽÍ NA PORDAÍ PSÍEMN V
SOLVE. JEIDNÁ DLEÔITŽÁ VEC JE, ABY BLOI PVRÉ
A PSOELNDÉ PÍMESNÁ NA SRPVÁONM MESTIE.
ZOBÝTK MŽÔE BÝŤ TÁTONLA ZEMS A TY TO SÁLTE
BEZ PORLBOMÉV PERTČÍ AŠ.**

**JE TO PTERO, ŽE ĽDUKSÝ MOZOG NČETÍA KDAŽÉ
PENSÍMO,**

ALE SVOLO JKAO CLEOK. ZJÍUAMVAÉ, ŽE ?

- **Rozpoznávanie ľudských tvárí**
 - prístup oprávnených osôb do priestorov
 - hľadanie teroristov na letiskách a staniciach
- **Identifikácia odtlačkov prstov, očných zreníc**
- **Medicína:**
 - vyhodnocovanie RTG snímok, tomografia (CT)
 - diagnostika nádorov
 - vyhodnocovanie EKG, EEG signálov



Predikcia, modelovanie

- **Aproximácia nelineárnych zobrazení, funkcií**
- **Modelovanie dynamických systémov**
- **Predikcia (predpovedanie) budúcich výstupov systému na základe aktuálnych (aj minulých) stavov**
- **Typy sietí :**
 - **Viacvrstvové perceptrónové siete (MLP)**
 - **RBF siete**
 - **Neurónové siete s ortogonálnymi aktivačnými funkciami**
 - **iné siete**

Príklady modelovania a predikcie

- **Interpolácia medzi nameranými údajmi, model n -rozmerného zobrazenia ($\mathbb{R}^n \rightarrow \mathbb{R}^m$), model funkcie n -premenných**
- **Modelovanie dynamiky fyzikálnych systémov v čase (dynamický systém)**
 - **model pohybu hladiny v nádrži**
 - **model pohybu lietadla, rakety**
 - **model dynamiky chemických reakcií**

- **Predikcia kvality výroby v reálnom čase → včasný zásah s cieľom zabránenia nepodarkovosti, optimalizácia výroby**
- **Predpovedanie problémov vo výrobe skôr než nastanú, zabránenie haváriám, kolíziám, finančným stratám, poruchám strojov**



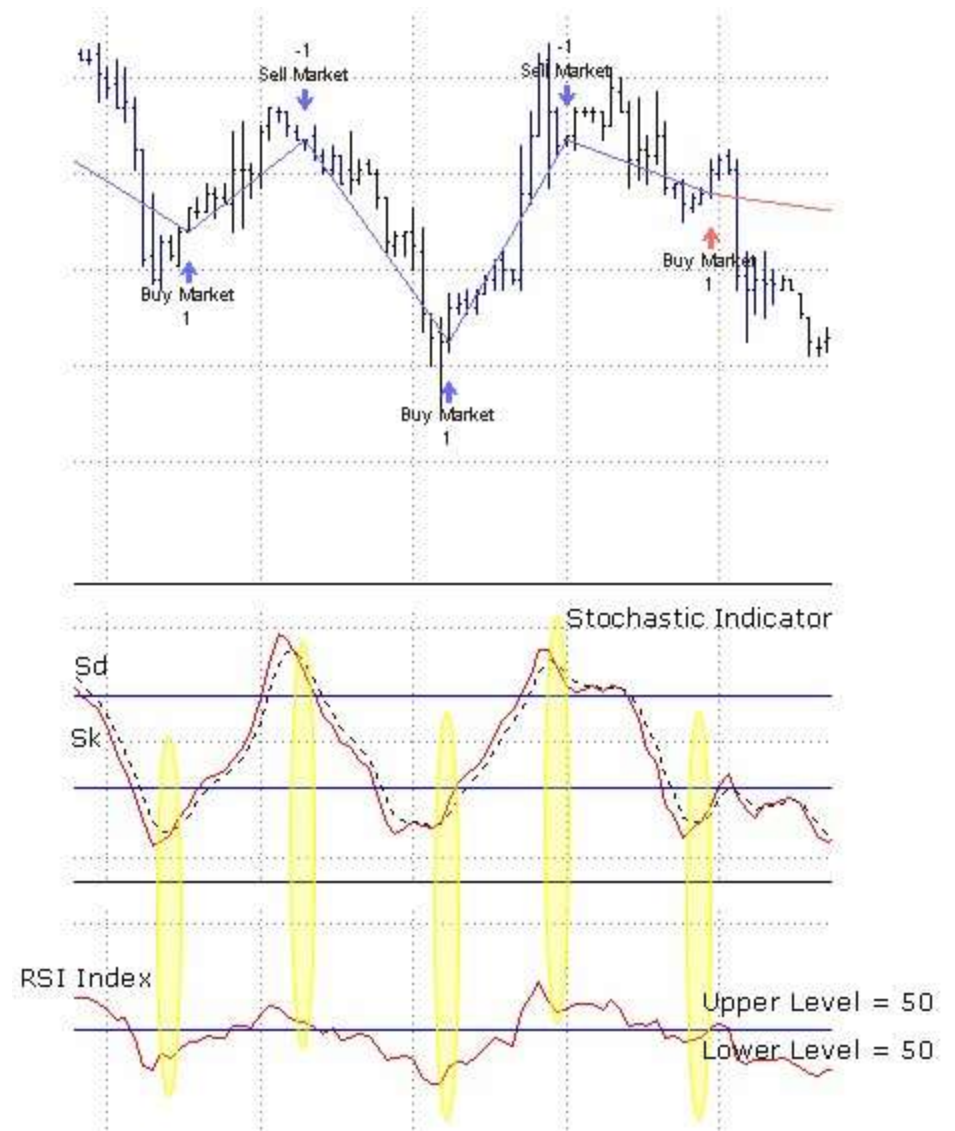
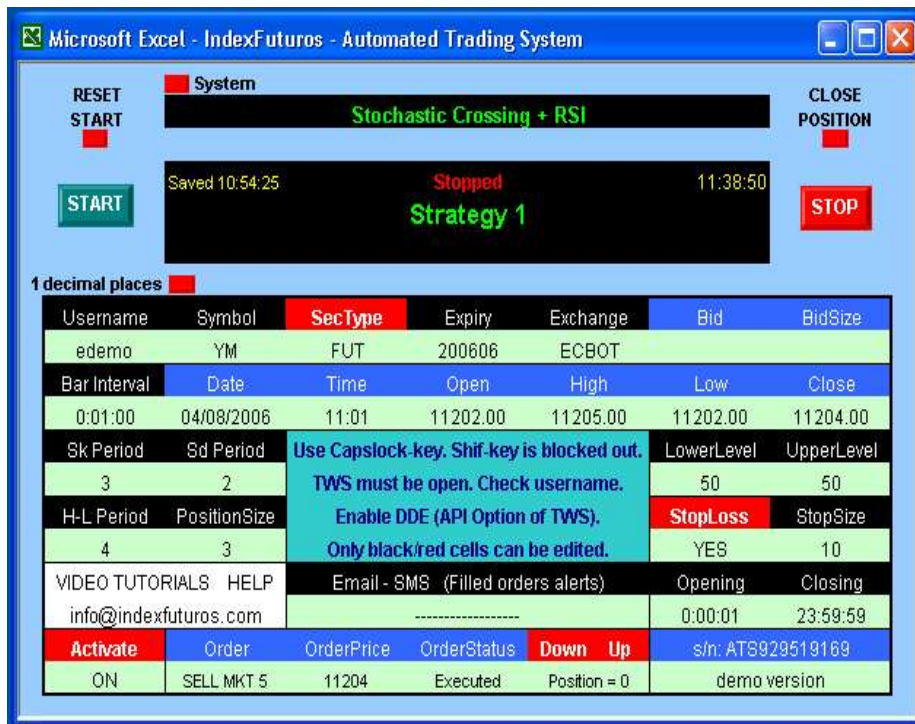
- **Predikcia dopraných problémov, včasné zabránenie zápcham ...**



- **Modelovanie a predikcia ekologických systémov, meteorologických a hydrologických javov ...**
 - **predikcia emisií**
 - **kvalita vôd, záplavy**
 - **kvalita ovzdušia**



- **Modelovanie a predikcia finančných systémov a javov**
 - predikcia pohybu cien komodít, akcií (až 97 % presnosť)
 - trendy pohybu kurzov



Konceptualizácia

- Triedenie objektov, údajov podľa nejednoznačných, nejasne definovaných kritérií
- Hľadanie hlbších súvislostí medzi informáciami
- Typy sietí :
 - Adaptívne rezonančné siete
 - Samoorganizujúce sa mapy
 - ...
- Príklad : vytvorenie zoznamu potenciálnych zákazníkov zo skupín ľudí na základe ich veku, vzdelania, výšky príjmov

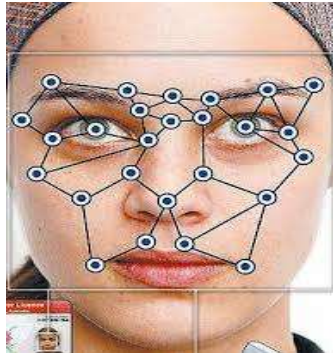
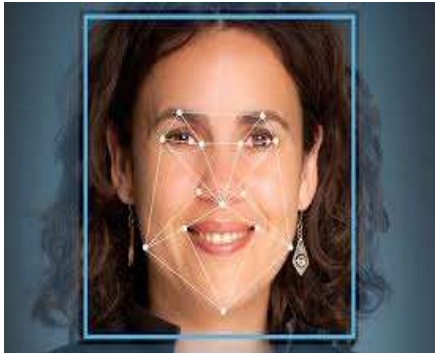
Filtrácia

- Vyhľadzovanie zašumených signálov
- Potláčanie šumu pri prenose dát
- Oddelenie šumu od hlasových signálov v lietadlách
- Typy sietí : Adaptívne lineárne siete (MADALINE)



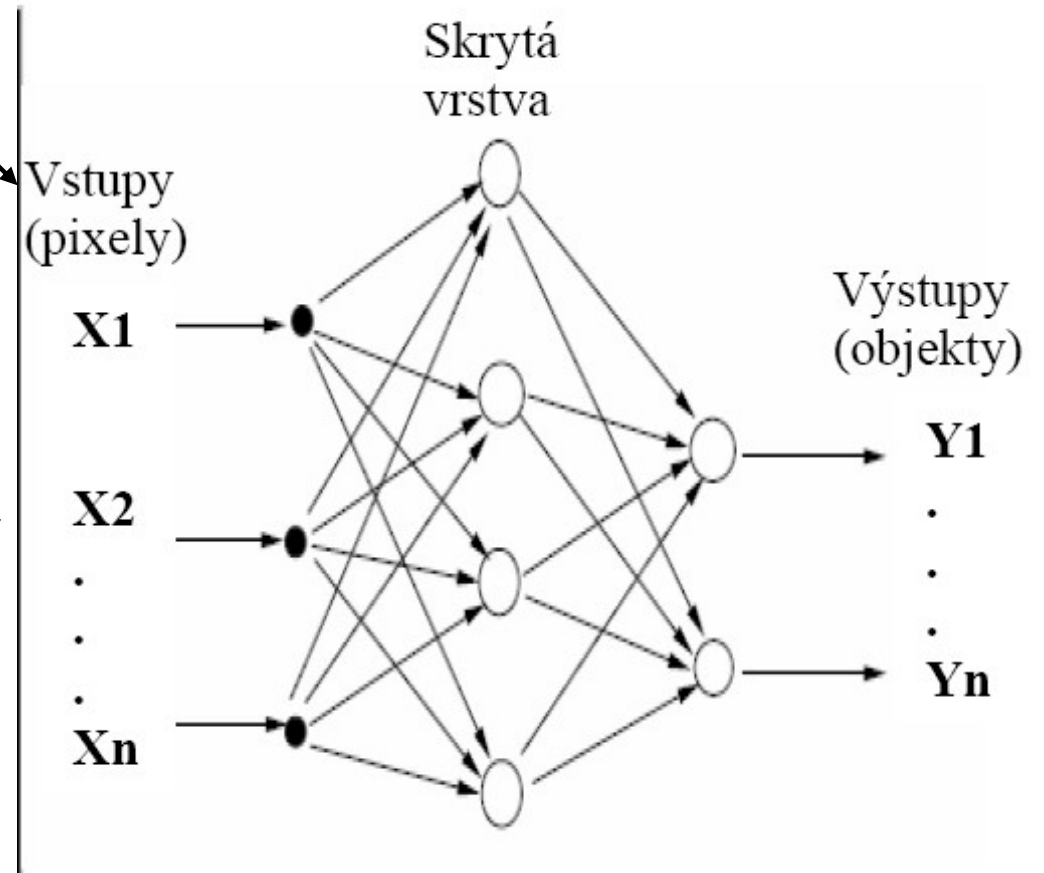
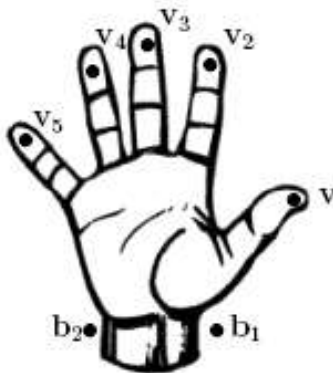
Príklady aplikácií UNS

Rozpoznávanie pomocou MLP sietí

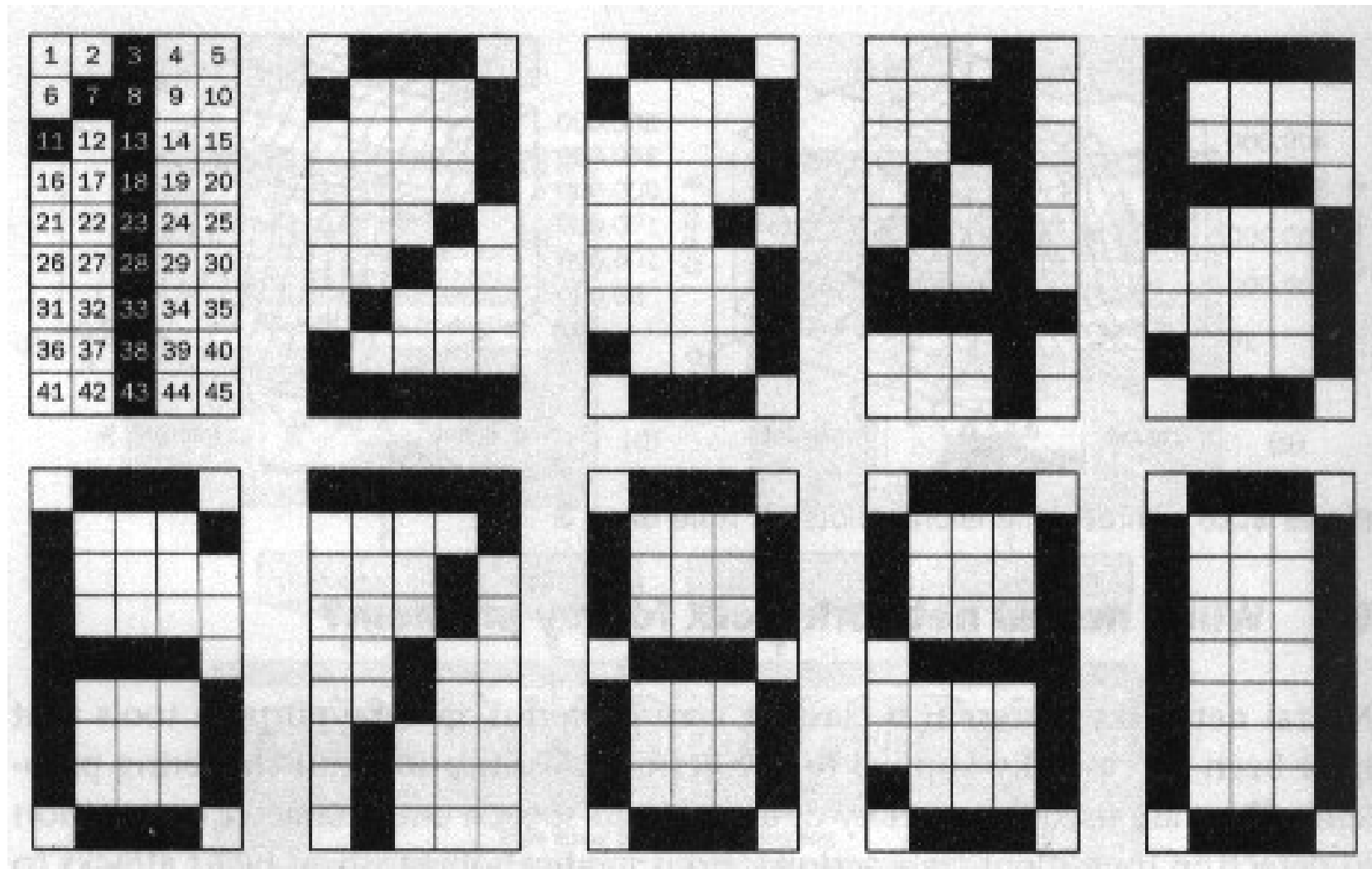


0 1 2 3 4

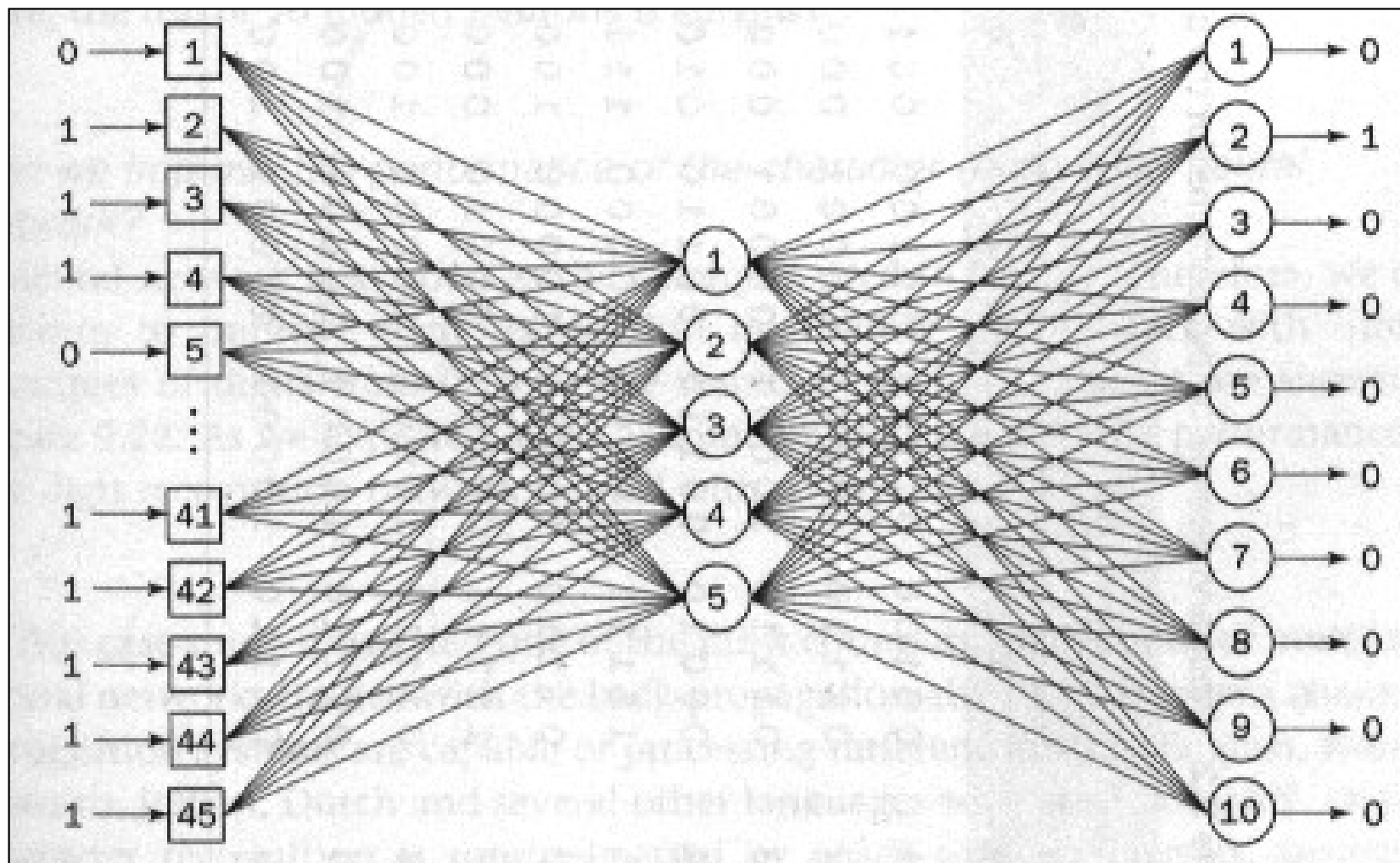
5 6 7 8 9



Rozpoznávanie znakov

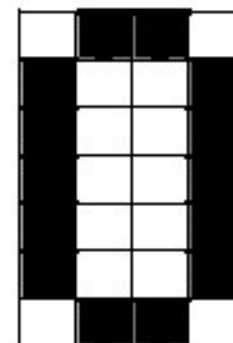
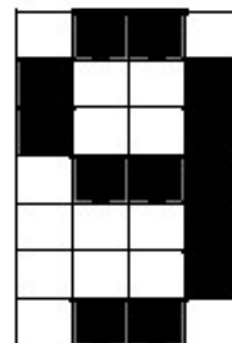
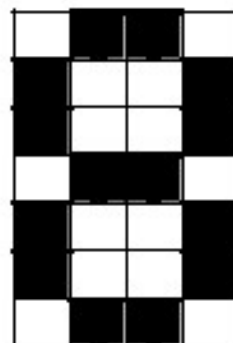
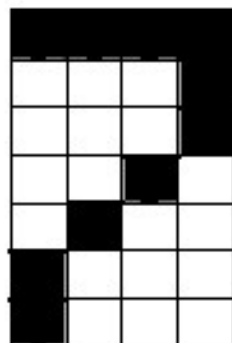
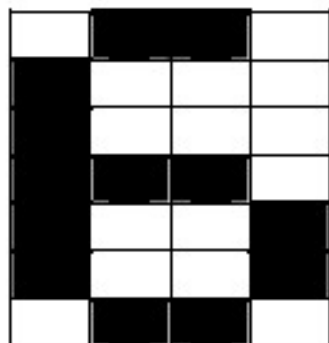
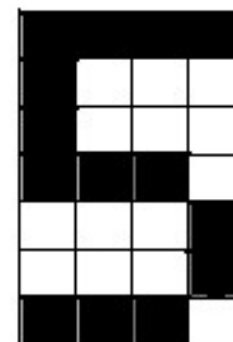
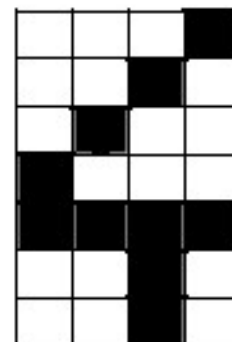
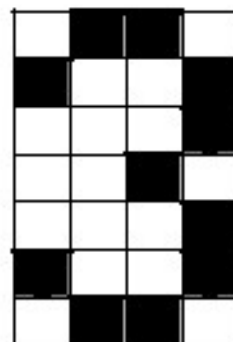
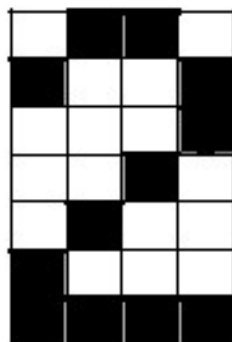


3-vrstvová MLP sieť



Rozpoznávanie číslíc

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28



```
% Program na natréovanie NS na rozpoznávanie číslíc 0 až 9
% NS je trojvrstvová sieť s aktivačnými funkciami 'logsig'
% NS má 28 vstupov - raster 4x7 - hodnota 0 až 1
% NS má 10 výstupov - aké číslo je rozpoznané - hodnota 0 až 1
```

```
% vygenerovanie dát pre číslice 0 až 9
[dataX,dataY]=gennumbers(0:9);
```

```
% zistenie rozmerov dát
[R,Q] = size(dataX); [S2,Q] = size(dataY);
```

```
% vytvorenie štruktúry NS, 28 vstupov z rozsahom (0,1), 10 výstupov s
% funkciou , 'softmax', 1 skrytá vrstva s počtom neurónov 15 a funkciou 'tansig'
S1 = 15;
```

```
net = patternnet(S1);
```



```
% nastavenie parametrov trenovania epoch  
net.trainParam.goal = 0.000001;% error goal.  
net.trainParam.show = 20;    % Frequency of progress displays (in epochs).  
net.trainParam.epochs = 5000; % Maximum number of epochs to train.
```

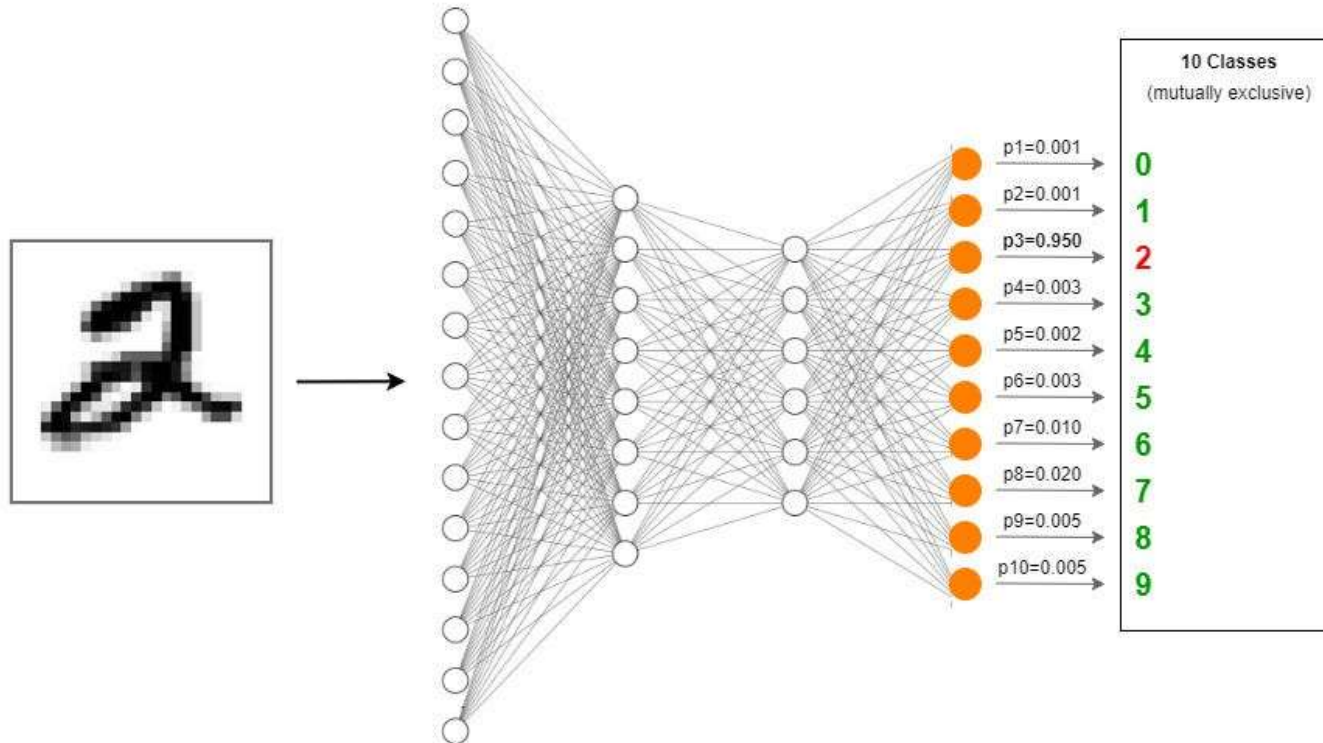
```
% trenovanie NS  
net = train(net,dataX,dataY);
```

```
% simulacia vystupu NS pre trenovacie data  
outnetsim = sim(net,dataX)
```

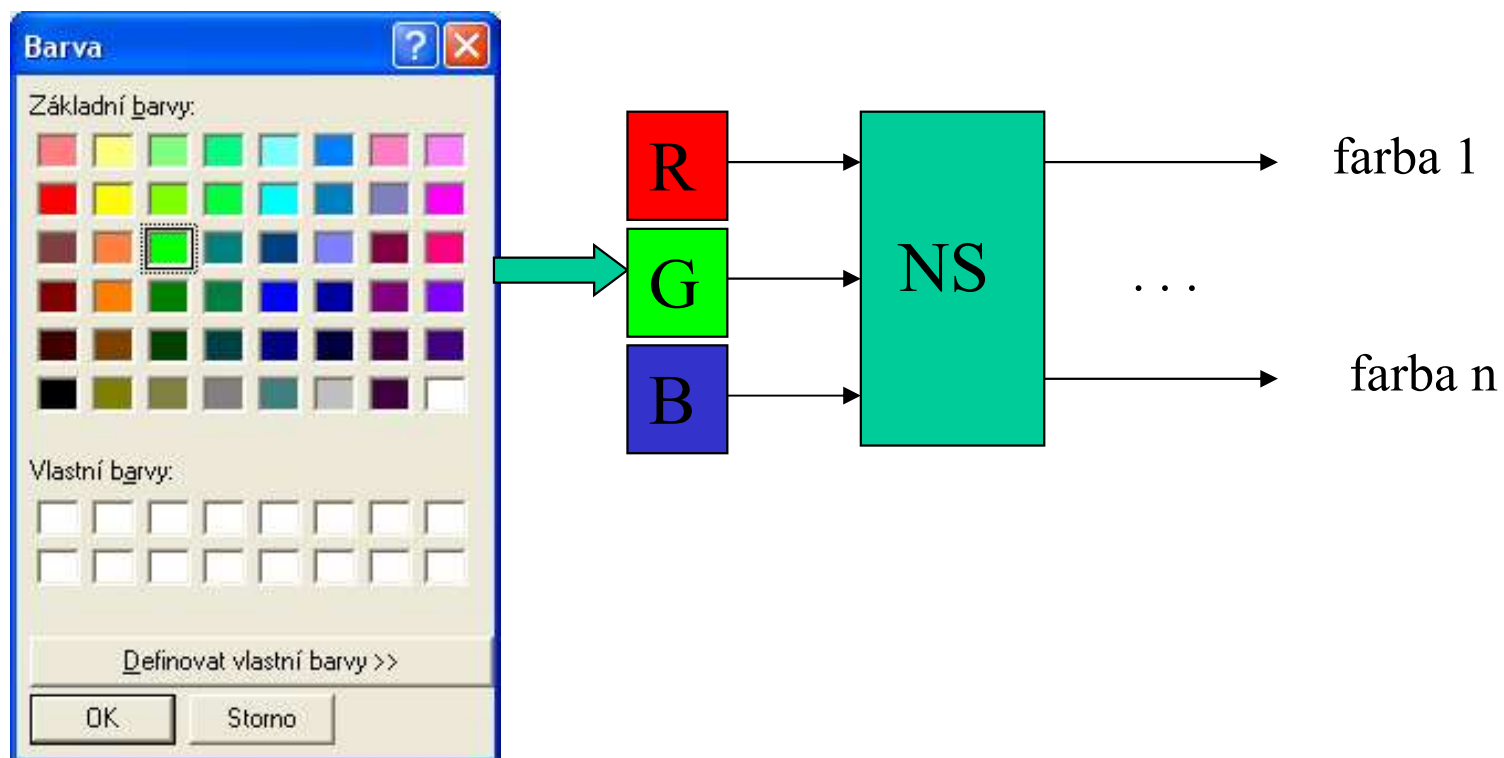
```
% chyba NS a dat  
err=(outnetsim-dataY);  
figure; surf(err); title('Odchylka NS od trenovacich dat')
```

```
% cislo 2 so sumom 0.4  
dataX=gennumbers(2,0.4);  
cislo = sim(net,dataX); cislo=round(cislo)'
```

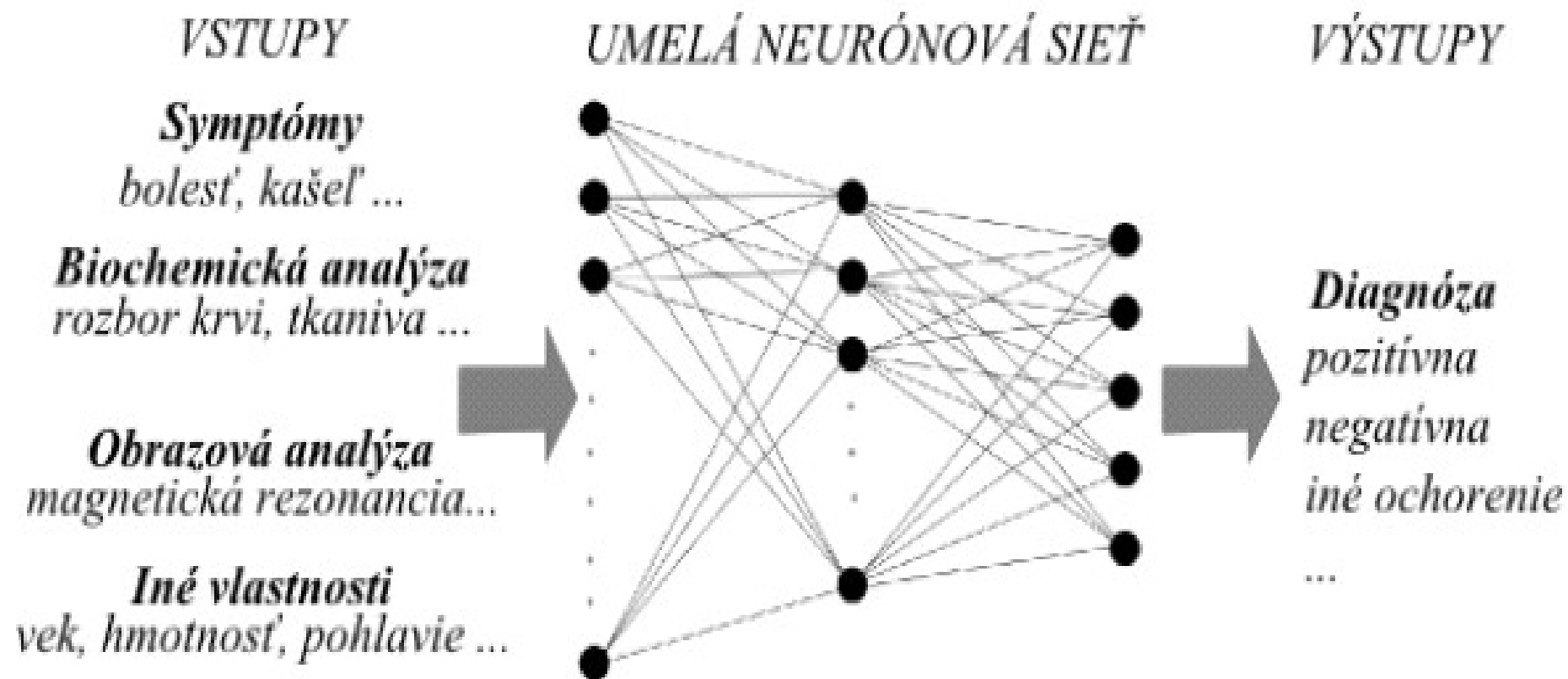
Rozpoznávanie rukou písaných číslíc - MNIST



Rozpoznávanie farieb



Diagnostika ochorení pomocou NS



Úspešnosť klasifikácie, senzitivita (citlivosť), špecificita

sensitivity, recall, hit rate, or true positive rate (TPR)

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} = 1 - FNR$$

specificity, selectivity or true negative rate (TNR)

$$TNR = \frac{TN}{N} = \frac{TN}{TN + FP} = 1 - FPR$$

precision or positive predictive value (PPV)

$$PPV = \frac{TP}{TP + FP} = 1 - FDR$$

negative predictive value (NPV)

$$NPV = \frac{TN}{TN + FN} = 1 - FOR$$

accuracy (ACC)

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN}$$

balanced accuracy (BA)

$$BA = \frac{TPR + TNR}{2}$$

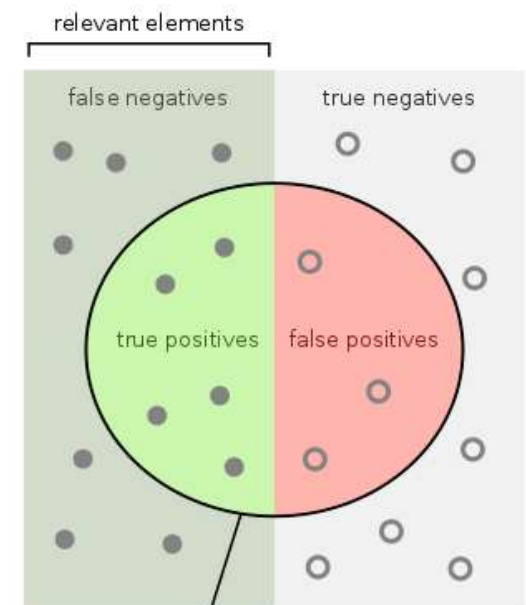
F1 score

is the **harmonic mean** of **precision** and **sensitivity**

$$F_1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$$

		Confusion Matrix		
Output Class	Negative	TN 19932 50.6%	FP 852 2.2%	TNR 95.9% 4.1%
	Positive	FN 700 1.8%	TP 17926 45.5%	TPR 96.2% 3.8%
		Negative	Positive	
		Target Class		
		NPV 96.6% 3.4%	PPV 95.5% 4.5%	ACC 96.1% 3.9%

Úspešnosť



selected elements

How many relevant items are selected?
e.g. How many sick people are correctly identified as having the condition.

How many negative selected elements are truly negative?
e.g. How many healthy people are identified as not having the condition.

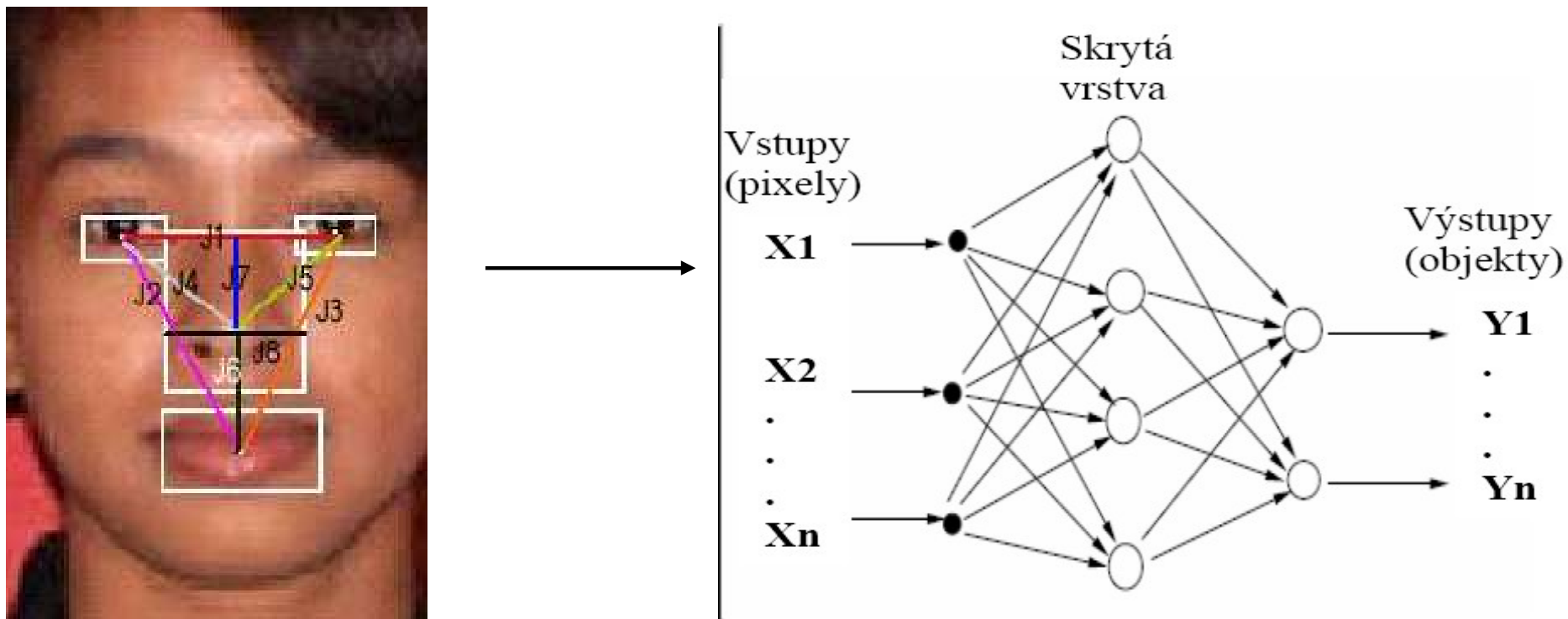
Sensitivity =



Specificity =



Rozpoznávanie tváre



























**Vstupné data NS sú normalizované vzdialenosti J1 až J8,
normalizované sú do rozsahu <0,1>**

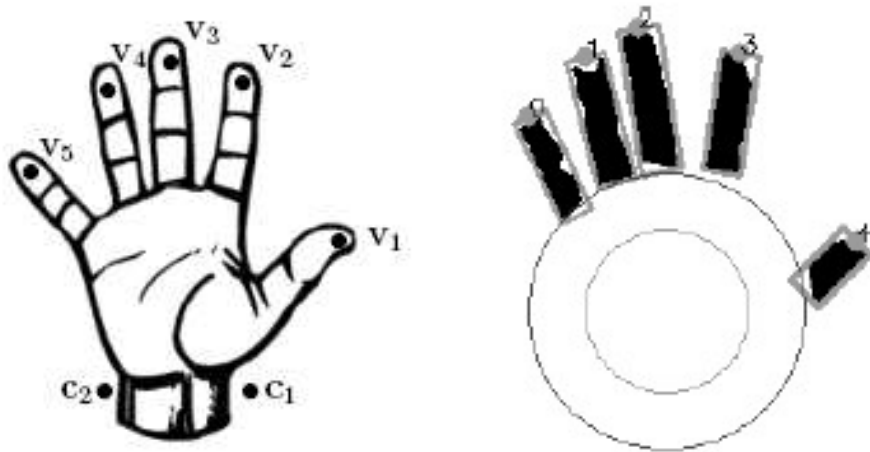
**Výstup NS je príslušnosť ku osobe (tvári),
15 výstupov**

$$Y=[1\ 0\ 0\ \dots\ 0; 0\ 1\ 0\ \dots\ 0; \dots; 0\ 0\ \dots\ 1]$$

Klasifikácia gest ruky

  0	  1	  2	  3	  4	  5
  6	  7	  8	  9	  10	  11

Klasifikácia gest ruky

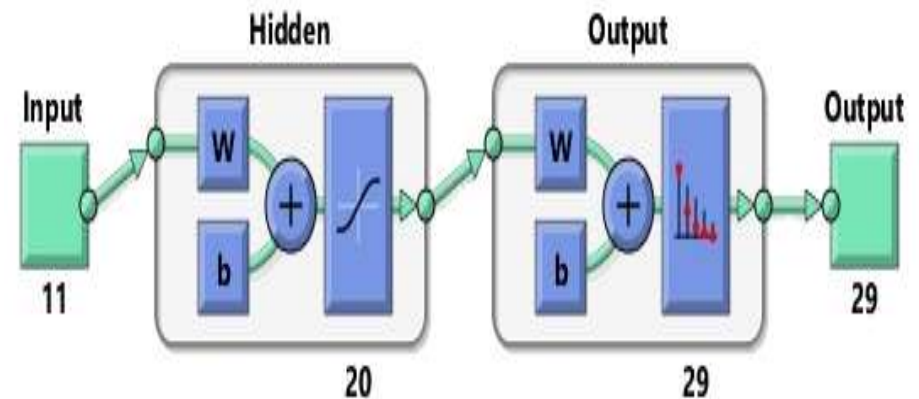
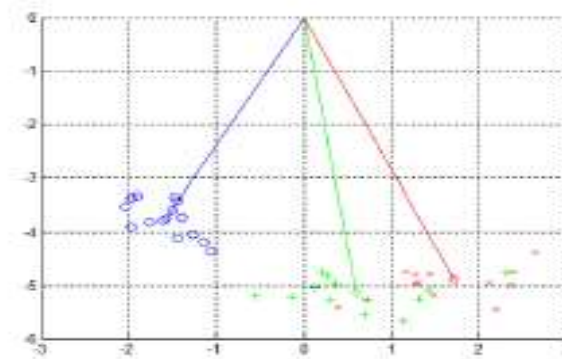
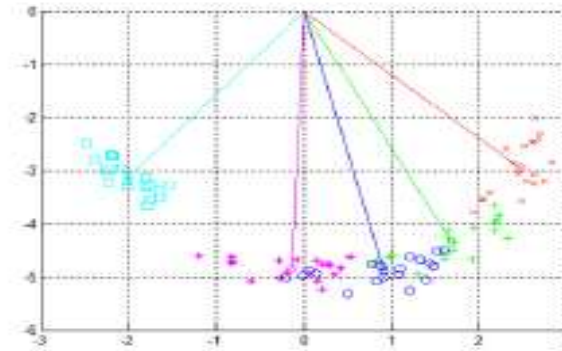


Vstup data NS sú normalizované pozície koncov prstov (x_i, y_i) voči stredu dlane a normalizovaná plocha dlane p .

$X = [x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, x_5, y_5, p]$

Výstup NS je príslušnosť ku gestu

$Y = [1 \ 0 \ 0 \ \dots \ 0; 0 \ 1 \ 0 \ \dots \ 0; \dots; 0 \ 0 \ \dots \ 1]$



Kritériálna (chybová) funkcia

MSE – stredná kvadratická chyba (mean squared error)

Používaná pri aproximácii.

$$E = \frac{1}{n} \sum_{i=1}^n (y_i - Y_i)^2$$

CE – krížová entropia (cross entropy)

Používaná pri klasifikácii

$$E = -\frac{1}{n} \sum_{i=1}^n [y_i \ln Y_i + (1 - y_i) \ln(1 - Y_i)]$$

n – počet vzoriek, y – vzorový (target) výstup, Y-výstup NS

Krížová validácia

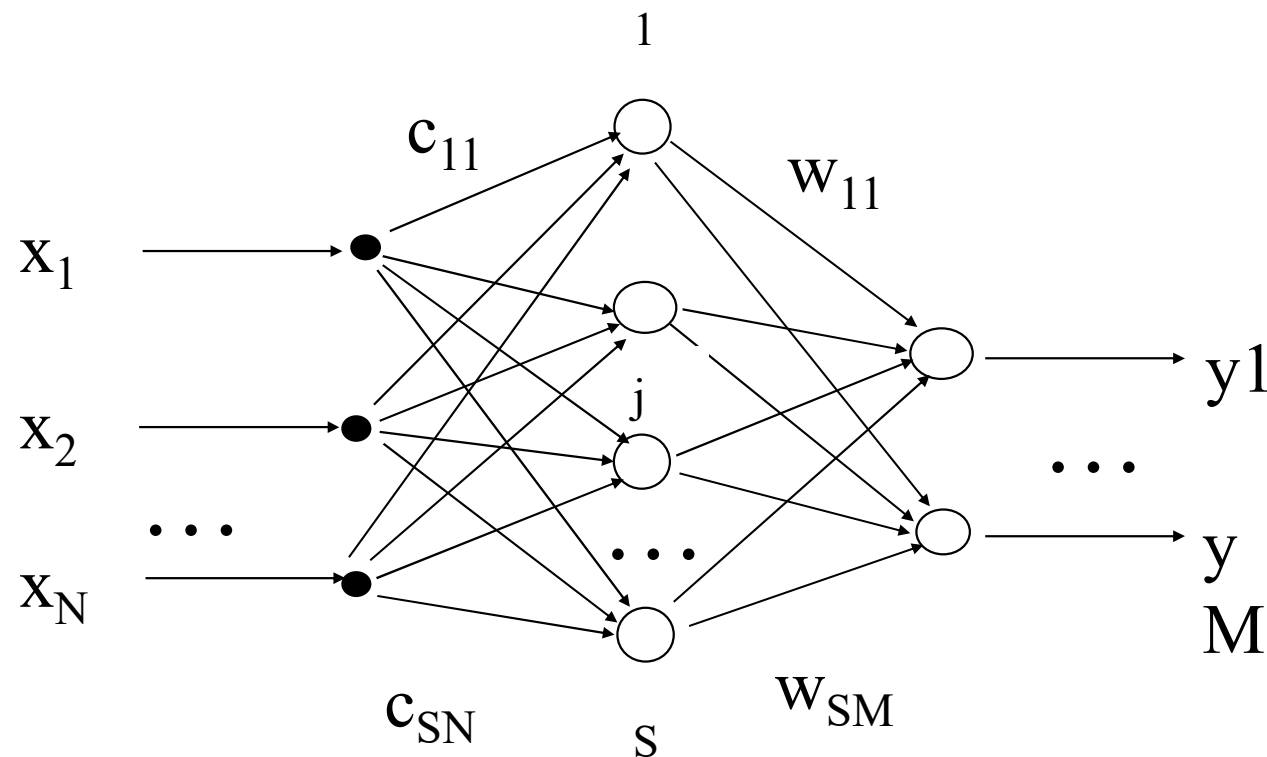
Krížová validácia slúži na zistenie vplyvu rozloženia tréningových dát na kvalitu modelu

- **Data sa rozdelia na N častí s rovnakým počtom vzoriek (5 alebo 10)**
- **Pri klasifikácii v každej časti dát by mali približne rovnaké počty vzoriek z každej skupiny**
- **Zvolíme rôzne kombinácie častí dát na tréningovanie a testovanie NS.
Napri.: máme 100 vzoriek, rozdelíme data na 5 častí po 20 vzoriek. Ak chceme 60% vzoriek dať na tréningovanie, potom časti dát 1,2,3 dáme na tréningovanie a časti 4,5 na testovanie. Ďalšie kombinácie: 2,3,4 -> 5,1; 3,4,5->1,2 ...**
- **Natrénujeme NS. Vyčíslíme chybu NS (úspešnosť klasifikácie) a vypočítame minimálnu, maximálnu a priemernú chybu.**

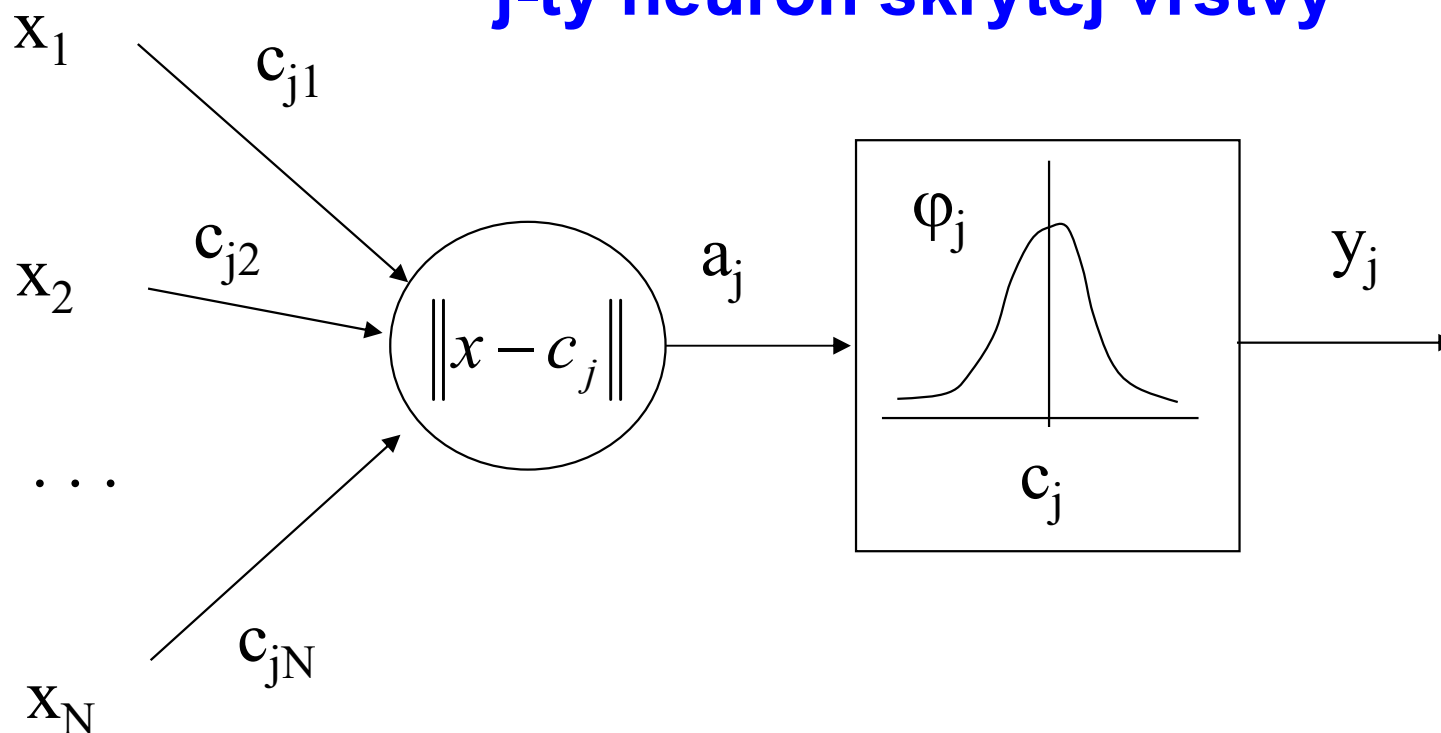
***Neurónové siete s radiálnymi
bázovými funkciami
(RBF - siete)***

RBF -siete

- Obsahujú tri vrstvy neurónov
- Vstupná vrstva distribuuje vstupné signály cez váhované spoje do neurónov skrytej vrstvy
- Skrytá vrstva realizuje nelineárnu transformáciu
- Výstupná vrstva uskutočňuje pre každý výstup siete iba lineárnu kombináciu výstupov neurónov skrytej vrstvy



j-ty neurón skrytej vrstvy



- Vstupný vektor x je v danom kroku privedený cez váhy c_{ji} do vstupnej časti neurónu
- Je vypočítaná norma (najčastejšie euklidovská), čo je vzdialenosť váhovaných vstupov od “centra” radiálnej bázovej funkcie c_j .
- Transformačná funkcia neurónu je spravidla symetrická n -rozmerná nelineárna funkcia so stredom v bode c_j . Stred c_j pritom predstavuje bod v priestore so súradnicami $[c_{j1}, \dots, c_{jN}]$, kde $i=1,2,\dots,N$ je počet vstupov a $j=1,2,\dots,S$ je počet “RBF” neurónov v skrytej vrstve.

**Typicky používanou transformačnou funkciou je
gaussovská f. v tvare**

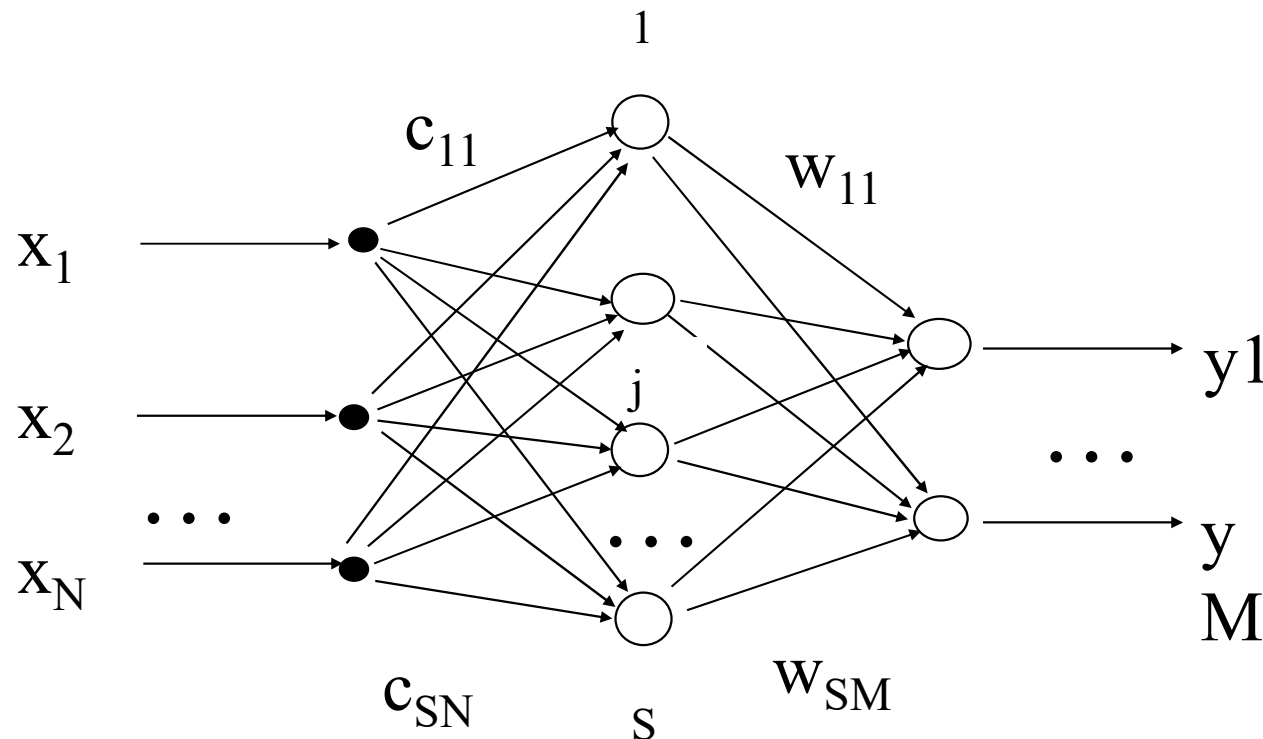
$$\varphi(a) = \exp\left(-\frac{a^2}{2\sigma^2}\right) \qquad a = \|x - c_j\|$$

Niekedy sa používa aj tzv. “thin-plate-spline” funkcia

$$\varphi(a) = a^2 \ln(a)$$

RBF sieť s N vstupmi, S neurónmi v skrytej vrstve a M výstupmi dokáže aproximovať ľubovoľné nelineárne zobrazenie cez priestory $R^N \rightarrow R^M$, ktoré možno vyjadriť v tvare

$$y_k(x) = \sum_{j=1}^S w_{jk} \varphi_j(\|x - c_j\|) \quad k = 1, 2, \dots, M$$



Trénovanie RBF siete

1.krok – určenie vstupných váh c_{ji} , čo zodpovedá polohe centier skrytých neurónov. Triviálne riešenia sú náhodný výber alebo rovnomerný výber bodov vo vstupnom priestore. Možno však použiť aj iné, účinnejšie algoritmy ako napr. minimalizáciu celkovej sumy vzdialeností medzi centrami a vstupnými bodmi x .

2.krok – nastavenie pomocných parametrov RBF siete, napr. parametrov σ (šírka) radiálnych bazových funkcií pomocou minimalizácie vhodne zvolenej krit. f. alebo heuristicky. Šírka jednotlivých RBF má vplyv na zovšeobecňovaciu schopnosť siete.

3. krok – nastavenie výstupných váh w_{jk} minimalizáciou krit. f.

$$E(w) = \frac{1}{2} \sum_{k=1}^N \left(\|d_k - y(x_k)\|^2 \right)$$

ktorá minimalizuje odchýlku výstupu siete oproti predlohe $D=\{d_k\}_{k=1,2,..,N}$.

D je množina vstupno/výstupných (vektorových) dvojíc a N je počet aproximovaných bodov, ktorý sa rovná počtu skrytých neurónov.

Rozdiely medzi sieťami RBF a MLP

- Perceptrón môže používať principiálne viac skrytých vrstiev, RBF sieť iba 1
- Každý neurón RBF zvyčajne máva svoje vlastné parametre
- Argumentom aktivačnej funkcie RBF je (euklidovská) norma vstupného vektora od centra RBF funkcie
- Na výstupnej hodnote perceptrónovej siete sa podieľa mnoho neurónov (alebo všetky), pri RBF pre daný vstup sú aktívne iba niektoré (tzv. „lokálne neuróny“)
- Perceptrónová sieť dokáže generalizovať aj v regiónoch vstupného priestoru, v ktorých je k dispozícii len málo tréningových dát. RBF nato, aby dokázala dostatočne dobre zovšeobecňovať potrebuje dostatočne vhodne „rozvrstvený“ súbor tréningových bodov a pre každý bod jeden RBF neurón.
- Trénovanie perceptrónovej siete je výpočtovo náročné a zdĺhavé, trénovanie RBF siete je rýchle.

Kohonenove siete

(Samoorganizujúce sa mapy –SOM)

Kohonenová sieť

Typ siete : Samoorganizujúca sieť nepotrebuje učiteľa k trénovaniu

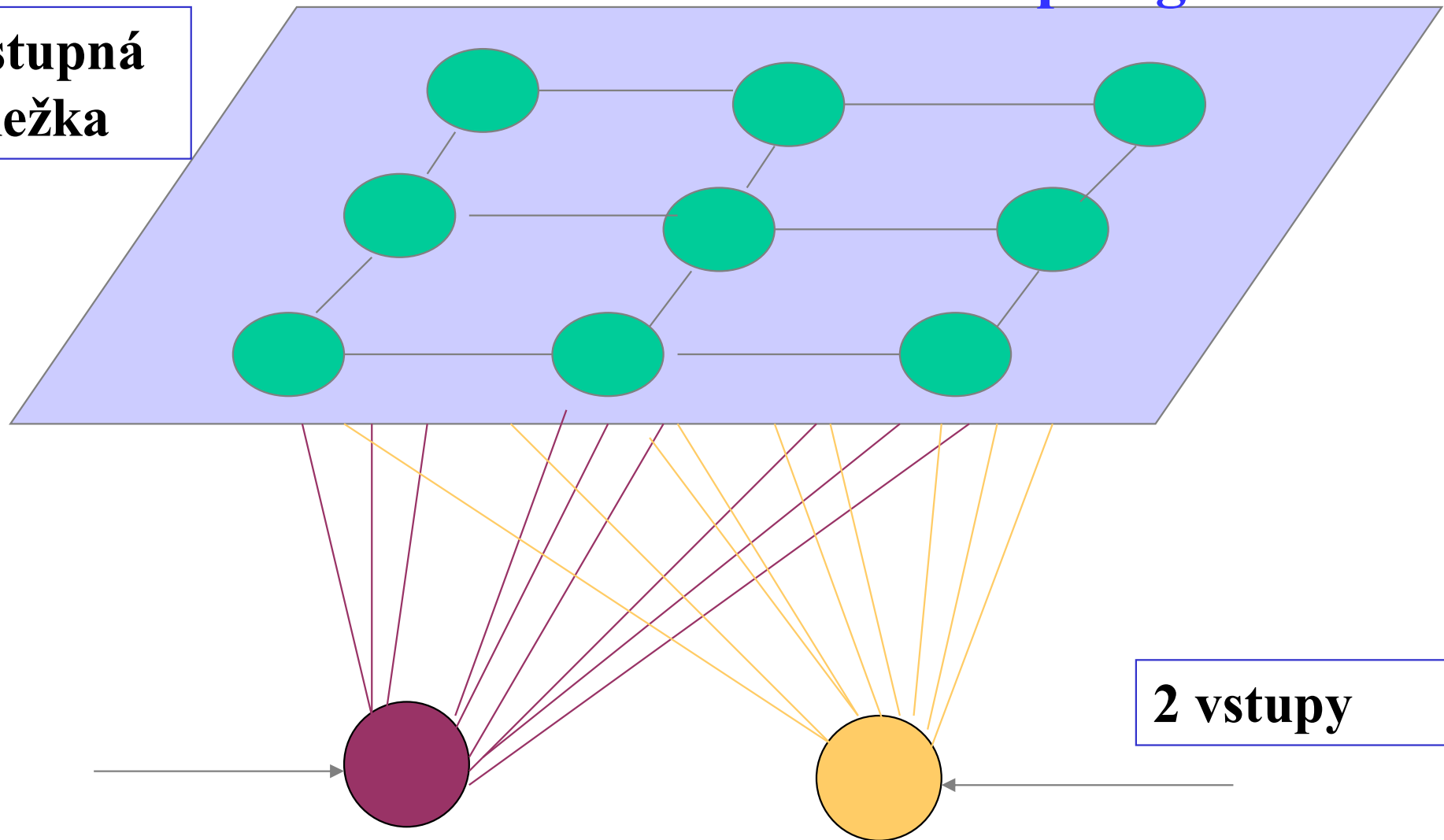
- Počet vstupov do neurónu je rovný počtu vstupov do Kohonenovej siete
- váhy vstupov slúžia k zakódovaniu vzorov
- neuróny nemajú prenosovú funkciu
- každý neurón je priamo aj výstupom (výstupné neuróny nie sú vzájomne prepojené, prepojenie je len so susednými neurónmi)
- počet výstupov je rovný počtu neurónov
- *Matematicky*, každý neurón počíta odchýlku predložených vzorov od vzoru zakódovaného vo váhach

$$d = \sum_{i=0}^{N-1} [x_i(t) - w_i(t)]^2$$

- počet neurónov je funkciou aplikácie (od 10 x 10 až do 30 x 30 neurónov)

Kohonenova sieť - topológia

Výstupná
mriežka



2 vstupy

- algorithmus učenia má za cieľ usporiadať neuróny v mriežke do určitých oblastí tak, aby boli schopné klasifikovať predložené údaje.
- proces učenia je adaptívny, iteračný výpočtový proces

Algoritmus učenia KS

- 1.krok : Inicializácia-nastavenie váh w_{ij} , pre všetky spoje z N vstupov do M výstupných neurónov

$$0 \leq i \leq N - 1$$

$$0 \leq j \leq M - 1$$

počiatočné hodnoty váh nastavujeme napr. náhodným algoritmom (rands)

- parametre učenia $\eta(0)$ (rýchlosť učenia) nastavíme na začiatku na hodnotu blízku 1, pričom

$$0 \leq \eta(t) \leq 1$$

- 2.krok predloženie vzorov na vstup $X(t) = \{x_0(t), x_1(t), \dots, x_{N-1}(t)\}$

- výpočet vzdialeností vzorov medzi predloženým vzorom a všetkými výstupmi

$$d_j = \sum_{i=0}^{N-1} [x_i(t) - w_{ij}(t)]^2$$

kde $x_i(t)$ sú prvky vstupného vzoru a w_{ij} sú váhy medzi i -tým vstupom a j -tým výstupným neurónom v ktorých sú zakodované vzory

4. Krok: Výber najviac podobného neurónu podľa vzťahu

$$d_j^* = \min_j(d_j)$$

5.Krok: Modifikácia - Adaptácia váh, korekcia váh pre j^* -ty neurón a jeho okolie (susediace neuróny) podľa vzťahu:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)[x_i(t) - w_{ij}(t)]$$

kde j sú všetky neuróny ležiace v okolí $N_j^*(t)$

6. Krok Pokračovanie- všetkými vzormi na vstupe, všetkými tréningovými krokmi, test podľa presnosti, *if áno then krok 2 else end*

Poznámka:

- počiatočné nastavenie váh sa realizuje náhodne-každý neurón má svoj vlastný originálny váhový vektor
- koeficient učenia (rýchlosť učenia) sa najprv nastavuje na hodnotu jedna a behom procesu učenia klesá k nule(lineárne, exponenciálne, hyperbolic.)
- voľba okolia - pre každý neurón na začiatku sú to všetky neuróny v mriežke, postupne sa okolie redukuje, minimálne okolie je jeden neurón

- víťazným neurónom je ten, ktorého váhový vektor odpovedá najviac vstupnému vzoru, pre takýto neurón sa budú adaptovať váhy
- výberom určitých víťazných neurónov - zvýšenie citlivosti reagovať aj na nenatréňovaný vzor-princíp zovšeobecnenia

Prevádzkovanie KS po natréňovaní

1. Krok - vstup nového neznámeho vzoru na vstupy do siete

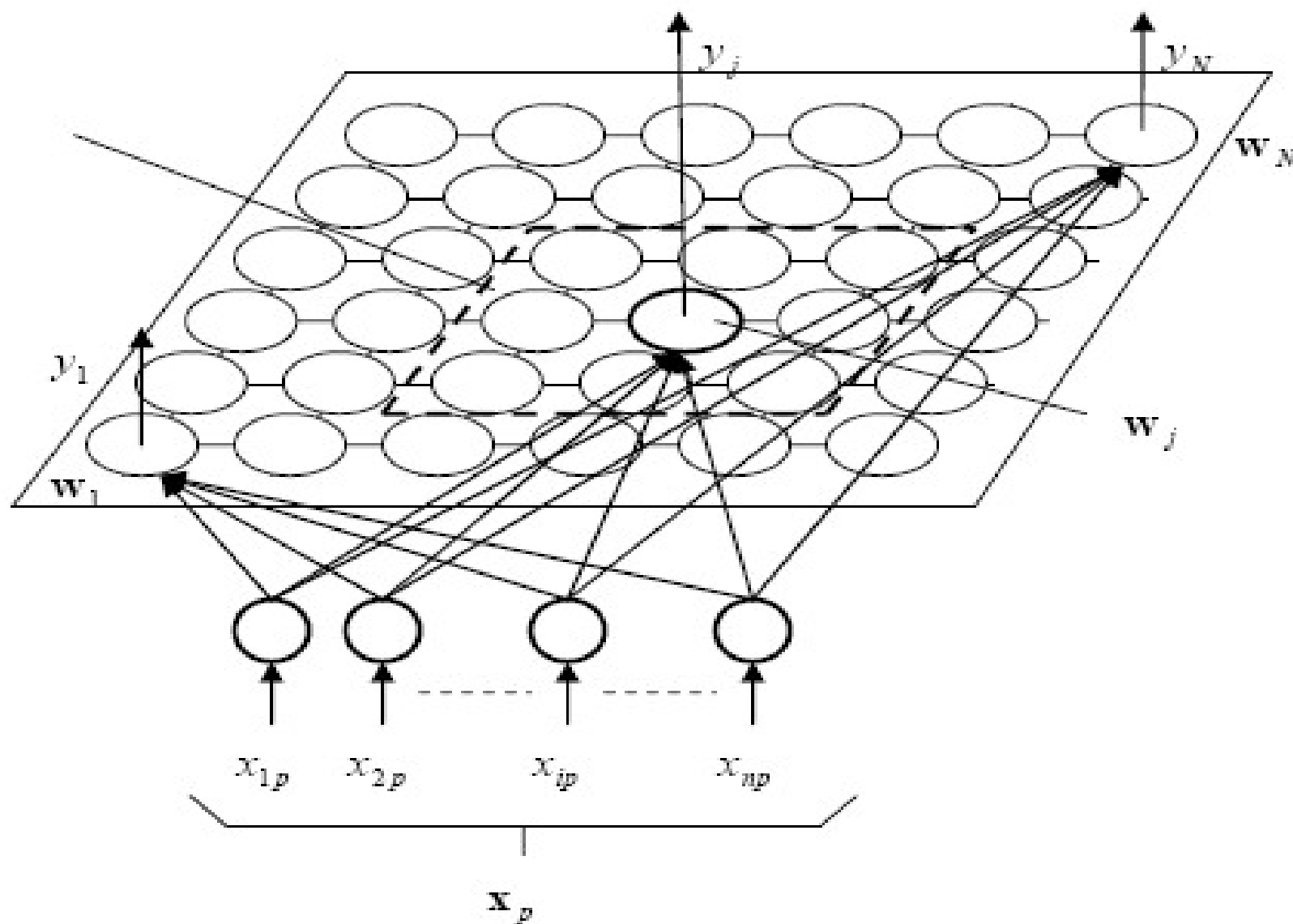
$$X(t) = \{x_0, x_1, \dots, x_{N-1}\}$$

2. Krok - výpočet najbližšieho vzoru pomocou vzťahu

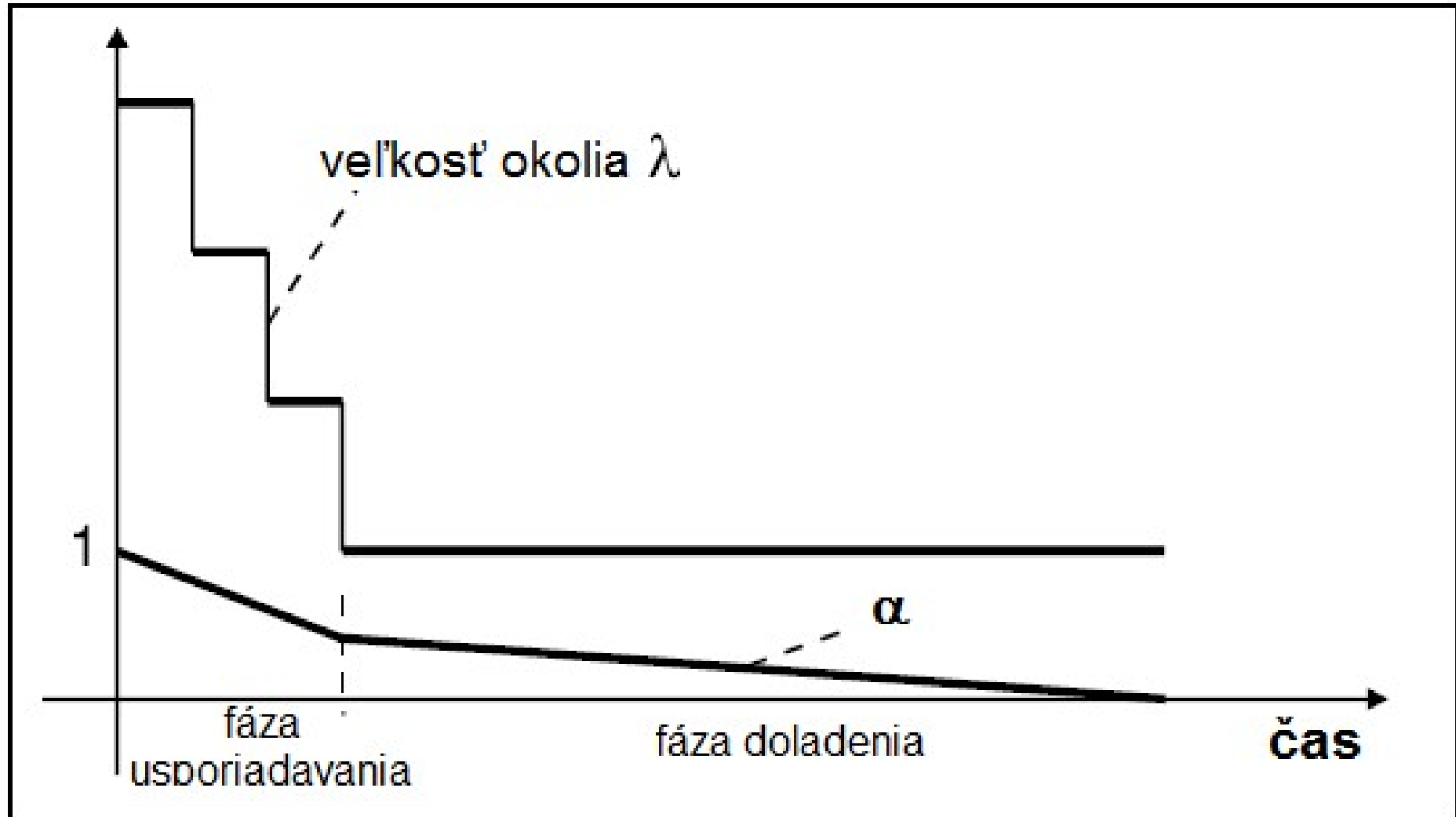
$$d_j = \sum_{i=0}^{N-1} [x_i(t) - w_{ij}(t)]^2$$

a výber takého neurónu, ktorý splňa podmienku $d_j^* = \min_j (d_j)$

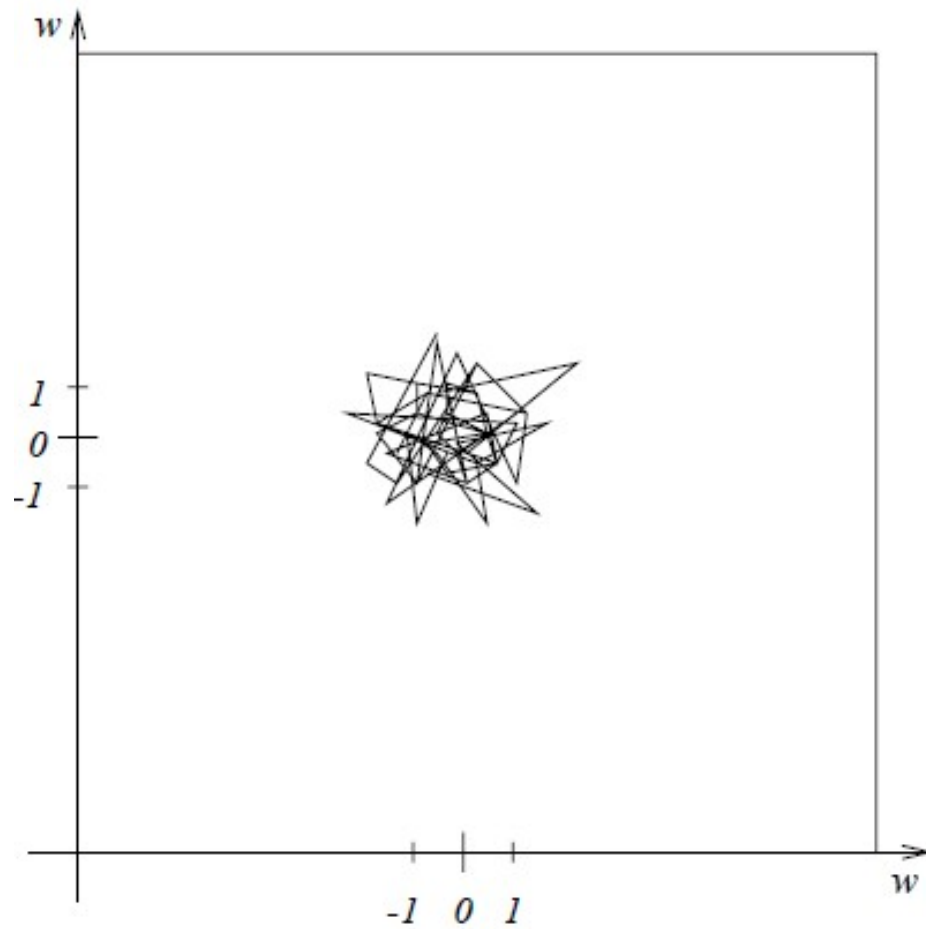
Kohonenova sieť – štruktúra s viacerými vstupmi



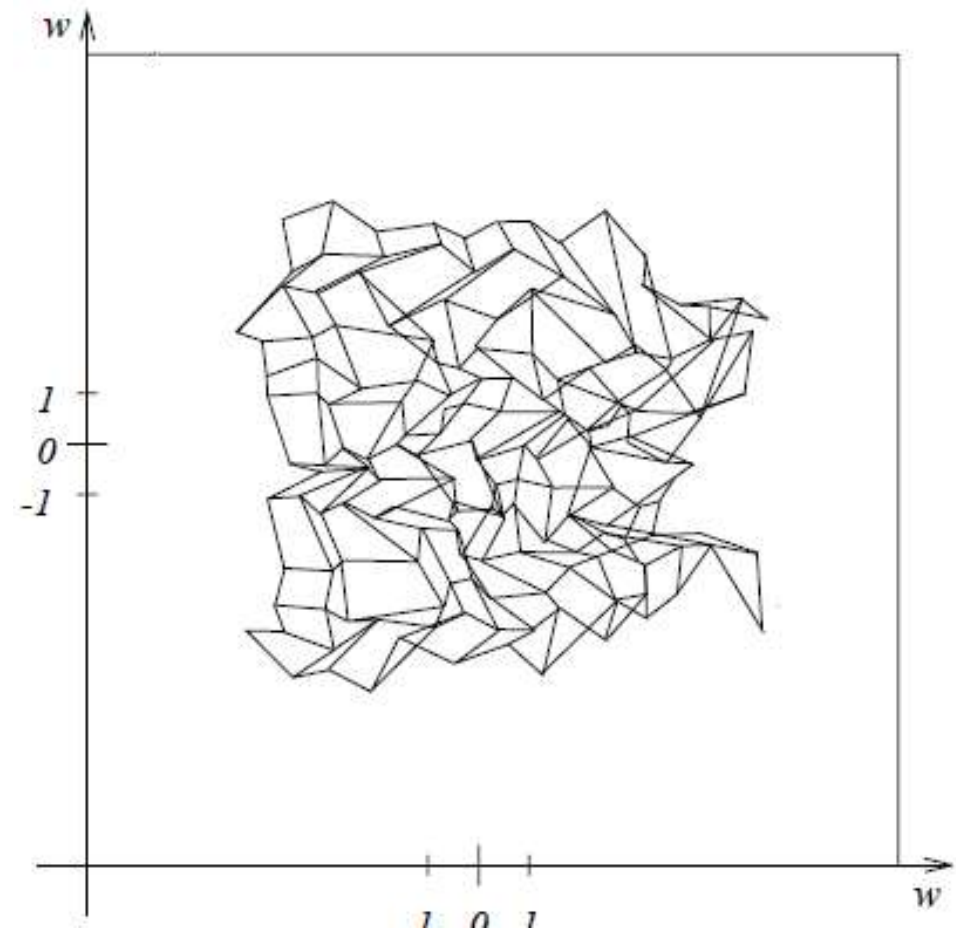
Kohonenova sieť – redukcia kroku učenia a okolia



Kohonenova sieť – priebeh učenia



Inicializácia váh:



Priebeh učenia:

Kohonenova sieť – regióny pre víťazný neurón

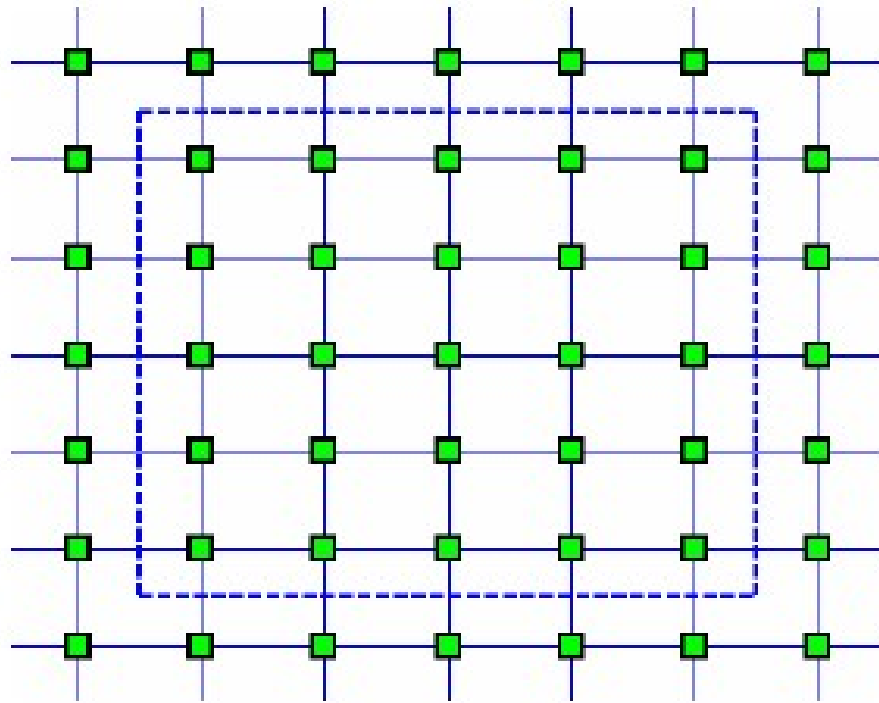


Figure 2: Square surroundings of the winning neuron, $R_c = 2$

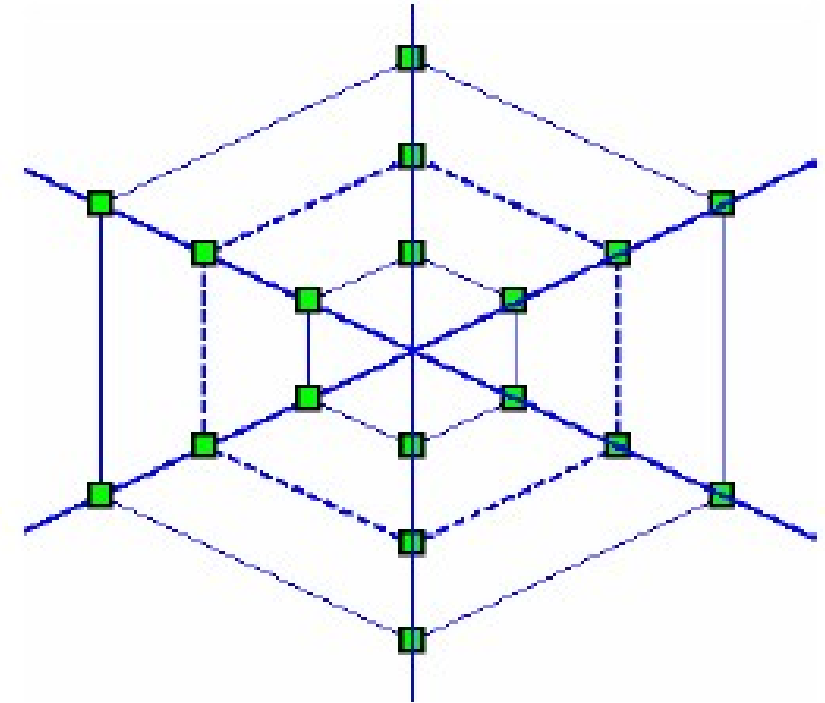


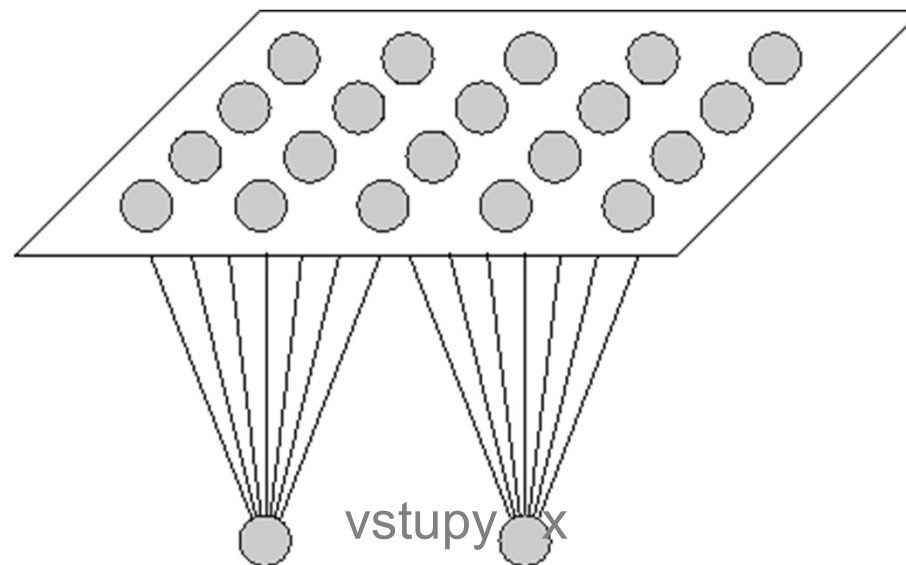
Figure 3: Hexagon surroundings of winning neuron, $R_c = 1$

Kohonenove siete

Kohonenove siete predstavujú veľmi dôležité rozšírenie konkurenčného učenia. Biologické systémy majú podobný tvar hlavne vo svojej percepčnej časti - oku. Rozšírenie konkurenčného učenia spočíva v tom, že sa pripúšťa princíp **viacerých víťazov - multiply WTA**.

Ďalej výstup z takejto NN je geometricky usporiadaný do nejakého útvaru napr. vedľa seba, kruhu, obdĺžníka, a teda existuje možnosť najdenia **suseda**. Túto vrstvu nazývame **Kohonenovou vrstvou**.

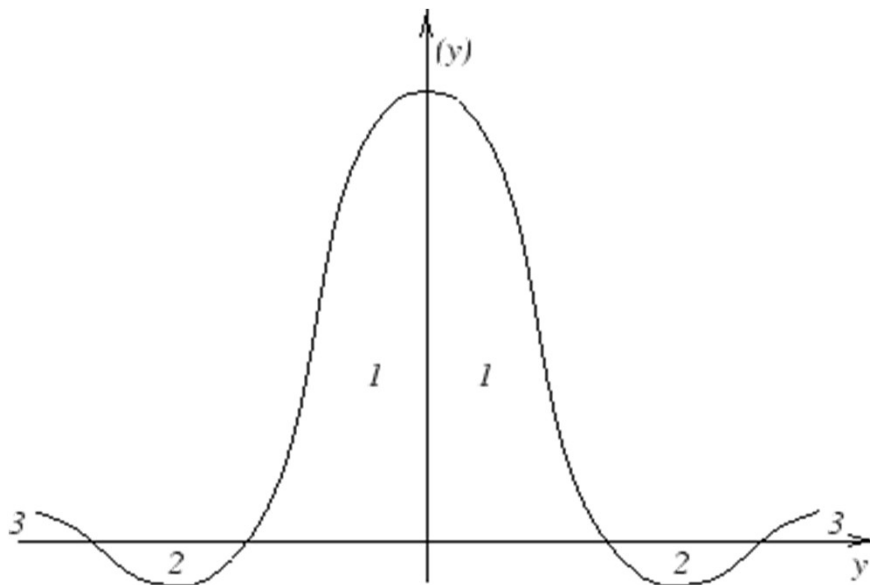
Obrázok: Typická topológia Kohonenovej NN



Zhlukovanie sa organizuje takým spôsobom, že susedné neuróny resp. synaptické váhy (SV), ktoré k nim smerujú, majú podobné hodnoty a tie, ktoré sú ďalej od seba, majú viac rozdielne hodnoty. Príčinou toho celého je zavedenie tzv. **susednej funkcie**. Vo väčšine prípadov ide o funkciu v tvare :

$$\lambda(j^*, j) = h(t) \cdot \exp\left(\frac{-d(j^*, j)}{r(t)}\right)$$

kde ***h*** je adaptačná výška, ***d*** je vzdialenosť medzi neurónmi v Kohonenovej vrstve a ***r*** predstavuje polomer priestorového susedstva v iterácii. Takáto funkcia môže byť vyjadrená v tvare **mexického klobúka**, ako je uvedené na obr. [5.5](#).



Obrázok: Možný tvar funkcie susednosti

Vzt'ah pre adaptáciu váh:

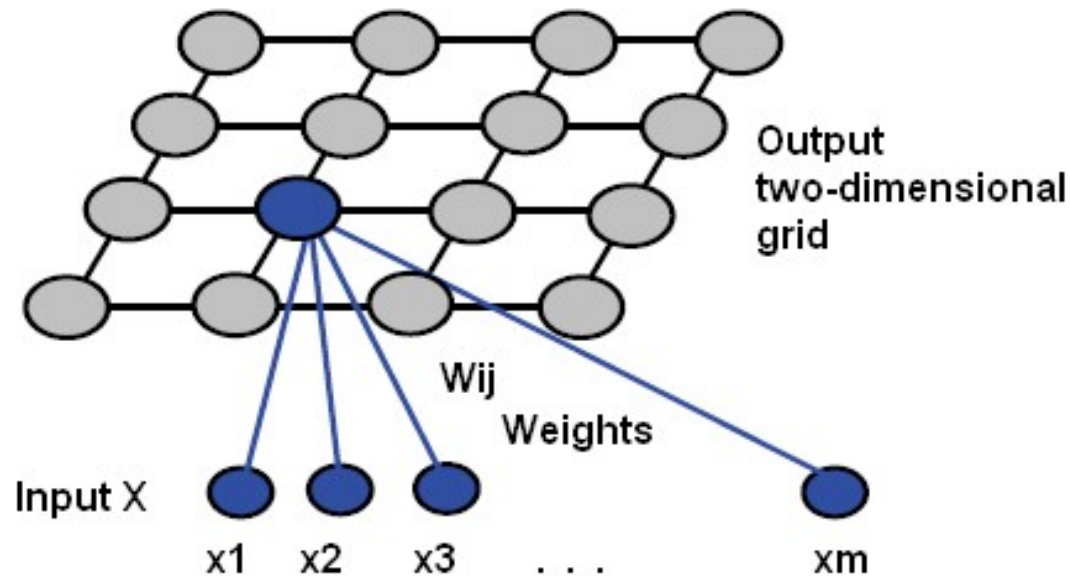
$$w_{ij}(t+1) = w_{ij}(t) + \eta(t) \cdot \lambda(j^*, j) [x_i(t) - w_{ij}(t)]$$

kde η je koef. učenia a λ je hodnota funkcie susednosti

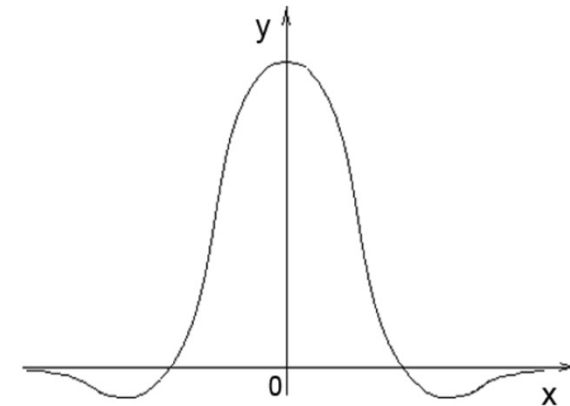
- Optimalizácia polohy distribučných centier
- Zatriedenie bytov do skupín v závislosti od (lokality, počet izieb, cena, zariadenie, ...)
- Zatriedenie a vyhľadávanie dokumentov, príspevkov, inzerátov ...
- Automatická kontrola pravopisu
- Asociácie výrokov (podmienka -> akcia)
- Rozpoznanie obrazcov
- Získavanie informácií - Data mining

Príklad 1 – klasifikácia do skupín

Štruktúra Kohonenovej siete pre klasifikáciu

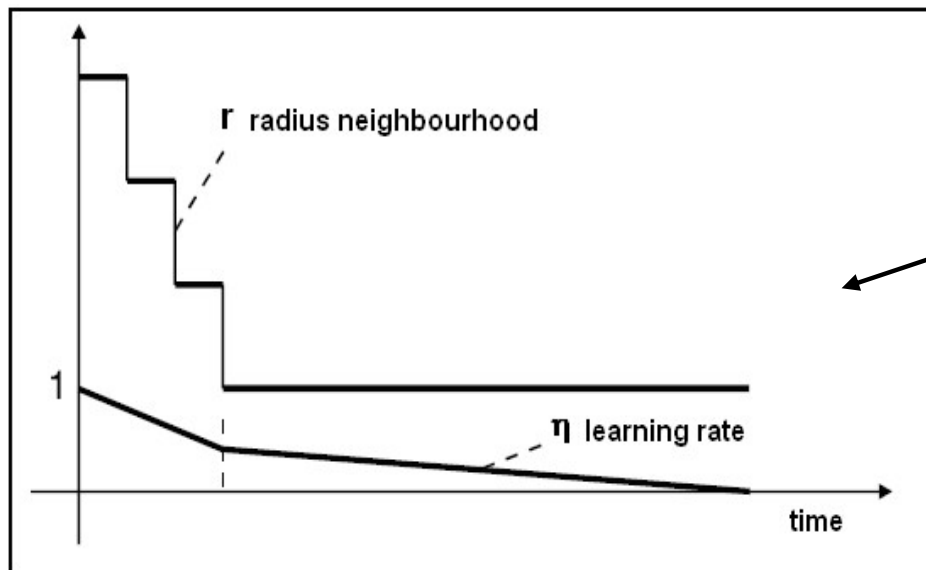


Funkcia susednosti



$$\lambda(j^*, j) = h(t) \cdot \exp\left(\frac{-d_E(j^*, j)}{r(t)}\right)$$

d_E – vzdialenosť neurónu od víťazného v štruktúre siete

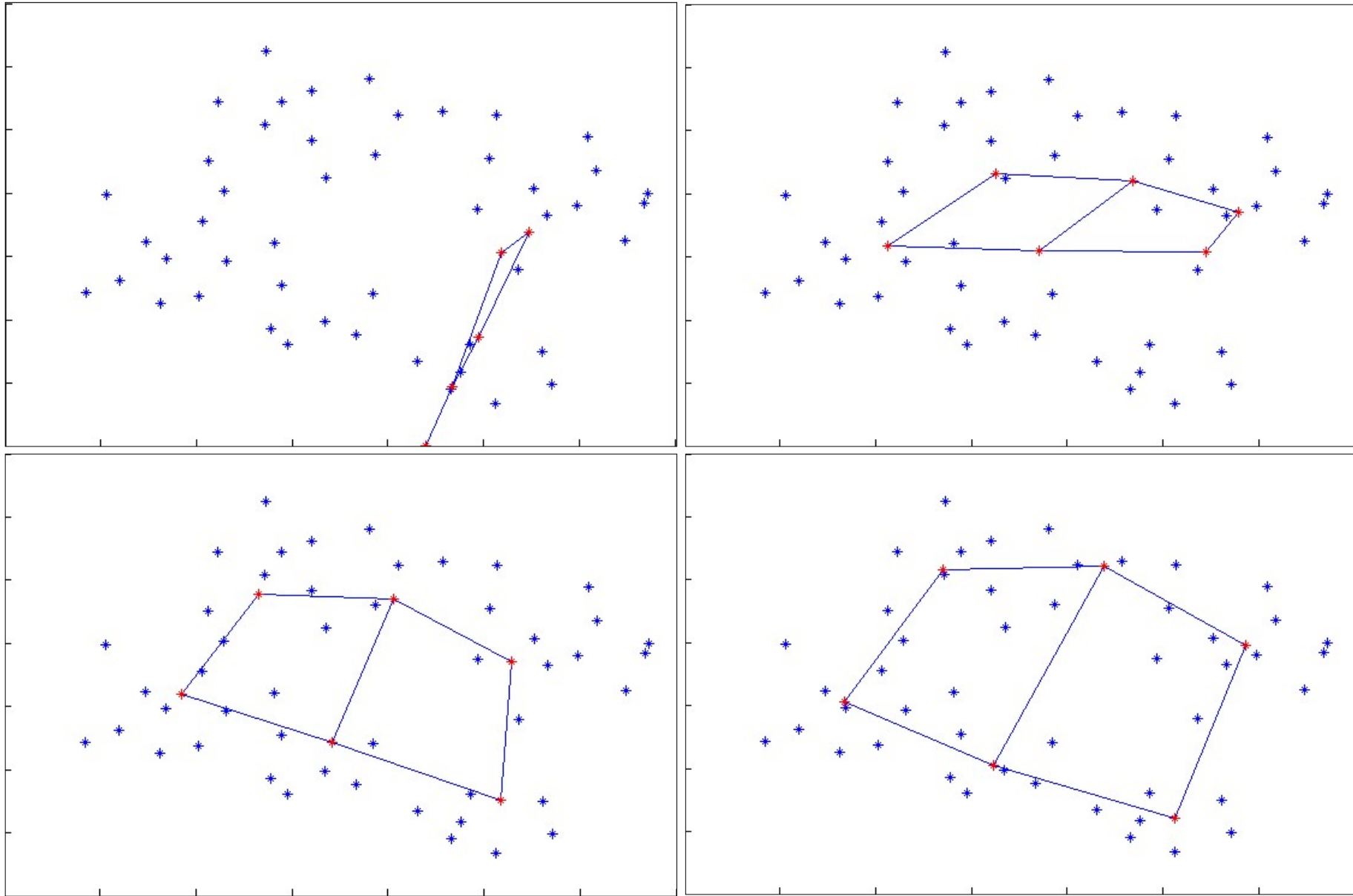


Úprava kroku učenia η a polomeru funkcie susednosti r

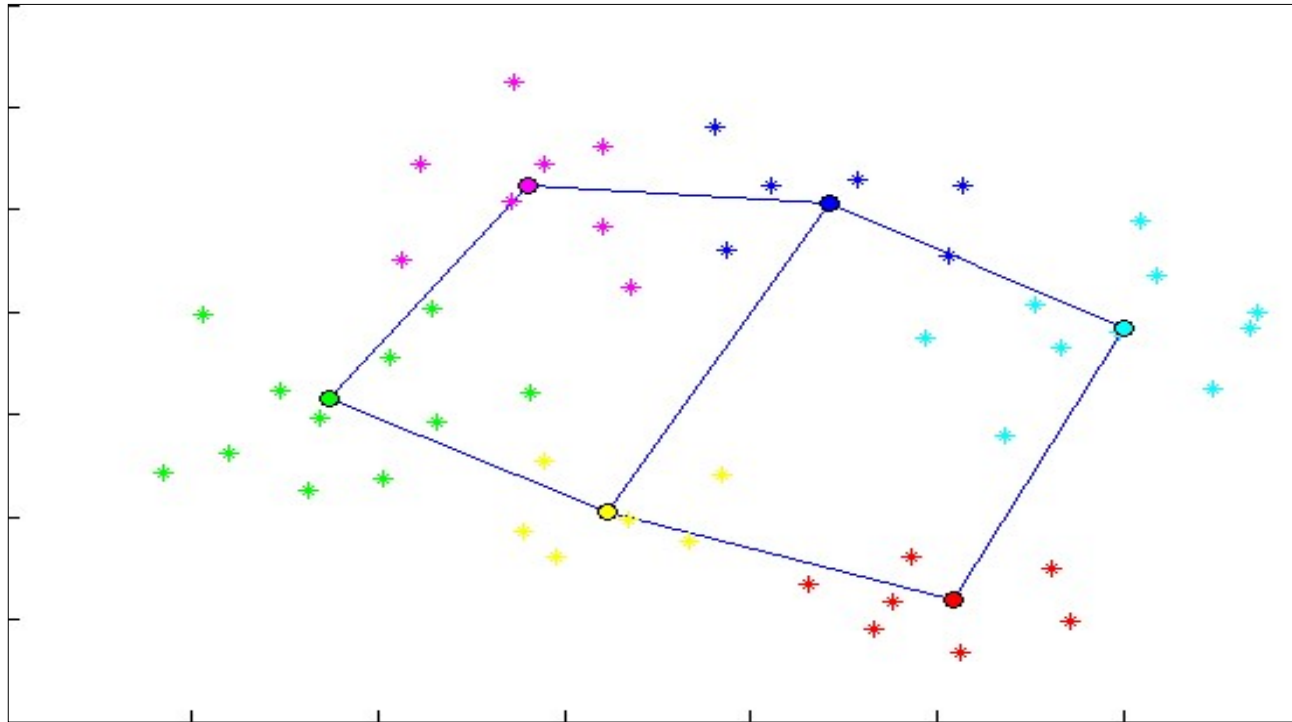
Adaptácia váh

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t) \cdot \lambda(j^*, j) [x_i(t) - w_{ij}(t)]$$

Príklad 1 – klasifikácia do skupín



Príklad 1 – klasifikácia do skupín

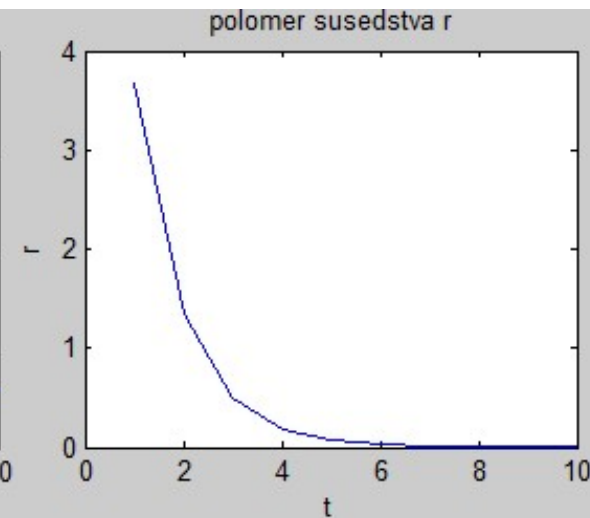
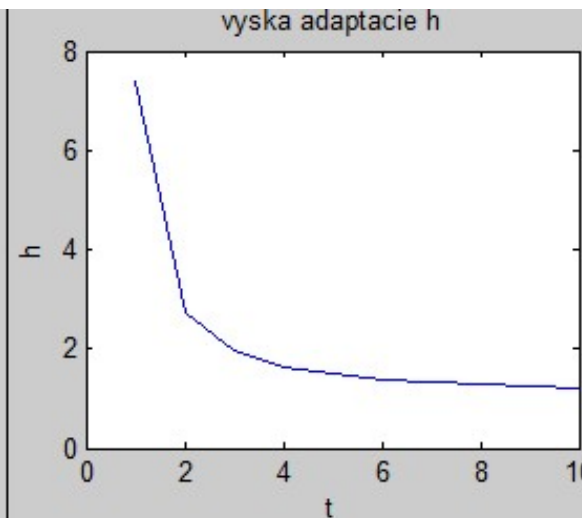
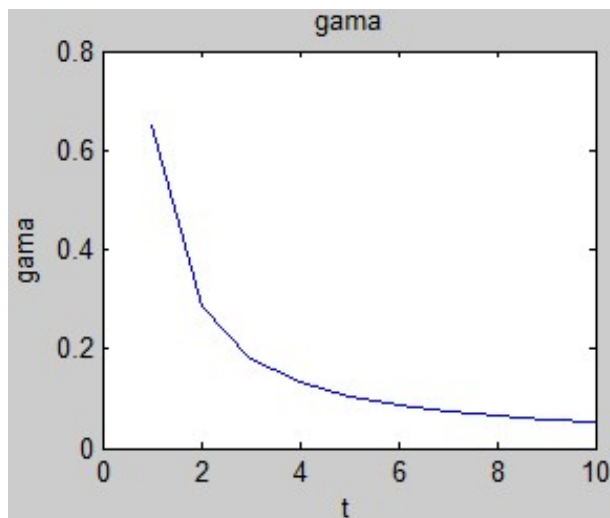


Počet cyklov (cycle)	8
Zač. krok učenia η_0	0,7
Polomer susedstva r_0	10
Výška funkcie susedstva h_0	1,2

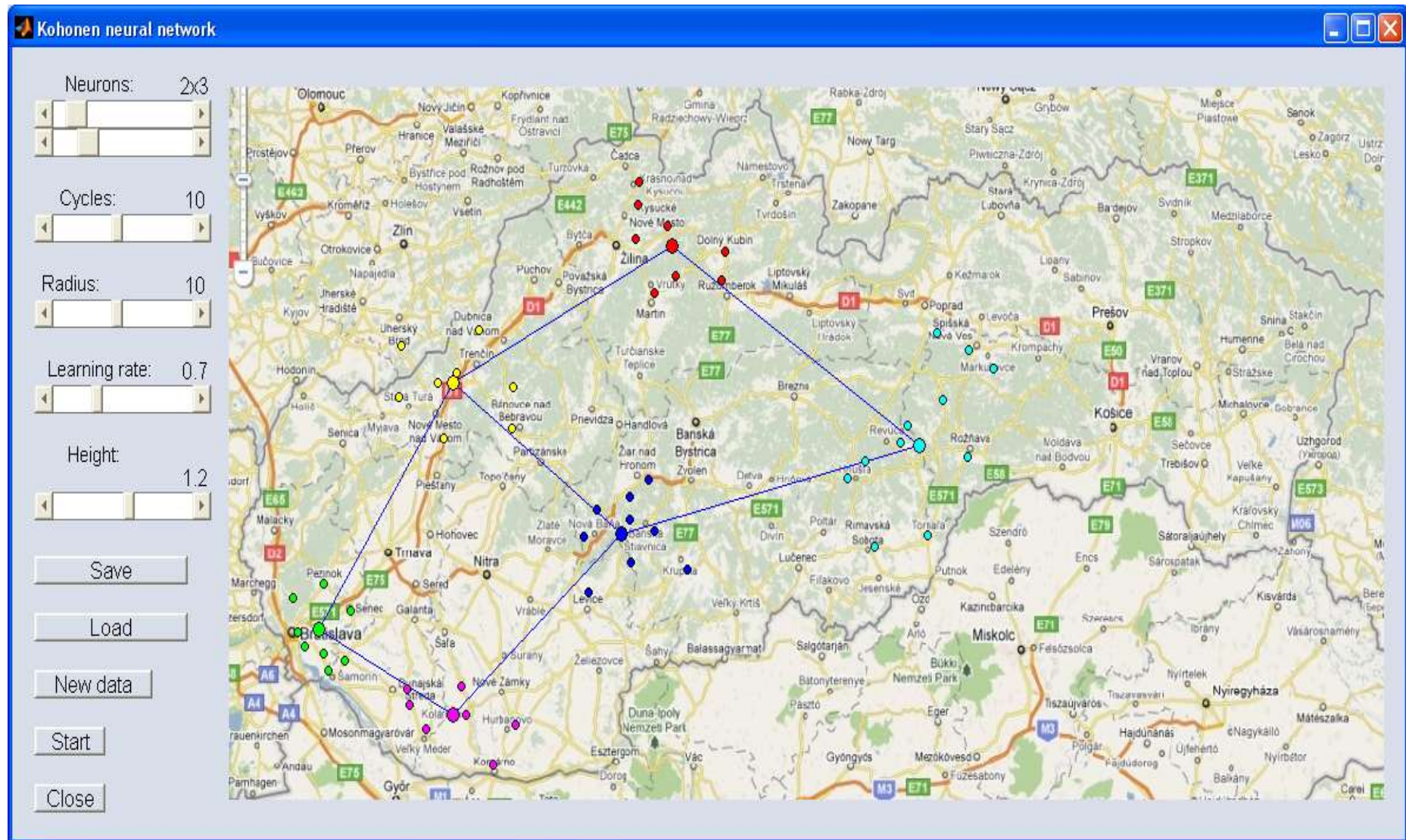
$$\eta(t) = \exp(\eta_0 / \text{cycle}) - 1$$

$$h(t) = \exp((1 - h_0) / \text{cycle})$$

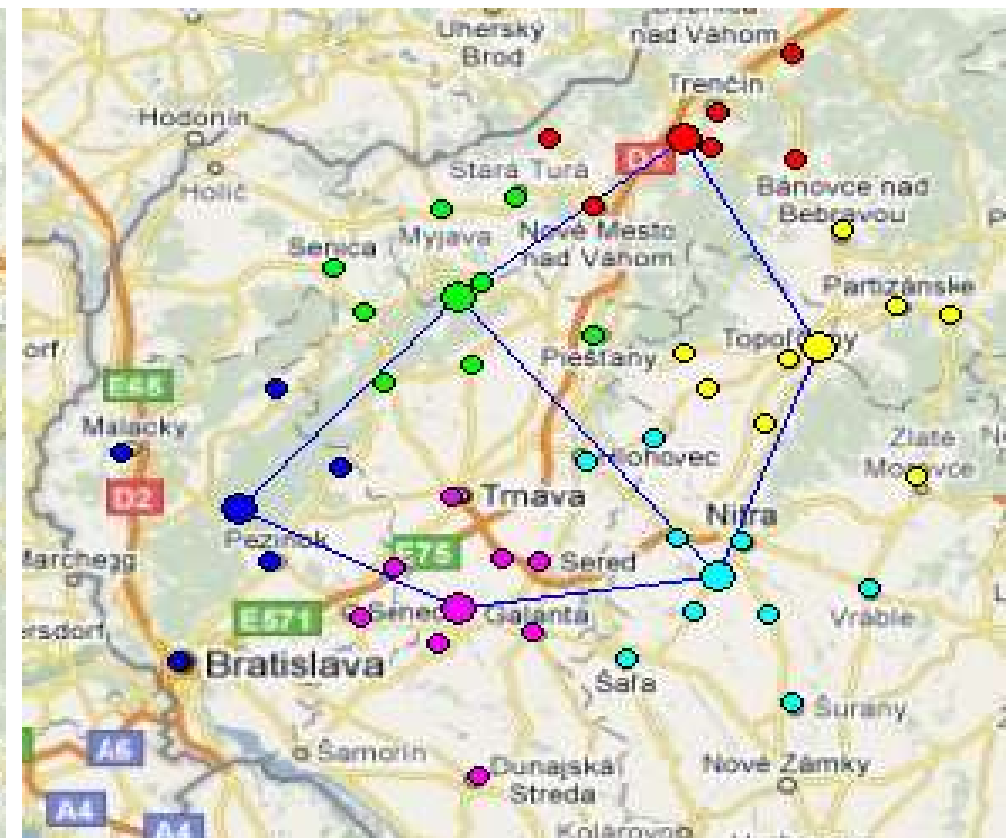
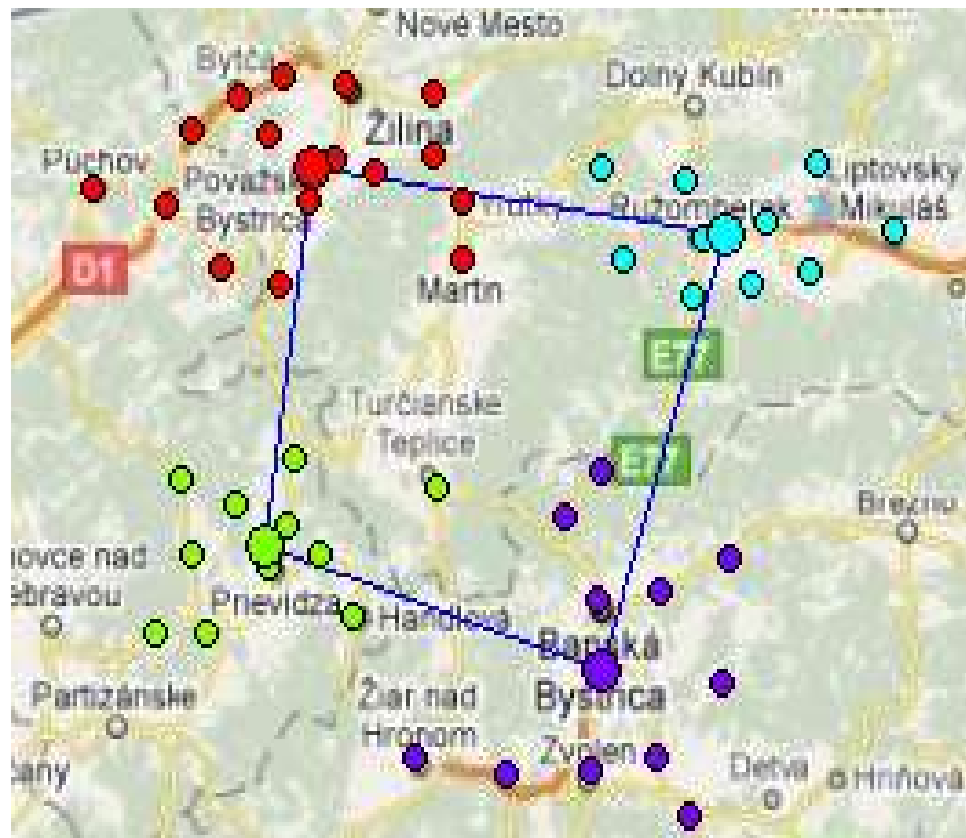
$$r(t) = r_0 \cdot \exp(-\text{cycle})$$



Príklad 2 – optimalizácia rozlož. distribučných centier

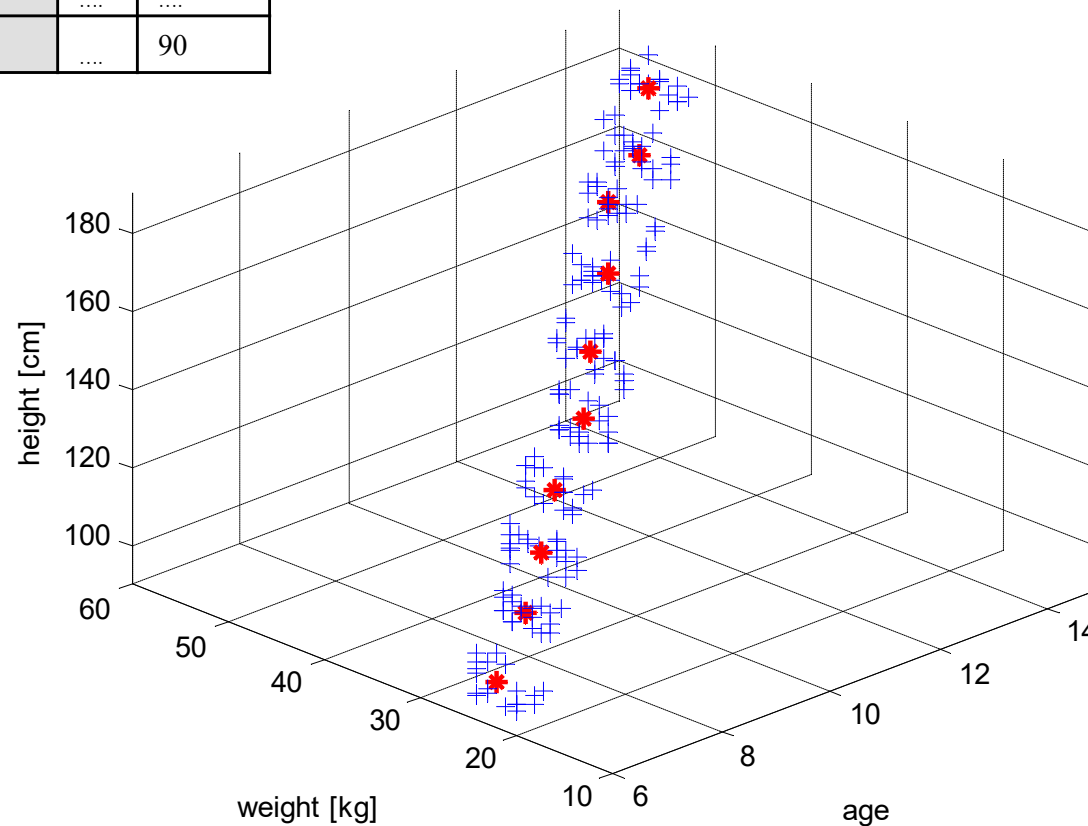


Príklad 2 – optimalizácia rozlož. distribučných centier



Príklad 3 – triedenie dát do skupín v databáze

No.	Name	Sex	Age	Weight	Height	Sport
1	Fero Mrkva	M	8	30	125	95
2	Anna Pekná	F	10	32	145	30
3	Marta Sivá	F	9	28	142	80
4	Peter Veľký	M	13	45	162	80
5	Dušan Malý	M	12	42	158	60
....
N	Pavol Ondava	M	10	38	141	90



Príklad 4 – triedenie ľudí podľa záujmov

- Prvé dva atribúty človeka, s ktorými sieť pracuje, sú *vek* (0-0.6) a *pohlavie* (0-0.2).
- Vlastnosti, ktoré vyjadrujú samotnú ľudskú osobnosť, sú *ambicióznosť*, *vízia sveta*, *nálada*, *temperament*, *športový duch* (0-1) (ďalej len povahové črty)
- Hľadá aj optimálnu štruktúru (počet neurónov – skupín)

Príklad 4 – triedenie ľudí podľa záujmov

Administrátor: Miloš Lajtman

Počet skupín: 3
Kritériálna funkcia (Jp): 20.34
Absolútna vzdialenosť (J): 15.02

Priemerný člen skupiny

Vek:
32

Pohlavie:
Muž

Skupina: 1 / Odchýlka:

Ambicióznosť:
26%

Vízia sveta:
29.8%

Naladenie:
35.4%

Temperament:
30.8%

Športová aktivita:
28%

Min. počet neurónov
2

Percentuálna odchýlka J
1.6

Váhovacia konštanta v krit. funkcii Jp (alfa)
1.5

Výška funkcie susedstva (h)
0.9

Polomer funkcie susedstva (r)
8

Krok/rýchlosť učenia (gama)
0.7

Počet opakovaní tréningu
3

Počet cyklov učenia
5

Príklad 5 – triedenie nehnuteľností v databáze

- Atribúty, ktoré špecifikujú určitú nehnuteľnosť, sú nasledovné: *typ, cena, počet izieb, rozloha, vek, stav a lokalita*. (0-1).

id	foto	adresa	psc	typ	cena	izby	rozloha	vek	stav	lokalita	cluster	variation
155	f39.jpg	Laurinská 209/6, 811 01 Bratislava	811 01	0.6	78800	4	90	3	0.85	1	3	0.66
154	h38.jpg	Mierová 2160/224, 821 05 Bratislava	821 05	0.8	149200	8	173	4	0.77	0.7	5	0.42
153	f37.jpg	Tulipánová 2197/14, 841 01 Dúbravka	841 01	0	131200	4	95	2	0.18	0.6	2	0.41
152	f36.jpg	Starhradská 1710, 851 05 Petržalka	851 05	0.4	75600	4	84	25	0.57	0.4	1	0.24
151	f35.jpg	Bieloruská 5193/46, 821 06 Bratislava	821 06	0.6	100500	5	85	25	0.13	0.2	3	0.57
150	f34.jpg	Haanova 20-22, 851 04 Bratislava-Petržalka	851 04	0.2	90000	4	83	9	0.25	0.45	3	0.34
149	f33.jpg	Mokrán záhon 16434, 821 04 Bratislava	821 04	0.4	89600	4	71	2	0.08	0.65	3	0.35
148	f32.jpg	Krajná 700, 900 41 Rovinka	900 41	0	114200	2	104	50	0.7	0	1	0.77
147	f31.jpg	Beniakova 3082/30, 841 05 Karlova Ves	841 05	0.6	88300	3	70	4	0.57	0.7	3	0.32
146	h30.jpg	Čelakovského 1395/14, 811 03 Bratislava	811 03	0.8	146800	8	172	12	0.33	1	5	0.36
145	h29.jpg	Vtáčnik 11534/3, 831 01 Bratislava	831 01	0.8	123500	5	137	7	0.21	0.9	5	0.41
144	h28.jpg	Zhorinska 1405/48, 841 03 Bratislava	841 03	0.8	135600	6	174	12	0.48	0.65	5	0.25
143	f27.jpg	Podlučinského 3185/7, 821 04 Bratislava	821 04	0.2	83800	4	70	40	0.08	0.65	3	0.64

Príklad 5 – triedenie nehnuteľností v databáze

Administrátor: Administrátor

Počet skupín: 5
Kritériálna funkcia (Jp): 70.6
Absolútna vzdialenosť (J): 42.18

Min. počet neurónov	Polomer funkcie susedstva (r)
<input type="text" value="2"/>	<input type="text" value="8"/>
Percentuálna odchýlka J	Krok/rýchlosť učenia (gama)
<input type="text" value="1.3"/>	<input type="text" value="0.7"/>
Váhovacia konšt. v krit. funkcii Jp (alfa)	Počet opakovaní tréningovania
<input type="text" value="1.5"/>	<input type="text" value="3"/>
Výška funkcie susedstva (h)	Počet cyklov učenia
<input type="text" value="0.9"/>	<input type="text" value="5"/>

Typ nehnuteľnosti:	Rodinný dom
Počet izieb:	6
Rozloha nehnuteľnosti:	126 m²
Vek nehnuteľnosti:	14.6 rokov
Stav nehnuteľnosti:	50.9%
Skupina:	4

113 973 €

Príklad 6 – zatriedenie farieb domov do skupín

