Radial Basis Function (RBF) siete

Príklady RBF

- Gaussova:

$$\varphi(r) = e^{-(\varepsilon . r)^2}$$

- Multikvadratická:

$$\phi(r) = \sqrt{1 + (\varepsilon . r)^2}$$

- Inverzná kvadratická:

$$\phi(r) = \frac{1}{1 + (\varepsilon . r)^2}$$

$$r = \| x - x_i \|$$

Definícia:

- Funkcia je radial symetrická (alebo je RBF), ak jej výstup závisí od vzdialenosti medzi vstupným vektorom a vektorom zapamätaným v tejto funkcii
  - Napríklad,

Vzdialenosť $r = \|x - x_i\|$ kde x je vstupný vektor, x_i je vektor

asociovaný s RBF

  - Pre výstup platí

$$\phi(r_1) \geq \phi(r_2), \text{ ak } r_1 < r_2$$

Príklady RBF

- Inverzná multikvadratická: $$\phi(r) = \frac{1}{\sqrt{1+(\varepsilon.r)^2}}$$
- Polyharmonický spline:
$$\phi(r) = r^k, k = 1,3,5,\ldots$$
$$\phi(r) = r^k \ln r, k = 2,4,6,\ldots$$
- Tenký spline (špeciálny polyharmonický spline):
$$\phi(r) = r^2 \ln r$$

Vlastnosti Gaussovej funkcie

$$\varphi(r) = e^{-(\varepsilon . r)^2}$$

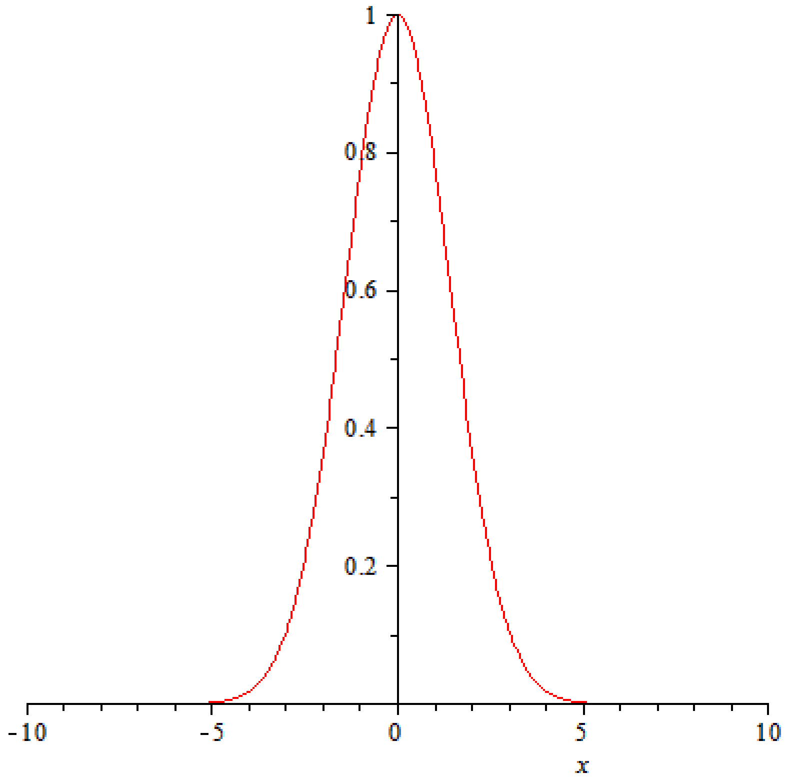
$$r = \| x - x_i \|$$

- **V 1-rozmernom priestore**
- Symetrická funkcia s centrom (so stredom) v x_i
- r nadobúda rovnaké hodnoty pre body rovnako vzdialené od centra

Gaussova RBF

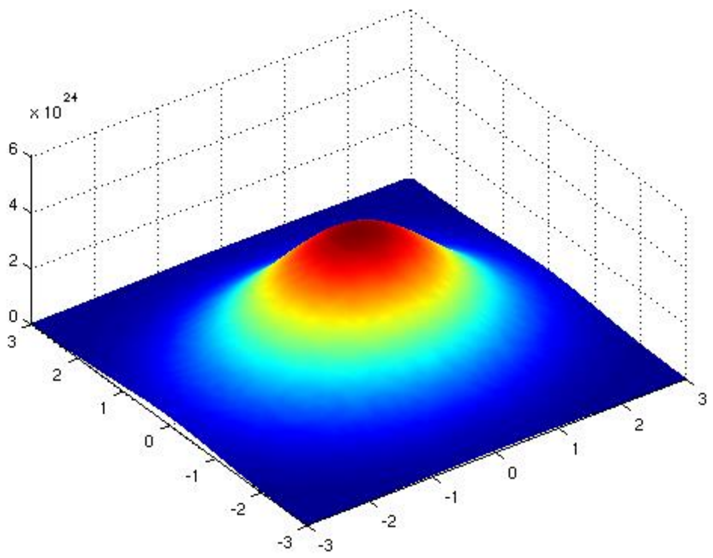
- Gaussova RBF monotónne klesá vzdialenosťou od centra
- Gaussove RBF sú lokálne (najväčšie odozvy dávajú v okolí centra)
- sú vo všeobecnosti najviac používané

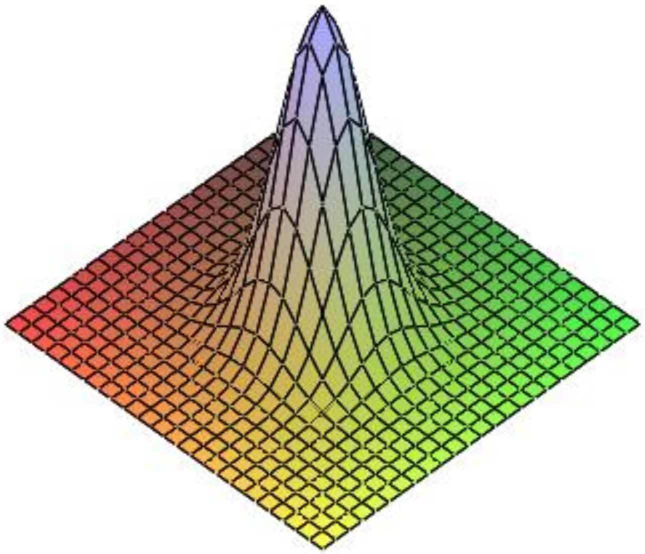
$$\phi(x) = \frac{1}{e^{(\varepsilon . x)^2}}$$



Vlastnosti Gaussovej funkcie

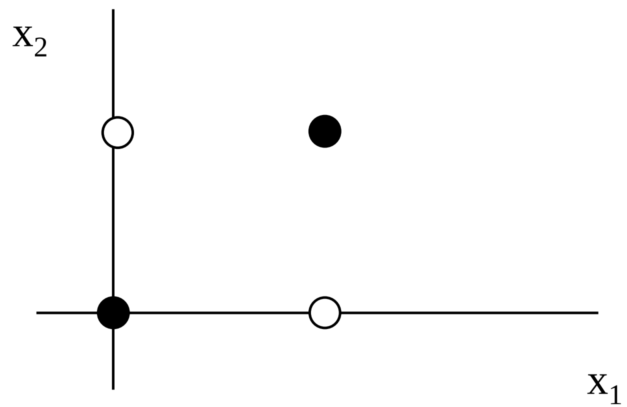
- V 3-rozmernom priestore

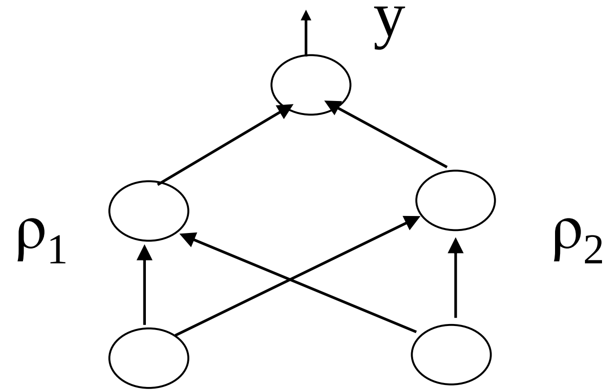




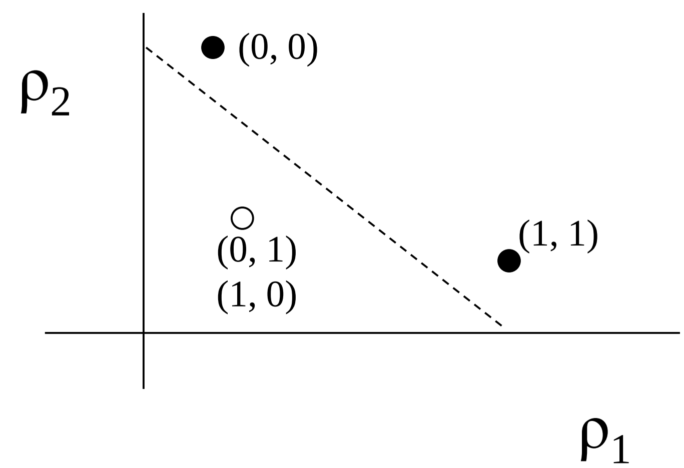
Motivácia: XOR funkcia

XOR





x	$\rho_1(x)$	$\rho_2(x)$
(1,1)	1	0.1353
(0,1)	0.3678	0.3678
(0,0)	0.1353	1
(1,0)	0.3678	0.3678



Radial Basis Function siete

RBF sieť je neurónová sieť navrhnutá tak, aby riešila problém hľadania čo najlepšej krivky - aproximačný problém vo vysoko dimenzionálnom priestore

Učenie je ekvivalentné hľadaniu takej funkcie viacerých premenných, ktorá sa najlepšie zhoduje s tréningovými dátami

Neurónové siete s Radial Basis Funkciami

Lepšia výkonnosť než u sigmoidálnych funkcií

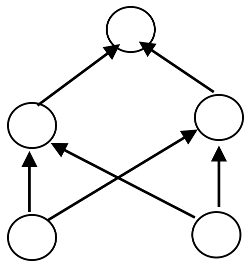
- Niektoré klasifikačné problémy
- Interpolácia funkcií

- **Ešte raz definícia:**
  - Funkcia je radial symetrická (alebo je RBF), ak jej výstup závisí od vzdialenosti medzi vstupným vektorom a vektorom zapamätaným v tejto funkcii
    - Vzdialenosť $r = \|x - x_i\|$ kde x je vstupný vektor, x_i je vektor asociovaný s RBF
    - Výstup $\phi(r_1) \geq \phi(r_2)$, ak $r_1 < r_2$

Neurónové siete s Radial Basis Funkciami

- XOR problém
  - 2-2-1 sieť
    - 2 skryté neuróny sú RBF
    - Výstupný neurón môže byť sign alebo sigmoidálny
    -

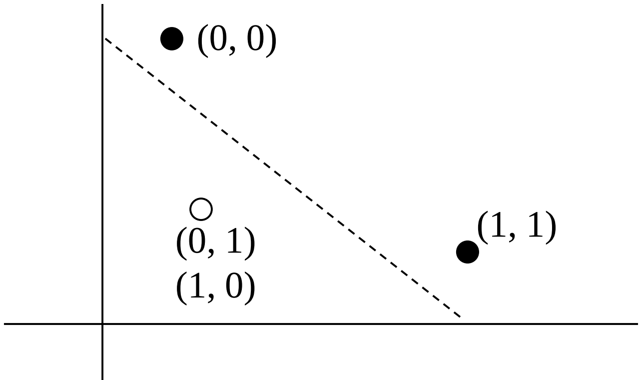
$$\rho_1(x) = e^{-\|x-t_1\|^2}, \; t_1 = [1,1]$$
$$\rho_2(x) = e^{-\|x-t_2\|^2}, \; t_2 = [0,0]$$



x	$\rho_1(x)$	$\rho_2(x)$
(1,1)	1	0.1353
(0,1)	0.3678	0.3678
(0,0)	0.1353	1
(1,0)	0.3678	0.3678

    - Ak je na vstupe *x:*
    - Skryté neuróny vypočítajú vzdialenosti a potom je vypočítaný výstup

$$\left\| x - t_j \right\|$$

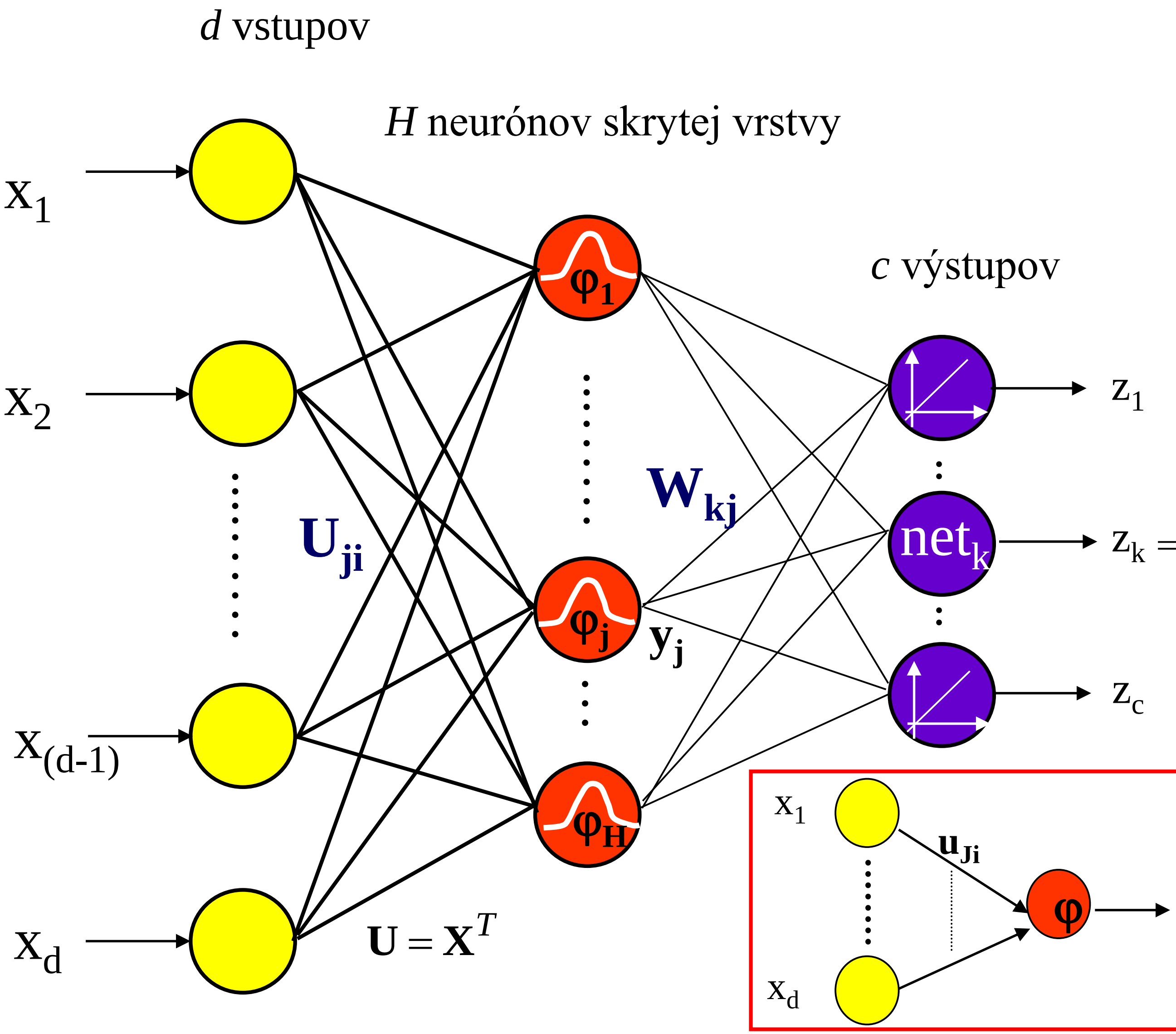


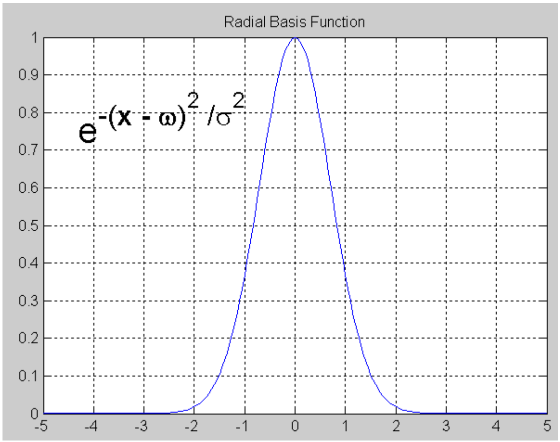
    - Všetky váhy do skrytých neurónov sú nastavené na 1
    - Váhy do výstupného neurónu sú trénované metódou Least Mean Squared Errors
    - t_1 a t_2 môžu byť tiež trénované

Popis RBF sietí

- Pozostáva z 3 vrstiev (vstupná, skrytá, výstupná)
- **Vstupná vrstva** je tvorená neurónmi, ktoré spájajú sieť s prostredím
- Na vstupe každého neurónu (skrytá vrstva), je vypočítaná vzdialenosť medzi centrom neurónu & vstupným vektorom
- Používa RBF (Gaussovu funkciu) na vytvorenie výstupu neurónov
- Výstupná vrstva je lineárna a vydáva odozvu siete na zadané vstupy
- Toto hovorí o veľkej popularite týchto sietí a sú hlavným konkurentom dopredných viacvrstvových sietí

RBF sieť - architektúra





$$z_k = f(net_k) = f\left(\sum_{j=1}^{H} w_{kj} y_j\right) = \sum_{j=1}^{H} w_{kj} y_j$$

Lineárna akt. funkcia

$$\varphi\left(net_J = \|\mathbf{x} - \mathbf{u_J}\|\right)$$

$$= e^{-\left(\frac{\|\mathbf{x} - \mathbf{u_J}\|}{\sigma}\right)^2}$$

σ: konštanta pre šírku

Výstupná vrstva

- Výstupná vrstva vykonáva jednoduchú váženú sumu s lineárnym výstupom.
- Ak je RBF sieť používaná na aproximáciu funkcií (zhoda s reálnym číslom) potom výstup je veľmi dobrý.
- Avšak, ak je požadovaná klasifikácia vzorov, tak je potrebné použiť funkciu signum alebo tvrdé ohraničenie, aby bol dosiahnutý výstup 0/1.

Klastrovanie

- Jedinečným základným rysom RBF siete je proces vykonávaný v skrytej vrstve.
- Idea je taká, že vzorky vo vstupnom priestore tvoria klastre.
- Ak sú známe centrá klastrov, tak je možné merať vzdialenosti od centier.
- Ďalej, táto vzdialenosť je meraná nelineárne, takže keď vzorka je v oblasti klastra (blízko centra), tak dáva hodnotu blízku 1.
- Mimo tejto oblasti hodnota dramaticky klesá.
- Dôležité je, že táto oblasť je radiálne symetrická okolo klastrového centra, preto táto nelineárna funkcia sa stáva známou ako radial-basis funkcia.

Meranie vzdialenosti

- Obvykle Euklidova vzdialenosť.
- Pre každý neurón skrytej vrstvy váhy reprezentujú súradnice centra odpovedajúceho klastra.
- Preto pre vstup X, vzdialenosť od centra je určovaná pomocou vzťahu

$$\boxed{r_j = \sqrt{\sum_{i=1}^{n} (x_i - w_{ij})^2}}$$

Šírka funkcie v skrytých neurónoch

$$(skrytý\_neurón)\varphi_j = \exp\left(-\frac{\sum_{i=1}^{n}(x_i - w_{ij})^2}{2\sigma^2}\right)$$

σ *definuje* šírku alebo radius zvona Gaussovej funkcie a niekedy je určovaná empiricky.

Ak vzdialenosť od centra Gaussovej funkcie dosiahne σ, výstup sa pohybuje v intervale 1 - 0.6.

Hľadanie šírky Gaussovej funkcie

- Keď máme centrá klastrov je možné určiť radius Gaussovej krivky.
- *Je to možné urobiť metódou P*-najbližších susedov.
- Počet *P* je možné zvoliť, a pre každé centrum je nájdených P najbližších susedov.

- Pre klaster c_j, máme:

$$\sigma_j = \sqrt{\frac{1}{P}\sum_{i=1}^{P}(c_k - c_i)^2}$$

- Typická hodnota pre *P* je 2.

Neurónové siete s RBF

- Vzorky klastrov
  - Príliš mnoho skrytých neurónov, keď # vzoriek je príliš veľký
  - Zgrupovanie podobných vzoriek spolu do N klastrov, každý s
    - Centrálnym vektorom x_i
    - Predpokladaným stredným výstupom φ_i
    - Sieťovým výstupom $$o = \frac{1}{N}\sum_{i=1}^{N} \varphi_i \rho(\|\boldsymbol{\mu}_i - \boldsymbol{x}\|)$$
    - Predpokladajme, že vieme určiť N a vieme ako rozdeliť všetkých P vzoriek do klastrov (nie je to ľahká úloha), φ_i a μ_i môžu byť nastavené učením

Neurónové siete s Radial Basis Funkciami

- Učenie (trénovanie) RBF siete
  - Parametre učenia:

$$w_1 = \frac{\varphi_1}{N}, \ldots, w_N = \frac{\varphi_N}{N}, \quad \mu_1, \ldots, \mu_N$$

minimalizovať

$$E = \sum_{p=1}^{P} E_p = \sum_{p=1}^{P} (d_p - o_p)^2$$

  - Metóda klesajúceho gradientu

$$\Delta w_i \;=\; \eta_i (d_p - o_p) \rho(\|x_p - \mu_i\|) \text{ and}$$

$$\Delta \mu_{i,j} \;=\; -\eta_{i,j} w_i (d_p - o_p) R'(\|x_p - \mu_i\|^2)(x_{p,j} - \mu_{i,j})$$

  - Je možno určiť μ_it iež inými klastrovacími technikami a až potom použiť metódu klesajúceho gradientu len pre váhy

Trénovanie skrytej vrstvy

- Skrytá vrstva v RBF sieti má neuróny, ktoré majú váhy odpovedajúce vektorovej reprezentácii centra klastra.
- Tieto váhy sú nájdené buď tradičnými klastrovacími metódami alebo sú adaptované pomocou základného Kohonenovho algoritmu.
- V oboch prípadoch je trénovanie bez dozoru, len počet ***k*** zvolených klastrov je nastavený dopredu. Algoritmus tak nájde ***k*** najlepších klastrov.
- **Algoritmus *k* –mean:**
- Na začiatku je zvolených *k* bodov (predbežných centier) v priestore náhodne.
- Potom sú nájdené vzdialenosti všetkých bodov v tréningovej vzorke od ***k*** centier.

Trénovanie skrytej vrstvy

- Pre každý tréningový bod je vybrané najbližšie centrum – začiatočná klasifikácia, každý bod patrí do jednej triedy určenej centrom.
- Vezmeme všetky body, ktoré patria k centru 1, vypočítame priemerné hodnoty súradníc a tie sa stanú súradnicami nového centra 1.
- Toto sa opakuje pre všetky triedy 2 – k. Tak vznikne k nových centier.
- Proces merania vzdialenosti bodov od centier a reklasifikácia prebieha dovtedy, pokým nastávajú nejaké zmeny – t.j. súčet vzdialenosti je monitorovaný a trénovanie zastaví, keď celková vzdialenosť sa už nezmenšuje.

Adaptívny k-means

- Alternatívou je použitie *k* -means algoritmu, ktorý je podobný Kohonenovmu učeniu.
- Vstupné vzory sú prezentované všetkým klastrovým centrám po jednom. Najbližšie klastrové centrum vyhráva a je posunuté smerom k novým údajom.
- Toto má svoju výhodu v tom, že môže byť robené on-line, údaje sú spracovávané len jedenkrát.

Pokles gradientu a chyba

- Pokles gradientu je všeobecný prístup, východiskom je stanovená chyba

$$E(n) = \sum_k \left[d_k(n) - o_k(n)\right]^2 = \sum_k e_k(n)^2$$

where k are output nodes (desired output are known)

  - Úprava váh

$$\Delta w_{i,j}(n) = -\eta \frac{\partial E(n)}{\partial w_{i,j}}$$

Polynomiálne siete

- Polynomiálne siete
  - Funkcie v neurónoch umožňujú priamo počítať hodnoty polynómov na vstupoch
  - Aproximácie funkcií vyšších radov pomocou viacerých neurónov (hoci bez skrytých neurónov) Každý neurón má viac váhových prepojení
- Siete vyšších radov

$$f\left(w_0 + \sum_{j_1} w_{j_1} i_{j_1} + \ldots + \sum_{j_1, j_2, \ldots, j_k} w_{j_1, j_2, \ldots, j_k} i_{j_1} i_{j_2} \ldots i_{j_k}\right)$$

  - # váh pre jeden neurón:

$$1 + \binom{n}{1} + \binom{2n}{2} + \cdots + \binom{kn}{k}$$

- Môžu byť trénované metódou Least Mean Squared Errors (LMS).

Polynomiálne siete

- Sigma-pi siete
$$f\left(w_0 + \ldots + \sum_{j_1 \neq j_2 \neq j_3 \neq j_1} w_{j_1,j_2,j_3} i_{j_1} i_{j_2} i_{j_3} + \ldots\right)$$
  - Nepovoľuje výrazy s vyššími mocninami vstupov, teda nie sú všeobecným aproximátorom
  - # váh na jeden neurón:
$$\boxed{1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{k}}$$
  - Môže byť trénovaná LMS
- Pi-sigma siete
  - Jedna skrytá vrstva so Sigma funkciou: $w_{k,0} + \sum_j w_{k,j} i_j$
  - Výstupné vrcholy s Pi funkciou: $f\left(\prod_k (w_{k,0} + \sum_j w_{k,j} i_j)\right)$
- Súčinové neuróny:
  - Neurón počíta súčin: $\prod_{j=1}^{n} i_j^{p_{j,i}}$
  - Celočíselná mocnina $P_{j,i}$ môže byť natrénovaná
  - Často sa tieto neuróny kombinujú s inými jednotkami (e.g., sigmoid)

Záver

- RBF sa trénuje rýchlejšie ako dopredné NS
- Je lepšia interpretácia použitej siete ako u dopredných NS
- Výsledky ukazujú, že RBF siete s adaptívnymi centrami pracujú lepšie než s fixnými
- Prečo to tak je, nám vysvetľuje Coverova veta.

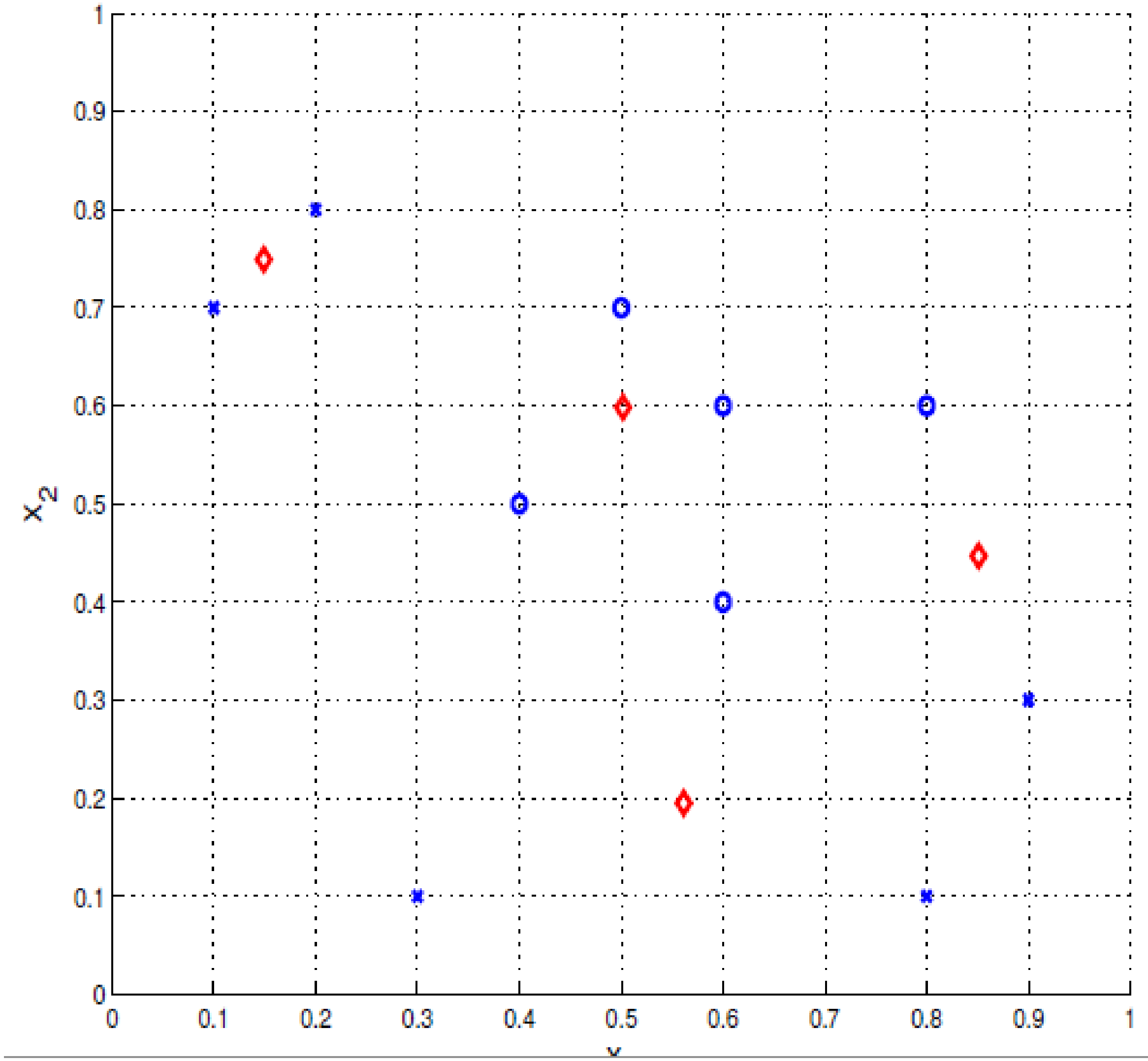
Príklad

- Tréningová vzorka: (**x1**; t1),…, (**x10**; t10)

i	1	2	3	4	5
x1i	0.5	0.4	0.6	0.6	0.8
x2i	0.7	0.5	0.6	0.4	0.6
ti	-1	-1	-1	-1	-1

i	6	7	8	9	10
x1i	0.2	0.1	0.9	0.8	0.3
x2i	0.8	0.7	0.3	0.1	0.1
ti	1	1	1	1	1

Grafické znázornenie



Určenie centier

- Stanovíme počet centier M = 4,
- 4 centrá boli určené algoritmom k-means
- $c1 = [0{,}1490;\ 0{,}7490]^T$,
- $c2 = [0{,}8510;\ 0{,}4471]^T$,
- $c3 = [0{,}5615;\ 0{,}1950]^T$,
- $c4 = [0{,}5018;\ 0{,}5984]^T$.

We set $\sigma = 1$ as before, and this gives us four basis functions $\phi_1(\mathbf{x})$ $\phi_2(\mathbf{x})$ $\phi_3(\mathbf{x})$ and $\phi_4(\mathbf{x})$. e.g.

$$\phi_1(\mathbf{x}) = \exp\left(-\frac{(x_{1,i} - 0.1490)^2 + (x_{2,i} - 0.7490)^2}{2}\right)$$

Over the given ten data samples, form the matrix Φ given by

$$\Phi = \begin{pmatrix} \phi_{1,1} & \phi_{1,2} & \phi_{1,3} & \phi_{1,4} \\ \phi_{2,1} & \phi_{2,2} & \phi_{2,3} & \phi_{3,4} \\ \vdots & \vdots & \dots & \vdots \\ \phi_{9,1} & \phi_{9,2} & \phi_{9,3} & \phi_{9,4} \\ \phi_{10,1} & \phi_{10,2} & \phi_{10,3} & \phi_{10,4} \end{pmatrix}$$

We can write ten linear equations

$$\begin{cases} \phi_{1,1}w_1 + \phi_{1,2}w_2 + \phi_{1,3}w_3 + \phi_{1,4}w_4 = t_1 \\ \phi_{2,1}w_1 + \phi_{2,2}w_2 + \phi_{2,3}w_3 + \phi_{2,4}w_4 = t_2 \\ \phi_{3,1}w_1 + \phi_{3,2}w_2 + \phi_{3,3}w_3 + \phi_{3,4}w_4 = t_3 \\ \cdots\cdots \\ \phi_{10,1}w_1 + \phi_{10,2}w_2 + \phi_{10,3}w_3 + \phi_{10,3}w_4 = t_{10} \end{cases}$$

which is

$$\Phi \mathrm{w} = \mathrm{t}$$

with $\mathbf{t} = [-1, -1, -1, -1, -1, 1, 1, 1, 1, 1]^T$. The least squares estimate is calculated as

$$\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$$

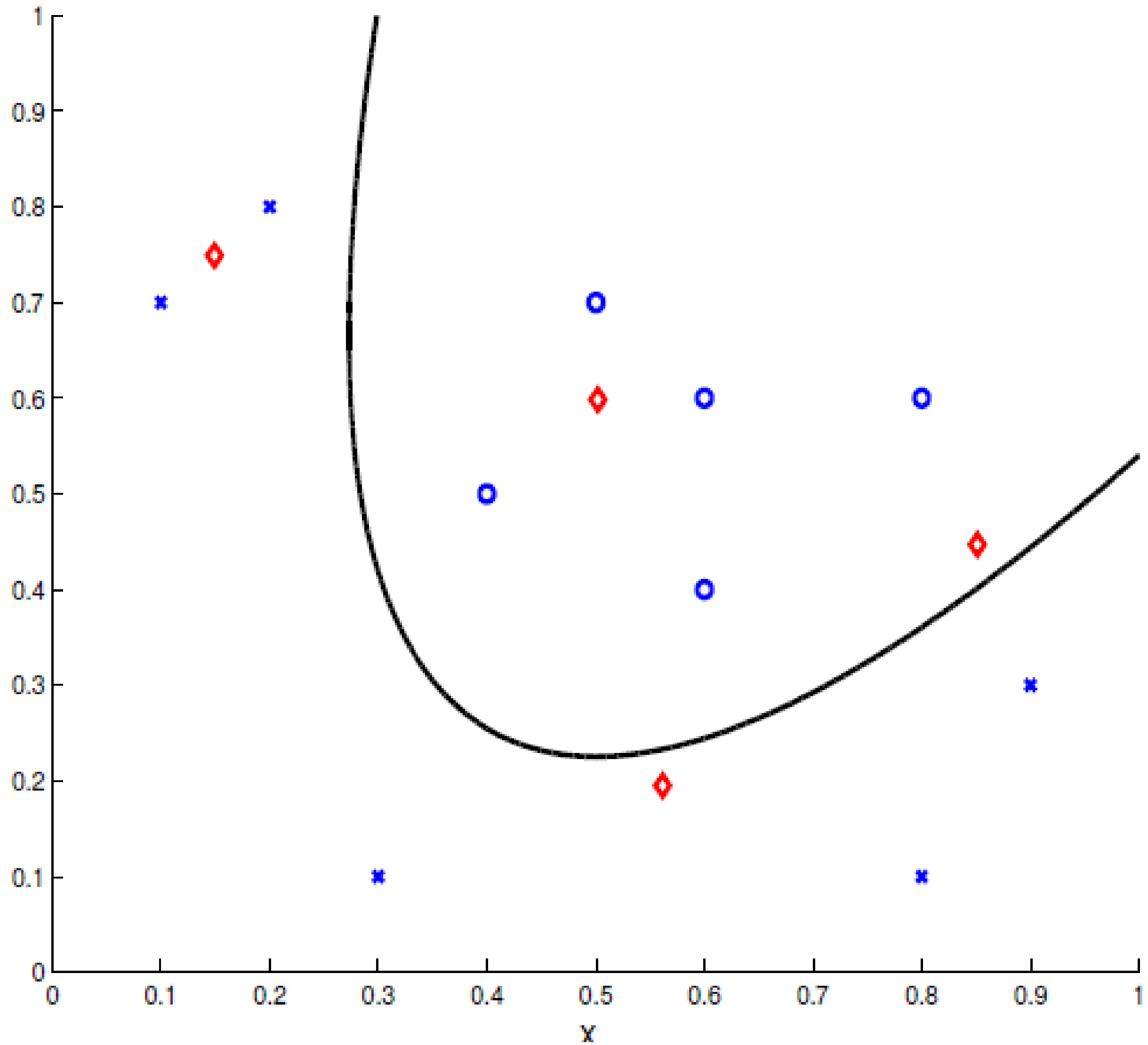
The RBF classifier is given by

$$g(\mathbf{x}) = \sum_{i=1}^{4} w_i \phi_i(\mathbf{x})$$

$\mathbf{w}$ is found to be

$\mathbf{w} = [-74.1191, -65.3503, -8.2930, 138.2853]^T$

Klasifikácia



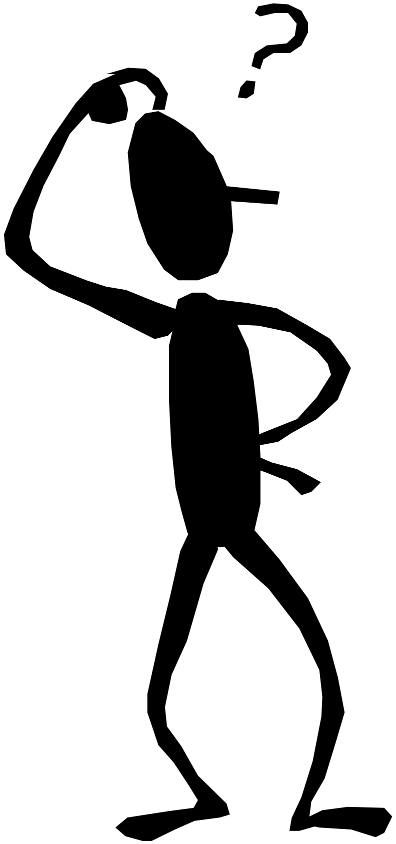
A summary of the construction of RBF network

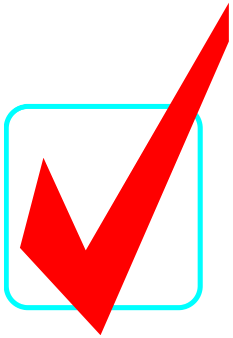
1. Predetermine the number the centers.
2. Use the k-means clustering algorithm to find centers ci;
2. Calculate ϕi(x) for all training data samples;
3. Form matrix Φ and t;
4. Calculate

$$w = (\Phi_T\Phi)_{-1}\, \Phi_T\, t$$

5. Use the resultant classier g(x) for classification.

Note that k-means clustering algorithm can also be used RBF neural networks center selection for regression.





Modifikované podľa Dr. Emad A. A. El-Sebahy