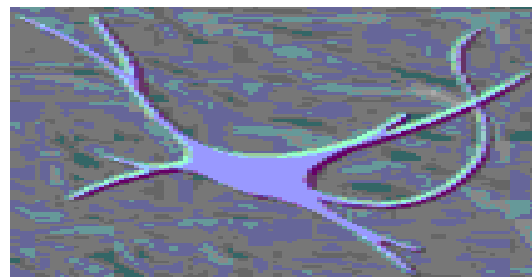


Modelovanie nelineárnych dynamických procesov inteligentnými metódami - umelými neurónovými siet'ami (UNS)



Neurón



Aplikácia umelých neurónových sietí v úlohách riadenia

- **Umelé nerónové siete (UNS) sú dnes široko využívané na vývoj empirických modelov pre širokú triedu vedeckých, inžinierskych a podnikateľských aktivít.**
- **V oblasti automatického riadenia sú UNS využívané pre modelovanie lineárnych a nelineárnych procesov, riadenie procesov - vývoj nových algoritmov typu PID, prediktívnych a adaptívnych.**
- **Priemyselné aplikácie UNS v posledných rokoch – nelineárne prediktívne algoritmy – komerčné riadiace systémy (softvérové balíky) (Honeywell, Yokogawa, Delta V, Siemens)**

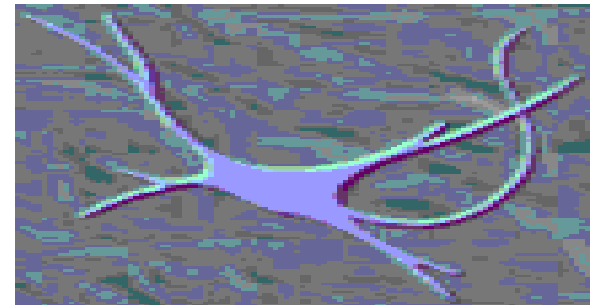
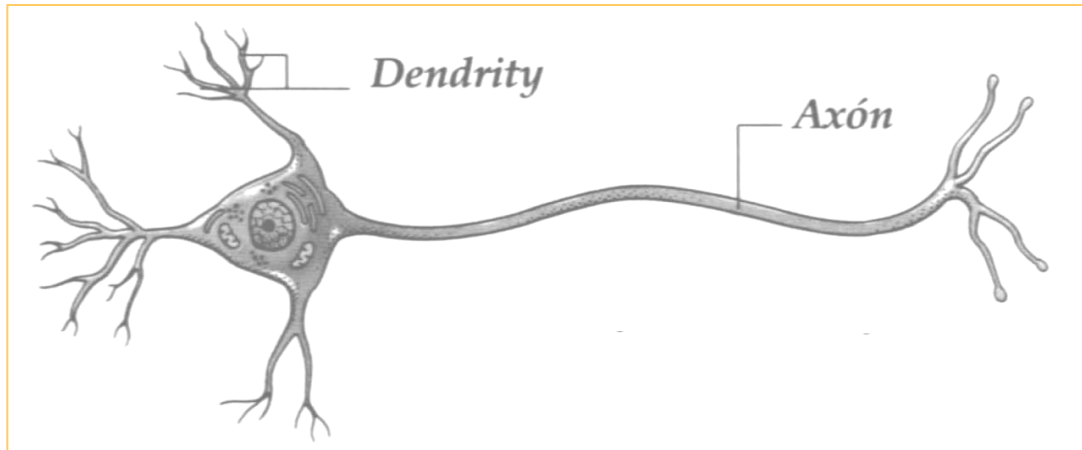
Model neurónovej siete (NS) : činnosti biologických mozgových a nervových štruktúr.

o Základnou bunkou týchto štruktúr NS je neurón - je to živá bunka: najúčelnejšie spracovanie, uchovávanie a prenos informácií.

o Ľudský mozog: súbor asi 20 až 100 miliárd buniek

o **Neurón** : základné telo (soma)

Jediný aj keď rozvetvený výstup: (**axón**). Vstup informácií z iných neurónov sa uskutočňuje priamo na **soma**, alebo na viac či menej rozvetvené vstupy (**dendrity**). Na mieste styku axonu s časťami iných neurónov pôsobia ďalšie funkčné časti - **synapsie**.



Základná funkcia neurónu:

- zbere elementárnych informácií z výstupu určitého počtu s ním spolupracujúcich neurónov,
- vyhodnotenia ich súhrnného stavu a vyhodnotenie zodpovedajúcej reakcie na výstupe.

Zjednodušené model neurónu 1943 *McCulloch a Pitts* vzťahom

$$a = f(q) = f\left(\sum_{i=1}^n x_i w_i\right)$$

n - je počet spolupracujúcich neurónov (počet vstupov do neurónu)

x - i -tý vstupný signál neurónu snímaný zo vstupu, alebo zo spolupracujúceho neurónu

w_i - váhy vstupných brán i -teho vstupu hrán (synaptické váhy - weights)

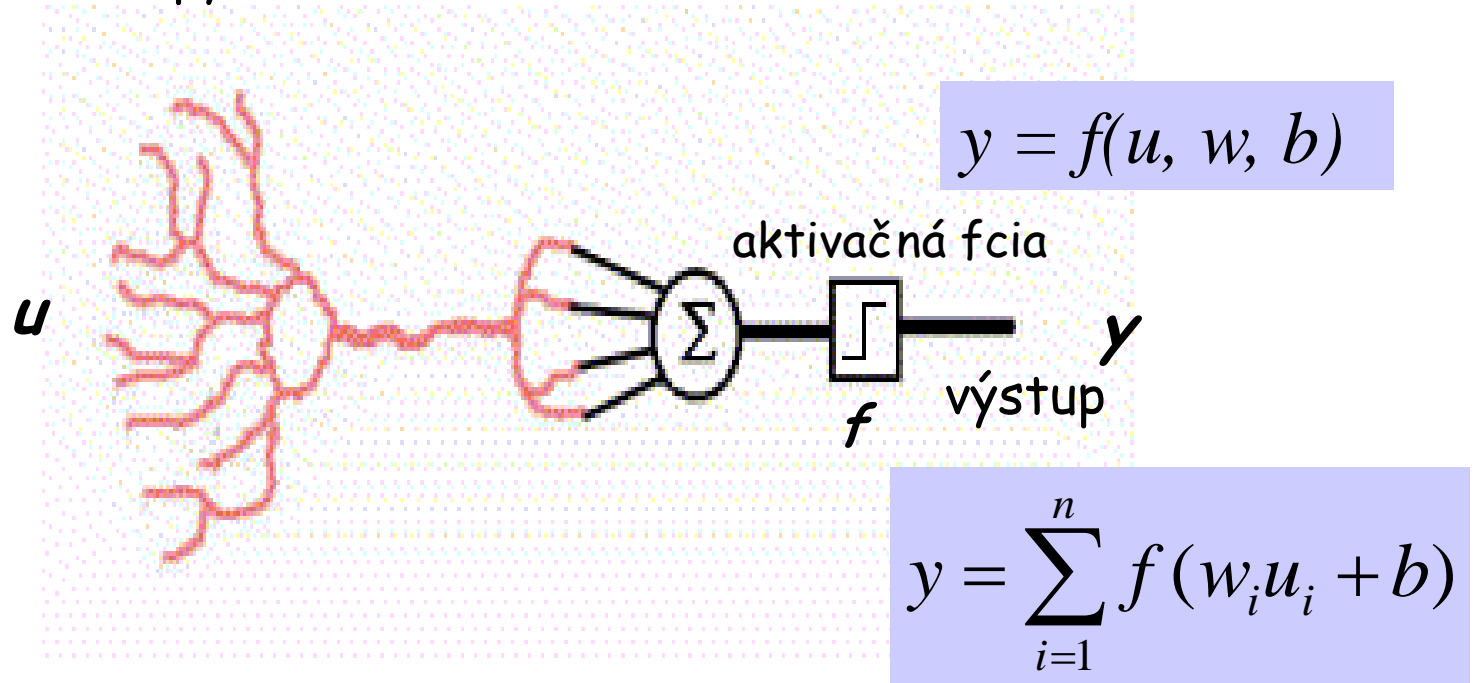
f - funkcia prenosu signálu (aktivačná funkcia neurónu)

a - výstupný signál neurónu

q - potenciál neurónu

Model neurónu typu perceptrón

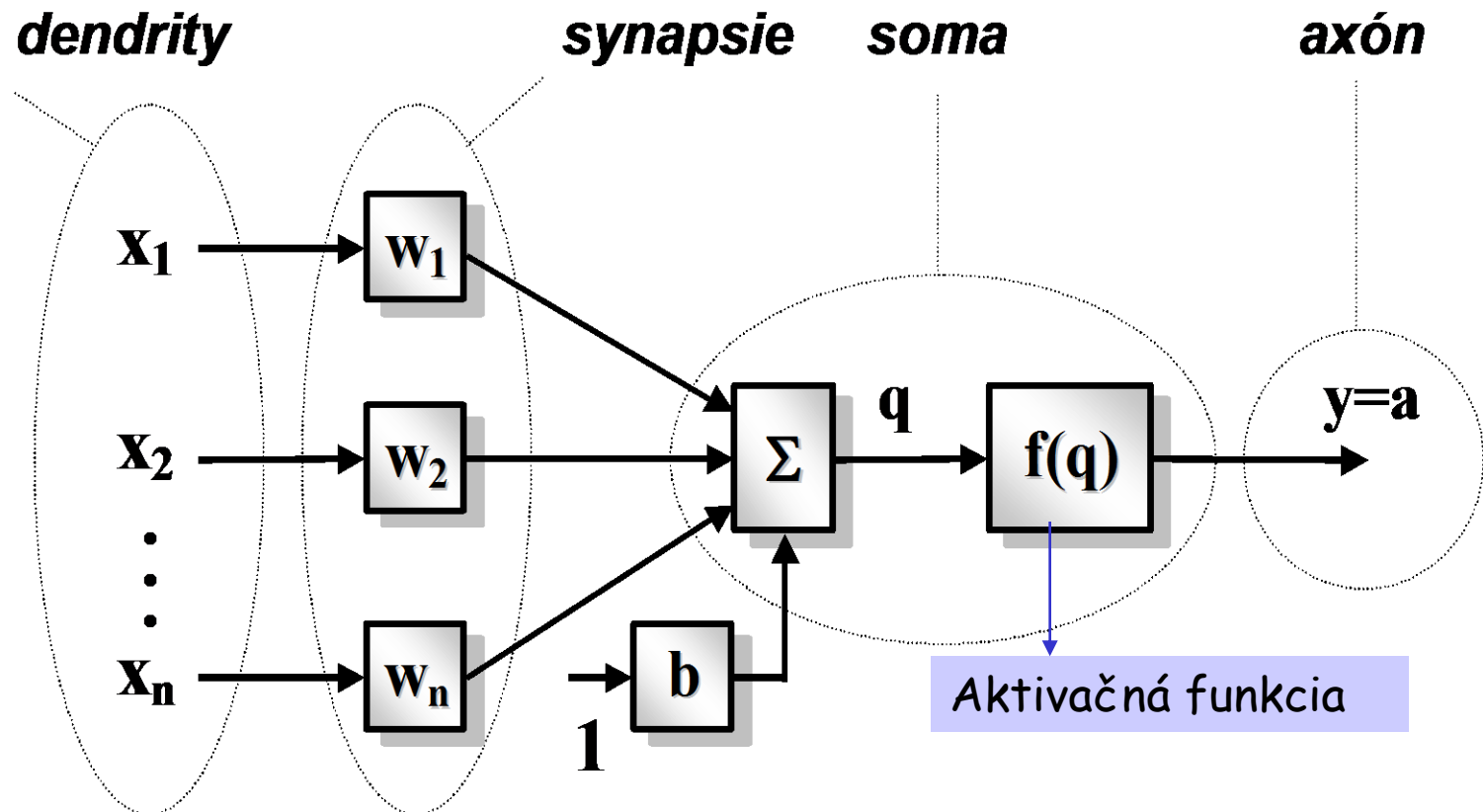
vstupy



$$Y = T(U)$$

Bias $b = w_0 \cdot x_0$, kde x_0 je vstup s hodnotou 1 a veľkosť váhy w_0 je potom rovná priamo hodnote prahu.

$$a = f\left(\sum_{i=1}^n w_i x_i + b\right)$$



Neurón : vlastnosti a schopnosti dominantne charakterizované prenosovou (aktivačnou) funkciou.

Použitie : niekoľko základných typov prenosové funkcie.

U jednoduchých modelov neurónov:

- lineárne a
- nelineárne aktivačné funkcie.

Najviac používané :

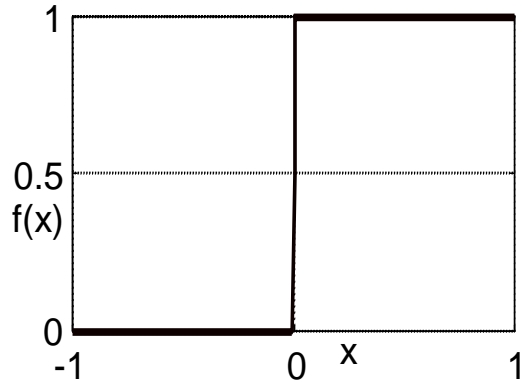
- lineárne aktivačné funkcie
- aktivačné funkcie v tvare silných nelinearít
- sigmoidálne aktivačné funkcie
- aktivačné funkcie v tvare Gaussovej funkcie (radial basis)

Príklady jednotlivých aktivačných funkcií

A. aktivačné funkcie v tvare silných nelinearít

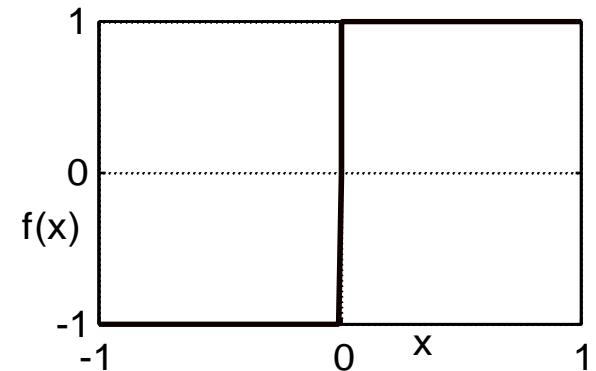
skoková binárna funkcia

$$f(x) = \text{sign}(x)$$



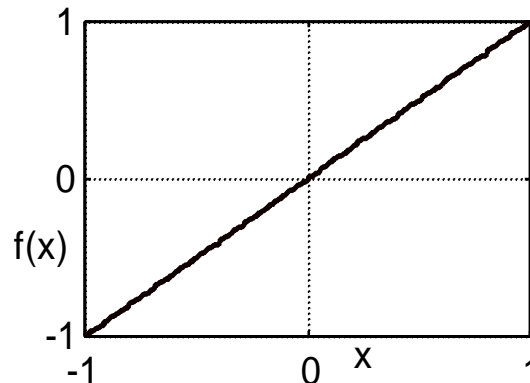
skoková bipolárna funkcia

$$f(x) = \frac{2 \cdot \text{sign}(x) + 1}{2}$$



lineárna funkcia

$$f(x) = x$$

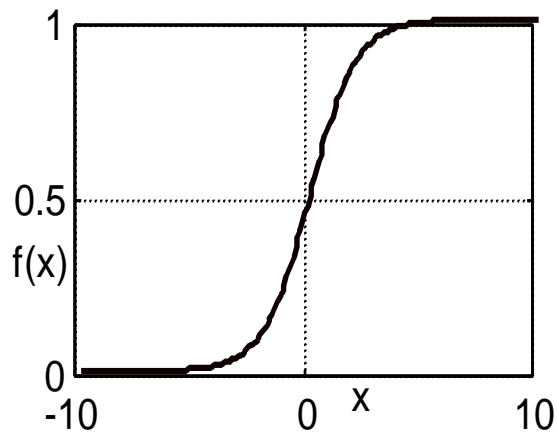


- sigmoidálne prenosové funkcie :

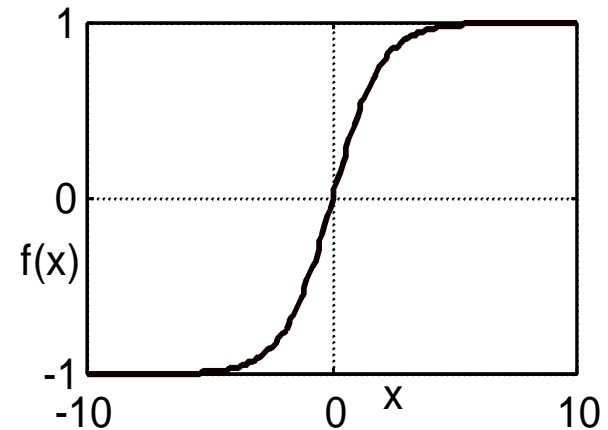
binárny (logický)
sigmoid bipolárny

sigmoid (hyperbolická tangenta)

$$f(x) = \frac{1}{1 + e^{-x}}$$

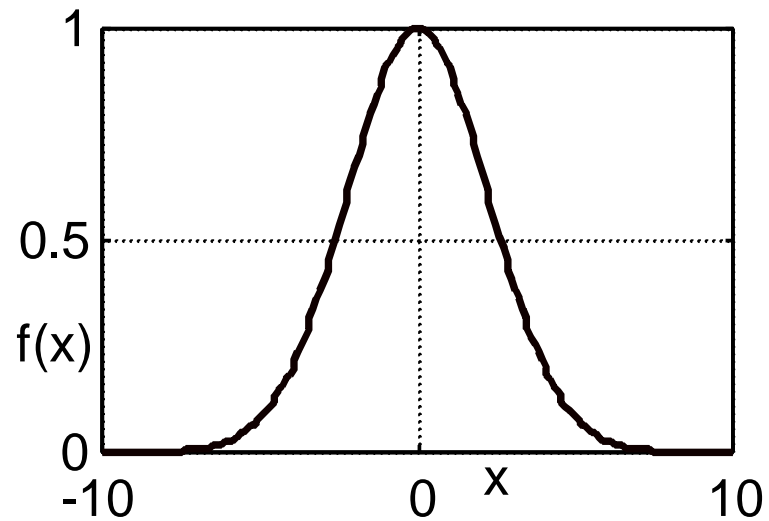


$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



- radiálne bázové prenosové funkcie (RBF):

$$f(x) = e^{-x^2}$$



Prechodové funkcie neurónov -Transfer functions

Competitive transfer function.

hardlim

Hard limit transfer function.

hardlims

Symmetric hard limit transfer function

logsig

Log sigmoid transfer function.

poslin

Positive linear transfer function

purelin

Linear transfer function.

radbas

Radial basis transfer function.

satlin

Saturating linear transfer function.

satlins

Symmetric saturating linear transfer function

softmax

Soft max transfer function.

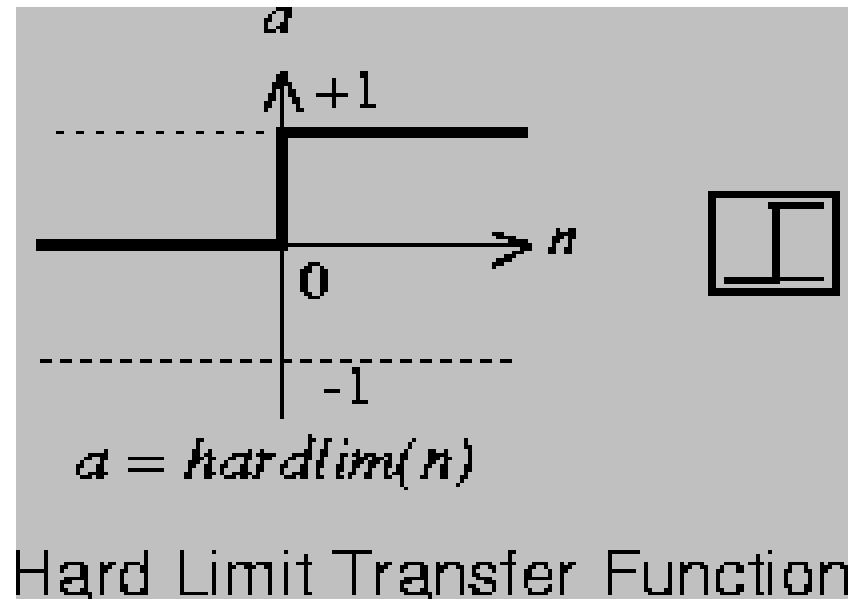
tansig

Hyperbolic tangent sigmoid transfer function.

tribas

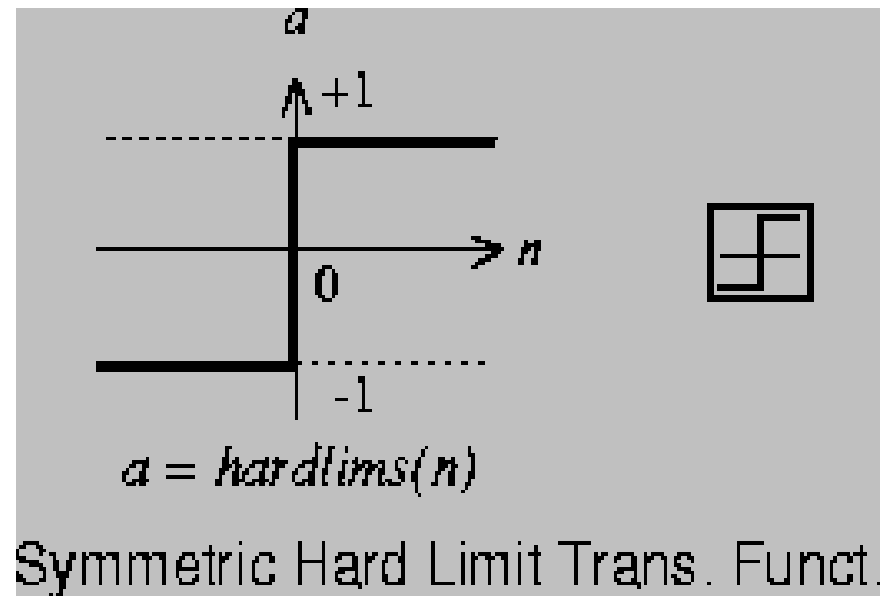
Triangular basis transfer function.

$$\text{hardlim}(n) = \begin{cases} 1, & \text{if } n \geq 0 \\ 0, & \text{otherwise} \end{cases}$$



Algorithm

$$\text{hardlims}(n) = \begin{cases} 1, & \text{if } n \geq 0 \\ -1, & \text{otherwise} \end{cases}$$



$$\text{logsig}(n) = 1 / (1 + \exp(-n))$$

```
n = -5:0.1:5; A = logsig(n);  
plot(n,A)
```

We calculate the layer's output a with LOGSIG and then the derivative of A with respect to N .

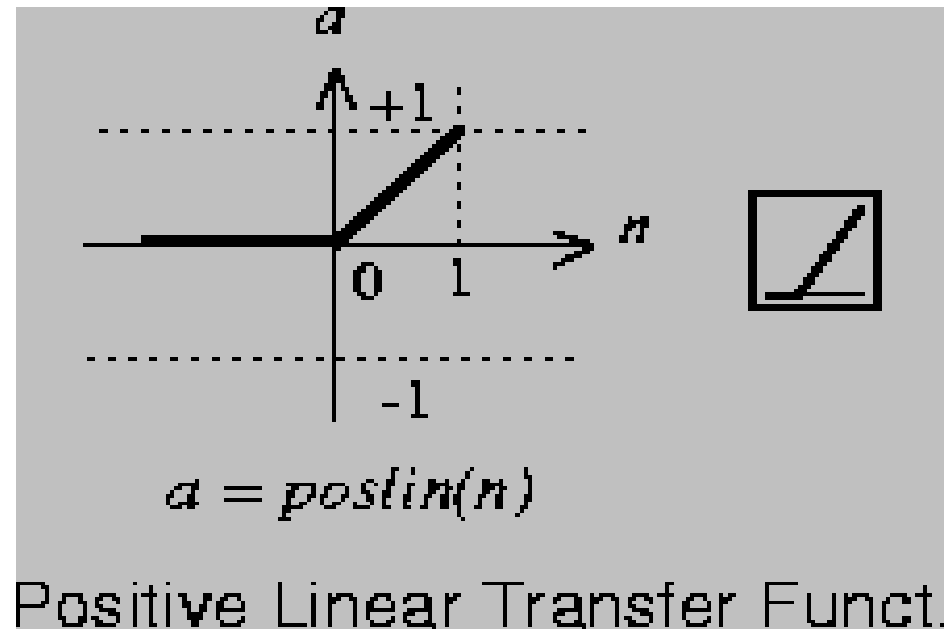
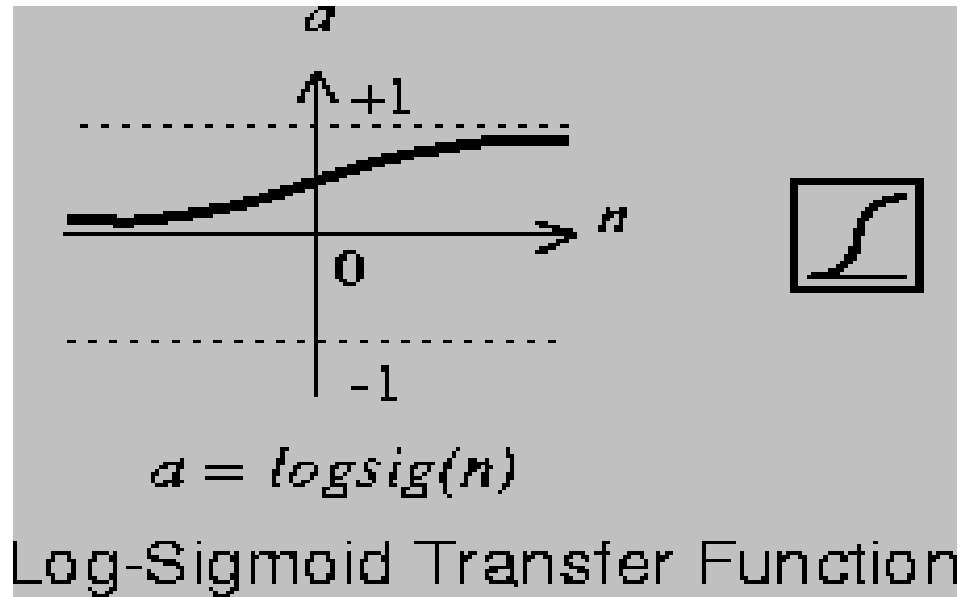
$$A = \text{logsig}(n) \quad dA_{dN} = d\text{logsig}(N,A)$$

The derivative of LOGSIG

$$d = a * (1 - a)$$

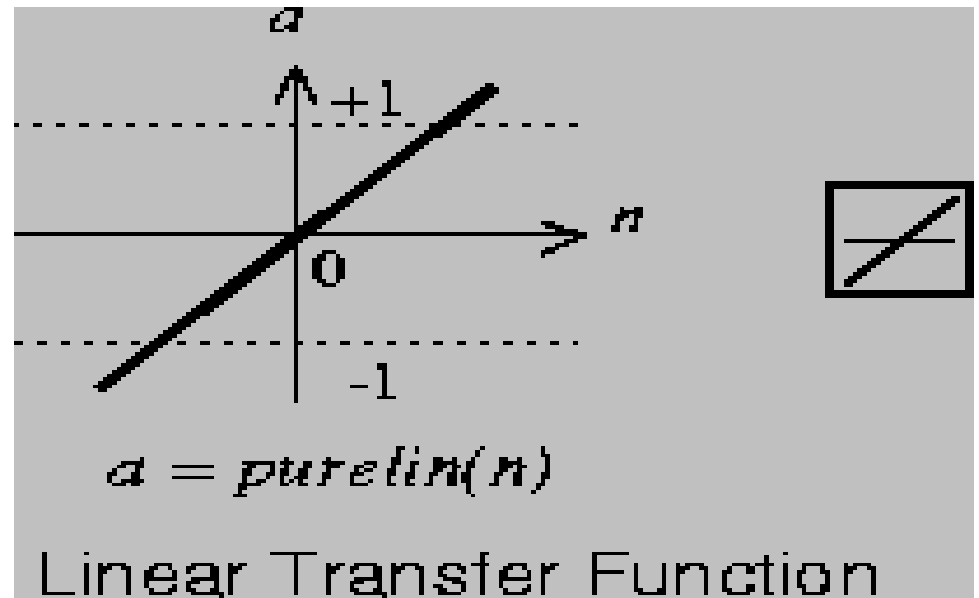
Algorithm

$$\begin{aligned} \text{poslin}(n) &= n, \text{ if } n \geq 0 \\ &= 0, \text{ if } n \leq 0 \end{aligned}$$



Algorithm

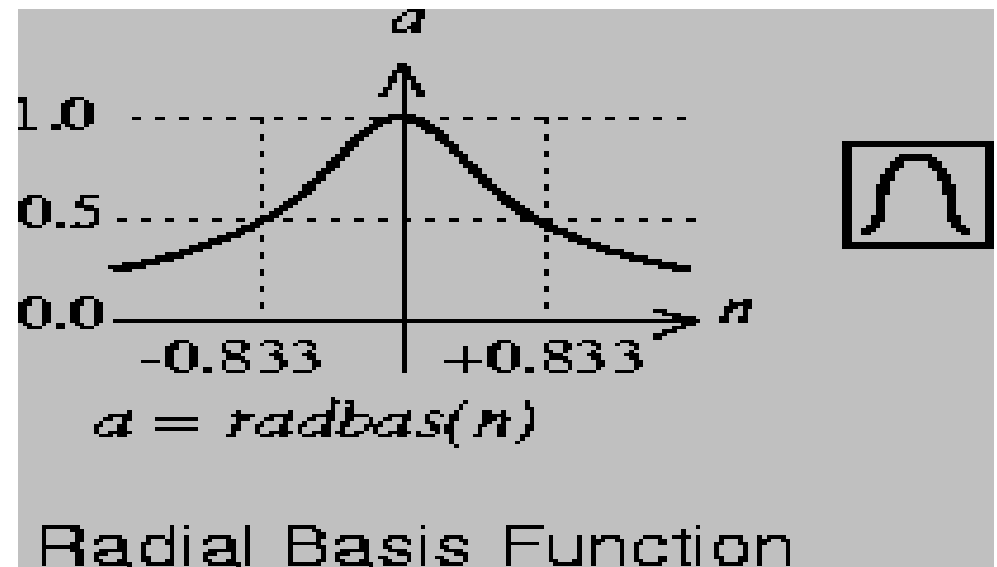
$$\text{purelin}(n) = n$$



Algorithm

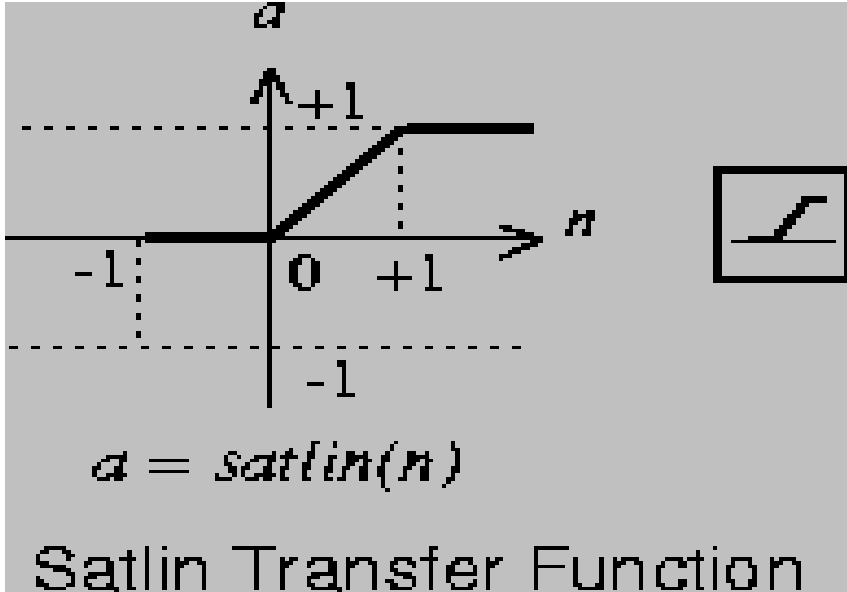
RADBAS(N)

calculates its output with
according to: $a = \exp(-n^2)$

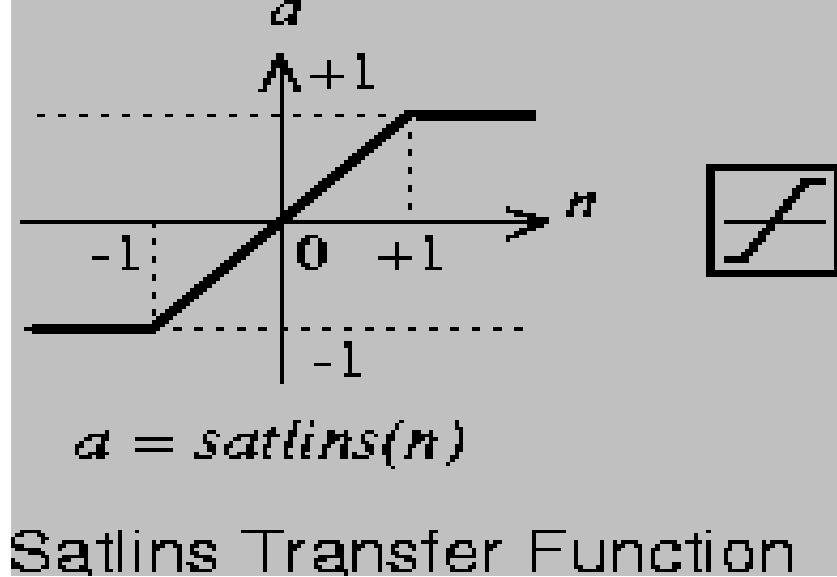


Algorithm

$$\text{satlin}(n) = \begin{cases} 0, & \text{if } n \leq 0 \\ n, & \text{if } 0 < n \leq 1 \\ 1, & \text{if } 1 < n \end{cases}$$

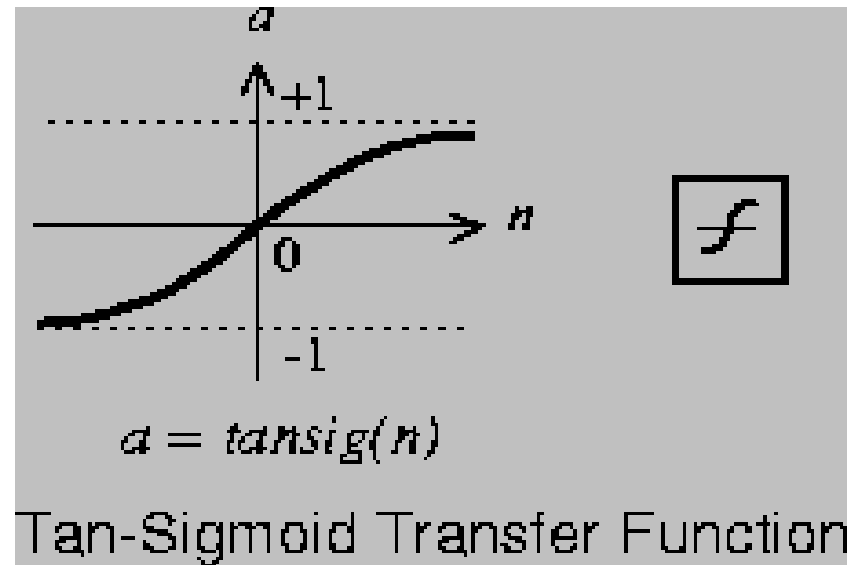


$$\text{satlins}(n) = \begin{cases} -1, & \text{if } n \leq -1 \\ n, & \text{if } -1 < n < 1 \\ 1, & \text{if } 1 \leq n \end{cases}$$



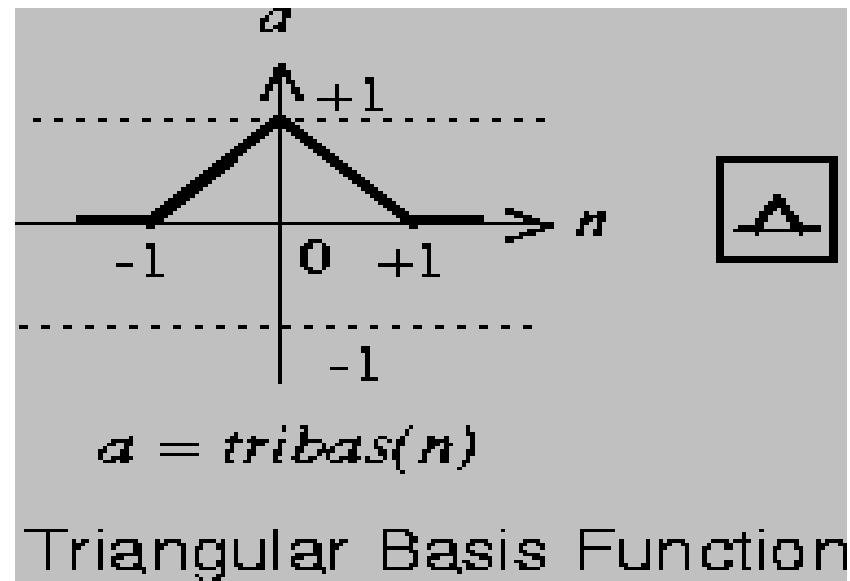
TANSIG(N) calculates its output according to:

$$n = 2/(1+\exp(-2*n))-1$$



TRIBAS(N) calculates its output with according to:

$$\begin{aligned} \text{tribas}(n) &= 1 - \text{abs}(n), \text{ if } -1 \leq n \leq 1 \\ &= 0, \text{ otherwise} \end{aligned}$$

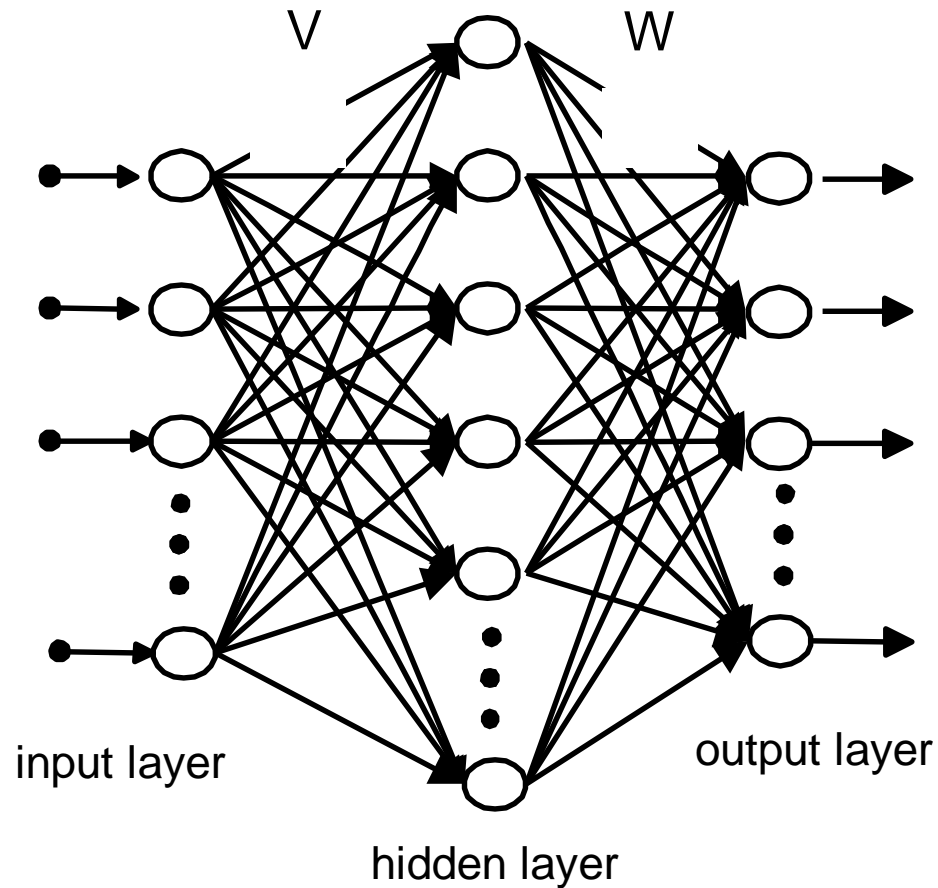


Dôsledky:

Sigmoidálne aktivačné funkcie-vlastnosti:

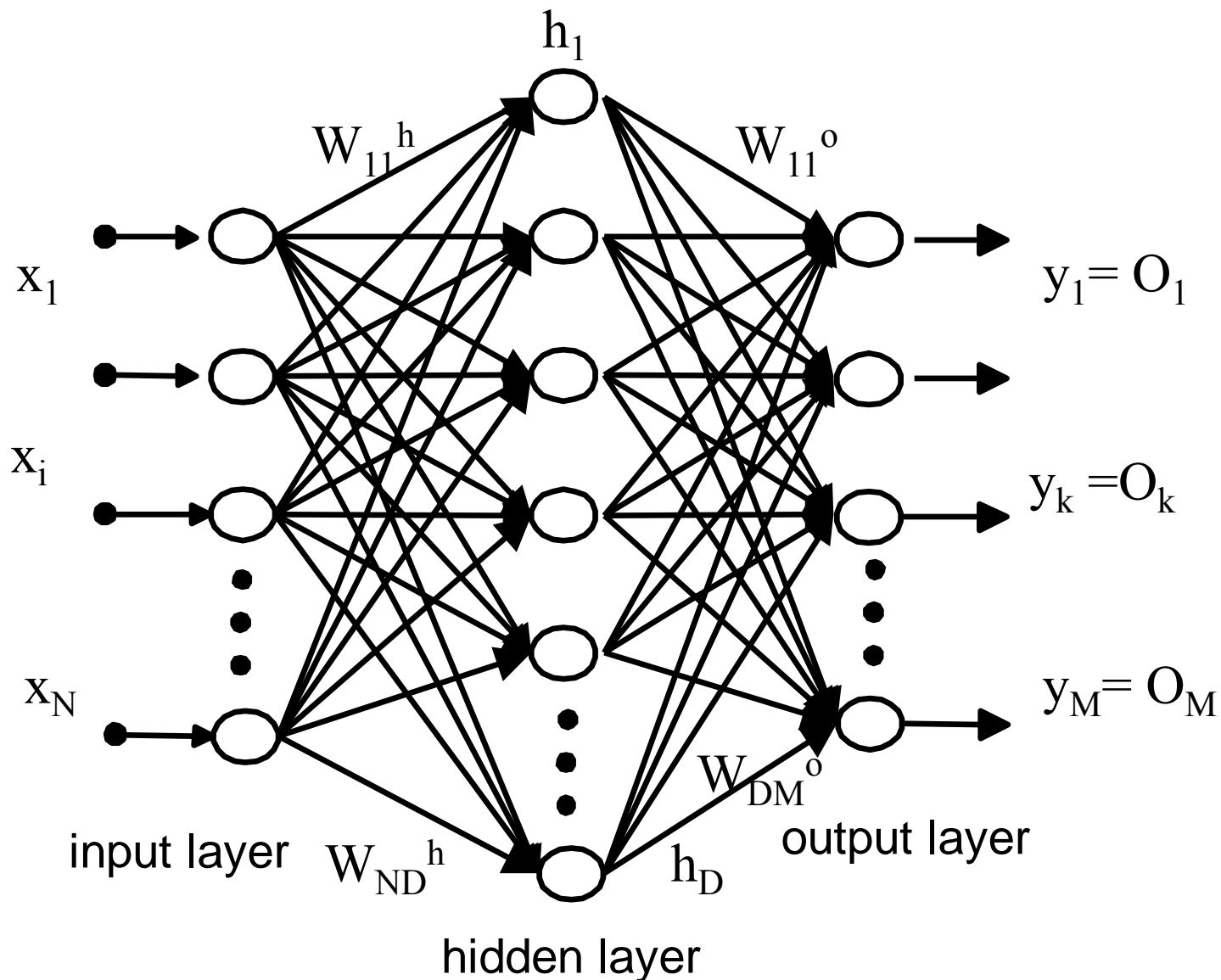
- umožňujú dosiahnuť veľkú citlivosť neurónu pre malé signály, pre väčšiu úroveň signálov klesá ich citlivosť
- výhoda využítiev UNS (ANN) na to, aby sieť bola schopná spracovávať signály v dostatočnom rozsahu ich dynamiky bez toho, aby hrozilo nebezpečie jej zahltenia pri príchode príliš veľkých signálov a naopak, aby pri príchode veľmi malých signálov bola zabezpečená jej dostatočná citlivosť.

Viacvrstvová štruktúra



Dopredná sieť

Transformácia signálov zo vstupu na výstup



Výstup zo siete

Skrytá vrstva

Aktivačná fcia - prenosová funkcia neurónu

$$h_j = \varphi \left(\sum_i^N W_{ij}^h x_i \right)$$

Výstupná vrstva (target function)

$$O_k = \varphi \left(\sum_j^D W_{jk}^o h_j \right)$$

Učenie v ANN sieti

Minimalizácia odchýlky...

$$\varepsilon = \sum_l \left(\sum_k \left(O_k^l - T_k^l \right)^2 \right)$$

↓
Výstup z modelu

↑
Cieľová množina

$$O_k = \varphi \left(\sum_j W_{jk}^o h_j \right)$$

Úloha : výpočet váh v každej vrstve

$$\frac{\partial \varepsilon}{\partial W_{rs}^p} = 0$$

Where $p = h$ or o , and r and s take on all appropriate values for each layer.