

Metóda backpropagation (BP)

- najpoužívanejší algoritmus učenia NS
- typ algoritmu - gradientny, iteračný
- 1984-1985 objavený
- vhodnosť pre viacvrstvové siete
- učenie na základe referenčných hodnôt (s učiteľom)

Neurónová sieť - reprezentácia modelu - merané údaje -
trénovacia množina $X \rightarrow Y$

Zobrazenie : $y = f(x, w, \theta)$

$F: X \rightarrow Y$, také, že funkcia f je aproximácia zobrazenia

Parametre učenia-váhy w a prahové funkcie θ

Kvalita učenia-kvalitou trénovacej množiny na základe merania,
pozorovania-počet vzorov (viac ako 100 až 1000)

Overenie navrhnutých algoritmov na reálnych laboratórnych modeloch procesov -prezentácia použitia neurónových sietí ako univerzálneho aproximátora nelineárnych funkcií a realizácia samonastavujúceho sa regulátora využívajúceho umelé neurónové siete na reálnych laboratórnych modeloch procesov.

Cieľ : vytvorenie takého neurónového modelu, ktorý by čo najpresnejšie vystihoval chovanie nelineárneho procesu.

Model procesu realizovaný pomocou konvenčnej neurónovej siete

Konvenčná neurónová sieť sa rozumie trojvrstvová umelá neurónová sieť s dopredným šírením signálu so sigmoidnými aktivačnými funkciami

Nelineárny proces je opísaný v tvare:

$$y_p(k+1) = \frac{0.5y_p(k)}{1 + y_p(k) + 0.8y_p^2(k)} + x^3(k)$$

kde je y_p výstupná veličina procesu, x je vstupná veličina.

- na identifikáciu procesu je vhodné použiť nelineárnu trojvrstvovú sieť s dopredným šírením signálu.

Model získaný identifikáciou je použiteľný pri návrhu riadenia napr. pre samonastavujúce sa regulátory.

Preto musí byť model predtrénovaný.

Trénovací (vstupný) signál procesu pri off – line modelovaní je definovaný ako :

$$x(k) = \cos(P\pi) - \sin\left(\frac{5P\pi}{2}\right) \quad P \in \langle -2, 2 \rangle$$

Riešenie : vstupná vrstva je tvorená šiestimi neurónmi, ktorých vstup tvorí predchádzajúca hodnota vstupnej prípadne výstupnej veličiny procesu t. z. vstupný vektor siete má tvar :

$$\bar{x} = [x(k-1) x(k-2) x(k-3) y_p(k-1) y_p(k-2) y_p(k-3)]^T$$

Skrytá vrstva obsahuje šesť neurónov s prenosovou funkciou bipolárneho sigmoidu definovaného vzťahom :

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

Vo výstupnej vrstve je jeden neurón, pretože sa jedná o SISO (single input single output) systém, ktorého prenosová funkcia je lineárna (purelin) .

V adaptačnom procese siete doporučenie použiť algoritmus *Levenberg - Marquardta*.

V prípade, že neexistuje také nastavenie váh a prahov, ktoré by zaručovalo riešenie s nulovou chybou, vracia funkcia realizovaná v toolboxe Nnet nastavenie váh a prahov s minimálnou kvadratickou chybou.

Kvadratická chyba SSE (sum - squared error) nazývaná aj globálna chyba je definovaná vzťahom:

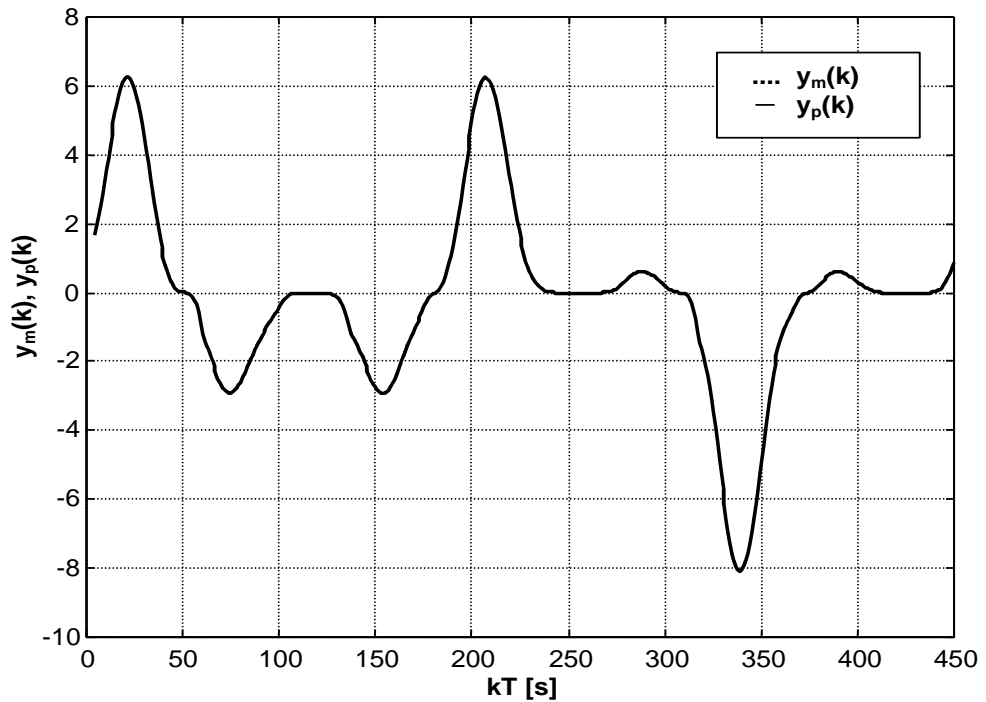
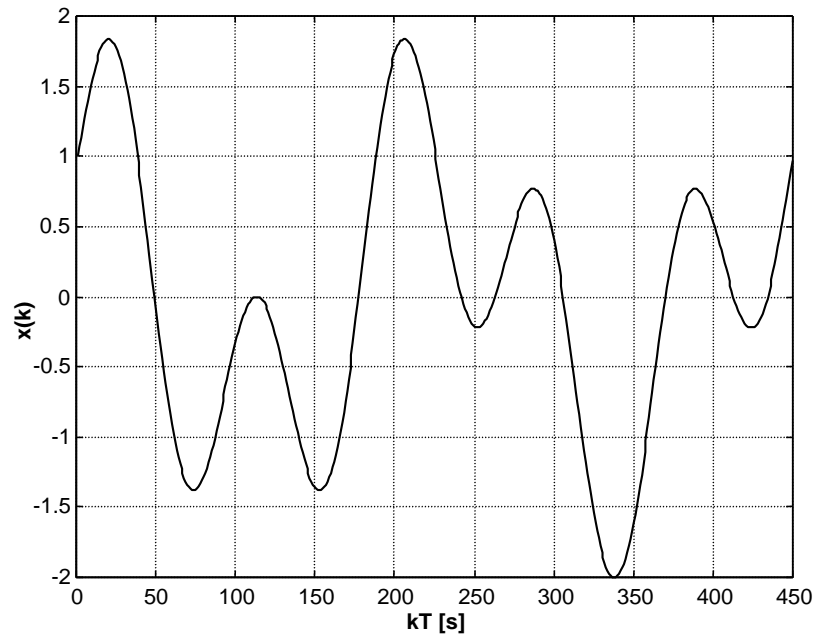
$$SSE = \frac{1}{2} \sum_{q=1}^Q \sum_{i=1}^M (t_i^q - y_i^q)^2$$

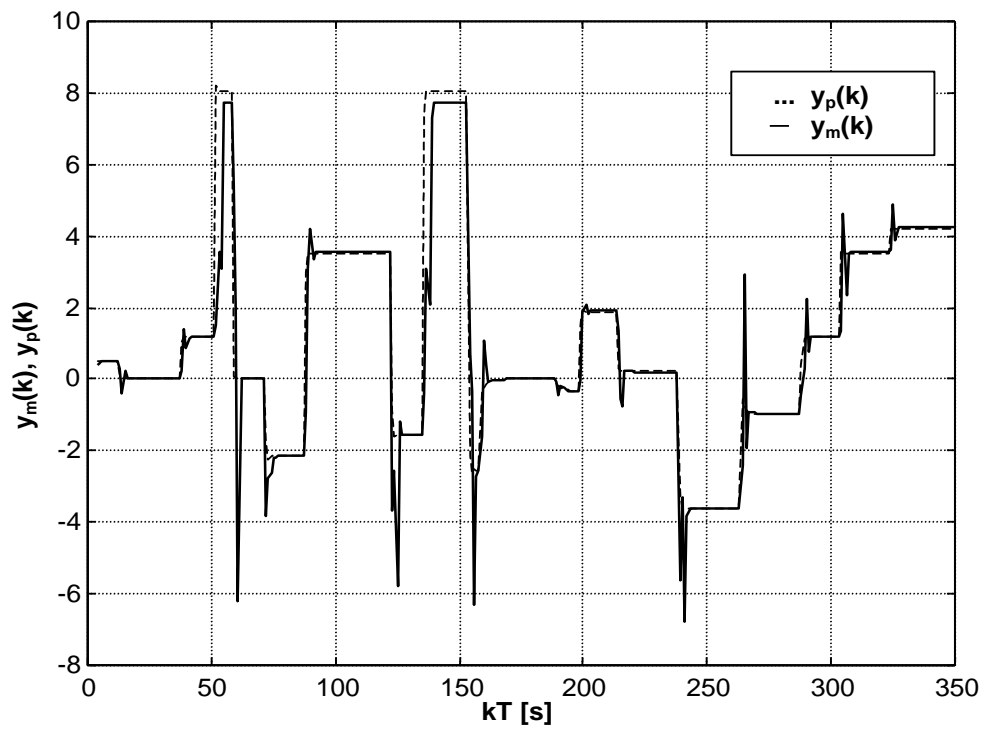
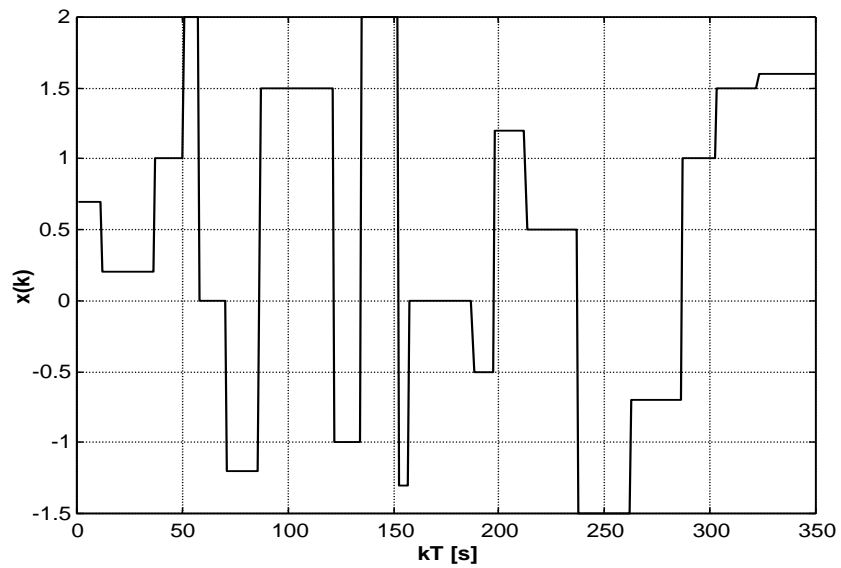
kde Q - počet tréningových vzoriek

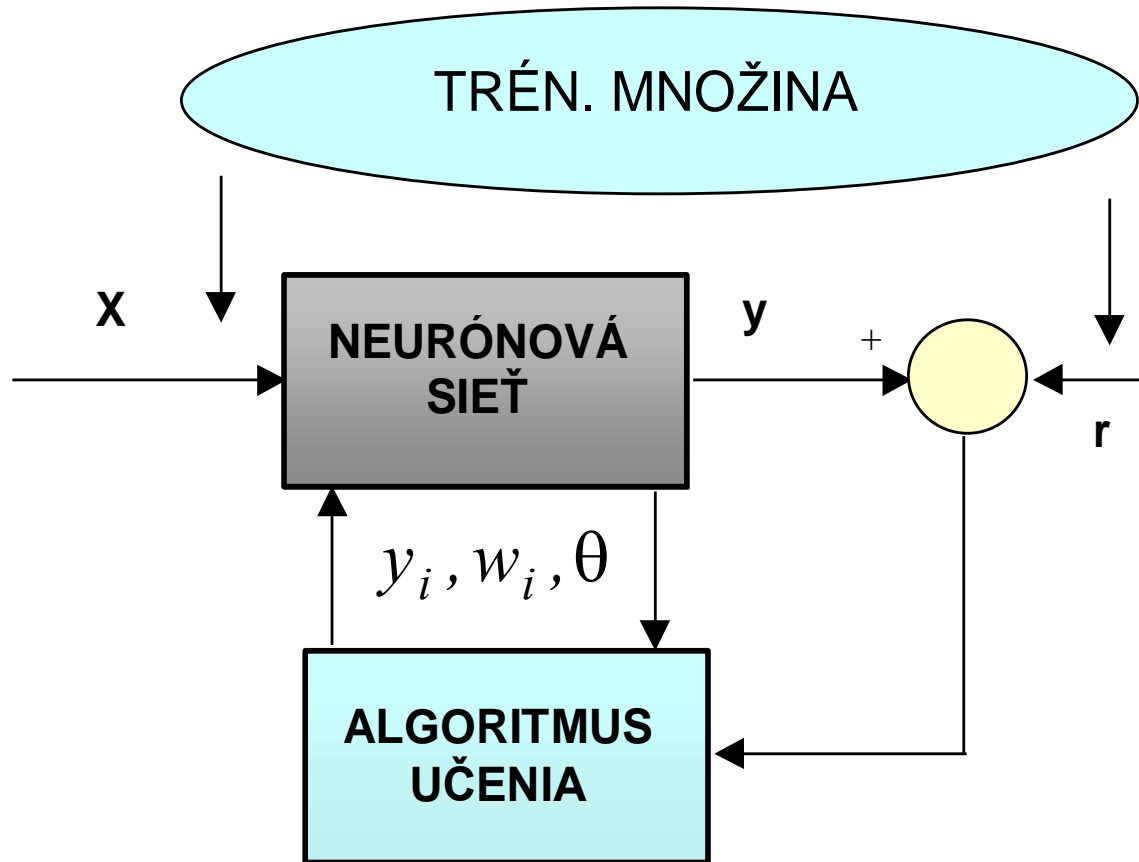
M - počet neurónov vo výstupnej vrstve

t_q^i - požadovaný výstup i - teho neurónu výstupnej vrstvy pri q - tej tréningovej vzorke

y_q^i - skutočný výstup i - teho neurónu výstupnej vrstvy pri q - tej tréningovej vzorke.







- nastavenie vstupov do siete
- inicializácia váh
- test ukončenia

Princíp učenia: Test vedomosti : *if nie ... then*
doučiť

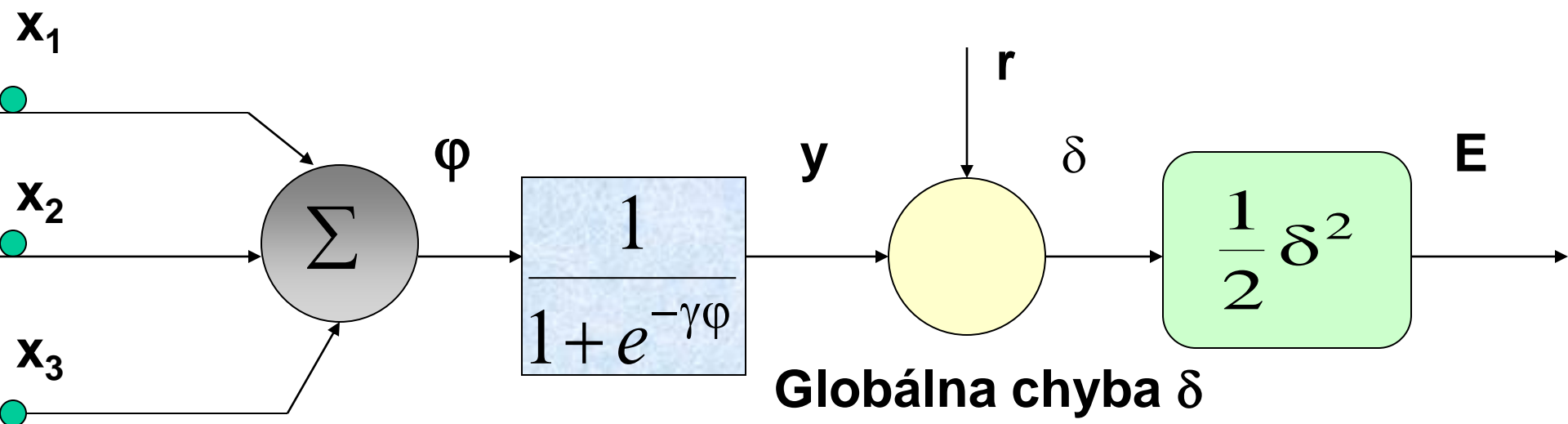
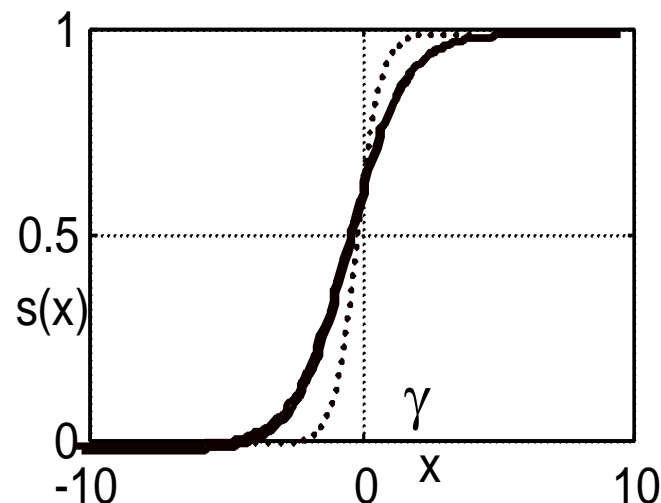
- Neurónová sieť: „študent“
- (NS) skúšanie - spôsob testovania či NS reaguje rovnako ako pri tréningu podľa tréningovej množiny (obsah-sylaby)
- zmena váhových koeficientov

Algoritmus BP:

1. náhodná inicializácia váh $i=0$
2. výber vzoru z-trenovacej množiny ($i=i+1$)
3. vstup vzoru na vstup siete
4. výpočet výstupu z NS porovnanie výst. so žiadanými hodnotami, modifikácia váh
5. test ukončenia *if $i = N$ then next step...*
6. test globálnej chyby $<$ norma -kritérium (hranica)

$$y = S\left(\sum_{i=1}^n w_i x_i + \theta\right)$$

$$y = S(\varphi) = \frac{1}{1 + e^{-\gamma\varphi}}$$



Globálna chyba δ

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

Energetická funkcia neurónu

Ciel' : minimalizácia

$$\min E = \min_{w, \theta} g(w, \theta, x)$$

Funkcia g (zložená funkcia) - hyperpriestor premenné (w_i, θ) a x je konštantný vstup

Označenie: o - výstupná vrstva

h - skrytá vrstva

i, j - indexy (i -výst. j -vstup)

- y_i^h výstup i -tého neurónu skrytej. vrstvy
- w_{ij}^o váha spoj. i -tý neurón výst. vrstvy s j -tým neurónom predch. skrytej vrstvy

- gradient energetického priestoru

- kde

$$\frac{\partial E}{\partial w_{ij}^o}$$

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

$$y = S\left(\sum_{i=1}^n w_i x_i + \theta\right)$$

$$y = S(\varphi) = \frac{1}{1 + e^{-\gamma\varphi}}$$

$$\frac{\partial E}{\partial w_{ij}^o} = \frac{\partial E}{\partial y_i^o} \frac{\partial y_i^o}{\partial \varphi_i} \frac{\partial \varphi_i}{\partial w_{ij}^o}$$

$$\frac{\partial E}{\partial y_i^o} = y_i^o - r_i$$



$$\frac{\partial y_i^o}{\partial \varphi_i^o} = \gamma y_i^o (1 - y_i^o)$$

pretože $\varphi_i^o = \sum_{j=1}^n w_{ij}^o y_j^h + \theta_i^o$

$$y = S(\varphi) = \frac{1}{1 + e^{-\gamma\varphi}}$$

$$\frac{\partial \varphi_i^o}{\partial w_{ij}^o} = y_j^h$$

Gradient

$$\frac{\partial E}{\partial w_{ij}^o} = \frac{\partial E}{\partial y_i^o} \frac{\partial y_i^o}{\partial \varphi_i^o} \frac{\partial \varphi_i^o}{\partial w_{ij}^o} = (y_i^o - r_i) \gamma y_i^o (1 - y_i^o) y_j^h$$

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

$$\frac{\partial E}{\partial y_i^o} = y_i^o - r_i$$

$$\frac{\partial E}{\partial w_{ij}^o} = \delta_i^o y_j^h$$

$$\delta_i^o$$

Gradient vo výstupnej vrstve

Vektor prírastku váh

$$\Delta W = -\eta \nabla E$$

Vzťah platí pre všetky vrstvy-váhy upravujeme vždy ako násobok zápornej hodnoty gradientu

$$w_{ij}^0(t+1) = w_{ij}^0(t) + \Delta w_{ij}^0(t)$$

Korekcia váh vo výstupnej vrstve :

$$\frac{\partial E}{\partial w_{ij}^0} = \nabla E = \delta_i^0 y_j^h$$

$$\Delta w_{ij}^0(t) = \eta \delta_i^0(t) y_j^h(t) + \alpha \Delta w_{ij}^0(t-1)$$

$$\Delta \theta_i^0 = \eta \delta_i^0(t) + \alpha \Delta \theta_i^0(t-1)$$

Koef. rýchlosti učenia

Koef. zotrvačnosti učenia

Zovšeobecnenie:

$$w_{ij}^0(t+1) = w_{ij}^0(t) + \Delta w_{ij}^0(t)$$

$$\theta_i^0(t+1) = \theta_i^0(t) + \Delta \theta_i^0(t)$$

Korekcia váh v skrytej vrstve-na základe gradientu energetickej funkcie:

$$\frac{\partial E}{\partial w_{ij}^h} = \frac{\partial E}{\partial y_k^o} \frac{\partial y_k^o}{\partial \varphi_k^o} \frac{\partial \varphi_k^o}{\partial y_i^h} \frac{\partial y_i^h}{\partial \varphi_i^h} \frac{\partial \varphi_i^h}{\partial w_{ij}^o}$$

\longleftrightarrow
 $\delta_k^o \quad w_{ki}^o$
 \longleftrightarrow
 δ_i^k

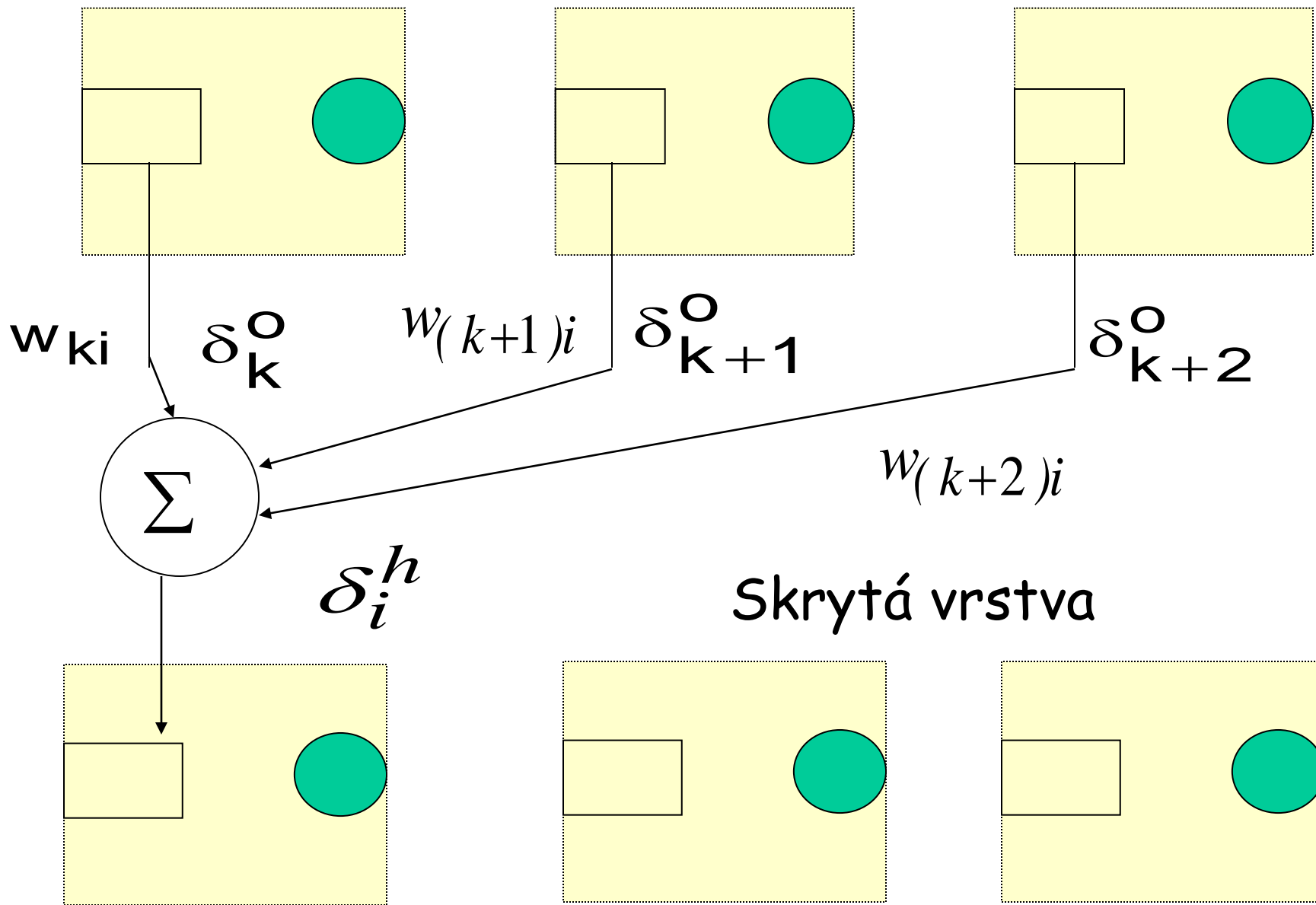
Nová hodnota chyby v skrytej vrstve δ_i^k je určená súčinom chyby vo výstupnej vrstve a príslušného váhového koeficienta – prenos chyby na skrytú vrstvu

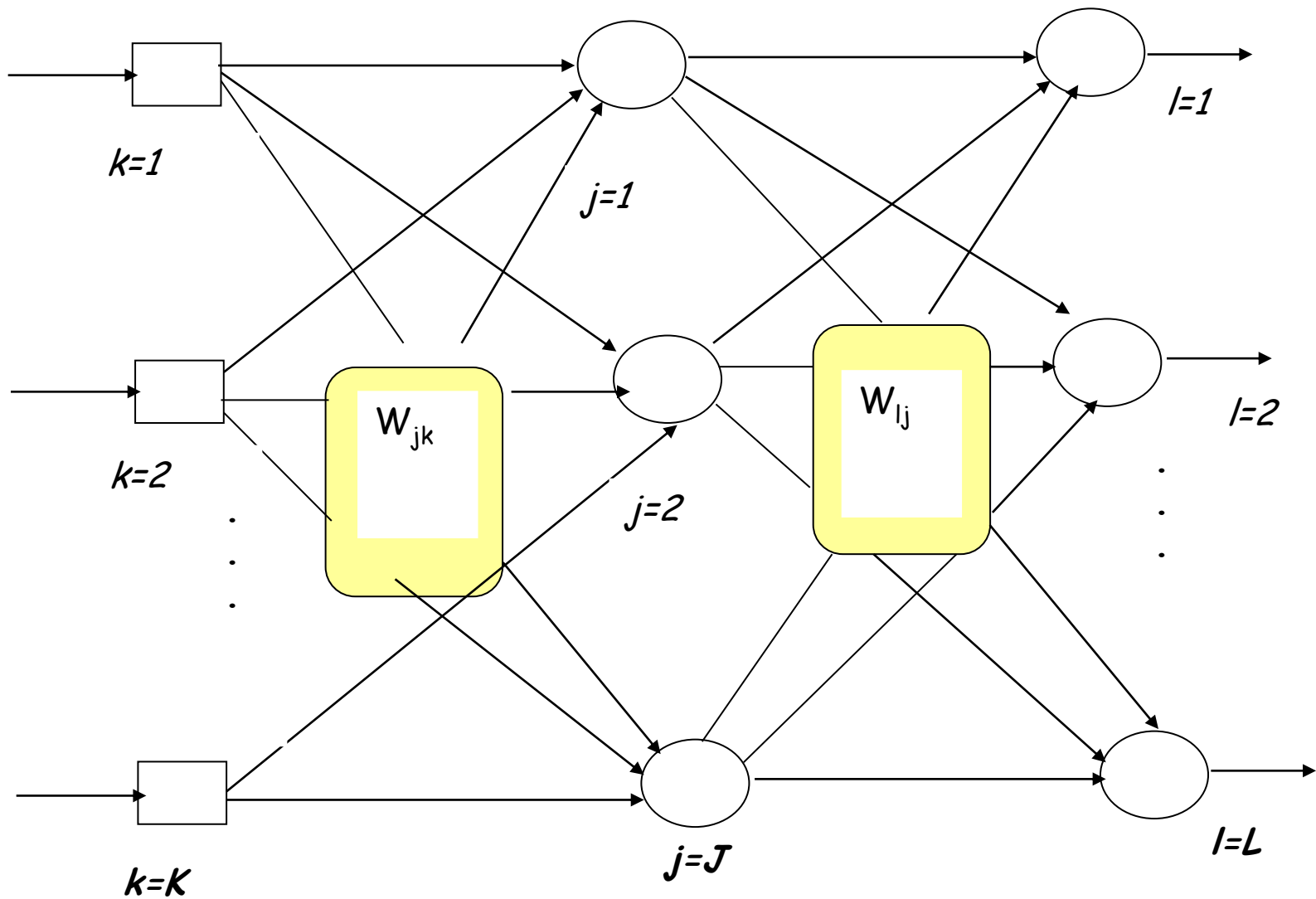
Pre viacero výstupov - skrytá vrstva má viac neurónov-výstup zo skrytej vrstvy je vetvený, t.j. hodnota chyby δ_i^k v skrytej vrstve je daná súčtom príspevkom od všetkých neurónov z výstupnej vrstvy

$$\sum w_{ki}^o \delta_k^o$$

$$\delta_i^h = y_i^h (1 - y_i^h) \sum_{k=1}^n w_{ki}^o \delta_k^o$$

Výstupná vrstva





VSTUPNÁ VRSTVA

SKRYTÁ VRSTVA

VÝSTUPNÁ VRSTVA

Korekcia váh a biasov v skrytej vrstve je realizovaná podľa vzťahov:

$$\Delta w_{ij}^h(t) = \eta \delta_i^h(t) x_j(t) + \alpha \Delta w_{ij}^h(t-1)$$

$$\Delta \theta_i^h = \eta \delta_i^h(t) + \alpha \Delta \theta_i^h(t-1)$$

$x_i(t)$ je j -tý vstup neurónovej siete pred prvou skrytou vrstvou, alebo j -tý výstup siete

$$w_{ij}^h(t+1) = w_{ij}^h(t) + \Delta w_{ij}^h(t)$$

$$\theta_i^h(t+1) = \theta_i^h(t) + \Delta \theta_i^h(t)$$

Zovšeobecnenie výpočtu chybovej funkcie:

$$\delta_i^{h-1} = y_i^{h-1} (1 - y_i^{h-1}) \sum_{k=1}^n w_{ki}^h \delta_k^h$$

Vzťah pre výpočet δ je aplikovateľný pre celú sieť od $O_u \longrightarrow I_n$

Poz. Číslovanie skrytých vrstiev sa zväčšuje smerom k výstupnej vrstve

Algoritmus BP:

1. krok INICIALIZÁCIA - VÁHY (NAHODNE V INT. < -0.3, 0.3)
Vstup vzorov (nameraných-trénovacích) do NS, výpočet výstupov neurónov smerom od IN do OU v NS podľa vzťahu:

$$y = S\left(\sum_{i=1}^n w_i x_i + \theta\right) \quad S(\varphi) = \frac{1}{1 + e^{-\varphi}}$$

2. krok Výpočet kritéria energie podľa vzťahu a výpočet chyby vo výstupnej vrstve :

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

3. krok Korekcia váh a biasov pre všetky neuróny v l-tej vrstve

$$\Delta W_{ij}^l(t) = \eta \delta_i^l(t) y_j^{l-1}(t) + \alpha \Delta W_{ij}^l(t-1)$$

$$\Delta \theta_i^l = \eta \delta_i^l(t) + \alpha \Delta \theta_i^l(t-1)$$

Spätne šírenie chyby a modifikácia váh a biasov

$$\delta_i^{h-1} = y_i^{h-1} (1 - y_i^{h-1}) \sum_{k=1}^n w_{ki}^h \delta_k^h$$

$$w_{ij}^l(t+1) = w_{ij}^l(t) + \Delta w_{ij}^l(t)$$

$$\theta_i^l(t+1) = \theta_i^l(t) + \Delta \theta_i^l(t)$$

Krok 4 opakujeme pre všetky vrstvy, start $l=0$, až $l=h$