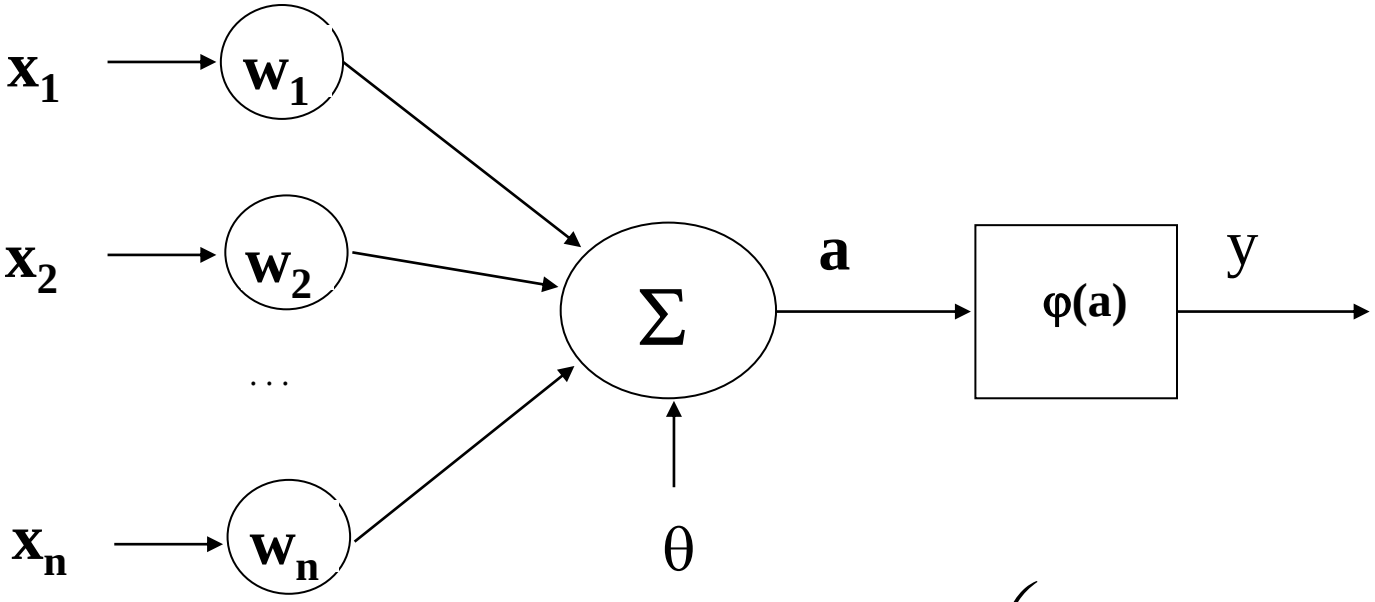
3 Umelé neurónové siete v modelovaní riadení procesov

3.1 Viacvrstvové perceptrónové siete (Multilayer Perceptron net - MLP)

Matematický (počítačový) neurón

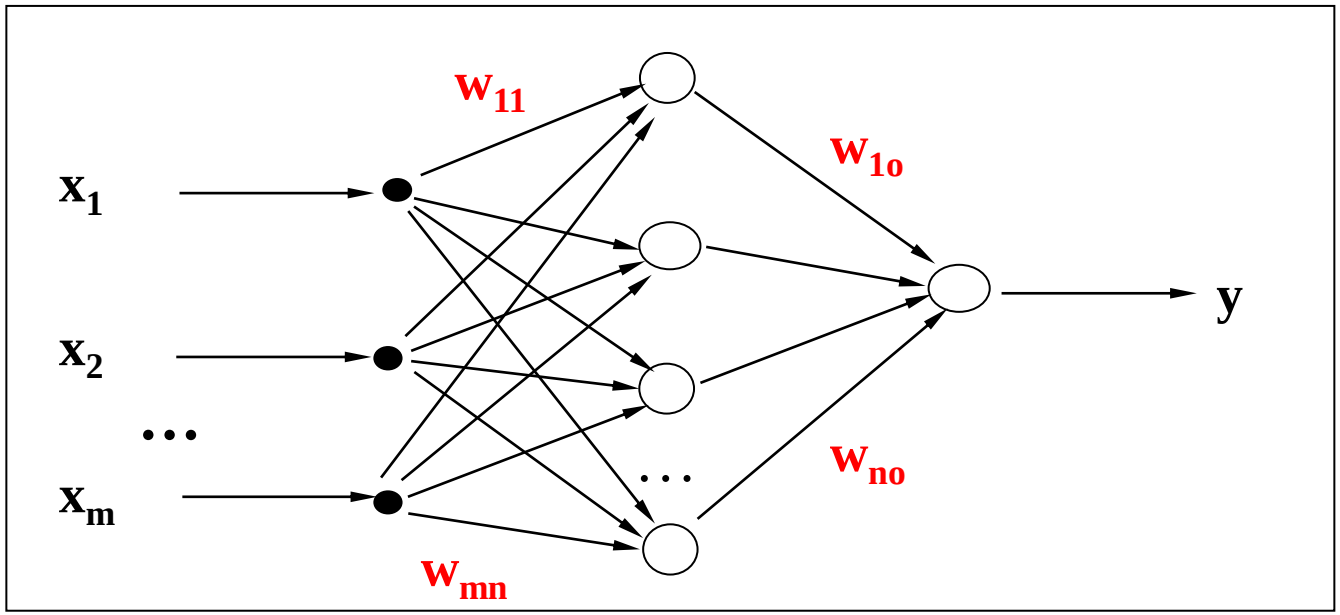


$$y = \varphi\left(\sum_{i=1}^{n}(w_i x_i) - \theta\right) = \varphi(a)$$

x_i - vstupy neurónu
w_i - váhy synaptických spojení
θ - prah (citlivosti) neurónu
a - vnútorná aktivita neurónu
φ - aktivačná funkcia neurónu
y - výstup neurónu

Viacvrstvové perceptrónové siete

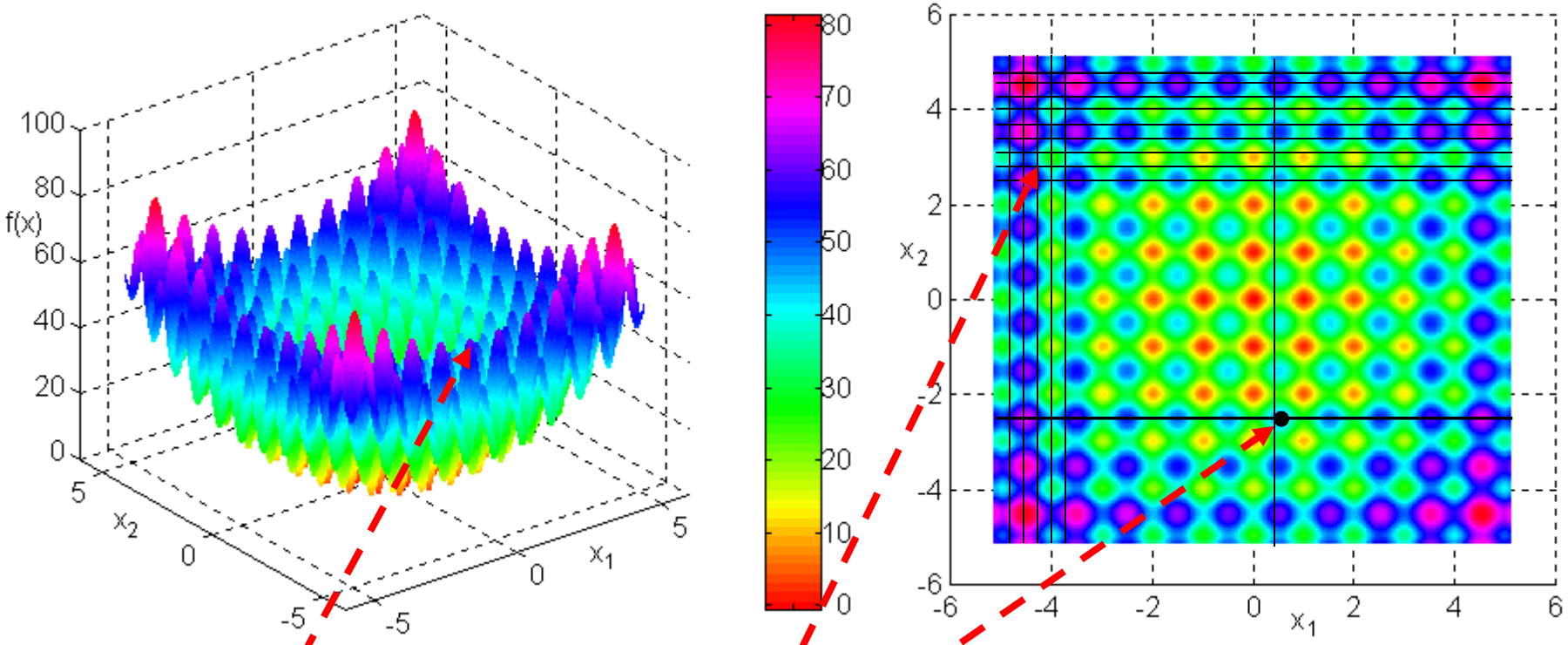
- **Obsahujú aspoň jednu skrytú vrstvu neurónov.**
- **Obsahujú v skrytej vrstve spojitú nelineárnu aktivačnú funkciu (obyčajne sigmoidu alebo hyperbolický tangens). Vo výstupnej vrstve obsahujú rovnakú aktivačnú f. alebo aj lineárnu.**
- **Sú schopné aproximovať ľubovoľnú nelineárnu transformáciu.**
- **Parametrizácia (trénovanie) takýchto sietí sa realizuje algoritmami na báze metódy "spätného šírenia chyby„ (učenie s učiteľom), učením s posiľňovaním (reinforcement learning) alebo neuro-evolúciou (bez učiteľa).**



$y=f(x)$

Trénovanie MLP :
Algoritmus spätného šírenia chyby
(„Back- Propagation“ algorithm)

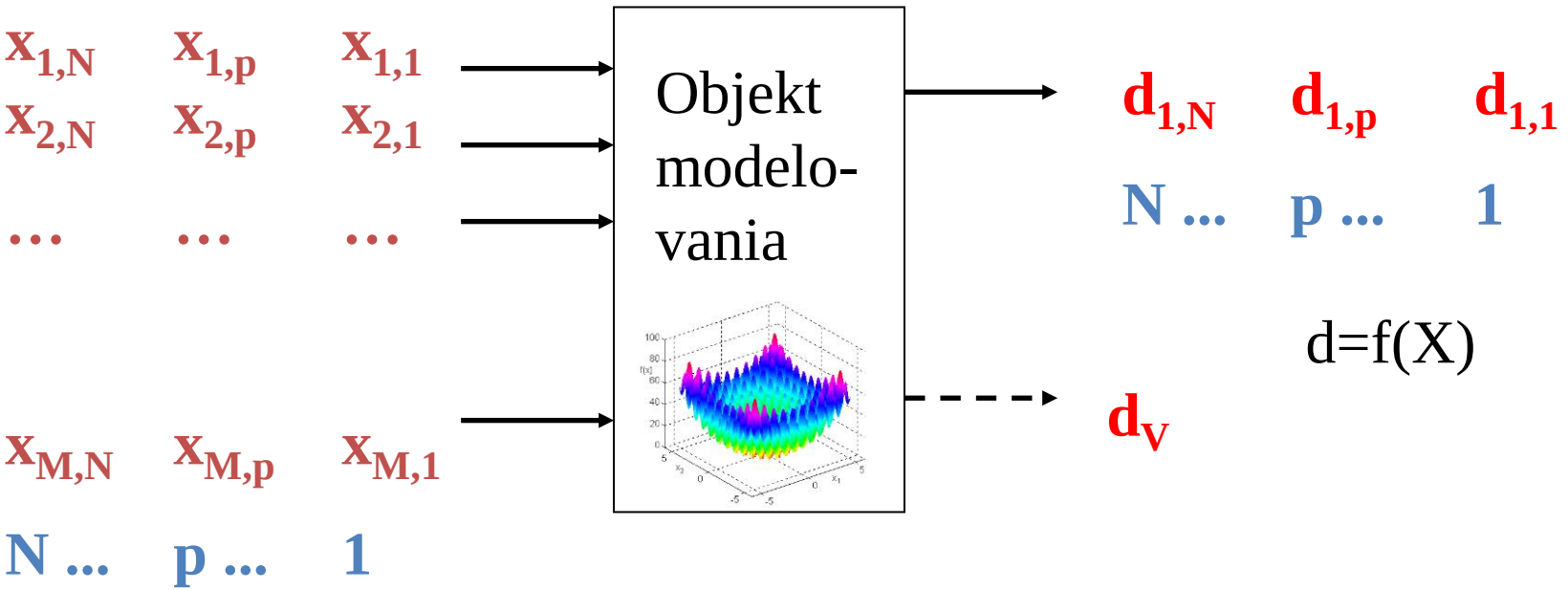
Aproximácia nelineárnej funkcie pomocou UNS

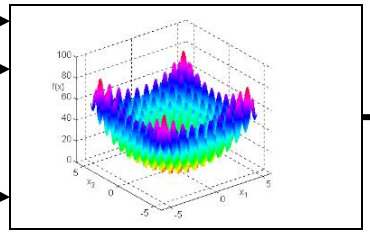


Definujeme sieť predlôh (trénovacích vzorov), ktoré sú tvorené vektormi nezávisle premenných x=[x1,x2, …] a im zodpovedajúcich hodôt y=f(x) – vstupno/výstupné dáta.

Predpokladajme trénovaciu množinu (vstupno/výstupných) dát $D=\{x_{pq},d_{pr}\}$;

p=1,...,N je počet vzoriek
q=1,...,M je počet vstupov siete
r=1,...,V je počet výstupov siete





d

\+

\-

ε

x_1 x_1 x_1

x_2 x_2 x_2

… … …

x_M x_M x_M

p+2 p+1 p

vzorka

NS

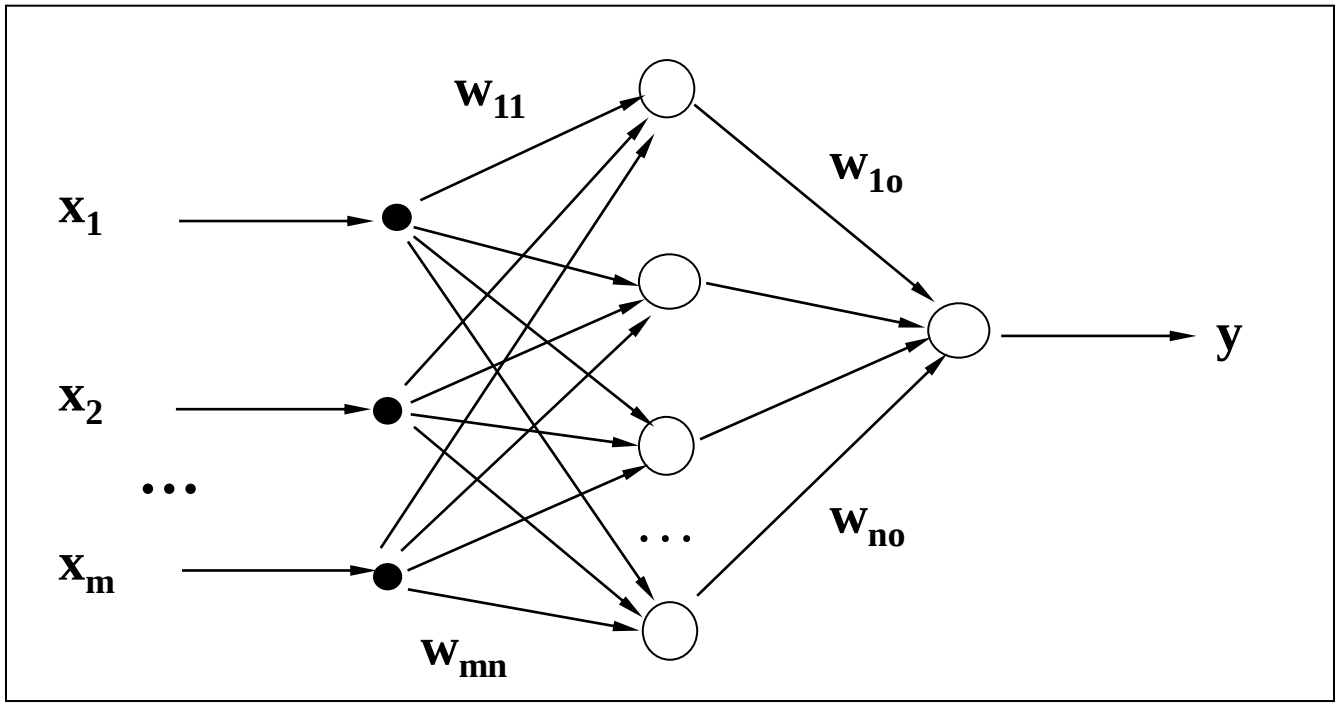
y y y

p+2 p+1 p

1 epocha: p=1,2,…,N

Trénovanie (parametrizácia) neurónovej siete prebieha vo viacerých cykloch („vlnách“) - tzv. EPOCHÁCH.

V každej epoche sa postupne cez sieť prešíri celá postupnosť trénovacích vzoriek x_p (postupnosť vstupných vektorov), porovnáva sa so skutočnými výstupmi a na základe odchýliek sa upravujú parametre siete).



w_{ij}=?

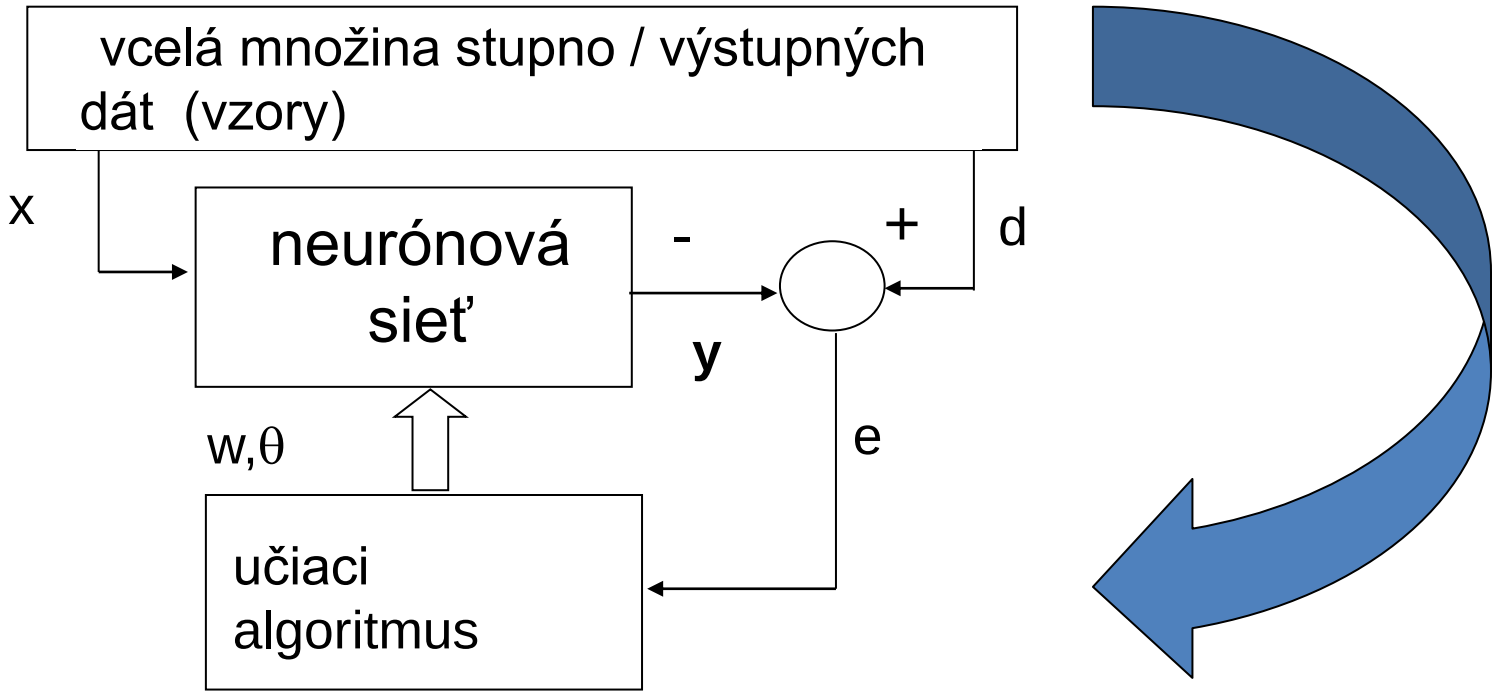
$$E = \sum_{p=1}^{N} \varepsilon_p^2 \leq chyba \qquad \varepsilon=(y-d)^2$$

N - počet vzoriek trénovacej množiny,

d – vzor, y – zodpovedajúci výstup modelu

Výpočet váh w_{ij} sa uskutočňuje iteračným algoritmom, kde sa postupne počítajú korekcie každej váhy Δw_{ij} ; $w_{ij}(t)=w_{ij}(t-1)+\Delta w_{ij}$

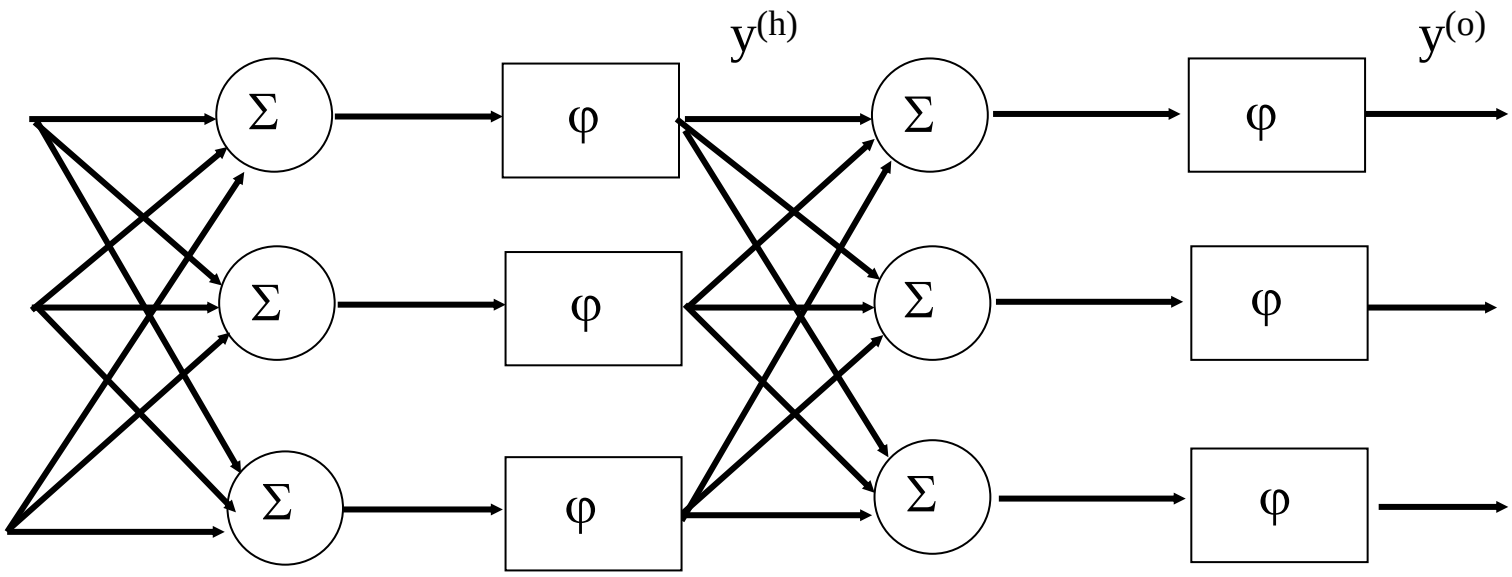
Algoritmus trénovania UNS



Globálna chyba siete:

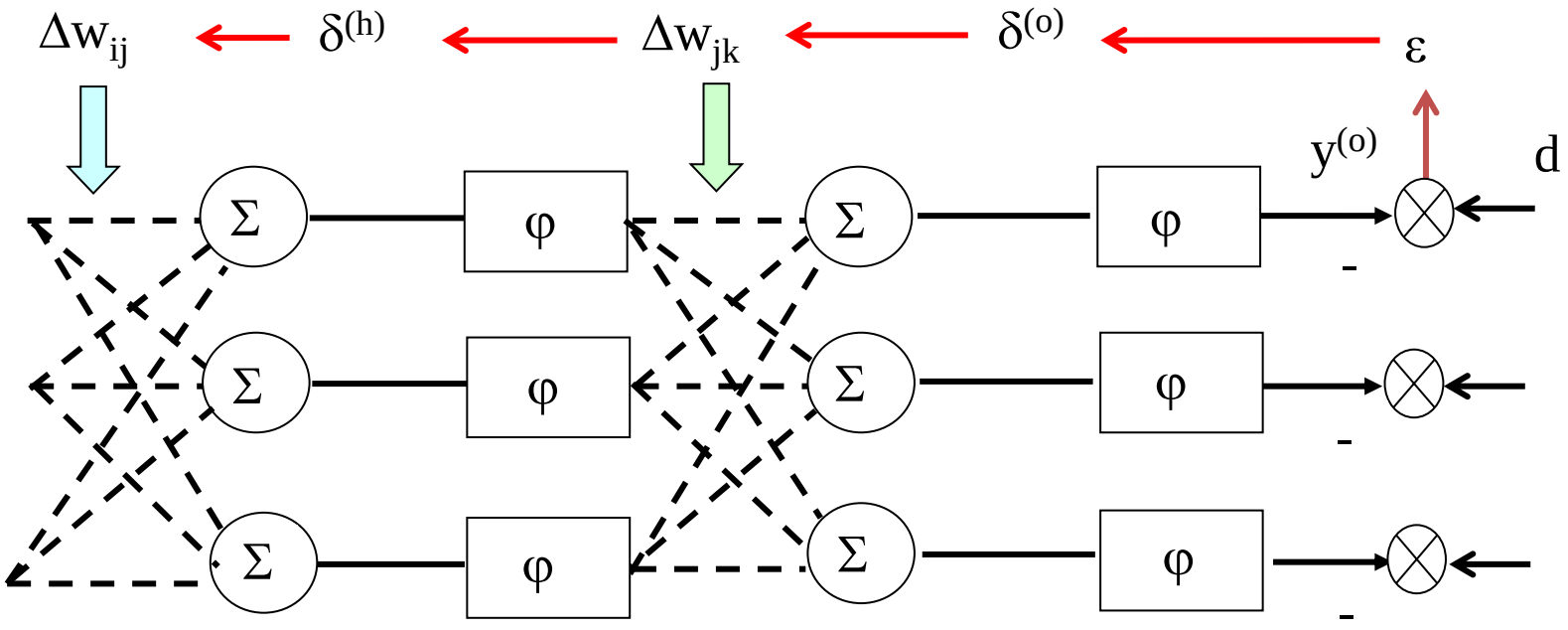
$$E = \sum_{p=1}^{N} \varepsilon_p^2 \leq chyba$$

Dopredná fáza širenia signálov v neurónovej sieti



(h) – index neurónov skrytej vrstvy (hidden),
(o) – index neurónov výstupnej vrstvy (output)

Spätná fáza šírenia signálov v neurónovej sieti

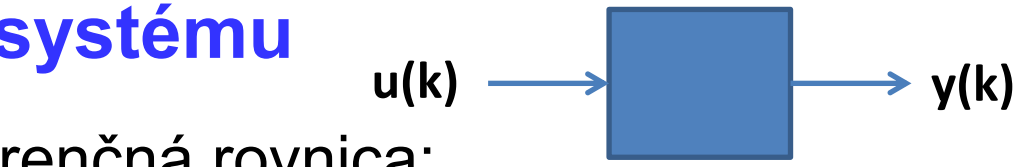


Ukončenie procesu trénovania

a) Dosiahnutie predpísanej presnosti modelu (globálnej chyby)
b) Uskutočnenie predpísaného počtu epoch trénovania
c) Alebo po úspešnom teste zovšeobecňovacej schopnosti n.s.

3.2 Použitie viacvrstvovej perceptrónovej siete na modelovanie nelineárnych dynamických systémov

Model dynamického systému



Lin. diskrétny systém – diferenčná rovnica:

$$\hat{y}(k)=f\,(\,y(k-1), y(k-2), \ldots, y(k-n), u(k-1), \ldots, u(k-m), \ldots\,)$$

alebo

$$\hat{y}(k)=f\,(\,y(k-1), y(k-2), \ldots, y(k-n), u(k-d), \ldots, u(k-d-m), \ldots\,)$$

ŷ(k) – výstup modelu, k - krok vzorkovania

f(.) – ne / lineárna funkcia

d≥1 – dopravné oneskorenie

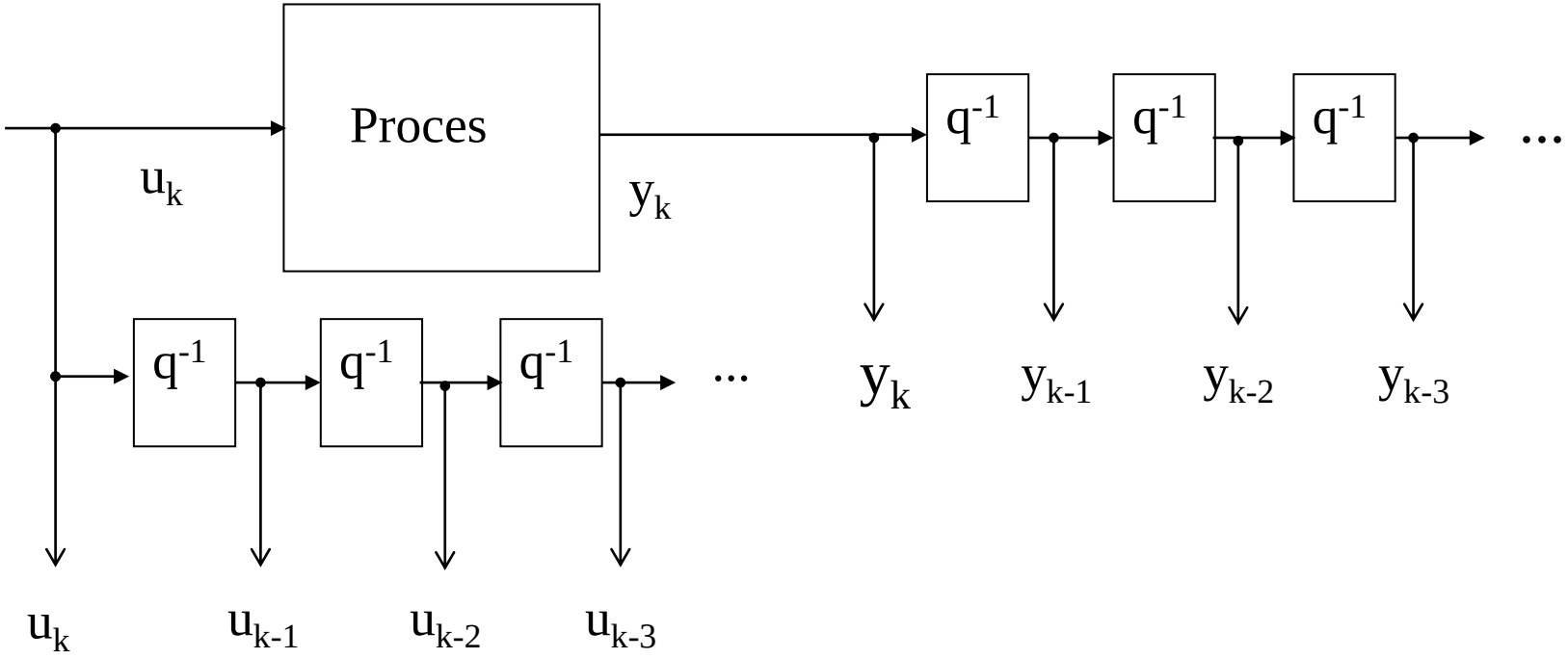
Lin. diskrétny systém – prenosová funkcia:

$$S(z^{-1})=[b_m(z^{-m})+\ldots b_1(z^{-1})+b_0] \,/\, [a_n(z^{-n})+\ldots+a_1(z^{-1})+a_0]$$

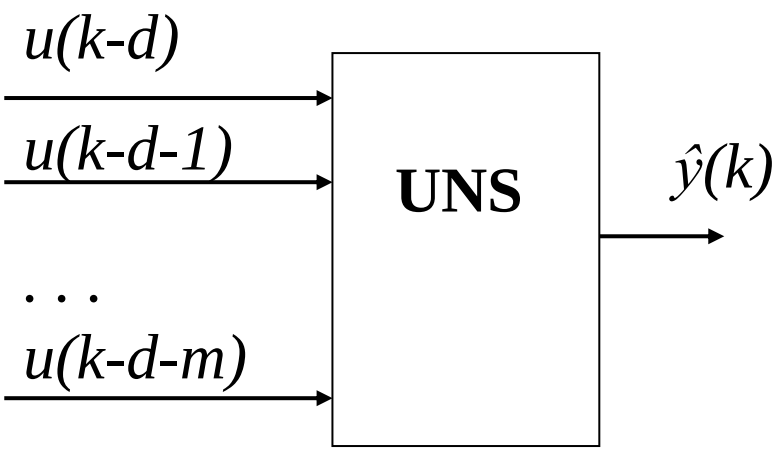
Generovanie V/V signálov dyn. systému

vybudenie systému – u

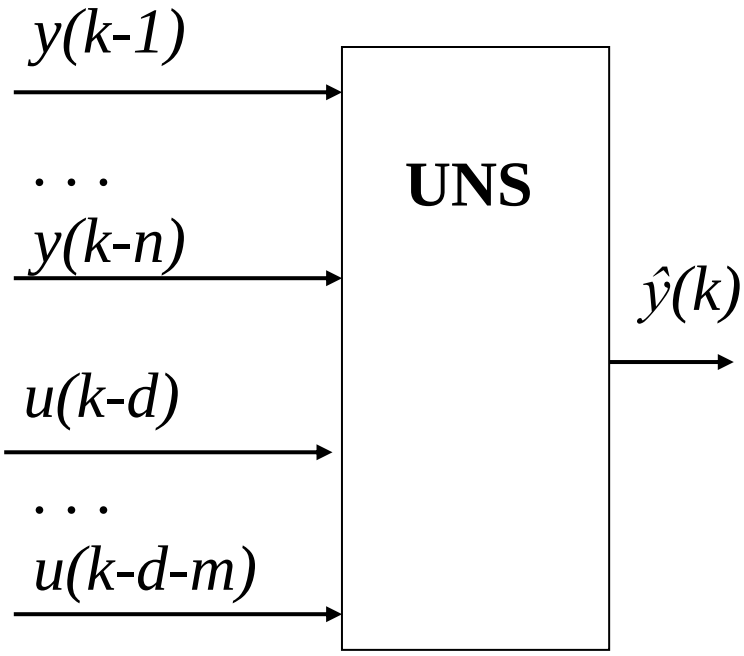
výstup systému



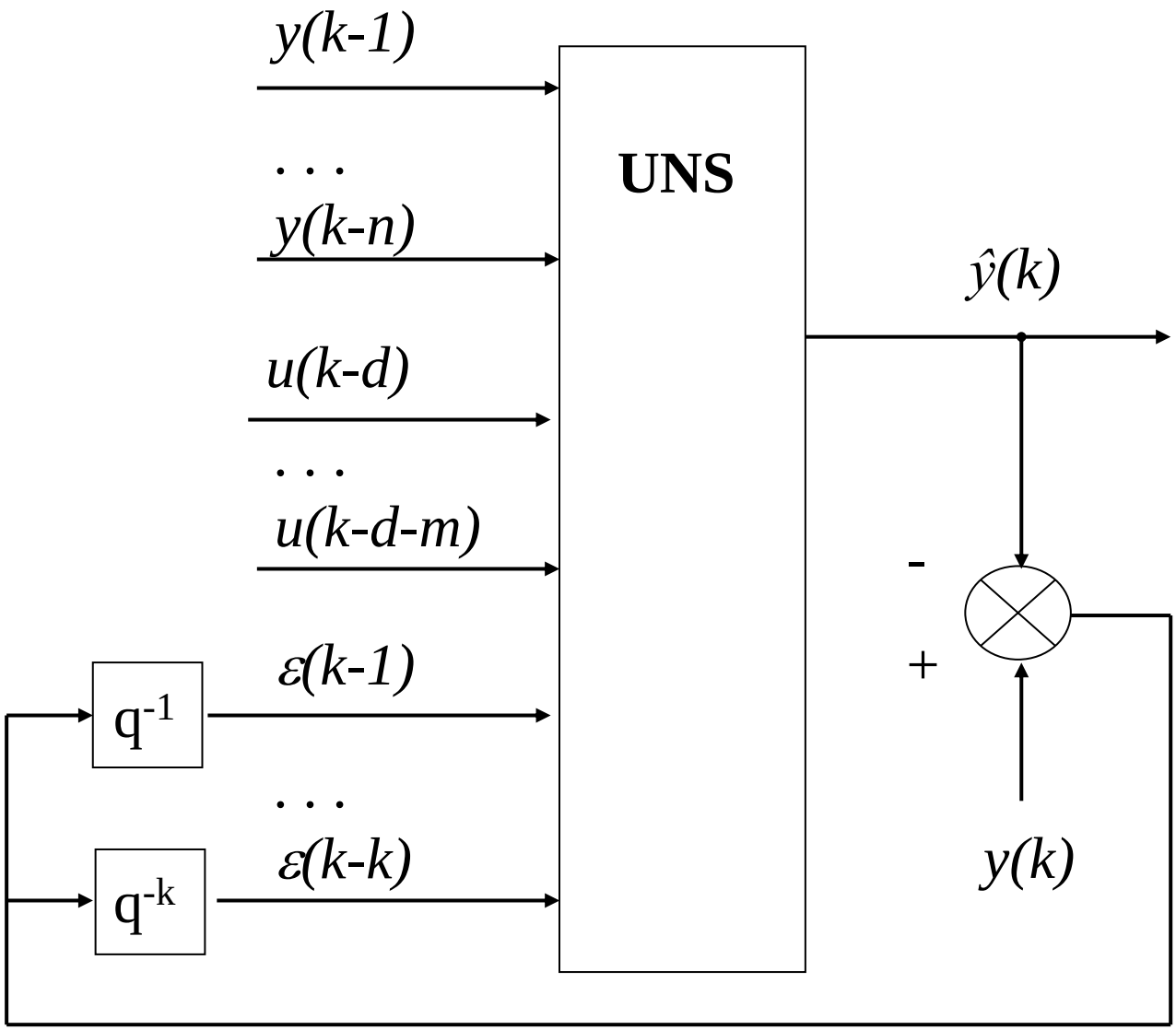
Typy dynamických modelov



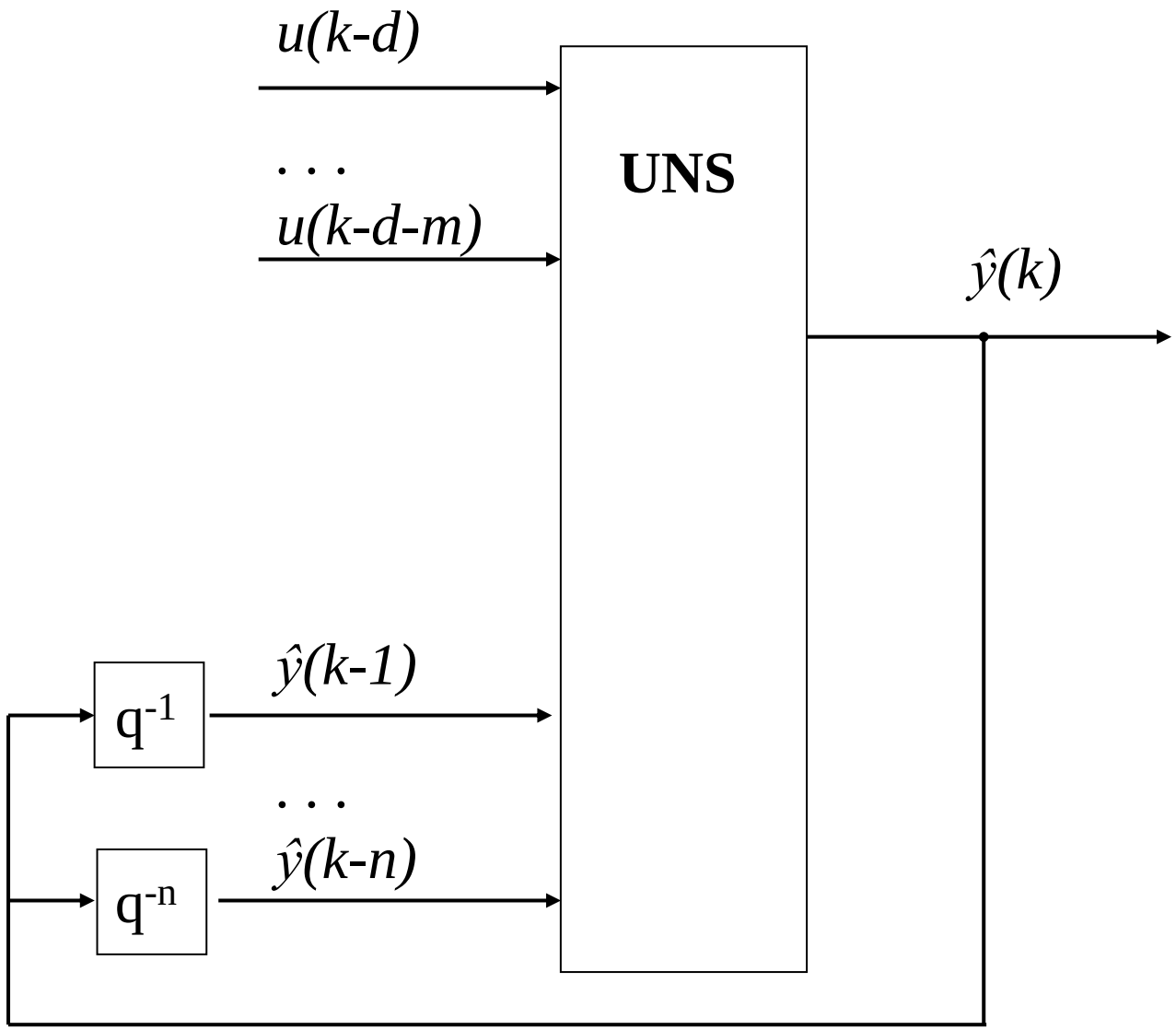
NNFIR



NNARX



NNARMAX



NNOE

Chyba modelu

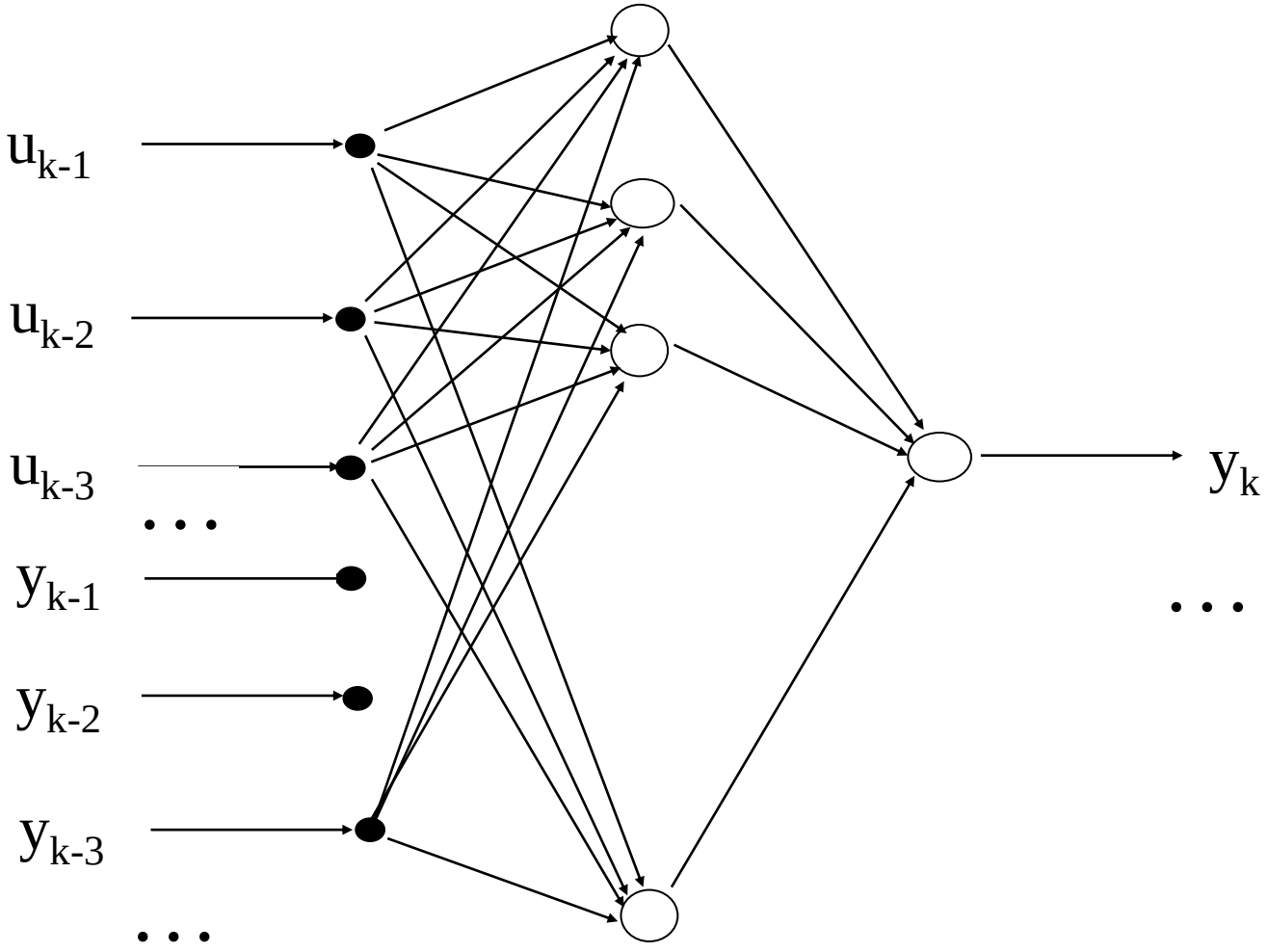
$$E = \frac{1}{2N} \sum_{t=1}^{N} \left[y(k) - \hat{y}(k) \right]^2 \rightarrow \min$$

y – *namerané dáta*

$\hat{y}$ - *modelované dáta*

Učenie s učiteľom, V/V dáta sú k dispozícii.
Backpropagation algoritmus

Vstupy a výstupy modelu dynamického systému s UNS (NNARX)



Zlepšenie presnosti dynamického modelu pomocou UNS

- zvýšenie počtu V/V dát (dlhší experiment, kratšia perióda vzorkovania)
- zväčšenie počtu neurónov v skrytej vrstve
- zvýšenie počtu oneskorení vstupných signálov (zvýšenie „rádu“ systému)
- zvýšenie počtu skrytých vrstiev
- použitie iných typov UNS (rekurentné siete, RBF, iné typy, hlboké siete...)

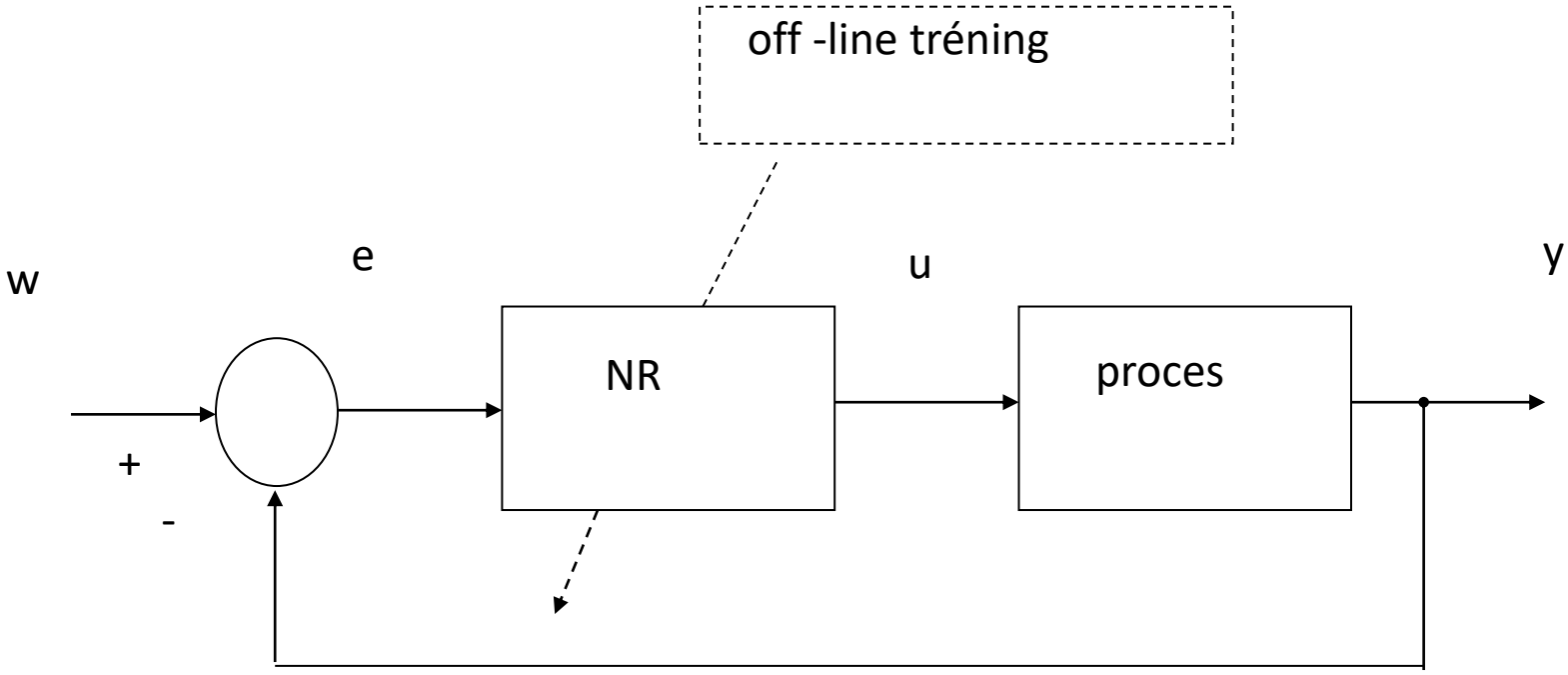
3.3 Neurónové siete v regulačných obvodoch

3.3.1 UNS ako priamy regulátor

Umelá neurónová sieť plní funkciu priameho regulátora, generuje riadiacu (rozhodovaciu) veličinu na priame ovplyvňovanie objektu riadenia.

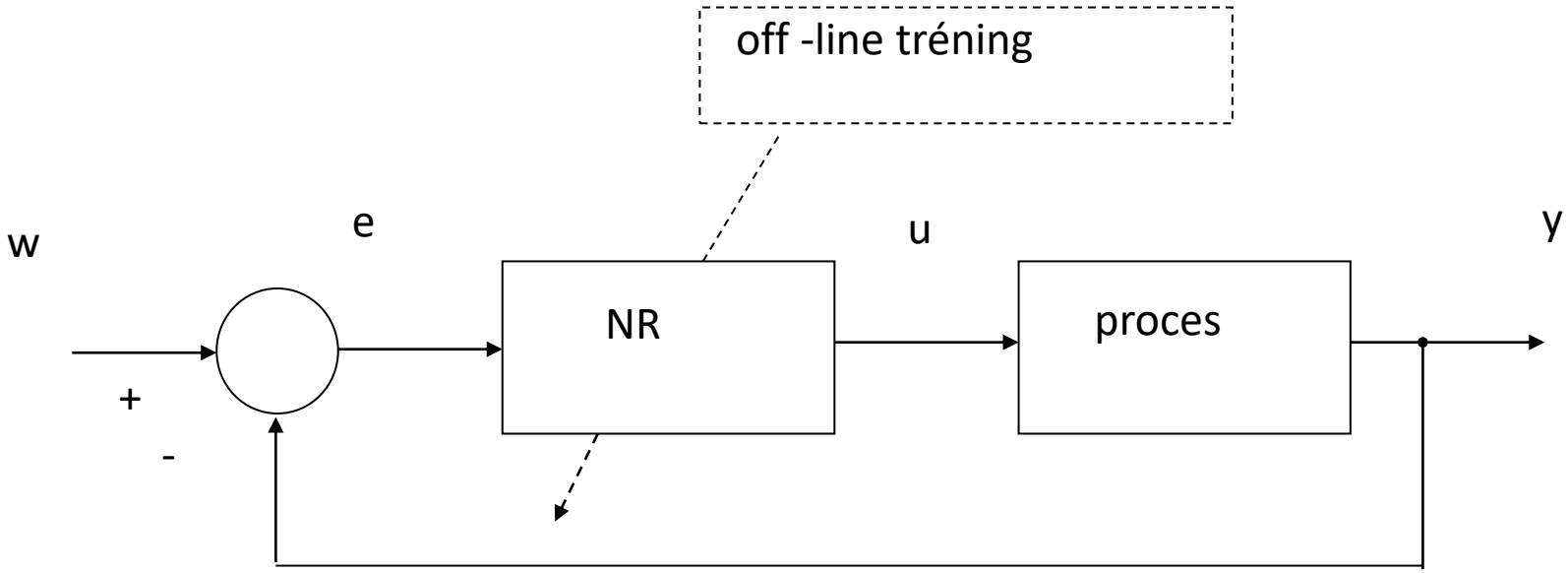
Napodobenie iného typu regulátora

Ak je k dispozícii funkčný algoritmus riadenia, ktorý môže byť výpočtovo veľmi náročný, problematicky realizovateľný ... , UNS si na základe dostupných vstupných a výst. informácií natrénuje jeho správanie a potom ho môže rovnocenne nahradiť napr. jednoduchším hardvérom, znížením výpočtovej náročhosti ... V/V dáta sú k dispozícii.



Napodobenie ľudského experta

UNS modeluje na základe dostupných V/V dát vypozorované správanie sa skúseného ľudského experta (ťažko algoritmizovatelné postupy riadenia, znalosti, skúsenosti, štatisticky vyhodnotené dáta ...)



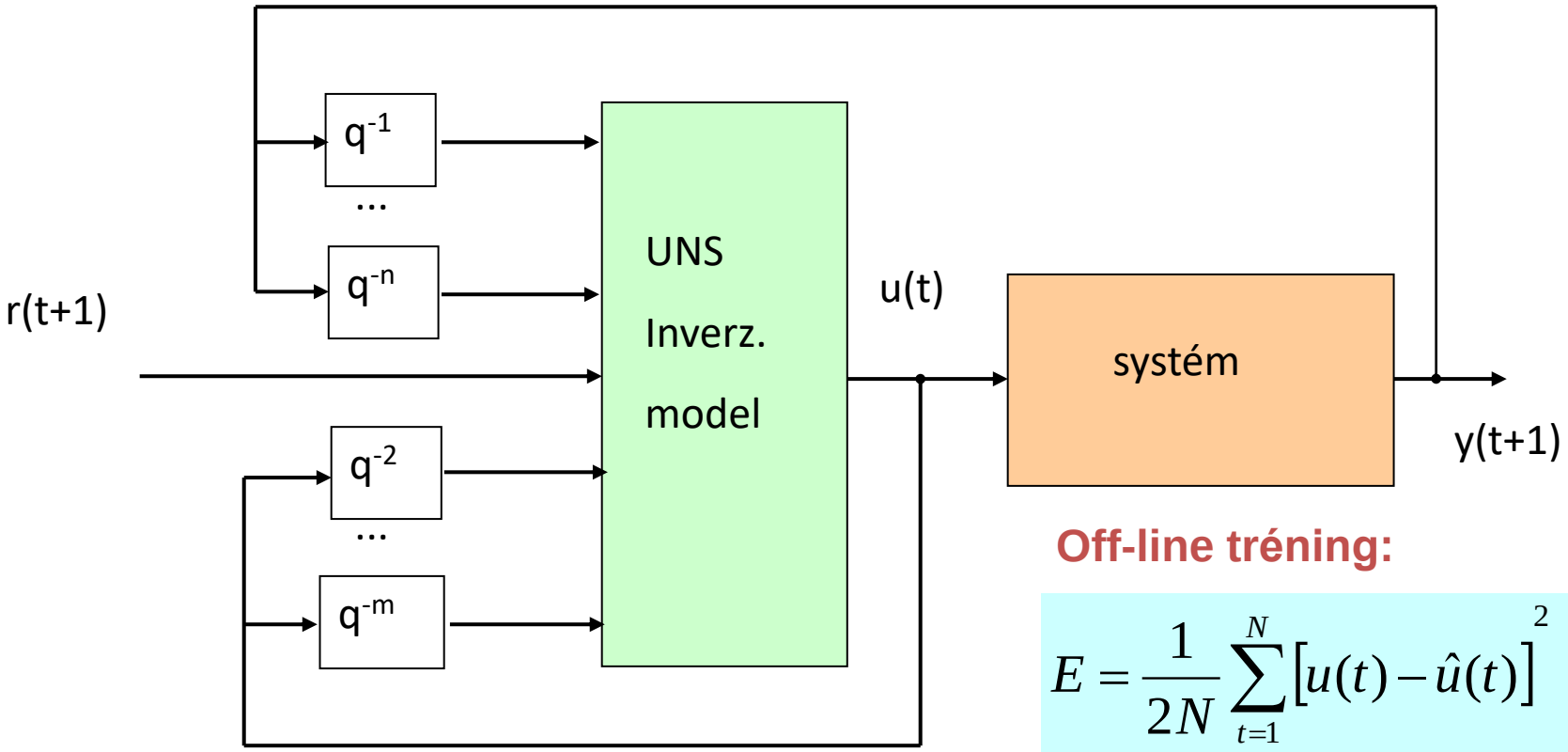
Priame inverzné riadenie

systém:

$$y(t+1)=f\,(\,y(t),\,y(t-1),\,\ldots,\,y(t-n+1),\,u(t),\,\ldots,\,u(t-m)\,)$$

neurónový regulátor - natrénovanie inverz. modelu:

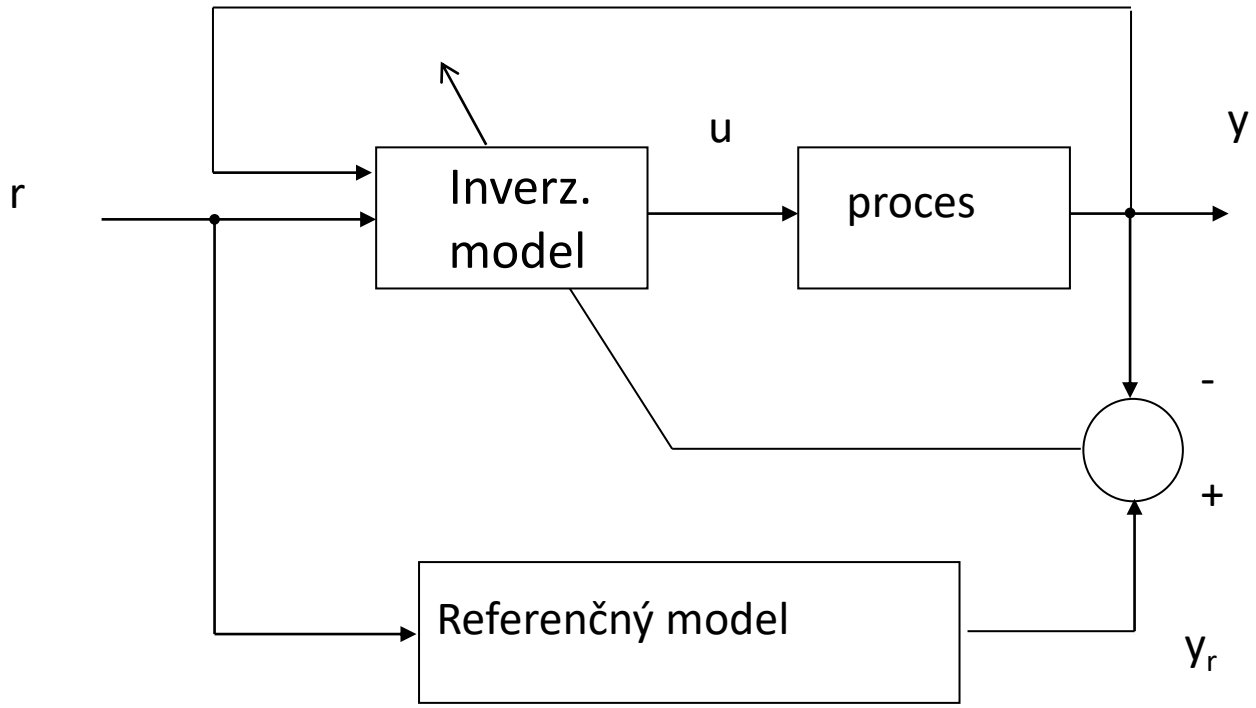
$$\hat{u}(t)=f^{-1}\,(\,y(t+1),\,y(t),\,\ldots,\,y(t-n+1),\,u(t-1),\,\ldots,\,u(t-m)\,)$$



Off-line tréning:

$$E=\frac{1}{2N}\sum_{t=1}^{N}\left[u(t)-\hat{u}(t)\right]^{2}\rightarrow\min$$

On-line verzia priameho inverzného riadenia



On-line tréning:

$$E = \frac{1}{2N}\sum_{t=1}^{N}\left[yr(t) - y(t)\right]^2 \rightarrow \min$$

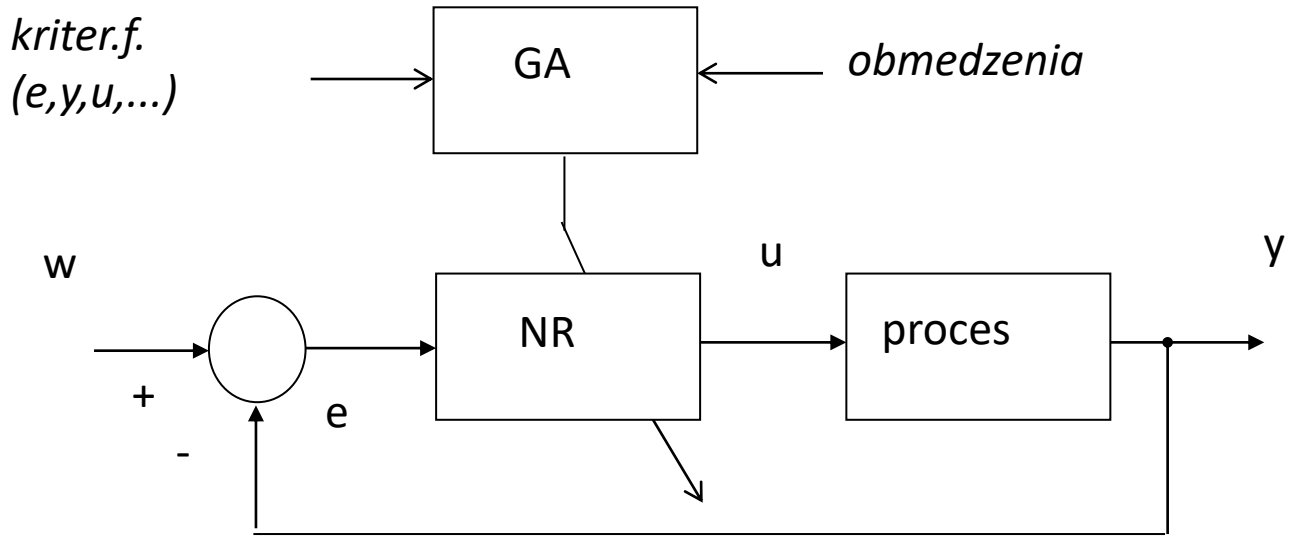
Výhody priameho inverzného riadenia

- **Jednoduché**
- **Dobrá schopnosť sledovania referenčného signálu**

Nevýhody priameho inverzného riadenia

- **Nepoužíva regulačnú odchýlku**
- **Nepracuje, ak inverzný model je nestabilný, ale aj silne kmitavý**
- **Nedostatok parametrov na ladenie**
- **Veľká citlivosť na poruchy a šum**

NR navrhnutý/optimalizovaný pomocou GA (neuroevolúcia)



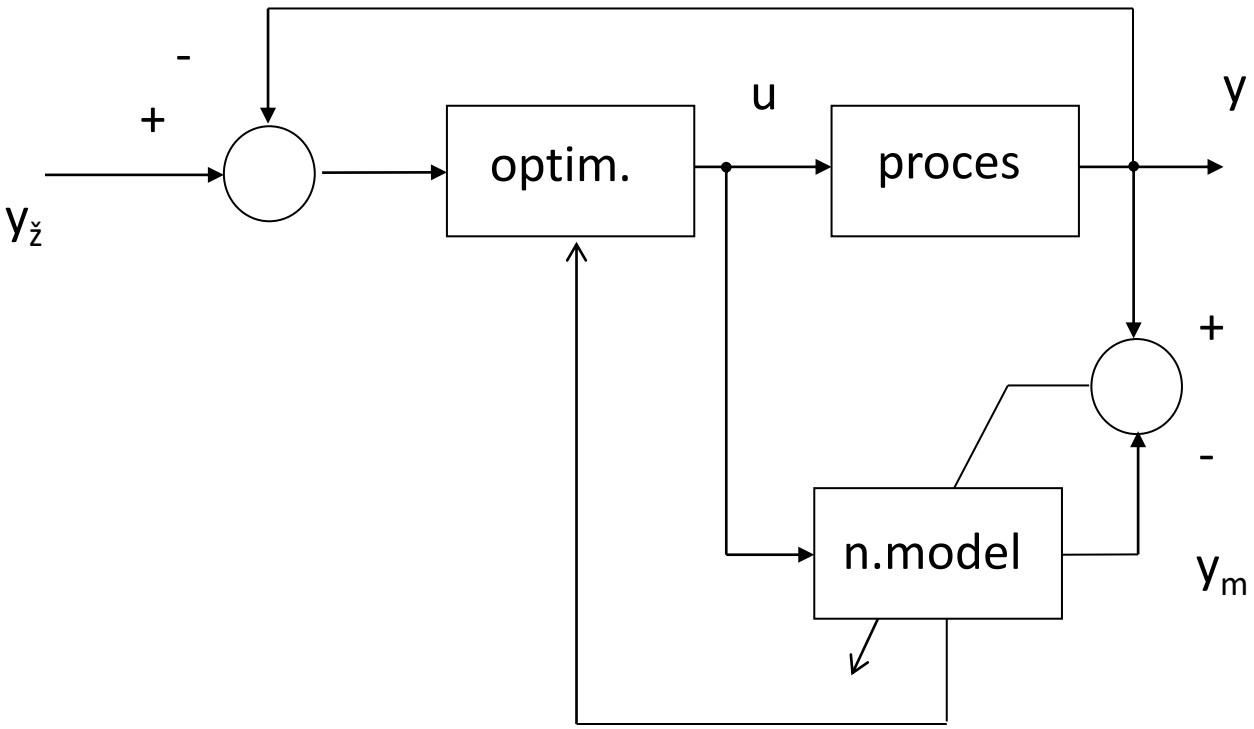
Učenie bez učiteľa, V/V dáta nie sú disponibilé. Minimalizuje sa vhodná kriteriálna funkcia.

Postup návrhu / optimalizácie je obdobný ako pri návrhu iných regulátorov (napr. PID) pomocou GA. Gény chromozómu sú všetky váhy a prahy UNS.

3.3.2 Hybridné štruktúry riadenia s UNS

Riadiace štruktúry na báze UNS sú spojené s inými typmi riadenia, väčšinou ho dopĺňajú, plnia funkciu modelu, pomocnú funkciu, zlepšujú ho alebo ho adaptujú.

Neuro-prediktívne riadenie



optim. - optimalizácia budúceho správania reg. obvodu

n.model - dostatočne presný neurónový model procesu (trénovaný off-line, on-line)

Predpokladajme, že máme k dispozícii vhodný model objektu riadenia, pomocou ktorého dokážeme predikovať budúce správanie sa systému na základe súčasných a minulých stavových (výstupných) veličín.

Prediktívny algoritmus potom pracuje na základe nasledovného cyklu:

1. Odhad potrebného počtu (1 alebo aj viacerých) budúcich krokov výstupu systému na základe priebehu predpokladaného referenčného signálu.
2. Výpočet budúcich krokov riadiaceho zásahu tak, aby bolo minimalizované zvolené kritérium J.
3. Aplikácia prvého vypočítaného kroku riadenia a skok na bod 1.

Najčastejšia forma kritéria je v tvare

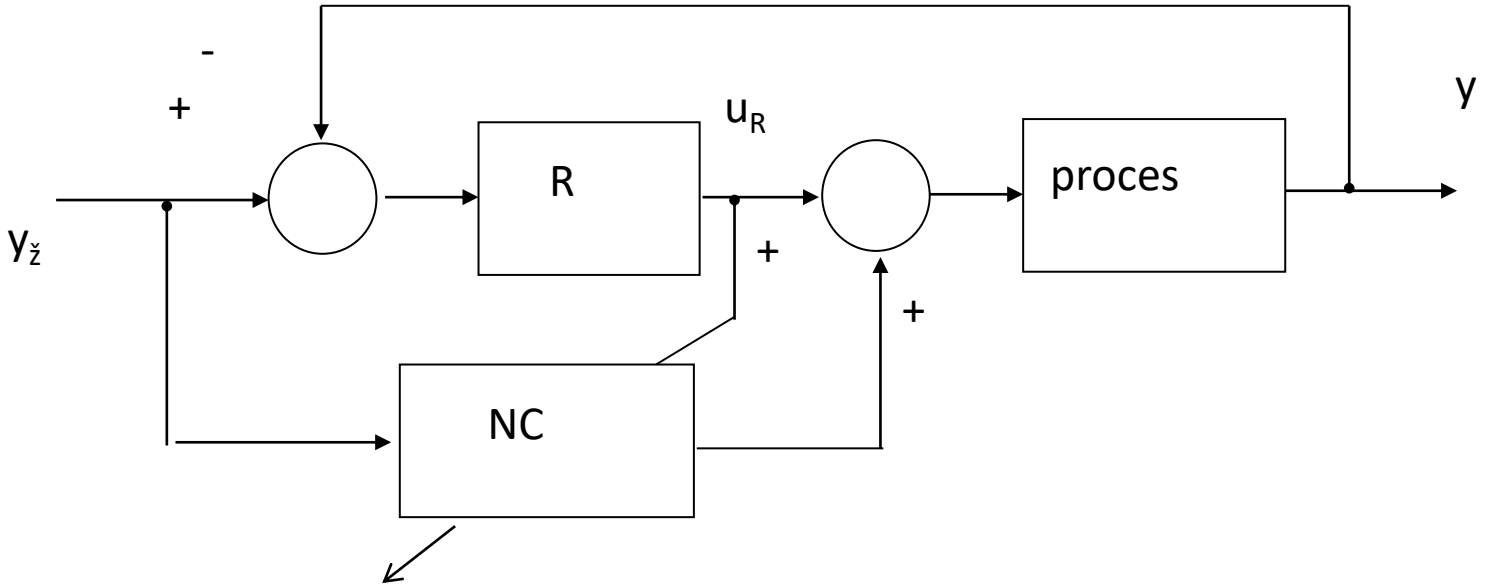
$$J = \sum_{i=N_1}^{N_2}[r(k+i) - \hat{y}(k+i)]^2 + \rho \sum_{i=1_1}^{Nu}[\Delta u(k+i-1)]^2 \rightarrow \ min$$

$$\Delta u(k+i) = 0 \qquad N_u \leq i \leq N_2$$

N_1 - dolný predikčný horizont
N_2 - horný predikčný horizont
N_u - horizont predikcie riadenia
r - predpokladaný referenčný signál (žiad. hod.)
$\hat{y}$ - predikovaný výstup procesu
ρ - váha tlmenia zmien výstupu

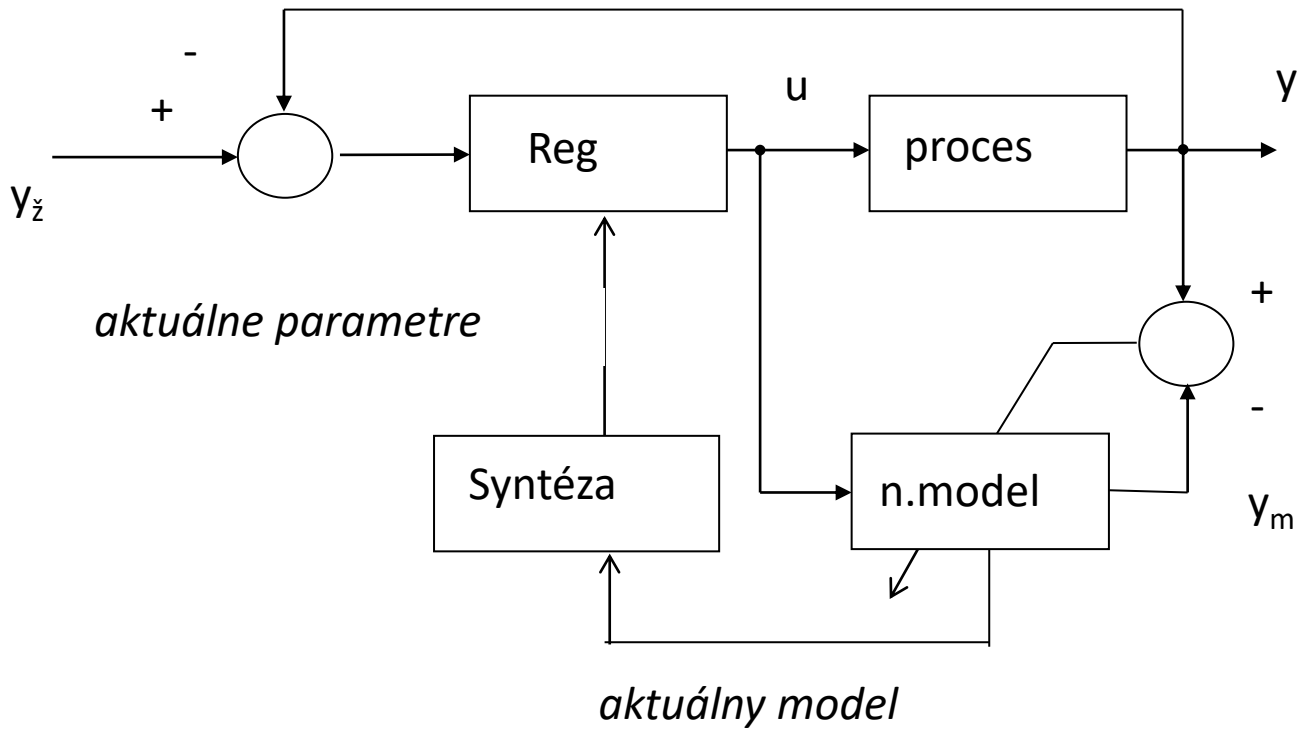
Cieľom algoritmu je zabezpečiť požadované správanie sa systému počas budúcich predpokladaných N_2 krokov.

Dopredný korekčný člen



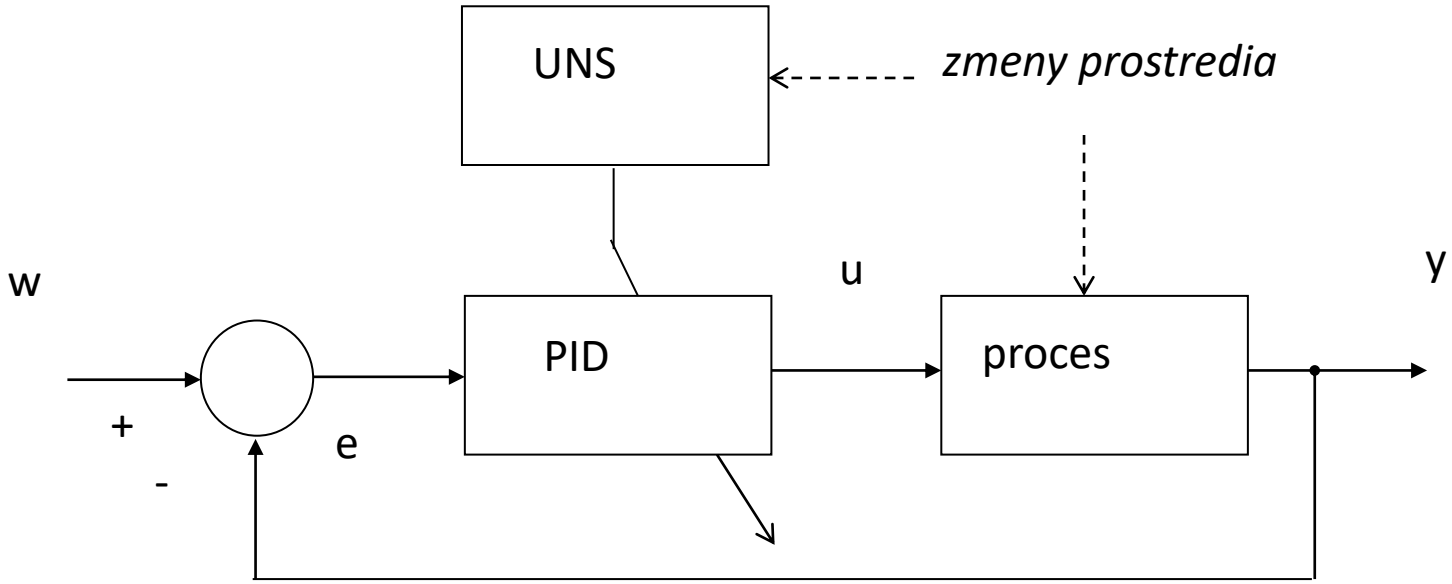
Neurónová sieť je vo funkcii dopredného korekčného člena (NC). Klasický regulátor (R) nemusí optimálne riadiť, ale musí zabezpečiť aspoň stabilizáciu procesu. Neurónový korektor je trénovaný minimalizáciou u_R (pomocou Back-Propagation). Pri regulácii potom minimalizáciou riadiaceho zásahu u_R optimalizuje regulačný pochod. Pokiaľ u_R obsahuje nenulovú ustálenú zložku (z integrátora), táto je na vstupe NR kompenzovaná.

Samonastavujúci sa regulátor ("selftuning")



Neurónový model je identifikovaný off-line alebo on-line aby dostatočne presne modeloval zmeny v riadenom procese. Na základe aktuálneho modelu sú zvolenou metódou syntézy aktualizované parametre regulátora. Možným prístupom návrhu regulátora sú genetické algoritmy.

Adaptácia klasického regulátora pomocou UNS



Neurónová sieť má natrénovanú závislosť medzi zmenami prostredia (napr. pracovnými bodmi systému, pôsobením poruchových veličín...) a zmenami parametrov regulátora (napr. PID). V závislosti od zmeny stavu prostredia sa potom adaptujú parametre regulátora (podobne ako fuzzy-gain-scheduling).

Iné štruktúry ...