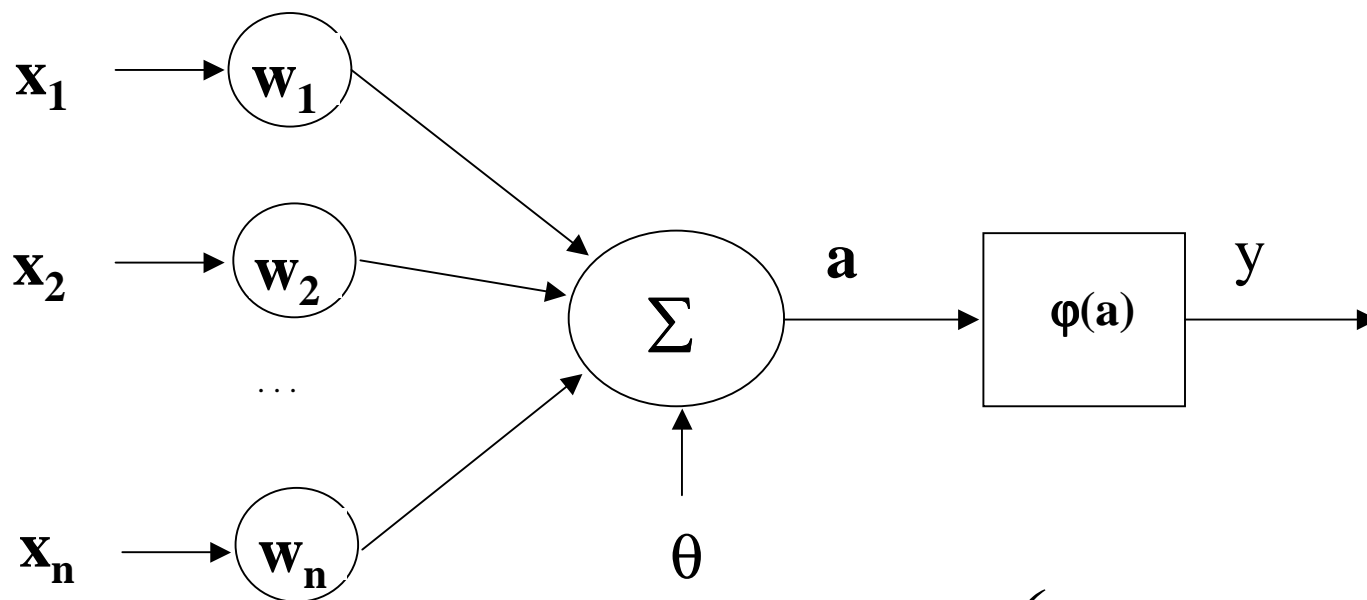


***3.5.2 Viacvrstvové
perceptrónové siete
(Multilayer Perceptron net - MLP)***

Matematický (počítačový) neurón



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

x_i - vstupy neurónu

θ - prah (citlivosti) neurónu

φ - aktivačná funkcia neurónu

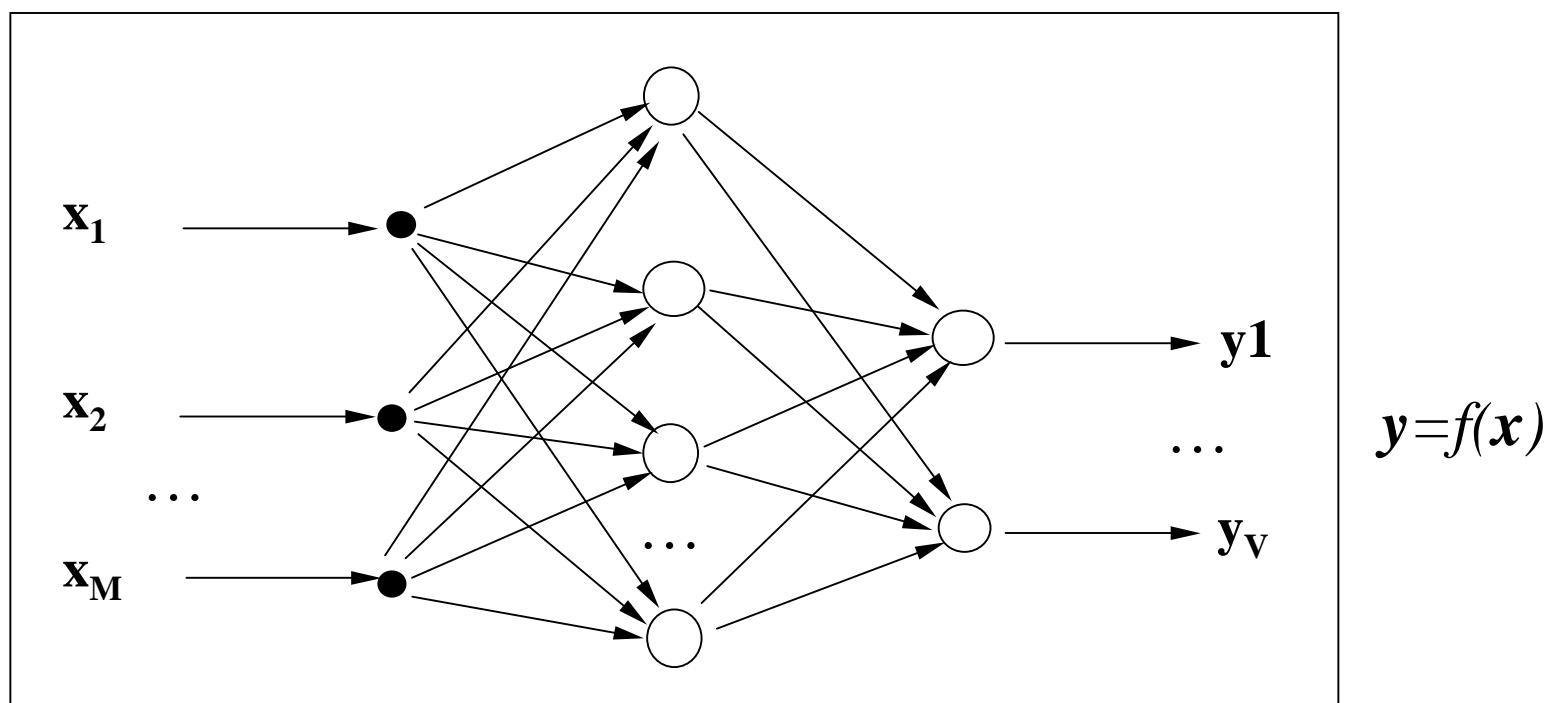
w_i - váhy synaptických spojení

a - vnútorná aktivita neurónu

y - výstup neurónu

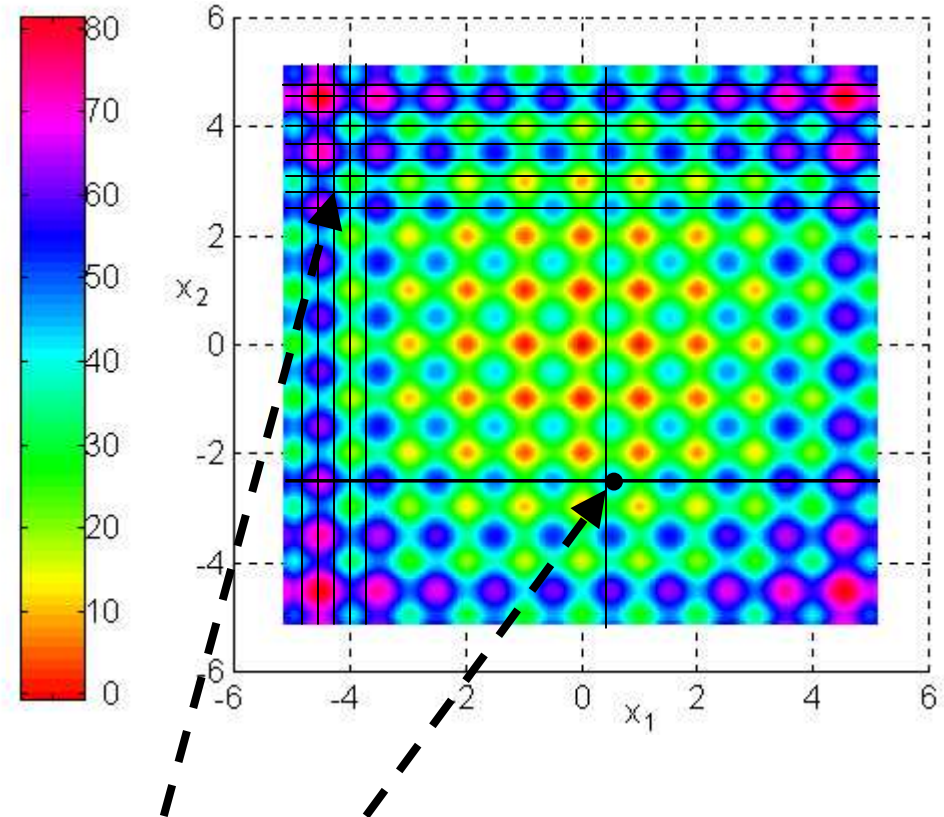
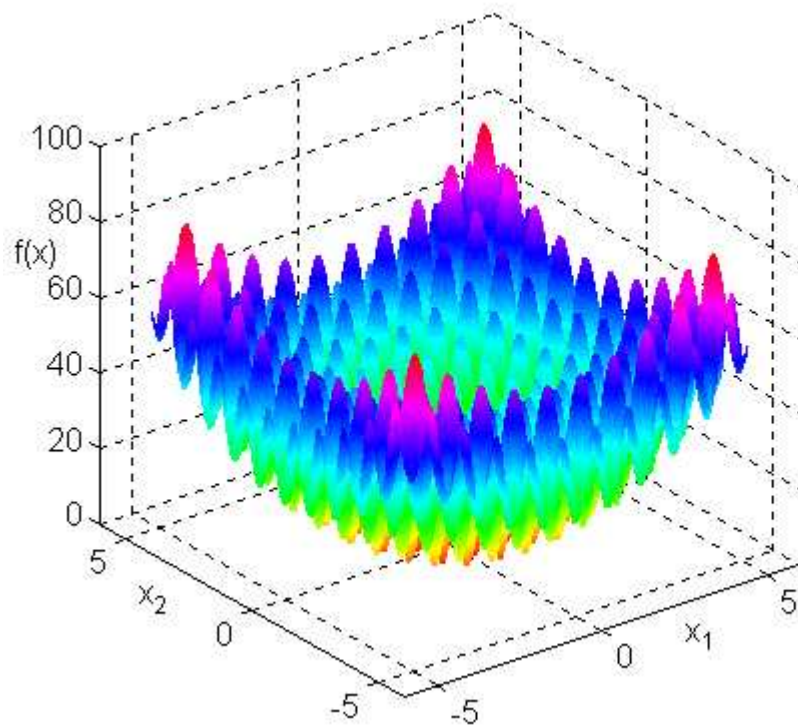
Viacvrstvové perceptrónové siete

- Obsahujú aspoň jednu skrytú vrstvu neurónov.
- Väčšinou obsahujú spojitú nelineárnu aktivačnú funkciu (obyčajne sigmoidu alebo hyperbolický tangens).
- Sú schopné aproximovať ľubovoľnú nelineárnu transformáciu.
- Parametrizácia takýchto sietí sa realizuje tréňovaním neurónovej siete algoritmom "spätného šírenia chyby".

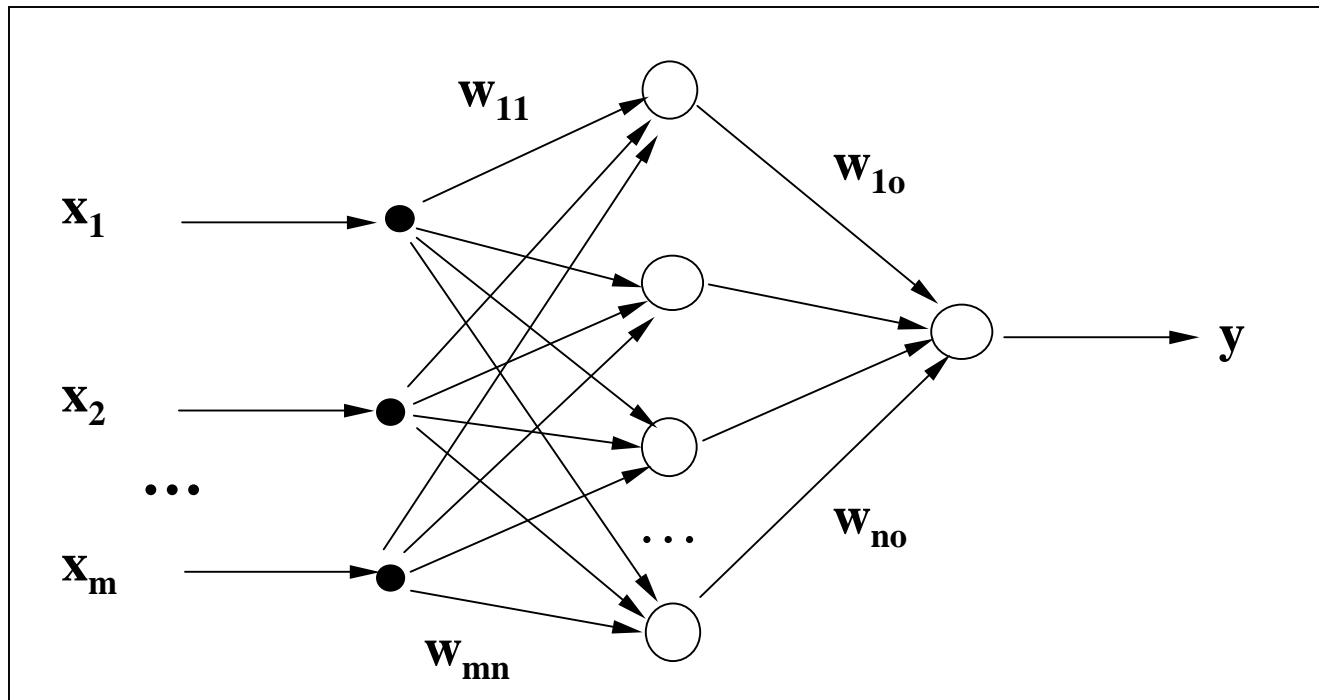


Trénovanie MLP :
Algoritmus spätného šírenia chyby
(„Back- Propagation“)

Aproximácia nelineárnej funkcie pomocou UNS



Definujeme sieť predloh (trénovacích vzorov), ktoré sú tvorené vektormi nezávisle premenných $x=[x_1, x_2, \dots]$ a im zodpovedajúcich hodôt $y=f(x)$



$w_{ij}=?$

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

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} ; $w_{ij}(t) = w_{ij}(t-1) + \Delta w_{ij}$

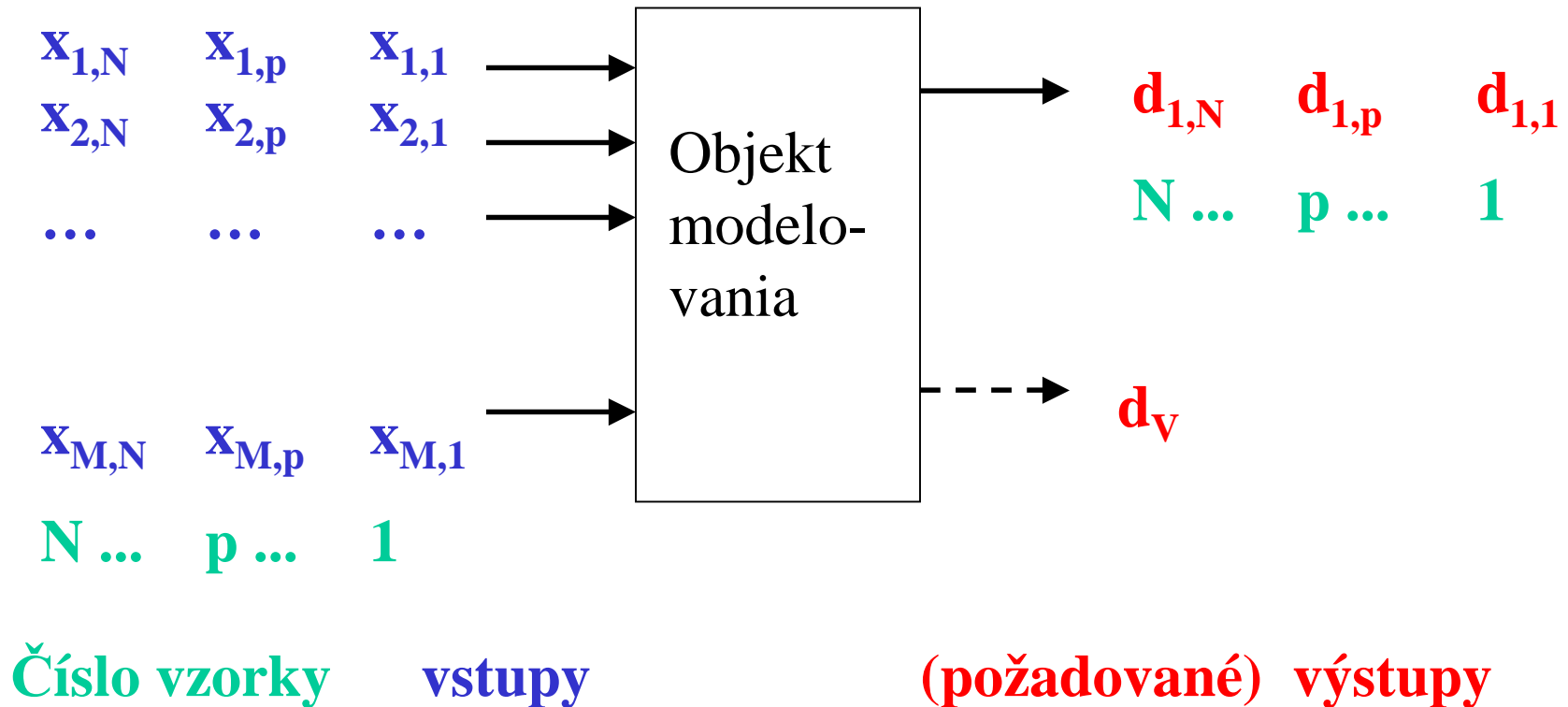
Predpokladajme trénovaciu množinu (vstupno/výstupných) dát

$$D = \{x_{pq}, d_{pr}\};$$

$p=1, \dots, N$ je počet vzoriek

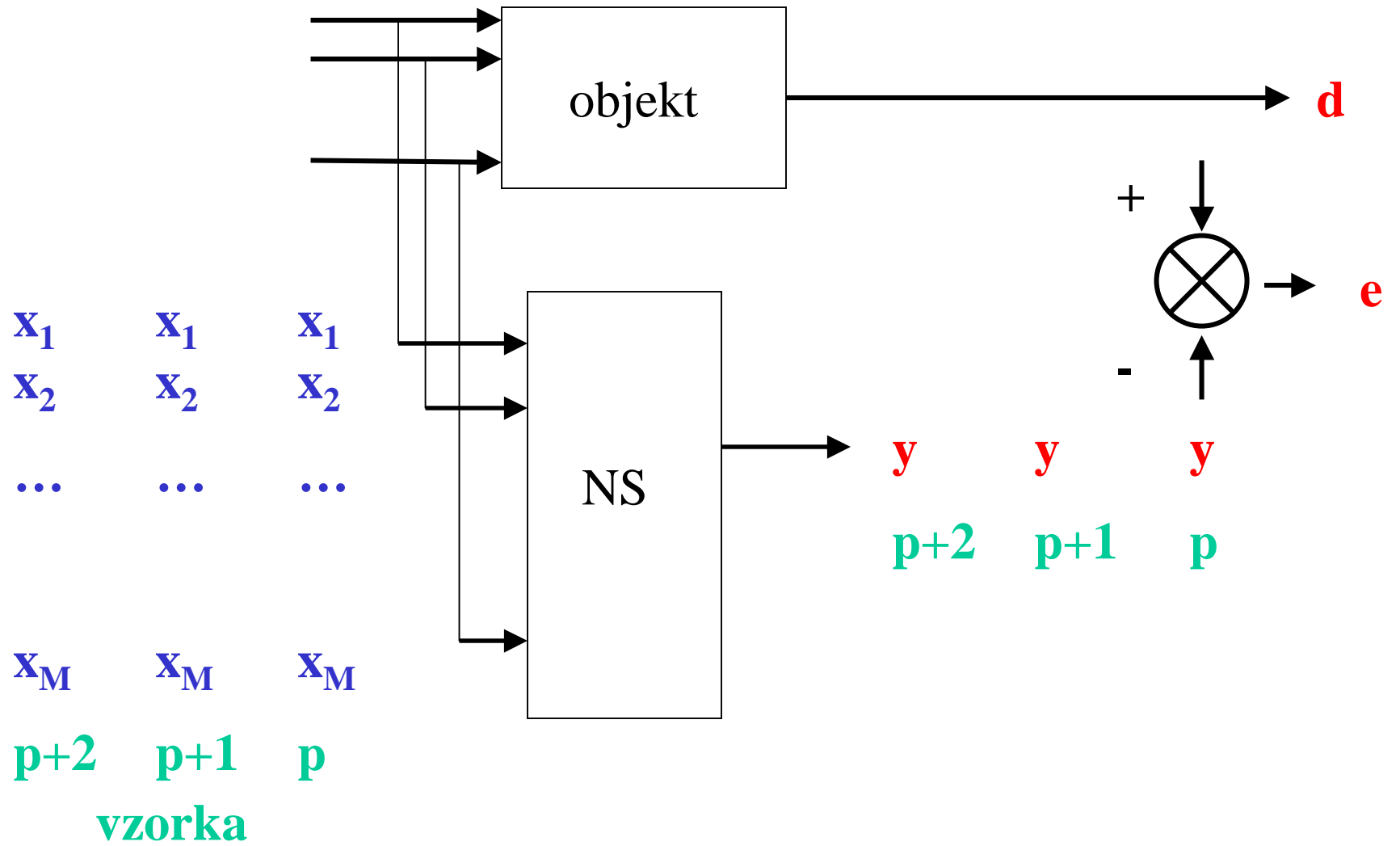
$q=1, \dots, M$ je počet vstupov siete

$r=1, \dots, V$ je počet výstupov siete

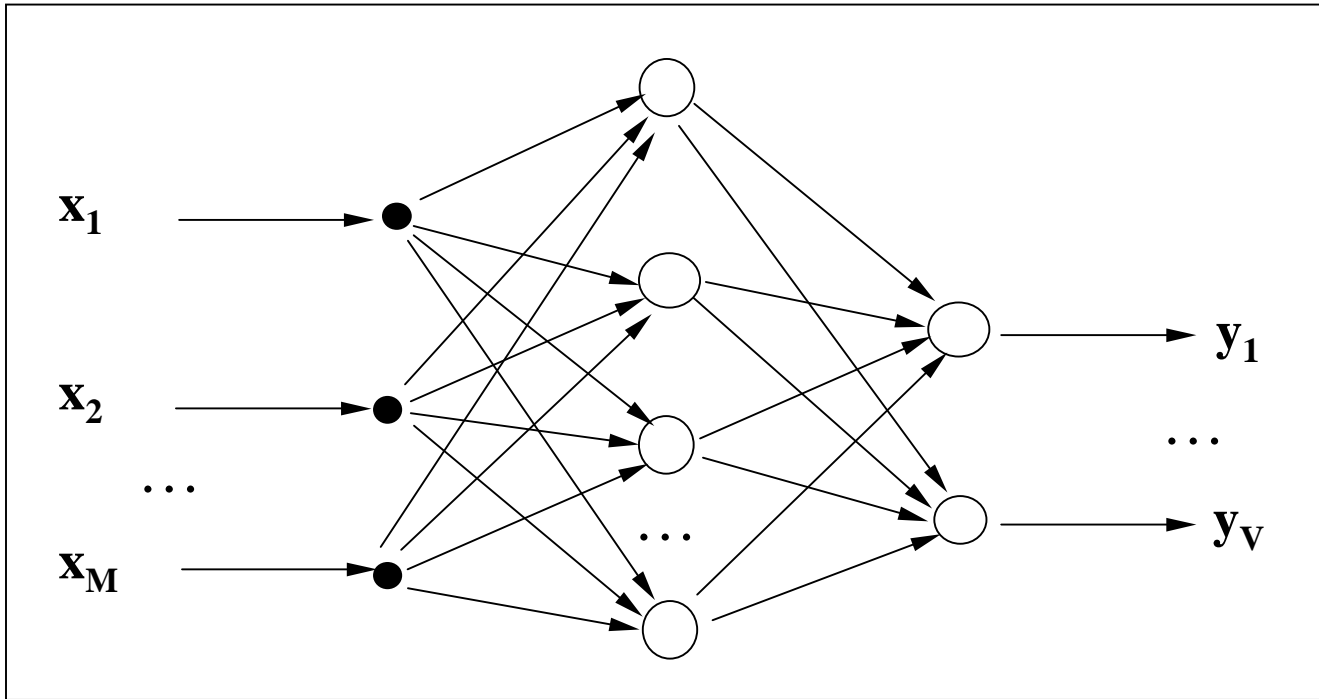


Trénovanie (parametrizácia) neurónovej siete prebieha vo viacerých cykloch („vlnách“) - tzv. EPOCHÁCH.

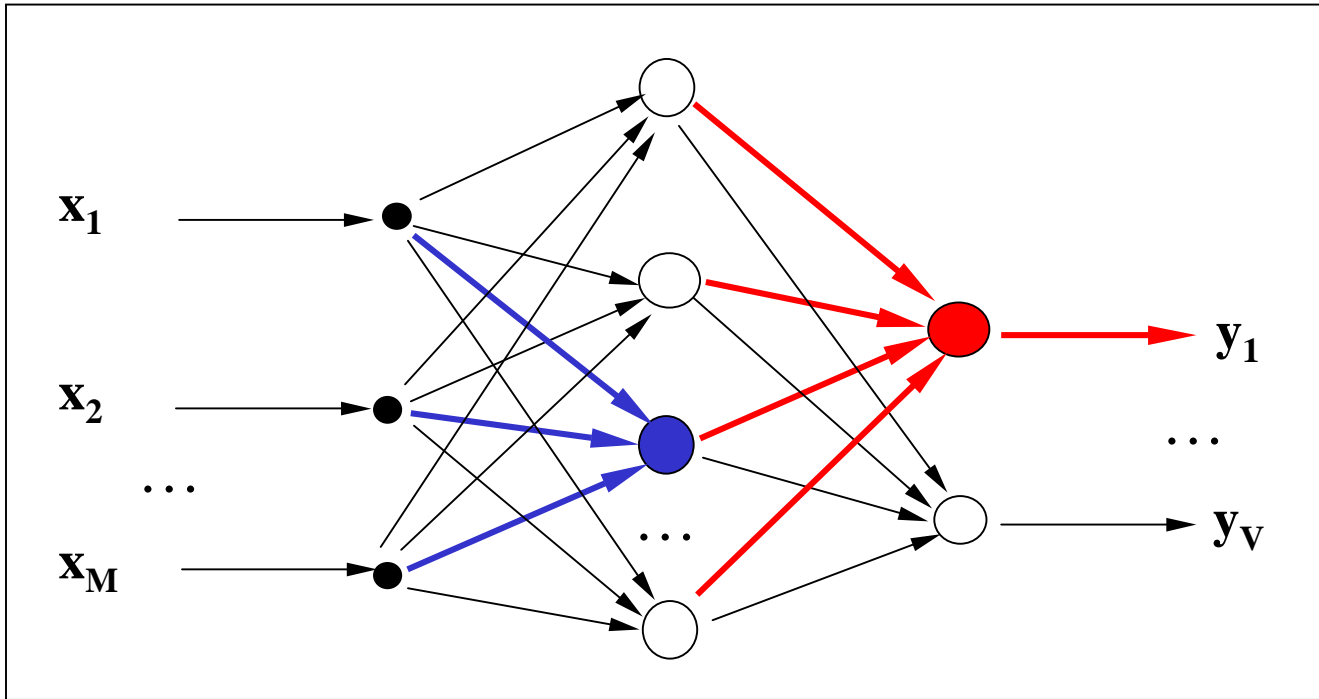
V každej epoche sa postupne cez sieť prešíri celá postupnosť trénovacích vzoriek x_p (postupnosť vstupných vektorov), porovnáva sa so skutočnými výstupmi a na základe odchýliek sa upravujú parametre siete).



1 epocha: $p=1,2,\dots,N$

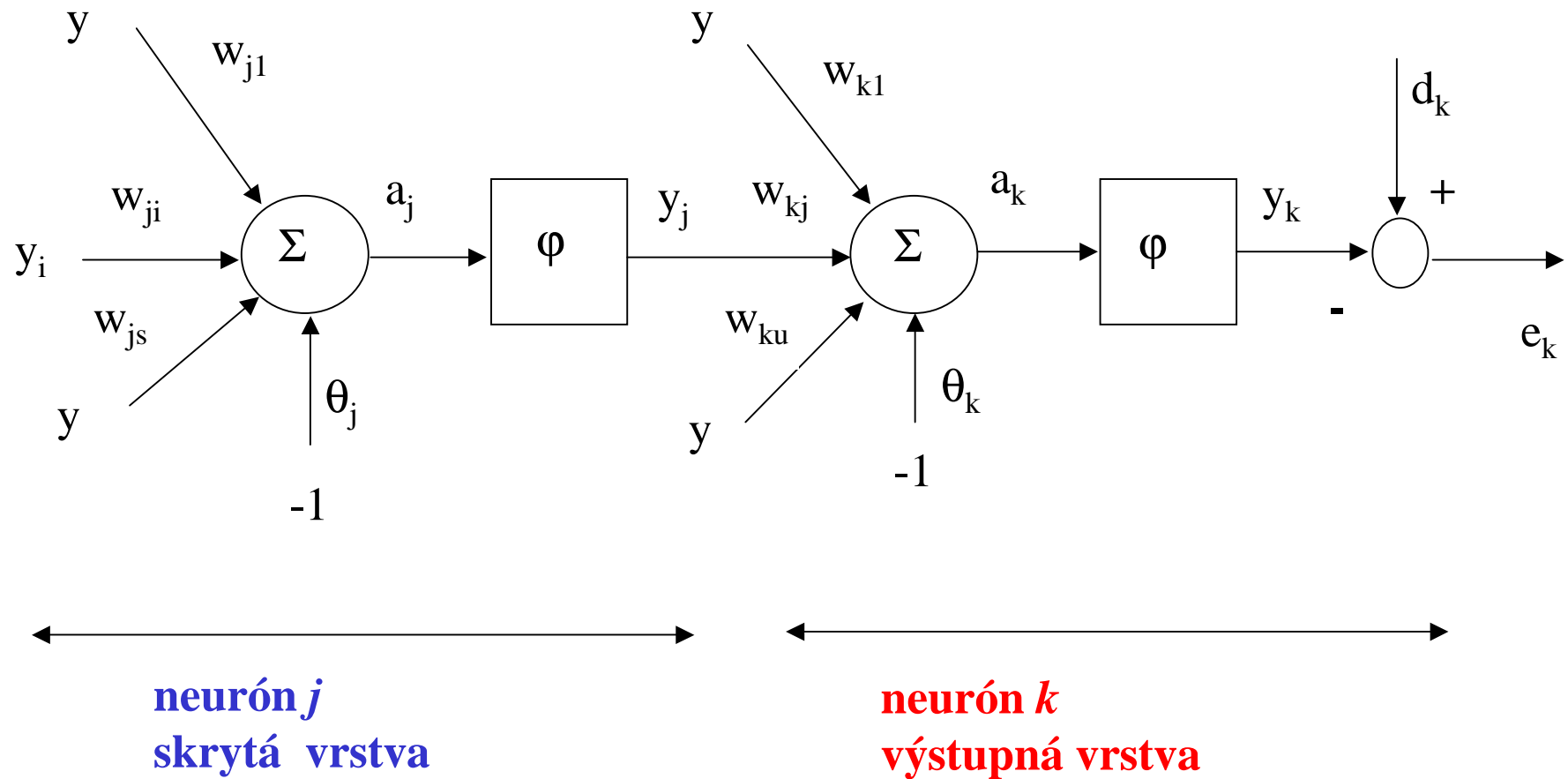


10-10



neurón j

neurón k



y_i - výstup neurónu vo vstupnej vrstve (je to vlastne jeden vstup siete)

y_j - výstup neurónu v skrytej vrstve

y_k - výstup neurónu vo výstupnej vrstve (je to vlastne jeden výstup siete)

Na vstupy neurónovej siete privedieme vektor tréningového súboru x

Na výstupe neurónu k bude potom hodnota y_k

Po jeho odčítaní od zodpovedajúcej vzorky tréningového súboru d_k získame chybu tohto výstupného neurónu

$$e_k = d_k - y_k$$

$$\varepsilon = \frac{1}{2} \sum_{k=1}^V e_k^2$$

- suma okamžitých kvadratických chýb na všetkých výstupoch (výstupných neurónoch)

Veľkosť ε závisí od hľadaných parametrov neurónovej siete (váh a prahov), predstavuje kritériálnu funkciu, ktorú minimalizujeme.

$$a_k = \sum_{j=1}^u w_{kj} y_j \quad - \text{vnútorná aktivita neurónu}$$

u – počet vstupov neurónu k

$$y_k = \varphi(a_k) \quad - \text{výstup neurónu}$$

Cieľ: hľadáme korekcie (prírastky) jednotlivých váh Δw_{kj} (resp. prahov), ktoré minimalizujú chybu siete

$\Delta w_{kj} \approx \partial \varepsilon / \partial w_{kj}$ - okamžitý gradient chyby
určuje smer minimalizácie kritériálnej funkcie

$$\frac{\partial \varepsilon}{\partial w_{kj}} = \frac{\partial \varepsilon}{\partial e_k} \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial a_k} \frac{\partial a_k}{\partial w_{kj}} \quad - \text{parciálna derivácia zloženej funkcie}$$

$$\frac{\partial \varepsilon}{\partial e_k} = e_k \quad \frac{\partial e_k}{\partial y_k} = -1 \quad \frac{\partial y_k}{\partial a_k} = \varphi'(a_k) \quad \frac{\partial a_k}{\partial w_{kj}} = y_j$$

dosadením dostaneme $\frac{\partial \varepsilon}{\partial w_{kj}} = -e_k \phi'(a_k) y_j$

definujme prírastok váh $\Delta w_{kj} = -\eta \frac{\partial \varepsilon}{\partial w_{kj}}$ η je parameter rýchlosti učenia

Znamienko mínus - gradientový zostup po ploche kriteriálnej f.

Poznámka: Hľadáme minimum hyper-plochy funkcie kvadratickej chyby gradientovou metódou, čiže hľadáme také parametre neurónovej siete, aby rozdiel medzi originálom a modelom bol čo najmenší.

$\Delta w_{kj} = \eta \delta_k y_j$ - korekcia váh výstupného neurónu

$\delta_k = -\frac{\partial \varepsilon}{\partial e_k} \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial a_k} = e_k \phi'(a_k)$ - "lokálny gradient" neurónu vo výstup. vrstve

Gradient využíva výstupnú chybu e_k výstupného neurónu. Pokiaľ sa jedná o neurón v skrytej vrstve, nevieme určiť jeho žiadanú odpoveď a preto ani jeho chybu. Chybový signál pre skrytý neurón musíme vypočítat' rekurzívne z chybových signálov neurónov, s ktorými je spojený.

Lokálny gradient pre neurón j v skrytej vrstve bude podobne

$$\delta_j = -\frac{\partial \varepsilon}{\partial y_j} \frac{\partial y_j}{\partial a_j} = -\frac{\partial \varepsilon}{\partial y_j} \varphi'(a_j)$$

Výpočet $\partial \varepsilon / \partial y_j$

$$\varepsilon = \frac{1}{2} \sum_{k=1}^V e_k^2$$

$$\frac{\partial \varepsilon}{\partial y_j} = \sum_{k=1}^V e_k \frac{\partial e_k}{\partial y_j} = \sum_{k=1}^V e_k \frac{\partial e_k}{\partial a_k} \frac{\partial a_k}{\partial y_j}$$

$$e_k = d_k - y_k = d_k - \varphi(a_k)$$

$$\frac{\partial e_k}{\partial a_k} = -\varphi'(a_k)$$

$$a_k = \sum_{j=0}^u w_{kj} y_j \quad - u \text{ je počet vstupov neurónu } k$$

$$\frac{\partial a_k}{\partial y_j} = w_{kj}$$

$$\frac{\partial \varepsilon}{\partial y_j} = -\sum_{k=1}^V e_k \varphi'(a_k) w_{kj} = -\sum_{k=1}^V \delta_k w_{kj}$$

Lokálny gradient pre skrytý neurón j bude

$$\delta_j = \varphi'(a_j) \sum_{k=1}^V \delta_k w_{kj}$$

- závisí od gradientov následne pripojených výstupných n. a teda od chybových signálov na výstupe n.s.

$$\Delta w_{ji} = \eta \delta_j y_i$$
 - úprava váh neurónov v skrytej vrstve

Poznámka: Výpočet korekcií prahov neurónov je analogický

Najčastejšou aktivačnou funkciou v n.s. býva sigmoidálna funkcia v tvare

$$y = \varphi(a) = \frac{1}{1 + e^{-a}} \quad \text{prípadne} \quad \varphi(a) = \frac{1}{1 + e^{-\beta a}}$$

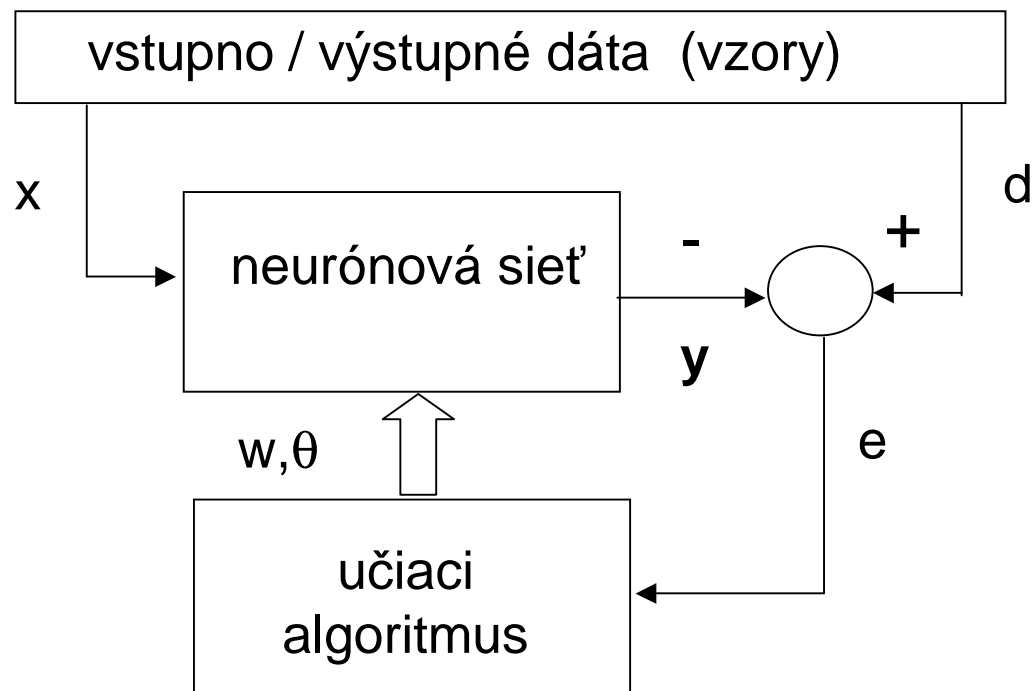
ktorá je spojitá, monotónna a diferencovateľná, čo sú podmienky každej aktivačnej f.

Jej derivácia je

$$\varphi'(a) = \frac{e^{-a}}{(1 + e^{-a})^2}$$

Derivácia φ' má maximum=0,25 v hodnote $a=0,5$ a minimum=0 pre $a=0$ alebo $a=1$. Keďže korekcia váhy je úmerná derivácii aktivačnej f., najviac sú upravované váhy tých neurónov, ktorých vnútorná aktivita sa práve pohybuje okolo stredného rozsahu sigmoidy. Toto prispieva k stabilite algoritmu spätného šírenia.

Algoritmus tréovania UNS



1. Inicializácia – váhy a prahy sa nastavujú na malé náhodné čísla
2. Postupne sa pre všetky vzorky (vektory) tréningového súboru dát vykoná nasledovný cyklus dopredného šírenia signálu a následne spätného šírenia poruchy – body 2a, 2b (bod 2 sa nazýva jedna „epocha“)
 - 2a. Na vstup siete sa privedie jeden vstupný vektor (jedna vzorka, krok) a vypočítajú sa výstupy siete a výstupná chyba pre každý výstup ako aj suma kvadrátov všetkých výstupných chýb
 - 2b. Späť sa počítajú všetky lokálne gradienty od výstupov ku vstupom a korigujú sa všetky váhy synaptických spojení a prahy neurónov
3. Pokiaľ sa nevykoná predpísaný počet tréningových epoch alebo globálna chyba nepoklesne pod určenú hranicu, pokračuje sa bodom 2, inak sa ukončí tréning.

Poznámky

Globálna chyba

$$E = \sum_{p=1}^N \mathcal{E} = \sum_{p=1}^N \frac{1}{2} \sum_{k=1}^V e_k^2$$

N - počet vzoriek trénovacej množiny, V - počet neurónov vo výstupnej vrstve

Vzorkový režim tréovania - váhy siete sa upravujú po privedení každej novej vzorky dát.

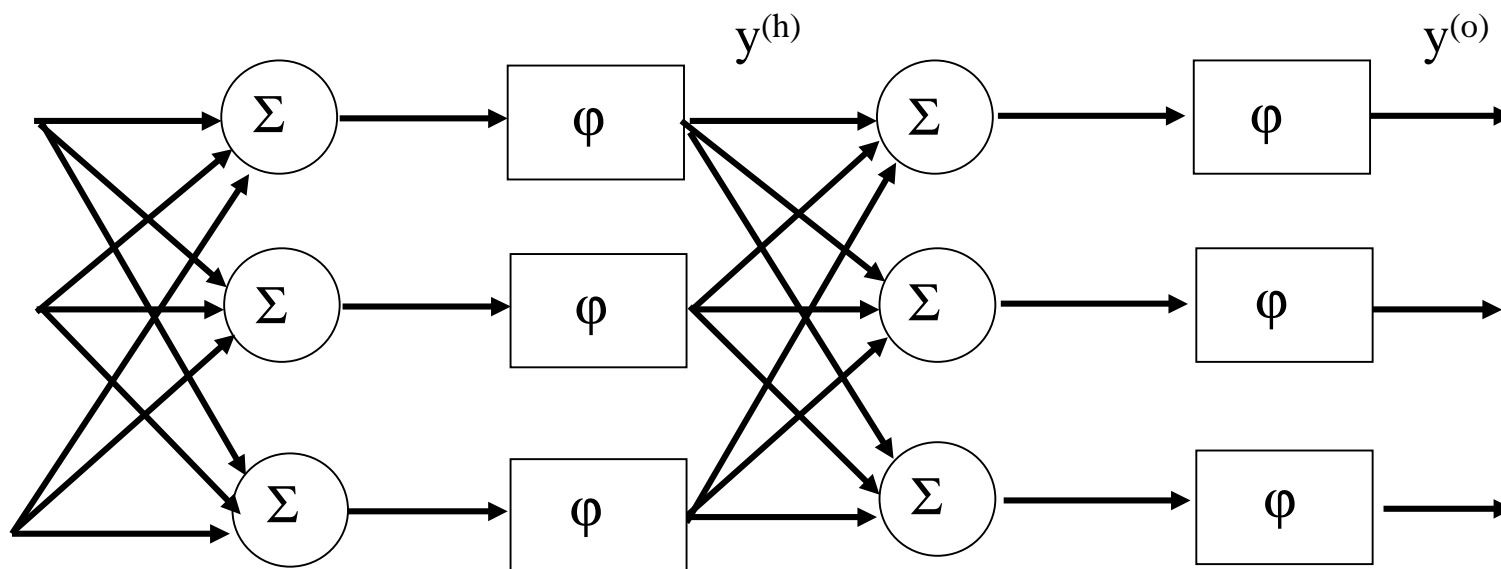
Dávkový režim tréovania - váhy sa korigujú až po celej epoche - po jednom kompletom prechode celého súboru tréovacích dát cez sieť, využitím globálnej chyby zo všetkých vzoriek a výstupov.

Ukončenie procesu tréovania

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

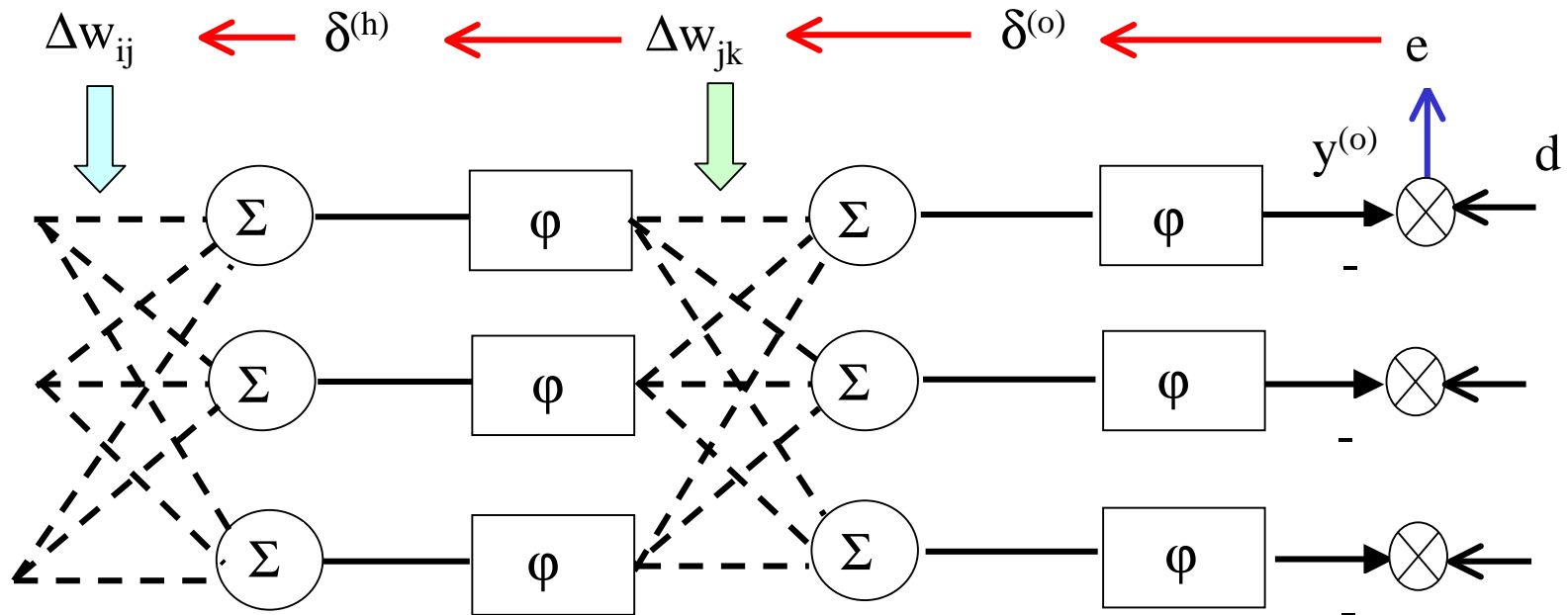
Proces tréovania je potrebné včas ukončiť, v opačnom prípade môže nastať stav "pretréovania", keď sa parametre n.s. už prestanú zlepšovať a začínajú na ne mať vplyv rôzne parazitné signály ako šum a pod., ktoré môžu následne zhoršiť vlastnosti siete (kopírovanie šumu ...).

Dopredná fáza šírenia signálov v neurónovej sieti



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

Spätná fáza šírenia signálov v neurónovej sieti



Príklad: Modelovanie funkcie n -premenných pomocou MLP

***Použitie viacvrstvovej perceptrónovej
siete na modelovanie nelineárnych
dynamických systémov***

Výstup modelu dynamického systému

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

alebo

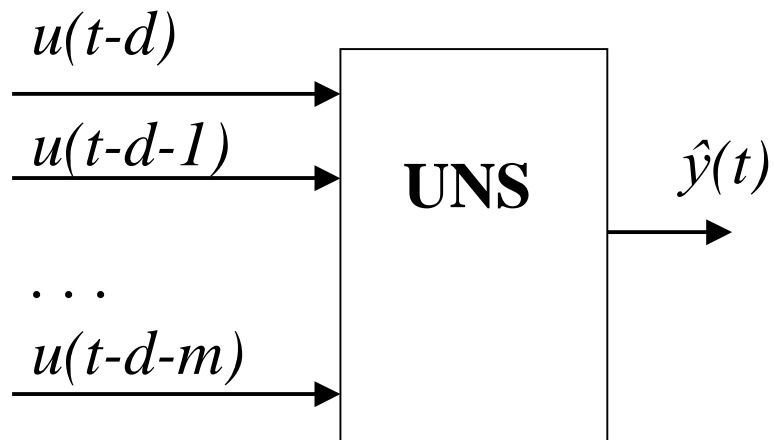
$$\hat{y}(t) = f (y(t-1), y(t-2), \dots, y(t-n), u(t-d), \dots, u(t-d-m), \dots)$$

$\hat{y}(t)$ – výstup modelu

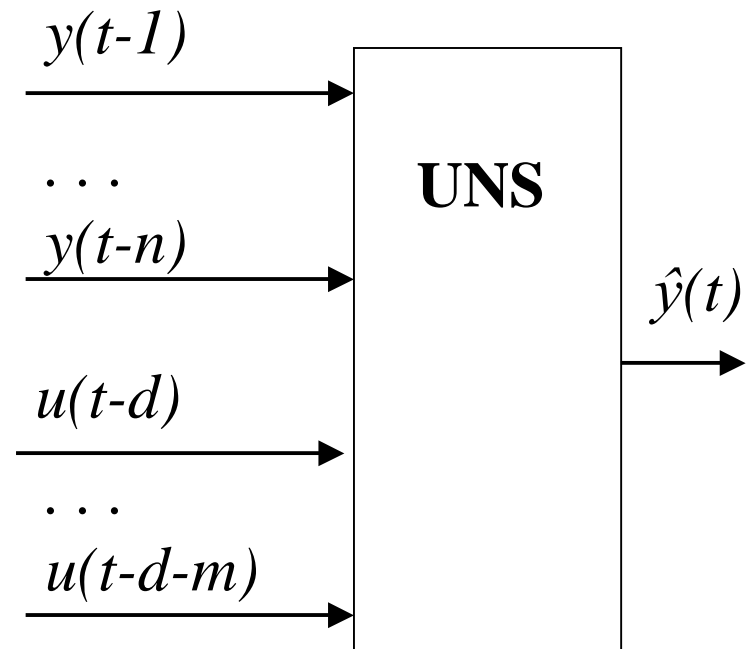
$f(\cdot)$ – nelineárna funkcia

$d \geq 1$

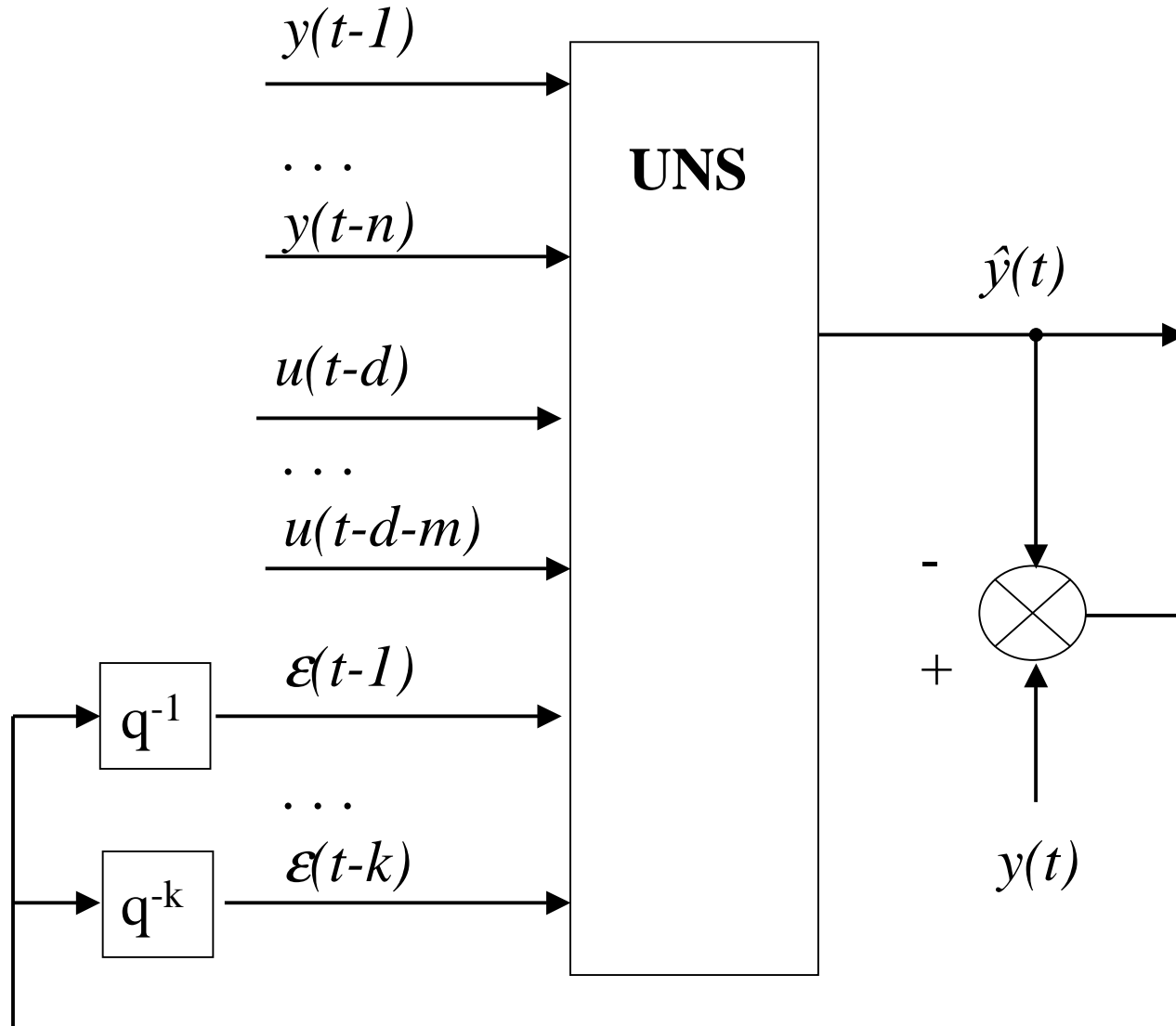
Typy dynamických modelov

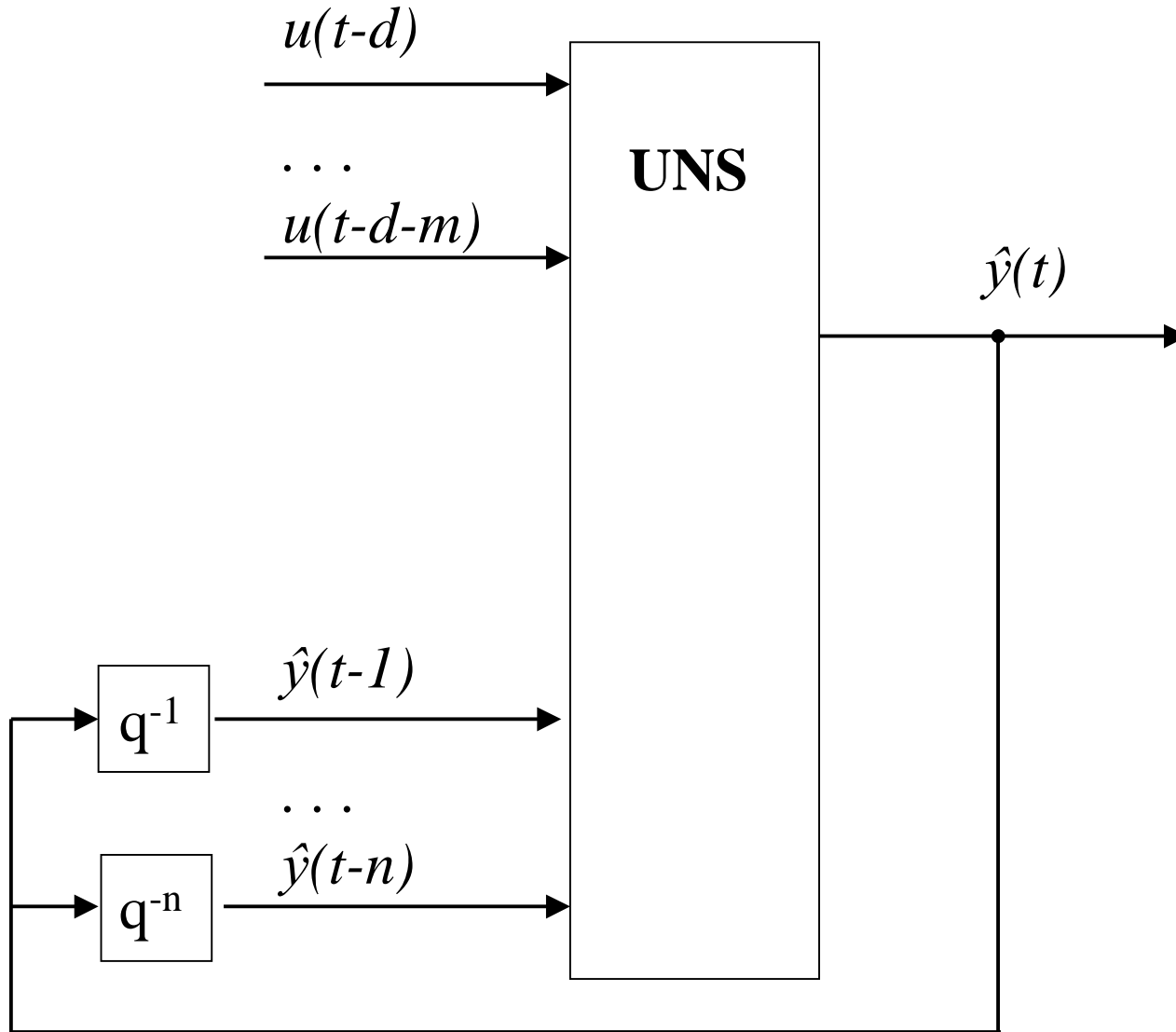


NNFIR



NNARX



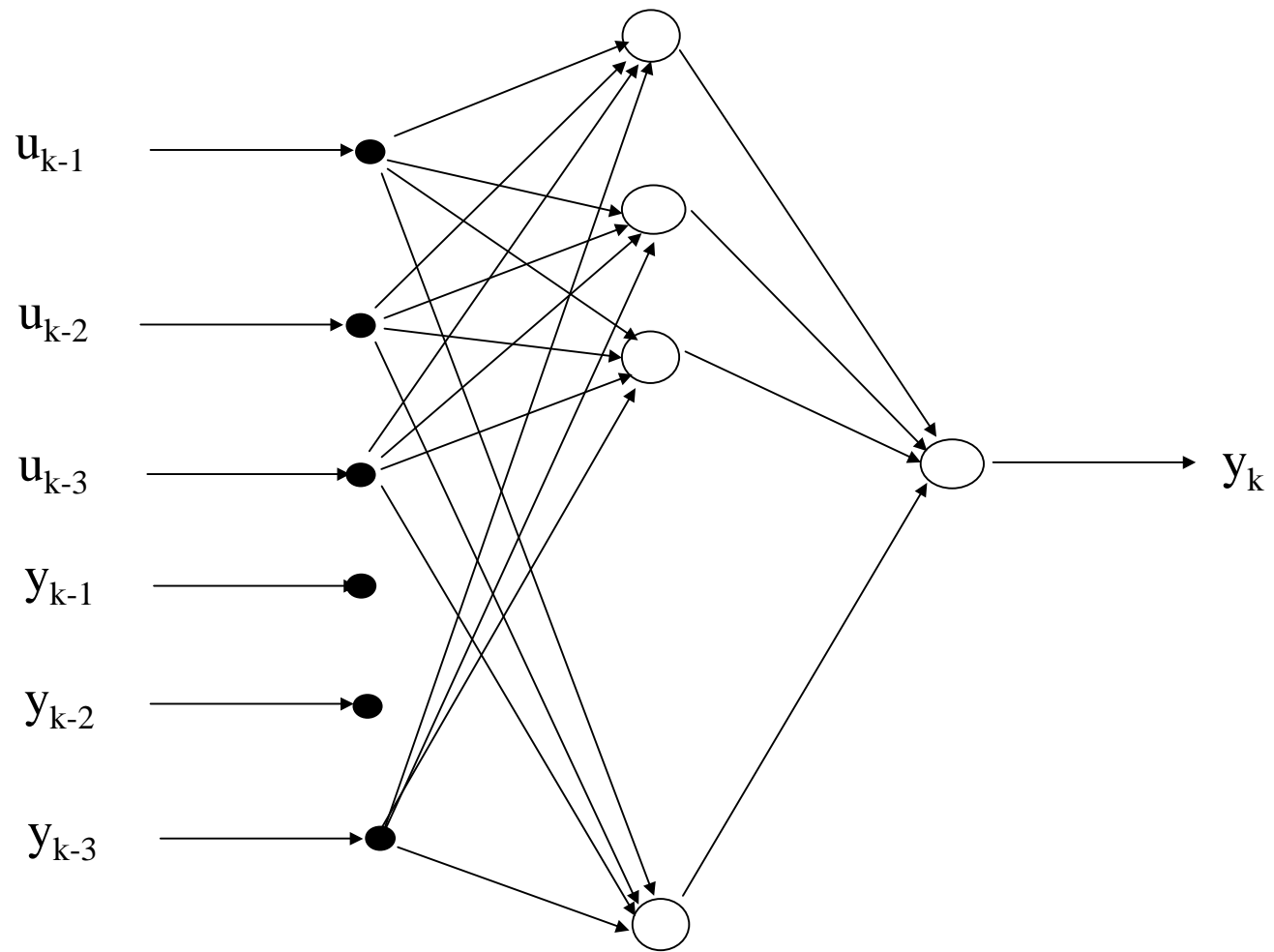


10-30

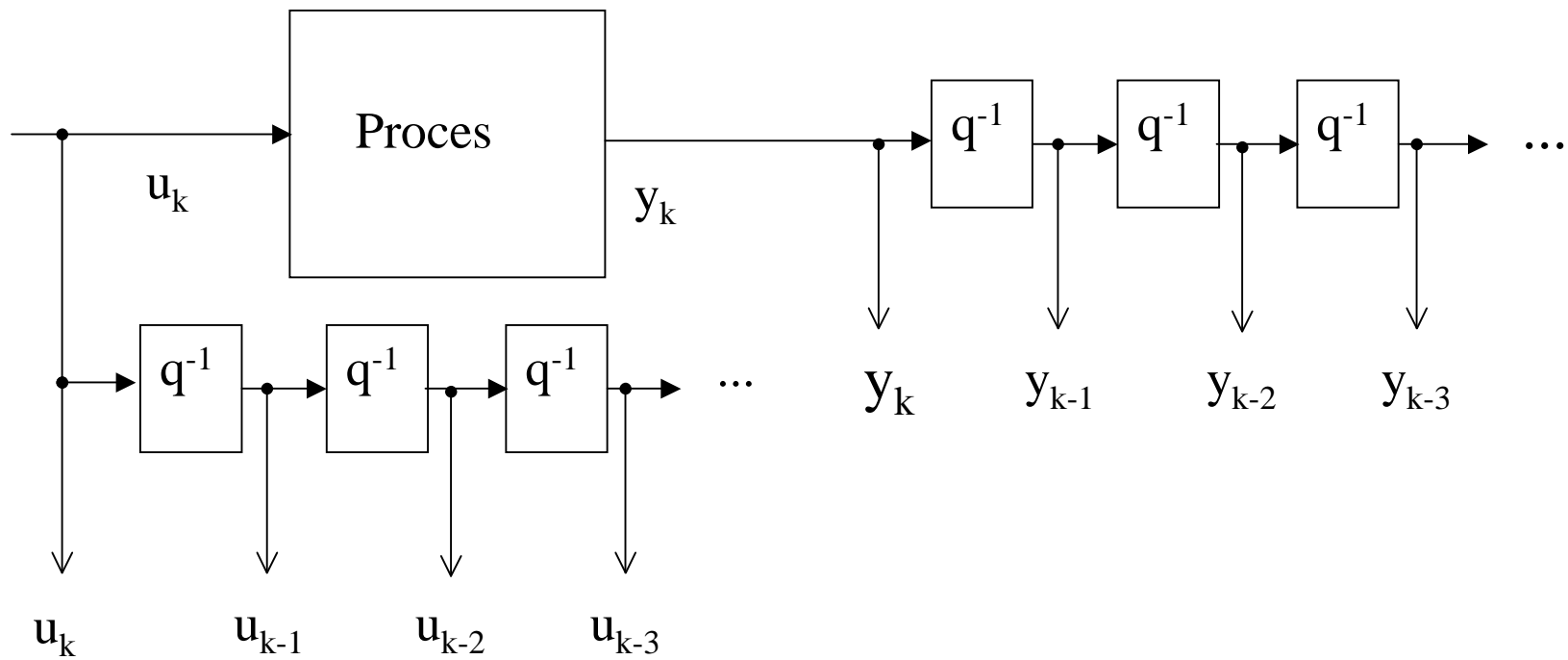
NNOE

Chyba modelu

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



10-32



Generovanie V/V signálov dyn. systému

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)

Použitie NNARX modelu

