Rozpoznávanie obrazcov šk.r. 2019-20

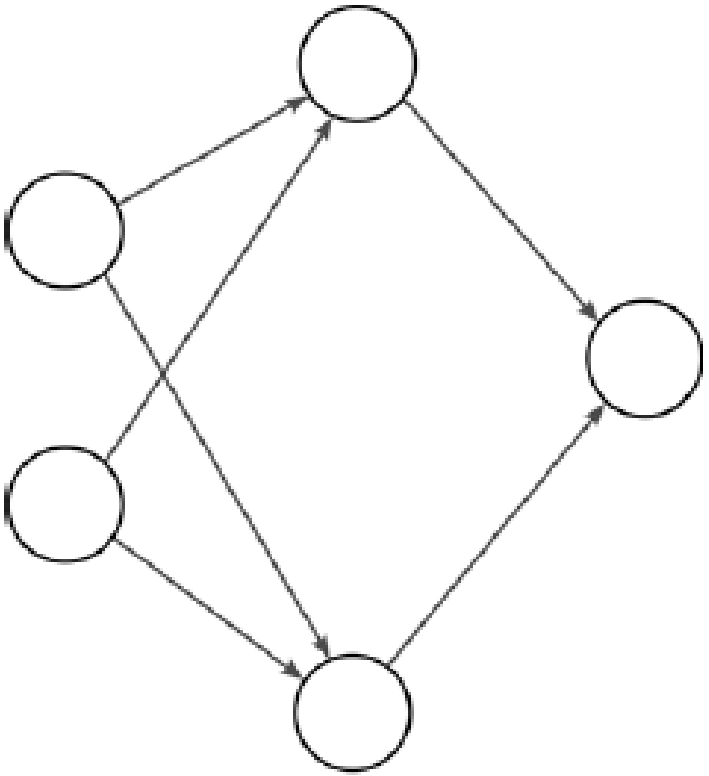
Umelé neurónové siete

Doc. RNDr. Milan Ftáčnik, CSc.

ANN

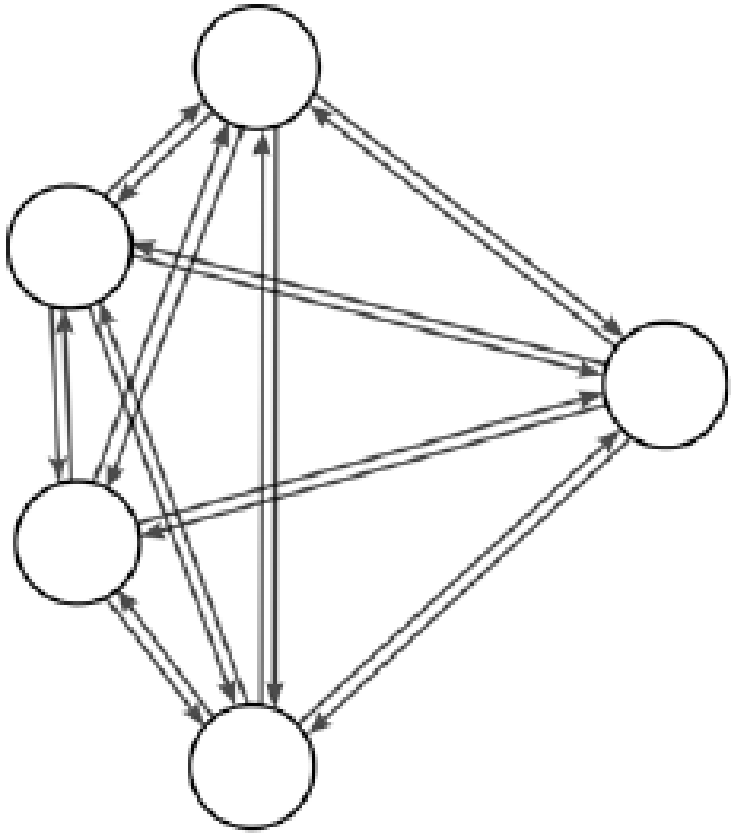
- Artificial Neural Networks:
  - Ide o siete pospájaných uzlov („neurónov“)
  - Informácia sa presúva medzi uzlami
  - Uzol vyšle informáciu na základe vstupu
- ANN môžu byť dopredné, spätné, samoorganizujúce a rôzne iné
- Sú výsledkom snahy napodobniť fungovanie ľudského mozgu

ANN II



Orientovaný acyklický graf

Dopredná sieť



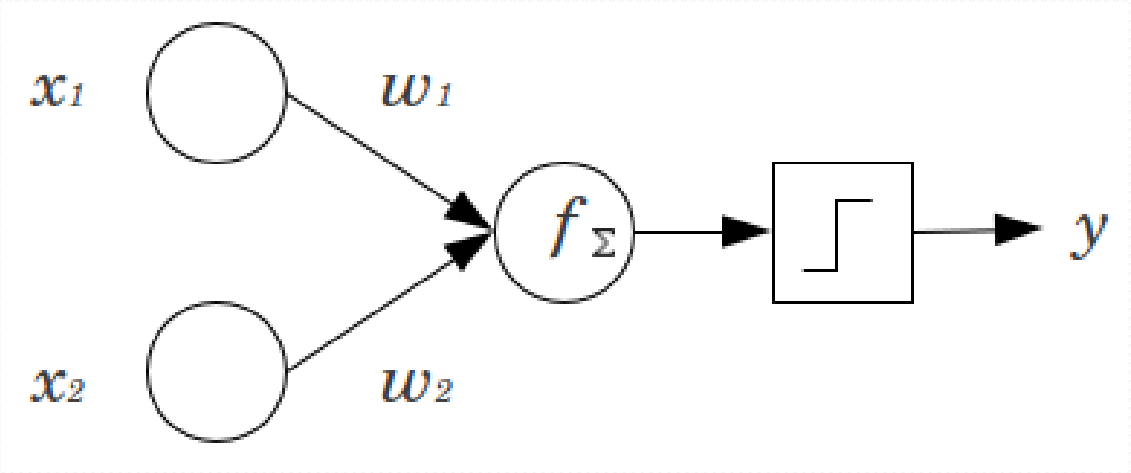
Spätná sieť

Postup práce ANN

- Výstupy z uzlov (neurónov) predošlej vrstvy = vstupy do uzla (neurónu) sa prenásobia váhami a spočítajú
- Na túto sumu sa v neuróne aplikuje nelineárna aktivačná funkcia ϕ (niekedy aj ako f)
- Výsledok funkcie ϕ je výstupom uzla
- (Postup sa opakuje v ďalšej vrstve)

Perceptrón

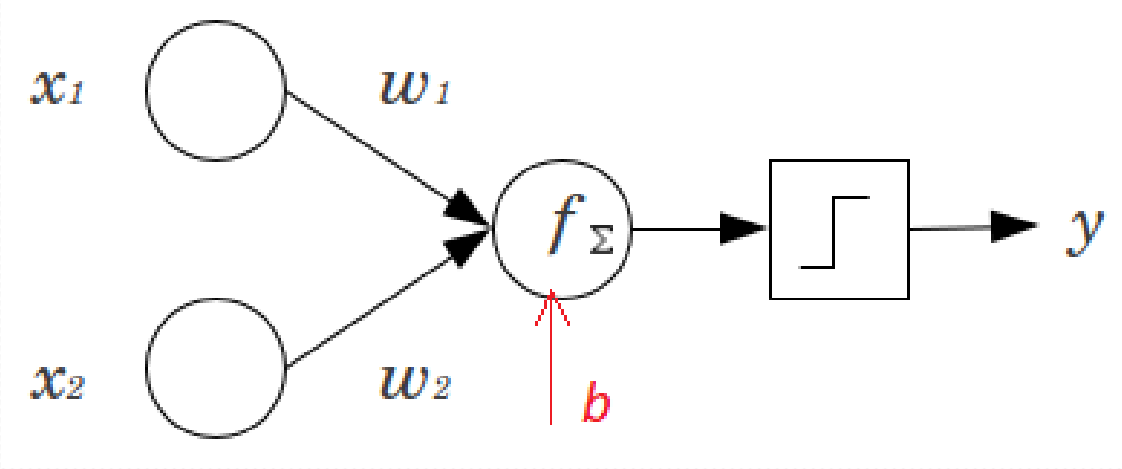
- Ako model klasifikátora do dvoch tried



- Ak by tam nebola nelineárna aktivačná funkcia tak sa sieť správa ako lineárny klasifikátor
- Môže sa doplniť prahová hodnota b

Perceptrón

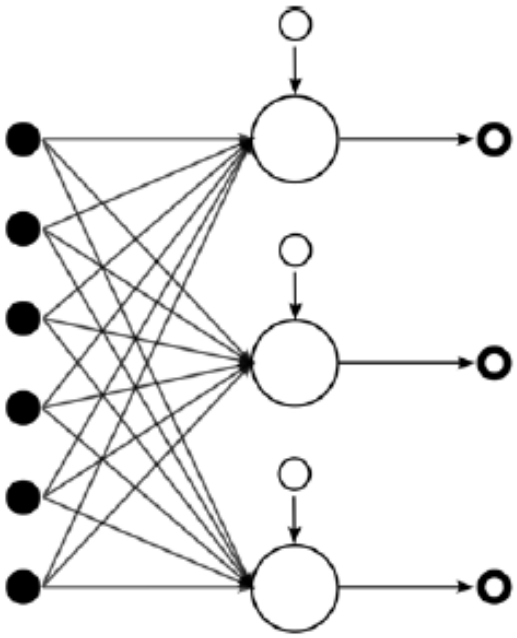
- Ako model klasifikátora do dvoch tried

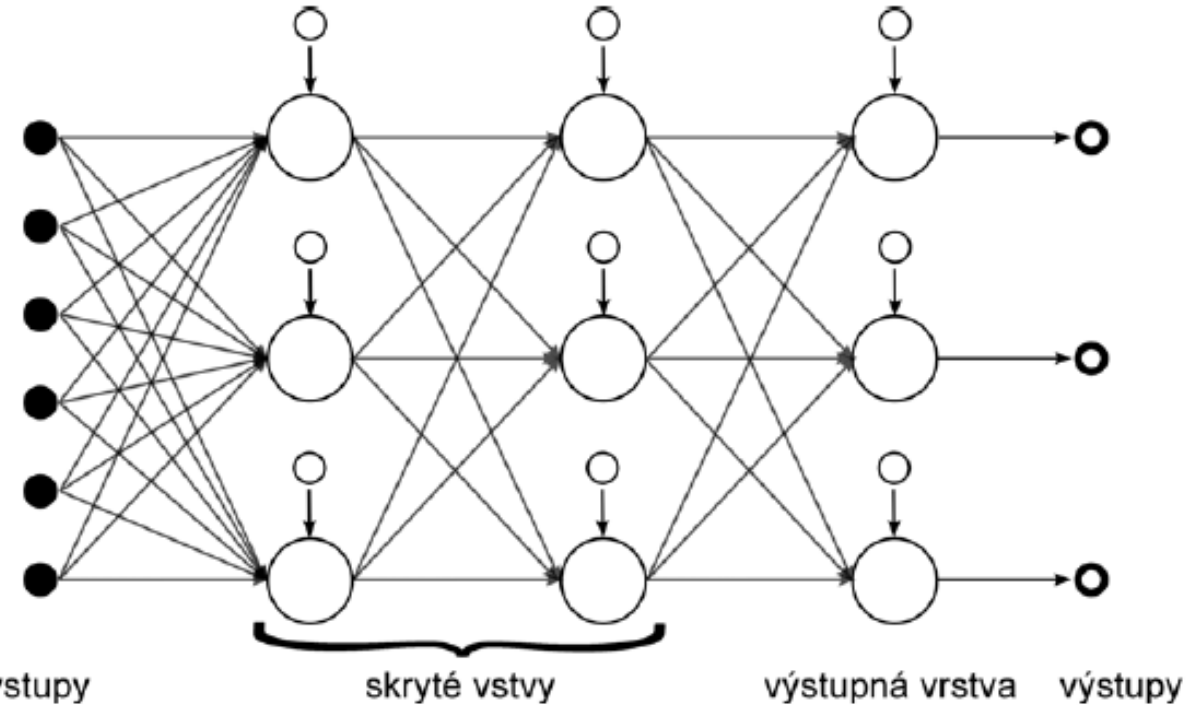


- Ak by tam nebola nelineárna aktivačná funkcia tak sa sieť správa ako lineárny klasifikátor
- Môže sa doplniť prahová hodnota b

Viacvrstvový perceptrón

- Klasifikácia do viacerých tried
- Má vstupnú a výstupnú vrstvu a skryté vrstvy

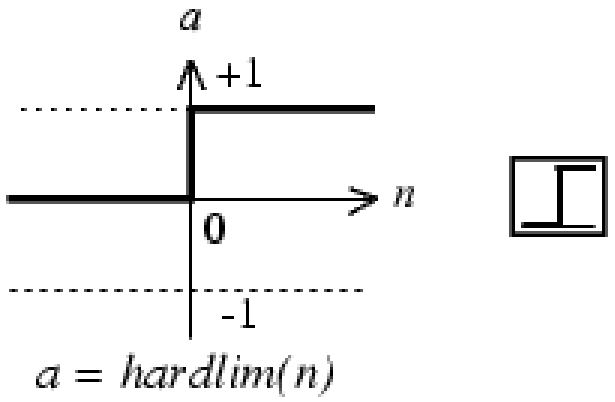




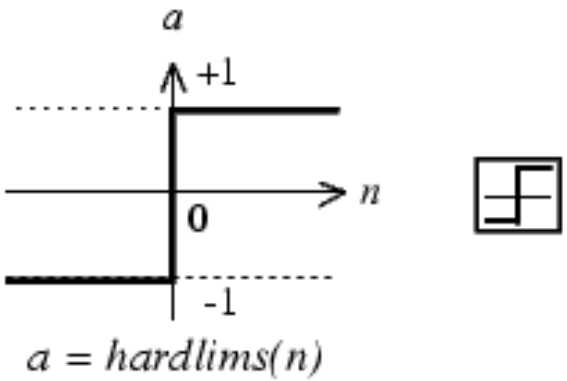
Aktivačná funkcia

- Prahová

$$\phi(x) = \begin{cases} 1 & \text{ak } x \geq 0, \\ 0 & \text{inak.} \end{cases}$$



Hard-Limit Transfer Function



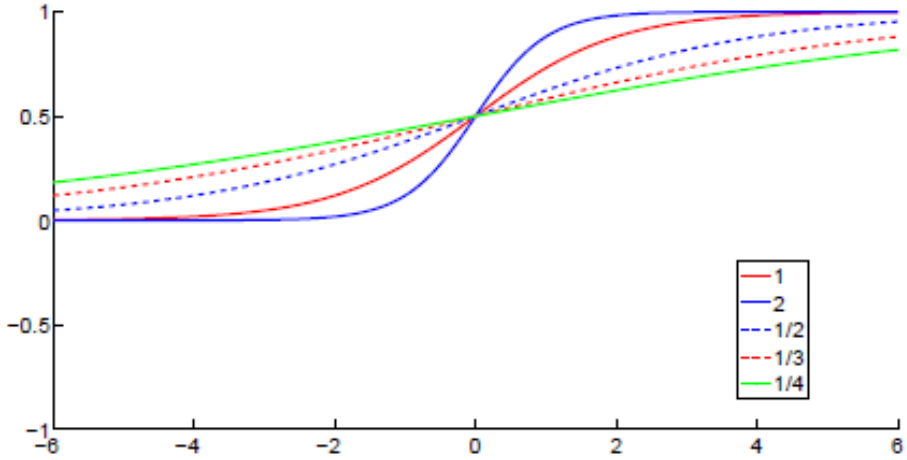
Symmetric Hard-Limit Transfer Function

Aktivačná funkcia II

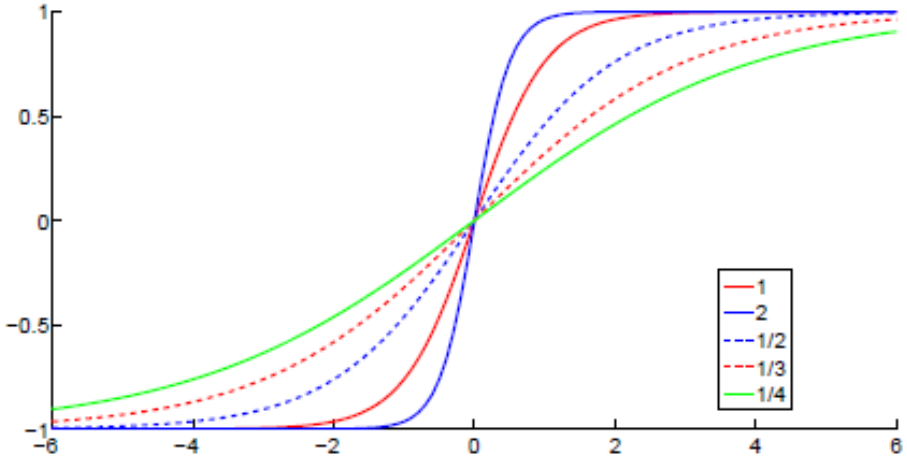
Logistická funkcia
(pre $\alpha = 1$ nazývaná sigmoida) Hyperbolický tangens

$$\phi(\alpha, x) = \frac{1}{1 + e^{-\alpha x}},$$

$$\phi(\beta, x) = \tanh(\beta x) = \frac{e^{\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}}.$$



(a) Logistická funkcia pre rôzne hodnoty parametra α.



(b) Hyperbolický tangens pre rôzne hodnoty parametra β.

Aktivačná funkcia III

- Obe tieto funkcie majú graf pripomínajúci tvar *S*, preto ich budeme nazývať *S*-funkcie
- Pri rozpoznávaní sa v skrytých vrstvách používajú prahová funkcia alebo *S*-funkcie a vo výstupnej vrstve lineárna aktivačná funkcia
- Nelineárna funkcia v skrytých vrstvách umožňuje sieti zachytiť nelineárne vzťahy medzi vstupom a výstupom siete

Učenie neurónovej siete

- Ide o určenie váhovej matice $\boldsymbol{W}$ (váhových vektorov v jednotlivých vrstvách)
- Hľadáme hodnoty také hodnoty $\boldsymbol{W}$, ktoré minimalizujú zvolenú cenovú funkciu chyby E
- Na nájdenie minima použijeme gradientnú metódu – ukážeme bližšie doprednú sieť (feed-forward) so spätnou propagáciou chyby (back propagation)

Učenie neurónovej siete II

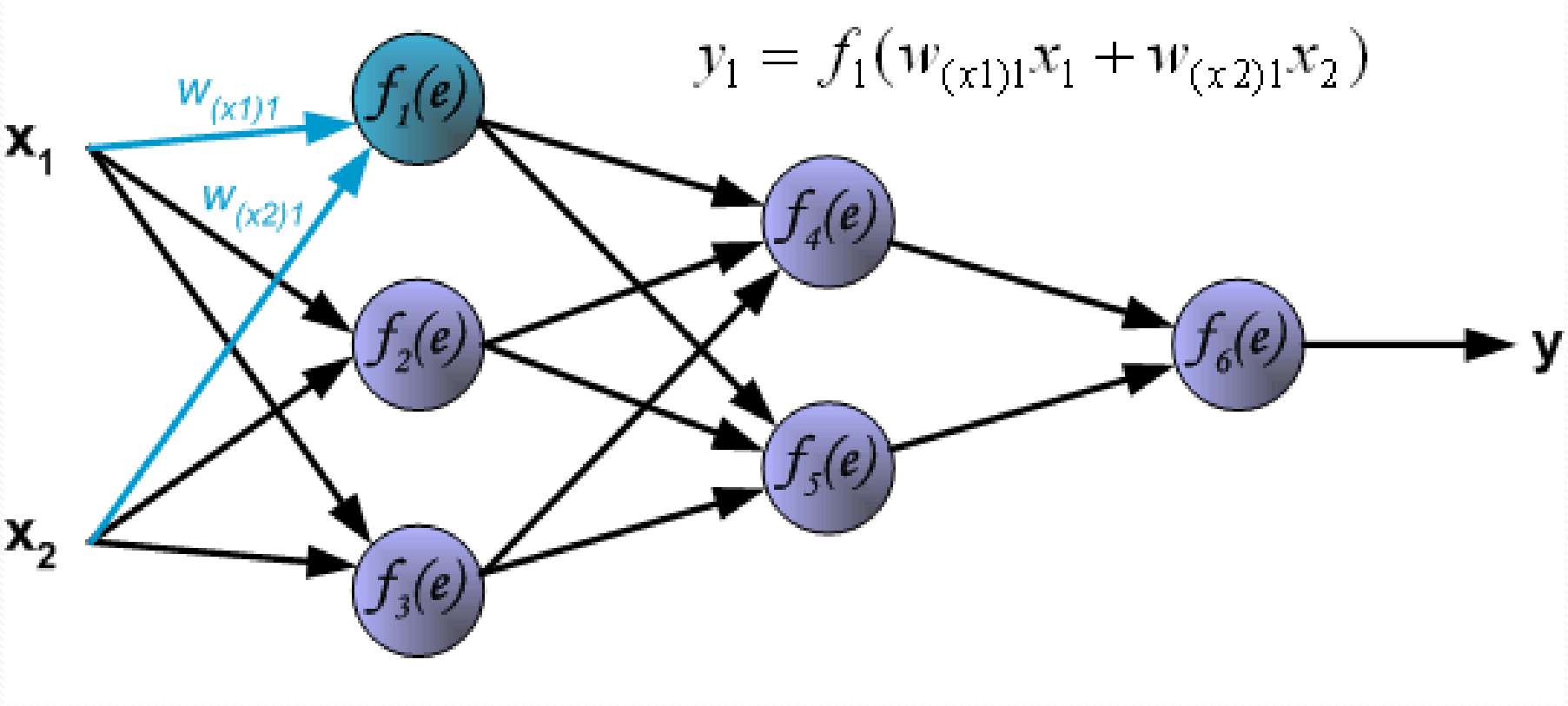
1. Inicializácia náhodnej váhovej matice

2. Vektory príznakov z trénovacej množiny postupne prichádzajú na vstup a cez skryté vrstvy siete prejdú až na koniec

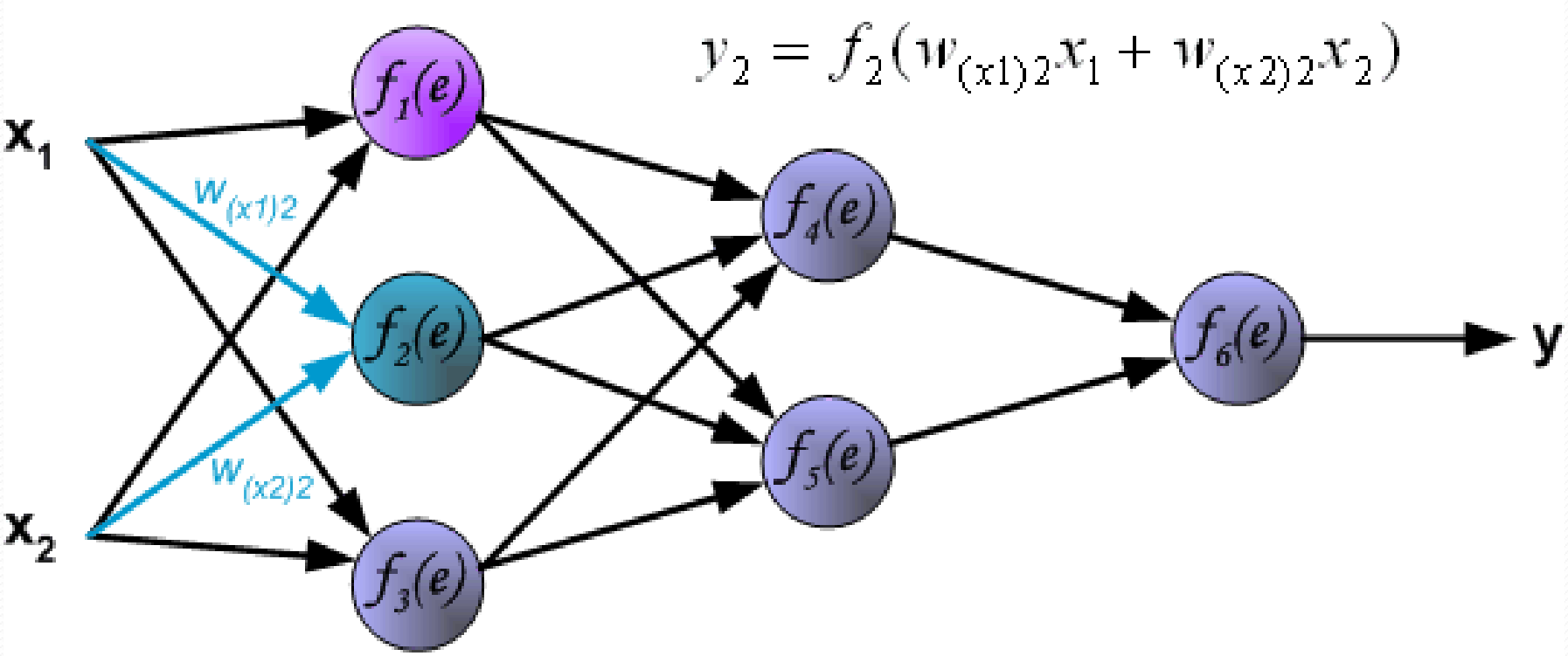
3. Výstupný vektor sa porovná so známou klasifikáciou a určí sa hodnota chybovej funkcie E

4. Chyba sa spätne šíri a upravujú sa hodnoty váhovej matice $\mathbf{W} \leftarrow \mathbf{W} + \Delta\mathbf{W}, \quad \Delta\mathbf{W} = -\eta \nabla E|_{\mathbf{W}}$

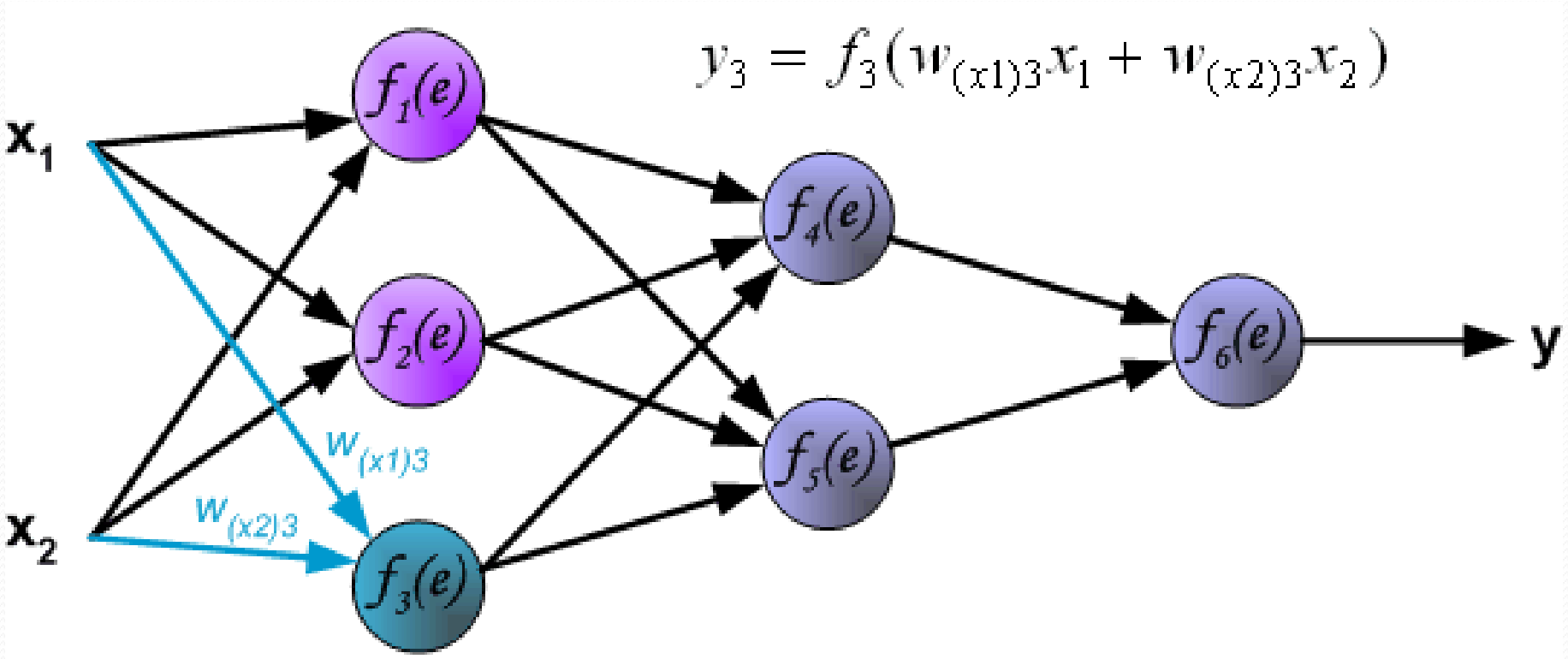
Dopredná sieť



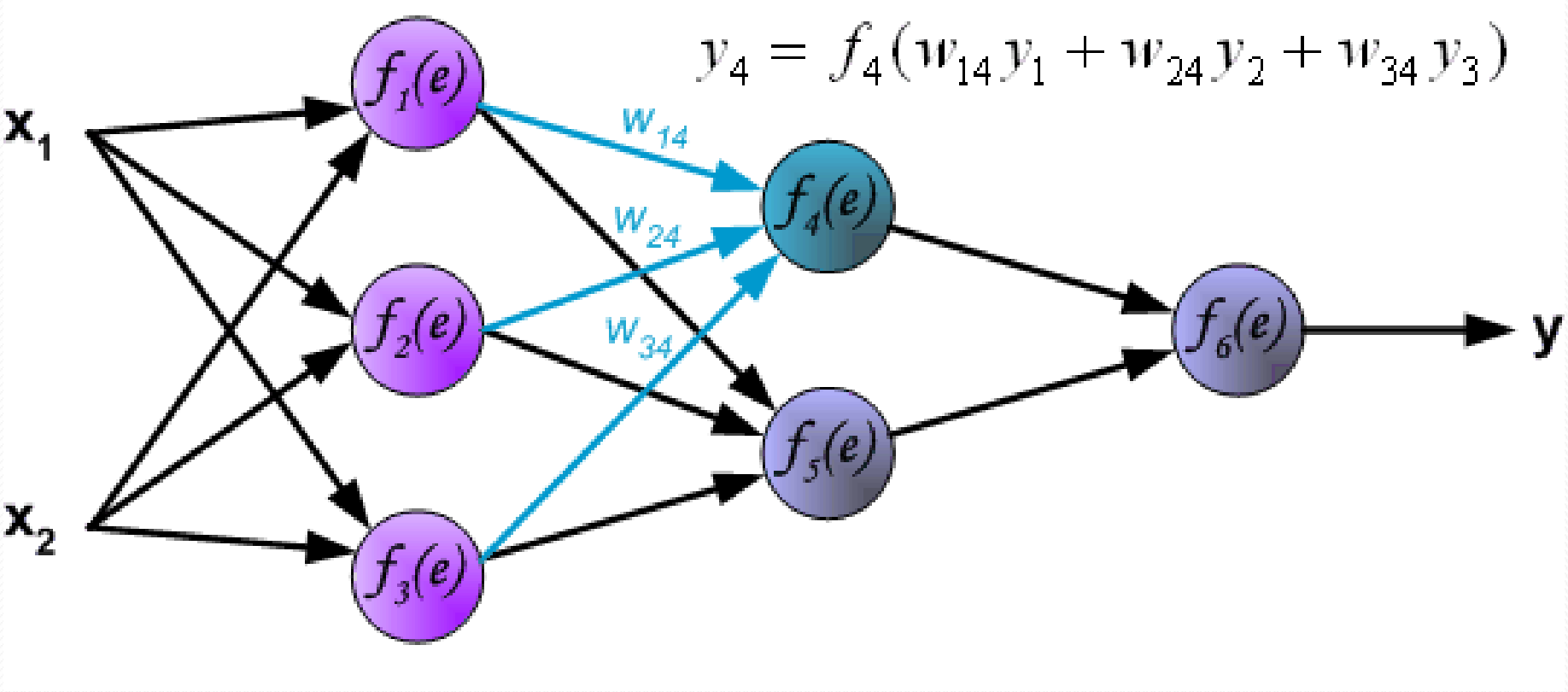
Dopredná sieť II



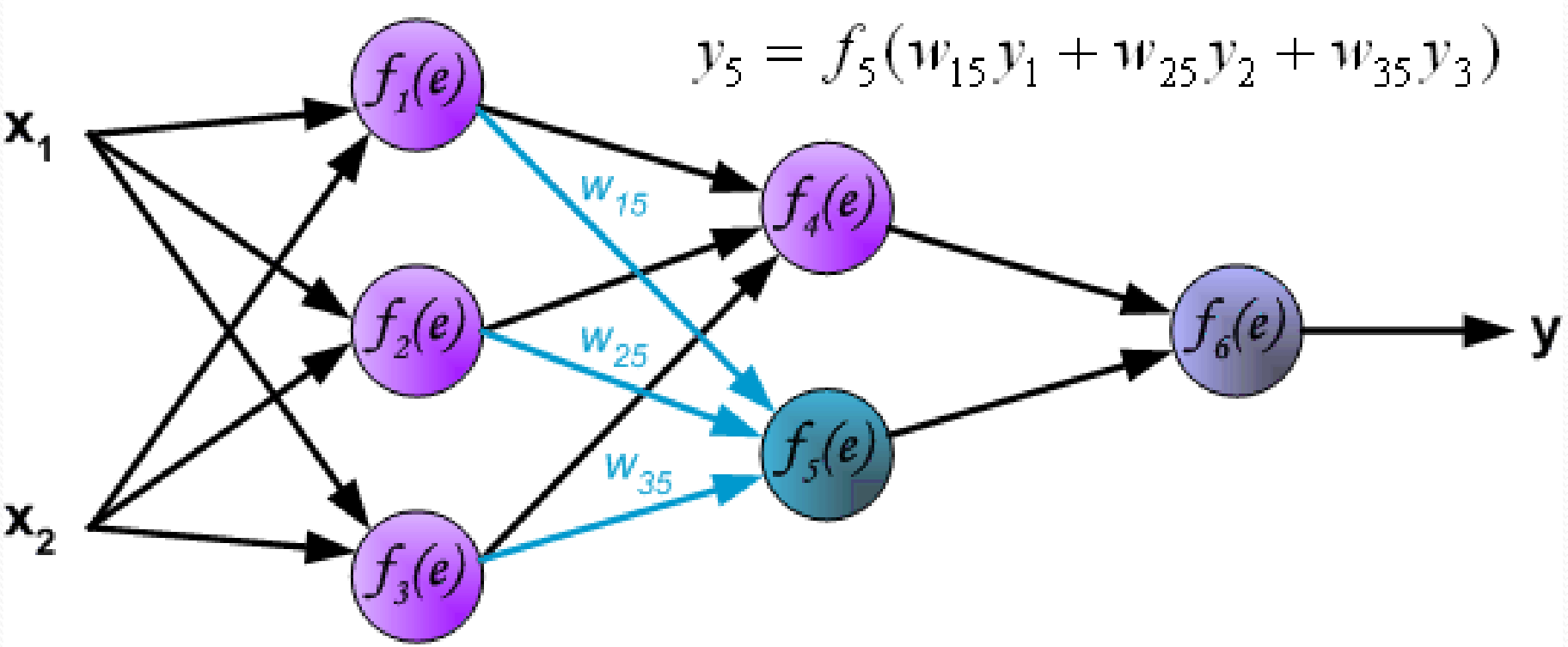
Dopredná sieť III



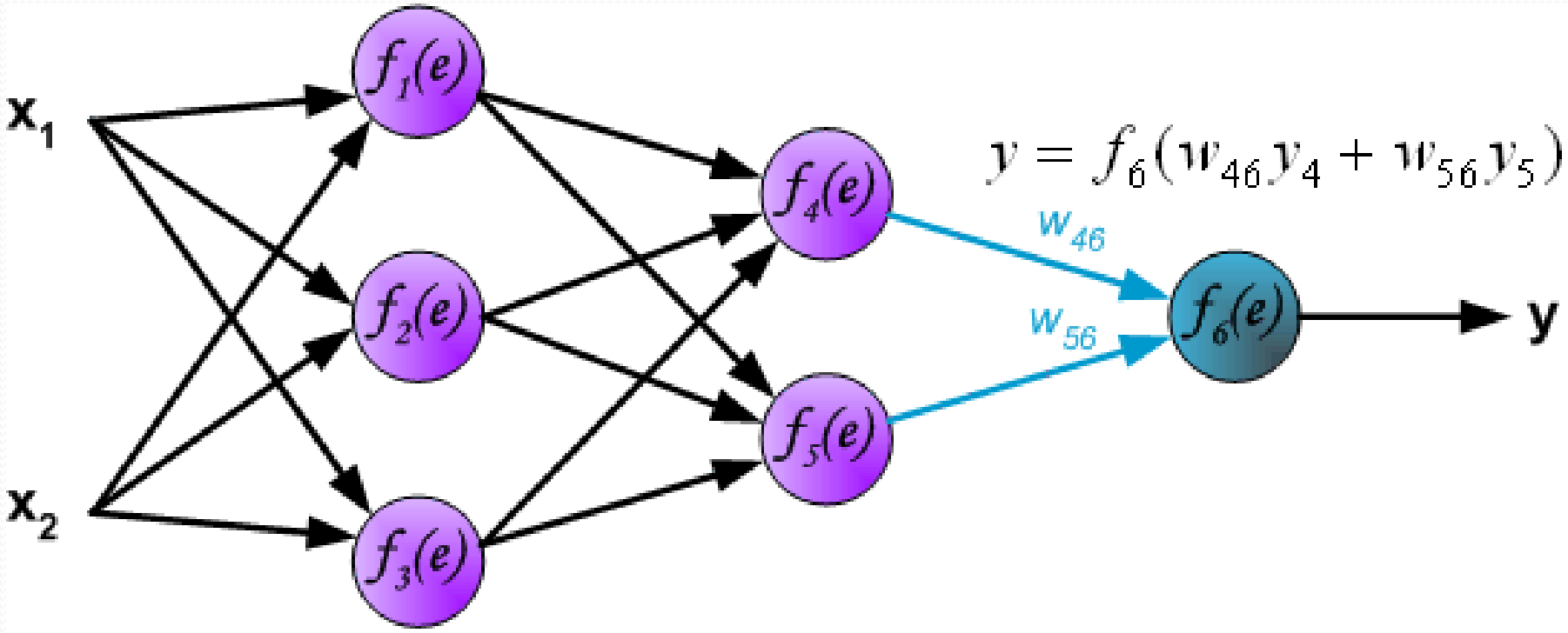
Dopredná sieť IV



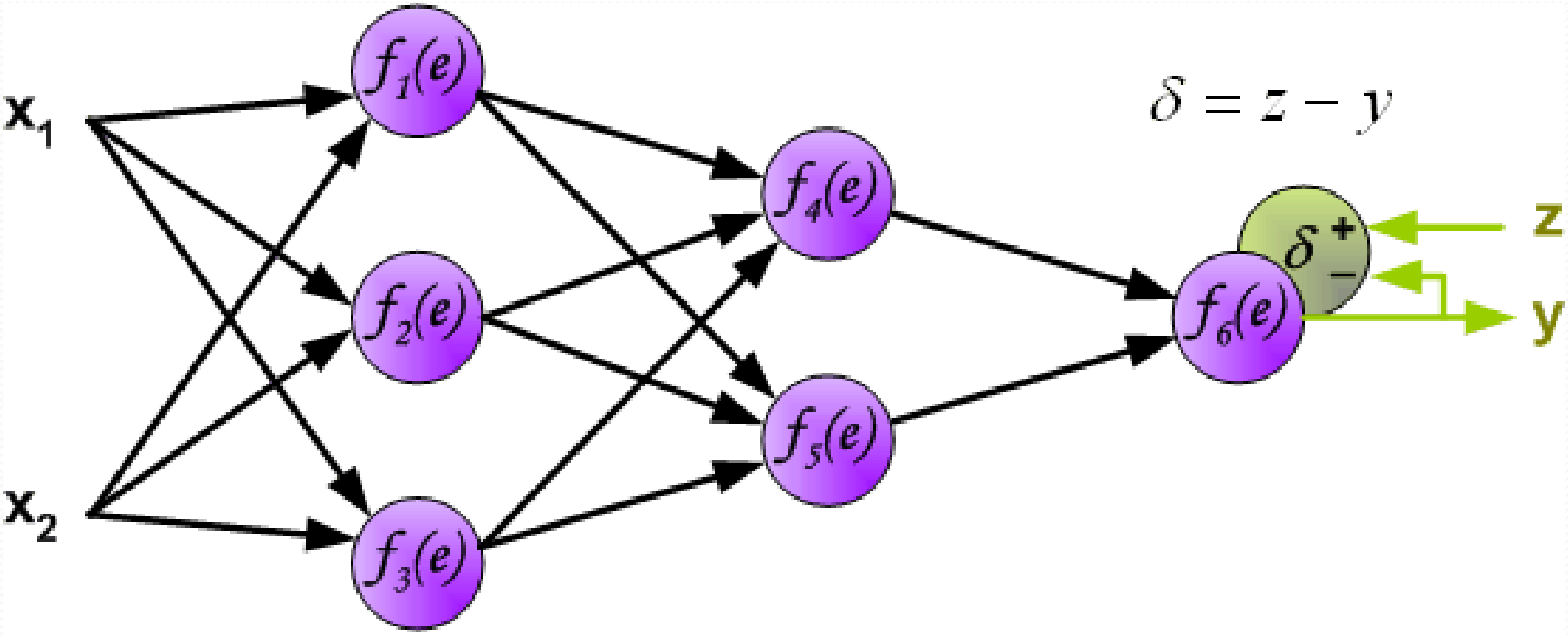
Dopredná sieť V



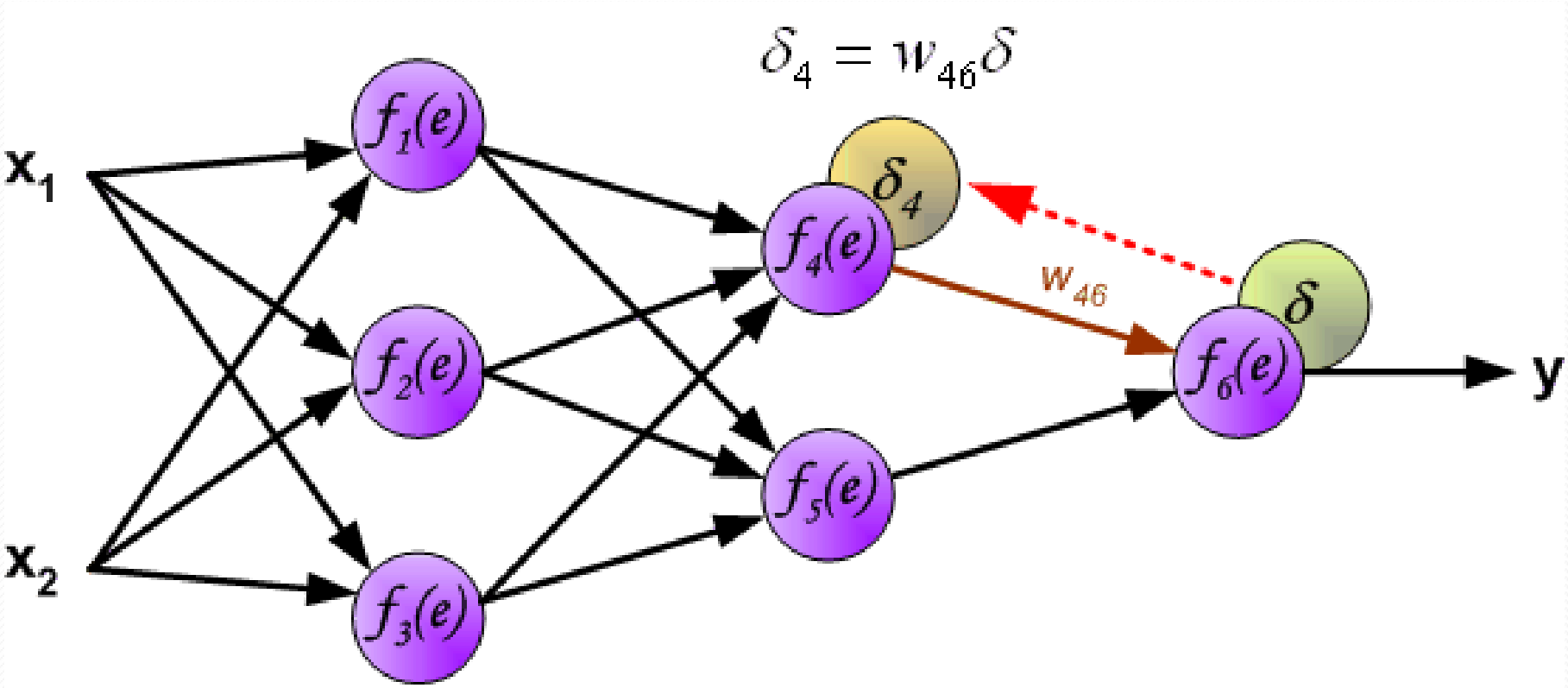
Dopredná sieť VI



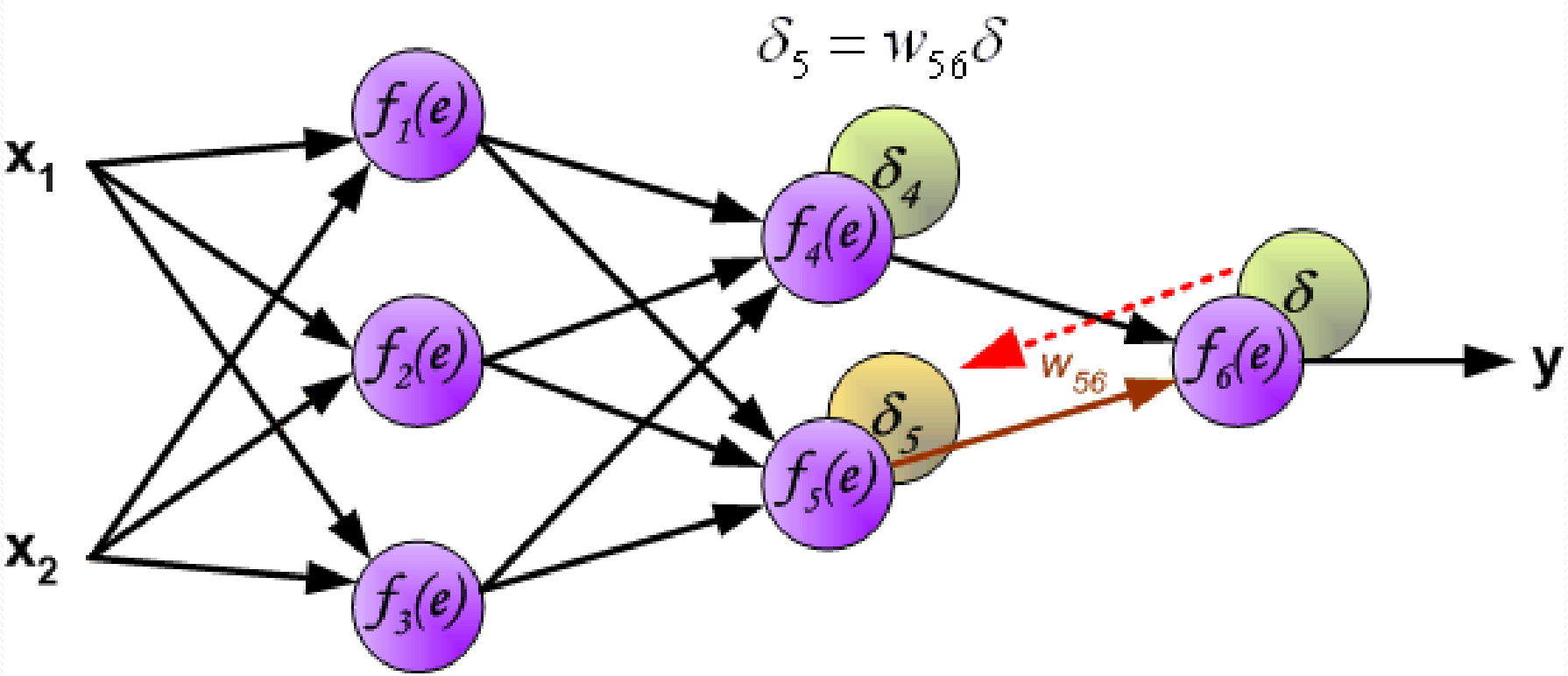
Dopredná sieť VII



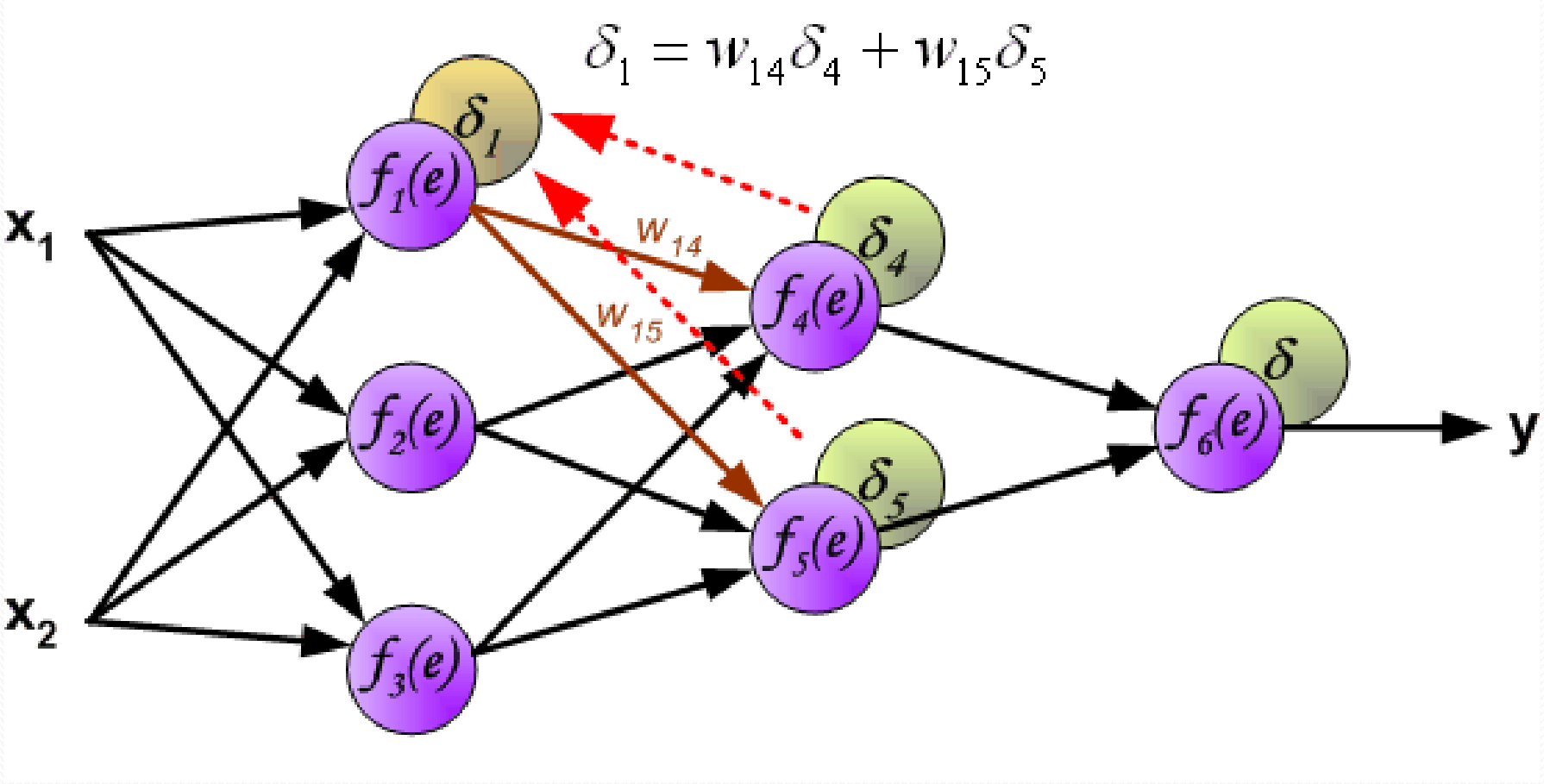
Dopredná sieť VIII



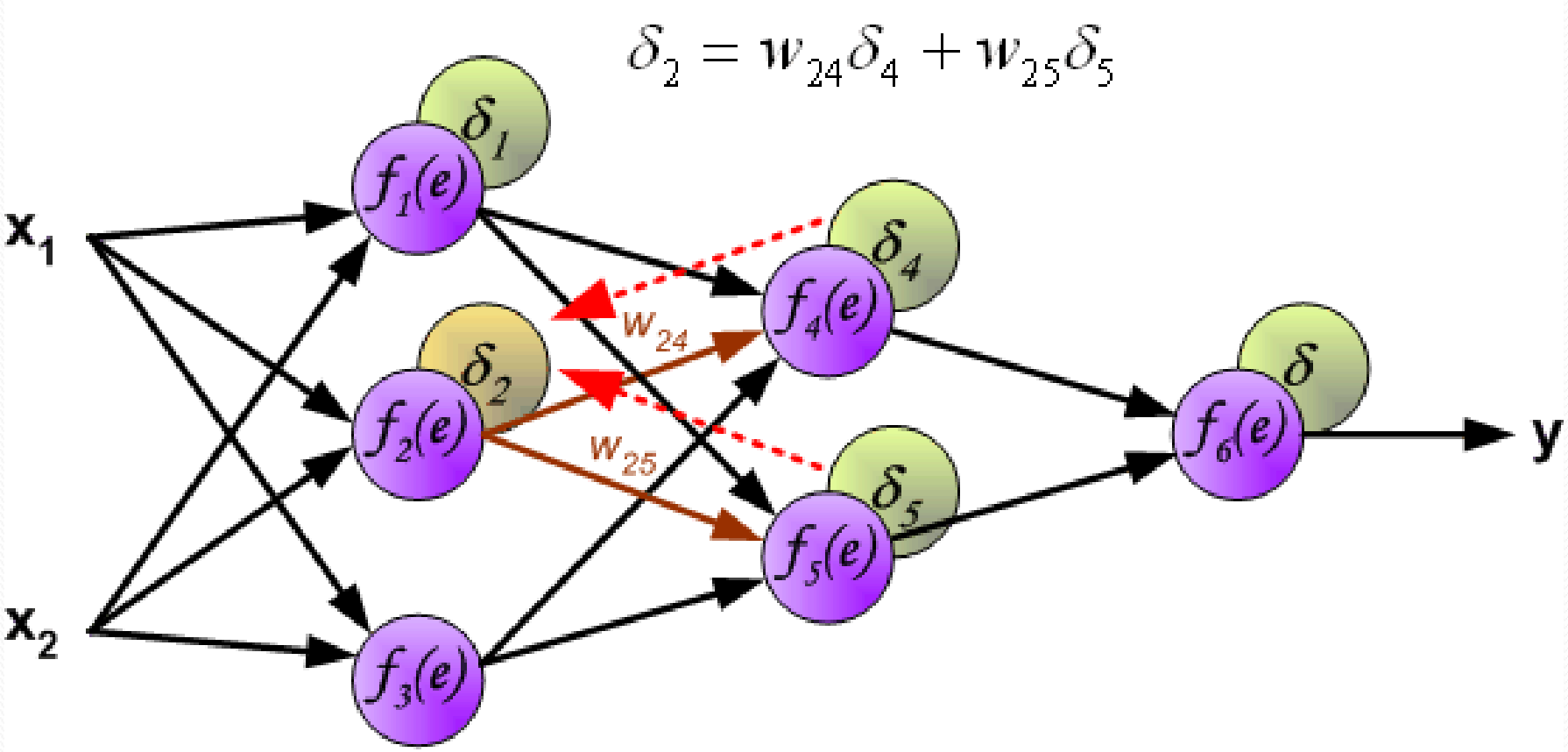
Dopredná sieť IX



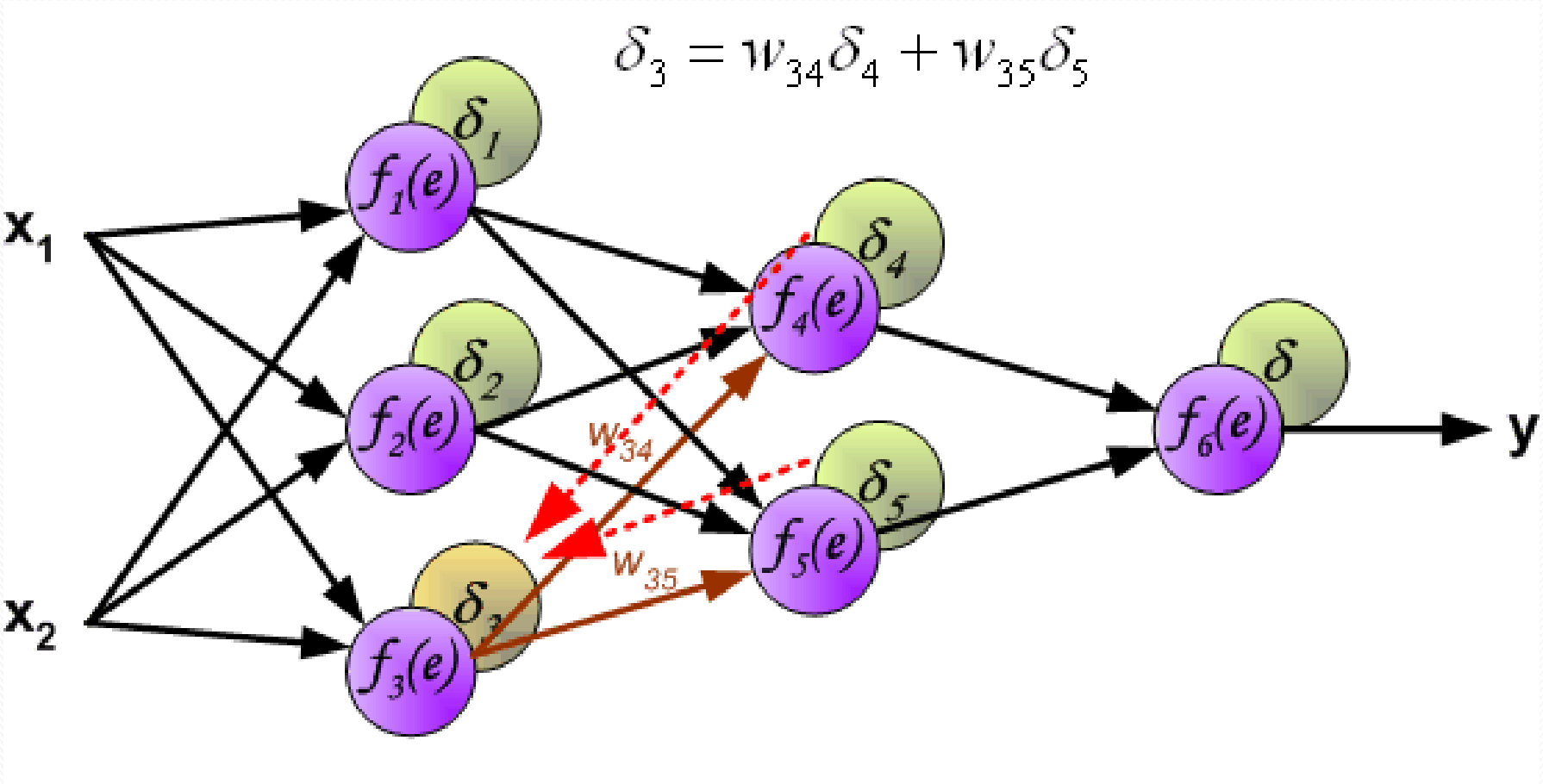
Dopredná sieť X



Dopredná sieť XI



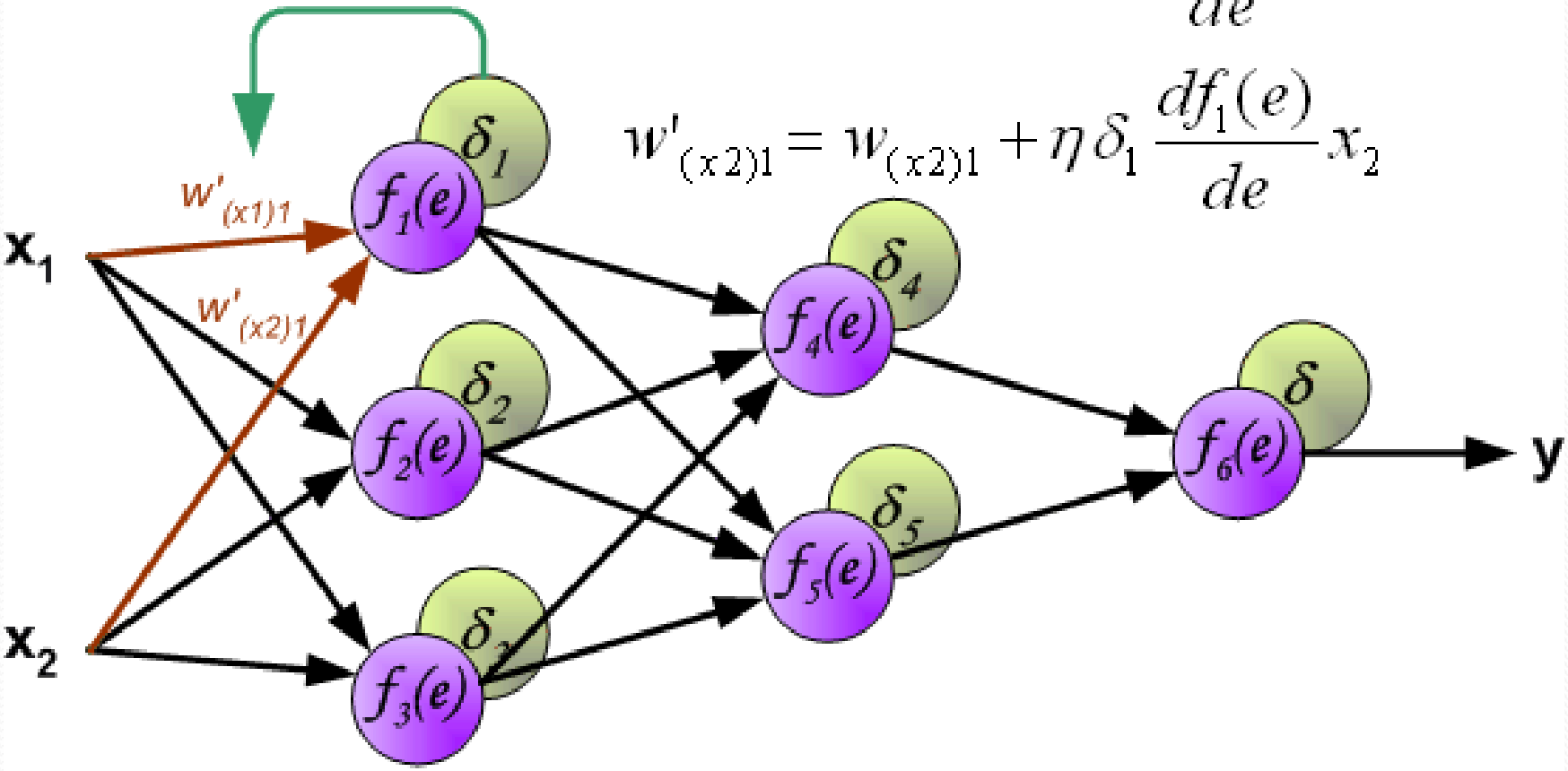
Dopredná sieť XII



Dopredná sieť XIII

$$w'_{(x1)1} = w_{(x1)1} + \eta \delta_1 \frac{df_1(e)}{de} x_1$$

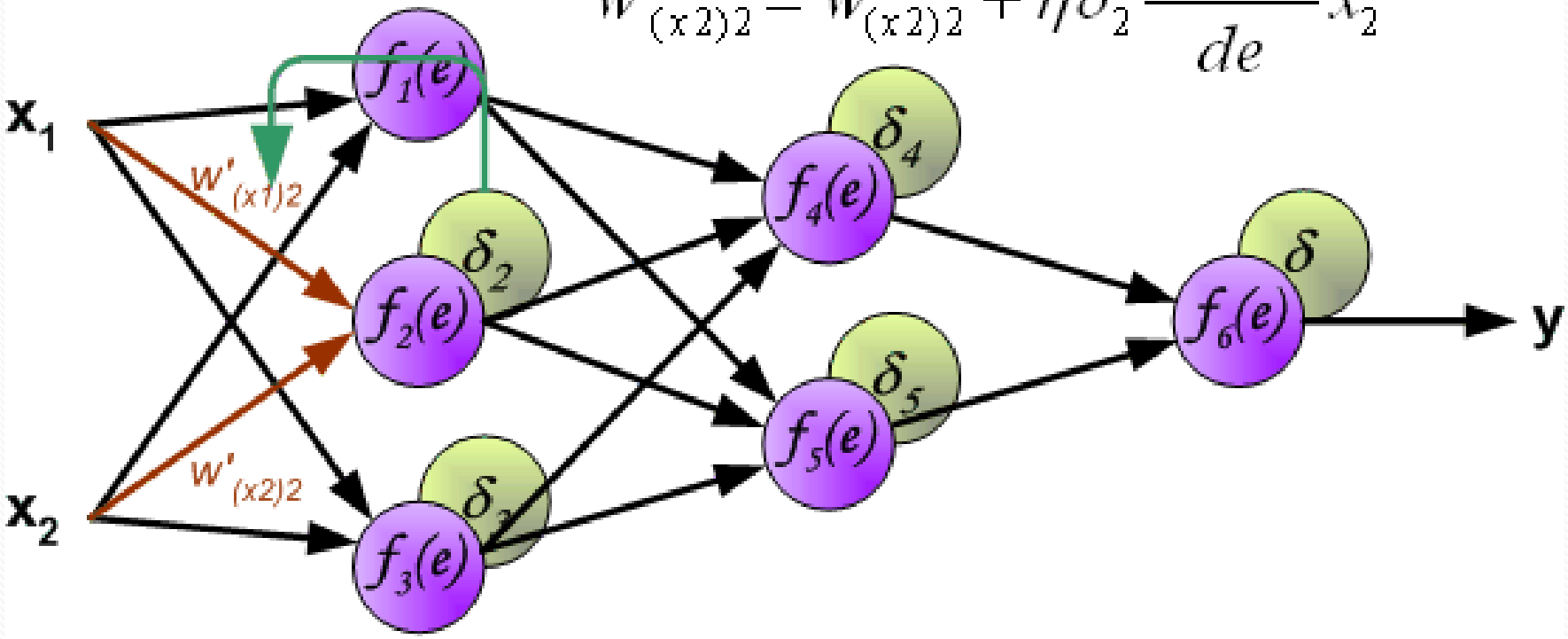
$$w'_{(x2)1} = w_{(x2)1} + \eta \delta_1 \frac{df_1(e)}{de} x_2$$



Dopredná sieť XIV

$$w'_{(x1)2} = w_{(x1)2} + \eta \delta_2 \frac{df_2(e)}{de} x_1$$

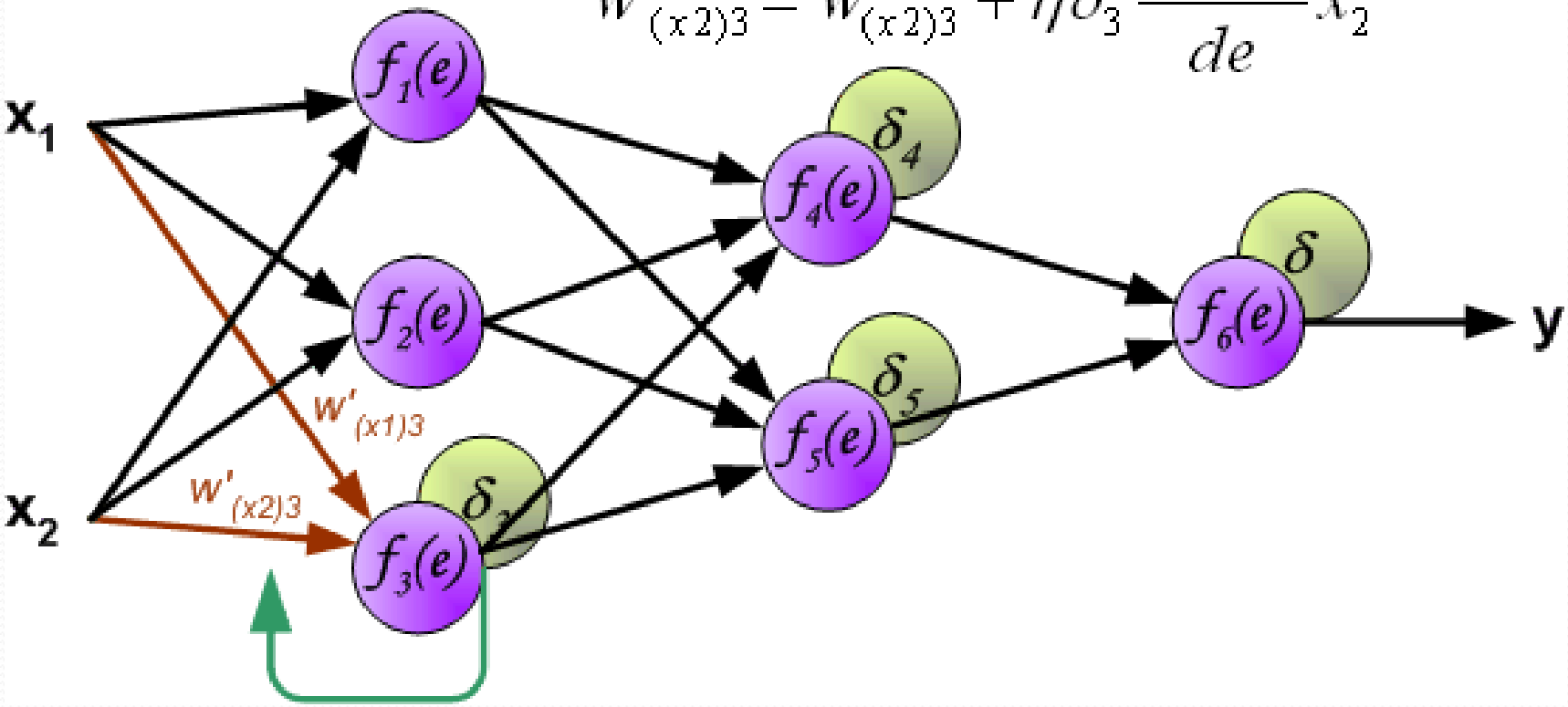
$$w'_{(x2)2} = w_{(x2)2} + \eta \delta_2 \frac{df_2(e)}{de} x_2$$



Dopredná sieť XV

$$w'_{(x1)3} = w_{(x1)3} + \eta \delta_3 \frac{df_3(e)}{de} x_1$$

$$w'_{(x2)3} = w_{(x2)3} + \eta \delta_3 \frac{df_3(e)}{de} x_2$$

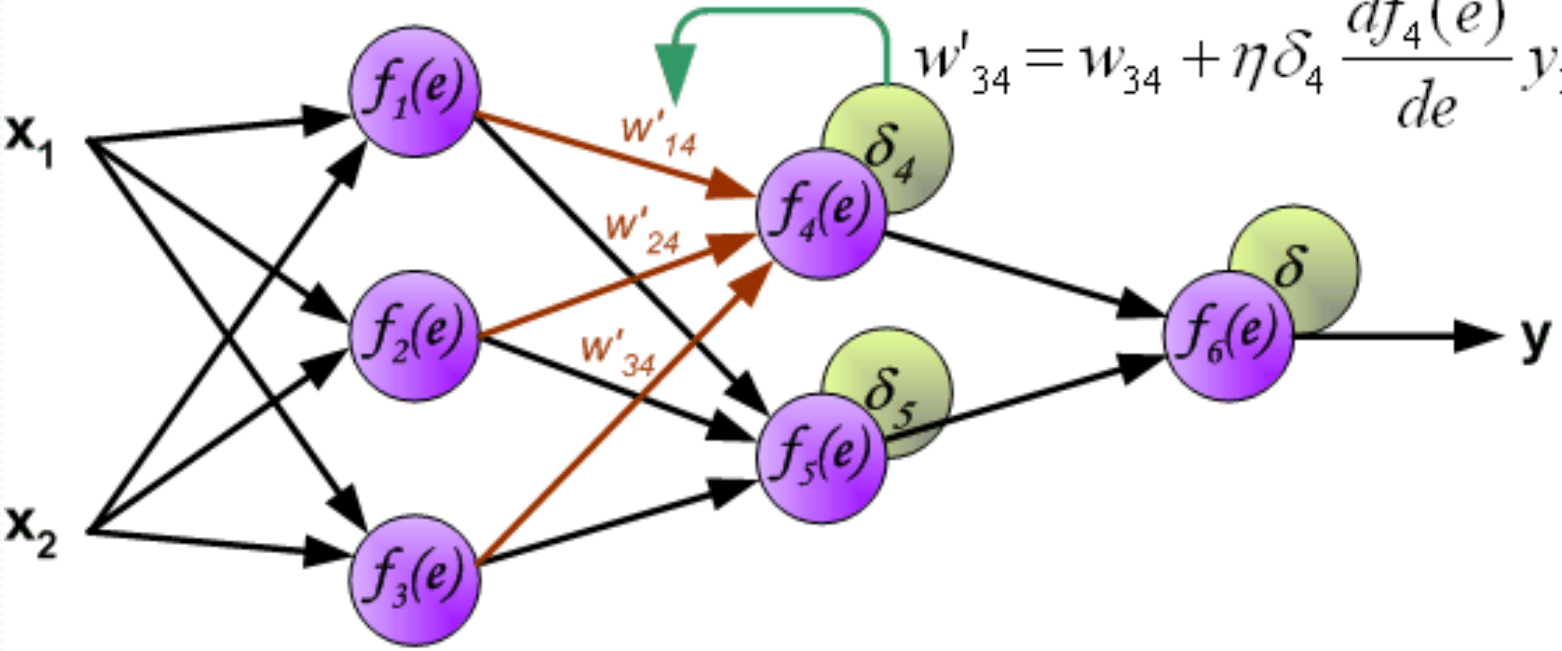


Dopredná sieť XVI

$$w'_{14} = w_{14} + \eta\delta_4 \frac{df_4(e)}{de} y_1$$

$$w'_{24} = w_{24} + \eta\delta_4 \frac{df_4(e)}{de} y_2$$

$$w'_{34} = w_{34} + \eta\delta_4 \frac{df_4(e)}{de} y_3$$

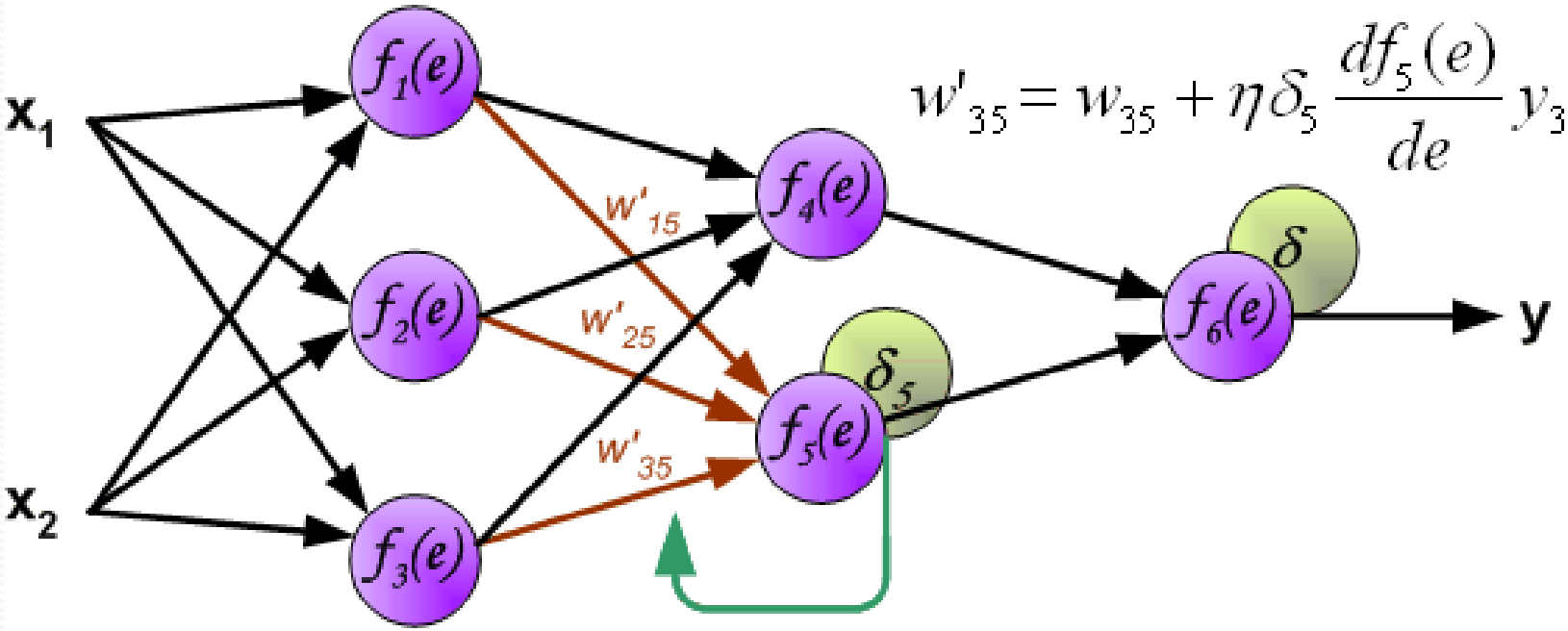


Dopredná sieť XVII

$$w'_{15} = w_{15} + \eta \delta_5 \frac{df_5(e)}{de} y_1$$

$$w'_{25} = w_{25} + \eta \delta_5 \frac{df_5(e)}{de} y_2$$

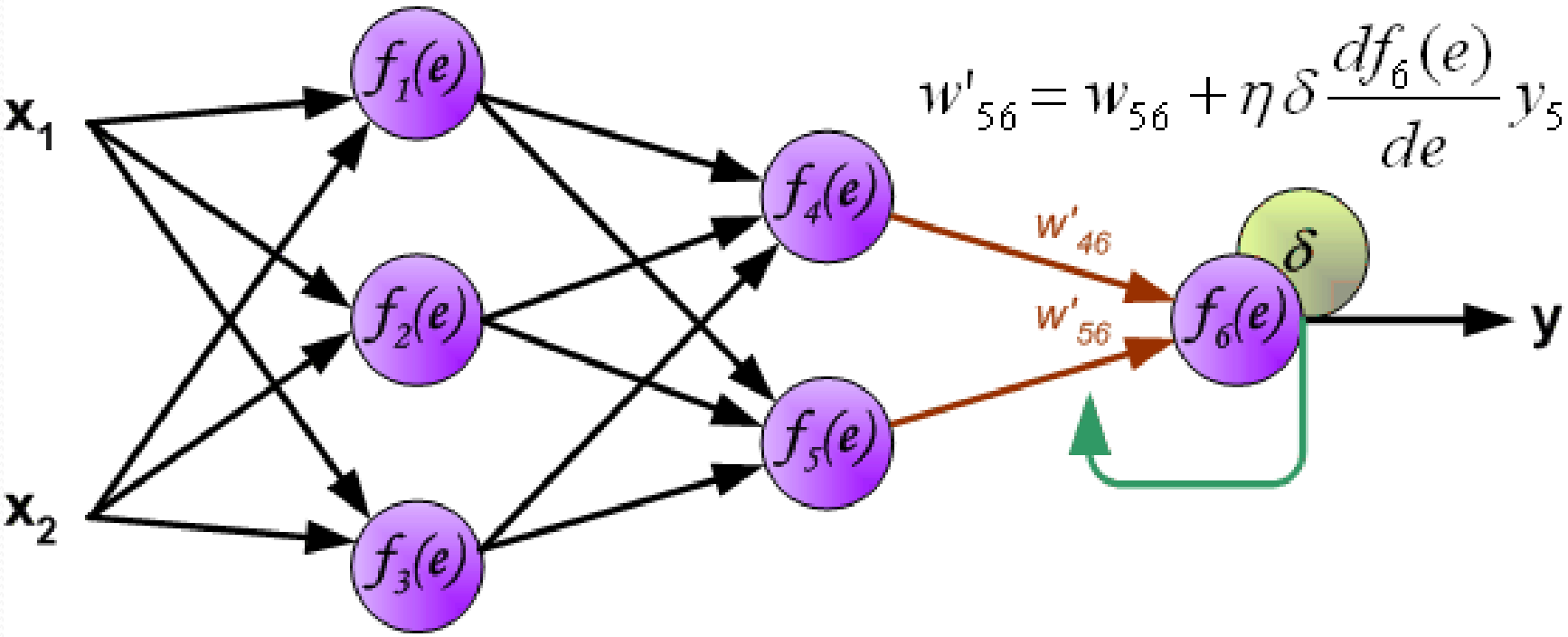
$$w'_{35} = w_{35} + \eta \delta_5 \frac{df_5(e)}{de} y_3$$



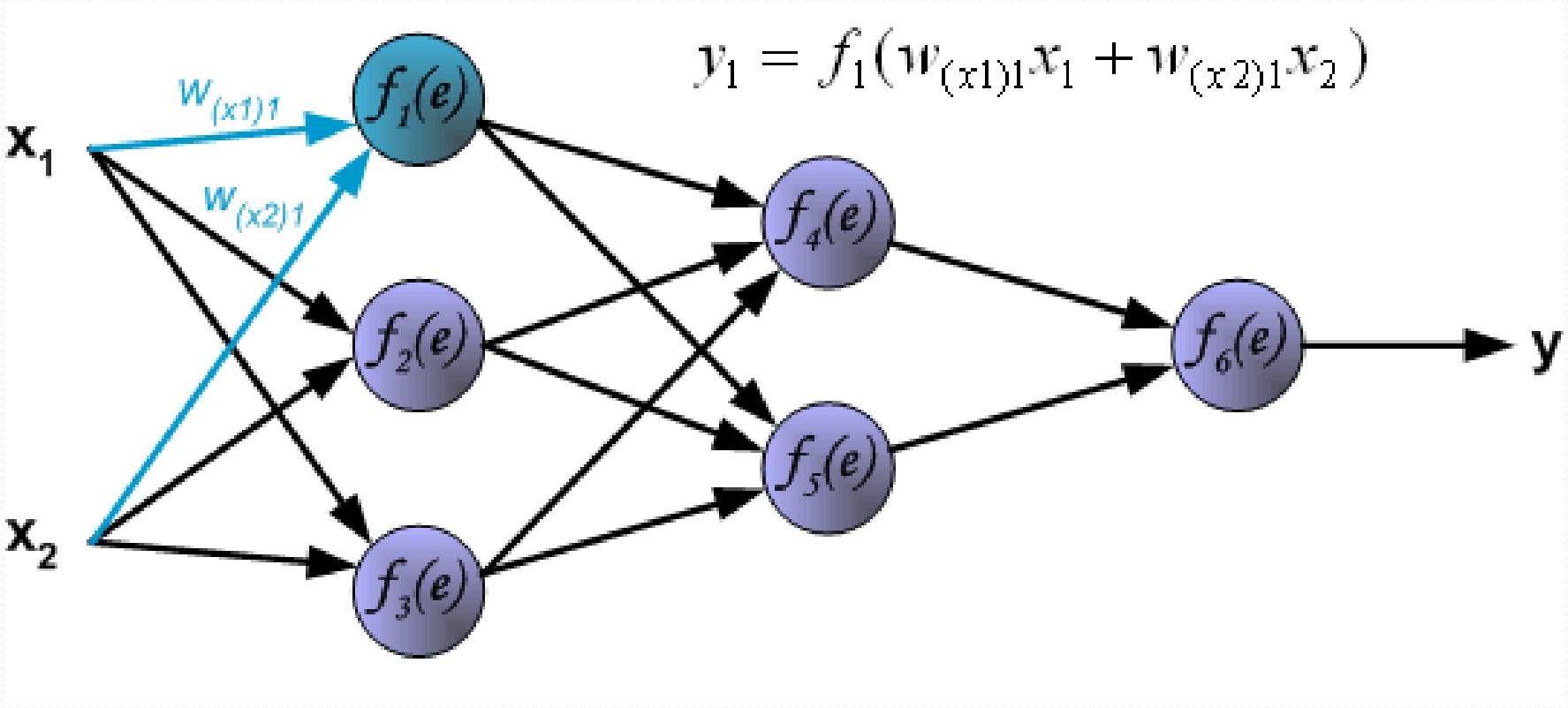
Dopredná sieť XVIII

$$w'_{46} = w_{46} + \eta\,\delta\,\frac{df_6(e)}{de}\,y_4$$

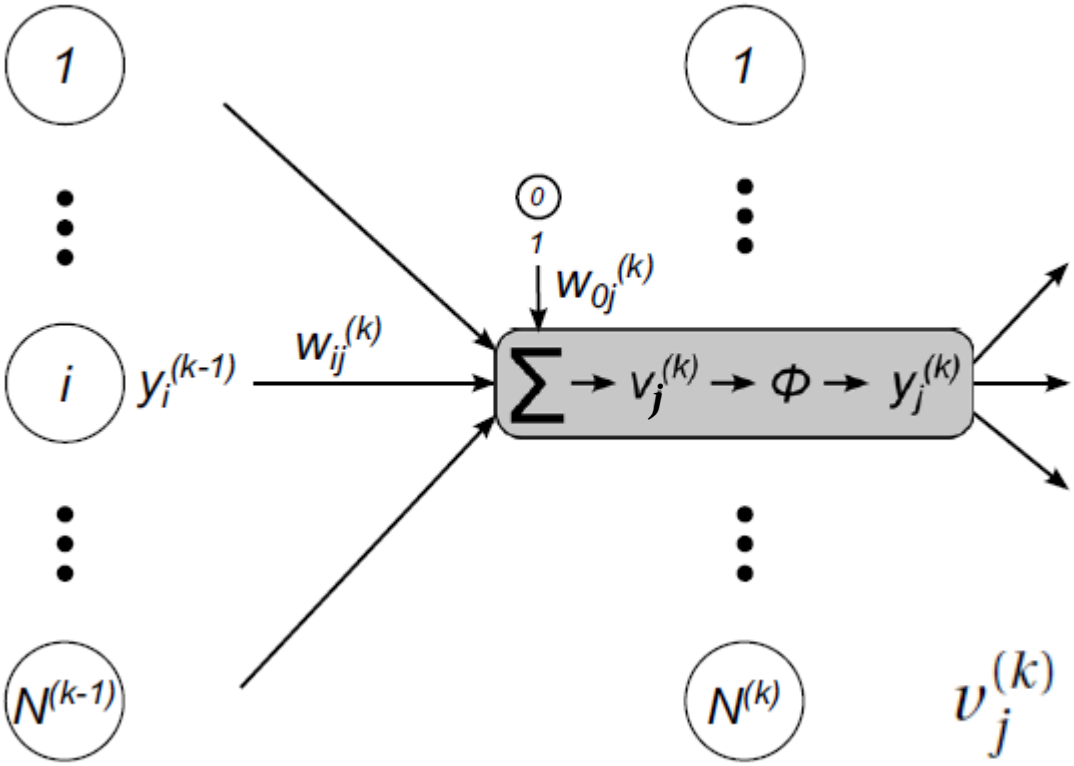
$$w'_{56} = w_{56} + \eta\,\delta\,\frac{df_6(e)}{de}\,y_5$$



Dopredná sieť XIX



Učenie neurónovej siete III



$$v_j^{(k)} = \sum_{i=1}^{N^{(k-1)}} w_{ij}^{(k)} y_i^{(k-1)} + w_{0j}^{(k)},$$

$$y_j^{(k)} = \phi^{(k)}\left(v_j^{(k)}\right)$$

Učenie neurónovej siete IV

$$\Delta w_{ij}^{(k)} = -\eta \frac{\partial E}{\partial w_{ij}^{(k)}}.$$

$$\Delta w_{ij}^{(k)} = -\eta \frac{\partial E}{\partial v_j^{(k)}} \frac{\partial v_j^{(k)}}{\partial w_{ij}^{(k)}}$$

$$\frac{\partial v_j^{(k)}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)}$$

$$\delta_j^{(k)} = -\frac{\partial E}{\partial v_j^{(k)}} = -\frac{\partial E}{\partial y_j^{(k)}} \frac{\partial y_j^{(k)}}{\partial v_j^{(k)}}$$

$$\Delta w_{ij}^{(k)} = \eta \, \delta_j^{(k)} \, y_i^{(k-1)}$$

- Treba vyčísliť $\delta_j^{(k)}$

Šírenie chyby vo výstupnej vrstve

- Vo výstupnej vrstve je chyba neurónu daná chybou, akú nadobúda voči očakávanému výstupu
- Často používaným optimalizačným kritériom je stredná kvadratická chyba (MSE)
- Stredná kvadratická chyba:

$$\mathrm{E}(\mathbf{W}) = \frac{1}{2}\sum_{j=1}^{M}(y_j^{(m)} - o_j)^2$$

Šírenie chyby vo výstupnej vrstve II

- Vypočítame $\delta_j^{(k)}$

$$\delta_j^{(m)} = -\frac{\partial E}{\partial v_j^{(m)}} = -\frac{\partial E}{\partial y_j^{(m)}} \frac{\partial y_j^{(m)}}{\partial v_j^{(m)}}$$

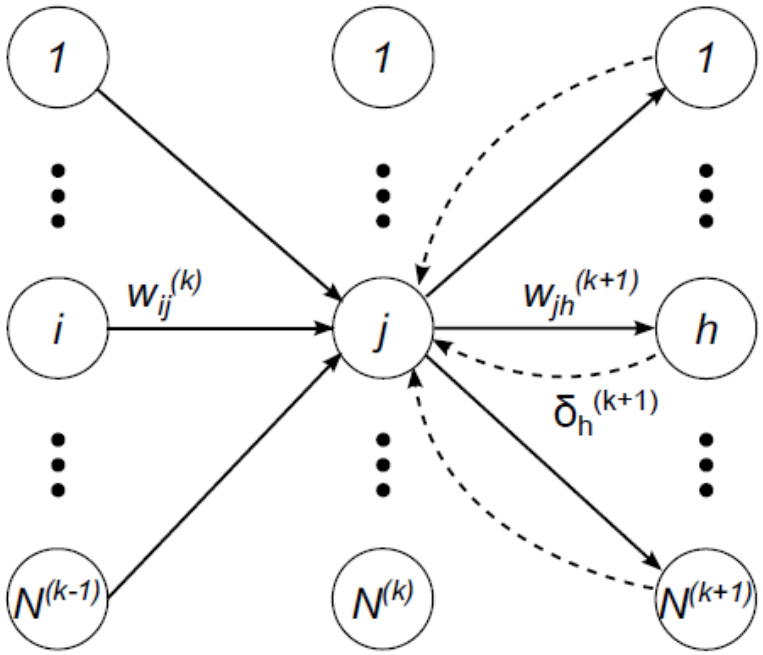
$$\frac{\partial E}{\partial y_j^{(m)}} = y_j^{(m)} - o_j$$

$$\frac{\partial y_j^{(m)}}{\partial v_j^{(m)}} = \phi'(v_j^{(m)})$$

$$\delta_j^{(m)} = (o_j - y_j^{(m)})\phi'(v_j^{(m)})$$

Šírenie chyby v skrytých vrstvách

- Chyba *j*-teho neurónu v *k*-tej vrstve sa propaguje zo všetkých neurónoch *(k+1)*-vej vrstvy



$$\frac{\partial E}{\partial y_j^{(k)}} = \sum_{h=1}^{N^{(k+1)}} \frac{\partial E}{\partial v_h^{(k+1)}} \frac{\partial v_h^{(k+1)}}{\partial y_j^{(k)}} =$$

$$= -\sum_{h=1}^{N^{(k+1)}} \delta_h^{(k+1)} w_{jh}^{(k+1)}$$

$$\frac{\partial y_j^{(k)}}{\partial v_j^{(k)}} = \phi'(v_j^{(k)})$$

Šírenie chyby v skrytých vrstvách II

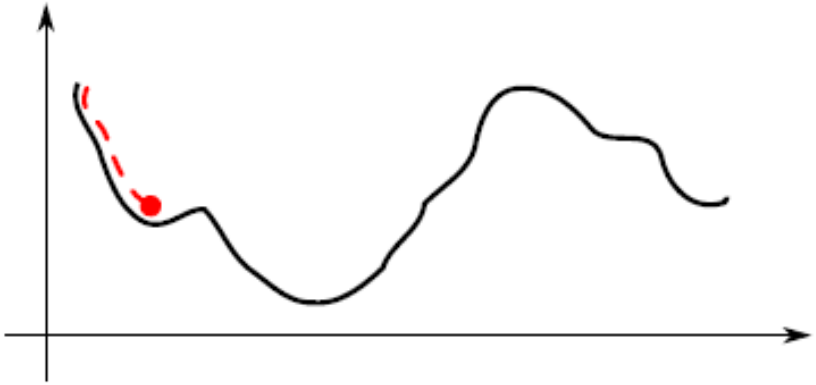
$$\delta_j^{(k)} = \phi'(v_j^{(k)}) \sum_{h=1}^{N^{(k+1)}} \delta_h^{(k+1)} w_{jh}^{(k+1)}$$

$$\delta_j^{(m)} = (o_j - y_j^{(m)})\phi'(v_j^{(m)})$$

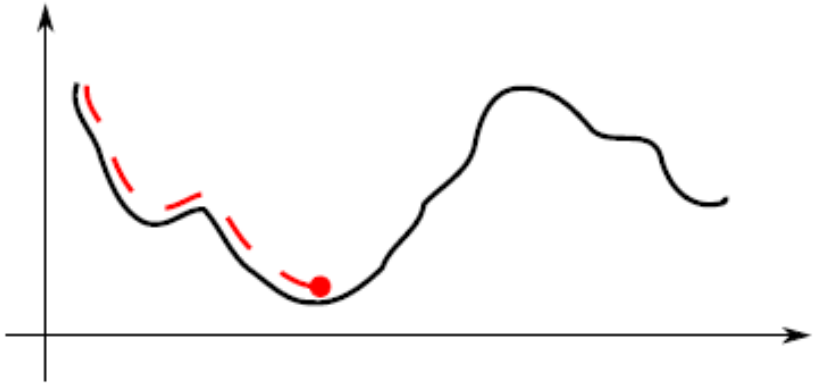
$$\mathbf{W} \leftarrow \mathbf{W} + \Delta\mathbf{W},$$

$$\Delta w_{ij}^{(k)} = \eta\, \delta_j^{(k)}\, y_i^{(k-1)}$$

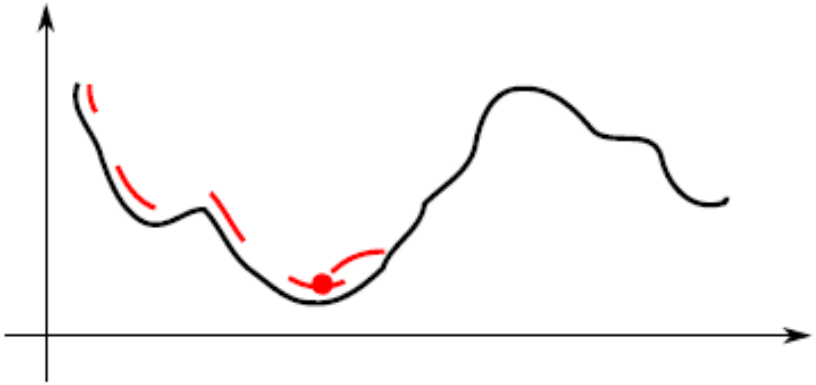
Rýchlosť učenia neurónovej siete



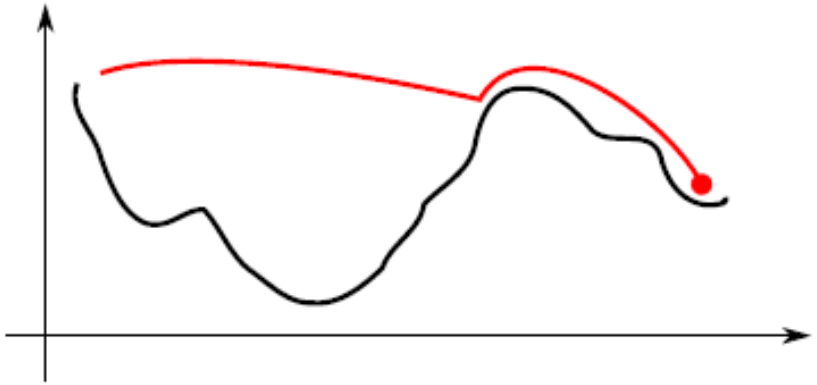
(a) Nízka rýchlosť učenia sa.



(b) Optimálna rýchlosť učenia sa.



(c) Vysoká rýchlosť učenia sa.



(d) Veľmi vysoká rýchlosť učenia sa.

Rýchlosť učenia neurónovej siete II

- Pri nízkej rýchlosti – pomalý tréning, riešenie môže uviaznuť v lokálnom minime
- Optimálna rýchlosť učenia – pomalý tréning, ale bez oscilácií, dosiahne globálne minimum
- Pri veľmi vysokej rýchlosti – rýchlejší tréning, ale riešenie môže oscilovať, prípadne divergovať a neskončiť v globálnom optime

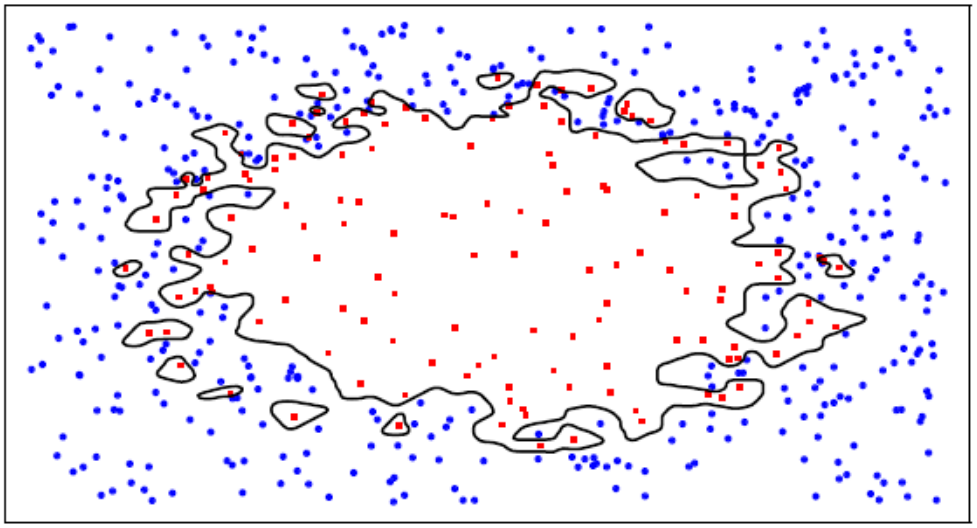
Premenlivá rýchlosť učenia

- Použijeme heuristiku:
  - Ak nová chyba prevýši starú o stanovený prah, ideme zlým smerom ⟹ nové váhy sa zahodia a rýchlosť učenia sa zníži
  - Malé zvýšenie chyby povolíme kvôli možnosti dostať sa z lokálneho minima. Ak je nová chyba vyššia ako stará o menej než stanovený prah, váhy sa upravia
  - Ak je nová chyba nižšia ako stará, smerujeme k minimu, upravíme váhy a zvýšime rýchlosť učenia

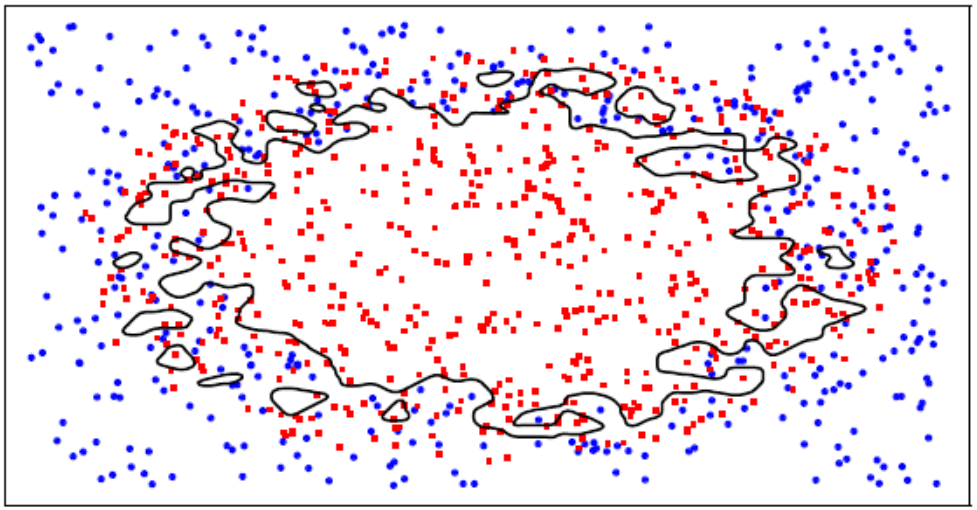
Ukončenie učenia sa NN

- Ideálne – keď dosiahneme globálne minimum chybovej funkcie
- Problém:
  - Preučenie siete – malá chyba na trénovacích dátach a veľká na testovacích
- Riešenie:
  - Ukončíme tréning, keď sa chyba v testovacej množine začne zvyšovať

Ukážka preučeného klasifikátora



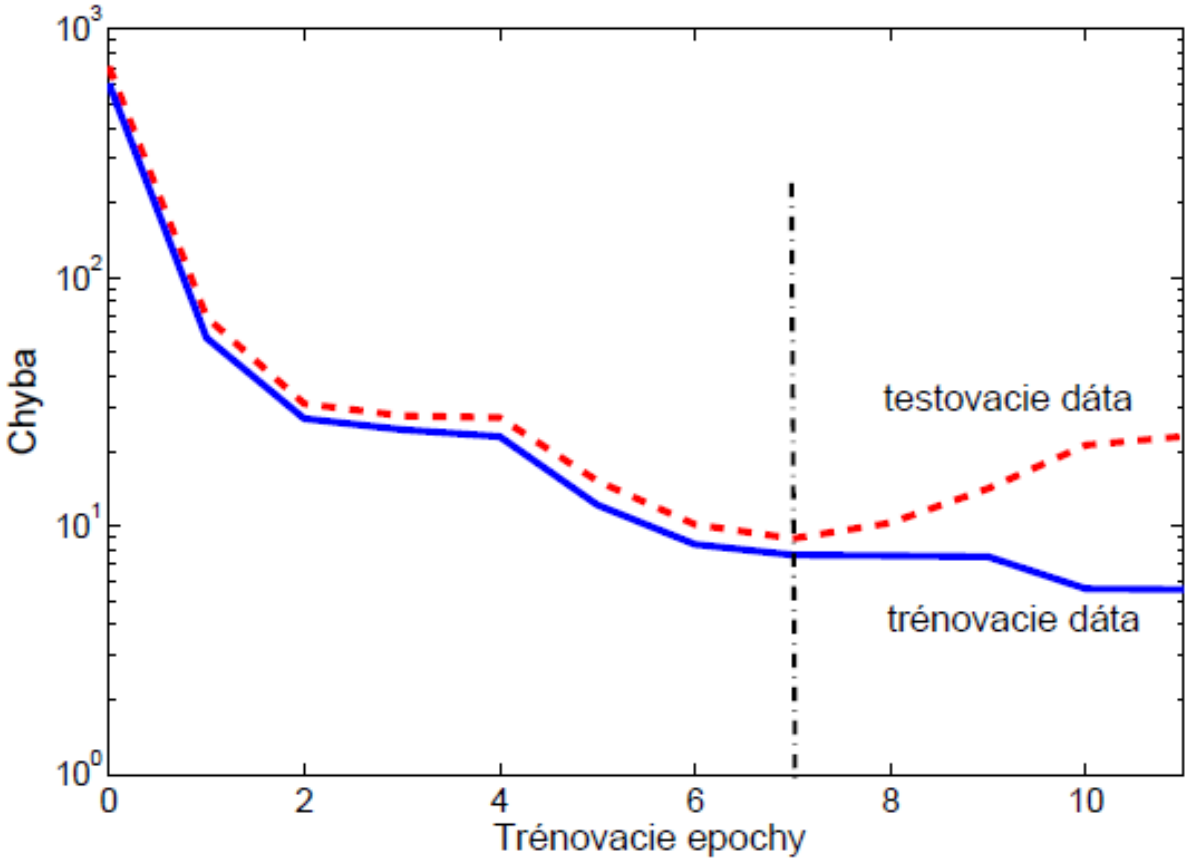
(a) Trénovacie dáta.



(b) Testovacie dáta.

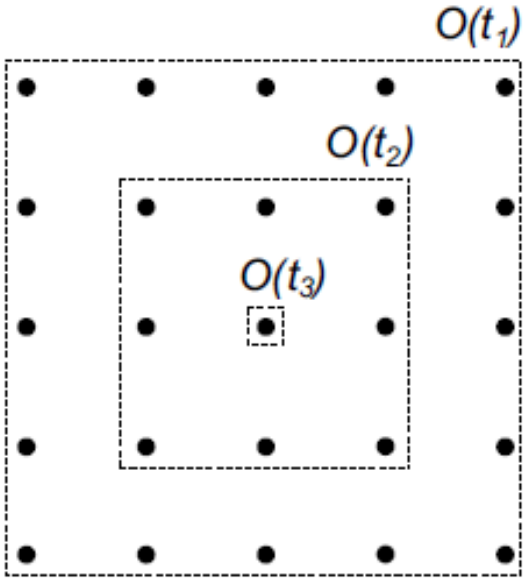
Ukončenie učenia sa NN II

- Riešenie:
  - Určíme trénovaciu a validačnú množinu
  - Ukončíme tréning, keď sa chyba vo validačnej množine začne zvyšovať

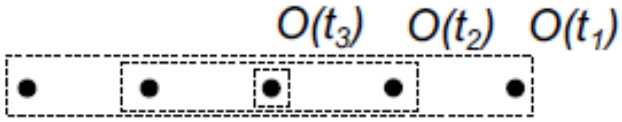


SOM

- Samoorganizujúce sa mapy (Kohonenove)
- Neuróny sú umiestnené v mriežke
- Vstupné dáta sú spojené s každým neurónom



- **Neriadená metóda**



SOM II

- SOM vytvorí nízko-rozmernú (obyčajne dvoj-rozmernú) diskrétnu reprezentáciu príznakového priestoru vzoriek z trénovacej množiny
- Táto reprezentácia sa nazýva mapa a môže slúžiť aj na redukciu dimenzionality
- SOM používa tzv. súťažné učenie, na rozdiel od ostatných ANN, ktoré používajú učenie zamerané na zníženie chyby (ako back propagation)

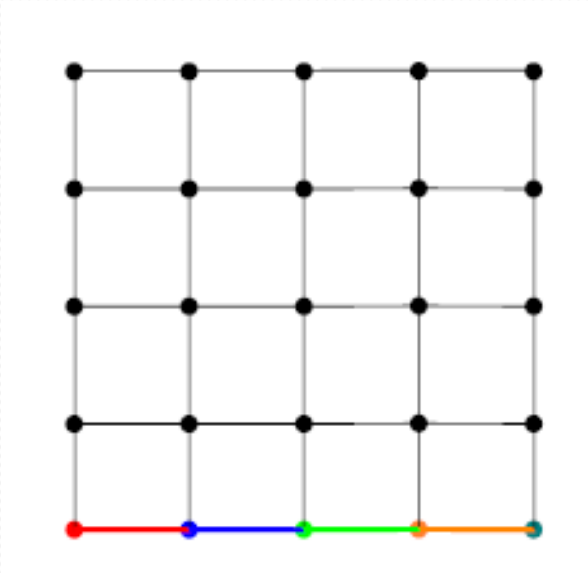
SOM III

- SOM je užitočný nástroj na zobrazenie nízko-rozmerných pohľadov na viacrozmerné dáta
- SOM s malým počtom vrcholov sa správajú podobne ako *K*-means, tie s väčším počtom vrcholov reorganizujú dáta topologicky
- Užitočný nástroj na vizualizáciu je tzv. U-matica, ktorá obsahuje priemernú vzdialenosť medzi váhami vrcholu a jeho susedov

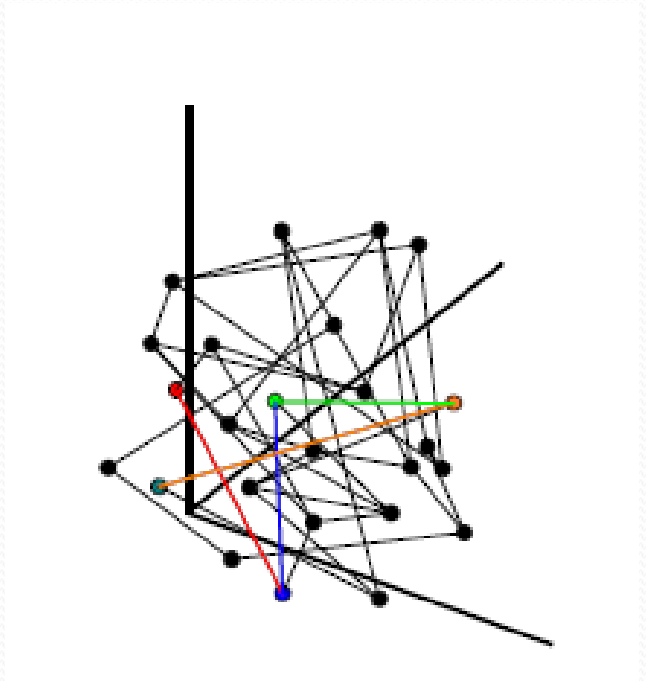
Dva priestory SOM

Mriežka SOM

Topologické susedstvo



Priestor váh



- Na začiatku môžu byť váhy topologicky blízkych susedov ďaleko od seba

Učenie SOM

Inicializácia
Učenie siete:
1. Vstup vektora príznakov
2. Určenie víťazného neurónu
3. Víťaz berie všetko

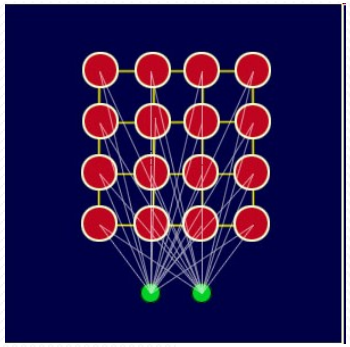
- 0. krok
- Inicializácia a normalizácia váhových vektorov

$$\overline{w}_i = \frac{w_i}{\| w_i \|}, \quad 1 \leq i \leq c$$

Učenie SOM II

1. Krok

Vstup vektora príznakov



2. Krok

Určenie víťazného neurónu:

$$i^* = \arg\min_i \|\mathbf{x} - \mathbf{w}_i\|$$

Jeho váhový vektor je najbližšie k aktuálnemu vstupu v zmysle Euklidovskej vzdialenosti

3. Krok – víťaz berie všetko – víťaznému neurónu zmeníme váhy

Učenie SOM III

- **Učenie:** váhové vektory víťazného neurónu a jeho topologických susedov sa posunú smerom k aktuálnemu vstupu podľa vzťahu:

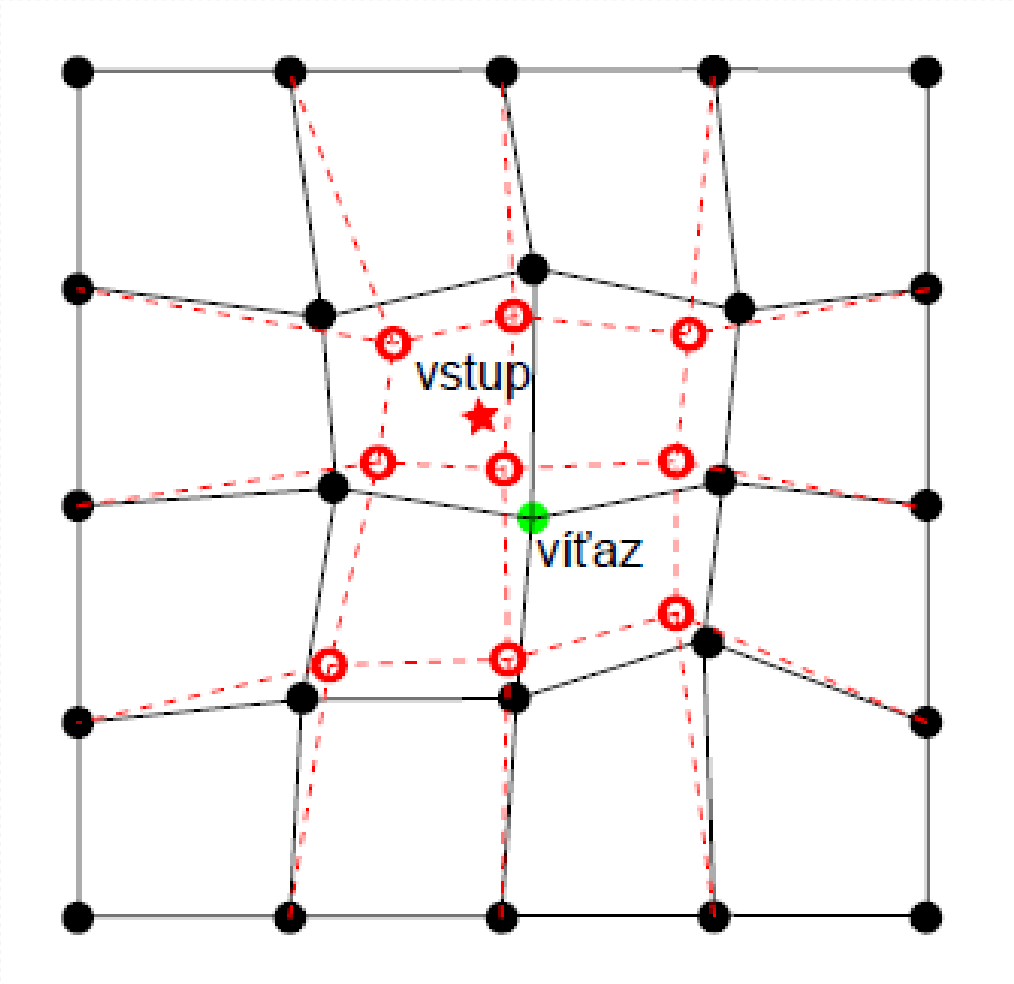
$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t) O(i, i^*, t) \|\mathbf{x} - \mathbf{w}_i(t)\|$$

$\alpha(t) \in (0,1)$ Rýchlosť učenia (časom klesá k nule), čím sa zabezpečí ukončenie procesu učenia

$O(i, i^*, t)$ Funkcia okolia – definuje rozsah kooperácie medzi neurónmi

Učenie SOM IV

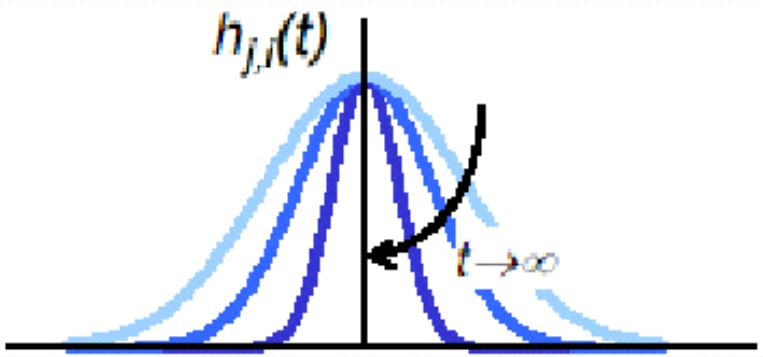
- Ilustrácia

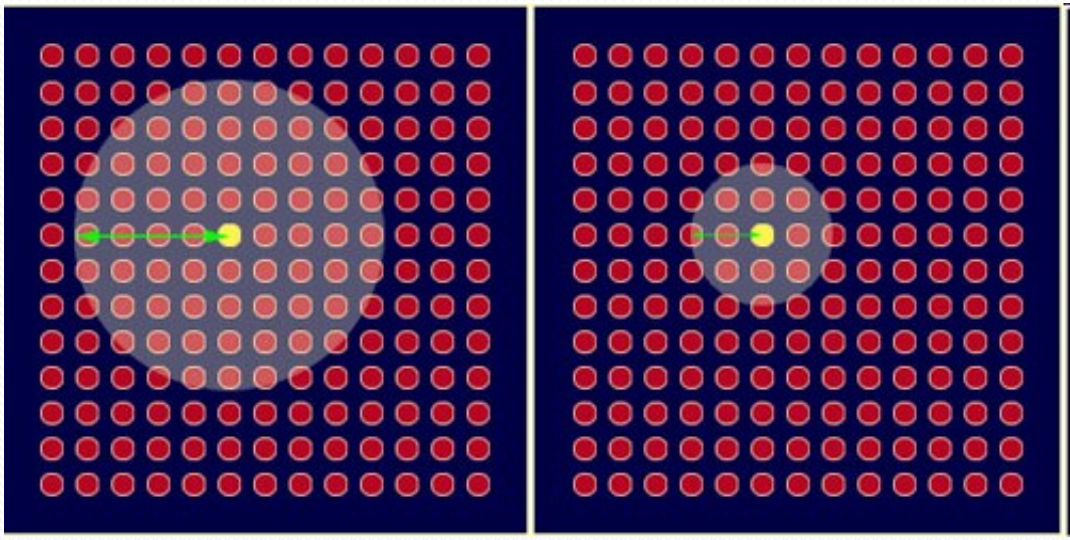


Voľba okolia

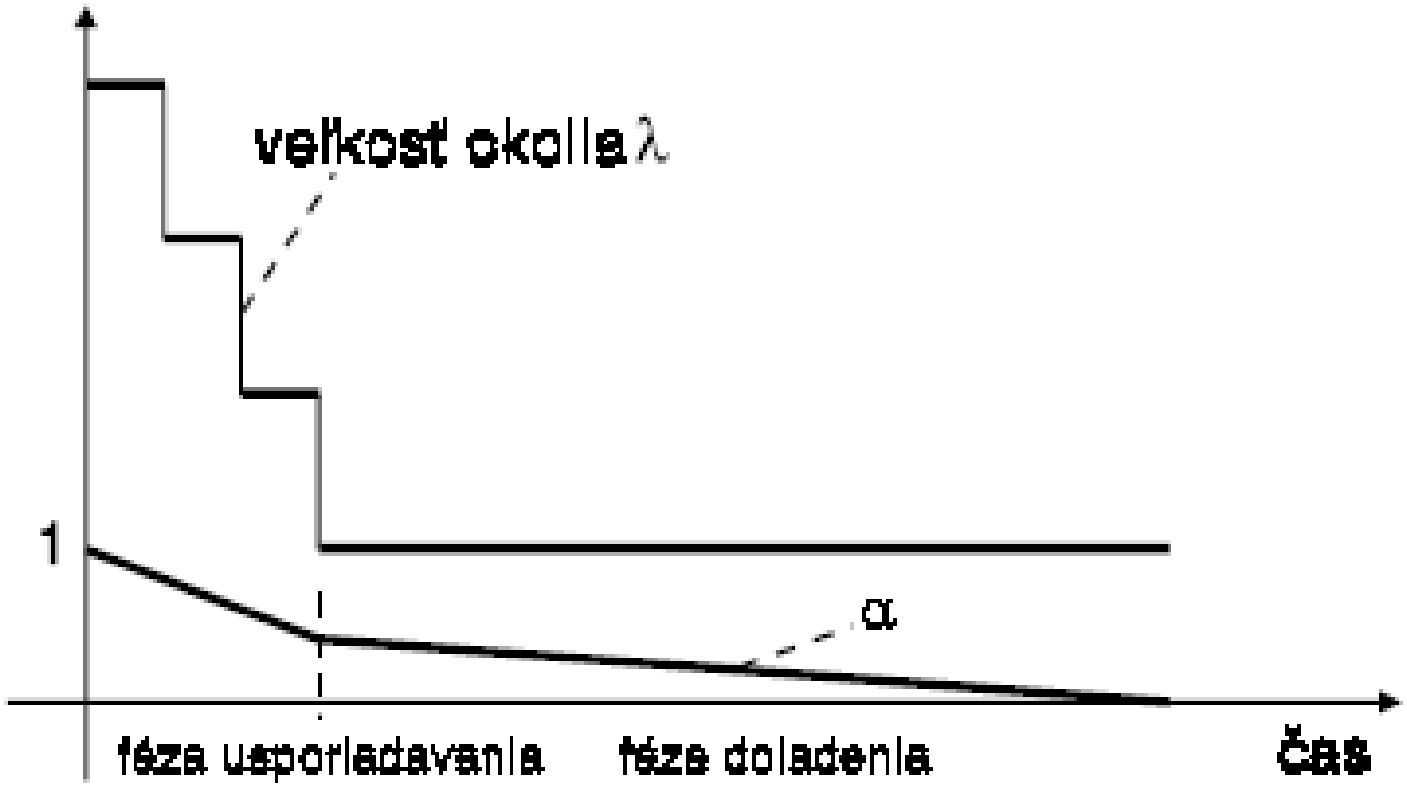
- Gaussovské okolie (alebo iné metriky)

$$O(i, i^*, t) = \exp\left(\frac{\mathbf{w}_{i^*}(t) - \mathbf{w}_i(t)}{\sigma}\right)^2$$





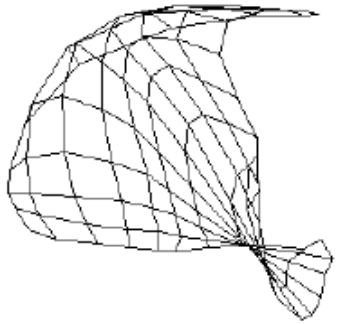
Veľkosť okolia a rýchlosť učenia



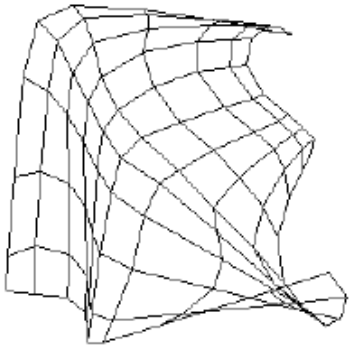
Obrázok 7.6. Príklad vhodnej voľby poklesu polomeru okolia a parametra α počas učenia SOM.

Ukážka učenia SOM

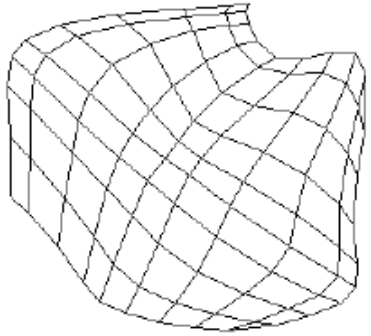
(a) Náhodné čísla z oblasti $\langle -0,5;0,5\rangle \times \langle -0,5;0,5\rangle$.



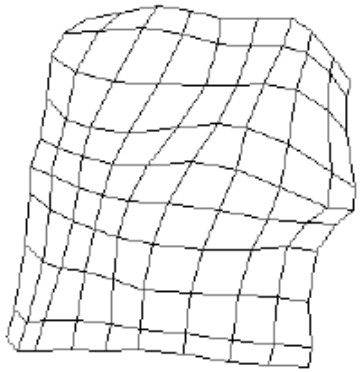
(b) Váhy po 100 iteráciách.



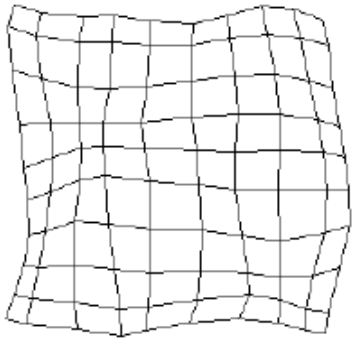
(c) Váhy po 200 iteráciách.



(d) Váhy po 600 iteráciách.



(e) Váhy po 3000 iteráciách.

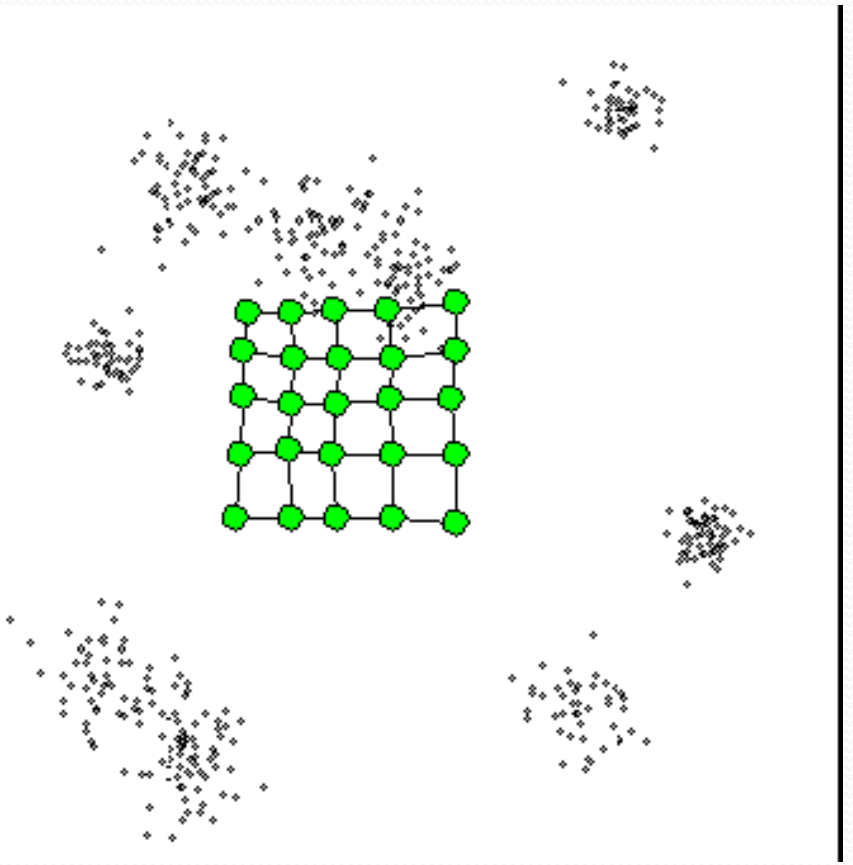


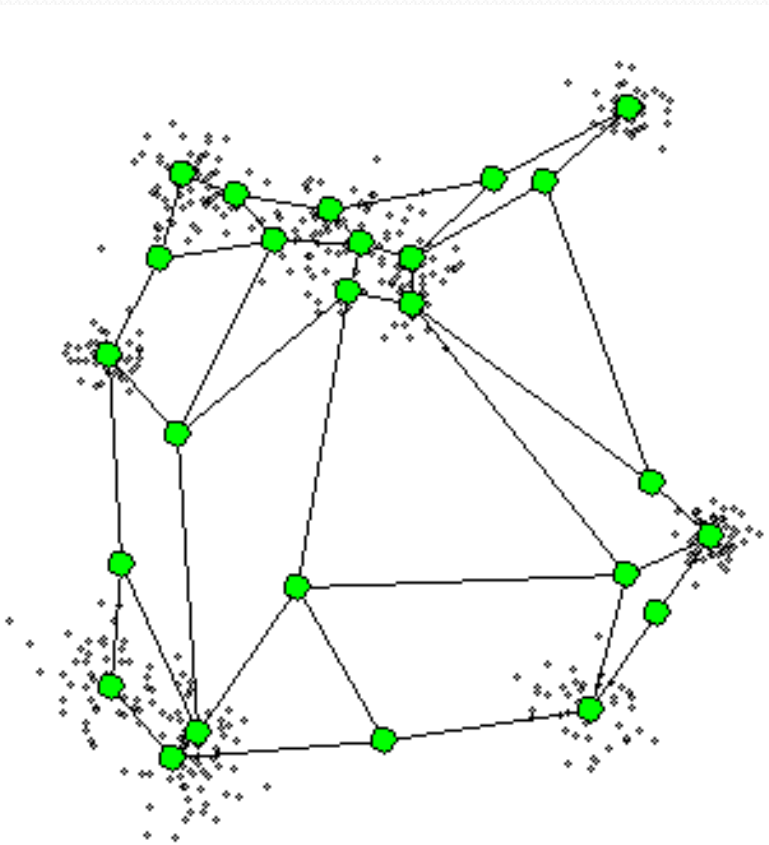
(f) Váhy po 7000 iteráciách.

Ukážka učenia SOM II

- Vstupnými vektormi sú rovnomerne rozdelené náhodné čísla z oblasti $\langle -1,1\rangle \times \langle -1,1\rangle$.
- Tréning SOM na rovnomerne rozdelených dátach. (a) Inicializácia (b), (c) a (d) Fáza usporiadavania. (e) a (f) Fáza dolaďovania.
- Na nerovnomerne rozdelených dátach sa váhové vektory presunú smerom do jednotlivých zhlukov dát

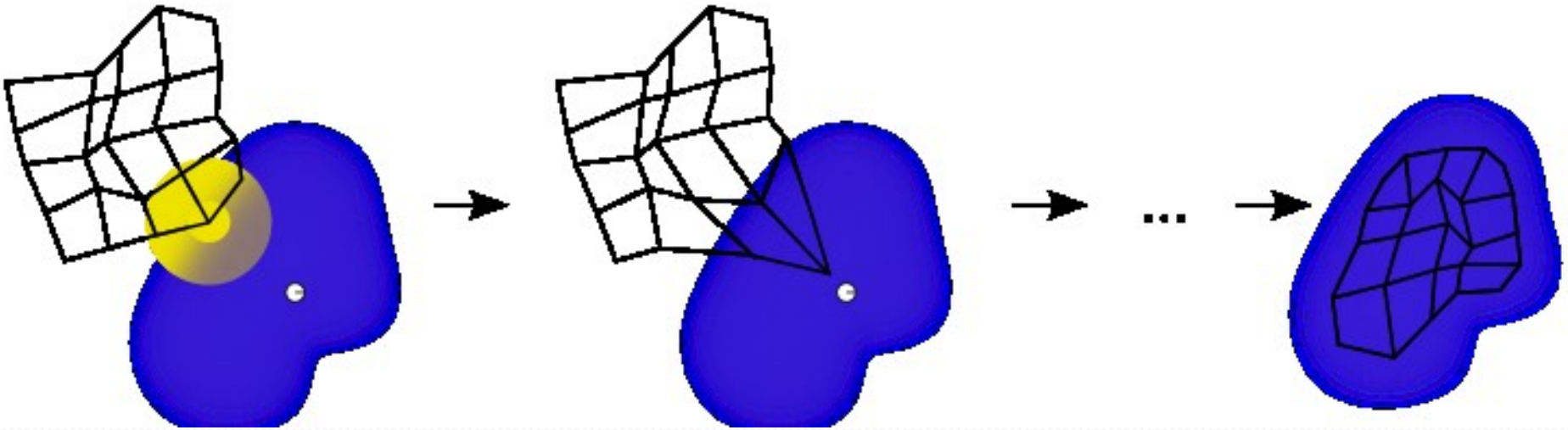
Ukážka učenia SOM III





Ukážka učenia SOM IV

- Modrá škvrna predstavuje rozdelenie trénovacích dát a biely kruh aktuálny trénovací údaj z tohto rozdelenia
- Najprv je SOM umiestnená náhodne. Žltý kruh označuje vrchol, ktorá je najbližšie k trénovacej vzorke a postupne sa k nej začne približovať aj so susedmi
- Po veľa iteráciách kopíruje SOM rozdelenie dát

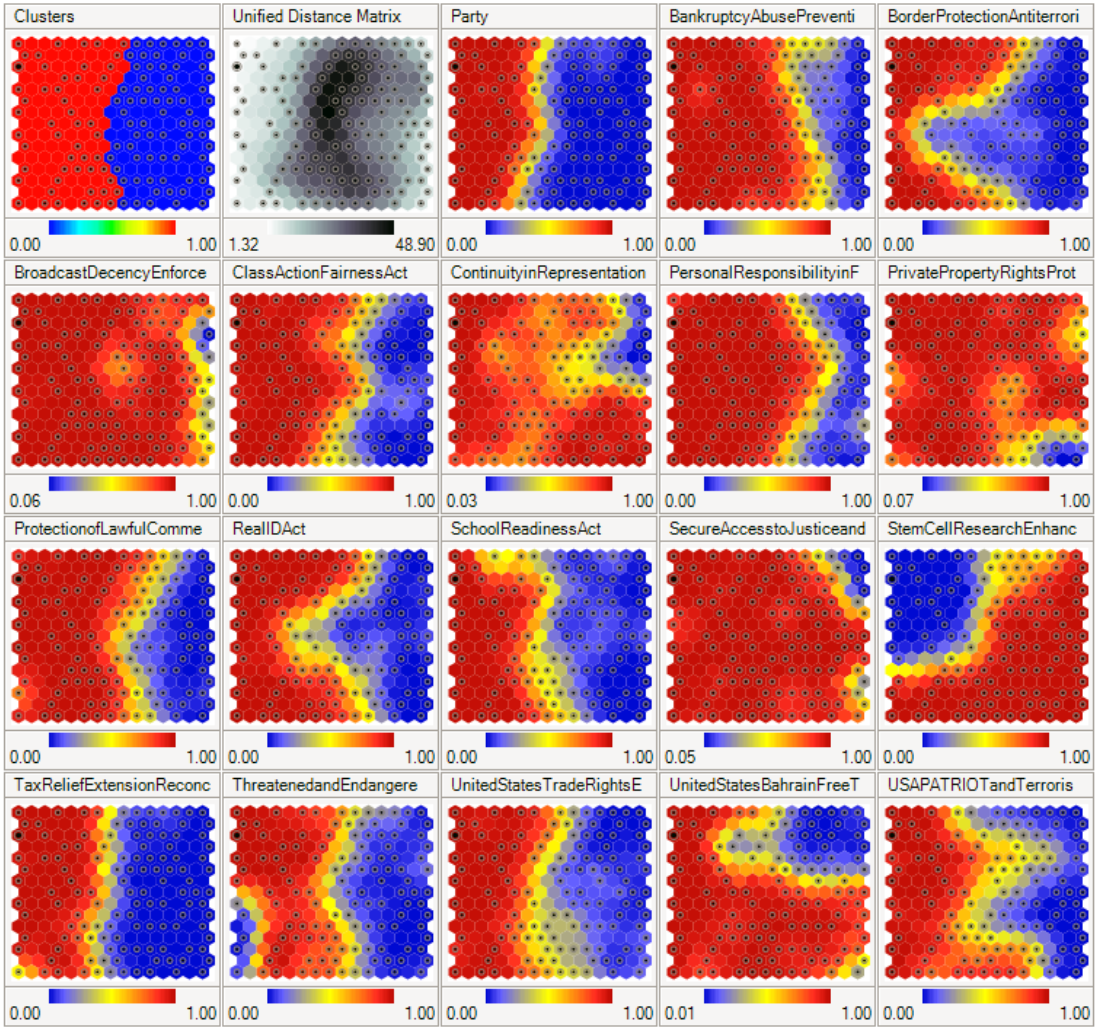


Klasifikácie pomocou SOM

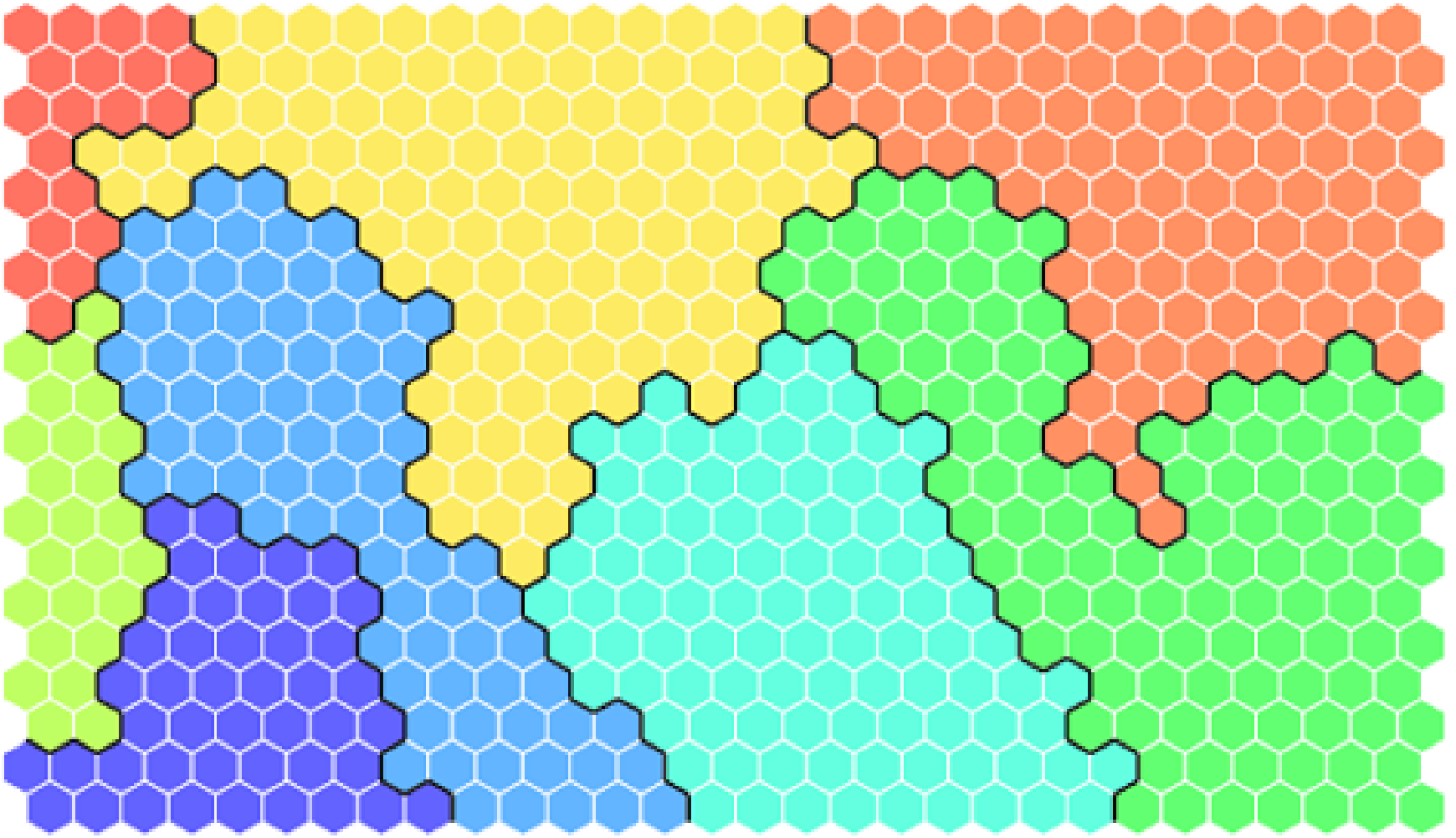
- SOM funguje v dvoch režimoch:
  - Učiaci (trénovací)
  - Klasifikačný
- V klasifikačnom režime zaradí neznámy vektor do klasifikačnej triedy, podľa toho, kde sa v príznakovom priestore nachádza
- Môže zobraziť aj vzťahy príznakov ku triedam

Klasifikácie pomocou SOM II

- Hlasovanie v kongrese USA -rozdelenie do zhlukov, U-matica a jednotlivé príznaky



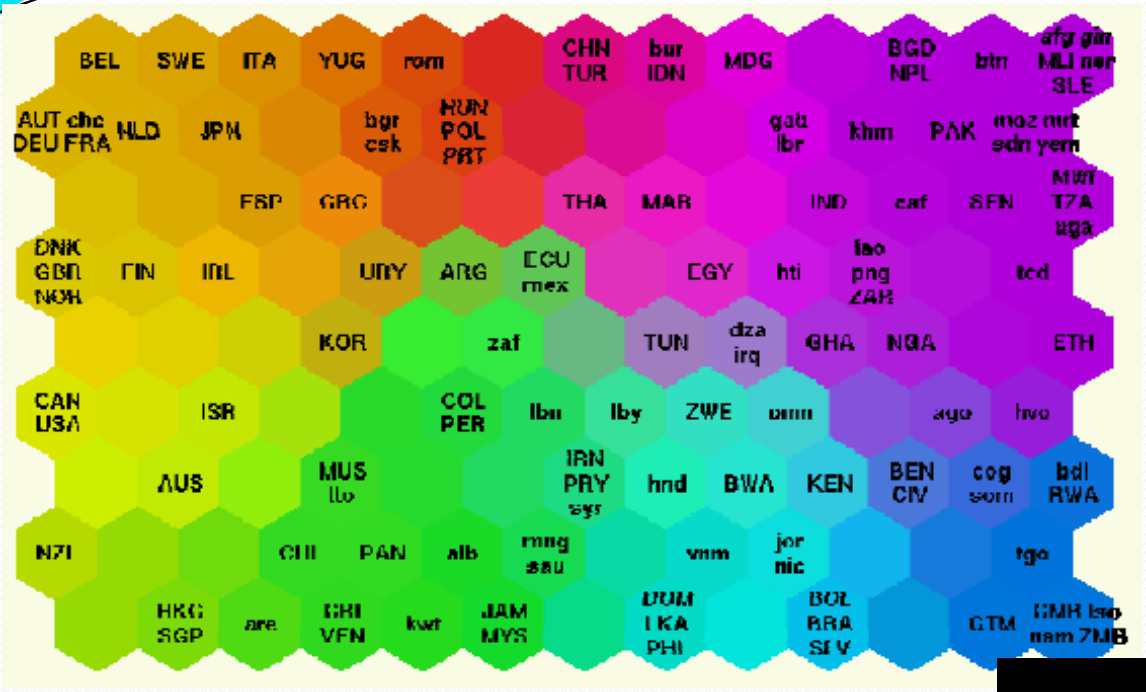
Ukážka klasifikácie SOM

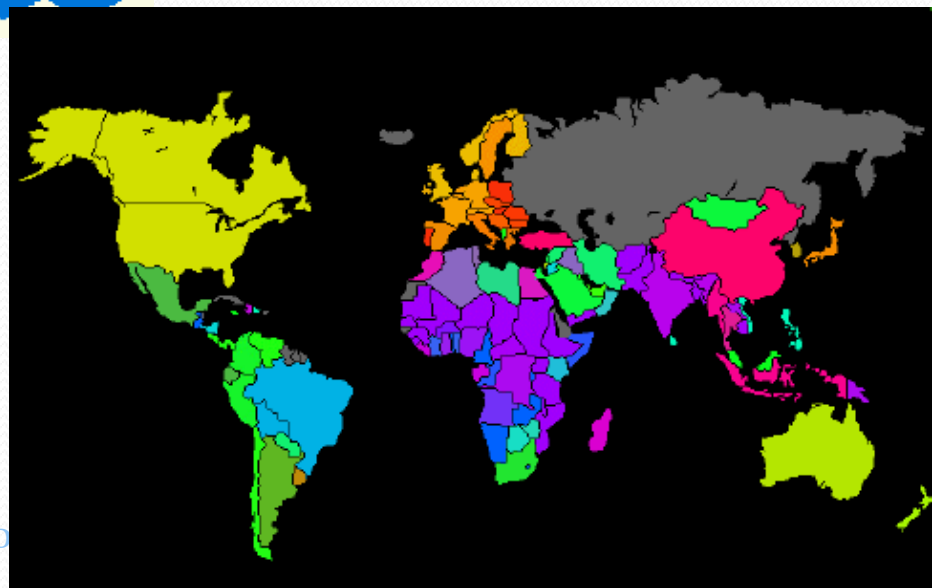


Ukážka klasifikácie SOM II

- **SOM – mapa chudoby**
  - Príklad z Helsinskej University of Technology, dáta z World Bank statistics of countries 1992
  - 39 rozmerný vektor príznakov popisujúci kvalitu života v krajinách – zdravie, výživa, vzdelanie, ...
  - Krajiny s podobnými hodnotami budú reprezentované blízkymi neurónmi

Ukážka klasifikácie SOM III





Ukážka redukcie rozmeru

- SOM možno vnímať aj ako nelineárne zovšeobecnenie PCA
- Ukážka jednorozmernej SOM vs. PCA