

# Typy umělých neuronových sítí

---

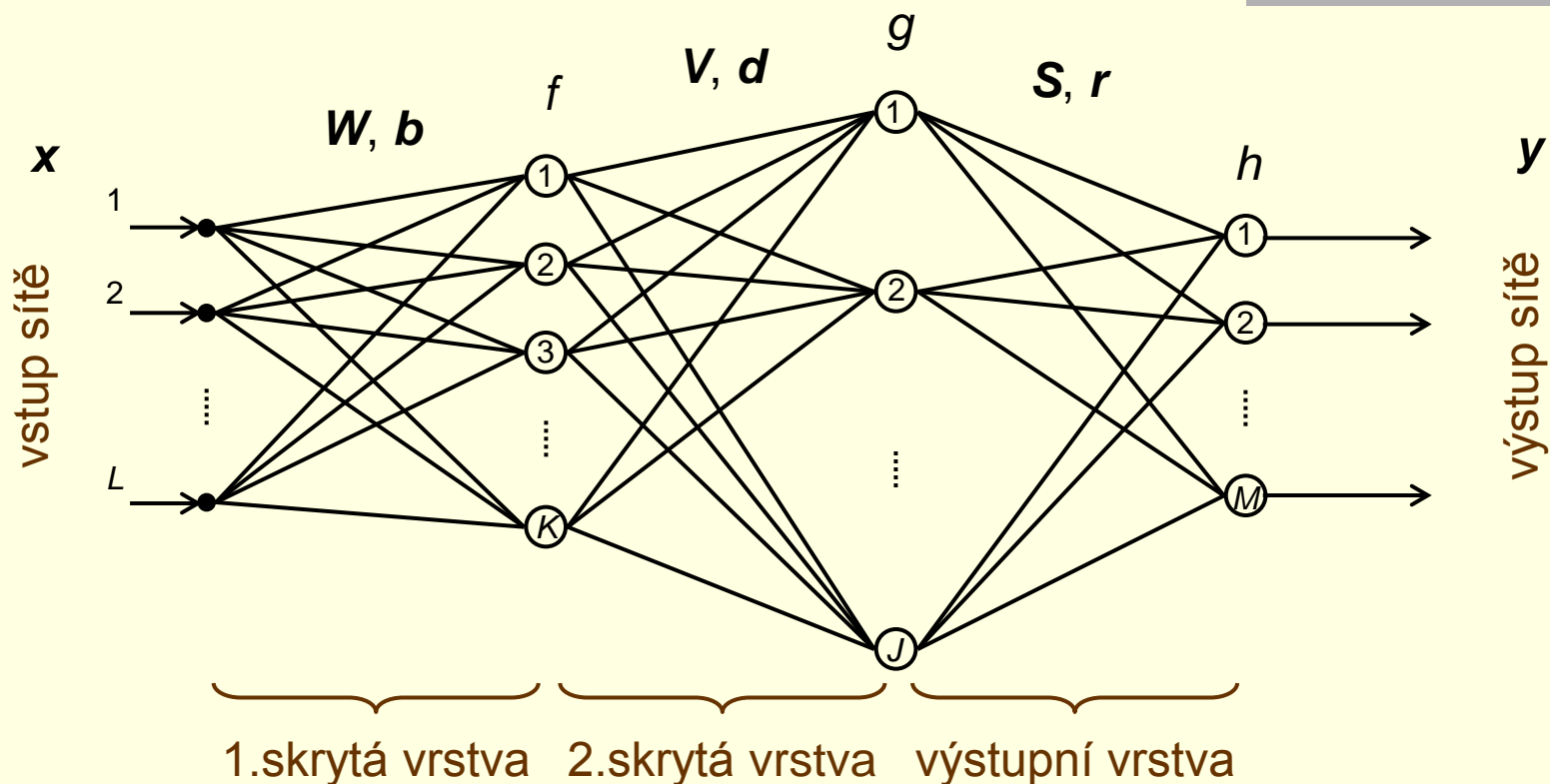
- umělá neuronová síť vznikne spojením jednotlivých modelů neuronů
- výsledná funkce sítě je určena způsobem propojení jednotlivých neuronů (tzv. topologií sítě), váhami těchto spojení a způsobem činnosti jednotlivých neuronů (tj. aktivační funkcí)
- základními typy jsou tzv. **vícevrstvé dopředné neuronové sítě** a **neuronové sítě se zpětnou vazbou**

# Typy umělých neuronových sítí (pokračování)

---

- **vícevrstvé dopředné sítě** (sítě s dopředným šířením, feedforward networks) – výstup jedné vrstvy je připojen na vstup následující vrstvy a signál se šíří pouze ze vstupu sítě na její výstup
- **sítě se zpětnou vazbou** (rekurentní sítě, feedback networks) – oproti dopředným sítím se zde signál šíří také z výstupu sítě zpět na její vstup

# Dopředné neuronové sítě



$$\text{výstup } m\text{-tého výstupního neuronu: } y_m = h \left( \sum_{j=1}^J s_{mj} g \left( \sum_{k=1}^K v_{jk} f \left( \sum_{l=1}^L w_{kl} x_l + b_k \right) + d_j \right) + r_m \right)$$

# Dopředné neuronové sítě (pokr.)

## ■ označení

$\mathbf{x} = [x_1, x_2, \dots, x_L]^T$  ... vstupní vektor sítě (též se nazývá vstupní obraz sítě)

$L$  ... počet vstupů sítě

$\mathbf{y} = [y_1, y_2, \dots, y_M]^T$  ... výstupní vektor sítě (též se nazývá výstupní obraz sítě)

$M$  ... počet výstupů sítě, tj. počet neuronů v poslední vrstvě

$\mathbf{w}_k = [w_{k1}, w_{k2}, \dots, w_{kL}]^T$  ... váhový vektor  $k$ -tého neuronu 1. vrstvy (obsahuje váhy vazeb vedoucích ze vstupu ke  $k$ -tému neuronu 1. vrstvy)

$w_{kl}$  ... váha vazby vedoucí od  $l$ -tého vstupu ke  $k$ -tému neuronu (první index určuje cíl vazby, druhý zdroj vazby)

$K$  ... počet neuronů v 1. skryté vrstvě

$J$  ... počet neuronů ve 2. skryté vrstvě

$\mathbf{W}$  ... váhová matice 1. skryté vrstvy (dimenze  $K \times L$ )

$\mathbf{V}$  ... váhová matice 2. skryté vrstvy (dimenze  $J \times K$ )

$\mathbf{S}$  ... váhová matice výstupní vrstvy (dimenze  $M \times J$ )

$\mathbf{b} = [b_1, b_2, \dots, b_K]^T$  ... prahový vektor 1. skryté vrstvy

$\mathbf{d} = [d_1, d_2, \dots, d_J]^T$  ... prahový vektor 2. skryté vrstvy

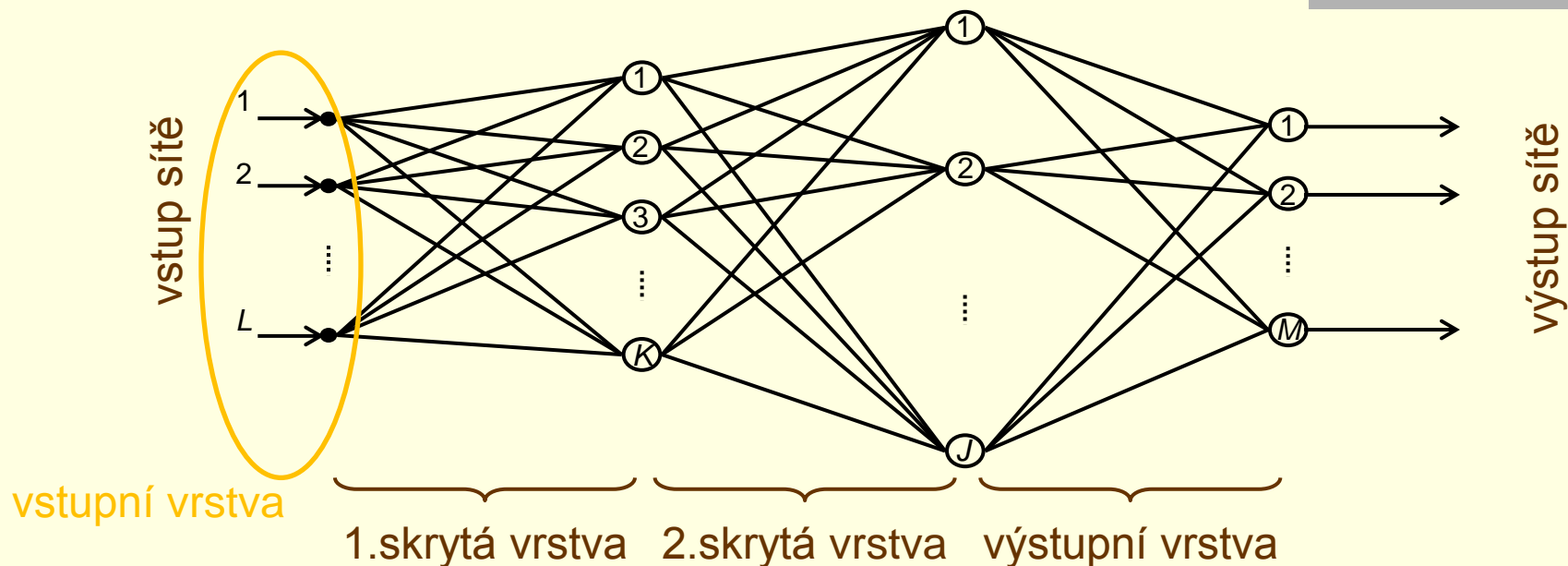
$\mathbf{r} = [r_1, r_2, \dots, r_M]^T$  ... prahový vektor výstupní vrstvy

$f$  ... aktivační funkce neuronů 1. skryté vrstvy

$g$  ... aktivační funkce neuronů 2. skryté vrstvy

$h$  ... aktivační funkce neuronů výstupní vrstvy

# Dopředné neuronové sítě



Kolikavrstvá síť je na obrázku?

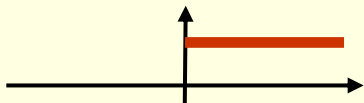
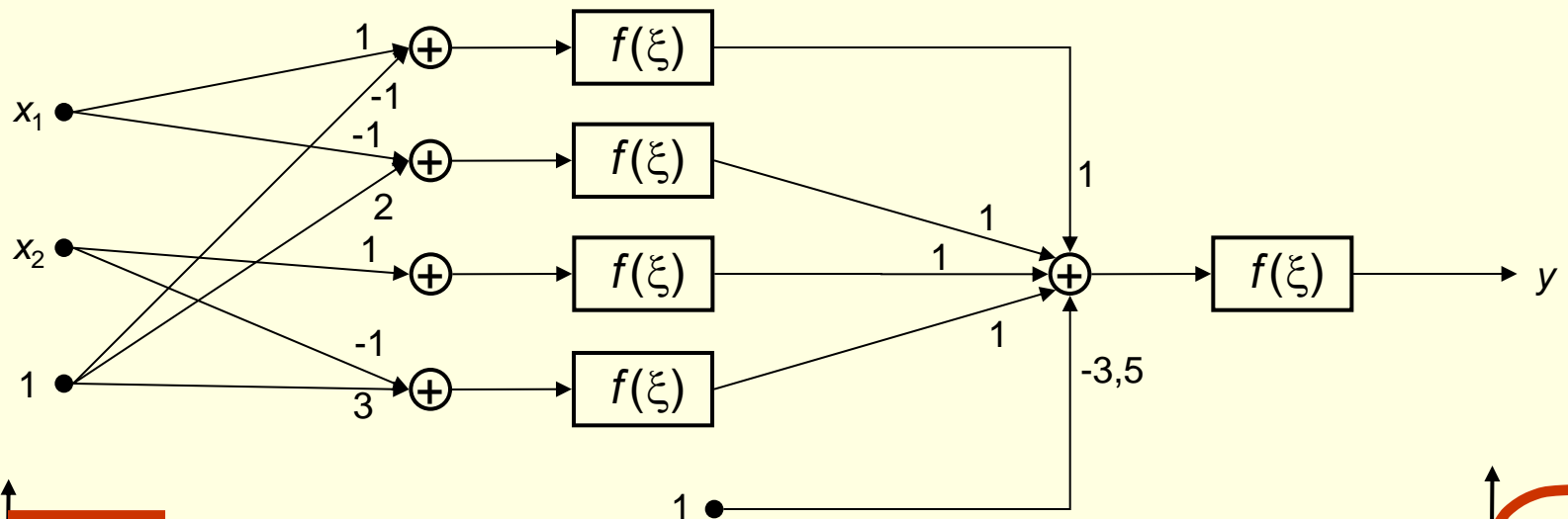
Vrstvy, ve kterých se počítá výstup neuronů, jsou 3, proto se síť označuje jako **třívrstvá síť**.

Topologie sítě se zkráceně zapisuje výrazem  $L-K-J-M$ , proto se síť označuje jako **čtyřvrstvá síť** (vstup se považuje za vrstvu).

Síť je možné též označit jako **síť se dvěma skrytými vrstvami**.

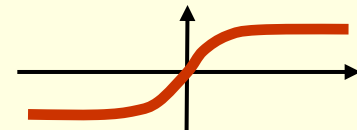
# Činnost dopředných neuronových sítí

- Po předložení vstupu je vygenerován odpovídající výstup.
  - Příklad: Analyzujte chování dvouvrstvé neuronové sítě znázorněné na obrázku (úkolem je zjistit závislost  $y$  na  $x_1$  a  $x_2$ ).



a)  $f(\xi) = \text{sgn}(\xi) = \begin{cases} +1 & \text{pro } \xi \geq 0 \\ -1 & \text{pro } \xi < 0 \end{cases}$

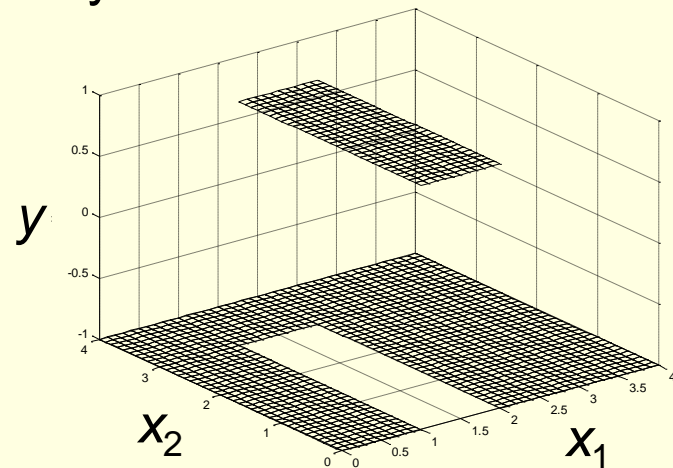
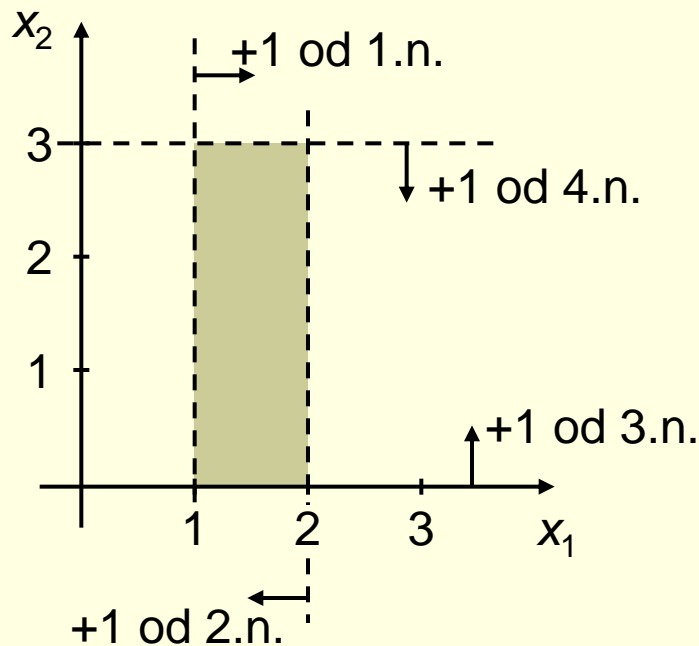
b)  $f(\xi) = \frac{2}{1 + \exp(-\lambda \xi)} - 1$



# Činnost dopředných neuronových sítí (pokračování)

## ■ Řešení: ad a)

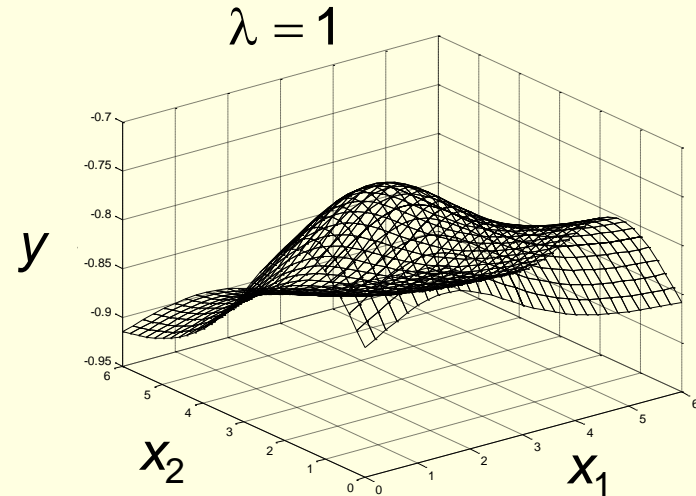
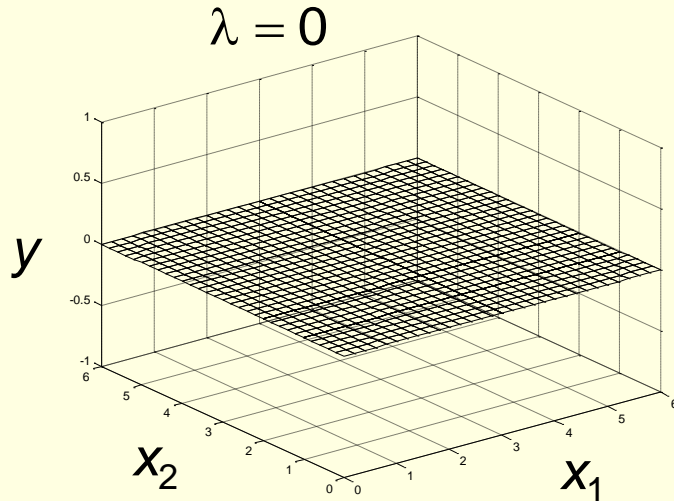
- výstup každého neuronu v 1.vrstvě může nabývat pouze hodnot +1 nebo -1, tzn. každý neuron rozdělí vstupní rovinu  $x_1x_2$  na 2 poloroviny. Jedné přiřadí +1 a druhé -1.
- na výstupu neuronu 2.vrstvy může být +1 pouze v případě, že na výstupech všech neuronů 1.vrstvy je +1, tzn. uvnitř vyznačeného obdélníku



# Činnost dopředných neuronových sítí (pokračování)

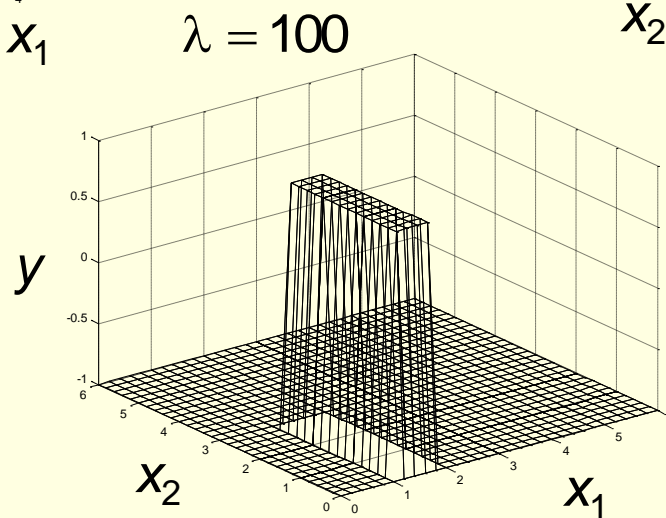
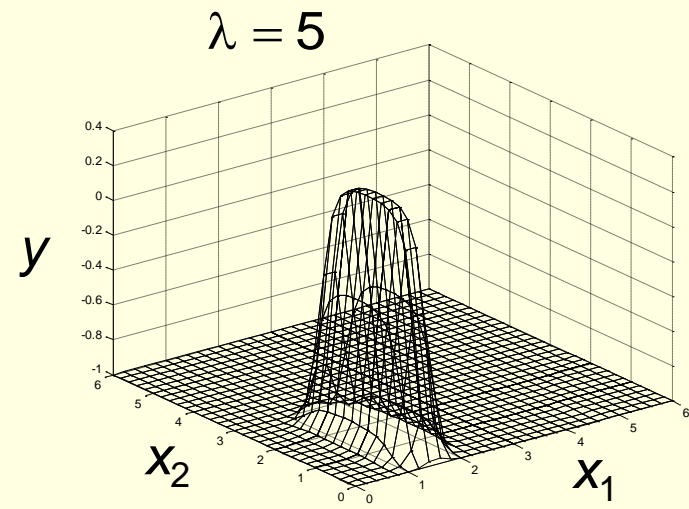
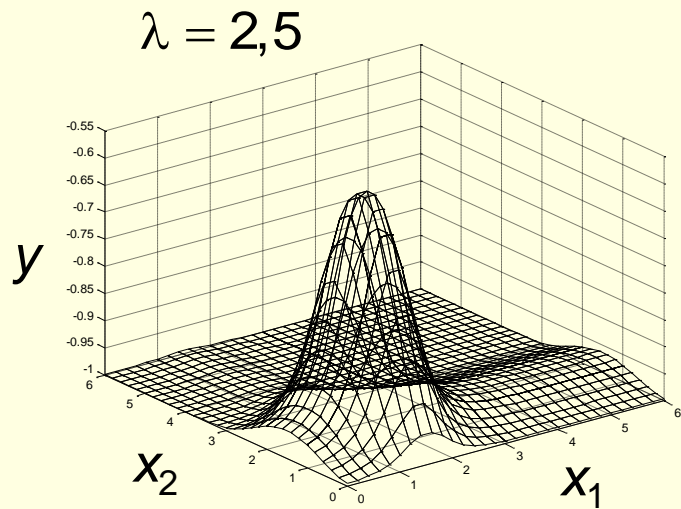
## ■ Řešení: ad b)

- Každý neuron 1.vrstvy opět rozdělí vstupní rovinu na 2 části. Jedna část odpovídá kladným hodnotám na výstupu příslušného neuronu a druhá záporným. Přejít mezi polorovinami však není skokový, nýbrž plynulý. Neuron 2.vrstvy opět zkombinuje výstupy 1.vrstvy.



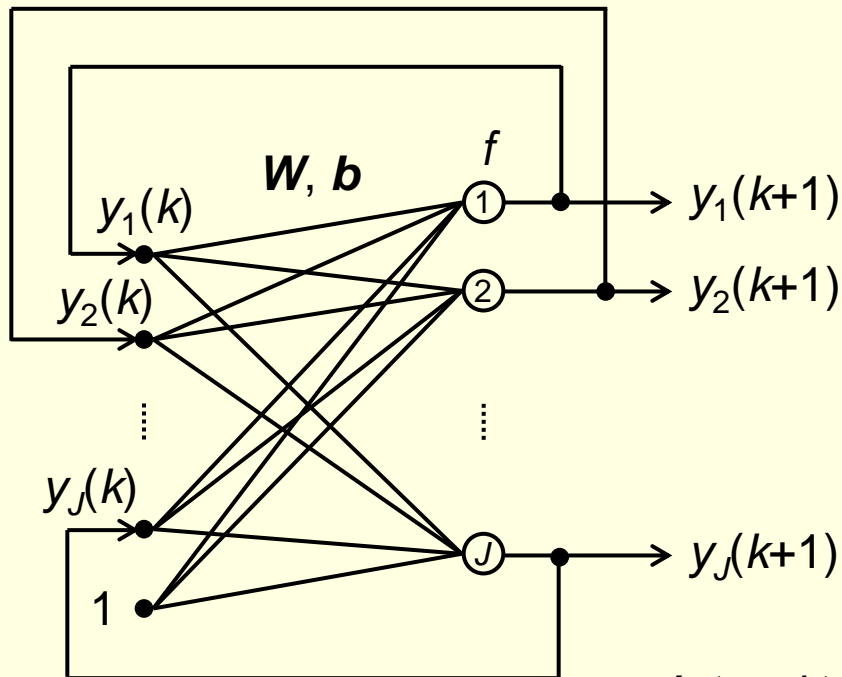


# Činnost dopředných neuronových sítí (pokračování)



# Neuronové sítě se zpětnou vazbou (rekurentní sítě, feedback networks)

- jednovrstvá síť se zpětnou vazbou, počet vstupů = počet výstupů



$W$  ... váhová matice sítě (řádu  $J$ )

$\mathbf{b} = [b_1, b_2, \dots, b_J]^T$  ... prahový vektor sítě

$J$  ... počet vstupů a výstupů sítě

$f$  ... aktivační funkce

$\mathbf{y} = [y_1, y_2, \dots, y_J]^T$  ... výstupní vektor (stav) sítě

$k$  ... čas

$\mathbf{y}(0)$  ... inicializační stav sítě

$$\text{výstup } j\text{-tého neuronu: } y_j(k+1) = f\left(\sum_{i=1}^J w_{ji} y_i(k) + b_j\right)$$

# Činnost neuronových sítí se zpětnou vazbou

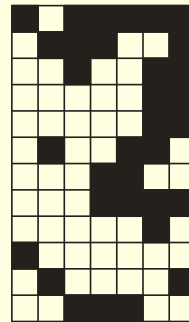
---

- po inicializaci v čase  $k=0$  se samovolně mění výstup sítě – síť přechází z jednoho stavu do jiného
- ke změně stavu může docházet buď **synchronně** (výstupy všech neuronů se mění najednou) nebo **asynchronně** (výstupy neuronů se mění postupně)

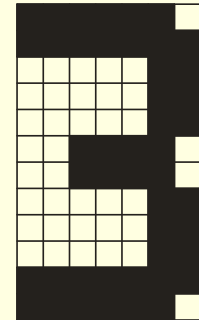
# Činnost neuronových sítí se zpětnou vazbou (pokračování)

- proces samovolné změny stavu končí v okamžiku, kdy síť dosáhne **rovnovážného stavu**, kdy se výstup sítě již nemění, tj.  $y(k+1)=y(k)$

vstup  
sítě



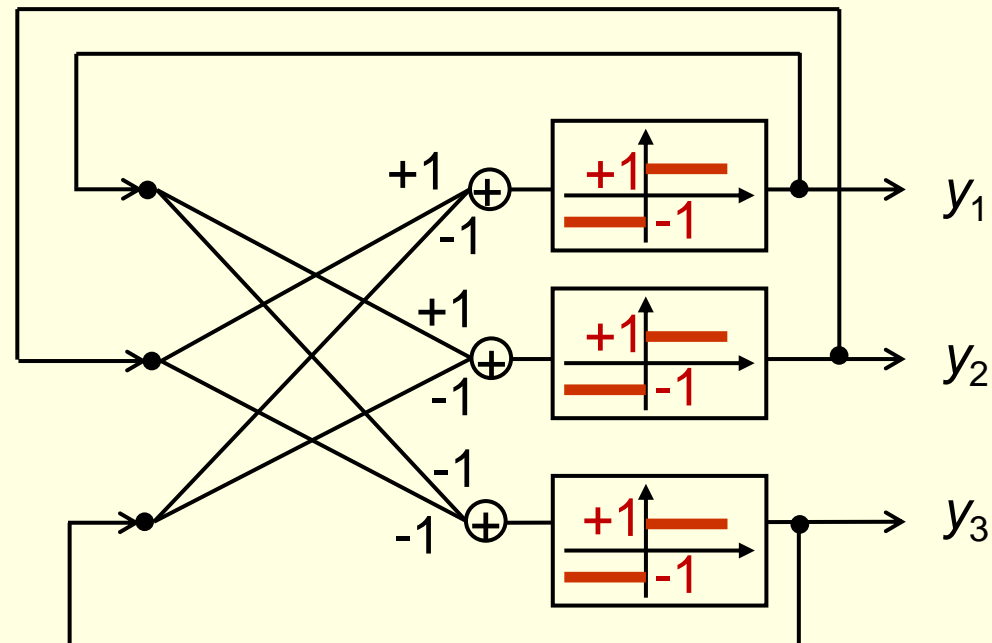
výstup  
sítě



- síť také může skončit v **rovnovážném cyklu**, který je tvořen periodicky se opakujícími stavy

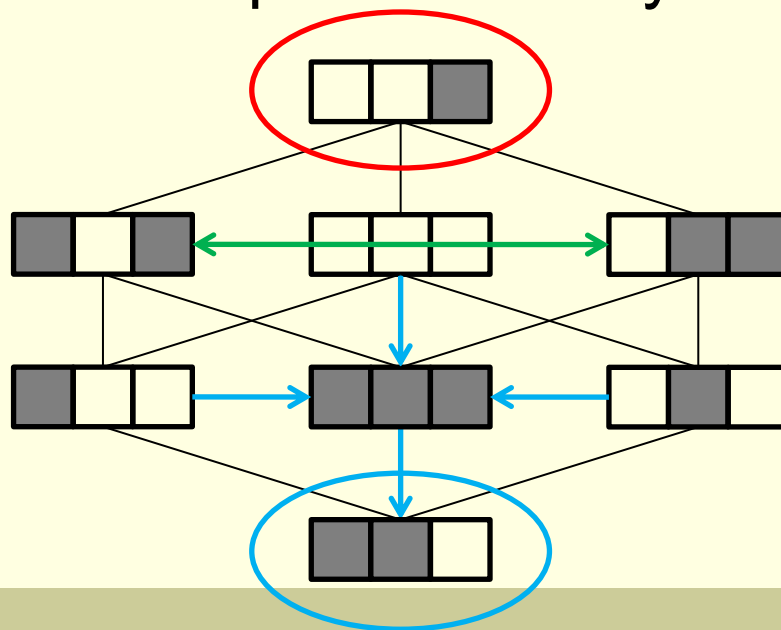
# Činnost neuronových sítí se zpětnou vazbou – příklad

- Analyzujte chování Hopfieldovy rekurentní sítě znázorněné na obrázku (úkolem je určit rovnovážné stavy, popř. rovnovážné cykly). Síť pracuje v synchronním režimu.



# Činnost neuronových sítí se zpětnou vazbou – příklad (pokr.)

- Řešení: Po inicializaci všemi možnými stavy zjistíme, že síť má rovnovážné stavy  $[+1 +1 -1]$  a  $[-1 -1 +1]$  a rovnovážný cyklus délky 2 tvořený stavy  $[-1 +1 +1]$  a  $[+1 -1 +1]$ , mezi kterými síť kmitá.
- Znázornění pomocí bitových map na jednotkové krychli:



prázdné políčko -1  
vyplněné políčko +1

rovnovážný stav  $[-1 -1 +1]$

rovnovážný stav  $[+1 +1 -1]$

rovnovážný cyklus  
 $[-1 +1 +1]$  a  $[+1 -1 +1]$

# Fáze činnosti neuronových sítí

---

## ■ Fáze nastavování

- cílem fáze nastavování je nastavit váhy a prahy sítě tak, aby prováděla požadovanou činnost (předpokládá se, že je dána tzv. **topologie** sítě, tzn. počet vstupů a výstupů sítě, počet vrstev a neuronů v nich, způsob propojení neuronů a že je dána aktivační funkce neuronů)

## ■ Fáze pracovní

- ve fázi pracovní síť reaguje na předložené vstupy podle svého předchozího nastavení

# Fáze nastavování

---

- cílem je nastavit váhy a prahy jednotlivých neuronů tak, aby síť prováděla požadovanou činnost
- možnosti nastavení vah a prahů
  - výpočtem (lze jen u některých typů sítí)
  - učením (trénováním)
    - učení s učitelem (supervised learning)
    - učení bez učitele (unsupervised learning)



# Učení s učitelem

---

- předpokládáme, že máme k dispozici tzv. **trénovací množinu**, tj. množinu dvojic [vstup, požadovaný výstup]
- neuronové sítě jsou postupně předkládány dvojice [vstup, požadovaný výstup] z trénovací množiny
- pro každý předložený vstup neuronová síť vygeneruje skutečný výstup, který se porovná s požadovaným výstupem, a trénovací algoritmus upraví nastavení vah a prahů tak, aby rozdíl mezi skutečným a požadovaným výstupem byl minimální

# Učení s učitelem (pokračování)

---

## ■ Trénovací množina:

[vstup 1, požadovaný výstup 1]

[vstup 2, požadovaný výstup 2]

[vstup 3, požadovaný výstup 3]

[vstup 4, požadovaný výstup 4]

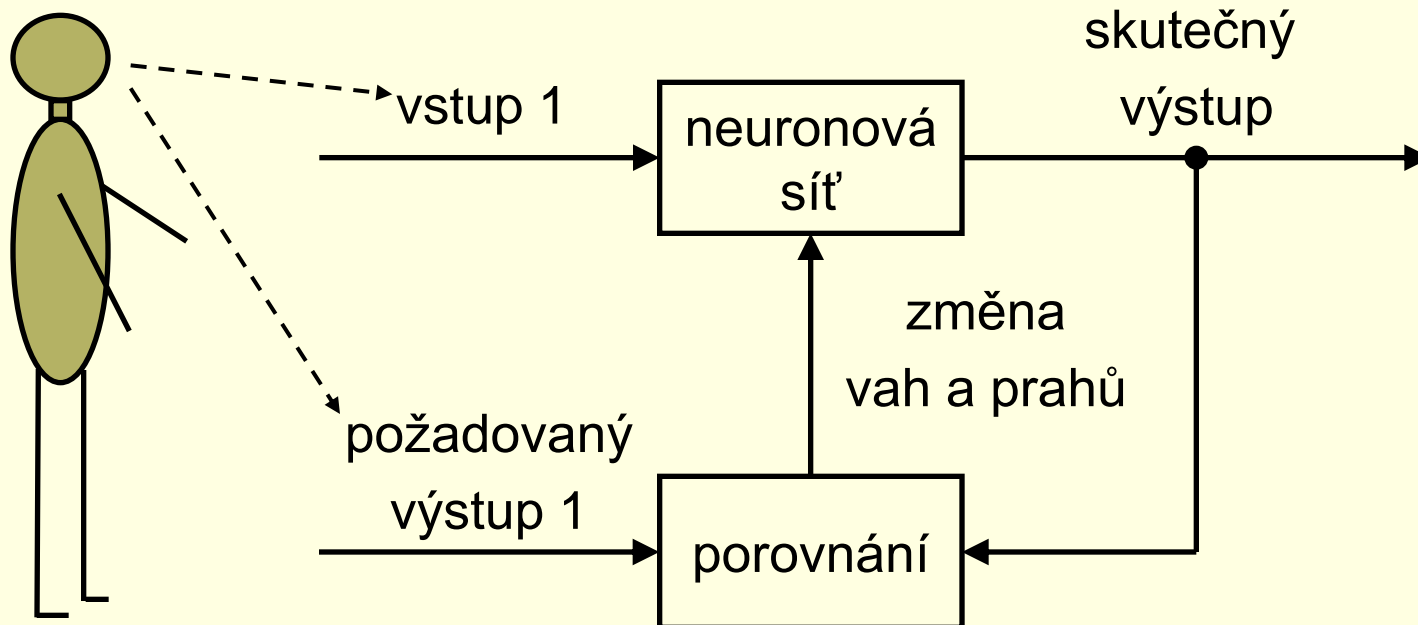
⋮

[vstup  $P$ , požadovaný výstup  $P$ ]

# Učení s učitelem (pokračování)

[ vstup 1, požadovaný výstup 1 ]

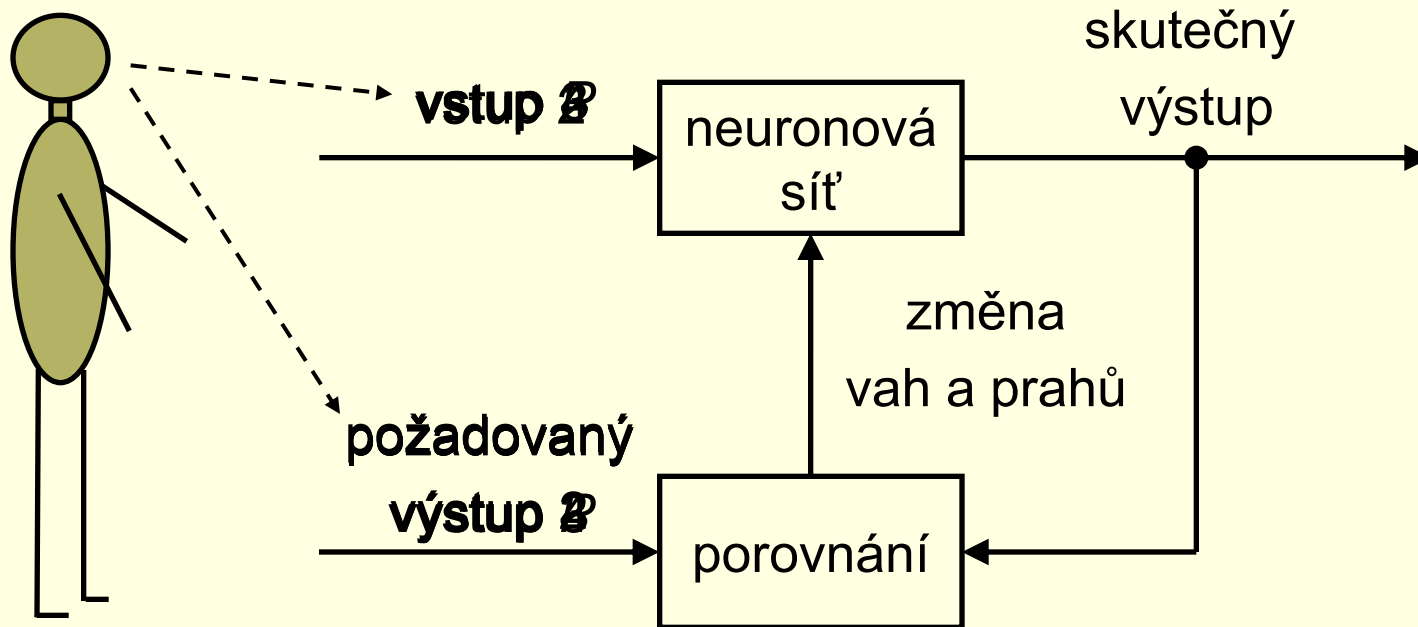
učitel



# Učení s učitelem (pokračování)

[vstup 2, požadovaný výstup 2]

učitel

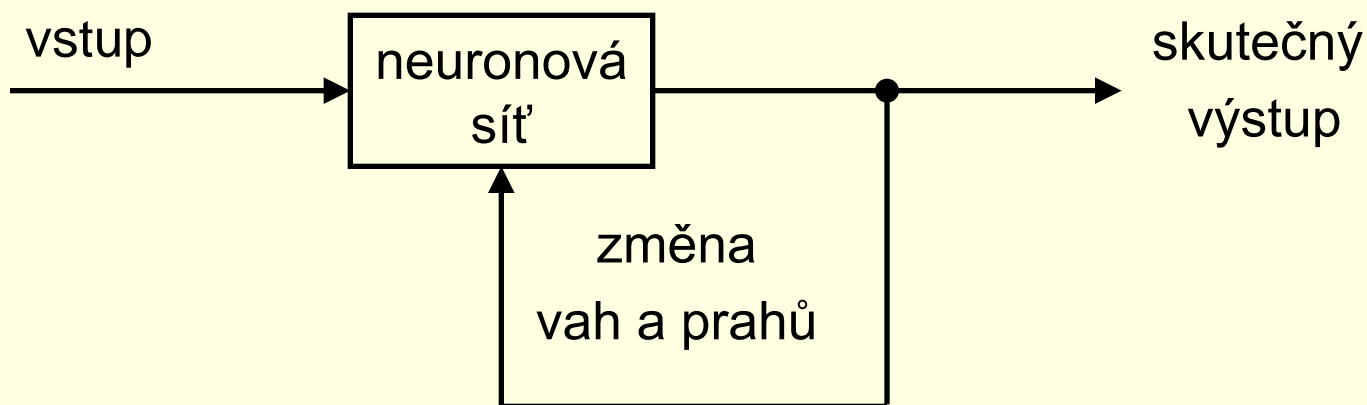


# Učení s učitelem (pokračování)

- Dvojice z trénovací množiny se přivádějí na síť opakovaně (opakování matka moudrosti)
- Trénování probíhá v tzv. **trénovacích cyklech**, kdy během jednoho trénovacího cyklu jsou síti všechny dvojice z trénovací množiny předloženy právě jednou
- Pořadí výběru trénovacích dvojic z trénovací množiny má vliv na průběh učení (doporučuje se vybírat trénovací dvojice z trénovací množiny náhodně)
- Podle chyby na výstupu sítě se nastavují váhy a prahy předchozích vrstev, proto se algoritmus učení s učitelem nazývá **algoritmus backpropagation** (algoritmus zpětného šíření chyby)
- Dobře natrénovaná síť je schopna to, co se naučila, zobecňovat (schopnost **generalizace**)
- Trénování může být **dávkové** (váhy a prahy se mění až po skončení celého trénovacího cyklu) nebo **sekvenční** (váhy a prahy se mění po každé trénovací dvojici)

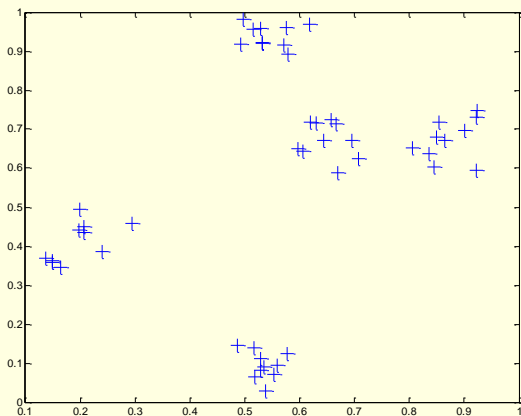
# Učení bez učitele

- Sítí jsou předkládány pouze vstupy, informace učitele (tj. požadovaný výstup) chybí. Síť se sama snaží najít zákonitosti ve vstupních datech a nastavit své váhy a prahy tak, aby na podobné vstupy reagovala podobnými výstupy.



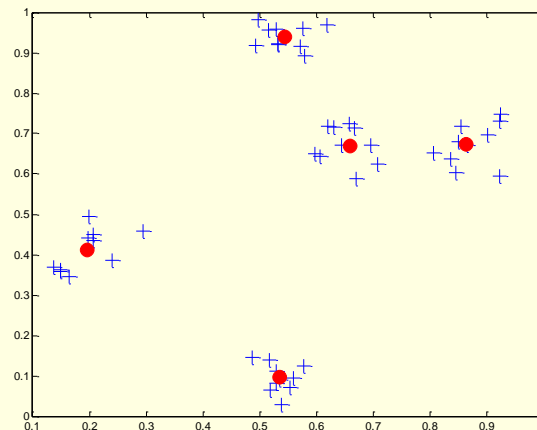
# Učení bez učitele (pokračování)

- Podobnost je obvykle definována eukleidovskou vzdáleností. Pak dochází k tzv. **shlukování** vstupních dat.



vstupní vektory jsou zobrazeny jako body v prostoru, jehož dimenze je dána počtem vstupů

- Jednotlivé shluky je pak možné reprezentovat jejich **centroidy**



# Oblasti použití neuronových sítí

---

- při řešení problémů, kde není znám jednoznačný algoritmus řešení, ale kde existuje dost velká množina příkladů, jejichž řešení známe
  - rozpoznávání složitých signálů a obrazců
  - modelování funkčních závislostí
  - optimalizační úlohy
  - rekonstrukce zašuměných dat
  - ...
- neuronové sítě nejsou vhodné pro přesné výpočty, pro zpracování numerických úloh apod.



# Příklady použití neuronových sítí

---

- **Klasifikátory** - úkolem je zařadit vstupní data do skupin (tříd) podle vzájemné podobnosti
- **Aproximátory funkcí** - z několika naměřených hodnot je třeba sestavit funkční závislost (např. predikce počtu slunečních skvrn, vývoj kursu koruny apod.)
- **Asociativní paměti** - na základě předloženého vstupu je síť schopna “vybavit si” odpovídající výstup