

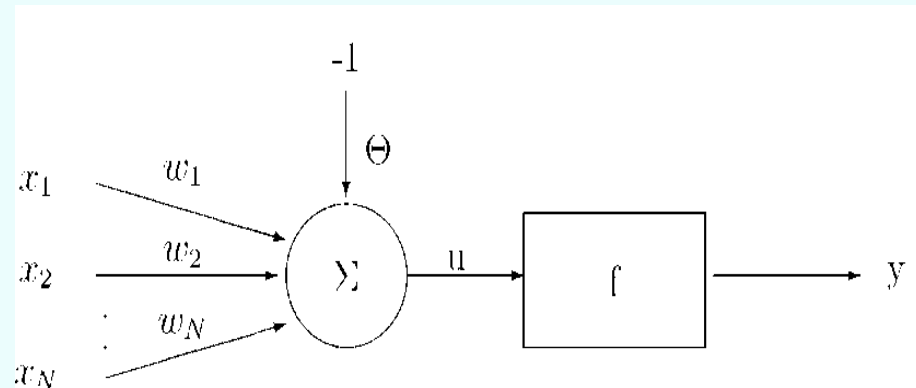
5. Umělé neuronové sítě

Umělé neuronové sítě

17. 3. 2018

5. Umělé neuronové sítě

Model umělého neuronu



$$y = f\left(\sum_{i=1}^N w_i x_i - \Theta\right)$$

y – výstup neuronu

u – vnitřní potenciál neuronu

w_i – váhy neuronu

x_i – vstupy neuronu

Θ – práh neuronu

f – neuronová aktivační funkce

5. Umělé neuronové sítě

Neuronové aktivační funkce:

sigmoidální funkce

$$f_s(\mathbf{u}) = \frac{1}{1+e^{-u}}$$

hyperbolický tangens

$$f_s(\mathbf{u}) = \mathbf{tgh}(\mathbf{u})$$

znaménková funkce

$$f_s(\mathbf{u}) = \mathbf{sgn}(\mathbf{u})$$

Heavisideova funkce

$$f_H(\mathbf{u}) = \begin{cases} \mathbf{1} & \text{pro } \mathbf{u} > 0 \\ \mathbf{0} & \text{pro } \mathbf{u} < 0 \end{cases}$$

5. Umělé neuronové sítě

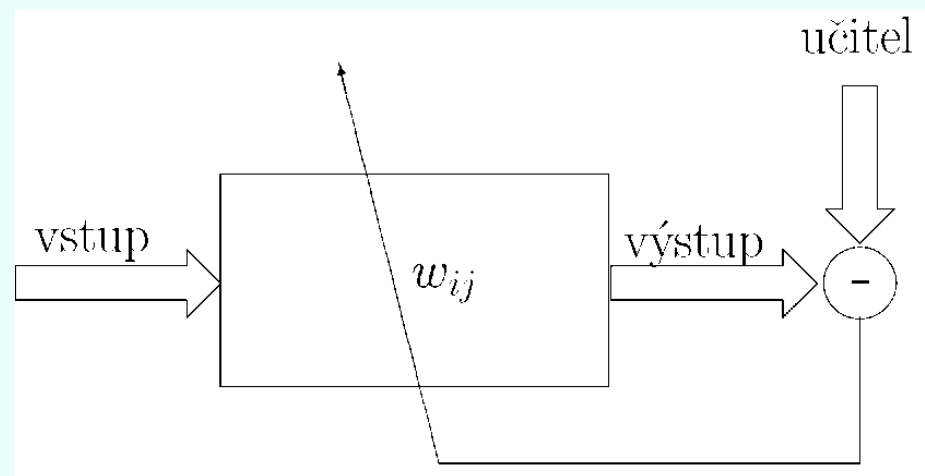
Učení neuronových sítí

Cílem učení je nastavit váhy spojení w_{ij} tak, aby síť vytvářela správnou odezvu na vstupní signál.

Základní způsoby učení:

- **učení s učitelem (supervised learning)**

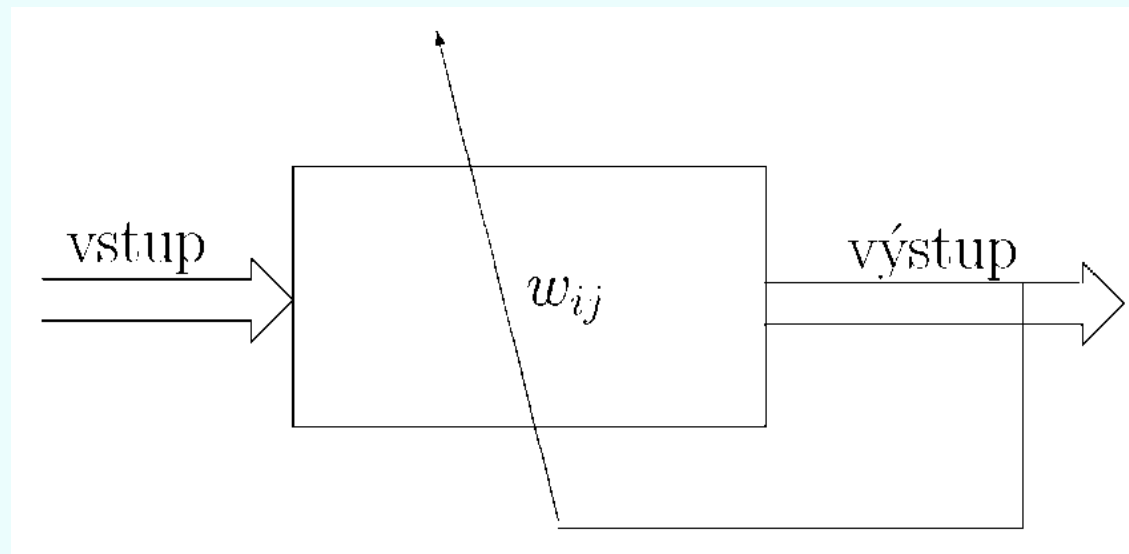
Neuronová síť se učí srovnáváním aktuálního výstupu s výstupem požadovaným (učitel) a nastavováním vah synapsí tak, aby se snížil rozdíl mezi skutečným a požadovaným výstupem.



5. Umělé neuronové sítě

- **učení bez učitele (unsupervised learning)**

Váhy spojení se nastavují tak, aby výstup sítě byl konzistentní, tj. aby síť poskytovala stejnou odezvu při stejných, popř. podobných vstupních vektorech.

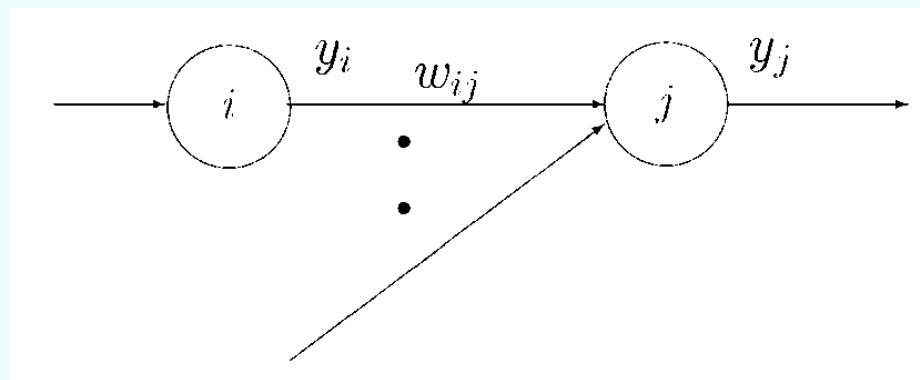


5. Umělé neuronové sítě

Základní model učení – D.O. Hebb (1949)

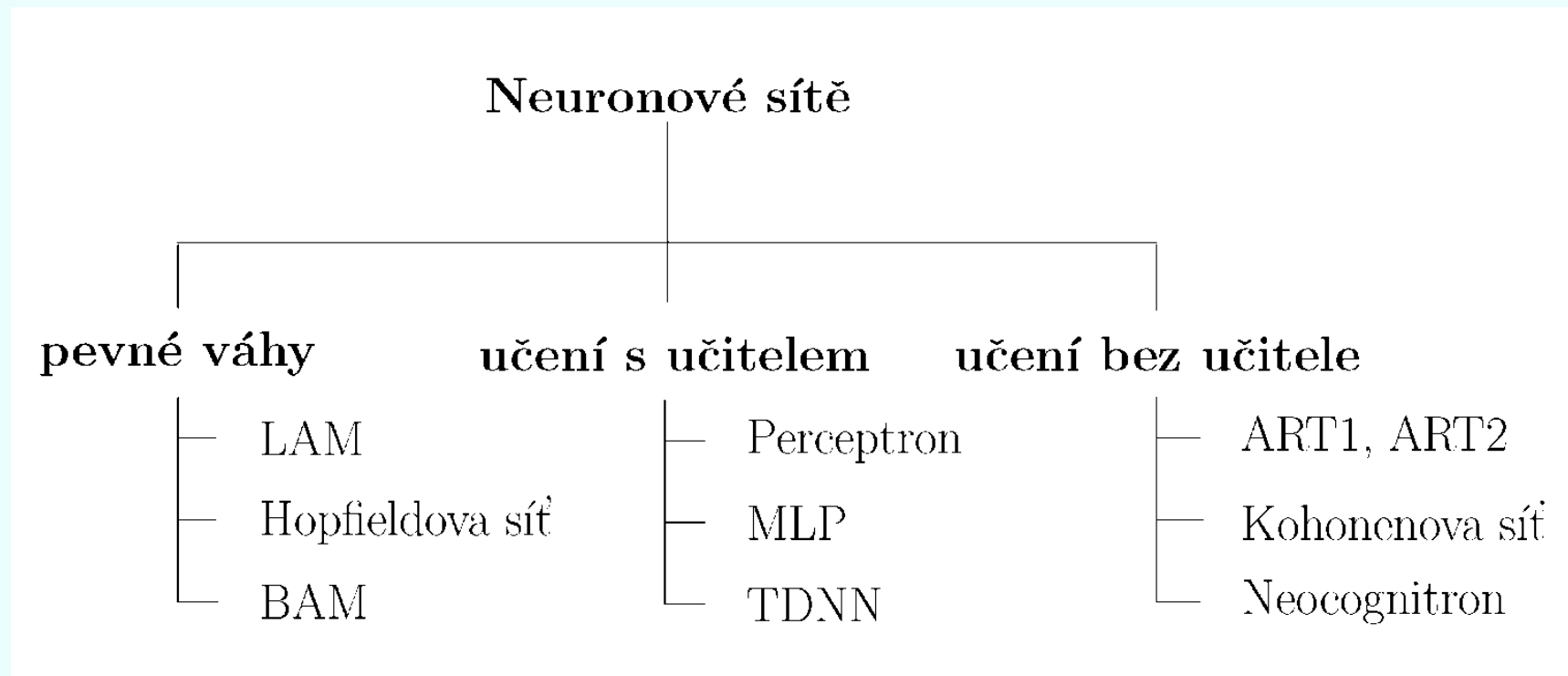
- učení bez učitele
- "síla" synaptického spojení mezi dvěma neurony se zvětší, pokud jsou oba tyto neurony aktivovány; změna hodnoty synaptického spojení je přímo úměrná součinu výstupních hodnot aktivovaných neuronů (α je tzv. koeficient učení)

$$w_{ij}(t+1) = w_{ij}(t) + \alpha y_i y_j$$

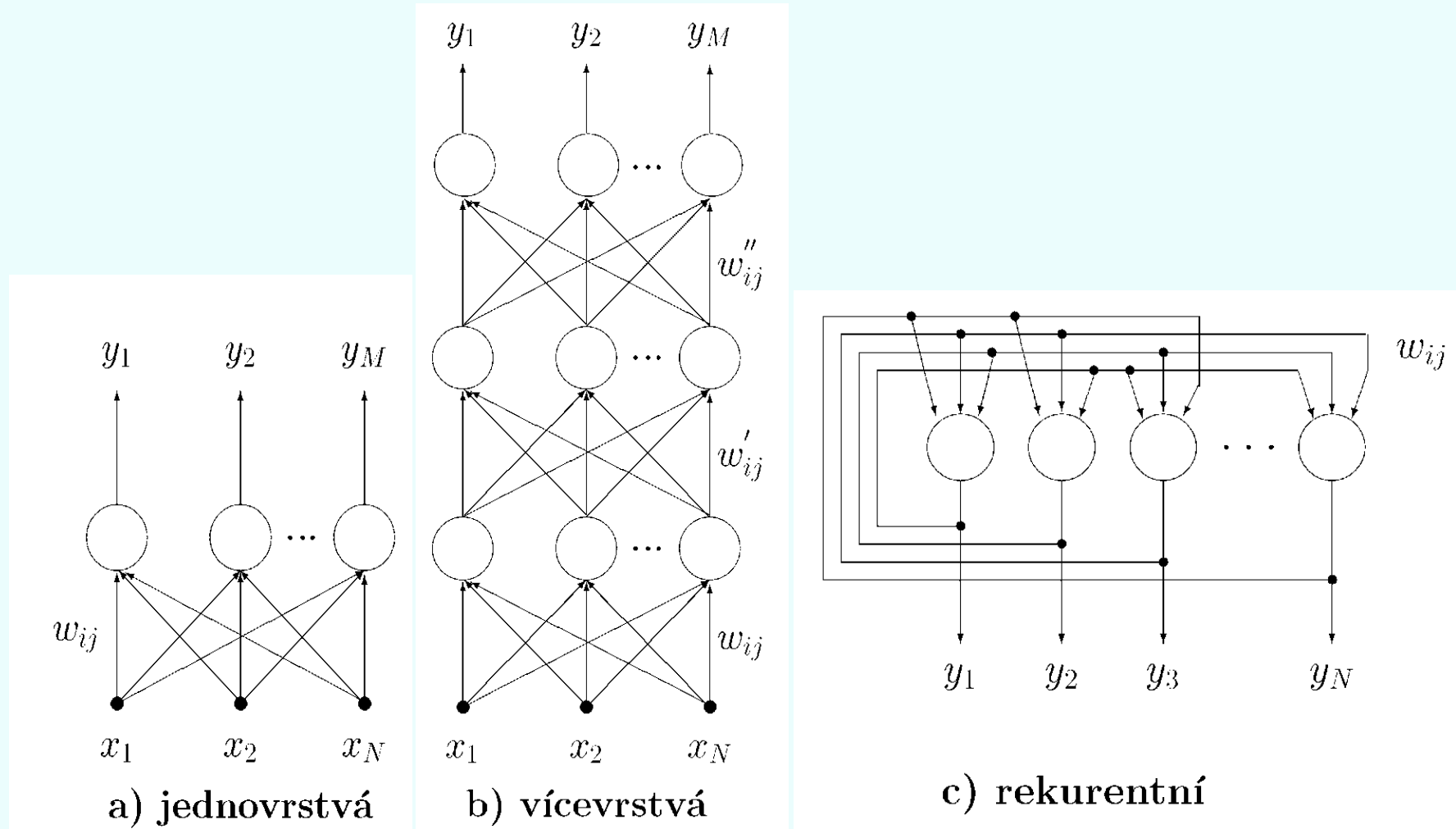


5. Umělé neuronové sítě

Typy neuronových sítí



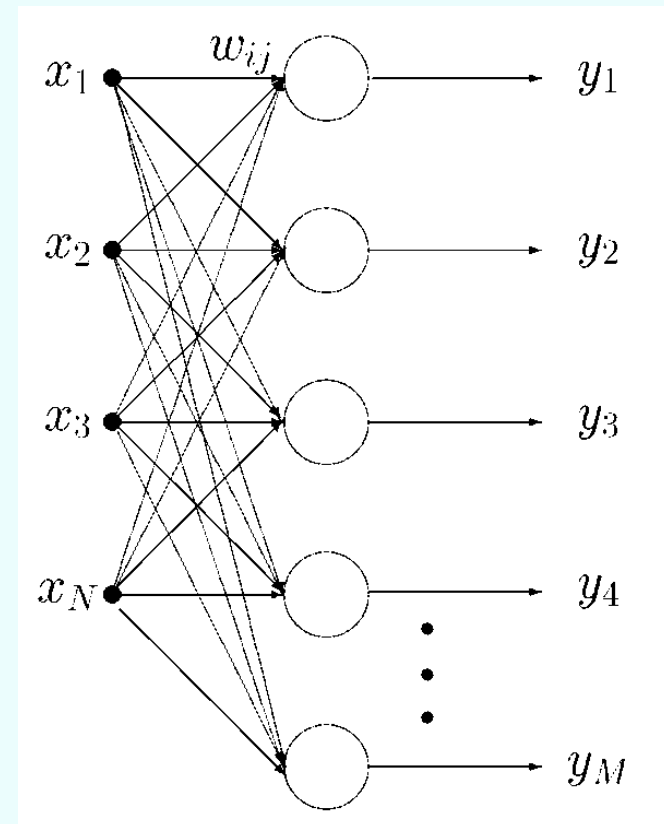
5. Umělé neuronové sítě



5. Umělé neuronové sítě

Lineární asociativní paměť (LAM)

- jednovrstvá síť s dopředným šířením a pevnými vahami
- vstupní vektory mohou být binární (0/1), popř. reálné



5. Umělé neuronové sítě

Výstupní hodnoty neuronů:

- pro reálné vstupy

$$y_i = \sum_{j=1}^N w_{ij}x_j, \text{ pro } i = 1, \dots, M$$

- pro binární vstupy

$$y_i = f_H \left(\sum_{j=1}^N w_{ij}x_j - \theta_i \right), \text{ pro } i = 1, \dots, M$$

Použití:

- hetero-/autoasociativní paměť
- rekonstrukce neúplných a šumem poškozených obrazů

5. Umělé neuronové sítě

Reálné vstupy:

Nastavení váhových koeficientů pro asociační seznam L :

$$L = \{(B^1, A^1), \dots, (B^P, A^P)\}$$

vstup: $B^k = [B_1^k, \dots, B_N^k]^T$

výstup: $A^k = [A_1^k, \dots, A_M^k]^T$

$$w_{ij} = \sum_{s=1}^P a_i^s b_j^s, \quad \text{pro } i = 1, \dots, N; j = 1, \dots, M$$

nebo maticově

$$W = [w_{ij}] = AB^T$$

kde $A = [M \times P] = [A^1, \dots, A^P]$, $B = [N \times P] = [B^1, \dots, B^P]$

5. Umělé neuronové sítě

Odezva sítě na vstupní vektor $X = [X^1, \dots, X^P]$

$$Y = WX = AB^T X = \sum_{s=1}^P (B^s X) A^s$$

tj. výstup je lineární kombinace vektorů A^s s koeficienty určenými skalárním součinem $B^s X$.

Jsou-li vstupní vektory ortonormální, je výstup shodný s obrazem uloženým v asociativní paměti.

Kapacita sítě (maximální počet uchovávaných vzorů):

$$P \leq N$$

kde P je počet vzorů uchovávaných v paměti, N je dimenze vstupního vektoru.

5. Umělé neuronové sítě

Binární vstupy:

Nastavení váhových koeficientů w_{ij} a hodnot prahů θ_i pro asociační seznam L

$$w_{ij} = (2a_i^s - 1)(2b_j^s - 1), \quad \text{pro } i = 1, \dots, N; j = 1, \dots, M$$

$$\theta_i = \frac{1}{2} \sum_{j=1}^N w_{ij}, \quad \text{pro } i = 1, \dots, M$$

Odezva sítě

$$y_i = \begin{cases} 0 & \sum_{j=1}^M (w_{ij}x_j - \theta_i) \leq 0 \\ 1 & \sum_{j=1}^M (w_{ij}x_j - \theta_i) > 0 \end{cases}$$

5. Umělé neuronové sítě

Příklad:

Asociační seznam $L = \{([1011]^T, [1011]^T), ([0101]^T, [0101]^T)\}$

$$W = (2A - \mathbf{1})(2B - \mathbf{1})^T = \begin{bmatrix} 1 & -1 \\ -1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & -2 & 2 & 0 \\ -2 & 2 & -2 & 0 \\ 2 & -2 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

$$\theta = \frac{1}{2} \sum_{j=1}^M w_{ij} = [1 \ -1 \ 1 \ 1]^T$$

5. Umělé neuronové sítě

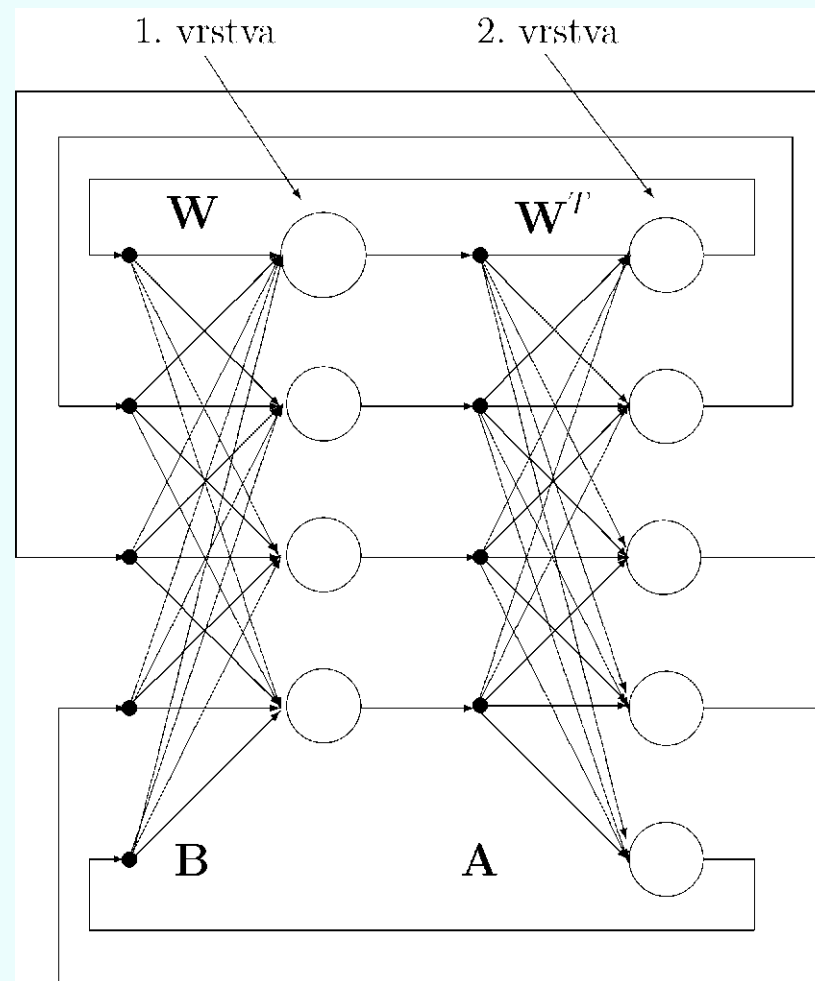
Testovací vektor $\mathbf{X} = [1101]^T$

$$\mathbf{W}\mathbf{x} - \boldsymbol{\theta} = \begin{bmatrix} 2 & -2 & 2 & 0 \\ -2 & 2 & -2 & 0 \\ 2 & -2 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix}$$

$$\mathbf{Y} = \mathbf{f}_H(\mathbf{W}\mathbf{x} - \boldsymbol{\theta}) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

5. Umělé neuronové sítě

Bidirectional Associative Memory (BAM)



5. Umělé neuronové sítě

BAM je

- dvouvrstvá rekurentní síť s pevnými vahami,
- vstupní vektory mohou být binární (0/1), popř. bipolární (-1/1)

Použití:

- heteroasociativní paměť
- rekonstrukce neúplných a šumem poškozených obrazů

5. Umělé neuronové sítě

Kapacita sítě (maximální počet uchovávaných vzorů):

Hrubý odhad: $P \leq n$, $n = \min \{N, M\}$,

kde N a M jsou počty neuronů v první a druhé vrstvě a P je maximální počet asociací.

Přesnější odhad: Pokud vybereme L vektorů tak, že

$$L < \frac{0,68n^2}{[\log_2(n) + 4]^2}$$

a každý vektor má $4 + \log_2(n)$ prvků rovných 1 a zbytek 0, pak lze vytvořit BAM, pro kterou je 98% stavů stabilních.

5. Umělé neuronové sítě

Algoritmus trénování a činnosti BAM

Krok 1: Inicializace váhových koeficientů w_{ij} a prahové hodnoty Θ :

$$1. \text{ vrstva} \quad \mathbf{W} = [w_{ij}] = \sum_{s=1}^P (2\mathbf{a}_i^s - \mathbf{1})(2\mathbf{b}_j^s - \mathbf{1})$$

$$\Theta'_i = \frac{1}{2} \sum_{j=1}^M w_{ij}, \quad i = 1, \dots, N$$

$$2. \text{ vrstva} \quad \mathbf{W}^T$$

$$\Theta''_j = \frac{1}{2} \sum_{i=1}^N w_{ij}, \quad j = 1, \dots, M$$

kde w_{ij} jsou váhové koeficienty mezi i -tým neuronem první vrstvy a j -tým neuronem druhé vrstvy, $\mathbf{a}_i^s, \mathbf{b}_i^s$ jsou i -té prvky s -tých vektorů asociačního seznamu (mohou nabývat pouze hodnot $\{0,1\}$), P je počet prvků asociačního seznamu, Θ' a Θ'' jsou vektory prahových hodnot neuronů první a druhé vrstvy (pro homogenní BAM jsou tyto vektory nulové).

5. Umělé neuronové sítě

Krok 2: Předložení neznámého vstupního vektoru $\mathbf{B} = [b_1, \dots, b_n]^T$, $\mathbf{B} \in \{0,1\}^N$

Krok 3: Iterace, dokud síť konverguje:

Během t -té iterace se pro

1. vrstvu:

1) vypočtou hodnoty $u'_j(t + 1)$ paralelně pro $i = 1, 2, \dots, M$ podle vztahu

$$u'_j(t + 1) = \sum_{i=1}^N w_{ij} b_i(t) - \theta'_j$$

2) vypočtou výstupní hodnoty $a_j(t + 1)$ pro $j = 1, 2, \dots, M$ podle vztahu

$$a_j(t + 1) = \begin{cases} 1 & \text{pro } u'_j(t + 1) > 0 \\ 0 & \text{pro } u'_j(t + 1) < 0 \\ a_j(t) & \text{pro } u'_j(t + 1) = 0 \end{cases}$$

5. Umělé neuronové sítě

2. vrstvu:

1) vypočtou hodnoty $u_j''(t + 1)$ paralelně pro $j = 1, 2, \dots, N$ podle vztahu

$$u_i''(t + 1) = \sum_{j=1}^M w_{ij} b_j(t) - \theta_i''$$

2) vypočtou výstupní hodnoty $b_i(t + 1)$ pro $j = 1, 2, \dots, N$ podle vztahu

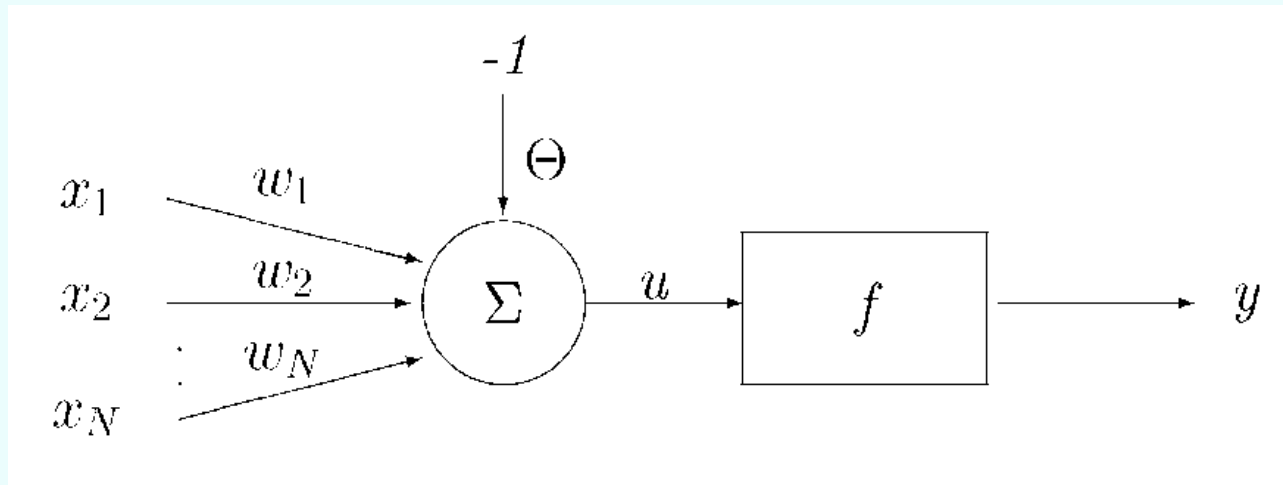
$$b_i(t + 1) = \begin{cases} \mathbf{1} & \text{pro } u_i''(t + 1) > \mathbf{0} \\ \mathbf{0} & \text{pro } u_i''(t + 1) < \mathbf{0} \\ b_i(t) & \text{pro } u_i''(t + 1) = \mathbf{0} \end{cases}$$

Tento proces se opakuje, dokud se výstup první, popř. druhé vrstvy mění. Po ukončení iterací odpovídá vstup první vrstvy BAM vzorovému vektoru z asociačního seznamu, který se nejvíce podobá neznámému vektoru předloženému na vstup, výstup pak odpovídá asociaci uložené v BAM během trénování.

Krok 4: Opakování od kroku 2 pro nový vstupní vektor.

5. Umělé neuronové sítě

Jednoduchý perceptron

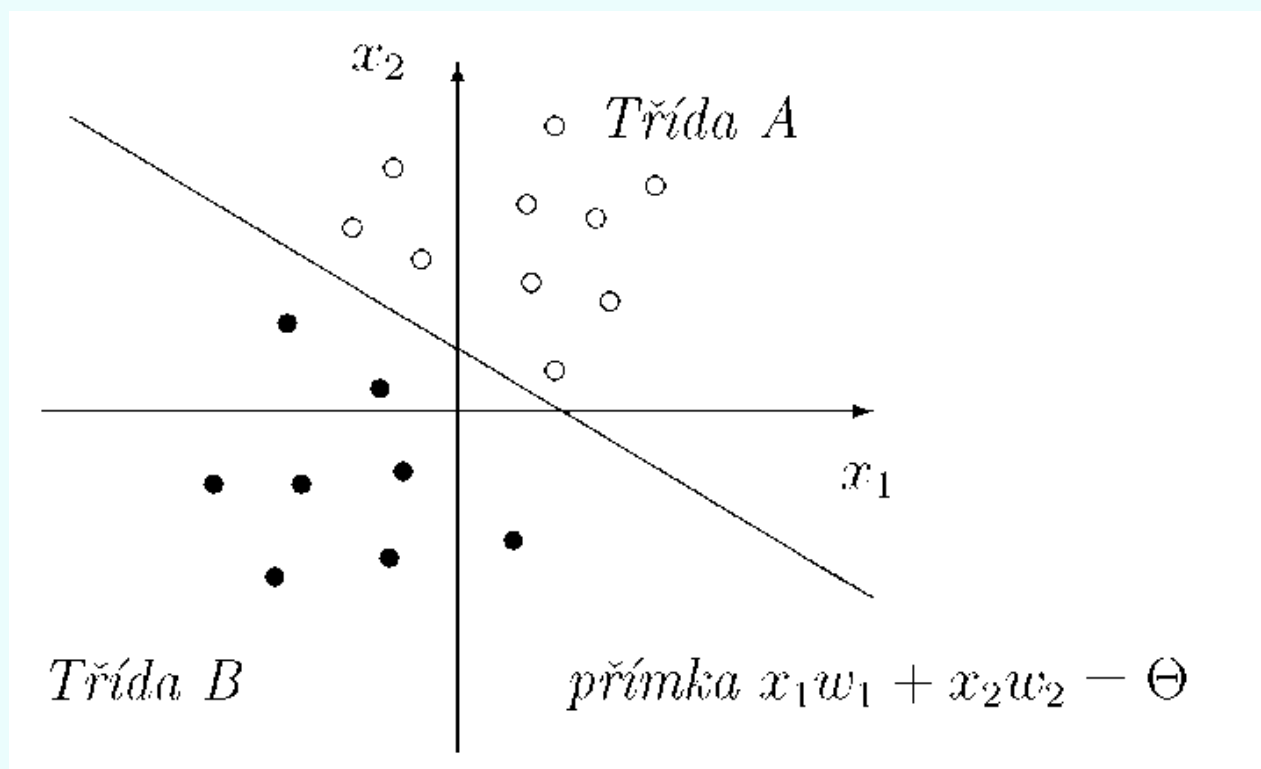


- jednoduchý neuron, popř. jednovrstvá síť s dopředným šířením
- učení: s učitelem
- neuronová aktivační funkce: znaménková funkce, popř. jednotkový skok

5. Umělé neuronové sítě

Příklad použití:

Klasifikátor pro lineárně separovatelné obrazy



5. Umělé neuronové sítě

Trénování jednoduchého perceptronu

Krok 1: Počáteční inicializace vah w_i , $1 \leq i \leq N$ a prahu θ (náhodně)

Krok 2: Přivedení vstupního vektoru a definice požadované výstupní odezvy

Krok 3: Výpočet odezvy perceptronu podle vztahu

$$y(t) = f_z \left(\sum_{i=1}^N w_i(t) x_i(t) - \theta \right), \quad 1 \leq i \leq N$$

5. Umělé neuronové sítě

Krok 4: Adaptace vah

$$\mathbf{w}_i(t + 1) = \mathbf{w}_i(t) + \boldsymbol{\eta}[\mathbf{d}(t) - \mathbf{y}(t)]\mathbf{x}_i(t), \quad 1 \leq i \leq N$$

$$d(t) = \begin{cases} +1, & \text{patří-li vstup do třídy A} \\ -1, & \text{patří-li vstup do třídy B} \end{cases}$$

kde $\mathbf{d}(t)$ je požadovaný výstup, $\boldsymbol{\eta}$ je tzv. zisk (koeficient učení), který nabývá hodnot z intervalu (0,1)

Krok 5: Opakování od kroku 2 pro všechny trénovací dvojice

Pro lineárně neseparovatelné obrazy — modifikované algoritmy:

- Kapsový algoritmus (Gallant)
- LMS algoritmus

5. Umělé neuronové sítě

Trénování perceptronu (Widrow — Hoff LMS)

Krok 1: Počáteční inicializace vah w_i a prahu θ

Krok 2: Přivedení vstupního vektoru $X = [x_1, \dots, x_N]^T$ a definice požadované výstupní odezvy d

Krok 3: Výpočet vnitřního potenciálu perceptronu podle vztahu

$$u(t) = \sum_{i=1}^N w_i(t)x_i(t) - \theta$$

Krok 4: Výpočet chyby prezentace podle vztahu

$$E(t) = d(t) - u(t)$$

5. Umělé neuronové sítě

Krok 5: Adaptace vah

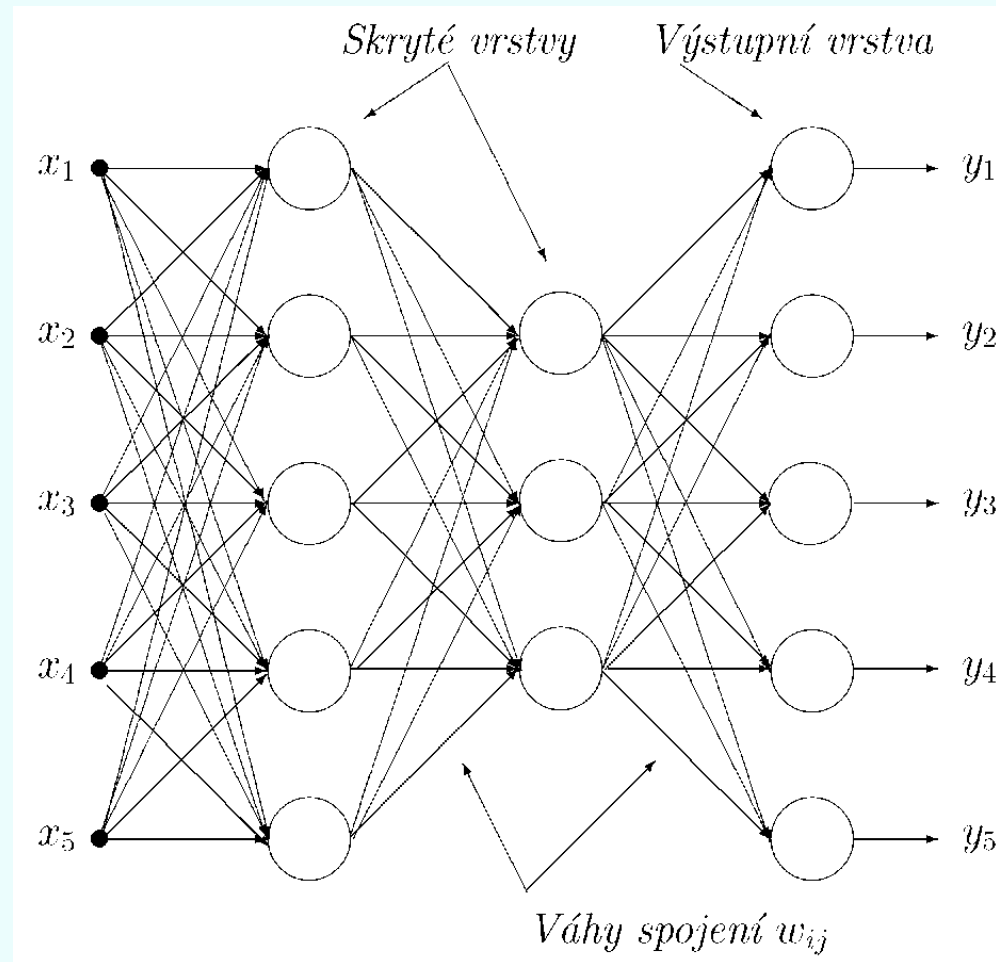
$$w_i(t + 1) = w_i(t) + \frac{\alpha E(t)}{||X||^2} x_i$$

kde $0 < \alpha < 1$ je tzv. koeficient učení

Krok 6: Opakování kroků 3-5, dokud není změna vah menší než požadovaná přesnost

5. Umělé neuronové sítě

Vícevrstvý perceptron



5. Umělé neuronové sítě

Vícevrstvý perceptron je

- vícevrstvá síť s dopředným šířením
- učení: s učitelem
- neuronová aktivační funkce: sigmoidální funkce, hyperbolický tangens, popř. jiná nelineární funkce spojitá a spojitě diferencovatelná v celém definičním oboru

Použití:

- klasifikace obrazů
- aproximace funkcí
- predikce časových řad
- řízení

5. Umělé neuronové sítě

Trénování vícevrstvého perceptronu (Backpropagation alg.)

Krok 1: Počáteční inicializace vah w_{ij} a prahů θ_i jednotlivých neuronů.

Krok 2: Přivedení vstupního vektoru $X = [x_1, \dots, x_N]^T$ a definice požadované výstupní odezvy $D = [d_1, \dots, d_M]^T$.

Krok 3: Výpočet aktuálního výstupu podle následujících vztahů (platí pro třívrstvý perceptron)

$$y_l(\mathbf{t}) = f_s \left(\sum_{k=1}^{N_2} w''_{kl}(\mathbf{t}) x''_k(\mathbf{t}) - \theta''_l \right), \quad 1 \leq l \leq M \quad \text{výstupní vrstva}$$

5. Umělé neuronové sítě

$$x_k''(t) = f_s \left(\sum_{j=1}^{N_1} w_{jk}'(t) x_j'(t) - \theta_k' \right), \quad 1 \leq k \leq N_2 \quad 2. \text{ skrytá vrstva}$$

$$x_j'(t) = f_s \left(\sum_{i=1}^N w_{ij}(t) x_i(t) - \theta_j \right), \quad 1 \leq j \leq N_1 \quad 1. \text{ skrytá vrstva}$$

Krok 4: Adaptace vah a prahů podle vztahu

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j x_i, \text{ popř.}$$

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j x_i + \alpha (w_{ij}(t) - w_{ij}(t-1))$$

5. Umělé neuronové sítě

Nastavení vah začíná u výstupních uzlů a postupuje zpětně směrem ke vstupům. V uvedených vztazích jsou w_{ij} váhy mezi i -tým skrytým uzlem (popř. vstupním uzlem) a uzlem j -tým v čase t , x'_i je výstup i -tého, popř. vstupního uzlu, η je koeficient učení (zisk), α je tzv. momentový koeficient a δ_j je chyba, pro kterou platí následující vztahy:

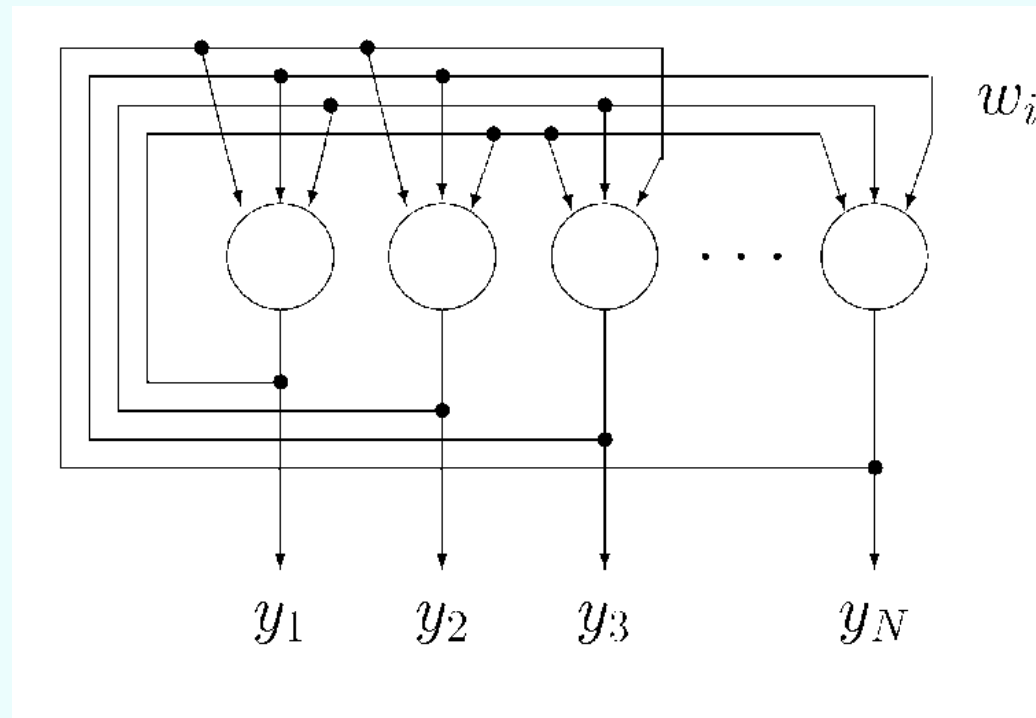
$$\delta_j = \begin{cases} y_j(1 - y_j)(d_j - y_j) & \text{pro výstupní uzly} \\ x'_j(1 - x'_j) \left(\sum_k \delta_k w_{jk} \right) & \text{pro vnitřní skryté uzly} \end{cases}$$

kde k se mění přes všechny uzly vrstvy, která následuje za uzlem j .

Krok 5 : Opakování kroků 3-5, dokud chyba není menší než předem stanovená hodnota.

5. Umělé neuronové sítě

Hopfieldova síť



- jednovrstvá rekurentní síť s pevnými vahami,
- vstupní vektory musí být binární (0/1), popř. bipolární (-1/1)

5. Umělé neuronové sítě

Neuronová aktivační funkce:

- funkce jednotkového skoku $f_H(\mathbf{u}) = \begin{cases} \mathbf{1} & \text{pro } \mathbf{u} > 0 \\ \mathbf{0} & \text{pro } \mathbf{u} < 0 \end{cases}$
- znaménková funkce (pro bipolární vstupní vektory)

Použití:

- autoasociativní paměť
- rekonstrukce neúplných a šumem poškozených obrazů
- optimalizační problémy

5. Umělé neuronové sítě

Kapacita sítě (maximální počet uchovaných vzorů):

$$P \leq 0,15 N,$$

kde P je počet vzorů uchovávaných v paměti, N je počet neuronů v síti

Energie:

$$E(\mathbf{t}) = -\frac{1}{2} \sum_i \sum_j w_{ij} a_i(\mathbf{t}) a_j(\mathbf{t}) - \sum_i \theta_i a_i(\mathbf{t})$$

5. Umělé neuronové sítě

Hopfieldova síť (synchronní model)

Krok 1: Inicializace váhových koeficientů w_{ij} a prahu θ_i podle vztahu

$$w_{ij} = \sum_{s=1}^P (2a_i^s - 1)(2a_j^s - 1), \quad 1 \leq i, j \leq N,$$
$$\theta_i = \frac{1}{2} \sum_{j=1}^N w_{ij}, \quad i = 1, \dots, N,$$

kde w_{ij} jsou váhové koeficienty mezi i -tým a j -tým neuronem, a_i^s je i -tý prvek vzorového vektoru s -té třídy (může nabývat pouze hodnot $\{0,1\}$), P je počet vzorových vektorů.

Krok 2: Předložení neznámého vstupního vektoru $X = [x_1, \dots, x_N]^T$, $x_i \in \{0,1\}$

$$y_i(0) = x_i, \quad 1 \leq i \leq N,$$

kde $y_i(0)$ je výstup i -tého neuronu v čase $t=0$, x_i je i -tý prvek vstupního vektoru.

5. Umělé neuronové sítě

Krok 3: Iterace, dokud síť konverguje:

Během t -té iterace se

1) vypočtou hodnoty $u_i(t + 1)$ paralelně pro $i=1,2,\dots, N$ podle vztahu

$$u_i(t + 1) = \sum_{j=1}^N w_{ij}y_j(t) - \theta_i$$

2) vypočtou výstupní hodnoty $y_i(t + 1)$ pro $i=1,2,\dots, N$ podle vztahu

$$y_i(t + 1) = \begin{cases} \mathbf{1} & \text{pro } u_i(t + 1) > \mathbf{0} \\ \mathbf{0} & \text{pro } u_i(t + 1) < \mathbf{0} \\ u_i(t) & \text{pro } u_i(t + 1) = \mathbf{0} \end{cases}$$

Tento proces se opakuje, dokud se výstup mění. Po ukončení iterací odpovídají výstupy vzorovému vektoru, který se nejvíce podobá vektoru vstupnímu.

Krok 4: Opakování od kroku 2 pro nový vstupní vektor

5. Umělé neuronové sítě

Příklad: Hopfieldova síť (synchronní model)

Dány trénovací vektory $\alpha^1 = [1110]^T$, $\alpha^2 = [1100]^T$

$$w_{ij} = \sum_{s=1}^P (2a_i^s - 1)(2a_j^s - 1), \quad i \neq j, \quad 1 \leq i, j \leq N, \quad \text{tj.}$$

$$\mathbf{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & -1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 0 & -2 \\ 2 & 2 & 0 & -2 \\ 0 & 0 & 2 & 0 \\ -2 & -2 & 0 & 2 \end{bmatrix}$$

$$\boldsymbol{\theta} = \frac{1}{2} \sum_{j=1}^M w_{ij} = [0101]^T$$

5. Umělé neuronové sítě

Pro testovací vektor $\mathbf{y}(0) = \mathbf{x} = [0100]^T$ dostaneme:

a) v 1. iteraci:

$$\mathbf{W}\mathbf{x} - \boldsymbol{\theta} = \begin{bmatrix} 2 & 2 & 0 & -2 \\ 2 & 2 & 0 & -2 \\ 0 & 0 & 2 & 0 \\ -2 & -2 & 0 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

$$\mathbf{y}(1) = f_H(\mathbf{W}\mathbf{x} - \boldsymbol{\theta}) = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

b) v 2. iteraci:

$$\mathbf{W}\mathbf{x} - \boldsymbol{\theta} = \begin{bmatrix} 2 & 2 & 0 & -2 \\ 2 & 2 & 0 & -2 \\ 0 & 0 & 2 & 0 \\ -2 & -2 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ -1 \\ -3 \end{bmatrix}$$

$$\mathbf{y}(2) = f_H(\mathbf{W}\mathbf{x} - \boldsymbol{\theta}) = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

5. Umělé neuronové sítě

Hopfieldova síť (asynchronní model)

Krok 1: Inicializace váhových koeficientů w_{ij} a prahu θ_i podle vztahu

$$w_{ij} = \begin{cases} \sum_{s=1}^P (2a_i^s - 1)(2a_j^s - 1), & i \neq j, \quad 1 \leq i, j \leq N \\ 0 & i = j \end{cases}$$
$$\theta_i = \frac{1}{2} \sum_{j=1}^N w_{ij}, \quad i = 1, \dots, N,$$

kde w_{ij} jsou váhové koeficienty mezi i -tým a j -tým neuronem, a_i^s je i -tý prvek vzorového vektoru s -té třídy (může nabývat pouze hodnot $\{0,1\}$), P je počet vzorových vektorů.

Krok 2: Předložení neznámého vstupního vektoru $X = [x_1, \dots, x_N]^T$, $x_i \in \{0,1\}$

$$y_i(0) = x_i, \quad 1 \leq i \leq N,$$

kde $y_i(0)$ je výstup i -tého neuronu v čase $t=0$, x_i je i -tý prvek vstupního vektoru.

5. Umělé neuronové sítě

Krok 3: Iterace, dokud síť konverguje:

Během t -té iterace se nastavují výstupy neuronů v sekvenčním pořadí pro $i=1,2,\dots,N$

1)

$$u_i(t+1) = \sum_{j=1}^N w_{ij} y_j(t) - \theta_i$$

2)

$$y_i(t+1) = \begin{cases} 1 & \text{pro } u_i(t+1) > 0 \\ 0 & \text{pro } u_i(t+1) < 0 \\ u_i(t) & \text{pro } u_i(t+1) = 0 \end{cases}$$

Krok 4: Opakování od kroku 2 pro nový vstupní vektor

5. Umělé neuronové sítě

Příklad: Hopfieldova síť (asynchronní model)

Dány trénovací vektory $\alpha^1 = [1110]^T$, $\alpha^2 = [1100]^T$

$$w_{ij} = \begin{cases} \sum_{s=1}^P (2a_i^s - 1)(2a_j^s - 1), & i \neq j, \quad 1 \leq i, j \leq N \\ \mathbf{0} & i = j \end{cases}, \text{ tj.}$$

$$W = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & -1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & 2 & 0 & -2 \\ 2 & \mathbf{0} & 0 & -2 \\ 0 & 0 & \mathbf{0} & 0 \\ -2 & -2 & 0 & \mathbf{0} \end{bmatrix}$$

$$\theta = \frac{1}{2} \sum_{j=1}^M w_{ij} = [0 \ 0 \ 0 \ -2]^T$$

5. Umělé neuronové sítě

Pro testovací vektor $\mathbf{y}(0) = \mathbf{x} = [0100]^T$ dostaneme:

a) v 1. iteraci:

$$\mathbf{y}(1) = f_H(\mathbf{W}\mathbf{x} - \boldsymbol{\theta}) = \begin{bmatrix} 0 & 2 & 0 & 2 \\ 2 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 \\ -2 & -2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ -2 \end{bmatrix} \quad \text{postupně pro}$$

1. bit: $[1100]^T$

2. bit: $[1100]^T$

3. bit: $[1100]^T$

4. bit: $[1100]^T$

5. Umělé neuronové sítě

b) v 2. iteraci:

$$\mathbf{y}(2) = f_H(\mathbf{W}\mathbf{x} - \boldsymbol{\theta}) = \begin{bmatrix} 0 & 2 & 0 & 2 \\ 2 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 \\ -2 & -2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ -2 \end{bmatrix} \quad \text{postupně pro}$$

1. bit: $[1100]^T$

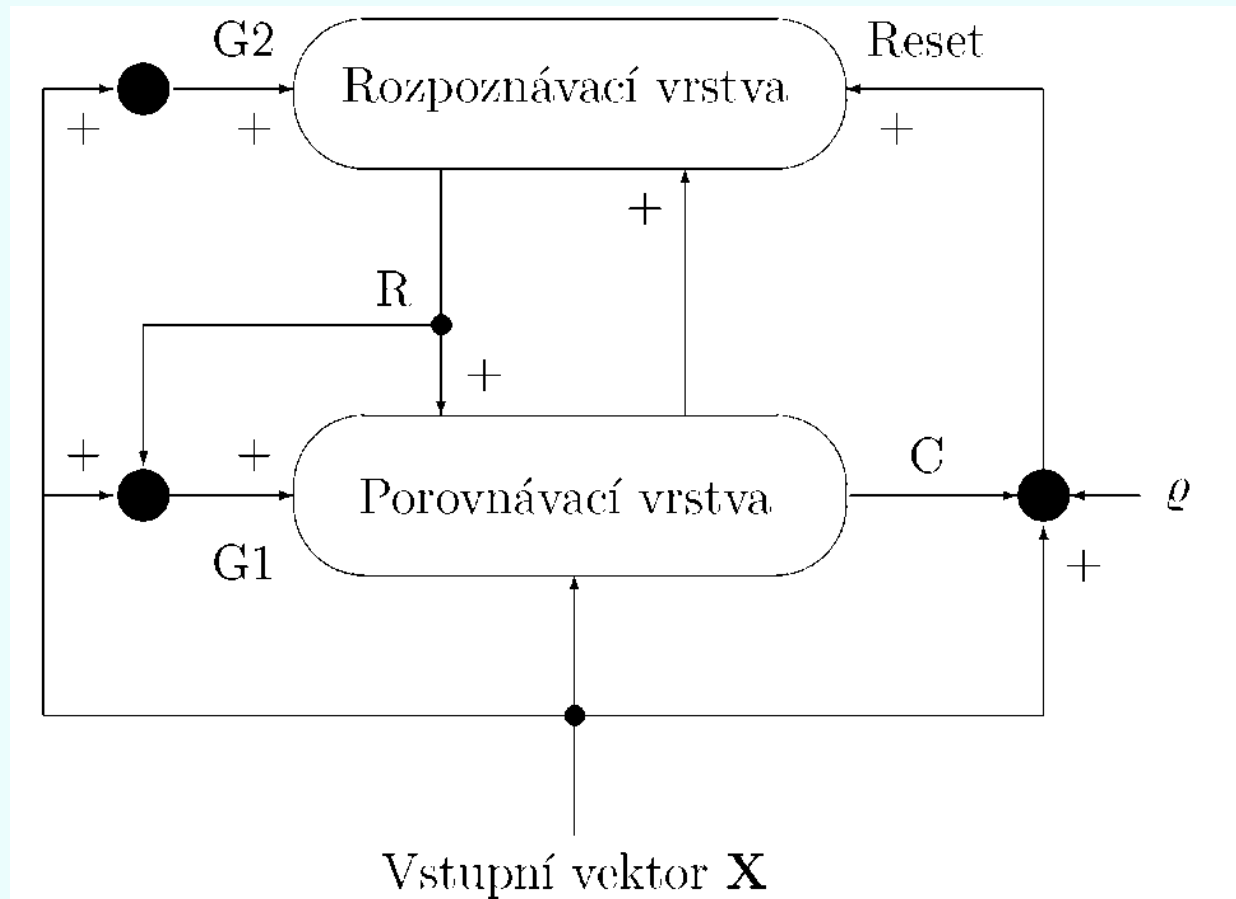
2. bit: $[1100]^T$

3. bit: $[1100]^T$

4. bit: $[1100]^T$

5. Umělé neuronové sítě

ART (Adaptive Resonance Theory)



5. Umělé neuronové sítě

ART je:

- dvouvrstvá rekurentní síť
- učení: bez učitele
- vstupní vektor \mathbf{X} :
 - binární pro model ART-1
 - reálný pro model ART-2

Použití:

- shlukování
- rozpoznávání znaků, řečových segmentů apod.

5. Umělé neuronové sítě

Řídicí signály

G2 – $G2 = 1$, je-li alespoň jeden prvek vstupního vektoru $x_i = 1$

G1 – $G1 = 1$, je-li alespoň jeden prvek $x_i = 1$. Pokud je však zároveň alespoň jeden prvek $r_i = 1$, je $G1=0$

Reset – je generován, je-li podobnost mezi vektorem **C** a **X** menší než předem zvolená hodnota prahu “ostrážitosti“ ρ . Podobnost mezi vektory se počítá jako skalární součin **X** a **T** dělený počtem jednotkových bitů vektoru **X**.

5. Umělé neuronové sítě

Porovnávací vrstva

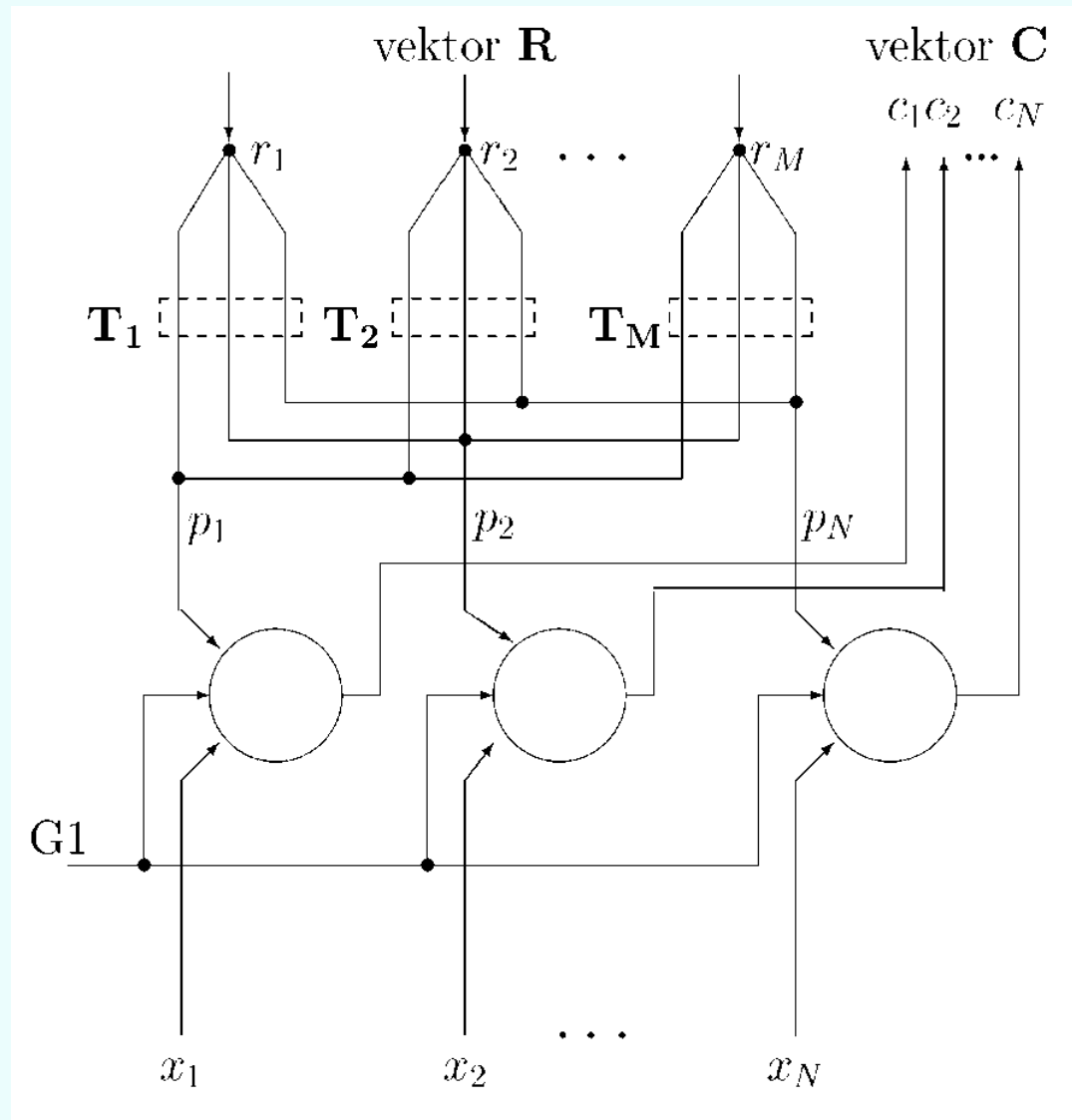
Každý neuron v této vrstvě přijímá tři binární signály:

- 1) složky x_i vstupního vektoru $\mathbf{X} = [x_1, x_2, \dots, x_N]^T$,
- 2) zpětnovazební signál p_j , který je vytvořen jako vážený součet výstupů neuronů z rozpoznávací vrstvy,
- 3) signál $G1$.

Výstup neuronů c_i je roven jedné, pokud jsou nejméně dva ze tří signálů rovny jedné.

Na počátku jsou všechny složky vektoru \mathbf{R} a signál $G1$ nastaveny na hodnotu 0 $\Rightarrow \mathbf{C} = \mathbf{X}$.

5. Umělé neuronové sítě



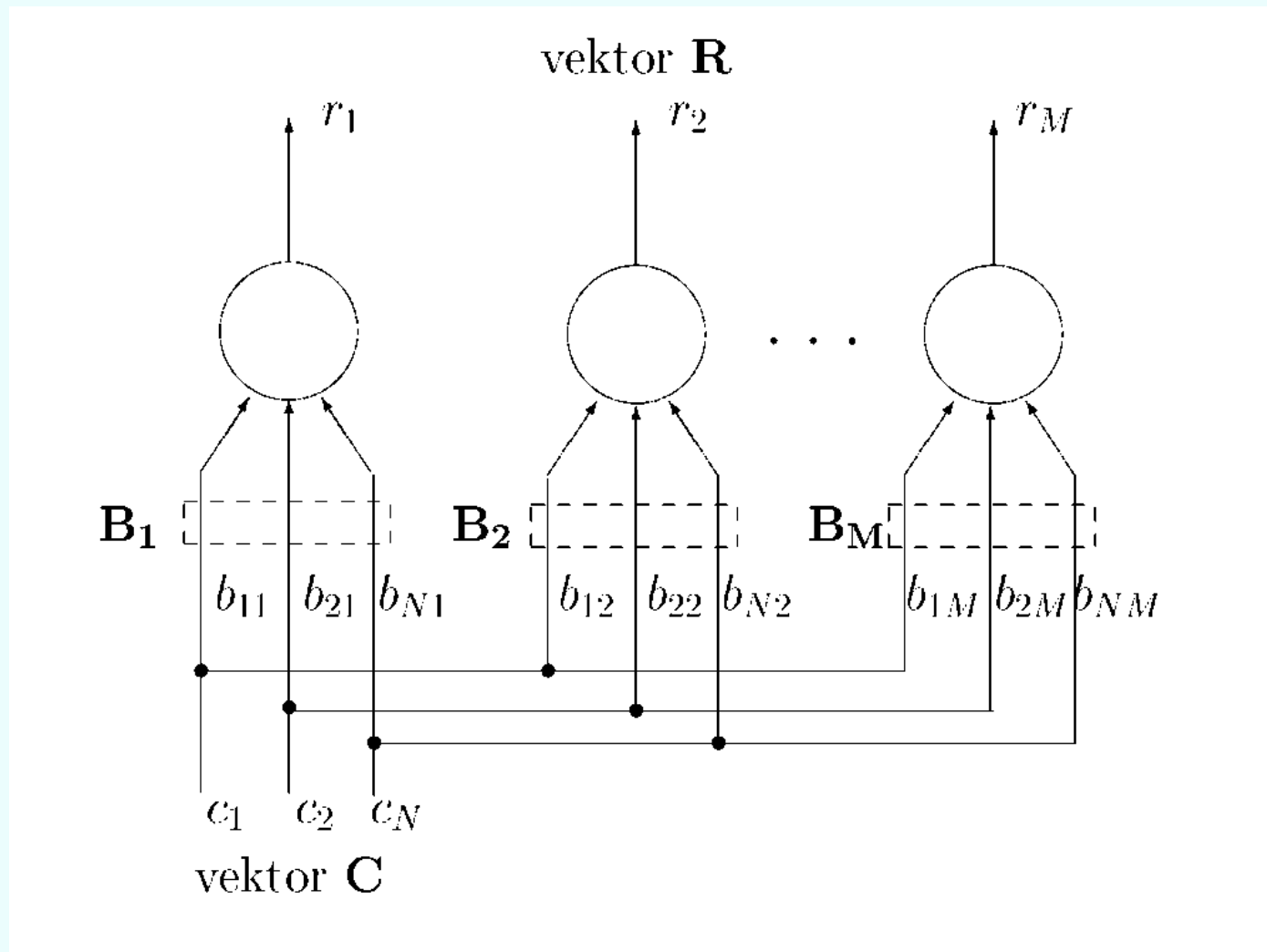
5. Umělé neuronové sítě

Rozpoznávací vrstva

Rozpoznávací vrstva klasifikuje vstupní vektor. Skládá se z M neuronů (M je počet klasifikačních tříd), vstupy neuronů jsou spojeny s výstupem porovnávací vrstvy. Spojením jsou přiřazeny vektory váhových koeficientů B_j ($b_{ji} \in \mathbf{R}$).

Po předložení vstupního vektoru je v rozpoznávací vrstvě aktivován pouze neuron s vektorem váhových koeficientů nejvíce podobným vstupu. Ostatní neurony budou mít výstup nulový.

5. Umělé neuronové sítě



5. Umělé neuronové sítě

Algoritmus použití ART sítě

Krok 1: Počáteční inicializace vah t_{ij}, b_{ij} a nastavení prahu ostrážitosti ρ

$$t_{ij}(0) = 1$$

$$b_{ij}(0) = \frac{1}{1 + N}$$

$$i = 1, 2, \dots, N,$$

$$j = 1, 2, \dots, M,$$

$$0 \leq \rho \leq 1$$

Krok 2: Přivedení vektoru $\mathbf{X} = [x_1, \dots, x_N]^T$ na vstup sítě

5. Umělé neuronové sítě

Krok 3: Výpočet odezvy neuronu v rozpoznávací vrstvě podle vztahu

$$y_j = \sum_{i=1}^N b_{ij}(t)x_i, \quad j = 1, 2, \dots, M$$

kde y_j je odezva j -tého neuronu v rozpoznávací vrstvě, x_i je i -tý prvek vstupního vektoru

Krok 4: Výběr neuronu s největší odezvou $y_k = \max(y_j)$

Krok 5: Výpočet podobnosti μ podle vztahu

$$\mu = \frac{\sum_{i=1}^N t_{ik}x_i}{\sum_{i=1}^N x_i}$$

Je-li $\mu \geq \rho$, pokračování krokem 7, jinak provedení kroku 6

5. Umělé neuronové sítě

Krok 6: Zablokování neuronu s největší odezvou a opakování od kroku 3. Výstup k-tého neuronu je dočasně nastaven na nulovou hodnotu a neuron nebude testován při dalším výběru maxima

Krok 7: Adaptace vah u neuronu s největší odezvou. V tomto kroku se modifikují váhy spojení t_{ik} a b_{ik} podle vztahů

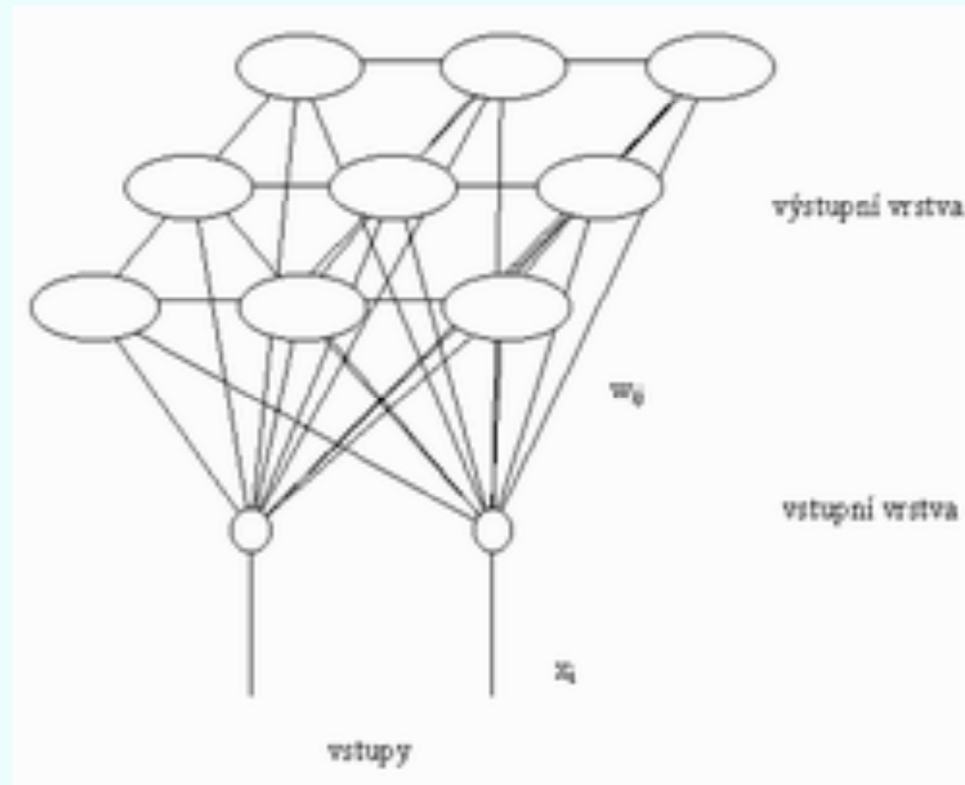
$$t_{ik}(t + 1) = t_{ik}(t)x_i$$

$$b_{ik}(t + 1) = \frac{t_{ik}(t)x_i}{0,5 + \sum_{i=1}^N t_{ik}(t)x_i}$$

Krok 8: Odblokování jednotek zablokovaných v kroku 6 a opakování od kroku 2

5. Umělé neuronové sítě

Kohonenova síť (Self Organizing Map – SOM)



5. Umělé neuronové sítě

Kohonenova síť je

- jednovrstvá síť s dopředným šířením
- učení: bez učitele
- vstupní vektor: $\mathbf{X} = [x_1, \dots, x_N]^T$, $x_i \in \mathbf{R}$
- výstupní hodnota neuronu je definována jako vzdálenost mezi vstupním a váhovým vektorem, t.j.

$$y_i(t) = \sum_{j=1}^N (x_j(t) - w_{ij}(t))^2$$

Použití pro:

- shlukování
- analýzu dat
- vytváření sémantických map aj.

5. Umělé neuronové sítě

Algoritmus (Kohonenova síť)

Krok 1: Počáteční inicializace vah w_{ij} a množiny $Ne_i(t)$, která definuje sousednost okolo každého uzlu $i = 1, 2, \dots, N$.

Krok 2: Přivedení vektoru $X = [x_1, \dots, x_N]^T$ na vstup sítě a výpočet vzdálenosti mezi vstupním vektorem a váhovým vektorem každého neuronu podle vztahu

$$d_i(t) = \sum_{j=1}^N (x_j(t) - w_{ij}(t))^2$$

Krok 3: Výběr i -tého neuronu s nejmenší vzdáleností d_i (tzv. vítězného neuronu)

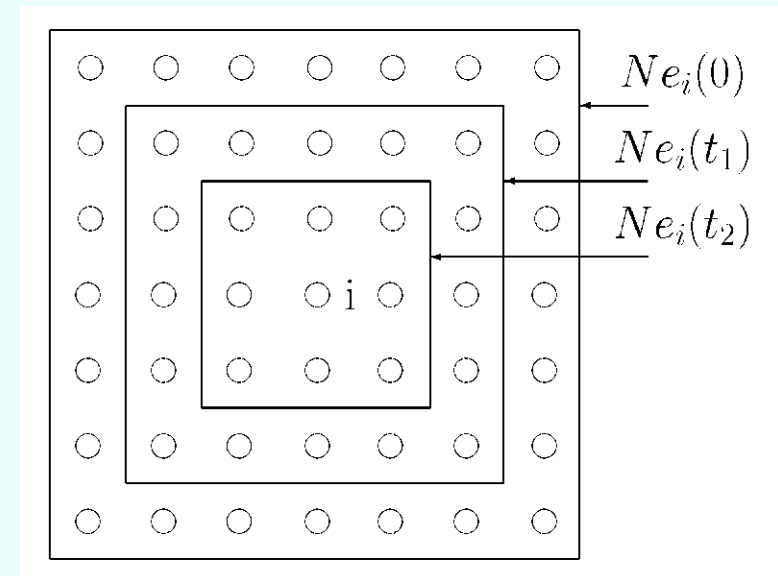
5. Umělé neuronové sítě

Krok 4: Adaptace vah vítězného neuronu a všech neuronů k , pro které platí $k \in Ne_i(t)$ (sousedí s i -tým neuronem) podle vztahu

$$w_{ki}(t+1) = w_{ki}(t) + \eta(t)(x_i(t) - w_{ki}(t)), \quad \forall k \in Ne_i(t), \quad i = 1, 2, \dots, N,$$

kde $\eta(t)$ ($0 < \eta(t) < 1$) je koeficient učení, jehož hodnota s rostoucím časem klesá

Krok 5: Změna sousednosti $Ne_i(t)$ pro $i=1,2,\dots,N$



Krok 6: Opakování kroků 2-5 pro všechny vstupní vektory, dokud dochází ke změně vah.

5. Umělé neuronové sítě

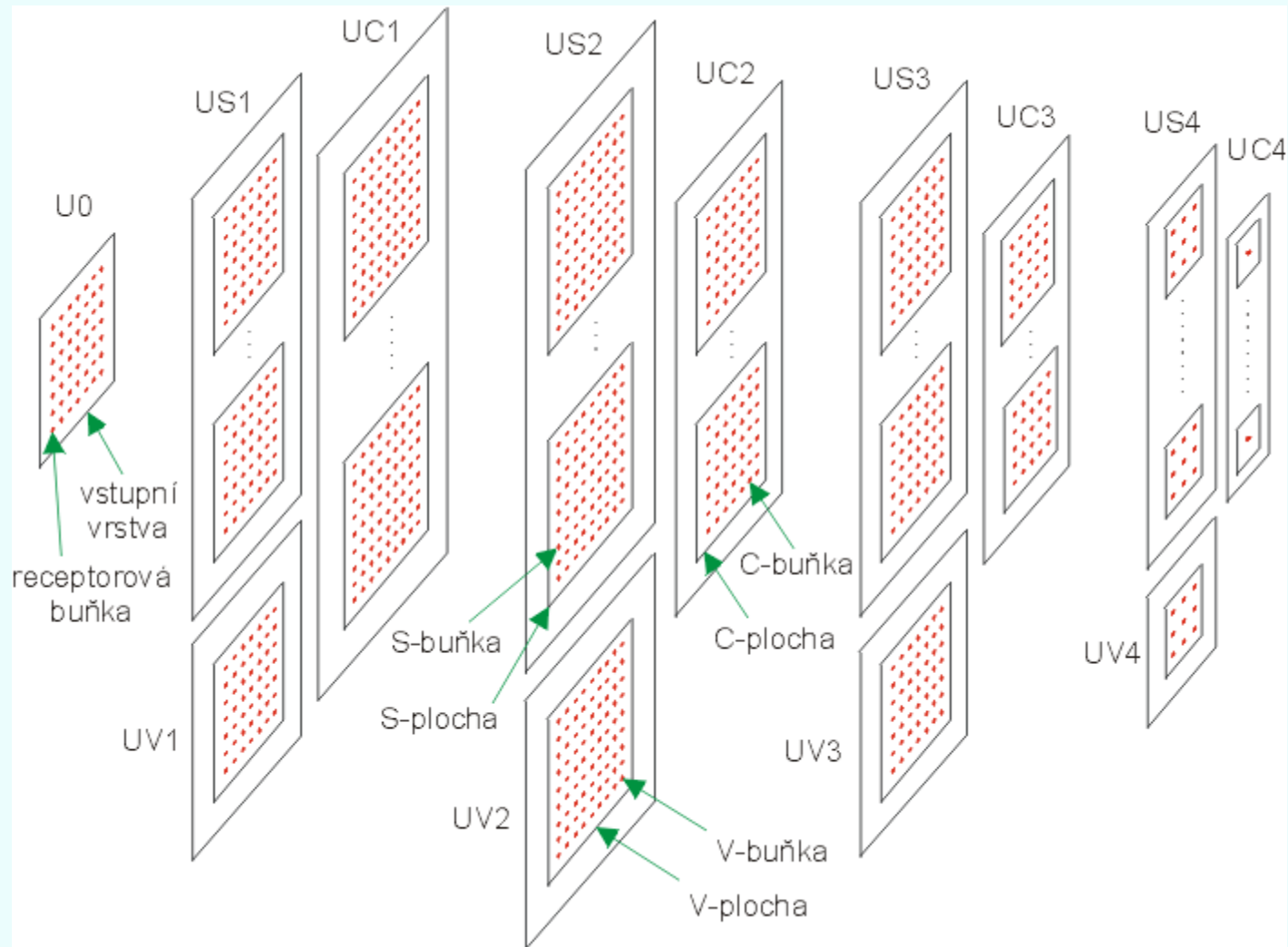
Neocognitron

- vícevrstvá hierarchická neuronová síť

Použití: např. rozpoznávání rukou psaných znaků

Výhoda: schopnost správně rozpoznávat nejen naučené obrazy, ale i obrazy, které vzniknou částečným posunutím, otočením nebo jinou deformací

5. Umělé neuronové sítě



5. Umělé neuronové sítě

Příklad použití: