

Neuronové sítě

Doc. RNDr. Iveta Mrázová, CSc.

Katedra teoretické informatiky

Matematicko-fyzikální fakulta

Univerzity Karlovy v Praze

Neuronové sítě

- Asociativní paměti –
- BAM a Hopfieldův model –

Doc. RNDr. Iveta Mrázová, CSc.

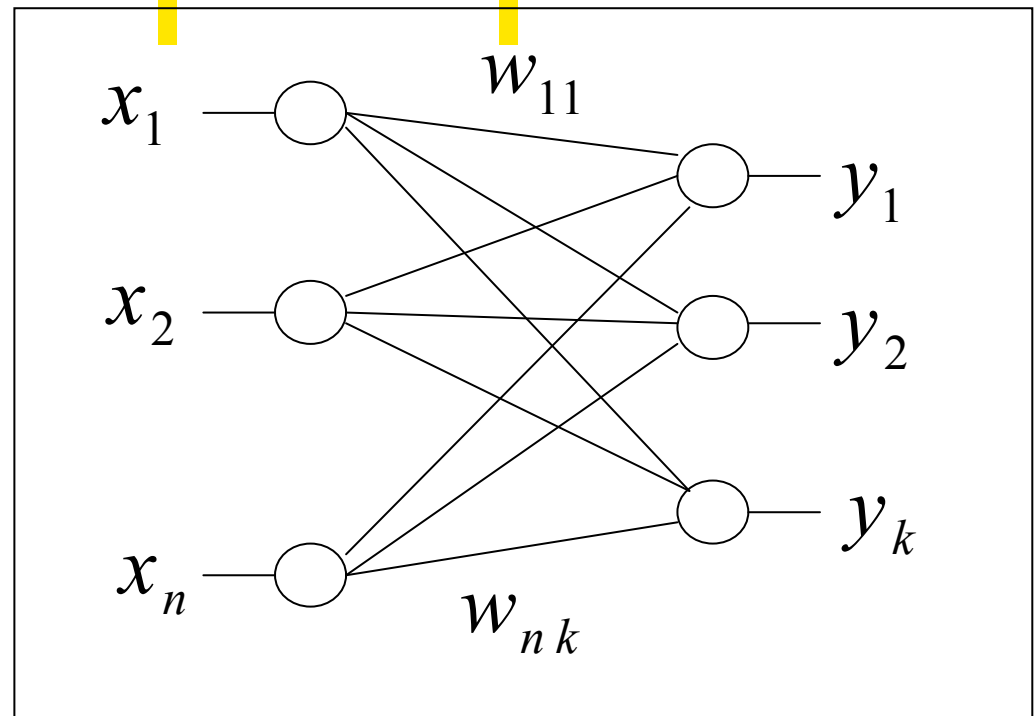
Katedra teoretické informatiky

Matematicko-fyzikální fakulta

Univerzity Karlovy v Praze

Bidirektivní asociativní paměť

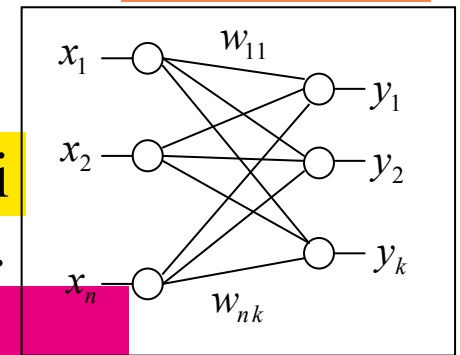
BAM – synchronní asociativní model s obousměrnými synapsemi



Bidirektivní asociativní paměť (2)

◆ Rekurentní asociativní paměť

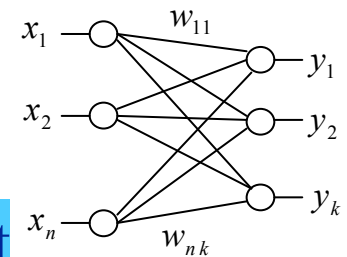
- Skládá se ze dvou vrstev neuronů, které si mezi sebou rekurzivně posílají informace.
- Vstupní vrstva posílá výsledky svého výpočtu výstupní vrstvě – prostřednictvím vah sítě.
- Výstupní vrstva vrací výsledky svých výpočtů zpět vstupní vrstvě – prostřednictvím stejných vah.



- ◆ Otázka: Dosáhne síť stabilního stavu, kdy se po několika iteracích už nebude měnit informace posílaná sem a tam?

BAM \approx Bidirectional Associative Memory (3)

- ◆ Rezonanční síť
- ◆ Aktivační (přenosovou) funkcí je **sgn**
- ◆ Informace kódována pomocí bipolárních hodnot

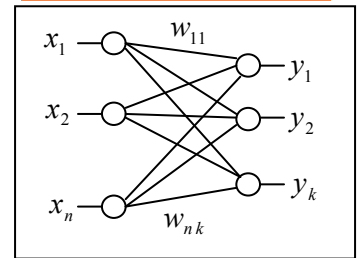


- ◆ Síť zobrazí n -rozměrný vektor $\vec{\mathbf{x}}_0$ na k -rozměrný vektor $\vec{\mathbf{y}}_0$
- ◆ Váhová matice sítě je $n \times k$ matice \mathbf{W} .
 - Po prvním průchodu sítě dostáváme: $\vec{\mathbf{y}}_0 = \text{sgn}(\vec{\mathbf{x}}_0 \mathbf{W})$
 - Vstup po zpětném průchodu sítě bude: $\vec{\mathbf{x}}_1 = \text{sgn}(\mathbf{W} \vec{\mathbf{y}}_0^T)$
 - Výstup po dalším průchodu: $\vec{\mathbf{y}}_1 = \text{sgn}(\vec{\mathbf{x}}_1 \mathbf{W})$

BAM \approx Bidirectional Associative Memory (4)

- Po m iteracích dostáváme $m + 1$ dvojic vektorů $(\vec{x}_0, \vec{y}_0), \dots, (\vec{x}_m, \vec{y}_m)$, které splňují podmínku:

$$\vec{y}_i = \text{sgn}(\vec{x}_i \mathbf{W}) \quad \text{a} \quad \vec{x}_{i+1} = \text{sgn}(\mathbf{W} \vec{y}_i^T)$$



- Otázka:** Nalezne systém po několika iteracích pevný bod (\vec{x}, \vec{y}) tak, aby platilo

$$\vec{y} = \text{sgn}(\vec{x} \mathbf{W}) \quad \text{a} \quad \vec{x} = \text{sgn}(\mathbf{W} \vec{y}^T) \quad ?$$

BAM \approx Bidirectional Associative Memory (5)

→ máme-li dvojici vektorů (\vec{x}, \vec{y}) a chceme nastavit váhy bidirektivní asociativní paměti tak, aby tato uspořádaná dvojice představovala pevný bod systému, lze k výpočtu odpovídající matice vah použít Hebbovské učení: $W = \vec{x}^T \vec{y}$

$$\rightarrow \vec{y} = \text{sgn}(\vec{x} W) = \text{sgn}(\vec{x} \vec{x}^T \vec{y}) = \text{sgn}(\|\vec{x}\|^2 \vec{y}) = \vec{y}$$

a zároveň:

$$\vec{x}^T = \text{sgn}(W \vec{y}^T) = \text{sgn}(\vec{x}^T \vec{y} \vec{y}^T) = \text{sgn}(\vec{x}^T \|\vec{y}\|^2) = \vec{x}^T$$

BAM \approx Bidirectional Associative Memory (6)

Chceme-li uložit do paměti více vzorů $(\vec{x}_1, \vec{y}_1), \dots, (\vec{x}_m, \vec{y}_m)$, bude Hebbovské učení efektivnější, pokud jsou vektory $\vec{x}_1, \dots, \vec{x}_m$ a $\vec{y}_1, \dots, \vec{y}_m$ navzájem ortogonální (menší „crosstalk“)

→ pro m dvojic vektorů bude matice W :

$$W = \vec{x}_1^T \vec{y}_1 + \vec{x}_2^T \vec{y}_2 + \dots + \vec{x}_m^T \vec{y}_m$$

→ navíc lze bidirektivní asociativní paměť použít i při konstrukci autoasociativních sítí, protože matice vah vytvořené při Hebbovském učení (anebo při výpočtu pseudoinverzní matice) jsou symetrické ($X = X W$ a $X^T = W X^T$)

Energetická funkce pro BAM

- ◆ Necht' pro danou síť BAM představuje dvojice (\vec{x}, \vec{y}) stabilní stav
- ◆ Při inicializaci je síti (zleva) předložen vstupní vektor \vec{x}_0
(časem by měla dokonvergovat k (\vec{x}, \vec{y}))
- ◆ Výpočet vektoru \vec{y}_0 podle: $\vec{y}_0 = \text{sgn}(\vec{x}_0 W)$
- ◆ Výstupní vektor \vec{y}_0 bude použit pro novou iteraci (zprava)

Energetická funkce pro BAM (2)

- ◆ Excitaci neuronů vlevo určit podle excitačního vektoru \vec{e} : $\vec{e}^T = W \vec{y}_0$
 - (\vec{x}_0, \vec{y}_0) by odpovídalo stabilnímu stavu sítě, pokud $\text{sgn}(\vec{e}) = \vec{x}_0$
 - tj. pokud je \vec{e} dostatečně blízko
 - (→ skalární součin $\vec{x}_0 \vec{e}^T$ by měl být větší než skalární součin jiných excitačních vektorů a \vec{x}_0 , i když jsou stejně dlouhé)

Energetická funkce pro BAM (3)

→ součin $E = - \vec{x}_0 \vec{e}^T = - \vec{x}_0 W \vec{y}_0^T$ tedy bude menší, pokud leží vektor $W \vec{y}_0$ blíže k \vec{x}_0

→ **možnost sledování konvergence systému ke stabilnímu stavu**

$E \sim$ energetická funkce

Lokální minima energetické funkce odpovídají stabilním stavům

Energetická funkce pro BAM (4)

Definice:

Nechť W je váhová matice sítě BAM a necht' výstup \vec{y}_i pravé vrstvy neuronů se v i – té iteraci spočítá podle $\vec{y}_i = \text{sgn} \left(\vec{x}_i W \right)$ a

A výstup \vec{x}_{i+1} levé vrstvy neuronů necht' se počítá podle $\vec{x}_{i+1} = \text{sgn} \left(W \vec{y}_i^T \right)$

Energetická funkce sítě BAM je pak určena pomocí:

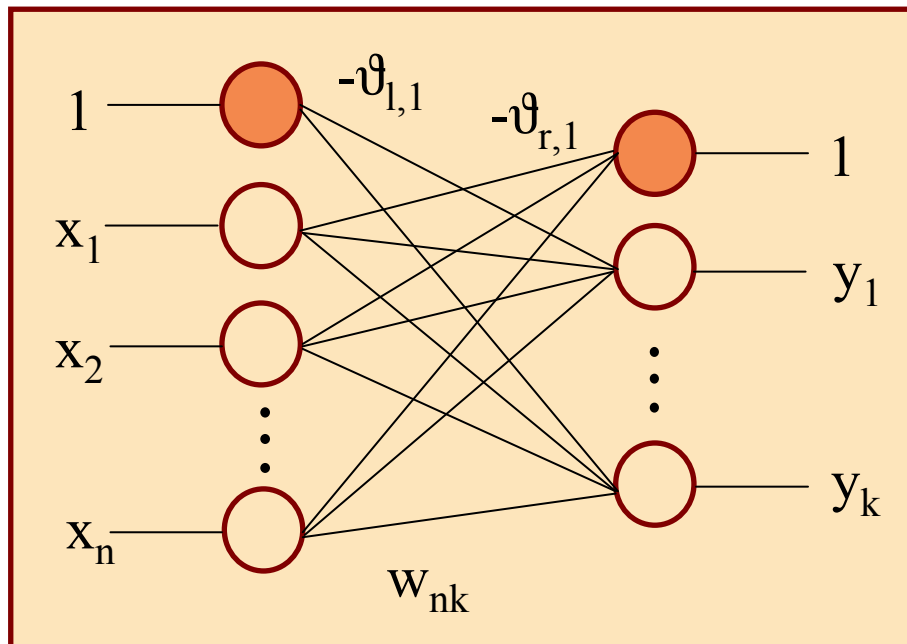
$$E \left(\vec{x}_i, \vec{y}_i \right) = - \frac{1}{2} \vec{x}_i W \vec{y}_i^T$$

Zobecnění energetické funkce

Uvažujeme práh a skokovou přenosovou funkci

- ♦ Každý n – rozměrný vektor \vec{x} bude transformován na vektor $(x_1, \dots, x_n, 1)$
- ♦ Každý k – rozměrný vektor \vec{y} , $y_j = \sum w_i x_i - \vartheta_{l,j}$ bude transformován na vektor $(y_1, \dots, y_k, 1)$
- ♦ Váhová matice W bude rozšířena na matici W' , která má oproti W řádek a sloupec navíc

Zobecnění energetické funkce (2)



- ♦ Záporné prahy neuronů z pravé vrstvy sítě BAM tvoří $(n+1)$ – ní řádku W'
- ♦ Záporné prahy neuronů z levé vrstvy sítě tvoří $(k+1)$ – ní sloupec W'
- ♦ Prvek $(n+1, k+1)$ matice W' bude 0

Zobecnění energetické funkce (3)

- ♦ Tato transformace odpovídá zavedení dalšího neuronu s konstantním výstupem **1** do obou vrstev
 - Váhy vedoucí z těchto přidaných neuronů odpovídají záporné hodnotě prahu neuronů, do nichž vedou

→ **Energetická funkce modifikované sítě BAM:**

$$E(\vec{x}_i, \vec{y}_i) = -\frac{1}{2} \vec{x}_i W \vec{y}_i^T + \frac{1}{2} \vec{\theta}_r \vec{y}_i^T + \frac{1}{2} \vec{x}_i \vec{\theta}_l^T$$

$\vec{\theta}_l^T$ Vektor prahů **k** neuronů (v levé vrstvě sítě)

$\vec{\theta}_r$ Vektor prahů **n** neuronů (v pravé vrstvě sítě)

Asynchronní síť BAM

- ◆ Každý neuron náhodně spočítá svou excitaci a
- ◆ Změní svůj stav na 1 nebo -1 nezávisle na ostatních (ale podle znaménka své excitace)
- ◆ Pravděpodobnost, že budou dva neurony současně měnit svůj stav, je nulová
- ◆ Předpoklad: stav neuronu se nemění, je-li celková excitace nulová

Asynchronní síť BAM (2)

BAM nalezne stabilní stav v konečném počtu iterací (sekvenční procházení neuronů sítě)

- ♦ **Stabilní stav**

~ dvojice vektorů (\vec{x}, \vec{y}) ; $\vec{y} = \text{sgn}(\vec{x}W)$ a $\vec{x}^T = \text{sgn}(W\vec{y}^T)$

Věta:

Bidirektivní asociativní paměť s libovolnou maticí vah W dosáhne stabilního stavu v konečném počtu iterací – a to jak pomocí synchronní, tak také pomocí asynchronní aktualizace.

Asynchronní síť BAM (3)

Důkaz:

- ♦ Pro vektory $\vec{x} = (x_1, x_2, \dots, x_n)$ a $\vec{y} = (y_1, y_2, \dots, y_k)$ a váhovou matici $n \times k$ $W = \{w_{ij}\}$ je energetická funkce $E(\vec{x}, \vec{y})$ rovna:

$$E(\vec{x}, \vec{y}) = -\frac{1}{2} (x_1, \dots, x_n) \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1k} \\ w_{21} & w_{22} & \cdots & w_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nk} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{pmatrix}$$

Asynchronní síť BAM (4)

Důkaz (pokračování):

- ♦ Součin i – té řádky W a \vec{y}^T udává míru excitace i – tého neuronu z levé vrstvy g_i
 - pro excitační vektor levé vrstvy (g_1, \dots, g_n) pak platí:

$$E(\vec{x}, \vec{y}) = -\frac{1}{2} (x_1, \dots, x_n) \begin{pmatrix} g_1 \\ \vdots \\ g_n \end{pmatrix}$$

- podobně pro pravou vrstvu a její excitační vektor (e_1, \dots, e_k) pak platí:

$$E(\vec{x}, \vec{y}) = -\frac{1}{2} (e_1, \dots, e_k) \begin{pmatrix} y_1 \\ \vdots \\ y_k \end{pmatrix}$$

Asynchronní síť BAM (5)

Důkaz (pokračování):

- ◆ Energetickou funkci lze vyjádřit dvěma navzájem ekvivalentními způsoby:

$$E(\vec{x}, \vec{y}) = -\frac{1}{2} \sum_{i=1}^k e_i y_i$$

$$\text{a} \quad E(\vec{x}, \vec{y}) = -\frac{1}{2} \sum_{i=1}^n g_i x_i$$

- ◆ V případě asynchronních sítí se v každém kroku vybere náhodně jeden neuron z levé či pravé vrstvy:
 - Spočítá se excitace a nový stav neuronu

Asynchronní síť BAM (6)

Důkaz (pokračování):

- Pokud se stav neuronu nezmění, zůstane beze změny i energie sítě
- Stav neuronu i z levé vrstvy sítě se změní pouze v případě, že má excitace g_i různé znaménko než jeho aktuální stav x_i
- Protože ostatní neurony svůj stav nemění (asynchronní dynamika), bude rozdíl mezi předchozí energií $E(\vec{x}, \vec{y})$ a novou energií $E(\vec{x}', \vec{y})$ odpovídat:

$$E(\vec{x}, \vec{y}) - E(\vec{x}', \vec{y}) = -\frac{1}{2} g_i (x_i - x'_i)$$

Asynchronní síť BAM (7)

Důkaz (pokračování):

- Protože $\mathbf{x}_i - \mathbf{x}_i'$ má různé znaménko od \mathbf{g}_i , bude:

$$E(\vec{x}, \vec{y}) - E(\vec{x}', \vec{y}) > 0$$

(pokud by $\mathbf{x}_i - \mathbf{x}_i'$ mělo stejné znaménko jako \mathbf{g}_i , nenastala by změna stavu neuronu)

→ Nový stav sítě (\vec{x}', \vec{y}) má tedy nižší energii než původní stav (\vec{x}, \vec{y})

- ◆ Analogicky pro neurony z pravé vrstvy sítě:

$$E(\vec{x}, \vec{y}) - E(\vec{x}, \vec{y}') > 0$$

(Pokud došlo ke změně stavu neuronu.)

Asynchronní síť BAM (8)

Důkaz (pokračování):

- ♦ Každá aktualizace stavu sítě vede ke snížení její celkové energie
 - ♦ Protože existuje konečně mnoho kombinací bipolárních hodnot stavů, musí proces skončit v nějakém stavu (\vec{a}, \vec{b}) , kdy už nelze energii sítě dále snižovat
- síť našla lokální minimum energetické funkce a stav (\vec{a}, \vec{b}) je atraktorem systému

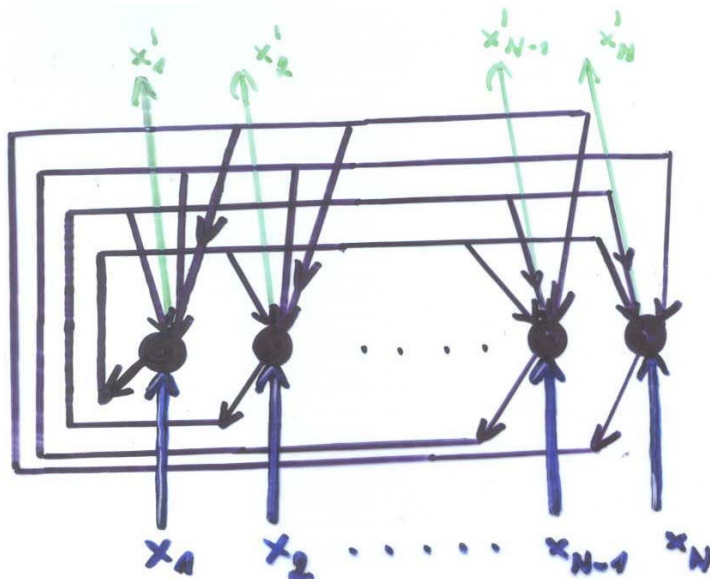
QED

Asynchronní síť BAM (9)

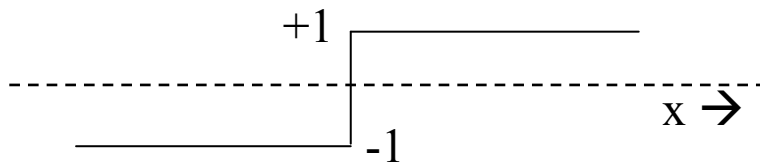
Poznámka:

- ♦ Věta platí i pro síť se synchronní dynamikou
- **libovolná reálná matice vah W má bidirektivní stabilní bipolární stavy**

Hopfieldovy sítě



Skoková přenosová funkce: f_h



- ♦ n neuronů se skokovou přenosovou funkcí
- ♦ Bipolární vstupy i výstupy $\{+1, -1\}$
- ♦ Synaptické váhy w_{ij} (mezi všemi neurony navzájem)
- ♦ m trénovacích vzorů (tříd)
- ♦ Učení s učitelem
- ♦ Rozpoznávání
- ♦ Použití:
 - Asociativní paměť
 - Optimalizační úlohy

Hopfieldův model (bipolární)

Krok 1: Učení - nastavte hodnoty synaptických vah

$$w_{ij} = \begin{cases} \sum_{s=1}^m x_i^s x_j^s & \text{pro } i \neq j \\ 0 & \text{pro } i = j \end{cases}$$

w_{ij} Váha synapse mezi neurony i a j
 $x_i^s \in \{-1, +1\}$ i – tá složka s – tého vzoru
 $1 \leq i, j \leq n$

Hopfieldův model (bipolární) (2)

Krok 2: Inicializace - předložte neznámý vstupní

vzor: $y_i(0) = x_i \quad 1 \leq i \leq n$

$y_i(t)$ Výstup neuronu i v čase t

$x_i \in \{-1, +1\}$ i – tá složka předloženého vzoru

Krok 3: Iterace

$$y_j(t+1) = f_h \left[\sum_{i=1}^n w_{ij} y_i(t) \right] \quad 1 \leq j \leq n$$

f_h Skoková přenosová funkce

Hopfieldův model (bipolární) (3)

Iterativní proces se při rozpoznávání opakuje, dokud se výstupy neuronů neustálí. Výstupy neuronů pak reprezentují ten trénovací vzor, který nejlépe odpovídá předloženému (neznámému) vzoru.

Krok 4: Přejděte ke Kroku 2.

Hopfieldův model (bipolární) (4)

Konvergence (Hopfield):

- ◆ Symetrické váhy: $w_{ij} = w_{ji}$
- ◆ Asynchronní aktualizace výstupu jednotlivých neuronů

Nevýhody:

- ◆ Kapacita ($m < 0.15 n$) $n / 2 \log n$
- ◆ Stabilita (\rightarrow ortogonalizace)

Hopfieldův model – příklad

Učení:

- ♦ Vzory: $[-1, -1, 1, 1]$
 $[1, -1, 1, -1]$

- ♦ Nastavení vah:

$$w_{ij} = \sum_{m=1}^M x_i^{(m)} x_j^{(m)} \quad i \neq j$$

$$w_{ij} = 0 \quad i = j$$

Hopfieldův model – příklad (2)

- ◆ Nastavení vah:

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

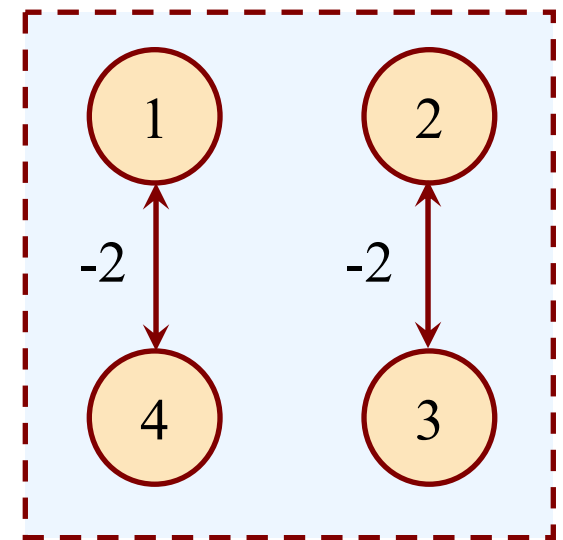
Rozpoznávání:

- ◆ Vzor:

-1, -1, 1, -1

[1, -1, 1, -1]

[-1, -1, 1, 1]



Hopfieldův model - rozpoznávání

Po předložení vzoru \vec{x}_1 bude vektor potenciálů sítě

$$\begin{aligned}\vec{\xi} &= \vec{x}_1 \cdot W = \vec{x}_1 \cdot \left(\vec{x}_1^T \vec{x}_1 + \dots + \vec{x}_m^T \vec{x}_m - mI \right) = \\ &= \underbrace{\vec{x}_1 \vec{x}_1^T \vec{x}_1}_{=n} + \underbrace{\vec{x}_1 \vec{x}_2^T \vec{x}_2}_{=\alpha_{12}} + \dots + \underbrace{\vec{x}_1 \vec{x}_m^T \vec{x}_m}_{=\alpha_{1m}} - m\vec{x}_1 I = \\ &= (n - m)\vec{x}_1 + \underbrace{\sum_{j=2}^m \alpha_{1j} \vec{x}_j}_{PERTURBACE}\end{aligned}$$

Hopfieldův model – rozpoznávání (2)

$\alpha_{12}, \dots, \alpha_{1m}$ Skalární součin \vec{x}_1 s každým dalším vektorem $\vec{x}_2, \dots, \vec{x}_m$

→ Stav \vec{x}_1 je stabilní, jestliže $m < n$ a

perturbace $\sum_{j=2}^m \alpha_{1j} \vec{x}_j$ je malá

$$\left(\Rightarrow \operatorname{sgn} \left(\vec{\xi} \right) = \operatorname{sgn} \left(\vec{x}_1 \right) \right)$$

→ Malý počet ortogonálních vzorů

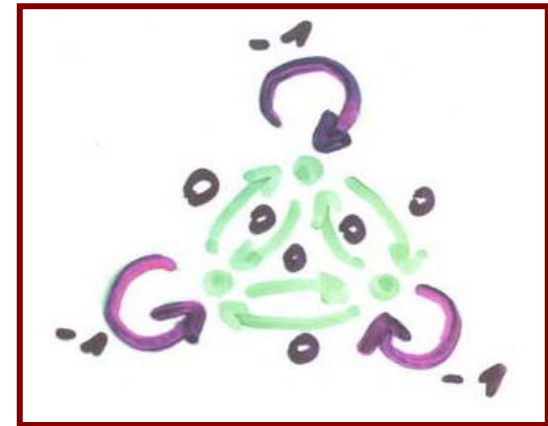
Hopfieldův model – rozpoznávání (3)

- ♦ Stav neuronů zachován, dokud nejsou vybrány k aktualizaci
 - ♦ Výběr pro aktualizaci se provádí náhodně
 - ♦ Neurony jsou navzájem plně propojeny
 - ♦ Symetrické váhy: $w_{ij} = w_{ji}$
 - ♦ $w_{ii} = 0$
- Konvergence ke stabilnímu řešení při rozpoznávání
- nutná podmínka:
symetrická váhová matice s nulovou diagonálou
a asynchronní dynamikou

Hopfieldův model - příklady

- ♦ Váhová matice s nenulovou diagonálou nemusí vést ke stabilním stavům

$$W = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$



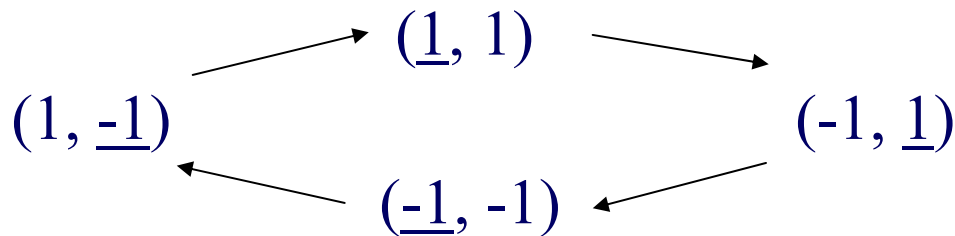
- Synchronní dynamika: $(-1, -1, -1) \leftrightarrow (1, 1, 1)$
- Asynchronní dynamika:
 - Náhodný výběr jednoho z osmi možných vzorů

Hopfieldův model – příklady (2)

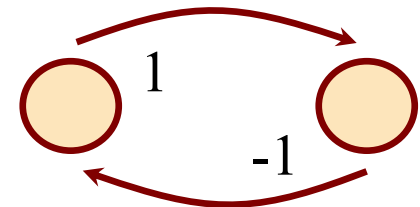
- ◆ Nesymetrická matice:

$$W = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

- Asynchronní dynamika:



cyklické změny



Energetická funkce

Energetická funkce $E(\vec{x})$ Hopfieldovy sítě s n neurony a váhovou maticí W vyjadřuje energii sítě ve stavu \vec{x} :

$$E(\vec{x}) = -\frac{1}{2} \vec{x} W \vec{x}^T = -\frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n w_{ij} x_i x_j$$

(Obdobně i pro sítě s prahovými neurony:

$$\begin{aligned} E(\vec{x}) &= -\frac{1}{2} \vec{x} W \vec{x}^T + \vec{\theta} \vec{x}^T = \\ &= -\frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n w_{ij} x_i x_j + \sum_{i=1}^n \theta_i x_i \end{aligned})$$

Energetická funkce (2)

Věta:

Hopfieldova síť s asynchronní dynamikou dosáhne z libovolného počátečního stavu sítě stabilního stavu v lokálním minimu energetické funkce.

Idea důkazu:

♦ Počáteční stav

- Předložený vzor: $\vec{x} = (x_1, \dots, \underline{x_k}, \dots, x_n)$

Energetická funkce (3)

Idea důkazu (pokračování):

$$E(\vec{x}) = -\frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n w_{ij} x_i x_j$$

K aktualizaci vybrán neuron k

- k nezmění svůj stav $\rightarrow E(\vec{x})$ se nezmění
- k změní svůj stav $\rightarrow \vec{x}' = \left(x_1, \dots, \underline{x_k'}, \dots, x_n \right)$

Energetická funkce (4)

Idea důkazu (pokračování):

$$E(\vec{x}') = -\frac{1}{2} \sum_{\substack{j=1 \\ j \neq k}}^n \sum_{\substack{i=1 \\ i \neq k}}^n w_{ij} x_i x_j -$$

symetrie vah \swarrow

$$- \sum_{i=1}^n w_{ik} x_i x_k' = \left\{ \begin{array}{l} -\frac{1}{2} \sum_{j=1}^n w_{kj} x_k' x_j \\ -\frac{1}{2} \sum_{i=1}^n w_{ik} x_i x_k' \end{array} \right.$$

Energetická funkce (5)

Idea důkazu (pokračování):

♦ Rozdíl energií:

$$E(\vec{x}) - E(\vec{x}') = - \sum_{i=1}^n w_{ik} x_i x_k - \left(- \sum_{i=1}^n w_{ik} x_i x_k' \right) =$$
$$w_{kk} = 0 \longrightarrow = - \underbrace{\left(x_k - x_k' \right)}_{\text{různé znaménko (jinak by nedošlo ke změně stavu)}} \underbrace{\sum_{i=1}^n w_{ik} x_i}_{\text{POTENCIÁL}} > 0$$

různé znaménko (jinak by
nedošlo ke změně stavu)

Energetická funkce (6)

Idea důkazu (pokračování):

- Vždy, když dojde ke změně stavu neuronu, sníží se celková energie sítě
- ◆ Konečný počet možných stavů
 - Stabilní stav, kdy energii sítě už nelze snižovat

QED

Ekvivalence Hebbovského a perceptronového učení pro Hopfieldův model

V některých případech nelze nalézt pomocí Hebbovského učení váhovou matici Hopfieldovy sítě tak, aby m daných vektorů odpovídalo stabilním stavům sítě (i když taková matice existuje)

→ pokud leží vektory, které mají být do sítě uloženy, hodně blízko, může být „crosstalk“ příliš velký

→ **horší výsledky Hebbovského učení**

Alternativa: Perceptronové učení pro Hopfieldovy sítě

Ekvivalence Hebbovského a perceptronového učení pro Hopfieldův model (2)

Perceptronové učení pro Hopfieldovy sítě

- ♦ Hopfieldova síť s neurony, které mají nenulový práh a skokovou přenosovou funkci
 - Neuron má stav 1 , je-li potenciál větší než 0
 - Neuron má stav -1 , je-li potenciál menší nebo roven 0

Ekvivalence Hebbovského a perceptronového učení pro Hopfieldův model (3)

- ◆ Uvažujme Hopfieldovu síť:

n počet neuronů

$W = \{w_{ij}\}$ $n \times n$ matice vah

ϑ_i práh neuronu i

- ◆ Má-li si síť „zapamatovat“ vektor $\vec{x} = (x_1, \dots, x_n)$, bude tento vektor odpovídat stabilnímu stavu sítě tehdy, jestliže se po jeho „předložení“ stav sítě nezmění

Ekvivalence Hebbovského a perceptronového učení pro Hopfieldův model (4)

- **potenciál neuronu by měl mít stejné znaménko jako jeho předchozí stav**
- Nulovým hodnotám bude odpovídat záporné znaménko
 - **Měly by platit následující nerovnosti:**

$$\text{Neuron 1: } \operatorname{sgn}(x_1)(0 + x_2 w_{12} + \dots + x_n w_{1n} - \mathcal{I}_1) > 0$$

$$\text{Neuron 2: } \operatorname{sgn}(x_2)(x_1 w_{21} + 0 + \dots + x_n w_{2n} - \mathcal{I}_2) > 0$$

...

...

...

...

$$\text{Neuron n: } \operatorname{sgn}(x_n)(x_1 w_{n1} + x_2 w_{n2} + \dots + 0 - \mathcal{I}_1) > 0$$

Ekvivalence Hebbovského a perceptronového učení pro Hopfieldův model (5)

- ♦ $w_{ij} = w_{ji} \rightarrow n \cdot (n - 1) / 2$ nenulových prvků váhové matice a n prahů

\rightarrow necht' \vec{v} je vektor dimenze $n + n \cdot (n - 1) / 2$
(složky \vec{v} odpovídají prvkům w_{ij} nad diagonálou matice W ; $i < j$; a n prahům se záporným znaménkem)

$$\vec{v} = \left(\underbrace{w_{12}, w_{13}, \dots, w_{1n}}_{n-1 \text{ složek}}, \underbrace{w_{23}, w_{24}, \dots, w_{2n}}_{n-2 \text{ složek}}, \dots, \underbrace{w_{n-1,n}}_{1 \text{ složka}}, \underbrace{-\mathcal{I}_1, \dots, -\mathcal{I}_n}_{n \text{ složek}} \right)$$

Ekvivalence Hebbovského a perceptronového učení pro Hopfieldův model (6)

→ transformace vektoru \vec{x} do n „pomocných“
vektorů $\vec{z}_1, \vec{z}_2, \dots, \vec{z}_n$ dimenze $n + n \cdot (n - 1) / 2$:

$$\vec{z}_1 = \left(\underbrace{x_2, x_3, \dots, x_n}_{n-1 \text{ složek}}, \underbrace{0, 0, \dots, 0, 1, 0, \dots, 0}_n \right)$$

$$\vec{z}_2 = \left(\underbrace{x_1, 0, \dots, 0}_{n-1 \text{ složek}}, \underbrace{x_3, \dots, x_n}_{n-2 \text{ složek}}, \underbrace{0, 0, \dots, 0, 1, \dots, 0}_n \right)$$

.....

$$\vec{z}_n = \left(\underbrace{0, 0, \dots, x_1}_{n-1 \text{ složek}}, \underbrace{0, 0, \dots, x_2}_{n-2 \text{ složek}}, \underbrace{0, 0, \dots, 0, 0, \dots, 1}_n \right)$$

Ekvivalence Hebbovského a perceptronového učení pro Hopfieldův model (7)

složky vektorů $\vec{z}_1, \vec{z}_2, \dots, \vec{z}_n$ umožňují ekvivalentní zápis předchozích nerovností:

$$\text{Neuron 1 : } \text{sgn}(x_1) \vec{z}_1 \cdot \vec{v} > 0$$

$$\text{Neuron 2 : } \text{sgn}(x_2) \vec{z}_2 \cdot \vec{v} > 0$$

...

...

...

...

$$\text{Neuron n : } \text{sgn}(x_n) \vec{z}_n \cdot \vec{v} > 0$$

Ekvivalence Hebbovského a perceptronového učení pro Hopfieldův model (8)

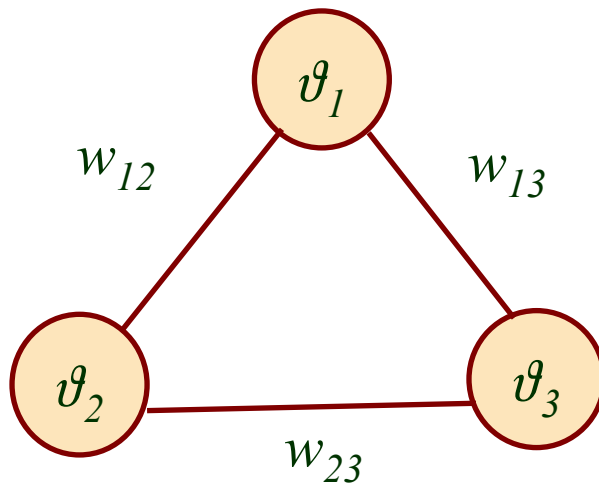
- k lineární separaci vektorů $\vec{z}_1, \vec{z}_2, \dots, \vec{z}_n$ (podle $\text{sgn}(x_i)$) lze použít perceptronové učení
- Spočítat vektor vah \vec{v} nutný pro lineární separaci $\vec{z}_1, \vec{z}_2, \dots, \vec{z}_n$ a nastavit váhovou matici W
- ◆ Pokud si má Hopfieldova síť „zapamatovat“ m vektorů $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_m$, je třeba použít popsanou transformaci pro každý vektor

Ekvivalence Hebbovského a perceptronového učení pro Hopfieldův model (9)

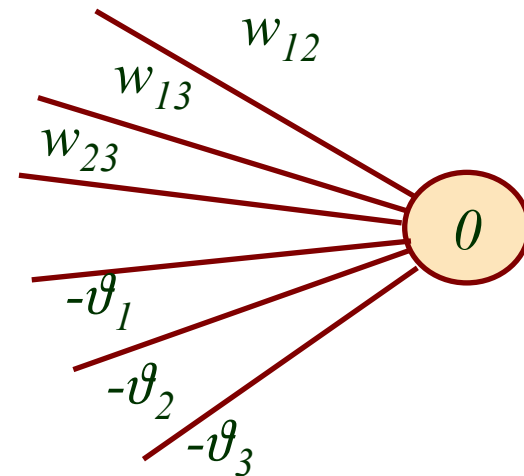
- $m \cdot n$ „pomocných“ vektorů, které je třeba lineárně odseparovat
- Pokud jsou vektory lineárně separabilní, najde perceptronové učení řešení „zakódované“ ve formě \vec{v}

Ekvivalence Hebbovského a perceptronového učení pro Hopfieldův model (10)

Příklad:



Učení Hopfieldovy sítě
s n neurony



Učení perceptronu s dimenzí
vstupního prostoru
 $n + n \cdot (n - 1) / 2$ ($= n \cdot (n + 1) / 2$)

Poznámka: „lokální aplikace delta-pravidla

Použití Hopfieldova modelu při řešení optimalizačních úloh

♦ Binární kódování: $0 / 1$

♦ Multiflop:

■ x_1, \dots, x_n ... Binární stavy jednotlivých neuronů Hopfieldovy sítě

■ Síť by se měla dostat do stavu, kdy bude právě 1 neuron aktivní; stav všech ostatních neuronů by měl být 0

■ Cíl: nalézt minimum funkce $E(x_1, \dots, x_n)$

$$E(x_1, \dots, x_n) = \left(\sum_{i=1}^n x_i - 1 \right)^2$$

Použití Hopfieldova modelu při řešení optimalizačních úloh (2)

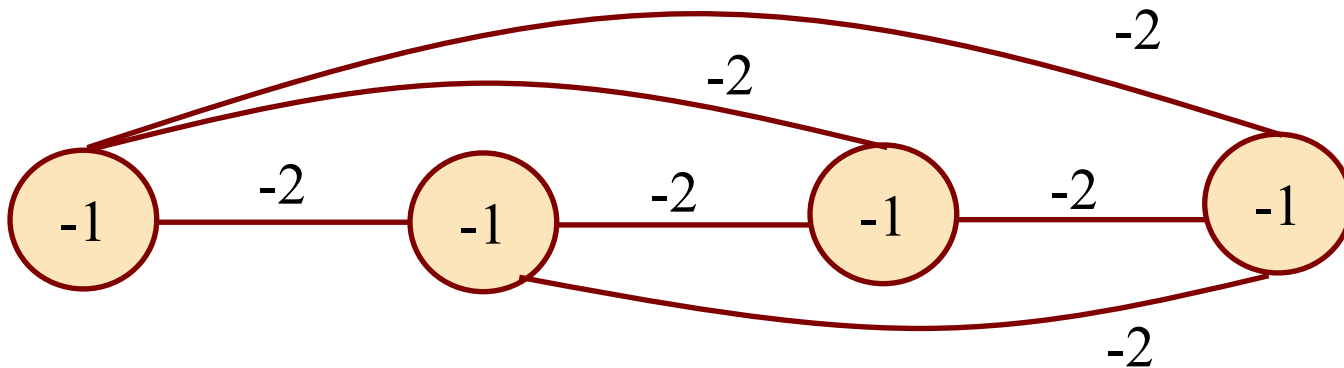
porovnání s
energetickou
funkcí
Hopfieldova
modelu

$$\begin{aligned} E(x_1, \dots, x_n) &= \left(\sum_{i=1}^n x_i - 1 \right)^2 = \sum_{i=1}^n x_i^2 + \sum_{i \neq j}^n x_i x_j - 2 \sum_{i=1}^n x_i + 1 = \\ &= \sum_{i \neq j}^n x_i x_j - \sum_{i=1}^n x_i + 1 = \quad = x_i \text{ pro binární stavy} \\ &= -\frac{1}{2} \sum_{i \neq j}^n (-2) x_i x_j + \sum_{i=1}^n (-1) x_i + 1 = \\ &\rightarrow \text{nastavení vah a prahů sítě} \end{aligned}$$

Použití Hopfieldova modelu při řešení optimalizačních úloh (3)

$$E(x_1, \dots, x_n) = -\frac{1}{2} \sum_{i \neq j}^n (-2) x_i x_j + \sum_{i=1}^n (-1) x_i + 1$$

nastavení vah a prahů sítě



Použití Hopfieldova modelu při řešení optimalizačních úloh (4)

♦ Problém n věží:

- Umístit na šachovnici $n \times n$, n věží tak, aby se navzájem neohrožovaly
 - každá věž by měla být v jiném řádku i sloupku než ostatní
 - x_{ij} stav neuronu na pozici ij šachovnice ($n \times n$)
 $\sum_{i=1}^n x_{ij}$ počet „stavů 1“ ve sloupci j
 - ✗ v každém sloupci by měla být jen jedna „jednička“

Použití Hopfieldova modelu při řešení optimalizačních úloh (5)

■ Minimalizace:

$$E_1(x_{11}, \dots, x_{nn}) = \sum_{j=1}^n \underbrace{\left(\sum_{i=1}^n x_{ij} - 1 \right)^2}_{\approx \text{MULTIFLOP}}$$

■ Podobně pro řádky:

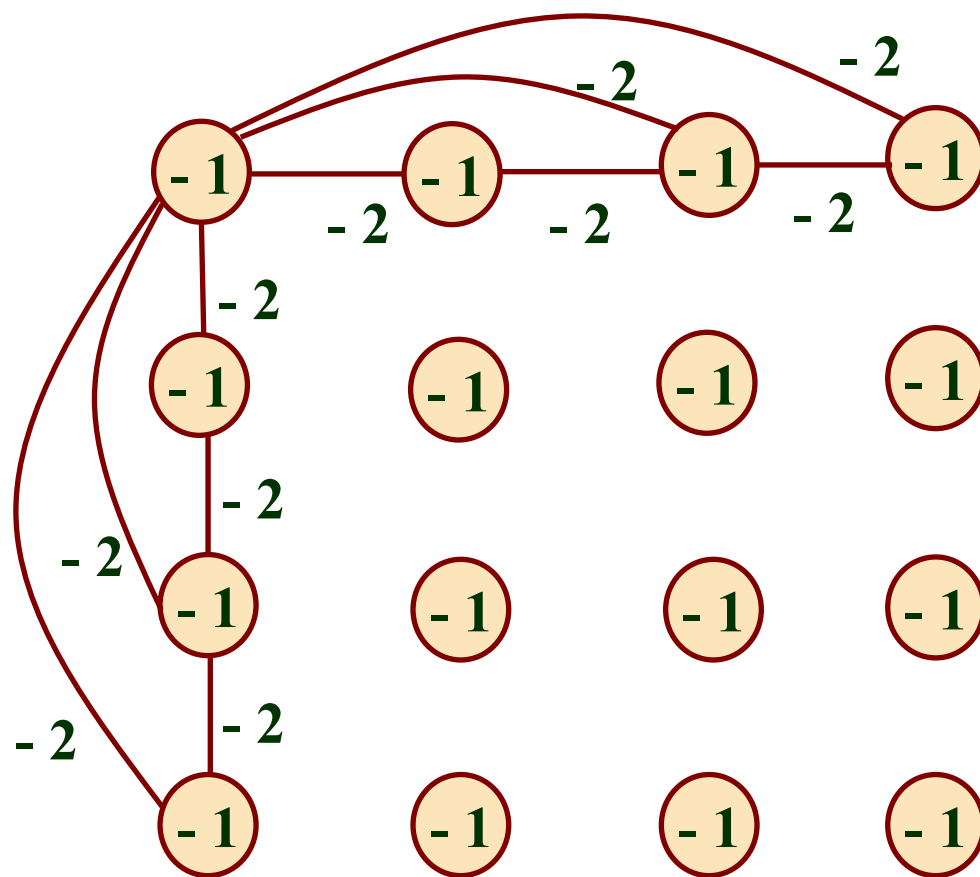
$$E_2(x_{11}, \dots, x_{nn}) = \sum_{i=1}^n \left(\sum_{j=1}^n x_{ij} - 1 \right)^2$$

■ Minimalizovat

$$E = E_1 + E_2$$

Použití Hopfieldova modelu při řešení optimalizačních úloh (6)

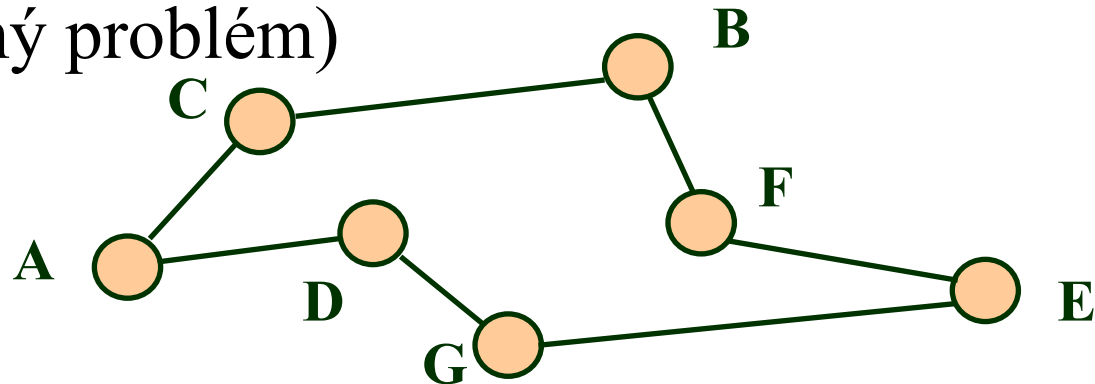
**Nastavení vah
a prahů sítě:**



Použití Hopfieldova modelu při řešení optimalizačních úloh (7)

◆ Problém obchodního cestujícího:

(~ NP-úplný problém)



- Nalézt cestu přes n měst M_1, \dots, M_n tak, aby bylo každé město navštíveno alespoň jednou a délka „okružní jízdy“ byla minimální

Použití Hopfieldova modelu při řešení optimalizačních úloh (8)

◆ Reprezentace pomocí matice:

		1	2	3	4
město	M_1	1	0	0	0
	M_2	0	1	0	0
„pořadí“ návštěv	M_3	0	0	1	0
	M_4	0	0	0	1

Konvence:

$(n+1)$ -ní sloupec
je stejný jako n
→ kružnice

- x_{ij} Stav neuronu ~ odpovídá údaji ik matice
- $x_{ik} = x_{jk+1} = 1$... Město M_i je navštíveno v k -tém kroku a M_j je navštíveno v $(k+1)$ -ním kroku
- d_{ij} Vzdálenost mezi M_i a M_j
→ d_{ij} přičíst k celkové délce cesty

Použití Hopfieldova modelu při řešení optimalizačních úloh (9)

♦ **Minimalizace délky cesty:** $L = \frac{1}{2} \sum_{i,j,k}^n d_{ij} x_{ik} x_{jk+1}$

× povolena jediná návštěva vždy jen jednoho města

→ **přidat požadavky na přípustnou cestu**

==> minimalizace E :

$$E = \frac{1}{2} \sum_{i,j,k}^n d_{ij} x_{ik} x_{jk+1} + \frac{\gamma}{2} \left(\sum_{j=1}^n \left(\sum_{i=1}^n x_{ij} - 1 \right)^2 + \sum_{i=1}^n \left(\sum_{j=1}^n x_{ij} - 1 \right)^2 \right)$$

Použití Hopfieldova modelu při řešení optimalizačních úloh (10)

♦ Nastavení vah a prahů sítě:

$$w_{ik,jk+1} = -d_{ij} + t_{ik,jk+1} \quad ; \quad \text{kde}$$

$$t_{ik,jk+1} = -\gamma$$

pro neurony ve stejné řádce či ve stejné sloupci

$$t_{ik,jk+1} = 0$$

jinak

$$\vartheta_{ij} = -\gamma / 2$$

Stochastické modely neuronových sítí

- ♦ Hopfieldův model se používá k řešení optimalizačních problémů, které lze vyjádřit ve formě minimalizované energetické funkce (i když není zaručeno nalezení globálního optima)
- ♦ **Problém:** zabránit „uvíznutí“ v lokálním minimu energetické funkce

Stochastické modely neuronových sítí (2)

Modifikace Hopfieldova modelu:

1. strategie: zvětšení počtu možných cest k řešení ve stavovém prostoru
→ dovolit i stavy ve formě reálných hodnot (sigmoidální přenosová funkce)
==> **spojitý model**
2. strategie: omezení lokálních minim energetické funkce pomocí „zašuměné dynamiky sítě“
→ dočasné povolení aktualizace stavu sítě i za cenu přechodného zvýšení energetické hladiny
==> **simulované žíhání, Boltzmannův stroj**

Spojité model Hopfieldovy sítě

- ♦ Aktivace neuronu i vybraného pro aktualizaci

podle:
$$x_i = s(u_i) = \frac{1}{1 + e^{-u_i}}$$

u_i označuje excitaci neuronu i

- ♦ Dodatečný předpoklad pomalé změny excitace neuronu v čase podle:

$$\frac{du_i}{dt} = \gamma \left(-u_i + \sum_{j=1}^n w_{ij} x_j \right) = \gamma \left(-u_i + \sum_{j=1}^n w_{ij} s(u_j) \right)$$

$\gamma > 0$ Adaptační parametr

w_{ij} váha mezi neuronem i a j

Spojité model Hopfieldovy sítě (2)

- ◆ Při simulacích spočítat diskrétní aproximaci du_i a přičíst ji k aktuální hodnotě u_i

výsledkem bude nový stav $x_i = s(u_i)$

- ◆ Asynchronní dynamika – vede k dosažení rovnovážného stavu
- ◆ **Energetická funkce pro spojitý model:**

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j + \sum_{i=1}^n \int_0^{x_i} s^{-1}(x) dx$$

Spojité model Hopfieldovy sítě (3)

- Po každé aktualizaci stavu neuronu se hodnota energetické funkce snižuje:

$$\frac{dE}{dt} = - \sum_{i=1}^n \sum_{j=1}^n w_{ij} \frac{dx_i}{dt} x_j + \sum_{i=1}^n s^{-1}(x_i) \frac{dx_i}{dt}$$

- Protože je síť symetrická, $w_{ij} = w_{ji}$
a zároveň $u_i = s^{-1}(x_i)$

$$\frac{dE}{dt} = - \sum_{i=1}^n \frac{dx_i}{dt} \left(\sum_{j=1}^n w_{ij} x_j - u_i \right)$$

Spojité model Hopfieldovy sítě (4)

♦ Přitom:

$$\frac{du_i}{dt} = \gamma \left(\sum_{j=1}^n w_{ij} x_j - u_i \right)$$
$$\Rightarrow \frac{dE}{dt} = - \frac{1}{\gamma} \sum_{i=1}^n \frac{dx_i}{dt} \frac{du_i}{dt}$$

♦ Protože $x_i = s(u_i)$

$$\frac{dE}{dt} = - \frac{1}{\gamma} \sum_{i=1}^n s^{-1}(u_i) \left(\frac{du_i}{dt} \right)^2$$

Spojité model Hopfieldovy sítě (5)

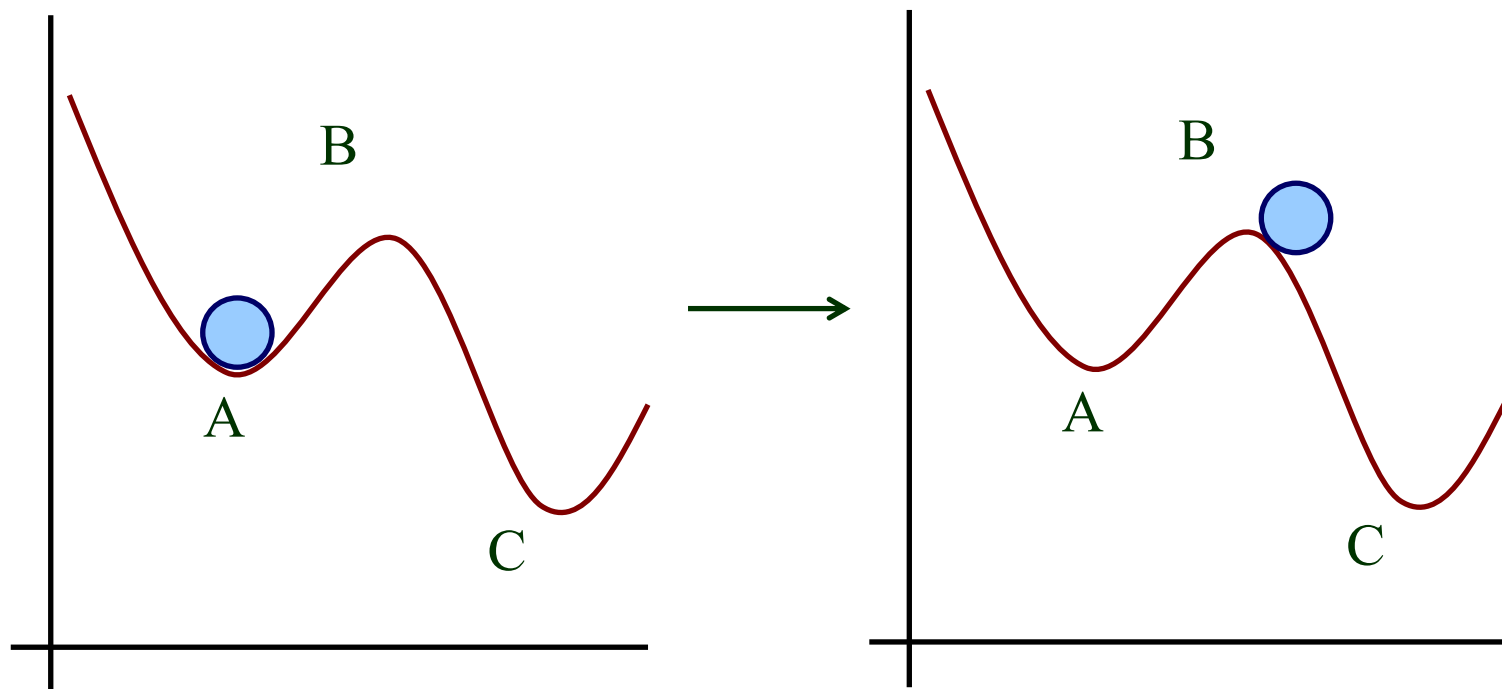
- ♦ Navíc platí $s^{-1}(x_i) > 0$
(sigmoida je monotonně rostoucí funkce)
- ♦ Protože $\gamma > 0$ $\frac{dE}{dt} \leq 0$
 - stabilního stavu síť dosáhne, pokud dE/dt „vymizí“
 - Taková situace nastane, pokud du_i/dt dosáhne „saturační oblasti“ sigmoidy, kde bude $du_i/dt \sim 0$

QED

Spojité model Hopfieldovy sítě (6)

- ◆ Pro kombinatorické problémy může spojitý model nalézt lepší řešení než model diskrétní
- ◆ Pro velmi složité problémy (typu TSP) ovšem spojitý model obecně nenachází výrazně lepší řešení

Simulované žíhání



Simulované žíhání (2)

- ◆ Při minimalizaci energetické funkce E se tento jev simuluje následujícím způsobem:
 - Hodnota proměnné \mathbf{x} se změní vždy, když může aktualizace $\Delta\mathbf{x}$ zmenšit hodnotu energetické funkce E
 - Pokud by se při aktualizaci \mathbf{x} naopak hodnota E zvýšila o ΔE , bude nová hodnota \mathbf{x} (tj. $\mathbf{x} + \Delta\mathbf{x}$) přijata s pravděpodobností $p_{\Delta E}$:

$$p_{\Delta E} = \frac{1}{1 + e^{\Delta E/T}}$$

kde T je tzv. teplotní konstanta

Simulované žíhání (3)

- ♦ Pro velké hodnoty T bude: $p_{\Delta E} \approx \frac{1}{2}$
a aktualizace stavu nastane zhruba v polovině těchto případů
- ♦ Pro $T = 0$ bude docházet pouze k takovým aktualizacím, kdy se hodnota E sníží
- ♦ **Postupná změna hodnot T z velmi vysokých hodnot směrem k nule odpovídá zahřátí a postupnému ochlazování v procesu žíhání**

Simulované žíhání (4)

- ◆ Navíc lze ukázat, že touto strategií lze dosáhnout (asymptoticky) globálního minima energetické funkce
- ◆ Sigmoida nejlépe odpovídá funkcím používaným v termodynamice (pro analýzu teplotní rovnováhy)

Boltzmannův stroj

Definice:

Boltzmannův stroj je Hopfieldova síť, která se skládá z n neuronů se stavy x_1, x_2, \dots, x_n .

Stav neuronu i se aktualizuje asynchronně podle pravidla:

$$x_i = \begin{cases} 1 & \text{s pstí } p_i \\ 0 & \text{s pstí } 1 - p_i \end{cases}$$

kde

$$p_i = \frac{1}{1 + e^{-\left(\sum_{j=1}^n w_{ij} x_j - \vartheta_i\right) / T}}$$

Boltzmannův stroj (2)

Ve vztahu:
$$p_i = \frac{1}{1 + e^{-\left(\sum_{j=1}^n w_{ij} x_j - \vartheta_i\right) / T}}$$

označuje T kladnou teplotní konstantu, w_{ij} váhy sítě a ϑ_i prahy neuronů

Energetická funkce Boltzmannova stroje:

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j + \sum_{i=1}^n \vartheta_i x_i$$

Boltzmannův stroj (3)

- ♦ Rozdíl mezi Boltzmannovým strojem a Hopfieldovým modelem spočívá ve stochastické aktivaci neuronů
- ♦ Pokud je T velmi malé, bude $p_i \sim 1$, jestliže je

$$\sum_{j=1}^n w_{ij} x_j - \mathcal{G}_i > 0$$

- × pokud je excitace neuronu záporná, bude $p_i \sim 0$

→ dynamika Boltzmannova stroje aproximuje dynamiku diskrétní Hopfieldovy sítě a Boltzmannův stroj najde lokální minimum energetické funkce

Boltzmannův stroj (4)

- ◆ Pro je $T > 0$ je pravděpodobnost změny anebo posloupnosti změn ze stavu $\mathbf{x}_1, \dots, \mathbf{x}_n$ do jiného stavu vždy nenulová
 - Boltzmannův stroj nezůstane v jediném stavu
 - snižování a zároveň možnost zvyšování energie systému
- ◆ Pro **veliké hodnoty** T projde síť téměř celý stavový prostor
 - ✗ V ochlazovací fázi má síť tendenci zůstat déle v oblastech blízkých atraktorům lokálních minim

Boltzmannův stroj (5)

Pokud se teplota snižuje správným způsobem, můžeme očekávat, že systém dosáhne globálního minima *s pravděpodobností 1*