

# Aplikace teorie neuronových sítí

Doc. RNDr. Iveta Mrázová, CSc.

Katedra teoretické informatiky

Matematicko-fyzikální fakulta

Univerzity Karlovy v Praze

# Aplikace teorie neuronových sítí

– Asociativní paměti a restaurace obrazu–

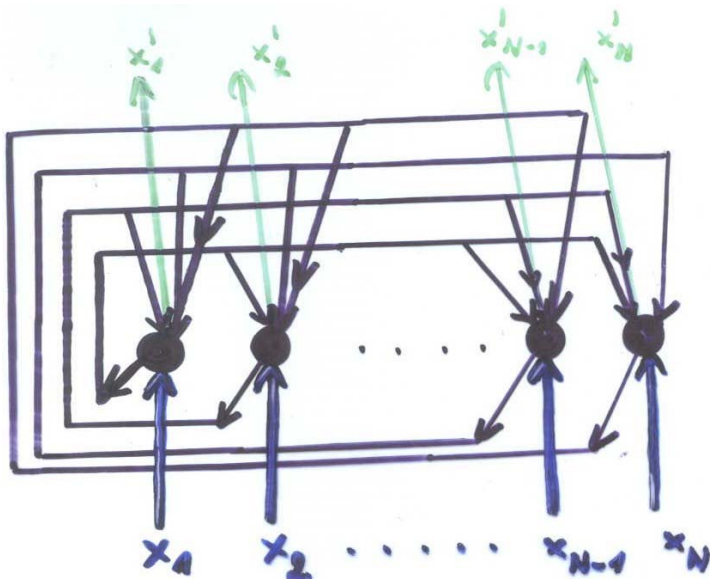
Doc. RNDr. Iveta Mrázová, CSc.

Katedra teoretické informatiky

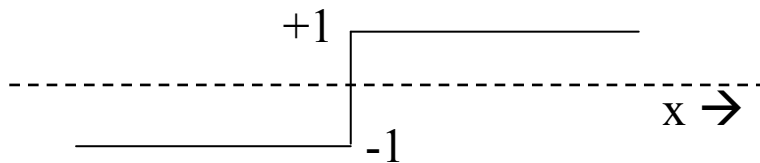
Matematicko-fyzikální fakulta

Univerzity Karlovy v Praze

# 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 - 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**

$(\Rightarrow \text{sgn}(\vec{\xi}) = \text{sgn}(\vec{x}_1))$

→ **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

# 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 (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***

# Použití Hopfieldova modelu při řešení optimalizačních úloh

- ◆ Binární kódování:  $0 / 1$
- ◆ **Multiflop:**
  - $x_1, \dots, x_n \dots$  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)

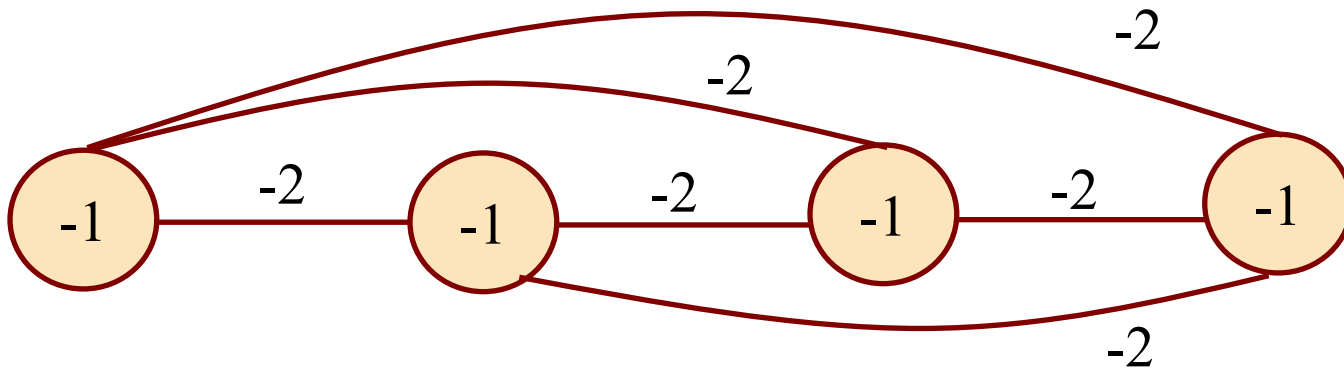
porovnání s energetickou funkcí Hopfieldova modelu

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

# 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**

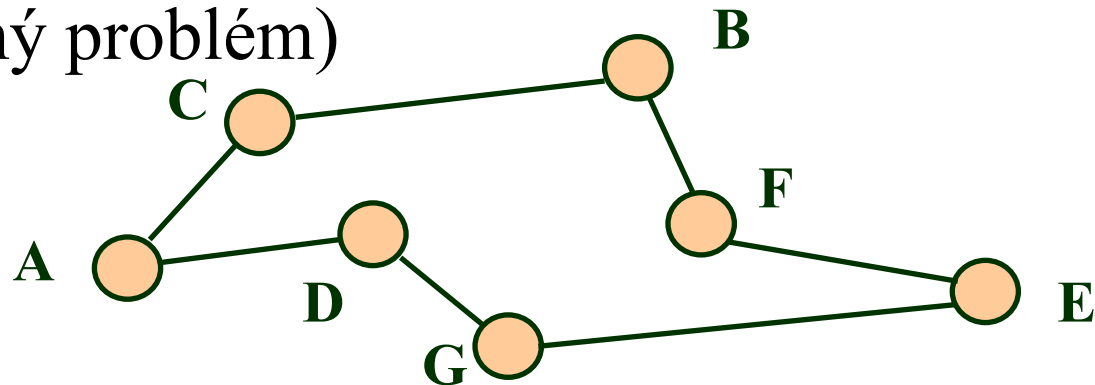




# Použití Hopfieldova modelu při řešení optimalizačních úloh (7)

## ◆ Problém obchodního cestujícího:

(~ NP-úplný problém)

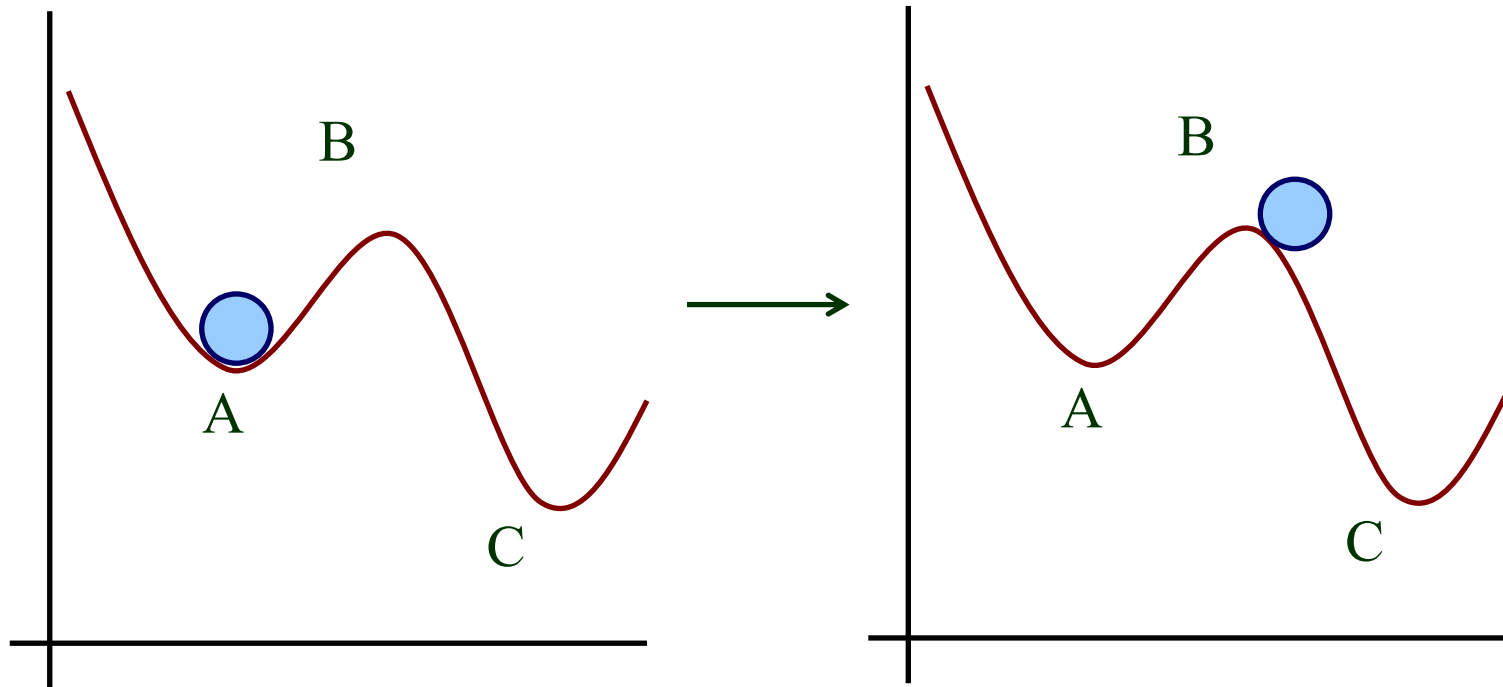


- Nalézt cestu přes  $n$  měst  $M_1, \dots, M_n$  tak, aby bylo každé město navštíveno alespoň jednou a délka „okružní jízdy“ byla minimální

# 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

# 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  $\Delta E$ , 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)

# Restaurace degradovaného obrazu

## Model degradace obrazu:

$$y(a, b) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(a, b; \alpha, \beta) x(\alpha, \beta) d\alpha d\beta + n(a, b)$$

$h(a, b; \alpha, \beta)$  ... rozmývací funkce

$n(a, b)$  ..... aditivní šum

$x(\alpha, \beta)$  ..... původní obraz

$y(a, b)$  ..... degradovaný obraz

- ◆ Je-li systém prostorově invariantní, lze rozmývací funkci vyjádřit jako:  $h(a, b; \alpha, \beta) = h(a - \alpha; b - \beta)$

# Restaurace degradovaného obrazu (2)

- ◆ Pokud lze vyjádřit rozmývací funkci  $h(\cdot)$  jako konvoluci pro malé okénko  $K \times K$  ( $K$  liché), lze spojitý model popsat diskrétně:

$$\begin{aligned} y(i, j) &= x(i, j) * h(i, j) + n(i, j) = \\ &= \sum_{k=-K'}^{K'} \sum_{l=-K'}^{K'} x(i-k, j-l) h(k, l) + n(i, j) \end{aligned}$$

\* ..... operátor konvoluce,  $K' = (K - 1) / 2$

$Y = HX + N$  ;      $H$  ..... rozmývací funkce

$N$  ..... aditivní šum

$X$  ..... původní obraz

$Y$  ..... degradovaný obraz



# Restaurace degradovaného obrazu (3)

## Reprezentace obrazu

- ◆ Neuronová síť obsahuje redundantní neurony pro reprezentaci různých úrovní šedi:

$L^2 \times M$  ..... vzájemně propojených neuronů

$L$  ..... velikost obrazu

$M$  ..... nejvyšší hodnota šedi

- ◆ Obraz je popsán konečnou množinou úrovní šedi:

$$\{x(i, j); 1 \leq i, j \leq L\}$$

$x(i, j)$  ..... celé kladné číslo – značí hodnotu šedi pixlu  $(i, j)$

# Restaurace degradovaného obrazu (4)

## Reprezentace obrazu (pokračování):

- ◆ Necht'  $V = \{ v_{i,k} ; 1 \leq i \leq L^2, 1 \leq k \leq M \}$  je binární množina stavů neuronové sítě  
 $v_{i,k}$  značí stav neuronu  $(i, k)$
- ◆ Funkce šedi je pro obraz reprezentována součtem stavových proměnných neuronu:  $x(i, j) = \sum_{k=1}^M v_{m,k} ;$  kde  $m = (i-1) \times L + j$
- ◆ Síť s takovou reprezentací má:  $\prod_{i=1}^L \prod_{j=1}^L \frac{M!}{x(i, j)! (M - x(i, j))!}$   
stabilních stavů pro obraz  $L \times L$

# Restaurace degradovaného obrazu (5)

## Reprezentace obrazu (pokračování):

- Výhody:
- ) Jeden neuron s opačným výstupem nezpůsobí velké škody v číselné reprezentaci
  - ) Více šancí nalézt správné řešení
  - ) Rychlejší konvergence

- ♦ Každý neuron  $(i, k)$  v tomto modelu dostává náhodně a asynchronně vstupy ode všech neuronů a prahový vstup:

$$u_{i,k} = \sum_{j=1}^{L^2} \sum_{l=1}^M T_{i,k;j,l} v_{j,l} + I_{i,k}$$

$T_{i,k;j,l}$  představuje váhu synapse (i zápornou) mezi neurony  $(i,k)$  a  $(j,l)$

$I_{i,k}$  prahový vstup

# Restaurace degradovaného obrazu (6)

## Reprezentace obrazu (pokračování):

### Předpoklady:

- ♦ Váhy jsou symetrické a neurony mají zpětnou vazbu (self-feedback):  $T_{i,k;j,l} = T_{j,l;i,k}$  a  $T_{i,k;i,k} \neq 0$
- ♦ Dále pro každý neuron:  $v_{i,k} = g(u_{i,k})$

kde  $g(x)$  je nelineární funkce: 
$$g(x) = \begin{cases} 1 & \text{pokud } x \geq 0 \\ 0 & \text{pokud } x < 0 \end{cases}$$

- ♦ Stav každého neuronu se aktualizuje pomocí aktuální informace od ostatních neuronů

# Restaurace degradovaného obrazu (7)

## Odhad parametrů modelu:

- ◆ Pomocí energetické funkce

$$E = -\frac{1}{2} \sum_{i=1}^{L^2} \sum_{j=1}^{L^2} \sum_{k=1}^M \sum_{l=1}^M T_{i,k;j,l} v_{i,k} v_{j,l} - \sum_{i=1}^{L^2} \sum_{k=1}^M I_{i,k} v_{i,k}$$

a minimalizace chybové funkce definované jako:

$$E = \frac{1}{2} \left\| Y - H \hat{X} \right\|^2 + \frac{1}{2} \left\| D \hat{X} \right\|^2$$

kde  $\|Z\|$  je  $L_2$ -norma,  $\lambda$  je konstanta

~ jiná formulace restauračního problému

# Restaurace degradovaného obrazu (8)

## Odhad parametrů modelu (pokračování):

$$\begin{aligned} \text{Potom: } E &= \frac{1}{2} \sum_{p=1}^{L^2} \left( y_p - \sum_{i=1}^{L^2} h_{p,i} x_i \right)^2 + \frac{1}{2} \lambda \sum_{p=1}^{L^2} \left( \sum_{i=1}^{L^2} d_{p,i} x_i \right)^2 = \\ &= \frac{1}{2} \sum_{i=1}^{L^2} \sum_{j=1}^{L^2} \sum_{k=1}^M \sum_{l=1}^M \sum_{p=1}^{L^2} h_{p,i} h_{p,j} v_{i,k} v_{j,l} + \\ &+ \frac{1}{2} \lambda \sum_{i=1}^{L^2} \sum_{j=1}^{L^2} \sum_{k=1}^M \sum_{l=1}^M \sum_{p=1}^{L^2} d_{p,i} d_{p,j} v_{i,k} v_{j,l} - \\ &- \sum_{i=1}^{L^2} \sum_{k=1}^M \sum_{p=1}^{L^2} y_p h_{p,i} v_{i,k} + \frac{1}{2} \sum_{p=1}^{L^2} y_p^2 \end{aligned}$$

zanedbat

# Restaurace degradovaného obrazu (9)

## Odhad parametrů modelu (pokračování):

Porovnáním s koeficienty energetické funkce dostaneme:

$$T_{i,k;j,l} = - \sum_{i=1}^{L^2} h_{p,i} h_{p,j} - \lambda \sum_{i=1}^{L^2} d_{p,i} d_{p,j}$$

$$I_{i,k} = \sum_{p=1}^{L^2} y_p h_{p,i}$$

# Restaurace degradovaného obrazu (10)

## Odhad parametrů modelu (pokračování):

### Zajímavé aspekty:

- ◆ Váhy nezávisí na  $k$  a  $l$
- ◆ Prahy nezávisí na  $k$
- ◆  $T_{i,k;i,k} \neq 0 \rightarrow$  nutný feedback
- ◆ Váhy určuje rozmývací funkce, operátor  $D$  a konstanta  $\lambda$   
 $\rightarrow T_{i,k;j,l}$  lze spočítat bez chyby za předpokladu, že známe rozmývací funkci



# Restaurace degradovaného obrazu (11)

## Zajímavé aspekty (pokračování):

- ♦ Prahy jsou funkcemi degradovaného obrazu  
→ je-li obraz degradován pouze prostorově invariantní rozmývací funkcí, lze určit  $I_{i,k}$ , jinak je práh  $I_{i,k}$  ovlivněn šumem

$$\begin{aligned}y_p &= \sum_{i=1}^{L^2} h_{p,i} x_i + n_p \\I_{i,k} &= \sum_{p=1}^{L^2} \left( \sum_{i=1}^{L^2} h_{p,i} x_i + n_p \right) h_{p,i} = \\&= \sum_{p=1}^{L^2} \sum_{i=1}^{L^2} h_{p,i} x_i h_{p,i} + \sum_{p=1}^{L^2} \underbrace{n_p}_{\text{šum}} h_{p,i}\end{aligned}$$

# Restaurace degradovaného obrazu (12)

## Restaurace:

- ◆ Provádí se vyhodnocováním neuronů a konstrukcí obrazu
- ◆ Jsou-li již známy parametry  $T_{i,k;j,l}$  a  $I_{i,k}$ , může každý neuron náhodně a asynchronně vyhodnotit svůj stav a aktualizovat ho
- ◆ Pokud bylo dosaženo quasi-optimálního energetického bodu,

lze konstruovat obraz: 
$$x(i, j) = \sum_{k=1}^M v_{m,k}$$

- ◆  $T_{i,k;j,l} \neq 0$  ( $\sim$  self-feedback)  
→ není zaručena konvergence sítě

# Restaurace degradovaného obrazu (13)

## Restaurace (pokračování):

- ◆ Návrh deterministického nebo stochastického rozhodovacího pravidla (podle toho, zda požadujeme konvergenci k lokálnímu nebo globálnímu minimu)
- ◆ **Deterministické pravidlo:**
  - Je-li změna energie  $\Delta E$  vzhledem ke změně stavu  $\Delta v_{i,k}$  menší než nula  $\Rightarrow$  nový stav neuronu  $(i, k)$ :  $v_{i,k}^{new}$
  - Je-li  $\Delta E$  vzhledem ke změně stavu větší než nula, nedojde ke změně stavu

# Restaurace degradovaného obrazu (14)

## Restaurace (pokračování):

### ◆ Stochastické pravidlo:

- Podobné jako u technik simulovaného žíhání
- Pravděpodobnost přijetí:

$$P_T \left\{ \text{accept} \left( v_{1,1}, \dots, v_{i,k}^{new}, \dots, v_{L^2M} \right) \right\} = \begin{cases} 1 & \Delta E < 0 \\ e^{-\Delta E/T} & \Delta E \geq 0 \end{cases}$$

$\Delta E$  je změna energie,  $T$  je parametr ( $\sim$  teplota)

- Nový stav  $v_{i,k}^{new}$ , jestliže  $P_T \left\{ \text{accept} \left( v_{1,1}, \dots, v_{i,k}^{new}, \dots, v_{L^2M} \right) \right\} = 1$
- Rychlost konvergence algoritmu závisí na volbě teploty  $T$

# Restaurace degradovaného obrazu (15)

## Restaurační algoritmus:

Krok 1: Inicializace stavu neuronů

Krok 2: Aktualizace stavu všech neuronů – náhodně, asynchronně podle rozhodovacího pravidla

Krok 3: Pokud se hodnota energetické funkce nemění, proved' Krok 4; jinak proved' Krok 2

Krok 4: Konstrukce obrazu pomocí: 
$$x(i, j) = \sum_{k=1}^M v_{m,k}$$

# Restaurace degradovaného obrazu (16)

## Restaurační algoritmus (pokračování):

Obtížná simulace na běžném počítači

- Rozměry obrázku:  $L \times L$  pixelů,  $M$  úrovní šedi
- $L^2 M$  neuronů a  $\frac{1}{2} L^4 M^2$  synaptických vah
- $L^4 M^2$  operací (sčítání, násobení) v každé iteraci
- Prostorová složitost:  $O(L^4 M^2)$
- Časová složitost:  $O(L^4 M^2 K)$ ,  $K \sim$  počet iterací
- Obvykle:  $10 \leq K \leq 100$ ,  $256 \leq L \leq 1024$ ,  $M = 256$

# Restaurace degradovaného obrazu (17)

## Restaurační algoritmus (pokračování):

- ◆ Možné zjednodušení za předpokladu, že se neurony aktualizují postupně:

$$u_{i,k} = \sum_{j=1}^{L^2} \sum_{l=1}^M T_{i,k;j,l} v_{j,l} + I_{i,k}$$

$$v_{i,k} = g(u_{i,k}) = \begin{cases} 1 & \text{pokud } u_{i,k} \geq 0 \\ 0 & \text{pokud } u_{i,k} < 0 \end{cases}$$

# Restaurace degradovaného obrazu (18)

## Možné zjednodušení (pokračování):

- ♦ Váhy nezávisí na  $k$  a  $l$ , prahy nezávisí na  $k$

$$T_{i,k;j,l} = - \sum_{p=1}^{L^2} h_{p,i} h_{p,j} - \lambda \sum_{p=1}^{L^2} d_{p,i} d_{p,j}$$

$$I_{i,k} = \sum_{p=1}^{L^2} y_p h_{p,i}$$



# Restaurace degradovaného obrazu (19)

## Možné zjednodušení (pokračování):

- ♦  $M$  neuronů pro reprezentaci úrovně šedi má stejné váhy a prahy  $\rightarrow$  redukce dimenze váhové matice o člen  $M^2$

$$\begin{aligned} u_{i,k} &= \sum_{j=1}^{L^2} T_{i,\cdot; j,\cdot} \left( \sum_{l=1}^M v_{j,l} \right) + I_{i,\cdot} = \\ &= \sum_{j=1}^{L^2} T_{i,\cdot; j,\cdot} x_j + I_{i,\cdot} \end{aligned}$$

$$\left( x(i, j) = \sum_{k=1}^M v_{m,k} ; \quad m = (i-1) \times L + j \right)$$

# Restaurace degradovaného obrazu (20)

## Možné zjednodušení (pokračování):

- ◆ Váhy jsou dány pomocí rozmývací funkce  $H$ , konstanty  $\lambda$  a operátoru  $D$ 
  - neurony jsou lokálně propojené a většina vah bude nulových
- ◆ Je-li rozmývací funkce prostorově invariantní a neuvažují-li se okraje obrázku, je třeba uchovávat mnohem méně údajů:
  - Stav  $(i, k)$  – tého neuronu se aktualizuje podle hodnoty vstupu  $u_{i,k}$ , a to pomocí rozhodovacího pravidla

# Restaurace degradovaného obrazu (21)

## Možné zjednodušení (pokračování):

- Změna stavu  $(i, k)$  – tého neuronu vyvolá změnu úrovně šedi  $x_i$ :

$$x_i^{new} = \begin{cases} x_i^{old} & \text{pokud } \Delta v_{i,k} = 0 \\ x_i^{old} + 1 & \text{pokud } \Delta v_{i,k} = 1 \\ x_i^{old} - 1 & \text{pokud } \Delta v_{i,k} = -1 \end{cases}$$

kde  $\Delta v_{i,k} = v_{i,k}^{new} - v_{i,k}^{old}$  je změna stavu neuronu  $(i, k)$

$x_i$  označuje úroveň šedi ( $\sim$  reprezentovanou výstupem  $M$  neuronů)

# Restaurace degradovaného obrazu (22)

## Možné zjednodušení (pokračování):

- Jednotlivé neurony „pracují“ postupně  
→ jiná formulace procesu aktualizace

$$u_{i,k} = \sum_{j=1}^{L^2} T_{i,\cdot; j,\cdot} x_j + I_{i,\cdot}$$

$$\Delta v_{i,k} = g(u_{i,k}) = \begin{cases} \Delta v_{i,k} = 0 & \text{pokud } u_{i,k} = 0 \\ \Delta v_{i,k} = 1 & \text{pokud } u_{i,k} > 0 \\ \Delta v_{i,k} = -1 & \text{pokud } u_{i,k} < 0 \end{cases}$$

# Restaurace degradovaného obrazu (23)

## Možné zjednodušení (pokračování):

$$x_i^{new} = \begin{cases} x_i^{old} + \Delta v_{i,k} & \text{pokud } \Delta E < 0 \\ x_i^{old} & \text{pokud } \Delta E \geq 0 \end{cases}$$

→ lze použít i stochastické rozhodovací pravidlo

- Kontrola hodnot úrovně šedi v každém kroku adaptace:  $0 \leq x_i^{new} \leq 255$

# Restaurace degradovaného obrazu (24)

## Restaurační algoritmus – zjednodušení:

Krok 1: Inicializace – pomocí degradovaného obrázku

Krok 2: Postupně projdi všechny pixly:

- ◆ Pro každý pixel proved' aktualizaci pomocí výše uvedených vztahů
- ◆ Pokud nedojde k žádné změně ( $\sim$  jestliže  $\Delta v_{i,k} = 0$  nebo  $\Delta E \geq 0$ ), přejdi k dalšímu pixlu

Krok 3: Zkontroluj hodnotu energetické funkce

- ◆ Pokud nedošlo k žádné změně, obrázek byl zrestaurován
- ◆ V opačném případě přejdi ke Kroku 2 a proved' další iteraci

# Restaurace degradovaného obrazu (25)

## Restaurační algoritmus – zjednodušení (pokračování):

- ◆ Výpočet vstupu  $u_{i,k}$  ( $i, k$ ) – tého neuronu a změny hodnoty energetické funkce  $\Delta E$  lze zjednodušit:
  - Provádíme-li opakovaně aktualizaci úrovně šedi pro příslušný pixel, lze využít předchozích hodnot:

$$u_{i,k} = u_{i,k-1} + \Delta v_{i,k-1} T_{i,\cdot; i,\cdot}$$

kde  $u_{i,k-1}$  je vstup ( $i, k-1$ ) – ního neuronu

$$\Delta E = -u_{i,k} \Delta v_{i,k} - \frac{1}{2} T_{i,\cdot; i,\cdot} (\Delta v_{i,k})^2$$

# Restaurace degradovaného obrazu (26)

## Redukce složitosti:

- ◆ Prostorová:  $O(L^4 M^2) \rightarrow O(L^2)$
- ◆ Výpočetní:  $O(L^4 M^2 K) \rightarrow O(M L^2 K)$
- ◆ Pro každý pixel je třeba jen několik málo aktualizací hodnoty šedi  
→ na každou iteraci:  $O(L^2)$