Asociativní sítě (paměti)

Cíl učení

- Asociace známého vstupního vzoru s daným výstupním vzorem
- Okolí známého vstupního vzoru $\vec{x}$ by se mělo také zobrazit na výstup $\vec{y}$ odpovídající $\vec{x}$
  $\rightarrow$ správný výstup pak lze přiřadit i „zašuměným" vzorům
- **Hebbovské učení**

Typická funkce

- Rozpoznat předem naučené vstupní vzory i v případě, že jsou neúplné nebo mírně poškozené

Asociativní sítě (paměti)

Základní typy asociativních sítí

- heteroasociativní
  - Zobrazujeme N vstupních vzorů $\vec{x}^1, \dots, \vec{x}^N$ v n-rozměrném prostoru na N výstupních vzorů $\vec{y}^1, \dots, \vec{y}^N$ v m-rozměrném prostoru tak, že
  $$\vec{x}^i \mapsto \vec{y}^i$$
  Jestliže $\|\vec{x} - \vec{x}^i\| < \epsilon$, pak
  $$\vec{x} \mapsto \vec{y}^i$$
- autoasociativní - oprava poškozených vzorů
- sítě pro rozpoznávání vzorů - identifikace třídy vstupního vzoru

Asociativní sítě (paměti)

Základní typy asociativních sítí

- heteroasociativní
- autoasociativní
  - $\vec{x}^i = \vec{y}^i$ pro $i = 1, \ldots, N$
  - oprava poškozených vzorů
- sítě pro rozpoznávání vzorů
  - Lze je pokládat za specílní typ heteroasociativní paměti
  - Každému vektoru $\vec{x}^i$ je přiřazena třída i pro $i = 1, \ldots, N$
  - identifikace třídy vstupního vzoru

Struktura asociativní paměti

Dva základní typy architektury asociativních pamětí:

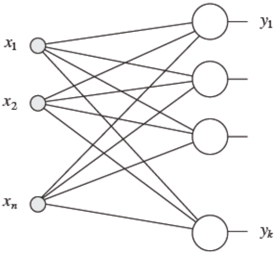
- asociativní sítě bez zpětné vazby
- asociativní sítě se zpětnou vazbou

Struktura asociativní paměti bez zpětné vazby

Asociativní paměť lze implementovat pomocí jedné vrstvy neuronů

Parametry

- w_{ij} ... váha mezi vstupem i a neuronem j
- W tvaru $n \times m$ ... matice vah, n je počet vstupů, m je počet výstupů



Struktura asociativní paměti bez zpětné vazby

Asociativní paměť lze implementovat pomocí jedné vrstvy neuronů

Výpočet

- vstupní vektor $\vec{x} = \{x_1, ..., x_n\}$ dává excitační vektor $\vec{e} = \vec{x}.W$
- Potom se pro každý neuron spočítá jeho výstup y_i jako hodnota přenosové funkce $f(e_i), i = 1, ..., m$
  - Pro identitu dostáváme lineární asociátor a výstupem bude právě $\vec{y} = \vec{x}.W$

Struktura asociativní paměti bez zpětné vazby

Obecně

- X ... matice vstupů tvaru $N \times n$ (řádky odpovídají vstupním vektorům)
- Y ... matice výstupů tvaru $N \times m$ (řádky odpovídají výstupním vektorům)

Cíl učení (pro přenosovou funkci identita)

- Najít takovou matici vah W, aby $XW = Y$ (obecně $f(XW) = Y$)
  - Pro autoasociativní síť: $XW = X$
  - Pro $n = m$: $W = X^{-1}Y$ ... řešení soustavy lineárních rovnic

Struktura asociativní sítě se zpětnou vazbou

Nejjednodušší zpětná vazba:

- výstup sítě se používá opakovaně jako její nový vstup, dokud síť nezkonverguje do stabilního stavu
- ne všechny sítě zkonvergují po předložení nového vzoru do stabilního stavu
  $\rightarrow$ nutná dodatečná omezení na architekturu sítě
- učení je rychlé, ale rozpoznávání je obtížnější

Asociativní sítě se zpětnou vazbou

Nejjednodušší asociativní sítě – s přenosovou funkcí identita

- bez zpětné vazby ... Lineární asociátory ... $XW = Y$
- se zpětnou vazbou ... Vlastní automaty ... $XW = X$

Vlastní automaty (eigenvector automata)

Lineární rekurentní asociativní síť

Výstup sítě představuje její nový vstup:

- všechny neurony počítají svůj výstup současně
- síť dostává v každém kroku (t) na vstup vektor $\vec{x}^t$ a dává nový výstup $\vec{x}^{t+1}$

Otázka

- Existuje pevný bod $\vec{x}$ takový, že $\vec{x} = \vec{x}.W$?
  ... vektor $\vec{x}$ je vlastním vektorem matice W s vlastním číslem 1
- síť se chová jako dynamický systém prvního řádu (každý nový stav je plně určen nejbližším předchůdcem)
  $\rightarrow$ hledáme pevné body dynamického systému

Vlastní automaty (eigenvector automata)

- Matice vah W tvaru $n \times n$ může mít až n lineárně nezávislých vlastních vektorů $\vec{x}^i$ a vlastních čísel λ_i.

$$\vec{x}^i W = \lambda_i \vec{x}^i \text{ pro } i = 1, ..., n$$

  $\rightarrow$ pak W definuje tzv. vlastní automat
- Po předložení libovolného vstupního vzoru bude nalezen vlastní vektor s největším vlastním číslem (pokud existuje) ... **atraktor**
  - přibližuje se směr, ne velikost
  - daná složka vektoru musí být nenulová

Vlastní automaty (eigenvector automata)

Příklad 1

- Matice $W = [2\ 0; 0\ 1]$ má 2 vlastní vektory (1,0) a (0,1) s vlastními čísli 2 a 1
- Po t iteracích počátečního vektoru $(x_1, x_2), x_1 \neq 0$ dostaneme $(2^t x_1, x_2)$ ... konverguje (co se týká směru) k $(1, 0)$

Příklad 2

- Matice $W = [0\ 1; -1\ 0]$ nekonverguje do stabilního stavu (rotace o 90%) ... cykly délky 4.

Učení asociativních sítí

Co je cílem učení?

- Použití asociativních sítí jako dynamických systémů, jejichž atraktory by odpovídaly těm vektorům, které chceme do paměti uložit
- Při návrhu sítě chceme rozmístit ve vstupním prostoru co možná nejvíce atraktorů.
- Každý atraktor by měl mít přesně danou a omezenou oblast vlivu
  (vlastní automaty – oblast vlivu jedinného atraktoru zahrnovala téměř celý vstupní prostor)

Učení asociativních sítí

- Chceme co nejvíce atraktorů s přesně danou a omezenou oblast vlivu
  - použijeme nelineární přenosovou funkci, např. skokovou:

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

- Biparitní kódování je vhodnější než binární (protože je větší pravděpodobnost vzájemné ortogonality vektorů)

Hebbovské učení (Donald Hebb – 1949)

Vstup

- Neuronová síť s jednou vrstvou a m neurony a skokovou přenosovou funkcí.

Cíl

- Nalézt odpovídající váhy pro zobrazení n–rozměrného vstupního vektoru $\vec{x}$ na m–rozměrný výstupní vektor $\vec{y}$ (obecně N vzorů)

Idea

- Dva neurony, které jsou současně aktivní, by měly mít ,,vyšší stupeň vzájemné interakce" než neurony, jejichž aktivita je nekorelovaná – v takovém případě by měla být vzájemná interakce hodně malá nebo nulová

Hebbovské učení

Adaptační pravidlo

- na vstupu je n–rozměrný vektor $\vec{x}^1$, na výstupu m–rozměrný vektor $\vec{y}^1$
- Aktualizace váhy w_{ij} ze vstupu i do neuronu j podle:

$$w_{ij} = w_{ij} + \triangle w_{ij},$$

kde

$$\triangle w_{ij} = \alpha x_i y_j$$

  - α ... parametr učení
  - W ... váhová matice (na začátku učení nulová)

$\rightarrow$ adaptovaná váhová matice W je **korelační maticí** pro vektory $\vec{x}^1$ a $\vec{y}^1$:

$$W = [w_{ij}]_{n\times m} = [\alpha x_i^1 y_j^1]_{n\times m} = \alpha \vec{x}^{1\,T} \vec{y}^1$$

Hebbovské učení

Matice $W = [x_i^1 y_j^1]_{n \times m}$ zobrazí nenulový vektor $\vec{x}^1$ právě na vektor $\vec{y}^1$:

- Spočteme excitační vektor:

$$\vec{x}^1 W = (y_1^1 \sum_{i=1}^{n} x_i^1 x_i^1, y_2^1 \sum_{i=1}^{n} x_i^1 x_i^1, \dots, y_m^1 \sum_{i=1}^{n} x_i^1 x_i^1) = \vec{y}^1 (\vec{x}^1 \cdot \vec{x}^1)$$

- Pro $\vec{x}^1 \neq \vec{0}$ platí $(\vec{x}^1 \cdot \vec{x}^1) > 0$ a výstup sítě je:

$$sgn(\vec{x}^1 W) = (y_1^1, \dots, y_m^1) = \vec{y}^1$$

- Pro $(-\vec{x}^1)$ je výstup sítě:

$$sgn(-\vec{x}^1 W) = -sgn(\vec{x}^1 W) = -\vec{y}^1$$

Hebbovské učení

Obecně:

- na vstupu je N n–rozměrných vektorů $\vec{x}^1, \ldots, \vec{x}^N$, na výstupu N m–rozměrných vektorů $\vec{y}^1, \ldots, \vec{y}^N$
- Použijeme Hebbovské učení pro každou dvojici $\vec{x}^l$ a $\vec{y}^l$
- Výsledná matice vah je $W = W^1 + \ldots + W^N$, kde $W^l = [x_i^l y_j^l]_{n \times m}$ je korelační matice $\vec{x}^l$ a $\vec{y}^l$
- Váhy se nastaví v jednom průchodu

Hebbovské učení

Předložíme vzor $\vec{x}^p$.

- Excitační vektor:

$$\begin{aligned}\vec{x}^p W &= \vec{x}^p (W^1 + \ldots + W^N) \\ &= \vec{x}^p W^p + \sum_{l \neq p}^{N} \vec{x}^p W^l \\ &= \vec{y}^p (\vec{x}^p \cdot \vec{x}^p) + \sum_{l \neq p}^{N} \vec{y}^l (\vec{x}^l \cdot \vec{x}^p)\end{aligned}$$

- excitační vektor tedy odpovídá $\vec{y}^p$ (vynásobenému kladnou konstantou) s perturbačním členem (**crosstalk**) $\sum_{l \neq p}^{N} \vec{y}^p (\vec{x}^l \cdot \vec{x}^p)$

Hebbovské učení

Výstup sítě bude roven:

$$sgn(\vec{x}^p W) = sgn\left(\vec{y}^p(\vec{x}^p \cdot \vec{x}^p) + \sum_{l \neq p}^{N} \vec{y}^l(\vec{x}^l \cdot \vec{x}^p)\right)$$

- Protože $(\vec{x}^p \cdot \vec{x}^p)$ je kladná konstanta:

$$sgn(\vec{x}^p W) = sgn\left(\vec{y}^p + \sum_{l \neq p}^{N} \vec{y}^l \frac{(\vec{x}^l \cdot \vec{x}^p)}{(\vec{x}^p \cdot \vec{x}^p)}\right)$$

Hebbovské učení

Kdy dá síť na výstupu požadovaný vektor $\vec{y}^p$?

- když je crosstalk nulový
  $\rightarrow$ pokud jsou vstupní vzory navzájem ortogonální
- když je crosstalk „menší" než $\vec{y}^p(\vec{x}^p \cdot \vec{x}^p)$

Hebbovské učení

- Aby byl výstup sítě roven $\vec{y}^p$, musí platit

$$\vec{y}^p = sgn\left(\vec{y}^p + \sum_{l \neq p}^{N} \vec{y}^l \frac{(\vec{x}^l \cdot \vec{x}^p)}{(\vec{x}^p \cdot \vec{x}^p)}\right)$$

- To bude splněno, pokud bude absolutní hodnota všech složek perturbačního členu menší než 1.
- Pro biparitní vektory to znamená, že skalární součin $(\vec{x}^l \cdot \vec{x}^p)$ musí být menší než druhá mocnina délky $\vec{x}^p$

Hebbovské učení

Pokud jsou náhodně zvoleným biparitním vektorům přiřazeny (jiné) náhodně zvolené biparitní vektory, je pravděpodobnost, že budou navzájem ortogonální, poměrně vysoká (pokud jich ovšem nebylo zvoleno příliš mnoho)

V takovém případě bude crosstalk malý a Hebbovské učení povede k volbě vhodných vah pro asociativní síť

Hebbovské učení - geometrická interpretace

- Pro jeden uložený vzor $\vec{x}^l$ a příslušnou matici W^l: vstupní vektor $\vec{z}$ bude zobrazen do lineárního podprostoru L_l určeného vektorem $\vec{x}^l$
  - pro autoasociativní síť: $\vec{z}W^l = \vec{z}\vec{x}^{lT}\vec{x}^l = c_l\vec{x}^l$, kde c_l je konstanta
  - obecně neortogonální projekce
- Celkem pro matici $W = W^1 + ... + W^N$ bude vstupní vektor $\vec{z}$ zobrazen do lineárního podprostoru určeného vektory $\vec{x}^1, ..., \vec{x}^N$:
  $\vec{z}W = \vec{z}W^1 + \vec{z}W^2 + ... + \vec{z}W^N = c_1\vec{x}^1 + c_2\vec{x}^2 + .. + c_N\vec{x}^N$

Hebbovské učení - analýza chování

Sféra vlivu jednotlivých atraktorů (pevných bodů systému)

- Pro biparitní vzory se měří Hammingovou vzdáleností ($\sim$ počet různých složek dvou vektorů).
- Index

$$I = \sum_{h=0}^{n/2} h p_h,$$

kde p_h je procento vzorů s Hamm. vzd. h, které ,,zkonvergovaly" k danému atraktoru.

Hebbovské učení - analýza chování

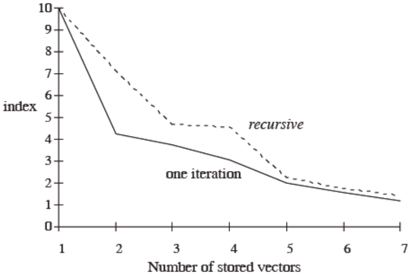
- S rostoucím počtem ukládaných vzorů se „sféry vlivu" jednotlivých atraktorů zmenšují a hrozí vznik **nepravých stabilních stavů**
  - Inverzní vzory k uloženým
  - Velký crosstalk - pokud bude příliš velký, může být dříve uložený vzor i „zapomenut"

Rekurentní síť (se zpětnou vazbou)

- Lepší konvergence oproti asociativní paměti bez zpětné vazby
- Větší „sféry vlivu" jednotlivých atraktorů $\rightarrow$ větší problém s kapacitou sítě

Hebbovské učení - analýza chování

Příklad - srovnání indexu pro síť s a bez zpětné vazby



Odhad kapacity sítě

- m ... počet vzorů, které lze uložit do autoasociativní paměti s maticí W tvaru $n \times n$
- $m \sim 0.18n$ ... maximální kapacita sítě
- Pro korelované vzory může být problém i pro $m < 0.18n$

Hebbovské učení - analýza chování

- V reálných aplikacích jsou vzory téměř vždy korelované a crosstalk pak může v případě Hebbovského učení ovlivnit rozpoznávání $\vec{x}^p W = \vec{y}^p(\vec{x}^p \cdot \vec{x}^p) + \sum_{l \neq p}^{N} \vec{y}^l(\vec{x}^l \cdot \vec{x}^p)$
- Vzájemná korelace ukládaných vzorů vede ke snížení kapacity asociativní paměti (počtu vzorů, které lze uložit a rozpoznat)
- ukládané vzory pak nepokrývají rovnoměrně celý příznakový prostor, ale soustředí se do menší oblasti

Metoda pseudoinverzní matice

$\rightarrow$ je třeba hledat alternativní metody učení schopné minimalizovat perturbaci mezi ukládanými vzory

$\rightarrow$ **použití pseudoinverzní matice namísto korelační**

Metoda pseudoinverzní matice

Definice

- Pseudoinverzní maticí k matici $n \times m$ reálných čísel X je matice X^+ reálných čísel s následujícími vlastnostmi:
  1. $XX^+X = X$
  2. $X^+XX^+ = X^+$
  3. X^+X a XX^+ jsou symetrické

Pseudoinverzní matice vždy existuje a je jednoznačně určena.

Metoda pseudoinverzní matice

Úloha

- Na vstupu je N n–rozměrných vektorů $\vec{x}^1, \ldots, \vec{x}^N$
  ... matice X tvaru $N \times n$
- Na výstupu je N m–rozměrných vektorů $\vec{y}^1, \ldots, \vec{y}^N$
  ... matice Y tvaru $N \times m$
- Hledáme takovou matici vah, že $XW = Y$
  (ve zjednodušeném případě, obecně ještě aplikace přenosové funkce ... $f(XW) = Y$)

Metoda pseudoinverzní matice

Hledáme matici W t.ž. $XW = Y$

- Nápad: $W = X^{-1}Y$
- Problém: k matici X nemusí existovat inverzní matice X^{-1}
  - obecně $m \neq n$
  - vektory $\vec{x}^1, \ldots, \vec{x}^N$ nemusí být navzájem lineárně nezávislé

$\rightarrow$ hledáme matici, která by minimalizovala $\|XW - Y\|^2$

… minimalizace pomocí $W = X^{+}Y$

Metoda pseudoinverzní matice

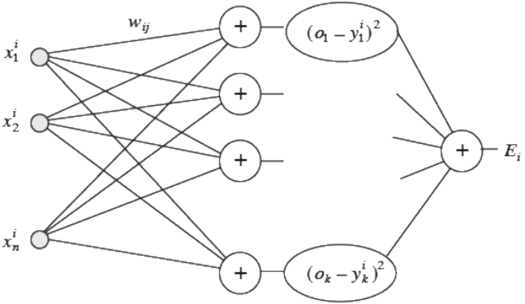
Věta

- Nechť X je matice reálných čísel tvaru $N \times n$, Y je matice reálných čísel tvaru $N \times m$. Matice $W = X^+ Y$ tvaru $n \times m$ minimalizuje $\|XW - Y\|^2$.
- Zároveň X^+ minimalizuje $\|XX^+ - I\|^2$.
  - pokud X^{-1} existuje, bude navíc $\|XX^+ - I\|^2 = 0$, tj. $X^+ = X^{-1}$.
  - pokud X^{-1} neexistuje, bude X^+ její nejlepší možná aproximace

Metoda pseudoinverzní matice

Jak spočítat pseudoinverzní matici?

- např. aproximace pomocí BP-sítí ... BP-síť využita k nalezení vah asociativní sítě



- o ... skutečný výstup
- y ... požadovaný výstup

Metoda pseudoinverzní matice

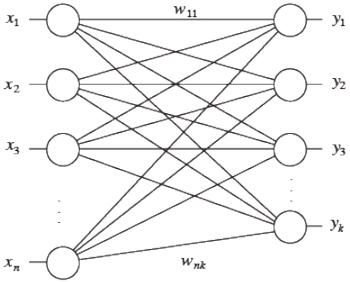
Výpočet pseudoinverzní matice pomocí BP-sítě

- Pro l-tý vstupní vektor se porovná výstup sítě s vektorem $\vec{y}^l$ a spočítá se E_l.
- Celková kvadratická odchylka $E = \sum_{i=l}^{N} E_l$ pak odpovídá $\|XW - Y\|^2$.
- Algoritmus zpětného šíření pak najde matici W, která minimalizuje $\|XW - Y\|^2$.

Bidirektivní asociativní sít (paměť)

BAM = Bidirectional Associative Memory

- synchronní rekurentní asociativní paměť s obousměrnými synapsemi.



Bidirektivní asociativní síť (paměť)

BAM = Bidirectional Associative Memory

- synchronní rekurentní asociativní paměť s obousměrnými synapsemi.
  - 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?

Bidirektivní asociativní síť (paměť)

Popis modelu BAM

- Skoková přenosová funkce (sgn).
- Biparitní kódování hodnot.
- Váhová matice W je tvaru $n \times m$

Výpočet BAM

- Síť zobrazí n–rozměrný vstupní vektor $\vec{x}^0$ na m–rozměrný výstupní vektor $\vec{y}^0$
- Po prvním dopředném průchodu sítí dostaneme: $\vec{y}^0 = sgn(\vec{x}^0 W)$
- Po prvním zpětném průchodu dostaneme: $\vec{x}^1 = sgn(W \vec{y}^{0\,T})$
- Po dalším dopředném průchodu: $\vec{y}^1 = sgn(\vec{x}^1 W)$

Bidirektivní asociativní síť (paměť)

Výpočet BAM

- Po N iteracích máme $N+1$ dvojic vektorů $\vec{x}^0, \ldots, \vec{x}^N$ a $\vec{y}^0, \ldots, \vec{y}^N$ pro které platí:

$$\begin{aligned} \vec{y}^i &= sgn(\vec{x}^i W) \\ \vec{x}^{i+1} &= sgn(W \vec{y}^{iT}) \end{aligned}$$

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

$$\begin{aligned} \vec{y} &= sgn(\vec{x} W) \\ \vec{x} &= sgn(W \vec{y}^T) \end{aligned}$$

Bidirektivní asociativní síť (paměť)

Výpočet váhové matice ... pomocí Hebbovského učení

- pro jeden vzor $(\vec{x}^l, \vec{y}^l)$:

$$W^l = \vec{x}^{lT}\vec{y}^l$$

$$\begin{aligned} \vec{y}^l &= sgn(\vec{x}^l W^l) = sgn(\vec{x}^l \vec{x}^{lT} \vec{y}^l) \\ &= sgn(\|\vec{x}^l\|^2 \vec{y}^l) = \vec{y}^l \end{aligned}$$

  a zároveň

$$\begin{aligned} \vec{x}^{lT} &= sgn(W^l \vec{y}^{lT}) = sgn(\vec{x}^{lT} \vec{y}^l \vec{y}^{lT}) \\ &= sgn(\vec{x}^{lT} \|\vec{y}^l\|^2) = \vec{x}^{lT} \end{aligned}$$

- pro více vzorů $\vec{x}^1, ..., \vec{x}^N$:

$$W = W^1 + ... + W^N$$

Bidirektivní asociativní síť (paměť)

Výpočet váhové matice

- Lze použít Hebbovské učení nebo Metodu pseudoinverzní matice
- Vzory by měly být navzájem ortogonální ($\rightarrow$ menší crosstalk)
- BAM lze použít i jako autoasociativní sítě, protože matice vah jsou symetrické ($X = XW$ a $X^T = WX^T$)

Asynchronní 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 BAM

Stabilní stav BAM

= dvojice vektorů $(\vec{x}, \vec{y})$, pro kterou platí:

$$\begin{aligned}\vec{y} &= sgn(\vec{x}W) \\ \vec{x} &= sgn(W\vec{y}^T)\end{aligned}$$

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. $\rightarrow$ libovolná reálná matice vah W má bidirektivní stabilní biparitní stavy

Asynchronní BAM

Definice: Energetická funkce sítě BAM

$$E(\vec{x},\vec{y}) = -\frac{1}{2}\vec{x}W\vec{y}^T = -1/2\sum_{i=1}^{n}\sum_{j=1}^{n}\vec{x}w_{ij}x_iy_j$$

Náznak důkazu věty

- Lokální minima energetické funkce odpovídají stabilním stavům.
- Každá aktualizace stavu sítě vede ke snížení její celkové energie.
- Stavů je konečně mnoho $\rightarrow$ výpočet skončí v konečném počtu kroků.