

Autoasociativní síť.

Organizační dynamika:

- ▶ úplná topologie, tj. každý neuron je spojen s každým
- ▶ všechny neurony jsou současně vstupní i výstupní
- ▶ označme ξ_1, \dots, ξ_n vnitřní potenciály a y_1, \dots, y_n výstupy (stavy) jednotlivých neuronů
- ▶ označme w_{ji} celočíselnou váhu spoje od neuronu $i \in \{1, \dots, n\}$ k neuronu $j \in \{1, \dots, n\}$.
- ▶ žádný neuron nemá bias a předpokládáme $w_{jj} = 0$ pro každé $j = 1, \dots, n$.

Hopfieldova síť - opakování

Adaptivní dynamika: Dána tréninková množina

$$\mathcal{T} = \{\vec{x}_k \mid \vec{x}_k = (x_{k1}, \dots, x_{kn}) \in \{-1, 1\}^n, k = 1, \dots, p\}$$

Adaptace probíhá podle Hebbova zákona (stejně jako u LAS).
Výsledná konfigurace je matice W vah

$$w_{ji} = \sum_{k=1}^p x_{kj} x_{ki} \quad 1 \leq j \neq i \leq n$$

Všimněte si, že $w_{ji} = w_{ij}$, tedy matice vah je symetrická.

Adaptaci lze vidět jako hlasování vzorů o vazbách neuronů:

$w_{ji} = w_{ij}$ se rovná rozdílu mezi počtem souhlasných stavů $x_{kj} = x_{ki}$ neuronů i a j a počtem rozdílných stavů $x_{kj} \neq x_{ki}$.

Hopfieldova síť - opakování

Aktivní dynamika: Iniciálně jsou neurony nastaveny na vstup sítě $\vec{x} = (x_1, \dots, x_n)$, tedy $y_j^{(0)} = x_j$ pro každé $j = 1, \dots, n$.

V t -tém kroku aktualizujeme neuron j , který splňuje $t = \tau \cdot (n - 1) + j - 1$ takto:

nejprve vypočteme vnitřní potenciál

$$\xi_j^{(t-1)} = \sum_{i=1}^n w_{ji} y_i^{(t-1)}$$

a poté

$$y_j^{(t)} = \begin{cases} 1 & \xi_j^{(t-1)} > 0 \\ y_j^{(t-1)} & \xi_j^{(t-1)} = 0 \\ -1 & \xi_j^{(t-1)} < 0 \end{cases}$$

Hopfieldova síť - opakování

Výpočet končí v kroku t^* pokud se síť nachází (poprvé) ve *stabilním* stavu, tj.

$$y_j^{(t^*+n)} = y_j^{(t^*)} \quad (j = 1, \dots, n)$$

Věta

Za předpokladu symetrie vah, výpočet Hopfieldovy sítě skončí pro každý vstup.

Dokážeme později pomocí energetické funkce.

Z toho plyne, že Hopfieldova síť počítá funkci z $\{-1, 1\}^n$ do $\{-1, 1\}^n$ (která závisí na hodnotách vah neuronů).

Označme $\vec{y}(W, \vec{x}) = (y_1^{(t^*)}, \dots, y_n^{(t^*)})$ hodnotu funkce sítě pro vstup \vec{x} a matici vah W .

Pokud bude W jasné z kontextu, budu psát jen $y(\vec{x})$.

Energetická funkce - opakování

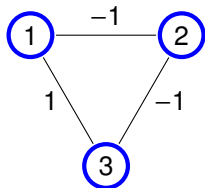
Energetická funkce E přiřazuje každému stavu sítě $\vec{y} \in \{-1, 1\}^n$ potenciální energii danou

$$E(\vec{y}) = -\frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n w_{ji} y_j y_i$$

- ▶ velké (kladné) $w_{ji} y_j y_i$ je stabilní a malé (záporné) $w_{ji} y_j y_i$ nestabilní
- ▶ stavy s nízkou energií jsou relativně stabilní (málo neuronů „chce“ změnit svůj stav), stavy s vysokou energií jsou nestabilní

V ideálním případě Hebbovo učení zakóduje vzory jako (globální) minima funkce E .

Hopfield - příklad



y_1	y_2	y_3	E
1	1	1	1
1	1	-1	1
1	-1	1	-3
1	-1	-1	1
-1	1	1	1
-1	1	-1	-3
-1	-1	1	1
-1	-1	-1	1

- ▶ Hopfieldova síť se třemi neurony
- ▶ naučili jsme ji jeden vzor $(1, -1, 1)$ pomocí Hebbova učení (síť se automaticky „naučila“ i vzor $(-1, 1, -1)$)

Pomocí pojmu energie lze snadno dokázat, že výpočet Hopfieldovy sítě vždy zastaví:

- ▶ v průběhu výpočtu se energie nezvyšuje:
 $E(\vec{y}^{(t-1)}) \geq E(\vec{y}^{(t)})$
- ▶ pokud dojde v kroku t ke změně stavu, pak
 $E(\vec{y}^{(t-1)}) > E(\vec{y}^{(t)})$
- ▶ existuje pouze konečně mnoho stavů sítě: výpočet dosáhne lokálního minima funkce E , ze kterého už se nedostane

Hopfieldova síť - odučování

Při učení podle Hebbova zákona mohou vznikat lokální minima funkce E , tzv. nepravé vzory (*fantomy*), které neodpovídají tréninkovým vzorům.

Fantomy je možné odučovat např. pomocí následujícího pravidla: Mějme fantom $(x_1, \dots, x_n) \in \{-1, 1\}^n$ a váhy w_{ji} , pak nové váhy w'_{ji} spočítáme pomocí

$$w'_{ji} = w_{ji} - x_i x_j$$

(tj. podobně jako při adaptaci podle Hebbova zákona, ale s opačným znaménkem)

Hopfieldova síť - reprodukce

Matice všech vah v síti W je dána $W = \sum_{k=1}^p \vec{x}_k \vec{x}_k^T$

Označme $\vec{y}^{(t)} = \begin{pmatrix} y_1^{(t)} \\ \vdots \\ y_n^{(t)} \end{pmatrix}$

Pak $\vec{y}^{(t)} = \text{sgn}(W\vec{y}^{(t-1)})$

Zde pro jednoduchost předpokládáme, že všechny složky $W\vec{x}$ jsou nenulové.

Daný vstup \vec{x} je **stabilní** pokud $\vec{x} = \text{sgn}(W\vec{x})$

Tj. výpočet sítě končí po n krocích.

Pro daný tréninkový vzor \vec{x}_r máme

$$\text{sgn}(W\vec{x}_r) = \text{sgn}\left(\vec{x}_r + \sum_{k \neq r} \vec{x}_k \frac{(\vec{x}_k^T \vec{x}_r)}{(\vec{x}_r^T \vec{x}_r)}\right)$$

Tedy \vec{x}_r je stabilní pokud $\left| \sum_{k \neq r} \vec{x}_k \frac{(\vec{x}_k^T \vec{x}_r)}{(\vec{x}_r^T \vec{x}_r)} \right| < 1$

Kapacita Hopfieldovy paměti je dána poměrem p/n .

Zde n je počet neuronů a p je počet vzorů.

Předpokládejme, že tréninkové vzory jsou voleny náhodně takto: při volbě \vec{x}_k volím postupně (nezávisle) jednotlivé složky (1 s pravd. $1/2$ a -1 s pravd. $1/2$).

Uvažme konfiguraci W , kterou obdržíme Hebbovským učením na zvolených vzorech.

Označme

$$\beta = \mathbf{P} \left[\vec{x}_k = \vec{y}(W, \vec{x}_k) \text{ pro } k = 1, \dots, p \right]$$

Pak pro $n \rightarrow \infty$ a $p \leq n/(4 \log n)$ dostaneme $\beta \rightarrow 1$.

Tj. maximální počet vzorů, které lze uložit do Hopfieldovy paměti je úměrný $n/(4 \log n)$.

Problém:

- ▶ příliš mnoho vzorů implikuje existenci lokálních minim funkce E , která neodpovídají vzorům (tzv. *fantomy*)
- ▶ lokální minima pro vzory mohou dokonce zanikat

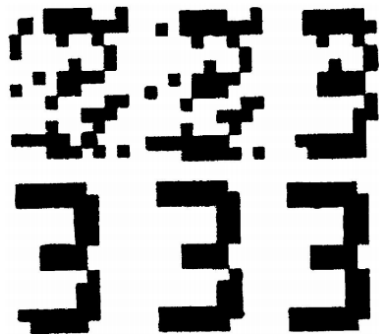
Podrobná analýza ukazuje následující

- ▶ Pro $p \leq 0.138n$ tréninkové vzory odpovídají lokálním minimům funkce E
- ▶ Pro $p > 0.138n$ lokální minima příslušející vzorům zanikají
- ▶ Pro $p < 0.05n$ tréninkové vzory odpovídají globálním minimům E a fantomy mají ostře větší energii

Tj. pro dobré zapamatování 10 vzorů je potřeba 200 neuronů a tedy 40000 spojů ohodnocených celočíselnými váhami
(autorovi knihy z roku 1996 to připadalo nepříliš praktické :-)

Pozn. Nevýhodou Hopfieldovy sítě je deterministický výpočet, který může skončit v mělkém lokálním minimu E bez možnosti uniknout. Tento problém částečně vyřeší Boltzmannův stroj - stochastické rozšíření Hopfieldovy sítě.

Hopfieldova síť - příklad kódování



- ▶ číslice 12×10 bodů
(120 neuronů, -1 je bílá a 1 je černá)
- ▶ naučeno 8 číslic
- ▶ vstup vygenerován ze vzoru 25% šumem
- ▶ obrázek ukazuje postup výpočtu Hopfieldovy sítě

Hopfieldova síť a optimalizační úlohy

Optimalizační úloha je zadána množinou přípustných řešení a účelovou funkcí. Cílem je nalézt přípustné řešení, které minimalizuje účelovou funkci.

Pro mnoho optimalizačních úloh lze nalézt Hopfieldovu síť (obvykle je nutné přidat biasy) takovou, že

- ▶ energetická funkce $E =$ účelová funkce
- ▶ minima $E \approx$ přípustná řešení

Cílem je nalézt globální minimum funkce E .

Problém: není jasné, v jakém stavu začít, abychom dosáhli globálního minima. Síť může skončit v mělkém minimu.

Řešení: V každém stavu umožníme s malou pravděpodobností přechod do stavů s vyšší energií. Tuto pravděpodobnost budeme postupně snižovat. Tohoto je schopen Boltzmannův stroj, viz. dále ...

Cíl: Chceme sestrojít síť, která bude reprezentovat dané pravděpodobnostní rozložení na množině vektorů $\{-1, 1\}^n$.

Velmi hrubá a nepřesná idea: Sestrojíme generalizaci Hopfieldovy sítě, jejíž stavy budou z množiny $\{-1, 1\}^n$ a která bude svoje stavy měnit náhodně (podle přechodových pravděpodobností daných konfigurací sítě).

Když takovou síť necháme běžet dost dlouho, potom budou frekvence návštěv jednotlivých stavů nezávislé na iniciálním stavu. Tyto frekvence budeme považovat za pravděpodobnostní rozložení reprezentované sítí.

V adaptivním režimu bude zadáno nějaké rozložení na stavech a cílem bude nalézt konfiguraci takovou, že rozložení reprezentované sítí bude odpovídat zadanému rozložení.

Organizační dynamika:

- ▶ Cyklická síť se symetrickými spoji (tj. libovolný neorientovaný graf)
- ▶ Množinu všech neuronů značíme N
- ▶ označme ξ_j vnitřní potenciál a y_j výstup (stav) neuronu j
- ▶ stav stroje: $\vec{y} \in \{-1, 1\}^{|N|}$.
- ▶ označme w_{ji} reálnou váhu spoje od neuronu i k neuronu j .
- ▶ žádný neuron nemá bias a předpokládáme $w_{jj} = 0$ pro $j \in N$.

Botzmannův stroj

Aktivní dynamika: Stavby neuronů jsou iniciálně nastaveny na hodnoty z množiny $\{-1, 1\}$, tj. $y_j^{(0)} \in \{-1, 1\}$ pro $j \in N$.

V t -tém kroku aktualizujeme náhodně vybraný neuron $j \in N$ takto: nejprve vypočteme vnitřní potenciál

$$\xi_j^{(t-1)} = \sum_{i \in j_{\leftarrow}}^n w_{ji} y_i^{(t-1)}$$

a poté náhodně zvolíme hodnotu $y_j^{(t)} \in \{-1, 1\}$ tak, že $\mathbf{P}[y_j^{(t)} = 1] = \sigma(\xi_j^{(t-1)})$ kde

$$\sigma(\xi) = \frac{1}{1 + e^{-2\xi/T(t)}}$$

Parametr $T(t)$ se nazývá **teplota** v čase t .

Boltzmannův stroj - teplota a energie

- ▶ Velmi vysoká teplota $T(t)$ znamená, že $\mathbf{P}[y_j^{(t)} = 1] \approx \frac{1}{2}$ a stroj se chová téměř náhodně.
- ▶ Velmi nízká teplota $T(t)$ znamená, že buď $\mathbf{P}[y_j^{(t)} = 1] \approx 1$ nebo $\mathbf{P}[y_j^{(t)} = 1] \approx 0$ v závislosti na tom, jestli $\xi_j^{(t)} > 0$ nebo $\xi_j^{(t)} < 0$. Potom se stroj chová téměř deterministicky (tj. jako Hopfieldova síť).

Podobně jako u Hopfieldovy sítě definujeme energetickou funkci E

$$E(\vec{y}) = -\frac{1}{2} \sum_{j \in N} \sum_{i \in j_{\leftarrow}} w_{ji} y_j y_i$$

Boltzmannův stroj funguje jako Hopfieldova síť rozšířená o náhodný šum na přechodech, tj. energie se může občas zvýšit.

Pravděpodobnost přechodů do vyšší energetické hladiny se exponenciálně zmenšuje s velikostí energetického skoku.

Simulované žihání

Následujícím postupem lze dosáhnout globálního minima funkce E :

- ▶ Na začátku výpočtu nastavíme vysokou teplotu $T(t)$
- ▶ Teplotu postupně snižujeme, např takto:
 - ▶ $T(t) = \eta^t \cdot T(0)$ kde $\eta < 1$ je blízko 1
 - ▶ nebo $T(t) = T(0) / \log(1 + t)$

Lze dokázat, že při vhodném postupu chlazení dosáhneme globálního minima.

Pozn:

- ▶ Tento proces je analogií žihání používané při výrobě tvrdých kovových materiálů s krystalickou strukturou: materiál se nejprve zahřeje, čímž se poruší vazby mezi atomy, v průběhu následného pomalého chlazení se materiál „usadí“ do stavu s minimální vnitřní energií a s pravidelnou vnitřní strukturou.
- ▶ Jedná se také o rozšíření fyzikální motivace Hopfieldovy sítě: orientace magnetů jsou ovlivněny nejen vnitřním a vnějším magnetickým polem, ale také termálními fluktuacemi.

Rovnovážný stav

Fixujeme teplotu T (tj. $T(t) = T$ pro $t = 1, 2, \dots$).

Boltzmannův stroj se po jisté době dostane do tzv. *termální rovnováhy*. Tj. existuje čas t^* takový, že pro libovolný stav stroje $\gamma^* \in \{-1, 1\}^{|N|}$ platí

$$\mathbf{P}[\vec{y}^{(t^*)} = \gamma^*] \approx p_N(\gamma^*)$$

Zde $p_N(\gamma^*) = \frac{1}{Z} e^{-E(\gamma^*)/T}$ kde

$$Z = \sum_{\gamma \in \{-1, 1\}^{|N|}} e^{-E(\gamma)/T}$$

tj. Boltzmannovo rozložení

Teorie Markovových řetězců říká, že $\mathbf{P}[\vec{y}^{(t^*)} = \gamma^*]$ je také dlouhodobá frekvence návštěv stavu γ^* .

Toto platí *bez ohledu na iniciální nastavení neuronů!* Síť tedy reprezentuje rozložení p_N .

Problém: Tak jak jsme si jej definovali má Boltzmannův stroj omezenou schopnost reprezentovat daná rozložení.

Proto množinu neuronů N disjunktně rozdělíme na

- ▶ množinu **viditelných** neuronů V
- ▶ množinu **skrytých** neuronů S .

Pro daný stav viditelných neuronů $\alpha \in \{-1, 1\}^{|V|}$ označme

$$p_V(\alpha) = \sum_{\beta \in \{-1, 1\}^{|S|}} p_N(\alpha, \beta)$$

pravděpodobnost stavu viditelných neuronů α v termálním ekvilibriu bez ohledu na stav skrytých neuronů.

Cílem bude adaptovat síť podle daného rozložení na $\{-1, 1\}^{|V|}$.

Adaptivní dynamika:

Nechť p_d je pravděpodobnostní rozložení na množině stavů viditelných neuronů, tj. na $\{-1, 1\}^{|V|}$.

Cílem je nalézt konfiguraci sítě W takovou, že p_V odpovídá p_d .

Vhodnou mírou rozdílu mezi rozděleními p_V a p_d je relativní entropie zvážená pravděpodobnostmi vzorů (tzv. Kullback Leibler distance)

$$\mathcal{E}(\vec{w}) = \sum_{\alpha \in \{-1, 1\}^{|V|}} p_d(\alpha) \ln \frac{p_d(\alpha)}{p_V(\alpha)}$$

Boltzmannův stroj - učení

$\mathcal{E}(\vec{w})$ budeme minimalizovat pomocí gradientního sestupu, tj. budeme počítat poslounost vektorů vah $\vec{w}^{(0)}, \vec{w}^{(1)}, \dots$

- ▶ váhy v $\vec{w}^{(0)}$ jsou inicializovány náhodně blízko 0
- ▶ v t -tém kroku (zde $t = 1, 2, \dots$) je $\vec{w}^{(t)}$ vypočteno takto:

$$w_{ji}^{(t)} = w_{ji}^{(t-1)} + \Delta w_{ji}^{(t)}$$

kde

$$\Delta w_{ji}^{(t)} = -\varepsilon(t) \cdot \frac{\partial \mathcal{E}}{\partial w_{ji}}(\vec{w}^{(t-1)})$$

je změna váhy w_{ji} v t -tém kroku a $0 < \varepsilon(t) \leq 1$ je rychlost učení v t -tém kroku.

Zbývá spočítat (odhadnout) $\frac{\partial \mathcal{E}}{\partial w_{ji}}(\vec{w})$.

Boltzmannův stroj - učení

Formálním derivováním funkce \mathcal{E} lze ukázat, že

$$\frac{\partial \mathcal{E}}{\partial w_{ji}} = -\frac{1}{T} \left(\langle y_j^{(t^*)} y_i^{(t^*)} \rangle_{fixed} - \langle y_j^{(t^*)} y_i^{(t^*)} \rangle_{free} \right)$$

- ▶ $\langle y_j^{(t^*)} y_i^{(t^*)} \rangle_{fixed}$ je průměrná hodnota $y_j^{(t^*)} y_i^{(t^*)}$ v termální rovnováze za předpokladu, že hodnoty viditelných neuronů jsou fixovány na počátku výpočtu dle rozložení p_d .
- ▶ $\langle y_j^{(t^*)} y_i^{(t^*)} \rangle_{free}$ je průměrná hodnota $y_j^{(t^*)} y_i^{(t^*)}$ v termální rovnováze bez fixace viditelných neuronů.

Celkově

$$\begin{aligned} \Delta w_{ji}^{(t)} &= -\varepsilon(t) \cdot \frac{\partial \mathcal{E}}{\partial w_{ji}} (\vec{w}^{(t-1)}) \\ &= \frac{\varepsilon(t)}{T} \left(\langle y_j^{(t^*)} y_i^{(t^*)} \rangle_{fixed} - \langle y_j^{(t^*)} y_i^{(t^*)} \rangle_{free} \right) \end{aligned}$$

Pro upřesnění:

$$\begin{aligned} \langle y_i^{(t^*)} y_j^{(t^*)} \rangle_{fixed} &= \\ &= \sum_{\alpha \in \{-1,1\}^{|V|}} p_d(\alpha) \sum_{\beta \in \{-1,1\}^{|S|}} \frac{p_N(\alpha, \beta)}{p_V(\alpha)} y_j^{\alpha\beta} y_i^{\alpha\beta} \end{aligned}$$

kde $y_j^{\alpha\beta}$ je výstup neuronu j ve stavu (α, β) .

$$\langle y_i^{(t^*)} y_j^{(t^*)} \rangle_{free} = \sum_{\gamma \in \{-1,1\}^{|M|}} p_N(\gamma) y_j^\gamma y_i^\gamma$$

Boltzmannův stroj - učení

Výpočet Δw_{ji} lze realizovat pomocí simulace.

Pro výpočet $\langle y_j^{(t^*)} y_i^{(t^*)} \rangle_{fixed}$ proved' následující:

- ▶ Polož $\mathcal{Y} := 0$ a proved' následující akce q krát:
 1. fixuj náhodně hodnoty viditelných neuronů dle rozložení p_d (tj. v průběhu následujících kroků je neaktualizuj)
 2. simuluj stroj tak dlouho, dokud nedosáhne termální rovnováhy při nízké teplotě (zde je možné použít tzv. simulované žhání, tedy pomalé snižování teploty).
 3. simuluj stroj po dalších ℓ kroků a v každém z těchto kroků přičti aktuální hodnotu $y_j y_i$ k proměnné \mathcal{Y} .
- ▶ $\mathcal{Y}/q\ell$ bude dobrým odhadem $\langle y_j^{(t^*)} y_i^{(t^*)} \rangle_{fixed}$ za předpokladu, že q a ℓ jsou dostatečně velká čísla.

$\langle y_j^{(t^*)} y_i^{(t^*)} \rangle_{free}$ se odhadne podobně, pouze se nefixují viditelné neurony (tj. v kroku 1. se zvolí libovolný startovní stav a v následném výpočtu se mohou aktualizovat všechny neurony).