Omezený Boltzmannův stroj (OBS)

Organizační dynamika:

- Cyklická síť se symetrickými spoji, neurony jsou rozděleny do dvou skupin:
  - V - viditelné
  - S - skryté

  Množina spojů je $V \times S$ (tj. úplný bipartitní 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} váhu spoje od neuronu i k neuronu j.
- obvykle se uvažuje bias, pro zjednodušení jej vynecháme.

Omezený Botzmannův stroj

Aktivní dynamika: Stavy viditelných neuronů jsou iniciálně nastaveny na hodnoty z množiny $\{-1, 1\}$.

V t-tém kroku aktualizujeme neurony takto:

- t liché: náhodně zvolíme nové hodnoty skrytých neuronů, pro $j \in S$

$$\mathbf{P}\left[y_j^{(t)} = 1\right] = 1 \Bigg/ \left(1 + \exp\left(-\sum_{i \in V} w_{ji} y_i\right)\right)$$

- t sudé: náhodně zvolíme nové hodnoty viditelných neuronů, pro $j \in V$

$$\mathbf{P}\left[y_j^{(t)} = 1\right] = 1 \Bigg/ \left(1 + \exp\left(-\sum_{i \in S} w_{ji} y_i\right)\right)$$

Rovnovážný stav

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

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

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 $\mathbf{P}\left[\vec{y}^{(t^*)} = \gamma^*\right]$ je také dlouhodobá frekvence návštěv stavu γ^*.

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

Omezený Boltzmannův stroj - učení

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 termální rovnováze bez ohledu na stav skrytých neuronů.

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)

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

Omezený Boltzmannův stroj - učení

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

- váhy v $\vec{w}^{(0)}$ jsou inicializovány náhodně blízko 0
- v t-tém kroku (zde $t = 1, 2, \ldots$) 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) $\frac{\partial \mathcal{E}}{\partial w_{ji}}(\vec{w})$.

Omezený Boltzmannův stroj - učení

Lze ukázat, že

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

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

Problém: výpočet $\left\langle y_j^{(t^*)} y_i^{(t^*)} \right\rangle_{free}$ trvá dlouho (musíme opakovaně přivést stroj do termální rovnováhy).

$\left\langle y_j^{(t^*)} y_i^{(t^*)} \right\rangle_{free}$ se proto nahrazuje $\left\langle y_j y_i \right\rangle_{recon}$ což je průměrná hodnota $y_j^{(3)} y_i^{(3)}$ za předpokladu, že iniciální hodnoty viditelných neuronů jsou voleny dle p_d.

Omezený Boltzmannův stroj - učení

Tedy

$$\Delta w_{ji}^{(t)} = -\varepsilon(t) \cdot \left(\left\langle y_j y_i \right\rangle_{fixed} - \left\langle y_j y_i \right\rangle_{recon} \right)$$

- $\left\langle y_j y_i \right\rangle_{fixed}$ se vypočte takto: Polož $\mathcal{Y} := 0$ a opakuj q krát:
  - fixuj náhodně hodnoty viditelných neuronů dle p_d
  - simuluj jeden krok výpočtu a přičti aktuální hodnotu $y_j y_i$ k $\mathcal{Y}$

  Pro vhodné q bude $\mathcal{Y}/q$ dobrým odhadem $\left\langle y_j y_i \right\rangle_{fixed}$
- $\left\langle y_j y_i \right\rangle_{recon}$ se vypočte takto: Polož $\mathcal{Y} := 0$ a opakuj q krát:
  - nastav náhodně hodnoty viditelných neuronů dle p_d
  - simuluj tři kroky výpočtu a přičti aktuální hodnotu $y_j y_i$ k $\mathcal{Y}$ (tj. vypočti hodnoty skrytých neuronů, potom hodnoty viditelných (tzv. rekonstrukci vstupu) a potom hodnoty skrytých)

  Pro vhodné q bude $\mathcal{Y}/q$ dobrým odhadem $\left\langle y_j y_i \right\rangle_{recon}$

Hluboké sítě

Problém: Omezením BS jsme (potenciálně) omezili vyjadřovací sílu.
Řešení: OBS se skládají nad sebe do tzv. hlubokých sítí

1. OBS se natrénuje na vstupních datech
2. přidá se další vrstva, skryté neurony starého OBS budou nyní uvažovány jako viditelné neurony pro nejvyšší OBS. Natrénuje se nejvyšší OBS
3. iterováním bodů 1. a 2. se přidává více a více vrstev

Na konci dostaneme jednu mnohovrstvou síť, která reprezentuje rozložení na datech. Z tohoto rozložení se dá samplovat takto:

- přiveď nejvyšší OBS do termální rovnováhy (to dá hodnoty neuronů v nejvyšších dvou vrstvách)
- propaguj hodnoty do nižších vrstev (tj. proveď jeden krok aktualizace stavů mezilehlých OBS)
- stav neuronů v nejspodnější vrstvě potom bude představovat vzorek dat; pravděpodobnost s jakou se tam objeví konkrétní stav je pravděpodností onoho stavu v rozložení reprezentovaném sítí

Hluboké sítě - klasifikace

Předpokládejme, že každý vstup patří do jedné ze dvou tříd. Chceme vstupy klasifikovat pomocí vícevrstvé sítě.

Vícevrstvou síť lze trénovat pomocí zpětné propagace. Ta je silně závislá na vhodné inicializaci, často hrozí dosažení mělkého lokálního minima apod. Dobře fungující síť si vyvine systém extraktorů vlastností (tj. každý neuron reaguje na nějakou vlastnost vstupu). Jak toho dosáhnout?

- Natrénuj hlubokou síť na datech (v této fázi ignoruj příslušnost do tříd)
- Uvažuj výslednou síť jako obyčejnou vícevrstvou síť, tj. zaměň dynamiku Boltzmannova stroje za sigmoidální aktivační funkce a obvyklé vyhodnocení zdola nahoru
- přidej výstupní vrstvu s jedním neuronem
- dolaď síť pomocí zpětné propagace (malá rychlost učení pro skryté vrstvy, velká pro výstupní vrstvu): Pro vstupy z jedné třídy uvažujeme očekávaný výstup 1 pro ostatní -1.

Spojitá Hopfieldova síť

Organizační dynamika:

- úplná topologie, tj. každý neuron je spojen s každým
- všechny neurony jsou současně vstupní i výstupní
- označme $\xi_1, \ldots, \xi_n$ vnitřní potenciály a $y_1, \ldots, y_n$ výstupy (stavy) jednotlivých neuronů
- označme w_{ji} *reálnou* váhu spoje od neuronu $i \in \{1, \ldots, n\}$ k neuronu $j \in \{1, \ldots, n\}$.
- každý neuron j má bias w_{j0} odpovídající formálnímu jednotkovému vstupu $y_0 = 1$
- přepokládáme $w_{jj} = 0$ pro každé $j = 1, \ldots, n$.

Spojitá Hopfieldova síť

Aktivní dynamika: Iniciálně jsou neurony nastaveny na vstup sítě $\vec{x}$, tedy $\vec{y}(0) = \vec{x}$.

Vývoj stavu sítě $\vec{y}(t)$ je spojitou funkcí času $t > 0$, která je dána následující soustavou diferenciálních rovnic:

$$\tau_j \frac{dy_j}{dt} = -y_j + \sigma\left(\xi_j\right) = -y_j + \sigma\left(\sum_{i=0}^{n} w_{ji} y_i\right)$$

Zde $\tau_j > 0$ jsou vhodné konstanty, σ je spojitá aktivační funkce, např.

$$\sigma(\xi) = \frac{1}{1 + e^{-\lambda\xi}}$$

Platí, že stavy neuronů jsou z intervalu $(0, 1)$.

Spojitá Hopfieldova síť

Tvrzení: Síť konverguje k stavu v němž platí $\frac{d\vec{y}}{dt} = 0$.

Definujeme energii pomocí Ljapunovy funkce:

$$E = -\frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n} w_{ji} y_j y_i - \sum_{j=1}^{n} w_{j0} y_j + \sum_{j=1}^{n}\int_0^{y_j} \sigma^{-1}(y) dy$$

Dá se ukázat, že

$$\frac{dE}{dt} = -\sum_{j=1}^{n} \tau_j \sigma'(\xi_j)\left(\frac{d\xi_j}{dt}\right)^2 \leq 0$$

Energie $E(t)$ během aktivního režimu klesá až do chvíle, kdy se síť dostane do lokálního minima fce E, kde platí $\frac{dE}{dt} = 0$, což odpovídá $\frac{dy_j}{dt} = \frac{d\xi_j}{dt} = 0$.

Díky poslednímu členu v definici E jsou stavy splňující $\frac{dE}{dt} = 0$ (téměř) binární, tj. $y_j \approx 1$ nebo $y_j \approx 0$.

Analýza hlavních komponent (PCA)

Mějme množinu vstupních dat $\{\vec{x}_1, \ldots, \vec{x}_p\}$ kde každé $\vec{x}_i \in \mathbb{R}^n$.

j-tou složku vektoru $\vec{x}_i$ budu značit x_{ij}.

Označme $\bar{x}$ průměrnou hodnotu vstupů:

$$\bar{x} = \frac{1}{p} \sum_{i=1}^{p} \vec{x}_i$$

Jednotlivé složky vektoru $\bar{x}$ budu značit $\bar{x}_j$.

Předpokládejme, že $\bar{x}_j = 0$ pro každé $j = 1, \ldots, n$.

Naším cílem je lineárně transformovat množinu $\{\vec{x}_1, \ldots, \vec{x}_p\}$ na množinu $\{\vec{z}_1, \ldots, \vec{z}_p\}$ kde $\vec{z}_i \in \mathbb{R}^\ell$ pro $\ell < n$.

Tato transformace by měla zachovat „informační obsah".

Analýza hlavních komponent (PCA)

Idea: Pokusme se nalézt směry maximálního rozptylu vstupů (tyto směry by se měly stát osami báze do níž budeme transformovat vstupy, směry s malým rozptylem budou ignorovány)

Uvažme projekce $\vec{x}_i^T \vec{q}$ vstupů $\vec{x}_i$ na fixní vektor $\vec{q}$ délky 1.

Protože $\bar{x} = 0$, průměr hodnot $\vec{x}_i^T \vec{q}$ je roven 0.

Označme $V_{\vec{q}}$ **rozptyl vstupů** $\vec{x}_i$ **ve směru** $\vec{q}$:

$$V_{\vec{q}} = \frac{1}{p} \sum_{i=1}^{p} \left(\vec{x}_i^T \vec{q} \right)^2$$

Nalezneme vektory $\vec{c}_1, \ldots, \vec{c}_n \in \mathbb{R}^n$ takové, že

- $\left\| \vec{c}_i \right\| = 1$ pro každé $i = 1, \ldots, n$
- $\vec{c}_i$ maximalizuje $V_{\vec{q}}$ mezi všemi vektory $\vec{q}$, které jsou kolmé na $\vec{c}_1, \ldots, \vec{c}_{i-1}$ a mají normu 1.

Zejména $\vec{c}_1$ maximalizuje rozptyl $V_{\vec{q}}$ mezi všemi vektory $\vec{q}$ s normou 1.

Analýza hlavních komponent (PCA)

Definujeme kovarianční matici C kde $C_{ij} = \frac{1}{p} \sum_{k=1}^{p} x_{ki} x_{kj}$.

Vektor $\vec{q}$ je vlastním vektorem matice C pokud existuje $\lambda > 0$ taková, že $C\vec{q} = \lambda\vec{q}$
($\vec{q}$ je potom vlastní vektor příslušný vlastní hodnotě λ).

Matice C je reálná a symetrická, má tedy n reálných vlastních hodnot $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ a ortonormální bázi vlastních vektorů: $\vec{c}_1, \ldots, \vec{c}_n$, tj.

- $\vec{c}_i \cdot \vec{c}_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$
- vektor $\vec{c}_i$ je vlastním vektorem matice C, který přísluší vlastní hodnotě λ_i

Věta

Každé $\vec{c}_i$ maximalizuje $V_{\vec{q}}$ mezi všemi vektory $\vec{q}$, které jsou kolmé na $\vec{c}_1, \ldots, \vec{c}_{i-1}$ a mají normu 1.

Reprezentace dat

Označme $\mathcal{C} = \{\vec{c}_1, \dots, \vec{c}_n\}$ ortonormální bázi prostoru $\mathbb{R}^n$ složenou z vlastních vektorů kovarianční matice C.

Označme Q matici $n \times n$ jejíž sloupce jsou tvořeny vektory $\vec{c}_i$, tj.

$$Q = (\vec{c}_1 \cdots \vec{c}_n) \qquad \text{kde vektory } \vec{c}_i \text{ jsou sloupcové}$$

Množina vektorů $\{\vec{a}_1, \dots, \vec{a}_p\}$, kde $\vec{x}_i = Q\vec{a}_i$, je vyjádřením množiny vstupů v bázi $\mathcal{C}$. Potom j-tá složka vektoru $\vec{a}_i$ se nazývá j-tá hlavní komponenta vstupu $\vec{x}_i$.

Navíc platí:

- $Q^T Q = QQ^T = I$ a tedy $\vec{a}_i = Q^T \vec{x}_i$

  Q^T je maticí přechodu ze standardní báze prostoru $\mathbb{R}^n$ do báze $\mathcal{C}$; všimněte si, že Q je maticí otočení
- $a_{ij} = \vec{c}_j^{\,T} \cdot \vec{x}_i$, tj. projekce vektoru $\vec{x}_i$ na osu $\vec{c}_j$
- $C = Q \Lambda Q^T$ kde Λ je diagonální matice taková, že $\Lambda_{ii} = \lambda_i$

Redukce dimenze dat

Kovarianční maticí množiny transformovaných vstupů $\{\vec{a}_1, \ldots, \vec{a}_p\}$ je diagonální matice Λ kde $\Lambda_{ii} = \lambda_i$. Zejména jsou jednotlivé složky vektorů $\vec{a}_i$ vzájemně nekorelované a rozptyl i-té složky je λ_i.

Vektor $\vec{x}_i \in \mathbb{R}^n$ zakódujeme pomocí vektoru $\vec{z}_i = (a_{i1}, \ldots, a_{i\ell}) \in \mathbb{R}^\ell$.

Vektor $\vec{z}_i = (a_{i1}, \ldots, a_{i\ell}) \in \mathbb{R}^\ell$ dekódujeme do vektoru

$$\hat{x}_i = \sum_{j=1}^{\ell} a_{ij}\vec{c}_j + \sum_{j=\ell+1}^{n} 0 \cdot \vec{c}_j = \sum_{j=1}^{\ell} a_{ij}\vec{c}_j$$

(0 je průměrná hodnota složek vektorů $\vec{a}_i = Q^T\vec{x}_i$).

(pokud jsou hodnoty $\lambda_{\ell+1}, \ldots, \lambda_n$ hodně malé, pak 0 dává dobrou aproximaci skutečných souřadnic v bázi C)

Definujeme chybu rekonstrukce vstupů

$$E = \frac{1}{2} \sum_{i=1}^{p} \left\| \vec{x}_i - \hat{x}_i \right\|^2$$

Dá se ukázat, že báze $C = \{\vec{c}_1, \ldots, \vec{c}_n\}$ minimalizuje chybu E mezi všemi ortonormálními bázemi prostoru $\mathbb{R}^n$.

Navíc platí

$$E = \sum_{i=\ell+1}^{n} \lambda_i$$

Redukce dimenze dat - kódování

Kódování z $\mathbb{R}^n$ do $\mathbb{R}^\ell$:

Vstup: Množina dat $\{\vec{x}_1, \dots, \vec{x}_p\}$ kde $\vec{x}_i \in \mathbb{R}^n$
Algoritmus:

1. (Normalizace vstupů odečtením průměru)
2. Výpočet kovarianční matice C
3. Výpočet (aproximace) ortonormální báze $\mathcal{C} = \{\vec{c}_1, \dots, \vec{c}_n\}$ vlastních vektorů a transformační matice $Q = (\vec{c}_1 \cdots \vec{c}_n)$
4. Výpočet vektorů hlavních komponent každého vstupu $\vec{a}_i = Q^T \vec{x}_i$ kde $i = 1, \dots, p$

Výstup: Množina $\{(\vec{a}_{i1}, \dots, \vec{a}_{i\ell}) \mid i = 1, \dots, p\}$ a vektory $\vec{c}_1, \dots, \vec{c}_\ell$

Redukce dimenze dat - dekódování

Dekódování z $\mathbb{R}^\ell$ do $\mathbb{R}^n$:

Vstup: Množina $\{\vec{z}_1, \ldots, \vec{z}_p\}$ kde $\vec{z}_i \in \mathbb{R}^\ell$ a vektory $\vec{c}_1, \ldots, \vec{c}_\ell$
Algoritmus:

1. Výpočet

$$\hat{x}_i = \sum_{j=1}^{\ell} z_{ij}\vec{c}_j \qquad \text{pro } i = 1, \ldots, p$$

Výstup: $\{\hat{x}_1, \ldots, \hat{x}_p\}$

Redukce dimenze dat a neuronové sítě

Uvažme vícevrstvou síť $n - \ell - n$ s lineárními aktivačními funkcemi.

Předpokládejme, že je trénována na množině vzorů

$$\mathcal{T} = \{(\vec{x}_i, \vec{x}_i) \mid \vec{x}_i \in \mathbb{R}^n, i = 1, \ldots, p\}$$

pomocí algoritmu, který minimalizuje chybu

$$E = \frac{1}{2} \sum_{i=1}^{p} \sum_{j=1}^{n} \left(y_j(\vec{x}_i) - x_{ij} \right)^2$$

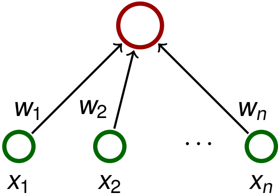
Existuje globální minimum chybové funkce. Dá se ukázat, že po natrénování síť realizuje projekci na podprostor generovaný vektory $\vec{c}_1, \ldots, \vec{c}_\ell$ (ne nutně ortogonální).

To stejné platí i pro sigmoidální aktivační funkce ve skrytých neuronech.

Redukce dimenze dat a Hebbovské učení

Uvažme jednovrstvou síť s jedním lineárním výstupním neuronem:

$$y = \sum_{i=1}^{n} w_i x_i$$



Mějme vstupy $\mathcal{T} = \{\vec{x}_1, \ldots, \vec{x}_p\}$.

Hebbovské učení: $w_i^{(t+1)} = w_i^{(t)} + \eta \cdot y(\vec{x}_t) \cdot x_{ti}$

Normalizované Hebbovské učení:

$$w_i^{(t+1)} = \left(w_i^{(t)} + \eta \cdot y(\vec{x}_k) \cdot x_{ki}\right) \Big/ \left(\sum_{i=1}^{n} \left(w_i^{(t)} + \eta \cdot y(\vec{x}_k) \cdot x_{ki}\right)^2\right)^{\frac{1}{2}}$$

kde k je voleno uniformně náhodně z $\{1, \ldots, p\}$.

Redukce dimenze dat a Hebbovské učení

Pro malé η platí

$$\left(\sum_{i=1}^{n}\left(w_i^{(t)} + \eta \cdot y(\vec{x}_k) \cdot x_{ki}\right)^2\right)^{\frac{1}{2}} = 1 + \eta \cdot y^2(\vec{x}_k) + \mathcal{O}(\eta^2)$$

což pro malé η dá

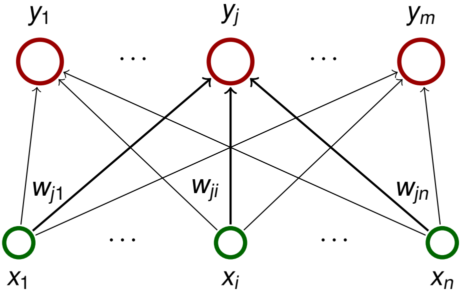
$$w_i^{(t+1)} = w_i^{(t)} + \eta \cdot y(\vec{x}_k) \cdot \left(x_{ki} - y(\vec{x}_k) \cdot w_i^{(t)}\right) + \mathcal{O}(\eta^2)$$

tedy ignorováním $\mathcal{O}(\eta^2)$ dostaneme

Ojovo pravidlo: $w_i^{(t+1)} = w_i^{(t)} + \eta \cdot y(\vec{x}_k) \cdot \left(x_{ki} - y(\vec{x}_k) \cdot w_i^{(t)}\right)$ kde k je náhodně vybrané z množiny $\{1, \ldots, p\}$.

Tvrzení: Pokud $\lambda_1 > \lambda_2$ a $\lambda_n > 0$ pak $\vec{w}^{(t)}$ konverguje (s pravd. 1) k vlastnímu vektoru $\vec{c}_1$ kovariační matice C (tj. k vlastnímu vektoru s maximálním vlastním číslem).

Redukce dimenze dat a Sangerovo pravidlo



Sangerovo pravidlo:

$$w_{ji}^{(t+1)} = w_{ji}^{(t)} + \eta(t) \cdot y_j(\vec{x}_k) \cdot \left(x_{ki} - \sum_{\ell=1}^{j} y_\ell(\vec{x}_k) \cdot w_{\ell i}^{(t)} \right)$$

kde k je náhodně vybrané z množiny $\{1, \ldots, p\}$.

Tvrzení: Pro $\lambda_1 > \cdots > \lambda_n > 0$ a $\eta(t) = \frac{1}{t}$ konverguje $\vec{w}_j^{(t)}$ (s pravd. 1) k vlastnímu vektoru $\vec{c}_j$ kovariační matice C.

Nelineární redukce dimenze dat

- Vícevrstvé sítě
  - síť $n - \ell - k - \ell - n$, kde $n > \ell > k$, se sigmoidálními aktivačními funkcemi provádí nelineární redukci z $\mathbb{R}^n$ do $\mathbb{R}^k$
  - dolní polovina sítě zobrazí vstupy z $\mathbb{R}^n$ do $\mathbb{R}^k$
  - horní polovina dekóduje z $\mathbb{R}^k$ do $\mathbb{R}^n$
  - učí se jako autoasociativní síť
  - lze použít i více vrstev - klasická aplikace hlubokých sítí a omezených Boltzmannových strojů
- Kohonenova mapa
  - k-rozměrná mapa na datech dimenze n provádí nelineární redukci dimenze dat z $\mathbb{R}^n$ do $\mathbb{R}^k$
  - Př. Kohonenova mapa jako mřížka $k \times k$. Nechť pro vstup $\vec{x}$ je aktivní neuron se souřadnicemi (i, j). Vektor $\vec{x}$ je potom transformován do bodu (i, j).