Boltzmannův stroj

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 přepokládáme $w_{jj} = 0$ pro $j \in N$.

Botzmannův stroj

Aktivní dynamika: Stavy 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 $\mathbf{P}\left[y_j^{(t)} = 1\right] = \sigma(\xi_j^{(t-1)})$ kde

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

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

Boltzmannův stroj

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

Boltzmannův stroj - reprezentace rozložení

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

Velmi hrubá a nepřesná idea: Boltzmannův stroj mění náhodně svůj stav z množiny $\{-1, 1\}^{|N|}$.

Když necháme B. stroj běžet dost dlouho s fixní teplotou, potom budou frekvence návštěv jednotlivých stavů nezávislé na iniciálním stavu.

Tyto frekvence budeme považovat za pravděpodobnostní rozložení na $\{-1, 1\}^{|N|}$ reprezentované B. strojem.

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

Rovnovážný stav

Fixujme teplotu T (tj. $T(t) = T$ pro $t = 1, 2, \ldots$).

Boltzmannův stroj se po jisté době dostane do tzv. *termální rovnováhy*. Tj. existuje čas t' takový, že pro libovolný stav stroje $\gamma^* \in \{-1, 1\}^{|N|}$ a libovolné $t^* \geq t'$ platí, že

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

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

$$Z = \sum_{\gamma \in \{-1,1\}^{|N|}} e^{-E(\gamma)/T} \qquad E(\gamma) = -\frac{1}{2} \sum_{i,j} w_{ij} y_i^\gamma y_j^\gamma$$

tj. Boltzmannovo rozložení

Pozn.: 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.

Boltzmannův stroj - učení

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 $\{-1, 1\}^{|V|}$.

Boltzmannův stroj - učení

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 divergence)

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

Boltzmannův stroj - učení

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

- váhy v $W^{(0)}$ jsou inicializovány náhodně blízko 0
- v ℓ-tém kroku (zde $\ell = 1, 2, \ldots$) je $W^{(\ell)}$ vypočteno takto:

$$W_{ji}^{(\ell)} = W_{ji}^{(\ell-1)} + \Delta W_{ji}^{(\ell)}$$

kde

$$\Delta W_{ji}^{(\ell)} = -\varepsilon(\ell) \cdot \frac{\partial \mathcal{E}}{\partial w_{ji}}(W^{(\ell-1)})$$

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

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

Boltzmannův stroj - učení

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

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

- $\left\langle y_j^{(t^*)} y_i^{(t^*)}\right\rangle_{fixed}$ 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.
- $\left\langle y_j^{(t^*)} y_i^{(t^*)}\right\rangle_{free}$ je průměrná hodnota $y_j^{(t^*)} y_i^{(t^*)}$ v termální rovnováze bez fixace viditelných neuronů.

Celkově

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

Boltzmannův stroj - učení

Pro výpočet $\left\langle y_j^{(t^*)} y_i^{(t^*)} \right\rangle_{fixed}$ proveď následující:

- Polož $\mathcal{Y} := 0$ a proveď následující akce q krát:
  1. fixuj náhodně hodnoty viditelných neuronů dle rozložení p_d (tj. v průběhu následujících kroků je neaktualizuj)
  2. simuluj stroj po t^* kroků
  3. přičti aktuální hodnotu $y_j^{(t^*)} y_i^{(t^*)}$ k proměnné $\mathcal{Y}$.
- $\mathcal{Y}/q$ bude dobrým odhadem $\left\langle y_j^{(t^*)} y_i^{(t^*)} \right\rangle_{fixed}$ za předpokladu, že q je dostatečně velké číslo.

$\left\langle y_j^{(t^*)} y_i^{(t^*)} \right\rangle_{free}$ 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).

Boltzmannův stroj - učení

Pro upřesnění analytická verze:

$$\left\langle y_i^{(t^*)} y_j^{(t^*)} \right\rangle_{fixed} =$$

$$= \sum_{\alpha \in \{-1,1\}^{|V|}} p_d(\alpha) \sum_{\beta \in \{-1,1\}^{|S|}} \frac{p_N(\alpha,\beta)}{p_V(\alpha)} y_j^{\alpha\beta} y_i^{\alpha\beta}$$

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

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

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: $\vec{y} \in \{0,1\}^{|N|}$.
- označme w_{ji} váhu spoje mezi neuronem i a neuronem j.
- Uvažme bias: w_{j0} je váha mezi neuronem j a "fiktivním" neuronem 0 jehož hodnota y_0 je stále 1.

Omezený Botzmannův stroj

Aktivní dynamika: Hodnoty viditelných neuronů jsou iniciálně nastaveny na hodnoty z množiny $\{0, 1\}$.

V t-tém kroku aktualizujeme neurony takto:

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

$$\mathbf{P}\left[y_j^{(t)} = 1\right] = 1 \Big/ \left(1 + \exp\left(-w_{j0} - \sum_{i \in V} w_{ji} y_i^{(t-1)}\right)\right)$$

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

$$\mathbf{P}\left[y_j^{(t)} = 1\right] = 1 \Big/ \left(1 + \exp\left(-w_{j0} - \sum_{i \in S} w_{ji} y_i^{(t-1)}\right)\right)$$

Rovnovážný stav

Definujeme energetickou funkci E

$$E(\vec{y}) = -\sum_{i \in V, j \in S} w_{ji} y_j y_i - \sum_{i \in V} w_{i0} y_i - \sum_{j \in S} w_{j0} y_j$$

Omezený Boltzmannův stroj se po jisté době dostane do *termální rovnováhy*. Tj. existuje čas t^* takový, že pro libovolný stav stroje $\gamma^* \in \{0,1\}^{|N|}$ platí

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

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

$$Z = \sum_{\gamma \in \{0,1\}^{|N|}} e^{-E(\gamma)}$$

tj. Boltzmannovo rozložení

Síť tedy reprezentuje rozložení p_N.

Omezený Boltzmannův stroj - učení

Pro daný stav viditelných neuronů $\alpha \in \{0,1\}^{|V|}$ označme

$$p_V(\alpha) = \sum_{\beta \in \{0,1\}^{|S|}} p_N(\alpha, \beta)$$

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 $\{0,1\}^{|V|}$.

Rozložení p_d může být dáno tréninkovou posloupností

$$\mathcal{T} = \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_p$$

tak, že

$$p_d(\alpha) = \#(\alpha, \mathcal{T})/p$$

kde $\#(\alpha, \mathcal{T})$ je počet výskytů α v posloupnosti $\mathcal{T}$

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

Omezený Boltzmannův stroj - učení

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

$$\mathcal{E}(\vec{w}) = \sum_{\alpha \in \{0,1\}^{|V|}} p_d(\alpha) \ln \frac{p_d(\alpha)}{p_V(\alpha)}$$

(Odpovídá maximální věrohodnosti vůči posloupnosti $\mathcal{T}$ v případě, že p_d je definováno pomocí $\mathcal{T}$)

Omezený Boltzmannův stroj - učení

$\mathcal{E}(\vec{w})$ budeme minimalizovat pomocí gradientního sestupu, tj. budeme počítat posloupnost 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

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

- $\left\langle y_j y_i \right\rangle_{fixed}$ 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 neuronů)

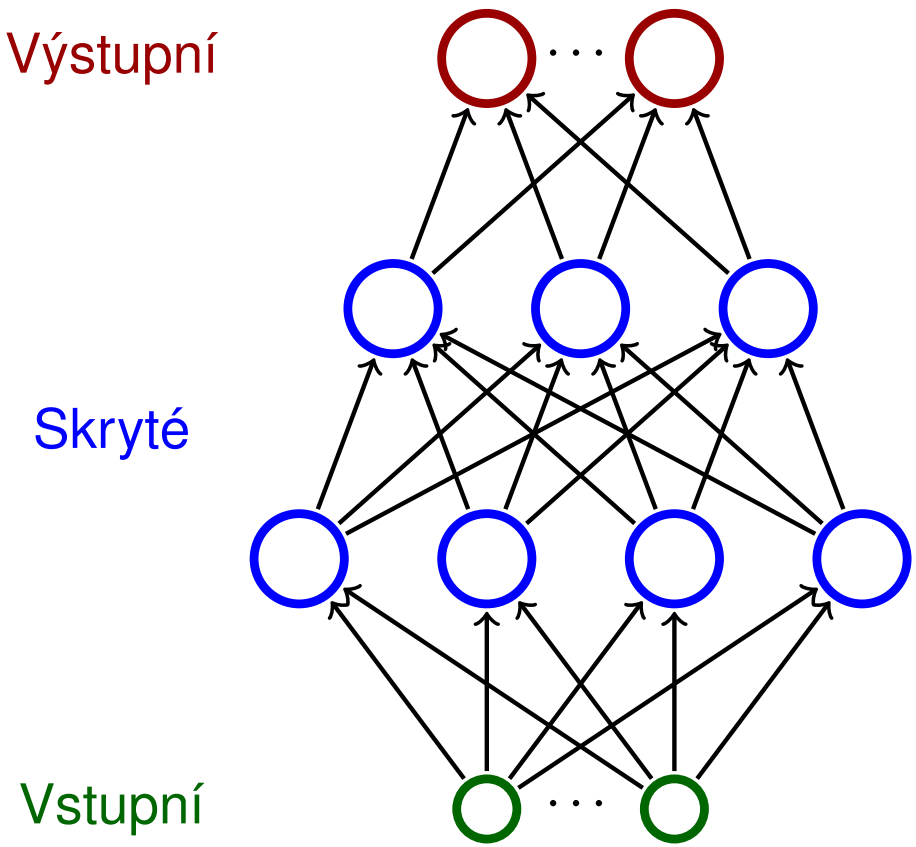
  Pro vhodné q bude $\mathcal{Y}/q$ dobrým odhadem $\left\langle y_j y_i \right\rangle_{recon}$

$\left\langle y_j y_i \right\rangle_{fixed}$ je možné počítat analyticky pro vhodně zadané p_d.

Hluboké sítě

Standardní vícevrstvá síť

Organizační dynamika:



- Neurony jsou rozděleny do **vrstev** (vstupní a výstupní vrstva, obecně několik skrytých vrstev)
- Vrstvy číslujeme od 0; vstupní vrstva je nultá
  - Např. třívrstvá síť se skládá z jedné vstupní, dvou skrytých a jedné výstupní vrstvy.
- Neurony v ℓ-té vrstvě jsou spojeny se všemi neurony ve vrstvě $\ell + 1$.

Hluboké sítě

Značení:

- Označme
  - X množinu vstupních neuronů
  - Y množinu výstupních neuronů
  - Z množinu všech neuronů (tedy $X, Y \subseteq Z$)
- jednotlivé neurony budeme značit indexy i, j apod.
- ξ_j je vnitřní potenciál neuronu j po skončení výpočtu
- y_j je stav (výstup) neuronu j po skončení výpočtu
- w_{ji} je váha spoje **od** neuronu i **do** neuronu j
- $j_{\leftarrow}$ je množina všech neuronů, **z nichž** vede spoj do j (zejména $0 \in j_{\leftarrow}$)
- $j^{\rightarrow}$ je množina všech neuronů, **do nichž** vede spoj z j

Hluboké sítě

Aktivní dynamika:

- vnitřní potenciál neuronu j:

$$\xi_j = \sum_{i \in j_{\leftarrow}} w_{ji} y_i$$

- aktivační funkce

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

pro všechny neurony stejná!

- Stav nevstupního neuronu $j \in Z \setminus X$ po skončení výpočtu je

$$y_j = \sigma(\xi_j)$$

(y_j závisí na konfiguraci $\vec{w}$ a vstupu $\vec{x}$, proto budu občas psát $y_j(\vec{w}, \vec{x})$)

- Síť počítá funkci z $\mathbb{R}^{|X|}$ do $\mathbb{R}^{|Y|}$. Výpočet probíhá po vrstvách. Na začátku jsou hodnoty vstupních neuronů nastaveny na vstup sítě. V kroku ℓ jsou vyhodnoceny neurony z ℓ-té vrstvy.

Hluboké sítě – adaptivní dynamika

Tréninková posloupnost $\mathcal{T}$ vzorů tvaru

$$\left(\vec{x}_1, \vec{d}_1\right), \left(\vec{x}_2, \vec{d}_2\right), \ldots, \left(\vec{x}_p, \vec{d}_p\right)$$

kde každé $\vec{x}_k \in \{0,1\}^{|X|}$ je vstupní vektor a každé $\vec{d}_k \in \{0,1\}^{|Y|}$ je očekávaný výstup sítě. Pro každé $j \in Y$ označme d_{kj} očekávaný výstup neuronu j pro vstup $\vec{x}_k$
(vektor $\vec{d}_k$ lze tedy psát jako $\left(d_{kj}\right)_{j \in Y}$).

Chybová funkce

$$E(\vec{w}) = \sum_{k=1}^{p} E_k(\vec{w})$$

kde

$$E_k(\vec{w}) = \frac{1}{2} \sum_{j \in Y} \left(y_j(\vec{w}, \vec{x}_k) - d_{kj}\right)^2$$

Používají se i jiné funkce.

Proč hluboké sítě

... když jedna vrstva stačí k aproximaci libovolné rozumné funkce?

- Jedna vrstva je často značně neefektivní, tj. může vyžadovat obrovský počet skrytých neuronů pro reprezentaci dané funkce

  Výsledky z teorie Booleovských obvodů ukazují, že nutný počet neuronů může být exponenciální vzhledem k dimenzi vstupu

... ok, proč neučit hluboké sítě pomocí obyčejné zpětné propagace?

- Rychlost učení rapidně klesá s počtem vrstev
- Hluboké sítě mají tendenci se snadno přetrénovat

Vícevrstvá síť - mizející gradient

Pro každé w_{ji} máme

$$\frac{\partial E}{\partial w_{ji}} = \sum_{k=1}^{p} \frac{\partial E_k}{\partial w_{ji}}$$

kde pro každé $k = 1, \ldots, p$ platí

$$\frac{\partial E_k}{\partial w_{ji}} = \frac{\partial E_k}{\partial y_j} \cdot \sigma_j'(\xi_j) \cdot y_i$$

a pro každé $j \in Z \setminus X$ dostaneme

$$\frac{\partial E_k}{\partial y_j} = y_j - d_{kj} \qquad \text{pro } j \in Y$$

$$\frac{\partial E_k}{\partial y_j} = \sum_{r \in j^{\rightarrow}} \frac{\partial E_k}{\partial y_r} \cdot \sigma_r'(\xi_r) \cdot w_{rj} \qquad \text{pro } j \in Z \setminus (Y \cup X)$$

Pro standardní logistickou sigmoidu a váhy inicializované blízko 0 je $\sigma_r'(\xi_r) \cdot w_{rj}$ menší než 1 (pro velké váhy zase větší než 1).

Hluboké sítě – adaptivní dynamika

Předpokládejme k vrstvou síť. Označme

- W_i matici vah mezi vrstvami $i-1$ a i
- F_i funkci počítanou částí sítě s vrstvami $0, 1, \ldots, i$

  tedy F_1 je funkce počítaná jednovrstvou sítí skládající se ze vstupní a první vrstvy sítě, F_k je funkce celé sítě

Všimněte si, že pro každé i lze vrstvy $i-1$ a i společně s maticí W_i chápat jako omezený Boltzmannův stroj B_i (předpokládáme $T = 1$)

Učení ve dvou fázích:

- předtrénování bez učitele: Postupně pro každé $i = 1, \ldots, k$ trénuj OBS B_i na náhodně volených vstupech z posloupnosti

  $$F_{i-1}(\vec{x}_1), \ldots, F_{i-1}(\vec{x}_p)$$

  pomocí algoritmu pro učení OBS (zde $F_0(\vec{x}_i) = \vec{x}_i$)

  (tedy B_i se trénuje na tréninkových vzorech transformovaných vrstvami $0, \ldots, i-1$)
- doladění sítě s učitelem např. pomocí zpětné propagace

Hluboké sítě

Po první fázi dostaneme k vrstvou síť, která reprezentuje rozložení na datech. Z tohoto rozložení lze 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 0.

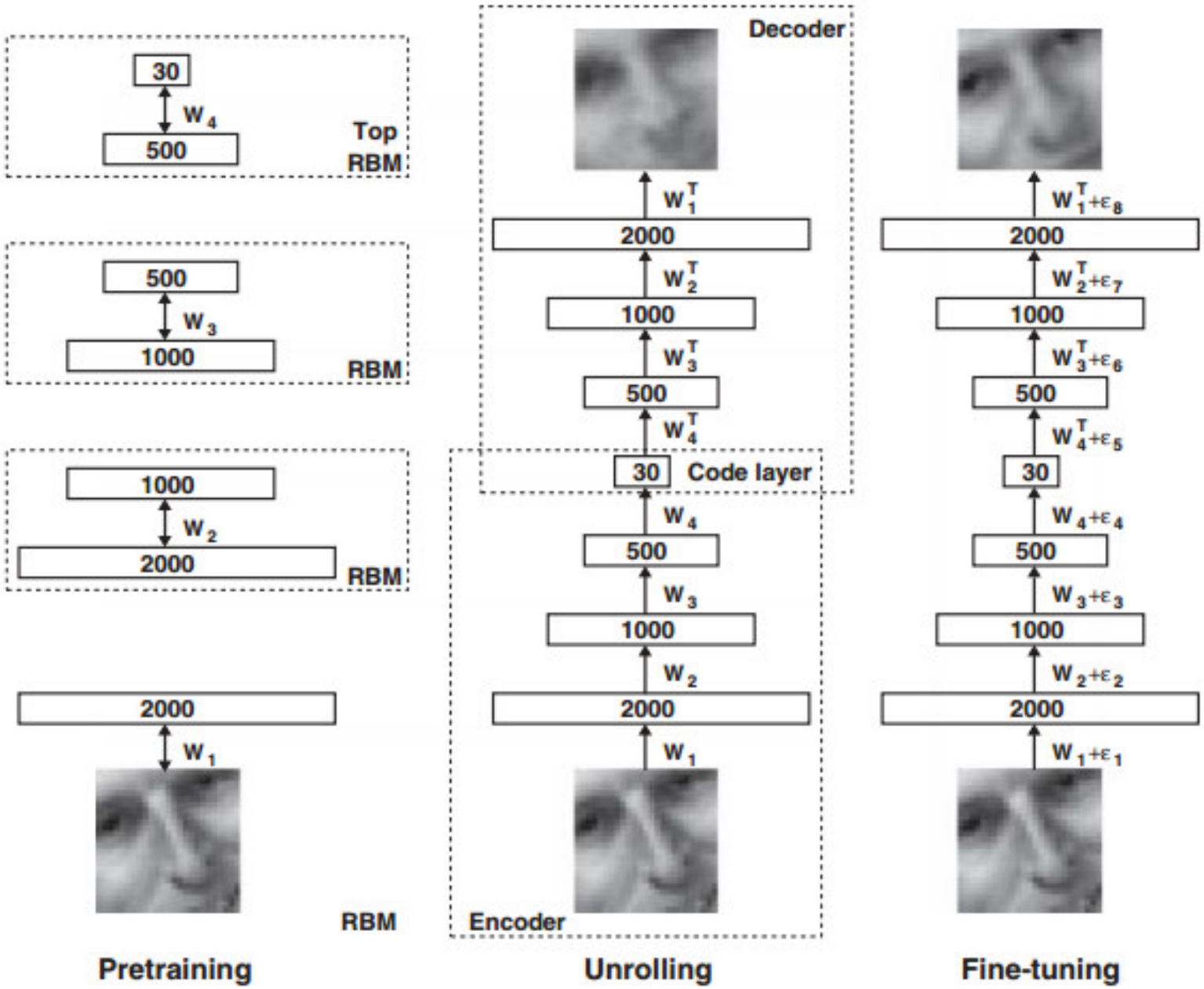
Aplikace – redukce dimenze

- Redukce dimenze dat: Tedy zobrazení R z $\mathbb{R}^n$ do $\mathbb{R}^m$ takové, že
  - $m < n$,
  - pro každý vzor $\vec{x}$ platí, že $\vec{x}$ je možné "rekonstruovat" z $R(\vec{x})$.
- Standardní metoda PCA (exituje mnoho lineárních i nelineárních variant)

Rekonstrukce – PCA

1024 pixelů komprimováno do 100 dimenzí (tj. 100 čísel).

Redukce dimenze pomocí hlubokých sítí



Obrázky – Předtrénování

- **Data:** 165 600 černobílých obrázků o rozměrech 25×25, střední intenzita bodu 0, rozptyl 1.

  Obdrženo z Olivetti Faces databáze obrázků 64×64 standardními úpravami.
- 103 500 tréninková sada, 20 700 validační, 41 400 testovací
- **Síť:** Struktura sítě 2000-100-500-30, trénink pomocí postupného vrstvení RBM.

Poznámky:

Trénink nejnižší skryté vrsty (2000 neuronů): Hodnoty pixelů "rozmlženy" Gaussovským šumem, rychlost učení nízká: 0.001, počítáno 200 iterací

Ve všech vrstvách kromě nejvyšší jsou hodnoty skrytých neuronů při tréninku binární (v nejvyšší jsou hodnoty vypočteny ze sigmoidální pravděpodobnosti přidáním šumu)

Hodnoty viditelných neuronů jsou při tréninku reálné z intervalu $[0, 1]$ (zde je mírná odchylka od našeho algoritmu)

Obrázky – Doladění

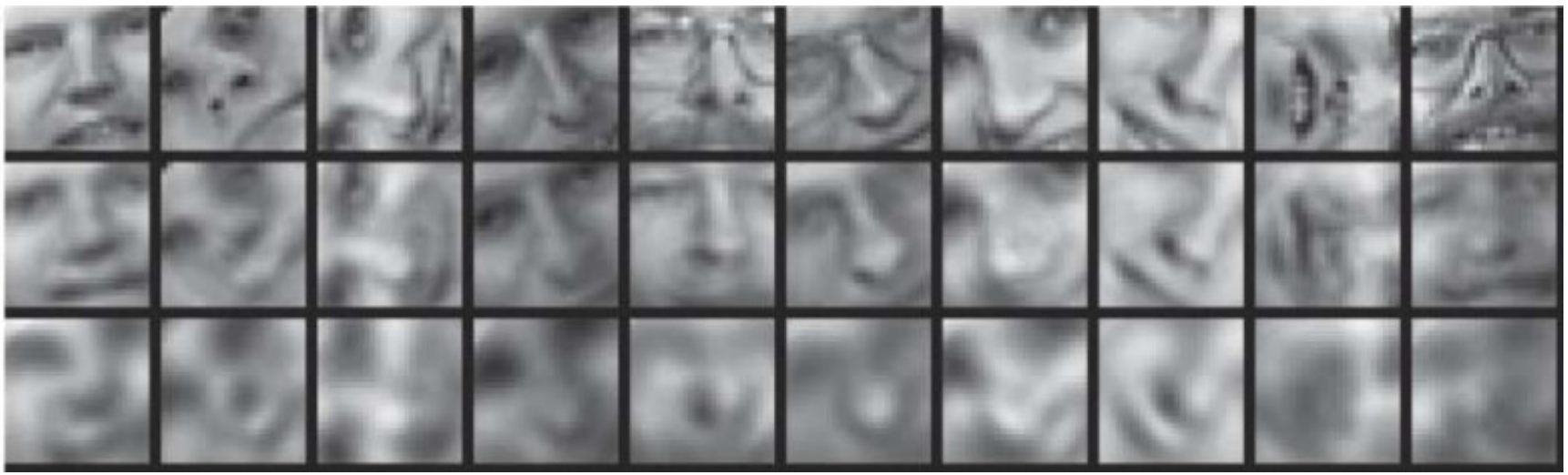
- Stochastické aktivace nahrazeny deterministickými

  Tj. hodnota skrytých neuronů není výsledkem náhodného pokusu, ale přímo výpočtu sigmoid udávajících pravděpodobnost.
- Zpětná propagace
- Chybová funkce cross-entropy

$$-\sum_i p_i \ln \hat{p}_i - \sum_i (1 - p_i) \ln(1 - \hat{p}_i)$$

kde p_i je intenzita i-tého pixelu ve vstupu a $\hat{p}_i$ v rekonstrukci.

Výsledky



1. Originál
2. Rekonstrukce hlubokou sítí (redukce na 30-dim)
3. Rekonstrukce PCA (redukce na 30-dim)