SNE 12 (MB)

7. Maszyna Boltzmanna i jej uczenie

7.1. Maszyna Boltzmanna

Maszyna Boltzmanna (skrótowo, MB) składa się z n neuronów. Każdy neuron jest połączony do wszystkich innych neuronów (nawet do siebie) tak jak w sieci Hopfielda. Niech

$$x = (x_1, x_2, \ldots, x_n)$$

będzie oznaczał stan całego układu: $x_i = 0$ lub 1 ($i = 1, 2, \ldots, n$). Każdy neuron "i" bierze wartość $x_i = 0$ lub 1. Więc w sumie istnieje 2^n możliwych stanów.

Oznaczenia Przypomnimy, że $P(A)$ oznacza prawdopodobieństwo zdarzenia A a $P(A|B)$ oznacza prawdopodobieństwo warunkowe zajścia zdarzenia A pod warunkiem zajścia zdarzenia B.

MB działa na dyskretnych chwilach $t = 0, 1, 2, \ldots$. Niech

$$x(t) = (x_i(t))_{i=1}^n = (x_1(t), \ldots, x_n(t))$$

$$x(t+1) = (x_i(t+1))_{i=1}^n = (x_1(t+1), \ldots, x_n(t+1))$$

będą stanami MB w chwili t i $t+1$ odpowiednie. Niech

$$f(u) = \frac{1}{1 + e^{-\frac{u}{T}}}.$$

Każdy neuron "i" przyjmuje sygnał x_j od neuronu "j" i sumuje z wagą w_{ij} i odejmuje wartość progową θ_i:

$$u_i = \left\{ \sum_{j=1}^n w_{ij} x_j \right\} - \theta_i.$$

Jak uruchamiamy MB, dostajemy ciąg stanów.

$$x(0) \to x(1) \to x(2) \to \cdots \to x(t) \to x(t+1) \to \cdots$$

Powyżej stan $x(0)$ jest generowany losowo. Stan $x(t+1)$ zależy tylko od poprzedniego stanu $x(t)$ (proces Markowa).

Algorytm MB
$t = 0$, $i = 1, \ldots, n$

$$x_i(0) = \begin{cases} 0 & \text{z prawdopodowieństwem } \frac{1}{2} \\ 1 & \text{z prawdopodowieństwem } \frac{1}{2} \end{cases}$$

$t = 0, 1, 2, \ldots, \quad i = 1, \ldots, n$

$$u_i(t) = \left\{ \sum_{j=1}^n w_{ij} x_j(t) \right\} - \theta_i$$

$$x_i(t+1) = \begin{cases} 1 & \text{z prawdopodowieństwem } f(u_i(t)) \\ 0 & \text{z prawdopodowieństwem } 1 - f(u_i(t)) \end{cases}$$

Algorytm$'$ MB

Warunek początkowy $x(0) = (x_i(0))_{i=1}^n = (x_1(0), \dots, x_n(0))$

Dla każdego $i = 1, \dots, n$, losować $\alpha_i = 1$ lub 0 ($P(\alpha_i = 1) = P(\alpha_i = 0) = \frac{1}{2}$).
$\alpha_i = \mathrm{rand}()\%2$

$$x_i(0) = \alpha_i$$

Przejście z $x(t) = (x_i(t))_{i=1}^n = (x_1(t), \dots, x_n(t))$
do $x(t+1) = (x_i(t+1))_{i=1}^n = (x_1(t+1), \dots, x_n(t+1))$

Dla każdego $i = 1, \dots, n$, losować $\beta_i \in [0, 1]$ z jednostajnym prawdopodobieństwem (tj. $P(\beta_i \in [x, x + \Delta x]) = \Delta x$).
$\beta_i = \mathrm{rand}()/\mathrm{Rand\_Max}$, $\beta_i = (\mathrm{rand}()\%100)/99$

$$x_i(t+1) = \begin{cases} 1 & \text{gdy } 0 \le \beta_i \le f(u_i(t)) \\ 0 & \text{gdy } f(u_i(t)) \le \beta_i \le 1 \end{cases}$$

gdzie

$$u_i(t) = \left\{ \sum_{j=1}^n w_{ij} x_j(t) \right\} - \theta_i,$$

$$f(u_i(t)) = \frac{1}{1 + e^{-\frac{u_i(t)}{T}}}.$$

Uwaga.
T jest analogią temperatury.
(1) Jeśli $T \gg 0$, to $f(u_i) \approx \frac{1}{2}$. Wówczas MB działa losowo.
(2) Jeśli T jest zbyt małe (niska temperatura), to MB działa według

$$x_i(0) = \begin{cases} 1 & \text{gdy } u_i(t) > 0 \\ 0 & \text{gdy } u_i(t) < 0. \end{cases}$$

To jest sieć Hopfielda.

Twierdzenie 7.1.1.
(1) Istnieje granica

$$\lim_{t \to \infty} P(x(t) = x | x(0) = x_0)$$

i nie jest zależny od warunku początkowego x_0. Czyli granica zależy tylko od $x = (x_1, \dots, x_n)$. Tą granicę napiszemy $P(x)$:

$$P(x) = \lim_{t \to \infty} P(x(t) = x | x(0) = x_0).$$

(2) $P(x)$ jest podany wzorem

$$P(x) = c e^{-\frac{E(x)}{T}},$$

gdzie energia

$$E(x) = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j + \sum_{i=1}^n \theta_i x_i$$

dla $x=(x_1,\dots,x_n)$.

Uwaga.
(1) Uważmy, że $\sum_{x\in\{0,1\}^n} ce^{-\frac{E(x)}{T}}=1$. Więc $c=\frac{1}{\sum_{x\in\{0,1\}^n} e^{-\frac{E(x)}{T}}}$.
(2) $x=(x_1,\dots,x_n)$ dla $t\to\infty$ nazywamy stanem równowagi. Jesli $E(x_1)>E(x_2)$, to $e^{-\frac{E(x_2)}{T}}>e^{-\frac{E(x_1)}{T}}$, tzn. $P(x_2)>P(x_1)$. Więc stan z minimalną enegią pojawia się najczęściej w toku MB. Czyli MB służy do minimalizacji energii $E=E(x)$.

Twierdzenie 7.1.2.
(A)=Twierdzenie 7.1.1 (Relaksacja)

$$\lim_{t\to\infty} P(x(t)=x|x(0)=x_0)=ce^{-\frac{E(x)}{T}}$$

(B) (Wyżarzanie, *ang.* annealing)

$$T(t)\geq\frac{c}{\ln(t+1)},\quad T(t)\searrow 0 \Longrightarrow \lim_{t\to\infty} P(x(t)=x_1\in\Omega|x(0)=x_0)=\frac{1}{\#\Omega}$$

$(\Omega=\{x_1\,|\,\forall\, y\,(E(x_1)\leq E(y))\})$

(C) (Ergodyczność)

$$\lim_{n\to\infty}\frac{1}{n+1}\sum_{t=0}^{n} Y(x(t))=\sum_{x_1\in\Omega} P(x_1)Y(x_1)$$

7.2. Uczenie Maszyny Boltzmanna (D. H. Ackley, G. E. Hinton, T. J. Sejnowski, 1985)

Podzielimy sieć n neuronów na dwie części: (i) warstwa widzialna (visual units) V (ii) warstwa ukryta (hidden units) H.

$$V=\{1,\dots,\nu\},\quad H=\{\nu+1,\dots,\nu+\mu\}\quad (n=\nu+\mu)$$

Zgodnie z tym podzielimy stan x.

$$x=(x_V;x_H)=(\underbrace{x_1,\dots,x_\nu}_{\text{visual}};\underbrace{x_{\nu+1},\dots,x_{\nu+\mu}}_{\text{hidden}})$$

$$x_V=(x_1,\dots,x_\nu),\quad x_H=(x_{\nu+1},\dots,x_{\nu+\mu})$$

W sumie mamy $2^{\nu+\mu}=2^n$ stanów.

$$x^{\alpha,\beta}=(x_V^\alpha;x_H^\beta)$$
$$x_V^\alpha=(x_1^\alpha,\dots,x_\nu^\alpha)\quad (\alpha=1,\dots,2^\nu)$$
$$x_H^\beta=(x_1^\beta,\dots,x_\mu^\beta)\quad (\beta=1,\dots,2^\mu)$$

Algorytm uczenia MB
Uczenie MB jest wykonane w następujący sposób.

Dzień 1: Ustalanie początkowych (dowolnych) w_{ij} (Rano) $\to$ Procedura w dzień $\to$ Procedura w nocy $\to$

Dzień 2: Regulowanie w_{ij} (Rano) $\to$ Procedura w dzień $\to$ Procedura w nocy $\to$
Dzień 3: Regulowanie w_{ij} (Rano) $\to$ Procedura w dzień $\to$ Procedura w nocy $\to$
... $\to$
Ostatni dzień n: Regulowanie w_{ij} (Rano) $\to$ Procedura w dzień $\to$ Procedura w nocy

Procedura w dzień
Warstwa widzialna V nie działa jako MB. Prawdopodobieństwa sygnałów V podane przymusowo z zewnątrz.
$(t = 0, \ldots, t_{\max})$

$$P(x_V(t) = x_V^\alpha) = P(x_V^\alpha)$$

Warstwa ukryta H działa jako MB.
$(t = 0,\ i = \nu + 1, \ldots, n)$

$$x_i(0) = \begin{cases} 0 & \text{z prawdopodowieństwem } \frac{1}{2} \\ 1 & \text{z prawdopodowieństwem } \frac{1}{2} \end{cases}$$

$(t = 0, \ldots, t_{\max} - 1,\ \ i = \nu + 1, \ldots, n)$

$$u_i(t) = \left\{ \sum_{j=\nu+1}^{n} w_{ij} x_j(t) \right\} - \theta_i$$

$$x_i(t+1) = \begin{cases} 1 & \text{z prawdopodowieństwem } f(u_i(t)) \\ 0 & \text{z prawdopodowieństwem } 1 - f(u_i(t)) \end{cases}$$

Obliczmy

$$p_{ij} = \lim_{t\to\infty} P(x_i(t) = x_j(t) = 1) = \lim_{t_{\max}\to\infty} \frac{1}{t_{\max} + 1} \sum_{t=0}^{t_{\max}} x_i(t) x_j(t).$$

Jeśli $t_{\max} \gg 0$, to praktycznie

$$p_{ij} \approx \frac{1}{t_{\max} - \lfloor \frac{t_{\max}}{2} \rfloor} \sum_{t=\lfloor \frac{t_{\max}}{2} \rfloor + 1}^{t_{\max}} x_i(t) x_j(t).$$

Uwaga.

$$\lim_{t\to\infty} P(x_i(t) = x_j(t) = 1) = \sum_{\alpha=0,1} \sum_{\beta=0,1} \lim_{t\to\infty} P(x_i(t) = \alpha, x_j(t) = \beta)\alpha\beta = \mathbb{E}(x_i x_j)$$

Zob. Twierdzenie 7.1.2 C (Ergodyczność).
Procedura w nocy
Cały układ $V \cup H$ działa jako MB.
$(t = 0, \ldots, t_{\max} - 1,\ \ i = 1, \ldots, n)$

$$u_i(t) = \left\{ \sum_{j=\nu+1}^{n} w_{ij} x_j(t) \right\} - \theta_i$$

$$x_i(t+1) = \begin{cases} 1 & \text{z prawdopodowieństwem } f(u_i(t)) \\ 0 & \text{z prawdopodowieństwem } 1 - f(u_i(t)) \end{cases}$$

Obliczmy

$$p'_{ij} = \lim_{t\to\infty} P(x_i(t) = x_j(t) = 1) = \lim_{t_{\max}\to\infty} \frac{1}{t_{\max}+1} \sum_{t=0}^{t_{\max}} x_i(t)x_j(t).$$

Jeśli $t_{\max} \gg 0$, to praktycznie

$$p'_{ij} \approx \frac{1}{t_{\max} - \lfloor \frac{t_{\max}}{2} \rfloor} \sum_{t=\lfloor \frac{t_{\max}}{2} \rfloor + 1}^{t_{\max}} x_i(t)x_j(t).$$

Regulowanie wag w_{ij} (Rano następunego dnia)

$$w_{ij}^{\text{dzisiaj}} = w_{ij}^{\text{wczoraj}} + \frac{c}{T}(p_{ij} - p'_{ij}) \qquad (c > 0)$$

Procedura ostatniego dnia n

$$P(x_V^\alpha) = \lim_{t\to\infty} P(x(t) = x_V^\alpha | x(0) = x_0)$$

$=$ prawdopodowieństwo, że stan x_V^α pojawia się w stanie równowagi w dzień

$$\sum_{\alpha=1}^{2^\nu} P(x_V^\alpha) = 1$$

Procedura ostatniej nocy n

$$P'(x_V^\alpha) = \lim_{t\to\infty} P(x(t) = x_V^\alpha | x(0) = x_0)$$

$=$ prawdopodowieństwo, że stan x_V^α pojawia się w stanie równowagi w nocy

$$\sum_{\alpha=1}^{2^\nu} P'(x_V^\alpha) = 1$$

Twierdzenie 7.2.1. Przy $n \to \infty$,

$$P'(x_V^\alpha) \to P(x_V^\alpha).$$

Komentarz o dowodzie Twierdzenia 7.2.1.
W regulacji wag w_{ij}, minimalizujemy Kullback-Leibler divergence (dywergencja Kullbacka-Leiblera).

$$D(P', P) = \sum_{x_V^\alpha \in \{0,1\}^\nu} P'(x_V^\alpha) \log \frac{P'(x_V^\alpha)}{P(x_V^\alpha)}$$

Można pokazać, że

$$\frac{\partial D(P', P)}{\partial w_{ij}} = -\frac{1}{T}(p_{ij} - p'_{ij}).$$

7.3. Uczenie MB z wejściami i wyjściami (input-output)

Podzielimy widzialną warstwę V na dwie części: (i) warstwa widzialna wejściowa U (input) (ii) warstwa widzialna wyjściowa Y (output).

$$U = \{1, \dots, \ell\}, \quad Y = \{\ell+1, \dots, \ell+m\}, \quad \ell + m = \nu, \quad U \cup Y = V$$

Zgodnie z tym podzielimy stan.

$$x = (x_U; x_Y; x_H) = (x_1, \dots, x_\ell; x_{\ell+1}, \dots, x_{\ell+m}; x_{\ell+m+1}, \dots, x_{\ell+m+\mu})$$

$$x_U = (x_1, \dots, x_\ell), \quad x_Y = (x_{\ell+1}, \dots, x_{\ell+m}), \quad x_H = (x_{\ell+m+1}, \dots, x_{\ell+m+\mu})$$

W sumie mamy $2^{\ell+m+\mu}$ stanów.

$$x^{a,b,c} = (x_U^a, x_Y^b, x_H^c)$$

$$x_U^a = (x_1^a, \dots, x_\ell^a) \quad (a = 1, \dots, 2^\ell)$$

$$x_Y^b = (x_{\ell+1}^b, \dots, x_{\ell+m}^b) \quad (b = 1, \dots, 2^m)$$

$$x_H^c = (x_{\ell+m+1}^c, \dots, x_{\ell+m+\mu}^c) \quad (c = 1, \dots, 2^\mu)$$

Twierdzenie 7.3.1. Wykonujemy procedurę uczenia MB tak jak w §7.2. Niech

$$P(x_Y = x_Y^b | x_U = x_U^a) = \lim_{t\to\infty} P(x_Y(t) = x_Y^b | x_U(t) = x_U^a(t))$$

będzie prawdopodobieństwem warunkowym zajścia zdarzenia, że $x_Y = x_Y^b$ pod warunkiem zajścia zdarzenia, że $x_U = x_U^a$ w stanie równowagi w dzień ostatniego dnia n.

Niech

$$P'(x_Y = x_Y^b | x_U = x_U^a) = \lim_{t\to\infty} P(x_Y(t) = x_Y^b | x_U(t) = x_U^a(t))$$

będzie prawdopodobieństwem warunkowym zajścia zdarzenia, że $x_Y = x_Y^b$ pod warunkiem zajścia zdarzenia, że $x_U = x_U^a$ w stanie równowagi w nocy ostatniego dnia n.

Wówczas mamy przy dzień $n \to \infty$

$$P'(x_Y = x_Y^b | x_U = x_U^a) \to P(x_Y = x_Y^b | x_U = x_U^a).$$

Dowód. Z definicji prawdopodowieństwa warunkowego

$$P'(x_Y = x_Y^b | x_U = x_U^a) = \frac{P'(x_U = x_U^a \wedge x_Y = x_Y^b)}{P'(x_U = x_U^a)},$$

$$P(x_Y = x_Y^b | x_U = x_U^a) = \frac{P(x_U = x_U^a \wedge x_Y = x_Y^b)}{P(x_U = x_U^a)}.$$

Za pomocą Twierdzenia 7.2.1 mamy

$$P'(x_U = x_U^a \wedge x_Y = x_Y^b) \to P(x_U = x_U^a \wedge x_Y = x_Y^b) \quad \text{i} \quad P'(x_U = x_U^a) \to P(x_U = x_U^a). \qquad \blacksquare$$

7.4. Ograniczona Maszyna Boltzmanna (Restricted Boltzmann Machine, RBM)

RBM ma strukturę taką, że neurony są połączone tylko między warstwami widzialną V a ukrytą H. Nie ma połączeń (tj. $w_{ij} = 0$) w wewnątrz V, H, żeby uniknąć złożoność regulowania wag.

$$\begin{array}{lll} H \ \bigcirc & \cdots & \text{Hidden units} \\ \quad\ \updownarrow & \cdots & \qquad\qquad\qquad (\updownarrow = \uparrow + \downarrow) \\ V \ \bigcirc & \cdots & \text{Visual units} \end{array}$$

7.5 Maszyny Boltzmanna związane do uczenia głębokiego

- Deep Boltzmann Machine (DBM)

$$
\begin{array}{cccl}
\mathrm{h}^3 & \bigcirc & \cdots & \text{hidden layer 3} \\
 & \updownarrow & \cdots & \\
\mathrm{h}^2 & \bigcirc & \cdots & \text{hidden layer 2} \\
 & \updownarrow & \cdots & \\
\mathrm{h}^1 & \bigcirc & \cdots & \text{hidden layer 1} \\
 & \updownarrow & \cdots & \\
\mathrm{v} & \bigcirc & \cdots & \text{visual layer (observable)}
\end{array}
$$

- Deep Belief Network (DBN)

$$
\begin{array}{cccl}
\mathrm{h}^3 & \bigcirc & \cdots & \text{hidden layer 3} \\
 & \updownarrow & \cdots & \\
\mathrm{h}^2 & \bigcirc & \cdots & \text{hidden layer 2} \\
 & \downarrow & \cdots & \\
\mathrm{h}^1 & \bigcirc & \cdots & \text{hidden layer 1} \\
 & \downarrow & \cdots & \\
\mathrm{v} & \bigcirc & \cdots & \text{visual layer (observable)}
\end{array}
$$

- Convolutional DBM/DBN = CNN + DBM/DBN