Sieci dynamiczne

Pamięć asocjacyjna

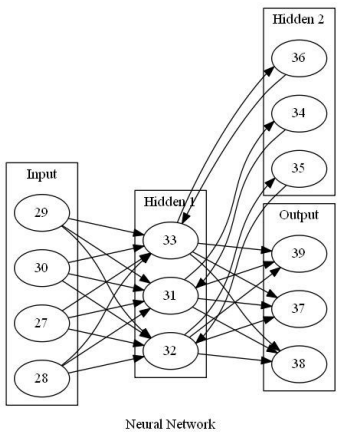
Plan

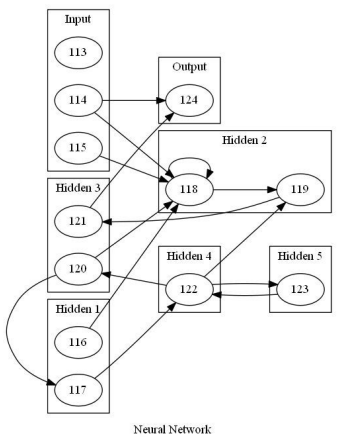
- Sieci dynamiczne: sieci ze sprzężeniami zwrotnymi
- Modele pamięci asocjacyjnej
- Model Hopfielda, siec Hamminga, BAM

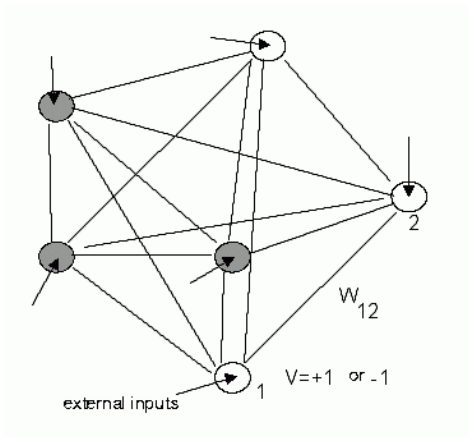
Sieci ze sprzężeniami zwrotnymi

Sieci rekurencyjne

- połączenia tworzą cykle w grafie połączeń
- skomplikowana dynamika - trudna do analizy
- zbliżone do sieci biologicznych, w układach biologicznych neurony mają silne sprzężenia zwrotne







Sieci ze sprzężeniami zwrotnymi

Najprostsze modele sieci z rekurencją:

- sieci Hopfielda,
- sieć Hamminga.
- BAM
- Restricted Boltzman Machines (RBM)

Modele złożone:

- RTRN - Real Time Recurrent Network, przetwarzająca sygnały w czasie rzeczywistym;
- sieć Elmana i inne o uproszczonej strukturze rekurencji
- RCC - Recurrent Cascade Correlation
- LSTM, Long Short Term Memory

Model Hopfielda (1982, 1984)

Model pamięci autoasocjacyjnej Hopefielda:

- pamięć skojarzeniowa - odtwarza zapamiętane wzorce zbliżone do sygnału wejściowego
- wszystkie neurony są ze sobą wzajemnie połączone, brak warstw
- macierz wag połączeń jest symetryczna, $w_{ij} = w_{ji}$. Symetria wag jest wygodna, upraszcza analizę sieci i pozwala wprowadzić funkcję energii; jednak jest nierealistyczna z biologicznego punktu widzenia.

Dyskretne stany neuronów, zmiany następują w dyskretnych jednostkach czasu

$$y_i = \pm 1 = \mathrm{sgn}(\mathbf{w}^T \mathbf{x})$$

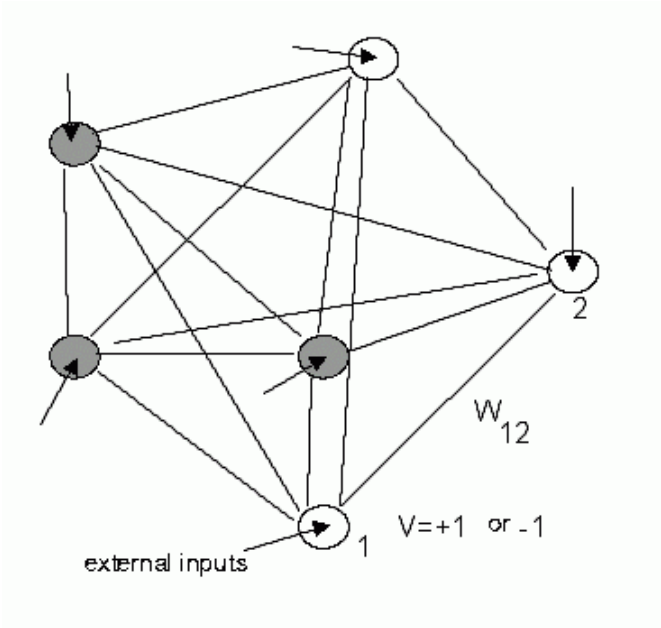
Bliżej bioligii są realizacje sieci o ciągłych funkcjach aktywacji, w których zmiany aktywacji nastepuja w sposób ciągły

Model Hopfielda

- każdy neuron jest wejściem i wyjściem sieci

$$y_i(t+1) = \mathbf{sgn}\left(\sum_j w_{ij} y_j(t)\right)$$

- brak sprzężenia własnego $w_{ii} = 0$
- wagi symetryczne $w_{ij} = w_{ji}$
- stany sieci $\mathbf{y}$ można utożsamić z wierzchołkami N wymiarowej kostki



Dwa tryby pracy

- uczenie (zapamiętywanie)
- odtwarzanie wzorców

Dynamika odtwarzania wzorców

- dla ustalonych wag $\mathbf{W}$ w chwili $t = 0$ doprowadzamy sygnał wejściowy $\mathbf{y}(0) = \mathbf{x}$
- w kolejnym kroku t aktualizowane są stany neuronów

$$y_i(t+1) = \mathrm{sgn}\left(\sum_j w_{ij} y_j(t)\right) \qquad i = 1, \ldots, N$$

- proces asynchroniczny - aktualizowane jest wyjście jednego losowo wybranego neuronu, jego wyjście zależy od już zaktualizowanych neuronów,
- proces zatrzymuje się gdy osiągnięto stan stacjonarny (minimum lokalne), brak zmian wyjść przy pełnym cyklu

$$\mathbf{y}(t+1) = \mathbf{y}(t)$$

- asynchroniczność minimalizuje występowanie cykli i sieć zbiega do atraktora punktowego
- autoasocjacja - wyjście odtwarza $\mathbf{x}$

Minimalizacja energii

Dla sieci o symetrycznych wagach taka dynamika prowadzi do minimalizacji funkcji energii

$$E(W) = -\frac{1}{2}\langle \mathbf{y}|\mathbf{W}|\mathbf{y}\rangle = -\frac{1}{2}\sum_{i\neq j} w_{ij} y_i y_j$$

Zmiana energii w czasie iteracji jest mniejsza od zera

$$\Delta E = -\Delta y_i \sum_j w_{ij} y_j = -\Delta y_i I_i$$

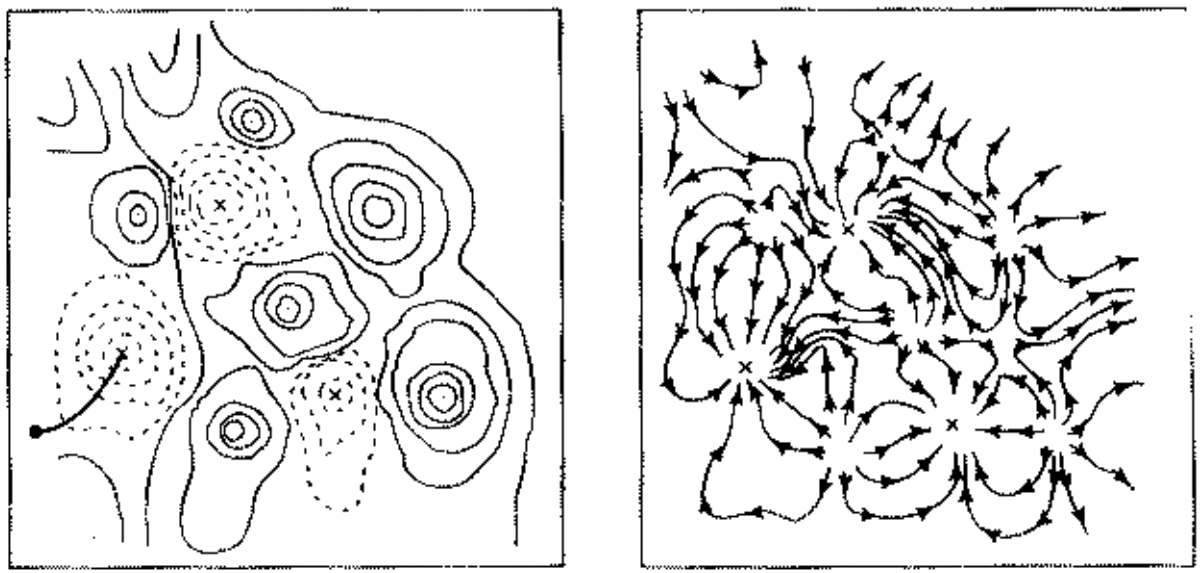
- jeżeli $I_i \geq 0$ to y_i nie może zmaleć
- jeśli $I_i < 0$ to $\Delta y_i < 0$

Każda zmiana stanu neuronu może tylko obniżyć energię sieci

Atraktory

Dynamika: ruch po hiperpowierzchni energii, zależnej od potencjałów neuronów, aż do osiągnięcia lokalnego minimum na takiej powierzchni.

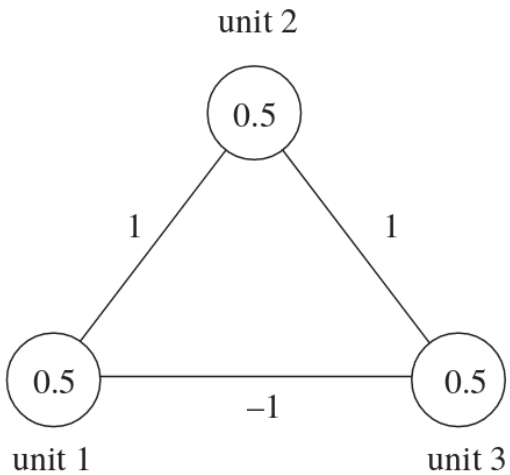
Jeśli y_i dyskretne $-1, +1$ to ruch po wierzchotkach hiperkostki

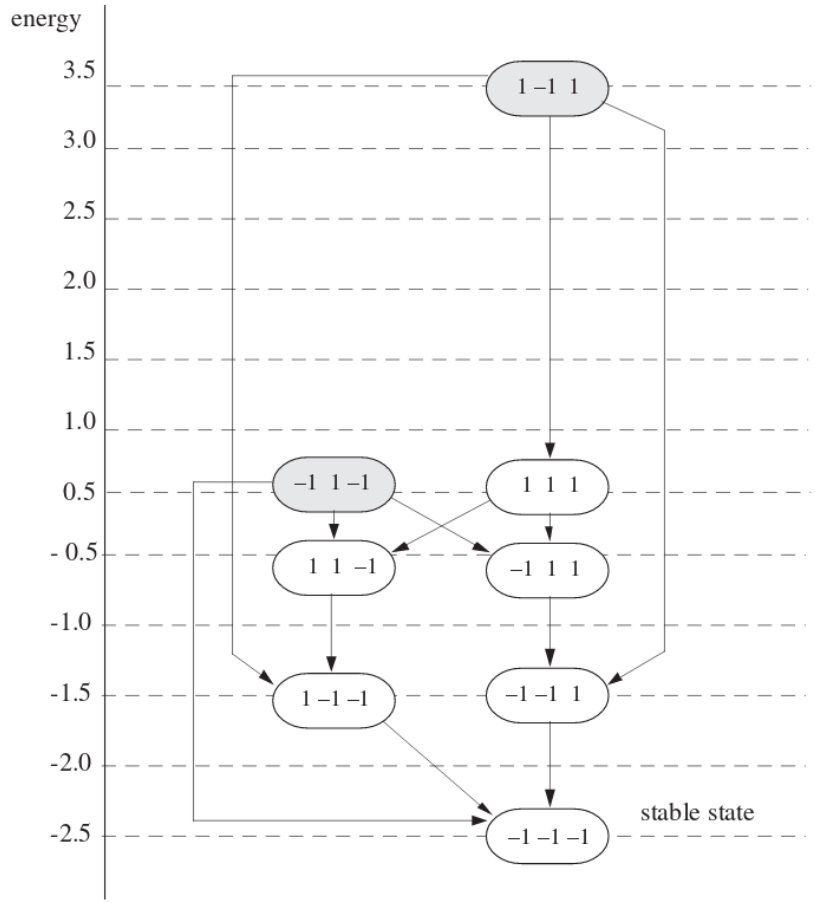


Obszary atrakcji sieci rekurencyjnej: linie ekwipotencjalne i kierunki zmian w trakcie uczenia

Rys: Ossowski, Sieci Neuronowe

Przykład: 3 neurony

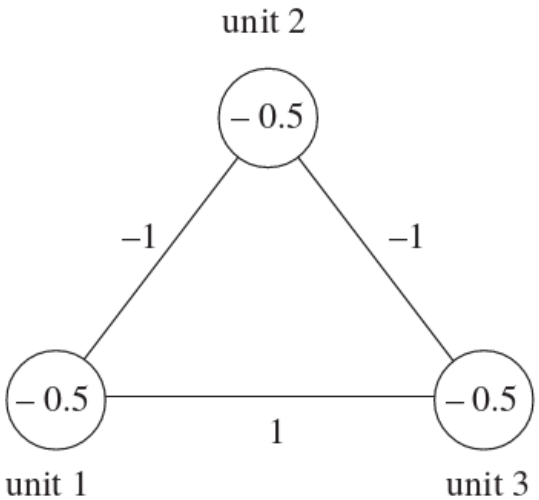


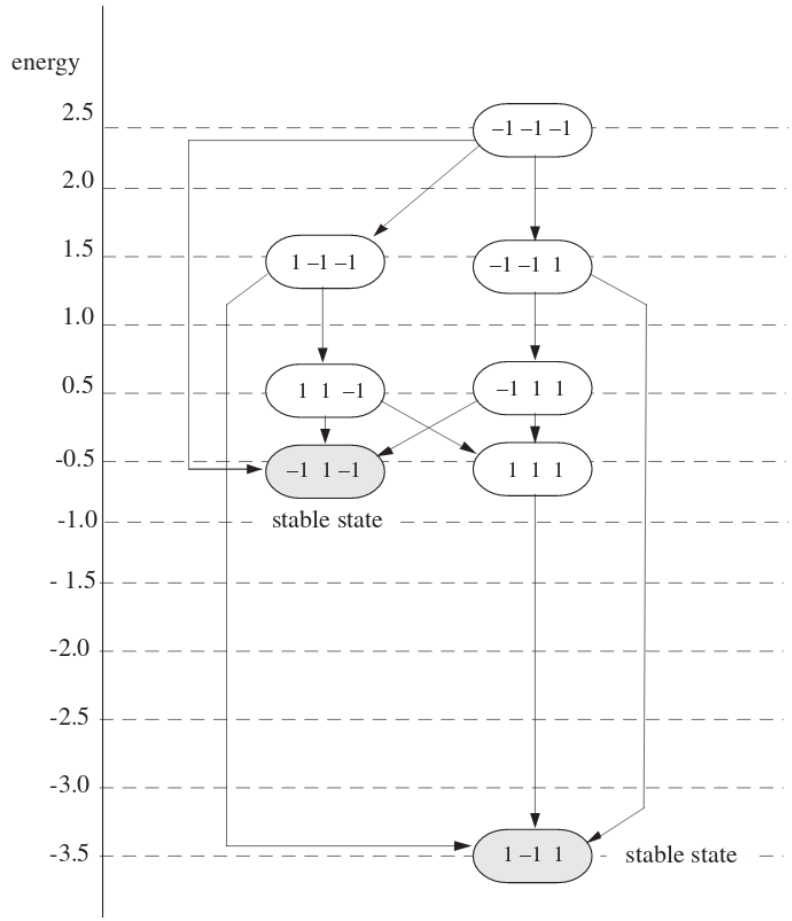


$$E = -y_1y_2 - y_2y_3 + y_1y_3 - 0.5y_1 - 0.5y_2 - 0.5y_3$$

Rys: Rojas, Neuroal Netowrks

Przykład: 3 neurony





$$E = y_1y_2 + y_2y_3 - y_1y_3 + 0.5y_1 + 0.5y_2 - 0.5y_3$$

Rys: Rojas, Neuroal Netowrks

Zastosowanie praktyczne

- pamięć asocjacyjna
  zapamiętuje wzorce $\mathbf{x}_1, \ldots, \mathbf{x}_p$, przy prezentacji wektora wejściowego $\mathbf{x}_i$ odpowiedzią sieci będzie jeden z zapamiętanych wzorców, najbardziej „podobny” do sygnału wejściowego.
  Każdy zapamiętany wzorzec odpowiada minimum lokalnemu energii
- problemy optymalizacyjne - odpowiedni konstrukcja sieci i funkcji energii pozwala znaleźć rozwiązania problemów optymalizacyjnych, także NP-trudnych
  - problem komiwojażera,
  - gospodarowanie zasobami,
  - optymalizacja połączeń w centralach telefonicznych,
  - rozmieszczanie układów scalonych,
  - rozwiązywanie problemów programowania liniowego i nieliniowego

Pamięć asocjacyjna

- pamięć autoasocjacyjna - skojarzenie tego samego obiektu (model Hopfielda)
- pamięć heteroasoscjacyjna - skojarzone są dwa różne obiekty (dla **a** sieć odpowiada **b**), np. BAM, sieć Hamminga

Pamięć asocjacyjna vs. RAM

- adresowanie kontekstowe - wzorzec na wyjściu uzyskiwany jest na podstawie odległości sygnału wejściowego od zapamiętanego wzorca
- pamięć rozproszona - brak wyraźnie określonego miejsca przechowywania wzorca
- lokalne uszkodzenie nie powinno całkowicie blokować możliwości odczytu

Odległość Hamminga

Odległość Hamminga dla wielkości binarnych
$x_i = \{0, 1\}, y_i = \{0, 1\}$

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \left[x_i \left(1 - y_i\right) + \left(1 - x_i\right) y_i \right]$$

Dla wartości bipolarnych $x_i, y_i = \{+1, -1\}$

$$(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \left(n - \sum_{j=1}^{n} x_i y_j \right)$$

liczba różniących się bitów

Pamięć asocjacyjna będzie odtwarzać wektory zapamiętane najbliższe względem wektora wejściowego zgodnie z odległością Hamminga

Uczenie pamięci asocjacyjnej

Dla pojedynczego wzorca **x** sieć powinna odtwarzać sygnał wejściowy

$$x_i = \mathrm{sgn}\left(\sum_{j=1}^{N} w_{ij} x_j\right)$$

wystarczy zażądać by:

$$w_{ij} \sim x_i x_j; \qquad \text{np. } w_{ij} = \frac{1}{N} x_i x_j \qquad \text{reguła Hebba}$$

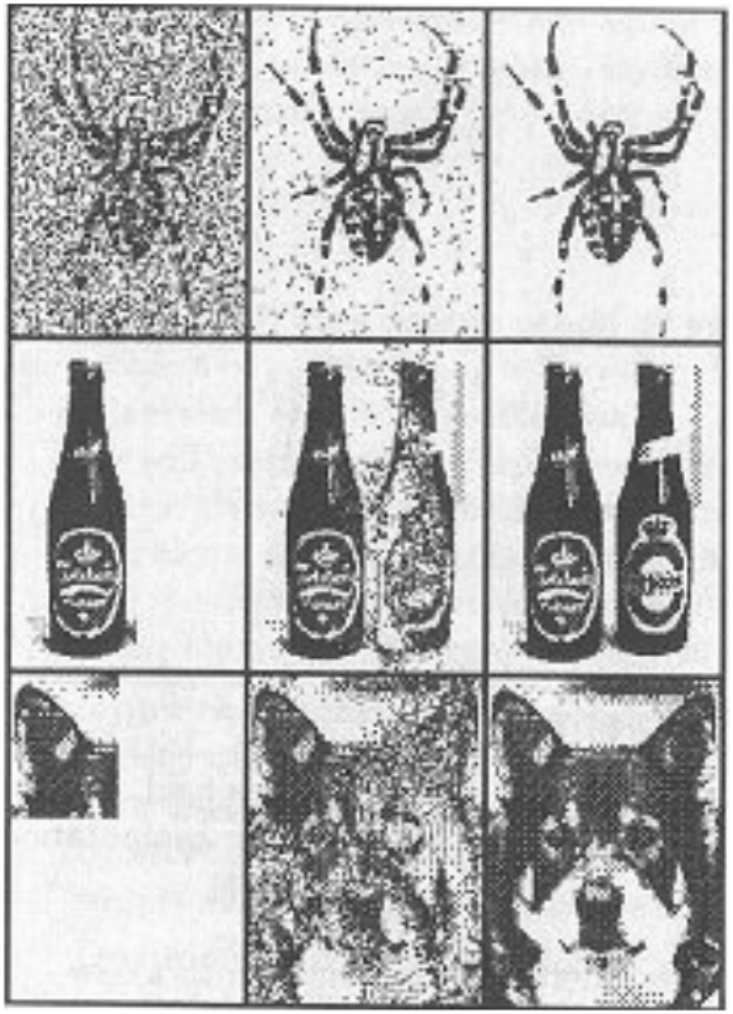
$$\frac{1}{N}\left(\sum_{j=1}^{N} x_i x_j x_j\right) = \frac{1}{N}\left(\sum_{j=1}^{N} x_i x_j^2\right) = \frac{1}{N}\left(\sum_{j=1}^{N} x_i\right) = x_i$$

Uczenie pamięci asocjacyjnej

Dla wielu wzorców $\mathbf{x}^i, i = 1, \ldots, p$ korzystamy z reguły Hebba uśredniając:

$$W_{ij} = \frac{1}{N} \sum_{k=1}^{p} x_i^{(k)} x_j^{(k)}$$

Przykłady działania



odtwarzanie niekompletnych lub zaszumionych danych

Stabilność pamięci

Stabilność działania wymaga

$$\text{sgn}\left(\sum_{j=0}^{N} w_{ij} x_j^{(l)}\right) = \text{sgn}\left(\frac{1}{N}\sum_{j=0}^{N}\sum_{k=1}^{p} x_i^{(k)} x_j^{(k)} x_j^{(l)}\right) = x_i^{(l)}$$

co można zapisać

$$\text{sgn}\left(x_i^{(l)} + \frac{1}{N}\sum_{j=0}^{N}\sum_{k\neq l} x_i^{(k)} x_j^{(k)} x_j^{(l)}\right) = \text{sgn}\left(x_i^{(l)} + C\right)$$

gdzie C nazywamy **przesłuchem**

Stabilność osiągamy gdy składnik przesłuchu C jest na tyle mały aby nie zmienić znaku $x_i^{(l)}$, tzn. $|C| < |x_i^{(l)}|$

Pojemność modelu

- Pojemność określa maksymalną liczbę odtwarzanych wzorców przy określonym poziomie błędu
- 2^N możliwych stanów sieci binarnej złożonej z N neuronów
- zbyt wiele wzorców $\Rightarrow$ chaos, zapominanie, zwiększając liczbę wzorców rośnie prawdopodobieństwo przekłamań
- dla uczenia regułą Hebba liczba poprawnie pamiętanych wzorców przy prawdopodobieństwie błędu 0.37% wynosi $0.138N$
- metoda rzutowania pozwala uzyskać pojemność maksymalną $N - 1$

Stany fałszywe

W sieci mogą pojawiać się stany fałszywe lub przekłamania pamięci

- funkcja energii jest symetryczna względem polaryzacji (negatywy stanów o tej samej energii)
- występuje mieszanie różnych składowych zapamiętanych wzorców i tworzenie stabilnego zmieszanego stanu
- przy przepełnieniu pamięci powstają pośrednie minima lokalne nie odpowiadające żadnemu wzorcowi zapamiętanemu

Metoda rzutowania

Zakładamy, że każdy wzorzec $\mathbf{x}$ podany na wejście generuje natychmiastowo na wyjściu sieci stan ustalony $\mathbf{x}$

$$\mathbf{W} \cdot \mathbf{X} = \mathbf{X} \quad \Rightarrow \quad \mathbf{W} = \mathbf{X} \cdot \mathbf{X}^{+}$$

gdzie $\mathbf{X} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(p)}]$
Dla $\mathbf{x}^{(i)}$ liniowo niezależnych możemy zastąpić psudoinwersję macierzy $N \times p$ inwersją macierzy $p \times p$

$$\mathbf{W} = \mathbf{X} \cdot \mathbf{X}^{+} = \mathbf{X} \left(\mathbf{X}^{T} \cdot \mathbf{X}\right)^{-1} \mathbf{X}^{T}$$

Możliwe też wyznaczenie $\mathbf{W}$ w sposób iteracyjny podczas jednokrotnej prezentacji wzorców uczących $i = 1, \ldots, p$

$$\mathbf{W}^{(i)} = \mathbf{W}^{(i-1)} + \frac{\left(\mathbf{W}^{(i-1)}\mathbf{x}^{(i)} - \mathbf{x}^{(i)}\right)\left(\mathbf{W}^{(i-1)}\mathbf{x}^{(i)} - \mathbf{x}^{(i)}\right)^{T}}{\left(\mathbf{x}^{(i)}\right)^{T}\mathbf{x}^{(i)} - \left(\mathbf{x}^{(i)}\right)^{T}\mathbf{W}^{(i-1)}\mathbf{x}^{(i)}}$$

gdzie $\mathbf{W}^{(0)} = \mathbf{0}$

Metoda rzutowania Δ

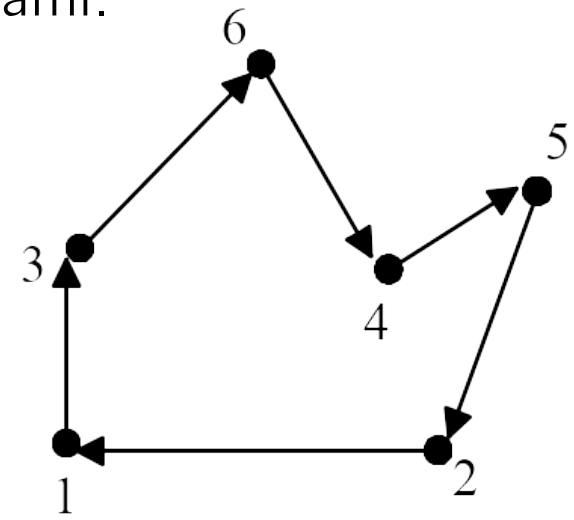
Gradientowa odmiana metody rzutowania

$$\mathbf{W} \leftarrow \mathbf{W} + \frac{\eta}{N}\left[\mathbf{x}^{(i)} - \mathbf{W}\mathbf{x}^{(i)}\right](\mathbf{x}^{(i)})^T$$

- η stała uczenia zazwyczaj z przedziału $[0.7 - 0.9]$
- wymaga wielokrotnej prezentacji wzorców aż do uzyskania zbieżności, np. gdy zmiana wartości wag nie przekracza pewnego progu tolerancji ϵ
- metoda rzutowania i rzutowania Δ zapewniają większą pojemność w stosunku do uczenia reguła Hebba, maksymalna liczba zapamiętanych wzorców $N - 1$

Optymalizacja - problem komiwojażera

Problem komiwojażera: znaleźć najkrótszą droga pomiędzy N miastami.



Miasto	Kolejność 1	2	3	4	5	6
1	1	0	0	0	0	0
2	0	0	0	0	0	1
3	0	1	0	0	0	0
4	0	0	0	1	0	0
5	0	0	0	0	1	0
6	0	0	1	0	0	0

Problem NP-trudny, $\frac{n!}{2n}$ wszystkich możliwości

- sieć zawierająca N^2 neuronów unipolarnych ciągłych
- stan neuronu $y_{i\alpha} = 1$ oznacza, że miasto i zostało odwiedzone w kolejności α
- numer miasta $i, j = 1, 2, \ldots, N$, kolejność odwiedzin $\alpha, \beta = 1, \ldots, N$

Przykład: problem komiwojażera

Ogólna postać funkcji energii

$$E = -\frac{1}{2} \sum_{i \neq k} \sum_{\alpha \neq \beta} w_{i\alpha,k\beta} y_{i\alpha} y_{k\beta}$$

Jak dobrać W?

Dobór wag

Kara za powtórne odwiedzenie miasta

$$E_1 = \frac{A}{2} \sum_i \sum_{\alpha \neq \beta} y_{i\alpha} y_{i\beta}$$

Kara za rozdwojenie komiwojażera (odwiedzenie dwóch miejsc w tym samym czasie)

$$E_2 = \frac{B}{2} \sum_{i \neq k} \sum_{\alpha} y_{i\alpha} y_{k\alpha}$$

Kara za wzbudzenie większej liczby neuronów niż N

$$E_3 = \frac{C}{2} \left(\sum_{i,\alpha} y_{i\alpha} - N \right)^2$$

Czynnik promujący najkrótszą trasę

$$E_4 = \frac{D}{2} \sum_{i \neq k} \sum_{\alpha} d_{ik} y_{i\alpha} \left(y_{k\alpha-1} + y_{k\alpha+1} \right)$$

Dobór wag

Całkowita energia

$$E = E_1 + E_2 + E_3 + E_4$$

stąd wagi sieci

$$w_{i\alpha,k\beta} = -A\left(1-\delta_{\alpha\beta}\right)\delta_{ik} - B\left(1-\delta_{ik}\right)\delta_{\alpha\beta} - C$$

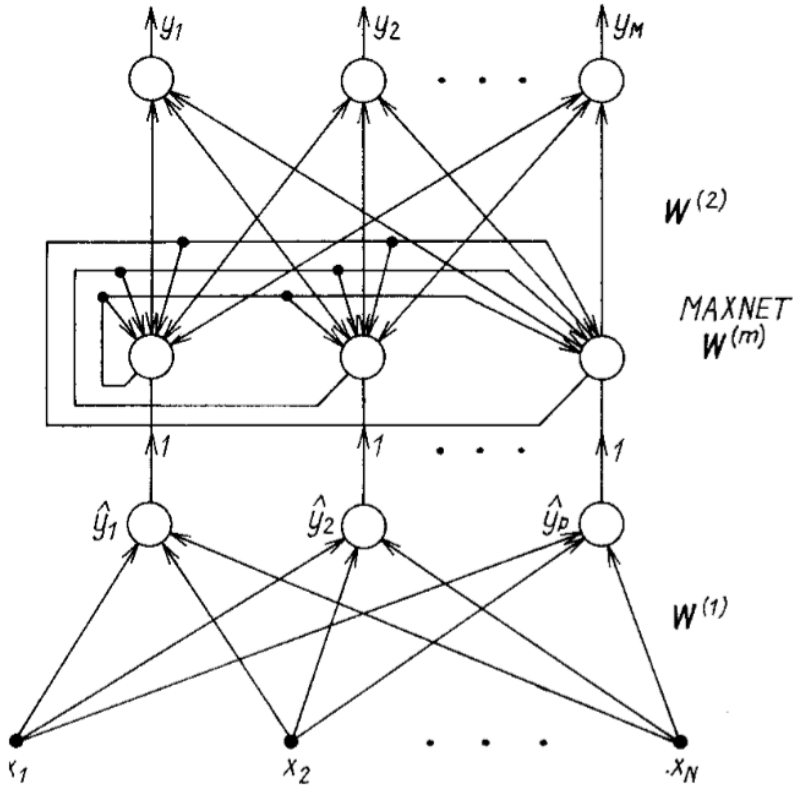
$$-Dd_{ik}\left(1-\delta_{ik}\right)\left(\delta_{\alpha-1,\beta}+\delta_{\alpha+1,\beta}\right)$$

oraz wartości progowe CN

Delta Kroneckera $\delta_{ij} = 1$ dla $i = j$ oraz $\delta_{ij} = 0$ dla $i \neq j$

Wartości A, B, C, D dobierane heurystycznie, np.
$A = B = C = 500$ i $C = 200$

Sieć Hamminga

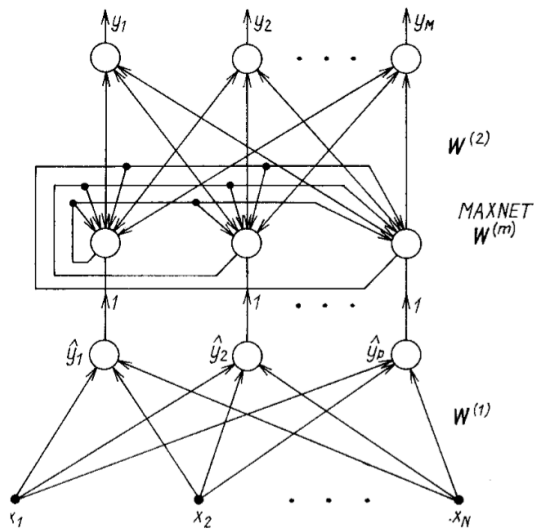


Sieć Hamminga

- 3 warstwowa sieć - rozszerzenie sieci Hopefielda
- realizuje pamięć heteroasocjacyjna:
  wejście $\mathbf{x} \in \{-1, +1\}^N$, wyjścia $\mathbf{y} \in \{-1, +1\}^M$
- wyjściem jest zapamiętany wzorzec o najmniejszej odległości Hamminga względem wektora wejściowego
- warstwa MAXNET (warstwa pamięci) ze sprzężeniami zwrotnymi, połączenia każdy z każdym,
  - rozmiar równy liczbie wzorców do zapamiętania p, pojedynczy neuron jest prototypem zapamiętanego wzorca
  - w drodze iteracyjnego procesu wybierany jest neuron zwycięzca (WTA) reagujący na wzorzec wejściowy

Działanie sieci Hamminga

1. podanie stanu **x** na wejście, zainicjowanie warstwy MAXNET
2. proces iteracyjny w warstwie MAXNET trwa dopóki wszystkie jednostki oprócz jednej osiągną stan 0. Neuron niezerowy reprezentuje klasę wektora
3. wytworzenie odpowiedzi sieci **y** na podstawie sygnału zwycięzcy



Dobór wag

1. wagi warstwy pierwszej odpowiadają sygnałom wejściowym

$$w_{ij}^{(1)} = -\frac{x_j^{(i)}}{2}, \qquad w_{i0}^{(1)} = \frac{N}{2}, \quad i = 1, \ldots, p$$

2. wagi warstwy wyjściowej kodują sygnały wyjściowe

$$w_{ij}^{(3)} = y_j^{(i)}, \quad i = 1, \ldots, p$$

3. warstwa MAXNET
   - połączenia zwrotne do tych samych neuronów mają charakter pobudzający

$$w_{ii}^{(m)} = 1$$

   - połączenia między różnymi neuronami inhibicyjne (hamujące)

$$-\frac{1}{p} < w_{ij}^{(m)} < 0$$

   gdzie p - liczba wzorców

Działanie sieci Hamminga

Pierwsza warstwa wyznacza odległość Hamminaga od podanego wzorca $\mathbf{x}$ do wszystkich zapamiętanych wzorców wejściowych $\mathbf{x}^{(i)}$

$$y_i^{(1)}(\mathbf{x}) = \sum_{j=1}^{N} w_{ij}^{(1)} x_j + w_{i0} = \frac{1}{2}\left(N - \sum_{j=1}^{N} x_j^{(i)} x_j\right)$$

Rekurencyjny proces wyłaniania zwycięzcy warstwy MAXNET

$$y_i(k) = f\left(\sum_{j=1}^{p} w_{ij}^{(m)} y_i(k-1)\right)$$

gdzie funkcja wyjściowa (ReLU)

$$f(y) = \begin{cases} y & \text{dla} \quad y \geq 0 \\ 0 & \text{dla} \quad y < 0 \end{cases}$$

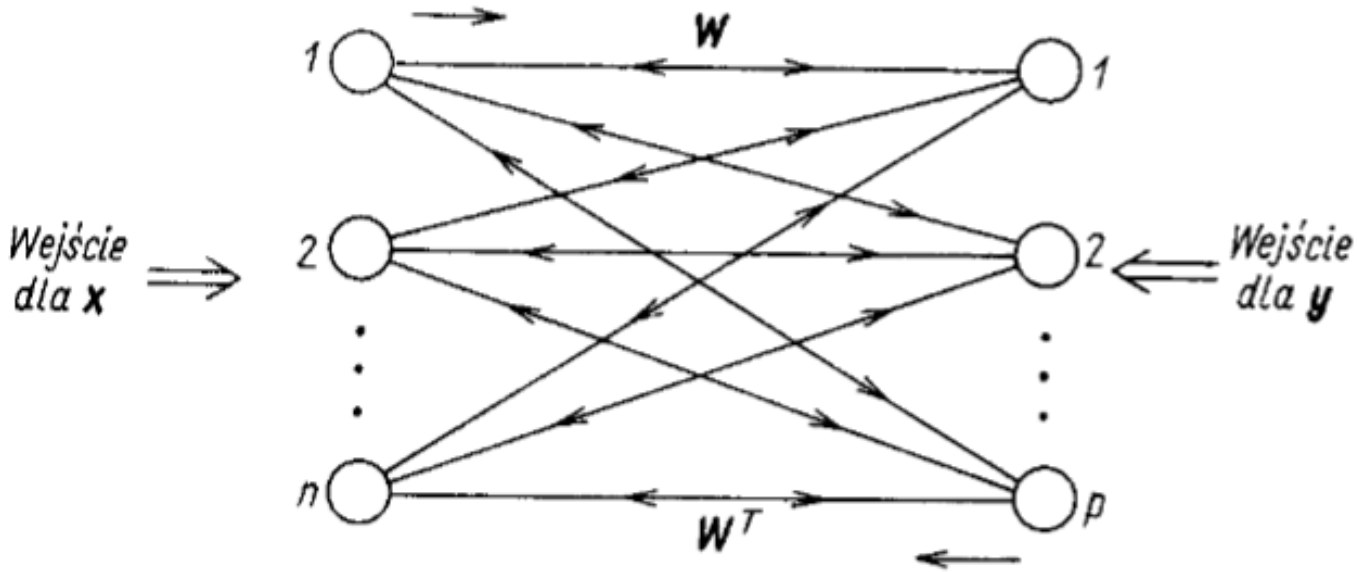
Sygnał zwycięskiego neuronu odtwarza na wyjściu wektor $\mathbf{y}$ najbliższy $\mathbf{x}$ względem odległości Hamminga

Hamming vs Hopfield

- pojemność sieci Hamminga = liczba neuronów MAXNET
- ilość wag $N \times p + p^2 + p \times M$
- dla $N = M$ możliwa realizacja pamięci autoasocjacyjnej ilość wag $2N \times p + p^2$
- dla sieci Hopfielda ilość wag N^2
- dla $p \ll N$ sieć Hamminga potrzebuje o wiele mniej wag do osiągnięcia takiej samej pojemności
- w sieci Hamminga nie występują stany fałszywe

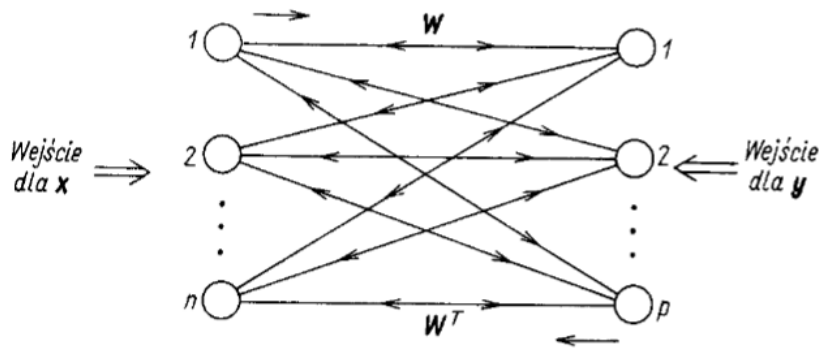
Sieć BAM

Bidirectional Associative Memory (BAM) - uogólnienie sieci Hopfielda na dwuwarstwową sieć rekurencyjną (Kasko)



BAM

- przepływ sygnału w dwóch kierunkach
- praca synchroniczna lub asynchroniczna
- aktywacje $f(x)$ typu skokowego (binarne lub bipolarne)
- $\mathbf{W}$ rzeczywista, niesymetryczna
- koduje pary wektorów $\mathbf{x}_i, \mathbf{y}_i$ - pamięć heteroasocjacyjna



Działanie BAM - zapamiętywanie

Trening: ustalenie wag wartościami macierzy korelacji

$$\mathbf{W} = \sum_{i=1}^{p} \mathbf{x}_i^T \mathbf{y}_i$$

Dla wejść bipolarnych $x_i, y_j \in \{-1, +1\}$ otrzymujemy uczenie hebbowskie - sygnały o tym samym znaku mają wagę wzmacniającą, o przeciwnym - hamującą
Przykład:

$$\begin{array}{ll} A_1 = (101010) & B_1 = (1100) \\ A_2 = (111000) & B_2 = (1010) \end{array} \quad \Rightarrow \quad \begin{array}{ll} \mathrm{x}_1 = (1,-1,1,-1,1,-1) & \mathrm{y}_1 = (1,1,-1,-1) \\ \mathrm{x}_2 = (1,1,1,-1,-1,-1) & \mathrm{y}_2 = (1,-1,1,-1) \end{array}$$

$$\mathrm{W} = \mathbf{x}_1^T \mathbf{y}_1 + \mathbf{x}_2^T \mathbf{y}_2 =$$

$$= \begin{pmatrix} 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 & -2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & -2 \\ -2 & 0 & 0 & 2 \\ 0 & 2 & -2 & 0 \\ -2 & 0 & 0 & 2 \end{pmatrix}$$

Działanie BAM - odtwaranie

Odtwarzanie sygnału

$$f(\mathbf{x}_0\mathbf{W}) = \mathbf{y}_1 \quad \rightarrow \quad f(\mathbf{y}_1\mathbf{W}^T) = \mathbf{x}_1 \quad \rightarrow \quad \ldots$$
$$\ldots \quad \rightarrow \quad f(\mathbf{x}_k\mathbf{W}) = \mathbf{y}_k \quad \rightarrow \quad f(\mathbf{y}_k\mathbf{W}^T) = \mathbf{x}_k$$

generuje dwa stabilne wzorce $\mathbf{x}_k$ i $\mathbf{y}_k$
Funkcja aktywacji progowa

$$f(x) = \begin{cases} 1, & x > 0 \\ 0, & x < 0 \end{cases}$$

Funkcja energii

$$E_k = -\mathbf{x}_k\mathbf{W}\mathbf{y}_k^T$$

Przykład:
$\mathrm{A}_1\mathrm{W} = (4, 2, -2, -4) \rightarrow (1, 1, 0, 0) = \mathrm{B}_1 \qquad E_1 = -6$
$\mathrm{A}_2\mathrm{W} = (4, -2, 2, -4) \rightarrow (1, 0, 1, 0) = \mathrm{B}_2 \qquad E_2 = -6$
gdy zaburzymy 1 bit w A_2
$(0, 1, 1, 0, 0, 0)\mathbf{W} = (2, -2, 2, -2) \rightarrow (1, 0, 1, 0) = B_2$ początkowa energia $E = -4$

Przykład

asynchroniczny BAM
wzorce zapamiętane:
(S,E) (M,V) (G,N)
wejście $10 \times 14 = 140$
wyjście $9 \times 12 = 108$
wzorzec początkowy (S,E)
z 40% szumem (99
odwrócone bity)

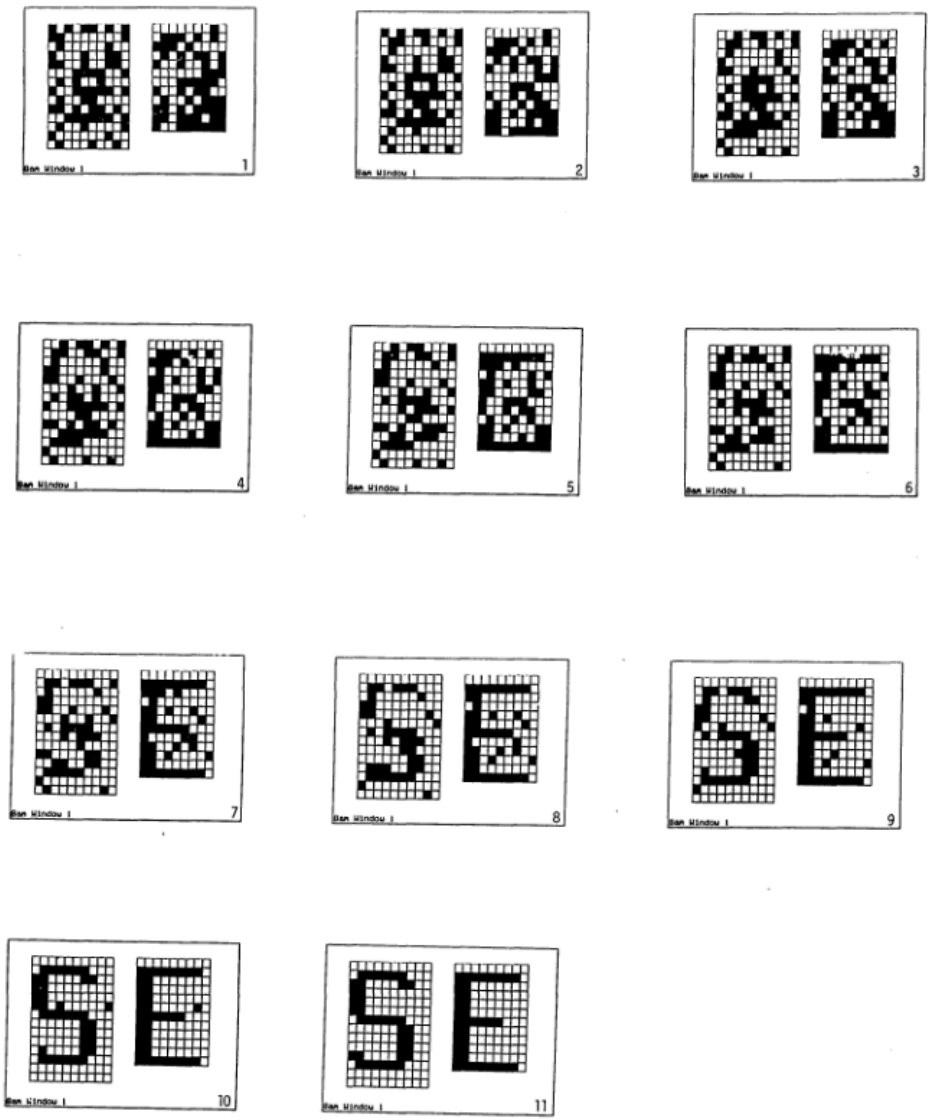


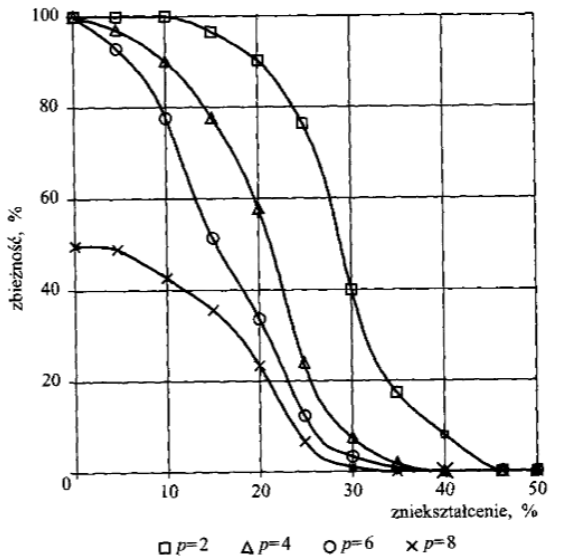
Fig. 1. Asynchronous BAM recall. Approximately six neurons update per snapshot. The associated spatial patterns (S,E), (M,V), and (G,N) are stored. Field F_A contains 140 neurons; F_B,108. Perfect recall of (S,E) is achieved when recall is initiated with a 40% noise-corrupted version of (S,E).

Kosko (1987) Ądaptive bidirectional associative memories"

Pojemność BAM

Pojemność sieci

$$p < \min(n, m)$$



Przykładowe wyniki dla $n = 25, m = 9$ i losowych wzorców ze średnią odległością 11