Sieci dynamiczne

Pamięć asocjacyjna

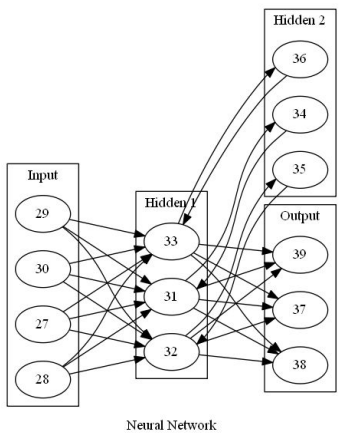
Plan

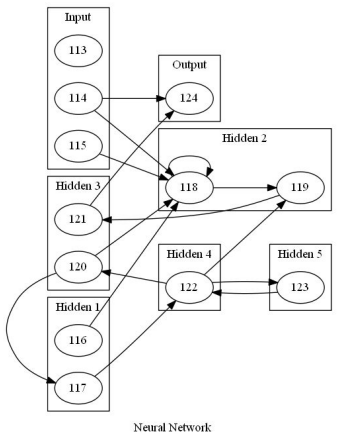
- **Sieci dynamiczne:** sieci ze sprzężeniami zwrotnymi (rekurencyjne), stan sieci w chwili t zależy od stanu w chwili poprzedniej $t-1$
- Modele pamięci asocjacyjnej (autoasocjacyjnej i heteroasocjacyjnej)
- Model Hopfielda, sieć Hamminga, BAM

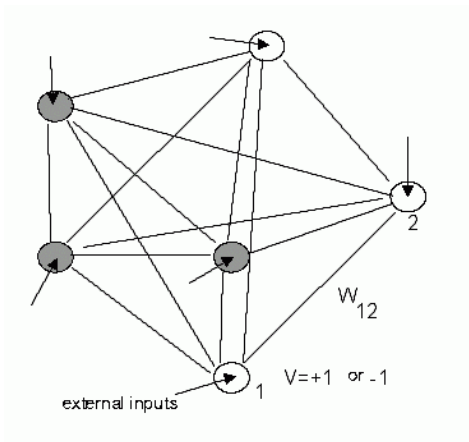
Sieci ze sprzężeniami zwrotnymi

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







Dynamika sieci: zmiana stanu jednego neuronu przenosi się za pomocą sprzężeń zwrotnych na całą sieć, tworząc nowe stany pośrednie

Sieci ze sprzężeniami zwrotnymi

Najprostsze modele sieci z rekurencją:

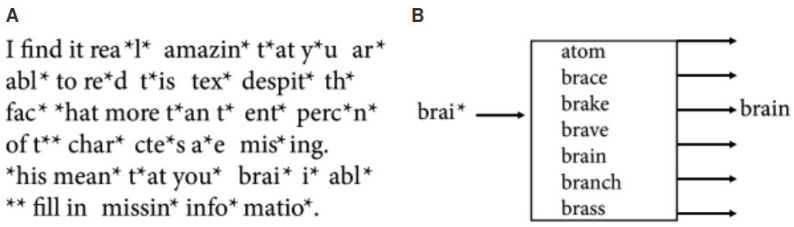
- sieci Hopfielda,
- sieć Hamminga.
- BAM
- Restricted Boltzman Machines (RBM)

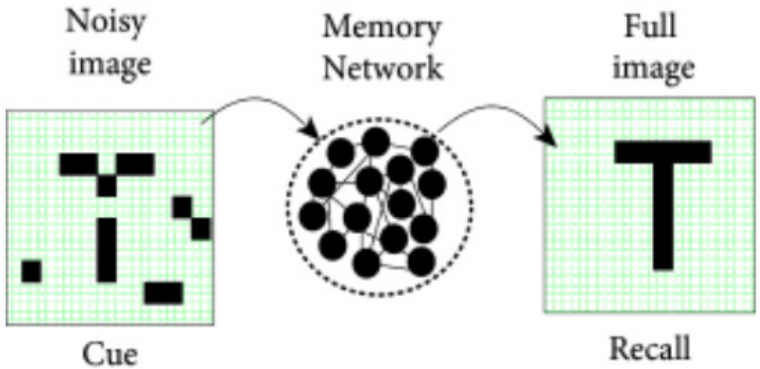
Modele złożone:

- sieć Elmana i inne o uproszczonej strukturze rekurencji
- RTRN - Real Time Recurrent Network, przetwarzająca sygnały w czasie rzeczywistym
- Głębokie sieci rekurencyjne (RNN), LSTM, Long Short Term Memory

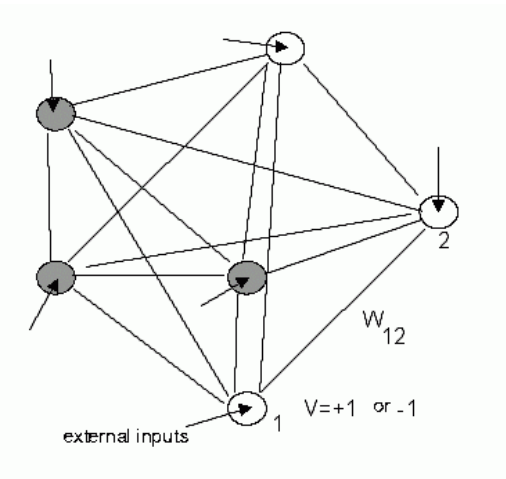
Model pamięci autoasocjacyjnej

Pamięć skojarzeniowa (asocjacyjna) - odtwarza zapamiętane wzorce zbliżone do sygnału wejściowego. Pamięć adresowana kontekstowo (CAM)





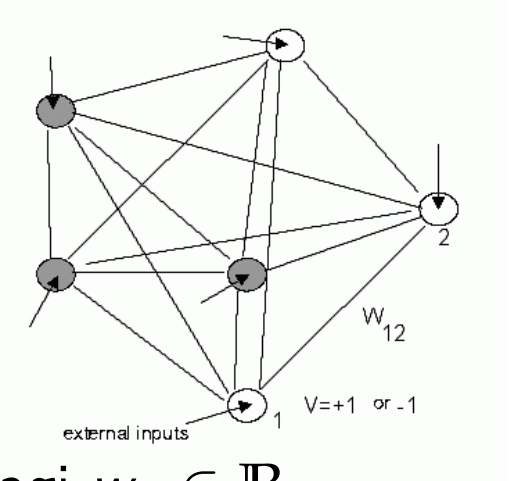
Model pamięci autoasocjacyjnej Hopfielda:



- realizuje pamięć skojarzeniową z binarnymi neuronami progowymi (dyskretne stany neuronów $y_i = \pm 1$)
- wszystkie neurony są ze sobą wzajemnie połączone, brak warstw, każdy neuron jest wejściem i wyjściem
- model stosowany do zrozumienia działania ludzkiej pamięci

Model Hopfielda (1982, 1984)

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



- dyskretne stany neuronów $y \in \{-1, +1\}$, wagi $w_{ij} \in \mathbb{R}$
- zmiany stanów następują w dyskretnych jednostkach czasu
- macierz wag połączeń jest symetryczna, $w_{ij} = w_{ji}$
- brak połączeń zwrotnych do tego samego neuronu $w_{ii} = 0$
- dwa tryby pracy:
  - uczenie - zapamiętywanie wzorców
  - odtwarzanie wzorców przy ustalonych wagach

Symetria wag upraszcza analizę sieci i pozwala wprowadzić funkcję energii, jednak jest nierealistyczna z biologicznego punktu widzenia. Bliżej bioligii są realizacje sieci o ciągłych funkcjach aktywacji, w których zmiany aktywacji następują w sposób ciągły.

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

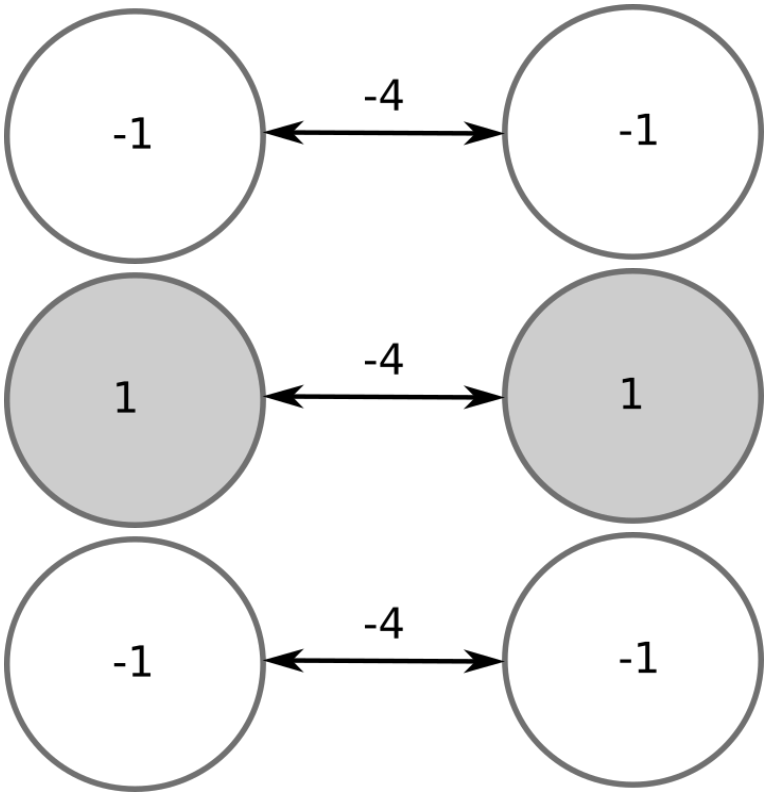
$$y_i(t+1) = \text{sgn}\left(\sum_j w_{ij} y_j(t)\right) \qquad i = 1, \ldots, n$$

- **proces asynchroniczny** - w chwili t aktualizowane jest wyjście pojedynczego (losowo wybranego) neuronu, jego wyjście zależy od stanu pozostałych neuronów z poprzedniego kroku
- **stan stacjonarny** - stan, w którym proces odtwarzania zatrzymuje się, brak zmian wyjść przy pełnym cyklu

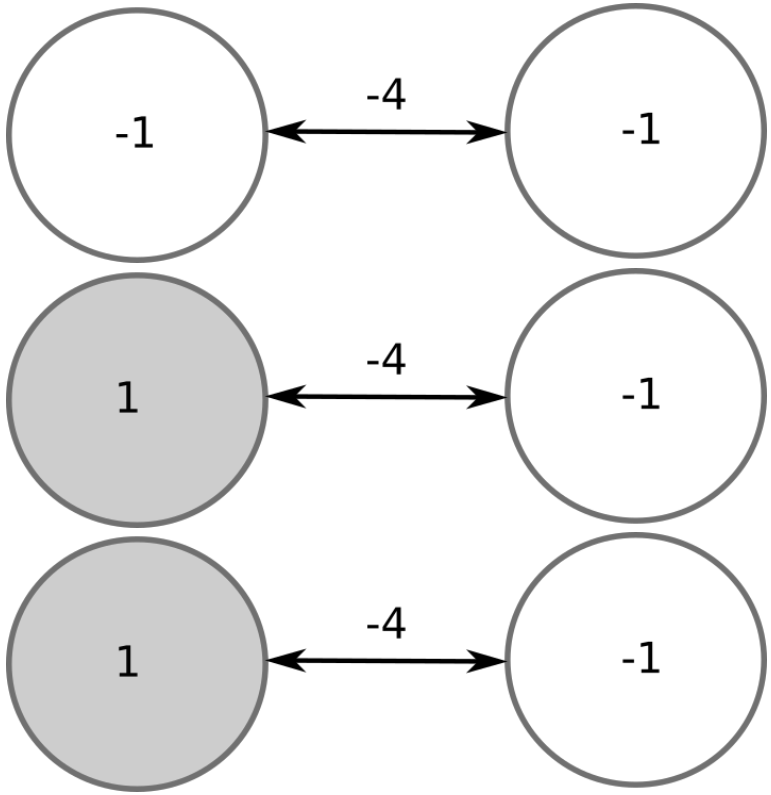
$$\mathbf{y}(t+1) = \mathbf{y}(t)$$

- asynchroniczność zapobiega występowaniu cykli i sieć zbiega do atraktora punktowego
- autoasocjacja - wyjście odtwarza pewien wzorzec wejściowy $\mathbf{x}$

Zmiany asynchroniczne i synchroniczne



Zmiana synchroniczna,
oba neurony zmieniają stan,
nieskończone oscylacje



Zmiana asynchroniczna,
aktualizacja pojedynczego
losowego neuronu,
sieć zbiega do stabilnego wzorca

Źródło:
https://julien-vitay.net/lecturenotes-neurocomputing/4-neurocomputing/2-Hopfield.html

Minimalizacja energii

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

$$E = -\sum_{i,j:i<j} w_{ij} y_i y_j - \sum_i b_i y_i$$

gdzie b_i wyraz wolny ($b_i = w_{0i}$ można włączyć do wektora wag)

Zmiana energii w czasie odtwarzania jest ujemna (energia nie rośnie)

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

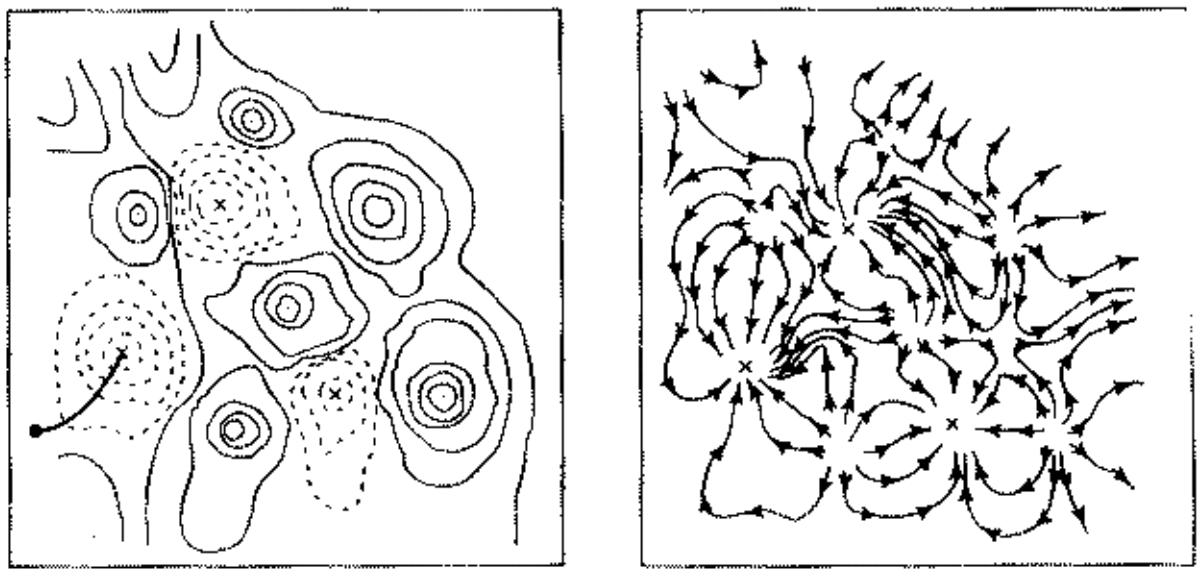
- jeżeli $I_i \geq 0$ to y_i osiąga stan $+1$ i $\Delta y_i \geq 0$
- jeśli $I_i < 0$ to y_i osiąga stan -1 i $\Delta y_i \leq 0$

Każda zmiana stanu neuronu może tylko obniżyć energię sieci

Atraktory

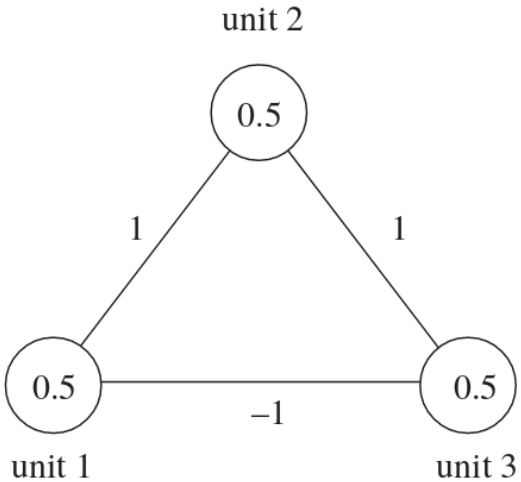
Dynamika: ruch po hiperpowierzchni energii, zależnej od potencjałów neuronów, aż do osiągnięcia lokalnego minimum energii

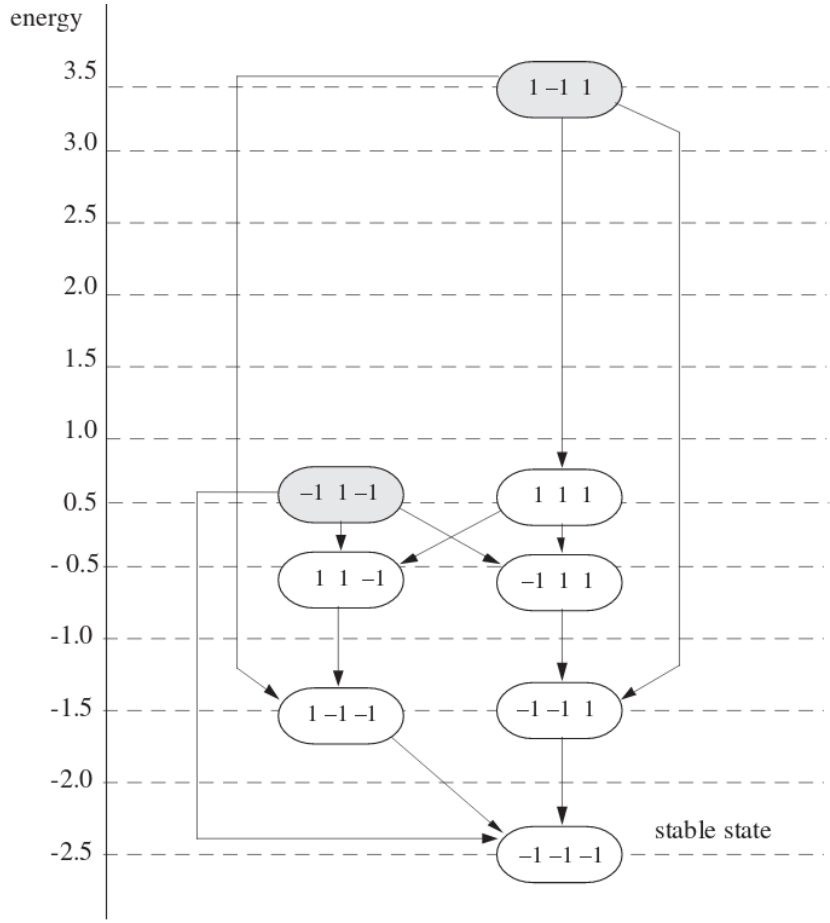
Portret fazowy układu dynamicznego sieci rekurencyjnej o połączeniach symetrycznych: linie ekwipotencjalne i kierunki zmian potencjału w trakcie uczenia



Jeśli y_i dyskretne $\{-1, +1\}$ to ruch odbywa się po wierzchotkach hiperkostki

Przykład: 3 neurony

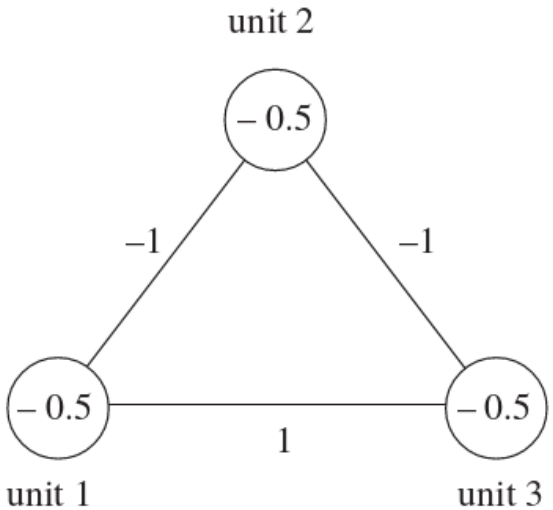


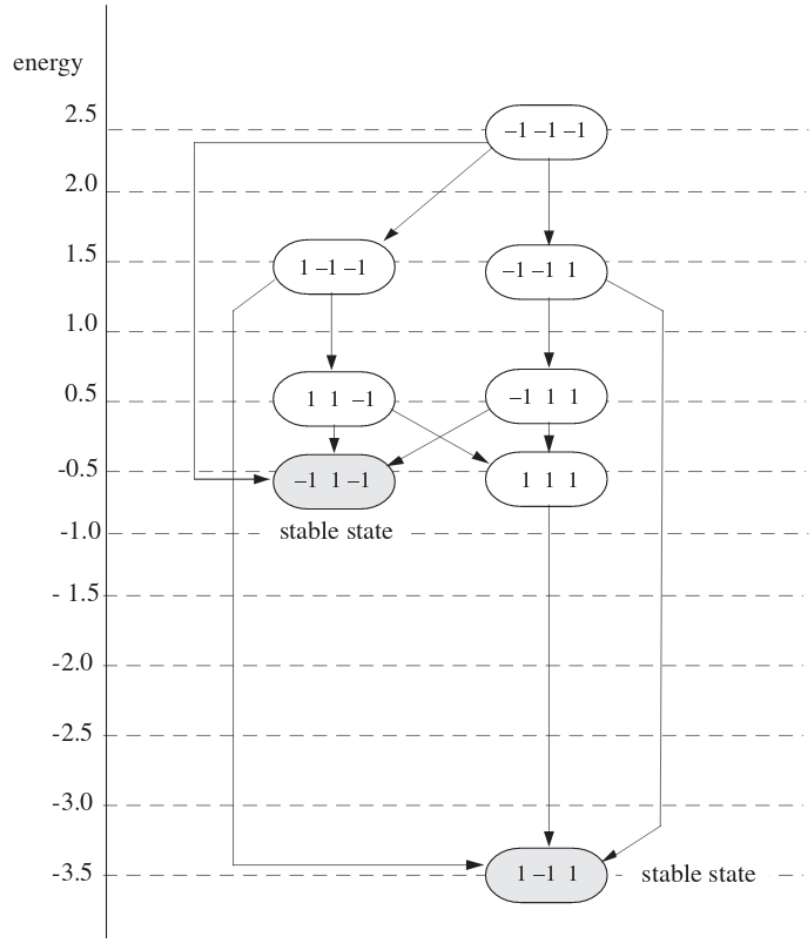


$$E = -y_1y_2 - y_2y_3 + y_1y_3 - 0.5y_1 - 0.5y_2 - 0.5y_3$$

Rys: Rojas, Neuroal Netowrks

Przykład: 3 neurony





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

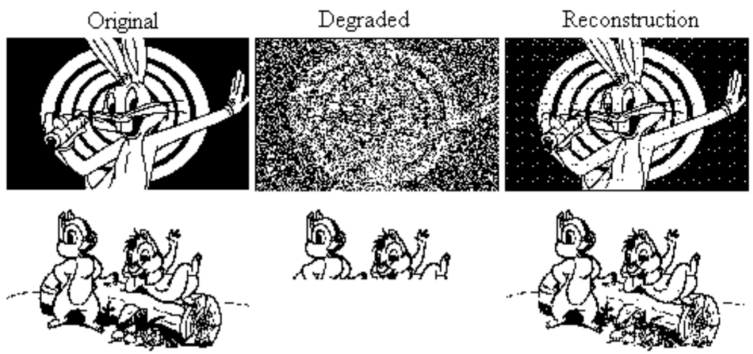
Rys: Rojas, Neuroal Netowrks

Zastosowanie praktyczne

Pamięć asocjacyjna

- **odtwarzanie wzorców**, przy prezentacji wektora wejściowego $\mathbf{x}_i$ odpowiedzią sieci będzie jeden z zapamiętanych wzorców $\mathbf{x}_1, \ldots, \mathbf{x}_p$
- **korekcja**, odtwarzany jest wzorzec najbardziej „podobny” do sygnału wejściowego.
- **uzupełnianie wzorca**: prezentowany jest fragment wzorca (ciąg pierwszych bitów) a siec odtwarza pozostałe (stany początkowych bitów się nie zmieniają)

Odtwarzanie i rekonstrukcja obrazów



Źródło:
https://www.cs.cmu.edu/~bhiksha/courses/deeplearning/Fall.2015/slides/lec14.hopfield.pdf

Zastosowanie praktyczne

Rozwiązywanie problemów optymalizacyjnych - odpowiednia konstrukcja sieci i funkcji energii pozwala rozwiązywać problemy optymalizacyjne, także NP-trudne

- problem komiwojażera,
- gospodarowanie zasobami,
- optymalizacja połączeń w centralach telefonicznych,
- rozmieszczanie układów scalonych,
- tworzenie rozkładów jazdy,
- rozwiązywanie problemów programowania liniowego i nieliniowego

W fizyce: modelowanie szkła spinowego

W neuronaukach: modele hipokampa, pamięć epizodyczna

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** - odtworzony wzorzec uzyskiwany jest na podstawie odległości (podobieństwa) od zapamiętanego wzorca
- **pamięć rozproszona** - wszystkie wagi kodują wzorce, brak wyraźnie określonego miejsca przechowywania wzorca
- **odporność na uszkodzenia** - lokalne uszkodzenie wzorca lub pamięci nie blokuje możliwości odczytu

Odległość Hamminga

Pamięć asocjacyjna odtwarza zapamiętane wzorce najbliższe względem wektora wejściowego (w zasięgu basenu atrakcji).
Dla stanów binarnych miarą odległości jest odległość Hamminga

Odległość Hamminga - liczba różniących się bitów

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

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

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

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

Uczenie pamięci asocjacyjnej

Dla pojedynczego wzorca $\mathbf{x}$ sieć powinna odtwarzać sygnał wejściowy

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

wystarczy zażądać by:

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

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

Uczenie pamięci asocjacyjnej regułą Hebba

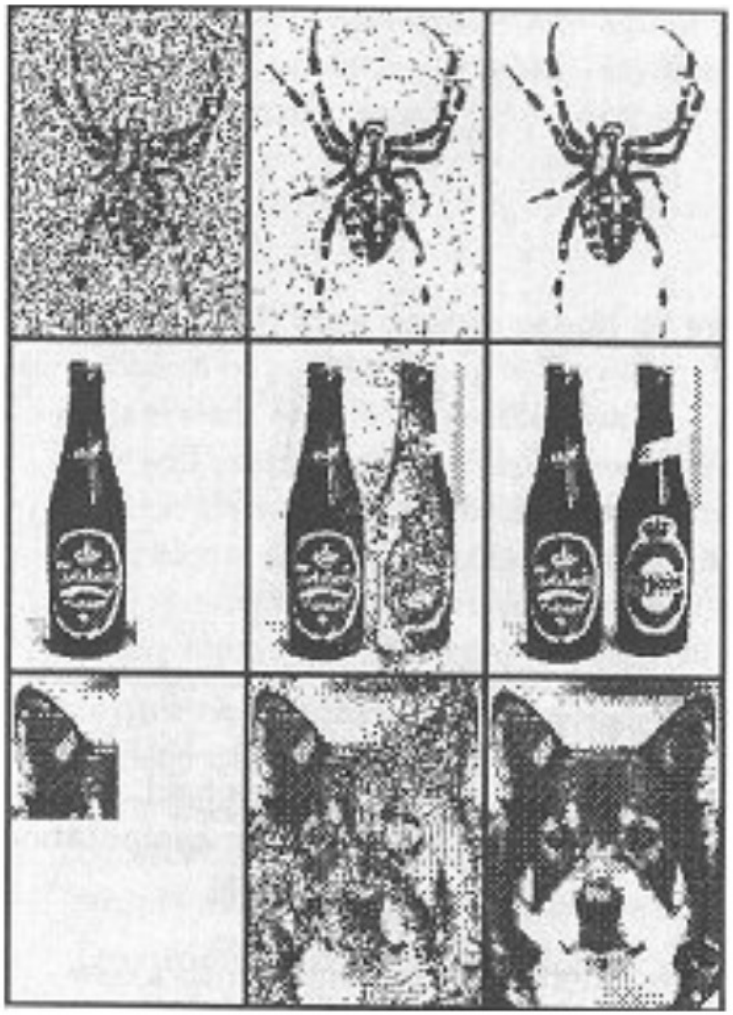
Dla wielu wzorców $\mathbf{x}^{(i)}, i = 1, \ldots, p$ korzystamy z reguły Hebba uśredniając:

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

W praktyce stosuje się często metodę iteracyjną, gdzie adaptacja przy prezentacji k-tego wektora $\mathbf{x}^{(k)}$ przybiera postać

$$\begin{aligned} \Delta w_{ij} &= x_i^{(k)} x_j^{(k)} \quad \text{dla} \quad i, j = 1, \ldots, n \\ \Delta b_i &= x_i^{(k)} \quad \text{dla} \quad i = 1, \ldots, n \end{aligned}$$

Przykłady działania



odtwarzanie niekompletnych lub zaszumionych danych

Stabilność pamięci

Stabilność działania wymaga

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

co można zapisać

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

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 p_{max} określa maksymalną liczbę odtwarzanych wzorców przy określonym poziomie błędu
- 2^N możliwych stanów sieci binarnej złożonej z N neuronów
- przy zapamiętywaniu regułą Hebba liczba poprawnie pamiętanych losowych wzorców dla sieci z n neuronami przy prawdopodobieństwie błędu 0.37% wynosi

$$p_{max} = 0.138 \cdot n$$

- na ogół (przy założeniu $n/2p > \log n$) pojemność spełnia

$$p_{max} = \frac{n}{2 \log n}$$

- mała efektywność, potrzeba n^2 wag o wartościach rzeczywistych

Stany fałszywe

W sieci mogą pojawiać się stany fałszywe lub przekłamania pamięci

- funkcja energii jest symetryczna względem polaryzacji (moga występować 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 zapamiętanemu wzorcowi (fałszywa pamięć)
- zbyt wiele wzorców $\Rightarrow$ chaos, zapominanie, zwiększając liczbę wzorców rośnie prawdopodobieństwo przekłamań

Uczenie pamięci metodą 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 $i = 1, \ldots, p$ bez konieczności wyznaczania macierzy odwrotnej

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

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

Metoda rzutowania Δ

Gradientowa odmiana metody rzutowania

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

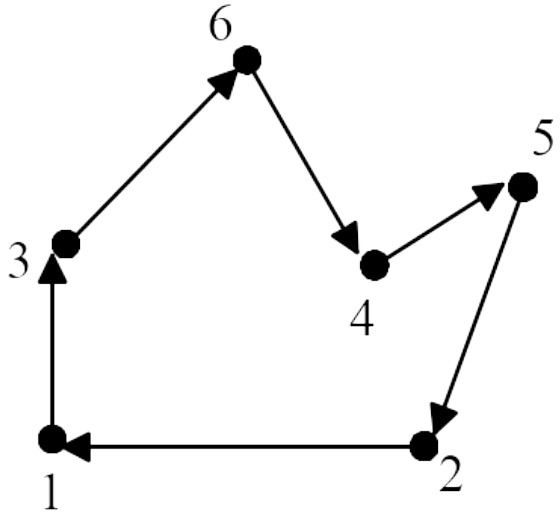
- η stała uczenia zazwyczaj z przedziału $[0.7 - 0.9]$
- wymaga wielokrotnej prezentacji wzorców aż do uzyskania zbieżności, np. gdy zmiana wartości wag nie przekracza pewnego progu tolerancji ϵ
- metoda rzutowania i rzutowania Δ zapewniają większą pojemność w stosunku do uczenia reguła Hebba, maksymalna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

Zastosowanie w problemach optymaliacyjnych

- jeżeli optymalizowana funkcja kosztu (z warunkami) może zostać zapisana w postaci funkcji energii Hopfielda to istnieje sieć Hopfielda, której stany stabilne odpowiadają rozwiązaniu problemu optymalizacji
- dynamika odtwarzania minimalizuje energię powodując zmniejszenie wartości funkcji kosztu przy zachowaniu warunków optymalizacji
- warunki optymalizacji są „zaszyte" w wagach sieci
- problemy NP-trudne - np. problem komiwojażera

Optymalizacja - problem komiwojażera

Problem komiwojażera: znaleźć najkrótszą droga pomiędzy N miastami.



		Kolejność 1 2 3 4 5 6
M	1	1 0 0 0 0 0
i	2	0 0 0 0 0 1
a	3	0 1 0 0 0 0
s	4	0 0 0 1 0 0
t	5	0 0 0 0 1 0
o	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

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

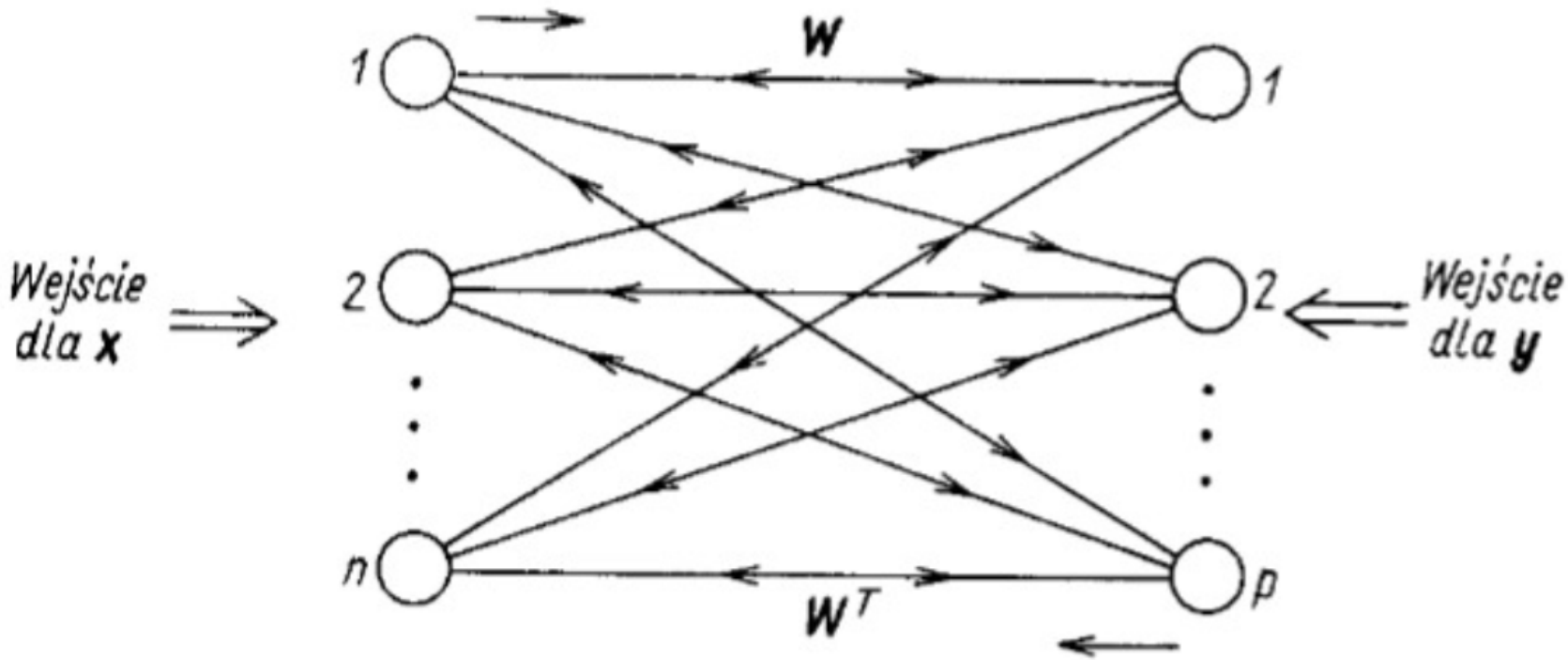
$$-Dd_{ik}\left(1-\delta_{ik}\right)\left(\delta_{\alpha-1,\beta}+\delta_{\alpha+1,\beta}\right)$$

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 $C = 200$

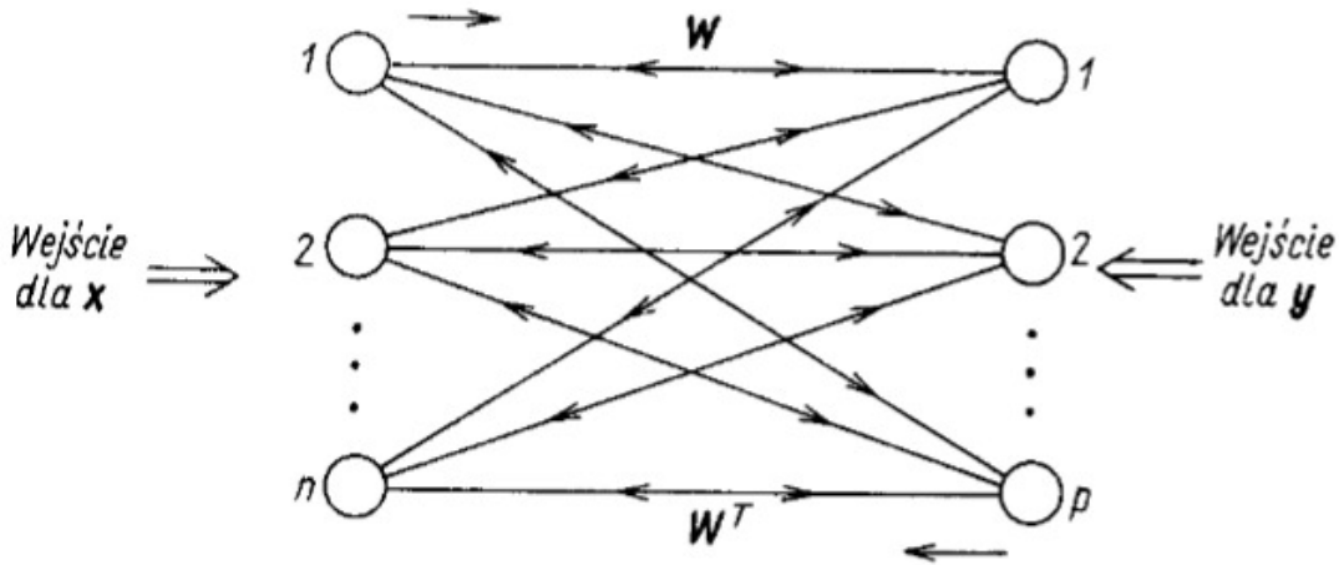
Sieć BAM

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



BAM

- przepływ sygnału odbywa się w dwóch kierunkach
- praca synchroniczna lub asynchroniczna
- progowa funkcja aktywacji $f(x)$ (binarna lub bipolarna)
- $\mathbf{W}$ o wartościach rzeczywistych, niesymetryczna
- zapamiętuje pary wzorców $\mathbf{x}_i, \mathbf{y}_i$
- realizuje **pamięć heteroasocjacyjną**
- pojemność $p < \min(n, m)$, dla n neuronów w warstwie wejściowej i m neuronów w warstwie wyjściowej



Działanie BAM

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

Odtwarzanie sygnału

$$\begin{aligned} f(\mathbf{x}_0\mathbf{W}) = \mathbf{y}_1 \quad &\rightarrow \quad f(\mathbf{y}_1\mathbf{W}^T) = \mathbf{x}_1 \quad \rightarrow \quad \dots \\ \dots \quad &\rightarrow \quad f(\mathbf{x}_k\mathbf{W}) = \mathbf{y}_k \quad \rightarrow \quad f(\mathbf{y}_k\mathbf{W}^T) = \mathbf{x}_k \end{aligned}$$

stan stacjonarny generuje dwa stabilne wzorce $\mathbf{x}_k$ i $\mathbf{y}_k$

Funkcja energii

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

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)

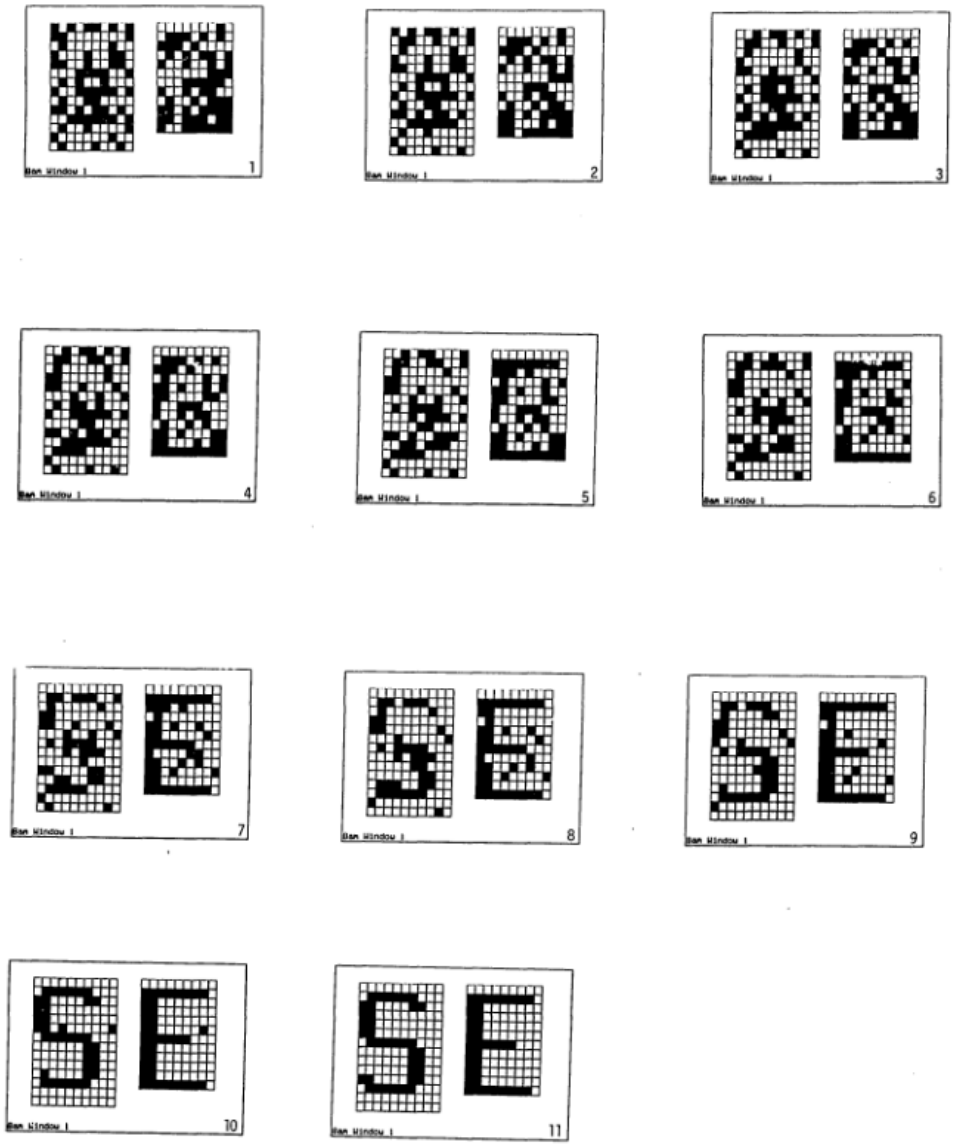


Fig. 1. Asynchronous BAM recall. Approximately six neurons update per snapshot. The associated spatial patterns (S,E), (M,V), and (G,N) are stored. Field F_A contains 140 neurons; F_B,108. Perfect recall of (S,E) is achieved when recall is initiated with a 40% noise-corrupted version of (S,E).

Kosko (1987) Ądaptive bidirectional associative memories"