Sieci Hopfielda

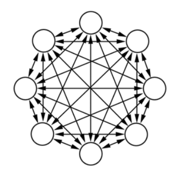
niesłusznie zapomniane
narzędzie neuroinformatyki

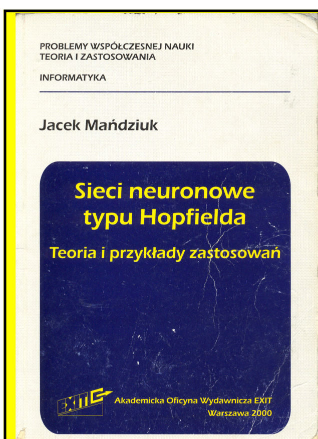
W większości komercyjnych programów przeznaczonych do modelowania i stosowania sieci neuronowych w menu rodzaju dostępnych sieci na próżno by szukać Sieci Hopfielda

Tymczasem **John Hopfield** był tym uczonym, który w „ciemnym okresie" badań sieci neuronowych (po opublikowaniu „demaskatorskiej" książki ***Perceptrons*** Marvina Minsky'ego i Saymoura Paperta) jako pierwszy wystąpił publicznie z publikacjami i referatami dowodzącymi, że pogardzane sieci są wartościowym naukowo i użytecznym narzędziem .

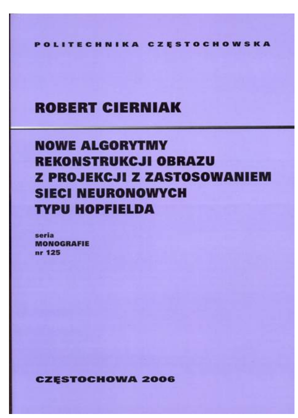
W 1982 roku Hopfield, będący już wtedy sławnym fizykiem (m. in. był już posiadaczem Medalu Diraca, przyznawanego zwykle Noblistom) wygłosił referat na posiedzeniu *National Academy of Sciences (Washington)* referat, w którym dowiódł, że sieci neuronowe są bardzo ważnym obiektem badań i bardzo silnym narzędziem obliczeniowym.

John Hopfield
i jego sieć



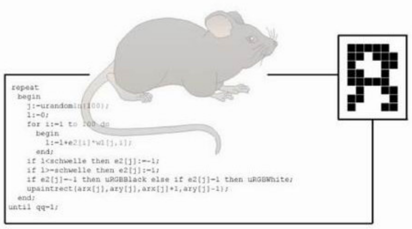


„Klasyczny" podręcznik do tej części wiedzy o sieciach neuronowych

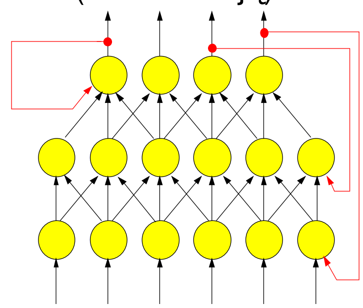


Jedno z nowszych opracowań na temat wykorzystania sieci Hopfielda

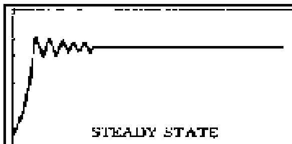
Istotna nowość wnoszona przez sieci Hopfielda polega na tym, że są w nich sprzężenia zwrotne, których inne sieci nie uwzględniają.



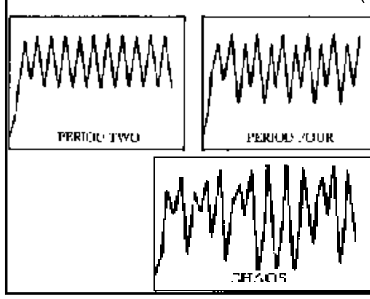
Przykładowa struktura sieci ze sprzężeniami zwrotnymi (z rekurencją)



Na skutek istnienia sprzężeń zwrotnych w sieciach rekurencyjnych pojawiają się dynamiczne procesy przejściowe, nieznane w innych rodzajach sieci.



Te dynamiczne procesy przejściowe mogą się po pewnym czasie stabilizować, dostarczając potrzebne rozwiązanie (odpowiedź sieci)

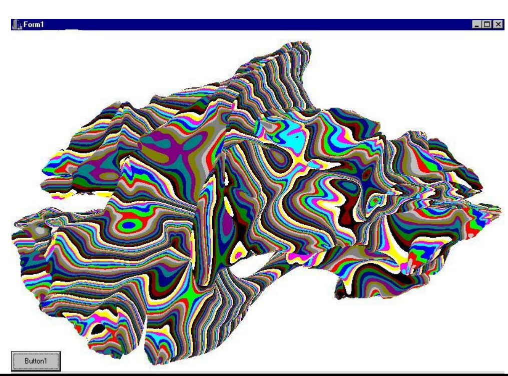


Czasem jednak procesy dynamiczne trwają w nieskończoność jako przebiegi periodyczne...

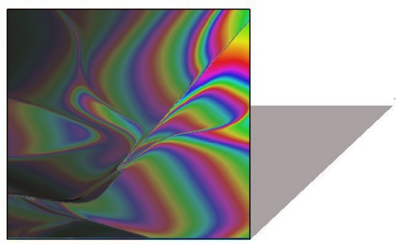
... albo co gorsza chaotyczne

Procesy te mogą więc prowadzić sieć do jednego z ustalonych punktów równowagi (wyznaczonych w przestrzeni sygnałów wyjściowych przez tzw. **atraktory**), ale mogą też prowadzić do procesów, w których sygnały wyjściowe sieci nigdy się nie zatrzymują, tylko zmieniają się w nieskończoność: **periodycznie** albo **chaotycznie**.

Mapa chaotycznej aktywności sieci neuronowej



Chaotyczne zachowania rekurencyjnych sieci neuronowych może prowadzić do generowania całkiem niebanalnych rozkładów pobudzeń na wyjściach neuronów budujących sieć

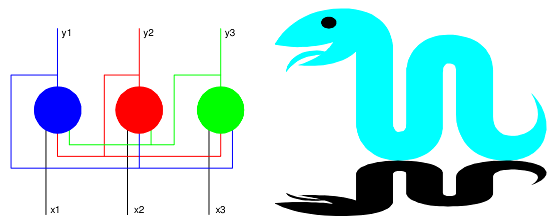


Pytanie:

czy sieć Hopfielda ma sprzężenia zwrotne?

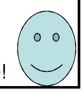
przypomina inne pytanie:

czy wąż ma ogon?

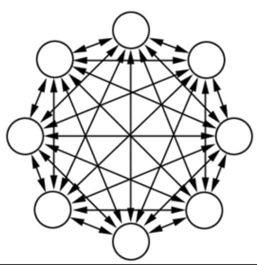


Odpowiedź w obydwu przypadkach jest ta sama:

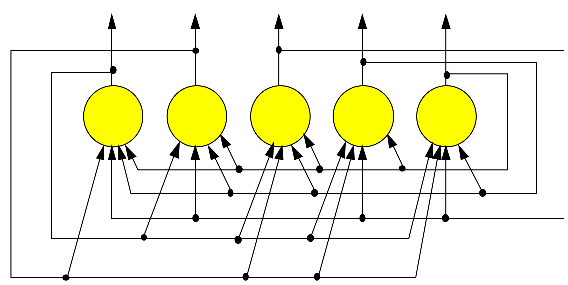
Wyłącznie!



Sieci Hopfielda są czasem rysowane w taki sposób, żeby można było wygodnie zaznaczyć wszystkie występujące w nich połączenia



Częściej jednak sieć Hopfielda rysowana jest także jako sieć warstwowa, chociaż składa się z samych sprzężeń zwrotnych!



Stan sieci Hopfielda określony jest w każdej chwili przez sygnały na wyjściach wszystkich neuronów.

Załóżmy, że sieć składa się z n neuronów, a sygnał na wyjściu neuronu o numerze i w chwili t oznaczony jest jako $y_i(t)$. Wówczas

$$y_i(t+1) = \varphi(\sum_{j=1}^{n} w_j^{(i)} y_j(t)\ ) + w_0^{(i)} \ + x_i(t)$$

gdzie $w_j^{(i)}$ oznacza wagę synaptyczną na *j-tym* wejściu neuronu o numerze i,

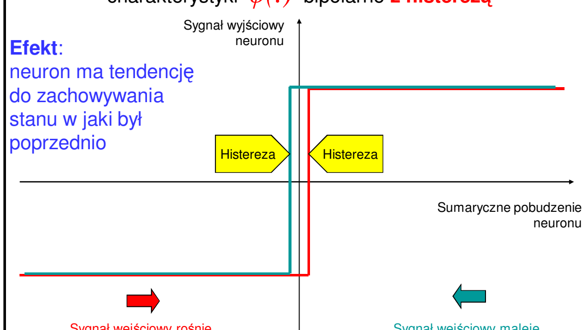
$w_0^{(i)}$ oznacza wyraz wolny (składową stałą, BIAS) neuronu o numerze i.

Warto zwrócić uwagę na następujące elementy podanego wzoru:

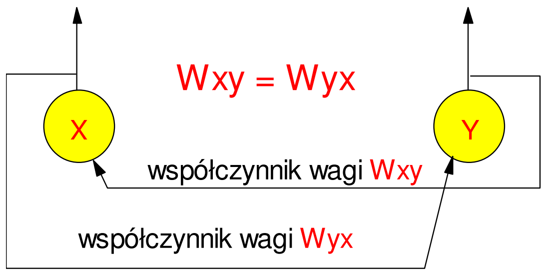
- na *j-tym* wejściu **każdego** neuronu podawany jest przez sprzężenie zwrotne sygnał wyjściowy z wyjścia *j-tego* neronu
- formuła pokazuje, że każdy neuron wnosi opóźnienie do przetwarzanych sygnałów, bo na wyjściu sygnał $y_i(t+1)$ pojawia się z opóźnieniem w stosunku do sygnałów na wszystkich wejściach $y_j(t)$, co pozwala na obserwację dynamiki procesów
- formuła może być poszerzona o sygnał wejściowy $x_i(t)$, który jest niezbędny do wprowadzenia danych startowych i dla pobudzenia sieci do działania (zwykle $x_i(t) \neq 0$ jedynie dla $t = 1$ oraz $x_i(t) = 0$ dla wszystkich $t > 1$)
- charakterystyka $\varphi(.)$ neuronu ma pewne unikatowe właściwości, pokazane dalej.

Żeby procesy przejściowe po osiągnięciu atraktorów już ich nie opuszczały, neurony w sieci Hopfielda mają z reguły charakterystyki $\varphi(.)$ bipolarne **z histerezą**

Efekt: neuron ma tendencję do zachowywania stanu w jaki był poprzednio



Współczynniki wag w sieci Hopfielda powinny spełniać wymaganie symetrii gwarantujące stabilne działanie sieci



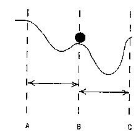
Działanie sieci Hopfielda sterowane jest przez dążenie do minimalizacji pewnej funkcji, którą zwyczajowo utożsamia się z „energią” sieci.

$$E(t) = (-1/2)\sum_{i=1}^{n}\sum_{j=1}^{n} w_j^{(i)} y_j(t) y_i(t) - \sum_{i=1}^{n} x_i(t) y_i(t) + \sum_{j=1}^{n} w_0^{(j)} y_j(t)$$

Można wykazać, że wszystkie procesy w sieci Hopfielda mogą przebiegać wyłącznie w takim kierunku, że „energia” $E(t)$ może tylko maleć lub pozostawać bez zmian – nigdy nie może rosnąć.

Oznacza to, że znając funkcję energii możemy przewidzieć trajektorie zmian sygnałów w całej sieci.

Dla uzasadnienie tego przywołam odpowiedni fragment mojej książki:

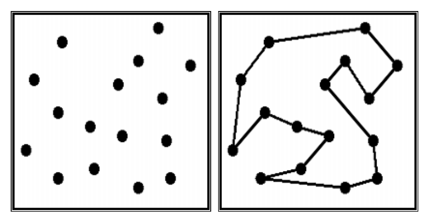


Na podstawie wyżej podanej definicji funkcji E można obliczyć zmianę δE zachodzącą na skutek zmiany stanu sieci wyrażającej się zmianą sygnału wyjściowego i-tego neuronu:

$$\delta E^{(j)} = -\Big[\sum_{m \neq i} w_m^{(i)} y_m^{(j)} + x_i^{(j)} - w_0^{(i)}\Big]\, \delta y_i^{(j)}$$

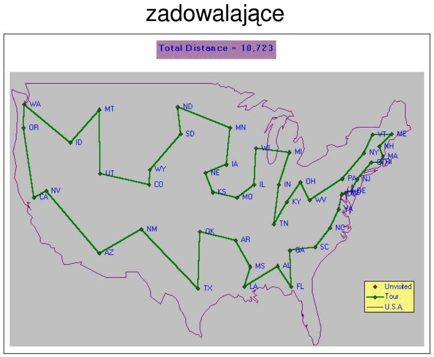
Fakt, że sieć Hopfielda w naturalny sposób dąży do minimalizacji funkcji „energii” umożliwia jej wykorzystanie do rozwiązywania zadań optymalizacji

Najbardziej znanym przykładem zastosowania tego typu było rozwiązanie przy pomocy sieci Hopfielda słynnego „Problemu Komiwojażera” (m.in. przez Hopfielda i Tanka)



Postawienie problemu — Rozwiązanie problemu

Rozwiązania dostarczane przez sieć Hopfielda dla Problemu Komiwojażera są często sub-optymalne, ale z praktycznego punktu widzenia są zadowalające



Kluczem do sukcesu przy stosowaniu sieci neuronowej w problemie TSP jest odnalezienie odpowiedniej reprezentacji danych. W opisanym przez Tanka rozwiązaniu problemu każde miasto reprezentowane jest za pomocą wiersza zawierającego n neuronów. W takim wierszu **dokładnie jeden** neuron powinien przyjmować wartość „1”, a **wszystkie pozostałe** mają sygnały wyjściowe odpowiadające wartości „0”. Pozycja (od 1 do n), na której występuje neuron sygnalizujący „1” odpowiada kolejności, w jakiej to właśnie miasto ma być odwiedzone przez wędrownego sprzedawcę. Tak więc jedynka na pierwszej pozycji oznacza miasto, które komiwojażer powinien odwiedzić jako pierwsze, jedynka na drugiej pozycji sygnalizuje drugie w kolejności odwiedzane miasto itd. Oczywiście z opisu tego wynika, że liczba wierszy musi odpowiadać liczbie rozważanych miast (czyli musi wynosić n), zatem łączna liczba potrzebnych neuronów wynosi n^2.

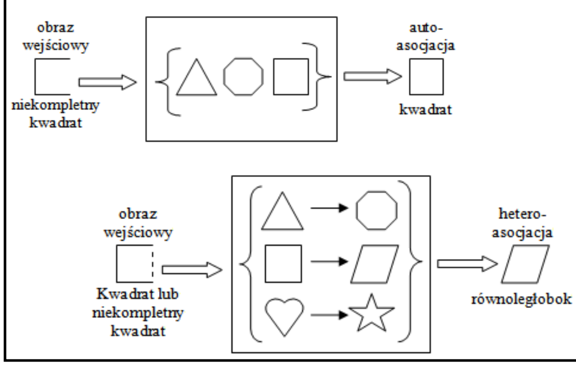
Opisując funkcję „energii” minimalizowanej przez rozważaną sieć trzeba brać pod uwagę cztery jej składniki:

$$E_1 = A/2 \sum_x \sum_i \sum_{i \neq j} y_{xi}\, y_{xj}$$
$$E_2 = B/2 \sum_i \sum_x \sum_{z \neq x} y_{xi}\, y_{zj}$$
$$E_3 = C/2 \left[\Big(\sum_x \sum_i y_{xi} \Big) - n \right]^2$$
$$E_4 = D/2 \sum_x \sum_{z \neq x} \sum_i d_{xz}\, y_{xi}\, (y_{z,i+1} + y_{z,i-1})$$

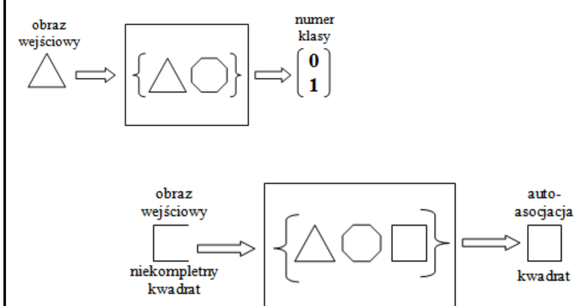
Nie rozwijam tu tego tematu (po szczegóły można zajrzeć do mojej książki **dostępnej w całości w Internecie**) ale chcę wskazać, że mając trudny problem optymalizacyjny, który musi być **szybko** rozwiązywany (np. przy sterowaniu cyfrowym w czasie rzeczywistym) warto sięgnąć do możliwości, jakie stwarza sieć Hopfielda.

Inne ciekawe możliwości zastosowań wiążą się z użyciem sieci Hopfielda jako pamięci skojarzeniowej

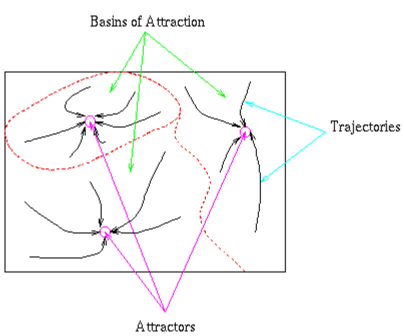
Autoasocjacja i heteroasocjacja



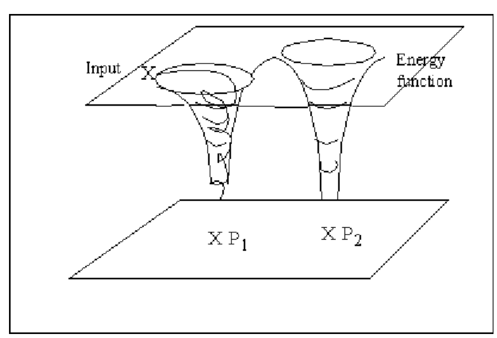
Asocjacja nie jest klasyfikacją!



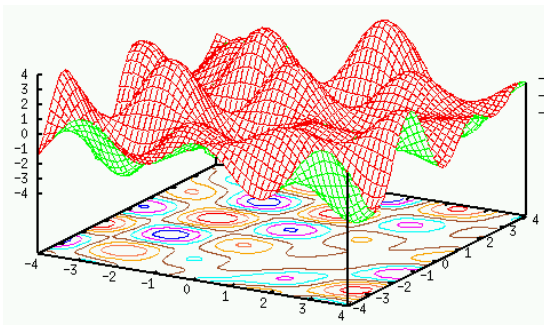
Uproszczony obraz atraktorów oraz trajektorii w sieci Hopfielda



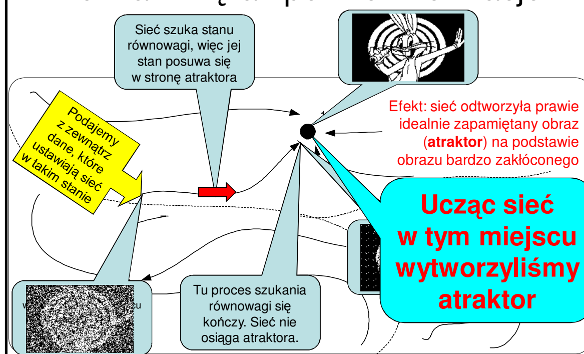
W miejscu występowania atraktora proces uczenia wytwarza „studnię" w powierzchni reprezentującej funkcję „energii" sieci



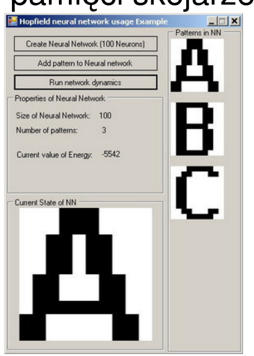
W praktyce jednak te „studnie” nie są aż tak bardzo strome i głębokie



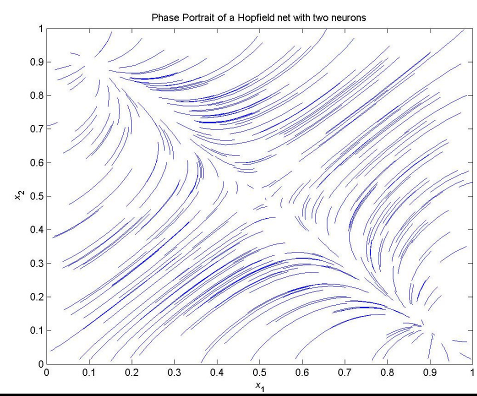
Z atraktorami w sieci Hopfielda można związać pewne informacje.



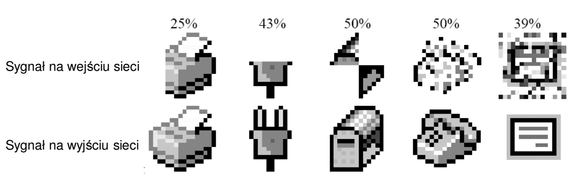
Przykład rzeczywistego systemu prostej pamięci skojarzeniowej



Portret fazowy rzeczywistej sieci Hopfielda



Nauczona sieć Hopfielda potrafi także odtwarzać całe obrazy na podstawie sygnałów wejściowych pokazujących obrazy fragmentaryczne



Dlatego główne zastosowanie Sieci Hopfielda wiąże się z budową tzw. pamięci autoasocjacyjnych

Ślady pamięciowe w sieci Hopfielda

Sposób powstawania śladów określonych wzorców w pamięci skojarzeniowej daje się uzasadnić w następujący sposób.

Załóżmy, że w sieci Hopfielda ma być zapamiętany tylko jeden wzorzec $D = < d_1, d_2, \ldots , d_n >^T$.

Jego stabilne odtwarzanie przez sieć będzie jednak możliwe wyłącznie wtedy, gdy dla każdego i-tego neuronu spełniony będzie warunek:

$$\varphi(\sum_{j=1}^{n} w_j^{(i)} d_j) = d_i$$

Równanie $\varphi(\sum_{j=1}^{n} w_j^{(i)} d_j) = d_i$ będzie spełnione

jeśli współczynniki wagowe w są wyrażane wzorem:

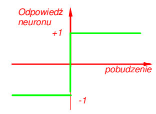
$$w_j^{(i)} = w_i^{(j)} = \eta \, d_i d_j$$

czyli mogą być wyznaczane zgodnie z regułą Hebba !

Zbadajmy to bliżej. Po podstawieniu otrzymujemy:

$$\sum_{j=1}^{n} \eta \, d_i d_j d_j = d_i \sum_{j=1}^{n} \eta \, d_j d_j = n \, \eta \, \|D\|^2 d_i$$

Uwzględniając fakt, że $d_i \in \{-1, 1\}$, a także biorąc pod uwagę kształt funkcji $\varphi\,(\,)$



możemy stwierdzić, że zawsze: $\boxed{\varphi\,(\sum_{j=1}^{n} \eta \; d_i d_j d_j) = d_i}$

W dalszych rozważaniach przyjmiemy $\eta = 1/n$ i będziemy zakładali (jeśli zajdzie potrzeba), że $\|D\| = 1$.

Zatem podstawowy wzór, opisujący sposób uczenia sieci będzie miał postać:

$$w_j^{(i)} = w_i^{(j)} = \frac{1}{n} d_i d_j$$

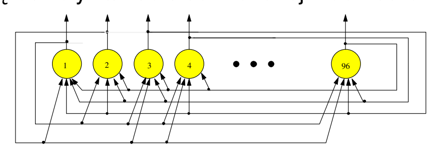
Można wykazać, że dla każdego X bliskiego D w sensie metryki Hamminga:

$$\rho_H(\mathbf{X}, D) < \varepsilon$$

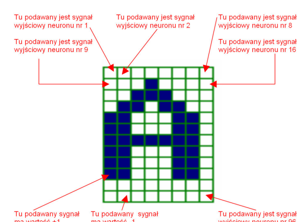
dla większości i zachodzi:

$$\varphi(\sum_{j=1}^{n} w_j^{(i)} x_j) = d_i$$

W prezentowanych dalej przykładach będziemy się odwoływać do sieci złożonej z 96 neuronów



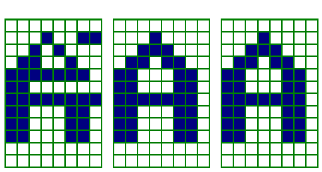
Żeby łatwiej obserwować, co się dzieje w tej sieci Hopfielda, załóżmy, że sygnały wyjściowe neuronów potraktujemy jako piksele pewnego obrazka



W sieci wyuczonej sygnałem D i pobudzonej sygnałem X przebiega pewien proces dynamiczny, polegający na kolejnym przyjmowaniu przez nią stanów $X_1, X_2, X_3, \ldots X_k$ systematycznie zbliżających się do wektora D

$$X \Rightarrow X_1 \Rightarrow X_2 \Rightarrow \ldots \Rightarrow D$$

Przykład:



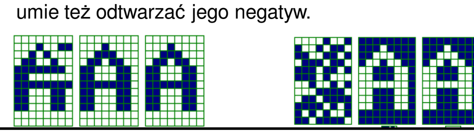
Nieparzystość funkcji φ gwarantuje, że z zależności:

$$\varphi(\sum_{j=1}^{n} w_j^{(i)} d_j) = d_i$$

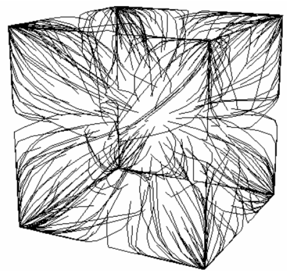
wynika natychmiast zależność:

$$\varphi(\sum_{j=1}^{n} w_j^{(i)} (-d_j)) = -d_i$$

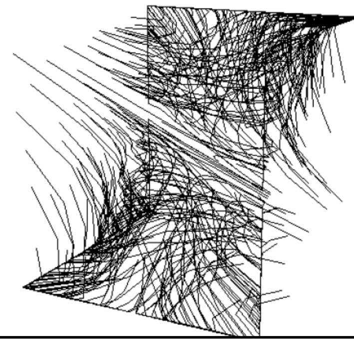
czyli sieć wyuczona zapamiętywania pewnego wzorca umie też odtwarzać jego negatyw.



Rzeczywiste trajektorie obserwowane w sieciach Hopfielda (przykład dla czterech atraktorów i ich zwierciadlanych przeciw-obrazów)



Inny przykład trajektorii obserwowanych w sieci Hopfielda, gdy procesy dynamiczne są rozbieżne



W przypadku, kiedy sieć musi zapamiętać większą liczbę wzorców D^{ν} możemy przeprowadzić podobne rozumowanie, jak wyżej przytoczone, wykazując, że **każdy** z tych wzorców jest atraktorem dla procesów dynamicznych przebiegających w sieci.

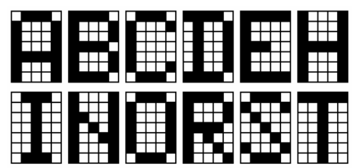
Oznacza to, że dla każdego ν i dla każdego i spełniony jest warunek:

$$\varphi(e^{(i)(\nu)}) = d_i^{(\nu)}$$

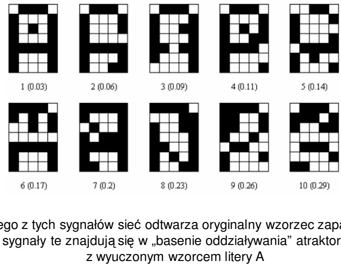
gdzie $e^{(i)(\nu)}$ oznacza wypadkowe pobudzenie *i*-tego neuronu po podaniu na jego wejście sygnałów odpowiadających składowym wzorca $D^{(\nu)}$, obliczane ze wzoru:

$$e^{(i)(\nu)} = \sum_{j=1}^{n} w_j^{(i)} d_j^{(\nu)}$$

Przykładowy zestaw wzorców dosyć często stosowanych przy badaniu sieci Hopfielda jako pamięci skojarzeniowych

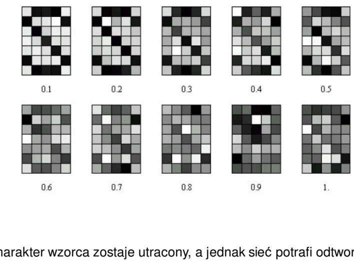


Różny sposób zaszumienia wzorca dla sieci Hopfielda prezentowany przy założeniu utrzymania binarnego charakteru wzorca (w nawiasach podane są odległości Hamminga od oryginału)



Po podaniu każdego z tych sygnałów sieć odtwarza oryginalny wzorzec zapamiętanego obrazu (litery A), bo sygnały te znajdują się w „basenie oddziaływania" atraktora związanego z wyuczonym wzorcem litery A

Jeden z wzorców zapamiętanych w sieci Hopfielda z różnym stopniem zaszumienia analogowego



Binarny charakter wzorca zostaje utracony, a jednak sieć potrafi odtworzyć oryginał!

Żeby wykazać, że każdy $D^{(\nu)}$ jest atraktorem, rozpiszemy wzór

$$e^{(i)(\nu)} = \sum_{j=1}^{n} w_j^{(i)} d_j^{(\nu)}$$

uwzględniając w nim sposób wyznaczania wartości wag $w_j^{(\nu)}$ zgodnie z regułą Hebba - tym razem oczywiście dla *M* wzorców $D^{(\nu)}$:

$$w_j^{(i)} = \frac{1}{n} \sum_{\mu=1}^{M} d_i^{(\mu)} d_j^{(\mu)}$$

Otrzymujemy:

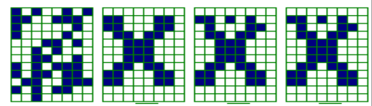
$$e^{(i)(\nu)} = \sum_{j=1}^{n} w_j^{(i)} d_j^{(\nu)} = \frac{1}{n} \sum_{j=1}^{n} \sum_{\mu=1}^{M} d_i^{(\mu)} d_j^{(\mu)} d_j^{(\nu)}$$

Wyrażenie dla $e^{(i)(\nu)}$ można rozpisać na dwie składowe:

$$e^{(i)(\nu)} = d_i^{(\nu)} + \frac{1}{n} \sum_{j=1}^{n} \sum_{\mu \neq \nu}^{M} d_i^{(\mu)} d_j^{(\mu)} d_j^{(\nu)}$$

z których drugą, obliczaną ze wzoru $\frac{1}{n} \sum_{j=1}^{n} \sum_{\mu \neq \nu} d_i^{(\mu)} d_j^{(\nu)} d_j^{(\mu)}$ nazywamy „przesłuchem”

Przesłuch jest to składnik pochodzący od innych wzorców $D^{(\mu)}$, utrudniający odtworzenie wzorca $D^{(\nu)}$



Jeśli przesłuch jest zaniedbywalnie mały, to znaczy jeśli:

$$\frac{1}{n} \sum_{j=1}^{n} \sum_{\mu \neq \nu} d_i^{(\mu)} d_j^{(\nu)} d_j^{(\mu)} < 1$$

wówczas nie zmienia on **znaku** pobudzenia $e^{(i)(\nu)}$ w stosunku do znaku składnika $d_i^{(\nu)}$

Wówczas spełniony jest warunek:

$$\varphi(e^{(i)(\nu)}) = d_i^{(\nu)}$$

a także zapewniona jest zbieżność procesów dynamicznych zachodzących w sieci do wzorca $D^{(\nu)}$ dla każdego *X* spełniającego warunek:

$$\rho_H(\mathbf{X}, D^{(\nu)}) << \rho_H(\mathbf{X}, D^{(\mu)})$$

Przytoczone wyżej rozumowanie dotyczące wielkości "przesłuchu" w sieci Hopfielda pozwala oszacować potencjalna pojemność takiej sieci M_{max} traktowanej jako pamięć asocjacyjna.

Przydatny w tych rozważaniach będzie wskaźnik stabilności *i*-tej składowej wektora generowanego przez sieć dla atraktora $D^{(\mu)}$.

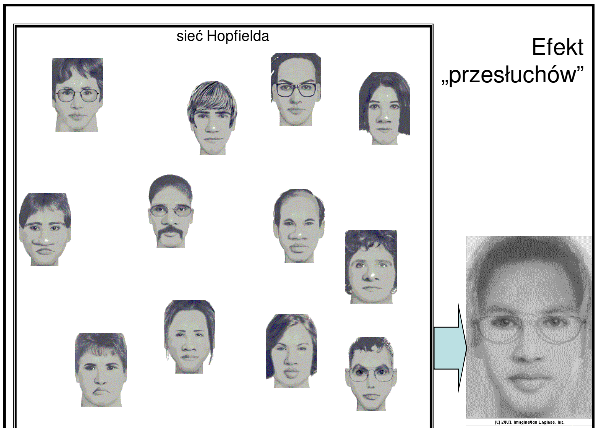
Wskaźnik ten oznaczony $c_i^{(\nu)}$ obliczymy ze wzoru:

$$c_i^{(\nu)} = -d_i^{(\nu)} \frac{1}{n} \sum_{j=1}^{n} \sum_{\mu \neq \nu}^{M} d_i^{(\nu)} d_j^{(\mu)} d_j^{(\nu)})$$

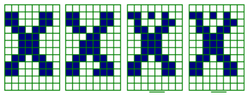
Wzór ten dość łatwo można zinterpretować: jest to pożądany sygnał wyjściowy *i*-tego neuronu $d_i^{(\nu)}$ przemnożony przez (ujemny) składnik "przesłuchu" we wzorze opisującym odtwarzanie zapamiętanych danych w sieci.

$$c_i^{(\nu)} = -d_i^{(\nu)} \frac{1}{n} \sum_{j=1}^{n} \sum_{\mu \neq \nu}^{M} d_i^{(\nu)} d_j^{(\mu)} d_j^{(\nu)})$$

- Jeśli $c_i^{(\nu)} < 0$ to składnik "przesłuchu" ma taki sam znak, jak pożądany sygnał wyjściowy $d_i^{(\nu)}$, a zatem nie ma zakłóceń.
- Jeśli $0 < c_i^{(\nu)} < 1$ to składnik "przesłuchu" jest za słaby, żeby zagłuszyć pożądany sygnał wyjściowy $d_i^{(\nu)}$, a zatem w sieci nadal nie ma zakłóceń.
- Jeśli natomiast $c_i^{(\nu)} > 1$, to pod wpływem działania "przesłuchu" zmieniać się będzie znak φ i w rezultacie zmieni się także wartość $d_i^{(\nu)}$.



Oznacza to, że przy $c_i^{(\nu)} > 1$ zachowanie sieci będzie niestabilne; nawet startując od poprawnej wartości $d_i^{(\nu)}$ sieć może się "zgubić".



Można przyjąć, że prawdopodobieństwo P_b popełnienia przez sieć błędu jest równe :

$$P_b = \mathrm{Prawd}(c_i^{(\nu)} > 1)$$

Zachowanie sieci zależy więc od wartości $c_j^{(\nu)}$, a te są z kolei zależne od ilości i jakości zapisywanych w pamięci wzorców $D^{(\nu)}$.

Oczywiście zapisywane w pamięci informacje są niemożliwe do apriorycznego przewidzenia, jednak możliwe jest wprowadzenie na ich temat prostych założeń o charakterze probabilistycznym.

Najprostszym założeniem może być to, że $d_i^{(\nu)}$ mogą przyjmować z jednakowym prawdopodobieństwem wartości +1 i -1 niezależnie od numeru wzorca n i niezależnie od wybranej składowej wektora *i*:

$$\forall_{\nu} \forall_{i} [\mathrm{Prawd}(d_i^{(\nu)} = 1) = \mathrm{Prawd}(d_i^{(\nu)} = -1)]$$

Jak z tego wynika wartości $c_i^{(\nu)}$ mogą być traktowane jako iloczyn stałego czynnika $1/n$ i sumy $(M\text{-}1)(n\text{-}1)$ liczb losowych, z których każda jest równa +1 lub -1.

Dla dużych *M* różnica miedzy *M* i (*M*-1) jest pomijalna, podobnie zaniedbamy w dalszych rozważaniach rozróżnienie między *n* i (*n*-1).

Ze statystyki wynika, że wartości $c_i^{(\nu)}$ podlegają wtedy **rozkładowi dwumianowemu** o zerowej średniej i wariancji $\sigma = M/n$.

Dla dużych wartości *M* oraz *n* **rozkład dwumianowy** aproksymować można **rozkładem normalnym**, zatem:

$$P_b = \Pr awd(c_i^{(\nu)} \rangle 1) = \frac{1}{2\sqrt{\pi}\sigma}\int_1^{\infty} e^{\frac{-x^2}{2\sigma}} dx$$

Dla potrzeb obliczeń numerycznych przytoczony wzór wygodnie jest przekształcić do postaci:

$$P_b = \Pr awd(c_i^{(\nu)} \rangle 1) = \frac{1}{2}\left[1 - erf\left(\sqrt{\frac{n}{2M}}\right)\right]$$

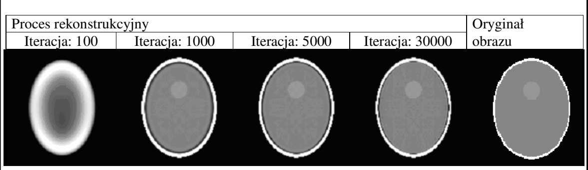
gdzie

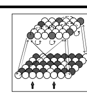
$$erf(x) = \frac{1}{\sqrt{\pi}}\int_0^x e^{-u^2} du$$

Po przeprowadzeniu stosownych obliczeń otrzymuje się następujące wartości oszacowania wielkości pojemności pamięci sieci M_{max} w zależności od liczby jej elementów *n* przy różnych akceptowanych stopach błędu P_b:

P_b	$\frac{M_{max}}{n}$
0.1	0.61
0.05	0.37
0.01	0.185
0.001	0.105

Przykład praktycznego zastosowania: Neuronalny algorytm rekonstrukcji obrazu z projekcji wykonanych w tomografie spiralnym





Pamięci asocjacyjne

Jednym z celów pamięci asocjacyjnych jest **analiza** i **reprodukcja umiejętności człowieka** dotyczących **kojarzenia faktów**

Podstawowe paradygmaty skojarzeniowe to: **auto-asocjacje**, **hetero-asocjacje** oraz asocjacje **epizodyczne**.
Nowymi formami skojarzeniowymi są asocjacje **wielokierunkowo epizodyczne,** zwane wieloskładnikowymi (tj. wiele-do-wielu).

Metody reprezentacji danych

Natura **rozproszonej równoległości przetwarzania informacji** za pomocą sieci neuronowych pasuje bardziej do przetwarzania informacji reprezentowanej w sposób **rozproszony** niż **lokalny**.

- W **rozproszonej** (kolektywnej) metodzie - informacja w sieci reprezentowana jest przez rozproszoną aktywność wielu neuronów, a **każdy neuron ma swój udział w reprezentowaniu wielu różnych informacji**.
- W metodzie **lokalnej** (skupionej) - **jedna informacja jest reprezentowana przez jeden neuron**.

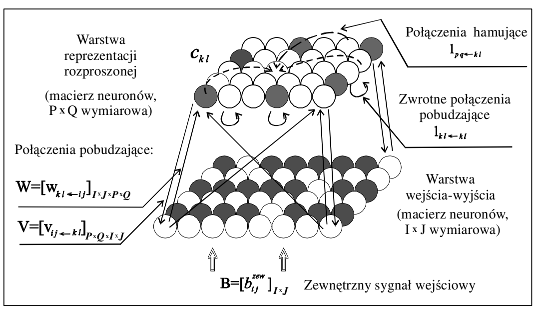
Sieci konkurencyjne o wielu zwycięzcach

Kiedy istotna jest **wydajność** reprezentowania informacji oraz **odporność sieci na zniszczenie**, to reprezentacja **rozproszona** jest zawsze **lepsza niż lokalna**.

Jednym ze **sposobów uzyskania reprezentacji rozproszonej** w sieciach neuronowych jest **realizowanie konkurencji z wieloma zwycięzcami**.

W literaturze do tej pory brak było przykładów samo-organizujących się sieci konkurencyjnych, które potrafiłyby automatycznie generować wewnętrzną reprezentację informacji wejściowej.

Struktura sieci MWSONN



W warstwie górnej użyta jest funkcja połączeń zwana **zasadą hamowań obocznych.**

Przetwarzanie danych w modelu MWSONN

Proces **zapamiętywania** w modelu pamięci MWSONN składa się z dwóch następujących po sobie procedur, tj. z **procedury konkurencji o wielu zwycięzcach** oraz z **procedury uczenia**.

Ponieważ obie te procedury są dwoma **niezależnymi** procesami, w procesie uczenia może być zastosowany także **inny algorytm uczenia** (np. dowolny algorytm ewolucyjny) i **nie zmieni to** podstawowych zasad **działania sieci** MWSONN.

Ewolucyjne algorytmy uczenia

- Algorytmy te będąc inspirowane przez ewolucyjne procesy „**przetrwania przystosowanych**” – „startują” z populacją osobników a nie z jednym punktem wejściowym.
- W kolejnych generacjach **stopniowo maksymalizują** one funkcję kosztu (zwaną **funkcją przystosowania**)

  i będącą heurystyczną oceną „chromosomu”.
- W każdym pokoleniu **oceniany jest każdy** nowy osobnik: „najlepsze - biorą udział w krzyżowaniu i mutacji; najsłabiej przystosowane osobniki są usuwane”.

Szybkość zbieżności analitycznej metody uczenia sieci MWSONN

Duża szybkość zbieżności oryginalnej analitycznej metody uczenia sieci, będąca konsekwencją jej **wysokiej precyzji** i **wyłącznie deterministycznie adaptacyjnie wyznaczonych poprawek**, a przez to efektywności w iteracyjnym modyfikowaniu wag, zdecydowanie **przewyższała swą jakością** możliwości badanych **klasycznych programów ewolucyjnych** (np. **algorytmów genetycznych**).

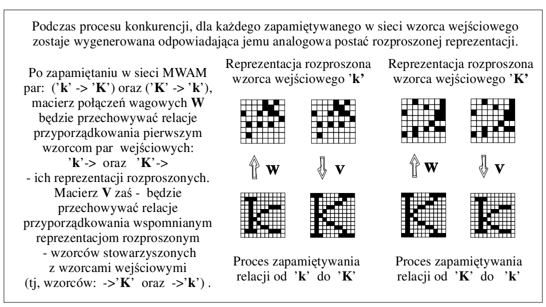
Włączenie wiedzy o rozwiązaniach a zbieżność algorytmu ewolucyjnego

Rezultaty symulacji potwierdziły opinie Prof. Zb. Michalewicza mówiące, że **włączenie wiedzy o rozwiązaniach zwiększa skuteczność algorytmu ewolucyjnego** oraz czyni efektywniejszym proces przeszukiwania przestrzeni rozwiązań dopuszczalnych, oraz że w przypadku **zadań o wielkiej skali** (tj. zadań o tysiącach zmiennych), co miało miejsce w tym przypadku, **decyduje to o jakości algorytmu**.

Losowość a wydajność uczenia algorytmem ewolucyjnym w sieci o 9 tysiącach połączeń

Czynnik losowości, wprowadzany w krokach modyfikujących osobniki populacji i ich **tysiące zmiennych połączeń wagowych**, które w rozważanym tutaj przypadku, dodatkowo mnożone były przez około 100 wzorców zbioru wejściowego, we wszystkich testowanych wersjach klasycznego algorytmu genetycznego, powodował **znaczną degradację** procesu zbieżności.

Hetero-asocjacyjna pamięć o wielu zwycięzcach (MWAM)

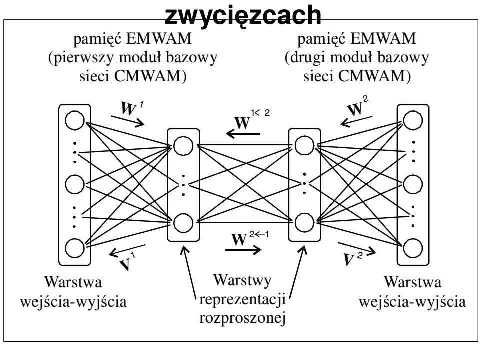


Epizodyczna pamięć asocjacyjna EMWAM o wielu zwycięzcach

Sieć ta może zapamiętywać **wiele sekwencji wielo-wzorcowych**. Proces ich zapamiętywania jest niemal Identyczny do odpowiadającego mu procesu w sieci MWAM, z tą różnicą, że **zamiast zapamiętywania** w sieci zbioru ***L*** **niezależnych par wzorców**: w sieci EMWAM **każda para jest „związana" z parą następną** w kolejności.

Wzorzec – to dowolne wartości **analogowe** macierzy kwadratowej o wymiarach warstwy wejściowej.
Wartości te zazwyczaj należą do przedziału **[-1, 1]** lub **[0, 1]**.

Dwu-kierunkowo epizodyczna pamięć asocjacyjna CMWAM o wielu zwycięzcach



Możliwości pamięci asocjacyjnej CMWAM

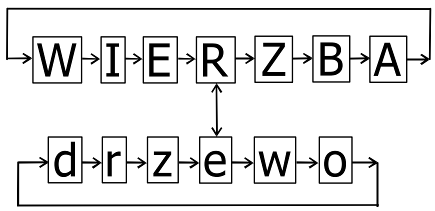
Sieć ta może zapamiętywać różne konstrukcje skojarzeniowe, takie jak: **auto-asocjacje**, **hetero-asocjacje,** asocjacje **sekwencyjne** (tj. epizodyczne) oraz dodatkowo: ***relacje pomiędzy dwoma dowolnymi asocjacjami***, tj. potrafi tworzyć asocjacje tzw. **dwu-epizodyczne,** które symbolicznie można określić jako:

epizod-1 ↔ epizod-2

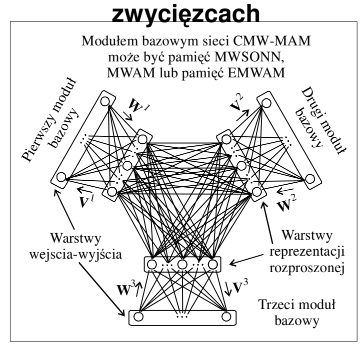
gdzie ***epizod-i*** definiujemy jako **zbiór asocjacji** pomiędzy wzorcami należącymi do *n*-wzorcowego epizodu.

Schemat odtwarzania dwóch epizodów

Sieć CMWAM może odtworzyć nie tylko zaszumiony wzorzec podany na odpowiednie wejście tej sieci, ale i całą sekwencję wzorców do której ten wzorzec należy oraz sekwencję z nią skojarzoną.



Wielokierunkowa epizodyczna pamięć asocjacyjna CMW-MAM o wielu zwycięzcach



Możliwości pamięci asocjacyjnej CMW-MAM

- Umiejętność zapamiętywania i odtwarzania **zwielokrotnionych oraz skomplikowanych asocjacji** oznacza **zdolność odtwarzania zwielokrotnionej** zawartości pewnej **małej** zawartości.
- W sieci CMW-MAM, **każdy z podstawowych modułów,** zależnie od zastosowanego w nim odpowiedniego procesu uczenia - **może realizować dowolne elementarne asocjacje**, tj. auto-asocjacje, hetero-asocjacje oraz asocjacje epizodyczne.

Dalsze możliwości pamięci asocjacyjnej CMW-MAM

Ponieważ moduł podstawowy może realizować **wszystkie trzy rodzaje asocjacji**, a sieć CMW-MAM zawiera w sobie także strukturę sieci MAM, więc **każdy *M*-warstwowy model** pamięci CMW-MAM może z kolei realizować **bardziej skomplikowane grupowe asocjacje** typu *„many-to-many”,* wzajemnie skorelowane poprzez połączenia wagowe pomiędzy warstwami reprezentacji.

Schemat odtwarzania trzech epizodów

Sieć CMWAM może odtworzyć nie tylko zaszumiony wzorzec podany na odpowiednie wejście tej sieci, ale i całą sekwencję wzorców do której ten wzorzec należy oraz sekwencję z nią skojarzoną.