Sieci Neuronowe

Wykład 6
Sieci pamieci skojarzeniowej:
sieci Hamminga, sieci Hintona, sieci BAM

wykład przygotowany na podstawie.
S. Osowski, Sieci neuronowe w ujęciu algorytmicznym, WNT, W-wa 1996.
R. Tadeusiewicz, "Sieci Neuronowe", Rozdz. 8. Akademicka Oficyna Wydawnicza RM, Warszawa 1993.

Pamięć skojarzeniowa

Pamięć skojarzeniowa (asocjacyjna) jest jednym z podstawowych atrybutów ludzkiego mózgu.

Ma dwie istotne cechy:

→ informacje zarejestrowane w pamięci asocjacyjnej mogą być dostępne poprzez podanie na wejściu systemu informacji skojarzonej, selekcjonującej jedną z zapamiętanych wiadomości *na drodze asocjacji*.

→ ślad pamięciowy, zwany *engramem*, nie ma w pamięci asocjacyjnej ściślej lokalizacji, każda zarejestrowana informacja zlokalizowana jest w istocie w *całej* pamięci, na zasadzie kolektywnego działania wszystkich jej elementów. Taka technika zapisu informacji nazywa się *hologiczna*. Jej fizycznym przykładem jest *hologram*.

Pamięć skojarzeniowa

Istnieje możliwość realizacji pamięci asocjacyjnej przy pomocy sieci neuronowej.

Omówimy przykłady:

sieci Hamminga, sieci Hintona, sieci BAM

Nb. klasyczne systemy obliczeniowe, bazy danych czy systemy ekspertowe selekcjonują zapamiętane informacje na drodze dostępu adresowego.

Odległość Hamminga

Zadaniem pamieci associacyjnej jest zapamiętanie *wzorców wejściowych* (uczących) w taki sposób, aby w trybie odtwarzania przy prezentacji nowego wzorca układ mógł wygenerować odpowiedź, która będzie odpowiadać jednemu z zapamiętanych wcześniej wzorców, położonemu najbliżej próbki testującej.

Najczęściej używaną miarą odległości między zbiorami w przypadku pamięci asocjacyjnej jest miara Hamminga.
Dla wielkości binarnych 0,1 odległość Hamminga dwóch wektorów

$$y = [y_1, y_2, \ldots y_n]^T \quad ; \quad x = [x_1, x_2, \ldots . x_n]$$

definiuje się w postaci

$$d_H = \sum [\, x_i \,(1-y_i) + (1-x_i)\, y_i \,]$$

Odległość Hamminga

Miara Hamminga jest równa zeru jedynie wówczas gdy y=x.

W przeciwnym przypadku jest ona równa liczbie bitów o które różnią się oba wektory.

Uczenie (training) sieci neuronowej do pełnienia funkcji pamięci asocjacyjnej ma za zadanie taki dobór wag W_{ij} poszczególnych neuronów, aby na etapie odtwarzania sieć była zdolna odnaleźć zbiór danych, najbliższych w sensie miary Hamminga, wektorowi testowemu.

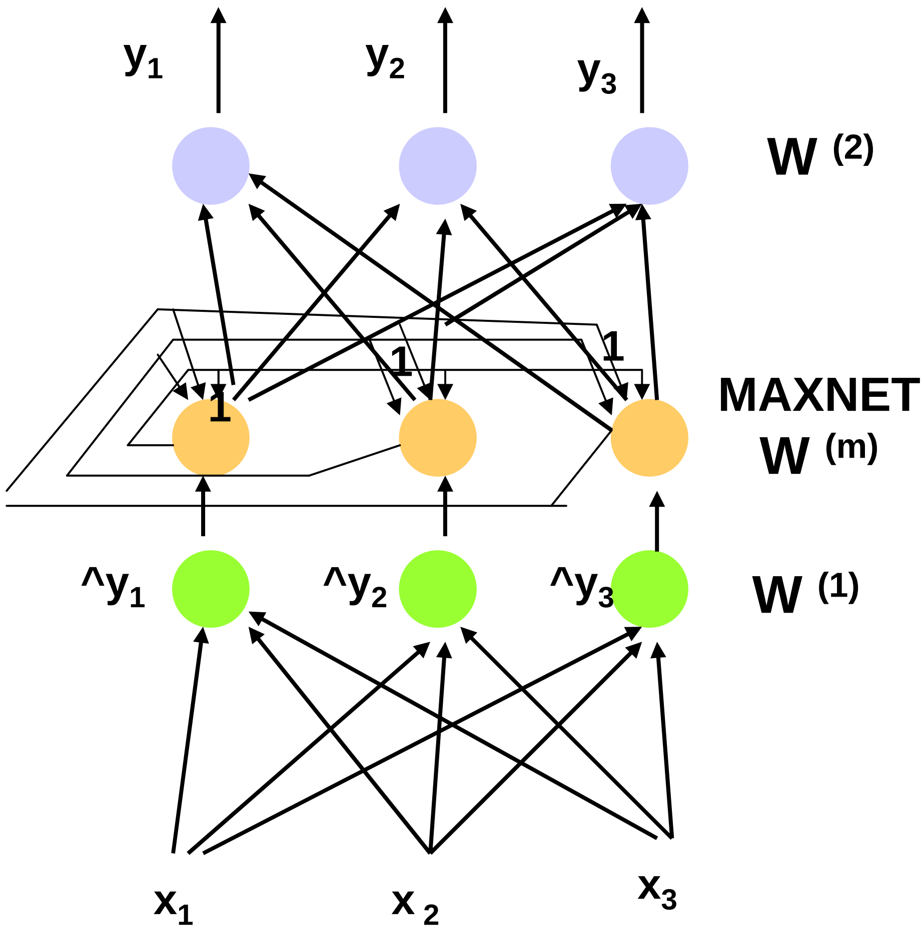
Proces uczenia sieci kształtuje *obszary przyciągania* (atrakcji) poszczególnych punktów równowagi, odpowiadających danym uczącym. W przypadku pamięci autoasocjacyjnej wystepuje wektor uczący x lub zbiór tych wektorów, które w wyniku przeprowadzonego uczenia sieci ustalają położenia poszczegolnych *atraktorów*.

Sieć Hamminga

Sieć Hamminga jest *trójwarstwową struktura rekurencyjną*, będącą rozwinięciem sieci Hopfielda. Jej zasada działania jako klasyfikatora wektorowego polega na minimalizacji odległości Hamminga wektora testowego podanego na wejście sieci od wektorów reprezentujących wzorce uczące, zakodowane w strukturze sieci.

- ✓ warstwa pierwsza jest jednokierunkowa z przepływem sygnałów od wejścia do wyjścia i ustalonymi wartościami wag
- ✓ warstwę drugą tzw. MAXNET stanowią neurony połączone sprzężeniem zwrotnym każdy z każdym, przy czym w odróżnieniu od struktury Hopfielda istnieje niezerowe sprzężenie neuronu z własnym wyjściem
- ✓ warstwa wyjściowa jednokierunkowa reprezentuje wektor wyjściowy stowarzyszony z wektorem wejściowym, wagi neuronów tej warstwy są ustalone w zależności od tego wektora.

Sieć Hamminga



Wagi w warstwie MAXNET są stale
Sprzężenie neuronu z własnym wyjściem ma wagę 1, pozostałe sprzężenia są gaszące ($\omega = \varepsilon$).
W stanie ustalonym tylko jeden neuron jest pobudzony, pozostałe są w stanie spoczynku.

Sieć Hamminga

W działaniu sieci można wyróżnić dwie fazy:

→ w fazie pierwszej na wejściu sieci jest prezentowany N-elementowy wektor x. Czas prezentacji wektora musi byc dostatecznie długi aby neurony warstwy pierwszej uzyskały ustalone wartości synałów wyjściowych, initializujące stan początkowy neuronów warstwy drugiej.

→ w fazie drugiej waktor x jest usuwany i rozpoczyna się proces iteracyjny w warstwie MAXNET. Trwa on do chwili aż wszystkie neurony w wyjątkiem jednego osięgną stan zerowy. Neuron o niezerowym sygnale wyjściowym reprezentuje klasę do której należy wektor wejściowy.

Sieć Hamminga

Neurony warstwy pierwszej określają odległość Hamminga pomiędzy aktualnie podanym wektorem wejściowym x a każdym z p zakodowanych wektorów wzorcowych $x^{(i)}$.

Neurony w warstwie **MAXNET** wykrywają wektor o najmniejszej odległości Hamminga, określając w ten sposób klasę, do której należy dany wektor wyjściowy x.

Wagi neuronów warstwy wyjściowej odtwarzają wektor stowarzyszony z danym wektorem wejściowym. Przy p wektorach w warstwie pierwszej pojemność pamieci Hamminga jest równa p, gdyż każdy neuron reprezentuje klasę. Jeżeli liczba zapamiętanych wzorców m jest mniejsza niż p, zakłada się że pozostałe p-m wzorców są zerowe.

Sieć Hamminga

Initializacja wag sieci jest bardzo prosta. Wagi warstwy pierwszej reprezentują kolejne wektory wzorcowe $x^{(i)}$; stąd

$$W_{ij}^{(1)} = x_j^{(i)}$$

Podobnie wagi neuronów warstwy wyjściowej reprezentują kolejne wektory wzorcowe $y_j^{(i)}$, stowarzyszone z $x_i^{(j)}$

$$W_{ji}^{(2)} = y_j^{(i)}$$

W przypadku neuronów w warstwie MAXNET, initializacja wag sieci ma za zadanie wzmocnić własny sygnał neuronu i osłabić pozostałe. Stąd przyjmuje się

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

oraz

$$- 1/(p-1) < W_{ij}^{(m)} < 0$$

Sieć Hamminga

Dla zapewnienia bezwględnej zbieżności algorytmu wagi $W_{ij}^{(m)}$ powinny się różnić między sobą. Orginalne rozwiązanie Lippmanna

$$\boxed{W_{ij}^{(m)} = -\,1/(p-1) + \xi}$$

przy czym ξ jest wartością losową o dostatecznie małej amplitudzie.

Na podstawie licznych badań eksperymentalnych wykazano że sieć rekurencyjna Hamminga daje lepsze rezultaty niż sieć Hopfielda, w szczególności dla przypadków , w których wektory stowarzyszone x i y są losowe.

Ważna zaleta sieci Hamminga jest duża oszczędność połączeń wagowych między neuronami.

Sieć Hintona

Rozważmy sieć o dwóch warstwach z pełnym ("każdy z każdym") schematem połączeń elementów pierwszej warstwy z elementami drugiej warstwy.
→ Niech sygnały wejściowe podane do pierwszej warstwy tworzą wektor X, a sygnały wyjściowe, powstające na wyjściach drugiej warstwy, tworzą wektor Y.

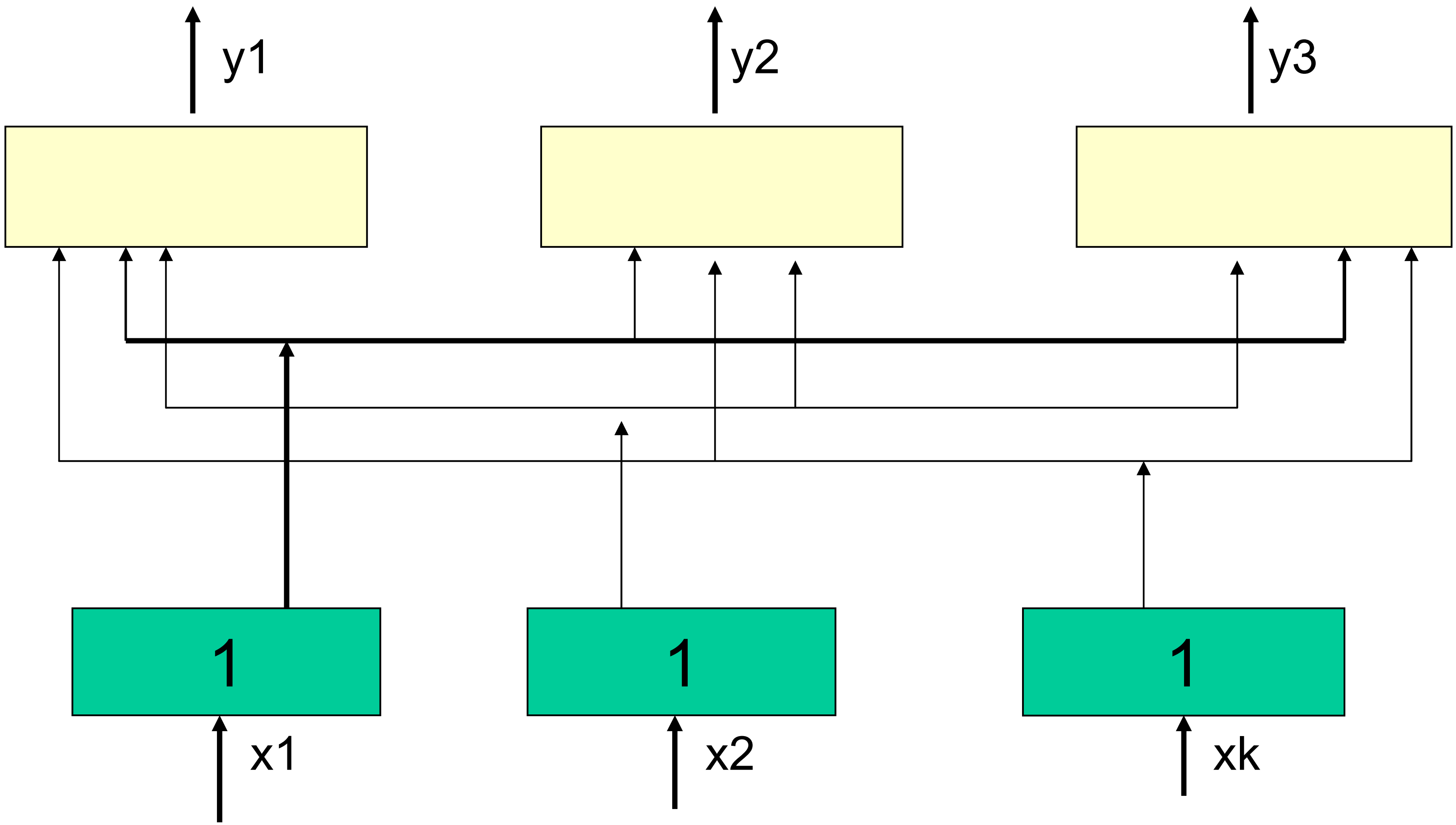
→ Pierwsza warstwa nie będzie prawie uczestniczyła procesie przetwarzania informacji i będzie pełniła raczej funkcje elementu normalizującego sygnały.

→ Sens działania sieci polega na odpowiednim doborze wag $\omega_i^{(j)}$, gdyż w nich zawierać się musi cala wiedza systemu. Wiedza ta pochodzi od ciągu uczącego U, którego struktura może być rozważana jako ciąg podlegających skojarzeniu par wektorów: wejściowego X i wyjściowego Y.

Zapiszmy ten ciąg w następujący sposób:

$$U = \{ < X^{(1)} , Y^{(1)} >, < X^{(2)} , Y^{(2)} > \ldots\ldots < X^{(N)} , Y^{(N)} >\}$$

Sieć Hintona



„każdy z każdym” schemat połączeń

Sieć Hintona

Zakładamy, że sieć podlega uczeniu metodą Hebba:

$$\omega_i^{(j)(k+1)} = \omega_i^{(j)(k+1)} + \eta x_i^{(k)} y_i^{(k)}$$

Dla całej macierzy wag W obowiązuje zatem rekurencyjna reguła

$$W^{(k+1)} = W^{(k)} + \eta \, X^{(k)} \, [Y^{(k)}]^T$$

po wykonaniu wszystkich N kroków uczenia mamy macierz wag W określoną wzorem

$$W = \sum_{k=1}^{N} X^{(k)} \, [Y^{(k)}]^T$$

Powyższy wzór jest prawdziwy pod warunkiem, że wszystkie wagi w sieci przed rozpoczęciem uczenia były zerowe ($W^{(1)} = 0$) oraz przy założeniu, że $\eta = 1$.

Sieć Hintona

Asocjacyjne i pamięciowe własności omówionej sieci mogą być prosto wykazane przy założeniu, że wektory wejściowe (wywołujące skojarzenia) będą ortonormalne, to znaczy

$$X^{(i)} [X^{(j)}]^T = \begin{cases} 1 & \text{gdy } i=j \\ 0 & \text{gdy } i \neq j \end{cases}$$

W takim przypadku, można precyzyjnie obliczyć

$$Y = W X^{(j)} = \sum_{k=1}^{N} X^{(k)} [Y^{(k)}]^T X^{(j)}$$

pomocniczo:

$$X^{(j)} [X^{(j)}]^T = 1$$

$$\sum_{k \neq j} X^{(k)} [X^{(j)}]^T Y^{(k)} = 0$$

Zgodnie z zasadami rachunku macierzowego

$$Y = \sum X^{(k)} [Y^{(k)}]^T X^{(j)} = \sum_{k \neq j} X^{(k)} [X^{(k)}]^T Y^{(j)}$$

co można łatwo przekształcić do postaci

$$Y = X^{(j)} [X^{(j)}]^T Y^{(j)} + \sum X^{(k)} [X^{(j)}]^T Y^{(k)}$$

co można zapisać jako

$$Y = Y^{(j)}$$

Sieć Hintona

Sieć Hintona po nauczeniu metodą Hebba, pobudzona sygnałem $X^{(j)}$, odtwarza na swoim wyjściu dokładnie skojarzony z tym sygnałem (podczas uczenia) sygnał wyjściowy $Y^{(j)}$.

Opisany wynik polegający na idealnym odtwarzaniu przez sieć zapamiętanych informacji – jest interesujący, ale ma raczej akademickie znaczenie, ponieważ *warunek ortonormalnosci* jest trudny do zachowania.

Dalsze analizy sieci Hintona wykazały że *można go znacznie osłabić*, bez utraty możliwości sensownego korzystania z pracy sieci.

Ogólne własności sieci Hintona

Podstawowe są prace Kohonena (1981), który rozpatrywał własności sieci autoasocjacyjnych i uzyskał bardzo ogólne wyniki, pozwalające określić warunki zapamiętywania i poprawnego odtwarzania informacji rejestrowanych przez sieć asocjacyjną.

Rozważamy sieć jednowarstwową o n-wejściach i k-wyjściach (liczba wyjść jest identyczna z liczbą neuronów w sieci).
Do każdego neuronu doprowadzone są wszystkie sygnały wejściowe, tworzące wektor $X = < x_1, x_2, ..., x_n >$ oraz *sygnał* f_i (gdzie i jest numerem neuronu), będący *"wymuszeniem"* powodującym określone zachowanie się sieci podczas procesu uczenia.

Ogólne własności sieci Hintona

Sygnał wyjściowy z neuronu o numerze i, wyznaczyć można z równania

$$y_i = \Sigma\, \omega_{ij}\, x_j + f_i$$

zaś proces uczenia prowadzony według zmodyfikowanego algorytmu Hebba opisuje

$$d/dt\, \omega_{ij} = \eta\, f_i\, x_j$$

Podane zależności dotyczą wszystkich neuronów i mogą być zapisane w formie

$$Y = W X + F$$
$$d/dt\, W = \eta\, F\, X^T$$

Rozwiązanie powyższego równania, przy założeniu że X i F pozostają niezmienne, ma postać

$$W(t) = \eta\, t\, F\, X^T$$

Ogólne własności sieci Hintona

Proces uczenia polegający na prezentacji ciągu uczącego złożonego z par wektorów $X^{(k)}$ i $F^{(k)}$:

$$U = \{ < X^{(1)}, F^{(1)} >, < X^{(2)}, F^{(2)} >, \ldots\ldots\ldots , < X^{(N)}, F^{(N)} > \}$$

przy czym zakładamy, że każda para $< X^{(k)}, F^{(k)} >$ prezentowana jest sieci w ciągu odcinka czasu $\tau=1/\eta$.

Po pełnym cyklu nauczania macierz wag W wyraża się wzorem

$$W = \Sigma\, X^{(k)} (F^{(k)})^{T}$$

Po zakończeniu procesu uczenia, sygnały wymuszające nie występują ($F=0$), natomiast sygnały wyjściowe z sieci są uwarunkowane wyłącznie przez jej sygnały wejściowe:

$$Y = W X$$

Oczekujemy, że sygnały te będą zbliżone do narzucanych w trakcie procesu uczenia, ale pewność że tak będzie mamy tylko dla ortonormalnych sygnalow $X^{(k)}$.

Ogólne własności sieci Hintona

Rozważmy sytuacje o łagodniejszych wymaganiach odnośnie struktury sygnałów $X^{(k)}$ w trakcie procesu uczenia.

Przy $X^{(k)}$ liniowo niezależnych możemy zapisać wynikową macierz wag po zakończeniu procesu uczenia jako

$$\boxed{W = F^{*} (X^{*T} X^{*})^{-1} X^{*T}}$$

gdzie przez F^{*} i X^{*} oznaczono zagregowane do postaci macierzowej wektory występujące w ciągu uczącym. Ich budowa jest następująca.

$$\boxed{F^{*} = [\, F^{(1)}\ F^{(2)}\ \ldots\ldots\ F^{(N)}\,], \quad X^{*} = [\, X^{(1)}\ X^{(2)}\ \ldots\ldots\ X^{(N)}\,]}$$

Jeśli jednak wektory $X^{(k)}$ nie są *liniowo niezależne* to wówczas macierz wag W może być wyznaczona w postaci

$\boxed{W = F^{*} X^{*+}}$, gdzie macierz X^{*+} jest sprzężona hermitowsko do X^{*}.

Ogólne własności sieci Hintona

Tak więc sieć po nauczeniu rejestruje w swojej pamięci korelacyjne związki miedzy sygnałem wejściowym X i wymuszającym F. Brak liniowej niezależności powoduje, że *zamiast idealnej macierzy korelacji* uzyskujemy jej *optymalną*
(w sensie średniokwadratowym) *aproksymację*.

Rozważmy ważny z praktycznego punktu widzenia przypadek szczególny $F^*=X^*$, wówczas

$$\boxed{W = F^* F^{*+}}$$

i operacja wykonywana przez sieć nazywana jest ortogonalną projekcją, a jej interpretacja jest następująca:

Ogólne własności sieci Hintona

Dane w ciągu uczącym N wektorów $F^{(k)}$ rozpina w k-wymiarowej przestrzeni pewna podprzestrzeń liniowa L. Każdy konkretny wektor F może być wtedy poddany rzutowaniu na te podprzestrzeń, w wyniku czego może być rozłożony na sumę dwóch wektorów

$$F = F_L + F_R$$

gdzie

$$F_L = F^* F^* + F$$

jest efektem rzutowania wektora F na podprzestrzeń L, a wektor F_R może być traktowany jako *residuum*, zależnie od interpretacji jest to albo *"wektor błędów"* (jeżeli celem jest optymalne wykonanie rzutowania) albo *"wektor nowości"* (jeżeli celem jest ma być wykrycie rozbieżności pomiędzy sygnałem aktualnie pojawiającym się na wejściu sieci, a wektorami zapamiętanymi w trakcie procesu uczenia.

Dwukierunkowa pamięć autoasocjacyjna

Rozwinięciem omówionych koncepcji Hopfielda i Hintona jest

BAM – Bidirectional Associative Memory

czyli dwukierunkowa pamięć skojarzeniowa.

Sieć *BAM* jest częściowo oparta na koncepcji sieci *ART*, gdyż kluczem jej działania jest wzajemne oddziaływanie sygnałów wejściowych i wyjściowych w zamkniętej pętli sprzężenia zwrotnego. Jednak cechy sieci *BAM* są na tyle odmienne od sieci *ART,* że na ogól rozważa się je oddzielnie.

Dwukierunkowa pamięć autoasocjacyjna

W sieci *BAM* występują dwa wektory sygnałów, oznaczane zwykle jako

$$X \ (X \in R^n) \text{ i } \ Y \ (Y \in R^m).$$

Zadaniem sieci jest przyjęcie w trakcie procesu uczenia i zarejestrowanie (w postaci wag w_{ij}) zależności pomiędzy wejściem X i wyjściem Y.

Wykorzystywany jest ciąg uczący postaci

$$U = \{ < X^{(1)}, Y^{(1)} >, < X^{(2)}, Y^{(2)} >, \dots\dots\dots , < X^{(N)}, Y^{(N)} > \}$$

w taki sposób, aby po ponownym pojawieniu się pewnego konkretnego wektora $X^{(k)}$ (albo wektora podobnego - sieć jest asocjacyjna, czyli jest w stanie uogólniać swoje doświadczenia) sieć odtwarzała na swoim wyjściu wektor $Y^{(k)}$ zapamiętany jako skojarzony z danym X.

Dwukierunkowa pamięć autoasocjacyjna

Wektor X podawany jest na wejścia wszystkich neuronów wejściowej warstwy sieci, która jest połączona z warstwą wyjściową w taki sposób, że współczynniki wagowe tworzą macierz W o wymiarach [n x m].

W wyniku przejścia sygnałów $x_j \in X$ przez neurony wyjściowej warstwy powstaje wektor Y, którego składowe otrzymuje się według znanej zależności:

$$y_i = \phi\,(\Sigma\, w_{ij}\, x_j\,)$$

gdzie ϕ jest nieliniową funkcją wiążąca wejście z wyjściem w pojedynczym neuronie.

Dwukierunkowa pamięć autoasocjacyjna

Zależność tą można zapisać w formie wektorowej

$$Y = \Phi\ (W\ X)$$

gdzie W jest macierzą wag wyjściowej warstwy sieci, Y jest wektorem wyjściowym drugiej warstwy sieci, a Φ jest wielowymiarowym nieliniowym odwzorowaniem Φ: $R^n \Rightarrow R^n$ stanowiącym formalny agregat odwzorowań realizowanych skalarnie przez funkcję ϕ, które zwykle przyjmowane są w postaci funkcji progowej albo sigmoidalnej logistycznej

$$\phi(e) = 1/\ (1+ \exp(-\beta e))$$

Dwukierunkowa pamięć autoasocjacyjna

W dalszej dyskusji będziemy rozważać tylko model funkcji progowej. Wynikiem takiego założenia jest fakt, że wszystkie sygnały sieci są *bipolarne*, tzn.

$x_i \in \{1-, 1\}$ oraz $y_i \in \{1-, 1\}$

Ma to dość istotny wpływ na zachowanie sieci, którego śledzenie jest dzięki temu łatwiejsze niż w przypadku sygnałów przyjmujących (jak to typowo się zakłada dla innych sieci) wartości 0 i 1.

Dodatkowym założeniem jest przypisanie elementom *własności "histerezy". Objawia się to pewna "niechęcią"* każdego neuronu do zmiany stanu.

Dwukierunkowa pamięć autoasocjacyjna

Przy założeniu, że funkcja ϕ ma postać progową oznaczać to będzie, że przejście od wartości –1 do +1 lub na odwrót musi być każdorazowo wymuszone wyraźnie dodatnim lub ujemnym sygnałem wejściowym. Jeśli ważona suma sygnałów wejściowych, docierających do określonego neuronu, ma wartość 0, wówczas element ten *utrzymuje taki sam sygnał wyjściowy, jaki miał poprzednio*.

$$y_i(t+1) = \begin{cases} +1 & \text{gdy } \Sigma\, w_{ij}\, x_j(t) > 0 \\ y_i(t) & \text{gdy } \Sigma\, w_{ij}\, x_j(t) = 0 \\ -1 & \text{gdy } \Sigma\, w_{ij}\, x_j(t) < 0 \end{cases}$$

Dwukierunkowa pamięć autoasocjacyjna

Wektor Y w sieci BAM zostaje skierowany na wejście pierwszej warstwy tworząc sprzężenie zwrotne, w którym wykorzystywana jest ta sama macierz W

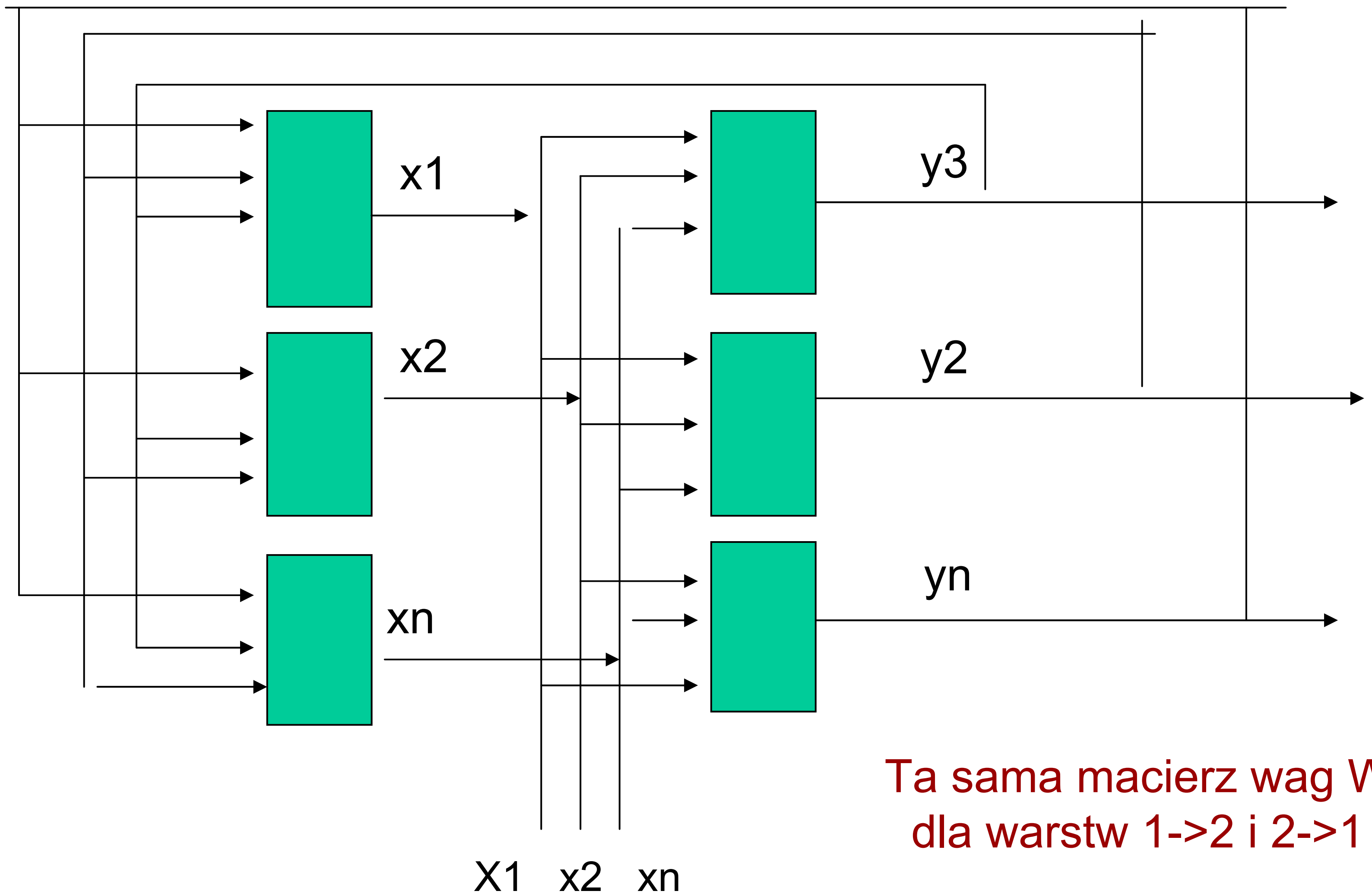
$$X = \Phi\,(W^T\,Y)$$

Sieć BAM ma bardziej regularna budowę, niż zwykle sieci Hopfielda czy Hintona.

Jest ponadto tworem czysto technicznym, trudno bowiem uznać za prawdziwy biologicznie proces uczenia utrzymujący stale ścisłą zgodność wartości pewnych wag synaptycznych w ustalonych parach neuronów.

Dzięki obecności w sieci BAM sprzężenia zwrotnego możliwe są w niej wewnętrzne obiegi sygnałów, podobnie jak w sieci Hopfielda. Sieć po podaniu sygnału wejściowego X zostaje pobudzona i poszukuje stanu równowagi, który osiąga *"przypomniawszy sobie"* odpowiedni sygnał *Y*.

Schemat sieci BAM



Ta sama macierz wag W
dla warstw 1->2 i 2->1

Uczenie sieci BAM i przykład jej działania

Uczenie sieci BAM podlega prawu Hebba przy współczynniku $\eta=1$, zatem po pokazaniu w ciągu uczącym pary sygnałów $< X^{(k)}, Y^{(k)} >$, macierz wag W zostaje zmodyfikowana o składnik będący macierzą korelacji $X^{(k)}, Y^{(k)}$.

$$W^{(k+1)} = W^{(k)} + \Delta W^{(k)} = W^{(k)} + X^{(k)} Y^{(k)T}$$

Po przeprowadzeniu procesu uczenia od początku do końca mamy więc następujący stan pamięci sieci (przy założeniu ze $W^{(1)} = 0$):

$$W = \sum \Delta W^{(k)}$$

Przeanalizujmy prosty przykład: $U = \{ < X^{(1)}, Y^{(1)} >, < X^{(2)}, Y^{(2)} >, < X^{(3)}, Y^{(3)} > \}$

gdzie

$$X^{(1)} = \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}, \quad X^{(2)} = \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}, \quad X^{(3)} = \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix}$$

$$Y^{(1)} = \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix}, \quad Y^{(2)} = \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}, \quad Y^{(3)} = \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}$$

$$\Delta W^{(1)} = \begin{pmatrix} -1 & -1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & -1 \end{pmatrix} \quad \Delta W^{(2)} = \begin{pmatrix} -1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix}$$

$$\Delta W^{(3)} = \begin{pmatrix} -1 & 1 & 1 \\ -1 & -1 & 1 \\ 1 & -1 & -1 \end{pmatrix}$$

Uczenie sieci BAM i przykład jej działania

Wynikowa macierz ma postać:

$$W = \begin{pmatrix} -1 & -1 & 3 \\ -1 & 3 & -1 \\ 3 & -1 & -1 \end{pmatrix}$$

Mając do dyspozycji *gotowa wynikową macierz wag* (czyli *zawartość pamięci sieci* BAM) możemy prześledzić, jak sieć działa podczas odtwarzania zapamiętanych informacji.
Niech na wejściu pojawi się sygnał: $X = \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}$

Porównanie z ciągiem uczącym ujawnia, że jest to sygnał $X^{(1)}$, który sieć powinna pamiętać. Obliczmy sygnał Y, jaki wygenerowuje sieć na swoim wyjściu:

$$Y = \Phi(WX) = \Phi\left(\begin{pmatrix} -1 & -1 & 3 \\ -1 & 3 & -1 \\ 3 & -1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix} \right) = \Phi\left(\begin{pmatrix} -3 \\ -3 \\ 5 \end{pmatrix} \right) = \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix}$$

Otrzymany wynik jest poprawny, odpowiada dokładnie $Y^{(1)}$, który w procesie uczenia był kojarzony z podanym na wejście sieci wektorem $X = X^{(1)}$.

Uczenie sieci BAM i przykład jej działania

Sieć poprawnie odtwarza skojarzone informacje w obu kierunkach. Rozważmy przykład zachowania sieci po podaniu na jej wejście Y sygnału odpowiadającego $Y=Y^{(1)}$.

$$Y = \Phi(W^T Y) = \Phi\left(\begin{pmatrix} -1 & -1 & 3 \\ -1 & 3 & -1 \\ 3 & -1 & -1 \end{pmatrix} \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}\right) = \Phi\left(\begin{bmatrix} 5 \\ -3 \\ -3 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$$

Siec zachowała się poprawnie i odtworzyła na wyjściu X sygnał odpowiadający $X = X^{(1)}$.

Podobnie można sprawdzić dla wektora $X^{(2)}$

$$Y = \Phi(W^T Y) = \Phi\left(\begin{pmatrix} -1 & -1 & 3 \\ -1 & 3 & -1 \\ 3 & -1 & -1 \end{pmatrix} \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}\right) = \Phi\left(\begin{bmatrix} -3 \\ 5 \\ -3 \end{bmatrix}\right) = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}$$

Dyskutowane przykłady odpowiadały sytuacji kiedy podany do sieci wektor X lub Y nie odpowiada idealnie wektorowi prezentowanemu w trakcie procesu uczenia.

Działanie sieci BAM przy braku zgodności ze wzorcem

W sytuacji, kiedy podany do sieci *BAM* sygnał X albo Y nie jest idealnie zgodny z żadnym wzorcem, *sieć poszukuje poprawnej odpowiedzi w sposób iteracyjny*.

W wyniku tego pojawia się pewien (dłuższy lub krótszy) proces przejściowy i po pewnym czasie *osiągany jest stan równowagi*, odpowiadający - jak zawsze przy sieciach zawierających sprzężenia zwrotne – minimum funkcji "energii" sieci.

Działanie sieci BAM przy braku zgodności ze wzorcem

Wartość *funkcji "energii"* związana z pojawieniem się w sieci BAM sygnałów X i Y wyraża się wzorem

$$E(X,Y) = -X^T W Y$$

Ponieważ

$$W = \Sigma \, \Delta W^{(k)}$$

Przeto

$$E(X,Y) = \Sigma \, E^{(k)} (X,Y) \quad \text{gdzie} \quad E^{(k)} (X,Y) = -X^T \Delta W^{(k)} Y$$

co uwzględniając fakt, że

$$\Delta W^{(k)} = X^{(k)} Y^{(k)T}$$

można przepisać w postaci

$$E^{(k)}(X,Y) = -X^T (X^{(k)} Y^{(k)T}) Y = - (X^T X^{(k)}) (Y^{(k)T} Y)$$

Działanie sieci BAM przy braku zgodności ze wzorcem

Łatwo zauważyć, że składnik $E^{(k)}(X,Y)$ ma tym mniejszą wartość, im większe wartości mają iloczyny $(X^T X^{(k)})$ oraz $(Y^{(k)T} Y)$, te zaś osiągają maksimum przy $X=X^{(k)}$ oraz $Y=Y^{(k)}$.

Jeśli jednak na wejściu sieci pojawi się sygnał $X \neq X^{(k)}$ dla wszystkich k, to wówczas w sieci rozpocznie się proces dynamiczny opisywany w kolejnych chwilach czasowych τ równaniami iteracyjnymi

$$Y(\tau+1) = \Phi(W X(\tau))$$
$$X(\tau+1) = \Phi(W Y(\tau+1))$$

Przyjmując $X(0) = X$ otrzymujemy kolejno $Y(1), X(1), Y(2), X(2), \ldots, Y(i), X(i), Y(i+1), X(i+1)$.

Działanie sieci BAM przy braku zgodności ze wzorcem

Rozważmy, jak się przy tym zmienia *"energia" sieci*. Przejście energii od wartości

$$E(i) \quad = - X(i)^T W Y(i)$$

do wartości

$$E(i+1) = - X(i)^T W Y(i+1)$$

musi zmieniać łączną energie, co łatwo uzasadnić faktem, że składowe wektorów X i Y są bipolarne, przeto zmiana wartości z E(i) na E(i+1) może zajść wyłącznie w wyniku *zmiany znaku* niektórych składowych wektora Y(i+1) w stosunku do Y(i).

Działanie sieci BAM przy braku zgodności ze wzorcem

Z równania opisującego sieć

$$Y(i+1) = \Phi\ (W\ X(i))$$

wynika, że zmiany te mogły nastąpić wyłącznie zgodnie ze znakami odpowiednich składowych wektora $X(i)^T W$, co dowodzi, że energia sieci mogła w wyniku tych zmian jedynie zmaleć.

Podobny argument można odnieść także do drugiego kroku iteracji.

$$X(i+1) = \Phi\ (W\ Y(i+1))$$

Z tego wynika, że sieć błądząc od jednego stanu $\langle X(i),Y(i)\rangle$ do kolejnego stanu $\langle X(i+1),Y(i+1)\rangle$ porusza się *zawsze w kierunku wynikającym z malejącej energii.*

Działanie sieci BAM przy braku zgodności ze wzorcem

Taki proces kończy się znalezieniem *lokalnego minimum.* Na ogół sieć znajduje najbliższe asocjacje, czyli najbliższej w sensie euklidesowym parze (X,Y) w stosunku do startowego punktu. Ale nie jest to 100% pewne.

W sieci *BAM*, jak w każdym systemie ze sprzężeniem zwrotnym, mogą pojawić się oscylacje. Jak wykazał Kosko (1987), wszystkie sieci BAM są *bezwarunkowo stabilne* bez względu na to, jaka jest macierz wag W.

Ta ważna własność wynika z faktu wykorzystania w strukturze BAM tej samej macierzy wag do połączenia pierwszej warstwy z drugą i drugiej z pierwszą.

Sieć BAM staje się siecią Hopfielda gdy macierz W jest kwadratowa i symetryczna.

Pojemność pamięci sieci BAM

Sieć BAM jest pamięcią, więc jej działanie polega na rejestrowaniu informacji.

Pytanie więc o pojemność tej pamięci czyli liczbę informacji L, jaka może być w niej zapamiętana.

W literaturze pojawiają się różne oszacowania. Kosko (1987) przyjmował że pojemność sięga liczby neuronów w mniejszej z dwóch warstw sieci

$$L = \min \{n,m\}$$

Ten wynik można osiągnąć tylko przy bardzo specjalnym kodowaniu informacji.

Pojemność pamięci sieci BAM

Bardziej realistyczne jest oszacowanie (McElievce, Posner, Rodemich, Vaukatesh)

$$L = n/(4 \log_2 n)$$

gdzie n jest liczba neuronów w mniejszej warstwie.
Warto zauważyć jak bardzo różnią się te oszacowania. Dla $n=1024$, pierwsza estymacja to $L=1000$, druga to $L=25$!

Możliwość zwiększenia pojemności pamięci to indywidualne dobieranie progów dla każdego z neuronów.

Odmiany sieci BAM

Oprócz omówionych już sieci BAM w formie cyfrowej (x_i, y_i = 1 lub –1) rozważane są sieci tego typu o elementach analogowych, z funkcją ϕ opisaną np. sigmoidą logistyczną. Sieci takie okazują się bardzo przydatne w adaptacyjnym przetwarzaniu sygnałów.

Podobnie rozważane są sieci o działaniu ciągłym (nasza dyskusja ograniczyła się do dyskretnej skali czasu i angażowała pewne procesy iteracyjne w sieci).
Inną *odmianą sieci BAM* jest *sieć adaptacyjna*. W sieci takiej dokonuje się permanentna powolna zmiana współczynników wagowych, zgodnie ze wzorem

$$\boxed{\omega_{ij} = \omega_{ij} + \eta\, x_i\, y_j}$$

Sieć taka może doskonalić swoje działanie w trakcie eksploatacji i nie wymaga oddzielnego procesu uczenia.

Odmiany sieci BAM

Jeszcze inna odmiana sieci związana jest z wprowadzeniem do niej elementu rywalizacji (tylko jeden element w każdej warstwie neuronów ma sygnał wyjściowy wynoszący +1, pozostałe mają wymuszony sygnał –1).

Sieć tego typu może służyć do kojarzenia specjalnych typów wektorów binarnych, na przykład w diagnostyce medycznej.

Istnieje *wiele różnych odmian* sieci BAM. Jest ona szczególnie *chętnie badana i analizowana przez inżynierów*, ponieważ jej działanie szczególnie łatwo daje się powiązać z konkretnymi zadaniami (na przykład z zastosowania systemów ekspertowych).

Zestaw pytań do testu

1. Czym się różni pamięć skojarzeniowa od pamięci adresowanej?
2. Co to jest miara Hamminga?
3. Wymień rodzaj warstw w sieci Hamminga.
4. Na czym polega sieć Hintona?
5. Co to jest sieć BAM?