Wstęp do inteligencji komputerowej – zajęcia nr 8

Jarosław Stańczak
WSISiZ

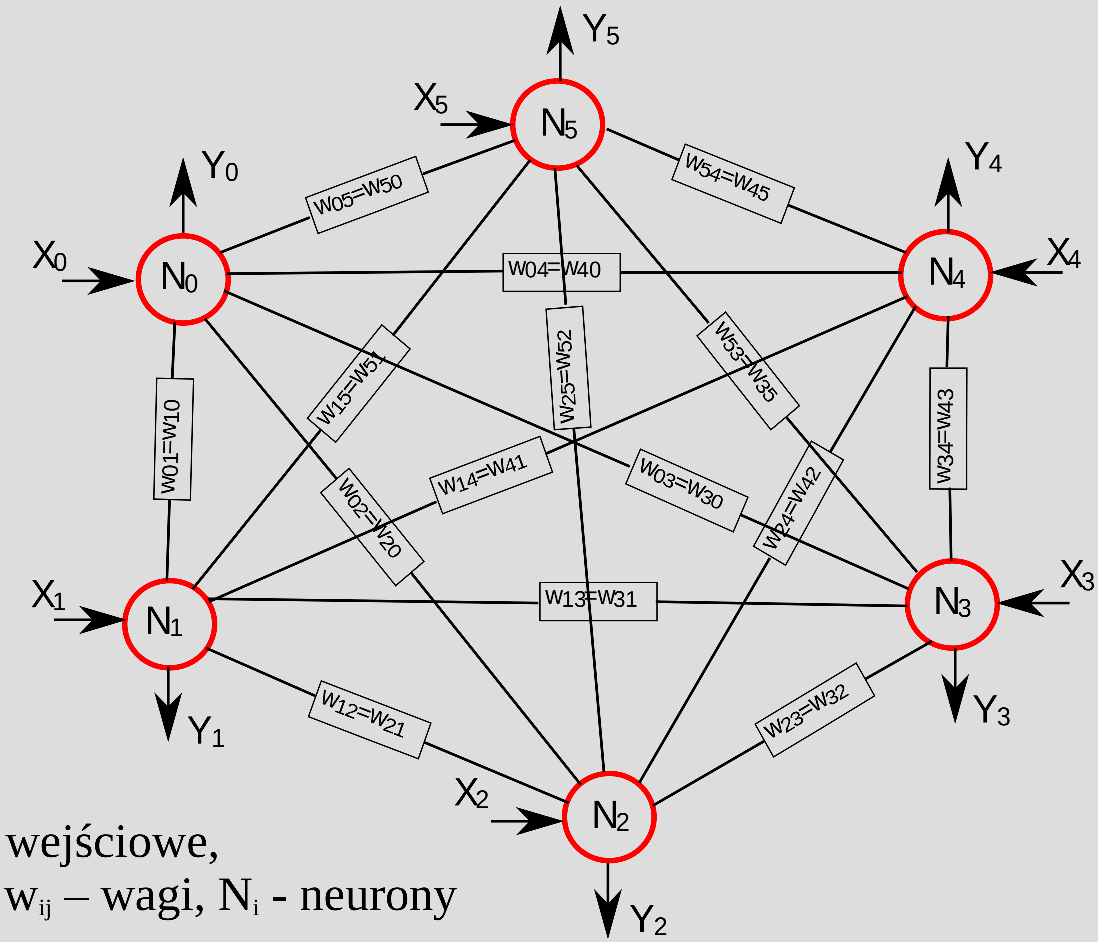
Sztuczne sieci neuronowe cz. 2:

- sieci ze sprzężeniem zwrotnym - pamięć w sieci neuronowej
- sieci samoorganizujące się
- zastosowania omawianych typów sieci neuronowych.

Sieci ze sprzężeniem zwrotnym –
rekurencyjne sieci neuronowe

Sieci ze sprzężeniem zwrotnym to drugi po sieciach jednokierunkowych model sieci, opracowany jeszcze przed ponownym wzrostem zainteresowania nimi w latach 90 XX w.
Pierwsza taka sieć została wymyślona przez J. Hopfielda w 1982 r. W zasadzie można ją rozpatrywać jako sieć jednowarstwową, w której wyjście z każdego neuronu (dostępne także jako sygnał wyjściowy z sieci) jest podawane na wejścia wszystkich innych neuronów oprócz swojego wejścia ($w_{ii}=0$). Wagi w tak połączonej sieci są symetryczne ($w_{ij}=w_{ji}$).
Oczywiście można sobie też wyobrazić wariant niesymetryczny, lecz nie jest on stosowany, zapewne z uwagi na brak lub małą liczbę stanów stabilnych sieci i trudności z analizą działania takiego układu.
Budowę sieci Hopfielda zilustrowano na następnym slajdzie.

Sieci ze sprzężeniem zwrotnym –
sieć Hopfielda



X_i – zewnętrzne sygnały wejściowe,
Y_i – sygnały wyjściowe, w_{ij} – wagi, N_i - neurony

Sieci ze sprzężeniem zwrotnym –
działanie sieci Hopfielda

Stan neuronu w sieci Hopfielda wyraża się wzorem:

$$s_i(k+1)=\sum_{j=1}^{n}\left(w_{ij}*s_j(k)-\Theta_i+x_i(k)\right) \qquad (1)$$

a sygnał wyjściowy:

$$Y_i(k)=\begin{cases}-1\,(\text{czasem również } 0) \text{ dla } s_i(k)<0\\ Y_i(k-1) \text{ dla } s_i(k)=0\\ 1 \text{ dla } s_i(k)>0\end{cases} \qquad (2)$$

Y_i – sygnał wyjściowy, Θ_i – próg zadziałania neuronu (często traktowany jako dodatkowa waga), w_{ij} – waga, $s_i(k)$ – stan neuronu w chwili k, $x_i(k)$ – sygnał wejściowy do neuronu.

Sieci ze sprzężeniem zwrotnym –
działanie sieci Hopfielda

Sieć neuronowa ze sprzężeniem zwrotnym staje się układem dynamicznym (we wzorze (1) widać zależność stanu sieci od chwil czasu - *k*), czyli układem obdarzonym „pamięcią” swojego stanu z chwil poprzednich. Stany następne zależą między innymi od tej zapamiętanej „historii” układu.
Stan układu może się zmienić także na skutek działania sygnałów zewnętrznych (wejścia X oraz wejść od innych neuronów).
Wpływ na stan sieci mają też wagi, decydują one o powstaniu stanów stabilnych sieci, do których sieć dąży po podaniu jej na wejścia pewnych wartości.

Sieci ze sprzężeniem zwrotnym –
działanie sieci Hopfielda

Właściwości takiej sieci najczęściej opisuje się używając funkcji analogicznej do energii układu dynamicznego, zwanej także funkcją Lapunowa:

$$E(k) = -\frac{1}{2} * \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} * s_i(k) * s_j(k) + \sum_{i=1}^{n} \Theta_i * s_i(k) \qquad (3)$$

$s_i(k)$ – stan neuronu, Θ_i - próg zadziałania neuronu, n – liczba neuronów

Minima takiej funkcji można traktować jako zapamiętane przez sieć wzorce, o ile zapiszemy je przez odpowiedni dobór wag.
W związku z tym Sieć Hopfielda można potraktować jak pamięć, tzw. pamięć skojarzeniową lub asocjacyjną, adresowaną zawartością (a dokładnie „fragmentami” tej zawartości).
Jest bardzo prawdopodobne, że jeden z mechanizmów działania pamięci w mózgu jest zbliżony do mechanizmu działania sieci Hopfielda.

Sieci ze sprzężeniem zwrotnym –

działanie sieci Hopfielda

Sieć Hopfielda może działać:

- synchronicznie, wtedy jest siecią z zegarem, jak typowy cyfrowy układ synchroniczny (np. typowy komputer) i wszystkie neurony jednocześnie zmieniają swój stan w momentach wyznaczanych przez zegar,
- asynchronicznie, wtedy tylko jedne neuron (np. wylosowany) może zmienić swój stan, możliwy jest też tryb działania, w którym cała sieć dochodzi do stanu równowagi (ustalonego) po pewnym czasie zależnym od parametrów użytych lub zaprogramowanych jednostek.

Sieci ze sprzężeniem zwrotnym –
działanie sieci Hopfielda

Osiągnięcie stanu równowagi pod wpływem pewnych wejść do układu jest równoznaczne z odnalezieniem zapisanej w sieci informacji.
Po podaniu na wejścia sieci (z uprzednio zapisanymi wzorcami) pewnego sygnału, sieć dochodzi w kilku krokach do stanu równowagi „energetycznej” odnajdując jednocześnie wzorzec najbardziej pasujący do zadanego sygnału. Sygnał wejściowy może być niekompletny, zaszumiony, przekłamany (oczywiście w pewnych granicach), a sieć i tak odnajduje właściwy wzorzec. Ta właściwość powoduje, że sieć taka jest przykładem pamięci asocjacyjnej.
Niestety wpisanie do niej „planowanych” danych powoduje też wpisanie dodatkowych, fałszywych minimów, czyli niepotrzebnych danych, co ogranicza jej pojemność. Istnieją metody „zasypywania” takich niepotrzebnych minimów.

Sieci ze sprzężeniem zwrotnym –
uczenie sieci Hopfielda metodą Hebba

Sieć Hopfielda można uczyć metodą wzorowaną na regule Hebba:

$$w_{ij} = \frac{1}{n} \sum_{m=1}^{p} x_{im} * x_{jm}$$

w_{ij} – waga między neuronami i oraz j, x_{im} – wzorzec m dla i-tego neuronu, p – liczba wzorców, n – liczba neuronów.

Jak widać przy użyciu reguły Hebba wzorce można zapisać „na raz” bez długotrwałego uczenia.

Sieci ze sprzężeniem zwrotnym –
uczenie sieci Hopfielda metodą Storkey'a

Uczenie sieci przyrostową regułą Storkey'a:

$$\Delta w_{ij}(k)=\frac{1}{n}x_{im}(k)*x_{jm}(k)-\frac{1}{n}x_{im}(k)*h_{jm}(k)-\frac{1}{n}x_{jm}(k)*h_{im}(k)+\frac{1}{n}*h_{im}*h_{jm}$$

$$h_{im}(k)=\sum_{l=1}^{n}w_{il}(k)*x_{lm}(k) \qquad h_{jm}(k)=\sum_{l=1}^{n}w_{jl}(k)*x_{lm}(k)$$

$$w_{ij}(k+1)=w_{ij}(k)+\Delta w_{ij}(k)$$

$w_{ij}(k)$, $\Delta w_{ij}(k)$ – waga (zmiana wagi) między neuronami *i*, *j*, x_{im} – wzorzec *m* dla *i*-tego neuronu, *p* – liczba wzorców, *n* – liczba neuronów, *k* – chwila czasowa.

Reguła Storkey'a umożliwia nauczenie sieci większej liczby wzorców niż reguła Hebba. Ma też lepszą zdolność do generalizacji, czyli poprawnego interpretowania zniekształconych danych.

Sieć Hopfielda -
pojemność pamięci

Sieć Hopfielda użyta jako pamięć asocjacyjna ma pojemność

$$p_{max} = n / (2 \log n) \leq 0{,}138 n$$

n – liczba neuronów.

Niestety nie jest ona imponująca, prowadzone są prace nad uzyskaniem wariantów sieci o większej pojemności pamięci.

Sieć Hopfielda -
zastosowania

Sieć Hopfielda jest używana:

- jako pamięć asocjacyjna (adresowana zawartością);
- jako optymalizator w problemach optymalizacji kombinatorycznej (zadania klasy NP typu problem plecakowy, problem komiwojażera...);
- w przetwarzaniu obrazów;

Sieć Hopfielda -

przykład zastosowania do rozwiązania problemu TSP

Ponieważ stan sieci Hopfielda opisuje się funkcją analogiczną do energii (3), a sama sieć ma właściwość samorzutnej minimalizacji tej energii, to w takim razie można spróbować „zakodować” w tej funkcji energii jakiś problem minimalizacji (a po niewielkich modyfikacjach i maksymalizacji) i odpowiednio dobrać wagi tak, aby wykorzystać tę naturalną właściwość sieci do rozwiązania tego typu problemów.

Sieć Hopfielda -
przykład zastosowania do rozwiązania problemu TSP

Wobec tego należy zbudować sieć o wielkości N^2 neuronów (gdzie N to liczba miast w rozwiązywanym problemie) składających się z N sekcji, w których tylko jeden neuron może mieć wartość jeden, a numer sekcji będzie określać, jako które z kolei miasto reprezentowane przez neuron o wartości 1 będzie odwiedzone (kod 1 z N). Pamiętając, że wyjścia z sieci są binarne, to można łatwo zastosować takie zakodowanie problemu.

Miasto\Kolejność	1	2	3	4
Miasto 1	0	0	1	0
Miasto 2	0	1	0	0
Miasto 3	0	0	0	1
Miasto 4	1	0	0	0

Rozwiązanie zakodowane w tabeli reprezentuje drogę: Miasto 4 – Miasto 2 – Miasto 1 – Miasto 3 – Miasto 4.

Sieć Hopfielda -
przykład zastosowania do rozwiązania problemu TSP

Do zbudowania sieci potrzebna jest funkcja celu, zawierająca również człony odpowiadające karze za ewentualne naruszenie ograniczeń: odwiedzenie wszystkich miast (C/2*...), tylko jeden neuron o wartości 1 w „wierszu” i „kolumnie” (reprezentacja jak w tablicy na poprzednim slajdzie)), czyli tylko 1 miasto odwiedzone w każdym kroku sieci (B/2*....) bez powtórzeń (A/2*...), człon (D/2*...) to minimalizowana odległość.

$$E(x)=\frac{A}{2}*\sum_{p=1}^{N}\sum_{j=1}^{N}\sum_{i=1}^{N}x_{pi}*x_{pj}+\frac{B}{2}*\sum_{i=1}^{N}\sum_{p=1}^{N}\sum_{q=1}^{N}x_{pi}*x_{qi}+\frac{C}{2}*\left(\sum_{p=1}^{N}\sum_{i=1}^{N}x_{pi}-N\right)^{2}+\frac{D}{2}*\sum_{p=1}^{N}\sum_{q=1}^{N}\sum_{i=1}^{N}d_{pq}*x_{pi}*\left(x_{q,i+1}+x_{q,i-1}\right)$$

x_{pi} – neuron reprezentujący miasto p na pozycji i trasy (może mieć wartość 0 lub 1); A, B, C, D – pewne dodatnie stałe, d_{pq} – odległość między miastami p i q, N – liczba miast i jednocześnie liczba neuronów w „wierszu” i „kolumnie” sieci.

Sieć Hopfielda -
przykład zastosowania do rozwiązania problemu TSP

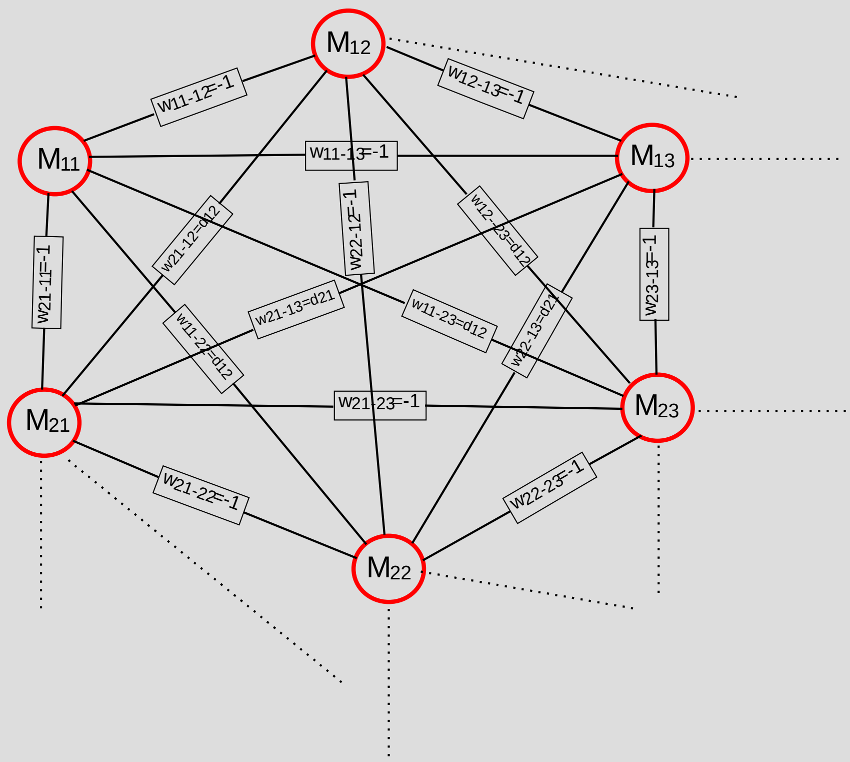
Wagi w sieci są symetryczne i są dobierane następująco:

- wagi między neuronami z kolejnych „kolumn” poza będącymi na tej samej pozycji są równe odległościom między miastami d_{pq};
- wagi w obrębie tej samej kolumny lub między tymi samymi miastami w różnych kolumnach są niewielkimi liczbami ujemnymi (ujemne sprzężenie zwrotne, blokujące multiplikowanie 1 na tych pozycjach).

W podobny sposób można rozwiązywać także inne problemy optymalizacyjne klasy NP.

Sieć Hopfielda -

przykład zastosowania do rozwiązania problemu TSP

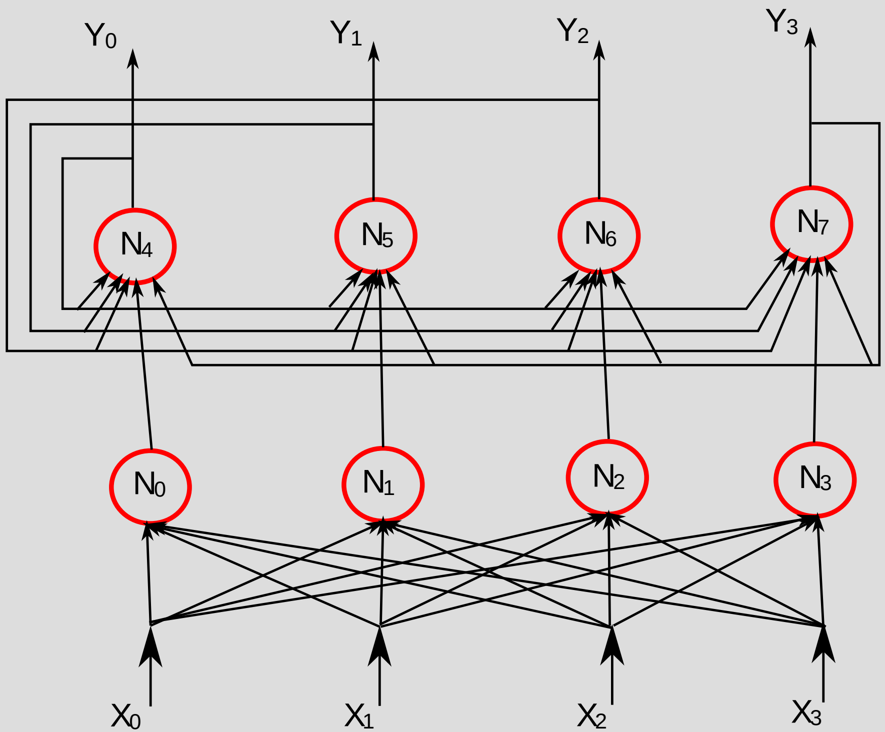


Fragment symetrycznej sieci Hopfielda służącej do rozwiązywania problemu TSP

Sieć Hamminga

Sieć Hamminga jest rozszerzeniem sieci Hopfielda. Rozszerzenie polega na dodaniu do sieci Hopfielda jednej warstwy (typu perceptronowego) realizującej obliczanie odległości Hamminga (jest to np. liczba bitów, którą różnią się dwa wzorce, sieć działa na sygnałach binarnych) podanego sygnału od najbliższego wzorca, co ma zwiększyć pojemność pamięciową całej sieci. W praktyce wygląda to tak, że po podaniu sygnału wejściowego powinno aktywować się tylko jedno wyjście z warstwy wejściowej, co pomaga odróżnić wzorce, a ich pamiętaniem zajmuje się warstwa wyjściowa o strukturze zbliżonej do sieci Hopfielda.
W niektórych wariantach dodaje się jeszcze trzecią warstwę - wyjsciową, która ma za zadanie nauczyć się odpowiadać nie tylko numerem/rodzajem klasy, ale też jej charakterystycznym przedstawicielem.
Ilustrują to rysunki pokazane na następnych dwóch slajdach.

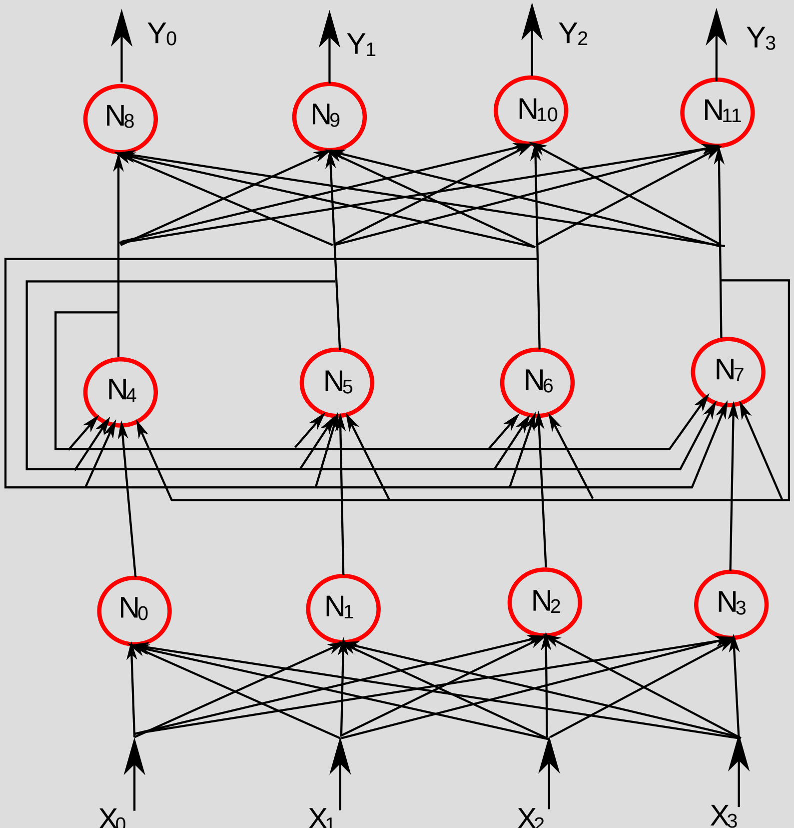
Sieć Hamminga
wersja 1



X_i – zewnętrzne sygnały wejściowe, Y_i – sygnały wyjściowe, N_0 -N_3 – neurony wyznaczające odległość Hamminga między wejściem, a zapamiętanymi wzorcami, N_4 -N_7 – neurony „pamięciowe” ze sprzężeniem zwrotnym, działające analogicznie jak w sieci Hopfielda

Sieć Hamminga

wersja 2



X_i – zewnętrzne sygnały wejściowe, Y_i – sygnały wyjściowe, N_0 -N_3 – neurony wyznaczające odległość Hamminga między wejściem, a zapamiętanymi wzorcami, N_4 -N_7 – neurony „pamięciowe” ze sprzężeniem zwrotnym, działające analogicznie jak w sieci Hopfielda, N_8 -N_{11} – neurony odtwarzające konkretne wzorce

Sieć Hamminga

Sieć Hamminga może zapamiętać tyle wzorców, ile jest neuronów w warstwie „pamięciowej” (rekurencyjnej). Jest to pewien postęp w stosunku do sieci Hopfielda, ale okupiony dołożeniem kolejnych warstw, przez co cała sieć, zależnie od wersji, ma dwa lub trzy razy tyle neuronów co w warstwie rekurencyjnej.

Sieć Hamminga
ustalanie wag

Poniższe opisy są ważne dla wersji dwuwarstwowej sieci Hamminga.

Wagi w warstwie wejściowej są ustawiane według wzorów:

$$w_{ij}=x_i^j \text{ i } \Theta_j=N/2$$

Wartość wyjścia neuronów warstwy wejściowej oblicza się na podstawie znanego wzoru opisującego neuron McCullocha-Pittsa:

$$U_j=f\left(\sum_{i=0}^{N-1} w_{ij}*x_i-\Theta_j\right)$$

w_{ij} – waga od wejścia *i* do *j*-tego neuronu warstwy wejściowej, x_i^j – *i*-ty bit *j*-tego wzorca wejściowego (każdy neuron warstwy wejściowej ma reagować na jeden wybrany wzorzec, dlatego indeks *j* odnosi się do neuronów i wzorców), Θ_j – próg aktywacji neuronów warstwy wejściowej; *N* – liczba wejść do sieci, *f*(...) - funkcja aktywacji, najczęściej skokowa .

Sieć Hamminga
ustalanie wag

Wagi neuronów od warstwy wejściowej do rekurencyjnej ustawiane są na wartość 1 (sygnał bez zmian). W warstwie rekurencyjnej dobór wag jest dość prosty, określa go poniższy wzór (jak widać inaczej niż w sieci Hopfielda istnieje tu sprzężenie zwrotne wejścia z wyjściem tego samego neuronu rekurencyjnego i hamowanie innych neuronów):

$$w_{kl} = \begin{cases} 1 \; dla \; k = l \\ -\varepsilon \; dla \; k \neq l, \varepsilon < \dfrac{1}{M} \end{cases}$$

Neurony w warstwie rekurencyjnej nie mają progu aktywacji

$$Y_k = F\left(\sum_{l=1}^{M} w_{kl} * U_l\right)$$

w_{kl} – waga między neuronami k i l w warstwie wyjściowej, M – liczba neuronów w warstwie rekurencyjnej, U_l – wejścia z warstwy wejściowej, $F(...)$ - funkcja aktywacji neuronów warstwy rekurencyjnej.

Sieć Hamminga
ustalanie wag

W wariancie sieci trójwarstwowej warstwa wyjściowa jest odpowiednikiem perceptronu z liniowymi wyjściami i może być uczona podobnie jak perceptron z sygnałem wejściowym: jedynka na odpowiednim wyjściu warstwy poprzedniej i odpowiadające jej wyjście.

Sieć Hamminga

działanie sieci

Sygnał wejściowy z warstwy wejściowej doprowadzany jest do sieci rekurencyjnej tylko na jedną iterację, następnie jest odłączany, a sieć iteruje aż znajdzie się w stanie równowagi, a jej wyjścia podadzą wynik, czyli sieć zadecyduje, do czego najbardziej podobny jest podany sygnał.

Ewentualna warstwa wyjściowa umożliwia przetworzenie sygnału charakterystycznego dla rozpoznanej klasy danych na konkretny uśredniony wzorzec danej klasy.

Sieć Hamminga - zastosowanie

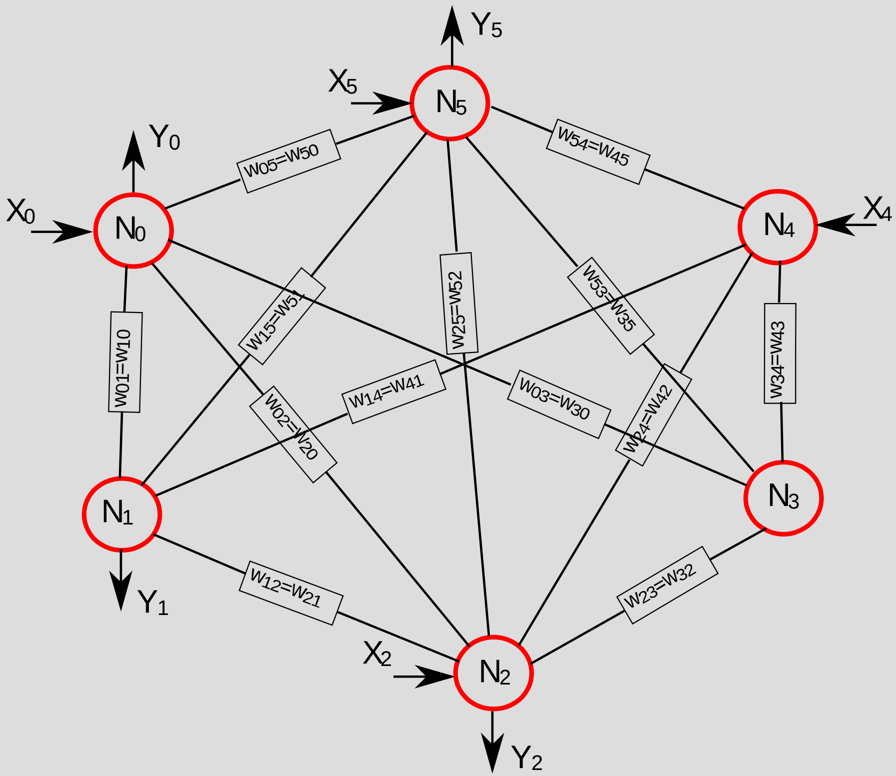
Sieć Hamminga jest stosowana jako:

- klasyfikator (w literaturze podkreśla się bardzo dobre jej właściwości w tej dziedzinie – może dobrze sklasyfikować wzorce przekłamane nawet o 43%)
- pamięć asocjacyjna o lepszej pojemności niż sieć Hopfielda
- maszyna wnioskująca i baza wiedzy w systemach ekspertowych.

Maszyna (sieć) Boltzmanna

Maszyna (lub sieć) Boltzmanna to kolejne uogólnienie sieci Hopfielda. Jest to sieć rekurencyjna ze sprzężeniami zwrotnymi o symetrycznych wagach, która zawiera neurony ukryte. Jednocześnie sieć taka jest stochastyczna, gdyż neurony posiadają prawdopodobieństwa osiągnięcia pewnego stanu określone rozkładem Boltzmanna.
Architektura takiej sieci jest podobna do sieci Hopfielda – każdy neuron połączony jest z każdym (mogą istnieć bezpośrednie połączenia z wyjścia na wejście tego samego neuronu), choć oczywiście niektóre wagi mogą być zerowe (brak połączenia), jednak nie do wszystkich neuronów dochodzą sygnały wejściowe, nie wszystkie widoczne są też na wyjściu. Te, które nie są widoczne ani od strony wejścia, ani od strony wyjścia, są neuronami ukrytymi. Ilustruje to rysunek na następnym slajdzie.

Maszyna Boltzmanna



W sieci Boltzmanna mogą występować neurony ukryte, na rysunku jest to N_3.

Maszyna Boltzmanna

Neurony w tej sieci mogą przyjmować stan s_i={-1, 1} zgodnie z rozkładem prawdopodobieństwa $p(h_i)$:

$$p(h_i)=\frac{1}{1+\exp(-2*\beta*h_i)}=\frac{1}{1+\exp\left(-2*\frac{h_i}{T}\right)}$$

$$h_i=\sum_{j=0}^{n} w_{ij}s_j$$

w_{ij} – waga między neuronami i i j , n – liczba neuronów, s_j – stan neuronu j, T – parametr rozkładu będący analogią do temperatury (można go modyfikować w trakcie działania sieci, zmieniając nieco jej działanie).

Maszyna Boltzmanna

Rozkład prawdopodobieństwa stanów dla całej sieci opisywany jest rozkładem Boltzmanna, stąd nazwa sieci, a jej właściwości również opisuje się funkcją energetyczną o postaci zbliżonej do tej, która opisuje sieć Hopfielda, jednak należy pamiętać, że w tym przypadku jest to funkcja probabilistyczna.
Regułę uczenia sieci wyprowadza się różniczkując funkcję energii (która pełni tu rolę kryterium jakości działania sieci) po wagach analogicznie jak dla sieci BP różniczkuje się błąd klasyfikacji po wagach i otrzymuje się stochastyczny wariant metody BP (funkcja energii np. (3) jest dość podobna do minimalizowanej w metodzie BP funkcji sumy kwadratów błędów), którego wzorów, ze względu na ich skomplikowanie, nie będę tu przytaczał.
Dla zainteresowanych można znaleźć je np. w pracy:
J. Hertz, A. Krogh, R. Palmer „Wstęp do teorii obliczeń neuronowych”, WNT, 1993.

Maszyna Boltzmanna -
właściwości i zastosowania

Generalnie maszyna Boltzmanna jest siecią bardzo skomplikowaną i trudną w realizacji, uczeniu i wykorzystaniu, doczekała się niewielu realizacji praktycznych.
Jej zastosowania są podobne do sieci Hopfielda:

- pamięć asocjacyjna
- klasyfikator danych.

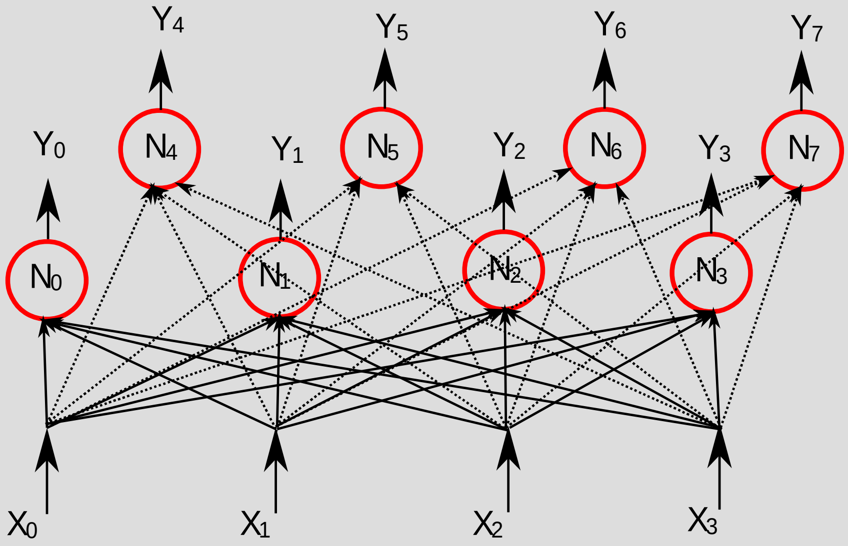
Sieć Kohonena

Sieć Kohonena jest przykładem sieci samoorganizującej się (zwanej także mapą samoorganizującą się), uczącej się bez nadzoru. Jej zadaniem jest mapowanie danych wielowymiarowych (wymiarowość danych, to w uproszczeniu liczba wejść do sieci) do przestrzeni o wymiarowości niższej, zazwyczaj dwuwymiarowej. Sieć grupuje przykłady jej prezentowane. W fazie uczenia generuje nowe zgrupowania danych (dlatego dane trenujące muszą być reprezentatywne dla problemu), a w fazie działania przyporządkowuje prezentowane dane do którejś z wytworzonych w czasie uczenia grup. Sieć wykrywa podobieństwa w prezentowanych danych do wytworzonych samodzielnie w trakcie uczenia wzorców i przydziela je do odpowiednich grup tak, aby jak najlepiej pasowały do wytworzonych wzorców. Odpowiedzią sieci jest „odpalenie” jednego neuronu, który reprezentuje wzorzec najbliższy prezentowanego. Sieć działa według zasady WTA - „Winner Takes All” - „zwycięzca bierze wszystko” (czasem zamienianej na MTA - „prawie wszystko”).
Sieć Kohonena jest przedstawicielem tzw. sieci komórkowych, w których istotne są połączenia lub interakcje między neuronami w określonym sąsiedztwie.

Sieć Kohonena - architektura

Sieć Kohonena jest siecią jednokierunkową, składa się z dwóch warstw:

- warstwy wejściowej, dystrybuującej sygnały wejściowe do wszystkich neuronów
- warstwy wyjściowej – topologicznej, neurony są połączone tylko z warstwą wejściową (istnieją zbliżone sieci, w których realizowana jest idea hamowania obocznego, realizująca model WTA, tu jednak zwycięzcę wykrywa „siła wyższa").



Sieć Kohonena - działanie

W sieci Kohonena wyjście neuronu oblicza się według tradycyjnego wzoru:

$$h_i = \sum_{j=0}^{n} w_{ij} x_j$$

w_{ij} – waga między neuronem *i* i wejściem *j* , x_j – sygnał na wejściu *j*
jednakże na wyjście przekazywana jest wartość 1 tylko dla najlepszego neuronu (ten najlepszy wybiera układ sterujący siecią – owa „siła wyższa”). Ponieważ najczęściej siec neuronowa jest realizowana przy użyciu standardowego komputera, to możliwe jest obliczanie różnicy między wejściem, a wektorem wag bez ubierania tych obliczeń w konkretny model neuronu.

Sieć Kohonena - uczenie

Uczenie sieci Kohonena nie jest skomplikowane:

$$\Delta w_{ij}(k+1)=\begin{cases}\alpha(k)*(x_j-w_{ij}(k)) \ dla \ i\in\Omega \\ 0 \ dla \ i\notin\Omega\end{cases}$$

w_{ij} – waga między neuronem i i wejściem j, x_j – sygnał na wejściu j, Ω - otoczenie zwycięskiego neuronu (z nim samym włącznie), $\alpha(k)$ – współczynnik uczenia, malejący w funkcji czasu.

Otoczenie zwycięskiego neuronu bywa definiowane rozmaicie, najczęściej jest to funkcja tzw. meksykańskiego kapelusza o postaci:

$$\zeta_i(k)=\begin{cases}1 \ dla \ r_i(k)=0 \\ \dfrac{\sin(a*r_i(k))}{a*r_i(k)} \ dla \ |r_i(k)|\in(0,2\pi/a) \\ 0 \ dla \ |r_i(k)|>2\pi/a\end{cases}$$

r_{il} - oznacza odległość między neuronem i a zwycięzcą (jest zależna od czasu), a – parametr regulujący zasięg funkcji.

W takim przypadku wzór na modyfikację wag można uprościć do:

$$\Delta w_{ij}(k+1)=\alpha(k)*\zeta_i(k)*(x_j-w_{ij}(k))$$

Sieć Kohonena -
zastosowania

Sieć Kohonena używana jest do:

- klasyfikacji danych
- odwzorowywanie przestrzeni danych
- klasteryzacji danych.

Niestety działanie sieci, a szczególnie jej uczenie nie należy do szybkich.

Sieć ART

Sieć ART nazywana jest tak od pojęcia adaptacyjnej teorii rezonansu (*ang. Adaptive Resonance Theory*) i jest kolejnym przykładem sieci uczącej się bez nauczyciela. Sieć klasyfikuje dochodzące do niej przykłady, zgodnie z nauczonymi wcześniej samodzielnie klasami, lecz jeśli napotka nowy przykład, który jej zdaniem nie jest podobny do wyuczonych wcześniej (decyduje o tym tzw. próg czujności sieci τ), to zapamiętuje go jako wzorzec nowej klasy danych.

Sieć ART

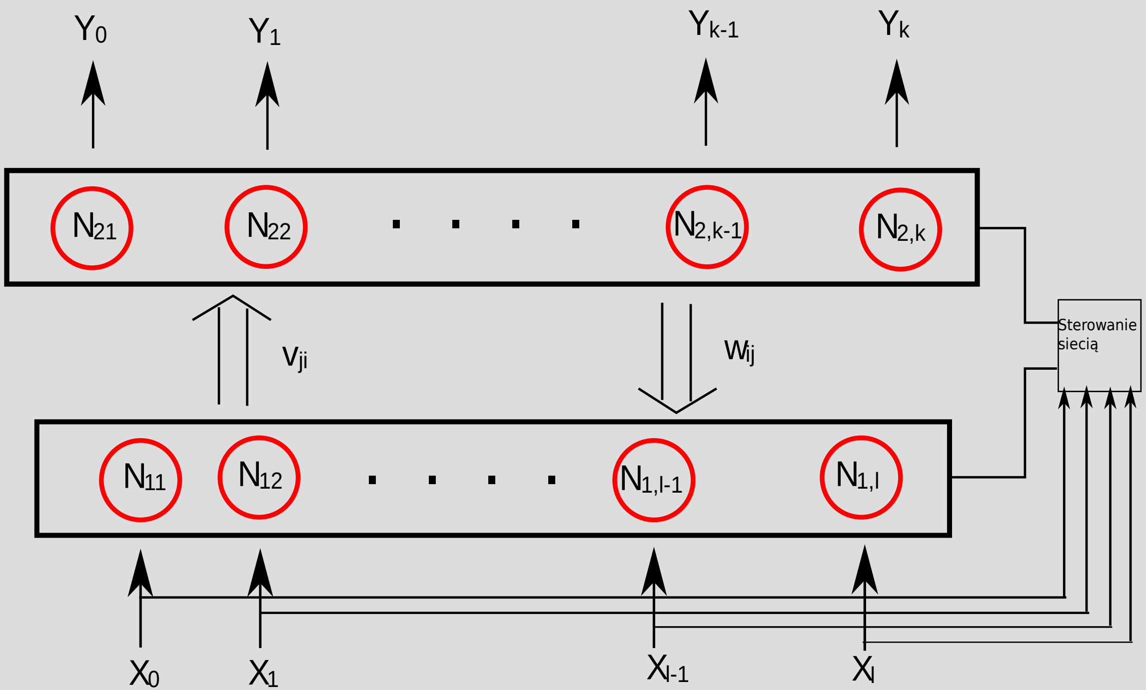
Sieć ART jest przykładem sieci dwuwarstwowej, które to warstwy przez wzajemne oddziaływanie decydują o przydziale nowej danej do jednej z ze znanych klas lub utworzeniu nowej kategorii danych. Liczby neuronów w obu warstwach są dość dobrze określone przez przetwarzane dane. Warstwa wejściowa ma tyle neuronów ile jest sygnałów wejściowych, np dla obrazu o rozdzielczości *n* x *m* będzie to *n* x *m* neuronów. W warstwie drugiej neuronów powinno być nieco więcej niż spodziewanych kategorii danych. Sieć jest cały czas w fazie uczenia i działania jednocześnie, musi wobec tego posiadać cały czas nadmiarowe neurony do wykorzystania na ewentualne nowe klasy danych.

Modyfikując wartość parametru czuwania można nieco modyfikować sposób działania sieci on-line, zmniejszając lub powiększając zdolność do tworzenia nowych kategorii danych.

Sieć ART1, ART2, ...

Powstało kilka odmian sieci ART o nazwach odpowiednio ART1, ART2 i ART3. Różnią się one tym, że pierwsza, nieco prostsza, działa na sygnałach binarnych, pozostałe na analogowych lub dyskretnych, lecz na pewno mających więcej niż 2 wartości poziomu sygnału wejściowego, dodatkowo sieć ART3 zapewnia większą stabilność działania w porównaniu do ART2. W dalszym ciągu zajmiemy się siecią binarną ART1, co jednak nie powinno wpłynąć negatywnie na zrozumienie sposobu działania tych sieci.

Sieć ART



Schemat budowy sieci ART.

N_{11}...$N_{1,l}$ - neurony warstwy wejściowej; N_{21}...$N_{2,k}$ - neurony warstwy wyjściowej - konkurencyjnej; X_0, … X_l – sygnały wejściowe, Y_0, … Y_k – sygnały wyjściowe, v_{ji} – wagi z warstwy wejściowej do wyjściowej, w_{ij} – wagi z warstwy wyjściowej do wejściowej, Sterowanie siecią – synchronizuje i kontroluje działania warstw, wykrywa obecność nowego sygnału na wejściu i inicjuje działanie sieci.

Sieć ART
działanie

1. Sieć otrzymuje na wejście nowy sygnał X, warstwa 2 jest zerowana, otrzymuje od warstwy 1 niezmieniony obraz X.
2. Neurony warstwy 2 obliczają swoje wyjścia, zwycięża ten, który daje najwyższy sygnał wyjściowy (jego wagi w są najbardziej zbliżone do obrazu) inne neurony są gaszone przez połączenia hamujące.
3. Obliczany jest iloraz między sygnałem wejściowym, a zwrotnym z warstwy 2 (od zwycięskiego neuronu oznaczonego *),

$$D=\frac{\sum_{i=1}^{l} w_{ij}^{*} * x_i}{\sum_{i=1}^{l} x_i}$$

a następnie:

- jeśli sygnały są prawie identyczne (decyduje o tym porównanie D z progiem czujności sieci τ , czyli $D<\tau$), to sieć jest w rezonansie, następuje uaktualnienie wag, aby zmniejszyć różnice zapamiętanego wzorca i otrzymanego sygnału;
- jeśli sygnały są różne, to zostaje przydzielony do zwycięskiego neuronu, który staje się reprezentantem nowej klasy danych, jego wagi są odpowiednio korygowane.

Sieć ART
działanie

4. Następnie sieć oczekuje na nowy sygnał, po otrzymaniu którego wraca do pkt. 1.

W trakcie pracy sieci istotne jest działanie układu sterującego, którego zadaniem jest wykrywanie stanów wejść oraz stanów sieci i odpowiednie reagowanie na nie, przez zmianę aktywności działań warstw.

Sieć ART
korekcja wag

1. Wagi są inicjowane następująco:

$$v_{ji}(0)=\frac{1}{1+l} \qquad w_{ij}(0)=1$$

2. Po zakończeniu cyklu pracy sieci (wytypowaniu zwycięzcy lub zwycięzcy i nowej kategorii) wagi są korygowane następująco:

$$w^*_{ij}(t+1)=w^*_{ij}(t)*x_i \qquad v^*_{ji}(t+1)=\frac{w^*_{ij}(t+1)}{0{,}5+\sum_{j=1}^{k} w^*_{ij}(t+1)*x_i}$$

w^*_{ij} – waga zwycięskiego neuronu z warstwy 2 do 1; v^*_{ij} – waga do zwycięskiego neuronu z warstwy 1 do 2; l – liczba neuronów w warstwie 1; k – liczba neuronów w warstwie 2; t, t+1 – kolejne chwile czasowe/kroki działania sieci.

Sieć ART
zastosowania

Sieci z rodziny ART są stosowane przede wszystkim jako klasyfikatory nieznanych danych, które bez wcześniejszej nauki podzielą na kategorie podobnych danych.

Sieci neuronowe - podsumowanie

Sieci neuronowe były pierwszymi praktycznie wykorzystanymi urządzeniami sztucznej inteligencji. Ich użycie budziło ogromne emocje, gdyż wydawało się, że droga do zbudowania sztucznego mózgu jest już bardzo łatwa i krótka. Jednak praktyka nie okazała się aż tak prosta. Mimo, że powstają coraz nowsze rozwiązania w tej dziedzinie: nowe sieci, rozwiązania hybrydowe różnych sieci, a także rożnych dziedzin SI z sieciami neuronowymi, to jednak droga do budowy sztucznego umysłu wydaje się jeszcze bardzo daleka. I w tej dziedzinie sieci na pewno nie spełniły pokładanych w nich nadziei, jednakże są z powodzeniem stosowane w tzw. „słabej sztucznej inteligencji” do przetwarzania informacji w zadaniach trudnych do wykonania innymi metodami.

Dziękuję za uwagę!