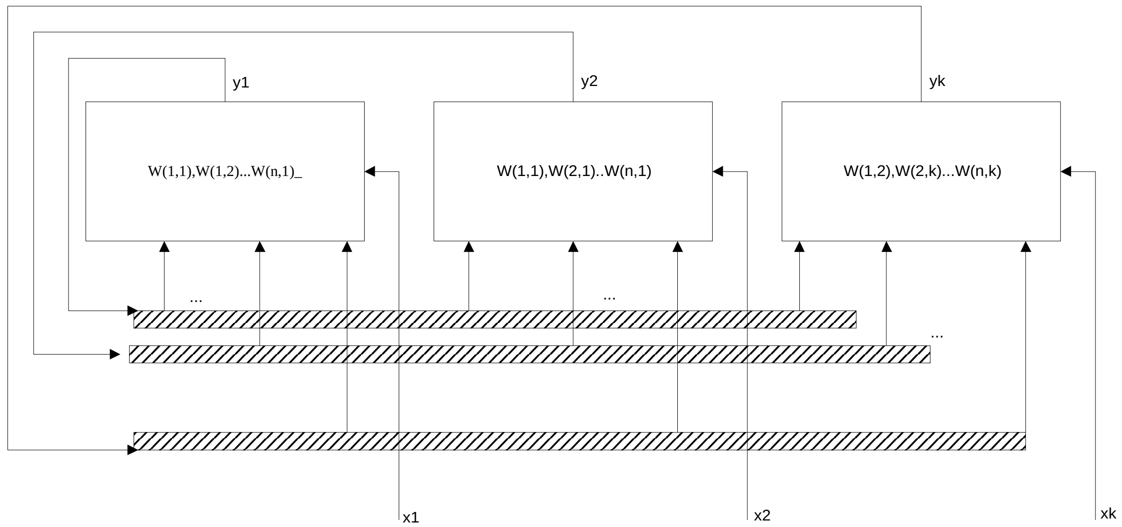
Sieć Hopfielda

Zdefiniowana w roku 1982 , wprowadziła sprzężenie zwrotne do struktur sieci .

Cechy charakterystyczne:

- brak jednokierunkowego przepływu sygnału
- nie możemy wyróżnić warstwy wejściowej, wyjściowej , pośredniej



Jednowarstwowy, jednowymiarowy model sieci.

W sieci tej neurony mają nieliniowe charakterystyki :

$$y_m^{(j)} = \varphi(e_m^{(j)}) \quad \text{gdzie:} \quad e_m^{(j)} = \sum_{i \in \xi} e_i^{(m)} * y_i^{(j)} + x_m^{(j)}$$

a nieliniowość $y = \varphi(e)$ dana jest prostą binarną funkcją:

$$y_m^{(j+1)} = \begin{cases} 1 & dla \quad e_m^{(j)} > w_0^{(m)} \\ y_m^{(j)} & dla \quad e_m^{(j)} = w_0^{(m)} \\ -1 & dla \quad e_m^{(j)} < w_0^{(m)} \end{cases}$$

- współczynniki wagowe $w_i^{(m)}$ łączące wyjście i- tego neuronu z wejściem m-tego neuronu **nie zależą od j ,** przy czym j określa chwilę czasową w procesie dynamicznego pobudzenia sieci, w której się ona obecnie znajduje.

- sumowanie sygnałów wyjściowych $y_i^{(j)}$ z poszczególnych neuronów we wzorze definiującym łączne pobudzenie $e_m^{(j)}$ odbywa się po **wszystkich** elementach sieci, a zatem w sieci przewidziane są połączenia z warstw wyjściowych do warstw wcześniejszych - czyli **sprzężenia zwrotne.**

Sieć o takim schemacie połączeń nazywać będziemy siecią **autoasocjacyjną.**

Procesy w sieciach Hopfielda

Ze względu na specyficzną architekturę sieci możliwe staje się generowanie przebiegów dynamicznych na skutek tego, iż w pewnym momencie j sygnały wyjściowe $y_m^{(j)}$ stają się automatycznie wartościami wejściowymi $y_i^{(j+1)}$.

Realizowane jest więc nieliniowe wektorowe odwzorowanie :

$$Y^{(j+1)} = \Xi(X^{(j)}, Y^{(j)})$$

przy założeniu iż $x_m^{(j)} \equiv 0$ dla wszystkich m i dla wszystkich $j > 0$ zależność upraszcza się do :

$$Y^{(j+1)} = \Xi(Y^{(j)})$$

Kolejne wartości $Y^{(1)}$, $Y^{(2)}$,..., $Y^{(j-1)}$, $Y^{(j)}$ obserwujemy w przestrzeni stanu $\gamma \subseteq R^k$ do której należą wszystkie wektory sygnałów wyjściowych $Y^{(j)}$. Możliwe więc jest pojawienie się oscylacji, przebiegów rozbieżnych, a nawet pojawienie się chaosu. W 1983 Cohen i Grossberg wykazali, iż sieć generuje stabilne rozwiązania, jeśli uniemożliwi się autoasocacyjność pojedynczych neuronów :

$$w_m^{(m)} = 0$$

oraz zapewni się symetrię sieci: $w_i^{(m)} = w_m^{(i)}$

Stany równowagi w sieci Hopfielda

Stan równowagi sieci można potraktować jako problem wyboru stanu o minimalnej ,,energii'' sieci. Definiuje się więc funkcję energii (funkcję Lapunowa) o następującej postaci:

$$E^{(j)} = -\frac{1}{2} \sum_{i\in\xi} \sum_{m\in\xi} w_i^{(m)} y_i^{(j)} y_m^{(j)} - \sum_{i\in\xi} x_i^{(i)} y_i^{(j)} + \sum_{i\in\xi} w_0^{(i)} y_i^{(j)}$$

Zmiana δE zachodząca na skutek zmiany stanu sieci spowodowanej zmianą sygnału wyjściowego *i*- tego neuronu wynosi:

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

albo:

$$\delta E^{(j)} = -[e_i^{(j)} - w_0^{(i)}]\delta y_i^{(j)}$$

Przyjmijmy, że dla kroku j pobudzenie neuronu $e_i^{(j)}$ przekracza próg $w_0^{(i)}$.

Wówczas na wyjściu pojawia się sygnał $y_i^{(j)} = 1$, zatem czynnik $\delta y_i^{(j)}$ musi być dodatni lub zerowy. Równocześnie ze względu na: $e_i^{(j)} > w_0^{(i)}$ czynnik w nawiasie kwadratowym musi być dodatni, a więc zmiana energii może być jedynie **ujemna** lub **zerowa.**

Analogicznie dla: $e_i^{(j)} < w_0^{(i)}$, zmiana energii jest również ujemna lub zerowa, natomiast przy $e_i^{(j)} = w_0^{(i)}$ energia nie zmienia się.

Powyższe rozumowanie potwierdza, iż sieć dąży do osiągnięcia minimum, następnie do zahamowania zmian i osiągnięcia stanu stabilnego.

Pamięć autoasocjacyjna

Pamięć autoasocjacyjna (pamięć skojarzeniowa)
nazywana czasami **CAM** od **C**ontent **A**dressable **M**emory.

Sieć powinna zapamiętać szereg wzorców D_j , a po pojawieniu się wektora $\boldsymbol{X}$ podobnego do jednego z zapamiętanych wzorców sieć powinna odtworzyć zapamiętany wektor $\boldsymbol{D}$ podobny do wektora $\boldsymbol{X}$.

Algorytm uczenia - metoda Hebba :

$$w_i^{(m)} = \sum_{j=1}^{M} y_i^{(j)} y_m^{(j)}$$

gdzie: $y_i^{(j)}$ jest i-tą składową wektora D_j .

Macierz **W** połączeń pomiędzy elementami sieci ma postać:

$$\mathbf{W} = \sum_{j=1}^{M} D_j^T D_j$$

Działanie sieci polega na jednorazowym podaniu sygnałów wejściowych $\boldsymbol{X}$ i oczekiwaniu na określony stan stabilny odpowiedzi wzorca $\boldsymbol{D}$ skojarzonego z sygnałem wejściowym.

Pojemność tak utworzonej sieci szacuje się na około 0.15N, gdzie N jest liczbą neuronów w sieci.

Sygnały wyjściowe przy pamięciach skojarzeniowych traktuje się jako ciągłe z przedziału domkniętego <-1,1> ($y_m^{(j)} \in$ [-1,1]) .

Nieliniowa funkcja aktywacji $y_m^{(j)} = \varphi(e_m^{(j)})$ może być przedstawiona w postaci:

- funkcji liniowej

$$y_m^{(j)} = \varphi(e_m^{(j)}) = k\, e_m^{(j)},$$

- gdzie k jest zadanym współczynnikiem,
- funkcji skoku jednostkowego

$$y_m^{(j)} = \varphi(e_m^{(j)}) = \begin{cases} 1 \ \ dla \ \ e_m^{(j)} > 0 \\ 0 \ \ dla \ \ e_m^{(j)} \le 0 \end{cases},$$

- funkcji signum

$$y_m^{(j)} = \varphi(e_m^{(j)}) = \begin{cases} 1 \ \ dla \ \ e_m^{(j)} > 0 \\ 0 \ \ dla \ \ e_m^{(j)} = 0, \\ -1 \ \ dla \ \ e_m^{(j)} < 0 \end{cases}$$

- zmodyfikowanej funkcji signum

$$y_m^{(j)} = \varphi(e_m^{(j)}) = \begin{cases} 1 \ \ dla \ \ e_m^{(j)} > 0 \\ -1 \ \ dla \ \ e_m^{(j)} \le 0 \end{cases},$$

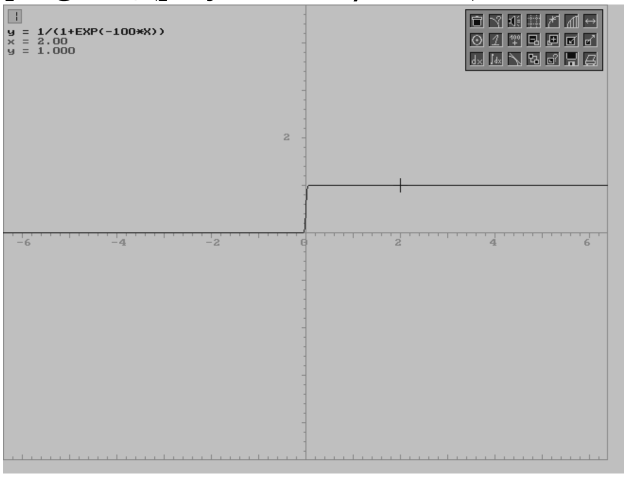
- funkcji perceptronowa

$$y_m^{(j)} = \varphi(e_m^{(j)}) = \begin{cases} e_m^{(j)} \ \ dla \ \ e_m^{(j)} > 0 \\ 0 \ \ dla \ \ e_m^{(j)} \le 0 \end{cases},$$

- funkcji sigmoidalnej

$$y_m^{(j)} = \varphi(e_m^{(j)}) = \frac{1}{1 + \exp(-\beta\, e_m^{(j)})},$$

  gdzie β jest zadanym parametrem. Pożądane jest duże β, dla którego funkcja przypomina funkcję progową (przykład dla $\beta = 100$) :



- funkcji tangensoidalnej

$$y_m^{(j)} = \varphi(e_m^{(j)}) = \mathrm{tgh}\left(\frac{\alpha e_m^{(j)}}{2}\right) = \frac{1 - \exp(-\alpha e_m^{(j)})}{1 + \exp(-\alpha e_m^{(j)})},$$

  gdzie: α jest zadanym parametrem, a $\mathrm{tgh}(\cdot)$ jest funkcją tangens hiperboliczny.

Maszyny Bolzmanna

Koncepcja maszyny Bolzmanna oparta jest na założeniu , że stan $y_m^{(j)}$ każdego neuronu (sygnału wyjściowego) może zmieniać się w sposób losowy z określonym prawdopodobieństwem zależnym od energii i temperatury.

$$p(E,T) = e^{-\frac{E}{kT}}$$

W odniesieniu do systemu informatycznego wzór powyższy przyjmuje postać:

$$p_m^{(j)} = 1/[1 + \exp(-\delta E_m^{(j)} / T^{(j)})]$$

gdzie: $E_m^{(j)} = e_m^{(j)} - w_0^{(m)}$ jest nadwyżką energii łącznego pobudzenia $e_m^{(j)}$ ponad próg pobudzenia $w_0^{(m)}$ dla każdego kroku j

δ - pewna arbitralnie dobierana stała

$T^{(j)}$ - symulowana temperatura sieci dla kroku j-tego

Algorytm doprowadzania sieci do równowagi:

1. Dla ustalonego $T^{(j)}$ wyliczane są wszystkie wartości $p_m^{(j)}$, a następnie **losowo** z prawdopodobieństwem $p_m^{(j)}$ ustawiane są wartości sygnałów wyjściowych $y_m^{(j)}$. Konkretnie dla każdego kroku j i dla każdego neuronu m losowana jest z konkretnym rozkładem prawdopodobieństwa wartość przypadkowa $\xi \in [0, 1]$, a następnie ustala się wartość $y_m^{(j)}$ zgodnie z regułą:

$$y_m^{(j))} = \begin{cases} 1 & dla \quad \xi \le p_m^{(j)} \\ 0 & w\ przeciwnym\ wypadku \end{cases}$$

2. Obniża się stopniowo wartość $T^{(j)}$ w kolejnych krokach np. według reguły:

$$T^{(j+1)} = T^{(j)} - \varepsilon$$

albo

$$T^{(j+1)} = T^{(j)}(1 - \varepsilon)$$

i powtarza się (1) aż do osiągnięcia stanu równowagi.

Uogólniony algorytm Hintona i Sejnowskiego

zbiór uczący $U = \{< \mathbf{X}^k, \mathbf{Z}^k >, k = 1,2, N\}$

X-wektor niektórych sygnałów wejściowych

Z-wektor określający oczekiwane sygnały wyjściowe niektórych elementów sieci

A. Obliczanie ,, związanych " prawdopodobieństw.

1. Wymusza się wynikające z ciągu uczącego wartości wejść X^k i wyjść Z^k wyróżnionych neuronów wejściowych i wyjściowych.
2. Pozwala się sieci dojść do stanu równowagi
3. Rejestruje się sygnały wyjściowe $y_m^{(k)}$ wszystkich elementów sieci
4. Powtarza się powyższe punkty dla wszystkich elementów ciągu uczącego x zbierając statystykę, dzięki której po zakończeniu pokazów wszystkich elementów ciągu uczącego możliwe jest obliczenie prawdopodobieństwa P_{ij}^+ tego że sygnały neuronów o numerach *i* oraz *j* mają **równocześnie** wartość 1

B. Obliczanie ,,nie związanych" prawdopodobieństw.

1. Wymusza się **przypadkowe** wartości sygnałów wyjściowych wszystkich elementów sieci
2. Pozwala się dojść sieci do stanu równowagi
3. Rejestruje się sygnały wyjściowe $y_m^{(k)}$ wszystkich elementów sieci
4. Powtarza się poprzednie punkty wielokrotnie, zbierając statystykę, dzięki której po zakończeniu pokazów wszystkich elementów ciągu uczącego możliwe jest obliczenie prawdopodobieństwa P_{ij}^- tego, że sygnały wyjściowe neuronów o numerach i oraz j mają **równocześnie** wartość 1

C. Na podstawie P_{ij}^+ oraz P_{ij}^- koryguje się wartości współczynników wagowych $w_i^{(j)}$ oraz $w_j^{(i)}$ łączących neurony o numerach *i* oraz *j*. Wartości te są zwiększane o wartość δ_{ij} wyliczoną ze wzoru :

$$\delta_{ij} = \eta[P_{ij}^+ - P_{ij}^-]$$

Wartości odpowiednich współczynników dają się znaleźć już po kilku iteracjach.

Rozwiązywanie problemu komiwojażera

Problem polega na ustaleniu optymalnej trasy objazdu *n* miast przez wędrownego sprzedawcę, który chce odwiedzić wszystkie miasta i przebyć najkrótszą drogę.

Czas rozwiązywania tego problemu rośnie wykładniczo przy wzroście liczby miast *n* i możliwe jest utworzenie D = n!/(2n) rozróżnialnych tras.

$$n=60 \text{ to } D=69.34155*10^{78}$$

Przy zastosowaniu do obliczeń sieci neuronowej kluczem jest odpowiednia reprezentacja danych wejściowych. Jedno z możliwych możliwości zakłada, że:

- każde miasto reprezentowane jest przez jeden wiersz zawierający *n* neuronów,
-
- w każdym wierszu **dokładnie** jeden neuron powinien przyjmować wartość 1,
-
- pozycja na której neuron przyjmuje wartość 1 określa jednoznacznie *nr miasta* i kolejność, w której ma być ono odwiedzone.

Łączna liczba neuronów wynosi n^2.

Funkcję energii opisują cztery składniki:

$$E_1 = \frac{A}{2} \sum_x \sum_i \sum_{i \neq j} y_{xi} y_{xj}$$

$$E_2 = \frac{B}{2} \sum_i \sum_x \sum_{z \neq x} y_{xi} y_{zj}$$

$$E_3 = \frac{C}{2} [\sum_x \sum_i y_{xi} - n]^2$$

$$E_4 = \frac{D}{2} \sum_x \sum_{z \neq x} \sum_i d_{xz} y_{xi} (y_{z,i+1} + y_{z,i-1})$$

Znaczenie poszczególnych składników:

E_1- ma zerową wartość wtedy i tylko wtedy jeśli w każdym wierszu jest najwyżej jedna jedynka,

E_2 - wartość zerowa wtedy i tylko wtedy, gdy w kolumnie oznaczającej konkretny etap podróży będzie najwyżej jedna jedynka,

E_3 - zero, gdy w macierzy jest dokładnie *n* jedynek,

E_4 - długość wybranej drogi.

Wybierane arbitralnie współczynniki A,B,C,D są względnymi wagami poszczególnych składników.

Przykładowe wartości tych współczynników są w pakiecie NeuralWorks Professional II/PLUS firmy NeuralWare następujące:

$$A \cong B \cong D \cong 1000$$

C dobierane jest w zależności od relacji pomiędzy parametrem *n* używanym we wzorze dla składnika E_3, a rzeczywistą liczbą miast n^*

C = 3/4*(n - n*)

W rozważanym pakiecie modelującym sieć neuronową współczynniki wagowe określające parametry połączeń pomiędzy *i*-tym neuronem z *x*-tej warstwy, a *j* - tym neuronem *z* - tej warstwy wyrażają się wzorem:

$$w_{xi,zj} = -A\delta_{xz}(1-\delta_{ij}) - B\delta_{ij}(1-\delta_{xz}) - C - D\delta_{xz}(\delta_{j,i+1} + \delta_{j,i-1})$$

gdzie: δ_{ij} oznacza funkcję Kroneckera

$$\delta_{ij} = \begin{cases} 1 & dla \quad i = j \\ 0 & w \ \ przeciwnym \ wypadku \end{cases}$$

Parametrami dodatkowymi są wyrazy wolne (progi pobudzenia) identyczne dla wszystkich elementów:

$$w_{xz}^{0} = Cn$$

Główną zaletą zastosowania sieci neuronowej do rozwiązywania problemu TSP jest to, iż wzrost rozmiaru problemu *n* nie powoduje wydłużenia czasu obliczeń, a jedynie pociąga za sobą rozbudowę sieci.