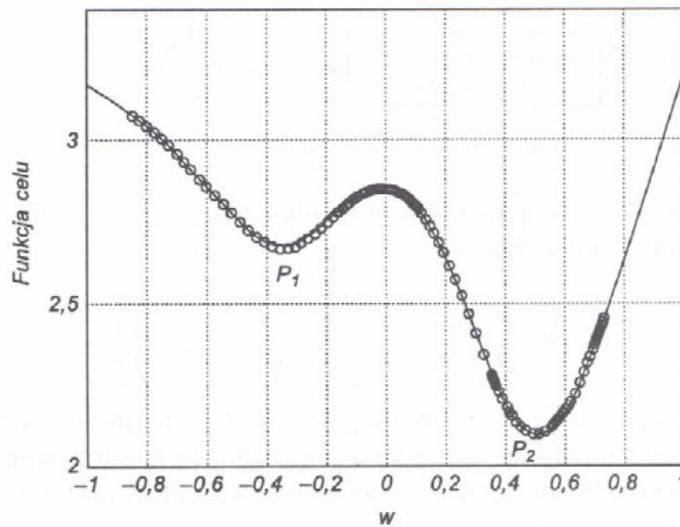


1. W jaki sposób radzimy sobie z problemem zatrzymywania się w lokalnym minimum błędu podczas procesu uczenia?

Stosujemy technikę *momentum*.

$$\Delta w(k) = \eta * \delta * f'(\varphi) * U + \alpha * \Delta w(k-1), \quad \alpha - \text{współczynnik momentum } [0;1]$$

Kiedy stosowany jest współczynnik/moment pędu, korekta wag neuronu zależy nie tylko od sygnału wejściowego i błędu, jaki neuron popełnił, ale również od tego, jak duża była korekta wag w poprzednim kroku uczenia. Jeżeli w kolejnych iteracjach kierunek modyfikacji wagi był ten sam, to składnik zawierający współczynnik momentum powoduje wzrost wartości wagi i przesunięcie jej z większą siłą w kierunku minimum. Natomiast w przypadku odwrotnym składnik ten powoduje zahamowanie gwałtownych zmian wag. Ogólnie działanie członu momentum uaktywnia się na płaskich odcinkach funkcji celu, jak również w pobliżu minimum lokalnego.



Zalety techniki momentum:

- szybsze uczenie,
- przeskakuje płytkie minima lokalne,
- tłumi oscylacje.

2. Opisać znane metody selekcji dla algorytmu genetycznego

• Metoda elitarna:

Jeżeli najlepszy chromosom nie wygra selekcji (nie trafi do populacji rodzicielskiej), wrzucamy go do populacji potomków w miejsce najslabszego. Wada: nacisk na przeszukiwanie lokalne kosztem globalnego (napór selekcji). Zaleta: łatwiej znajduje minima lokalne.

• Metoda deterministyczna:

a) dla każdego chromosomu liczymy prawdopodobieństwo jego wylosowania:

$$p(CH_i) = \frac{f(CH_i)}{\sum_{i=1}^N f(CH_i)}$$

b) liczymy oczekiwaną liczbę potomków dla każdego chromosomu:

$$e_i = p(CH_i) * N$$

c) każdy chromosom otrzymuje tyle potomków, ile wynosi część całkowita z e_i

d) sortujemy wartości oczekiwane e_i pod wg części ułamkowej (malejąco)

e) dopełniamy populację najlepszymi pod względem części ułamkowej wartościami oczekiwanymi chromosomów

- **Metoda wartości oczekiwanej:**

$$M(ch) = \frac{F(ch)}{Fsr(ch)}, \text{ czyli: liczba potomnych} = \text{przystosowanie chromosomu} / \text{przystosowanie całej populacji};$$

W każdej iteracji przed przystąpieniem do losowania obliczana jest oczekiwana ilość potomków wg powyższego wzoru. Jeśli chromosom jest do reprodukcji i jednocześnie do krzyżowania to $M=M-0.5$. Jeżeli chromosom wylosowany jest do reprodukcji i do mutacji to $M=M-1$. Jeżeli $M < 0$ wówczas chromosom nie bierze udziału w losowaniu.

- **Metoda stochastyczna z resztą:**

a) dla każdego chromosomu liczymy prawdopodobieństwo jego wylosowania:

$$p(CH_i) = \frac{f(CH_i)}{\sum_{i=1}^N f(CH_i)}$$

b) liczymy oczekiwaną liczbę potomków dla każdego chromosomu

$$e_i = p(CH_i) * N$$

c) każdy chromosom otrzymuje tyle potomków, ile wynosi część całkowita z e_i

d) dla każdego chromosomu rodziców koło ruletki jest skalowane wg części ułamkowej e_i

e) dopełniamy populację losując za pomocą koła ruletki

- **Metoda turniejowa:**

Zakładamy liczbę k chromosomów, które mają brać udział w rozgrywce. Losujemy k chromosomów i najlepszy z nich wygrywa rywalizację i trafia do populacji. Metoda ta nie posiada ona takich ograniczeń jak metoda ruletki; można szukać min., max., wartości funkcji przystosowania mogą być dodatnie lub ujemne. Dostosowując wartość k szukamy kompromisu pomiędzy najlepszym wynikiem, a szybkością obliczeń.

- **Selekcja rankingowa:**

Sortujemy chromosomy od najlepszego do najgorszego (wg wartości przystosowania) i nadajemy im wagę ($rank=1$ – bestest ;). Następnie skalujemy koło ruletki:

$$p(rank) = q(rank - 1) * r, \quad q, r - \text{parametry dobierane empirycznie przez użytkownika}$$

3. Wymienić operatory występujące w kodowaniu rzeczywistym

Operator krzyżowania ma za zadanie łączyć w różnych kombinacjach cechy pochodzące z różnych osobników populacji, zaś **operator mutacji** ma za zadanie zwiększać różnorodność tych osobników.

Rodzaje krzyżowań:

- proste (jedno- lub wielopunktowe), np. 011|11 x 110|10 -> 01110, 11011
- proste jednolite: dla każdego genu losujemy, do którego potomka natrafić
- arytmetyczne: chromosomy potomne są kombinacją liczbową rodziców
- heurystyczne: wybór heurystyczny, powstaje 1 lub 0 potomków

Rodzaje mutacji:

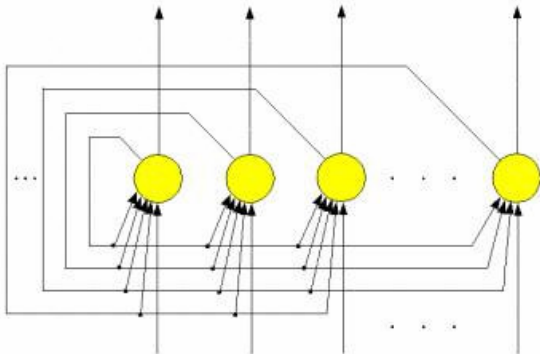
- równomierna: gen jest zastępowany poprzez losową wartość
- nierównomierna: losowanie binarne, początkowo mutacja jest bardzo losowa, ale w miarę postępowania iteracji będzie oscylować wokół optimum
- brzegowa: losowanie binarne spośród brzegów przedziału dla genu

4. Różnice w budowie i działaniu pomiędzy perceptronem a RBF

	Perceptron	RBF
Budowa	Zbudowana z 1 lub większej ilości warstw	Zwykle tylko 2 warstwy
Funkcja aktywacji	Skokowa: [0 1] lub [-1 1]	Funkcja radialna, której wartość zależy od odległości od pewnego centrum $\varphi(\ x - c\)$, np. funkcja Gaussa
Uczenie	Nadzorowane, nienadzorowane	Polega na doborze ilości neuronów oraz środków i szerokości dla RBF
Separacja	Liniowa	Wielomianowa

5. Opisać budowę, działanie i uczenie sieci Hopfielda

Sieć Hopfielda jest siecią rekurencyjną (tj. dopuszcza się istnienie sprzężeń zwrotnych tzn. gdy wyjścia neuronów mogą być połączone z wejściami neuronów tej samej warstwy lub warstw pośrednich, ale bez połączeń zwrotnych). Informacja oscyluje między warstwami lub w obrębie warstw aż do spełnienia pewnego kryterium zbieżności i dopiero wtedy jest przekazywana na wyjście sieci.



Rysunek jest trochę brzydki, bo uwzględnia sprzężenia zwrotne neuronów do samych siebie. Sieć Hopfielda takich sprzężeń nie dopuszcza.

Założenia:

- Wszystkie neurony są ze sobą połączone z wagami synaps W_{ij} .
- Macierz wag połączeń jest symetryczna, $W_{ii}=0$, $W_{ij} = W_{ji}$ - symetria jest wygodna z teoretycznego punktu widzenia (pozwala wprowadzić funkcję energii), choć jest nierealistyczna z biologicznego punktu widzenia.

Działanie sieci Hopfielda

• Tryb uczenia

- zakładamy jeden lub więcej wzorców uczących x
- wagi w_{ij} przyjmują wartości uśrednione wielu próbek uczących - dobierane są wg reguły Hebba:

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

• Tryb odtwarzania

- wartości wag są zamrożone
- zakłada się określony stan początkowy neuronów ($y(0)=x$)
- następuje proces przejściowy, przebiegający według zależności:

$$y_i(k) = \text{sgn} \left(\sum_{j=1, j \neq i}^N w_{ij} y_j(k-1) \right)$$

który kończy się w określonym minimum lokalnym, w którym $y(k)=y(k-1)$.

6. Różnica pomiędzy uczeniem nadzorowanym a nienadzorowanym

Uczenie nadzorowane, zwane jest również uczeniem z nauczycielem, polega na bezpośrednim porównaniu sygnału wyjściowego sieci ze znanymi prawidłowymi odpowiedziami. Szczególnym przypadkiem tego sposobu uczenia sieci jest uczenie ze wzmocnieniem (sygnał zwrotny niesie informację czy sygnał wyjściowy jest prawidłowy, czy nie - sama prawidłowa odpowiedź nie jest podawana). Celem uczenia pod nadzorem jest minimalizacja odpowiednio zdefiniowanej funkcji celu.

Uczenie bez nadzoru stosujemy kiedy cel uczenia nie jest określony przez konkretne, prawidłowe przykłady. Jedyną informacją, która musi wystarczyć jest sama wewnętrzna struktura sygnałów wejściowych (przykładów, przy pomocy których można wytrenować sieć). Na podstawie wewnętrznych zależności w podanych informacjach sieć neuronowa musi stworzyć własne kategorie i będzie rozpoznawać (klasyfikować) podane sygnały wejściowe (generując odpowiednie sygnały na wyjściu).

7. Metody uczenia nienadzorowanego

Reguła Hebba:

1. Jeżeli neurony A i B połączone synapsą są pobudzane jednocześnie (tj. synchronicznie) to połączenie synaptyczne je łączące jest wzmacniane.
2. Jeżeli neurony A i B połączone synapsą są pobudzane niejednocześnie (tj. asynchronicznie) to połączenie synaptyczne je łączące podlega osłabieniu.

Innymi słowy wzmocnieniu ulegają te wagi dla których duże pobudzenie powoduje silną reakcję.

Reguła przedstawiona jest wzorem:

$$W_{ij}(k+1) = W_{ij}(k) + \eta * \varphi_i * \varphi_j$$

gdzie:

w_{ij} – waga połączenia synaptycznego między neuronem i oraz j

η – współczynnik uczenia

U_i - wejście

φ – aktywność i-tego i j-tego neuronu

Siec ma własności uśredniające - ucząc się generalizuje próbki.

Modyfikacje reguły Hebba

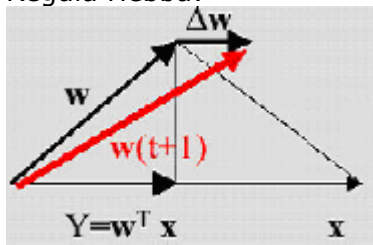
- Narzucenie na wagi górnego i dolnego ograniczenia (celem eliminacji rozbieżności)
- Normalizacja wektora wag po każdym kroku liczenia (kosztowne obliczenia)
- Modyfikacja wektora wag, tak aby podlegał samoorganizacji

Reguła Oja:

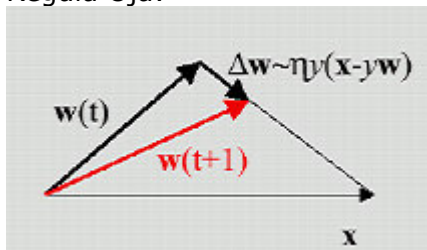
(uzupełnienie reguły Hebba)

$$W_{ij}(k+1) = \eta * \varphi^{(p)} \left[U^{(p)} - \varphi^{(p)} * W_{ij}(k) \right]$$

Reguła Hebba:



Reguła Oja:



Uczenie konkurencyjne:

W konkurencyjnej metodzie uczenia sieci, tylko jeden element wyjściowy może znajdować się w stanie aktywnym. Nazywany jest on zwycięzcą, a schemat takiej reguły aktywacji neuronów określany jest mianem "zwycięzca bierze wszystko" (ang. Winner Takes All - **WTA**). Algorytmy WTA, w których tylko jeden neuron może podlegać adaptacji w każdej iteracji, są jednak algorytmami słabo zbieżnymi, szczególnie przy dużej liczbie neuronów. W praktyce zostały one zastąpione algorytmami **WTM** (ang. Winner Takes Most), w których oprócz zwycięzcy uaktualniają swoje wagi również neurony z jego sąsiedztwa.

$$\Delta w = \eta(k) * S(i) * (U^{(p)} - w),$$

$S(i)$ – sąsiedztwo, np. prostokątne, Gaussa

Kohonen zaproponował regułę uczenia konkurencyjnego, w której, w odróżnieniu od reguły standardowej, modyfikacji wag dokonuje się nie tylko dla neuronu zwycięskiego, lecz również dla pewnej liczby neuronów z jego otoczenia. Reguła ta wymaga wprowadzenia tzw. topologicznego uporządkowania neuronów w warstwie wyjściowej (zwanej też mapą cech), dla której można wprowadzić pojęcie otoczenia neuronu zwycięskiego. W klasycznej sieci Kohonena neurony są logicznie rozmieszczone, sąsiedztwo liczone jest w logicznej strukturze. *Reguła ma zastosowanie w klasyfikacji i kompresji danych (zwłaszcza obrazów).*

8. Ograniczenia nakładane na funkcje w algorytmach genetycznych

Istnieją ograniczenia:

- liniowe:
 - + ograniczenia w postaci nierówności: $G_j(x) > 0$
 - + ograniczenia w postaci równań $H_j(x) = 0$ (gorsze)
- nieliniowe

9. Jak pozbyć się ograniczeń w algorytmach genetycznych:

Metody radzenia sobie z ograniczeniami

1. Odpowiednie zaprojektowanie zadania, ustalenie kodowania i operatorów genetycznych.
2. Metoda algorytmów naprawczych.
3. Metoda kary: polega na *karaniu* chromosomów nie spełniających zadanych ograniczeń. Zwykle realizuje się to poprzez wzbogacenie funkcji przystosowania:

$$F'(ch) = F(ch) + r \sum_{i=1}^m \phi(G_i(ch)),$$

gdzie:

r – współczynnik kary,
 ϕ – funkcja kary

4. Metoda kary śmierci.

„Każdy najgorszy dopuszczalny chromosom musi być lepszy niż najgorszy niedopuszczalny”

10. PRBF

Probabilistyczne sieci RBF wykorzystywane są w celach klasyfikacyjnych oraz do aproksymacji. W założeniach PRBF ma tyle neuronów radialnych, ile mamy próbek, a każdy z nich ma taki sam promień. Środek każdego neuronu pokrywa się z jedną z próbek uczących. Próbkę uczącą muszą być znormalizowane i nie powinny być zaszumione ani zdublowane (generalnie danych powinno być jak najmniej).

Klasyfikacja

Podczas działania, funkcja wyznacza estymaty (oszacowania) funkcji gęstości prawdopodobieństwa przynależności danych klasyfikowanych próbek do klasy K

$$g_k(x) = \frac{1}{\eta_k * \sigma_k} \sum_{i=1}^{n_k} G\left(\frac{|x - x_i^{(k)}|}{\tau_k}\right)$$

η_k – ilość próbek w klasie K
 σ_k – szerokość neuronu dla klasy K
 x_i – środki neuronów powiązane ze środkami klas
 x – klasyfikowana próbka

Aproksymacja

Wzór na tyle skomplikowany, że nie warto go nawet przepisywać :)

11. Jak pozbyć się ograniczeń w algorytmach genetycznych?

Metody radzenia sobie z ograniczeniami

1. Odpowiednie przygotowanie próbek uczących (normalizacja).
2. Odpowiednie skonstruowanie problemu, ustalenie kodowania, itp.
3. Metoda algorytmów naprawczych.
4. Metoda kary: polega na *karaniu* chromosomów nie spełniających zadanych ograniczeń. Zwykle realizuje się to poprzez wzbogacenie funkcji przystosowania:

$$F'(ch) = F(ch) + r \sum_{i=1}^m \phi(G_i(ch)),$$

gdzie:

r – współczynnik kary,

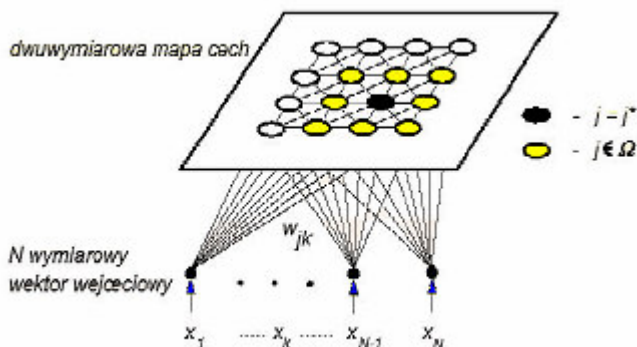
Φ – funkcja kary

5. Metoda kary śmierci.

„Każdy najgorszy dopuszczalny chromosom musi być lepszy niż najgorszy niedopuszczalny”

12. Reguła Kohonena

Kohonen zaproponował regułę uczenia konkurencyjnego, w której, w odróżnieniu od reguły standardowej, modyfikacji wag dokonuje się nie tylko dla neuronu zwycięskiego, lecz również dla pewnej liczby neuronów z jego otoczenia. Reguła ta wymaga wprowadzenia tzw. topologicznego uporządkowania neuronów w warstwie wyjściowej (zwanej też mapą cech), dla której można wprowadzić pojęcie otoczenia neuronu zwycięskiego. W klasycznej sieci Kohonena neurony są logicznie rozmieszczone, sąsiedztwo liczone jest w logicznej strukturze. *Reguła ma zastosowanie w klasyfikacji i kompresji danych (zwłaszcza obrazów).*



Algorytm uczenia sieci Kohonena

1. Przypisz wagom sieci o M neuronach warstwy wyjściowej i N wejściach niewielkie liczby losowe. Ustal liczbę neuronów należących do początkowego otoczenia neuronu ($> M/2$)
2. Dołącz nowy wektor uczący $x = [x_1, \dots, x_k, \dots, x_N]$ do wejścia.
3. Wyznacz odpowiedź każdego neuronu warstwy wyjściowej
4. Znajdź neuron zwycięski j^* (największa wartość odpowiedzi).
5. Wyznacz nowe wartości wag dla neuronu zwycięzcy j^* i jego sąsiedztwa (tj. jego otoczenia) stosując regułę Kohonena.
6. Zmień odpowiednio wartości współczynnika uczenia i otoczenia.
7. Powtórz 2-6 dla następných wzorców wejściowych aż do chwili ustalenia się odpowiedzi sieci.

12a. Reprezentacja wiedzy w systemach ekspertowych

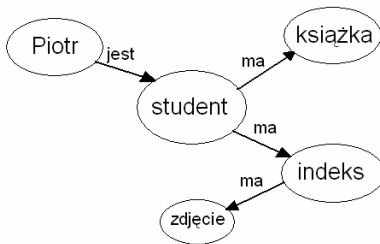
Widza = fakty + relacje + procedury

- 1) **Rachunek zdań** – każde zdanie może być T/F, zdania mogą być łączone za pomocą spójników zdaniotwórczych
- 2) **Rachunek predykatów** – rachunek zdań uzupełniony o \forall i \exists ; predykat – funkcja zdaniowa szablonowa
- 3) **Stwierdzenia**, np.
(**<OBIEKT>**, **<ATRYBUT>**, **<WARTOŚĆ>**), **<CF>** - wskaźnik pewności: [0;1] lub [-1;1]
budynek wysokość 20m
- 4) **Reguły**: *IF warunek THEN wniosek/akcja (ELSE)*

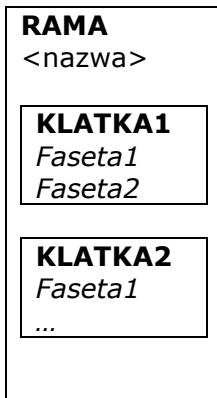
Macierze reguł:

	a	b	c	x	y	
R1	T	*	*	T	*	a – jestem przygotowany
R2	F	*	*	F	*	b – jest ładna pogoda
R3	*	T	*	*	T	c – jestem zmęczony
R4	F	*	T	F	F	x – idę na egzamin
						y – idę na wycieczkę

- 5) **Sieci semantyczne**:



- 6) **reprezentacja ramowa**:



Ramy są tworem zbliżonym pojęciowo do klasy. Jest to zbiór danych i zbiór procedur. Umożliwia ona reprezentację deklaratywną i proceduralną. Rama jest złożoną strukturą opisującą obiekt. Jest ona podzielona na klatki, klatki natomiast dzielą się na fasety. Każda rama, klatka i faset posiada unikalną nazwę. Każda klatka opisuje różnorodnie dany obiekt. Fasety przechowują wartości cech, informacje dodatkowe i sposoby działania.

- 7) **za pomocą modeli obliczeniowych**:

OBLICZ y_1, y_2 PRZY x_1, \dots, x_n ZNAJĄC m , np. *OBLICZ pole PRZY a, b, c ZNAJĄC trójkąt*

14. W jakim celu wprowadza się dane uczące i testujące podczas uczenia z nadzorem?

Najbardziej pożądaną cechą sieci jest jej zdolność do generalizacji swojej wiedzy na nowe przypadki. Tymczasem w rzeczywistości dzieje się często tak, że sieć uczona jest w sposób zapewniający minimalizację błędu wyłącznie dla zbioru uczącego, co (z wyjątkiem sytuacji, w której dostępny byłby doskonały i nieskończenie duży zbiór uczący) nie jest tym samym co minimalizacja błędu na podstawie "rzeczywistej" powierzchni błędu, a więc takiej, do wyznaczenia której niezbędna byłaby znajomość wszystkich możliwych przypadków. Dlatego oprócz danych uczących stosuje się dane testujące, które sprawdzają zdolności generalizacji sieci neuronowej, nauczonej na podstawie danych uczących.

Sieć posiada zdolność do uogólniania jeśli jest w stanie prawidłowo zaklasyfikować dane należące do zbioru testującego, po nauczaniu jej za pomocą danych ze zbioru uczącego (i ewentualnym zweryfikowaniu za pomocą danych ze zbioru sprawdzającego).

Dane testujące pozwalają wykazać, czy sieć jest nauczona odpowiednio, czy może jest przeuczona (modeluje próbki zamiast procesu) lub niedouczona (niedobór próbek nie pozwala zamodelować procesu).

Dane testujące stanowią zwykle ok. ¼ wszystkich próbek.

15. Co to jest funkcja kary? Jak działa? Wymień rodzaje.

Funkcja kary wykorzystywana jest w problemie ograniczeń algorytmów ewolucyjnych do eliminowania najgorszych osobników. Polega na *karaniu* chromosomów nie spełniających określonych ograniczeń. Zwykle realizuje się to poprzez wzbogacenie funkcji przystosowania:

$$F'(ch) = F(ch) + r \sum_{i=1}^m \phi(G_i(ch)) \quad , \text{ gdzie } r \text{ jest współczynnikiem kary, a } \phi - \text{ funkcją kary}$$

Zmodyfikowanym przypadkiem funkcji kary jest kara śmierci, czyli eliminacji słabego osobnika.

16. Na czym polega uczenie konkurencyjne?

Uczenie konkurencyjne jest jedną z podstawowych metod uczenia bez nadzoru. W tej metodzie uczenia sieci, poszczególne neurony *konkurują* ze sobą o prawo do reprezentacji danych wyjściowych - tylko jeden element wyjściowy może znajdować się w stanie aktywnym. Nazywany jest on zwycięzcą, a schemat takiej reguły aktywacji neuronów określany jest mianem Winner Takes All - **WTA**. Algorytmy WTA, w których tylko jeden neuron może podlegać adaptacji w każdej iteracji, są jednak algorytmami słabo zbieżnymi, szczególnie przy dużej liczbie neuronów. W praktyce zostały one zastąpione algorytmami **WTM** (ang. Winner Takes Most), w których oprócz zwycięzcy uaktualniają swoje wagi również neurony z jego sąsiedztwa.

$$\Delta w = \eta(k) * S(i) * (U^{(p)} - w) \quad S(i) - \text{ sąsiedztwo, np. prostokątne, Gaussa}$$

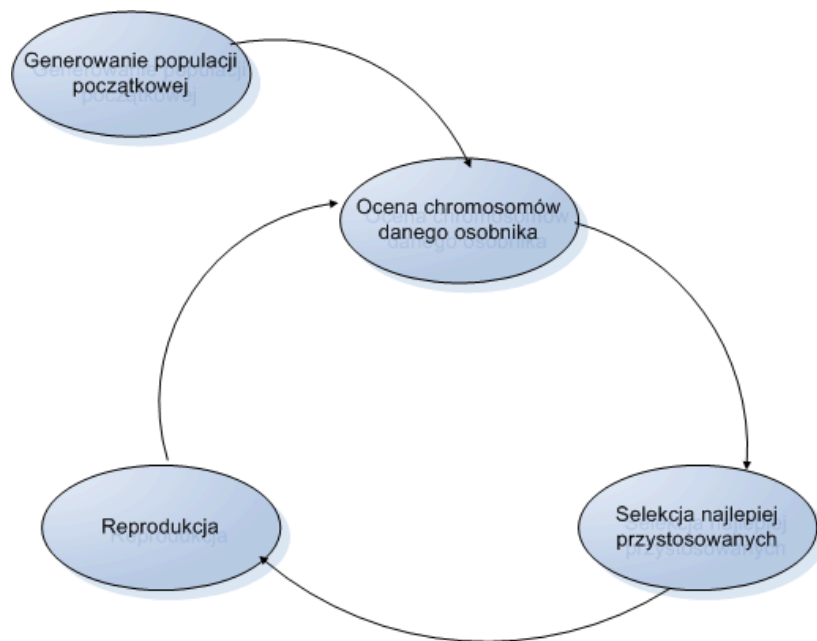
Innym podejściem jest algorytm Kohonena - jest to metoda, w której, w odróżnieniu od reguły standardowej, modyfikacji wag dokonuje się nie tylko dla neuronu zwycięskiego, lecz również dla pewnej liczby neuronów z jego otoczenia. Reguła ta wymaga wprowadzenia tzw. topologicznego uporządkowania neuronów w warstwie wyjściowej (zwanej też mapą cech), dla której można wprowadzić pojęcie otoczenia neuronu zwycięskiego. W klasycznej sieci Kohonena neurony są logicznie rozmieszczone, sąsiedztwo liczone jest w logicznej strukturze.

17. Przedstawić algorytm genetyczny.

Algorytmy genetyczne są algorytmami, starającymi się naśladować ewolucję.

- Ewolucja jest procesem działającym na chromosomach (nie na osobnikach – właścicielach tych chromosomów).
- W toku naturalnej selekcji osobniki lepsze (a tym samym związane z nimi chromosomy) mają większą szansę na reprodukcję.
- Reprodukacja jest najważniejszym momentem ewolucyjnym - tutaj występuje krzyżowanie lub mutacja.
- Całość wiedzy o populacji jest skumulowana w chromosomach (ewolucja nie posiada pamięci).

Sam algorytm przedstawia się następująco:



Warunkiem stopu dla algorytmu może być:

- ilość iteracji
- brak zmian chromosomów
- brak zmian w optimum

Podczas oceny chromosomów oblicza się wartość funkcji przystosowania dla każdego osobnika. Selekcja może się odbywać poprzez koło ruletki, metodą elitarną, stochastyczną, itd. (vide 2.). Reprodukacja to odpowiednie krzyżowanie i mutacja genów.

18. Neuron z nieliniową funkcją aktywacji (?)

Przykłady nieliniowych funkcji aktywacji w neuronach:

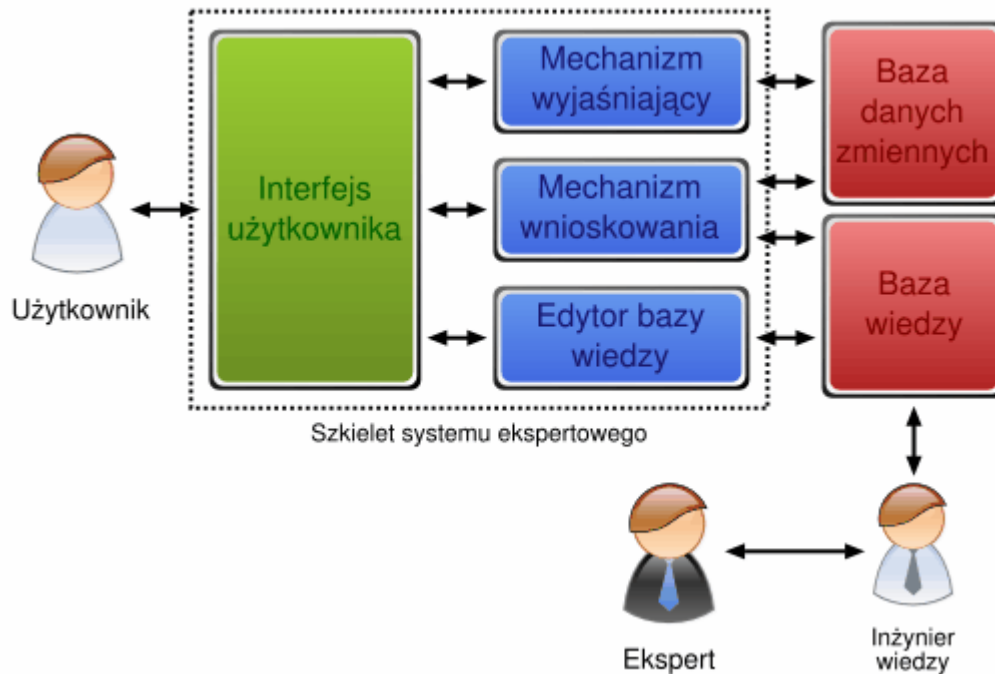
Funkcja sigmoidalna:

$$F(\varphi) = \frac{1}{1 + e^{-\beta(\varphi - \varphi_n)}}$$

Tangens hiperboliczny:

$$F(\varphi) = \frac{1 - e^{-\beta(\varphi - \varphi_n)}}{1 + e^{-\beta(\varphi - \varphi_n)}} = \operatorname{tgh}(\varphi)$$

19. Budowa systemu ekspertowego



Składniki systemu ekspertowego to:

- Szkielet systemu składający się z
 - Interfejsu użytkownika. Użytkownik korzysta z systemu komunikując się z nim za pomocą interfejsu użytkownika. Sprowadza się to najczęściej do zadawania pytań, udzielania informacji systemowi, oraz odbierania od systemu odpowiedzi i wyjaśnień.
 - Edytora bazy wiedzy. Dzięki wbudowanemu edytorowi możliwa jest modyfikacja wiedzy zawartej w systemie, co pozwala na rozbudowę systemu.
 - Mechanizmu wnioskowania. Jest to najważniejszy składnik systemu ekspertowego, jego zadaniem jest wyciąganie wniosków z przesłanek i pytań wprowadzanych przez użytkownika i generowanie odpowiedzi.
 - Mechanizmu wyjaśniającego. Mechanizm ten umożliwia wyjaśnienie na życzenie użytkownika dlaczego system udzielił takiej, a nie innej odpowiedzi, albo dlaczego system zadał użytkownikowi określone pytanie.
- Baza wiedzy. Jest to drugi pod względem ważności składnik systemu. W bazie wiedzy zawarta jest wyekstrahowana od ludzkich ekspertów wiedza dotycząca określonej dziedziny. Wiedza ta zwykle zapisana jest za pomocą wybranego sposobu reprezentacji wiedzy, na przykład za pomocą reguł lub ram.
- Baza danych zmiennych. Jest to pomocnicza baza danych w której przechowywane są wnioski uzyskane przez system podczas jego działania. Baza ta umożliwia odtworzenie sposobu wnioskowania systemu i przedstawienie go użytkownikowi za pomocą mechanizmu wyjaśniającego.

Ekstrakcją wiedzy od ekspertów zajmują się na ogół inżynierowie wiedzy. Jest to zwykle długi i żmudny proces, ponieważ wiedza stosowana przez ludzkich ekspertów jest zwykle wiedzą praktyczną i intuicyjną.