**PSI SCENARIUSZ 5**

**Anna Ruchała, IS III, GR 3**

Cel: Celem ćwiczenia jest poznanie budowy i działania sieci Kohonena przy wykorzystaniu reguły

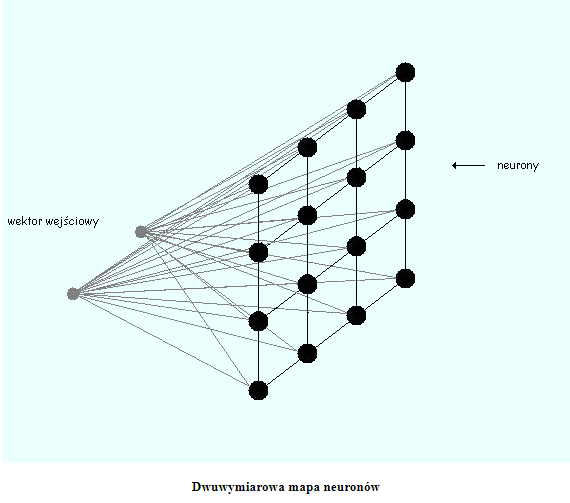
WTA do odwzorowywania istotnych cech kwiatów.

Sieci Kohonena są jednym z podstawowych typów sieci samoorganizujących się. Właśnie dzięki zdolności samoorganizacji otwierają się zupełnie nowe możliwości - adaptacja do wcześniej nieznanych danych wejściowych, o których bardzo niewiele wiadomo. Wydaje się to naturalnym sposobem uczenia, który jest używany chociażby w naszych mózgach, którym nikt nie definiuje żadnych wzorców, tylko muszą się one krystalizować w trakcie procesu uczenia, połączonego z normalnym funkcjonowaniem. Sieci Kohonena stanowią synonim całej grupy sieci, w których uczenie odbywa się metodą samoorganizującą typu konkurencyjnego. Polega ona na podawaniu na wejścia sieci sygnałów, a następnie wybraniu w drodze konkurencji zwycięskiego neuronu, który najlepiej odpowiada wektorowi wejściowemu. Dokładny schemat konkurencji i późniejszej modyfikacji wag synaptycznych może mieć różną postać. Wyróżnia się wiele podtypów sieci opartych na konkurencji, które różnią się dokładnym algorytmem samoorganizacji.

Bardzo istotną kwestią jest struktura sieci neuronowej. Pojedynczy neuron jest mechanizmem bardzo prostym i przez to niewiele potrafiącym. Dopiero połączenie wielu neuronów ze sobą umożliwia prowadzenie dowolnie skomplikowanych operacji. Ze względu na raczej niewielką wiedzę o faktycznych zasadach funkcjonowania ludzkiego mózgu, powstało wiele różnych architektur, które starają się naśladować budowę i zachowanie poszczególnych fragmentów układu nerwowego. Najczęściej stosuje się w tego typu sieciach architekturę jednokierunkową jednowarstwową. Jest to podyktowane faktem, że wszystkie neurony muszą uczestniczyć w konkurencji na równych prawach. Dlatego każdy z nich musi mieć tyle wejść ile jest wejść całego systemu.

Zasady działania sieci Kohonena:

* Wejścia (tyle, iloma parametrami opisano obiekty) połączone są ze wszystkimi węzłami sieci
* Każdy węzeł przechowuje wektor wag o wymiarze identycznym z wektorami wejściowymi
* Każdy węzeł oblicza swój poziom aktywacji jako iloczyn skalarny wektora wag i wektora wejściowego (podobnie jak w zwykłym neuronie)
* Ten węzeł, który dla danego wektora wejściowego ma najwyższy poziom aktywacji, zostaje zwycięzcą i jest uaktywniony
* Wzmacniamy podobieństwo węzła-zwycięzcy do aktualnych danych wejściowych poprzez dodanie do wektora wag wektora wejściowego (z pewnym współczynnikiem uczenia)
* Każdy węzeł może być stowarzyszony z pewnymi innymi, sąsiednimi węzłami - wówczas te węzły również zostają zmodyfikowane, jednak w mniejszym stopniu.
* Inicjalizacja wag sieci Kohonena jest losowa. Wektory wejściowe stanowią próbę uczącą, podobnie jak w przypadku zwykłych sieci rozpatrywaną w pętli podczas budowy mapy. Wykorzystanie utworzonej w ten sposób mapy polega na tym, że zbiór obiektów umieszczamy na wejściu sieci i obserwujemy, które węzły sieci się uaktywniają. Obiekty podobne powinny trafiać w podobne miejsca mapy.



Nauka z algorytmem Winner Takes All:

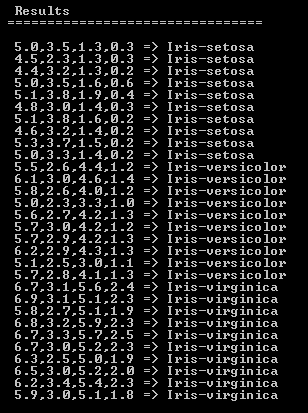
1. inicjalizacja wag sieci
2. wzór na obliczenie odległości wektora wejściowego do wag każdego z neuronów:
3. d = sqrt((w1 – x1)2 + (w2 – x2)2)
4. wybranie neuronu zwycięzcy (wygrywającego) dla którego odległość wag od wektora wejściowego jest najmniejsza (na podstawie Euklidesowej miary odległości).
5. zmiana wartości poszczególnych wag tego neuronu przyjmując, że szybkość nauki wynosi n = 0,1
6. powtórzenie kroków 2-5 dla wszystkich przykładów uczących.

Algorytmy WTA, w których tylko jeden neuron może podlegać adaptacji w każdej iteracji, są 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.

Moje dane testujące znajdują się w pliku Program.cs, w którym każdy wers określa jedną daną uczącą. Jeden rekord składa się z 4 liczb oddzielonych przecinkami określającymi układ kwiatów irysa oraz na końcu nazwa odpowiadająca temu zestawu. Np.: 5.1,3.5,1.4,0.2,Iris-setosa

Natomiast dane testujące znajdują się w pliku dane\_test.txt, który posiada podobną strukturę, lecz nie zawiera nazwy kwiatu.

Wydruk wyników programu:



Wnioski:

W sieciach samoorganizujących się stosowany jest algorytm uczenia, nazywany „uczeniem konkurencyjnym” (competitive learning). Oznacza to, że po prezentacji wzorca wejściowego (wektora uczącego x) nie wszystkie neurony, jak to występuje w innych typach sieci, modyfikują swoje wagi. Tutaj neurony „konkurują” ze sobą, by zostać neuronem zwycięskim. Zwycięzcą zostaje ten, którego wektor wag jest najbardziej zbliżony (ma najmniejszą odległość) do prezentowanego wzorca wejściowego. Tylko neuron wygrywający oraz ewentualnie neurony znajdujące się w jego są- siedztwie mają możliwość uaktualnienia swoich wag, tak by jeszcze bardziej zbliżyć je do podanego właśnie wektora uczącego. Gdy tylko neuron zwycięski zmienia swoje wagi, mamy do czynienia z najbardziej skrajną formą uczenia konkurencyjnego, zwaną „zwycięzca bierze wszystko”.

Przy równomiernym rozkładzie próbek uczących (np. na płaszczyźnie) i jednakowo częstym ich prezentowaniu, neurony-zwycięzcy również rozmieszczeni są równomiernie. Jeśli pewne próbki są uprzywilejowane (prezentowane częściej na wejściu), wówczas zwycięzcy lokowani są nierównomiernie, gęściej w okolicy odpowiadającej próbkom prezentowanym częściej. W ten sposób sieć Kohonena może odzwierciedlać nie tylko rozkład przestrzenny próbek wejściowych, ale również częstotliwość ich prezentowania w procesie uczenia.

W wyniku zastosowanego trybu uczenia poszczególne neurony sieci otrzymują wagi, które mogą być zinterpretowane w przestrzeni, w identyczny sposób jak współrzędne wektorów wejściowych. Rozmieszczenie ich w przestrzeni tworzy węzły, które połączone z najbliższymi sąsiadami (w sensie sąsiedztwa topologicznego) reprezentują kształt odtwarzający rozkład danych wejściowych. Jest to rozkład odpowiadający gęstości pojawiania się danych wejściowych w danym rejonie a nie współrzędnych odpowiednich wektorów, gdyż liczba prezentowanych wektorów wejściowych jest zwykle o kilka rzędów wyższa niż liczba neuronów w sieci. Liczba neuronów w sieci jest w praktyce zwykle większa od liczby klas sygnałów wejściowych. Może być więc kilka neuronów konkurujących o reprezentację danej klasy. Odpadając z konkurencji próbują one przełączać się na inna klasę, co objawia się chaotycznymi zmianami wartości ich wag.

Uczenie można by przerwać w momencie, gdy zmiany wektorów wag stają się nieznaczne. Łatwo jest to sprawdzać wtedy, gdy poprawki są kumulowane dla całego zbioru uczącego - wtedy badamy długość wektora największej poprawki dla każdego neuronu. Jeżeli staje się ona bardzo mała, to możemy być pewni, że zbieżność jest zadowalająca. Inaczej jest, gdy adaptacja wektorów wag neuronów odbywa się po każdym podaniu wektora uczącego. Nawet wówczas, gdy zbieżność wydaje się być zadowalająca, jeden przypadek może odepchnąć wagi, podczas gdy następny popchnie je z powrotem. Mały współczynnik uczenia może załagodzić te wahania, ale jednocześnie może zamaskować brak zbieżności. Tak więc trzeba być bardzo ostrożnym przy ocenie zbieżności sieci, gdy stosuje się mały współczynnik uczenia. Poprawki mogą być wtedy małe z powodu małego współczynnika uczenia.