**PSI SCENARIUSZ 6**

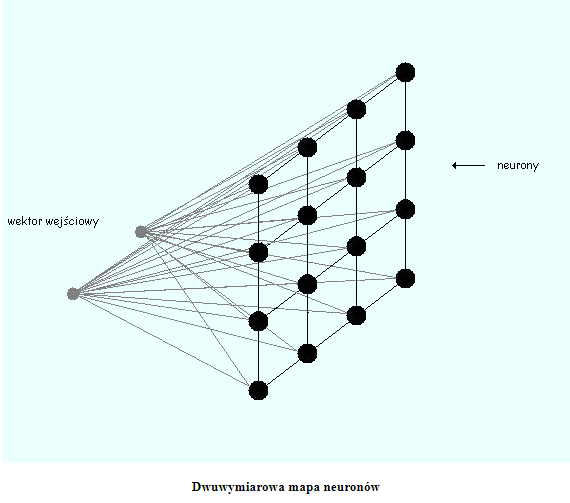
**Anna Ruchała, IS III, GR 3**

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

Winner Takes Most do odwzorowywania istotnych cech liter alfabetu.

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 WTM:

* inicjalizacja wag sieci
* wzór na obliczenie odległości wektora wejściowego do wag każdego z neuronów:

d = sqrt((w1 – x1)2 + (w2 – x2)2)

* wybranie neuronu zwycięzcy (wygrywającego) dla którego odległość wag od wektora wejściowego jest najmniejsza (na podstawie Euklidesowej miary odległości).
* Uaktualnienie wagi neuronów sąsiedztwa wg wzoru

https://lh4.googleusercontent.com/paorEVrFrbavnLvTHbIQkYOvE3MUSD-z5IgVkuhqmBoo3S7Pxv2z7V6IStXMwCx8d0GLj2DNDNA3m-iJ3yPMN_NpnQ5dAlZbRcOvCbkoGHkl92TGeC4J3GRv2fbr2Hy5xA

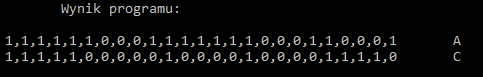
* zmiana wartości poszczególnych wag zwycięskiego neuronu przyjmując, że szybkość nauki wynosi n = 0,1
* powtórzenie kroków 2-5 dla wszystkich przykładów uczących.

Moje dane testujące znajdują się w pliku: dane.txt, w którym każdy wers określa jedną daną uczącą. Jeden rekord składa się z 25 liczb oddzielonych przecinkami określającymi jedną literkę w postaci tablicy 5x5 oraz na końcu literka odpowiadająca temu zestawu. „1” oznacza że w danym punkcie istnieje zawartość, natomiast wartość „0” oznacza, że pole jest puste.

Np.: 1,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1,0,0,0,1,1,0,0,0,1

Wynik programu:

Na podstawie danych testujących z pliku dane.txt program bada jaka literka odpowiada podanemu ciągowi liczb 0 i 1.



Wnioski:

Istotną rolę w sieci Kohonena odgrywa „sąsiedztwo”. 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 sąsiedztwie 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”.

Ważnym elementem dla zasady WTM jest określenie tzw. sąsiedztwa. Jest ono rozumiane w sensie geometrycznym jako położenie neuronu względem zwycięzcy. Zasięg sąsiedztwa określany jest jako tzw. promień sąsiedztwa, wraz ze zwiększaniem wartości współczynnika uczenia zmniejsza się liczba iteracji, co ma wpływ na działanie programu i powoduje uzyskanie błędnych wyników. Aby zredukować liczbę błędnych wyników można zwiększyć liczbę iteracji.

Dobór wag ma bezpośredni wpływ na działanie perceptronów w sieci, w zależności od ich wartości zmienia się poprawność wyników i liczba iteracji. Wartości wag mają największy wpływ na efekt końcowy, często wagi są ustalane losowo. Dane uczące wpływają na poprawność uczenia perceptronu, przy zbyt małej ilości otrzymujemy błędne wyniki. Aby uzyskać lepsze wyniki powinniśmy dostarczyć wystarczającą liczbę danych wejściowych.