**Urszula Król,**

**grupa projektowa 2**

**IS WIMiIP**

**Sprawozdanie nr 2**

**1. Temat ćwiczenia:**

Budowa i działanie sieci jednowarstwowej.

**2. Cel ćwiczenia:**

Celem ćwiczenia jest poznanie budowy i działania jednowarstwowych sieci neuronowych oraz uczenie rozpoznawania wielkości liter.

**3. Zadania wykonane w ramach ćwiczenia:**

1. Wygenerowanie danych uczących, zawierających 10 dużych i 10 małych liter dowolnie wybranego alfabetu w postaci dwuwymiarowej tablicy np. 5x7 pikseli dla jednej litery.

2. Przygotowanie (implementacja) jednowarstwowej sieci według algorytmu Adaline.

3. Uczenie sieci dla różnych współczynników uczenia.

4.​ Testowanie sieci dla danych testowych zawierających 20 różnych liter: 11 dużych i 9 małych.

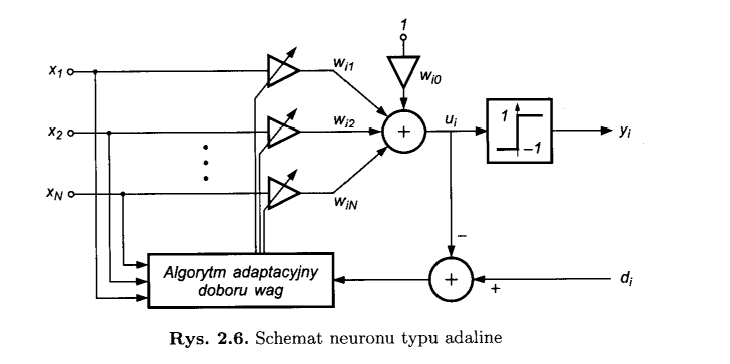
**4. ​Syntetyczny opis budowy oraz wykorzystanych sieci i algorytmów uczenia:**

Sieć neuronowa – to układ neuronów odpowiednio ze sobą połączonych w warstwy, przy czym wyróżnia się warstwę wejściową, warstwy ukryte i warstwę wyjściową:

****

Budowa neuronu Adaline jest bardzo podobna do modelu perceptronu. Różnica dotyczy algorytmu uczenia. Sygnał wyznacza się w ten sam sposób, co w przypadku uczenia perceptronu. Jednak w przypadku neuronu typu Adaline porównuje się sygnał wzorcowy z sygnałem na wyjściu części liniowej neuronu.

Schemat neuronu przedstawia poniższy rysunek:



Realizacja programu:

- określenie losowych wag z zakresu od 0 do 1,

- wyliczenie sumy wagowej sygnałów wejściowych,

- wyliczenie zmiennej *delta*, czyli różnicy między wartością oczekiwaną, a otrzymaną,

- sposób korekty wag:

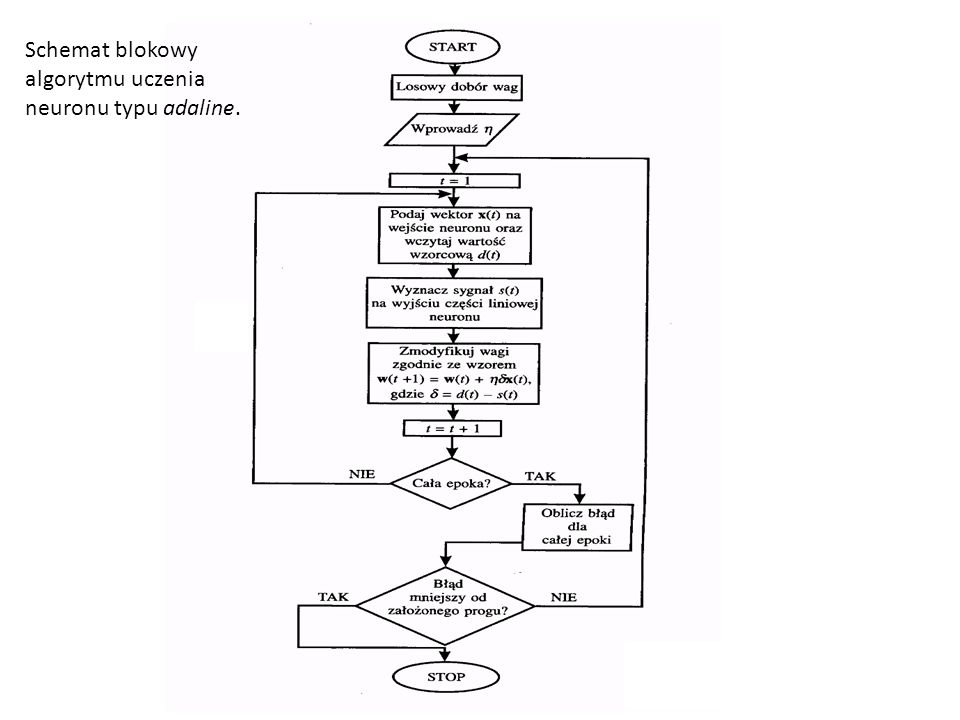


η – współczynnik uczenia się

- uczenie neuronu sprowadza się do minimalizacji funkcji określonej w sposób następujący:

C:\Users\Ula\Desktop\wzor.PNG

Algorytm uczenia neuronu typu Adaline przedstawiony jest w postaci schematu blokowego poniżej:



**5. ​Zestawienie otrzymanych wyników:**

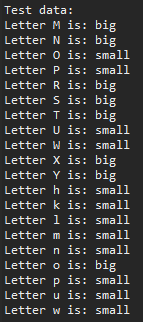
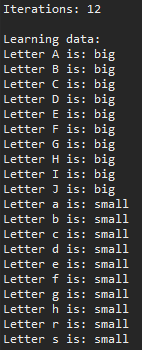
Tabela zależności liczby iteracji i liczby błędów od wartości współczynnika uczenia:

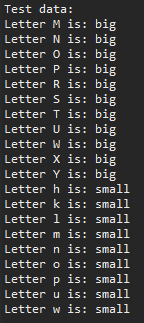
|  |  |  |  |
| --- | --- | --- | --- |
| Lp. | Współczynnik uczenia | Iteracje | Liczba błędnych wyników |
| 1 | 0.1 | 12 | 5 |
| 2 | 0.01 | 48 | 4 |
| 3 | 0.001 | 376 | 2 |
| 4 | 0.0001 | 22633 | 0 |
| 5 | 0.00001 | 107734 | 1 |
| 6 | 0.000001 | 204389 | 4 |

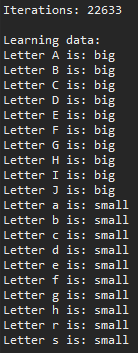
Wykres zależności liczby iteracji od współczynnika uczenia:

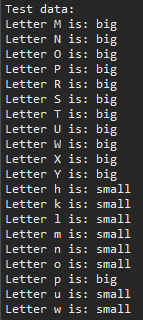
Wykres zależności liczby błędnych wyników testu od współczynnika uczenia:

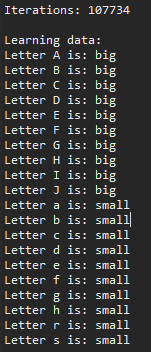
**6. Przykładowe zrzuty ekranu z działania programu:**











**7.​ Analiza i dyskusja błędów uczenia i testowania opracowanych sieci w zależności od wartości współczynnika uczenia:**

- na podstawie utworzonego wykresu zależności ilości iteracji od wartości współczynnika uczenia możemy zauważyć, że im mniejszy współczynnik uczenia tym większa jest liczba epok uczenia oraz wydłuża się czas działania programu,

- wykres zależności liczby błędnych wyników testu od współczynnika uczenia pozwala dostrzec, że poprawność otrzymanych wyników zależy od współczynnika uczenia,

- najmniej błędów uzyskano przy współczynniku uczenia równym 0.0001, w miarę oddalania się od tej wartości liczba błędów wzrastała,

- im większa jest wartość dopuszczalnego błędu, tym mniejszy jest procent poprawnych wyników,

- proces uczenia trwa do momentu aż dla wszystkich danych wejściowych błąd będzie mniejszy od założonego progu błędu uczenia.

**8. Wnioski:**

- budowa neuronu Adaline jest bardzo podobna do modelu perceptronu, a jedyna różnica dotyczy algorytmu uczenia,

- na działanie sieci mają wpływ czynniki takie jak: współczynnik uczenia, liczba danych uczących, funkcja aktywacji a także metoda uczenia,

- uczenie neuronu sprowadza się do minimalizacji funkcji błędu średniokwadratowego,

- wartość współczynnika uczenia znacznie wpływa na poprawność uzyskanych wyników,

- odpowiednie dobranie współczynnika uczenia ma kluczowe znaczenie w prawidłowości działania programu i jego wydajności,

- właściwy dobór struktury sieci wpływa na dokładność działania i czas uczenia.

**9. Listing kodu programu:**

Pełny listing kodu programu znajduje się w repozytorium: <https://github.com/ukrol/PSI_GP02_zima_2017-2018_Urszula_Krol>