Sprawozdanie Scenariusz 1 Aleksandra Karaś

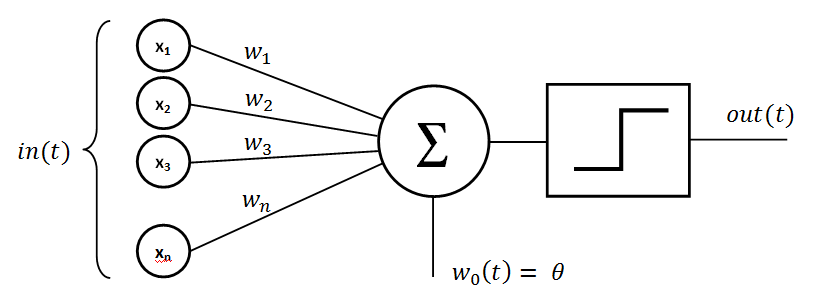
Budowa i działanie perceptronu

Cel ćwiczenia:

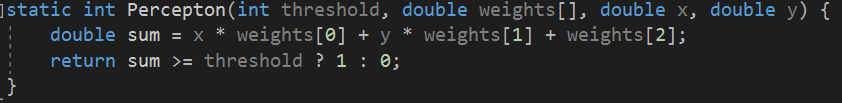
Głównym celem ćwiczenia było poznanie budowy i działania perceptronu poprzez implementację oraz uczenie perceptronu realizującego wybraną funkcję logiczną dwóch zmiennych.

Implementacja perceptronu:

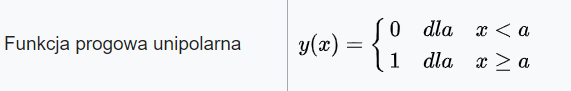
Implementację w programie wykonałam na podstawie niżej załączonego obrazku



Kod:



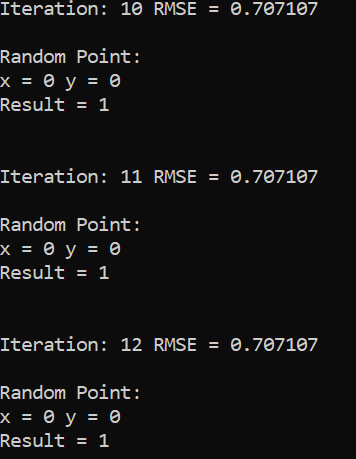
W implementacji użyłam jako funkcji aktywacji funkcję unipolarnej. Początkowa wartość progowa wynosiła a=0 ( w moim programie threshold = 0).



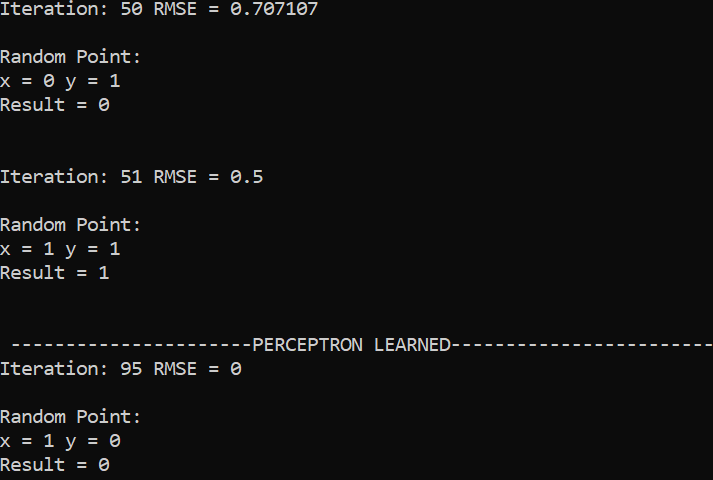
Implementacja algorytmu uczenia:

1. Zadeklarowanie współczynnika uczenia się, liczby danych oraz wartości progowej
2. Wprowadzenie danych wejściowych, wyjściowych (zgodnie z bramką logiczną AND)
3. Inicjacja wag, liczbami losowymi bliskimi 0
4. Rozpoczęcie nauczania:
   1. Wyznaczenie sumy iloczynu skalarnego wi\*xi oraz sprawdzenie nierówności suma> thereshold -funkcja Percepton.
   2. Wyznaczenie błędu lokalnego, globalnego i średniej kwadratowej błędów - średniej różnicy między estymatorem a wartością estymowaną.
   3. Wyznaczenie nowych wag, poprzez dodanie iloczynu błędu lokalnego, danych wejściowych i współczynnika uczenia się.
   4. Porównanie wartości oczekiwanej z wartością perceptronu lub sprawdzenie czy nie skończyły się dane uczące.
5. Po nauczeniu RMSE = 0, sprawdzenie jeszcze 5 losowych punktów

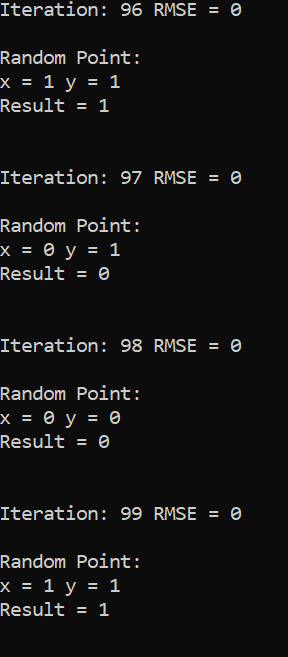
Przykład działania:



Strzały perceptronu w trakcie nauki. Widzimy wysoką średnią kwadratową błędu i błędne wyniki bramki logicznej AND

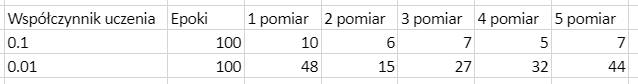


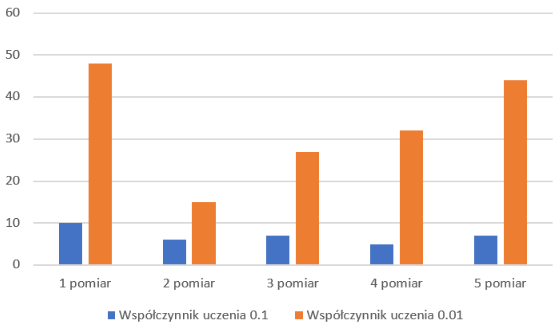
W wyżej załączonym zdjęciu widzimy jak średnia kwadratowa błędu maleje, aż staje się równa 0, co świadczy o tym, że perceptron został nauczony. Następnie generujemy losowe punkty w celu sprawdzenia poprawności. Widzimy, że następna iteracja wykonała się poprawnie.

Przy kolejnych kontrolnych próbach widzimy ze perceptron bezbłędnie podaje wyniki bramki logicznej AND. Średnia kwadratowa błędu równa się w każdym wypadku 0.

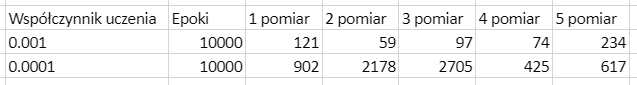
Wyniki przeprowadzonego ćwiczenia:

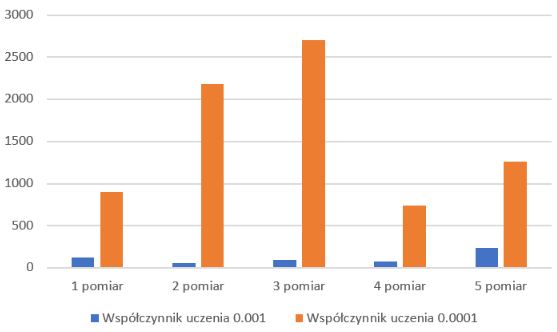
W zależności od przeprowadzonego testu, zmieniałam współczynnik uczenia. Każdy test zawierał 5 prób. Na podstawie tych prób powstały wykresy zależności wartości współczynnika uczenia a ilość iteracji potrzebnych do nauczenia prawidłowych odpowiedzi na zagadnienie bramki logicznej AND.





W porównaniu współczynników uczenia się 0.1 i 0.01 widać, zmniejszenie tego współczynnika dziesięciokrotnie powoduje, że potrzeba znacznie więcej iteracji by z sukcesem nauczyć perceptron swojego zadania. Różnica w liczbie iteracji wynosi około od 2 – 4 razy.





Kolejne zmniejszenie współczynnika, pokazuje nam, że liczba iteracji diametralnie się zmienia. W niektórych przypadkach jest nawet kilkadziesiąt razy większa.

Niestety nie mogliśmy porównać współczynnika uczenia 0.0001 oraz 0.001 z liczbą iteracji jak przy współczynnikach 0.1 i 0.01 ze względu na to, że uczenie byłoby tak wolne, że by nie nastąpiło. Zatem w przypadku mniejszego współczynnika uczenia, musieliśmy zwiększyć ilość iteracji.

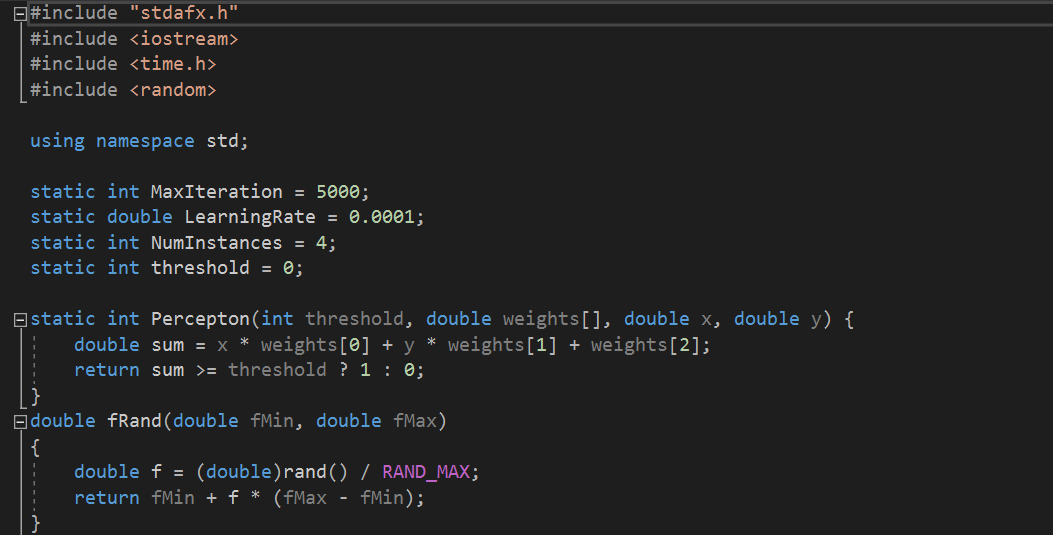
Wnioski oraz analiza:

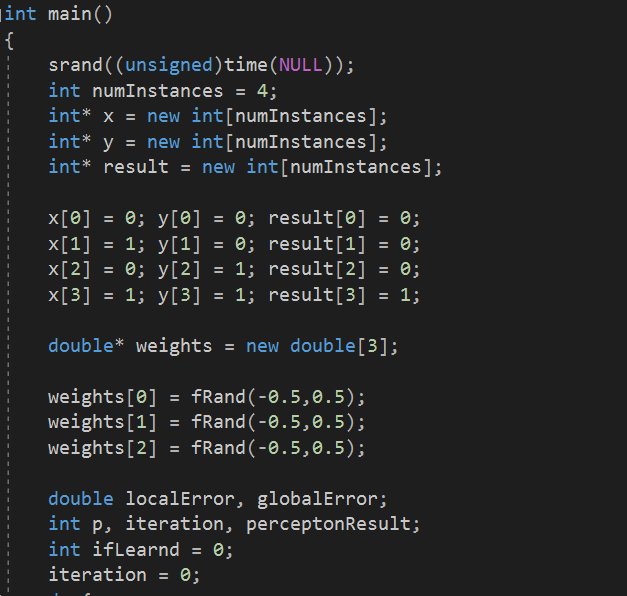
W narzędziu jakim jest perceptron, musimy zwrócić szczególną uwagę na uważne dobieranie liczby epok, które w zależności od współczynnika uczenia może mieć duże wahania. Gdy zostanie zadanie zbyt mała liczba, może nie dojść do nauczenia się perceptronu i nie będzie w stanie poprawnie wykonywać zadania.

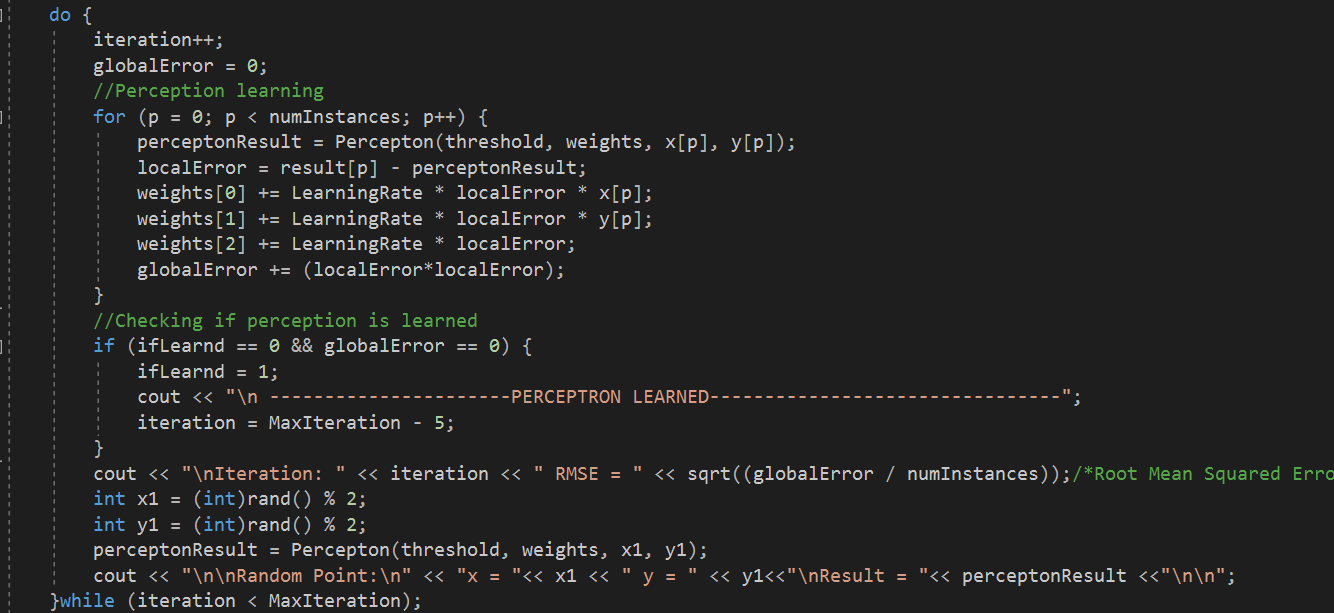
Kolejną rzeczą, na którą trzeba zwrócić uwagę jest dobieranie odpowiedniego współczynnika uczenia. To właśnie za jego sprawą, możemy kontrolować, jak szybko perceptron powinien się uczyć i odnieść sukces w uczeniu. Musimy jednak pamiętać, że współczynnik uczenia ma bezpośredni wpływ na liczbę epok.

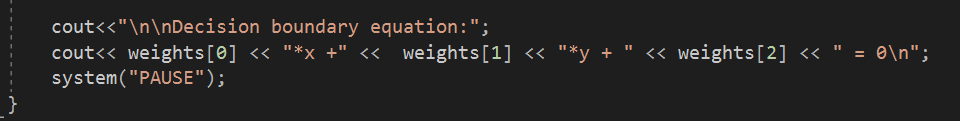
Przedstawiony w zadaniu perceptron jednowarstwowy jest bardzo skutecznym narzędziem do klasyfikowania zbiorów liniowo separowalnych. Oznacza to, że uniemożliwia na przykład wytrenowanie złożonego z jednego neuronu perceptronu, który wykonywałby logiczną operację [XOR](https://pl.wikipedia.org/wiki/Alternatywa_wykluczaj%C4%85ca) na wartościach wejść. W takim przypadku trzeba zbudować sieć z więcej niż jednego nautrona.

Listing kodu:









Materiały źródłowe:

<https://pl.wikipedia.org/wiki/Perceptron>

<http://www.algorytm.org/sztuczna-inteligencja/sztuczny-neuron.html>

<https://www.ii.uni.wroc.pl/~aba/teach/NN/w4.pdf>

<http://edu.pjwstk.edu.pl/wyklady/nai/scb/wyklad3/w3.htm>

<http://galaxy.uci.agh.edu.pl/~vlsi/AI/wstep/>

http://www.cs.put.poznan.pl/rklaus/assn/percep.htm