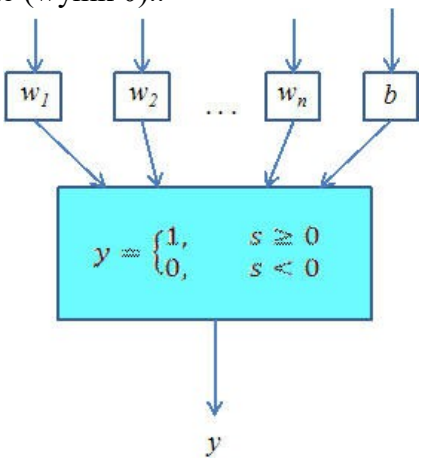
Aleksandra Mendrek, gr. 2

Sprawozdanie z ćwiczenia:
Budowa i działanie perceptronu

1. Cel ćwiczenia
   Celem ćwiczenia jest poznanie budowy i działanie perceptronu poprzez implementację oraz uczenie perceptronu realizującego wybraną funkcję logiczną dwóch zmiennych.
2. Wstęp teoretyczny
   Perceptron - prosty element obliczeniowy, który sumuje ważone sygnały wejściowe i porównuje tę sumę z progiem aktywacji. W zależności od wyniku perceptron może być albo wzbudzony (wynik 1), albo nie (wynik 0)..



Celem perceptronu jest rozwiązanie danego problemu na podstawie otrzymanych danych na wejściach (w1...wn) wraz z wagami odpowiadające danemu wejściu. Perceptron wylicza sumę danych przemnożonych przez odpowiednie wagi i przekazuje ją do funkcji aktywacji, która określa czy badany problem wystąpił(1) czy nie(0).

Aby nauczyć nasz perceptron kiedy problem zachodzi a kiedy nie należy dostarczyć mu zestaw uczący takich danych wraz z odpowiadającym im wynikiem. Dzięki algorytmowi uczenia nasz perceptron potrafi w taki sposób dobrać wagi na podstawie analizy danych uczących, aby moc określić wystąpienie problemu dla dowolnych danych.

Aby tego dokonać należy użyć algorytmu uczenia perceptronu - tzn. automatycznego doboru wag na podstawie napływających przykładów. W uproszczonym (ze względu na ilustrację) przypadku dwuwymiarowym algorytm wygląda następująco:

*Inicjujemy wagi losowo.
*Dla każdego przykładu uczącego obliczamy odpowiedź perceptronu.
*Jeśli odpowiedź perceptronu jest nieprawidłowa, to modyfikujemy wagi:

$$w1 += n * (d-y) * x1$$
$$w2 += n * (d-y) * x2$$
$$b += n * (d-y)$$

gdzie n jest niewielkim współczynnikiem uczenia (n > 0), d - oczekiwana odpowiedź a y - odpowiedź neuronu.

Po wyczerpaniu przykładów zaczynamy proces uczenia od początku, dopóki następują jakiekolwiek zmiany wag połączeń. Opisany schemat jest w miarę przejrzysty tylko dla pojedynczych perceptronów lub niewielkich sieci. Ciężko jest stosować reguły

tego typu dla bardziej skomplikowanych modeli.

Algorytm uczenia perceptronu:
- Zadeklarowanie współczynnika uczenia, liczby danych oraz wartości granicznej.
- Wprowadzenie danych wejściowych oraz wyjściowych (w tym przypadku dane zgodne z bramką logiczną OR).
- Inicjalizacja wag liczbami rzeczywistymi (w tym przypadku zakres -1 do 1).
- Rozpoczęcie cyklu poniższych instrukcji do momentu uzyskania satysfakcjonującego wyniku, jakim będzie nauczenie perceptronu:
- Wykonanie tak zwanych obliczeń wartości granicznej. Można je sprowadzić do wyznaczania sumy iloczynu skalarnego wi*xi i sprawdzenie nierówności w*x >= θ;
- Wyznaczenie błędu lokalnego, globalnego oraz średniej kwadratowej błędów, czyli średniej różnicy pomiędzy estymatorem i wartością estymowaną;
- Przypisanie nowych wag, poprzez dodanie iloczynu błędu lokalnego, danych wejściowych oraz współczynnika uczenia.
- Skonfrontowanie wyniku z wartościami oczekiwanymi, bądź sprawdzenie czy nie skończyły się dane uczące.
- Po poprawnym nauczeniu (root mean square error = 0), sprawdzenie na 20 kolejnych próbach poprawności działania perceptronu.

3. Działanie i wykonanie:
Próba I:

```
F:\Programowanie\Java\jdk1.8.0_121\bin\java ...
Wagi: -0.36653911726527943, 0.5461082012017668
Numer iteracji: 1 : RMSE = 0.8660254037844386

=======
Kolejny punkt:
x = 1,y = 1
Obliczony wynik = 0
Wagi: -0.3465391172652794, 0.5661082012017669
Numer iteracji: 2 : RMSE = 0.8660254037844386

=======
Kolejny punkt:
x = 0,y = 1
Obliczony wynik = 0
Wagi: -0.3265391172652794, 0.5861082012017669
Numer iteracji: 3 : RMSE = 0.8660254037844386

=======
Kolejny punkt:
x = 1,y = 1
Obliczony wynik = 0
Wagi: -0.3065391172652794, 0.6061082012017669
Numer iteracji: 4 : RMSE = 0.8660254037844386

=======
Kolejny punkt:
x = 1,y = 0
Obliczony wynik = 0
Wagi: -0.28653911726527936, 0.6261082012017669
Numer iteracji: 5 : RMSE = 0.8660254037844386

=======
Kolejny punkt:
x = 0,y = 1
Obliczony wynik = 0
```

```
=======
Kolejny punkt:
x = 1,y = 1
Obliczony wynik = 1
Wagi: 0.24346088273472083, 0.756108201201767
Numer iteracji: 50 : RMSE = 0.5

=======
Kolejny punkt:
x = 1,y = 0
Obliczony wynik = 1
Wagi: 0.25346088273472084, 0.756108201201767
---------------------------NAUCZONY-------------------------------
Numer iteracji: 480 : RMSE = 0.0

=======
Kolejny punkt:
x = 0,y = 0
Obliczony wynik = 0
Wagi: 0.25346088273472084, 0.756108201201767
Numer iteracji: 481 : RMSE = 0.0

=======
Kolejny punkt:
x = 0,y = 1
Obliczony wynik = 1
Wagi: 0.25346088273472084, 0.756108201201767
Numer iteracji: 482 : RMSE = 0.0
```

Próba II:

```
F:\Programowanie\Java\jdk1.8.0_121\bin\java ...
Wagi: -0.5423456335974091, -0.39349047922231706
Numer iteracji: 1 : RMSE = 0.8660254037844386

=======
Kolejny punkt:
x = 1,y = 1
Obliczony wynik = 0
Wagi: -0.522345633597409, -0.37349047922231704
Numer iteracji: 2 : RMSE = 0.8660254037844386

=======
Kolejny punkt:
x = 0,y = 0
Obliczony wynik = 0
Wagi: -0.502345633597409, -0.353490479222317
Numer iteracji: 3 : RMSE = 0.8660254037844386

=======
Kolejny punkt:
x = 1,y = 0
Obliczony wynik = 0
Wagi: -0.482345633597409, -0.333490479222317
Numer iteracji: 4 : RMSE = 0.8660254037844386

=======
Kolejny punkt:
x = 0,y = 0
Obliczony wynik = 0
Wagi: -0.462345633597409, -0.313490479222317
Numer iteracji: 5 : RMSE = 0.8660254037844386

=======
Kolejny punkt:
x = 1,y = 0
Obliczony wynik = 0
Wagi: -0.44234563359740897, -0.29349047922231697
Numer iteracji: 6 : RMSE = 0.8660254037844386
```

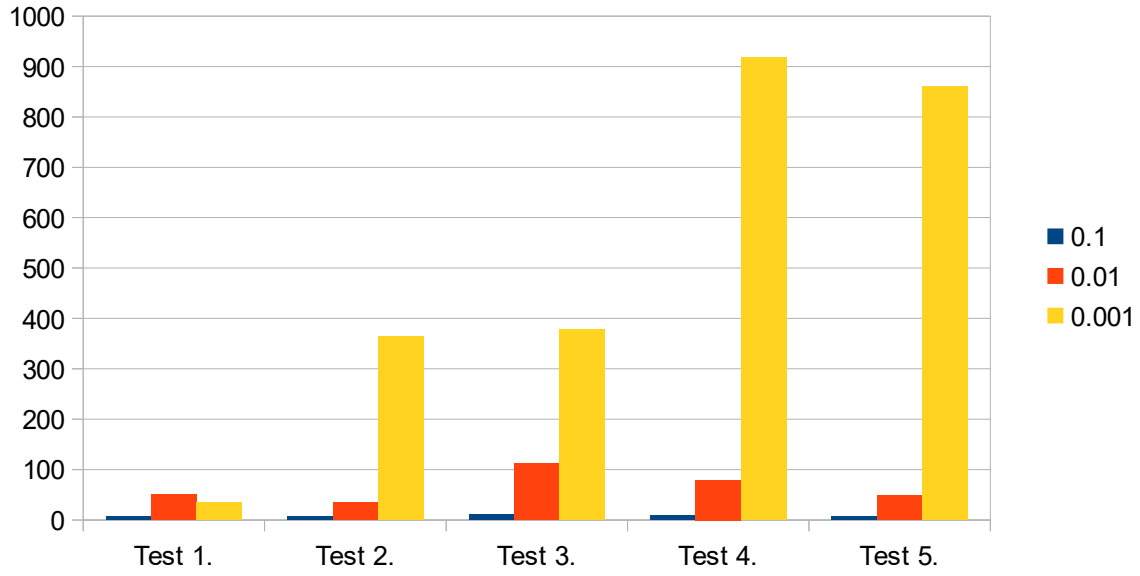
```
Kolejny punkt:
x = 0,y = 1
Obliczony wynik = 1
Wagi: -0.002345633597408716, 0.056509520777683155
Numer iteracji: 33 : RMSE = 0.7071067811865476

=======
Kolejny punkt:
x = 1,y = 0
Obliczony wynik = 1
Wagi: 0.007654366402591284, 0.056509520777683155
Numer iteracji: 34 : RMSE = 0.5

=======
Kolejny punkt:
x = 1,y = 0
Obliczony wynik = 1
Wagi: 0.007654366402591284, 0.056509520777683155
--------------------------NAUCZONY-------------------------------
Numer iteracji: 480 : RMSE = 0.0

=======
Kolejny punkt:
x = 0,y = 1
Obliczony wynik = 1
Wagi: 0.007654366402591284, 0.056509520777683155
Numer iteracji: 481 : RMSE = 0.0

=======
Kolejny punkt:
x = 1,y = 0
Obliczony wynik = 1
Wagi: 0.007654366402591284, 0.056509520777683155
Numer iteracji: 482 : RMSE = 0.0
```

Na przykładzie przeprowadzonych prób widać, że w zależności od wylosowanych/dobranych wag czas uczenia perceptronu będzie dłuższy lub krótszy. Tak samo widać, że gdy wagi są już lepiej dobrane, a perceptron zna podany algorytm, to błąd średniokwadratowy zmierza zmierza do zera.

Po tym jak perceptron zostanie nauczony, ilość iteracji zostaje zwiększona do maksymalnej zadanej, pomniejszonej o 20, by wykonać tyle testów kontrolnych. Testy są przeprowadzone, by sprawdzić, czy perceptron na pewno nauczył się danych bramki logicznej OR.

4. Wyniki:
   Przeprowadziłam 5 prób dla zadanych różnych współczynników uczenia i epok do nauczenia. Wyniki zawarłam w tabeli:

Learn rate	Liczba epok	Test 1.	Test 2.	Test 3.	Test 4.	Test 5.
0.1	500	6	6	11	9	7
0.01	500	50	34	112	79	49
0.001	1500	35	364	378	918	860



Analizując powyższy wykres, widać że im mniejszy współczynnik nauczania, tym więcej epok perceptron potrzebuje, by nauczyć się podanego algorytmu.

5. Wnioski

*Perceptron jest prostym w budowie i szybki narzędziem do rozwiązywania przeróżnych zagadnień. Jednak głównie do tych niezbyt skomplikowanych.
*Wpływ na działanie perceptronu ma wiele składowych: współczynnik uczenia, liczba danych uczących, dobór wag.
*Źle dobrane wagi powodują iż perceptron nie może rozwiązać problemu. Błąd uczenia nie zmierza do 0 co skutkuje bardzo długą analizą danych wejściowych.
*Współczynnik uczenia ma wpływ na szybkość jak i poprawność uczenia się perceptronu.
*Im mniejszy współczynnik uczenia się tym niezawodność perceptronu zwiększa się, jednak do pewnego stopnia ponieważ zbyt mały współczynnik powoduje narastanie błędów.
* Dane uczące maja wpływ na szybkość jak i poprawność uczenia się perceptronu.
*Zbyt duża ilość danych powoduje wydłużenie procesu uczenia się. Gdy danych jest o wiele za dużo może to powodować problemy z działaniem.
* Im mniejszy współczynnik uczenia, tym więcej epok potrzebuje do nauczenia się.
*Na poprawne działanie perceptron ma wpływ dużo składowych dlatego jedną z podstaw przy tworzeniu jest dokładna analiza problemu i dostosowanie owych parametrów w odpowiedni sposób.

6.Literatura, źródła:

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

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