Dominik Kędzior

Inżynieria Obliczeniowa

Nr albumu: 293094

**Sprawozdanie 1**

**Budowa i działanie perceptronu**

# Wstęp

Celem ćwieczenia było poznanie budowy oraz działania perceptronu poprzez implementację i uczenie go wybranej funkcji logicznej dwóch zmiennych.

Perceptron - najprostsza sieć neuronowa, składająca się z jednego bądź wielu niezależnych neuronów McCullocha-Pittsa. Perceptron jest funkcją, która potrafi określić przynależność parametrów wejściowych do jednej z dwóch klas. Może być wykorzystywany tylko do klasyfikowania zbiorów liniowo separowalnych.

Do funkcji liniowo separowalnych należą np. funkcje logiczne AND i OR. Natomiast przykładem funkcji która nie może być rozwiązana przy pomocy pojedynczego perceptronu jest funkcja XOR.

Do wykonania projektu wykorzystałem algorytm z wykłady oraz posiłkowałem się wiedzą ze stron przytoczonych w źródłach. A jego działenie przedstawiłem na przykładzie funkcji OR



Rysunek 1. Bramka logiczna OR

# Opis programu

W programi stworzyłem klasę Perceptron w skład której wchodziły trzy pola tj.

- Weights – wagi

- epochs – ilość epok treningowych

- learning\_rate – wspóczynnik uczenia

Oraz trzy metoda tj.

- activation(x) – funkcja aktywacji która jest implementacją funkcji skoku jednostkowego.

- predict(x) – funkcja która wylicza wynik z wykożystaniem wzorów

- fit(X,d) – funkcja ucząca oparta na zasadzie wzorów

w – wagi

e – bład

y – wynki działania funkcji predict na danych uczących wejściowych

d – dane uczące wynikowe

η – współczynnki uczenia

# Wyniki działania programu

Dla danych uczących [0,0], [0,1], [1,0], [1,1] →[0, 1, 1, 1]

Dane testowe [0,0], [0,1], [1,0], [1,1] →[0, 1, 1, 1]

Współczynnki uczenia = 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Liczba epok | 0 | 1 | 3 | 6 | 10 |
| wyniki | [1, 1, 1, 1] | [1, 1, 1, 1] | [0, 1, 1, 1] | [0, 1, 1, 1] | [0, 1, 1, 1] |

Dla danych uczących [0,0], [0,1], [1,0] →[0, 1, 1]

Dane testowe [0,0], [0,1], [1,0], [1,1] →[0, 1, 1, 1]

Współczynnki uczenia = 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Liczba epok | 0 | 1 | 3 | 6 | 10 |
| wyniki | [1, 1, 1, 1] | [1, 1, 1, 1] | [0, 1, 1, 1] | [0, 1, 1, 1] | [0, 1, 1, 1] |

Dla danych uczących [0,0]→[0]

Dane testowe [0,0], [0,1], [1,0], [1,1] →[0, 1, 1, 1]

Współczynnki uczenia = 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Liczba epok | 0 | 1 | 3 | 6 | 10 |
| wyniki | [1, 1, 1, 1] | [0, 0, 0, 0] | [0, 0, 0, 0] | [0, 0, 0, 0] | [0, 0, 0, 0] |

Dla danych uczących [0,1], [0,0], [1,1], [1,0], [0,0], [0,1]→ [1, 0, 1, 1, 0, 1]

Dane testowe [0,0], [0,1], [1,0], [1,1] →[0, 1, 1, 1]

Współczynnki uczenia = 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Liczba epok | 0 | 1 | 3 | 6 | 10 |
| wyniki | [1, 1, 1, 1] | [0, 1, 1, 1] | [0, 1, 1, 1] | [0, 1, 1, 1] | [0, 1, 1, 1] |

Analizując powyższe tabele można zauważyć że istotny wpływ na poprawność działania naszej sieci ma ilość epok. W większości przypadków poprawne wyniki otrzymywałem już od trzeciej epoki. Nie można jednak pominać faktu że istotny wpływ na poprawność wyników mają dane uczące w przypadku gdy danych uczących było za mało perceptron nigdy nie podał dobrych wyników, w przypadku dużej liczby danych uczących program diałał poprawnie już od pierwszej epoki.

Nie bez znaczenia jest również współczynnik uczenia od którego wartość opisuje jak szybko dana sieć ma się uczyć.

Wykres błędu dla

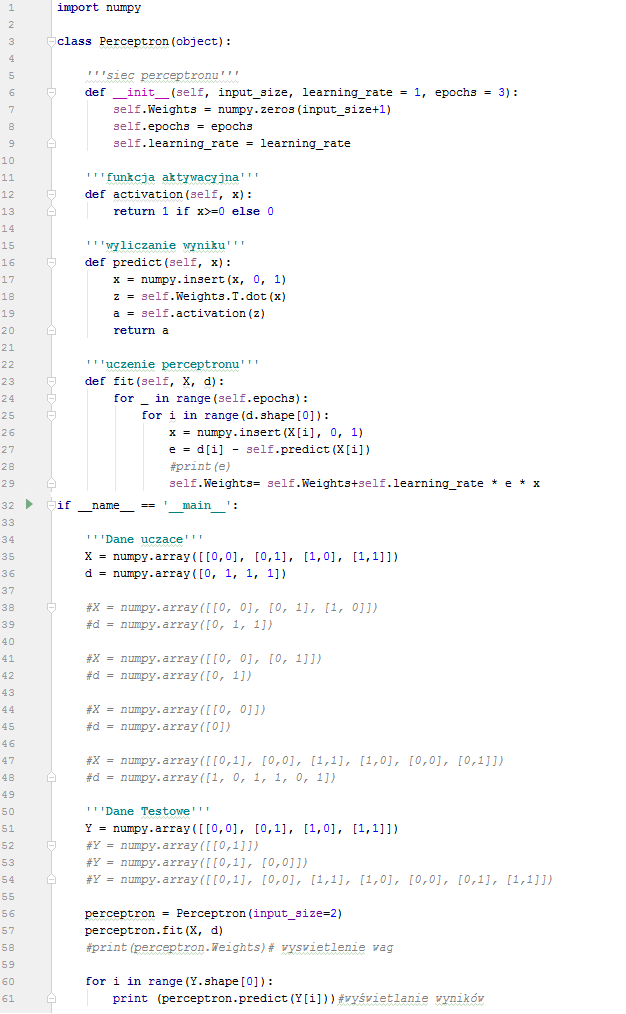
Dla danych uczących [0,0], [0,1], [1,0], [1,1] →[0, 1, 1, 1]

Liczba epok = 4

Na powyższym wykresie można zaobserwować że ostatni niezerowy błąd miał miejsce w pierwszej iteracji trzeciej epoki co pokrywa się z obserwacjami że dla omawianego przypadku poprawne wyniki uzyskujemy od trzeciej epoki.

# Wnioski

Na działenie perceptronu ma wpływ kilka czynników tj. Ilość epok treningowych, których zbyt mała ilość sprawia że program nie przedstawia rzeczywistości. Jeżeli jednak ich liczba będzie zbyt duża wynki również mogą być niedokładne w związku z przuczeniem sieci. Kolejnym elementem jest współczynnik uczenia którego niska wartość poprawia jakość uczenia się, wyskoka natomiast zakłóca poprawne działanie perceptronu. Jednak jedną z najważniejszych ról pełnią dane uczące bo to na nich opiera się działanie całego programu. Jeżeli ilość tych danych jest zyt mała, lub co gorsza któryś wynik nie odzwierciedla rzeczywistość działanie programu również będzie powielać ten bład, aby perceptron działał poprawnie osoba która go projektuje musi się wykazać wiedzą na temat symulowanego zjawiska.



Rysunek 2. Kod programu

# Źródła:

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

<https://pythonmachinelearning.pro/perceptrons-the-first-neural-networks/>

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

<https://uk.mathworks.com/help/deeplearning/ref/perceptron.html;jsessionid=c07c91e5317eaffb754b701b3338>