

PSI-Scenariusz 1

Kacper Pawlikowski

IS. Gr.3

W swoim programie zastosowałem Neuron McCullocha-Pittsa z dwoma wejściami. Dla każdego wejścia x_i perceptronu przypisana jest waga w_i . Dla stanów wejściowych liczymy sumę ważoną:

$$s = \sum_{i=1}^n x_i w_i + b$$

b-wartość odchylenia, odpowiada za nieliniowe przekształcenie wejść w wyjście

Funkcję progową unipolarną w postaci:

$$y = \begin{cases} 0 & \text{dla } s < 0 \\ 1 & \text{dla } s \geq 0 \end{cases} \quad \text{y-wyjście neuronu}$$

użyłem jako funkcję aktywacji.

Skorzystałem z następującego algorytmu uczenia:

- Początkowe wagi zostały wylosowane z zakresu $\langle -0.5, 0.5 \rangle$
- Sprawdzam czy na podstawie przygotowanych danych wejściowych otrzymam oczekiwany wynik. Jeżeli nie:
 - Obliczam błąd: $e = \text{uzyskany_wynik} - \text{oczekiwany_wynik}$
 - Modyfikuję wagi:
 $\text{Waga} = \text{Waga} + \text{współczynnik_uczenia} * e * \text{dana_wejściowa}$, oraz $b = b + \text{współczynnik_uczenia} * e$
- Procedurę powtarzam dla wszystkich przygotowanych zestawów danych (kolejność użycia zestawów jest losowa) a następnie sprawdzam błąd średniokwadratowy:

$$E = \frac{1}{2} \sum_{i=1}^p (d_i - y_i)^2$$

p-liczba przykładów do nauki

d_i -oczekiwana odpowiedź perceptronu

y_i -uzyskana odpowiedź

- Jeżeli $e > 0$ to powtarzam proces uczenia

Dane do uczenia:

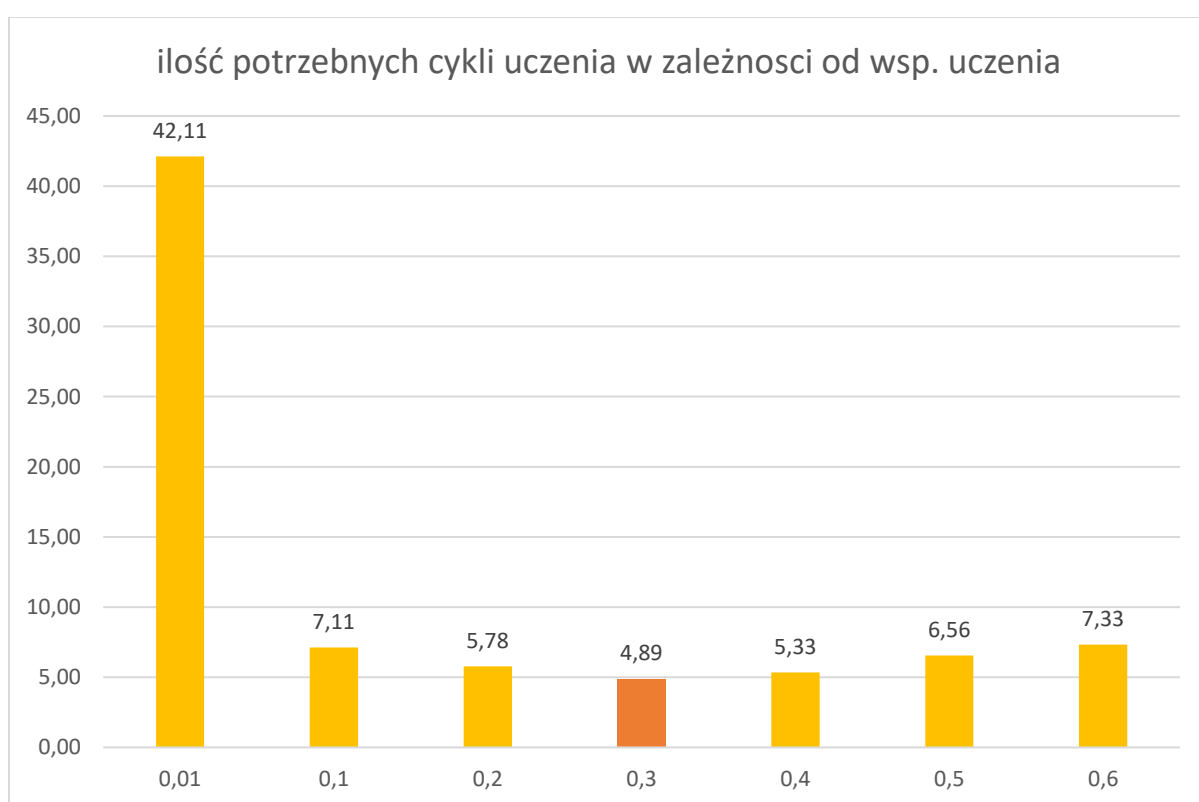
Neuron ma realizować procedurę OR

Dane wejściowe: $\{(0,0), (0,1), (1,0), (1,1)\}$

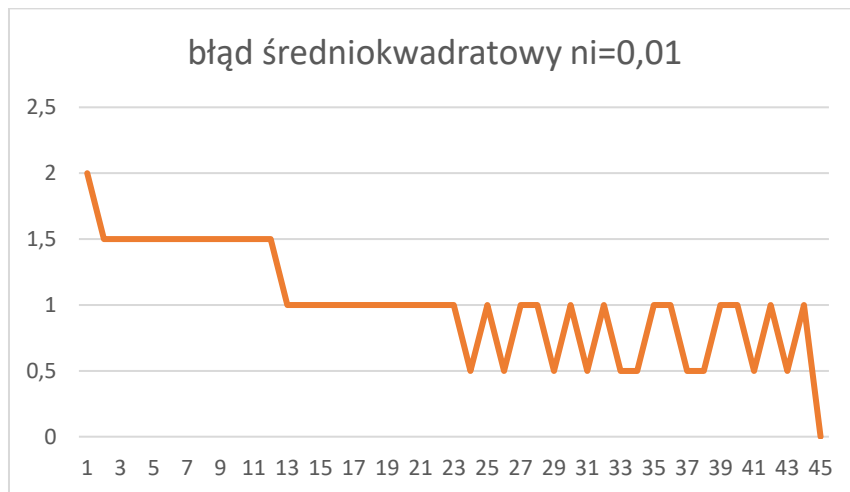
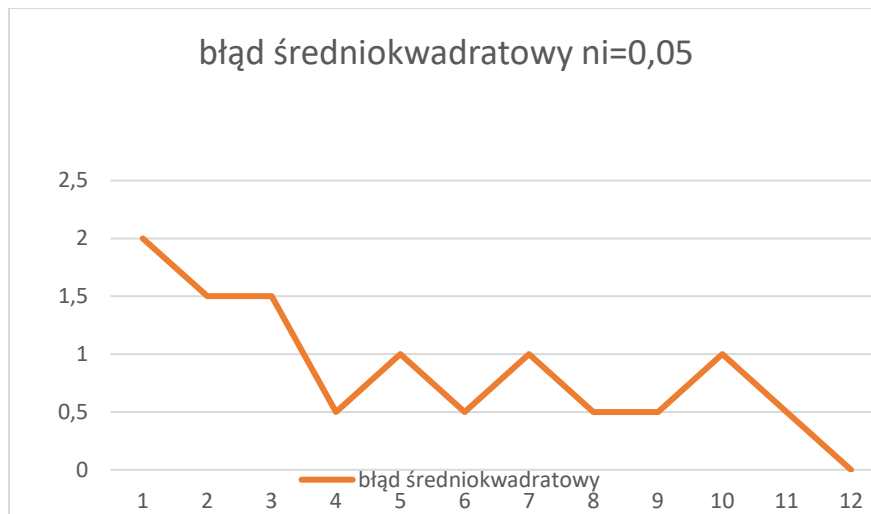
Dane wyjściowe: $\{0, 1, 1, 1\}$

Zależność pomiędzy współczynnikiem uczenia a ilością niezbędnych cykli uczenia

ni	ilość potrzebnych cykli uczenia									
	test 1	test 2	test 3	test 4	test 5	test 6	test 7	test 8	test 9	średnia
0,01	48	51	51	35	37	37	45	9	66	42,11
0,1	5	10	5	10	6	12	4	6	6	7,11
0,2	6	4	10	7	4	5	7	6	3	5,78
0,3	4	2	7	6	8	6	4	3	4	4,89
0,4	6	8	4	4	2	7	5	8	4	5,33
0,5	5	5	3	9	5	4	9	9	10	6,56
0,6	7	9	4	10	4	8	5	10	9	7,33



Błąd średniokwadratowy w kolejnych iteracjach:



Wnioski:

- Im współczynnik uczenia jest większy, tym większa poprawka wag jest wykonywana przy takiej samej wartości błędu. Dlatego dla małych wartości η potrzeba znacznie więcej iteracji. Należy jednak pamiętać, że dla dużych wartości η poprawka wag może być za duża i nie przybliży nas do rozwiązania. W moim przypadku optymalna wartość współczynnika uczenia wynosiła ok 0,3.
- Początkowo zamierzałem wykonać funkcję XOR ale nigdy nie otrzymałem prawidłowych wyników. Doczytałem, że wiąże się to z tym, że pojedynczy neuron nie jest w stanie odróżnić zbiorów nieseparowalnych liniowo, czyli takich, że między punktami z odpowiedzią na tak i odpowiedzią negatywną nie da się poprowadzić prostej rozgraniczającej. Problem ten można rozwiązać poprzez zastosowanie sieci neuronów.
- W początkowej fazie nauki perceptronu błąd średniokwadratowy maleje regularnie natomiast gdy zbliża się do 0 to zaczyna się zachowywać w nie przewidywalny sposób. Dzieje się tak ponieważ na początku wartości wag znacząco różnią się od wymaganych. W późniejszej fazie dokonywane są już niewielkie korekty które znacząco wpływają na wynik.

