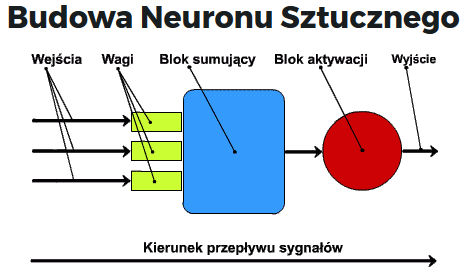
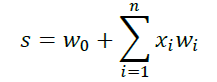
**SZTUCZNY NEURON -** toprosty system przetwarzający wartości sygnałów wprowadzanych na jego wejścia w pojedynczą wartośd wyjściową, wysyłaną na jego jednym wyjściu (dokładny sposób funkcjonowania określony jest przez przyjęty model neuronu). Jest to podstawowy element sieci neuronowych, jego pierwowzorem był biologiczny neuron.



**Neuron McCullocha-Pittsa**– jeden z matematycznych modeli neuronu. Posiada wiele wejść i jedno wyjście. Każdemu z wejść przyporządkowana jest liczba rzeczywista - waga wejścia(liczba rzeczywista). Neuron ten jest podstawowym budulcem sieci neuronowej perceptron. Wartość na wyjściu neuronu obliczana jest w następujący sposób:

1. obliczana jest suma iloczynów wartości xi podanych na wejścia i wag wi wejść:



2. na wyjście podawana jest wartość funkcji aktywacji f(s) dla obliczonej sumy

**FUNKCJA AKTYWACJI -** funkcja, według której obliczana jest wartość wyjścia neuronów sieci neuronowej.

Wykorzystałam neuron **McCullocha-Pittsa** z dwoma wejściami. Jako funkcję aktywacji wykorzystałam funkcję progową unipolarną w postaci:

y-wyjście neuronu

Skorzystałam z następującego algorytmu uczenia:

* Początkowe wagi zostały wylosowane z zakresu <-0.5, 0.5>
* Sprawdziłam czy na podstawie tak przygotowanych danych wejściowych otrzymam oczekiwany wynik.

Jeżeli nie uzyskałam oczekiwanego wyniku to wykonałam następujące czynności:

* + Obliczyłam błąd: e=uzyskany\_wynik - oczekiwany\_wynik
  + Modyfikujacja wagi: Waga=Waga+współczynnik\_uczenia\*e\*dana\_wejściowa, oraz b=b+współczynnik\_uczenia \*e
* Procedurę powturzyłam dla wszystkich przygotowanych zestawów danych (kolejność użycia zestawów jest losowa) a następnie sprawdziłam błąd średniokwadratowy:

E=

p-liczba przykładów do nauki

-oczekiwana odpowiedź perceptronu

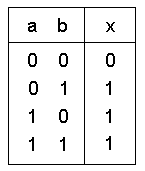
-uzyskana odpowiedź

* Jeżeli e>0 to powtarzam proces uczenia

W moim programie neuron miał wykonać operacje logiczną OR:

Operacja logiczna OR przyjmuje 2 parametry binarne i odpowiada wartością 0 lub 1.

Tabela przedstawia odpowiedzi funkcji XOR dla wszystkich możliwych danych wejściowych:



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | liczba cykli uczenia | | | | | | | | | |
| ni | test 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | średnia |
| 0,01 | 50 | 48 | 51 | 41 | 38 | 39 | 45 | 10 | 59 | 42,33 |
| 0,1 | 10 | 6 | 5 | 11 | 8 | 9 | 10 | 7 | 6 | 7,89 |
| 0,2 | 5 | 6 | 10 | 7 | 5 | 5 | 6 | 7 | 4 | 6,10 |
| 0,3 | 5 | 3 | 6 | 6 | 7 | 4 | 6 | 3 | 4 | 4,88 |
| 0,4 | 6 | 9 | 3 | 4 | 3 | 8 | 5 | 7 | 5 | 5,55 |
| 0,5 | 7 | 6 | 4 | 8 | 5 | 4 | 8 | 9 | 10 | 6,55 |
| 0,6 | 8 | 9 | 5 | 10 | 6 | 8 | 5 | 8 | 7 | 7,33 |

Tabela przedstawia zależność pomiędzy współczynnikiem uczenia a ilością potrzebnych cykli uczenia.

Wnioski:

Dzięki wynikom zebranym w tabeli można zauważyć, że jeśli współczynnik uczenia jest większy to przy takiej samej wartości błędu jest wykonywana większa poprawka wag, więc dla małych wartości ƞ potrzebne jest więcej iteracji. Jednak może się też zdarzyć, że poprawka wag może być za duża dla dużych wartości ƞ i nie przybliży nas do rozwiązania. W moich testach najlepsza wartość współczynnika uczenia wyniosła 0,3.