Podstawy Sztucznej Inteligencji Monika Michalik – gr.2

**Scenariusz 1. Budowa i działanie perceptronu**

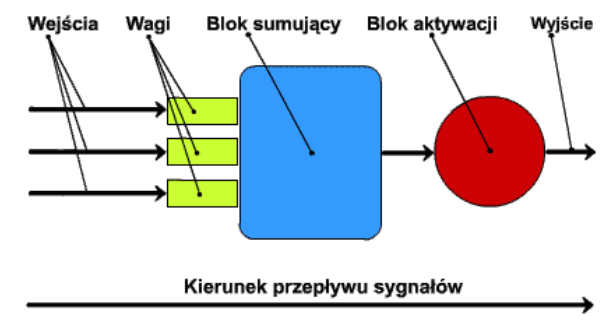
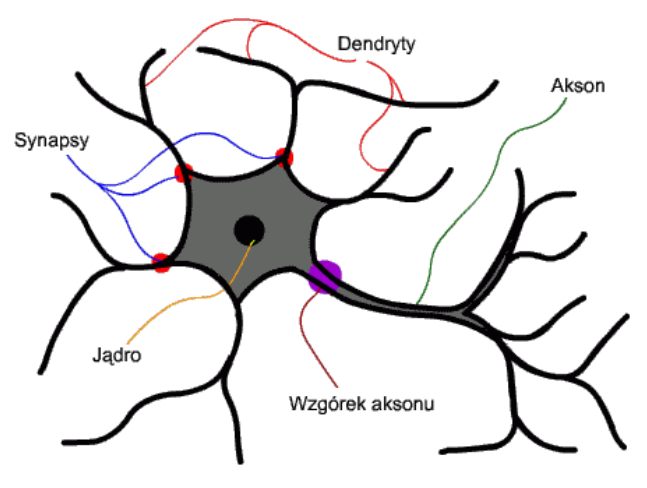
**Cel ćwiczenia:**

Celem ćwiczenia jest poznanie budowy i działanie perceptronu poprzez implementację oraz uczenie perceptronu realizującego wybraną funkcję logiczną dwóch zmiennych.

**Zadania do wykonania:**

**1. Implementacja sztucznego neuronu.**

Budowa neuronu biologicznego | Budowa neuronu sztucznego



Porównanie elementów obu neuronów:

1. Dendryty – Wejścia
2. Synapsy – Wagi
3. Jądro – Blok sumujący
4. Wzgórek aksonu – Blok aktywacji
5. Akson – Wyjście

Na wejściu umieszczane są dane, które zostają przemnożone przez wartości wag. Dla każdego powiązania dendryt<->synapsa zachodzi równanie:

**R = d\*w**

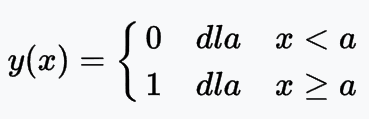
R – wynik dla pojedynczego połączenia dendryt<->synapsa  
d – informacja przechowywana w dendrycie  
w – wartość wagi w synapsie

Wyniki wszystkich połączeń trafiają do bloku sumującego, w którym są sumowane. W efekcie otrzymujemy tzw. potencjał membranowy. Proces ten obrazuje równanie:

**Pm = R1 + R2 + R3 +…+ Rn, dla n > 0**

Pm – potencjał membranowy  
R – wynik połączenia synaptycznego  
n – ilość połączeń synaptycznych w neuronie

Ostatecznie potencjał membranowy przechodzi przez blok aktywacyjny. W tym miejscu zostaje poddany funkcji aktywacyjnej, która przyjmuje postać funkcji progowej unipolarnej:

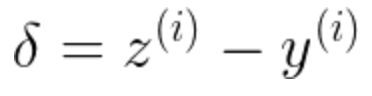


a – zadana wartość progowa, a = 0

Na wyjście zostaje oddana wartość zmodyfikowana w bloku aktywacyjnym.

**Implementacja algorytmu uczenia**

Dla każdego wzorca liczymy błąd, wg wzoru:

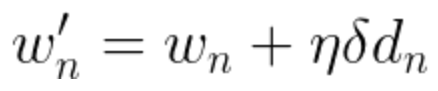


***z*** - oczekiwana odpowiedź (wzorcowa),

***y*** - wartość, która pojawiła się na wyjściu sieci.

**i** – numer wzorca z naszego zestawu (każdy wzorzec to p,q i wartość p^q)

**Zmianę wag w synapsie dokonujemy wg wzoru:**



+ w0

Do wag na poszczególnych synapsach neuronu dodajemy iloczyn: współczynnika uczenia (ɳ), błędu neuronu i wartości wejściowej na dendrycie. w0 – wartość wagi dla wejścia bias

**Dodatkowe wejście**: Neuron może być wyposażony w tak zwany bias, czyli dodatkowe wejście, na którym występuje stała wartość. Waga dla tego wejścia jest modyfikowana w trakcie procesu uczenia tak jak wszystkie pozostałe wagi. W naszym przypadku na wejściu bias występuje sygnał stale równy jeden. Jeżeli przyjmiemy wartość na wejściu bias równą 0 to otrzymamy wzór dla zwykłego neuronu. W przypadku neuronu jednowejściowego zastosowanie biasu umożliwia przesuwanie progu aktywacji wzdłuż osi X. Dlatego neuron z biasem powinien uczyć się nawet takich wektorów których zwykły neuron nie byłby w stanie się nauczyć. Wprowadzenie do neuronu dodatkowego wejścia bias powoduje wzrost ich możliwości uczenia się. Jest to związane z umożliwianiem przesuwania progu aktywacji w zależności od wagi biasu. Zastosowanie biasu powoduje jednak wzrost ilości obliczeń wynikających z konieczności ustalania dodatkowej wagi.

**2. Wygenerowanie danych uczących i testujących.**

Aby przeprowadzić proces uczenia potrzebny jest nam zestaw wzorców.

Wybrana funkcja logiczna: **AND**

  
1-prawda, 0-fałsz

p, q – dane uczące

p ^ q – dane testujące

**3. Uczenie perceptronu dla różnej liczby danych uczących, różnych współczynników uczenia.**

**a) różny współczynnik uczenia, wagi losowe**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | ŚREDNIA |
| Liczba danych | 4 | 4 | 4 | 4 | 4 |
| Waga początkowa | Losowa | Losowa | Losowa | Losowa | Losowa |
| Współczynnik uczenia | 0.01 | 0.01 | 0.01 | 0.01 | **0.01** |
| Liczba iteracji | 57 | 32 | 54 | 35 | **~45** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | ŚREDNIA |
| Liczba danych | 4 | 4 | 4 | 4 | 4 |
| Waga początkowa | Losowa | Losowa | Losowa | Losowa | Losowa |
| Współczynnik uczenia | 0.001 | 0.001 | 0.001 | 0.001 | **0.001** |
| Liczba iteracji | 319 | 353 | 204 | 355 | **~308** |

**Obserwacja:** Badając zmiany dla różnych współczynników uczenia obserwujemy, że jego zmiana znacząco wpływa na liczbę iteracji. Czym większy współczynnik, tym mniej iteracji zostaje wykonanych.

**b) różny współczynnik uczenia, wagi ustalone**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Liczba danych | 4 | 4 | 4 | 4 | 4 |
| Waga początkowa | 0.0 | 0.5 | **0.8** | 0.0 | **0.8** |
| Współczynnik uczenia | 0.01 | 0.01 | **0.01** | 0.1 | **0.1** |
| Liczba iteracji | 5 | 28 | **45** | 3 | **5** |

**Obserwacja:** Przy mniejszym współczynniku uczenia mamy większą rozbieżność liczby iteracji dla różnych wag początkowych. Wraz ze wzrostem wagi początkowej, liczba iteracji rośnie.

**c) różna liczba danych uczących**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Liczba danych | 10 | 10 | 10 | 10 | 10 |
| Waga początkowa | 0.0 | 0.5 | 0.8 | 0.0 | 0.8 |
| Współczynnik uczenia | 0.01 | 0.01 | 0.01 | 0.1 | 0.1 |
| Liczba iteracji | 2 | 13 | 21 | 2 | 2 |

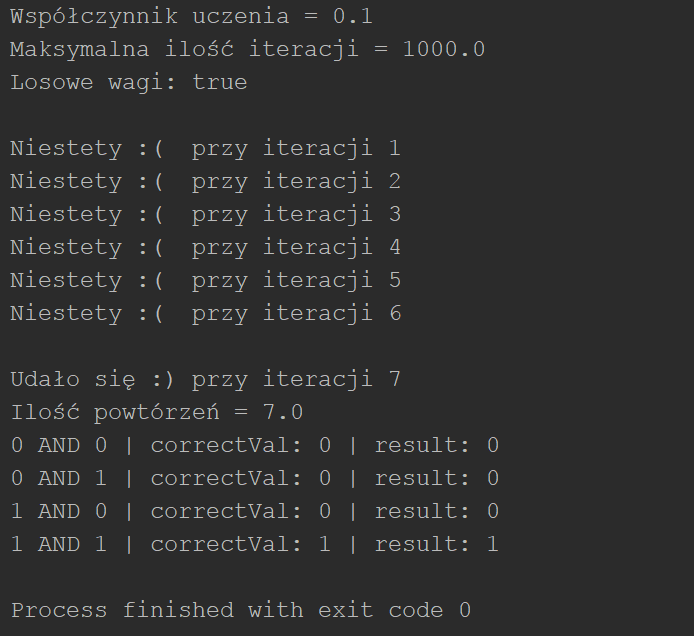
**Obserwacja: Porównując liczbę danych z powyższą tabelą – zwiększając liczbę danych wejściowych, w konsekwencji zmniejszamy liczbę iteracji.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Liczba danych | 10 | 10 | 10 | 10 | 10 |
| Waga początkowa | Losowa | Losowa | Losowa | Losowa | Losowa |
| Współczynnik uczenia | 0.01 | 0.01 | 0.01 | 0.1 | 0.1 |
| Liczba iteracji | 21 | 11 | 16 | 3 | 2 |

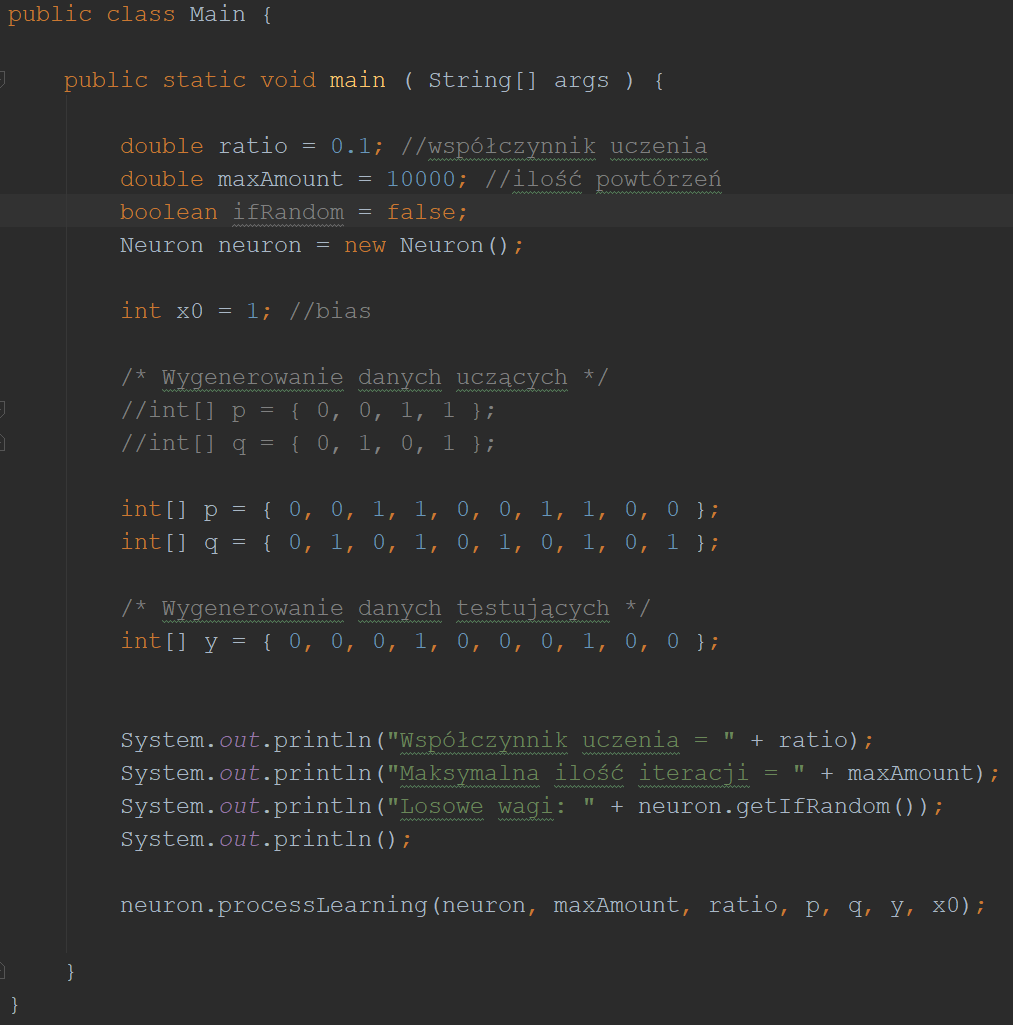
**Obserwacja: Niższy współczynnik uczenia wpływa na wyższą liczbę iteracji.**

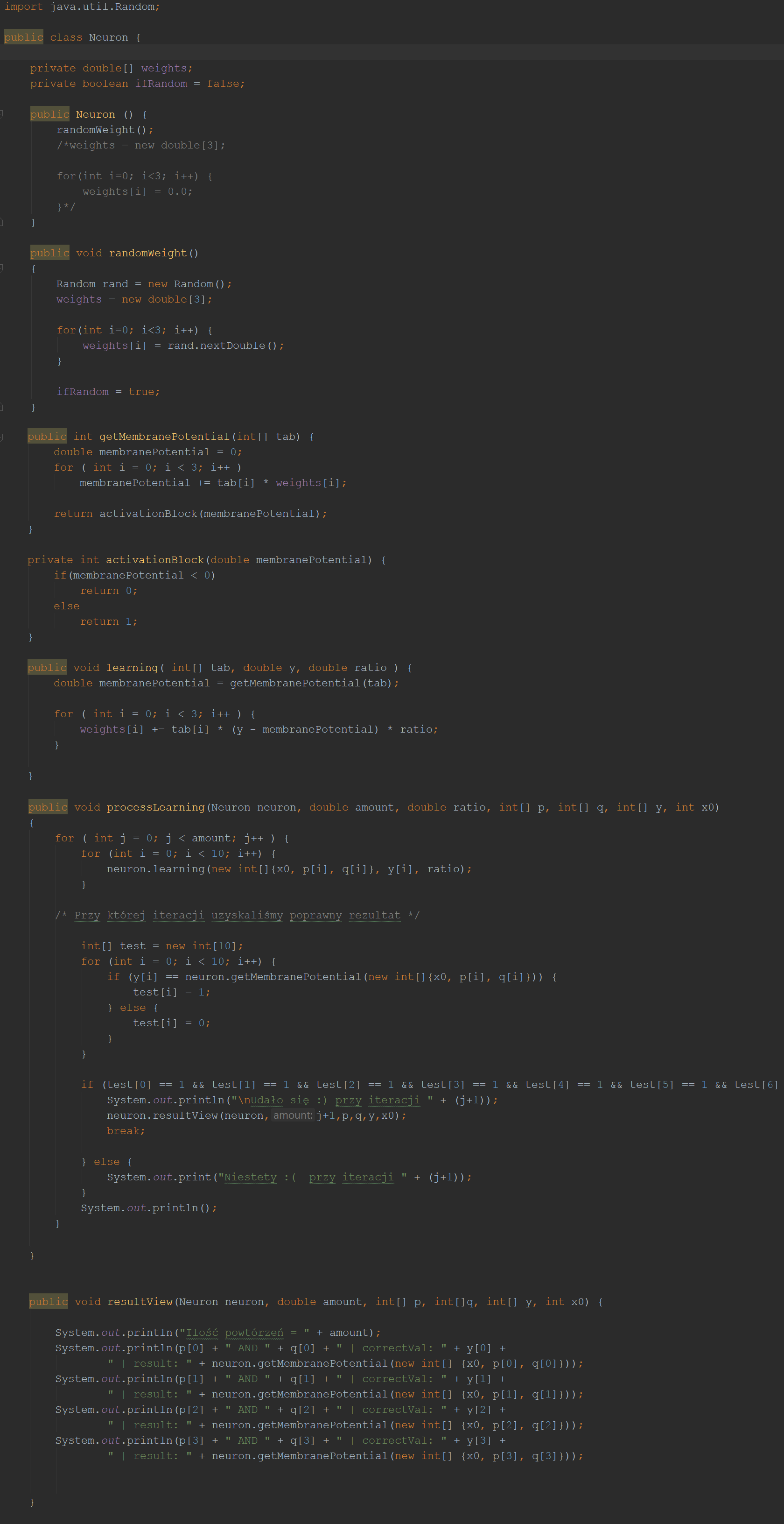
**4. Wnioski.**

Współczynnik uczenia odpowiada za prędkość uczenia sieci. Wraz z jego wzrostem – uczenie przebiega szybciej, jednak mniej precyzyjnie. Niższe wartości powodują uczenie wolniejsze, ale za to dokładne i precyzyjne.

****

**5. Listing kodu:**

****

****