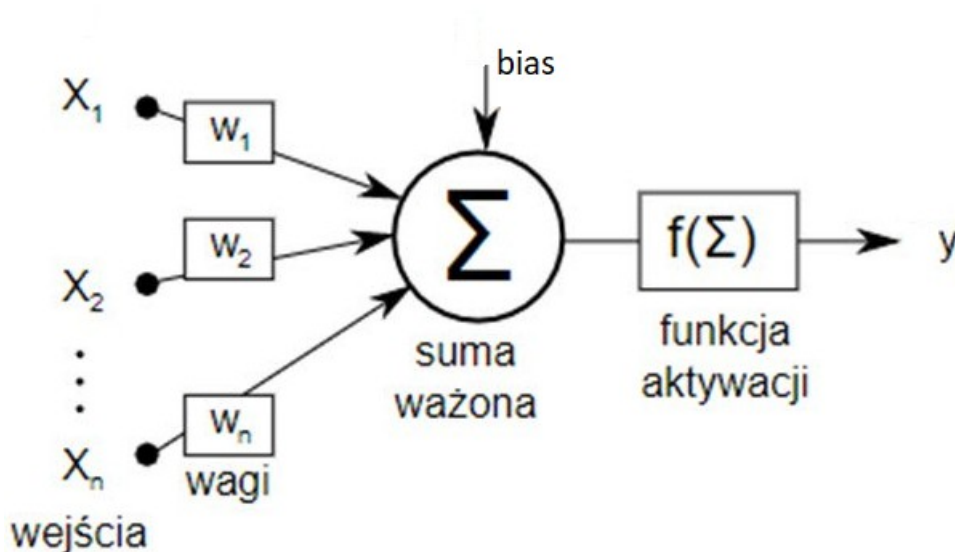


Sprawozdanie lab05

Metody inteligencji obliczeniowej – Informatyka Stosowana, WFIIS,
Jakub Salamon, II rok

Celem zajęć laboratoryjnych nr 5 była klasyfikacja za pomocą modelu pojedynczego sztucznego neuronu.

Wstęp



Model pojedynczego sztucznego neuronu – perceptron

Perceptron to najprostsza sieć neuronowa. Jest to funkcja, która potrafi określić, do której z dwóch klas należy parametr wejściowy. Na wejściu otrzymuje wektor wejściowy $X = [x_1, \dots, x_n]$, a na wyjściu daje nam wartość y . Do każdego x jest przypisana waga. Najpierw wykonywana jest agregacja danych według wzoru:

$$net = \sum_{i=0}^n w_i x_i + bias \text{ (z wagą o indeksie zerowym, dla którego wejście jest zawsze równe 1)}$$

Następnie wyznaczana jest wartość wyjściowa za pomocą funkcji aktywacji $f()$.

$$y = f(net)$$

Sieć ADALINE

Budowa neuronu jest bardzo podobna do modelu perceptronu, ale ma ona inny algorytm uczenia. Sieć ADALINE porównuje sygnał wzorcowy z sygnałem, który otrzymuje na wyjściu części liniowej neuronu. Nie uwzględnia przez to funkcji aktywacji. Aktualizacja wag odbywa się więc w zależności od sumy ważonej wejść. Jest to podstawowa różnica między tą siecią a perceptronem, który aktualizuje wagi w zależności od funkcji aktywacji.

Dane losowe

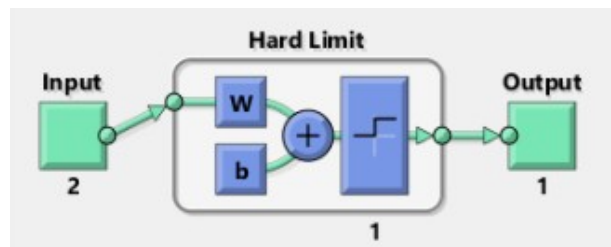
Wygenerowaliśmy losowo z rozkładu normalnego dwuwymiarowe wektory danych należące do jednej z dwóch klas:

$$K_1 = N([0, 0], 1)$$

$$K_2 = N([2, 0], 1)$$

Uczyliśmy neuron na podstawie 50 danych wejściowych dla każdej z klas, a następnie przetestowaliśmy neuron dla 1000 danych wejściowych testowych także dla każdej z klas.

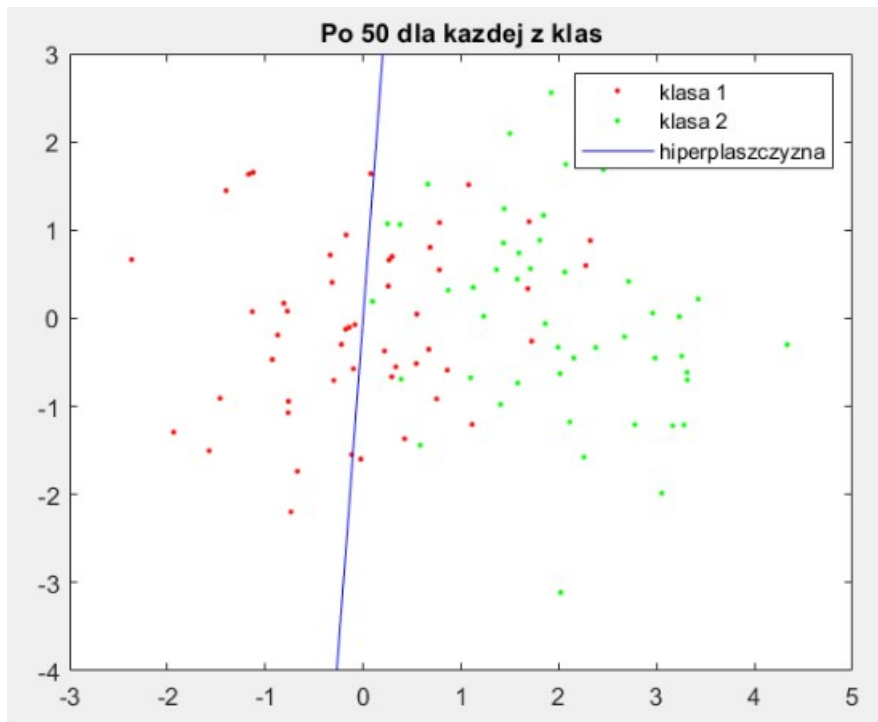
Tak wygląda nasz perceptron



Po przetrenowaniu sieci przez 50 epok otrzymujemy takie wyniki.

Training Confusion Matrix			
Output Class	0	1	
	27 27.0%	0 0.0%	100% 0.0%
	23 23.0%	50 50.0%	68.5% 31.5%
	54.0% 46.0%	100% 0.0%	77.0% 23.0%
	0	1	
	Target Class		

Jak widzimy dla danych uczących neuron bardzo dobrze klasyfikuje dane do drugiej klasy, bo w 100% poprawnie. Natomiast dla klasy pierwszej poprawność wynosi tylko 54%. Daje to w sumie 77% poprawnej klasyfikacji danych wejściowych.



W przypadku danych testowych poprawna klasyfikacja miała miejsce dla 73.5% danych wejściowych.

Funkcja hiperplaszczyny jest przedstawiona na wykresie przez niebieską linię. Rozdziela ona klasy na wykresie. Wyraża się wzorem:

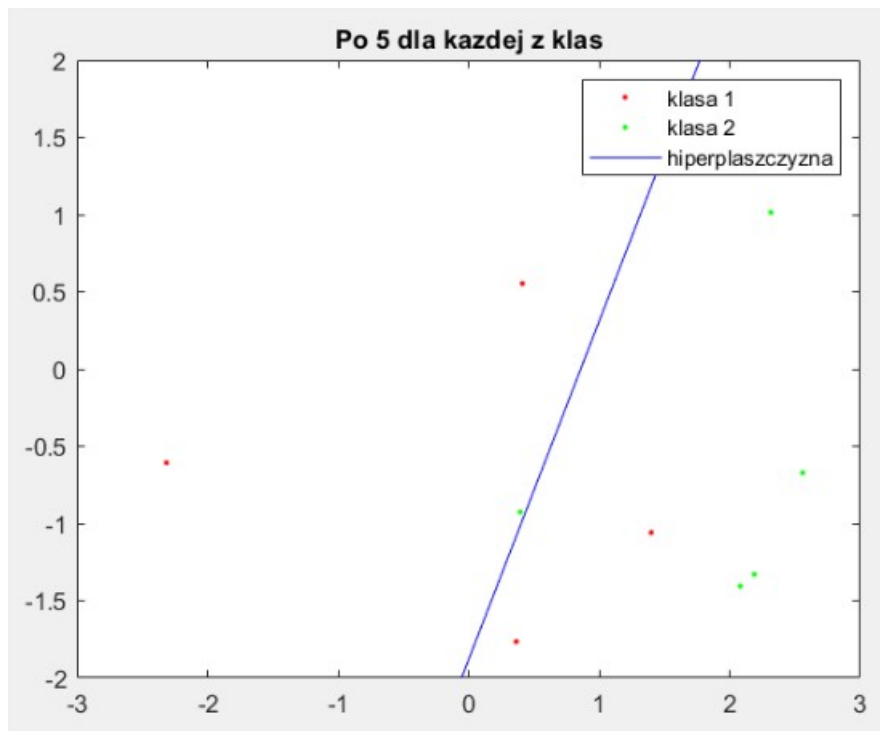
$$w_1 x + w_2 y + b = 0 \quad \text{więc} \quad y = -\frac{w_1}{w_2} x + \frac{-b}{w_2}, \quad \text{gdzie } w_1, w_2 - \text{wagi}$$

Wzór funkcji hiperplaszczyny na wykresie: $y = 14.908x + 0$

Następnie mieliśmy przeanalizować proces klasyfikacji, gdy liczba wzorców uczących wynosiła 5, 10, 20 oraz 100 dla każdej z klas, a liczba epok pozostała taka sama (50).

Dla 5 dla każdej z klas

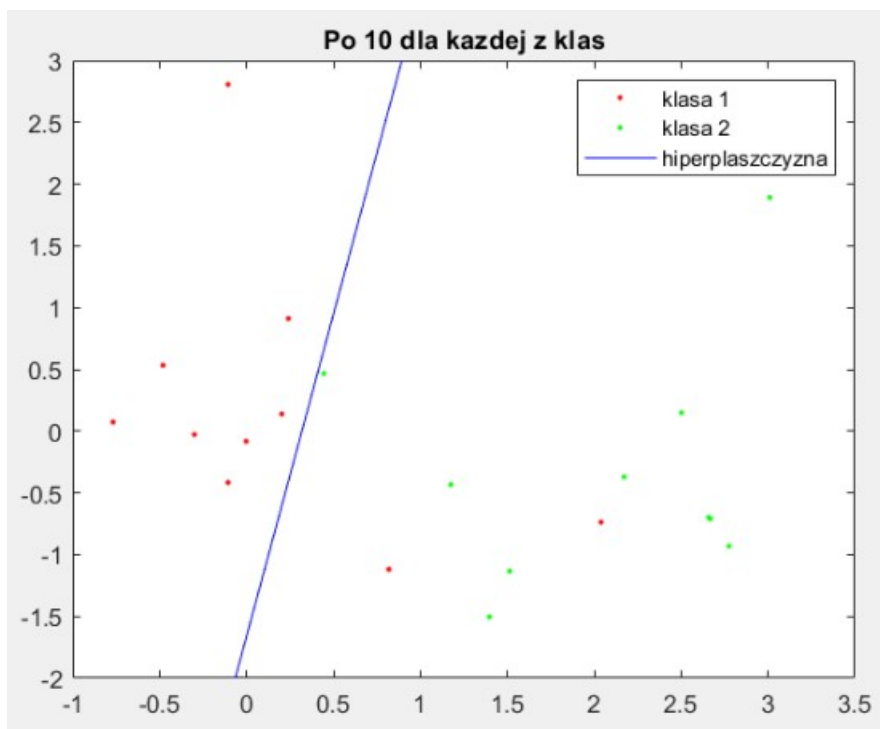
W takim przypadku neuron działał bardzo dobrze. Dla wielu uruchomień trenowanie potrafiło zakończyć się po mniej niż 50 epokach. W takim przypadku klasyfikacja danych uczących wynosiła 100%, a danych testowych 84.25%. W przypadku przejścia przez 50 epok dostajemy wynik 90% poprawnej klasyfikacji dla danych uczących oraz 82.3% dla danych testujących.



Wzór funkcji hiperplaszczyny na wykresie: $y = 4.6215x - 2.7021$

Dla 10 dla każdej z klas

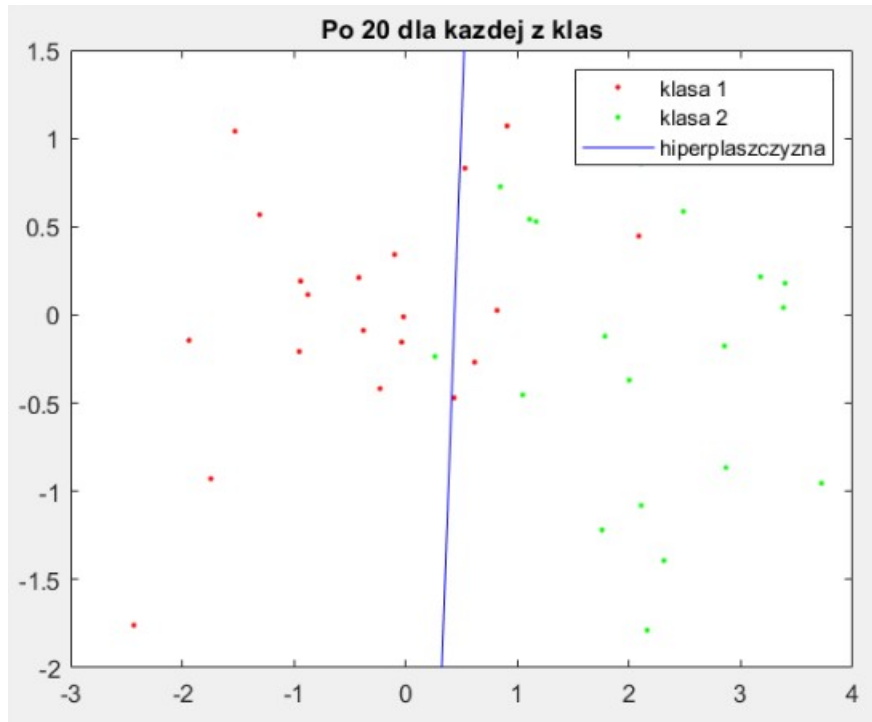
Otrzymaliśmy klasyfikacje dla danych uczących na poziomie 100%, a dla danych testowych 81.85%.



Wzór funkcji hiperplaszczyny na wykresie: $y = 5.2186x - 1.6651$

Dla 20 dla każdej z klas

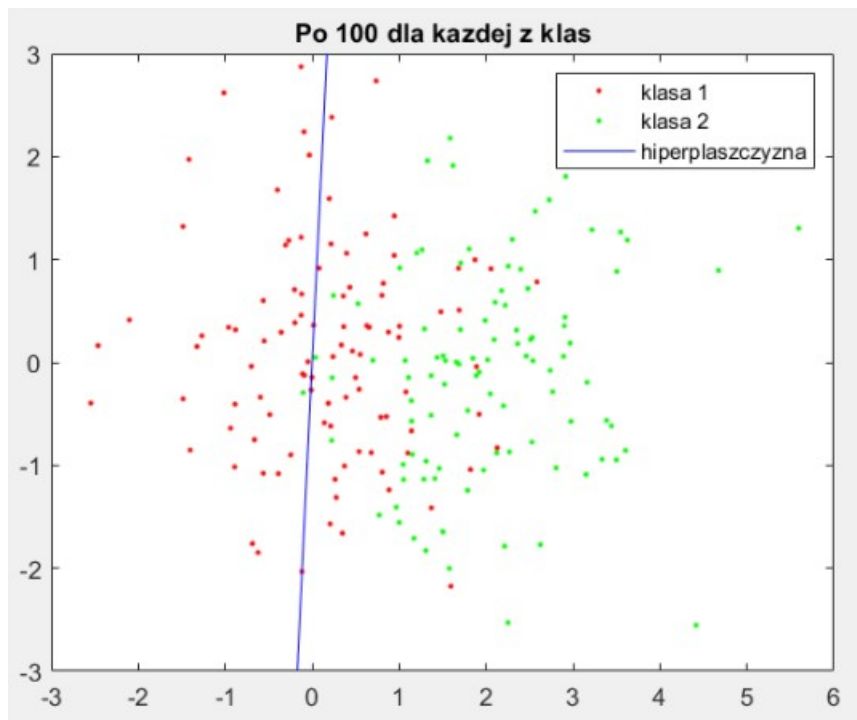
Otrzymaliśmy poprawną klasyfikację na poziomie 82.5% dla danych uczących oraz 80.25% dla danych testowych.



Wzór funkcji hiperplaszczyny na wykresie: $17.6542x - 7.7245$

Dla 100 dla każdej z klas

Otrzymaliśmy dobrą klasyfikację na poziomie 71% dla danych uczących oraz 75.65% dla danych testowych.



Wzór funkcji hiperplaszczyny na wykresie: $17.5743x + 0$

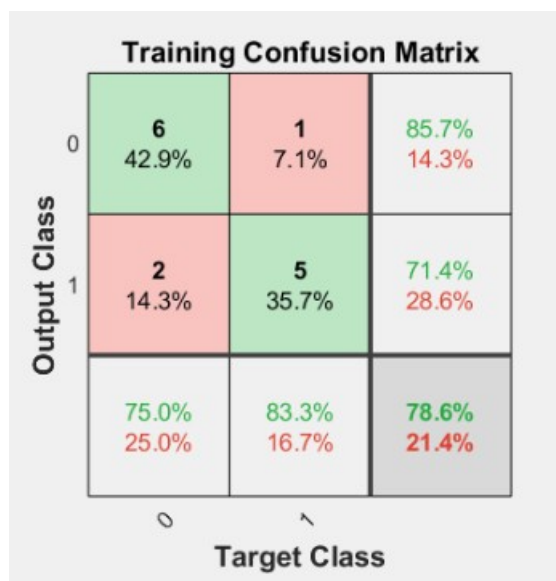
Wnioski

Z powyższych wyników widzimy, że im mniej danych uczących tym lepszy wynik jesteśmy w stanie uzyskać. Jest to sprzeczne z naszą intuicją, ale wynika z faktu, że dane wejściowe są bardzo proste i większa ich ilość tylko psuje poprawne klasyfikowanie. Gdy dane wejściowe są bardziej skomplikowane lepiej jest stosować większą próbę uczącą.

Dane rzeczywiste

Mamy sprawdzić skuteczność sieci z pojedynczym neuronem na podstawie otrzymanych danych otrzymanych próbek benzyny i ich klasyfikacji do klasy czystości (0, 1).

Poprawność klasyfikacji dla danych uczących w przypadku dzielenia danych wejściowych na zbiór 14 elementów uczących i 16 trenujących (szczegóły w tabelce).



Określiłem maksymalną liczbę epok nauki na 10000. Algorytm może jednak zakończyć ją wcześniej, gdy dane wejściowe będą bardzo dobre i uzyska najlepsze sklasyfikowanie.

Poprawność sklasyfikowania danych		Punkt dzielenia danych wejściowych	Liczba epok
Uczące	Testowe		
100%	80%	15	327
100%	59.0909%	8	51
100%	60%	25	368
100%	81.8182%	19	277
100%	83.3333%	12	208
Średnia poprawność sklasyfikowania		72.8485%	

Jak widzimy najlepsze sklasyfikowanie otrzymujemy dla podziału 12 danych uczących i 18 testowych. Zauważyłem też, że możemy uzyskać poprawność sklasyfikowania 100% gdy danymi uczącymi będzie 29 próbek, a testowymi pozostała reszta.

Wagi dla punktu podziału 14:

71.6861	148.9547	-29.5993
oraz bias	95	