

**Zadanie nr 1 - uczenie i testowanie
neuronu liniowego z wieloma wzorcami
treningowymi**
Inteligentna Analiza Danych

Karol Kazusek - 254189, Sebastian Zych - 254264

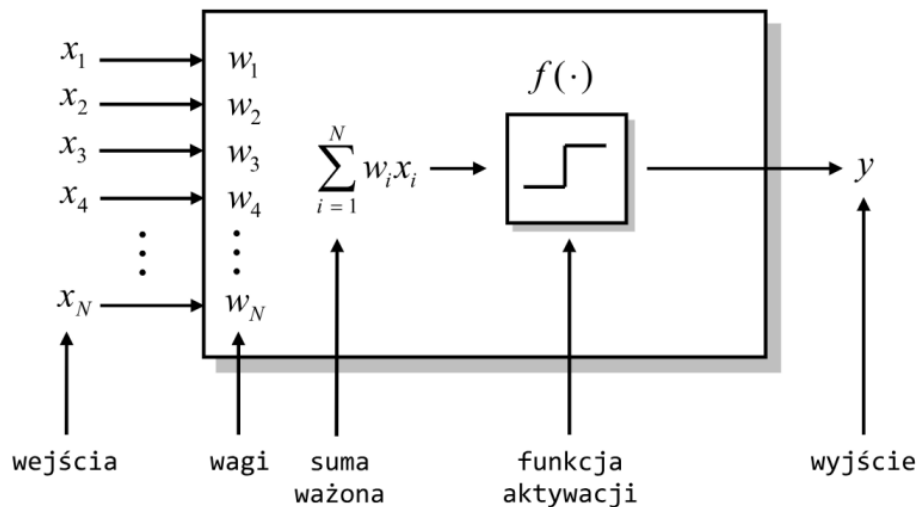
14.10.2024

1 Cel zadania

Celem zadania była implementacja neuronu liniowego, a następnie analiza wpływu różnych ilości wzorców treningowych na wyniki działania neuronu. Naszym zadaniem było kilkakrotne przeprowadzenie algorytmu treningowego przy użyciu losowych zestawów danych i startowych wag neuronów, a następnie wyciągnięcie wniosków na temat jego efektywności. Dodatkowo, celem było sprawdzenie, czy algorytm uczący skutecznie trenował neuron na danym zbiorze treningowym oraz czy trening był efektywny dla każdego z trzech przypadków.

2 Wstęp teoretyczny

Sztuczne neurony i sieci neuronowe czerpią inspirację z natury, a konkretnie z biologicznych neuronów oraz sieci neuronowych obecnych w mózgach i układach nerwowych ludzi i zwierząt. Choć niektóre typy sztucznych sieci próbują naśladować naturalne procesy, większość współczesnych rozwiązań działa na odmiennych zasadach, dostosowanych do specyficznych zadań i wymagań technicznych. Sztuczny neuron Rys. 1. jest to uproszczoną wersją swojego biologicznego odpowiednika, służącą do przetwarzania informacji, lecz jego funkcjonalność jest znacznie bardziej schematyczna niż w przypadku naturalnych neuronów.



Rysunek 1: Model sztucznego neuronu

Neuron składa się z: Wektora wejściowego

$$\mathbf{x} = [x_1, x_2, \dots, x_N] \quad (1)$$

i wag

$$\mathbf{w} = [w_1, w_2, \dots, w_N] \quad (2)$$

Wyjście jest obliczane przy pomocy wzoru:

$$\sum_{i=1}^N w_i x_i = w_1 x_1 + w_2 x_2 + \dots + w_N x_N = y \quad (3)$$

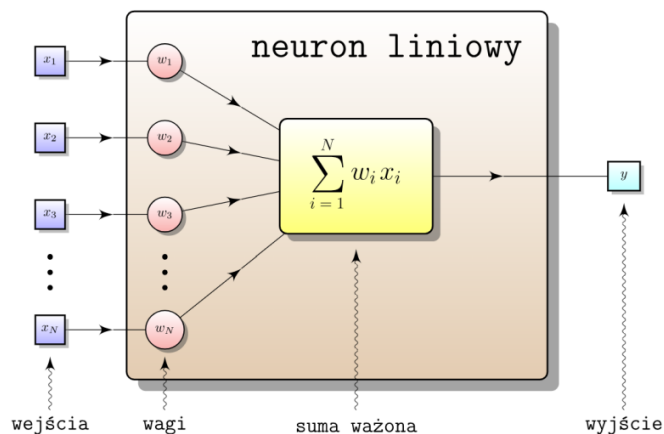
gdzie:

- y - wyjście neuronu,
- w_i - i -ta waga,
- x_i - i -te wejście,
- n - liczba wejść.

Jako funkcję aktywacji wykorzystamy funkcję liniową o postaci:

$$f(x) = a \cdot x \quad (4)$$

Neuron o takiej charakterystyce nazywamy Neuronem liniowym Rys. 2. to jeden z najprostszych typów sztucznych neuronów. Jego działanie polega na obliczeniu sumy ważonej wejść (sygnałów).



Rysunek 2: Model sztucznego neuronu liniowego

3 Eksperymenty i wyniki

3.1 Eksperyment nr 1

3.1.1 Założenia

Analiza wektora wag neuronu w kontekście wielokrotnego zastosowania algorytmu treningowego, w którym liczba wag neuronu N jest mniejsza niż liczba próbek treningowych M (czyli $N < M$).

Tabela 1: Założenia parametrów wyjściowych - eksperyment nr 1

Parametr	Wartość
Liczba wag neuronu (N)	5
Liczba wzorców treningowych (M)	10
Zakres wartości wag neuronu	$[-1, 1]$
Liczba epok (K)	14000
Krok treningowy	0.8

Tabela 2: Założenia wag dla eksperymentu nr 1

Neruron	Wagi początkowe neuronu
1	[0.7022602 0.01750665 0.79466921 0.92162914 0.90279926]
2	[0.41036501 0.54634851 0.1133011 0.26291431 0.10229738]
3	[0.35029543 0.53505635 0.51316328 0.62997295 0.84010926]
4	[0.09378692 0.3522956 0.81628097 0.48155047 0.33005786]
5	[0.97057748 0.33640186 0.21091255 0.69016781 0.7327417]

3.1.2 Przebieg

Algorytm treningowy dla neuronu liniowego został uruchomiony, korzystając z parametrów przedstawionych w Tabeli 1. Za każdym razem wykorzystano ten sam zbiór danych treningowych wygenerowane losowo. Wartości wag były inicjowane losowo przy uruchomieniu.

3.1.3 Rezultat

Tabela 3: Rezultaty eksperymentu nr 1

Neruron	Wagi końcowe neuronu
1	[1.10044389 5.19383826 0.27373934 1.73954117 -4.72341669]
2	[1.10044389 5.19383826 0.27373934 1.73954117 -4.72341669]
3	[1.10044389 5.19383826 0.27373934 1.73954117 -4.72341669]
4	[1.10044389 5.19383826 0.27373934 1.73954117 -4.72341669]
5	[1.10044389 5.19383826 0.27373934 1.73954117 -4.72341669]

Tabela 4: Wyniki 1 neuronu z wykorzystaniem zbioru treningowego jako wektory wejściowe neuronu wytrenowanego:

Numer przypadku	Klasyfikacja neurona	R. klasyfikacja
1	1.62866269	1
2	0.75101212	1
3	0.88028287	0
4	2.23147244	1
5	1.40843858	1
6	2.43287648	1
7	2.78033059	0
8	4.54720541	1
9	4.44872991	0
10	3.10839639	1

3.2 Eksperyment nr 2

Analiza wektora wag neuronu w kontekście wielokrotnego zastosowania algorytmu treningowego, w którym liczba wag neuronu N jest równa liczbie próbek treningowych M (czyli $N = M$).

Tabela 5: Założenia parametrów wyjściowych - eksperyment nr 2

Parametr	Wartość
Liczba wag neuronu (N)	5
Liczba wzorców treningowych (M)	5
Zakres wartości wag neuronu	[-1, 1]
Liczba epok (K)	14000
Krok treningowy	0.8

Tabela 6: Założenia wag dla eksperymentu nr 2

Neruron	Wagi początkowe neuronu
1	[0.25718081 0.23160417 0.12374798 0.2100194 0.43299329]
2	[0.9268277 0.03963894 0.59703405 0.13471647 0.06900094]
3	[0.30719827 0.43147593 0.11097351 0.13570207 0.27112304]
4	[0.63129549 0.00481451 0.72005156 0.74442693 0.73818147]
5	[0.23434677 0.5381695 0.76920115 0.57128691 0.6599291]

3.2.1 Przebieg

Algorytm treningowy dla neuronu liniowego został uruchomiony, korzystając z parametrów przedstawionych w Tabeli 5. Za każdym razem wykorzystano ten sam zbiór danych treningowych wygenerowane losowo. Wartości wag były inicjowane losowo przy uruchomieniu.

3.2.2 Rezultat

Tabela 7: Rezultaty eksperymentu nr 2

Neruron	Wagi końcowe neuronu
1	[1.75545992 0.72352631 -0.45735995 0.30204395 -0.62659508]
2	[1.75545992 0.72352631 -0.45735995 0.30204395 -0.62659508]
3	[1.75545992 0.72352631 -0.45735995 0.30204395 -0.62659508]
4	[1.75545992 0.72352631 -0.45735995 0.30204395 -0.62659508]
5	[1.75545992 0.72352631 -0.45735995 0.30204395 -0.62659508]

Tabela 8: Wyniki 1 neuronu z wykorzystaniem zbioru treningowego jako wektory wejściowe neuronu wytrenowanego:

Numer przypadku	Klasyfikacja neuronu	R. klasyfikacja
1	1.00000000e+00	1
2	1.00000000e+00	1
3	-4.62567283e-16	0
4	1.00000000e+00	1
5	1.00000000e+00	1

3.3 Eksperyment nr 3

Analiza wektora wag neuronu w kontekście wielokrotnego zastosowania algorytmu treningowego, w którym liczba wag neuronu N jest większa niż liczba próbek treningowych M (czyli $N > M$).

Tabela 9: Założenia parametrów wyjściowych - eksperyment nr 3

Parametr	Wartość
Liczba wag neuronu (N)	5
Liczba wzorców treningowych (M)	2
Zakres wartości wag neuronu	[-1, 1]
Liczba epok (K)	14000
Krok treningowy	0.8

Tabela 10: Założenia wag dla eksperymentu nr 3

Neruron	Wagi początkowe neuronu
1	[0.042668 0.24044789 0.39155331 0.72852958 0.18479858]
2	[0.82432786 0.08099875 0.74098371 0.63442987 0.18990254]
3	[0.47082526 0.67922145 0.38200882 0.77416749 0.02424662]
4	[0.63795142 0.32744639 0.56587628 0.52658763 0.81397631]
5	[0.44989125 0.83818744 0.62580719 0.2009764 0.86055305]

3.3.1 Przebieg

Algorytm treningowy dla neuronu liniowego został uruchomiony, korzystając z parametrów przedstawionych w Tabeli 9. Za każdym razem wykorzystano ten sam zbiór danych treningowych wygenerowane losowo. Wartości wag były inicjowane losowo przy uruchomieniu.

3.3.2 Rezultat

Tabela 11: Rezultaty eksperymentu nr 3

Neruron	Wagi końcowe neuronu
1	[0.10632304 0.80405649 0.4017791 0.45961974 0.28680895]
2	[0.70692232 0.59012639 0.63854747 0.09975784 0.12163089]
3	[0.38247018 0.85732429 0.31598488 0.5082556 -0.04300429]
4	[0.43267451 0.5126257 0.42481907 0.06093346 0.64003502]
5	[0.27104984 0.54160379 0.52762757 0.10003904 0.67356591]

Tabela 12: Wyniki 1 neuronu z wykorzystaniem zbioru treningowego jako wektory wejściowe neuronu wytrenowanego:

Numer przypadku	Klasyfikacja neurona	R. klasyfikacja
1	1	1
2	1	1

4 Wnioski

Wnioski z przeprowadzonych eksperymentów dowodzą, że

- Na wyjściu dla przypadku $N < M$ wyniki znacznie różniły się od oczekiwanych wartości dla wzorców treningowych. Powód można znaleźć w nie dostatecznej ilości wag.
- Dla przypadku $N = M$ oraz $N > M$ wartość końcowych wag nie są zmienne. Oznacza to brak przeszkód w nauce wzorców oraz przewidywań wartości wag. W przypadku $N < M$ nie jest w stanie skutecznie nauczyć wzorzec. Zwraca ona różne wartości wag.
- Dało się zaobserwować podobieństwo działania neuronu liniowego do problemu rozwiązywania układu równań z wieloma niewiadomymi.

Bibliografia