Metody Inteligencji Obliczeniowej
Laboratorium 3

Aproksymacja funcji za pomocą sieci neuronowych

Kamil Pyla

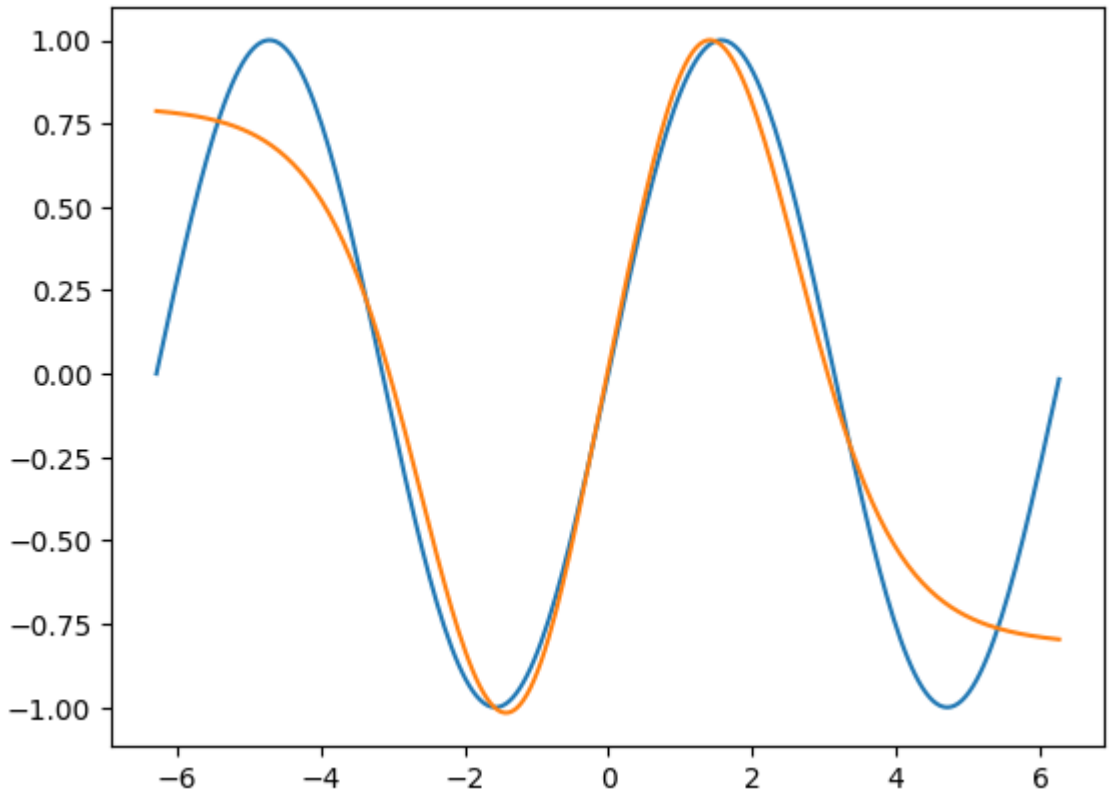
Wyniki zadania 1:

funkcja aktywacji	ilość iteracji	ilość neuronów w warswie	ilość warstw	dokładność dopasowania	
				train	test
relu	2000	80	5	91,83	94,72
relu	1000	50	5	81,97	86,7
relu	2000	60	5	88,80	92,93
tanh	2000	80	5	84,83	87,67
tanh	1000	50	5	84,81	88,37
tanh	1000	60	5	58,56	75,53
tanh	2000	60	8	86,24	89,82

Wnioski: Wyniki otrzymane w ćwiczeniu bywały bardzo rozbieżne. Dla tych samych konfiguracji sieci otrzymywałem skrajne wartości. Jednak dla sieci złożonej z większej ilości neuronów i warstw wahania były mniejsze. W tym

przypadku funkcja aktywacji relu osiągnęła większą dokładność niż tanh jednak ten wniosek dotyczy tylko obserwacji tego przykładu.

Wyniki zadania 2:



Dokładność dopasowania: 90,73 %

coefs = [array([[0.83114892, 0.81899223, 0.60681264]]), array([[-1.09696669],
[1.75706497],
[-1.44599947]])]

intercepts = [array([2.23714592, -0.11238383, -1.42722978]),
array([-0.03218777])]

Odtworzony wzór funkcji:

$$f(x) = -1.097tanh(0.8311x + 2.237) + 1.757tanh(0.819x - 0.112) - -1.097tanh(0.607x - 1.427) - 0.032$$

Sądząc po postaci funkcji, można wyciągnąć wnioski, że sieć neuronowa rozwija funkcję sin(x) w szereg.

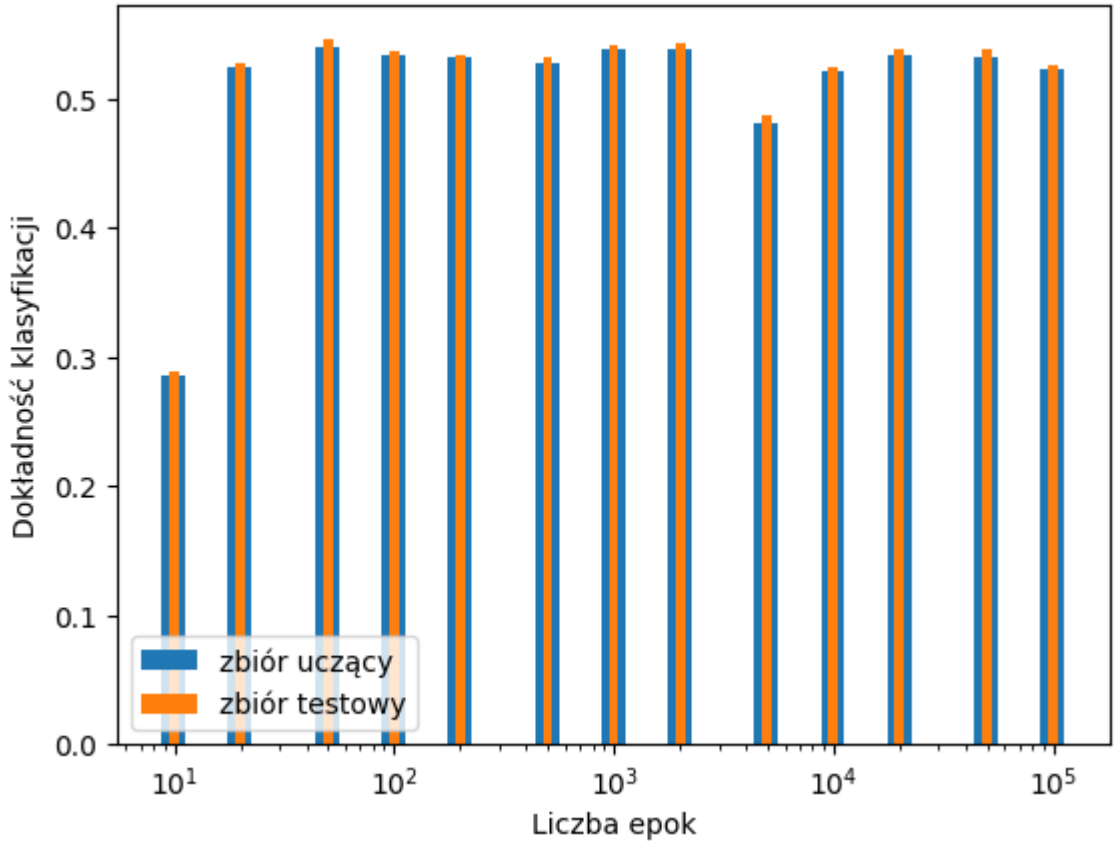
tanh = $= \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$ zatem możliwe jest rozwinięcie funkcji sinus używając tanh

Wyniki zadania 3:

funkcja aktywacji	ilość iteracji	ilość neuronów w warswie	ilość warstw	R^2	
				train	test
tanh	2000	50	5	0.684	0.696
tanh	2000	50	6	0.663	0.679
tanh	2000	80	5	0.645	0.651
tanh	5000	30	4	0.635	0.639
tanh	3000	30	4	0.611	0.621
tanh	10000	10	3	0.6	0.61
relu	2000	50	5	0.579	0.590
relu	2000	50	6	0.595	0.608
relu	2000	80	5	0.559	0.568
relu	2000	80	6	0.554	0.564

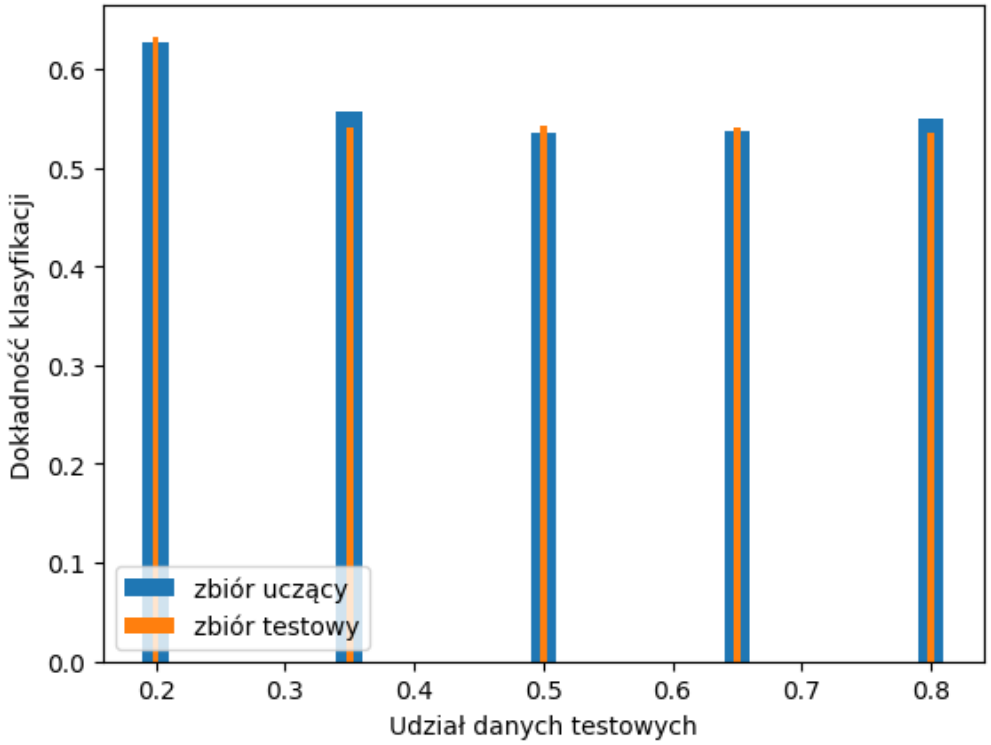
Wnioski:
Wraz ze wzrostem ilości warstw i ilości neuronów zwiększa się czas wykonywania programu, jednak dokładność dopasowania od pewnego momentu nieznacznie się zmienia, lepiej jest zatem zaczynać od mniejszej ilości neuronów w sieci i zwiększać ilość iteracji, ponieważ to może dać nam wynik stosunkowo dokładny przy znacznie którszym czasie wykonania.

Wyniki zadania 4:



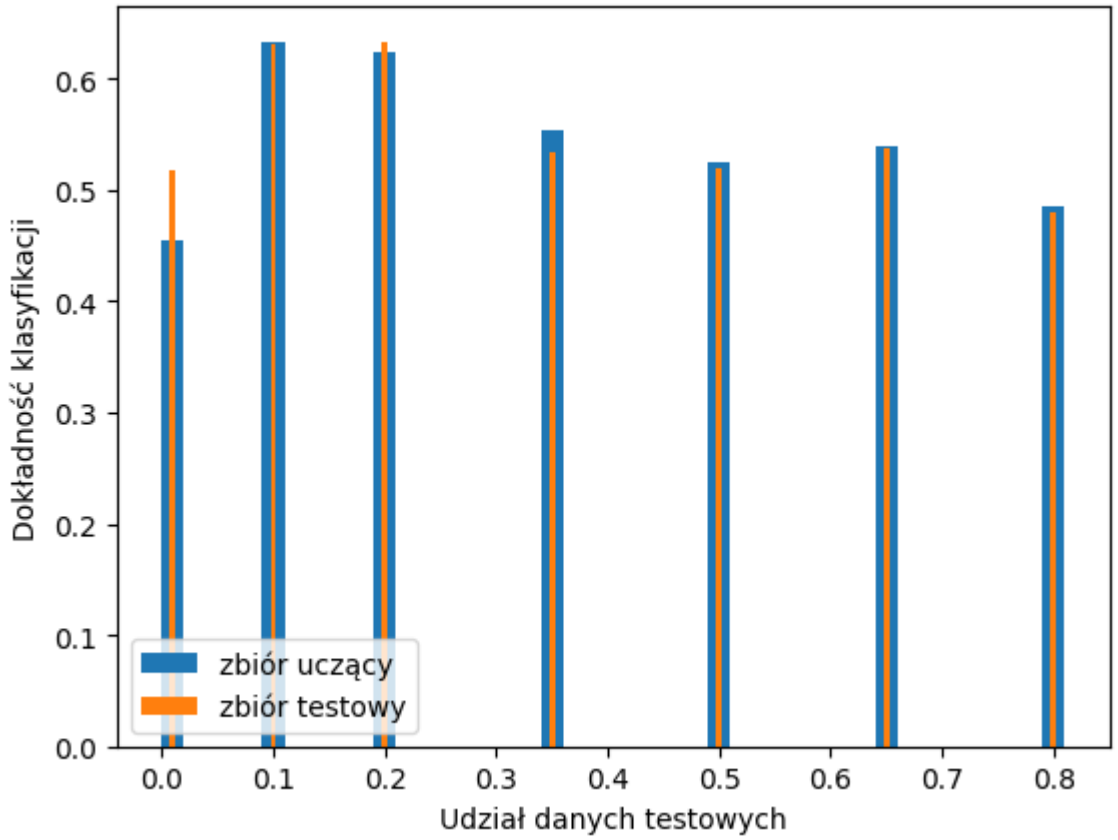
Wnioski: Zwiększanie ilości epok również nie zawsze skutkuje zwiększeniem dokładności, należy szukać rozwiązania które prowadzi nas do jak najlepszego stosunku dokładności do czasu wykonania

Wyniki zadania 5:



Wnioski: Wyraźnie widać, że w zbiorze zawierającym 80 % danych uczących dopasowanie wyszło największe. W przypadku aproksymacji jest do uzasadnione, im więcej punktów uwzględnimy w aproksymacji, tym większą dokładność uzyskamy. Jednak należy pamiętać o tym, że dzięki podziałowi danych możemy pozbyć się punktów, które odbiegają od wykresu funkcji, co może przełożyć się na dokładniejszą aproksymację. Oczywiście dokładność zależy również od jakości danych które użyjemy do uczenia.

Dokładność gdy cały zbiór użyliśmy do uczenia: 0.6384183701528914



Link do repozytorium z kodem:
https://github.com/KamilPyla/MIO_2023/tree/master/lab_03