

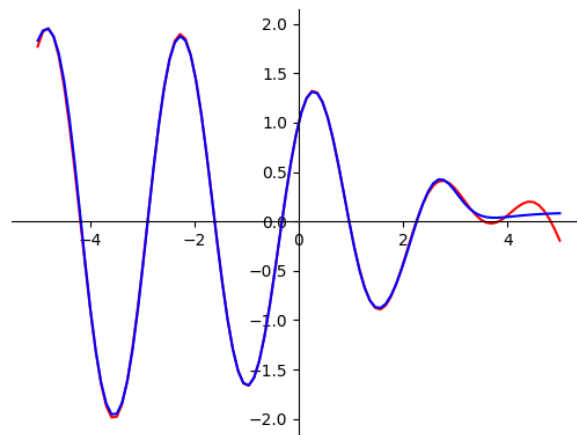
## WSI - Ćwiczenie 5

Ostatnie cyfry indeksów – 6, 4.

Najlepszy wynik, który udało się osiągnąć: squared error – 0.24, mean absolute error – 0.03

Parametry: liczba neuronów=13, learning rate=0.05, liczba iteracji=100000, batch size=5.

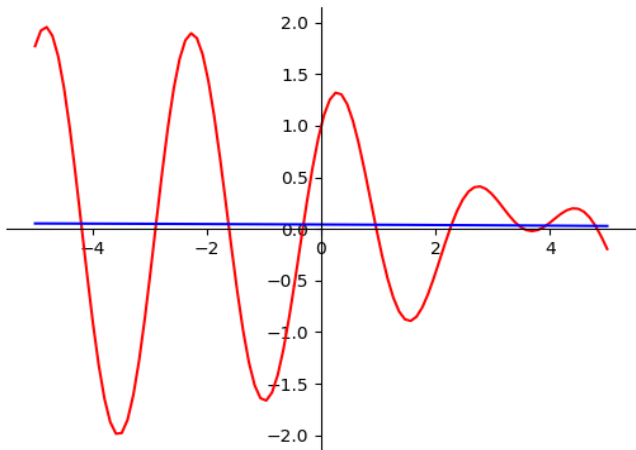
Hidden layout size: 13, learning rate: 0.05

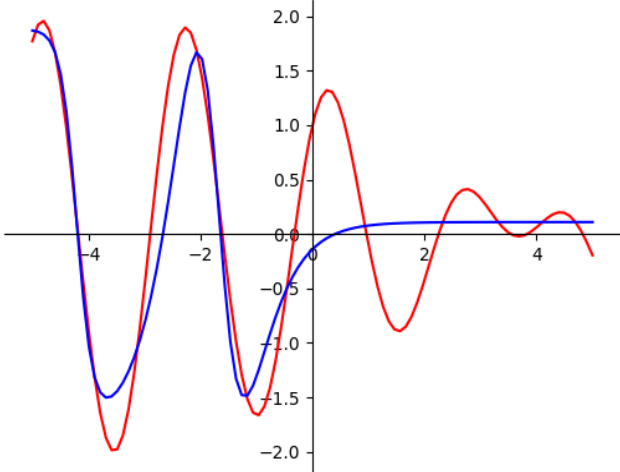
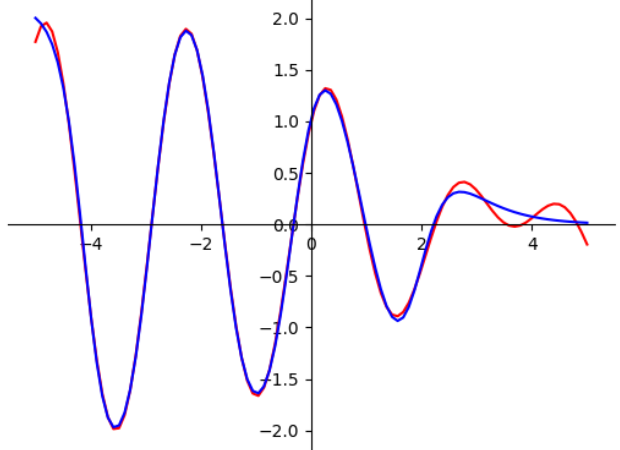
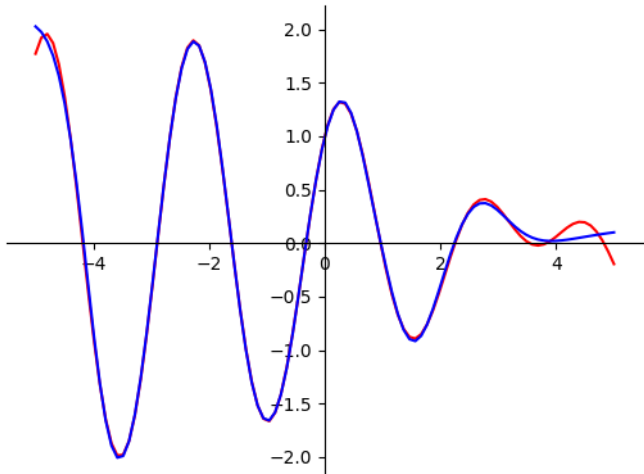


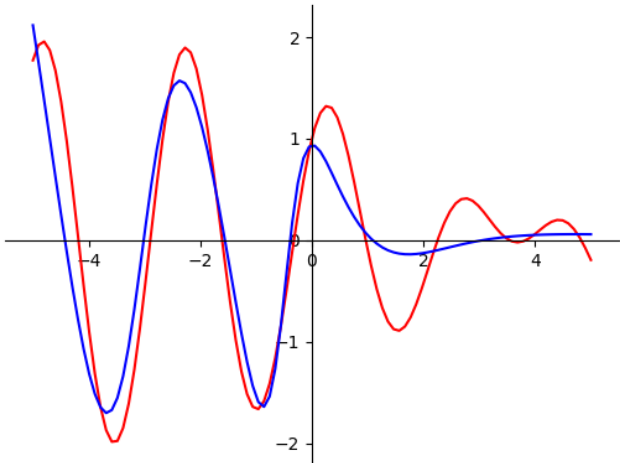
Odpowiedzi na pytania:

1) Jak liczba neuronów w warstwie ukrytej wpływa na jakość aproksymacji?

Domyślne parametry: rozmiar zbioru - 100, liczba epok – 40000, learning rate – 0,01, batch size - 1.

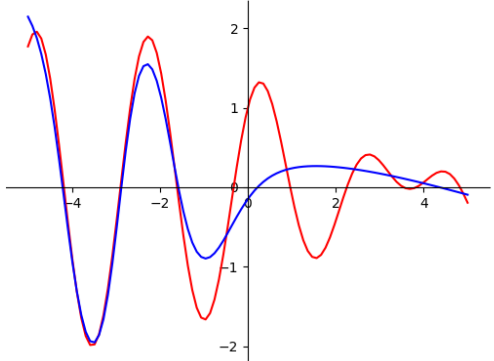
Liczba neuronów	Jakość aproksymacji	Wykres
1	<b>Squared error: 102.8</b> <b>Mean absolute error: 0.79</b>	<p>Hidden layout size: 1, learning rate: 0.01</p> 

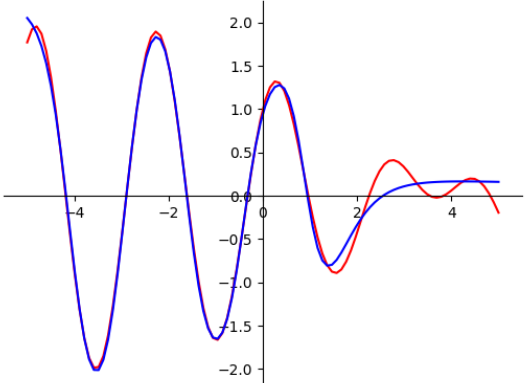
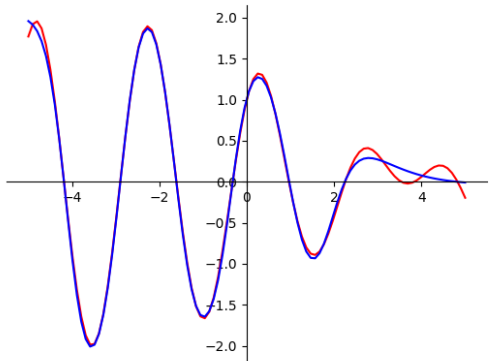
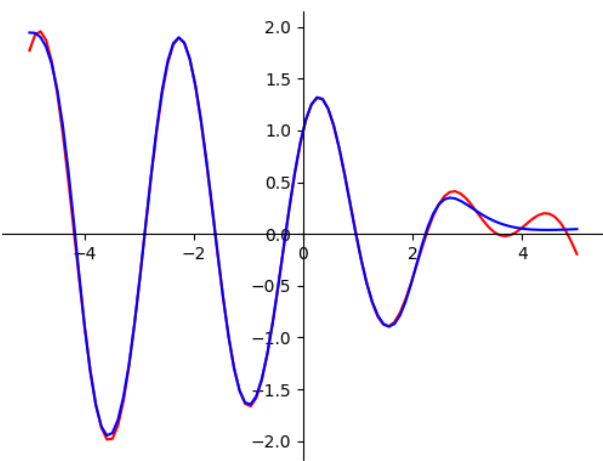
3	<b>Squared error: 27.86</b> <b>Mean absolute error: 0.38</b>	<p>Hidden layout size: 3, learning rate: 0.01</p> 
6	<b>Squared error: 0.43</b> <b>Mean absolute error: 0.045</b>	<p>Hidden layout size: 6, learning rate: 0.01</p> 
20	<b>Squared error: 0.33</b> <b>Mean absolute error: 0.035</b>	<p>Hidden layout size: 20, learning rate: 0.01</p> 

80	<b>Squared error: 83.58</b> <b>Mean absolute error: 0.33</b>	<p>Hidden layout size: 80, learning rate: 0.01</p> 
----	---	---

Z powyższej tabeli można wywnioskować, że przy zwiększeniu liczby neuronów w warstwie ukrytej, zwiększa się jakość aproksymacji. Dla stosunkowo małych wartości (np. 1, 3, 6) jakość aproksymacji poprawia się w sposób znaczący. Natomiast przy dalszym zwiększeniu liczby neuronów możemy zauważyć pogorszenie się jakości wyników. Z tego możemy wywnioskować, że większa liczba neuronów nie oznacza lepsze wyniki (występuje problem przeuczenia się modelu oraz zwiększa się czas trenowania).

2) Wpływ liczby iteracji:

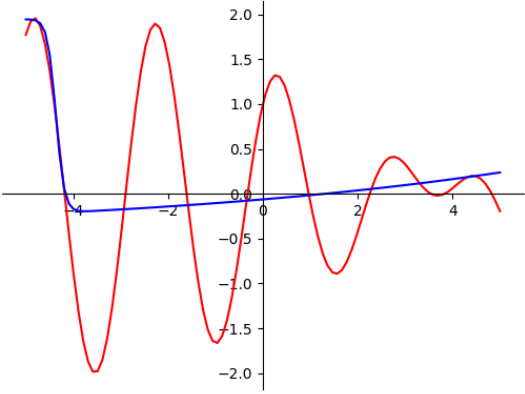
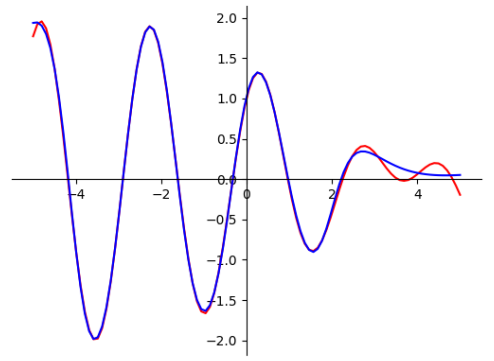
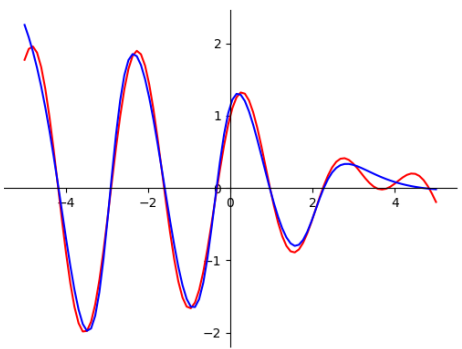
Liczba iteracji	Jakość aproksymacji	Wykres
2000	<b>Squared error: 43.87</b> <b>Mean absolute error: 0.38</b>	<p>Hidden layout size: 20, learning rate: 0.01</p> 

5000	<b>Squared error: 1.73</b> <b>Mean absolute error: 0.08</b>	<p>Hidden layout size: 20, learning rate: 0.01</p> 
10000	<b>Squared error: 0.48</b> <b>Mean absolute error: 0.05</b>	<p>Hidden layout size: 20, learning rate: 0.01</p> 
20000	<b>Squared error: 0.31</b> <b>Mean absolute error: 0.039</b>	<p>Hidden layout size: 20, learning rate: 0.01</p> 

Jak widać, im więcej iteracji tym lepiej wynik. Ale po osiągnięciu określonej jakości aproksymacji wynik prawie przestaje rosnąć (jak w przypadku z 10000 i 20000 iteracji).

3) Wpływ batch size.

Batch size	Jakość aproksymacji	Wykres
------------	---------------------	--------

1	<b>Squared error: 83.3</b> <b>Mean absolute error: 0.67</b>	<p>Hidden layout size: 13, learning rate: 0.05</p> 
8	<b>Squared error: 0.39</b> <b>Mean absolute error: 0.038</b>	<p>Hidden layout size: 13, learning rate: 0.05</p> 
48	<b>Squared error: 1.9</b> <b>Mean absolute error: 0.12</b>	<p>Hidden layout size: 13, learning rate: 0.05</p> 

Jak widać, batch size dla każdej kombinacji learning rate oraz liczby neuronów trzeba odpowiednio dobierać. Bardzo mała wartość batch size (zarówno jak i bardzo duża) skutkuje tym że algorytm będzie działał gorzej. Np. dla kombinacji  $lr=0.01$  oraz liczby neuronów=13 optymalny batch size =1. Natomiast dla kombinacji  $lr=0.05$  i tej samej liczby neuronów optymalny batch size=8.

Zwiększenie batch size (zamiast zmniejszenia learning rate) zmniejsza liczbę obliczeń oraz czas trenowania modelu.