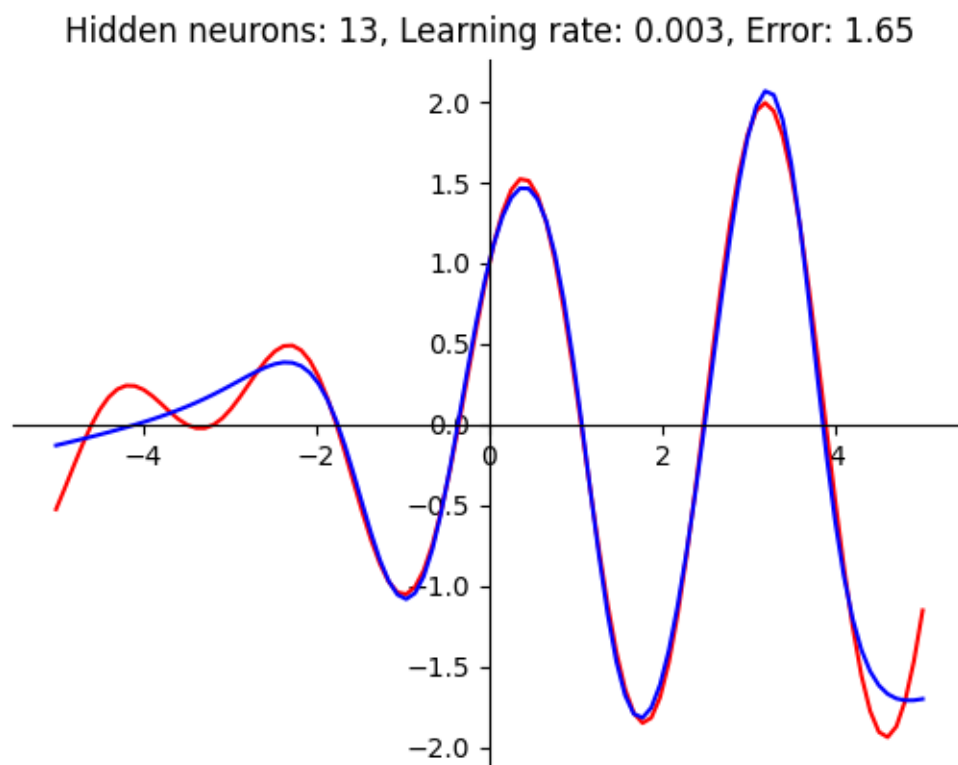


## Polecenie:

Proszę zaimplementować perceptron dwuwarstwowy i nauczyć go reprezentować funkcję  $J : [-5, 5] \rightarrow \mathbb{R}$ , daną wzorem:  $J(x) = \sin(x \cdot \sqrt{p[0]+1}) + \cos(x \cdot \sqrt{p[1]+1})$ , gdzie  $p[0]$  i  $p[1]$  to najmłodsze cyfry numerów indeksów wykonawców.

Ostatnie cyfry indeksów – 5, 3

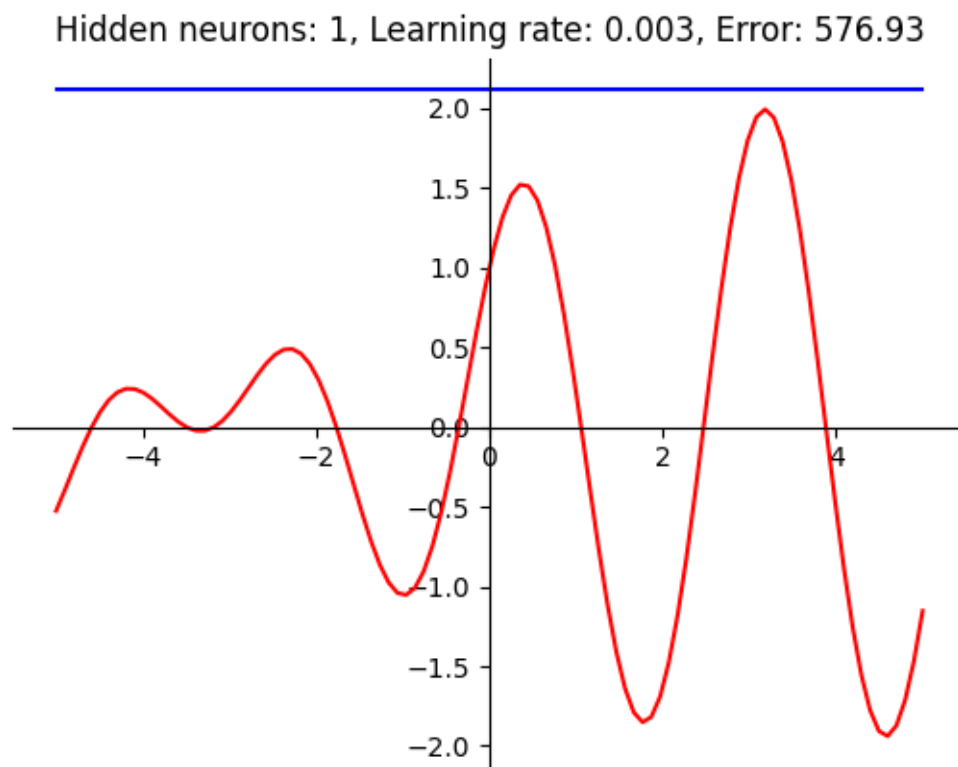
Wynik dla parametrów początkowych: liczba neuronów = 9, learning rate = 0.003, liczba iteracji = 15000



Jak liczba neuronów w warstwie ukrytej wpływa na jakość aproksymacji?

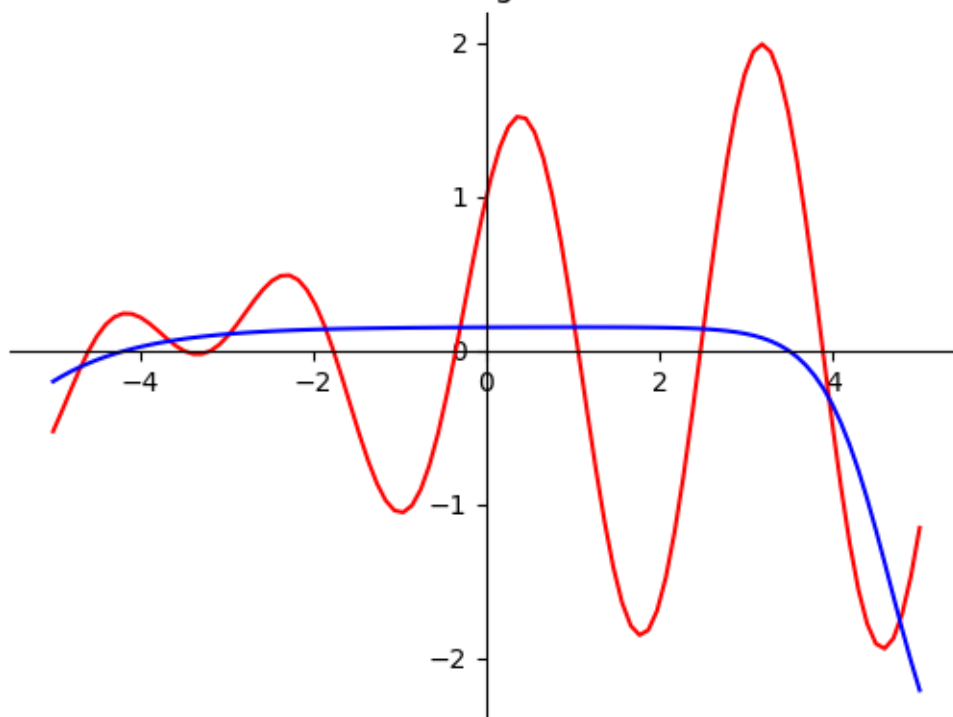
Testy:

- Dla 1 neurona:



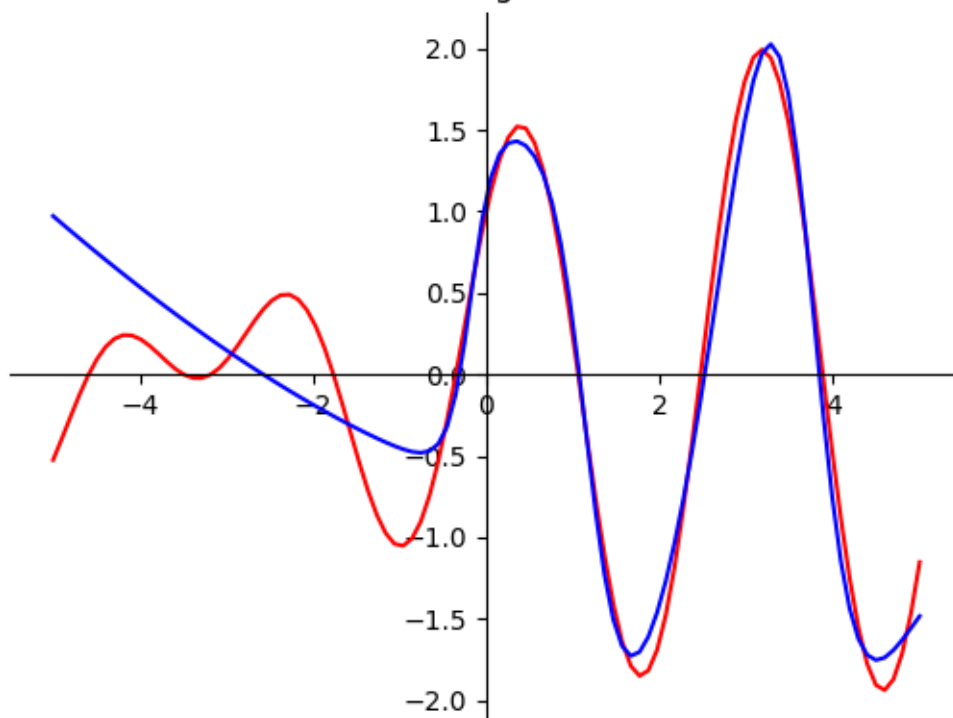
- Dla 3 neuronów:

Hidden neurons: 3, Learning rate: 0.003, Error: 83.41



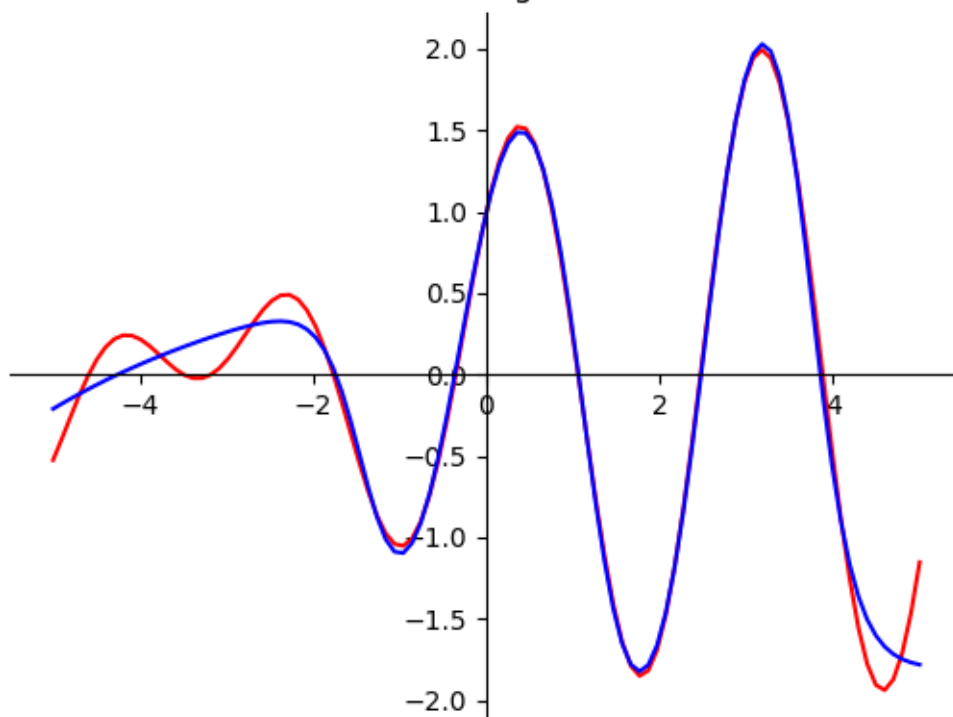
- Dla 5 neuronów:

Hidden neurons: 5, Learning rate: 0.003, Error: 14.53



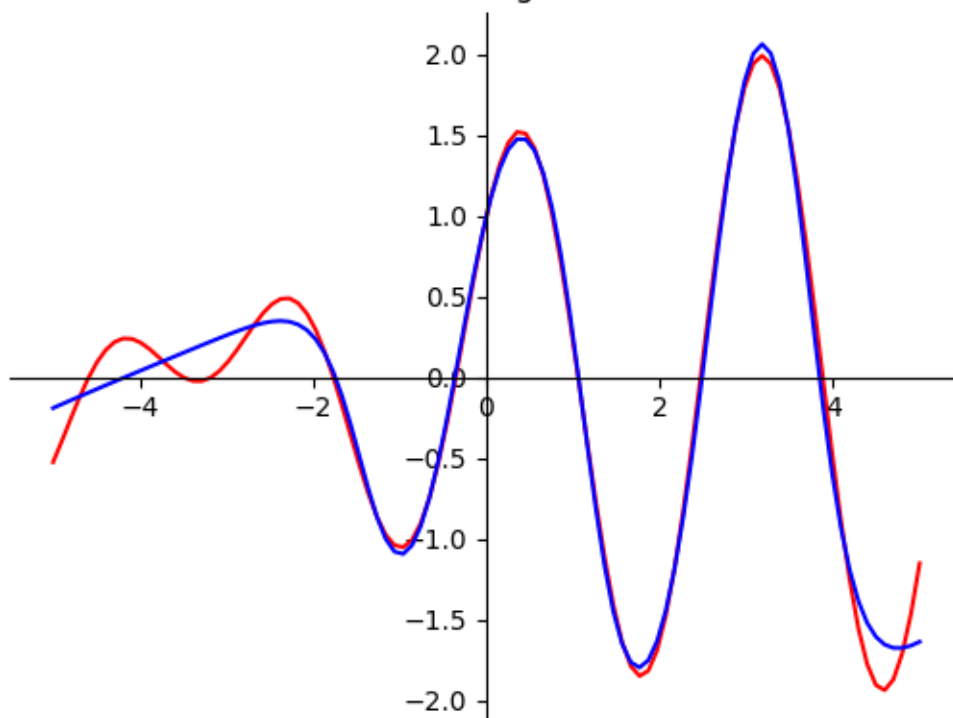
- Dla 7 neuronów:

Hidden neurons: 7, Learning rate: 0.003, Error: 1.7



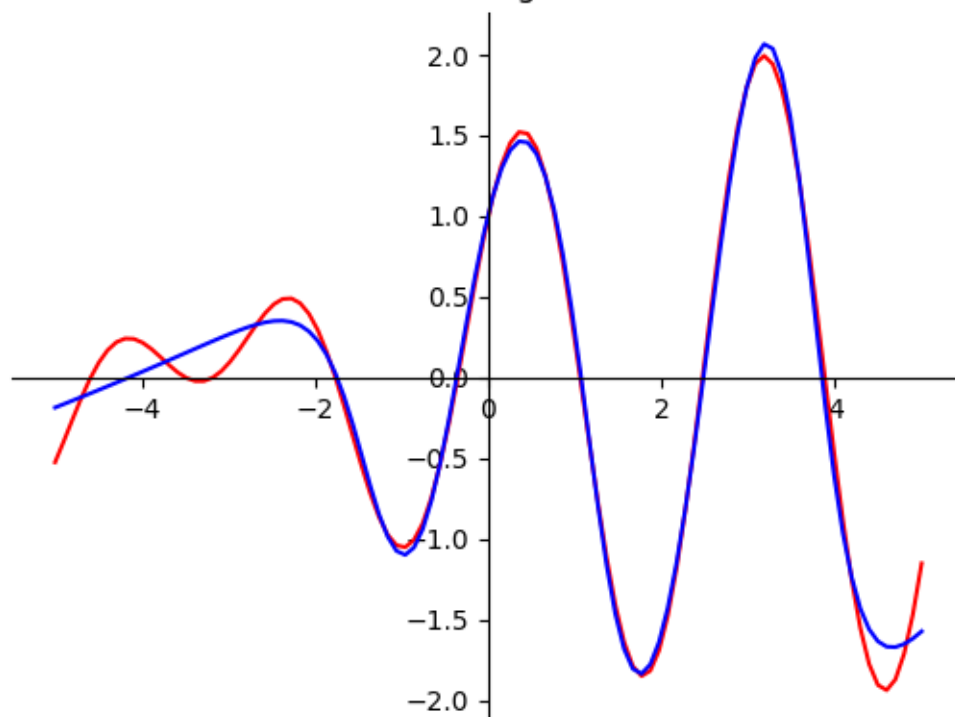
- Dla 12 neuronów:

Hidden neurons: 12, Learning rate: 0.003, Error: 1.59



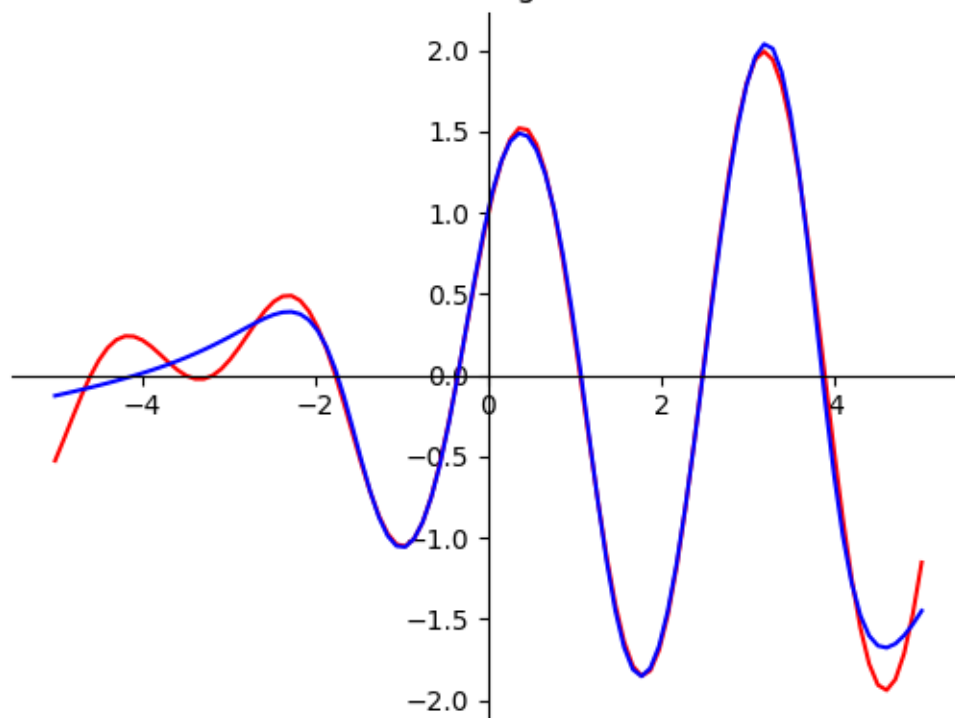
- Dla 15 neuronów:

Hidden neurons: 15, Learning rate: 0.003, Error: 1.49



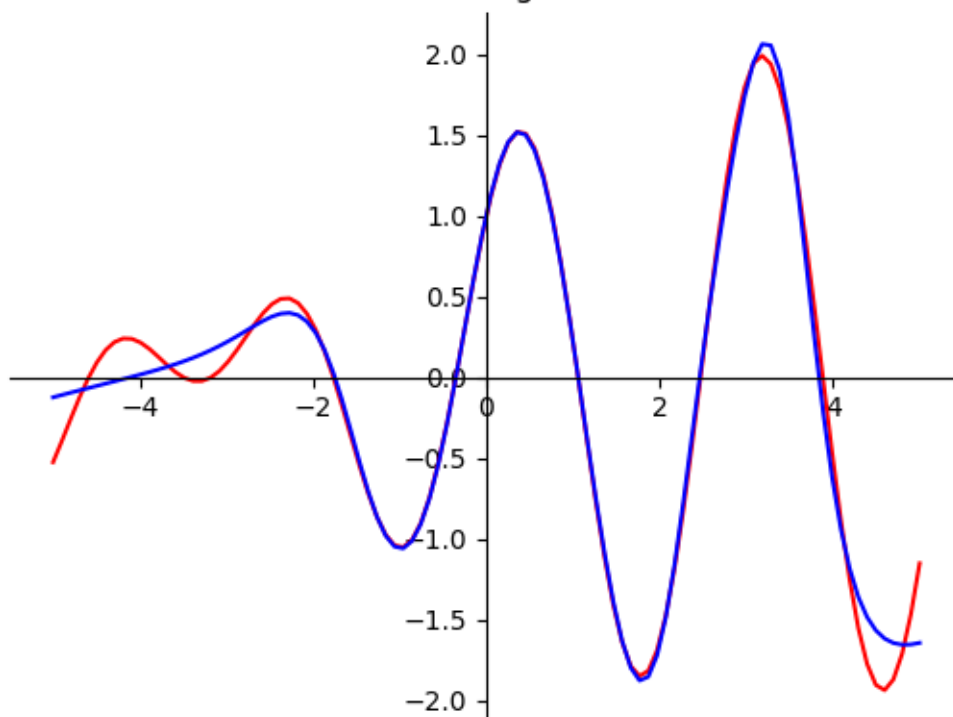
- Dla 20 neuronów:

Hidden neurons: 20, Learning rate: 0.003, Error: 1.23



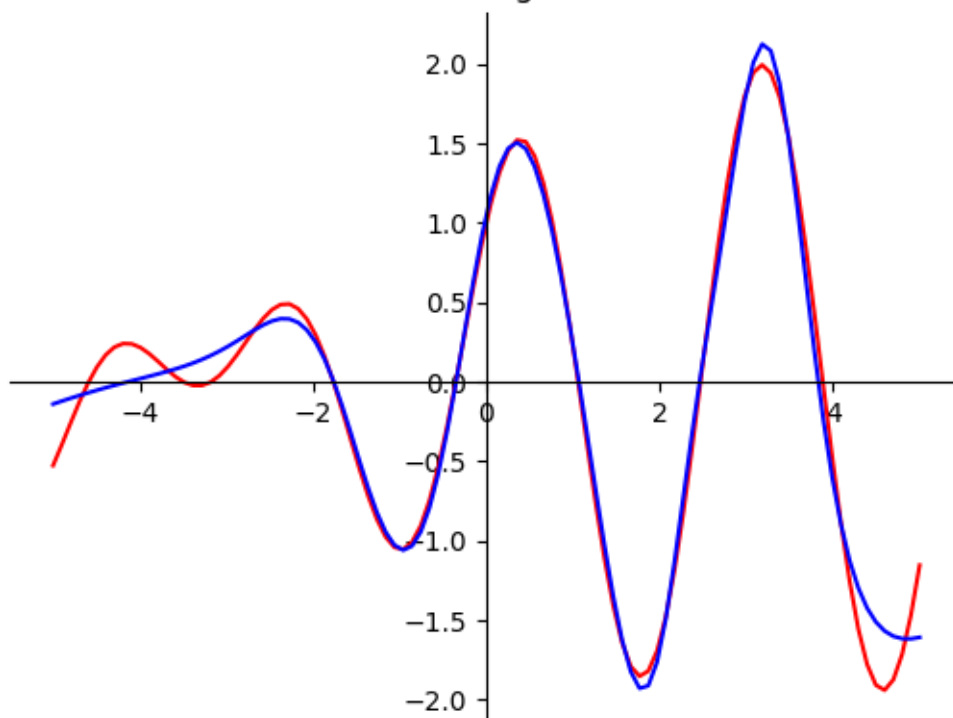
- Dla 30 neuronów:

Hidden neurons: 30, Learning rate: 0.003, Error: 1.62



- Dla 40 neuronów:

Hidden neurons: 40, Learning rate: 0.003, Error: 1.94



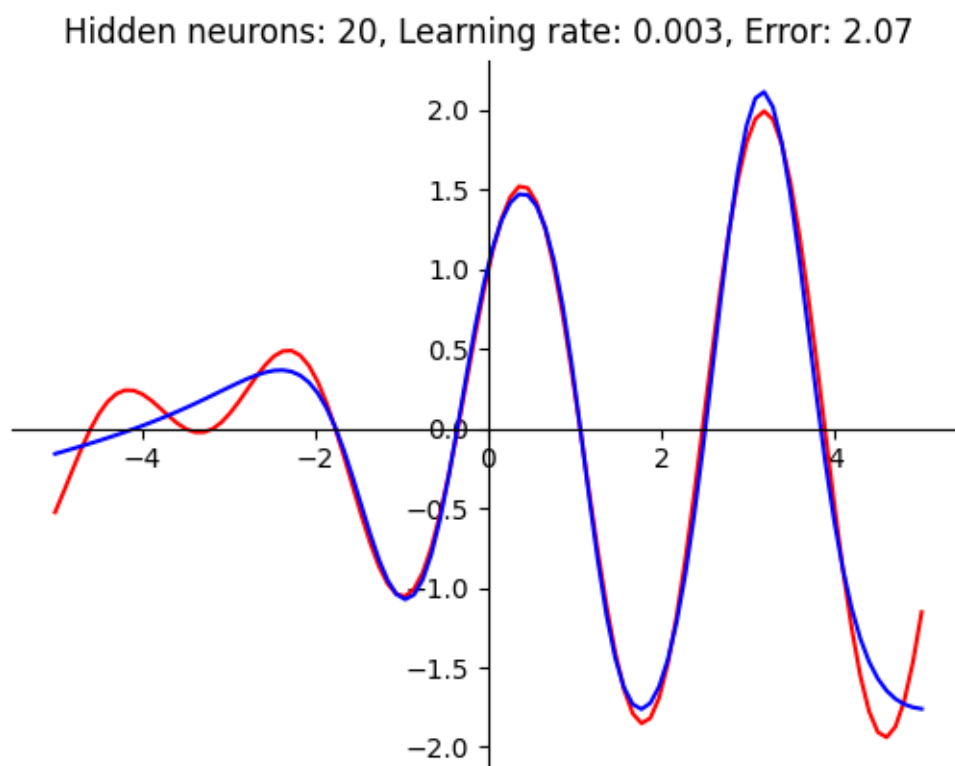
Ilość neuronów w warstwie ukrytej	1	3	5	7	9	12	15	20	30	40
Błąd	576,93	83,41	14,53	1,7	1,65	1,59	1,49	1,23	1,62	1,94

Najlepszy wynik został osiągnięty dla 20 neuronów.

Dla tej liczby testowaliśmy zmianą ilości iteracji.

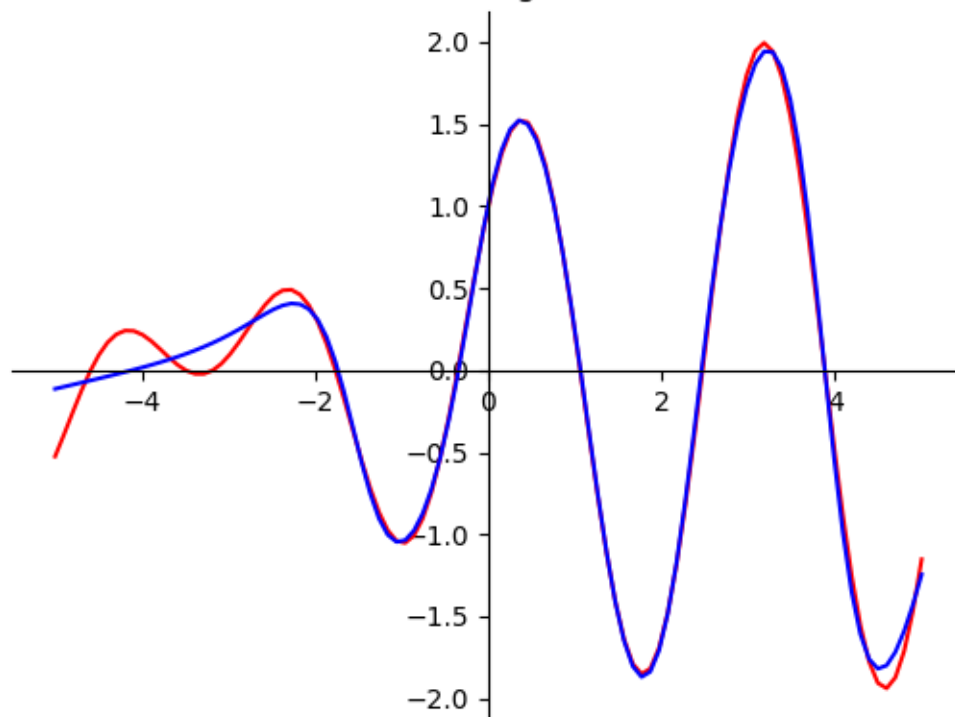
Testy:

- Dla 7,5 tys. iteracji:



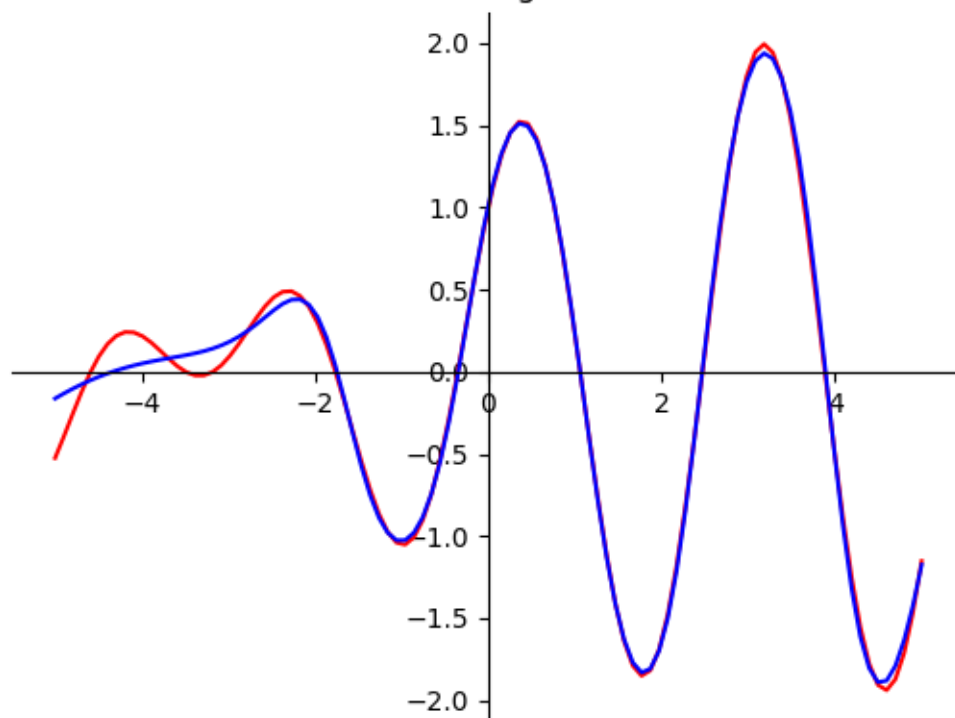
- Dla 30 tys. iteracji:

Hidden neurons: 20, Learning rate: 0.003, Error: 0.96



- Dla 60 tys. iteracji:

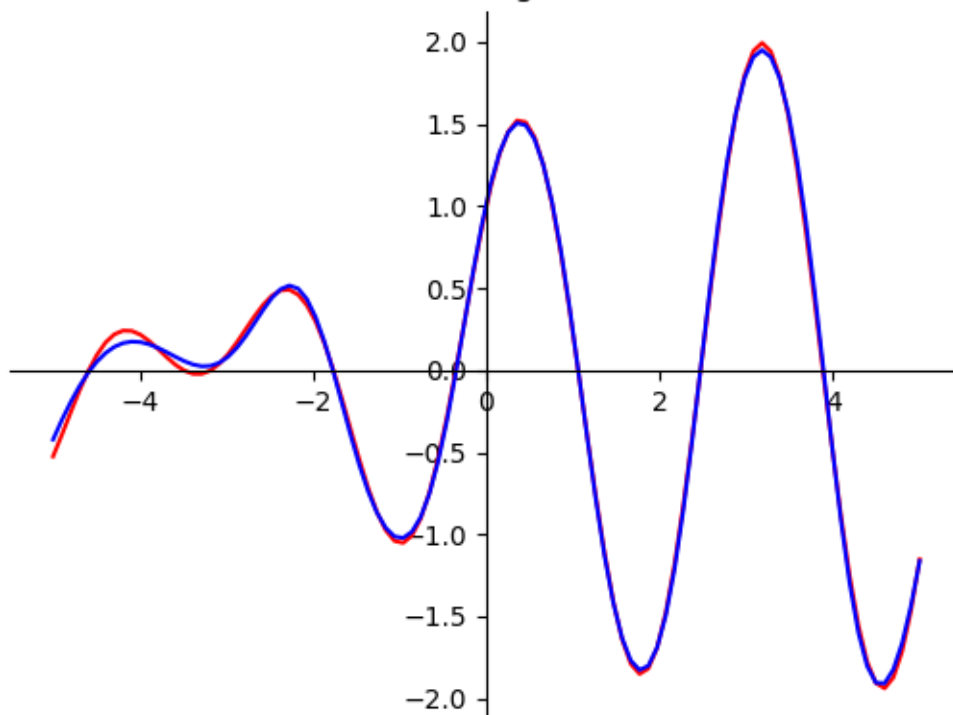
Hidden neurons: 20, Learning rate: 0.003, Error: 0.65





- Dla 90 tys. iteracji:

Hidden neurons: 20, Learning rate: 0.003, Error: 0.12



Liczba iteracji [tys]	7,5	15	30	60	90
Błąd	2,07	1,23	0,96	0,65	0.12

Najlepszy wynik został osiągnięty dla największej testowanej liczby iteracji i 20 neuronów w warstwie ukrytej. Sieć praktycznie idealnie odwzorowuje aproksymowaną funkcję.

### Wnioski:

Liczba neuronów potrzebnych do dobrej aproksymacji problemu jest zależna od jego skomplikowania. Gdy jest ich zbyt mało (w testowanym przypadku 1, 3, 5 neuronów) sieć będzie niedokładna. Można zauważyć, że za duża ilość neuronów również może negatywnie wpłynąć błąd sieci (dla 30, 40 neuronów zaczyna być widoczny wzrost błędu), może wynikać to z przewymiarowania i przeuczenia sieci, i zdecydowanie wydłużyć czas obliczeń.

Z testów modyfikacji liczby iteracji, można zauważyć, że gdy jest ich za mało to sieć nie ma wystarczająco dużo czasu, żeby się nauczyć.