Kacper Ptaszek

Documentation

A two-layer perceptron is implemented and is taught to represent the function

[-5;5] -> R: J(x) = sin(x*$\sqrt{2}$)+cos(x*$\sqrt{8}$)

The number of input samples is 300.

The approximation quality indicator that we used to compare the obtained results is the **mean square error (MSE)** and **the maximum error**.

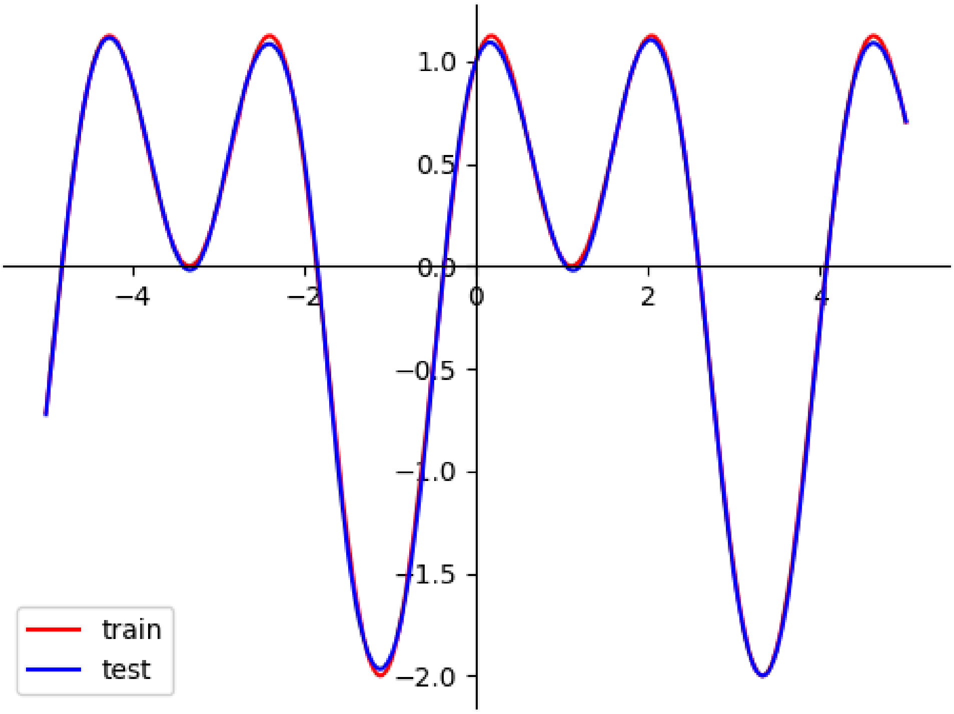
Parameters for the best approximation of the function and its graph:

Number of iterations: 15000

Number of neurons: 13

learning rate: 0.1

mini-batch size: 10



MSE: 0,00042

Maximum error: 0,051

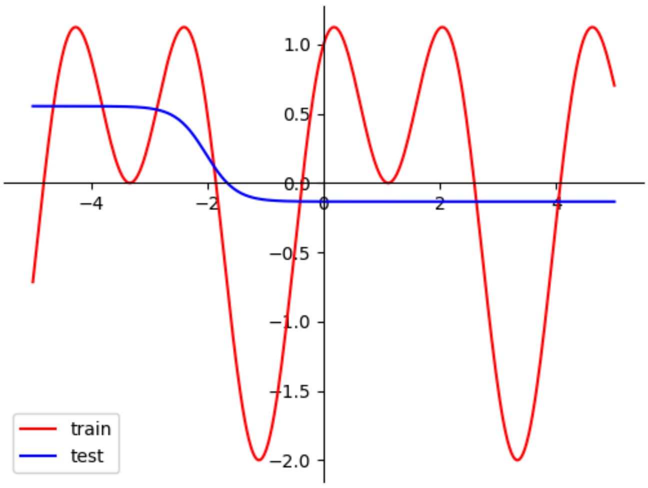
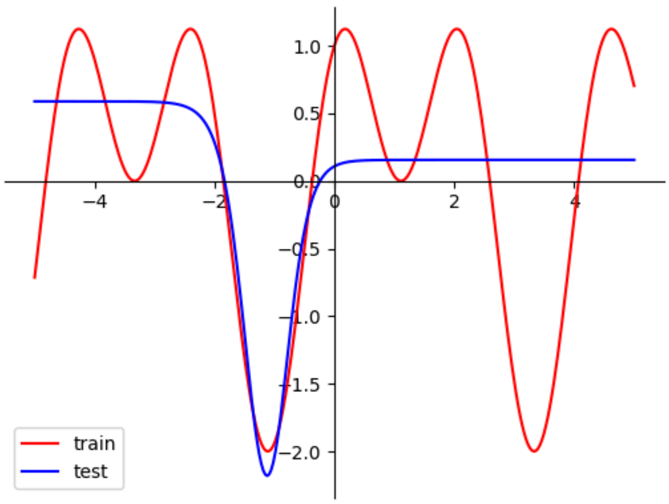
How does the number of neurons affect the quality of the approximation?

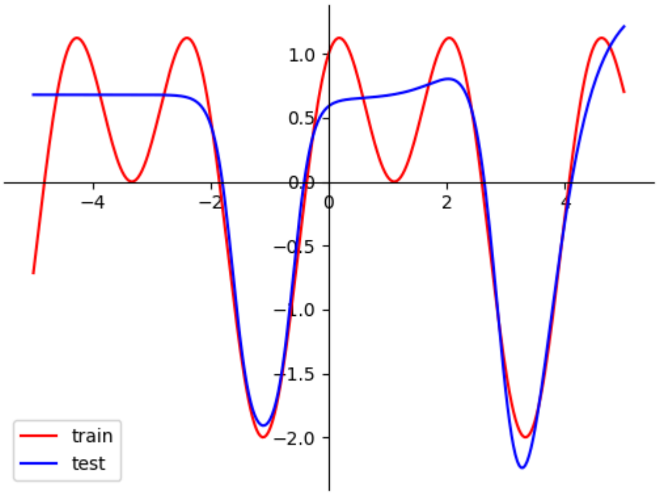
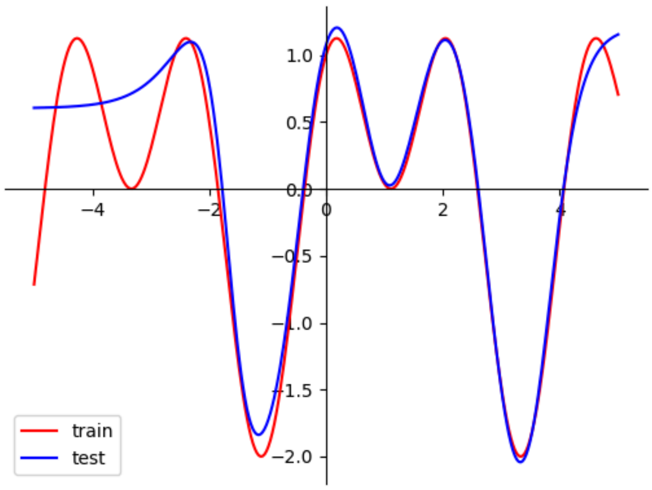
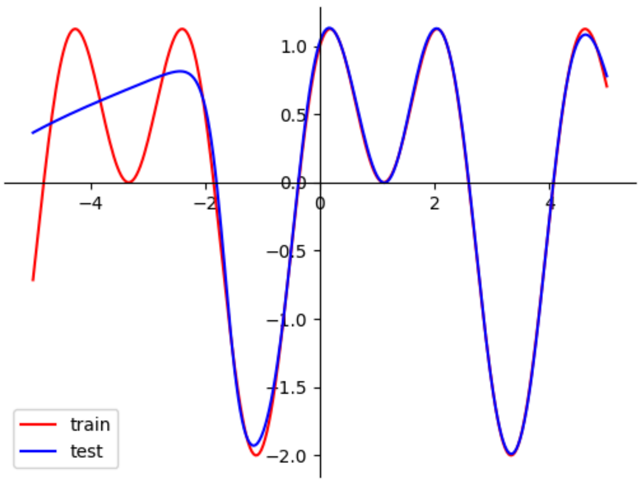
For each measurement, the parameters are:

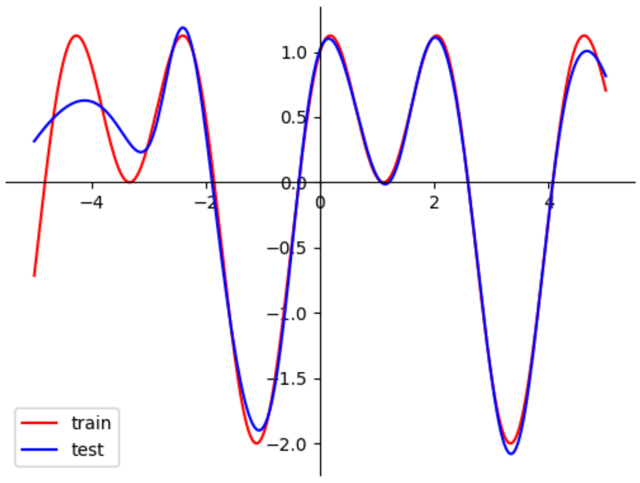
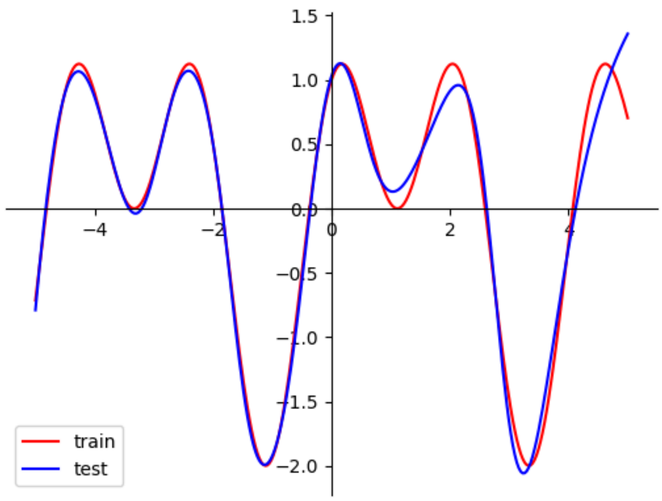
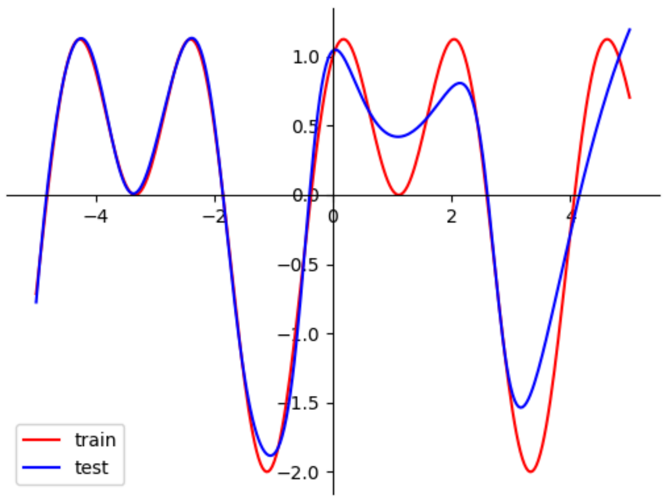
number of iterations: 5000

learning rate: 0.1

mini-batch size: 10

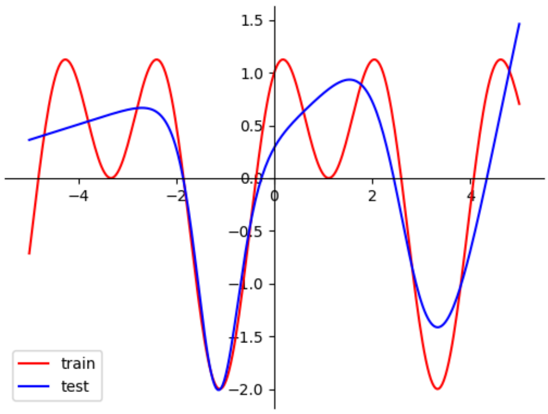
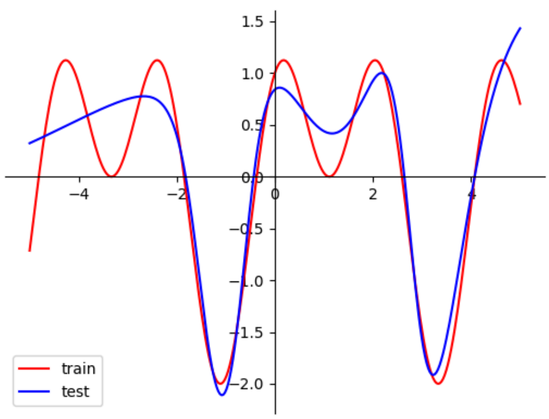
Number of neurons	MSE	Maximum error	Graph
1	0,871	1,934	
2	0,585	2,167	

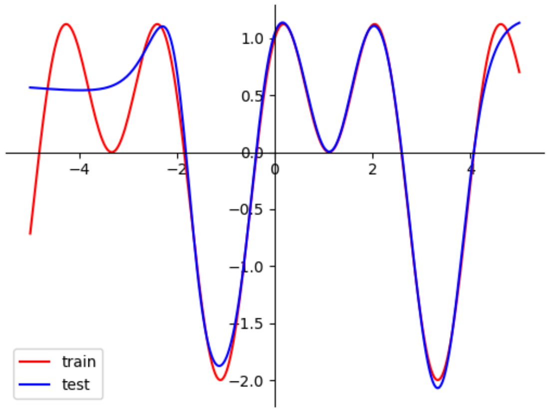
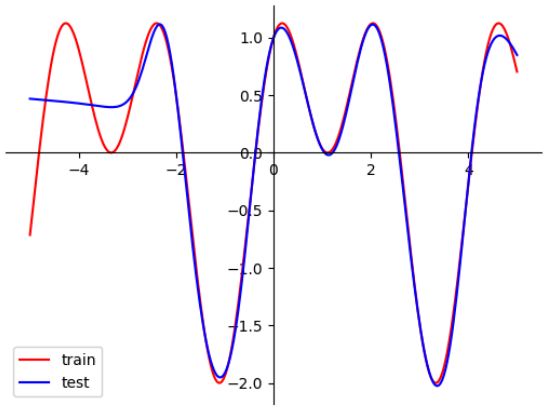
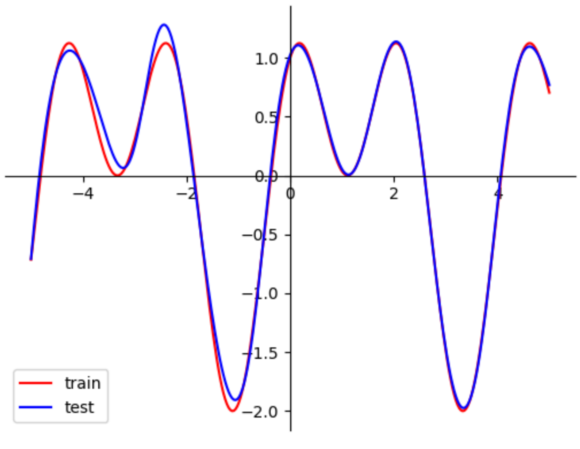
4	0,121	1,391	
8	0,073	1,312	
13	0,055	1,072	

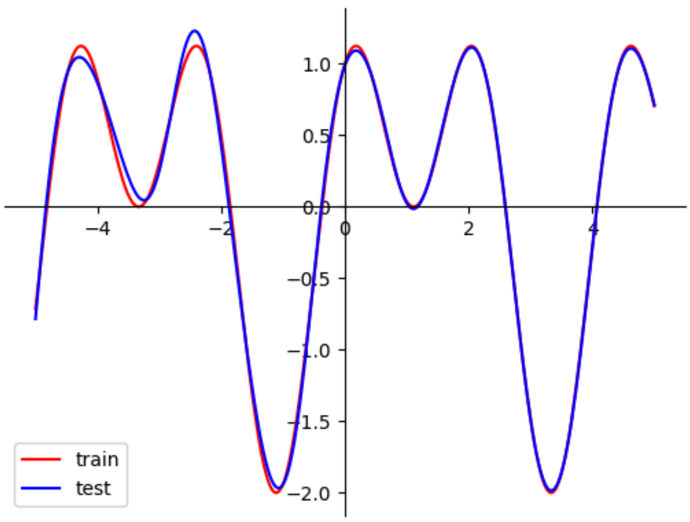
30	0,032	1,029	
50	0,015	0,652	
80	0,058	0,653	

Conclusions:
The number of neurons in the hidden layer has a significant impact on the approximation results. Initially, for a small number of neurons, adding a small number of new neurons significantly improves the quality of the approximation. But later, adding more neurons does not necessarily mean improved approximation, as for 50 neurons we obtained better results than for 80 neurons. Also, one should be careful when choosing the number of neurons in the hidden layer so that network overfitting does not occur with too many neurons.

The effect of the number of iterations on the approximation results
For each measurement, the parameters are:
number of neurons: 10
learning rate: 0.1
mini-batch size: 10

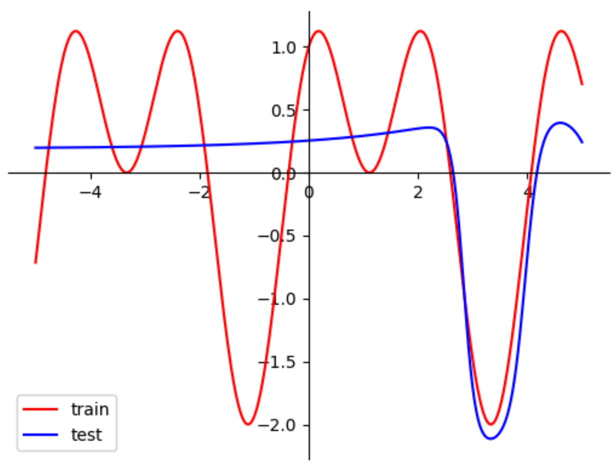
Number of iterations	MSE	Maximum error	Graph
500	0,207	1,076	
1000	0,093	1,034	

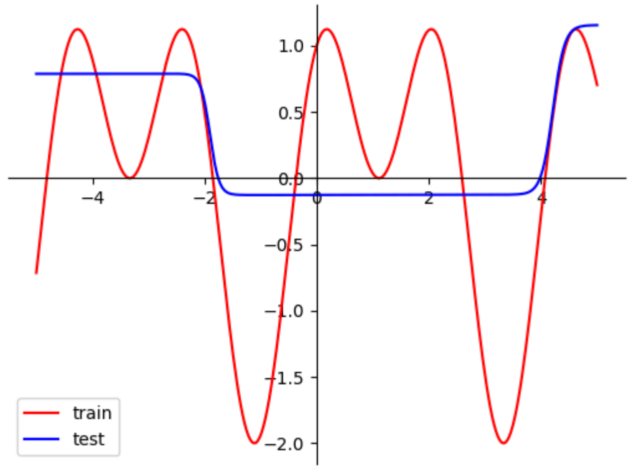
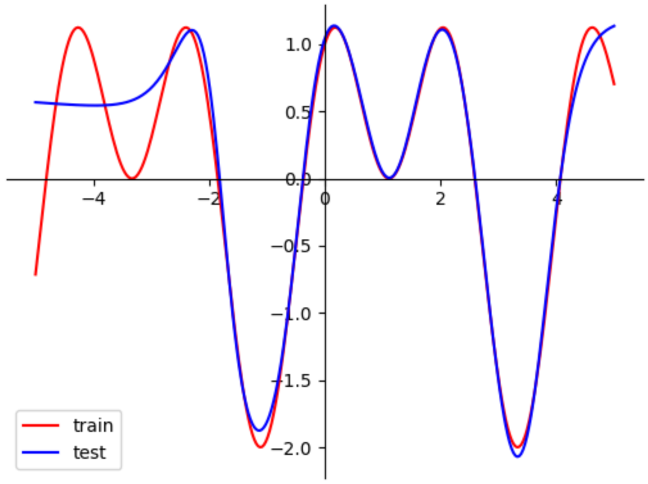
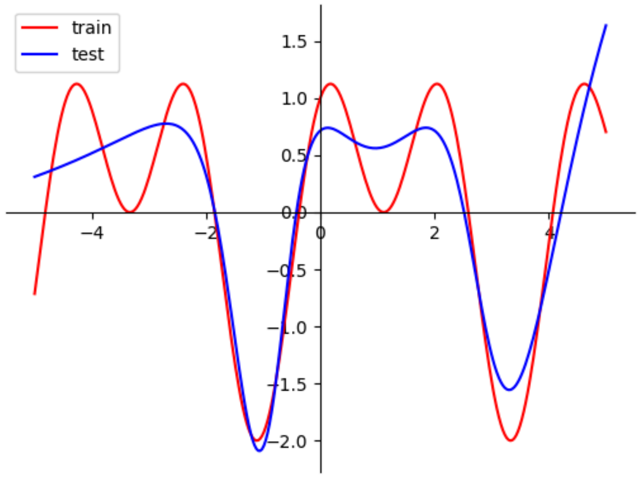
5000	0,054	1,214	
10000	0,047	1,138	
20000	0,0045	0,197	

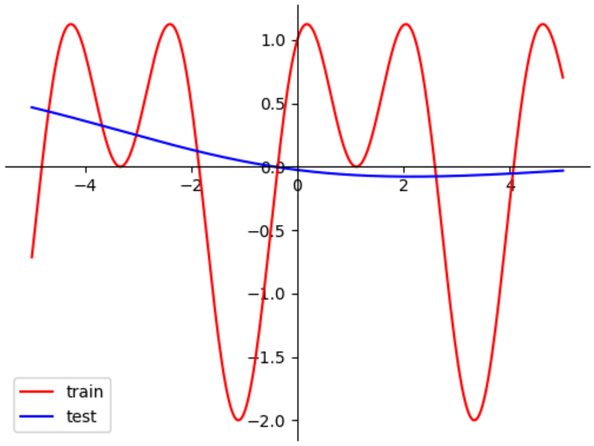
40000	0,0043	0,148	

Conclusions:
The higher the number of iterations, the better results we get. However, beyond a certain point, the results do not improve significantly or remain similar and stay at a certain level.

Effect of learning rate on function approximation
For each measurement, the parameters are:
number of neurons: 10
number of iterations: 5000
mini-batch size: 10

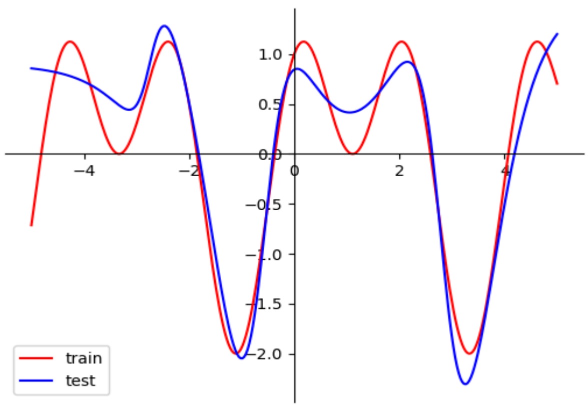
Learning rate	MSE	Maximum error	Graph
0,75	0,593	2,228	

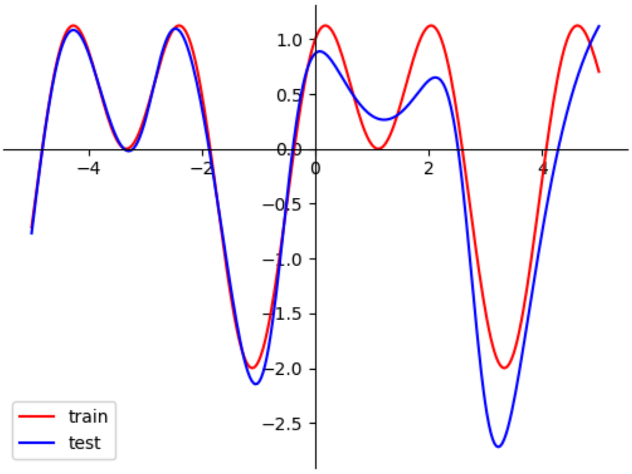
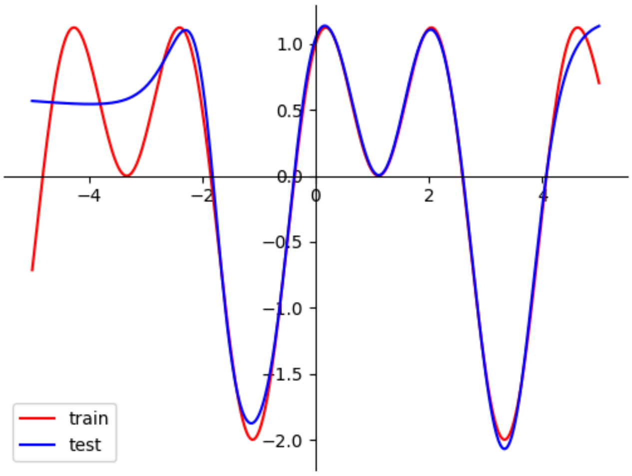
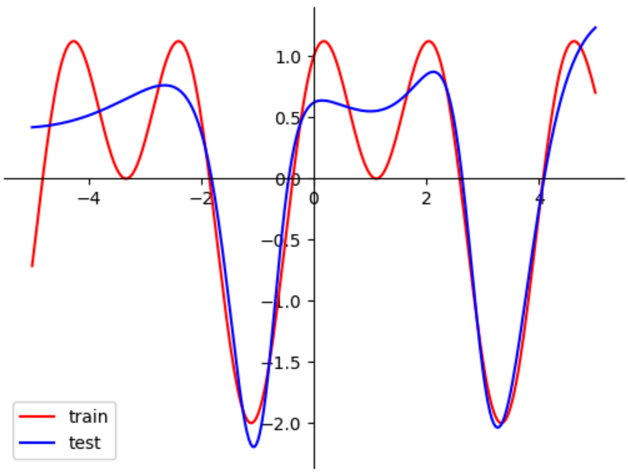
0,5	0,733	1,872	
0,1	0,054	1,214	
0,01	0,135	1,146	

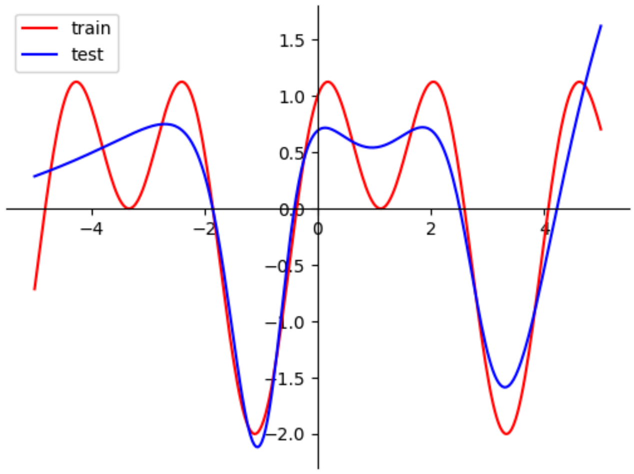
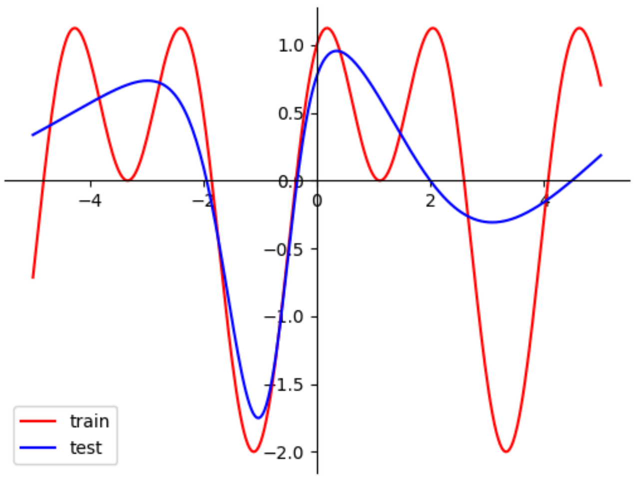
0,001	0,943	2,053	

Conclusions:
Too high learning rate causes the network to want to find the optimal parameters for minimizing the loss function too quickly. As a result, the optimizer may “jump” over the minimum or even destabilize the learning process, resulting in a lack of convergence or oscillations in the loss function values. On the other hand, a learning rate that is too small results in steps to find appropriate weights, and finding a good approximation would require a much larger number of iterations. Moreover, there is a risk that the model will get stuck in the local minimum without reaching the global minimum.

The effect of mini-set size on function approximation
For each measurement, the parameters are:
number of neurons: 10
number of iterations: 5000
learning rate: 0.1

Mini-batch size	MSE	Maximum error	Graph
2	0,095	1,567	

5	0,093	0,865	
10	0,054	1,214	
50	0,105	1,131	

100	0,136	0,993	
200	0,445	1,702	

Conclusions:
The choice of mini-batch size when training a neural network should depend on the values of other parameters. For example, a smaller mini-batch will require more iterations to appropriately approximate the function. For that, a large mini-batch can lead to overfitting and thus average function approximation, as can be seen from the results obtained.