Exercise 4

Philipp Hanslovsky, Robert Walecki

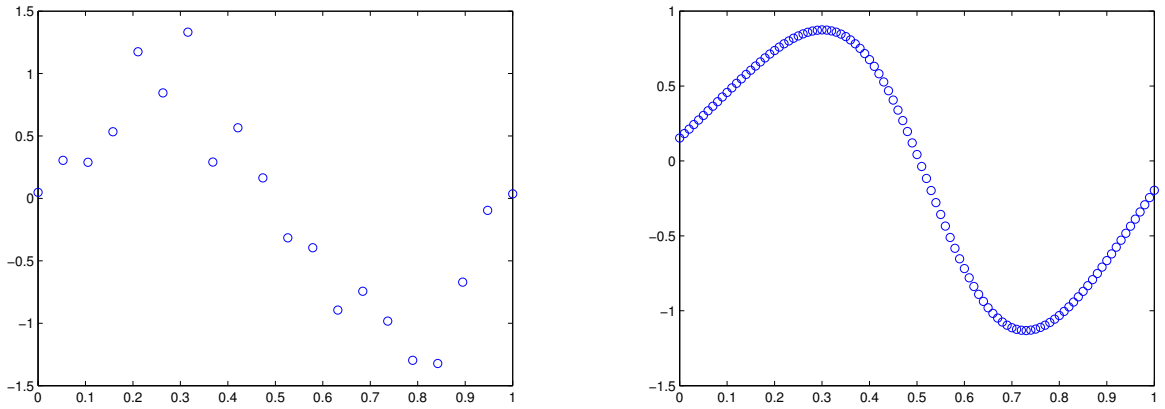
May 19, 2012

4.1.1

The example trains a multi layer perceptron (MLP) using the input array x containg 20 values ranging from 0 to 1 and the target array t, which holds one value for each corresponding value of x. x and t form a sine function distorted by random values. The MLP has one input, three hidden units (perceptrons) and produces a single output. It is created by the command net = mlp(nin, nhidden, nout, outfunction). netopt optimizes the MLP based on input x and target t. Finally a prediction is made (mlpfwd) for an input array plotvals containing 101 values ranging from 0 to 1. The resulting array y is then shown in a plot (see fig 1). Even with the highly distorted sine used for training and using only 3 hidden units the MLP applied to plotvals produces a function close to a sine. The result of increasing the number of hidden units is shown in fig 2 for 50 hidden units and in fig 3 for 100 hidden units.

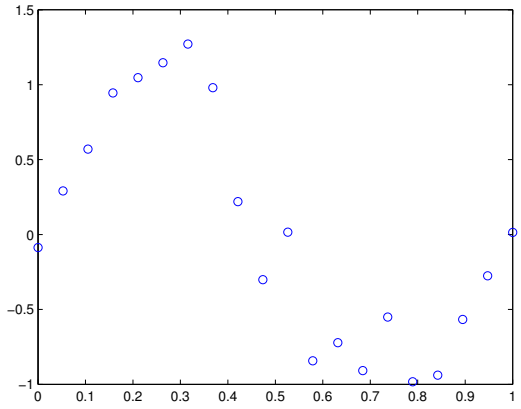
4.1.2

Increasing the weight of the noise results in

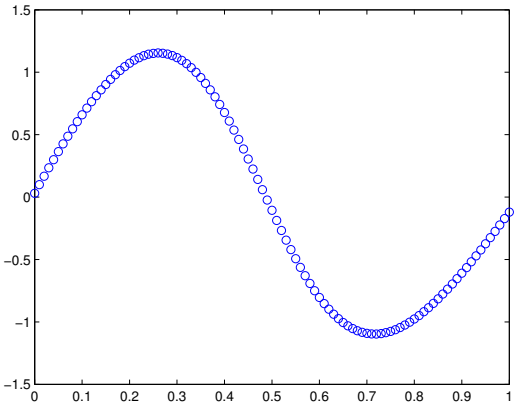


(a) target array t used for MLP training **(b)** output of applying MLP on input array plotvals

Figure 1: Plots of Training Dataset and output (3 hidden units)

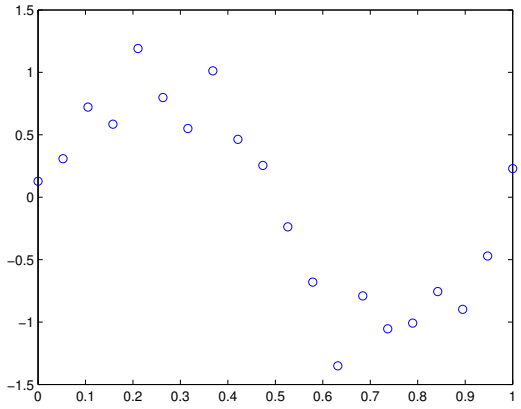


(a) target array t used for MLP training

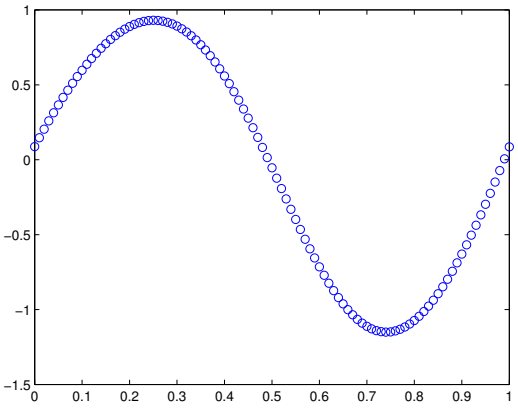


(b) output of applying MLP on input array plotvals

Figure 2: Plots of Training Dataset and output (50 hidden units)



(a) target array t used for MLP training



(b) output of applying MLP on input array plotvals

Figure 3: Plots of Training Dataset and output (100 hidden units)