Seminar 5: Radial Basis Function Networks

In this seminar, you will work on questions related to RBF networks.

Question 1

Download the two sets of 5-dimensional data available on Study Direct named (`sem5_q1_points.mat`) and (`sem5_q1_centres.mat`). Calculate the distances (Euclidean distance) of the 100 points from each of the 10 centres given. Matlab users should find the function `pdist2` most interesting...
Python users: Look for function cdist in scipy.spatial.distance.

Question 2

This question follows the RBF interpolation example from Bishop (1995). Begin by plotting the function $y = 0.5 + 0.4\sin(2\pi x)$ between $x = 0$ and $x = 1$. Now generate 30 random data points between 0 and 1 and add Gaussian noise with standard deviation 0.05 to the corresponding y output. Plot the resulting points on the same graph and proceed as follows:

- Using the Gaussian RBF given in the lecture and *exact interpolation* 'train' an RBF network and plot the resulting fitted curve (in this example you need to keep σ small to avoid numeric problems).
- Now using the identity RBF and (still) exact interpolation, train a new network and fit the resulting curve. What is the difference between using the different basis functions?
- Making use of Matlab's `kmeans` algorithm generate different centres and train a network with both the Gaussian and identity RBF (the pseudoinverse can be calculated in MATLAB using `pinv`). Experiment with different numbers of centres and think about the value you use for σ.
- Finally try different RBFs (i.e. multiquadratic, inverse quadratic) and compare the resulting output.

Python users: A number of options are available. For *kmeans* you may consider Scipy or Scikit-learn. For the pseudo-inverse, it is pinv from numpy.linalg.

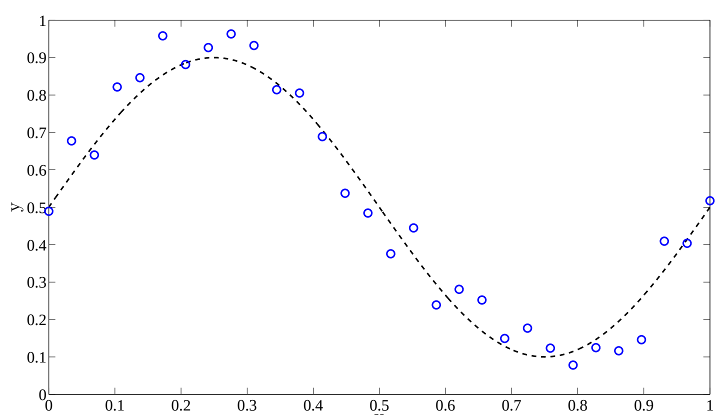

Figure 1: Example of the function and data points