Project # 1 - MLP and Generalized RBF Network

In this project we will implement neural networks for regression. We want to reconstruct in the region [-2; 2] x [- 2; 2] a two dimensional function F : R^2 -> R. We do not have the analytic expression of the function but only a data set obtained by randomly sampling 200 points so that the data set is

$$\{(x^p, y^p) : x^p \in \mathbb{R}^2,\ y^p \in \mathbb{R},\ p = 1, \dots, 200\}.$$

We must split randomly the data set into a training set and a test set. Choose the percentage of training data equal to 75% . Let P = 150 be the number of instances in the training set.

1. (Full minimization)

We must construct a shallow Feedforward Neural Network (FNN) (one only hidden layer), either a MLP or a RBF network, that provides the model f (x). We denote by p the hyper-parameters of the network to be settled by means of an heuristic procedure and w the parameters to be settled by minimizing the regularized training error

$$E(\omega;\pi) = \frac{1}{2P} \sum_{p=1}^{P} \|f(x^p) - y^p\|^2 + \rho \|\omega\|^2$$

where the hyper parameter ρ stays in the range [10^{-5}: 10^{-3}].

1_1. Construct a shallow MLP with a linear output unit, namely

$$f(x) = \sum_{j=1}^{N} v_j g\left(\sum_{i=1}^{n} w_{ji} x_i - b_j\right)$$

where ω = (v; w; b). The activation function g(t) is the hyperbolic tangent

$$g(t) := \tanh(t/2) = \frac{1 - e^{-\sigma t}}{1 + e^{-\sigma t}}$$

The hyper-parameters are

- the number of neurons N of the hidden
- layer the spread σ in the activation function g(t)
- the regularization parameter ρ

1_2. Construct a RBF network

$$f(x) = \sum_{j=1}^{N} v_j \phi(\|x^i - c_j\|)$$

$$\omega = (v, c) \; v \in \mathbb{R}^N \text{ and } c_j \in \mathbb{R}^2 \; j = 1, \ldots, N.$$

We choose as RBF function φ(t) the Gaussian function

$$\phi(\|x - c_j\|) = e^{-(\|x - c_j\|/\sigma)^2} \quad \sigma > 0$$

The hyper-parameters are

- the number of neurons N of the hidden layer
- the spread $\sigma > 0$ in the RBF function φ
- the regularization parameter ρ

2. (Two blocks methods)

2_1.

Consider again the shallow MLP with linear output unit as defined at 1_1 of Question 1. Use the values of N; ρ; σ we fixed at 1_1.

Write a program which implements an Extreme Learning procedure, namely that fix randomly the values of w_{ij}; b_j for all i; j and use the ”best” Python routine or any other optimization library to minimize the quadratic convex training error E(v) of the only variables $v^2 \in R^N$.

2_2.

Consider again an RBF network as in 1_2 of Question 1.

The number of RBF units N of the hidden layer and the positive parameter s in the RBF function are those chosen at 1_2 Question 1.

Write a program which implements a method with unsupervised selection of the centers. We can either select the centers randomly on the P points of the training set or used a Clustering procedure as implemented in Phyton.

Choose the weights by minimizing the regularized error E(v) using a suitable Python routine or any other optimization library to minimize the quadratic convex function.

3. (Decomposition method)

Consider either the shallow MLP or the shallow RBF network.

Write a program which implements a two block decomposition method which alternates the convex minimization with respect to the output weights v and the non convex minimization with respect to the other parameters (either (w; b) for the MLP or the centers c for the RBF network respectively).

Set the regularization parameter r, number of neurons N and spread s at the values we selected in Question 1.