Optmization Methods for Machine Learning - Fall 2016

Assignment # 2 - MLP and Generalized RBF Network

Laura Palagi

Dip. di Ingegneria informatica automatica e gestionale A. Ruberti, Sapienza Università di Roma

Posted on October 28, 2016 - due date November 21, 2016

Instructions

Homework will be done individually: each student must hand in their own answers. It is acceptable for students to collaborate in figuring out answers and helping each other solve the problems. We will be assuming that, as participants in a graduate course, you will be taking the responsibility to make sure you personally understand the solution to any work arising from such collaboration.

Homework must be send by an email both to the teaching assistant Ing. Umberto Dellepiane (umberto.dellepiane@act-OperationsResearch.com) and to laura.palagi@uniroma1.it with subject **[OMML-2016] Project 2**. After you submit, you will receive an acknowledgement email that your project has been received. If you have not received an acknowledgement email within 2 days after you submit then contact the instructor.

The mail must contain as attachment a .zip or .tar.gz file both a typed report in English and the source code (including instructions to run it).

Evaluation criteria You have two questions Q1 and Q2 both with two points (Q1.1; Q1.2; Q2.1, Q2.2). Answering only the first question allows to get up to 25 (up to 23 for Q1.1 and up to 25 for both Q1.1 and Q1.2). Answering questions 1 and 2 allows to get up to 31 (30 cum lade) (up to 28 for Q1.1 and up to 31 for both Q1.1 and Q1.2)
The second homework accounts for 25% of the total vote of the exam.
Homework is due at latest at midnight on the due date. For late homework, the score will be decreased. It is worth 85% for the next 48 hours. It is worth 70% from 48 to 120 hours after the due date. It is worth 50% credit after 120 hours delay.
For the evaluation of the first homework the following criteria will be used:

1. check of the implementation (60% Umberto Dellepiane)
2. Quality of the explanation document and of the overall job (40% - Laura Palagi).

The grade are Italian style namely in the range [0,30], being 18 the minimum degree to pass the exam.

In this assignment you will implement neural networks for regression. We want to reconstruct in the region $[0,1]\times[0,1]$ the Franke's function (see http://www.sfu.ca/ ssurjano/franke2d.html):

$$\begin{aligned} f(x) \; &= 0,75\exp\left(-\frac{(9x_1-2)^2}{4}-\frac{(9x_2-2)^2}{4}\right)+0,75\exp\left(-\frac{(9x_1+1)^2}{49}-\frac{(9x_2+1)}{10}\right) \\ &+0,75\exp\left(-\frac{(9x_1-7)^2}{4}-\frac{(9x_2-3)^2}{4}\right)-0,2\exp\left(-(9x_1-4)^2-(9x_2-7)^2\right) \end{aligned}$$

The data set is obtained by sampling on 100 random points x^i the function and adding a uniform noise, i.e. $y^i = f(x^i)+\varepsilon^i$ and ε^i is a random number in $[0,1]$. You can use the rand function in matlab and initialize the generator using your student number (matricola) as a seed. The data set obtained is

$$\{(x^i,y^i):\; x^i\in\mathbb{R}^2,\; y^i\in\mathbb{R},\; i=1,\dots,100\}.$$

You divide the data set for the learning problem obtained by sampling into a training set and a test set (choose percentage of training data between 70-80%). Let $P<100$ be the number of instances in the training set, as you fixed them.

Question 1.

1. (max score up to 23)

   Consider a two layer MLP of the type

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

   where $g(\cdot)$ is an activation function. As activation function you can choose one of the two functions below, each depending on a parameter $c>0$

   - either the logistic function

   $$g_{log}(t)=\frac{1}{1+e^{-ct}},\quad c>0 \tag{1}$$

   - or the *hyperbolic tangent*

   $$g_{tanh}(t):=\tanh(t/2)=\frac{1-e^{-ct}}{1+e^{-ct}},\quad c>0 \tag{2}$$

   You need to have the number of neurons N of the hidden layer and the positive parameter c in the activation function as parameters that can be changed as input.

   Write a program which implements the error function

   $$E(v,w,b)=\frac{1}{2}\sum_{i=1}^{P}\|y(x^i)-y^i\|^2$$

and use a matlab routine of the optimization toolbox for determine v_j, w_{ji}, b_j which minimize it.

Please note that gradient is not required to be evaluated but it may be useful to use it.

Analyse the occurrence of overfitting/underfitting varying the number of neurons N and c. Produce a plot of the approximating function found.

In the report you must state:

- which activation function do you choose;

- the final setting for the number of neurons N and for the parameter c; how do you choose them and if you can put in evidence over/under fitting, when and why, if you have an explanation for this behaviour;

- which optimization routine do you use for solving the minimization problem and the setting of its parameters, if any;

- the value of the error on the training and test set;

- the plot of the function representing the approximating function obtained by the MLP in comparison with the true one.

2. (max score up to 25) Consider an RBF network of the type

$$y(w,c;x) = \sum_{j=1}^{N} w_j \phi(\|x^i - c_j\|)$$

You can choose as RBF function $\phi(\cdot)$ one of the two functions below, each depending on a parameter σ

- the Gaussian function

$$\phi(\|x - c_j\|) = e^{-(\|x-c_j\|/\sigma)^2} \quad \sigma > 0 \tag{3}$$

- the *Inverse Multiquadric*

$$\phi(\|x - c_j\|) = (\|x - c_j\|^2 + \sigma^2)^{-1/2}, \quad \sigma > 0 \tag{4}$$

You need to have the number of RBF units N of the hidden layer and the positive parameter σ in the RBF function as parameters that can be changed as input.

Write a program which implements the error function of the RBF network

$$E(w) = \frac{1}{2}\sum_{i=1}^{P}\left(\sum_{j=1}^{N} w_j \phi(\|x^i - c_j\|) - y^i\right)^2 + \frac{\rho_1}{2}\|w\|^2 + \frac{\rho_2}{2}\|c\|^2,$$

where $\rho_1, \rho_2 > 0$ are regularization parameters to be defined by the user. Use a matlab routine of the optimization toolbox for its minimization with respect to both (w,c).

Please note that gradient is not required to be evaluated but it may be useful to use it.

Analyse the occurrence of overfitting/underfitting varying the number of units N and of the spread parameters σ. Produce a plot of the approximating function found.

In the report you must state:

- which RBF function do you choose;

- the values of the parameters ρ (chose in $[10^{-5} \div 10^{-3}]$)

- the final setting for the number of neurons N and of the parameter σ; how do you choose them and if you can put in evidence over/under fitting, when and why, if you have an explanation for this behaviour;

- which optimization routine of the Optimization toolbox do you use and the setting of its parameters, if any;

- the value of the error on the training and test set;

- the plot of the function representing the approximating function obtained by the RBF network in comparison with the true one.

- a comparison of performance between MLP and RBF netowrk both in terms of quality of approximation (training and test error) and in efficiency of the optimization (number of function/gradient evaluation and computational time needed to get the solution). Please put these values in a table.

Question 2.

Consider again a RBF network of the type

$$y(w,c;x) = \sum_{j=1}^{N} w_j \phi(\|x^i - c_j\|)$$

You can choose as RBF function $\phi(\cdot)$ either the Gaussian function (3) or the Inverse Multiquadric (4) (it can be the same as in Question 1).

The number of RBF units N of the hidden layer and the positive parameter σ in the RBF function are input parameters.

1. (max score up to 28)

   Write a program which implements a method with unsupervised selection of the centers. Select the centers randomly on the P points of the training set.

   Choose the weights by minimizing the convex quadratic function

$$E(w) = \frac{1}{2}\sum_{i=1}^{P}\left(\sum_{j=1}^{N} w_j \phi(\|x^i - c_j\|) - y^i\right)^2 + \frac{\rho_1}{2}\|w\|^2,$$

   using a suitable routine of the optimization toolbox.

In the report you must state:

- which RBF function do you choose;

- the values of the parameters ρ (it can be the same used in Question 1)

- the number N of the units in the hidden layer and the parameter σ in the RBF network (you can choose the values you got in Question 1)

- which optimization routines of the Optimization toolbox do you use to solve the weights subproblems and the setting of its parameters, if any. Choose at least a gradient based method;

- the value of the error on the training and test set;

- the plot of the function representing the approximating function obtained by the RBF with unsupervised selection of the centers in comparison with the true one.

2. (max score up to 31) Write a program which implements a supervised selection of both weights and centers using a two block decomposition method which alternates the minimization with respect to weights and centers. Set the regularization parameter $\rho_1, \rho_2 > 0$ at the value you defined in Question 1. You can use appropriate matlab routines of the optimization toolbox for solving the block minimization problems both with respect to centers and with respect to weights(choose at least a gradient based method if possible) or you can also implement an approximate gradient method for the center minimization problem. Compare the results with those obtained by using a routine at the second point of Question 1.

   In the report you must state:

   - which RBF function do you choose;

   - the values of the parameters ρ (it can be the same used in the preceding questions)

   - which optimization routines of the Optimization toolbox do you use to solve the two block subproblems and the setting of its parameters, if any.

   - Comparison with the network obtained at second point of Q1 and at the first point of Q1; comparison are both in the quality of the solutions (training and test error), the number of outer iterations (number of subproblems solved), and in efficiency of the optimization (number of function/gradient evaluation and computational time needed to get the solution). Please put these values in a table.

   - the plot of the function representing the approximating function obtained by the RBF with block decomposition in comparison with the true one.