CS671 - Deep Learning and its Applications

Course Instructor : Aditya Nigam (Session : Feb-May 2017)

Assignment-01

Submission Date : February 28, 2017

Note:

1. Answer all questions. Maximum score is 10 points.
2. Make sure you clearly identify the question number and your final answer, and solve all parts of a question together. Show all your working and clearly mark all relevant axes and intercepts in plots.
3. You are free to make any reasonable assumption that you may need to logically answer a question.
4. Code has to be well commented and as general as possible.
5. You are expected to submit a make file to compile your code and a README file containing how to run your codes.

---

1. **Function Approximation using Ridge Regression**
   (a) Assume any function $f(x)$, where x may be univariate or multivariate.
   (b) Generate N randomly perturbed samples, sampled using assumed function and perturbed using some fixed ϵ.
   (c) Divide them into a training set D and a testing set T, assuming some r, dividing ratio.
   (d) Assume model complexity in terms of M (some kind of polynomial function).
   (e) Now use the training set D, to approximate the underlying function by optimizing the error function assuming a polynomial curve.
   (f) Generate the graphs (using gnuplot) as discussed in the class to justify your solution, given below.
      i. Plot original function along with perturbed training data and the original errors, as shown in Fig. 1.

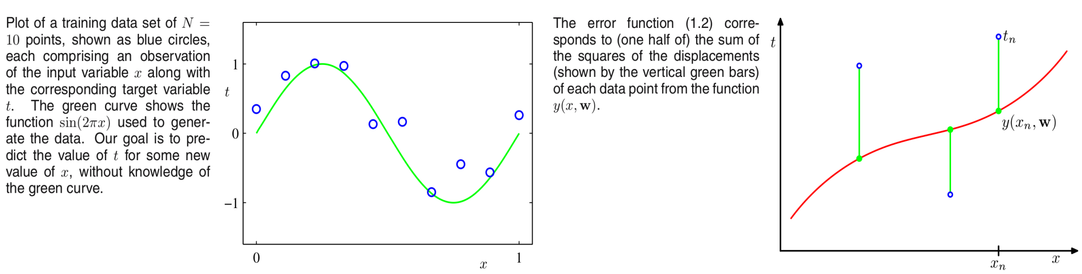

Figure 1: Original Function and Error

      ii. Plot the approximated function by varying M but fixed N, as shown in Fig. 2. Compute the weight vectors for each settings.
      iii. Plot approximated function by varying N and fixed M, as shown in Fig. 3. Compute the weight vectors for each settings.
      iv. Finally apply the linear model of regression with quadratic regularize (Ridge Regression). Assume different values of λ and justify your performance over training and test set in terms of committed error, as shown in Fig. 4.

   (10 points)

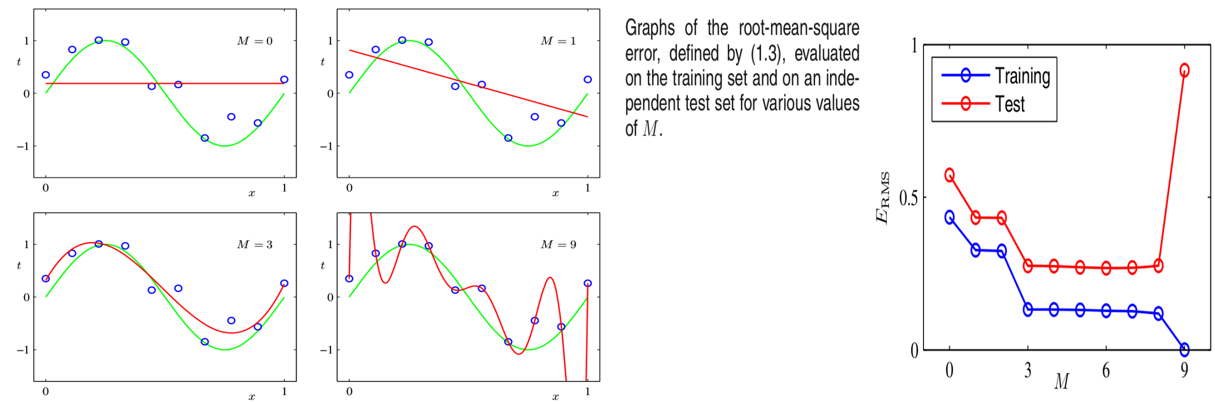

Figure 2: Approximated function by varying M but fixed N, along with Testing and Training error

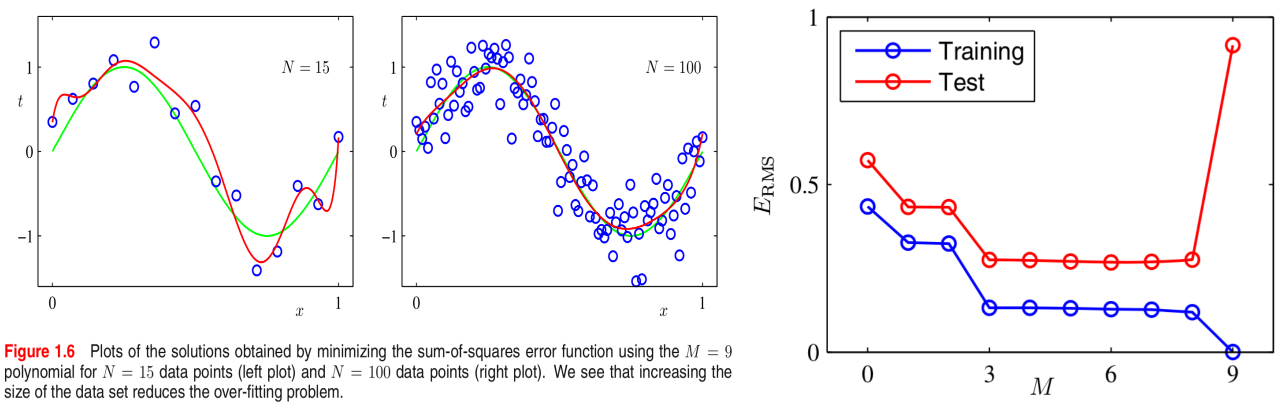

Figure 3: Approximated function by varying N but fixed M, along with Testing and Training error

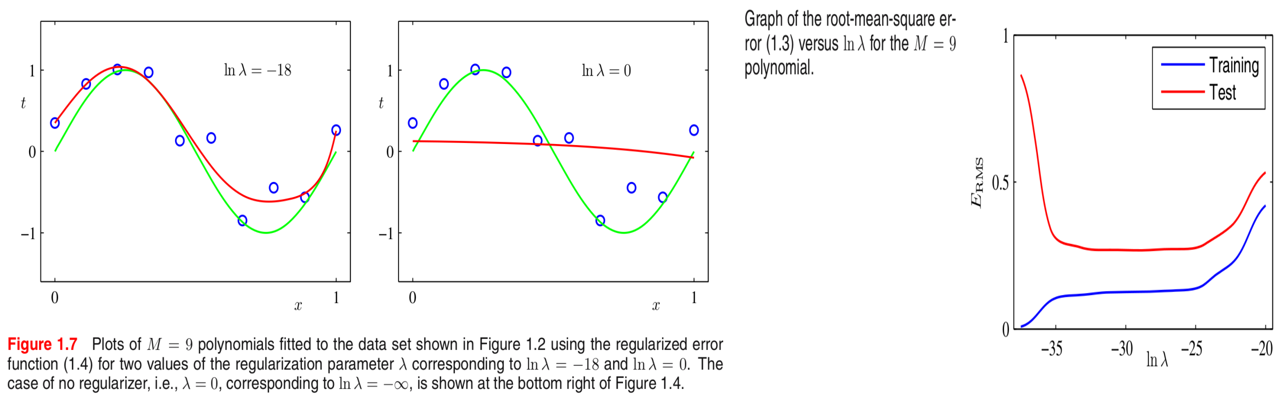

Figure 4: Same experimentation but this time with quadratic regularize, along with Testing and Training error.