Assignment 1 - Trade-off between Overfitting and Underfitting

Due Date: September 29, 11:30 pm

Late submission

If you submit the assignment after the deadline, the following penalty is applied:

- ◇ 10% penalty if the submission is before October 6, 11:30 pm (if the mark before applying the penalty is 78 out of 100, after applying the penalty it is 78 - 7.8 = 70.2 out of 100);
- ◇ 50% penalty if the submission is after October 6, 11:30 pm and before Dec. 6, 11:30 pm.

Description:

In this assignment, you are required to perform an experiment similar to the experiment in Section 1.1 of PRML [1]. **Therefore, you have to read that experiment and its discussion first.** You will have to compare several regression models to illustrate the trade-off between overfitting and underfitting. You will use data generated synthetically. The prediction is to be performed based on only one feature, denoted by x. The target t is a noisy measurement of the function of $f_{true}(x)$. Thus t satisfies the following relation

$$t = \underbrace{sin(4\pi x + \pi/2)}_{f_{true}(x)} + \epsilon, \tag{1}$$

where ϵ is random noise with a Gaussian distribution with 0 mean and variance 0.0625.

Construct a training set consisting of only 12 examples $\{(x^{(1)}, t^{(1)}), \cdots, (x^{(10)}, t^{(12)})\}$, where $x^{(1)}, \cdots, x^{(12)}$ are uniformly spaced in the interval $[0, 1]$, with $x^{(1)} = 0$, $x^{(12)} = 1$, and $t^{(1)}, \cdots, t^{(12)}$ are generated using relation (1), the noise being randomly generated. Construct a validation set consisting of 120 examples with features uniformly spaced in the interval $[0, 1]$, with $x^{(1)} = 0$, $x^{(120)} = 1$ and targets generated randomly according to relation (1). **When generating the random data use a four-digit number containing the last 4 digits of your student ID (in any order), as seed for the pseudo number generator.**

You have to train twelve least squares regression models of increasing capacity (corresponding to M from 0 to 11) and record and compare their training and validation errors. For model M, $0 \leq M \leq 11$, the prediction function has the form

$$f_M(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M.$$

Note that, when $M = 0$, the prediction function is just a constant function. For each M you have to train the model using least squares and record the training and validation errors. For each M, plot the prediction $f_M(x)$ function and the curve $f_{true}(x)$ versus $x \in [0, 1]$. Additionally, include in the figure all the points in the training set and in

the validation set with their true target values. Use different colours for the training examples, the validation examples, the prediction function and the true function, i.e., four colours. In addition, plot the training and validation errors versus M. Also include in this plot the average squared error between the targets and the true function $f_{true}(x)$ for the examples in the validation set (this will be a horizontal line). **What does this value represent?**

Additionally, for $M = 11$ you have to train the model with regularization in order to control overfitting. Here you have to try several values for λ until you find a value λ_1 that eliminates the overfitting. You also have to find a value λ_2 for which underfitting occurs. Then plot the training and validation errors versu λ, for all the λ values that you tried. Also include in this plot the average squared error between the targets and the true function $f_{true}(x)$ for the examples in the validation set (this will be a horizontal line). For each of λ_1 and λ_2 plot the prediction function $f_M(x)$ and the curve $f_{true}(x)$ against x, and all the points in the training set and in the validation set with their true targets.

You have to write a report to present your results and their discussion. Discuss what you observe in the plots. Justify your choice for λ_1 and λ_2. Report the parameter vectors for λ_1 and λ_2 and compare them. Do they fullfill your expectations? The report should be clear and concise.

Besides the report, you have to submit your numpy code. The code has to be modular and should use vectorization instead of for loops whenever possible. Write a function for each of the main tasks. Also, write a function for each task that is executed multiple times (e.g, to compute the average error). The code should include instructive comments.

Code for feature standardization:

When training a linear model with regularization, the features have to be standardized first. You may use the code given below for feature standardization.

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
XX_train = sc.fit_transform(XX_train)
XX_valid = sc.transform(XX_valid)
```

Note that the standardization is performed based on the training data. Then the same transformations are applied to the validation data in order to be able to use the trained predictor on the validation data. In other words, the value that is subtracted from each feature and the value used for scaling in the validation data are the same ones that were used for the training data.

Submission Instructions:

- Submit the files in the Assignments Box on Avenue. Specific instructions for the format of the submitted files is provided on Avenue

References

[1] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006 (ISBN 9780387310732), available for free download at `https://www.microsoft.com/en-us/research/people/cmbishop/prml-book/`.