Machine Learning
Università della Svizzera italiana

# Assignment 1

Maggioni Claudio

May 4, 2021

The assignment is split into two parts: you are asked to solve a regression problem, and answer some questions. You can use all the books, material, and help you need. Bear in mind that the questions you are asked are similar to those you may find in the final exam, and are related to very important and fundamental machine learning concepts. As such, sooner or later you will need to learn them to pass the course. We will give you some feedback afterwards.
!! Note that this file is just meant as a template for the report, in which we reported **part of** the assignment text for convenience. You must always refer to the text in the README.md file as the assignment requirements.

## Regression problem

This section should contain a detailed description of how you solved the assignment, including all required statistical analyses of the models' performance and a comparison between the linear regression and the model of your choice. Limit the assignment to 2500 words (formulas, tables, figures, etc., do not count as words) and do not include any code in the report.

### Task 1

Use the family of models $f(\mathbf{x}, \boldsymbol{\theta}) = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \theta_3 \cdot x_1 \cdot x_2 + \theta_4 \cdot \sin(x_1)$ to fit the data. Write in the report the formula of the model substituting parameters $\theta_0, \ldots, \theta_4$ with the estimates you've found:

$$f(\mathbf{x}, \boldsymbol{\theta}) = \_ + \_ \cdot x_1 + \_ \cdot x_2 + \_ \cdot x_1 \cdot x_2 + \_ \cdot \sin(x_1)$$

Evaluate the test performance of your model using the mean squared error as performance measure.

### Task 2

Consider any family of non-linear models of your choice to address the above regression problem. Evaluate the test performance of your model using the mean squared error as performance measure. Compare your model with the linear regression of Task 1. Which one is **statistically** better?

### Task 3 (Bonus)

In the **Github repository of the course**, you will find a trained Scikit-learn model that we built using the same dataset you are given. This baseline model is able to achieve a MSE of **0.0194**, when evaluated on the test set. You will get extra points if the test performance of your model is better (i.e., the MSE is lower) than ours. Of course, you also have to tell us why you think that your model is better.

## Questions

### Q1. Training versus Validation

1. **Explain the curves' behavior in each of the three highlighted sections of the figures, namely (a), (b), and (c).**

   In the highlighted section (a) the expected test error, the observed validation error and the observed training error are significantly high and close toghether. All the errors decrease as the model complexity increases. In (c), instead, we see a low training error but high validation and expected test error. The last two increase as the model complexity increases while the training error is in a plateau. Finally, in (b), we see the test and validation error curves reaching their respectively lowest points while the training error curve decreases as the model complexity increases, albeit in a less steep fashion as its behaviour in (a).

2. **Is any of the three section associated with the concepts of overfitting and underfitting? If yes, explain it.**

   Section (a) is associated with underfitting and section (c) is associated with overfitting.

   The behaviour in (a) is fairly easy to explain: since the model complexity is insufficient to capture the behaviour of the training data, the model is unable to provide accurate predictions and thus all MSEs we observe are rather high. It's worth to point out that the training error curve is quite close to the validation and the test error: this happens since the model is both unable to learn accurately the training data and unable to formulate accurate predictions on the validation and test data.

   In (c) instead, the model complexity is higher than the intrinsic complexity of the data to model, and thus this extra complexity will learn the intrinsic noise of the data. This is of course not desirable, and the dire consequences of this phenomena can be seen in the significant difference between the observed MSE on training data and MSEs for validation and test data. Since the model learns the noise of the training data, the model will accurately predict noise fluctuations on the training data, but since this noise is completely meaningless information for fitting new datapoints, the model is unable to accurately predict for validation and test datapoints and thus the MSEs for those sets are high.

   Finally in (b) we observe fairly appropriate fitting. Since the model complexity is at least on the same order of magnitude of the intrinsic complexity of the data the model is able to learn to accurately predict new data without learning noise. Thus, both the validation and the test MSE curves reach their lowest point in this region of the graph.

3. **Is there any evidence of high approximation risk? Why? If yes, in which of the below subfigures?**

   Depending on the scale and magnitude of the x axis, there could be significant approximation risk. This can be observed in subfigure (b), namely by observing the difference in complexity between the model with lowest validation error and the optimal model (the model with lowest expected test error). The distance between the two lines indicated that the currently chosen family of models (i.e. the currently chosen gray box model function, and not the value of its hyperparameters) is not completely adequate to model the process that generated the data to fit. High approximation risk would cause even a correctly fitted model to have high test error, since the inherent structure behind the chosen family of models would be unable to capture the true behaviour of the data.

4. **Do you think that by further increasing the model complexity you will be able to bring the training error to zero?**

Yes, I think so. The model complexity could be increased up to the point where the model would be so complex that it could actually remember all x-y pairs of the training data, thus turning the model function effectively in a one-to-one direct mapping between input and output data of the training set. Then, the loss on the training dataset would be exactly 0. This of course would mean that an absurdly high amount of noise would be learned as well, thus making the model completely useless for prediction of new datapoints.

5. **Do you think that by further increasing the model complexity you will be able to bring the structural risk to zero?**

   No, I don't think so. In order to achieve zero structural risk we would need to have an infinite training dataset covering the entire input parameter domain. Increasing the model's complexity would actually make the structural risk increase due to overfitting.

## Q2. Linear Regression

Comment and compare how the (a.) training error, (b.) test error and (c.) coefficients would change in the following cases:

1. $x_3$ **is a normally distributed independent random variable** $x_3 \sim \mathcal{N}(1, 2)$
2. $x_3 = 2.5 \cdot x_1 + x_2$
3. $x_3 = x_1 \cdot x_2$

## Q3. Classification

1. **Your boss asked you to solve the problem using a perceptron and now he's upset because you are getting poor results. How would you justify the poor performance of your perceptron classifier to your boss?**
2. **Would you expect to have better luck with a neural network with activation function** $h(x) = -x \cdot e^{-2}$ **for the hidden units?**
3. **What are the main differences and similarities between the perceptron and the logistic regression neuron?**