ft_linear_regression

An introduction to machine learning

Summary: In this project, you will implement your first machine learning algorithm.

Version: 4

Contents

Chapter I

Foreword

What i think is the best definition for machine learning :

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Tom M. Mitchell

Chapter II

Introduction

Machine learning is a growing field of computer science that may seem a bit complicated and reserved only to mathematicians. You may have heard of neural networks or k-means clustering and don't undersdand how they work or how to code these kinds of algorithms...

But don't worry, we are actually going to start with a simple, basic machine learning algorithm.

Chapter III

Objective

The aim of this project is to introduce you to the basic concept behind machine learning. For this project, you will have to create a program that predicts the price of a car by using a linear function train with a gradient descent algorithm.

We will work on a precise example for the project, but once you're done you will be able to use the algorithm with any other dataset.

Chapter IV

General instructions

In this project you are free to use whatever language you want.

You are also free to use any libraries you want as long as they do not do all the work for you. For example, the use of python's numpy.polyfit is considered cheating.

You should use a language that allows you to easily visualize your data : it will be very helpful for debugging.

Chapter V

Mandatory part

You will implement a simple linear regression with a single feature - in this case, the mileage of the car.

To do so, you need to create two programs :

- The first program will be used to predict the price of a car for a given mileage. When you launch the program, it should prompt you for a mileage, and then give you back the estimated price for that mileage. The program will use the following hypothesis to predict the price :

$$estimatePrice(mileage) = \theta_0 + (\theta_1 * mileage)$$

  Before the run of the training program, theta0 and theta1 will be set to 0.

- The second program will be used to train your model. It will read your dataset file and perform a linear regression on the data.
  Once the linear regression has completed, you will save the variables theta0 and theta1 for use in the first program.
  You will be using the following formulas :

$$tmp\theta_0 = learningRate * \frac{1}{m} \sum_{i=0}^{m-1} (estimatePrice(mileage[i]) - price[i])$$

$$tmp\theta_1 = learningRate * \frac{1}{m} \sum_{i=0}^{m-1} (estimatePrice(mileage[i]) - price[i]) * mileage[i]$$

  I let you guess what m is :)

  Note that the estimatePrice is the same as in our first program, but here it uses your temporary, lastly computed theta0 and theta1.

  Also, don't forget to simultaneously update theta0 and theta1.

Chapter VI

Bonus part

Here are some bonuses that could be very useful :

- Plotting the data into a graph to see their repartition.
- Plotting the line resulting from your linear regression into the same graph, to see the result of your hard work !
- A program that calculates the precision of your algorithm.

The bonus part will only be assessed if the mandatory part is PERFECT. Perfect means the mandatory part has been integrally done and works without malfunctioning. If you have not passed ALL the mandatory requirements, your bonus part will not be evaluated at all.

Chapter VII

Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your folders and files to ensure they are correct.

Here are the points that your peer-corrector will have to check :

- The absence of libraries that do the work for you
- The use of the specified hypothesis
- The use of the specified training function