ECE241 PROJECT 3: The First Step in Machine Learning
Due: Dec 17, 2021, 11:50pm on Gradescope

Introduction

In this project, you will implement your first machine learning algorithm to estimate the house pricing. You will use Linear Regression and Gradient Decent to build a model and estimate the price of given house.

This project consists of two parts: a written part and a programming part. In the written part, you have to finish the tasks by hand and answer some basic questions. *You need show all the steps necessary to get the solution.* In the programming part, you have to write code to solve the problems and will be asked to show the results and answer some (short) questions related to your results. You may NOT use any machine learning specific libraries in your code, e.g., TensorFlow, PyTorch, or scikit-learn. You may use libraries like numpy, pandas, and matplotlib, though (and we encourage you to use them). Your code should be written in Python3.

Part I (Written)

In this part, we consider the data points with the features of housing price problem discussed in class. The data can be found in "part1.csv". You can download the csv file from Moodle. You can use Excel to get solutions/draw figures to provide the answers to the problems below. In this case, report a screenshot of your Excel worksheet and report the function/formula you used.

We use feature vector X to represent a data point in the dataset, and we are trying to find a set of weights $W = [w_1, w_2, \ldots, w_n]$ and bias term b such that

$$\hat{y} = \sum_{i=1}^{n} x_i \times w_i + b$$

can be used to estimate the house price where n is the number of features.

1. **[8 points]** Suppose an randomly initialized $W = [10, 1, 1, 1, 1]$ and $b = 0$ estimates the housing price in the following way:

   $$\hat{y} = 10 \times x_1 + x_2 + x_3 + x_4 + x_5$$

   where the $\hat{y}$ is the estimated value for house price. Calculate and report the estimated house price for all data points in the csv file.

2. **[6 points]** Evaluate how good the model is by computing the Mean Squared Error (MSE) for the W above.

   $$\text{MSE} = \frac{1}{|X|} \sum_{k} \left(\hat{y}^{(k)} - y^{(k)} \right)^2$$

   where $|X|$ is the number of data points and $X^{(k)}$ is the kth data point in the dataset. $y^{(k)}$ is the true housing price for the kth house.

3. **[6 points]** Explain why we cannot use the "sum over all errors" to evaluate how good a model is. i.e., what would be the problem if we propose "Mean Total Error (MTE)" to evaluate a model like this

$$\text{MTE} = \frac{1}{|X|}\sum_{k}\left(\hat{y}^{(k)} - y^{(k)}\right)$$

4. **[6 points]** In class, we discussed about the gradient decent algorithm such that our model can estimates the house price a little bit better every time. A formal definition of the gradient descent can be written as

$$W_{t+1} = W_t - \alpha\nabla\text{MSE}(W_t)$$

where $\nabla\text{MSE}(W_t)$ represent the gradient of the loss function with respect to weight W at time t. Describe what α is and why it is necessary.

5. **[7 points]** Suppose your model now "perfectly" outputs the house price (for example, with 0 MSE). Is your model essentially a "good" model for this problem? What could go wrong if the model were to deployed for real-world usage (simply reply "does not predict correctly" won't work, you need to explain why)?

6. **[6 points]** In order to fix the problem in the previous question, what method discussed in class can be helpful? Write down the name of the method and describe how you can use it to fix the problem. Argue why this can help you with this problem.

Part II (Programming)

In this part, you will have access to the full housing price dataset and build a regression model to predict the house price. **You need to implement the algorithms from scratch. Using ML-specific libraries will result in 0 points in this part!** The datasets to be used in this part are named "train.csv" and "test.csv". You can download the csv files from Moodle. You can ignore the bias term **b** in the programming part for simplicity.

1. Read the training data from "train.csv".

2. **[7 points]** Before you start, always remember to take a look at the data you are going to deal with. Analyse the training set and report the following metrics:

   - How many records are there in the training set.
   - What is the mean value of the price.
   - What is the minimal and maximal price.
   - What is the standard derivation of the price.

3. **[7 points]** Show a histogram of the sales price.

4. **[7 points]** Some features are correlated with each other. Report a pair-wise scatter plot of the following features and report what you found. Describe what you could do to accelerate the training process without compromising too much accuracy. (You do not have to implement what you proposed here in the following questions.)

   - GrLivArea
   - BedroomAbvGr
   - TotalBsmtSF
   - FullBath

5. Implement function `pred` that calculates the predicted value of the price based on the current weights and feature values.

$$\hat{y} = \sum_{i=1}^{n} w_i \times x_i$$

6. Implement function `loss` that calculates the loss based on a set of predicted sale price and the correct sale price. In this task, you should implement the mean squared error as the loss function.

$$\text{MSE}(\hat{Y}, Y) = \frac{1}{|Y|} \sum_{k=1}^{|Y|} \left(\hat{y}^{(k)} - y^{(k)}\right)^2$$

7. Implement function `gradient` that calculates the gradient of loss function based on the predicted price and the correct price. For simplicity, we provide the gradient, you do not need to derive the expression for gradient in your code.

$$\nabla\text{MSE}(W) = \frac{2}{|Y|} \times X^T(\hat{Y} - Y)$$

8. Implement function `update` that updates weights based on the gradient.

$$W_{t+1} = W_t - \alpha\nabla\text{MSE}(W_t)$$

9. Keep training your weights in your main function.

Algorithm 1: Train you model.

```
Let W to be a randomly initialized weight matrix;
for each iteration do
    Ŷ ← pred(X) ;
    MSE ← loss(Ŷ, Y) ;
    ∇ ← gradient(Ŷ, Y, X) ;
    W ← W − α∇ ;
```

10. **[6 points]** First set α to be 0.2. Does your algorithm finds the minimal MSE? If so, report the number of iterations your algorithm converges. If not, what's happening and explain why that is the case.

11. **[12 points]** Now set $\alpha = 10^{-11}$ and $\alpha = 10^{-12}$. Run your algorithm for 500 iterations under both configurations and report a learning curve where the x-axis is the number of iterations and y-axis is the MSE. Your learning curve should have two lines in one plot and clearly identify the α value. An example learning curve is shown in Figure 1. Note, this is only an demonstration of what your plot should look like, this is not a learning curve of anything!

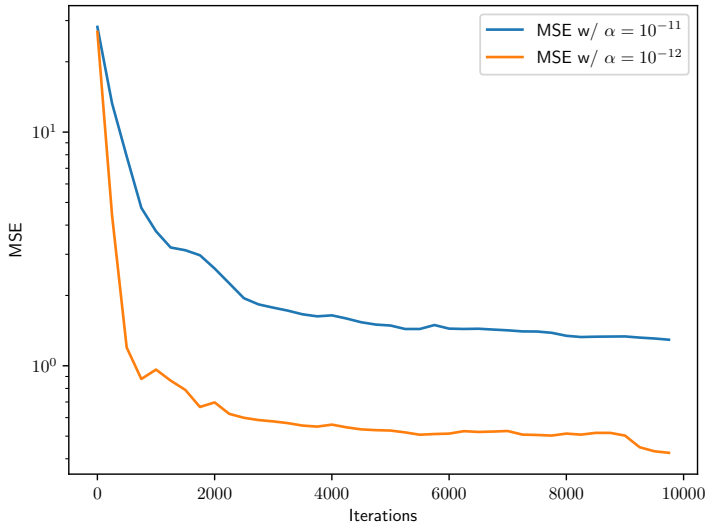

Figure 1: An example learning curve.

12. **[6 points]** For the two α, which one converges faster? Describe why it is this case.

13. **[6 points]** Predict the house price for the test set (file "test.csv"). Report the MSE for your model on the test set. In general, will your model achieves better MSE on the test set than the training set? If not, why?

Submission Instructions

- You should submit a report in PDF format answering all questions in the written part and show the required output in the programming part. You should also submit the code you wrote in the programming part.
- Your PDF report should have response to all tasks that has a blue indicator with corresponding points at the beginning.
- The report and the code should be submitted separately on Gradescope before the submission deadline. There will be no autograder setup for this project. Please ignore the 1 point from the Gradescope, that only indicates whether you submitted the required file. You won't get any points for submitting a file named "project3.py".

- Your code for part II should be in one file named "project3.py", any other file submitted to the Gradescope will be IGNORED! The code should be well documented and allow the grader to reproduce your result. Remember, code structure and readability consists of 10 points for this project! Failure to reproducing the results in your report will result in half of the credits for that task.
- The course honesty policy requires you to write all code yourself. Your submitted code will be compared to the code submitted by your colleagues, and also with many implementations available online. Note that our checking program is not confused by changed variable or method names.