ENGR-UH 4560

Selected Topics in Information and Computational Systems: Machine Learning

Name: Nishant Aswani

Net ID: nsa325

Assignment Title: Mini-Project 1

Climate Change Data Prediction and Modifying Gradient Descent

Nishant Aswani, nsa325@nyu.edu

Selected Topics in Information and Computational Systems: Machine Learning(ENGR-UH 4560), Professor Hwasoo Yeo

1 Introduction

The Gradient Descent Method is an iterative method used to find the minimum of a given function. In this mini project, modifications are made to the standard stochastic gradient descent method to minimize the cost function and obtain a linear model for socioeconomic factors affecting city emissions. The report also discusses the reasons for these modifications and its practical use.

The dataset was obtained from a Nature article titled "A global dataset of CO2 emissions and ancillary data related to emissions for 343 cities" authored by Nangini et. al [1]. The authors gathered emissions and ancillary data for 343 cities from a variety of sources, carried out data quality control, and combined it to form a dataset. The data covers "data from CDP (187 cities, few in developing countries), the Bonn Center for Local Climate Action and Reporting (73 cities, mainly in developing countries), and data collected by Peking University (83 cities in China)" [1].

2 Methodology

2.1 Data Selection

As there was Scope 1 Green House Gas (GHG) Emissions data available for all 343 cities, it was selected to be the target variable. Scope 1 GHGs refer to "transport, industrial, waste and local power plants emissions" [1].

From the larger dataset, a small range of potentially interesting features were initially selected. Using a heatmap, the two most interesting features were selected: Population and Built-up Area.

Data verification for the target variable showed that there was an extreme outlier; for simplicity, this row was dropped out of the dataframe. Moreover, all rows with unavailable, or `NaN` values, were dropped.

2.2 Data Preprocessing

Since ""[learning estimators] might behave badly if the individual features do not more or less look like standard normally distributed data," [2] the data was preprocessed using sklearn's RobustScaler method. The result of standardization can be seen below in kernel density estimation (KDE) plot. Furthermore, the dataset was split into training (80%) and testing (20%).

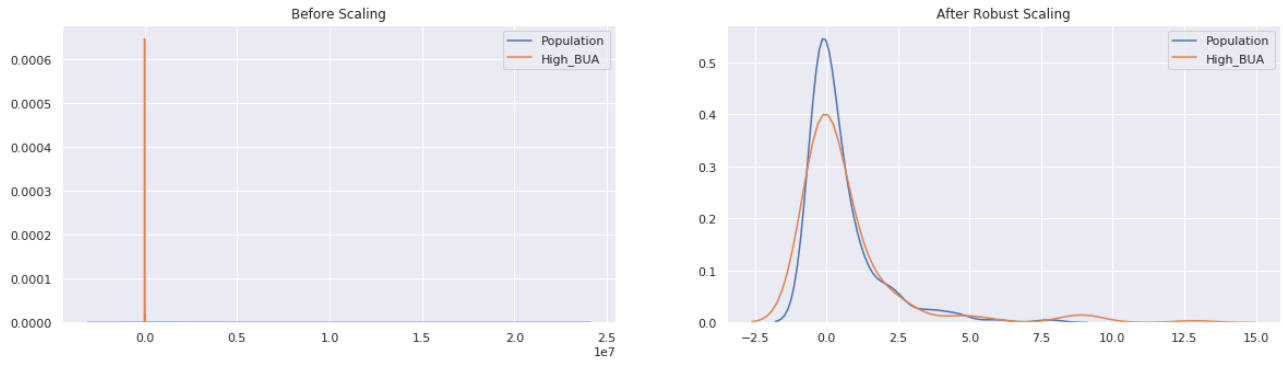

Figure 1: Before and after standardization of the features

2.3 Gradient Descent

As a benchmark to compare gradient descent methods, the built-in sklearn linear regression model was used to train standardized training data set.Next, a model was trained using a standard stochastic gradient descent, with a rational function as the learning schedule.

The modified gradient descent sought to tweak two of the core functionalities: the learning schedule and determination of the gradient.

2.3.1 Learning Schedule

The original learning schedule is a rational function $\frac{t_0}{t+t_1}$, where t is updated at each iteration. The hyperparameters t_0 and t_1 are selected so that at $t = 0$, the learning rate is 0.1. The learning rate then rapidly decreases until it plateaus towards the end.

Initially, it was thought to update the learning schedule's hyperparameters so that learning reflected the depth or shallowness of the cost function parabaloid. However, it was realized that θ does account for the gradient at each iteration. Hence, factoring the gradient into the learning schedule might over or underplay its effect in calculating the next θ value.

Often, gradient descent methods face the challenge of finding themselves stuck at local minima instead of converging at the global minima. Hence, the modified gradient descent attempted to use a linearly descending, dampened sine wave to "shock" the model out of local minima situations. A dampened-sin wave function was derived for a given epochs of 50.

It is clear that the goal of the original learning schedule is to aid in converging as rapidly as possible and then tweak the model slowly. On the other hand, the dampened sin wave initially provides larger shocks to the system and then gradually converts into a negative linear slope.

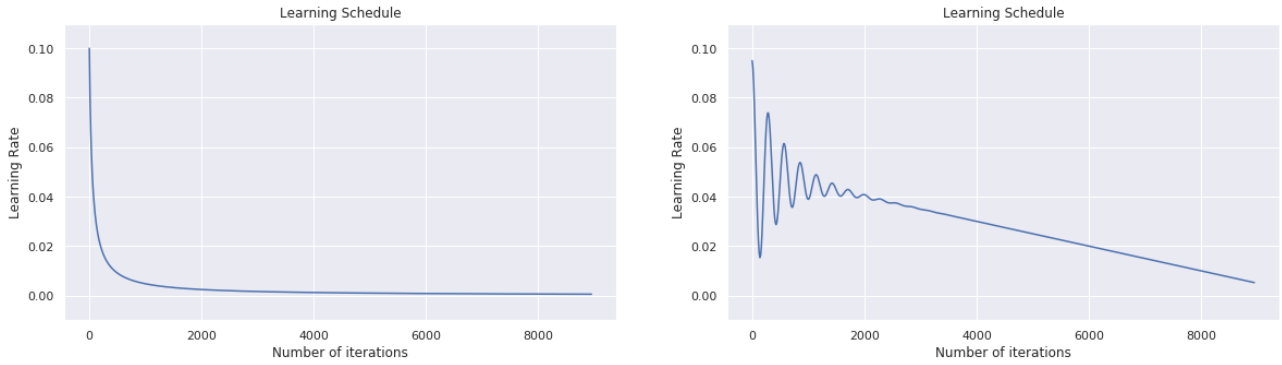

Figure 2: The original learning schedule compared with the modified learning schedule

A variation of the dampened sin wave was initially considered, pictured at the bottom-right. However, the model was discarded as it provided too large of a variation even towards the end of the iterations.

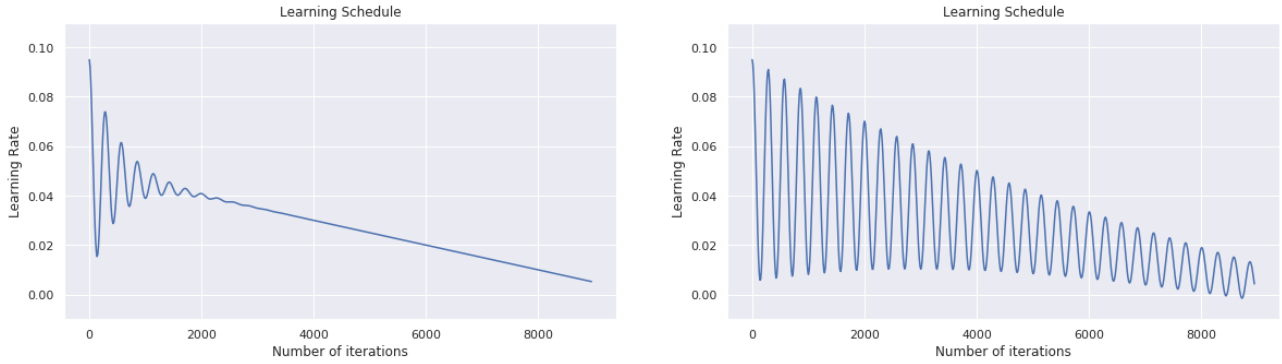

Figure 3: The original learning schedule compared with the modified learning schedule

However, in the case of linear regression there is no concern for being stuck at local minima, as there is only one minimum to which the model must converge to.

2.3.2 Gradient Averaging

Another modification made was the method of determining the gradient. It was realized that gradient calculation was reliant upon the random selection of a data point from the training data. To modify gradient calculation, the training space was divided into three sections. In each iteration, three random data points were selected, one from each of the split sections.

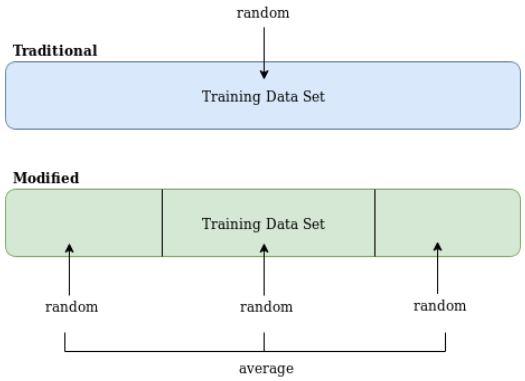

Figure 4: Gradient averaging, where the pipe represents finding the gradient of the random datapoint

Since sklearn's train test splitting method randomizes the data after splitting, there was no fear of data bias. The gradient of each of the three data points was calculated and averaged to obtain a single gradient value for each feature. As before, this was used to update the theta. In reality, splitting the data into three section has little effect, aside from increasing the probability of each data point being selected. However, selecting an average gradient of several random points, instead of just one, prevents an unreasonably large gradient from affecting the step size.

The reason for why averaging the gradient works lies in understanding that the derivative of a sum of functions is equal to the sum of a derivative functions.

3 Results

For this project, the built-in `LinearRegression` method was used as a benchmark to compare the unmodified and modified gradient descent (GD) approaches. The table below shows the results of 100 runs of each method.

Method	Pop. Coeff.	Pop. Coeff. Variance	BUA Coeff.	BUA. Coeff. Variance	Time(s)
sklearn LinReg	0.776708	N/A	0.294042	N/A	N/A
Traditional SGD	0.779071	0.00314603	0.29797795	0.00159789	8.15
Modified SGD	0.77412671	0.00112532	0.2980754	0.00071944	39.2

Table 1: Execution statistics comparing the two GD methods

From the modified method, we are given the model, where we can use the intercept determined by sklearn to replace the bias term:

$$y = 0.015627 + 0.77412671x_1 + 0.2980754x_2 \tag{1}$$

4 Discussions and Conclusion

As shown by the results, the traditional method has a higher variance compared to the modified method. We can thus infer that the modified method is more likely to converge at the minimum. However, at 100 iterations, the modified method is approximately five times slower. This is because the modified method runs an extra for-loop, decreasing convergence time.

Moreover, the learning rate used in the modified method would be more applicable for problems beyond linear regression, where being trapped within a local minima would pose as a real risk. Another drawback of the learning rate is that it would have to be re-crafted if the number of epochs were to change because the dampened sin wave is a sensitive function.

Overall, the modified method trades-off convergence time for likelihood of convergence, which could be useful depending on the resources at hand and endgoal.

References

[1] Cathy Nangini et al. "A global dataset of CO 2 emissions and ancillary data related to emissions for 343 cities". In: *Scientific data* 6 (2019), p. 180280.

[2] *Preprocessing data*. `https://scikit-learn.org/stable/modules/preprocessing.html`.