Lab Course Machine Learning
Exercise Sheet 3

Prof. Dr. Dr. Lars Schmidt-Thieme, Shayan Jawed
Information Systems and Machine Learning Lab
University of Hildesheim

November 12th, 2020
Submission on November 18th, 2020 at 12 noon, (on learnweb, course code 3116)

Instructions

Please following these instructions for solving and submitting the exercise sheet.

1. You should submit a jupyter notebook detailing your solution.
2. Please explain your approach i.e. how you solved a given problem and present your results in form of graphs and tables.
3. Please submit your jupyter notebook to learnweb before the deadline. Please refrain from emailing the solutions except in case of emergencies.
4. Unless explicitly noted, you are not allowed to use scikit, sklearn or any other library for solving any part.
5. Please refrain from plagiarism.

Exercise 1: Gradient Descent on Rosenbrock function (5 Points)

In this part, you are required to optimize the *Rosenbrock function*. This function serves to benchmark all optimization algorithms alike. The function can be stated mathematically as follows:

$$f(x, y) = (a - x)^2 + b(y - x^2)^2 \tag{1}$$

The function is known to be challenging for optimization. The global minimum is however known to be at:

$$(a, a^2)$$

where the function value is:

$$f(x, y) = 0$$

For the purpose of this exercise, let $a = 1$ and $b = 100$

1. Implement a 3D plot to visualize the function (Use Matplotlib's 3D utilities)
2. Derive the partial gradients. (Please look into how to typeset latex in Jupyter notebooks)
3. Convert the function and gradient of this function into equivalent code representation.
4. Optimize the function with Gradient Descent. Set the appropriate hyperparameters like initial valus of (x,y) and the steplength α through trial and error.
5. Visualize the trajectory on the same 3D plot. This trajectory should ideally lead to the function minimum, starting off with $(x = 10, y = 10)$ for example.

Exercise 2: Linear Regression with Gradient Descent

Part A: (Datasets) (3 Points)

Airfare and demand: target $->$ price
Wine Quality: target $->$ quality
Parkisons Dataset: target $->$ total_UPDRS

You are required to pre-process the datasets by following these steps:

1. Convert any non-numeric values to numeric values. For example you can replace a country name with an integer value or more appropriately use hot-one encoding. [Hint: use pandas.get_dummies]. Please explain your solution.

2. If required drop out the rows with missing values or NA. In next lectures we will handle sparse data, which will allow us to use records with missing values.

3. Split the dataset into 80% Train set and 20% Test set.

4. Are there any columns that can be dropped? if so, which ones are why.

Part B: Linear Regression with Real-World Data (5 Points)

In this part you are required to implement linear regression algorithm with gradient descent algorithm. Reference lecture `https://www.ismll.uni-hildesheim.de/lehre/ml-20w/script/ml-02-A1-linear-regression.pdf`
For each dataset given above:

- 1. A set of training data D_{train} = $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(N)}, y^{(N)})\}$, where $x \in R^M, y \in R$, where N is number of training examples and M is number of features

- Linear Regression model is given as $\hat{y}^n = \sum_{m=1}^{M} \beta_m x_m^n$

- Least square loss function is given as $l(x, y) = \sum_{n=1}^{N} (y^n - \hat{y}^n)^2$

- Minimize the loss function $l(x, y)$ using Gradient Descent algorithm. Implement (learn-linregGD and minimize-GD algorithms given in the lecture slides). Choose i_{max} between 100 to 1000. Explain your choice [hint: the following plots might be useful in your choice.]

- You can choose three suitable values of step length $\alpha > 0$. For each value of step length perform the learning and record
  - In each iteration of the minimize-GD algorithm calculate $|f(x_{i-1}) - f(x_i)|$ and (when i_{max} is reached), plot it against iteration number i. Explain the graph.
  - In each iteration step also calculate RMSE on test set $RMSE = \sqrt{\frac{\sum_{q=1}^{T} (y_{test}^q - \hat{y}^q)^2}{T}}$ and plot it against iteration number i. Explain the graph.

Exercise 3: Steplength Control for Gradient Descent (2+2+3)

This task is based on the Gradient Descent algorithm above. You have to implement the following steplength controlling algorithms:

1. *steplength-backtracking* as given in lecture slides

2. *steplength-bolddriver* as given in lecture slides

3. *Look-ahead optimizer* please refer to publication here: `https://arxiv.org/pdf/1907.08610.pdf`

For each step length algorithm and for each dataset in Exercise 2:

- In each iteration of the minimize-GD algorithm calculate $|f(x_{i-1}) - f(x_i)|$and plot it against iteration number i. Explain the graph.
- In each iteration step also calculate RMSE on test set $RMSE = \sqrt{\frac{\sum_{q=1}^{T}(y^q_{test} - \hat{y}^q)^2}{T}}$, plot it against iteration number i. Explain the graph.

Declare a winning step-length controller based on the metric RMSE. You should tune the associated hyperparameters of the step-length controller(s).