CS 591 Programming Project 3

SGD Optimizer & Neural Network

Grade: 100 marks in total (Program: 70%, Team Review: 30%)

Type: Group work (Team size cannot exceed 6)

Programming language & Framework: Python, TensorFlow or any other Neural Network frameworks.

Hints: Review Stochastic Gradient Descent feature weights update video. Review Jupyter Notebook files (.ipynb) files on Canvas to build your own Neural Networks with TensorFlow framework

Project Overview: The goal of this assignment is to know how to update the feature weights by using Stochastic Gradient Descent optimizer and solve practical problems by using self-designed Neural Networks.

Tasks:

Part 1: Stochastic Gradient Descent

Part1_x_y_Values.txt file contains 100 pairs of (X, Y) points. You will consider the problem of using a curve to fit those data points. Figure 1 shows the 100 data points.

You can use **multiple features** to draw the curve and fit the 100 data points.

For example, if you use 2 features, your model should be like:

$Y = a * X + b$ (1)

The 2 features are **a** and **b**.

If you use 3 features, your model should be like:

$Y = a * X^2 + b * X + c$ (2)

The 3 features are **a**, **b** and **c**.

You can use as many features as you want by following the rule above.

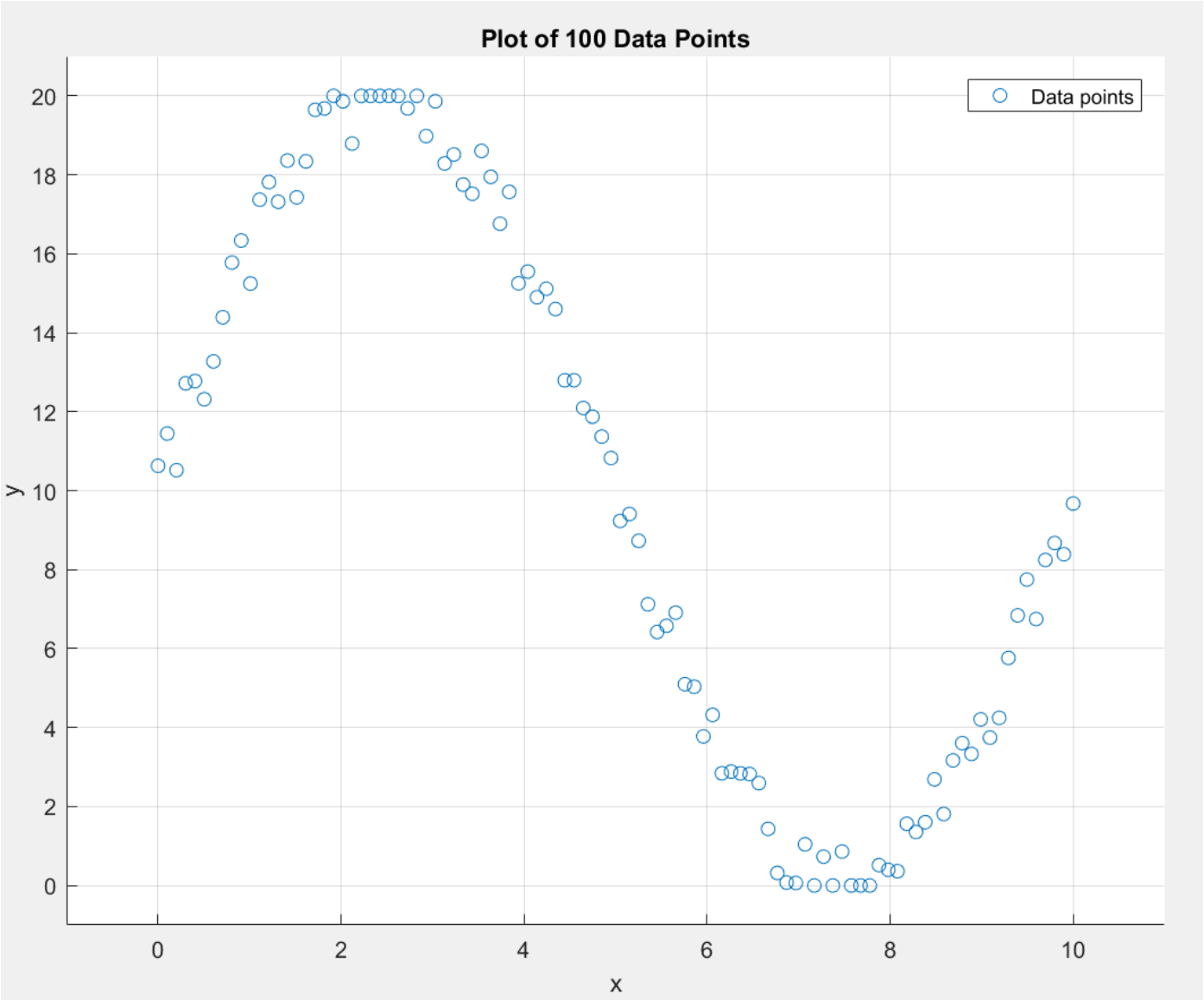

Figure 1

Your report should include:

- Implement Gradient Descent (GD) optimizer to update the feature weights, you need to use 3 features at least. Use learning rate = 0.01, show your updated fitting curve for each epoch and report the history of the feature weights update for every epoch.
- Implement Stochastic Gradient (SGD) optimizer to update the feature weights, you need to use 3 features at least. Use learning rate = 0.01, show your updated fitting curve for each epoch and report the history of the feature weights update for every epoch. (**SGD or mini-batch SGD**)
- For **both** Gradient Descent and Stochastic Gradient Descent, you can get **extra 5** points if you do experiments on **two** different number of features that is larger than 3.

- For **both** Gradient Descent and Stochastic Gradient Descent, you can get **extra 5** points if you do experiments on **two** other learning rate values.
- For **both** Gradient Descent and Stochastic Gradient Descent, you can get **extra 10** points if you can show your updated fitting curve (e.g.,3 features fitting curve is $a * X^2 + b * X + c$) for each epoch.

Part 2: Neural Network

Choose a problem/topic of interest and try to solve it using deep learning/ machine learning methods similar or related to the ones described in class. The problem/topic of interest should represent a scientific challenge or an application. Ideally, the problem/topic of the project will align with the research interests of the Team.

Elements that need to be in the report:

The report should contain the following information:

- Detail the motivation and significance of the problem being approached.
- Include relevant references (well cited, high impact factor journals and conferences, etc.).
- Describe and discuss the final methodology employed by the team to solve the problem.
- Present and discuss the results achieved by the team.
- Discuss strengths and limitations of the proposed methodology or application.

Each of the components of the final project (report & presentation, source code, peer review) will be given a score between 0 and 100 and will be weighted accordingly to obtain the final grade.

Submission Instructions

You will need to turn in the following:

1. A **report** in **PDF format**. The report should briefly describe your implemented solution and fully answer the questions posed above. **Remember: you will not get credit for any solutions you have obtained, but not included in the report. Your report should (ideally) make it possible for me to understand your solution without having to run your source code.**

   Project report template (for reference only): a project report may include at least the following items: **abstract**, **problem description**, **algorithm (pseudocode)**, **implementation details**, **running results and analysis**, **conclusions on what you have learned in the project**, and **references** (if applicable, also explicitly cite them in your project report).

2. At the **beginning** of the report, the report must contain all the team members' name, ID, graduate or undergraduate (e.g., Juefei Yuan, S0000000, graduate)

3. Your **source code**. The code should be well commented, and it should be easy to see the correspondence between what's in the code and what's in the report. You don't need to include executables or various supporting files (e.g., utility libraries) whose content is irrelevant to the assignment. If I find it necessary to run your code in order to evaluate your solution, I will get in touch with you.

Please **zip** both the report and source code into one package and name it based on the following rule:

"**Team leader_First_Name_ Team leader _Last_Name_P3.zip**"

(i.e., Juefei_Yuan_P3.zip).
Submission venue: Submit your files through Canvas.

Late policy: You lose 25% of the points for every day the assignment is late. If you have a compelling reason for not being able to submit the assignment on time and would like to make a special arrangement, you must let me know **at least a week before the due date** (any genuine emergency situations will be handled on an individual basis).

Academic integrity: Feel free to discuss the assignment with each other in general terms, and to search the Web for general guidance (not for complete solutions). Coding should be done individually. If you make substantial use of some code snippets or information from outside sources, be sure to acknowledge the sources in your report.