Deep Learning

Homework and Programming Assignment 1

**Total Points: 100**

**Deadline: Feb 1, 2023**

1. [**Points 15**] We have given 5 students aptitude test marks and their statistically computed grades below.

|  |  |  |
| --- | --- | --- |
| Student | Marks | Grade |
| 1 | 95 | 85 |
| 2 | 85 | 95 |
| 3 | 80 | 70 |
| 4 | 70 | 65 |
| 5 | 60 | 70 |

1. [**Points 10**] We are interested to compute grades from given test marks using linear regression. Can you please estimate the line equation using the normal equation method? Please show the detailed computation of each step. Also, show the final line equation and its parameters.
2. [**Points 5**] Predict grades for the following given student marks using your computed equation in step 1.

Given marks [65, 75, 77, 83, 87]

1. [**Points 50**] You are asked to predict weather temperature. You only know the linear regression model and luckily the given samples can be mapped using it. In your given dataset temperature value depends on both humidity and visibility. To execute the tasks, you are given skeleton codes. Write your own code by modifying, updating, inserting code as necessary to estimate a linear equation for the given datasets. Write your code using raw python code. You can use NumPy, pandas, etc. However, you are not allowed to use any high-level API (such as TensorFlow, PyTorch, Maxnet, etc.). As a reference, you can see/follow d2l.ai logistic regression implementations.

* 1. [**Points 5**] get\_data () function returns the data and split it into training and test set. Write data\_iter() function to create batch-wise data and return batches as needed during your training.
  2. [**Points 5**] You are fitting these data samples using a linear equation. Write a function create\_model\_parameter(mu, sigma, row, column) to create the parameters and initialize values with normal random values. mu and sigma represent mean and standard deviation, respectively.
  3. [**Points 5**] Write your code for the linear regression given as model() function in the skeleton code.
  4. [**Points 5**] Compute loss function using squared\_loss() function.
  5. [**Points 5**] Compute gradient using gradient() function for each parameter of your model.
  6. [**Points 7**] Update you model parameter using sgd() function
  7. [**Points 8**] Write your train() function to execute your linear regression for all the samples given.
  8. [**Points 10**] Draw a single figure for training loss vs number epochs for three different batch sizes. Write your own function by modifying draw\_loss(). Please choose batch sizes as small, large, and just exact. Explain the effect of batch sizes on the training loss.

1. [**Points 10**] How the logistic regression algorithm works - explain answers with respect to the followings - formulating a hypothesis, hypothesis evaluation, and loss minimization and updating parameters. Your answer should also contain different scale transformations (e.g., probabilistic scales, log odds scales.) explanations.
2. [**Points 25**] Implement logistic regression from scratch without using any high-level API (e.g., TensorFlow, PyTorch, Caffee, etc.) You may use NumPy, Pandas. Use a publicly available dataset to showcase how it works. Please plot training loss vs the number of epochs. Your code should be modular (each individual function should perform one task) and show prediction accuracy for test examples. You will be graded as zero without executable code and visible output.

**Submission Instructions:**

**Important.**

**Python notebook:** If you are using python, I would recommend using jupyter notebook/lab. So that I can see the visible output of your source code. the cell outputs should be visible with a sample test example if applicable. In case of you are using python scripts instead of notebook, then create a demo video presentation of your source execution showing the intermediate states and submit this demo with your source.

**C/C++ or other programming language user:** Who uses c/c++, should create a demo video presentation of your source execution showing the intermediate states and submit this demo with your source.

Please note that rephrasing or renaming variables and function names do not mean you are not copying/cheating. For this similar situation, you will be graded zero.

**You should not zip your submission. – you may submit two files – one for question answering and another for coding.**

**In both cases:** the source should be executable without error. Without visible outputs, you may get zero for those corresponding questions. You can submit your text question answer in a separate PDF/Doc (if needed) or in the same pdf/doc.

**Late submission or Extension:** Late HomeWorks/assignment will not be accepted unless an extension is approved by me in advance. Requests for extensions must be made at least three days before the due date with valid reason. **2 points** will be deducted for each day after the submission deadline from your grade even if you are approved for extension. For details, please see the **Homework and Exam Policies** section of your syllabus for more details.

**Grading Policy/Rule:** Copying/cheating/plagiarism is strictly prohibited as mentioned in our introductory lectures and syllabus. This policy holds for each assignment/homework/exam. In case of copying/cheating/plagiarism etc. you will be graded zero for the assignment as well as ‘F’ for the subject. Note that the first incident of cheating will result in the student getting a final grade of ‘F’ for the course. The second incident, by CCSE rules, will result in a semester suspension from the College.