

UNIVERSITY OF PENNSYLVANIA
ESE 5460: PRINCIPLES OF DEEP LEARNING
HOMEWORK 0 (NOT GRADED)

Instructions. Read the following instructions carefully before beginning to work on the homework. Please note that this homework **will not be graded**. However, you are still expected to submit the homework to familiarize yourself with the submission process.

- You will submit solutions typeset in L^AT_EX on Gradescope (strongly encouraged). You can use hw_template.tex on Canvas in the “Homeworks” folder to do so. If your handwriting is unambiguously legible, you can submit PDF scans/tablet-created PDFs.
- Clearly indicate the name and Penn email ID of all your collaborators on your submitted solutions.
- Start a new problem on a fresh page and mark all the pages corresponding to each problem. Failure to do so may result in your work not being graded completely.
- For each problem in the homework, you should mention the total amount of time you spent on it. This helps us keep track of which problems most students are finding difficult.
- You can be informal while typesetting the solutions, e.g., if you want to draw a picture feel free to draw it on paper clearly, click a picture and include it in your solution. Do not spend undue time on typesetting solutions.
- This assignment will have two Gradescope submission assignments. **You must submit to both.** The files to submit for each section are:
 - HW0 PDF
 - * The PDF file with all of your written answers. Please note that you will be required to page match your answer to each question. Failure to do so may result in your work not graded completely.
 - HW0 Problem 4 Code:
 - * pennID_hw0_problem4.py
- Note, we will not accept .ipynb files (i.e., Jupyter notebooks), you should only upload .py files. If you are using Google Colab to do your homework (and I suggest that you don’t...), you can export the notebook to a .py file.
- **This is very important.** While this assignment is not graded, future assignments will be. In these assignments, your code files will be **autograded** or run by the instructors, so your code should be such that it can be executed

independently without any errors to create all output/plots required in the problem.

Credit. The points for the problems may add up to more than 100. If so, you only need to solve for 100 points to get full credit, i.e., your final score will be $\min(\text{your total points}, 100)$.

Problem 1 (15 points). For the function

$$f(x) = 2x_1^2 - 1.05x_1^4 + \frac{1}{6}x_1^6 - x_1x_2 + x_2^2$$

- (a) (5 points) find the global minima in the region $-3 \leq x_1 \leq 3$ and $-3 \leq x_2 \leq 3$.
- (b) (5 points) are there any other stationary points? If yes, what are they?
- (c) (5 points) plot the contour plot of $f(x)$ and verify your answers to the previous two questions.

Problem 2 (15 points). For the loss function $f(x, y) = x^2 + y^2 - 6xy - 4x - 5y$

- (a) (5 points) show analytically using Lagrange multipliers how to minimize the loss subject to constraints

$$\begin{aligned} y &\leq -(x - 2)^2 + 4, \text{ and} \\ y &\geq -x + 1. \end{aligned}$$

Hint: read about Lagrange Multipliers from <https://ocw.mit.edu/courses/18-02-multivariable-calculus-fall-2007/resources/lecture-13-lagrange-multipliers/>

- (b) (5 points) how is the optimal loss affected if the first constraint is changed to

$$y \leq -(x - 2)^2 + 4.1.$$

Estimate the difference and explain your answer. Remember that the gradient of the loss at a stationary point is a linear combination of the constraints with the Lagrange multipliers acting as the co-efficients.

- (c) (5 points) Write a Python script to confirm your results to parts (a) and (b). You can use the Scipy function `scipy.optimize.minimize` (<https://docs.scipy.org/doc/scipy/reference/optimize.html>) to perform this constrained optimization. List the code and your results in the solution PDF.

Problem 3 (15 points). Let X and Y be independent random variables taking values in $\{-1, 1\}$. We have $P(X = 1) = q$ while Y is uniformly distributed on $\{-1, 1\}$. Let $Z = XY$.

- (a) (5 points) Find the conditional distribution $P(Y | Z)$. Note that conditional probabilities are a function of $Z = z$.
- (b) (5 points) The conditional mean of Y given $Z = z$ is

$$E[Y | Z = z] = \sum_{y \in \{-1, 1\}} y P(y | z).$$

Find $E[Y | Z = z]$ as a function of z .

- (c) (5 points) The conditional mean of Y given Z (where the latter is viewed as a random variable) is

$$\mu_{Y|Z} = E[Y | Z] = \sum_{y \in \{-1, 1\}} y P(y | Z).$$

Since $\mu_{Y|Z}$ is a function of the random variable Z , it too is a random variable. Compute the probability distribution of $\mu_{Y|Z}$.

Problem 4 (25 points). In this problem, we will run linear regression using the “scikit-learn” library. You can install this library using

```
pip install scikit-learn
```

Load the Boston housing dataset (<https://www.kaggle.com/datasets/vikrishnan/boston-house-prices>) using

```
from sklearn.datasets import load_boston
ds = load_boston()

# explore the dataset
print(ds.keys())
print(ds.DESCR)
```

The variable `ds.data` contains the features with names `ds.feature_names` and the variable `ds.target` contains the value of the house. Our goal is to predict the value given the features using a linear model.

- (a) (15 points) If the i^{th} datum is denoted by $x^i \in \mathbb{R}^{13}$ and target by $y^i \in \mathbb{R}$, we intend to solve for weights $w \in \mathbb{R}^{13}$ and $b \in \mathbb{R}$ such that

$$y^i \approx w^\top x^i + b$$

for all pairs (x^i, y^i) dataset. A natural way to find w, b is to minimize the average of the *residuals*

$$\ell(w, b) := \frac{1}{2n} \sum_{i=1}^n (y^i - w^\top x^i - b)^2$$

where $n = 506$ is the number of samples in the dataset. Write the *data matrix* as $X \in \mathbb{R}^{506 \times 13}$ and the target matrix as $Y \in \mathbb{R}^{506}$; each row of X is therefore a datum x^i . We can now rewrite $\ell(w, b)$ as

$$\ell(w, b) = \frac{1}{2n} \|Y - Xw - b\mathbb{1}\|_2^2$$

where $\mathbb{1}$ is a vector of all ones. Compute an analytical expression for

$$w^*, b^* = \underset{w, b}{\operatorname{argmin}} \ell(w, b).$$

- (b) (10 points) Code up your expression for w^*, b^* . In order to check whether the linear model is accurate for the Boston housing dataset, we should evaluate the model on some held-out data. If the model predicts the values on this held-out data accurately, i.e., has a small residual, we can confidently use the linear model in practice. Split the 506 samples into 80% (= 405) samples that are used for computing w^*, b^* as above. Compute the average of the residuals on the remaining 20% of the data (= 101 samples); this is known as the validation error because the model w^*, b^* was not fitted on this data. Similarly the average of the residuals, namely $\ell(w, b)$, on the 80% data used to fit the weights is called the training error. Perform this experiment 2-3 times, each time sampling a different training set of size

80% from the original dataset. Report the mean and standard-deviation of the training and validation error.