CS5785 / ORIE5750 / ECE5414 - Homework 1

This homework is due on **Friday, September 8th, 2023 at 11:59PM EST**. The homework is split into programming exercises and written exercises. Upload your homework to Gradescope. Your submission will have two parts:

1. A write-up as a single `.pdf` file. Submit this under the `hw1-report` assignment in Gradescope.
2. Source code and data files for all of your experiments (AND figures) in `.ipynb` files (file format for IPython Jupyter Notebook). These files should be placed in a folder titled `hw1` and uploaded to the `hw1-code` assignment in Gradescope.

The write-up should contain a general summary of what you did, how well your solution works, any insights you found, etc. On the cover page, include the class name, and homework number. You are responsible for submitting clear, organized answers to the questions. You could use online LaTeX templates from Overleaf, under "Homework Assignment" and "Project / Lab Report".

Please include all relevant information for a question, including text response, equations, figures, graphs, output, etc. If you include graphs, be sure to include the source code that generated them. Please pay attention to Canvas for announcements, policy changes, etc. and Piazza for homework related questions.

If you need help

There are several strategies available to you.

- If you get stuck, we encourage you to post a question on Piazza [1]. That way, students can help each other and instructors can provide feedback and support.
- The professor and your TAs will offer office hours, which are a great way to get some one-on-one help. You can help us select the best times for office hours by completing the Canvas Survey titled "Office Hours: Select All That Works For You" and can be found under the Quizzes tab.
- You are allowed to use well known libraries such as `scikit-learn`, `scikit-image`, `numpy`, `scipy`, etc. in this assignment. Any reference or copy of public code repositories should be properly cited in your submission (examples include Github, Wikipedia, Blog, ChatGPT).

[1] https://piazza.com/cornell/fall2023/cs5785orie5750ece5414

PROGRAMMING EXERCISES

Please use different `.py` or `.ipynb` files for different parts

Part I. Warm Up

1. Let `a = np.array([1,2,3,4,5,6,7,8])`. Reshape `a` into a 2 by 4 matrix.
2. Let `a` be a pytorch tensor constructed with elements `[1,3,5,6]`, and let `b` be a tensor constructed with elements [5,6,8,9]. Write a sequence of lines, one each to perform the following operations on `a` and `b`. [Hint: use the function `torch.tensor()`.]
   - Elementwise addition
   - Elementwise multiplication
   - Elementwise power (each element of `a` raised to the power given by corresponding element of `b`)
   - Dot product between `a` and `b`
   - Dot product between an elementwise exponentiation of `a` with base e and an elementwise natural logarithm of `b`
3. Use `tensor` and `autograd` from the pytorch package to complete the following questions:
   (a) Calculate the gradient of

   $$g\left(x, y, z, k\right) = e^x x^2 + 3\, e^y y^2 + 5 e^z z^2 + 6 e^k k^2$$

   evaluated at the point: $\left(x = 5,\ y = 6,\ z = 8,\ k = 9\right)$.

   Hints: 1. you can rewrite the function g using the tensors [1, 3, 5, 6] and [5, 6, 8, 9] with correct tensor type; 2. set `requires_grad` to True for the correct tensor; 3. using the function '`.backward()`'; 4. obtain the gradient by calling '`.grad`'.

   (b) Let `A` be a matrix with values [[4,3],[7,9]] and `B` be a matrix with values [[3,5],[1,11]]. Calculate the gradient of the following function `f(A)` with respect to the entries of `A` evaluated at the point where `A` takes the above values.

   $$f(A) = \log \| A^T A B^T A A^T A B \|^2$$

   In the above expression $\| \cdot \|^2$ denotes the squared $L2$ norm, i.e., the sum of the squares of all entries of the matrix inside the norm expression.

   Hints: 1. to calculate matrix multiplication, you need to use the function `torch.matmul`; 2. to calculate $L2$ norm, you need to use the function `torch.norm()` and set $p = 2$.

   (c) Calculate the gradient of

   $$F\left(x,\ y\right) = \tanh(x) + \tanh\left(y\right)$$

   at the point $\left(x = 3,\ y = 7\right)$.
4. Let `a` be a torch integer tensor containing the values [1,2,3].

- convert `a` to a numpy array and store it under a new variable `b`
- convert `a` into a float tensor

5. Answer the following questions using the package `Numpy`:
   - What is the product of matrices of matrices $[[1,3,5],[2,1,5]]$ and $[[8,4],[3,6],[2,7]]$?
   - What is the Frobenius norm of the 1 x 3 matrix $[100,2,1]$?

Part II. The Housing Prices

1. Join the House Prices - Advanced Regression Techniques competition on Kaggle. Download the training and test data.

2. Give 3 examples of continuous and categorical features in the dataset; choose one feature of each type and plot the histogram to illustrate the distribution.

3. Pre-process your data, explain your pre-processing steps, and the reasons why you need them. (Hint: data pre-processing steps can include but are not restricted to: dealing with missing values, normalizing numerical values, dealing with categorical values etc.)

4. One common method of pre-processing categorical features is to use a one-hot encoding (OHE).

   Suppose that we start with a categorical feature x_j, taking three possible values: $x_j \in \{R, G, B\}$. A one-hot encoding of this feature replaces x_j with three new features: x_{jR}, x_{jG}, x_{jB}. Each feature contains a binary value of 0 or 1, depending on the value taken by x_j. For example, if $x_j = G$, then $x_{jG} = 1$ and $x_{jR} = x_{jB} = 0$.

   Give some examples of features that you think should use a one-hot encoding and explain why. Convert at least one feature to a one-hot encoding (you can use your own implementation, or that in pandas or scikit-learn) and visualize the results by plotting feature histograms of the original feature and its new one-hot encoding.

5. Using ordinary least squares (OLS), try to predict house prices on this dataset. Choose the features (or combinations of features) you would like to use or ignore, provided you justify your choice. Evaluate your predictions on the training set using the MSE and the R^2 score. For this question, you need to implement OLS in 2 ways: 1) using the `scikit-learn` package, and 2) using `autograd` from `pytorch`.

6. Compare and discuss the two results that you obtained in Q5. In addition, identify situations where applying gradient descent is more desirable in the context of linear regression.

7. Train your model using all of the training data (all data points, but not necessarily all the features), and test it using the testing data. Submit your results to Kaggle.

   Please submit 1) a screenshot that highlights your position on Kaggle, and 2) your code that incorporates the evaluation of your model on the test dataset.

WRITTEN EXERCISES

1. For each of the problem, identify whether it's more naturally characterized as a binary classification, multiclass classification, multilabel classification, regression, clustering, density modeling,

or RL problem.

(a) Given a stream of customers each characterized by some attributes, learn which ads to show them given that you can only show each customer one ad.

(b) Classify emails as spam or not spam.

(c) Given a news article, predict which topics it covers.

(d) Given a pair of images of faces, identify whether they depict the same person.

(e) Learn to play a board game against randomly matched opponents on the internet.

(f) Identify whether a new data point is expected given those you have seen before or extremely unlikely.

(g) Figure out whether a group of patients happens to naturally break down into some number of subgroups.

(h) Recognize celebrities based on photographs scraped from Twitter.

(i) Predict the starting salaries of new graduates based on their academic record.

(j) Identify the best (personalized) treatment among a set of drugs for a given chronic condition.

2. Based on the materials covered so far, in supervised machine learning, why must we make assumptions? Why can't we just learn from data alone?

3. Analytical solution of the Ordinary Least Squares Estimation. Consider we have a simple dataset of n labeled data $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})\}$, where data $x^{(i)} \in \mathbb{R}$ and $y^{(i)} \in \mathbb{R}$ is its corresponding label. We use a simple estimated regression function of:

$$\hat{y}^{(i)} = \theta_0 + \theta_1 x^{(i)}$$

Instead of gradient descent which works in an iterative manner, we try to directly solve this problem. We define the cost function as the residual sum of squares, parameterized by θ_0, θ_1:

$$J(\theta_0, \theta_1) = \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2$$

(a) Calculate the partial derivatives $\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$ and $\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$.

(b) Consider the fact that $J(\theta_0, \theta_1)$ has an unique optimum, which we denote as θ_0^*, θ_1^*. The analytical solution for minimizing θ_0^*, θ_1^* can be obtained by the following normal equations:

$$\frac{\partial}{\partial \theta_0} J(\theta_0^*, \theta_1) = 0$$
$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1^*) = 0$$

Prove the following proprieties:

$$\theta_0^* = \overline{y} - \theta_1 \overline{x}$$

and

$$\theta_1^* = \frac{\sum_{i=1}^n x^{(i)}(y^{(i)} - \overline{y})}{\sum_{i=1}^n x^{(i)}(x^{(i)} - \overline{x})}$$

(Note: $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x^{(i)}$ and $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y^{(i)}$.)

(c) For the optimal θ_0^*, θ_1^*, calculate the sum of the residuals $\sum_{i=1}^{n} e^{(i)} = \sum_{i=1}^{n} (y^{(i)} - (\theta_0^* + \theta_1^* x^{(i)}))$. What can you learn from the value of $\sum_{i=1}^{n} e^{(i)}$?