Assignment 1

CMSC 678 — Introduction to Machine Learning

Due Friday February 10, 11:59 PM

Item	Summary
Assigned	Monday January 31st
Due	Friday February 10
Topic	Math and Programming Review
Points	110

Machine learning often requires combining multivariate calculus, numerical optimization, and computational thinking. In this assignment you will review and cover some of the basic techniques that are necessary for understanding many machine learning methods.

You are to *complete* this assignment on your own: that is, the code and writeup you submit must be entirely your own. However, you may discuss the assignment at a high level with other students or on the discussion board. Note at the top of your assignment who you discussed this with or what resources you used (beyond course staff, any course materials, or public Discord discussions).

The following table gives the overall point breakdown for this assignment.

Question	1	2	3	4
Points	25	20	30	35

What To Turn In Turn in a **PDF** writeup that answers the questions; turn in all requested code necessary to replicate your results. Be sure to include specific instructions on how to build (compile) and run your code. Answers to the following questions should be long-form. Provide any necessary analyses and discussion of your results.

How To Submit Submit the assignment on the submission site:

https://www.csee.umbc.edu/courses/graduate/678/spring23/submit.

Be sure to select "`Assignment 1`."

Full Questions

0. (**0 points**) While there are a number of libraries that are popular to use for machine learning, especially when it comes to the neural/deep learning aspects of ML, Pytorch is a very popular library. Especially if you are not familiar with Pytorch, go through the Pytorch tutorials (`https://pytorch.org/tutorials/`) listed under "Introduction to PyTorch" (on the left). The most important, *general* ones for now are the "Basics," "Tensors," "Build the Neural Network," "Automatic Differentiation," and "Optimizing Model Parameters" tutorials. I recommend opening an interpreter, such as a Colab notebook, and running the code in the tutorial as you read it.

   There is nothing to turn in for this question.

1. (**25 points**) Hal Daumé III has a very nice "refresher" tutorial called "Math for Machine Learning:" `http://www.umiacs.umd.edu/~hal/courses/2013S_ML/math4ml.pdf`. Where indicated, some of the following questions are taken from that primer. I encourage everyone to read the primer, but especially if you have difficulty answering the following questions. In the following, assume $\log$ means natural log, unless specified otherwise.

   (A) **[Exercise 1.2]** Compute the derivative of the function $f(\theta) = \exp(-\frac{1}{2}\theta^2)$.

   (B) **[Exercise 1.5]** Compute the derivative of the function $f(\theta) = \log(\theta^2 + \theta - 1)$.

   (C) **C**ompute the derivative with respect to x of the function

   $$f(\theta, x, K) = \log(\sum_{k=1}^{K} \exp(k(x-\theta)^k)),$$

   for finite, positive, integral K.

   (D) **C**ompute the derivative with respect to x of the function

   $$f(\theta, x, K) = \log(\prod_{k=1}^{K} \exp(k(x-\theta)^k)),$$

   for finite, positive, integral K. (Hint: this is different from the previous problem. Remember properties of $\log$, where for a concrete example, think about $\log_b 4 = \log_b(2 * 2) = 2\log_b 2$.)

   (E) Let $K = 2$ and $x = 1.5$. Use Pytorch to verify the derivatives you computed for parts (C) and (D) at $\theta = 1$. Turn in your code. The values you find should be in as simplified a form as possible.

   (F) **[Exercise 7]** Compute the Euclidean, Manhattan, Maximum, and Zero norms on the three vectors $(1, 2, 3)$, $(1, -1, 0)$, $(0, 0, 0)$.

   (G) **[Exercise 14 and 19]** Given the matrix $A = \begin{pmatrix} 5 & 10 \\ -2 & 0 \\ 1 & -1 \end{pmatrix}$, compute the values $A^\intercal A$, $AA^\intercal$, and the traces $\mathrm{tr}(AA^\intercal)$, $\mathrm{tr}(A^\intercal A)$. Discuss how the traces relate to the Frobenius norm.

   (H) Using Pytorch, verify your answers to (G). Turn in your code.

   (I) **[Exercise 24]** For the multivariate function $f(u, v) = \exp(u^\intercal v)$, where $u, v \in \mathbb{R}^K$, compute the gradients $\nabla_u f$ and $\nabla_v f$.

2. (**20 points**) This question gets you acquainted with one of the most core building blocks in modern ML: a linear (tensor-tensor) computation. Feel free to open a REPL and play around with Pytorch to verify your answers!

(A) Describe how a matrix-matrix product can be computed as a collection of vector dot products.

(B) Referring to the `nn.Linear` module in Pytorch (see the documentation): Let A be a $3x2$ matrix and $b = 0$. When x is a vector, how many elements are in x?

(C) Referring to the `nn.Linear` module in Pytorch (see the documentation): Let A be a $3x2$ matrix and $b = 0$. When x is given as a row vector, explain in prose what is being computed.

(D) Referring to the `nn.Linear` module in Pytorch (see the documentation): Let A be a $3x2$ matrix and $b = 0$. Let x be as before, and let Y be a matrix with an arbitrary (but fixed) number of rows, and as many columns as elements in x. Explain in prose what is being computed when you provide Y as input to the linear layer. (When trying this out in Pytorch, if you want you can stack (replicate) x across the rows of Y.)

3. (**30 points**) This semester you will be reading published papers, analyzing them into a written form, and producing a paper or report as part of your graduate project. This question concerns proper scholarship and citations. This question is meant as review: the material should have been covered in an online class you took prior to enrolling at UMBC. If you are at all confused or unsure, *please ask*.

   Read CIML Ch. 2 (`http://ciml.info/dl/v0_99/ciml-v0_99-ch02.pdf`) and answer the following questions.

   (A) When used with a citation, directly quoting text and rewriting it in your own words are both two ways of correctly citing someone else's work. Using CIML Ch 2.1, provide an example of both methods. For the rewriting example, explain why your example is okay.

   (B) *Note:* for this question (3.(B)), and this question alone, copying text from the source or otherwise improperly citing will not be a violation of academic integrity. For all other questions and work done, however, proper scholarship is required, subject to the "Academic Honesty" section of the syllabus.

   Directly quoting text without a citation, lightly rewriting it without a citation, and near copying (with or without a citation) are all ways of incorrectly citing someone else's work; these are violations of academic integrity. Using CIML Ch 2.2, provide an example of all three methods. For each, explain why what you have written would be a violation and how you would fix it.

   (C) If (B) were part of group work and your partner quoted text without a citation, who in the group would be penalized?

   (D) Write a short (as a rough guide, around 200-300 words) summary of CIML Ch 2. In this summary, you should discuss what *you* took away from this chapter, and identify and discuss items that were confusing, underspecified, or counter-intuitive. Be sure to follow proper scholarship standards.

4. (**35 points**) For this question, you will be implementing the mathematical function [Eq-1] as a Pytorch layer. In order to compute this function, this function f depends on N binary values $y_i \in \{0, 1\}$. It also depends on two variables (ω_0 and ω_1):

$$f\left(\omega = (\omega_0, \omega_1)\right) = \sum_{i=1}^{N} \left[-\omega_{y_i} + \log\left(\sum_{j \in \{0,1\}} \exp(\omega_j) \right) \right]. \qquad \text{[Eq-1]}$$

   You may assume the log function refers to the natural logarithm (base e).

   For example, if $N = 2$ ($y_1 = 1, y_2 = 1$), then $f(\omega) = -2\omega_1 + 2\log\left(\exp(\omega_0) + \exp(\omega_1)\right)$.

(A) Let $N = 5$, where $y = (y_1 = 1, y_2 = 0, y_3 = 0, y_4 = 1, y_5 = 1)$. Compute the value $f(\omega_0 = 0.5, \omega_1 = -0.2)$.

(B) Write a Pytorch layer to compute [Eq-1]. Your code may assume that there are only the two weights ω_0 and ω_1. However, your code must be able to accept an arbitrary number of y_i values. I have written a stub example

https://colab.research.google.com/drive/1hW_4OqnWuWEbfEx47PVSTGfHHcAHGeik?usp=sharing.

you *may* use this as a starting point if you wish.

i. Use your code to verify the value you computed in (A).

ii. Use your code to compute $f(\omega_0 = 0.5, \omega_1 = -0.5)$ where $y = (y_1 = 1, y_2 = 0, y_3 = 0, y_4 = 1, y_5 = 1, y_6 = 0, y_7 = 1, y_8 = 1)$.

(C) Now, you'll compute the gradient of f.

i. Write the equation of the gradient when there are two variables (ω_0 and ω_1) but arbitrary N. That is, your answer should be symbolic and of the form $\nabla_\omega = (\frac{\partial f}{\partial \omega_0}, \frac{\partial f}{\partial \omega_1})$.

ii. Compute, by hand, the value of the gradient at $(\omega_0 = 0.5, \omega_1 = -0.2)$ where $y = (y_1 = 1, y_2 = 0, y_3 = 0, y_4 = 1, y_5 = 1)$.

iii. Use Pytorch to verify your computation in 4((C))ii. Provide these values in your writeup, and turn in your code.

(D) When $y = (y_1 = 1, y_2 = 0, y_3 = 0, y_4 = 1, y_5 = 1)$, use Pytorch to find the optimal values of ω_0 and ω_1. Provide these values in your writeup, and turn in your code.