CSCE 421: Machine Learning (Spring 2025)
Assignment #3

Due 11:59PM on 03/20/2025

1. You need to submit (1) a report in PDF and (2) your code files, both to Canvas.

2. Your PDF report should include (1) answers to the non-programming part, and (2) results and analysis of the programming part. For the programming part, your PDF report should at least include the results you obtained, for example the accuracy, training curves, parameters, etc. You should also analyze your results as needed.

3. Please name your PDF report "HW#_FirstName_LastName.pdf". Please put all code files into a compressed file named "HW#_FirstName_LastName.zip". Please submit two files (.pdf and .zip) to Canvas (i.e., do not include the PDF file into the ZIP file).

4. Only write your code between the following lines. Do not modify other parts.

   ### YOUR CODE HERE

   ### END YOUR CODE

5. LFD refers to the textbook "Learning from Data".

6. All students are highly encouraged to typeset their reports using Word or LaTeX. In case you decide to hand-write, please make sure your answers are clearly readable in scanned PDF.

7. Unlimited number of submissions are allowed and the latest one will be timed and graded. If you make a resubmission after the due date, it will be considered late.

8. Please read and follow submission instructions. No exception will be made to accommodate incorrectly submitted files/reports.

9. Please start your submission to Canvas at least 15-30 minutes before the deadline, as there might be latency. We do NOT accept E-mail submissions.

---

1. (10 points) This question is related to the loss functions we discussed in class.

   (a) Describe what are hinge loss, logistic regression loss, and 0-1 loss mathematically. Describe their similarities and differences using the unified picture we developed in class.

   (b) By relying on the result in the above question, consider a data point that is correctly classified and distant from the decision boundary. Why would SVM's decision boundary be unaffected by this point, but the one learned by logistic regression be affected?

2. (20 points) Exercise 7.8 (e-Chap:7-15) in the book "Learning from Data". For your convenience, the question is repeated below:

   Exercise 7.8: Repeat the computations in Example 7.1 for the case when the output transformation is the identity. You should compute $\mathbf{s}^{(\ell)}, \mathbf{x}^{(\ell)}, \boldsymbol{\delta}^{(\ell)}$ and $\partial \mathrm{e}/\partial \mathrm{W}^{(\ell)}$.

   Note the "output transformation" refers to the nonlinear function in the output layer only.

3. (10 points) You'd like to train a fully-connected neural network with 5 hidden layers, each with 10 hidden units (i.e., $d^{(\ell)} = 10$ for these hidden layers using the notations in the textbook). The input is 20-dimensional and the output is a scalar. What is the total number of trainable parameters in your network (include bias in your calculation)?

4. (60 points) **Multi-layer perceptron models for Handwritten Digits Classification**: In this assignment, you will implement the multi-layer perceptron model using PyTorch on a partial dataset from MNIST. In this classification task, the model will take a 16×16 image of handwritten digits as inputs and classify the image into three different classes (0,1,2). The "data" folder contains the dataset which has already been split into a training set and a testing set. All data examples are saved in dictionary-like objects using "npz" file. For each data sample, the dictionary key 'x' indicates its raw features, which are represented by a 256-dimensional vector where the values between $[-1, 1]$ indicate grayscale pixel values for a 16×16 image. In addition, the key 'y' is the label for a data example, which can be 0, 1, or 2. The "code" folder provides the starting code (main training logic has already been implemented). You must implement the models using the starting code. Please learn the PyTorch tutorial before starting this assignment: https://pytorch.org/tutorials/beginner/basics/intro.html

   (a) (20 points) In the function *prepare_X* from "code/DataReader.py", please copy your implementation from HW2, and don't change other parts in "code/DataReader.py". Then, implement the multi-layer perceptron model in the "code/MLP.py". Please use only one hidden layer in the MLP (Hint: you can use *torch.nn.Linear* class in PyTorch). In this assignment, we will use a new nonlinear function ReLU that will be discussed later in this class (PyTorch has implemented this function so you do not need to be concerned about how to implement it). It outputs the same value if the input is positive and outputs zero if the input is negative (Hint: use *torch.nn.functional.relu*). The sequential layers are:

   Linear → ReLU → Linear (3 output channels)

   (b) (20 points) Implement *evaluation* function in "code/main.py". This function should evaluate classification accuracy of the trained model on the test set. Please remember to load the best model checkpoint (based on validation accuracy) before running the evaluation on the test set.

   (c) (20 points) Tune the *HIDDEN_SIZE* hyperparameter in "code/main.py" to find the best model performance on the test set. Please include the *HIDDEN_SIZE* you've tried and the corresponding classification accuracy on the test set in your submission.