Homework II

STAT/CS 6685/5685 - Fall semester 2023

Due: Friday, Sep 29, 2023 - 5:00 PM

Please upload all relevant files & solutions into Canvas. Include your answers (minus code) in a single PDF titled `lastname_firstname_HW2.pdf`, and all your code in a single jupyter notebook titled `lastname_firstname_HW2Code.ipynb`. Any requested plots should be sufficiently labeled for full points.

Unless otherwise stated, programming assignments should use built-in functions in Python, Tensorflow, and PyTorch. In general, you may use the `scipy` stack [1]; however, exercises are designed to emphasize the nuances of machine learning and deep learning algorithms - if a function exists that trivially solves an entire problem, please consult with the TA before using it.

Stater code is provided for all homeworks at `https://github.com/KevinMoonLab/STAT-6685`. You are not required to use this code, although it is highly recommended. In some cases you will be asked to describe what certain parts of the code are doing.

Problem 1 - 10 points

Read this paper on avoiding machine learning pitfalls: `https://arxiv.org/pdf/2108.02497v1.pdf`. Choose two sections. Write a paragraph for each section that summarizes some of the key principles you learned from it.

Problem 2 - (6685-30pts, 5685-20pts)

1. (**6685**-10pts) Let $f : \mathbb{R}^d \to \mathbb{R}$. Show that if f is strictly convex, then f has at most one global minimizer.

2. (**5685**-7pts) Use the Hessian to give a simple proof that the sum of two convex functions is convex. You may assume that the two functions are twice continuously differentiable.
   Hint: You may use the fact that the sum of two PSD matrices is also PSD.

3. (**6685**-4pts, **5685**-7pts) Consider the function $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} + c$ where A is a symmetric $d \times d$ matrix. Derive the Hessian of f. Under what conditions on A is f convex? Strictly convex?

4. (**6685**-4pts) Using the definition of convexity, prove that the function $f(x) = x^3$ is not convex.

5. (**6685**-4pts) Using the fact that $f(x) = x^3$ is twice continuously differentiable, prove that f is not convex.

6. (**6685**-4pts, **5685**-6pts) A function f is concave if $-f$ is convex. Prove that for all $x > 0$, the function $f(x) = \ln(x)$ is concave.

7. (**6685**-4pts) Let $f(x) = ax + b$ where $a, b \in \mathbb{R}$. f is called an *affine function* (a linear function plus an offset). Prove that f is both convex and concave. Does f have a global minimum or a global maximum? Are there any other functions that are twice continuously differentiable and both convex and concave that do not have the form of an affine function? Explain your reasoning.

Problem 3 - (6685-20pts, 5685-30pts)

1. (5 pts) Suppose we take all the weights and biases in a network of perceptrons, and multiply them by a positive constant, c > 0. Show that the behavior of the network doesn't change. (Exercise in Ch1 Nielsen book)
   Hint: Consider a single perceptron first. For a fixed input, does multiplying the weights and bias by a positive constant change the output of the perceptron? Now argue that the output of the entire network is unchanged.

2. (5 pts) Given the same setup of `problem 2.1` - a network of perceptrons - suppose that the overall input to the network of perceptrons has been chosen and fixed. Suppose the weights and biases are such that $wx + b \neq 0$ for the input x to any particular perceptron in the network. Now replace all the perceptrons in the network by sigmoid neurons, and multiply the weights and biases by a positive constant $c > 0$. Show that in the limit as $c \to \infty$ the behavior of this network of sigmoid neurons is exactly the same as the network of perceptrons. How can this fail when $wx + b = 0$ for one of the perceptrons? (Exercise in Ch1 Nielsen book)
   Hint: Use a similar approach as in problem 2.1 where you first consider the behavior of a single sigmoid neuron and then extend to the entire network.

3. (**6685**-5pts, **5810**-10pts) For each possible input of the MLP in Figure 1, calculate the output. I.e., what is the output if $X = [0, 0, 0]$, $X = [0, 0, 1]$, etc. You should have 8 cases total.

4. (**6685**-5pts, **5810**-10pts) If we change the perceptrons in Figure 1 to sigmoid neurons what are the outputs for the same inputs (e.g., inputs of [0,0,0], [0,0,1], ...)?

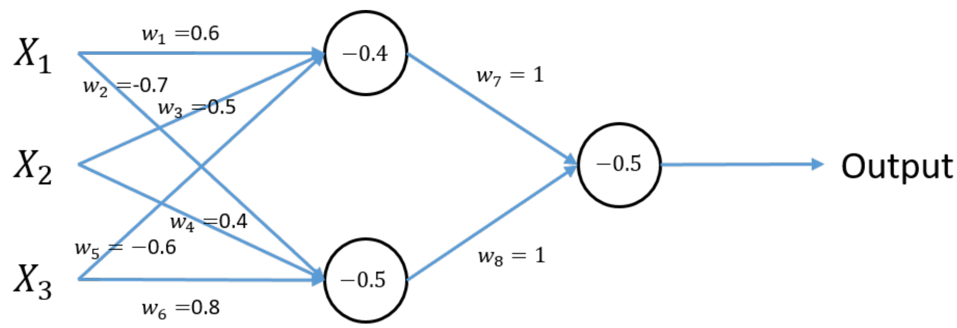

Figure 1: Multilayer perceptron with three inputs and one hidden layer.

Problem 4 - 5 points

For each of the scenarios below, describe whether it can be solved as a regression problem or classification problem, or potentially both. Give your reasoning.

1. Determine which kind of tree is in a picture.
2. Predict when someone will die based on their current health conditions.
3. Predict the probability that a person will survive for more than 1 year after a cancer diagnosis based on their current health conditions.
4. Predict how many stars out of five a person will rate a product on Amazon based on previous reviews and browsing history.
5. Text prediction. I.e. predict the next word in a phrase or sentence.

Problem 5 - 35 points

In this problem, you will use PyTorch (or Tensorflow) to build and fit regression models and answer the corresponding questions. You will apply linear regression to a simulated second-order polynomial dataset. The function to generate the data has been provided at the starter code:

`https://github.com/KevinMoonLab/STAT-6685/tree/master/HW-2`.

The function `generate_poly_data(N, sigma)` takes two arguments: the sample size `N` and the noise level `sigma`. The function returns data that is used to generate a response value (i.e. the y variable) that follows a polynomial function. Starter code for performing polynomial regression in PyTorch is also provided in this notebook. To answer the questions below, you may find following the instructions in the notebook marked with `***` helpful. You are not required to use the starter code, although it is highly recommended. If you use Tensorflow, you may build your model from scratch and similar grading criterion will be used for each question.

1. (15 pts) Implement and fit a linear regression model using PyTorch (or Tensorflow). That is, define a loss function and optimize it using an optimizer. Use stochastic gradient descent (SGD) for this. (5 pts). Complete the training routine passing the inputs (i.e. the xs) through the network, computing the loss, calling the back-propagation function, and computing a step with the optimizer (10 pts). Plot the training data and the prediction function using the test data (see the starter code for how to do this).

2. (10 pts) Let's fit the data to a 2nd degree polynomial. Augment the data matrix by adding a new feature x^2. So for each data point, you should have an x^2 term, a x term, and an offset. Fit the model, and plot the training data and the predictions of the test data. Then print the parameters of your regressionLayer or your model.module. Did you obtain something close to the true parameters?

3. (5 pts) Using a similar approach, fit the data to a degree 5 polynomial. Does the result meet your expectation compared to previous two models?.
   Hint: you might need to play with the learning rate until you obtain a good value

4. (5 pts) Create a for loop for different values of `sigma = [0.1, 0.5, 1]`, and `N = [15, 100]`. Fit a degree 5 polynomial model with different values for regularization `weight_decay = [0, 0.2, 0.5]`. Compute the testing error. Under each noise setting, which regularization parameter worked the best?

Include your answers and a picture of your adder in a PDF titled `<lastname and initials>_HW2.pdf`. Include your answers and plots in your PDF file.

Optional Problem

This problem will not be graded and thus should not be turned in. However, it will give you practice in training a neural network.

Run the Python code given in Chapter 1 of the Nielsen book in the section titled "Implementing our network to classify digits". You can find a link to the code at the beginning of the section. Verify that you understand each line of the code and that you obtain the same results as given in the book.

References

[1] "The scipy stack specification." [Online]. Available: https://www.scipy.org/stackspec.html