**The University of Texas at Austin**

**Introduction to Deep Learning - Spring 2023**

**Homework - 1**

**Dimakis** **Due: Feb 9, 11:59 PM**

---

**Question 1**

In this question, we will define tensors in PyTorch and use them to find the derivative of a linear function w.r.t its variables. We then find the mean squared error and conclude with finding the optimum parameters of a linear model.

**1.1** For the function y = f(x) = w.x + b, where w=[2,1] and b=3, find the partial derivates of y w.r.t the components of x (i.e.: dy/dx1 and dy/dx2) at x = [4,2] both on paper and using PyTorch.

**1.2** For a model y = f(x), the predicted and true values are as follows:

y_true = [0,1,1,0]

y_pred = [0.1,0.95,1.10,0.2]

Find the mean squared error both on paper and using PyTorch. In PyTorch, solve it using an inbuilt function and also by defining your own squared error function.

**1.3** You are given the following dataset (note that it describes an XOR function):

| x1 | x2 | y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Assume that we fit a linear function, y = w.x + b to this dataset with w = [2,1] and b=3.

i) Find the mean squared error (loss) over this dataset for the above weights and bias.

ii) Which direction should the weights move in to decrease the loss by maximum amount? Find that direction both on paper and using PyTorch.

iii) For what values of w and b is the loss minimum? Solve on paper only.

**Extra-credit:**
Can you solve for the optimum w and b using PyTorch?
**Hint:** use gradient descent

**Question 2**
In this question, we will learn to perform 1-d and 2-d convolutions using different strides on a grayscale image.

**2.1** Given a 1-d input x=[1,-1,3,4,4] and a kernel=[1,1], find the 1-d convolution for stride=1 and stride=2. Solve both on paper and using pyTorch.

**2.2** You're given the following grayscale image (any matrix in 2 dimensions is a grayscale image):

| | | | |
|---|---|---|---|
| 0.1 | -0.6 | 0.4 | 0.8 |
| -0.4 | 0.3 | 0.9 | 0.2 |
| 0.5 | 0.2 | 0.8 | -0.7 |
| 0.3 | 0.7 | -0.4 | 0.1 |

i) You're given 2 2*2 filters (kernels): [[1,0],[0,1]] and [[0,1],[1,0]]. Find the output when the image is convolved using each filter with stride=1. Solve both on paper and by using pyTorch.

Note that you'll get 2 outputs - one for each filter - each output is known as a channel.

ii) What are the dimensions of each channel in (i)? What will be the dimensions of the output when stride=2?

ii) The given image is a 2-d matrix. How can you convolve it so that the output channel has only one dimension?

iv) Perform a 2*2 max-pooling with stride=1 on the image both on paper and using PyTorch. What are the dimensions of the channel after max-pooling?

**Extra-credit**

If the size of the output channel after performing a 2*2 convolution (stride=1) and a 2*2 max-pooling (stride=1) is the same, why do you think we needed max-pooling when we could perform convolution to decrease the size

of the image. Think in terms of the advantages that max-pooling may offer over convolution. When do you think a max-pooling operation may not be advantageous (and in fact may hurt the network)?

**Question 3**

This question deals with the addition of perturbations to an image x to create an adversarial example from that image.

**3.1 Identify True/False**

i) In FGSM method, the parameters of the trained model change.

ii) In FGSM method, the pixel values of input image changes.

iii) FGSM can only be used for undirected adversarial attacks (An undirected adversarial attack is one in which the aim is to only perturb the original image in the direction of maximum loss. The attack does not care about which incorrect class the input example is classied into after perturbation).

iv) PGD always finds points inside the threat model.

**3.2** Solve this question assuming that you're the attacker. You're given a linear classifier which classifies each input into a dog or a cat:

$$f(x) = Pr(x = cat) = sigmoid(w'x)$$

Given w = [1,1].

i) It is given that $x_1 = [2, 1]$ represents a cat. What is $Pr(x = dog)$ and $Pr(x = cat)$?

ii) Now you want to change the components of x (i.e, change from [2,1] to something else) so that the probability of $x_1$ being a cat decreases. However, you can only change x acc. to the threat model $||x - x_1||_{inf} <= 0.1$. The threat model specifies the region in which the input $x_1$ is allowed to vary.
Plot the threat model on a 2-d graph and specify the co-ordinates of the corners of the quadrilateral thus formed.

iv) Perform an undirected attack to decrease the probability of $x_1$ being a cat with the following step sizes : 0.001, 0.1, and 1. Perform the attack in 2 ways - first by using the sign of gradients, and second by using the actual values of gradients.

Which of these FGSM attacks lies in the threat model?

v) Perform a directed attack to increase the probability of $x_1$ being a dog with the following step sizes : 0.001, 0.1, and 1. Perform the attack in 2 ways - first by using the sign of gradients, and second by using the actual values of gradients.

Which of these attacks lies in the threat model?

vi) What are your observations from the results in (iv) and (v)?