1 [12 points]

Consider a perceptron with two inputs, one output, and a threshold activation function. If the two weights are $w_1 = 1$ and $w_2 = 1$ and the bias is -0.4, then

1. What is the output of the perceptron for the input values (0,0), (0,1), (1,0), (1,1), (0,0.5), (0.5,0), (0.5,0.5)? (4 points)
2. Plot the decision boundary for this perceptron using the points in the previous question. Is the decision boundary linear? (5 points)
3. If we assume that the inputs are binary, i.e., each input can be either 0 or 1. What logical function does this perceptron correspond to? (3 points)

1. $w = [1, 1]^T$, $b = -0.4$ $\quad \sigma(z) = \begin{cases} 1 & z \geq 0 \\ 0 \end{cases}$

let $x^{(1)} = (0,0)$ $\quad \sigma(x^1) = 0$

$x^{(2)} = (0.1)$ $\quad \sigma(x^2) = 1$

$x^{(3)} = (1,0)$ $\quad \sigma(x^3) = 1$

$x^{(4)} = (1,1)$ $\quad \sigma(x^4) = 1$

$x^{(5)} = (0, 0.5)$ $\quad \sigma(x^5) = 1$

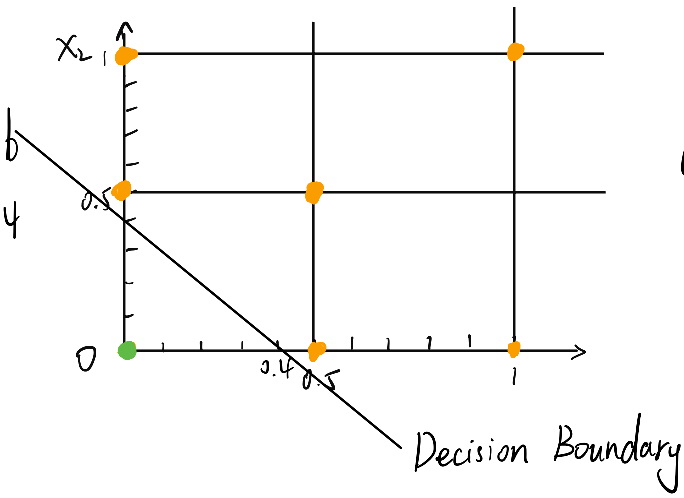
$x^{(6)} = (0.5, 0)$ $\quad \sigma(x^6) = 1$

$x^{(7)} = (0.5, 0.5)$ $\quad \sigma(x^7) = 1$

2. $z = w^T x + b$

$= w_1 x_1 + w_2 x_2 + b$

$= x_1 + x_2 - 0.4$



Orange : 1

Green : 0

It is linear.

3. OR Gate

As the inputs are Binary. $x^{(1)} = [0 \ 0]^T$ $\quad \sigma^1 = 0$ $\quad$ -0.4

$x^{(2)} = [0 \ 1]^T$ $\quad \sigma^2 = 1$ $\quad$ 0.6

$x^{(3)} = [1 \ 0]^T$ $\quad \sigma^3 = 1$ $\quad$ 0.6

$x^{(4)} = [1 \ 1]^T$ $\quad \sigma^4 = 1$ $\quad$ 1.6

2 [5 points]

Output the weights and biases of a perceptron that represents:

1. Logical NAND operation (2.5 points)
2. Logical NOT operation (2.5 points)

$z = w_1x_1 + w_2x_2 + b$

1. NAND Gate

x_1	x_2	Output
0	0	1
0	1	1
1	0	1
1	1	0

Since the output is 0 when both inputs are 1, each weight should be negative.

Try $w_1 = -1$, $w_2 = -1$, put $[1\ 1]^T$ in it,

as we need $z \geq 0$, $\begin{cases} -1 + b \geq 0 \\ -2 + b < 0 \end{cases}$

Thus, $1 \leq b < 2$ when $w_1 = -1$, $w_2 = -1$

One possible answer is $w_1 = -1$, $w_2 = -1$, $b = 1$

2. NOT Gate

$z = wx + b$

x_1	Output
0	1
1	0

$0 \cdot w + b \geq 0$

$b \geq 0$

$1 \cdot w + b < 0$

$b < -w$

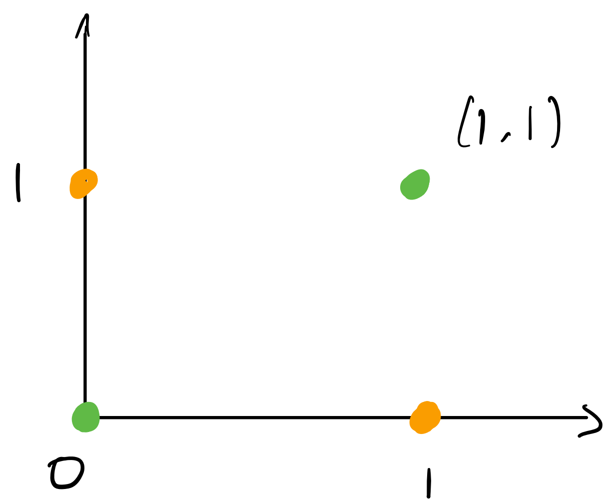
Thus, $w_1 = -1$, $b = 0.5$ is a possible answer.

3 [10 points]

Consider the operation of a 2 input XOR gate. It outputs a 1 when the number of inputs are odd. Otherwise it outputs a 0.

1. Plot a graph that shows the class label - 0 or 1 based on the output of the XOR gate for input values (0,0), (0,1), (1,0), (1,1). (4 points)
2. Are you able to obtain linear separability in the two classes as per the plot above? (2 points)
3. Can you construct a perceptron for the 2 input XOR? If yes, write the weights and bias. If no, why? (4 points)

1.



Orange : 1
Green : 0

XOR Gate

x_1	x_2	Output
0	0	0
0	1	1
1	0	1
1	1	0

$$\sigma(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \end{cases}$$

2. No. it's not linearly separable

3.
$$\begin{cases} w_1 \cdot 0 + w_2 \cdot 0 + b < 0 \Rightarrow 0 \\ w_1 \cdot 0 + w_2 \cdot 1 + b \geq 0 \Rightarrow 1 \\ w_1 \cdot 1 + w_2 \cdot 0 + b \geq 0 \Rightarrow 1 \\ w_1 \cdot 1 + w_2 \cdot 1 + b < 0 \Rightarrow 0 \end{cases}$$

By the Graph and the inequations above, it is not possible to do it. Since there is no single straight line to seprate the inputs that has outputs 0 and 1.

4 [20 points]

Consider the data shown in Table 1:

1. Initialize a perceptron model with weight $w = 0.1$ and bias $b = 0.1$. Using step size $\eta = 0.1$, compute the values of the weight and bias after 1 epoch of the perceptron algorithm. Iterate through the inputs in the following order: 1, 2, 3, 4. (5 points)
2. Now, initialize an Adaline model with weight $w = 0.1$ and bias $b = 0.1$ and calculate the loss function, i.e., the total loss over the training examples. (5 points)
3. Using step size $\eta = 0.1$, compute the values of the weight and bias for the Adaline model in the previous question after 1 epoch of the gradient descent algorithm. (7 points)
4. Does the order of inputs matter in the gradient descent algorithm? Please explain. (3 points)

x	y
1	0
2	0
3	1
4	1

Table 1: Input Data

1. Initial $w = 0.1$
Initial $b = 0.1$
Learning Rate $\eta = 0.1$

For $x = 1,\ y = 0$

$w \times x + b = 0.1 \times 1 + 0.1 = 0.2 \geq 0$

$\hat{y} \Rightarrow 1$. which is incorrect.

Using $\eta = 0.1$,

$w = w + \eta(y^{(i)} - \hat{y}^{(i)})x = 0.1 + 0.1 \cdot (0 - 1) \cdot 1 = 0$

$b = b + \eta(y^{(i)} - \hat{y}^{(i)}) = 0.1 + 0.1 \cdot (-1) = 0$

For $x=2$, $y=0$

$w \times x + b = 0 \times 2 + 0 = 0 \geq 0$ $\quad \hat{y} \Rightarrow 1$

$w = 0 + 0.1(0-1)2 = -0.2$

$b = 0 + 0.1(-1) = -0.1$

For $x=3$, $y=1$

$wx + b = -0.2 \cdot 3 - 0.1 = -0.7 < 0$ $\quad \hat{y} \Rightarrow 0$

$w = -0.2 + 0.1(1-0)3 = 0.1$

$b = -0.1 + 0.1(1-0) = 0$

For $x=4$, $y=1$

$wx + b = 0.1 \cdot 4 + 0 = 0.4 \geq 0$ $\quad \hat{y} \Rightarrow 1$

$w = 0.1 + 0.1(0)4 = 0.1$

$b = 0 + 0.1 \cdot (0) = 0$

After 1 epoch, $w = 0.1$, $b = 0$.

2. Now, initialize an Adaline model with weight $w = 0.1$ and bias $b = 0.1$ and calculate the loss function, i.e., the total loss over the training examples. (5 points)

$$L(w, b) = \frac{1}{n} \sum_{i=1}^{n} (y^{(i)} - \sigma(z^{i}))^2$$

For $x=1$, $y=0$ $\quad 0.1 \cdot 1 + 0.1 = 0.2$

$L = \frac{1}{4}(0 - 0.2)^2 = 0.01$

For $x=2$, $y=0$ $\quad 0.1 \cdot 2 + 0.1 = 0.3$

$L = \frac{1}{4}(0 - 0.3)^2 = 0.0225$

For $x=3$, $y=1$ $\quad 0.1 \cdot 3 + 0.1 = 0.4$

$L = \frac{1}{4}(1 - 0.4)^2 = 0.09$

For $x=4$, $y=1$ $\quad 0.1 \cdot 4 + 0.1 = 0.5$

$L = \frac{1}{4}(1-0.5)^2 = 0.0625$

Total loss function $= 0.01 + 0.0225 + 0.09 + 0.0625 = \boxed{0.185}$

3. Using step size $\eta = 0.1$, compute the values of the weight and bias for the Adaline model in the previous question after 1 epoch of the gradient descent algorithm. (7 points)

$\eta = 0.1$

$w = 0.1$

$b = 0.1$

$$\frac{\delta L}{\delta w_j} = -\frac{2}{n}\sum_{i=1}^{n}\left(y^{(i)} - \sigma'z^{(i)}\right)x_j^{(i)}$$

$$w \leftarrow w - \eta\frac{\delta L}{\delta w_i}$$

$$\frac{\delta L}{\delta b} = -\frac{2}{n}\sum_{i=1}^{n}\left(y^{(i)} - \sigma'z^{(i)}\right)$$

$$b \leftarrow b - \eta\frac{\delta L}{\delta b}$$

$$w = 0.1 - \left[(-0.2)\times 1 + (-0.3)\times 2 + 0.6\times 3 + 0.5\times 4\right]\cdot 0.1 \cdot -\frac{2}{4}$$

η

$$= 0.1 + 0.3 \cdot 0.5$$

$$= \underline{0.25}$$

$$b = 0.1 - \left[-0.2 - 0.3 + 0.6 + 0.5\right]\cdot 0.1 \cdot -\frac{2}{4}$$

$$= 0.1 + 0.06 \cdot 0.5$$

$$= \underline{0.13}$$

4. Does the order of inputs matter in the gradient descent algorithm? Please explain. (3 points)

No, Because the total gradient will be the same.

5 [30 points]

Train a Perceptron to the entire Pima Indians data set (Available at `https://www.kaggle.com/datasets/kumargh/pimaindiansdiabetescsv/data`). Take a look at the data card to understand the dataset. Do not split the data into training and test for this problem!

1. What are the input features and the output classes? (2 points)
2. Experiment with different learning rates and report the highest (training set) classification accuracy you can obtain. (10 points)
3. For the learning rate that gives the highest classification accuracy, plot the number of misclassification errors against the number of epochs, similar to Figure 2.7 from the textbook. (8 points)

Submit your code in a file named pimaPerceptron.py (10 points)

1. input:
   preg = Number of times pregnant
   plas = Plasma glucose concentration a 2 hours in an oral glucose tolerance test
   pres = Diastolic blood pressure (mm Hg)
   skin = Triceps skin fold thickness (mm)
   test = 2-Hour serum insulin (mu U/ml)
   mass = Body mass index (weight in kg/(height in m)^2)
   pedi = Diabetes pedigree function
   age = Age (years)

   output: Class variable (1:tested positive for diabetes, 0: tested negative for diabetes)

2&3: Check the code and README file in Zip.

```
Learning Rate: 0.01, Iterations: 100, Accuracy: 0.6688396349413298
Learning Rate: 0.01, Iterations: 500, Accuracy: 0.6088657105606258
Learning Rate: 0.01, Iterations: 1000, Accuracy: 0.6688396349413298
Learning Rate: 0.05, Iterations: 100, Accuracy: 0.6297262059973925
Learning Rate: 0.05, Iterations: 500, Accuracy: 0.6414602346805737
Learning Rate: 0.1, Iterations: 100, Accuracy: 0.5658409387222947
Learning Rate: 0.1, Iterations: 500, Accuracy: 0.6140808344198174
Learning Rate: 0.1, Iterations: 1000, Accuracy: 0.7001303780964798
Learning Rate: 0.5, Iterations: 100, Accuracy: 0.6701434159061278
Learning Rate: 0.5, Iterations: 500, Accuracy: 0.6962190352020861
Learning Rate: 0.5, Iterations: 1000, Accuracy: 0.6349413298565841
Best Accuracy: 0.7001303780964798 with Learning Rate: 0.1 and Iterations: 1000
```