1. **Cross Entropy Loss**
In Lecture 3, we derived a formula to compute the partial derivative of the loss function with respect to the parameters of the model, *i.e.*, w and b. The loss function ($\mathscr{L}$) that we considered was the mean squared error loss. Let us assume there is only one point in the dataset, (x, y). We now define a new loss function known as the cross entropy loss function as follows,
$$\mathscr{L}(w, b) = -y * \log f(x)$$
where,
$$f(x) = \Big(\frac{1}{1 + e^{-(wx+b)}}\Big)$$
and w and b are the parameters of the model. Note that y is the true value given x whereas $f(x)$ is the output of the model given x as input.
Derive an expression for the partial derivative of the cross-entropy loss function with respect to w and b and select the correct option from the options given below.

A. $\nabla w = \frac{\partial \mathscr{L}(w,b)}{\partial w} = y * (1 - f(x)) * x$
$\nabla b = \frac{\partial \mathscr{L}(w,b)}{\partial b} = y * (1 - f(x))$

B. $\nabla w = \frac{\partial \mathscr{L}(w,b)}{\partial w} = -y * (1 - f(x))$
$\nabla b = \frac{\partial \mathscr{L}(w,b)}{\partial b} = -y * (1 - f(x))$

C. $\nabla w = \frac{\partial \mathscr{L}(w,b)}{\partial w} = -y * (1 - f(x)) * x$
$\nabla b = \frac{\partial \mathscr{L}(w,b)}{\partial b} = -y * (1 - f(x))$

Solution: The partial derivative of the cross-entropy loss function with respect to w can be derived as follows :

$$\begin{aligned}
\mathscr{L}(w, b) &= -y * \log f(x) \\
\nabla w = \frac{\partial \mathscr{L}(w, b)}{\partial w} &= \frac{\partial}{\partial w}[-y * \log f(x)] \\
\nabla w &= -y * \frac{\partial}{\partial w}[\log f(x)] \\
&= -y * [\frac{1}{f(x)} * \frac{\partial}{\partial w}(f(x))] \\
&= -y * \frac{1}{f(x)} * \frac{\partial}{\partial w}\Big(\frac{1}{1 + e^{-(wx+b)}}\Big) \qquad (1)
\end{aligned}$$

Let's find the partial derivative of $f(x)$ with respect to w. It can be calculated as follows,

$$\begin{aligned}
\frac{\partial f}{\partial w} &= \frac{\partial}{\partial w}\Big(\frac{1}{1+e^{-(wx+b)}}\Big) \\
&= \frac{-1}{(1+e^{-(wx+b)})^2}\frac{\partial}{\partial w}(e^{-(wx+b)})) \\
&= \frac{-1}{(1+e^{-(wx+b)})^2} * (e^{-(wx+b)})\frac{\partial}{\partial w}(-(wx+b))) \\
&= \frac{-1}{(1+e^{-(wx+b)})} * \frac{e^{-(wx+b)}}{(1+e^{-(wx+b)})} * (-x) \\
&= \frac{1}{(1+e^{-(wx+b)})} * \frac{e^{-(wx+b)}}{(1+e^{-(wx+b)})} * (x) \\
&= f(x) * (1-f(x)) * x \qquad (2)
\end{aligned}$$

From (1) and (2),

$$\begin{aligned}
\nabla w = \frac{\partial \mathscr{L}(w,b)}{\partial w} &= -y * \frac{1}{f(x)} * f(x) * (1-f(x)) * x \\
&= -y * (1-f(x)) * x
\end{aligned}$$

Similarly, the partial derivative of the cross-entropy loss function with respect to b can be derived as follows :

$$\begin{aligned}
\mathscr{L}(w,b) &= -y * \log f(x) \\
\nabla b = \frac{\partial \mathscr{L}(w,b)}{\partial b} &= \frac{\partial}{\partial b}[-y * \log f(x)] \\
\nabla b &= -y * \frac{\partial}{\partial b}[\log f(x)] \\
&= -y * [\frac{1}{f(x)} * \frac{\partial}{\partial b}(f(x))] \\
&= -y * \frac{1}{f(x)} * \frac{\partial}{\partial b}\Big(\frac{1}{1+e^{-(wx+b)}}\Big) \qquad (3)
\end{aligned}$$

Let's find the partial derivative of $f(x)$ with respect to b. It can be calculated as follows,

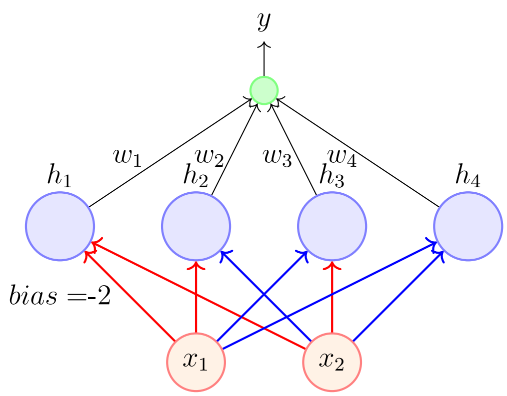
$$
\begin{aligned}
\frac{\partial f}{\partial b} &= \frac{\partial}{\partial b}\Big(\frac{1}{1+e^{-(wx+b)}}\Big) \\
&= \frac{-1}{(1+e^{-(wx+b)})^2}\frac{\partial}{\partial b}(e^{-(wx+b)})) \\
&= \frac{-1}{(1+e^{-(wx+b)})^2} * (e^{-(wx+b)})\frac{\partial}{\partial b}(-(wx+b))) \\
&= \frac{-1}{(1+e^{-(wx+b)})} * \frac{e^{-(wx+b)}}{(1+e^{-(wx+b)})} * (-1) \\
&= \frac{1}{(1+e^{-(wx+b)})} * \frac{e^{-(wx+b)}}{(1+e^{-(wx+b)})} \\
&= f(x) * (1 - f(x)) \qquad (4)
\end{aligned}
$$

From (1) and (2),

$$
\begin{aligned}
\nabla b = \frac{\partial \mathscr{L}(w,b)}{\partial b} &= -y * \frac{1}{f(x)} * f(x) * (1 - f(x)) \\
&= -y * (1 - f(x))
\end{aligned}
$$

$\therefore$ **Option C** is the correct answer

2. For this question let us assume True $= +1$ and False $= -1$. Consider the Multilayer Perceptron Network shown in the figure below with 2 inputs, x_1 and x_2 and 4 perceptrons in the hidden layer. The outputs of these 4 perceptrons are denoted by h_1, h_2, h_3, h_4. Each input is connected to all the 4 perceptrons with specific weights represented by red and blue edges in the figure below. The bias (w_0) of each perceptron is -2 (*i.e.*, each perceptron will fire only if the weighted sum of its input is ≥ 2). Each of these perceptrons is connected to an output perceptron by weights w_1, w_2, w_3 and w_4. The output of this perceptron (y) is the output of the network.
We have to find the weights w_1, w_2, w_3, w_4 such that this network represents the truth table of the $XNOR$ boolean function with two inputs.



red edge indicates $w = -1$
blue edge indicates $w = +1$

Under which of the following conditions will the above network behave as the $XNOR$ boolean function?

A. $w_1 < w_0$, $w_2 \geq w_0$, $w_3 \geq w_0$, $w_4 < w_0$

B. $w_1 = w_0$, $w_2 = w_0$, $w_3 = w_0$, $w_4 = w_0$

C. $w_1 \geq w_0$, $w_2 < w_0$, $w_3 < w_0$, $w_4 \geq w_0$

D. $w_1 \geq w_0$, $w_2 = w_0$, $w_3 = w_0$, $w_4 \geq w_0$

Solution: Each perceptron in the middle layer will fire only for certain inputs and no two perceptrons fire for the same input. For example, output of the first perceptron will be 1 only if the input is $\{-1, -1\}$. We can summarize the activities of the network by the following table.

x_1	x_2	$XNOR$	h_1	h_2	h_3	h_4	$\sum_{i=1}^{4} w_i h_i$
-1	-1	1	1	0	0	0	w_1
-1	+1	0	0	1	0	0	w_2
+1	-1	0	0	0	1	0	w_3
+1	+1	1	0	0	0	1	w_4

This results in the following four conditions to implement $XNOR$:
$w_1 \geq w_0$, $w_2 < w_0$, $w_3 < w_0$, $w_4 \geq w_0$.
$\therefore$ **Option C** is the correct answer

3. The logistic function is defined as follows,

$$f(x) = \frac{1}{1 + e^{-(wx+b)}}$$

where w and b are parameters.
What would happen if w increases?

A. The slope of the logistic function increases
B. The slope of the logistic function decreases
C. The centre point of the logistic function moves to the left
D. The centre point of the logistic function moves to the right

Solution: Option A is the correct answer

4. Keeping in mind the logistic function defined in question 3, what would happen if b increases?

A. The slope of the logistic function increases
B. The slope of the logistic function decreases
C. The centre point of the logistic function moves to the left
D. The centre point of the logistic function moves to the right

Solution: Option C is the correct answer

5. In this question you will implement the Gradient Descent algorithm on a toy 2-D dataset which consists of 40 data points. You can download the dataset from the following URL: `https://drive.google.com/open?id=1am0ZwVt5-a6o8s31gP18bC8y1LZbsvU3`
For this question you have to use the squared error loss function which is given as,

$$loss = \frac{1}{2}(\hat{y} - y)^2$$

where $\hat{y}$ is the output of your model (Refer slide 36 of Lecture 3).
Now given the following hyperparameter settings,

- learning rate = 0.01
- initial weight, w = 1
- initial bias, b = 1
- number of iterations = 100

Which of the following values is the closest to the value of loss that you get at the end of 100 iterations?

A. loss = 0.028

B. loss = 0.0

C. loss = 1.28

D. loss = 0.28

Solution: You can find the value of the loss using the code snippet given below:

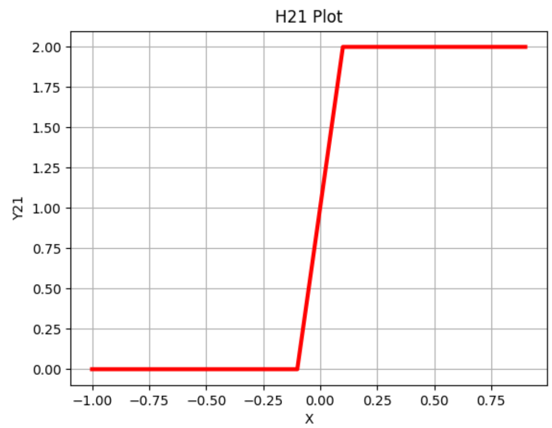
```python
import pandas as pd
import numpy as np

def f(w,b,x):
    return 1.0 / (1.0 + np.exp(-(w*x + b)))

def error(w, b): #Calculate loss/error
    err = 0.0
    for x,y in zip(X,Y) :
        fx = f(w,b,x)
        err += 0.5 * (fx - y) ** 2
    return err

def grad_b(w,b,x,y) :
    fx = f(w,b,x)
    return (fx - y) * fx * (1 - fx)

def grad_w(w,b,x,y) :
    fx = f(w,b,x)
    return (fx - y) * fx * (1 - fx) * x

def do_gradient_descent(X, Y, w, b, eta, max_epochs): # Gradient descent update
    dw = 0
    db = 0
    for i in range(max_epochs) :
        for x,y in zip(X,Y):
            dw += grad_w(w, b, x, y)
            db += grad_b(w, b, x, y)
        w = w - eta * dw
        b = b - eta * db
        print("Epoch {} : Loss = {}".format(i, error(w,b)))
    return w,b

if __name__ == "__main__":
    filename = 'A2_Q4_data.csv'
    df = pd.read_csv(filename) # Loading data
    X = df['X']
    Y = df['Y']
    initial_w = 1
    initial_b = 1
    eta = 0.01
    max_epochs = 100
    w,b = do_gradient_descent(X, Y, initial_w, initial_b, eta, max_epochs)
    error = error(w,b)
    print("error = {}".format(error))
```

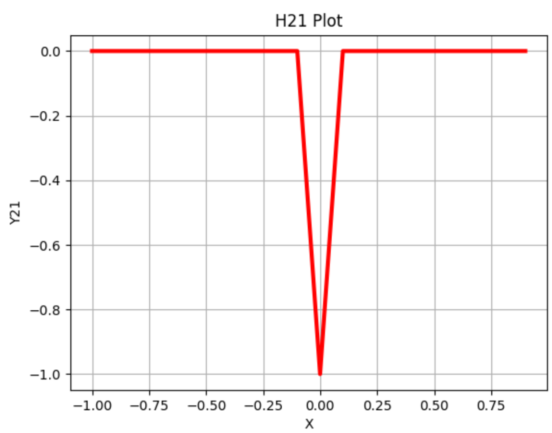
∴ **Option A** is the correct solution.

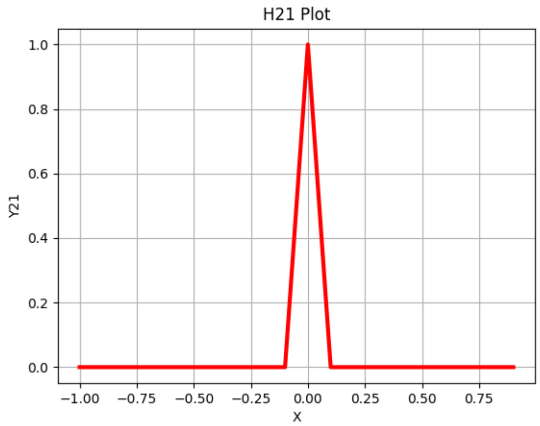
6. Consider the variable x and functions $h_{11}(x)$, $h_{12}(x)$ and $h_{21}(x)$ such that

$$h_{11}(x) = \frac{1}{1 + e^{-(400x+24)}}$$
$$h_{12}(x) = \frac{1}{1 + e^{-(400x-24)}}$$
$$h_{21}(x) = h_{11}(x) - h_{12}(x)$$

Plot the function $h_{21}(x)$ and choose the option which closely matches the shape of this function for $x \in (-1, 1)$

A. 

B. 

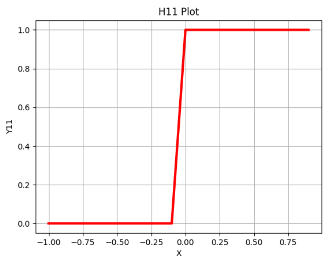
C. 

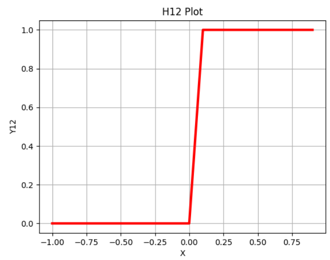
Solution: From the question, it can be seen that the function $h_{21}(x)$ is the subtraction of two functions $h_{11}(x)$ and $h_{12}(x)$. We can plot the functions using the following code :

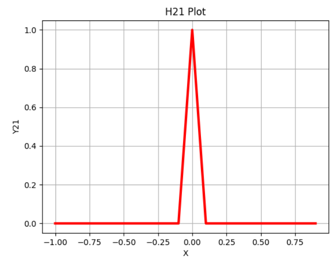
```
import numpy as np
import matplotlib.pyplot as plt

X = np.arange(-1,1,0.1)

#Plotting h_11, h_12 and h_21 as three separate figures.
Y11 = 1/(1+np.exp(-400*X - 24))
fig = plt.figure()
plt.title('H11 Plot')
plt.plot(X,Y11,linewidth = 3,color = 'r')
plt.xlabel('X')
plt.ylabel('Y11')
plt.grid()
plt.show()

Y12 = 1/(1+np.exp(-400*X + 24))
fig = plt.figure()
plt.title('H12 Plot')
plt.plot(X,Y12,linewidth = 3,color = 'r')
plt.xlabel('X')
plt.ylabel('Y12')
plt.grid()
plt.show()

Y21 = Y12 - Y11
fig = plt.figure()
plt.title('H21 Plot')
plt.plot(X,Y21,linewidth = 3,color = 'r')
plt.xlabel('X')
plt.ylabel('Y21')
plt.grid()
```

```
plt.show()
```

On plotting, the function $h_{11}(x)$, $h_{12}(x)$ and $h_{21}(x)$ looks like the following :





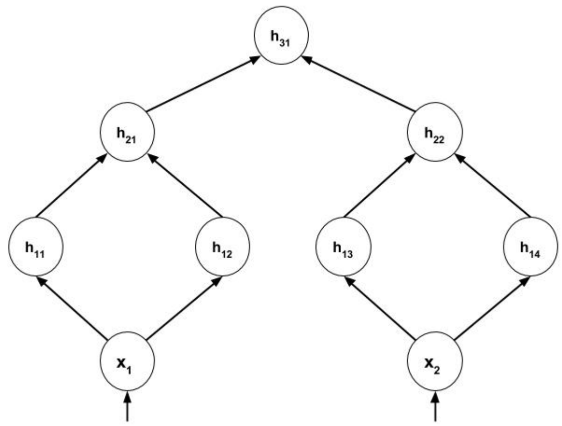


Th function $h_{21}(x)$ is the subtraction of two sigmoids and the result is a 2d tower shaped curve.
$\therefore$ **Option C** is the correct answer.

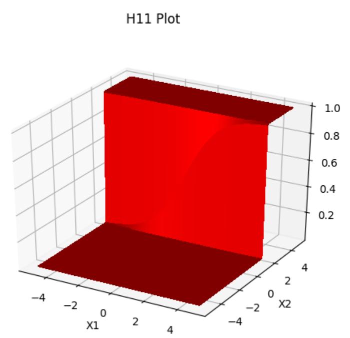
7. Now consider the variables x_1, x_2 and the following functions :

$$\begin{aligned}
h_{11}(x_1, x_2) &= \frac{1}{1 + e^{-(x_1 + 100x_2 + 200)}} \\
h_{12}(x_1, x_2) &= \frac{1}{1 + e^{-(x_1 + 100x_2 - 200)}} \\
h_{13}(x_1, x_2) &= \frac{1}{1 + e^{-(100x_1 + x_2 + 200)}} \\
h_{14}(x_1, x_2) &= \frac{1}{1 + e^{-(100x_1 + x_2 - 200)}} \\
h_{21}(x_1, x_2) &= h_{11}(x_1, x_2) - h_{12}(x_1, x_2) \\
h_{22}(x_1, x_2) &= h_{13}(x_1, x_2) - h_{14}(x_1, x_2) \\
h_{31}(x_1, x_2) &= h_{21}(x_1, x_2) + h_{22}(x_1, x_2) \\
f(x_1, x_2) &= \frac{1}{1 + e^{-(50h_{31}(x) - 100)}}
\end{aligned}$$

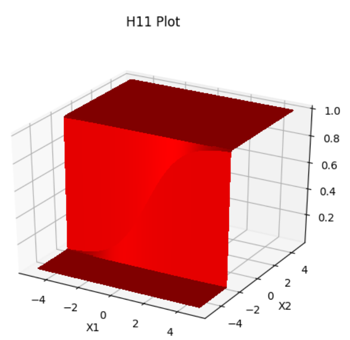
The above set of functions are summarized in the graph below.



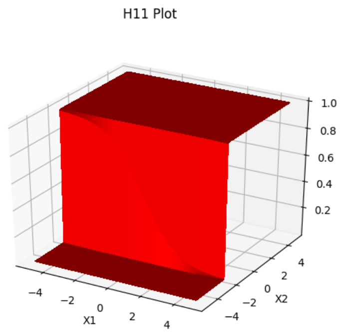
Plot the function $h_{11}(x_1, x_2)$ and choose the option which closely matches the shape of this function for $x \in (-5, 5)$



A.



B.

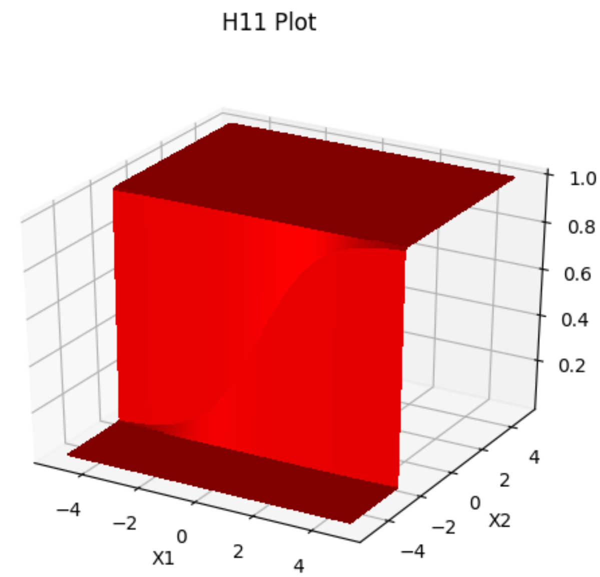


C.

Solution: We can plot the function using the following code :

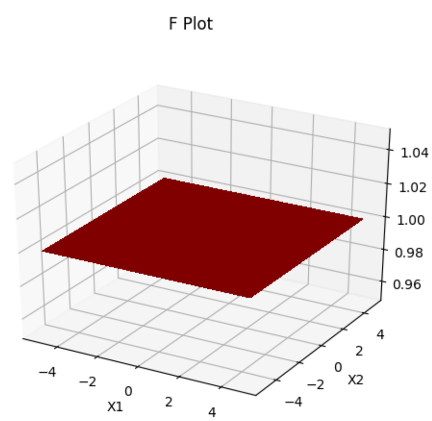
```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D # for 3d plotting

X1 = np.arange(-5,5,0.1)
X2 = np.arange(-5,5,0.1)
X,Y = np.meshgrid(X1,X2)
Z11 = 1/(1+np.exp(-X - 100*Y - 200))

fig = plt.figure()
plt.suptitle('H11 Plot')
ax = fig.add_subplot(1,1,1, projection='3d')
ax.plot_surface(X,Y,Z11,rstride = 1,cstride = 1,color = 'r',antialiased
    = False)
plt.xlabel('X1')
plt.ylabel('X2')
plt.show()
```

The function $h_{11}(x_1, x_2)$ is the 3d sigmoid function which can be seen below.

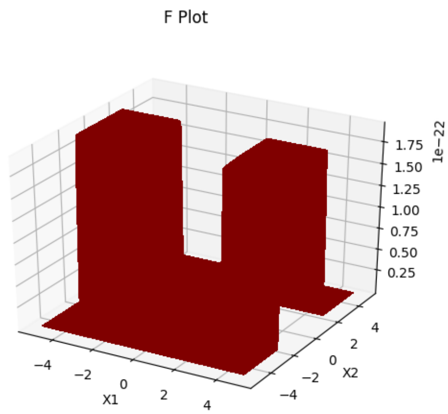


∴ **Option B** is the correct option.

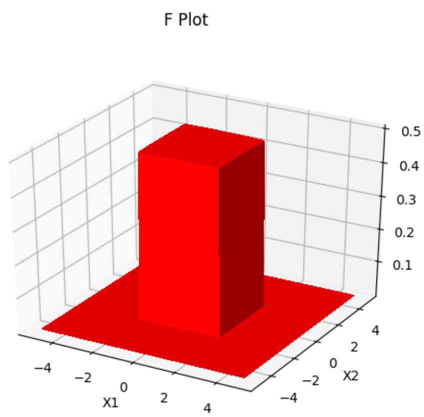
8. Plot the function $f(x_1, x_2)$ as defined in question 7 and choose the option which closely matches the shape of this function for $x \in (-5, 5)$.



A.

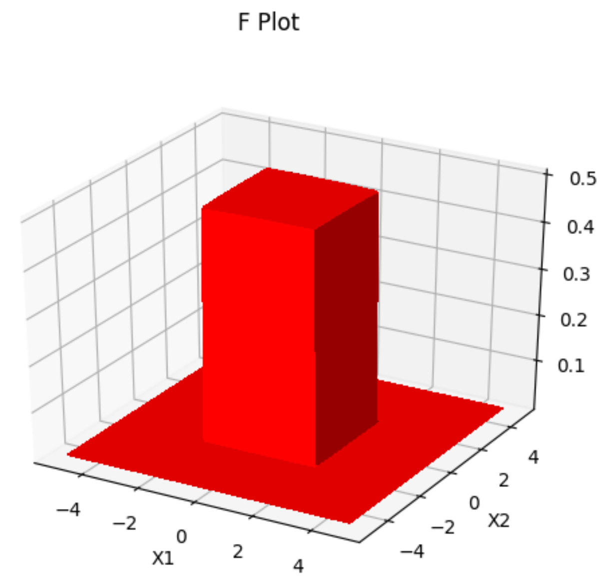


B.



C.

Solution: Using similar code snippet given in question 7 and by following the graph structure given in the question, we can plot the function $f(x_1, x_2)$. The output is the 3d tower function which can be seen below.



$\therefore$ **Option C** is the correct option.

9. Based on the plot, what is the maximum value of the function $f(x_1, x_2)$?

   A. 0.4

   B. 0.5

   C. 1

Solution: On plotting the function $f(x_1, x_2)$, we can see that the maximum value of the function is 0.5.
$\therefore$ **Option B** is the correct option.

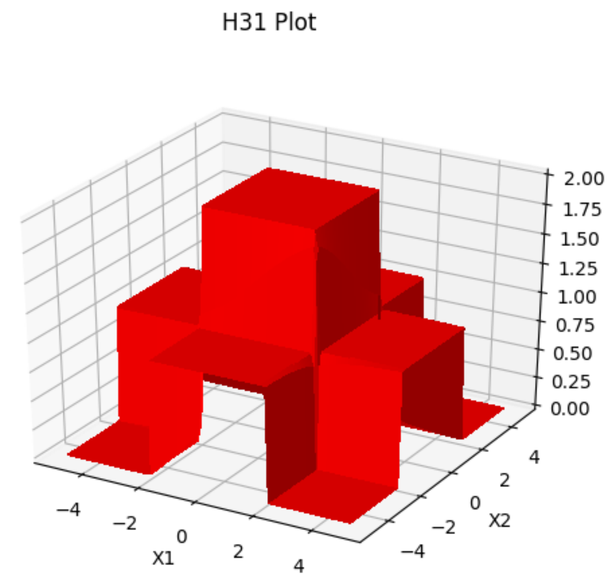
10. What is the maximum value of the function $h_{31}(x_1, x_2)$?

    A. 2

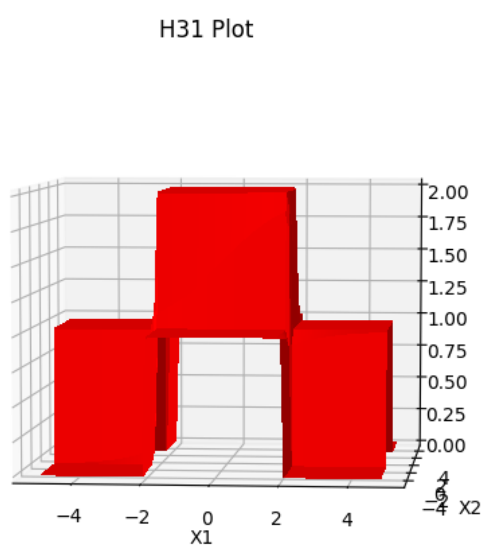
    B. 1.5

    C. 1

Solution:



Using similar code snippet given in part(a) and by following the graph structure given in the question, we can plot the function $f(x_1, x_2)$. On plotting the function $h_{31}(x_1, x_2)$ and rotating the plot conveniently, we obtain the following graph,



where we can see that the maximum value of the function is 2.
$\therefore$ **Option A** is the correct option.