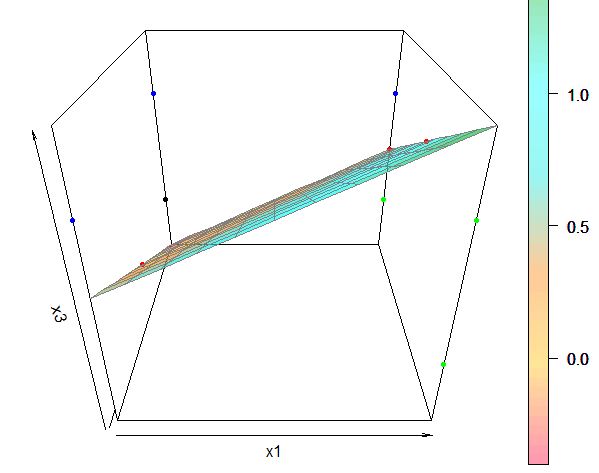
**Problem 1: Perceptron Learning**

In the class, we discussed about perceptron with step activation function. Given the following dataset,do the following tasks:



 Is the above data linearly separable?

[Ans:] The 3-D plot of the above data is as follows:



The +1 points (1,0,0) (1,1,0) (1,0,1) are marked in green.

The -1 points (0,1,1) (1,1,1) (0,0,1) are marked in blue.

We need to predict the class/color of the point (1,1,0) marked in **black**.

We clearly see a plane dividing the green and blue points. So, the data is linearly separable. Also, we expect the perceptron to predict the point (1,1,0) as blue colored or having the output -1.

*Note:* The plane plotted above is defined by the red colored points (1,0.5,1) (1,1,0.5) (0,0.5,0) and has the equation x1 – x2 – x3 = -0.5. There are many other planes that separate the +1 and -1 points. But all such planes will have the **black** point (1,1,0) on the same side as that of the blue points.

 Perform perceptron learning by hand for the above dataset. Show the weights after each epoch. Assume all the weights as 0 (including bias) and learning rate as 1.

**[Ans:]** For perceptron learning, we always assume x0 = 1. Let our initial random weights (wo,w1,w2,w3) be (0,0,0,0). We also assume ‘yeta’ as 1.

We refine the weights using the formula, w = w + (y – output)\*(x)

Activation function f(x > 0) = +1; f(x <= 0) = -1.

We calculate (a,b,c,d)\*(e,f,g,h) = ae + bf + cg + dh.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Row ID | X0 | X1 | X2 | X3 | Output |
| a | 1 | 1 | 0 | 0 | 1 |
| b | 1 | 0 | 1 | 1 | -1 |
| c | 1 | 1 | 1 | 0 | 1 |
| d | 1 | 1 | 1 | 1 | -1 |
| e | 1 | 0 | 0 | 1 | -1 |
| f | 1 | 1 | 0 | 1 | 1 |

**Epoch – 1**

1. (0,0,0,0)\*(1,1,0,0) = 0. f(0) = -1 does not match the output 1. So, refine weight using w = (0,0,0,0) + [1-(-1)]\*(1,1,0,0) = (2,2,0,0)
2. (2,2,0,0)\*(1,0,1,1) = 2. f(2) = 1 does not match output -1.

w = (2,2,0,0) + (-1-1)\*(1,0,1,1) = (0,2,-2,-2)

1. (0,2,-2,-2)\*(1,1,1,0) = 0. f(0) = -1 does not match output 1.

w = (0,2,-2,-2) + [1-(-1)]\*(1,1,1,0) = (2,4,0,-2)

1. (2,4,0,-2)\*(1,1,1,1) = 4. f(4) = 1 does not match output -1.

w = (2,4,0,-2) + (-1-1)\*(1,1,1,1) = (0,2,-2,-4)

1. (0,2,-2,-4)\*(1,0,0,1) = -4. f(-4) = -1 matches the output. So, weight does not need to be refined.
2. (0,2,-2,-4)\*(1,1,0,1) = -2. f(-2) = -1 does not match output 1.

w = (0,2,-2,-4) + [1-(-1)]\*(1,1,0,1) = (2,4,-2,-2)

Weight after epoch 1 is (2,4,-2,-2)

**Epoch – 2**

1. (2,4,-2,-2)\*(1,1,0,0) = 6. f(6) = 1 matches the output. So, weight does not need to be refined.
2. (2,4,-2,-2)\*(1,0,1,1) = -2. f(-2) = -1 matches the output.
3. (2,4,-2,-2)\*(1,1,1,0) = 4. f(4) = 1 matches the output.
4. (2,4,-2,-2)\*(1,1,1,1) = 2. f(2) = 1 does not match output -1.

w = (2,4,-2,-2) + (-1-1)\*(1,1,1,1) = (0,2,-4,-4)

1. (0,2,-4,-4)\*(1,0,0,1) = -4. f(-4) = -1 matches the output.
2. (0,2,-4,-4)\*(1,1,0,1) = -2. f(-2) = -1 does not match output 1.

w = (0,2,-4,-4) + [1-(-1)]\*(1,1,0,1) = (2,4,-4,-2)

Weight after epoch 2 is (2,4,-4,-2)

**Epoch – 3**

1. (2,4,-4,-2)\*(1,1,0,0) = 6. f(6) = 1 matches the output.
2. (2,4,-4,-2)\*(1,0,1,1) = -4. f(-4) = -1 matches the output.
3. (2,4,-4,-2)\*(1,1,1,0) = 2. f(2) = 1 matches the output.
4. (2,4,-4,-2)\*(1,1,1,1) = 2. f(0) = -1 matches the output.
5. (2,4,-4,-2)\*(1,0,0,1) = 0. f(0) = -1 matches the output.
6. (2,4,-4,-2)\*(1,1,0,1) = 4. f(4) = 1 matches the output.

Weight after epoch 3 remains unchanged at (2,4,-4,-2). These final weights constitute the learning from the perceptron model.

 How many epochs are required?

**[Ans:]** We stop after 3 epochs since the weights at the beginning of epoch 3 classified all the inputs correctly and thus remained unchanged throughout the epoch.

 What would the prediction by the learned perceptron for x1=0, x2=1, x3=0?

[Ans:] For (0,1,0), we have (2,4,-4,-2)\*(1,0,1,0) = -2. f(-2) = -1.

We see that the perceptron prediction matches our intuition/expectation.

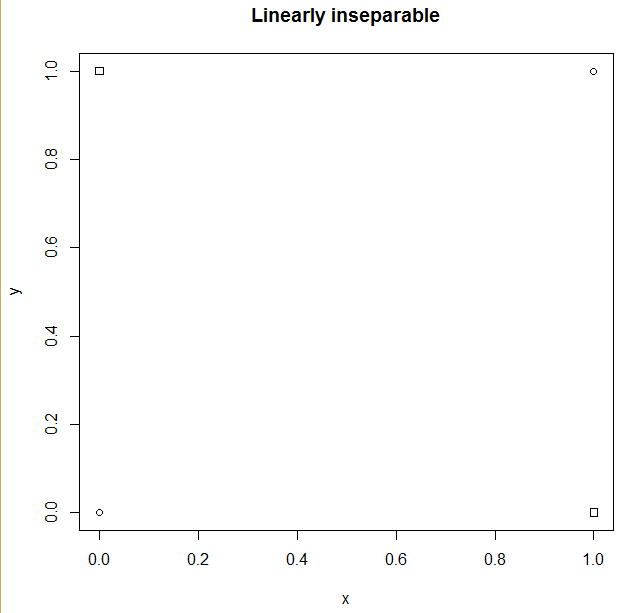
**Problem 2: Perceptron Learning**

In the class, we discussed about perceptron with step activation function. Given the following dataset, do the following tasks:



 Is the above data linearly separable?

[Ans:] In the plot below, we see the -1 points (0,0) (1,1) denoted by a circle. The +1 points (1,0) (1,1) are denoted by a square. Obviously, there is no line that can separate the +1 and -1 points, one on each side. So, we say that the data is NOT linearly separable. We thus do not expect the perceptron to converge on a particular weight.



 Perform perceptron learning by hand for the above dataset. Show the weights after each epoch. Assume all the weights as 0 (including bias) and learning rate as 1.

**[Ans:]** For perceptron learning, we always assume x0 = 1. Let our initial random weights (wo,w1,w2) be (0,0,0,0). We also assume the learning rate ‘yeta’ as 1.

We refine the weights using the formula, w = w + (y – output)\*(x)

Activation function f(x > 0) = +1; f(x <= 0) = -1.

We calculate (a,b,c)\*(d,e,f) = ad + be + cf.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Row ID | X0 | X1 | X2 | Output |
| A | 1 | 0 | 0 | -1 |
| B | 1 | 0 | 1 | 1 |
| C | 1 | 1 | 0 | 1 |
| D | 1 | 1 | 1 | -1 |

**Epoch – 1**

1. (0,0,0)\*(1,0,0) = 0. f(0) = -1 matches the output. So, weight does not need to be refined.
2. (0,0,0)\*(1,0,1) = 0. f(0) = -1 does not match output 1.

w = (0,0,0) + [1-(-1)]\*(1,0,1) = (2,0,2)

1. (2,0,2)\*(1,1,0) = 2. f(2) = 1 matches the output.
2. (2,0,2)\*(1,1,1) = 4. f(4) = 1 does not match output -1.

w = (2,0,2) + (-1-1)\*(1,1,1) = (0,-2,0)

Weight after epoch 1 is (0,-2,0)

**Epoch – 2**

1. (0,-2,0)\*(1,0,0) = 0. f(0) = -1 matches the output.
2. (0,-2,0)\*(1,0,1) = 0. f(0) = -1 does not match output 1.

w = (0,-2,0) + [1-(-1)]\*(1,0,1) = (2,-2,2)

1. (2,-2,2)\*(1,1,0) = 0. f(0) = -1 does not match the output.

w = (2,-2,2) + [1-(-1)]\*(1,1,0) = (4,0,2)

1. (4,0,2)\*(1,1,1) = 6. f(6) = 1 does not match output -1.

w = (4,0,2) + (-1-1)\*(1,1,1) = (2,-2,0)

Weight after epoch 2 is (2,-2,0)

**Epoch – 3**

1. (2,-2,0)\*(1,0,0) = 2. f(2) = 1 does not match the output.

w = (2,-2,0) + (-1-1)\*(1,0,0) = (0,-2,0)

1. (0,-2,0)\*(1,0,1) = 0. f(0) = -1 does not match output 1.

w = (0,-2,0) + [1-(-1)]\*(1,0,1) = (2,-2,2)

1. (2,-2,2)\*(1,1,0) = 0. f(0) = -1 does not match the output.

w = (2,-2,2) + [1-(-1)]\*(1,1,0) = (4,0,2)

1. (4,0,2)\*(1,1,1) = 6. f(6) = 1 does not match output -1.

w = (4,0,2) + (-1-1)\*(1,1,1) = (2,-2,0)

Weight after epoch 2 is (2,-2,0)

Weight after epoch 3 remains unchanged at (2,-2, 0). Besides, this weight does not classify all the inputs correctly.

 How many epochs are required?

**[Ans:]** There is no convergence of weights and the same incorrect weights get repeated epoch after epoch. So, any number of epochs will not be sufficient

 Do you see perceptron converging or not?

**[Ans:]** The perceptron does not converge. This implies that the input does not have a linear binary classification. This also matches our initial expectation.