```python
import numpy as np
```

[1] ✓ 0.0s Python

```python
n = 2
m = 4
data_x = [
    [1,2,2],
    [1,-2,2],
    [1,1,-2],
    [1,-1, 0]]
data_y = [-1, -1, 1, 1]

data_x = np.array(data_x)
data_y = np.array(data_y)
```

[2] ✓ 0.0s Python

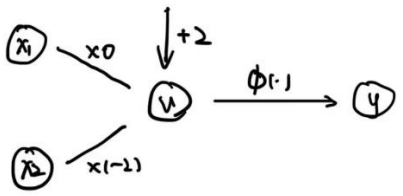
```python
def SGN(x):
    if x > 0:
        return 1
    else:
        return -1
```

[3] ✓ 0.0s Python

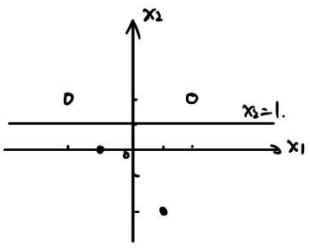
```python
w = [0, 0, 0]
w = np.array(w)
result = []

for e in range(2):
    print("epoch:", e)
    for i in range(m):
        k = e*n + i
        # print("k:", k)
        x = data_x[i]
        y = data_y[i]
        wx = np.dot(w, x)
        # print("w^T * x:", wx)
        y_est = SGN(wx)
        # print("y_estimate:", y_est)
        if y_est * y <= 0:
            w = np.add(w, y*x)
        # print("w:", w)
        # print()

        print(f"k: {k:2d}, w^T * x: {wx:5.2f}, y: {data_y[i]:2d}, y_estimate: {y_est:2d}, It is {data_y[i]==y_est},  w_k+1: {w}")
```

[4] ✓ 0.0s Python

```
epoch: 0
k:  0, w^T * x:  0.00, y: -1, y_estimate: -1, It is True,  w_k+1: [0 0 0]
k:  1, w^T * x:  0.00, y: -1, y_estimate: -1, It is True,  w_k+1: [0 0 0]
k:  2, w^T * x:  0.00, y:  1, y_estimate: -1, It is False,  w_k+1: [ 1  1 -2]
k:  3, w^T * x:  0.00, y:  1, y_estimate: -1, It is False,  w_k+1: [ 2  0 -2]
epoch: 1
k:  2, w^T * x: -2.00, y: -1, y_estimate: -1, It is True,  w_k+1: [ 2  0 -2]
k:  3, w^T * x: -2.00, y: -1, y_estimate: -1, It is True,  w_k+1: [ 2  0 -2]
k:  4, w^T * x:  6.00, y:  1, y_estimate:  1, It is True,  w_k+1: [ 2  0 -2]
k:  5, w^T * x:  2.00, y:  1, y_estimate:  1, It is True,  w_k+1: [ 2  0 -2]
```



Output.



$-2x_2 + 2 = 0$

$x_2 = 1.$