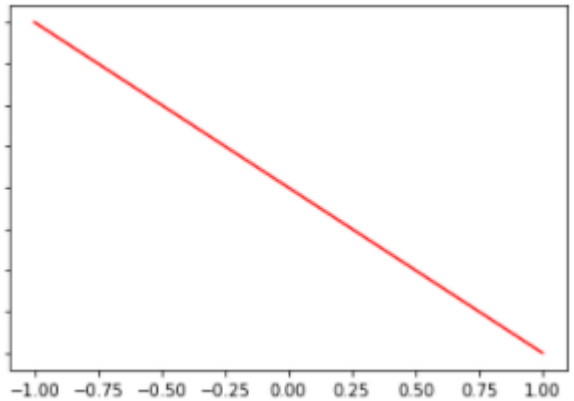
# Hebb for AND and OR

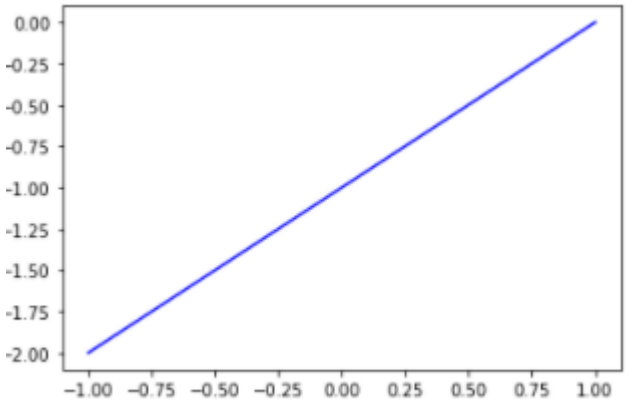
AND

```
delte wazn ha [1, 1, 1]
wazn updaate shode ===== [1, 1, 1]
moadele khat:
1 + 1*x1 + 1*y =0


delte wazn ha [-1, 1, -1]
wazn updaate shode ===== [0, 2, 0]
moadele khat:
0 + 0*x1 + 2*y =0


delte wazn ha [1, -1, -1]
wazn updaate shode ===== [1, 1, -1]
moadele khat:
-1 + 1*x1 + 1*y =0


delte wazn ha [1, 1, -1]
wazn updaate shode ===== [2, 2, -2]
moadele khat:
-2 + 2*x1 + 2*y =0
```




OR

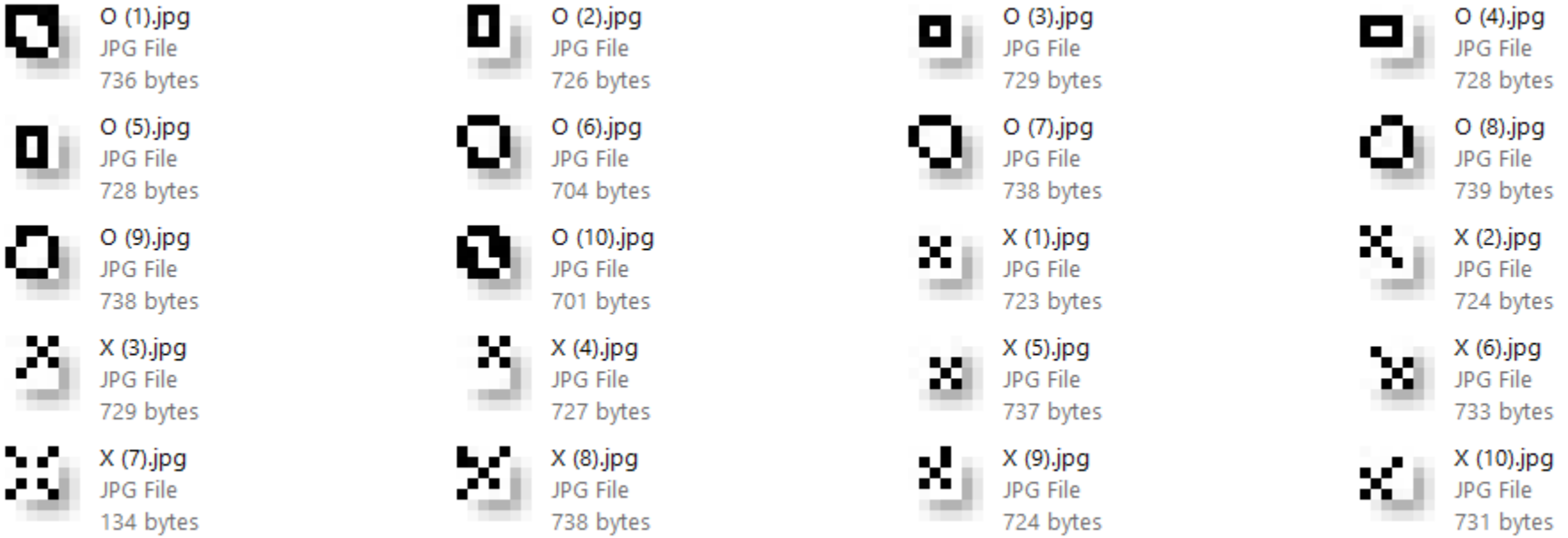
```
delte wazn ha [1, 1, 1]
updaated weight ===== [1, 1, 1]
moadele khat:
1 + 1*x1 + 1*y =0


delte wazn ha [1, -1, 1]
updaated weight ===== [2, 0, 2]
moadele khat:
2 + 2*x1 + 0*y =0


delte wazn ha [-1, 1, 1]
updaated weight ===== [1, 1, 3]
moadele khat:
3 + 1*x1 + 1*y =0


delte wazn ha [1, 1, -1]
updaated weight ===== [2,
moadele khat:
2 + 2*x1 + 2*y =0
```



# Train weight


```
newWeight [0.7, -0.7, -0.7, -0.7, 0.7, -0.7, 0.7, 0.7, 0.7, -0.7, -0.7, -0.7, 0.7, -0.7, -0.7, -0.7, 0.7, 0.7, 0.7, -0.7, 0.7, -0.7, -0.7, -0.7, 0.7]
(yhads zade shode  ,  y asli)
 [(-1, -1.0), (-1, -1.0), (-1, -1.0), (-1, -1.0), (-1, -1.0), (-1, -1.0), (-1, -1.0), (-1, -1.0), (-1, -1.0), (-1, -1.0), (-1, -1.0), (1, 1.0), (1, 1.0), (1,
1.0), (1, 1.0), (1, 1.0), (1, 1.0), (1, 1.0), (1, 1.0), (1, 1.0), (1, 1.0), (1, 1.0), (1, 1.0), (1, 1.0), (1, 1.0)]
accuracy:  1.0
```

# Test final weight

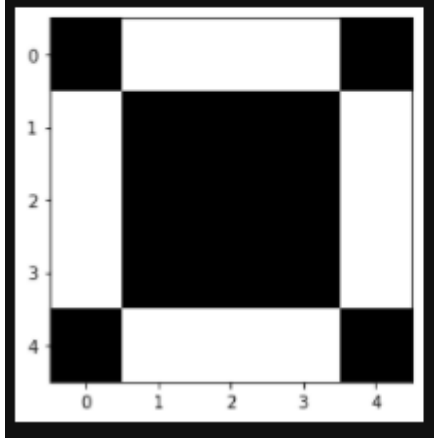



```
1 yt2=list_predict(testx,testy,wazn_nahayi)
2 score(testy,yt2)
3
```

```
(yhads zade shode  ,  y asli)
 [(-1, -1.0), (-1, -1.0), (1, -1.0), (1, -1.0), (1, -1.0), (1, -1.0), (-1, -1.0), (-1, -1.0), (-1, -1.0), (-1, -1.0), (1, -1.0), (1, 1.0), (1, 1.0), (1, 1.
0), (1, 1.0), (1, 1.0), (1, 1.0), (1, 1.0), (1, 1.0), (1, 1.0)]
accuracy:  0.75
0.75
```

```
1 jj=dst[5]
2 jj=np.reshape(jj,(5,5))
3 plt.imshow(jj,cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x1949439eb20>
```




```
1 jj=dst[15]
2 jj=np.reshape(jj,(5,5))
3 plt.imshow(jj,cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x1949443eeb0>
```