

Hebb for AND and OR

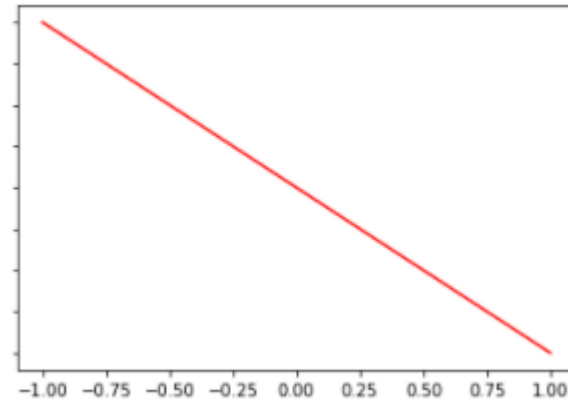
AND

```
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
```



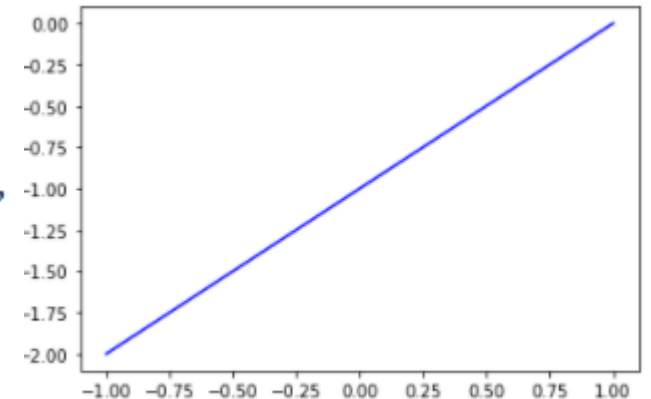
OR

```
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, 2, -2]
moadele khat:
2 + 2*x1 + 2*y =0
```



Train weight

[illegible]

Test final weight



O (1).jpg
JPG File
736 bytes



O (2).jpg
JPG File
726 bytes



O (3).jpg
JPG File
729 bytes



O (4).jpg
JPG File
728 bytes



O (5).jpg
JPG File
728 bytes



O (6).jpg
JPG File
704 bytes



O (7).jpg
JPG File
738 bytes



O (8).jpg
JPG File
739 bytes



O (9).jpg
JPG File
738 bytes



O (10).jpg
JPG File
701 bytes



X (1).jpg
JPG File
723 bytes



X (2).jpg
JPG File
724 bytes



X (3).jpg
JPG File
729 bytes



X (4).jpg
JPG File
727 bytes



X (5).jpg
JPG File
737 bytes



X (6).jpg
JPG File
733 bytes



X (7).jpg
JPG File
134 bytes



X (8).jpg
JPG File
738 bytes



X (9).jpg
JPG File
724 bytes



X (10).jpg
JPG File
731 bytes

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

(yhads zade shode , y asli)

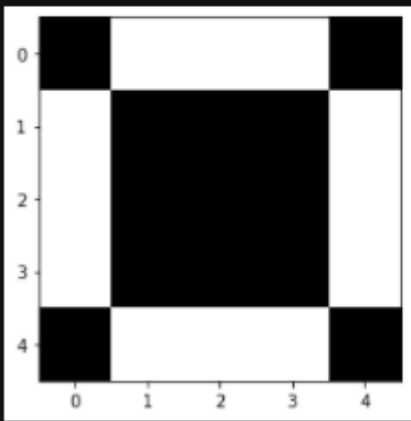
[(-1, -1.0), (-1, -1.0), (1, -1.0), (1, -1.0), (1, -1.0), (1, -1.0), (-1, -1.0), (-1, -1.0), (-1, -1.0), (-1, -1.0), (1, -1.0), (1, 1.0), (1, 1.0), (1, 1.0), (1, 1.0), (1, 1.0), (1, 1.0), (1, 1.0), (1, 1.0)]

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>

