2D Convolution (1)

I_1(x,y)

100	100	0	0
100	100	0	0
100	100	0	0
100	100	0	0

k(x,y)

-1	0	1
-1	0	1
-1	0	1

O_1(x,y)

0	-100	-100	0
0	-100	-100	0
0	-100	-100	0
0	-100	-100	0

I_2(x,y)

100	100	100	0
100	100	0	0
100	100	0	0
100	100	0	0

k(x,y)

-1	0	1
-1	0	1
-1	0	1

O_2(x,y)

0	0	-100	-100
0	-100	-100	0
0	-100	-100	0
0	-100	-100	0

2D Convolution (2)

I_3(x,y)

100	100	100	0
100	100	100	0
100	100	0	0
100	100	0	0

k(x,y)

-1	0	1
-1	0	1
-1	0	1

O_3(x,y)

0	0	-100	-100
0	0	-100	-100
0	-100	-100	0
0	-100	-100	0

I_4(x,y)

100	100	100	0
100	100	100	0
100	100	100	0
100	100	0	0

k(x,y)

-1	0	1
-1	0	1
-1	0	1

O_4(x,y)

0	0	-100	-100
0	0	-100	-100
0	0	-100	-100
0	-100	-100	0

2D Convolution (3)

I_5(x,y)

100	100	100	0
100	100	100	0
100	100	100	0
100	100	100	0

k(x,y)

-1	0	1
-1	0	1
-1	0	1

O_5(x,y)

0	0	-100	-100
0	0	-100	-100
0	0	-100	-100
0	0	-100	-100

I_6(x,y)

0	100	100	0
100	100	100	0
100	100	100	0
100	100	100	0

k(x,y)

-1	0	1
-1	0	1
-1	0	1

O_6(x,y)

100	100	-100	-100
0	0	-100	-100
0	0	-100	-100
0	0	-100	-100

2D Convolution (4)

x

y

I_7(x,y)

0	100	100	0
0	100	100	0
100	100	100	0
100	100	100	0

k(x,y)

-1	0	1
-1	0	1
-1	0	1

x

y

O_7(x,y)

100	100	-100	-100
100	100	-100	-100
0	0	-100	-100
0	0	-100	-100

x

y

I_8(x,y)

0	100	100	0
0	100	100	0
100	100	100	0
100	100	100	0

k(x,y)

-1	0	1
-1	0	1
-1	0	1

x

y

O_8(x,y)

100	100	-100	-100
100	100	-100	-100
0	0	-100	-100
0	0	-100	-100

2D Convolution (5)

x

y

I_9(x,y)

0	100	100	0
0	100	100	0
0	100	100	0
100	100	100	0

k(x,y)

-1	0	1
-1	0	1
-1	0	1

x

y

O_9(x,y)

100	100	-100	-100
100	100	-100	-100
100	100	-100	-100
0	0	-100	-100

x

y

I_10(x,y)

0	100	100	0
0	100	100	0
0	100	100	0
0	100	100	0

k(x,y)

-1	0	1
-1	0	1
-1	0	1

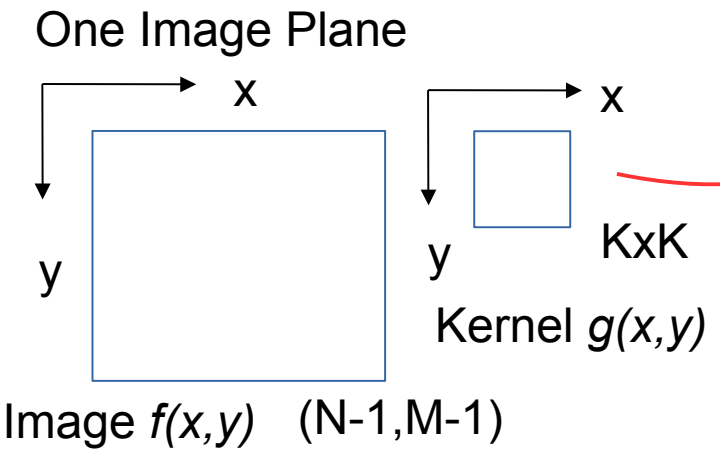
x

y

O_10(x,y)

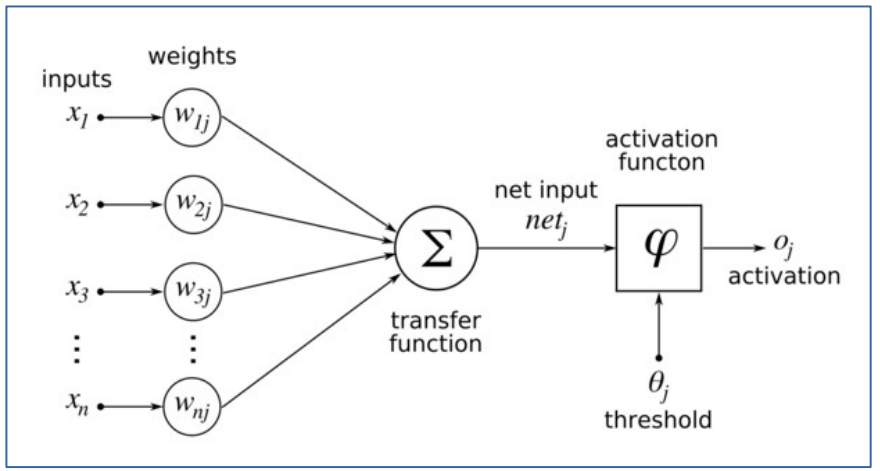
100	100	-100	-100
100	100	-100	-100
100	100	-100	-100
100	100	-100	-100

Kernel Coefficients to Neural Nets



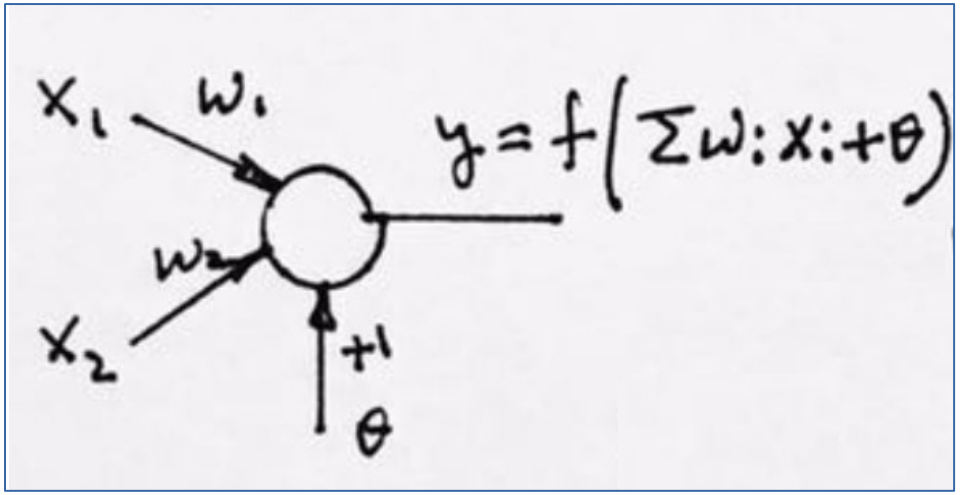
w_11	w_12	w_13
w_21	w_22	w_23
w_31	w_32	w_33

N layers



Input from image I(x,y)	weight
x_11	w_11
x_12	w_12
x_13	w_13
x_21	w_21
x_22	w_22
x_23	w_23
x_31	w_31
x_32	w_32
x_33	w_33

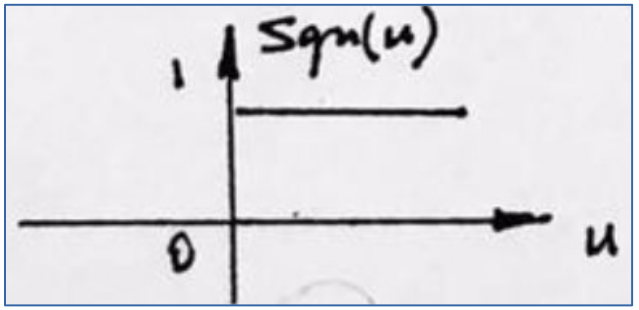
Map to Single Layer NN (1)



$$y = f(\Sigma w_i x_i + \theta) \quad \ldots (1)$$

Where

$$y = f(\cdot) \text{ as } Sgn(\cdot) \quad \ldots (2)$$



Group 1, C1, as those from edge pixels

Group 2, C2, as those from non-edge pixels

Hence, from the following, we have

100	100	0	0
100	100	0	0
100	100	0	0
100	100	0	0

0	-100	-100	0
0	-100	-100	0
0	-100	-100	0
0	-100	-100	0

(1,1), y = 0; (1,2), y=1; (1,3), y=1; (1,4), y=0

(2,1), y = 0; (2,2), y=1; (2,3), y=1; (2,4), y=0

(3,1), y = 0; (3,2), y=1; (3,3), y=1; (3,4), y=0

(4,1), y = 0; (4,2), y=1; (4,3), y=1; (4,4), y=0

Continue till the las pair as

Map to Single Layer NN (2)

From the following,

0	100	100	0
0	100	100	0
0	100	100	0
0	100	100	0

100	100	-100	-100
100	100	-100	-100
100	100	-100	-100
100	100	-100	-100

(1,1), y = 1; (1,2), y=1; (1,3), y=1; (1,4), y=1

(2,1), y = 1; (2,2), y=1; (2,3), y=1; (2,4), y=1

(3,1), y = 1; (3,2), y=1; (3,3), y=1; (3,4), y=1

(4,1), y = 1; (4,2), y=1; (4,3), y=1; (4,4), y=1

The training algorithm:

$$w_i^+ = w_i^- + \eta(d-y)x_i$$

where

η is gain.

d : desired the output.

y: actual output.

Map to Single Layer NN (2)

From the following,

0	100	100	0
0	100	100	0
0	100	100	0
0	100	100	0

100	100	-100	-100
100	100	-100	-100
100	100	-100	-100
100	100	-100	-100

(1,1), y = 1; (1,2), y=1; (1,3), y=1; (1,4), y=1

(2,1), y = 1; (2,2), y=1; (2,3), y=1; (2,4), y=1

(3,1), y = 1; (3,2), y=1; (3,3), y=1; (3,4), y=1

(4,1), y = 1; (4,2), y=1; (4,3), y=1; (4,4), y=1

The training algorithm:

$$w_i^+ = w_i^- + \eta(d-y)x_i$$

where

η is gain.
d : desired the output.
y: actual output.

So, we have

Input from image I(x,y)	weight	Output image O(x,y)
x_11	w_11	y_11
x_12	w_12	y_12
x_13	w_13	y_13
x_21	w_21	y_21
x_22	w_22	y_22
x_23	w_23	y_23
x_31	w_31	y_31
x_32	w_32	y_32
x_33	w_33	y_33

Now, train the NN find weights w_ij