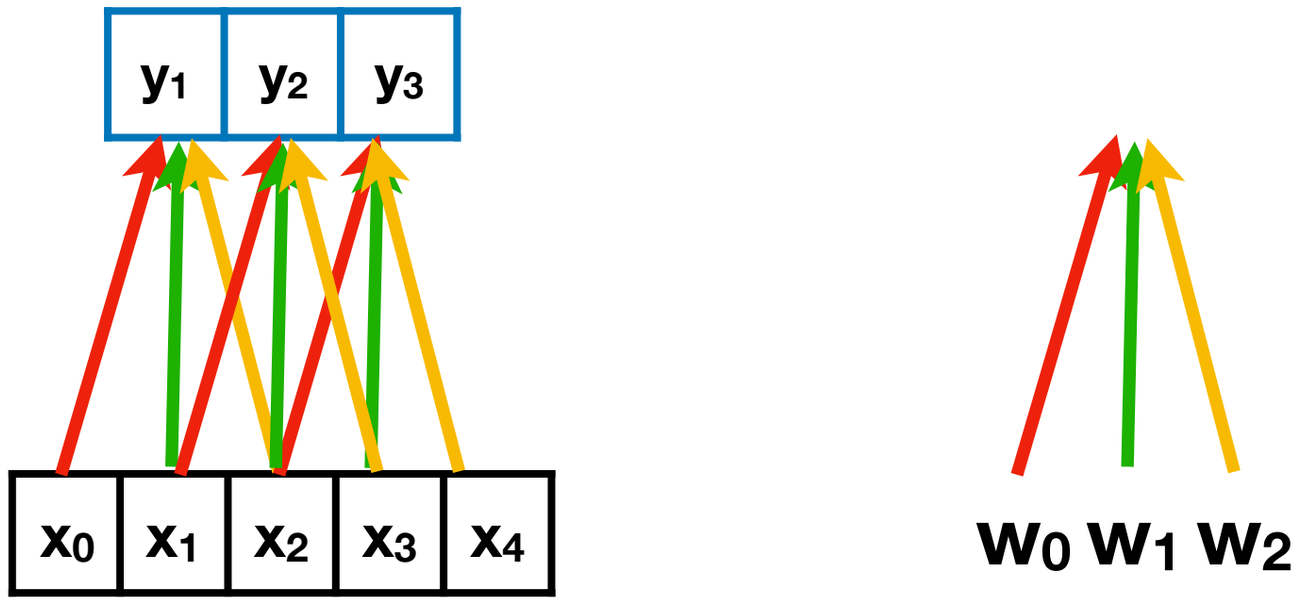
Forward and Backward Pass of Convolutional layers

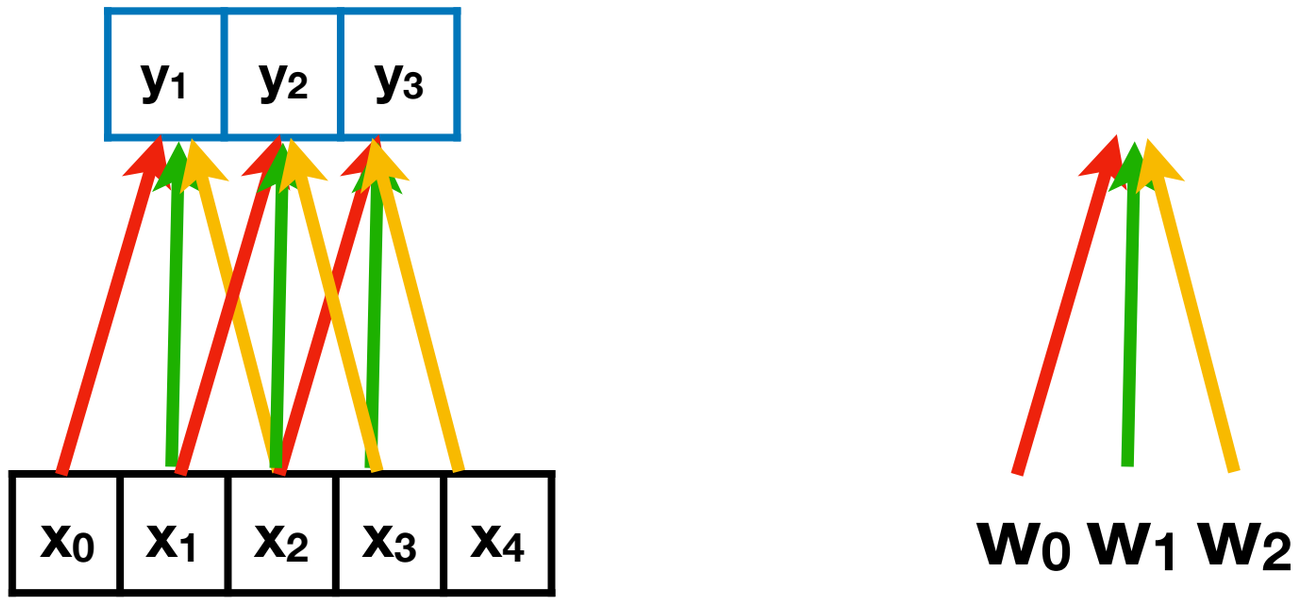
Jong-Chyi Su, 3/2/2018

Forward pass - 1D case



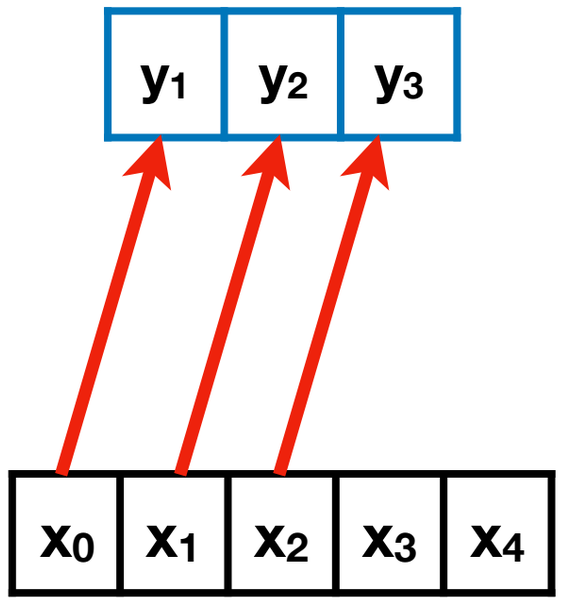
$$y_1 = w_0 * x_0 + w_1 * x_1 + w_2 * x_2 + b$$
$$y_2 = w_0 * x_1 + w_1 * x_2 + w_2 * x_3 + b$$
$$y_3 = w_0 * x_2 + w_1 * x_3 + w_2 * x_4 + b$$

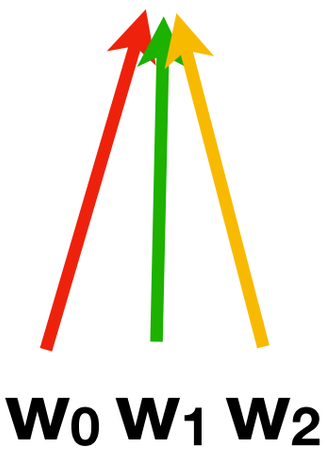
Backward pass - 1D case



How to compute dw_0?

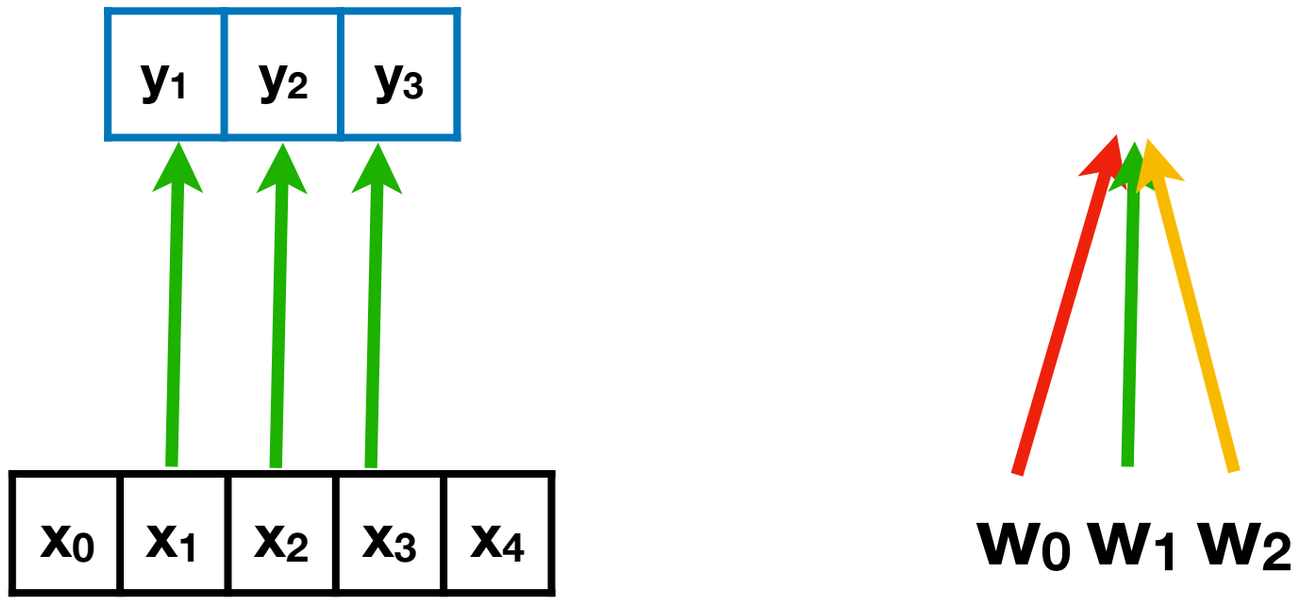
Backward pass - 1D case





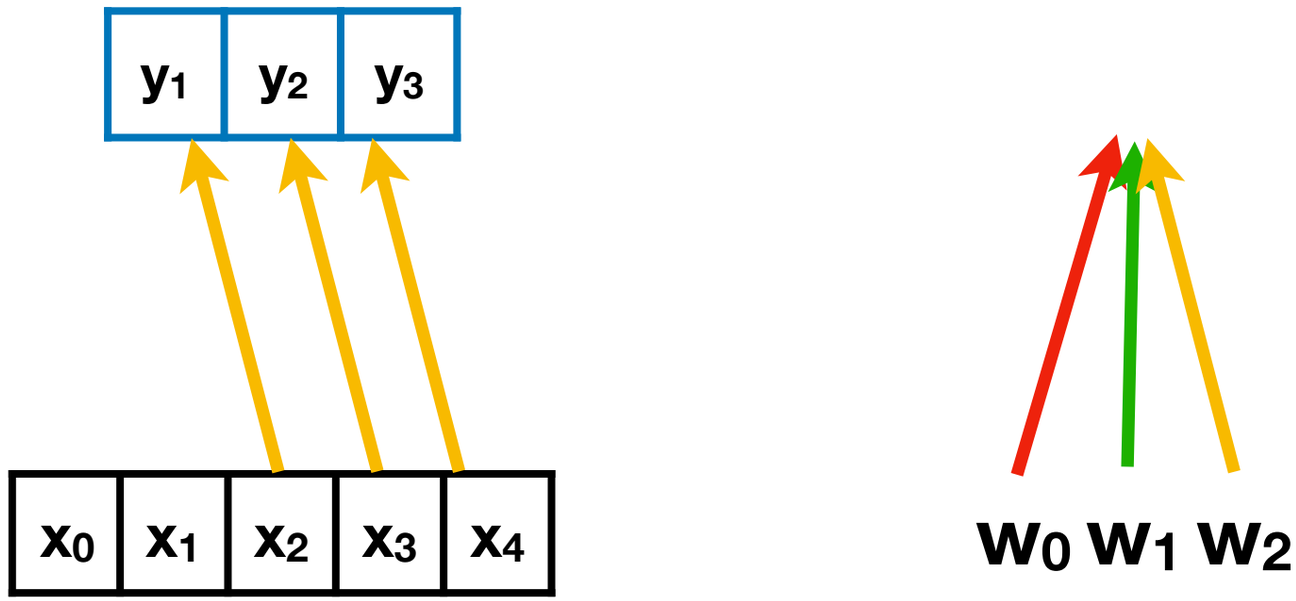
$$dw_0 = dy_1 * x_0 + dy_2 * x_1 + dy_3 * x_2$$

Backward pass - 1D case



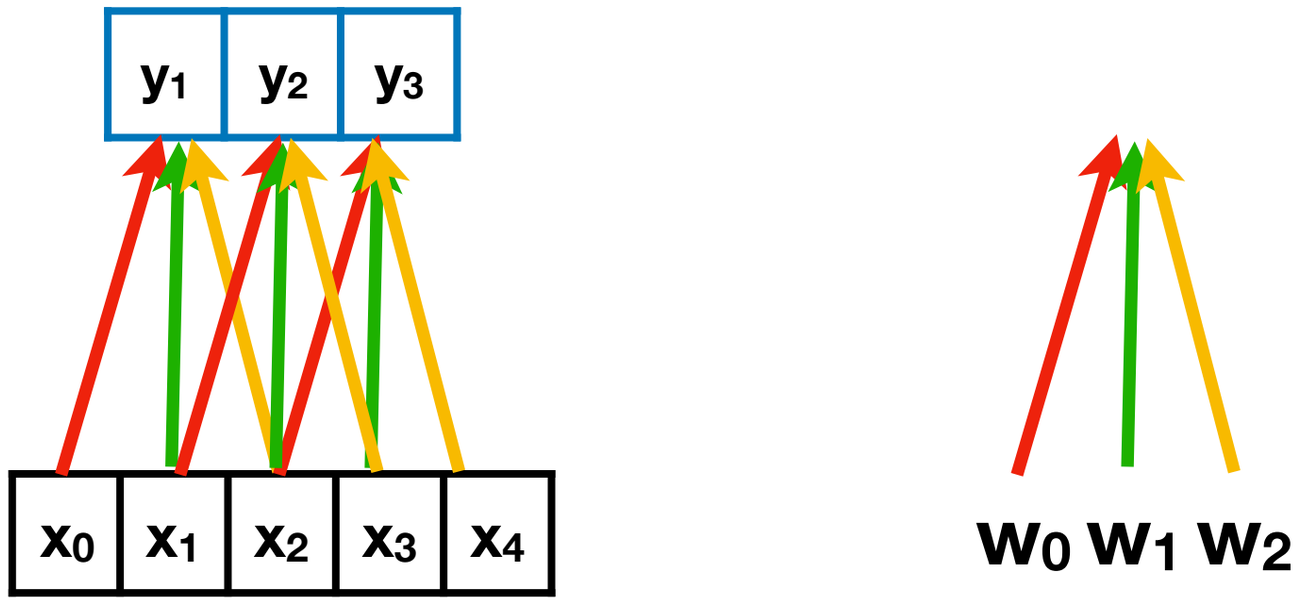
$$dw_1 = dy_1 * x_1 + dy_2 * x_2 + dy_3 * x_3$$

Backward pass - 1D case



$$dw_2 = dy_1 * x_2 + dy_2 * x_3 + dy_3 * x_4$$

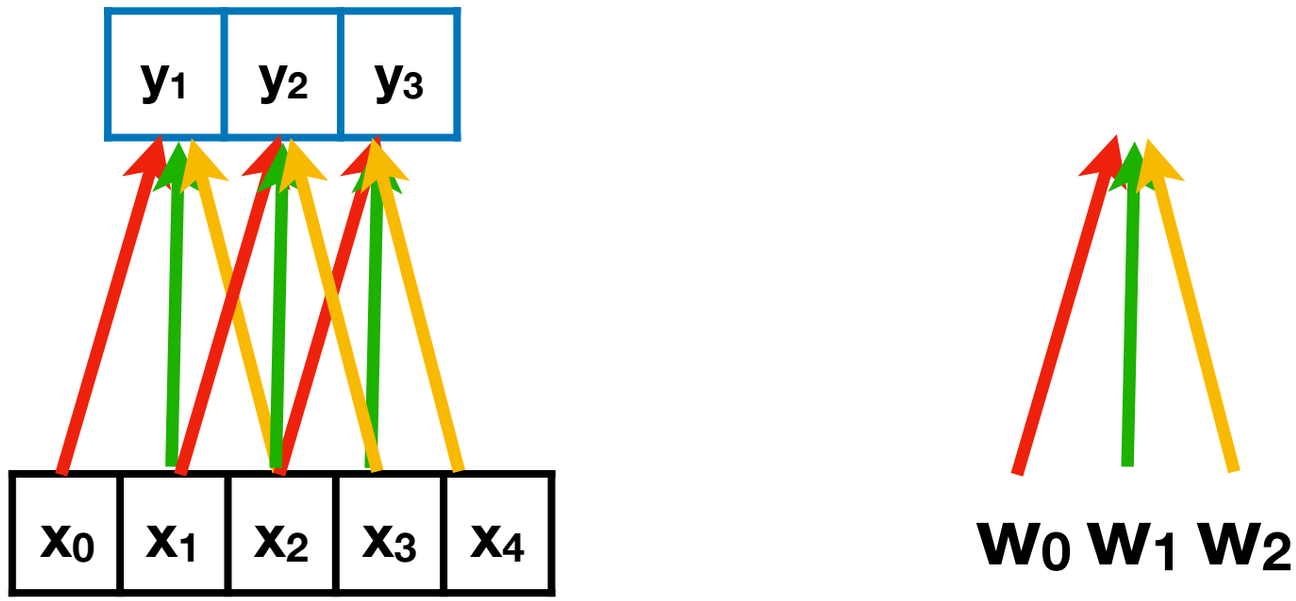
Backward pass - 1D case



$$dw_0 = dy_1 * x_0 + dy_2 * x_1 + dy_3 * x_2$$
$$dw_1 = dy_1 * x_1 + dy_2 * x_2 + dy_3 * x_3$$
$$dw_2 = dy_1 * x_2 + dy_2 * x_3 + dy_3 * x_4$$

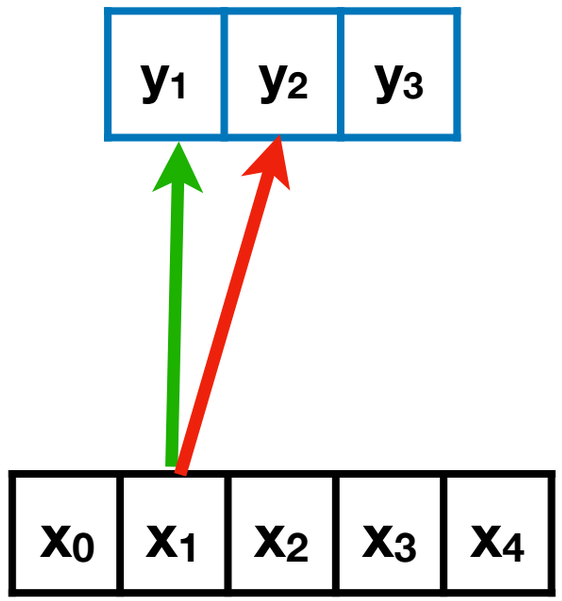
This is the convolution of Y and X

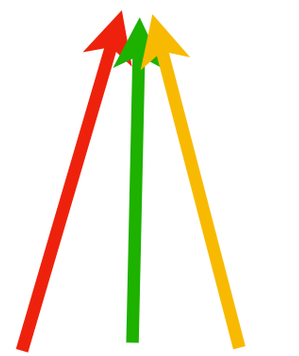
Backward pass - 1D case



Now compute dX

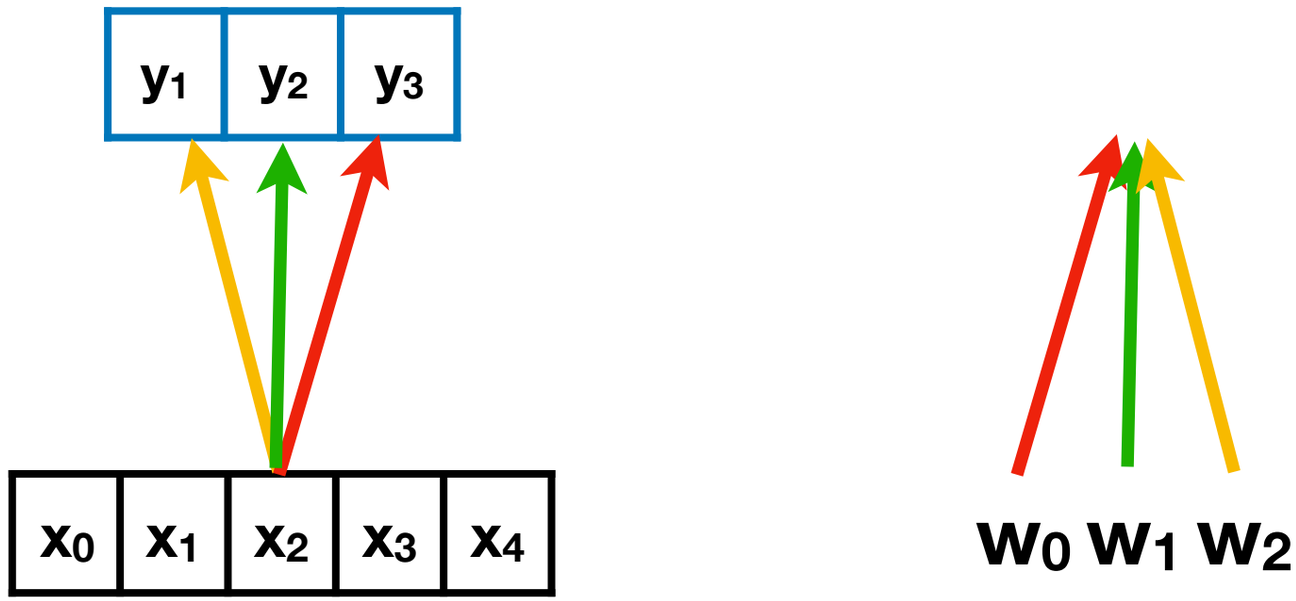
Backward pass - 1D case





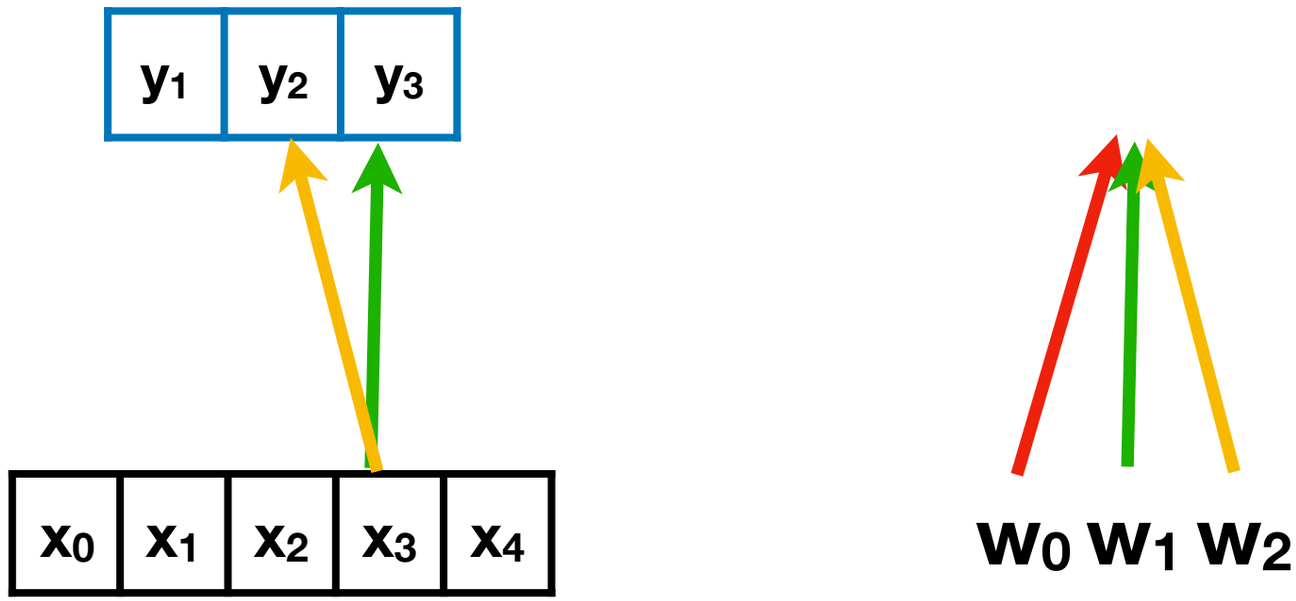
$$dx_1 = \quad + dy_1 * w_1 + dy_2 * w_0$$

Backward pass - 1D case



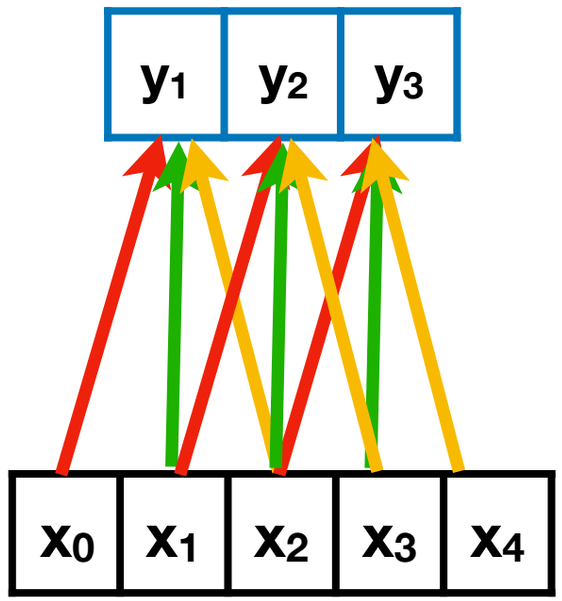
$$dx_2 = dy_1 * w_2 + dy_2 * w_1 + dy_3 * w_0$$

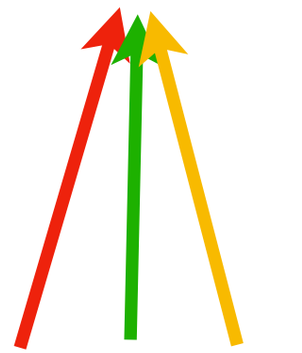
Backward pass - 1D case



$$dx_3 = dy_2 * w_2 + dy_3 * w_1$$

Backward pass - 1D case



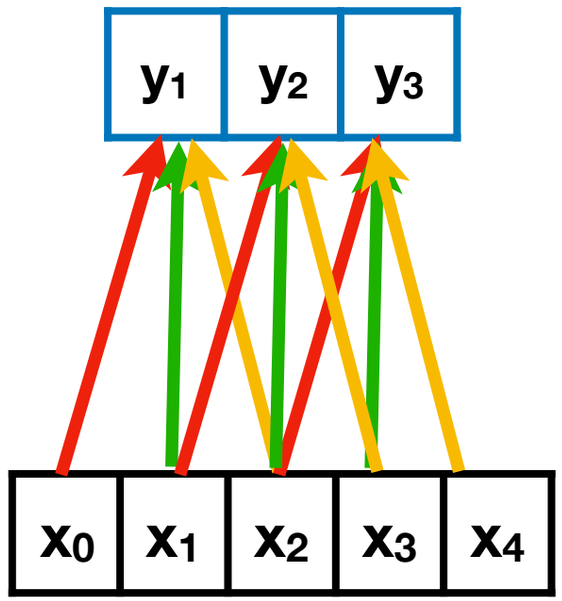


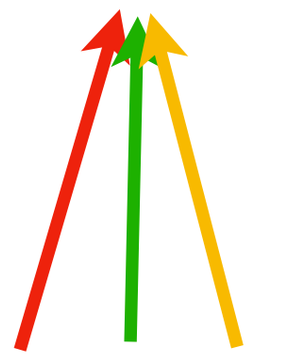
$$dx_1 = dy_1 * w_1 + dy_2 * w_0$$

$$dx_2 = dy_1 * w_2 + dy_2 * w_1 + dy_3 * w_0$$

$$dx_3 = dy_2 * w_2 + dy_3 * w_1$$

Backward pass - 1D case

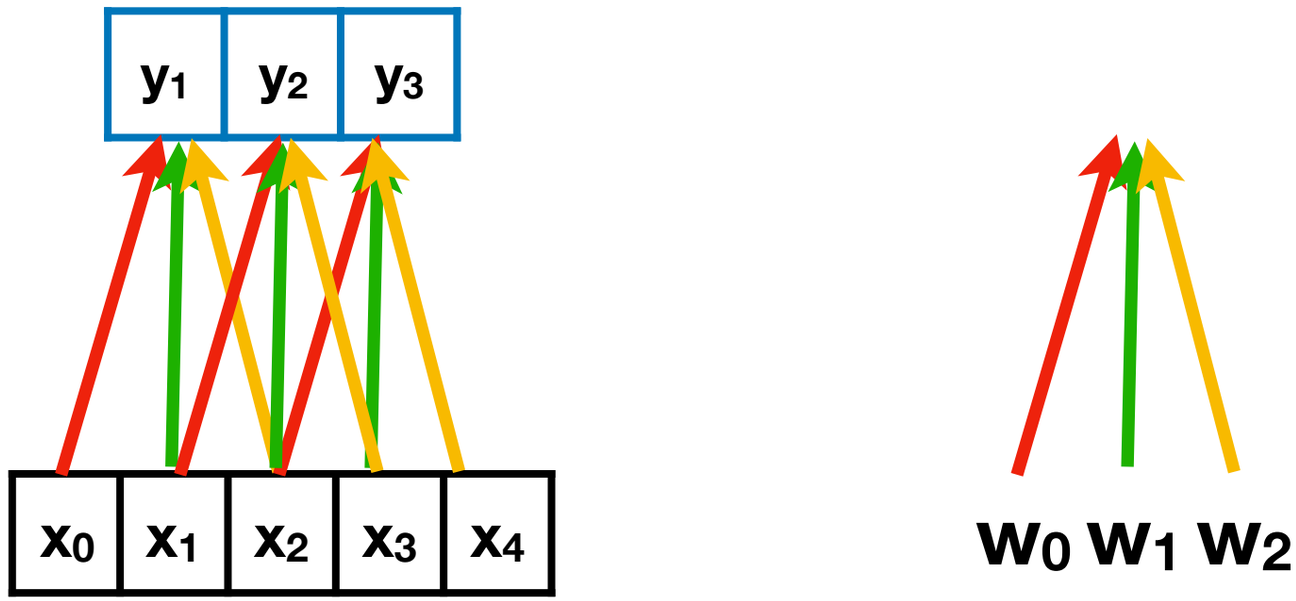




$$dx_1 = dy_1 * w_1 + dy_2 * w_0$$
$$dx_2 = dy_1 * w_2 + dy_2 * w_1 + dy_3 * w_0$$
$$dx_3 = dy_2 * w_2 + dy_3 * w_1$$

This is convolution of Y and flipped filter W !

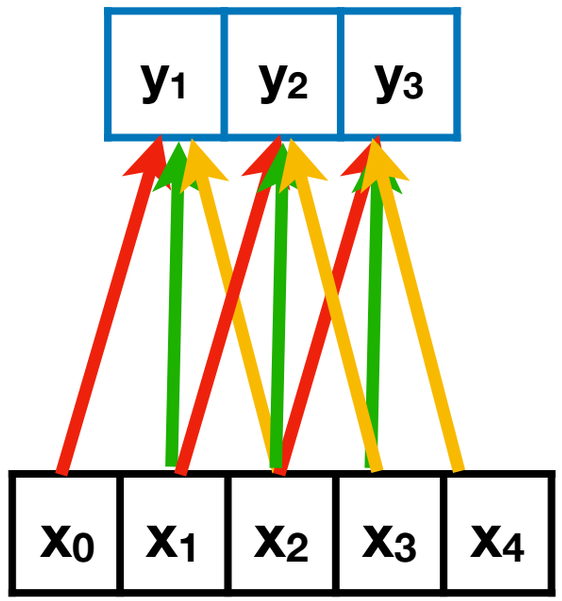
Backward pass - 1D case

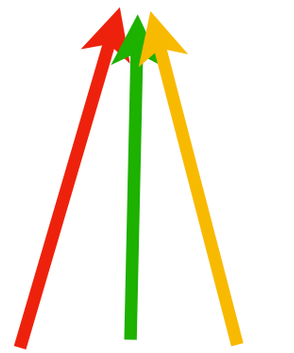


How about db?

$$y_1 = w_0 * x_0 + w_1 * x_1 + w_2 * x_2 + b$$
$$y_2 = w_0 * x_1 + w_1 * x_2 + w_2 * x_3 + b$$
$$y_3 = w_0 * x_2 + w_1 * x_3 + w_2 * x_4 + b$$

Backward pass - 1D case





$$db = dy_1 * 1 + dy_2 * 1 + dy_3 * 1$$

Sum over dY

Now back to 2D case

Always check dimension first:

X —> (N,C,H,W)
W —> (F,C,H_f, W_f)
b —> (F)
Y = W*X + b —> (N,F,H_{out},W_{out})
where H_{out} = (H + 2*pad - H_f) / stride + 1

Now consider only one image and one channel
One filter with size 3x3, input size 5x5

Dimensions:
X —> (N,C,H,W) —> (1,1,5,5)
W —> (F,C,H_f, W_f) —> (1,1,3,3)
b —> (F) —> (1)
Y = W*X + b —> (N,F,H_{out},W_{out}) —> (1,1,3,3)
H_{out} = (H + 2*pad - H_f) / stride + 1 = (5+0-3)/1 + 1 = 3

x_{00}	x_{01}	x_{02}	x_{03}	x_{04}
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}
x_{30}	x_{31}	x_{32}	x_{33}	x_{34}
x_{40}	x_{41}	x_{42}	x_{00}	x_{44}

X 5*5

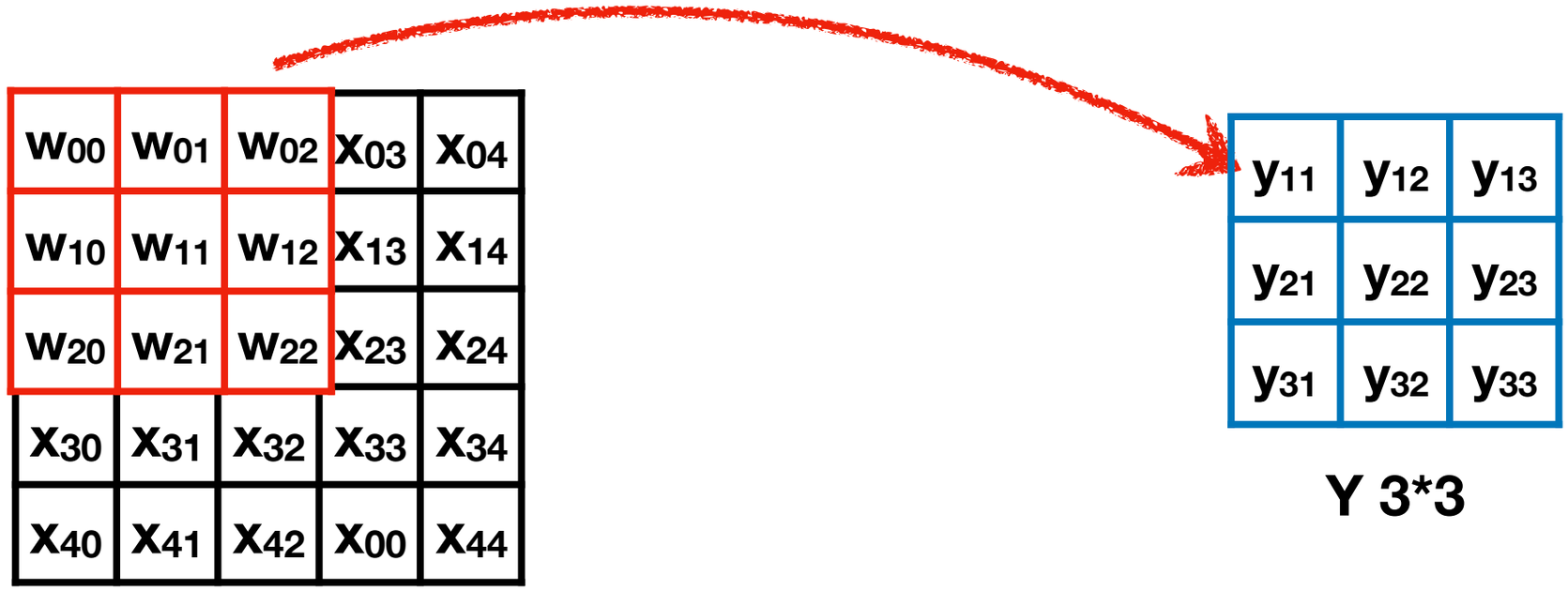
w_{00}	w_{01}	w_{02}
w_{10}	w_{11}	w_{12}
w_{20}	w_{21}	w_{22}

W 3*3

y_{11}	y_{12}	y_{13}
y_{21}	y_{22}	y_{23}
y_{31}	y_{32}	y_{33}

Y 3*3

Forward pass - 2D case



$$y_{11} = \sum_{ij} (W_{ij} * X_{ij}) + b$$
$$\text{for } i,j = 0,1,2$$

Backward pass - 2D case

Compute dX

What locations in y are connected to x_{22}?

x_{00}	x_{01}	x_{02}	x_{03}	x_{04}
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}
x_{30}	x_{31}	x_{32}	x_{33}	x_{34}
x_{40}	x_{41}	x_{42}	x_{00}	x_{44}

X 5*5

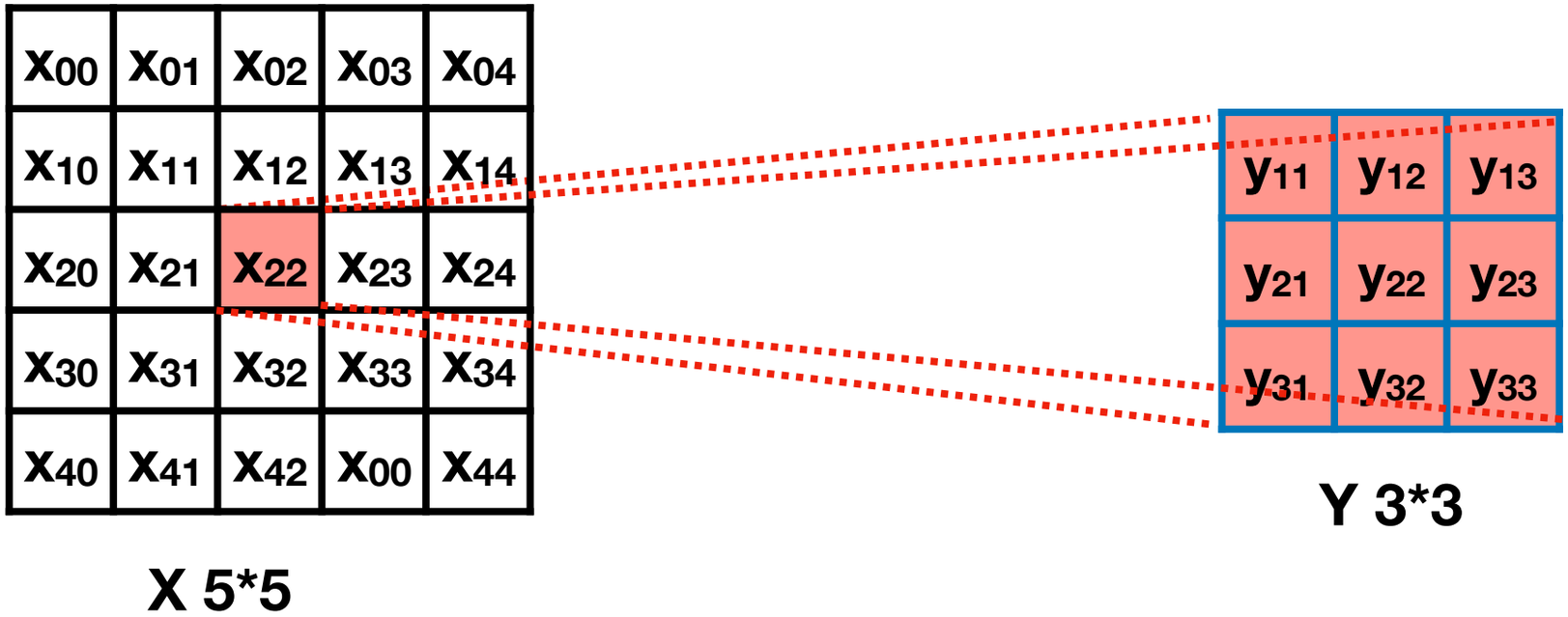
y_{11}	y_{12}	y_{13}
y_{21}	y_{22}	y_{23}
y_{31}	y_{32}	y_{33}

Y 3*3

Backward pass - 2D case

Compute dX

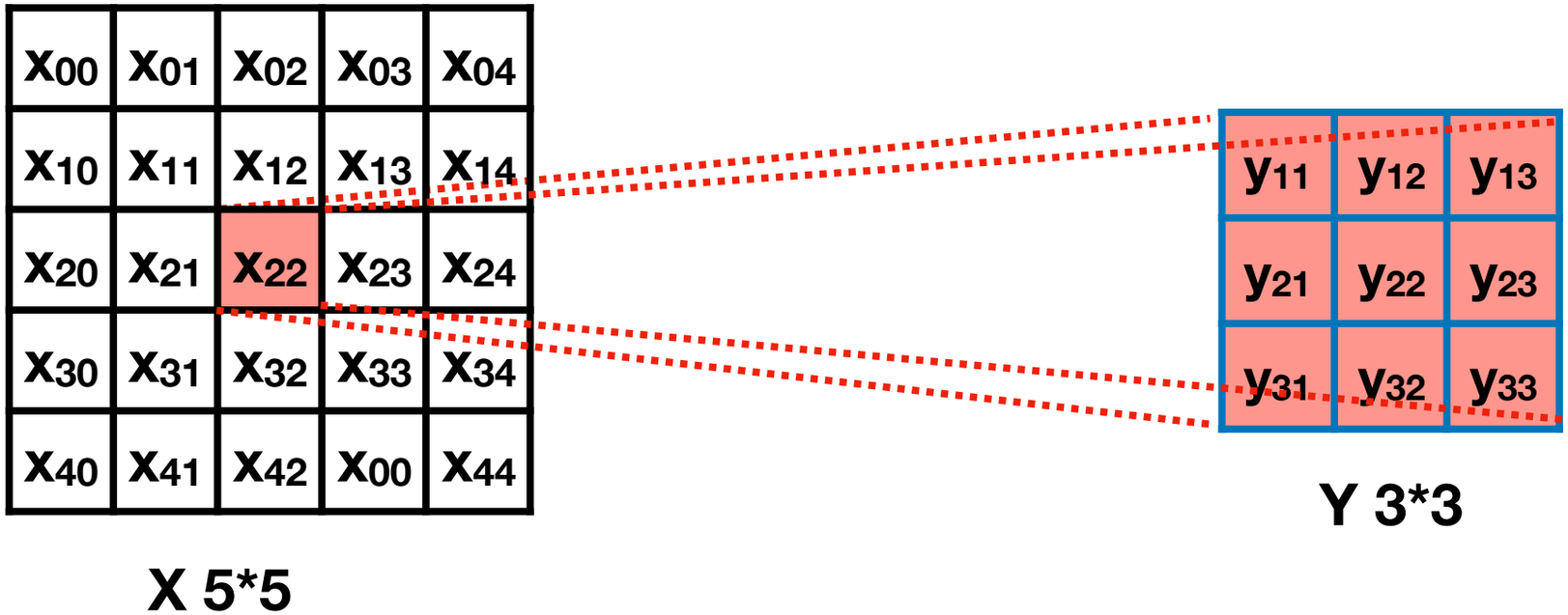
What locations in y are connected to x_{22}?



Backward pass - 2D case

Compute dX

What locations in y are connected to x_{22}?



Like 1D case, you can do convolution between Y and inverted W

Backward pass - 2D case

Compute dX

What locations in y are connected to x_{00}?

x_{00}	x_{01}	x_{02}	x_{03}	x_{04}
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}
x_{30}	x_{31}	x_{32}	x_{33}	x_{34}
x_{40}	x_{41}	x_{42}	x_{00}	x_{44}

X 5*5

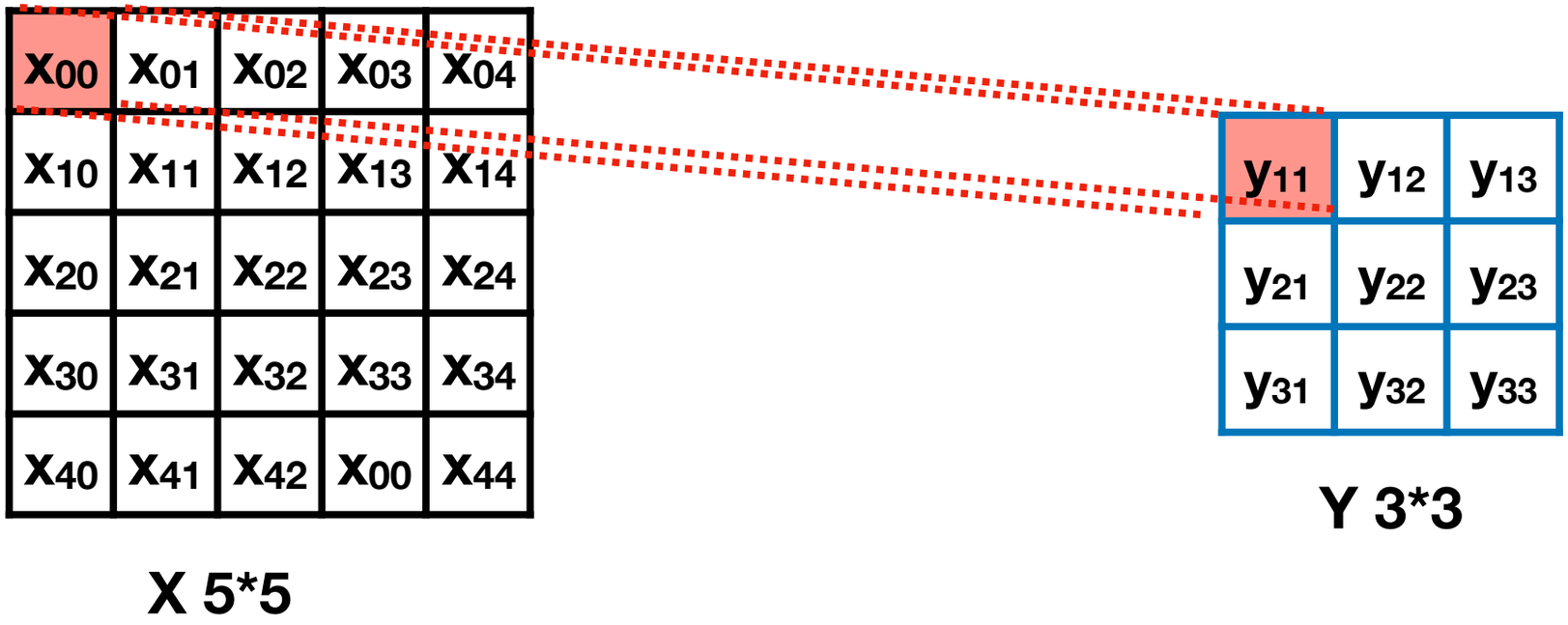
y_{11}	y_{12}	y_{13}
y_{21}	y_{22}	y_{23}
y_{31}	y_{32}	y_{33}

Y 3*3

Backward pass - 2D case

Compute dX

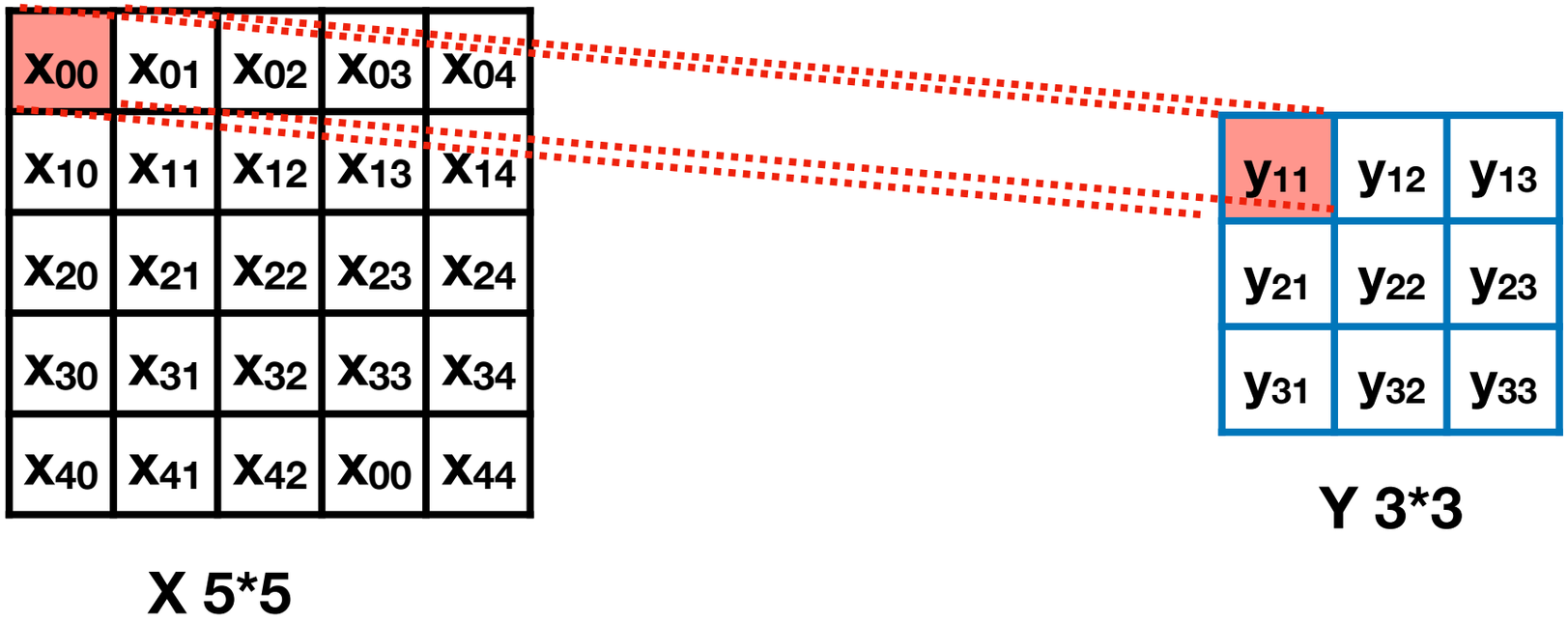
What locations in y are connected to x_{00}?



Backward pass - 2D case

Compute dX

What locations in y are connected to x_{00}?



You can use zero-padding for Y first when computing dX

Backward pass - 2D case

Compute dW

Which x and y pairs are computed using w_{01}?

x_{00}	x_{01}	x_{02}	x_{03}	x_{04}
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}
x_{30}	x_{31}	x_{32}	x_{33}	x_{34}
x_{40}	x_{41}	x_{42}	x_{00}	x_{44}

X 5*5

w_{00}	w_{01}	w_{02}
w_{10}	w_{11}	w_{12}
w_{20}	w_{21}	w_{22}

W 3*3

y_{11}	y_{12}	y_{13}
y_{21}	y_{22}	y_{23}
y_{31}	y_{32}	y_{33}

Y 3*3

Backward pass - 2D case

Compute dW

Which x and y pairs are computed using w_{01}?

x_{00}	x_{01}	x_{02}	x_{03}	x_{04}
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}
x_{30}	x_{31}	x_{32}	x_{33}	x_{34}
x_{40}	x_{41}	x_{42}	x_{00}	x_{44}

X 5*5

w_{00}	w_{01}	w_{02}
w_{10}	w_{11}	w_{12}
w_{20}	w_{21}	w_{22}

W 3*3

y_{11}	y_{12}	y_{13}
y_{21}	y_{22}	y_{23}
y_{31}	y_{32}	y_{33}

Y 3*3

Summary

1. **Always check dimensions**
2. **Pad input X first**
3. **Remove the padding after dX is calculated**
4. **You can also add padding for dY for backward pass**
5. **Add channels**
6. **Add stride**
7. **Add batch**
8. **You can use convolution on backward pass (with flipping)**
9. **There are multiple ways to implement, e.g. you can use many for loops over each location**

Why do we use convolutional layers?

1. **share filters, fewer weights in the model**
2. **Convolution filters are useful for image processing**