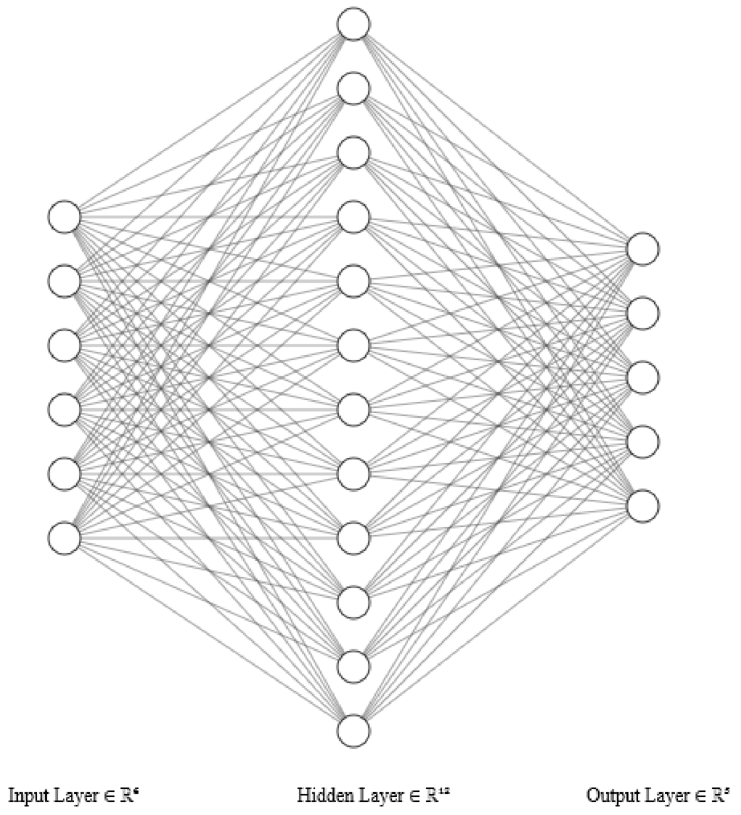
Back propagation for CNN

Prayag and Amrutha

In this lecture we will

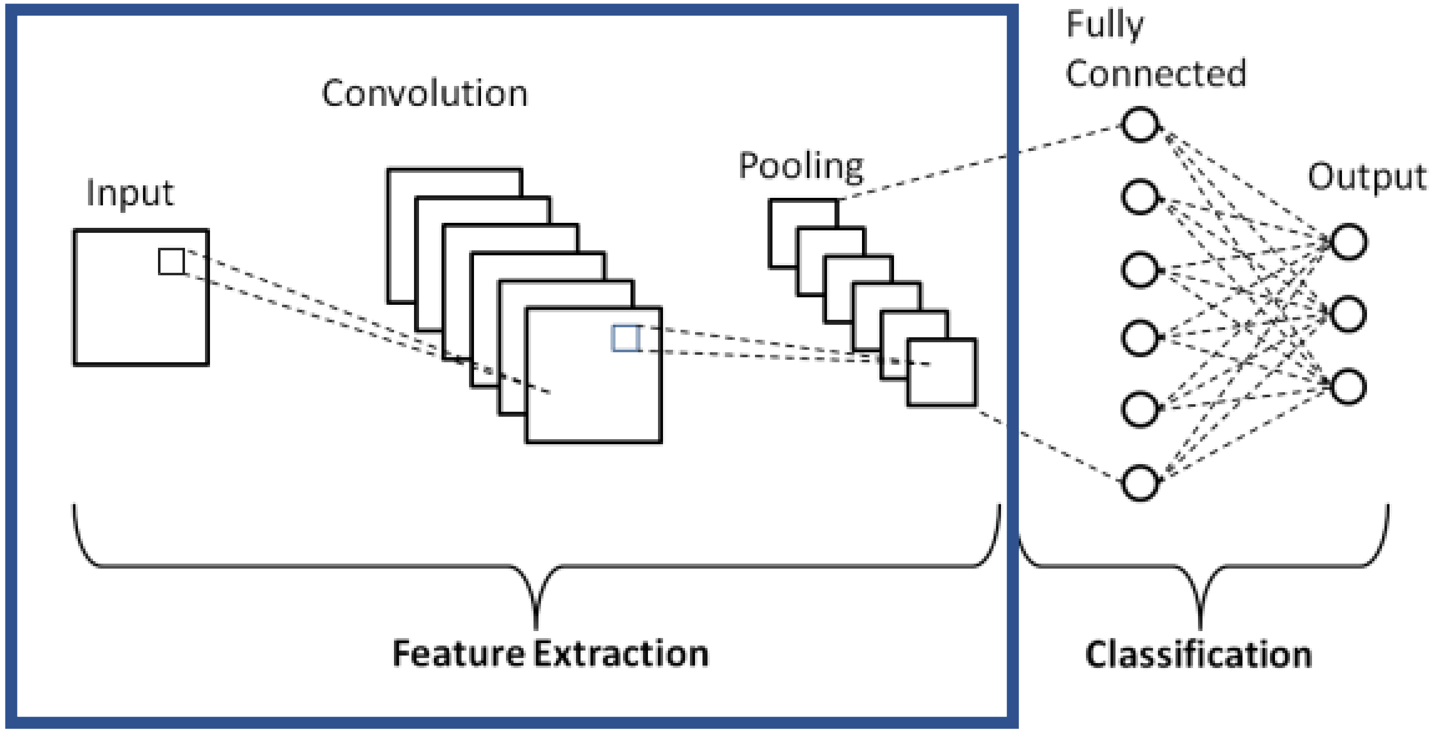
- Briefly revisit MLP
- Look at single block of CNN architecture
- Understand weight sharing concept
- Compute local gradients
- Derive update rule
- Summarize

Multi-layer perceptron



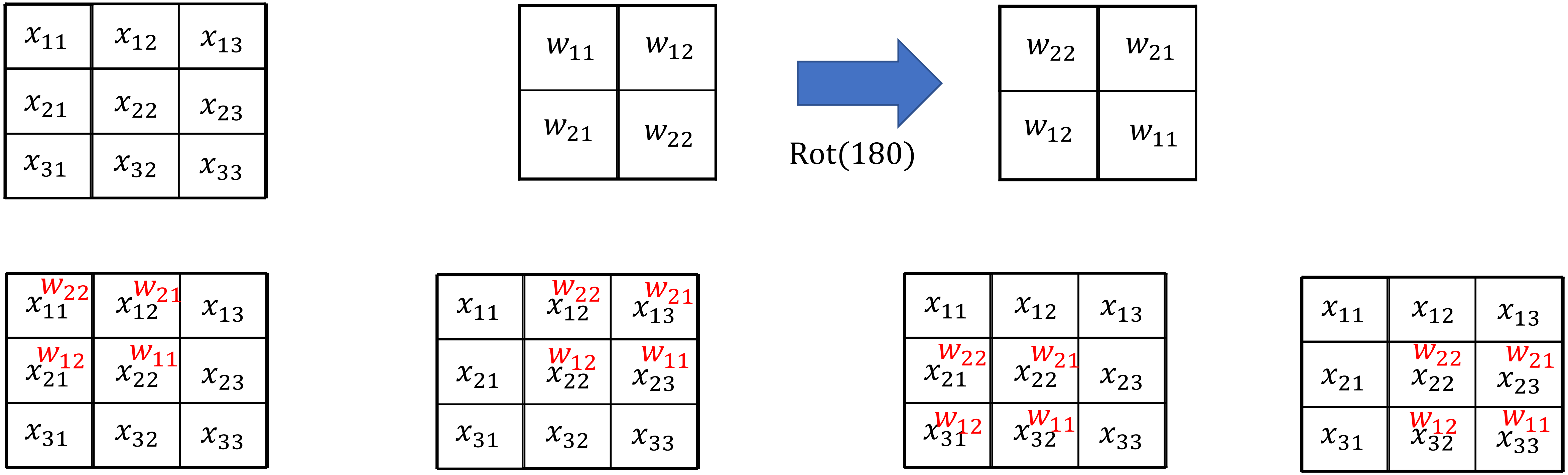
1. Forward pass: Compute the output without any updates.
2. Backward pass:
   - Compute the error E(output, desired signal)
   - Compute the gradient
   - Update the weight parameters of the network

CNN architecture

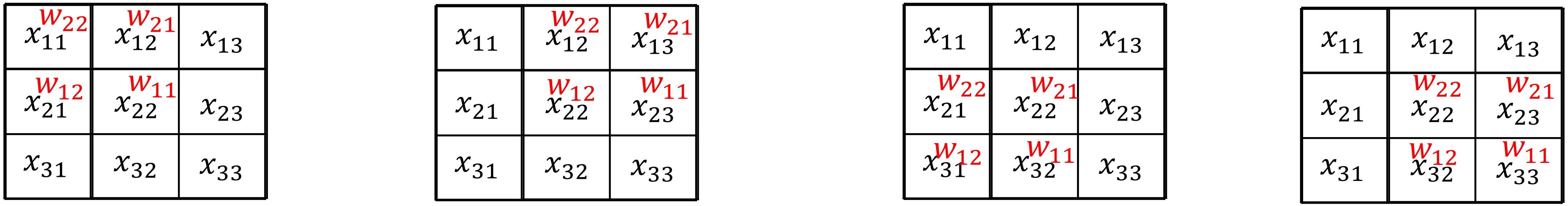


- The block is repeated many times
  - Depending on the application
  - Depending the computational resources available

CNN cont.


x_{11} x_{12} x_{13}
x_{21} x_{22} x_{23}
x_{31} x_{32} x_{33}
w_{11} w_{12}
w_{21} w_{22}
Rot(180)
w_{22} w_{21}
w_{12} w_{11}

CNN cont.



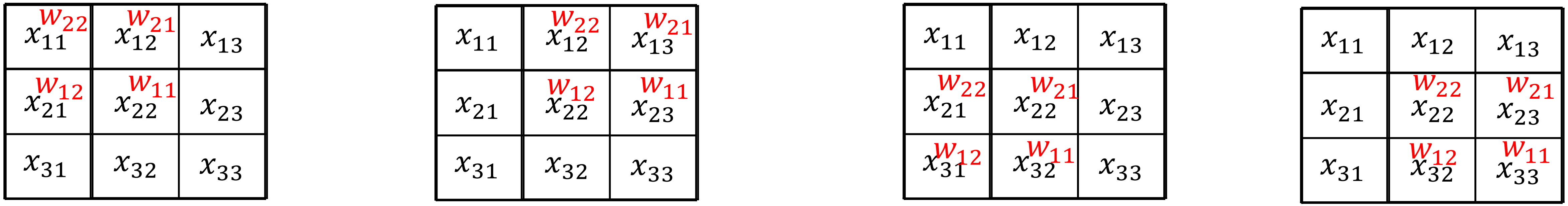
$$h_{11}=x_{11}w_{22} + x_{12}w_{21}+ x_{21}w_{12}+ x_{22}w_{11}$$

$$h_{12}=x_{12}w_{22} + x_{13}w_{21}+ x_{22}w_{12}+ x_{23}w_{11}$$

$$h_{21}=x_{21}w_{22} + x_{22}w_{21}+ x_{31}w_{12}+ x_{32}w_{11}$$

$$h_{22}=x_{22}w_{22} + x_{23}w_{21}+ x_{32}w_{12}+ x_{33}w_{11}$$

Activation function and pooling

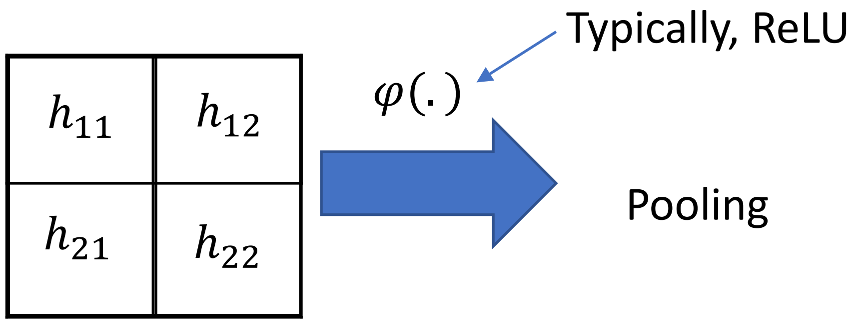


$$h_{11}=x_{11}w_{22} + x_{12}w_{21}+ x_{21}w_{12}+ x_{22}w_{11}$$

$$h_{12}=x_{12}w_{22} + x_{13}w_{21}+ x_{22}w_{12}+ x_{23}w_{11}$$

$$h_{21}=x_{21}w_{22} + x_{22}w_{21}+ x_{31}w_{12}+ x_{32}w_{11}$$

$$h_{22}=x_{22}w_{22} + x_{23}w_{21}+ x_{32}w_{12}+ x_{33}w_{11}$$



Weight sharing

x_{11} w_{22}	x_{12} w_{21}	x_{13}
x_{21} w_{12}	x_{22} w_{11}	x_{23}
x_{31}	x_{32}	x_{33}

x_{11}	x_{12} w_{22}	x_{13} w_{21}
x_{21}	x_{22} w_{12}	x_{23} w_{11}
x_{31}	x_{32}	x_{33}

x_{11}	x_{12}	x_{13}
x_{21} w_{22}	x_{22} w_{21}	x_{23}
x_{31} w_{12}	x_{32} w_{11}	x_{33}

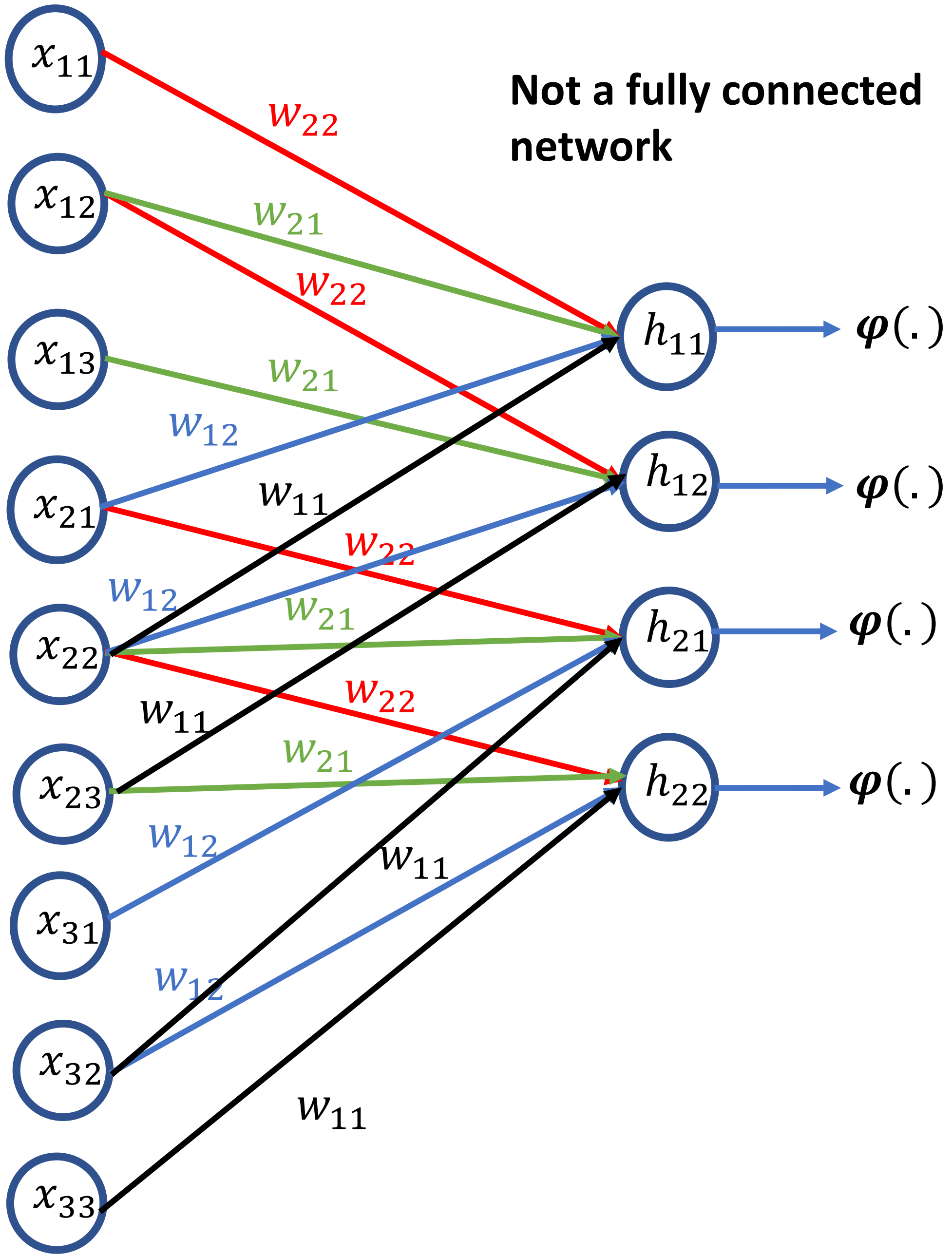
x_{11}	x_{12}	x_{13}
x_{21}	x_{22} w_{22}	x_{23} w_{21}
x_{31}	x_{32} w_{12}	x_{33} w_{11}

$$h_{11}=x_{11}w_{22} + x_{12}w_{21}+ x_{21}w_{12}+ x_{22}w_{11}$$

$$h_{12}=x_{12}w_{22} + x_{13}w_{21}+ x_{22}w_{12}+ x_{23}w_{11}$$

$$h_{21}=x_{21}w_{22} + x_{22}w_{21}+ x_{31}w_{12}+ x_{32}w_{11}$$

$$h_{22}=x_{22}w_{22} + x_{23}w_{21}+ x_{32}w_{12}+ x_{33}w_{11}$$



Forward pass

x_{11} w_{22}	x_{12} w_{21}	x_{13}
x_{21} w_{12}	x_{22} w_{11}	x_{23}
x_{31}	x_{32}	x_{33}

x_{11}	x_{12} w_{22}	x_{13} w_{21}
x_{21}	x_{22} w_{12}	x_{23} w_{11}
x_{31}	x_{32}	x_{33}

x_{11}	x_{12}	x_{13}
x_{21} w_{22}	x_{22} w_{21}	x_{23}
x_{31} w_{12}	x_{32} w_{11}	x_{33}

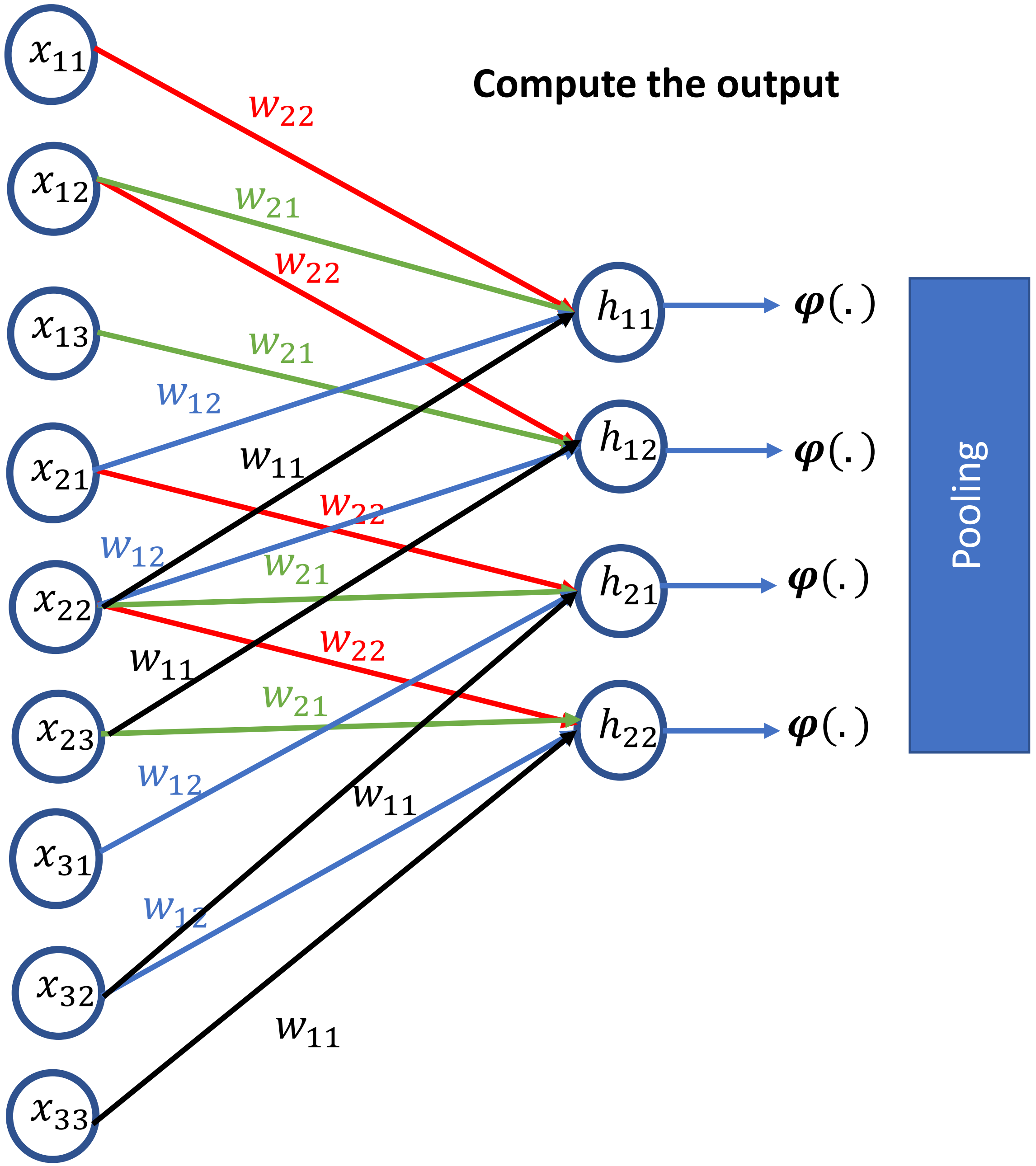
x_{11}	x_{12}	x_{13}
x_{21}	x_{22} w_{22}	x_{23} w_{21}
x_{31}	x_{32} w_{12}	x_{33} w_{11}

$$h_{11}=x_{11}w_{22} + x_{12}w_{21} + x_{21}w_{12} + x_{22}w_{11}$$

$$h_{12}=x_{12}w_{22} + x_{13}w_{21} + x_{22}w_{12} + x_{23}w_{11}$$

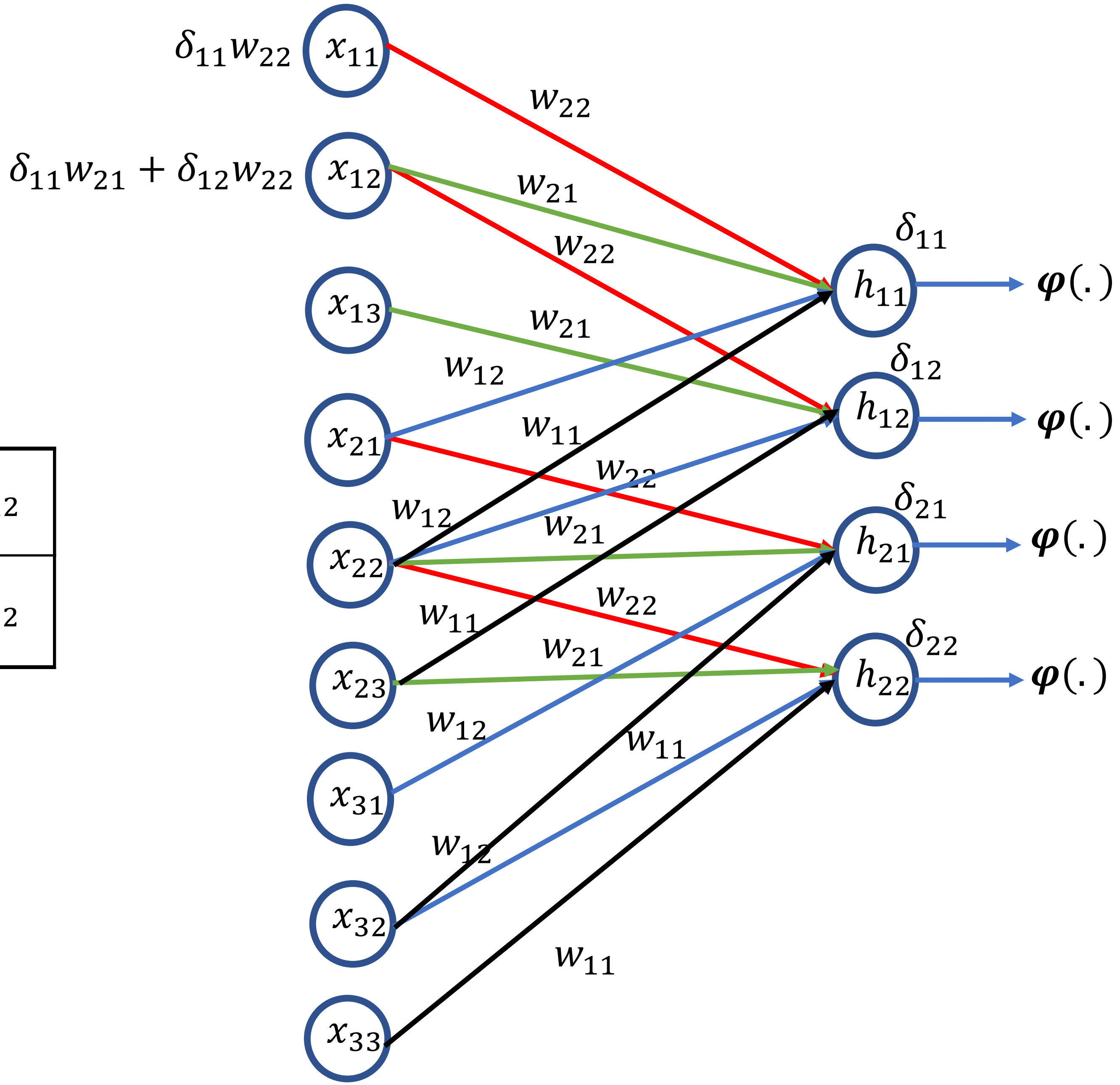
$$h_{21}=x_{21}w_{22} + x_{22}w_{21} + x_{31}w_{12} + x_{32}w_{11}$$

$$h_{22}=x_{22}w_{22} + x_{23}w_{21} + x_{32}w_{12} + x_{33}w_{11}$$



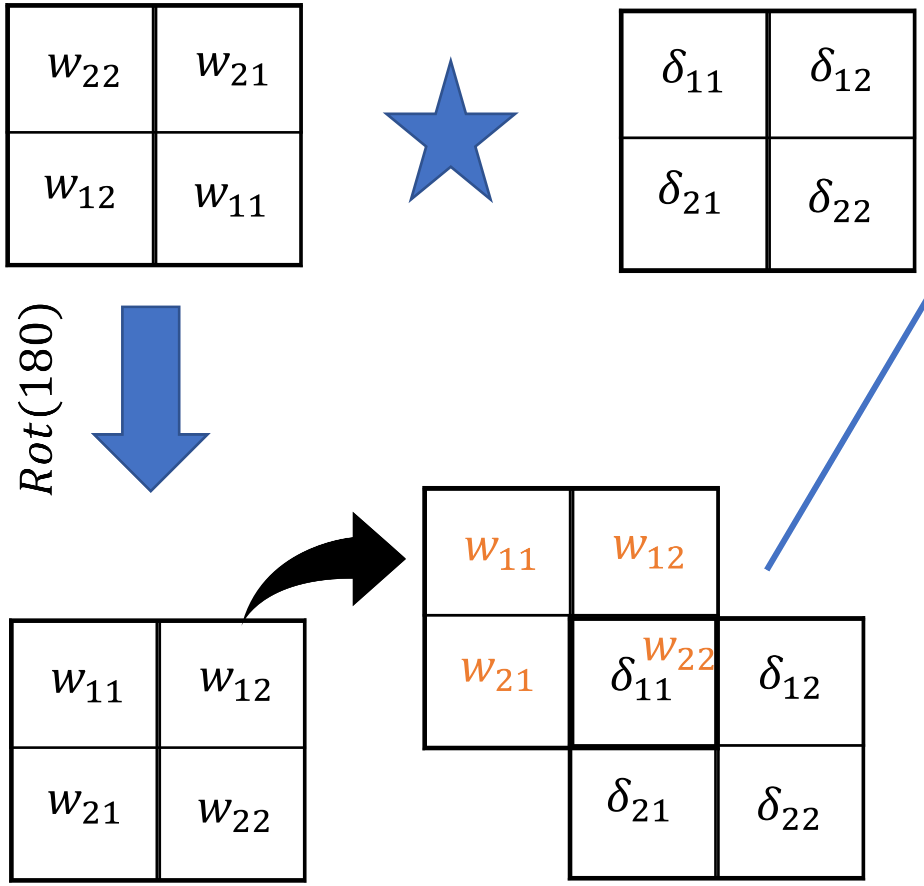
Backward pass

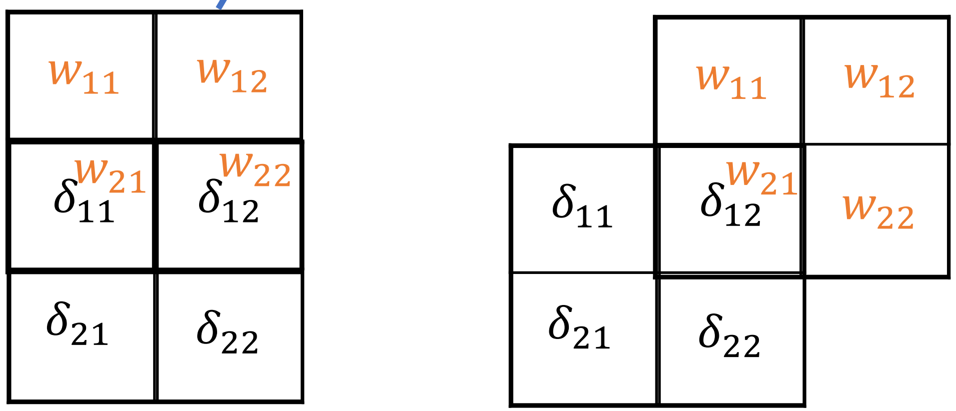
$$\begin{bmatrix} w_{22} & w_{21} \\ w_{12} & w_{11} \end{bmatrix} \star \begin{bmatrix} \delta_{11} & \delta_{12} \\ \delta_{21} & \delta_{22} \end{bmatrix}$$



Backward pass

$\delta_{11}w_{22}$	$\delta_{11}w_{21} + \delta_{12}w_{22}$	$\delta_{12}w_{21}$
$\delta_{11}w_{12} + \delta_{21}w_{22}$	$\delta_{11}w_{11} + \delta_{12}w_{12} +$ $\delta_{21}w_{21} + \delta_{22}w_{22}$	$\delta_{12}w_{21} + \delta_{12}w_{22}$
$\delta_{21}w_{12}$	$\delta_{21}w_{11} + \delta_{22}w_{12}$	$\delta_{22}w_{11}$





Local gradients

- In the MLP, the local gradients of neuron j in the l^{th} layer is given by $\delta_j^{(l)} = \frac{\partial E}{\partial v_j^{(l)}}$
- Where $v_j^{(l)} = \sum_k w_{jk}^{(l)} \phi(v_k^{(l-1)}) + b_j^{(l)}$
- This can be written in a dot product form
- In CNN, the dot product is replaced by a convolution operator and we define the local gradient as $\delta_j^{(l)} = \frac{\partial E}{\partial v_j^{(l)}}$
- where $v_{x,y}^{(l)} = \sum_a \sum_b w_{a,b}^{(l)} \phi(v_{x-a,y-b}^{(l-1)}) + b_{x,y}^{(l)}$

Local gradients cont.

$$\delta^{(l)}_{x,y} = \sum_{x'}\sum_{y'} \frac{\partial E}{\partial v^{(l+1)}_{x',y'}} \frac{\partial v^{(l+1)}_{x',y'}}{\partial v^{(l)}_{x,y}} = \sum_{x'}\sum_{y'} \delta^{(l+1)}_{x',y'} \frac{\partial(\sum_{a}\sum_{b} w^{(l+1)}_{a,b}\phi(v^{(l)}_{x'-a,y'-b}) + b^{(l)}_{x',y'})}{\partial v^{(l)}_{x,y}}$$

Let $x' - a = x$ and $y' - b = y$

$l + 1$

$$\delta^{(l)}_{x,y} = \sum_{x'}\sum_{y'} \delta^{(l+1)}_{x',y'} w^{(l+1)}_{x'-xy'-y}\phi(v^{(l)}_{x,y})' = \delta^{(l)} \star w^{(l+1)}_{-x,-y}\phi(v^{(l)}_{x,y})'$$

Where $w^{(l+1)}_{-x,-y} = Rot_{180}(w^{(l+1)}_{x,y})$

Gradient computation

$$\frac{\partial E}{\partial w_{a,b}^{(l)}} = \sum_x \sum_y \frac{\partial E}{\partial v_{x,y}^{(l)}} \frac{\partial v_{x,y}^{(l)}}{\partial w_{a,b}^{(l)}} = \sum_x \sum_y \delta_{x,y}^{(l)} \frac{\partial(\sum_{a'} \sum_{b'} w_{a',b'}^{(l)} \phi(v_{x-a',y-b'}^{(l-1)}) + b_{x,y}^{(l)})}{\partial w_{a,b}^{(l)}}$$

When $a = a'$ and $b = b'$

$$\frac{\partial E}{\partial w_{a,b}^{(l)}} = \sum_x \sum_y \delta_{x,y}^{(l)} \phi(v_{x-a,y-b}^{(l-1)}) = \delta^{(l)} \star \phi(v_{-a,-b}^{(l-1)})$$

Where $\phi(v_{-a,-b}^{(l-1)}) = Rot_{180}(\phi(v_{a,b}^{(l-1)}))$

Summary

- Compute the error E at the output
- For every input compute $v_{x,y}^{(l)} = \sum_a \sum_b w_{a,b}^{(l)} \phi(v_{x-a,y-b}^{(l-1)}) + b_{x,y}^{(l)}$
- During back propagation, we compute the local gradient $\delta_{x,y}^{(l)}$ as

$l+1$

$$\delta_{x,y}^{(l)} = \sum_{x'} \sum_{y'} \delta_{x',y'}^{(l+1)} w_{x'-xy'-y}^{(l+1)} \phi(v_{x,y}^{(l)})' = \delta^{(l)} \star w_{-x,-y}^{(l+1)} \phi(v_{x,y}^{(l)})'$$

- Compute $\frac{\partial E}{\partial w_{a,b}^{(l)}} = \sum_x \sum_y \delta_{x,y}^{(l)} \phi(v_{x-a,y-b}^{(l-1)}) = \delta^{(l)} \star \phi(v_{-a,-b}^{(l-1)})$