UE21CS343BB2

Topics in Deep Learning

Dr. Shylaja S S
Director of Cloud Computing & Big Data (CCBD), Centre for Data Sciences & Applied Machine Learning (CDSAML)
Department of Computer Science and Engineering
shylaja.sharath@pes.edu

Ack: Divya K,
Teaching Assistant

Overview of lecture

- Recap: Backpropagation
- Convolution Forward Pass
- Convolution Backward Pass
    - Calculation of loss gradient w.r.t filter
    - Calculation of loss gradient w.r.t bias
    - Calculation of loss gradient w.r.t input

- Conclusion: Backpropagation in Convolution Layer

Recap: Backpropagation

PES UNIVERSITY

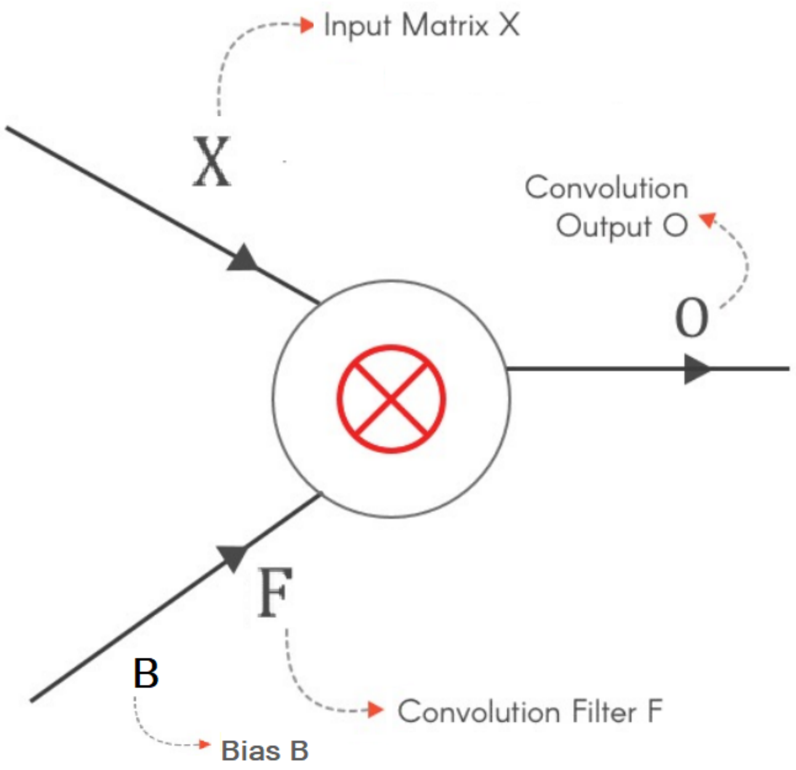
- Backpropagation is an algorithm used to train neural networks by adjusting the weights of the network based on the error between the predicted output and the actual output.

- Backpropagation calculates the gradient of the loss function with respect to each parameter in the network and updates the network parameters in such a way that it minimizes the loss function.

- In CNNs the loss gradient is computed w.r.t the input and also w.r.t the filter, w.r.t the bias.

Convolution Forward Pass

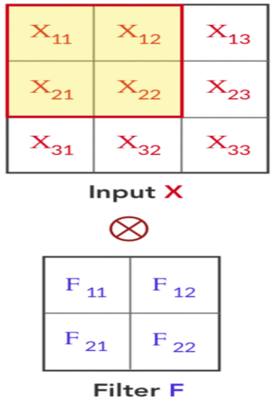
Convolution between Input X and Filter F, gives us an output O. This can be represented as:



$$\underset{\text{Output } O}{\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix}} = \text{Convolution}\left(\underset{\text{Input } X}{\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}}, \underset{\text{Filter } F}{\begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}} \right) + \text{Bias } B$$

Convolution Forward Pass

Applying the convolution operation, we get:



$X_{11}F_{11}$	$X_{12}F_{12}$	X_{13}
$X_{21}F_{21}$	$X_{22}F_{22}$	X_{23}
X_{31}	X_{32}	X_{33}

$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$$

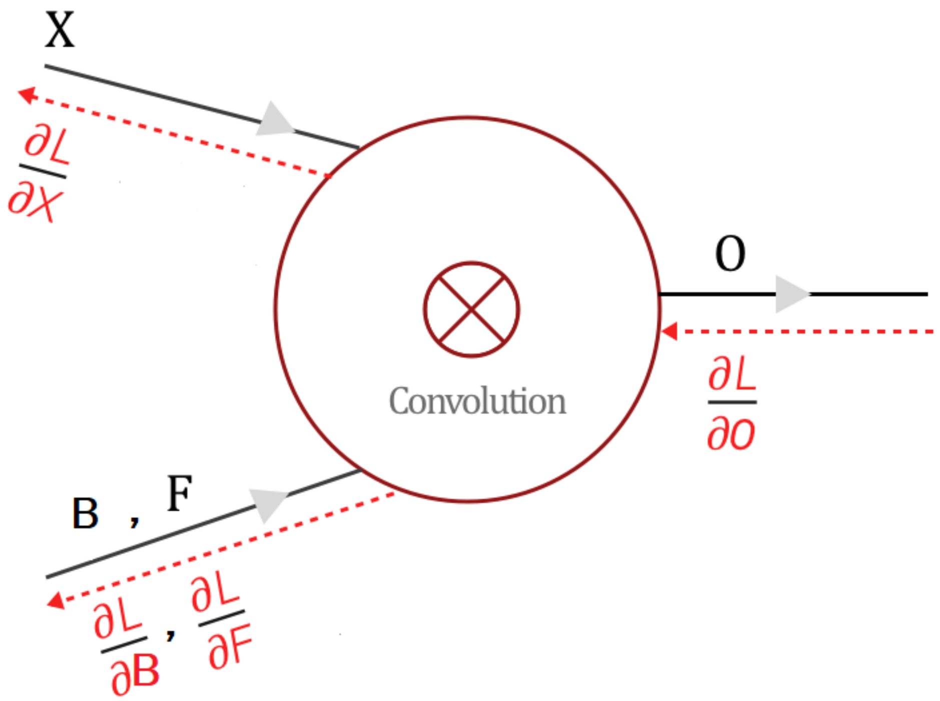
$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22} + B$$

$$O_{12} = X_{12}F_{11} + X_{13}F_{12} + X_{22}F_{21} + X_{23}F_{22} + B$$

$$O_{21} = X_{21}F_{11} + X_{22}F_{12} + X_{31}F_{21} + X_{32}F_{22} + B$$

$$O_{22} = X_{22}F_{11} + X_{23}F_{12} + X_{32}F_{21} + X_{33}F_{22} + B$$

Convolution Backward Pass



In ANNs, we update the weights as
W = W - α*∂L/∂W
In CNNs, we update the network parameters as:
F = F - α***∂L/∂F**
B = B - α***∂L/∂B**
(where α is the learning parameter)
Since X is the output of the previous layer, **∂L/∂X** becomes the loss gradient for the previous layer.
So, we need to calculate **∂L/∂F, ∂L/∂B** and **∂L/∂X**.

Convolution Backward Pass

➢ Calculation of loss gradient w.r.t the filter, **∂L/∂F:**

We can use the chain rule to obtain the gradient w.r.t the filter as shown in the equation.

$$\underbrace{\frac{\partial L}{\partial F}}_{\text{Gradient to update Filter F}} = \underbrace{\frac{\partial L}{\partial O}}_{\text{Loss Gradient from previous layer}} * \underbrace{\frac{\partial O}{\partial F}}_{\text{Local Gradients}}$$

For every element of F

$$\frac{\partial L}{\partial F_i} = \sum_{k=1}^{M} \frac{\partial L}{\partial O_k} * \frac{\partial O_k}{\partial F_i}$$

Convolution Backward Pass

➢ Calculation of loss gradient w.r.t the filter, **∂L/∂F:**

On expanding the chain rule summation, we get:

$$\frac{\partial L}{\partial F_{11}} = \frac{\partial L}{\partial O_{11}} * \frac{\partial O_{11}}{\partial F_{11}} + \frac{\partial L}{\partial O_{12}} * \frac{\partial O_{12}}{\partial F_{11}} + \frac{\partial L}{\partial O_{21}} * \frac{\partial O_{21}}{\partial F_{11}} + \frac{\partial L}{\partial O_{22}} * \frac{\partial O_{22}}{\partial F_{11}}$$

$$\frac{\partial L}{\partial F_{12}} = \frac{\partial L}{\partial O_{11}} * \frac{\partial O_{11}}{\partial F_{12}} + \frac{\partial L}{\partial O_{12}} * \frac{\partial O_{12}}{\partial F_{12}} + \frac{\partial L}{\partial O_{21}} * \frac{\partial O_{21}}{\partial F_{12}} + \frac{\partial L}{\partial O_{22}} * \frac{\partial O_{22}}{\partial F_{12}}$$

$$\frac{\partial L}{\partial F_{21}} = \frac{\partial L}{\partial O_{11}} * \frac{\partial O_{11}}{\partial F_{21}} + \frac{\partial L}{\partial O_{12}} * \frac{\partial O_{12}}{\partial F_{21}} + \frac{\partial L}{\partial O_{21}} * \frac{\partial O_{21}}{\partial F_{21}} + \frac{\partial L}{\partial O_{22}} * \frac{\partial O_{22}}{\partial F_{21}}$$

$$\frac{\partial L}{\partial F_{22}} = \frac{\partial L}{\partial O_{11}} * \frac{\partial O_{11}}{\partial F_{22}} + \frac{\partial L}{\partial O_{12}} * \frac{\partial O_{12}}{\partial F_{22}} + \frac{\partial L}{\partial O_{21}} * \frac{\partial O_{21}}{\partial F_{22}} + \frac{\partial L}{\partial O_{22}} * \frac{\partial O_{22}}{\partial F_{22}}$$

➢ Calculation of loss gradient w.r.t the filter, ∂L/∂F:

To calculate **∂O/∂F**:

From these equations,

$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22} + B$$

$$O_{12} = X_{12}F_{11} + X_{13}F_{12} + X_{22}F_{21} + X_{23}F_{22} + B$$

$$O_{21} = X_{21}F_{11} + X_{22}F_{12} + X_{31}F_{21} + X_{32}F_{22} + B$$

$$O_{22} = X_{22}F_{11} + X_{23}F_{12} + X_{32}F_{21} + X_{33}F_{22} + B$$

We get:

$$\frac{\partial O_{11}}{\partial F_{11}} = X_{11}, \frac{\partial O_{11}}{\partial F_{12}} = X_{12}, \frac{\partial O_{11}}{\partial F_{21}} = X_{21}, \frac{\partial O_{11}}{\partial F_{22}}$$

$$\frac{\partial O_{12}}{\partial F_{11}} = X_{12}, \frac{\partial O_{12}}{\partial F_{12}} = X_{13}, \frac{\partial O_{12}}{\partial F_{21}} = X_{22},$$

$$\frac{\partial O_{21}}{\partial F_{11}} = X_{21}, \frac{\partial O_{21}}{\partial F_{12}} = X_{22}, \frac{\partial O_{21}}{\partial F_{21}} = X_{31},$$

$$\frac{\partial O_{22}}{} = X_{22}, \frac{\partial O_{22}}{} = X_{23}, \frac{\partial O_{22}}{} = X_{32},$$

Convolution Backward Pass

PES UNIVERSITY

- Calculation of loss gradient w.r.t the filter, **∂L/∂F:**

On substituting the values of the local gradient, we get:

$$\frac{\partial L}{\partial F_{11}} = \frac{\partial L}{\partial O_{11}} * X_{11} + \frac{\partial L}{\partial O_{12}} * X_{12} + \frac{\partial L}{\partial O_{21}} * X_{21} + \frac{\partial L}{\partial O_{22}} * X_{22}$$

$$\frac{\partial L}{\partial F_{12}} = \frac{\partial L}{\partial O_{11}} * X_{12} + \frac{\partial L}{\partial O_{12}} * X_{13} + \frac{\partial L}{\partial O_{21}} * X_{22} + \frac{\partial L}{\partial O_{22}} * X_{23}$$

$$\frac{\partial L}{\partial F_{21}} = \frac{\partial L}{\partial O_{11}} * X_{21} + \frac{\partial L}{\partial O_{12}} * X_{22} + \frac{\partial L}{\partial O_{21}} * X_{31} + \frac{\partial L}{\partial O_{22}} * X_{32}$$

$$\frac{\partial L}{\partial F_{22}} = \frac{\partial L}{\partial O_{11}} * X_{22} + \frac{\partial L}{\partial O_{12}} * X_{23} + \frac{\partial L}{\partial O_{21}} * X_{32} + \frac{\partial L}{\partial O_{22}} * X_{33}$$

Convolution Backward Pass

➢ Calculation of loss gradient w.r.t the filter, **∂L/∂F:**

If we look closely, this can be represented as a **convolution operation between input X** and **loss gradient ∂L/∂O** as shown below:

$$\begin{bmatrix} \frac{\partial L}{\partial F_{11}} & \frac{\partial L}{\partial F_{12}} \\ \frac{\partial L}{\partial F_{21}} & \frac{\partial L}{\partial F_{22}} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} \frac{\partial L}{\partial O_{11}} & \frac{\partial L}{\partial O_{12}} \\ \frac{\partial L}{\partial O_{21}} & \frac{\partial L}{\partial O_{22}} \end{bmatrix} \right)$$

where

$$\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix} = \text{Input X}$$

$$\begin{bmatrix} \frac{\partial L}{\partial O_{11}} & \frac{\partial L}{\partial O_{12}} \\ \frac{\partial L}{\partial O_{21}} & \frac{\partial L}{\partial O_{22}} \end{bmatrix} = \frac{\partial L}{\partial O}$$

Loss gradient from previous layer

$$\frac{\partial L}{\partial F} = \text{conv}\left(X, \frac{\partial L}{\partial O}\right)$$

∂L/∂F = Convolution of input matrix X and loss gradient ∂L/∂O

➢ Calculation of loss gradient w.r.t the bias, **∂L/∂B:**

We can use the chain rule to obtain the gradient w.r.t the bias as shown in the equation.

$$\frac{\partial L}{\partial B} = \frac{\partial L}{\partial O} * \frac{\partial O}{\partial B}$$

Gradient to update **Bias B**

Loss Gradient from previous layer

Local Gradients

$$\frac{\partial L}{\partial B} = \sum_{k=1}^{M} \frac{\partial L}{\partial O_k} * \frac{\partial O_k}{\partial B}$$

➢ Calculation of loss gradient w.r.t the bias, **∂L/∂B:**

To calculate **∂O/∂B,** we just partially derive $\mathbf{O_{11}}$, $\mathbf{O_{12}}$, $\mathbf{O_{21}}$, and $\mathbf{O_{22}}$ with respect to **B.** Since there is only one **B** term in each **O** term (as shown), the partial differentiation just returns 1.

$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22} + B$$
$$O_{12} = X_{12}F_{11} + X_{13}F_{12} + X_{22}F_{21} + X_{23}F_{22} + B$$
$$O_{21} = X_{21}F_{11} + X_{22}F_{12} + X_{31}F_{21} + X_{32}F_{22} + B$$
$$O_{22} = X_{22}F_{11} + X_{23}F_{12} + X_{32}F_{21} + X_{33}F_{22} + B$$

$$\longrightarrow$$

$$\frac{\partial O_{11}}{\partial B} = 1$$

$$\frac{\partial O_{12}}{\partial B} = 1$$

$$\frac{\partial O_{21}}{\partial B} = 1$$

$$\frac{\partial O_{22}}{\partial B} = 1$$

➢ Calculation of loss gradient w.r.t the bias, **∂L/∂B:**

So **∂L/∂B** is just equal to the summation of **∂L/∂O** terms.

$$\frac{\partial L}{\partial B} = \sum \frac{\partial L}{\partial O_k} * \cancel{\frac{\partial O_k}{\partial B}}^{1}$$

$$\frac{\partial L}{\partial B} = \frac{\partial L}{\partial O_{11}} + \frac{\partial L}{\partial O_{12}} + \frac{\partial L}{\partial O_{21}} + \frac{\partial L}{\partial O_{22}}$$

$$\frac{\partial L}{\partial B} = \sum_{k=1}^{M} \frac{\partial L}{\partial O_k}$$

$$\boxed{\frac{\partial L}{\partial B} = \text{sum}\left(\frac{\partial L}{\partial O}\right)}$$

Convolution Backward Pass

➢ Calculation of loss gradient w.r.t the input, **∂L/∂X:**

We can use the chain rule to obtain the gradient w.r.t the input as shown in the equation.

$$\frac{\partial L}{\partial X} = \frac{\partial L}{\partial O} * \frac{\partial O}{\partial X}$$

Gradient to update **input X**

Loss Gradient from previous layer

Local Gradients

For every element of X_i

$$\frac{\partial L}{\partial X_i} = \sum_{k=1}^{M} \frac{\partial L}{\partial O_k} * \frac{\partial O_k}{\partial X_i}$$

Convolution Backward Pass

➢ Calculation of loss gradient w.r.t the input, **∂L/∂X:**

On expanding the chain rule summation and substituting the values of the local gradients, we get:

$$\frac{\partial L}{\partial X_{11}} = \frac{\partial L}{\partial O_{11}} * F_{11}$$

$$\frac{\partial L}{\partial X_{12}} = \frac{\partial L}{\partial O_{11}} * F_{12} + \frac{\partial L}{\partial O_{12}} * F_{11}$$

$$\frac{\partial L}{\partial X_{13}} = \frac{\partial L}{\partial O_{12}} * F_{12}$$

$$\frac{\partial L}{\partial X_{21}} = \frac{\partial L}{\partial O_{11}} * F_{21} + \frac{\partial L}{\partial O_{21}} * F_{11}$$

$$\frac{\partial L}{\partial X_{22}} = \frac{\partial L}{\partial O_{11}} * F_{22} + \frac{\partial L}{\partial O_{12}} * F_{21} + \frac{\partial L}{\partial O_{21}} * F_{12} + \frac{\partial L}{\partial O_{22}} * F_{11}$$

$$\frac{\partial L}{\partial X_{23}} = \frac{\partial L}{\partial O_{12}} * F_{22} + \frac{\partial L}{\partial O_{22}} * F_{12}$$

$$\frac{\partial L}{\partial X_{31}} = \frac{\partial L}{\partial O_{21}} * F_{21}$$

$$\frac{\partial L}{\partial X_{32}} = \frac{\partial L}{\partial O_{21}} * F_{22} + \frac{\partial L}{\partial O_{22}} * F_{21}$$

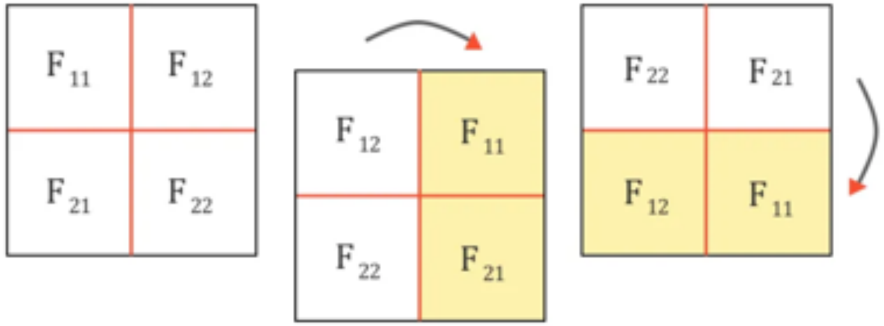
$$\frac{\partial L}{\partial X_{33}} = \frac{\partial L}{\partial O_{22}} * F_{22}$$

➢ Calculation of loss gradient w.r.t the input, **∂L/∂X:**

If we look closely, this can be represented as a **“full convolution” operation between flipped / 180° rotated Filter F and loss gradient ∂L/∂O** as shown below:

❑ the term "full convolution" is often used interchangeably with "convolution with zero-padding." In the context of convolutional neural networks (CNNs), "full convolution" typically means performing a convolution operation with zero-padding applied to the input. Here, we mean padded loss gradient ∂L/∂O.

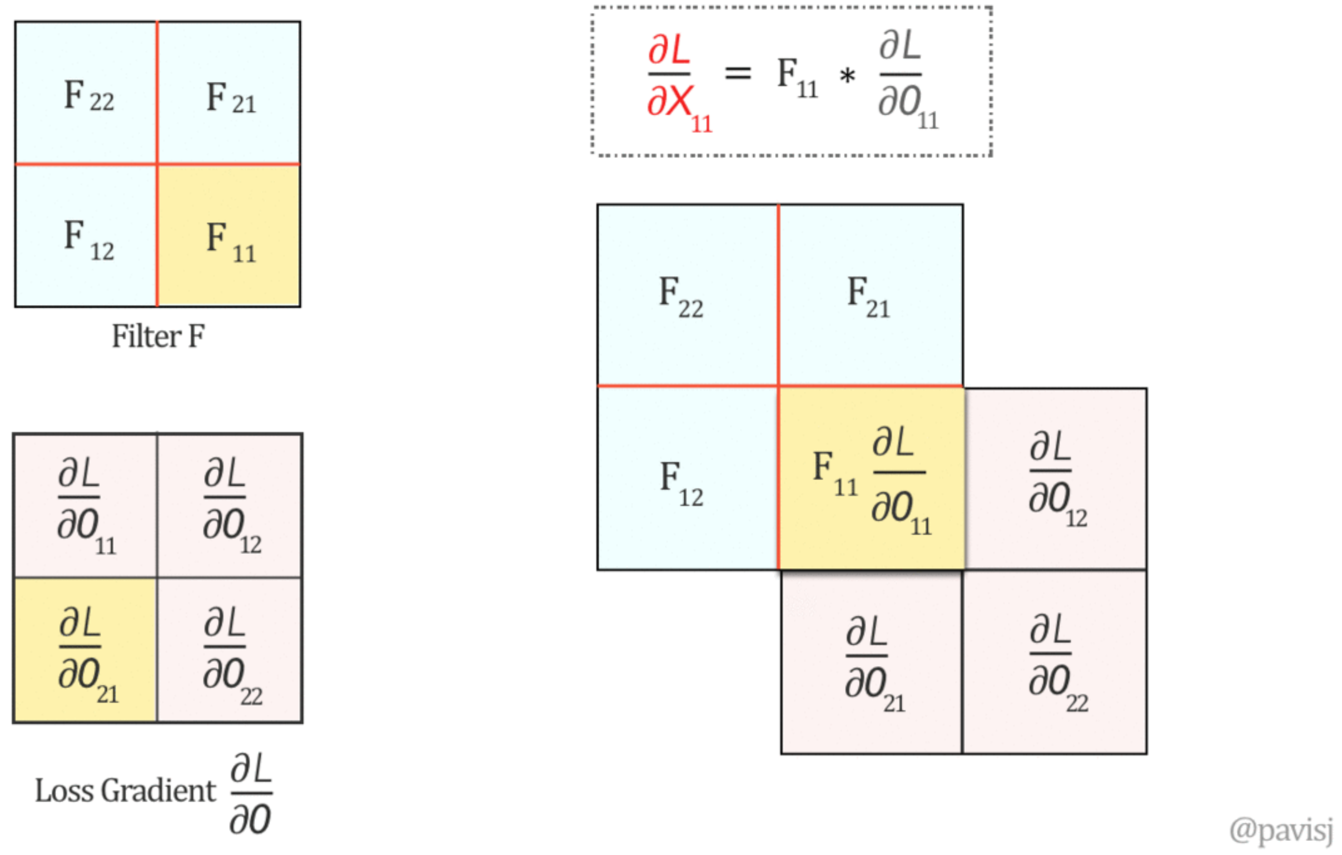
Flipping Filter F by 180 degrees



Convolution Backward Pass

➢ Calculation of loss gradient w.r.t the input, **∂L/∂X:**

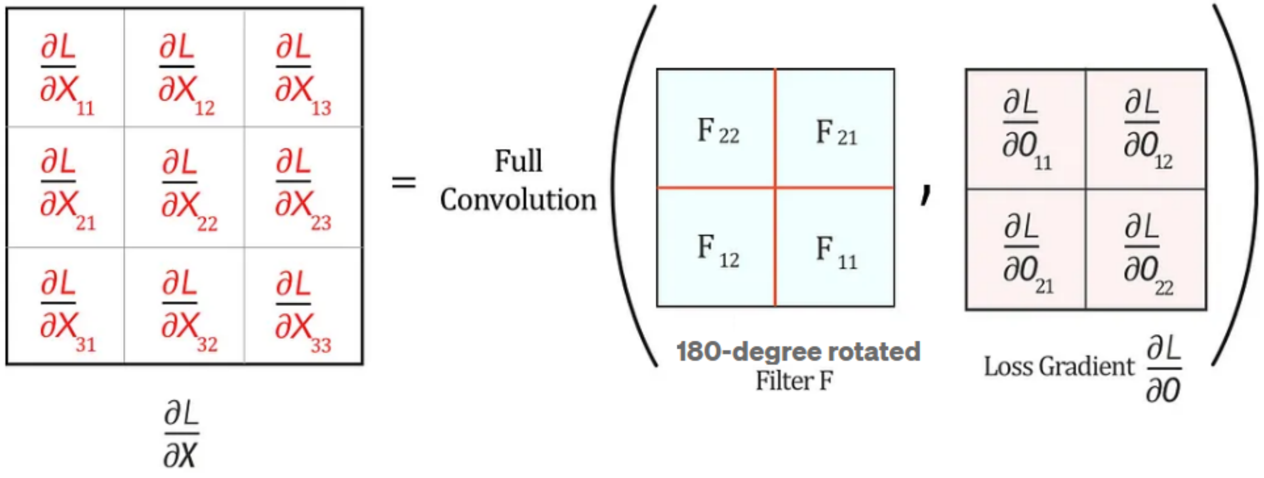
Now, let us do a ‘full’ convolution between this flipped Filter F and ∂L/∂O, as visualized below:



Convolution Backward Pass

➢ Calculation of loss gradient w.r.t the input, **∂L/∂X:**

The full convolution above generates the values of ∂L/∂X and hence we can represent ∂L/∂X as:



∂L/∂X can be represented as 'full' convolution between a 180-degree rotated Filter F and loss gradient ∂L/ ∂O

$$\frac{\partial L}{\partial X} = \text{full-conv}(180^\circ \text{ flipped F}, \frac{\partial L}{\partial O})$$

OR

$$\frac{\partial L}{\partial X} = \text{conv}(180^\circ \text{ flipped F}, \text{padded}(\frac{\partial L}{\partial O}))$$

Conclusion: Backpropagation in Convolution Layer

$$\frac{\partial L}{\partial F} = \text{conv}\left(X, \frac{\partial L}{\partial O}\right)$$

$$\frac{\partial L}{\partial B} = \text{sum}\left(\frac{\partial L}{\partial O}\right)$$

$$\frac{\partial L}{\partial X} = \text{full-conv}\left(180^\circ \text{ flipped } F, \frac{\partial L}{\partial O}\right)$$

OR

$$\frac{\partial L}{\partial X} = \text{conv}\left(180^\circ \text{ flipped } F, \text{padded}\left(\frac{\partial L}{\partial O}\right)\right)$$

In CNNs, we update the network parameters as:
F = F - α***∂L/∂F**
B = B - α***∂L/∂B**
(where α is the learning parameter)
Since X is the output of the previous layer, **∂L/∂X** becomes the loss gradient for the previous layer.

Note : This is done for a single filter F and stride = 1, CNNs have a lot more filters. The CNN Backpropagation operation with stride > 1 is identical to a stride = 1

Acknowledgements

https://youtu.be/Pn7RK7tofPg?si=bcY3RxuXhOa_GZOg

https://deeplearning.cs.cmu.edu/F21/document/recitation/Recitation5/CNN_Backprop_Recitation_5_F21.pdf

https://pavisj.medium.com/convolutions-and-backpropagations-46026a8f5d2c

UE21CS343BB2
Topics in Deep Learning

Dr. Shylaja S S
Director of Cloud Computing & Big Data (CCBD), Centre for Data Sciences & Applied Machine Learning (CDSAML)
Department of Computer Science and Engineering
shylaja.sharath@pes.edu

Ack: Divya K, Teaching Assistant