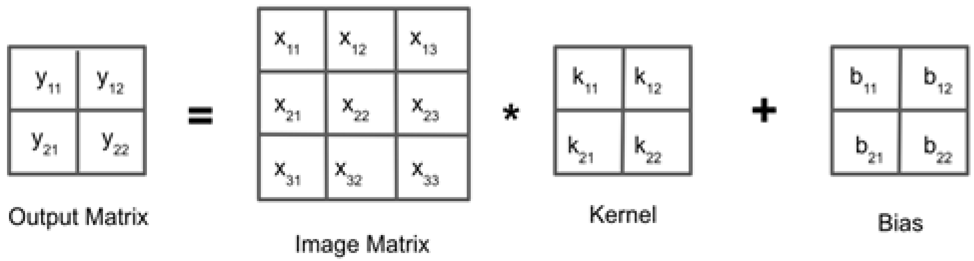
Forward propagation in CNN

The forward function iterates through every filter in every image channel and

- it pass over a kernel of specified size over the image and
- perform the convolution operation between the kernels and the input image.



$$Y_i = X_j * K_{ij} + B_i$$

Forward propagation mathematically for the Convolutional Layer is as follows:

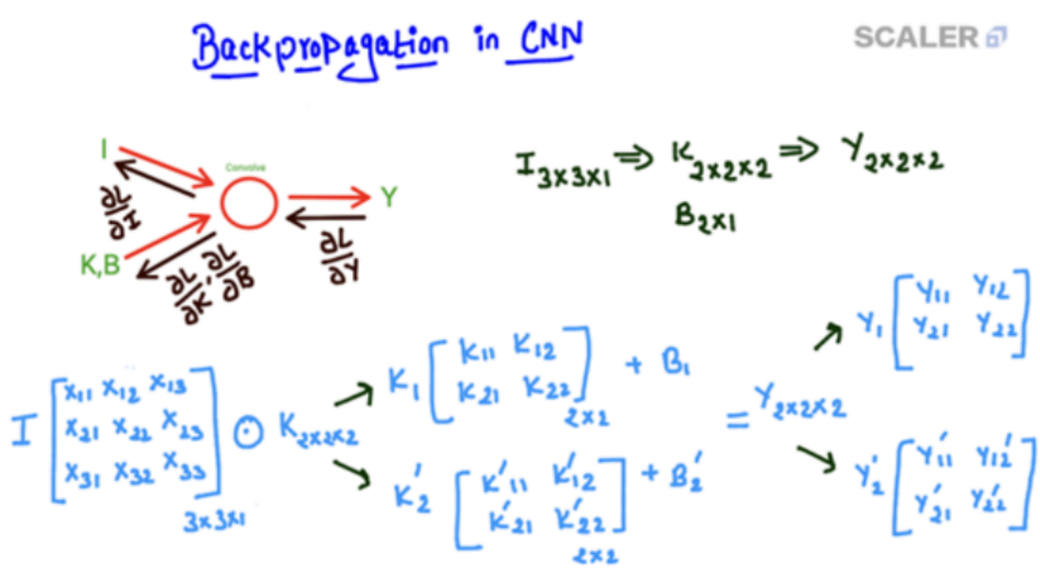
$$Y_i = \sum_{j=1}^{n} X_j * K_{ij} + B_i \quad , \qquad i = 1\ldots depth/n\_filters$$

Backward propagation in CNN

Backward propagation refers to updating the weights and biases of our model with respect to the loss of our model.

The derivative of the error with respect to:

- Kernel
- Bias
- Input is calculated during backward propagation



Loss gradients

Derivative of the Loss with respect to the Kernels

$$\frac{\partial L}{\partial K} = Conv\left(X, \frac{\partial L}{\partial Y_1}\right)$$

Derivative of the Loss with respect to the input of the layer

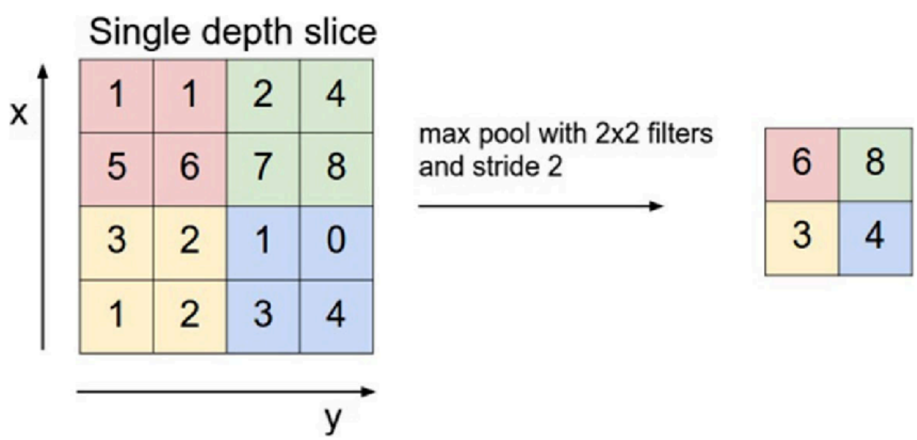
$$\frac{\partial L}{\partial X} = Conv\left(Padded\left(\frac{\partial L}{\partial Y_1}\right), 180^{\circ} Rotated Kernel K_1\right)$$

Derivative of the Loss (L) with respect to the bias of the layer

$$\frac{\partial L}{\partial B_1} = sum\left(\frac{\partial L}{\partial Y_1}\right)$$

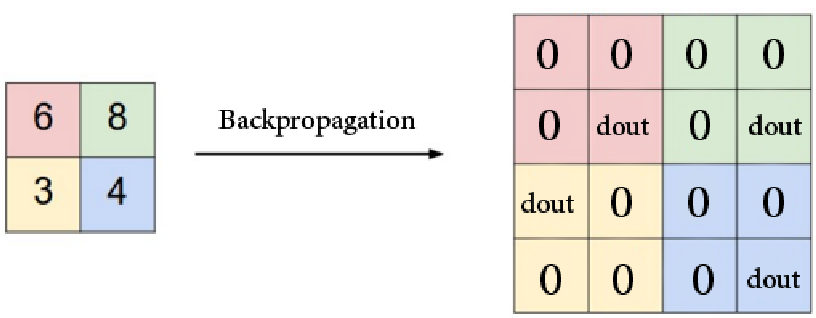
Max pooling for Forward propagation

Select the max value inside the kernel window

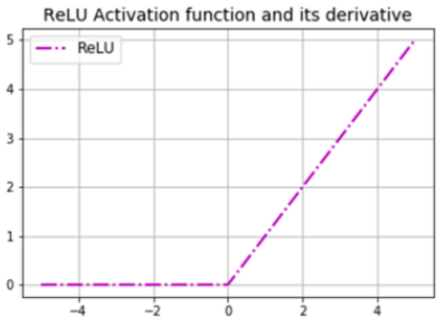


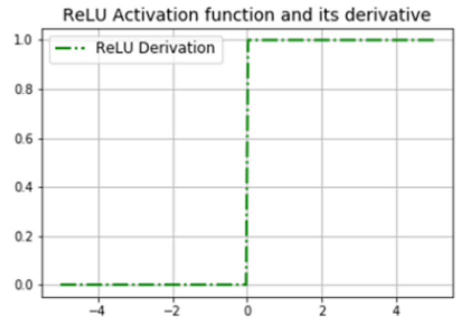
Max pooling for backward propagation

Select only the elements in our input matrix that were selected during the forward propagation of the maxpool layer, zeroing out all the other indices that were not selected during forward propagation.



Relu Activation





Forward Propagation

$$RELU(x) = \begin{cases} 0 \text{ if } x < 0 \\ x \text{ if } x >= 0 \end{cases}$$

Backward Propagation

$$RELU'(x) = \begin{cases} 0 \text{ if } x < 0 \\ 1 \text{ if } x >= 0 \end{cases}$$

Softmax Activation

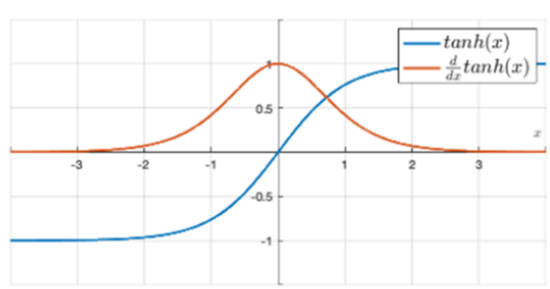
Forward Propagation

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Backward Propagation

$$\frac{\partial(Softmax)}{\partial x_1} = \frac{e^{x_1}.(e^{x_2} + e^{x_3})}{(e^{x_1} + e^{x_2} + e^{x_3})^2}$$

Tanh Activation



Forward Propagation

$$f(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$$

Backward Propagation

$$f'(x) = 1 - \tanh^2(x)$$

Log loss function

Forward Propagation

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Backward Propagation

$$\frac{1}{N}.\left(\frac{1 - y_{true}}{1 - y_{pred}} - \frac{y_{true}}{y_{pred}}\right)$$