PL are used to reduce the dimensions of the Feature map, thus, it reduces the no. of parameters to learn & the amount of computation performed in the n/w.

Formula :

$$\frac{(n_h - f + 1)}{s} \times \frac{(n_w - f + 1)}{s} \times n_c$$

n_h - height of feature map

n_w - width " " "

n_c - no. of channels " "

f - size of filter

s - stride length

) The Layers summarise the features present in a region of the feature map generated by a convolution Layer.

Pooling :

It is the key element in CNN that reduces the size of feature maps and the no of features map computations required for traning.

Types of pooling :

1) Max pooling
2) Min pooling
3) Average pooling
4) Global pooling

1) Max pooling :

Selects the maximum pixel value in a batch. It is often added to CNN after convolutional Layers to reduce the number of pixels in the output.

Formula : $y = max(x_i), \forall x_i \in R$

Eg :

$$\begin{bmatrix} 2 & 2 & 7 & 3 \\ 9 & 4 & 6 & 1 \\ 8 & 5 & 2 & 4 \\ 3 & 1 & 2 & 6 \end{bmatrix} \xrightarrow{\text{Max pool}} \begin{bmatrix} 9 & 7 \\ 8 & 6 \end{bmatrix}$$

Filter = (2×2)
stride = (2,2)

2) Average pooling:

> It computes the average of the elements present in the region of feature map covered by the filter.

> It gives the average of features present in a patch.

Formula: $y = \frac{\sum x_i}{|R|}, \forall x_i \in R$

Eg:

$$\begin{bmatrix} 2 & 2 & 7 & 3 \\ 9 & 4 & 6 & 1 \\ 8 & 5 & 2 & 4 \\ 3 & 1 & 2 & 6 \end{bmatrix} \xrightarrow[\substack{F-(2\times 2) \\ S-(2,2)}]{\text{Average Pool}} \begin{bmatrix} 4.25 & 4.25 \\ 4.25 & 3.5 \end{bmatrix}$$

3) Global pooling:

> It reduces each channel in the feature map to a single value.

> $n_h \times n_w \times n_c$ feature map is reduced to $1 \times 1 \times n_c$ feature map.

> It can be either global max pooling or global average pooling.

Q (Max pooling :) $y = \max(x_i)$, $\forall x_i \in$ Feature Map

max (2, 2, 7, 3, 9, 4, 6, 1, 8, 5, 2, 4, 3, 1, 2, 6)

o/p ⇒ 9.

Q (Average pooling). $y = \frac{\sum x_i}{F}$, $\forall x_i \in$ F.M

$$\frac{2+2+7+3+9+4+6+1+8+5+2+4+3+1+2+6}{16.}$$

o/p ⇒ 4.06.

4) Min pooling :

It is similar to max pooling but instead of selecting the maximum value from each patch, it selects the minimum value.

Formula : $y = \min(x_i)$, $\forall x_i \in R$.

Eg :

$$\begin{bmatrix} 2 & 2 & 7 & 3 \\ 9 & 4 & 6 & 1 \\ 8 & 5 & 2 & 4 \\ 3 & 1 & 2 & 6 \end{bmatrix} \xrightarrow[\substack{F-(2\times 2) \\ S-(2,2)}]{\text{Min pool}} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

Adv:

* Dimensionality Reduction
* Translation invariance
* Feature selection.

Dis - Adv:

* Information loss
* over - smoothing.
* Hyperparameter Tuning.