CNN Architecture

CNNs are a special type of neural network with a unique architecture that includes : (a) Convolution Layer (b) Pooling layer (c) Dense Layer

How CNNs process images?

→ When an image is fed into a CNN, the initial layers perform edge detection to identify basic features.

→ As the image data moves deeper into the network, subsequent convolutional layers combine these basic features to create more complex ones.

→ Finally, these complex features are used for tasks like classification.

For Ex: In face detection, the network starts by identifying edges, then forms more complex pattern like eyes & nose, and finally combines them to recognize a complete face.

Filter values in deep learning: In deep learning, the values in the filter are not set manually. They are initialized randomly and then learned automatically through **backpropagation** during the training process, similar to how weights are learned in a traditional neural network.

Ques. Why is Padding necessary?

Ans→ (a) Decreasing feature map size: Standard convolution operation reduces the size of feature map with each layer, leading to information loss.

(b) Unequal pixel contribution: Pixels on the border of the image are used in fewer convolution operations than pixel in the center, meaning their information is not as well represented in the output.

* Padding in Keras:

→ **Valid**: This is the default setting, where no padding is applied.

→ **same**: Keras automatically adds the necessary padding to ensure that the output feature map has the same dimensions as the input image.

Ques. Why are Strides necessary?

Ans→ (a) High-level feature extraction: Strides are useful when we only need to extract high level, coarse features from an image, as they cause the CNN to skip over fine-grained details.

(b) Reduced computational power: Using larger strides can speed up the training process by reducing the number of convolution operations.

Ques. What is Pooling and why is it necessary?

Ans→ A pooling layer is typically added to CNN after a convolution and activation layer. It's a downsampling operation that reduces the spatial dimensions (height and width) of the feature map. It is needed because-

(a) Memory Issues: Convolution layers can generate very large feature maps, which consume a lot of memory. Pooling helps to reduce this memory footprint, preventing potential crashes and making the network more efficient.

(b) Translation Invariance: Convolution layers can be sensitive to the location of features in the image. Pooling helps to make the model more robust to smaller translations or shifts in the position of features. This means the network can recognize a feature even if its exact location changes slightly.

Types of Pooling:

(a) Max Pooling → Selects the maximum value from the pooling window.
(b) Min Pooling → Selects the minimum value from the pooling window.
(c) Average Pooling → Calculates the average of the values in the pooling window.
(d) Global Max Pooling → Reduces the entire feature map to a single value by taking the maximum value across the entire feature map.
(e) Global Average Pooling → Reduces the entire feature map to a single value by taking the average value of all present in the feature map. Global pooling is always used at the end of a CNN, just before the fully connected layers, to reduce the number of parameters and help prevent overfitting

Advantages of Pooling:

(a) Reduced Size → Significantly reduces the dimensions of the feature maps which saves memory and computational resources.
(b) Translation Invariance → Makes the model more robust to small shifts or translations of features in the input image.
(c) Enhanced Features → By selecting the maximum value, max pooling tends to highlight the most prominent feature.
d) No training needed → Pooling operations are fixed mathematical functions (like taking the max or average) and do not have any trainable parameters, which makes them computationally fast.

Disadvantages of Pooling:

(a) Loss of information → A significant amount of information is discarded during the downsampling process.
(b) Not always suitable → For tasks where the precise location of features is important (e.g. image segmentation), pooling might not be the best choice because it sacrifices positional information.

Overfitting Prevention in training CV models

While training CV models for tasks like object detection, object classification, segmentation, etc. it becomes to prevent overfitting otherwise the model will perform well on the train dataset but will perform poorly on the test / unseen data.

We can use the same techniques we discussed during the training of simple neural networks to prevent overfitting like—

(a) Using regularization (L1 & L2)

(b) Using dropout layers

(c) Using early stopping

(d) Using batch-normalization layer.