← Go Back to Deep Learning

Course Content

ANNs vs CNNs

We have seen in the previous lecture, how ANNs can help capture the complex patterns in the data but there are some limitations as well while using ANNS.

One such limitation comes up while working with the image data. Hence, we use a special type of neural network called Convolutional Neural Networks (CNNs) that are mainly used while working with the image data.

Let's see what are the two main benefits of using CNNs over ANNs.

Local Spatiality

Local Spatiality means the relative positioning of features within the image.

If an image processing task is supposed to be done through ANN, the images are flattened to be 1-D vectors, hence, the input layer has as many input nodes as the length of 1-D vector formed after flattening the image vector.

Flattening an image vector and flattening the mirror image of the same image would give us completely different 1-D vectors. Even if we rotate our picture by certain degrees, the 1-D arrays we receive are vastly different. Since the Artificial Neural Networks accept these 1-D arrays as inputs, it fails to learn local information from the image data. Therefore, we can say that ANN doesn't do a good job of understanding the local representation or patterns in the image data, i.e., local spatiality.

On the other hand, in convolution layers, we have a filter that strides over the entire image and does the convolution operation (as discussed in the previous pre-read), so we are able to detect underlying information available from the image such as an edge or a curve and many such features. Therefore, CNNs learn the local spatiality within the image.

Weight Sharing

In Convolutional Neural Networks, a filter/detector is used to detect features from all over the image. Since each filter slides over the whole image, the weights of any filter remain the same for all the patches of the image. This is called **weight sharing**.

The greatest benefit of this is that we can think of each filter as one particular information detector. Therefore, it looks to extract that information from the entire image and is not restricted to a particular region.

As can be seen in the below graphic, the same filter is sliding all over the whole image. The original image has a size of 5 x 5 but we need only 9 weights because the size of the filter is 3 x 3. Hence, more than one pixel in the image array is receiving the same weight for each filter. So, we can say that weight sharing is happening within the convolutional layers of CNN.

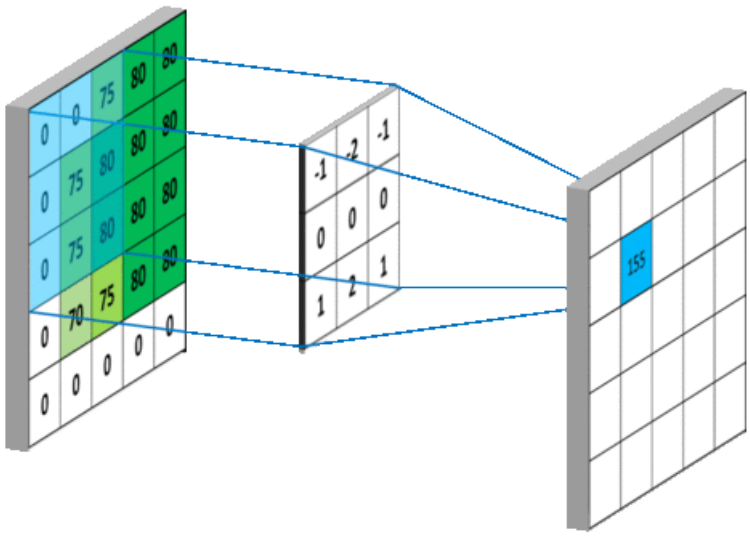

image source

However, in ANNs, each input node is multiplied by a weight of its own. Each of those weights is updated separately while

Help