Chapter 8

Convolutional Neural Networks

Most of the data generated by users these days is images and videos and there is a growing need for machines to smartly interact with this data. Some common examples would be security applications, where we want to detect certain events or objects in images and videos. Image classification and detection, which is particularly useful in image search etc. Object recognition, where we want to recognize objects in videos, this is useful in robotics, where robot have to interact with their environment.

The fundamental building block of the data for all of the above applications is an image (videos are just a sequence of images). In order to extract useful features from images we use convolutions in artificial intelligence and machine learning applications.

What is Convolution?

Mathematically convolution is defined as:

$$y(\mathbf{x}) = \int_{\mathbf{t}} x(\mathbf{t}) f(\mathbf{t} - \mathbf{x}) d\mathbf{t} \tag{8.1}$$

where **x** is the input (2 dimensional in case of an image) **f** if the filter (2D in case of an image). The mathematical definition might seem a bit complex, a less formal but easy to understand definition that I like to use is as follows: The output of the convolution operation at a point is the sum of the product of element of x and f around that point (the values of f and the size of f characterizes each unique filter).

Why Convolution?

A good question to ask here is why do we need convolutions? Why can't we just use the extended formulation of neural networks here, and should that be sufficient for images? The answer is, yes, we can extend the neural networks to images, where we just flatten the image and use it as an input vector. This kind of architecture will be able to learn important feature in images but there are a few major drawbacks of this approach that we highlight below.

Drawbacks of Neural Networks for Images:

- Using neural network setup for image will require an excruciatingly large number of parameters. Suppose we have an image of size 256 × 256 and we want to extract 10 channels from this image each where the image is down-sampled by a factor of two. The first layer of our neural network will have 256*256*10*128*128 ≈ 10,000,000,000 parameters. This if order of magnitudes larger then the state-of-the-art CNNs we use in the field.

- The reduction in parameters in the CNN's don't necessarily come with a decline in performance. The reason for this is that in images we might have repeated patterns, hence we can use same filter at multiple points in the image to extract these patterns (neural networks are incapable of doing this, CNNs do it).

- CNNs provide better in-variance properties in images then NNs.

The details expressions of convolutions and the gradients of convolution operations are beyond the scope of this course work module. However, these are simple extensions of what we did for simple neural networks.