ImageNet Classification with Deep Convolutional Neural Networks

Seminar

Sidharth S

November 15, 2020

Rajagiri School of Engineering and Technology

- AlexNet
- One of the most influential papers published in computer vision.
  - AlexNet paper has been cited over 70,000 times according to Google Scholar.
- The network achieved a top-5 error of 15.3%.
- It is the first CNN where multiple convolution operations were used.

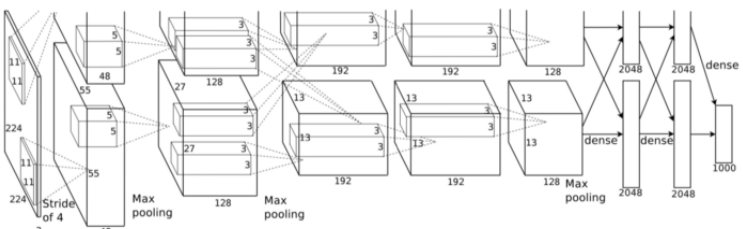

Figure 1: AlexNet Architecture

Abstract

- Deep convolutional neural network to classify the 1.2 million high-resolution image.
- 60 million parameters and 650,000 neurons.
- Convolutional Layers, Maxpooling layers, Fully connected layers and Softmax
- GPU implementation of the convolution operation.
- Dropout regularization.

Fully Connected Neural Networks

Fully connected neural networks (FCNNs) are a type of neural network where the architecture is such that all the nodes, or neurones, in one layer are connected to the neurones in the next layer.

For a $64x64x3$ image,
No of parameters in input layer = $12,288$

For a $225x225x3$ image,
No of parameters in input layer = $151,875$

- Networks having large number of parameter face several problems, for e.g. slower training time, chances of overfitting e.t.c.

Convolution

In mathematics, convolution is a mathematical operation on two functions f and g that produces a third function $f * g$ that expresses how the shape of one is modified by the other.

- The main image matrix is reduced to a matrix of lower dimension in the first layer itself
- The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction.

How Convolution is performed

We can use an input image and a filter to produce an output image by convolving the filter with the input image.

Steps:

1. Overlaying the filter on top of the image at some location.
2. Performing **element-wise multiplication** between the values in the filter and their corresponding values in the image.
3. Summing up all the element-wise products. This sum is the output value for the destination pixel in the output image.
4. Repeating for all locations.