Convolutional Neural Networks and LeNet

#computervision #CNN

Introduction:

A convolutional neural network consists of a 3D kernel and a 3D input. A convolution is basically a rotation of kernel and the performance of the cross correlation operation on the input. Their is a bias associated with each kernel. Each Kernel has a set of weights, which are learnt in the CNN. Each Kernel is used to detect a specific characteristic of the input. A Convolutional layer usually has more than 1 kernel to extract different kind of features from the input image. Convolution is a sliding window operation.

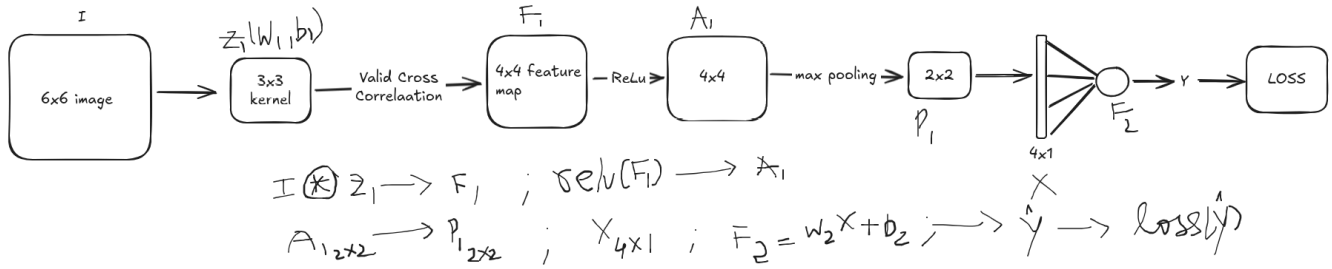
The output matrix is called the feature matrix. The dimensions of the feature matrix depends upon the type of cross correlation employed or the stride length. In a Full Cross Correlation the dimensions of the output matrix are given by the formula:

$$\text{output} = \text{input} + \text{kernel} - 1$$

The other type of cross correlation is called Valid Cross Correlation.

$$\text{output} = \text{input} - \text{kernel} + 1$$

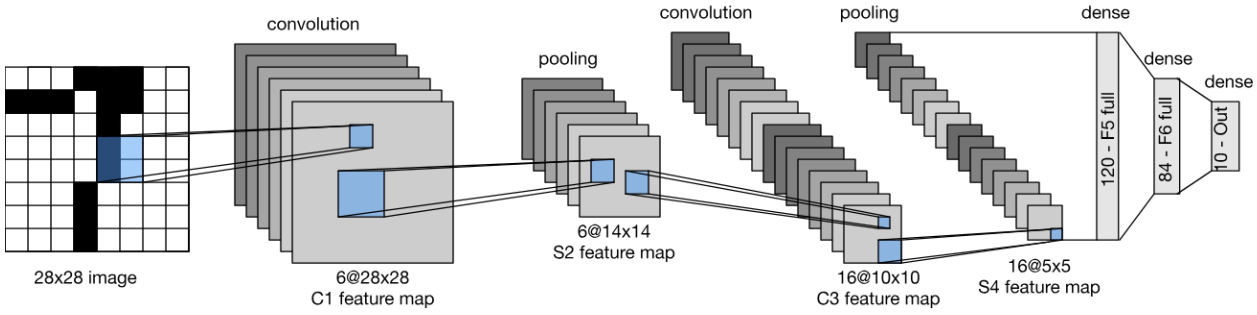
Forward Propagation through a Convolutional Layer:



Drawback of Deep Convolutional Neural Networks:

- When deeper networks start converging, a degradation problem is noticed. Degradation is basically loss of training accuracy. With the networks getting deeper, the accuracy gets saturated and then degrades rapidly. Such degradation is not caused by overfitting. Adding more layers to a suitably deep model leads to higher training error. The degradation error says that not all systems are easy to optimizer. (Deep Residual Learning for Image Recognition, He et al., 2015)

LeNet:



LeNet by Yan Le Cun was the breakthrough paper which introduced convolutional neural networks. It was a 7 layer CNN designed for handwritten digit classification. This research paper also popularized the MNIST dataset. The network was designed for 32x32x1 gray scale images.

Architectural Breakdown:

Layer Name	Layer Type	Details	Activation	Output Shape
Input	Input Image	32x32 Grayscale Image	-	`32x32x1`
C1	Convolutional	6 filters @ 5x5, stride 1	`tanh`	`28x28x6`
S2	Average Pooling	2x2 window, stride 2	-	`14x14x6`
C3	Convolutional	16 filters @ 5x5, stride 1	`tanh`	`10x10x16`
S4	Average Pooling	2x2 window, stride 2	-	`5x5x16`
F5 (C5)	Fully-Connected	120 neurons	`tanh`	`120`
F6	Fully-Connected	84 neurons	`tanh`	`84`
Output	Fully-Connected	10 neurons	`Softmax`	`10`

- The `ReLU` activation is better than `tanh`, but during the time of the paper, ReLU wasn't popular.
- It established the classic and effective pattern of alternative convolutional and pooling layers.
- Unlike modern network which often use max pooling, LeNet used average pooling.
