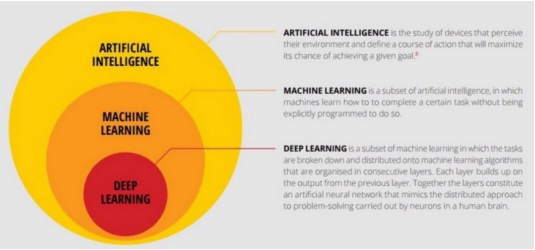
Deep learning

April 2020

What is deep learning?

- a sub-field of machine learning dealing with algorithms inspired by the structure and function of the brain called artificial neural networks



- it mirrors the functioning of our brains
- similar to how nervous system structured where each neuron connected each other and passing information

Deep Convolutional Neural Networks

(ConvNets) are deep artificial neural networks used:

- to classify images (e.g. name what they see)
- cluster them by similarity (photo search)
- perform object recognition within scenes

algorithms that can identify faces, individuals, street signs, tumors, perform optical character recognition (OCR).

Deep Convolutional Neural Networks

- pre-processing required in a ConvNet is much lower as compared to other classification algorithms
- architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain - Visual Cortex
- a ConvNet captures the spatial and temporal dependencies in an image

Tensors

An n^{th} - rank tensor in m-dimensional space is a mathematical object that has n indices and m^n components and obeys certain transformation rules.

- generalizations of scalars, vectors, and matrices to an arbitrary number of indices.

-used in physics such as elasticity, fluid mechanics, and general relativity.

Notations for tensors

- $a_{ijk\ldots}$, $a^{ijk\ldots}$, $a_i^{jk\ldots}$, etc., may have an arbitrary number of indices
- a tensor (rank $r+s$) may be of mixed type (r,s): r "contravariant" (upper) indices and s "covariant" (lower) indices.

- the positions of the slots in which contravariant and covariant indices are placed are significant!

$$a^{\lambda}_{\mu\nu} \neq a^{\nu\lambda}_{\mu}$$

Convolution operation

an operation on two functions $x(t)$ (**input**) and $w(t)$ (**kernel**) of a real - valued argument

$$s(t) = \int x(a)w(t-a)da$$

notation:

$$s(t) = (x * w)(t)$$

if t is discreet the integral turns into a sum

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a)$$

Convolution operation

in practice we have two tensors:

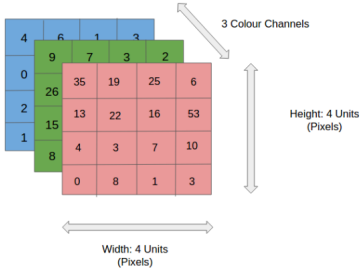
- the input - a multidimensional array of data
- the kernel - a multidimensional array of parameters (adapted by the learning algorithm)

- stored separately $\longrightarrow$ that these functions are zero everywhere but in the finite set of points
- we can implement the infinite summation as a summation over a finite number of array elements
- convolutions can be over more than one axis at a time

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)$$

Images as tensors

ConvNets ingest and process images as tensors.



- reduce the images $\longrightarrow$ form easy to process (without losing critical features)

Using a kernel

we want apply a filter over the image

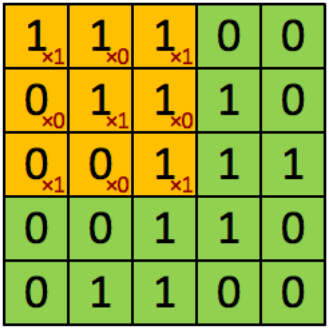
AIM: **extract the high-level features** (edges, color, gradient orientation)

Example: *Image Dimensions = 5 (Height) x 5 (Breadth) x 1 (Number of channels, eg. RGB)*

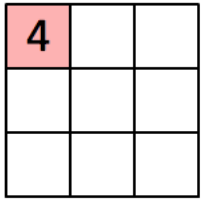
Kernel / filter:

$$K = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Using a kernel

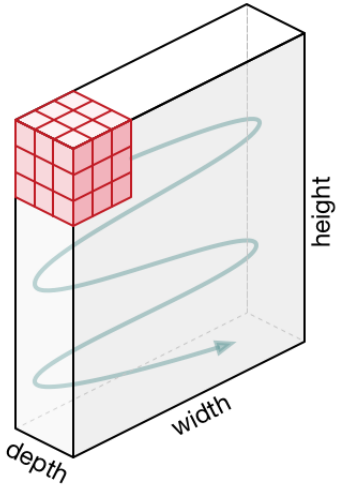


Image



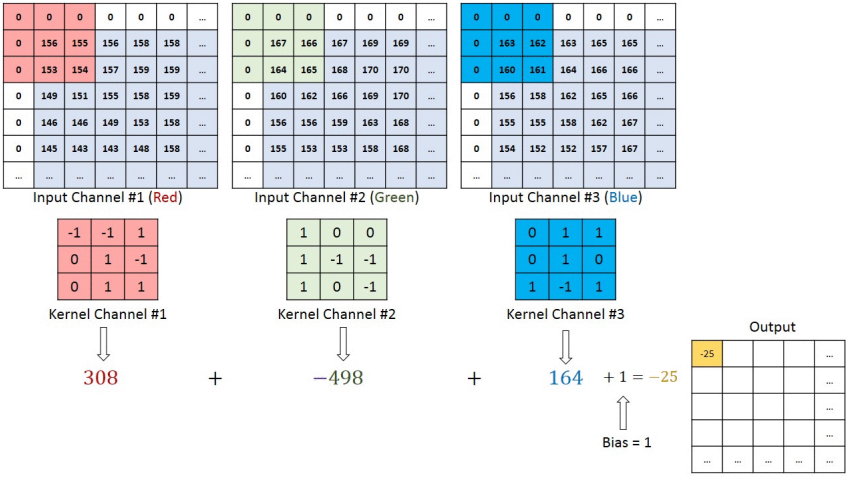
Convolved Feature

Using a Kernel - movement of the Kernel



- Kernel shifts 9 times because of Stride Length = 1 (Non-Strided)
- every time performing a matrix multiplication operation between K and the portion P of the image over which the kernel is hovering

Using a kernel - 3 channels



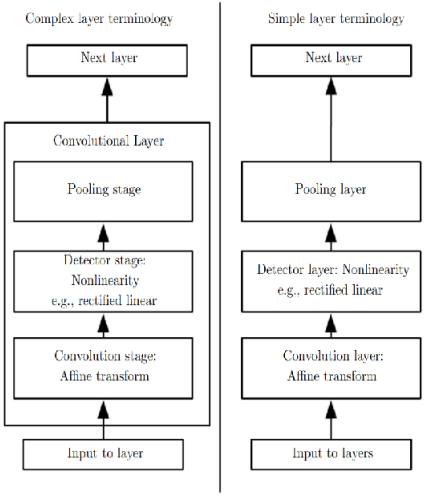
Layers of a convolutional network

Typical three stages:

- first: several convolutions in parallel
- second: a nonlinear activation function (ex: rectified linear activation function) - **detector stage**
- third: **pooling** function to modify the output

Examples: *max pooling* (Zhou and Chellappa, 1988 - maximum output within a rectangular neighborhood, the average of a rectangular neighborhood, L^2 norm of a rectangular neighborhood, a weighted average based on the distance from the central pixel.

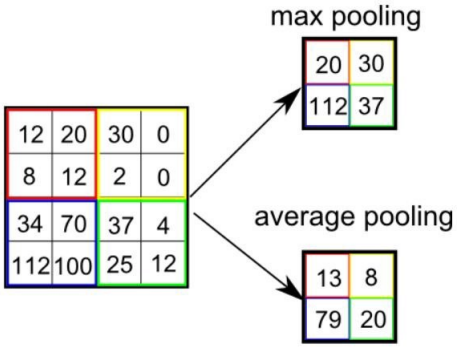
Example of ConvNet Layer



Pooling

helps to make the representation approximately invariant to small translations of the input
- useful property if we care more about whether some feature is present than exactly where it is
- essential for handling inputs of varying size
- Some guidance as to which kinds of pooling one should use in various situations : Boureau et al., 2010.

Pooling



- Max Pooling - returns the maximum value from the portion of the image covered by the Kernel
- Average Pooling returns the average of all the values from the portion of the image covered by the Kernel